UNIVERSITÀ DEGLI STUDI DI PAVIA
FACOLTÀ DI SCIENZE MATEMATICHE FISICHE E NATURALI
DOTTORATO DI RICERCA IN FISICA

Architectural studies of the Event Filter in the ATLAS experiment and development of a *b*-tagging selection strategy

Tesi di dottorato di
Andrea Negri

XIII CICLO

PREFACE

A new generation of High Energy Physics (HEP) experiments will start operating in the year 2005 at the future Large Hadron Collider (LHC), which is currently under construction at CERN, the European Laboratory for Particle Physics. LHC is designed to collide high energy beams of protons or heavy-ions providing insight in the particle physics at the TeV energy scale. In the four interaction points four experiments will be installed : two high luminosity general purpose *pp* detectors (ATLAS and CMS), a heavy-ion detector (ALICE) dedicated to study the physics of strongly interacting matter at extreme energy densities and a fixed target beauty physics experiment (LHC-b) designed to make precise measurements of CP violation and rare B-decay.

The ATLAS (A Toroidal LHC ApparatuS) experiment has been designed to exploit the full physics potential provided by the head-on collision of the very intense proton beams. The Chapter 1 is dedicated to the description of the accelerator, the detector and the ATLAS physics potential.

The 7 TeV proton beams will collide each 25 ns corresponding to 10^9 *pp* interactions per second, most of which, however, will produce events of little interest for high energy physics. The detector has hence an huge on-line data processing challenge. The data acquisition system of the ATLAS experiment must be able to process, after each particle collision, the detector data produced by more than 10^7 electronic channels, discarding less interesting events and saving the most interesting ones. The required data reduction factor, equivalent to a rejection factor of about 7 orders of magnitude, is achieved on-line via a data acquisition system organized in different levels (triggers). Each level refines the decisions made at the previous one and, where necessary, applies additional selection criteria. The time available for event processing increases in each level of the trigger, allowing for the use of an increasing amount of information to either accept or reject the event. The first level of trigger (LVL 1), realized in hardware by custom electronics, reduces the data rate from the 40 MHz LHC rate to about 75 kHz. The High Level Triggers (LVL 2 and Event Filter), implemented on two different computer farms, provide a further reduction factor of about 10^3. The LVL 2, which operates on a limited subset of the data, reduces the trigger rate to a level which can be sustained by the event building system (in the range of few kilohertz). This collects, from the ~1000 read-out units, the data fragment of LVL 2 accepted events and merge them in a single location. The Event Filter performs the final selection analysing the full assembled events. The computing instrument of the Event Filter is organized as a set

of independent sub-farms, each connected to one output of the Event Building switch fabric. Each sub-farm comprises a number of processors analysing several complete events in parallel. The description of the ATLAS data acquisition system architecture and its challenges is the subject of the Chapter 2.

A pre-prototype programme, called the DAQ/EF-1 project, has been developed in order to design and implement a full "vertical" slice of the functional DAQ and EF architecture outlined in the ATLAS Technical Proposal. The global DAQ/EF-1 project is presented in Chapter 3, while Chapter 4 is dedicated to the Event Filter part of the project. In order to validate the global Event Filter architecture and to evaluate the requirements of the various subsystem elements three different EFF prototypes were implemented and assembled based on three somewhat different configuration: Commodity PCs, Intel Commodity Multi-Processors and Commercial Symmetric Multi-Processor (SMP).

The latter is the major subject of this thesis. In Chapter 5 the design and implementation of the SMP prototype are presented together with the performance, robustness and scalability tests made on different hardware platforms. The first part of Chapter 6 is dedicated to the description of prototype integration in the global DAQ/EF-1 project, to the comparison of the different prototypes and to the presentation of the main common conclusion. This has led to a proposed architecture for the EF system that has been published in the *'ATLAS High-level Triggers, DAQ and DCS Technical Proposal'*. In the same Chapter questions concerning the implementation of a common software framework for the EFF data-flow are addressed. This will permit the software integration of the different existing prototypes and the exploration of new protocols or architectures.

Event filtering is executed on fully assembled events, containing the entire detector information, including the results of the lower level triggers and the on-line calibration and alignment constants. The selection is based on the off-line algorithms, which, obviously, cannot be executed earlier. Depending on the processing time needed by the algorithms, the processing power available, and the stability of the calibrations and beam position, one may aim at completing in the EF the reconstruction to such a degree that the subsequent analysis steps have only to perform hypothesis-dependent updates of the reconstruction. In Chapter 7, after an overview of the ATLAS trigger strategy, an application of the Event Filter selection capabilities is described, addressing the identification of jets with *b* content.

Table Of Contents

CHAPTER 1

The ATLAS experiment at the Large Hadron Collider

1.1 The Large Hadron Collider

The crowning achievement of the past fifty years of theoretical and experimental studies in High Energy Physics (HEP) has been the construction of the Standard Model (SM) of elementary particles, which embodies the summary of our present understanding of the elementary processes and building blocks of our physical universe. A central role in the uncovering of the SM has been played by the particles accelerators, which provide us with experimental insights on the universe that could not plausibly be obtained in other ways. The experiments at these accelerators have shown the intrinsic naivete of any pre-accelerator picture of the Universe, that had the proton, neutron, electron and photons as the only elementary particles, and they have also continuosly highlighted the limitation of our knowledge. The discovery potential of the next energy frontier collider will be essential to achieve a more satisfying level of understanding about the physical foundation of universe. Figure 1-1 is the famous Livingstone plot showing the historical exponential growth with time in the energy reach of both lepton and hadron collider. Energy frontier hadron collider are gen-

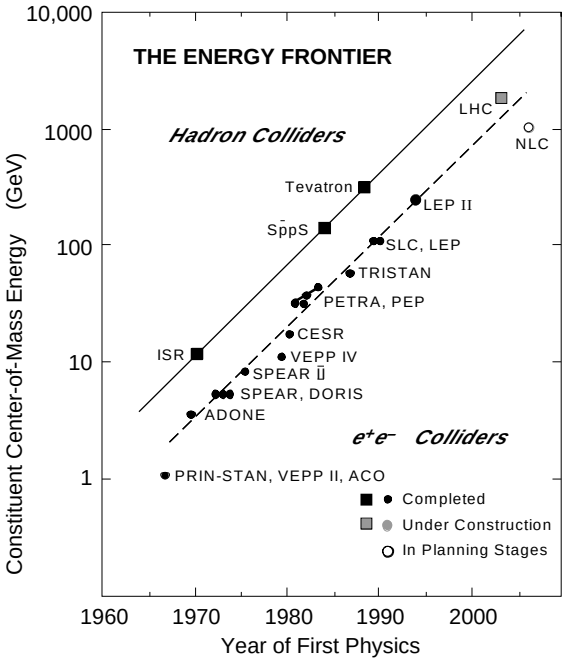

Figure 1-1 The energy frontier of particle physics. The effective constituent energy of existing and planned colliders versus the year of first physics results from each[1-2].

erally regarded more as discovery machines, while the lepton colliders, that operate in a cleaner experimental conditions, are considered mainly as follow-up machines for precision studies.

The Large Hadron Collider [1-1] is the next step for hadron collider evolution; it is presently under construction at CERN of Geneva and will start operation in 2005. The TeVatron collider, located at the Fermilab of Chicago, is the present machine with the highest centre of mass energy ($\sqrt{s}$). It accelerates proton and antiproton up to a momentum of 1 TeV providing an effective centre of mass energy[1] ($\sqrt{\hat{s}}$) of ~350 GeV. LHC is planned to provide two proton beams with an energy of 7 TeV (a factor about seven larger than the TeVatron) and therefore it will be able to surpass the TeV threshold in the effective centre of mass energy. The opening of a window in this new energy domain for the experimental research could provide important answers to the fundamental problems in the high energy physics. A key feature of the LHC will be the challenging planned luminosity of $10^{34}\ cm^{-2}s^{-1}$, which is about 100 times greater than previous machines (TeVatron: $2.1 \times 10^{32}\ cm^{-2}s^{-1}$, $Sp\bar{p}S\ 6 \times 10^{30}\ cm^{-2}s^{-1}$, LEP $10^{32}\ cm^{-2}s^{-1}$ [1-3]).

1.1.1 Motivation for LHC

There are many physics reasons to suppose that new phenomena could appear in the mass range comprised between the electro-weak scale (about 100 GeV) and the TeV domain. Despite the success of the Standard Model (SM) of elementary particles, whose predictions have been verified at the level of 0.1%[1-4] or better by the LEP, SLC and Tevatron experiments, some aspects of the theory are still obscure.

A still open question concerns the origin of the particle masses and the motivation for the mass hierarchy of leptons, quarks and gauge bosons. In the SM a way to break the electro-weak symmetry and, therefore, to endow particles with masses, is provided by the Higgs mechanism which entail, as a by-product, a new scalar particle, the Higgs boson. The mass of this particle is not predicted by the model but should not exceed about 1 TeV to preserve unitarity of the theory at high energy (the present theoretical bounds on the Higgs boson mass are outlined in Figure 1-2 [1-5]). Indirect experimental bounds for the Higgs mass are obtained from fits to precision measurements of electroweak observables and to the measured top and $W^{\pm}$ masses. Currently the best fit value is $m_H = 77^{+69}_{-39}$ and $m_H < 215$ GeV is obtained at 95% confidence level [1-3], still consistent with the SM being valid up to the GUT scale.

The ALEPH collaboration has recently announced [1-7] (paper submitted to Physics Letter B) the observation of an excess of events beyond the background expectation consistent with the production of a Higgs boson with a mass of about 114 GeV. The observed statistical significance of the effect, although in agreement with that expected from such a signal, is just about 3 σ, which is still to low to claim that a statistical fluctuation of the background is excluded (see Figure 1-3). Given the definitive LEP shutdown, the Higgs boson search will be pushed at the TeVatron, which is about to start the second phase of its evolution (Run II), but where the experimental conditions and the planned luminosity makes the 5 σ discovery limit a difficult goal to reach. Therefore a new machine could be needed to provide an adequate statistic sample for such a signal and to explore the mass range from 120 GeV up to the theoretical upper bound of about 700 GeV, if the LEP observation will be not confirmed. The present official lower limit on its mass, provided by the LEP experiments, in about 108 GeV[1-6]. CAMBIARE

1. The effective centre of mass energy of the quarks and gluons interaction is given by $\sqrt{\hat{s}} = \sqrt{x_1 x_2 s}$, where x_i are the fraction of the proton momentum carried by the two colliding partons.

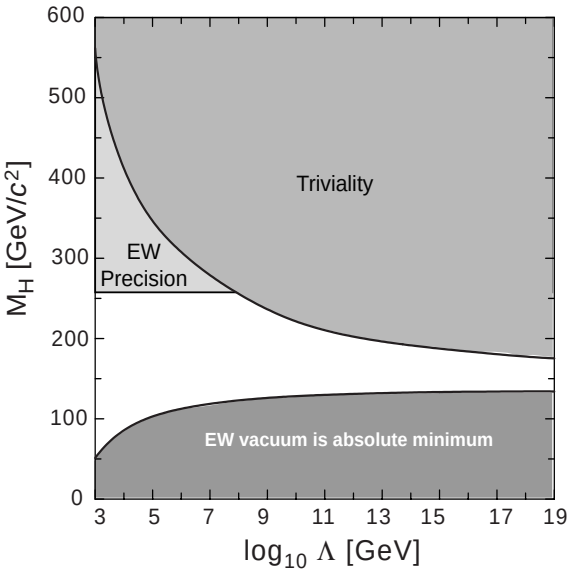

Figure 1-2 Bounds on the Higgs mass that follow from requirements that the electro-weak theory be consistent up to the energy scale Λ. The upper bound follows from triviality conditions; the lower bound follows from the requirement that the electro-weak vacuum is an absolute minimum. Also shown is the range of masses permitted at the 95% confidence level by precision electro-weak measurements [1-5].

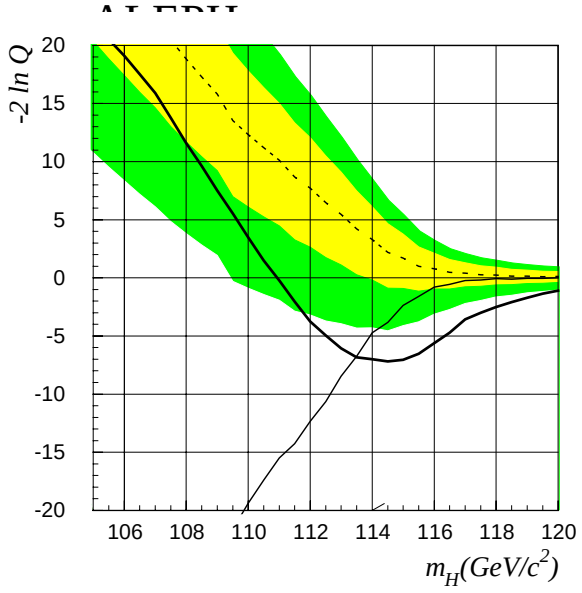

Figure 1-3 The negative log-likelihood ratio (a measure of the compatibility of an experiment with a given hypothesis) as a function m_H. The shaded bands represent the 1 σ and 2 σ compatibility bounds with the background-only hypothesis, while the dash-dotted curve is the expectation in presence of the signal. The full line curve, which reproduces the observe result, develops a minimum at ~114 GeV, with a similar statistical significance, in agreement with that expected from a Higgs boson signal at this mass.[1-7]

Even if there are no indications on possible faults of the SM at energy greater than the electro-weak scale, there are some experimental observations that could indicate underlying new theories beyond the SM:

- the gauge symmetry of strong interaction allows the presence of a CP-violating term in the lagrangian density that describes the theory; the fact that the consequences of this term are not experimentally observed could suggests the presence of a still unobserved symmetry;
- the analysis of the galaxies' rotational velocity implies that about 90% of the mass of the universe could be not barionic; this could denote the existence of new stable neutral particles;
- the SM is not able to justify the value of CP-violation needed to explain the observed asymmetry (~ 10^9) between matter and antimatter;
- the observed number of solar neutrinos is significantly less that the value predicted by stellar models based on the SM; this could implies news in the neutrino physics;

There are, also, several reasons to believe that the SM is not the ultimate theory of particle interactions. Despite its predictive power for existing HEP experiments, the SM appear to be no more than a stop-gap theory with a limited domain of applicability. It is phenomenological rather than fully predictive, incorporating 19 free parameters, which could be only experimentally determined. It is reasonable to think that the number of degree of freedom is to big for a fundamental theory. Furthermore in the SM the strong and the electro-weak force are not unified, it is

impossible to incorporate the gravity and it becomes logically inconsistent when one tries to extrapolate it to experimentally inaccessible energy scale. When one tries to extend the standard model to a more general symmetry, with a unification scale (M_{GUT}~10^{16} GeV or M_{Planck}~10^{19} GeV) much higher than the vacuum expectation value of the Higgs boson, appears the so called "*naturalness problem*". Indeed the introduction of the Higgs mechanism in such a theory leads to divergent radiative corrections to the Higgs mass unless "unnaturaless" fine-tuned cancellation occurs. This problem is elegantly solved in supersymmetric theories (SUSY) via the introduction of a new symmetry between barionic and fermionic states. In order to explain the stability of the electro-weak scale in presence of radiative corrections the mass difference between ordinary particles and the respective supersymmetric partners must be *O*(1 TeV). Therefore the SUSY theories foresee the existence of a zoo of new particles in the mass range comprised between the actual experimental limits and some hundreds of the GeV. The present limits on the masses of charginos and neutralinos (χ_1^0 excluded) are provided by e^+e^- collider experiment and are of the order of $\sqrt{s}/2$ (because the particles are produced via pair production), ~100 GeV at LEP. The experimental limit on squarks are about 250 GeV and are provided by TeVatron [1-3].

Therefore a collider able to provide effective centre of mass energy of the order of 1 TeV could:

- check the validity of the Higgs mechanism in the SM via the search of the Higgs boson in the whole mass range theoretically expected;
- discover the hypothetical supersymmetric particles;
- search new quarks or leptons;
- verify the possible existence of new physics at the electro-weak scale;
- search hypothetical new gauge bosons *Z'* and *W'* with mass greater than 1 TeV;
- perform precision measurements on the known particles and interactions in order to observe possible deviations from the SM predictions;

and, of course, discover new physics not foreseen by any model.

1.1.2 The collider

The LHC project has been designed in order to exploit as much as possible the existing CERN accelerator resources. It will be located in the tunnel currently housing the Large Electron Positron collider (LEP [1-3]), after the dismission of the latter scheduled for the end of 2000, and, as outlined in Figure 1-4, it will use the present CERN's accelerator complex in order to achieve the required collision energy. The LINAC, the Booster, the Proton Synchrotron (PS) and the Super Proton Synchrotron (SPS) will be used to gradually accelerate the collision particles and finally inject them into LHC. The impossibility to obtain to required luminosity with a $p\bar{p}$ collider, because of the difficulty to produce a sufficiently intense antiproton beam, led to the choice of a *pp* collider. But in order to collide two beams of equally charged particles they must circulate in separate and opposite magnetic lines. In the LEP tunnel there is hardly room for two separate magnetic rings. A solution to this problem was found using a twin-aperture magnet with two coils and beam channels using the same mechanical structure and cryostat. This solution has proved to be an economical alternative to two separate rings and allows also enough free space in the existing tunnel for a possible future reinstallation of a lepton ring above the LHC ring for e-p physics.

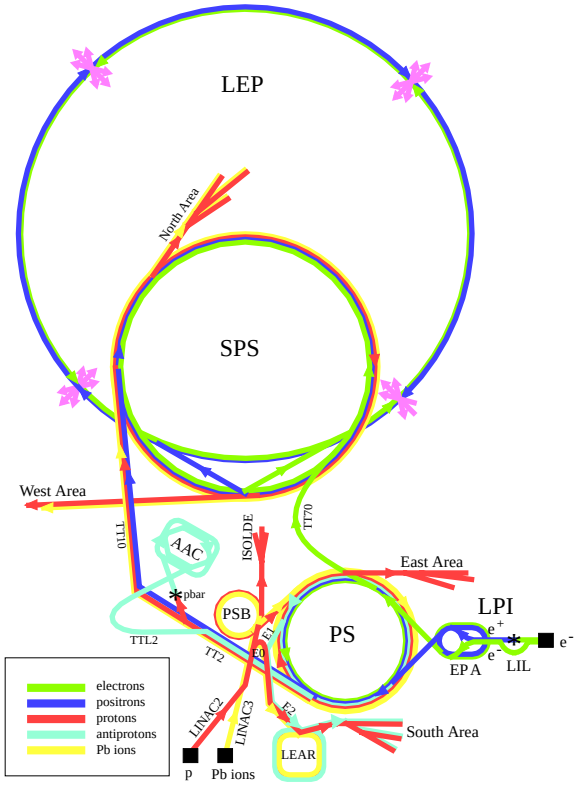

Figure 1-4 The present CERN's accelerator complex. The LHC injector scheme will be the following: the protons will be produced and accelerated to 50 MeV by the proton Linac, before being injected into the 1.4 GeV Proton Synchrotron (PS) booster (PSB). The PS itself will accelerate the protons to 26 GeV and finally the Super Proton Syncrotron (SPS) will inject protons at 450 GeV into the LHC, placed in the LEP tunnel.

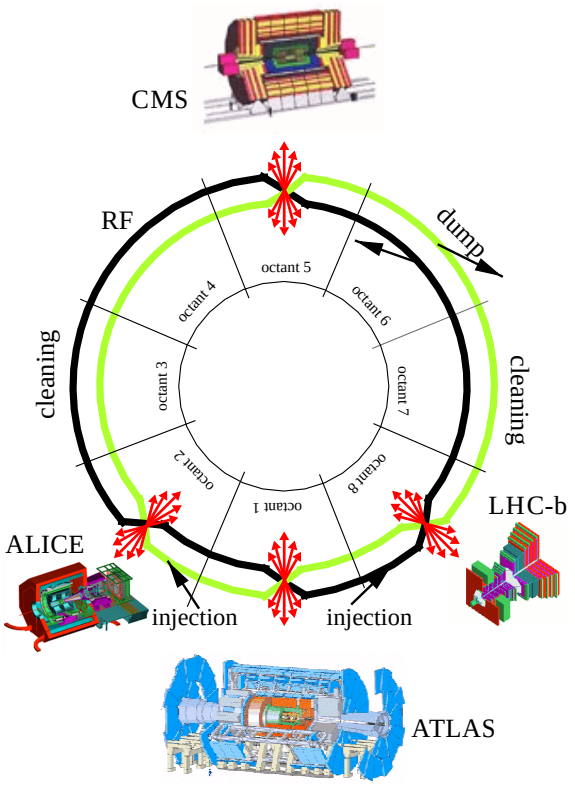

Figure 1-5 The basic layout of the LHC: experimental insertion and utilities are located on eight long straight sections, each approximately 500 m in length. The two high luminosity insertion, located at diametrically opposite straight sections, will house the two general purpose experiment, ATLAS and CMS. ALICE, optimized for heavy ion collision, and LHCb will be located at the other two collision points.

The aim of LHC, which main parameters are summarized in Table 1-1, is to provide centre of mass energy greater than 1 TeV at parton level. The centre of mass energy achievable is limited by the bending power needed to keep the beams circulating in the 27 Km circumference of the LEP tunnel. The most powerful magnets used in the accelerator field are the superconducting dipoles installed in the HERA *ep* collider. These magnets, working at a temperature of 4.2 K, provide a field of 4.62 T. To go beyond this field strength, the LHC magnets will be cooled with superfluid Helium down to 1.9 K, below the lambda point of the Helium. These serious technological challenges, to both the magnet designer and cryogenic engineers, will allow a magnetic field of 8.4 T. More than 1200 of such superconducting dipoles will lead to the required centre of mass energy of 14 TeV. Furthermore the LHC will be the first hadron machine to produce appreciable synchrotron radiation (3.6 kW of irradiated power) which, together with the heat load due to image currents, has to be absorbed at cryogenic temperatures.

As shown in Figure 1-6, at centre of mass energy of 14 TeV the interesting process will have a very small cross-section respect to the foreseen total inelastic *pp* cross-section (σ_{pp}) of about 100 mb. The Higgs boson production cross-section is more than a factor of ten lower. To detect such signals the luminosity (L), defined as the ratio between the rate and the cross section ($L=R/\sigma$), must be maximized. Because the signals with the clearest signature, e.g. the decay of

Table 1-1 Main parameters of the LHC compared with the best current collider[1-10]: the Large Electron Positron Collider (LEP), which will leave its tunnel to the LHC, and the Tevatron *pp* collider, which represent the benchmark for the LHC.

Parameters	Units	LEP	TeVatron	LHC
Particles collided		e^+e^-	$p\bar{p}$	pp
Circumference	Km	26.66	6.28	26.66
Beam energy	TeV	0.1	1	7
Luminosity	$\times 10^{32}$ cm^{-2}s^{-1}	1	2.1	100
Time between collisions	ns	22000	396	25
Bunch length	cm	1	38	7.5
Luminosity lifetime	h	10	7÷30	10÷20
Particles per bunch	$\times 10^{10}$	45÷60	p: 27, $\bar{p}$: 7.5	10.5
Bunches per ring per species		4÷8	36	2835
Beam current per species	mA	4÷6	p: 81, $\bar{p}$: 22	536
Dipoles in ring		3280	774	1232
Peak magnetic field	T	0.135	4.4	8.3

the Higgs boson in 4 muons, are very rare (the branching ratio is about 10^{-4}), in order to obtain some tens of events per year the luminosity must be at the level of 10^{34} cm^{-2}s^{-1} corresponding to a collision bunch rate of 40 MHz. To allow the tuning of the machine and of the experiments, LHC is assumed to operate in the first three years at the reduced luminosity of 10^{33} cm^{-2}s^{-1}.

The beams will cross in four of the possible eight interaction points allowed by the eight fold symmetry of the LEP tunnel (see Figure 1-5). In these colliding points will be situated the four planned experiments: two general purpose experiment (ATLAS[1-12] and CMS[1-13]), an experiment optimized for B-physics studies (LHCb[1-14]) and an heavy ion collision experiment (ALICE[1-15]). Experimental caverns are currently under construction.

1.1.3 The experimental challenge of the two general purpose experiments

The two general purpose *pp* detector, ATLAS and CMS, are specifically designed with the aim to exploit the full physical research potentialities of the LHC. The experiments, as the accelerator, will face and overcome many technological challenges.

The pile-up is one of the most serious difficulties for the experimental operation at the LHC and has had a major impact on the detector design. Protons are grouped in bunches of about 10^{11} protons colliding at a given interaction point every 25 ns, a bunch spacing 1000 times shorter than LEP (21 μs) and more than a factor of 10 shorter than Tevatron RunII (396 ns) (see Table 1-1). Therefore the LHC detector must have a fast response time to avoid the integration of the detector signals over many bunch crossing. This requires sophisticated and highly performing read-out electronics.

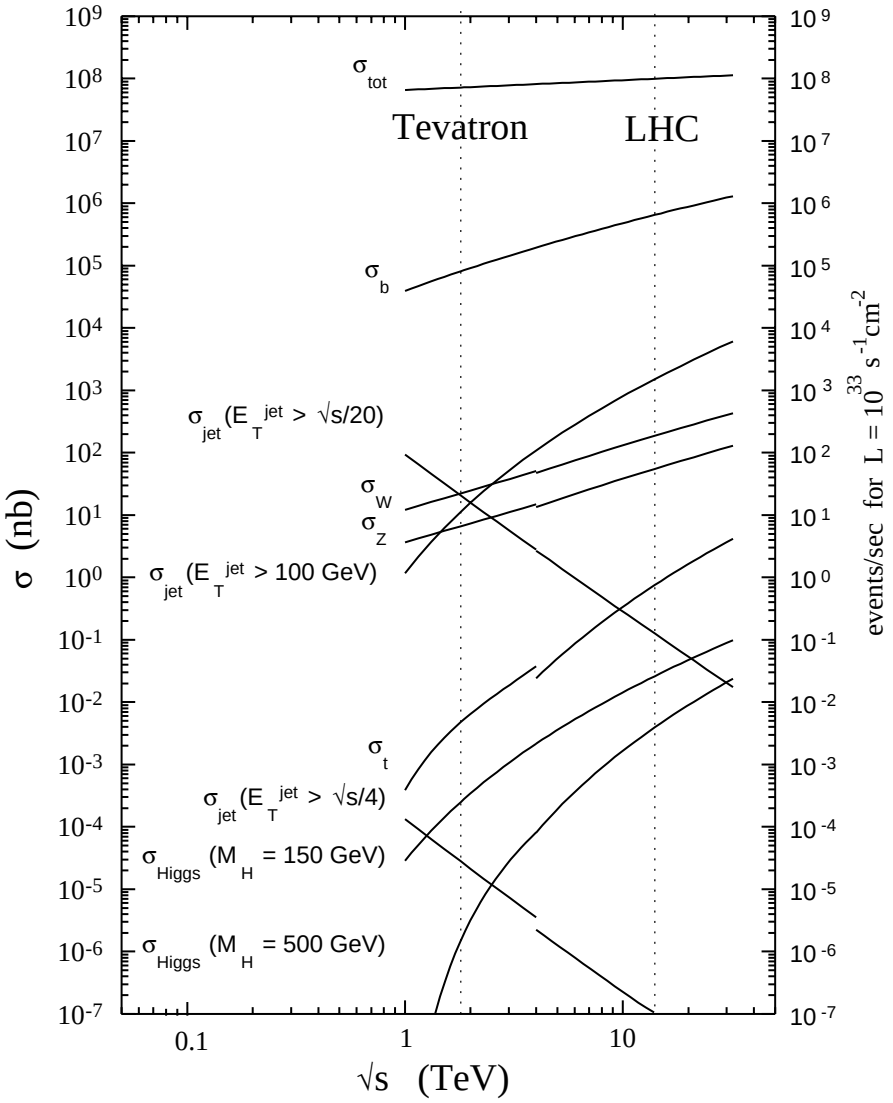

Figure 1-6 Cross section for hard scattering versus the centre of mass energy. The cross section values at LHC centre of mass energy are: σ_{tot} = 99.4 mb, σ_b = 0.633 mb, σ_t = 0.888 nb, $\sigma_H(m_H = 150\ \text{GeV})$ = 23.8 pb, $\sigma_H(m_H = 500\ \text{GeV})$ = 8.82 pb, $\sigma_{tot}(E_T > 100\ \text{GeV})$ = 1.57 mb, $\sigma_{tot}(E_T > \sqrt{s}/20)$ = 0.133 nb [1-11]. The right scale represent the expected event rate at a luminosity of $10^{33}\ \text{cm}^{-2}\text{s}^{-1}$.

Furthermore, at the design luminosity ($10^{34}\ \text{cm}^{-2}\text{s}^{-1}$), on average 25 soft interactions (minimum bias events) occur simultaneously at each crossing, overlapping interesting high-p_T events. This gives rise, every 25 ns, to about 1000 charged particles in the detector see Figure 1-7). So, in order to discriminate the different tracks and efficiently reconstruct the interesting events, the LHC detectors must have a fine read-out granularity, limited by the cost of a so huge number of electronic channel.

Another experimental challenge is due to the huge QCD background foreseen at LHC. As shown in Figure 1-6, the rate of high p_T events at a hadron collider is indeed dominated by QCD jet production from the fragmentation of quarks and gluons. Jet production is a strong process and therefore has a large cross-section. On the other hand, the most interesting physics processes at the LHC are rare processes, either because they involve the production of heavy particles, or because they have weak cross-sections. At $\sqrt{s}$ = 14 TeV the production cross-section for jets with $p_T > 100$ GeV is five orders of magnitude larger than the cross-section for a Higgs boson of mass 150 GeV. Therefore, there is no hope to detect a Higgs boson decaying into jets (unless it is produced in association with additional particles), since such final states are swamped by the much larger jet rate. So decays to leptons and photons have to be used instead. For the same reasons, and as another example, at LHC there is no hope, unlike at LEP, to observe the production of W pairs with both W bosons decaying into two jets. This process can only be detected if at least one of the two Ws decays into leptons. Since decays into leptons or photons have a smaller branching ratio than decays into quarks, the prize to pay to get rid of the QCD background is a smaller event rate.

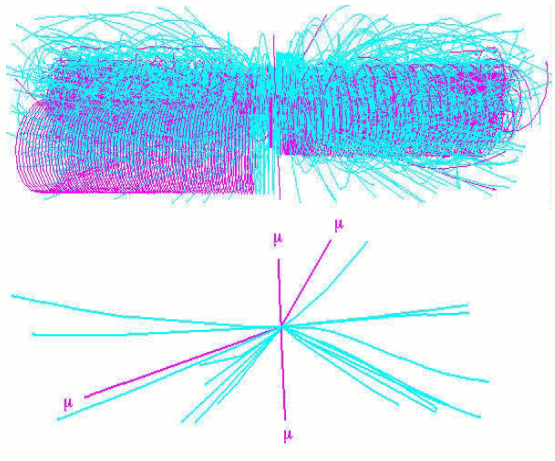

Figure 1-7 Simulation of a $H \rightarrow \mu\mu\mu\mu$ event in the CMS inner detector overlapped with 30 minimum bias events. In the top part all the charged particles are shown, in the bottom only tracks with p_T>2.

Finally the LHC detectors must be radiation resistant in order to survive the high flux of particles coming from the *pp* collision.

The main physics performance requirements of ATLAS and CMS can be summarised as follows:

- Leptons should be identified and measured over the p_T range from a few GeV up to a few TeV. The lower limits is set by the need to detect the soft leptons produced in the decays of *B*-hadrons, the upper one by the necessity to look for heavy particles decaying into leptons (e.g. additional gauge bosons *W*' and *Z*').
- Very good electromagnetic calorimetry for electron and photon identification and measurements; the granularity must be high enough to allow the reconstruction of the $H \rightarrow \gamma\gamma$ decay.
- Calorimetry should hermetically cover the full azimuthal angle and the pseudo-rapidity region $|\eta| \leq 5$ (i.e. down to 1° from the beam axis). This is required mainly for a reliable measurement of the event transverse energy, which is in turn needed to obtain information about weakly interacting particles as neutrinos, LSPs (Lightest Supersymmetric Particle), etc.. A large calorimetric coverage is also needed to detect the forward jets produced in association with a heavy Higgs boson.
- Excellent mass resolution (~ 1%) for particles of masses up to a few hundreds GeV decaying into photons, electrons or muons. This is needed, for instance, to extract a Higgs boson signal in the $\gamma\gamma$ decay mode over the irreducible background. Since the cross-sections for the irreducible backgrounds are often much larger than the signal cross-sections, the

reconstructed mass spectrum of the signal particle must be narrow enough to allow observation of a clear resonance above the continuum.

- Efficient vertex tagging, in order to provide secondary vertex information necessary for the reconstruction of *B*-hadron decay, for the tagging of *b*-jets (crucial in the identification of $H \rightarrow b\bar{b}$ decay, $H \rightarrow hh \rightarrow b\bar{b}b\bar{b}$ and in the selection of events related to the heavy-quark physics) and for the recognition of events containing the τ lepton ($H^{\pm} \rightarrow \tau\nu_{\tau}$, $A \rightarrow \tau\tau$). Silicon vertex detector with spatial resolutions of about 10 μm had been very important in the discovery of the top quark at Tevatron collider.
- Excellent electron/jet and photon/jet separation capabilities. Jets faking photons should be rejected by a factor of ~10^3, for a photon efficiency of ~80%, in order to observe a possible $H \rightarrow \gamma\gamma$ signal over the background. The ratio (e/jet) between the inclusive rate of electrons and the inclusive rate of jets with $p_T > 20$ GeV is e/jet $\approx 10^5$ at the LHC. Therefore jets faking electrons should be rejected by a factor of $\approx 10^6$ in order to observe a clean inclusive electron signal above the QCD background. This same ratio is e/jet $\approx 10^3$ at the Tevatron $p\bar{p}$ collider ($\sqrt{s}$ = 2 TeV), which indicates that the particle identification capabilities of ATLAS and CMS must be two orders of magnitude better than those of the Tevatron experiments CDF [1-8] and DØ [1-9].
- Triggering at LHC will be much more difficult than at present machines. The interaction rate of 10^9 events/s must be reduced to about 100 recorded events per second, which is an affordable data storage rate. Therefore, a very selective and at the same time efficient trigger, providing a rejection of ~ 10^7, is needed. Excellent particle identification capabilities are needed already at the trigger stage, in order to extract efficiently the interesting physics signals while reducing the large QCD backgrounds down to acceptable rates.

1.1.4 Overview of the four planned experiments

ATLAS and CMS are general purpose *pp* experiments with a broad physics programme, including both search of new particles and precision measurements. The general layouts are similar to that of past and present collider experiments. From the interaction point outwards they consist of:

- an inner detector immersed in a magnetic field, used to detect charged particles and to measure their momenta and charges, as well as to detect secondary vertices (e.g. from b-quark decay);
- an electromagnetic calorimeter, which measures the energy and position of electrons and photons and contributes to their identification;
- a hadron calorimeter, used to measure the energy and position of hadrons and jets, as well as the total event missing transverse energy;
- a muon spectrometer, used to identify muons and to measure they momentum together with the inner detector.

But, although the aims of the two experiments are the same, ATLAS and CMS used different and complementary experimental technics, mostly driven by the different magnetic system configuration

ATLAS[1-12] is a large (diameter 22 m, length 42 m) general-purpose detector with a 2 Tesla solenoidal field in the inner tracking volume and air-core toroids for the stand-alone measurement of muons. The inner detector consists of precision silicon detectors (pixels and strips) fol-

lowed by a transition-radiation tracker (TRT) with plastic-fibre and plastic-foil radiators and straw tube detectors. The pulse-height in the TRT can be used to improve the identification of electrons for momenta down to about 0.5 GeV. The electromagnetic calorimeters and the forward hadronic calorimeters exploit the LAr technology; the barrel hadronic calorimeters are based on scintillator read-out.

CMS[1-13] is smaller than ATLAS (diameter 15 m, length 30 m, including the very forward calorimeter), but two times heavier (14500 ton). The entire calorimeter system is inside a 4 T solenoidal field. The iron return yoke provides a second measurement of the momenta of the muons. The inner detector contains silicon pixels and strips and multi-strip gas chambers. The electromagnetic calorimeter is made of lead tungstate crystals. The hadronic calorimeter uses plastic scintillating tiles.

LHCb[1-14] is a collider experiment designed to study B physics at a constant luminosity of 1.5×10^{32} cm^{2}s^{-1}, concentrating on specific B decay modes. Characteristic signals include secondary and tertiary vertices and high-p_T leptons and hadrons from B decays. The LHCb apparatus is a forward spectrometer with a dipole magnet and planar geometry detectors as in fixed-target experiments. The apparatus extends from 10 mrad to 400 mrad in a single arm extending out from the interaction zone. The detector consists of a micro-vertex detector, a tracking system, aerogel and gas RICH counters for π/K separation, electromagnetic and hadronic calorimeters and a muon system. There will also be small-angle Roman-pot spectrometers in both arms.

LHC is also a heavy-ion collider able to provide Pb-Pb collision with a centre of mass energy of 1148 TeV (about 5.5 TeV at single nucleon level) and a luminosity of 10^{27} cm^{-2}s^{-1}. ALICE[1-15] is designed to investigate the quark-gluon plasma that is expected to be produced in these heavy-ion collisions. Most of the data will be taken at a constant event rate of about 8 kHz (luminosity 10^{27} cm^{2}s^{-1} for Pb-Pb collisions). Some characteristic signals include prompt photons, J/Ψ and upsilon production, and strange particle production. The detector is placed in a weak (0.2 Tesla) solenoidal field, with an inner tracker (silicon plus a cylindrical TPC to measure tracks down to 0.1 GeV) surrounded by time-of-flight counters for particle identification. A small-area, high-resolution electromagnetic calorimeter placed under the main apparatus will be used to measure photons. Another small-area detector, placed above the apparatus, will be used to identify high-momentum particles. One of the forward arms of the detector will be equipped for muon identification.

1.2 The ATLAS detector

1.2.1 The overall detector layout

The overall detector layout is shown in Figure 1-8. The detector geometry is based around the magnets' configuration. Most of the collider experiments, and also CMS, are build around a single solenoid magnet, which supplies to both central trackers and muon spectrometer the magnetic field needed for particles momentum measurement. Instead ATLAS is based on a different philosophy, which entail the use of four supeconducting magnets. A solenoid provides the magnetic field to the central trackers, while the muons are measured in three large superconducting air core toroids, that surround the calorimeter system.

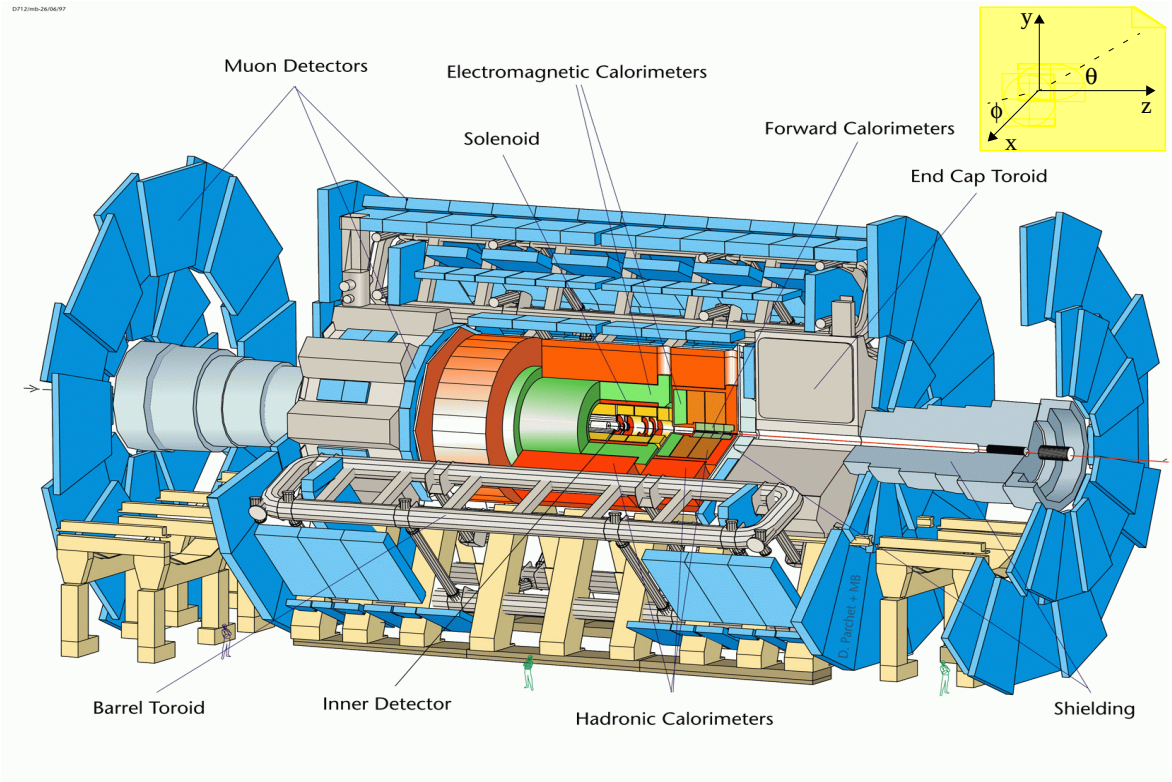

Figure 1-8 The ATLAS detector. The beam direction defines the *z*-axis, and the *xy* is the plane transverse to the beam direction.The positive *x*-axis is defined as pointing from the interaction point to the centre of the LHC ring, and the positive *y*-axis is pointing upwards. The azimuthal angle ϕ is measured around the beam axis, and the polar angle θ is the angle from the beam axis. The pseudorapidity is defined as $\eta = \ln \tan(\theta/2)$. The transverse momentum p_T and the transverse energy E_T are defined in the *xy* plane.

The use of two different magnet systems permits the placing of the solenoid inside the electromagnetic calorimeter, avoiding mechanical and technological constrains on the calorimeter detectors and reducing the lateral spread of the calorimeter showers. The solenoid have a radius of 1.22 cm, is 5.3 m long and provide a central field of 2 T to the inner detector (ID) contained inside. The drawback of this configuration is the increase of the amount of material in front of the calorimeter and therefore, to minimize this material, the solenoid share, in the barrel region, a common vacuum vessel with the electromagnetic calorimeter. In the lateral regions (called end-cap) there are two cryostats that enclose the electromagnetic end-cap calorimeters, the LAr hadronic calorimeters and two calorimeters, called very-forward, which cover the high pseudorapidity region ($3.2 < |\eta| < 4.9$). The three cryostats are surrounded in their whole length by a scintillating tiles hadron calorimeter. All the system is contained inside a cylindrical support that acts also as return joke for the solenoid's field lines.

The calorimeter system is in turn surrounded by the air core toroids of the muon spectrometer (see Figure 1-9) . The toroidal magnetic field configuration allows for a muon momentum measurement up to pseudorapidity $|\eta| = 2.7$, while the air core solution minimize the contribution of multiple scattering to the momentum resolution. The Barrel Toroid (BT) extends over a length of 25 m, with an inner bore of 9.4 m and an outer diameter of 20.1 m. As shown in Figure 1-10 the two End-Cap Toroids (ECTs) are inserted in the barrel at each end. They have a length of 5.0 m, an inner bore of 1.65 m and an outer diameter of 10.7 m. Each toroid consists of eight flat coils assembled radially and symmetrically around the beam axis. The ECT coils are

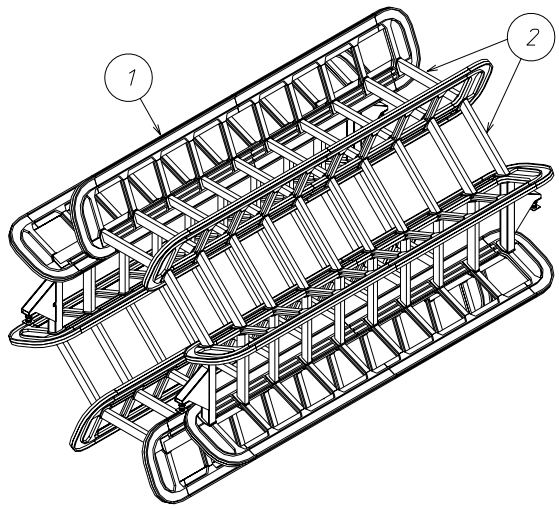

Figure 1-9 The barrel toroids: 1) the 8 coils, 2) the ribs.

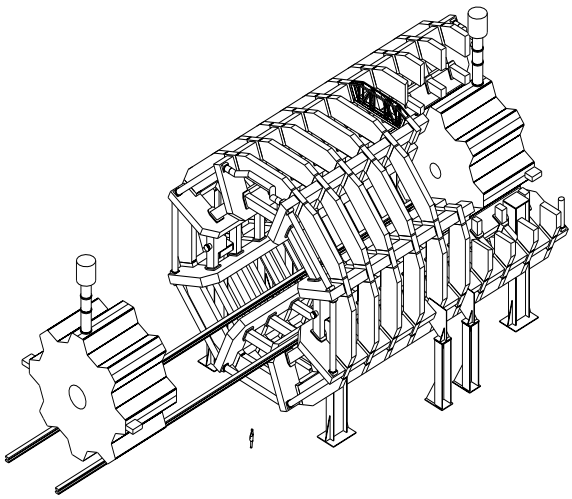

Figure 1-10 The end-cap toroids are inserted inside the barrel toroids.

rotated in azimuth by an angle of 22.5˚ with respect to the BT coils to provide for radial overlap, and to optimize the bending power in the transition region between the two toroids. The BT coils are contained in individual cryostats and are held rigidly together by means of eight rings of voussoirs and struts that contain the gravitational and magnetic forces. The eight coils of each ECT are assembled in a single large cryostat. The magnetic field provides for typical bending powers of 3 Tm in the barrel and 6 Tm in the end-cap regions. Owing to the finite number of coils, the field configuration is not perfectly toroidal and presents a regularly rippled profile. These effects are most visible in the transition region between the BT and the ECT, where there exist significant radial field components, as well as small regions with degraded momentum resolution.

Because of the smaller value of the magnetic field in air respect to iron, this configuration is more bulky: the diameter of the muon spectrometer is about 22 m and the total length is defined by the outer end-cap chambers mounted on the cavern hall, that are 46 m apart. This is considerably larger than current experiments at CERN or indeed anywhere in the world.

1.2.2 The inner detector

The ID, whose three dimensional view is shown in Figure 1-11, is 7 m long, has a radius of 115 cm and is contained inside the supeconducting solenoid magnet. It consist of three parts: a cylindrical section, covering the central pseudorapidity region ($|\eta| \leq 1$, ± 80 cm from the collision point) and two “end-cap” disk regions, covering the domain $1 \leq |\eta| \leq 2.5$.

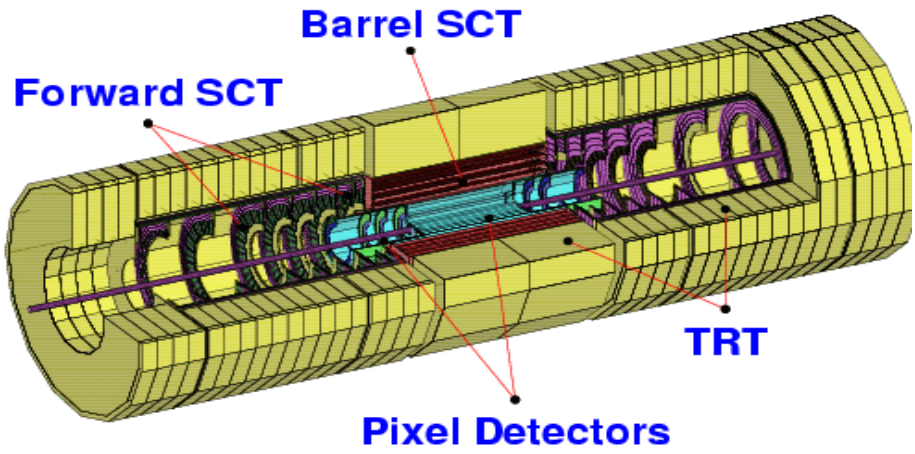

Figure 1-11 Three dimensional view of the ATLAS inner detector.

The momentum and vertex resolution needed in the high track density environment expected at LHC require high precision measurement to be made by fine granularity detector. This lead to the choice of the silicon semiconductor technology: pixel detectors close to the interaction region and silicon strip in the outer part. Due to the significant amount of material introduced by the silicon trackers and because of their cost, to obtain a large number of tracking points required for the pattern recognition, straw tube trackers are used at higher radii. This gives the possibility of continuous track following with much less material per channel and a lower cost. The combination of the two techniques offers very robust pattern recognition and high precision in both ϕ and z coordinates. Tracks emerging from the interaction point cross (see Figure 1-12) first of all the beam pipe (1 mm of beryllium at a radius of 2.5 cm), then at least three pixel layers, four double silicon strips planes (SCT: SemiConductor Tracker) and about 36 straw tubes of the Transition Radiation Tracker (TRT)

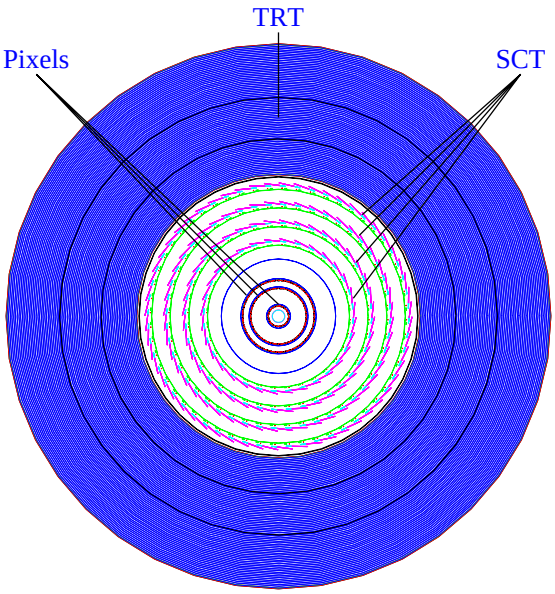

Figure 1-12 Transverse view of the Inner Detector.

Pixel Detector

The pixel detector provide three precision measurements over the full pseudorapidity range. Its major function is to provide secondary vertex information for efficiently b-jets tagging and full reconstruction of *B*-hadron decays. For this aim the first layer (called B-layer) is placed as close as possible to the beam line (about 4 cm from the interaction point). It was originally intended for the low luminosity phase only, now, however, it is considered very important to replace it for the high luminosity phase in order to maintain adequate tracking performance. The system consists of three barrels at average radii of 4, 10, 13 cm and five disk on each side between radii 11 and 20 cm along the beam line. In total will be installed about 140 million of pix-

els, with cell size of 50×300 μm each. The position resolution is about 10 μm in the Rϕ plane and about 50 μm in θ.

SemiConductor Tracker

The SCT detector is placed in the intermediate radial range and is planned to provide eight precision position measures per track. It consist of four concentric barrel layers positioned between radii 30 and 52 cm and 9 disks on either side covering the required pseudorapidity range. The basic element is composed by two detectors bonded together edge-to-edge. Each detector is 6.36×6.36 cm^2 silicon crystal corresponding to 768 strips of 80 μm. The resulting module is 12.82 cm long and 285 μm thick. Pairs of these modules are sandwiched together to form a barrel SCT component. The innermost detector is given a relative rotation (stereo angle) of 40 mrad with respect to the outermost, which lies with its long axis parallel to the beam pipe, to obtain the z measurement. The two end-cap is made of SCT modules arranged in rings, which are in turn arranged in wheels. The module are built in a similar fashion to the barrel ones, except for the fact that each modules has a trapezoidal shape. The SCT detector provides typically four space points per track with a resolution of ≈ 16 μm in $R\phi$ and ≈ 580 μm in the z direction. Tracks can be distinguished if separated by more than about 200 μm.

Transition Radiation Detector

The TRT is composed by straw tube detectors (drift chambers), surrounded by radiator material (foil and form) in order to enable electron identification from the detection of transition radiation X-rays created by the passage of the electrons in the radiator. The tubes are indeed filled with a mixture containing Xenon to detect these photons. The straw are 65 μm thick Kapton tubes and have an internal diameter of only 4 mm to limit the occupancy. The detector consist of a central section which has a barrel geometry for $|\eta| \leq 0.8$ and two end-caps sections consisting of multi-plane wheels at higher $|\eta|$. The barrel region has a total of 73 layers of axial straw tubes and extends from an inner radius of 56 cm to an outer radius of 107 cm. The two end-caps have 18 wheels of radially oriented straw tubes, the first 14 nearest the interaction point cover a radius of 64 to 102 cm and the last four wheels extend down to a radius of 48 cm to provide coverage of the full pseudorapidity range. The TRT detector provides typically 36 measurements per track for nearly all pseudorapidity with a spatial resolution of 170 μm. This contribute significantly to the momentum measurement, since the lower precision per point compared to the silicon is compensated by the large number of measurements and the higher average radius.

The performance of the inner detector is well characterized by the inverse momentum resolution for muons and transverse impact parameter[1] resolution of muon tracks from secondary vertices:

$$\sigma\left(\frac{1}{p_T}\right) \approx 3.6\times10^{-4} \oplus \frac{1.3\times10^{-2}}{p_T \cdot \sqrt{\sin\theta}} \quad (\mathrm{GeV}^{-1}) \qquad \sigma(d_0) \approx 11 \oplus \frac{73}{p_T \cdot \sqrt{\sin\theta}} \quad (\mu\mathrm{m}^{-1})$$

As described in Chapter 7, this performance allows to identify high-p_T jets from b-quark fragmentation; a capability which is fundamental for the identification and analysis of many physics channels in ATLAS.

1. The transverse impact parameter (d_0) is defined as the transverse distance to the beam axis at the point of closest approach.

1.2.3 Calorimetry

Figure 1-13 gives an overview of the ATLAS calorimeter. It consists of an internal barrel cylinder and two end-caps, which are surrounded over their full length by the hadronic scintillator tile calorimeter (HC). The HC, covering the pseudorapidity region $|\eta| < 1.7$, is divided in three sections: a barrel and two extended barrel. The barrel and the end-cap electromagnetic calorimeter (EM) cover the region $|\eta| < 3.2$. The latters share the same cryostat with the hadronic end-cap calorimeters (HEC, $1.5 < |\eta| < 3.2$) and the forward calorimeters (FCAL) covering the high rapidity region ($3.1 < |\eta| < 4.9$).

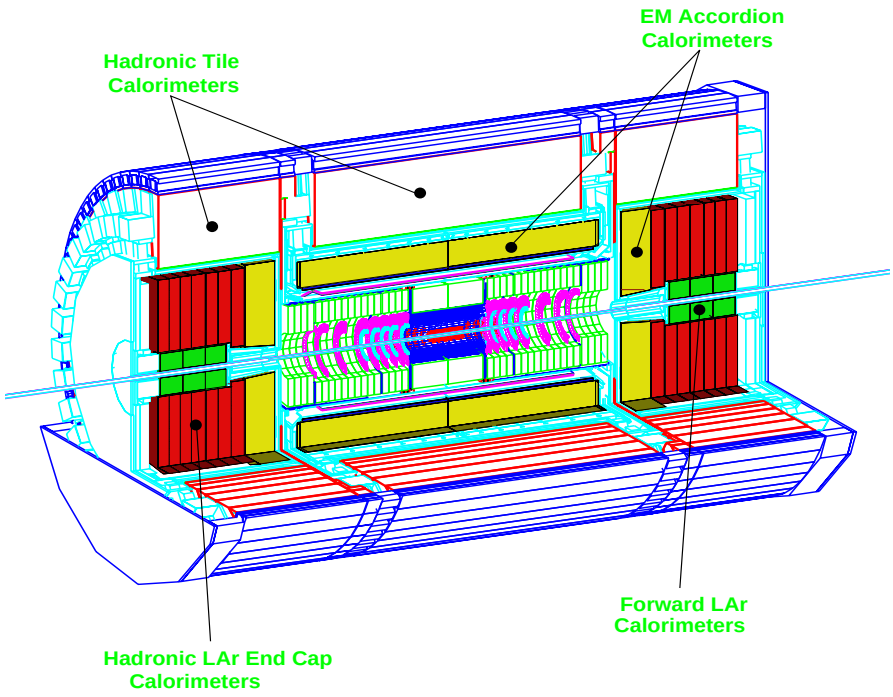

Figure 1-13 Three-dimensional cutaway view of the ATLAS calorimeters.

Electromagnetic calorimeters

Both barrel and end-cap calorimeters are sampling calorimeter based on the LAr technology because of its intrinsic radiation resistant properties. The lead absorber are arranged in an accordion shape, the waves of the plates and read-out being perpendicular to the incident particles. This geometry provides complete ϕ symmetry without azimuthal cracks and homogenous detector response. The system has been studied to let electron and photon identification and reconstruction in a wide energy range: from ~ 1 GeV (electrons from semileptonic *b*-quark decay) up to some TeV (for $Z' \rightarrow e^+e^-$, $W' \rightarrow e\nu$ decay). The end-cap modules assure the coverage up to $|\eta| = 3.2$ needed by the search of rare decay as $H \rightarrow \gamma\gamma$ or $H \rightarrow e^+e^-e^+e^-$.

In the region $|\eta| < 2.5$, dedicated to precision physics, the EM calorimeter is longitudinally segmented in three sections (see Figure 1-14):

- The inner section is finely segmented in η by narrow read-out strips (~ 4 mm) providing a granularity of $|\Delta\eta \times \Delta\phi| \approx 0.003 \times 0.1$. This ensure an excellent η position resolution, which enhances particle identification, γ/π^0 and e/π separation and easies the observability of the $H \rightarrow \gamma\gamma$ decay. The section thickness is about 6 X_0.
- The second and thickest sampling is composed by cells with granularity of 0.025×0.025 corresponding to about 4×4 cm^2 at η=0. The thickness is about 18 X_0.
- The outer section is reached only by the most energetic electromagnetic showers and so it has a coarser η segmentation (0.05×0.025). The total thickness is $X_0 > 24$ in the barrel and $X_0 > 26$ in the end-cap.

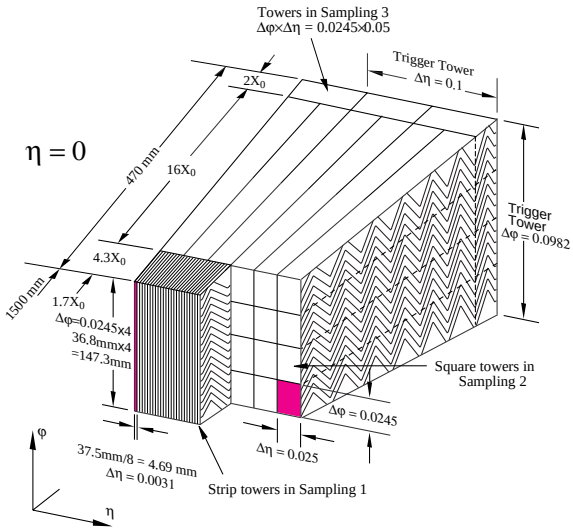

Figure 1-14 Lateral and longitudinal segmentation of the electromagnetic sampling calorimeter.

Outside the precision physics region ($2.5 < |\eta| < 3.2$) the calorimeter is segmented in only two longitudinal sections and has a coarser lateral granularity 0.1×0.1. In the region, where the amount of material in front of the calorimeter exceed 2 X_0, a presampler is used to correct for the energy lost by electron and photons in the inner detector. The region $1.37 < |\eta| < 1.52$ is not used for precision physics measurements involving photons because of the large amount of material situated in front of the EM calorimeter. The energy resolution measured on a full scale module prototype is[1] [1-16]

$$\frac{\Delta E}{E} \approx \frac{10\%}{\sqrt{E}} \oplus \frac{300\text{MeV}}{E} \oplus 0.4\%$$

1. The percent resolution of a calorimeter is in general expressed as the sum of three terms: $\frac{\Delta E}{E} = \frac{a}{\sqrt{E}} \oplus \frac{b}{E} \oplus c$; a is associated to the shower formation (stocastic term), b is the noise term and c depend on the particular detector (constant term)

Hadronic calorimeters

The hadronic calorimeters cover the range $|\eta| < 4.9$ using different techniques best suited for the widely varying requirements and radiation environment over the large η-range:

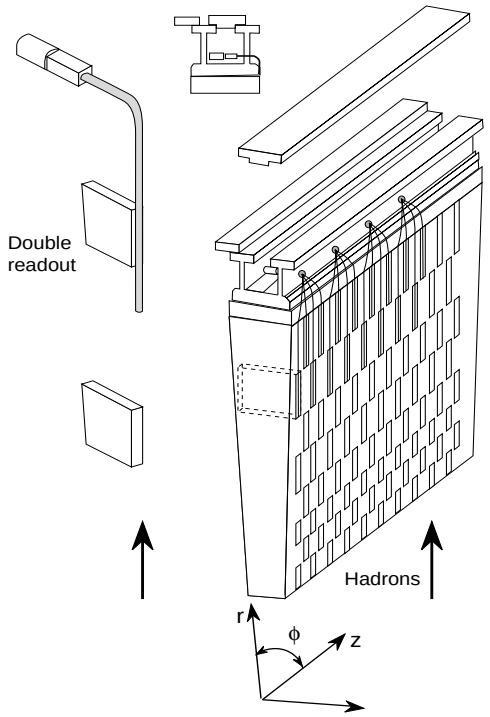

Figure 1-15 Scintillating hadron calorimeter module.

- The barrel and extended barrel calorimeters, covering the range $|\eta| < 1.7$, are sampling calorimeter using iron as absorber and scintillating tiles as active material (Figure 1-15). The barrel cylinder covers the region $|\eta| < 1.0$, while the extended barrel the region $0.8 < |\eta| < 1.7$. The tiles, 3 mm thick, are placed radially and staggered in depth. Two sides of the scintillating tiles are read out by wave-length shifting fibres connected to photomultipliers. Radially the tile calorimeter extends from an inner radius of 2.28 m to an outer radius of 4.25 m. It is longitudinally segmented in three layers, approximately 1.4, 4.0 and 1.8 interaction lengths thick at $\eta = 0$. Azimuthally, the barrel and extended barrels are divided into 64 modules. In η, the read-out cells, built by grouping fibres into photomultipliers, are pseudo-projective towards the interaction region. The resulting granularity is $|\Delta\eta \times \Delta\phi| \approx 0.1 \times 0.1$ (0.2×0.2 in the last layer). The total number of channels is about 10 000. The calorimeter is placed behind the EM calorimeter ($\approx$ 1.2 interaction lengths[1]) and the solenoid coil. The total thickness at the outer edge of the tile-instrumented region is 9.2 λ at $\eta = 0$. The thickness of the calorimeter in the gap is improved by the ITC, which has the same segmentation as the rest of the tile calorimeter.
- The hadronic end-cap calorimeter (HEC) is based on LAr technique that offers intrinsic resistant properties. It extends from $|\eta| \sim 1.5$ up to $|\eta| = 3.2$ and is integrated in the same cryostat that contains the EM and the FCAL. Each HEC consists of two independent wheels, of outer radius 2.03 m. The upstream wheel is built out of 25 mm copper plates, while the cheaper other one, farther from the interaction point, uses 50 mm plates. In both wheels, the 8.5 mm gap between consecutive copper plates is equipped with three parallel electrodes, splitting the gap into four drift spaces of about 1.8 mm. The read-out electrode is the central one, which is a three layer printed circuit, as in the EM calorimeter. To minimise the dip in the material density at the transition between the end-cap and the forward calorimeter (around $|\eta| = 3.1$), the end-cap EM calorimeter reaches $|\eta| = 3.2$, thereby overlapping the forward calorimeter.
- The FCAL is a particularly challenging detector owing to the high level of radiation it has to cope with. It is integrated into the end-cap cryostat, with a front face at about 4.7 m from the interaction point. Compared to layouts with a forward calorimeter situated at much larger distances from the interaction point, the survival of such a calorimeter in

1. The interaction length λ_{int} is defined as the mean free path of a proton between two elastic collisions and characterizes the longitudinal scale of an hadronic shower.

terms of radiation resistance is clearly more difficult. On the other hand, the integrated FCAL provides clear benefits in terms of uniformity of the calorimetric coverage as well as reduced radiation background levels in the muon spectrometer. To minimize the reflected neutron flux from the calorimeter face to the inner detector (highest at high η), the FCAL is recessed by about 1.2 m with respect to the EM Calorimeter front face. This severely limits longitudinal space for installing about 9.5 active interaction lengths, and therefore calls for a high-density design, which also avoids energy leakage from the FCAL to its neighbours. The FCAL consists of three sections: the first one is made of copper, while the other two are made out of tungsten. In each section the calorimeter consists of a metal matrix with regularly spaced longitudinal channels filled with concentric rods and tubes. The rods are at positive high voltage while the tubes and matrix are grounded. The LAr in the gap between is the sensitive medium. This geometry allows for an excellent control of the gaps which are as small as 250 mm in the first section.

The planned resolution, verified by the prototypes, are:

$$\frac{\Delta E}{E} \approx \frac{50\%}{\sqrt{E}} \oplus 3\% \qquad \text{for } |\eta| < 3$$

$$\frac{\Delta E}{E} \approx \frac{100\%}{\sqrt{E}} \oplus 10\% \qquad \text{for } |\eta| > 3$$

1.2.4 Muon spectrometer

The issue with the strongest impact on the design of the ATLAS muon spectrometer is the requirement fro stand-alone measurement capability. This safeguards against any unanticipated backgrounds and ensures a good discovery potential in the presence of unexpected topologies due to unknown physics at the TeV scale. To accomplish a good moment measurement even at low momenta this requirement leads to the choice of an air-core magnet system, which however part puts high demands on the muon chamber system due to the unavoidable field inhomogenities and the relatively weak fields in such system. ATLAS has chosen a toroidal magnetic field configuration to allow for a muon momentum measurement up to pseudorapidity $|\eta| = 2.7$.

The overall layout of the muon chambers in the ATLAS detector is shown in Figure 1-16 and Figure 1-17, which indicates the different regions in which the four chamber technologies are employed. The chambers are arranged such that particles from the interaction point traverse three stations of chambers. The positions of these stations are optimized for good hermeticity and optimum momentum resolution. In the barrel, particles are measured near the inner and outer field boundaries, and inside the field volume, in order to determine the momentum from the sagitta of the trajectory. In the end-cap regions, for $|\eta| > 1.4$, the magnet cryostats do not allow the positioning of chambers inside the field volume. Instead, the chambers are arranged to determine the momentum with the best possible resolution from a point-angle measurement.

As shown in Figure 1-16, the barrel chambers form three cylinders concentric with the beam axis, at radii of about 5, 7.5, and 10 m. They cover the pseudorapidity range $|\eta| < 1$. The end-cap chambers cover the range $1 < |\eta| < 2.7$ and are arranged in four disks at distances of 7, 10, 14, and 21 m from the interaction point, concentric with the beam axis. All chambers combined provide almost complete coverage of the pseudorapidity range $1 < |\eta| < 2.7$. For the precision measurement of muon tracks in the principal bending direction of the magnetic fields, Monitored Drift Tube (MDT) chambers are used except in the innermost ring of the inner sta-

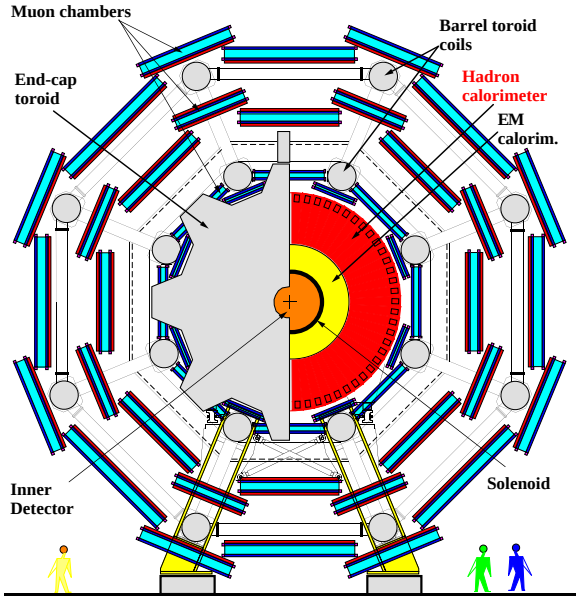

Figure 1-16 Transverse view of the ATLAS detector showing the location of the three barrel muon stations.

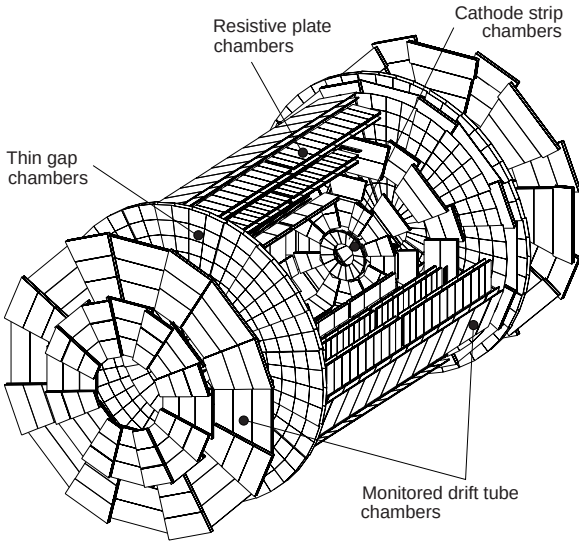

Figure 1-17 Three dimensional view of the muon spectrometer instrumentation indicating the areas covered by the four different chamber technologies.

tion of the end-caps, where particle fluxes are highest. In this region, covering the pseudorapidity range $2 < |\eta| < 2.7$, Cathode Strip Chambers (CSCs) are employed. The trigger function in the barrel is provided by three stations of Resistive Plate Chambers (RPCs). They are located on both sides of the middle MDT station, and either directly above or directly below the outer MDT station. In the end-caps, the trigger is provided by three stations of Thin Gap Chambers (TGCs) located near the middle MDT station.

The muon spectrometer is designed for a momentum resolution of about 2% for p_T = 100 GeV and 10% for p_T = 1 TeV; at smaller momenta, the resolution is limited to a few per cent by multiple scattering in the magnet and detector structures, and by energy loss fluctuations in the calorimeters. To achieve this resolution by a three-point measurement, with the size and bending power of the ATLAS toroids, each point must be measured with an accuracy better than 50 mm. The precision measurement of the muon tracks is made in the *rz* projection, in a direction parallel to the bending direction of the magnetic field; the axial coordinate (*z*) is measured in the barrel and the radial coordinate (*r*) in the transition and end-cap regions. Monitored Drift Tube (MDT) chambers are used for this purpose over most of the solid angle covered by the spectrometer. They provide for a single-wire resolution of ~80 µm when operated at high gas pressure (3 bar), and for robust and reliable operation due to the mechanical isolation of neighboured wires. In the first station in the end-cap region and for pseudorapidities $|\eta| > 2$, Cathode Strip Chambers (CSCs) are used to provide a finer granularity which is required to cope with the demanding rate and background conditions in this region of the apparatus.

1.3 The physics potential of the ATLAS experiment

The high luminosity provided by LHC will lead to large production rates for many relevant processes. As shown is Table 1-2, even in the low luminosity phase (10^{33} cm^{-2}s^{-1}) LHC will produce every second about one $t\bar{t}$ pair, 5 *Z* bosons decaying in lepton pairs, 50 *W* bosons, 100 QCD

jets with $p_T > 200$ GeV and 500000 $b\bar{b}$ pairs. These two latters processes are expected to be a huge background to many interesting physics channels. Furthermore, one SM Higgs of 115 GeV, corresponding to the events observed at LEP (Section 1.1.1), will be produced every 40 seconds. Integrated over one year of data taking at low luminosity, these rates lead up to samples of millions of events in almost all channels. The LHC can therefore be considered as a factory of a large number of particles: *W* and *Z* bosons, *t* e *b* quarks, and possibly also Higgs boson(s) and supersymmetric particles.

Table 1-2 Cross-section, approximated expected number of events per seconds and the sample after one year of data taking (30 fb^{-1}) for some representative channels at low luminosity (10^{33} $cm^{-2}s^{-1}$)

Process	σ (pb)	Events/second	Events/year
$b\bar{b}$	5×10^8	5×10^5	10^{12}
Inclusive jets; p_T>200 GeV	10^5	10^2	10^9
$W \rightarrow e\nu$	1.5×10^4	5	10^8
$W \rightarrow e^+e^-$	1.5×10^3	1.5	10^7
$t\bar{t}$	800	0.8	10^7
$\tilde{g}\tilde{g}$ ($m_{\tilde{g}}$ = 1 TeV)	1	10^{-3}	10^4
H (m_H = 115 GeV)	22.7	2.3×10^{-2}	10^5

Even if the search for the Higgs boson was the first benchmark for detector optimization, the design of the ATLAS experiment has been optimized to cover a large spectrum of possible physics signatures, accessible at the high luminosity and centre-of-mass energy of the LHC. ATLAS will be able to search for other phenomena possibly related to the symmetry breaking, such as particles predicted by supersymmetry or technicolour theories, for new gauge bosons, for compositeness of the fundamental fermions. The ATLAS detector will also allow many precision measurements in a large number of physics channels.

1.3.1 Search for the Higgs boson

The Higgs boson total production cross-section is larger than 100 fb even for m_H as high as 1 TeV. Therefore more than 1000 events are expected to be produced in one year of running at low luminosity and more than 10000 events in one year of running at high luminosity for $m_H \sim 1$ TeV. The Higgs boson width (Γ_H) increases with the third power of the mass. For masses of about 100 GeV the Higgs width is of the order of a few MeV, increasing to about 100 GeV for masses of 60 GeV. The branching ratios of the Higgs as a function of its mass is shown in Figure 1-18. The Higgs boson couplings to fermions are proportional to the fermion masses: for $m_H < 120$ GeV the $H \rightarrow b\bar{b}$ decay mode dominates, since b-quarks are the

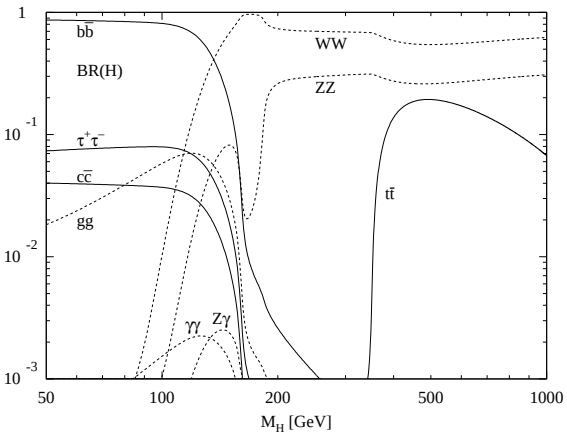

Figure 1-18 Branching ratios of the SM Higgs boson as a function of its mass [1-17]

most massive fermions kinematically accessible in this region. For $m_H \sim 2m_Z$ also the vector gauge bosons become kinematically accessible and consequently the $H \rightarrow WW^{(*)}$ and $H \rightarrow ZZ^{(*)}$ decays dominate the high m_H region. Therefore the SM Higgs boson will be searched for in various decay channels, the choice of which is given by the signal rates and the signal to background ratios in the various mass regions:

- High mass region: $m_H > 2m_Z$. Up to about 600 GeV, the $H \rightarrow ZZ \rightarrow 4\,l$ decay mode is the most reliable channel for the discovery of a SM Higgs boson. The expected background, which is dominated by the continuum production of Z boson pairs, is smaller than the signal. A very massive Higgs boson ($m_H > 600$ GeV) can be instead discovered in the channels $H \rightarrow ZZ \rightarrow ll\nu\nu$ and $H \rightarrow WW \rightarrow l\nu jj$, which offers branching ratios respectively 6 times and 150 times larger than that of the gold plated Higgs decay in four leptons.
- Intermediate mass region: 130 GeV $< m_H < 2m_Z$. The most promising channels for the experimental searches are $H \rightarrow ZZ^{(*)} \rightarrow 4l$ and $H \rightarrow WW^{(*)} \rightarrow l\nu l\nu$. For Higgs boson masses close to 170 GeV, the signal significance in the four lepton channel is reduced, due to the suppression of the $ZZ^{(*)}$ branching ratio as the WW decay mode opens up. This region is covered by the $H \rightarrow WW^{(*)} \rightarrow l\nu l\nu$ decays. In this channel however, it is not possible to reconstruct the Higgs-boson mass peak. Instead, an excess of events has to be used to identify the presence of a Higgs-boson signal. The signal significance in this channel depends critically on the absolute knowledge of the background. If a relative uncertainty on the background of ± 5% is assumed, the signal significance exceeds 5σ in the mass range between ~ 150 GeV and 190 GeV.
- Low mass region: $m_H < 130$ GeV. This is the mass region indicated by the LEP Higgs observation ($m_H \sim 115$ GeV) [1-7]. The decay modes containing vector gauge boson are closed and the search is based on the challenging channels $H \rightarrow \gamma\gamma$ and $H \rightarrow b\bar{b}$.

  The $H \rightarrow \gamma\gamma$ channel has a branching ratio at the level of 10^{-3} and therefore a small cross-section (~20 fb). The signal signature is simple but the background is very hard to fight. The $\gamma\gamma$ production, for the continuum irreducible background, has a cross-section 60 times larger than the signal. The reducible background are is to γj and jj production, where one or both jets fake a photon. Although the probability for a jet to fragment into a single isolated π^0 is small, the cross-section for γj and jj production is about 10^6 times larger than the cross-section for the $\gamma\gamma$ continuum. The jet rejection factor (about 1000) achieved by the electromagnetic calorimeter and its excellent energy and angular resolutions permits the extraction of the narrow resonant peak above the overwhelming continuum background with a signal to background ratio of about 10^{-2}.

  The $H \rightarrow b\bar{b}$ has a branching ratio close to 100% in most of this mass region, and therefore inclusive Higgs production followed by $b\bar{b}$ pair decay has a large cross-section (~ 20 pb). However, since the signal-to-background ratio for the inclusive production is smaller than 10^{-5}, it will be impossible to observe this channel above the QCD background and even to select it at the trigger level. On the other hand, the associated production $t\bar{t}H$, WH, ZH, with $H \rightarrow b\bar{b}$ and with an additional lepton coming from the decay of the accompanying particles, has a much smaller cross section (< 1 pb) but gives rise to final states which can be extracted from the background. The discovery of this decay mode depend considerably on the impact parameter measurement performance and on the b-tagging capability of the inner detector.

The overall sensitivity for the SM Higgs boson discovery is shown in Figure 1-19 for two different values of integrated luminosity. A Standard Model Higgs boson can be discovered by the ATLAS experiment over the full mass range from the LEP2 lower limit up to the TeV range with a high significance. A 5σ-discovery can already be achieved over the full mass range after a few

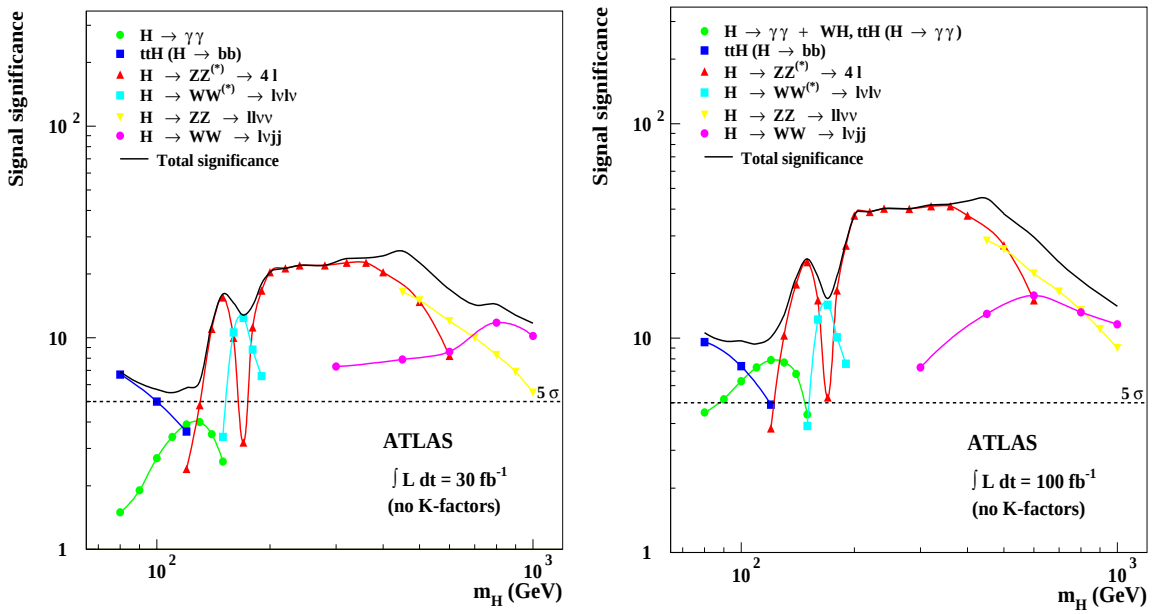

Figure 1-19 ATLAS sensitivity for the discovery of a SM Higgs boson. The statistical significances are plotted for individual channels, as well as for the combination of all channels, assuming two different values of integrated luminosity[1-18]: 30 fb^{-1} in the left plot (corresponding to 3 years of data taking at low luminosity) and 100 fb^{-1} in the right one (corresponding to 3 years of data taking at low luminosity plus 1 year at high luminosity.).

years of running at low luminosity. Over a large fraction of the mass range the discovery of a Standard Model Higgs boson will be possible in two or more independent channels.

Also important SM Higgs boson parameters like the mass and the width as well as production rates can be measured with a reasonable precision. With an integrated luminosity of 300 fb^{-1} ATLAS would measure the Higgs-boson mass with a precision of 0.1% over the mass range 80 - 400 GeV, the Higgs-boson width with a precision of 6% over the mass range 300 - 700 GeV, the Higgs-boson production rate with a precision of ~ 10%, and several important couplings and branching ratios with a precision of the order of 25%.

Also in the MSSM Higgs sector, the ATLAS experiment has a large potential of investigation. If the SUSY mass scale is large and supersymmetric particles do not appear in the Higgs decay products, the full parameter space in the conventional (m_A, $tan\beta$) plane can be covered. The 5σ discovery contour curves are shown in Figure 1-20 for individual channels and for different integrated luminosities. In addition to the channels discussed for the Standard Model case, the MSSM Higgs search relies heavily on the $H/A \rightarrow \tau\tau$ channel, on the $t\bar{t}h$ with $h \rightarrow b\bar{b}$ and on the direct and associated $H \rightarrow \gamma\gamma$ channels. Over a large fraction of the parameter space more than one Higgs boson and/or more than one decay mode would be accessible. For almost all cases, the experiment would be able to distinguish between the Standard Model and the MSSM models.

1.3.2 Search for supersymmetry

If Supersymmetry (SUSY) exists at the electro-weak scale, its discovery at the LHC should be straightforward. The SUSY cross section is dominated by gluinos and squarks, which are strongly produced with cross sections comparable to the Standard Model backgrounds at the

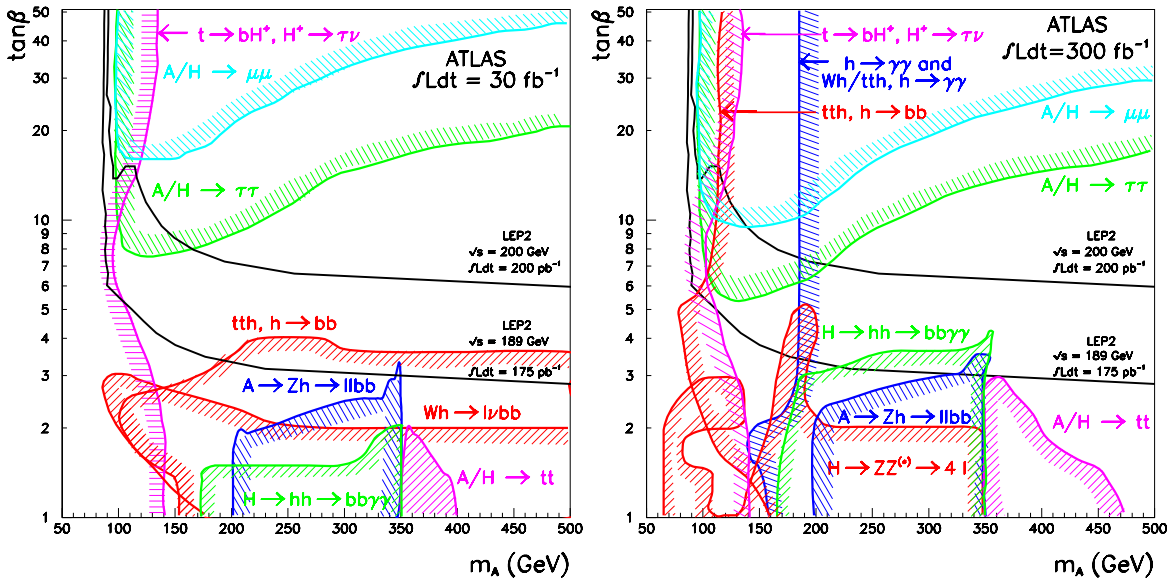

Figure 1-20 ATLAS sensitivity for the discovery of MSSM Higgs bosons (in case of minimal mixing) [1-18]. The 5σ discovery contour curves are shown in the (m_A,tanβ) plane for individual channels and for integrated luminosity of 30 fb^{-1} (left) and 300 fb^{-1} (right). Also included are the present LEP2 limit (for an integrated luminosity of 175 pb^{-1} per experiment) and the expected ultimate LEP2 limit (for an integrated luminosity of 200 pb^{-1} per experiments at a centre of mass energy of 200 GeV).

same Q^2. For squark or gluino masses of 1 TeV, the pair production cross-section is about 1 pb at LHC (see Table 1-2), therefore about 10000 events should be produced in one year of data taking at low luminosity. The decay modes of SUSY particles depend on the model parameters, namely particle masses, couplings, etc..These cascade decays produce final states containing leptons, many jets, W and Z bosons, sometimes b-quarks and even top-quarks, plus missing transverse energy from the escaping LSP (Lightest Supersymmetric Particle). Furthermore, the decay products have usually large p_T since they originate from heavy particles. Such spectacular signatures are therefore quite easy to extract from the Standard Model background. If SUSY will be discovered, many precise measurements of the SUSY particle masses should be possible at the LHC [1-18]. This should allow the parameters of the underlying theory to be constrained with precision at the level of the percent for minimal models such as minimal Supergravity [1-18].

1.3.3 Other physics beyond the Standard Model

In the absence of a scalar Higgs boson, the principal probe for the mechanism of electroweak symmetry breaking will be gauge boson scattering at high energies. It has been shown that ATLAS will be sensitive to the presence of resonances, such as in the *WZ* system, up to masses around 1.5 TeV[1-18].

As already remarked, the ATLAS detector will be able to test various others theoretical scenario in addition to supersymmetry[1-18]. ATLAS will have the potential to discovery:

- excited quarks in the photon plus jet channel up to masses of 6 TeV;
- leptoquark up to a mass scale of about 1.5 TeV;
- new vector bosons (*W'* and *Z'*) with masses lower than 5÷ 6 TeV;

- technicolour resonances in their decay to a pair of gauge bosons or to a tecni-pion and a gauge boson (the mass limit is about 1 TeV);

1.3.4 Precision measurements of the standard model parameters

LHC will be not only a machine dedicated to the discovery of new physics but it will allow to carry out precision measurements in many different sectors: vector gauge bosons, heavy quarks physics, Triple-Gauge coupling, etc. As a consequence of the high statistic (see Table 1-2), significant improvements on the future TeVatron and LEP results are expected even after only few years of operation. Indeed the statistical error, which scale as $1/\sqrt{N}$ (where N is the number of selected events), will be negligible in most measurements. The uncertainty will instead be dominated by the component of the systematic error which is independent on the number of collected events. However large statistic will allow hard cuts to be applied in order to select clean and well-understood events. Furthermore, high-statistics "control-sample" will be available to study the detector response and the physics p_{T} in great detail. The main sources of uncertainty which will affect precision measurements will be the lepton and momentum scale, related to the calibration and intercalibration of the different sub-detectors, the jet energy scale and the knowledge of the absolute luminosity, which will contribute to the uncertainty on all cross-section measurements.

Precision measurements in the B hadrons sector

During the initial phase at low luminosity, about one collision in every hundred will produce a b-quark pair. The huge number of b-quarks produced will allow a wide range of precision measurements in the B-physics sector:

- precise measurements of the periods of flavour oscillations in B_s^0 and B_d^0 mesons; these measure will overconstrain the CKM matrix, possibly giving indirect evidence for new physics;
- precise measurements of CP violation in *B*-meson decays, which in the Standard Model is due to a single phase in the CKM matrix;
- searches for and measurements of very rare decays which are strongly suppressed in the SM and where significant enhancements could provide indirect evidence for new physics.

Many of the ATLAS measurements will be more precise than those from experiments at lower-energy machines and some will be competitive with the foreseen results of the LHCb experiment. All these measurements rely on the b-tagging capability of the ATLAS detector, on the precise secondary vertex determination and on full reconstruction of final states with relatively low-p_{T} particles.

Precision measurements in the top quark sector

The top quark is the only known fermions with a mass (m_t) on the electroweak scale and therefore precision measurements in the top sector are important to get more clues on the origin of the fermion mass hierarchy and to test the electroweak symmetry breaking mechanism. Furthermore, because of the large value of m_t, it plays a special role in radiative corrections: within the SM, an accurate measurement of the top quark mass help constrain the mass of the SM Higgs boson. At LHC, top quark measurements will benefit from the large $t\bar{t}$ samples, so that not only the mass and the production cross-section, but also the branching ratios, coupling and rare decays can be studied in detail. It should be noted that $t\bar{t}$ production is expected to be the main background to many new physics process. The $t\bar{t}$ production cross section at LHC is expected to be about 2 order of magnitude larger than TeVatron. Considering also the higher lumi-

nosity, LHC will be able to collect, in the initial low luminosity phase only, an event sample at least 1000 times greater than the one expected at the end of the TeVatron program. The m_t will be measured by the analysis of the semileptonic decay of the $t\bar{t}$ sample. The high p_T lepton and the large E_T^{miss} produced by the first top decay will be used to tag the event, while the m_t will be determined from the hadronic decay of the second top, as the invariant mass of the three jets originating from the same top ($m_t = m_{jjb}$). Provided the energy scales for jets and *b*-jets can be understood at the 1% level, a total error below 2 GeV should be obtainable; a precision of 3 GeV would be achieved by the TeVatron in the future before LHC start-up.

An important tool for selecting clean top quark samples, particularly in the single lepton plus jets mode, is the ability to identify b-quarks. With a tagging efficiency of 60% for *b*-jets, a rejection of at least 100 can be achieved against prompt jets at low luminosity. At high luminosity, a rejection factor of around 100 is obtained with a reduced b-tagging efficiency of 50%.

Measurements of the W mass

At the time of the LHC start-up, the *W* mass will be known with a precision of ~ 30 MeV from measurements at the Tevatron and LEP2. The motivation to improve this result is mainly that precise measurements of the W mass, of the top mass and of the Higgs mass will provide stringent tests of the consistency of the underlying theory. With a top mass measured with an accuracy of ~ 1.5 GeV (see above), the *W* mass should be known with a matching precision of 15 MeV, in order not to become the dominant source of uncertainty in the test of the radiative corrections. Such precision should constrain the Higgs mass to better than 30%. At hadron colliders, the W mass is obtained from the distribution of the W transverse mass, that is the invariant mass of the W decay products evaluated in the plane transverse to the beam. Because at LHC about 10^8 $W \rightarrow l\nu$ decays will be reconstructed per year at low luminosity (Table 1-2), the statistical error on the W mass measurements is expected to be small. Like at the Tevatron, also at the LHC the dominant uncertainty will originate from the calibration of the absolute lepton energy scale. For the W mass to be measured to better than 20 MeV, the lepton scale has to be known with a precision of 0.02%, which represents the most serious challenge for this measurement.

Measurements of the Triple-Gauge coupling

The study of Triple-Gauge Couplings, i.e. couplings of the type $WW\gamma$ or WWZ, provides a direct test of the non-abelian structure of the Standard Model gauge group and at the same time may provide hints for new physics, since many new processes are expected to give anomalous contributions to the triple-gauge vertex. Anomalous couplings can affect both the total cross section and the shape of the differential distributions of the gauge boson pairs ($W\gamma$, WZ and WW) coming from Triple-Gauge couplings. Already with the modest integrated luminosity expected after one year at low luminosity, these measurements would already improve on the final Tevatron and LEP2 precision, providing powerful constraints on physics beyond the SM.

1.4 References

1-1 D. Boussard et al., '*The Large Hadron Collider Conceptual Design Report*', CERN / AC / 95-05 (1995).

1-2 The NLC Accelerator Design Group and the NLC Physics Working Group, '*Physics and Technology of the Next Linear Collider*', FERMILAB-Pub-96 / 112

1-3 *'Review of Particle Physics'*, The European Physical Journal C 15 (2000) 1

1-4 ALEPH, DELPHI, L3 and OPAL Collab., The LEP Electroweak Working Group and the SLD Heavy Flavour and Electroweak Groups, '*A Combination of Preliminary Electroweak Measurements and Constraints on the Standard Model*', CERN-EP-2000-016 (January 2000)

1-5 Chris Quigg, '*Electroweak Symmetry Breaking and the Higgs Sector*', FERMILAB-Conf-99 / 033-T (May 1999).

1-6 ALEPH, DELPHI, L3 and OPAL Collab., The LEP working group for Higgs boson searches, '*Searches for Higgs bosons: Preliminary combined results using LEP data collected at energies up to 202 GeV*', CERN-EP-2000-055 (April 2000)

1-7 ALEPH Collaboration, '*Observation of an Excess in the search for the SM Higgs Boson at ALEPH*', CERN-EP / 2000-138 (2000), paper submitted to Physics Letter B.

1-8 The CDF II Collaboration, '*The CDF II Detector: Technical Design Report*', FERMILAB-PUB-96 / 390-E, October 1996

1-9 The DØ Collaboration, '*The DØ Upgrade*', Fermilab Pub-96 / 357-E (1996)

1-10 D.E. Groom et al., '*The European Physical Journal*' C15 (2000) 1

1-11 Altarelli, G; Mangano, M L., '*QCD*', proceding at 1999 CERN Workshop on Standard Model Physics (and more) at the LHC, CERN, Geneva, Switzerland, CERN-2000-4

1-12 ATLAS Collaboration, '*ATLAS Technical Proposal*', CERN / LHCC / 94-43(1994)

1-13 CMS Collaboration, 'CMS Technical Proposal', CERN / LHCC / 94-38 (1994)

1-14 LHCb Collaboration, '*LHCb Technical Proposal*', CERN / LHCC / 98-4 (1998)

1-15 ALICE Collaboration, '*ALICE Technical Proposal*', CERN / LHCC / 95-71 (1995)

1-16 ATLAS Liquid Argon group, '*The ATLAS Liquid Argon Electromagnetic Calorimeter*', ATLAS internal note, ATL-CONF-2000-007

1-17 M.Spira, '*Higgs Production and Decay at Future Machines*', CERN-TH / 97-323 (November 1997)

1-18 ATLAS Collaboration, '*ATLAS: Detector and Physics Performance Technical Design Report*', CERN / LHCC / 99-14 (May 1999)

CHAPTER 2

The ATLAS Data Acquisition System

2.1 The Trigger and DAQ System of the HEP Experiments

A common characteristic of high energy physics experiment is the necessity to select (trigger) rare interesting events from a huge background composed by generally already discovered phenomena. The selection is necessary to reduce the event data rate and volume to manageable values for permanent storage and subsequent data analysis. The ability to produce physics results are very dependent upon the capability of the data acquisition system to acquire data at high rate and to select the most interesting events to be saved for analysis. This task must be executed maintaining excellent and unbiased efficiency for the interesting signals. The required selection is achieved on-line via a data acquisition system organized in different levels (triggers). Each level refines the decisions made at the previous one and, where necessary, applies additional selection criteria. The time available for event processing increases in each level of the trigger, which permits the use of an increasing amount of information to either accept or reject the event.

The trigger system of an HEP experiment has typically a three level architecture with each level providing a rate reduction sufficient to allow for processing in the next level with minimal dead-time. The first two levels (Level-1 and the Level-2) are usually implemented on custom hardware and operate on a limited subset of the data, while the last level (Level-3) is a processor based trigger implemented on a processor farm running on the full event read-out. Figure 2-1 shows the schematic for the trigger and data acquisition system of the CDF II [2-1] experiment, which utilises a typical trigger architecture of present collider experiments and represents the primary benchmark for the two general purpose LHC detectors.

2.1.1 First trigger level

The first trigger level operates at the full bunch crossing frequency and must provides a decision for each bunch crossing. Therefore it is characterized by very low fixed latency[1], which is however larger than the bunch spacing. The data coming from collisions occurring during the latency period are held in local buffers (pipelines) located on or near each detector. For instance the RunII configuration of the TeVatron collider foresees a bunch spacing of only 296 ns and the CDF II Level-1 latency is 5.3 μs, corresponding to about 18 bunch crossing.

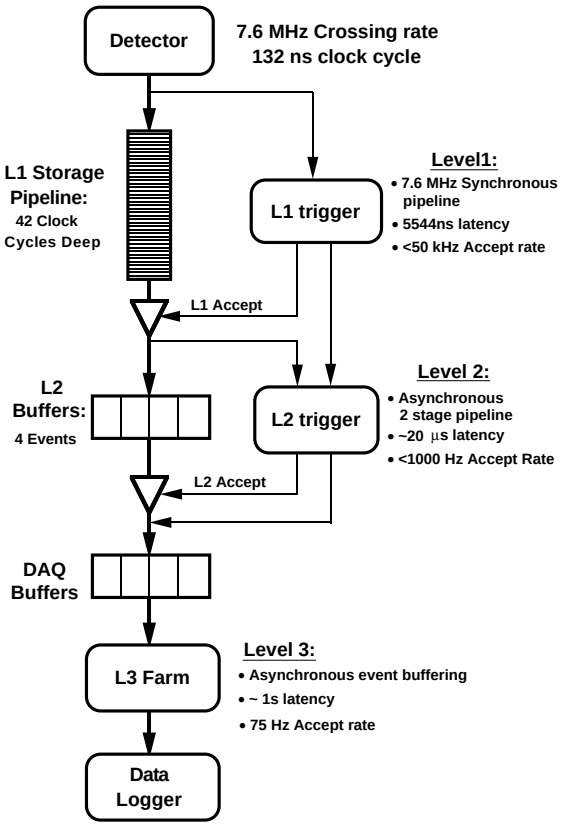

Figure 2-1 Functional block diagram of the CDF II data-flow [2-1].

The first trigger level of the HEP experiments uses custom designed hardware to find physics objects (i.e.: muons, jets, e.m. clusters, etc.) based on a subset of the detector information. The input comes from the calorimeters, tracking chamber, and muon detectors. The decision to retain an event for further processing is based on the number and energies of electron, muon, and jet candidates, as well as the $E_{\mathrm{T}}^{\mathrm{miss}}$ in the event. The hardware usually consists of parallel synchronous processing streams, which feed inputs of the single global Level-1 decision unit. These streams concern different sub-detectors: one stream can be dedicated to the search of calorimeter based objects, while another one can be devoted to identify tracks in the muon system. Event accepted by the Level-1 system are moved by the front-end electronics to the Level-2 asynchronous buffers (the Level-1 pipelines are instead synchronous with the collision rate).

2.1.2 Second trigger level

The task of the second trigger level is to reduce the Level-1 rate to a value at which full event building is possible. It has access to more and improved input data respect to Level-1. Generally the Level-2 system takes advantage of improved momentum resolution for tracks, finer angular matching between muon stubs and central tracks, and data from the central calorimeter detector for improved identification of electrons and photons.

The Level-2 trigger of the CDF experiment consists of several asynchronous subsystems which provide input data to programmable processors, which evaluate if any of the Level-2 triggers is satisfied. It has access to more and improved input data respect to Level-1. Furthermore, the Silicon Vertex Tracker provides, for the first time in a hadron-collider experiment, the ability to trigger on tracks with large impact parameters. The average decision time is about 20 μs.

1. The time taken to collect data, form the trigger decision and distribute it.

Differently from CDF, which has an hardware based Level 2, other experiments implements the functionally of the second level of trigger (i.e.: the need to reduce the rate to a level at which event building is possible) on processing farm. For instance, the second level of trigger of the HeraB experiment [2-2] at Desy is based on a PC farm shared with the third level of trigger. The processing time is limited to about 1 ms.

2.1.3 Third trigger level: the Event Filter

The Event Filter selection stage has become a fundamental component of the HEP Trigger and Data Acquisition architectures. The third level of trigger (the fourth level for some experiments) of a HEP experiments is located downstream of the Event Builder (EB), whose task is to collect and assemble all the data fragments relevant to a bunch crossing, as recorded by the different sub-detectors, into a complete event. Whereas before event building each event is composed of many fragments, with one fragment in each front end crate, after event building the full event is stored in a single location.Therefore the third level operates on full event data and it is generally referred to as “Event Filter” (EF). The computing instrument of the EF is organized as a set of independent sub-farms, each connected to one output of the EB switch. The Data Acquisition system (DAQ) is responsible for collecting data fragments from front-end electronics systems for events satisfying the Level-2 trigger and sending them to the EF subsystem.

The EF primary function is the reduction of data flow and rate to a value acceptable by the mass storage and by the subsequent offline data analysis step. The aggregate throughput reduction can be achieved both by rejecting events and by compressing them. The EF can also provide initial event sorting into stream for offline production and global physics and detector monitoring, essential to ensure the quality of recorded data. Indeed the geographical location of the EF in the data-flow is ideal to perform such checks at minimal cost in terms of required CPU and bandwidth. Full event information allows to investigate data characteristics which are not available at earlier stages and to monitor trigger performance: inter-detector correlation to evaluate the global experiment efficiency, monitoring of the LVL1 and LVL2 selection, alignment checks.

The EF is generally characterized by modest input rate, as low as possible output rate (compatibly with the physics goals) and very high computational needs. Indeed, whereas the first and the second level of trigger are latency and bandwidth dominated, the EF is highly processing time dominated. For these reasons the present trend is toward the implementation of the EF on commercial PC-farm which offer very high performance to cost ratio (PCs are commodity products, thus cheap, and by today provide considerable computing power.).

The Figure 2-2 outlines the DAQ system of the CDF II experiment. If an event is accepted by the Level-2 trigger, the DAQ collects all data fragments from front-end electronics system and transfers them, via a commercial 16×16 ATM network switch, to a Level-3 CPU node, where the complete event is assembled and analysed in order to reduce the rate from 300 Hz to about 50 Hz. The CDF II EF is a processor based filtering mechanism designed as a PC farm, which is organized in basically independent sub-farms each hanging off from one ATM switch output. The farm will be composed of 144 bi-processors PC (500 MHz Intel Pentium III), for a total processing power of about 1.1 kSPECint95.

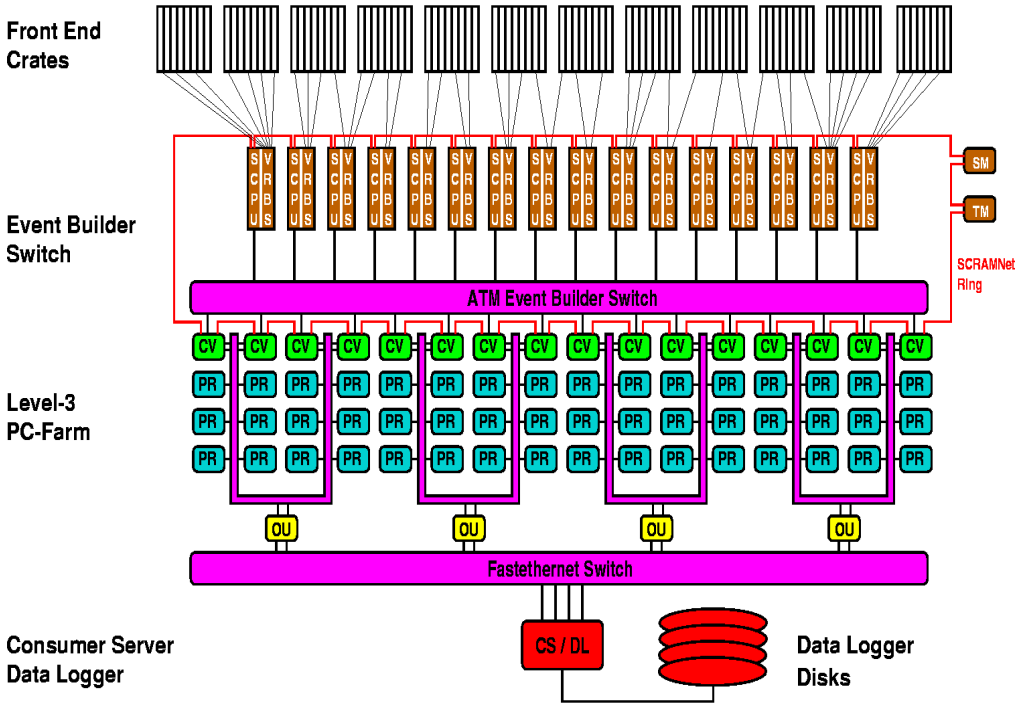

Figure 2-2 The CDF run II data acquisition system which is responsible of moving the event data from the detector read-out links to the final mass storage.

2.2 The challenges for the Trigger/DAQ architectures at LHC

The performance and the physics research potentialities that will be provided by the LHC entail significant challenges for the design and implementation of the trigger and data acquisition systems of the four planned LHC experiments. The selection of rare interesting physics processes with high efficiency, while rejecting much higher rate background phenomena, must start at the unprecedented total interaction rate of about 10^9 interaction per second, two order of magnitude higher than the ones at current experiments. Recording all the information from all the LHC collisions requires, for every second of operation, the equivalent 10000 Encyclopaedia Britannica. The tolerable storage data flux is instead “only” of the order of 100 MB/s, equivalent to a data production of 10 TB per day.

The task of the Trigger/DAQ system is to select, out of these millions of events, the most interesting 100 per second, and then store them for further analysis. The challenging collision rate, the dimensions and complexities of the detectors and the huge number of electronic channels (about 10^8) per experiment, require a number of connections and a data flow flux never been previously met. A large number of event builder input source (about 1000) are required for reading out the detector data. A similar number of destinations or event filter units are needed to handle the resulting event rate. Therefore the bandwidth requirements for such a communication network are very challenging. The data rate handled by the LHC event builders (about 500 Gbit/s for CMS) is equivalent to the amount of data exchanged in a second by world telecom today. The “slow control” (temperatures, voltages, status, etc.) alone of an LHC experiment entails a data rate comparable with the current LEP experiment read-out rate (about 100 kB/s).

The data-flow needs of the LHC DAQ systems are shown in Figure 2-3 and Table 2-1 together with the data-flow performance of some present experiments. Both the DAQ input rate and

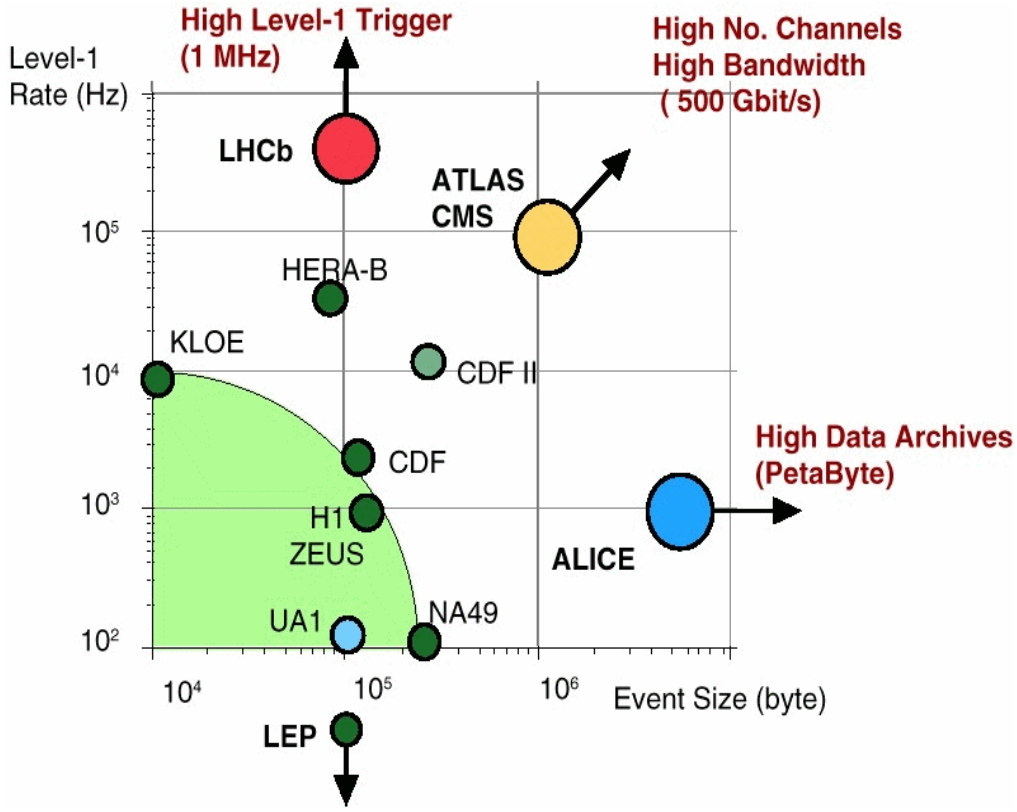

Figure 2-3 The trigger DAQ evolution [2-7].

the event size of the LHC experiments are at least one order of magnitude greater than the presents experiments. Both two general purpose detectors [2-3][2-4] need a total communication bandwidth up to 500 Gb/s, the LHCb experiment [2-5] is characterized by a huge Level-1 trigger rate while the major challenge of the ALICE detector [2-6] is the enormous event size of about 67 MB.

The total computing power needed by the LHC Event Filter farms is estimated to be about 10^5 kSPECint95, two order of magnitude greater than the CDF II one.

Table 2-1 Some DAQ parameters of the planned LHC detectors compared to two current experiment

	HERA-B	CDF II	ATLAS	CMS	ALICE	LHCb
Electronic Channels (M)	0.6	0.75	150	98	274	0.9
Event Dimension (kB)	100	150	~1000	1000	67000	100
EF input rate (Hz)	500	300	~2000	1000	50	50000
Logging rate (Hz)	20	75	~200	100	50	200
Logging Flow (MB/s)	2	4.4	~200	100	1250	20
Proc. Power (kSPECint95)	~1	1.25	~100	125	125	125

2.2.1 Overview of the trigger strategy at LHC

The two general purpose detectors, ATLAS and CMS, having about the same physics goals, share the same requirements and challenges for the trigger system. The first trigger level of the two DAQ architecture reduces the data rate from the 40 MHz bunch-crossing rate to about 100 kHz using reduced granularity calorimeter and muon data. The latency is few μs. Then the High Level Triggers reduce the event rate to a level that can be written to permanent storage: the order of 100 MB/s equivalent to ~100 Hz of full events. The full granularity data, including tracking informations, are analysed in processor farms. The present ATLAS design envisages separate farms for LVL2 and EF, while the CMS one is based on a single computing farm for the two levels. This solution entail the use of a unique network switch for the two HLT and so a greater network bandwidth.

The LHC-b trigger design [2-5] is based on four levels of selection. Parallel Level 0 triggers select high p_T muon, electron, and di-hadron candidates in the muon and calorimeter systems. The Level 0 also has a pile-up veto, designed to select bunch-crossings with a single interaction by rejecting events with large energy deposit in the calorimeters (2.2 TeV or more). The second level of trigger consists of a hardware track/vertex trigger with a maximum latency of 50 μs. Level 2 and Level 3 algorithms are performed in a processor farm. The average event size at LHC-B is about 100 kB, the maximum event building rate is 40 kHz and the storage rate is 200 Hz.

The ALICE [2-6] trigger design must take into account the very low beam luminosity planned for nucleon-nucleon interactions (10^{-27} $cm^{-2}s^{-1}$ for Pb-Pb). The event rate in the TPC is limited to about 8 kHz. A Level 0 analog track-multiplicity trigger, with a latency of 1.2 μs, will be used to signal bunch-crossings with an interaction (roughly one in a thousand). The Level 1 trigger, with a latency of 2.4 μs, is based on muon and calorimeter data, in addition to the multiplicity signal. The Level 2 trigger uses the data from the silicon detectors and the TPC in specific processors with latencies limited to 100 μs to apply more selective algorithms, such as mass cuts. The Level 2 trigger will also provide past and future protection to ensure that there is a single interaction present in the TPC. After event building, further data reduction will be realized in the commercial processors responsible for compressing and logging the selected events to permanent storage.

2.3 The ATLAS trigger/DAQ architecture

2.3.1 The ATLAS trigger requirements

ATLAS trigger requirements can be summarized as follows:

- Rates: the cross-section range for various processes spans more than 10 order of magnitude. The non diffractive pp cross-section of 70 mb corresponds to an interaction rate of 10^9 Hz at design luminosity. The physics process of interest have cross-sections that are many orders of magnitude smaller (see Figure 1-6). $b\bar{b}$ production occurs at 7 MHz, W production at 2 kHz and t production 80 Hz. For a SM Higgs boson of m_H = 150 GeV the expected rate is about 3 Hz. These rates are further reduced by the small branching ratios to detectable channels (e.g.: the $H \rightarrow \gamma\gamma$ branching ratio is less than 10^{-3}).

- Throughput: the huge number of electronic channels lead to an average total size per event of $1 \div 2$ MB of compressed data. On account of a collision rate of 40 MHz, this lead to a prompt data throughput of about 40 TB/s. This unmanageable data rate must be reduced to about 100 MB/s because of the practical limitation in storage capabilities and offline computing power.
- Pseudorapidity acceptance: the trigger capability of calorimeters extends over the same range as the measurement one ($|\eta| < 2.5$ for tracking and electron/photon identification, $|\eta| < 3.2$ for jets detection and $|\eta| < 4.9$ for missing-energy measurements.) The muon trigger covers only the $|\eta| < 2.4$ region, while the precision chambers extends up to $\eta = 2.7$.
- Transverse momentum range: the huge physics programme requires the ability of trigger objects over a wide p_T range. The B-physics studies, scheduled for the initial low luminosity phase, requires the trigger of muon with p_T down to about 6 GeV. A two photon trigger with a p_T threshold of about 20 GeV is required for the $H \rightarrow \gamma\gamma$ search. For jets much higher E_T threshold must generally be used given the large cross section for high p_T QCD processes, although the possibility of using b-jet tagging at the trigger level might allow somewhat lower thresholds to be used for multi b-jet signatures.
- Resolution: the p_T and E_T resolutions determine the sharpness of trigger thresholds. Because of the steeply falling p_T distributions of most processes dominating the trigger rate, thresholds sharpness is essential to limit the feeding of low p_T signals into the p_T range accepted by the trigger.
- Efficiency: trigger thresholds must be somewhat lower than those applied at the data-analysis stage, both to ensure high efficiency for the signal and to allow study of the shape, normalization and background composition.
- Bunch crossing identification: the LVL1 trigger must identify the bunch crossing that gave rise to the trigger without ambiguities.
- Flexibility: the trigger should be able to:
    - follow the changing beam conditions to make optimum use of the available luminosity (for example by changing pre-scale factors during the fill). Indeed the luminosity of the LHC beams will decay during the proton fill. (A possible change of selection criteria during a fill therefore has to be considered at the LHC). The beam size may change somewhat from fill to fill; however within a fill, it is expected to remain constant within few microns. From fill to fill the transverse position of the collision point may fluctuate by up to $\pm$ 50 μm. For tracking algorithms it may therefore be necessary to determine at the start of each fill the transverse beam position (in a coordinate system linked to the ATLAS detector components), and to update the corresponding parameters of the trigger algorithms.
    - to evolve and take maximum advantage of technological advances over a period of about 15 years.
- Selection process should be:
    - tolerant to uncertainties in our current knowledge of rates, data volumes, algorithms, rejection factors and selection strategies.
    - able to allow event rejection to be done at different levels [i.e. LVL2 Trigger or Event Filter (EF)] in order to optimize the efficiency of the selection as a function of the specific physics channel.

- highly flexible and adaptable to future possible evolution of the LHC machine (mode of operation, backgrounds, luminosity, etc.) and to new physics signatures.

Furthermore, for cost and reliability reasons, all the systems should be able to facilitate the use of commercial and commodity equipment[1] and to benefit from industrial standards, where possible. The use of COTS products reduces resources for design, implementations and maintenance of the system.

2.3.2 The ATLAS Trigger/DAQ functional scheme

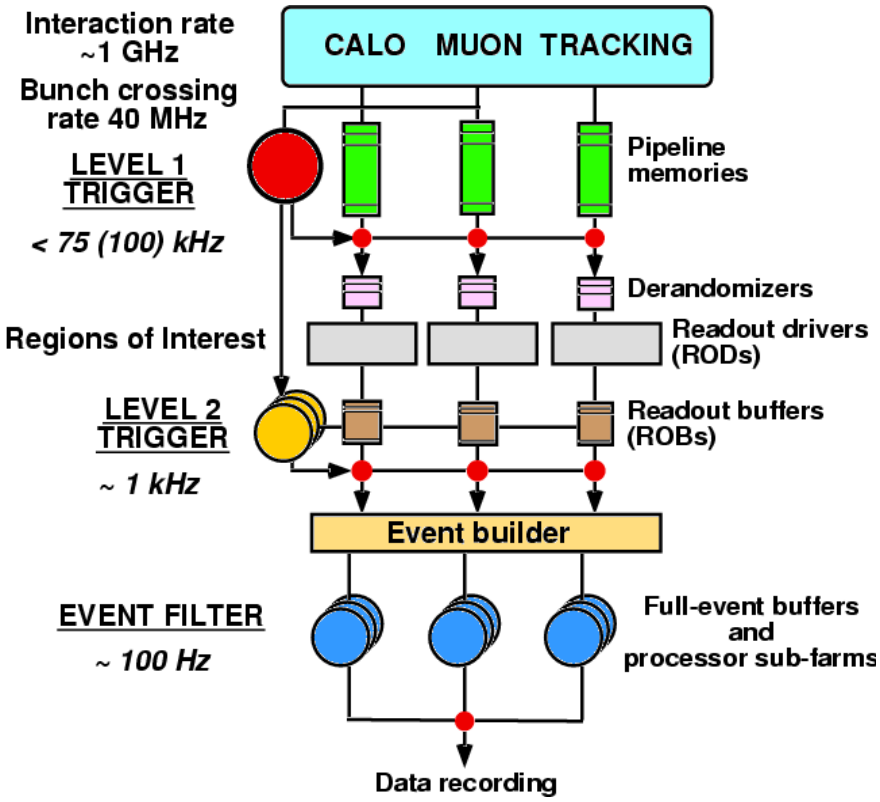

Figure 2-4 Block diagram of the ATLAS Trigger/DAQ system.

Figure 2-4 show the Trigger and Data AcQuisition system (DAQ) of the ATLAS detector as outlined in the ATLAS Technical proposal [2-3] and detailed in Ref. [2-12]. It must be capable of reducing the event data produced at the LHC rate to a manageable data volume for permanent storage and moving and distributing the corresponding huge volume of data from the detector front-end electronics to permanent storage. Rate and data reduction are organized in three trigger levels. The first level of trigger (LVL1), realized in hardware by custom electronics, reduces the data rate from the 40 MHz LHC rate to about 75 kHz. The High Level Triggers (HLT), implemented on two different computer farms, provide a further reduction factor of about 10^3.

1. Commodity products, also called Commercial Off The Shelf (COTS) products, are a subset of the commercial ones. They are products available from general purpose suppliers, e.g. PCs, while commercial also includes specialist products, e.g. VMEbus.

The ATLAS LVL1 trigger is a fast pipelined system accepting data at the full LHC bunch-crossing rate of 40 MHz (every 25 ns). The maximum output rate is expected not to exceed 75 KHz at the maximum LHC luminosity (10^{34} $cm^{-2}s^{-1}$). However the DAQ, the second level of trigger (LVL2) as well as all the detectors are supposed to plan on the basis of a future upgrade in which the LVL1 maximum rate is raised to 100 kHz. At the designed luminosity, each bunch crossing produces an average of 23 pp collisions, which result in an aggregate data throughput of at least 100 GB/s into the data acquisition system. During the time taken to form and distribute the LVL1 trigger decision (latency), the information for all the 10^8 detector channels is stored in pipeline memories placed on or close to the detectors. To reduce the size, and hence the cost of pipeline memories, the LVL1 trigger total latency is limited to about 2 μs, corresponding to 80 bunch-crossing. Local pattern recognition and transverse energy evaluation is performed on prompt, coarse-grained information, which is provided by the fast muon trigger chambers or the tower summing electronics of the electromagnetic and hadronic calorimeter.

When an event is accepted, the DAQ system come into play and the corresponding data fragments from all the pipeline memories are transferred to off-detector read-out cards, containing read-out buffers (ROBs). The ROBs held all the detector data for the bunch crossing selected by the LVL1 trigger until the event is rejected by the LVL2 trigger (in which case the data are discarded), or, in case the event is accepted by LVL2, until the data have been successfully transferred by the DAQ system to storage associated to the event filter (EF). The LVL2 and the Event Builder (EB) only access detector data from the ROBs and thus a clean interface is needed. Data from all the detector must arrive at the ROBs in a specified, uniform format. For this purpose, Read-out Drivers (RODs) interface the detector specific front-end electronics to the common ROB. Intermediate buffers, called derandomizers, average out the high istantaneous data rate at the output of the pipeline memories to match the available input bandwidth of the RODs. The read-out networks of the DAQ system (data-flow) supply the HLT with events or part of them. While the EF is part of the DAQ system, the LVL2 system is separated from the main data flow. It accesses the data required by the LVL2 algorithms, from the ROBs via its own network.

The LVL1 latency enables the high level triggers to analyse many events in parallel on farms of commercially manufacted computers. While the trigger algorithms at LVL1 must be relatively simple in order to be implemented in very fast custom hardware processors, at LVL2 and EF there are more freedom for algorithm complexity and programmability. Indeed both the HLTs may be well implemented using very similar communication and computing structures. They are then distinguished only by the way of accessing detector data and by the framework of software and database access.

The LVL2 architecture is based on the use of region of interest (RoIs). The LVL1 trigger is used to identify, for each event, regions of the detector containing interesting features such as high-p_T electromagnetic clusters, jets and muons. The LVL2 trigger then has to access and process only a small fraction of the total detector data, with corresponding advantages in terms of the required processing power and data movement capacity. So the LVL2, which is still latency bound, applies dedicated algorithms to only a small fraction of the event data in order to calculate some event parameters (p_T, E_T, track multiplicity, etc.) used to select the event. On the other hand, latency is not an issue at EF (because the data are not longer in the ROBs), which has access to all the information of the events selected by the previous level. So more offline-like algorithms are applied in the EF.

The second level is expected to provide a reduction factor of about 100, resulting in a event building rate of the order of 1kHz. The Event Filter (EF) provides the final reduction of a factor 10. However the optimum sharing of the selection task between LVL2 and EF remains to be optimized. Indeed, the technology evolution indicates an increase in CPU processing power by an

order of magnitude over the next five year and an increase in memory density by a factor of four every two years. Therefore a firm division between LVL2 and EF is premature and not desirable. The boundary of operation between the HLTs is overlapped during the research and development phase so that the trade-off of moving event selection between the two can be examined. This boundary is likely to remain flexible during running to optimize the physics performance of ATLAS.

For the about 1 kHz of the LVL2 accepted events, the full detector data are transferred by the DAQ system from the ROBs to the EF system via the Event Builder. It will make use of modern switching technologies to route the event fragments to the appropriate destination. Events, with an average size of $1 \div 2$ MB, should be built at a rate of about $1 \div 5$ kHz. The aggregate bandwidth required is of the order of 5 GB/s and there are several hundred nodes participating to event building. Due to the enormous data rate and volume produced in ATLAS compared to the present HEP experiments, the EB will consist of a big network which will have to be treated in a way similar to the Internet or to the telephony networks, which are compelled to deliver a reliable service to the end users, avoiding congestion in the network and dead time for the clients.

The computing instrument of the EF will be organized as a set of independent sub-farms, each of which is connected to a different output port of the Event Builder switch. Each sub-farm comprises a number of processors analysing several complete events in parallel.The EF must achieve a data storage rate of ~100 MB/s by reducing rate and/or the event size with average decision times of the order of 1 s. An average event size of $1 \div 2$ MB allows a maximum event rate of the order of 100 Hz. The EF will be highly CPU intensive and, unlike at LVL2, the data I/O will be a small fraction of the latency.

2.4 The ATLAS LVL1 trigger system

The ATLAS LVL1 trigger [2-3][2-8], whose block diagram is shown in Figure 2-5, is synchronous with the 40 MHz LHC clock, whose signal is distributed to all the system by the Timing, Trigger and Control system (TTC). The maximum output rate is limited to 75 kHz.

The LVL1 trigger itself is divided into three sub-systems: the calorimeter trigger, the muon trigger and the Central Trigger Processor (CTP). Muon and calorimeter trigger sub-systems perform the initial selection based on reduced granularity data from a subset detectors. The muon trigger uses only the trigger chambers, while the calorimeter trigger uses all of the ATLAS calorimeter, but with reduced granularity. The muon and calorimeter trigger are implemented by purpose-built hardware processors based on ASICs (Application Specific Integrated Circuits) and FPGAs (Field Programmable Gate Array), with a fixed set of algorithms with programmable parameters. They work independently and in parallel

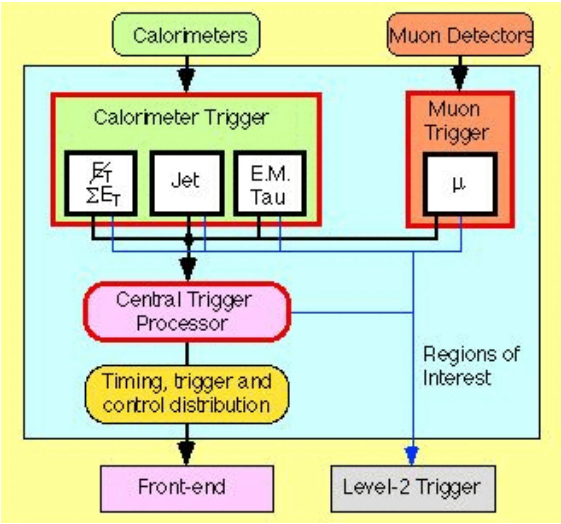

Figure 2-5 Block diagram of the ATLAS LVL1 trigger.

to parameterize events in terms of particle transverse energy (E_T) or momentum (p_T) and multiplicity value.

The features from these triggers are passed to the central trigger processor (CTP) which takes the overall decision to accept or reject the event. Then The CTP passes the decision on to the TTC system for distribution to the front-end system and detector elements. The accepted data, stored in the local pipeline memories until the decision is taken, are sent to the RODs then to the ROBs.

The LVL1 trigger accepts externally-generated trigger signals enabling calibration of the trigger. The LVL1 trigger is connected to the Detector Control System (DCS), which monitors and controls the electronics' racks and crates (temperature, voltage, etc.)

Other requirements on the LVL1 trigger include unique bunch-crossing identification (BCID), which is a particular problem with the calorimeters and the provision of "region of interest" (RoIs) to the LVL 2 trigger so that it only has to read in data around all the trigger objects found at LVL1.

2.4.1 The LVL1 calorimeter trigger

The input of the LVL1 calorimeter trigger [2-8] is made up of trigger towers with a coarse granularity of 0.1×0.1 in $\Delta\eta \times \Delta\phi$, giving in total 7200 channels. The chosen granularity is a balance between rejection of background and cost and complexity of the trigger processor. The energy for each trigger tower is provided as an analog sum of the energies of the corresponding cells. There are separate set of trigger towers for EM and hadronic calorimeter. The LVL1 calorimeter trigger perform four basic algorithms: an e/γ, a hadron$/\tau$, a jet and a sum-E_T/ E_T^{miss} trigger. The basics elements for the algorithm construction are shown in Figure 2-6. Each algorithm identifies objects classified to their transverse energy. The number of object passing a programmable E_T-threshold is counted as multiplicity. The multiplicity of objects which have passed is sent for eight different thresholds to the CTP. The coordinates (RoIs) of the identified object are sent to the LVL2 trigger.

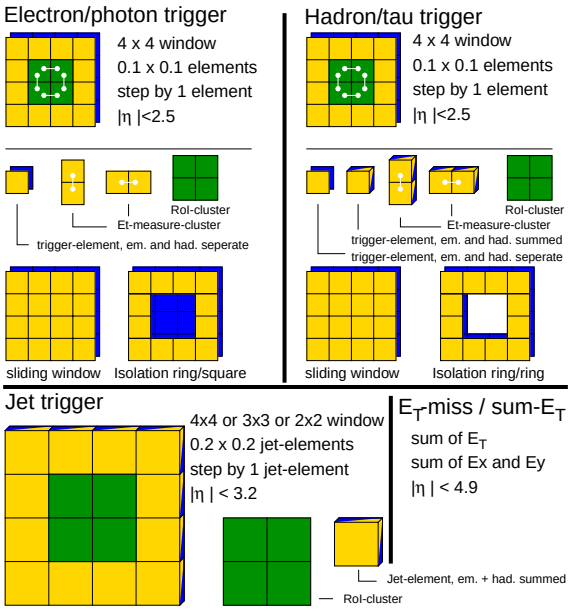

Figure 2-6 The basic elements of the calorimeter trigger algorithms: the electron/photon trigger, the hadron/tau trigger, the jet trigger and the sum-E_T/$E_T$miss trigger.

2.4.2 The LVL1 muon trigger

The LVL1 muon trigger [2-8] is based on the measurement of muon trajectories in two or three different planes, called stations, whose location is outlined in Figure 2-7. The stations are equipped with dedicated fast muon detectors. In the barrel these are Resistive Plate Chambers

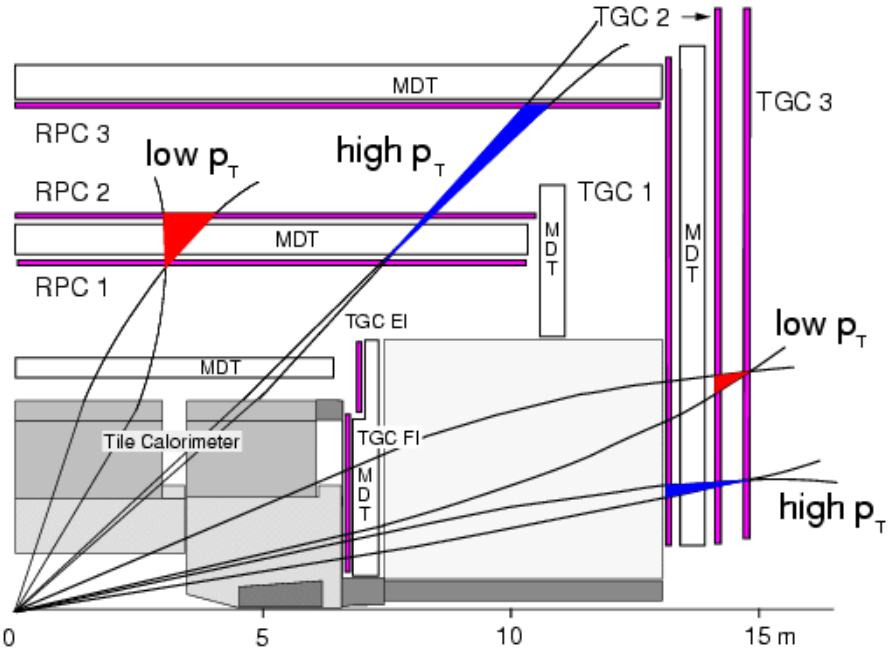

Figure 2-7 The trigger chamber location in the muon spectrometer and the principle of the LVL1 algorithm used to identify muons. Depending on the value of the nominal p_T threshold, roads of different size are used to form coincidences between hits in different layers.

(RPC), in the end-cap Thin Gap Chambers (TGC). Each station has a chamber doublet, except for the inner end-cap station which has a triplet. The timing resolution of both kinds of detectors is sufficient to provide unambiguous identification of the bunch crossing containing a high-p_T muon candidate. Muons are bent by the magnetic field generated by the toroids and their angle of deflection depends on their momentum, the field integral along their trajectory and the Coulomb scattering in the material lying in front of the trigger planes. The energy loss fluctuation is also important for low-p_T triggers.

The algorithm is based on the measurement, using the three trigger station, of the reconstructed track residual respect to the infinite momentum track. The trigger plane farthest from the interaction point, in the end-cap, or nearest to the interaction point, in the barrel, is called the pivot plane. Two different lever arm from the pivot to the other two trigger planes provide two different measurements of the residuals. The two different level arms allow trigger thresholds to cover a wide range of momenta: the shorter level arm (the pivot and station 2) covers a lower momentum region ($p_T > 6$ GeV) and the longer one (the pivot and station 1 in the end-cap, the pivot and station 3 in the barrel) a higher momentum region ($p_T > 20$ GeV).

The p_T resolution in turn is a function of the trigger detector geometry, the magnetic field intensity and its inhomogeneities, the multiple Coulomb scattering in the central calorimeter and the width of the interaction region. The most important physics effect is the Coulomb scattering mainly for the low-p_T threshold. The typical p_T resolution at trigger threshold is 30% in the barrel and 10% in the end-cap for muon of 6 GeV and 40% in the barrel and 15% in the end-cap for muon of 20 GeV.

A tight time coincidence among hits is also required, to

- identify the bunch crossing,

- reduce the trigger rate from accidental coincidences induced by the cavern background.

2.4.3 The Central Trigger Processor

The central trigger processor (CTP) makes the overall decision to accept or reject the event on the basis of the multiplicities of the following local and global trigger object for various p_T thresholds:

- muon;
- electromagnetic cluster, where isolation can be required;
- narrow jets (isolated hadronic τ decays or isolated single hadrons);
- jets;
- the missing transverse energy;
- the total scalar transverse energy.

The global decision at LVL1 is made by the CTP by comparing the list of detected trigger objects to the trigger menu, which have been derived from the physics requirements. They classify the signatures such that a combination of trigger objects is sufficient to select events. Thresholds and attributes for the trigger objects are optimized to meet the requirements of high efficiencies and acceptable rates. A total of up to 96 menu items are currently foreseen for the CTP. The triggers are inclusive, and cover not only physics but also detector monitoring, which must run continuously during physics data taking.

The CTP allows the implementation of programmed pre-scale factors which can be used to down-scale high-rate triggers. These can be used to understand and monitoring thresholds and background, and to take data for standard physics studies (e.g. QCD measurements). The CTP is also responsible for the control of the dead times, which has been introduced in order to have a relatively simple and safe read-out system relying on the presence of data for every event and to simplify the front-end electronics systems by imposing an upper limit on the event rate and a minimum time between consecutive events. The CTP could introduce dead time:

- to accommodate front-end electronics limitations (a short dead time of four bunch crossing is systematically introduced after each LVL1 accept signal)
- to prevent derandomizer overflows (the CTP limits the number of LVL1 accept signals that can be generated within a given period of time)
- to handle the occupancy of the ROD and ROB buffers (the CTP can be vetoed with an external signal)

2.4.4 The TTC system

The ATLAS read-out elements, such as the front-end electronics, the RODs and the ROBs need the bunch-crossing signal (BC) and the LVL1 accept signal (L1A). The TTC system allows these signals to be distributed to the read-out electronics elements. The timing signals comprise the LHC clock and the synchronization signals. The trigger signals include the L1A, test and calibration triggers. The TTC system allows the timing of these signals to be adjusted.

An important feature of the TTC system is that it allows the partitioning[1] of the ATLAS detector for the purposes of detector commissioning, calibration, testing and debugging. The TTC itself is partitioned and sub-detectors can be run with the central ATLAS timing and trigger signals, or independently, with their own specific timing and trigger signals injected directly in the relevant TTC partition. The number of TTC partitions defines the number of concurrent and disjunct data taking sessions that the DAQ will have to be able of support. At present, it is foreseen to have about 40 TTC partitions.

2.5 The ATLAS LVL2 trigger system

2.5.1 Basic architecture

The basic architecture of the ATLAS LVL2 trigger is described in [2-3] and [2-12]. The LVL2 trigger acts on events at the rate retained by the LVL1 trigger, which is estimated at 75 kHz and has to be scalable to 100 kHz. Its role is to reduce the trigger rate to a level which can be sustained by the event building system (in the range of few kilohertz), so that the EF can perform final selection before permanent storage.

The LVL2 trigger uses full granularity, full-precision data from most of the detectors, but examines only regions of the detector identified by the LVL1 as containing interesting information (Regions of Interest, RoIs; Figure 2-8). Consequently only a small fraction of the total detector data must be accessed and processed, with corresponding advantages in terms of required processing power and bandwidth. The RoI information comprises the (η,ϕ) position and p_T threshold range of the local LVL1 objects, such as muon or electromagnetic clusters. The RoI reports are completed by general event properties such as the total scalar E_T and the E_T^{miss} vector. This information is used to decide which ROBs from each sub-detector are associated with each RoI. Only data from these ROBs are transferred to LVL2 processors. For each RoI the features from all the sub-detectors are combined to provide better particle identification and more precise measurements than the ones, that were possible at LVL1. In order to allow for additional requirements to be made at LVL2, the LVL1 trigger provides RoI information also for objects that did not contribute to the selection of the event. Such RoIs, typically for objects of relatively low-p_T, are called secondary RoIs. They are passed to LVL2 purely as additional information about the event.

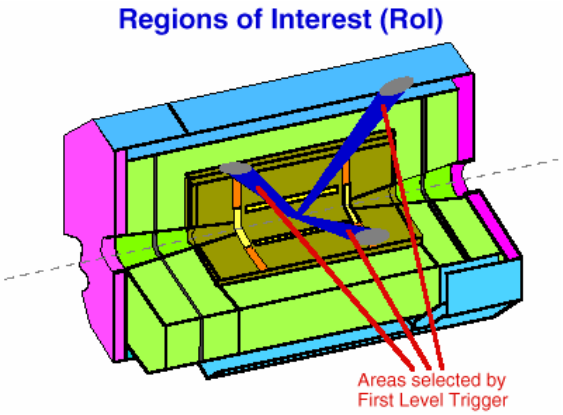

Figure 2-8 The RoI concept..

1. A partition is a subset of the experiment with the capability to acquire data independently. It can be a complete sub-detector, a part of a sub-detector or the combination of several sub-detectors (in other words, several partitions can be combined into a higher-level partition). The data-taking can be done at the level of the RODs or through the complete DAQ system. A partition requires an independent TTC system and an independent handling of the dead time.

In order to keep the average latency for a LVL2 decision to about 10 ms, optimized code for the trigger algorithms (in terms of execution time and amount of data need) has to be used. Most of these algorithms could be implemented in standard general purpose processors but for some tasks it is foreseen the use of custom made FPGA co-processors.

The LVL2 system is separated from the main data flow and accesses the RoI data, the only data required by the LVL2 algorithms, from the ROBs via its own network. The aggregate bandwidth required for the total LVL2 network traffic is estimated to be of the order of several GB/s.

2.5.2 Event selection

The LVL2 must reduce the trigger rate by a factor of about 100. The physics event selection is based on the above-mentioned RoI mechanism and on sequential processing and selection. An average processing time of ~10 ms per event is currently assumed for the LVL2 trigger.The processing sequence at LVL2 trigger can be decomposed into these steps (see Figure 2-9):

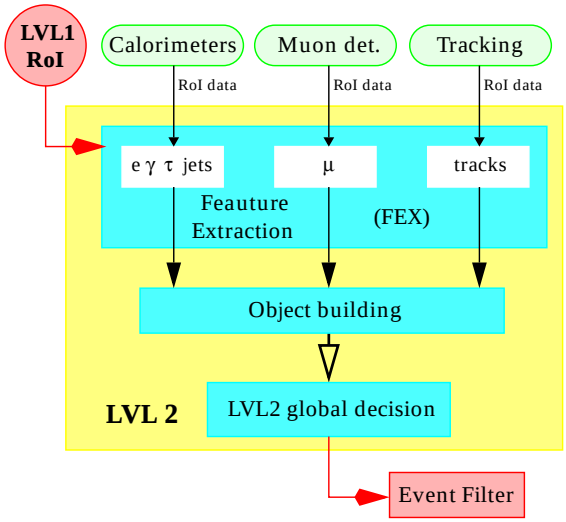

Figure 2-9 Block diagram of the ATLAS LVL2 trigger.

- Data collection and preprocessing. Data associated with the RoIs indicated by the LVL1 are selectively collected from the corresponding ROBs. Preprocessing may be performed prior to transmission of the data to the LVL2 processors. Possible preprocessing functions are zero-suppression or data compression to reduce the required network bandwidth, data reordering or reformatting to simplify the subsequent feature extraction processing.
- Feature extraction (FEX). It is performed for each detector system, starting with the confirmation of the LVL1 RoI in the system from which it originated (muon system or calorimeters). For the calorimeter, this process would take cell information and produce cluster parameter; for a tracker, the basic hit information would be converted to track or track-segment parameter. In parallel of after confirmation of LVL1 RoI, additional features may be searched for in other detectors, such as the SCT/Pixel and TRT. This is the case for muon, e.m. cluster and τ RoIs. Jet RoIs are only processed in the calorimeters, with the possible exception of *b*-jet tagging, which requires tracking detectors to evaluate the impact parameters of tracks.
- Object building. Features from all relevant detectors are combined, to form identified LVL2 trigger objects, which are candidates for muons, electrons, photons, taus, and jets, as well as generalised missing-E_T and *B*-physics objects
- Global decision. All objects found in the event are combined and compared with the topologies (particle types, transverse momenta, inclusive/missing mass, etc.) for a menu of physics selections. A global decision is than taken based on trigger menus.

The FEX algorithms are at the heart of the LVL2 trigger processing. The performance obtained in efficiency and background rejection power determines the overall performance of the LVL2 trigger. The data collection and preprocessing step, which precedes feature extraction, is

important and may be time consuming, but the bulk of the algorithmic complexity lies in the feature extraction. The subsequent object building step, as well as the global-decision algorithm, is comparatively simple.

The LVL2 trigger codes are optimized for fast execution using full granularity data from the RoI. Several trigger types will be implemented: inclusive triggers, triggers on global quantities, combinations of signatures and *B*-physics triggers. The rate reduction at LVL2 is achieved by sharpening the threshold measurements done at LVL1 and running more complex algorithms than were possible for LVL1. In particular, the LVL2 is the first place where data from the tracking detectors are available. The feature extraction (FEX) methods are however simpler and less iterative than those in offline code. They also cannot depend on the ultimate resolution of the detector as not all the calibration data will be available at the trigger level. The trigger selection will change with the LHC luminosity, more importance being given to the low-p_T channels in the initial, lower luminosity running.

The LVL2 strategy for confirming trigger objects is still under study. Such studies are part of the overall optimization of the trigger implementation, taking into account processing power, data bandwidth and cost requirements, which is a joint task of the LVL2, EF and Physics and Event Selection Algorithms (PESA) group (see Section 7.1.2). Much effort is going into the development of algorithms and selection criteria to define trigger objects. Once these are defined, the final global decision is straightforward.

In the case of the muon trigger, the techniques to reduce the rate from LVL1 use data from the muon spectrometer to remove fake LVL1 triggers resulting from noise hits due to radiation in the cavern and will reduce the rate by making a sharper p_T cut. Further rate reduction can be expected from using the Inner Detector to sharpen the p_T cut and to remove some of the background from decays in flight of pions and kaons by requiring a good match between the tracks reconstructed in the Inner Detector and the muon spectrometer. A even better rejection factor is expected from requiring isolation of the muon using information from the calorimeters.

For isolated electrons, rejection power at LVL2 comes from using the full-granularity calorimeter information and requiring a matching high-p_T charged track in the Inner Detector; the transition-radiation signature provides additional rejection power. Similarly, background to the τ trigger can be reduced by requiring the presence of a track matched to the calorimeter cluster. For photons, less rejection power is possible than in the case of electrons, since the ID cannot be used. The photon trigger uses calorimeter features to reduce the background from jets and preserve high efficiency for $H \rightarrow \gamma\gamma$ events. Improvement of the jet trigger is possible for low-E_T jets, but is marginal for high-E_T jets. On the other end the reconstruction of the tracks impact parameter in the ID permits the implementation of a *b*-tagging trigger (see Section 7.2).

B-physics processing is different from standard RoI processing. ATLAS B-physics studies are based on a low-p_T single muon trigger at LVL1. This muon may then be confirmed at LVL2 in the muon spectrometer system and the inner detector systems. For events retained after this initial selection, a full track search must then be performed to allow decisions based on semi-exclusive B-physics hypotheses. The present strategy is to search for tracks in the TRT with very low-p_T thresholds. The resulting TRT tracks are used to define additional RoIs (so-called LVL2 RoIs) that guide further track searches in the SCT. The reconstructed SCT tracks, giving information in three dimensions, allow for the calculation of invariant masses, or they may be extrapolated into the calorimeter or muon systems to confirm low-p_T lepton candidates, in conjunction with the transition-radiation signature from the TRT in the case of electrons.

2.5.3 Functional description of the LVL2 component.

The fundamental technical problem for the implementation of the LVL2 is how to make the data available to the processors executing the trigger algorithms and how to control the use of these processors. The full size of each event is 1 ÷ 2 Mbyte, they are received at the LVL1 accept rate (up to 100 kHz), the data for each event is spread over the ~1700 ROBs of the system, the processing power required for the LVL2 algorithms is of the order of about 25 KSPECint95, the decision for each event is required in an average time of ~10 ms. Indicative performance requirements for the LVL2 components are shown in Table 2-2.

After the submission of the ATLAS Technical Proposal [2-3], it was decided to put in place programmes to investigate the requirements for each part of the LVL2 system and the implications of these requirements for architectures and technologies.

Table 2-2 Indicative parameters and performance requirements for LVL2 components.

Parameter	Value	Comment
Total bandwidth in LVL2 network	~ 5 GB/s	Reduces to ~ 3 GB/s with pre-selection of calorimeter data and TRT data compression
Max RoI request rate per ROB	~ 14 kHz	
Max output bandwidth to LVL2 per ROB	~ 9 MB/s	Reduces to ~ 4 MB/s with pre-selection of calorimeter data and TRT data compression
Max RoI Builder/Supervisor rate	75 kHz	
Average number of ROBs required to supply data per RoI	10–35	Depends on RoI type
Typical number of ROBs per data request	~ 4	In e.m. calorimeter only true if layers treated separately
Data for TRT full scan	~ 200 kB from ~ 256 ROBs	Reduces to ~ 80 kB with TRT data compression
Typical number of RoIs per event	1–2 primary/ ~ 3 secondary	
Average number of sequential steps executed	~ 2	
Event rate per LVL2 processor	> 0.1 kHz	

2.5.3.1 The Demonstrator Programme

The Demonstrator Programme (see references in [2-9]), running from late 1996 to February 1998, studied components for a number of representative architectures and different technology options which might be used for the LVL2 trigger. These included studies of components required for the definition of the architecture which has been presented in the *"ATLAS High-level Triggers, DAQ and DCS Technical Proposal"* [2-10]. Small test-beds were used to study the different components and architectural ideas. The results of the Demonstrator Programme were described in Chapter 4 of the *"DAQ, EF,LVLE2 and DCS Technical Progress Report"* [2-9]. The main conclusions were as follows:

- There was increased confidence that affordable commercial networks would be able to handle the traffic in a single network - a total of a few GB/s between of the order of 1000 ports.
- Standard commercial processors (especially PCs) were favoured for the general LVL2 processing, rather than VME-based systems, since they offer a better price/performance ratio.
- FPGAs could provide fast and powerful processing for specific tasks (e.g. pre-processing and TRT Full Scan).
- Sequential processing steps are needed and sequential selection offers clear advantages. the use of sequential selection reduce the network and processor requirements and allows more complex algorithms to be run at lower rates.

The work and conclusions of the Demonstrator Programme had produced a preferred architecture, described in the next paragraph, and prototype components for some parts of an implementation. However it was clear that further work was required for some key components.

2.5.3.2 The LVL2 Pilot Project

The principal aims of the Pilot Project [2-11] were to validate the LVL2 architecture produced by the Demonstrator Programme and to investigate technologies likely to be required for its implementation. The Pilot Project started in early 1998, and was based on a hardware architecture which has evolved to that shown in Figure 2-10. This architecture aims to use commodity items

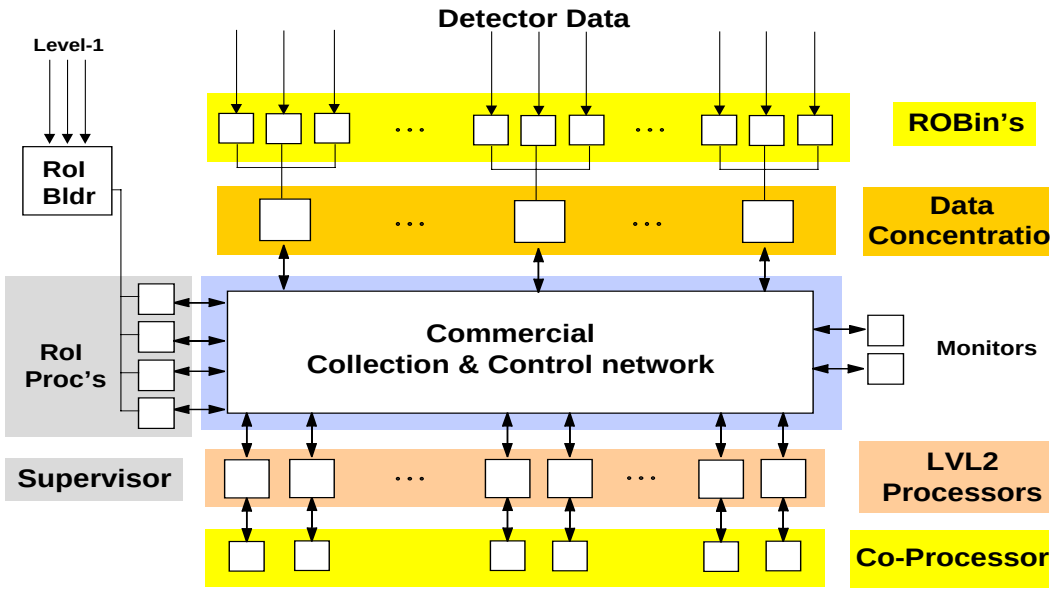

Figure 2-10 The Pilot Project LVL2 hardware architecture.

(processors, operating system (OS) and network hardware) wherever possible.

On a LVL1 accept signal (L1A), the LVL1 trigger prepares and transfers to LVL2 RoI fragments that pinpoint interesting features of the event in specific sub-detectors used by LVL1. The transfer is via fast links (currently seven S-LINKs) with event formatting following the standard ROD protocols with the event number embedded in every fragment. The RoI Builder (RoIB) re-

ceives these event fragments, assembles them into event-specific records, and selects an RoI Processor (in the Supervisor farm) to which it transfers the record. The RoIB is capable of operating at the full LVL1 accept rate (100 kHz). To achieve the necessary performance the design is done in hardware making extensive use of large-scale FPGAs.

A block diagram showing the RoIB/Supervisor structure and associated connections can be seen in Figure 2-11. The data from LVL1 are received in parallel on RoIB cards. Each card is connected to (at most) two RoI processors of the LVL2 Trigger Supervisor. The design is scalable by adding more RoIB cards and more RoI processors. The Supervisors assign each event to one of the LVL2 processors. The LVL2 processor performs one or more steps of data collection and analysis from relevant ROBs. The controlling processor for an event can delegate part of the processing to another processor, for example the FPGA-based co-processor for the TRT Full Scan. The trigger decision can be issued at any step. It is returned to the Supervisor which distributes it to the ROBs. Rejected events are discarded; accepted events are passed to the Event Filter for further analysis. Within this architecture the event selection process is sequential, all processors can access all ROBs, and data collection from the ROBs is initiated by the processors using a request response protocol.

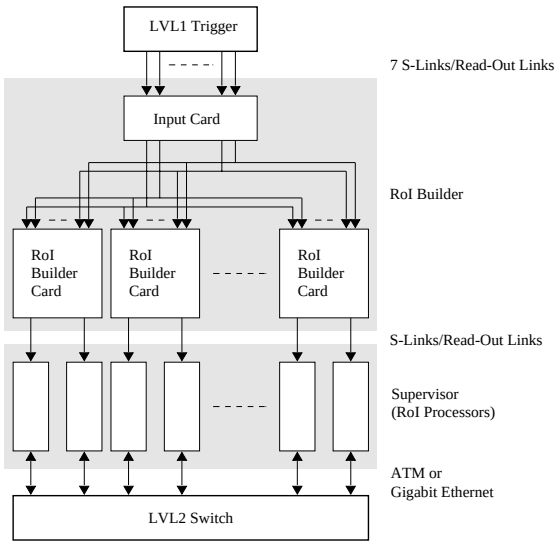

Figure 2-11 Block diagram of the RoIBuilder/Supervisor system.

The work of the Pilot Project was thus divided into three main areas: functional components, test-beds and system design. The functional components covered optimized components for the Supervisor and RoI Builder, the ROB Complex, networks and processors (especially FPGAs). Test-beds covered the development of the Reference Software, a prototype implementation for the complete LVL2 process, and the construction and use of moderately large application test-beds to use this software. Finally, system design covered modelling activities and an integration activity to consider issues related to how the LVL2 system integrates with other subsystems and the requirements it has to meet. Particular emphasis was placed on the so called Reference Software and the application test-beds. The Reference Software was to provide a single software framework across the Pilot Project, building on the conclusions of the Demonstrator Programme. The test-beds were to use this software with the aim of checking that individual components meet the required performance, providing information on scaling up to moderate size systems and producing data for the full-system computer models.

The main conclusions of the Pilot Project are:

- The Reference Software has been run in many configurations, using various network technologies. It has been shown to scale to systems of up to ~ 100 nodes. The tests indicate that the required component performance can be obtained with commodity hardware (PCs) and OS software (Linux). Though not yet optimized the I/O performance obtained with the test-beds gives good indications that the requirements of the LVL2 trigger can be met with this software architecture.

- Extensive results were obtained in test-beds of up to 48 nodes with three optimized network technologies (ATM, Fast/Gigabit Ethernet and SCI) and over MPI on a 96-node commercial cluster. ATM and Ethernet remain as candidate technologies for this application. In both technologies the cost of a large network (~ 1000 ports), including switches and host adapters, appears to be within the ATLAS cost estimates.
- It has been demonstrated that the ROB requirements can be satisfied with current technology: projected input rates can be handled, output to LVL2 and to the Event Builder is achievable at the necessary rates and bandwidth, some on-the-fly pre-processing is possible. COTS hardware seems to be able to support the output requirements but perhaps not the input rates.
- A prototype RoI Builder has been built using FPGAs in a highly parallel architecture. The design uses custom hardware for the most demanding tasks, which reduces the demands on the other Supervisor components, so that they can be implemented using standard processors.
- FPGA co-processors could be included in standard processors in a transparent way, offering significant performance improvements for suitable compute-intensive algorithms and hence a reduction in the size of the processor farms required.
- Models of a full-scale system indicate that the high-p_T LVL2 triggers may be handled by similar number (100 ÷ 200) of 25 SPECint95 processors at both high and low luminosity; the maximum data volume for LVL2 through the network is 20 ÷ 40 Gbit/s for low luminosity (including the inner detector full scan for B-physics) and 10 ÷ 25 Gbit/s for high luminosity; the inner-detector full scan increases the processing power and network bandwidth required in the LVL2 processors significantly. Current estimates indicate that ~ 450 processors, each of 25 SPECint95 and with an FPGA co-processor, would be needed. Without the co-processors the number increases to about 770.

2.6 Event Filter

For the about 1 kHz of the LVL2 accepted events, the full detector data are transferred by the DAQ system from the ROBs to the EF system via the Event Builder. The computing instrument of the EF will be organized as a set of independent sub-farms, each of which is connected to a different output port of the Event Builder switch. Each sub-farm comprises a number of processors analysing several complete events in parallel.The EF must achieve a data storage rate of ~100 MB/s by reducing rate and/or the event size with average decision times of up to several seconds. An average event size of 1 ÷ 2 MB allows a maximum event rate of the order of 100 Hz. The EF will be highly CPU intensive and, unlike at LVL2, the data I/O will be a small fraction of the latency.

Event filtering is executed on fully assembled events, containing the entire detector information, including the results of the lower level triggers and the on-line calibration and alignment constants. The selection is based on the off-line algorithms, which, obviously, cannot be executed earlier, either because they require correlated information from different parts of the detector or because they are too complex to be implemented on the LVL2 processors within a time compatible with the latency. Depending on the processing time needed by the algorithms, the processing power available, and the stability of the calibrations and beam position, one may aim at completing in the EF the reconstruction to such a degree that the subsequent analysis steps have only to perform hypothesis-dependent updates of the reconstruction. Algorithms

that could be executed at this level are vertex reconstruction and track fitting, including bremsstrahlung recovery for electrons, or operation that require larger RoI than that used at LVL2, such as γ conversion search or calculation requiring the complete event data (missing E_T calculation). The EF completes the classification of the events, establishes a catalogue of discovery type events and stores accepted events in the database. Events may be directed to separate output stream, for example if they are needed for calibration or alignment only.

In addition to its primary task of event filtering and data reduction, the EF, being the first element in the DAQ chain having access to complete events, will also be used for calibration and monitoring studies. Many types of monitoring, including inter-detector efficiencies, physics quality assurance and trigger performance can be envisaged, as well as specialized calibration studies.

The EF is, functionally, an integral part of the main DAQ data-flow. Events being processed by the EF program are stored in the EF processor memory before being discarded or accepted for permanent storage. Therefore, the DAQ and the EF are considered as one system (DAQ/EF).

For the DAQ/EF system, a pre-prototype programme, the DAQ/EF-1 project [2-13], has been developed. This project aims at producing a system with full functionality but not necessary the full performance required for the final system. DAQ/EF-1 covers almost the full functionality of the DAQ and the Event Filter systems. The only missing aspect is the LVL2 DataFlow, which was studied in combination with the LVL2 Selection system in the Pilot Project (Section 2.5.3.2). The description of the studies and the results of the DAQ/EF-1 project is the argument of the next chapter of this thesis.

2.7 References

2-1 The CDF II Collaboration, '*The CDF II Detector: Technical Design Report*', FERMILAB-PUB-96/390-E, October 1996

2-2 The HERA-B Collaboration, '*HERA-B, Report on Status and Prospects*', DESY-PRC 00/04 (2000)

2-3 ATLAS Collaboration, '*ATLAS Technical Proposal*', CERN/LHCC/94-43 (1994) ISBN 92-9083-067-0,

2-4 CMS Collaboration, 'CMS Technical Proposal', CERN/LHCC/94-38 (1994)

2-5 LHCb Collaboration, '*LHCb Technical Proposal*', CERN/LHCC/98-4 (1998)

2-6 ALICE Collaboration, '*ALICE Technical Proposal*', CERN/LHCC/95-71 (1995)

2-7 S.Cittolin, *'LHC experiments Data Communication and Data Processing systems'*, 1999 CERN School of Computing, Stare Jablonki Poland; http://cmsdoc.cern.ch/cms/TRIDAS/distribution/3.Misc/talks/CSC99.pdf

2-8 ATLAS Collaboration, *'Level 1 Trigger TDR'*, CERN/LHCC/98-14 (1998)

2-9 ATLAS Collaboration, *'DAQ, EF,LVLE2 and DCS Technical Progress Report'*, CERN/LHCC/98-16 (1998)

2-10 ATLAS Collaboration, *'ATLAS High-level Triggers, DAQ and DCS Technical Proposal'*, CERN/LHCC/2000-17 (2000)

2-11 ATLAS Collaboration, *'The Level-2 Pilot Project'*, ATLAS Internal Note, ATL-DAQ-118 (1998).

2-12 L.Mapelli, *'Global architecture for the ATLAS DAQ and Trigger'*, ATLAS internal note, ATL-DAQ-95-022 (1995)

2-13 G.Ambrosini et al., *'The ATLAS DAQ and Event Filter prototype '-1' project'*, CHEP97, Berlin (1997)

CHAPTER 3

The DAQ/EF -1 project

3.1 Introduction

The DAQ/EF prototype "-1" project[3-1] (DAQ/EF -1), started in January 1996, consists in the design and implementation of a full “vertical” slice of the functional DAQ and EF architecture outlined in the ATLAS Technical Proposal[3-2] and detailed in Ref. [3-3]. It follows, and in part is based upon, previous investigations of DAQ-related issues for the LHC experiments [3-4]. The prototype takes its name from the definition of the project to be able to support the full functionality of the final system, but not necessarily to achieve the final performance. It includes all the hardware and software elements of the data-flow, its control and monitoring, as well as all the elements of a complete on-line system, from detector read-out to data recording. It is designed to support the evaluation of various technological and architectural solutions. Issues related to the software development environment, from the operating systems to software engineering and CASE tools, are also addressed.

The project has been organized in a number of systems, following the features of the global architecture, as shown in Figure 3-1:

- The DataFlow system is responsible for moving the event data from the detector read-out links to the final mass storage. It provides event data for monitoring purposes and implements local control for the various DataFlow elements.
- The BackEnd system encompasses the software needed to configure, control and monitor the DAQ, but excludes the processing and transportation of physics data. The BackEnd software is essentially the glue that holds the subsystems together. It does not contain any elements that are detector specific as it will be used by all possible configurations of the DAQ and detector instrumentation. The DAQ system includes interfaces required by the triggers, processor farms, accelerator, event builder, data-flow and detector control system (DCS).

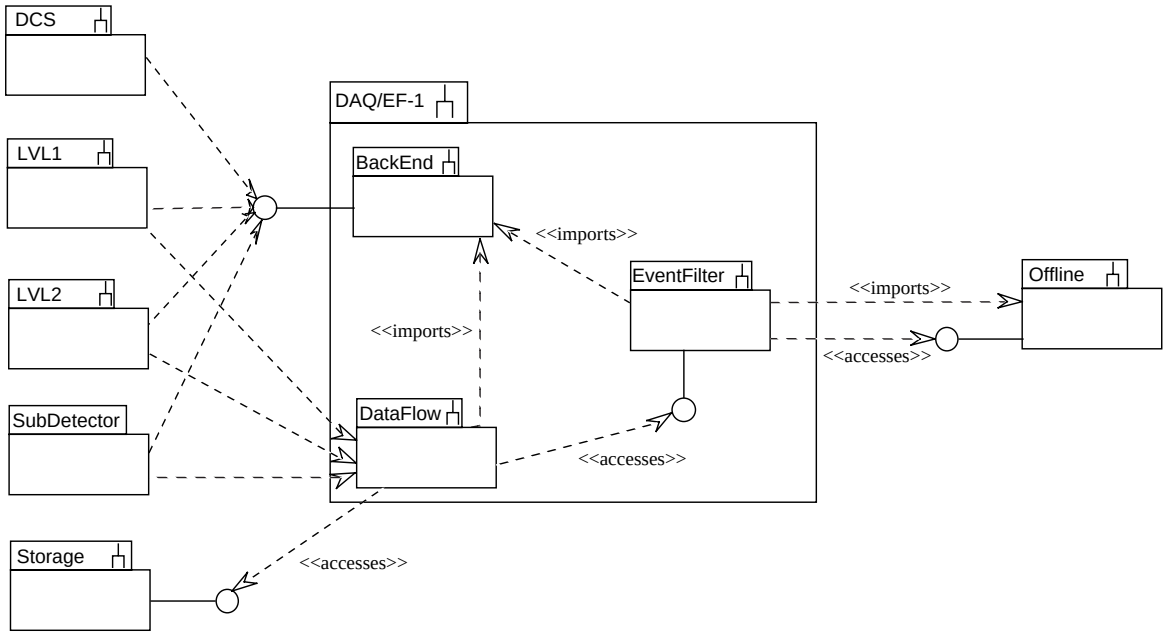

Figure 3-1 Top-level diagram of the DAQ/EF -1 architecture and its context.

- The Event Filter system is in charge of distributing events to the different processing nodes of the SubFarms connected to the ports of the Event Builder (EB) and of providing the means to supervise (control and monitoring) the whole Farm. Algorithms to be used for the filtering operation are the concern of the PESA (Physics and Event Selection Algorithm) group (Section 7.1.2).

Not shown in Figure 3-1 is the Detector Interface System. It is not a system in a classical sense. In the context of the DAQ/EF -1 overall system, it can be seen as a virtual subsystem encompassing the set of requirements and critical detector information necessary for the prototype design. The collection of such requirements and detector parameters and the understanding of the impact of relevant detector features on the DAQ/EF -1 design was done via a Detector Interface Group (DIG). Originally set up in the context of and as an integral part of the DAQ/EF -1 project, the DIG soon grew to assume the role of a very useful forum between the DAQ/EF -1 team and the detector and other ATLAS system communities to discuss and analyse common DAQ issues. Areas such as run control, database access, system partitioning, monitoring, calibration and local data acquisition were addressed by this requirement collection process. The DIG has proved to be an extremely important element in the development of the DAQ/EF -1 prototype. Viewing detectors as systems which have to be serviced by the DAQ has been very valuable.

3.2 The DataFlow system

3.2.1 Design

The DataFlow system is responsible for moving the event data from the detector read-out links to the final mass storage. It provides event data for monitoring purposes and implements

local control for the various DataFlow elements. It interfaces with other parts of the ATLAS detector, in particular the first and second level trigger. Furthermore, it provides the means for partitioning the data acquisition system, allowing subsets of the detector to take data independently and concurrently for calibration, debugging and testing purposes.

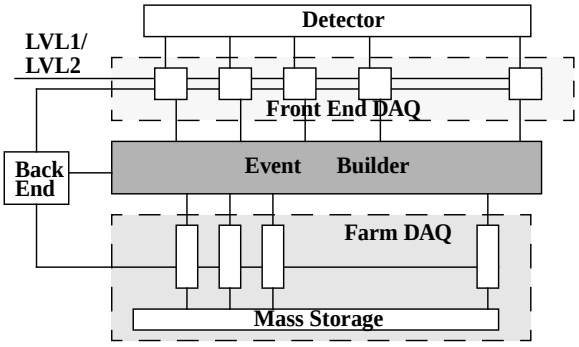

Figure 3-2 The DataFlow system.

The DataFlow system provides the functionality of: receiving and buffering data from the sub-detector RODs; building of complete events from individual event fragments; distribution of events and event fragments to the HLTs; the sending of events to final storage. A global view of the DataFlow is shown in Figure 3-2.

The segmentation of the detector read-out into ReadOut Driver (ROD) modules suggests to organise the Front-End DAQ into a number of modular, independent elements, the ReadOut Creates (ROCs), each supporting the read-out from one or more RODs and having one or more connections to the EB. In a similar way, the Event Filter Farm is (logically or physically) segmented into independent modular elements, each connected to one of the EB outputs and treating one or more events. Such a modular element is called a SubFarm, its DAQ part is the SubFarm Crate (SFC).

The ReadOut Crate (ROC) and the SubFarm Crate (SFC) are modular subsystem that works concurrently and independently with respect to any other ROC (SFC) in the system. Each modular element connects independently to the EB. These permits the fulfilling of the DAQ partitioning requirement, as shown in Figure 3-3.

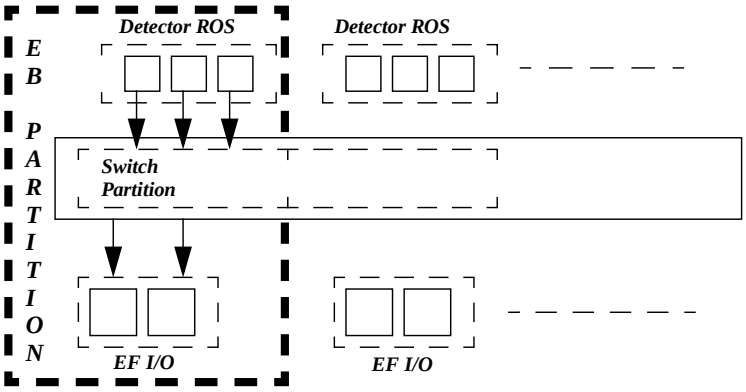

Figure 3-3 An example of DAQ partition.

The DataFlow, composed of the ROC, the EB and the SFC physical subsystems, has been is factorized in three logical subsystems:

- The DAQ-Unit: the component of the ROCs and SFCs responsible for receiving, buffering, merging and distributing detector event data. The ROC receives, buffers and merges ROD fragments from the detector and forwards fragments to the LVL2 system and to the EB subsystem. The SFC receives and buffers full events from the EB; sends events to and

retrieves events from the Event Handler component of the Event Filter; and sends events to mass storage.

- The EB: the component of the DataFlow which merges data fragments coming from different parts of the detector into full, formatted events. It interfaces to a ROC via an Event Builder Interface (EBIF) element and to an SFC via a SubFarm Input (SFI) element.
- The Local DAQ (LDAQ): this subsystem provides all the common DAQ functions which are not related to moving event data. These functions are common throughout the DataFlow system: local control, initialization and configuration, support for event monitoring. In addition the LDAQ subsystem is also the interface and integration point between the DataFlow and Back-End systems. An integrated DAQ/EF -1 system will include instances of the LDAQ, specialized towards a specific DataFlow system: one LDAQ per ROC, one LDAQ per SFC and one LDAQ per partition.

3.2.2 The DAQ-Unit

The implementation of DAQ applications has been organized such that the actual DataFlow Tasks are decoupled from the framework of an application. A DataFlow task is defined as the entity which provides all the functionality required to handle an external I/O channel, such as a detector read-out link or an Event Builder port. The infrastructure of every application has been coded in a Generic I/O Module library: here the scheduling policy for the Tasks is defined and the communication with the LDAQ is established. This factorization allows to easily try out different groupings of DataFlow Tasks into a single application in order to optimize the DataFlow performance. The combination of one or more Tasks and the Generic I/O Module library forms an I/O Module (IOM). Examples of IOMs are the EBIF, in which the output from the ROC to the Event Builder is handled, or the SFI, in which the input of full events from the Event Builder and the output of the events towards the Event Filter are handled.

The implementation of the DAQ-Unit software is based on layering. It means that the hardware and operating-system-independent software packages have been insulated from the hardware and operating-system-specific packages, thus supporting portability and maintenance.

A fundamental principle in the deployment has been the maximum use of COTS products. The DataFlow prototype has been implemented on VME bus [3-5] based single board computers running LynxOS[1] as operating system. As an alternative, PCs running Linux [3-7] have been introduced at the SFC and LDAQ level. Communication protocols have been defined and developed to support both the intracrate [3-8] and the event building traffic [3-9]. These protocols hide the technology specific implementation aspects of the communications allowing to change bus types, switching networks, etc. Furthermore, the implementation of DAQ applications has been organized such that the data flow tasks are decoupled from the infrastructure (e.g. task scheduling and buffer management) of a process. Such a factorization gives the possibility to move tasks between processors and find the optimal CPU and buses utilization.

1. LynxOS is a UNIX-compatible, POSIX-conformant multi-process and multi-threaded real-time operating system for embedded applications [3-6].

3.2.3 The EB

The Event Building system is responsible, following a signal from a trigger element, for collecting all the event fragments relevant to a bunch crossing into a complete event at a single location. As outlined in Table 3-1, the EB will have to connect hundreds of ROCs and SFCs, build the events at a rate of the order of 1 kHz and provide and aggregate bandwidth of about 5 GB / s.

Table 3-1 Full event builder parameters.

Fragment size / source[a]	1–15 kbyte	Number of sources[a.]	100–1500
LVL2 Accept rate	1–5 kHz	Number of destinations	100–200
Data rate / source[b]	1–75 Mbyte / s		
Total data rate	1–8 Gbyte / s	Number of nodes	200–1700

a. Assumes 1–15 ROBs / source and 1 kbyte / ROB

b. Assumes data from all sources

The EB subsystem and its interfaces to other components of the Trigger / DAQ are depicted in Figure 3-4. The Event Builder [3-10] consists of logical objects and a High Level Protocol

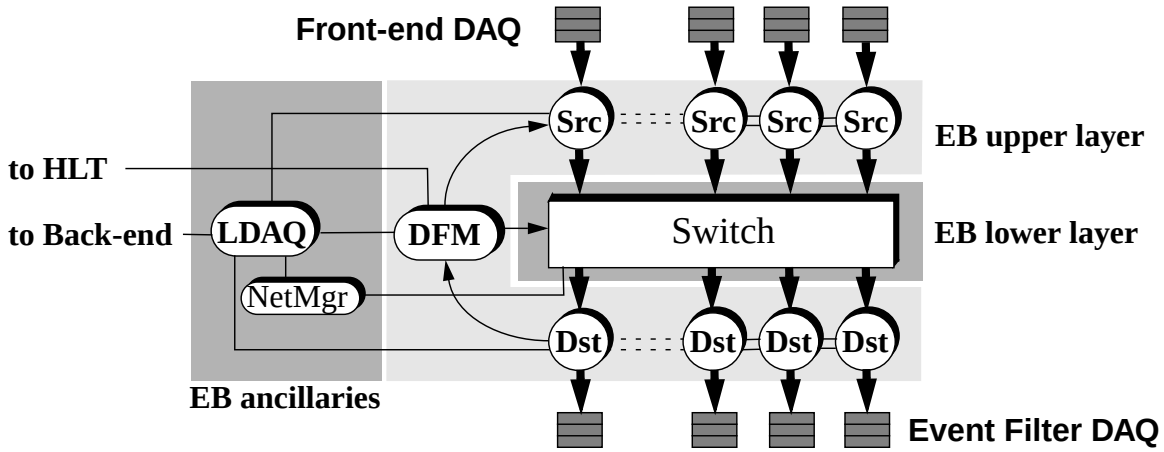

Figure 3-4 Concept diagram of the Event Builder.

(HLP), which defines how the actual event building is performed. Five logical objects have been identified:

- the Source (Src): the logical element which sends out data fragments from the ReadOut Subsystem once the second level Trigger has accepted an event.
- the Destination (Dst): the logical element which receives event fragments and assembles them into fully formatted events.
- the Data Flow Manager (DFM): the logical object which ensures the correct flow of event fragments between Sources and Destinations. When an event is scheduled for Event Building by the trigger system (LVL2 or LVL1 in case of calibration or debugging) the DFM informs the Source of the Destination Identifier (Did) and of the Global Identifier

(Gid) of an event fragment which as to be exchanged with the Destination. Then it traces the reception of event fragments at Destinations.

- the Network Manager (NetMgr): it monitors the status of the network and provides information about the status of the other elements in the EB system. It has, in particular, to ensure the correct initialisation of the network before Srcs, Dsts and DFM can use it.
- the Network: it comprises all those elements which are needed for the movement of event fragments and control data within the Event Builder. It consists of Network Interface Cards, a switching fabric for data transfer and, possibly, a separate message passing system for control messages. While all other logical elements can be designed and implemented independently from the communication technology, the network is, by definition, technology dependent.

The EB has been designed to be physically partitionable, i.e. to allow concurrent disjunct data-taking sessions for different detectors. Furthermore, logical partitions are supported, allowing events to be assigned to different Dsts according to their event type. A single DFM will support several logical partitions, while it is foreseen to have a different DFM for each physical partition.

The logical elements of the EB have been mapped onto physical network nodes in a way which ensures that, independently of the number of nodes in the system, none of the EB applications has to process data and control messages with a rate significantly higher than the EB rate. The EB prototype implementation consists of two set-ups. One is based on RIOII[3-11] VME Single Board Computers, running the LynxOS operating system, and PCs, running Linux; this set-up runs on top of ATM and Fast-Ethernet networks. The other is a 16 node Gigabit Ethernet set-up, based on PCs running Linux.

3.2.4 LDAQ

The LDAQ [3-12] is the interface point to the Back-End DAQ system for: run control, configuration databases and information exchange. In the absence of the Back-End, the LDAQ provides suitable emulation of the Back-End functions, allowing a single subsystem (e.g. a ROC) to be operated. LDAQ has been implemented on VMEbus-based processors running LynxOS and on Linux. The LDAQ communication is implemented on top of VMEbus and Fast Ethernet. The LDAQ control is implemented as a multi-threaded application using POSIX threads.

3.2.5 Summary of the Event Flow

A typical event flows through the DataFlow system, as it has been implemented in the DAQ/EF -1 prototype, according to the following scheme. When an event is accepted by the first trigger level, data are read out and stored in the ROBs which reside in the different ROCs. Here they are made available to the second level trigger and kept until the LVL2 decision is made. If the event is rejected by LVL2, data are discarded from the ROBs, otherwise they are forwarded to the Event Builder Interface (EBIF). If more than one ROB is associated to a single EBIF, ROB-fragments are collected and formatted before being accessed by the Event Builder.

The event ID of a LVL2 accepted event is sent to the Data Flow Manager of the EB which assigns an SFC for the construction of the full event. Event ID and SFC ID are broadcast to the EBIFs, which send the corresponding data fragments via a network to the appropriate Sub Farm

Input. The SFI counts the arrived fragments, builds a full, formatted event, notifies the DFM when the event is completed and forwards the data to the Event Filter farm. If the event passes the EF selection criteria, data are sent, via the Sub-Farm Output (SFO) to mass storage.

Since in DAQ/EF -1 there is no LVL2 trigger, no data are really moved from the ROBs to this system, and the response of the second level trigger is only emulated.

3.3 The Back-End system

The Back-End [3-13] software (see Figure 3-5) assembles all the software related to do with configuring, controlling and monitoring the DAQ but specifically excludes the management, processing or transportation of physics data. It also includes the online-monitoring infrastructure and graphical user interfaces used for control and configuration, and the means for handling distributed information management including database management and tools.

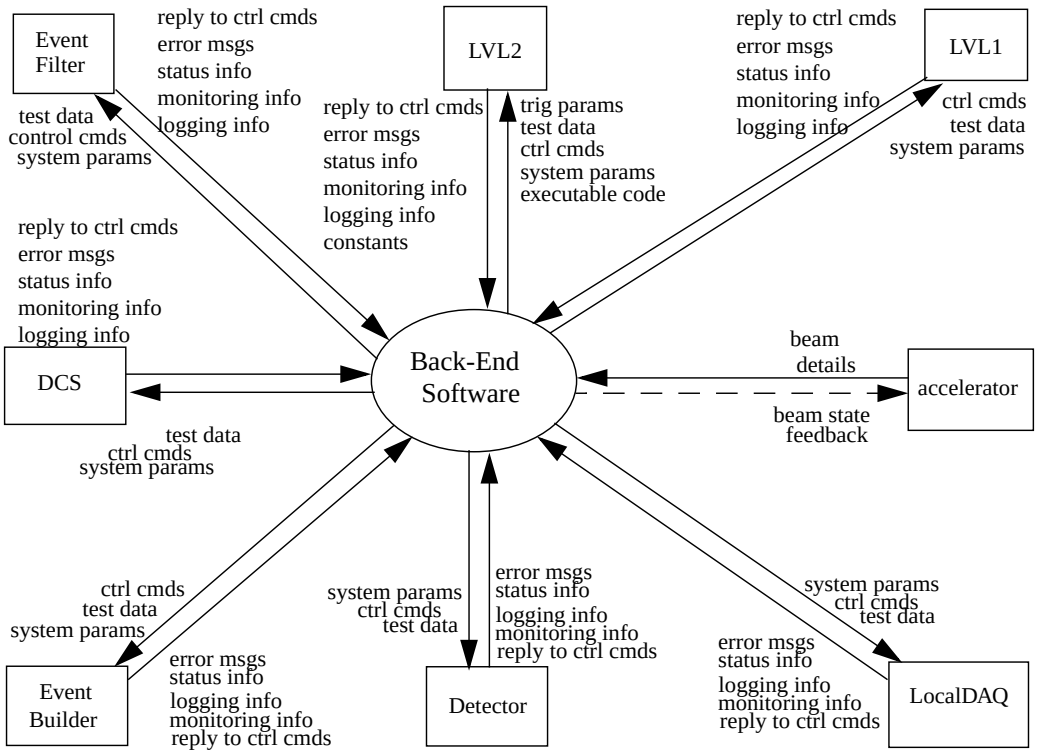

Figure 3-5 Back-End DAQ software context diagram showing exchanges with other sub-systems.

It is expected that this environment will be a heterogeneous collection of workstations, PCs running Windows NT or Linux and embedded systems (known as LDAQ processors) running various flavours of real-time UNIX operating systems (e.g. LynxOS) connected via a Local Area Network (LAN).

The Run-Control component of the Back-End system controls the data-taking activities by coordinating the operations of the DAQ subsystems, Back-End software components and external systems. It has user interfaces for the shift operators to control and supervise the data-taking

session and software interfaces with the DAQ subsystems and other Back-End software components.

The Message Reporting System (MRS) allows all software components in the ATLAS DAQ system and related subsystems to report error messages to other components of the distributed DAQ system. The MRS performs the transport, filtering and routing of messages. It provides a facility for users to define unique error messages which will be used in the application programs.

The basic job control of the DAQ software components is performed by the Process Manager (PMG). It is capable of starting, stopping and monitoring the basic status (e.g. running or exited) of software components on the DAQ workstations and LDAQ processors independent of the underlying operating system.

The Information Service (IS) provides an information exchange facility for software components of the DAQ. Information (defined by the supplier) from many sources can be categorized and made available to requesting applications asynchronously or on demand.

3.4 The Event Filter system

The Event Filter computing instrument will be provided by a processor farm. Data coming via the SFI will be distributed to several processors, working in parallel, each one analysing a full event. In its final configuration the farm must provide enough processing power to build the high-level information which will allow the Event Filter to take the decision of accepting or rejecting the event. Further requirements might come from the necessity of accepting and processing different trigger types, including physics events, calibration and monitoring data. In this context special care must be taken to avoid the creation of bottlenecks due to a lack of resources while retaining the highest possible flexibility. It is recognised that the final performance of this system will be determined by the maximum manageable complexity and the available financial resources. The study of the EF will be the topic of the future chapters of this thesis.

3.5 Global System Integration and Performance

DataFlow

The DataFlow design has allowed different concepts to be tested and technology trends to be followed. The implementation has given a strong indication that the 100 kHz rate required by the ATLAS experiment is within reach. The use of COTS products has decreased the resources required during the design and deployment. Therefore, a ROC based on the DAQ/EF-1 design could be the basis for building a system meeting the ATLAS performance requirements. The same design and implementation also supports the distribution of data to the LVL2 system. In the Event-Builder subsystem, the scalability studies indicate that the protocol scales with the size of the switch. The layered approach taken in the software structuring allows for running the same applications on different technologies.

The LDAQ has proven to be a resource-consuming task in terms of CPU time, memory and networking. This justifies the model of separating it from the DAQ-Unit.

Back-End

The successful completion of almost all the Back-End DAQ packages has proven the validity of the component model adopted in this part of the data-acquisition system. This model has also demonstrated to be particularly suitable for an efficient distribution of the work amongst collaborators who are geographically dispersed. The use of a formal software development process has clearly shown its usefulness in terms of software quality and day-to-day organization of the development work. It has been proved that the design of the Back-End DAQ components can be adopted as a basis for the corresponding components of the final HLT/DAQ system.

Event Filter

The Event Filter architecture is characterized by a clean boundary and interface between the EF and the data-acquisition functionality. The resulting flexibility in the integration of different hardware (PCs and SMPs) and software implementations, of the same designs, of EF SubFarms has supported the concept of independent SubFarms, which naturally solves problems related to Farm partitioning and, more generally, flexibility and scalability. No technical problems were encountered in running EF SubFarms geographically separated from their DAQ data source. The design and the conclusion of the Event Filter system will be addressed in detail in the next Chapter.

In 1999 the DAQ/EF -1 prototype has been fully integrated bringing together the integrated DataFlow system, the integrated Back-End system and the Event Filter system. The external Trigger system is emulated and event data are generated internally by the ROCs as no detector was ready to be integrated. The baseline prototype (the 2×2 *baseline system)* consists of two ROCs, two SFCs, ATM as Event-Building technology and was controlled by the Back-End.

A full vertical slice of the Data Acquisition and Event Filter architecture has been successfully implemented (see Figure 3-6). The prototype has been constructed using existing technologies and has been used to extract information at the functional level. The use of common software technologies for data persistence, communication etc. has reduced the total programming effort, made the developers aware of the structure of all the components and allowed them to share software. The prototype does not have the ambition to completely fulfil the ATLAS requirements in terms of performance. Nevertheless the project has produced an implementation and detailed performance studies have been carried out.

A full DAQ/EF -1 prototype design and implementation has been demonstrated. Measurements carried out on the fully integrated system have shown that, at least for small-scale prototypes, the DAQ/EF -1 implementation meets the required performance. The simulation of the Event-Building system has shown that the DAQ/EF -1 design is scalable and, therefore, the design could meet the requirement of the full ATLAS DAQ system. These results has been the basis for the HLT/DAQ/DCS architectural proposal described in *"ATLAS High-Level Triggers, DAQ and DCS Technical Proposal"* [3-14] published in march 2000. Areas of further possible improvement have been identified concerning both hardware and software.

The DAQ/EF -1 prototype has been selected by ATLAS to be the basis of the DAQ system on the ATLAS test beams, the baseline for the evolution of the ATLAS Data Acquisition and Event Filter, as well as the basis for ROD crate data acquisition. The system is now ready to be integrated with the detector read-out links and exploited as data acquisition system for the ATLAS test beams.

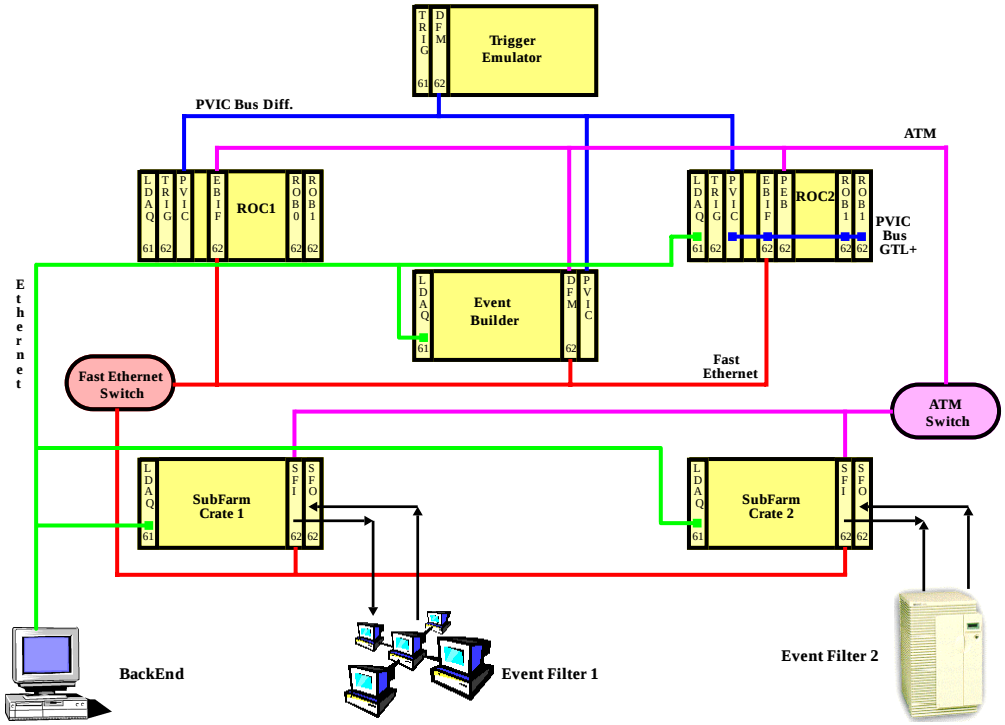

Figure 3-6 The integrated DAQ/EF -1 prototype (the 2×2 baseline system).

3.6 The proposed Trigger and DAQ architecture

The hardware and software studies and the prototypes work done within the DAQ/EF -1 project and the LVL2 Pilot Project was the base for the architectural proposal presented in the *"ATLAS High-Level Trigger, DAQ and DCS Technical Proposal"* [3-14] published in March 2000. This common architectural framework is being used to study and evaluate different implementation alternative for the Technical Design Report, at present foreseen for October 2002.

3.6.1 Overview of System and Subsystem

An overview of the proposed ATLAS Trigger/DAQ architecture is presented in Figure 3-7. The details of the architecture can be found in [3-14]. The global system has been subdivided in four main functional elements: the Data Acquisition (DAQ), the LVL1 Trigger, the High Level Triggers (HLT) and the Detector Control System (DCS). The arrows in the figure indicate the direction of dependence between the different systems.

The DAQ system

The DAQ system comprises the DataFlow and Online Software systems. They collaborate, to meet the data acquisition requirements of normal data-taking and to serve event data to the HLT (more specifically, LVL2 DataFlow serves data to the LVL2 Selection, and EF I/O to the Event Filter). The DataFlow and Online Software systems also collaborate, optionally with the Event Filter (depending on run requirements), to implement the data acquisition and analysis of runs where the LVL2 Trigger element of the HLT is not involved (e.g. cosmic-ray runs and dedicated detector calibration runs). In this case, the LVL2 DataFlow subsystem has no role to play.

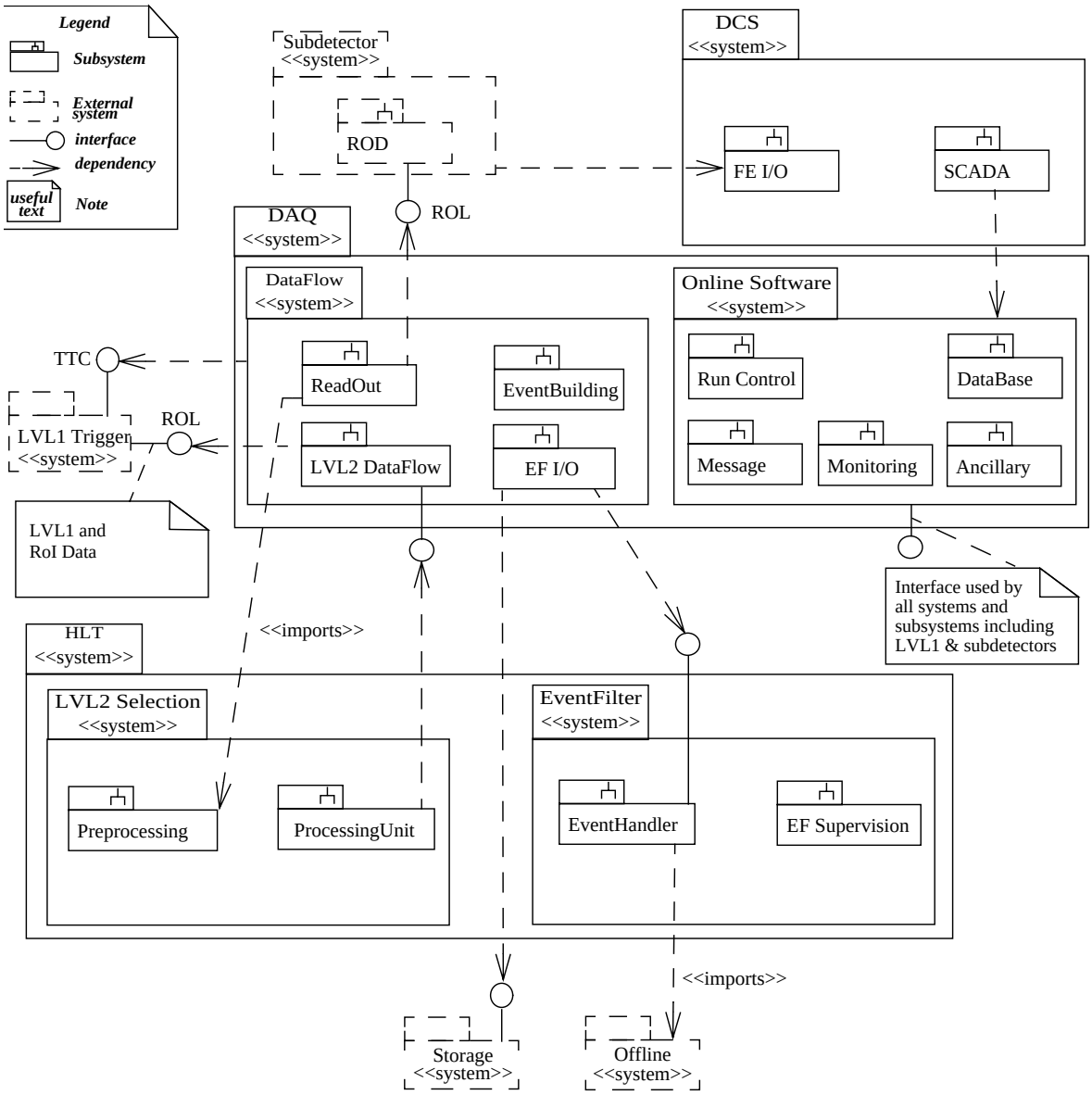

Figure 3-7 Diagram showing the main systems and subsystems of the proposed HLT/DAQ/DCS architecture.

- The DataFlow system provides the functionality of receiving and buffering data from the sub-detector RODs, distribution of events and event fragments to the HLTs and the sending of events to mass storage. It contains four subsystems:
  - The Read Out Subsystem (ROS) provides the receiving and buffering of sub-detector data.
  - The LVL2 DataFlow provides the distribution of the selected event data to the LVL2 Selection system.
  - The Event Building provides the merging of event fragments into complete events

- The EF I/O provides the distribution of events to the Event Filter system and the sending of events to mass storage.

The dependence of the LVL2 DataFlow on the LVL1 Trigger represents the transfer of RoI information from the latter to the former. The dependence of the DataFlow on the LVL1 Trigger represents the triggering of the ROS and Event Building in the absence of a LVL2 Selection (e.g. during a dedicated calibration run).

- The Online Software system is responsible for the configuration and control (in collaboration with the DCS) of the detector and HLT/DAQ/DCS systems. To this end it provides configuration, including the management of detector and DAQ partitions; run control; distributed information management; monitoring infrastructure; graphical user interfaces for the purpose of control and configuration. It contains five subsystems: Run Control, Message, Database, Monitoring and Ancillary. It is implicitly understood to have connections to all sub-detector systems and other HLT/DAQ/DCS systems.

The HLT system

HLT functionality is provided by the LVL2 Selection and Event Filter systems. Data are served to them by the DataFlow system.

- The LVL2 Selection system contains two subsystems:
  - The Processing Unit receives RoI data from LVL1 via the LVL2 DataFlow and steers the processing of each event, requesting and processing selected event data based on the RoI mechanism.
  - The Preprocessing provides the preparation of selected data prior to processing. The Preprocessing functionality is developed as part of the LVL2 Selection system and may be implemented as a component of the ROS, as shown by the dependency of the ROS on the Preprocessing subsystems.
- The Event Filter system provides the final stage of the HLT event selection. It contains two subsystems:
  - The Event Handler subsystem provides the functionality of event selection and other ancillary tasks, i.e. physics monitoring, detector monitoring and calibration analysis.
  - The EF Supervision subsystem provides the configuration, control and operational monitoring of the Event Handler. The Event Handler receives built events from, and returns accepted events to the EF I/O (for transfer to mass storage). It imports as far as possible, filtering algorithms and their implementation from the Offline system.

The DCS system

The DCS is responsible for the coherent and safe operation of the ATLAS detectors and associated systems, and for the interfacing with the LHC accelerator and the services of the CERN infrastructure. It deals principally with non-physics event data which are time-stamped. The DCS has implicit connection to all the sub-detectors, including LVL1, to the elements of the HLT/DAQ which require its services and to the LHC machine. It comprises two component subsystems:

- The SCADA system (Supervisory Controls And Data Acquisition) which: Reads/writes data from/to FE I/O; processes, displays, and archives this data; implements control procedures; interacts with operator; handles commands, messages and alarms; connects to systems external to ATLAS.

- The FE I/O system (Front-End Input/Output) which: reads out and digitizes signals from detector hardware; preprocesses these data; implements low-level control procedures; drives actuators.

3.7 References

3-1 G.Ambrosini et al., *'The ATLAS DAQ and Event Filter prototype '-1' project'*, CHEP97, Berlin (1997)

3-2 ATLAS Collaboration, *'ATLAS Technical Proposal'*, CERN/LHCC/94-43 (1994)

3-3 L.Mapelli, *'Global architecture for the ATLAS DAQ and Trigger'*, ATLAS internal note, ATL–DAQ–95–022 (1995)

3-4 RD13 Collaboration, *'Status Report of a Scalable Data Taking System at a Testbeam for LHC'*, CERN/LHCC/95-47 (1995)

3-5 VMEbus Technology, http://www.vita.com

3-6 LynxOS operating system, http://www.lynuxworks.com

3-7 Linux operating system, http://www.linux.org

3-8 G. Crone et al., *'The DAQ-unit Message Passing API'*, http://atddoc.cern.ch/Atlas/Notes/091/Note091-1.html

3-9 G. Ambrosini et al., *'Event Builder Network I/O Library'*, http://atddoc.cern.ch/Atlas/Notes/067/Note067-1.html

3-10 G.Ambrosini et al., *'Summary Document of the Event Building Studies in the DAQ/EF -1 Project'*, ATLAS internal note, ATL-DAQ-2000-048 (ATL-COM-DAQ-2000-041).

3-11 Creative Electronics Systems, http://www.ces.ch

3-12 *'The LDAQ in the ATLAS DAQ prototype -1'*, ATLAS internal note, ATL-DAQ-98-094

3-13 *'Back-end summary document'*, ATLAS internal note. ATL-DAQ-2000-005 (2000)

3-14 ATLAS Collaboration, *'ATLAS High-level Triggers, DAQ and DCS Technical Proposal'*, CERN/LHCC/2000-17 (2000), ISBN 92-9083-160-0

CHAPTER 4

The Event Filter subsystem in the DAQ/EF -1 project

4.1 Introduction

The EF is an essential part on the ATLAS DAQ chain and it was studied in the common DAQ/EF -1 project. In this chapter the global system architecture and the general characteristic of the EF project is presented. In order to validate the global architecture and to evaluate the requirements of the various subsystem elements three different Event Filter Farm (EFF) prototypes are implemented. The prototypes have been successfully integrated in the global DAQ/EF -1 projects and the results and the studies carried out in their frameworks have been the basis of the EF architecture proposed in the *"ATLAS High-Level Trigger, DAQ and DCS Technical Proposal"* [4-1].

4.1.1 The EF assignments

4.1.1.1 Event Filtering

The EF is situated downstream the Event Builder and will operate on complete event data with access to geometry and final calibration constants. It should provide enough processing power to fully reconstruct the event and execute filtering algorithms which will allow to reduce the data flow by a factor of the order of 10. This rate reduction from 1 GB/s down to the nominal maximum rate for data storage of a few hundred MB/s can be achieved either by reducing the accepted event rate or size. For some triggers, the full event data will be recorded, whereas for others it may be sufficient to send only a selected subset of the event data to storage. In any case, the EF farm should provide sufficient computing power to achieve the above nominal rate.

The EF will be highly CPU-bound with data I/O representing a small fraction of the overall latency. It is very difficult to estimate now the total computing power needed to achieve these goals. Extrapolations, based on reconstruction time of the old CDF/DØ events and of ATLAS Geant 3 [4-2] based Monte Carlo events provide an estimate of approximately 25 SPECint95-sec per event. This brings the total amount of CPU required by the event filter process to 2.5×10^4 SPECint95. However, those numbers should be carefully considered and viewed as only a minimal estimation. Furthermore the total CPU power, and so the EF tasks and performances, could be limited by funding.

Contrary to the LVL2 trigger, no special algorithm will be developed for the EF. Algorithms will be inherited as much as possible from the offline ones. This has strong implications on the design of the reconstruction software, because it implies a funnel shape[1] of the selection sequence, to take advantage of an early, fast rejection of unwanted events. In this spirit, since the policy of inheriting algorithms stems from the desire to simplify the transition between the two worlds, avoiding parallel development and allowing direct comparison of physics objects, the EF code should be considered as the core of the reconstruction software. The possibility of moving algorithms across the different levels (included the offline stage) should also be considered a desirable goal, depending on the availability of hardware resources and the understanding of the detector and of the selection scheme.

Moreover it is important to ensure the quality of the software used in the online environment. The potential impact on the physics performance resulting from errors or inefficiencies in the selection code is of course tremendous. Events that are wrongly rejected are lost forever and code not robust enough is bound to crash often, reducing the lifetime of the online data acquisition and hence the luminosity integrated by the experiment. Therefore, particular attention should be paid to the testing and validation procedures of the software which will run in the online farms in order to ensure the highest possible robustness.

4.1.1.2 Event Tagging

The EF will be the first system to see complete ATLAS events. Furthermore, it will analyse every event sent to permanent storage, independent of the event classification. Thus, it can provide a means of tagging events, facilitating the early discovery of new physics. Some of the new physics channels are expected to have very high event rates. The EF must be able to:

- Tag individual events (e.g. large γ–γ mass, events with four leptons, large missing-E_T) for immediate analysis during shift;
- Send tagged events to a storage medium for subsequent express physics analysis,
- Store histograms of critical variables (e.g. mass plots, background ditributions) and perform automated searches for significant deviations from smooth behaviour;
- Report the tagging and storage of events and the deviations of critical parameters from smooth behaviour.

The storage of specific event types for express physics analysis will be based on the event classification performed by the high-level trigger decisions.

1. i.e.: the sequence is arranged to minimize the computational load in the initial selection steps, with the computationally heavy parts treated as late as possible, while retaining good signal efficiency and background reduction.

The details of the event classification schemes and of the physics selection sequences was not studied and developed within the DAQ/EF -1 project. These topics, fundamental for the implementation of the final ATLAS data acquisition system, concern the PESA group and will be described in Chapter 7.

4.1.1.3 Physics and Detector Monitoring

In order to have a complete online diagnosis of the data taking and detector behaviour, which is necessary for the efficient running of the overall detector, physics and global detector monitoring will also be performed within the EF environment. The availability of fully reconstructed events permit to evaluate the performance of the sub-detectors, including the reliability of the selection performed by the LVL1 and LVL2 trigger systems. The analysis of correlations between information coming from different parts of the detector opens up detector monitoring capabilities which are not available at the front-end level. In addition, the EF algorithms themselves will continuously monitor their own performance in order to provide continuous quality control. Further physics monitoring will also be required at the EF stage in order to control the relative production rates and quality of the acquired physics channels.

Monitoring tasks should run without affecting the event filtering performance itself, though a certain minimum guaranteed monitoring sampling rate should be foreseen

The definition of detector monitoring or calibration procedures is responsibility of the various detector groups in the Collaboration, although it is clear that some members of the EF group, also engaged in physics groups, will be heavily involved in implementing the analysis algorithms for the EF. The EF should be designed according to the requirements of the experiment (both from the point of view if its physics goals and the various detector needs) in this area. The ability to run monitoring procedures, together with a means to access the data flow for such procedures must be provided.

4.1.1.4 Calibration and Alignment

The quality of the EF decision relies strongly on the quality of the sub-detector calibration and alignment parameters. Often, specific calibration triggers are required (pulser, minimum bias, cosmic rays, etc.). In general, these parameters need to be calculated more or less continuously, using assembled data from the detector. The only logical level in the TDAQ architecture where one can perform these calculations is that of the EF. Otherwise, they must be done either entirely offline (may well not be appropriate) or on some separate engine from the EF (creating duplication of effort). If statistics can be gathered on a long-term basis, one could envisage the full execution of the calibration and alignment tasks directly in the EF farm. It would also be possible to analyse directly the special events generated by the detectors for calibration and alignment purposes, if these events are transferred to the EF. Procedures to determine calibration parameters will be very similar to the monitoring tasks mentioned above, and similar constraints are appropriate.

A few examples of physics events which will be of use for calibration and alignment tasks are given below:

- inclusive single electrons with $p_T > 20$ GeV, $Z^0 \rightarrow e^+ e^-$ and $W^{\pm} \rightarrow e\nu$ decays for the inner detector and the electromagnetic calorimeters;

- single muons with $p_T > 6$ GeV, $Z^0 \rightarrow \mu^+ \mu^-$ and $W^{\pm} \rightarrow \mu\nu$ decays for the inner detector and the muon chambers;
- γ + jet, Z^0 + jet events, isolated charged hadrons from $W^{\pm} \rightarrow \tau\nu$ decays for the hadron calorimeters;
- $W^{\pm} \rightarrow e\nu$ and $W^{\pm} \rightarrow \mu\nu$ decays to study the E_T^{miss} scale and resolution;
- minimum-bias events to measure the beam position (transverse and longitudinal).

4.2 EF Architecture high level design

4.2.1 Global view

The high-level design of the EF addresses questions of overall supervision, control, and the internal flow of data. It also addresses the issues of interfacing the flow of data within and between the processors of the EF with the overall data flow.

The aim is to produce a generic design for a sub-farm unit which is totally independent of implementation and architectural details, with a view to being able to use the same design for the different physical prototypes which are constructed within the context of DAQ/EF-1 project. Scalability must be a strong requirement since it will allow the progression of the system from the size of the initial prototype to that of the final system.

Because the building of the event is performed by a switch where the different sources are connected to independent destinations, it is natural to factorize the EFF into independent sub-farms, each of them connected to a different output port of the EB as well as to the mass storage (Figure 4-1). Given the estimated running lifetime of more than 15 years, the EFF must be both easily and gradually upgradable. The global organization of the EF into several sub-farms, the number of which could reach the order of one hundred in the final implementation, makes the design scalable in terms of reaching the required computing power. The design of the EF sub-farms will be as modular as possible, so that any change in this design can be implemented in a smooth and transparent way.

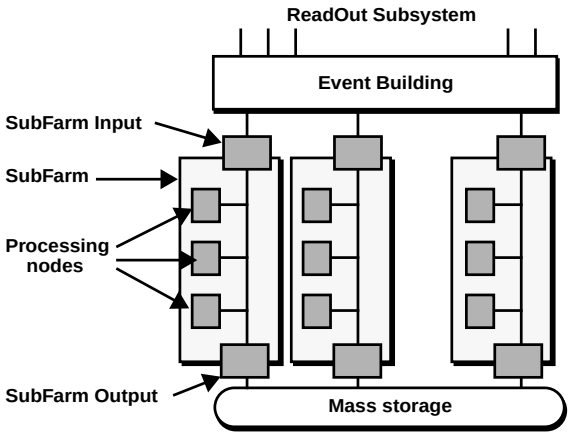

Figure 4-1 Overview of the EFF architecture.

Every sub-farm contains one or several processors linked together. The design will be independent of the technology of the processors as well as of the communication protocol between these processors (bus, crossbar switch, local or wide area network, etc.). The required number of sub-farms will be determined from the available technology and the total CPU power needed.

The DAQ system and the Event Filter are interfaced based on the following assumptions:

- the Event Builder receives fragments from the ReadOut Buffers and assembles them into fully built events;

- the events are passed to the Event Filter through multiple independent ports;
- after processing, the events are collected by the data flow to be sent to permanent storage.

No assumption is made on the technology which is used for both the event building (switch, shared memory, etc.) and the mass storage (single channel, multi channel, database, etc.).

4.2.2 Sub-farm architecture

The EF sub-farm architecture has been factorized into several components in order to aid parallel development and to identify the functional composition of the sub-farm (see Figure 4-2). A sub-farm consists of a sub-farm DAQ and an associated Event Handler (EH).

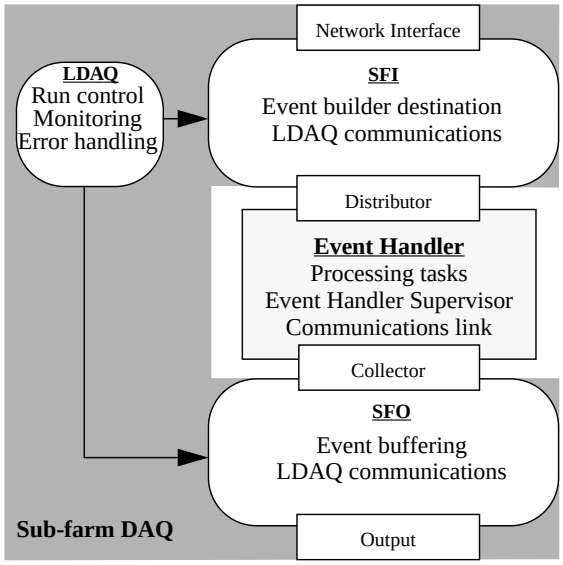

Figure 4-2 Subfarm architecture.

The latter is the component of the EF sub-farm, which implement the filtering capabilities as well as other processing functionalities (such as monitoring, calibration, etc.).

The sub-farm DAQ unit (SFD) is responsible for channelling all event data coming from the event builder through the EH. Events transformed by the EH system are then passed by the SFD to the mass storage system. The SFD as well as the LDAQ, which provides the means to control the flow of data in the SFD and makes the interface with the Back-End software, are part of the DataFlow system [Section 3.2].

On the data flow side, the interfaces with the EH are provided by the Sub Farm Input (SFI) and Sub Farm Output (SFO) which are instances of a data flow I/O module. The SFI receives fully assembled events from an output channel of the Event Builder, performs final event formatting and makes the events available to the Event Handler via the distributor interface. Accepted events are passed via the collector interface to the SFO from where they are sent to data storage. These interfaces between SFD and EH is defined by two Application Programmers Interfaces (APIs) and allows a task within the SFI to send events to the Distributor, or a task within the SFO to receive events from the Collector.

The particular function of the distributor and collector is to insulate the internal functions and architecture of the data flow in the SFI and SFO from those of the event handler. The latter will therefore appear as a black box to the former. The distributor will provide the functionality for the SFI to inject events into the EH, and the collector will provide access to selected events to the SFO for further disposal. This approach permits parallel development of components of the data flow and EF, enables removal and insertion of different sub-farm prototypes into the overall DAQ/EF -1 architecture and also establishes a clear boundary of responsibility.

Because of the architecture choice of the Main Data Flow, events are pushed from the SFI to the EH and the processed events are pulled from the Collector by the SFO. The flow of events through the EH is then purely data driven. The regulation of the flow of events is naturally done by back pressure based on availability of resources. Load balancing across the sub-farms,

system partitioning and the detailed strategy of event distribution to the SFIs, and consequently to the sub-farms, is managed by the DataFlow system. It may well be desirable, for example, to direct certain specific event types (such as dedicated detector monitoring events) to a particular SFI so that they would subsequently be processed by a dedicated sub-farm unit of the EF.

4.2.3 Event Handler High Level Design

The component high-level design, presented in the technical DAQ/EF-1 note 61 [4-3], was been kept as general as possible to allow different models of the data movement within the EH and to allow changes in the model to be incorporated as the high-level designs of other element mature.

The EH consists of the distributor, the collector, an EH supervisor and one or more Processing Tasks (PTs). These objects as well as the operations between them are shown in Figure 4-3. Minimization of the synchronization and coupling between the event flow and the supervisor has been one of our first concerns in designing the EH. The flow of the events through the EH is purely data driven and does not rely on transactions with the Supervisor.

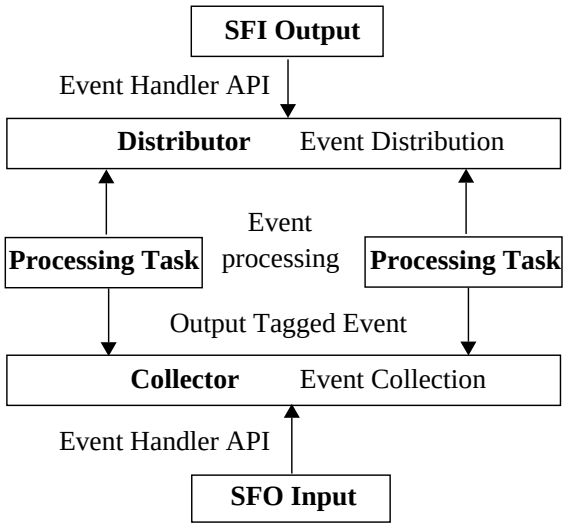

Figure 4-3 EH context diagram.

Distributor

The Distributor receives events from the Sub-Farm DAQ by implementing the Event Handler API, and distributes them according to event type to the processing tasks within its Event Handler. The treatment of events according to event type is an option which is made available in the design. It may well prove desirable to have the ability to filter events of differing type (e.g. physics, calibration, monitoring) to processing tasks which have differing functionality. However, it is not mandatory to use this feature, and by allowing only one single event type, all incoming events will be treated equally by the Distributor.

The Distributor maintains a copy of the event during processing by a processing task. In addition, events are immediately buffered in a global (to the sub-farm) data store (Distributor Global Buffer - DGB) on reception from the SFI. This buffer is associated to a permanent storage device so that data security is ensured all along the lifetime of the event inside the sub-farm. When the event has been either accepted and successfully transferred to the downstream component, or rejected, it is removed from the DGB. On the occurrence of a problem while being processed, given the fact that the event is still in the DGB, a recovery procedure can be launched according to the origin of the problem.

Collector

The Collector “receives” events from one or more processing tasks and stores a safe copy of them in the Collector Global Buffer (CGB), which plays a role similar to the one played by the DGB in the Distributor. The two buffers have been separated to handle in a simpler way any extra information which may have been added by the processing task. By implementing the Event

Handler API, it allows the SFO to receive events from the EH, and on reception of the event by the SFO, the Collector then removes that event from the CGB.

Processing tasks

Processing task is the generic name for the task which processes an event in the event filter. Events are not limited to the strict definition of the assembly of fragments coming from the front-end crates (physical events) but it is foreseen the possibility of treating the result of data not coming from a beam-crossing interaction. Such non physical events, could be the result of laser shots or any action in view of calibration or detector integrity check. Physical and non-physical events will be distinguished by the event type, located in the event header. The primary instance of the Processing Task is event filtering; at the end of this procedure, the event is tagged for passing to the SFO and mass storage if accepted, or it is rejected. The processing task will be a stateless object which does not receive commands from the run control. As soon as it has been launched and initialized, it waits for events to process. It is completely data driven. Clearly, the failure of a single processing task shall not have any repercussions on the system functionality.

EH Supervisor

Each Event Handler is a separate control domain which can be independently supervised and controlled from the DAQ Run Control. The interface between the EH and the outside control world is the EH Supervisor (one per EH). EF supervision must be decoupled from the global DAQ system so that the most adequate technologies can be used for the different prototypes without being affected by upstream choices. The integration of the EF supervision with the global Run Control will be performed at a later stage. The Event Handler Supervisor, whose high level design is described in technical DAQ/EF-1 note 92 [4-4], should provide the following functions:

- run control: it provides the interface between the EH and the overall Run Control system. A local stand-alone control system must allow some local direct actions on the EH without affecting the functioning of other parts of the Event Filter (e.g. elements of the sub-farm DAQ: SFI, SFO).
- process management: it must be able to launch and stop the different EH tasks. It should also be able to detect process crashes and take appropriate action. A failure of a supervisor task does not affect the filtering task. The latter continues processing while the supervisor is restarted. A crash of a processing task is however a normal event for the supervisor, which is designed to handle it. Processing tasks are not aware of the supervisors existence.
- synchronized access to databases: it is responsible for providing the processing tasks with the path to up-todate databases for calibration and alignment. It should also inform the other EH components of any conditions and general purpose variables (e.g. the run number) which drive their behaviour, as well as accessing the local EH configuration database for process management purposes.
- monitoring: information on the EH behaviour must be collected and published. Errors should be reported to the BackEnd error report facility (MRS). It should also be able to collect and publish information provided by the processing tasks.

These functions are highly correlated. In particular, error detection will in most cases trigger an action of the run control and/or the process management. It will be an issue of the prototypes to understand and further define these correlations.

For practical reasons, the Supervisor will be initially studied separately from the Sub-Farm LDAQ (which is responsible for the control and supervision of the SFI and SFO DataFlow elements) [Section 3.2.4], since it must provide functions which cannot be easily performed by the LDAQ, such as monitoring inside the EH.

The Supervisor functions are performed via several types of devices (see Figure 4-4):

- a command user interface used to receive commands from an operator in local mode
- a set of specific functions to handle process management
- a set of monitoring displays to present the EH states and variable values to users
- an external dynamic database accessed:
    - to publish EH states and variables which have to be made available to the external world
    - to read external DAQ system states and variables which may be of concern to the EH activities.
- a message system to report asynchronously to the end users.

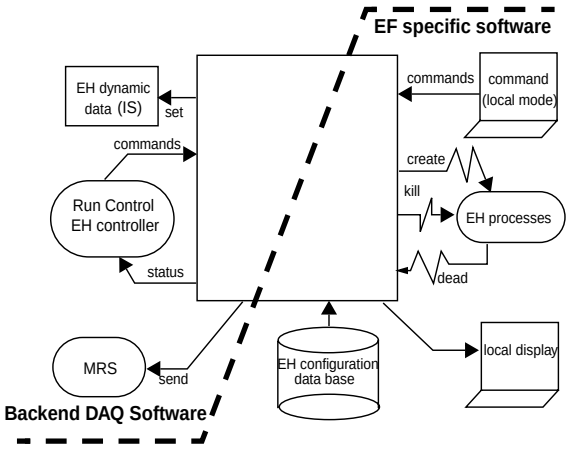

Figure 4-4 The EH run control context diagram.

Those last two items can be fulfilled by the BackEnd DAQ packages IS and MRS [Section 3.3].

4.2.4 Error handling and recovery

The need to minimize the loss of data and guarantee data integrity at every step of the processing is obvious. Opportunities for failure are numerous, considering the size of the processing farm and the complexity of the software. Failure may occur at the input or output of the EF or while moving or processing the event within the farm. It is not easy to be exhaustive on the possible causes and it will be an issue of the DAQ/EF-1 to help to identify, understand and propose correction procedures for possible defects. One may already identify the following global categories of failure:

- hardware failure at all levels (data I/O, processing),
- run-time error while processing the event,
- data-integrity loss while moving the data

A real-time backup of all events on input to and output from the EF is necessary in order to be able to recover events in the case of failure during processing or output. The DGB and CGB safe storage areas, described above, are used for temporary storage until the event has been safely disposed of; they prevent event loss in case of hardware or software fault.

Efficient strategy will be provided to recover from crashes in a safe and transparent manner. They will in particular provide the means to perform some consistency checks so that both data

and run integrity are ensured. It is of major importance to evaluate as precisely as possible the normal fault rate so that the EF can be oversized and provide the required processing power at a minimal cost

Most run time errors will be caused by unforeseen event topologies. The chance that simple re-processing solves the error is very small. However, such events must clearly not be rejected: this should lead to systematic biases in the recorded physical sample. Therefore they could be direct to a dedicated storage channel for further detailed analysis. Error messages and logging will of course complement the debugging information available on such failures. A detailed study of debugging techniques and procedures has be undertaken within the bounds of the EF prototypes (see below).

System errors are more likely to happen during the data transfers or in the control procedures. Reprocessing of the event will be the action undertaken when they occur. Events will be buffered during their progress through the event filter pipeline and kept until they have been safely transferred to the output (SFO) stage.

Considering the mean time between failures (MTBF ~ 17000 hours for currently available PC processors) of the hardware, and the huge number of processors and devices in the farm, more than one hardware error is likely to happen every day. In the event of a failure, recovery procedures will be used to reprocess the event on another processing task. Replacement of faulty material should affect as little as possible the overall activity of the sub-farm, making the capability of live insertion of components very desirable.

4.3 Prototypes

There were several possible candidate hardware architectures for an event filter computing farm (EFF) capable of treating an input data rate of about 1 GB/s (1 kHz) and effecting the required rate reduction (a factor of the order of 10) by applying complex physics constraints to the events. The required processing power was estimated to be at least 25 kSPECint95. Any final hardware choice must clearly be left until the latest possible moment compatible with building and commissioning the EFF in time. It was also noted that given the estimated running lifetime of more than 15 years, the farm must be both easily and gradually upgradable. Furthermore this indicates the possibility that a variety of hardware and even operating systems may be running in parallel during the lifetime of the EFF.

In order to progress in the understanding of the problems inherent in implementing the EF in the ATLAS DAQ architecture, it was decided to build several prototypes. According to the high level design previously outlined, three EFF prototypes have been implemented and assembled based on three somewhat different configuration: Commodity PCs, Intel Commodity Multi-Processors and Commercial Symmetric Multi-Processor (SMP). The first two prototypes are based on network interconnected PC processors and entail contrasting distributed architecture, while the SMP solution permits the implementation of an entire sub-farm within a single machine.

Each prototype will be implemented as a single sub-farm with a view to integrating it into the DAQ/EF-1 architecture. This will enable the comparison of the various prototypes in a homogeneous and realistic environment. Issues of physics algorithm performance, system management and control, reliability, error handling, and scalability are among the many issues which will be investigated. Of particular interest to the SMP-PC farm comparison will be an es-

timate of the primary investment and installation cost of an SMP farm vs. the cost of the more complex system management and maintenance of a PC farm. The various prototypes will also be the subject of system modelling studies in order to try to extend the knowledge gained from the prototypes to larger scale implementations of the same architectures.

The distributed architecture prototypes are presented in this chapter, while the SMP one, which is a major subject of these thesis, is described in detail in Chapter 5. The results of the different prototypes, the main conclusions of the DAQ/EF-1 project in the EF system and the present development works are the subject of the Chapter 6.

4.3.1 Commodity PC prototype

This prototype [4-6] aims to exploit the ever decreasing price to performance ratio of the commodity component computers. Communication is not expected to be a bottleneck in an Event Filter sub-farm since applications running will be CPU bound. The distributed nature of the Event Filter complies rather well with the hardware architecture of farms of PCs, facilitating scalability and hardware upgrades. However, management of very large distributed configurations will be a complex task and will necessitate specific developments (provided by the operating system itself in the case of SMP machines).

The PC prototype study involve the *Institut de Fisica d'Altes Energies* (IFAE) of the Barcellona University, the CERN IT division and the *Centre de Physique des Particules de Marseille* (CPPM). It is based on bi-processor Pentium II machines interconnected with Fast Ethernet and with a Gigabit Ethernet up-link to an external workstation used to inject events in the Farm. The adopted strategy consists of developing an implementation of the chosen design on a small scale prototype (4 bi-processor machines) located in Marseille and performing then the scalability tests at CERN on dedicated large configurations. The first test-bed comprises 15 machines of the PCSF cluster [4-5] located in the CERN Computing Centre, while the other is a farm of 25 Dual-PC entirely dedicated to Event Filter tests for the LHC experiment.

4.3.1.1 The data flow implementation

The software architecture is based on distributed computing techniques. The various sub-farm components are implemented by different processes running on different nodes. In the first prototype version, the communication among sub-farm object was based on the ILU [4-8] implementation of the CORBA[1] architecture [4-7], which allows to invoke remote object methods in the same way as local object methods making its usage transparent for the programmer.

This transport protocol turns out to be very complex to manage and not completely reliable (ILU does not provide any means for a Server Object to catch a return failure), therefore a com-

1. CORBA (Common Object Request Broker Architecture) [4-7] allows to invoke remote object methods in the same way as local object methods making its usage transparent for the programmer. In C++ it is implemented by use of a local object (Proxy Object) representing the remote object. Those Proxy Objects have the same methods as the remote object they represent but their method implementation handles the transfer of parameters and results with the remote object. On the remote computer, a server, named Object Request Broker (ORB), will dispatch invocations to the appropriate object instance, the remote object. ILU [4-8] is one of the many implementation of CORBA. It is developed at Xeroxs Palo Alto Research Centre and made available as freeware.

munication layer based on native and standard TCP has been implemented. Events are transferred as an opaque byte sequence through a TCP connection. The event transfer is “on demand”: events are only sent if a request is received.

The prototype has been firstly implemented using Windows NT as operating system and then ported on Linux, because the former was no longer supported by the DAQ / EF -1 project.

The prototype has been used as a test bed for the design and development of the common EF data-flow code, as described in Section 6.4

4.3.1.2 The supervision implementation

The supervisor of the prototype makes large use of the capabilities of the Java mobile agent technique. The Java Virtual Machine and Javas class loading model, coupled with several of the Java features (serialization, remote method invocation, multi-threading) have made building mobile agent systems a fairly simple task [4-9]. Java Mobile Agent systems have a number of key characteristics:

- all Java Mobile Agent systems provide an agent server, which is a contact point on a given machine (Figure 4-5). Those server objects act as warehouses, or workplaces into which agents move and in which agents act. A server provides a means of hosting and managing its own agent in an environment that is secure from malicious agents.

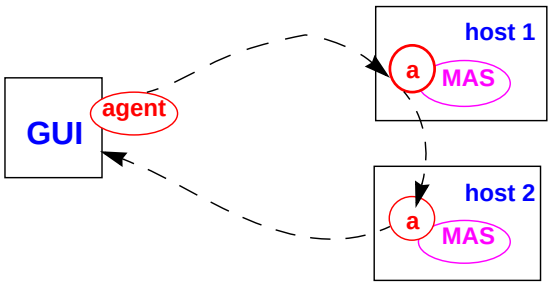

Figure 4-5 Use of Mobile Agent Servers by Java Mobile Agent.

- agents can migrate from server to server, carrying their state with them. After having moved into a server, an agent becomes a local user, and it can do everything that a local user can do, such as getting system resource information (e.g. process, disk, memory, cpu, network usages), create / delete processes, etc.
- agents can load their code from a variety of sources. In general, since all the agent systems use a specialized version of the Java class-loader, they can load Java class files from the local file system, the Web, and ftp servers.
- they are 100% pure Java. This means that they should run on any computer with a compatible Java run-time library.

The chosen Java Mobile Agent implementation is *ObjectSpace Voyager Core Technology* [4-10]. The implementation scheme is shown in Figure 4-6. There are 3 levels of monitoring: system supervision (at the machine level, possibly provided by the operating system), Mobile Agent for event data flow control and monitoring, offline analyse tools for monitoring data archived in a database. Mobile agents collect information from the various supervised components. The information can be directly exploited by the user interface or can be stored in a database for subsequent timewise analysis. Figure 4-7 illustrates how commands and status requests can be sent from the Control Interface to remote nodes by mobile agents and how status data is collected and returned to any monitoring interface. We have developed a unique interface for control and monitoring which can be used either in stand-alone mode or over the Web. Several copies of the

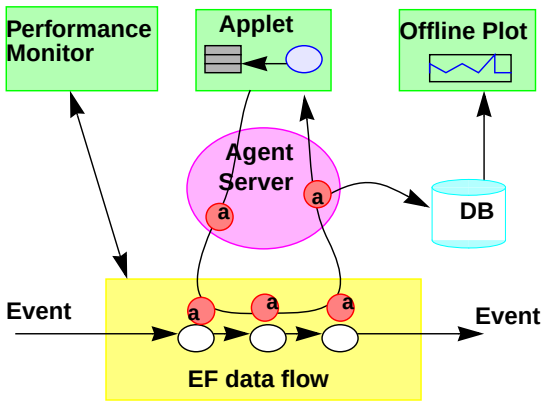

Figure 4-6 Implementation scheme of the supervisor.

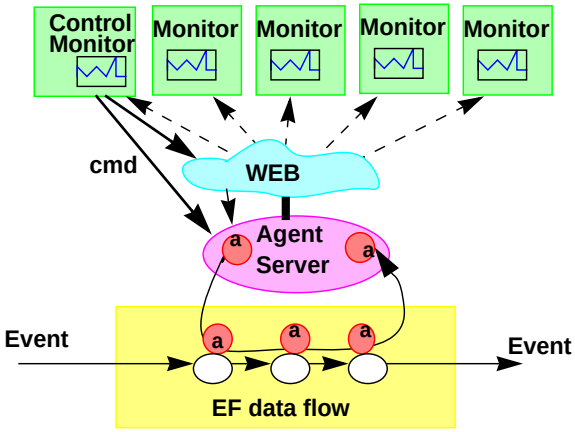

Figure 4-7 Control scheme.

interface run at the same time, but only one is allowed to perform control tasks. The agent server notifies all running interfaces when the EF status changes.

The GUI can run in application mode or applet mode. A password-based access control is used for monitoring / control switching, ensuring that a unique GUI for control is enabled. Different viewers allow to display the entire EF hardware configuration and the event data flow structure. Control of the use of the archiving database is provided: user can switch on / off DB, change the rate of collection. An embedded mini browser allows to select configuration file over the web. It is possible to add / delete process at run-time. User can see agent travelling status via agent itinerary window. Online histograms and plotting packages are provided.

4.3.2 Intel Commodity Multiprocessor Prototype

The main goal of the THOR project [4-11] at the University of Alberta is to provide a commodity component prototype for the EFF that may be used to study various algorithms and implementations of the data flow and analysis. THOR aims to provide a 1 / 10 size (128 processor) prototype of this farm in its final version which is planned for the end of 2000. Figure 4-8 shows the THOR configuration at June 2000.

The working goals of the THOR prototype are as follows:

- Use commodity components and open source software as much as possible. It is hoped that open source software such ad Linux and gnu compilers will allow EF software to quickly take advantages in commodity components hardware as well as to reduce the costs associated with software production for the system.
- Evaluate the use of PC platforms in terms of reliability and fault tolerance for the EF application
- Evaluate both distributed and shared memory designs for the EF. The shared memory testing will take place using SMP and Scalable Coherent Interconnect (SCI).
- Provide a useful hardware prototype for the next EF developments and tests. The prototype is being moved to CERN, where it will became a common tool for the EF community.

The prototype configuration used in the tests consists of 20 dual PII / PIII 450 MHz SMP machines connected with fast ethernet. Each machine has 256 MB of RAM and a 4 GB IDE hard

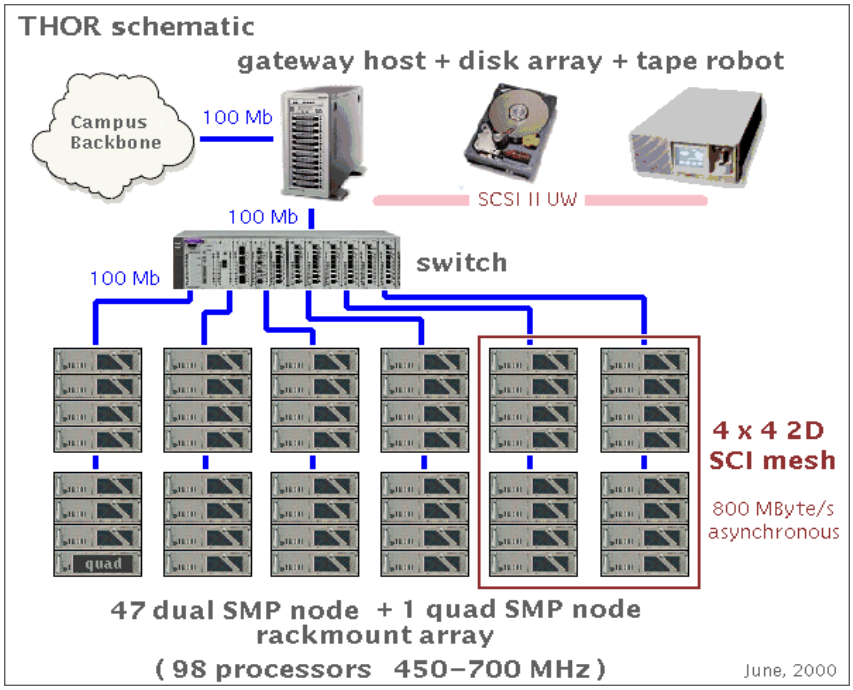

Figure 4-8 The THOR prototype at June 2000.

disk. In addition, nine of the nodes are also connected using SCI. The THOR cluster uses a mixture of RedHat Linux 5.2 and 6.0 as the operating system with kernel version 2.2.2.

The design is based on a cluster of independent processors and multiple sub-farm can be implemented inside the prototype. Many processes are running on the different machines in each sub-farm. The interprocess communication transport is based on raw TCP socket, because TCP/IP is a well-established, well-tested protocol and it offers excellent guarantees of reliability. Moreover, some specific features have been implemented on top of TCP/IP to improve message routing between the different components.

4.4 References

4-1 ATLAS Collaboration,*'ATLAS High-level Triggers, DAQ and DCS Technical Proposal'*, CERN/LHCC/2000-17 (2000), ISBN 92-9083-160-0

4-2 R. Brun et al., *'GEANT3'*, CERN/DD/EE/84-1 (1996).

4-3 C.P. Bee et al., *' High Level Design of the Sub-Farm Event Handler'*, http://atddoc.cern.ch/Atlas/Notes/061/Note061-1.html

4-4 C.P. Bee et al., *'Event Handler Supervisor High Level Design'*, http://atddoc.cern.ch/Atlas/Notes/092/Note092-1.html

4-5 F. Hemmer et al., *'PCSF - A PC based Simulation Facility running Windows NT'*, CHEP 98 Conference, Chicago, IL (USA), 31 August - 4 September 1998

4-6 C.P.Bee et al., *'Proposal for an Event Filter prototype based on PC computers in the DAQ-1 context'*, http://atddoc.cern.ch/Atlas/Notes/045/Note045-1.html

4-7 *Object Management Group*, http://www.omg.org/

4-8 *Inter Language Unification*, ftp://ftp.parc.xerox.com/pub/ilu/ilu.html

4-9 Joseph Kiniry, Daniel Zimmerman, '*A hands-on look at Java mobile agents*' , IEEE Internet Computing, Vol.1, No. 4, July-august 1997, p. 23-30, http://computer.org/internet

4-10 *Voyager*, http://www.objectspace.com

4-11 The THOR project, http://thor-gw.phys.ualberta.ca

CHAPTER 5

SMP prototype

5.1 Introduction

This chapter presents the implementation of the EF sub-farm High Level Design, outlined in the previous chapter, on an Symmetric Multi-Processor machine with a software solution specially designed to better exploit the hardware resources.

Since the processing power of each sub-farm is supplied by several processors working in parallel, a complete sub-farm can be implemented on a single SMP machine. The SMP architecture offers advantages in data sharing and transfer between the different hardware and software components. Every processor has its own memory cache; the main memory and many other resources can be accessed symmetrically by all the processors through a very high speed system interconnect (system bus, crossbar switch, etc.).

An Event Filter Prototype based on a SMP architecture has been proposed in March 1998 [5-3], developed during 1998 and integrated in the DAQ/EF-1 project in the middle of 1999. The details of the implementation and the measured sub-farm performance are reported in Ref. [5-4] and [5-6].

5.2 Multiprocessor Architectures

5.2.1 Symmetric Multiprocessors

Symmetric multiprocessor (SMP) systems are basically tightly coupled shared memory systems. As depicted in Figure 5-1, typical SMP systems have processors, memory modules and

I/O devices connected to a common high performance bus system. They are symmetric in the sense that processors and I/O devices have equal access to memory. Additionally, any processor can access independently the I/O devices. There are no privileged processors on the system for any operations, like access to the I/O sub-system or execution of the operating system code.

SMP systems are a recent and a rather successful addition to the high-performance computer marketplace. There are many reasons for that, amongst them:

- Utilization of traditional uni-processor components
- Small scale systems with very high computing capacity
- Possibility to right-size the system to the application, by adding more processors and memory modules.

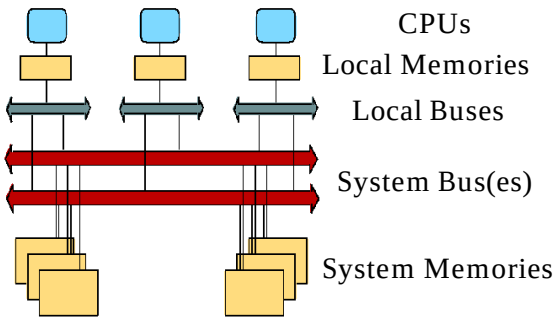

Figure 5-1 General SMP architecture.

One important parameter of an SMP system is scalability, i.e., the ability to increase the total performance of the system by adding more processors. Every system has an upper limit of scalability (the scaling factor never reach the number of processor), which usually depends on the characteristics of the interconnection (bandwidth, arbitration policy, latency, etc.), the memory bandwidth, the I/O bandwidth, and the applications running on the system. This limitation is described by the Amdahl's Law [5-1], which relates processors, parallelism and synchronization. More sophisticated systems, come with multiple buses and few of them, with crossbar interconnects in order to increase scalability.

In a SMP system different applications can run on different processors, without interference between them. However, applications may share the same memory. Different synchronization techniques are offered by the operating systems, to ensure data integrity. At the hardware level though, if more than one processor reference the same memory location, then arbitration will be done at the bus level to serialize the memory request. In conjunction with the system s memory model, the arbitration techniques could be quite complex.

On SMP systems, applications do not necessarily run faster. Each processor may execute different applications, hence an SMP will offer at least more computing capacity. However, it is possible for specially built applications to run different thread of the process on more than one processor simultaneously, resulting in lower total execution time.

5.3 Software architecture

In order to achieve a better exploitation of the SMP architecture we have chosen to base the software architecture on multi-threading programming technique [5-2].

Multi-threading is a technique that allows one program to execute multiple task concurrently dividing it into multiple threads: i.e. different stream of control that can execute its instructions independently. Therefore on a multi-processor machine many different threads can run in parallel and concurrently on different processors; it is permitted the overlap on input, output and other operations with computational ones. Contrary to the process, a thread is a light-weight user level entity, comprising the registers, stack and some other data. The rest of

the process structure (address space, file descriptors, etc) is shared by all the process threads. Much (and sometimes all) the thread structure is in the user-space, allowing for very fast access. The actual code is global and can be executed on any thread.

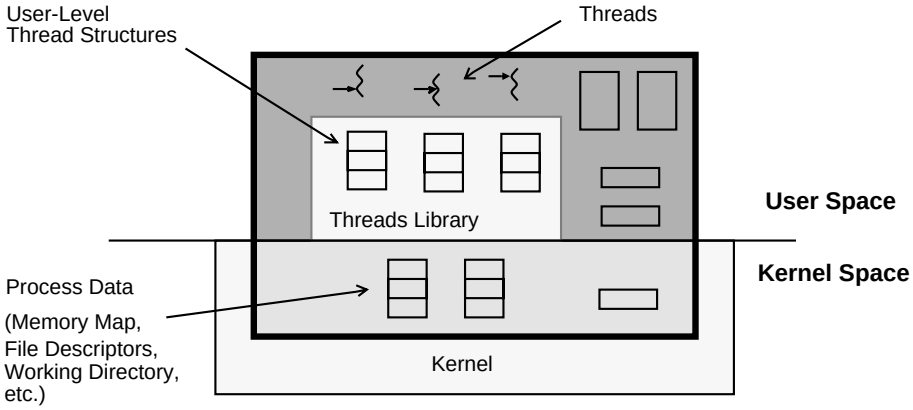

Figure 5-2 The process and thread structures (From [5-2]).

On an SMP machine the parallelism and the concurrence needed for the multi-task sub-farm work, can be realised in a similar fashion with either a multi-process program (several processes in communication) or a multi-threaded single process program. The last solution leads to a simpler implementation and entails a better exploitation of the hardware parallelism. Indeed, in comparison with a process, a thread uses only a small fraction of the operating system resources used by a process. In particular, the multi-threaded implementation eases the communication and synchronisation between the different components: a multi-process application communicates among its processes through traditional inter-process communication facilities (e.g. pipes, inter-process shared memory, Unix signals and sockets), while the threaded application communicates via the inherently shared memory of the process. The threads can maintain separate connections while sharing data in the same address space. Furthermore the concurrence and the synchronisation of the various tasks, that generally need a large effort in multi-process program, are easily obtained with specific system calls that coordinate in a natural manner the activity of the threads.

There are several definitions for thread libraries. The POSIX[1] specification (IEEE 1003.1c) is intended for all computing platforms and implementations are available, or under development, for almost the major UNIX system. Using the POSIX threads standard it is possible to ensure the portability of the code on different platforms.

5.4 Detailed sub-farm design

The design of the Event Filter sub-farm prototype is schematically shown in Figure 5-3. It is compliant with the High Level Design outlined in Section 4.2.3: the sub-farm receives and buffers events data coming from the Event Building sub-system (the DataFlow system) and subse-

1. POSIX (Portable Operating System Interface) is a set of standard operating system interfaces based on the UNIX operating system. The POSIX interfaces were developed under the auspices of the Institute of Electrical and Electronics Engineers (IEEE). The POSIX.4 committee concerns with creating a standard for writing multi-threaded programs.

quently forwards them to the processing tasks and to the mass storage. The interfaces between the sub-farm input (SFI) the sub-farm output (SFO) and the event handling system are provided by a data Distributor and a data Collector respectively.

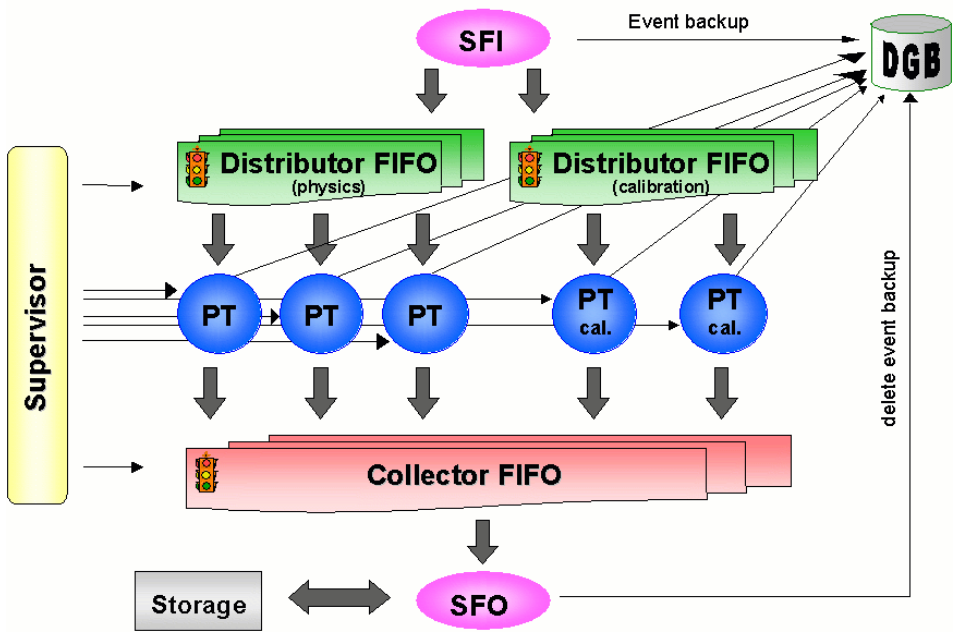

Figure 5-3 Design of the SMP sub-farm prototype.

In order to achieve a better exploitation of the hardware resources, all the components of the sub-farm have been implemented within a single multi-thread process. Every sub-farm component is assigned a thread scheduled directly by the OS kernel (1×1 scheduling model: to each user thread corresponds one thread scheduled directly by the kernel). One clear advantage is that, with this choice, load balancing, a critical parameter in the sub-farm operation, is automatically provided by the operating system scheduler.

The choice of the multi-threaded implementation stems from the fact that it eases in particular the communication and the synchronisation among the different architectural elements. The communication between the components can be achieved using the memory address space of the process itself, which is visible by all the threads: the Distributor and the Collector elements are simply FIFO buffers containing the pointers to the events stored in the process memory space. As a consequence, the event data-flow through the sub-farm is achieved by passing these pointers to the different components, whereas the real event does not change its physical location in memory.

The flow of the events in the sub-farm is regulated by the availability of the FIFO resources that are established by their level of occupancy. This mechanism ensures the data security during the passage of the events through the sub-farm: at least one pointer to the memory stored event is always present inside the Distributor and the Collector.

All the events are also stored in the Distributor Global Buffer (DGB) as soon as they are received by the SFI and they are removed after having been rejected by the processing tasks or sent to the final storage. With respect to the High Level Design, the Collector Global Buffer (CGB) has not been used, although it has been implemented, because there is no need to have a double copy of the same events (DGB and CGB): the lost events can be simply recovered from

the DGB and then reprocessed with a negligible loss of time. In any case using an SMP machine it is impossible to lose the events during their transit through the different components of the sub-farm because they all reside on the same machine: the only way to lose events is a system crash.

5.4.1 SFI

In the test configuration, the sub-farm input (or injector) is implemented by a thread that gets the events from a file and stores them in memory and in the Distributor Global Buffer.

The events come from the ATLAS simulation (DICE) and are converted to the CZebra format [5-7]. To simulate the calibration events, to be processed by calibration algorithms, the event type, available in the event header, is changed for a fraction of the total events; to simulate the real event size their original dimension (~50 KB) is padded to 100 KB or to 1000 KB according to the established type (calibration or physics respectively). Therefore the events are selected according to their type (e.g. physics or calibration events) and are sent to different Distributor FIFOs to be processed with different algorithms.

5.4.2 Distributor and Collector

The data Distributor and the data Collector are simply implemented by FIFO buffers. They coordinate directly the sub-farm main tasks through the thread synchronisation objects contained in them (mutex[1] and condition variables[2]). In this way the availability of the FIFOs resources, established by their level of occupancy, regulates the flow of the events in the sub-farm, giving the data driven behaviour of the system. The control of the occupancy level of the FIFOs is provided by semaphores based on mutexes and conditional variables; in this way it is possible to create semaphores more versatile than the standard POSIX ones.

The FIFO buffers are implemented by global c-structures visible by all the threads and containing the pointers to the events stored in memory. The Distributor FIFO receives the events from the SFI and distributes them to the processing tasks on their request; similarly the Collector FIFO is filled by the processing tasks and emptied by the SFO.

5.4.3 Processing Tasks

The processing task threads get the events from the Distributor FIFOs and process them according to their event type; eventually they fill the Collector FIFO with the filtered events. Two types of processing task have been implemented with suitable algorithms according to the different event types:

1. Mutex (mutual exclusion object) is a specially handled binary variable that all threads agree to use to guard access to a shared structure, handle or other resource. There is only one owner of the mutex at a time. A thread can manipulate the resource if it holds the lock on the associated mutex; other threads await the release of the mutex variable.
2. A condition variable is a synchronization variable, which creates a safe environment to test a condition under the protection of an associated mutex. If the variable is false the thread sleep on it and be awakened when it might have become true.

- the physics processing task runs a C++ version of the ATLAS Electromagnetic Calorimeter reconstruction software (Calorec++);
- the calibration processing task consumes CPU and memory by means of mathematical operation (e.g. loops of square roots and exponential using memory locations); their processing time is set to be ~10% of the Calorec++ processing task.

Besides a filter has been implemented inside the physics processing tasks: it rejects a fraction (typically 10%) of the events according to the cluster reconstructed energy. The rejected events are therefore removed from the DGB and the filtered ones fill the Collector FIFO.

5.4.4 SFO

The sub-farm output is also implemented by a thread; it gets the events from the Collector FIFO and, if required, stores them on a disk partition for the final storage. Finally the event files are removed from the DGB. The implementation of this component has been provided only for completeness and will be superseded by a proper integration with the DataFlow.

5.4.5 Distributor Global Buffer

The purpose of the Distributor Global Buffer is to ensure that events are not lost during their passage through the sub-farm. In the current implementation the DGB is a disk partition on which the events are stored as different files. They are stored as soon as they are received by the SFI and they are removed when they are rejected by the processing tasks or after being disposed of by the SFO.

5.4.6 Supervisor

The supervisor shall provide management functionality within the sub-farm. It controls all the sub-farm activities and provides a global co-ordination functions for the sub-farm. In this prototype the main thread controls a server socket and acts as a supervisor. Through a client socket it is possible to dynamically:

- control the sub-farm operations as the initialization, starting, stopping, resetting and re-starting of the sub-farm in a totally co-ordinate fashion;
- change some sub-farm parameters as the number of processing tasks, processing time, DGB activity, etc;

Moreover it is also possible to monitor the sub-farm status. Indeed, as each component of the sub-farm has a private set of counter variables that can be polled at any time, it is possible to monitor:

- the state and the level of occupancy of the FIFOs;
- the activity of each processing task (the number of processed, filtered, rejected events and the processing time);
- the activity of SFI and SFO threads (the number of events passed through the sub-farm).

5.4.7 Error handling

One of the must critical points of this design is the error handling. Since all the sub-farm is implemented in one single process, the failure of a processing task thread could lead to the crash of the entire sub-farm. The probability of this occurring can be drastically reduced by exploiting all the means available in the thread POSIX library. Indeed the synchronous signals, generated by the operating system as a consequence of a program error, are delivered directly to the offending thread. This thread can run itself the appropriate error recovering routine (signal handling routines and cancellation handling routines) and then kill itself preventing the loss of events.

In some cases, addressing errors in the common data memory space of the process may not lead to error signals: in this situation the error can induce faults in other threads at different times making a recovery strategy very hard to apply. This problem is currently under study and the key point is the evaluation of the probability of this event occurring. Indeed in case of a sub-farm crash the events, backed-up in the DGB, can be reprocessed by a new instance of the sub-farm process. For instance, to reduce the start-up time a spare sleeping process could be run in background ready to replace a crashed one.

5.5 Prototype's hardware

In order to avoid as much as possible interference of critical operating system aspects in the Sub-Farm code implementation itself (and to obtain a better reliability of both hardware and software components) the prototype has been developed on a commercial SMP architecture with a proprietary operating system, but has also been ported on an SMP commodity PC running Linux.

The technical choice has been Hewlett-Packard SMP servers running version 11.0 of the HP-UX operating system that provides kernel level POSIX thread and is POSIX 1003.1c compliant (draft 10). POSIX compliance allows for an easy porting of the code on other operating systems obeying the same standard. Indeed the code has been ported in other environments: Solaris, Tru64-Unix, Linux. In the latter case minor changes have been applied to the code due to the different level of the POSIX standard draft implemented in the kernel 2.2.x. The new kernel 2.4, which is about to be released, will be compliant to an higher level of the POSIX standard and will provide significant better SMP performance and scalability than is available in the 2.2 series (according to the tests performed on the development 2.3 kernel [5-5]).

The prototype has been developed on a 4-way HP K220 server based on 120 MHz PA-7200 processors with 1 MB + 1 MB L2 cache and with 512 MB of RAM. The shared memory and external resources are concurrently available via the (64 bits @ 120 MHz) RunWay bus, with a sustainable rate of 750 MB/s.

Scalability tests has been done on a 8 CPU HP N4000 and on a 20 CPU HP V2500 SMP servers powered by 440 MHz 64-bit PA-RISC PA-8500 processors with 1.5 MB L1 cache on chip (0.5 MB instruction + 1 MB data). The 8 CPU N4000 SMP server has an architecture based on two IA-64 system busses, with a total bandwidth of 3.8 GB/s, that connects the processors to the central memory controller which manages 8 GB of SDRAM; 240 MB/s I/O channels ensure a total I/O bandwidth of 5.8 GB/s. The architecture of the 20 CPU V2500 SMP server, shown in Figure 5-4, is based on a non-blocking 8×8 crossbar HyperPlane that provides 15.36 GB/s memory bandwidth with bi-directional 960 MB/s per port; the physical memory is 16 GB of

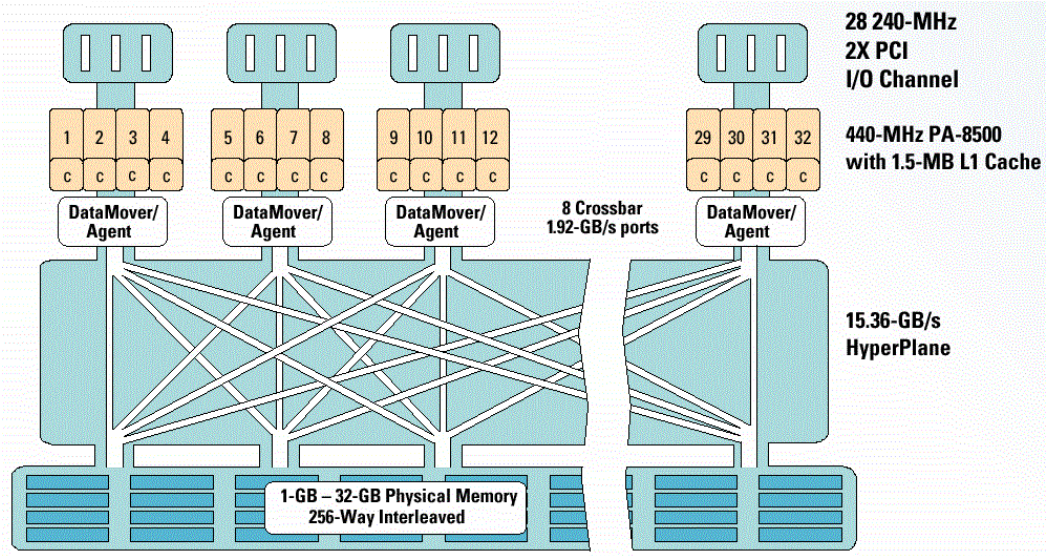

Figure 5-4 Architectural scheme of an HP V2500 server based on a non-blocking 8 × 8 crossbar HyperPlane.

SDRAM and the peak aggregate I/O channel bandwidths is 1.9 GB/s [5-8]. We acknowledge CILEA [5-9], a computer centre located near Milan, for dedicating us the servers N4000 and V2500, allowing us to perform the necessary tests. These machines cost beyond the present ATLAS funding but have been successfully used to validate the architecture.

Given the excellent performance to cost ratio of the SMP intel based platform and the good level of reliability and maturity achieved by the SMP support of the Linux Operating System [5-5], the prototype has been also ported on a SMP commodity PC running Linux: a COMPAQ ProLiant 5500 with 4 CPU PII Xeon clocked at 400 MHz and with 512 MB of RAM. The 4-way commodity SMP servers are emerging as the leading edge in the standards-based servers. The new 8-way boxes already offer superior price/performance ratio compared to proprietary server architectures and, if the cost trend will be the confirmed, they could be the base elements for implementing entire sub-farms on a single machine.

5.6 Test and performance

The general aim of the tests was to measure the global throughput and the scalability of the sub-farm and to check the software and hardware architectures. The tests have been performed on all the available SMP machines varying the running conditions such as the number of processing tasks, the event size, the processing time (by looping many times the reconstruction task to simulate different realistic values), the number of processors and the use of the Distributor Global Buffer. The software executed by the processing tasks is a C++ version of the ATLAS Electromagnetic Calorimeter reconstruction software. Every run consists of more than 20,000 events and in order to simulate the real ATLAS event size, their original size (~50 KB Monte Carlo data) is padded to 100 KB or to 1 MB according to the established type (calibration or physics respectively). Many of the results presented in the following for the K220 server are more extensively described in [5-4]. The same type of tests have been also performed on the N and V servers to study in particular the scalability of the implementation.

5.6.1 Throughput

In order to prove the robustness of the software architecture a series of tests have been performed varying the number of physics processing task threads up to 400 on the different available machines (HP K220, HP N4000 and COMPAQ ProLiant 5500). In this way the operating

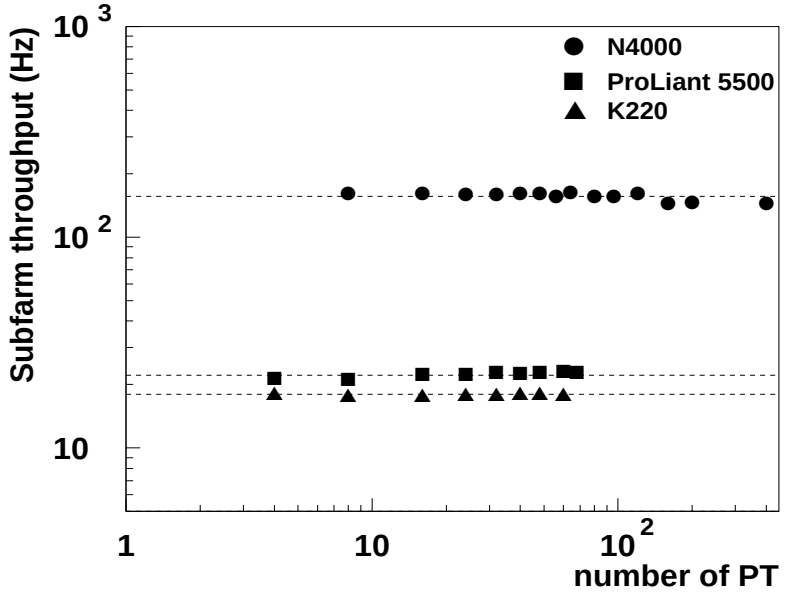

Figure 5-5 Sub-farm throughput as a function of the number of processing tasks on different architecture. The independence of the number of running PTs underlines the robustness of the software architecture.

system has to balance a very large number of processing tasks. The obtained results, displayed in Figure 5-5, show that the global throughput is not affected by the number of running processing tasks. This is the proof that the software architecture is able to manage several processing tasks for each CPU node without any degradation in the global performance.

Then series of runs has also been taken modifying the processing time from 0 to 50 units: one processing time unit is ~220 ms on K220, ~40 ms on N4000 and ~100 ms on ProLiant 5500. The results show that, as expected, the throughput is inversely proportional to the processing time and independent of the event size for realistic processing time. On both the K220 and ProLiant 5500 this test has been performed with an event size equal either to 1 MB or to 250 KB (see Figure 5-6 and Figure 5-7), the results show that the throughput of both set of measures overlaps perfectly, except for the case at processing time close to 0. A processing time equal to 0 means that the events are not processed and are passed directly from the Distributor FIFO to the Collector one; in this case the throughput approaches 450 Hz on N4000, 100 Hz on K220 and 61 Hz on ProLiant at 1 MB (400 Hz on K220 and 100 Hz on ProLiant 5500 at 250 KB): these values measure essentially the copying time in memory. Indeed, approaching very small event sizes, the throughput reaches 60 KHz on N4000 and 7 KHz on K220 showing that the hardware architecture of the backplane interconnects does not limit the performances: these results show the absence of bandwidth limits and bottleneck on transfer.

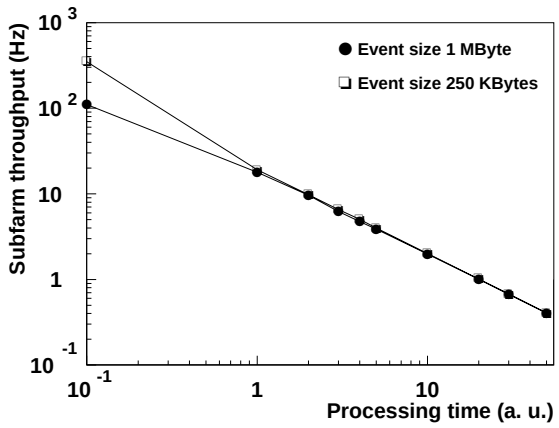

Figure 5-6 Global throughput as a function of processing time with different event size on K220 HP server. The processing time unit is about 220 ms.

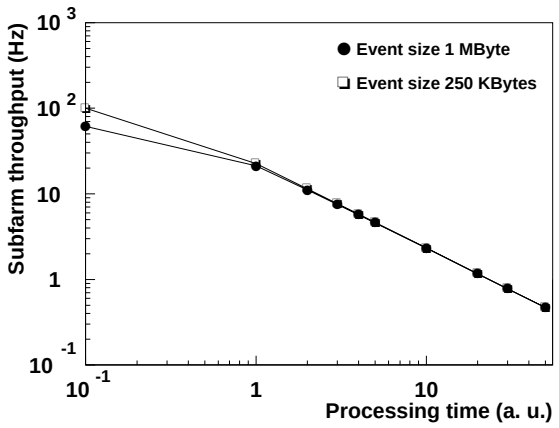

Figure 5-7 Global throughput as a function of processing time with different event size on commodity PC ProLiant 5500. The processing time unit is about 100 ms

5.6.2 Load balancing

The load balancing, which is a critical parameter of the sub-farm behaviour, is automatically ensured by the operating system scheduler and does not need any programming effort.

To better simulate the processing time expected with real reconstruction software, the

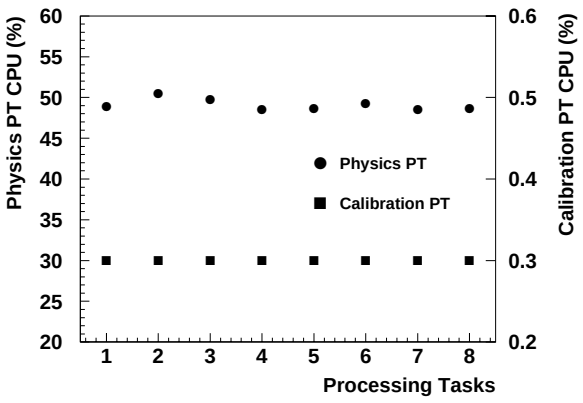

Figure 5-8 Load balancing of 8 physics and 8 calibration processing tasks on K220. Each processing tasks is well balanced against the others of the same type.

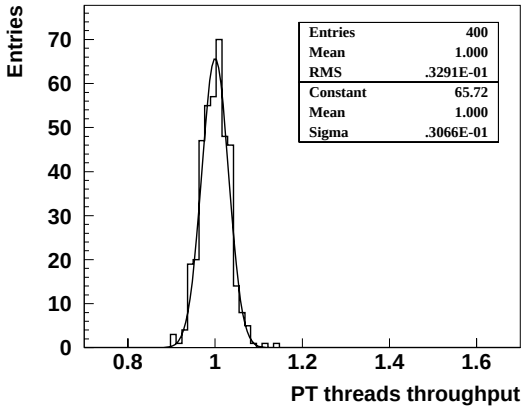

Figure 5-9 Distribution of the throughput of 400 processing task threads running on N4000. The load balancing is achieved at a level of 3%.

sub-farm has been tested running the reconstruction task either with a fixed number or with a random number of loops (from 0.2 s to 4 s on a K220). Figure 5-8 shows the results of the tests performed running concurrently two different types of processing tasks (physics and calibration) with an equal number of items (8 physics and 8 calibration). The achieved results prove that, with reasonable accuracy, each processing task is well balanced against the others of the same type and that the relative composition in the number of processing tasks does not affect the load balancing.

The load balancing is also ensured independently of the number of processing task as one can see in Figure 5-9: the operating system is able of balancing 400 threads at a level of few percent.

5.6.3 Scalability

One of the most interesting aspects of the multi-thread implementation is that, in principle, if there is no limitation in the hardware, the sub-farm multi-thread software allows for an exact scaling with the number of available processors in the SMP server. To check this important issue, a series of run increasing the number of the active CPUs in the V2500 server has been performed (see Figure 5-10). It is indeed remarkable that the global throughput scales almost

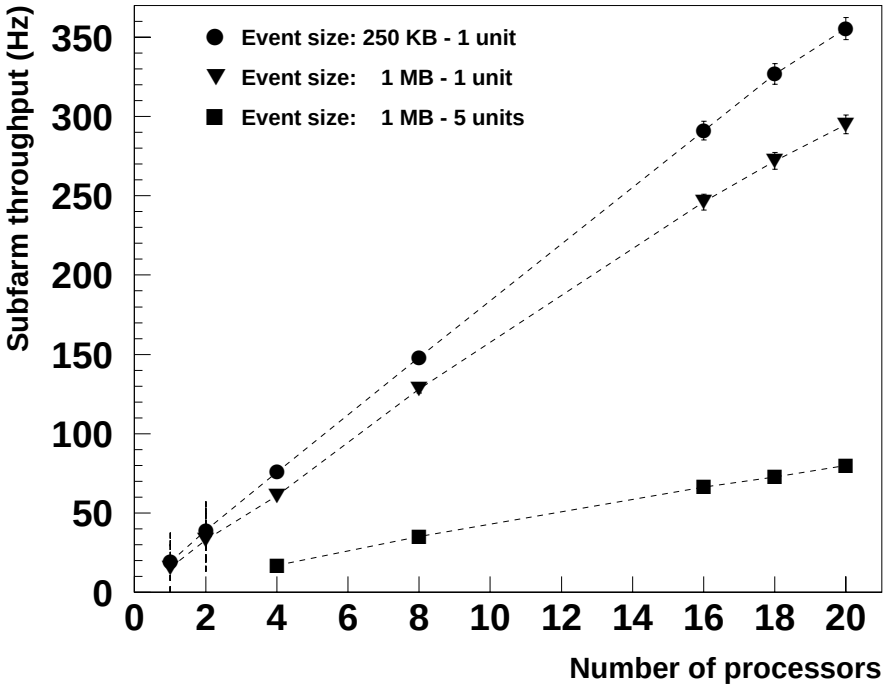

Figure 5-10 Throughput of the SMP prototype implemented on a V2500 server as a function of the number of CPUs with different value of event size and processing time (1 processing time unit = 40 ms). The latter has been taken low to test the architecture in the most critical conditions (at low processing time the shared resources are accessed more frequently.)

perfectly with the number of running processors, showing that an SMP sub-farm with up to 20 CPUs is feasible and that the V2500 hardware and the underlying operating system do not show bottlenecks of any kind. Different conditions as the event size and the processing time do not affect the results on scalability.

5.6.4 Robustness

The purpose of the Distributor Global Buffer is to ensure that events are not lost during their passage through the sub-farm. In this implementation the DGB is a disk partition on which the events are stored as different files. They are stored as soon as they are received by the SFI and

they are removed after being rejected by the processing tasks or disposed of by the SFO. In principle, another solution is to rely on the core-dump function of the operating system itself. This is particularly elegant because all the events in the distributor are in memory, and a copy of them will hence be found in the dump. This would avoid the time spent in disk writing for each event going through the sub-farm, but it has the obvious disadvantage of spending more time in the recovery process and of not addressing the crash of the sub-farm itself.

The correct DGB behaviour has been studied in many ways. First of all, at the end of run no event should remain in the DGB, and this has been found to be indeed the case. Moreover, in case of a simulated system crash, the events that were still to be processed have been found in the DGB and the recovery system embedded in the multi-thread implementation ensures that they are firstly processed when the sub-farm restarts before accepting new events from the Distributor: so no event is lost. The use of the DGB influences the performance as expected: it reduces the global throughput at small processing time and becomes negligible increasing the processing time as it is shown in Figure 5-11.

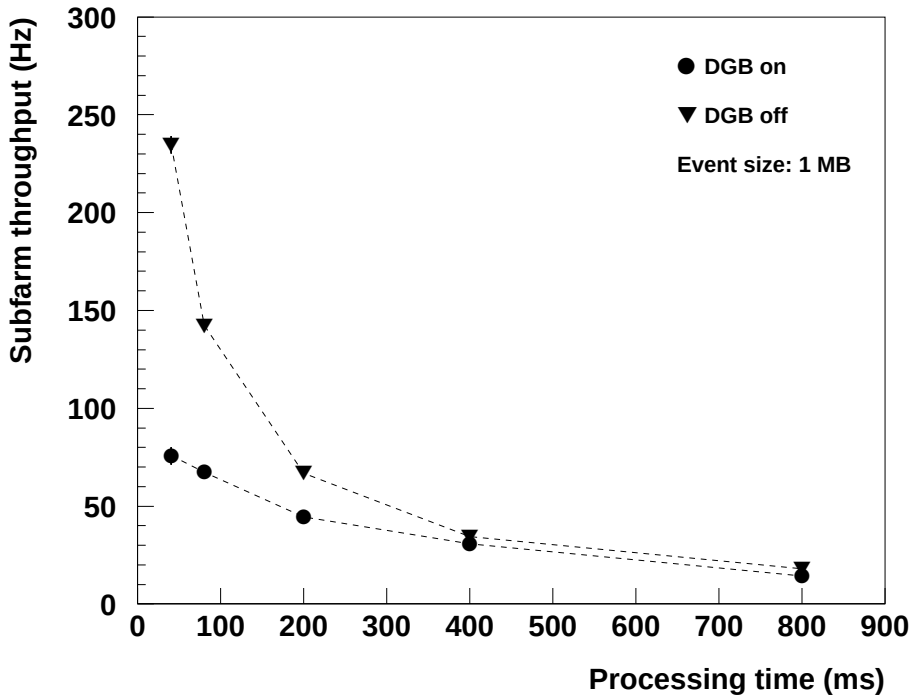

Figure 5-11 Influence of the use of the DGB on the SMP prototype implemented on a V2500 server (16 processing tasks running on 20 CPUs).

Finally a long term reliability test has been performed: the sub-farm processed more than 4 millions events in 3 days without any problem.

5.7 References

5-1 G.M.Amdahl, *'Validity of single-processor approach to achieving large-scale computing capability'*, Proceedings of AFIPS Conference, Reston, VA. 1967. pp. 483-485.

5-2 B.Lewis and D.J.Berg, *'Threads Primer: A Guide to Multithreaded Programming'*, (1996) ISBN: 0-13-443698-9

5-3 P.Anelli et al., *'Proposal for an Event Filter Prototype based on a Symmetric Multi Processor architecture'*, ATLAS internal note, ATLAS DAQ -1 Note 82 (1998), http://atddoc.cern.ch/Atlas/Notes/082/Note082-1.html

5-4 A.Negri et al., *'Detailed Design and Implementation of an SMP Event Filter Sub-farm'*, ATLAS DAQ -1 Note 128 (1999), http://atddoc.cern.ch/Atlas/Notes/128/Note128-1.html

5-5 Ray Bryant et al., *'SMP Scalability Comparisons of Linux Kernels 2.2.14 and 2.3.99'*, http://www.usenix.org/publications/library/proceedings/als2000/full_papers/bryantscale/bryantscale_html/index.html

5-6 *'Event Filter summary document'*, ATLAS internal note. ATL-DAQ-2000-005 (2000)

5-7 C. Meessen, *'The CZebra file format'*, http://marcpl2.in2p3.fr/EF/CZebra

5-8 Hewlett-Packard servers, http://www.unixsolutions.hp.com/products/servers/index.html

5-9 CILEA, Consorzio Interuniversitario per L'Elaborazione Automatica, Milan, http://www.cilea.it

CHAPTER 6

System integration and development

6.1 System integration: Event Handler API

The three different prototypes has been integrated within the full DAQ/EF-1 prototype through the implementation of a common Application Program Interface (API). The Event Handler API, which describes the interface between the DataFlow and the Event Filter, has been proposed in the EF/DAQ -1 Technical Note 98 [6-1]. This API provides the data-flow with an interface which is independent of the sub-farm implementation: it has been kept deliberately general with no account taken of possible specific implementation details or specificity. The effectiveness of the approach has been demonstrated when the three prototypes described in the previous chapters have been integrated with the main data flow.

For the purpose of overall design flexibility, two options are proposed for the event sending mechanism from the SFI to the Distributor and for the event sending mechanism from the Collector to the SFO. The so-called *Blocking* and *Non-Blocking* schemes are defined as:

- *Blocking:* the action of sending a message prevents the sending party from executing any other action until the message has been (successfully or otherwise) sent;
- *Non-blocking:* after initiating the dispatch of a message, the sending party may execute any other unconnected action, returning later to check the status of the dispatched message;

Message exchanges are executed in the hypothesis of synchronous operation: a message may be sent by a party only if the other is prepared to receive it and the receiving “process” has been started.

The functions of the API are intended to be called by the DataFlow in order to have I/O access to the Event Handler:

- **SFI - Distributor functions**
  - EH_OpenDist: open connections between the SFI and the Distributor - initialisation
  - EH_CloseDist: close connections between the SFI and the Distributor
  - EH_ResetDist: reset SFI-Distributor communication links
  - EH_BlkReady: check that the Distributor is ready to receive an event (blocking call)
  - EH_BlkSend: send an event from the SFI to the Distributor (blocking call)
  - EH_NblkReady: check that the Distributor is ready to receive an event (non-blocking call)
  - EH_NblkSend: send an event from the SFI to the Distributor (non-blocking call)
  - EH_StatusDist: check transmission status of a non-blocking send

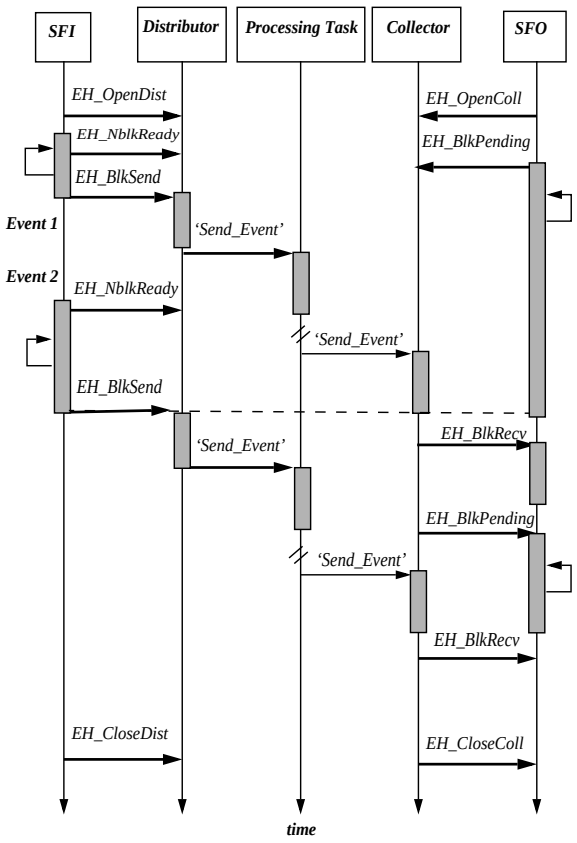

Figure 6-1 Sequence Diagram of the interaction of the Event Filter with the Main Data Flow.

- **Collector - SFO functions**
  - EH_OpenColl: open connections between the SFO and the Collector - initialisation
  - EH_CloseColl: close connections between the SFO and the Collector
  - EH_ResetColl: reset SFO-Collector communication links
  - EH_BlkPending: check that the Collector is ready to receive an event and return the number of pending events (blocking call)
  - EH_BlkRecv: receive an event from the Collector to the SFO (blocking call)
  - EH_NblkPending: check that the Collector is ready to receive an event and return the number of pending events (non-blocking call)
  - EH_NblkRecv: receive an event from the Collector to the SFO (non-blocking call)
  - EH_StatusColl: check transmission status of a non-blocking receive

The timing chart in Figure 6-1 illustrates the overall function call flow as seen from the Data-Flow point of view. It does not take into account the internal operation details of the Distributor and Collector. In the present implementation, the readiness of the Distributor and Collector are checked by non blocking calls while the transfer operations are performed by a blocking call.

All the prototypes have been integrated with the data-flow through the implementation of this Event Handler API; the transport protocol used is CORBA (ILU [6-2]). The integration has

been successfully tested running the sub-farms on the hardware prototypes located in the corresponding home institutes (Pavia for the SMP prototype, Marseille for the commodity PC prototype and Alberta for the THOR prototype) and the SFI and SFO elements on a RIO2 board located in the DAQ-1 laboratory in Building 40 at CERN: all the events sent to the sub-farm from the SFI had been correctly received back in the RIO2 board. Clearly this integration test was aimed at establishing the ease of the integration procedure rather than making performance measurements.

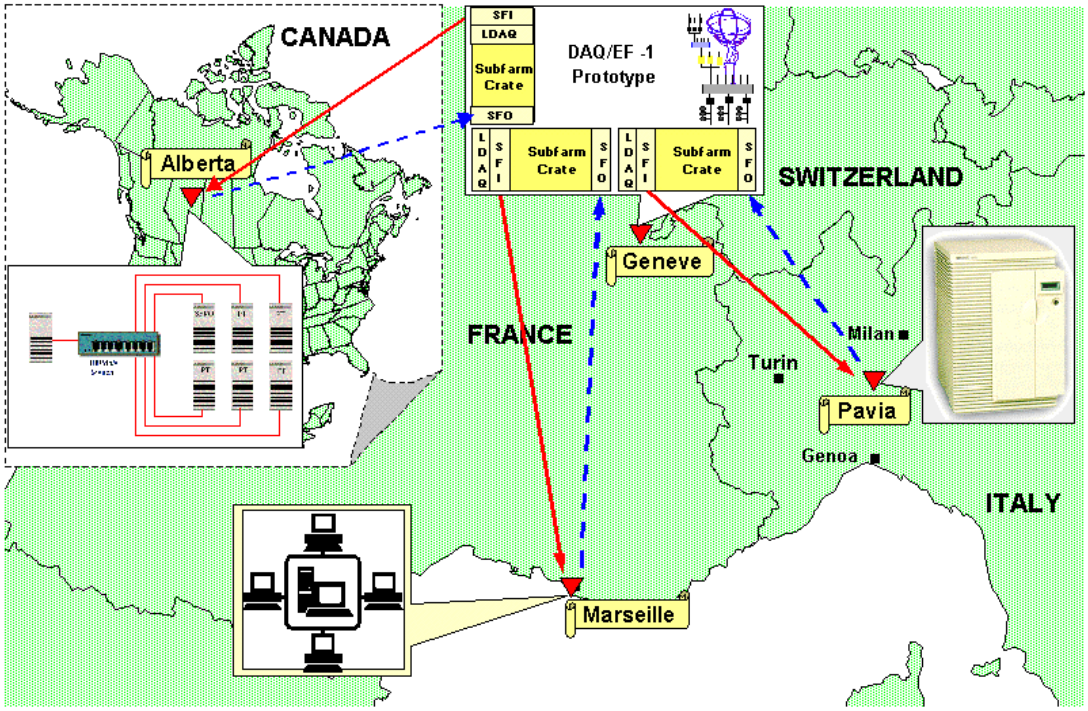

Figure 6-2 Integration of the three EFF prototypes

6.2 Comparison of the prototypes and main common conclusions

The previous chapter describes the available results from the SMP prototype, in terms of implementation of sub-farm software, throughput, scalability and robustness. This subsection presents a comparison between all the three different EF prototypes introduced in Chapter 4 and a summary of the main conclusions of the EF prototype work carried out in the framework of the DAQ/EF-1 project.

6.2.1 Data redundancy and robustness

All the prototypes implement a form of the Distributor Global Buffer, as described in the High Level Design. In all implementations this currently takes the form of a disk partition, where events are stored during their passage through the sub-farm. This is important in order to be able to recover from a crash of the processing task or of one Distributor, or even of the sub-farm itself. As seen from the measurements in all the prototypes, and as expected, the time spent in disk writing is important only if the processing task time is smaller or comparable to it.

With the prospected speed of disks, or disk-arrays, compared to the ATLAS reconstruction and analysis time (the order of the second), one could say that this item should not be a limitation in the performance of the sub-farm. On the other hand, since an error-free sub-farm is clearly not achievable, a better estimate of the tolerable, unbiased data loss rate for the ATLAS detector should be given in the future.

6.2.2 Data communication mechanisms

Several mechanisms have been studied in the prototypes to send data to the various processing tasks on different processors. In the distributed implementations (Commodity PC and THOR) this is achieved via either CORBA (ILU [6-2]) or TCP structures.

In the framework of the Commodity PC prototype a comparative study between these two different communication protocols lead to the conclusion that CORBA is not really needed and in some aspects not appropriate for event transfer in the Event Filter data flow. A communication mechanism based on the TCP protocol appears more suited for our needs: performance and reliability with a very small set of operations.

As described in the previous chapter, the least demanding solution in term of resources is the one adopted for the SMP sub-farm, where the computer memory is used to hold the events and hence pointers, passing event addresses to the processing threads, are the only communication mechanism used. Given the RAM costs, this should not constitute a problem for reasonably sized sub-farms, even though the hardware of the sub-farm will be intrinsically more costly.

An interesting alternative solution is represented by the SCI interconnect between processors, currently studied by Alberta, which builds a sub-farm of the cc-NUMA[1] type, using commodity components. The SCI hardware, from Dolphin Interconnect Solutions Inc. [6-3], is capable of a maximum full duplex bandwidth of 8 Gb/s with latency estimated at 2.3 µs at the applications layer. A comparison between SCI and 100 Mb ethernet is shown in Figure 6-3. This performance level will allow THOR to be used as a shared memory machine. The SCI hardware is shipped with a full API for Linux, and hence programming for the event filter application will be simplified. Upon delivery of the SCI hardware, work will begin to upgrade the event filter code to comply with the common EF data flow software, which is described in Section 6.4, and to write I/O objects to take advantage of the SCI bandwidth. Work will also begin to test distributed versus shared memory models of the event filter with the help of the SCI interconnect.

One might note that, in all the latter examples (with respect to CORBA/ILU), some more work is needed to accomplish the correct identification of objects which need to be known in the outside world (e.g. for Back-End software purposes).

6.2.3 Throughput and scalability

The output of the ATLAS Event Builder, after LVL2 selection, is currently estimated at ~ 1 KHz. From this number, together with estimates related to the processing time needed for each event (the foreseen average processing time will be of the order of one second), in the *“AT-*

1. cc-NUMA (cache coherent Non Uniform Memory Access) is a method of configuring a cluster of microprocessors in a multiprocessing system so that they can share memory locally. Cache coherence is provided by a coherence controller on each node.

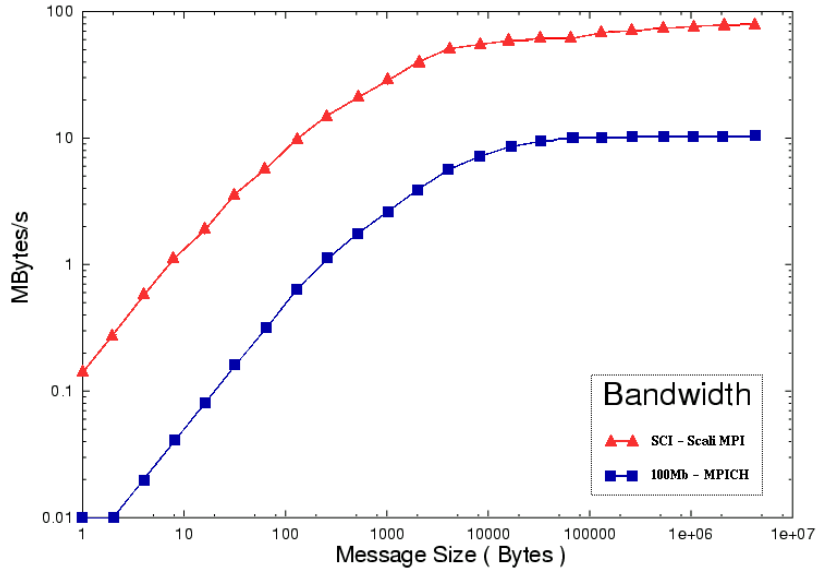

Figure 6-3 Scalable Coherent Interconnect (SCI) compared to 100 Mb ethernet. MPI Bandwidth as a function of Message size (Concurrent across 16 nodes - Bandwidth per node).

LAS Technical Proposal" [6-4] several figures have been given, related to the size and the granularity of the Event Filter sub-system. From the measurements obtained by the three prototypes, one could easily infer that no problem is envisaged concerning the fulfilment of the requirements imposed by the ATLAS DAQ system. In fact, all sub-farms are currently achieving, or are close to achieving, throughputs consistent with the required ATLAS numbers.

Complex scenarios arise from the mixture of physics and non-physics data (e.g. calibration data) flowing through the farm. In this case, the current understanding is that load balancing of events within and between sub-farms will be a key issue, if one wants to merge those data types in the same sub-farm.

In the scalability studies, very interesting performances have been measured. As shown in the previous Chapter, in the SMP case, where the use of kernel threads implies relying heavily on the POSIX operating system structure, the prototype has been shown to scale up to several tens of processors (see Figure 5-10) and hundreds of processing tasks (Figure 5-9). In the distributed architectures, assuming the average event size of 1 MB, the use of new generation commercial networking should be capable of ensuring the necessary performances. The throughput and scalability plots of the two distributed prototypes are shown in Figure 6-4 and Figure 6-5. A particular role is played by the THOR prototype, where the scalability studies are extending beyond the scope of DAQ/EF-1 Prototype, addressing the problem of multiple sub-farms and their partitioning.

6.2.4 Sub-farm configuration and monitoring

Configuring the sub-farm and keeping track of its behaviour is an essential ingredient which all the prototypes have considered in their studies. In the Commodity PC implementation a lot of progress as been made in understanding the supervisor element, based on Java Voyager (see Section 4.3.1.2). This is particularly relevant to demonstrate the feasibility of a complex distributed architecture, where bottlenecks might arise from the inability of communicating to and

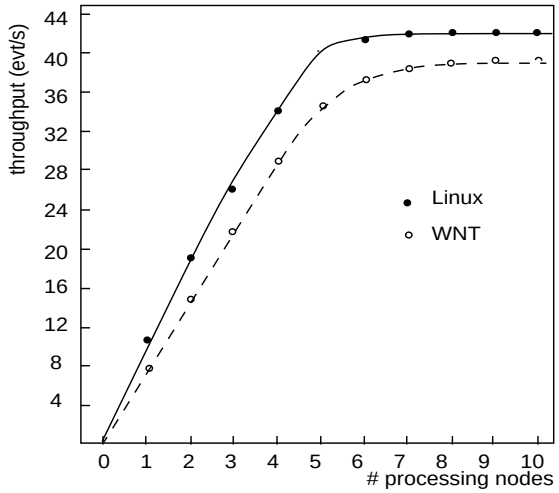

Figure 6-4 Commodity PC prototype (CERN EFF facility) throughput as a function of the number of processing node (Dual processors PCs) on Linux and WindowsNT environments. Two processing task are running on each node and the event size is 2 MB.

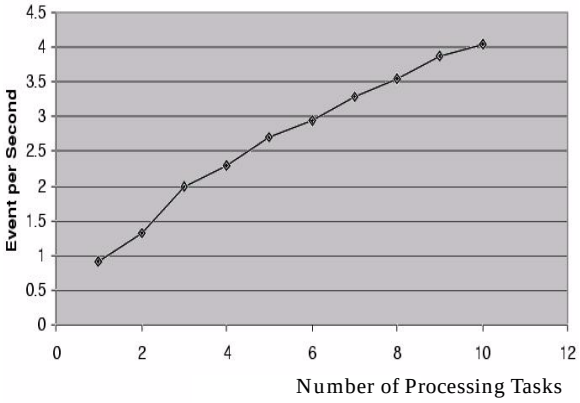

Figure 6-5 THOR prototype throughput as a function of the number of processing tasks. The higher slope in the throughput curve between two and three processing tasks reflects the fact that two nodes (and hence two Ethernet cards) are used for the three processor case while only one Ethernet card in one node is used for the two processor case.

keeping control of thousands of processors. The achieved results are extremely positive and are of course beneficial also to other implementations: the test show that the supervisor has been able to control 1023 distributed components without any problems.

The monitoring component has been studied in the same context, too, providing a graphical interface to the sub-farm performances, which extends for THOR across many sub-farms. In the SMP case, internal monitoring is provided through proprietary tools which control the behaviour of the processing tasks, whilst for general supervision a tool such as the one developed in Marseille could be adopted.

6.2.5 Event Filter System Assessment

The concept of independent SubFarms for the EF has been shown to be desirable, because it naturally solves problems related to farm partitioning and, more generally, flexibility and scalability. The integration of different hardware and software implementations of EF Sub-Farms has been helped by the clean definition of an API connecting the EFF to the DataFlow. No problem has been met in running EF SubFarms geographically separated from their DAQ data source.

Different data communication mechanisms have been studied, showing that many types of EF SubFarm are possible, from widely distributed configurations to large SMP ones. The importance of including data security features, such as the Distributor Global Buffer (DGB) for facilitating error recovery in the EF has been shown. The configuration and monitoring of the EF system has been shown to be a key issue for the maintainability of large Farms and solutions (e.g. based on Java mobile agents) have been proposed.

The values of global throughput and the performance scalability (both in terms of processing elements and components) measured in all the prototypes are already close to the ATLAS requirements.

6.2.6 Architectures and costs

The selection of the best processor technologies for the EF system will be important from the point of view of performance and cost. The problem to evaluate different architectural solutions satisfying the requirements of the ATLAS Event Filter sub-system has been tackled using distinctive points of view: on one side, the use of low-cost hardware lead to the studies of widely distributed architectures, on the other side, the quest for a robust, self-consistent solution based on the adoption of commercial SMPs.

In the outside world, there are many examples of similar architectures, in particular big computing centres are more and more based on PC-like clusters, whilst mission critical data-driven companies tend to adopt proprietary multi-processor systems. In evaluating the cost of each solution, one should bear in mind that an Event Filter sub-system will need to last several (~15) years, hence the cost is not merely limited to the buying of the hardware. Maintenance of the entire system is an issue which has to be addressed as well to evaluate properly the cost, including software and hardware upgrades, mean time between failures, hardware modifications, operating system issues and, of course, the associated manpower.

The success of the PC-like solutions, anyway, backed-up by the presence of performant SMP-compliant operating systems, with thread-based kernel, suggests a very promising implementation based on multi-processors machines built around Intel chips of the new IA-64 generation. This will allow for a low-cost, reasonably distributed architecture for the Event Filter. Currently all the major vendors present on the market Intel SMP boxes with 8 processors based on the Intel Profusion chipset, which is being standardize and considerably reducing the costs of designing and building commodity component 8-way SMP system. Many joint ventures and partnerships exist in the commercial world for the new chip and this will certainly be beneficial for us, on the right time scale. It should be noted, in conclusion, that, at the present stage, even for the whole farm there are two approaches that look equally appealing. One gathers processing power by building a self-made architecture around small boxes and the other tries to exploit commercial solutions and partnerships to fulfil the Event Filter goals. Investigations in both areas are in progress, to come up with the best, most performant and robust solution for ATLAS.

6.3 The proposed EF system architecture

The conclusions of the prototype work executed in the framework DAQ/EF-1 project and reported above have led to a proposed architecture for the EF system that has been published in the *'ATLAS High-level Triggers, DAQ and DCS Technical Proposal'* [6-5]. The global HTL-DAQ architecture has been presented in the Section 3.6.

6.3.1 Major Use Cases

Figure 6-6 shows the context diagram of the EventFilter. It interacts with the other systems in the following way:

- Receives fully assembled events from the EF I/O and communicates selected events to the mass storage via the EF I/O. The data communicated back to the EF I/O will depend on the type of event analysed and may contain only the results of the EventFilter selection.
- Provides the sub-detectors with the means of performing event-data monitoring and calibration.
- Receives configuration and control commands from the Online Software system and provides status information for the purpose of operational monitoring.
- Accesses calibration, alignment and geometry information from the databases.
- Provides updated information to the databases after having processed calibration specific event data.
- Uses algorithms developed and implemented by the Offline software group.

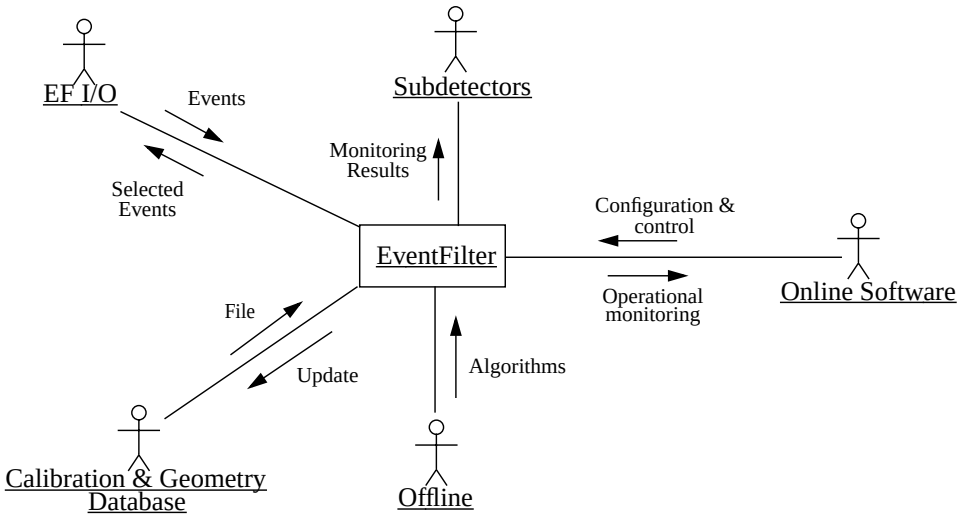

Figure 6-6 EventFilter context diagram.

6.3.2 Main Functional Elements

Figure 6-7 shows the global view of the EF system, containing the EventHandler and EF Supervision subsystems. The latter implements the interface to the Online Software system and also uses the interface implemented by the Online Software system.

The EventFilter is factorized in several independent EventHandlers, which exploit the computing power necessary to accomplish the various tasks of the system. No restriction is made on the possible implementations (PC subfarm, SMP machine) or hardware architecture (single or separate farms for LVL2 and EventFilter). This factorization provides an easy way to implement flexibility and scalability. It is the responsibility of the EventBuilder subsystem to ensure the load balancing among the different EventHandlers of the EF, by keeping account of the relative occupancy of its destinations and assigning new events accordingly.

The EventHandler is responsible for moving events through the different steps of the processing operation. The EF Supervision controls and monitors the operation of the whole EF. The EventHandler contains the EventFlow and the Processing Task subsystems. They also con-

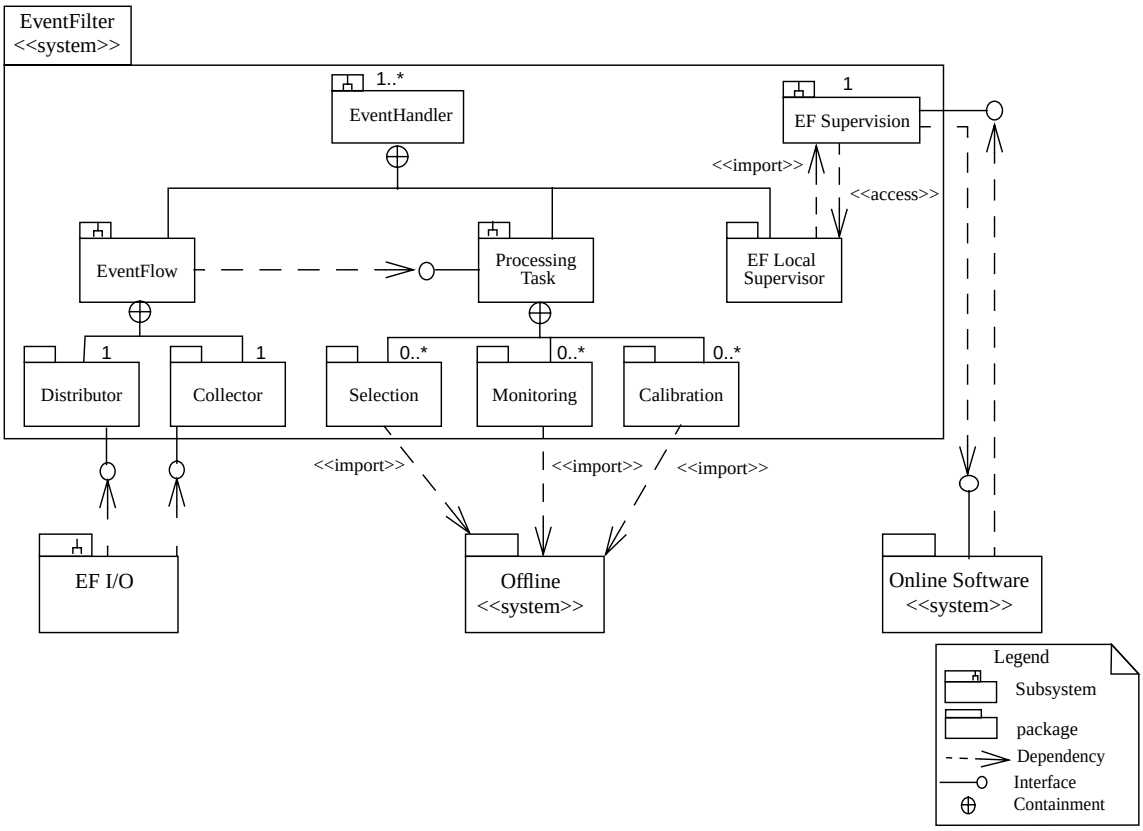

Figure 6-7 EventFilter package diagram.

tain an EF Local Supervisor component which imports into the EventHandler the functionalities of the EF Supervision.

The EventFlow contains one Distributor and one Collector. The Distributor receives events from the EF I/O (see Figure 6-7) and communicates them to the different processing tasks of the EventHandler via an interface implemented by the Processing Task which isolates the different possible technologies of the EventHandlers. The Distributor distributes events according to event type. It also keeps a safe copy of the event throughout its lifetime of an event in the EventHandler. The Processing Tasks implement the selection criteria as well as other ancillary tasks such as event-data monitoring or calibration analysis. They work independently on the basis of one event per processing task. The Collector sends the selected events back to the EF I/O (again via an API).

6.4 Common EF data-flow software

This section presents the common software implementation of the EFF data-flow. The purpose is to provide a common software framework based on object oriented design, with a modular architecture that permits the software integration of the different existing prototypes and the exploring of new protocols or architectures.

The specific software data-flow of each prototype was written in C and was especially designed for the implementation specific communication protocol: ILU for the commodity PC prototype, TCP for the THOR one, reference passing in memory for the SMP implementation.

On one extreme we have many small commodity network hosting the different components run by independent processes. On the other we have a single computer running the EF sub-farm where all components are contained in a single process and run by different threads. Therefore care must be taken to separate the communication protocol from the application layer so that different protocols and hardware architectures can be easily supported.

The first version of the common EF data-flow software has been developed in the framework of the PC commodity prototype and therefore was mostly suited for a distributed environment. This version, called Version 3 (Section 6.4.1) and written in C++, is documented in the DAQ/EF -1 technical note 132 [6-6]. The acquired experience and the feedbacks from the porting of the Version 3 framework on a non-distributed architecture (the SMP prototype) led to a more implementation independent software architecture, called Version 4 (Section 6.4.6) and documented in [6-7].

6.4.1 The Version 3 framework

According to the High Level Design presented in [6-8] (and described in Section 4.2.3), the events flow in the EF system through three main elements: the Distributor, which receives the events from the SFI, the Processing Tasks, where they are analysed and eventually selected and tagged, and the Collector, which send the accepted events to the SFO.

The main functionality of the Distributor is logically split in three components named D1, D2 and D3. The first implements the EH API to receive events from the SFI and is responsible for the event queue management. According to their type (contained in the event header), events are sent to the appropriate instance of the D2 component. The D3 component acts as a buffer for the processing tasks and requests events from the D2 component as soon as they have free space for a new event. The separation of D2 and D3 is needed in a distributed architecture (Commodity PC prototype), but in a single machine implementation the two components can be collapsed. The Collector functionality is split in two components: C1 and C2. The first, of which there will be an instance per processing task, will provide the processing task with the means of disposing of the event after processing. C2, of which there will be one instance per sub-farm, will receive events sent by the C1s and is responsible for the buffer management of the CGB. As the D3 element the C1 is need only in a distributed implementation.

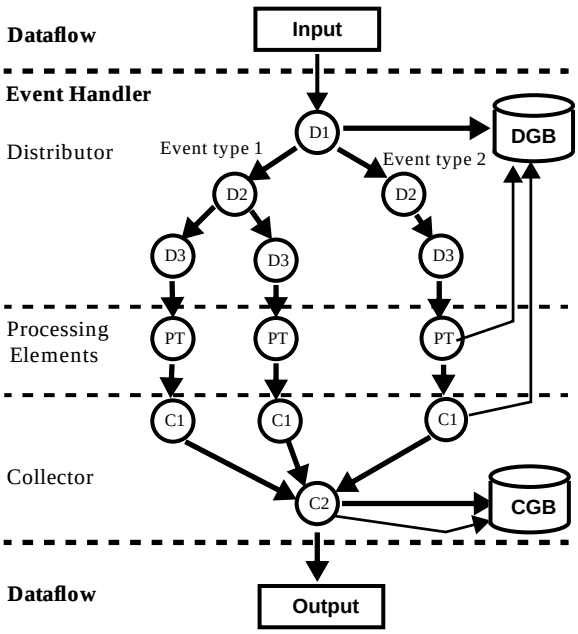

Figure 6-8 Functional model of the EF data flow.

6.4.2 User requirements

The experience of the first implementations has led to define the following user requirements. Requirements containing the word *shall* are mandatory. Those containing the word *should* are strongly recommended and justification is needed if they are not followed.

- Event transfer between component and with SFI-SFO:
  - *Shall* allow to use any event transfer protocol.
  - *Should* support client-server connection resets, even in a middle of an event transfer
  - *Shall* support process relocation if the architecture is distributed.
  - *Should* provide automatic reconnection of clients.
  - *Shall* support binding at run time (dynamic) between components.
  - *Shall* ensure that events are recoverable in case of error: corrupted data, incomplete transfer, processing task crash, etc.
- Component control:
  - *Shall* allow to use any component control protocol.
  - *Shall* support binding at run time (dynamic) with controlled components.
  - *Shall* support dynamic component parameters modifications.
  - *Shall* support stopping and restarting a component.
  - *Shall* support resetting of components.
  - *Shall* provide information on the component activity: throughput, number of events that traversed it, number of event contained in the component, number of events rejected by the processing task.
  - *Shall* inform a supervisor of errors or key steps in the component activity.
- Component behaviour:
  - *Shall* be independent of the event transfer and control protocol used.
  - *Shall* work with client or server for input or output.
  - *Shall* share the same control interface whatever its behaviour is, including activity information.
- The Event Filter:
  - The distributor shall direct events toward type specific processing tasks
  - A processing task shall be associated to a single event
  - Shall support dynamic reconfiguration
  - Failure of a processing task shall not have any repercussions on the system functionality

6.4.3 Component architecture

The components are just a functional unit of the EH. All components can be seen as simple FIFO queues of a given depth; from this point of view the PT component is thus a queue that can

hold at most one event. Events traversing the sub-farm are moved from one queue to the other, where they may be processed and eventually deleted. Components, identified by a unique name, are interconnected according to the client server paradigm. Each queue has an input and an output, that can behave as server or client. Servers accept connection and requests of multiple clients; clients connect and interact with only one server at a time. Each client must have an associated source or destination component.

In order to permit the use of different communication protocols between the components, the latters has been split in three different type of base elements (see Figure 6-9). The pivot element of the component is the *CoreElement* which implements the component specific behaviour. Each *CoreElement* can support an input and output element (*IOElement*), which encapsulate the event transfer protocol and, and a control element (*CtlElement*) that provides the component interface for the supervisor. Each base element is implemented by a corresponding C++ class.

A component is thus an aggregation of these elements and is constructed at start-up time. Many different type of components can be built combining these basic elements. The component composition can be dynamically modified at run time. Each component is implemented by a different process and each element is run by its own thread.

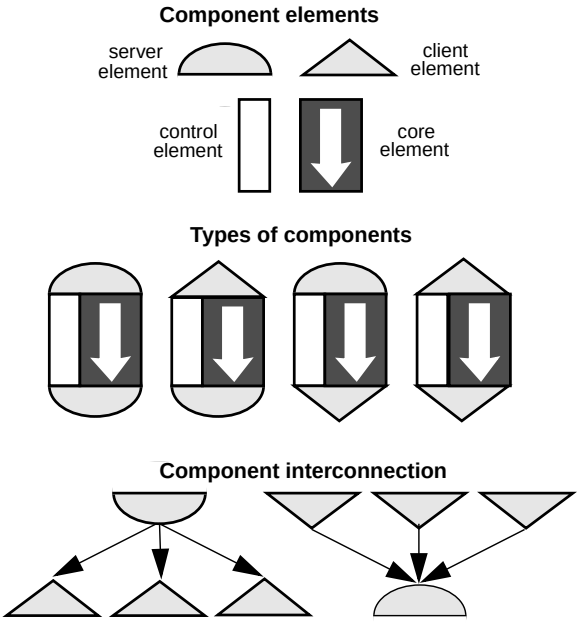

Figure 6-9 The different basic elements, how the different element can be aggregated into component and how components can be interconnected.

6.4.3.1 The *CoreElement* base class

The *CoreElement* is the base class and corner stone of the component. All other elements are connected to it. The *CoreElement* has a set of parameters that are accessible to the *IOElement* and the *CtlElement*. It is basically a FIFO queue managed by a set of methods schematized in Figure 6-10.

Event insertion or extraction are not necessarily atomic for some event transfer protocols, which are slow and could fail. For this reason this operations have been split into basic and atomic steps. Before putting an event into a remote component, the client might first be sure there is space for that event into the remote component. If there is, it must reserve this space before sending the event. If the sending fails for whatever reasons, it will need to release the reserved space. Otherwise it can store the event. The queue is therefore divided in the four logical zones shown in Figure 6-10. The methods quoted in the lower part of the figure are used to move events from one division to the next one; all operations are reversible. Incoming events are events for which space has been reserved (in the case of a server) and which are not yet fully copied in the queue. Similarly, outgoing events have been requested for output and have not yet been fully copied to the client. The methods in the upper part of the figure allow to know the actual size of the dynamic zones.

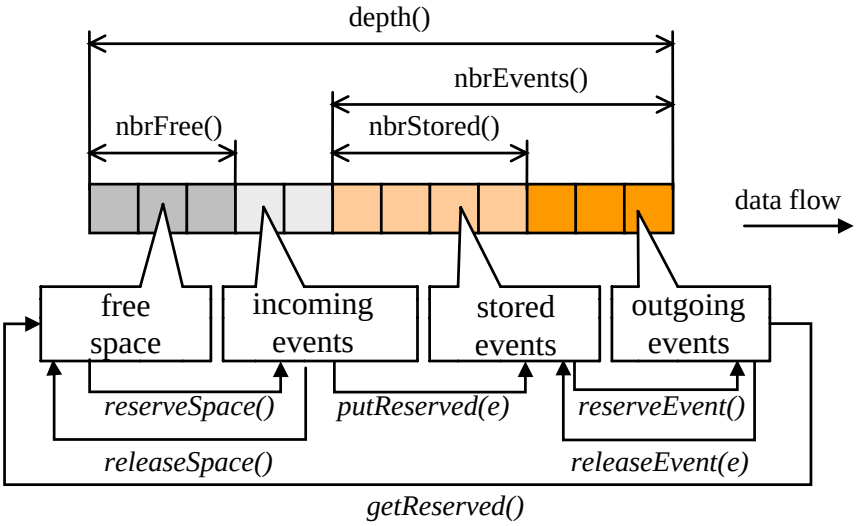

Figure 6-10 The CoreElement queue and the methods to manage the FIFO.

The classes corresponding to the buffer elements (D1, D2, D3, C1 and C2) and the processing tasks (PT) are derived by inheritance rom the *CoreElement* class.

6.4.3.2 The *IOElement* base class

These elements are the links between components. They are responsible of the transfer of events using their specific protocol. The *IOElement* class is the base class for all types of input and output element. A distinction is made between input and output *IOElements* and client and server *IOElements.* Clients are connected to one server *IOElements.* An input client will pull events and an output client will push events. Server *IOElement* may have multiple simultaneous connected clients and waits and executes client requests.

Components are interconnected by the client server connections. To keep the specification of this interconnection simple, each component is given a unique identifier by the user. The association between names and addresses is provided by the naming service. When a component is built, a source or a destination component identifier for events may be defined. When a source or a destination is defined, it means that the component should behave as a client for the input or the output. Otherwise it will behave as a server.

In the standard Version 3 framework the only supported event transfer protocols are CORBA (ILU) and TCP / IP.

6.4.3.3 The *CtlElement* base class

The purpose of control element is to provide an interface with the supervisor. A wrapper template class may encapsulates the communication protocol. The *CtlElement* is used to interact with the *CoreElement* from the outside world. It may be used to get the component parameters, change them, or get statistics about the component behaviour. The *CtlElement* uses the pointer to the *CoreElement* to call the appropriate methods or access the appropriate variables of the *CoreElement*. The default communication protocol in UDP: each *CtlElement* is executed by a thread that manage a server UDP socket.

6.4.4 Implementation of SMP prototype in the Version 3 framework

La Figure 6-11 shows a detailed view of a distributed implementation of an Event Filter sub-farm. The EF components are distributed across different machines and each of them is implemented inside a different processes. Inside each process the various components elements are executed by different threads. Data is pushed by the SFI towards the server input element of D1. A thread gets events from D1 (a server output element) and sends them to the proper D2 server input element. Events are then extracted from D2 by the client input element of D3. Processing tasks extract events from the server output element of D3 and pushes the selected ones to the server input element of C1 which itself pushes events towards C2 from which they are pulled by the SFO.

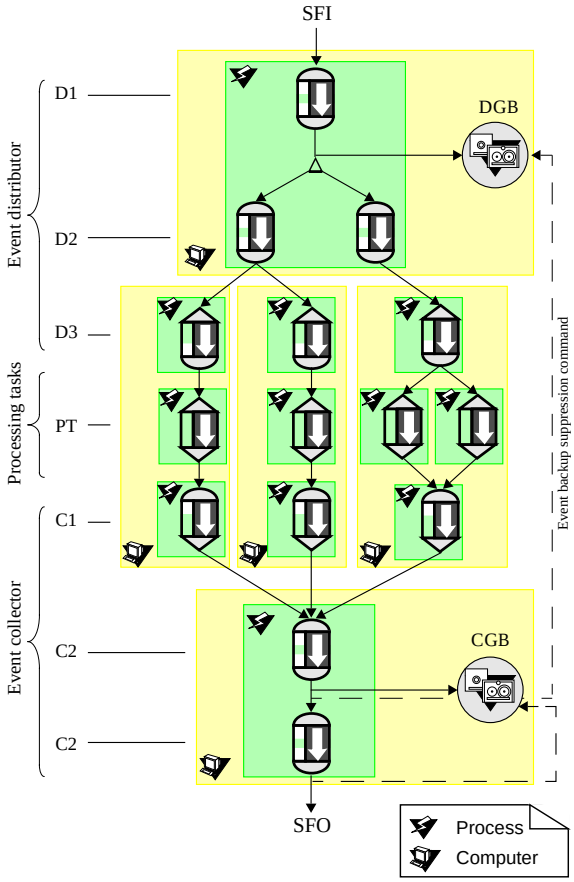

Figure 6-11 Detailed representation of a distributed implementation of an Event Filter sub-farm (Commodity PC prototype).

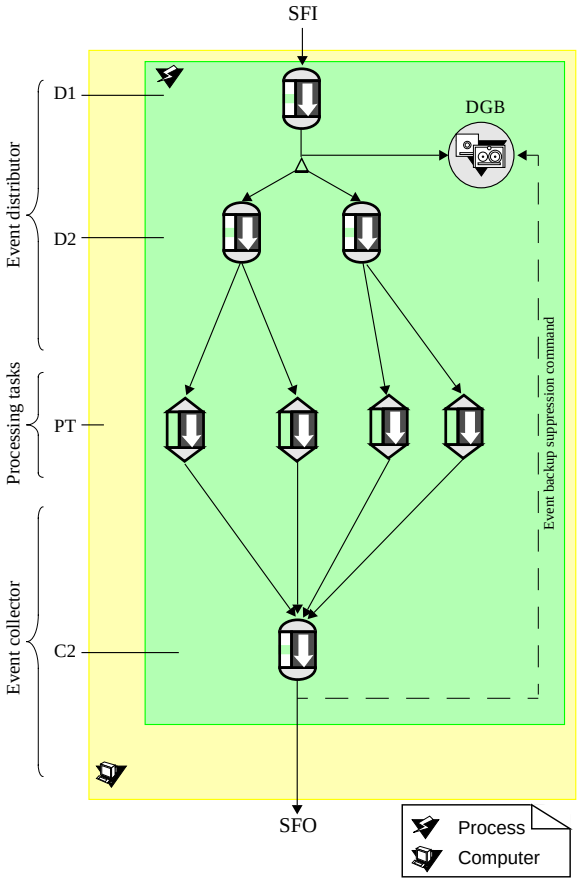

Figure 6-12 Detailed representation of the a “single process” implementation of an Event Filter sub-farm.

The original SMP prototype, described in Chapter 5, is based on a contrasting philosophy: all the sub-farm components reside on the same machine and are implemented within a single multi-threaded process. The MT design allows a better exploitation of the SMP advantages in data sharing and communication between the different components of the sub-farm. The SMP prototype has been integrated in the Version 3 framework safeguarding this software architectural philosophy (see Figure 6-12). The D3s and C1s elements are redundant and therefore are not implemented. The Processing Tasks directly request events from the D2s and send the accepted ones to the C2.

Exploiting the inheritance mechanism a *SMP_IOElement* class, whose instances can be used to interconnect components in the same memory space (through reference passing), has been derived from the base *IOElement* class.

The Version 3 code, which was developed in the framework of the PC commodity prototype, foresees the use of a thread for each IOElement. This solution can be favourable in a distributed environment, where each component is implemented inside a different process and possibly a different computer. In this case the communication between the distributed component can be optimized running asynchronously and in parallel the input and the output tasks and eventually the processing task. Indeed, given the slowness of the communication mechanism (e.g. TCP), for distributed implementation, this solution avoid possible dead-time in the event flow related to a sequential execution of the two I/O tasks. But for fast and error-free communication solutions (as reference passing in memory for the SMP implementation) this asynchronous behaviour is not only redundant but entails also the overhead connected to the management of two unnecessary threads. The decoupling between the two I/O operations had not been trivial and many methods of the base classes had been completely overloaded.

In the Version 3 code each component has a *CTLElement* UDP server executed by a different thread of the component process. This approach is not applicable in the “single process” SMP implementation, because only a thread of the process can manage communication with the other processes (catch the unix-signal, handle sockets, etc). However the components information resides in the same address space and therefore a single thread can control the behaviour of all the sub-farm components and manage the communications with the supervisor. This approach entails the almost complete redesign of the *CTLElement* and of the naming server concept[1].

6.4.5 Comparison of the different implementations on SMP

In order to compare the performance of the different software framework, three different implementations have been tested on 4-way Compact Proliant 5500 (Intel Pentium II Xeon clocked at 400 MHz):

- The original SMP prototype (written in C and labelled V2), specially designed for this hardware architecture.
- The basic Version 3 implementation (labelled V3tcp), more suited for distributed environment. Each component is implemented inside different processes, which however reside in the same machine; the communication mechanism is based on TCP/IP.
- The SMP prototype implemented in the Version 3 framework (labelled V3smp) as described above.

In order to allow the comparison of the three implementations and to avoid inter-computer communication bottlenecks, the sub-farms are configured in a stand-alone mode: there is no connections with the external data-flow elements (the SFI and SFO elements are not implemented), the data are reads directly from disk. In all the prototypes the D1 elements are filled by an “injector” thread that read the events from disk, while the C2 elements are emptied by another thread, that subsequently delete the pulled event.

1. A method to reference (from the supervisor) the different components inside the process must be implemented.

Figure 6-13 shows the throughput of the three sub-farm implementations as a function of the processing time. The overhead due to the integration of the SMP prototype in the Object Oriented Version 3 framework entails a throughput loss of ~16% respect to the original specially designed code. As expected at low processing time, the effect of a communication bottleneck, due to the TCP/IP protocol, is visible for standard distributed implementation (V3tcp).

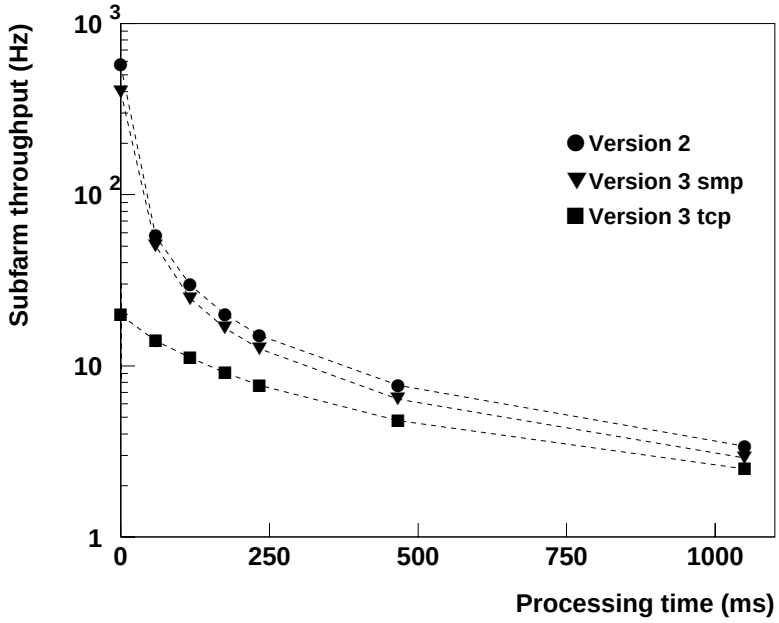

Figure 6-13 Prototypes throughput as a function of the processing time.

Clearly the TCP communication mechanism between components appears to be inadequate in a SMP architecture. Figures 6-14 and 6-15 show the throughput loss of the TCP based implementation with respect of the Version 3 compliant SMP prototype as a function of the number of processing tasks and therefore of the number of processors. On a 4-way machine and considering an event size of 2 MB, the measured loss is 13% for processing time of 1 s and raises to 25% for t_{PT}=0.5 s. The scalability straight line in the plots allows to roughly[1] forecast the throughput loss on a 8-way machine: the forecasted loss varies from 22% (for processing time of 1 s) to 45% (for t_{PT}=0.5 s).

Another possible implementation of the EF on SMP boards is a multi-processes software architecture with communication based on System V IPC shared memory[2]. The expected performance is almost the same as the multi-thread single-process implementation, but the effort needed to co-ordinate the access of the various component to the shared data were greater (the MT library itself provides the necessary synchronization system calls).

This implementation could, however, reduce the possibility of un-recoverable processing errors. The difficulty to implement a safe and reliable error recovery procedure, in case of process-

1. The hardware communication architectures are clearly different. The considered 4-way machine is based on two buses, while cross-bar switches are used on the 8-way systems.
2. The Inter-Process Communication (IPC) protocol provides a set of interfaces to manage the exchange of data between processes. The “shared memory” methods control access to memory zones accessible by more than one process.

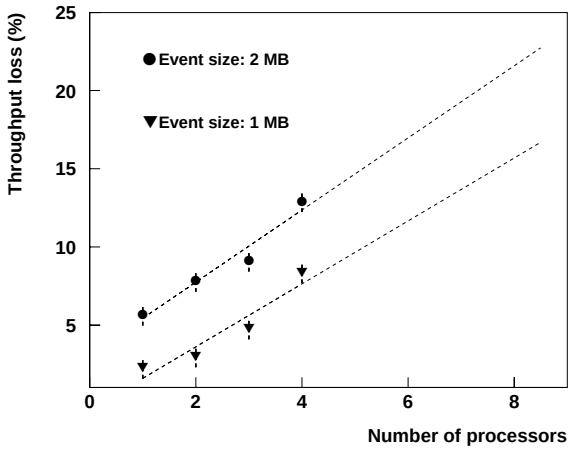

Figure 6-14 Throughput loss of the V3tcp implementation respect to the V3smp one as a function of the number of used processors on the 4-way machine for a processing time of 1 s.

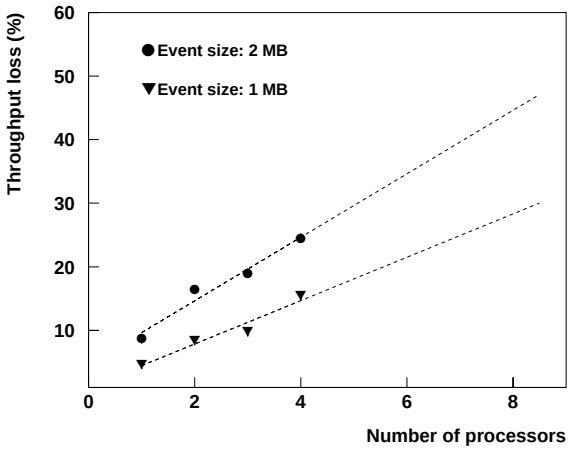

Figure 6-15 Throughput loss of the V3tcp implementation respect to the V3smp one as a function of the number of used processors on the 4-way machine for a processing time of 0.5 s.

ing task failure (e.g. segmentation violation signal), is indeed the major drawback of the multi-thread implementation. In theory, a MT program can handle a system error signal, determine exactly what happened, kill any dangerous thread responsible for the corruption, repair the data and continue. But it is impossible to exactly identify and correct any error in the shared process memory space. The multi-process solution based on IPC shared memory could reduce the total amount of data shared by the sub-farm components and therefore could limit the occurrence of these errors.

6.4.6 The Version 4 framework

The Version 4 of the common EF data-flow software is mainly a simplification of the Version 3 implementation to take into account past experience and comments and criticism from users and reviewers. The simplification is also the result of the experience gained in translating the data-flow software into Java. The implementation, whose class inheritance tree in shown in Figure 6-16, is done in C++, but some Java ideas for the helper class (Socket, Thread, String) has been adopted.

The implementation of Java equivalent classes has lead to a simplification of the management of sockets and threads. Also the access to the component parameters and statistics and the building and combination of the components has been streamlined.

A *RefIOElement* class, based on reference passing, has been implemented for event transfer between components residing in the same address space. This mechanism fundamental in the SMP implementation can be used also in a distributed architecture; e.g. the D3, PT and C1 components reside in the same computer and can be implemented within a single process.

To facilitate the reference passing communication mechanism, the asynchronous behaviour of the two *IOElements* of each component has been modified, as in the Version 3 compliant SMP implementation (see Section 6.4.4). The rigid corresponding “one thread for each component element” has been relaxed. Depending on the required functionality, each of the four basic elements may or may not be run in their own thread: a processing task *CoreElement* is run in its

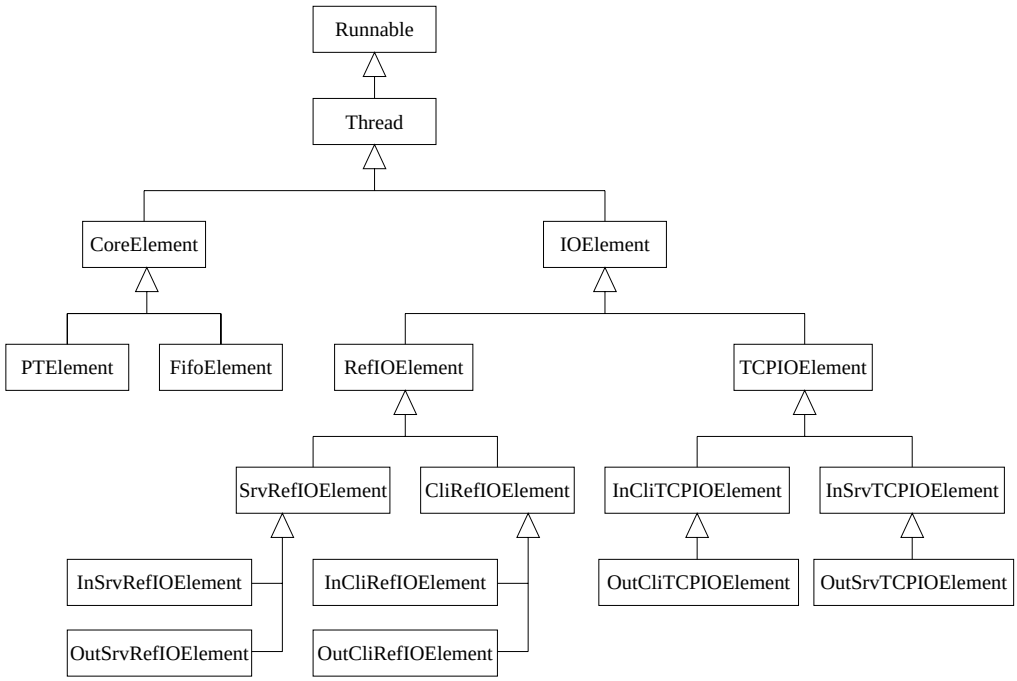

Figure 6-16 Class inheritance tree.

own thread, while a Fifo *CoreElement* code is executed by a thread of another component; *RefIOElement* client and server are not run in a thread, but *TCPIOElement* servers are; *TCPIOElement* client may possibly be run in their own thread. Furthermore two new methods are added to the *CoreElement* class: the *pullEventReq()* and *pushEventReq()* methods allows the *CoreElement* thread to execute directly and synchronously the I/O methods coded in *IOElement* class. If the *CoreElement* is a passive object (i.e: it has not its own thread) the methods required to fetch or insert the events are executed by the threads owned by the associated *IOElements* or by client threads of another component. Indeed *IOElements* that are run by their own thread or whose operation is executed by the client thread may ignore the *pullEventReq* and *pushEventReq* methods.

Two classes are inherited from the *CoreElement* class: the *FifoElement* class, which permit the implementation of the distributor an collector components, and the *PTElement* class, which defines the processing task component.

The *FifoCoreElement* class is a simple buffer for events with a specified depth. It maintains a queue of events protected by a mutex for asynchronous access by the input and output elements and semaphores to control the *reserveSpace()* and *reserveEvent()* operations. Since the component is passive and performs no special operations on the stored event, there is no thread associated to it. Thus if the input or output element are clients, a thread that calls their *pullEventReq()* and *pushEventReq()* methods must be launched repetitively.

The *PTCoreElement* is the base class for the processing task. It can only contain and process one event at a time. When processing is completed, the PT will delete the event or push it out after having set a semaphore indicating that the event is ready. This strategy ensures that it is possible to use server or client for both input and output.

Currently there are two set of classes for two different communication models. The transfer of events between components located in different processes and/or different computer is ful-

filled by the *TCPIOElement* classes, which are based on the TCP/IP protocol. The *ReFIOElement* classes, instead, concern the transfer of events in the same address space.

Each element of the *TCPIOElement* classes has its own socket. To connect to a remote server a client needs to obtain its TCP/IP address (host name or IP address and port number) via the naming server. For the *TCPIOElement* the naming service is a process running on a specific computer to which components may send requests to register an association or get the associated address. When servers are created, they create a listening socket. Clients will get the address of the remote server using the naming service. If the server component crashes or is moved on a different computer the connections will be restored automatically via the naming service.

The *RefIOElement* instances are used to interconnect components in the same address space. When a server element is created, it registers its name and address (this*) association in a map (STL) which is a static variable of the base class. The client can easily retrieve this address or wait a predefined time until the other threads have had time to start up and register the server name and address. The client then calls a connect method giving a reference to itself. The server will use it to inform clients of the obsolescence of its own reference if it is destroyed.

In order to easy the single-process SMP implementation, a specially designed *UDPCtlServer* class allows to manage multiple *CtlElement* in the same process so that there is a unique listening port and thread to process supervisor requests.

6.5 References

6-1 *'The Event Handler API'*, ATLAS DAQ -1 Technical Note 98, http://atddoc.cern.ch/Atlas/Notes/098/Note098-1.html

6-2 *Inter Language Unification*, ftp://ftp.parc.xerox.com/pub/ilu/ilu.html

6-3 Dolphin Interconnect Solutions Inc., http://www.dolphinics.no

6-4 ATLAS Collaboration, *'ATLAS Technical Proposal'*, CERN/LHCC/94-43 (1994), ISBN 92-9083-067-0

6-5 ATLAS Collaboration, *'ATLAS High-level Triggers, DAQ and DCS Technical Proposal'*, CERN/LHCC/2000-17 (2000), ISBN 92-9083-160-0

6-6 C.Meessen et al., '*Event filter farm component implementation design Version 3*', ATLAS DAQ/EF -1 Technical Note 132, http://atddoc.cern.ch/Atlas/Notes/132/Note132-1.html

6-7 C.Meessen et al., *'EventFilter Dataflow Software'*, ATLAS Internal Note ATL-COM-DAQ-2000-056

6-8 C.P. Bee et al., *' High Level Design of the Sub-Farm Event Handler'*, http://atddoc.cern.ch/Atlas/Notes/061/Note061-1.html

CHAPTER 7

b-tagging with High Level Trigger

7.1 The ATLAS Trigger Strategy

7.1.1 Overview of the ATLAS trigger strategy

The ATLAS trigger strategy foresees a ~10^7 reduction of the event rate at three levels: LVL1, LVL2 and Event Filter. The accepted rates at each level are given in Figure 7-1. The output target rate for LVL1 is ~40 kHz, which allows a safety factor of almost two, compared to the initial design capability of 75 kHz. The estimated uncertainty on the *pp* inelastic cross-section is about 30%.The total uncertainty on the main background processes, however, could be as large as a factor of two.

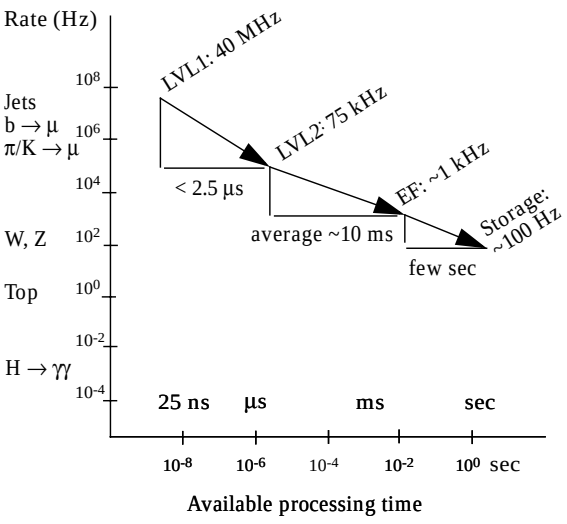

Figure 7-1 The three levels of the ATLAS trigger and their event rates and processing times.

After a 10^3 rejection factor achievable at LVL1 [7-1], a reduction of another three orders of magnitude is still needed to reduce the rate to an adequate value. The boundary conditions are a maximal LVL1 rate of 75 kHz and a maximum bandwidth to mass storage of approximately ~100 MB/s (which corresponds for instance to an output rate of 100 Hz for an event size of 1 MB). The output target rate for LVL2 is around 1 to 2 kHz, but it depends on the optimum separation between LVL2 and the Event Filter.

Through the selection chain from LVL1 to the EF, the trigger objects are progressively refined and made more specific. New trigger objects may be added at LVL2 and in the EF. Trigger objects are combined in lists of selection criteria (physics menus), which cover the majority of LHC physics studies. They are intended to provide a common focus for physics and trigger-performance studies. They are designed to be simple, inclusive and to satisfy the physics requirements with as short a list of trigger items as possible. The one exception to this is *B*-physics, where selection of particular decay modes must be done in the LVL2 trigger.

LVL1 objects are characterised by a small number of attributes and a set of discrete p_T-threshold values. The thresholds are generally given at the point where the LVL1 algorithms are 95% efficient. The isolation thresholds will change with the p_T of the trigger object, becoming looser for higher p_T candidates and being completely removed at very high p_T. The LVL1 menus for low and high luminosity are shown in Table 7-1.

Table 7-1 Low and high-luminosity menus [7-1]. The first letters are the trigger type, the following number is the threshold value in GeV, the "I" label identifies a possible isolation request and the last number is the required multiplicity. LVL1 trigger objects are shown in capital letters, whereas LVL2 trigger objects start with lower case letters.

Low luminosity		High luminosity	
Trigger	**Rate (kHz)**	**Trigger**	**Rate (kHz)**
MU6	23	MU20	3.9
		MU6 × 2	1
EM20I	11	EM30I	22
EM15I × 2	2	EM20I × 2	5
J180	0.2	J290	0.2
J75 × 3	0.2	J130 × 3	0.2
J55 × 4	0.2	J90 × 4	0.2
J50 + XE50[a]	0.4	J100 + XE100	0.5
T20 + XE30	2	T60 + XE60	1
		MU10 + EM15I	0.4
Other triggers	5	Other triggers	5
Total	**44**	**Total**	**40**

a. XE: Missing energy.

At low luminosity the MU6 trigger selects events for *B*-physics studies. The threshold for the two EM object trigger is set as low as possible to maximise the efficiency for $H \rightarrow \gamma\gamma$ and $Z \rightarrow ee$ decays. The inclusive-jet threshold has been set high to reduce the rate and make more room for other triggers. This is because the additional jet rejection available at LVL2 is small, so the useful LVL1 threshold is effectively limited by LVL2 rate requirements. Multi-jet and jet + E_T^{miss} triggers are given priority when sharing out the rate budget for these types of triggers. The thresholds of the multi-jet triggers are also chosen to give acceptable rates for LVL2. For the J50 + XE50 and T20 + XE30 triggers, the thresholds and rates should be taken as indicative. The LVL1 high-luminosity contains mostly the same objects as the low luminosity menu,

but with higher thresholds and/or rates. An additional trigger at high luminosity is MU10 + EM15I. Another extra trigger EM20I + XE is being studied.

7.1.2 Physics and event selection strategy

The aim of the HLT strategy is to provide an implementation independent scheme for the selection of LVL1 accepted events by LVL2 and the EF, which should be flexible enough to cope with changing run conditions and needs for physics processes. From the physics selection point of view, the HLT should be seen as a single logical unit. The actual selection process will be shared between the LVL2 and the EF, where the latter operates on full events after event building. In order to maximize the global selection performance, the robustness and flexibility offered by the distinguishing and complementary architectural features of LVL2 and EF (as described in Chapter 2) must be properly exploit performing the needed selection at the right level. Flexibility is needed to adapt to changes in the luminosity (even during a fill of the LHC), variations in the background conditions, and new requirements derived from physics and from improved understanding of the detector. In order to keep the selection as unbiased as possible towards new physics processes, only inclusive selections using combinations of very few objects are used, wherever feasible.

Selections in the HLT stem from primary and secondary RoIs (defined in Section 2.4) provided by LVL1. Most of the rejection is achieved after processing only the primary ones. This reduces the average number of RoIs to be inspected to less than two per event.

The physics and event selection strategy proposed is composed of four elements:

- **Algorithms.** They are used to identify objects (such as electrons or jets) together with their properties, or to determine global features of the event.
- **Sequences.** The sequence defines the order of execution of the algorithms (e.g. according to complexity) and aims to reduce the size and cost of the HLT system, while maintaining the flexibility needed and maximizing the physics potential.
- **Selection.** It is the procedure which, using the objects with their properties and a set of criteria, decides whether the event is to be rejected or not. The selection schemes are derived from the classification described below. They must be modular to allow the implementation of different parts of the selections at different stages of the HLT. In order to maximize the discovery potential, the selection schemes generally only use inclusive signatures. Except for the case of B-physics, reconstruction of exclusive decays is not required and no topological variables (e.g. the calculation of invariant masses from a combination of several high-p_T objects) are used in the selection, although this is technically feasible at LVL2 or in the EF (e.g. to select $Z \rightarrow l^+l^-$ decays exclusively).
- **Classification.** The classification scheme summarizes the signatures used to classify the accepted events for subsequent physics analyses. The schemes [7-2] have been derived from the analysis-level selection criteria for the various physics channels, as documented in Ref. [7-8].

  The properties associated with the trigger objects used for the classification are their transverse momentum and in some cases an isolation requirement. The criteria always contain high-p_T objects (with transverse momenta typically above 20 GeV); however, additional objects with lower transverse momenta are used in some cases.To allow studies

of the background and to provide more flexibility to the analysis, lower thresholds, respect to the ones foreseen for the offline analysis, will be chosen for the trigger selection wherever possible and appropriate. Where the system requirements will not allow this to be done for all events passing the cuts, pre-scaled samples will be accumulated for the lower thresholds.

The full list of the classification schemes can be found in Ref. [7-2]. The classification does not reject any events; the event is only categorized according to the signatures listed in order to facilitate subsequent analysis.

7.1.3 Selection Algorithms and Selection Sequence

In this section an overview of some selection algorithms and their performance is presented. These studies show that the currently available HLT selection algorithms have good physics performance, i.e. they provide good efficiency for physics objects and therefore high acceptance for the physics channels of interest. At the same time, they lead to a sizeable rate reduction of LVL1 accepted events. This physics performance is generally achieved while simultaneously fulfilling the requirements on the system performance; e.g. most of the RoI-guided LVL2 algorithms are already close to running within the final LVL2 latency constraints.

However the calculated HLT output rate is somewhat larger than the assumed number of about 100 Hz to be written to mass storage. Further reduction of the rate due to genuine high-p_{T} objects can be obtained by either increasing the p_{T} threshold or by requiring a more exclusive selection. For the latter, more specific physics processes would have to be targeted at the HLT selection level, and not only at the classification stage.

7.1.3.1 Selection of electron and photons

The inclusive electron and photon triggers are expected to contribute an important fraction of the total high-p_{T} trigger rate. After the selection in LVL2 and the EF, the remaining rate will contain a significant contribution from signal events from Standard Model physics processes containing real isolated electrons or photons ($\mathrm{W} \rightarrow \mathrm{e}\nu$, $\mathrm{Z} \rightarrow \mathrm{ee}$, direct photon production, etc.).

The electron and photon triggers can be viewed as a series of selection steps of increasing complexity. After receiving the LVL1 electromagnetic (e.m.) trigger RoI position, the LVL2 trigger performs a selection of isolated e.m. clusters using the full calorimeter granularity and detailed calibration. This selection is based on cluster E_T and shower-shape quantities that distinguish isolated e.m. objects from jets. Electrons are identified at LVL2 by associating the e.m. cluster with a track in the Inner Detector.

At EF the improved performance, which is due to the availability of more detailed calibrations and more sophisticated algorithms with access to the full-event data, results in sharper thresholds and better background rejection. In the case of electrons, bremsstrahlung recovery will be performed for the first time at the EF. In addition, a photon-conversion recovery procedure will be applied to photon candidates at the EF.

Reconstructing jets at the EF level improves the total jet trigger rate compared to LVL2 by only 20–30%. No significant reduction is achieved unless the algorithm is tuned for lower efficiencies. The expected rate after the HLT for an inclusive selection of isolated electrons (nominal LVL1 threshold of 20 GeV) at low luminosity is 41 ± 6 Hz. This corresponds to a reduction in

rate of a factor of 141 and an efficiency of 80.8% with respect to LVL1. Of this reduction, LVL2 contributes a factor of about 41 and the EF adds a further factor of about 3.4.

After the HLT photon selection (excluding the conversion identification algorithm), the inclusive rate for single isolated photons (nominal threshold of 40 GeV) is 57 ± 7 Hz for low luminosity. This corresponds to a reduction in rate of a factor 100 and an efficiency of 84.7% with respect to LVL1 (where nominal thresholds of 20 GeV are used), to which LVL2 contributes with a factor of 70. Since the LVL2 calorimeter trigger has access to the full granularity information, most of the overall HLT rejection comes from this step. The additional rejection by the EF is at the cost of a loss of about 10% of the single-photon efficiency.

7.1.3.2 Selection of muons

The purpose of the high-level muon trigger is the identification of muon tracks in RoIs indicated by the LVL1 trigger, and the accurate measurement of their position and transverse momentum. LVL2 and the EF must reject LVL1 muon candidates arising from the cavern background, non-prompt muons (such as those from decays of K and π mesons), and muons of p_T below the required threshold.

The tracks found in the LVL2 muon trigger are extrapolated for combination with Inner Detector and Calorimeter information. Matching between muon tracks measured independently in the Muon Spectrometer and in the Inner Detector validates the prompt muon selection whilst reducing the contamination of secondary muons from π and K decays. This is especially important for the LVL2 B-physics trigger in low-luminosity running, for which the selection of prompt low-p_T muon events defines the input sample to the full scan of the Inner Detector.

The expected rate for $p_T > 6$ GeV muons after LVL2 selection is 4.3 kHz. This corresponds to a rate reduction of about a factor of five with respect to LVL1. that the p_T resolutions achievable using the LVL2 stand-alone muon algorithms (5.5% at 6 GeV and 4% at 20 GeV) are close to the ultimate offline performance (4.5% and 2.5%, respectively). The EF will have access to all the data of the event and will be able to run offline algorithms to reconstruct muons in the ATLAS detector. The possible improvement coming from offline analysis can only be checked with EF studies, which are not yet available.

7.1.3.3 Selection of event with missing transverse energy

The E_T^{miss} + jet trigger is an example of a trigger based on the combination of a global variable (E_T^{miss}) and localized Regions of Interest (RoIs) in the detector. The bulk of the trigger rate will result from fluctuations in the energy measurements of QCD jets, partly as a result of the presence of large amounts of material in front of the calorimeters at the interface regions between different elements of the calorimetry. The main instrumental effects arise from the difference in response between the various calorimeter technologies used and from the fact that the e.m. calorimeter is highly non-compensating. The contribution of the EF to reducing the LVL1 missing transverse energy (E_T^{miss}) trigger rate should be significant for essentially three reasons. Firstly, accurate calorimeter calibration and inter-calibration are available. Secondly, the effect of using a different calibration for low-energy calorimeter cells and for cells outside clusters can be taken into account. Thirdly, the cell E_T cutoff applied to suppress the noise contribution can be tuned accurately. At present, it is not foreseen at LVL2 to recalculate E_T^{miss} using the full-granularity calorimeter data, because the large amount of data to be transferred in this case.

One could envisage, however, improving the LVL1 result by taking into account overflows in the LVL1 trigger towers.

Depending on the different requirements imposed by the physics analyses (either keeping the LVL1 nominal threshold or using the EF sharper threshold), a reduction factor ranging from 10 to 30 is achievable at the EF. The expected rates after the EF selection amount to 38 Hz for $E_{\mathrm{T}}^{\mathrm{miss}}60$ + j60 and 139 Hz for $E_{\mathrm{T}}^{\mathrm{miss}}50$ + j50.

7.1.3.4 Selection of jets

The high-level jet trigger is required to reduce the rate of events containing jets compared to LVL1 by improving the transverse-energy measurement, using refined energy calibration and jet definition. It is important to keep in mind that jets are the dominant high-p_{T} process, so rate reduction cannot be expected from removing fake jet objects, as is possible with, for example, electromagnetic clusters.

The expected reduction of the LVL1 jet rate after the HLT selection is approximately a factor of two. LVL2 reduces the total jet trigger rate by a factor of 1.8. Reconstructing jets at the EF level improves the total jet trigger rate compared to LVL2 by only 20–30%. No significant reduction is achieved unless the algorithm is tuned for lower efficiencies

For low luminosity the total rate expected is 500 Hz for the nominal thresholds j180, 3j75 and 4j55 used at LVL1. In order to obtain the reasonable total HLT jet output rate of 25 Hz, the nominal thresholds for the single-, three- and four-jet triggers have to be raised to the following values at low luminosity: j360, 3j150 and 4j100. Jets with lower thresholds will be accepted by pre-scaling of these events.

Therefore the thresholds for the jet triggers at LVL1 could be increased. The one exception could be the case of *b*-jets, for which the flexibility in the HLT scheme can be exploited further by making use of *b*-jet tagging at LVL2 and / or the EF. In particular, for topologies containing multi *b*-jets, the ability to separate, at LVL2 / EF, *b*-jets from light quark and gluon jets could lead to an increase of the acceptance for signal events, if the LVL1 E_{T} thresholds will be increased as explained.

7.2 Use of *b*-tagging with HLT

The possibility to efficiently identify high-p_{T} jets from b-quark fragmentation (b-tagging) has become a key requirement for current and future experiments. The b-tagging capability provided by the silicon detectors of the CDF and D0 experiment has been fundamental for the discovery of the top quark at Fermilab. As well all the four LEP experiment profits from *b*-tagging capability. This has played a key role in the observation of an excess of 3σ beyond the background expectation, consistent with the production of the SM Higgs boson with mass near 114 GeV [7-3]. Indeed much of this excess is seen in the four-jet final states ($ee \rightarrow ZH$ with $H \rightarrow b\bar{b}$ and $Z \rightarrow q\bar{q}$), whose reconstruction and analysis requires to tag the produced *b*-jets.

The b-tagging capability in ATLAS will be important for the observation of many important physics channels having final states involving *b*-quark:

- SM Higgs boson decay: if the signals observed at LEP will be confirmed, the Higgs boson at LHC will be observable only in the decay channel $H \rightarrow b\bar{b}$ (see Figure 1-19) from associated production $t\bar{t}H$ [7-4].
- MSSM Higgs bosons search [7-5]: possible channels are $h \rightarrow b\bar{b}$ decays from $t\bar{t}H$, $A/H \rightarrow Zh$ and $H \rightarrow hh \rightarrow b\bar{b}b\bar{b}$, and the decay of the charged Higgs bosons $H^{\pm} \rightarrow tb$. The *b*-tagging capability can be also used to suppress background: e.g. *bbH* and *bbA* productions with $A/H \rightarrow \tau\tau, \mu\mu$
- cascade decay of SUSY particles [7-6]
- top quark decays ($t \rightarrow Wb$) [7-7]

The multiplicity of *b*-quarks in these final states ranges from two to four, and the background from processes involving light quarks and gluon jets are very large. Furthermore these backgrounds often demand that all *b*-quarks present in the final state under study be identified. Therefore the observability of signals in multi b-jets final states requires a high rejection of non *b*-jets and the highest possible efficiency for *b*-jets identification.

As explained in the previous subsection the implementation of a *b*-tagging selection trigger at HLT (LVL2/EF) could provide additional flexibility and thus possibly better performance with respect to the standard strategy, especially if the LVL1 thresholds (Table 7-1) will be raised. Anyhow *b*-tagging algorithms can be implemented at EF first at all for event classification purposes: the labelling of events containing *b*-jets can reduce consistently the amount of data to be processed in the physics analysis phase.

The evaluation of the *b*-tagging performance, both for the final analysis and for the HLT selection and classification, require to software simulation of the detector. The ATLAS software simulation and reconstruction packages, based on the Fortran77 language and the GEANT 3 simulation package [7-13], are briefly summarized in the next Section.

7.3 The ATLAS simulation and reconstruction packages

7.3.1 ATLAS simulation

The ATLAS experiment is simulated using the DICE package (Detector Integration for a Collider Experiment [7-9]), which can be logically divided into three separate modules: event generation, detector simulation and digitisation. These three parts communicate through a set of ZEBRA [7-10] banks and can be run separately or in sequence. The framework for the simulation program is provided by a package called SLUG (Simulation for LHC Using GEANT). SLUG [7-11] provides the basic infrastructure for handling ZEBRA banks. It also provides a set of facilities for dealing with event generation, detector geometry and simulation, event merging for pile-up studies, together with stubs for user-defined routines to gain access to every step in the simulation process and tools for managing histograms and n-tuples

Event generation

The event generation phase is normally run separately in order to have a consistent input stream which can be used many times. Event generation facilities are implemented within SLUG by using the GENZ [7-12] package, which provides a common interface with the most widely used event generators (PYTHIA, HERWIG, ISAJET). The event generation includes the

creation of particles in the hard *pp* scattering process, the initial (ISR) and final (FSR) state radiation and the hadronisation processes. The complete particle level information is saved in the event history GENZ banks.

Detector simulation

The detector simulation part is the most time-consuming and critical; it can be run with different initial conditions on the same set of physics events in order to understand the impact of a change in the detector on the physics performance. The particle four-vectors are tracked through the various detector systems using the GEANT [7-13] package, which is used for the geometry description and the full detector simulation. All stable particles generated via GENZ and those which are created by interaction in the detector are stored in the KINE banks.

The HITS banks, instead, contain information hit positions (for tracking detectors) and energy losses (for calorimeters) and they provide the basis for the simulation of the detector response, which takes place at the digitisation step. The information collected in the HITS banks, although dependent on the geometry used by GEANT for event tracking, is nevertheless very general and does not contain any assumption on the detector read-out structure. HITS banks can be added together in order to simulate event superposition or pile-up. The contents of the HITS banks are the most valuable output of the detector simulation program, since most of the CPU time used goes into producing them.

Digitisation

The digitisation step is a second level of detector simulation, placed just at the interface with the reconstruction program. The physical information registered within the HITS bank is collected, re-processed in order to simulate the detector output, and eventually written out (in the DIGI banks) to be then used by the reconstruction programs. The output from the digitisation is obtained in a form similar to the one which might be expected from the read-out electronics in the real experiment. The digitisation step is very fast, except when pile-up at high luminosity is included. It is often therefore rerun as the first step in the reconstruction chain, if noise levels or single-channel efficiencies in some of the detectors are to be varied.

7.3.2 ATLAS reconstruction

The reconstruction of particles and other physics objects in the ATLAS detector is implemented in a single program named ATRECON [7-14], based on the SLUG framework. ATRECON is mostly written in Fortran77, using ZEBRA as memory manager, although some parts have already been rewritten in C++, but not with a fully object-oriented design. ATRECON runs on fully simulated GEANT 3.21 data.

The reconstruction proceeds in two stages. First, data from each detector is reconstructed in a stand-alone mode. Second, the information from all detectors is combined to get the most accurate measurements and identification of the final objects used in the analysis: photons, electrons, muons, τ-leptons, jets, b-jets, $E_{\mathrm{T}}^{\mathrm{miss}}$, primary vertex, etc. The output of the various algorithms can be stored in standardised ZEBRA banks and/or in a menu-driven combined n-tuple, which allows rapid checks, analysis in a PAW framework and comparisons between algorithms. This data represent the input for the various analysis-specific part: reconstruction of exclusive *B* decays, *W*'s, *Z*'s, top quark decays, Higgs bosons, SUSY particles, etc.

7.3.2.1 Stand-alone reconstruction

For the Inner Detector and the Muon Spectrometer, tracks are reconstructed from hits in the individual detector elements. In the Calorimeter System, cells with deposited energy are grouped together in cluster.

Reconstruction in the Inner Detector

Hit coordinates are reconstructed in the precision tracker and in the TRT. Tracks from charged particles are searched for. Three rather different algorithms with comparable performances have been developed:

- iPatRec [7-15] starts from a combinatorial search in the precision tracker;
- PixlRec [7-16] uses a track-following algorithm starting from the pixel B-layer outwards;
- xKalman [7-17] finds tracks in the TRT with a histogramming method and follows them inwards using Kalman-filtering techniques.

Track reconstruction can be performed over the full Inner Detector (except for iPatRec), or over a limited $\Delta\eta \times \Delta\phi$ range around 'seeds' found by the other detectors (electromagnetic clusters, jets, muons) and around Monte-Carlo truth information for checks. Seeds cannot be used for events like inclusive b production ($b\overline{b} \rightarrow \mu X$, i.e. events containing a muon with $p_T > 6$ GeV), where all tracks need to be reconstructed. The use of seeds in the reconstruction algorithm can save a factor of up to 100 in CPU time for events including the pile-up expected at high luminosity.

Reconstruction in the Calorimeter

Matrices containing the energies in all calorimeter cells are filled with the energies stored (DIGI banks) by the simulation code, after cell-to-cell calibration (Electronic and pile-up noise can be added at this stage). Then, starting from that matrices, jets are built following various algorithms with the cone algorithm used as a default. The E_T^{miss} vector is computed from the vector sum of the cell transverse energies. Electromagnetic clusters are reconstructed in the barrel and end-cap EM Calorimeters. Modulations in the measurement of the positions and the energies, due to the finite cell / cluster sizes or to the detector geometry, are corrected for by using the known correlations between shower position and biases in position and energy. Stand-alone electron / gamma identification is performed using shower-shape variables.

Reconstruction in the muon spectrometer

Muon track segments in the Muon System are found from a combinatorial search of the single-station track segments, followed by a fit using the hits. The tracking, performed in the highly non-homogeneous magnetic field, takes into account multiple scattering in the material of the apparatus. The output is a list of tracks with parameters extrapolated back to the interaction region, and also a list of track segments in the inner stations, possibly corresponding to low-p_T muons, which cannot be reconstructed with high efficiency in a stand-alone mode.

7.3.2.2 Combined reconstruction

In a second stage, information from several detectors is combined. Muons reconstructed in the Muon System are refined by matching the track to an Inner Detector track. This improves the momentum resolution, especially at moderate p_T, and yields accurate track parameters at the vertex. Lower-p_T (down to 2 GeV) muons are found by matching an Inner Detector track to the Tile Calorimeter cells.

Photon conversions and K_s^0 decays are searched for by pairing Inner Detector tracks. The primary vertex is reconstructed using all the tracks in the event. High-p_T (above 10 GeV) photon identification requires electromagnetic cluster shower-shape variables and the absence of reconstructed tracks in the Inner Detector, except for identified conversions. High-p_T electron identification requires a track reconstructed in the Inner Detector with transition-radiation hits and a measured momentum matching a calorimeter energy deposition compatible with an electromagnetic shower. Softer non-isolated electrons (down to $p_T \sim 1$ GeV) are identified by extrapolating Inner Detector tracks to the EM Calorimeter. Finally, τ-leptons are identified from narrow jets associated with a small number of charged tracks. Candidate *b*-jets are tagged by combining the impact parameter of high-quality tracks with soft electrons or muons.

7.4 b-tagging in the offline framework

The *b*-quarks give rise to *b*-hadrons, which are significantly different from all other particles. They have large mass, long lifetime, big decay multiplicity and takes more energy than light hadrons from the initial quarks. However, usually only its long lifetime is used for the tagging. For example, in the OPAL experiment the reconstructed secondary vertices are used as the signature of *b*-production [7-18], while in DELPHI the tracks with significant impact parameter, resulting from the large distance between the production and decay points of the B-hadrons, are exploited [7-19]. The large mass of *B*-hadrons was also included in the tagging procedure by the ALEPH experiments [7-20]. Another *b*-tagging methods is based on the identification of electron and muons with $p_T \geq 1 \div 2$ GeV inside jets (soft lepton tags). However the Tevatron and LEP experiments have shown experimentally that vertexing algorithms account for most of the *b*-tagging performance, while soft lepton tags contributes to a lesser extent, due to the limited fraction of *B*-hadron decays containing leptons and to the difficulty of efficiently identifying low-p_T lepton.

The *b*-tagging procedure used in ATLAS is based on a vertexing algorithm, which gives the possibility of constructing one tagging discriminating variable from all the tracks reconstructed in a jet. The probability methods is tuned comparing the signal *b*-jets to a background from so-called prompt jets, which by definition do not contain any long lived particles.

7.4.1 The event samples for b-tagging study

The *b*-tagging capability of the ATLAS Inner Detector and the *b*-tagging algorithm performance was tested using events simulating the hadronic decay of Higgs bosons from associated production ($pp \rightarrow WH + X$ and $W \rightarrow \mu\nu$). The signal sample consists of $H \rightarrow b\bar{b}$ events while the background samples are composed of $H \rightarrow c\bar{c}$, $H \rightarrow u\bar{u}$ and $H \rightarrow gg$ events. While these background processes actually have negligible rates, the decays are considered representative of actual backgrounds which will be encountered at the LHC, and, by generating them from Higgs decays, direct comparisons can be made with background jets having the same kinematics as the b-jets. The events, generated using the PYTHIA 5.7 Monte Carlo generator, are required to have hard partons (before Final States Radiation) from the Higgs decay with $p_T > 15$ GeV and $|\eta| < 2.5$. They are grouped in two samples concerning two different value of the Higgs mass. A sample of about 20000 events for the signal and each background processes were generated for $m_H = 100$ GeV. For $m_H = 400$ GeV the sample are composed of 10000 events for each channel.As shown in Figures 7-2 and 7-3, these sample allows to cover with sufficient statistics the p_T range between ~ 15 and ~ 400 GeV and the full pseudorapidity range $|\eta| < 2.5$.

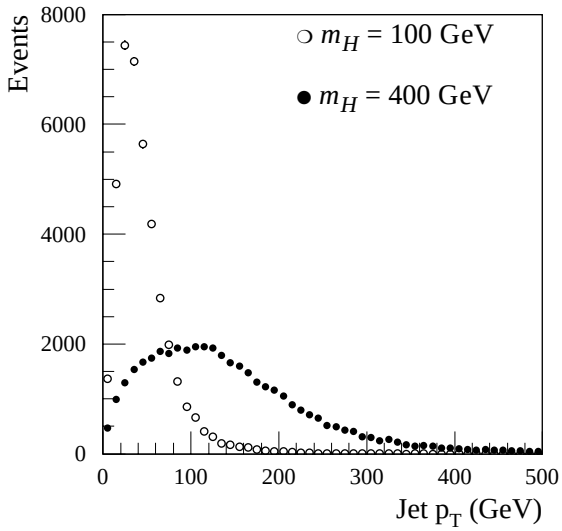

Figure 7-2 Transverse momentum distribution of b-jets from $H \rightarrow b\bar{b}$ decay.

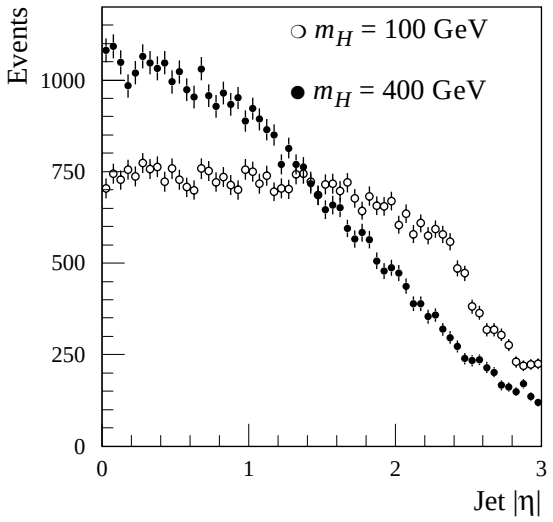

Figure 7-3 Pseudorapidity distribution of b-jets from $H \rightarrow b\bar{b}$ decay.

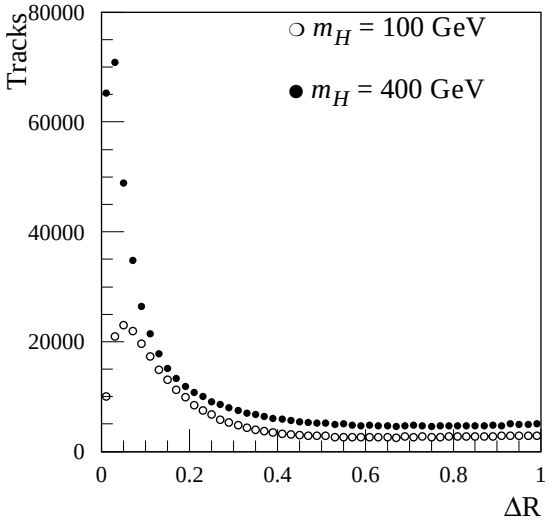

Figure 7-4 Distance ΔR of all charged particles with $p_T > 1$ GeV to the b-jet direction in $H \rightarrow b\bar{b}$ decay.

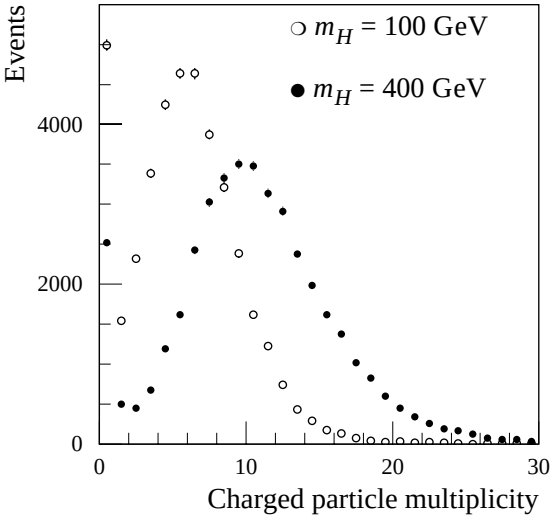

Figure 7-5 Multiplicity of charged particles with $p_T > 1$ GeV in a cone $\Delta R < 0.4$ around the b-jet direction in $H \rightarrow b\bar{b}$ decay.

The reconstruction efficiency, and hence the *b*-tagging performance, depends from the spatial density of the tracks inside jets; Figure 7-4 shows the distance ΔR of charged particles to the b-quark direction[1] in η–ϕ. For both the sample 99% of charged particles produced from b-quark

1. The b-jet direction and b-quark direction can differ. b-quarks in the final state of an interaction or a decay can radiate gluons (Final State Radiation, FSR) and therefore change the direction. As, in practice, the reconstruction of the jet direction and energy is done using the energy deposited in the calorimeters, the difference between the jet direction as measured by the calorimeters and the b-quark direction has been checked. The resolution of the b-jet direction is measured much better than the standard cone size $\Delta R < 0.4$, therefore the use of a finite cone size does not cause significant loss of tracks.

fragmentation are found in a cone of $\Delta R < 0.4$, which is the standard cone size used by the calorimeter reconstruction software. The charged multiplicity in this cone is shown in Figure 7-5. For m_H = 100 GeV the mean charged multiplicity is 5.5, 60% of which come from daughters of the B-hadron decay, and in 12.5% of the jets there are no charged particles with $p_T > 1$ in the cone. In this case, therefore, it is impossible to establish the jet flavour. For m_H = 400 GeV the multiplicity average rises to 10, 40% of which come from daughters of the B-hadron decay, while the percentage of jets without reconstructed charged particles decreases to 6.3%.

The *b*-tagging methods described below requires to reconstruct accurately at least four tracks arising from *B*-hadron decays. The transverse momentum of these particles are shown in Figure 7-6. The average p_T of particles with $p_T > 1$ GeV varies from 4.9 GeV (in the low-m_H sample) to 8.7 GeV (for the other one). In both cases 70% of particles have p_T less than the averages. Such a low-p_T particles must be reconstructed inside densely populated jets in a region where multiple scattering dominates the impact parameter resolution.

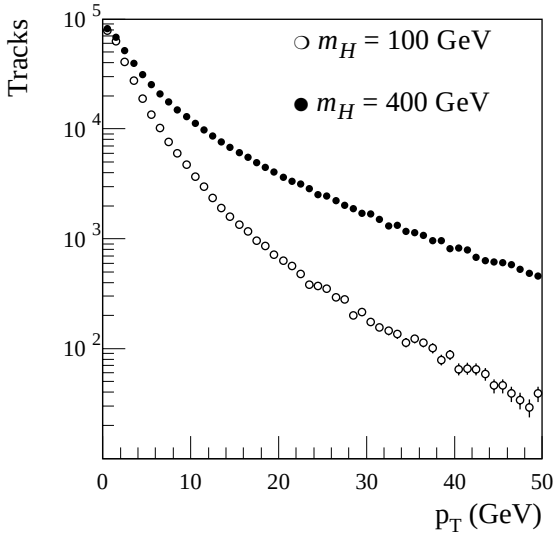

Figure 7-6 Transverse momentum distribution of charged particles in a b-jet from $H \rightarrow b\bar{b}$ decay

7.4.2 Reconstruction procedure

The reconstruction procedure has to be as realistic as possible, therefore, obviously, no use was made of the generated event information. A jet in the $H \rightarrow b\bar{b}$ events was labelled as a *b*-jet if a *b*-quark from the original hard process pointed, after final state radiation, along the jet within an angular distance of $\Delta R < 0.2$. Similarly, jets were labelled as for *u*-jets, *c*-jets and gluon jets for the other data samples. The procedure is the following:

- First jets were identified as clusters in the combined electromagnetic and hadronic calorimeters, using a noise threshold of 2σ on each cell and a p_T threshold of 10 GeV for the reconstructed jets.
- Then the Inner Detector pattern recognition programs iPatRec or xKalman were run, using as seeds the reconstructed jet directions and a road half-width of 0.5; the p_T threshold for reconstructed tracks was set to 0.7 GeV.
- Finally the xConver algorithm [7-21] was run in order to identify, and reject, electrons coming from photon conversions.
- The final analysis (the calculation of the *b*-tagging variable) is performed examining the Combined Physics N-Tuples (CBNTs), which provide an easy access to both basic quantities from all ATLAS sub-detector (tracks, electromagnetic clusters, jets, muon,...) and to more elaborate quantities combining information from several detectors. The *b*-tagging method has been developed and tuned analysing the CBNT, this procedure avoids to re-run the reconstruction programs (ATRECON).

7.4.3 Track selection

Track finding was performed in restricted cones using iPatRec. Only tracks with $p_T > 0.7$ GeV and in a cone of $\Delta R < 0.4$ around the jet direction were considered. The efficiency for finding any primary charged track in a b-jet was 92.8%.

A set of cuts has been developed in [7-22] in order to optimize the *b*-tagging performance. These rely in particular on the *B*-layer itself to obtain the best possible impact parameter resolution and reject tracks with impact parameters which are too large. To understand and hence improve the quality of the reconstruction, the reconstructed track parameters have been compared with those from the original Monte Carlo tracks.

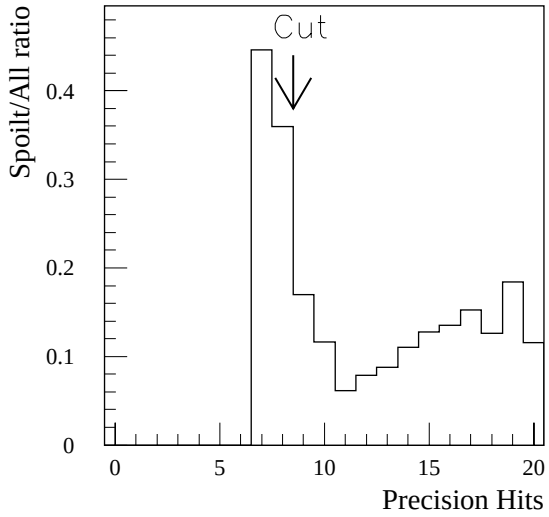

Figure 7-7 Ratio of number of precision hits for spoilt tracks compared with all tracks.

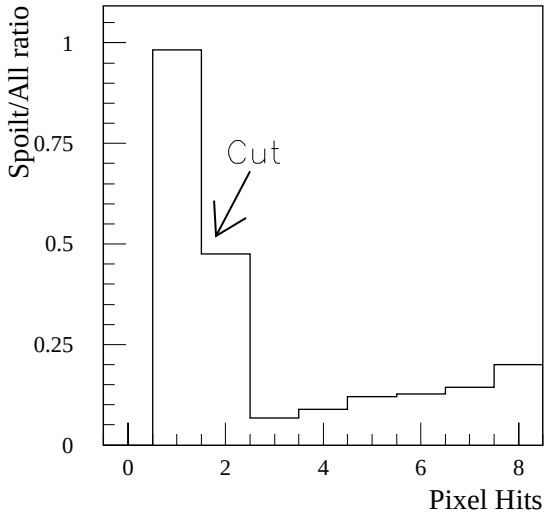

Figure 7-8 Ratio of number of pixel hits for spoilt tracks compared with all tracks.

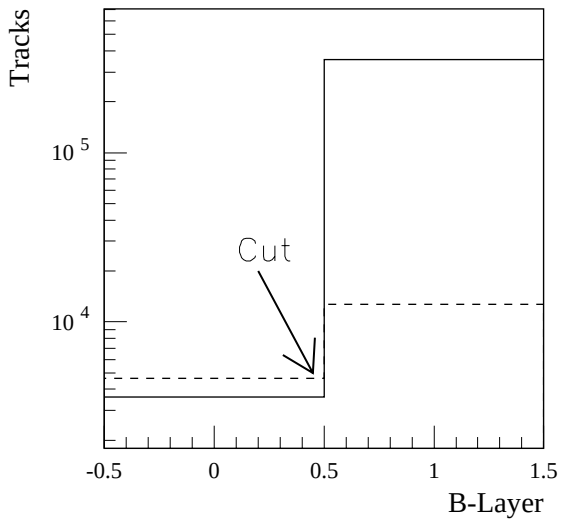

Figure 7-9 Presence of hits on the B-layer for the primary (solid line) and secondary (dotted line) tracks.

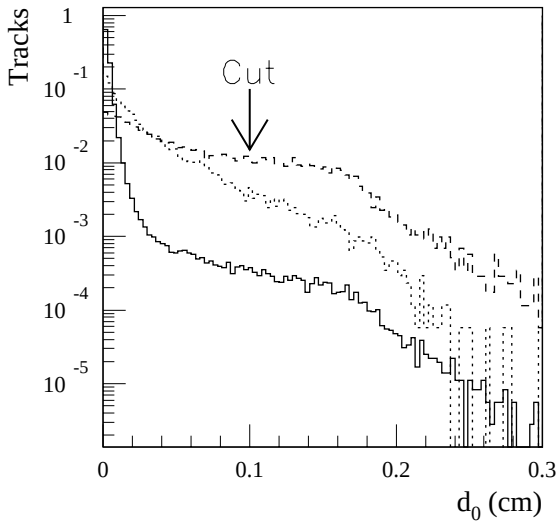

Figure 7-10 Impact parameter distribution of primary (solid), B-tracks (dotted) and secondaries (dashed)

The ratios of the number of precision layer hits and number of pixel hits for spoilt[1] tracks compared with all tracks are shown in Figures 7-7 and 7-8. As shown in Figure 7-9, only 1% of the primary tracks (solid line) do not have an associated B-layer hit, in comparison to 27% of the secondary tracks (dotted line). Finally, Figure 7-10 shows that only 0.7% of the primary tracks (solid line) have $|d_0| > 1$ mm, in comparison to 30% of the secondary tracks (dashed line) and to 5% of the tracks produced in a B-vertex (dotted line).

From these distributions, the following selection cuts (called 'standard cuts') have been derived:

- Number of precision hits ≥ 9.
- Number of pixel hits ≥ 2.
- At least one associated hit in the B-layer.
- $|d_0| \leq 1$ mm

After all these cuts the track finding efficiencies for tracks from the primary vertex is 88.2% (see Figure 7-11). The main limitation to the *b*-tagging performance arises from secondary tracks, i.e. tracks originating from interactions of primary tracks with detector material. Secondary tracks can only be rejected by applying appropriate cuts on track quality. These additional cuts are:

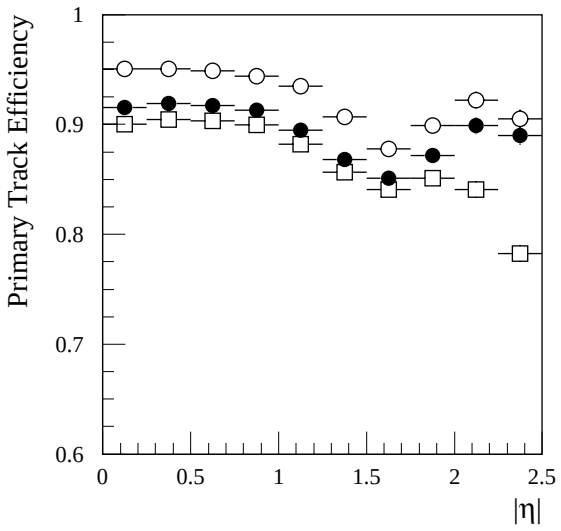

Figure 7-11 Primary track efficiency for iPatRec as a function of pseudorapidity without cuts (open circles), after applying standard cuts (closed circles) and after

- impact parameter in the *R-z* plane relative to the reconstructed primary interaction point < 1.5 / sin(θ) mm; the cut is only important at high luminosity, when it allows to reject tracks coming from the other ~20 primary vertexes generated by the superimposed minimum bias events;
- track fit χ^2 / d.o.f. < 3;
- at most one hit in the Pixels shared with another reconstructed track;
- at most two hits in the precision detectors (pixel and SCT) shared with another reconstructed track.

After these cuts 37% of the secondaries were electrons from conversions and these could be removed using the program xConver, which identify electrons from conversions. Tracks were rejected if they had at least 5% high-threshold hits in the TRT (or no TRT hits at all, which happens in case of hard bremsstrahlung) and formed a good conversion candidate ($\chi^2 < 50$ for the conversion fit and conversion radius > 2 cm) with another track in the same jet. This cut removed 15% of the secondary tracks without affecting the track-finding efficiency for primary tracks.

As an example of the reconstruction performance using these cuts, Figures 7-12 and 7-13 show the primary track finding efficiency and the fraction of secondary and fake tracks in the final track sample. The quality cuts tend to reduce the track sample, especially for $|\eta| > 2$ (see

1. Reconstructed tracks whose matching χ^2 probability (compared to the original Monte Carlo tracks) is greater than 0.04 are classified as spoilt tracks.

7-11), without degrading substantially the *b*-tagging performance, which is limited much more by the amount of secondaries than that of fakes. The quality cuts effectively select a smaller but better-measured sample of tracks.

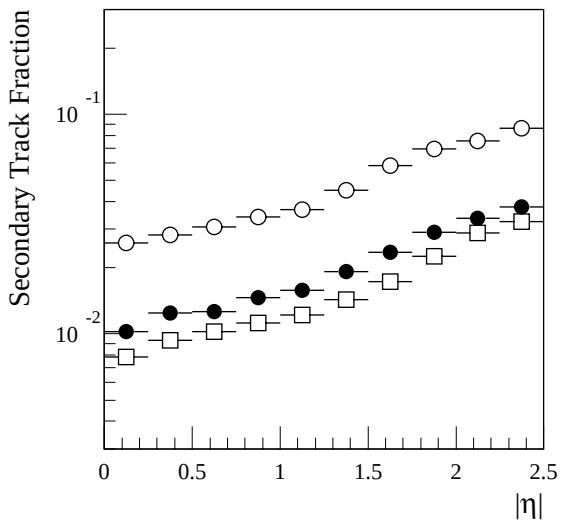

Figure 7-12 Fraction of secondary for iPatRec tracks as a function of pseudorapidity without cuts (open circles), after applying standard cuts (closed circles) and after applying quality cuts (open squares).

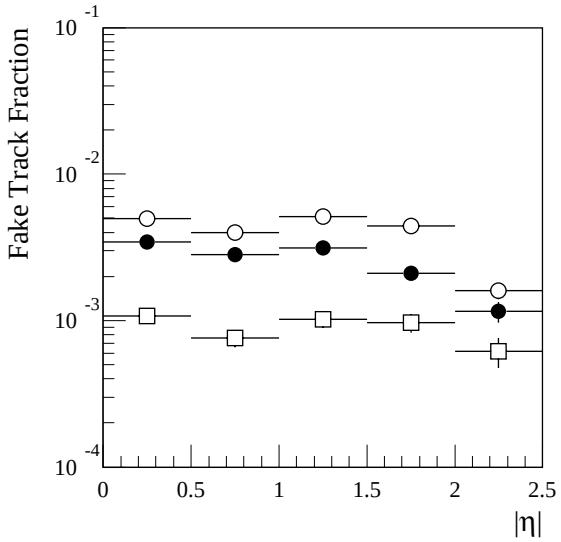

Figure 7-13 Fraction of fake tracks for iPatRec as a function of pseudorapidity without cuts (open circles), after applying standard cuts (closed circles) and after applying quality cuts (open squares).

7.4.4 b-tagging methodology

The rejection of non *b*-jets is dependent on the fact that for light quarks most of the stable particles, which can be reconstructed in the Inner Detector, come from the decays of short-lived objects and hence appear to come from the primary vertex, while the relatively long lifetimes (around 1.5 ps) of *b*-hadrons give rise to experimentally observable displaced vertices ($c\tau \approx 470\ \mu m$). These displaced vertices can be tagged by either explicitly reconstructing the vertex or by examining the impact parameter in the transverse plane (d_0) of the reconstructed tracks.

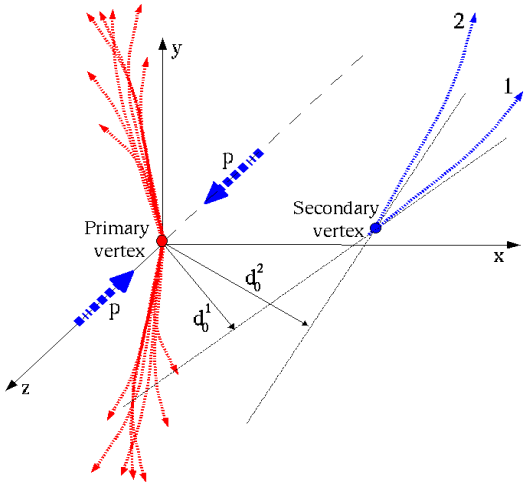

Figure 7-14 Track displacement relative to the primary vertex is measured using the perpendicular distance of closest approach d_0

The variable d_0 is defined as the minimum distance between the track trajectory and the primary interaction point (primary vertex) in the plane perpendicular to the beam direction (see Figure 7-14). The sign of d_0 is defined as positive if the vector joining the primary vertex and the point of closest approach of the track lies in the same direction as the jet to which the given track belongs. The jet axis is determined accurately from the calorimeter. From such a definition the tracks from the decays of *B* hadrons have positive impact parameters, whereas non

zero impact parameters arising from inaccurate reconstruction of particles' trajectories are equally likely to be positive or negative. Therefore the use of tracks with positive impact parameters increases the tagging efficiency

.The efficiency of the method is determined by the impact parameter resolution $\sigma(d_0)$ of the detector, by the rate of conversions and nuclear interactions in the detector material and by the frequency of pattern recognition errors. As shown in Figure 7-15 the impact parameter resolution for primary pions in jets from $H \rightarrow b\bar{b}$ events varies between ~20 μm for p_T <10 GeV to ~40 μm for low-p_T tracks ($3 < p_T < 6$ GeV). The rise with $|\eta|$ is mainly due to the increase in material in the forward direction.

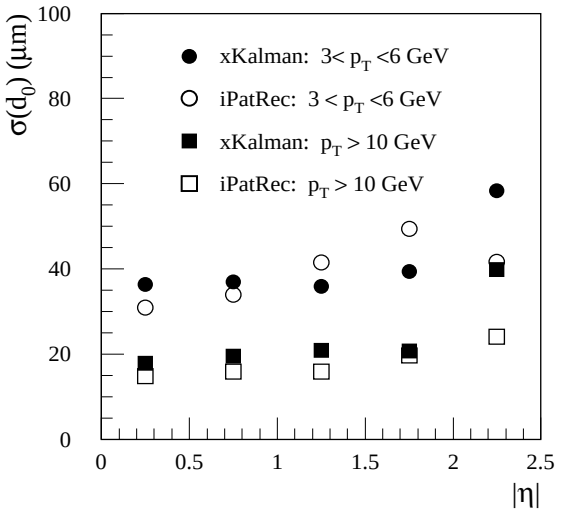

Figure 7-15 Impact parameter resolution (*rms* value) for primary pions reconstructed in b-jets from H(100 GeV) $\rightarrow b\bar{b}$ decay.

The d_0 variable is determined with respect to nominal beam position in the transverse plane, which will be measured with high accuracy as a function of time. In principle, the *x* and *y* position of the primary vertex can be reconstructed event by event from prompt tracks, but the gain is small because the spread of the beam-spot is only 15 μm. Therefore the uncertainty on the impact parameter is the combination of the measurement error $\sigma(d_0)$ and the spread of the beam-spot taken in quadrature. The effect of this is to increase $\sigma(d_0)$ by about 4 μm.

The *z* coordinate of the primary interaction point can be reconstructed simply by taking the truncated weighted average of the z_0 coordinates of all well-reconstructed tracks in the event which have a transverse impact parameter $d_0 < 2 \cdot \sigma(d_0)$. This simple and fast method achieves a resolution ~35 μ*m* for events with at least four reconstructed tracks. However this information is used only at high luminosity to reject tracks arising from the superimposed minimum bias events. At low luminosity the algorithm do not use any *z*-information for vertexing and do not explicitly reconstruct secondary vertices, even in the transverse plane.

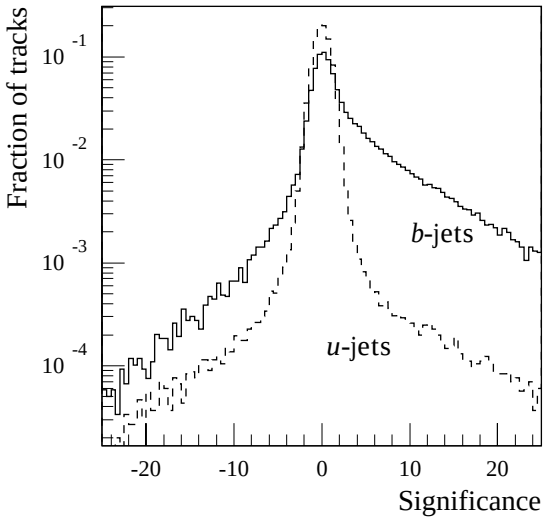

Figure 7-16 Significance distribution ($S=d_0/\sigma(d_0)$) for reconstructed tracks in *b*-jets and *u*-jets. The distributions are normalised to the same area.

To correctly weight the different d_0 measures, the b-tagging analysis is based on the significance *S* of this variable: i.e. the ratio of the signed impact parameter of the track to its total error. Figure 7-16 shows a comparison of the significance distribution for tracks reconstructed in *b*-jet and *u*-jets.

Both distributions have significant ‘cores’ which can be described by a gaussian of width close to one reflecting the detector resolution and therefore represent correctly reconstructed tracks coming from the primary vertex.

The *b*-jets contain tracks with large positive significance, corresponding to genuine lifetime content. By contrast, light quarks and gluon jets have only a small excess of tracks which appear to contain lifetime arising from:

- interactions with material, e.g. photons converting to e^+e^- pairs or pions having nuclear interactions;
- daughters of V^0's;
- daughters of heavy quarks formed in the fragmentation (relevant only for gluon jets).

In the ideal case the negative part of the significance distribution should have a gaussian shape; in practice there is a long non gaussian tail and depends significantly on the criteria which are used for the selection of tracks and events. For the *b*-jets, the tail comes mainly from an incorrect determination of the sign (corresponding to the uncertainty in determining the b-hadron direction) and from decays of charmed states. For the *u*-jets is it largely due to tracks measured wrongly by the tracking system and partly to particles from the secondary decay and interaction.

7.4.5 Likelihood ratio method

The likelihood ratio method is the technique which was proven to offer the best b-tagging performance in the ATLAS framework. The method is based on the comparison between the significance probability distribution functions of signal ($f_b(S)$) and background ($f_u(S)$, $f_c(S)$ and $f_g(S)$). These probability distribution functions, shown in Figure 7-17, are obtained fitting the corresponding significance distributions of the different jets sample.

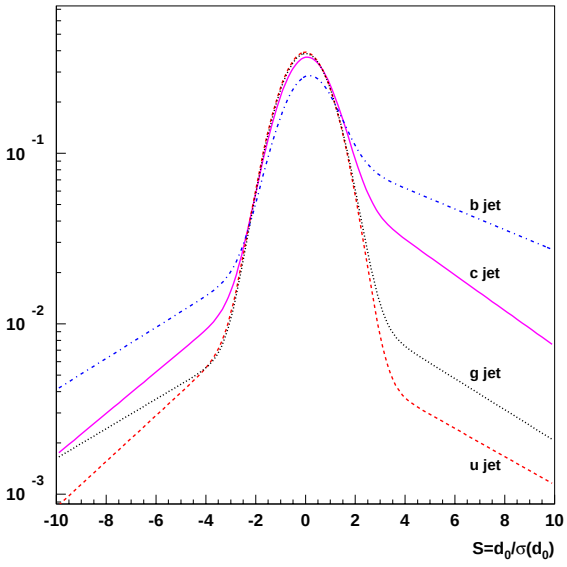

Figure 7-17 Probability distribution functions for signal (jets from $H \rightarrow b\bar{b}$ decay) and background.

The emphasis is on individual jets, rather than complete events. For each jet seeded by the calorimeter a *b*-jet weight variable *W* is calculated as follows:

1. For each selected track *i* in the jet, the significance S_i was calculated.
2. The ratio of the values of the significance probability distribution functions for *b*-jets and *u*-jets was computed: $r_i = f_b(S_i)/f_u(S_i)$.
3. A jet weight was constructed from the sum of the logarithms of the ratios: $W = \Sigma \, log \, r_i$. The product of the r_i values allows to construct a variable independent from the total number of considered tracks.

The jets are *b*-tagged if the calculated *W* is grater than a threshold value, which defines the efficiency ε_b for keeping b-jets and the corresponding rejection R_j (defined as the inverse of the

efficiency ε_j) for different background jets *j*. By using the significance distribution $f_u(S_i)$, the method was optimized for the rejection of *u*-jets. In the case of real data, since the jet type will not be a priori known, the rejection will have to be optimized for each specific background under study.

7.4.6 Results

The jets weights calculated with the likelihood ratio methods are shown in Figure 7-18 for *u*-jets from $H \to u\bar{u}$ decay and *b*-jets from $H \to b\bar{b}$. As all the relevant distributions are compatible in the p_T range where the two data sets overlap, in this analysis the jets from the decay of 100 and 400 GeV Higgs are used together.

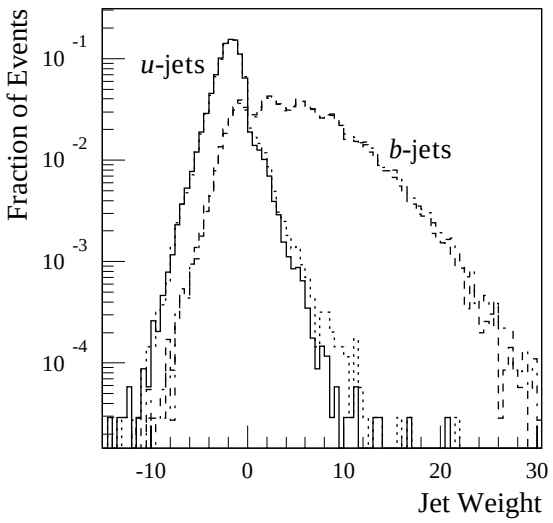

Figure 7-18 Jets weights from likelihood ratio.

The following background compositions can be deduced for iPatRec:

- 21% of tracks arise from interaction in the material of the detector (32% of these tracks are electrons from photon conversions, the other particles are products of nuclear interactions).
- 21% of tracks are produced in the decays of hadrons with significant life-time (mainly K_s^0); hadrons containing heavy quarks are produced in 1.7% of *u*-jets.
- 58% of tracks are produced at the primary vertex, with large deflections from multiple scattering or possibly pattern recognition problems.

From Figure 7-18 it is clear that, varying the cut on the jet weights, it is possible to select samples with different relative composition of *u* and *b*-jets. Increasing the *W*-cut, the purity of the selected *b*-sample grows (the rejection versus *u*-jets $R_u = 1/\varepsilon_b$ grows), but clearly ε_b is reduced. The curves of R_u, R_c and R_g rejections as a function of *b*-tag efficiency obtained varying the *W*-cut is shows in Figure 7-19. The rejection of *c*-jets is limited by the non negligible lifetime of charmed hadrons: for $D^{\pm}$ $c\tau$ = 317 μm while for D^0 $c\tau$ = 124 μm. The rejection of gluon jets is instead limited by gluon splitting: $BR(g \to cc)$ = 6% and $BR(g \to bb)$ = 4% for m_H = 400 GeV.

The rejection for two different reference values of b-tag efficiency, which are used in the analysis of many channel, are shown in Table 7-2. For m_H = 100 GeV this *b*-tagging procedure allows to tag the *b*-jets with an efficiency of 60%, accepting only one *u*-jets in more than a hundred.

7.5 *b*-tagging at LVL2

The use of *b*-tagging at LVL2 requires fast track reconstruction and therefore precludes the use of the standard EF/offline track-reconstruction packages. Furthermore the complete track reconstruction and track fit using all the information from the Inner Detector requires a computing power and data volume to be accessed, which is not negligible compared to the LVL2 laten-

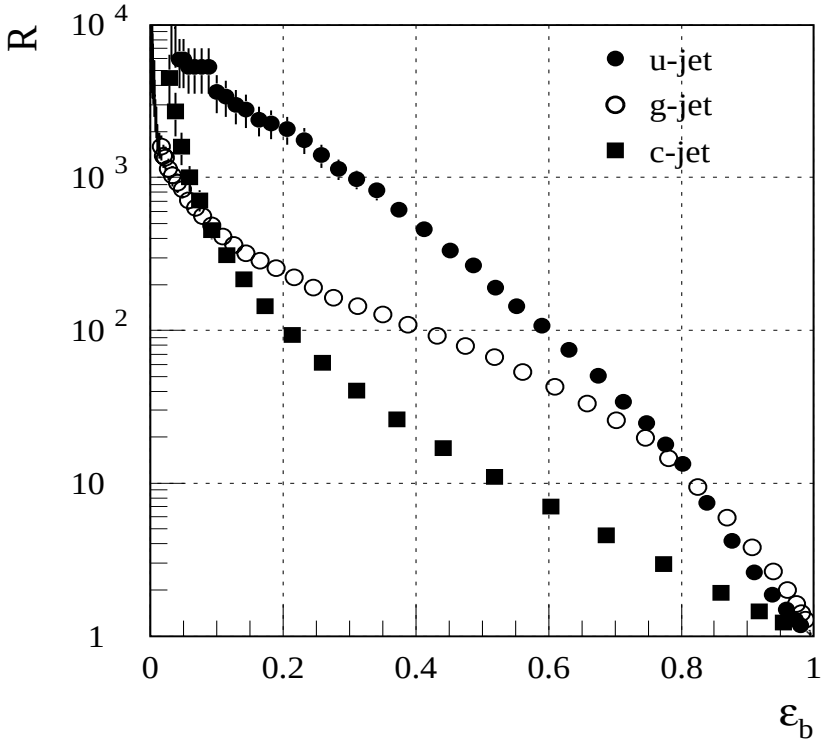

Figure 7-19 Background rejection as a function of *b*-jet efficiency obtained with iPatRec.

Table 7-2 Rejection of different types of background jets for *b*-tag efficiency of 50% and 60%.

	ε_b = 50%		ε_b = 60%	
	m_H = 100 GeV	m_H = 400 GeV	m_H = 100 GeV	m_H = 400 GeV
R_u	391 ± 49	183 ± 17	148 ± 11	80 ± 5
R_g	148 ± 14	58 ± 3	74 ± 5	37 ± 1
R_c	11.7 ± 0.3	13.4 ± 0.4	6.9 ± 0.1	7.3 ± 0.2

cy and resources. Therefore, the track impact parameter in the transverse plane are obtained using an algorithm based solely on the pixel system (PixTrig [7-23]), which benefit from the direct availability of the three space-points over $|\eta| < 2.5$.

The *b*-tagging selection proceeds as follows. PixTrig performs a three-dimensional track reconstruction in the RoIs defined by each relevant LVL1 jet trigger, using the pixel-hit clusters contained in a region $(\Delta\phi, \Delta\eta) < (0.5, 0.5)$ around the RoI axis. Every possible triplet of points in different layers of the pixel detector, compatible with a track of $p_T > 2$ GeV, is formed. Ambiguities are removed on the basis of the track quality; then, for each track, the value of d_0 is calculated together with its error.

The *b*-tagging is based on the likelihood ratio method and is very similar to the off-line one: the ratio of the probability densities for the track to come from a *b*-jet or a *u*-jet is calculated: $f_b(S_i) / f_u(S_i)$ and the product W of these ratios over all reconstructed tracks in the jet is computed. The final tagging variable, slight different from the one used in the offline algorithm, is $X = W / (1 + W)$*:* Jets are tagged as *b*-jets if $X \sim 1$ and *u*-jets if $X \sim 0$.

The *b*-tagging algorithm has been characterized on the same event sample used for the offline studies and described in Section 7.4.1 ($H \rightarrow b\bar{b}/u\bar{u}$ decays with m_H = 100 GeV). The efficiencies (ε_b^{LVL2}) for *b*-jets versus the rejection factors (R_u) against u-jets are shown Figure 7-20. The processing time has been measured on a Pentium III (500 MHz) running Linux and found to be on average 4.4 ms per jet, dominated by the time needed for the pattern-recognition stage of the track reconstruction.

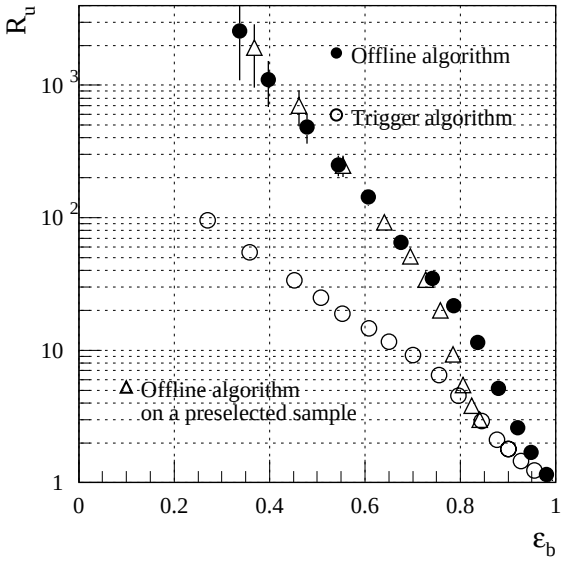

Figure 7-20 Rejection R_u against u-jets as a function of the b-jet efficiency ε_b for $H \rightarrow b\bar{b}/u\bar{u}$ decays with m_H = 100 GeV at low luminosity as obtained for the trigger and offline algorithms

The performance of the LVL2 trigger algorithm has been directly compared to that of the offline algorithm. As an example, a trigger selection corresponding to $R_u = 3$ and $\varepsilon_b^{LVL2} = 85\%$ has been applied as a first step to the *b*-jets and *u*-jets sample. For jets selected in this way, the offline selection was then applied and the results are shown in the same Figure. It is found that the full offline performance is restored for a final b-tagging efficiency $\varepsilon_b < 70\%$. This demonstrates that the trigger and offline selection are well correlated and that, as long as the LVL2 efficiency is kept above 80%, it is possible to provide subsequent analyses with an unbiased jet sample in the region $\varepsilon_b < 70\%$. Different combinations of working points of LVL2 trigger selection and offline analysis could be chosen depending on the required offline b-tagging efficiency; examples of which can be found in Ref. [7-24].

7.6 *b*-tagging at EF

7.6.1 Algorithm and performance

As explained in the previous chapters no development of specific algorithms is foreseen at EF. The filtering program must use the offline reconstruction and physics analysis code, adapted to suit the particular needs of the EF environment: perform event classification and selection without introducing biases on the recorded data sample.

Therefore the *b*-tagging studies at EF consist mainly in adapting the offline code, in order to optimize the performance to processing-time ratio. The basic approach is based on dropping the less relevant and / or time consuming part of the offline procedure without modifying the code. Indeed, to facilitate this interplay between EF and offline the new object oriented ATLAS simulation framework, currently under development, has a modular design to allow a selective activation of the different modules.

Table 7-3 shows the execution times (in seconds for a PA8000 180 MHz processor rating 7 SPECint95) in the different steps for $H \rightarrow b\bar{b}$ event reconstruction from WH associated production. For m_H=100 GeV the total execution time is 2.5 s, while the subsequent calculation of the b-tagging variable (from the analysis of the CBNT) requires only a negligible amount of time (only few hundredth of seconds). Therefore the execution time is dominated by track reconstruction. The iPatRec execution time is already low (1.3 s for m_H=100 GeV) but can be more than halved without significant loss of performance.

Table 7-3 Reconstruction time for $H \rightarrow bb$ event (from WH associated production) on a HP 9000/879 rating 7 SPECInt95.

	m_H=100 GeV	m_H=400 GeV
Calorec	0.9 s	1.2
iPatRec	1.3 s	4.0
xConver	0.3 s	1.4
Total	2.5 s	6.6

The impact parameter resolution is dominated by central tracker and especially by the first pixel layer (B-layer) located at 4 cm from the interaction point. Therefore the TRT data, which through the measurements of the track curvature can improve the d_0 resolution, can not be fundamental. Only iPatrec allows the deactivation of the TRT code in the reconstruction, while in xKalman this operation is impossible because all the algorithm is based on the TRT measurements. For this reason xKalman was not used in the analysis[1]. However turning off the TRT code from the inner detector reconstruction removes the possibility to identify and remove electrons from photon conversion via the use of the xConver program. Nevertheless the TRT module deactivation and the removal of the xConver program from the reconstruction procedure do not significantly change the b-tagging performance. For m_H =100 GeV the rejection versus u-jets decreases of less than 10%, while the processing time are reduced by the 50%, from 1.6 s (iPatrec plus xConver in Table 7-3) to 0.81 s (see Figures 7-21 and 7-22)

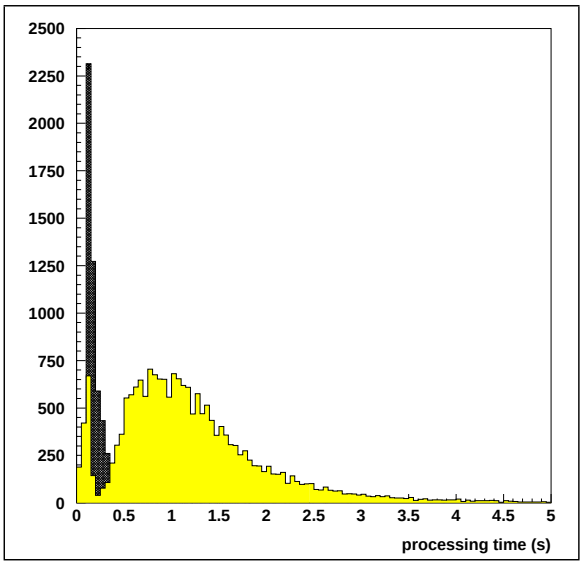

Figure 7-21 Reconstruction processing time (iPatRec) with the TRT module active. The black histogram is to the xConver contribute.

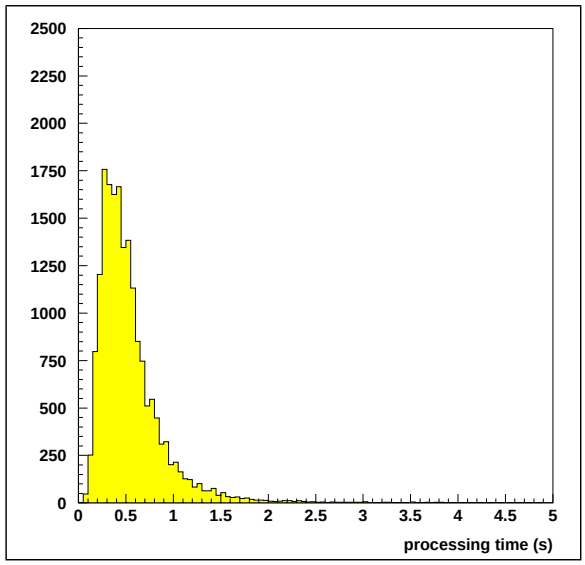

Figure 7-22 *Reconstruction processing time (iPatRec) after the removal of the TRT module.

1. The new object oriented version of the xKalman program (xKalman++) allows to deactivate the TRT module.

The processing time can be further reduced increasing the p_T threshold for tracks recon-

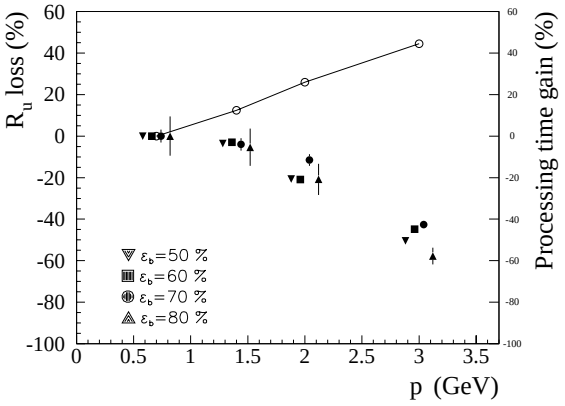

Figure 7-23 Percentage R_u loss versus the p_T reconstruction threshold. The line represent the corresponding processing-time gain.

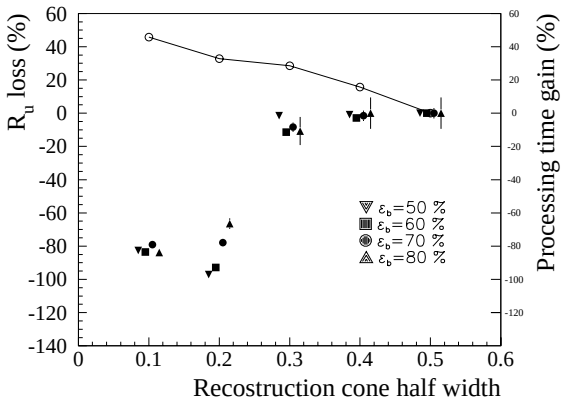

Figure 7-24 Percentage R_u loss versus the reconstruction cone width. The line represent the corresponding processing-time gain.

struction in iPatrec and reducing the half width of the reconstruction cone. Figures 7-23 and 7-24 show the processing time gain (in per cent) and the corresponding loss in rejection factor versus *u*-jet obtained varying the above cuts. The of the p_T threshold from the nominal value of 0.7 GeV to 1.5 GeV do not degrade the performance, because tracks with lower transverse momentum do not contribute importantly to the construction of the discriminating variable (*W*). However the *u*-jet rejection rapidly falls with the raising of the threshold. Figure 7-24 reveals that the performance are somewhat stable down to $\Delta\eta = \Delta\phi = 0.3$. Therefore setting p_T =1.5 GeV and $\Delta\eta = \Delta\phi$=0.3 it is possible to reduce the processing time to 0.5 s (less than 1/3 of the offline reconstruction time) without compromise the *b*-tagging results. A possible future improvement (i.e. reduction of processing time without significantly loss of performance) is the optimization of the p_T threshold and reconstruction cone half width as a function of the p_T and $|\eta|$ values of the jets seed.

Table 7-4 Comparison between the b-tagging performance of the offline and the EF.

	R_u		R_c	
	Offline	**EF**	**Offline**	**EF**
ε_b **= 50%**	362 ± 36	342± 33	9.6 ± 0.2	9.14 ± 0.2
ε_b **= 60%**	134 ± 8	124 ± 1	6.3 ± 0.2	6.1 ± 0.1
ε_b **= 70%**	45.1 ± 1.6	41.3 ± 1.4	3.92 ± 0.05	3.80 ± 0.05
ε_b **= 80%**	14.9 ± 0.3	13.3 ± 0.3	2.48 ± 0.02	2.44 ± 0.02

Table 7-4 shows the comparison between off-line and EF final performance for the rejection of *u*-jets and *c*-jets, which exhibits a similar behaviour in the performance / processing-task optimization. As shown also in Figure 7-25, after these simplification the EF *b*-tagging performance is very similar to the off-line results.

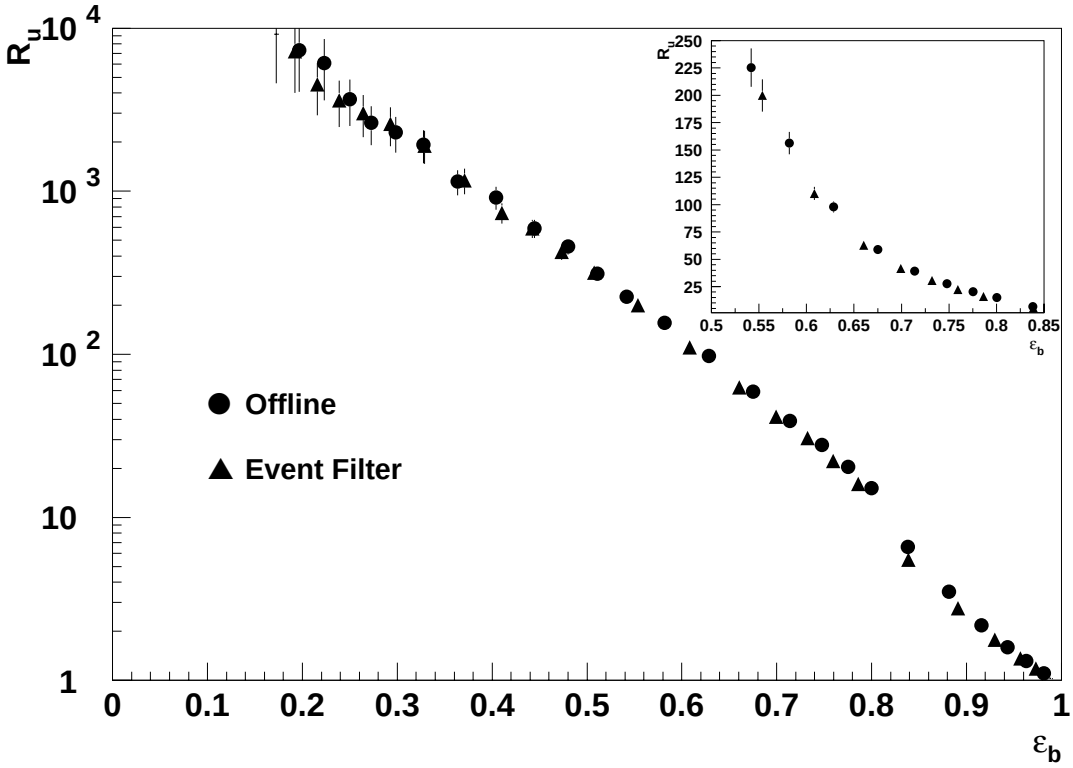

Figure 7-25 b-tagging performance comparison between off-line and EF.

7.6.2 b-tagging classification

The analysis of many physical channels requires the identification of *b*-quarks and most of them lead to final states containing several *b*-jets. The main drawbacks of these channels are the enormous background from QCD multi-jet production and the low production rate. Given the performance comparable with the off-line one and the negligible contribution to the global EF latency, this *b*-tagging algorithm can be applied at EF to label events containing *b*-jets improving the selectivity of the event classification. This reduces considerably the amount of data to be examined in the subsequent off-line analysis step.

The one example is the $H \rightarrow b\bar{b}$ channel from $t\bar{t}H$ associated production, which is the most reliable one in the mass range indicated by the LEP data. The channel is triggered by an isolated lepton from the semileptonic decay of one of the top quarks ($t \rightarrow Wb$ with $W \rightarrow l\nu$). This provides a final state topology with one isolated lepton and at least four reconstructed jets originating from the *b*-quarks or the hadronic decay of the second *W* boson. Either three or four of these reconstructed jets are required to be identified as *b*-jets by the off-line selection procedure. The Higgs signal would appear as a peak in the invariant *bb* mass distribution, above the combinatorial background from the signal itself and from the various background processes. The reducible backgrounds contain jets miss-identified as *b*-jets, such as ttjj, Wjjj, Wjbb. Their magnitude depends on the quality of the *b*-tagging performance.

Table 7-5 from [7-25] shows the expected number of signal and background events collected during the initial low-luminosity phase as a function of sequential steps taken in the off-line reconstruction algorithm. The number are obtained assuming a b-tagging efficiency of 60% and a corresponding rejection versus *u*-jets and *c*-jets of 100 and 10 respectively (The actual EF algo-

rithm the measured rejection for $\varepsilon_b = 60\%$ are $R_u = 124 \pm 7$ and $R_c = 6.1 \pm 0.1$). After the first selection based on the trigger lepton and at least 4 jets, the request that 4 jets are *b*-tagged reduces the $t\bar{t}$ + jets background of a factor of ~10^3 and the *W* + jets of a factor ~10^5. Requiring the tag of only 3 jets should reduce the background rejection to about 200 for $t\bar{t}$ + jets and ~8000 for W + jets. Therefore including a 3 jets b-tagging selection in the EF classification of the $t\bar{t}H$, $H \rightarrow b\bar{b}$ channel the total amount of data to be analysed can be reduced from 1.3 $\times 10^6$ to ~ 2 $\times 10^4$.

Table 7-5 Expected number of signal and background events for an integrated luminosity of 30 fb^{-1} as a function of the sequential steps taken in the off-line reconstruction algorithm (the assumed *b*-tagging performance is $R_u = 100$ and $R_c = 10$ for $\varepsilon_b = 60\%$) [7-25]

Process	One isol. lepton + 4 jets in $\|\eta\| < 5$	4 b-tagged jets	m_{bb} in 2σ mass windows	Total acceptance
$t\bar{t}H$	7953	440	206	1.8%
$t\bar{t}Z$	800	50	15	1.5%
$t\bar{t}Z$+jets	$2.8 \cdot 10^6$	2740	1040	$3.6 \cdot 10^{-4}$
W+jets	$10 \cdot 10^6$	74	30	$4.8 \cdot 10^{-6}$

7.6.3 b-tagging selection

The *b*-tagging algorithm can also be applied at EF for selection purpose. This can add flexibility to the ATLAS trigger selection strategy, improving the observability of some channel selected with poor efficiency with the actual LVL1 trigger thresholds.

Indeed given the huge QCD multi-jet production, rather high E_T thresholds are set on the individual jets in order to maintain a reasonable LVL1 trigger rate. With the LVL1 menu for three- and four-jets (3j75, 4j55) at low luminosity and (3j130, 4j90) at high luminosity presented in [7-8], some channels with multi-jet final states are selected with a poor efficiency. If a *b*-tagging selection were implemented at LVL2 and EF the jet E_T thresholds at LVL1 could be lowered usefully. It has been already proved in [7-24] that applying only the LVL2 *b*-tagging algorithms (see Section 7.5) the acceptance for the $H \rightarrow hh \rightarrow b\bar{b}b\bar{b}$ channel could be enhanced from 19% to 58% keeping the constraint that the offline sample remains unbiased for ε_b up to 70%. Correspondingly the LVL1 rate is increased from 0.3 to 2.7 Hz. This value could obviously be reduced if a lower increase in efficiency were sufficient. The EF could contribute to this selection chain increasing the global HLT rejection power, without affecting the offline b-tagging efficiency.

The HLT *b*-tagging selection can also be used as an additional trigger tool if the standard trigger menu produces higher rate than expected. In this case the use of *b*-tagging at LVL2 / EF can restore the expected rates. A possible candidate is the $t\bar{t}H$, $H \rightarrow b\bar{b}$ channel. Indeed it is at present triggered at LVL1 by MU20 or EM30I, which are associated with an already very high rate (22 kHz).

The final tuning of these trigger component will be a trade-off between the maximum acceptable LVL1 and LVL2 rate, the interplay between LVL2 and EF and the relevance of the physical channel based on the discovery potential.

7.7 References

7-1 ATLAS Collaboration, *'ATLAS Level-1 Trigger Technical Design Report'*, CERN / LHCC / 98-14.

7-2 S.Tapprogge, 'P*hysics requirements for the ATLAS high-level trigger'*, ATLAS internal note, ATL–DAQ–2000–033 (2000)

7-3 ALEPH Collaboration, *'Observation of an Excess in the search for the SM Higgs Boson at ALEPH'*, CERN-EP / 2000-138 (2000)

7-4 D. Froidevaux and E.Richter-Was, *'Is the channel $H \rightarrow b\bar{b}$ observable at LHC?'*, Atlas Internal Note ATL-PHYS-94-043 (1994)

7-5 D.Cavalli et al., '*Minimal Supersymmetric Standard Model Higgs rates and backgrounds in ATLAS*', ATLAS Internal Note, ATL-PHYS-96-074 (1996)

7-6 G.Polesello et al., 'Precision SUSY Measurements with ATLAS: Introduction and Inclusive Measurements', Atlas Internal Note ATL-PHYS-97-107 (1997)

7-7 M.Cobal, *'Top Physics at LHC'*, Atlas Internal Note ATL-PHYS-96-093 (1996)

7-8 The ATLAS collaboration, *'ATLAS Detector and Physics Performance Technical Design Report'*, CERN / LHCC / 99-14, ISBN 92-9083-141-3

7-9 A.Artamonov et al., 'DICE-95', Atlas Internal Note ATL-SOFT-95-014 (1995)

7-10 M. Goosens et al., *'The ZEBRA System'* , CERN Program Library Long Writeups Q100 / Q101, CERN, Geneva, Switzerland, February, 1995.

7-11 DeWolf et al., '*SLUG Manual - SLUG Interface to GEANT*', ATLAS Internal Note, ATL-SOFT-93-012 (1993)

7-12 DeWolf et al., *'GENZ - General Event Handling using Zebra'*, ATLAS Internal Note, ATL-SOFT-94-013. (1994)

7-13 R. Brun et al., *'GEANT3'*, CERN / DD / EE / 84-1 (1996).

7-14 *'Draft ATLAS ATRECON Manual'*, ATLAS Internal Note ATL-SOFT-96-033 (1996)

7-15 R.Clift and A.Poppleton, *'IPATREC: inner detector pattern recognition and track finding'*, ATLAS Internal Note, SOFT-NO-009, (1994)

7-16 *'PIXLRec pattern recognition'*, http: / / marpix1.in2p3.fr / Pixel / Plan / lv.html

7-17 I. Gavrilenko, *'Description of Global Pattern Recognition Program (xKalman)'*, ATLAS Internal Note INDET-NO-165 (1996)

7-18 The OPAL Collaboration, '*A Measurement of R_b using a Double Tagging Method*', CERN-EP / 98-137 (1998)

7-19 The DELPHI Collaboration, '*Measurement of the Forward-Backward Asymmetry of $e^+e^- \rightarrow Z \rightarrow b\bar{b}$ using prompt leptons and a lifetime tag*', CERN-PPE / 94-161 (1994)

7-20 ALEPH Collaboration, '*Search for the neutral Higgs bosons of the MSSM in the e^+e^- collisions at $\sqrt{s}$ from 130 to 172 GeV*', Phys. Lett. **B412** (1997) 173

7-21 U.Egede, *'A short documentation of the Xconver module for reconstructing conversion'*, LUNDF6 / (NFFL-714X)1997 (1997)

7-22 The ATLAS collaboration, *'Inner Detector Technical Design Report '*, CERN / LHCC / 97-16, ISBN 92-9083-102-2 (1997)

7-23 *'PixTrig: a track finding algorithm for LVL2'*, ATLAS internal note, ATL–DAQ–2000–025 (2000)

7-24 F.Parodi et al., *'b-tagging event selection for the ATLAS high-level trigger'*, ATLAS internal note, ATL–DAQ–2000–023 (2000)

7-25 E.Richter-Was and M.Sapinski, *'Search for the SM and MSSM Higgs boson in the $t\bar{t}H$, $H \rightarrow b\bar{b}$ channel'*, ATLAS internal note, ATL-PHYS-98-132 (1998)

CONCLUSIONS

In this thesis a complete study of the Event Filter subsystem of the ATLAS experiment, from EFF prototype design, implementation and integration to the evaluation of a selection and classification algorithm, which could be executed at EF.

The Event Filter sub-farm High Level Design has been successfully implemented on a Symmetric Multi-Processor machine. To better exploit the hardware resources, a single process multi-threaded implementation has been chosen for the sub-farm code. The robustness of the software architecture has been checked by performing many tests with varying running conditions. The global throughput achieved is independent of the number of processing tasks and is inversely proportional to the processing time. The load balancing among the processing tasks is excellent (a by-product of the OS scheduler itself) and is not affected by the relative composition in the number and type of processing tasks. The scalability test proves that the performances scale according to the number of processors (more work is needed to assess the compatibility of the results with their defined SPEC figures). All the presented results obtained are a good indication that the scalability is assured and that the software and hardware architectures do not limit the behaviour of the sub-farm. The SMP prototype, together with other two distributed EFF implementation, has been also integrated in the global DAQ/EF -1 project, which has been chosen by ATLAS to be the basis of the DAQ system on the ATLAS test beams and the baseline for the evolution of the ATLAS Data Acquisition and Event Filter system. Subsequently the prototype has been ported on a common object oriented framework, which had permitted the software integration of the three different prototypes.

It has been proven that a performant and efficient *b*-tagging algorithm can be performed at EF with negligible contribution to total available latency. The use of this tool at classification level can facilitate the final offline analysis reducing drastically the amount of data, which must be examined. Furthermore it can add flexibility to the ATLAS trigger selection strategy, improving the observability of some channel selected with poor efficiency with the actual LVL1 trigger thresholds, and providing an additional trigger tool if the standard trigger menu produces higher rate than expected.

ACKNOWLEDGEMENTS

First of all I want to express my thanks to my supervisor Valerio Vercesi. His scientific advice and human support have been essential to the fullfillment of this thesis.

I am thankful to all the members of the ATLAS group of Pavia for the friendly atmosphere they have created in these years, for the constant support and for the help to get in touch with the HEP physics.

I would like to thank Christophe Meessen, who has developed the common sub-farm data-flow software, for his help and suggestion in the integration of the SMP prototype in the DAQ/EF -1 project and in the common software framework.

I would like to thank Dario Barberis and Fabrizio Parodi for their advise, supports and suggestions in the *b*-tagging matter