CERN-THESIS-2008-220

Universidad de Cantabria

Facultad de Ciencias

Master en Computación

Instituto de Física de Cantabria

Grupo de computación distribuida y Grids

Tesis de Master

Operación y desarrollo de Herramientas para la gestión de datos en un entorno GRID

Candidato
Ibán Cabrillo Bartolome

Tutor
Dr. **Isidro González Caballero**

Curso Académico 2007-2008

Dedico el presente trabajo a mi esposa Cristina y a mi hija Lara, por estar siempre a mi lado, por su constante apoyo y consejo y por hacer de mi mejor persona.

AGRADECIMIENTOS

A mi tutor Dr.Isidro Gonzalez Caballero por su persistente guía, sin la cual este trabajo no se hubiera acabado.

A todos mis compañeros Rafa, David, Alvaro, Pablo, Clara, Cote, Luis, Jordi, Javi, Nuria y Flori, por su paciencia con mis interminables preguntas.

Y un agradecimiento especial al Dr. Francisco Matorras por sus constantes aportes en la finalización de este trabajo.

Índice general

Índice de figuras

Capítulo 1

Introducción

La Física de Altas Energías y el CERN como uno de sus mayores exponentes, constituyen uno de los principales motores de las tecnologías GRID y están habituados desde hace décadas a realizar e-Ciencia. La e-Ciencia (del inglés e-Science o enhance Science), se define como una actividad científica colaborativa en la que la herramientas están distribuidas geográficamente.

El volumen de datos que serán generados por los experimentos del Large Hadron Collider (LHC) a partir de la primavera del 2009, no tienen precedentes. El experimento CMS, uno de los cuatro detectores que operan en el LHC, generará del orden $150MB$ de datos cada segundo una vez superados los distintos niveles de selección.Todo esto producirá un volumen total aproximado de 10PB incluyendo los diferentes reprocesados y copias. Éstos datos, han de ser archivados y puestos a disposición de la comunidad científica implicada en el experimento.

El objetivo de esta Tesis de Máster es explicar la instalación, operación y desarrollo de un sistema de almacenamiento, gestión y transferencia de datos en un entorno de computación distribuida y más concretamente dentro del experimento CMS del LHC, usando la infraestructura de computación GRID del Instituto de Física de Cantabria (IFCA).

El capítulo 2 contiene una breve descripción del acelerador y el experimento para el cual se ha desarrollado el sistema de almacenamiento. El capitulo 3 describe el entorno de computación sobre el que se asientan las herramientas de *software* para el análisis y procesado de los datos de CMS. La arquitectura de *hardware*, los detalles de la instalación del mismo y la solución de *software* del sistema de almacenamiento se especifican en el capítulo 4. En el capítulo 5, se detallan las herramientas de acceso a los datos almacenados, su configuración y su optimización. El capítulo 6 se centra en las herramientas y ejercicios realizados de transferencia masiva de datos. y finalmente, en el capítulo 7 se recogen las conclusiones de este trabajo.

Capítulo 2

Antecedentes

2.1. El CERN

El Laboratorio Europeo de Física de Partículas (Centre Européenne pour la Reserche Nucléaire, CERN)[1] se encuentra en Suiza, cerca de Ginebra, y próximo a la frontera con Francia (ver figura 2.1). Fundado en 1954 por 12 países europeos, el CERN es hoy en día un modelo de colaboración científica internacional y uno de los centros de investigación más importantes en el mundo. Actualmente cuenta con 20 estados miembros, entre los cuales se encuentra España (los estados miembros comparten la financiación y la toma de decisiones en la organización). Otros 28 países no miembros participan con científicos de 220 instituciones o universidades utilizando sus instalaciones. El Instituto de Física de Cantabria (IFCA) participa en experimentos en el CERN desde 1995. De los países no miembros, 8 estados y organizaciones tienen calidad de observadoras, participando en las reuniones del consejo.

Figura 2.1: Imagen aérea del CERN

2.2. LHC (Large Hadron Colider)

El Gran Colisionador de Hadrones (del inglés Large Hadron Colider o LHC siglas por las que es generalmente conocido) es un acelerador colisionador circular de partículas localizado en el CERN. El LHC se diseñó para hacer colisionar haces de protones de 7 TeV de energía Su propósito principal es examinar la validez y los límites del Modelo Estándar, el que es actualmente el marco teórico de la física de partículas.

El LHC se convertirá en el acelerador de partículas más grande y energético del mundo. Más de 5000 físicos de 34 países y cientos de universidades y laboratorios han participado en su construcción.

Los primeros haces de partículas fueron inyectados el 1 de agosto de 2008 y el primer intento para hacer circular los haces por toda la trayectoria del colisionador se produjo el 10 de septiembre de 2008 (ver figura 2.2), mientras que las primeras colisiones a alta energía tendrán lugar durante el 2009.

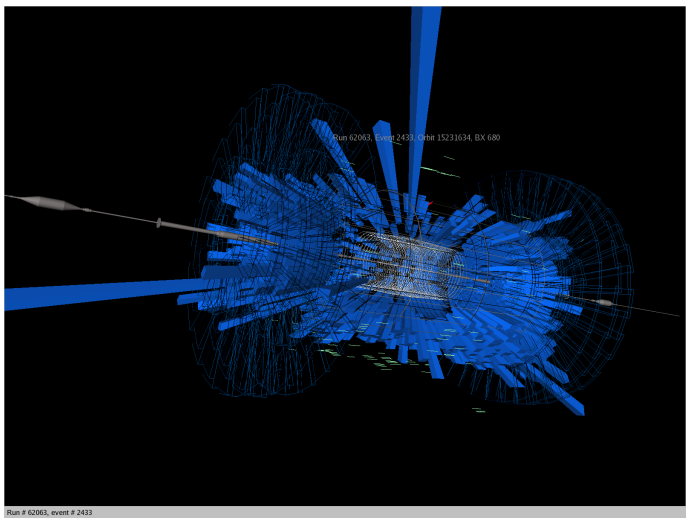
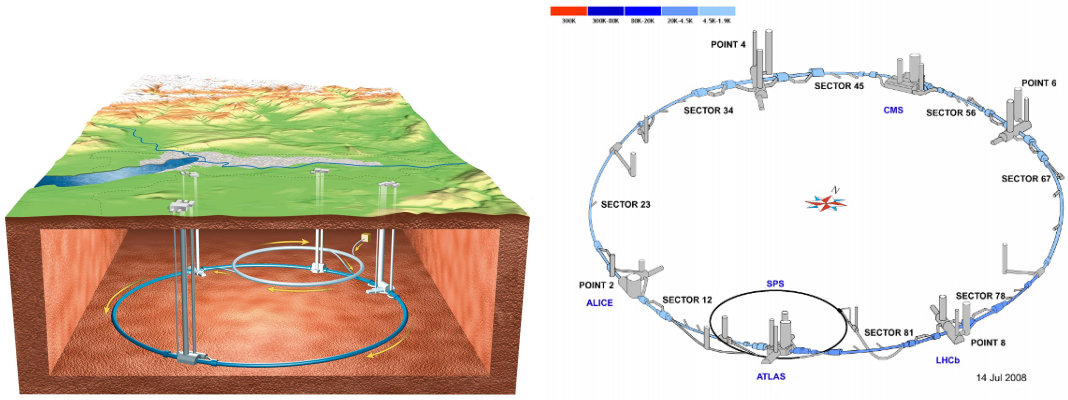


Figura 2.2: Suceso proveniente de interacciones del halo de protones del LHC registrado por el experimento CMS el 10 de Septiembre de 2008.

Uno de los resultados esperados una vez que el acelerador esté en funcionamiento, es la producción y detección de la partícula masiva conocida como el *bosón de Higgs*. La observación de esta partícula confirmaría las predicciones del Modelo Estándar, permitiendo explicar el mecanismo por el cual otras partículas fundamentales adquieren masa. Verificar la existencia del bosón de Higgs sería un paso significativo en la búsqueda de una teoría de la gran unificación, teoría que pretende unificar tres de las cuatro fuerzas fundamentales conocidas: fuerza nuclear débil, fuerza nuclear fuerte y las fuerzas electromagnéticas. Además este bosón podría explicar por qué la gravedad es tan débil comparada con las otras tres fuerzas.

Junto al bosón de Higgs también podrían producirse otras nuevas partículas que son predichas o postuladas por otras teorías fundamentales entre las que destacan los modelos supersimétricos.

El nuevo acelerador usa el túnel de 27 km de circunferencia, creado para el Gran colisionador de Electrones y Positrones (ver figura 2.3). Las partículas serán aceleradas mediante 1232 dipolos superconductores que deben operar a -271°C. Otro conjunto de imanes superconductores (cuadrupolos y hexapolos) se encargan de dirigir y focalizar los haces de partículas.



(a) (b)

Figura 2.3: Esquema de situación en profundidad (a) y superficie (b) de los detectores del LHC

2.3. Compact Muon Solenoid (CMS)

2.3.1. ¿Qué es?

El Solenoide Compacto de Muones (del inglés Compact Muon Solenoid)[2], es uno de los dos detectores de partículas de propósito general que han sido construidos en el LHC. En su construcción han colaborado unas 2.600 personas procedentes de 180 institutos científicos diferentes, entre los que se encuentra el IFCA.

CMS (ver figurar 2.4) se encuentra instalado en la caverna situada en el punto de colisión número 5 a unos 100 metros por debajo del pueblo francés de Cessy. Tiene una forma cilíndrica con 21 metros de largo y 16 de diámetro. Su peso es de unas 12.500 toneladas. La parte central del cilindro se denomina *barril* mientras que las tapas reciben el nombre de *zona forward*.

CMS es capaz de estudiar múltiples aspectos de las colisiones de protones a 14 TeV, la energía del LHC. Contiene sistemas para medir la energía y la cantidad

de movimiento de fotones, electrones, muones y otras partículas producto de las colisiones. La capa detectora más interna es un detector de trazas de silicio semiconductor. A su alrededor, un calorímetro electromagnético de cristales centelleadores, rodeado de un calorímetro hadrónico. El detector de trazas y el calorímetro son lo suficientemente compactos como para entrar dentro del imán solenoidal de CMS. Este imán puede generar un campo magnético de hasta 4 T. En el exterior del imán se sitúan grandes detectores de muones.

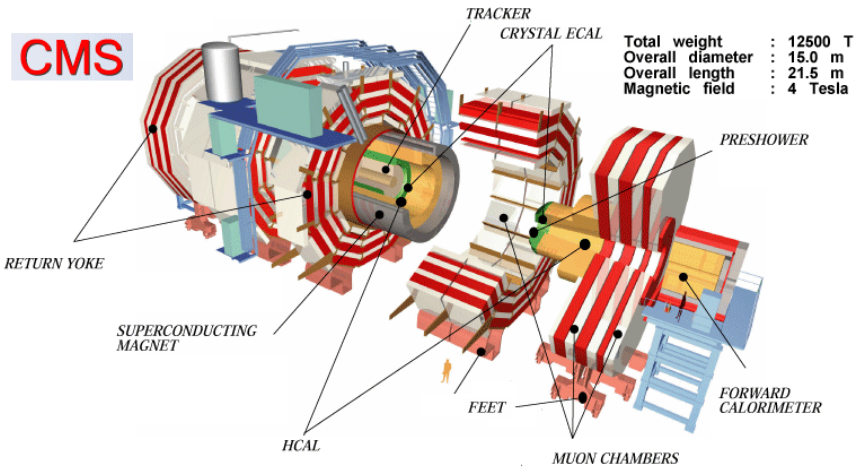


Figura 2.4: Esquema genera de las diferentes partes que constituyen el detector CMS

2.3.2. Partes

Vamos a describir de una manera superficial, las diferentes partes, o mejor dicho capas, que forman el detector. La descripción se va a realizar desde el interior hacia el exterior como podemos observar en el siguiente corte transversal (ver figura 2.5)

I. **Región Central:** En esta zona colisionan los haces de protones. Los imanes de enfoque del LHC, fuerzan a los protones, que giran en sentido opuesto, a colisionar en el centro del detector. Los haces de protones se distribuyen en paquetes, con unos 100.000 millones de protones formando cada paquete. Los protones son tan pequeños que la probabilidad de que choquen es muy reducida, con una tasa de unas 20 colisiones por cada 200.000 millones de protones. Cuando dos protones colisionan a esas energías, el intercambio de materia y energía implica la aparición de nuevas partículas. Muchos de esos procesos de producción de partículas son bien conocidos y se estima que sólo 100 de cada 10^9 colisiones producirán eventos interesantes. Por tanto, interesa producir la mayor cantidad de colisiones posibles, con lo que los paquetes,

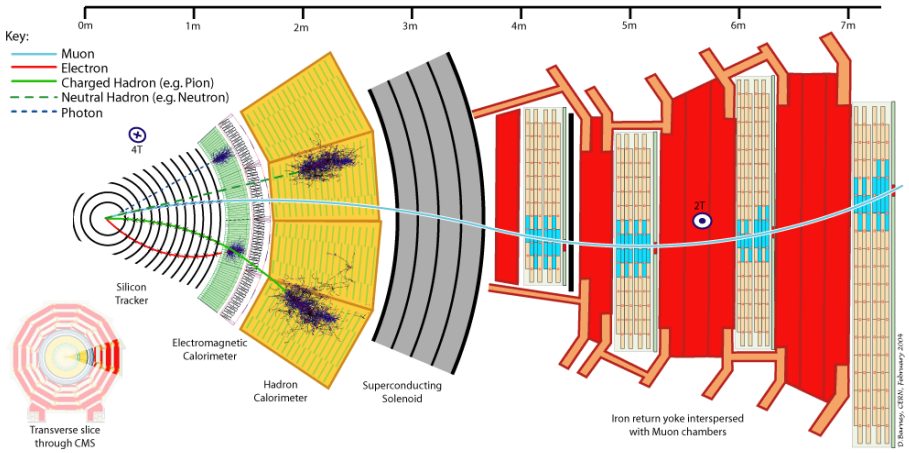


Figura 2.5: Corte transversal del Detector CMS

que viajan muy juntos en el haz, se cruzarán unas 40 millones de veces por segundo, un cruce cada 25 nanosegundos.

II. **Tracker:** Está compuesto por finos segmentos de sílicio (*strips* y píxeles) que permiten medir el momento y la trayectoria de las partículas cargadas. También revelan la posición donde se desintegran partículas inestables de vida media larga. CMS contiene el mayor detector de silicio del mundo, con 205 m^2 de sensores (el área aproximada de una cancha de tenis). Está formado por 9,3 millones de *strips* y 66 millones de píxeles.

III. **Calorímetro Electromagnético:** Está constituido por unos 80.000 cristales centelleadores de tungstato de plomo ($PbWO_4$), lo cual le proporciona una gran granularidad. Permite medir con precisión las energías de fotones y electrones.

IV. **Calorímetro Hadrónico:** Formado por capas de material denso (bronce o acero) y capas de centelleadores plásticos y fibras de cuarzo. Determinan la energía de los hadrones que la atraviesan, esto es, partículas como los protones, neutrones, piones y kaones.

V. **Imán:** Al igual que muchos detectores de partículas, CMS tiene un gran imán solenoidal. Este imán permite determinar la relación masa/carga de las partículas que lo atraviesan a partir del análisis de la curva que recorren en el seno del campo magnético. Mide 13 m de largo y 6 de diámetro, y su núcleo superconductor de niobio-titanio está refrigerado criogénicamente con helio líquido. Opera con un campo magnético de 3,8 T. La inductancia del imán es de 14 henrios y la intensidad de corriente que lo atravesará será de 19,500 A, con lo que almacenará un total de 2,66 GJ, el equivalente a media tonelada de TNT.

Hay circuitos preparados para disipar de forma segura un exceso de energía que podría fundir el imán.

VI. **Detector de Muones:** Para detectar muones y su cantidad de movimiento, el CMS usa tres tipos de detectores: tubos de deriva (*drift tubes*), cámaras de tiras catódicas (*cathode strip chamber*) y cámaras de tiras resistivas (*resistive plate chambers*). Los TD se usan para mediciones precisas de la trayectoria en la región central (el barril), mientras las CTC se usan en las partes más externas. Las CTR devuelven una señal rápida cuando un muón atraviesa el detector muónico, y están instaladas en el barril y en la parte externa.

2.3.3. ¿Cómo Funciona?

Las nuevas partículas que se producirán en CMS serán en general inestables y se desintegrarán rápidamente en una cascada de partículas más ligeras y conocidas. Las partículas que atraviesen CMS dejarán señales que permitirán reconocerlas, así que a través de su existencia se podrá inferir la presencia de partículas nuevas. Podríamos explicarlo como si en cada colisión el detector tomara una fotografía del impacto: muchas de las colisiones no serán interesantes ya que no aportarán nada nuevo, serán colisiones conocidas y se podrán destruir; sin embargo otras muchas habrá que guardarlas para su posterior análisis.

Cada suceso registrado por CMS ocupa aproximadamente un volumen de 1,5 MB de datos. Un potente sistema de filtrado y selección (*trigger*) se encarga de seleccionar por medio de una electrónica veloz especializada y *software*, solo los sucesos interesantes para su almacenado y procesado final. El objetivo es escribir 100 de las 40 millones de colisiones que se producirán cada segundo. Se estima que cuando el LHC este funcionando a pleno rendimiento operará durante unos 116 días (10^7 segundos). Esto nos da una producción de datos anuales de:

$$\text{Datos} = 1 \text{ año} \times 10^7 \text{ s/año} \times 100 \text{ col./s} \times 1.5 \text{ MB/col.} = 1{,}5 \text{ PB/año}$$

Esta es la cantidad de datos que serán generados por el detector. Después de varios reprocesados en los centros, copias, etc, según el modelo de computación de CMS[3], obtendremos un volumen total de datos aproximado de 10 PB anuales.

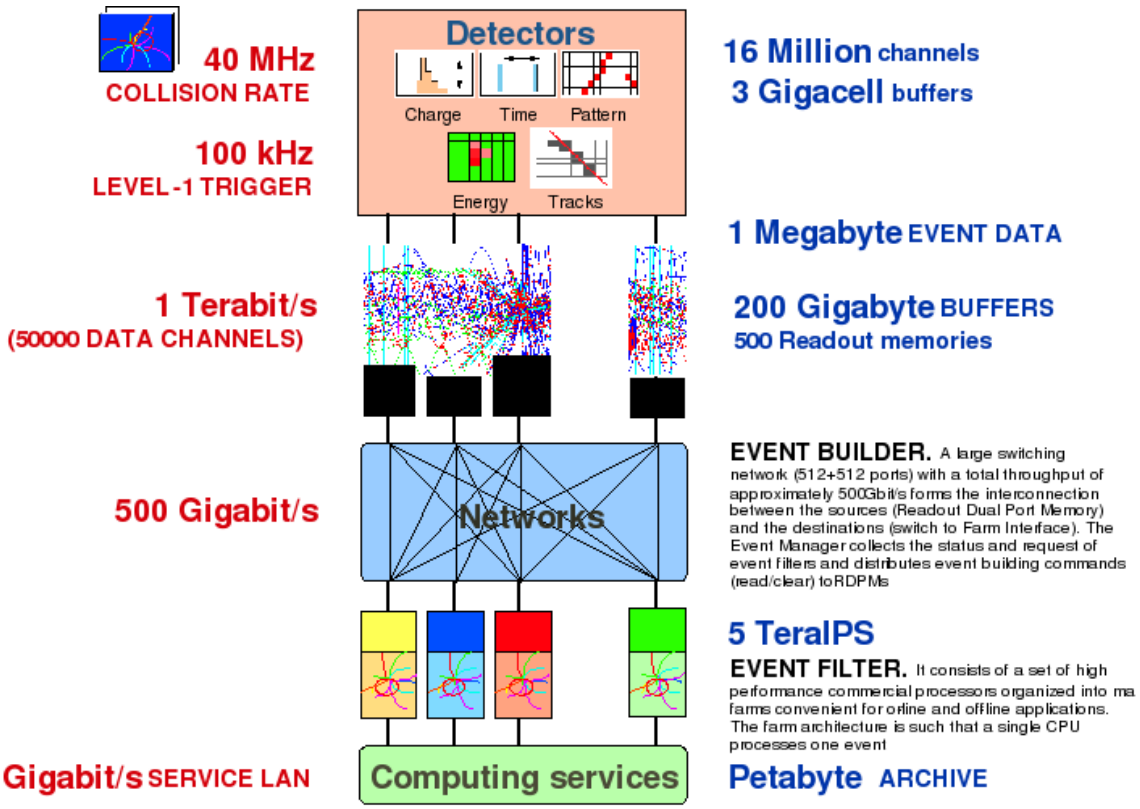


Figura 2.6: Proceso de adquisición de Datos en el Detector CMS

Capítulo 3

Entorno Computacional

3.1. Computación GRID

La palabra GRID, que significa malla en castellano, se usa en el entorno de las redes eléctricas pues esta diseñada como un mallado de lineas de corriente interconectadas unas con otras, de forma que, para el usuario final (nosotros en nuestra casa, por ejemplo), es totalmente transparente el uso de la misma.

Al igual que cuando pulsamos un interruptor, la energía que consumimos en ese momento puede ser suministrada desde varios lugares diferentes, siendo lo único importante que haya corriente cuando pulsamos el interruptor, y no de donde proviene la energía.

Con esta idea básica nace el concepto de computación GRID. Lo que pretende la computación GRID es que con una simple conexión a la red (un terminal por ejemplo) se disponga de acceso a diferentes sistemas de computación, almacenamiento, etc, de forma transparente para el usuario, pudiendo estar éstos distribuidos por diferentes partes del mundo.

3.1.1. Historia

Los primeros pasos hacia la computación GRID de hoy se dieron en un entorno educativo de Estados Unidos, con el fin de ampliar las posibilidades de comunicación del conocimiento. Ian Foster [4], investigador de IBM, profesor de ciencias de la computación en la Universidad de Chicago y al frente del Laboratorio de Distribución de Sistemas de Argonne, dio forma a la idea en compañía de Steven Tuecke y Carl Kesselman, creando así el Globus Project [5] en 1996. En este sentido, el proyecto obtuvo fondos de entidades del gobierno norteamericano, lo cual hizo posible

una pronta adopción del GRID en ciertos sectores.

Más tarde se sumarían organizaciones del gobierno europeo a la apuesta, además de referentes de la industria informática, como Microsoft e IBM. Este último se convirtió en el principal aliado del Globus Project. De esta manera, con IBM se presentó un paquete de *software* libre destinado a la construcción y a la expansión de nuevos GRID, así mismo IBM fue uno de los fundadores del Global GRID Forum [6], un espacio de encuentro para el desarrollo de estándares y herramientas para la emergente tecnología.

3.1.2. ¿Qué es?

La computación GRID es una forma de compartir potencia de cálculo y recursos de almacenamiento sobre Internet. Está posibilitando grandes contribuciones a la ciencia ayudando a los científicos de todo el mundo a analizar y almacenar grandes cantidades de datos. Debe cumplir una serie de características:

I. **Recursos Compartidos:** Los recursos han de ser compartidos a escala global, diferentes centros, paises etc, usando una infraestructura común existente. Esta es la verdadera esencia de la computación GRID; si los recursos son compartidos localmente, hablamos de un cluster pero no de un GRID.

II. **Acceso Seguro:** Tiene que haber gran nivel de confianza entre las partes que comparten sus recursos, dado que se permite el acceso de usuarios externos a diversas instalaciones, lo cual podría entrar en conflicto con la política de seguridad de muchos centros. Un usuario malintencionado podría intentar el uso de estas infraestruccturas para su propio beneficio, o simplemente para obtener informacion sensible de otros usuarios del sistema. Se requiere, por tanto, la utilización de sistemas de acceso seguro y autenticado.

III. **Utilización de Recursos:** El uso de los recursos ha de ser balanceado y distribuido eficientemente entre todos los centros que los comparten.

IV. **Latencia:** La distancia entre los diferentes sitios no puede ser una desventaja a la hora de compartir recursos, ya que la velocidad de acceso a los mismos debe ser del mismo orden de magnitud para cada uno de ellos.

V. **Estándares Abiertos:** Son necesarios para asegurar que los diferentes nodos pueden interactuar y que cualquiera puede contribuir al desarrollo del GRID.

Existen varias infraestructuras GRID, repartidas por el mundo como son Enabling Grids for E-sciencE (EGEE), Inteugrid[1] u Open Science Grid (OSG)[2]. Dado que los diferentes experimentos del LHC y la mayoría de los centros europeos de *Altas Energías* (y en particular el IFCA) operan dentro de la infraestrucctura EGEE, la explicaremos más en detalle a continuación.

3.1.3. EGEE

Enabling Grids for E-sciencE (EGEE) [7], financiado por la Comisión Europea, se encarga del desarrollo e incorporación de herramientas a la infraestructura GRID dando servicio a la comunidad cientifica independientemente de su situación greografica, 24 horas al día. Es actualmente el proyecto GRID más grande de Europa en cuanto a personas involucradas, procedentes de más de 30 países y unos 150 centros de investigación. Se encuentra actualmente en la III fase y es el sucesor de proyectos anteriores como CrossGrid y DataGrid. El IFCA (participante en EGEE I, II y III) y otros centros de la comunidad científica española, como son el Centro de Investigaciones Energéticas Medioambientales y Tecnológicas (CIEMAT)[3], Centro de Supercomputación de Galicia (CESGA)[4] o Instituto de Física Corpuscular (IFIC)[5], son miembros de este proyecto, aportando conocimiento y desarrollo en el mismo y participando activamente en él. El proyecto se centra principalmente en tres areas:

- Crear una Infraestuctura GRID sólida, robusta y segura, con la intención de seguir atrayendo mas recursos de computación al proyecto.
- Continuo desarrollo y mantenimiento del *middelware*.
- Atraer nuevos usuarios de la industria y la investigación asegurándose de que reciban el entrenamiento y soporte necesario.

En la figura 3.1 podemos ver de forma más o menos esquemática los servicios básicos que se deben ejecutar en GRID dentro de la infraestructura EGEE y como interactuan entre ellos.

Estos servicios pueden ser instalados cada uno en un único nodo o por el contrario varios servicios en el mismo. A continuación haremos una breve descripción de alguno de ellos:

[1] http://www.i2g.eu/
[2] OSG web, http://www.opensciencegrid.org/
[3] http://www.ciemat.es/
[4] http://www.cesga.es/
[5] http://ific.uv.es/

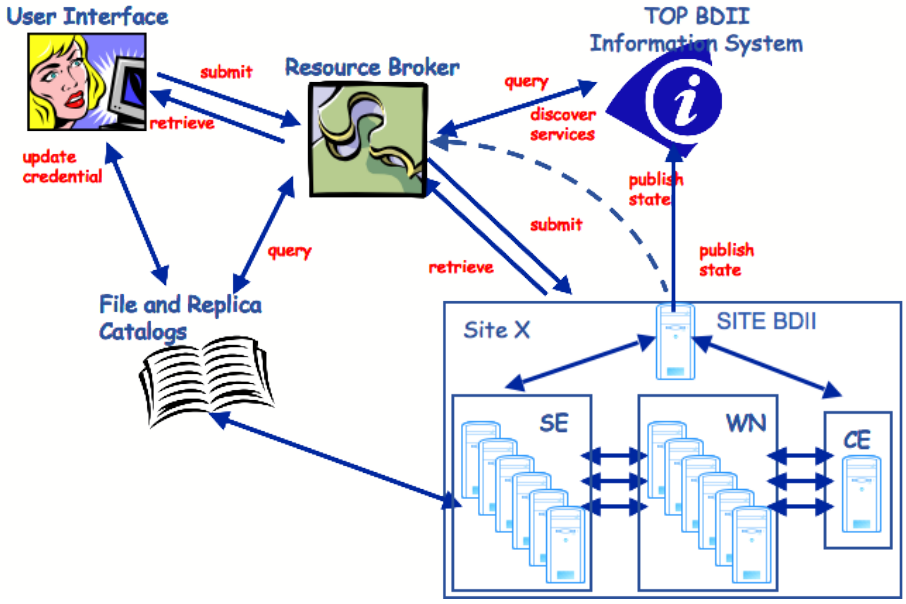


Figura 3.1: Servicios en un entorno GRID y relaciones entre ellos

I. **User Interface (UI):** es el servicio que permite a los usuarios interactuar con el GRID. Dispone de todo el *middleware* y las herramientas necesarias para que un usuario pueda comunicarse con el resto de los servicios.

II. **Computing Element (CE):** Es el interfaz entre un nodo y el resto de la infraestructura GRID, encargandose de gestionar los trabajos localmente.

III. **BDII:** Es el sistema de información, tanto de un cluster local (Site BDII), como de toda una infraestructura GRID (Top BDII). Es el encargado de publicar la información de cada uno de los nodos (capacidad de almacenamineto, número de CPUs, software instalado), de manera que sea accesible para todos los usuarios.

IV. **Resource Broker (RB):** Es el encargado de analizar las necesidades del trabajo que el usuario quiere ejecutar en el GRID y de buscar, utilizando el BDII un *Computing Element (CE)* que reúna las características solicitadas para el proceso que se quiere realizar (potencia mínima de la CPU, software adicional, etc).

V. **Worker Node (WN):** Cada una de las máquinas donde se realiza la computación.

VI. **Storage Element (SE):** Es el encargado de dar soporte de almacenamiento masivo, tanto desde el *User Interface (UI)* para recuperar los datos de salida del trabajo, como desde los *Worker Nodes (WN)*, para acceder a los datos de entrada.

3.1.4. LHC Computing GRID (LCG)

EL LHC Computing GRID Project (LCG)[8], es una colaboración del CERN, entre los experimentos del LHC[6] y los centros de computación de los diferentes intitutos de física implicados, trabajando todos juntos para desarrollar y preparar el entorno de computacional necesario para dar soporte de almacenamiento y análisis a los datos del LHC[9].

Todos las partes implicadas, aportan recursos al proyecto LCG siendo éste el encargado de crear un entorno coherente de comunicación entre todos ellos, dada la heterogenidad dentro de una colaboración internacional. Los principales objetivos del proyecto son:

I. Desarrollo de nuevas aplicaciones de software de física, librerias científicas, herramientas de almacenamiento, gestion de acceso a datos, etc.

II. Desarrollo y despliegue de servicios de computación basados en el modelo de computación GRID.

III. Gestión de usuarios, deberes y derechos, dentro de una colaboración internacional en un entorno GRID, no centralizado.

IV. Colaboración con la redes nacionales y locales de investigación como la National Research and Education Network (NREN), para asegurar un alto ancho de banda entre los centros participantes.

V. Coordinar y poner a prueba los diferentes servicios del LHC.

3.2. Modelo Computacional de CMS

La mayor parte de los colaboradores del experimento de CMS no se encuentran en el CERN si no que están distribuidos geográficamente por todo el mundo (principalmente Europa y EEUU).

Debido al elevado volumen de datos que CMS está manejando (aprox. 10 PB), el coste y la complejidad en un único centro para poder procesar estos datos lo convierten en inviable. Por ello tenemos que recurrir a la computación distribuida, compartiendo recursos geográficamente dispersos entre los centros participantes en el experimento. Este es el motivo por el que encaja tan bien el modelo GRID, aunque hay que adaptarlo, optimizarlo y extenderlo a las características de un experimento de Altas Energías.

[6] ALICE,ATLAS, CMS y LHC-b

Un aspecto importante del modelo computacional de CMS es que está fuertemente condicionado por la localización de los datos. Los trabajos se ejecutan cerca de los datos, de modo que se evita un excesivo flujo de transferencias. Esto hace que la correcta gestión de las transferencias y almacenamiento, sea crucial para garantizar el éxito científico de CMS.

Además no tendría ningún sentido, si los usuarios finales (científicos en nuestro caso), no pudieran acceder de una manera rápida y fiable a los datos que quieren analizar, procesar, inyectar, etc. Por ello dentro del experimento CMS y el resto de los experimentos del LHC, se han diseñado estructuras jerárquicas para la distribución de estos datos, las cuales han ido evolucionando con el paso del tiempo con idea de estar totalmente preparados para la puesta en marcha del LHC.

El modelo computacional de Compact Muon Solenoid (CMS) quedaría definido si decimos que es un modelo que además de distribuido, como hemos mencionado anteriormente, es jerárquico y orientado a los datos.

3.2.1. Modelo Jerárquico

Los experimentos del LHC siguen una organización jerárquica basada en el modelo desarrollado por MONARC[10], estructurado en niveles o *Tiers*.

El nivel en el que un centro se encuentra define tanto sus responsabilidades para con la colaboración, como la calidad del servicio que tiene que proporcionar. Las responsabilidades asignadas en CMS a cada *Tier* son:

- ***Tier-0* (T0), CERN:**
  - Primer procesado de datos directamente del detector en el CERN.
  - Almacenar datos RAW (ver sección 3.2.2).
  - Ejecución solo de actividades programadas. No se permite el acceso a usuarios.
  - Distribuir segundas copias a los *Tier-1*.
  - Determinar calibraciones y alineamientos para la reconstrucción rápida.
  - Soporte 24x7.
- ***Tier-1* (T1):**
  - Responsabilidad en la custodia permanente de parte de los datos RAW, datos sin procesar y RECO, datos con un primer procesado (ver sección 3.2.2).

- Re-procesado, programado de datos, filtrado de datos.
- Calcular calibraciones y alineamientos más precisos.
- Reprocesado y análisis a gran escala de datos.
- Servir datos a los *Tier-2*.
- Soporte 24x7.

- ***Tier-2* (T2):**
  - Análisis locales de comunidades de científicos.
  - Análisis basado en GRID para todo el experimento.
  - Producción de Monte-Carlo.
  - Soporte 12x5.

- ***Tier-3* (T3):**
  - Instalaciones últimas de computación para universidades y laboratorios.
  - Acceso interactivo a datos y procesados en *Tier-2s* y *Tier-1s*.

3.2.2. Modelo Orientado a los datos

La idea General de CMS es reducir el número de transferencias de modo que los datos se almacenen donde vayan a ser usados. Todo ello teniendo en cuenta las necesidades de los científicos en el caso de los *Tiers-2* (y *Tiers-3*), o para su almacenamiento y custodia en el caso de los *Tiers-1*.

El conjunto de los datos se agrupa en *DataSets*. En función del tipo de colisiones que contengan, por razones prácticas, se divide en *FileBlocks*, que a su vez se agrupan en *ficheros* de entre 1 y 3 GB. Cada fichero contiene entre cien y mil *sucesos* (ver figura 3.2). Los datos pueden ser movidos entre los diferentes sitios a nivel de *DataSets* preferentemente o *FileBlocks*, ocasionalmente[11].

Además, en función de su contenido y nivel de procesado, se distingen distintos tipos de datos:

I. **RAW Data:** Datos en bruto provenientes del detector (aprox. 1,5$MB/suceso$).

II. **FEVT:** Contienen los datos RAW junto con todos los objetos creados en el proceso de reconstrucción $FEVT = RAW + RECO$ (aprox. $2MB/suceso$).

III. **RECO:** Contiene un subconjunto de **FEVT** suficiente para aplicar calibraciones para el reprocesado (aprox. $500KB/suceso$).

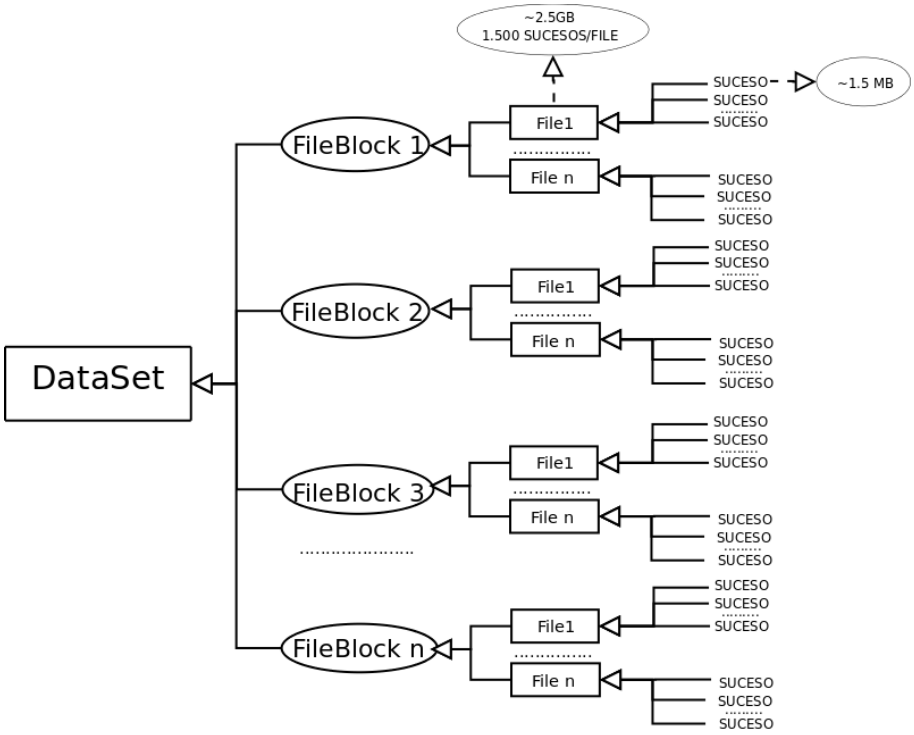


Figura 3.2: Empaquetado de datos en CMS

IV. **AOD:** Subconjunto de RECO suficiente para la mayoria de los análisis físicos estándar (aprox. $100KB/suceso$).

También podemos crear muestras de procesos del detector a traves de simulaciones de Montecarlo, para ello necesitamos varios pasos:

I. **GEN(generacion):** Sucesos de entrada para la simulación, producidos por un código de generador de sucesos externo.

II. **SIM(similacion):** Simulación del paso de las partículas generadas a través del detector mediante el conjunto de herramientas Geant4[12].

III. **DIGI(digitalizacion):** Simulación de la respuesta electrónica a los impactos en el detector.

En la figura 3.3 podemos observar la evolución de los datos de CMS a través de los diferentes *Tiers*.

El formato final de los datos *DIGI* es equivalente al de los datos *RAW*, de manera que a partir de este punto es independiente de si se trata de datos simulados o reales.

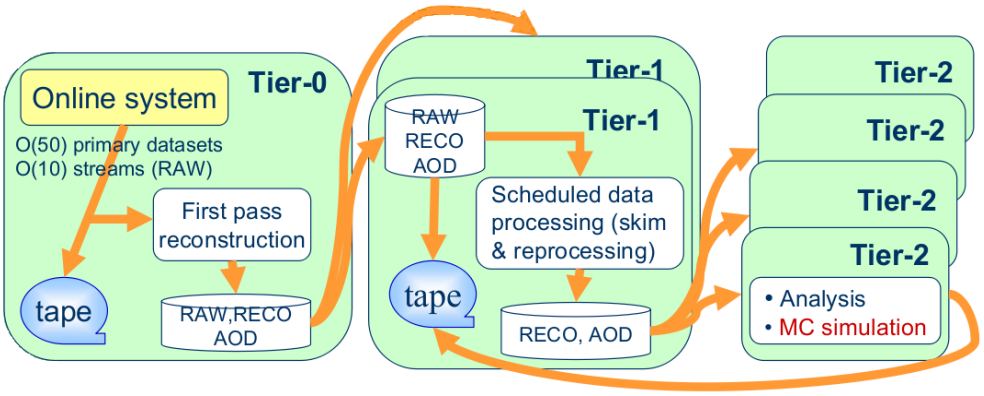


Figura 3.3: Proceso de datos en CMS desde el Tier-0 hasta los Tier-2

3.2.3. Herramientas de Gestión de Datos

Durante estos últimos años se han realizando pruebas de carga, así como sucesivas modificaciones, tanto del *software* como del *hardware* implicado en todo el proceso de almacenamiento y transferencia de datos, todas ellas para comprobar la solidez y eficacia dentro del sistema de transferencias de CMS, afinándolo para que esté a punto para la llegada de los datos.

En el entorno de CMS y LCG, se han desarrollado diversas herramientas especializadas para realizar las actividades relacionadas con la operación normal del detector, algunas de ellas exclusivas de CMS, y otras desarrolladas por EGEE y LCG.

Entre las herramientas más directamente relacionadas con los objetivos de esta tesis estan:

I. **GridFtp:** El GridFtp es un protocolo de transferencia de ficheros, seguro y de alto rendimiento. Está basado en el protocolo File Transfer Protocol (FTP), y optimizado para infraestructuras GRID. Permite la modificación diferentes parámetros como el numero de hilos, tamaño del *buffer*, tamaño de bloque, encriptación, etc. Constituye el pilar básico en la transferencia de datos.

II. **SRM (Storage Resource Managment):** La heterogeneidad de los diferentes sistemas de almacenamiento es uno de los mayores problemas que podemos encontrar en el manejo, réplica y acceso a los datos. Define una interfaz subyacente de acceso a los recursos de almacenamiento independientemente la tecnología. Está basado en especificaciones comunes que han ido surgiendo con el tiempo y evolucionando mediante una colaboración internacional de todos los miembros del LHC. Esta aproximación a los estandares abiertos ha permitido que diferentes instituciones con sistemas de almacenamiento heterogéneos

posean una capa homogenea de acceso al GRID.

III. **FTS (File Transfer service):** FTS es la última evolución en la gestión de las transferencias de ficheros dentro del proyecto LCG. El FTS (actualmente en la version 2.0), define un canal de administración de transferencias entre *Tier-0* $\longleftrightarrow$ *Tier-1*, *Tier-1* $\longleftrightarrow$ *Tier-1* y *Tier-1* $\longleftrightarrow$ *Tier-2*. Proporciona la posibilidad de regular el número de transferencias, modificando la relación *trabajos* $\times$ *hilos* para cada canal FTS creado. Posibilita diferentes configuraciones para diferentes centros. En el modelo de distribución de datos de CMS actual, cada *Tier-0*, *Tier-1* tiene la capacidad de administrar los canales FTS, creando y estableciendo los parámetros que se estimen necesarios con cada uno de los centros para los cuales el canal exista.

3.2.4. Topología de Red en CMS

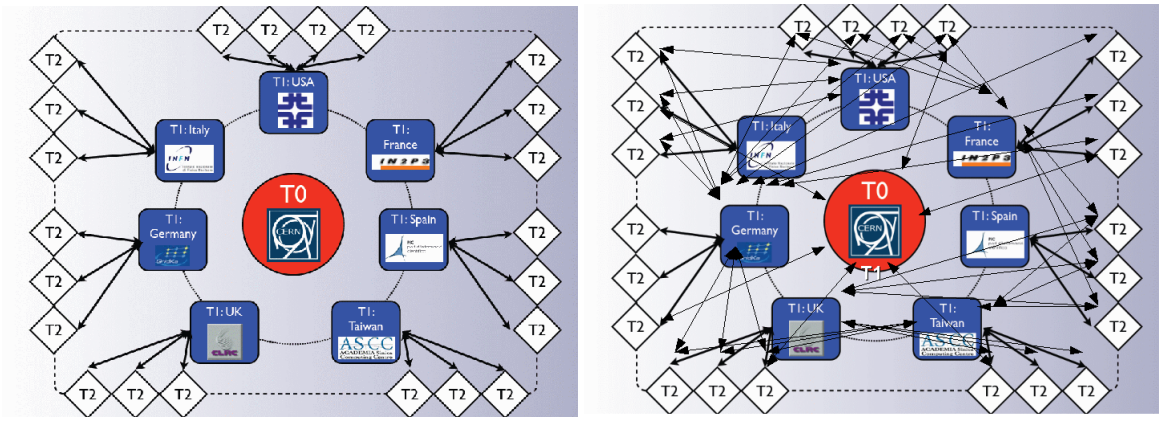
La topología de CMS ha sufrido diversos cambios durante los dos últimos años debido a la mejora de servicios e infraestructuras. Esto ha provocado que tuvieramos que rediseñar la forma en la que los datos eran movidos y accedidos En el modelo inicial (ver figura 3.4 a) cada *Tier-2* tenía un *Tier-1* al que estaba asociado. El movimiento de datos en cada *Tier-2* solo se realizaba con su *Tier-1* asociado. Esta topología tan estructurada, obligaba en ocasiones, a realizar varios saltos entre *Tiers* para acceder a los datos.

En el modelo actual (ver figura 3.4 b) los *Tier-2* pueden descargar y subir datos desde cualquier otro *Tier-1*, y no solo desde el *Tier-1* asociado. Esta estructura, más compleja de optimizar, presenta la ventaja de agilizar las transferencias de los datos evitando saltos innecesarios entre *Tier-1*. Aún así no desaparece la figura de Tier-1 regional que sigue siendo la primera línea de soporte y ayuda para sus *Tier-2*.

Con el fin de asegurar que las transferencias se realizan de manera fluida y que no se ven obstaculizadas por malas configuraciones o ineficiencias de cualquier tipo, cada enlace entre un *Tier-1* y un *Tier-2* ha de superar una serie de requisitos mínimos y pruebas sobre la velocidad de transferencia y la calidad de las mismas, cuestión que se abordará en la sección 6.3.

3.3. Tier-2 Español

El *Tier-2* Español, es un *Tier-2* federado, es decir, está formado por varios centros distribuidos geográficamente, en este caso por el IFCA, situado en Santander, y el Centro de Invest. Energéticas Medioamb. y Tecnológicas (CIEMAT), situado en



(a) (b)

Figura 3.4: Cambios de la Topología de Red en CMS : Topología de Red en CMS hasta 2007 (a) y Topología de red actual de CMS (b)

Madrid, y tiene como centro *Tier-1* asociado al Port d'Informació Científica (PIC), que se encuentra en Barcelona.

Como contrato subscrito con el proyecto CMS, el *Tier-2* español adquirió el compromiso de proporcionar el 5 % de los recursos de CMS tanto de computación como de almacenamiento. En concreto El *Tier-2* español tiene como objetivo:

I. Dar servicio a las necesidades de análisis de entre 20 y 100 investigadores, proporcionándoles los recursos informáticos necesarios para procesar reiteradamente las muestras seleccionadas, las cuales deben de residir localmente.

II. Proporcionar los recursos informáticos necesarios para la producción de datos simulados Montecarlo.

III. La correcta instalación y monitorización de estos servicios, así como su coordinación con el resto de los recursos no dedicados en exclusiva a CMS: es fundamental para que el IFCA pueda participar de manera activa en la producción de resultados y en el análisis de los datos.

IV. Participar activamente en los diversos eventos que programará CMS, CSAs, DDTs, etc (ver sección 6.2) con ánimo de mejora y depuración del servicio.

Para poder llevar a cabo las tareas anteriormente descritas, es necesario que cada *Tier-2* ejecute una serie de servicios, para que la transferencia entre los diferentes *Tiers* se realize de manera transparente. El T2_ES_IFCA (es el nombre con el que se conoce al IFCA dentro de la infraestructura de computación de CMS) debe ejecutar al menos los siguientes servicios y herramientas:

I. **PhEDEx:** Es el servicio que gestiona las tranferencias de datos dentro de CMS. Actualmente se ejecuta en una maquina IBM Xseries 3550 con 3GB RAM. Anteriores versiones de los agentes de PhEdEx consumian una gran cantidad de memoria RAM. Se describe más detalladamente en la sección 6.1.

II. **Squid/Frontier:** Servicio cache local, que almacena una copia de la base de datos de las constantes de alineamiento que son usadas por los trabajos de análisis. De esta forma se limita el uso de la base de datos central en el CERN, disminuyendo la carga de la misma. Actualmente se ejecuta en una maquina IBM Xseries 3550 con 2GB RAM.

III. **CRAB:** Herramienta que facilita la interacción con el GRID a los usuarios en el *User Interface*.

Aparte de los servicios anteriormente mencionados, el *Tier-2* debe de monitorizar y velar por el correcto funcionamiento de todos ellos. Para ello, durante los últimos años, la comunidad ha desarrollado diversas herramientas para poder monitorizar todos estos servicios facilitando el trabajo de los operadores de los diferentes *tiers*, algunas de ellas son *SAM*[7], *PhEDEx web*[8], *Frontier Monitoring*[9], *CMS DashBoard*[10], *widgets*[11], en el IFCA se monitorizan todos estos servicios utilizando GANGLIA[13].

[7]SAM monitoring, http://lxarda16.cern.ch/dashboard/request.py/latestresultssmry?sites=IFCA-LCG2, (Sep. 2008)

[8]PhEDEX monitoring, http://cmsweb.cern.ch/phedex/prod/Info::Main, (Sep. 2008)

[9]Frontier monitoring, http://frontier.cern.ch/squidstats/indexcms.html, (Sep. 2008)

[10]CMS General monitoring, http://arda-dashboard.cern.ch/cms/, (Sep. 2008)

[11]Widget monitoring tools, http://www.netvibes.com/gonis#Main, (Sep. 2008)

Capítulo 4

Sistema de Almacenamiento en el IFCA

Hasta febrero del 2008 en el IFCA los diferentes proyectos GRID, empleaban soluciones de almacenamiento sencillas, como la instalación de pequeños servidores RAID (ver sección 5.1.1) de diferentes marcas. Se observó que a medida que los proyectos crecían, las necesidades de almacenamiento, la fiabilidad, escalabilidad y accesibilidad, no podían ser satisfechas por este tipo de soluciones.

Debido a todos estos motivos y teniendo en cuenta la necesidad de los diferentes proyectos en los que participa el IFCA (posiblemente de todos ellos el más exigente en cuanto a necesidades de I/O y volumen de almacenamiento era el *Tier-2* de CMS), las necesidades de los diferentes grupos de investigación, y tras un largo poceso de selección entre diversos fabricantes de soluciones de almacenamiento (Hewlett Packard (HP)[1], EMC2[2], Bull[3], IBM[4]), se optó por la compra de la solución propuesta por IBM que consistía en:

- Storage Area Network (SAN) compuesta por controladoras DS4700 y cabinas de discos EXP810 con soporte Internet SCSI (iSCSI)/Serial Advanced Technology Attachment (SATA) mayor de 1TB.
- Sistema de ficheros General Parallel File System (GPFS)[14].

Una vez que el *hardware* estuvo en las instalaciones del IFCA, se procedió a su instalación como se detalla en las sucesivas secciones.

[1]HP, http://www.hp.es/
[2]EMC2, http://spain.emc.com/
[3]Bull, http://www.bull.com/index.php
[4]IBM, http://www.ibm.com/es/

4.1. Hardware

En la figura 4.1, puede verse de forma esquemática la solución de almacenamiento implementada en el IFCA, basada en la propuesta de IBM anteriormente mencionada.

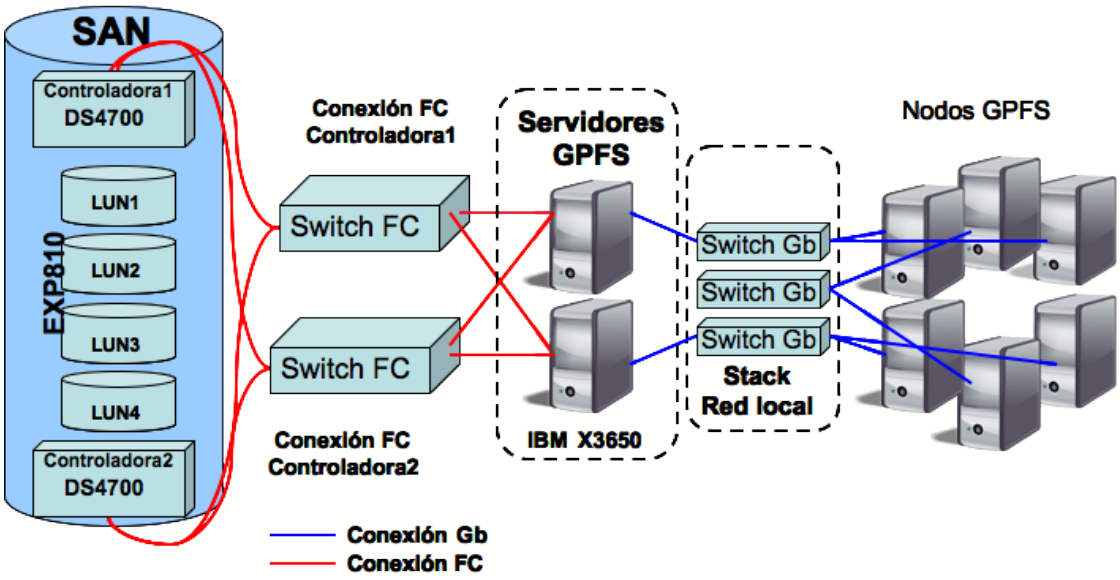


Figura 4.1: Esquema general de interconexión del hardware de almacenamiento implementado en el IFCA.

4.1.1. Instalación de controladoras DS4700 y cabinas de Discos EXP(810)

El primer paso fue la instalación de las controladoras de la cabina Storage Area Network (SAN). Además ofrecen conexiones Fibre Channel (FC) a 4 Gbps de alto rendimiento redundantes. Admite hasta 112 módulos de unidades de disco mediante la conexión de seis unidades apiladas de expansión EXP810 (ver figura 4.2) y tiene hasta 33,6 TB de capacidad de almacenamiento físico en discos FC, y 84 TB de capacidad de almacenamiento físico en discos Serial Advanced Technology Attachment (SATA), además de potentes funciones de gestión del sistema y de protección y gestión de datos.

Disponen de Redundant Array of Inexpensive Disks (RAID) 0, 1, 5 y 6, así como discos de intercambio (*Hot Spare*), dándonos la posibilidad de diferentes configuraciones en cuanto al tema de protección de datos se refiere. Un RAID es una forma de presentar al sistema operativo un único volumen a partir de varios discos. Puede realizarse mediante *software* o *hardware*, siendo este último de un

rendimineto mucho mayor, aunque el precio de su implementación también es más alto. Dependiento del tipo de implementación (0, 1, 5, 6) el RAID posee una serie de ventajas:

- ***RAID 0 (Data Striping)*:** Distribuye los datos equitativamente entre dos o más discos sin información de paridad que proporcione redundancia. Es importante señalar que el RAID 0 no era uno de los niveles RAID originales y que no es redundante. El RAID 0 se usa normalmente para incrementar el rendimiento, aunque también puede utilizarse como forma de crear un pequeño número de grandes discos virtuales a partir de un gran número de pequeños discos físicos.

- ***RAID 1 (Data mirroring)*:** Crea una copia exacta de un conjunto de datos en dos o más discos. Esto resulta útil cuando el rendimiento en lectura es más importante que la capacidad. Un conjunto RAID 1 sólo puede ser tan grande como el más pequeño de sus discos. Un RAID 1 clásico consiste en dos discos en espejo, lo que incrementa exponencialmente la fiabilidad respecto a un solo disco, es decir, la probabilidad de fallo del conjunto es igual al producto de las probabilidades de fallo de cada uno de los discos (pues para que el conjunto falle es necesario que lo hagan todos sus discos).

- ***RAID 5*:** Usa división de datos a nivel de bloques, distribuyendo la información de paridad entre todos los discos miembros del conjunto. El RAID 5 ha logrado popularidad gracias a su bajo coste de redundancia. Generalmente, el RAID 5 se implementa con soporte hardware para el cálculo de la paridad, tolerando el fallo de un disco.

- ***RAID 6*:** Amplía el nivel RAID 5 añadiendo otro bloque de paridad, por lo que divide los datos a nivel de bloques y distribuye los dos bloques de paridad entre todos los miembros del conjunto, tolerando el fallo de dos discos.

Aunque existen otros tipos de RAID[15], no son soportados por las controladoras DS4700.

Dados nuestros propósitos y teniendo en cuenta que tenemos que mantener un equilibrio entre seguridad y rendimiento, la mejor opción en nuestro caso es el RAID 5, manteniendo discos de intercambio en proporcion 1/25 según recomendaciones del fabricante.

Actualmente el IFCA dispone de dos chasis completos, uno con discos de 500 GB y otro de 750 GB SATA, que después de la implementación de los RAID 5 y la asignación de los discos de intercambio (*Hot Spare*), nos deja un volumen total de

Figura 4.2: Controladora DS4700

130 TB para el sistema de ficheros, disponibles para el almacenamiento de datos de los diferentes proyectos y grupos en el IFCA.

4.1.2. Instalación de Servidores X3650

Una vez que tuvimos instalada la SAN (controladora DS4700 y cabinas EXP810), se procedió a la implementación de los servidores de acceso, que son la vía de comunicación entre el resto del cluster y la propia SAN. Los servidores dedicados para este propósito son dos IBM X3650 (ver figura 4.3).

Figura 4.3: Servidor X3650

Dado que la disponibilidad y acceso a los datos es un tema crítico, en este tipo de servidores se puede implementar una serie de módulos, para conseguir una tolerancia a fallos de red. En el caso de IBM este controlador se llama RDAC y da la posibilidad de gestionar dos o más caminos de acceso a una SAN. Este controlador se encarga de gestionar el acceso usando uno u otro camino, en función de su disponibilidad, pero nunca ambos a la vez (en el caso de que uno de los accesos a la SAN fallase siempre dispondríamos de otro de *backup* que nos permita acceder).

El controlador *RDAC* consta de dos modulos :

1. ***mppUpper*:** El módulo es cargado antes que cualquier controladora de red (HBA) se anuncie al sistema y evitando que los volumenes que se cargan de la SAN sean anunciados al sistema operativo a través de más de una interfaz de red.

2. ***mppVhba*:** Este módulo crea un Host Bus Adapter (HBA) virtual cargado en el sistema operativo antes que los HBA's reales. El HBA virtual es el responsable de agrupar los diferentes caminos al mismo volumen, organizando el trafico I/O y presentándoselo al sistema operativo como un único dispositivo.

Con la instalación de estos módulos, se crea un nuevo kernel de arranque del sitema operativo con soporte para el controlador RDAC (ver salida: 4.1.1):

```
title Red Hat Linux (2.6.18-53.el5) with MPP support
                root (hd0,0)
                kernel /vmlinuz-2.6.18-53.el5 ro root=LABEL=/1
                initrd /mpp-2.6.18-53.el5.img
```

Salida 4.1.1, Nuevo Kernel con soporte para módulos MPP, Sep. 2008

Ejecutando el comando ***#mppUtil*** (proporcionado por los módulos anteriores), podemos ver los dispositivos que maneja el módulo ***mppUpper*** (ver salida 4.1.2):

```
      H5C0T1  Active          Active  GPFS_02-GAES_01
             H3C0T1L064  Up              H3C0T2L064  Up
             H4C0T1L064  Up              H4C0T2L064  Up
             H3C0T1L065  Up              H3C0T2L065  Up
             H4C0T1L065  Up              H4C0T2L065  Up
             ....................................
             H3C0T1L093  Up              H3C0T2L093  Up
             H4C0T1L093  Up              H4C0T2L093  Up
             H3C0T1L094  Up              H3C0T2L094  Up
             H4C0T1L094  Up              H4C0T2L094  Up

        H5C0T2  Active          Active  GPFS_03-GAES_02
             H3C0T4L128  Up              H3C0T5L128  Up
             H4C0T4L128  Up              H4C0T5L128  Up
             H3C0T4L129  Up              H3C0T5L129  Up
             H4C0T4L129  Up              H4C0T5L129  Up
             ....................................
             H3C0T4L173  Up              H3C0T5L173  Up
             H4C0T4L173  Up              H4C0T5L173  Up
             H3C0T4L174  Up              H3C0T5L174  Up
             H4C0T4L174  Up              H4C0T5L174  Up
```

```
  Missing Arrays
There are no missing arrays
```

Salida 4.1.2, Cada uno de los diferentes caminos de acceso a la SAN H5C0T1 para la primera cabina y H5C0T2 para la segunda. La columna de la izquierda es para el acceso através de la primera controladora (H3C0T1), y la de la derecha a través de la segunda (H3C0T2). Sep. 2008.

El módulo ***mppVhba***, crea un dispositivo virtual con los diferentes caminos de acceso, presentándose al sitema operativo como un único dispositivo físico. Si usamos el comando `lsscsi`, el cual nos muestra los dispositivos Small Systems Computer Interface (SCSI) que tenemos en nuestro sistema, obtenemos la salida 4.1.3:

```
[5:0:1:64]   disk    IBM      VirtualDisk      0916  /dev/sdn
[5:0:1:65]   disk    IBM      VirtualDisk      0916  /dev/sdo
.........................................................
[5:0:1:93]   disk    IBM      VirtualDisk      0916  /dev/sdaq
[5:0:1:94]   disk    IBM      VirtualDisk      0916  /dev/sdar
[5:0:2:128]  disk    IBM      VirtualDisk      0916  /dev/sdas
[5:0:2:129]  disk    IBM      VirtualDisk      0916  /dev/sdat
.........................................................
[5:0:2:173]  disk    IBM      VirtualDisk      0916  /dev/sdcl
[5:0:2:174]  disk    IBM      VirtualDisk      0916  /dev/sdcm
```

Salida 4.1.3, Mapeo de los virtualdisk que permite identificar el hardware SAN, al que se está accediendo. El HBA virtual muestra un único dispositivo al sistema operativo, Sep 2008.

Si nos fijamos en la salida 4.1.3, vemos que quedan así inequivocamente identificada cada cabina/array/volumen, ya que se ha tenido especial cuidado al crear los RAID y volúmenes desde el software de gestión de la SAN. Así por ejemplo las cuatro cifras de la izquierda, empezando por la más alejada, nos indican, primero el dispositivo Host Bus Adapter (HBA) (5, en este caso el el HBA virtual creado anteriormente), la tercera cifra, (1 ó 2) es el número de la cabina de la SAN que esta exportando el volumen, y el siguiente número, (64, 65...) es el número asociado a cada disco o Logical Unit Number (LUN) asignado por el *hardware*.

4.2. Instalación de Software: General Parallel File System (GPFS)

Como se ve en la salida 4.1.3, la cantidad de volumenes virtuales que ve el sitema ronda los 174, cantidad de disco que sería ingobernable si fueran instalados como unidades independientes de almacenamiento. General Parallel File System (GPFS) nos proporciona una solución a este problema ya que es un disco compartido de alto rendimiento (sistema de ficheros para clusters) desarrollado por IBM. GPFS como otros sistemas de ficheros para clústers, proporciona acceso de alta velocidad concurrente a aplicaciones ejcutándose en múltiples nodos. Puede utilizarse con clusters AIX, AIX 5L, Linux, o un híbrido formado por AIX y nodos Linux. Además de proporcionar un sistema de almacenamiento de ficheros, GPFS posee herramientas para la gestión y la administración de clusters GPFS y permite accesos compartidos al sistema de ficheros desde sistemas GPFS remotos.

GPFS soporta accesos a datos de alta velocidad desde uno o múltiples nodos. La configuración existente de mayor tamaño supera los 2.000 nodos. GPFS esta presente en Linux desde 2001.

La instalación de este software, hace posible que los diferentes volúmenes virtuales montados en los servidores, desde la SAN, sean vistos por el sistema operativo como un único volúmen. El usuario ve de manera transparente así un volumen genérico.

GPFS implementa además un sistema propio de tolerancia a fallos sobre el RAID *harware* que ha implementado la SAN, por lo que existe una doble tolerancia a los posibles fallos y a la pérdida de datos. Aunque esta pérdida es posible, en el IFCA desde la implementación del GPFS no hemos tenido ningún problema del cual no se haya podido recuperar toda la información.

Una de las pocas desventajas, es que este *software*, solo está soportado por IBM para distibuciones de Linux, como son *RedHat Enterprise* y *Suse Enterprise*. De todas formas es posible instalarlo en otras distribuciones, gracias a la universalidad del kernel de linux. Hemos conseguido instalarlo en sistemas operativos como Debian, sobre máquinas virtuales, aunque aún no hemos realizado pruebas de rendimiento.

En el IFCA, la mayoría de los nodos usan como sistema operativo *Scientific Linux CERN (SLC)*. Durante la implementación de GPFS sobre este sistema tuvimos varios problemas de compatibilidad. El SLC es una modificación de *RedHat Enterprise Linux* realizada por el CERN para usos científicos. En principio no debería de haber existido ningún problema de instalación, pero GPFS para cerciorarse de que se esta instalando sobre un sistema compatible, busca dentro del

fichero `/etc/redhat-release` el nombre *Red Hat Enterprise*. Si este nombre no se encuentra en dicho fichero, como es en el caso del SLC, el software no puede ser instalado. Además GPFS es muy sensible a las diferentes versiones de kernel de cada nodo teniendo que ser compilado para cada diferente versión. Los 138 nodos del IFCA soportan kernels distintos por lo cual hubo que realizar tres compilaciónes del *software*.

Un vez que el *software* ha sido instalado en todos los nodos, hay que proceder a la creación del cluster GPFS. Este proceso resulta largo y tedioso dependiendo de la cantidad de volúmenes que vayan a crearse y de la cantidad de nodos que tengamos en el cluster. El procedimiento es el siguiente:

1. Asignación de los *quorum nodes* (servidores GPFS), en nuestro caso los servidores X3650.
2. Asignación de los Network Share Disk (NSD)'s (volúmenes virtuales montados en los servidores de GPFS), a los diferentes volúmenes que se van a crear.
3. Inclusión de los nodos en el Cluster GPFS.
4. Inicio de los servicios GPFS en todos los nodos.
5. Configuración de los volúmenes de datos creados en los nodos del cluster.

Ejecutando el comando ***#df -k*** obtenemos los volumenes actualmente montados en este sistema (ver salida 4.2.1).

```
#df -k
Filesystem     1K-blocks      Used       Available Use\% Mounted on
/dev/gpfsifca 14635865600  864794624   13771070976 6\%  /gpfs_ifca
/dev/gpfsgaes 114183861248 69595952128 44587909120 61\% /ifca.es
```

Salida 4.2.1 Sistemas de ficheros GPFS, montados en el IFCA, Sep. 2008.

Ejecutando el comando ***#mmlscluster***, implementado por el software GPFS nos muestra los nodos que pertenecen a este cluster y alguna de sus características (ver salida 4.2.2).

```
#mmlscluster
GPFS cluster information
========================
  GPFS cluster name:         gpfs.xxx.xx
  GPFS cluster id:           13948294495582669798
  GPFS UID domain:           gpfs.xxx.xxx
  Remote shell command:      /usr/bin/ssh
  Remote file copy command:  /usr/bin/scp

GPFS cluster configuration servers:
-----------------------------------
  Primary server:    gpfs.xxx.xxx
  Secondary server:  gpfs.xxx.xxx

 Node  Daemon node name   IP address     Admin node name   Designation
--------------------------------------------------------------------------
   1   gpfsx.xxx.xxx    193.xxx.xxx.xxx   gpfs.xxx.xxx      quorum
   2   gpfsx.xxx.xx     193.xxx.xxx.xxx    gpfs.xxx.xxx     quorum
   5   ingrid02.ifca.es   10.10.3.2        ingrid02.ifca.es
   7   ingrid01.ifca.es   10.10.3.1        ingrid01.ifca.es
  .....................................................
  20   ingrid15.ifca.es   10.10.3.15       ingrid15.ifca.es
  21   ingrid16.ifca.es   10.10.3.16       ingrid16.ifca.es
  22   cms01.ifca.es      10.10.4.1        cms01.ifca.es
  .....................................................
  48   cms27.ifca.es      10.10.4.27       cms27.ifca.es
  49   cms28.ifca.es      10.10.4.28       cms28.ifca.es
  52   eifca14.ifca.es    10.10.2.14       eifca14.ifca.es
  55   eifca17.ifca.es    10.10.2.17       eifca17.ifca.es
 100   egaes04.ifca.es    10.10.5.4        egaes04.ifca.es
 101   egaes05.ifca.es    10.10.5.5        egaes05.ifca.es
 .....................................................
 111   egaes18.ifca.es    10.10.5.18       egaes18.ifca.es
 112   pool01prv.ifca.es  10.10.0.64       pool01prv.ifca.es
 113   pool02prv.ifca.es  10.10.0.65       pool02prv.ifca.es
 114   pool03prv.ifca.es  10.10.0.66       pool03prv.ifca.es
 115   pool04prv.ifca.es  10.10.0.67       pool04prv.ifca.es
 116   stormprv.ifca.es   10.10.0.63       stormprv.ifca.es
```

Salida 4.2.2, Listado de nodos del Cluster GPFS IFCA ejecutando ***#mmlscluster***, Sep. 2008.

GPFS es un software complejo, que incorpora mas de cien comandos para su control, además de contar con infinidad de parámetros que pueden ser modificados para adecuarlo al tipo de aplicaciones que se van a ejecutar sobre él. En el caso del IFCA, podemos ver en la salida 4.2.3 la configuración que hemos establecido para nuestros propósitos:

```
#mmlsconfig
Configuration data for cluster gpfs.ifca.es:
--------------------------------------------
clusterName gpfs.ifca.es
clusterId 13948294495582669798
clusterType lc
autoload yes
minReleaseLevel 3.2.0.3
dmapiFileHandleSize 32
subnets 10.10.0.0
maxblocksize 1M
pagepool 256M
[cms01,cms02,cms03,cms04,cms05,cms06,cms07,cms08,cms09,cms10,cms11,
cms12,cms13,cms14,cms15,cms16,cms17,cms18,cms19,cms20,cms21,cms22,
cms23,cms24,cms25,cms26,cms27,cms28]
pagepool 1024M
[gpfs01,gpfs02]
pagepool 512M
maxMBpS 400
[common]
prefetchThreads 72
worker1Threads 92
[gpfs01,gpfs02]
worker1Threads 478

File systems in cluster gpfsxx.ifca.es:
---------------------------------------
/dev/gpfs_gaes
/dev/gpfs_ifca
```

Salida 4.2.3, Configuración de las variables de GPFS ***#mmlsconfig***, Sep. 2008.

Como vemos en la salida anterior, dependiendo de las características *hardware* de cada máquina, diferentes valores pueden ser aplicados.

Una de las ventajas de GPFS frente a otros sistemas de ficheros, es su paralelismo, es decir, cualquiera de las operaciones que se llevan a cabo puede ser

realizada por cualquiera de los nodos que pertenen el cluster. Si hay que añadir un nuevo NSD al sistema de ficheros, GPFS asignará a cualquiera de los nodos del cluster el formato. De esta forma, cualquiera de los nodos podrá llevar a término la operación asignada. Esto implica que el acceso entre todos los nodos del cluster ha de ser excelente en cuanto a capacidades de red, ya que las latencias o fallos de red pueden provocar un mal funcionamiento de algunas operaciones, y dada la cantidad de nodos que puede haber (un maximo de 2000), serían muy difíciles de depurar. Una buena configuración de los parámetros de red en los diferentes nodos, así como una adecuada configuración en los *switches* (*spaning tree*, *Jumbo Frames*, etc) y demás electrónica de red, es esencial para un correcto funcionamiento del sistema.

La seguridad entre todas las máquinas que pertenecen al cluster es una prioridad, ya que a diferencia de otros sistemas de ficheros, donde el servidor exporta por red al resto de nodos parte o todo el sistema (Network File System (NFS), Andrew File System (AFS), Zettabyte File System (ZFS)), GPFS lo monta directamente en todos sus nodos. Éstos acceden directamente a él, teniendo el usuario *root* de cada nodo todos los privilegios sobre el sistema de ficheros. Los nodos comparten claves Shecure Shell (SSH), para que la comunicación entre ellos no dependa de contraseña, es por esto que todos los nodos se encuentran en una red privada compartiendo claves ssh (ver salida 4.2.2), minimizando en lo posible accesos desde el exterior.

Se prevee que tanto la capacidad de almacenamiento como el número de nodos se duplique durante el año 2009.

Capítulo 5

Acceso a datos CMS

Durante estos últimos años, el IFCA ha sufrido un proceso evolutivo buscando la mejora y la optimización de su sistema de almacenamiento. Durante este proceso dos de estas soluciones han sido probadas: Una basada en DPM en una primera fase de adaptación a las necesidades del experimento y otra basada en StoRM + GPFS como solución actual, debido a su total compatibilidad con el *hadrware* y *software*, que dispone el IFCA.

Durante este tiempo, sobre todo desde que se dispone de una solución de almacenamiento estable, se han realizado bastantes puebas e investigaciones orientadas a la mejora de estos servicios en cuanto a calidad, velocidad y disponibilidad.

5.1. Integración del Sistema de Almacenamiento del IFCA en CMS

No serían de ninguna utilidad todas las transferencias y datos de los que hemos hablado en el apartado anterior si no fueramos capaces de ponerlos a disposición de los científicos que quieran analizarlos. Es necesaria la implementación de un sistema de un SE soportado por CMS, que implemente la interface SRM, para que diferentes sistemas como Disk Pool Manager (DPM)[1], dCache[2], StoRM[3] o CERN Advance Storage Manager (CASTOR)[4] puedan comunicarse sin problemas.

[1]DPM, https://twiki.cern.ch/twiki/bin/view/LCG/DpmAdminGuide
[2]dCache, http://www.dcache.org/
[3]StoRM, http://storm.forge.cnaf.infn.it/
[4]CASTOR, http://www.cern.ch/castor/

5.1.1. Disk Pool Manager (DPM)

DPM es un sistema de almacenamiento desarrollado dentro del proyecto LCG como una evolución de CASTOR, para ser implementada en centros pequeños. Fácil de instalar y mantener. Este *Storage Element* se implementó en el IFCA durante los primeros pasos del proyecto (años 2006 - 2007) dada la sencillez de instalación y gestión, mientras se decidía y tramitaba la compra de un sistema de almacenamiento más estable y fiable.

DPM es un sistema de almacenamiento basado en disco. Implementa las versiones de SRM v1 y SRM v2, que como hemos indicado anteriormente (ver sección 3.2.4) es fundamental para su uso dentro del proyecto LCG, y particularmente en CMS. Consta de un nodo servidor (*head node*) y varios servidores de disco (ver figura 5.1). El nodo servidor se encarga de crear un sistema de ficheros sobre todos los discos que estén montados en los servidores de disco. De esta forma un cliente que acceda el sistema de ficheros, a través de los comandos SRM o los propios de DPM, ve un único volumen.

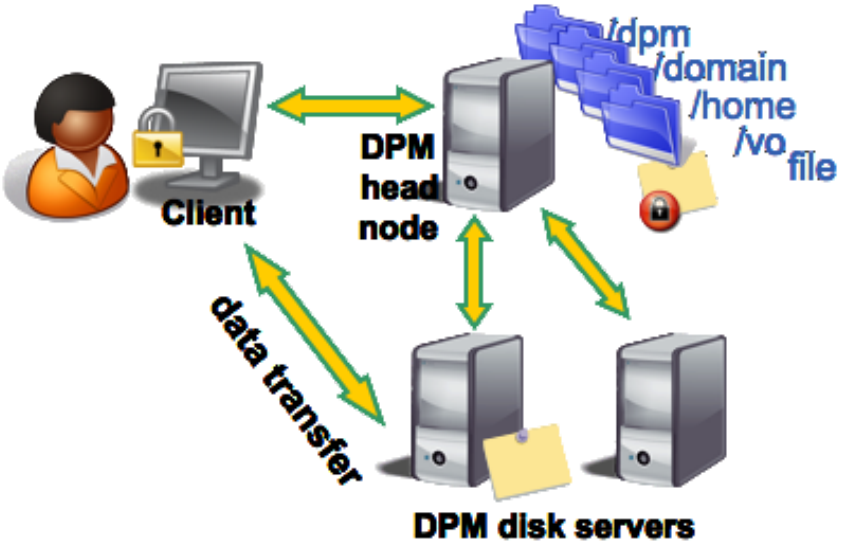


Figura 5.1: Esquema de funcionamiento de Disk Pool Manager (DPM)

El espacio se puede ir incrementando añadiendo nuevos servidores. Para asegurar la integridad de los datos cada servidor debe llevar su propia controladora RAID. Se llegaron a tener 8 servidores de disco HPML 320 a los cuales se les adaptó dos cajas ADAPTED SATA con capacidad para 4 discos cada una de ellas, con discos de 500 GB, 750 GB y 1 TB). Entre los problemas que se encontraron está el hecho de que las controladoras RAID no soportaban volúmenes mayores de 2 TB, de modo que si se usaban discos de mucho tamaño se perdía mucho espacio a la hora de poder implementar un RAID con redundancia de datos (RAID 5 o RAID 6) y si se

daba prioridad al volumen de almacenamiento se perdía ésta redundancia. Por otro lado la heterogeneidad de las soluciones *hardware*, los agresivos requerimientos de lectura y escritura y la gestión no óptima por parte de DPM no facilitan en ninguna medida la restauración del sistema tras la pérdida de datos.

5.1.2. StoRM

StoRM es una solución SRM orientado a sistemas de almacenamiento basados en discos. Implementa la interface SRM v2, con lo que nos posibilita su uso dentro de CMS. Está diseñado para su implementacion bajo un sistema POSIX, como puede ser EXT3, NFS, GPFS, etc, y está especiálmente diseñado para proporcionar un alto rendimiento sobre GPFS. Obviamente éste es uno de los principales motivos por los que ha sido elegido como solución de almacenamiento en el IFCA.

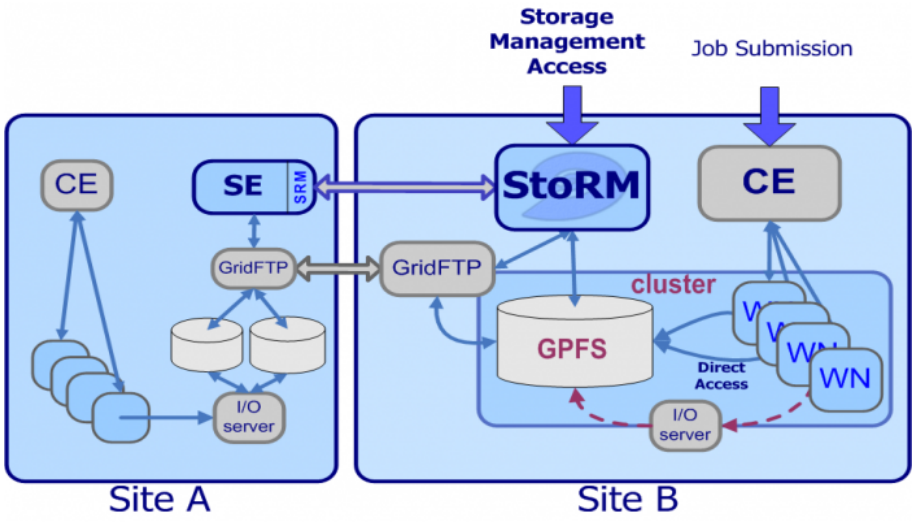


Figura 5.2: Esquema de funcionamiento de StoRM

StoRM consta de tres partes (ver figura 5.2):

- ***FrontEnd:*** que es el recibe y administra la peticiones I/O almacenandolas en una base de datos (mySql en este caso).
- ***Base de Datos (MySql):*** Es la encargada de almacenar de forma transitoria las peticiones de transferencia.
- ***BackEnd:*** que comprueba el acceso al sistema de ficheros definiendo los accesos según su configuración, define las variables de configuración del *Storage Element*, como puerto de eschucha (8444 por defecto), dirección del servicio gridFtp, etc, es también el encargado de gestionar la interfaze SRM.

5.1.3. Integración de cluster de almacenamiento en CMS

Como se ha descrito en la sección 4, el IFCA dispone como sistema de almacenamiento de una infraestructura hardware basada en controladoras DS4700 con cabinas de disco EXP810 de IBM y como solución software GPFS. Con ésta configuración, la mejor opción para integrar este sistema de almacenamiento dentro del modelo computacional de CMS es StoRM, el cual implementa la interfaz SRM que necesitamos para poder comunicar nuestro *Storage Element* con el resto de los que se encuentran dentro del proyecto CMS. La elección del *Storage Element* a emplear fue motivada por la siguientes razones:

I. DPM, dChache, Castor y implementan su propio sistema de ficheros. Puesto que GPFS ya nos proporciona un sistema de ficheros optimizado y global, las soluciones anteriores provocaría una gran perdida de rendimiento I/O.

II. CASTOR y dCache son bastante complejos de implementar y tediosos en su mantenimiento.

III. DPM, aún siendo de fácil implementación, no responde a nuestras necesidades en términos de fiabilidad y rendimiento (ver sección 5.1.1)

IV. StoRM (ver sección 5.2) tiene una implementación sencilla y poco mantenimiento al no tener que implementar ningun sistema de ficheros.

V. StoRM está diseñado para trabajar sobre sistemas de ficheros posix y especialmente sobre GPFS.

Si bien es verdad que StoRM es un sistema de nuevo desarrollo, es la solución de almacenamiento que mejor se adecúa a la configuración de *hardware y software* que dispone el IFCA.

Según los compromisos que el IFCA adquirió con el proyecto CMS, durante el año 2008 deben de ponerse a disposición del experimento 100 TB de disco y se debería de llegar a los 200 TB durante el 2009. Los compromisos relativos al 2008 han sido cumplidos por IFCA aportando al experimento unos 107 TB en estos momentos.

5.2. Desarrollo

Aunque como hemos citado anteriormente, varios *Storage Element* han sido implementados para el acceso a datos dentro del experimento CMS en el IFCA, nos

centraremos en el desarrollo realizado sobre StoRM y el sistema de almacenamiento basado en la SAN de IBM y GPFS, ya que a parte de ser mucho más estable que el anterior, es con el que tendremos que trabajar durante los proximos años.

5.2.1. Primera Fase

Durante la primera fase de la implementación de StoRM como *Storage Element* en el IFCA para el proyecto CMS, tanto el servicio StoRM, como el GridFtp se encontraban instalados sobre una única máquina. Como se recoge en el punto anterior, el *FrontEnd* es el encargado de aceptar un determinado numero de peticiones I/O. Este número, que en un principo está por defecto determinado a 20, puede ser modificado ya sea aumentando o disminuyendo, jugando así con la capacidad de recepción de nuestro sistema.

Se observó que si StoRM acceptaba más de 20 conexiones, el sistema comenzaba a sobrecargarse. Ésto es debido a que el Servicio GridFtp palaleliza la copia llegando hasta 20 hilos por proceso GridFtp teniendo como resultado 400 intentos de acceso al sistema de ficheros desde una misma máquina. Por encima de estas 20 conexiones comenzaban a aparecer fallos de acceso y procesos GridFtp terminados antes de tiempo, así como un consumo excesivo de memoria *swap*. Cuando el sistema se estresa por el numero de conexiones, el acceso al sistema se ralentiza, y el GridFtp aumenta el número de hilos para intentar mantener la velocidad de acceso, teniendo que reiniciar el sistema en algunas ocasiones.

5.2.2. Segunda Fase

Para solucionar el problema anterior y dado que StoRM permite separar cada uno de sus servicios en diferentes máquinas, se decidió que la mejor opción era dejar una única máquina ejecutando los procesos StoRM propiamente dichos (*FrontEnd*, *BackEnd* y base de datos) y sacar el servicio GridFtp a otra máquina.

Como el GridFtp era realmente el proceso que sobrecargaba el sistema, si lo migrabamos a otra única máquina, el servicio StoRM no tendría problemas, pero el servicio GridFtp seguiría colapsando la nueva. Se decidió entonces la instalación del servicio GridFtp no en una sino en cuatro máquinas cuyo acceso estaría balanceado a través del Domain Name Service (DNS).

Hay que puntualizar que el uso del balanceo a través de DNS no es un balanceo de carga (*Load Balancing*). En el DNS, se asigna un nombre genérico para todas las máquinas involucradas en el servicio GridFtp, y mediante el algoritmo *Round Robbin*, va asignando las peticiones por orden de colocación, cuando llega a la última

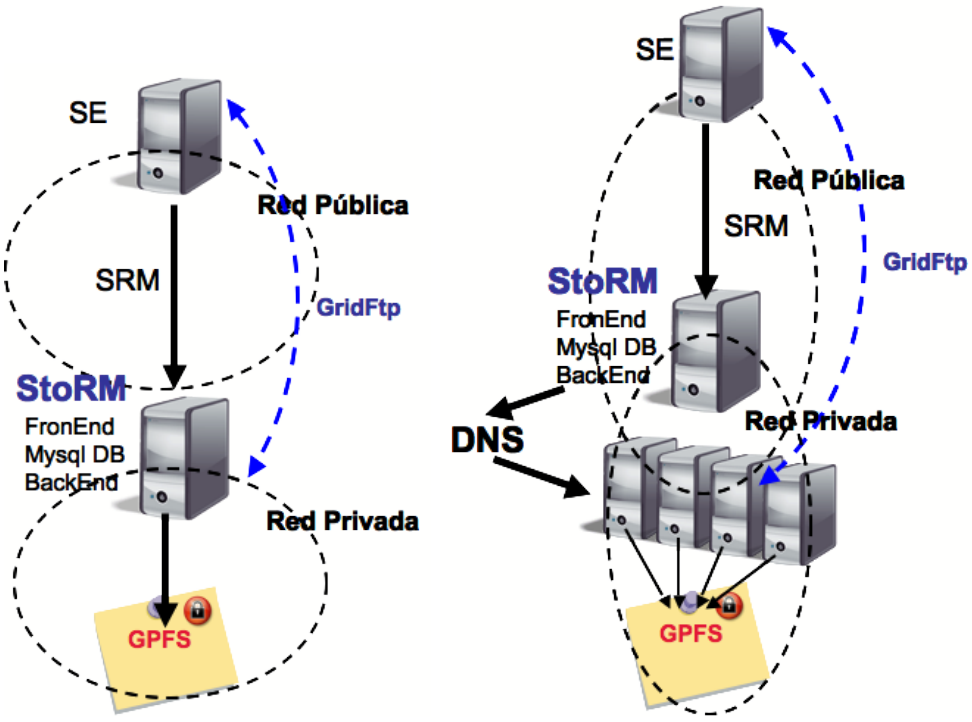


Figura 5.3: Posibles configuraciones hardware de StoRM: Con todos los servicios en un solo servidor (a) y con los servicios repartidos en varias máquinas (b)

vuelve a empezar. Así las peticiones van rotando entre la máquinas asignadas al nombre genérico que tienen en el DNS. De este modo, la carga se comparte (*Load Sharing*) y no se balancea. Como todas la peticiones van a tener aproximadamente la misma carga, la diferencia de concepto entre compartir y balancear no supone problema en este caso particular.

La configuracion del DNS, donde asignamos en nombre genérico a las máquinas que van a ejecutar el servicio GridFtp podemos verlo en la salida 5.2.1:

```
gridftp.ifca.es. IN A 19x.xxx.xxx.x21
gridftp.ifca.es. IN A 19x.xxx.xxx.x22
gridftp.ifca.es. IN A 19x.xxx.xxx.x23
gridftp.ifca.es. IN A 19x.xxx.xxx.x24
........
pool01              A 19x.xxx.xxx.x21
pool02              A 19x.xxx.xxx.x22
pool03              A 19x.xxx.xxx.x23
pool04              A 19x.xxx.xxx.x24
```

Salida 5.2.1, Configuración de los nodos GridFtp en el DNS, Sep. 2008

De esta forma cualquier conexión desde el exterior al nodo `gridftp.ifca.es` se redigirá secuencialmente a las máquinas `pool01, pool02 pool03 y pool04`, mientras que por separado, cada máquina mantiene su propia entidad, lo cual es necesario para poder instalar el *software* GPFS.

Para evitar intentos de acceso no deseados al sistema de almacenamiento, cada una de estas máquinas dispone de un cortafuegos propio además de una doble tarjeta de red. Una en *red pública* para tener acceso a StoRM y al resto de *Storage Elements* y otra dentro de la *red privada* de almacenamiento por la cual accede al sistema de ficheros `/gpfs/ifca.es`.

5.2.3. Tercera Fase

Una vez que el sistema de almacenamiento adquirió un grado de estabilidad aceptable, se detectaron dos problemas relacionados con la carga del sistema y la configuracion de red. Durante la anterior etapa se constató que con la configuración por defecto del kernel de linux estas máquinas consumían recursos de memoria que este tipo de procesos no deberían de consumir, como memoria (*swap y buffer*), hasta el punto de que en ocasiones había que reiniciarlas. También se vio que la red no estaba configurada por defecto para este tipo de máquinas pues abren muchos subprocesos de copia.

Como ya hemos mencionado, un proceso GridFtp puede estar trabajando hasta con 20 hilos por debajo, y una máquina en codiciones de carga puede ejecutar más de treinta de estos procesos, con lo que abriría $20 \times 30 = 600$ conexiones al sistema de ficheros. Teniendo en cuenta que los servidores están montados en sistemas operativos de 32 Bits, se modifican los parámetros de red, accediendo al fichero de configuración `/etc/sysctl.conf`, tal y como se ve en la salida 5.2.2.

```
net.ipv4.tcp_window_scaling = 1
net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_sack = 0
net.core.rmem_max = 1048576
net.core.rmem_default = 87380
net.core.wmem_max = 131072
net.core.wmem_default = 32768
net.ipv4.tcp_rmem = 4096 87380 1048576
net.ipv4.tcp_wmem = 4096 32768 131072
net.ipv4.tcp_mem = 65536 87380 98304
```

Salida 5.2.2, Configuración de red para los servidores GridFtp en el fichero `/etc/sysctl.conf`.

Los problemas relacionados con la memoria podían ser solucionados evitando que estas máquinas usaran la memoria *cache y buffer*, ya que los procesos GridFtp no necesitan este tipo de memoria. Simplemente se necesita que escriban directamente al sistema de ficheros. Para ello es muy importante la modificación de los parámetros pertenecientes a la memoria virtual del sistema, añadiendo las siguientes líneas al fichero anterior, `/etc/sysctl.conf` de la salida 5.2.3:

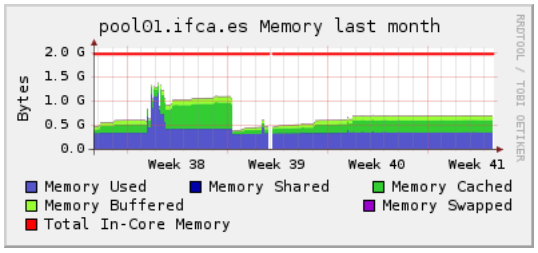
```
vm.min_free_kbytes = 65536
vm.overcommit_memory = 2
vm.overcommit_ratio = 2
vm.dirty_ratio = 10
vm.dirty_background_ratio = 3
vm.dirty_expire_centisecs = 500
```

Salida 5.2.3, Configuración de la memoria virtual, para los servidores GridFtp en el fichero `/etc/sysctl.conf`.

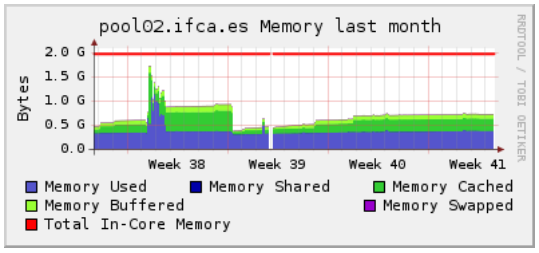
De este modo se reducen los tiempos y el porcentaje de memoria que la máquina emplea para volcar las páginas cache y *buffer* al disco. En este tipo de máquinas nos interesa que sean muy bajos pues queremos que escriban directamente al disco.

Los resultados obtenidos fueron excelentes. Ni la memoria cache ni la *buffer* han superado el GB de RAM desde la modificación de los parámetros señalados en la salida 5.2.3, dando total estabilidad al sistema (ver figura 5.4 y 5.5).

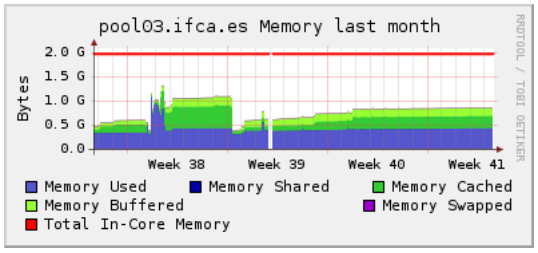
Como vemos en la figura 5.6, se consiguió transferir aproximadamente 7TB en menos de 24h, con total estabilidad tanto de la red, como del *Storage Element*.



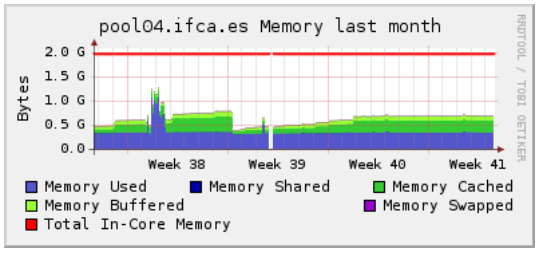
(a) Consumo de memoria en pool01.



(b) Consumo de memoria en pool02.



(c) Consumo de memoria en pool03.



(d) Consumo de memoria en pool04.

Figura 5.4: Consumo de memoria en los servidores GridFtp durante el último mes.

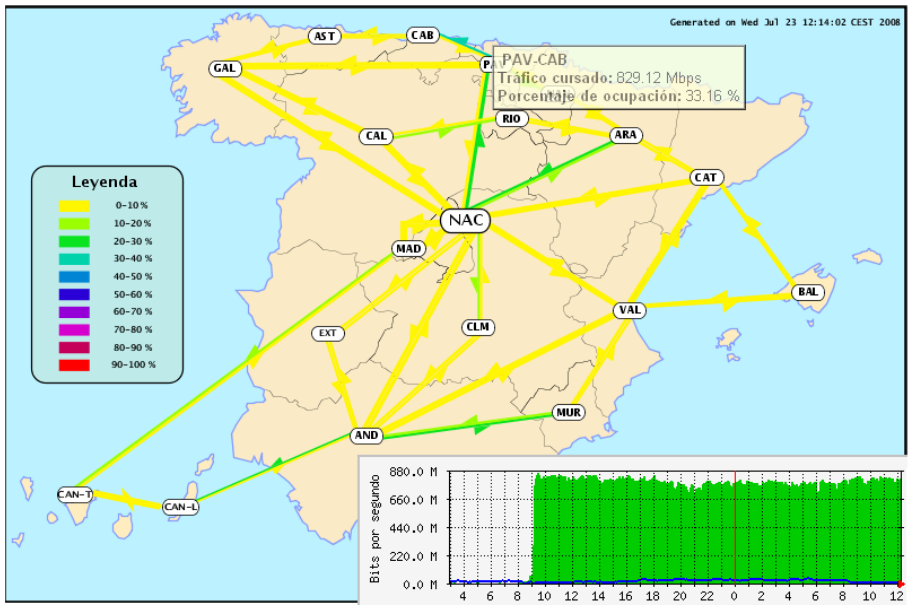


Figura 5.5: Gráfica de red, después de la mejoras introducidas en los servidores GridFtp. Tasa de transferencia a 880 Mb/s durante más de 24 horas.

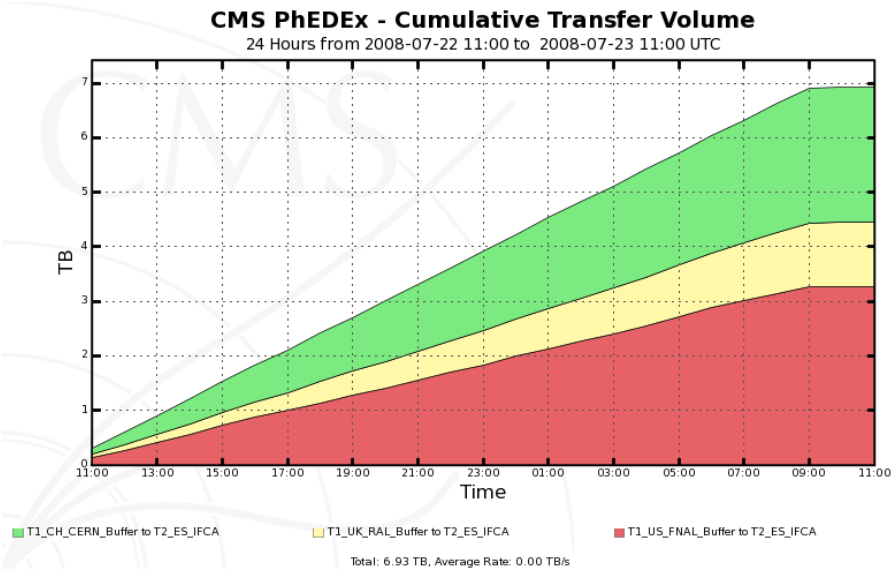


Figura 5.6: Volumen transferido por PhEDEx después de la mejoras introducidas.

5.3. Configuración local: Nodos de acceso interactivo y nodos GRID

El sistema de almacenamiento además de proporcionar espacio para las tranferencias, debe de suministrar servicio a los usuarios y máquinas locales:

- A través de nodos interactivos en los que los usuarios puedan acceder al sistema de ficheros para realizar salvaguardas de sus datos y poder ejecutar análisis directamente sobre datos del sistema de ficheros.
- A través de los WN dándoles acceso al sistema de ficheros GPFS para poder acceder a la entrada de datos.
- Resto de usuarios sin acceso al Cluster GPFS, en redes externas.

Una buena configuración de los parametros de red, de todos los nodos que pertenecen al cluster GPFS, es vital para su correcto funcionamiento, por ello hay que aplicar una serie de modificaciones a todos sus nodos a excepción de los anteriormente mencionados (`pool01, pool02, pool03 y pool04`), que dado sus características especiales (servicio GridFtp), usan sus propias configuraciones.

El resto de máquinas que hemos mencionado son principalmente *Worker Nodes)* y *User Interface*. En el IFCA, todos estos nodos pertenecientes al cluster GPFS se hayan conectados entre si a través de enlaces *Gigabit Ethernet*[16], entonces aplicaremos los siguientes parámetros al fichero anteriormente mencionado

`/etc/sysctl.conf`, que como vemos son diferentes de los aplicados a las máquinas GridFtp, vistos en la sección anterior (Ver salida 6.3.1):

```
# increase Linux TCP buffer limits
net.core.rmem_max = 8388608
net.core.wmem_max = 8388608
# increase default and maximum Linux TCP buffer sizes
net.ipv4.tcp_rmem = 4096 262144 8388608
net.ipv4.tcp_wmem = 4096 262144 8388608
# increase max backlog to avoid dropped packets
net.core.netdev_max_backlog=2500
```

Salida 5.3.1, Configuración de red, para los nodos GPFS con enlaces *Gigabit Ethernet* en el fichero `/etc/sysctl.conf`.

Las principales tareas de un *Tier-2* es la de proporcionar soporte y acceso a computación a un grupo de físicos determinados. En el IFCA se han instalado 20 nodos de acceso interactivo a los datos. Los usuarios acceden a los nodos interactivos que pertenecen al cluster GPFS y por tanto se encuentran en una red privada, a través de una máquina pasarela *User Interface (UI)*, que escucha redes públicas y la red privada del cluster GPFS (ver figura 3.1).

A su vez todas estas máquinas, al ser instaladas como servicios *User Interface*, disponen de toda la infraestructura necesaria para que los usuarios puedan enviar trabajos al GRID a través de herramientas como CRAB.

Se ha instalado una serie de nodos de este tipo, para dar una mayor versatilidad a su uso a través de Linux Virtual Server (LVS). Mediante éste procedimiento hemos conseguido que los usuarios accedan a diferentes equipos, de forma transparente para ellos, ya que todos los nodos comparte el mismo espacio de usuario, que es exportado por otro nodo que funciona como servidor NFS.

En éstos nodos tuvimos que hacer una instalación particular ya que, para que el `LVS` pueda ser ejecutado en éstos nodos, el kernel de estas máquinas tuvo que ser recompilado incluyendo el módulo lvs. Para ello hubo que instalar la última versión de kernel existente en este momento (versión 2.6.20.21), la cual difería con las instaladas en el resto de nodos del cluster. Como queríamos que estos nodos formasen parte del cluster GPFS, hubo que compilar las librerias del software especialmente para ellos. De esta forma los usuarios pueden acceder a su área de usuario local en cualquiera de las máquinas LVS y tener un acceso directo a los datos almacenados en el sistema de ficheros (GPFS), pudiendo así realizar análisis de modo interactivo

(ver figura 5.7).

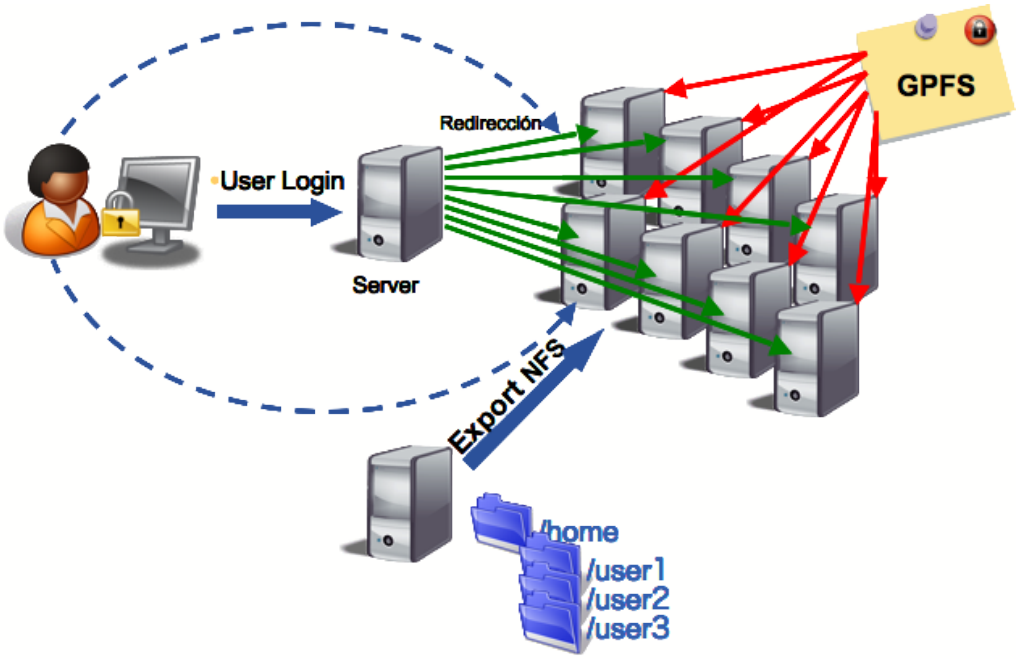


Figura 5.7: Esquema de funcionamineto del Linux Virtual Server en el IFCA.

Existen otra serie de nodos que pertenecen a la infraestructura del IFCA, incluso a la propia infraestructura GRID, pero no al cluster GPFS. Todos estos nodos pertenecen a redes públicas y su inclusión dentro del cluster prodría dar lugar a fallos de seguridad, como se ha citado anteriormente (ver sección 4.2).

GPFS tiene la capacidad de poder ser exportado por NFS desde cualquiera de los nodos que pertenecen al cluster, y esta es una solución perfecta para poder compartir el sistema de ficheros entre redes públicas y privadas, manteniendo un grado de seguridad aceptable ya que, de este modo sólo exportamos a la red pública partes del sistema de ficheros que nos interesan. Además NFS nos permite bloquear los permisos de escritura para los usuarios administradores de los nodos clientes NFS, evitando así que un fallo de seguridad en estas máquinas ponga en peligro todo el Cluster GPFS (ver figura 5.8).

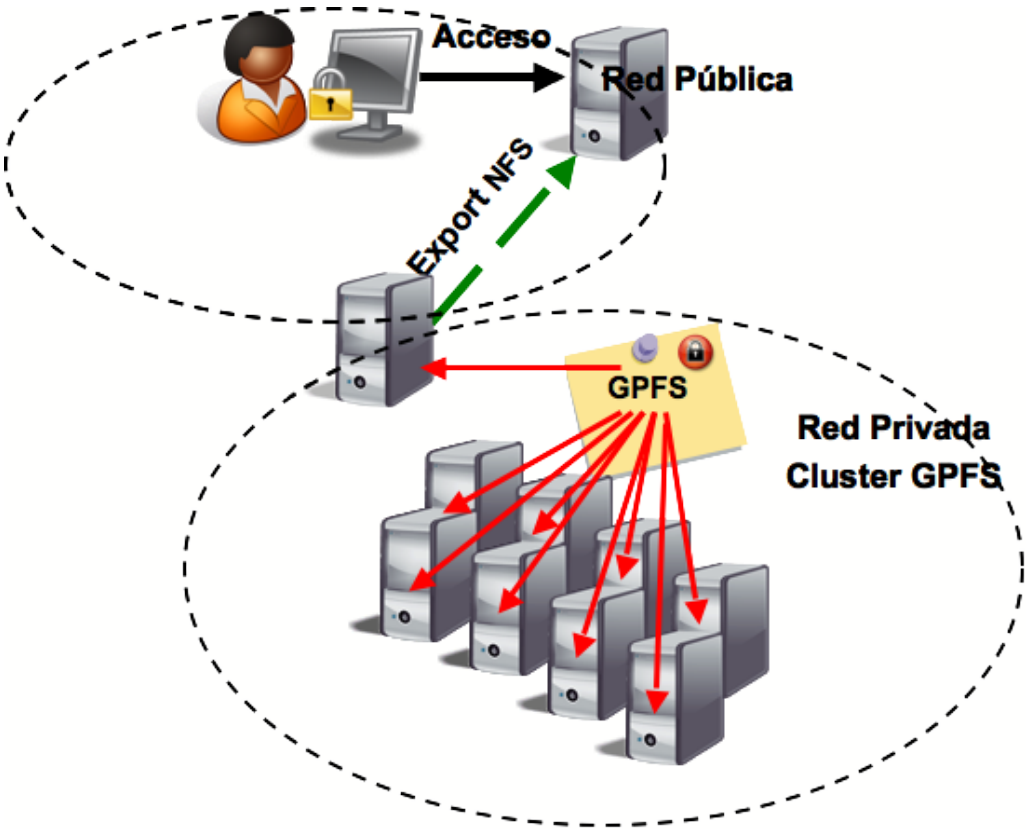


Figura 5.8: Acceso Al sistema de Ficheros para nodos en la Red Pública.

Capítulo 6

Transferencia de datos

Como se ha comentado anteriormente, el modelo computacional de CMS está fuertemente dirigido por la localización de los datos. Por tanto la correcta gestión de las transferencias es de vital importancia para el éxito del experimento. CMS ha desarrollado una herramienta propia encargada de optimizar el movimiento de datos, denominada PhEDEx.

Durante los tres últimos años se han estado realizando de forma más o mneos continuada diferentes pruebas del sistema, algunas de ellas con datos reales y otras con datos ficticios, con el fin de comprobar la fiabilidad del sistema así como la respuesta de los heterogéneos sistemas de almacenamiento y conexiones de red entre los centros participantes en el experimento:

- retos con datos reales: *Computing, Software and Analysis Challenge (CSA)*
- retos con datos ficticios: *Debug Data Transfer (DDT)*

6.1. PhEDEx

PhEDEx es la herramienta desarrollada por CMS encargada de gestionar, automatizar y optimizar el movimiento de los grandes volúmenes de datos de la colaboración. Ésta herramienta administra grandes cantidades de transferencias de datos entre el CERN y decenas de centros de computación distribuidos por el mundo. Los principales componentes de esta herramienta son (ver figura 6.1):

I. Base de datos de gestión de transferencias (*Transfer Management Database (TMDB)*).

II. Agentes de transferencia: gestionan el movimiento de los ficheros entre los diferentes sitios. Son los encargados de comprobar que la transferencia se ha

realizado correctamente mediante algoritmos de chequeo de redundancia (*Checksums*).

III. Agentes de gestión, agentes localizadores (*allocators*), encargados de asignar los ficheros a sus destinos basados en las subscripciones de cada sitio, y agentes de mantenimiento, que proporcionan información de los ficheros durante su transferencia.

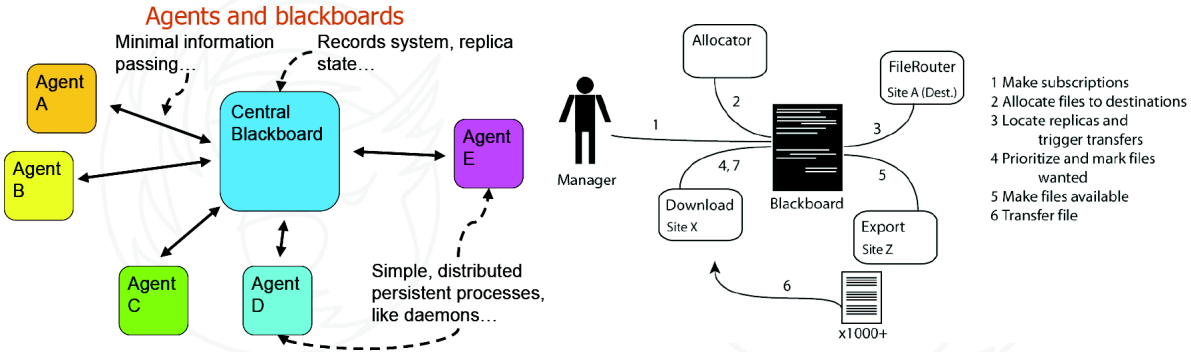
IV. Herramientas para gestionar las peticiones de transferencia.

V. Agentes locales, para gestionar los ficheros tanto locales como los producidos en los propios *Tiers*, e inyectarlos en la TMDB.

VI. Herramientas de monitorización web[1].

Cada centro participante en PhEDEx debe de correr una serie de agentes de conexión con el sistema central, de forma que cada uno de ellos realiza una pequeña parte del trabajo total (unos se encargan de las tranferencias, otros del borrado de datos, etc).

Durante los dos últimos años el número de agentes de PhEDEx ha ido cambiando, pasando de los cinco iniciales a los tres que se ejecutan actualmente y a los cuatro que se emplearán en un futuro próximo. La mayor parte de los agentes de PhEdEx se comunican a través de una *"Blackboard"*, implementada sobre una base de datos de alto rendimiento. Toda esta estructura ha sido diseñada para funcionar con una gran cantidad de accesos concurrentes a la base de datos (ver figura 6.1).



(a) Agentes de PhEDEx (b) WorkFlow de PhEDEx

Figura 6.1: Funcionamiento de PhEDEx

Los agentes son dotados de varios niveles de autonomía, siendo capaces de tomar decisiones dependiendo de las condiciones locales.

[1]Monitorización PhEDEx, http://cmsweb.cern.ch/phedex (Sep. 2008)

PhEDEx ha sufrido constantes evoluciones y versiones. Prácticamente ha sido reprogramado pasando del lenguaje Perl en el que estaba programado inicialmente a Python. También ha sido mejorado en cuanto a fiabilidad, prestaciones y agilidad en los que toda la comunidad ha participado, ya sea desarrollando o depurando. El IFCA ha participado en diversas ocasiones en la fase de pruebas y depuración de errores de varias de estas versiones.

Para comprobar la fiabilidad y calidad del servicio, se han creado dentro de PhEDEx varios modos de conexión de manera que, actualmente cada centro ejecuta al menos dos instancias de PhEDEx, es decir, los agentes de PhEDEx son ejecutados dos veces contra diferentes bases de datos, una para datos reales (instancia Prod) y otra para pruebas (instancia Debug).

Los usuarios finales que quieren analizar una serie de datos en un determinado centro pueden realizar una subscripción a través de la web de PhEDEx[2]. Un operador, denominado *Data Manager* se encarga de controlar y aprobar las subscripciones con el objetivo de evitar el llenado del *Storage Manager* y de priorizar las necesidades de la comunidad a la que da soporte.

A través de PhEDEx los centros asociados al experimento son capaces de acceder a los datos que han sido producidos en el detector CMS para su posterior reconstrucción, análisis etc. También es posible que los científicos, una vez analizados los datos, inyecten sus resultados en el sistema para compartirlos con el resto de la comunidad.

Una vez que los datos han sido descargados, el usuario de análisis debe consultar el CMS Dataset Bookkeeping System (DBS)[18]. DBS es una base de datos sincronizada con TMBD, con una interfaz de acceso web para los usuarios, de modo que pueden comprobar si un *Dataset* o un *FileBlock* se encuentran en un determinado *Tier* o no.

Este sistema ha sido capaz de mover y registrar aproximádamente 60 PB de datos reales durante los años 2007 y 2008 (ver figura 6.2). Si tenemos en cuenta que los ficheros tienen un tamaño de unos $2G$ B, obtenemos unas 30,000,000 tranferencias durante este periodo entre los diferentes centros implicados en el experimento CMS.

El IFCA por su parte durante los años 2007 y 2008 ha transferido unos $640TB$ hacia su SE y otros $210TB$ desde su SE (ver figura 6.3 y 6.4).

Esta cifra está de acuerdo con respecto al tamaño del IFCA dentro de CMS y con los compromisos adquiridos con la colaboración.

[2]http://cmsweb.cern.ch/phedex/prod/Data::Subscriptions

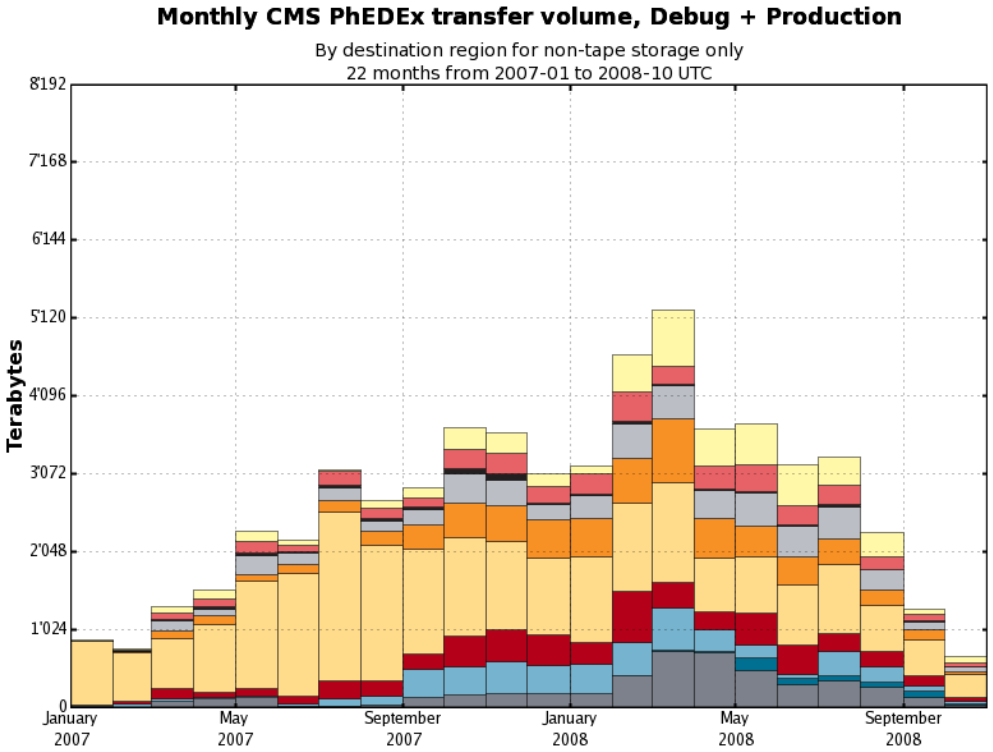


Figura 6.2: Volumen total de datos movidos por PhEDEx entre los distintos centros que participan en el experimento CMS durante los años 2007 y 2008.

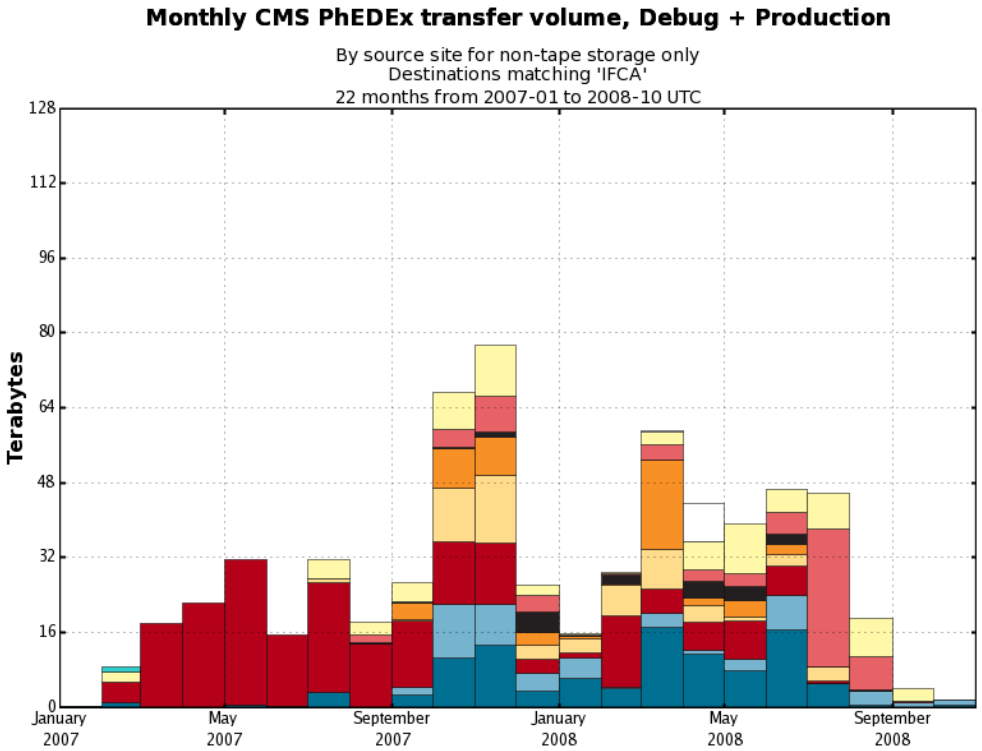


Figura 6.3: Volumen de datos transferido hacia el IFCA durante el periodo 2007 y 2008

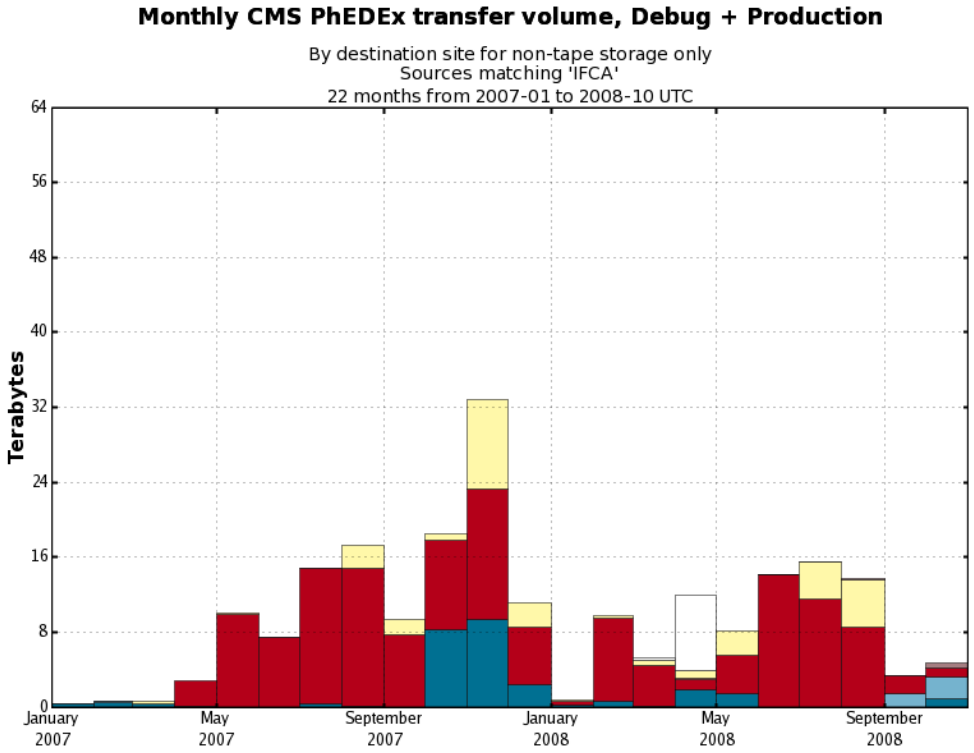


Figura 6.4: Volumen de datos transferido desde el IFCA durante el periodo 2007 y 2008

6.2. Retos de Computación, Software y Analysis

6.2.1. CSA06

El objetivo del reto CSA06 (Computing Software and Analysis Challenge 2006)[19] que tuvo lugar durante los meses de Octubre y Noviembre de 2006, era comprobar el funcionamiento de toda la cadena de *software* y computación, desde el primer procesado en el CERN, hasta el análisis de los usuarios en los *Tier-2* a un nivel del 25 % de las necesidades esperadas para el comienzo de la toma de datos a finales del 2008.

Durante este evento se transfirió al IFCA un volumen de datos cercano a los 25TB (ver figura 6.5). Aproximadamente la mitad de ellos llegaron desde el PIC, y el resto desde el Fermi National Accelerator Laboratory (FNAL). La aportación del resto de *Tier-1* fue escasa, dado que durante este periodo la topología de red de CMS se encontraba en su primer estado (ver sección 3.2.4). Estos datos, fueron almacenados DPM.

La calidad de las transferencias y la eficiencia en la copia fue muy alta, estando próxima al 90 % durante la mayoría del tiempo (ver figura 6.6). Se detectaron

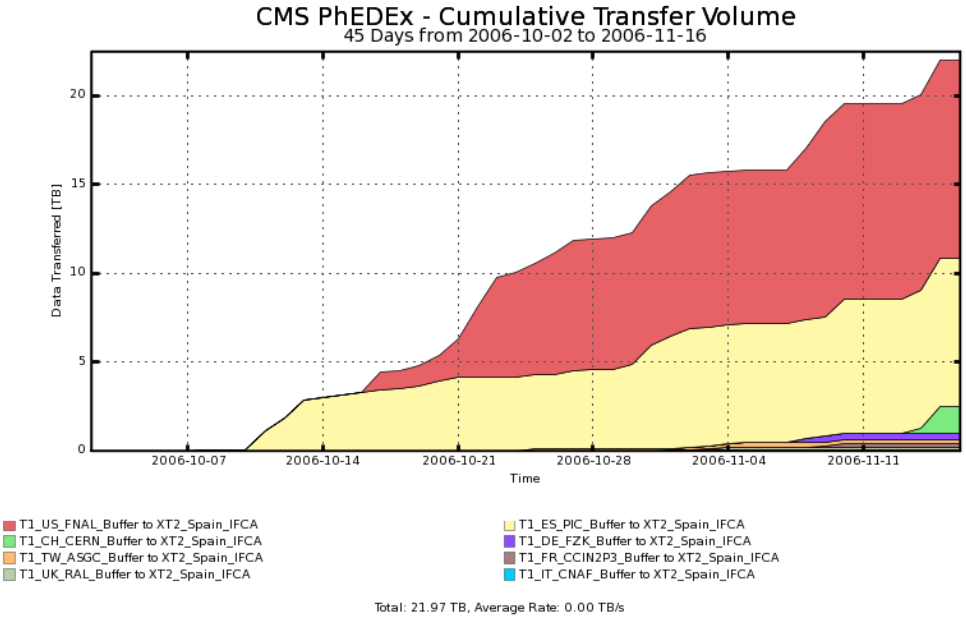


Figura 6.5: Tranferencias al IFCA desde diferentes *Tier-1* durante el evento CSA06

problemas cuya razón estaba mayoritariamente en el origen de los datos y fuera de nuestro control. Aún así la buena colaboración entre los centros facilitó en general una rápida solución.

Durante la CSA06 se ejecutaron en el *Tier-2* IFCA más de 59.000 trabajos de análisis que leían los datos almacenados previamente (ver figura 6.7).

Aproximadamente un 65 % de los trabajos se ejecutaron sin problema. El 35 % de fallos estaba en trabajos deficientemente configurados, *bugs* en los sitemas GRID tanto en el IFCA como en los servicios centrales, y problemas de lectura de datos (ver figura 6.7).

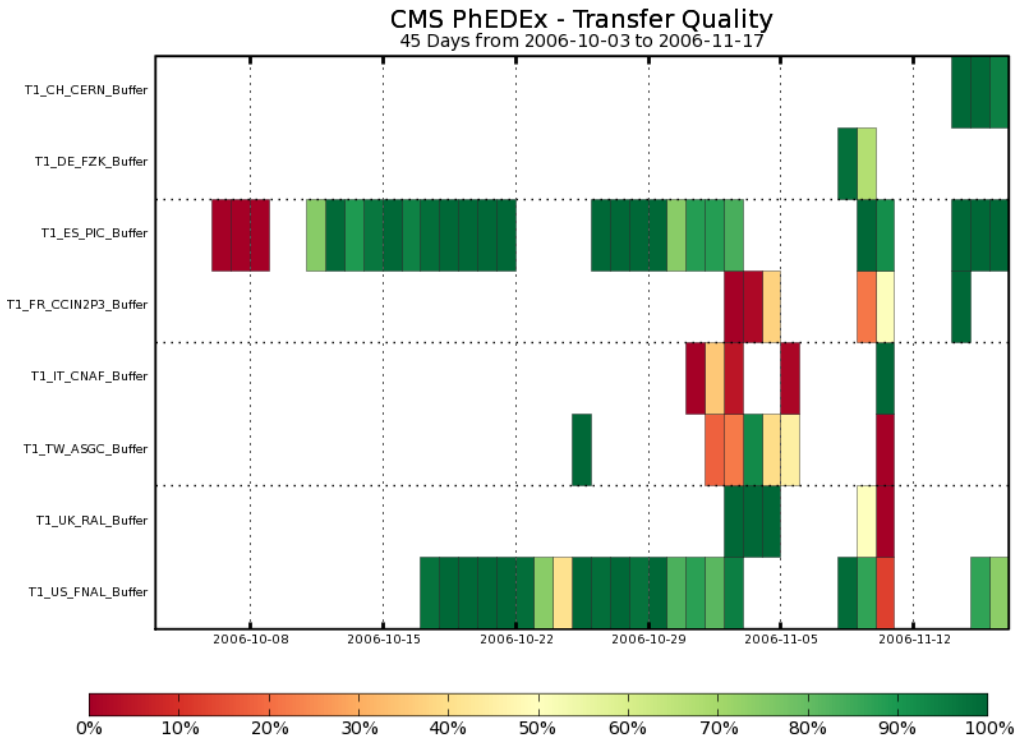


Figura 6.6: Calidad de las tranferencias en el IFCA desde diferentes *Tier-1* durante el evento CSA06

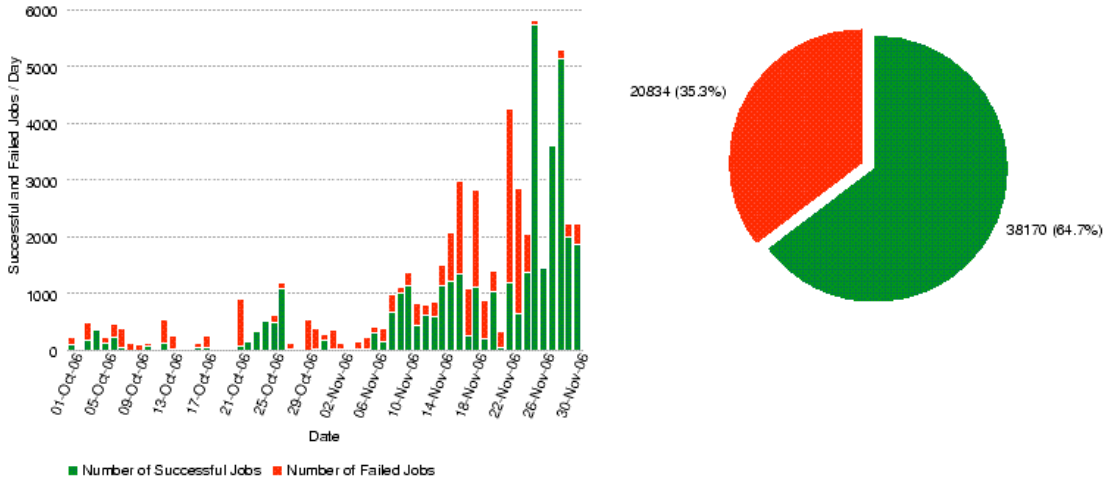


Figura 6.7: Número de trabajos ejecutados con y sin éxito en el *Tier-2* del IFCA durante la CSA06

6.2.2. CSA07

El objetivo del reto CSA07 (Computing Software and Analysis Challenge 2007)[20], que tuvo lugar durante los meses de octubre y noviembre de 2007, era comprobar el funcionamiento de toda la cadena de *software* y computación, desde el primer procesado en el CERN, hasta el análisis de los usuarios en los *Tier-2* al 50 % de las necesidades esperadas para el comienzo de la toma de datos a finales del 2008 (el doble que la CSA06). Además se intentaba probar la nueva estructura de red de CMS, dejando un poco de lado la figura de *Tier-1* regional que hasta este momento había tenido bastante relevancia.

Durante este evento fue transferido al IFCA un volumen de datos cercano a los 30TB (ver figura 6.8). Al contrario que durante la CSA06, en la CSA07 la contribución de todos los *Tier-1* de CMS a este volumen de datos fue notable, pudiendo realizarse descargas desde práticamente todos los *Tier-1* implicados en el experimento. Los datos fueron almacenados también DPM.

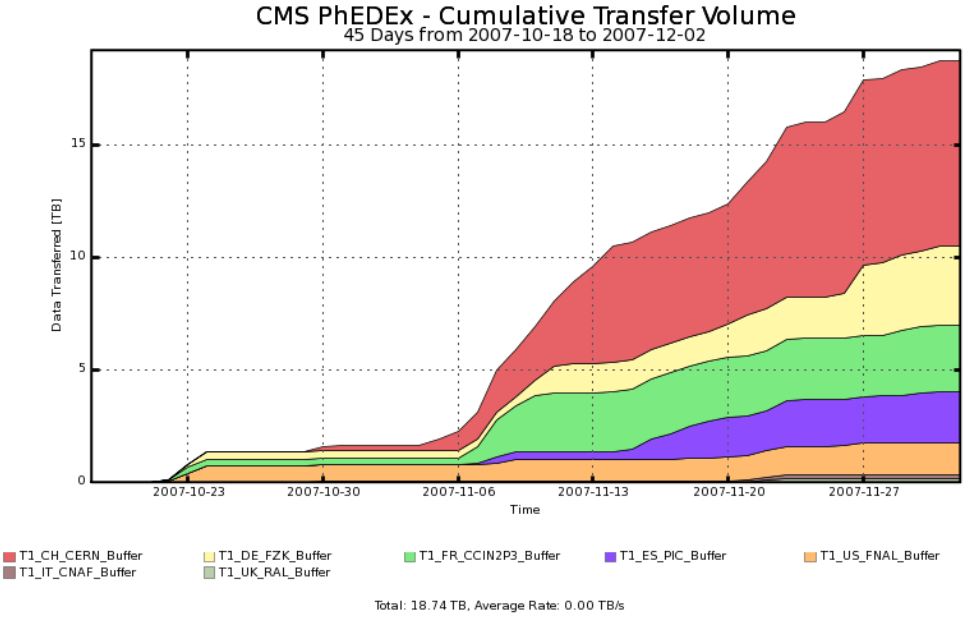


Figura 6.8: Tranferencias al IFCA desde diferentes *Tier-1* durante el evento CSA07

Fue este el momento crítico donde se comenzó a constatar que con el sistema de almacenamiento que disponíamos (DPM), no se cumplía todas las necesidades para el correcto desarrollo de nuestras operaciones dentro del experimento. La escalabilidad del sistema dependía del ingreso de nuevos servidores de disco.

Esto nos conducía a tener que realizar un rebalanceo de toda la carga del sistema de ficheros para evitar el colapso del nuevo servidor y que las tranferencias estuvieran balanceadas sobre todo el *hardware* de almacenamiento. De esta forma

se usaría todo el poder de las transferencias distribuidas, donde todos los servidores de disco acceden al sistema de ficheros, repartiendose las transferencias/carga entre todos ellos.

La calidad de las transferencias (ver figura 6.9) fue razonable encontrándose entre los mejores *Tier-2* de CMS.

Este evento coincidió además con la actualización forzosa del sistema operativo desde SLC3 a SLC4 para poder usar las herramientas desarrolladas por LCG. Esto tuvo repercusiones en el software desarrollado por el propio experimento CMS, generando algunas incompatibilidades entre diferentes librerías, teniendo como resultado la incapacidad de acceder al sistema de ficheros para los trabajos ejecutados en los nodos del IFCA durante algunos períodos del reto. De ahí que la cantidad de trabajos ejecutados sea inferior a la CSA06 (unos 34.000) con una tasa de éxito menor (ver figura 6.10).

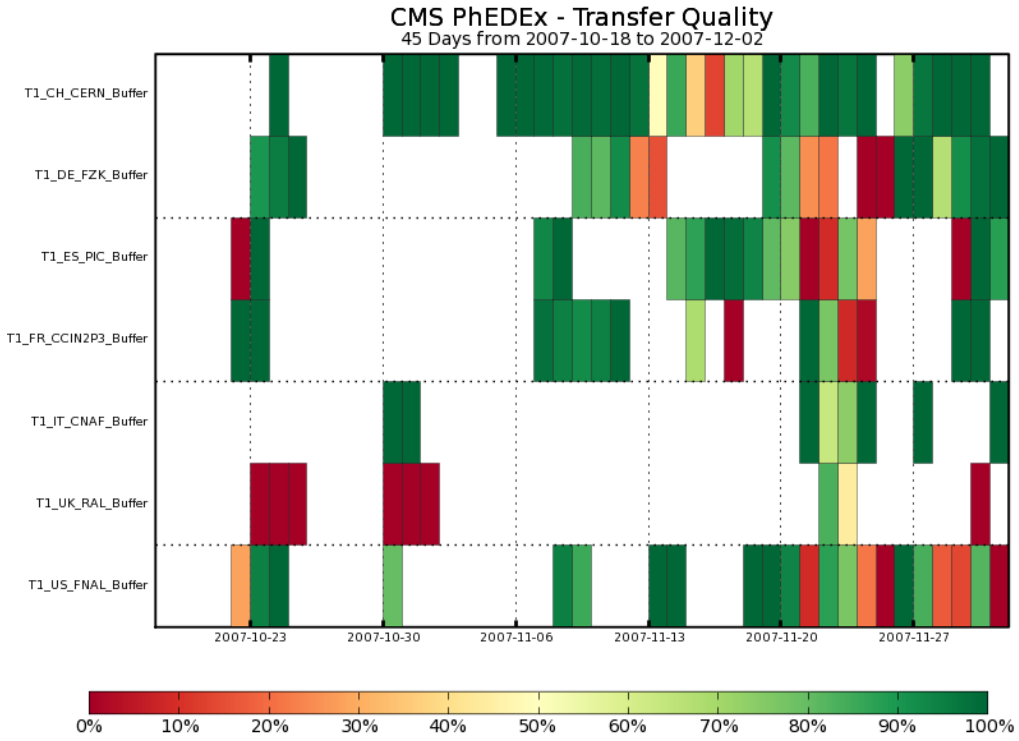


Figura 6.9: Calidad de las tranferencias en el IFCA desde diferentes *Tier-1* durante el evento CSA07

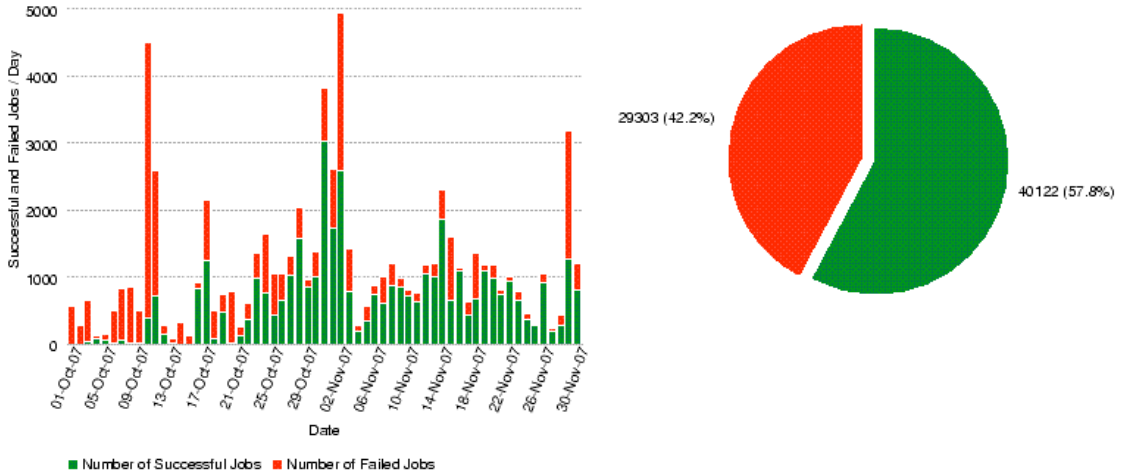


Figura 6.10: Número de trabajos ejecutados con y sin éxito en el *Tier-2* del IFCA durante la CSA07

6.2.3. CSA08

El objetivo del reto CSA08 (Computing Software and Analysis Challenge 2008)[21] que tuvo lugar durante el mes de mayo del 2008, era comprobar el funcionamiento de toda la cadena de *software* y computación, desde el primer procesado en el CERN, hasta el análisis de los usuarios en los *Tier-2*, al 100 % de las necesidades esperadas para el comienzo de la toma de datos a finales del 2008 (el cuadruple que la CSA06 y el doble que la CSA07).

Durante este evento se transfirió al IFCA un volumen de datos cercano a los 35TB (ver figura 6.11). Al igual que durante la CSA07, la contribución de todos los *Tier-1* a este volumen de datos fue notable, pudiendo realizarse descargas desde todos los *Tier-1*, sin excepción, implicados en el experimento. Los datos fueron almacenados en el nuevo hardware que se instaló en el IFCA a principios de Febrero del 2008 (ver seccion 4).

En la imagen 6.11, se puede ver que en la mitad de tiempo que las anteriores CSA's, conseguimos transferir un volumen de datos mayor, satisfaciendo los objetivos deseados para este reto.

En la figura 6.12, vemos como durante éste reto el IFCA sirvió apróximadamente 3 TB de datos en 3 días a FNAL y el CERN.

La calidad de las tranferencias fue muy por encima de lo esperado para un Tier-2, salvo puntuales errores con alguno de los *Tier-1* (ver figura 6.13).

Durante la CSA08, se enviaron casi 50,000 trabajos al IFCA en tan solo un mes. La calidad fue bastante buena durante la mayor parte del ejercicio, como se

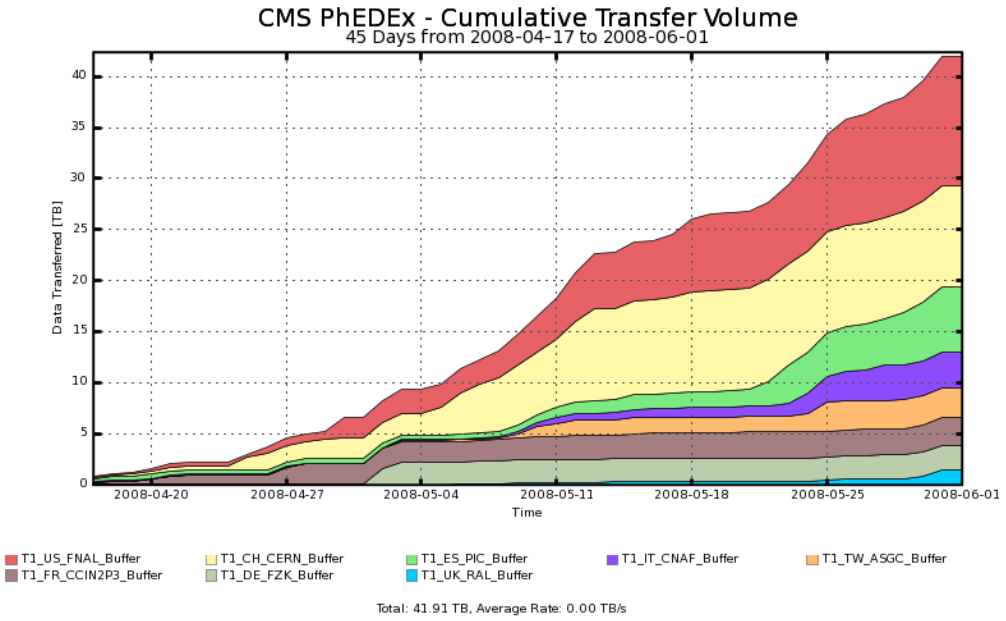


Figura 6.11: Tranferencias al IFCA desde diferentes *Tier-1* durante el evento CSA08

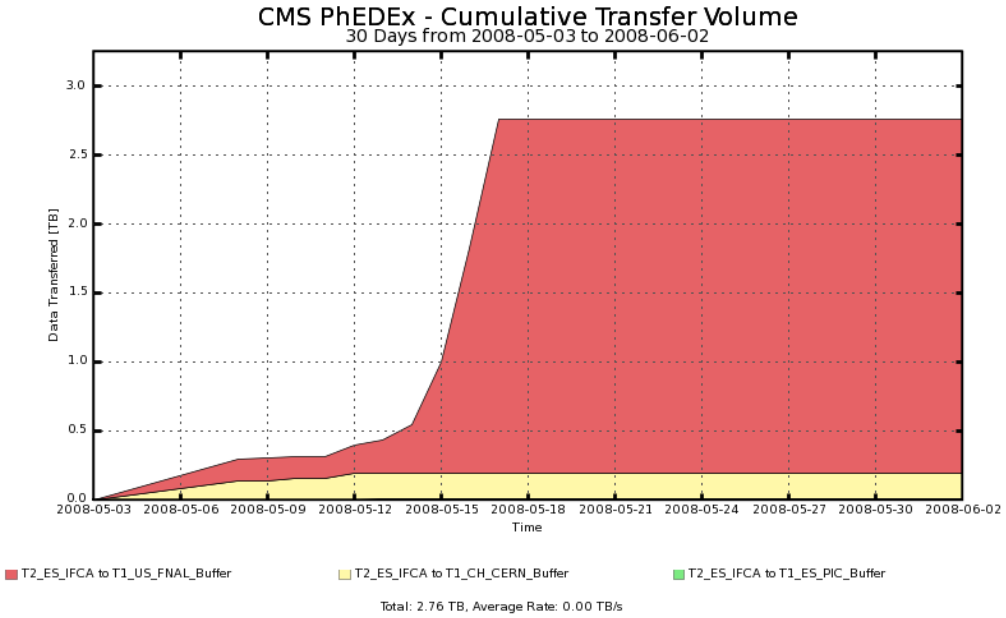


Figura 6.12: Tranferencias desde el IFCA a diferentes *Tier-1* durante el evento CSA08

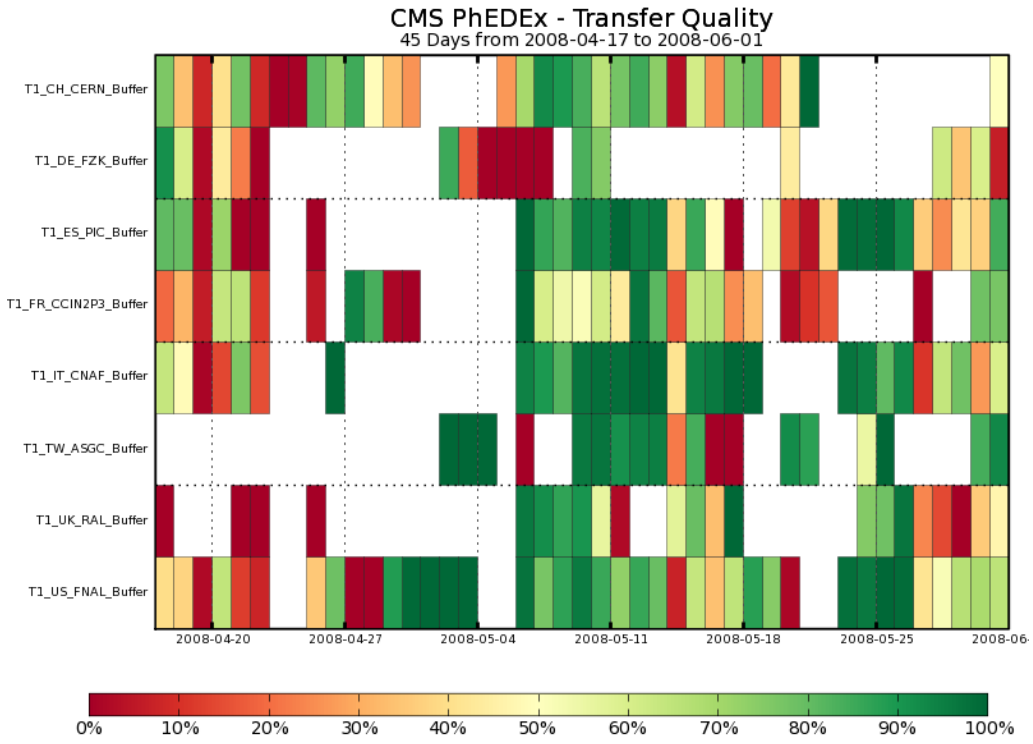


Figura 6.13: Calidad de las tranferencias en el IFCA desde diferentes *Tier-1* durante el evento CSA08

puede comprobar en la gráfica 6.14. Los mayores problemas se concentraron en los últimos días de mayo (entre el 25 y el 29) donde se realizó una prueba de carga lanzando mas de 8,000 trabajos en un solo día contra nuestro centro. Ésto provocó problemas de gestión de los mismos por parte del *Computing Element*, teniendo que actualizar esta máquina a una más potente. El sistema de almacenamiento y transferencia de datos funcionó de manera eficiente durante todo el reto.

Este reto fue particularmente exigente en terminos de esfuerzo por parte de los operadores de las distintas infraestructuras computacionales, y en particular para los encargados de mantener y gestionar los sistemas de almacenamiento.

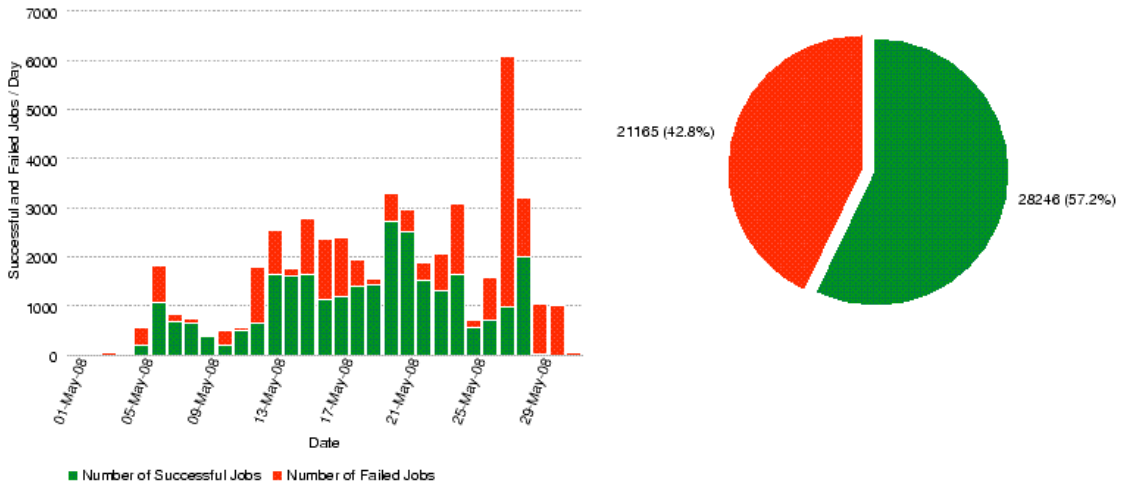


Figura 6.14: Número de trabajos ejecutados con y sin éxito en el *Tier-2* del IFCA durante la CSA08

6.3. Debug Data Transfers (DDTs)

A diferencia de los eventos CSA, los eventos *DDT*[22] tienen como finalidad comprobar y testear el sistema de tranferencias de datos dentro de CMS y de éste modo proporcionar un entrenamiento a los operadores de los diferentes *Tiers* para que estén preparados en el momento de la recepción de los datos reales.

Al principio de estos eventos se marcan una serie de objetivos para cada *Tier*, en función del nivel en que se encuentre, por tener grados de compromiso diferentes con el experimento.

A principios de agosto del 2007 se propuso que para que existiese un enlace activo entre dos centros en la instancia de producción (Prod), antes debe de pasar una serie de pruebas de calidad. Ésta serie de pruebas se llevan a cabo en la instancia *Debug*, donde se inyectan datos ficticios para realizar las transferencias entre centros. De esta forma se evitan interferencias entre los datos reales y los de prueba.

Estas pruebas surgieron a partir de la constatación durante la CSA06 de que algunos *Tiers* no cumplían una serie de métricas de red, o una estabilidad en su *hardware* mínima, para el buen desarrollo de las actividades propuestas. De este modo si un centro no supera los criterios establecidos, su canal FTS con el centro con el que no ha pasado el test, es desabilitado por los servicios centrales. Así se evita interferencias en el canal de datos reales. Para que en el canal vuelva a ser activado en la instancia Prod, ha de volver a pasar los test en el canal de prueba, Debug.

Como hemos mencionado en puntos anteriores, según las recomendaciones de

CMS, cada *Tier-2* debe de tener comisionados todos sus enlaces con los *Tier-1*, 7 más uno especial con el CERN en descarga, y al menos 2 en subida, uno de ellos con el *Tier-1* asociado.

Según la versión actual, cada enlace será clasificado como:

1. **NOT-TESTED:** Nunca probado.
2. **PENDING-COMMISSIONING:** Probando pero con problemas.
3. **PENDING-RATE:** Tasa de transferencia y/o estabilidad aún no han pasado las métricas.
4. **COMMISSIONED:** En producción.
5. **COMMISSIONED-SUSPENDED:** Aún en producción, pero no se tiene en cuenta para las medidas.
6. **PROBLEM-RATE:** Funcionando pero con problemas de transferencia en el enlace de producción.
7. **PROBLEM-QUALITY:** Funcionando, pero con calidad mala o peor en el enlace en producción.

6.3.1. DDT1

El reto DDT1 comenzó a principios de Agosto del 2007, como preparación para la CSA07 (ver 6.2.2), que comenzaría en octubre del mismo año. La principal utilidad fue depurar los enlaces entre los diferentes *Tier-1* $\rightarrow$ *Tier-2* y viceversa. La métrica que un enlace debería cumplir para ser comisionado era la siguiente:

1. Transferencias de más de 300GB al día durante un periodo de 7 días consecutivos.
2. Al menos un día de los siete con más de 500GB transferidos.

Se perdería esta condición si durante siete días seguidos, las transferencias estuvieran por debajo de los 300GB/día. El IFCA fue el segundo centro de entre los más de 25 *Tier-2*, que había en aquel momento, en conseguir los objetivos marcados por CMS. Es decir, conseguir comisionar los 7 enlaces de descarga con los 7 *Tier-1* más el CERN y el de subida con su *Tier-1* asociado y otro de subida a modo de *backup*.

Una vez que el enlace adquiere el estatus de comisionado, es periódicamente comprobado, pudiendo ser decomisionado. Esto dio lugar a una excesiva carga de trabajo para los operadores de los diferentes *Tiers* ya que, requería una constante supervisión de todo el sistema. Además de almacenar y transferir, había que crear diferentes paquetes de datos de prueba entre cada centro, lo cual era tedioso.

En la figura de tranferencia de datos durante la DDT1 (ver figura 6.15) y en la de calidad (ver figura 6.16), puede verse como el IFCA tuvo algunos problemas durante el periodo 11 al 18 de septiembre, debido a la perdida de discos en varios servidores DPM, teniendo bastantes problemas a la hora de estabilizar el sistema de nuevo.

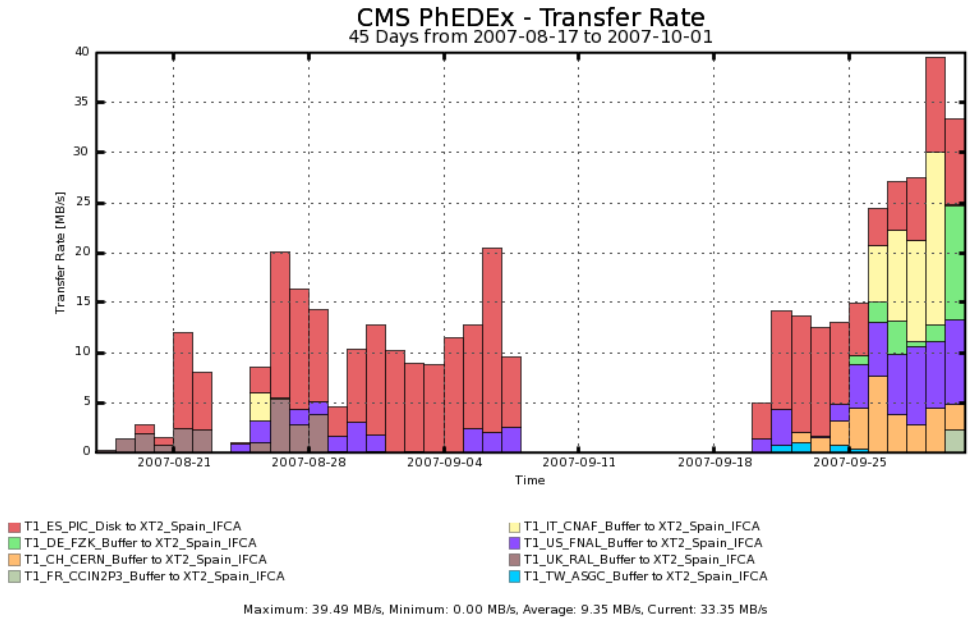


Figura 6.15: Tranferencias al IFCA desde diferentes *Tier-1* durante el evento DDT1

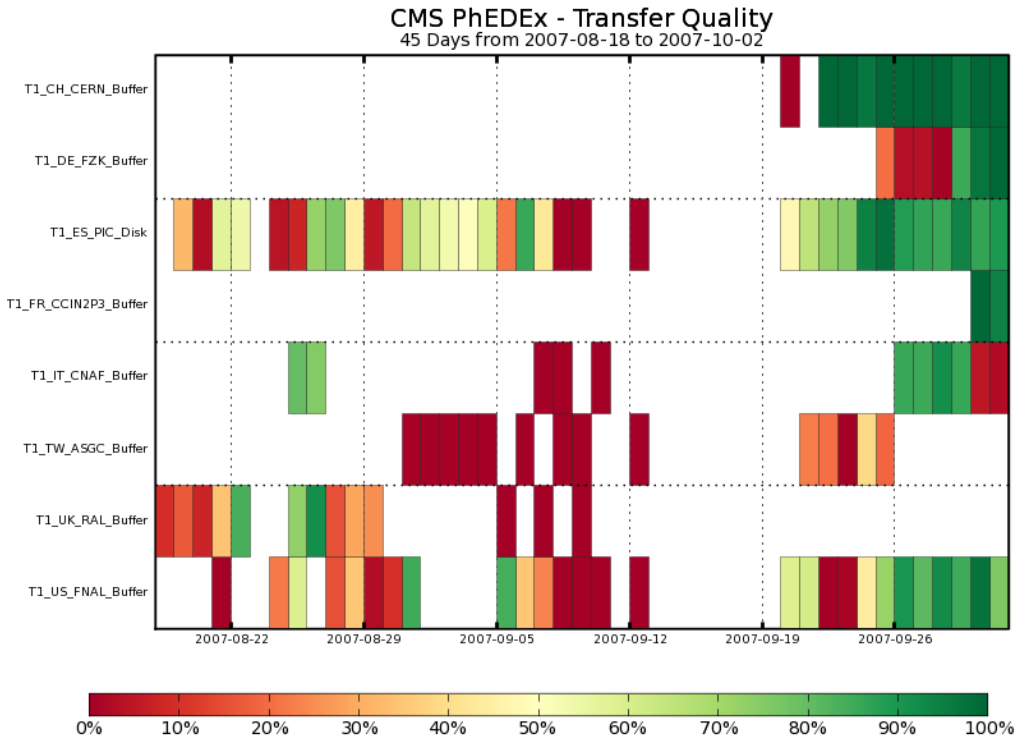


Figura 6.16: Calidad de las transferencias hacia el IFCA desde diferentes *Tier-1* durante el evento DDT1

6.3.2. DDT2

El reto DDT2 comenzó a mediados de Febrero del 2008, como preparación final para el asalto a los datos reales que se esperaba que comenzarían a fluir a finales del mismo año. Debido a la excesiva carga de trabajo que hubo en el evento anterior (DDT1), esta vez se centralizó la mayor parte del trabajo (creación de los paquetes de datos de prueba, adecuar las velocidades de descarga de los mismos para los diferentes enlaces, etc), dejando a los operadores la supervisión del sistema. La métrica para entrar o mantener un enlace comisionado fue la siguiente:

- Mantener una descarga continuada con una tasa de $20MB/s$ durante almenos 12h o en su defecto descargar en 12h 875GB, para los enlaces *Tier-1* $\rightarrow$ *Tier-2*.
- Mantener una tasa de transferencia continuada de $5MB/s$ durante 12h consecutivas para los enlaces *Tier-2* $\rightarrow$ *Tier-1*.

A diferencia de la DDT1, en la DDT2 los enlaces eran ejercitados en períodos concretos durante los cuales se verificaba si eran capaces de cumplir la métrica. Una vez iniciada la prueba en un enlace determinado, este dispondría de tres días para suerar los requisitos marcados.

El IFCA superó las métricas establecidas para este reto sin problema, tanto en el canal de descarga (ver figura 6.17), como en el de subida (ver figura 6.18).

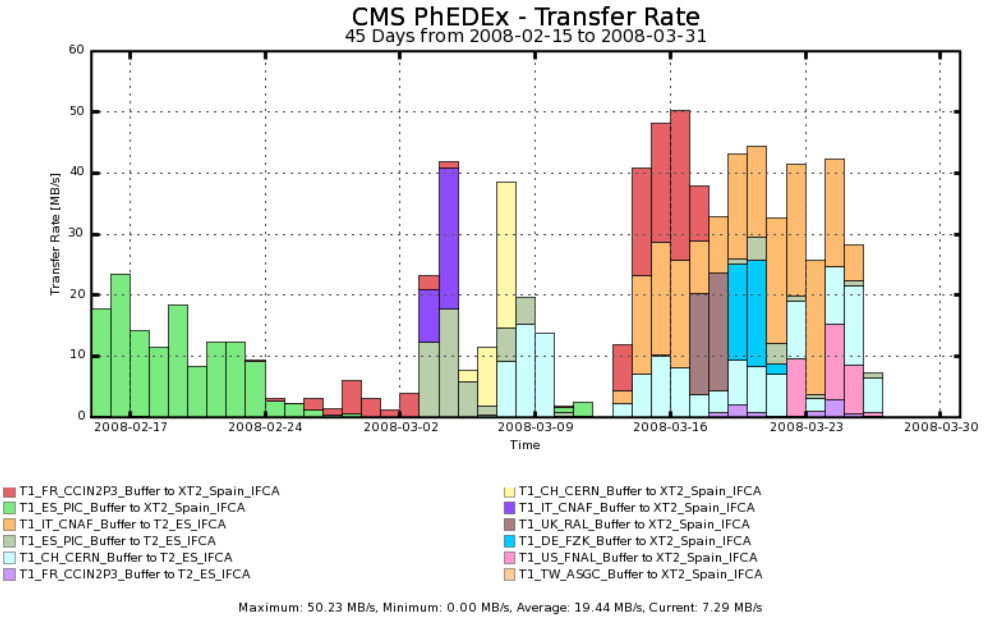


Figura 6.17: Transferencias hacia el IFCA desde diferentes *Tier-1* durante el evento DDT2

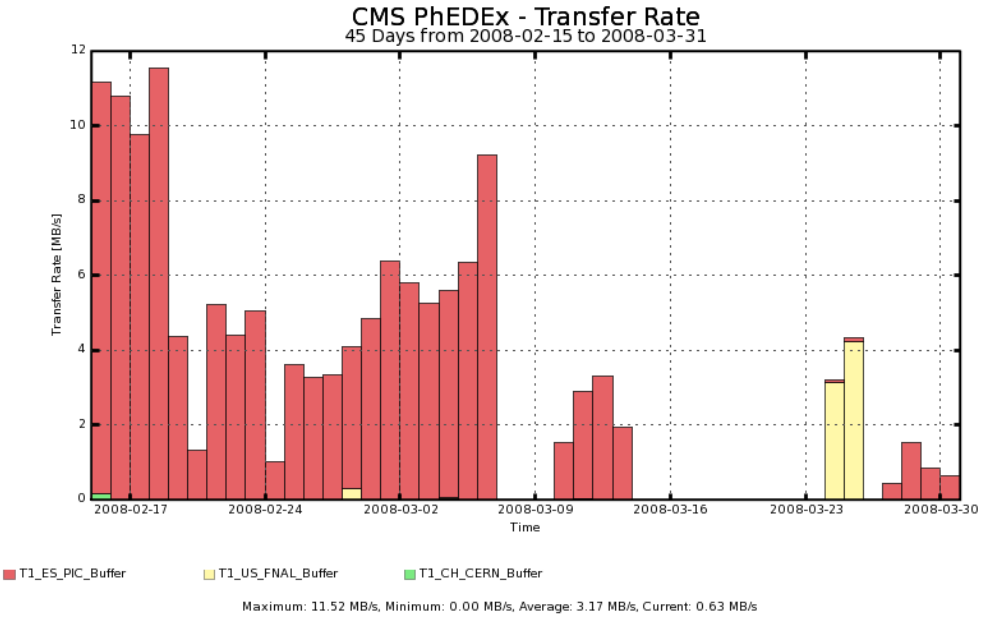


Figura 6.18: Tranferencias desde el IFCA a diferentes *Tier-1* durante el evento DDT2

Aunque hubo algunos incidentes durante este periodo, coincidiendo con la migración del *Storage Element* DPM (ver sección 5.1.1) al *Storage Element* StoRM (ver sección 5.1.2) como se puede observar en las gráfica de calidad (ver figura 6.19)

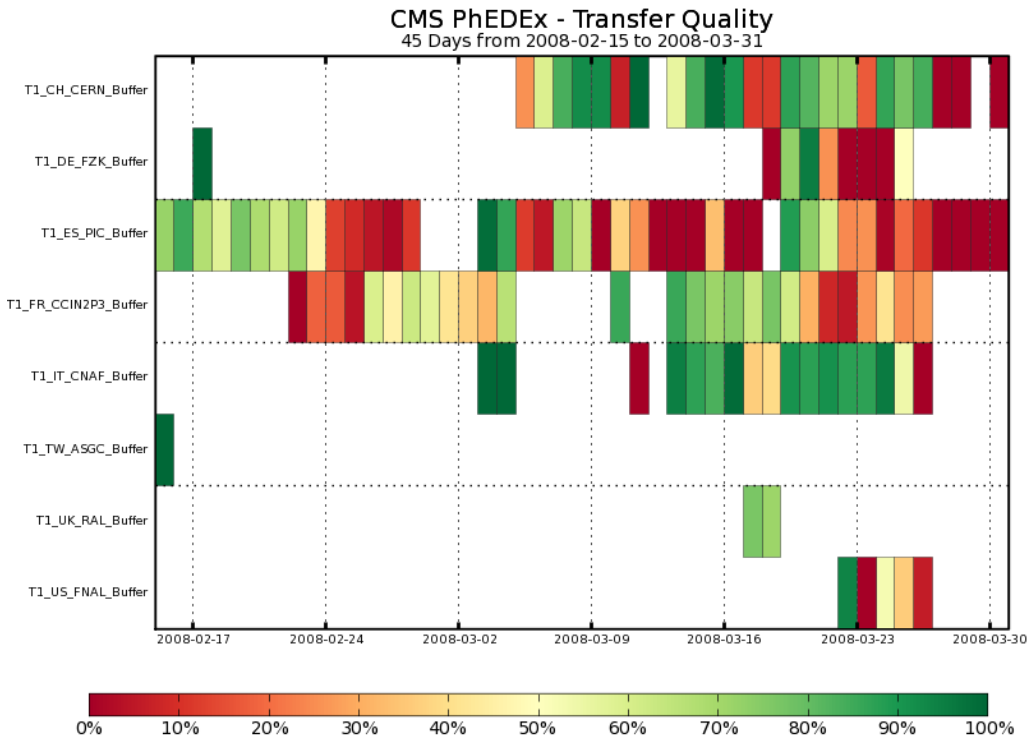


Figura 6.19: Calidad de las tranferencias en el IFCA desde diferentes *Tier-1* durante el evento DDT2

Capítulo 7

Conclusiones

Se ha presentado el proceso de instalación y configuración en el centro de computación del Instituto de Física de Cantabria de dos soluciones de almacenamiento, transferencia y análisis de grandes volúmenes de datos. Ambas soluciones se han integrado con el modelo computacional del detector de partículas CMS que opera en el colisionador de protones LHC.

Una primera solución, basada en la agregación de servidores de disco heterogéneos y DPM como sistema de ficheros y *Storage Element*, fue especialmente útil en las primeras fases de la implementación del modelo computacional de CMS, permitiendo la participación del IFCA en la producción de simulación de Montecarlo y el análisis de esos datos, pero demostró una importante falta de escalabilidad y estabilidad.

La segunda solución, instalada a partir de 2008, basada en *hardware* y *software* IBM, con GPFS como sistema de ficheros y StoRM como *Storage Element* e interfaz SRM, ha demostrado tras una intensa labor de configuración y optimización, una escalabilidad, eficiencia y estabilidad mucho mayores que el sistema anterior, acorde con los requisitos de manejo y acceso a datos necesarios en los experimentos del LHC.

Así mismo se ha descrito la labor de instalación y optimización de algunas herramientas desarrolladas en el contexto del proyecto LCG o específicamente por CMS, para el tratamiento de los datos como PhEDEx, así como de los parámetros específicos de GPFS, a partir de la experiencia e información obtenidas en los diferentes retos de transferencia y análisis de datos en los que el IFCA, como parte del Tier-2 español para CMS, ha participado.

El resultado final es un sistema estable, escalable y altamente eficiente de almacenamiento y transferencia de datos que permite a la comunidad de físicos de la colaboración CMS y en particular a los investigadores del IFCA, participar

en condiciones óptimas en la explotación científica de los resultados de uno de los experimentos más complejos y relevantes en física de partículas de este siglo.

Se ha participado con éxito en todo el trabajo descrito, en un largo proceso que va desde el estudio de opciones y la selección del *hardware* y *software* más adecuado, hasta su despliegue final, pasando por la instalación del *hardware* finalmente adquirido, el cableado eléctrico y de red, la actualización del *firmware* de los distintos elementos, la instalación del *software* especializado, la optimización y monitorización del sitema, la detección y corrección de problemas, y la gestión de modificaciones siguiendo la evolución del *software*, *hardware* y el modelo computacional de CMS.

Bibliografía

[1] The Cern Web Page. *"http://www.cern.ch",* consultado en Septiembre 2008

[2] CMS Collaboration. *"The Compact Muon Solenoid Technical Proposal",* CERN/LHCC 94-38 (1994). LHCC/P1.

[3] CMS Collaboration. *"The Computing Project Technical Design Report",* CERN/LHCC 2005-23 (2005).

[4] I. Foster. *"What is the Grid: A Three Point Cheklist",* Grid Today, 20 Julio, 2002
I. Foster and C. Keeselman. *"The Grid: Blueprint for a New Computing Infraestructure",* Morgan Kaufmann Publishers, 1999.

[5] The Globus Project. *"http:www.globus.org",* consultado en Septiembre 2008.

[6] The Grid Forum. *"http://www.gridforum.com",* consultado en Septiembre 2008.

[7] Enabled Grids For e-Science (EGEE). *"http://www.eu-egee.org",* consultado en Septiembre 2008.

[8] The LHC Project. *"http://www.lcg.web.cern.ch",* consultado en Septiembre 2008.

[9] D. Rodriguez. *"Data Management and Data Mining in High Energy Physics in the CrossGrid Project",* Tesis de Doctorado, Universidad de Cantabria, 2007.

[10] The www page for MONARC Project. *"http://www.cern.ch/MONARC",* consultado en Septiembre 2008.

[11] The Tier-1 Computing Model. *"https://twiki.cern.ch/twiki/bin/view/CMS/CmsTier1ComputingModel",* consultado en Septiembre 2008.

[12] S. Agostinelli et al. *"GEANT4: A simulation toolkit",* Nucl. Instrum. and Methods A506 (2003) 250-303.
The GEANT 4 Web. *"http://geant4.cern.ch",* consultado en Septiembre 2008.

[13] GANGLIA. *"http://monitor.ifca.es,"* consultado en Septiembre 2008.

[14] The GPFS web. *"http://www.ibm.com/systems/clusters/software/gpfs.html",* consultado en Septiembre 2008.

[15] The Wikipedia Raid web. *"http://es.wikipedia.org/wiki/RAID",* consultado en Septiembre 2008.

[16] IBM. *"General Parallel File System: Concepts, Planning, and Installation Guide",* GA76-0413-01.

[17] M. Bowden, M. Crawford, W. Wu. *"dCache Network Tuning",* 2006 Nov 02.

[18] The DBS web. *"https://cmsweb.cern.ch/dbs_discovery",* consultado en Septiembre 2008.

[19] CMS collaboration. *"CMS Computing, Software and Analysis Challenge in 2006 (CSA06) Summary",* CERN/LHCC2007-010 (2007).
I. Cabrillo et al. *"CSA06 Computing, Software and Analysis Challenge at the Spanish Tier-1 and Tier-2 sites",* CMS NOTE 2007/022.

[20] The CSA07 web. *"https://twiki.cern.ch/twiki/bin/view/CMS/CSA07",* consultado en Septiembre 2008.

[21] CMS collaboration. *"The 2008 CMS Computing Software and Analysis Challenge",* en preparación.

[22] The DDT Program web. *"https://twiki.cern.ch/twiki/bin/view/CMS/DDT",* consultado en Septiembre 2008.

Glosario

Andrew File System (AFS)

Sistema de archivos distribuido a través de la red que fue desarrollado como parte del proyecto Andrew por parte de la Universidad Carnegie Mellon.

CERN Advance Storage Manager (CASTOR)

Sistema de almacenamiento desarrollado por el CERN orientado al uso tanto de disco como de cinta.

CERN

El CERN, situado en Ginebra, es el acronimo de Centre Européenne pour la Reserche Nucléaire (originalmente Conseil Europeen pour la Recherche Nucleaire).

Centro de Supercomputación de Galicia (CESGA)

Centro de cálculo de altas prestaciones de Galicia en España. Entre sus instalaciones cuenta con el supercomputador Finisterrae la segunda máquina más potente de España tras el superordenador Marenostrum (2008).

Centro de Invest. Energéticas Medioamb. y Tecnológicas (CIEMAT)

Organismo público de investigación y desarrollo tecnológico español que depende del Ministerio de Educación y Ciencia.

Compact Muon Solenoid (CMS)

Detector de partículas de propósito general dentro del LHC (ver sección 2.3).

Computing, Software and Analysis Challenge (CSA)

Eventos realizados por CMS para el testeo de toda la cadena de software y hardware dentro del experimento.

dCache

Sistema de almacenamiendo desarrollado por el DESY y FNAL diseñado para su implementacion en sistemas basados en disco sobre servidores heterogeneos, implementa la interface SRM.

Debug Data Transfer (DDT)

Eventos realizados dentro de CMS orientados a verificar la calidad de los enlaces entre los diferentes Tiers.

Domain Name Service (DNS)

Base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet

Disk Pool Manager (DPM)

Sistema de almacenamiento desarrollado por el CERN orientado a servidores de disco y centros pequeños.

Enabling Grids for E-sciencE (EGEE)

Organización Europea que se encarga del desarrollo e incorporacion de herramientas a la infraestrucctura Grid.

EXT3

Es un sistema de archivos con registro por diario. Es el sistema de archivo más usado en distribuciones Linux.

Fibre Channel (FC)

Tecnología de red utilizada principalmente para redes de almacenamiento disponible primero a la velocidad de 1 Gb/s y posteriormente a 2 4 y 8 Gb/s.

Fermi National Accelerator Laboratory (FNAL)

Laboratorio de física de altas energías llamado así en honor al físico Enrico Fermi; se encuentra localizado 50 kilómetros al oeste de Chicago. En el Fermilab está instalado el segundo acelerador de partículas más potente del mundo (el primero es el LHC) el Tevatrón. Usado para descubrir el top quark.

File Transfer Protocol (FTP)

Protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente servidor.

General Parallel File System (GPFS)

Disco compartido de alto rendimiento (sistema de ficheros para clusters) desarrollado por IBM.

GRID

Paradigma de la computación distribuida en el que todos los recursos de un número indeterminado de computadoras geográficamente distribuidas que son tratadas de forma global come un unico supercomputador de una manera transparente para el usuario.

Host Bus Adapter (HBA)

Dispositivo que conecta un host a una Red de datos o un Sistema de almacenamiento.

Hewlett Packard (HP)

La mayor empresa de tecnologías de la información del mundo (año 2008). Fabrica y comercializa hardware y software además de brindar servicios de asistencia relacionados con la informática.

Instituto de Física de Cantabria (IFCA)

Instituto mixto del Consejo Superior de Investigaciones Científicas (CSIC) y la Universidad de Cantabria (UC) dedicado a la estructura de la materia y la astrofísica.

Instituto de Física Corpuscular (IFIC)

Instituto mixto del Consejo Superior de Investigaciones Científicas (CSIC) y la Universidad de Valencia (UV) dedicado a la física de partículas y nuclear experimental y teórica.

Internet SCSI (iSCSI)

Es un estándar oficial ratificado el 11 de Febrero de 2003 por la Internet Engineering Task Force que permite el uso del protocolo SCSI sobre redes TCP/IP.

Large Hadron Collider (LHC)

Nuevo acelerador de partículas construido en la frontera entre Francia y Suiza en el CERN para acer colisionar protones e iones pesados (ver sección 2.2).

Logical Unit Number (LUN)

En almacenamiento un LUN es una dirección para una unidad de disco duro y por extensión el disco en sí mismo.

Linux Virtual Server (LVS)

Solución para gestionar el balanceo de carga en sistemas Linux. Es un proyecto de código abierto iniciado por Wensong Zhang en mayo de 1998. El objetivo es desarrollar un servidor Linux de alto rendimiento que proporcione buena escalabilidad confiabilidad y robustez usando tecnología clustering.

Network File System (NFS)

Es un protocolo de nivel de aplicación según el Modelo OSI. Es utilizado para sistemas de archivos distribuidos en un entorno de red de computadoras de área local.

National Research and Education Network (NREN)

Proveedor especial de Internet dedicado a dar soporte a comunidades de investigación de diferentes países.

Network Share Disk (NSD)

Es un dispositivo o pieza de información en una computadora que puede ser accedido remotamente desde otra computadora.

Open Science Grid (OSG)

Infraestucctura de Computación GRID orientada a proyectos científicos de gran escala construída y gestionada por un consorcio de Universidades y laboratorios de EEUU.

Port d'Informació Científica (PIC)

Centro que da soporte a comunidades científicas que trabajan en proyectos que necesitan sofisticados recursos de procesamiento y almacenamiento para la computación distribuida.

Redundant Array of Inexpensive Disks (RAID)

Sistema de almacenamiento que usa múltiples discos duros entre los que distribuyen o replican los datos.

Storage Area Network (SAN)

Es una red concebida para conectar servidores matrices (arrays) de discos y librerías de respaldo. Está basada principalmente en tecnología fibre channel y más recientemente en iSCSI. Su función es la de conectar de manera rápida segura y confiable los distintos elementos que la conforman.

Serial Advanced Technology Attachment (SATA)

Interfaz de transferencia de datos entre la placa base y algunos dispositivos de almacenamiento como puede ser el disco duro u otros dispositivos.

Small Systems Computer Interface (SCSI)

Es un interfaz estándar para la transferencia de datos entre distintos dispositivos del bus de la computadora.

Shecure Shell (SSH)

Intérprete de comandos seguro. Es el nombre que recibe un protocolo y del programa que lo implementa y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X (en sistemas Unix) corriendo.

StoRM

Sistema de almacenamiendo desarrollado por el INFN y CNAF diseñado para su implementacion en sistemas basados en disco. Implementa la interface SRM y esta expecialmente diseñado para su uso con GPFS como GFS (ver sección 5.1.2).

Zettabyte File System (ZFS)

Sistema de ficheros desarrollado por Sun Microsystems para su sistema operativo Solaris.