An update on perfmon and the struggle to get into the Linux kernel

Andrzej Nowak, CERN openlab

<Andrzej.Nowak@cern.ch>

Abstract. At CHEP2007 we reported on the perfmon2 subsystem as a tool for interfacing to the PMUs (Performance Monitoring Units) which are found in the hardware of all modern processors (from AMD, Intel, SUN, IBM, MIPS, etc.). The intent was always to get the subsystem into the Linux kernel by default. This paper reports on how progress was made (after long discussions) and will also show the latest additions to the subsystems.

1. Introduction and goals

1.1. Introduction

CERN openlab is a hi-tech computing research collaboration, working closely with industrial partners on behalf of CERN. It is therefore not surprising that the quest for performance and performance assessment is of the highest importance.

In order to do well in this job, one needs not only advanced computing knowledge, but also the proper tools. This document outlines the work done by openlab on such a tool – a performance monitoring framework for Linux called perfmon.

1.2. Goals

The ultimate goal of the perfmon project at CERN is to provide CERN with a unified performance monitoring framework, which is lightweight and robust at the same time, and could be used in the background, without affecting the performance of any applications running on the machine. A side effect is the provisioning of CERN openlab with a framework which could be used to monitor both software and hardware performance on multiple levels of abstraction.

This can be achieved by optimizing perfmon in order to work reliably and efficiently with High Energy Physics (HEP) software and frameworks (especially with multiple execution splits and dynamically loaded libraries), and a significant part of that work has been already successfully carried out [Jur03][Now03].

2. Perfmon history and brief description

Perfmon was originally started at HP Labs, who was a CERN openlab partner during openlab I and II, as a performance monitoring project for the Itanium Processor Family. At that time openlab conducted a review of available performance monitoring tools and their capabilities, in order to choose the most appropriate ones for daily use [Jur01]. Initially perfmon and perfctr were considered, however the future of perfctr was not clear in certain areas. Perfmon was a natural fit, as many top of the line

platforms, including Itanium, are evaluated here. Times changed, and x86 rose to fame, especially with its x86_64 variant, which gained a lot of ground in the server and data center area. Many features present up until then exclusively in the Itanium Processor Family moved over to the x86 side. Thus, perfmon was refitted to work on many more architectures, including x86_64, x86, MIPS, Itanium, Sparc, Power and Cell. The general concept of the architecture is presented on the figure below.

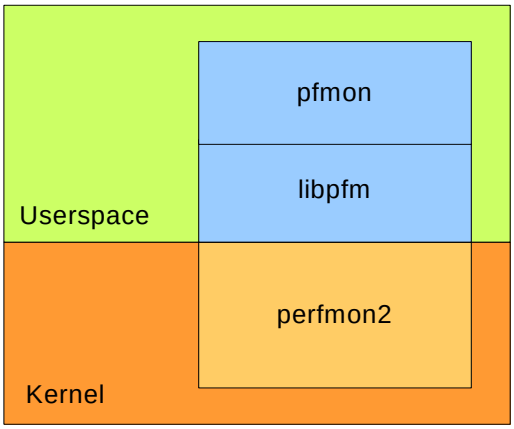

Figure 1: perfmon architecture

The perfmon2 framework consists of a part which resides in the Linux kernel, a "glue" library called "*libpfm*" and a client tool called "*pfmon*". The individual components of the framework can be characterized as follows:

- *Kernel part* – orchestrates the monitored process(es), responsible for hardware access; handles interrupts and context switches,
- *Libpfm* – communicates with the kernel, renders the results into an intermediate format which is then presented to pfmon,
- *Pfmon* – originally just a demo application for the framework, it grew to be a stand alone performance monitoring tool and integral part of perfmon2; renders performance data to be shown to the user, takes care of some parts of process orchestration and manages debug breakpoints.

3. CERN's initial involvement and motivation

Perfmon initially attracted CERN's interest because of its lightweight nature combined with the excellent Itanium PMU. As x86 CPUs gained an expanded set of performance monitoring counters and as perfmon moved to the x86 domain, it became clear that it might be a generally useful tool. Also encouraging was the vague perspective of the inclusion of the one-for-all platforms approach in the Linux kernel. The turning point was the Gelato ICE 2006 conference [Era01][Jur02]. Given the state of pfmon at the time, it was decided that the aspects which were of the highest importance for CERN were accurate address to symbol translation in profiles and significantly improved reliability, which would enable the analysis of complex High Energy Physics software frameworks. Interestingly enough, the "todo" list for pfmon was quite close to CERN's wishlist for the tool.

CERN already had a proof of concept implementation of the proposed changes, and it was well received. In order to satisfy the mentioned requirements, an initial patch for pfmon was created. It added some, if not most, of the desired functionality, and was transmitted to the author at HP Labs. Very good results were reported with many of the huge frameworks used at CERN. However, the patch significantly modified the source code of pfmon, which prompted a fear that the changes might have a potentially negative impact on reliability. A special session was held between HP and CERN, in order to determine the strategy for the several months which would follow. It was decided that

memory optimizations and some other improvements were required. Both HP and CERN worked intensively over late Spring 2007 in order to make this happen, and numerous optimization and bugfix patches developed at CERN were merged with the mainline version of the tool. Also, the developed improvements influenced the future shape of pfmon. At the same time, a summer student was chosen to work on the project. His task was to enhance dynamic library handling and finalize certain pieces of code responsible for result consolidation and printing. A number of perfmon test systems were set up, which were used by groups and people collaborating with openlab. During the same period a new effort has started, the objective of which was to familiarize the HEP community with performance monitoring tools [Jur03].

4. Continued efforts

In mid 2007, the project was stalled for several months. Several factors which contributed to this state were the departures of several related programmers, and a heavy workload burden imposed both on HP and openlab. After this period of low activity, perfmon efforts were picked up again in fall 2007 and started moving forward at a fast pace. At the same time, it also became apparent that HP Labs would be undergoing some significant changes in late 2007, and thus it was reasonable to make an extra effort to conclude the core perfmon work before that time.

It was decided that both the pfmon tool and the API needed a heavy overhaul, to make the software more lightweight and to ease the insertion of CERN's desired set of options, thus reducing the overall size of the patch. CERN's contribution was revised in terms of scope and functionality, and was rewritten from scratch. In mid-December 2007, after several months of very intensive efforts, CERN's codebase was merged with that of pfmon, and CERN's contributions were committed to the central perfmon CVS repository. The test systems running perfmon were replaced with newer versions of the toolset and testing continued. With time, the new CVS version of pfmon was disseminated to end users through the SourceForge page.

In early 2008, it was announced that the author of the tool would continue his work on perfmon in his new company (Google) [Era03]. This news was very welcome, especially given the fact that CERN already had some additional patches to the CVS source, both relating to x86_64 and Itanium. Also, perfmon releases stabilized, and the code was regularly published on SourceForge in step with kernel releases.

At the same time, a pilot project was started at CERN, the objective of which was to gather real time background data from hardware performance counters on machines executing Grid jobs, in order to assess the performance and utilization of batch servers. Some interesting findings were initially reported [Bal01].

Both testing and improvement efforts continued, in close collaboration with the author of the tool, to improve pfmon and perfmon even further. This involved new patches, testing on yet unreleased beta Intel systems such as the Nehalem and Atom, and general feedback related to reliability, functionality and performance. In addition, a rudimentary testing framework was created to help with validation.

Moreover, work on a graphical frontend for the tool was started, in order to facilitate performance monitoring for non-expert users [Now06]. Based on python and gtk2, the graphical interface tool allows for aided event selection, creating and managing performance monitoring projects and data analysis. However, development was halted at version 0.9 due to conflicting priorities.

In summer 2008, a summer student's task was to improve the simple testing framework already in place. New tests have been added and the framework received a facelift, but some work still remains to be done before the code can be made available to the public.

In the end, CERN has contributed the following technical items to the perfmon framework:

- Heavy functionality testing and performance testing (mostly x86_64 and Itanium)
- Several month long background testing sessions
- Address to symbol translation mechanism for pfmon plugins
- 64-bit and 32-bit compatibility

- Numerous bugfixes, performance and functionality improvements
- A simple black box testing framework focused on pfmon
- A graphical frontend, “gpfmon”

5. Current status

5.1. General activities and continuation

The general outcome of CERN’s efforts is positive, and the collaboration is ongoing. All new pfmon functionality items have been committed to CVS and are working properly, and also many bugs and performance issues were fixed. Regular feedback on both userspace and kernel space aspects of perfmon is being provided, especially relating to new architectures which are still in the beta phase. Occasionally, new patches for pfmon are being submitted.

5.2. Inclusion of perfmon in the Linux kernel

As of this writing, perfmon is still being prepared for inclusion in the Linux kernel [Era01][Era02][Era03]. The version marked for inclusion was dubbed “perfmon3”, and the following paragraph describes the background of this work, which is being conducted by the author of perfmon at Google.

The concept of perfmon has crystallized over many years of trials and errors. At present, the desired functionality of the tool is pretty much set, and the work which remains to be done is mostly technical. As stated earlier, including a piece of code in the Linux kernel is a long and painstaking process, where merit is not always at the center of discussion. Thanks to the relentless efforts of the author of perfmon, there now is a strategy, agreed upon by key kernel people, for moving the perfmon codebase into the main Linux tree. It should be noted, however, that this strategy has yet to be executed. The original patch with full functionality, called “perfmon2”, was split up into very small pieces, which are easy to review and functionally divided. What constitutes “perfmon3” is a small set of patches, implementing only rudimentary functionality (counting mode, x86 support only), and having a minor impact on the overall codebase of the Linux kernel. However, it is at the same time a foundation for all future work, and the feature set of the patch will gradually be expanded by functionality, which is present today in “perfmon2”.

Not surprisingly, the “glue” layer, libpfm, is also undergoing some changes in order to interface with the new kernel structures and calls.

Another recent development has been the development of a rival performance monitoring patch by a RedHat engineer. However, due to a certain amount of both technical and functional deficiencies, CERN openlab doesn’t expect to favor this approach. One worrying aspect of this development is the fact that the inclusion of perfmon in the kernel has been halted for the moment.

5.3. Teaching

In addition, regular workshops on computer architecture and performance optimization are being organized, and perfmon is the pivotal tool during the classes. In fact, these efforts have been noticed by the director of the Cern School of Computing, who added performance optimization related classes to the CERN School of Computing[1].

6. Future prospects

6.1. Inclusion of perfmon in the Linux kernel

This section outlines several possible future scenarios concerning perfmon in the Linux kernel.

[1] The CERN School of Computing is a well established computing course held every year since the 1970’s, and taught by computing experts and physicists collaborating with CERN.

- The scenario which would be the most advantageous for CERN would be one in which the perfmon3 code base becomes a part of the kernel, and is gradually and swiftly expanded to match perfmon2 in terms of functionality. Also, in line with earlier hopes, perfmon3 might in that case become the standard performance monitoring interface in the kernel, rendering undermaintained frameworks such perfctr and oprofile obsolete. Naturally, that would further strengthen perfmon's position and cement its place in the kernel.
- Another possible scenario is one in which perfmon3 makes it into the kernel, but without other later changes associated with more robust functionality do not. In that case, it would be up to RedHat and/or CERN to patch the kernel to achieve the required level of potency. The scenario in which perfmon3 becomes a standard performance monitoring interface in the kernel would still be probable.
- In the worst case, in which perfmon's movement towards the Linux kernel would be stalled indefinitely yet another time, CERN would be left in the current situation. In short, this would mean manual patching of the kernel and continued non-standard deployments. Should the code base proposed by RedHat be included, perfmon and libpfm might converge towards using the new facilities within the kernel.

Regardless of what the future will bring, perfmon will still be a tool heavily used at CERN for performance monitoring and optimization, being relatively simple to use but robust enough at the same time.

6.2. Background monitoring for batch servers at CERN

Moreover, other efforts at CERN continue to evaluate perfmon in terms of readiness for background monitoring. There is a specific chain of dependencies which need to be satisfied in order for perfmon to be widespread in the computing center. First of all, if one wishes to use perfmon by default on batch and centrally managed machines, the Scientific Linux distribution requires perfmon to be included in RedHat Enterprise Linux. Secondly, for perfmon to be shipped with RedHat, it needs to have a foothold in the Linux kernel. RedHat has been a long time supporter and user of perfmon, however the current situation is not clear. Yet another condition is the acceptance of the scientific community. However, since the overhead while monitoring is small and most applications go through a certification process for the default platform (SLC), it is not unreasonable to expect a smooth adoption. Once the mentioned conditions are fulfilled, it is possible that we will be able to enjoy default perfmon installations on SLC nodes, with hardware performance monitoring capabilities out of the box. That, in turn, would enable a wide variety of data recipients to conduct analyses of the running jobs, and potential integration into tools such as Lemon might even be described as a goal in itself.

6.3. Continued collaboration with the scientific community

CERN openlab is also looking forward to further dissemination of perfmon into the scientific community, by continuing computer architecture and performance optimization courses and by simplifying the tool. Although "gpfmon", the main vessel for bringing pfmon closer to the non-expert user, is currently stalled, we expect that it will get picked up at some point in time.

6.4. Other activities

Another development worth noting is one related to a different openlab partner, Intel, who was considering using perfmon as the base for its software performance monitoring tools. Again, this is an item which depends heavily on the inclusion of perfmon in the mainline Linux kernel tree and currently is just one option out of many.

Since perfmon was meant as a general-purpose performance monitoring framework from the beginning, other projects based on it, such as the PAPI toolkit, should be expected to grow and evolve.

7. Summary

Overall, the perfmon project at CERN has brought a lot of benefit to all collaborating sides. CERN openlab has gained a valuable tool used for both software and hardware performance assessment, which works well with many architectures and software configurations [Now02]. HP Labs and Google have received a lot of feedback from real life usage, as well as patches to improving the code base; also, openlab has disseminated the results of the collaboration in the form of workshops, presentations and articles engaging the computing community at large. Moreover, the scientific community now has the opportunity to extract the most intricate details of their applications' performance with a single command line [Dan01][Rom01]. In addition, CERN IT is able to monitor a select group of counters which yield high-level figures about data center utilization. And, finally, CERN openlab now has the ability to conveniently benchmark both applications and hardware.

8. Acknowledgments

The following people have contributed to the development and promotion of perfmon at CERN openlab: Gyorgy Balazs (Budapest Tech), Jorge Canizales Diaz (University of Madrid), Jose M. Dana Perez (CERN openlab), Stephane Eranian (HP Labs / Google), Andreas Hirstius (CERN openlab), Sverre Jarp (CERN openlab), Ryszard Jurga (CERN openlab), M. Shuaib Khan (NUST), Andrzej Nowak (CERN openlab), William A. Romero (Universidad de Los Andes)

References

[Bal01] G. Balazs (CERN openlab) – Performance monitoring of Production Batch Servers
[Dan01] J. M. Dana (CERN openlab), W. A. Romero (Universidad de los Andes) - Performance Monitoring of the Software Frameworks for LHC Experiments, February 2009
[Era01] S. Eranian (HP Labs) – Gelato ICE 2006 presentation
[Era02] S. Eranian (HP Labs) – Perfmon2: a flexible performance monitoring interface for Linux, OLS 2006, July 2006
[Era03] S. Eranian (Google) – Perfmon2: a standard performance monitoring interface for Linux, January 2008
[Jarp01] S. Jarp (CERN openlab) – Perfmon2, a leap forward in performance monitoring, CHEP 2007, September 2007
[Jur01] R. Jurga (CERN openlab) – Performance Monitors
[Jur02] R. Jurga (CERN openlab) – Practical experience with performance monitors, October 2006
[Jur03] R. Jurga (CERN openlab) – Performance profiling of Experiments' Geant4 Simulations, Geant 4 Technical Forum, January 2007
[Now01] A. Nowak (CERN openlab) – An update on perfmon, February 2009
[Now02] A. Nowak (CERN openlab) – Understanding Performance Tuning, Spring 2009
[Now03] A. Nowak (CERN openlab) – Perfmon activity report, October 2008
[Now04] A. Nowak (CERN openlab) – Perfmon2 overview, March 2008
[Now05] A. Nowak (CERN openlab) – Multi-threaded Geant 4 scalability, April 2009
[Now06] A. Nowak (CERN openlab) – gpfmon: http://cern.ch/andrzej.nowak/
[Rom01] W. A. Romero (Universidad de los Andes) - Performance monitoring of the software frameworks for LHC experiments, Summer 2008