
# Using OPT to Profile Applications on the IBM Cell BE Processor

March 2008

chris.january@allinea.com

# Using OPT to Profile Applications on the IBM Cell BE Processor

The Cell Architecture is the result of collaboration between IBM, Sony and Toshiba to design a high-performance and power-efficient processor that could drive applications in the fields as diverse as gaming, HDTV and supercomputing.[1]

The Cell is an innovative heterogeneous multiprocessor consisting of:

- The PPE – The Power Processing Element containing an IBM 64-bit Power Architecture™ core, the so-called Power Processing Unit (PPU).
- Eight SPEs – Specialized co-processor units each containing a Synergistic Processing Unit (SPU).

with a coherent on-chip bus for communication between the elements.

Whilst the PPE has a familiar processor, with ordinarily at least 256Mb RAM (global memory) available, each SPE has a large register set and a local store of 256Kb. Access to the global memory from the SPEs is performed using DMA via the bus, or by exchanging messages with the PPE through a "mailbox" mechanism.

## Profiling and the Cell

Standard PowerPC programs will run unmodified on a Cell system such as the IBM QS21 BladeCenter or the Sony PS3™ which can run the Fedora™, Red Hat Enterprise Linux™ or Yellow Dog™ distributions of the Linux operating system – using just the PPE. However High-Performance Computing users of the Cell will be keen to exploit the capability of the SPUs fully. Through profiling, a programmer can identify the parts of their program most suited to running on an SPU.

Profiling also allows a programmer to identify areas of their program that would benefit from optimisation, allowing them to reduce its run time. A shorter run time will help reduce queue wait times, allowing a faster generate-and-test cycle for all users, therefore reducing costs.

Although a small selection of scalar profiling tools are provided in the Cell SDK they crucially lack support for profiling HPC programs running across multiple nodes.

This white paper introduces the Allinea Optimisation and Profiling Tool (OPT), a profiler for both MPI and scalar applications with support for the Cell processor.

## Profiling with OPT

OPT offers an event based profiler for MPI calls and a statistical profiler using sampling for general procedure and function calls. The OPT sampler can also record resource usage such as disk I/O.

The OPT sampler includes support for the Cell processor. When a PPU program calls the `spe_context_run` API, the OPT profiling library will start sampling the program being run on the SPU. When the `spe_context_run` call finishes the library will resume sampling the PPU program.

## Flat Profile

You can view the data collected by the sampler in OPT's Call Graph View. The PPU/SPU profile data is stored under the Profiling/Sampler metric. Select this metric from the list then click the Update button in the bottom left of the window. A flat profile of your program will show in the Flat Profile. Functions in SPU programs are prefixed by the program name, i.e. `spu_program:function`.

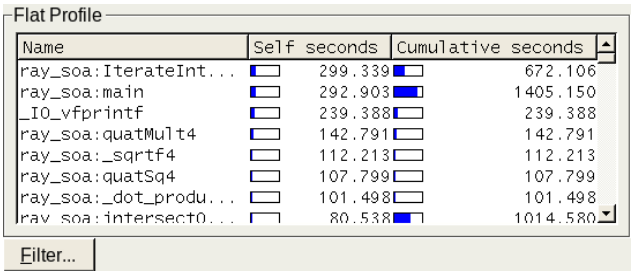


*Figure 1: Flat Profile*

OPT allows you to filter the functions shown in the Flat Profile using a regular expression. You can set

the current filter by clicking the Filter button. For example, to only view functions in `spu_program` enter:

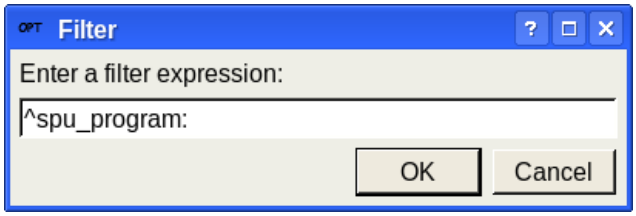


*Figure 2: Flat Profile Filter*

If you only want to view functions in the PPU program you can select the rusage/ru_utime metric. This metric records only the PPU time used by your program.

When you click on a function in the Flat Profile its combined call graph is shown in the area above. This call graph shows all the callers and callees of the function along with their relative CPU times. The values corresponding to each colour are shown in the key on the right.

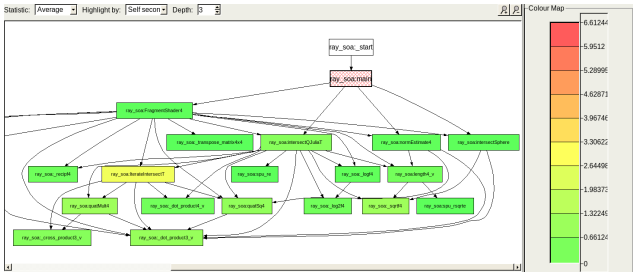

*Figure 3: Call Graph*

Unlike scalar profiling tools, OPT can collect profiling data from multiple processes running across multiple nodes. By default the Call Graph View shows the Average time spent in each function across all the processes. You can, however, also choose to view the Minimum, Maximum, Variance or Sum.

## Other Views

All of the views in OPT are designed to help you identify potential candidates for optimisation in your program. Each presents the profile data in a different way; together they allow you to see the bigger picture.

OPT is more than capable of profiling scalar programs but it also offers many features designed specifically for optimising MPI programs.

For example, the MPI Time Line allows you to see an overview of all the MPI calls made by your program and the communications between each process. You can see communication hot spots at a glance and zoom in to see more detail.

MPI function calls and communications can be charted in a number of different ways allowing you to compare processes quickly and easily.

The Message Profile view plots the overall MPI communication (time or bytes) between each process.

Finally the Metric Time View allows you to see how metrics such as disk I/O or MPI bytes in/out vary over the run time of your program.


*Figure 4: Metric Time Line*

## Summary

The IBM Cell brings extensive benefits to demanding users in terms of performance, but to exploit these fully requires a new way of programming. This paper has shown how OPT can be used to identify the areas of your programs that would most benefit from running on the SPU, as well as areas in existing PPU/SPU code that would benefit from optimisation. OPT is not only a capable scalar profiler, but can also be used to profile clusters of Cells.

OPT for Cell is available from http://www.allinea.com. Supported platforms are the IBM QS20 and QS21 Blade Centers and Sony PS3™ running Fedora Core 6 and IBM Cell SDK 2.1 or Fedora Core 7 and IBM Cell SDK 3.0.

## About Allinea Software

Allinea Software is a leading supplier of development tools for parallel and high performance computing. Allinea was founded by computer scientists from Warwick and Oxford

Universities, giving the company unrivalled expertise in scientific and parallel computing and an insight into the challenges of exploiting parallelism as it enters the mainstream.

## References

1. The Cell Project at IBM Research (http://www.research.ibm.com/cell)