Successful Technology Transfer with Siemens – The RAPID Project

Siemens is a true giant when it comes to software. Its products, systems, and solutions are built on billions of Euro invested in software R&D. A major part of its software portfolio was developed for single-core processors. In the future, however, there will be fewer and fewer single-core chips, with the consequence that both the existing software portfolio and new software will need to be prepared for use on multi-core processors. An additional problem with existing software is that it has grown over time and parts of it have not been touched for long periods. These parts work, but no one is really

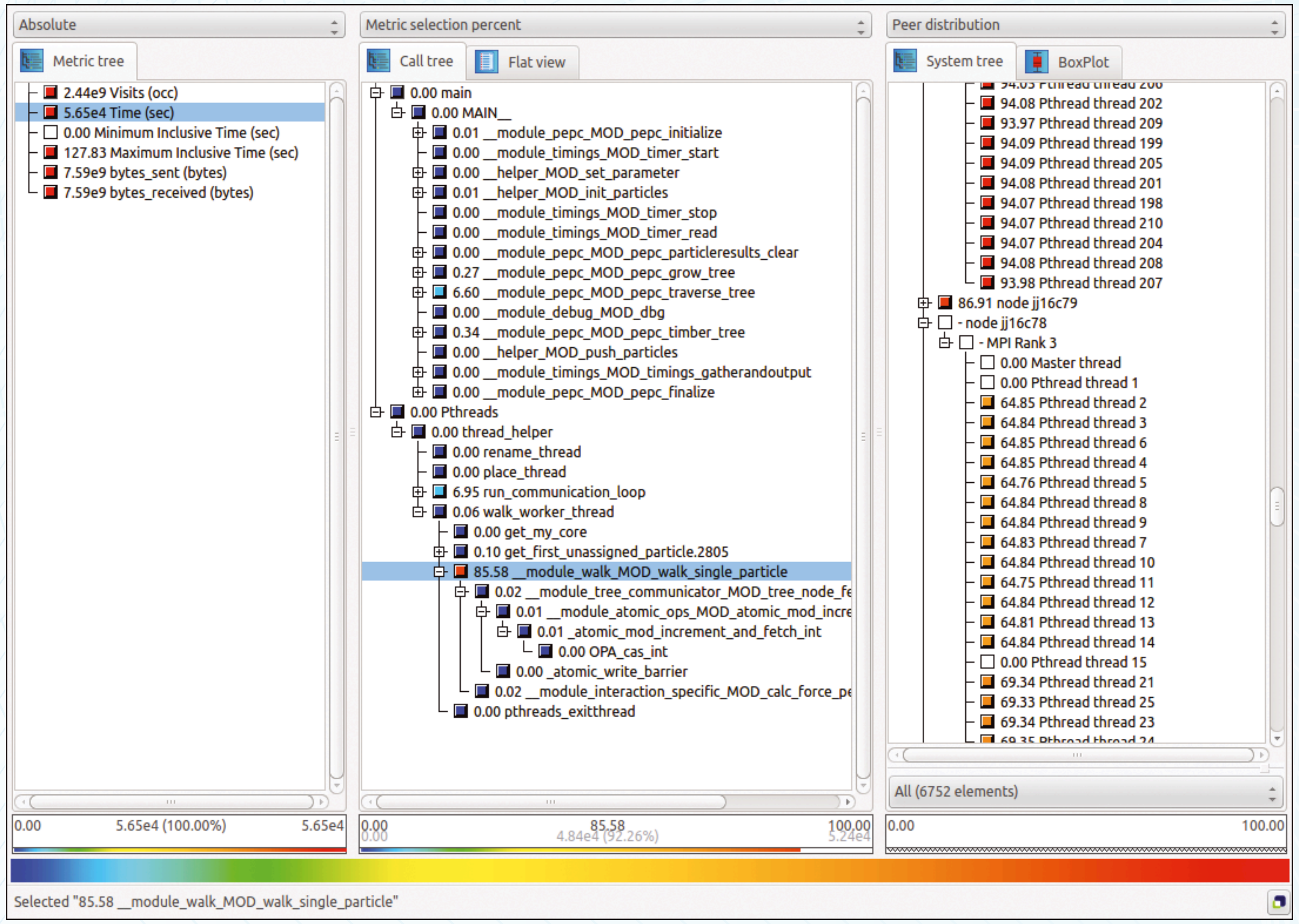

Figure 1: Profile data of a Pthreads application visualized in the Cube performance report explorer.

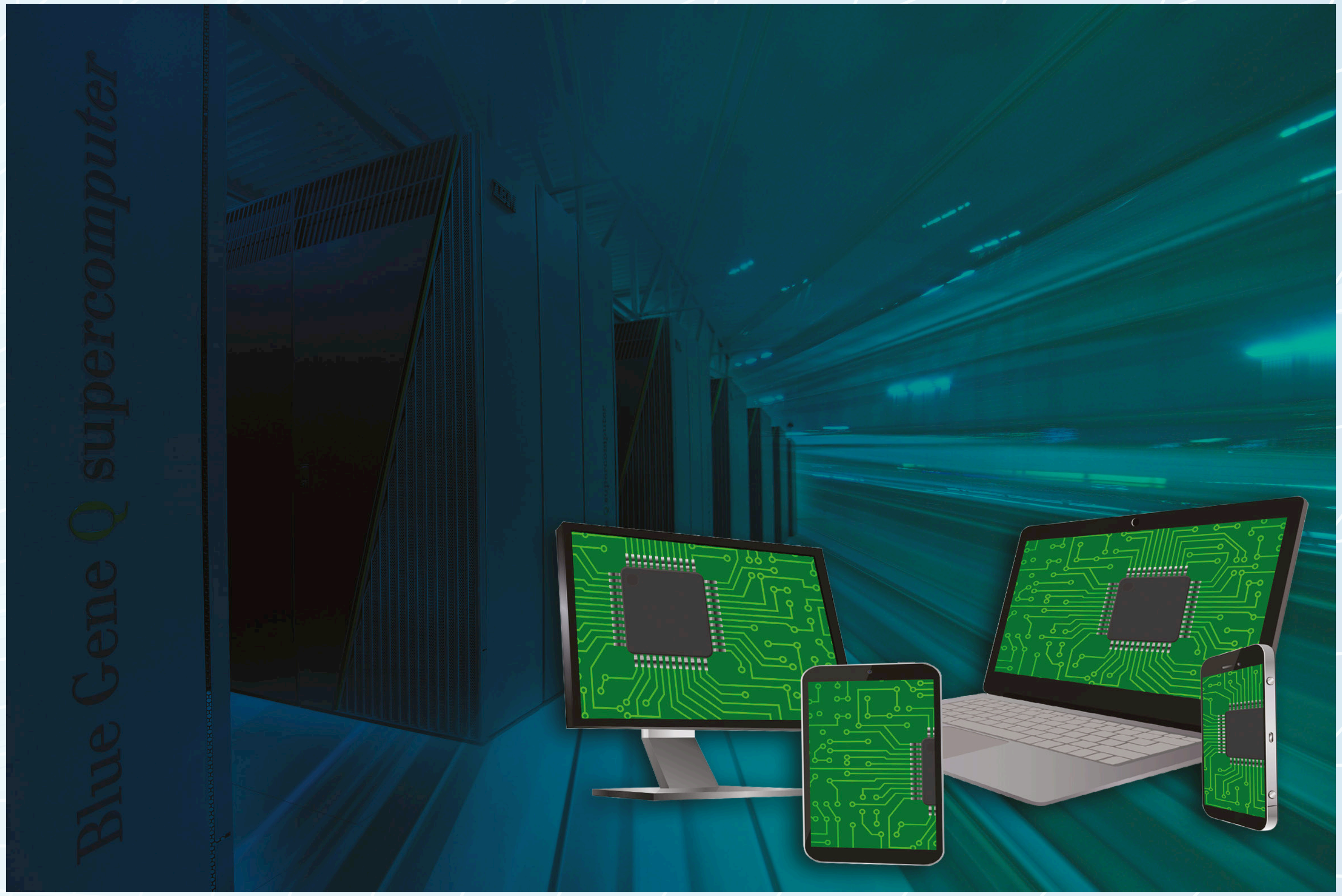

Figure 2: RAPID successfully applies HPC technology in industrial software development. © Forschungszentrum Jülich/fotolia.com.

familiar with the details anymore. In some cases, the original developers now work elsewhere or have retired.

Optimizing Software to Work on Multi-Core Processors

If subprograms that previously worked sequentially on one processor are simply distributed across parallel CPU cores on a multi-core processor, the usual problems of parallel programs emerge: race conditions when accessing shared data and deadlocks when trying to synchronize the access to this data. A particular pitfall is that these errors often do not occur at all on single-core processors and happen in a non-deterministic way in a multi-core environment. As a result, developers often build many synchronization operations into their applications. But while this eliminates race conditions, the chance for deadlocks increases. Additionally, too many synchronization operations slow down the applications – sometimes making them even slower than the single-core versions.

To remedy this situation, tools to examine the communication and synchronization aspects of the often huge software packages are needed.

In High Performance Computing (HPC), these tools already exist for a long time. Tools like the measurement and instrumentation framework Score-P [2] provides profiling and tracing data for performance analysis and visualization tools like Scalasca [3], Vampir [4], TAU [5], and Periscope [6]. However, these established tools are targeted towards the prevailing HPC programming paradigms, namely MPI and OpenMP. As these tools already exist, it seems natural to contribute to and enhance the existing tools rather than to develop new tools from scratch.

RAPID: Runtime Analysis of Parallel applications for Industrial software Development

As the Siemens software originates more from an embedded system's ecosystem than from HPC, the mentioned tools cannot be applied out-of-the-box. This is why Siemens Corporate Technology [12] and Forschungszentrum Jülich, Jülich Supercomputing Centre, collaborate in the project RAPID - Runtime Analysis of Parallel applications for Industrial software Development [1]. Goal of this project is to adapt the measurement and analysis tools Score-P and Scalasca to serve Siemens' needs. In particular, support for new threading models like POSIX threads, Windows threads, Qt threads [7], and ACE threads [8] are integrated into Score-P. In addition, support for leveraging task parallelism using MTAPI [9] is being developed. Besides supporting new programming paradigms, additional work has to be done with regards to portability. Although Score-P is already quite portable as it is running on all relevant supercomputer architectures, systems like Windows and operating systems for embedded systems have not been targeted so far. On the analysis side, new methods targeting thread-based synchronization patterns, e.g., a lock-contention analysis, are being implemented in Scalasca.

Another goal of RAPID is to assist developers in getting insight into the huge and often complex software packages by providing visual call graphs of application runs, thus overcoming the limitation of static analysis tools when it comes to the use of polymorphism or indirect calls via function pointers. To accomplish this, the established tool Cube [10] is enhanced by providing a generic plugin interface that allows various kinds of analyses on the profiling data generated during a measurement run.

The mentioned tools Score-P, Scalasca, and Cube come with the 3-clause BSD open source license [11]. All contributions to these tools developed within RAPID will be made available under the same license, which will allow the community to further improve and maintain them.

For more information please visit http://www.fz-juelich.de/ias/jsc/EN/Research/HPCTechnology/PerformanceAnalyse/RAPID/rapid.html

References

[1] http://www.fz-juelich.de/ias/jsc/EN/Research/HPCTechnology/PerformanceAnalyse/RAPID/rapid.html

[2] http://score-p.org/

[3] http://scalasca.org/

[4] http://vampir.eu/

[5] http://tau.uoregon.edu/

[6] http://www.lrr.in.tum.de/~periscop/

[7] http://qt-project.org/

[8] http://www.dre.vanderbilt.edu/~schmidt/ACE.html

[9] http://www.multicore-association.org/workgroup/mtapi.php

[10] http://www.scalasca.org/software/cube-4.x/download.html

[11] http://opensource.org/licenses/BSD-3-Clause

[12] Siemens AG, Corporate Technology http://www.siemens.com/innovation/en/

contact: Christian Rössel, c.roessel@fz-juelich.de

- Christian Rössel [1]
- Bernd Mohr [1]
- Markus Geimer [1]
- Daniel Becker [2]

[1] Jülich Supercomputing Centre (JSC)

[2] Siemens AG, Corporate Technology, Multicore Expert Center