HOPSA – A big Jump forward in HPC System and Application Monitoring

To maximize the scientific and commercial output of a High-Performance Computing system, different stakeholders pursue different strategies. While individual application developers are trying to shorten the time to solution by optimizing their codes, system administrators are tuning the configuration of the overall system to increase its throughput. Yet, the complexity of today's machines with their strong interrelationship between application and system performance demands for an integration of application and system programming.

The HOPSA project (HOlistic Performance System Analysis) therefore set out for the first time in the HPC context for combined application and system tuning developing an integrated diagnostic infrastructure. Using more powerful diagnostic tools, application developers and system administrators can easier identify the root causes of their respective bottlenecks. With the HOPSA infrastructure, it is more effective to optimize codes running on HPC systems. More efficient codes mean either getting results faster or being able to get higher quality or more results in the same time.

The work in HOPSA was carried out by two coordinated projects funded by the EU under call FP7-ICT-2011-EU-Russia and the Russian Ministry of Education and Science. Its objective was the new innovative integration of application tuning with overall system diagnosis and tuning to maximize the scientific output of our HPC infrastructures. While the Russian consortium focused on the system aspect, the EU consortium focused on the application aspect.

The HOPSA Performance Tool Workflow

One of the main results of the project was the specification and documentation of the indented usage and sequence of application of the performance tools in the form of the HOPSA performance-analysis workflow [9]. The workflow was also successfully used to structure training classes on the use of HOPSA tools, as it nicely captures the high integration of our tools set. As shown in Fig. 1, the workflow consists of three basic steps. During the first step ("Performance Screening"), we identify all those applications running on the system that may suffer from inefficiencies. This is done via system-wide job screening supported by a lightweight measurement module (LWM2) dynamically linked to every executable. The screening output identifies potential problem areas such as communication, memory, or file I/O, and issues recommendations on which diagnostic tools can be used to explore the issue further in a second step ("Performance Diagnosis"). If a more simple, profile-oriented static performance overview is not enough to pinpoint the problem, a more detailed, trace-based, dynamic performance analysis can be performed in a third step ("In-depth analysis").

The HOPSA performance tools are available as a combination of open-source offerings (the trace visualizer Paraver [6] and its measurement library Extrae and the associated performance modeling tool Dimemas [1] from BSC, the performance analysis tool Scalasca [2] and its result browser CUBE from GRS/JSC, and the community-developed performance instrumentation and measurement infrastructure Score-P [5]) and commercial products (the trace visualizer Vampir [4] from TUD and the thread and memory analyzer ThreadSpotter [3] from Rogue Wave). In the project, the individual tools have been considerably enhanced in their functionality and regarding scalability, enabling them to analyze parallel real-world applications executed with very large numbers (ten to hundred thousands) of processes and threads. Integration between the separate tool sets of the project partners also has been considerably improved. All enhancements are either already part of the latest public releases of the software packages, or at least are scheduled to be included in the next public release. Also, with the end of the project, a single unified installation package for all tools was provided [10].

Integration among the HOPSA Performance Analysis Tools

Sharing the common measurement infrastructure Score-P and its data formats and providing conversion utilities if direct sharing is not possible, the performance tools in the HOPSA environment and workflow already make it easier to switch from higher-level analyses provided by tools like Scalasca to more in-depth analyses provided by tools like Paraver or Vampir. To simplify this transition even further, the HOPSA tools are integrated in various ways. Fig. 2 gives an overview of the already implemented and envisioned tool interactions within the HOPSA tool set.

For example, with its automatic trace analysis, Scalasca locates call paths

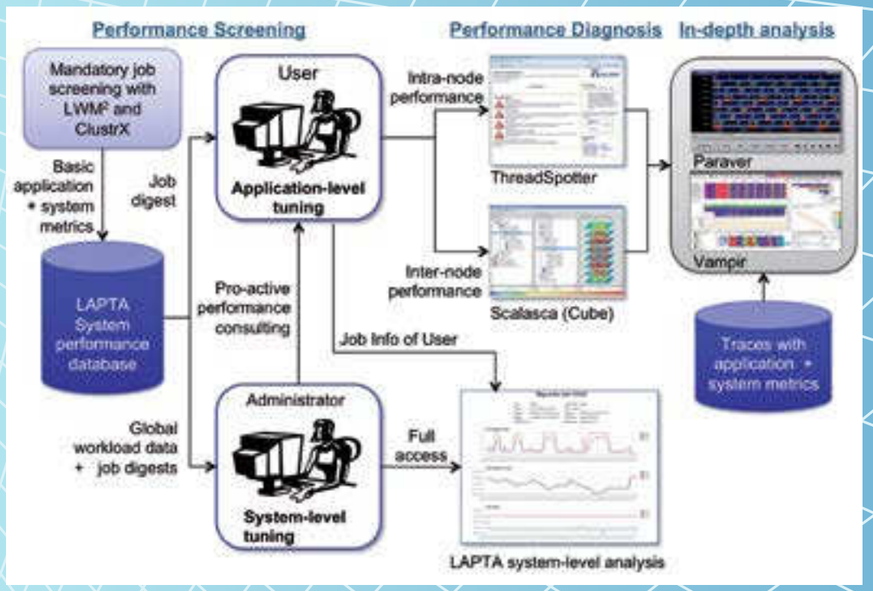

Figure 1: Overview of the performance analysis workflow.

affected by wait states caused by load or communication imbalance. However, to find and fix these problems in a user application, it is in some cases necessary to understand the spatial and temporal context leading to the inefficiency, a step naturally supported by trace visualizers like Paraver or Vampir. To make this step easier, the Scalasca analysis remembers the worst instance for each of the performance problems it recognizes. Then, the Cube result browser can launch a trace browser and zoom the timeline into the interval of the trace that corresponds to the worst instance of the recognized performance problems (see Fig. 3).

In the future, the same mechanisms will be available for a more detailed visual exploration of the results of Scalasca's root cause analysis as well as for further analyzing call paths involving user functions that take too much execution time. For the latter, ThreadSpotter will be available to investigate their memory, cache and multi-threading behaviour. If a ThreadSpotter report is available for the same executable and dataset, Cube will allow launching detailed ThreadSpotter views for each call path where data from both tools is available.

Integration of System Data and Performance Analysis Tools

The Russian ClustrX.Watch management software [7] and LAPTA system data analysis and management software [8] provides node-level sensor information that can give additional insight for performance analysis of applications with respect to the specific system they are running on. This allows populating Paraver and Vampir traces with LAPTA system information collected by Clustrx, Ganglia, and other sources (the granularity will depend on the overhead to obtain the data) and to analyze them with respect to the system-wide performance.

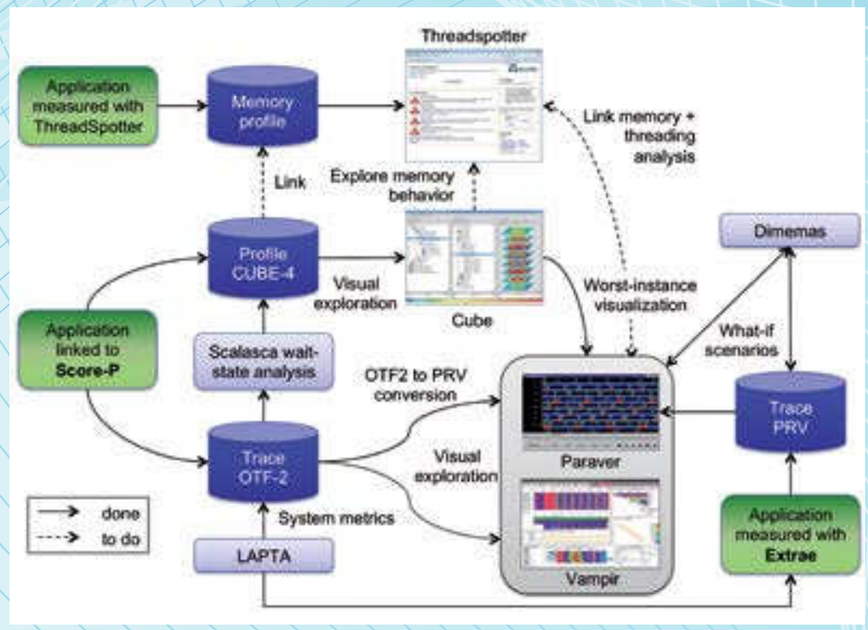

Figure 2: HOPSA Performance Tool Integration.

In the project, the Vampir team implemented a prototype Score-P adapter that enhances OTF2 traces at the end of the measurement. For evaluation, the benchmark code HPL was instrumented with Score-P. In addition to the application and MPI events, the trace was enhanced with HOPSA node-level metrics and per-process PAPI counters. Tested and working HOPSA sensors include node memory usage values and Infiniband packet counts. In the HPL code visualization (Fig. 4) one can see rising floating point operations (second timeline) resulting in a higher memory consumption per node (third timeline). Equivalent functionality was also implemented for the BSC tools Extrae and Paraver.

Conclusion

The HOPSA project delivered an innovative holistic and integrated tool suite for the optimization of HPC applications integrated with system-level monitoring. The tools are already used by the HPC support teams of project partners in their daily work. All results, documentation, and publication are available at the EU project website (http://www.hopsa-project.eu) or the Russian project website (http://hopsa.parallel.ru). For a short two-year project, dissemination was extremely successful: The project was presented at eleven events (including ISC and SC), often by multiple partners, 25 training events involving HOPSA tools have been organized, and 17 project-related publications have been published and presented at conferences.

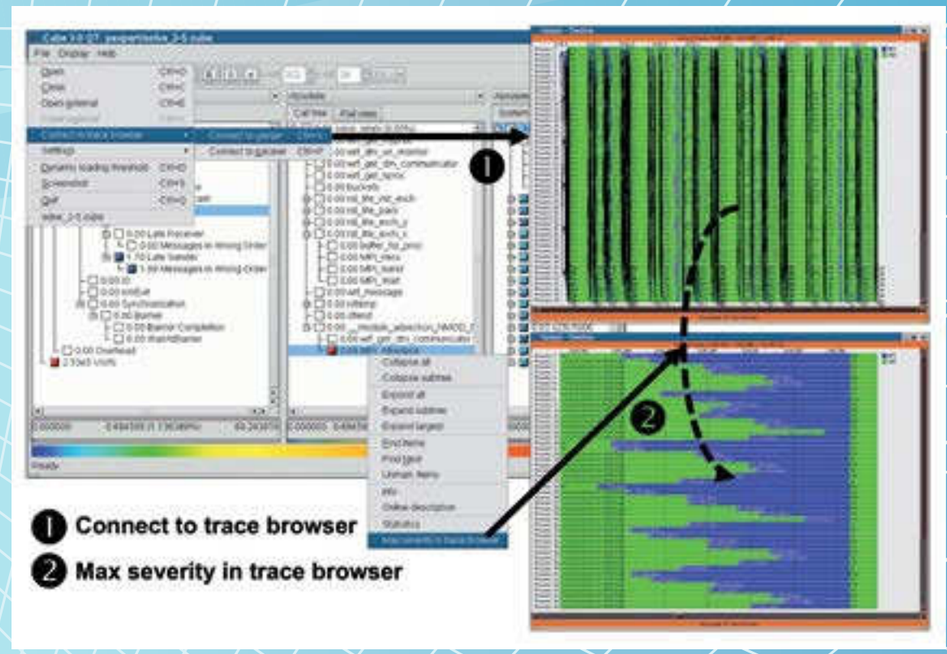

Figure 3: Scalasca →Vampir or Paraver Trace browser integration. In a 1st step, when the user requests to connect to a trace browser, the selected visualizer is automatically started and the event trace, which was previously the basis of Scalasca's trace analysis, is loaded. Now, in a 2nd step, the user can request a timeline view of the worst instance of each performance bottleneck identified by Scalasca. The trace browser view automatically zooms to the right time interval. Now the user can use the full analysis power of these tools to investigate the context of the identified performance problem.

Autumn 2013 • Vol. 11 No. 2 • inSiDE

Taking an integrated approach for the first time in an HPC context worldwide, the involved seven universities and research institutions considerably strengthened their scientific position as competence centres in HPC. Dresden University and Rogue Wave Software enriched their commercial software with unprecedented features and T-Platforms are to ship their HPC computer systems with the most advanced software offering, enabling all three of them to increase their respective market shares. Using the HOPSA tool infrastructure, the scientific output rate of a HPC cluster system can be increased in three ways: First, the enhanced tool suite leads to better optimization results, expanding the potential of the codes to which they are applied. Second, integrating the tools into an automated diagnostic workflow ensures that they are used both (i) more frequently and (ii) more effectively, further multiplying their benefit. Application programmers will ultimately benefit from higher HPC application performance by for example more accurate climate simulations or a faster market release of medication. Finally, the HOPSA holistic approach leads to a more targeted optimization of the interactions between application and system. In addition, the project resulted in a much tighter collaboration of HPC researchers from the EU and Russia.

EU Project Partners (HOPSA-EU)

- Forschungszentrum Jülich (EU Coordinator)
- Jülich Supercomputing Centre
- Barcelona Supercomputing Center
- Computer Sciences Department
- German Research School for Simulation Sciences
- Laboratory for Parallel Programming
- Rogue Wave Software AB (formerly ACUMEM)
- Technische Universität Dresden, Center for Information Services and High Performance Computing

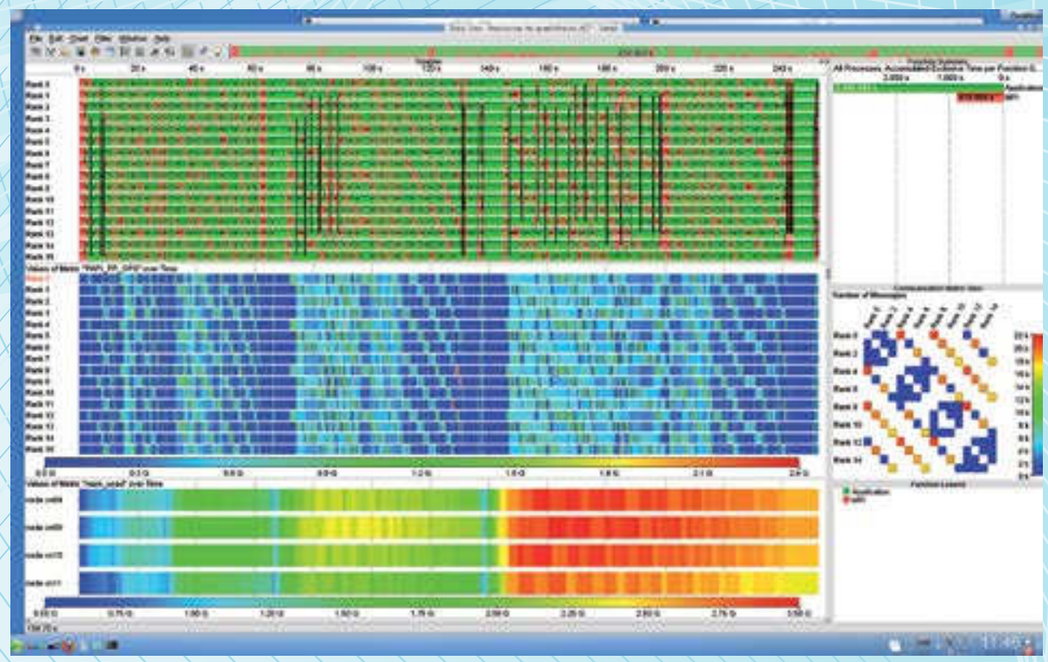

Figure 4: Vampir's Trace Visualization of the benchmark code HPL including the HOPSA node level metric "mem_used" (used memory) in the Performance Radar.

Russian Project Partners (HOPSA-RU)

- Moscow State University (RU Coordinator)
- Research Computing Center
- T-Platforms
- Russian Academy of Sciences
- Joint Supercomputer Center
- Southern Federal University, Scientific Research Institute of Multiprocessor Computer Systems

References

[1] **Labarta, J., Girona, S., Pillet, V., Cortes, T., Gregoris, L.**
DiP: A parallel program development environment, Proceedings of the 2nd International Euro- Par Conference, Lyon, France, Springer, 1996

[2] **Geimer, M., Wolf, F., Wylie, B.J.N., Abraham, E., Becker, D., Mohr, B.**
The Scalasca performance toolset architecture, Concurrency and Computation: Practice and Experience, 22(6):702–719, April 2010

[3] **Berg, E., Hagersten, E.**
StatCache: A Probabilistic Approach to Efficient and Accurate Data Locality Analysis, Proceedings of the 2004 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS-2004), Austin, Texas, USA, March 2004

[4] **Nagel, W., Weber, M., Hoppe, H.-C., Solchenbach, K.**
VAMPIR: Visualization and Analysis of MPI Resources. Supercomputer, 12(1):69–80, 1996

[5] **an Mey, D., Biersdorff, S., Bischof, C., Diethelm, K., Eschweiler, D., Gerndt, M., Knüpfer, A., Lorenz, D., Malony, A.D., Nagel, W.E., Oleynik, Y., Rössel, C., Saviankou, P., Schmidl, D., Shende, S.S., Wagner, M., Wesarg, B., Wolf, F.**
Score-P: A Unified Performance Measurement System for Petascale Applications. Competence in High Performance Computing 2010 (CiHPC), pp. 85–97. Gauß-Allianz, Springer, 2012

[6] **Servat, H., Llort, G., Giménez, J., Labarta, J.**
Detailed performance analysis using coarse grain sampling, Euro-Par 2009 - Parallel Processing Workshops, Delft, The Netherlands, August 2009, LNCS 6043, pp. 185–198. Springer, 2010

[7] T-Platforms, Moscow, Russia, Clustrx HPC Software: http://www.t-platforms.com/products/software/clustrxproductfamily.html, last accessed September 2012

[8] **Adinets, A.V., Bryzgalov, P.A., Vad, V., Voevodin, V., Zhumatiy, S.A., Nikitenko, D.A.**
About one approach to monitoring, analysis and visualization of jobs on cluster system (in Russian), Numerical Methods and Programming, 2011, Vol. 12, pp. 90–93

[9] **Mohr, B., Voevodin, V., Giménez, J., Hagersten, E., Knüpfer, A., Nikitenko, D.A., Nilsson, M., Servat, H., Shah, A., Winkler, F., Wolf, F., Zhujov, I.**
The HOPSA Workflow and Tools, Proceedings of the 6th International Parallel Tools Workshop, Stuttgart, September 2012, Springer, to appear

[10] Jülich Supercomputing Centre, Jülich, Germany, UNITE (UNiform Integrated Tool Environment): http://apps.fz-juelich.de/unite/, last accessed September 2013

• Bernd Mohr

Jülich Supercomputing Centre

Projects

Autumn 2013 • Vol. 11 No. 2 • inSiDE