Project Final Report

Grant Agreement number	277463
Project acronym	HOPSA-EU
Project title	Holistic Performance System Analysis-EU
Funding Scheme	CP (Collaborative project)
CALL	FP7-ICT-2011-EU-Russia
Period covered	February 1st, 2011 to January 31st, 2013
Project website	http://www.hopsa-project.eu

Scientific representative:

Dr.-Ing. Bernd Mohr
Forschungszentrum Jülich GmbH
Jülich Supercomputing Centre
52425 Jülich

Tel: +49 2461 613218
Fax: +49 2461 616656
E-mail: b.mohr@fz-juelich.de

Table of Contents

Glossary

Abbreviation / acronym	Description
API	Application Programming Interface
BSC	Barcelona Supercomputing Center, Spain
CEPBA	European Center for Parallelism of Barcelona (UPC, BSC)
ClustrX	Cluster monitoring system (T-Platform)
CUBE	Performance report explorer for Scalasca (JSC) and Score-P (GRS, JSC, TUD)
CUDA	Compute Unified Device Architecture (Proprietary Programming Interface for Nvidia GPGPUs)
Dimemas	Message passing performance analysis and prediction tool (BSC)
Extrae	Instrumentation and measurement component for Paraver visualizer (BSC)
GRS	German Research School for Simulation Sciences GmbH, Aachen, Germany
GPGPU	General Purpose Graphical Processing Unit
GUI	Graphical User Interface
HMPP	Hybrid Multicore Parallel Programming (Proprietary Programming Model for Heterogeneous Architectures)
HOPSA	HOlistic Performance System Analysis. EU FP7 project
HPC	High Performance Computing
H4H	Hybrid Programming For Heterogeneous Architectures. EU ITEA2 project
I/O	Input/Output
JSC	Jülich Supercomputing Centre (of Forschungszentrum Jülich GmbH), Germany
LAPTA	Database and analysis system for cluster monitoring data (MSU)
LWM^2	Light Weight Monitoring Module (GRS) (Used for system-wide application performance screening)
MPI	Message Passing Interface (Programming Model for Distributed Memory Systems)
MSU	Moscow State University
OpenCL	Open Computing Language (Programming interface for heterogeneous platforms consisting of CPUs and other execution units like GPUs)
OpenMP	Open Multi-Processing (Programming Model for Shared Memory Systems)
OTF2	Open Trace Format Version 2

PAPI	Performance Application Programming Interface (Library for portable access to hardware performance counter)
Paraver	Event trace analysis and visualization tool (BSC)
PMPI	Standard monitoring API for MPI
RW	Rogue Wave Software AB, Sollentuna, Sweden
Scalasca	SCalable Analysis of LArge SCale Applications (Performance instrumentation, measurement and analysis tool from JSC/GRS)
Score-P	Scalable Performance Measurement Infrastructure for Parallel Codes (Community open-source project of GRS, JSC, TUD and others)
SMPSs	Pragma-based programming model for parallel task (Ss = Superscalar) for shared memory parallel computers (SMP) from BSC
UPC	Universitat Politècnica de Catalunya, Barcelona
T-Platforms	Russian HPC cluster vendor
ThreadSpotter	Commercial memory and multi-threading performance analysis tool (RW)
TUD	Technische Universität Dresden, Germany
UNITE	UNiform Integrated Tool Environment (Unified documentation and installation procedures for HPC tools)
Vampir	Visualization and Analysis of MPI Resources (Commercial event trace analysis and visualization tool from ZIH/TUD)
VampirTrace	Instrumentation and measurement component for Vampir visualizer (ZIH/TUD)
ZIH	Zentrum für Informationsdienste und Hochleistungsrechnen. (Center for information services and HPC of TUD).

1. Final publishable summary report

1.1 Executive summary

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 sets out for the first time for combined application and system tuning in the HPC context 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 indented usage and application of the performance tools was specified and documented in the HOPSA performance-analysis workflow. 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. 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 (LWM^2) 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 pin-point 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 (Extrae, Paraver, Dimemas, Scalasca, CUBE, Score-P) and commercial products (Vampir, ThreadSpotter). 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.

The HOPSA project delivers 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 Russian project website (***http://hopsa.parallel.ru***). 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.

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. European citizens 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.

1.2 Summary description of project context and objectives

Computer-based simulation will be a key technology of the 21st century. Numerous examples ranging from the improved understanding of matter to the discovery of new materials – and from there to the design of complete cars, ships, and aircrafts – give evidence of its tremendous potential for science and engineering. Mastery of this technology will decide not only on the economic competitiveness of a society but will ultimately influence everything that depends on it, including the society's welfare and stability. Moreover, there is broad consensus that computer simulation is indispensable to address major global challenges of mankind such as climate change and energy consumption.

As a natural consequence of this insight, the demand for computing power needed to solve the numerical equations behind simulation models of rapidly increasing complexity is continuously growing. In their effort to answer this demand, supercomputer vendors work alongside computing centres to find good compromises between technical requirements, tight procurement and energy budgets, and market forces that dictate the prices of key components. The results are innovative architectures that combine unprecedented numbers of processor cores into a single coherent system, leveraging commodity parts or at least their designs to lower the costs where it is still in agreement with design objectives. The current trend favours shared-memory nodes linked with fast interconnects, where each of the nodes may offer one or more sockets available to multicore processors. As a common trend that can be observed in response to the proliferation of multicore chips with their rising numbers of cores per die, the shared-memory nodes most clusters are composed of are becoming much wider. Inside a node, multiple levels of cache exist with varying degrees of sharing between cores. Data items travel along complex network hierarchies including inter-node links as well as node-internal buses or switching networks. Different latencies and bandwidths encountered on their way have to be taken into account to achieve satisfactory performance. In addition, memory is increasingly recognised as a limiting factor - not only in terms of bandwidth and latency but also in terms of manufacturing cost and energy consumption, which is why many experts expect the memory-per-core ratio to shrink in the future.

One alternative to enhance the performance of general purpose computers are field programmable gate arrays (FPGAs), whose functionality can be configured by the customer or designer after manufacturing. Although their flexibility combined with their low non-recurring engineering costs offer advantages, they so far found adoption only among a limited set of HPC applications. In contrast, a larger number of recent cluster architectures take advantage of powerful graphics processing units (GPUs), which evolved during the past decade from specialised graphics hardware to general purpose streaming processors. Originally designed for the consumer electronics market, for which they are produced in large quantities, they offer a very competitive price-performance ratio, exploiting the economy of scale – not to mention the low energy demand of their relatively simple control logic. Most suitable for highly data-parallel workloads, heterogeneous systems composed of GPUs attached to standard CPUs have been found to support remarkable speedups for a broad spectrum of scientific and engineering workloads. Nevertheless, vendors are currently experimenting with a number of heterogeneous design options and it is hard to predict which technology will prevail in a few years from now.

Regardless of where the journey goes, the big picture is expected to remain stable at least for the near future: We will have to deal with hierarchical systems where each level may support parallelism in a different way. On the software side, this is reflected by the increased use of hybrid programming practices. Hybridisation refers to the combination of different parallel programming models in a single application to allow different levels of parallelism to be exploited in a complementary manner. For example, in response to widening shared-memory nodes, many code developers now resort to using OpenMP for node-internal work sharing, while employing MPI for parallelism among different nodes. This has the advantage that (i) the extra memory needed to maintain separate private address spaces (e.g., for ghost cells or communication buffers) is no longer needed, (ii) the effort to copy data between these address spaces can be reduced, and (iii) the number of external MPI links per node can be kept at a minimum to improve scalability. Another motivation for hybrid programming are GPUs. The main program out-sources small but frequently executed core computations to a node-local GPU installed as an accelerator of the main processor.

1.2.1 Motivation

Having the potential to improve efficiency and scalability, hybridisation usually comes at the price of increased programming complexity. Given that critical applications are developed by large multidisciplinary teams over long periods of time, code development and maintenance are significant cost factors in addition to the procurement and operation of the necessary hardware. To ameliorate unfavourable effects of hybrid parallelisation on programmer productivity, developers therefore depend even more on powerful and robust performance analysis and optimisation tools that help them tune the performance of their codes. While such tools already exist, their capabilities are still limited and they are rarely used in a concerted way to compensate the weaknesses of one with the strengths of another. Sometimes, performance problems even go unnoticed as long as the allocation of compute time is large enough to obtain the desired results or because the application developer lacks the time and/or expertise to address them. Finally, users often simply do not know which tool will offer them the insights they need most.

While the above considerations discuss performance from the perspective of an individual application, there is also the view of system providers who want to maximise the scientific output of their users. This can be achieved in two ways, either by optimising application performance or system throughput. System throughput is influenced by several factors including system configuration, scheduling decisions, faulty components, and system software. Some of them cannot be easily changed, while others can. For example, OS jitter, which can degrade application performance significantly, can be reduced by disabling unnecessary OS daemons. As systems grow bigger, so-called hardware jitter triggered by unexpected component failures becomes increasingly aggressive. Affecting especially long-running codes that use tens of thousands of processor cores, monitoring the hardware for early signs of failure such as raised temperatures can mitigate this effect to some degree.

In general, the output rate of a whole system in terms of science per time and energy unit, and thus its return on investment, depends on decisions made at both the system and the application level, that is, on decisions usually made by different groups of people. Likewise, the performance of an individual application depends on both the way it is coded and the way the underlying system is configured. A prominent example for this interrelationship is parallel I/O, whose performance responds to file access patterns as sensitively as it reacts to changes in the file-system layer. More than often, applications themselves mutually degrade their runtimes when accessing global resources such as the file system or the network. However, in spite of all these obvious insights, it is still common practise that system administrators and users carry out the optimisation of their systems and applications separately from each other, often without exchanging important findings relevant to the other party. In particular, the potential of systematic application screening for system throughput optimisation is still largely untapped.

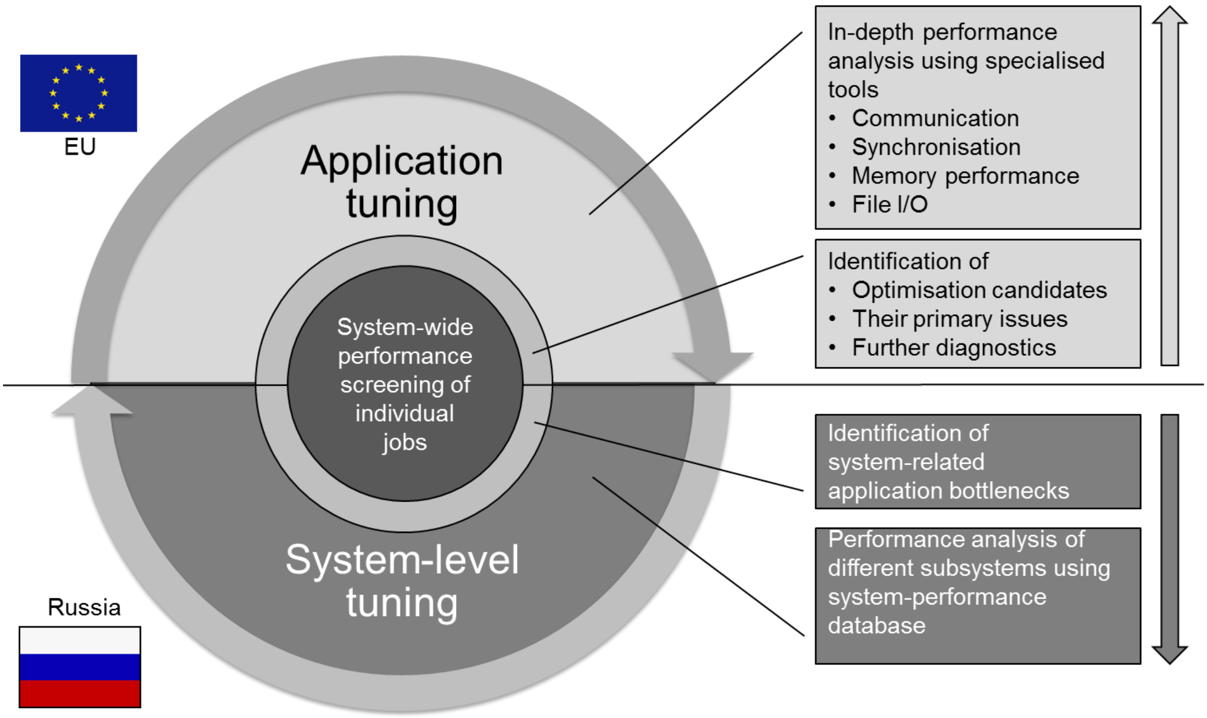

Figure 1: System-level tuning (bottom), application-level tuning (top), and system-wide performance screening (centre) use common interfaces for exchanging performance properties

1.2.2 Objectives

The main objective of the two coordinated proposals (HOPSA-EU, grant no. FP7-277463; coordinated with the Russian project, contract no. 07.514.12.4001 of Dec 24th, 2010, project duration Jan 14, 2011 to Dec 15th, 2012, from now on called HOPSA-RU) is therefore the integration of application tuning with overall system diagnosis and tuning to maximise the scientific output of our HPC infrastructures. On a technical level, we will give emphasis to the specific problems of hybrid parallel applications encountered on heterogeneous hierarchical systems. While the Russian consortium will focus on the system aspect, the EU consortium will focus on the application aspect. At the interface between these two facets of our holistic approach, which is illustrated in Figure 1, will be the system-wide performance screening of individual jobs, pointing at both inefficiencies of individual applications such as high communication overhead and system-related performance issues such as above-average waiting time in the queue. In the following, we will describe only the objectives and tasks of the EU project, for the Russian part please refer to their report. The EU consortium will pursue the following subgoals:

1. Basic end-to-end performance analysis for all jobs running on a given system from their submission to their completion. This will be accomplished by analysing the raw binary behaviour in combination with using a light-weight performance monitoring module linked to the application prior to its execution.
2. Identification of key performance issues and notification of the user and system performance database after job completion. This will also include recommendations to the user on how to conduct further diagnostics using the tool suite provided by the consortium.
3. Enhancement of individual tools in the suite to make them fit for petascale computations and beyond as well as integrating them with each other where useful. The idea here is not to start new research directions, but rather to finalise (i.e., “productise”) current research ideas and make them part of our regular tool products.

The light-weight monitoring module, which will be implemented as a shared library so that it can be loaded prior to the execution of the parallel job by the job launcher, will collect basic performance metrics such as execution time, hardware counters, and message-passing statistics. To keep the overhead at a minimum, only those metrics will be collected which do not require expensive instrumentation. The output of the module will be enriched with additional data from the Russian batch system and system hardware sensors to generate a job digest report with the following information.

Depending on the outcome, the user will be guided through a well-defined workflow of diagnostic procedures supported by our tool suite, which includes the ThreadSpotter, Dimemas, Paraver, Scalasca, and Vampir. The tools cover a wide range of performance aspects such as communication and synchronisation, memory access, and I/O. Most of them already provide ample support for MPI/OpenMP hybrid programming. Enhancements of the individual tools will cover the following aspects:

- Scalability: Methods and algorithms to make the tools more scalable in terms of both the number of cores and the length of execution.
- Analysis of asynchronous tasking: Emerging programming models employ the concept of asynchronous tasks. Examples are OpenMP 3.0 or StarSs tasks, CUDA, OpenCL, and generic uncoordinated (POSIX) threading. In HOPSA, we will develop an abstract characterisation of the performance of asynchronous tasking which will allow all tools to support these new models in a coherent way.
- Root cause analysis: Current tools tend to report more the symptoms of performance problems than the actual cause. Methods to locate the root cause of performance bottlenecks need to be improved and further developed.
- Tool integration: One performance tool is typically not enough to measure and analyse all aspects of the dynamic behaviour of parallel programs. To allow the user to employ all HOPSA tools in a coherent way, we will develop an overall performance tool workflow, provide a single configuration and installation package for all tools, and enhance tool interactions. For example, Scalasca’s interactive report explorer could be used to drive the detailed analysis with Vampir or Paraver, and the performance data exchange between the different tools could be simplified.

A final objective of the HOPSA project is also to provide performance tools which support hybrid programming for heterogeneous architectures. However, due to the complexity and immaturity of this area, a short (two-year) and small (five partners) project like HOPSA alone cannot provide any major breakthrough here. Therefore, we will leverage some work from other projects in which the HOPSA partners are involved:

General support of the performance tools of the HOPSA partners for the measurement and analysis of programming for heterogeneous computing is expected to be provided via the H4H project (Oct 2010-Sep 2013), which is funded through the European ITEA-2 program. Almost every performance tool group of HOPSA, i.e., ThreadSpotter(RW), Scalasca (JSC/GRS), and Vampir (TUD), is also partner in the H4H project. Unfortunately, the Paraver group (BSC), originally participant in the H4H proposal, could not participate in the project due to funding issues with the Spanish government.

In the context of H4H, the Score-P measurement system (which is the future common measurement system for both of the Scalasca and Vampir toolsets) will be extended to allow measurement of low-level API-based (CUDA, OpenCL) and high-level pragma-based (GPUSs, HMPP) programming models for heterogeneous systems, and Vampir and Scalasca will be enhanced to analyse and visualize high-level parallel programming constructs. Rogue Wave will extend its capability to optimize execution for non-uniform memory architectures (NUMA), such as some existing multicore chips (e.g., AMD Magny Cours) or nodes built from several multi-core chips, such as most HPC servers sold today. In addition, H4H will provide integration with the research groups on programming models, e.g., GPUSs (Univ. Jaume) and HMPP (CAPS), which is important for the efficient and effective implementation of the performance measurement and analysis modules; something we cannot accomplish in the HOPSA project due to its small size and short duration. Finally, integration of the StarSs programming model from BSC and the Scalasca toolset is funded through the EU FP7 project TEXT (Jun 2010 – May 2012).

Once the enhancements for the measurement and analysis of heterogeneous architectures of ThreadSpotter, Scalasca, and Vampir are available from the H4H and TEXT projects, they will be integrated into the regular product versions of the tools which in turn will be part of the HOPSA unified tools package. Of course, the necessary aspects of heterogeneous architectures will be also considered in the definition of the interface between system-level and application level performance analysis and in the definition of the overall performance analysis workflow.

In summary, all resources and development work necessary for supporting performance analysis of programming models for heterogeneous architectures (e.g. CUDA or HMPP) will be done in other projects; however these results will also be very useful in the context of the HOPSA project.

1.3 A description of the main S&T results/foregrounds

In the following, we describe the two major outcomes of the HOPSA project: the well-defined performance analysis workflow, which guides application developers through the process of analysing and optimizing their codes, and integrated HOPSA performance tool set.

1.3.1 The HOPSA Workflow

1.3.1.1 Overview

The performance-analysis workflow (Figure 2) consists of two basic steps. During the first step, 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 (LWM^2) 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. Available application performance analysis tools include Paraver/Dimemas [1][12], Scalasca [2], ThreadSpotter [3], and Vampir [4]. The data collected by LWM^2 is also fed into the Clustrx.Watch hierarchical cluster monitoring system [13] which combines it with system and hardware data and forwards it to the LAPTA cluster monitoring and analysis system [14] for further analysis by system administrators.

In general, the workflow successively narrows the analysis focus and increases the level of detail at which performance data are collected. At the same time, the measurement configuration is optimised to keep intrusion low and limit the amount of data that needs to be stored. To distinguish between system and application-related performance problems, some of the tools allow also system-level data to be retrieved and displayed. The system administrator, in contrast, has access to global performance data. He can use this data to identify potential system performance bottlenecks and to optimise the system configuration based on current workload needs. In addition, the administrator can identify applications that continuously underperform and proactively offer performance-consulting services. In this way, it becomes possible to reduce the unnecessary waste of expensive system resources.

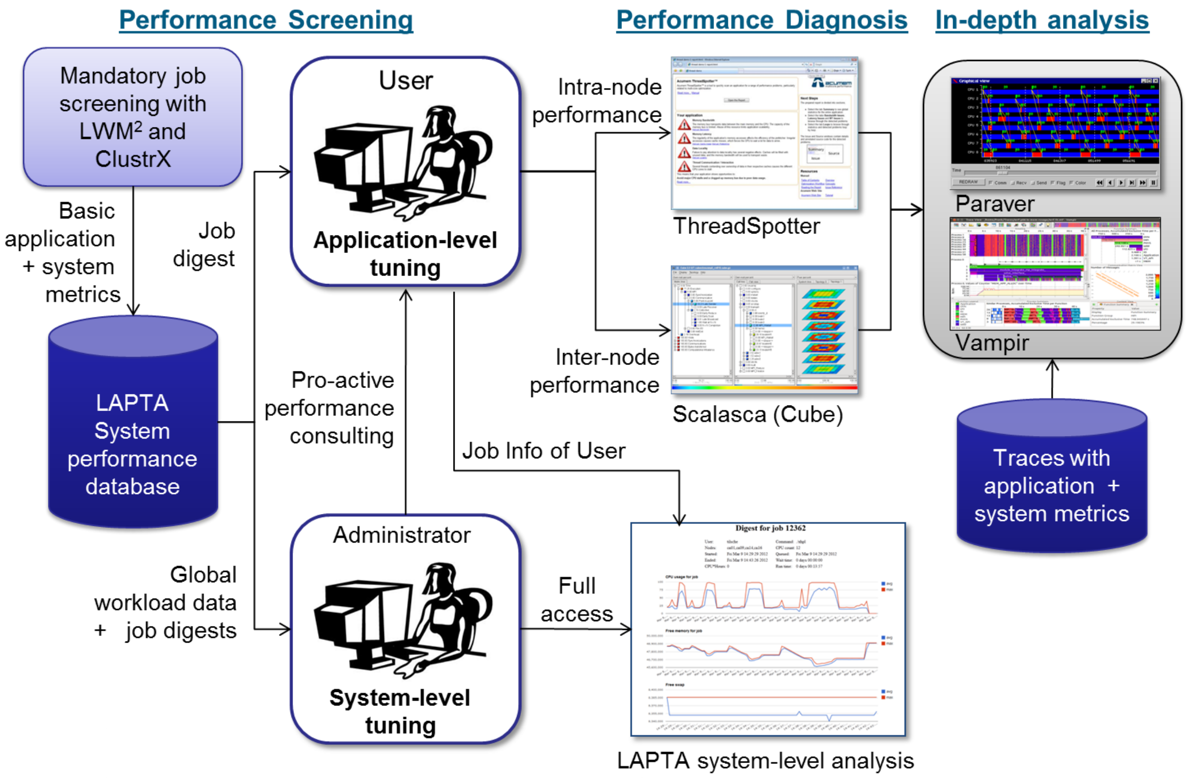

Figure 2: Overview of the performance analysis workflow.

1.3.1.2 Performance Screening

This step decides whether an application behaves inefficiently. On the side of the user, nothing has to be done except running the application as usual. Upon application start, LWM^2 is automatically and transparently linked to the executable through library pre-loading. At runtime, the module collects basic performance data with very low overhead. The performance data characterise various aspects such as sequential performance, parallel performance, and file I/O. At the end of execution, the user receives a job digest that contains the most important performance metrics. The digest also recommends further diagnostics in the case certain key metrics show unexpected values, which may often be indicative of a performance problem. If needed, the user can disable LWM^2, for example, to avoid interference with the analysis tools used in subsequent stages of the tuning process.

1.3.1.2.1 The Lightweight Measurement Module LWM^2

The lightweight measurement module LWM^2 collects basic performance data for every process of a parallel application. It supports applications based on MPI and multithreaded applications based on POSIX Threads or any higher-level model implemented on top of it, which usually includes OpenMP. Multithreaded MPI applications and applications that additionally use CUDA are supported as well. To keep the overhead at a minimum, the module applies a combination of sampling and careful direct instrumentation via interposition wrappers. Direct instrumentation is needed to track the state of a thread (e.g., whether it executes inside or outside an MPI function) and to access relevant communication or I/O parameters such as the number of bytes sent or written to disk. Based on the state tracking performed by the instrumentation, sampling partitions the execution time into different components such as computation, communication, or I/O. LWM^{2}refrains from direct time measurements as far as possible. Hardware counters deliver basic information on single-node performance. To save storage space, the performance data of individual threads are folded into per-process metrics such as the average number of threads.

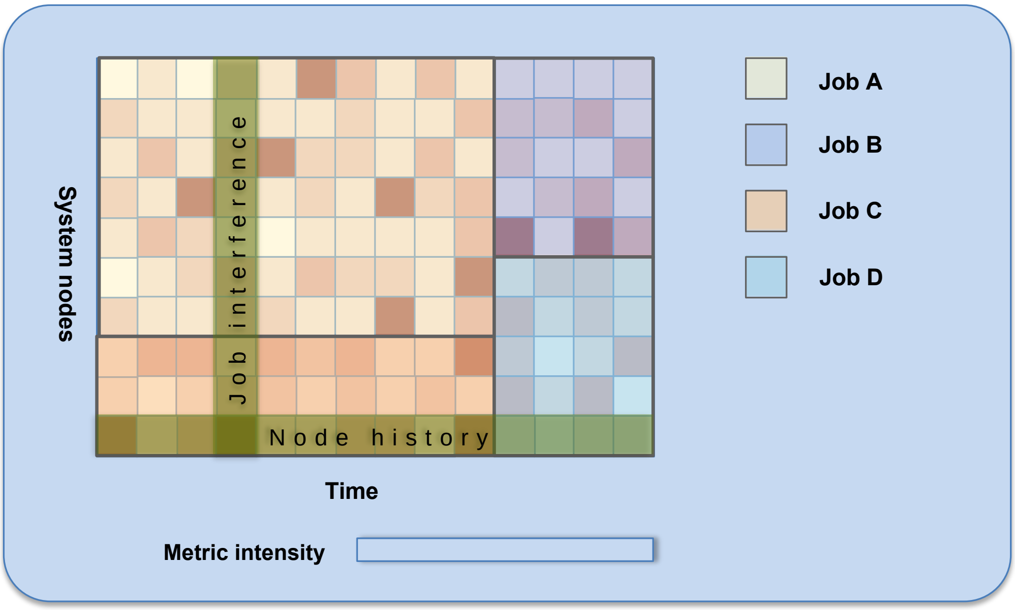

Figure 3: LWM^2 maps selected performance metrics collected during program execution onto a time-and-space grid. The space dimension consists of system nodes, while the time dimension consists of time slices of length 10s, which are synchronized across the entire

In addition to collecting performance data separately for each process, LWM^2 divides the time axis into disjoint slices, recording selected metrics related to the use of shared resources at this finer granularity. The slices have a length of 10s and are synchronized across the entire machine. Together with the location of each process on the cluster, which LWM^2 records along with the performance data, LWM^2 provides performance data for each active cell of a cluster-wide time-space grid (see Figure 3). The discretised time axis constitutes the first dimension, the nodes of the system the second one.

The purpose of organising the performance data in this way is threefold: First, by comparing the data of different jobs that were active during the same time slice, it becomes possible to see signs of interference between applications. Examples include reduced communication performance due to overall network saturation or low I/O bandwidth due to concurrent I/O requests from other jobs. Second, by looking at the performance data of the same node across a larger number of jobs and comparing it to the performance of other nodes during the same period, anomalies can be detected that would otherwise be hidden when analysing performance data only on a per-job basis. Third, collecting synchronised performance data from all the jobs running on a given system will open the way for new directions in the development of job scheduling algorithms that take the performance characteristics of individual jobs into account. For example, to avoid file-server contention and waiting time that may occur in its wake, it might be wiser not to co-schedule I/O-intensive applications. In this way, overall system utilisation may be further improved.

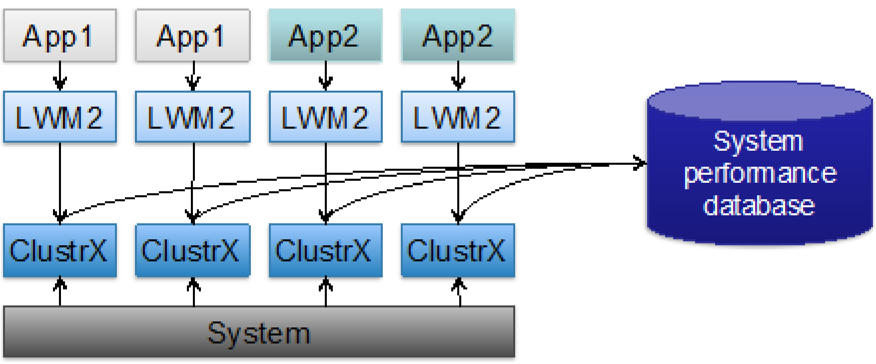

Figure 4: Interaction between LWM^2 and Clustrx.Watch.

After the expiration of every time slice, LWM^2 passes the data of the current time slice on to Clustrx.Watch, a system-monitoring infrastructure running on each node. Clustrx augments these data with system data collected using various sensors and forwards them to the LAPTA system performance database as shown in Figure 4.

1.3.1.2.2 How to disable LWM^2

While the low runtime overhead of LWM^2 guarantees the undisturbed program execution under normal circumstances, it may interfere with some of the other performance tools that employ similar mechanisms to collect their performance data. For this reason, a user can disable the preloading of LWM^2 in the batch script. The precise way of specifying this option still needs to be determined. The most likely solution is to set an LWM^2-specific batch system variable for disabling LWM^2 for the current batch job.

1.3.1.2.3 LAPTA

LAPTA is a pAckage for Performance moniToring and Analysis. The software is aimed at providing flexible, scalable and extendable infrastructure for system-level performance analysis. It includes special tools and interfaces for data collection supporting various data collectors (Clustrx, Ganglia, LWM^2, etc.), data storage supporting wide range of databases (MongoDB, Cassandra, etc.) and both stored and streamed data access and analysis. LAPTA provides interfaces to access the collected system monitoring data for both query models: post mortem and on-the-fly. For example LAPTA serves as the basis for Job Digest generation based on system-level performance monitoring data. The screening of general job behavior through Job Digest is very useful for users and tuners to understand the possible bottlenecks that can be seen at a glance (like network overload, bad data locality, inefficient memory usage, too intensive I/O, etc.). Also, performance data of the same application collected over an extended period of time will document the tuning and scaling history of this application allowing further even more detailed analysis of the dynamic application behavior. Studying the performance behaviour of the entire job mix will allow making conclusions on the optimal system configuration for the given workload. For example, system providers will learn whether requirements to amount of physical memory available, I/O or network bandwidth and other system hardware requirements were over- or underestimated.

1.3.1.2.4 Job digest

How to access it

The job digest is accessible after job completion through a web-based interface. When the user logs onto the performance database, a list of completed jobs will appear. After selecting a specific job, summary metrics for this particular job will be displayed. For those metrics, for which time-sliced data is available, the user can view graphs that show the evolution of these metric over time. An important feature is the ability to correlate the evolution of different metrics over time by comparing their graphs. Since the graphs for a single application run cover the same time interval, correlation can be easily observed.

Performance metrics

Below, we provide a detailed list of metrics contained in the job digest. For all metrics where it is applicable, the digest lists minimum, average and maximum values across processes. In addition to defining metrics, we also provide guidance in interpreting them and make recommendations on further analyses if a given metric or group of metrics does not match expectations. In general, it is highly application-dependent whether a metric value should be considered too high or too low. We therefore do not define any fixed thresholds but rather refer to the expectations a user may have.

General information

- Duration of the job in terms of wall clock time
- Number of MPI processes

Message-passing performance

- Time spent in all MPI calls [%]
- Time spent in MPI point-to-point calls [%]
- Time spent in MPI collective calls [%]

- Average size of point-to-point messages [Byte]
- Average size of collective messages sent [Byte]
- Average size of collective messages received [Byte]
- Frequency of MPI point-to-point calls [/s]
- Frequency of MPI collective calls [/s]
- MPI point-to-point transfer rate [Byte/s]. Ratio of the number of bytes sent and the time spent in MPI point-to-point communication
- MPI collective transfer rate [Byte/s]. Ratio of the number of bytes sent and the time spent in MPI collective communication

In general, message passing means communication or synchronisation as opposed to computation and therefore does not directly contribute to the calculation of results. Therefore, communication should be minimized as much as possible and the fraction of time spent in MPI kept low. If the fraction of time spent in MPI calls grows with the number of processes, the application has usually a scalability problem. If communication is dominated by larger numbers of small messages, network latency may be the limiting factor. In contrast, if the majority of messages are large, the limiting factor may be network bandwidth. Asymmetries in the MPI time across processes, indicated by different minimum and maximum times, can be signs of load or communication imbalance, a performance property that usually prevents scaling to larger processor counts. In any case, as the workflow in Figure 2 suggests, Scalasca is the first tool that should be used to analyse communication performance. After identifying the main problems using Scalasca, further, more detailed analysis can follow using Paraver/Dimemas or Vampir. Finally, low communication performance may also be caused by application interference when multiple jobs that run simultaneously compete for the network. This can be verified by comparing the temporal evolution of communication metrics such as the frequency of MPI point-to-point calls with the system-wide communication volume during a given interval.

I/O performance

- Time spent in MPI file I/O calls [%]
- Time spent in POSIX file I/O calls [%]
- Amount of data written to files [Byte]
- Amount of data read from files [Byte]
- Write bandwidth [Byte/s]. Ratio of the number of bytes written to files and the time spent in write functions
- Read bandwidth [Bytes/s]. Ratio of the number of bytes written to files and the time spent in read functions

These metrics indicate whether the application places too much load on the I/O subsystem. The user should always check whether I/O of the given application coincides with I/O of other applications, which is visible in the web-based digest. In such a case, the I/O performance may improve in subsequent runs when such interference is absent. In general, I/O performance is subject to variation and may change significantly between runs. This means, diagnosing an I/O bottleneck usually requires multiple runs under different overall load conditions. Scalasca may help identify expensive I/O calls, while Vampir can help visually discern the overall I/O pattern.

Multithreaded performance

- Average number of threads for the execution: Ratio of the total number of samples and the number of samples taken on the master thread
- Total number of threads in the execution

The average number of threads tells whether the degree of concurrency is as expected. For example, long periods of sequential execution in OpenMP applications may degrade concurrency and limit the benefits of parallel regions for the overall program. Again, Scalasca can help identify places in the code where the concurrency is low, while Paraver and Vampir may provide detailed insights into the change between sequential and parallel phases. Moreover, if the application fails to scale linearly when adding more threads, it could mean that the increased pressure on the memory subsystem causes threads to stall for increasing amounts of time. ThreadSpotter may help figure out the reason.

Sequential performance

- Average cycles per instruction (CPI)

- Fraction of floating-point operations among all instructions [%]
- L1 data cache hit ratio
- Last-level miss frequency

Sequential-performance metrics tell how well the cores of the underlying machine are utilized. If the cycles per instructions are much higher than the theoretical minimum, then memory access latency or pipeline hazards may be the reason. Also, some operations such as complex floating-point operations may simply take longer than others. The fraction of floating-point operations tells to which degree floating-point performance is the dominant theme. A low L1 hit ration usually indicates low locality and may explain a high CPI value. The last-level miss frequency is equivalent to the frequency of main-memory accesses and may point to memory-bandwidth saturation. Note that a platform may miss some of the hardware counters required for the full set of sequential performance metrics or that some of the required hardware counters cannot be measured simultaneously. In this case, LWM^2 provides only a subset of the above metrics. ThreadSpotter is the first candidate to explore memory performance issues. The folding analysis of Paraver can also shed light on high CPI values as it shows correlations with other hardware metrics [12].

CUDA performance

- Time spent in CUDA calls [%]
- Average data volume transferred from host to device [Byte]
- Average data volume transferred from device to host [Byte]
- Frequency of data transfers [/s]

These metrics provide just a very rough indicator of CUDA performance. If these metrics show unexpected values, Scalasca may help identify expensive CUDA calls. Paraver and Vampir may give additional insight.

System-oriented metrics

In addition to the more application-oriented metrics listed above, we also plan to include system-oriented metrics related to CPU usage and network communication health, which are collected by Clustrx.

1.3.1.2.5 Application interference

As LWM^2 is used with every job running on the system, and the data from the system side is also collected continuously from the complete system, it is possible to present global summary metrics in the job digest. The main examples for application interference are:

- Average I/O load: If the file system was in heavy use by other jobs running on the system at the time the current job was trying to access the file system, it is possible that this interference caused significant performance degradation.
- Average network load: If the communication network was in unusually heavy use at certain time intervals during the execution of the current job, it is important to be able to identify these time periods. If the current job had unusually low network performance in this interval, the reason was probably the interference from other jobs, and not a performance problem with the job itself.

In both of these cases, heavy system load happening at the same time when the current job is trying to use the shared resources leads to performance degradation. Therefore, it is crucial to be able to correlate performance data that was measured at the same time. As these metrics are collected in a time-sliced manner, we can correlate events happening at the same time with a granularity of 10 seconds, which is the default value for the length of time slices.

1.3.1.2.6 How to access the performance database

There are multiple use cases for accessing the database, with widely different characteristics:

- When a user accesses the database, he can view metric data collected about his own jobs, which gives him insight into the performance characteristics of a single job, and also allows for comparison between different executions of the same executable.
- When the system administrator accesses the database, he can get a complete overview about all jobs in a given time interval, which allows for pinpointing jobs with sub-standard execution performance characteristics.
- A special case of the previous use case is when a data mining algorithm is ran on the database to pinpoint problematic jobs automatically, making the system administrator’s work much more efficient.

1.3.1.3 Performance Diagnosis

This step decides why an application behaves inefficiently. It is only needed if the screening identifies a potential performance problem. Depending on the recommendation made by LWM[2], the user chooses one or more of the performance-analysis tools offered by the HOPSA tool environment. The general strategy of the diagnosis is to start with an overview and then to go deeper as more information on the problem's root cause becomes available.

1.3.1.3.1 Overview of the Performance Analysis Tool Suite

An overview of the HOPSA performance analysis tool suite is presented in Table 1. For the analysis of intra-node performance, ThreadSpotter is the primary tool, with the possibility of more detailed analyses using Paraver. For investigating internode performance, looking at a performance profile using Scalasca's Cube browser is a good starting point. For even more detailed analyses, the results of the Scalasca trace-analyser can be displayed in Cube, or the Vampir and Paraver/Dimemas tools can be used for a detailed visual exploration of the traces. For understanding I/O- related issues, profiles displayed in the Cube browser give a good overview, while Vampir can be used for more in- depth analysis.

	Inter-node performance	**Intra-node performance**	**I/O**
Overview	ThreadSpotter	Scalasca (Cube)	Scalsca (Cube)
In-depth analysis	ThreadSpotter, Paraver, Dimemas	Scalasca trace analyser + Cube, Paraver, Vampir	Vampir

Table 1: Classification of tools based on problem class and level of detail.

1.3.1.3.2 The Score-P Instrumentation and Measurement System

The Score-P [5] measurement infrastructure is a highly scalable and easy-to-use tool suite for profiling, event tracing, and online analysis of HPC applications. It collects performance data that can be analysed using the HOPSA tools Scalasca and Vampir. In addition, it supports the performance tools Persicope [6] and TAU [7] developed outside the HOPSA project. Score-P has been created in the projects SILC and PRIMA funded by the German Ministry of Education and Research and the US Department of Energy, respectively. It will be maintained and further enhanced in a number of follow-up projects including HOPSA.

The main performance data formats produced by Score-P are CUBE-4 [8] for profiles and OTF2 [9] for event traces. Profiles provide a compact performance overview, while event traces allow the in-depth analysis of parallel performance phenomena. While classic profiles aggregate performance metrics across the entire execution, time-series profiles treat individual iterations of the application's main loop separately, which allows studying the temporal evolution of the performance behaviour. They provide less detail than event traces, but can cover longer executions. Together, the above-mentioned options form a hierarchy of performance data types with increasing level of detail. The main advantage of Score-P is that a user needs to become familiar with only one set of instrumentation commands to produce all these data types, which can be analysed using the majority of the tools listed Table 1. Figure 5 provides an overview of the different performance data types supported by Score-P and the tools that can be used to analyse them. Below we cover the individual data types in more detail.

Profiles

Profiles in the CUBE-4 format map a set of performance metrics such as the time spent on some activity or the number of messages sent or received onto pairs of call paths and processes (or threads in multithreaded applications). Metrics with a specialization (i.e., subset) relationship can be arranged and displayed in a hierarchy. The call-path dimension forms the natural call-tree hierarchy. Processes and threads are also arranged in an inclusion hierarchy together with hardware components such as the nodes they reside on. In addition, it is possible to define Cartesian process topologies to represent network or virtual topologies. Profiles can be visually explored using the Cube browser. Compared to its predecessor CUBE-3, CUBE-4 files have been optimized for fast writing by storing the metric values in a binary file.

Time-Series Profiles

Time-series profiles are like normal CUBE-4 profiles except that they maintain a separate sub-tree in the call tree for each iteration of the time-step loop. This allows the user to distinguish individual iterations and to observe the evolution of the performance behaviour along the time axis. Time-series profiles are created by annotating the body of the time-step loop with special instrumentation, which tells Score-P when an iteration ends and when a new one begins. They can be analysed using the normal Cube display. A future version of Cube (to be completed after this project ends) will provide special iteration diagrams that offer an easy way to judge how the performance changes over time. To avoid that profiling data exceeds the available buffer space, future versions of Score-P will support the dynamic compression of time-series profile data using an online clustering algorithm [15].

Event Traces

Event traces include all events of an application run that are of interest for later examination, together with the time they occurred and a number of event-type-specific attributes. Typical events are entering and leaving of functions or sending and receiving of messages. Event traces produced by Score-P are stored in the Open Trace Format Version 2 (OTF-2), a new trace format whose design is based on the experiences with the two predecessor formats OTF [10] and EPILOG [11], the former native formats of Vampir and Scalasca, respectively. The main characteristics of OTF-2 are similar to other record-based parallel event trace formats. It contains events and definitions and distributes data storage over multiple files. In addition, it is more memory efficient, offering the possibility to achieve measurements with less perturbation due to memory buffer flushes. In contrast to OTF, the event traces are stored in a binary format, which reduces the size of the trace files without the need for a separate compression step. OFT-2 traces are the foundation for further analysis. Vampir can display OTF-2 traces visually using different kinds of displays, including a zoomable timeline. The Scalasca trace analyser identifies wait states and their root causes, producing a CUBE-4 file that provides a higher-level view of the application performance data. This is typically recommended to get an idea of key performance issues before visually exploring the traces directly using a trace browser. Moreover, there is a conversion tool which can export OTF2 traces in the Paraver format so that they can be analysed using Paraver (visual exploration) and Dimemas (what-if analysis).

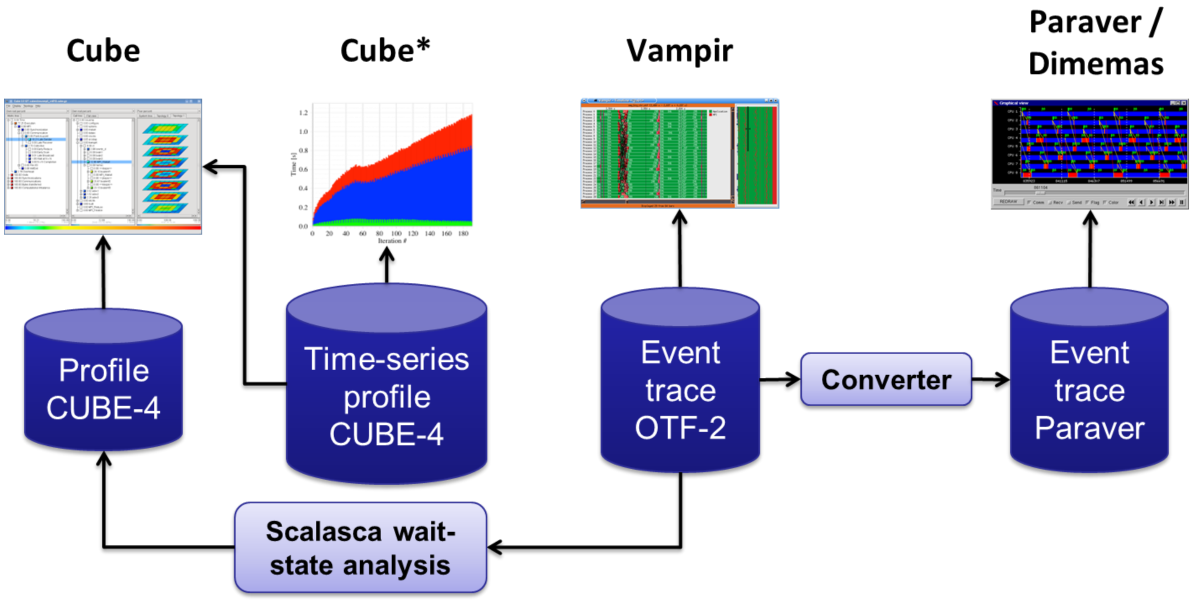

Figure 5: Performance data types supported by Score-P and the tools that can be used to analyse them. The * next to the second mentioning of Cube indicates a display type that will be provided in a future version.

1.3.1.3.3 Overhead Minimisation

Another important aspect is the quality of the collected performance data in terms of intrusion and their size. To keep both intrusion and data size small, the Score-P measurement system offers a systematic approach of expanding the level of detail while at the same time narrowing the measurement focus:

1. Generate a summary profile with generous instrumentation while measuring the overhead. If the overhead is too large (> 10%), reduce instrumentation, for example, through the application of filter lists. Measure overhead again and iterate until the overhead is satisfactory.

2. Generate a new summary profile with acceptable overhead. This provides an overview of the performance behaviour across the entire execution time and allows the identification of suspicious call paths and processes.
3. Generate a time-series profile, which provides a separate summary profile for every iteration of the time-step loop. This shows to which degree the performance behaviour changes as the simulation progresses and allows the identification of iterations that warrant deeper analysis. A semantic compression algorithm will ensure that the size of time-series profiles stays within reasonable limits.
4. For the identified iterations, generate event traces. Event traces provide the highest level of detail and offer a number of interesting analysis options including automatic wait-state analysis and visual exploration.

1.3.2 The HOPSA Performance Tools

This section introduces the various HOPSA performance tools.

1.3.2.1 Dimemas

Dimemas [12] is a performance prediction tool for message-passing programs. The Dimemas simulator reconstructs the time behaviour of a parallel application using as input an event trace that captures the time resource demands (CPU and network) of a parallel application. The target machine is modeled by a reduced set of key factors influencing the performance that model linear components like the point-to-point transfer time as well as non-linear factors like resources contention or synchronisation. Using a simple model, Dimemas allows performing parametric studies (see Figure 6) in a very short time frame. The supported target architecture is a cloud of parallel machines, each one with multiple nodes and multiples CPUs per node allowing the evaluation of a very wide range of alternatives, despite the most common environment is a computing cluster (see Figure 7). Dimemas can generate as part of its output a Paraver trace file, enabling the user to conveniently examine the simulated run and understand the application behaviour.

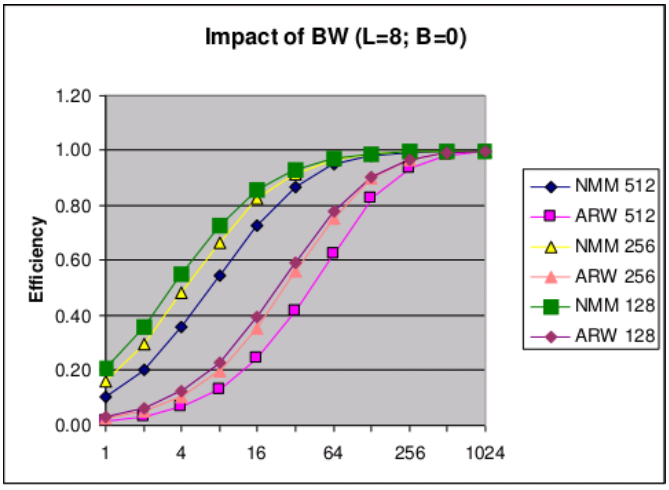

Figure 6: Dimemas parametric study example.

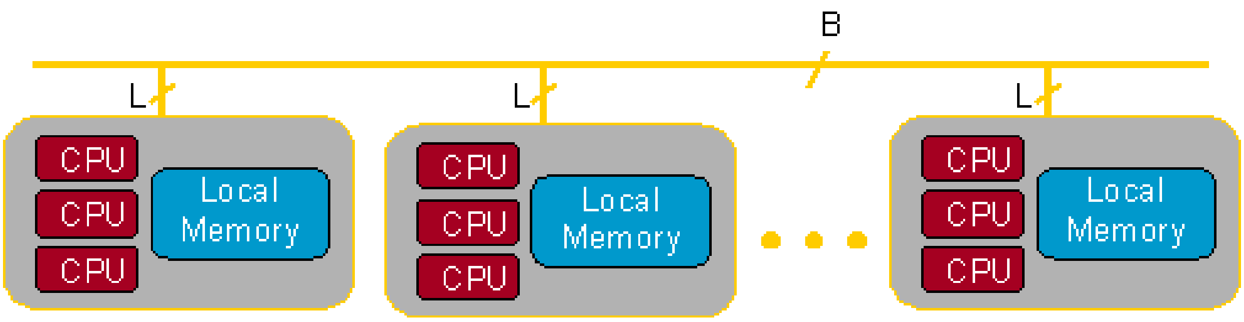

Figure 7: Dimemas's architectural model.

Typical questions Dimemas helps to answer

- How would my application perform in a future system?
- Can increasing the network bandwidth improve the application performance?
- Would my application benefit from asynchronous communication?
- Is my application limited by the network or by serialisation and dependency chains in my code?
- What is the sensitivity of my application to different system parameters?
- What would be the impact of accelerating specific regions of my code?

Supported programming models

Dimemas targets message-passing programming models as well as task-oriented programs. The current instrumentation allows using Dimemas with MPI or MPI+OmpSs applications.

Input sources

The analyses offered by Dimemas rely on event traces in the Dimemas format generated by the Paraver to Dimemas trace translator prv2dim or directly by the runtime measurement system Extrae.

Simulations with Dimemas

Dimemas enables two main types of analyses: **"what-if" studies** to simulate how an application would perform in an hypothetical scenario (e.g., would reducing the network latency by a factor of two have more impact than moving to a CPU twice as fast?), and **parametric studies** running multiple simulations to analyse the sensitivity of the application to the system parameters (e.g., to plot the execution time when varying the network bandwidth from 100Mb/s to 16Gb/s).

A first step to use Dimemas is to translate a Paraver trace file to the Dimemas trace format. It can be the full application execution as well as a representative region with a reduced number of iterations. Then the user specifies through the Dimemas GUI the application trace file to use as input, the architectural parameters of the target machine (such as the latencies and bandwidths for inter-node and intra-node communications, number of network devices...) and the mapping of the tasks onto the different nodes. This information is saved in a file that will be passed as parameter to the simulator. Typically, the user will add an option to generate as output a Paraver trace file that can be later compared with the original run using the Paraver tool.

Instrumentation

Dimemas traces are typically translated from a Paraver trace, but they can also be directly generated by the Extrae tool. Refer to the Paraver section for further details on the available instrumentation mechanisms.

License model

Dimemas is available open source under the terms of the GNU Lesser General Public License (LGPL) v2.1.

Further documentation

- Website: www.bsc.es/dimemas
- Support email: tools@bsc.es

1.3.2.2 Paraver

Paraver [1] is a very flexible data browser that is part of the CEPBA-Tools toolkit. Its analysis power is based on two main pillars. First, its trace format has no semantics; extending the tool to support new performance data or new programming models requires no changes to the visualiser – just capturing such data in a Paraver trace. The second pillar is that the metrics are not hardwired in the tool but can be programmed. To compute them, the tool offers a large set of time functions, a filter module, and a mechanism to combine two timelines. This approach allows displaying a huge number of metrics with the available data. To capture the expert's knowledge, any view or set of views can be saved as a Paraver configuration file. After that, re-computing the view with new data is as simple as loading the saved file. The tool has been demonstrated to be very useful for performance analysis studies, giving much more details about the application behaviour than most other performance tools.

Performance information in Paraver is presented with two main displays that provide qualitatively different types of information. The timeline display represents the behaviour of the application along time and processes, in a way that easily conveys to the user a general understanding of the application behaviour and simple identification of phases and patterns. The statistics display provides numerical analysis of the data that can be applied to any user-selected region, helping to draw conclusions on where and how to focus the optimisation effort. See Figures 4 and 5 for an example of Paraver's main displays.

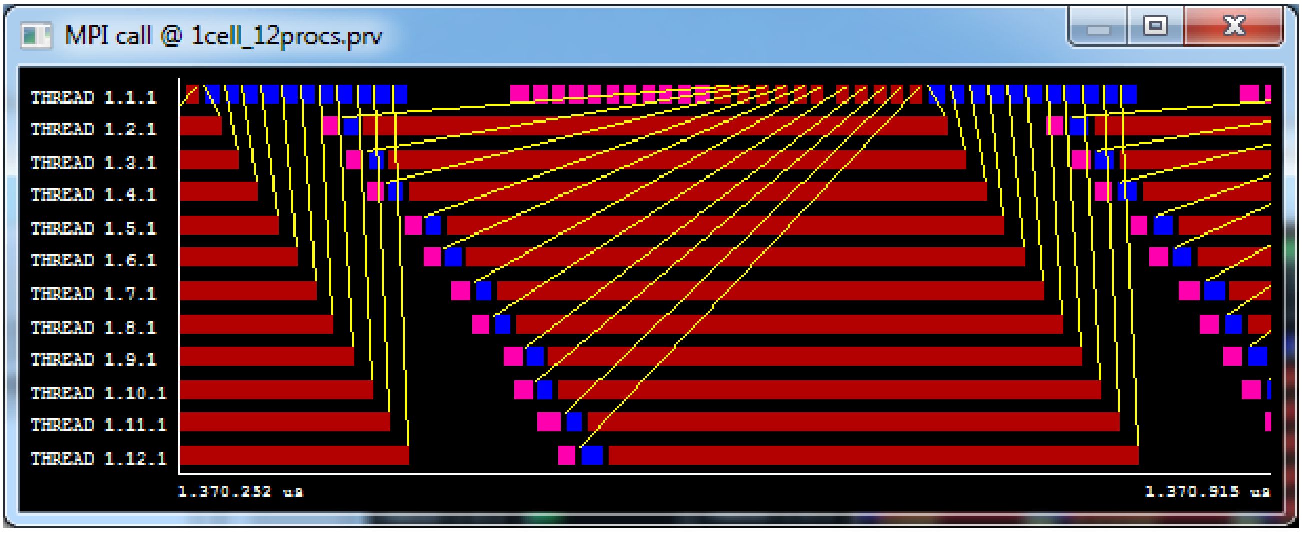

Figure 8: Paraver timeline display.

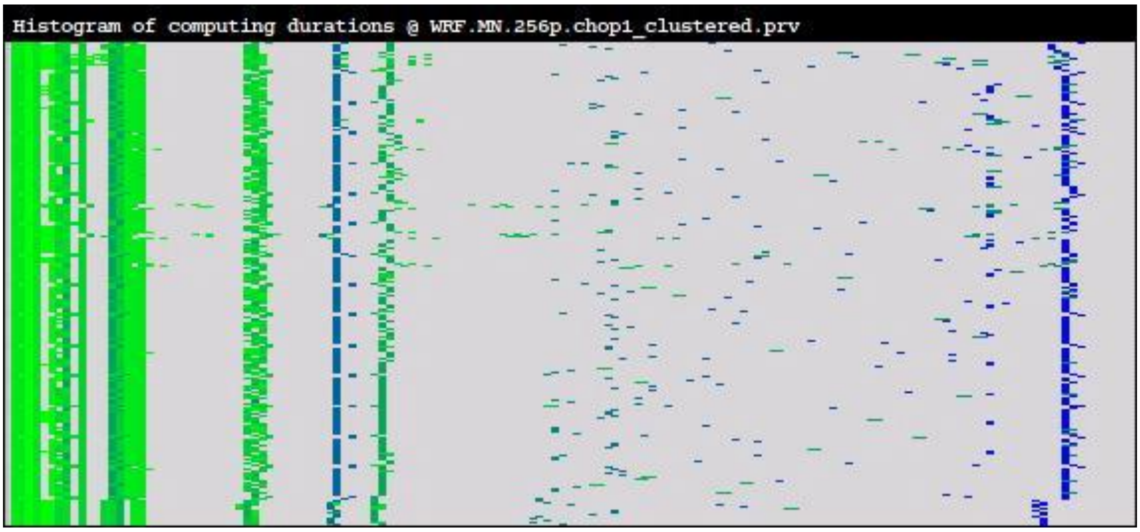

Figure 9: Paraver histogram display.

Typical questions Paraver helps to answer

- What is the parallelisation efficiency and the performance of communication?
- What are the differences that can be observed between two different executions?
- Does the behaviour of the application change over time?
- Are performance or workload variations the cause of load imbalances in computation?
- Which performance issues do the microprocessor's hardware counters reflect?

Supported programming models

Paraver is not tied to any programming model as long as the model used can be mapped onto the three levels of parallelism expressed in the Paraver trace. An example of a two-level parallelism would be hybrid MPI + OpenMP applications. The runtime measurement system Extrae that generates Paraver traces currently supports the programming interfaces MPI, OpenMP, pthreads, OmpSs and CUDA.

Input sources

The analyses offered by Paraver rely on event traces in the Paraver format generated by the runtime measurement system Extrae.

Performance analyses

The analysis with Paraver typically starts from a set of pre-conceived configuration files that are available to the user. Each configuration describes a certain view of the performance data, such as the time distribution of functions, MPI primitives or parallel loops called, the value of a given performance metric (e.g., cache misses, floating-point operations, or network bandwidth), and statistics (e.g., profile of the MPI calls – average duration, percentage of time, number of invocations – histogram of the computation regions duration, correlation between duration and instructions). The tool provides an extensive initial set of configurations that cover those parameters that are usually of highest interest to study, and applying them is as easy as loading a file.

The typical analysis cycle then consists of loading one or more views, zooming into the details of specific processes or code phases, computing histograms and profiles, classifying the data, identifying performance issues, and correlating where these issues happen through the execution, in a process that goes back and forth from the *timelines* to the *statistics.*

The tool offers a very flexible way to combine multiple views, so as to generate new representations of the data and more complex derived metrics. Once a desired view is obtained, it can be stored in a configuration file to apply it again to the same trace or to a different one. Sharing the traces and the corresponding configuration files allows views of the trace and the information obtained to be easily shared.

Instrumentation

Extrae enables four main modes of code instrumentation: manual source-code modification using the Extrae API, library interposition through static linking or dynamic pre-loading, and binary memory image modification at load time using the Dyninst instrumentor. OpenMP constructs are instrumented by wrapping the runtime calls through the dynamic interposition mechanisms, and MPI calls are intercepted through the PMPI profiling interface.

License model

Paraver is available as open source under the terms of the GNU Lesser General Public License (LGPL) v2.1.

Further documentation

- Website: www.bsc.es/paraver
- Support email: tools@bsc.es
- Built-in tutorial (Help →Tutorials)

1.3.2.3 Scalasca

Scalasca [2] is a free software tool that supports the performance optimisation of parallel programs by measuring and analysing their runtime behaviour. The tool has been specifically designed for use on large-scale systems including IBM Blue Gene and Cray XE, but is also well suited for small- and medium-scale HPC platforms. The analysis identifies potential performance bottlenecks – in particular those concerning communication and synchronization – and offers guidance in exploring their causes.

The user of Scalasca can choose between two different analysis modes: (i) performance overview on the call-path level via profiling and (ii) the analysis of wait-state formation via event tracing. Wait states often occur in the wake of load imbalance and are serious obstacles to achieving satisfactory performance. Performance-analysis results are presented to the user in an interactive explorer called Cube (Figure 10) that allows the investigation of the performance behaviour on different levels of granularity along the dimensions performance problem, call path, and process. The software has been installed at numerous sites in the world and has been successfully used to optimise academic and industrial simulation codes.

Typical questions Scalasca helps to answer

- Which call-paths in my program consume most of the time?
- Why is the time spent in communication or synchronisation higher than expected?
- Does my program suffer from load imbalance and why?

Supported programming models

Scalasca supports applications based on the programming interfaces MPI and OpenMP, including hybrid applications based on a combination of the two. Support for CUDA and StarSs is in progress.

Input sources

The analyses offered by Scalasca rest on profiles in the CUBE-4 format and event traces in the OTF-2 format. Both performance data formats can be generated using Score-P.

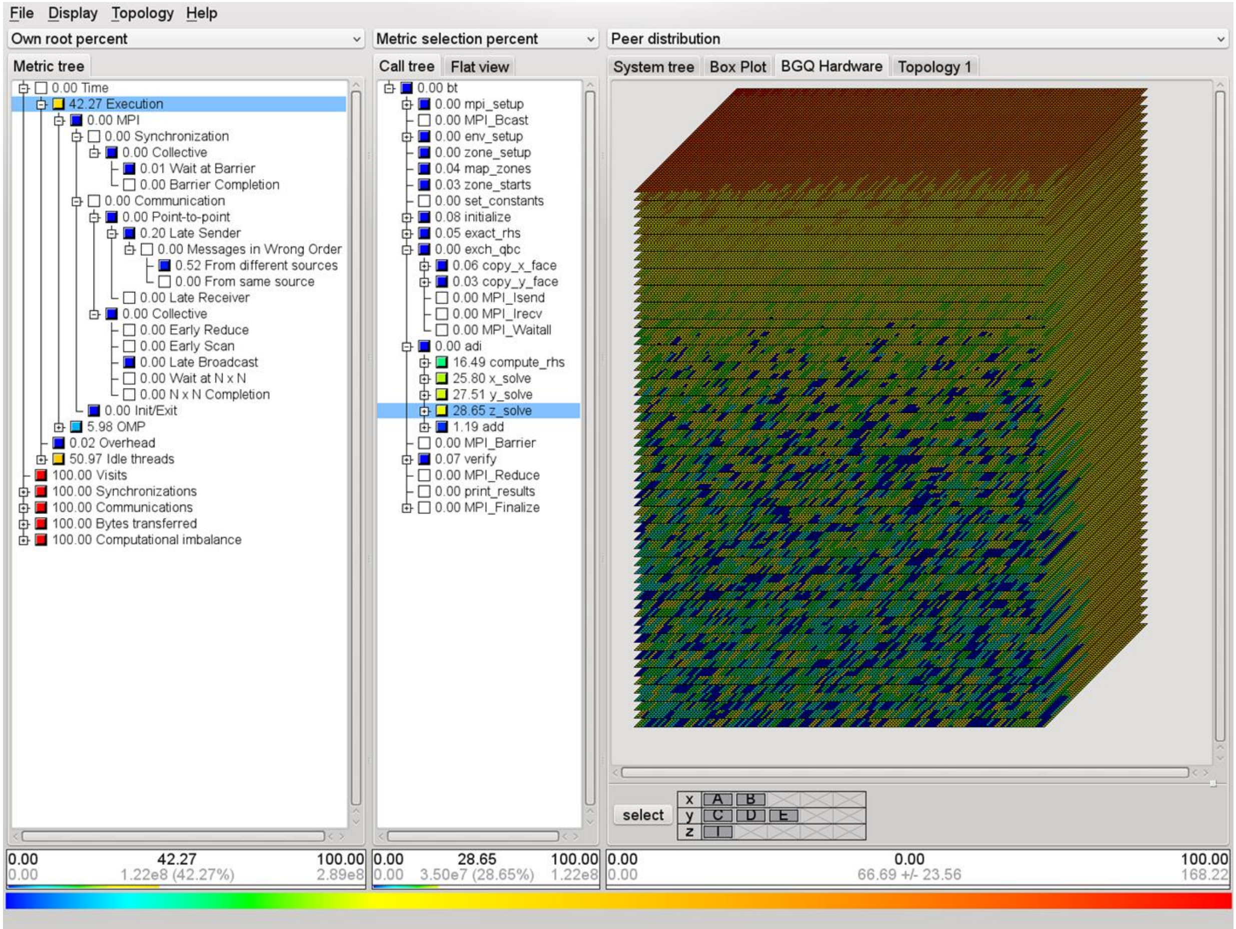

Figure 10: : Interactive exploration of performance behaviour in Scalasca along the dimensions performance metric (left), call tree (middle), and process topology (right). The picture shows a Scalasca trace analysis of NAS BT-MZ executed with 524,288 threads.

Performance analyses

- Summary profile: The summary profile can be used to identify the most resource-intensive call paths or processes. It tells how the execution time and other performance metrics including hardware counters are distributed across the call tree and the set of processes or threads.
- Time-series profile: The time-series profile can be used to analyse how the performance behaviour evolves over time – even if the application runs for a longer period. Essentially, a time-series profile provides a separate summary profile for every iteration of the main loop.
- Wait state analysis: This analysis extracts from event traces the location of wait states. Detected instances are both classified and quantified. High amounts of wait states usually indicate load or communication imbalance.
- Delay analysis: The delay analysis extends the wait-state analysis in that it identifies the root causes of wait states. It traces wait states back to the call paths causing them and determines the amount of waiting time a particular call path is responsible for. It considers both direct wait states and those created via propagation.
- Critical-path analysis: This trace-based analysis determines the effect of imbalance on program runtime. It calculates a set of compact performance indicators that allow users to evaluate load balance, identify performance bottlenecks, and determine the performance impact of load imbalance at first glance. The analysis is applicable to both SPMD and MPMD-style programs.

Instrumentation

User code is instrumented in source code (automatically by compiler or PDT instrumentor, or manually with macros or pragmas). OpenMP constructs are instrumented in source code (automatically by the OPARI2 instrumentation tool). MPI calls are intercepted automatically through library interposition.

License model

The software is available under the New BSD license.

Further documentation

- Website: www.scalasca.org
- Support email: scalasca@fz-juelich.de

1.3.2.4 ThreadSpotter

ThreadSpotter [3] is a commercial tool that will help programmers optimise their programs with respect to architectural bottlenecks such as cache size and memory system bandwidth and point out inefficient communication modes between threads. Its scope is a single process, including both single-threaded as well as multi-threaded applications.

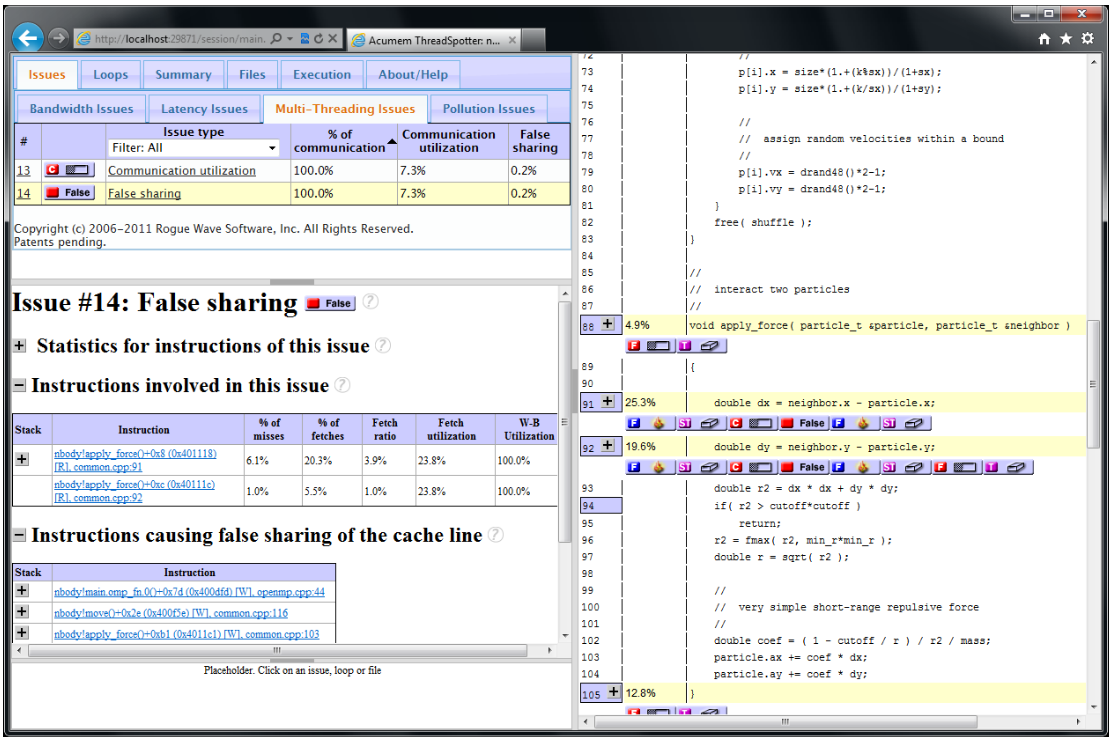

Figure 11: Highlighting a “false sharing” situation. Top left part contains lists of problems. Lower left contains details, and annotated source code is to the right.

Some programming styles will exercise the memory system in suboptimal ways that can reduce performance drastically. Examples of these are failure to observe or exploit locality properties in code or data. Inappropriate communication through shared memory between threads may cause the coherence traffic to become a bottleneck.

ThreadSpotter explains the inefficiencies of observed memory access patterns on a high level in a graphical user interface (Figure 11) and provides pointers to suggestions to optimise the code. It offers deep explanations on hardware level to back up the suggestions, educating the user as he uses the tool.

Typical questions ThreadSpotter helps to answer

• How does my program abuse the memory system and what can I do about it?

• Do the threads of my program exchange data with each other in an inefficient way?

• When adjusting my program, are the changes actually helping to minimise the footprint of the application?

Supported programming models

ThreadSpotter focuses on a single sequential or multi-threaded process at a time. In distributed environments the user may collect independent information from multiple ranks and investigate the behaviour of these separately. It operates completely on object code level and no particular requirements are expressed on how programs are written, compiled or linked. It supports all threading paradigms.

Input sources

ThreadSpotter includes its own data collection agent, which monitors the application's behaviour as it executes a representative workload. It sparsely collects platform-independent access patterns from the application and stores this data in a small file. This data is then analysed to produce a report. In distributed environments, each rank produces its own data and each one is the source of a separate report, allowing the user to explore runtime behaviour from different parts of the cluster.

Performance analyses

- Overall performance verdict – quickly get a statement on the relative performance and existing problems.
- Summary – graphically and numerically see key metrics from the application as a whole. This can help the user further comparing behaviour between generations of his program on a high level.
- Advice:
  - Spatial locality problems – Explore the high level reasons why the program may not use all of the data that is brought to the cache
  - Temporal locality problems – Find opportunities to reorganise algorithms to use data in the cache more times
  - Prefetch analysis and cache pollution – Instruct the processor to bypass the cache where it makes sense.
  - Multi-threading problems – Find traces of inefficient patterns of data sharing between threads, such as false sharing.
  - Bandwidth and latency: Identify areas in the code where prefetching does not work. Identify program areas taxing the memory bandwidth the hardest.
- Statistics
  - Fundamental cache and bandwidth related metrics: fetch ratio, miss ratio, write-back ratio.
  - Higher level metrics: fetch utilisation, write-back utilisation and communication utilisation.
  - Metrics can be decomposed along different dimensions:
    - Program scope: global, function, loop, instruction
    - Per thread
    - Per type (capacity, coherence, compulsory, …)
- What-if analysis – Perform different experiments from one single fingerprint file:
  - Learn which optimisations are appropriate for different architectures.
  - See what the effect will be of binding threads differently.

Instrumentation

ThreadSpotter uses dynamic binary instrumentation. Thus, it operates on unmodified, production optimised binaries. If debug information is available for the binaries, then ThreadSpotter will be able to point to source code.

License model

The software is available under a commercial license.

Further documentation

- Website: http://www.roguewave.com/products/threadspotter.aspx
- Support email: threadspottersupport@roguewave.com
- Manual: http://www.roguewave.com/support/product-documentation/threadspotter.aspx

1.3.2.5 Vampir

Vampir [4] is a graphical analysis framework that provides a large set of different chart representations of event-based performance data. These graphical displays, including timelines and statistics, can be used by developers to obtain a better understanding of their parallel program's inner working and to subsequently optimise it. See Figure 12 for an impression of the Vampir GUI.

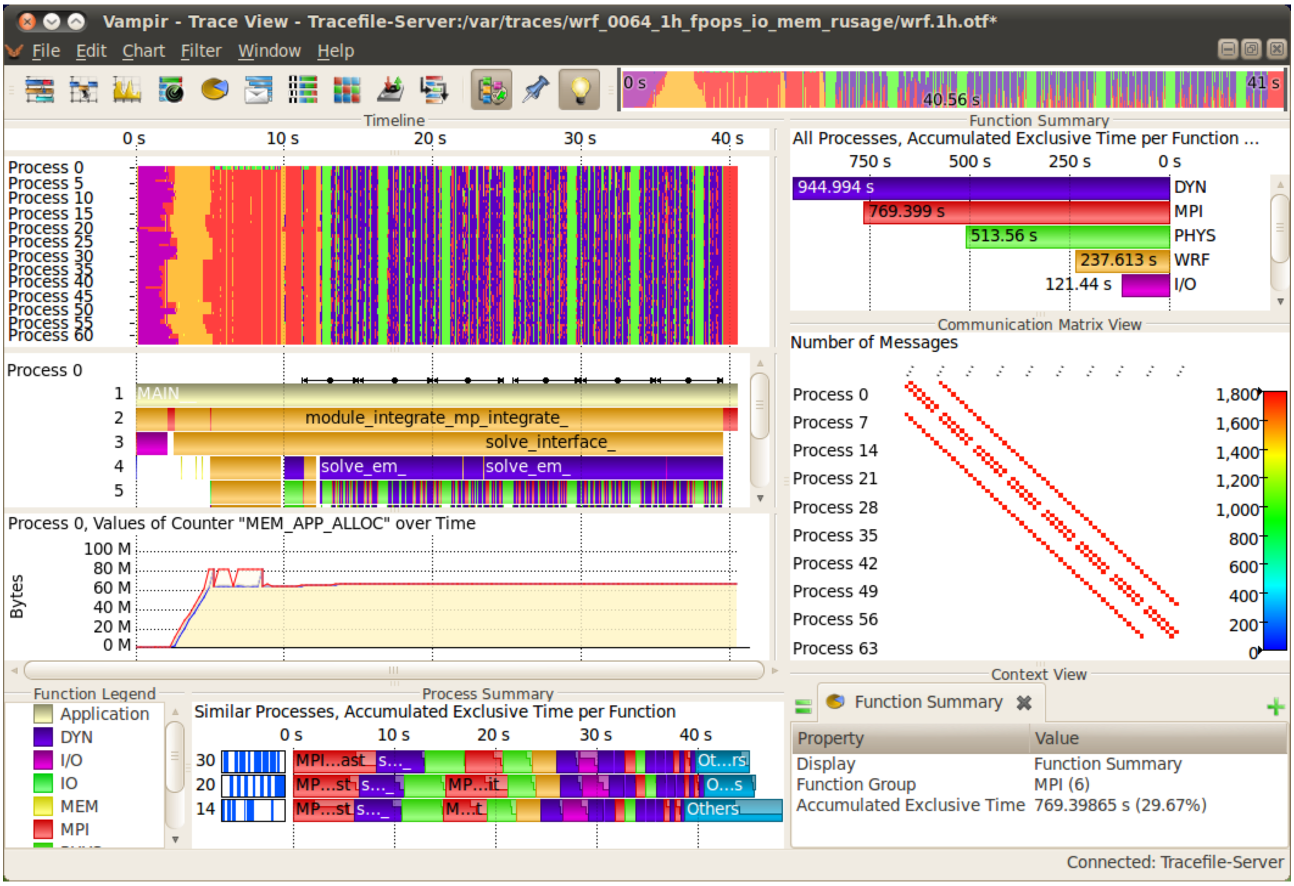

Figure 12: Vampir GUI

Vampir is designed to be an intuitive tool, with a GUI that enables developers to quickly display program behavior at any level of detail. Different timeline displays show application activities and communication along a time axis, which can be zoomed and scrolled. Statistical displays provide quantitative results for the currently selected time interval. Powerful zooming and scrolling along the timeline and process/thread axis allows pinpointing the causes of performance problems. All displays have context-sensitive menus, which provide additional information and customisation options. Extensive filtering capabilities for processes, functions, messages or collective operations help to narrow down the information to the interesting spots. Vampir is based on Qt and is available for all major workstation operation systems as well as on most parallel production systems. The parallel version of Vampir, VampirServer, provides fast interactive analysis of ultra large data volumes.

Typical questions Vampir helps to answer

- What happens in my application execution during a given time in a given process or thread?
- How do the communication patterns of my application execute on a real system?
- Are there any imbalances in computation, I/O or memory usage and how do they affect the parallel execution of my application?

Supported programming models

Vampir supports applications based on the programming interfaces MPI and OpenMP, including hybrid applications based on a combination of the two. Furthermore Vampir also analyses hardware accelerated applications using CUDA and/or OpenCL.

Input sources

The analyses offered by Vampir rest on event traces in the OTF format generated by the runtime measurement system VampirTrace. The latest Vampir release also supports OTF2 that is generated by Score-P.

Performance analysis via timeline displays

The timeline displays show the sequence of recorded events on a horizontal time axis that can be zoomed to any level of detail. They allow an in-depth analysis of the dynamic behavior of an application. There are several types of timeline displays.

- Master timeline: This display shows the processes of the parallel program on the vertical axis. Point-to-point messages, global communication, as well as I/O operations are displayed as arrows. This allows for a very detailed analysis of the parallel program flow including communication patterns, load imbalances, and I/O bottlenecks.
- Process timeline: This display focuses on a single process only. Here, the vertical axis shows the sequence of events on their respective call-stack levels, allowing a detailed analysis of function calls.
- Counter data timeline: This chart displays selected performance counters for processes aligned to the master timeline or the process timelines. This is useful to locate anomalies indicating performance problems.

Performance analysis via statistical displays

The statistical displays are provided in addition to the timeline displays. They show summarised information according to the currently selected time interval in the timeline displays. This is the most interesting advantage over pure profiling data because it allows specific statistics to be shown for selected parts of an application, e.g., initialisation or finalisation, or individual iterations without initialisation and finalisation. Different statistical displays provide information about various program aspects, such as execution times of functions or groups, the function call tree, point-to-point messages, as well as I/O events.

Instrumentation

Application code can be instrumented by the compiler or with source-code modification (automatically by the PDT instrumentor, or manually using the VampirTrace/Score-P user API). OpenMP constructs can be instrumented by the OPARI tool using automatic source-to-source instrumentation. MPI calls are intercepted automatically through library interposition.

License model

Vampir is a commercial product distributed by GWT-TUD GmbH. For evaluation, a free demo version is available on the website.

Further documentation

- Website: www.vampir.eu
- Support email: service@vampir.eu
- Vampir manual: installation directory under $VAMPIR_ROOT/doc/vampir-manual.pdf

1.3.3 Performance Tools Success Story: Scalasca delay analysis of the CESM sea ice model

In this case study, we applied Scalasca to a trace experiment of the Community Earth System Model (CESM) **Error! Reference source not found.**climate code, where the delay analysis illustrates problems in the domain decomposition and communication pattern. The measured configuration used a 1º dipole grid of the Earth and a Cartesian grid decomposition on 2048 MPI processes. This configuration suffers from severe load imbalances. The first of these imbalances is a result of the domain decomposition in the form of a uniform, high-resolution grid of the Earth. Since the sea ice model obviously only applies to ocean regions, processes assigned exclusively to land regions do not participate in computation or communication at all. Also, processes assigned to regions where land and ocean overlap have less workload than processes assigned to all-ocean regions. A second load imbalance exists between processes assigned to open ocean regions and those assigned to sea ice (i.e., polar) regions, which have a significantly higher workload than the others. As a result of these imbalances, many processes suffer late-sender wait states in the point-to-point nearest-neighbour MPI data exchange following the computation.

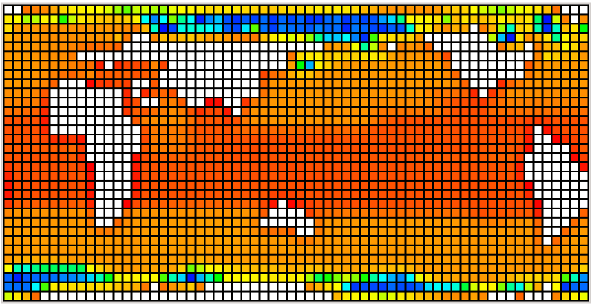

Figure 13: Total late-sender wait states

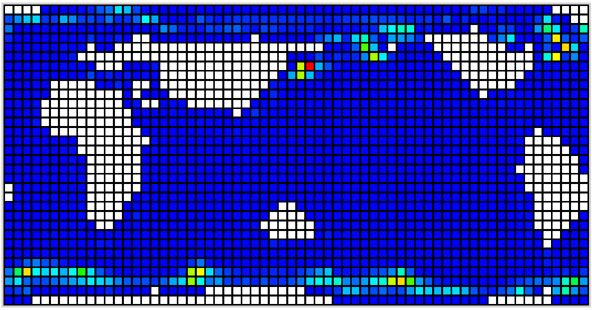

Figure 14: Delay costs

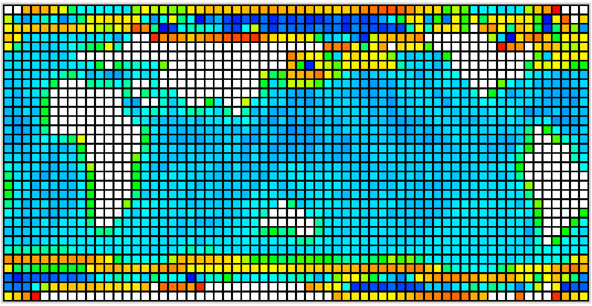

Figure 15: Direct late-sender wait states

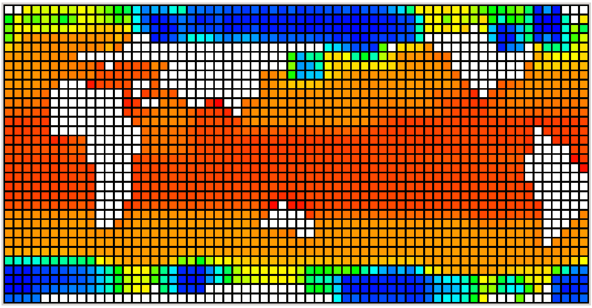

Figure 16: Indirect late-sender wait states

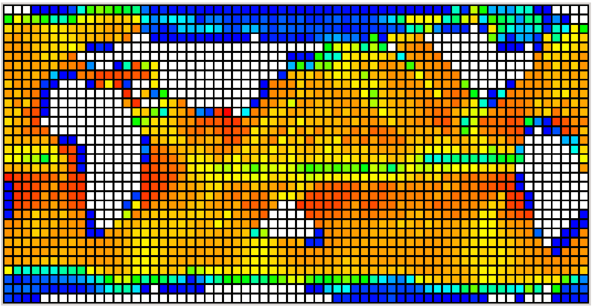

Figure 17: Propagating late-sender wait states

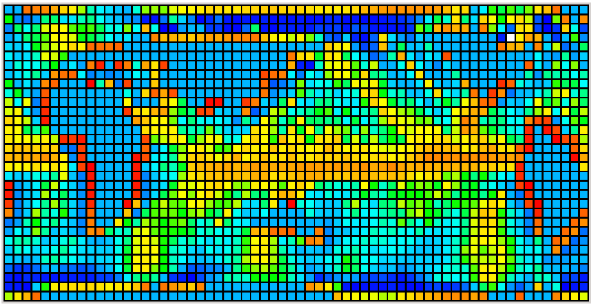

Figure 18: Terminal late-sender wait states

Figure 13 through Figure 18 visualize the performance metrics determined by Scalasca in the 64x32 2D process grid used by the sea ice simulation. To allow an easy comparison between the different performance metrics, the figures only show the topology (right-most) panel of the Cube result browser. Red colours denote high values, blue colours denote low values. On the top left, Figure 13 illustrates the distribution of late-sender waiting time across the computational grid. Essentially, all processes assigned to open ocean regions incur wait states. Note that processes assigned to land-only grid points do not participate in the data exchange at all, and therefore do not incur late-sender wait states.

The delay analysis now allows us to understand the original causes of the wait states and the wait-state propagation pattern. It pinpoints source code and process locations where excess computation time leads to wait states at subsequent synchronization points. Delay costs denote the total amount of waiting time caused by a particular source code / process location. As shown by the distribution of delay costs in Figure 14, the delays responsible for the late-sender wait states in the sea ice model are located on the border between sea ice and open ocean processes. Processes assigned to an area north of Japan particularly stand out: further investigation revealed that this is likely the result of the complex topography around the Kamchatka peninsula, which makes the sea ice computations more expensive for this area. The delay analysis easily pinpointed this area as a hotspot of the simulation.

Because of the nearest neighbour communication pattern, the delays on the open sea/ice border only affect a small number of processes directly. However, the wait states they cause propagate to processes further away in subsequent iterations, so that they eventually affect the entire process grid. The classification of wait states into direct versus indirect and propagating versus terminal wait states performed by the delay analysis allows us to observe the propagation effect directly. First of all, the distribution of indirect wait states in Figure 16 confirms that wait states on most processes are indeed produced by propagation. Direct wait states, which are the wait states directly caused by excess computation, are only located on processes immediately surrounding the sea ice regions in the process grid, as can be seen in Figure 15. Propagating wait states are wait states which themselves delay subsequent communication operations, thereby causing more wait states on other processes. From the distribution of propagating wait states in the sea ice simulation shown in Figure 17, we can see that wait states propagate both southwards from the northern polar region as well as northwards from the southern polar region towards the equatorial region of the domain. Terminal wait states (Figure 18), which do not affect subsequent communications, are primarily located on processes assigned to the equatorial ocean regions and on shores, where the propagation chain ends. Hence, with the help of the delay analysis, we are able to gain detailed insight into the causes and propagation pattern of wait states.

It should be noted that the experiments were run at a larger scale than typically used in production runs. The limited scalability of the regular domain decomposition for the sea ice calculation is known, and is expected to produce significant imbalance for large-scale runs. However, these large-scale experiments magnify domain decomposition and communication issues early, and therefore provide useful performance insights. Currently, the developers investigate better load balancing schemes, in particular space-filling curves, to achieve better balance at large scale.

1.3.4 Integration among 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. Figure 19 gives an overview of the already implemented and envisioned tool interactions within the HOPSA tool set.

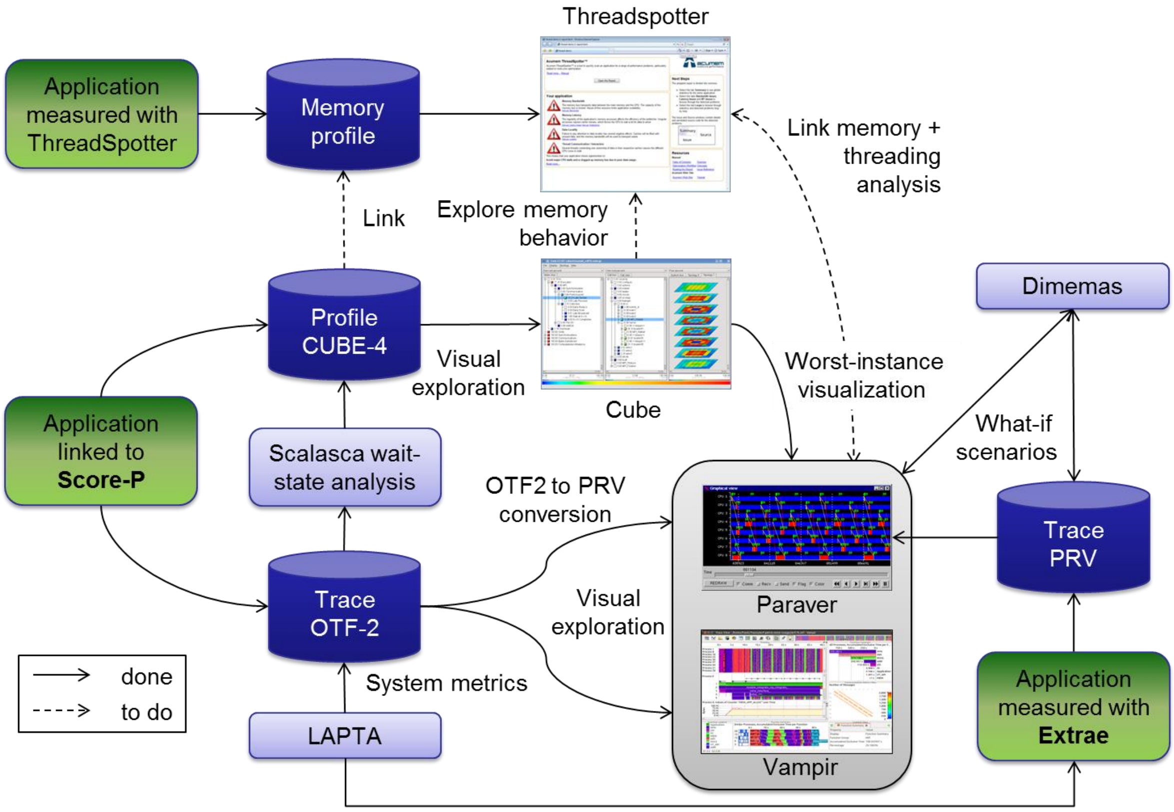

Figure 19: HOPSA Performance Tool Integration

For example, with its automatic trace analysis, Scalasca locates call paths 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 Figure 20).

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.

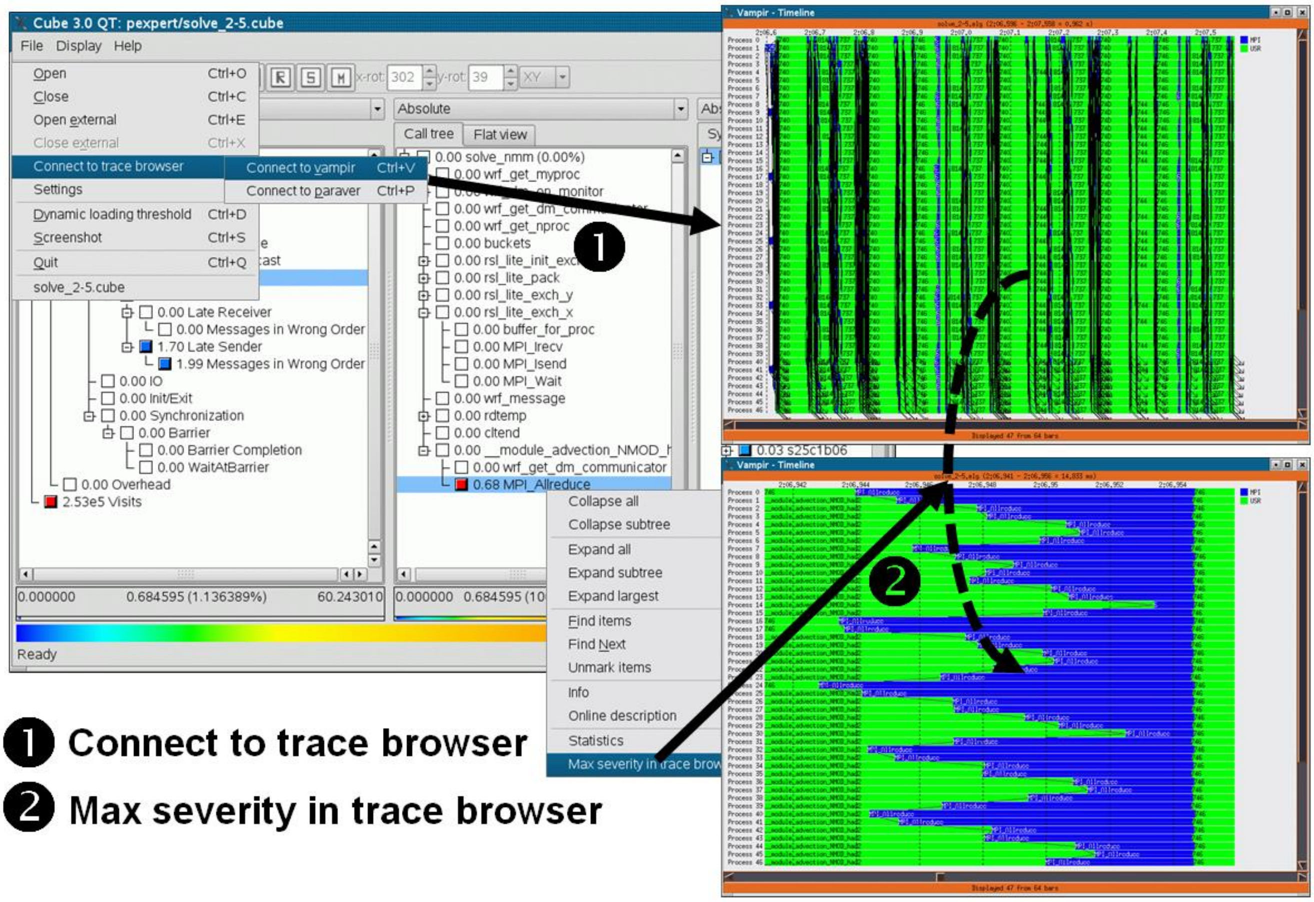

Figure 20: Scalasca →Vampir or Paraver Trace browser integration. In a 1st step, when the user requests to connect to a trace browser, the selected visualizer is automatically started and the event trace, which was previously the basis of Scalasca's trace analysis, is loaded. Now, in a 2nd step, the 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.

1.3.5 Integration of System Data and Performance Analysis Tools

The Russian ClustrX.Watch management software 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 system information (the granularity will depend on the overhead to obtain the data) and to analyze them with respect to the system-wide performance.

The Russian LAPTA system data analysis and management software 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 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.

The system offers two different ways to access to the collected data:

Historic information is stored with a given granularity for all the sensors and all the IP (nodes) on the system. The initial granularity was very coarse (one minute) and did not seem useful for the population of application trace files because there can be many different program phases in a one minute interval. On the other hand, the circular buffer provides historical information with fine-grained detail (coarser or equal to 1 second depending on the sensor) for the last minutes (300 measurements).

Streamed information can be requested for any range of sensors and IPs. The interface provides at least a value every 10 seconds unless there is a change greater than a 10%. Currently, the finest available granularity is 1 second.

Both mechanisms use a connection through an HTTP protocol that in the case of the streamed data has to be refreshed periodically or dies after 5 minutes. We evaluated both alternatives to see their potential and identified possible drawbacks.

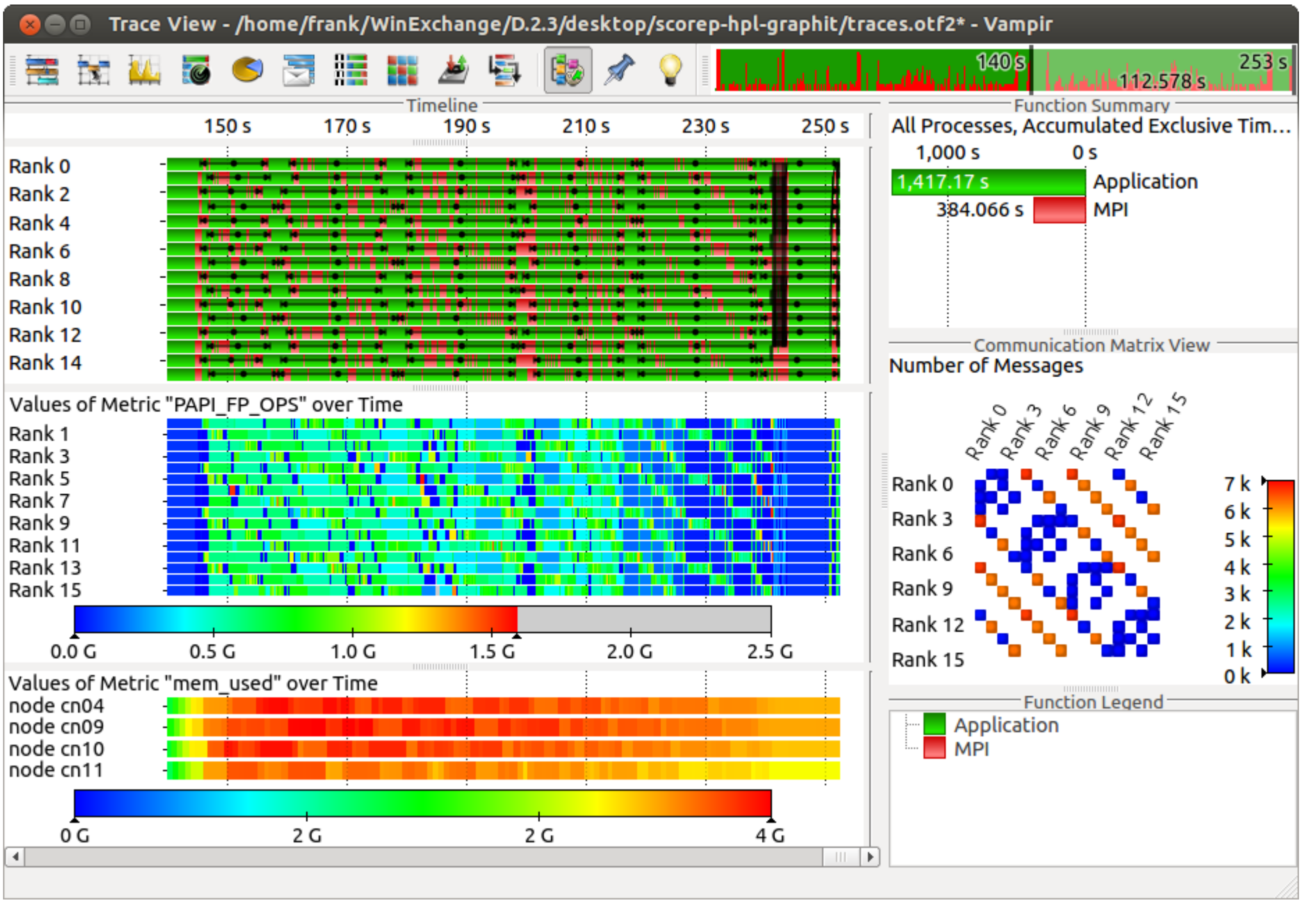

Figure 21: Vampir's Trace Visualization of the benchmark code HPL including the HOPSA node level metric "mem_used" (used memory) in the Performance Radar

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 (Figure 21) one can see rising floating point operations (second timeline) resulting in a higher memory consumption per node (third timeline).

Another powerful approach is to record InfiniBand communication between nodes. This allows the identification of possible communication bottlenecks due to problems of the InfiniBand network. Therefor a MPI implementation of a cannon algorithm has been traced along with the measurement of InfiniBand performance data counter. Figure 22 shows the MPI communication next to the InfiniBand data metrics per node. The correlation between both data sets becomes clearly visible.

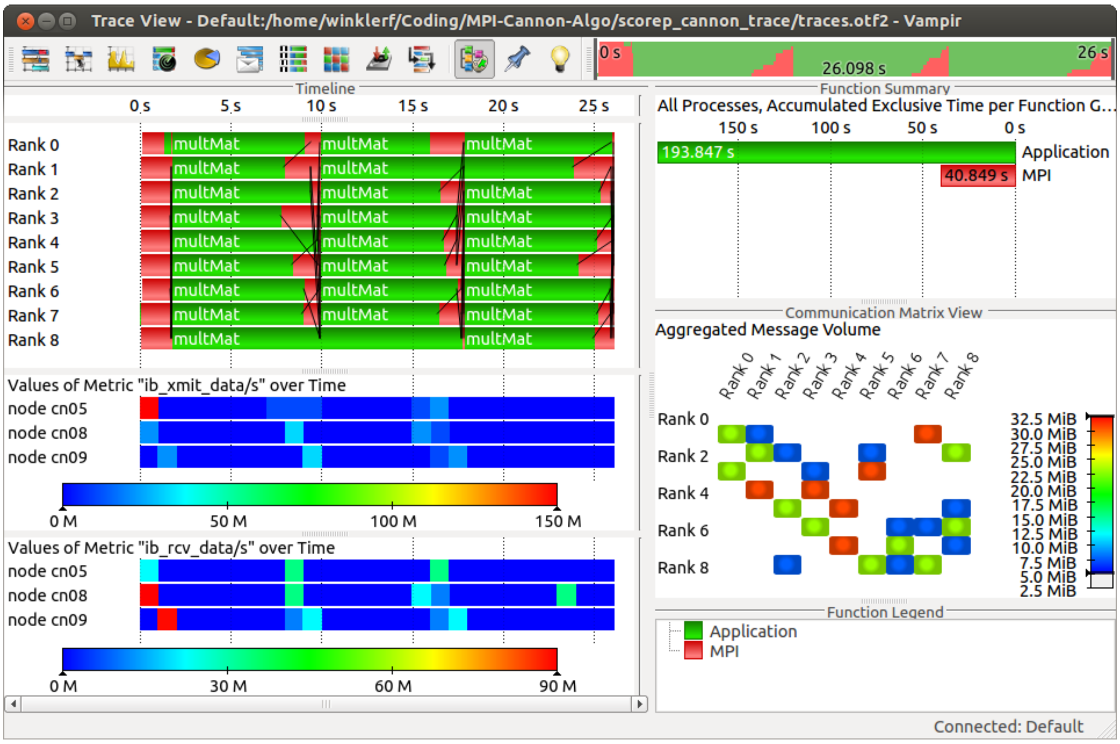

Figure 22: Vampir's Trace Visualization of a cannon algorithm showing the MPI communication next to the InfiniBand data counter

As a use case, this integration allows the user to analyze how the application utilizes network hardware of each node or how shared usage of network resources affects the application execution. Currently the sensor values are available in 1 second granularity for the last 500 seconds and 1 minute granularity before that.

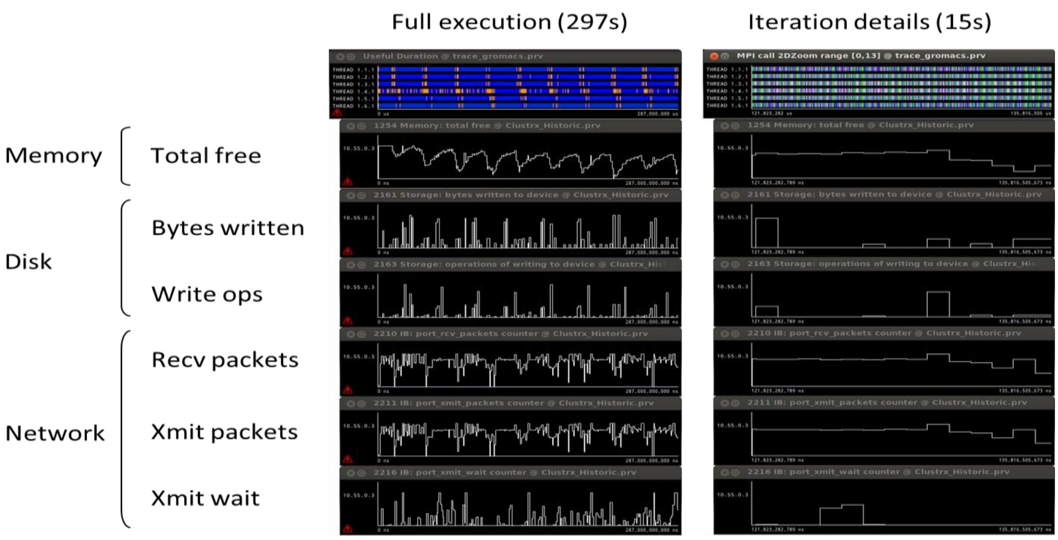

Figure 23: Paraver screendumps showing aligned system and application data.

The Paraver team experimented with Gromacs, a popular production code in life sciences, trying to correlate the sensors values to the activity of the application. As one can see in Figure 23, on a higher level, it is possible to correlate system metrics with application program phases (left side of the picture). However, due to the limited resolution of the system metrics data, this is not possible on a more detailed level (see right side of the figure).

1.3.6 Opportunities for System Tuning

Several opportunities for system tuning arise from the availability of historic performance data collected by LWM[2]. First, data on individual system nodes along an extended period of time in comparison to other nodes can be analysed to spot anomalies and detect deficient components. Second, data on the entire workload can be used to improve the understanding of the workload requirements and configure the system accordingly. The insights obtained may guide the evolution of the system and influence future procurement decision. Finally, knowledge of the resource requirements of individual jobs offers the chance to develop resource-aware scheduling algorithms that avoid oversubscription of shared resources such as the file system or the network.

1.3.7 Conclusions

The HOPSA project creates an integrated diagnostic infrastructure for combined application and system tuning. Starting from system-wide basic performance screening of individual jobs, an automated workflow routes findings on potential bottlenecks either to application developers or system administrators with recommendations on how to identify their root-cause using more powerful diagnostics. The main result of the project is the specification of the performance analysis workflow that connects the different steps. At the same time, it provides an impression of the overall vision behind the project.

Although the specification is based on long experience with HPC application developers and how they tend to use performance tools, it is a blueprint that needs to be validated in practice. This validation was done in November 2012 at Moscow State University, where the HOPSA performance workflow and its performance tools were successfully tested during a week-long training week with application developers.

Beyond the lifetime of the project, the HOPSA infrastructure is supposed to collect large amounts of valuable data on the performance of individual applications as well as the system workload as a whole. It will be of interest in three ways: to tune individual applications, to tune the system for a given workload, and finally to observe the evolution of this workload over time. The latter will allow the effectiveness of our strategy to be studied. An open research issue to be tackled on the way will be the reliable tracking of individual applications, which may change over time, across jobs based on the collected data. In this way, it will become possible to document the performance history of code projects and demonstrate the effects of our tool environment over time.

References

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

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

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

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

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

[6] M. Gerndt and M. Ott. Automatic Performance Analysis with Periscope. Concurrency and Computation: Practice and Experience, 22(6):736–748, 2010.

[7] S. Shende and A. D. Malony. The TAU Parallel Performance System. International Journal of High Performance Computing Applications, 20(2):287–331, 2006. SAGE Publications.

[8] M. Geimer, P. Saviankou, A. Strube, Z. Szebenyi, F. Wolf, B. J. N. Wylie: Further improving the scalability of the Scalasca toolset. In: Proceedings of PARA 2010: State of the Art in Scientific and Parallel Computing, Part II: Minisymposium Scalable tools for High Performance Computing, Reykjavik, Iceland, June 6–9 2010, volume 7134 of Lecture Notes in Computer Science, pages 463–474, Springer, 2012.

[9] D. Eschweiler, M. Wagner, M. Geimer, A. Künpfer, W. E. Nagel, F. Wolf: Open Trace Format 2 - The Next Generation of Scalable Trace Formats and Support Libraries. In: Proceedings of the International Conference on Parallel Computing (ParCo), Ghent, Belgium, 2011, volume 22 of Advances in Parallel Computing, pages 481–490, IOS Press, 2012.

[10] A. Knüpfer, R. Brendel, H. Brunst, H. Mix, W. E. Nagel: Introducing the Open Trace Format (OTF), In: Vassil N. Alexandrov, Geert Dick van Albada, Peter M. A. Sloot, Jack Dongarra (Eds): Computational Science - ICCS 2006: 6th International Conference, Reading, UK, May 28–31, 2006, Proceedings, Part II, Springer Verlag, ISBN: 3-540-34381-4, pages 526–533, Vol 3992, 2006.

[11] F. Wolf, B. Mohr: EPILOG Binary Trace-Data Format. Technical Report FZJ-ZAM-IB-2004-06, Forschungszentrum Jülich, 2004.

[12] H. Servat Gelabert, G. Llort Sanchez, J. Gimenez, and J. Labarta. Detailed performance analysis using coarse grain sampling. In: Euro-Par 2009 - Parallel Processing Workshops, Delft, The Netherlands, August 2009, volume 6043 of Lecture Notes in Computer Science, pages 185–198. Springer, 2010.

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

[14] A.V. Adinets, P.A. Bryzgalov, V. Vad, V. Voevodin, S.A. Zhumatiy, D.A. Nikitenko. About one approach to monitoring, analysis and visualization of jobs on cluster system (In Russian). In: Numerical Methods and Programming, 2011, vol. 12, Pp. 90–93

[15] Z. Szebenyi, F. Wolf, B. J.N. Wylie. Space-Efficient Time-Series Call-Path Profiling of Parallel Applications. In: Proc. of the ACM/IEEE Conference on Supercomputing (SC09), Portland, OR, USA, ACM, 2009.

1.4 Potential impact, main dissemination activities and exploitation of results

1.4.1 Strategic impact

This project created an integrated diagnostic infrastructure for combined application and system tuning. Starting from system-wide basic performance screening of individual jobs, an automated workflow will route findings on potential bottlenecks either to application developers or system administrators with recommendations on how to identify their root-cause using more powerful diagnostics. To this end, the European partners contributed a collection of mature high-level tools for application performance analysis which were further enhanced with respect to their scalability, the depth of their analysis, and their support for asynchronous tasking. The tools were integrated into the workflow to ensure their most effective deployment.

On future large-scale systems, with their heterogeneous architectures and their increasingly dynamic configuration, which is needed in response to their higher frequency of component failures, asynchronous tasking is believed to be a competitive alternative to the classic and more rigid fork-join execution model. In addition to the simplicity of the task abstraction, major advantages also include the higher autonomy and flexibility of the runtime system in scheduling the different parts of a computation. In this way, the programmer is shielded from many low-level decisions such as when and to which type of heterogeneous device a task will be dispatched. On the other hand, the lack of tools that can analyse the performance implications of the additional level of parallelism represented by tasks makes engineering well-performing codes a complex undertaking.

This is precisely the scenario our project results will help to master. On a general level, our tuning environment encompassing both application and system performance analysis will help improve the efficiency of hybrid codes including those that utilise asynchronous tasking by providing insights into their performance behaviour and, thus, by guiding performance-relevant design decisions. The degree of automation offered by our environment will help achieve these improvements also faster, as tedious manual instrumentation and analysis of potentially unwieldy performance data sets will become dispensable and productivity of programmers is increased. Ultimately, the significant performance gain we expect will not only expand the potential of applications, making them fit for larger and more complex problems, but will also save valuable compute resources in terms of money and energy and, thus, lift the “scientific efficiency” of our computing infrastructure to higher levels. Below, we explain how exactly these benefits will materialise.

More frequent and more effective tool usage

System-wide performance screening without exception will distinguish the codes that utilise the underlying hardware well from those which do not and could therefore benefit from optimisation. This opens the way to implementing system usage policies intended to maximise the overall system throughput. At the beginning, users are just notified of their screening results with recommendations on how to proceed, that is, which further diagnostics should be conducted using which tool(s) or whom to ask for help. An important element of our project is that the initial classification of the performance behaviour during the screening allows the most suitable tool to be selected. If the performance problems persist even after a certain grace period has expired, which will not go unnoticed, a performance consultant from the service team may pro-actively contact the user to offer assistance. Further options include creating incentives for application tuning such as extra or discounted compute time, or an upgrade to a more powerful machine. In many cases, the screening will uncover otherwise hidden performance problems and will motivate the user to apply the tool that is most promising under the given circumstances. Systematically motivating a larger user base together with increased tool success because of the more accurate matching of problems with tools will increase the frequency and effectiveness of using our tool suite. At the same time, supporting the user in his choice of the right tool will avoid frustration that may occur as a result of using the wrong tool. This is important in view of the effort that is still required to become familiar with a new tool. A successful tool user is more likely to integrate tools into his daily routine and even to try new tools when they become available.

Enhancements of individual tools

In addition to their more frequent use and their more directed application to target codes, the tools being part of this project were also substantially enhanced, allowing the user to gain deeper insights into performance issues and, thus, to yield better optimisation results.

ThreadSpotter. Rogue Wave Software AB provides a world-leading technology for analysing the efficiency of parallel execution in coherent shared memory, i.e., OpenMP programs. One of the most important features of the ThreadSpotter technology is to hide the complexity of the matter to the programmer and only present the information that matters for the performance improvement of this application. In this project, the same methodology was taken one step further. Analysing the efficiency for 1000s of MPI strands would potentially increase the amount of information presented to a programmer 1000-fold. The solutions implemented in this project automatically filter out the unique strands' behaviours and reduce the optimisation problem to that of a single (or handful of) strands. This greatly simplifies the optimisation of hybrid program for exascale systems.

Paraver. The main enhancement for Paraver and CEPBA-Tools is related to the tools interoperability. The potential of the tools has been demonstrated in many cases, but this high potential makes them not very easy to use. The improved interoperability of the tools will open new ways of using the tools. A second enhancement was with respect to the scalability of the tools that up to now have been tested with up to 16k processes (except for Dimemas).

Scalasca. The new critical-path analysis simplifies the identification of optimisation targets in the code, substantially shortening the optimisation cycle time. The new distributed recording of communicators was the last step in a longer sequence that allows Scalasca to be comfortably used with more than 100,000 MPI processes and that significantly lowers the measurement overhead in the interest of more reliable performance data. Finally, the new compression of sampled time-series profiles helps analyse the evolution of performance phenomena in applications written in C++, a language used by a growing number of simulation-code developers.

Vampir. The Vampir tool was improved in terms of scalability, integration of system-level analysis and interoperability with partner tools. The scalability improvements in particular include trace recording support for long running programs and selective tracing of time intervals or processes/ threads. The system-level analysis provides monitoring data from the system monitoring of our project partners which is influencing the application-level performance behaviour. Finally, the new integration with the Scalasca and Vampir improves the ability to use several tools for different aspects of the performance analysis of one application.

Support for asynchronous tasking

In the past decade, the industry standard OpenMP 2.0 provided a stable foundation for programming parallel HPC platforms with shared memory. The basic execution model of OpenMP followed a strict fork-join scheme, which tightly synchronises the worker threads with the master thread. Its simplicity made it not only popular among application programmers, but also allowed performance tools to easily spot and highlight performance issues related to multithreading. However, this simple forkjoin-model does not fit current multi-core processors and especially accelerators such as GPUs very well. This resulted in new research on multi-threaded programming models. The basic abstraction all these new emerging programming models employ is the concept of asynchronous tasks. Examples are OpenMP 3.0 or StarSs with a higher level of abstraction as well as CUDA, OpenCL, or generic uncoordinated (POSIX) threading as lower-level alternatives. They all slightly differ in the way their runtime system schedules the tasks and the degree to which application programmers can influence scheduling decisions. Although all these programming models promise better performance for multi-threaded applications executing on heterogeneous systems, they pose the challenge to application programmers of how to structure their code into loosely-synchronised, asynchronous tasks such that optimal execution is guaranteed. Likewise, it is a challenge for performance tools to capture this asynchronous behaviour, to pinpoint performance issues, and to present the results to users in a meaningful way that is suitable to guide optimisation decisions. By developing an abstract characterisation of the performance of asynchronous tasking, the HOPSA project will allow code developers to better take advantage of higher-level programming abstractions that embody asynchronous tasks and, thus, develop their codes in a more portable way. The common performance abstractions we developed will also enable the different performance tools to present multi-threaded performance issues in a unified manner across all tools. Since all of our tools support at the same time MPI, and will continue to do so even in combination with tasking, hybrid applications that combine these two models, as often required in cluster environments, will equally benefit.

Greater harmony between tools

In addition to the enhancement of individual tools, their effectiveness is also promoted through closer integration. The new and well-defined HOPSA performance tool workflow guides even un-experienced application developer safely through the performance analysis and optimization process, where the tools represent actors in a chain of successively refined diagnostic steps. For example, if Scalasca identifies wait states in a certain code region, the time-line displays of Paraver or Vampir now allow the exploration of their precise circumstances via cross-tool controls. Moreover, a unified download, configuration, build and installation package drastically simplifies the combined installation (for system administrators) and usage (for application developers) of our tool suite.

Greater harmony between applications and system

Although many application performance problems can and should be addressed by the developer himself, for example, via re-coding relevant parts or replacing components with more efficient alternatives, some issues are in fact symptoms of a system-level bottleneck that may affect more than one application. Knowing the difference is crucial to ensure that the valuable time of application developers is not wasted on a problem he is not responsible for and that remediation is initiated as soon as possible. If the problem can be fixed by changing the system configuration, our diagnostic workflow guarantees that system administrators are informed at an early stage. In the same way, interference between applications running simultaneously will be pinpointed, supporting informed configuration decisions with respect to the capabilities of shared resources such as the network or the file systems. For more details on the impact of system tuning, please refer to the report of our Russian partners.

1.4.2 Socio-economic impact and the wider societal implications of the project so far

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. European citizens 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.

1.4.3 Main dissemination activities and exploitation of results

The HOPSA project partners were very successful in promoting both the HOPSA project as well as the HOPSA software tools as part of presentations, posters, BoFs, and flyers at major international HPC conferences (ISC, SC, EuroPar) and training activities of the leading European initiatives (PRACE, DEISA, VI-HPS). The following lists all dissemination actions for both years of the project in detail:

General dissemination actions

2013

- Article ***HOPSA erleichtert Optimierung*** (German) in Exascale Newsletter 1/2013, Forschungszentrum Jülich, Germany

2011

- Article ***HOPSA Project launched*** in InSiDE newsletter Nov 2011, GCS, Germany
- Article ***HOPSA Project launched*** in JSC News Feb 2011, JSC, Germany
- A coordinated press release about the HOPSA project was prepared in cooperation with the public relation departments of all EU and Russian partners. It was published on June 21st, 2011 just in time for the ISC 2011 conference in Hamburg. The press releases are available at
    - ***http://www.t-platforms.com/about-company/press-releases/183-hopsa.html***

- ***http://www.fz-juelich.de/SharedDocs/Pressemitteilungen/UK/EN/2011/11-06-20hopsa.html***
- ***http://www.bsc.es/media/4469.pdf***
- ***http://tu-dresden.de/die_tu_dresden/zentrale_einrichtungen/zih/publikationen/dateien/hopsa_en.pdf***
- ***http://www.grs-sim.de/news-events/news-archive/faster-computations-with-hopsa.html***

The press release got picked up by many other sides including HPCwire, InsideHPC, Primeur, Scientific Computing, and many more.

- A two-page fact sheet was prepared summarizing the project. It is available at the project website.
- A basic project website is available since June 2011 under the URL ***http://www.hopsa-project.eu*** describing the objectives and goals of the project and listing the project partners. The web site is embedded in the web site of the Virtual Institute – High-Productivity Supercomputing (VI-HPS) as the HOPSA project has many synergies with other funded project and training activities of VI-HPS.

Dissemination at international HPC conferences and workshops

In addition to the events listed in this section there also have been presentations at the 5th and 6th International Parallel Tools Workshop, PROPER 2012, PSTI 2012, IWOMP 2012, IPDPS 2012, ParCo 2012, EuroMPI 2011, and ICPADS 2011 (see section ***Refereed publications*** below).

2013 (planned)

- BSC will present the Tutorial ***Understanding applications performance with Paraver*** at EuroMPI 2013, Madrid, Spain on September 15th, 2013.
- The project will be well presented at the ***International Supercomputing Conference*** (ISC 2013) in Leipzig, June 16th to 20st, 2013. The EU partners BSC, JSC, Rogue Wave and TUD will have booths in the research or commercial exhibition of the conference. Our Russian partners MSU and T-Platforms will also have exhibition booths. The project will be presented with posters, flyers, and on-line demonstrations on the show floor. Furthermore:
  - JSC will present a half-day tutorial ***Performance Analysis & Optimization on Extreme-scale Systems*** on June 16th, 2013.
  - JSC, RW, and TUD will participate in the BoF ***Execution Analysis & Optimization of Parallel Applications*** on June 19th, 2013.
- JSC and TUD presented Score-P, Scalasca, and Vampir in the Tutorial ***Trace-based Performance Analysis with Vampir and Scalasca*** at SEA 2013, UCAR, Boulder, USA on April 4th and 5th, 2013.

2013

- TUD presented the HOPSA project at the APOS/HOPSA Workshop ***Exploiting Heterogeneous HPC Platforms*** in conjunction with HiPEAC 2013, Berlin, Germany on January 22nd, 2013.

2012

- The project was well presented at the ***International Conference for High Performance Computing, Networking, Storage and Analysis*** (SC 2012) in Salt Lake City, November 10th to 16th, 2012. The EU partners BSC, JSC, Rogue Wave and TUD and for the first time MSU had booths in the research or commercial exhibition of the conference. The project was presented with posters, flyers, and on-line demonstrations on the show floor. In addition, the HOPSA project and the HOPSA tool set was introduced and promoted in the following activities:
  - A half-day tutorial ***Supporting Performance Analysis and Optimization on Extreme-Scale Computer Systems*** on November 12, 2012. Involved partners: JSC.
  - GRS organized a Workshop ***Extreme-Scale Performance Tools*** on November 16, 2012. Involved partners: BSC, GRS, JSDC, TUD.

- JSC presented the HOPSA project at the Workshop ***International Research Collaboration in Computing Systems*** at HiPEAC Computing Systems Week, Gent, Belgium on October 17th, 2012.
- BSC, JSC, RW and TUD presented their tools at the ***Workshop on Tools for Exascale*** at CEA, Bruyères-le-Châtel, France on October 1st and 2nd, 2012.
- TUD and JSC presented Score-P [4], Scalasca [3], Vampir [5] and the HOPSA workflow [2] at the ***6th International Parallel Tools Workshop*** at HLRS, Stuttgart, Germany on September 25th and 26th, 2012.
- TUD and JSC presented Score-P, Vampir and Scalasca at the tutorial ***Practical Hybrid Parallel Application Performance Engineering*** at EuroMPI 2012, Vienna, Austria on September 23rd, 2012.
- The project was well presented at the ***International Supercomputing Conference*** (ISC 2012) in Hamburg, June 17th to 21st, 2012. The EU partners BSC, JSC, Rogue Wave and TUD had booths in the research or commercial exhibition of the conference. Our Russian partners MSU and T-Platforms also had exhibition booths. The project was presented with posters, flyers, and on-line demonstrations on the show floor. Furthermore:
    - A half-day tutorial ***Supporting Performance Analysis & Optimization on Extreme-Scale Computer Systems*** on June 17, 2012. Involved partners: JSC
    - A BoF ***Parallel Application Execution Analysis & Optimization*** on June 20th, 2012. Involved partners: RW, JSC
    - A BoF ***Automatic Node & System Performance Analysis at Scale*** on June 20th, 2012. Involved partners: RW, JSC
- JSC, MSU, and EPCC organized the EU-Russia APOS/HOPSA workshop at the Academy of Sciences, Moscow, Russia on May 30th, 2012.

2011

- The project was well presented at the ***International Conference for High Performance Computing, Networking, Storage and Analysis*** (SC 2011) in Seattle, November 14th to 17th, 2011. Again, the EU partners BSC, JSC, Rogue Wave and TUD had booths in the research or commercial exhibition of the conference. The project was presented with posters, flyers, and on-line demonstrations on the show floor. In addition, the HOPSA project and the HOPSA tool set was introduced and promoted in the following activities:
    - A full day tutorial including hands-on training ***Hands-on Practical Hybrid Parallel Application Performance Engineering*** was organized and held on November 13th, 2011. Involved partners: GRS, JSC, TUD.
    - A Birds-of-a-Feather (BoF) session ***System Monitoring Meets Application Performance Analysis - The HOPSA EU-Russia Project*** was organized and held on November 17th, 2011. Involved partners: GRS, JSC, TUD.
    - A Birds-of-a-Feather (BoF) session ***The Score-P Community Project -- An Interoperable Infrastructure for HPC Performance Analysis Tools*** was organized and held on November 17th, 2011.
- BSC, JSC, and TUD presented “Score-P - A Joint Performance Measurement Run-time Infrastructure for Periscope, Scalasca, TAU, and Vampir” [9], “Trace-Based Performance Analysis of GPU Accelerated Applications” and “Folding: Detailed Analysis with Coarse Sampling” [10] at the ***5th Parallel Tools Workshop***, HLRS/TUD, Dresden on Sep 26th to 27th, 2011.
- The project was well presented at the ***International Supercomputing Conference*** (ISC 2011) in Hamburg, June 19th to 23rd, 2011. The EU partners BSC, JSC, Rogue Wave and TUD had booths in the research or commercial exhibition of the conference. Our Russian partners MSU and T-Platforms also had exhibition booths. The project was presented with posters, flyers, and on-line demonstrations on the show floor. Furthermore:
    - A full day tutorial ***Hands-On Course on Debugging & Optimizing Parallel Programs*** which included ThreadSpotter was organized and held on June 19th, 2011 by Rogue Wave.
- Rogue Wave presented “How to port applications to new architectures” at the ***3rd PRACE Industrial Seminar***, Stockholm, Sweden, on March 29th, 2011.

HPC training events

2013 (planned)

- BSC will introduce Extrae, Paraver, and Dimemas at the PRACE Advanced Training Centre (PATC) training course ***Performance Analysis and Tools***, Barcelona, Spain on May 13th to 14th, 2013.
- JSC, GRS and TUD will provide a PATC "Score-P, Scalasca and Vampir Introduction and Hands-on Training" at the ***11th VI-HPS Tuning Workshop***, CEA, Saclay, France on April 22nd to 26th, 2013.

2012

- RW provided a ***ThreadSpotter Training Day***, Jülich Supercomputing Centre, Jülich, Germany on December 4th, 2012.
- All EU project partners introduced the complete HOPSA workflow and toolset at the ***APOS/HOPSA Training Week***, Research Computing Center, Moscow State University, Russia on November 27th to 30th, 2012.
- JSC, GRS and TUD provided a PATC "Score-P, Scalasca and Vampir Introduction and Hands-on Training" at the ***10th VI-HPS Tuning Workshop***, LRZ, Munich, Germany on October 16th to 19th, 2012.
- JSC and TUD the Scalasca and Vampir toolsets at the Training workshop ***Scalable Performance Analysis Tools for HPC Applications,*** CSCS, Lugano, Switzerland on August 9th and 10th, 2012.
- JSC gave an introduction into parallel performance analysis and presented the HOPSA tool set at the ***2012 European-U.S. Summer School on HPC Challenges in Computational Sciences***, Dublin, Ireland on June 26, 2012.
- JSC presented the Scalasca and Vampir toolsets at a ***Program Analysis and Tuning Workshop***, DKRZ, Hamburg, Germany on June 25th and 26th, 2012.
- BSC introduced Extrae, Paraver, and Dimemas at the PATC training course ***Performance Analysis and Tools***, Barcelona, Spain on May 21st to 22nd, 2012.
- JSC, GRS and TUD provided a PATC "Score-P, Scalasca and Vampir Introduction and Hands-on Training" at the ***9th VI-HPS Tuning Workshop***, Université de Versailles, St-Quentin-en-Yvelines, France on April 23rd to 27th, 2012.
- JSC gave an introduction into parallel performance analysis and presented the HOPSA tool set at the ***Doctoral School: Computational Interdisciplinary Modelling*** of the University of Innsbruck at Obergurgl, Austria on January 28th to 31st, 2012.
- BSC presented Paraver and Dimemas at ***CAPAP-H winter seminar***, Valladolid, Spain on January 26th to 27th, 2012.

2011

- BSC presented Paraver and Dimemas at the ***PRACE Autumn School 2011***, Bruyères-le-Châtel, France on October 25th to 27th, 2011.
- BSC, JSC, GRS and TUD provided a "Scalasca, Paraver and Vampir Introduction and Hands-on Training" at the ***8th VI-HPS Tuning Workshop***, RWTH, Aachen, Germany on September 5th to 9th, 2011.
- JSC presented "Performance analysis tools for massively parallel applications" and a "Scalasca Introduction" at the ***PRACE Summer School: Taking the Most Out of Supercomputers***, Helsinki, Finland on August 29th to September 1st, 2011.
- JSC gave an introduction into parallel performance analysis and presented the HOPSA tool set at the DEISA/PRACE/TeraGrid ***2011 European-US Summer School on HPC Challenges in Computational Sciences*** at Lake Tahoe, USA on Aug 9th, 2011.
- JSC presented an "Introduction to Performance Engineering and Scalasca" at the ***Joint HP-SEE, LinkSCEEM-2 and PRACE HPC Summer Training***, Athens, Greece on July 15th, 2011.

- JSC gave an introduction into parallel performance analysis and presented the HOPSA tool set at the ***Russian Summer School on Supercomputing Technologies*** at MSU, Moscow, Russia on July 1st, 2011.
- BSC presented Paraver and Dimemas at ***HPC-EUROPA2/TAM 2011***, Barcelona, Spain on June 8th to 9th, 2011.
- BSC and GRS presented Paraver, Dimemas and Scalasca at the INRIA Sumner School ***Toward petaflop numerical simulation on parallel hybrid architectures***, INRIA, Sophia Antipolis, France on June 6th to 10th, 2011.
- JSC presented "Parallel application performance analysis with Scalasca" at the ***DEISA/PRACE Spring School on Tools and Techniques for Extreme Scalability***, EPCC, Edinburgh, UK, on March 29th to 31st, 2011.
- RW, JSC, GRS and TUD provided a "ThreadSpotter, Scalasca and Vampir Introduction and Hands-on Training" at the ***7th VI-HPS Tuning Workshop***, HLRS, Stuttgart, Germany on March 28th to 30th, 2011.

Refereed publications

Please note that if one of the publications below were part of a conference or workshop proceedings that of course the paper was also presented at the corresponding workshop or conference even if the event is not explicitly listed under the section ***Dissemination at international HPC conferences and workshops*** above.

2013

[1] Youssef Hatem, ***Critical Path Analysis of Parallel Application Using OpenMP Tasks***. Master thesis. GRS Aachen, Forschungszentrum Jülich, to appear.

[2] Bernd Mohr, Vladimir Voevodin, Judit Giménez, Erik Hagersten, Andreas Knüpfer, Dmitry A. Nikitenko, Mats Nilsson, Harald Servat, Aamer Shah, Frank Winkler, Felix Wolf, and Ilya Zhujov. ***The HOPSA Workflow and Tools***. In: Proceedings of the 6th International Parallel Tools Workshop, Stuttgart, September 2012, Springer. To appear.

[3] Daniel Lorenz, David Böhme, Bernd Mohr, Alexandre Strube, and Zoltán Szebenyi. ***New developments in Scalasca***. In: Proceedings of the 6th International Parallel Tools Workshop, Stuttgart, September 2012, Springer. To appear.

[4] Andreas Knüpfer, Robert Dietrich, Jens Doleschal, Markus Geimer, Marc-André Hermanns, Christian Rössel, Ronny Tschüter, Bert Wesarg and Felix Wolf. ***Generic Support for Remote Memory Access Operations in Score-P and OTF2***. In: Proceedings of the 6th International Parallel Tools Workshop, Stuttgart, September 2012, Springer. To appear.

[5] Holger Brunst und Matthias Weber. ***Custom Hot Spot Analysis of HPC Software with the Vampir Performance Tool Suite***. In: Proceedings of the 6th International Parallel Tools Workshop, Stuttgart, September 2012, Springer. To appear.

[6] Harald Servat, Xavier Teruel, Germán Llort, Alejandro Duran, Judit Giménez, Xavier Martorell, Eduard Ayguadé and Jesús Labarta.***On the Instrumentation of OpenMP and OmpSs Tasking Constructs***. In: Proceedings of the PROPER2012 Workshop, Springer.
DOI:10.1007/978-3-642-36949-0_47

2012

[7] Zoltán Szebenyi. ***Capturing Parallel Performance Dynamics***. PhD thesis, RWTH Aachen University, volume 12 of IAS Series, Forschungszentrum Jülich, 2012, ISBN 978-3-89336-798-6.
URI:http://hdl.handle.net/2128/4603

[8] Daniel Lorenz, Peter Philippen, Dirk Schmidl, Felix Wolf. ***Profiling of OpenMP tasks with Score-P***. In Proc. of the 41st International Conference on Parallel Processing Workshops (ICPPW 2012), Pittsburgh, PA, USA, 2012.
DOI:10.1109/ICPPW.2012.62

[9] Andreas Knüpfer, Christian Rössel, Dieter an Mey, Scott Biersdorff, Kai Diethelm, Dominic Eschweiler, Markus Geimer, Michael Gerndt, Daniel Lorenz, Allen D. Malony, Wolfgang E. Nagel, Yury Oleynik, Peter Philippen, Pavel Saviankou, Dirk Schmidl, Sameer S. Shende, Ronny Tschüter, Michael Wagner, Bert Wesarg, Felix Wolf. ***Score-P – A Joint Performance***

Measurement Run-Time Infrastructure for Periscope, Scalasca, TAU, and Vampir. In Proc. of 5th Parallel Tools Workshop, 2011, Dresden, Germany, pages 79-91, Springer Berlin Heidelberg, September 2012.
DOI:10.1007/978-3-642-31476-6_7

[10] Harald Servat, Germán Llort, Judit Giménez, Kevin Huck, Jesús Labarta., ***Folding: detailed analysis with coarse sampling***. In Proc. of 5th Parallel Tools Workshop, 2011, Dresden, Germany, Springer Berlin Heidelberg, September 2012.
DOI:10.1007/978-3-642-31476-6_9

[11] Marc-André Hermanns, Markus Geimer, Bernd Mohr, Felix Wolf. ***Scalable detection of MPI-2 remote memory access inefficiency patterns***. Intl. Journal of High Performance Computing Applications (IJHPCA), 26(3):227–236, August 2012.
DOI:10.1177/1094342011406758

[12] Dirk Schmidl, Peter Philippen, Daniel Lorenz, Christian Rössel, Markus Geimer, Dieter an Mey, Bernd Mohr, Felix Wolf. ***Performance Analysis Techniques for Task-Based OpenMP Applications***. In Proc. of the 8th International Workshop on OpenMP (IWOMP), Rome, Italy, volume 7312 of Lecture Notes in Computer Science, pages 196–209, Berlin / Heidelberg, Springer, June 2012.
DOI:10.1007/978-3-642-30961-8_15

[13] David Böhme, Bronis R. de Supinski, Markus Geimer, Martin Schulz, Felix Wolf. ***Scalable Critical-Path Based Performance Analysis***. In Proc. of the 26th IEEE International Parallel & Distributed Processing Symposium (IPDPS), Shanghai, China, IEEE Computer Society, May 2012.
DOI:10.1109/IPDPS.2012.120

[14] David Böhme, Markus Geimer, Felix Wolf. ***Characterizing Load and Communication Imbalance in Large-Scale Parallel Applications***. In Proc. of the 26th IEEE International Parallel & Distributed Processing Symposium (IPDPS) PhD Forum, Shanghai, China, IEEE Computer Society, May 2012.
DOI:10.1109/IPDPSW.2012.321

[15] Dominic Eschweiler, Michael Wagner, Markus Geimer, Andreas Knüpfer, Wolfgang E. Nagel, Felix Wolf. ***Open Trace Format 2 - The Next Generation of Scalable Trace Formats and Support Libraries***. In Proc. of the Intl. Conference on Parallel Computing (ParCo), Ghent, Belgium, August 30 – September 2 2011, volume 22 of Advances in Parallel Computing, pages 481–490, IOS Press, 2012.
DOI:10.3233/978-1-61499-041-3-481

[16] Markus Geimer, Pavel Saviankou, Alexandre Strube, Zoltán Szebenyi, Felix Wolf, Brian J. N. Wylie. ***Further improving the scalability of the Scalasca toolset***. Proc. PARA 2010, Minisymposium Scalable tools for High Performance Computing, Reykjavik, Iceland, LNCS 7134, pp. 463-474, Springer, 2012.
DOI:10.1007/978-3-642-28145-7_45

2011

[17] Markus Geimer, Marc-André Hermanns, Christian Siebert, Felix Wolf, Brian J. N. Wylie. ***Scaling Performance Tool MPI Communicator Management***. Proc. EuroMPI 2011, Santorini, Greece, LNCS 6960, pp.178-187, Springer, 2011.
DOI:10.1007/978-3-642-24449-0_21

[18] Germán Llort, et al. ***Trace Spectral Analysis toward Dynamic Levels of Detail***. Proc. ICPADS 2011, Tainan, Taiwan, pp. 332 - 339, 2011.
DOI:10.1109/ICPADS.2011.142

1.5 Contact information

EU Project website: http://www.hopsa-project.eu

EU Project coordinator: Dr.-Ing. Bernd Mohr
Forschungszentrum Jülich GmbH
Jülich Supercomputing Centre
52425 Jülich

Tel: +49 2461 613218
Fax: +49 2461 616656
E-mail: b.mohr@fz-juelich.de

RU Project website: http://hopsa.parallel.ru

RU Project coordinator: Prof. Dr. Vladimir Voevodin
Moscow State University
Research Computing Center
Moscow, Russia

E-mail: voevodin@parallel.ru

2. Use and dissemination of foreground

2.1.1 Plan for the use and dissemination of foreground

The main result of the HOPSA project is a comprehensive, innovative, integrated, and proven set of performance measurement, analysis, and visualisation tools for parallel programs for HPC systems with heterogeneous components as well as well-defined specification of the overall performance analysis workflow. It will allow developers of compute-intensive application programs to optimally exploit the computational power of current and future HPC systems. Future dissemination and exploitation of results will be achieved in various ways:

- We will continue to maintain the project website established at http://www.hopsa-project.eu. At this website, one can find project deliverables, publications, news about upcoming and past dissemination and training events as well as the documentation and updates of the developed software. To foster the distribution of the project results, the project website will facilitate access to the publicly available software downloads.
- The project will continue to plan adequately resourced activities devoted to dissemination for specialised constituencies and the general public, in particular for awareness and educational purposes. The channels to be used will include web-based communication, brochures, booklets, multimedia material, etc. Of course, a proper acknowledgement of the funding source (the FP7 logo and the EU flag, etc.) will appear in all dissemination activities.
- There will continue to coordinate dissemination activities between EU and Russian partners to jointly communicate project objectives and results. For example, project will have a Birds-of-a-Feather (BOF) session for ISC 2013 where we will report on the final project results.
- The academic partners (BSC via UPC, GRS, and TUD) will, on the one hand, publish papers in journals and conferences, and on the other hand, build on HOPSA results to upgrade their courses so that their students can also start to experiment with the new technology.
- The research and HPC centres (BSC, JUELICH) will also participate in conferences where they present papers and demonstrate their tools. In addition, they regularly organise or contribute to international workshops where other projects are invited with a mix of academic and industrial people, and will continue to do so. The lessons learnt from the work carried out in HOPSA will also strengthen their position in international forums such as EESI2 or ETP4HPC, while the developed tools will enable them to provide high-level services to the scientific community that uses their high-performance computing systems.
- The HPC software providers (RW, TUD via GWT) will build on HOPSA technology to extend, improve, and optimise their products (ThreadSpotter, Vampir) and the services they offer, thus enabling their users to rapidly benefit from these enhancements. The project results are already (or will soon be) directly integrated into the commercial offerings. Moreover, these tools – having been designed and implemented in a coordinated way in HOPSA – will mutually leverage each other and reinforce their customer base (e.g. buyers of tool A will be encouraged to buy tool B that brings complementary features in a consistent way and vice-versa).
- The HOPSA exploitation plan also includes significant contributions to the development of several free and Open Source software solutions:
  - Scalasca (JUELICH, GRS)
  - Cube (JUELICH, GRS)
  - Opari2 (JUELICH, GRS)
  - OTF2 library (TUD)
  - Score-P (GRS, JUELICH, TUD)
  - Paraver (BSC)
  - Dimemas (BSC)
  - Extrae (BSC)
- Results and experiences are also directly exploited concurrently with other research projects the project partners are involved in.
- All project partners will continue to attend relevant conferences, workshops, and other events or will even organise some themselves to present project results in form of publications, presentations, tutorials, or posters. Our project will target the following conferences:

- Supercomputing (SC), http://supercomputing.org
- International Supercomputer Conference (ISC), http://www.supercomp.de
- IEEE International Parallel & Distributed Processing Symposium (IPDPS), http://www.ipdps.org
- Euro-Par, http://www.europar.org
- International Conference on Computational Science (ICCS), http://www.iccs-meeting.org
- and others

Our Russian partners will cover conferences located in Russian or conferences in Russian, for example:

- Parallel Computing Technologies (PaCT) International Conferences Series, http://ssd.sscc.ru/conferences.htm
- International Conference "Parallel computing technologies" (PaVT), http://agora.guru.ru/display.php?conf=pavt2011&page=item007
- Russian National Supercomputer Conference "Scientific Service in the Internet", http://agora.guru.ru/display.php?conf=abrau2011&page=item1
- and others

and through the Russian Supercomputing webportals http://supercomputers.ru/ and http://parallel.ru/ .

At large events such as SC the US and ISC in Germany, which offer an exhibition in addition to a technical program, RW, BSC, JUELICH, and TUD as well as T-Platforms will have booths showcasing their latest technology, using live-demos of tools to attract visitors. This will give these partners and their results high visibility. Another possibility to highlight our results is organising Bird-of-a-Feather Sessions (BoFs) at the above-mentioned events, e.g., with open discussions on the lessons learnt using new versions of our parallel performance tools.

In the following the exploitation and dissemination plans of the partners are described in more detail:

Rogue Wave Software AB (RW)

RW will exploit the results from the HOPSA project along two dimensions:

1. The new profile-based view of performance will enable enhanced versions of the existing RW products (supporting threads executing in a coherent shared address space) to be offered.
2. In a second dimension, RW will be enabled to offer more scalable analyses of performance for 1000s of MPI strands based on the work in this project. This will open a whole new business possibility for RW.

RW's current products offer a memory-centric issue-based view of performance, where a programmer can choose to penetrate performance issues presented in five sorted (worst-first) lists: bandwidth issues, latency issues, thread-interaction issues, cache-pollution issues and, finally, a loop-centric list concentrating loops with the worst issues. In the current form, a ThreadSpotter issue is defined as a performance problem related to the memory system. The programmer can, for example, be pointed to a piece of code that is wasting 15% of the overall memory bandwidth and told how to fix the problem.

However, currently a programmer cannot see which fraction of the overall execution time this issue is responsible for. The new profile-based view created in this project will allow an alternative entry point into ThreadSpotter and allow the programmer to, for example, start concentrating on a loop which is responsible for 15% of the execution time and to be confronted with the performance issues contained in that loop. This will greatly enhance the flexibility and productivity for general usage of ThreadSpotter and will not only apply to its usage in the MPI world. Also, less scalable versions of RW's tools, such as its Visual Studio plug-in, will be able to leverage this new feature.

While the profiling-based view will enhance all current RW products, the new scalable analysis of multiple MPI strands will create a completely new product for RW. The new technology developed will allow a programmer to concentrate on only the (few) unique performance behaviours that the many MPI strands will experience. It is expected that most strands will have a similar behaviour, which this new technology will automatically detect. Thus, the programmer will no longer need to wade through performance data collected from 1000s of MPI strands, and can instead concentrate on performance issues of the (expected) few different unique strand behaviours.

Barcelona Supercomputing Center (BSC)

Performance analysis tools are one of the research topics BSC have been working on for close to 20 years. It is very important for us to work on increasing the tools' interoperability and scalability improving the way that performance tools are used. BSC tools are freely distributed as open source, so all the developments and improvements implemented in HOPSA are open to the HPC community and all tool users will benefit from the enhancements implemented.

BSC is using the tools on public funded project as well as in our cooperation with companies to offer consulting services for the performance analysis of applications, providing recommendation son how to optimise their codes. This is applied to both code developers and users as well as system developers. To promote the use of this technology, BSC continues organising workshops and tutorials to train in the usage of performance analysis tools.

Forschungszentrum Jülich GmbH (JUELICH)

With our Scalasca software, JUELICH is, together with its partner GRS, the world-leading expert in automatic trace-based performance analysis of highly scalable parallel applications. Privileged access to the most powerful computer system of Europe (Juqueen) and more than a decade of experience in this area gives JUELICH an advance of a few years compared to other projects. The HOPSA project helped JUELICH maintain and expand its leading position.

The Scalasca toolset (and its predecessor KOJAK) used in the project has been open source since its first release in 2003 and therefore can be used practically without restrictions by the HPC community. We use the very liberal New BSD License that also allows for free commercial use. This makes us an attractive partner for well-established computer vendors like IBM, Intel or Cray. Current vendor collaborations are the Exascale Innovation Center (EIC) together with IBM and the ExaCluster Laboratory (ECL) with Intel.

The performance analysis tools developed in HOPSA are installed on the JSC production computer systems as well as on many internal cluster system used for project work and research. Thus, users as well as JSC user support personnel already benefited from the advances of the HOPSA project, allowing them to more easily analyse the performance of their applications, which will result in more optimised and efficient use of the systems.

The HOPSA tools will also be exploited in our manifold education and training programs, which will further strengthen our position as competence center for parallel programming and program optimisation. This includes not only training classes lasting one or more days for the users of our production computer systems but also seminars and courses being part of the bachelor program Technomathematics and the master program Scientific Programming, which we offer in cooperation with the FH Aachen ("technical college"), as well as the master and the Ph.D. program of our partner GRS (see below). In this way, results of the HOPSA project will directly influence the education of next-generation scientists in the area of computational science. JUELICH as partner in the German Gauss Centre for Supercomputing (GCS) is also PRACE Advanced Training Centers (PATC).

JUELICH will continue to organise workshops and other events promoting the use of performance tools for parallel programs. For example, JUELICH, in cooperation with LLNL, will organise the Dagstuhl seminar " Connecting Performance Analysis and Visualization to Advance Extreme Scale Computing" (see http://www.dagstuhl.de/14022). The topic of the seminar has a close relationship to the goal of the HOPSA project in the same manner as our efforts in the organisation and teaching of tutorials on performance tools for parallel programs (such as our well-received tutorials at Supercomputing (SC) and at the International Supercomputing Conference (ISC)).

The increased visibility and competence that we achieved with the participation in the HOPSA project will also allow our activities in the research area to be further developed. It will help us to successfully apply for further German, EU, or other international research funding.

German Research School for Simulation Sciences (GRS)

As a partner in the Scalasca project, the GRS plans to contribute the extensions of Scalasca it has developed to new versions of the software, which will be released together with JUELICH under the New BSD license in regular intervals. The software, which is installed at numerous sites in several countries, will be used by application developers to tune the performance of their codes. A support email list, which is answered quickly by staff from JUELICH and GRS, provides assistance in installing and using the software. Releases typically happen shortly before major conferences such as ISC in June or SC in November to maximise the attention the announcements made at these conferences can receive. Additional advertisement for new releases with a list of new features will be placed in every public Scalasca-related presentation of GRS staff and will be distributed via a number of community email lists including the Scalasca news list. The Scalasca website, which the GRS has redesigned with the help of a corporate publishing company, explicitly refers to the HOPSA project and features a link to the HOPSA website. Further attention to HOPSA results can be expected from the Virtual Institute - High Productivity Supercomputing (VI-HPS), which is coordinated by Felix Wolf, HOPSA's principal investigator from GRS. The institute's widely known training program with at least two multi-day tuning workshops per year in and beyond Germany teaches the effective use of several HPC programming tools including Vampir and Scalasca. During such workshops, staff from partner organisations including GRS work together with application scientists on the optimisation of their codes. In addition to the optimisation successes that can be achieved right there, these workshops are also a suitable medium to receive feedback from early users.

Moreover, GRS is preparing a public release of the lightweight measurement module LWM2 which has been developed in HOPSA, again under the New BSD license. To simplify deployment outside the HOPSA testbed, GRS plans to develop a portable and independent database solution for LMW2 that is more lightweight than LAPTA and does not depend on a specific cluster middleware. GRS is already working with a number of institutes and companies for dissemination of tools and research of HOPSA project. This includes collaboration with Tokyo Institute of Technology with a potential deployment of LWM2 on Tsubame-2 HPC system, and with CSCS Swiss National Supercomputing Centre for deployment of LWM2 on HPC systems, where the tool is already being tested on a limited basis.

As a public research institution, the GRS will publish research results using the classic academic dissemination channels such as workshops, conferences, and peer-reviewed journals. Committed to education in the methods of simulation sciences, the GRS also plans to enrich its lecture program in parallel programming in the international master program Simulation Sciences with insights gained from the project. Moreover, GRS will continue to promote the young scientists hired as part of the project. Within our various research projects, our staff will also use features developed in HOPSA when they cooperate with application groups. Finally, the GRS plans to leverage the project results as a basis for further collaborations with HOPSA partners beyond the official end of the project.

.

Technische Universität Dresden (TUD)

The HOPSA project helped to strengthen the established position of the Vampir software as the most well known and most scalable commercial event trace visualiser in the worldwide HPC community. The distribution of the Vampir GUI will continue in a commercial way in cooperation with the GWT TU Dresden GmbH, a company associated with the university for the transfer of research and technology. The improvements in scalability and interoperability with tools of our partners will be an advantage in the competition with other tools offered by hardware vendors. In addition, it will be a major benefit for common training activities.

Besides the commercial Vampir tool, the packages VampirTrace and OTF were and are distributed as Open Source software, which is important for the acceptance among academic and industry users when linking with other free or proprietary application software. In the same way, the new Score-P monitoring software will be distributed under the New BSD Open Source license. The free license was also essential for the integration of VampirTrace and OTF into the widely used Open MPI project. After the transition from VampirTrace to the Score-P measurement system, we will strive to establish the integration into the Open MPI package again, potentially also in further 3rd party software projects.

Besides the development and distribution of the performance analysis software tools, TU Dresden together with JUELICH and GRS as well as external partners like RWTH Aachen and TU Munich offer training events. They cover not only a single tool but a variety of complementary tools, which is why increased interoperability is of great benefit. In the past, a number of tutorials including hands-on practical exercises were offered; many organised or related to the VI-HPS. The training events are important for bringing maximum benefit to the users of our tools and also to increase visibility of our tools in the HPC community.

Section A (public)

TEMPLATE A1: LIST OF SCIENTIFIC (PEER REVIEWED) PUBLICATIONS, STARTING WITH THE MOST IMPORTANT ONES										
NO.	Title	Main author	Title of the periodical or the series	Number, date or frequency	Publisher	Place of publication	Year of publication	Relevant pages	Permanent identifiers[1] (if available)	Is open access[2] provided to this publication?
1	The HOPSA Workflow and Tools	Bernd Mohr	Proceedings of the 6th International Parallel Tools Workshop	annual	Springer	Berlin	2013			no
2	New developments in Scalasca	Daniel Lorenz	Proceedings of the 6th International Parallel Tools Workshop	annual	Springer	Berlin	2013			no
3	Generic Support for Remote Memory Access Operations in Score-P and OTF2	Andreas Knüpfer	Proceedings of the 6th International Parallel Tools Workshop	annual	Springer	Berlin	2013			no

[1] A permanent identifier should be a persistent link to the published version full text if open access or abstract if article is pay per view) or to the final manuscript accepted for publication (link to article in repository).

[2] Open Access is defined as free of charge access for anyone via Internet. Please answer "yes" if the open access to the publication is already established and also if the embargo period for open access is not yet over but you intend to establish open access afterwards.

4	Custom Hot Spot Analysis of HPC Software with the Vampir Performance Tool Suite	Holger Brunst	Proceedings of the 6th International Parallel Tools Workshop	annual	Springer	Berlin	2013			no
5	On the Instrumentation of OpenMP and OmpSs Tasking Constructs	Harald Servat	Proceedings of the PROPER2012 Workshop	annual	Springer	Berlin	2013		***DOI:10.1007/978-3-642-36949-0_47***	no
6	Capturing Parallel Performance Dynamics	Zoltán Szebenyi	IAS Series	volume 12	Forschungszentrum Jülich	Jülich	2012		ISBN 978-3-89336-798-6	yes
7	Profiling of OpenMP tasks with Score-P	Daniel Lorenz	Proc. of the 41st International Conference on Parallel Processing Workshops (ICPPW 2012)	annual	IEEE		2012		***DOI:10.1109/ICPPW.2012.62***	no
8	Score-P – A Joint Performance Measurement Run-Time Infrastructure for Periscope, Scalasca, TAU, and Vampir	Andreas Knüpfer	Proc. of 5th Parallel Tools Workshop	annual	Springer	Berlin	2012	79-91	***DOI:10.1007/978-3-642-31476-6_7***	no
9	Folding: detailed analysis with coarse sampling	Harald Servat	Proc. of 5th Parallel Tools Workshop	annual	Springer	Berlin	2012		***DOI:10.1007/978-3-642-31476-6_9***	no
10	Scalable detection of MPI-2 remote memory access inefficiency patterns	Marc-André Hermanns	Intl. Jl. of High Performance Computing Applications (IJHPCA)	26(3)			2012	227-236	***DOI:10.1177/1094342011406758***	no

11	Performance Analysis Techniques for Task-Based OpenMP Applications	Dirk Schmidl	Proc. of the 8th International Workshop on OpenMP (IWOMP)	LNCS 7312	Springer	Berlin	2012	196–209	***DOI:10.1007/978-3-642-30961-8_15***	no
12	Scalable Critical-Path Based Performance Analysis	David Böhme	Proc. of the 26th IEEE International Parallel & Distributed Processing Symposium (IPDPS)	annual	IEEE Computer Society		2012		***DOI:10.1109/IPDPS.2012.120***	no
13	Characterizing Load and Communication Imbalance in Large-Scale Parallel Applications	David Böhme	Proc. of the 26th IEEE International Parallel & Distributed Processing Symposium (IPDPS)	annual	IEEE Computer Society		2012		***DOI:10.1109/IPDPSW.2012.321***	no
14	Open Trace Format 2 - The Next Generation of Scalable Trace Formats and Support Libraries	Dominic Eschweiler	Proc. of the Intl. Conference on Parallel Computing (ParCo)	volume 22 of Advances in Parallel Computing	IOS Press		2012	481-490	***DOI:10.3233/978-1-61499-041-3-481***	no
15	Further improving the scalability of the Scalasca toolset	Markus Geimer	Proc. PARA 2010, Minisymposium Scalable tools for High Performance Computing	LNCS 7134	Springer	Berlin	2012	463-474	***DOI:10.1007/978-3-642-28145-7_45***	no

16	Scaling Performance Tool MPI Communicator Management	Markus Geimer	Proc. EuroMPI 2011	LNCS 6960	Springer	Berlin	2011	178-187	***DOI:10.1007/978-3-642-24449-0_21***	no
17	Trace Spectral Analysis toward Dynamic Levels of Detail	Germán Llort	Proc. ICPADS 2011				2011	332 - 339	***DOI:10.1109/ICPADS.2011.142***	no

TEMPLATE A2: LIST OF DISSEMINATION ACTIVITIES

NO.	Type of activities[3]	Main leader	Title	Date/Period	Place	Type of audience[4]	Size of audience	Countries addressed
1	Web	JSC	Project website http://www.hopsa-project.eu	Feb 1, 2011	Jülich, DE	All		International
2	Flyer	JSC	2-page HOPSA fact sheet	Feb 1, 2011	Jülich, DE	All		International
3	Other: Newsletter	JSC	HOPSA Project launched, JSC News	Feb 2011	Jülich, DE	All		EU
4	Other: Training	JSC	7th VI-HPS Tuning Workshop	Mar 28-30, 2011	Stuttgart, DE	Research, Industry	30	EU
5	Other: Training at Workshop	JSC	“Parallel application performance analysis with Scalasca”, DEISA/PRACE Spring School on Tools and Techniques for Extreme Scalability	Mar 29-31, 2011	Edinburgh, 2011	Research, Industry		EU
6	Other: Presentation at Workshop	RW	“How to port applications to new architectures”, 3rd PRACE Industrial Seminar	Mar 29, 2011	Stockholm, SE	Industry		EU
7	Other: Training at Workshop	BSC	“Paraver and Dimemas”, Sumner School Toward petaflop numerical simulation on parallel hybrid architectures	June 6-10, 2011	Sophia Antipolis, FR	Research, Industry		EU

[3] A drop down list allows choosing the dissemination activity: publications, conferences, workshops, web, press releases, flyers, articles published in the popular press, videos, media briefings, presentations, exhibitions, thesis, interviews, films, TV clips, posters, Other.

[4] A drop down list allows choosing the type of public: Scientific Community (higher education, Research), Industry, Civil Society, Policy makers, Medias, Other ('multiple choices' is possible).

8	Other: Training at Workshop	GRS	“Scalasca”, Sumner School Toward petaflop numerical simulation on parallel hybrid architectures	June 6-10, 2011	Sophia Antipolis, FR	Research, Industry		EU
9	Other: Training at Workshop	BSC	“Paraver and Dimemas”, HPC-EUROPA2/TAM 2011	June 8, 2011	Barcelona, ES	Research, Industry		EU
10	Other: Tutorial at Conference	RW	Hands-On Course on “Debugging & Optimizing Parallel Programs”, ISC2011	Jun 19, 2011	Hamburg, DE	Research, Industry		International
11	Exhibition	BSC	Research booth, ISC2011	Jun 20-23, 2011	Hamburg, DE	All		International
12	Exhibition	JSC	Research booth, ISC2011	Jun 20-23, 2011	Hamburg, DE	All		International
13	Exhibition	TUD	Research booth, ISC2011	Jun 20-23, 2011	Hamburg, DE	All		International
14	Exhibition	MSU	Research booth, ISC2011	Jun 20-23, 2011	Hamburg, DE	All		International
15	Exhibition	RW	Commercial booth, ISC2011	Jun 20-23, 2011	Hamburg, DE	All		International
16	Exhibition	T-Platform	Commercial booth, ISC2011	Jun 20-23, 2011	Hamburg, DE	All		International
17	Press release	JSC	Faster Computations with HOPSA	Jun 21, 2011	Jülich, DE	All		International
18	Press release	T-Platform	Faster Computations with HOPSA	Jun 21, 2011	Moscow, RU	All		International
19	Press release	BSC	Faster Computations with HOPSA	Jun 21, 2011	Barcelona, ES	All		International
20	Press release	TUD	Faster Computations with HOPSA	Jun 21, 2011	Dresden, DE	All		International
21	Press release	GRS	Faster Computations with HOPSA	Jun 21, 2011	Aachen, DE	All		International
22	Other: Training at Workshop	JSC	“Introduction into parallel performance analysis”, Russian Summer School on Supercomputing Technologies	Jul 1, 2011	Moscow, RU	Higher Education	35	RU
23	Other: Training at Workshop	JSC	“Introduction to Performance Engineering and Scalasca”, Joint HP-SEE, LinkSCEEM-2 and PRACE HPC Summer Training	Jul 15, 2011	Athens, GR	Research, Industry		EU

24	Other: Training at Workshop	JSC	Introduction into parallel performance analysis", DEISA/PRACE/TeraGrid 2011 European-US Summer School on HPC Challenges in Computational Sciences	Aug 9, 2011	Lake Tahoe, US	Higher Education	60	EU, US
25	Other: Training at Workshop	JSC	"Performance analysis tools for massively parallel applications" and "Scalasca Introduction", PRACE Summer School: Taking the Most Out of Supercomputers	Aug 29-Sep1, 2011	Helsinki, FI	Research, Industry		EU
26	Other: Presentation at Conference + Publication	JSC	"Open Trace Format 2 - The Next Generation of Scalable Trace Formats and Support Libraries", ParCo2011	Aug 30 - Sep 2, 2011	Gent, BE	Research, Industry		International
27	Other: Training	GRS	8th VI-HPS Tuning Workshop	Sep 5-9, 2011	Aachen, DE	Research, Industry	32	EU
28	Other: Presentation at Conference + Publication	JSC	"Scaling Performance Tool MPI Communicator Management", EuroMPI 2011	Sep 18-21, 2011	Santorini, GR	Research, Industry		International
29	Other: Presentation at Workshop + Publication	BSC	"Folding: Detailed Analysis with Coarse Sampling", 5th Parallel Tools Workshop	Sep 26-27, 2011	Stuttgart, DE	Research, Industry		International
30	Other: Presentation at Workshop	JSC	"Trace-Based Performance Analysis of GPU Accelerated Applications", 5th Parallel Tools Workshop	Sep 26-27, 2011	Stuttgart, DE	Research, Industry		International
31	Other: Presentation at Workshop + Publication	TUD	"Score-P - A Joint Performance Measurement Run-time Infrastructure for Periscope, Scalasca, TAU, and Vampir", 5th Parallel Tools Workshop	Sep 26-27, 2011	Stuttgart, DE	Research, Industry		International
32	Other: Training at Workshop	BSC	"Paraver and Dimemas" PRACE Autumn School 2011	Oct 25-27, 2011	Bruyères-le-Châtel, FR	Research, Industry		EU

33	Other: Newsletter	JSC	HOPSA Project launched, inSiDE newsletter	Nov2011	Stuttgart, DE	All		International
34	Other: Tutorial at Conference	JSC	Hands-on “Practical Hybrid Parallel Application Performance Engineering”, SC2011	Nov 13, 2011	Seattle, US	Research, Industry	30	International
35	Exhibition	BSC	Research booth, SC2011	Nov 14-17, 2011	Seattle, US	All		International
36	Exhibition	JSC	Research booth, SC2011	Nov 14-17, 2011	Seattle, US	All		International
37	Exhibition	TUD	Research booth, SC2011	Nov 14-17, 2011	Seattle, US	All		International
38	Exhibition	RW	Commercial booth, SC2011	Nov 14-17, 2011	Seattle, US	All		International
39	Other: BoF at Conference	JSC	“System Monitoring Meets Application Performance Analysis - The HOPSA EU-Russia Project”, SC20011	Nov 17, 2011	Seattle, US	Research, Industry	12	International
40	Other: BoF at Conference	TUD	“The Score-P Community Project -- An Interoperable Infrastructure for HPC Performance Analysis Tools”, SC2011	Nov 17, 2011	Seattle, US	Research, Industry	30	International
41	Other: Presentation at Conference + Publication	BSC	“Trace Spectral Analysis toward Dynamic Levels of Detail”, ICPADS 2011	Dec 7-9, 2011	Tainan, TW	Research, Industry		International
42	Publication	JSC	“Further improving the scalability of the Scalasca toolset”, PARA2010	2012		Research, Industry		International
43	Thesis	GRS	“Capturing Parallel Performance Dynamics”	2012	Aachen, DE	Research, Industry		International
44	Other: Training at Workshop	BSC	“Paraver and Dimemas”, CAPAP-H winter seminar	Jan 26-27, 2012	Valladolid, ES	Research, Industry		ES
45	Other: Training at Workshop	JSC	“Introduction into parallel performance analysis“ , Doctoral School: Computational Interdisciplinary Modelling	Jan 28-31, 2012	Obergurgl, AT	Higher education	60	AT

46	Other: Training	JSC	9th VI-HPS Tuning Workshop	Apr 23-27, 2012	St-Quentin-en-Yvelines, FR	Research, Industry	35	EU
47	Other: Training	BSC	"Extrae, Paraver, and Dimemas", PATC training course Performance Analysis and Tools	May 21-22, 2012	Barcelona, ES	Research, Industry		EU
48	Other: Presentation at Conference + Publication	GRS	"Scalable Critical-Path Based Performance Analysis", IPDPS2012	May 21-25, 2012	Shanghai, CN	Research, Industry		International
49	Other: Presentation at Conference + Publication	GRS	"Characterizing Load and Communication Imbalance in Large-Scale Parallel Applications", IPDPS2012	May 21-25, 2012	Shanghai, CN	Research, Industry		International
50	Workshop	JSC	EU-Russia APOS/HOPSA Workshop	May 30, 2012	Moscow, RU	Research, Industry, Policy Maker	60	EU, RU
51	Other: Presentation at Workshop + Publication	JSC	"Performance Analysis Techniques for Task-Based OpenMP Applications", IWOMP2012	Jun 11-13, 2012	Rome, IT	Research, Industry		International
52	Other: Tutorial at Conference	JSC	"Supporting Performance Analysis & Optimization on Extreme-Scale Computer Systems", ISC2012	Jun 17, 2012	Hamburg, DE	Research, Industry	30	International
53	Exhibition	BSC	Research booth, ISC2012	Jun 18-21, 2012	Hamburg, DE	All		International
54	Exhibition	JSC	Research booth, ISC2012	Jun 18-21, 2012	Hamburg, DE	All		International
55	Exhibition	TUD	Research booth, ISC2012	Jun 18-21, 2012	Hamburg, DE	All		International
56	Exhibition	MSU	Research booth, ISC2012	Jun 18-21, 2012	Hamburg, DE	All		International
57	Exhibition	RW	Commercial booth, ISC2012	Jun 18-21, 2012	Hamburg, DE	All		International

58	Exhibition	T-Platform	Commercial booth, ISC2012	Jun 18-21, 2012	Hamburg, DE	All		International
59	Other: BoF at Conference	RW	“Parallel Application Execution Analysis & Optimization”, ISC2012	Jun 20, 2012	Hamburg, DE	Research, Industry	45	International
60	Other: BoF at Conference	JSC	“Automatic Node & System Performance Analysis at Scale”, ISC2012	Jun 20, 2012	Hamburg, DE	Research, Industry	45	International
61	Other: Training at Workshop	JSC	“Scalasca and Vampir”, Program Analysis and Tuning Workshop	Jun 25-26, 2012	Hamburg, DE	Research, Industry	16	International
62	Other: Training at Workshop	JSC	“Introduction into parallel performance analysis“, 2012 European-U.S. Summer School on HPC Challenges in Computational Sciences	June 26, 2012	Dublin, IR	Higher Education	60	EU, US
63	Publication	GRS	“Scalable detection of MPI-2 remote memory access inefficiency patterns”, Intl. Journal of High Performance Computing Applications (IJHPCA), 26(3):227–236	Aug 2012		Research, Industry		International
64	Other: Training	JSC	“Scalasca and Vampir”, Training workshop Scalable Performance Analysis Tools for HPC Applications	Aug 9-10, 2012	Lugano, CH	Research, Industry	10	CH
65	Other: Presentation at Workshop + Publication	BSC	“On the Instrumentation of OpenMP and OmpSs Tasking Constructs”, PROPER2012	Aug 27-31, 2012	Rhodes, GR	Research, Industry		International
66	Other: Presentation at Workshop + Publication	JSC	“Profiling of OpenMP tasks with Score-P”, ICPPW 2012	Sep 10-13, 2012	Pittsburgh, US	Research, Industry		International
67	Other: Tutorial at Conference	JSC	“Practical Hybrid Parallel Application Performance Engineering”, EuroMPI 2012	Sep 23, 2012	Vienna, AT	Research, Industry	12	International

68	Other: Presentation at Workshop + Publication	TUD	"Custom Hot Spot Analysis of HPC Software with the Vampir Performance Tool Suite", 6th Parallel Tools Workshop	Sep 25-26, 2012	Stuttgart, DE	Research, Industry		International
69	Other: Presentation at Workshop + Publication	TUD	"Generic Support for Remote Memory Access Operations in Score-P and OTF2", 6th Parallel Tools Workshop	Sep 25-26, 2012	Stuttgart, DE	Research, Industry		International
70	Other: Presentation at Workshop + Publication	JSC	"The HOPSA Workflow and Tools", 6th Parallel Tools Workshop	Sep 25-26, 2012	Stuttgart, DE	Research, Industry		International
71	Other: Presentation at Workshop + Publication	JSC	"New developments in Scalasca", 6th Parallel Tools Workshop	Sep 25-26, 2012	Stuttgart, DE	Research, Industry		International
72	Other: Presentation at Workshop	BSC	"Paraver", Workshop on Tools for Exascale	Oct 1-2, 2012	Bruyères-le-Châtel, FR	Research, Industry	100	EU
73	Other: Presentation at Workshop	JSC	"Scalasca", Workshop on Tools for Exascale	Oct 1-2, 2012	Bruyères-le-Châtel, FR	Research, Industry	100	EU
74	Other: Presentation at Workshop	RW	"ThreadSpotter", Workshop on Tools for Exascale	Oct 1-2, 2012	Bruyères-le-Châtel, FR	Research, Industry	100	EU
75	Other: Presentation at Workshop	TUD	"Vampir", Workshop on Tools for Exascale	Oct 1-2, 2012	Bruyères-le-Châtel, FR	Research, Industry	100	EU
76	Other: Presentation at Workshop	JSC	"HOPSA", Workshop International Research Collaboration in Computing Systems at HiPEAC Computing Systems Week	Oct 17, 2012	Gent, BE	Research, Industry, Policy Maker	20	EU
77	Other: Training	JSC	PATC 10th VI-HPS Tuning Workshop	Oct 16-19, 2012	Munich, DE	Research, Industry	12	EU
78	Other: Tutorial at Conference	JSC	Hands-on "Supporting Performance Analysis and Optimization on Extreme-Scale Computer Systems", SC2012	Nov 12, 2012	Salt Lake City, US	Research, Industry	30	International
79	Exhibition	BSC	Research booth, SC2012	Nov 13-16, 2012	Salt Lake City, US	All		International

80	Exhibition	JSC	Research booth, SC2012	Nov 13-16, 2012	Salt Lake City, US	All		International
81	Exhibition	TUD	Research booth, SC2012	Nov 13-16, 2012	Salt Lake City, US	All		International
82	Exhibition	MSU	Research booth, SC2012	Nov 13-16, 2012	Salt Lake City, US	All		International
83	Exhibition	RW	Commercial booth, SC2012	Nov 13-16, 2012	Salt Lake City, US	All		International
84	Workshop	GRS	"Extreme-Scale Performance Tools", SC2012	Nov 16, 2012	Salt Lake City, US	Research, Industry	50	International
85	Other: Training	JSC	APOS/HOPSA Training Week	Nov 27-30, 2012	Moscow, RU	Research, Industry	18	EU, RU
86	Other: Training	JSC	ThreadSpotter Training Day	Dec 4, 2012	Jülich, DE	Research, Industry	41	DE, NL
87	Other: Newsletter	JSC	HOPSA erleichert Optimierung, FZJ Exascale newsletter	Jan 2013	Jülich, DE	All		DE
88	Other: Presentation at Workshop	TUD	"HOPSA", APOS/HOPSA Workshop Exploiting Heterogeneous HPC Platforms in conjunction with HiPEAC 2013	Jan 22, 2013	Berlin, DE	Research, Industry		EU
89	Other: Presentation at Conference	TUD	"Interactive Performance Analysis with Vampir", SEA2013	Apr 3, 2013	Boulder, US	Research, Industry		International
90	Other: Presentation at Conference	JSC	"The Scalasca Performance Analysis Toolset", SEA2013	Apr 3, 2013	Boulder, US	Research, Industry		International
91	Other: Presentation at Conference	JSC	"The Score-P run-time measurement system for the TAU, Vampir, and Scalasca tools", SEA2013	Apr 3, 2013	Boulder, US	Research, Industry		International
92	Other: Tutorial at Conference	JSC	Hands-on "Trace-based Performance Analysis with VAMPIR and Scalasca", SEA2013	Apr 4-5, 2013	Boulder, US	Research, Industry		International

93	Other: Training	JSC	11th VI-HPS Tuning Workshop	Apr 22-26, 2013	Saclay, FR	Research, Industry		EU
94	Other: Training	BSC	"Extrae, Paraver, and Dimemas", PATC training course Performance Analysis and Tools	May 13-14, 2013	Barcelona, ES	Research, Industry		EU
95	Other: Tutorial at Conference	JSC	"Performance Analysis & Optimization on Extreme-scale Systems", ISC2013	Jun 16, 2013	Leipzig, DE	Research, Industry		International
96	Exhibition	BSC	Research booth, ISC2013	Jun 17-20, 2013	Leipzig, DE	All		International
97	Exhibition	JSC	Research booth, ISC2013	Jun 17-20, 2013	Leipzig, DE	All		International
98	Exhibition	TUD	Research booth, ISC2013	Jun 17-20, 2013	Leipzig, DE	All		International
99	Exhibition	MSU	Research booth, ISC2013	Jun 17-20, 2013	Leipzig, DE	All		International
100	Exhibition	RW	Commercial booth, ISC2013	Jun 17-20, 2013	Leipzig, DE	All		International
101	Exhibition	T-Platform	Commercial booth, ISC2013	Jun 17-20, 2013	Leipzig, DE	All		International
102	Other: BoF at Conference	JSC	"Execution Analysis & Optimization of Parallel Applications", ISC2013	Jun 19, 2013	Leipzig, DE	Research, Industry		International
103	Other: Tutorial at Conference	BSC	"Understanding applications performance with Paraver", EuroMPI 2013	Sep 15, 2013	Madrid, ES	Research, Industry		International
104	Thesis	GRS	"Critical Path Analysis of Parallel Application Using OpenMP Tasks"	2013	Aachen, DE	Research, Industry		International

Section B (Public)

Part B1

TEMPLATE B1: LIST OF APPLICATIONS FOR PATENTS, TRADEMARKS, REGISTERED DESIGNS, ETC.					
Type of IP Rights[5]:	Confidential Click on YES/NO	Foreseen embargo date dd/mm/yyyy	Application reference(s) (e.g. EP123456)	Subject or title of application	Applicant (s) (as on the application)

[5] A drop down list allows choosing the type of IP rights: Patents, Trademarks, Registered designs, Utility models, Others.

Part B2

Please complete the table hereafter:

Type of Exploitable Foreground[6]	Description of exploitable foreground	Confidential	Foreseen embargo date dd/mm/yyyy	Exploitable product(s) or measure(s)	Sector(s) of application[7]	Timetable, commercial or any other use	Patents or other IPR exploitation (licences)	Owner & Other Beneficiary(s) involved
General advancement of knowledge	Extrae: Instrumentation and measurement package for Paraver	NO		Open-source software	J58.2.9 - Other software publishing		Open source	BSC
General advancement of knowledge	OTF1 to Paraver and OTF2 to Paraver converters	NO		Open-source software	J58.2.9 - Other software publishing		Open source	BSC
General advancement of knowledge	Paraver: Trace-based performance analysis and visualization tool	NO		Open-source software	J58.2.9 - Other software publishing		Open source	BSC
General advancement of knowledge	Dimemas: Performance modelling and prediction package	NO		Open-source software	J58.2.9 - Other software publishing		Open source	BSC

[19] A drop down list allows choosing the type of foreground: General advancement of knowledge, Commercial exploitation of R&D results, Exploitation of R&D results via standards, exploitation of results through EU policies, exploitation of results through (social) innovation.

[7] A drop down list allows choosing the type sector (NACE nomenclature) : http://ec.europa.eu/competition/mergers/cases/index/nace_all.html

Type of Exploitable Foreground[6]	Description of exploitable foreground	Confidential	Foreseen embargo date dd/mm/yyyy	Exploitable product(s) or measure(s)	Sector(s) of application[7]	Timetable, commercial or any other use	Patents or other IPR exploitation (licences)	Owner & Other Beneficiary(s) involved
General advancement of knowledge	Scalasca: Performance Analysis toolset	NO		Open-source software	J58.2.9 - Other software publishing		Open source	GRS, JUELICH
General advancement of knowledge	Cube3 and Cube4: Performance Data Visualizer	NO		Open-source software	J58.2.9 - Other software publishing		Open source	GRS, JUELICH
General advancement of knowledge	Opari2: OpenMP instrumentation package	NO		Open-source software	J58.2.9 - Other software publishing		Open source	GRS, JUELICH
General advancement of knowledge	Score-P: Instrumentation and measurement package for Vampir, Scalasca, TAU and Periscope	NO		Open-source software	J58.2.9 - Other software publishing		Open source	GRS, JUELICH, TUD
General advancement of knowledge	OTF1 and OTF2: event trace format libraries	NO		Open-source software	J58.2.9 - Other software publishing		Open source	TUD
General advancement of knowledge	VampirTrace: Instrumentation and measurement package for Vampir	NO		Open-source software	J58.2.9 - Other software publishing		Open source	TUD
Commercial exploitation of R&D results	Vampir: Trace-based performance analysis and visualization package	NO		Software product	J58.2.9 - Other software publishing		Commercial license	TUD

Type of Exploitable Foreground[6]	Description of exploitable foreground	Confidential	Foreseen embargo date dd/mm/yyyy	Exploitable product(s) or measure(s)	Sector(s) of application[7]	Timetable, commercial or any other use	Patents or other IPR exploitation (licences)	Owner & Other Beneficiary(s) involved
Commercial exploitation of R&D results	ThreadSpotter	NO		Software product	J58.2.9 - Other software publishing		Commercial license	RW

A more detailed description of the software packages can be found in Section 1.3.2.

3. Report on societal implications

Replies to the following questions will assist the Commission to obtain statistics and indicators on societal and socio-economic issues addressed by projects. The questions are arranged in a number of key themes. As well as producing certain statistics, the replies will also help identify those projects that have shown a real engagement with wider societal issues, and thereby identify interesting approaches to these issues and best practices. The replies for individual projects will not be made public.

A General Information *(completed automatically when **Grant Agreement number** is entered.*

Grant Agreement Number:	277463
Title of Project:	Holistic Performance System Analysis-EU
Name and Title of Coordinator:	Dr.-Ing. Bernd Mohr

B Ethics

1. Did your project undergo an Ethics Review (and/or Screening)? • If Yes: have you described the progress of compliance with the relevant Ethics Review/Screening Requirements in the frame of the periodic/final project reports? Special Reminder: the progress of compliance with the Ethics Review/Screening Requirements should be described in the Period/Final Project Reports under the Section 3.2.2 *'Work Progress and Achievements'*	***No***
2. Please indicate whether your project involved any of the following issues (tick box) :	***NO***
RESEARCH ON HUMANS	
• Did the project involve children?	
• Did the project involve patients?	
• Did the project involve persons not able to give consent?	
• Did the project involve adult healthy volunteers?	
• Did the project involve Human genetic material?	
• Did the project involve Human biological samples?	
• Did the project involve Human data collection?	
RESEARCH ON HUMAN EMBRYO/FOETUS	
• Did the project involve Human Embryos?	
• Did the project involve Human Foetal Tissue / Cells?	
• Did the project involve Human Embryonic Stem Cells (hESCs)?	
• Did the project on human Embryonic Stem Cells involve cells in culture?	
• Did the project on human Embryonic Stem Cells involve the derivation of cells from Embryos?	
PRIVACY	
• Did the project involve processing of genetic information or personal data (eg. health, sexual lifestyle, ethnicity, political opinion, religious or philosophical conviction)?	
• Did the project involve tracking the location or observation of people?	
RESEARCH ON ANIMALS	
• Did the project involve research on animals?	
• Were those animals transgenic small laboratory animals?	
• Were those animals transgenic farm animals?	
• Were those animals cloned farm animals?	
• Were those animals non-human primates?	
RESEARCH INVOLVING DEVELOPING COUNTRIES	
• Did the project involve the use of local resources (genetic, animal, plant etc)?	

• Was the project of benefit to local community (capacity building, access to healthcare, education etc)?	
DUAL USE	
• Research having direct military use	No
• Research having the potential for terrorist abuse	No

C Workforce Statistics

3. Workforce statistics for the project: Please indicate in the table below the number of people who worked on the project (on a headcount basis).

Type of Position	**Number of Women**	**Number of Men**
Scientific Coordinator		1
Work package leaders		3
Experienced researchers (i.e. PhD holders)	2	3
PhD Students		11
Other	1	11

4. How many additional researchers (in companies and universities) were recruited specifically for this project?	
Of which, indicate the number of men:	

D Gender Aspects

5. Did you carry out specific Gender Equality Actions under the project?

○ Yes
X No

6. Which of the following actions did you carry out and how effective were they?

	Action	Not at all effective → Very effective
❑	Design and implement an equal opportunity policy	○ ○ ○ ○ ○
❑	Set targets to achieve a gender balance in the workforce	○ ○ ○ ○ ○
❑	Organise conferences and workshops on gender	○ ○ ○ ○ ○
❑	Actions to improve work-life balance	○ ○ ○ ○ ○
○	Other:	

7. Was there a gender dimension associated with the research content – i.e. wherever people were the focus of the research as, for example, consumers, users, patients or in trials, was the issue of gender considered and addressed?

○ Yes- please specify

X No

E Synergies with Science Education

8. Did your project involve working with students and/or school pupils (e.g. open days, participation in science festivals and events, prizes/competitions or joint projects)?

○ Yes- please specify

X No

9. Did the project generate any science education material (e.g. kits, websites, explanatory booklets, DVDs)?

○ Yes- please specify

X No

F Interdisciplinarity

10. Which disciplines (see list below) are involved in your project?

X Main discipline[8]: 1.1

○ Associated discipline[8]: ○ Associated discipline[8]:

G Engaging with Civil society and policy makers

[8] Insert number from list below (Frascati Manual).

11a Did your project engage with societal actors beyond the research community? *(if 'No', go to Question 14)*	○ X	Yes No

11b If yes, did you engage with citizens (citizens' panels / juries) or organised civil society (NGOs, patients' groups etc.)?

- ○ No
- ○ Yes- in determining what research should be performed
- ○ Yes - in implementing the research
- ○ Yes, in communicating /disseminating / using the results of the project

11c In doing so, did your project involve actors whose role is mainly to organise the dialogue with citizens and organised civil society (e.g. professional mediator; communication company, science museums)?	○ ○	Yes No

12. Did you engage with government / public bodies or policy makers (including international organisations)

- ○ No
- ○ Yes- in framing the research agenda
- ○ Yes - in implementing the research agenda
- ○ Yes, in communicating /disseminating / using the results of the project

13a Will the project generate outputs (expertise or scientific advice) which could be used by policy makers?

- ○ Yes – as a **primary** objective (please indicate areas below- multiple answers possible)
- ○ Yes – as a **secondary** objective (please indicate areas below - multiple answer possible)
- ○ No

13b If Yes, in which fields?

Agriculture		Energy		Human rights	
Audiovisual and Media		Enlargement		Information Society	
Budget		Enterprise		Institutional affairs	
Competition		Environment		Internal Market	
Consumers		External Relations		Justice, freedom and security	
Culture		External Trade		Public Health	
Customs		Fisheries and Maritime Affairs		Regional Policy	
Development Economic and Monetary Affairs		Food Safety		Research and Innovation	
Education, Training, Youth		Foreign and Security Policy		Space	
Employment and Social Affairs		Fraud		Taxation	
		Humanitarian aid		Transport	

13c If Yes, at which level?

- ○ Local / regional levels
- ○ National level
- ○ European level
- ○ International level

H Use and dissemination

14. How many Articles were published/accepted for publication in peer-reviewed journals?	**17**
To how many of these is open access[9] provided?	**1**
How many of these are published in open access journals?	
How many of these are published in open repositories?	**1**
To how many of these is open access not provided?	**16**
Please check all applicable reasons for not providing open access:	
X publisher's licensing agreement would not permit publishing in a repository ❑ no suitable repository available ❑ no suitable open access journal available ❑ no funds available to publish in an open access journal ❑ lack of time and resources ❑ lack of information on open access ❑ other[10]: ……………	

15. How many new patent applications ('priority filings') have been made? *("Technologically unique": multiple applications for the same invention in different jurisdictions should be counted as just one application of grant).*	**0**

16. Indicate how many of the following Intellectual Property Rights were applied for (give number in each box).	Trademark	**0**
	Registered design	**0**
	Other	**0**

17. How many spin-off companies were created / are planned as a direct result of the project?	**0**
Indicate the approximate number of additional jobs in these companies:	

18. Please indicate whether your project has a potential impact on employment, in comparison with the situation before your project:

X	Increase in employment, or	❑	In small & medium-sized enterprises

[9] Open Access is defined as free of charge access for anyone via Internet.

[10] For instance: classification for security project.

X	Safeguard employment, or	X	In large companies
❑	Decrease in employment,	❑	None of the above / not relevant to the project
❑	Difficult to estimate / not possible to quantify		

19. For your project partnership please estimate the employment effect resulting directly from your participation in Full Time Equivalent (*FTE = one person working fulltime for a year*) jobs: Difficult to estimate / not possible to quantify	*Indicate figure:* *7 FTE increase in employment* *+* *9 FTE safeguard employment* ❑

I Media and Communication to the general public

20. As part of the project, were any of the beneficiaries professionals in communication or media relations?

○ Yes X No

21. As part of the project, have any beneficiaries received professional media / communication training / advice to improve communication with the general public?

○ Yes X No

22 Which of the following have been used to communicate information about your project to the general public, or have resulted from your project?

X	Press Release	X	Coverage in specialist press
❑	Media briefing	X	Coverage in general (non-specialist) press
❑	TV coverage / report	❑	Coverage in national press
❑	Radio coverage / report	❑	Coverage in international press
X	Brochures /posters / flyers	X	Website for the general public / internet
❑	DVD /Film /Multimedia	❑	Event targeting general public (festival, conference, exhibition, science café)

23 In which languages are the information products for the general public produced?

X	Language of the coordinator (German)	X	English
X	Other language(s)	X	Russian

Question F-10**:** Classification of Scientific Disciplines according to the Frascati Manual 2002 (Proposed Standard Practice for Surveys on Research and Experimental Development, OECD 2002):

FIELDS OF SCIENCE AND TECHNOLOGY

1. NATURAL SCIENCES

1.1 Mathematics and computer sciences [mathematics and other allied fields: computer sciences and other allied subjects (software development only; hardware development should be classified in the engineering fields)]

1.2 Physical sciences (astronomy and space sciences, physics and other allied subjects)

1.3 Chemical sciences (chemistry, other allied subjects)

1.4 Earth and related environmental sciences (geology, geophysics, mineralogy, physical geography and other geosciences, meteorology and other atmospheric sciences including climatic research, oceanography, vulcanology, palaeoecology, other allied sciences)

1.5 Biological sciences (biology, botany, bacteriology, microbiology, zoology, entomology, genetics, biochemistry, biophysics, other allied sciences, excluding clinical and veterinary sciences)

2 ENGINEERING AND TECHNOLOGY

2.1 Civil engineering (architecture engineering, building science and engineering, construction engineering, municipal and structural engineering and other allied subjects)

2.2 Electrical engineering, electronics [electrical engineering, electronics, communication engineering and systems, computer engineering (hardware only) and other allied subjects]

2.3. Other engineering sciences (such as chemical, aeronautical and space, mechanical, metallurgical and materials engineering, and their specialised subdivisions; forest products; applied sciences such as geodesy, industrial chemistry, etc.; the science and technology of food production; specialised technologies of interdisciplinary fields, e.g. systems analysis, metallurgy, mining, textile technology and other applied subjects)

3. MEDICAL SCIENCES

3.1 Basic medicine (anatomy, cytology, physiology, genetics, pharmacy, pharmacology, toxicology, immunology and immunohaematology, clinical chemistry, clinical microbiology, pathology)

3.2 Clinical medicine (anaesthesiology, paediatrics, obstetrics and gynaecology, internal medicine, surgery, dentistry, neurology, psychiatry, radiology, therapeutics, otorhinolaryngology, ophthalmology)

3.3 Health sciences (public health services, social medicine, hygiene, nursing, epidemiology)

4. AGRICULTURAL SCIENCES

4.1 Agriculture, forestry, fisheries and allied sciences (agronomy, animal husbandry, fisheries, forestry, horticulture, other allied subjects)

4.2 Veterinary medicine

5. SOCIAL SCIENCES

5.1 Psychology

5.2 Economics

5.3 Educational sciences (education and training and other allied subjects)

5.4 Other social sciences [anthropology (social and cultural) and ethnology, demography, geography (human, economic and social), town and country planning, management, law, linguistics, political sciences, sociology, organisation and methods, miscellaneous social sciences and interdisciplinary , methodological and historical S1T activities relating to subjects in this group. Physical anthropology, physical geography and psychophysiology should normally be classified with the natural sciences].

6. HUMANITIES

6.1 History (history, prehistory and history, together with auxiliary historical disciplines such as archaeology, numismatics, palaeography, genealogy, etc.)

6.2 Languages and literature (ancient and modern)

6.3 Other humanities [philosophy (including the history of science and technology) arts, history of art, art criticism, painting, sculpture, musicology, dramatic art excluding artistic "research" of any kind, religion, theology, other fields and subjects pertaining to the humanities, methodological, historical and other S1T activities relating to the subjects in this group]

