Grant agreement for: Collaborative project*

Annex I - "Description of Work"

Project acronym: HOPSA-EU

Project full title: " HOlistic Performance System Analysis-EU "

Grant agreement no: 277463

Session submission date: 2011-03-28

Table of Contents

Part A

Workplan Tables

A1:
Project summary

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

One form per project

General information

Project title [3]	HOlistic Performance System Analysis-EU		
Starting date [4]	01/02/2011		
Duration in months [5]	24		
Call (part) identifier [6]	FP7-ICT-2011-EU-Russia		
Activity code(s) most relevant to your topic [7]	:		
Free keywords [8]		performance analysis tools for high-performance computing	

Abstract [9]

To maximise the scientific 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 optimising 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 presents serious challenges to achieving these goals.
The HOPSA project (HOListic Performance System Analysis) therefore sets out to create an integrated diagnostic infrastructure for combined application and system tuning - with the former being under EU and the latter being under Russian responsibility. 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 diagnostic tools. Developers can choose from a variety of mature performance-analysis tools developed by our consortium. Within this project, the tools will be further integrated and enhanced with respect to scalability, depth of analysis, and support for asynchronous tasking, a node-level paradigm playing an increasingly important role in hybrid programs on emerging hierarchical and heterogeneous systems.
Using our infrastructure, the scientific output rate of a system will be increased in three ways: First, the enhanced tool suite will lead to better optimisation results, expanding the potential of the codes to which they are applied. Second, integrating the tools into an automated diagnostic workflow will ensure that they are used both (i) more frequently and (ii) more effectively, further multiplying their benefit. Finally, our holistic approach will lead to a more targeted optimisation of the interactions between application and system.

A2:
List of Beneficiaries

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

List of Beneficiaries

No	Name	Short name	Country	Project entry month[10]	Project exit month
1	FORSCHUNGSZENTRUM JUELICH GMBH	JUELICH	Germany	1	24
2	ROGUE WAVE SOFTWARE AB	RW	Sweden	1	24
3	BARCELONA SUPERCOMPUTING CENTER - CENTRO NACIONAL DE SUPERCOMPUTACION	BSC	Spain	1	24
4	GERMAN RESEARCH SCHOOL FOR SIMULATION SCIENCES GMBH	GRS	Germany	1	24
5	TECHNISCHE UNIVERSITAET DRESDEN	TUD	Germany	1	24

A3:
Budget Breakdown

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

One Form per Project

Participant number in this project [11]	Participant short name	Fund. % [12]	Ind. costs [13]	Estimated eligible costs (whole duration of the project)					Requested EU contribution
				RTD / Innovation (A)	Demonstration (B)	Management (C)	Other (D)	Total A+B+C+D	
1	JUELICH	75.0	A	346,241.00	0.00	55,870.00	0.00	402,111.00	315,550.00
2	RW	50.0	F	328,500.00	0.00	24,300.00	0.00	352,800.00	188,550.00
3	BSC	75.0	A	362,619.00	0.00	20,552.00	0.00	383,171.00	292,516.00
4	GRS	75.0	T	364,424.00	0.00	27,752.00	0.00	392,176.00	301,070.00
5	TUD	75.0	T	370,400.00	0.00	24,480.00	0.00	394,880.00	302,280.00
Total				1,772,184.00	0.00	152,954.00	0.00	1,925,138.00	1,399,966.00

Note that the budget mentioned in this table is the total budget requested by the Beneficiary and associated Third Parties.

*** The following funding schemes are distinguished**

Collaborative Project (if a distinction is made in the call please state which type of Collaborative project is referred to: (i) Small of medium-scale focused research project, (ii) Large-scale integrating project, (iii) Project targeted to special groups such as SMEs and other smaller actors), Network of Excellence, Coordination Action, Support Action.

1. Project number

The project number has been assigned by the Commission as the unique identifier for your project, and it cannot be changed. The project number **should appear on each page of the grant agreement preparation documents** to prevent errors during its handling.

2. Project acronym

Use the project acronym as indicated in the submitted proposal. It cannot be changed, unless agreed during the negotiations. The same acronym **should appear on each page of the grant agreement preparation documents** to prevent errors during its handling.

3. Project title

Use the title (preferably no longer than 200 characters) as indicated in the submitted proposal. Minor corrections are possible if agreed during the preparation of the grant agreement.

4. Starting date

Unless a specific (fixed) starting date is duly justified and agreed upon during the preparation of the Grant Agreement, the project will start on the first day of the month following the entry info force of the Grant Agreement (NB : entry into force = signature by the Commission). Please note that if a fixed starting date is used, you will be required to provide a detailed justification on a separate note.

5. Duration

Insert the duration of the project in full months.

6. Call (part) identifier

The Call (part) identifier is the reference number given in the call or part of the call you were addressing, as indicated in the publication of the call in the Official Journal of the European Union. You have to use the identifier given by the Commission in the letter inviting to prepare the grant agreement.

7. Activity code

Select the activity code from the drop-down menu.

8. Free keywords

Use the free keywords from your original proposal; changes and additions are possible.

9. Abstract

10. The month at which the participant joined the consortium, month 1 marking the start date of the project, and all other start dates being relative to this start date.

11. The number allocated by the Consortium to the participant for this project.

12. Include the funding % for RTD/Innovation – either 50% or 75%

13. Indirect cost model

A: Actual Costs
S: Actual Costs Simplified Method
T: Transitional Flat rate
F :Flat Rate

Workplan Tables

Project number

277463

Project title

HOPSA-EU—HOlistic Performance System Analysis-EU

Call (part) identifier

FP7-ICT-2011-EU-Russia

Funding scheme

Collaborative project

WT1
List of work packages

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

LIST OF WORK PACKAGES (WP)

WP Number [53]	WP Title	Type of activity [54]	Lead beneficiary number [55]	Person-months [56]	Start month [57]	End month [58]
WP 1	Project Management	MGT	1	9.00	1	24
WP 2	HPC application-level performance analysis	RTD	5	106.00	1	23
WP 3	Integration of system and application performance analysis	RTD	4	76.00	1	24
			Total	191.00		

WT2:
List of Deliverables

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

List of Deliverables - to be submitted for review to EC

Delive-rable Number [61]	Deliverable Title	WP number [53]	Lead benefi-ciary number	Estimated indicative person-months	Nature [62]	Dissemi-nation level [63]	Delivery date [64]
D1.1	Intermediate Activity Report	1	1	1.00	R	PP	12
D1.2	Final Activity Report	1	1	1.00	R	PP	24
D2.1	Intermediate Tool Set	2	3	50.00	P	PU	12
D2.2	Final Tool Set	2	5	50.00	P	PU	21
D2.3	Tool Validation Report	2	5	6.00	R	PP	23
D3.1	API Requirements Report	3	4	5.00	R	PP	6
D3.2	Workflow Report	3	2	5.00	R	PU	15
D3.3	Light-weight Monitoring Module	3	4	24.00	P	PU	18
D3.4	UNITE Package	3	4	12.00	P	PU	22
			Total	154.00			

WT3:
Work package description

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

One form per Work Package

Work package number [53]	WP1	Type of activity [54]	MGT
Work package title	Project Management		
Start month	1		
End month	24		
Lead beneficiary number [55]	1		

Objectives

This work package deals with the overall project management. More specifically, it has the following objectives:
• Perform the overall project management of the project, including administrative and financial management as well as legal aspects of the project.
• Perform the technical coordination of the project.
• Monitor the progress of the partners, detect possible problems and perform risk management.
• Ensure the quality management and assurance.
• Ensure the correct flow of information between partners and with the European Commission.
• Synchronise the activities of the EU and Russian coordinated projects.

Description of work and role of partners

WP1 Description of work

This work package is decomposed into two tasks. Each of them is lead by the project coordinator with the support of the technical manager, the project management team, and with the assessment and input from the rest of the partners.

Task T1.1: Administrative and financial management.

Lead by the project coordinator, this task will establish the corresponding procedures, tools and methodologies to enable a correct project management. It will also coordinate the timely production of deliverables, organise the kick-off meeting and reviews, and organise and manage audits requested by the commission. This task will produce the necessary annual project reports. See Section 2.1 for more details on the management structure.

Task T1.2: Technical coordination.

Lead by the technical coordinator, this task will perform the technical coordination of the project by means of monitoring the progress of the work packages, technical coordination of the meetings, appointing reviewers to assess the quality of the deliverables before their delivery to the EC, and solving technical conflicts.

Person-Months per Participant

Participant number [10]	Participant short name [11]	Person-months per participant
1	JUELICH	5.00
2	RW	1.00
3	BSC	1.00
4	GRS	1.00
5	TUD	1.00
	Total	9.00

WT3:
Work package description

List of deliverables

Delive-rable Number [61]	Deliverable Title	Lead benefi-ciary number	Estimated indicative person-months	Nature [62]	Dissemi-nation level [63]	Delivery date [64]
D1.1	Intermediate Activity Report	1	1.00	R	PP	12
D1.2	Final Activity Report	1	1.00	R	PP	24
		Total	2.00			

Description of deliverables

D1.1) Intermediate Activity Report: This document will describe the project activities done since the beginning of the project to month 12. [month 12]

D1.2) Final Activity Report: This document will describe the project activities done month 13 to month 24. [month 24]

Schedule of relevant Milestones

Milestone number [59]	Milestone name	Lead benefi-ciary number	Delivery date from Annex I [60]	Comments

WT3:
Work package description

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

One form per Work Package

Work package number [53]	WP2	Type of activity [54]	RTD
Work package title	HPC application-level performance analysis		
Start month	1		
End month	23		
Lead beneficiary number [55]	5		

Objectives

The overall objective of work package 2 is to enhance and extend the already existing individual performance measurement and analysis tools (ThreadSpotter, Paraver, Dimemas, Scalasca, Vampir) of the project partners to make them fit for the analysis of 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 the regular tool products.

Description of work and role of partners

WP2 Description of work

Task T2.1: Enhancing functionality of the tools.

Develop and implement methods and algorithms to make the individual tools of the partners fit for the analysis of petascale computations and beyond:

• To increase the depth of the analysis offered by Scalasca, we will add new functionality to locate the root cause of wait states. The existing prototypical implementation of the delay analysis [4], which received the Best Paper Award of the ICPP Conference 2010, will be consolidated and further optimized with respect to runtime and memory consumption. In addition, the approach, which was originally developed for MPI only, will be extended to hybrid applications that use MPI and OpenMP in combination. (GRS, JSC) In the same way, Dimemas simulations can be used to perform the root cause analysis as the tool allows to analyse different what-if scenarios. The tool will be extended to give details on the selected scenario to an external tool (e.g., Scalasca) (BSC)

• Introduce a ”system background view” for Vampir. Besides the usual process-related events it will provide information on background activities which cannot be mapped to a single process/thread, but which are influencing all/several processes/threads in a node/partition/system. Examples are the network throughput or the storage system read/write rates. It will present performance counter values in fixed or dynamic intervals or newly defined events to the user. This requires extensions to the measurement system, the event trace format, and the visualisation displays. (TUD)

• Collect and analyse profile-based performance data for a shared-memory process, such as an MPI strand, based on timer interrupts. Design a new entry into ThreadSpotter based on this analysis and integrate with the existing issues-based (bandwidth, latency, inter-thread and pollution issues) and loop-based views. (RW)

Task T2.2: Enhancing scalability of the tools.

Develop and implement methods and algorithms to make the tools more scalable, i.e., that they can perform larger (in terms of number of cores) and longer (in terms of the length of execution) performance analysis experiments:

• To improve Scalasca’s scalability in terms of the number of cores and to reduce its runtime overhead, we will develop a more space-efficient distributed scheme to record MPI communicators. Avoiding rank translation at runtime will be a key requirement. After successfully testing a reference implementation for the native Scalasca

measurement system, the new scheme will be transferred to the SILC measurement system so that it can also benefit Vampir. (GRS, JSC)

• To improve Scalasca's scalability along the time axis, we will integrate a new method for the semantic runtime compression of time-series profiles [31], which so far exists only as an offline prototype. Finally, the new approach, which so far only supports MPI, will be extended towards hybrid MPI/OpenMP codes. Again, after successfully testing a reference implementation for the native Scalasca measurement system, the new scheme will be transferred to the SILC measurement system. Together, these changes will enable the detailed analysis of time-dependent performance behavior even for long-running experiments. (GRS, JSC)

• Extend the Open Trace Format for profile snapshots and restart points. The profile snapshots will allow a coarse summary about the performance behaviour of predetermined time intervals. The restart points need to provide all status information required to start reading trace events from this point without the preceding events. (TUD in cooperation with previous task)

• Based on the previous task, Vampir will be extended to allow partial loading and visualisation in the time and space (processes/threads) dimensions. Based on profile snapshots, a pre-selection of time intervals and processes will be offered to the user. (TUD)

• Extend the run-time measurement for selective trace recording for Vampir with respect to time and location (processes/threads). The selection may be specified in fixed form before the trace recording or by adding control statements within the target application similar to phase markers. (TUD)

• Introduce a long-term event-trace recording mode to the SILC monitoring component. It will allow to discard the preceding section of the event trace at certain control points or phase markers. The live decision whether to keep or discard a section can depend on the presence or absence of certain behaviour patterns as well as on similarity or difference with other sections. (TUD)

• Design a scalable method for collecting ThreadSpotter's performance fingerprint data from each of the MPI strands running in a scalable system. This involves designing a filtering function that identifies MPI strands with similar performance characteristics, based on the fingerprint and the new profile-based data. That way, a user will not have to wade through all performance data collected from 1000s of strands and can concentrate on the unique behaviour. (RW)

• Increase the scalability of CEPBA-Tools by intelligently deciding which information is emitted. This work will be based on the current on-line analysis implemented within the Extrae instrumentation library. The scalability would also be enhanced by parallelising the Paraver kernel and improving the file management. (BSC)

• Currently, the Dimemas simulator only scales up to few thousand processes and it is necessary to extend its scalability to be used at large scale. This implies the redesign of some internal structures of the simulator and the reimplementation of the module that generate the output trace. (BSC)

Task T2.3: Tool integration.

Design and implement methods to integrate the single tools from the different partners so that they can be more easily used together for the analysis of HPC simulation programs:

• Integration of Scalasca's interactive report explorer on the one side with Vampir or Paraver on the other side to allow detailed investigation of the most severe instances of performance problems located by the Scalasca analysis with the comprehensive statistical and visualisation features of Vampir and Paraver. A demonstration prototype of this has been implemented based on KOJAK (a sequentially-working predecessor of Scalasca) and an older version of Vampir (based on Motif). The analysis/location part has to be parallelised and integrated into the parallel Scalasca trace analyser. The remote-trace-browsing control part has to be re-implemented for the latest version of Vampir (based on Qt) and Paraver. The implemented tool interaction protocols (DBUS, UNIX signals) will be enhanced. (BSC, JSC, TUD)

• Extend CEPBA-Tools to work with Open Trace Format (OTF) [18] traces. The objective is to allow Paraver to load and work with OTF traces and to implement a new version of Extrae that supports the generation of the traces in OTF format. (BSC, TUD)

• 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 so that analysis results obtained with one tool can be interpreted in the light of results obtained with another. This is an important prerequisite for assigning individual tools their role in an analysis workflow capable of tracking down inefficiencies related to asynchronous tasking. Key elements of this unification will be a common terminology as well as common methaphors for representing analysis results to the user. (All-EU)

• Paraver and Dimemas are integrated at the level of traces: Paraver traces can be translated to Dimemas and Dimemas simulation can produce a Paraver trace as part of its output. We will extend this integration to allow that Paraver calls Dimemas to analyse what-if scenarios producing new Paraver traces that can be loaded on the visualizer. (BSC)

• Investigate integration between timeline-based visualizers (Paraver, Vampir) and/or Scalasca and ThreadSpotter. For example, use ThreadSpotter's identification of MPI strands with unique behaviour to superimpose regularity/irregularity information on Paraver/Vampir displays. Alternatively, reduce the Paraver/Vampir displays to show only one representative per behaviour group. (RW, BSC, TUD).

Task T2.4: Tool validation.

The performance tools of project (ThreadSpotter, Paraver, Dimemas, Scalasca, Vampir) are already in daily use on the production systems of the computing centres of the partners (BSC, JSC, ZIH) or by customers of Rogue Wave where they will be applied to application codes of computing centre users. This will allow to validate whether the proposed enhancements of work package 2 (described by Tasks T2.1 to T2.3) are working and are useful in the analysis of real-world applications. These use cases will be documented. In addition, we will apply the tools to a set of benchmark codes (e.g. from the SPEC MPI, NAS, ASCI benchmark suites) at the begin and end of the project to investigate and report the improved efficiency and scalability of the tools. (All-EU)

Person-Months per Participant

Participant number [10]	Participant short name [11]	Person-months per participant
1	JUELICH	19.00
2	RW	18.00
3	BSC	31.00
4	GRS	18.00
5	TUD	20.00
	Total	106.00

List of deliverables

Deliverable Number [61]	Deliverable Title	Lead beneficiary number	Estimated indicative person-months	Nature [62]	Dissemination level [63]	Delivery date [64]
D2.1	Intermediate Tool Set	3	50.00	P	PU	12
D2.2	Final Tool Set	5	50.00	P	PU	21
D2.3	Tool Validation Report	5	6.00	R	PP	23
		Total	106.00			

Description of deliverables

D2.1) Intermediate Tool Set: Partially integrated version of the partner's measurement and analysis software with prototype implementation of the functionality and scalability enhancements. [month 12]

D2.2) Final Tool Set: Fully integrated version of the partner's measurement and analysis software with completed implementation of the functionality and scalability enhancements. [month 21]

D2.3) Tool Validation Report: Report on documented use cases where enhancements of the tools facilited the performance analysis of real-world applications. Will also document enhancements in the efficiency and scalability of the tools. [month 23]

Schedule of relevant Milestones

Milestone number [59]	Milestone name	Lead benefi-ciary number	Delivery date from Annex I [60]	Comments
MS1	Intermediate Tool Set	5	12	D2.1
MS2	Final Tool Set integrated as UNITE package	5	23	D2.2, D2.3, D3.4

WT3:
Work package description

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

One form per Work Package

Work package number [53]	WP3	Type of activity [54]	RTD
Work package title	Integration of system and application performance analysis		
Start month	1		
End month	24		
Lead beneficiary number [55]	4		

Objectives

The objective of work package 3 is to combine and integrate the work done for the HPC system-level performance analysis (by RU-Topic1 in Russia) and for application-level performance analysis (by WP2 in the EU) into a coherent and holistic performance analysis environment (see Figure 1 of Section B1.1). It will provide
• Low-overhead end-to-end performance analysis for all jobs running on a given system from their submission to their completion.
• Identification of key performance issues and notification of the user and system performance database after job completion.
• Detailed scalable performance analysis for petascale applications based on well-accepted and robust performance measurement and analysis tools.

Description of work and role of partners

WP3 Description of work

As can be seen in Figure 2 of Section B1.3.1, the integration between RU-Topic1 (system-level performance analysis) and WP2 (application-level performance analysis) is achieved on four different levels: by a low-overhead monitoring module getting overall performance data for every parallel job in the system (Task T3.3), by providing relevant system data to high-level application analysis (Task T3.1), by defining an overall performance analysis workflow (Task T3.2) and finally providing all software in one common package (Task T3.4).

Task T3.1: Definition of the interface between system-level and application-level performance analysis.

Define an interface to interchange performance related results between the system level, job level, and low-level application analysis on the one hand and the high-level performance tools on the other hand.

Part of this interface will be a performance report which compiles essential information from system-level monitoring and application-centric measurement into a performance repor document. This document will give an overview about the essential performance properties, including hints about potential performance deficiencies and how to verify their presence with other tools. Furthermore, it should allow to compare the performance behaviour with past reports of the same application to survey the success of analysis and tuning. Another part of the interface is to specify how performance related results of the system-, job-, and low-level application analysis is made available to the high-level performance tools. (All-EU and All-RU)

Task T3.2: Definition of the overall performance analysis workflow.

Definition of an HOPSA overall performance tool process and workflow which guides the application developers in the process of tuning and optimising their codes for performance in the form of a written user guide. The guide will give an overview of all tools available for the performance analysis of user applications and will describe which tools and in which order they should be used to accomplish specific common typical performance analysis tasks; also taking into account the results of experiments already performed (i.e. historic data) (All-EU and All-RU)

Task T3.3: Light-weight monitoring module.

A 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 that do not require expensive instrumentation. (GRS, JSC, TUD and All-RU)

Task T3.4: Unified download, configuration, build and installation package.

High-performance clusters often provide multiple MPI libraries and compiler suites for parallel programming. This means that parallel programming tools which often depend on a specific MPI library, and sometimes on a specific compiler, need to be installed multiple times, once for each combination of MPI library and compiler which has to be supported. In addition, over time, newer versions of the tools also get released and installed. One way to manage many different versions of software packages used by many computing centres all over the world is the ”module” software (see http://www.modules.org). However, each centre provides a different set of tools, has a different policy on how and where to install different software packages, and how to name the different versions. Our proposal ”UNiform Integrated Tool Environment” (UNITE) will improve this situation for debugging and performance tools by
• specifying exactly how and where to install the different versions of tool software packages (including integrating the tools to the maximum possible degree),
• defining standard module names for tools and their different versions, and
• supplying predefined module files which provide standardised, well-tested tool configurations,
• but still being flexible enough to be able to coexist with site-local installations, restrictions, and policies.

In addition, a ”meta”-Installation tool will be developed capable of configuring, building, and installing all HOPSA tools as a common package but hiding tool-specific aspects of the various phases. This work will be based on a successful prototype which can handle Scalasca and Vampir as well as a few other open-source tools developed as part of the EU ITEA-2 project
ParMA. (JSC in cooperation with All-EU and All-RU)

Person-Months per Participant

Participant number [10]	Participant short name [11]	Person-months per participant
1	JUELICH	18.00
2	RW	9.00
3	BSC	10.00
4	GRS	19.00
5	TUD	20.00
	Total	76.00

List of deliverables

Deliverable Number [61]	Deliverable Title	Lead beneficiary number	Estimated indicative person-months	Nature [62]	Dissemination level [63]	Delivery date [64]
D3.1	API Requirements Report	4	5.00	R	PP	6
D3.2	Workflow Report	2	5.00	R	PU	15
D3.3	Light-weight Monitoring Module	4	24.00	P	PU	18
D3.4	UNITE Package	4	12.00	P	PU	22
		Total	46.00			

Description of deliverables

WT3:
Work package description

D3.1) API Requirements Report: Report on the requirements for the interface between system-level and application level performance analysis. [month 6]

D3.2) Workflow Report: Report on the overall performance analysis workflow in the form of a user guide for application developers. [month 15]

D3.3) Light-weight Monitoring Module: Final version of software module. [month 18]

D3.4) UNITE Package: Unified package of all tools developed in the HOPSA project which allows to download, configure, build and install all tools at once and in a coherent way. [month 22]

Schedule of relevant Milestones

Milestone number [59]	Milestone name	Lead benefi-ciary number	Delivery date from Annex I [60]	Comments
MS1	Intermediate Tool Set	5	12	D2.1
MS2	Final Tool Set integrated as UNITE package	5	23	D2.2, D2.3, D3.4

WT4:
List of Milestones

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

List and Schedule of Milestones

Milestone number [59]	Milestone name	WP number [53]	Lead beneficiary number	Delivery date from Annex I [60]	Comments
MS1	Intermediate Tool Set	WP2, WP3	5	12	D2.1
MS2	Final Tool Set integrated as UNITE package	WP2, WP3	5	23	D2.2, D2.3, D3.4

WT5:
Tentative schedule of Project Reviews

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

Tentative schedule of Project Reviews

Review number [65]	Tentative timing	Planned venue of review	Comments, if any
RV 1	12	BSC, Barcelona, Spain	
RV 2	24	JSC, Jülich, Germany	

WT6:
Project Effort by Beneficiary and Work Package

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

Indicative efforts (man-months) per Beneficiary per Work Package

Beneficiary number and short-name	WP 1	WP 2	WP 3	Total per Beneficiary
1 - JUELICH	5.00	19.00	18.00	42.00
2 - RW	1.00	18.00	9.00	28.00
3 - BSC	1.00	31.00	10.00	42.00
4 - GRS	1.00	18.00	19.00	38.00
5 - TUD	1.00	20.00	20.00	41.00
Total	9.00	106.00	76.00	191.00

WT7:
Project Effort by Activity type per Beneficiary

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

Indicative efforts per Activity Type per Beneficiary

Activity type	Part. 1 JUELICH	Part. 2 RW	Part. 3 BSC	Part. 4 GRS	Part. 5 TUD	Total
1. RTD/Innovation activities						
WP 2	19.00	18.00	31.00	18.00	20.00	106.00
WP 3	18.00	9.00	10.00	19.00	20.00	76.00
Total Research	37.00	27.00	41.00	37.00	40.00	182.00
2. Demonstration activities						
Total Demo	0.00	0.00	0.00	0.00	0.00	0.00
3. Consortium Management activities						
WP 1	5.00	1.00	1.00	1.00	1.00	9.00
Total Management	5.00	1.00	1.00	1.00	1.00	9.00
4. Other activities						
Total other	0.00	0.00	0.00	0.00	0.00	0.00
Total	42.00	28.00	42.00	38.00	41.00	191.00

WT8:
Project Effort and costs

Project Number [1]	277463	Project Acronym [2]	HOPSA-EU

Project efforts and costs

Benefi-ciary number	Beneficiary short name	Estimated eligible costs (whole duration of the project)						Requested EU contribution (€)
		Effort (PM)	Personnel costs (€)	Subcontracting (€)	Other Direct costs (€)	Indirect costs OR lump sum, flat-rate or scale-of-unit (€)	Total costs	
1	JUELICH	42.00	264,642.00	0.00	21,000.00	116,469.00	402,111.00	315,550.00
2	RW	28.00	273,000.00	0.00	21,000.00	58,800.00	352,800.00	188,550.00
3	BSC	42.00	165,060.00	0.00	24,000.00	194,111.00	383,171.00	292,516.00
4	GRS	38.00	221,110.00	0.00	24,000.00	147,066.00	392,176.00	301,070.00
5	TUD	41.00	225,800.00	0.00	21,000.00	148,080.00	394,880.00	302,280.00
Total		191.00	1,149,612.00	0.00	111,000.00	664,526.00	1,925,138.00	1,399,966.00

1. Project number

The project number has been assigned by the Commission as the unique identifier for your project. It cannot be changed. The project number **should appear on each page of the grant agreement preparation documents (part A and part B)** to prevent errors during its handling.

2. Project acronym

Use the project acronym as given in the submitted proposal. It cannot be changed unless agreed so during the negotiations. The same acronym **should appear on each page of the grant agreement preparation documents (part A and part B)** to prevent errors during its handling.

53. Work Package number

Work package number: WP1, WP2, WP3, ..., WPn

54. Type of activity

For all FP7 projects each work package must relate to one (and only one) of the following possible types of activity (only if applicable for the chosen funding scheme – must correspond to the GPF Form Ax.v):

• **RTD/INNO =** Research and technological development including scientific coordination - applicable for Collaborative Projects and Networks of Excellence

• **DEM =** Demonstration - applicable for collaborative projects and Research for the Benefit of Specific Groups

• **MGT =** Management of the consortium - applicable for all funding schemes

• **OTHER =** Other specific activities, applicable for all funding schemes

• **COORD =** Coordination activities – applicable only for CAs

• **SUPP =** Support activities – applicable only for SAs

55. Lead beneficiary number

Number of the beneficiary leading the work in this work package.

56. Person-months per work package

The total number of person-months allocated to each work package.

57. Start month

Relative start date for the work in the specific work packages, month 1 marking the start date of the project, and all other start dates being relative to this start date.

58. End month

Relative end date, month 1 marking the start date of the project, and all end dates being relative to this start date.

59. Milestone number

Milestone number:MS1, MS2, …, MSn

60. Delivery date for Milestone

Month in which the milestone will be achieved. Month 1 marking the start date of the project, and all delivery dates being relative to this start date.

61. Deliverable number

Deliverable numbers in order of delivery dates: D1 – Dn

62. Nature

Please indicate the nature of the deliverable using one of the following codes

R = Report, **P** = Prototype, **D** = Demonstrator, **O** = Other

63. Dissemination level

Please indicate the dissemination level using one of the following codes:

• **PU =** Public

• **PP =** Restricted to other programme participants (including the Commission Services)

• **RE =** Restricted to a group specified by the consortium (including the Commission Services)

• **CO =** Confidential, only for members of the consortium (including the Commission Services)

• **Restreint UE =** Classified with the classification level "Restreint UE" according to Commission Decision 2001/844 and amendments

• **Confidentiel UE =** Classified with the mention of the classification level "Confidentiel UE" according to Commission Decision 2001/844 and amendments

• **Secret UE =** Classified with the mention of the classification level "Secret UE" according to Commission Decision 2001/844 and amendments

64. Delivery date for Deliverable

Month in which the deliverables will be available. Month 1 marking the start date of the project, and all delivery dates being relative to this start date

65. Review number

Review number: RV1, RV2, ..., RVn

66. Tentative timing of reviews

Month after which the review will take place. Month 1 marking the start date of the project, and all delivery dates being relative to this start date.

67. Person-months per Deliverable

The total number of person-month allocated to each deliverable.

PART B
COLLABORATIVE PROJECT

Contents

B1 Concept and objectives, progress beyond state-of-the-art, S/T methodology and work plan

B1.1 Concept and project objective(s)

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 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.

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 multi-disciplinary 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.

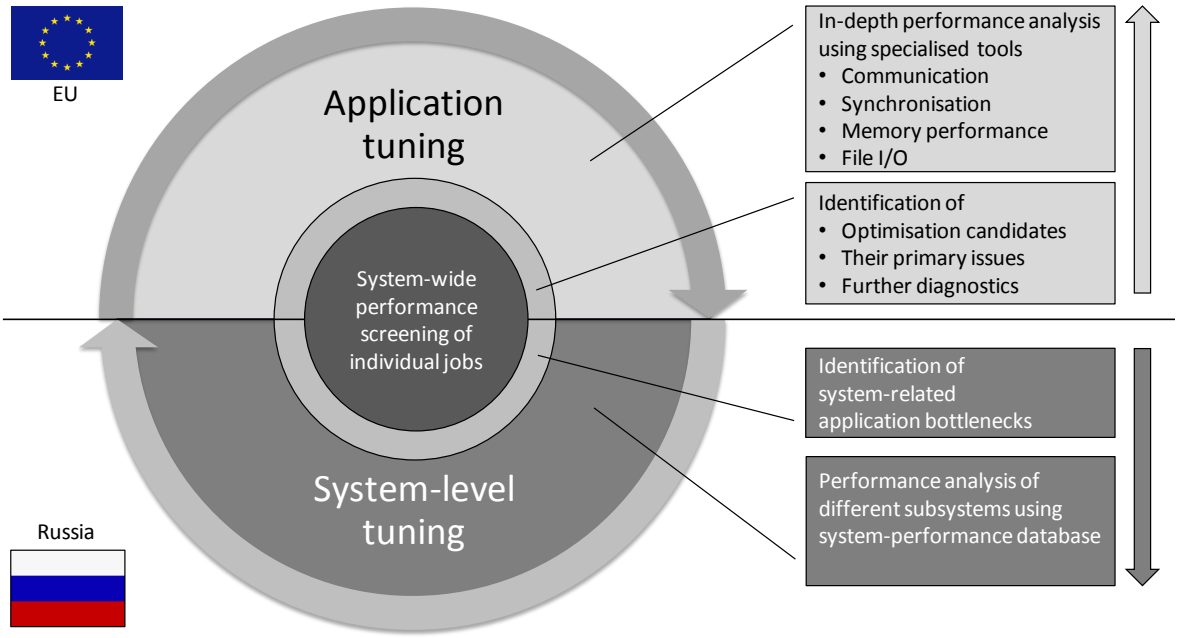

Figure 1: Holistic system performance analysis. While the Russian partners will focus on system-level application tuning (bottom), the EU partners will focus on user-level application tuning (top). System-wide performance screening of individual jobs (centre) at the interface between the two parts is common to both consortia.

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 proposal. 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 that 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 report with the following information:

- Time in queue and reasons why it was not executed earlier.
- System events, e.g., periods of heavy message or I/O traffic during job execution, which might have been adversely influencing its performance.
- Suboptimal performance aspects with respect to memory access, computation, or messaging.
- Potentially suggestions on which diagnostic tools to use for a follow-up analysis including instructions on how to use them.

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. A more detailed description of their functionality can be found in Section B1.2. 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 ar 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 partipant in the H4H proposal, could not participate in the project due to funding issues with the Spanish goverment.

In the context of H4H, the SILC 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 visualise high-level parallel programming constructs. Rogue Wave will extend its capability to optimise 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 (see Task T3.4). 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 (Task T3.1) and in the definition of the overall performance analysis workflow (Task T3.2).

In summary, all resources and development work necessary for supporting performance analysis of programming models for heterogenous 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. For some more information on the H4H, SILC, and TEXT projects see also the subsection ”Related projects” in the next section.

B1.2 Progress beyond the state-of-the-art

State-of-the-art

Developers of parallel applications can choose from a variety of performance-analysis tools, often with overlapping functionality but still following distinctive approaches and pursuing different strategies on how to address today’s demand for scalable performance solutions. From the user’s perspective, the tool landscape can be broadly classified according to the depth of analysis a tool provides. Some merely hint at performance phenomena, while others try to identify their root cause. To clarify the difference, Führlinger et al. use motor vehicles as an analogy [10]. Most modern cars feature warning lights that indicate the occurrence of a problem and prompt the vehicle owner to consult a mechanic. In a next step, the mechanic applies more sophisticated diagnostic tools to uncover the problem’s origin. While usage of the mechanic’s toolbox promises better insights, it typically requires also more effort and expertise.

Among performance tools, IPM2 [10] and Perfsuite [19] clearly fall into the warning-light category. With their simplicity and ease of use, they serve as a convenient entry point to more elaborate analyses. Both tools collect very basic performance metrics related to computation, messaging, and

I/O. As their underlying measurement technology, they either use sampling or direct instrumentation. While sampling can easily control measurement dilation by adjusting the sampling frequency, its statistical nature together with the difficulty of accessing parameters of a sampled event make it unsuitable for obtaining certain communication metrics such as the size of message payloads. Direct instrumentation via PMPI interposition wrappers [8], which is preferable for capturing message-passing events, can dilate measurements more than desired, an effect that might become worse under extreme strong scaling when the messaging frequency rises beyond a certain threshold. A primary research objective of this proposal in the context of the light-weight measurement module will be to investigate how these two methods can be effectively controlled and/or combined to ensure that the overhead never exceeds an acceptable limit, which would compromise its mandatory use in production. Both tools already address hybrid programming by supporting the combination of MPI with OpenMP: Whereas Perfsuite supports POSIX Threads as the threading library underneath most OpenMP implementations, making it unaware of OpenMP constructs, IPM2 leverages the OPARI source-code translation framework [20] which instruments OpenMP constructs as they are seen by the programmer. In this project, we plan to offer a light-weight solution for different combinations of MPI, OpenMP, and accelerator devices.

Over the years, numerous performance-analysis tools with more advanced diagnostic capabilities have been created for the toolbox of the "mechanic". Unfortunately, almost all of them support either only a single programming model or only a single platform, or they are restricted in both ways. For example, the profilers mpiP [32], FPMPI2 [2], and ompP [9], all three situated on the middle ground between warning light and expert tool, support either MPI or OpenMP but not the two in combination. Moreover, Cray Apprentice2 [5] works only on Cray XT systems. Finally, the two proprietary MPI tools IBM HPCT [4] and Intel Trace Collector and Analyzer [14] only support their vendor's platforms. Very few approaches support more than one programming model – typically the standards MPI, OpenMP, and the combination of both – and are at the same time available for a wide range of current computer systems. In addition to several tools our consortium contributes to this project, which are described in more detail below, this class includes, for example, TAU [29], a profiling and tracing toolset, and HPCToolkit [1], a statistical profiler that can be applied to multi-threaded MPI codes, although without specific support for OpenMP constructs.

As multicore systems increasingly adopt heterogeneous designs, where general purpose processors are complemented by more specialised ones to accelerate certain operations via customised hardware, performance analysts face the question of what to measure and how to measure it. The former question targets the selection of events to be observed, whereas the latter question emphasises the difficulty of instrumenting code in and extracting performance data from restricted environments that, for example, provide only very little memory that must be shared between measurement and application code. For heterogeneous platforms based on the Cell B.E. and for GPGPU accelerators, there are only few tools with early support for those alternative parallel programming environments, for example TAU and VampirTrace. Likewise, asynchronous tasking, as it is employed by evolving and emerging programming models such as OpenMP and StarSs [26] – either on homogeneous or heterogeneous multiprocessors – poses also significant challenges for performance analysis: The additional dimension of parallelism represented by tasks breaks traditional design patterns of tools committed to the classic fork-join execution model.

While the above-mentioned tools target mostly communication and synchronisation issues prevalent in many message-passing and multi-threading programs, delays in the memory subsystem present another important source of inefficiency. To support memory optimisation, Callgrind [33], a cache simulator based on the instrumentation framework Valgrind [22], attributes critical memory events to individual call paths. The substantial runtime overhead introduced by this type of instrumentation is

avoided in the memory analysis tool ThreadSpotter [27], another software package developed by our consortium.

Although many of the tools above impress with their list of features, it is not uncommon that extreme scales prevent the user from reaching the point where these features can become effective in the first place. For example, although Scalasca [11], one of the most scalable tools, available today has mastered performance experiments on a complete 72-rack Blue Gene/P system with almost 300k cores, some of its components will need to be re-engineered before such scales can become common practice among its less experienced users. Moreover, while the data many tools produce help localise performance bottlenecks, their expressiveness is often insufficient to remove them without intensive reasoning on the side of the application developer. Finally, in our eyes one of the biggest shortcomings of today's tools landscape to be addressed in this project is the weak link between warning-light and expert tools on the one hand and between application-level and system-level performance analysis on the other hand. In particular, the lack of an at least partially automated diagnostic workflow that not only identifies performance problems but also routes this information to the right people (e.g., application developer vs. system provider) with instructions on how to proceed (e.g., which tool to use next) still leaves substantial room for improvement.

Prior work

The members of this consortium are leading developers of performance-analysis tools for high-performance computing applications. The following tools will be introduced into this project as preexisting software, forming the basis for the enhancements described in our work program. All of them already passed the prototypical state and are widely used in production.

ThreadSpotter. The ThreadSpotter performance optimization technology has been developed in the startup Acumem AB – a spin-out from research at Uppsala University in Sweden. Since the start, the focus has been on performance debugging tools that explains to a programmer what actions need to be taken to achieve optimal performance. While an ordinary binary is running in a production environment, this new performance debugger collects sparse information about its execution behaviour into a "fingerprint" file. Typically, only a couple of megabytes is needed to store the fingerprint data from several hours of real execution.

It should be noted that the information collected in the fingerprint file is architecturally independent, i.e., it correctly represents the access locality of the application at an abstract level. Based on this information, the cache performance of any size cache, any size cache line and several replacement policies can be estimated off-line. Actually, the curve showing the miss-rate as a function of cache size for the entire application, as well as per-loop and per-instruction is generated at a fraction of a second off-line based on this data.

While such curves could prove themselves useful for performance experts, the biggest strength of this technology goes far beyond that. ThreadSpotter's analysis technology also detects performance bugs in the applications, i.e. certain access patterns that result in a sub-optimal performance. ThreadSpotter organizes such performance bugs into four issue groups: bandwidth issues, latency issues, thread interaction issues and cache pollution issues. For each issue group, the individual performance bugs are sorted in a worst-first order and presented in a table form together with an ample of statistics. Clicking on one such issue takes you to the source code where the performance bug has been committed and opens up a window with more information guiding the programmer towards a more efficient alternative. This enables even non-experts to tune their code towards optimal performance.

Paraver. Paraver [7, 15] 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 on the visualizer, just to capture such data in a Paraver trace. The second pillar is that the metrics are not hardwired on the tool but 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 to display a huge number of metrics with the available data. To capture the experts 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 applications behaviour than most performance tools. CEPBA-Tools have been successfully used for the analysis of large-scale runs in the range of ten thousand processes. Today CEPBA-Tools are developed by BSC and are available for download under an LGPL open-source license.

Scalasca. Scalasca [11] 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 XT, but is also well suited for small- and medium-scale HPC platforms. The analysis identifies potential performance bottlenecks – in particular those concerning communication and synchronisation – and offers guidance in exploring their causes. Scalasca mainly targets scientific and engineering applications based on the programming interfaces MPI and OpenMP, including hybrid applications based on a combination of the two. The user of Scalasca can choose between two different analysis modes: (i) performance overview on the call-path level via runtime summarisation (aka profiling) and (ii) in-depth study of application behaviour via event tracing. A distinctive feature of Scalasca is its ability to identify wait states that occur, for example, as a result of load imbalance – even at very large scales. The software is installed at numerous sites in several countries and has been successfully used to optimise academic and industrial simulation codes. Scalasca, which is jointly developed by JUELICH and GRS, is available for download under the New BSD open-source license.

Vampir. Vampir ("Visualisation and Analysis of MPI Resources") is a very well-known event trace visualisation software which is available since 1996 as a commercial product [18]. It offers intuitive parallel event trace visualisation with many displays showing different aspects of the parallel performance behaviour. It provides interactive zooming and browsing to show either a broad overview or very small details. Together with the VampirTrace instrumentation and run-time measurement package it supports not only MPI parallel programs but also OpenMP threads, POSIX threads, the IBM Cell architecture, GPGPU computing with CUDA or OpenCL, and combinations of them.

All recent versions of Vampir support parallel trace data processing; furthermore a special analysis server allows to cope with very large traces: While the display component runs on the local desktop or laptop machine, the server component processes extensive event trace data sets remotely and in parallel on a part of an HPC system. By this means, Vampir is able to visualise traces with several thousand processes/threads and tens of gigabytes in size while still providing an interactive working experience. Vampir is available for all major HPC platforms, including common Linux/Unix systems as well as Windows HPC Server. Today, Vampir is developed by ZIH, TU Dresden and is commercially distributed by the university-owned company GWT TU Dresden GmbH.

Related projects

We plan to exploit synergies with a number of ongoing projects, whose goals and consortia overlap with ours.

SILC (2009 - 2011). Funded by the German Ministry of Education and Research (BMBF), the goal of this project is the design and implementation of a scalable and easy-to-use performance measurement infrastructure for supercomputing applications (MPI, OpenMP, and hybrid) as a basis for several existing performance-analysis tools including Vampir and Scalasca. Enhancements of Scalasca and Vampir at the level of the measurement system that are proposed in HOPSA will be applied to this infrastructure instead of the native code bases of the individual tools.

Common partners: GRS, JUELICH, TUD

PRIMA (2009 - 2012). Funded by the US Department of Energy, this project re-engineers core components of the two performance-analysis systems Scalasca and TAU for evolution to petascale and beyond. An important subgoal of PRIMA is to create an interface between TAU and the SILC measurement infrastructure, significantly enlarging its user base. Since HOPSA leverages this infrastructure, this will also mean a richer set of analyses becoming available to HOPSA users such as the TAU performance database [12].

Common partners: GRS, JUELICH

H4H (2010 - 2013). Funded under the ITEA-2 program by several national funding agencies, one goal of this project will be the extension of the SILC measurement infrastructure (see above) towards heterogeneous systems so that at the end a solution for combinations of MPI, OpenMP, and heterogeneous programming will be available to be leveraged in HOPSA. Based on these enhancements, H4H will also extend Scalasca and Vampir to support performance analysis of hybrid programs including accelerated sections based on the GPUSs/CellSs [25] and HMPP [3] high-level parallel programming systems. Finally, the ThreadSpotter technology will be extended to scale to larger systems and to be able to automatically filter out and highlight the important information to a programmer in an intuitive way. As almost all EU partners in HOPSA are also partners in H4H, interactions between these two projects will occur naturally.

Common partners: RW, JUELICH, TUD

VI-HPS (2007 - 2011). In the Virtual Institute - High Productivity Supercomputing, which is funded by the Helmholtz Association of German Research Centres, seven partners in Germany and the US are developing and integrating state-of-the-art programming tools for high-performance computing. Besides pure development, the virtual institute also offers training workshops with guided hands-on training in the effective use of the tools. Since this training program covers also the HOPSA tools Scalasca and Vampir, it will provide a powerful additional dissemination channel for our project results.

Common partners: GRS, JUELICH, TUD

PRACE (2008 - 2010). The FP7 initiative PRACE aims at building a coherent European supercomputing infrastructure. The results produced in HOPSA will benefit several PRACE sites including those managed by HOPSA partner institutions.

Common partners: BSC, JUELICH

TEXT (2010 - 2012). The EU FP7 funded project "Towards EXascale applicaTions" (TEXT) will try to demonstrate the benefits of the hybrid MPI/SMPSs programming model on a rich set of real-world applications. SMPSs [16], developed by Barcelona Supercomputing Center, is a new task-based programming model which can be seen as an extension to the latest OpenMP 3.0 standard. By taking data dependencies into account, the SMPSs runtime can schedule the tasks more efficiently which results in a more scalable execution of the application program. This also reduces the burden on the application programmer making this new programming model easier to use. Additionally, JUELICH will enhance its performance analysis toolset Scalasca to support measurement and analysis of hybrid MPI/SMPSs applications.

Common partners: BSC, JUELICH

UPMARC (2008 - 2018). UPMARC is a holistic multicore project funded by the Swedish Research Council (VR) at Uppsala University. It is taking a broad view of the multicore problems. Research activities span from tools and modelling of multicore execution, all the way to scheduling of work, language design and formal methods for verification of parallel programs. UPMARC is a direct result from some of the Uppsala research leading up to Acumem (now Rogue Wave Software AB), thus HOPSA will benefit from UPMARC activities. Professor Erik Hagersten is currently sharing his time between RW and Uppsala University, creating also shared people between the two projects.

Common partner: RW

Progress beyond the state-of-the-art

The progress beyond the state-of-the-art in this project will not so much lie in new or radically restructured performance tools for hybrid/heterogeneous programming, but rather in the more effective use and enhancement of of established tools that already provide support for hybrid programming to some degree, usually in the form of MPI combined with OpenMP. While some of the open gaps such as heterogeneous hybridisation will be closed in related projects, especially in H4H, others such as asynchronous tasking, a node-level paradigm playing an increasingly important role in hybrid programs on emerging hierarchical and heterogeneous systems, will be addressed here. In addition, this project also sets out to improve general characteristics of our tools including scalability and the depth of the analysis performed. The specific contributions of this project are as follows:

- A lightweight performance measurement module (i) that can be applied to applications following contemporary hybridisation approaches (MPI, OpenMP, accelerator) and (ii) whose overhead is so small that it can be applied without exception to all parallel jobs running on a given system.
- An integrated diagnostic workflow that routes an application through a chain of successively refined diagnostics, starting from the output of the light-weight measurement module and involving the tools supplied by our consortium. This measure will ensure that many more applications can reap the benefits of advanced performance-analysis technology. In addition to application-induced bottlenecks, the workflow may also point at system-related inefficiencies, in this case branching into the system-tuning realm covered by our Russian partners.
- An enhanced suite of production-quality tools (i) in support of state-of-the-art hybrid programming models including those based on asynchronous tasking and (ii) capable of delivering relevant insights into the formation of performance bottlenecks even at very large scales.

B1.3 S/T methodology and associated work plan

B1.3.1 Overall strategy and general description

Given the rather small number of partners (five) and the short duration (two years) of the project, we propose a rather simple structuring of the work plan of the project into three work packages (see also Table WT1):

WP1: Project management
This work package performs the technical coordination of the project, monitors the progress of the partners, detects possible problems and performs risk management, ensure the quality management and assurance, and synchronises the activities of the EU and Russian coordinated projects. It is decomposed into the tasks:

T1.1 Administrative and financial management
T1.2 Technical coordination

WP2: HPC application-level performance analysis
This contains all research and technical development which only involves EU partners. The overall objective of work package 2 is to enhance and extend the already existing individual performance measurement and analysis tools (ThreadSpotter, Paraver, Dimemas, Scalasca, Vampir) of the project partners to make them fit for the analysis of 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 the regular tool products. It is decomposed into the tasks:

T2.1 Enhancing functionality of the tools
T2.2 Enhancing scalability of the tools
T2.3 Tool integration
T2.4 Tool validation

WP3: Integration of system and application performance analysis
This contains all research and technical development which is done in cooperation with the partners of the coordinated Russian project. Its objective is to combine and integrate the work done for the HPC system-level performance analysis (by RU-Topic1 in Russia) and for application-level performance analysis (by WP2 in the EU) into a coherent and holistic performance analysis environment. It will provide low-overhead end-to-end performance analysis for all jobs running on a given system from their submission to their completion, identification of key performance issues and notification of the user and system performance database after job completion, and detailed scalable performance analysis for petascale applications based on well-accepted and robust performance measurement and analysis tools. It is decomposed into the tasks:

T3.1 Definition of the interface between system- and application-level performance analysis
T3.2 Definition of the overall performance analysis workflow.
T3.3 Light-weight monitoring module
T3.4 Unified download, configuration, build and installation package.

For a detailed description of the work packages see Table WT3 as well as the list of deliverables (Table WT2) and list of milestones (Table WT4).

The coordinated Russian project HOPSA-RU works on two topics:

RU-Topic1 HPC system-level performance analysis
RU-Topic2 Analysis of FPGA-based systems

Figure 2 shows the overall structure of the project and the major software packages which are developed and enhanced in the course of the project.

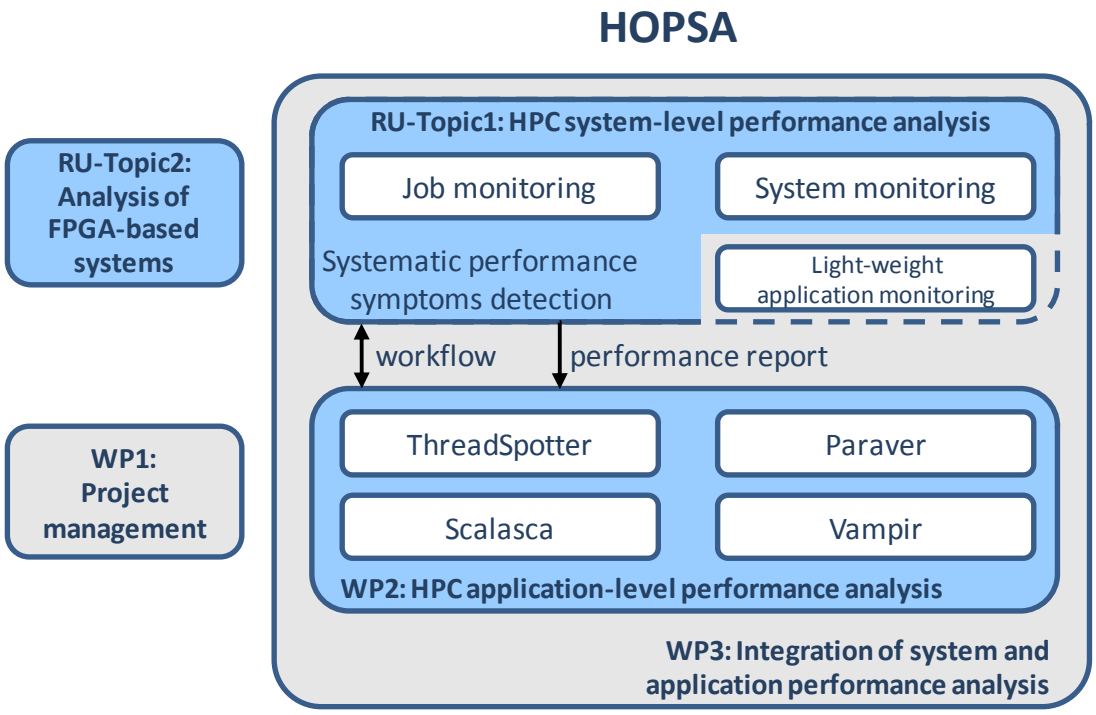

Figure 2: Overall structure of the project.

Significant risks and contingency plans

As in all projects, there are general risks to the success of the proposed plan. Examples of these risks are delays in milestones or deliverables, under-performing partners, or partners who abandon the project. In these cases, the risk is low due to previous collaborations between the chosen partners and if these problems do arise, the Management Board and/or Technical Manager will take action to mitigate the situation. Additional detail is given in paragraph "Risk assessment and contingency strategies" in Section 2.1.

The following table summarises the specific risks and contingency plans associated with the project.

Risk	Impact	Probability	Mitigation
Key milestones or deliverables cannot be completed in time	The project results will be delayed	Low	The Coordinator (and the work package leaders) will foresee possible problems and put the necessary pressure on the partners.
Problems in Agreement with Partners	Productivity is reduced and/or key decisions are delayed	Low	Partners have collaborated previously and responsibilities of each of them are clear. Coordinator will mediate in disputes.

Risk	Impact	Probability	Mitigation
Partner's problems (underperforming, staff changes)	Delays in the project, reduction of quality results	Low	The partners have collaborated before and the Coordinator will work on conflict resolution or perform the required actions to reallocate efforts and budget.
Expertise risks	Partners are not capable to perform the planned activities	Low	Partners have been chosen carefully based on their proven experience
Communication with the Russian partners is difficult because of language barrier, time-zone differences and travel difficulties (e.g. visas)	Will make work on common documents and software very hard	Low	The Coordinator (JUELICH) has an Russian Coordination Office as well as science team members who speak Russian who will be able to help communicate. Partners agreed that documents and discussions will be in English, with translations into Russian as necessary
The integration of Russian and EU parts of the software is not working or not perfect.	The Russian or EU software parts can only be used separately	Medium	The separate parts are still useful. The envisioned interface is simple (Task T3.1) and scheduled early, so there is enough time to monitor this task and to solve or workaround problems early.
The integration of the various EU tools is not working or not perfect.	The tools can only be used separately	Very low	The separate tools are still useful. Proof-of-concept implementation of integration between Scalasca and Paraver or Vampir already exists.
Proposed overall performance analysis workflow is less comprehensive than desired	May limit the number of users that benefit from the project results	Low	Project partners have a long-term experience in analyzing the performance of HPC applications and in the development of tools. This will help covering most common use cases. Usage of the software and procedures on HPC production clusters throughout the project will help finding missing corner cases.

Risk	Impact	Probability	Mitigation
Heterogeneous computing is not adopted	No impact. The performance tools are useful also for the common hybrid MPI/OpenMP case	Very low	The partners will monitor the evolution of hardware and software practices and react in consequence if necessary.
Restrictions due to tools licensing	The fact that some of the tools have a fee-based license may limit the number of users that benefit from the project results	Medium	Most of the involved tools are open source. Especially the first steps (light-weight monitoring) and basic MPI/OpenMP analysis (Scalasca) are open-source. Detailed MPI/OpenMP trace analysis and HW metrics analysis are available in both commercial (Vampir) and open-source (Paraver) form. Only memory and threading analysis (ThreadSpotter) is available commercially only. As the commercial tools are used as the last step in the envisioned performance analysis workflow only, the open-source portion is still useful by itself.
EU system administrators of HPC servers may not allow or limit the installation of the HOPSA system monitoring software (particular coming from third-party sources and especially from Russia)	Some or all of the derived system and application monitoring may not be available	High	Allow partial monitoring or user-voluntary system monitoring. Also, Russian software will be open-source, so its exact working can be checked by everyone interested.
System administrators of HPC servers may not allow or limit the installation of the HOPSA system monitoring software as it very probably requires system privileges	Without system privileges the functionality of the system monitoring may be severely limited or even useless	High	Russian software will be open-source, so its exact working can be checked by everyone interested. Software will be used and tested on pilot systems at the Jülich and Barcelona computing centres. In the worst case, user need to focus exclusively on application-level monitoring.

B1.4 Timing of work packages and their components

Figure 3 shows the timing of the different tasks of the work packages and of the associated deliverables and milestones.

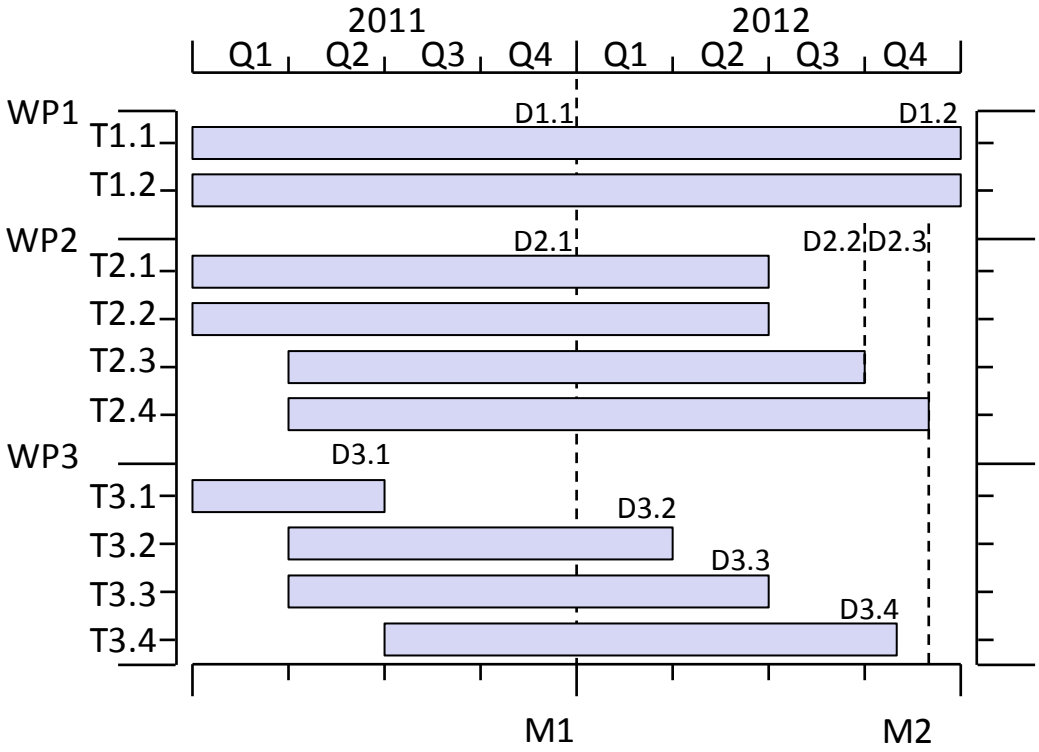

Figure 3: Work package duration gantt chart including deliverables and milestones

B2 Implementation

B2.1 Management structure and procedures

Overview

The management structure for the project evolves from successful models used by the partners in previous projects, taking into account the specific needs of a project that aims to deploy a complex integrated system in a short time frame as well as the contractual restrictions of the Seventh Framework Programme and special regulations resulting from the coordination with our Russian partners.

The EU consortium is constituted by five partners with different organisational cultures (universities, research centres, supercomputing facilities and industry) that contribute to the project with complementary expertise. It is important to note that each of the project partners has been working together in European and national projects. The management structure for the project is designed to provide an appropriate level of professional management to mediate efficiently between the different interests and cultures of the partners; it is based on well-known best practice methodologies. The main purposes of the management structure are:

- to define procedures that ensure timely completion of quality project deliverables keeping within budget,
- to provide an efficient organisation that ensures the involvement of all partners,
- and to provide mechanisms for the management of knowledge and intellectual property and the resolution of conflicts.

The Consortium Agreement to be signed at the beginning of the project will set out the high-level operational rules for this project, including responsibilities of the different management bodies and Intellectual Property Rights (IPR) management. A matching Coordination Agreement will cover the necessary additional coordination with our Russian partners. However, the following sections describe the general organisational and decision-making structure in addition to the more important management procedures. These procedures aim to establish the necessary mechanisms to ensure the success of the project while at the same time keeping the administrative effort to a minimum.

Components

The organisational structure of the project includes the following key components:

- Coordinator: Technical Manager (TM) and Project Management Team (PMT)
- Management Board (MB)
- Work Package Leaders (WPL)

The interactions between the different management components are described below.

Coordinator

Forschungszentrum Jülich GmbH will serve as Coordinator of the HOPSA-EU project. This role is a responsibility shared between the Technical Manager (TM), Bernd Mohr, and a Project Management Team (PMT).

The **Technical Manager (TM)** ensures that the scientific and technical objectives of the project are met. The TM chairs the Management Board (described below). The TM defines the high-level technical strategy and drives the project team to act according to that strategy. In implementing this strategy, the TM also ensures that the project maintains its relevance to the ICT program and its strategic objectives. Moreover, the TM organises technical presentations of the project progress to external parties, provides clear and accurate periodic reports to the EU Project Officer, and ensures the appropriate involvement and visibility of the members of the project. The TM will also act as the official point of contact between the Commission and the Beneficiaries for normal purposes. Finally, he also serves as the main contact point for the coordinator of the associated Russian project HOPSA-RU.

The TM is supported by a **Project Management Team (PMT)**, which is responsible for all administrative, legal, and financial aspects of the project. This includes the provisioning of periodic reports and financial statements as well as interacting with the Financial Department of Forschungszentrum Jülich to ensure an efficient distribution of EU funding. The PMT will coordinate the creation of the Grant and Consortium Agreements. The PMT of Forschungszentrum Jülich GmbH has extensive experience in managing EU-funded as well as commercial projects.

The Coordinator will ensure the timely delivery of project objectives and deliverables by continuously monitoring the project progress against the plan of record. The Coordinator identifies and tracks issues and proposes suitable corrective actions (i.e., resource reallocation, task force creation, etc.) that might require a formal decision by the Management Board. The TM is responsible for calling the Management Board meetings and reviews as well as compiling / distributing minutes and actions. The Coordinator defines the procedures for change control (proposed changes to the plan of record), risk management and quality.

Management Board (MB)

The Management Board is the formal decision-making body that holds the highest level of authority in the project. It is chaired by the Technical Manager and consists of one representative from each partner of the EU proposal and is formally responsible for successful project completion. The MB makes decisions by consensus when possible. In the case that the MB cannot obtain consensus, the MB puts decisions to a vote that is decided by simple majority. In the event of a tie vote, the Technical Manager casts the deciding vote. The MB holds monthly conference calls in order to review the project progress on a regular basis; it has ample powers to make decisions on daily implementation issues. It is also responsible for resource allocation, the review / approval of the periodic reports and deliverables, the preparation of project reviews and the coordination of exploitation plans.

Work Package Leaders (WPL)

The Work Package Leaders are responsible for the scientific and technical work of their respective work packages. This includes the planning and control of all activities within the work package, the preparation of deliverables, and the collection of the contributions from other partners participating in the respective work packages for internal and external reports. They meet regularly via teleconference or face-to-face as a part of the Management Board and arrange for additional technical meetings when necessary. They are expected to raise critical issues to the MB. They must actively participate in the regular project-related meetings as well as prepare technical and status presentations as required.

Meetings and communication

The Management Board will hold regular monthly teleconferences to evaluate progress against project plans, identify major problems, and coordinate project-related interactions among the WP Leaders. The MB will also meet face-to-face at least twice a year, with the meeting's location rotating between the project partners. Every other meeting, but at least once a year, the meeting is shared with the Management Board of the Russian coordinated project to synchronise work, issues, and results between the two projects. The TM will document these meetings (agenda, minutes, action items and plans). The main mechanism for project communication will be email. Mailing lists will be set up for sub-groups (i.e., work packages) as appropriate. The leader of Work Package 1 (WP1) will be responsible for providing project-internal collaborative tools (intranet, distribution lists, etc.)

Knowledge management

The WP1 leader will also be responsible for tracking and disseminating the knowledge produced throughout the duration of the project. The other work package leaders will be responsible for communicating the knowledge produced from their respective work packages to WP1. The WP1 leader will create a knowledge repository as part of the project internal collaborative tools, where all knowledge products as well as critical supportive knowledge material will be documented and stored in an organised and easily accessible manner.

Emergency and conflict resolution procedures

Any event that may jeopardise the overall completion date of the project should be reported immediately to the TM. The TM will call an emergency Management Board meeting or teleconference as required. Each party involved in the issue must present a short document describing their respective understanding of the conflict that includes at least one proposed solution. The MB reviews the conflict documents and following the procedures of the MB, each member votes for one of the proposed solutions. The solution receiving the simple majority is implemented with the chairperson casting the tie-breaking vote as necessary.

Intellectual property management

The Management Board is responsible for handling issues that may arise related to intellectual property rights and innovation activities. The Consortium Agreement will set out the rules for all aspects of the STREP operation, including IPR management, according with the IST FP7 regulations. The Consortium Agreement will include a schedule of any existing IPR ("know-how") that partners are bringing to the project as a basis for their RTD and which will remain their property. They will identify any items that are freely available to partners for access and/or for use; and those that are subject to commercial restrictions or payment of license fees. The purpose of documenting project-generated technology components that are derived from the project is to facilitate exploitation once the project has ended. The WP1 leader maintains a consolidated IPR Register (document) on an ongoing basis and makes it available to the consortium via the project portal. The TM assists in clarification of IPR and licensing issues as required.

Quality control & assurance

The project progress and results will be assessed with a number of internal and external control procedures. Quality control and assurance will allow maximum flexibility while maintaining a clear distinction of roles and responsibilities of all partners involved. To this end, the project will establish appropriate mechanisms and procedures, involving all partners. An informal quality plan will be produced at the first stages of the project. The goal is to ensure the detection of errors and deviations as early as possible in the project's life cycle. This will enable the consortium to systematically apply corrective actions or contingency plans, if necessary. Quality control and assurance will be the basis of self-assessment for the project and will control the input and output as well as the interactions between all work packages within the project. It will also identify the additional controls to be applied within work packages. Finally, it will ensure that the smallest possible administrative overhead is imposed, consistent with necessary control to achieve quality.

Risk assessment and contingency strategies

The TM is responsible for the risk management in the project and will be continuously observant of risk situations that have developed and may develop during the project, in order to detect and contain the project risks. The TM will report on risks and issues to the management board and will keep a risk / issue log for the project as well as assign actions or contingency plans to be executed as required so as not to impact the overall outcome or objectives of the project.

B2.2 Beneficiaries

B2.2.1 Forschungszentrum Jülich GmbH, Jülich Supercomputing Centre (Coordinator)

Forschungszentrum Jülich (JUELICH) is one of Germany's largest national laboratories. With a strong background in physics and scientific computing, it is devoted to multidisciplinary research and development in the areas of health, energy & environment, and information technology. To support the research centre's scientific mission and also to offer leadership computing power to scientists all over Germany and in Europe (e.g., via the PRACE and DEISA efforts), the Jülich Supercomputing Centre (JSC) runs one of the most powerful scientific computing centres in Europe. Combining expertise from computational science, computer science, and applied mathematics, JSC's research and development activities concentrate on the methodological advancement of supercomputing and the operation of supercomputers as scientific large-scale devices alongside the required information technology infrastructure for mass data storage, broadband communication, Grid computing, and multimedia.

Being an essential component of its scientific mission, research on tools for parallel programming has a long tradition in Jülich, resulting in two decades of experience in developing and using performance-analysis tools for parallel and distributed applications. With the introduction of the KOJAK toolkit [34] and its highly-scalable successor Scalasca [28], JSC maintains world-wide leadership in automatic trace analysis. Since 2010, Scalasca is developed and maintained in cooperation with the German Research School for Simulation Sciences (GRS). Finally, JSC is actively investigating innovative multiprocessor architectures including GPGPU and Cell/B.E.-based clusters.

In this project, JSC will take the role of the project coordinator. On the technical side, JSC, in cooperation with GRS, will work on the Scalasca toolset to enhance scalability and functionality, will integrate it with the Vampir and Paraver visualizers, and is the main developer of the UNITE package. Finally, we will contribute to the overall integration work in WP3.

JUELICH key people involved

Bernd Mohr started to design and develop tools for performance analysis of parallel programs already with his diploma thesis at the University of Erlangen in Germany, and continued this in his Ph.D. work. During a three year PostDoc position at the University of Oregon, he was responsible for the design and implementation of the original TAU performance analysis framework for the parallel programming language pC++. Since 1996 he is a senior scientist at the Research Centre Jülich where he is leading the KOJAK and Scalasca performance tools efforts together with Felix Wolf at the GRS. He was a founding member and work package leader of the European Community IST working group on automatic performance analysis: APART and is Secretary of the IBM Scientific Users Group ScicomP. He is the author of several dozen conference and journal articles about performance analysis and tuning of parallel programs.

B2.2.2 Rogue Wave Software AB (RW)

In September 2010 Acumem was acquired by Rogue Wave Software and the new Swedish company Rogue Wave Software AB was formed, where the entire Acumem development team, including its founder Erik Hagersten, now work. Acumem was founded in 2006 as a spin-out of Uppsala University's Architectural research. The company has a mixture of industrial and University expertise among its employees. It launched its first product SlowSpotter in November 2007, which forms an analysis and enhancement suite for single-threaded applications, and its second product ThreadSpotter in November 2008, performing multithreaded analysis. The products perform automatic analyses of the execution behaviour of a binary and suggest hands-on changes at the source level to a programmer. The products have quickly been adopted by the market and have been backed by Acumem's active partners: HP, IBM, Intel, AMD and Microsoft.

Acumem developed a new and efficient set of technologies, in particular the ThreadSpotter tool, providing a key advantage in the transition to multicore systems. A small performance fingerprint is captured from a multi-threaded application at runtime, while adding only 20% overhead to the execution for long-running applications. Efficient modeling techniques model the application's behavior with respect to a memory system based on the fingerprint, either at runtime or off-line. The whole process can be thought of as an ultra-fast "simulator". The technique models any level in the memory hierarchy and can calculate the respective miss-rate, fetch-rate, prefeching activity and inter-thread communication in fractions of a second. The low overhead of the technology allow for production-sized workloads to be analyzed. Today, RW's existing performance products, SlowSpotter and ThreadSpotter, leverage this and other features of the technology to automatically detect performance bugs in a multi-threaded application and suggest source-code changes to the programmer. It is interesting to note that the fingerprint captured only contains architecturally-independent data and that any memory hierarchy can be modeled in fractions of a second. This allows the system to model the performance of the application on many different memory architectures based on a single fingerprint.

In this project RW will enhance the scalability of its technology and also reduce the amount of data and complexity presented to a programmer of scalable applications.

Rogue Wave key people involved

Professor **Erik Hagersten** shares his time between computer architecture research at Uppsala University, Sweden, and Rogue Wave Software AB, where he is the CTO. He has previous experience from large server designs as the chief architect for Sun Microsystem's high-end server engineering

division. He led the architecture research group at the Swedish Institute of Computer Science (SICS) for five years. The group produced novel research contributions, such as the full-system simulator Simics, which later was spawned off as the company Virtutech Inc, the original idea of the Cache Only Memory Architecture (COMA) as well the practical implementation thereof: Simple COMA and the Data Diffusion Machine (DDM). Previously to joining SICS, Hagersten was a visiting scientist at the MIT LCS lab and worked on of fault-tolerant embedded CPUs designs at Ericsson. While at Uppsala, Hagersten and his students developed the StatCache technology, which is instrumental for many of the research activities described in this application, together with two of his PhD students. Hagersten is the author of about 50 academic papers and holds more than 100 patents. He is a member of the Royal Swedish Academy of Science and Engineering (IVA).

Mats Nilsson is the VP of engineering at Rogue Wave Software AB and a co-founder of Acumem. He brings to bear architectural and project leading skills from advanced software development at the companies Elekta, XWare and Siemens, both in Sweden and France. Nilsson is the software architect of the current products. He holds an MS in electrical engineering from the Royal Institute of Technology in Stockholm, Sweden.

B2.2.3 Barcelona Supercomputing Center (BSC)

Barcelona Supercomputing Center - Centro Nacional de Supercomputación BSC-CNS is the Spanish National Supercomputing Center. The mission of BSC is to investigate, develop and manage information technology in order to facilitate scientific progress. BSC provides access to MareNostrum (100 TF machine) to a large community of users. BCS-CNS is not only a supercomputing services center, but also a research center with 125+ researchers in all levels of supercomputer design (architecture, programming models, operating system, performance tools) and several application areas (engineering, life sciences and earth sciences).

This multidisciplinary structure brings together a critical mass that enables a holistic view of supercomputer design. BSC builds on previous experience in international and cooperative research projects by CEPBA-UPC, including R&D, LTR (NANOS, POP,...) and management of technology transfer (PACOS, TTN). BSC is also involved in projects related to infrastructure (DEISA, PRACE) and mobility of researchers (HPC-Europa). Since its creation, BSC has been able to gain a respected position within top HPC service and research institutions.

The work in the area of performance tools started in 1991 within the framework of CEPBA-UPC. The Paraver and Dimemas tools were initially developed for internal usage to overcome the limitations of the tools available on the market. Since 2000 the tools have been widely distributed, initially under a proprietary license and since 2009 as open source codes.

BSC contributes to work packages WP2 and WP3 of the project bringing a strong experience in performance analysis tools. In WP2 we will increase the scalability of trace-based approaches by intelligently deciding the level of detail of the emitted information, using Dimemas to perform root cause analysis and better integrate Paraver with Scalasca's report browser CUBE. In WP3 we will contribute to the definition of the global workflow and integrating our tools within the unified package.

BSC key people involved

Jesús Labarta has been full professor at the Computer Architecture department at UPC since 1990. Since 1981 he has been lecturing on computer architecture, operating systems, computer networks and performance evaluation. His research interest has been centred on parallel computing, covering areas from multiprocessor architecture, memory hierarchy, parallelising compilers and

programming models, operating systems, parallelisation of numerical kernels, metacomputing tools and performance analysis and prediction tools. He led the technical work of UPC in some 15 industrial R+D projects. Significant performance improvements were achieved in commercial codes owned by partners with whom he has cooperated. In 1995 he became director of CEPBA and currently he is director of the Computer Sciences research department at BSC.

B2.2.4 German Research School for Simulation Sciences (GRS)

The German Research School for Simulation Sciences (GRS) is a joint venture of Forschungszentrum Jülich and RWTH Aachen University, combining the specific strengths of the two founders in the fields of science, engineering, and high-performance computing in a unique synergistic way. Located in dedicated modern facilities on the Aachen and Jülich campuses and equipped with privileged access to world-class computing and visualisation resources, the school is committed to research and education in the applications and methods of HPC-based computer simulation in science and engineering. As an essential element of its mission, the school provides a Master's and a doctoral program designed to train the next generation of computational scientists and engineers. Affiliated with the computer science department of RWTH Aachen University, the Laboratory for Parallel Programming, one of the school's four research divisions, specialises in tools that support simulation scientists in exploiting parallelism at massive scales. The laboratory is also partner in the Scalasca project, where it concentrates on tool scalability and emerging programming models.

In this project, GRS will work on the enhancement of Scalasca's functionality and scalability together with JSC, providing expertise in the study of time-dependent performance behaviour and in sampling-based profiling techniques. GRS will contribute the latter also to the development of the light-weight and low-overhead measurement module at the interface between system and application analysis.

GRS key people involved

Felix Wolf is head of the Laboratory for Parallel Programming at the German Research School for Simulation Sciences in Aachen and a full professor at RWTH Aachen University, where he teaches parallel programming. His research concentrates on programming tools for large-scale parallel computers. Wolf has published more than 60 refereed articles in journals and conference or workshop proceedings. He has obtained research funding from German and American funding agencies including BMBF, DFG, DOE, Helmholtz Association, and NSF. Wolf is a principal author of the Scalasca performance-analysis software, now a large team effort that he leads jointly with Bernd Mohr from the Jülich Supercomputing Centre. Moreover, Wolf is founder and spokesman of the Virtual Institute – High Productivity Supercomputing, an international initiative of leading academic HPC programming-tool builders aimed at the enhancement, integration, and deployment of their products.

B2.2.5 Technische Universität Dresden, ZIH (TUD)

The Center for Information Services and High Performance Computing (ZIH) is a central scientific unit of the Technische Universität Dresden with a broad spectrum of services and research competencies. It is responsible for the communication infrastructure of the university and operates the central information technology servers and services. In addition, with its interdisciplinary orientation, ZIH supports other departments and institutions in their research and education for all matters related to information technology and computer science.

Furthermore, ZIH offers its HPC resources to academic users as well as support for HPC application developers regarding parallelisation methods and performance optimisation. ZIH's own research activities that are significant for this proposal include interactive performance analysis and visualisation based on event tracing methods. The Vampir tool [18, 30] developed at TU Dresden which is the world-wide market leader in this area is distributed commercially. The software packages VampirTrace [21, 31] and Open Trace Format (OTF) [17, 24] developed by ZIH are offered as open source and are official parts of the Open MPI distribution. Newer research activities of ZIH focus on GPGPU computing for scientific applications and energy efficiency in HPC.

In this project, TUD will work on the runtime monitoring system, the event trace format, and on the Vampir trace visualisation. For all three components, TUD will introduce new features, enhance the scalability and improve or start integration with the partner tools.

TUD key people involved

Wolfgang Nagel is director of ZIH at TU Dresden and professor of Computer Architecture at the Department of Computer Science of TU Dresden. From 2006 to 2009 he was dean of this department. His other professional activities include, among others, the position as head of the advisory board of HLRS Stuttgart, member of the DFG commission for IT-Infrastructure (KfR) and chairman of the Gauß-Allianz. Furthermore, he was chairman and organiser of several conferences, he is member of the steering committee of the EuroPar Conference series, and associated editor of "Informatik Spektrum". He is the original initiator of the well-established Vampir tool [18], which he has been supervising for more than 15 years. His publications include several dozen conference and journal articles about many different aspects of high performance computing.

Matthias S. Müller is deputy director and CTO of ZIH at TU Dresden. He received his PhD in Computational Physics from Stuttgart University in 2001. From 1999 to 2005 he worked at the High Performance Computing Center in Stuttgart, Germany, which he left as a deputy director. His research interests include programming methodologies and tools, computational science on high performance computers and Grid computing. Among other tasks he is head of the VampirTrace development group. He is a member of the German Physical Society (DPG), the expert group of the European Exascale Software Initiative (EESI) and Vice Chair of SPEC' High Performance Group.

Andreas Knüpfer is a research scientist at the Center for Information Services and HPC at Technische Universiät Dresden. His fields of interest are parallel programming paradigms and HPC performance analysis. He is involved in the development of the VampirTrace and SILC performance monitoring systems and the training activities for Vampir and VampirTrace. Furthermore, he was the main designer of the Open Trace Format (OTF).

B2.3 Consortium as a whole

The HOPSA project, even though small in size, still gathers an impressive and complementary set of leading European researchers in performance measurement and analysis tools for parallel programs, the work programme topic this proposal addresses.

Long-term and well-acknowledged experience in performance tools

All partners involved in HOPSA-EU are world-wide well-accepted experts in the area of performance measurement, analysis, modelling, and visualisation of parallel programs with an experience which in some cases spans almost 25 years. The tools of Rogue Wave, SlowSpotter and ThreadSpotter, have

been unique products in the area of automatic diagnosis of memory and threading related performance since their introduction in 2007. The tools group of BSC lead by Prof. Jesús Labarta has been working on tracing-based performance measurement, modelling, and visual analysis tools for almost 20 years now. The semantic-free trace format coupled with the programmability of the analysis makes their Paraver performance data browser the most flexible trace-based performance tool available today. Bernd Mohr of JUELICH started to work on performance tools for parallel programs already in 1987 when implementing trace monitoring tools for the parallel computer prototype DIRMU at the University of Erlangen-Nuremberg. In 1994, he was responsible for the design and implementation of the original TAU performance analysis framework, now the most used open-source performance tool in the U.S. Together with Prof. Felix Wolf, now at GRS, he invented KOJAK, the world's first automatic trace analyser for parallel programs. Since 2006, they have been jointly working on the Scalasca project, the world's most scalable performance tool which already successfully analysed parallel programs running on 294,912 cores. The Vampir group lead by Prof. Wolfgang Nagel has also been working on the measurement and visualisation of parallel programs for over 20 years. Since 1996, Vampir has been the most robust and portable product for parallel trace visualisation. They also pioneered the parallelisation of trace analysis, showing that trace-based analysis and visualisation is feasible even for highly-parallel applications.

Complementarity between participants

Although all project partners of HOPSA-EU are all working in the narrow field of parallel-program analysis, they still bring together all complementary aspects needed for the successful execution of an international project as they are rooted in different forms of organisations:

- BSC, and especially JUELICH with its over 4300 employees, are **multi-disciplinary research centres** providing easy access to outside expertise when necessary.
- Barcelona Supercomputing Center (BSC), JUELICH in the form of its Jülich Supercomputing Centre (JSC), and TUD via its Center for Information Services and High Performance Computing (ZIH) are among Europe's most powerful **HPC computing centres**. All offer regional, national and pan-European (e.g., via PRACE or DEISA) access to high-performance computers for the scientific community. They have experience in running and maintaining HPC systems and through their help desks they know the requirements, problems, and issues of their users in regard to parallel programming. The Paraver, Scalasca, and Vampir tool groups are often directly involved in providing support to users of their respective computing centres when it comes to analyse, tune, and optimise the performance of the user's application programs.
- The embedding of an research project in a **university and educational context** is also very important. In HOPSA-EU, this is accomplished through BSC, via its associate member Universitat Politecnica de Catalunya (UPC), through the German Research School for Simulation Sciences (GRS) via it stakeholder RWTH Aachen University, and through the Technical University of Dresden (TUD).
- Finally, expertise in **commercial exploitation** is helpful for a successful research project. Both our company partner Rogue Wave and TUD via the GWT TU Dresden GmbH have been effective in marketing and selling performance tools to the HPC community for many years.

The complementary composition of the EU partners is equally matched by our coordinated Russian project. Here, the leading Russian HPC centres (the Research Computing Center of Moscow State University and the Joint Supercomputer Center of the Russian Academy of Sciences), a research center (Russian Academy of Sciences), universities (Moscow State University, Southern Federal University) and a commercial HPC provider (T-Platforms) also form a powerful consortium.

Long-term experience in project participation and management

All HOPSA-EU project partners have a long experience in participating and managing national, European, and international research projects. H4H is the first EU project for Rogue Wave Software AB, even though Erik Hagersten himself has co-created and participated in several projects, such as MP-MIMD and PEPMA. BSC builds on previous experience in international and cooperative research projects by CEPBA-UPC, including R&D, LTR (NANOS, POP,...) and management of technology transfer (PACOS, TTN). BSC is also involved in projects related to infrastructure (DEISA, PRACE) and mobility of researchers (HPC-Europa). Felix Wolf of GRS has obtained research funding from German and American funding agencies including BMBF, DFG, DOE, Helmholtz Association, and NSF. Moreover, Wolf is founder and spokesman of the Virtual Institute – High Productivity Supercomputing (VI-HPS), an international initiative of leading academic HPC programming-tool builders aimed at the enhancement, integration, and deployment of their products. Like BSC, JUELICH is involved in national (eeClust, SILC) and European research projects (ParMA, TEXT), working groups (APART, EuroTools), and infrastructure projects (LOFAR, DEISA, PRACE). TUD also participates in a number of research collaborations funded by BMBF in Germany like SILC, eeClust, HI-CFD, and TIMaCS as well as EU projects like ParMA in the past and H4H in the present. Furthermore, TUD is involved in the VI-HPS project, the CoolComputing Project which is part of the German "Spitzencluster" Cool Silicon, and is a member of the German Gauß-Allianz, which is currently chaired by Prof. Nagel.

Moreover, the project partners worked already successfully together in many research projects and collaborations. Currently, the parties are collaborating in the SILC, PRIMA, H4H, VI-HPS, PRACE, DEISA, and TEXT projects. For a more detailed description of these projects see the subsection "Related projects" of Section 1.2 of this proposal.

Ensuring exploitation and dissemination of project results

The project partners also bring together all the necessary experience and are involved in important national, European, and international collaborations, projects, and organisations to ensure a successful exploitation and dissemination of project results. BSC, JUELICH, and TUD will not only install and employ the performance tools in their HPC computing centres, but their involvement in the German HPC Gauß-Alliance (JUELICH, TUD) and the European PRACE and DEISA initiatives (BSC, JUELICH) will assure that efficient and effective parallel performance tools are available to the European HPC community as a whole.

BSC and JUELICH are also partners in the European Exascale Software Initiative (EESI) [6] funded by EU FP7 which will define a European vision and roadmap to address the challenge of the new generation of massively parallel systems composed of millions of heterogeneous cores which will provide Petaflop performance in 2010 and Exaflop performance in 2020. In addition, Jesús Labarta of BSC, Bernd Mohr of JUELICH, and Wolfgang Nagel of TUD are participating from the beginning in the International Exascale Software Project (IESP) [13] which coordinates the world-wide efforts to implement system software for Exascale computing.

As already explained, BSC via its associate member Universitat Politecnica de Catalunya (UPC), GRS via its stakeholder RWTH Aachen University, and TUD will use project results for the university education of future students in the HPC area, while our company partner Rogue Wave and TUD via the GWT TU Dresden GmbH will ensure the commercial exploitation of project results. Section B3.2 describes the exploitation and dissemination activities of the project partners in more detail.

BSC third party: UPC

The BSC is a consortium that is composed of the following member institutions: Universitat Politècnica de Catalunya (UPC), Spanish Council for Scientific Research (CSIC), as well as the Spanish and the Catalan governments. As part of the charter of this consortium, each member institution must contribute resources either in cash or in kind. Both UPC and CSIC contribute in kind by making human resources available to work on projects. The relationship between BSC and CSIC / UPC is defined in an agreement with each institution that was established prior to the start of this project.

Universitat Politècnica de Catalunya (UPC)

The High Performance Computing research group of the Computer Architecture Department at the Universitat Politècnica de Catalunya (UPC) is the leading research group in Europe in topics related to high performance processor architectures, runtime support for parallel programming models, performance tuning applications for supercomputing.

Directly derived from the research effort at the Computer Architecture Department, the CEPBA (European Center for Parallelism in Barcelona) was founded in 1991 to offer supercomputing resources to the research community and as a development center for industrial computing technology products. In 2000, IBM joined forces with CEPBA to form the CIRI (CEPBA-IBM Research Institute Joint Lab) in Barcelona in order to strengthen relationships between IBM and UPC researchers in computer architecture. In 2005, the Spanish and Catalan governments signed an agreement with IBM to buy the 4th supercomputer in the world and extend the operations of CIRI to become the Barcelona Supercomputing Center (BSC).

The High Performance Computing research group at the UPC shares many key resources with the BSC, including several key personnel that will be dedicated to this project. There is a signed Collaboration Agreement between the UPC and the BSC establishing the framework of the relationship between these two entities. According to this agreement, several professors of the UPC are made available to the BSC to work on projects. Professor Jesus Labarta and Judit Gimenez are personnel from the UPC. They carry out their research activities in association with the Barcelona Supercomputing Center - Centro Nacional de Computación (BSC) on the BSC premises.

B2.4 Resources to be committed

The HOPSA consortium aggregates the corresponding personnel and equipment capable for realising the scientific and technological plan described in section 1.3 and ensuring its impact on the European community.

The partners already have personnel with the required expertise, to ensure a quick start of the project. Besides, all partners will put the required effort on contracting the additional personnel required for the project. The additional staff resources not covered by the amount requested to the EC will be covered by the partners from different sources, depending on the nature of the institutions.

The needed resources are essentially human effort, travel, hardware and software for the developers and for demonstration purposes. The five partners will bring an overall total effort of 191 person-months over 2 years. The overall budget of the project is 1,925,138 €out of which a contribution of the EC of 1,399,966 €is requested.

RTD and management

The main costs of the project are personnel costs, as can be observed in the table WT8. These costs are split between RTD and management activities. The management is kept to a minimum, but adequate for efficient and lean management. The majority of the management effort has been allocated to the project coordinator.

Travel costs

Direct costs are totally devoted to cover travel expenses of partners to review meetings, project meetings and attendance to conferences and other dissemination-related activities. The following project meetings are planned:

month 3:	EU only meeting
month 5:	Joint EU-Russia meeting
month 8:	EU only meeting
month 12:	Intermediate EU review
month 14:	Joint EU-Russia meeting
month 20:	EU only meeting
month 23:	Final EU-Russia meeting / review
month 23:	Final EU review

The following table summarises the allocation to travel costs per partner:

RW	21,000 €
BSC	24,000 €
JUELICH	24,000 €
GRS	24,000 €
TUD	21,000 €

Additional resources made available for the project

The HOPSA-EU consortium will put together complementary resources of different nature to the project, including:

Hardware resources

- BSC has different hardware platforms available where the project partners can run their applications and tools: first, the MareNostrum supercomputer with 10240 IBM Power PC 970MP processors; second, the MariCel PRACE prototype, which has an hybrid architecture with 12 JS22 blades with Power6 processors and 72 QS22 blades with PowerXCell processors (total of 1344 cores); additionally, the SGI Altix with 128 cores offers a platform for trying homogeneous SMP environments and the centre has also NVIDIA based systems.
- JUELICH provides access to JUGENE, a Blue Gene/P currently with a total 72 racks, organised in 73728 compute nodes, each of them 4-way SMP. The processor type is a 32-bit PowerPC 450 core running at 850 MHz and the total count is 294912 processors. This platform will be of course very interesting to be used for scaling tests with a large number of cores. JUELICH cannot promise a large amount of computing time, but test runs will be possible. In case there is a need for larger run times, the partners can submit a proposal for computing time. JUELICH also can provide access to JUROPA, a Linux cluster with 2208 compute nodes each with 2 Intel Xeon X5570 (Nehalem-EP) quad-core processors and an Infiniband interconnect. It is planned

that JUELICH will also acquire a computing platform with heterogeneous architecture (e.g. with GPU-based acceleration) within the project timeframe.

- TUD provides access to its HPC resources which consists of several HPC clusters with up to 2500 CPU cores, among them a smaller cluster equipped with NVIDIA GPUs for parallel GPGPU computations, and a large shared memory machine especially designed for data intensive computing. The latter is a SGI Altix 4700 machine with 2048 Intel Itanium cores and 6.5 TB shared main memory. The orientation towards data intensive computations will be an important aspect for the successor installation planned during the HOPSA project. Furthermore, advanced system monitoring possibilities will be key features for this machine, in order to allow detailed investigation of energy consumption. This will allow to incorporate the important question of energy efficiency into the analysis of computational efficiency, i.e., performance. The planned installation will also be made available to the HOPSA project.
- We expect that we also get access to the systems of the computing centres at Moscow State University and the Russian Academy of Sciences.

All partners involved in the consortium will have equipment for virtual conferencing and for technical meetings available. Since all these resources will be available at the partner sites, they do not represent any additional cost to the proposal. All these systems are connected to the Internet and therefore available from the partner sites.

Software resources

There is a set of software resources provided from partners from the beginning of the project (in their initial status) for the development of the HOPSA objectives and installed on at least one of the platforms available for the project:

- ThreadSpotter (RW)
- Scalasca performance analysis tool including CUBE (JUELICH, GRS)
- Paraver, Dimemas (BSC)
- The Vampir toolset (TUD)

Besides, all the basic software (editors, operating systems, SDK, etc.) will be installed on the hardware platforms to enable the developments of the project. Other software for the development of the website (i.e., a content management system) will be available, as well as a versioning system (CVS, SVN, etc.) to facilitate the maintenance of software versions.

Third parties (other than subcontractors)

Some of the work that will be carried out at the Barcelona Supercomputing Center - Centro Nacional de Supercomputación (BSC) will be contributed by the Third Party, Universitat Politècnica de Catalunya (UPC):

	WP1	WP2	WP3	total
UPC	0.5 PM	6 PM	9 PM	15.5 PM
BSC	0.5 PM	25 PM	1 PM	26.5 PM
total	1 PM	31 PM	10 PM	42 PM

A detailed description of the UPC is provided in Section B2.3 in addition to a list of the individuals that will participate in the project.

B3 Impact

B3.1 Strategic impact

This project will create 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 will contribute a collection of mature high-level tools for application performance analysis to be further enhanced with respect to their scalability, the depth of their analysis, and their support for asynchronous tasking. The tools will be made part of 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 will allow 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 will also be 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 will be 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 outlined in this proposal would automatically filter out the unique strands' behaviours and reduce the optimisation problem to that of a single (or handful of) strands. This will greatly simplify the optimisation of hybrid program for exascale systems.

Paraver. The main enhancement expected 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 interoperability of the tools would open new ways of using the tools. A second enhancement would be with respect to the scalability of the tools that up to now have been tested with up to 16k processes (except for Dimemas).

Scalasca. Critical-path analysis is expected to simplify the identification of optimisation targets in the code, substantially shortening the optimisation cycle time. The distributed recording of communicators is the last step in a longer sequence that will allow Scalasca to be comfortably used with more than 100,000 MPI processes and that will significantly lower the measurement overhead in the interest of more reliable performance data. Finally, the compression of sampled time-series profiles will help 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 will be 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 lang running programs and selective tracing of time intervals or processes/threads. The system-level analysis will provide monitoring data from the system monitoring of our project partners which is influencing the application-level performance behaviour. Finally, the integration with the Scalasca and ThreadSpotter will improve 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 fork-join-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 envisage 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 will also be promoted through closer integration. This will allow the extended interpretation of our diagnostic workflow as a cross-tool workflow, 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 will allow the exploration of their precise circumstances via cross-tool controls. Moreover, a unified download, configuration, build and installation package will drastically simplify the combined installation and usage 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 proposal of our Russian partners.

B3.2 Plan for the use and dissemination of foreground

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

- There will be a project website under the `.eu` domain for overall presentation and dissemination and of the project. At this website, the contact at the EC will find project-internal deliverables, interested individuals will find public deliverables, as well as the news regarding the progress of the project 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. In addition, the project will maintain wiki-based systems for internal documentation as well as tools for bug tracking or trouble-ticket-systems.

  The Project Manager and the WP1 leader in particular will be monitoring the results achieved by all dissemination activities, and will collect all relevant information including papers, contributions to conferences such as posters and invited talks to make them available at the project website. To build up a unified identity of the project, a logo will be designed. Publications, presentations, and posters will include this logo to increase recognition.
- The project will plan adequately resourced activities devoted to dissemination for specialised constituencies and the general public, in particular for awareness and educational purposes. The dissemination has to consider the objectives of the project including its societal and economic impact. The channels to be used should include web-based communication, press releases, brochures, booklets, multimedia material, etc.. The ”dissemination material” should be regularly updated to provide the latest version of the project status and objectives. Electronic and/or paper versions of this ”dissemination material” will be made available to the Project Officer beforehand for consultation and upon its final release. A proper acknowledgement of the funding source (the FP7 logo and the EU flag, etc.) will appear in all dissemination activities.
- There will be coordinated dissemination activities between all EU and Russian partners to jointly communicate project objectives and results. This will include at least a common press release in month 5 (in time for ISC 2011) and month 22 (in time for SC 2012). The project will also propose a Birds-of-a-Feather (BOF) session for ISC 2012 where we will report on early project results.
- The academic partners (BSC via UPC, GRS, 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 IESP or EESI, 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 (SlowSpotter, ThreadSpotter, Vampir) and the services they offer, thus enabling their users to rapidly benefit from these enhancements. The project results will 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)
  - OTF library (TUD)
  - VampirTrace (TUD)
  - Paraver (BSC)
  - Dimemas (BSC)
  - Extrae (BSC)
- Results and experiences will be directly exploited concurrently with other research projects the project partners are involved in. For a more detailed description of these projects, see the subsection "Related projects" of Section 1.2 of this proposal.
- All project partners will 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" (ПаВТ), `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 webportal (`http://supercomputers.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:

B3.2.1 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, an 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.

B3.2.2 Barcelona Supercomputing Center (BSC)

BSC is highly committed to the HOPSA project and very interested in exploiting its results. Performance analysis tools is one of the research topics we have been working on for close to 20 years. It is very important for us to participate in this initiative to increase the tools' interoperability and scalability that we believe in some sense would change and improve the way that performance tools are used.

CEPBA-Tools are freely distributed as open source, so all the developments and improvements implemented in HOPSA would be open to the HPC community and all the tool users would benefit from the enhancements implemented.

BSC will continue organising workshops and tutorials to promote and train in the usage of performance analysis tools.

B3.2.3 Forschungszentrum Jülich GmbH (JUELICH)

JUELICH has the following plans for dissemination and exploitation of the HOPSA project:

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-parallel computer system of the world (Jugene) 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 will help 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 will be installed on the JSC production computer systems already during the project. Thus, users as well as JSC user support personnel will immediately benefit 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 manyfold 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 will continue to organise workshops and other events promoting the use of performance tools for parallel programs. For example, JUELICH, in cooperation with project partner BSC, organised the Dagstuhl seminar ”Program Development for Extreme-Scale Computing” (see `http://www.dagstuhl.de/10181`). 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) from 1999 to 2009 and at the International Supercomputing Conference (ISC)).

The increased visibility and competence that we will achieve 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.

B3.2.4 German Research School for Simulation Sciences (GRS)

As a partner in the Scalasca project, the GRS plans to contribute the extensions it will develop 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 is currently redesigning with the help of a corporate publishing company, will explicitly refer to the HOPSA project and feature a link to the HOPSA website. Further attention to HOPSA results can be expected from the Virtual Institute – High Productivity Supercomputing (VI-HPS, see Section B1.2 on related projects), 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 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.

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, the project will present an ideal opportunity to promote young scientists. The GRS will use project funds to hire graduate students who will be trained using small project-related tasks. Ph.D. students will be given a chance to contribute ideas and to pick up results for their own work. 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.

B3.2.5 Technische Universität Dresden (TUD)

The HOPSA project will help 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 an 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 SILC 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 [23]. After the transition from VampirTrace to the SILC 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.

References

[1] L. Adhianto, S. Banerjee, M. Fagan, M. Krentel, G. Marin, J. Mellor-Crummey, and N. R. Tallent. HPCToolkit: Tools for performance analysis of optimized parallel programs. *Concurrency and Computation: Practice and Experience*, 22(6):685–701, April 2010.

[2] Argonne National Laboratory. FPMPI-2: Fast profiling library for MPI. `http://www.mcs.anl.gov/research/projects/fpmpi/WWW/`.

[3] F. Bodin and S. Bihan. Heterogeneous multicore parallel programming for graphics processing units. (to appear in Scientific Programming Journal).

[4] I.H. Chung, R.E. Walkup, H.F. Wen, and H. Yu. A study of MPI performance analysis tools on Blue Gene/L. In *Proc. of the 20th IEEE International Parallel & Distributed Processing Symposium*, Rhodes Island, Greece, 2006.

[5] L. DeRose, B. Homer, D. Johnson, S. Kaufmann, and H. Poxon. Cray performance analysis tools. *Tools for High Performance Computing*, pages 191–199, 2008.

[6] ESSI. European exascale software initiative. `http://http://eesi-project.eu/`.

[7] V. Pillet et al. Paraver: A tool to visualize and analyze parallel code. In *18th World OCCAM and Transputer User Group Technical Meeting*, April 1995.

[8] Message Passing Interface Forum. MPI: A message-passing interface standard, version 2.2, September 2009. Chapter 14: Profiling Interface.

[9] K. Fürlinger and S. Moore. Capturing and analyzing the execution control flow of OpenMP applications. *International Journal of Parallel Programming*, 37(3):266–276, 2009.

[10] K. Fürlinger, N. J. Wright, and D. Skinner. Effective performance measurement at petascale using IPM2. In *In Proc. of the IEEE International Conference on Cluster Computing*, Heraklion, Crete, Greece, September 2010. IEEE Computer Society.

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

[12] K.A. Huck, A.D. Malony, R. Bell, and A. Morris. Design and implementation of a parallel performance data management framework. In *Proc. International Conference on Parallel Processing (ICPP 2005, Oslo, Norway)*. IEEE Computer Society, June 2005.

[13] IESP. International exascale software project. `http://http://www.exascale.org/`.

[14] Intel. Intel trace analyzer and collector. `http://software.intel.com/en-us/intel-cluster-toolkit-compiler/`.

[15] J. Gimenez J. Labarta. *Parallel Processing for Scientific Computing*, chapter 2: Performance Analysis: From Art to Science. SIAM, 2006.

[16] Rosa M. Badia Josep M. Perez and Jesús Labarta. A dependency-aware task-based programming environment for multi-core architectures. In *IEEE Cluster 2008*, September 2008.

[17] Andreas Knüpfer, Ronny Brendel, Holger Brunst, Hartmut Mix, and Wolfgang E. Nagel. Introducing the Open Trace Format (OTF). In Vassil N. Alxandrov, Geert D. Albada, Peter M. A. Slot, and Jack J. Dongarra, editors, *6th International Conference on Computational Science (ICCS)*, volume 2, pages 526–533, Reading, UK, 2006. Springer.

[18] Andreas Knüpfer, Holger Brunst, Jens Doleschal, Matthias Jurenz, Matthias Lieber, Holger Mickler, Matthias S. Müller, and Wolfgang E. Nagel. The Vampir performance analysis toolset. In M. Resch, Rainer Keller, Valentin Himmler, Bettina Krammer, and Alexander Schulz, editors, *Tools for High Performance Computing*, pages 139–155. Springer Verlag, July 2008.

[19] Rick Kufrin. Perfsuite: An accessible, open source performance analysis environment for Linux. In *6th International Conference on Linux Clusters: The HPC Revolution*, Chapel Hill, NC, April 2005.

[20] Bernd Mohr, Allen D. Malony, Sameer Shende, and Felix Wolf. Design and prototype of a performance tool interface for OpenMP. *The Journal of Supercomputing*, 23(1):105–128, August 2002.

[21] Matthias S. Müller, Andreas Knüpfer, Matthias Jurenz, Matthias Lieber, Holger Brunst, Hartmut Mix, and Wolfgang E. Nagel. Developing scalable applications with Vampir, VampirServer and VampirTrace. In Christian Bischof, Martin Bücker, Paul Gibbon, Gerhard Joubert, Thomas Lippert, Bernd Mohr, and Frans Peters, editors, *Parallel Computing: Architectures, Algorithms and Applications*, volume 15 of *Advances in Parallel Computing*, pages 637–644. IOS Press, 2007. ISBN 978-1-58603-796-3.

[22] N. Nethercote and J. Seward. Valgrind: A program supervision framework. *Electronic Notes in Theoretical Computer Science*, 89(2):44–66, 2003.

[23] Open MPI: Open Source High Performance Computing, A High Performance Message Passing Library. `http://www.open-mpi.org/`.

[24] OTF – Open Trace Format. `http://www.tu-dresden.de/zih/otf/`.

[25] Rosa M. Badia Pieter Bellens, Josep M. Perez and Jesús Labarta. CellSs: a programming model for the Cell BE architecture. In *ACM/IEEE Supercomputing SC 2006*, November 2006.

[26] Judit Planas, Rosa M. Badia, Eduard Ayguadé, and Jesus Labarta. Hierarchical task based programming with StarSs. *International Journal of High Performance Computing Applications*, 23(3):284–299, August 2009.

[27] Rogue Wave Software AB. Acumem performance productivity tools. `http://www.acumem.com/`.

[28] Scalasca: Scalable parallel performance analysis of large-scale applications. `http://www.scalasca.org/`.

[29] S. S. Shende and A. D. Malony. The TAU parallel performance system. *International Journal of High Performance Computing Applications*, 20(2):287–331, 2006.

[30] The Vampir event trace visualization tool. `http://www.vampir.eu/`.

[31] VampirTrace runtime measurment system. `http://www.tu-dresden.de/zih/vampirtrace/`.

[32] J. Vetter and C. Chambreau. mpiP: Lightweight, scalable MPI profiling. `http://mpip.sourceforge.net/`.

[33] Josef Weidendorfer, Markus Kowarschik, and Carsten Trinitis. A tool suite for simulation based analysis of memory access behavior. In *ICCS 2004: 4th International Conference on Computational Science*, volume 3038 of *LNCS*, pages 440–447. Springer, 2004.

[34] F. Wolf and B. Mohr. Automatic performance analysis of hybrid MPI/OpenMP applications. *Journal of Systems Architecture*, 49(10-11):421–439, 2003. Special Issue "Evolutions in parallel distributed and network-based processing".