Performance Tools for the Grid: State of the Art and Future *
APART White Paper

Version 1.0, 7.01.04

Michael Gerndt, Roland Wismüller
Institut für Informatik, LRR
Technische Universität München

Zoltán Balaton, Gábor Gombás, Péter Kacsuk, Zsolt Németh, Norbert Podhorszki
MTA SZTAKI
Hungarian Academy of Sciences

Hong-Linh Truong
Institute for Software Science
Universität Wien

Thomas Fahringer
Institut für Informatik
Universität Innsbruck

Marian Bubak
Institute for Computer Science
ICS-AGH, CYFRONET

Erwin Laure
European Organization for Nuclear Research

Thomas Margalef
Computer Science Department
Universitat Autonoma de Barcelona

Abstract

This white paper is aimed at creating a directory of existing performance monitoring and evaluation tools. The detailed categorization enables finding relevant properties, similarities and differences, and comparing the tools. The paper is neutral: there are no comments or assessment. The catalogue helps grid users, developers, and administrators in finding an appropriate tool according to their requirements.

The white paper is intended to be updated by the APART community until the end of the APART-2 project and possibly beyond. Grid monitoring and performance analysis tool developers are supposed to check the categorization of their product and modify it if they find anything incorrect in our classification. Moreover, they are expected to send update messages of new versions and prototypes and the white paper will be updated accordingly. Finally, feedback about practical experiences learned from deploying and testing these tools are intended to be added to the paper in the future so that it will show a real classification among development trends.

*This work is funded by the European Commission via the working group on Automatic Performance Analysis: Real Tools (APART), www.fz-juelich.de/apart

Contents

1 Introduction

With the advance of grid computing techniques it is an arising question how the quality or performance of the grid infrastructure and the grid applications can be measured. This information is essential for end-users, developers and administrators. Yet, there is no widely accepted and deployed technique that can solve all aspects of this problem.

In fact, there is no clear definition for grid performance. The term “grid performance” should characterize a large-scale heterogeneous and dynamic resource sharing mechanism, including the resources that are offered and the applications that consume them from quantitative, qualitative and economic point of view. Needless to say, there is no such characterization that is able to handle the diversity and heterogeneity of resources, the large number and dynamism of applications, the lack of universal metrics, the different expectations of application writers and resource owners, the intelligent processing and presentation of performance related data and so on. While performance evaluation of parallel and distributed systems is well investigated and there exist practical solutions, in most cases these techniques cannot be transferred directly to grids. These years the real meaning of grid performance is being explored as different research communities introduce novel approaches to performance monitoring and evaluation.

The numerous approaches yield several different tools. Some of them are fully grid enabled, some are just labeled so, some try to give a whole solution, some others just focus on a partial problem. Despite the technical differences, at user level they may turn out to be equally useful, may realize exactly the same functionalities or may be complementary. All these practical questions are unanswered and in fact, hard to be answered due to the diversity of principles, techniques, supported systems, realizations, target communities and applications.

This white paper is aimed at creating a directory of existing performance monitoring and evaluation tools. The detailed categorization enables finding relevant properties, similarities and differences, and comparing the tools. The paper is neutral: there are no comments or assessment. The catalogue helps grid users, developers, and administrators in finding an appropriate tool according to their requirements.

The white paper is intended to be updated by the APART community until the end of the APART-2 project and possibly beyond. Grid monitoring and performance analysis tool developers are supposed to check the categorization of their product and modify it if they find anything incorrect in our classification. Moreover, they are expected to send update messages of new versions and prototypes and the white paper will be updated accordingly. Finally, feedback about practical experiences learned from deploying and testing these tools are intended to be added to the paper in the future so that it will show a real classification among development trends.

2 Classification

A natural question arises: There are numerous existing performance tools, but, what makes some of them a ”grid performance tool”? There is no commonly accepted definition for grids. Grid computing is typically defined by usage scenarios that span a wide range. The initial idea of grid computing described a large-scale resource sharing mechanism that virtually unifies thousands of resources across continents yielding a dynamic heterogeneous computing environment. Since then several other scenarios were labeled as grid including company level resource dispatchers, geographically distributed clusters, heterogeneous clusters and so on. While these settings may conform all characteristics of grid computing - they can also be realized without the necessary abstractions thus, lacking the ability to adapt to different conditions.

This white paper is not aimed at judging the grid conformity of the tools. Rather, a set of characteristics are listed here that are required or recommended for a tool to support grid performance analysis. This summary is not taxative rather, just highlights the sensitive points. Clear and unambiguous classification is not really possible since the importance of some features are application (and user) dependent. For instance, in most cases performance evaluation does not require security measures, in some grid applications security can be a major issue. No tools are excluded due to the lack of some characteristics, the degree of conformity shows the versatility of a certain approach. Similarly, holes among the features can show the cases where the tool fail to provide adequate service. Whether a certain tool is really proper and powerful for grid performance evaluation, it can be proven in real-life practice only.

The detailed features of each tool are summarized in Chapter 4.

2.1 Target communities

The term ”grid environment” covers more than just a technical infrastructure, users and developers. Therefore, grid performance analysis has more aspects and should support some or all of the following target groups:

- Grid end users
- Grid application developers
- Grid middleware developers
- System administrators
- Components of the grid middleware

2.2 Functionality

Performance analysis covers two levels:

- Data acquisition: instrumentation, monitoring, preprocessing, data delivery, etc.
- Data processing: analysis, presentation, visualization, problem detection, problem location, control, etc.

Grid performance analysis (categorized as monitoring, analysis, steering and visualization in this paper) poses special requirements and additional tasks at both levels (See details in [17].)

1. Grid performance is determined by a certain resource configuration and an application

- Data acquisition should include both application and infrastructure monitoring
- Grid monitors should cope with diverse and multiple level IDs of jobs and resources belonging to a certain application

2. Grid applications cannot be replayed since the collection of resources they utilize is dynamic and diverse.

   - Performance analysis should be on-line or semi-on-line, in some cases preferably, interactive
   - Performance analysis should have active feedback (control) to the system

3. Observation of grid applications may involve tremendous data volume

   - Monitors should have intelligent and configurable filtering, preprocessing, converting features
   - Presentation of performance data should include highly abstract visualization, feature extraction and intelligent human interfaces

4. The complexity of grid infrastructure introduces more possible performance flaws

   - Analysis techniques should be able to automatically detect and spot performance problems by their symptoms

5. The grid is substantially heterogeneous and dynamic with diverse types of applications

   - Since there is little a human observer can do, tools should be automated and intelligent as much as possible
   - Analysis and monitoring tools should provide conversion between different performance metrics
   - Tools should be able to be customized with respect to application, metrics, controls, etc.

2.3 Architecture and interfaces

Performance analysis tools (especially those at the data acquisition level) are not independent from the grid infrastructure. Architecture and interfaces describe how the tools are connected to the grid substrate or integrated with them, what kind of data they receive and provide, how they can interoperate what services they provide and other issues. Grid enabled tools must have a clear architecture and interface definition otherwise they fail to operate in a dynamic heterogeneous environment.

- Grid enabled performance tools may have resource or service oriented approach

- A tool can be designed to conform various requirements that appear in their organization as well. Typical organization principles are hierarchical and flat (peer-to-peer), centralized and distributed.
- Some tools may be deployed together with the grid infrastructure whereas others should have the ability to be installed on-demand.
- The multiple domain property of the grid poses severe security requirements as well.
- Tools must have a well defined interface for the operations, the data format and the services they support.

3 Selected Tools

This section outlines the tools compared in the tables in Section 4. To meet the goal to give an overview of a large set of tools, we restricted the description of each tool to the most essential information. More detailed information can be found in the referenced downloadable paper and at the given web site.

3.1 Askalon Visualization Diagrams

The ASKALON Visualization Diagrams offer a rich set of 2- and 3-dimensional diagrams for post-mortem and on-line visualization of basically arbitrary data, in particular performance and result data. These diagrams are based on an open Java interface (with an option to import XML data) that can be used by external tools to automatically display data without user interaction. Among others, scalable visualization is supported that reflects deep architecture hierarchies ranging from CPUs to Grid infrastructures and their relation to Grid applications. Zoom in/out features are provided for flexible and hierarchical architecture and application views.

Institution:	University of Vienna, University of Innsbruck
Key persons:	Thomas Fahringer
Web page:	`www.par.univie.ac.at/project/askalon/visualization`
Reference:	[1]
Usage:	local testbed
Availability:	Source on request

3.2 Condor Hawkeye

Hawkeye provides mechanisms for collecting, storing, and using information about computers. A hawkeye system can be used to monitor various attributes of a collection of systems. The system includes a set of modules that monitor attributes in a Condor pool.

Institution:	University of Wisconsin - Madison
Key persons:	Miron Livny
Web page:	www.cs.wisc.edu/condor/hawkeye
Reference:	
Usage:	US/CMS test bed
Availability:	Source

3.3 DIMEMAS

Dimemas is a performance prediction tool for message-passing applications. It provides a prediction of the performance of a parallel application on a target machine characterized by a set of performance parameters. Dimemas generates trace files that can be visualized using Paraver or Vampir. In the DAMIEN project it has been extended to predict the behavior on Grid environments.

Institution:	CEPBA European Center for Parallelism of Barcelona UPC
Key persons:	Jesus Labarta
Web page:	www.cepba.upc.es/dimemas
Reference:	[4]
Usage:	Local Testbed
Availability:	Commercial

3.4 EDG Data Access Prediction

The EDG *Replica Optimization Service (ROS)* provides a data access prediction that estimates the time (in seconds) required to access (replicated) data from any location on the grid. It is based upon the *EDG Network Cost Estimation Service* and performance information of the storage as provided e.g. by the EDG Storage Element or the CrossGrid Data Access Estimator. Estimates are computed for entire files or parts of a file. The performance data is used for replica selection and resource scheduling.

Institution:	CERN, ITC-irst, PPARC
Key persons:	Kurt Stockinger, Peter Kunszt
Web page:	cern.ch/edg-wp2/optimization
Reference:	[7]
Usage:	EDG
Availability:	Source

3.5 EDG Network Cost Estimation Service

Network Cost Estimation Service is a grid service estimating the cost of grid network action. This cost is used by grid middleware services (e.g. Replica Optimization Service) to optimize network access/data transfer. It is based on network monitoring data collected by NMA and stored in the information system. Modular architecture enables access to different data storages, like R-GMA, LDAP or raw files, but also the usage of different cost models reflecting the needs of the concrete application.

Institution:	CNRS, INRIA
Key persons:	Robert Harakaly, Franck Bonnassieux, Pascale Primet
Web page:	`ccwp7.in2p3.fr/nces`
Reference:	[21]
Usage:	EDG
Availability:	Source/Binary

3.6 EDG Logging and Bookkeeping Service

The *Logging and Bookkeeping Service (LB)* is a component of the EDG *Workload Management System (WMS)* responsible for reliably gathering and storage of job life cycle events. LB receives (from the various components of the WMS) and stores job events which are further processed to give a high-level view on the job state (such as *running*, *terminated*, etc.). The LB is a distributed service and information from the LB may be either queried directly or streamed into R-GMA whose notification mechanisms can be exploited.

Institution:	CESNET, INFN
Key persons:	Zdeněk Salvet, Aleš Křenek, Luděk Matyska
Web page:	`server11.infn.it/workload-grid`
Reference:	[14]
Usage:	EDG, LCG, CrossGrid
Availability:	Source

3.7 Ganglia

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. It is based on a hierarchical design targeted at federations of clusters, relies on a multicast-based listen/announce protocol to monitor state within clusters and uses a tree of point-to-point connections amongst representative cluster nodes to federate clusters and aggregate their state. Data is represented in XML and compressed using XDR. The Ganglia Web Frontend can be used to inspect for example CPU utilization in the last hour or last month.

Institution:	University of California Berkeley
Key persons:	Matt Massie
Web page:	ganglia.sourceforge.net
Reference:	[3]
Usage:	Berkeley Millenium Grid (monitor.millennium.berkeley.edu/) Titech Grid (bata.cc.titech.ac.jp/ganglia/)
Availability:	Source

3.8 GRaDS/AutoPilot

Autopilot is a distributed monitoring and tuning system. It is used in the GRaDS system to monitor performance contracts via application level autopilot sensors. Fuzzy logic decision support is used to detect performance contract violation. In case of contract violation applications can be dynamically tuned or rescheduled.

Institution:	University of Illinois at Urbana-Champaign
Key persons:	Daniel Reed
Web page:	www-pablo.cs.uiuc.edu/Project/Autopilot
Reference:	[24]
Usage:	GRaDS
Availability:	Source

3.9 G-PM/OCM-G

The Crossgrid project is currently developing an OMIS-compliant application monitor based called OCM-G. It provides configurable online monitoring via a central manager which forwards information requests to the local monitors. G-PM is a graphical performance analysis tool that allows to request standard performance metrics as well as user-defined metrics at runtime. The measured data are periodically transferred from the monitor to the front end and visualized via various performance diagrams.

Institution:	Institute for Computer Science ICS-AGH and CYFRONET, Technische Universität München
Key persons:	Marian Bubak, Wlodzimierz Funika, Roland Wismüller
Web page:	grid.fzk.de/CrossGrid-WP2
Reference:	[9]
Usage:	Crossgrid
Availability:	Source

3.10 GridICE

GridICE is a monitoring infrastructure targeting Grid systems. It enables monitoring of the distributed resources and collection of a rich set of measurements. These measurements are collected through the Grid Information Service and then presented with a web graphic interface in different aggregation dimensions: the Grid Operations Center view, the Virtual Organization view and the Site view. Detection&Notification service and network statistics are being included. GridICE is developed by the INFN within the European DataTAG project, Work Package 4.

Institution:	Istituto Nazionale di Fisica Nucleare
Key persons:	Sergio Andreozzi, Sergio Fantinel, Gennaro Tortone
Web page:	`grid.infn.it/gridice`
Reference:	
Usage:	CMS-LCG0, INFN-GRID, LCG (Release 2)
Availability:	Source (INFN license, BDS-like)

3.11 GridMon

GridMon is a network performance monitoring toolkit to identify faults and inefficiencies. The toolkit is composed of a set of tools that are able to provide measures concerning different aspects related to network performance: Connectivity, Inter-packet Jitter, Packet loss, Round Trip Time (RTP), and TCP and UDP throughput.

Institution:	CCLRC Daresbury Laboratory
Key persons:	Mark Leese
Web page:	`gridmon.dl.ac.uk`
Reference:	[16]
Usage:	UK e-Science
Availability:	None

3.12 GridRM

GridRM is a resource monitoring system, based on GMA. At each grid site, a GridRM Gateway controls access to information of local resources and provides a mechanism for the user to query that information. GridRM actually accesses data from a variety of monitoring services such as MDS, SNMP and provides that data to clients in a homogeneous view. GridRM offers a Web client for the user to control and conduct the monitoring and employs a relational database to store monitoring information and general site metadata.

Institution:	Distributed Systems Group, Institute of Cosmology and Gravitation, University of Portsmouth
Key persons:	Mark Baker, Garry Smith
Web page:	`gridrm.org`
Reference:	[5]
Usage:	GridRM Testbed (UK, Australia, etc.)
Availability:	Not Available yet

3.13 GRM/PROVE

GRM and PROVE are trace based application performance monitor and visualization tools for message passing programs. GRM uses the Mercury grid monitoring system to deliver trace data to the host of visualization. PROVE visualizes trace information on-line during the execution of the grid applications. Several statistics help the user to identify bottlenecks in the program. The application targets are message passing programs running on one grid resource (cluster or supercomputer), however, trace collection from several resources (e.g. metacomputing) is also possible by using Mercury monitor.

Institution:	SZTAKI, Hungarian Academy of Sciences
Key persons:	Peter Kacsuk, Norbert Podhorszki
Web page:	`www.lpds.sztaki.hu`
Reference:	[18]
Usage:	EU DataGrid, Hungarian SuperGrid
Availability:	DataGrid version: Source with EDG License

3.14 Lemon

Lemon is a scalable and flexible fabric monitoring system. Distributed clients launch and control local sensors, schedule measurements, collect data and send them to one or several repositories. The repository stores samples in a database (Oracle or flat file). The system provides sensors for common performance and exception monitoring. Other sensors can be plugged easily. A SOAP based interface is available to query and subscribe to measurements. The tool is running in production at CERN at over 1600 nodes. It is also used in EU DataGrid and by GridICE.

Institution:	CERN
Key persons:	Sylvain Chapeland, David Front, Jan van Eldik
Web page:	`www.cern.ch/lemon`
Reference:	
Usage:	EDG
Availability:	Source

3.15 MapCenter

MapCenter is a flexible monitoring system and presentation layer of the services and applications available on a Grid. It incorporates automatic resources discovery from MDS, R-GMA, or Web browsing as well as stealth monitoring techniques to avoid logs on end systems and ease the crossing of firewalls. Backends for several Grid services (storage, replication systems, etc.) exist. The dynamic discovery mechanisms, and the transparent and efficient monitoring techniques allowed to deploy it quickly on numerous environments.

Institution:	CNRS, INRIA
Key persons:	Franck Bonnassieux
Web page:	mapcenter.in2p3.fr
Reference:	[8]
Usage:	EDG, DataTAG, CrossGrid, LCG, GridIreland, PlanetLab, L-Bone Atlas Grid, E-Toile, CEOS Grid, Nanyang Campus Grid, etc.
Availability:	Source

3.16 MDS-2/MDS-3

MDS is the Grid information service used in the Globus Toolkit. It uses an extensible framework with a hierarchical structure for managing static and dynamic information about the status of Grid components generated by Information Providers. Index Services provide an aggregation service of lower level data using a soft-state registration protocol and caching to minimize the transfer of un-stale data. MDS2 is built on top of LDAP, MDS3 is based on the GT3 Information Services component using Service Data Elements defined in OGSA.

Institution:	ANL, USC/ISI
Key persons:	Karl Czajkowski, Ian Foster, Carl Kesselman
Web page:	www.globus.org
Reference:	[11]
Usage:	In many Globus based grids
Availability:	Source

3.17 Mercury

The Mercury Monitor is designed to satisfy requirements of grid performance monitoring: it provides monitoring data represented as metrics via both pull and push access semantics and also supports steering by controls. It supports monitoring of grid entities such as resources and applications in a generic, extensible and scalable way. Its architecture is based on the GGF GMA, and implemented in a modular way with emphasis on simplicity, efficiency, portability and low intrusiveness on the monitored system.

Institution:	MTA SZTAKI
Key persons:	Zoltán Balaton, Gábor Gombás, Péter Kacsuk
Web page:	`www.gridlab.org/WorkPackages/wp-11`, `www.lpds.sztaki.hu`
Reference:	[6]
Usage:	EU GridLab, EU Datagrid, Hungarian SuperGrid
Availability:	Source

3.18 Nagios

Nagios is a host and service monitoring system operating through external "plugins" which publish status information to Nagios. In case problems are detected, notifications are sent in a variety of ways. Status information, historical logs, and reports can be accessed via a web browser. Nagios features include the monitoring of network services (SMTP, POP3, HTTP, NNTP, PING, etc.), host resources (processor load, disk and memory usage, running processes, log files, etc.), and environmental factors such as temperature.

Institution:	
Key persons:	Ethan Galstad
Web page:	`www.nagios.org`
Reference:	[13]
Usage:	local testbed
Availability:	Source

3.19 Netlogger

NetLogger (*Net*worked Application *Logger*), is a set of tools for monitoring the behavior of all the elements of the application-to-application communication path, applications, operating systems, hosts, and networks. It includes tools for generating timestamped event logs to provide detailed end-to-end application and system level monitoring; and tools for visualizing the log data and real-time state of the distributed system. Applications are instrumented; system monitoring is based on standard UNIX and networking tools.

Institution:	LBNL
Key persons:	Brian Tierney
Web page:	www-didc.lbl.gov/NetLogger
Reference:	[22]
Usage:	Globus, EDG, Grid3 ...
Availability:	Source

3.20 NMA—EDG Network Monitor Architecture

NMA is built upon a set of basic network monitoring tools: *Pinger*, *Iperf* and *UDPmon*. Measurements are scheduled using the *Probes Coordination Protocol (PCP)*. Data collected by this layer are published in the MDS or R-GMA systems and graphically displayed via *MapCenter* or the *network archive browser* for archived data. The *Network Cost Estimation Service (NCES)* uses this data to produce a site matrix indicating the costs for data transfer between individual sites using an appropriate pluggable cost model.

Institution:	CNRS
Key persons:	Franck Bonnassieux, Robert Harakaly
Web page:	`ccwp7.in2p3.fr`
Reference:	[2]
Usage:	EDG
Availability:	Source

3.21 NWS

Network Weather Service (NWS) is a distributed system for producing short-term performance forecasts based on historical performance measurements. NWS provides a set of system sensors for periodically monitoring end-to-end TCP/IP performance (bandwidth and latency), available CPU percentage, and available non-paged memory. Based on collected data, NWS dynamically characterizes and forecasts the performance of network and computational resources.

Institution:	University of California, Santa Barbara
Key persons:	Rich Wolski, Martin Swany
Web page:	`nws.cs.ucsb.edu`
Reference:	[25]
Usage:	NSF Middleware Initiative (NMI) Grid, NPACI Grid, GrADS Testbed, etc.
Availability:	Source

3.22 R-GMA

R-GMA is an implementation of the GGF *Grid Monitoring Architecture (GMA)* based on a relational model. It provides the user with an SQL-like query language hiding the details of the underlying producer/consumer model. R-GMA offers a global view of the information as if each Virtual Organization had one large relational database. It is being used both for information about the grid (primarily to find out what services are available at any one time) and for application monitoring.

Institution:	PPARC
Key persons:	Steve Fisher (mailto:s.m.fisher@rl.ac.uk)
Web page:	`www.r-gma.org`
Reference:	[10]
Usage:	EDG
Availability:	Source

3.23 Scalea-G

SCALEA-G is a unified monitoring and performance analysis system for the Grid. SCALEA-G provides an OGSA-based infrastructure for conducting online monitoring and performance analysis of a variety of Grid computational and network resources, and applications. Both push and pull model are supported, providing flexible and scalable monitoring and performance analysis for the Grid. Source code and dynamic instrumentation are exploited to perform profiling and tracing of Grid applications.

Institution:	University of Vienna, University of Innsbruck
Key persons:	Thomas Fahringer, Hong-Linh Truong
Web page:	www.par.univie.ac.at/project/scaleag
Reference:	[23]
Usage:	local testbed
Availability:	Source on request

3.24 VAMPIR

VAMPIR is a tool for performance analysis of message passing applications based on program traces. VAMPIR is currently being extended in the DAMIEN project for multisite Grid applications. The applications are coupled via MpCCI implemented on top of PACX-MPI. In addition, enhancements to VAMPIR allow to analyze the communication infrastructure.

Institution:	TU Dresden
Key persons:	Wolfgang Nagel
Web page:	www.hlrs.de/organization/pds/projects/damien
Reference:	[12]
Usage:	local testbed
Availability:	Commercial

3.25 Virtue

Virtue is a collaborative, immersive visualization system with real-time performance measurement and adaptive control of applications on the grid. It constitutes a whole performance evaluation system together with SvPablo for instrumentation, and Autopilot for real-time adaptive control.

Institution:	University of Illinois at Urbana-Champaign
Key persons:	D.A. Reed, E. Scaffer, S. Whitmore, B. Schaeffer
Web page:	www-pablo.cs.uiuc.edu/Software/Virtue/virtue.htm
Reference:	[20]
Usage:	unknown
Availability:	Source and binary

3.26 visPerf

VisPerf is a monitoring and visualization system for grid resources. It is based on sensors that extract information from log files or obtain information via API included in the grid middleware. It was developed in the GridSolve system. The visualization allows the user to connect to NetSolve servers and inspect system information.

Institution:	Kwangju Institute of Science and Technology, Innovative Computing Laboratory, University of Tennessee
Key persons:	Dong Woo Lee, Jack Dongarra
Web page:	www.cs.utk.edu/~leepro/monitor
Reference:	[15]
Usage:	Netsolve
Availability:	Source

4 Tool Comparison

The tools introduced in the previous section are compared according to the classification presented in Section 2. For each table, a glossary is provided as an appendix since the terminology is not always precisely defined in the literature. We marked a column with an 'X' if the tool matches this characteristic. If the tool does not match the characteristic, the characteristic is not applicable to the tool, or it is unknown from the literature whether the tool has it, the column is marked with a '-'.

The tables are based on the three main areas specified in Section 2. The target community and the main functionality of the tools, i.e., monitoring, analysis, prediction, steering, and visualization, can be found in Table 1. Table 2 provides a more detailed classification according to the functionality. The provided data, the usage mode, and the application languages are presented in Table 3. The different instrumentation techniques for resource and application monitoring are given in Table 4. Architectural issues, such as the grid type, the tool organization, its scalability, deployment issues and security aspects, can be found in Table 5. Table 6 provides a classification according to the access interfaces and specifies whether the tool is implemented as a service.

	Target Community					Functionality				
	Grid end-user	Application developer	Middleware components	Middleware developers	System administrator	Performance monitoring	Performance analysis	Performance prediction	Performance steering	Performance visualization
Askalon-Visualization	-	X	X	X	-	-	-	-	-	X
Autopilot	X	-	-	-	-	X	-	-	X	-
Dimemas	-	X	-	-	-	-	-	X	-	-
EDG Data Access Prediction	X	-	X	-	-	-	-	X	-	-
EDG Logging and Bookkeeping	X	-	X	-	-	X	-	-	-	-
EDG Network Estimation Service	-	-	X	-	-	X	-	X	-	-
Ganglia	X	-	-	X	X	X	-	-	-	-
G-PM / OCM-G	X	X	X	X	-	X	X	-	-	X
GridICE	X	X	X	X	X	X	X	-	-	X
GridMon	-	-	X	-	X	X	-	-	-	-
GridRM	X	-	X	-	X	X	-	-	-	X
GRM/Prove	X	X	-	X	-	X	X	-	-	X
Hawkeye	X	-	-	-	X	X	-	-	-	-
Lemon	X	-	-	-	X	X	X	-	-	X
MapCenter	X	-	-	-	X	X	-	-	-	X
MDS	X	-	X	-	X	X	-	-	-	-
Mercury	X	X	X	X	X	X	-	-	X	-
Nagios	X	-	-	X	X	X	-	-	-	X
NetLogger	X	X	-	X	-	X	X	-	-	X
NMA	X	-	X	-	X	X	-	-	-	-
NWS	X	-	X	-	-	X	-	X	-	-
RGMA	X	-	X	-	X	X	-	-	-	-
Scalea-G	-	X	X	X	-	X	X	-	-	X
Vampir	-	X	X	X	-	X	X	-	-	X
Virtue	X	X	-	-	-	-	-	-	-	X
VisPerf	X	-	-	-	-	X	-	-	-	X

Table 1: Classification according to user community and functionality. The glossary can be found in Appendix A.

	Performance Monitoring			Performance Analysis						Performance Steering				Performance Visualization			
	Sampling	Tracing	Profiling	Profiling analysis	Trace file analysis	Overhead analysis	Property analysis	Bottleneck detection	User-defined metrics	Application specific	Automatic library / Middleware tuning	Adaptive compilation	Adaptive re-scheduling	Application-oriented	Resource-oriented	Virtual reality	Portal-based
Askalon-Visualization	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-
Autopilot	-	-	X	-	-	-	-	-	-	X	-	X	X	-	-	-	-
Dimemas	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDG Data Access Prediction	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-
EDG Logging and Bookkeeping	-	X	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
EDG Network Estimation	-	X	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-
Ganglia	X	X	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-
G-PM / OCM-G	X	X	X	X	-	-	-	-	X	-	-	-	-	X	-	-	-
GridICE	X	X	X	-	-	-	-	-	-	-	-	-	-	-	X	-	X
GridMon	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GridRM	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X
GRM/Prove	-	X	-	-	X	-	-	-	-	-	-	-	-	X	-	-	-
Hawkeye	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lemon	X	-	-	-	-	-	-	-	X	-	-	-	-	-	X	-	-
MapCenter	X	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-
MDS	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Mercury	X	X	-	-	-	-	-	-	-	X	X	-	X	-	-	-	-
Nagios	X	X	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-
NetLogger	-	X	X	X	X	-	-	-	-	-	-	-	-	X	X	-	-
NMA	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NWS	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RGMA	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Scalea-G	X	X	X	X	-	X	-	-	X	-	-	-	-	X	X	-	-
Vampir	-	X	-	X	X	-	-	-	-	-	-	-	-	X	-	-	-
Virtue	-	X	-	-	X	-	-	-	-	-	-	-	-	X	X	X	-
VisPerf	-	-	X	-	-	-	-	-	-	-	-	-	-	-	X	-	-

Table 2: Classification according to detailed functionality. The glossary can be found in Appendix B.

	Provided Data						Usage			Supported Languages in Application Monitoring					
				resource-oriented											
	Application-oriented	Job-status oriented	Middleware-oriented	Computing resource	Storage resource	Network resource	Off-line	On-line	Interactive	Language independent	FORTRAN	C	C++	Java	Other languages
Askalon-Visualization	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-
Autopilot	X	-	-	-	-	-	-	X	-	X	-	-	-	-	-
Dimemas	X	-	-	-	-	-	X	-	-	-	X	X	X	-	-
EDG Data Access Prediction	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-
EDG Logging and Bookkeeping	-	X	-	-	-	-	X	X	-	-	-	-	X	X	-
EDG Network Estimation Service	-	-	-	-	-	X	-	X	-	X	-	-	-	-	-
Ganglia	-	-	-	X	X	X	X	-	-	-	-	-	-	-	-
G-PM / OCM-G	X	-	-	X	-	-	-	X	X	-	X	X	X	-	-
GridICE	-	-	X	X	X	-	X	X	-	-	-	-	-	-	-
GridMon	-	-	-	-	-	X	-	X	-	-	-	-	-	-	-
GridRM	-	X	X	X	X	X	-	X	-	-	-	-	-	-	-
GRM/Prove	X	X	-	-	-	-	X	X	-	-	X	X	X	-	-
Hawkeye	-	-	-	X	X	X	-	X	X	-	-	-	-	-	-
Lemon	-	-	-	X	X	X	X	X	-	-	-	-	-	-	-
MapCenter	-	-	X	-	-	-	X	-	-	-	-	-	-	-	-
MDS	X	X	X	X	X	X	-	X	-	X	-	-	-	X	-
Mercury	X	-	-	X	X	X	-	X	X	-	-	X	-	-	-
Nagios	-	-	X	X	X	X	X	X	-	-	-	-	-	-	-
NetLogger	X	-	-	X	X	X	X	X	X	-	-	X	-	X	X
NMA	-	-	-	-	-	X	X	X	X	X	-	-	-	-	-
NWS	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
RGMA	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-
Scalea-G	X	-	X	X	-	X	-	X	-	-	X	X	X	-	-
Vampir	X	-	-	-	-	-	X	-	-	X	-	-	-	-	-
Virtue	X	-	-	-	-	-	X	X	X	X	-	-	-	-	-
VisPerf	-	X	X	X	X	X	-	X	-	-	-	-	-	-	-

Table 3: Classification according to features. The glossary can be found in Appendix C.

	Instrumentation for Resource and Application Monitoring										
	Automatic	Manual	Static	Dynamic	Controllable	Source	Library	Binary	Memory image	Pre-defined	User-defined
Askalon-Visualization	-	-	-	-	-	-	-	-	-	-	-
Autopilot	X	X	X	X	-	X	X	-	-	X	-
Dimemas	-	-	-	-	-	-	-	-	-	-	-
EDG Data Access Prediction	-	-	-	-	-	-	-	-	-	-	-
EDG Logging and Bookkeeping	-	-	-	-	-	-	-	-	-	-	-
EDG Network Estimation Service	-	-	-	-	-	-	-	-	-	-	-
Ganglia	-	-	-	-	-	-	-	-	-	-	-
G-PM / OCM-G	X	X	X	-	X	X	X	-	-	X	X
GridICE	X	-	X	-	-	-	-	X	-	X	-
GridMon	X	-	X	-	-	-	-	-	-	X	-
GridRM	-	-	-	-	-	-	-	-	-	-	-
GRM/Prove	-	X	X	-	-	X	-	-	-	X	X
Hawkeye	X	-	-	X	X	-	-	-	-	X	-
Lemon	-	-	-	-	-	-	-	-	-	-	-
MapCenter	-	-	-	-	-	-	-	-	-	-	-
MDS	-	X	X	-	-	-	-	-	-	-	-
Mercury	X	X	X	-	X	X	X	-	-	-	X
Nagios	-	-	-	-	-	-	-	-	-	-	-
NetLogger	-	X	-	-	-	X	-	-	-	-	X
NMA	-	-	-	-	-	-	-	-	-	-	-
NWS	-	-	-	-	-	-	-	-	-	-	-
RGMA	-	X	-	-	-	X	-	-	-	-	X
Scalea-G	X	X	X	X	-	X	X	-	X	X	X
Vampir	-	-	-	-	X	-	X	-	-	X	X
Virtue	X	X	X	-	X	X	-	-	-	X	-
VisPerf	X	X	X	-	X	-	-	-	-	X	-

Table 4: Classification according to instrumentation support. The glossary can be found in Appendix D.

	Grid Type		Organization		Scalability		Deployment Issues		Security				
	Resource-oriented	Service-oriented	Hierarchical	Peer-to-peer	Centralized analysis	Distributed analysis	Persistent service	Transient service	Authentication	Authorization	Encryption	Firewalls	Private IP addresses
Askalon-Visualization	-	-	-	-	-	-	-	-	-	-	-	-	-
Autopilot	X	-	-	X	X	-	-	-	-	-	-	-	-
Dimemas	X	-	-	-	-	-	-	-	-	-	-	-	-
EDG Data Access Prediction	X	-	-	X	-	X	X	-	X	X	X	X	-
EDG Logging and Bookkeeping	-	X	-	X	-	X	X	-	X	X	X	X	-
EDG Network Estimation Service	X	-	-	-	-	X	X	-	X	X	X	X	-
Ganglia	X	-	X	-	-	X	X	-	X	X	X	X	X
G-PM / OCM-G	X	-	X	-	X	X	X	X	X	X	-	X	X
GridICE	X	X	X	-	X	-	X	-	-	-	-	X	-
GridMon	-	X	-	X	-	X	X	-	-	-	-	-	-
GridRM	X	-	X	-	-	-	X	-	-	-	-	-	-
GRM/Prove	X	X	X	-	X	-	-	X	-	-	-	-	-
Hawkeye	X	-	X	-	-	-	X	-	-	-	-	-	-
Lemon	X	-	-	-	X	-	-	-	-	-	-	-	
MapCenter	-	X	X	-	-	-	X	-	-	-	-	X	-
MDS	X	X	X	-	X	-	X	-	X	X	X	X	-
Mercury	X	-	X	-	X	X	X	-	X	X	-	X	X
Nagios	X	-	-	X	-	X	X	-	X	X	X	X	X
NetLogger	X	X	X	-	X	-	-	X	-	-	-	-	-
NMA	X	-	X	X	-	X	X	-	-	-	-	-	-
NWS	X	-	X	-	X	-	X	-	-	-	-	-	-
RGMA	X	X	X	X	X	X	X	-	X	X	X	X	-
Scalea-G	-	X	X	-	X	-	X	X	X	X	-	-	-
Vampir	X	-	X	-	X	-	-	-	-	-	-	-	-
Virtue	-	-	-	-	-	-	-	-	-	-	-	-	-
VisPerf	X	-	-	X	X	-	X	-	-	-	-	X	X

Table 5: Classification according to architectural issues. The glossary can be found in Appendix E.

	Data Format			Supported Operations			Service		Languages Supported by API					
	Standard	Self-describing	Non-standard	Query	Subscribe	Controlling	OGSI compliant	Others	Language ind.	FORTRAN	C	C++	Java	Other languages
Askalon-Visualization	-	X	X	-	-	-	-	-	-	-	-	-	X	-
Autopilot	-	-	X	-	X	-	-	-	-	-	X	X	-	-
Dimemas	-	-	X	-	-	-	-	-	-	-	-	-	-	-
EDG Data Access Prediction	-	-	X	X	-	-		X	X	-	-	-	-	-
EDG Logging and Bookkeeping	-	-	X	X	-	-	-	X	-	-	X	X	-	-
EDG Network Estimation Service	-	-	X	X	-	-	X	-	X	-	-	-	-	-
Ganglia	-	X	-	X	-	X	-	X	-	-	-	-	-	X
G-PM / OCM-G	-	-	X	X	X	X	-	X	-	-	X	X	-	-
GridICE	X	-	-	X	X	X	-	MDS/Postgres	X	-	-	-	-	-
GridMon	-	-	X	-	-	-	X	-	-	-	-	-	-	-
GridRM	-	X	-	X	-	-	-	X	-	-	-	-	-	-
GRM/Prove	-	-	X	-	-	-	X	-	-	X	X	X	-	-
Hawkeye	-	-	X	-	-	-	-	-	-	-	-	-	-	-
Lemon	-	-	X	X	X	-	-	-	-	-	X	-	-	X
MapCenter	X	-	-	X	X	X	-	Web Service	X	-	-	-	-	-
MDS	X	X	-	X	X	-	X	X (MDS2)	X	-	X	-	X	X
Mercury	-	-	X	X	X	X	-	X	-	-	X	-	X	X
Nagios	-	-	X	X	X	X	-	Hierar. Daemons	-	-	-	-	-	X
NetLogger	-	-	X	X	X	X	-	mySQL	-	-	X	X	X	-
NMA	X	-	-	X	X	X	-	RGMA Web service	X	-	-	-	-	-
NWS	-	-	X	X	-	X	-	X	-	-	X	X	X	-
RGMA	X	-	-	X	X	X	-	Web service	X	-	-	-	-	-
Scalea-G	-	X	-	X	X	X	X	Web service	-	-	X	X	X	-
Vampir	-	-	X	-	-	-	-	-	-	-	-	-	-	-
Virtue	-	X	-	-	X	X	-	-	X	-	-	-	-	-
VisPerf	X	-	-	-	X	-	-	-	-	-	-	-	-	-

Table 6: Classification according to provided interfaces. The glossary can be found in Appendix F.

5 Summary and Future Plan

Although the set of tools covered in the white paper is far not complete, this collected set already demonstrates that there is a large set of available tools for monitoring and analyzing the various Grid systems. Yet, surprisingly, Grid users complain about the lack of such tools. This contradiction is one of the main reasons why we have decided to create this white paper. Looking at the tables and analyzing them we can find some answers for this problem.

First of all, individual tools typically address one or only few aspects of the many monitoring and performance related issues. Therefore applying only one of these tools is typically not enough to make all the Grid users satisfied. We have classified Grid monitor users into five categories and looking at Table 1 it is clear that the large majority of the tools support only 2-3 types of Grid monitor users. A clear consequence is that either these tools should be completely interoperable or at least several of them somehow should be integrated into families that together can serve all the types of Grid monitor users and provide most of the functionalities and features that are required (see the functionality and feature tables).

Already some steps have been made for creating such families. Here are some examples.

1. In the EU DataGrid project many monitoring and performance tools are integrated together.

    - EDG Data Access Prediction
    - Lemon
    - GridICE
    - EDG Network Monitor Architecture
    - MapCenter
    - EDG Logging and Bookkeeping Service
    - EDG Network Cost Estimation Service
    - R-GMA
    - GRM/PROVE
    - Mercury

    These tools together cover all the types of Grid monitor users and provide most of the functionalities and features that are required. The only problem is that the common platform on which these tools work together is very limited (Red Hat Linux 7.2) and therefore requires a homogeneous Grid. This is a very strong limitation that significantly restricts the usage of this toolset family.

2. GRM/PROVE and Mercury

    This family supports all the types of Grid monitor users and provide a significant set of the functionalities and features that are required. This family was already tested on heterogeneous Grid test-beds like the Hungarian SuperGrid where supercomputers and clusters with various operating systems were working together. This family currently does not support the new OGSA/OGSI standards although one of its components, PROVE, is already available as GT3 service.

3. SCALEA-G and ASKALON

   This family also supports all the types of Grid monitor users and provides a significant set of the functionalities and features that are required. The advantage of this family is that it is based on the new OGSA/OGSI standards and hence its components can be accessed as Grid services. The only problem with this family that it was not tested yet in real Grid test-beds and hence its robustness is not guaranteed yet.

The examples above illustrate that the interoperability of these tools would be extremely important. If we look at Fig. 6 it is clear again that the integration of these tools in their current status is very difficult because most of them use non-standard data formats and only very few are OGSI compliant. This is the main reason why they are so separated and why the Grid monitor users who would like to use such integrated families feel that there is no really usable Grid monitoring tool available.

This white paper is the first step on the road to create and provide such usable Grid monitoring families. Based on the white paper Grid monitor developers can create alliances to develop such families. Analyzing these tables they can find those complimentary tools that can potentially be integrated into such families and they can initiate such joint projects in the future. These future projects can help in the creation of generally accepted Grid monitoring standards, too.

Looking at the GGF working and research group structure we were surprised that currently there is no working or research group in GGF which deals with the problem of standardizing the Grid monitoring and performance analysis tools. This document clearly shows that such a group is highly needed and therefore, we will propose to the GGF leadership to establish such a group.

An important step into this standardization direction is the creation of the "Performance Analysis / Grid Monitoring" working group of the GRIDSTART project (`www.gridstart.org/index.shtml`) that combines the research results of the ten first-wave European Grid projects. Our aim with this document is also to assist the work of this working group and strongly collaborate with it in order to establish the necessary standards and to further develop this registry of the Grid monitoring and performance analysis tools for the benefit of the whole Grid community.

The potential Grid monitor users can also beneficially study the tables of this document in order to find those tools and tool-families that best fit to their requirements and then they can contact the tool providers. This is the reason why we provide the contact information for every tool in the white paper.

In order to better support the Grid monitor users it would have been useful to provide a quality and performance comparison of the tools covered in the document. Interestingly, though we talk about performance tools, there are no performance metrics, tools and measurement methodologies by which we can compare the performance of these Grid performance tools. Accordingly, when we started to create this document, our first decision was neutrality. We decided not to evaluate the tools, just categorize them due to the problem mentioned above. Another reason was the time constraint. We wanted to produce this document quickly in order to involve the whole Grid community into the further development of the document as soon as possible. We thought that as a starting point it was enough to register and categorize the available Grid monitoring and performance analysis tools and then later an important step forward should be the elaboration the necessary performance metrics, tools and measurement methodologies. We hope that the involvement of the whole Grid community will accelerate the solution of this problem.

As we mentioned before we are aware that the set of Grid monitoring tools covered in the white paper is far not complete. We therefore encourage Grid monitor developers to contact us by sending a short description of their tools in the same form as we did in the white paper, as well as the categorization features of their tools putting the right items into the categorization tables. Similarly, authors of the covered tools are requested to regularly update the description and categorization of their tools when they are further developed.

We are going to place the downloadable electronic version of this white paper on the APART-2 WP3 web page (`www.lpds.sztaki.hu/~zsnemeth/apart`) and monthly update it according to the received updates. Please, send your updates to Zsolt Nemeth (zsnemeth@lutra.sztaki.hu). In this way we hope that the white paper really becomes a state-of-the-art directory of existing Grid monitoring and performance analysis tools that can be beneficially used by the whole Grid community.

A Glossary of Table Target

Grid end-user : A grid end-user is a person who uses the grid to run applications.

Application developer : A person who develops applications to be executed on the grid.

Middleware component : It is a piece of software that is involved in some of the grid architecture layers and provides some specific functionality.

Middleware developer : A person developing some of the components of the grid software layers.

System administrator : A system administrator is a person who is administrating some of the nodes of the grid.

Performance monitoring : Performance monitoring is the action of collecting all the required information to determine the performance of the system, application, etc.

Performance analysis : It is the analysis process to determine the performance of the system, application, etc.

Performance prediction : It is the study carried out to estimate the performance of the system, application, etc. under certain conditions without testing the system under such conditions.

Performance steering : It is the capability of modifying the behavior of the system to improve its performance.

Performance visualization : It is the ability to show the performance of the system, application, etc, by providing a set of charts, diagrams, graphics, etc.

B Glossary of Table Functionality

Performance monitoring, sampling : It is the action of collecting information by taking samples of the required measures.

Performance monitoring, tracing : It is the action of collecting events occurred during the operation of the system and recording them in a trace file.

Performance monitoring, profiling : It is the action of aggregating information in order to provide some summarized information of the behavior of the system.

Performance analysis, profiling analysis : It is the analysis done based on profile information.

Performance analysis, trace file analysis : It is the analysis based on the information recorded in a trace file.

Performance analysis, overhead analysis : It is the analysis based on measuring the time spent on ”non-productive” tasks.

Performance analysis, property analysis : It is the analysis based on measuring some defined performance property.

Performance analysis, bottleneck detection : It is the analysis based on detecting the conditions that slowdown the performance of the system, application, etc.

Performance analysis, user-defined metrics : It is the analysis based on measuring metrics defined by the user.

Performance steering, application specific : The performance steering (analysis and tuning) is directed to improve the application behavior.

Performance steering, automatic library/middleware tuning : The steering is based on tuning the library or middleware parameters.

Performance steering, adaptive compilation : The steering is based on modifying the compilation, e.g., profile-based compilation.

Performance steering, adaptive re-scheduling : The steering is done adapting the scheduling at run-time taking into account the behavior of the system.

Performance visualization, application oriented : The provided visualization represents the behavior of the application.

Performance visualization, resource oriented : The provided visualization represents the resource usage.

Performance visualization, virtual reality : The performance information is presented in a virtual reality environment.

Performance visualization, Portal-based : The performance information is presented in a web portal.

C Glossary of Table Features

Data, application oriented : Data about the internal behavior of an application program and/or its components, e.g. its memory usage or the execution time of a specific routine.

Data, job status oriented : Data about the execution status of a submitted job as a whole, e.g. whether it is still running, or completed.

Data, Grid middleware oriented : Data about the behavior of Grid middleware components, i.e. general software components on the Grid which do belong to specific applications.

Data, resource oriented : Data about the hardware resources in the Grid, e.g. compute nodes, storage devices, or the network.

Computing resource : A hardware resource that offers computing power to the user. A compute resource may allow a user to run arbitrary applications processes or may just offer predefined compute services.

Storage resource : A hardware resource which offers services to store and retrieve data.

Network resource : The term network resources comprises all hardware resources necessary for the communication between computing resources, storage resources, and the user's local computer.

Usage, off-line : A performance tool works in off-line mode if it provides data about an executing system (e.g. an application program) only *after* its execution has terminated.

Usage, on-line : A performance tool works in on-line mode if it provides data about an executing system (e.g. the execution of an application program) while the system still executes. Although there is some inevitable delay in providing the data, the delay is small and at least statistically bounded.

Usage, interactive : A performance tool supports interactive usage, if it operates in on-line mode and, in addition, allows the measurements to be changed while the observed system still executes. This allows the user to interactively refine and/or adapt the measurements.

Language, supported : A performance tool providing application oriented data (Grid middleware oriented data) is said to support a programming language, if the tool can be used to monitor and/or analyze applications (Grid middleware) written in this language.

Language independent : A performance tool providing application oriented data (Grid middleware oriented data) is said to be language independent, if the tool can be used to monitor and/or analyze applications (Grid middleware) written in *any* programming language.

D Glossary for Table Instrumentation

Instrumentation : (1) Instrumentation is the process of modifying a (software and/or hardware) system in order to acquire information about its execution behavior and/or to control its execution behavior. The modification typically consists in the insertion of sensors and/or actuators. (2) The term instrumentation is also used to refer to the sensors / actuators inserted into a system.

Instrumentation, automatic : The instrumentation is performed by the performance tool. The user is either not involved at all, or just specifies his interest in certain data and certain parts of the system under observation.

Instrumentation, manual : The user is fully responsible for inserting the correct instrumentation at the proper places in the system under observation.

Instrumentation, static : The instrumentation is inserted in the system *before* the system under observation starts executing.

Instrumentation, dynamic : The instrumentation is inserted (and may also be removed again) while the system under observation is already executing.

Instrumentation, controllable : The instrumentation is inserted into the system under observation before it starts executing, however, this instrumentation is generic, such that its behavior (e.g. the acquired data) can be widely controlled while the system is executing.

Instrumentation, source code : The instrumentation is inserted into the source code of a software system.

Instrumentation, library : The instrumentation is inserted into programming libraries used by a software system.

Instrumentation, binary : The instrumentation is inserted into the binary code (i.e. the executable file) of a software system.

Instrumentation, memory image : The instrumentation is inserted into the machine code of a software system, after it has been loaded into the main memory of the compute resource executing the software system.

Instrumentation, hardware : The instrumentation consists of additional hardware, which is connected to the hardware resources used by (or being part of) the system under observation.

Instrumentation, operating system : The instrumentation is inserted into (or is by default contained in) the operating system of the compute, storage, or network resources used to execute the (software) system under observation.

Instrumentation, middleware : The instrumentation is inserted into (or is by default contained in) the Grid middleware used to execute the (software) system under observation.

Instrumentation, pre-defined : The places where some instrumentation can be inserted are predefined by the performance tool. It is not possible for the user to insert instrumentation at places not foreseen by the tool.

Instrumentation, user-defined : The performance tool allows the user to insert instrumentation at arbitrary places in the system under observation.

E Glossary of Table Architecture

Resource oriented grid : A grid where the basic functional building blocks are resources

Service oriented grid : A grid where the basic functional building blocks are service providing entities

Hierarchical : Any network-based system that uses client entities to request a specific service, and corresponding server entities to provide the service from another resource on the network

Peer-to-peer : A distributed network architecture may be called a Peer-to-Peer (P-to-P, P2P) network, if the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers). These shared resources are necessary to provide the Service and content offered by the network (e.g. file sharing or shared workspaces for collaboration). They are accessible by other peers directly, without passing intermediary entities. The participants of such a network are thus resource (Service and content) providers as well as resource (Service and content) requestors (Servent-concept). [19]

Centralized : A computing environment or service realized by a single computing element

Distributed : A computing environment or service realized by loosely coupled computing elements

Persistent service : A service that exists independently from the requests

Transient service : A service that is installed or triggered upon a request and terminated after the request has been served

Authentication : A procedure where the identity of an entity is checked

Authorization : A procedure where an authenticated entity is given rights to perform some action

Encryption : An algorithm used to scramble data which makes it unreadable to everyone except the recipient

Firewall : Firewall is a policy based packet filtering middlebox function, typically used for restricting access to/from specific devices and applications. The policies are often termed Access Control Lists (ACLs).

Private IP address : An IP address from a private address space (a set of hosts with no network layer connectivity to the outside world, see RFC 1918.)

F Glossary of Table Interfaces

Standard data format : A data format defined and accepted by a standardization organization

Self-describing data format : A data format defined by a standardized description method

Non-standard data format : A data format that is neither standard nor self-describing.

Query : Realizes pull model data delivery: data are transferred on request

Subscribe : Realizes push model data delivery: data is transferred when they become available.

Steering : See performance steering

OGSI compliant service : A service that conforms the Open Grid Service Infrastructure specification.

References

[1] ASKALON Visualization Diagrams. Institute for Software Science, University of Vienna. www.par.univie.ac.at/project/askalon/visualization/index.html. 8

[2] Eu datagrid deliverable d7.4. edms.cern.ch/document/414132. 16

[3] The Ganglia Distributed Monitoring System: Design, Implementation, and Experience. submitted for publication. ganglia.sourceforge.net/talks/parallel_computing/ganglia-twocol.ps.gz. 11

[4] Rosa M. Badia, Jess Labarta, Judit Gimenez, and Francesc Escale. DIMEMAS: Predicting MPI applications behavior in Grid environments. *Workshop on Grid Applications and Programming Tools (GGF8), 2003*. www.cepba.upc.es/dimemas/docs/dimemas_updt.pdf. 9

[5] Mark Baker and Garry Smith. GridRM: An Extensible Resource Monitoring System. In *Proceedings of the 5th IEEE Cluster Computing Conference (CLUSTER2003)*, pages 207–?, Hong Kong, December 01-04 2003. IEEE Computer Society Press. 13

[6] Z. Balaton and G. Gombás. Resource and Job Monitoring in the Grid. In *Proceedings of the Euro-Par 2003 International Conference*, 2003. www.lpds.sztaki.hu/publications/europar-2003-monitor.pdf. 15

[7] W.H. Bell, D.G. Cameron, L. Capozza, P. Millar, K. Stockinger, and F. Zini. Design of a Replica Optimisation Framework. Technical Report DataGrid-02-TED-021215, EU DataGrid Project, 2002. edms.cern.ch/file/337977/1.7.2/wp2_replicaopt_api.ps. 9

[8] F. Bonnassieux, R. Harakaly, and P. Primet. MapCenter: an Open GRID Status Visualization Tool. In *ISCA 15th International Conference on Parallel and Distributed Computing Systems*, Louisville, Kentucky, USA, September 2002. mapcenter.in2p3.fr/mapcenter.pdf. 14

[9] Marian Bubak, Wlodzimierz Funika, and Roland Wismller. The crossgrid performance analysis tool for interactive grid applications. In *D. Kranzlmller, P. Kacsuk, J. Dongarra, and J. Volkert, editors, Recent Advances in Parallel Virtual Machine and Message Passing Interface, 9th European PVM/MPI Users' Group Meeting, volume 2474 of Lecture Notes in Computer Science, pages 50-60, Linz, Austria, September 2002*, 2002. 11

[10] A. Cooke et al. R-GMA: An Information Integration System for Grid Monitoring. In *Proceedings of the Tenth International Conference on Cooperative Information Systems*, 2003. edms.cern.ch/file/427817/1/COOPIS-2003.pdf. 16

[11] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid information services for distributed resource sharing. In *Proceedings of the 10th IEEE International Symposium on High-Performance Distributed Computing (HPDC-10)*, 2001.
www.globus.org/research/papers/MDS-HPDC.pdf. 14

[12] Edgar Gabriel, Rainer Keller, Peggy Lindner, Matthias S. Müller, and Michael Resch. Software development in the grid: The damien tool-set. In *Proceedings of ICCS2003, St. Petersburg, Russia, June 2nd-4th, 2003*, 2003.
www.hlrs.de/organization/pds/projects/pacx-mpi/papers/iccs03.ps.gz. 17

[13] Ethan Galstad. *Nagios Version 1.0 Documentation*, 2003.
nagios.sourceforge.net/docs/1_0/. 15

[14] Daniel Kouřil et al. Advances in the L&B Grid Job Monitoring Service. In *Terena Network Conference 2004*, 2004. submitted.
lindir.ics.muni.cz/dg_public/lb2.pdf. 10

[15] D. Lee, J. Dongarra, and R. Ramakrishna. Visperf: Monitoring tool for grid computing. In *ICCS 2003, Lecture Notes in Computer Science, Springer Verlag, Heidelberg, Volume 2659, pp. 233-243, 2003*, 2003.
icl.cs.utk.edu/news_pub/submissions/visperf.pdf. 18

[16] Mark Leese and Robin Tasker. GridMon: Grid Network Performance Monitoring for e-Science. *UK e-Science All Hands Meeting, Nottingham UK, 2-4 September 2003*.
rtlin1.dl.ac.uk/~mjl/allhands03/AllHands03GridMon.pdf. 12

[17] Zs. Németh, G. Gombás, and Z. Balaton. Performance evaluation on grids: Directions, issues, and open problems. In *Proceedings of the Euromicro PDP 2004, A Coruna, Spain, IEEE Computer Society Press*, 2004. 6

[18] Norbert Podhorszki and Peter Kacsuk. Presentation and Analysis of Grid Performance Data. In H. Helwagner H. Kosch, L. Böszörményi, editor, *EuroPar 2003 - 9th International Conference, Klagenfurt, Austria*, number 119-126 in LNCS 2790. Springer Verlag, 2003. 13

[19] R. Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Proceedings of the First International Conference on Peer-to-Peer Computing (P2P 01)*, 2001. 32

[20] Eric Shaffer, Daniel A. Reed, Shannon Whitmore, and Benjamin Schaeffer. Virtue: Performance Visualization of Parallel and Distributed Applications. *Computer*, 32(12):44–51, 1999.
citeseer.nj.nec.com/shaffer99virtue.html. 17

[21] K. Stockinger, H. Stockinger, R. Harakaly, P. Primet, and F. Bonnassieux. Replica Access Optimisation in a Data Grid Using Network Cost Estimation Service. *submitted to Journal of Grid Computing*. 10

[22] B. Tierney and D. Gunter. NetLogger: A Toolkit for Distributed System Performance Tuning and Debugging. Technical Report LBNL-51276, LBNL, 2002.
www-didc.lbl.gov/papers/NetLogger.overview.pdf. 15

[23] Hong-Linh Truong and Thomas Fahringer. SCALEA-G: a Unified Monitoring and Performance Analysis System for the Grid. Technical report, Institute for Software Science, University of Vienna, October 2003.
www.par.univie.ac.at/˜truong/publications/auroratr2003-22.ps.gz. 17

[24] F. Vraalsen, R. Aydt, C. Mendes, and D. Reed. Performance contracts: Predicting and monitoring grid application behavior. In *Proceedings of the 2nd International Workshop on Grid Computing/LNCS (GRID 2001), Springer-Verlag Lecture Notes in Computer Science, Denver, Colorado, November 12, 2001, Volume 2242, pp. 154-165*, 2001.
www-pablo.cs.uiuc.edu/Publications/Papers/Performance_Contracts_Predicting_and_Monitoring.pdf. 11

[25] R. Wolski, N. Spring, and J. Hayes. The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. *Journal of Future Generation Computing Systems*, 15(5-6):757–768, October 1999. 16

G Authors

Figure 1: The authors at the APART WP3 workshop in Rackeve, Hungary.