DELIVERABLE D2.1

PART III: SOFTWARE REQUIREMENTS SPECIFICATION FOR GRIDBENCH (THE CROSSGRID BENCHMARK SUITE)

WP2.3 Metrics and Benchmarks

Document Filename:	**CG2.3-D2.1-v1.1-UCY001-SRS.doc**
Work package:	**WP2.3 Metrics and Benchmarks**
Partner(s):	**UCY, TUM**
Lead Partner:	**UCY**
Config ID:	**CG2.3-D2.1-v1.1-UCY001-SRS**
Document classification:	**PUBLIC**

Abstract: This document describes the basic requirements for the CrossGrid Benchmark Suite, which will be used to investigate experimentally the performance of Grid sites and constellations. The CrossGrid Benchmark Suite will provide a variety of tools, such as *Micro-benchmarks, Micro-kernel benchmarks* and *Application-kernel benchmarks*. These tools can be used to measure the performance of the different elements that make up the Grid. Benchmarking these elements (such as networking and computational power) will provide essential information that is useful for administration of the Grid, real application performance prediction, cost estimation and numerous other uses.

Delivery Slip

	Name	**Partner**	**Date**	**Signature**
From	WP2, task 3	UCY. TUM	30/04/2002	
Verified by	N. Meyer, P. Wolniewicz, R. M. de Llano	PSNC, U. Cantabria	21/05/2002	
Approved by				

Document Log

Version	**Date**	**Summary of changes**	**Author**
Draft	30/4/2002	Draft version	M. Dikaiakos G. Tsouloupas
0.9	15/5/2002	Pre-Final	M. Dikaiakos G. Tsouloupas
1.0	1/6/2002	Final	M. Dikaiakos G. Tsouloupas
1-0-B	2/6/2002	Final layout	H.Marten
1.1	12/8/2002	1st Update	M. Dikaiakos

CONTENTS

1. INTRODUCTION

1.1. PURPOSE

This document seeks to provide the Software Requirements Specifications for the CrossGrid benchmark suite, to be developed under task WP2.3 of the CrossGrid project. Software Requirements Specifications will serve as a guideline for future WP2.3 work. Furthermore, this document presents an initial description of the various functionalities and services that WP2.3 software expects to invoke and/or incorporate from other CrossGrid tasks (e.g., WP2.4, WP3).

The intended audience is comprised of members of the CrossGrid consortium, and the Grid community at large that is interested in Grid performance evaluation. WP2.3 seeks to:

- Propose models and performance metrics for the Grid, to capture and describe concisely:
  - o The performance of *Grid constellations,* i.e. of collections of computing resources (clusters, parallel computers) connected through wide-area networks, belonging to a Virtual Organization and used concurrently for the solution of a particular problem.
  - o The performance of Grid applications deployed upon a Grid constellation.
- Provide a specification of a suite of benchmarks (GridBench - the CrossGrid Benchmark Suite) capturing such metrics.
- Provide a reference implementation of GridBench on the CrossGrid test-bed and run benchmarking experiments.
- Provide members of the CrossGrid consortium with access to the benchmark suite and to historical data collected from benchmark invocations, through the CrossGrid portal interface.
- Assess the usefulness of benchmarks in:
  - o Identifying factors that affect Grid application performance.
  - o Providing a mechanism for an early, low-cost performance evaluation of Grid constellations, and the cost prediction of Grid applications.
- Providing parameter estimates to the analytical performance prediction models, to be developed in CrossGrid.

1.2. SCOPE

The products of Task 2.3 are:

1. Definition of performance metrics that capture and describe in a concise manner essential performance properties of Grid elements and Grid configurations. Metrics will be defined upon and interpreted on top of a simple performance model for the Grid.

2. GridBench, a suite of benchmarks, designed to produce those metrics and describe performance properties of Grid elements. GridBench will be comprised of three different layers of benchmarks: (i) Micro-benchmarks; (ii) Micro-kernels, and (iii) Application kernels.

3. Reference implementations of GridBench on the CrossGrid test-bed.

4. Experimentation with and assessment of GridBench on the CrossGrid test-bed.

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

CCA	Common Component Architecture
CE	Computing Element
APART	Automatic Performance Analysis: Real Tools
API	Application Program Interface
ASL	APART Specification Language
CG	CrossGrid
CrossGrid	The EU CrossGrid Project IST-2001-32243
DataGrid	The EU DataGrid Project IST-2000-25182
G_PM	Grid-enabled Performance Measurement tool
Grid Constellation	A collection of "sites" (see "*Site*"), connected within a Virtual Organization and used for the solution of a common problem
GridBench	The CrossGrid benchmark suite
GridFTP	A Grid-enabled version of FTP (Part of Globus)
GRIS	Grid Resource Information Service
HLAC	High-level analysis component
HPC	High Performance Computing
HTC	High Throughput Computing
MPI	Message Passing Interface
MPICH-G2	A free, Globus-enabled MPI implementation
NAS	NASA Advanced Supercomputing
PVFS	Parallel Virtual File System
SE	Storage Element
Site	A set of Computing Elements, Storage Elements or other resources in a single geographical location.
SPLASH	Stanford Parallel Applications for Shared Memory
VO	Virtual Organization

1.4 REFERENCES

[1] Bailey, D. H., E. Barszcz, J. Barton, D. Browning, R. Carter, L. Dagum, R. Fatoohi, S. Fineberg, P. Frederickson, T. Lasinski, R. Schreiber, H. Simon, V. Venkatakrishnan, and S. Weeratunga, "The NAS Parallel Benchmarks," Technical Report RNR-94-007, NASA Ames Research Center, (March 1994).

[2] Dikaiakos, M. "Analyzing the Workload of Scientific Visualization Tools: A Preliminary Study with TIPSY." In *Proceedings of the 1997 International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems - MASCOTS '97*, IEEE-Computer Society.

[3] Dikaiakos, M., Samaras, G., "Performance Evaluation of Mobile Agents: Issues and Approaches." In *Performance Engineering. State of the Art and Current Trends,* Dumke, Rautenstrauch, Schmietendorf and Scholz (eds). Lecture Notes of Computer Science series, State of the art survey, vol. 2047, pages 148-166, Springer, May 2001.

[4] Dikaiakos, M., Rogers, A. and Steiglitz, K., "Performance Modeling through Functional Algorithm Simulation," in *Advanced Computer System Design*, Chapter 3, G. Zobrist, K. Bagchi and K. Trivedi (editors), pages 43-62, Gordon & Breach Science Publishers, 1998, ISBN 90-5699-634-7.

[5] Dongarra, J., Luszczek, Petitet, A. The LINPACK Benchmark: Past, Present, and Future, 2001.

[6] Foster, I. and C. Kesselman, C. (editors), "*The Grid: Blueprint for a New Computing Infrastructure.*" Morgan Kaufmann, 1998.

[7] Saphir, W., van der Wjingaart, R., Woo, A., Yarrow, M., Bailey, D., Harris, T., "The NAS Parallel Benchmarks 2.0." *The International Journal of Supercomputer Applications*, 1995.

[8] Woo, S., Ohara, M., Torrie, E., Singh, J.P., and Gupta, A. "The SPLASH2 Programs: Characterization and Methodological Considerations." In *Proceedings of the Twenty-first International Symposium on Computer Architecture*, June 1995.

[9] Grid Benchmarking Research Group, http://www.nas.nasa.gov/GGF/Benchmarks

[10] Funika, W., Rivera, F., Wismuller, R., "WP2.4: Interactive and semiautomatic performance evaluation tools." In *Deliverable D2.1, Part IV: Software Requirements Specification for Grid-enabled Performance Measurement and Performance Prediction*, CG-2.4-DOC-CYFRONET001-SRS. CrossGrid Project, March 2002.

[11] APART Specification Language (ASL), http://www.par.univie.ac.at/~tf/apart/wp2/report-wp2.ps.gz

[12] Palak, B. et al., "Task 3.1 SRS." CG-3.1-SRS-0017, CrossGrid Project, March 2002.

[13] van der Wjingaart, R., "Establishing a Global-Grid Forum Research Group on Benchmarking."I *Presentation at GGF5,* Edinburgh, UK, July 2002.

[14] Bramley, R., Ghiu, K., Diwan, S., Gannon, D., et al. "A Component Based Services Architecture for Building Distributed Applications." In *Proceedings of the 9th IEEE International Symposium on High-Performance Distributed Computing*, pages 51-59, 2000.

1.5 OVERVIEW

This document provides the software requirements for GridBench, the CrossGrid benchmarking suite. Section 2 provides an overall description of the goals and the context of GridBench development, giving the product perspective and GridBench interfaces functions, user characteristics

2. OVERALL DESCRIPTION

This task will propose a set of performance metrics to describe the performance capacity of Grid constellations and Grid application performance, and it will define, develop, and implement benchmarks that are representative of Grid workloads. Such benchmarks can be used to estimate the values of performance metrics for different Grid constellations, to identify important factors that affect end-to-end application performance, and to provide application developers with the initial estimates of expected application performance and cost-performance ratios. GridBench will be deployed and validated in the CrossGrid Testbed.

Benchmark definition will follow the activities of, and contribute to the *Research Group on Benchmarking* of the *Global Grid Forum* [9], which was established during GGF5 Conference in Edinburgh on July 2002.

2.1 PRODUCT PERSPECTIVE

The "product" of WP2.4 is a reference implementation of GridBench, the CrossGrid Benchmark Suite. This implementation will entail a number of representative benchmarks, implemented upon the CrossGrid testbed and representing a small part of the overall contribution of WP2.4, which includes the high-level performance model for the Grid, the set of performance metrics representing Grid and Grid-application performance, the layered benchmarking framework, the definition of Grid benchmarks at different layers, experimentation with GridBench and the resulting Performance Engineering research.

As a product, GridBench will provide members of the CrossGrid community with an interface to the reference implementation of Grid-oriented benchmarks, enabling them to conduct benchmarking experiments to test the performance of various levels of the Grid architecture.

The definition of elements of the GridBench suite will be based upon established benchmarks of the Parallel Computing community (such as NAS, Linpack, Parbench), which will be evaluated and adapted to the needs of the Grid. Furthermore, the GridBench effort will follow and contribute actively to the activities of the Research Group on Benchmarking of the Global Grid Forum (established during GGF5 at Edinburgh, UK, on July 2002). Selected benchmarks of GridBench will be representative of CrossGrid applications and middleware. GridBench will seek to capture the performance characteristics of three different domains of the Grid:

a. At the site-level, that is a set of computing resources connected via a local network, under a common administrative domain (e.g. a cluster);

b. At the Grid-constellation level, which involves multiple "sites" connected within a Virtual Organization and used for the solution of a common problem (e.g. a large FFT computation, a flood simulation, etc.), and,

c. At the Grid middleware level, i.e. libraries and services supported by CrossGrid, DataGrid and Globus, and used at the application-programming level to develop Grid-enabled applications.

2.1.1 System interfaces

The deployment of GridBench takes as granted the installation of the following system-software packages:

- Linux OS, kernel 2.2 or higher
- X-Windows X11R6 (XFRee86)
- A Web Server and a Java Servlet engine such as the one integrated with the Apache Web Server.

- A full installation of JAVA (SDK2.3 or higher).

Moreover, the GridBench reference implementation will use the interfaces to the CrossGrid Middleware as it evolves.

GridBench will export its results in files to be stored in Storage Elements accessible through the User Portal. Alternative approaches for storing and/or publishing benchmarking data through CrossGrid Information Services (e.g. GRIS) will be explored in the future.

2.1.2 User interfaces

Graphical interface between user and the GridBench will be provided in the context of the GRID desktop and implemented as a module of the Application Portal Server. A user will employ the various input panels of the Application Portal Server to login, select input files and benchmarks, and launch a benchmarking experiment.

Aggregate benchmarking results will be presented through the Benchmark and Metrics Panel. Additionally, the user will be able to monitor measured performance properties through the user interface of the G-PM module of WP2.4. The UI will provide two options for reviewing performance measurements:

1. Through a hypertextual interface, implemented as an HTML page and enabling the user to click on the descriptions of machines/sites upon which a benchmark was executed in order to view the performance metrics derived from his benchmarking experiment.

2. Through a graphical representation of the high-level, abstract model of the Grid configuration used. The user will be able to click upon the icons of the machines/sites upon which the benchmark was executed in order to retrieve the performance metrics derived from his benchmarking experiment. Figure 1 presents a high-level view of the graphical user-interface.

2.1.3 Hardware interfaces

N/A

2.1.4 Software interfaces

Furthermore, as a number of GridBench benchmarks will follow the application-programming model adopted by CrossGrid, the following software packages will be required for its implementation:

- Coding in FORTRAN/C/C++;

- MPICH-G2 for parallelization and inter-process communication;

- The G-PM Tools for measurements definitions and retrieval: As the purpose of benchmarking is to measure carefully selected performance properties, it is very important to have the proper interfaces with the G-PM tool to be developed under WP2.4, in order to gather proper and accurate measurements of GridBench metrics.

- The User Portal for defining and interacting with GridBench experiments.

- Access Grid services through the APIs of CrossGrid middleware, which will also include Globus and DataGrid middleware services.

Additional interfaces to the Common Component Architecture Framework (CCA) could be employed in some benchmarks, if this framework is eventually adopted in the application programming effort of WP1.

GridBench will export its results in XML-encoded files to be stored in Storage Elements accessible through the User Portal. Alternative approaches for storing and/or publishing benchmarking data

through CrossGrid Information Services (e.g. GRIS) will be explored in the future.

2.1.5 Communications interfaces

Benchmarks will utilize MPICH-G2 for communication. Alternative communication mechanisms and protocols will be explored, if deemed necessary from the study of application-programming models adopted by WP1.

2.1.6 Memory constraints

Memory constraints of benchmarks will be configurable by benchmark parameters determining the problem size of a particular benchmark invocation. The goal is to come up with benchmark memory requirements representative of typical WP1-application runs.

2.1.7 Operations

CrossGrid administrators and users will be able to launch a benchmark on a target configuration, a Grid constellation or a particular Site, through the Portal developed in Task 3.1.

During the project, we will investigate the development of a *Grid Benchmarking Service*, so that another service (such as scheduler or the Resource Manager) may find it useful to run the benchmarks on demand or at regular intervals.

2.1.8 Site adaptation requirements

N/A

2.2 PRODUCT FUNCTIONS

GridBench seeks to investigate performance properties of three "layers" of the Grid architecture:

(i) The *Site*, which is a collection of resources interconnected through a local- or system-area network, belonging to one administrative domain (e.g. a cluster of PCs, a symmetric multiprocessor system);

(ii) The *Grid Constellation*, which includes multiple sites constituting the computing platform of a Virtual Organization, and used for the solution of a problem;

(iii) The *Middleware* that is the software layer providing access to shared resources of a Grid constellation, through an API available at the application programming level.

GridBench will be composed of three families of benchmarks: *Micro-benchmarks, Micro-kernels* and *application-kernels*.

		Benchmark Targets		
		Site*	Grid Constellation	Middleware
Benchmark Categories	Micro-Benchmarks	(1)		(2)
	Micro-Kernels	(3)	(4)	
	Application Kernels	(5)	(6)	

Table 1: the GridBench matrix

* Benchmarks that target specific Sites will also indirectly measure performance of the Computing and Storage Elements at those sites.

Micro-benchmarks will target Sites (1) and Middleware (2) (e.g., GridFTP transfers). Micro-kernels will target Sites (3) and Grid constellations (4). Application kernels will target Sites (5) and Grid constellations (6). Even though (4) and (6) do not target the middleware explicitly, the middleware will affect their performance.

2.2.1 Benchmark Categories

Micro-benchmarks

Micro-benchmarks are small and simple programs written specially to isolate performance characteristics regarding processing, local memory, input/output, communication, synchronization, contention, etc. GridBench micro-benchmarks seek to identify basic performance properties of Grid services, sites, and constellations. Each micro-benchmark will test a single performance aspect, through "stress testing" of a simple operation invoked in isolation. Metrics captured represent computing power (flops), memory capacity and throughput, I/O performance, network, etc.

Micro-kernels

- "Stress-test" several performance aspects of a system at once.
- Generic HPC/HTC kernels, including general and often-used kernels in Grid/Cluster environments.
- Fit for cost estimation.

Application kernels

- Characteristic of representative CG applications (extracted from CrossGrid applications).
- Capturing higher-level metrics, e.g. completion time, throughput, speedup.

2.2.2 Benchmark Targets

Each benchmark category targets a particular "layer" of Grid architecture. In particular, we anticipate benchmarks that:

- Isolate performance properties of Sites participating in a Grid constellation.

- Capture the performance behavior of a particular Grid constellation at a particular point in time.

- Isolate the performance capacity of Grid middleware services, which are available at the application-programming level.

Benchmarks that target a Site will report on metrics that are applicable to sites (clusters) such as flops, I/O, and network performance. Benchmarks targeting a Grid constellation will report on similar metrics. Benchmarking the middleware aims at (a) investigating the effects of middleware on end-to-end application performance, and (b) characterizing middleware performance.

2.2.3 General use case

GridBench is intended for use by a wide variety of users, ranging from application programmers to site administrators. Before invoking a particular benchmark, a user will have to make the necessary decisions based on what attributes he wishes to measure. In particular, the user will specify:

1. Which benchmark to run?
2. Benchmark-specific parameters.
3. Location (where to run it).

Parameter specification and benchmark invocation will be performed through the User Interface of the CrossGrid Portal (Task 3.1), which will employ the job submission mechanisms provided by the CrossGrid platform.

2.2.4 A Usage Scenario

The following scenario of use is provided to describe a typical use-case of the GridBench:

George is a CrossGrid user who wants to run n-body simulations.

1. George is developing an N-body code for a molecular dynamics application. He wants to have an estimate on where he can conduct his simulations. To this end, he selects a benchmark with an N-body micro-kernel and spawns it on his local Grid site. He runs a few simulations with different input parameters, studying the scalability of measured performance, for increasing problem sizes (up to 500,000 particles and 50 time-steps) and processor numbers. After a couple of hours of benchmarking, performance monitoring, and metrics' study, George has shaped an early idea of the rough performance he can receive locally.

2. George deploys his code on the local site and runs a first simulation on a realistic problem size (20 million particles, 1000 time-steps) expecting to receive results after 24 hours.

3. While waiting for his first simulation to finish, George decides to run a larger simulation that could benefit from the aggregated computing power of the CrossGrid Virtual Organization, deploying his computation in four remote sites: two sites in Spain, one in Greece, and one in Poland.

4. To assess the potential performance improvement and the incurring cost, George configures the N-body benchmark to run on 5,000,000 particles for 20 time-steps. He submits this benchmark through the CrossGrid Portal to the four sites. The Portal reports back to him that the benchmarking cycles of the first Spanish site are reserved for the next week by someone else and provides him with a pre-reservation for the other three sites.

5. George dispatches the benchmark to the three sites, with more modest parameters (2,000,000 particles for 10 time-steps). While the benchmark runs, he monitors its performance using

CrossGrid tools and observers a much higher than expected communication to computation ratio, which leads to a very slow overall execution.

6. George aborts the benchmark and invokes a few micro-benchmarks to conclude that the network link connecting the Polish to the Spanish site is congested.

7. George re-spawns the N-body benchmark, excluding the Polish site. Following four hours of execution, the CrossGrid Portal provides George with the outcome of his benchmark: statistics about computing time, memory consumption, network bandwidth used, and an estimate of the cost to be charged for a simulation of similar size on the same nodes. George compares these results with historical data from prior N-body benchmark executions on the same and other nodes of the CrossGrid Virtual Organization. The overall conclusions are positive.

8. In the mean time, the simulation running on the local site has finished in 20 hours. Now, George has a better understanding of what he should expect from his own code and from the resources available through CrossGrid, in terms of performance and completion time. Therefore, he goes on to reserve the Spanish, Greek and Polish sites for a number of large runs of his code.

2.3 USER CHARACTERISTICS

GridBench will be geared towards the following main groups of users:

1. **End Users:** This is the group of users specified by all the applications of WP1. These users will run mostly micro-kernels and application-kernel benchmarks.

2. **Administrators:** Site administrators or VO integrators (i.e. people who put together Virtual Organizations) that need to evaluate periodically the performance characteristics of their Grid configurations and publicize the performance capacity of their nodes; they may need to run any type of benchmark.

2.4 CONSTRAINTS

Application-kernel benchmarks will have constraints similar to those of applications in WP1. Invocation of benchmarks will comply with the security policies of CrossGrid.

2.5 ASSUMPTIONS AND DEPENDENCIES

The design, implementation and experimentation with GridBench depend on a number of choices and developments inside and outside the CrossGrid project. In particular:

1. GridBench seeks to follow and contribute to the Research Group (RG) on Benchmarking of the Global Grid Forum [9]. Early work of this group has provided descriptions of a couple of "computationally intensive" benchmarks, which are initial adaptations to the Grid of known NAS kernels [13]. Moreover, at this early stage, the RG has reasonably taken a bottom-up approach leaving for the future the investigation of comprehensive frameworks for Grid benchmarking [13]. The GGF RG, however, is expected to drive the standardization efforts in the area of Grid benchmarking, and to establish standards for measuring Grid performance. In conclusion, it will be considered a significant success for CrossGrid and task WP2.3, to contribute significantly to the works of the respective GGF RG. To this end, we need:

   a. To take into account the benchmark designs of the GGF-RG, and incorporate the relevant proposals in GridBench and in the reference implementation of GridBench for the CrossGrid platform.

   b. To provide towards the RG on Benchmarking ideas regarding a comprehensive benchmarking framework (along the lines of the layered approach presented earlier) and application-oriented benchmarks inspired by the CrossGrid application suite of WP1.

2. Application-kernel benchmarks are inherently dependent on the choices of programming model, structure of applications (with the use of, for instance, frameworks like CCA or XCAT), and middleware services employed by the WP1 applications. Therefore, many of the

assumptions and dependencies of the WP1 applications will be "inherited" by the respective benchmarks. As these are still in limbo for some of the WP1 applications, assumptions and dependencies of the application-kernel benchmarks remain undefined.

3. Micro-kernel benchmarks will assume a working installation of Globus and MPICH-G2 at each test-bed site (as dictated by WP4). Platform uniformity, to the extent dictated by WP4, is another important dependency.

4. WP2.3 will try to take into account applications of the European DataGrid project, in order to ensure the representation of the widest possible range of applications in the GridBench suite.

2.5.1 Dependence on other Tasks:

- Experimentation with GridBench depends to a large extent on Task 2.4 for the development of the Performance Measurement mechanisms. Instrumentation of benchmarking codes with G-PM "probes," invocation of G-PM and HLAC functions, and HLAC- and ASL-compliant specifications represent building blocks, which are essential for GridBench experimentations and measurements.

- GridBench is dependent on Task 3.1 for the provision of the final User Interface.

- There is also dependence on Task 3.3 for archival of benchmark data (in the form of monitoring) if it is decided that archival is necessary.

2.6 APPORTIONING OF REQUIREMENTS

By the end of Month 6:
- Updated version of requirements specification given feedback from WP1, WP2 and WP3.
- Early design of GridBench.
- Adoption of methodology for benchmark analysis and design.

By the end of Month 8:
- Definition of a high-level performance model representative of the CrossGrid testbed.
- Definition of the set of performance metrics to describe the performance capacity of CrossGrid configurations and application performance.
- Definition of micro-benchmarks (in paper-and-pencil).

By the end of Month 10:
- Definition of representative micro-kernels.
- Definition of a benchmark representative of a selected WP1 application (probably HEP).
- Definition of first-round deployment and experimentation scenarios.

By the end of Month 12:
- First prototypes of micro-benchmarks and representative micro-kernels.
- Integration with first prototypes of G-PM.

Apportioning of requirements beyond month 12 will be provided after an evaluation of the progress achieved in the first year of CrossGrid inside WP2.3, in other tasks upon which WP2.3 depends, and in external activities such as the Global Grid Forum and the DataGrid project. New apportioning of requirements will be published in a future version of the SRS document.

3. SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACES

The primary goal of most benchmarking efforts described in the literature (e.g., [7,13]) is to come up with definitions of benchmarks, which consist of paper-and-pencil descriptions and unambiguous directions on how to execute a particular benchmark so that its results (measured metrics) can be directly comparable to results from that benchmark's invocations on different computing platforms. Actual benchmarking is conducted with reference implementations in selected platforms, which are duly instrumented, launched, and measured to provide the performance metrics sought. This part of the benchmarking effort in GridBench will be provided by the external interfaces of the reference GridBench implementation of CrossGrid:

- Functions of the G-PM tool of WP2.4 will be used to define performance measurements, and to derive and extract performance data.

- Functions of the User Portal of WP3.1 will provide users with the necessary functionality to launch benchmarks on the CrossGrid testbed and review the results through a high-level representation.

Figure 1 gives an overview of task 2.3 interfaces. A user will be able to invoke benchmarks through the portal (WP3.1). Performance measurements of metrics will be specified according to and carried out by the Grid-enabled Performance Measurement Tool (G-PM) of WP2.4. Benchmarking results will be conveyed back to the user through the G-PM tool and will be summarized in a higher-level description through the User Interface of the Portal. Alternative approaches for storing and/or publishing benchmarking data (e.g. through GRIS) will be explored in the future.

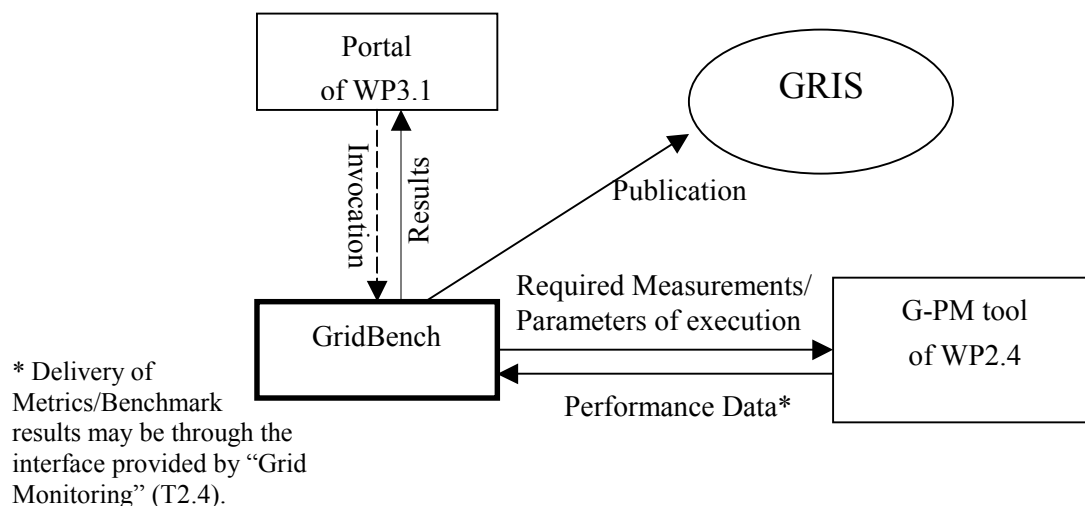More details on the interfaces with G-PM and the Portal are given below.

Figure 1:Diagramatic view of general functionality and interaction with other components

G-PM tool of WP 2.4

Task 2.4 plans to develop G-PM, a tool for providing performance measurement information. In particular, G-PM: (i) collects measurements of various aspects of program execution on the Grid, through its Performance Measurement Component (PMC); (ii) extracts high-level performance properties of an application through its High-Level Analysis Component (HLAC); (iii) predicts application performance based on an analytical modelling and prediction module, the Performance Prediction Component (PPC); and (iv) enables the visualization of performance data through its User Interface and Visualization Component (UIVC).

Of particular importance for the benchmarking measurements is the employment of the HLAC component of G-PM. HLAC "allows to measure application specific performance properties" by "inserting probes at strategic places of the application code"[10]. HLAC employs the APART Specification Language ASL [11] to specify, "how application specific properties are computed from the data delivered by these probes and the generic performance measurement data" [10].

Therefore, and according to what is expected from applications by the G-PM [10], the ASL specification of all GridBench specific performance properties will be stored in configuration files compliant to HLAC specifications. These files will drive the measurements and reports of GridBench metrics via the G-PM tools. Notably, the configuration files provided by GridBench will provide an automated alternative to the manual configuration of measurements through the "*measurement definition window*" of User Interface of the G-PM.

Besides *application-related requests*, the G-PM tools enable the execution of *infrastructure-related requests* that specify "information to be returned on the whole or part of the Grid infrastructure," such as the availability of resources, static and dynamic resource information, etc [10]. Infrastructure-related measurements can focus on sites and hosts and can be defined on different time granularities. This capability will be used by the GridBench components in order to establish the properties of the high-level GridBench performance model.

Output of the G-PM measurements will be provided to the user via the X-Window interface of G-PM, WP2.4. Furthermore, measurements will be extracted and stored in an XML file available for presentation through the CrossGrid Portal. The performance properties derived from benchmarking can be used by WP2.4 for specifying the values of parameters employed in the performance prediction model. Alternative options for publishing benchmark data will be considered, such as the use of facilities supplied by Globus (through GIS).

User Portal of WP3.1

Benchmarks will be invoked via the portal created by Task 3.1 [12]. Therefore, the user of CrossGrid benchmarks should be able to use the portal to log-in and define a benchmarking job in terms of the particular benchmark that he wants to execute among the GridBench suite, its configuration (input files, configuration parameters), and the resources he wants to reserve for his job. Site administrators should also be able to define and run benchmarks, collect and store measurements in databases for the perusal of Grid users. In contrast to real applications, which are executed to provide some useful output data, benchmarks are executed simply for collecting performance measurements. Besides the output through the X-Window based tools of G-PM, GridBench measurements will be summarized through the CrossGrid portal in hypertextual (HTML) and graphical format.

In summary, the User Portal to be produced by WP3.1 [12] will provide users with a number of **services**: (i) giving access to a database of prior benchmarking results; (ii) defining benchmarking jobs, and (iii) requesting the job output in graphical and/or alphanumerical format. Furthermore, through the portal, a user will have access to the various configuration and input files required for benchmark invocation and performance measurements. These files will be saved in storage elements and described in metadata catalogues encoded in XML format.

The following is a **list of actions** that a GridBench user can perform in a general scenario, through the User Portal of WP3.1:

1. A user starts with the login operation authenticating him using his certificate to be accepted as a member of the CrossGrid VO.

2. Once accepted, the user sees a list of available benchmarks and associated configuration, input files and parameters on the client.

3. Within this list, the user selects the relevant benchmark to be executed in the computational grid.

4. The user defines the input data for the selected benchmark. The user submits the selected GridBench job with the defined input data.

5. If the user feels the benchmark takes too long to execute, he can send a request to stop it (like CTRL-C command).

6. At the end of the session, the user can save the whole session trace in XML format either in their local storage or in the benchmark repository within the system for the perusal of other users.

3.2 FUNCTIONS

As stated earlier, GridBench will provide three layers of benchmarks. Each layer targets a different "aspect" of performance properties that determine Grid-application performance.

3.2.1 Micro-benchmarks

Micro-benchmarks are small, specially written programs seeking to isolate simple performance characteristics. In other words, micro-benchmarks measure performance properties of programming primitives used at the application level for inter-process communication, I/O, invocation of basic Grid services, etc. These are applicable to API's that are provided by the middleware and MPICH-G2, and that have a strong and measurable impact on the overall Grid or Site performance. Middleware micro-benchmarks will seek to isolate the performance properties and extract the performance capacity of middleware services that are invoked by typical CrossGrid applications through the CrossGrid API's.

Some of these benchmarks will address the performance of the CE's (Compute Elements) and SE's (Storage Elements) directly. The properties of CE's and SE's will be measured either individually or collectively at a site.

Especially in the case of SE's, the performance of each *autonomous* SE (i.e. a machine or a set of machines that is independent of other SE's) will be measured. "Autonomous" is stressed because it is not our purpose to study performance at the single machine level. An autonomous SE could be a cluster of machines running (for example) PVFS, in which case the performance of the whole set of machines will be measured collectively, not each node of the specific PVFS system.

3.2.2 Micro-kernels

Micro-kernels will be based on existing kernels (i.e. pieces of code that form the core of a computation) with well-known codes and well-known behavior that appear often in parallel processing, such as NAS, SPLASH, etc.

- For "stress testing" and identification of basic performance of sites and Grid constellations. They characterize a set of grid nodes constituting a Grid constellation.

- Codes with HPC and/or HTC requirements.

- Measure Computing power (flops) at the Grid level, Storage/Network capacity, cost estimate, scalability, etc.

- To investigate the effects of middleware on end-to-end application performance.

The user will be able to choose to run all or a subset of the developed benchmarks at the Grid constellation or Site level.

3.2.3 Application kernels

- Characteristic of representative CG applications.

- Higher-level metrics (completion time, throughput, speedup…).

The applications from WP1 will be analysed and, in cooperation with WP1, the critical kernels will be identified. These kernels will then be utilized in benchmark development.

Application kernel benchmarks at the **Grid level** will be a measure of performance of a Grid constellation. A *Grid constellation* is the set of Grid nodes belonging to the same Virtual Organization and used collectively in the solution of a particular problem.

Benchmarking at the Grid constellation level is necessary, as the performance of a Grid constellation cannot be derived directly from the aggregate performance of participating Sites: when using a Grid constellation to run a particular application, different components of the application may run at separate sites, and communicate or transfer files over Internet. Therefore, the benchmark should mimic the collective use of the constellation's resources in the execution of that particular application.

3.3 PERFORMANCE REQUIREMENTS

There will be upper bounds on completion time for benchmarks, which are to be determined and imposed.

3.4 LOGICAL DATABASE REQUIREMENTS

GridBench will use different database engines only indirectly through other components (Such as Task 2.4 components or the Globus GRIS)

3.5 DESIGN CONSTRAINTS

Benchmarks must be compatible with the basic building blocks of CrossGrid. Therefore benchmark designs will be based on Globus, DataGrid software and other middleware, as dictated by WP1 and WP4.

3.6 STANDARDS COMPLIANCE

Benchmarking code will comply with MPICH-G2 where applicable.

3.7 SOFTWARE SYSTEM ATTRIBUTES

N/A

4. INDEX

Application kernel, 9, 10, 14

Benchmark invocation, 7, 8, 10, 13

Constellations. *See* Grid Constellations

Constraints, 8, 11, 14

DataGrid 4, 5, 8, 14

Dependencies, 3, 12

Flops, 9, 10, 13

Globus 4, 5, 7, 8, 11, 14

Grid constellations, 4, 5, 6, 8, 9, 13

Interfaces, 7, 8, 13

Metrics, 4, 5, 6, 7, 10, 14

Micro-benchmarks, 9, 13

Micro-kernel, 12

Middleware, 4, 8, 9, 10, 11, 13, 14

MPICH-G2, 6, 8, 12, 13, 14

Usage Scenario, 10

Use case, 10

Users, 11

Virtual Organization, 4, 6, 9, 11, 14

VO. *See* Virtual Organization