An Evaluation Methodology for Computational Grids*

Eduardo Huedo[1], Rubén S. Montero[2], and Ignacio M. Llorente[1,2]

[1] Laboratorio de Computación Avanzada, Simulación y Aplicaciones Telemáticas, Centro de Astrobiología (CSIC-INTA), 28850 Torrejón de Ardoz, Spain
[2] Departamento de Arquitectura de Computadores y Automática, Universidad Complutense, 28040 Madrid, Spain

Abstract. The efficient usage of current emerging Grid infrastructures can only be attained by defining a standard methodology for its evaluation. This methodology should include an appropriate set of criteria and metrics, and a suitable family of Grid benchmarks, reflecting representative workloads, to evaluate such criteria and metrics. The establishment of this methodology would be useful to validate the middleware, to adjust its components and to estimate the achieved quality of service.

1 Introduction

Benchmarking is a widely accepted method to evaluate the performance of computer architectures. Traditionally, benchmarking of computing platforms has been successfully performed through low level probes that measure the performance of specific aspects of the system when performing basic operations, e.g. LAPACK [1], as well as representative applications of the typical workload, e.g. SPEC [2] or NPB [3]. In this sense, benchmarking has been proved helpful for investigating the performance properties of a given system, either for prediction or comparison purposes.

Grid benchmarks can be also grouped in the two aforementioned categories: low level probes that provide information of specific aspects of system's performance; and benchmarks that are representative of a class of applications. In this first category, the Network Weather Service [4] provides accurate forecast of dynamically changing performance characteristics from a distributed set of computing resources. Also, a set of benchmark probes for Grid assessment have been proposed [5]. These probes exercise basic Grid operations with the goal of measuring the performance and the performance variability of basic Grid operations, as well as the failure rates of these operations. Finally, the GridBench tool [6] is a benchmark suite for characterizing individual Grid nodes and collections of Grid resources. GridBench includes micro-benchmarks and application kernels to measure computational power, inter-process communication bandwidth,

* This research was supported by Ministerio de Ciencia y Tecnología, through the research grants TIC 2003-01321 and 2002-12422-E, and by Instituto Nacional de Técnica Aeroespacial "Esteban Terradas" - Centro de Astrobiología.

L.T. Yang et al. (Eds.): HPCC 2005, LNCS 3726, pp. 499–504, 2005.
© Springer-Verlag Berlin Heidelberg 2005

and I/O performance. In the second category, the *Grid Benchmarking Research Group*, within the *Global Grid Forum*, proposes to create a set of representative Grid benchmarks [7], which will embody challenging usage scenarios with special emphasis on large data usage. The *NAS Grid Benchmarks* (NGB) [8] was the first Grid benchmark specification available.

The aim of this paper is, firstly, to propose a set of criteria and metrics which allow evaluating the capabilities of a computational Grid environment from a user's point of view; and, secondly, to apply these criteria and metrics in the evaluation of a Grid environment based on Globus basic services using Grid*W*ay [9] as metascheduler and NGB as test programs. As an initial phase of this work, the paper-and-pencil specification of this benchmark suite for computational grids has been implemented by using the *Distributed Resource Management Application API* (DRMAA) supported by Grid*W*ay [10].

In Section 2, we describe the criteria and evaluation metrics used in this work. Then, the main characteristics of the NGB suite are detailed in Section 3. In Section 4, we describe the evaluation process of a Grid infrastructure. Finally, Section 5 presents the main conclusions of our work.

2 Criteria for Grid Evaluation

We propose functionality, reliability and performance as general criteria to evaluate a Grid environment from a user's point of view, and to guarantee the extension of its use. We have tried to keep the evaluation criteria simple and objective. In this sense, each metric is easy to measure and provides a boolean or numeric value, which is therefore also easy to compare.

In spite of the great research effort made over the last decade, application development and execution on grids require a high level of expertise due to its complex nature. Therefore, functionality should be considered as a valuable criterion, and a Grid evaluation methodology should reflect the ability of the environment to execute unattended distributed *communicating* applications. The NGB suite falls in this category, and the capability to execute it constitutes a suitable metric to test the functionality of the environment. Moreover, benchmarks should be expressed by using standard high level interfaces, like DRMAA.

Grid environments are difficult to efficiently harness due to their heterogeneous nature and unpredictable changing conditions. Adaptive scheduling and execution are some of the techniques proposed in the literature [11,12] to achieve a reasonable degree of application performance and fault tolerance. Therefore, a suitable methodology for Grid evaluation should also help to determine the reliability and dynamic adaptation capabilities of the Grid environment. As simple metrics, we propose that a job could, transparently to the user, continue its execution (at least from the beginning) in other resource when some of the following failure or loss of quality of service conditions take place [13]:

- Job cancellation (failure) or suspension (QoS loss)
- System crash (failure) or saturation (QoS loss)
- Network disconnection (failure) or saturation (QoS loss)

The coordinated performance of all the involved resources and services should be considered when analyzing the performance of a Grid infrastructure. An evaluation methodology for grids should provide tools and metrics to measure and adjust its performance. As user-level performance metrics, we propose:

– Turnaround time (T): It is the waiting time from the job execution request until the results are available.
– Productivity (P): As usual, productivity is defined as the number of completed tasks or benchmark instances per unit of time.

Moreover, it is very important to quantify the overheads of the involved components and to analyze their influence in the global performance. Therefore, we should also consider other more appropriate metrics for diagnostic and tuning purposes, interesting for application and middleware developers, and Grid architects:

– Response time (T_r): It is the time between submitting a job and the starting of the stage-in phase on the execution host. It provides information about the overhead induced by the scheduler and the Grid middleware.
– Transfer and execution time (T_{xfr} and T_{exe}): These metrics (total or averaged) are useful to evaluate the impact of data movement strategies, individual resource performance or the influence of the interconnection network.
– Resource usage (U): Represents the usage of resources throughout the benchmark execution and the achieved level of parallelism. It is defined as follows:

$$U = \frac{T_{exe}}{T}.$$

3 The NAS Grid Benchmarks

The *NAS Grid Benchmarks* [8] are presented as a data flow graph encapsulating an instance of a *NAS Parallel Benchmarks* (NPB) code in each graph node, which *communicates* with other nodes by sending/receiving initialization data.

Figure 1 shows the four families defined in NGB. Each benchmark comprises the execution of several NPB codes that symbolize scientific computation (flow solvers SP, BT and LU), post-processing (data smoother MG) and visualization (spectral analyzer FT). Like NPB, NGB specifies several different classes (problem sizes) in terms of number of tasks, mesh size and number of iterations.

4 Results

The proposed methodology does not attempt to measure the performance of the underlying Grid hardware, but the functionality, reliability and performance of the Grid environment. However, a clear understanding of the hardware configuration of the Grid resources will aid the analysis of the subsequent experiments. Table 1 shows the characteristics of the machines in our small research testbed,

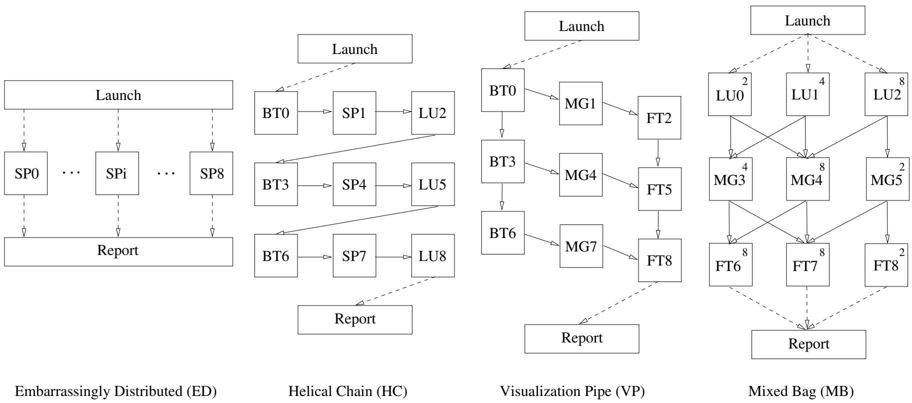

Fig. 1. The four families of the NAS Grid Benchmarks [8]

Table 1. Characteristics of the machines in the research testbed

Name	Site	Processors	Speed	Mem.	OS	DRMS
pegasus	UCM	Intel P4	2.4GHz	1GB	Linux 2.4	fork
hydrus	UCM	Intel P4	2.5GHz	512MB	Linux 2.4	fork
cygnus	UCM	Intel P4	2.5GHz	512MB	Linux 2.4	fork
cepheus	UCM	Intel PIII	600MHz	256MB	Linux 2.4	fork

based on the Globus toolkit 2.X. In the following experiments, cepheus is used as client and stores the executable and input files, and receives the output files.

Regarding the functionality criterion, the ED benchmark is a typical case of a parameter sweep application, directly supported by Grid*W*ay [14], the HC benchmark has been easily implemented using the Grid*W*ay DRMAA interface, and the VP and MB benchmarks have been programmatically implemented, through DRMAA, as workflow applications.

The HC benchmark constitutes an excellent probe to evaluate the reliability criterion of grids since output files of each task can be used as checkpoints for the next task. The metrics proposed to evaluate the reliability of the Grid environment have been implemented in the following way:

– Job cancellation or suspension: During the execution of a HC task, the job is cancelled and suspended. Grid*W*ay is able to detect the job cancellation when the task exit code is not specified and, in such case, to reschedule this task on other resource from the last saved checkpoint. Job suspension is detected when the task remains suspended longer than a given threshold.
– System crash or saturation: During the execution of a HC task, the resource where it is executing is saturated. Grid*W*ay is able to detect the performance degradation through a performance profile and, in such case, reschedule the job on a new resource from the last saved checkpoint. Resource failures are managed on an equal basis as network failures, as it is described below.

Table 2. Results obtained for each benchmark family

			Benchmark			
Metric	Description	Units	ED.A	HC.A	VP.A	MB.A
T	Turnaround time	minutes	18.88	17.57	21.67	16.80
P	Productivity	jobs/hour	28.60	30.73	24.92	32.14
T_r	Response time	minutes	-	3.09	-	-
T_{xfr}	Transfer time	minutes	5.50	7.10	8.10	9.70
T_{exe}	Execution time	minutes	38.30	7.38	22.93	23.03
U	Resource usage	-	2.02	0.42	1.06	1.37

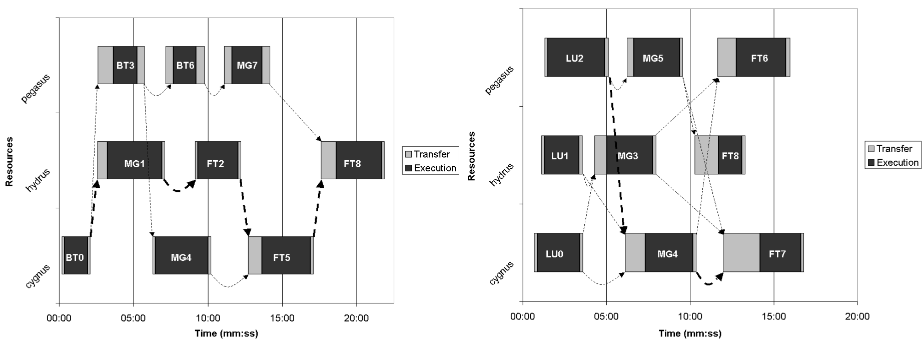

Fig. 2. Execution profile of the VP.A and MB.A benchmarks

– Network disconnection or saturation: During the execution of a HC task, the resource where it is executing is disconnected. Grid*W*ay is able to detect the disconnection by periodically probing the GRAM *job manager* in the remote resource. Grid*W*ay does not consider network saturation as a failure condition, instead, it uses network status information to rank resources [15].

Table 2 shows the values for the metrics proposed to evaluate the performance of the Grid environment. Grid*W*ay doesn't provide a measure for the response time, but can be calculated for HC.

The resource usage gives an idea of the characteristics of each benchmark family. ED is fully parallel, but the structural dependencies in the testbed (only three candidate resources) prevents a value of U close to 9. On the contrary, HC is fully sequential so it can not use the resources efficiently.

Figure 2 shows the execution profile of benchmarks VP and MB. Both exhibit some degree of parallelism ($U > 1$), that could be increased by widening the pipe (limited to three jobs) and by reducing the Grid overhead. The parallelism obtained by VP is also limited by the stages of filling and draining the pipe. MB has a wider pipe width from the beginning, which enables a better use of the resources. Both benchmarks are of great help to adjust the services of a grid, and even to compare different strategies to schedule workflows.

5 Conclusions

The presented proposal of criteria and metrics, along with the NGB suite, could be considered the first step to reach an agreed evaluation methodology for grids. The methodology provides diagnostic information of interest for middleware developers and Grid architects, that can be used to explore the behaviour and adjust the performance offered by each layer in a Grid environment. The use of standard interfaces allows the comparison between different Grid implementations, since neither NGB nor DRMAA are tied to any specific Grid middleware.

References

1. Anderson, E., Bai, Z., Bischof, C., et al.: LAPACK Users' Guide. SIAM (1999)
2. : The Standard Performance Evaluation Corporation. (http://www.spec.org)
3. Bailey, D.H., Barszcz, E., Barton, J.T.: The NAS Parallel Benchmarks. Intl. J. Supercomputer Applications **5** (1991) 63–73
4. Wolski, R., Spring, N., Hayes, J.: The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. Future Generation Computing Systems **15** (1999) 757–768
5. Chun, G., Dail, H., Casanova, H., Snavely, A.: Benchmark Probes for Grid Assessment. In: Proc. 18th Intl. Parallel and Distributed Processing Symposium (IPDPS). (2004)
6. Tsouloupas, G., Dikaiakos, M.D.: Gridbench: A Tool for Benchmarking Grids. In: Proc. 4th Intl. Workshop on Grid Computing (GRID 2003), IEEE CS (2003) 60–67
7. Snavely, A., Chun, G., Casanova, H., et al.: Benchmarks for Grid Computing: A Review of Ongoing Efforts and Future Directions. ACM Sigmetrics PER **30** (2003)
8. Frumkin, M.A., Van der Wijngaart, R.F.: NAS Grid Benchmarks: A Tool for Grid Space Exploration. J. Cluster Computing **5** (2002) 247–255
9. Huedo, E., Montero, R.S., Llorente, I.M.: A Framework for Adaptive Execution on Grids. Software – Practice and Experience (SPE) **34** (2004) 631–651
10. Herrera, J., Huedo, E., Montero, R.S., Llorente, I.M.: Developing Grid-Aware Applications with DRMAA. In: Proc. 10th Intl. Conf. Parallel and Distributed Processing (Euro-Par 2004). Volume 3149 of LNCS. (2004) 429–435
11. Berman, F., Wolski, R., Casanova, H., et al.: Adaptive Computing on the Grid Using AppLeS. IEEE Trans. Parallel and Distributed Systems **14** (2003) 369–382
12. Vadhiyar, S., Dongarra, J.: A Performance Oriented Migration Framework for the Grid. In: Proc. 3rd Intl. Symp. Cluster Computing and the Grid (CCGrid). (2003)
13. Lee, H.M., Chung, K.S., Jin, S.H., et al.: A Fault Tolerance Service for QoS in Grid Computing. In: Proc. Intl. Conf. Computational Science (ICCS). LNCS (2003)
14. Huedo, E., Montero, R.S., Llorente, I.M.: Experiences on Adaptive Grid Scheduling of Parameter Sweep Applications. In: Proc. 12th Euromicro Conf. Parallel, Distributed and Network-based Processing (PDP2004), IEEE CS (2004) 28–33
15. Montero, R.S., Huedo, E., Llorente, I.M.: Grid Resource Selection for Opportunistic Job Migration. In: Proc. 9th Intl. Conf. Parallel and Distributed Computing (Euro-Par 2003). Volume 2790 of LNCS. (2003) 366–373