Evaluating NTU Campus Grid Performance with NPB/NGB

Liang Peng, Simon See, Jie Song, Yueqin Jiang\textsuperscript{1}, Appie Stoelwinder, Hoon Kang Neo, ChiHung Chan\textsuperscript{2}, and Hee Khiang Ng\textsuperscript{3}

\textit{Asia Pacific Science and Technology Center, Sun Microsystems Inc.}
\textit{Nanyang Center for Supercomputing and Visualization,}
\textsuperscript{1}\textit{School of Electrical and Electronics Engineering}
\textsuperscript{2}\textit{Managed Computing Competency Center}
\textsuperscript{3}\textit{School of Computer Engineering}
\textit{N3-1-C, 50 Nanyang Avenue, Nanyang Technological University, Singapore 639798}

\{pengliang, simon, songjie, apple, norman\}@apstc.sun.com.sg
\{jiangyueqin\}@pmail.ntu.edu.sg
\{chihung\}@mc-3.org.sg
\{mhkng\}@ntu.edu.sg

Abstract: Computational grids provide scientific and engineering computing a platform which is technically and semantically different from traditional parallel and distributed systems. The performance evaluation in computation grid systems is difficult and very little research effort has been put on grid performance because of the complexity of the grid environments. The performance evaluation approach and metrics need to be reconsidered for the new grid environments.

In this paper, we try to analyze the performance metrics like response time, grid efficiency, and system utilization. Meanwhile, instead of calculating the grid system utilization in the traditional way, we present a concept of relative grid utilization, which describes how close is a single grid application performance to the performance of the same application submitted without grid middleware. We use NAS Parallel Benchmarks and NAS Grid Benchmarks to simulate the workloads with Poisson distribution and evaluate the APSTC cluster grid and NTU Campus Grid performance according various job arrival rate. Our results show that the system overhead of job submission via NTU Campus Grid ClearingHouse portal is larger than that of submissions within the cluster grid.

1 INTRODUCTION

Computational grids provide larger number of computing resources, more dynamic environment and diverse platforms for modern scientific computing than traditional parallel and distributed systems do. In the meantime, these inherent features of computational grid environment arise challenging research topics in Grid performance.

The traditional performance metrics are not necessarily appropriate for grid environments or insufficient to describing the characteristics of grids. Novel and emerging grid applications may need different terms of grid performance. So far there is no widely adopted performance metrics for computational grid environment. This is mainly due to the high complexity and dynamism nature of the grids. In this paper, we try to present some analysis on the grid performance metrics (mainly the response time, grid efficiency, and the system utilization).

With a technically and semantically different environment [NS02] comparing with the traditional parallel and distributed computing environment, the grid performance evaluation put more focus on examining the allocation of computing resources with respect to the requirements of the grid jobs. Pure speedup may not be the dominant concern of the users. In this paper, we mainly examine the efficiency of grid middleware like Sun Grid Engine and Globus.

NGB (NAS Grid Benchmarks) [dWF02] is a recently proposed Grid benchmarks based on widely used NAS Parallel Benchmarks. NGB is one of the few developed
benchmarks for computational grids. In this paper, we introduce the organization of our Sun cluster grid and NTU (Nanyang Technological University) Campus Grid [ntu], which an established and running grid computing environment. The performance evaluation results by using NGB is also presented. In our experiments, We mainly focus on the turnaround time and CPU utilization of the cluster grid. Our results shows that the overhead of SGE and Globus middleware is negligible especially for bigger problem sizes. Meanwhile, the NGB results show very low resource utilization, and this implies the traditional system utilization may not be suitable for grids and relative grid utilization could be a better metric.

The remainder of this paper is organized as follows: In Section 2 we analyze some performance metrics in computational environments; Section 3 describes the Sun Grid Engine as a grid computing middleware. Section 4 introduces NGB benchmark suite. The experimental results are presented and analyzed in Section 5. Some related work are introduced in Section 6 and finally we give a conclusion in Section 7.

2 GRID EFFICIENCY AND UTILIZATION

There are very few performance metrics particularly defined for computational grids. In traditional parallel computing, response time (or turnaround time) is a major concern of user, and system utilization is an important metric from the perspective of system engineers/administrators. In computational grid environment, although sometimes users submit job to grid because of the reasons other than speedup, response time still remains an important factor in consideration.

For a single grid job, response time can be defined as $T_{\text{end}} - T_{\text{submit}}$, where $T_{\text{end}}$ is the time when the job is finished, and $T_{\text{submit}}$ is the time when the job is submitted to the system. For a number of jobs, sometimes we also use average response time, which can be calculated as $\Sigma(T_{\text{end}} - T_{\text{submit}}) / N$, where $N$ is the total number of submitted jobs.

In traditional parallel computing, the system utilization can be computed as the ratio of the achieved speed to the peak speed of a given computer system. With this definition, the system utilization is usually very low (typically ranging from 4% to 20% ( [HX98])). This concept can also be applied to computational grid environment, but it can be expected that the system utilization could be even lower. Therefore, for single grid job, we find it more appropriate to define the relative grid utilization based on the traditional parallel system resource utilization: $(\text{system utilization of grid job submission}) / (\text{system utilization of parallel job submission})$. This concept reveals how close is the grid application to parallel execution on the same machines, instead of calculate the "absolute" value of utilization. Since applications running on computation grid environment is supposed to have more overhead and is expected to be slower, we can treat the corresponding parallel system as the upper bound of the application.

Another way to measure grid utilization is calculating the overall ratio of consumed CPU cycles and the available computational resources defined in [SOB03], i.e. the grid efficiency. With the consideration of multiple components within single grid job, an improved definition of grid utilization is presented as follows:

\[
U_G = \frac{\sum_{i=1}^{N_{\text{jobs}}} \sum_{j=1}^{\Sigma_{\text{CPUs}}} (T_{\text{end}_{i,j}} - T_{\text{submit}_{i,j}}) \times P_j}{(T_{\text{end}_{\text{last job}}} - T_{\text{sub_{first job}}}) \times \Sigma_{k=\text{servers}} N_k \times P_k}
\]

where $T_{\text{end}}$ is the time when the job is finished, and $T_{\text{sub}}$ is the time when the job is submitted, $P_j$ is the speed of $j$th CPU where component $j$ is running, and $N_k$ is the total number of CPUs on $k$th server.

3 CLUSTER GRID AND NTU CAMPUS GRID

The cluster grids are the simplest form of a grid which can be used to compose higher logical level enterprise/Campus grids and global grids. The key benefit of cluster grid architecture is to maximize the use of compute resource and increase throughput for user jobs.

The cluster grid architecture can be divided into three non-hierarchical logical tiers: Access tier, Management tier, and Compute Tier.

The access layer provides the means to access the cluster grid for job submission, administration and so on. The management layer provides the major cluster grid services such as job management, monitoring, NFS, etc. The computer layer provides the compute power for the cluster grid, and supports the runtime environments for user applications.

The performance of higher level grids largely relies on that of lower level grids. In order to evaluate the performance of the enterprise/campus level grids, classifying performance of cluster grid is necessary and meaningful.

The cluster grid in APSTC is a part of NTU campus grid, which consists of multiple cluster grids running
SGE/Globus at different schools and the whole campus grid is managed by ClearingHouse. The cluster grid of APSTC is illustrated in Figure 1. APSTC cluster grid consists of two Sun Fire V880 servers (totally twelve CPUs) with Sun Grid Engine (SGE). One server is the master host, and both servers are submission host and execution host.

The SGE distributed resource management (DRM) software is the core component of the Sun Cluster Grid software stack. It provides the DRM functions such as batch queuing, load balancing, job accounting statistics, user-specifiable resources, suspending and resuming jobs, and cluster-wide resources. The procedure of job flow in Sun Grid Engine is illustrated in Figure 2. In this job flow, each step may result in extra overhead in job submission and execution. In order to give a brief overview of the overhead as well as the resource utilization, we use NGB to do a performance evaluation on our cluster grid.

The cluster grids in the NTU campus grid can run SGE or Globus, or both. In our scenario, when jobs are submitted from NTU campus grid ClearingHouse portal, it is forwarded to the local Globus gatekeeper and handled by Globus. Another approach is to integrate Globus with local SGE. But we do not use this mixed approach in our experiments in order to separate their individual effects on performance. A demonstration is shown in Figure 3. In this scenario, when jobs are submitted from ClearingHouse client, it is sent to the ClearingHouse Server. The ClearingHouse server will record some relative information in local database and then forward the job request to the user-selected cluster grid, which is handled by its local Globus gatekeeper. There will be some procedure like user account mapping, CA checking, etc. Finally the job will be executed there.

4 NAS GRID/PARALLEL BENCHMARKS

4.1 Grid Benchmarking

A benchmark is a performance testing program that is supposed to capture processing and data movement characteristics of a class of applications. In performance benchmarking, selection of benchmark suite is critical. The benchmark suite should be able to test the affecting factors in the system and results should reveal the characteristics of the systems reasonably.

Strictly speaking, there is no complete grid benchmarks existing for grid platforms like the parallel benchmarks for parallel computing systems, mainly because of the inherent complexity of the grids. Grid benchmarks should take into account more factors which are related to the grid environment but are not major considerations in traditional high performance computing system. For example, the grid benchmark designers may need to think about the various types of grid jobs which may possibly consists of multiple applications running over wide area networks.
4.2 NAS Grid Benchmarks

NGB (NAS Grid Benchmarks) [dWF02] is a recently proposed benchmark suite for grid computing. It is evolved from NPB (NAS Parallel Benchmarks) [BBB+94], which was designed and widely used for performance benchmarking on parallel computing systems.

In NPB, there are eight benchmarks (i.e. BT, CG, EP, FT, IS, LU, MG, SP) on behalf of various types of scientific computation (for more details please refer to [BBB+94]). In current NGB, there are four problems representing four different typical classes of grid applications:

- Embarrassingly Distributed (ED) represents a class of applications which execute the same program for multiple times with different parameters. In ED, the SP program, which is selected from NPB, is executed for several times depending on the problem size. There is no data exchange between each execution of SP, so it is very loosely coupled.

- Helical Chain (HC) stands for long chains of processes which are executed repeatedly. Three programs, BT, SP, and LU, are selected from NPB. During execution, the output of one program is fed into another, and these procedures repeat several times.

- Visualization Pipeline (VP) consists of three NPB programs: BT, MG, and FT, which fulfill the role of flow solver, post processor, and visualization module respectively. This triplet simulates a logically pipelined process.

- Mixed Bag (MB) is similar to VP except that it introduces asymmetry. In MB, different volumes of data are transferred between different tasks and the workload of some tasks may be more than the others.

NGB contains computation-intensive programs and it mainly addresses grid computing system’s ability to execute distributed communicating processes. It does not specify how to implement or choose other grid computing components such as scheduling, grid resource allocation, security, etc.

5 EXPERIMENTAL RESULTS
5.1 Testbed and Workloads

Our benchmarking is performed on APSTC cluster grid, which consists of a four UltraSPARC III CPU (900MHz, 8GB memory) node (hostname: sinope) and an eight UltraSPARC III CPU (750MHz, 32GB Memory) node (hostname: ananke). They all run Solaris with Sun Grid Engine 5.3p4 and Globus toolkit 2.4. All NGB benchmarks are compiled with JDK1.4.

The jobs can be either submitted to the cluster grid locally, or submitted from the NTU campus grid Clearing-House portal.

We use NPB programs to simulate the workload of NTU campus grid. Specifically, we use all the eight benchmarks in NPB suit with problem sizes S, W, A, and B. The jobs that require relatively short CPU time (S size) take 10% of the total number of jobs; the jobs that require relative long CPU time (B size) also take 10% of the total number of jobs. The rest 80% of the jobs evenly consists of W and A size jobs. We submit totally 200 jobs in each run. The jobs distribution is the Poisson distribution, and the arrival rate is various.

In order to test the performance of processing multiple job submission, we use NPB benchmarks to simulate the NTU campus grid workloads for SGE running on the cluster grid. The NPB jobs are submitted to the Sun Grid Engine on APSTC cluster grid. The average response times of the NPB jobs (when $\lambda = 0.02$ are listed in Table 1. and

<table>
<thead>
<tr>
<th>Bmk/Size</th>
<th>S</th>
<th>W</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>BT</td>
<td>3.99</td>
<td>102.96</td>
<td>667.59</td>
<td>2840.08</td>
</tr>
<tr>
<td>CG</td>
<td>60.82</td>
<td>297.21</td>
<td>331.48</td>
<td>991.67</td>
</tr>
<tr>
<td>EP</td>
<td>7.15</td>
<td>28.22</td>
<td>65.02</td>
<td>269.38</td>
</tr>
<tr>
<td>FT</td>
<td>15.2</td>
<td>151.96</td>
<td>177.64</td>
<td>644.78</td>
</tr>
<tr>
<td>IS</td>
<td>2.31</td>
<td>31.47</td>
<td>58.01</td>
<td>72.01</td>
</tr>
<tr>
<td>LU</td>
<td>11.27</td>
<td>310.67</td>
<td>468.85</td>
<td>2271.95</td>
</tr>
<tr>
<td>MG</td>
<td>13.56</td>
<td>290.7</td>
<td>326.65</td>
<td>385.69</td>
</tr>
<tr>
<td>SP</td>
<td>60.45</td>
<td>196.06</td>
<td>694.11</td>
<td>1615.6</td>
</tr>
</tbody>
</table>

Table 1: Average Response Times of NPB Benchmarks when $\lambda = 0.02$ (in Seconds).

the overall average response time for $\lambda = 0.02$ is 419.89.

The average response times of the NPB jobs (when $\lambda = 0.03$ are listed in Table 2. and the overall average response time for $\lambda = 0.03$ is 629.89.

The average response times of the NPB jobs are listed in Table 3.

<table>
<thead>
<tr>
<th>Bmk/Size</th>
<th>S</th>
<th>W</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>BT</td>
<td>22.99</td>
<td>332.90</td>
<td>997.93</td>
<td>3686.75</td>
</tr>
<tr>
<td>CG</td>
<td>363.01</td>
<td>445.91</td>
<td>636.05</td>
<td>1209.82</td>
</tr>
<tr>
<td>EP</td>
<td>315.79</td>
<td>517.53</td>
<td>291.87</td>
<td>876.42</td>
</tr>
<tr>
<td>FT</td>
<td>211.80</td>
<td>312.26</td>
<td>529.87</td>
<td>1598.30</td>
</tr>
<tr>
<td>IS</td>
<td>590.94</td>
<td>323.74</td>
<td>452.51</td>
<td>157.03</td>
</tr>
<tr>
<td>LU</td>
<td>74.23</td>
<td>120.75</td>
<td>692.41</td>
<td>2520.82</td>
</tr>
<tr>
<td>MG</td>
<td>223.09</td>
<td>436.05</td>
<td>526.90</td>
<td>1293.48</td>
</tr>
<tr>
<td>SP</td>
<td>115.29</td>
<td>1542.80</td>
<td>785.32</td>
<td>2770.77</td>
</tr>
</tbody>
</table>

Table 2: Average Response Times of NPB Benchmarks when $\lambda = 0.03$ (in Seconds).

the overall average response time for $\lambda = 0.03$ is 780.48 seconds.

According to the description in Section 2, we can calculate the overall average response time is 780.48 seconds.

The average response times of the NPB jobs (when $\lambda = 0.05$ are listed in Table 4. and the overall average response time for $\lambda = 0.05$ is 1116.97.

In our experiments it is noted that the smaller size jobs may not necessarily have smaller response time than larger size jobs depending on the queue length.

Figure 4 shows how the overall average response time is affected by the job arrival rate $\lambda$. When job arrival rate is small (which means the average time interval of jobs is large), the average response time is shorter than the one when the job arrival time is large. This happens no matter the jobs are submitted to the cluster grid locally or submitted from NTU campus grid Clearing-House portal. However, for the job submissions via Clearing-House, since there is extra overhead of the portal and globus, the average job response time is generally longer than that of local cluster grid submissions.
<table>
<thead>
<tr>
<th>Bmk/Size</th>
<th>S</th>
<th>W</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>BT</td>
<td>1817.25</td>
<td>1625.96</td>
<td>2102.54</td>
<td>3081.64</td>
</tr>
<tr>
<td>CG</td>
<td>7.71</td>
<td>1570.19</td>
<td>1433.28</td>
<td>1010.33</td>
</tr>
<tr>
<td>EP</td>
<td>18.93</td>
<td>152.37</td>
<td>275.98</td>
<td>392.74</td>
</tr>
<tr>
<td>FT</td>
<td>10.99</td>
<td>1130.31</td>
<td>1710.33</td>
<td>2382.17</td>
</tr>
<tr>
<td>IS</td>
<td>337.56</td>
<td>1608.2</td>
<td>507.19</td>
<td>59.1</td>
</tr>
<tr>
<td>LU</td>
<td>827.27</td>
<td>892.41</td>
<td>1900.45</td>
<td>2842.4</td>
</tr>
<tr>
<td>MG</td>
<td>318.62</td>
<td>1395.13</td>
<td>1420.35</td>
<td>381.8</td>
</tr>
<tr>
<td>SP</td>
<td>388.34</td>
<td>1621.11</td>
<td>706.81</td>
<td>1813.62</td>
</tr>
</tbody>
</table>

Table 4: Average Response Times of NPB Benchmarks when $\lambda = 0.05$ (in Seconds).

### 5.2 Resource Utilization

Resource utilization is another major concern in grid computing. At this moment, we mainly consider the CPU utilization. Table 5 shows resource utilization of the NGB on our cluster grid. We calculate the CPU utilization by dividing the performance of the benchmarks (in MFlop) by theoretical peak performance. In our experiments, the CPU utilizations in all cases are very low (far less than 1%). The low utilization of the cluster grid reveals that the traditional utilization metric may not be appropriate for grids. So in Table 5, we also show the relative grid utilizations which are calculated according to the description in Section 2.

### 6 RELATED WORK

Very little efforts has been put in computational grid performance, although a lot work has been done in performance evaluation [Jai92] and benchmarking [Eig01] for traditional high performance systems.

Zsolt Nemeth analyzed the grid performance evaluation on grid in [Nem04]. He argue that the grid environment is technically and semantically different from the traditional parallel and distributed system [NS02], so the performance metrics need to be revised or created to describe the new features of the grids. However, he did not propose any novel grid performance metrics in his paper.

The most recently work include grid job superscheduler architecture and performance in computational grid environments by Shan et al. [SOB03]. In their work they propose several different policies for superschedulers and use both real and synthetic workloads in simulation to evaluation the performance of the superschedulers. They also present several grid performance metrics including response time and grid efficiency. But the their definition of grid efficiency does not consider the situation where the sub-jobs within a single job are computed on different servers (with different CPU speeds), and this concept is improved in our work.

There is a working group in GGF [ggf] working on grid benchmarking, but so far no detailed results have been published or declared. There is also a grid performance working group in GGF and they proposed a grid monitoring architecture [TAG+02] as well as a simple case study [AGS+02]. They mainly use a producer-consumer...
model to define the relationship between the nodes. Their architecture is more or less a specification of the required functionality in grid monitoring, and they adopt a model consists of producer, which generates and provides the monitoring data, consumer, which receives the monitoring data, and directory service, which is responsible for maintaining the control information or meta-data. They pose many design issues and problems need to be considered but without in-depth description of solutions, and currently it has not been implemented. The producer-consumer-directory service architecture mainly describes the scenario of how the performance can be monitored. But it is basically a simplified specification and many important open issues (like scalability, performance, etc) are not addressed.

Some initiative work has been done by NASA people based on NAS Grid benchmarks [dWF02, BBB“94, FSJY03]. They also did some work on tools and techniques for measuring and improving grid performance [BFSdW02].

Hamscher et al. tried to evaluate grid job scheduling strategies [HSSY00] with a simulation environment based on discrete event simulation instead of running benchmarks or applications on grids. Their performance metrics are also common criteria like average response-time and utilization of the machines.

GridBench [gri] is a tool for benchmarking grids and it is a subproject of CrossGrid [cro]. GridBench provides a framework for the user to run the benchmarks on grid environments by providing functions like job submission, benchmarking results collection, archiving and publishing. Although some traditional benchmark suites (like Linpack, NPB, etc) are revised and included by GridBench, currently it is mainly focused on providing the portal and environment for users instead of developing benchmarking applications. The GridBench people also discuss the grid benchmark metrics, but so far no novel metrics are proposed and measured.

There are also some benchmark probes for Grid assessment done by Chun et al. [CDCS02]. They developed a set of probes that exercise basic grid operations by simulating simple grid applications. Their probes fall into the low level measurement on basic grid operations such as file transfers, remote execution, and queries to Grid Information Services. The collected data include compute times, network transfer times, and Globus middleware overhead. They also declare that they are focusing on data-intensive benchmarks based on applications in domains such as bioinformatics or physics. Basically their problems are rather simple. They mainly measure the performance of pipelined applications which transfer a large volume of data from database node to compute node and transfers the result file to the results node. The real data grid situation can be much more complex and more sophisticated models are needed.

Performance forecasting in metacomputing environment also has been explored in FAST system [Qui02] by Quinson et al. The FAST system heavily relies on the Wolski et al.’s Network Weather Service [WSH99]. It also provides routine benchmarking to test the target system’s performance in executing the standard routines so that a prediction can be made on these results.

7 CONCLUSION

Grid performance evaluation is a not much explored area mainly because of the complexity of the grids. In this paper we present some preliminary analysis and experimental results on grid performance evaluation. We use NGB/NPB to evaluate the APSTC cluster grid and NTU Campus Grid. Our experiments with NGB show that the average job response time is heavily affected by the average job arrival rate, and the campus grid has more overhead than the cluster grid because of the extra work done by Globus and ClearingHouse job submission portal. We also show that traditional resource utilization may not be appropriate for computations grids and relative grid utilization could be a more suitable metric. Our ongoing work on campus grid performance evaluation will perform some deeper analysis of the experimental results, the performance evaluation of the whole campus grid, defining new performance metrics to describe and measure the character of the grids.

8 ACKNOWLEDGEMENT

We thank Professor Francis Lee of NTU campus grid team for providing us the relative information about the campus grid and their cooperation for our work.

References


