Computation Cost in Grid Computing Environments

Enis Afgan, Purushotham Bangalore
University of Alabama at Birmingham, CIS Dept.
1300 University Blvd., CH 131
Birmingham, AL 35294
{afgane, puri}@uab.edu

Abstract

Computational grid is an upcoming environment based on open standards and virtualization of individual resources created with a goal of supporting collaboration and resource sharing. As the technology advances and user base grows, grid computing will advance beyond research institutions and find its place in the commercial sector, in need of supporting the service-oriented goals imposed by such community. With the advent or various services, Quality of Service (QoS) levels will need to be shaped to provide quantitative and qualitative measure of service satisfaction for various user categories. A major concern among these users will be the associated cost for individual services and their ability to compare these costs and services between different providers. This paper raises some initial issues in the given area, analyzes only the few currently available approaches and provides the foundation for future work in terms service comparison and cost analysis.

1. Introduction

Computational grid [1] is the emerging computational infrastructure introducing pervasive computing at low-cost due to the availability of its geographically distributed resources on as-needed basis. Grid enables coordination and sharing of these resources facilitating development of new applications which can take advantage of distributed resources and compose their power and functionality into a single virtual resource [2]. The upcoming set of applications will be able to solve large scale computational and data-intensive problems, previously limited by constraints of resource availability. Current state of grid infrastructure is built on Service Oriented Architecture (SOA) which supports the idea of encapsulating any form of functionality made available into a service [3]. Each service's functionality is exposed and available through standard interface definition [4]. The use of services allows creation of loosely-coupled and platform independent distributed applications based on established web service standards supporting transparency, adoption, and virtualization [3].

Beyond the application development process alone, after a grid application has been tested on local resources, it is deployed; i.e., the entire application (*i.e.*, source code, dataset, scripts) is transferred to the remote site, compiled on the remote host, and made available for execution. The end-result is grid applications incur significant effort involved in the application deployment process. The deployment process is often customized to the specific application and is often strictly dependent on available hardware, software, and the end-users. Once deployed, given application requires regular maintenance with tasks ranging from simple data backup and ensuring network data availability to software updating and maintenance of the area the resource resides in. These additional requirements incur additional costs not typically found in traditional software development process. Required tasks may require coordination of effort at multiple organizations resonating across user community through associated cost. The initial analysis of such considerations and tradeoffs user and resource owners must deal with in the new environment are explored and discussed in this paper.

The paper is organized as follows, section 2 motivates the problem through sample scenario. Section 3 presents current attempts at pricing available services within grid computing environment.

2. Motivation

As a brief sample scenario motivating cost related issues that will become more noticeable as the use of grid advances and service providers appear, we can observe what options a user has in executing a

29th International Conference on Software Engineering Workshops(ICSEW'07)
0-7695-2830-9/07 $20.00 © 2007 IEEE

computationally intensive application. A commonly used bioinformatics sequence analysis tool called Basic Local Alignment Search (BLAST) [5] was chosen as a test application. The choices the user has follow:

1. *Use NCBI[1] website to submit a BLAST search job for free*: while the access to the resources is simple and free based on a web interface, users are limited in number and size of jobs they can submit in a given amount of time. Queue times are possibly long and of variable length.
2. *Install the datasets and the application on local workstation*: this option initially requires expertise on installing and maintaining local system, which leads to questionable data security and backup options. Additionally, hardware capabilities of the system are very limited to exercise any sort of parallelism and high performance computation.
3. *Have BLAST installed on set of resources found at the local institution*: this option transitions into a distributed environment but requires action by local system administrator(s) in terms of application installation and maintenance. This may raise additional issues within the organization dealing with ownership, policy management, and allocation time, to name a few. The computational power for given application is limited to the number of locally available resources and to the load of individual systems.
4. *Use resources available in the grid*: the new technology enables a service-oriented environment where users are exposed to an unlimited number of resources offering needed service. In this environment, individual resources are maintained by an owner who advertises availability of various services (such as BLAST application) and handles the necessary maintenance tasks on behalf of the user. The process of job submission and management is handled automatically for the user through a predefined interface such as a web based portal or a custom application.

Using the grid, users receive a level of service and thus a cost must be associated with grid job execution. For service providers, deciding on how much to charge for the service is hard to decide on. Many factors come into play when maintaining a grid resource ranging from the man power to setup the system and perform daily maintenance to obtaining and renewing software licenses and paying for floor space and system cooling [6]. As the grid user and application base grow, the answers to some of these questions will be reached, but currently we are in the dark stages.

[1] http://www.ncbi.nlm.nih.gov/

3. Current approaches

While grid computing is still mostly restricted to research institutions where resources readily are shared with the common goal of advancing science and only method of cost is applied through allocation time on individual resources, some initial attempts have been made in the commercial world to provide users with grid-like functionality. Each of the devised efforts has proposed an appropriate cost model.

The most straight forward example of an early grid and matching price model is the Sun Grid [7]. The Sun Grid offers a pool of pre-installed applications as well as options for users to deploy their own applications on Sun's resources. The pricing model for this service is straight forward: $1/CPU-hour. This is an all-inclusive price encompassing job submission, result storage and retrieval, and even minimal workflow support. The repercussions of this pricing model are still to be seen though. If we continue with BLAST scenario and submit several jobs, comparison of execution parameters with the associated cost, as if we had run on Sun Grid, leads to following data (see Figure 1):

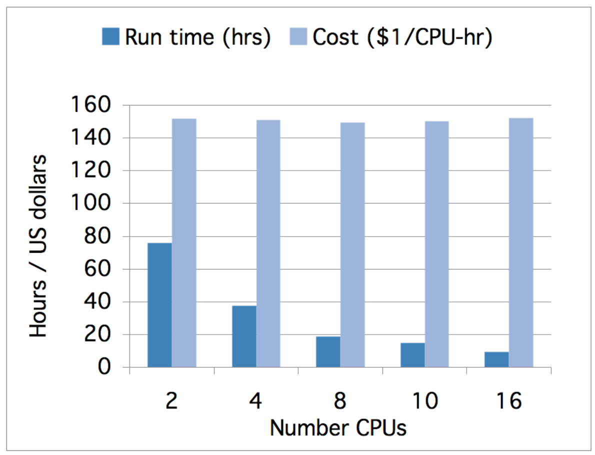

Figure 1. Comparison of cost vs. run-time on AMD-64 1.6. GHz Opteron when a 10,000 query file is used to search against nr (821 MB) database using query splitting BLAST.

The resource used during this testing is similar to the resources available through Sun Grid offering comparative performance. Since BLAST is an embarrassingly parallel application, speedup observed is linear and thus the cost remains leveled providing direct benefit to the user if additional resources are available. If we consider an alternative to using Sun Grid and decide to invest into purchasing a cluster, paying for the floor space, and an administrator to manage the system, we may obtain the data found in Table 1 [6, 8]:

Table 1. Itemized cost of cluster ownership (all figures are rough estimates).

Cluster purchase (32 nodes)	$100,000
System administrator	$15,000 / year (25% of salary)
Power/cooling/floor space	≈$10,000 / year
Software	$0
Resale of cluster after 5 years	$?
Total cost for 5 year ownership	**≈$225,000**

Comparing cost of using Sun Grid as a source of long-term computational resources and cost of purchased system, we conclude that each of the 32 CPUs should be used for a minimum of 10 minutes per day in a five year period where the cost of renting the same compute power would be evened out with the cost of the purchase. That is approximately 16% of the total 5 year period. It would appear that the obtained figures reasonably well justify purchase of a cluster rather than using a service provider such as Sun Grid. The obvious limitations to the above calculation deal with the availability of additional resources for times of high-use vs. times of inactivity. Also, the above scenario assumes cost of local software installation to be free, and even though today's tendency may be such, service provider may offer additional benefits and functionality included with the service price. Since there is cost associated with each design choice an application service provider decides to implement, the application service provider has to incorporate these costs within the cost for executing the application and could charge different prices for different versions of the application. From the perspective of a service provider, the available resources must be able to handle variable demand but also justify the initial investment. Sun is a computer vendor with access to low cost resources and may be financially viable for them to undertake the necessary investment, but the question of whether a third party provider can have a viable business remains to be seen.

Similar to Sun Grid, IBM offers On Demand Business service [9] where customers also experience the pay-as-you-go service thus minimizing investment and maintenance expenses associated with needed excess hardware and software. The actual pricing model has not been disclosed as is the case with Sun but is rather on per-customer basis.

Beyond commercial grids, research grids such as the TeraGrid [10] and SURAgrid [11] exist where the cost is not directly tied to currency but rather to allocation time. Regardless of how one approaches the grid, there is a cost associated with the available services. The problems of what, how, and how much to charge for available services requires further research and ultimately, experience.

4. Discussion

Current stage of grid development regarding association of cost with available services is at best primitive. Execution of grid applications is dependent on many factors. Resources found in the grid are heterogeneous resulting in variable application performance [12]. The results of such performance variability test are shown in Figure 2 below. Variable performance on individual resources provides multitude of scheduling alternatives and options while it also shows an obvious issue with the uniformity of the flat pricing model used by Sun Grid.

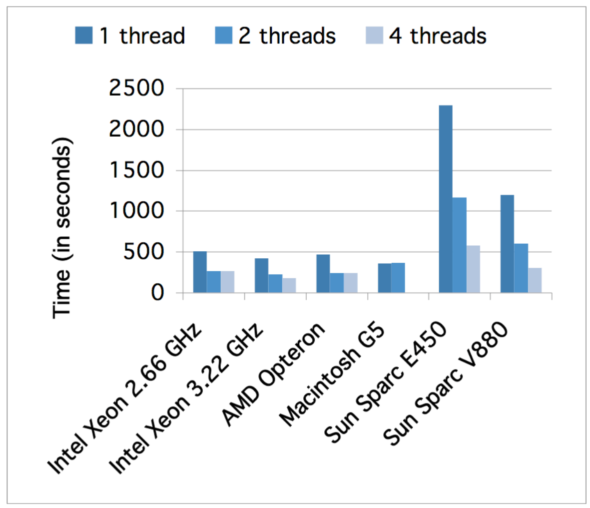

Figure 2. Comparison of performance of different processor types and BLAST search time (in seconds) against nr database and input file with 10 queries using shown number of threads.

Different algorithms available for the same application result in additional performance differences [12]. Such considerations offer additional customization options and opportunity to provide multiple levels of Quality of Service (QoS). Size of a job, run time, output data, network congestion, and realizing peak hours of a day, week, or month need all be considered in delivering the most appropriate service levels on individual user basis. As with other products widely available in the commercial sector today, tradeoffs between functionality (*e.g.*, time) and cost must be realized and noted. People are willing to pay for different service levels but the plausibility of fulfilling the contract must be supported by underlying methodologies and tools. This observation results in the need to establish multiple levels of service, each associated with the appropriate cost. At the current stage of development, at least one user category is at a loss due to sub-optimal performance of resources and/or pricing models. Further conclusions are not available today but in order to support realistic value-based computing, these questions must be answered.

5. Conclusions

There is a need within the research community to analyze issues raised in this paper as well as understand additional issues in order to provide support methods and tactics for easing the transition of grid acceptance. The goal is not only to be able provide associated value for a service supported by metrics and proofs, but also to advance QoS issues through the SOA model by provisioning optimized service on per-user and/or per-job level. The success and true validation of proposed methods are only going to be seen after a user base has been established and examined. The goal of this paper is not to present concrete solutions and results to the above issues and ideas, but rather to compose a solid foundation in terms of viable options and questions one needs to consider when dealing with the variability found in pricing options within grid computing.

6. References

[1] *The Grid: Blueprint for a New Computing Infrastructure*, 1st ed: Morgan Kaufmann Publishers, 1998.

[2] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid," *Lecture Notes in Computer Science*, vol. 2150, pp. 1-28, 2001.

[3] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Global Grid Forum, Open Grid Service Infrastructure Working Group June 22 2002.

[4] V. Silva, *Grid Computing For Developers*, 1st ed. Hingham, MA: Charles River Media,2005.

[5] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Mol Biol*, vol. 215, pp. 403-410, 1990.

[6] APC, "Determining Total Cost of Ownership for Data Center and Network Room Infrastructure," American Power Conversion November 16 2005.

[7] "Sun Grid," [Last accessed January 17th, 2007], Available from http://www.sun.com/service/sungrid/.

[8] L. A. Barroso, "The price of performance," *Queue*, vol. 3, pp. 48-53, 2005.

[9] "IBM - On Demand Business," [Last accessed January 17th, 2007], http://www.ibm.com/e-business/ondemand/us/index.html

[10] "TeraGrid," [Last accessed January 17th, 2007], http://www.teragrid.org.

[11] "SURAgrid," [Last accessed January 17th, 2007], Available from http://www1.sura.org/3000/SURAgrid.html.

[12] E. Afgan, P. Sathyanarayana, and P. Bangalore, "Dynamic Task Distribution in the Grid for BLAST," presented at Granular Computing 2006, Atlanta, GA, 2006.

29th International Conference on Software Engineering Workshops(ICSEW'07)
0-7695-2830-9/07 $20.00 © 2007 IEEE

IEEE COMPUTER SOCIETY