Identification of the Benefits of Grid Computing

Pinaki Chakraborty

School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi – 110067, India.
E-mail: pinaki_chakraborty_163@yahoo.com

Abstract

Grid computing creates the illusion of an enormously powerful computer system using a diverse assortment of simple computers. Grid computing provides many benefits that are inconceivable in simple computers and conventional distributed systems. The objective of current article is to theoretically identify the most important of these benefits. In a grid, underutilized resources including unutilized processing resources and unused disk spaces can be proficiently utilized. Various underutilized resources in the grid can be also harnessed using proper design techniques. Grid computing facilitates massive parallel processing that can be used in different disciplines. Grid-aware applications are now being developed to exploit the features of the grid which cannot be used by conventional applications. Grid computing can virtualize resources to provide interoperability among various types of computers in the grid. The other main benefits of grid computing are extensive load sharing and enhanced reliability. Moreover, grid computing is opening new avenues for research in system software and various other fields.

Keywords: *Grid computing, grid device driver, massive parallel processing, grid-aware application, autonomic computing*

1. Introduction

Grid computing [1,2] is still a new and important field of computer science that is growing rapidly. It can be naively described as a natural evolution of the distributed computing. Nevertheless, grid computing can be differentiated from conventional distributed computing by is objectives of large scale resource sharing, innovational applications and high performance.

The goal is to create the illusion of an extremely powerful computer system using a large collection of various types of simple computers and other resources interconnected by a network [1-3]. The prerequisite for the development of such a computer system is the availability of standards for communication and resource sharing between computers using different hardware and software architectures [3]. A grid computer system should support coordinated resource sharing among time variant groups of computers and allow their users to pursue common goals without any central control authority. A grid computer system should be flexible and secure. Such a system should facilitate coordinated problem solving for dynamic and multi-institutional organizations. Grid computer systems enable the development of large scientific and business applications on an unparalleled scale.

This paper provides a timely commentary on the major benefits of grid computing, like high degree of parallel processing and superior reliability, for a general scientific audience.

2. The Benefits

Grid computing possesses a large collection of benefits and merits that is unimaginable in a simple computer or even in a conventional distributed system [3]. This section lists some of the most important benefits and merits of grid computing.

2.1. Using Underutilized Resources

The processing resources are by far the most important resources in a computer system. In most organizations, processing resources are not utilized efficiently. As a result, a large amount of processing resource is left unutilized. In most of these organizations, most personal computers, workstations and other simple computers are busy only for a small fraction of time. In several organizations, even servers, mainframes and other large computers lie idle for considerable intervals of time. Grid computing can help in tapping these underutilized processing resources using the concept of resource sharing and to some extent the concept of load balancing.

To take advantage of the unutilized processing resource of another computer, a job can be executed on that machine. If the computer on which that job should run in a normal circumstance is busy then it can be migrated to another computer subjected to fulfillment of the following three preconditions.

1. The overhead associated with the migration and the remote execution of the job must not be too high.
2. The remote computer must be able to support the execution of the job and fulfill all requirements of the job.
3. The remote execution must not breach the security or degrade the reliability of the system, nor should it leave the system in an inconsistent state.

Although the processing resources are the most important resources in a computer, they are not the only important resources. Disk space is another important resource. Very often enormous amount of disk space is left unused. In a grid, these unused disk spaces can be used by formulating a distributed virtual memory (Figure 1). This will enhance the performance to cost ratio and facilitate the execution of grid-aware applications. Such a grid is also known as data grid.

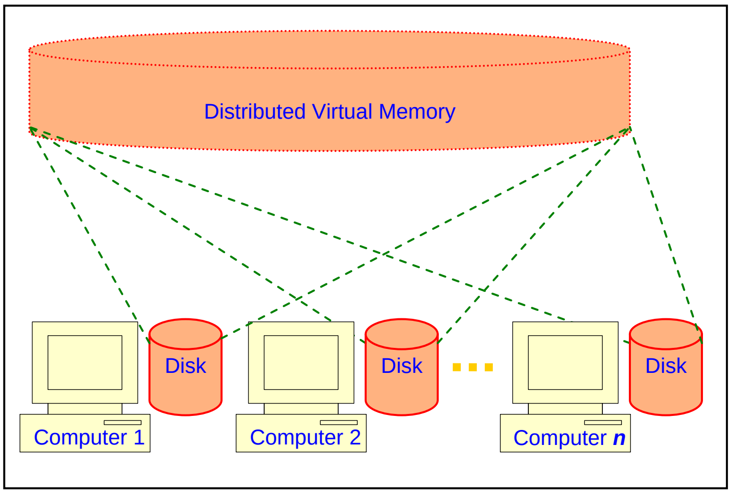

Figure 1: Formulation of a distributed virtual memory

Apart from processing and storage resources, a grid can provide access to a plethora of other resources like special equipments, software, licenses and other services. Network bandwidth, expensive licensed software and specialized hardware instruments like microscope, camera, scanner, printer and fax machine can be easily shared by the computers in a grid.

Conventional distributed systems and network systems use remote device drivers for accessing devices connected to other computers in the system. The next evolutionary step of these device drivers is the grid device drivers that will allow access to remote devices in the grid in a transparent manner. Such transparency will make the grid look like a large virtual computer with a collection of virtual resources beyond what is available on just one conventional computer.

2.2. Massive Parallel Processing

The potential for massive parallel processing (MPP) is one of the most lucrative promises made by grid computing. MPP is being used in various fields including bioinformatics, medicine, data mining, financial modeling, weather forecasting, remote sensing and animation.

Application desiring to use the MPP feature of a grid should use algorithms that can be partitioned into simultaneous and independently running portions. A computation intensive grid-aware application should behave like a collection of a large number of small jobs executing in parallel on different computers in the grid. In an ideal situation, an application takes ***n*** times less time to complete if it is parallelized and executed on ***n*** computers (Figure 2).

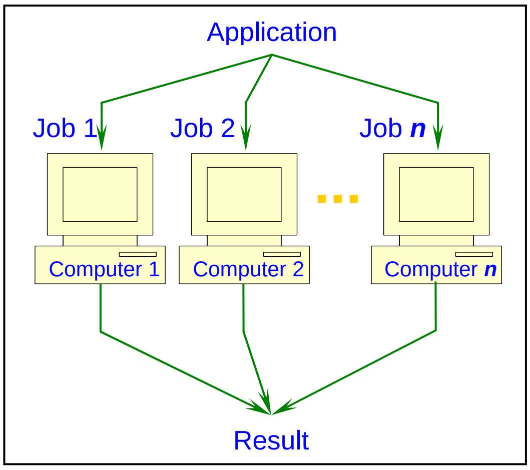

Figure 2: Parallel processing of an application

However, such efficient parallel processing is difficult to achieve and even more difficult to scale due to the following three reasons.

1. Most algorithms can be partitioned into only a small number of independent portions. The number of these independent portions may be very less in comparison to the number of available computers. Some algorithms cannot be partitioned at all.
2. The portions of the algorithm obtained after partitioning may not be absolutely independent. The two most important forms of dependencies in grid computing are flow dependency and resource dependency.
3. There are several implementation oriented issues that hinders MPP. These include protocols for communication and synchronization between the computers.

2.3. Grid-Aware Applications

An application is said to be grid-aware if and only if it has the knowledge of the existence of the different components of the grid and it uses one or more features of the grid. A grid-aware application is developed to be executed on a computer in a grid such that its performance is significantly better than its equivalent conventional application.

Unfortunately, there are no practical tools for transforming an arbitrary conventional application into its equivalent grid-aware version. Moreover, there is no guarantee that grid-aware applications generated by such a tool will be scalable. The techniques for automatic translation of applications are still in their infancy. Standardizing these techniques is difficult and requires best computer scientists and computer engineers. So, new computation intensive applications are being designed and implemented as grid-aware applications. These grid-aware applications are ready for highly scalable parallel processing. These applications are compiled using either parallelizing compilers or concurrency preserving compilers depending upon their source languages as illustrated in Figure 3.

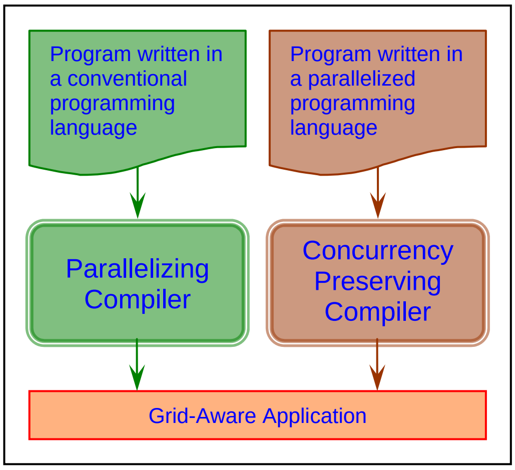
Program written in a conventional programming language
Program written in a parallelized programming language
Parallelizing Compiler
Concurrency Preserving Compiler
Grid-Aware Application

Figure 3: Compiling a grid-aware application

2.4. Virtualization of Resources

A grid can expand the potentialities of data usage in the organization. Files can seamlessly span over many computers and thus can be larger in size than on any single computer in the grid. Such spanning can enhance the rate of data transfer. Replication of data can be done for backup. Replicas of a data can be hosted on or near the machines most likely to need the data. Data is not the only resource that can be shared. Many other resources, including equipments, software, services and licenses, can be shared. These resources can be virtualized to give them a more uniform interoperability among various types of computers in the grid.

2.5. Load Balancing

Another significant benefit of grid computing is the possibility of load balancing. In the strictest sense, perfect load balancing is difficult to obtain in a distributed system and almost impossible in a grid. So, most researchers refer to this concept as load sharing. Load balancing or load sharing can be very useful in handling occasional flux of loads in one part of a large organization. This can be done in any of the following three ways.

1. If a particular computer in the grid receives an unexpectedly high number of jobs in a very short period of time, then some of the jobs can be routed to the relatively idle computers in the grid.
2. If all the computers in the grid are already fully utilized, then the lowest priority jobs being executed in the grid can be temporarily suspended and later resumed or even cancelled and later restarted to make room for the higher priority jobs.
3. The higher priority jobs can be executed on a faster computer and the ordinary computers can be used to execute the other mundane jobs.

2.6. Reliability

Reliability is an important issue in most modern computer systems. In a grid, if one computer crashes due to some unforeseen reason, the other computers in the grid are not generally affected. The operating system of the grid can automatically resubmit jobs of the crashed computer to the other computers in the grid. In critical and real-time applications, multiple copies of the same job are executed in parallel on different computers spread over the grid. The individual results of these computers are checked for any kind of inconsistency and the final result is obtained by voting among these computers (Figure 4). Such a scheme of processing is called autonomic computing.

Autonomic computing can automatically heal several problems in the grid even before the users are aware of them.

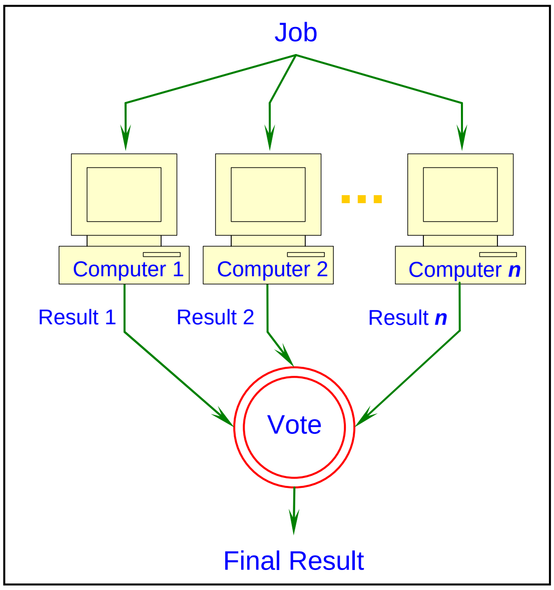

Figure 4: Enhancing reliability using autonomous computing

2.7. Scope for Research on System Software

Grid computing has opened a new avenue for the design and implementation of system software like device driver, compiler and operating system. The scope for research on device drivers and compilers has been discussed in subsections 2.1 and 2.3, respectively. The operating system running on the computers in the grid may be of two types. In the first option, each computer in the grid runs a part of the same operating system (Figure 5a). In the second option, each computer in the grid runs it own local operating system and a grid-wide middleware is constructed above the layer of these local operating systems to provide a single system image (Figure 5b). The first option is more efficient but requires considering the intricacies of designing and implementing operating systems. So, often the second option is preferred as it is easier to develop and performs satisfactorily in most cases.

3. Discussion

The previous section discusses the most prominent advantages of grid computing. These merits are numerous and diverse. However, the success of grid computing lies in several factors including the organizational structure, the application domain and the available resources. When properly designed and implemented, grid computing proves immensely beneficial for the organization. From the research point of view, grid computing is still a growing discipline and many concepts are yet to be standardized. The most challenging issues that are drawing mass attention [4] are as follows.

1. *Architectures of grid software*. The different feasible software architectures that will facilitate load balancing and resource sharing among applications and computers in the grid need be standardized.
2. *System software technologies*. Software technologies for scheduling, resource profiling, communication and efficient execution of grid-aware applications are required to be developed.
3. *Language support*. Languages, compilers, environments and other tools to develop grid-aware application should be developed.

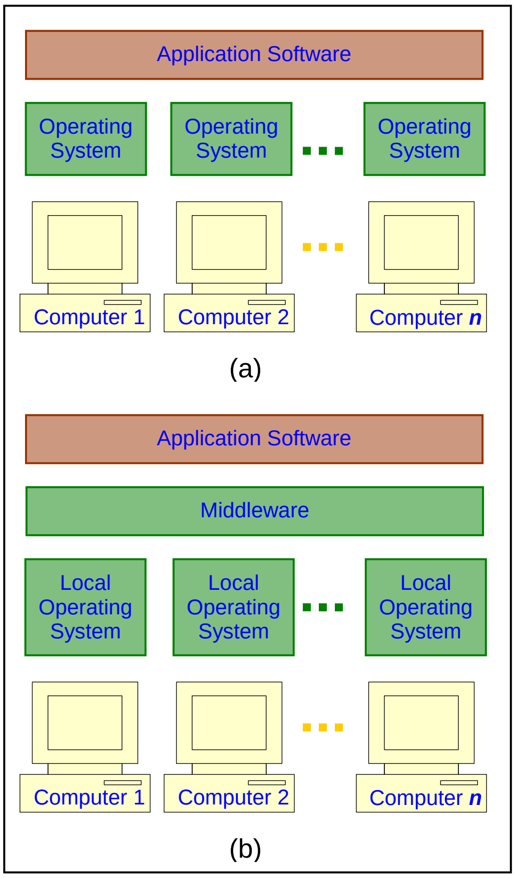

Figure 5: Operating systems in a grid (a) without any middleware and (b) with a grid-wide middleware

4. Conclusions

It can be concluded that grid computing is successful in creating a very powerful virtual computer out of the available resources. Grid computing handles a diverse collection of computers more uniformly than ever before. Grid computing allows sharing of available resources not only in a single project but also among the different projects in an organization. Collaborative utilization of available information and resources over a large set of projects can significantly enhance the productivity of a large organization.

5. Acknowledgements

The author would like to acknowledge late Professor R. G. Gupta for his perspective discussions with the author on the subject.

References

[1] Foster, I. and Kesselman, C. (Eds.). *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann, 1999.

[2] Foster, I., Kesselman, C. and Tuecke, S. The anatomy of the grid enabling scalable virtual organizations. *The International Journal of High Performance Computing Applications*, 2001, **15**(3), pp. 200-222.

[3] Ferreira, L., Berstis, V., Armstrong, J., Kendzierski, M., Neukoetter, A., Takagi, M., Bing-Wo, R., Amir, A., Murakawa, R., Hernandez, O., Magowan, J. and Bieberstein, N. *Introduction to Grid Computing with Globus*. 2nd ed., IBM Corp, 2003.

[4] Laforenza, D. Grid programming: some indications where we are headed. *Parallel Computing*, 2002, **28**(12), pp. 1733-1752.

Paper received: 2008-09-22