Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION, Corfu, Greece, August 17-19, 2005 (pp1-6)

Modified Application Centric Load Balancing Algorithm for Satellite-based Grids

BRAJENDRA K. SINGH
Space Applications Centre,
Indian Space Research Organization,
Ahmedabad – 380053
INDIA

HARESH S. BHATT
Space Applications Centre,
Indian Space Research Organization,
Ahmedabad – 380053
INDIA

AKSHAI K. AGGARWAL
School of Computer Science,
University of Windsor,
Ontario
CANADA

Abstract: - A load balancing algorithm for a heterogeneous distributed computing environment attempts to improve the response time of a parallel or distributed application by ensuring optimum utilization of available resources. When the compute-resources are geographically dispersed, as is the case for grids, these resources may be connected through low speed satellite networks. Hence network delay for data transfer may become a significant factor in the load balancing strategy. In this paper, we have enhanced the application centric load balancing strategy to take network delay into consideration in satellite-based grid computing. The modified algorithm uses a static scheme for allocation of computer and network resources. This can satisfy the QoS requirements of applications on the grid. We have extensively analyzed and tested the performance of our algorithm.

Key-Words: - Network delay, workload simulator, centralized static load balancing, satellite based grid computing etc.

1 Introduction

A grid [1, 2] seamlessly integrates geographically spread out compute-resources and it provides resources on demand to users. Sharing of multiple resources for processing an application may involve large amounts of data transfer from one resource to the other. When resources are located at large distances, the terrestrial networks may have to be augmented by satellite-based networks [3, 4].

Current data communications satellites like EDUSAT [5] provide 64 Kbps to 4 Mbps speed among the interconnected nodes. Future satellites like GSAT-4 [6] aim to provide data communication speed ranging from 30 Mbps to 100 Mbps. Satellite-based grid computing project [7] aims to extend the terrestrial grid to remote location. The satellite network is configured asymmetric where forward channel is high speed at 30 to 100 Mbps and return channel is low speed at 64 kbps. The high speed channel is used for sending or receiving the data while low speed channel is used for acknowledgement packets. The low speed channel is also used for communication of grid commands.
Although these high speed satellite channels can be used dedicatedly for an application, their speed is very low when compared to that of 10 Gbps terrestrial network. Current grid environments like CondorG [8], NimrodG[9] and OpenPBS[10] do not consider data communication delays due to the availability of high speed terrestrial networks. GRAM [11] implementation in Globus toolkit [12] has sequential input data staging, executable staging, computation and output data staging. But it does not include the consideration of communication networks. Ganesh [13] is capable of paralleling computation and communication tasks but it has not implemented network resource optimization in its application centric load balancer [14, 17, 18]. Satellite Grid Model [15] extends Ganesh to parallel communication and computation for applications like disaster management, which need to apply single algorithm on multiple data sets to reach the rescuing decisions. Its pattern generator generates directed acyclic graph (DAG), paralleling communication and computation for repetitive tasks, based on user requirement. It assumes that all the required resources are available to the application as and when required. It leaves the resource reservation and optimization task to the load balancer.

Application centric load balancer is able to optimize the computation resources among the different user applications. In this paper we have extended the same to consider the satellite data communication channels as resource. We optimize these

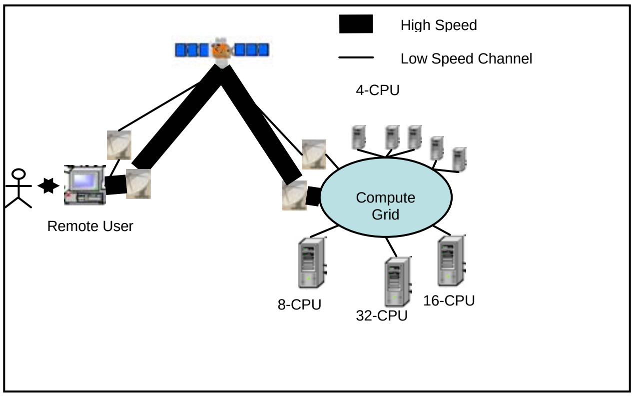

Fig.1: Single Compute Grid

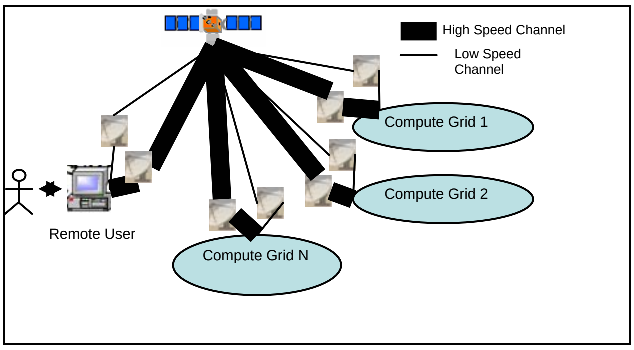

Fig.2: Multiple Compute Grids

communication resources along with computation resources with aim to complete each application as fast as possible and to complete as many applications as possible within a session.
The Modified Application Centric LOad Balancing Algorithm (MACLOBA), does a static allocation of compute and network resources. Allocation is made static so as to provide the promised QoS to the applications. We have used a large number of various types of applications generated by rule-based fuzzy simulator [19] to exhaustively test MACLOBA. Resources are assigned to these applications using MACLOBA for the simulated Ganesh environment and the performance of the algorithm is measured.

Though we have developed load balancing strategy for Ganesh, MACLOBA is generic in nature. It could be applied to any satellite-based grid computing environment with some customization as per the specific environment.

Satellite based grid computing is quite useful [16] for applications like disaster management, Mobile computing, Telemedicine, Tele-education, National Spatial Data Infrastructure (NSDI), Internet via satellite, audio and video distribution, financial data delivery, remote industrial control, managing activity of public administration at remote locations, database distribution etc.

2 Environment

In Ganesh, each compute-resource is called a node. A group of nodes, at a particular geographical location, is interconnected through 10 Gbps links. This group of nodes constitutes a localized Compute Grid Network (CGN) which could be extended to remote locations via satellite link. Fig.1 shows the extension of a single compute grid. Several such localized CGNs are located at large geographical distances apart from one another. The CGNs are connected among themselves through satellite links with speeds ranging from 2 Mbps to 100 Mbps. Fig.2 shows the extension to multiple compute grids.

3 Goals and Issues

Our goal is to analyze the effect of considering network delay in the performance of the load balancing algorithm. The parameters, which affect the performance of the load balancing algorithm, when network delays are considered, should be identified and adjusted properly.

First to accomplish the above goal, load balancing strategy should adopt static allocation scheme for the satellite grid environment. It means that network resources, once allocated, would not be withdrawn or reallocated due to QoS constraint of satellite grid networks. But it may lead to wastage of resources, in case the application does not use the resource at the allocated time. Secondly, in spite of emphasizing the static allocation, certain high priority requests may, during run time, need immediate attention.

In the modified load balancing algorithm, while CGNs are allocated statically, the node resources at a localized CGN could be dynamically withdrawn provided node resource reassignment does not lead to larger network delays.

4 The MACLOBA Model

Applications are generated by the Rule-based fuzzy simulator. A typical simulated application is shown in Fig. 4. A new simulator is designed that simulates the localized Compute Grid Networks (CGNs) and the connecting network links among them. The environment simulated by it is denoted NodeVector (n1,n2,…….nm). For example NodeVector (n1,n2,…….nm) signifies that n1 nodes are from CGN1, n2 node are from CGN2, and so on.

The applications are submitted to the simulated version of Ganesh environment which, in turn, submits these applications to the load balancer. The load balancer does the resource allocation using MACLOBA. Depending upon the type and location of resources, for a process of an application, the load balancer provides the resources through either high speed terrestrial network or through satellite based network.

5 The Modified Algorithm

The MACLOBA algorithm is as follows:

1. Take an application
2. Take a process to be allocated with start time ST(p)
3. Start trying allocation on nj th type of CGN of NodeVector (n1, … nj,….nm)
4. Find all the n parents of this process
5. Find the end time ET(i) of i th parent wrt ST(p).
6. Consider the data volume DV(i) of the i th parent
7. Calculate the data transfer time from the CGN of the i th parent to nj th CGN as
8. DT(i) = (DV(i)) / nj(speed))
9. where nj(speed) is the data transfer rate between the nj th CGN to the CGN of the i th parent
10. Calculate the earliest start time for the process
11. ΔT = Max { ET(i) + DT(i)} for all n parents
12. New start time of the process ST(p) = ST(p) + ΔT
13. Try allocation of the process, with this new start time, as per the earlier application centric load balancing algorithm and find the earliest start for the process on this nj th type of CGN
14. Repeat step 3 for all m types of CGN
15. Allocate the process at the minimum of earliest start times for the process across all m types of CGN
16. Repeat step 2 for all processes of an application
17. Repeat step 1 for all applications.

6 Example Applications for Performance Evaluation

Assumptions:

1. Load balancing strategy is static i.e. network resources, once allocated, will not be withdrawn or reallocated, except in case of higher priority requests.
2. Arrival pattern of applications, once assumed, will not be changed through out the session analysis.
3. Average process duration for the process is much higher than the overhead of the load balancer thread. Hence it is ignored for the session under consideration.
4. Only one process can transfer data through satellite network at a time i.e. multiplexing of satellite network resource is not allowed.
5. Only CPU bound and I/O bound processes are considered.
6. Nodes in a CGN are considered of the same class [6].
7. All applications are considered of the same priority.

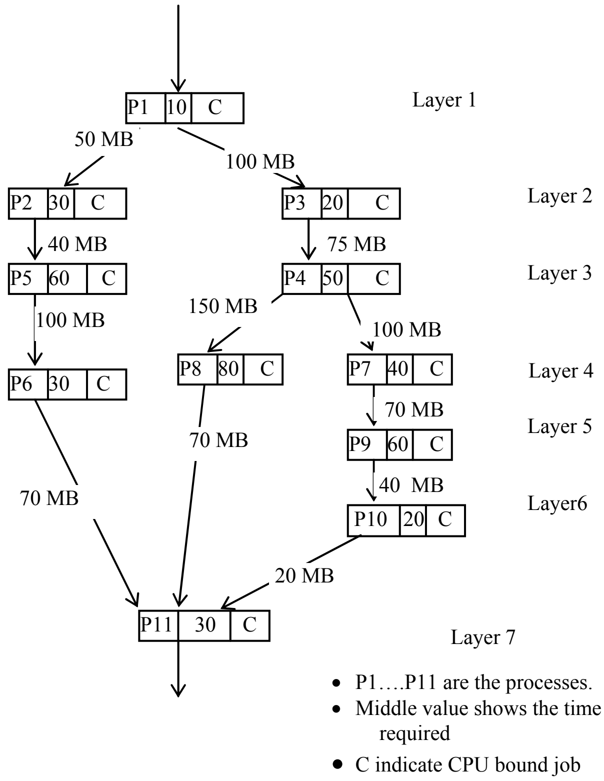

Fig.4: A typical application with its processes simulated

Rule-based fuzzy simulator has generated 16 applications. The total compute requirements of these applications were ranging from 1500 minutes to 2000 minutes. A new simulator has also generated 4 localized CGNs namely CGN1, CGN2 and so on. Each CGN consists of 20 nodes. For satellite-based network, it has generated 10 pairs of spot beams of 50 Mbps, 2 Mbps and 64 Kbps data transfer capacities. CGN1 is taken as the central compute grid where load balancer is running. All processes are residing initially on CGN1. CGN2 could be accessed through 50 Mbps satellite link. CGN3 could be accessed through 2 Mbps satellite link. CGN4 could be accessed through 64 kbps satellite link. At a time, these applications are scheduled on 20 nodes only. These 20 nodes may consist of any combination of nodes from the 4 CGNs. A combination of nodes is denoted by the NodeVector (n1, n2, n3, n4), which shows that n1 nodes from CGN1, n2 nodes from CGN2, n3 nodes from CGN3 and n4 nodes from CGN4 are considered where sum of these is equal to 20. All such permutations of NodeVector (n1, n2, n3, n4) are checked to study the effect of Network transfer delay on session completion time.

7 Results and Discussion

Fig.5 show the session completion time (SCT) statistics for the NodeVector (i, 0, 0, 20-i), where i is number of node on the central Grid CGN1

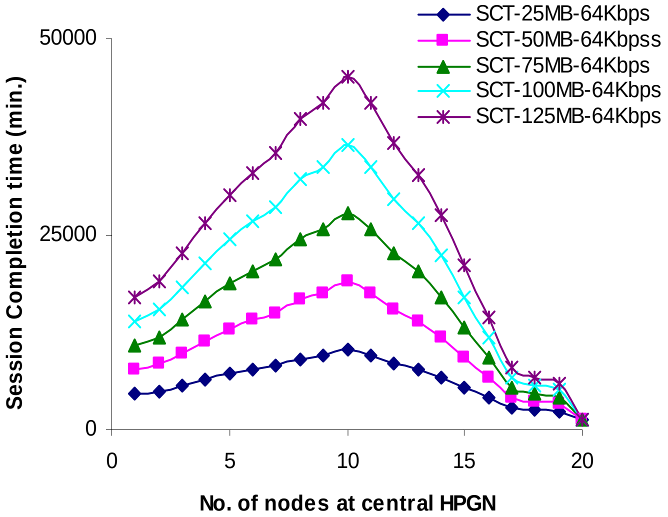

Figure5: Session completion time Vs No. of nodes at central HPGN with data volume to be transferred on remote HPGNs as parameter

considered for allocation at a time. The remaining 20-i nodes are taken from CGN4. This figure shows the effect of Average Data Transfer Required (ADTR) before a process could be allocated on CGN4. The lowest line on this figure shows that SCTs are large for lower value of i as compared to SCTs for when i reaches towards its maximum values 20. The reason for this is that the required number of nodes is less in the CGN1. Hence processes are transferred on the remote CGN4 through 64Kbps Data Transfer Rate (DTR) via satellite link which causes delay in session completion time. Once the initial parent processes of the applications are transferred on the CGN4, most of their dependent would start on CGN4 only without any data transfer delay. For large value of I, most of the processes across all application would be scheduled locally, hence data transfer delay would very less, which causes SCT to be as low as possible for a given resource availability on CGN1. The middle region of lowest lone of the figure gives maximum SCT as compared to start and end portion of the line. This region corresponds to equal resource availability on CGN1 and CGN4. Hence, frequent data transfer would be required between parent and child processes scheduled on different CGNs due to resource scarcity on any individual CGN. The other graph shows the effect of ADTR on the session completion. The ADTR is a fuzzy input parameter to the Rule-based fuzzy simulator, which generates the applications as per this requirement. If ADTR required is doubled, the SCT is also increased. But the increase in SCT is not linear as shown in the figure. It is more for lower values of i as compared to that for higher values of i. The

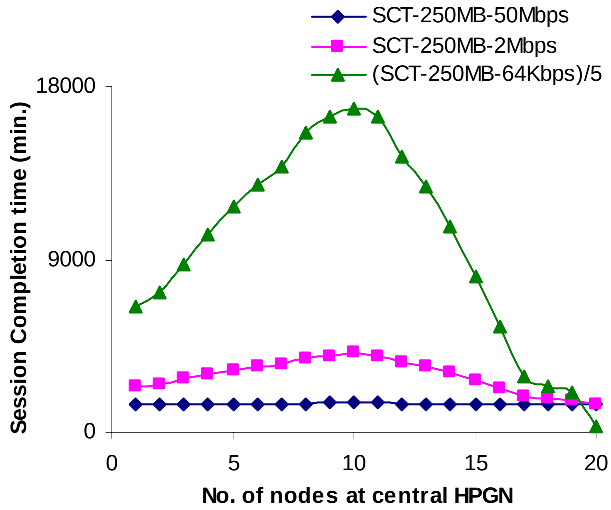

Figure6: Session completion time Vs No. of nodes at central HPGN with satellite link speeds for remote HPGNs as parameter

reason is quite similar to the discussion given above pertaining to lower line of the figure. In the middle range of i, the increase in SCT tend to be linear with increase in ADTR because of the high probability of frequent data transfer between CGN1 and CGN4, due to equal resource distribution among the two locations. All the lines on Fig.5 are converging to the same point for the higher value i. It is due to the amount of work load submitted to the load balancer. In this simulation experiment, the work load is insufficient to keep the 20 nodes busy. So the five cases shown in Figure 5 converge, when the resource abundance situation is reached. In cases of resource scarcity, SCT would not saturate, but it would still follow the pattern given in Fig.5. Thus Fig.5 explains the effect of ADTR on the SCT with the parameter DTR taken as constant.

Fig.6 shows the effect of DTR via satellite link on the SCT with ADTR taken as constant. DTR is the second contributor to the network delay. It shows the SCTs for the NodeVector (i,20-i,0,0), NodeVector (i,0,20-i,0) and NodeVector (i,0,0,20-i).

The lowest line in Fig.5 corresponds to NodeVector (i,20-i,0,0) which signify the resource sharing among the CGN1 and CGN2. This curve is parallel to the x axis, which means that SCT for i nodes on CGN1 and 20-i nodes node on CGN2 is more or less same. It is due to the face that CGN1 and CGN2 are connected with 50 Mbps satellite link. Since ADTR for this figure is taken as 250 MB, it takes less that a minute to transfer this data across the CGNs at 50 Mbps. Since the average process duration of the processes of this simulated workload is of the order of 100 minutes, network delay of less than a minute does not increase the SCT significantly. Hence in this case, it doesn't matter much as to whether process is running locally on CGN1 or on remote CGN2. Middle line of this Fig.6 is corresponding to NodeVector (i,0,20-i,0) which denotes the process transfer between CGN1 and CGN3. Network delay due to DTR between these CGNs is of the order of 10 minutes, which is comparable to the average process duration. Hence it affects the SCT to a great extent. The explanation of shape of this curve is similar to the explanation of the curves of Fig.5. Fundamentally, the factor affecting the shape of SCT curves is network delay. Hence, when the network delay is of the order of average process duration, the SCT curves are going to take inverted V shape, irrespective of the fact whether the network delay is due to ADTR or DTR. The uppermost curve of Fig.6 is scaled down by a factor of 5. It corresponds to the NodeVector (i,0,0,20-i), which signifies data transfer between CGN1 and CGN4. Here network delay due to DTR is more than 10 times larger than average process duration of our simulated workload. Hence it affects SCT excessively.

8 Conclusion and Future Scope

In this paper a modified Application Centric Load Balancing Algorithm for satellite-based grids is proposed to take network delay into consideration. This network delay is caused by two factors, namely volume of data transfer required among the processes of an application and data rate by which these satellite-based grids are connected. The effect of these two parameters on session completion time is explored extensively. It is observed that if network delay is of the order of process duration or more, the session completion curves are going to follow the inverted V shape; else they would be flat, for a given resource availability condition across the satellite-based grids. In future, we will explore the effect of network delay on resource utilization. Till now, we tested our load balancing algorithm on the simulated environment for satellite-based grids. Now we plan to test and implement it on a satellite-based grid environment like Ganesh.

Acknowledgements

Authors acknowledge the guidance, support and encouragement of Mr. P. Dhar, Mr. P.L. Kulkarni and Mr. V.H. Patel for carrying out this work.

References:

[1] Foster, I. and Kesselman, C. (eds.), *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufman, 2003.

[2] Ian Foster, Carl Kesselman, Steven Tuecke, *The Anatomy of The grid: Enabling Scalable Virtual Organizations*, International Journal of Supercomputer Applications, 15(3), 2001.

[3] Sengupta A., Hanjura A.K., Mathur B.S., *Satellite broadcasting of time and frequency signals*, Proceedings of the IEEE, July 1991 Pages: 973-982

[4] Amadesi P., *The role of VSATs and other small satellite terminals in the evolving telecommunications environment*, Satellite communications ICSC 1994, Volume-1, Pages: 59-66.

[5] *Project Report on GSAT-3 (EDUSAT) Applications programme*, SAC & DECU /ISRO, March 2003.

[6] *Project Report on GSAT-4 Applications programme*, SAC & DECU /ISRO, April 2004,

[7] H. S. Bhatt, H. J. Kotecha, B.K. Singh, V. H Patel, K. Bandyopadhyay, A. R. Dasgupta, *Handshaking of Grid technology and Communication satellites*, XIIth International Conference on Advanced Computing & Communication, organised by IEEE and ACS india, pp. 60-70 (Dec.2004).

[8] Frey, J., Tannenbaum, T., Foster, I., Livny, M. and Tuecke, S., *Condor-G: A Computation Management Agent for Multi-Institutional Grids*. In *10th International Symposium on High Performance Distributed Computing*, (2001), IEEE Press, 55-66

[9] Buyya R, Abramson D, and Giddy J, *Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid*, Proc. 4th International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia'2000), Beijing, China. IEEE Computer Society Press, USA, 2000.

[10] OpenPBS http://www.openpbs.org/

[11] Czajkowski, K., Foster, I., Karonis, N., Kesselman, C., Martin, S., Smith, W., Tuecke, S., *A Resource Management Architecture for Metacomputing Systems*, Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing, 1998.

[12] Gregor von Laszewski, Ian Foster, Jarek Gawor, and Peter Lane, *A Java Commodity Grid Kit*, Concurrency and Computation: Practice and Experience, vol. 13, no. 8-9, pp. 643-662, 2001, http:/www.cogkits.org/.

[13] Bhatt, H. et al., *GANESH: Grid Application maNagement and Enhanced ScHeduling*, Proceeding of 12th International conference on Computing and Communications: ADCOM 2004, 15-18 Dec 2004, pp 51-69.

[14] Bhatt H. S., Singh B.K., Aggarwal A.K., *Application centric load balancing based on hybrid model*, International conference for advance computing and communication ADCOM-2001, organised by IEEE and ACS India, Dec. 2001, pp. 231-238.

[15] Haresh S. Bhatt, VH Patel, K Bandyopadhyay, Arup Dasgupta, Keyur Patel, *Extending Wireless Grids to Remote Locations*, Eigth International Symposium on Wireless Personal Multimedia Communications, 18-22, Sep 2005

[16] Sastri kota, Mario marchese, *Quality of service for satellite IP networks: a survey*, International journal of satellite communication and networking, Volume-21, number 4-5, July-October 2003, Pages: 303-349

[17] B. K. Singh, H. S. Bhatt, A. K. Aggarwal, *Parametric optimization of load balancing algorithm*, XIIth International Conference on Advanced Computing & Communication, organised by IEEE and ACS india, pp. 519-526 (Dec.2004).

[18] A. K. Aggarwal,H. S. Bhatt, B. K. Singh, *Optimization of load balancing algorithm using Workload Simulator for a HPC environment*, Proceedings of HPCS-2003, Sherbrooke, pp. 231-238 (MAY 2003).

[19] Bhatt H. S., Singh B.K., Aggarwal A.K., *Rule-based Fuzzy Simulator for testing Load Balancing Algorithms*, International Journal of Distributed and Ubiquitous Computing, April 2003, pp. 16-27.