International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 3
ISSN: 2454-4248
37 –40

Secured and Adaptive Load Balancing with Backup Approach for Computational Grids

Dr. B.Jayanthi, Mr.S.Vijayakumar
Department of Computer Science(P.G.)
Kongu Arts and Science College(Autonomous)
Erode, Tamilnadu, India
sjaihere@gmail.com,sakthiveluvijayakumar@gmail.com

Mr. M. Chandru
Department of Computer Science
Kongu Arts and Science College(Autonomous)
Erode, Tamilnadu, India
emchandru@gmail.com

***Abstract*—**Load Balancing is one of the big issues in Grid Computing.This work aims to develop a secured load balancing algorithm which reduces the download time, network overhead and improve the packet delivery ratio of the resources. This work enhances the PWSLB algorithm for load balancing, fault tolerant scheduling and security. The experimental results show an average of 0.2 to 8 % increase in Packet delivery Ratio and 0.080 to 0.1 % of network overhead reduction at 0.1324 milliseconds reduction in Download time. Finally this work Reduces, the download time, network overhead of tasks and also increases the packet delivery ratio

Keywords- *Grid computing, Load balancing, Fault tolerant scheduling, security*

*****

I. INTRODUCTION

Grid computing is a collection of computer resources from multiple locations to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. Grid computing has an emerged as the next generation of parallel and distributed computing methodology that aggregates dispersed heterogeneous resources for solving various kinds of large scale applications in science, engineering and commerce [9]. Load Balancing is one of the big issues in Grid Computing [1]. Load balancing Algorithm types and three policies are Information policy, Triggering Policy, and Selection Policy in Grid Environment are discussed in [10], [6]. In general load balancing algorithms can be classified as centralized or decentralized, and static or dynamic. In the centralized approach one node in the system acts as a scheduler and makes all the load balancing decisions. Information is sent from the other nodes to this node. In the decentralized approach [8], all nodes in the system are involved in the load balancing decisions. Many fault-tolerant schemes have been proposed for grid systems [2], [3], and [7]. Backup overloading to reduce replication cost of independent jobs introduced in [5]. SHA-3 preserves the online nature of SHA-2. That is, the algorithm process comparatively small blocks (512 or 1024) at a time instead of requiring the entire message to be buffered in memory before processing it [4].

II. *MATERIALS AND METHODS*

A. *Piggybacking*

In this study, piggybacking technique is introduced for load balancing. Each resource maintains the load information of other resources by using the state object. The state object helps a resource to estimate the load and efficiency of other resources at any time without message transfer. Each item in state object of neighbor or partner resource has a property list such as load, efficiency, time. Load denotes the load information of neighbor or partner resource, efficiency denotes the efficiency value of neighbor or partner resource, time denotes the neighbor's or partner's local time. When the load information or efficiency value is reported, each resource collects and maintains the load information of only its neighbor's and partner's. In order to minimize the overhead of information collection, load information exchange is done by piggybacking. Specifically, when resource transfers a packet to neighbor or partner resource for processing, resource appends the load information and efficiency values of itself, its neighbors, its partners to the packer and sent to neighbor or partner resource by piggybacking. Neighbor or partner resource updates the corresponding load information and efficiency values of its state object by comparing the timestamps if the resource contained in the packet belongs to its neighbors or partners. Similarly, neighbor or partner resource inserts the current load information and efficiency values of itself, its neighbors and its partners in the acknowledgement to resource. So resource can update its state objects. An advantage of piggybacking strategy reduces the message overhead and can takes small amount of network bandwidth. In this way, the load information packet should be simple and small sized as possible.

B. *Boundary Schedules*

Fault tolerant scheduling is an imperative step for large scale computational grid systems, as often geographically distributed nodes co-operate to execute a job [3]. Primary-backup approach is a commonly used for fault tolerance wherein each packet has a primary copy and backup copy on

 ISSN: 2454-4248
37 –40

two different processors. This method is used to find the scheduling time of the backup copy. Scheduling the backup of job with its start time and/or finish time collide with boundaries of the interval or boundaries of over loadable backup schedules is referred to as a boundary schedule. A schedule is eligible if it is within the time and does not overlap with any primary schedule or non over loadable backup schedule as shown in Fig.1. The pseudocode for boundary schedule is,

Boundary Schedule ($t_s(j_i)$)
(1) If schedule eligibility($t_s(j_i)$)=true
(2) Then
(3) Cost <——— replication cost($t_s(j_i)$)
(4) EndIf
(5) If cost is less than $R^R(j_i)$ or they are equal and ($t_s(j_i)$) +($t_e(j_i)$) is Less than $t^{Bf}(j_i)$ then
(6) $C_B(j_i)$ <————c_i
(7) $R^R(j_i)$ <————cost
(8) $t^{Bf}(j_i)$ <————($t_s(j_i)$) +($t_e(j_i)$)
(9) EndIf

Fig. 1 Pseudocode for boundary schedule

C. *Hash Encryption*

Hash algorithms are used to map binary values of an arbitrary length to small binary values of a fixed length, known as hash values. A hash value is a numerical representation of data. Sender would write a message, and then create a hash of that message by using the selected algorithm. If the hashes match, Receiver knows two things: 1.The message was not altered. 2. The sender of the message is authentic. This method prevents message tampering by preventing anyone from modifying the hash value. Although the message and its hash can be read by anyone, the hash value can be changed only by Sender. An attacker who wants to impersonate Sender would require access to Sender's Web site. None of the previous methods will prevent someone from reading Sender's messages, because they are transmitted in plaintext. Full security typically requires digital signatures (message signing) and encryption.

D. *PWSLB algorithms analysis*

The grid scheduler selects a neighboring resource for processing from the state object. This is done by obtain the corresponding transfer delay. The efficiency value of the processor and load of the processor and completion time of the processor are selects from the state object list. Then the grid scheduler updates the nearest resource with effective and sent to the nearest resource by piggybacking load information. Primary and backup approach is used for distributed fault tolerance. Backup copy of the resource is activated when the primary copy of the resource failure. Scheduling the backup copy is calculated using the boundary schedules. The message is padding and hashing using SHA-3 algorithm [4]. Then concatenate the message and forward the message.

III. Experimental Results

This research is implemented in the NS2 simulator. This research focused results relating to objective metrics, according to various numbers of tasks and nodes. The result of this research is analyzed in graph. The graph, that provides the comparison of AOMDV, PDLB and proposed PWSLB. Packet delivery ratio analysis in fig.2 and that value of comparison in table 1, download time analysis in fig.3 and that value in table 2 and network overhead analysis in fig.4 and that comparison values in table 3.

A. *Packet Delivery Ratio analysis (No Units)*

This metric gives us an idea of how well the PWSLB is performing in terms of packet delivery at different speeds using different traffic models. Mathematically, PDR can define as,

$$Packet\ delivery\ ratio = \frac{Packets\ received\ by\ all\ sinks}{Packets\ sent\ by\ all\ sources}$$

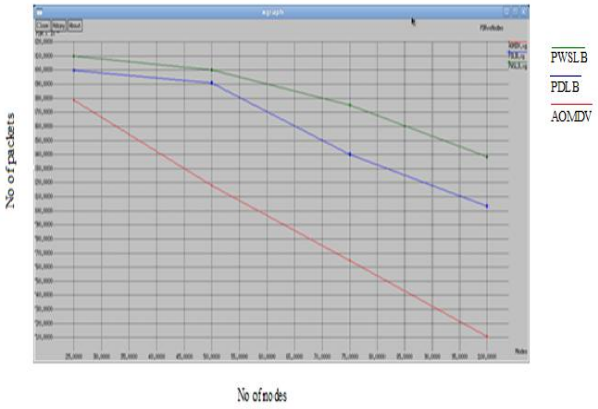

Fig. 2 AOMDV Vs PDLB Vs PWSLB

No of Nodes	AOMDV (%)	PDLB (%)	PWSLB (%)
25	0.87862	0.89992	0.91012
50	0.81791	0.89105	0.90012
75	0.76454	0.84022	0.87511
100	0.71034	0.80339	0.83823

Table 1: Comparison of AOMDV, PDLB, PWSLB for PDR

 ISSN: 2454-4248
37 –40

B. Download Time analysis (micro seconds)

The average time it takes a data packet to reach the destination. This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue. This metric is calculated by subtracting time at which first packet was transmitted by source from time at which first data packet arrived to destination. Mathematically, it can be defined as:

$$\text{AVG. DOWNLOAD TIME} = \frac{\text{Sum of the time spent to deliver packets}}{\text{Number of packets received by destination}}$$

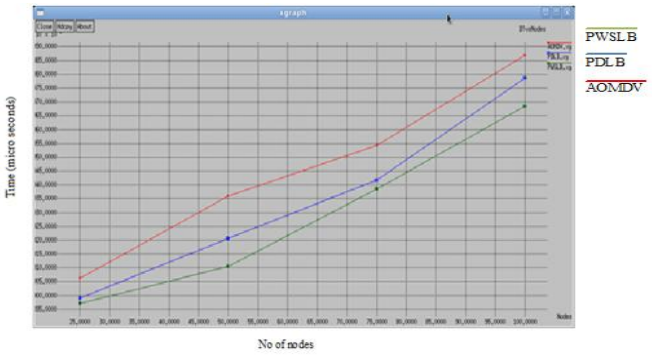

Fig. 3 AOMDV Vs PDLB Vs PWSLB

Table 2: Comparison of AOMDV, PDLB, PWSLB for PDR

No of Nodes	AOMDV (mic sec)	PDLB (mic sec)	PWSLB (mic sec)
25	0.9062	0.899068	0.897071
50	0.9360	0.920664	0.910561
75	0.9542	0.941723	0.93860
100	0.9869	0.978700	0.968561

C. Network overhead analysis (No Units)

Network overhead is defined as amount of non data packets sent to maintain the grid setup and perform load balancing tasks. Network overhead is also called a control overhead.

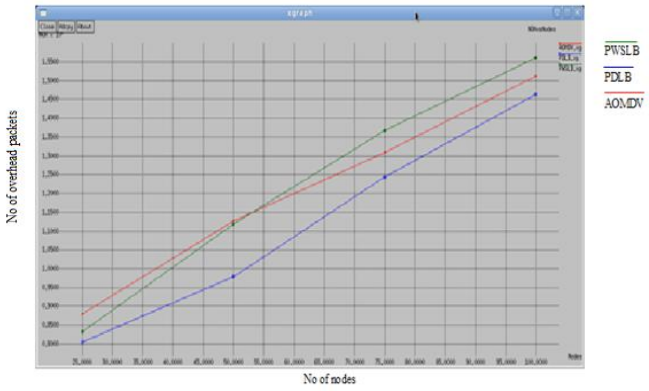

Fig. 4 AOMDV Vs PDLB Vs PWSLB

No of Nodes	AOMDV	PDLB	PWSLB
25	880	805	832
50	1127	978	1118
75	1309	1244	1368
100	1513	1464	1561

Table 3: Comparison of AOMDV, PDLB, PWSLB for NOH

IV. DISCUSSION

In Fig.2, AOMDV is Adhoc On demand Multipath Distance Vector routing without secured load balancing and fault tolerance. PDLB is the Performance Driven Load balancing without security. PWSLB is a performance weight based secured load balancing. The average packet delivery ratio curves of different approaches show that, only PWSLB achieve fairness during different node scales. The AOMDV method works on without load balancing and fault tolerance. So the PDR is lower than the PDLB and PWSLB. The PDLB can balance the load. However, its ratio is lower than the proposed approach and may result in over utilization or under utilization of some resources. This is because, PDLB have no fault tolerance and security. On the other hand, the proposed approach PWSLB is the fault tolerance and security so the packets sent to the processor securely and tolerating the faults. So the packet delivery ratio is improved by comparing others as shown in table 1. In Fig.3 AOMDV consistently shows the highest download time because it works without load balancing. So the download time of the packets is higher. PDLB has the lowest download time compared to the AOMDV. But the enhanced PWSLB has the lowest download time compared to the others as shown in table 2. In Fig.4 Network overhead of PDLB is reduced compared to the AOMDV in 25 nodes. Because AOMDV is focused on without load balancing so overhead is increased. But PDLB is focused load balancing but no security so less routing packets compared to the PWSLB. But in 75 and 100 numbers of nodes PWSLB overhead is increased compared to the AOMDV and PDLB. So in PWSLB, the network overhead in increased when the number nodes increased as shown in table 3.

V. CONCLUSION

The PWSLB algorithm is developed to address the following objectives, 1.Reducing, whenever possible, the download time, network overhead of tasks submitted to the processor. 2. Increasing the packet delivery ratio. 3. Respecting the constraints of security for packets.

 ISSN: 2454-4248
37 –40

REFERENCES

[1] Daphne Lopez, S. V. Kasmir raja," A Dynamic Error Based Fair Scheduling Algorithm for a Computational Grid", Journal of Theoretical and Applied Information Technology - 2009 JATIT.

[2] G. Manimaran and C.S.R. Murthy, "A Fault-Tolerant Dynamic Scheduling Algorithm for Multiprocessor Real-Time Systems and Its Analysis," IEEE Trans. Parallel and Distributed Systems, vol. 9, no. 11, pp. 1137-1152, Nov. 1998.

[3] R.A. Omari, A.K. Somani, and G. Maninaran, "A New Fault-Tolerant Technique for Improving Schedulability in Multiprocessor Real-Time Systems," Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS), 2001.

[4] William Stallings, "cryptography and network security: principles and practice", fifth edition, published by Pearson education,inc, publishing as prentice hall, 2011.

[5] Geoffrey Falzon, M. L. (2010). "Enhancing list scheduling heuristics for dependent job scheduling in grid computing environments" J Supercomput.

[6] S. Ghosh, R. Melhem, and D. Mosse, "Fault-Tolerance through Scheduling of Aperiodic Tasks in Hard Real-Time Multiprocessor Systems," IEEE Trans. Parallel and Distributed Systems, vol. 8, no. 3, pp. 272-284, Mar. 1997.

[7] J.H. Abawajy, "Fault-Tolerant Scheduling Policy for Grid Computing Systems," Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS), 2004.

[8] Grosu, D., Chronopoulos, A.T." Noncooperative load balancing in distributed systems", Journal of Parallel Distrib. Comput. 65(9), 1022–1034 (2005).

[9] M.Kamarunisha, S.Ranichandra, T.K.P.Rajagopal," Recitation of Load Balancing Algorithms In Grid Computing Environment Using Policies And Strategies An Approach," International Journal of Scientific & Engineering Research Volume 2, Issue 3, March-2011

[10] Bora Uar, Cevdet Aykanat, K. K. M. I. (2006). Task assignment in heterogeneous computing systems. J. Parallel Distrib. Comput., 66, 32–46.