ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Fuzzy Logic Based Job Scheduling In Computational Grid with Minimum Communication and Replication Cost

Deepa N K[1], L M Nithya[2]

PG scholar, Department of Information Technology, SNS College of Technology, Coimbatore-35, Tamilnadu, India [1]

HOD, Department of Information Technology, SNS College of Technology, Coimbatore-35, Tamilnadu, India [2]

Abstract- Grid computing allows sharing and coordinated use of diverse resources in dynamic distributed virtual organizations. The objective is to best scheduling of tasks which attain maximum resource utilization, minimum response time and a well-balanced load across all the resources involved in a grid. Fault-tolerant scheduling is an imperative step for large-scale computational Grid systems which uses primary-backup approach as the methodology for fault tolerance. For independent tasks, the backup overloading is using for reducing the cost. The heterogeneous nature of the resources and their deferring security policies are complicated and complex in the grid environment. The main challenges it faces are integration, interoperability of security framework and the trust relationship between participants. The proposed method considers the trust relationship between the participants. It offers a fuzzy-logic-based self-adaptive job replication scheduling (FSARS) algorithm. FSARS uses the security demand of the task and Trust level of the resources as the main parameters. An empirical membership functions are using for transformation of security conditions in to the fuzzy set. The proposed approach gives a robust performance against resource failures and increased scheduling success rate.

Index Terms- Grid Computing, Fault Tolerant Scheduling, Security, Fuzzy logic

I. INTRODUCTION

Grid computing is the collection of computer resources from multiple locations to reach a common goal. A distributed system with non-interactive workloads that involve a large number of files is considered as a grid. Grid computing differentiate from conventional high performance computing systems such as cluster computing is that grids tend to be more heterogeneous ,geographically dispersed and loosely coupled [1]. A single grid can be dedicated to a particular application, or for a variety of purposes. Grids are often built with general-purpose grid middleware software libraries. Computational Grid architectures that focus on setting aside resources specifically for computing power; i.e. solving mathematical problems and complex equations. The machines participating in this type of grid are usually high-performance servers [2]. The most common vulnerabilities in this type of grid are the programs with infinite loops can be used to bring down nodes of this grid, decreasing functionality.

Load balancing performs an important role in grid computing. It must be integrated in to the grid system in order to avoid the delay in processing and resource over commitments .load balancing should be effective to reduce the difference between the heaviest load and lightest load. Load balancing is categorised in to centralized, decentralized and hierarchical. The decentralized approach of grids, the flexibility and the heterogeneous nature of their infrastructure, and the aim of being a general purpose system are a challenge to grid middleware's to provide a manageable distributed, secure, stable, and high quality-of-service system to its users[3]. Distributing workloads among multiple resources needs information exchange .The information exchange can be done by two methods: neighbour based and cluster based. In neighbour based method the information is transferring to the physical or

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

logical neighbours of the computing node. The load balancing algorithm in which computing nodes are partitioned in to different clusters and information exchange is takes place inside the clusters.

Scheduling is a popular method to implement fault tolerant in grid environment. It is done by allocating multiple tasks to different resources called replication. Replication can be active or passive. In active replication multiple copies of the task are distributed over multiple resources and they are executing parallel to achieve fault tolerance. Passive replication includes the primary and backup method. The primary and backup nodes contain the copies of the task. The backup copy is activated only if the primary node fails.

In a grid environment security schemes using are very complex and complicated due to the heterogeneous nature resources and their differing security policies. The computing resources are placed in different security domains and heterogeneous platforms. Security requirements are concentrated on data integrity, confidentiality and information pricing [4]. Resource sharing among heterogeneous virtual organization participants is very complex process due to many challenges. The challenges are integration challenge, interoperability challenge and trust relationship challenge.

Numerous security framework and standards are available today. The individuals and organizations prefer the one which is most suitable for their environment. Security frameworks are not replicable and can't make alternate. So it will become a big challenge for the participants to integrate them. Resource sharing is one of the main features of grid computing. But these resources sharing will leads to many security problems by their respective needs of security interoperability at each service layer implementation. Trust among the participants included in the grid environment is very difficult to achieve.

Fuzzy set theory is the way to represent the real world knowledge in the face of uncertainty. Fuzzy set can be defined mathematically by assigning a grade of membership in the fuzzy set. In fuzzy different degree of membership are allowed, between 0 and 1. A membership function maps the elements of universe in to the value between 0 and 1.In many situations, it's very easy to take a decision if the output of the system is fuzzy which a crisp value is. The conversion of the fuzzy value in to a crisp value is called defuzzification process [5].

In this paper the fault tolerant scheduling is integrated in to the fuzzy logic to establish a trust relationship between the resources and tasks. The rest of this paper include the system model, fault tolerant algorithm using, logic behind to integrating the fuzzy logic to the grid environment for each task, the backup is scheduled after its primary.

II. RELATED WORKS

In grid computing, load balancing, scheduling and fault tolerance are active areas of research in grid environments. For computational grids Mura et al.and Elsasser et al. [6] introduced a load-balancing schemes. But they omitted the overhead involved for collecting the state information for load balancing. Nandagopal et al. [7] proposed a decentralized dynamic sender-initiated load-balancing scheme for multicluster computational grid environment (SI-DDLB). SI-DDLB is an extended study of the approach proposed in [10]. But SI-DDLB did not consider the execution scheme for data distribution and could not model the impact of accuracy of job execution time estimation. For grid systems many fault-tolerant schemes have been proposed [10], and [11]. Ghosh et al.] Introduced backup overloading to reduce replication cost of independent jobs. It could not guarantee to find an optimal schedule for backups of independent jobs in terms of replication cost. Luo et al. [12] and Zheng et alproposed a dynamic and reliability-driven real-time fault tolerant scheduling algorithm. Zhu et al. [19] proposed QoS-aware fault-tolerant scheduling algorithm for heterogeneous clusters.

III. SYSTEM MODEL

3.1 Task Model

In this experiments a set of computing nodes are considering. We are considering that each computing node have single or multiple processors. For our experiment we are considering n number of tasks, resources and schedulers. Each scheduler is connected to the tasks and processors through a network. The task node generates the task which is to be executed by the processor node. Tasks generated by the task node or user node are send to the

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

scheduler for effectively allocating them to the processor. Scheduler receives tasks for allocating to the processors[7]. Here we are using the fuzzy logic. Scheduler considers the trust factors of the task and resources for scheduling. Processors execute the tasks which are schedule to it.

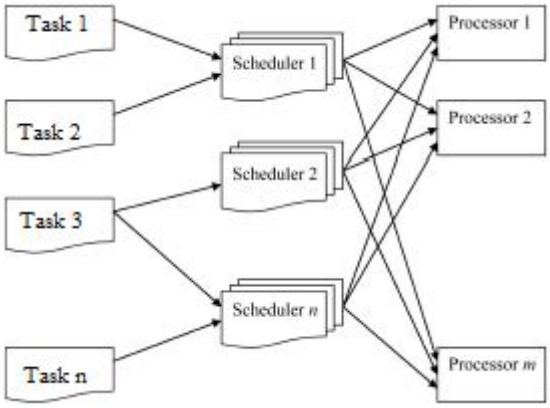

Fig 1. Participants in Task Model

3.2 Fault Model

Tasks will fail when the hardware faults occurs in the processor where they are located fails .The faults can be transient or permanent and are assumed to be independent[8]. The primary and back up methods are using for fault tolerant scheduling. Tasks are first allocating to the primary. For each task, the backup is scheduled after its primary. When a primary fails the backup activated. However, it is not necessary that backups of all tasks in a job must be scheduled after primaries of all tasks in a job. A fault detection mechanism called fail-signal and acceptance test to detect processor and task failures. At most, one version of a job is expected to encounter a fault. If the primary of a job fails, its backup always succeeds[9]. This condition is guaranteed by the assumption that the minimum required value of mean time of failure (MTTF) is always greater than or equal to the maximum job execution time in a primary backup approach[10].

Two techniques are using for task scheduling in primary and backup method: Backup overloading including scheduling backups of multiple primary at the same time. It efficiently utilise the processor time[11]. When the primary executed successfully backup deallocated.

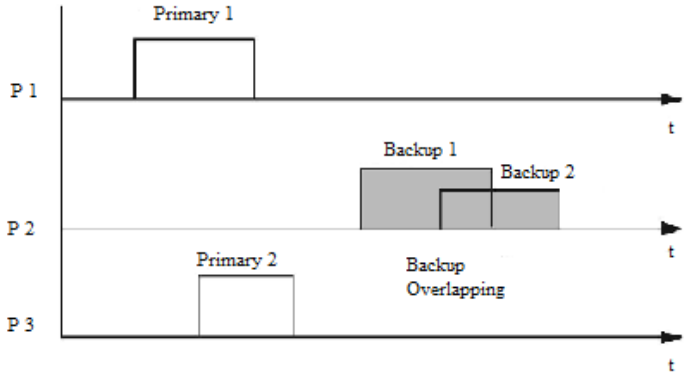

Fig 2. Backup Overlapping Between Tasks

3.3 Load Balancing Model

Each resource c_i maintains p number of neighbouring resources $LNSet_i$, which the grid scheduler will use to select a neighboring resource for offloading jobs. Neighbors for each resource are formed in terms of transfer delay[20]. For c_i, c_k is considered as its neighbor resource as long as the transfer delay between c_i and c_k is within α time of the transfer delay between c_i and the nearest resource and load of c_k is less than c_i. For each resource, other neighbor resources are sort by transfer delay in ascending order. After this process, the first-ranked resource is chosen as the nearest resource

$$\alpha = \frac{TD_{ik}}{TD_{inearest}} \quad (1)$$

Where TD_{ik} denotes the transfer delay from resource c_i to c_k. $TD_{inearest}$ denotes the transfer delay from nearest resource of resource c_i to itself.

3.3.1 Resource Efficiency Estimation Policy

Each grid scheduler creates a score according to the jobs' states in the Submitted Job List, namely

$$s(c_q) = \begin{Bmatrix} -1 & job\ is\ finished \\ 0 & no\ job \\ +1 & job\ is\ unfinished \end{Bmatrix} \quad (2)$$

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

If multiple jobs are submitted to the same resource, there will be an independent score for each job. Finally, the health of the resource can be calculated by adding up all the scores.

$$h(c_q) = \sum_{i=1}^{k(q)} s(i) \quad (3)$$

Where k(q) denotes the number of jobs being submitted to the resource c_q currently. The efficiency for each resource can be updated as

$$O(k).efficiency = O(k).efficiency + h(cq) \quad (4)$$

$O^i(k).efficiency$ is the efficiency of resource c_k maintained by grid scheduler g_i on resource c_i.

3.3.2 Performance Benefit Factor

We include dynamic communication cost in the cost calculation as follows:

$\forall c_i \in C$, the communication cost of sending a job $j_x \in J$ from c_i to c_k at time instant t is estimated by c_i as

$$cc(j_x, c_i, c_k, t) = TRANS\_IN(j_x, c_i, c_k, t) + LD_k + O^i(k).efficiency + TRANS\_OUT(j_x, c_k, c_i, t) \quad (5)$$

3.3.3 Instantaneous Distribution Policy

When a new job arrives at a resource c_i, the policy decides whether the job has to be executed locally or sent to other resources in $LPSet_i$ $LNSet_i$. The decision depends on whether it can gain performance benefit if it is distributed to a resource in $LPSet_i$ or $LNSet_i$.

3.3.4 Load Adjustment Policy

The load adjustment policy for a resource ci tries to continuously reduce load difference among ci and its neighboring resources in LNSeti in which lightly loaded resources are moving jobs from heavily loaded resources. The Load adjustment policy is triggered whenever ci receives load information from its neighboring resources. To initiate a job migration the load balancing algorithm uses the most recent efficiency value and load status information. We are considering that all resources are exchanging their loads in parallel and the dynamic nature of the environment; it will make the network reaches to the local optimum very rapidly. According to the minimum load among all nodes, each resource submits the jobs to one of its neighbor resources. No job is submitted from the current resource if all its neighbors are busier than the resource itself,. If there are any jobs in job queue waiting to be executed, the resource attempt to submit them to a minimum loaded neighbor resource and therefore achieves a fair load distribution among resources.

3.3.5 Enhanced MIF policy

An enhanced mutual information feedback (EMIF) is proposed by adding resource efficiency to the existing mutual information feedback [21] mechanism and making the job request small sized and simple. To make the job transfer request simple and small sized while transferring job from ci to ck, instead of the job transfer request have the job to ck, it carry the source resource information including a unique job ID of the source resource to which the job was transferred. So when the job transfer request reaches its destination ck, it can be downloaded from the source as needed. By using the state object Oi each resource ci maintains the state information of other resources. The the load and efficiency of other resources can be determined by the state object at any time without any message transfer. Each item in Oi[k] maintains a property list which includes load, efficiency, and time. Oi[k] load represents the load information of resource ck, Oi[k] efficiency represents the efficiency value of resource ck, and Oi[k] time represents ck's local time when the efficiency value and load status is reported. Only neighbours and partners state information is collected and maintained by each resource.

3.4 Fuzzy Logic

3.4.1 Fuzzy parameter calculation

This module applies the concept of fuzzy set to the process from grid security conditions to replication number of each job. It offers a FSARS algorithm [22]. This method first assigns a SD to a user job when user submitted it. The trust model assesses the resource site's trustworthiness, namely, the TL. TL quantifies how much a user can trust a site for successfully executing a given job. Only the job can be

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

successfully finished when SD and TL satisfy a security assurance condition (SDİTL) when scheduling the jobs. The binding is achieved by periodic exchange of site security information and matchmaking to satisfy user job demands. The security demand of the task set is calculating by using the equation:

$$\overline{SD} = \frac{\sum_{i=1}^{n}(SD_i \times \sum_{j=1}^{q_i}\frac{e_{ij}}{q_i})}{\sum_{i=1}^{n}\sum_{j=1}^{q_i}\frac{e_{ij}}{q_i}} \quad (6)$$

SD is the security demand of each task and e is the expected execution time. q represents the hosts satisfies the security assurance condition. The trust level of the grid environment is calculating using the equation

$$\overline{TL} = \frac{\sum_{j=1}^{m} TL_j \times p_j}{\sum_{j=1}^{m} p_j} \quad (7)$$

TL is trust level of each resources and P represents the speed. Another parameter is security error ratio. It is calculating by using the equation.

$$SE_i = \frac{\overline{TL} - SD_i}{SD_i} \quad (8)$$

3.4.2 Fuzzy inference process

In this module membership functions are using for mapping the parameters in to the interval [0,1].Using the membership functions five levels of SD,SD bar,SE and K is obtaining. The five levels are very low, low, medium, high and very high. This will create the membership degrees for SD, SE and TL bar. The fuzzy inference rules are using to improve the performance of the grid environment.

3.4.3 Defuzzification process

In defuzzification process the derivation of replication number each task is doing. The conversion of a fuzzy set to a single crisp value is called defuzzification. It is the opposite process to fuzzification.

IV CONCLUSION

In Grid environment security is the major issue. In this paper we establish a trust relationship between the grid participants to ensure security. A fixed number of replicated copies of the tasks in fault tolerant scheduling utilize excessive hosts or resources. The proposed approach gives a robust performance against resource failures and increased scheduling success rate.

REFERENCES

[1] The Grid: Blueprint for a Future Computing Infrastructure, I. Foster, C. Kesselman, eds., second ed. Morgan Kaufmann, 2004.
[2] Zohre Zare, "Security in grid computing", 5th SASTech, Khavaran Higher-education Institute, Mashhad, Iran. May 12-14, 2011.
[3] P.K Suri and M. Singh, "An Efficient Decentralized Load Balancing Algorithm for Grid," Proc. IEEE Int'l Advance Computing Conf., pp. 10-13, 2010.
[4] Joshy Joseph and Craig Fellenstein, "Grid Computing".
[5] S. Rajasekaran,G.A.Vijayalekshmi Pai,"Neural Networks,Fuzzy Logic And Genetic Algorithms".
[6] R. Elsasser, B. Monien, and R. Preis, "Diffusion Schemes for Load Systems, Balancing on Heterogeneous Networks," Theory of Computing vol. 35, no. 3, pp. 305-320, 2002.
[7] N. Malarvizhi and R.V. Uthariaraj, "Hierarchical Status Information Exchange Scheduling and Load Balancing for ComputationaExchange Scheduling and Load Balancing for Computational Grid Environments," Int'l J. Computer Science and Network Security vol. 10, no. 2, pp. 177-185, 2011.
[8] Q. Zheng, C.-K. Tham, and B. Veeravalli, "Dynamic Load Balancing and Pricing in Grid Computing with CommunicationDelay," J. Grid Computing, vol. 6, pp. 239-253, 2008.
[9] S. Ghosh, R. Melhem, and D. Mosse, 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.
[10] Qin Zheng, Bharadwaj Veeravalli and Chen-Khong Tham ,"On the Design of Fault-Tolerant Scheduling Strategies Using Primary-Backup Approach for Computational Grids with Low Replication Costs". IEEE trans, computers, vol. 58, no. 3, march 2009
[11] Ching-Chih Han, Member, IEEE, Kang G. Shin, Fellow, IEEE, and Jian Wu, student Member, IEEE, "A Fault-Tolerant Scheduling Algorithm for Real-Time Periodic Tasks with Possible Software Faults" IEEE transactions on computers, vol. 52, no. 3, march 2003
[12] S. Dhakal, M.M. Hayat, J.E. Pezoa, C. Yang, and D.A. Bader,"Dynamic Load Balancing in Distributed Systems in the Presenceof Delays: A Regeneration Theory Approach," IEEE Trans. ParallelDistributed Systems, vol. 18, no. 4, pp. 485-497, Apr. 2007..

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

[13] K.-Q. Yan, S.-S. Wang, S.-C. Wang, and C.-P. Chang, "Towards a Hybrid Load Balancing Policy in Grid Computing System," ExpertSystems with Applications, vol. 36, pp. 12054-12064, 2009.

[14] P.K Suri and M. Singh, "An Efficient Decentralized LoadBalancing Algorithm for Grid," Proc. IEEE Int'l Advance ComputingConf., pp. 10-13, 2010.

[15] K. Li, "Optimal Load Distribution in Non Dedicated Heterogeneous Cluster and Grid Computing Environments," J. SystemsArchitecture: The EUROMICRO J., vol. 54, no. 2, pp. 11-123, 2008.

[16] K. Qureshi, A. Rehman, and P. Manuel, "Enhanced GridSim Architecture with Load Balancing," The J. Supercomputing, vol. 57,pp. 265-275, 2010.

[17] R. Subrata, A.Y. Zomaya, and B. Landfeldt, "Game-TheoreticApproach for Load Balancing in Computational Grids," IEEETrans. Parallel and Distributed Systems, vol. 19, no. 1, pp. 66-76, Jan.2008.

[18] Abbas Karimi, Faraneh Zarafshan , Adznan b. Jantan, A.R. Ramli, M. Iqbal b.Saripan, " A New Fuzzy Approach for Dynamic Load Balancing Algorithm", (IJCSIS) International Journal of Computer Science and Information Security,Vol. 6, No. 1, 2009

[19] Xiaomin Zhu, Member, IEEE,XiaoQin, Senior Member, IEEE, and Meikang Qiu, Senior Member, IEEE , "QoS-Aware Fault-Tolerant Scheduling for Real-Time Tasks on Heterogeneous Clusters" IEEE Trans.Computers, vol. 60, no. 6, pp. 800-813, June 2011

[20] Jasma Balasangameshwara , Nedunchezhian Raju, "Performance-Driven Load Balancing with a Primary-Backup Approach for Computational Grids with Low Communication Cost and Replication Cost". IEEE Transactions On Computers, VOL. 62, NO. 5, MAY 2013.

[21] K. Lu, R. Subrata, and A.Y. Zomaya, "On the Performance Driven Load Distribution for Heterogeneous Computational Grids," J. Computer and System Science, vol. 73, no. 8, pp. 1191- 1206, 2007.

[22] WANG Cheng, JIANG Congfeng , LIU Xiaohu ,"Fuzzy Logic-Based Secure and Fault Tolerant Job Scheduling in Grid" ISSN 1007-0214 08/49 pp45-50 Volume 12, Number S1, July 2007

[23] Shanshan Song, Kai Hwang, Fellow, IEEE, and Yu-Kwong Kwok, Senior Member, IEEE, "Risk-Resilient Heuristics and Genetic Algorithms for Security-Assured Grid Job Scheduling" , computers, vol. 55, no. 6, june 2006