Australian Journal of Basic and Applied Sciences, 9(5) March 2015, Pages: 13-20

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

An Efficient Globus-QoS-driven Job Scheduling Approach in Grid Environment

[1] Kannan K.S. and [2]Dr. Saravanan R.

[1] NPR College of Engineering and Technology, Department of Computer Science and Engineering, Kannan .K.S, Box.3030. Dindigul, Tamil Nadu, India.
[2] RVS Educational's Trusts Group of Institutions, Computer Science and Engineering, Dr.Saravanan .R, Dindigul, TamilNadu, India.

ARTICLE INFO

Article history:
Received 12 November 2014
Received in revised form 26 December 2014
Accepted 29 January 2015
Available online 10 February 2015

Keywords:
CPU usage, Failure rate, Global Information System (GIS), Grid Computing, Job Scheduling, Memory usage, Minimum Cost Match Schedule (MCMS). Resource and Quality of Service (QoS).

ABSTRACT

Background: An Efficient Globus-QOS-driven Job Scheduling Approach in Grid Environment**Objective:**Grid Computing permits resource sharing from heterogeneous and distributed locations. Grid Computing has an extensive variety of application areas comprising medical, scientific and research areas. But there are also some encounters which rises in the grid computing environment. One of the main experiments in grid computing is Resource or Job scheduling in the grid. Job Scheduling is selecting the most appropriate resource for a job to complete its implementation either in terms of waiting time, turnaround time or cost. In this paper, an efficient job scheduling approach is proposed based on the Quality of Service (QoS) factor. The QoS factors like CPU usage, Memory usage, Failure rate and Network speed are estimated to schedule the jobs to appropriate resources. The Global Information System (GIS) maintains the information about individual user certificates, job details, resource history and QoS measures. The user jobs are submitted into the job pool, the jobs are dispatched with the help of GIS. The implementation shows that the proposed model can perform better than the existing Minimum Cost Match Schedule (MCMS) in terms of average response time, execution time and service reliability.**Results:**The experimental evaluation of the proposed approach is conducted to evaluate the performance of the system. Globus toolkit 4.0.5 is used to create the grid environment. The toolkit supports the system to schedule the jobs to appropriate resources. **Conclusion:** In this paper, an efficient Globus-QoS driven job scheduling approach for grid environment is presented. The major contributions of this approach are to scrutinize with the QoS factors. The QoS measures like CPU usage, Memory usage, Failure rate and Network speed are evaluated and monitored for scheduling the jobs into an appropriate resource. GIS helps to maintain the user job history, user details, certificates and job details. The jobs are scheduled to the suitable resources based on the GIS reports. The grid environment is constructed by the Globus toolkit. The experimental result shows that the proposed model can schedule the jobs within a short period of time and they also the resource request can be granted within a short period of time. The service reliability also comparatively better than the existing Minimum Cost Match Schedule (MCMS). In future, the Geospatial Web Provisioning Service is incorporated with this proposed framework to enhance the query processing in grid environment.

To Cite This Article: Kannan K.S. and Dr. Saravanan R., An Efficient Globus-QoS-driven Job Scheduling Approach in Grid Environment. ***Aust. J. Basic & Appl. Sci.,*** *9(5): 13-20, 2015*

INTRODUCTION

GRID Computing has materialized as the next-generation distributed computing policy that aggregated the discrete diverse resources for resolving numerous kinds of large-scale applications. There is a massive range of diverse and geographically scattered resources are available in the grid. They are computational resource, equipment resource, storage resource etc. Grid resources are geographically scattered among several executive domains and possessed by several organizations. For the majority of grid models, scheduling is a very significant mechanism. An easy way to schedule the jobs is to assign the incoming tasks to available well-matched resources. It is more advantageous to use more progressive and sophisticated schedulers. Schedulers can be systematized as hierarchical form or distributed to deal with the huge scale of the grid. The scheduler is used to accomplish the jobs and resources. The scheduler executes two main roles. First scheduler chooses the suitable computational resource for the job and then allocates the resource to the jobs. Job Scheduling is done in order to make the

Corresponding Author: Kannan K.S., NPR College of Engineering and Technology, Department of Computer Science and Engineering, Kannan .K.S, Box.3030. Dindigul, Tamil Nadu, India.

well-organized use of resources and for continuous execution of jobs. The key objective of scheduling is to decrease the completion time of an application by correctly assigning the jobs to the processors. The two major parties in grid computing are termed as users/resource consumers and resource providers. The resource consumers submit various claims and resource providers share their resources. As one of the important criteria for scheduling the jobs into resources is termed as Quality of Service (QoS).

Load Balancing is a methodology that equally distributes the workload across many devices/computers to improve the resource utilization and also helps to reduce the response time. The major goal of load balancing is to balance the load among all the processors. A decent scheduling algorithm can efficiently assign the jobs to resources and balance the system load (Manimala, R. and P. Suresh, 2013). Load balancing can be categorized into two types. One is static load balancing and another one is dynamic load balancing. In static approach, the decisions linked to load balance are decided at compile time during the resource requirements are projected. The algorithm accurately works when the nodes are having less deviation in the load. Hence, these algorithms are not well-matched in grid environment. In dynamic approach, the decisions are made to distribute the load during run time.

There is a lot of research are mainly focused on scheduling the dynamically arrived independent parallel job for set of resources. The metrics used for job scheduling such as throughput, system utilization, wait time etc. Though, there has been only few works addressing the Quality of Service (QoS) in parallel job scheduling. QoS is the combined determination of service performance that estimates the degree of fulfillment of the user service. It is estimated based on the qualitative measures like latency, completion time, throughput, reliability, execution time and packet loss rate. In this paper, an efficient QoS based job scheduling approach is proposed for job dispatching among the given set of resources.

The remainder of this paper is organized as follows. Section 2 summarizes the related works in job scheduling in grid environment. Section 3 describes the about the environment creation. Section 4 describes about the QoS based job scheduling approach. Section 5 describes the performance analysis. And finally, the paper is ended with the conclusion and future work at section 6.

Related Work:

This section deals with the several grid computing job scheduling models with the fault recovery and QoS measures. *Guo et al* presented a Local Node Fault Recovery (LNFR) mechanism into grid structures. The authors (Guo, S., 2011) were analyzed the fault recovery and presented an in-depth study on grid service reliability modeling. The lifetime of subtasks and the number of reclamations were executed among the grid nodes. Moreover, multi objective task scheduling optimization model and Ant Colony Optimization (ACO) algorithm was developed for fault recovery on grid service reliability. *Azgomi et al* suggested a task scheduling and task execution framework. The task scheduling and execution phase were executed based on Resource Management System (RMS) and Colored Petri Nets. The CPN based modelling pattern clarifies the process of task distribution and execution in grid environment (Azgomi, M.A. and R. Entezari-Maleki, 2010). *Taheri et al* presented a heuristic approach called JDS-HNN. This model simultaneously schedules the jobs and replicate the data files into diverse entities of a grid model. This scheme was stimulated by a natural distribution of a variability of stones among diverse jars. A Hopfield Neural Network was utilized for optimization (Taheri, J., 2013).

Taheri et al proposed a Bee Colony based optimization algorithm termed Job Data Scheduling with Bee Colony (JDC-BC). JDS-BC resides two mechanisms to schedule the jobs and replicate the data files on storage nodes (Taheri, J., 2013).*Balasangameshwara et al* developed a load balancing methodology with the neighbor based and cluster based load balancing approaches. The fault tolerant scheduling approach called MinRc was explored. Also, performance driven fault-tolerant load balancing algorithm was implemented for independent jobs (Balasangameshwara, J. and N. Raju, 2013). *Dutot et al* proposed an approximation algorithm for the multi-organization scheduling problem. Each organization aims at reducing the execution time of their own jobs/tasks. A global centralized mechanism was used to design a solution to enhance the global performance of the organism (Dutot, P.F., 2011). *Hasham et al* formulated a Complex Scientific workflow execution based on intelligent job scheduling and data access policies. The pilot job concept was used, which has the intelligent data reuse and job execution tactics to reduce the scheduling, execution, queuing and data access latencies (Hasham, K., 2011). *Kim et al* designed a model for job partitioning and allocation. It was designed based on the prediction model. It calculates the response time to process the distributed application in every mobile node. The prediction model algorithm partitions and allocates the distributed jobs to the obtainable nodes for fast job processing (Kim, T.K., 2009).

Deepan et al proposed a job scheduling technique. This approach was based on genetic algorithm to schedule the jobs in the grid. Evolutionary Algorithms (EA) drives on a population of potential solutions Natural evolution approaches like inheritance, mutation, selection and crossover were utilized for optimization problems (Deepan

Babu, P. and T. Amudha, 2014). *Yousif et al* suggested a Discrete Firefly Algorithm (DFA) for job scheduling across computational grid. The DFA algorithm maps the jobs into available resources to complete the submitted jobs within a short period of time. This methodology uses the population based candidate solutions to avoid tricking in local optimizations (Yousif, A., 2014). *Conejero et al*proposed a Service Level Agreement (SLA) based framework for Meta-scheduling. This scheme uses two layers on the top of the grid framework. The layers study the SLA standard protocol by the Open Grid Forum (OGF) (Conejero, J., 2011). *Kang et al* presented a global optimal service selection model. The analytic hierarchy process was used to compute the satisfactions of service QoS and trust. Based on the two metrics the synthesized satisfaction is estimated. Then, 0-1 integer programming was formulated based on the synthesized satisfaction (Kang, G.S., 2011).

Revar et al suggested a machine learning approach for load balancing in grid environment. The authors (Revar, A., 2010) were analyzed the load balancing requirements across grid domain. *Yagoubi et al* proposed a load balancing model. This model can symbolizeany grid structure into a forest structure. The load balancing was achieved based on the following two levels. They were the reduction of average response time and their transferring cost (Yagoubi, B. and M. Meddeber, 2010). *Naseera et al* designed a load balancing grid model. This methodology considers the trustworthiness of the resource to participate in the load balancing process. The system transfers the partial workload of the system from the workload nodes into some of the idle nodes (Naseera, M.S. and K.M. Murthy, 2010). *Mandloi et al* proposed an adaptive job scheduling for computational grid. The methodology was based on ant colony optimization with the generic parameter selection (Mandloi, S. and H. Gupta, 2013). *Mayuri et al* discovered a dynamic load balancing algorithm to overcome the issues of centralized and decentralized methods (Mehta, M. and D. Jinwala, 2011). *Pooranian et al*modelled a hybrid scheduling algorithm to solve the sovereign task scheduling issue. Hence Particle Swarm Optimization (PSO) was explored with the Gravitational Emulation Local Search (GELS) algorithm called PSO-GELS (Pooranian, Z., 2013). The authors in (Jayapandian, N. and A.M.Z. Rahman, 2011) also proposed the PSO algorithm for job scheduling across grid structure.

Environment Creation-Globus 4.0.5:

The proposed grid environment is created based on the Globus Toolkit 4.0.5. The definition of Globus is

- A *community* of developers and user, work together on the development of open source software and related documentation for distributed totaling.
- *Software*: It is a set of libraries and programs that resolves the issues occurred during the constructing of the distributed model services.
- *Infrastructure,* which supports these communities-email lists, code repositories, problem tracking system etc.

Globus software is formulated to permit the applications that subordinate with the scattered resources. The earlier Globus toolkit versions are not supported by Java packages for programming the grid environment. Hence, the most popular globus toolkit 4.0.5 is used for creating the proposed grid environment.

Structure Of Globus 4.0.5:

The Globus 4.0.5 architecture is shown in fig.1. It has three main team of services which are accessible through a secure layer. They are Resource Management, Data Management and Knowledge Services. The native service layer contains the software services, network services like TCP/IP, cluster scheduling services, job-submission, question of queues, and so on.

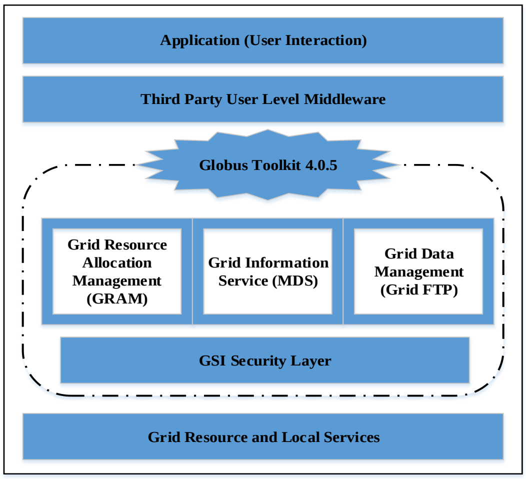

Fig. 1: Architecture of Globus 4.0.5.

The upper layers of the Globus model alter the combination of multiple or heterogeneous clusters. The core service layer contains the Globus toolkit building blocks. They are security, job submission, knowledge management, and resource data management. The high-level services and tool layer contains the tools that integrate the lower level services or implement the missing functionality.

Gsi Security Layer:

The Grid Security Infrastructure (GSI) provides strategies for authentication of grid users and secure communication. It supports SSL (Secure Sockets Layer), PKI (Public Key Infrastructure) and X.509 Certificate Architecture. The GSI provides services and protocols which associates in libraries to realize the subsequent aims for Grid security:

- Single sign-on for mistreatment of grid services through user certificates
- Resource authentication through host certificates
- Knowledge encryption
- Authorization

Resource Management:

The resource management package allows resource allocation through job submission, staging of viable files, job observance and result gathering. The elements of Globus among this package are:

Globus Resource Allocation Manager (Gram):

GRAM provides remote execution capability and report status for the course of execution. A consumer request for employment submission to the gatekeeper daemon on the remote host. Once authentication is over, the gatekeeper starts a job manager that initiates and monitors the work execution. Job manager's area unit are created based on the native scheduler count on that system. The job details area unit through the Globus Resource Specification Language (RSL) that could be a half of GRAM. RSL provides syntax which consist of attribute-value pairs for describing the resources. They are needed for employment, which includes the minimum memory and the variety of CPUs.

Information Services:

The data service package provides static and dynamic properties of the nodes. The area unit is connected to the Grid. The Globus part among this package is named Monitoring and Discovery Service (MDS). MDS provides support for business and querying of resource data. Within MDS, an outline class represents the varied properties of the system. MDS has a three-tiered structure at the bottom of the Information suppliers (IPs). It gathers the knowledge regarding about resource properties and translate them into the format defined by the article categories. The Grid Resource Information Service (GRIS) forms the second tier and runs on one resource. GRIS responds to queries about the resource properties and updates its cache at intervals. The intervals are outlined by the time-to-live scheme by querying the relevant IPs. At the top level, the Grid Information Index Service (GIIS) indexes the resource data which provided by different GRISs and GIISs. The GRIS and GIIS run on Light-weight Directory Access Protocol (LDAP) backend. Here, the knowledge is represented as a hierarchy of entries. Every entry consisting of zero or additional attribute-value pairs. The quality set of IPs provides knowledge on mainframe sort, system design, variety of processors and memory offered among others.

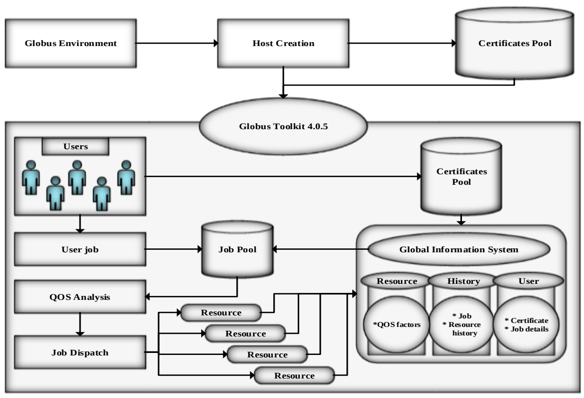

Fig. 2: Structure of proposed Globus-QOS driven Job Scheduling Approach.

Grid ftp:

Grid FTP provides GSI support for authenticated knowledge transfer, third-party transfer invocation and stripes, parallel and partial knowledge transfer support. Replica Location and Management part supports the multiple locations for an equivalent file throughout the grid. The files are often registered with the Replica Location Service (RLS) and its replicas are often created and deleted. Within RLS, a file is known by its Logical File Name (LFN) and it is registered among a logical assortment. The record of the file points to its physical locations. This data is available from the RLS upon querying.

Construction Of Qos Based Job Scheduling Framework:

The user jobs are scheduled based on the QoS analysis in order to provide better resource utilization in grid environment. One of the key strengths of the proposed model is to calculate the system parameters like QoS parameters (CPU usage, Memory usage, Failure rate and Network speed). Each system and each user have individual certificate. A system may have one or more users. The user jobs are submitted into job pool. The certificate pool contains the certificates for users and their corresponding systems. The Global Information System (GIS) maintains the resource, history and user information.

Construction Of Qos Based Job Scheduling Framework:

The overall steps followed in the construction of the proposed model are outlined in the subsequent steps:

1. Primarily, set the Globus toolkit package
2. Create the set of services in the grid resources
3. Initialize the number of users
4. Collect the user request and put it into job pool
5. Allocate the jobs to appropriate services based on the matched services with the job request
6. Consider the QoS factors like resource details and their history to overcome from failures
7. Allocate the jobs to the selected resource
8. Execute the jobs in the grid environment

Qos Analysis:

The system parameters are evaluated based on the following measures:

a. CPU usage
b. Memory usage
c. Failure rate
d. Network speed

The CPU usage, memory usage, failure rate and network speed for each user's job is evaluated and monitored. The grid system is a difficult system that spans multiple heterogeneous groups. It becomes complex to guarantee that there will be no failure arises. The failures may occur due to software bugs, electronic component failures, human operator errors, severe congestion or performance overload. Usually, when a node executes a task, at that time a grid node failure occurs, then the output of the task will be incorrect. Otherwise, no results will be sent to the RMS. Likewise, if a communication link failure happens during the data transfer, then the received information will be unpredicted.

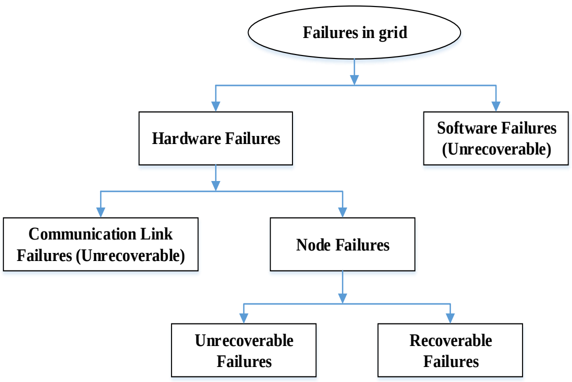

Fig. 3: Categorization of failures in the grid based on recoverability.

Fig.3. demonstrates the categorization of failures in grid systems according to recoverability. The local node fault tolerance mechanism can be incorporated to achieve the fault tolerance by recovering from failures. Hence the failed constituents/modules are repaired or replaced by new modules. The fault tolerance mechanism can save resources and execution time. Even though, it is not able to recover all the failures in the grid. Based on the recoverable property, the failures can be classified into two kinds:

1. Recoverable failures
2. Unrecoverable failures

Due to unrecoverable hardware failures the subtask may be terminated. Recoverable failures are triggered by human operation errors or overload. The interrupted execution of subtask can be restarted by recovering the required information once the failed node becomes functional.

Performance Analysis:

The experimental evaluation of the proposed approach is conducted to evaluate the performance of the system. Globus toolkit 4.0.5 is used to create the grid environment. The toolkit supports the system to schedule the jobs to appropriate resources.

Average Response Time Investigation:

Average response time is the average amount of time the requester (user) should wait before a global resource request could be granted. It depends on cycle time. i.e the amount of time that the message needed to complete the cycle. The average response time in noted for the proposed Globus-QoS and the existing Minimum Cost Match Schedule (MCMS). The result is shown in fig.4. It shows that the proposed model takes lesser time to respond back to the requester.

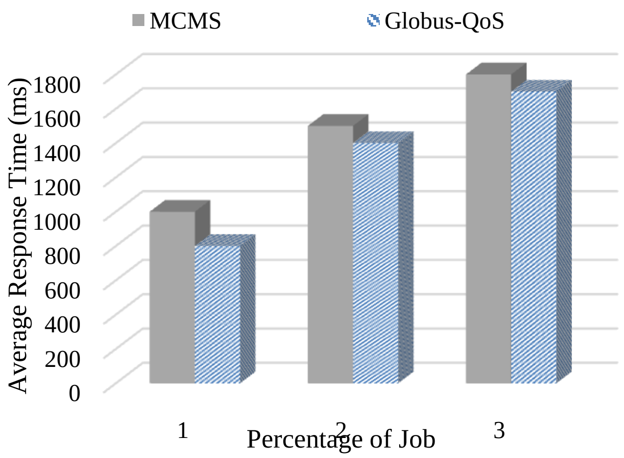

Fig. 4: Average Response Time for Globus-QoS (proposed) and MCMS (existing).

System reliability analysis:

Reliability can define to be the possibility that the system will not fail during the time that it is executing the tasks. The reliability of the system is monitored and exposed in fig.5. It obviously shows that the proposed Globus-QoS model provides better reliability than the existing approach.

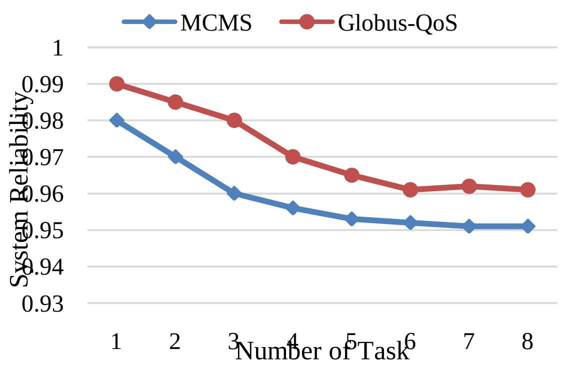

Fig. 5: System Reliability with respect to the number of tasks for Globus-QoS (proposed) and MCMS (existing).

Table I: system reliability study for globus-qos and mcms model.

Number of Jobs	MCMS	Globus-QoS
1	0.98	0.99
2	0.97	0.985
3	0.96	0.98
4	0.956	0.97
5	0.953	0.965
6	0.952	0.961
7	0.951	0.962
8	0.951	0.961

In TABLE I, the system reliability measure is evaluated and illustrated to show the effectiveness of the proposed model. The average rate of the proposed Globus-QoS model results 0.972 and the existing model MCMS results 0.95.

Expected And Execution Time Based Analysis:

The system performance is enhanced based on the expected time and execution time. The robust system should execute the task within the expected time. If the system takes more time to execute the job, then automatically the performance degrades. The proposed model almost satisfies the above property. i.e. the Globus-QoS can execute the task with more or less the expected time. Hence the proposed model can provide efficient and better system performance which is shown in fig.5.

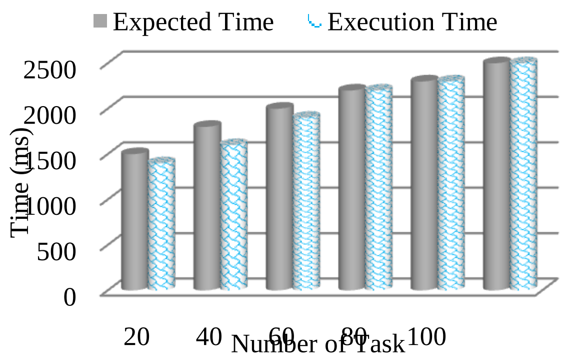

Fig. 5: Execution time with the expected time analysis with respect to the number of task.

The execution time for the diverse numbers of job is monitored and compared with the existing model. It is illustrated in fig.6. It is noted that the Globus-QoS model can execute the job within a short period of time than the MCMS model.

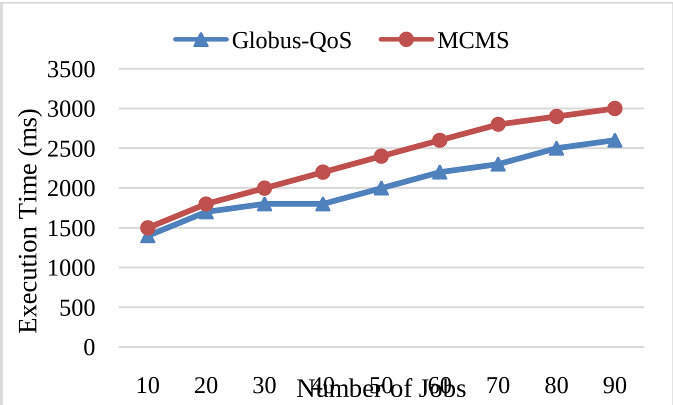

Fig. 6: Execution time with respect to the number of jobs for Globus-QoS and MCMS.

The performance of the system is experimented by the Globus environment. From the experimental results, it can be evidently proves that the proposed Globus-QoS driven job scheduling approach can results better in terms of response time, execution time and system reliability.

Conclusion And Future Work:

In this paper, an efficient Globus-QoS driven job scheduling approach for grid environment is presented. The major contributions of this approach are to scrutinize with the QoS factors. The QoS measures like CPU usage, Memory usage, Failure rate and Network speed are evaluated and monitored for scheduling the jobs into an appropriate resource. GIS helps to maintain the user job history, user details, certificates and job details. The jobs are scheduled to the suitable resources based on the GIS reports. The grid environment is constructed by the Globus toolkit. The experimental result shows that the proposed model can schedule the jobs within a short period of time and they also the resource request can be granted within a short period of time. The service reliability also comparatively better than the existing Minimum Cost Match Schedule (MCMS). In future, the Geospatial Web Provisioning Service is incorporated with this proposed framework to enhance the query processing in grid environment.

REFERENCES

Manimala, R. and P. Suresh, 2013. Load balanced job scheduling approach for grid environment. in Information Communication and Embedded Systems (ICICES), International Conference on, 336-339.

Guo, S., H.Z. Huang, Z. Wang and M. Xie, 2011. Grid service reliability modeling and optimal task scheduling considering fault recovery.Reliability, IEEE Transactions on, 60: 263-274.

Azgomi, M.A. and R. Entezari-Maleki, 2010. Task scheduling modelling and reliability evaluation of grid services using coloured Petri nets. Future Generation Computer Systems, 26: 1141-1150.

Taheri, J., A.Y. Zomaya, P. Bouvry and S.U. Khan, 2013. Hopfield neural network for simultaneous job scheduling and data replication in grids.Future Generation Computer Systems, 29: 1885-1900.

Taheri, J., Y. Choon Lee, A.Y. Zomaya and H.J. Siegel, 2013. A Bee Colony based optimization approach for simultaneous job scheduling and data replication in grid environments.Computers & Operations Research, 40: 1564-1578.

Balasangameshwara, J. and N. Raju, 2013. Performance-Driven Load Balancing with a Primary-Backup Approach for Computational Grids with Low Communication Cost and Replication Cost.Computers, IEEE Transactions on, 62: 990-1003.

Dutot, P.F., F. Pascual, K. Rzadca and D. Trystram, 2011. Approximation Algorithms for the Multiorganization Scheduling Problem.Parallel and Distributed Systems, IEEE Transactions on, 22: 1888-1895.

Hasham, K., A.D. Peris, A. Anjum, D. Evans, S. Gowdy, J.M. Hernandez, 2011. CMS Workflow Execution Using Intelligent Job Scheduling and Data Access Strategies.Nuclear Science, IEEE Transactions on, 58: 1221-1232.

Kim, T.K., J.H. Park, S.S. Yeo, H.S. Seo and J. Kwak, 2009. Job partition and allocation using the prediction model in non-dedicated heterogeneous wireless network environments.Communications, IET, 3: 764-771.

Deepan Babu, P. and T. Amudha, 2014. A Novel Genetic Algorithm for Effective Job Scheduling in Grid Environment. in Computational Intelligence, Cyber Security and Computational Models, 246. G. S. S. Krishnan, R. Anitha, R. S. Lekshmi, M. S. Kumar, A. Bonato, and M. Graña, Eds., ed: Springer India, 385-393.

Yousif, A., S. Nor, A. Abdullah and M. Bashir, 2014. A Discrete Firefly Algorithm for Scheduling Jobs on Computational Grid. in Cuckoo Search and Firefly Algorithm, 516. X.-S. Yang, Ed., ed: Springer International Publishing, 271-290.

Conejero, J., C. Carrión and B. Caminero, 2011. A SLA-based Framework with Support for Meta-scheduling in Advance for Grids. in Intl. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA).

Kang, G.S., J.X. Liu, and R. Hu, 2011. Global Optimal Service Selection Model Based on Grid Service.Journal of Chinese Computer Systems, 32: 1880-1803.

Revar, A., M. Andhariya, D. Sutariya and M. Bhavsar, 2010. Load balancing in grid environment using machine learning-innovative approach.International Journal of Computer Applications (0975–8887), 8: 1.

Yagoubi, B. and M. Meddeber, 2010. Distributed load balancing model for grid computing.Revue ARIMA, 12: 43-60.

Naseera, M.S. and K.M. Murthy, 2010. Trust Aware Load Distribution in a Data Grid Environment.International Journal, 2.

Mandloi, S. and H. Gupta, 2013. Adaptive job Scheduling for Computational Grid based on Ant Colony Optimization with Genetic Parameter Selection. International Journal.

Mehta, M. and D. Jinwala, 2011. A Hybrid Dynamic Load Balancing Algorithm for Heterogeneous Environments.in Proc. Int. conf. Grid Computing and Applications, 61-65.

Pooranian, Z., M. Shojafar, J.H. Abawajy and A. Abraham, 2013. An efficient meta-heuristic algorithm for grid computing.Journal of Combinatorial Optimization, 1-22.

Jayapandian, N. and A.M.Z. Rahman, 2011. A particle swarm optimization algorithm for job shop scheduling in grid environment.