IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2016

An Efficient Job Scheduling in Computational Grid using Multi-Gap Elimination Approach

Haruna Abdu[1], Abubakar Aliyu[2], Faisal Murtala[3], Aliyu Aminu AbdulHadi[4]

Department of Computer Science, Federal University Lokoja, Kogi, Nigeria [1]

Department of Computer Science, Umaru Musa Yar'adua University Katsina, Katsina, Nigeria [2, 4]

Department of Computer Science, Al-Qalam University Katsina, Katsina, Nigeria [3]

Abstract: The proper functioning of a grid depends mainly on the efficient management and use of grid resources to carry out the various jobs that users send for execution. In grid computing system management of resources and scheduling of jobs are the most crucial problem, a lot of effort had been made to efficiently schedule and manage resource in computational grid systems, scheduling policies such as, First Come First Serve which is a simple policy used to improve efficiency of the scheduler, even though it have been widely used, it therefore end up with invoking low system utilization of both global and local scheduler as a result of the gaps that exists in between the jobs submitted in the waiting queue. To improve system utilization and efficiency of both main and local schedulers back filling approach is used, this approach allows tasks or jobs in the waiting queue to fill in those gaps that exist. This paper proposes an efficient scheduling algorithm based on multi-gap elimination approach. The proposed algorithm uses intelligent agents in the scheduler to perform scheduling in a collaborative and coordinated way. The algorithm uses gap filling strategy to optimize priority rule algorithms in grid scheduling system by considering the available gaps in both global and local schedulers.

Keywords: Gap filling, Scheduler, Algorithm, Task.

I. INTRODUCTION

Computer grids are systems containing heterogeneous, autonomous and geographically distributed nodes that are capable of executing both local and external jobs. With the advancement of computing paradigms, Grid computing has emerged as a promising attractive distributed computing. However, Computational Grids aim to aggregate the power of heterogeneous, geographically distributed, multiple-domain-spanning computational resources to provide high performance or high-throughput computing [1]. Grid computing technology provides users with promising potentials such as; Resources balancing, Exploiting underutilized resources, Collaboration, Reliability and Scalability. To achieve the promising potentials of computational Grids, an effective and efficient scheduling of jobs and resource management is immensely needed. Grid scheduling system is unlike traditional scheduling system due to its heterogeneous and dynamic nature [2].

The grid scheduling problem deals with assigning resources to a set of tasks that enter the grid through different nodes at any instant of time, considering availability (dynamic and autonomous) and computing capacity (heterogeneous), among other things [3]. The different parameters and requirements relevant to the grid's clients and its resources must also be considered to ensure the quality of the services for the different actors in the grid. This paper focuses on the design of efficient scheduling algorithm for computational grids that uses gap filling techniques used to optimize First come First Serve (FCFS). First of all, works that had been carried out related to job scheduling in computational grid are investigated and their relevant short comings. Then we design a job scheduling architecture which is useful for guiding the design of the efficient algorithm. A common grid scheduler architecture is briefly discussed with the aim at having insight in to possible components for grid scheduling. Finally an efficient algorithm is proposed that will enhance the efficiency of both local and global schedulers.

This research is of important to enhance the promising potentials that grid technology provides to both clients and grid service providers. However, it will also help researchers who have interest in research on grid scheduling or grid computing at large.

II. REVIEW OF RELATED WORKS

An Over the past several years, a lot of research has been conducted to study the problem of job scheduling and resource management in grid environment, this leads to the development of many scheduling algorithms by different researchers to tackle such problems based on some specific application domains. Gap filling techniques plays an important role in grid computing environment for scheduling tasks.

Some of the related literatures reviewed are summarized in table 1, which highlight the works done by some researchers and their limitations.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2016

TABLE I SUMMARY OF REVIEWED RELATED WORKS

S/N	Author(s)	Year	Title of the Research	Work done	Limitation
1.	[4]	2016	Proposed Efficient Scheduling Algorithm in Grid Computing Using Gap Filling Approach	The authors proposed an algorithm aim to eliminate gaps in the global job queue	The algorithm consider only the global queue in the global scheduler
2.	[5]	2011	Proposed A New Approach To Solve The Decentralized Constrained Dynamic Multi-project Scheduling.	The author Develop Multi-agent System for Dynamic Multi-project Scheduling.	The algorithm useful only in project constrained environment
3.	[6]	2013	An efficient Resource Management and Scheduling Technique for Fault Tolerance in Grid Computing.	The presented techniques are based on queues based polices and are easy to implement.	This method is not very effective in mapping jobs to resources.
4.	[7]	2002	Eight Agent Based Algorithms for Solving Multimode Resource Constrained Project Scheduling Problem.	The work presented is based on small number of agents.	The algorithm is restricted to work with only eight agents
5.	[8]	2011	An Improved Backfilling Algorithm: SJF-BF.	Proposes An Improved Backfilling Algorithm Based On SJF.	The Algorithm works only with SJF policy.
6.	[9]	2013	Intelligent Agent Based Grid Resource Management System.	Agents are used in each computing node for scheduling	This work uses easy backfilling algorithm.
7.	[10]	2014	An efficient Job scheduling Approach in Grid Environment.	This work uses EGDF Method to fill biggest gap.	This work is used to fill earliest biggest gap only.
8.	[11]	2012	Agent Based Approach To Grid Scheduling Problem.	This work proposes communication mechanism between two or more agents	The algorithm is developed and used in Auvergrid only.
9.	[12]	2013	Job Scheduling in Grid Computing.	In this paper, the author explain job scheduling and resource sharing concept.	The scheduling concept is limited to the main scheduler. It did not cover the local schedulers.
10.	[17]	2016	Comparative Study of Various Task Scheduling Algorithm in Grid computing	This paper is a survey about various task scheduling algorithms which control the task scheduling of entire system	The paper is limited to some certain parameters such as, QoS, failure recovery, replication etc.

III. TASK SCHEDULING IN GRID SYSTEM

Grid computing as a promising attractive distributed computing paradigm that aim at converging heterogeneous and geographically distributed computing resources to be shared among users, efficient scheduling is need to achieve the promising potentials of grid system. Compared to traditional systems for distributed environment, such as clustering computing, Grid scheduling systems have to take into account diverse characteristics of both various Grid applications and various Grid resources [3].

Grid environment contains a lot of computing resources that can be shared and used among users. The grid resources can be allocated or deallocated based on the availability of the resource; this is as a result of the uncertainty of the presence of a particular node present in the grid.

A node can join and leave at any time, due to the dynamic nature of grid system resources availability, the grid management system gives higher priority to local service

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2016

request and then consider the external service request for execution.
After client sent a job for execution it will remain in the queue, it's the function of grid information system to choose which node is suitable for execution of such a job. Since, the grid information system contains information about all the grid resources such as scheduler, the total number or resources available, CPU capacity, memory, bandwidth, current demand on every node etc. There are two kind of scheduler in grid environment:

A. Local Scheduler
It's also called a scheduler; it's responsible for scheduling of jobs and managing resources at a particular node. This scheduler it's reside in a node that is assigned or chosen to execute a job by the grid scheduler.

B. Global Scheduler
It's also called Grid Scheduler, which is responsible for selecting appropriate local site and mapping of jobs on to the selected site or domain [14].

Scheduling can also be classified into static and dynamic scheduling. In static scheduling, before execution the jobs are assigned to the suitable machines and those machines will continue executing those jobs without interruption. In dynamic scheduling, the rescheduling of jobs is allowed [15]. The jobs executing can be migrated based on the dynamic information about the work load of the resources [16].

IV. TASK SCHEDULING IN GRID SYSTEM

In computational grid, there are a large number of computational resources available for many tasks, such that different jobs within the local and global job queue can be executed by different suite of available grid resources. Generally the global objective of job scheduling is to have good load balancing among the processors, whereas for the later minimization of overall execution time is the main concern [13] The fig. 1 is a complex job scheduling architecture that represents the process of job scheduling right from its submission to the final stage of processing.

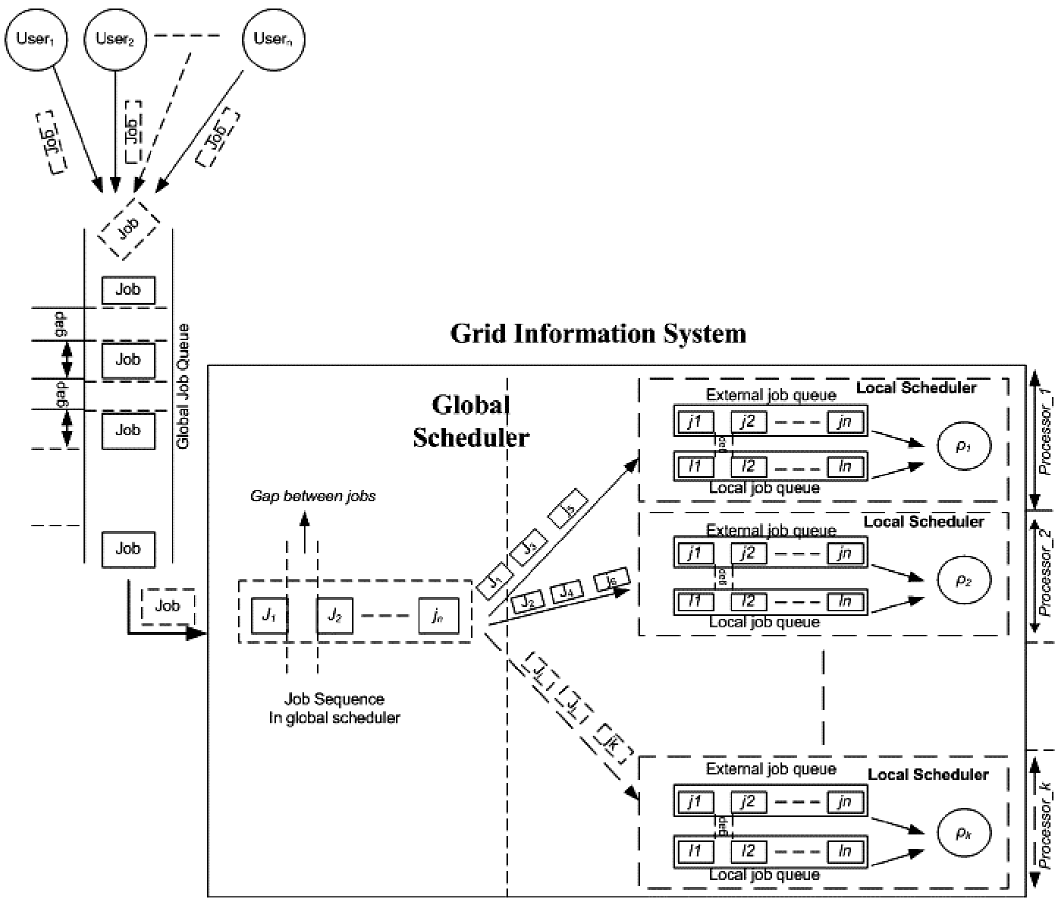

Fig. 1. Job Scheduling Architecture

i. Assumptions
In this system we consider all the available processors are active through, that will make the system static in terms of the number of available resources (processors). This is because if the processors are made dynamic (they can leave and join) it will cause instability or delay in job execution. In addition we consider also there exist no gap between the local jobs in any active processor as a result of their high priority to external jobs.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2016

ii. System Outline

In this section we try to define some content of the architecture, which will immensely help in developing a propose algorithm for efficient utilization and management of the available computing resources in the architecture.

Definition 1 (Number of Task): The set of tasks or jobs in the global (main) queue waiting to be allocated to available node(s) by global scheduler is represented by η.

Such that: $\eta = \langle \tau_1, \tau_2, \ldots\ldots\ldots\ldots, \tau_m \rangle$

Where, η_n represent the total number jobs in the global queue, which is given by:

$$\eta_n = \sum_{i=1}^{m} \tau_{i,} \qquad (1)$$

Definition 2 (Gaps in the Queue): ϑ represent the set of gaps that exist in the global queue, Such that $\vartheta = \langle g_1, g_2, \ldots\ldots\ldots\ldots, g_n \rangle$.

Where, ϑ_n represent the total number of gaps in the global queue, and it's given by:

$$\vartheta_n = \sum_{i=1}^{n} g_{i,} \qquad (2)$$

Such that: η_n must be greater than ϑ_n ($\eta_n > \vartheta_n$)

$$\sum_{i=1}^{m} \tau_i > \sum_{i=1}^{n} g_i$$

Definition 3 (Available nodes): let ρ represent a set of active number of available nodes (processors) in the Grid Information System (GIS) that are capable of executing jobs assign to them. The set is given by; $\rho = \langle \rho_1, \rho_2, \ldots\ldots\ldots\ldots, \rho_k \rangle$

Where, ρ_n represent the total number of active processors in the GIS, which is given by:

$$\rho_n = \sum_{i=1}^{k} \rho_i \qquad (3)$$

Definition 4 (Number of external Jobs in the Local queue): We represent the total number of external jobs to be executed by a particular processor (ρ_i) by $\Gamma_{\rho i}$ and Γ represent the entire distributed external tasks among the active processors.

Therefore:

$$\Gamma = \Gamma_{\rho 1} + \Gamma_{\rho 2} + \ldots\ldots\ldots\ldots + \Gamma_{\rho k}$$

$$\Gamma = \sum_{i=1}^{k} \Gamma_{\rho i} \qquad (4)$$

Definition 5 (Local Jobs): Individual processors has their own local jobs to be executed without waiting that we must capture because of their high priority to external jobs in execution process. We represent the total number of local jobs to be executed by a particular processor (ρ_i) by $\mathcal{L}_{\rho i}$ and $\mathcal{L}$ represent the entire distributed local tasks among the active processors.

Therefore:

$$\mathcal{L}_{\rho i} = \mathcal{L}_{\rho 1} + \mathcal{L}_{\rho 2} + \ldots\ldots\ldots\ldots + \mathcal{L}_{\rho k}$$

$$\mathcal{L} = \sum_{i=1}^{k} \mathcal{L}_{\rho i} \qquad (5)$$

Definition 6 (Local Gaps): After jobs is assign for processing in active processors in the system, there would be gap(s) in the local job queue of the individual nodes. We represent the total number of local gaps that exist in a particular processor (ρ_i) by $\lambda_{\rho i}$ and λ is the total number of gaps of the entire local queues of the individual processors.

Where:

$$\lambda_{\rho i} = \lambda_{\rho 1} + \lambda_{\rho 2} + \ldots\ldots\ldots\ldots + \lambda_{\rho k} \text{ and}$$

$$\lambda = \sum_{i=1}^{k} \lambda_{\rho i} \qquad (6)$$

V. PROPOSE SCHEDULING ALGORITHM

Algorithm: - Multi-gap elimination Algorithm.
Input: - Jobs <1, 2……. m>
Output: - Efficient scheduling through gap(s) elimination

Phase I (global Scheduler)

If $\rho_n >= \eta_n$ then
    For i=1 to m
        $\rho_i \leftarrow \tau_i$
    End for
Else
For i =1 to m-1
If there exist g_i between τ_i and τ_{i+1}
add g_i to ϑ[i]
end if
end for
end if
For i=1 to ϑ_n
If τ_i.size can fit g_i then
$g_i \leftarrow \tau_i$
else
i ++
end else
end if
end for

Phase II (Local Scheduler)

If $\exists \mathcal{L}_{\rho i}$ and $\exists \mathcal{L}_{\rho i}$then
For i=1 to $\mathcal{L}_{\rho k}$
$\rho_{i\leftarrow} \mathcal{L}_{\rho i}$
I++
End for
Else
For i=1 to n

If there exist $\lambda_{\rho i}$ between λ_i and λ_{i+1}
Add it to l[i]
//were l [1…n] is a local dynamic array
End if
End for
end else
end if

For i=1 to λ_k
If $\mathcal{L}_i$ .size can fit λ_i then
$\lambda_{i} \leftarrow \mathcal{L}_i$
else
i ++
end else
end if
end for

The algorithm is divided in to two phase's, the first phase takes care of all the external jobs that are yet to be executed or process by waiting in the global queue while waiting for the global scheduler to schedule them to the existing active nodes in the system that are capable of executing them. Gaps that existin the global queue are filled by jobs that can fit the size.

Them the second phase, takes care of both the local and external jobs that are schedule to be executed by a particular processor in the grid system, then the individual local schedulers in the each nodes gives high priority to local job and later consider the scheduled external job. Gap(s) that exist in the local queue is/are filled by the existing jobs scheduled to that node.

VI. CONCLUSION

Grid computing strive to aggregate diverse, heterogeneous, geographically distributed and multi-domain spanning unutilized computing resources to provide a platform that will make such unutilized resources to be useful to needful most individuals or group of peoples or companies; that will lead to high performance resource sharing in a secured and coordinated manner. Parallel execution of independent jobs or task in computational grid is the one of the most attractive feature of it. However, sharing and managing such unutilized computational resources is a crucial problem in grid systems. This paper proposed an algorithm that will help in scheduling and managing grid resources and make both the local and global schedulers to work efficiently.

This research is of important to enhance the promising potentials that grid technology provides to both clients and grid service providers. However, it will also help researchers who have interest in research on grid scheduling or grid computing at large.

REFERENCES

[1] Yanmin Zhu, Lionel M. Ni, (2013) A Survey on Grid Scheduling Systems, Technical Report SJTU_CS_TR_200309001, Department of Computer Science and Engineering, Shanghai Jiao Tong University, 2013.

[2] Suganya G. (2014) An Efficnet Job Scheduling Approach in Grid Environment Using Biggest Hole Priority Algorithm. International Journal of Computer & Organization Trends – Volume 9 Number 1 – Jun 2014

[3] Blythe, J., Deelman, E., Gil, Y., Kesselman, C., Agwal, A., Mehta, G., Vehi, K., (2003) The Role of Planning in Grid Computing, in Proceeding of international conference on AI planning and Scheduling, (ICAPS). Treanto (Italy).

[4] Haruna A., Abubakar A. and Faisal M. (2016), Proposed Efficient Scheduling Algorithm in Grid Computing Using Gap Filling Approach, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 6, Ver. IV (Nov.-Dec. 2016), PP 59-62

[5] Adhau, S., and Mittal, M. L. (2012) A multi-agent based approach for dynamic multi-project scheduling, International Journal of Advanced Operations Management, vol. 3, No. 3/4, pp 230–238

[6] Deeptanoy et al (2013) An efficient Resource Management and Scheduling Technique for Fault Tolerance in Grid Computing in Cracow Grid Workshop '05 Proceedings. Academic Computer Centre CYFRONET AGH, Cracow, Poland.

[7] Knotts et al (2002) Eight Agent Based Algorithms for Solving Multimode Resource Constrained Project Scheduling Problem.IEEE Trans. Vol. 22(2), pp 187-189.

[8] Mishra S., and Kushwah, D.S. (2011) An Improved Backfilling Algorithm: SJF-BF. Int. J. on Recent Trends in Engineering & Technology. Vol. 05 (1), pp. 8-16.

[9] Muthuchelvi, S. and Anandha, R. (2013) Intelligent Agent Based Grid Resource Management System. Proceedings of the 8th International Conference of Distributed Computing Systems, pages 104-111.

[10] Suganya (2014) An efficient Job scheduling Approach in Grid Environment, in Proceedings of the 25th IASTED International Multi-Conference, Parallel and Distributed Computing and Networks. Vol. 6(7), pp 22-34. Innsbruck, Austria.

[11] Victor, P., Solar M., Rojas J., Mendoza M., and Raul, M. (2012) A Multiagent Based Approach To Grid Scheduling Problem.Clei Electronic Journal, vol 15, no. 2, p5, pp 190-105.

[12] Khusboo Y., Deepika J., and Ramandeep S (2013) Job Scheduling in Grid Computing. International Journal of Computer Applications (0975-8887), Volume 69 –No. 22. May, 2013.

[13] Gyung, L.P., (2004), Performance Evaluation of a list Scheduling Algorithm in Distributed Memory Multiprocessor System.

[14] Foster, I., Jennings, N., Kesselman, C. (2004), Brain Meets Brawn: Why Grid and Agents Need Each Other?, Autonomous Agents and Multiagent Systems, InternationalJoint Conference on, Vol.1(8) pp–15-31.

[15] Khushboo Y., Deepika J. and Ramandeep S. (2013), Job Scheduling in Grid computing. International Journal of Computer Applications, Vol-69- No. 22.

[16] Chtpen, (2005), Dynamic Scheduling in grids system. Six Firw, PhD Symposium, Faculty of Engineering, Ghent University.

[17] Er. Harkiran K. and Er. Mandeep K., (2016), Comparative Study of Various Task Scheduling Algorithms in Grid Computing. International Journal of Advance Research in Computer and Communication Engineering, Vol. 5. Issue 5, May 2016.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2016

BIOGRAPHY

Abdu Haruna received a B.Sc. Degree in Computer Science in 2011 and M.Sc. Degree in Computer Science in 2016 from Umaru Musa Yar'adua University Katsina, Nigeria. He is currently an Assistant Lecturer with Department of Computer Science, Federal University Lokoja, Kogi State, Nigeria. His research interest includes Grid Computing, Cloud computing, Fog computing, IoT and performance analysis of computer and network systems.

Abubakar Aliyu obtained his B.Sc. Degree in Computer Science in 2011 and M.Sc. Degree in Computer Science in 2016 at Umaru Musa Yar'adua University Katsina, Nigeria. His Research interest includes Grid Computing, Cloud Computing, Internet of Things and Fog Computing.

Faisal Murtalaobtained his B.Sc. Degree in Computer Science in 2013 from Al-qalam University Katsina, Nigeria. His research interest includes software engineering, cloud computing and computational grids.

Aliyu Aminu AbdulHadi received a B.Sc. Degree in Computer Science in 2011 and he is currently undertaking his M.Sc. programme in Computer Science at Umaru Musa Yar'adua University Katsina, Nigeria. His research interest includes Distributed Systems, IoT, Mobile Communication.