An Effective and Optimal Approach for Computational Resource utilization to Improve Computational Accuracy in Grid Computing Environment

Prof. Hajare Mangesh [1] , Prof.Pujari Shankar Suryappa [2]

[1,2]Assistant Professor, Sanjeevan Engineering & Technology Institute Panhala. Kolhapur(Maharashtra) India

---

Abstract – In modern days, due to the rapid technological advancements, improving computational accuracy in grid computing has become an essential area of research. Grid computing has raised a new field, differentiated from traditional distributed computing. It concentrates on large-scale resource sharing, innovative forms and in some instances, high-performance orientation. A grid is a network of computational devices that may conceivably span many continents. The grid serves as a whole and complete system for organizations by which the maximum utilization of resources is accomplished. The computational resource utilization is a means which involves resource management and effective job sharing among the resources. Therefore, it is considered to be very valuable in grid systems. These methods are applicable for various systems depending upon the requirements of the computational grid, the type of environment, resources, virtual machines and job profile it is supposed to work with. Each of these models has its qualities and demerits which forms the subject matter of this survey.

Keywords: Grid computing, Distributed systems, Load balancing, Resource allocation, scheduling.

1. Introduction

Grid computing has been growing over the past few years as a means of immense computing power and resource sharing. Grid computing is a distinctive type of parallel computing that relies on complete computers connected in a network by a conventional network interface providing commodity network, compared to the lower productivity of inventing and constructing a smaller form of supercomputers. Grid Computing openly seeks and is capable of adding an infinite number of computing devices into any grid environment, adding to the computing capabilities of the grid.

At present, a very complex set of technology challenges need to be encountered, and the appropriate grid computing solution environment needs to be designed to mitigate these challenges. Grid computing solutions are constructed using a variety of technologies and open standards. This computing provides extremely high-performance mechanisms to access remote computing resources seamlessly.

The quantity and quality requirements for some business related advanced computing applications are day-by-day becoming more and more complex. The industry has now realized that they need, and are conducting numerous complex scientific experiments, advanced modeling scenarios, genome matching, astronomical research, the wide variety of simulations and real-time portfolio management. These requirements can exceed the demands and availability of installed computational power within an organization. The computational grid computing has now become a reality and yet inexpensive computational power for its customers.

The grid provides a scalable, secure and reliable mechanism for collective utilization of computational and other resources. As the computational resources (CPUs) in Grid are heterogeneously scattered and are of the variety of architectures, it becomes necessary to learn and understand the various behavioral parameters. This additional information will provide more effective scheduling and allocation of computational resources to solve complex problems more accurately.

2. Literature Review

For a practical and in-depth understanding of the behavior of computational resources, it is imperative to set a schedule of resource allocation [1] for a given objective. The authors proposed two techniques such as MMS and EMS for job scheduling in Grid. Whenever the scheduler has jobs to schedule, MMS and EMS select the resources based on maximum suitable mapping and exact mapping of job requirement value with resource capable value respectively. The Job manager can monitor the execution of the job and returns the results after the successful completion.

The resource management system may consider the resource economy [2]. Effective resource allocation may consider market participants, resource consumers, and resource owners. The authors investigate the pricing of resources in dynamic Grids based on the computational commodity market of CPU resources. The several categories of CPU are characterized by execution speed. The cost and performance may be used interchangeably in executing jobs which help the selection of suitable resources.

The current scenario of grid computing faced a problem of job failure and increase of execution time of jobs. The growth in execution time depends on resource allocation and task scheduling of job in grid computing [3]. To the reduction of failure and task scheduling of resources need optimization of the process of distribution and execution of the task. Various authors proposed different optimization techniques for process scheduling in Grid Computing. These techniques are based on heuristic algorithms. The usually used algorithms are: Opportunistic Load Balancing (OLB) [5], Max-Min, Min-Min, Minimum Execution Time (MET), Minimum Completion Time (MCT), User Directed Assignment (UDA), Generic Simulated Annealing (GSA) [3], Tabu Search (TS), Genetic Algorithm (GA), Ant Colony Optimization (ACO) [7] and Particle Swarm Optimization (PSO) [5]. All these algorithms prescribe various mechanisms of the Grid scheduling process.

The authors [3] proposed the effective resource allocation based on Grid resource monitoring and Grid resource forecast. Monitoring cares about the running state, distribution load and failure of resource; and resource forecast focuses on the variation trend and running track of resources in grid System utilizing, modeling and analyzing historical monitoring data. Authors also added the historical information generated by computational power resources might be useful to eliminate the bottleneck, diagnosing the fault and minimizing the execution time. The Grid application use resources such as processors (CPUs) in multiple clusters. So processor co-allocation is required.

The computational power thought of as a utility similar to electricity or water [4]. The authors proposed the Faucet framework that aims at providing compute power distribution and selection of computer servers for each job for effective utilization of resources across the Grid. The computational resources (servers) selection improves the quality of service in job execution.

Users can share grid resources by submitting a computing task to grid System. The participation of resources can be active or inactive within the Grid. Therefore, it is impossible for anyone to assign the jobs to the computing resources in Grid manually. Grid scheduling systems select the resources and allocate the user submitted jobs to appropriate resources in such a way that the user and the application requirements are met. Hence, the scheduling process consists of a mapping of the jobs to specific time intervals of the Grid resources. The authors in this paper [6] define the problem of scheduling m jobs within given processing time on n resources. They have considered independent user jobs with heterogeneous resources. The resource allocation should be based on many CPU cycles required for job and number of CPU cycles available with resources. The length of job cycles and the speed of resources assumed to be known based on user-supplied information, experimental data and application profiling or any other relevant techniques. They have proposed a job scheduling algorithm that minimizes the Makespan. Makespan is the measure of throughput of the heterogeneous computing system.

Grid scheduling is an NP-Complete problem [7]. Due to the architecture of the grid, one original issue associated with efficient utilization of heterogeneous resources in a Grid is Grid Scheduling. It is a critical design issue, and challenging since the capability and availability of the resources vary dynamically. As the size of the Grid increases, scheduling becomes even more complicated. A new area of research is developed to design optimal methods. It focuses on heuristic techniques that provide an optimal or near optimal solution for large Grids. The author has proposed Ant Colony Optimization based on Grid scheduling algorithm that allocates an application to a host from a pool of available hosts and applications by selecting the best match.

3. Problem Statement

It is necessary to analyze the resource behavior of dynamically available computational resources in the grid environment and to optimize their allocation considering conventional methods of computational resource allocation and other strategies. Dynamic availability of the computational resource is not effectively utilized for allocation in the Grid environment.

4. Proposed Work

The proposed work consists of the following:

1. Application of computational resources to a computational job with -
   a. Conventional methods
   b. Using Dart algorithm
2. Complex computations by varying computational power up to optimum accuracy level.

1) Scope:

In the Grid environment, computational resources are present in a variety of architectures. These resources are dynamically allocated for job execution. It is necessary to understand the resource behavior in advance for their effective utilization. In the proposed work, the analysis of the computational resource behavior by applying various tasks and resources will be done.

2) Methodology:

The work consists of the need of Grid environment and computational resource cluster to perform various experiments of the execution of jobs with various combinations of sets of resources. The proposed system includes:

1. The setting of the Grid environment as shown in the Fig.1

2. The setting of OpenMPI cluster
3. Experimentation based on conventional and dynamic (Dart Algorithm) resource allocation strategies.
4. The analysis to be done by
   a. Varying sets of the computational resource with a fixed number of instances of the task to study the CPUs responses.
   b. Varying instances of the task with static sets of resources to study the CPU responses
   c.Comparing the results of (a) & (b) to determine the effective allocation of computational resources.

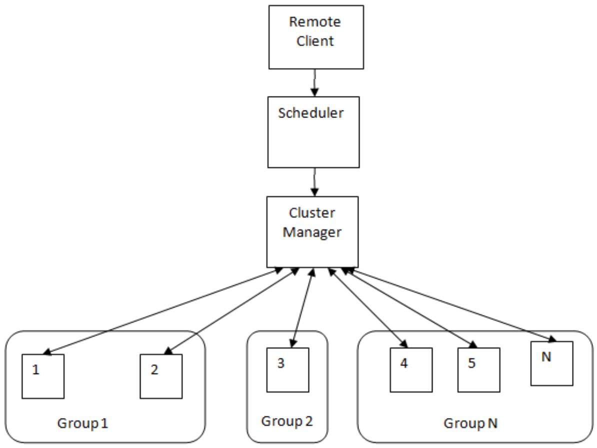

Fig. 1. Proposed system architecture

5. Experimental Setup

Basic requirements

i. OS: Linux.
ii. Cluster Management: OpenMPI.
iii. C Language for coding.

A. Hardware Configuration

Table 1: Hardware Configuration

Node Name	CPU	CORES	SPEED
Node 1	CPU Intel Pentium(R) G3220	2 (RAM 2GB)	3 GHz
Node 2	CPU Intel Pentium (R) G3220	2 (RAM 2GB)	3 GHz
Node 3	CPU Intel Pentium(R) G3220	2 (RAM 2GB)	3 GHz
Node 4	Intel Core(TM)(R) G3220	4 (RAM 4GB)	3.40 GHz

B. Software Configuration

As shown in the figure1, the Grid environment is configured with the help of Globus Toolkit. Open source platform like linux is useful to form the Grid environment. Globus Toolkit global schedule helps to receive the job from the remote client and schedules the job according to the policy defined in the Globus Toolkit. Then the job scheduler sends job for resource scheduling and execution in the cluster to the master node.

The master node schedules the computational resources based on the availability through manual intervention by knowing the status of the computational resources.

Table 2: Iteration result for pi value calculation in the grid environment.

		Node and Core Selection													
		Node 1		Node 2		Node3		Node 4							
Sr. No.	Tasks	0	1	0	1	0	1	0	1	2	3	Time	Calculated Value of PI	True value PI	Deviation in Result
1	10	√										0.950206	3.14140576		0.00018689
2	20	√		√								0.972099	3.14144848		0.00014417
3	30	√		√		√						1.030156	3.14131392	3.14159265	0.00027873
4	40	√		√		√		√				1.101187	3.14134984		0.00024281
5	50	√	√	√		√		√				1.418158	3.14138264		0.00021001
6	60	√	√	√	√	√		√				1.666837	3.14138587		0.00020678
7	70	√	√	√	√	√	√	√				2.056263	3.14145151		0.00014114
8	80	√	√	√	√	√	√	√	√			2.85969	3.14146301		0.00012964
9	90	√	√	√	√	√	√	√	√	√		3.432987	3.14139658		0.00019607
10	100	√	√	√	√	√	√	√	√	√	√	3.197035	3.14145029		0.00014236

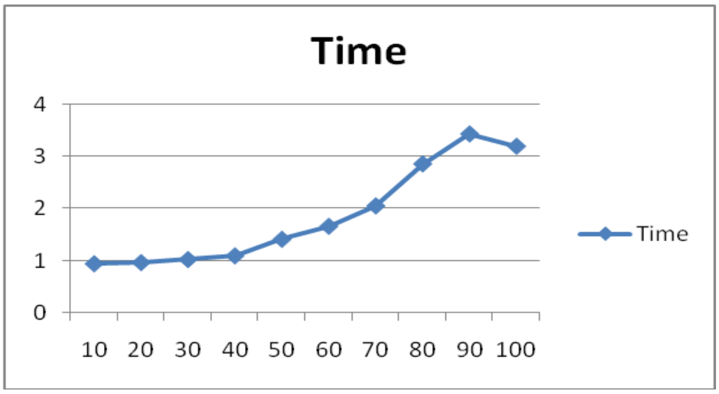

Fig. 2. Execution time to execute no. of task

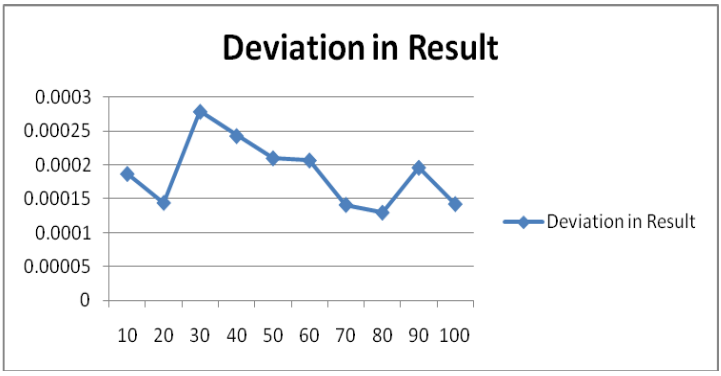

Fig. 2. Deviation in result to execute no. of task

Experiment is performed to calculate pi value using different CPUs with varying configuration shown in table 2. Here execution time is calculated for 50,00,000 iterations for single task. If number of tasks are N then number of iterations are, N*50,00,000. The execution is done in parallel and the average value of pi is calculated by DART algorithm which indicates the level of complexity of task.

6. Conclusion

This paper performs an extensive survey of various load balancing methods that have been suggested over the years for usage in the Grid. The load balancing methods that are available in the literature have been appropriately judging by a different algorithm. The algorithm, research focus, contribution, features, compared model, performance metrics, improvement, gap and future work of each load balancing technique have been analyzed and conferred.

References

[1] Muhammad Ibrahim Munir, Tasneim Aldhanhani, Khalifa Hassan Al Hosani" Control of Grid Connected PV Array using P&OMPPT Algorithm"2017 Ninth AnnuIEEE Green Technologies Conference,

[2] Jindal S., Bansal R. K., Bansal S. Efficient load balancing scheduling technique on Grid Computing. Communication, Information and Computing technology, 2015 International IEEE Conference.

[3] K. Abdelkader, J. Broekhove, Kurt. Vanmechelen Economic based resource management for dynamic computational Grid: Extension to substitutable CPU resources CSDL Home ACCSA 2009 2013 ACS International Conference on Computer Systems and applications.

[4] Sourabh Mandloi, Hitesh Gupta, A Review of resource allocation and Task Scheduling for computational Grids based on Meta-Heuristic Function. International Journal of Research in Computer and Communications Technology Vol2, Issue3, March 2013.

[5] Laxmikant Kale, Sameer Kumar, Mani Potnuru, Jayant DeSouza, Sindhura Bandhakari Faucets: Efficient Resource Allocation on the computational Grid IEEE Transaction Journal PP396-405, 2004.

[6] D. Thilagavathi, Dr. Antony Selvadoss Thanamani. A survey on dynamic job scheduling in Grid Environment Based on Heuristic Algorithms, International Journal of Computer Trends and Technology Volume 3, Issue 4, 2012.

[7] S. Selvi, Dr. D. Manimegalai and Dr. Surnliandi, Efficient Job Scheduling on Computational Grid with Different Evolution Algorithm, International Journal of Networking and applications Vol.3, Issue 2, April 2011.

[8] AshishChandak, Bibhudatta Sahoo, Ashok Kumar Turuk, Heuristic Task Allocation strategies for computational Grid. Int J. Advanced Networking & Applications to 2, Issue: 5 Pages: 804-810, 2011.

[9] Kousalya K. and Balasubramanie P. An Enhanced Ant Algorithm for Grid Scheduling Problem IJCSNS International Journal of Computer Science & Network Security, Vol:8, No:4, April 2008.

[10] A book on Grid computing – Joshy Joseph, Craig Fellenstein IBM Press (Pearson).

[11] Vang-Suk kee, Carl Kesselman, Grid Resource Abstraction, Virtualization and Provisioning for Time Targeted Applications, Eight IEEE International Symposium on cluster computing and the Grid – 2008.l

[12] Lilian Noronha Nassif, Jose Marcos Nogueira, Florio Vinicius de Andrade, Distributed Resource Selection in Grid using Decision Theory, Seventh IEEE International Symposium on cluster computing and the Grid – 2007.

[13] GalstyanA,CzajkowskiK,LermanK.Resourceallocationinthegridwithlearning agents. JGridComput2005;3:91–100.

[14] Ritchie, Levine J. A fast, active local search for scheduling independent jobs in heterogeneous computing environments.In: Proceedings of 22nd workshop of the UK planning and scheduling special interest group.Glasgow;2003:p.178– 83.

[15] Zikos, Karatza HD. Communication cost-effective scheduling policies of non- clairvoyant jobs with load balancing in a grid.J Syst Softw 2009;82:2103–16.

[16] Subrata R, ZomayaAY, Land field t B.Artificial life techniques for load balancing in computational grids. J Comput Syst Sci 2007;73:1176–90.

[17] Li K.Optimal load distribution in the nondedicated heterogeneous cluster and grid computing environments. J Syst Archit 2008;54:111–23.

[18] Deb K, Agrawal S, Pratap A, Meyarivan T.A fast and elitist multi-objective genetic algorithm: NSGA-II.IEEE Trans Evol Comput 2002;6:182–97.