A Flexible Dynamic Load Balancing Model for Independent Tasks in Grid Computing

[I]Seyed Mohsen HashemiGhiri, [II]Abbas Akkasi, [III]Amin Shamohammadi,
[IV]Mohammadnabi Omidvar
[I]Sama Technical and Vocational Training College, Islamic Azad University,
Firoozabad Branch,Firoozabad,Iran.
[II]Computer Engineering Dept. of Islamic Azad University Science and Research Branch,Tehran,Iran.
[III]Islamic Azad University, Larestan, Branch, Iran.
[IV]Dept. of Computer Engineering, Firoozabad Branch, Islamic Azad University, Firoozabad, iran.

Abstract

By the rapid growth of data, distributed computing and grid computing has been attracted so much attention. Grid computing is a distributed parallel processing system that resources have different architectures in it. Large number of computations that are done in a certain time in the grid can't be done by the best supercomputers. The load balancing algorithm is used to increase the efficiency of the grid. The purpose of load balancing algorithms is to use all the resources which are available in the grid. The management of the resources in grid as a process of diagnosis and resource allocation schedules and monitors the resources. Scheduling process directs the tasks to the appropriate resources, to ensure maximum use of them. In this paper for task allocating a probability scheduling algorithm is presented for the purpose of load balancing. In this algorithm Workclass, cost, deadline and herd are considered. Probability load balancing algorithm based on the failure of resources, choses a source that had a better performance in the past, and assigns the task to it. Resources are Rewarded or punished based on their performance. In the load balancing algorithms the tasks migrate from heavy load resources to the light load ones if the tasks are imbalanced.

Keywords

Computing grid, load balancing, scheduling, Task, resource management

1. Introduction

Nowadays, supercomputers that performed all the computational tasks are replaced by networks of small computers. The purpose of producing a network is to share the equipment such as printers, scanners, processors and information sources. One of the most important resources for sharing, are processors. The purpose of the processor sharing is to increase the throughput, faster computing and solving complex problems. Due to the possibility of using a shared processor and its importance, recently new discussions appeared in the computer computing space which is called "distributed computing". The purpose of the grid is to easily use the existing computing resources for complex calculations by the sites that are geographically distributed. The main idea in the grid is that the participant resources are used by software transparently and reliably.

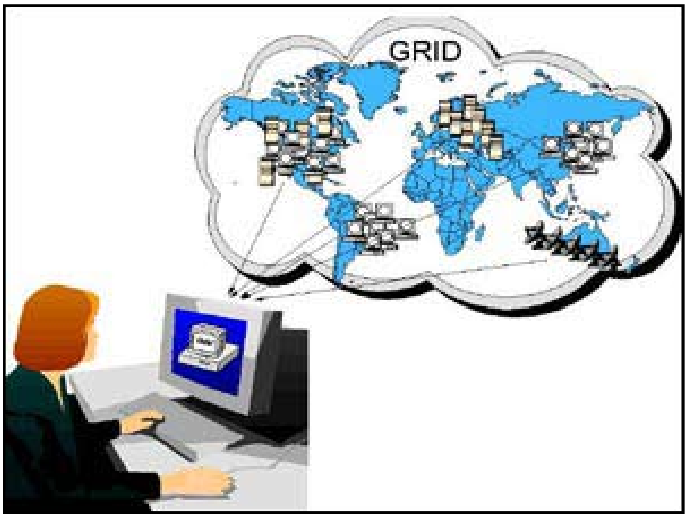

Fig. 1: Grid structure [2]

As a result, grid computing is a model of distributed computing that uses different resources which are geographically distributed [1].In grid computing users have transparent access to computers and data regardless of their position for operating system and other details. Computing grid is an important field that increases the distributed computing [2]. Significant progress is made in [3].In grid failure of a source is common [4] in fact, many resources are available in grid and there is a high probability that some resources fail. There is no boundary between the grids [5-6], the grid can be classified in terms of scale and operational.

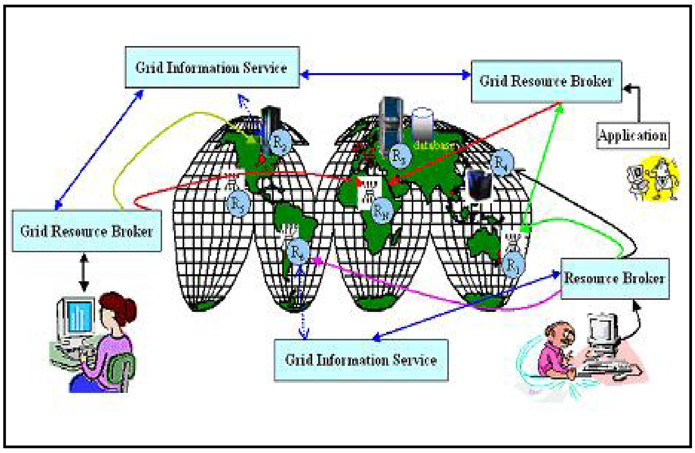

Fig. 2: Workflow in grid [7]

In Figure 2 grid resources have been recorded in one or more grid information service. So the grid environment contains security, information discovery, resource allocation, application development, performance management, density of resources and timing. Software tools and providers that provide a service to connect between data sources to perform a distributed processing are known as the grid middleware [7].

Scheduling components can be divided into categories such as centralized schedule, hierarchical and decentralized. In the centralized organization, only one scheduling controller is existed. In the hierarchical organization, the scheduling controller are organized in a hierarchy. Decentralized organization, is other than the two above cases [1]. The use of load balancing technology is to respond to the user demand, as well as the use of a large number of servers simultaneously. Load balancing avoid the long wait and use of unbalanced resources [8]. Should be noted that some resources, has a heavy load and others have light load or unemployed. As a result, to improve throughput, this load should be divided equally among the available resources. The main purpose of load balancing is optimizing the average response time [9] which often means balance and equality of workload ratio on a system resource.

II. Related Work

Grid resource management can be implemented as a multi agent system along with the ability to discover the resources. A technique already have been used to build a large-scale distributed software system .it was developed by dynamic performance and in its implementation an agent base resource management system (ARMS) was used [10]-[12].The ARMS (Agent base resource management system) is a hierarchical system with heterogeneous factors [13]-[14] and can achieve grid load balancing, so we try to meet the quality of service requirements.

Ant algorithm, instead of wandering randomly, additional load storages are used so that smarter ants can move faster towards regions that are believed to be more heavy load or light load. When an ant wants to make decisions on where to move next, its residence agent will contact all its neighboring agents for latest workload information. An ant always chooses the most heavy load neighboring agent as the next stop. Agents use learning techniques like reinforcement learning [10] [16], learning fuzzy expert systems [17] and other learning methods [18] to manage grid resources.

The path algorithm is a load balancing algorithm and the application workload is distributed on the available computing resources equally. The path algorithm is designed to work on heterogeneous environments with high scalability [45].

Diffusion Algorithms are [19]-[21] category of algorithms for scheduling and load balancing in a dynamic environment. This algorithm assumes that workload is a continuous quantity and the machines are connected as a graph.

Genetic algorithm which is based on Biological evolution is appropriate to search large solution spaces. [45] [20]-[25] these algorithms have been used for dynamic and static scheduling.

Grid computing resources shall perform real-time load forecasting where the results will be returned to each power system for decentralized load balancing operations [26].

Economic methods for the allocation and management of resources is considered [27]-[30] market-based models, satisfy some basic requirements for a grid.

In game theory based methods competition for achieving computational resources and the way of cooperating with other applicants in order to achieve the best solution

Competition based on game theory methods obtain computing resources and how to cooperate with other applicants in order to achieve the best solutions, components, game theory, and it solves the model [31]-[32].

The ability of reinforcement learning is used to allocate resources in computational grid and other heterogeneous computing systems [16] [32]-[34].

In [34] a Stochastic Learning Automaton is presented. Each automata has several probability vectors which each vector is used to choose in global status. In [35] the behavior of the Scheduling has been studied. In [36] a framework for assigning work in heterogeneous computing system is provided.

opportunistic Load balancing [37]-[39] assigns each task to a light load machine ignoring time expected to operate and its purpose is to keep all the machines busy as much as possible.

The minimum execution time algorithm assigns each job to the machine on which it has the least expected run-time, ignoring any other loads on the machines. The goal is to assign the task to the best machine, but it leads to imbalance.

The minimum completion time algorithm assigns each task to the machine on which it has the least expected completion time. The Switching algorithm uses the MCT and MET in a cyclic fashion depending on the load distribution across the machines. The purpose is to have a heuristic with the desirable properties of both the MCT and the MET. The SA heuristic begins mapping tasks using the MCT heuristic until the value of the load balance index increases. After that, the SA heuristic begins using the MET heuristic to perform task mapping. This causes the load balance index to decrease. [40]-[41]

Min-Min algorithm set, chooses the work/machine pair which has the minimum completion time and assigns the task to [42]. The max-min algorithm set, chooses the work/machine pair which has the maximum completion time and assigns the task to [42].

Duplex algorithm is [37]-[38] min-min and max-min algorithm combination. Duplex algorithm with execution of both algorithms chooses the best response as the final solution.

Genetic algorithms (GAs) are a promising heuristic approach to finding near-optimal solutions in large search spaces [42]-[44], in [44] a schedule based genetic algorithm has been studied.

III. The METHOD

In grid the more precise resource management leads to the better scheduling and balance. As a result, here we are looking for a flexible model for precise management of our system. Users are looking for the cheapest resource to operate their tasks, in addition to allocating tasks to inexpensive resources should be considered whether the tasks are executable on the resource or not, so Workclass (security & national policy) issues will rise.

In Grid, task scheduling is an important topic, efficient scheduling is a scheduling that can perform tasks before the deadline, with considering execution time, completion time and load balancing in grid. In this paper, a scheduling algorithm in allocating tasks is presented that can consider completion time, execution time, workclass, herd, deadlines ,costs and load balance in grid.

To present the computational grid a general model based on the tree is used in which tree grows by expanding the grid. The model includes hierarchical characteristics, support heterogeneity and scalability independent of the physical architecture.

Hierarchical structure includes different levels and grid can be modeled to this structure in different forms, it means based on grid level trees grow, for this purpose grids can be divided into several different types and different models can be proposed. For example, we can say that the entire grid is connected to a manager, or scheduler. The grid is divided into several sites and each site includes a number of nodes .grid, includes several regions, each region contains several sites, and each site includes a number of nodes. Grid consists of several clusters, each cluster contains several regions and each region includes several domains and each domain consists of several groups and each group contains several sites and each site includes a number of nodes.

Heterogeneity means a grid includes multiple different resources. Scalability is the growth of resources from light to heavy.as a result the process of adding and deleting resources (entities such as computing elements, etc.) is easy. It means process of adding or deleting nodes or sub-trees in the tree is easy.

In section which is independent of physical architecture transformation of grid into tree is uniform. Therefore, one and only one tree matches a grid which is independent of grid complex topology.

In this system, the nodes run tasks. When a new node is added into the system introduces itself to the nearest scheduler and sends information about itself including workload, speed, cost, class and etc to scheduler.

The role of dispatchers in the system is management and scheduling of tasks. Dispatcher should allocate tasks based on load balancing algorithm policy such that load remain balanced. Dispatcher decides to send which job to which node. first tasks are assigned to manager of the grid, then manager assigns tasks to site manager which best fulfils policies and finally site manager executes tasks on its best resource.

Scheduler node is the place where tasks should be transferred to the sources. Therefor tasks should stand in queue and assign based on scheduler decision. Scheduler has a section called “history information database” on which decides to assign tasks to resources. This site includes properties about resources which scheduler uses them to assign tasks to resources. These properties include load information, resource economic cost, and resources class, past status, communication cost and computing power.

Resources and tasks have specific structures. A grid’s resource model includes m resource(R) which each R_j resource includes computational capacity of CP_j, workload LI_j, the communication status of CS_j and resource class of RC_j, the past status PS_j and resource economic costs REC_j. As a result, the following relations can be written for resources.

$R= \{R_j|j=1, 2\dots m\}$
$CP= \{CP_j|j=1, 2\dots m\}$
$LI= \{LI_j|j=1, 2\dots m\}$
$CS= \{CS_j j=1, 2\dots m\}$
$RC= \{|j=1, 2\dots m\}$
$PS= \{PS_j|j=1, 2\dots m\}$
$REC= \{REC_j|j=1, 2\dots m\}$

In the created model, each resource has properties that the model is as follows.

$$R_j=(CP_j, LI_j, CS_j, RC_j, PS_j, REC_j)$$

In modeling of tasks in system, there are n number of tasks (T), and each T_i task includes start time ST_i, the task length TL_i, deadline DL_i‘, task Class TC_i and task economic cost TEC_i which are assigned to appropriate resource by the scheduler. Therefor the following relations can be written.

$T= \{T_i|i=1, 2\dots n\}$

$ST= \{ST_i|i=1, 2\dots n\}$

$TL= \{TL_i|i=1, 2\dots n\}$

$DL= \{DL_i|i=1, 2\dots n\}$

$TC= \{TC_i|i=1, 2\dots n\}$

$TEC= \{TEC_i|i=1, 2\dots n\}$

When introducing tasks some users are concerned about the time of completion, and some other are concerned about the costs of executing their tasks, therefore the user can determine a weight for completion and execution cost of their tasks (W_C,W_t). The weights are in the range [0, 1] and their summation is 1. For example, if for a task W_C is 0.7 this means that the user is 70% concerned about execution financial cost and 30% concerned about the completion time. The task scheduler must find resources to implement the task that improve cost 70% and time 30%. So T_i is modeled as follows:

$$T_i = (ST_i\ , TL_i\ , DL_i\ , TC_i\ , TEC_i,\ W_C,W_t)$$

In the proposed index computing power of resources and task length are considered .tasks are based on instructions and computing power of each resource is based on number of instructions processed per second (service rate).so each resource can be estimated based on the number of instructions processed.

If in task scheduling, load is ignored an imbalance is caused in the system that increases system response time, to solve this problem, cyclic load balancing algorithm is recommended which its purpose is migration of tasks from heavy load resources to light load ones. For load balancing in this structure, two methods are proposed. First load balance is established within computing nodes which is called “inter-site load balancing”, and the other one is load balancing across the sites which is called “intra-site load balancing”.

As a result tree-based hierarchical model is presented for the grid. The tree with the expansion of the grid, grows. The purpose of this model is establishing centralized management and prevention of herd. For the workload of resources, load index and the new resource status based on symmetric neighborhood is presented and policies such as workclass, cost, herd and deadline for the proposed algorithms have been considered. Two types of algorithms are recommended 1. Scheduling algorithm with the purpose of load balancing 2. Load balancing algorithm. The purpose of these algorithms is establishing the load balance in system and decreasing the response time and the percentage of resource failure.

The main purpose of studied load balancing algorithms is decreasing the response time and decreasing the resources failure rate. For the assignment of tasks to resources must obtain the appropriate resources list, then tasks assign to the appropriate resources. Appropriate resources are light load resources and normal resources that meet deadlines, costs and workclass of tasks. In random load balancing algorithm, tasks after meeting workclass conditions, deadlines and costs randomly have been mapped to the resources. In random algorithm, tasks may be assigned to resources with heavy loads, but in random load balancing algorithm, the task after meeting of listed conditions, randomly assigned to a light load resource. This leads to that the balance load to be maintained in the system, but the average response time and tasks failure rate may not improve. In probability load balancing algorithm, scheduling and load balancing use successful execution of tasks on resources. Based on the tree model, scheduler node as manager assigns task based on past status. The past status of resources includes three parts 1. Total tasks assigned to the resource 2. Tasks executed by the resource 3. Tasks which resource not able of execution.

IV. Simulation

To test the system, a number of random resources have been created that these resources have random properties. These properties include resource number, workclass, economic costs, computing power, and resources failure rate. A number of tasks are produced in the system randomly with random propertied. Some of these properties include task number, time of arrival, the number of instructions, deadlines, economic costs, workclass, cost factor, time factor. Tasks are being assigned to resources based on algorithm policy. All algorithms are evaluated by identical simulator, operating system and hardware.

To evaluate the proposed algorithms several effectiveness index including average Response Time (ART), execution time (ET), scheduling cost saving percent, percent of normal resources in system, task failure percent have been considered.

Simulated and studied algorithms are as follows 1.scheduling algorithms with purpose of load balancing which include random algorithms, random load balancing algorithms, minimum load algorithms, probability load balancing algorithm, minimum execution time algorithm, minimum execute load balancing algorithm, minimum execute probability load balancing algorithm, minimum completion time algorithm, minimum complete load balancing algorithm, minimum complete probability load balancing algorithm, minimum cost algorithm, minimum cost load balancing algorithm, minimum cost probability load balancing algorithm, minimum complete and cost algorithm, minimum complete and cost load balancing algorithm, minimum complete and cost load balancing probability algorithm.2.load balancing algorithms that include non-load balancing algorithm and load balancing algorithm.

In figure (3) average response time has been illustrated. The less average response time the better algorithm operation. In figure (4) total execution time, in figure (5) cost saving percent, in figure (6) task failure percent and in figure (7) normal resources percent in system has been shown.

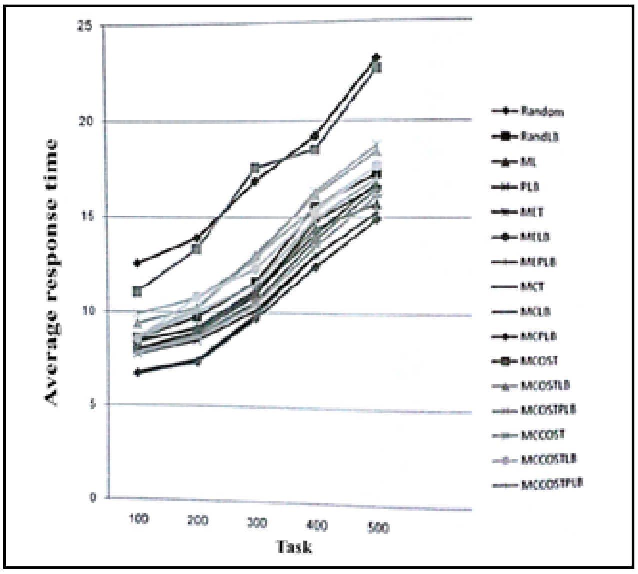

Fig. 3: Average response time of algorithms

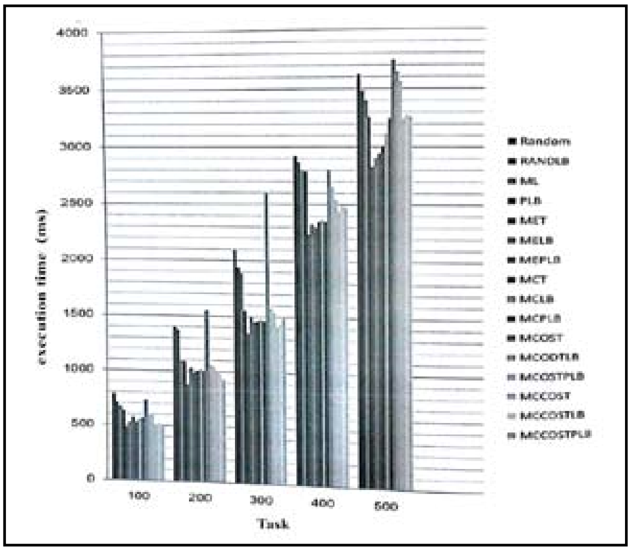

Fig. 4: The total execution time of algorithms

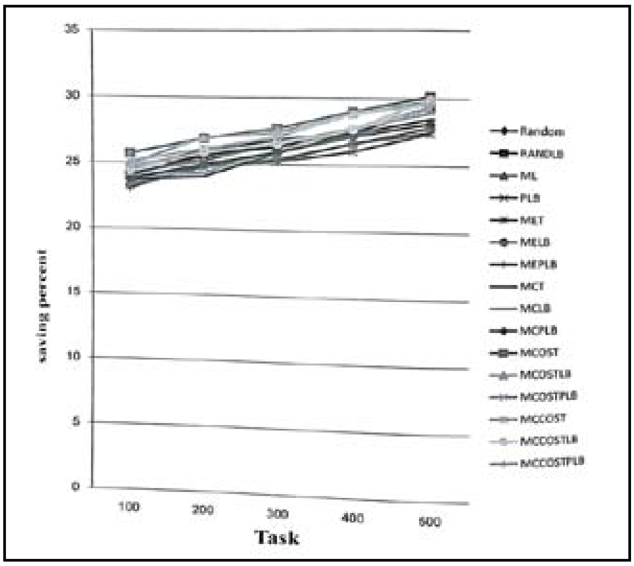

Fig. 5: Task saving percent of algorithms

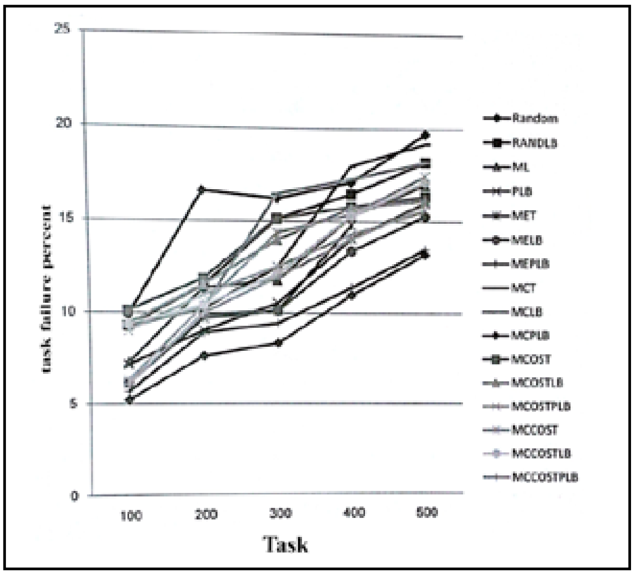

Fig. 6: Task failure percent of algorithms

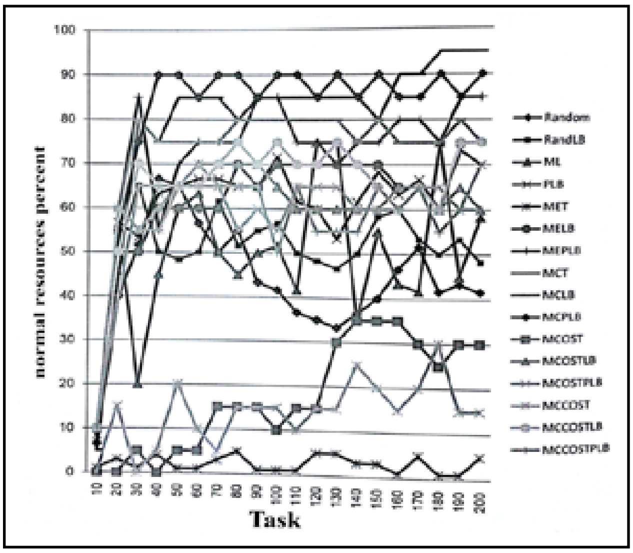

Fig. 7: Normal resources percent in the system.

Two algorithms are discussed in 1. Scheduling algorithm with purpose of load balancing 2. Load Balancing Algorithm. Load balancing algorithm can be divided into five categories: 1.minimum load algorithms 2.minimum execution time algorithms 3. Minimum completion time algorithms 4.minimum cost algorithms 5.minimum cost and completion time algorithms.

In the evaluation of first category, the PLB algorithm improved all indexes and the load balance in the system. In the evaluation of the second group, MEPLB algorithm improves response time and failure percent but operates as MET algorithm in execution time. In the evaluation of third group algorithms, algorithms MCPLB improves response time and failure percent, but operates as MCT in execution time index. In the evaluation of forth group algorithms, MCOSTPLB improves response time and failure percent but operates as MCOST in cost index. In the evaluation of fifth group algorithms, MCCOSTPLB improves response time and failure percent, but operates as MCCOST in cost index.

Simulation results show that probability algorithms operate well and this algorithms decrease response time and task failure percent and establish load balance in system. finally all algorithms were compared which the best algorithm considering response time is MCPLB, considering execution time is MET, considering cost is MCOST, considering task failure percent is MCPLB and considering normal resources in system is probability algorithms. in second type of proposed algorithms it is assumed that task distribution in system is done ignoring load control and leads to load imbalance in system.to establish load balance task migration from heavy load resources to light load ones is used. Average response time index is used for this algorithm which decreases response time very well.

V. Conclusions

The aim of this paper is to present a flexible model for independent tasks in the computational grid. Different algorithms are presented for scheduling and load balancing and the purpose is to achieve load balance and minimizing the average response time and the task failure percent. One of the key issues in grid is scheduling and load balancing. The purpose of this paper is the load balancing of the grid load. In this paper, a general model based on tree is used to illustrate computational grid in which the tree level grows by expanding the grid; this structure is hierarchical and makes scheduler to completely manage resources and this management prevents herd. load index and the status of the new resources have been proposed based on symmetric neighborhood for resources load, also issues such as Workclass, deadlines and costs have been considered.in scheduling algorithms with the purpose of load balancing modern probability algorithms have been presented. in probability method past of resources are being used and task is assigned to the resource operated more successful. It means to execute the tasks in future we experience lower rate of failures. First of all, algorithms are divided into five categories1. Load balancing algorithms 2. The minimum execution time algorithms 3. Minimum completion time algorithms 4.minimum cost algorithms 5. Minimum cost and completion time algorithms.

To evaluate the algorithms, different indexes were studied, that the most notable ones are the normal resource percent in the system, the response time and the percentage of failed tasks and the results show that probability algorithms in distributing tasks work in a way that the resources stay in the normal resource region. In addition to load balancing in the system, probability algorithms decrease the average response time and the task failure percent. The second work that was done was load balancing algorithm. This algorithm migrate the tasks when the system is experiencing imbalances. This algorithm is compared with the system which works normally. The evaluation results show that the load balancing algorithm reduces the average response time in the system.

References

[1] *Akshay Luther, RajkumarBuyya, Rajiv Ranjan, and SrikumarVenugopal, "Peer-to Peer Grid computing and a .NET-based Alchemy Framework", GRID Laboratory, The University of Melbourne, Australia*

[2] *Ian Foster, Carl Kesselman Steven Tuecke, "The Anatomy of the Grid Enabling Scalable Virtual Organizations", Intl J. Supercomputer Application, 2001*

[3] *The Gridlab Project, http://www.gridlab.org*

[4] *Ian Foster, Argonne National Laboratory & University of Chicago, "what is the Grid? A Three Point Checklist".*

[5] *Hans-Ulrich Heiss and Michael Schmitz "Decentralized Dynamic Load Balancing The Particles Approach".*

[6] *Scheduling and Resource Management in Computational Mini-Grids,July 1,2002.*

[7] *Foster I,KesselmanC,Tuecke S, "The Anatomy of the Grid", Intl J. Supercomputer Application, 2001*

[8] *ShijueZ, WannengS, GuangdongC,"a Load Balancing Method Based on Campus Grid"*

[9] *Javier Bustos Jimenez, Robin Hood: An Active Objects Load Balancing Mechanism for Intranet.*

[10] *D.Vengero, and H.R Berenji, "Adaptive coordination among fuzzy reinforcement learning International Conference on Fuzzy Systems (FUZZ-IEEE 02), 2002, PP.179-184.*

[11] *J.Gomoluch, and M.Schroeder, "Information agents on the move: A survey on with mobile agents, "Software Focus Vol.2, No.2, 2001, pp.31-36.load-balancing*

[12] *A.Galstyan,S.kolar, and K.larman, "Resource allocation games with changing resource",Proceedings of International Conference on Autonomous Agents and Multi-Agent systems(AAMAS'03),2003,pp.145-152.*

[13] *J.Cao, S. A. Jarvis et al., "ARMS: An agent-based management system for grid computing, "Scientific Programming, Vol.10, 2002, PP.135-148.*

[14] *J.Cao, D. J. Kerbyson and G.R. Nudd, "Use of agent-based service discovery for resource management in metacomputing*

environment," Proceedings of 7th International Euro-Par Conference, 2001, PP.882-886.

[15] JunweiC,"Self-Organizing Agents for Grid Load Balancing"

[16] J.Parent, K. Verbeeck, and J. Lemeire, "Adaptive load balancing of Parallel applications with reinforcement learning on heterogeneous networks," Proceedings of the International Symposium on Distributed Computing and Applications to Business. Engineering and Science (DCABES'02), 2002, PP.16-20

[17] A. E. El-Abd, "Load Balancing in distributed computing systems using fuuzy expert systems," Proceedings of international conference of Modern Problems of Radio Engineering, Telecommunications and computer Science, 2002, pp.141-144.

[18] G.Cybenko, G. Jiang and D.Bilar,"Machine learning applications in grid computing," Proceedings of 37th Annual Allerton conference on communication, control, and computing, 1999, PP.348-357.

[19]. A. Cortes,M. A.Senar, F code and E. Luque,"On the stability of a distributed dynamic load balancing algorithm, "proceeding of the IEEE international conference on parallel and distributed system,1998,pp.435-446.

[20]. J.M.Ccorrea and A.C.Melo,"USING a classifier system to improve dynamic load balancing, 'Proceedings of the 30the international conference on parallel processing (ICPP), 2001, PP.411-416.

[21] J.Baumgartner D.J.Cook and B.Shirazi,"Genetic solutions to the load balancing problem," proceeding of ICPP95 Workshop.1995, pp.72-76.

[22]. K.DUSSA-Zieger and M.Schwehm, 'Scheduling of parallel programs on configurable multiprocessors by genetic algorithms," Journal of Approximate Reasoning, Special issue on 'Approximate methods in scheduling,"Vol.19, No. (1-2), 1998, pp.23-38.

[23]. M.Grajcar,"Genetic List scheduling algorithm for scheduling and allocation on a loosely coupled heterogeneous multiprocessor system."Proceedings of the 36th design and Automation conference (DAC), 1999, PP.280-285.

[24]. F.E.Sandes and G.M.Megson,"Improved static multiprocessor Scheduling using cyclic task graph – a genetic algorithm approaches,"IPP'97 Workshop on Randomized Parallel Computing, 1997.

[25]. W.H.Woo,S. B. Yang and S.D.Kim, and T. D. Han, "Task scheduling in distributed Computing system with a genetic algorithm ,"Proceeding of High-Performance Computing on the information superhighway(WPC-Asia'97),1997,pp.301-305.

[26] Al-Khannak.r,Bitzer.b, "Load balancing for distributed and Integrated power system using Grid Computing"

[27]. R. Buyya, J.Giddy, and D.Abramson. "An evaluation of economy-based resource trading and scheduling on computational power grids for parameter sweep applications,"2 and international workshop on Active middleware services (AMS'00), 2000.

[28]. J.Gomoluch and M.Schoedder,"Market-based resource allocation for grid computing: A model and simulation,"1st international workshop on middleware for Grid computing MGC2003), 2003.

[29]. R. Buyya and S. Vazhkudai,"Compute power market: towards a market-oriented grid," proceeding of the 1st IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid'01), 2001, pp.574-581.

[30]. D. Abramson, R. Buyya, and J.Giddy, "A computational economy for Grid computing and its implementation in the Nimrod-G resource broker," journal of Future Generation computer systems (FGCS) Vol.18,No.8,2002,pp.1061-1074.

[31]. T. Roughgarden, 'Stackelberg scheduling strategies," Proceeding of the 33rd Annual ACM symposium on Theory of computing (STOC'2001), 2001, PP.104-113.

[32]. A. Galstyan, S.Kolar,and K.Lerman,"Resource allocation games with changing resource," proceeding of international conference on Autonomous Agents and Multi-Agent systems (AAMAA'03),2003,PP.145-152.

[33]. A. Schaerf,Y.Shoham.And M.Tennenholtz,"Adaptive load balancing: A study in learning," journal of Artificial intelligent Research,Vol.2,1995,pp.47-500.

[34]. R. Mirchandaney and j.A. Stankovic,"Using stochastic learning automata for job scheduling in distributed processing system," journal of Parallel and distributed Computing, 1986, pp.527-551.

[35]. R.D. Venkataramana and N.Ranganathan,"Multiple cost optimization for task assignment in heterogeneous computing systems using learning automata," Heterogeneous computing Workshop (HCW'99), 1999, PP.137-145

[36]. T. Kunz," The learning behavior of a scheduling using stochastic learning automata," Technical Report T1-7/91, Institute fur theoreticsche Informatik, Fachbereich Informatik, Technische Hochschule Darmstadt,1991.

[37]. R. F. Freund,M. Gherrity,S. Ambrosius,M. Campbell,M. Halderman,D. Hhengsen,E. Keith,T. Kidd,M. Kussow,J. D. Lima,F. Mirabile,L. Moore,B. Rust, and H. J. Siegel," scheduling resources in multi-user , Heterogeneous, computing environments with smartNet,"proceedings of the 7th IEEE Heterogeneous computing Workshop (HCW'98),1998,pp.184-199.

[38]. R. Armstrong,D. Hensgen,and T. Kidd, "The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions," proceeding of the 7th IEEE Heterogeneous computing Workshop(HCW'98),1998,PP.79-87.

[39]. R. F. Freund and H. J. Siegel," Heterogeneous processing," IEEE Computing, Vol.26, No6, 1993, pp.13-17.

[40]. M. Macheswaran,S. Ali,H. J. Siegel,D. Hensgen,R.F. Freund, "dynamic mapping of a class of independent tasks onto Heterogeneous computing systems," Journal of parallel distributed computing,Vol.59,No2,1999,pp.107-131.

[41] T. D. Braun, H. J. Siegel, and N. Beck, "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," Journal of Parallel and Distributed computing, Vol.61, 2001, PP.810-837.

[42] R,-M. Chen and Y.-M. Huang, "Multiconstraint task scheduling in multi-processor systems by neural network," Proceedings of the 10th IEEE Conference on Tools with Artificial Intelligence, 1998, PP.288-294.

[43] Y. G. Tirat-Gefen and A. C. Parker, "MEGA: An approach to system-level design of application specific heterogeneous multiprocessors," Proceedings of the 5th IEEE Heterogeneous Computing Workshop (HCW'96), 1996, pp.105-117.

[44] L. Wang, H. J. Siegel, V. P. Roychowdhury, and

A.A.Maciejewski,"Task matching and Scheduling in heterogeneous computing environments using a genetic-algorithm-based approach," Journal of Parallel Distributed Computing, Vol.47,NO.1,1997,PP.1-15.

[45] Kuo-Qin Yan, Shun-Sheng Wang, Shu-Ching Wang, Chiu-Ping Chang, "Towards a hybrid load balancing policy in grid computing system", 2009ElsevierLtd

[46] Ramya R and Shalini Thomas. Article: An Optimal Job Scheduling Algorithm in computational Grids. IJCA Special Issue on International Conference on Communication, Computing and Information Technology ICCCMIT (1):12-16, February 2013.

[47] S. Kardani-Moghaddam, F. Khodadadi, R. Entezari-Maleki, and A. Movaghar, "A Hybrid Genetic Algorithm and Variable Neighborhood Search for Task Scheduling Problem in Grid Environment," International Workshop on Information and Electronics Engineering (IWIEE), Procedia Engineering 29, 2012, pp. 3808-3814.

[48] Salman Meraji, M. Reza Salehnamadi, "A Batch Mode Scheduling Algorithm for Grid Computing" J. Basic. Appl. Sci. Res., 3 (4) 173-181, 2013, 2013, Text Road Publication

[49] P. Keerthika and N. Kasthuri, "An Efficient Grid Scheduling algorithm with Fault tolerance and User Satisfaction", Mathematical problems in Engineering, Volume 201. 3(2013), Article ID 340294.