Towards a Reference Model for Surveying a Load Balancing

Asmaa Y. Hamo and Alaa A. Saeed

University of Mosul collage of computer science and mathematics, Mosul, Iraq.

Summary
In recent years, the complexity in distributed system increases, which increases the need to the process of sharing resources provided by the computers of the network. The process of sharing called the load balancing, during which the tasks are transferred from heavy loaded to lightly loaded computers. This leads to increase the speed of tasks execution and reduces the response time, which leads to increase and improve the system performance. In this research we present a survey on load balancing, by explaining its resources that need to be balanced, algorithms that is used in implementing it, polices that the algorithm depend on, and metrics for performance evaluation.

Key words:
Static load balancing, dynamic load balancing, tasks, resources, performance metrics.

1. Introduction

Load balancing has always been an issue since the emergence of distributed systems. Distributed systems are configured by connecting a large number of computers in different places. In these systems the resources of each computer can be shared with other computers for many reasons such as: increase response time, fault tolerant, and enhance the overall performance. The process of sharing resources can be achieved through load-balancing systems. The term "load balancing" refers to the technique that tries to distribute work load between several computers, network links, CPUs, hard drives, or other resources, in order to get optimal resource utilization, throughput, or response [2]. A good load balancing scheme needs to be general, stable, scalable, and minimize the overhead of the system.[3].

Tasks arrive to each computer in the grid, and the number of the arrived tasks differ from one computer to another. In term of load balancing these tasks should be balanced . Tasks need to be balanced can be classified into three main type: CPU-intensive tasks, memory-intensive tasks, and I/O-intensive tasks[1].

Tasks are CPU-intensive if they spend most of their time doing calculation in the processor (e.g. video game and 3D CAD animation). Tasks are memory-intensive if they spend most of their time in the memory (e.g. graph embedding). And finally if the tasks spend most of their time doing I/O operation then these tasks are I/O- intensive (e.g. archiving of raw and processed remote sensing data, and multimedia and web-based applications [46].

Manuscript received February 5, 2013
Manuscript revised February 20, 2013

In general, and before we survey the load balancing approaches, we need to distinguish between three terms: system, architecture, and algorithm from the load balancing systems point of view. By the system we mean things that have both architecture and algorithm, by the architecture we mean the main component contained in the system, and by the algorithm we mean how the architecture's component operate.

This paper present the most important issues in the load balancing systems which are: resources need to be balanced, type of load-balancing algorithms, polices used in implementing load-balancing systems, and metrics used to evaluate the performance.

2. The resources to be balanced

As known the distributed systems has many things to be balanced such as :

- Computer processor time resource: the most important resource in operating system is the CPU-time. When distributed system is used the CPU-time need to be balanced. Some of the research made a CPU load balancing such as[4][5][6].
- Computer memory resource: is another important resource in the computer , also when these computers are connected across the grid, this resource need to be balanced to improve its performance. some of the research made load balancing on the memory such as [7][8] .
- Computer I/O resources: balancing the load of CPU and memory is not enough in some time, so the I/O resources which is depends on the effective usage of storage, in addition to that of CPU and memory, need to be balanced. Some of the research made a load balancing in this resources such as [47][48].

3. Load balancing issue

Load balancing can be mainly categorized as static and dynamic. Static load balancing uses a priori knowledge of the applications and statistical information about the system such as[12][13][14][15]. Dynamic load balancing base their decision making process on the current state of the system such as [16][17][18][19].

However, we can demonstrate the static and dynamic load balancing independently in the following sections:

3-1 Static load balancing

The static load balancing, assigns a given job to a fixed processor or node. Every time the system is restarted, the same binding task-processor (allocation of a task to the same processor) is used without considering changes that may occur during the system's lifetime. Moreover, static load distribution may also characterize the strategy used at runtime, in the sense that it may not result in the same task-processor assignment, but assigns the newly arrived jobs in a sequential or fixed fashion. For example, using a simple static strategy, jobs can be assigned to nodes in a round-robin fashion so that each processor executes approximately the same number of tasks.[20]
Static load balancing can be classified into two types as stated in [21], deterministic and probabilistic as shown in fig.1.

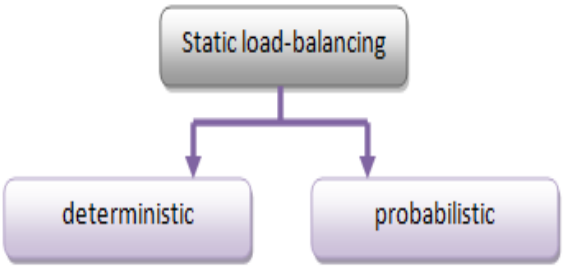

Fig. 1 the first classification of the static load balancing.

- In deterministic load balancing: the jobs are joined to the shortest workstation queue since routing decisions are based on system state.
- In probabilistic load balancing: scheduling policies are described by state independent branching probabilities. Jobs are dispatched randomly to workstations with equal probability.

But [3][22.] classify the static load balancing in a different manner, as shown in fig.2.
By distributed he means that the work involved in making decisions is distributed among many decision makers. By centralized he mean that there are only one decision maker or the common decision of many cooperating decision makers is made in a centralized way.
Also, in this classification, the distributed and centralized load balancing are subdivided into cooperative, non-cooperative and global, by cooperative he means that the decision makers (e.g. users, processors) cooperate between each other, but in non-cooperative the decision makers make their decisions non-cooperatively. By the global he say that the Global scheduling is the problem of deciding where to execute a process, and the job of local scheduling is left to the operating system of the processor to which the process is ultimately allocated.

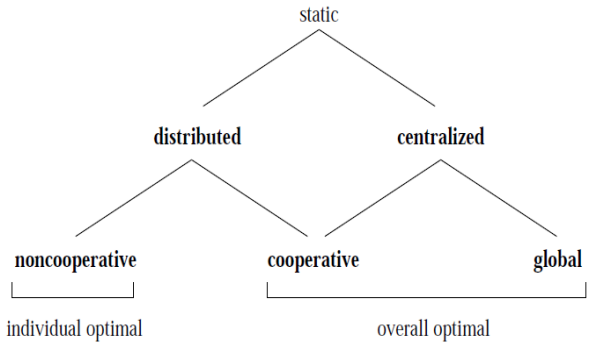

Fig.2 The second classification of the static load balancing.

3-2 Dynamic load balancing

The dynamic load balancing takes into account that the system parameters may not be known beforehand and therefore using a fixed or static scheme will eventually produce poor results. A dynamic strategy is usually executed several times and may reassign a previously scheduled job to a new node based on the current dynamics of the system environment.[20]
This type of load balancing can be explained in more detail by providing a new classification which contain four main parts: control, policies, algorithms and metrics as shown in fig.3.

3.2.1 The control issue

There are two main type of the controlling issue as stated in [23] [24][25]: distributed, and non-distributed, Non-distributed has two types: centralized or semi-distributed.
the system is "distributed" if the load balancing algorithm is executed by all nodes in the system and the responsibility of load balancing is shared among them. The system is "centralized" if the load balancing algorithm is only executed by one node of the distributed system: the central node. Otherwise, if the system nodes are segmented into clusters, and the load balancing within each cluster is centralized; a central node is nominated to take charge of load balancing within this cluster, then the developer said that his system is "a semi-distributed" system.
The distributed load balancing algorithm can be further classified into two type: cooperative and non-cooperative. In cooperative algorithm, processes work together toward a common system-wide global balance. Scheduling decisions are made after considering their effects on some global effective measures (for example, global completion time). On the other hand, in the non-cooperative algorithm each node has autonomy over its own resource scheduling. That is, decisions are made independently of the rest of the system and therefore the node may migrate or allocate tasks based on local performance.[20]

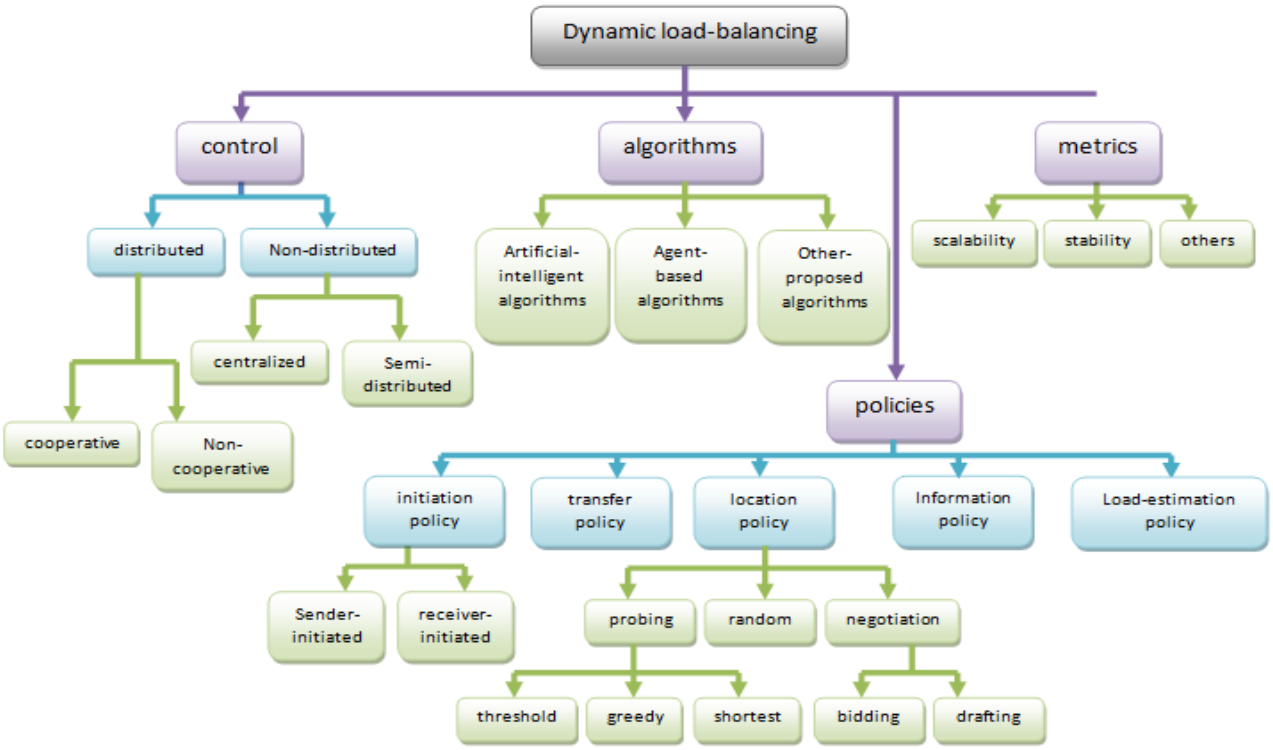

Fig. 3 A classification of dynamic load-balancing.

3.2.2 The dynamic load balancing polices
In this section we demonstrate the policies used in the load balancing algorithm. There are number of main polices in dynamic load balancing[23][26][9][27]:
3.2.2.1 Information policy
Is the part of a dynamic load balancing responsible for collecting information about nodes in the system. It is responsible for providing location and transfer strategies at each node with the necessary information needed to make their load balancing decisions.

Initiation Policy
determines who starts the load balancing process. The process can be initiated by an overloaded server (called sender-initiated) or by an under-loaded server (called receiver-initiated).

Transfer policy
determines when job reallocation should be performed and which job(s) (i.e., client requests) should be reallocated. Job reallocation is activated by a threshold-based strategy. In a sender-initiated method, the job transfer is invoked when the workload on a server exceeds a threshold. In a receiver-initiated method, a server starts the process to fetch jobs from other servers when its workload is below a threshold. The threshold can be a pre-defined static value or a dynamic value that is assessed at runtime based on the load distribution among the servers. When job reallocation is required, the appropriate job(s) will be selected from the job queue on a server and transferred to another server.

3.2.2.4 location policy
The location policy determines which processor is selected for a potential process migration. Some of the approaches used to select a destination node for a transferred job are: random, probing and negotiation, as stated in [23].further, the probing approach has three location strategies: threshold [49], greedy [50], and shortest [49].and the negotiation has two location strategies: bidding, e.g., [51] and drafting, e.g.,[52].

3.2.2.5 Load estimation policy
which determines how to estimate the workload of a particular node of the system, and if there is any tool used in this estimation.

3.2.3 Algorithms used in implementing the system
Each load balancing system has its own algorithm, this algorithm is followed in the process of building the load balancing system. there are many type of these algorithms such as:

- Artificial intelligent-based algorithms: such as genetic algorithms used in [28][29][30], fuzzy concept used in [31][32],and ant algorithms used in [33][34] [35].

- Agent-based algorithm: such as those used in [36][37][38][39].
- Hybrid of both types can be formulated to give better performance, such as using agent system with genetic algorithm e.g. [40][41],or agent system with ant's algorithm e.g. [42][43].

3.2.4 Metrics used for performance evaluation of load-balancing algorithms

There are a number of metrics that can be used to demonstrate the robustness points in the load balancing algorithm and to evaluate its performance, these metrics is as followed [44][26][45]:

- Load balancing time – Amount of time that elapses between the job arrival time and the time at which the job is finally accepted by a node.
- Make-span – Make-span is the total completion time taken to allocate all tasks to a resource. It is the measure of the throughput of the grid.
- Average resource utilization rate – This means the usage of all the resources in the grid.
- Scalability – It is the ability of the algorithm to perform load balancing for a grid with any finite number of nodes.
- Fault tolerance – It is the ability of the algorithm to perform uniform load balancing in spite of arbitrary node or link failure.
- Reliability –This factor is related with the reliability of algorithms in case of some machine failure occurs.
- Stability – Stability can be characterized in terms of the delays in the transfer of information between processors and the gains in the load balancing algorithm by obtaining faster performance by a specified amount of time.
- Availability: Even when servers within the cluster fail, the remaining servers are performing normally
- Processor Thrashing: Processor thrashing occurs when most of the processors of the system are spending most of their time migrating processes without accomplishing any useful work in an attempt to properly schedule the processes for better performance.

3. Conclusion

The purpose of this paper was to present a classification of the load-balancing system, which generally classified as static and dynamic, and then we classified these two types into its main components. In dynamic load-balancing part we present a new classification, though which we demonstrate the control issue, algorithms, polices, and metrics.

References

[1] K. Qureshi, B. Majeed , J. H. Kazmi, S. A. Madani, "Task partitioning, scheduling and load balancing strategy for mixed nature of tasks",Springer Science + Business Media, LLC 2011.

[2] S. Salleh and A.Y. Zomaya." Scheduling in Parallel Computing Systems: Fuzzy and Annealing Techniques". The Springer International Series in Engineering and Computer science, 1999..

[3] D. GROSU, "LOAD BALANCING IN DISTRIBUTED SYSTEMS: A GAME THEORETIC APPROACH", Ph.D. Dissertation in Computer Science, The University of Texas at San Antonio, pp. 1-133., 2003.

[4] A. Chandak, A. Kanfade, A. Joshi, "Dynamic Load Balancing of Virtual Machines using QEMU-KVM", International Journal of Computer Applications, Vol. 46– No.6, pp. 0975 – 8887, May 2012.

[5] G. Dittmann, A. herkersdorf, " Network Processor Load Balancing for High-Speed Links", Proceedings of the Int'l Symp. Performance Evaluation of Computer and Telecomm. Systems (SPECTS), 2002.

[6] E. Altman, U. Ayesta, B. Prabhu, "Load Balancing in Processor Sharing Systems", Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools, No. 12, October 20, 2008,

[7] G. CYBENKO, "Dynamic Load Balancing for Distributed Memory Multiprocessors", JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING No.7, pp. 279-301, 1989.

[8] S. Marchesin, C. Mongenet, J. M. Dischler, "Dynamic Load Balancing for Parallel Volume Rendering", Proceedings of the Eurographics Symp. Parallel Graphics and Visualization 2006 .

[9] J. Cao, Y. Sun, X. Wang, S. K. Das, "Scalable Load Balancing on Distributed Web Servers Using Mobile Agents", J. Parallel Distributed Computer, Vol. 63, pp. 996-1005, 2003.

[10] H. Li, "Load Balancing Algorithm for Heterogeneous P2P Systems Based on Mobile Agent", Electric Information and Control Engineering (ICEICE),pp. 1446 – 1449,2011.

[11] N. Handigol, S. Seetharaman, M. Flajslik, A. Gember, N. McKeown, G. Parulkar, A. Akella, N. Feamster, R. Clark, A. Krishnamurthy, V. Brajkovic and T. Anderson," Aster*x: Load-Balancing Web Traffic over Wide-Area Networks", 2009.

[12] C. Kim and H. Kameda, "An Algorithm for Optimal Static Load Balancing in Distributed Computer Systems", IEEE Transaction Computer ,vol. 41, no. 3, pp. 381-384, March 1992.

[13] A. N. Tantawi and D. Tawsley, "Optimal Static Load Balancing in Distributed Computer Systems," J. of Association Computer, vol. 32, no. 2, pp. 445-465, April 1985.

[14] X. Tang and S.T. Chanson, "Optimizing Static Job Scheduling in a Network of Heterogeneous Computers," Proceedings of the International Conference on Parallel Processing, pp. 373-382, August 2000.

[15] S. H. Bokhari, "Dual Processor Scheduling with Dynamic Reassignment," IEEE Transaction of Software Engneering, vol. SE-5, no. 4, pp. 341-439, July 1979

[16] A. Karimi, F.Zarafshan, A.b.Jantan, A.Ramil, M.Iqbal and b. Saripan, " A New Fuzzy Approach for Dynamic Load Balancing algorithm", International Journal of Computer Science and Information Security, Vol. 6, No.1 2009.
[17] S. Surana, B. Godfrey, K. Lakshminaraynan, R. Karp and I Stoica, "Load Balancing in dynamic structured Peer to Peer system", IEEE, pp. 2253-2262, 2004.
[18] K. Son, S. Chong and G. "Dynamic Association for Load Balancing and Interference Avoidance in Multi-Cell Networks", IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 8, NO. 7, pp. 3566-3576, JULY 2009.
[19] D. Cedermany and P. Tsigasz ,"On Dynamic Load Balancing on Graphics Processors", Proceedings of the 23rd ACM SIGGRAPHEUROGRAPHICS symposium on Graphics hardware, pp. 57-64, 2008.
[20] J. Ghanem, "Implementation of Load Balancing Policies in Distributed Systems", M.S. thesis , Electrical Engineering, University of New Mexico, 2004.
[21] H. D. KARATZA, "A COMPARISON OF LOAD SHARING & JOB SCHEDULING IN A NETWORK OF WORKSTATIONS", I.J. of SIMULATION, Vol. 4 No 3-4, 2003.
[22] T. Casavant and J. G. Kuhl, "A taxonomy of scheduling in general-purpose distributed computing systems". IEEE Transaction of Software Engineering, 14(2), pp. 141–154, February 1988.
[23] A. M. Alakeel, "A Guide to Dynamic Load Balancing in Distributed Computer Systems", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.
[24] A. B. Saxena and D. Sharma , "ANALYSIS OF THRESHOLD BASED CENTRALIZED LOAD BALANCING POLICY FOR HETEROGENEOUS MACHINES", International Journal of Advanced Information Technology (IJAIT), Vol. 1, No.5, October 2011.
[25] A. Singh, M. Hemalatha , "An Approach on Semi-Distributed Load Balancing Algorithm for Cloud Computing System", International Journal of Computer Applications, Volume 56– No.12, pp. 0975 – 8887, October 2012.
[26] A. A. Rajguru, S. S. Apte, "A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters", International Journal of Recent Technology and Engineering (IJRTE) ISSN: Vol. 1, No. 3, pp. 2277-3878, August 2012.
[27] A. L. Blais, "EVALUATION OF LOAD BALANCING ALGORITHMS AND INTERNET TRAFFIC MODELING FOR PERFORMANCE ANALYSIS", M.S. thesis, University of Colorado, Science Department of Computer Science, 2000.
[28] A. Y. Zomaya and Y. Hwei, "Observations on Using Genetic Algorithms for Dynamic Load-Balancing", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 12, NO. 9, SEPTEMBER 2001.
[29] B. Sahoo1, S. Mohapatra and S. K. Jena, " A Genetic Algorithm Based Dynamic Load Balancing Scheme for Heterogeneous Distributed Systems", Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, 14-17, Vol. 2., pp. 499-505 July 2008.
[30] W. A. Greene, "Dynamic Load-Balancing via a Genetic Algorithm", Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence , pp. 121–129, 2001.
[31] R. Bhardwaj, V. S. Dixit and A. Upadhyay, "A Fuzzy Intra-Clustering Approach for Load Balancing in Peer-to-Peer System", Journal of Information and Computing Science Vol. 7, No. 1, , pp. 019-024, 2012.
[32] Y. Kwok and L. Cheung, "A new fuzzy-decision based load balancing system for distributed object computing", J. Parallel Distributed Computer, Vol. 64, pp. 238–253,2004.
[33] S. SURYADEVERA, J. CHOURASIA, S. RATHORE and A. JHUMMARWALA, "Load Balancing in Computational Grids Using Ant Colony Optimization Algorithm", International Journal of Computer & Communication Technology (IJCCT) ISSN (ONLINE), Vol.3, No.3, 2012.
[34] R. Mishra1 and A. Jaiswal, " Ant colony Optimization: A Solution of Load balancing in Cloud", International Journal of Web & Semantic Technology (IJWesT), Vol.3, No.2, pp. 33-50, April 2012.
[35] H. Une and F. Qian, " Network Load Balancing Algorithm using Ants Computing", Proceedings of the IEEE, WIC International Conference on Intelligent Agent Technology (IAT'03),2003.
[36] M. A. Salehi and H. Deldari, "A Novel Load Balancing Method in an Agent-based Grid", IEEE, 2006.
[37] R. Ezumalai, G. Aghila and R. Rajalakshmi, "Design and Architecture for Efficient Load Balancing with Security Using Mobile Agents", IACSIT International Journal of Engineering and Technology, Vol. 2, No.1, February 2010.
[38] N. Nehral, R.B.Patel, V.K.Bhat, "A Multi-Agent system for Distributed Dynamic Load Balancing on Cluster", Enformatika Transactions on Computing, Engineering and its Applications, Vol. 14, 264-269, Aug.2006.
[39] M.Aramudhan, S. Karthikeyan, K.Mohan and V.R. Uthaiaraj, "ELDMA: Enhanced Load balancing Decision making using Decentralized Mobile Agent Framework", Proceedings of the International Conference on Computer and Communication Engineering, pp. 11 – 14, May 2008.
[40] T. S. N. Htwe , A. Thida and P. P. Nyunt, "Mobile Agents Based Load Balanced Resource Scheduling System", IEEE, 2011.
[41] J. Cao, D. P. Spooner, S. A. Jarvis, and G. R. Nudd, "Grid Load Balancing Using Intelligent Agents", Future Generation Computer Systems ,pp. 135-149, 2005.
[42] A. Montresor, H. Meling, and O. Babaoglu, " Messor: Load-Balancing through a Swarm of Autonomous Agents", Proceedings of the International Workshop on Agents and Peer-to- Peer Computing in conjunction with AAMAS , Volume 2530, 2003, pp. 125-137, 2003.
[43] J. Cao, "Self-Organizing Agents for Grid Load Balancing", Proceedings of 5th IEEE/ACM International Workshop on Grid Computing, pp. 388-395, November 2004.
[44] S. Kumar and N. Singhal, "A Study on the Assessment of Load Balancing Algorithms in Grid Based Network ", International Journal of Soft Computing and Engineering (IJSCE) , Vol. 2, No. 1, March 2012.
[45] A. Diasse, F. Kone, " Dynamic-Distributed Load Balancing for Highly-Performance and Responsiveness Distributed-

GIS (D-GIS)", Journal of Geographic Information System, Vol. 3, No. 2, pp. 128-139, 2011,.

[46] S. Lakra, "LOAD BALANCING TECHNIQUES FOR I/O INTENSIVE TASKS ON HETEROGENEOUS CLUSTERS", M.S. thesis SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS ,Computer Science & Engineering, 2007.

[47] L. Xiao, Y. Qu , and X. Zhang, "A dynamic load balancing scheme for i/o-intensive applications in distributed systems", Proceedings of the International Conference on Parallel Processing Workshops (ICPPW03), pp. 431–438, 2003.

[48] X. Qin , H. Jiang , Y. Zhu , and D. Swanson, "Dynamic load balancing for i/o-intensive tasks on heterogeneous clusters", Proceedings of the International Conference on High Performance Computing (HiPC03)., Dec 2003.

[49] D.L. Eager, E.D. Lazowski, and J. Zahorjan, "Adaptive Load Sharing in Homogeneous Distributed Systems," IEEE Trans. Software Eng., vol. SE-12, no. 5, pp. 662-675, May 1986.

[50] S. Chowdhury, "The Greedy Load Sharing Algorithms," J. Parallel and Distributed Computer, vol. 9, pp. 93-99, May 1990.

[51] J. A. Stankovic and I. S. Sidhu, "An Adaptive Bidding Algorithm for Processes, Cluster and Distributed Groups," Proceedings of the 4th International Conference Distributed Computer System, pp. 49-59, 1984

[52] L. Ni, C. Xu, and T. Gendreau, “A Distributed Drafting Algorithm for Load Balancing,” IEEE Transactions on Software Engineering, Vol. SE-11, No. 10, pp. 1153-1161, October 1985.