REPORT ON HIERARCHIAL LOAD BALANCING TECHNIQUE IN GRID ENVIRONMENT

By

NEERAJ RATHORE * INDERVEER CHANA **

* PhD Scholar, Thapar University.
** Associate Professor in Computer Science and Engineering Department, Thapar University, Patiala.

ABSTRACT

Grid computing offers the network with large scale computing resources. Load balancing is effective for balancing the load of large scale heterogeneous grid resources that are typically owned by different organizations. Not all the techniques provide the same benefits for users in utilizing the resources in a quick response time. Similarly, the profit earned by resource providers also differs for different Load balancing technique. We surveyed the Load Balancing and Job Migration technique used in grid computing since its inception until 2013. The author discussed their advantages and disadvantages and analyze their suitability for usage in a dynamic grid environment. To the best of our knowledge, no such survey has been conducted in the literature up to now. A comparative study of some of them along with their pitfalls in case of huge distributed environment, like Grid, is discussed in this paper. The author also proposed efficient hierarchial Load Balancing algorithm to close all the existing gaps.

Keywords: Grid Computing, Grid Sim, Architecture, Load Balancing, Communication Overhead.

INTRODUCTION

The explosion of the internet, the computational power of servers PCs, GPUs and high-speed computer network give an idea of a grid of computational resources [3]. Grid is a vast distributed system that may span many continents. Grid is a huge system that means to build a network of computational resources spreading around the globe and to utilize each small resource that are flexible and combines globally distributed computers and information systems for creating a universal source of computing power and information [1, 2]. Sharing resources among organizational and institutional boundaries need an infrastructure to coordinate resource of boundaries within so-called virtual organizations. Such infrastructure should offer an easy management for the formation of virtual organizations, sharing resources, discovering services and consuming services. Due to uneven task arrival patterns and unequal computing capacities and capabilities, the computers in one grid site may be heavily loaded while others in a different grid site may be lightly loaded or even idle. It is therefore, desirable to transfer some jobs from the heavily loaded computers to the idle or lightly loaded ones in the grid environment aiming to efficient utilization of the grid resources, improve the response time, maximal utilization of available resources and minimize the average job response time. This characteristic of Grid makes Load Balancing one of the critical features of Grid infrastructure. There are a number of factors, which can affect the grid application performance like load balancing, heterogeneity of resources and resource sharing in the Grid environment. This paper, focus on Dynamic Distributed Load Balancing and tried to present the impacts of Load Balancing on grid application performance and introduced the problem of balancing the load among the participant in the Grid.

A Grid can offer a resource balancing effect by scheduling and load balancing of grid jobs at machines with low utilization. A proper load balancing across the grid can lead to improve overall system performance and a lower turnaround time for individual jobs. This is a very crucial concern in distributed environment, to assign fair jobs to resources. The main goal is to distribute the jobs among processors to maximize throughput, maintain stability, and resource utilization. This is achieved by proper load balancing techniques. Load balancing is done basically to provide the following benefits. The basic steps for load balancing [10, 11] are depicted in Figure 1.

- Load balancing reduces mean job response time under job transfer overhead.
- Load balancing increases the performance of each host and.
- Small jobs will not suffer from starvation.

The taxonomy of Grid load balancing [11] is broadly categorized as Classes, Policies/Mechanism, Algorithms, Strategies, Processes, and Challenges as shown in Figure 2 from the top to the bottom; this structure can be identified as what follows this paper. Entire load balancing techniques are compared and an optimal load balancing techniques for Grids is proposed on the basis of best finding technique. The rest of the paper is organized as follows. Section, presents literature survey in which some definitions and concepts, policies and

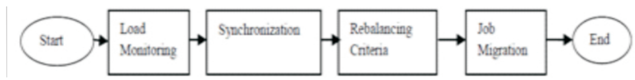

Figure 1. Basic load balancing steps

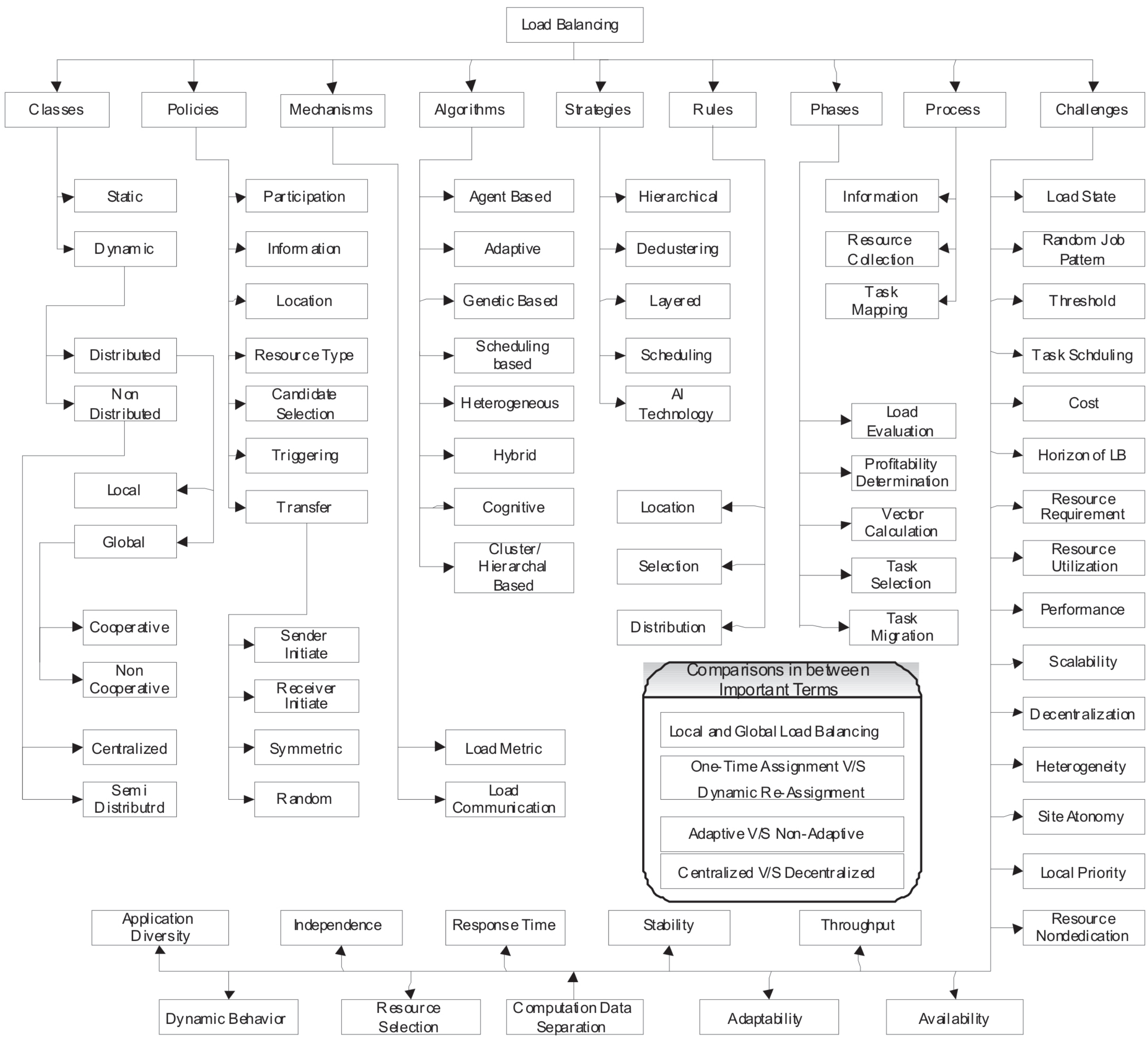

Figure 2. Classification of load balancing schemes

comparisons are discussed. A comparison of different Load balancing techniques is Shown in Section 2. Section 3 exhibits proposed model of tree based approach and Section concludes the paper.

1. Classification of Dynamic Load Balancing

Load balancing can be broadly categorized as Classes, Policies/Mechanism, Algorithms, Strategies, Processes, and Challenges etc. The problem of Load balancing came into the limelight as soon as the concept of multiprocessor as well as multi computation system architecture was proposed. Today, the modern era of computing is mostly dominated by high Speed processors with incredible processing power, advanced system architecture, complex storage hierarchy, lightning fast network services and powerful application supports, but the problem of load balancing has not been comprehensively solved yet. Various load balancing algorithms have been proposed in the last twenty years or so. The organization of the different load balancing schemes is shown in Figure 2 from the top to the bottom; this structure can be identified as what followed.

Local balancing is the assignment of processor time quantum to task as it is done by every traditional operating system. Global balancing, on the other hand, is the process of deciding where to execute a task in a multicomputer environment. Global balancing can be implemented using a centralized approach, or it may be distributed among various processing elements. Global balancing methods, are further classified into two major groups: Static Load Balancing and Dynamic Load Balancing [124] [110] [111]. In Distributed Scheme, Dynamic load balancing algorithm is executed by all nodes in the system and the responsibility of load balancing is shared among them. Distributed load balancing can take two forms: cooperative and non-cooperative. In Non-Distributed Scheme, the responsibility of load balancing is either taken on by a single or some nodes, but never with all nodes. It can take two forms: centralized and semi distributed. On Semi-distributed, nodes of the distributed system are segmented into clusters. Load balancing within each cluster is centralized; a central node is nominated to take charge of load balancing within this cluster. While in centralized, the central node is only responsible for load balancing of the whole distributed system.

There has been an extensive research in the development of the appropriate load balancing policy. The policy issue is the set of choices that are made to balance the load (which tasks should be executed remotely and where). The mechanism [118] issue carries out the physical facilities to be used for remote task execution and provides any information required by the policies. Figure 2, illustrates a suitable decomposition with each leaf representing a distinct component of a load distribution scheme. The emphasis is about the components of the policy and the provision of information to the policy [116-119]. Sender/Receiver/Symmetrically/Random initiated balancing is the part of transfer policy. Transfer Load Balancing is commonly used to start the load balancing activity, the time a new job arrives or is created on a node and the time a finished job departs from a node. This aspect mainly deals with allocation of nodes to jobs. To Sender-initiated, load distributing activity is initiated by an overloaded node (sender) trying to send a task to an under load node (receiver). It is convenient for remote invocation strategies. In Receiver-initiated Algorithms, load distributing activity is initiated from an under load node (receiver), which tries to get a task from an overloaded node (sender). In Symmetrically Initiated Algorithms, combine the advantages of these two by requiring both senders and receivers to look for appropriate sites. A Random Policy chooses the destination node randomly from all nodes in a distributed system. This simple strategy can result in a significant performance improvement.

A load balancing mechanism is used to harness the computational power of the grid. Such mechanism attempt to balance the load with the result of maximizing resource utilization and optimizing performance. Two mechanisms are used for that which is given below: Load metric mechanism is the characteristic used to describe the load on a resource. This determines the type of information that makes a load index (queue CPU length, memory size) and the way such information is communicated to other loaders (broadcasting, focused addressing, polling) and Load communication mechanism identifies the method by which information, such as the load on a resource, is communicated between the resource and the load distribution policy and mechanisms. The load communication policy can also include the communication between cooperating distributed policies.

To date, there has been a number of exiting initial efforts at developing load balancing systems for grid environments. It is often difficult to make comparisons between distinct efforts because each load balancing is usually developed for a particular system environment or particularly greedy application with different assumptions and constraints. The author attempt to outline the methodologies adopted and ideas behind the popular load balancing system like Agent Based, Cluster/ Hierarchy (Tree based), Scheduling Based, Heterogeneous, Hybrid, Adaptive, Cognitive etc shown in Table 2. The comparisons of few loads balancing Technique are given below in Table 1.

Next classification, Load balancing strategies, try to distribute the workload uniformly across all computers in a grid. Load balancing can be done without measuring the current load to avoid the overhead and temporary balancing. Many loads-balancing strategies dynamically react to load imbalances by comparing a load metric to a threshold and transferring workloads to other computers if the threshold is exceeded. Other load balancing strategies use workload priorities and characteristics of the workload to do load balancing. Collecting workload characteristics in advance can decrease the mean response time of batches of requests. It can be classified into many categories like Hierarchical, De-clustering, AI technologies, Layered Strategy, Scheduling strategy etc.

The load balancing processed can be defined in three roles: the location, distribution and selection rules. Zomaya ET. Al [123], describe that the location rule determines which resource domain will be included in the balancing operation. The domain may be local, i.e. inside the node, or global, i.e. between different nodes. The distribution rule establishes the redistribution of the workload among available resources in the domain, while the selection rule decides whether the load balancing operation can be performed preemptive or not.

Load Balancing Phases [113] [114] is categorized into five

LB Technique / Parameters	Prediction Based Approach	Agent Based Approach	Supportive Node Approach	Recitation (Policy and Strategy) Based	Recent Neighbour Approach
Environment	Heterogeneous (Centralized Load Manager)	Distributed heterogeneous P2P	Distributed Heterogeneous	Centralized and sender Initiate	Distributed Heterogeneous
Aim	Effective resources utilization & reduce average job response time	Self Adaptive, Avoiding large amount of data transfer and find under loaded nodes more quickly	Reduces communication delay using supporting nodes	Reduce response time	Reduce communication delay
Strength	Better resource utilization	Migrate load on the basis of state information exchange and location policy	Minimum traffic due to attached central node at each cluster	Based On Load balancing Policy	Load index as a decision factor for scheduling tasks
Criteria	Selected task basis of I/O-intensive, CPU-intensive and memory intensive	Load estimates based on the job queue length and job arrival rates	Primary and centralized approach	Information, triggering and Selection Policies	Total Execution Time, Communication Delay, No. of Task and No. of cluster
Compose in Module/ Matrices	(1) Predictor (2) Selector (3) Scheduler	(1) Job scheduler (2) Communication layer (3) Resource managers	Provide support while migrating load in individual node pool	(1)Information policy (2) Triggering policy (3) Selection Policy	(1) Grid level (2) Cluster level (3) Leaf nodes
Behavior	Cluster Based Queuing model	Analyzed by varying the number of jobs in a nest	Performance criteria is load index & queue-length	Based on CPU consumption and Queue Length	Load Index based on CPU Utilization, Queue length and Communication delay.
Gap	Need to evaluate in large scale cluster	Number of nodes increases in a nest will cause to increase the complexity of program	Complexity in implementation	Resources submit & withdrawn in any movement (Affect performance)	Performance degradation in nesting of cluster

Table 1. Comparisons of Load balancing Techniques

Type	Techniques
Agent Based Dynamic Distributed heterogeneous Network	Azin & Simone et. al. [54]: Ant colony optimization
	Qingqi et. al. [55]: Approximate Optimization
	Jasma et. al. [56]: Decentralized Recent Neighbor (RN) LBA
	Yuan Rao et. al. [57]: On-board routing schemes
	Safa Khalouli et. al. [58]: NP Hard
	Leandro dos et. al. [59]: Evolution algorithms
	M.H. Afshar et. al.[60]: Ant Colony optimization
	Al-Dahoud et. al. [61]: Ant Colony Optimization (ACO)
	Liang Bai et. al. [62]: Multiple Ant Colony optimization
	Husna Jamal et. al. [63]: Focused on local pheromone trail update & trail limit
	Antony Lidya Therasa. S et. al. [64]: Dynamic Adaptation of Checkpoints and Fault tolerance
	Biagio Cosenza, et. al. [65]: Agent Based (i.e., between neighbor workers)
	Yong Hee Kim et. al. [66]: Reflecting Agent Workload and Multi-Agent System
	Magdy Saeb et. al. [67]: Based on agent lifetime on the algorithm & the variance of the workload over the cluster
	Ana et. al. [68]: Diffusion algorithm Searching unbalanced domains
	K. Saruladha et. al. [69]: It explores and find the under load nodes more quickly
Cluster/ Hierarchal Based load Balancing	Sotirious et. al. [70]: Non-content aware load balancing algorithm
	PENG et.al. [71]: Proximity-aware fashion
	Robson et. al. [72]: Local and cluster monitoring mechanisms
	Deepti et. al. [73]: Cluster & non cluster based LBA
	Hung-Chang et. al. [74]: Novel load balancing algorithm with virtual servers
	Ghada F et. al. [75]: Direct Acyclic Graph (DAG) based
	P. K. Suri et. al. [76]: Intra and inter cluster (grid) load balancing
	Weiwei lin et. al. [77]: Combine use of centralized & distributed hierarchical method
	Somayeh Abdi et. al. [78]: Hierarchical Replication Strategy with Job Scheduling & Data Replication Mechanism
	Malarvizhi Nandagopal et. al. [79]: Hierarchical Status Information Exchange Scheduling
	Malarvizhi Nandagopal et. al. [80]: Sender Initiated Decentralized Dynamic LBA
	Gengbin Zheng et. al. [81]: Periodic and hierarchical LBA
	Bin Wang et. al. [82]: P2P System (Distributed Hash Table)
	Robson E. De Grande et. al. [83]: High Level Architecture (HLA) based
	Xiao Qin et. al. [84]: Dynamic communication-aware load balancing scheme
	Dongliang Zhang et. al. [85]: Based on BST (Adaptive balancing)
	Chang Hui et. al. [86]: Deferred shading parallel rendering applications
	Azzedine Boukerche et. al. [87]: DLB for HLA-Based
	Cheng-Jia Lai et. al. [88]: Randomized algorithm based on High-performance queuing method
	Abhinav Bhatele et. al. [89]: Dynamic Topology Aware LBA
	Thein et. al. [90]: Decentralized dynamic load balancing
	Ching-jung et. al. [91]: Tree based/ Mapping/load-balancing methods
	Xiao Qin et. al. [92]: Effective I/O-aware load-balancing schemes
Genetic algorithm for job scheduling (GA)	Shoukat Ali et. al. [93]: On-Load machine Balance schemes
	Chao-Chin Wu et. al. [94]: Fault-tolerance mechanisms with check pointing and the job replication
	Wanneng Shu et. al. [95]: Min-min chromosome genetic algorithm (MCGA) with GA.
	Salahuddin et.al. [56]: Central scheduler with effective threshold policy and high scalability.
	P. V. Prasad et. al. [96]: Network reconfiguration distribution system
Scheduling Algorithm	Wenmin et.al. [97]: Load-balancing dynamic scheduling (LBDS) algorithm
	Jie Chang et. al. [98]: Polling scheduling dynamic load balancing algorithm
	Minglong et. al.[99]: State balancing policy & Fault Tolerance in Video on Demand(VOD)
	Youngjoon et. al. [100]: Self scheduling for data centric load balancing
	P Visalakshi et. al. [101]: Hybrid Particle Swarm Optimization
	T.R.V. Anandharajan et. al. [102]: job scheduling and job selection algorithm
	Hemant et. al. [103]: Usage service based approach to data access, integration & resource registration.
	Jaehwan Lee et. al. [104]: PUSH and PULL job migration for P2P desktop grids
	Zhengping Qian et. al. [105]: Deadlock Detection & resolving algorithm
Hybrid load Balancing	Mohsen Moradi et.al. [106]: Time Optimizing Probabilistic LBA
	Yajun Li et. al. [107]: FCFS with GA algorithms
	Moammed et. al. [108]: Annotated sequential program
	Shakti Mishra et. al. [109]: Prioritizing processes

Table 2. Comparisons of Dynamic Load Balancing in Different Environments

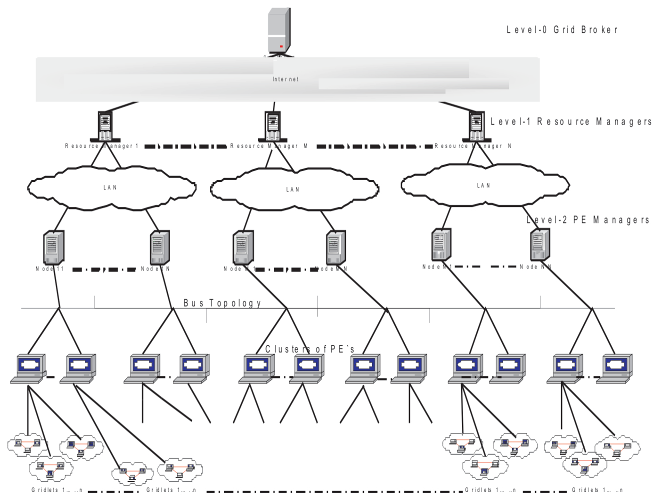

Figure 3. Hierarchical Load Balancing Model

parts, which is Profitability Determination, Work Transfer Vector Calculation, Task Selection, Task Migration etc.

Grid is a system of high diversity, which is rendered by various applications, middleware components and resources. Schoft [115] generalize a load balancing process in the grid into three stages: Information collection is the basis for providing current state information of the resources. It should be performed during the whole course of the system running. Resource Selection is performed in two steps. In the first step, the initial filtering is done with the goal of identifying a list of authorized resources that are available to a given application. In the second step, those resources are aggregated into small collection such that each collection is expected to provide performance desired by the given application. Tasks mapping: The third phase involves mapping the given set of tasks onto a set of aggregated resources including both the computational resources and network resources. Various heuristics may be used to reach a near-optimal solution. The effort of mapping in conjunction of resource selection produces a set of candidate submission. Once the set of candidate submission is ready; the balancer starts to select the best submission subject to the performance goal, based on the mechanisms provided by the performance model.

In grid environment unique characteristics make the design of load balancing algorithm more challenging as below like Load State, Random Job Pattern, Horizon of Load Balancing, Resource Requirement, Resource utilization, Performance, Scalability, Decentralization, Heterogeneity [116], Computation data Separation, Resource Selection, Dynamic Behaviour, Application Diversity, Response Time, Task scale, Throughput, Adaptability [58], Stability, Task Scheduling, Independence etc. [112]. This need to be resolved.

In Figure 2, the author compared between few important terms like Adaptive vs. nonadaptive, one-time assignment vs. dynamic re-assignment, local and global load balancing, centralized vs decentralized.

In adaptive scheme, one (or more parameters) do not correlate to the program performance, it is weighted less next time. On the other hand parameters used in balancing remain the same regardless of the system's past behaviour in the non-adaptive scheme. Onetime assignment vs. dynamic reassignment, in this classification, balanced entities on considered. The one-time assignment of a task may be done dynamically, but once it is scheduled to a given Resource, it can never be migrated to another one. On the other hand, in the dynamic reassignment process, jobs can be migrated from one node to another even after the initial placement is made. A negative aspect of this scheme is that tasks may endlessly circulate about the system without making much progress. Local and Global Load Balancing, falls under the distributed scheme for a centralized scheme should always act globally. In a local load balancing, each resource polls other resources in its neighbourhood and uses this local information to decide on a load transfer. This local neighbourhood is usually denoted as the migration space. The primary objective is to minimize

Load Balancing/ Parameters	Graph based	Tree based	Agent Based	Learning Based
Architecture	Centralized	Decentralized	Decentralized	Centralized
Communication Overhead	Medium	Medium	Medium	Low
Response Time	High	Medium	Medium	Low
Resource Utilization	NA	Medium	Medium	Medium
Load Balance	Medium	Medium	Medium	High
Examples	Improved Spectral Bisection Algorithm (ISBA) , Repartitioning Hyper graph Model (RHM), Diffusion Based Graph Partitioning Methods (DIBAP)	Hierarchical Tree Based Strategy (LBSTRG), Refinement Tree based Partitioning Method (REFTREE), Cluster Aware random Stealing Algorithm (CRS)	Performance- Driven Task Scheduler (PDTS), Messor System (MESSOR), Self Organizing Agents (SOA)	Multi state Q-learning Approach (MSQL), Machine Learning Approach (ML)

Table 3. Comparison between Load balancing Technique [56]

Parameters	Round Robin and Randomized Algorithms	Local Queue	Central Queue	Central Manager	Threshold
Environment	Static and Decentralized	Dynamic and Decentralized	Dynamic and Centralized	Static and Centralized	Static and Decentralized
Load Monitoring	Round Robin fashion By Manager	By Local and global Load Manager	By Local and Central Load Manager	Through Central Manager Only	Through Distributed processes
Synchronization	Each processor maintains report of all resources	Randomly send Remote messages by load manager	Communication between Local and Central Load Manager	Interprocess communication between Remote Processors	Each Process creates a private copy of the system`s load through remote messages
Rebalancing	Round Robin	Automatic Load balancing	FIFO	Tree mechanism (parent children balancing)	Using Local and Global remote process
Job Migration	Randomly find under-loaded node using round robin fashion	To find out Under load node in the ready Queue	To find out the Process in the process Request Queue	While Dynamic Activity created by different host in the cluster	After checking the Load Limit of the process in the pool

Table 4. Comparative Load Balancing Algorithms [48-53]

remote communication as well as effectively balance the load on the resources. However, in a global balancing scheme,

Considerable amount of information to be exchanged in the system may affect its scalability. Here, Centralized and Decentralized Load Balancing algorithm is now compared on the basis of a few parameters.

In traditional distributed systems, grid architecture has the following properties that characterize like Heterogeneity, Scalability, Adaptability, Hierarchical, Independent etc. These properties make the load balancing problem more complex than traditional parallel and distributed systems, which offer homogeneity and stability of their resources. Also, interconnected networks on grids have very disparate performances and tasks submitted to the system can be very diversified and irregular. These various observations show that it is very difficult to define a load balancing system which can integrate all these factors.

2. Comparisons of Load Balancing Techniques

After comparing existing load balancing techniques, we have used our proposed algorithm. Table 3. easily shows that Tree based and Agent based Load balancing technique is better than others. So, tree based technique is used in the proposed work. Table 4 shows comparitive load balancing algorithm.

3. Proposed Load Balancing Model

On the basis of above comparisons, we have proposed Grid architecture for load balancing depicted through Figure 3.

Poisson process has been used for random job arrival with a random computation length. Considering that the jobs are sequenced, mutually independent with the arrival rate, it can be executed on any site. Furthermore, the site should have to meet the job demand for computing resource and the amount of data transmitted. Each processor can only execute one job at a time and execution of a job cannot be interrupted or moved to another process during execution. This model has been divided into three levels: Level-0 Broker, Level-1 Resource, and Level-2 Machine Level. These entities can have customized characteristics. When a new Gridlet arrives at a machine, it may go under lightly loaded, lightly loaded, overloaded and normal loaded resources by load calculation being computed at each node. In order to

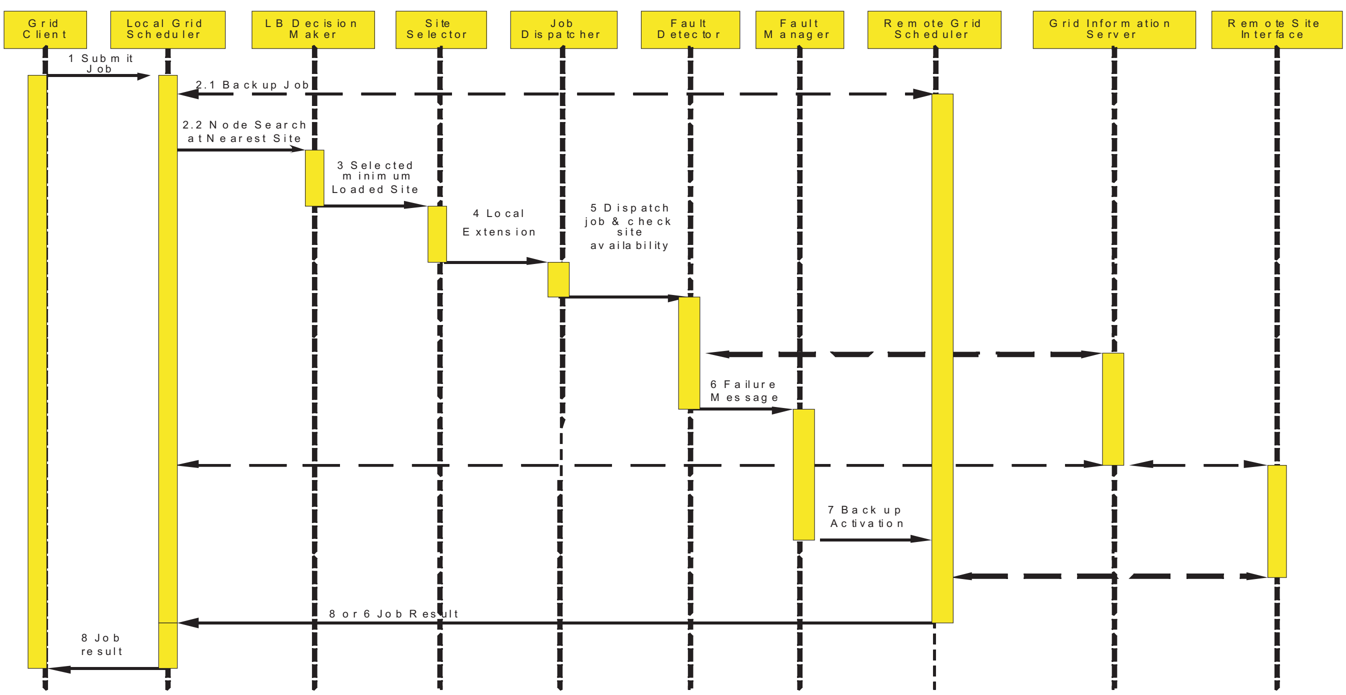

Figure 4. Sequence Diagram of Components and their interaction

compute the mean job response time analysis one Grid Broker (GB) section as a simplified Grid model has been considered. Grid Broker is the top manager of a Grid environment, which is responsible for maintaining the overall Grid activities of scheduling and rescheduling. It acquires the information of the work load from Grid resources and sends the tasks to resources for optimization of load. Resource that comes next to Grid Broker in the hierarchy, are connected through internet. The resource manager is responsible to maintain the scheduling and load balancing of its machines and it also sends an event to the Grid broker during overload. The machine is a Processing Entity (PE) manager, responsible for task scheduling and load balancing of its PEs connected with various resources via LAN. Processing Entity (PE) manager also sends an event to resource during overload. PE`s next to machines are mainly responsible for calculating workload and threshold values for load balancing, job migration and passes the load information upwards to machines via buses. Gridlet is considered as a load and assigned to any of the PE`s according to their capability (i.e. Computational speed, queue length, etc.). The proposed hierarchical based Load Balancing technique in a heterogeneous Grid environment, assumed the heterogeneity in terms of processing capability and the same is applicable to the PEs.

4. System Methodology

Grid topology and Grid mapping into a tree-based model has been shown by Figure 4 with some application and sequence diagram design. The interaction of each site by computing nodes in a network is given through the following steps: (i) In the network each site may contain multiple machines and each machine may have a single or multiple processing elements (PEs). Grid scheduler is a software component that runs on each site. It manages the system related information (i.e. CPU utilization, remaining CPU capability, remaining memory etc.) computing nodes and sites in order to join the Grid, provide resources, receives jobs from Grid clients, assigns them to the processors in the Grid system. (ii) It provides back up to the jobs and discovers the candidate set of nearest site based on the job requirements, resource characteristics and sends the list to load balancing decision maker. (iii) Decision maker decides on the basis of the minimum Grid site, whether the job required an execution at the local site or remote site and transfers the job accordingly. (iv) Job dispatcher ships the jobs and check site availability to the fault detector and monitors the state of sites. (v) If site failure or system degradation occurs, the fault detects or sends a failure message to fault manager. If the fault manager receives a failure message, it reschedules the job using a primary backup approach [125].

In this system, sites play the role of assisting the execution of jobs. When any site is in an idle state and can provide its resource, a join message and related hardware information will be transmitted to the Grid scheduler of the nearest neighbour site. When it can no longer provide its resource, it will transmit an exit message to the Grid scheduler of the nearest neighbour site. The Grid scheduler uses a derived threshold to effectively select appropriate sites. This threshold is based on the load and demands for resource needed to execute the job and will evaluate the sites in the system.

This paper proposes a hierarchical load balancing technique, a strong point of dynamic load balancing method over static load balancing. In static load balancing state, when a job has to be executed, it will be assigned to an appropriate site. The minimum requirement of job's needs for sites is considered as the threshold for the table of effective sites. Sites passing this threshold are called effective sites. As a grid is composed of sites with different performances, the execution time of a job on each site is also different. In dynamic load balancing state, the system will adjust dynamically until it reaches a balance. In a grid environment, effectiveness of sites varies with time. Thus, the assignment of jobs must be adjusted dynamically in accordance with the variation of the site status.

Conclusion

These challenges pose significant obstacles on the problem of designing an efficient and effective load balancing system for Grid environments. Some problem resulting from the above is not solved successfully yet and still open research issues. As a result, the new load balancing framework must be developed for Grid, which should reflect the unique characteristics of grid systems. Figure 3 demonstrates the significance of Load Balancing models in grid computing. The papers are presented in descending order of years so that the reader can easily perceive recent works on Load Balancing models in Table 2. It can be seen from the table that the Hierarchical/ Tree based are the most widely proposed models in the grid.

After the extensive literature survey, various research gaps come into the picture of the current research work in the area of grid computing and load balancing. Load balancing in grid environment is significantly complicated by the unique characteristics of grids. The reason can be found by going through the assumption underlying traditional system such that:

- All resource resides within a single administrative domain.
- To provide a single system image, the balancer controls all of the resources.
- The resource pool is invariant.
- Contention caused by an incoming application can be managed by the balancer according to some polices, so that its impact on the performance that the site can provide to each application can be well predicted.
- Computation and their data reside in the same site or data staging are a highly predictable process, usually from a predetermined source to a predetermined destination which can be viewed as a constant overhead.

In this paper, a dynamic, distributed tree based load balancing approach for a Grid has been proposed, wherein machines and resources, in addition to the grid broker, also participate in the load balancing operations. The outcome is reduced communication overhead and reduced response time, as the amount of information passed to the upper level entities is typically limited due to the local load balancing performed at lower levels in the hierarchy. The idle time of processing entities also decreases, as load balancing is effected much faster.

References

[1]. S. Prakash, D. P. Vidyarthi, (2011). "Load Balancing in Computational Grid Using Genetic Algorithm", *Advances in Computing:* 1(1): 8-17 DOI: 10.5923/j.ac.20110101.02, (Workload calculation)

[2]. Nazir B, Qureshi K, Manuel P, (2008). "Adaptive fault tolerant job scheduling strategy for economy based grid", *J Super Comput* (October): 116–134,

[3]. Qureshi K, Hussain SS, (2008). "A comparative study of parallelization strategies for fractal image compression on a cluster of workstations". *Int J Comput Methods* 5(3):463–482,

[4]. Rehman A, Qureshi K, Manuel P, Rashid H, (2008). "Resource topology aware GridSim: a step ahead". *J Comput* 19(2):13–22; special issue on Grid and Cluster Computing,

[5]. Foster I, Kesselman C, Tuecke S, (2001). "The anatomy of the Grid: enabling scalable virtual organization". *Int J High Perform Comput Appl* 15(3):200–222,

[6]. Berman F, Fox G, Hey AJ, (2003). "*Grid computing: making the global infrastructure a reality*". Wiley, New York.

[7]. Hussain S, Qureshi K, Rashid H, (2009). "Local predecimation with range index communication parallelization strategy for fractal image compression on a cluster of workstations". *Int Arab J Inf Technol* 6(3):293–296,

[8]. Arif Wani M, Arabnia HR (2003). "Parallel edge-region-based segmentation algorithm targeted at reconfigurable multi-ring network". *J Supercomput* 25(1):43–63,

[9]. Bhandarkar SM, Arabnia HR, (1995). "The Hough transform on a reconfigurable multi-ring network". *J Parallel Distrib Comput* 24(1):107–114,

[10]. Qureshi K, Manuel P, (2007). "Adaptive pre-task scheduling strategy for heterogeneous distributed ray-tracing system. *Int J Comput Electr Eng* 33:70–78,

[11]. Lu K, Subrata R, Zomaya AY, (2006). "An efficient load balancing algorithm for heterogeneous grid systems considering desirability of grid sites". *In: Proceedings of* *the 25th IEEE international conference on performance, computing, and communications,* 10–12 April.

[12]. **Fernandes de Mello R, Senger LJ, Yang LT, (2006).** "A routing load balancing policy for grid computing environments". *In: Proceedings of the 20th international conference on advanced information networking and applications (Aina'06)*, Vol 1, 18–20 April.

[13]. **Touzene A, Al Yahia S, Day K, Arafeh B, (2005).** "Load balancing grid computing middleware". In: *IASTED international conference on web technologies, applications, and services (WTAS 2005),* Calgary, Canada, July.

[14]. **Krauter K, Buyya R, Maheswaran M, (2002).** "A taxonomy and survey of grid resource management systems for distributed computing. *Softw Pract Exp* 32:135–164,

[15]. **Yagoubi B, Slimani Y, (2007).** "Task load balancing strategy for grid computing". *J Comput Sci* 3(3):186–194,

[16]. **Li Y, Lan Z (2005).** A survey of load balancing in grid computing. In: *Lecture notes in computer science,* Vol. 3314. Springer, Berlin, Heidelberg, pp 280–285

[17]. **Quetier B, Cappello F, (2005).** "A survey of Grid research tools: simulators, emulators and real life platforms". In: *Proceeding of 17th IMACS world congress* Paris, France,

[18]. **Murshed M, Buyya R, Abramson D, (2001).** "*GridSim: A toolkit for the modeling and simulation of global grids*". Technical Report, Monash, CSSE,

[19]. **Yagoubi B, Slimani Y, (2006).** "Dynamic load balancing strategy for grid computing". *Trans Eng Comput Technol* 13:260–265,

[20]. **Yagoubi B, Slimani Y, (2007).** "Load balancing strategy in grid environment". *J Inf Technol Appl* 1(4):285–296.

[21]. **Payli RU, Yilmaz E, Ecer A, Akay HU, Chien S, (2006).** "DLB—a dynamic load balancing tool for grid computing, scalable computing". *Pract Exp* 7(2):15–23,

[22]. **R. Venkatesan and M. Blessy Rathna Solomi, (2011).** "Analysis of Load Balancing Techniques in Grid", V.V. Das and N. Thankachan (Eds.): *CIIT, CCIS* 250, pp. 147–151, © Springer-Verlag Berlin Heidelberg.

[23]. **Ruchir Shah, Bhardwaj Veeravalli, (2007).** "On the Design of Adaptive and Decentralized Load-Balancing Algorithms with Load Estimation for Computational Grid Environments", *IEEE Transactions on Parallel And Distributed Systems,* Vol. 18, No. 12, December.

[24]. **Belabbas Yagoubi and Yahya Slimani, (2007).** "Load Balancing Strategy in Grid Environment" *Journal of Information Technology and Applications pp*. 285-296, Vol. 1 No. 4 March,

[25]. **S. Prakash, D. P. Vidyarthi, (2011).** "Load Balancing in Computational Grid Using Genetic Algorithm" *Advances in Computing*: 1(1): 8-17 DOI: 10.5923/j.ac.20110101.02.

[26]. **Said Fathy El-Zoghdy, Shebin El-Koom, Egypt, (2011).** "A Load Balancing Policy for Heterogeneous Computational Grids", *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 2, No. 5,

[27]. **PENG Limin, XIAO Wenjun, (2011).** "A Binary-Tree based Hierarchical Load Balancing Algorithm in Structured Peer-to-Peer Systems", *Journal of Convergence Information Technology,* Volume 6, Number 4. April.

[28]. **Deepti Sharma and Archana B.Saxena, (2011).** "Framework to Solve Load Balancing Problem in Heterogeneous Web Servers", *International Journal of Computer Science & Engineering Survey (IJCSES)* Vol.2, No.1, Feb 2011 DOI : 10.5121/ijcses..2104.

[29]. **Shakti Mishra, Dharmender Singh.Kushwaha, Arun Kumar Misra, (2010).** "An Efficient Job Scheduling Technique in Trusted Clusters for Load Balancing", *The First International Conference on Cloud Computing,* GRIDs, and Virtualization:Cloud Computing.

[30]. **Mohsen Moradi, Mashaala Abbasi Dezfuli and Mohammad Hasan Safavi, (2010).** "A New Time Optimizing Probabilistic Load Balancing Algorithm in Grid Computing" *IEEE.*

[31]. **Hemant Kumar Mehta, Manohar Chandwani and Priyesh Kanungo, (2010).** "A Modified Delay Strategy for Dynamic Load Balancing in Cluster and Grid Environment", *IEEE.*

[32]. I. Stoica, R. Morris, DR. Karger, et al, (2003). "Chord: A scalable peer-to-peer lookup protocol for internet applications", *IEEE/ACM Trans. on networking*, Vol.11, No.1, pp.17-32,

[33]. Shih-Jie Lin, Min-Chun Huang, Kuan-Chou Lai and Kuo-Chan Huang, (2008). "Design and Implementation of Job Migration Policies in P2P Grid System" *Asia-Pacific Services Computing Conference IEEE.*

[34]. Lalitha Hima Bindu.P., Venkatesan R.,Ramalakshmi.K., (2011). "Perspective Study on Resource level Load balancing in Grid Computing Environments" *IEEE.*

[35]. K. Saruladha, G. Santhi, (2007). "Behavior of Agent Based Dynamic Load Balancing Algorithm for Heterogeneous P2P Systems", *International Conference on Computational Intelligence and Multimedia Applications.*

[36]. Kai Lu and Albert Y. Zomaya, (2007). "A Hybrid Policy for Job Scheduling and Load Balancing in Heterogeneous Computational Grids" , *Sixth International Symposium on Parallel and Distributed Computing (ISPDC'07) IEEE.*

[37]. Belabbas Yagoubi and Yahya Slimani, (2006). "Dynamic Load Balancing Strategy for Grid Computing", *World Academy of Science, Engineering and Technology* 19

[38]. Mor harchol-balter and Allen b. Downey, "*Exploiting Process Lifetime Distributions for Dynamic Load Balancing",*

[39]. *ACM Transactions on Computer Systems,* Pages 253–285, Vol. 15, No. 3, August 1997.

[40]. Ossama Othman, and Douglas C. Schmidt, "*Optimizing Distributed System Performance via Adaptive Middleware Load Balancing",*

[41]. Work funded in part by ATD, BBN, Cisco, DARPA contract 9701516, and Siemens MED.

[42]. Srinivas Mandalapu, "*Dynamic Load Balancing Design and Modeling* in *MPIAB".*

[43]. Prof. T. Fahringer, "Seminar work on Grid Computing 2 (SE 2.0, 703822) SIMULATING THE GRID WS 2006/07.

[44]. *http://www.cloudbus.org/Gridsim/*.

[45]. H.Shan, L. Oliker, R. Biswas and W. Smith, (2004). "Scheduling in Heterogeneous Grid Environments: The effects of data migration", *in Proc. of ADCOM2004: International Conference Advanced Computing and Communication,* Ahmedabad Gujarat, India.

[46]. M. J. Litzkow, M.Livny, M. W. Mutka, (1988). "Condor-A Hunter of Idle Workstations", *8th IEEE International Conference of Distributed Computing and Systems*, pp. 104-111.

[47]. Parag Kulkarni & Indranil Sengupta Department of Computer Science & Engg. Indian Institute of Technology, Kharagpur, "Load Balancing With Multiple Token Policy" *ICTA'07,* April 12-14, Hammamet, Tunisia.

[48]. Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma, (2008). "Performance Analysis of Load Balancing Algorithms" *World Academy of Sci. Engg and Tech*. 38,

[49]. W. I. Kim and C. S. Kang, (2003). "An adaptive soft handover algorithm for traffic-load shedding in WCDMA mobile communication system", presented at *WCNC*'.

[50]. W. Leinberger, G. Karypis, and V. Kumar, (2000). "Load Balancing Across Near-Homogeneous Multi-Resource Servers", presented at, Cancun, Mexico,

[51]. Zhong Xu, Rong Huang, (2009). "Performance Study of Load Balancing Algorithms in Distributed Web Server Systems", *CS213 Parallel and Distributed Processing Project Report, IJCSIS,* Volume 6 No. 1 October.

[52]. R. Motwani and P. Raghavan, (1996). "Randomized algorithms", *ACM Computing Surveys (CSUR)*, 28(1):33-37,

[53]. William Leinberger, George Karypis, Vipin Kumar, (2000). "Load Balancing Across Near-Homogeneous Multi-Resource Servers", 0-7695-0556- 2/00, *IEEE*,

[54]. Simone A. Ludwig, Azin Moallem, (2011). "Swarm Intelligence Approaches for Grid Load Balancing" *J Grid Computing,* Springer Science Business Media B.V.

[55]. Qingqi Long, Jie Lin, Zhixun Sun, (2011). "*Agent scheduling model for adaptive dynamic load balancing in agent-based distributed simulations*", Tongji University, Shanghai ,China Simulation Modelling Practice and Theory, 1021–1034 Science Direct Elsevier, .

[56]. **Jasma Balasangameshwara, Nedunchezhian Raju, (2010).** "A Decentralized Recent Neighbour Load Balancing Algorithm for Computational Grid", *International Journal of ACM Jordan* (ISSN 2078-7952), Vol. 1, No. 3, September.

[57]. **Yuan Rao, Ru-chuan Wang, (2010).** "Agent-based load balancing routing for LEO satellite networks", *Computer Networks*, 3187–3195, ELSEVIER.

[58]. **Safa Khalouli, Fatima Ghedjati, Abdelaziz Hamzaoui, (2010).** "A Meta-Heuristic approach to solve a JIT scheduling problem in hybrid flowshop", *Engineering Applications of Artificial Intelligence*, 765–771, Elsevier,

[59]. **Leandro dos Santos Coelho, Diego Luis De Andrade Bernert, (2010).** *"A modified ant colony optimization algorithm based on differential evolution for chaotic synchronization",* Elsevier.

[60]. **M.H. Afshar, (2010).** "A parameter free Continuous Ant Colony Optimization Algorithm for the optimal design of storm sewer networks: Constrained and unconstrained approach", *Advances in Engineering Software* 41, 188–195,

[61]. **Al-Dahoud Ali, Mohamed A. Belal and Moh'd Belal Al-Zoubi, (2010).** "Load Balancing of Distributed Systems Based on Multiple Ant Colonies Optimization" *American Journal of Applied Sciences* 7 (3): 433-438, Science Publications,

[62]. **Liang Bai, Yan-Li Hu, Song-Yang Lao and Wei-Ming Zhang, (2010).** "Task Scheduling with Load Balancing using Multiple Ant Colonies Optimization in Grid Computing", *Sixth International Conference on Natural Computation (ICNC)*

[63]. **Husna Jamal Abdul Nasir, Ku Ruhana and Ku-Mahamud, (2010).** "Grid Load Balancing Using Ant Colony Optimization", *Second International Conference on Computer and Network Technology, IEEE ICCNT.*

[64]. **Antony Lidya Therasa. S, Sumathi. G and Antony Dalya .S, (2010).** "Dynamic Adaptation of Checkpoints and Rescheduling in Grid Computing", *International Journal of Computer Applications* (0975 – 8887) Volume 2 – No.3, May.

[65]. **B. Cosenza, G. Cordasco, R. De Chiara and V. Scarano, (2010).** "Distributed Load Balancing for Parallel Agent-based Simulations",*http://www.isislab.it/projects/DistrSteer/Dist* Steer: Parallel Distributed Agent-Based Simulations,

[66]. **Yong Hee Kim, Seungwok Han, Chang Hun Lyu and Hee Yong Youn, (2009).** "An Efficient Dynamic Load Balancing Scheme for Multi-Agent System Reflecting Agent Workload", *International Conference on Computational Science and Engineering*,

[67]. **Magdy Saeb and Cherine Fathy, (2010).** "*Performance Evaluation of Mobile Agent-based Dynamic Load Balancing Algorithm*",

[68]. **Ana Cortes Fite, (2000).** "*A new distributed diffusion algorithm for dynamic load balancing in parallel systems"*, Barcelona (Spain) September.

[69]. **K. Saruladha, G. Santhi, (2007).** "Behavior of Agent Based Dynamic Load Balancing Algorithm for Heterogeneous P2P Systems", Pondicherry Engineering College, Pondicherry, *International Conference on Computational Intelligence and Multimedia Applications,*

[70]. **Sotirios Kontogiannis, Stavros Valsamidis, and Alexandros Karakos, (2011).** "ALBL, ALBL/HSC algorithms: Towards more scalable, more adaptive and fully utilized balancing systems", *Journal of Computing*, Volume 3, Issue 2, Issn 2151-9617, February.

[71]. **PENG Limin, XIAO Wenjun, (2011).** "A Binary-Tree based Hierarchical Load Balancing Algorithm in Structured Peer-to-Peer Systems", *Journal of Convergence Information Technology,* Volume 6, Number 4, April.

[72]. **Robson E. De Grande, Azzedine Boukerche, (2011).** "Dynamic balancing of communication and computation load for HLA-based simulations on large-scale distributed systems", *J. Parallel Distrib. Comput.* 40–52,

[73]. **Deepti Sharma, Archana B.Saxena, (2011).** "Framework to Solve Load Balancing Problem in Heterogeneous Web Servers", *International Journal of Computer Science & Engineering Survey (IJCSES)* Vol.2, No.1, Feb.

[74]. **Hung-Chang Hsiao, Member, IEEE Computer Society, Hao Liao, Ssu-Ta Chen, and Kuo-Chan Huang, (2011).** "Load Balance with Imperfect Information in Structured Peer-to-Peer Systems", *IEEE Transactions on Parallel and Distributed Systems,* Vol. 22,

[75]. **Ghada F. El Kabbany, Nayer M. Wanas, Nadia H. Hegazi, Samir I. Shaheen, (2010).** "A Dynamic Load Balancing Framework for Real-time Applications in Message Passing Systems" *Springer Science Business Media,* LLC 28 April.

[76]. **P. K. Suri, Manpreet Singh, (2010).** "An Efficient Decentralized Load Balancing Algorithm For Grid", *IEEE*,

[77]. **Weiwei Lin,Wuyao Shen, (2010).** "Tree-Based Task Scheduling Model and Dynamic Load-Balancing Algorithm for P2P Computing", South China University of Technology, Guangzhou, China, *10th IEEE International Conference on Computer and Information Technology (CIT).*

[78]. **Somayeh Abdi and Somayeh Mohamadi, (2010).** "The Impact of Data Replication on Job Scheduling Performance in Hierarchical Data Grid", *International Journal on applications of graph theory in wireless ad hoc networks and sensor networks (GRAPH-HOC)* Vol.2, No.3, September.

[79]. **Malarvizhi Nandagopal and Rhymend V Uthariaraj, (2010).** Hierarchical Status Information Exchange Scheduling and Load Balancing For Computational Grid Environments", *IJCSNS International Journal of Computer Science and Network Security,* Vol.10 No.2, February.

[80] **Malarvizhi Nandagopal, K.Gokulnath and V.Rhymend Uthariaraj, (2010).** "Sender Initiated Decentralized Dynamic Load Balancing for Multi Cluster Computational Grid Environment", *A2CWiC*.

[81]. **Gengbin Zheng, Abhinav Bhatele, Esteban Meneses and Laxmikant V. Kale, (2010).** "*Periodic Hierarchical Load Balancing for Large Supercomputers*",

[82]. **Bin Wang and Qing-guo Shen, (2010).** "ID Management and Allocation Algorithm for P2P Load Balancing", *IEEE*.

[83]. **Robson E. De Grande, Azzedine Boukerche, (2010).** "Dynamic balancing of communication and computation load for HLA- based simulations on large-scale distributed systems", *J. Parallel Distrib. Computer*,

[84]. **Xiao Qin, Senior Member, IEEE, Hong Jiang, Adam Manzanares, Xiaojun Ruan and Shu Yin, (2010).** "Communication-Aware Load Balancing for Parallel Applications on Clusters", *IEEE Transactions on computers,* Vol. 59, No. 1, January.

[85]. **Dongliang Zhang, Changjun Jiang and Shu Li, (2009).** "A fast adaptive load balancing method for parallel particle-based simulations", *Simulation Modelling Practice and Theory,*

[86]. **Chang Hui, Lei Xiaoyong, Dai Shuling, (2009).** "A Dynamic Load Balancing Algorithm for Sort-first Rendering Clusters", *IEEE*,

[87]. **Azzedine Boukerche and Robson Eduardo De Grande, (2009).** "Dynamic Load Balancing Using Grid Services for HLA-Based Simulations on Large-Scale Distributed Systems", *13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications.*

[88]. **Saravanakumar E. and Gomathy Prathima, (2010).** "A novel load balancing algorithm for computational grid", *International Journal of Computational Intelligence Techniques,* ISSN: 0976–0466 E-ISSN: 0976–0474 Volume 1, Issue 1 PP-20-26,

[89]. **Abhinav Bhatele, Laxmikant V. Kale and Sameer Kumar, (2009).** "Dynamic Topology Aware Load Balancing Algorithms for Molecular Dynamics Applications", *ICS'09,* New York town Heights, New York, USA, ACM,

[90]. **Thein Thein Aye and Htway Htway Hlaing, (2008).** "An Efficient Dynamic Load Balancing Policy in Cluster Computing System", *GMSARN International Conference on Sustainable Development:* Issues and Prospects for the GMS 12-14 Nov.

[91]. **Ching-Jung Liao, Yeh-Ching Chung, (1999).** "Tree-Based Parallel Load-Balancing Methods for Solution-Adaptive Finite Element Graphs on Distributed Memory Multicomputers", *IEEE Transactions on Parallel And Distributed Systems,* Vol. 10, No. 4, April

[92]. **Xiao Qin,Hong Jiang,Adam Manzanares, Xiaojun Ruan, and Shu Yin, (2009).** "Dynamic Load Balancing for

I/O-Intensive Applications on Clusters" *ACM Transactions on Storage 5:3*, Article 9, November.

[93]. **Shoukat Ali, Behdis Eslamnour, Zehra Shah, (2010).** "*A Case for on-machine load balancing*", Elsevier.

[94]. **Chao-Chin Ren-Yi Sun, (2010).** "An integrated security-aware job scheduling strategy for large-scale computational grids", *Future Generation Computer Systems,*

[95]. **Robert Elsässer and Thomas Sauerwald, (2010).** "*Discrete Load Balancing is (almost) as Easy as Continuous Load Balancing*", PODC Zurich, Switzerland, ACM ,

[96]. **P. V. Prasad, S. Sivanagaraju and N. Sreenivasulu, (2010).** "*Network Reconfiguration for Load Balancing in Radial Distribution Systems Using Genetic Algorithm*".

[97]. **Wenmin Miao, Dongni Li and Wei Zhang, (2010).** "A Load-Balancing Dynamic Scheduling Algorithm under Machine Failure Conditions", *International Conference on Intelligent Computation Technology and Automation, IEEE,*

[98]. **Jie Chang, Wen'an Zhou, Junde Song and Zhiqi Lin, (2010).** "Scheduling Algorithm of Load Balancing Based on Dynamic Policies", *Sixth International Conference on Networking and Services,*

[99]. **Minglong Zhang, Boqin Feng, (2008).** "A Novel Migration Algorithm Based-on "States-balancing" in a Distributed Multimedia Services System", *International Conference on Multimedia and Ubiquitous Engineering, IEEE*

[100]. **Youngjoon Jo and Milind Kulkarni, (2010).** "Brief Announcement: *Locality-aware Load Balancing for Speculatively-parallelized Irregular Applications*", Purdue University, SPAA'10, Thira, Santorini, Greece.ACM 978-1-4503-0079-7/10/06, June 13–15,

[101]. **P Visalakshi and S N Sivanandam, (2009).** "Dynamic Task Scheduling with Load Balancing using Hybrid Particle Swarm Optimization", *Int. J. Open Problems Compt. Math.*, Vol. 2, No. 3, Copyright ©ICSRS Publication, September.

[102]. **T.R.V. Anandharajan, Dr. M.A. Bhagyaveni, (2011).** "Co-operative Scheduled Energy Aware Load-Balancing technique for an Efficient Computational Cloud", *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 2, March.

[103]. **Hemant Kumar Mehta, Priyesh Kanungo and Manohar Chandwani, (2011).** "*Performance Enhancement of Scheduling Algorithms in Clusters and Grids using Improved Dynamic Load Balancing Techniques*", WWW 2011 – Ph. D. Symposium, March 28–April 1, , Hyderabad, India.

[104]. **Jaehwan Lee, Pete Keleher and Alan Sussman, (2010).** "Decentralized Dynamic Scheduling across Heterogeneous Multi-core Desktop Grids", *IEEE-*

[105]. **Zhengping Qian, Ming Zeng, Deyu Qi, Kefu Xu, (2008).** "*A Dynamic Scheduling Algorithm for Distributed Kahn Process Networks in a Cluster Environment*", DOI 10.1109/PACIIA.2008.190, *IEEE.*

[106]. **Mohsen Moradi, Mashaala Abbasi Dezfuli, Mohammad Hasan Safavi, (2010).** "A New Time Optimizing Probabilistic Load Balancing Algorithm in Grid Computing" *IEEE,*

[107]. **Yajun Li, Yuhang Yanga, Maode Mab and Liang Zhou, (2009).** "A hybrid load balancing strategy of sequential tasks for grid computing environments", *Future Generation Computer Systems,*

[108]. **Mohammed Javeed Zaki, Wei Li, and Srinivasan Parthasarathy, (1997).** "Customized Dynamic Load Balancing for a Network of Workstations", *Journal of Parallel And Distributed Computing* 43, 156–162,

[109]. **Shakti Mishra, D.S. Kushwaha and A.K. Misra, (2010).** "Hybrid reliable load balancing with MOSIX as middleware and its formal verification using process algebra", *Future Generation Computer Systems*,

[110]. **Z. Zeng, B. Veeravalli (2004).** Rate-Based and Queue-Based Dynamic Load Balancing Algorithms in Distributed Systems, *Proc. 10th IEEE Int. Conf. On Parallel and Distributed Systems,* pp. 349- 356.

[111]. **M. Amini, H. Deldari (2006).** Grid Load Balancing Using an Echo System of Ants, *Proc. of 24th IASTED Int. Conf,* Innsbruck, pp. 47–52.

[112]. **Parveen Jain, Daya Gupta**, "*An Algorithm for*

Dynamic Load Balancing in Distributed Systems with Multiple Supporting Nodes by Exploiting the Interrupt Service".

[113]. **Jerrell Watts and Stephen Taylor,** "*A Practical Approach to Dynamic Load Balancing* "

[114]. **Yagoubi and 2Y. Slimani,** "*Task Load Balancing Strategy for Grid Computing"*

[115]. **Schopf, J., (2001).** *Ten actions when super scheduling.* Document of scheduling working group, global grid forum, *http://www.ggf.org/documents/GFD.4.pdf.*

[116]. **Kabalan, K.Y., W.W.Smar and J.Y. Hakimian,.** "Adaptive load sharing in heterogeneous systems: Policies, modifications and simulation." *Intl. J. Simulation,* 3: 89-100. 2002

[117]. **Xu, C. Z. and F. C. M. Lau,** *Load balancing in parallel computers: Theory and practice,* Kluwer, Boston, MA. 1997.

[118]. **Nazir B, Qureshi K, (2008).** Manuel P Adaptive fault tolerant job scheduling strategy for economy based grid, *J Super Comput* (October): 116–134

[119]. **H.C. Lin, C.S. Raghavendra**, A Dynamic Load-Balancing Policy with a Central Job Dispatcher (LBC), *IEEE Transaction on Software Engineering,* Vol. 18, No. 2, pp. 148-158. (1992)

[120]. **Casanova, H., M. Kim, J. S. Plank and J. J. Dongarra,. (1999).** Adaptive scheduling for task farming with grid middleware. In the *International J. High Performance Computing Application*, pp: 231-240.

[121]. **Alaoui, Frieder and El-hzawi,. (1999).** A parallel genetic algorithm for task mapping on parallel machine. *Job scheduling stragies for parallel processing:* {*IPPS*} `95 *Workshop*,LNCS.

[122]. **Belabbas Yagoubi and Yahya Slimani** "*Dynamic Load Balancing Strategy for Grid Computing"*

[123]. **Y. Zomaya, and Y. (2001).** The Observations on using genetic algorithms for dynamic load-balancing, *IEEE Trans. on Parallel and Distributed Systems*, Vol. 12, No. 9, pp. 899-911.

[124]. **H.C. Lin, C.S. Raghavendra, (1992).** A Dynamic Load-Balancing Policy with a Central Job Dispatcher (LBC), *IEEE Transaction on Software Engineering*, Vol. 18, No. 2, pp. 148-158.

[125]. **Jasma Balasangameshwara, Nedunchezhian Raju, (2012).** "A hybrid policy for fault tolerant load balancing in Grid computing environments" *Journal of Network and Computer Applications* 35, pp. 412– 422.

[126]. **Said Fathy El-Zoghdy, (2012).** "A Hierarchical Load Balancing Policy for Grid Computing Environment", *I. J. Computer Network and Information Security*, 5, 1-12.

ABOUT THE AUTHORS

Neeraj Rathore is presently pursuing PhD under the guidance of Dr. Inderveer Chana from Thapar University. He obtained his B.E. in Computer Science in 2006 and M.E. in Computer Engineering from Thapar University in 2008. He has over 6 publications in International Journals and Conferences of repute and a book which was published in October 2010 in Germany. His areas of interests include Parallel and Distributed Computing, Load Balancing on Grid Computing, DBMS and Data structure.

Dr. Inderveer Chana is currently working as an Associate Professor in Computer Science and Engineering Department of Thapar University, Patiala. She has completed her B.Tech in Computer Science (1997) and M.E. in Software Engineering from TIET (2002) and Ph.D. in Resource Management in Grid Computing from Thapar University, Patiala (2009). She has thirteen years of teaching and research experience. She has over 30 publications in International Journals and Conferences of repute. More than 20 Master students completed their degree so far under her supervision, and currently she is supervising 9 Doctoral candidates in the area of Grid and Cloud Computing. Her research interests include Grid computing, Cloud Computing and resource management challenges in Grids and Clouds.