Journal of Network and Computer Applications 65 (2016) 103–119

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Survey of load balancing techniques for Grid

Deepak Kumar Patel [a,*], Devashree Tripathy [b], C.R. Tripathy [a]

[a] Department of Computer Science & Engineering, Veer Surendra Sai University of Technology, Burla, Sambalpur 768018, Odisha, India
[b] CSIR-Central Electronics Engineering Research Institute, Pilani 333031, Rajasthan, India

ARTICLE INFO

Article history:
Received 28 November 2014
Received in revised form
8 February 2016
Accepted 17 February 2016
Available online 24 February 2016

Keywords:
Grid computing
Distributed systems
Load balancing

ABSTRACT

In recent days, due to the rapid technological advancements, the Grid computing has become an important area of research. Grid computing has emerged a new field, distinguished from conventional distributed computing. It focuses on large-scale resource sharing, innovative applications and in some cases, high-performance orientation. A Grid is a network of computational resources that may potentially span many continents. The Grid serves as a comprehensive and complete system for organizations by which the maximum utilization of resources is achieved. The load balancing is a process which involves the resource management and an effective load distribution among the resources. Therefore, it is considered to be very important in Grid systems. The proposed work presents an extensive survey of the existing load balancing techniques proposed so far. These techniques are applicable for various systems depending upon the needs of the computational Grid, the type of environment, resources, virtual organizations and job profile it is supposed to work with. Each of these models has its own merits and demerits which forms the subject matter of this survey. A detailed classification of various load balancing techniques based on different parameters has also been included in the survey.

Contents

* Correspondence to: Veer Surendra Sai University of Technology, Burla, Sambalpur 768018, Odisha, India.
E-mail addresses: patel.deepak42@gmail.com (D.K. Patel), devashree.tripathy@gmail.com (D. Tripathy), crt.vssut@yahoo.com (C.R. Tripathy).

http://dx.doi.org/10.1016/j.jnca.2016.02.012
1084-8045/

1. Introduction

A *Grid* is a computing and data management infrastructure that provides the electronic underpinning for a global society in business, government, research, science and entertainment (Berman et al., 2003). A computational Grid constitutes the software and hardware infrastructure that provides dependable, consistent, pervasive and inexpensive access to high end computational capabilities (Foster and Kesselman, 1999; Foster, 2002). The Grid integrates networking, communication, computation and information to provide a virtual platform for computation and data management in the same way that the Internet integrates resources to form a virtual platform for information (Berman et al., 2003). The Grid can also be considered as a collection of distributed computing resources over a local or wide area network that appear to an end user as one large virtual computing system (Myer, 2003). The speedy development in computing resources has enhanced the performance of computing systems with reduction in cost. The availability of low cost, high speed networks, powerful computers coupled with the advances and the popularity of the Internet has led the computing environment to be mapped from the traditional distributed systems to the Grid environments (Rathore and Channa, 2014).

A *computational Grid* enables the effective access to high performance computing resources. It supports the sharing and coordinated use of resources, independently from their physical type and location, in dynamic virtual organizations that share the same goal (Rathore and Channa, 2011). Grid infrastructure provides us with the ability to dynamically link together resources as an ensemble to support the execution of large-scale, resource-intensive, and distributed applications (Berman et al., 2003). With its multitude of heterogeneous resources, a proper scheduling and efficient load balancing across the Grid is required for improving the performance of the system (Shah et al., 2007).

Load balancing has been discussed in traditional distributed systems literature for more than three decades. Various strategies and algorithms have been proposed, implemented, and classified in a number of studies. In those studies, the load balancing algorithms attempt to improve the response time of the user's submitted applications by ensuring maximal utilization of available resources. The main goal of this type of algorithm is to prevent, if possible, the condition in which some processors are overloaded with a set of tasks while others are lightly loaded or even idle (Hao et al., 2012). The process of load balancing algorithms in Grids can be generalized into the following four basic steps as shown in Fig. 1 (Yagoubi et al., 2006; Rathore and Channa, 2014).

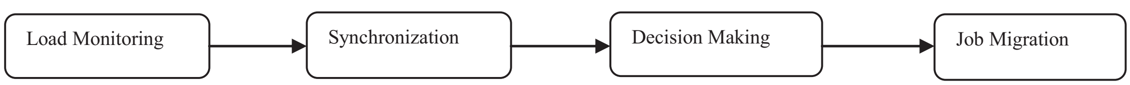

Fig. 1. Basic load balancing steps.

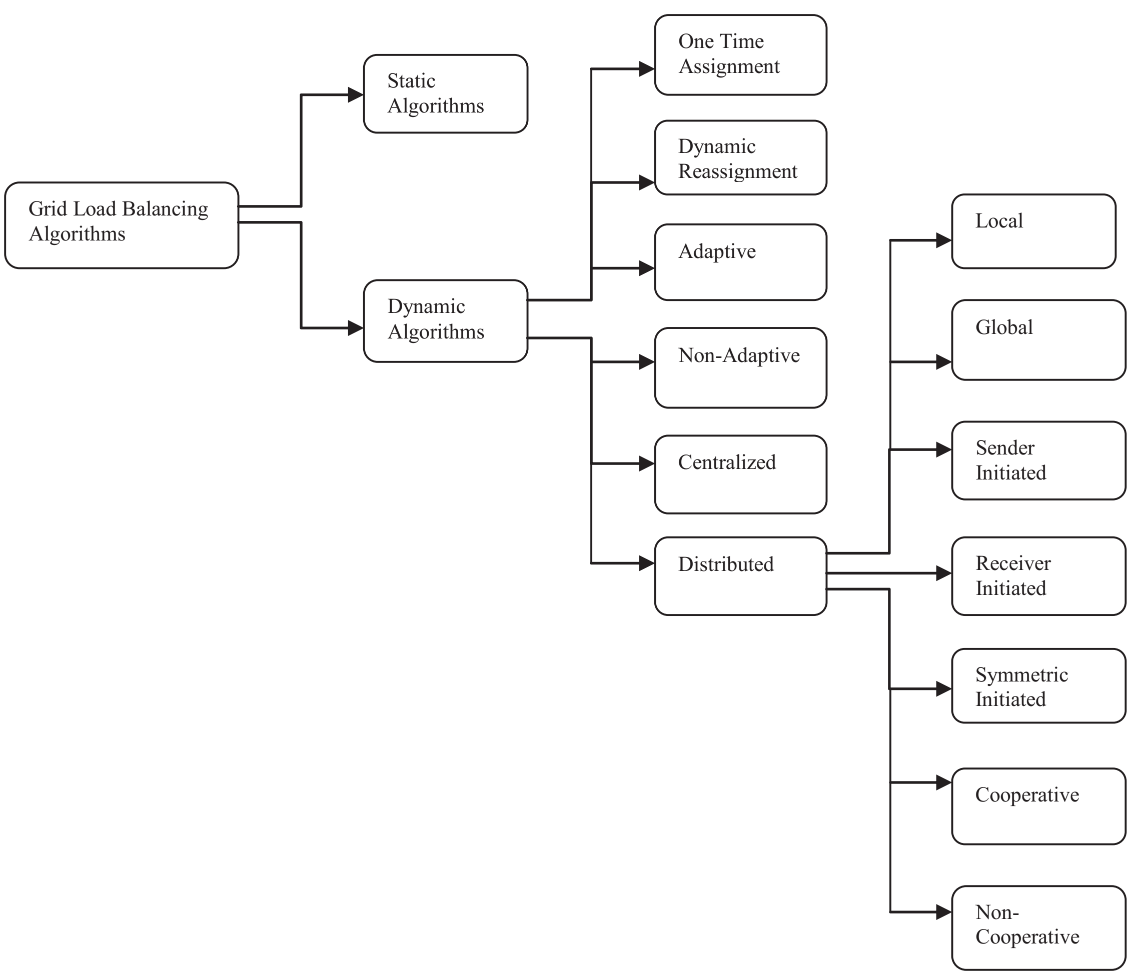

Fig. 2. Grid load balancing tree.

(i) Load Monitoring: Monitoring the resource load and state.
(ii) Synchronization: Exchanging load and state information between resources.
(iii) Decision Making: Calculating the new work distribution and making the work moment decision.
(iv) Job Migration: Actual data movement.

The load balancing can be defined by the implementation of these policies (Hao et al., 2012; Yagoubi and Slimani, 2006, 2007a).

(i) The Information Policy specifies what workload information is to be collected, when it is to be collected, and from where.
(ii) The Triggering Policy determines the appropriate time at which to start a load balancing operation.
(iii) The Resource Type Policy classifies a resource as a server or a receiver of tasks according to its status availability and capabilities.
(iv) The Location Policy uses the results of the Resource Type Policy to find a suitable partner for a resource provider or a resource receiver.
(v) The Selection Policy defines the tasks that should be migrated from overloaded resources (source) to the idlest resources (receiver).

The rest of the paper is organized as follows. In Section 2, we summarize the challenges of load balancing in heterogeneous Grid environments and the various methods of performing load balancing in Grid. A detailed survey of various load balancing techniques is presented in Section 3. Section 4 discusses various load balancing applications. In Section 5, the adoption of load balancing techniques is described. Finally, the concluding remarks are presented in Section 6.

2. Background

This section puts this work in perspective by briefly summarizing the challenges of load balancing in heterogeneous Grid environments. It then discusses the various methods of performing load balancing in Grid and the performance metrics of load balancing.

2.1. Load balancing challenges in Grid computing

A distributed system adopts various policies for the use of the resources and for the resources themselves. The policies include load balancing, scheduling, and fault tolerances. Although a Grid belongs to the class of distributed systems, the traditional policies of the distributed systems cannot be applied as such into a Grid directly. In addition, the load balancing methods used in conventional parallel and distributed systems are not applicable in Grid architectures. Because of the distribution of a large number of resources in a Grid environment and the size of the data to be moved, the traditional distributed approaches to not provide accurate results in a Grid. The heterogeneity, autonomy, scalability, adaptability, dynamic behavior, application diversity, resource non-dedication, resource selection and computation-data separation of a Grid makes the load balancing more difficult and challenging (Yagoubi et al., 2006; Hao et al., 2012).

2.1.1. Heterogeneity

The heterogeneity exists in both the computational and networks resources. Firstly, the networks used in Grids may differ significantly in terms of their bandwidth and communication protocols. Secondly, computational resources are usually heterogeneous. Because resources may have different hardwares such as instruction set, processors, CPU speed, memory size and different softwares like operating systems, file systems and so on (Yagoubi and Slimani, 2006, 2007a).

2.1.2. Autonomy

Typically a Grid may comprise of multiple administrative domains. Each domain shares a common security and management policy. Each domain usually authorizes a group of users to use the resources in the domain. Thus, the application from non-authorized users should not eligible to run on the resources in some specific domains. Because, the multiple administrative domains share Grid resources, a site is viewed as an autonomous computational entity. It usually has its own scheduling policy, which complicates the task allocation problem. A single overall performance goal is not feasible for a Grid system since each site has its own performance goal and the scheduling decision is made independently of other sites according to its own performances (Yagoubi and Slimani, 2006, 2007a).

2.1.3. Scalability

A Grid may grow from a few resources to millions. This raises the problem of potential performance degradation as the size of a Grid increases (Yagoubi and Slimani, 2006, 2007a).

2.1.4. Adaptability

In a Grid, the resource failure may occur frequently. That means the probability that some resources may fail is naturally high. The resource managers must tailor their behavior dynamically so that they can extract the maximum performance from the available resources and services (Yagoubi and Slimani, 2006, 2007a).

2.1.5. Dynamic behavior

The pool of resources can be assumed to be fixed or stable in the traditional parallel and distributed computing environments. However, in Grid, both the networks and computational resources may work dynamically. First, a network shared by many execution domains cannot provide guaranteed bandwidth. This is particularly true when the wide-area network like the internet is involved. Second, both the availability and capability of computational resources may exhibit dynamic behavior. On one hand new resources may join the Grid and on the other hand, some of the existing resources may become unavailable due to problems like network failure. The resource managers must tailor their behaviors dynamically so that they can extract the maximum performance from the available resources and services (Yagoubi et al., 2006).

2.1.6. Application diversity

The Grid applications involve a wide range of users, each having its own special requirements. For example, some applications may require sequential executions, some may consist of a set of independent jobs and other may consist of a set of dependent jobs. In this context, building a general purpose load balancing system seems extremely difficult. An adequate load balancing system should be able to handle a variety of applications (Yagoubi et al., 2006).

2.1.7. Resource non-dedication

The resource usage contention appears as a major issue due to the non-dedication of resources. This results in inconsistency of behavior and performance. For example, in wide area networks, the network characteristics such as latency and bandwidth may be varying over time. Under such an environment, designing an accurate load balancing model is extremely difficult (Yagoubi et al., 2006).

2.1.8. Resource selection and computation-data separation

In traditional systems, the executable codes of applications and input/output data are usually in the same site, otherwise, the input sources and output destinations are determined before the submission of an application. Thus, the cost for data staging can be neglected or the cost-constant is determined before execution. So, the load balancing algorithms need not consider it. But in a Grid, the computation sites of an application are usually selected by the Grid scheduler according to resource status and some performance criterion. Additionally, the communication bandwidth of the underlying network is limited and is shared by a host of background loads, so the communication cost cannot be neglected. This situation brings about the computation-data separation problem. The advantage of it is brought by selecting a computational resource that can provide the low computational cost by neutralizing its high access cost to the storage site (Yagoubi et al., 2006).

The above said challenges put significant obstacles to the problem of designing an efficient and effective load balancing system for the Grid environments. Some such problems resulting from the above have not yet been solved successfully and still remains as an open research issue. Thus, it is a challenging problem to design a load balancing system for the Grid environments that can integrates all the above said factors (Hao et al., 2012).

2.2. Methods of performing load balancing in Grid

Fig. 2 depicts a diagrammatic picture of various methods of performing Grid load balancing (Yagoubi et al., 2006).

In general, the load-balancing algorithms are classified as *static and dynamic* (Yagoubi et al., 2006; Shah et al., 2007; Subrata et al., 2008). The *static* load-balancing algorithms assume that the information governing load-balancing decisions which include the characteristics of the jobs, the computing nodes, and the communication networks are known in advance. The load-balancing decisions are made deterministically or probabilistically at compile time and remain constant during runtime. However, this is considered to be the drawback of the static algorithm. In contrast, the *dynamic* load-balancing algorithms attempt to use the runtime state information to make more informative load-balancing decisions. Here, the responsibility for making global decisions may lie with one centralized location, or be shared by multiple distributed locations. Undoubtedly, the static approach is easier to implement and has minimal runtime overhead. However, the dynamic approaches results in better performance. The advantage of dynamic load balancing over static is that the system need not be aware of the runtime behavior of the application before execution.

The dynamic load-balancing algorithms are classified as *adaptive and non-adaptive* (Yagoubi et al., 2006; Shah et al., 2007). The *adaptive* algorithms are a special type of dynamic algorithms where the parameters of the algorithm and/or the scheduling policy itself is changed based on the global state of the system. Here, the scheduled decisions take into consideration the past and the current system performance and are affected by previous decisions or changes in the environment. A dynamic solution takes the environment inputs into account while making decisions. On the other hand, an adaptive solution takes the environment stimuli into account to modify the load balancing policy itself. In the *non-adaptive* scheme, the parameters used in the balancing remain the same regardless of the system's past behavior.

The dynamic load-scheduling algorithms could also be classified as *centralized or distributed* algorithms (Yagoubi et al., 2006; Shah et al., 2007; Subrata et al., 2008). In the *centralized* approach, one node in the system acts as a scheduler and makes all the load-balancing decisions. The information is sent from the other nodes to the scheduler. In the *distributed* approach, all the nodes of the system remain involved in the load-balancing decisions. It therefore, becomes very costly for each node to obtain and maintain the dynamic state information of the whole system. Here, each node obtains and maintains only the partial information locally to make suboptimal decisions. In distributed load balancing, the state information is distributed among the nodes that are responsible in managing their own resources or allocating tasks residing in their queues to other nodes. However, the distributed algorithms suffer from the problem of communication overheads incurred by frequent information exchange between processors. The centralized strategy on the other hand has the advantage of ease of implementation, but it suffers from the lack of scalability, fault tolerance and the possibility of becoming a performance bottleneck. Therefore, the centralized algorithms are found to be less reliable than the decentralized algorithms.

In distributed load balancing, the assignment or reassignment of a task among the resources should also be considered (Yagoubi et al., 2006). The *one-time assignment* of a task may be dynamically done 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, the jobs can migrate 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.

The *local* and *global* load balancing fall under the distributed scheme since a centralized scheme should always act globally (Yagoubi et al., 2006). In *local* load balancing, each resource polls other resources in its neighborhood and uses this local information to decide up on a load transfer. The primary objective is to minimize remote communication and to efficiently balance the load on the resources. However, in *global* load balancing scheme, the global information of all or a part of system is used to initiate the load balancing. This scheme requires a considerable amount of information to be exchanged in the system which may affect its scalability.

If a distributed load balancing mode is adapted, the next issue that should be considered is whether the nodes involved in job balancing are working cooperatively or independently (non-cooperatively) (Yagoubi et al., 2006). In the *non-cooperative* case, the individual loaders act alone as autonomous entities and arrive at decisions regarding their own optimum objects independent of the effects of the decision on the rest of the system.

The techniques of balancing tasks in the distributed systems are divided mainly into three types. Those are *sender-initiated, receiver-initiated* and *symmetrically-initiated* (Yagoubi et al., 2006; Shah et al., 2007). In the *sender-initiated* models, the overloaded nodes transfer one or more of their tasks to more under-loaded nodes. In the *receiver-initiated* schemes, the under-loaded nodes request tasks to be sent to them from nodes with higher loads. In the *symmetrically-initiated* approach, both the under-loaded as well as the loaded nodes initiate the load transfers.

2.3. Load balancing performance metrics

The performance impact of any load balancing algorithm can be measured using the following performance metrics.

(1) *Makespan or execution time*: It is the total application execution time that is measured from the time the first job is sent to the Grid until the last job comes out of the Grid.
(2) *Average response time*: If n no. of jobs are processed by the system, then the average response time (ART) is given by

$$\text{Average Response Time (ART)} = \frac{1}{n}\sum_{i=1}^{n}(\text{Finish}_i + \text{Arrival}_i)$$

where the Arrival_i is the time at which the ith job arrives, and Finish_i is the time at which it leaves the system.

Table 1
Survey of load balancing techniques.

Algorithm	Proposed by: research focus/contribution/features	Compared algorithm	Performance metrics/ improvement	Gap/future work
Tree based approach				
In the tree based approach, the hierarchical load balancing method comes up with the dynamic tree based model of Grid for managing the workload. It decreases the amount of exchange messages in the Grid environment and thereby leads to the decrease in communication overhead. The load balancing algorithms based on this approach are found in Hao et al. (2012), Rathore and Channa, (2014), Qureshi et al. (2010), Yagoubi and Slimani (2006), (2007a), (2007b), Buyya and Murshed (2002a), (2002b), Nanthiya and Keerthika (2013) and Goswami and Sarkar (2013).				
EGDC	Hao et al. (2012): Pays attention towards deadline of tasks and presents a load balancing mechanism based on deadline control	WLB (Buyya and Murshed, 2002a, 2002b), LBEGS (Qureshi et al., 2010), FPLTF (Saha et al., 1995; Paranhos et al., 2003), Min–Min (Maheswaran et al., 1999), Max–Min (Maheswaran et al., 1999)	Finished jobs, unfinished jobs, makespan, resubmitted time	Considers bandwidth, resource processing ability, requirement of job
PLBA	Rathore and Channa (2014): Proposes a hierarchical load balancing technique based on variable threshold value	WLB (Buyya and Murshed, 2002a, 2002b), LBEGS (Qureshi et al., 2010), Min–Min (Rings et al., 2009), Max–Min (Suresh and Balasubramanie, 2013; Chen et al., 2013)	Response time, resource allocation efficiency, communication overhead time, makespan	Extents by adjusting the balanced threshold function
LBEGS	Qureshi et al. (2010): Proposes that the machine entity should be active and should participate in load balancing at its level, this enhancement in GridSim known as the Enhanced GridSim	WLB (Buyya and Murshed, 2002a, 2002b), LBGS (Yagoubi and Slimani, 2006, 2007a, 2007b)	Communication overhead, response time, percentage response time gain	Implements various other scheduling and fault tolerance techniques
LBGS	Yagoubi and Slimani (2006)', (2007a) and (2007b): Proposes a load balancing strategy based on a tree model, representation of a Grid architecture	Not compared	Average communication time	Not given
WLB	Buyya and Murshed (2002a) and (2002b): Discuss an object-oriented toolkit, called GridSim, for resource modeling and scheduling simulation	Not compared	Job completion rate, time utilization, budget utilization	Focuses on strengthening the network model by supporting various types of networks with different static and dynamic configurations and cost -based quality of services
HLBFT	Nanthiya and Keerthika (2013): Addresses the issues of resource failures and user deadline for distribution of the load	LBEGS (Qureshi et al., 2010)	Makespan, communication overhead, hit rate	Not given
NDFS	Goswami and Sarkar (2013): Proposes an algorithm to solve the prevailing problem of dynamic load balancing with respect to deadline of job submitted by the clients	WLB (Buyya and Murshed, 2002a, 2002b), LBGS (Yagoubi and Slimani, 2006, 2007a, 2007b)	Finished jobs	Focuses in the direction of varying number of processing elements, and reduction of communication overheads
Estimation based approach				
In the estimation based approach, we perform load balancing by estimating the expected finish time of a job on processors on each job arrival. The load balancing algorithms estimate the various system parameters such as the job arrival rate, CPU processing rate, and load on the processor and then balance the load by migrating jobs to the processors by taking into account the job transfer cost, resource heterogeneity, and network heterogeneity. The load balancing algorithms based on this approach are described in Anand et al. (1999), Shah et al. (2007) and Malarvizhi and Uthariaraj (2009).				
Algorithm	Proposed by: research focus/contribution/features	Compared algorithm	Performance metrics/improvement	Gap/future work
MELISA, LBA	Shah et al. (2007): Considers the job migration cost, resource heterogeneity, and network heterogeneity, performs load balancing by parameter estimation such as the expected finish time of a job, job arrival rate, CPU processing rate and load on the processor	PIA (Anand et al., 1999), ELISHA (Anand et al., 1999)	Total execution time, average response time	Extents by providing fault tolerance into the system
ELISHA	Anand et al. (1999): Uses estimated state information based upon periodic exchange of exact state information between neighboring nodes to perform load scheduling	PIA, NS, RS, NH (Ni and Hwang, 1985)	Mean response time, idle time/ elapsed time	Extents by studying the effect of limiting the buddy set to a fixed number of processors
HLB	Malarvizhi and Uthariaraj (2009): Considers problems such as scalability, heterogeneity of computing resources and considerable job transfer delay/communication cost for computational intensive jobs	MCT (Ritchie and Levine, 2003), PIA (Anand et al., 1999)	Average response time, average processing time	Considers precedence constraint among different tasks of a job and some fault tolerant measures
Optimization based approach				
In the optimization based approach, the Grids are utilized optimally using a good load balancing algorithm. This approach proposes two new distributed swarm intelligence inspired load balancing algorithms. One algorithm is based on ant colony optimization and the other algorithm is based on particle swarm optimization. Here, the goal of the load balancing is to find an optimal load distribution strategy for generic tasks on heterogeneous servers preloaded by different amounts of dedicated tasks such that the overall average response time of the generic applications is minimized. The load balancing algorithms based on this approach are available in Ludwig and Moallem (2011), Li (2008), Chen (2008), Rahmeh and Johnson (2010), Nasir et al. (2010), Moradi et al. (2010) and Nikkhah et al. (2010).				

Table 1 (*continued*)

Algorithm	Proposed by: research focus/contribution/features	Compared algorithm	Performance metrics/ improvement	Gap/future work
ANTZ, PRAC-TICALZ	Ludwig and Moallem (2011): Proposes two new distributed swarm intelligence inspired load balancing algorithms	SBA (Zhu et al., 1996)	Makespan, number of communications	Addresses the problem of dynamic resource failure and security in the Grid
Not named	Li (2008): Addresses the optimal load distribution problem in a non-dedicated Grid computing system with heterogeneous servers processing both generic and dedicated applications	Not compared	Average response time	Formulates for other nondedicated cluster or Grid computing systems such as clusters of clusters or multi-cluster systems where each server itself is a cluster
ACO, GJAP	Chen (2008): Considers the heterogeneity of Grid resources, the overhead of job transferring among computing nodes	FIFO,TABU (Armentano and Yamashita, 2000)	Makespan, machine usage	Focuses on deal with machine crash or failure by fault tolerance
BRS	Rahmeh and Johnson (2010): Introduces a latency reduction factor in the random sampling	Not compared	Communication latency, sampling length	Not given
EANT	Nasir et al. (2010): Focuses on pheromone trail update and trail limit, determine the best resource to be allocated to the jobs based on job characteristics and resource capacity, and at the same time to balance the entire resources	ANTZ (Moallem and Ludwig, 2009)	Average completion time	Not given
MCPLB	Moradi et al. (2010): Considers workclass, cost, deadline and herd behavior, suggestions on loading indexes and new resource conditions in accordance with synchronous neighborhood	RandLB, OLB, MCLB, Random (Zikos and Karatza, 2008), MCT (Ritchie and Levine, 2003)	Average response time, execution time, cost-percentage, task failure percentage	Not given
PLB, MEPLB, MCPLB, MCOSTPLB, MCCOSTPLB	Nikkhah et al. (2010): Considers workclass, cost, deadline and herd behavior, suggestions on loading indexes and new resource conditions in accordance with synchronous neighborhood	RandLB, ML, MET, MELB, MCLB, MCOST, MCOSTLB, MCCOST, MCCOSTLB, Random (Zikos and Karatza, 2008), MCT (Ritchie and Levine, 2003)	Average response time, execution time, cost-percentage, task failure percentage	Not given
Agent based approach				
In this approach, a combination of intelligent agents and multi-agent approaches is applied to both the local Grid resource scheduling and the global Grid load balancing. Each agent is a representative of a local Grid resource and it utilizes predictive application performance data with iterative heuristic algorithms to engineer local load balancing across multiple hosts. At a higher level, the agents cooperate with each other to balance workload using a peer-to-peer service advertisement and discovery mechanism. The load balancing algorithms based on this approach are described in Cao et al. (2003), (2005), Ahmad et al. (2004), Chen et al. (2004), Cao (2004), Salehi et al. (2006), Wang et al. (2006) and Salehi and Deldari (2006).				
Not named	Cao et al. (2003): An agent-based Grid management infrastructure is coupled with a performance-driven task scheduler that has been developed for local Grid load balancing	Not compared	Advance time of application execution completion, resource utilization, load balancing level	Test the scalability of the system
Not named	Ahmad et al. (2004): Presents the design and implementation of distributed analysis and load balancing system for hand-held devices using multi-agents system, also proposes a system, in which mobile agents will transport, schedule, execute and return results for heavy computational jobs submitted by handheld devices	Not compared	Time distribution	Not given
Not named	Chen et al. (2004): Introduces into the practical protein molecules docking applications, which run at the DDG, a Grid computing system for drug discovery and design	Not compared	Robustness	Concerns more elements in the algorithm other than be confined to only CPUs and network bandwidth
Not named	Cao (2004): Proposes to perform self-organizing load balancing of batch queuing jobs with no explicit QoS requirements across distributed Grid resources and also to evaluate quantitative performance using a modeling and simulation approach	Not compared	Ants, ants wandering steps, ants wandering style	Focuses on the refinement of the system prototype and the ant algorithm, discussions on security and data management
Not named	Cao et al. (2005): Combination of intelligent agents and multi-agent approaches is applied to both local Grid resource scheduling and global Grid load balancing. Here agents cooperate with each other to balance workload using a peer-to-peer service advertisement and discovery mechanism	Not compared	Total application execution time, average advance time of application execution completion, average load utilization rate, load balancing level	Extents the agent framework with new features such as automatic QoS negotiation, self-organizing coordination, semantic integration, knowledge-based reasoning and ontology based service brokering
MLBLM	Salehi et al. (2006): Here overloaded nodes get balances through layers. In the first layer, which is node-level, an efficient scheduler tries to use node's resources equally. The	Not compared	Efficiency, convergence speed, communication count	Plans to prove MLBM mathematically and to promote ant's intelligence and adaptation

	second layer, which is called neighbor-level, periodically scatters the extra load of overloaded nodes to a limited domain. The third layer, which is Grid-level, is a colony of intelligent ants which spread the regional extra load throughout the Grid			
Not named	Wang et al. (2006): Apply the agents to enable service-level load balancing and fault tolerance. To improve the scheduling efficiency, a degree of dependability is defined to concisely denote availability of the Grid resources and the Grid services	Not compared	Throughput, scheduling requests	Not given
Not named	Salehi and Deldari (2006): Provides more accurate load measurement/estimation method which relies on the time needed for executing current jobs, implemented on an agent-based resource management system, called ARMS	Not compared	Time overhead, efficiency, load balancing level	Discusses on security, billing contracts between agents when they exchange the load of their customers

Artificial life techniques

The artificial life techniques have been used to solve a wide range of complex problems in recent times. The power of these techniques stems from their capability in searching large search spaces, which arise in many combinatorial optimization problems, very efficiently. Due to their popularity and robustness, a genetic algorithm (GA), Simulated Annealing (SA), Fuzzy operators and tabu search (TS) are used to solve the Grid load balancing problem. The load balancing algorithms based on this approach are found in Akhtar (2007), Subrata et al. (2007), Ma (2010), Wu et al. (2011), Salimi et al. (2014), (2012) and Prakash and Vidyarthi (2011).

Not named	Akhtar (2007): Predicts the execution time for each task with respect to the resource it is assigned to. The prediction time is based on the current attributes of task, current and historical parameters, like load, memory of resources	Not compared	Makespan, correlation coefficient, root mean square error	Examine the application of the GA based algorithm
GA, TS	Subrata et al. (2007): Here adaptive memory is used to guide problem solving, also useful in situations where the solution space to be searched is huge, making sequential search computationally expensive and time consuming	BEST FIT, RANDOM (Zikos and Karatza, 2008), MIN–MIN (Maheswaran et al., 1999), MAX–MIN (Maheswaran et al., 1999), SUFFERAGE (Ibarra and Kim, 1977)	Makespan	Overcomes the drawback that they incur extra storage and processing requirement at the scheduling node
HGLBA	Ma (2010): Aims to assign proper tasks to processor according its performance, so as to minimize the time that execute the applied program, and to precisely estimate the load on the server, assigning new tasks to each server	MIN–MIN (Martino and Mililotti, 2004), MAX–MIN (Wolski et al., 1999)	Average fitness value, average response time, average finish time	Not given
OSLS	Wu et al. (2011): This approach circumvents the scalability of job scheduling problem by using an ordinal distributed learning strategy, and realizes multi-agent coordination based on an information sharing mechanism with limited communication	LLS (Galstyan et al., 2005), RS (Galstyan et al., 2005), SLS (Galstyan et al., 2005), DMMS (Freund et al., 1998)	Job arrival rate, average load of resources, makespan	Not given
FUZZY NSGA-II	Salimi et al. (2014): Improves the famous multi-objective genetic algorithm known as NSGA-II using fuzzy operators to improve quality and performance of task scheduling in the market-based Grid environment	NSPSO (Li, 2003)	Makespan, price	Not given
NSGA-II WITH FUZZY MUTATION	Salimi et al. (2012): Addresses scheduling problem of independent tasks in the market-based Grid where resource providers can request payment from users based on the amount of computational resource that used by them	NSGA-II (Deb et al., 2002; Coello and Lechuga, 2002), MOPSO (Deb et al., 2002; Coello and Lechuga, 2002)	Makespan, price	Not given
Not named	Prakash and Vidyarthi (2011): Suggests necessity of quantification of load and the objective function is derived based on the load distribution to the computational nodes	Not compared	Load balancing observation, load distribution observation	Incorporates with the scheduling algorithm to achieve better load balancing and better system utilization

Hybrid based approach

The hybrid load balancing method combines the principles of both the static and dynamic load balancing for addressing the problem of resource allocation. They use the metric of update interval for reducing the delay and deadlock. It reduces the waiting time of the jobs and assigns the priority. The load balancing algorithms based on this approach are found in Yan et al. (2009), Li et al. (2009) and Yan et al. (2007).

VF	Yan et al. (2009): Proposes a hybrid load balancing policy to integrate static and dynamic load balancing technologies. When a node reveals the possible inability to continue providing resources, the system will then obtain a new replacement node within a short time, to maintain system execution performance	FCFS (Ritchie and Levine, 2003), LIFO (Yang et al., 2003), CPU-BASED (Yang et al., 2003), RANDOM (Yang et al., 2003), MCT (Ritchie and Levine, 2003), MIN–MIN (Ritchie and Levine, 2003)	Task redistribution time, task completion time	Not given
HGA	Li et al. (2009): Proposes a novel load balancing strategy using a combination of static and dynamic load balancing strategies, combine a first-come-first-served algorithm	FCFS (Zomaya and Teh, 2001), DGA (Cao et al., 2005)	Makespan, average node utilization, mean square deviation,	Not given

Table 1 (*continued*)

Algorithm	Proposed by: research focus/contribution/features	Compared algorithm	Performance metrics/ improvement	Gap/future work
	with a special-designed GA to form a hybrid so as to take full advantage of their respective merits			
VF	Yan et al. (2007): Proposes a hybrid load balancing policy which integrated static and dynamic load balancing technologies to assist in the selection for effective nodes. If any selected node can no longer provide resources, it can be promptly identified and replaced with a substitutive node to maintain the execution performance and the load balancing of the system	FCFS (Ritchie and Levine, 2003; Cao et al., 2005), LIFO (Yang et al., 2003), CPU-BASED (Yang et al., 2003)	Task redistribution time, task completion time	Not given
Neighbor based approach The neighbor based approach is a dynamic load-balancing technique that allows the nodes to communicate and transfer tasks with their neighbors so that the whole system is balanced after a number of iterations. Since this technique does not require a global coordinator, it is inherently local, fault tolerant and scalable. The load balancing algorithms based on this approach are described in Balasangameshwara and Raju (2013), (2012) and (2010).				
PD_MinRC	Balasangameshwara and Raju (2013): Integrate the proposed load-balancing approach with fault-tolerant scheduling namely MinRC and develop a performance-driven fault-tolerant load-balancing algorithm or PD_MinRC for independent jobs	PD_NoMinR, DA (Lu et al., 2007), ASAP (Zhu et al., 2011)	Response time, load balancing level, back up response time, replication cost	Consider issues related to security
AlgHybrid_LB	Balasangameshwara and Raju (2012): Takes into account Grid architecture, computer heterogeneity, communication delay, network bandwidth, resource availability, resource unpredictability and job characteristics. AlgHybrid_LB juxtaposes the strong points of neighbor-based and cluster based load balancing algorithms	MCT (Braun et al., 2001), MIN–MIN (Braun et al., 2001)	Job redistribution time, job completion time, average response time	Consider issues related to security
OP	Balasangameshwara and Raju (2010): Proposes a dynamic, symmetric initiated model which takes a decentralized approach to load balancing, the computing nodes in a cluster interact with each other through a symmetrically initiated strategy	Nobel Fault tolerant technique	Mean response time	Study the impact of communication delay on the model under varying load conditions
Partitioning based approach The partitioning of an adaptive Grid for distribution over parallel processors is considered in the context of adaptive multilevel methods for solving partial differential equations. The efficient parallel execution of Grid-oriented scientific calculations requires the partitioning of the Grid that minimizes both the load imbalance and interprocessor communication. For unstructured static Grids, good partitions are obtained with the recursive spectral bisection heuristic, applied to the interdependency graph of the Grid. The load balancing algorithms based on this approach are available in Keyser and Roose (1995), Mitchell (2007), Driessche and Roose (1995) and Kejariwal and Nicolau (2005).				
Not named	Keyser and Roose (1995): The issues involved in the parallel implementation of an unstructured multi-Grid algorithm with run-time Grid refinement for the steady Euler equations is discussed on a distributed memory computer	Not compared	Mathematically proof	Reduces the double flux computation by increasing the size of the parts
Not named	Mitchell (2007): Uses a tree representation of the refinement process with weights representing the amount of work associated with each element. The method applies to almost all types of elements and refinement strategies in two dominant for a large number of processors	Not compared	Mathematically proof	Not given
Not named	Driessche and Roose (1995): For Grid-oriented problems as a graph partitioning problem, proposes the dynamic load balancing problem by extending the interdependency graph of the mesh with virtual vertices and edges that represent the transfer costs	Not compared	Mathematically proof	Multilevel implementations of the spectral bisection algorithm can easily be applied to our alternative spectral bisection heuristic that are an order of magnitude faster
Not named	Kejariwal and Nicolau (2005): Presents a geometric approach for partitioning N-dimensional nonrectangular iteration spaces for optimizing performance on heterogeneous parallel processor systems	CAN PARTITIONING TECHNIQUE (Sakellariou, 1996)	Mathematically proof	Extends to partition iteration spaces at run-time
Others				
RADIS	Viswanathan (2007): Specially designed to handle large volumes of computationally intensive arbitrarily divisible	Not compared	Load arrival rate	A fading memory could be plugged.

	loads submitted for processing at Grid systems involving multiple processing nodes, adopts the divisible load paradigm, referred to as the divisible load theory (DLT)			
DLT	Bharadwaj et al. (2003): Divisible load theory is a methodology involving the linear and continuous modeling of partitionable computation and communication loads for parallel processing. It adequately represents an important class of problems with applications in parallel and distributed system scheduling.	Not compared	Speed up curves, optimal finish time curves	Not given
A^2DLT	Othman et al. (2008): Presents a new divisible load balancing model known as adaptive ADLT (A^2DLT) for scheduling the communication intensive Grid applications	CDLT (Wong et al., 2003), ADLT (Othman et al., 2007)	Makespan	Integrate in the existing data Grid schedulers in order to improve the performance
Not named	Yang (1997): In order to balance loads among different processors, we employ small sub domains with fine Grids for rapidly-changing solution areas, and big sub domains with coarse Grids for slowly-changing solution areas	Not compared	Mathematically proof	Dynamic changes in domain decompositions
Not named	Fatta and Berthold (2007): Presents a distributed computing framework for problems based on a search strategy. It employs a decentralized dynamic load balancing technique that is enhanced by global statistics to cope with highly irregular problems	Not compared	Running time, fairness index, relative load imbalance index, speed up	Adopts a decentralized solution for the centralized server for job statistics
Not named	Mezmaz et al. (2007): Proposes a new dynamic load balancing approach for the parallel branch and bound algorithm on the computational Grid. The approach is based on a particular numbering of the tree nodes allowing a very simple description of the work units distributed during the exploration.	Not compared	Mathematically proof	Extends the scalability limits and exploits the load balancing strategies to more and more processors
GT	Subrata et al. (2008): Combines the inherent efficiency of the centralized approach and the fault-tolerant nature of the decentralized approach. The algorithm does not assume any particular distribution for service times of tasks, it only requires the first and second moments of the service times as input.	PS (Chow and Kohler, 1979)	Average task completion time, fairness	Not given
ARI	Fei et al. (2009): Focuses on balancing the workload by transferring jobs to idle sites at prime time to minimize the response time and maximize the resource utilization and power management by switch the idle sites to sleeping mode at non-prime time to minimize the energy consume.	RI (Shivaratri et al., 1992), SI (Shivaratri et al., 1992)	Average response time, throughput, utilization	Extents by providing fault tolerance into the resource management system
CCOOP, NCOOPC	Penmatsa and Chronopoulos (2011): Using cooperative game theory, CCOOP algorithm provides fairness to all the jobs in a single-class job distributed system and using non-cooperative game theory, NCOOPC algorithm provides fairness to all users in a multi-user job distributed system by taking the communication costs into account	OPTIM (Kim and Kameda, 1992), PROP (Chow and Kohler, 1979), GOS (Kim and Kameda, 1990), PROP_M (Kim and Kameda, 1992)	Expected response time, fairness index, communication time	Provides fairness by taking the current system load into account based on dynamic game theory and also consider other aspects of heterogeneity
Not named	Anousha and Ahmadi (2013): Proposes new scheduling algorithm based on well known task scheduling algorithms, Min–Min. The proposed algorithm firstly estimates of the completion time of the tasks on each of resources and then selects the appropriate resource for scheduling	MIN–MIN (He et al., 2003), MAX–MIN (Etminani and Naghibzadeh, 2007)	Makespan, average resource utilization rate	Apply other issues like deadlines on tasks and resources
Not named	Arora et al. (2002): Considers the overheads of coordination and communication between the Grid nodes which were assumed to be N-resource servers that varied in their respective capacities across resources, introduces a new load balance Triggering Policy based on the endurance of a node reflected by its current queue length.	Not compared	Mean node capacity, mean communication time, execution time	Not given
DLB	Lu et al. (2006): Operates on two job scheduling and load balancing policies. The first is Instantaneous Distribution Policy, which tries to control the job processing rate on each site in the system. The second is Load Adjustment Policy, which tries to continuously reduce load difference among a site and its neighbor sites. Considers the different	LOCAL, RANDOM (Zikos and Karatza, 2008)	Average response time	Model the impact of accuracy of job execution time estimation, study the execution scheme for data distribution, consider the resource requirements of jobs, the network and hardware failure

Table 1 (*continued*)

Algorithm	Proposed by: research focus/contribution/features	Compared algorithm	Performance metrics/ improvement	Gap/future work
	network communication delays between sites can reduce the cost of load movement, and enable quick response to load imbalances			
Not named	Rajavel (2010): Provides the decentralized load balancing in both meta-scheduler and cluster or resource level. The Triggering Policy is used to initiate the load balancing algorithm, which determines the appropriate time period to start the load balancing operation using the boundary value and threshold value approach.	NORMAL LOAD BALANCER	Job waiting time	Working towards load balancing and job migration between the meta-scheduler in the real Grid environment.
LPAS_DEC	Azzoni and Down (2009): Uses an effective mechanism for state information exchange, which significantly reduces the communication overhead, while quickly updating the state information in a decentralized fashion.	MCT (Ritchie and Levine, 2003)	Average task completion time	Not given
AlgMinT, AlgMinD	Zheng et al. (2008): Study the effect of pricing on load distribution by considering a simple pricing function. Develop distributed algorithms to decide which group the load should be allocated to, taking into account the communication cost among groups. These algorithms use different information exchange methods and a resource estimation technique to improve the accuracy of load balancing.	NASH (Grosu and Chronopoulos, 2005), NASHP (Penmatsa and Chronopoulos, 2005)	Mean response time, mean cost	Not given
PLBPs	Fathy and Zoghdy (2012): Proposes a fully decentralized two level load balancing policy for balancing the workload in a multi-cluster Grid environment where clusters are located at administrative domains, takes into account the heterogeneity of the Grid computational resources, and it resolves the single point of failure problem which many of the current policies suffer from.	No. LB, Random (Zikos and Karatza, 2008), Min (Balasangameshwara and Raju, 2010)	Mean response time	Study the effect of the length of information update periodical interval at the global scheduler and local scheduler, increase the reliability of the proposed policy by considering some fault tolerance measures
HLB	Lu and Zomaya (2007): Integrates static and dynamic approaches to make load distribution and redistribution driven by performance benefit jobs, achieves a balance between the inherent efficiency of centralized approach, and the autonomy, load balancing and fault tolerant features offered by distributed approach	MCT (Maheswaran et al., 1999)	Average response time	Proposes job execution cost-estimation to reduce the possible impact
PAD, FZF-PAD	Zikos and Karatza (2009): Study the performance of three scheduling policies at Grid scheduler level i.e. Basic Hybrid, PAD, FZF-PAD which utilize dynamic site load information to route nonclairvoyant jobs to heterogeneous sites, in a 2-level Grid system	H_GS (Zikos and Karatza, 2008)	Response time, load information traffic, resource utilization fairness	Apply optimizations on scheduling policies at Grid scheduler level, examine additional metrics such as throughput for feedback between sites and Grid scheduler, simulate the experiment in case of highly variable job service demands
AWLB	Korkhov et al. (2009): Proposes to enhance the quality of handling multi-task jobs in Grid environment by integrating the AWLB developed for parallel applications on heterogeneous resources	FIFO	Iteration time, balancing speed up, processors capacity	Plans to enhance the resource selection and matchmaking mechanisms by further development of the automated application performance analysis
CPU_PM	Singh and Awasthi (2011): Focuses on dynamic load balancing on a network of workstations and to develop a distributed scheduling algorithm for load balancing which takes heterogeneity CPU, memory and disk resource into account	CM_PM, IO CM_RE, IO CM_PM	Mean slowdown	Evaluate performance of the proposed scheme using feedback control technique
Not named	Karthikumar et al. (2013): Design a fair scheduling approach with equal opportunity to all the jobs, follows the hybrid scheduling by calculating the residue value for each job for a number of iterations until the residue gets down to zero	Not compared	Fair rates	Design an optimal fault tolerance approach based on check-pointing, classify the incoming job request into local and external site request to optimize the task completion by inducing priority to the jobs
Not named	Lee and Huang (2002): Review the effects of the spatial and temporal heterogeneity on performance of a target task	Not compared	Average parallel execution time	Develop an application to channel bandwidth allocation in mobile computing

DA	Lu et al. (2006): Consider heterogeneity of sites, makes more powerful sites carry more loads, as jobs executed at fast sites are more likely to execute at high speed, taking into account the different network communication delays between sites can reduce the cost of load movement, and enable quick response to load imbalances	NN (Sanders, 1999; Xu et al., 1995)	Average response time	Study better approaches for selection of partner sites
Not named	Wang and Wang (2005): Enhances orbus1.1 software with load balancing service on request chain processing, which should to be emphasized in Grid workflow	Not compared	Fault tolerant service	Not given
DLB	Lu et al. (2007): Uses site desirability for processing power and transfers delay to guide load assignment and redistribution, transfer and location policies are a combination of the Instantaneous Distribution Policy (IDP) and the Load Adjustment Policy (LAP) that are performance driven to minimize execution cost.	LOCAL, BN, RANDOM (Zikos and Karatza, 2008),	Average response time	Model the impacts of accuracy of job execution time estimation, utilize migration threshold dynamically based on real-time observation of load behavior of system resources, consider network and hardware failure
BILB	Rzadca and Trystram (2009): Proposes a simple mathematical model for such systems and a novel function for computing the cost of the execution of foreign jobs depends both on the size of a job and on the local load	Not compared	Mathematically proof	Enhance our algorithm in order to reduce the dispersion of the results observed in the experiments
AWLB	Korkhov et al. (2009): Suggests a hybrid resource management approach operates on both application and system levels, combines user-level job scheduling with dynamic workload balancing algorithm that automatically adapts a parallel application to the heterogeneous resources	FIFO	Balancing speed up, execution time	Test other connectivity schemes, such as the different Master–Worker modes, as well as Mesh, Ring and Hypercube topologies
Not named	Zoghdy and Aljahdali (2012): Proposes a two-level load balancing policy for the multi-cluster Grid environment where computational resources are dispersed in different clusters which are located in different local area networks	RANDOM (Zikos and Karatza, 2008), UNIFORM (Zikos and Karatza, 2008)	System mean response time	Not given
DLBA	Suri and Singh (2010): Performs intra-cluster and inter-cluster (Grid) load balancing, considers load index as well as other conventional influential parameters at each node for scheduling of tasks	WDLBA	Execution time, cost	Intend to use the new load balancing algorithm in an actual environment for practical evaluation
Not named	Nasir et al. (2010): Based on the combination of local pheromone update and trail limits	Not compared	Mathematically proof	Not given
mDELAY	Mehta et al. (2010): Presents a modified delay strategy to significantly enhance delay-based scheduling algorithm, for delaying the scheduling of new jobs instead of dispatching them to one of the overloaded workstations	DELAY (Hui and Chanson, 1999), ROUND ROBIN	Average completion time	Proposes a two-level service based decentralized framework to implement the mDELAY scheduling strategy for improved performance over the centralized scheduler
MACO	Bai et al. (2010): Here, multiple ant colonies work together and exchange information to collectively find solutions with a objective of minimizing the execution time of tasks and the degree of imbalance of computing nodes	FCFS (Zomaya and Teh, 2001), ACS	Makespan	Not given
PLBA	Rathore and Chana (2013): Proposes technique based on variable threshold value which can be found out using load deviation is responsible for transfer the task and flow of workload information, introduces a sender initiated policy to reduce the communication overhead	WLB (Buyya and Murshed, 2002; Buyya and Murshed, 2002), LBEGS (Qureshi et al., 2010)	Response time, resource allocation efficiency	Adjusts the function of the balance threshold and make it more adaptive to differing environments
PROPOSED	Nandagopal et al. (2010): Addresses the problem of load balancing using Min-Load and Min-Cost policies while scheduling jobs to the resources in multi-cluster environment, develops a heuristic taking both the resource load and the network cost into consideration to evaluate the benefits of scheduling jobs to resources in different clusters	RANDOM (Zikos and Karatza, 2008)	Response time, slow down	Considers some fault tolerant measures to increase the reliability of our algorithm
HJS	Reddy and Roy (2012): Addresses two common parameters, namely CPU utilization and heap memory are employed for load balancing and a computational intensive job is executed on a Grid test bed deployed using Gridgain.	FJS	Total execution time	Not given
Not named	Erciyes and Payli (2005): The Grid consists of clusters and each cluster is represented by a coordinator. Each coordinator first attempts to balance the load in its cluster and if	Not compared	Mathematically proof	Implements the recovery procedures

Table 1 (*continued*)

Algorithm	Proposed by: research focus/contribution/features	Compared algorithm	Performance metrics/ improvement	Gap/future work
	this fails, communicates with the other coordinators to perform transfer or reception of load			
Not named	Mello and Senger (2006): Distributes equally the workload of tasks of parallel applications over Grid computing environments	RANDOM (Zhou and Ferrari, 1987), LOWEST (Zhou and Ferrari, 1987), CENTRAL (Zhou and Ferrari, 1987), DPWP (Araujo et al., 1999; Araujo et al., 1999), TLBA (Mello et al., 2004), GAS (Senger et al., 2005)	Average response time	Not given
DLB	Liao et al. (2010): Presents a Grid-based dynamic load balancing approach for data-centric storage for wireless sensor networks. This scheme is based on two mechanisms, the cover-up and the multi-threshold. The cover-up mechanism can adjust to another storage node dynamically when a storage node is full, while the multi-threshold mechanism can spread the data into several storage for load balancing of the sensor nodes	GHT (Ratnasamy et al., 2002)	Total energy consumption, average of storage space, hotspot storage space, standard deviation of storage, dropped events	Not given
Not named	Ma et al. (2011): Incorporates functional modules Buffer Management and Load Balancing Management over a Grid networking platform, to buffer the read data and share the middleware loading, thereby solving the overloading issues in RFID applications	TRADITIONAL RFID SYSTEM (Park et al., 2007; Cui and Chae, 2007; Pan et al., 2005), CONNECTION POOL MECHANISM (Park et al., 2007; Park et al., 2007)	Processing time, packet loss ratio	Adjusts the number of readers and middleware hosts to enable the system to reach the optimal efficiency, concerns about the security problem
Not named	Khanli et al. (2012): Uses the subtraction of forward and backward ants as a competency rank to take the priority of the sites, and also uses a control word to search the suitable resource as well. The main purpose is to devote jobs to the existing resources based on their processing power.	B&B (Mezmaz et al., 2007)	Makespan, tardiness, cost	Increases the number of existing resources and the jobs entered to the environment can be increased. Also devotes the jobs to the existing resources in the form of grouping
Not named	Erdil and Lewis (2012): Describes information dissemination protocols that can distribute load, without using load rebalancing through job migration, which is more difficult and costly in large-scale heterogeneous Grids.	Not compared	Query satisfaction, packet overhead, resource utilization, reservation requests	Not given
M^2ON, M^2ON^*	Jiang et al. (2009): Presents Min-cost and Max-flow Channel based Overlay Network (M^2ON), here the communication capability is denoted as M^2C (Min-cost and Max-flow Channel) which is obtained using a Labeled Tree Probing (LTP) method	BON (Bridgewater et al., 2007)	Mean executing time	Obtain accurate topology matching by a better and more flexible fusion function which in turn further optimize the load balancing process

(3) *Finished and unfinished jobs*: The finished rate of jobs or hit rate can be defined as the number of jobs that are successfully completed on the Grid system on the first schedule. Some of the jobs may not be executed before their deadline. The numbers of jobs that cannot be finished on time (unfinished jobs) are also selected as the standard performance criteria.

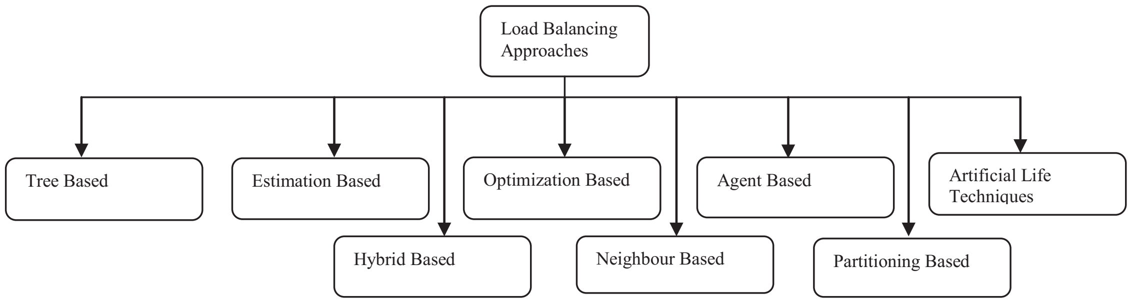

Fig. 3. Grid load balancing approaches.

Table 2 Load balancing applications.

Application	Proposed by: research focus/contribution/features	Gap/future work
Communication Data Management	Lee (2004): Describes an intelligence balancing for communication data management. The intelligence balancing allows to execute a complex large-scale Grid computing system and share dispersed data assets collaboratively, focuses on the intelligence balancing to each Grid component and various degrees of intelligence	Not given
Successive Over Relaxation (SOR)	Dobber et al. (2004): Analyze the impact of the fluctuations in the processing speed on the performance of Grid applications as resources are shared among numerous applications, and therefore, the amount of resources available to any given application highly fluctuates over time	Improvise the running times for more complex computation-intensive applications with more complex structures
Aligning Long DNA Sequences	Chen and Schmidt (2004): Apply the computational Grid concept to aligning long DNA sequences and study the new load balancing techniques for hierarchical Grids called "scheduler-worker" under disturbance and for different levels of application-level inter-cluster bandwidths	Identifies more biology applications that profit from hierarchical Grid systems and presents more efficient parallel models to map these applications onto hierarchical Grid systems
Scatter Operations	Genaud et al. (2004): Modifies of the data distributions used in scatter operations, presents a general algorithm which finds an optimal distribution of data across processors, a quicker guaranteed heuristic relying on hypotheses on communications and computations and a policy on the ordering of the processors	Not given
Scatter Operations	Genaud et al. (2003): Study the replacement of scatter operations with parameterized scatters, allow custom distributions of data	Not given
Barnes-Hut Algorithm	Alt et al. (2005): Proposes a high-level approach to Grid application programming, based on generic components or skeletons with prepackaged parallel and distributed implementations and integrated load-balancing mechanisms, present an experimental java-based programming system with skeletons and use it on a non-trivial, dynamic application, the Barnes-Hut algorithm	Not given
Lattice Boltzmann Model	Farina et al. (2006): Modifies the original Lattice Boltzmann model to approximate a diffusive phenomenon that suitably solves the dynamic load balancing problem	Not given
Cosmology SAMR Simulations	Lan et al. (2006): Design to improve the performance of distributed cosmology simulations, focuses on reducing the redistribution cost through a hierarchical load balancing approach and a run-time decision making mechanism	Investigates multi-level approach and evaluate it against the proposed two-level approach
Distributed and Integrated Power Systems	Al-Khannak and Bitzer (2007): Develop an interface between the power systems and the Grid computing which interacts with other power systems connected to the Grid computing. Grid computing resources perform real time load forecasting where the results will be returned to each power system for decentralized load balancing operations	Not given
Grid-based Virtual Reactor	Korkhov et al. (2008): Introduce a generic technique for adaptive load balancing of parallel applications on heterogeneous resources and evaluate it using a case study application: a Virtual Reactor, contains a number of parallel solvers originally designed for homogeneous computer clusters that needed adaptation to the heterogeneity of the Grid	Integrates the adaptive load-balancing algorithm with the DIANE user-level scheduling system, which extends the testing ground to the multitude of real applications executed on the EGEE Grid
HLA-Based Simulations	Boukerche and Grande (2009): Supports the re-distribution of load for HLA-based simulations running on large-scale distributed systems	Consider the simulation intercommunication to minimizing the communication
High Level Architecture (HLA) Based Simulations	Grande and Boukerche (2011): Proposes to evenly distribute the load of large-scale HLA based simulations on non-dedicated, heterogeneous environments when computational and communication imbalances are present	Detects communicative federates, achieve better detection of and reactivity to load imbalances by different communications and computation balancing techniques

(4) *Resubmitted time or task redistribution time*: In a Grid, some Gridlets cannot be finished at the first resource scheduling, but can be scheduled again as its request. The sum of resubmitted time is another standard for our test.
(5) *Communication overhead*: The communication overheads are calculated by counting the number of messages over Internet, LAN, and Machine.
(6) *Efficiency*: This is the property of any load balancing algorithm which relate to the amount of resources used by the algorithm. An algorithm must be analyzed to determine its resource usage. For the maximum efficiency, the algorithm should minimize the resource usage.
(7) *Throughput*: Throughput is the amount of jobs that a system can execute in a given time period.
(8) *Fairness*: Access to any resources is formally rated by a fairness measure. The fairness measures or metrics determine whether users or applications are receiving a fair share of the system's resources.
(9) *Robustness*: It is the ability of a computer system to cope with errors during the execution. Robustness can also be defined as the ability of an algorithm to continue operating despite abnormalities in input, calculations, etc.
(10) *Latency*: It is the time interval between the stimulation and response, or, from a more general point of view, measure of the time delay or waiting that is experienced by some jobs on the system.

3. Load balancing survey

Table 1 summarizes various load balancing techniques that have been proposed over the years for usage in the Grid. The load balancing techniques have been appropriately classified under different approaches as shown in Fig. 3. Their research focus, contribution, features, compared model, performance metrics, improvement, gap and future work have been analyzed.

4. Load balancing applications

Various load balancing applications are discussed below in Table 2.

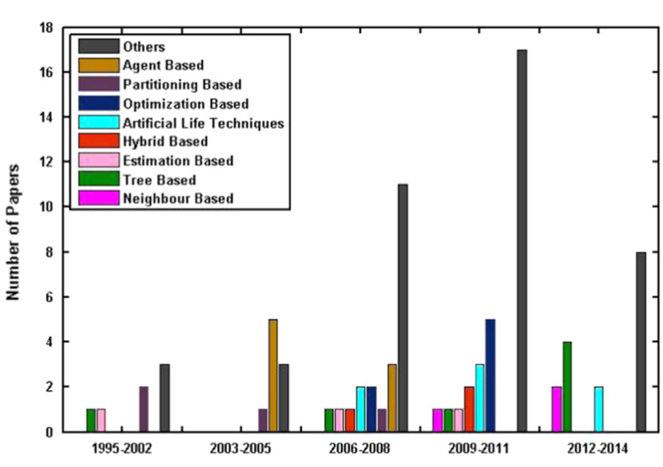

Fig. 4. Adoption graph of load balancing techniques.

5. Adoption graph of load balancing techniques

On the basis of the survey, an analysis of trends in publication of load balancing techniques for Grid has been described in Fig. 4.

6. Conclusions

This paper presents an extensive survey of various load balancing techniques that have been proposed over the years for usage in the Grid. The load balancing techniques that are available in the literature have been appropriately classified under different headings. The algorithm, research focus, contribution, features, compared model, performance metrics, improvement, gap and future work of each load balancing technique have been analyzed and presented.

References

Ahmad N, Ali A, Anjum A, Azim T, Bunn J, Hassan A, Ikram A, Lingen F, McClatchey R, Newman H, Steenberg C, Thomas M, Willers I. Distributed analysis and load balancing system for grid enabled analysis on hand-held devices using multi-agents systems. In: Proceedings of the grid and cooperative computing workshops. Lecture Notes In Computer Science; 2004, vol. 3251. p. 947–50.
Akhtar Z. Genetic load and time prediction technique for dynamic load balancing in grid computing. Inf Technol J 2007;6(7):978–86.
Al-Khannak R, Bitzer B. Load balancing for distributed and integrated power systems using grid computing. In: Proceedings of the international conference on clean electrical power (ICCEP'07); May 2007. p. 123–7.
Alt M, Muller J, Gorlatch S. Towards high-level grid programming and load-balancing: a Barnes-Hut case study. In: Proceedings of the 11th international Euro-Par conference. Lecture Notes in Computer Science, vol. 3648; 2005. p. 391–400.
Anand L, Ghose D, Mani V. ELISA: an estimated load information scheduling algorithm for distributed computing systems. Comput Math Appl 1999;37:57–85.
Anousha S, Ahmadi M. An improved Min–Min task scheduling algorithm in grid computing. In: Proceedings of the international conference on grid and pervasive computing (GPC'13). Lecture Notes in Computer Science; 2013, vol. 7861. p. 103–13.
Araujo APF, Santana MJ, Santana RHC, Souza PSL. A new dynamical scheduling algorithm. In: Proceedings of the international conference on parallel and distributed processing techniques and applications (PDPTA'99). Las Vegas, Nevada, USA; 1999.
Araujo APF, Santana MJ, Santana RHC, Souza PSL. DPWP: a new load balancing algorithm. In: Proceedings of the 5th international conference on information systems analysis and synthesis (ISAS'99). Orlando, USA; 1999.
Armentano VA, Yamashita DS. Tabu search for scheduling on identical parallel machines to minimize mean tardiness. J Intell Manuf 2000:453–60.
Arora M, Das SK, Biswas R. A de-centralized scheduling and load balancing algorithm for heterogeneous grid environments. In: Proceedings of the international conference on parallel processing workshops (ICPPW'02); 2002. p. 1–7.
Azzoni IA, Down DG. Decentralized load balancing for heterogeneous grids. In: Proceedings of the international conference on computation world; November 2009. p. 545–50.
Bai L, Hu YL, Lao SY, Zhang WM. Task scheduling with load balancing using multiple Ant Colonies optimization in grid computing. In: Proceedings of the sixth international conference on natural computation (ICNC'10); 2010. p. 2715–19.
Balasangameshwara J, Raju N. A decentralized recent neighbour load balancing algorithm for computational grid. Int J ACM Jordan 2010;1(3):128–33.
Balasangameshwara J, Raju N. A hybrid policy for fault tolerant load balancing in grid computing environments. J Netw Comput Appl 2012;35:412–22.
Balasangameshwara J, Raju N. Performance-driven load balancing with a primary-backup approach for computational grids with low communication cost and replication cost. IEEE Trans Comput 2013;62(5).
Balasangameshwara J, Raju N. A fault tolerance optimal neighbor load balancing algorithm for grid environment. In: Proceeding of the international conference on computational intelligence and communication systems; 2010. p. 428–33.
Berman F, Fox G, Hey AJ. Grid computing: making the global infrastructure a reality. New York: Wiley; 2003.
Bharadwaj V, Ghose D, Robertazzi TG. Divisible load theory: a new paradigm for load scheduling in distributed systems. Clust Comput 2003;6:7–17.
Boukerche A, Grande RED. Dynamic load balancing using grid services for HLA-based simulations on large-scale distributed systems. In: Proceedings of the 13th IEEE/ACM international symposium on distributed simulation and real time applications; 2009. p. 175–83.
Braun T, Siegel HJ, Beck N, Boloni L, Maheswaran M, Reuther A. A comparison of eleven static heuristics for mapping a class of independent tasks onto

heterogeneous distributed computing systems. J Parallel Distrib Comput 2001;61(6):810–37.
Bridgewater J, Boykin P, Roychowdhury VP. Balanced overlay networks (BON): an overlay technology for decentralized load balancing. IEEE Trans Parallel Distrib Syst 2007;18(8):1122–34.
Buyya R, Murshed M. GridSim: a toolkit for the modeling and simulation of distributed management and scheduling for Grid computing (Technical report). Monash University; 2002.
Buyya R, Murshed M. GridSim: a toolkit for the modeling and simulation of distributed management and scheduling for Grid computing. J Concurr Comput: Pract Exp 2002;14:13–5.
Cao J, Spooner DP, Jarvis SA, Nudd GR. Grid load balancing using intelligent agents. Future Gener Comput Syst 2005;21:135–49.
Cao J, Spooner DP, Jarvis SA, Saini S, Nudd GR. Agent-based grid load balancing using performance-driven task scheduling. In: Proceedings of the international parallel and distributed processing symposium (IPDPS); 2003.
Cao J. Self-organizing agents for grid load balancing. In: Proceedings of the 5th IEEE/ACM international workshop on grid computing; November 2004. p. 388–95.
Chen C, Schmidt B. Load balancing for hierarchical grid computing: a case study. In: Proceedings of the 11th international conference on high performance computing (HiPC'04). Lecture Notes in Computer Science; 2004, vol. 3296. p. 353–62.
Chen H, Wang F., Helian N, Akanmu G. User-priority guided min-min scheduling algorithm for load balancing in cloud computing. In: Proceedings of national conference on parallel computing technologies (PARCOMPTECH); 2013.
Chen S, Zhang W, Ma F, Shen J, Li M. A novel agent-based load balancing algorithm for grid computing. In: Proceedings of the grid and cooperative computing workshops. Lecture Notes in Computer Science; 2004, vol. 3252. p. 156–63.
Chen Y. Load balancing in non-dedicated grids using ant colony optimization. In: Proceedings of the 4th international conference on semantics, knowledge and grid; 2008: p. 279–86.
Chow YC, Kohler WH. Models for dynamic load balancing in a heterogeneous multiple processor system. IEEE Trans Comput 1979;28:354–61.
Coello CAC, Lechuga MS. MOPSO: a proposal for multiple objective particle swarm optimizations. In: Proceeding of the congress on evolutionary computation (CEC'2002). Piscataway, New Jersey; May 2002, vol. 2. p. 1051–56.
Cui JF, Chae HS. Agent-based design of load balancing system for RFID middlewares. In: Proceedings of the IEEE international workshop on future trends of distributed computing systems; March 2007. p. 21–30.
Deb K, Agrawal S, Pratap A, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6:182–97.
Dobber M, Koole G, Mei RV. Dynamic load balancing for a grid application. In: Proceedings of the 11th International conference on high performance computing (HiPC'04). Lecture Notes in Computer Science; 2004, vol. 3296. p. 342–52.
Driessche RV, Roose D. An improved spectral bisection algorithm and its application to dynamic load balancing. Parallel Comput 1995;21:29–48.
Erciyes K, Payli RU. A cluster-based dynamic load balancing middleware protocol for grids. In: Proceedings of the international conference on advances in grid computing. Lecture Notes in Computer Science; 2005, vol. 3470. p. 805–812.
Erdil DC, Lewis MJ. Dynamic grid load sharing with adaptive dissemination protocols. J Supercomput 2012;59:1139–66.
Etminani K, Naghibzadeh M. A Min–min Max–min selective algorithm for grid task scheduling. in: Proceeding of the 3rd IEEE/IFIP international conference on internet. Uzbekistan; 2007.
Farina F, Cattaneo G, Dennunzio A. Grid and HPC dynamic load balancing with lattice boltzmann models. In: Proceedings of the international conference on the move to meaningful internet systems, Part-II. Lecture Notes in Computer Science; 2006, vol. 4276. p. 1152–62.
Fathy S, Zoghdy E. A hierarchical load balancing policy for grid computing environment. Int J Comput Netw Inf Secur 2012;5:1–12.
Fatta GD, Berthold MR. Decentralized load balancing for highly irregular search problems. Microprocess Microsyst 2007;31:273–81.
Fei Y, Changjun J, Rong D, Jianjun Y. Grid resource management policies for load-balancing and energy-saving by vacation queuing theory. Comput Electr Eng 2009;35:966–79.
Foster I. What is grid? A three point checklist Argonne National Laboratory and University of Chicago; 2002.
Foster I, Kesselman C, editors. The grid: blueprint for a future computing infrastructure. Morgan Kaufmann Publishers; 1999.
Freund RF, Gherrity M, Ambrosius S. Scheduling resources in multiuser, heterogeneous, computing environments with SmartNet. In: Proceedings of the 7th IEEE heterogeneous computing workshop (HCW'98). Orlando, USA; March 1998. p. 184–99.
Galstyan A, Czajkowski K, Lerman K. Resource allocation in the grid with learning agents. J Grid Comput 2005;3:91–100.
Genaud S, Giersch A, Vivien F. Load-balancing scatter operations for grid computing. Parallel Comput 2004;30:923–46.
Genaud S, Giersch A, Vivien F. Load-balancing scatter operations for grid computing. In: Proceedings of the international parallel and distributed processing symposium (IPDPS'03); April 2003. p. 1–10.
Goswami S, Sarkar AD. A comparative study of load balancing algorithms in computational grid environment. In: Proceedings of the 5th IEEE international conference on computational intelligence, modelling and simulation; 2013: p. 99–04.
Grande RED, Boukerche A. Dynamic balancing of communication and computation load for HLA-based simulations on large-scale distributed systems. J Parallel Distrib Comput 2011;71:40–52.
Grosu D, Chronopoulos AT. Noncooperative load balancing in distributed systems. J Parallel Distrib Comput 2005;65(9):1022–34.
Hao Y, Liu G, Wen N. An enhanced load balancing mechanism based on deadline control on GridSim. Future Gener Comput Syst 2012;28:657–65.
He X, Sun XH, Laszewski GV. QoS guided Min–min heuristic for grid task scheduling. J Comput Sci Technol 2003;18:442–51.
Hui CC, Chanson ST. Improved strategies for dynamic load balancing. IEEE Concurr 1999;7.
Ibarra OH, Kim CE. Heuristic algorithms for scheduling independent tasks on nonidentical processors. J ACM 1977:280–9.
Jiang W, Baumgarten M, Zhou Y, Jin H. A bipartite model for load balancing in grid computing environments, Front. Comput Sci China 2009;3(4):503–23.
Karthikumar SK, Preethi MU, Chitra P. Fair scheduling approach for load balancing and fault tolerant in grid environment. In: Proceedings of the IEEE international conference on emerging trends in computing, communication and nanotechnology (ICECCN'13); 2013. p. 446–51.
Kejariwal A, Nicolau A. An efficient load balancing scheme for grid-based high performance scientific computing. In: Proceedings of the 4th international symposium on parallel and distributed computing (ISPDC'05); July 2005. p. 217–25.
Keyser JD, Roose D. Run-time load balancing techniques for a parallel unstructured multi-grid Euler solver with adaptive grid refinement. Parallel Comput 1995;21:179–98.
Khanli LM, Razzaghzadeh S, Zargari SV. A new step toward load balancing based on competency rank and transitional phases in grid networks. Future Gener Comput Syst 2012;28:682–8.
Kim C, Kameda H. An algorithm for optimal static load balancing in distributed computer systems. IEEE Trans Comput 1992;41(3):381–4.
Kim C, Kameda H. Optimal static load balancing of multi-class jobs in a distributed computer system. In: Proceeding of the 10th international conference on distributed computing systems; May 1990. p. 562–69.
Korkhov VV, Krzhizhanovskaya VV, Sloot PMA. A grid-based virtual reactor: parallel performance and adaptive load balancing. J Parallel Distrib Comput 2008;68:596–608.
Korkhov VV, Moscicki JT, Krzhizhanovskaya VV. The user-level scheduling of divisible load parallel applications with resource selection and adaptive workload balancing on the grid. IEEE Syst J 2009;3(1):121–30.
Korkhov VV, Moscicki JT, Krzhizhanovskaya VV. Dynamic workload balancing of parallel applications with user-level scheduling on the grid. Future Gener Comput Syst 2009;25:28–34.
Lan Z, Taylor VE, Li Y. DistDLB: improving cosmology SAMR simulations on distributed computing systems through hierarchical load balancing. J Parallel Distrib Comput 2006;66:716–31.
Lee JS. Intelligence balancing for communication data management in grid computing. In: Proceedings of the international conference on grid and cooperative computing (GCC'03). Lecture Notes in Computer Science; 2004, vol. 3033. p. 250–53.
Lee SY, Huang J. A Theoretical approach to load balancing of a target task in a temporally and spatially heterogeneous grid computing environment. In: Proceedings of the grid computing workshops. Lecture Notes in Computer Science; 2002, vol. 2536. p. 70–81.
Li K. Optimal load distribution in nondedicated heterogeneous cluster and grid computing environments. J Syst Archit 2008;54:111–23.
Li Y, Yang Y, Ma M, Jhou L. A hybrid load balancing strategy of sequential tasks for grid computing environments. Future Gener Comput Syst 2009;25:819–28.
Li X. A non-dominated sorting particle swarm optimizer for multi-objective optimization. In: Proceeding of the genetic and evolutionary computation conference (GECCO'03). USA; 2003.
Liao WH, Shih KP, Wu WC. A grid-based dynamic load balancing approach for data-centric storage in wireless sensor networks. Comput Electr Eng 2010;36:19–30.
Lu K, Subrata R, Zomaya AY. On the performance-driven load distribution for heterogeneous computational grids. J Comput Syst Sci 2007;73:1191–206.
Lu K, Subrata R, Zomaya AY. On the performance driven load distribution for heterogeneous computational grids. J Comput Syst Sci 2007;73(8):1191–206.
Lu K, Subrata R, Zomaya AY. An efficient load balancing algorithm for heterogeneous grid systems considering desirability of grid sites. In: Proceedings of the 25th international conference on performance, computing, and communications (IPCCC'06); April 2006. p. 311–19.
Lu K, Subrata R, Zomaya AY. Towards decentralized load balancing in a computational grid environment. In: Proceedings of the international conference on advances in grid and pervasive computing (GPC'13). Lecture Notes in Computer Science; 2006, vol. 3947. p. 466–77.
Lu K, Zomaya AY. A hybrid policy for job scheduling and load balancing in heterogeneous computational grids. In: Proceedings of the sixth international symposium on parallel and distributed computing (ISPDC'07); July 2007. p. 1–19.
Ludwig SA, Moallem A. Swarm intelligence approaches for grid load balancing. J Grid Comput 2011:279–301.
Ma YW, Chao HC, Chen JL, Wu CY. Load-balancing mechanism for the RFID middleware applications over grid networking. J Netw Comput Appl 2011;34:811–20.

Ma J. A novel heuristic genetic load balancing algorithm in grid computing. In: Proceedings of the 2nd international conference on intelligent human-machine systems and cybernetics; 2010. p. 166–69.

Maheswaran M, Ali S, Siegel HJ, Hensgen D, Freund R. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing system. J Parallel Distrib Comput 1999;59:107–31.

Maheswaran M, Ali S, Siegel HJ, Hensgen D, Freund R. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. In: Proceedings of the 8th IEEE heterogeneous computing workshop (HCW'99); April 1999. p. 30–44.

Maheswaran M, Ali S, Siegel HJ, Hensgen DA, Freund RF. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. In: Proceedings of the 8th heterogeneous computing workshop; 1999.

Malarvizhi N, Uthariaraj VR. Hierarchical load balancing scheme for computational intensive jobs in grid computing environment. In: Proceedings of the 1st IEEE international conference on advanced computing; 2009. p. 97–04.

Martino VD, Mililotti M. Sub-optimal scheduling in a grid using genetic algorithm. Parallel Comput 2004;30(5/6):553–65.

Mehta HK, Chandwani M, Kanungo P. A modified delay strategy for dynamic load balancing in cluster and grid environment. In: Proceedings of the international conference on information science and applications (ICISA); April 2010. p. 1–8.

Mello RFD, Senger LJ. A routing load balancing policy for grid computing environments. In: Proceedings of the 20th international conference on advanced information networking and applications (AINA'06); April 2006. p. 1–6.

Mello RF, Trevelin LC, Paiva MS, Yang LT. Comparative study of the server-initiated lowest algorithm using a load balancing index based on the process behaviour for heterogeneous environment, In Networks, Software Tools and Application, ISSN 1386-78572004.

Mezmaz M, Melab N, Talbi EG. An efficient load balancing strategy for Gridbased branch and bound algorithm. Parallel Comput 2007;33:302–13.

Mezmaz M, Melab N, Talbi EG. An efficient load balancing strategy for grid-based branch and bound algorithm. Parallel Comput 2007;33:302–13.

Mitchell WF. A refinement-tree based partitioning method for dynamic load balancing with adaptively refined grids. J Parallel Distrib Comput 2007;67:417–29.

Moallem A, Ludwig SA. Using artificial life techniques for distributed grid job scheduling. In: Proceedings of ACM symposium on applied computing. Hawaii, USA; 2009. p. 1091–97.

Moradi M, Dezfuli MA, Safavi MH. A new time optimizing probabilistic load balancing algorithm in grid computing. In: Proceedings of the 2nd international conference on computer engineering and technology; 2010. p. 232–37.

Myer T. Grid computing: conceptual flyover for developers, IBM's developerswork grid library. IBM Corporation; 2003.

Nandagopal M, Uthariaraj RV. Hierarchical status information exchange scheduling and load balancing for computational grid environments. Int J Comput Sci Netw Secur 2010;10(2):177–85.

Nanthiya D, Keerthika P. Hierarchical load balancing GridSim architecture with fault tolerance. Int J Sci Eng Res 2013;4(5):399–403.

Nasir HJA, Mahamud KRK, Din AM. Load balancing using enhanced ant algorithm in grid computing. In: Proceedings of the 2nd IEEE international conference on computational intelligence, modelling and simulation; 2010. p. 160–65.

Nasir HJA, Ruhana K, Mahamud K. Grid load balancing using ant colony optimization. In: Proceedings of the second international conference on computer and network technology; 2010: p. 207–11.

Ni LM, Hwang K. Optimal load balancing in a multiple processor system with many job classes. IEEE Trans Softw Eng 1985;11(10):1141–52.

Nikkhah M, Safaeipour R, Moradi M. Investigating of probabilistic load balancing algorithms in grid computing. In: Proceedings of the 2nd international conference on education technology and computer (ICETC); 2010. p. 461–66.

Othman M, Abdullah M, Ibrahim H, Subramaniam S. A^2DLT: divisible load balancing model for scheduling communication-intensive grid applications. In: Proceedings of the international conference on computational science (ICCS'08). Lecture Notes in Computer Science; 2008, vol. 3251. p. 246–53.

Othman M, Abdullah M, Ibrahim H, Subramaniam S. Adaptive divisible load model for scheduling data-intensive grid applications. In: Proceedings of the international conference on computational science (ICCS'07). Lecture Notes in Computer Science; 2007, vol. 4487. p. 446–53.

Pan YL, Lee YC, Wu F. Job scheduling of savant for grid computing on RFID EPC network. In: Proceedings of the IEEE international conference on services computing; July 2005, vol. 2. p. 75–82.

Paranhos D, Cirne W, Brasileiro F. Trading cycles for information: using replication to schedule bag-of-tasks applications on computational Grids. In: Proceedings of international conference on parallel and distributed computing (EuroPar). Lecture Notes in Computer Science; 2003, vol. 2790. p. 169–80.

Park JG, Chae HS, So ES. A dynamic load balancing approach based on the standard RFID middleware architecture. In: Proceedings of the IEEE international conference on e-business engineering; October 2007. p. 337–40.

Park SM, Song JH, Choi WY, Kim CS, Lee SW, Kim JJ. Kim RFID middleware system supporting priority service. In: Proceedings of the ninth international conference on advanced communication technology; 2007, vol. 1. p. 427–31.

Park SM, Song JH, Kim CS, Kim JJ. Load balancing method using connection pool in RFID middleware. In: Proceedings of the fifth ACIS international conference on software engineering research, management & applications; August 2007. p. 132–37.

Penmatsa S, Chronopoulos AT. Game-theoretic static load balancing for distributed systems. J Parallel Distrib Comput 2011;71:537–55.

Penmatsa S, Chronopoulos AT. Job allocation schemes in computational Grids based on cost optimization. In: Proceedings of the 19th IEEE international parallel and distributed processing symposium, Denver; 2005.

Prakash S, Vidyarthi DP. Load balancing in computational grid using genetic algorithm. Adv Comput 2011;1(1):8–17.

Qureshi K, Rehman A, Manuel P. Enhanced GridSim architecture with load balancing. J Supercomput 2010:1–11.

Rahmeh OA, Johnson P. A load balancing scheme for latency optimization in grid networks. In: Proceedings of the 5th international conference on digital information management (ICDIM); July 2010. p. 347–52.

Rajavel R. De-centralized load balancing for the computational grid environment. In: Proceedings of the international conference on communication and computational intelligence; December 2010. p.419–24.

Rathore N, Channa I. Load balancing and job migration techniques in grid: a survey of recent trends. Wirel Pers Commun 2014:1–37.

Rathore N, Channa I. Variable threshold-based hierarchical load balancing technique in Grid. Eng Comput 2014:1–19.

Rathore N, Channa I. A cogitative analysis of load balancing technique with job migration in grid environment. In: IEEE Proceedings of the World congress on information and communication technology (WICT); December 2011. p. 77–82.

Rathore NK, Chana I. A sender initiate based hierarchical load balancing technique for grid using variable threshold value. In: Proceedings of the IEEE international conference on signal processing, computing and control (ISPCC); September 2013. p. 1–6.

Ratnasamy S, Karp B, Yin L, Yu F, Govindan R, Shenker S. GHT: a geographic hash table for data-centric storage. In: Proceedings of the first ACM international workshop on wireless sensor networks and applications; 2002.

Reddy KHK, Roy DS. A hierarchical load balancing algorithm for efficient job scheduling in a computational grid testbed. In: Proceedings of the 1st international conference on recent advances in information technology (RAIT); March 2012. p. 363–68.

Rings T, Caryer G, Gallop J, Grabowski J, Kovacikova T, Schulz S, Stokes-Rees I. Grid and cloud computing: opportunities for integration with the next generation network. J Grid Comput 2009;7(3):375–93.

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

Rzadca K, Trystram D. Promoting cooperation in selfish computational grids. Eur J Oper Res 2009;199:647–57.

Saha D, Menasce D, Porto S. Static and dynamic processor scheduling disciplines in heterogeneous parallel architectures. J Parallel Distrib Comput 1995;28(1):1–18.

Sakellariou R. On the quest for perfect load balance in loop-based parallel computations (Ph.D. thesis). Department of Computer Science, University of Manchester; 1996.

Salehi MA, Deldari H, Dorri BM. MLBLM: A multi-level load balancing mechanism in agent-based grid. In: Proceedings of the 8th international conference on distributed computing and networking; 2006. p. 157–62.

Salehi MA, Deldari H. A novel load balancing method in an agent-based grid. In: Proceedings of the international conference on computing & informatics; June 2006. p. 1–6.

Salimi R, Motameni H, Omranpour H. Task scheduling using NSGA II with fuzzy adaptive operators for computational grids. J Parallel Distrib Comput 2014;74:2333–50.

Salimi R, Motameni H, Omranpour H. Task scheduling with load balancing for computational grid using NSGA II with fuzzy mutation. In: Proceedings of the 2nd IEEE international conference on parallel, distributed and grid computing; 2012. p. 79–84.

Sanders P. Analysis of nearest neighbour load balancing algorithms for random loads. Parallel Computing 1999;25:80.

Senger LJ, de Mello RF, Santana MJ, Santana RHC. Santana Improving scheduling decisions by using knowledge about parallel applications resource usage; 2005. p. 1–10.

Shah R, Veeravalli B, Misra M. On the design of adaptive and decentralized load-balancing algorithms with load estimation for computational grid environments. IEEE Trans Parallel Distrib Syst 2007;18:1675–87.

Shivaratri NG, Krueger P, Singhal M. Load distributing for locally distributed systems. Computer 1992;25(12):33–44.

Singh A, Awasthi LK. Performance comparisons and scheduling of load balancing strategy in grid computing. In: Proceedings of the international conference on emerging trends in networks and computer communications (ETNCC); April 2011. p. 438–43.

Subrata R, Zomaya AY, Landfeldt B. Artificial life techniques for load balancing in computational grids. J Comput Syst Sci 2007;73:1176–90.

Subrata R, Zomaya AY, Landfeldt B. Game-theoretic approach for load balancing in computational grids. IEEE Trans Parallel Distrib Syst 2008;19(1):66–76.

Suresh P, Balasubramanie P. User demand aware grid scheduling model with hierarchical load balancing. Math Prob Eng 2013, . http://dx.doi.org/10.1155/2013/439362 Art No. 439362.

Suri PK, Singh M. An efficient decentralized load balancing algorithm for grid. In: Proceedings of the IEEE international conference on advance computing conference (IACC); February 2010. p. 10–13.

Viswanathan S. Resource-aware distributed scheduling strategis for large-scale computational cluster/grid systems. IEEE Trans Parallel Distrib Syst 2007;18(10):1450–60.

Wang J, Wang Y. A robust load balancing pattern for grid computing. In: Proceedings of the first international conference on semantics, knowledge, and grid (SKG'05); November 2005. p. 1–3.
Wang J, Wu QY, Jheng D, Jia Y. Agent based load balancing model for service based grid applications. In: Proceedings of the international conference on computational intelligence and security; November 2006. p. 486–91.
Wolski R, Spring NT, Hayes J. The network weather service: a distributed resource performance forecasting service for metacomputing. Future Gener Comput Syst 1999;15(5):757–68.
Wong HM, Veeravalli B, Dantong Y, Robertazzi TG. Data intensive grid scheduling: multiple sources with capacity constraints. In: Proceeding of the IASTED conference on parallel and distributed computing and systems. Marina del Rey, USA; 2003.
Wu J, Xu X, Zhang P, Liu C. A novel multi-agent reinforcement learning approach for job scheduling in Grid computing. Future Gener Comput Syst 2011;27:430–9.
Xu C, Lau F, Monien B, Luling R. Nearest neighbour algorithms for load balancing in parallel computers. Concurr Pract Exp 1995.
Yagoubi B, Slimani Y. Dynamic load balancing strategy for grid computing. World Acad Sci Eng Technol 2006:90–5.
Yagoubi B, Slimani Y. Task Load balancing strategy in grid environment. J Comput Sci 2007;3:186–94.
Yagoubi B, Slimani Y. Load balancing strategy in grid environment. J Inf Technol Appl 2007;1(4):285–96.
Yagoubi B, Lilia HT, Maussa HS. Load balancing in grid computing. Asian J Inf Technol 2006;5(10):1095–103.
Yan KQ, Wang SC, Chang CP, Lin JS. A hybrid load balancing policy underlying grid computing environment. Comput Stand Interfaces 2007;29:161–73.
Yan KQ, Wang SS, Wang SC, Chang CP. Towards a hybrid load balancing policy in grid computing system. Expert Syst Appl 2009;36:12054–64.
Yang D. A parallel grid modification and domain decomposition algorithm for local phenomena capturing and load balancing. J Sci Comput 1997;12(1):99–117.
Yang K, Guo X, Galis A, Yang B, Liu D. Towards efficient resource on demand in Grid computing. Oper Syst Rev 2003;37(2):37–43.
Zheng Q, Tham CK, Veeravalli B. Dynamic load balancing and pricing in grid computing with communication delay. J Grid Comput 2008;6:239–53.
Zhou S, Ferrari D. An experimental study of load balancing performance (Technical Report UCB/CSD 87/336). PROGRES Report No. 86.8. Berkeley (California): Computer Science Division (EECS), Universidade da California; January 1987. 94720.
Zhu X, Qin X, Qiu M. Qos-aware fault-tolerant scheduling for real-time tasks on heterogeneous clusters. IEEE Trans Comput 2011;60(6):800–13.
Zhu W, Sun C, Shieh C. Comparing the performance differences between centralized load balancing methods. In: Proceedings of IEEE international conference on systems, man, and cybernetics;1996, 3. p. 1830–35.
Zikos S, Karatza HD. Communication cost effective scheduling policies of nonclairvoyant jobs with load balancing in a grid. J Syst Softw 2009;82:2103–16.
Zikos S, Karatza HD. Resource allocation strategies in a 2-level hierarchical grid system. In: Proceedings of the 41st annual simulation symposium (ANSS). IEEE Computer Society Press, SCS; April 2008. p. 157–64.
Zoghdy SFE, Aljahdali S. A two-level load balancing policy for grid computing. In: Proceedings of the international conference on multimedia computing and systems (ICMCS); May 2012. p. 617–22.
Zomaya AY, Teh YH. Observations on using genetic algorithms for dynamic load-balancing. IEEE Trans Parallel Distrib Syst 2001;12(9):899–912.