A Survey of Task Allocation and Load Balancing in Distributed Systems

Yichuan Jiang, *Senior Member, IEEE*

Abstract: In past decades, significant attention has been devoted to the task allocation and load balancing in distributed systems. Although there have been some related surveys about this subject, each of which only made a very preliminary review on the state of art of one single type of distributed systems. To correlate the studies in varying types of distributed systems and make a comprehensive taxonomy on them, this survey mainly categorizes and reviews the representative studies on task allocation and load balancing according to the general characteristics of varying distributed systems. First, this survey summarizes the general characteristics of distributed systems. Based on these general characteristics, this survey reviews the studies on task allocation and load balancing with respect to the following aspects: 1) typical control models; 2) typical resource optimization methods; 3) typical methods for achieving reliability; 4) typical coordination mechanisms among heterogeneous nodes; and 5) typical models considering network structures. For each aspect, we summarize the existing studies and discuss the future research directions. Through the survey, the related studies in this area can be well understood based on how they can satisfy the general characteristics of distributed systems.

Keywords: Distributed systems; task allocation; load balancing; networks; resource allocation; survey; taxonomy.

1 Introduction

Distributed systems, in which the distributed computational units are connected and organized by networks to meet the demand of large-scale and high performance computing, have received considerable attention over the past decades [1][2][3]. There are many types of distributed systems, such as grids [4], peer-to-peer (P2P) systems [5], ad hoc networks [6], cloud computing systems [7], pervasive computing systems [8], and online social network systems [9].

Currently, the applications in distributed systems are various, such as web service,

systems [12]. In past decades there are significant amount of studies for this area, which could cause people to be puzzled by the enormous amount of wide variety of research results.

Although there have been some related surveys previously, but most of which only made a very preliminary review on the state of art of one single type of distributed system, such as the survey of load balancing in grids [91][92], the survey of load balancing in cloud computing [93][94], and the survey of load balancing in P2P systems

[95]. Then, how to correlate the rela
types of distributed systems and mal
on them?

To solve the above problem, fi
summarizes the typical characteristic
ed systems, such as the distributed c
bution, open environments, heter
network structures. Then, the survey
existing studies on task allocation an
on how they can satisfy the charac
systems, which includes the follo
models, resource optimization, ensu
nation mechanisms, and measures
structures. With this survey, a gen
review of task allocation and load
types of distributed systems can be
have higher generality much than th
surveys for only one single type of sy

The remainder of this survey is o
Section 2, we discuss the problems
allocation and load balancing in d
Section 3, we review the control mo
and load balancing; in Section 4, w
optimization methods in task allocati
in Section 5, we review the reliability
load balancing; in Section 6, we re
among heterogeneous nodes for tas
balancing; in Section 7, we review th
ering network structures; and finally
vey in Section 8.

2 The Problems and Objectives of Task Allocation and Load Balancing in Distributed Systems

2.1 Problems

2.1.1 Overview on Distributed Systems

- *The author is with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China, and also with the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China. E-mail: yjiang@seu.edu.cn.*
- *Manuscript received 30 Oct.2014*

Currently, a large number and variety of applications have been extensively developed for distributed systems linking computers and other computing devices together in a seamless and transparent way. In distributed systems, various nodes act autonomously and cooperate with each other, which can achieve the purposes of resource sharing, openness, concurrency, scalability, fault-tolerance, and transparency [16][17]. In summary, real distributed systems have the following general characteristics.

- *Lack of a global control unit*. Due to the large-scale, dynamic, and heterogeneous nature of distributed system [17], it is difficult to implement a global control unit for the entire system. For example, in ad hoc networks, wireless nodes can communicate with each other in the absence of a fixed infrastructure and without central control [18]; in sensor networks, many sensor nodes are densely deployed and may not have global identifications because of the large amount of overhead and large number of sensors [19]; in P2P systems, a decentralized architecture is applied in which individual nodes in the network act as both suppliers and users of resources, in contrast to the centralized client-server model where client nodes request access to resources provided by central servers [5].
- *Distribution and sharing of resources*. There are various resources in the systems, such as data, replicas of popular web objects, processors, computational capacity, and disk storage. Resources are distributed among nodes, and nodes often agree to share their local resources with each other to implement tasks [20]. To implement the effective sharing of resources, resource management is very crucial, which often includes resource placement, resource provision, resource discovery, and resource negotiation [5][21][22]. Moreover, resource caching is often used to improve the performance of resource access, where some nodes cache the resources to serve future requests instead of fetching resources from the original nodes [23].
- *Openness and unreliability of systems*. Distributed systems are often open, and some nodes and network structures may be unreliable. For example, there are no efficient ways to prevent malicious peers from joining open systems, thus the distributed systems are very vulnerable to abuses by selfish and malicious users [24]. Moreover, due to the autonomy of nodes, some nodes may fail to operate independently. Therefore, reliability and fault-tolerance are crucial for real open distributed systems.
- *Heterogeneous nodes*. In some distributed systems, the nodes are heterogeneous. For example, in grids, the nodes may be attributed to different organizations [20]; in sensor networks, the nodes may have different capacities and sensing ranges [19][25]; in social network systems, the nodes may have different response times and thresholds for diffusion [26]. When tasks are executed, heterogeneous nodes should coordinate with each other to maximize the total utilities. Moreover, heterogeneous nodes may compete for critical resources when they execute tasks.
- *Constraints of network structures*. In a distributed system, all nodes are constrained by the network structure. In the network structure, each node can only communicate with its neighboring nodes; the communication between two non-neighboring nodes should be relayed by other intermediate nodes [1][11]. Therefore, the system performance is significantly influenced by the network structures.

Accordingly, the tasks in distributed systems always encounter the following situations:

- *The tasks may be implemented by negotiation among nodes without a central control*. In a distributed system, the tasks that require more than one node may be implemented in a manner such that many nodes coordinate with each other and negotiate resources. Because distributed systems often lack a global control unit, as stated above, the negotiation in the task allocation may be implemented by nodes autonomously.
- *The tasks are always implemented by accessing required resources distributed in the networks*. To execute tasks efficiently, one of the key elements is to improve the performance of accessing the resources necessary for the tasks. Therefore, resource optimization in networks can significantly influence the performance of tasks.
- *The tasks may be implemented unreliably*. Distributed systems are open and may be invaded by malicious users [9]; moreover, there may be unreliable resources and communications in distributed systems [16]. Therefore, the tasks may be implemented unreliably, and sometimes the desired results may not be achieved [27].
- *The tasks performed by heterogeneous nodes may vary significantly and compete for critical resources*. Because the nodes are heterogeneous and the number of users may vary dynamically in real time, the tasks may vary significantly at different times and at different nodes. Moreover, the users at different nodes are often self-motivated and will attempt to maximize their own benefits. Therefore, the execution of tasks of different users may compete for critical resources, which may bring about conflicts in the systems.
- *The performance of tasks may be influenced significantly by network structures*. First, the network structures may influence the resource access performance for tasks; thus, the resource optimization in task allocation should consider the network structures. Second, the interaction and coordination among nodes often obey certain network structures, thus a suitable network structure can improve the coordination performance of nodes in task allocation.

2.1.2 Task Allocation and Load Balancing in Distributed Systems

Without loss of generality, the tasks in distributed systems are always implemented by accessing required resources in the networks [1][28][29]. For example, a file storage task will seek the required memory resources in the network. Therefore, many existing models of task allocation and load balancing have been implemented based on the accessibility of required resources [11][27][28][29], i.e., a node's probability of being allocated tasks is determined by its accessibility to required resources for the tasks. We now provide the formal definition of task allocation in distributed systems, which is implemented based on the resource requirements of tasks.

Definition 1 Task Allocation in Distributed Systems [27]. *Given a distributed system, $N=<A, E>$, where A is the set of nodes and each node owns a different set of resources, and $\forall <a_i,a_j> \in E$ indicates the existence of a network link between node a_i and a_j, the set of resources in node a_i is assumed to be R_{ai}, and the set of resources required by task t is assumed to be R_t. If task t arrives at the network, the task allocation in N can be defined as the mapping of task t to a set of nodes, A_t, which can*

satisfy the following situations:
1) The resource requirements of t can be satisfied, i.e., $R_t \subseteq \ _{\forall a_i \in A_t} R_{a_i}$ *;*
2) The predefined objective can be achieved by the task execution of A_t, such as minimizing the execution time [85] or maximizing the reliability [86].
3) The nodes in A_t can execute the allocated tasks under the constraint of the network structure, e.g. $\forall a_i, a_j \in A_t \Rightarrow P_{ij}$ E, where P_{ij} denotes the interaction path between a_i and a_j.

According to Definition 1, if a node has a higher probability of accessing the necessary resources for tasks, it may be allocated more tasks. However, if too many tasks are crowded on certain nodes with high probabilities of accessing the tasks' required resources, the tasks will require significantly more time to wait for the necessary resources [1][4][11]. Therefore, we should now apply load balancing to the task allocation.

Definition 2 Load Balancing in Task Allocation [27]. G*iven a distributed systems, N=<A, E>, where A is the set of nodes, $\forall a_i \in A$, the team of tasks that queue for resource r_k of node a_i can be denoted as Q_{ik}. The size of Q_{ik} is s_{ik} and the processing capability of a_i is v_i; the original probability of node a_i to receive tasks (which need k type resources) is $P_i(k)$, which can be calculated by considering the resources of a_i. We should perform load balancing when s_{ik} is too large, which is implemented by discounting the probability of a_i's receiving new tasks, $P_i(k)$, to a revised probability, $DP_i(k)$:*

$$DP_i(k) = \psi(s_{ik} / v_i) \cdot P_i(k) \qquad (1)$$

where ψ is an attenuation function, $0 \le \psi \le 1$; the value of $\psi(s_{ik}/v_i)$ decreases monotonically from 1 to 0 as s_{ik}/v_i increases.

2.1.3 Typical Problems

Because of the typical characteristics of distributed systems stated in Section 2.1.1, some typical problems occur during the task allocation and load balancing, shown as follows.

- *Control models*. To achieve the globally optimal result for task allocation and load balancing, a centralized control model is required to collect the status information of the entire system in real time; however, it is difficult to implement such totally centralized control model because distributed systems are always large, dynamic, and lacking global control units. By contrast, the totally decentralized approach may require relatively high computational costs from the nodes, which may place heavy loads on large systems and make the task allocation processes difficult to control effectively [27].
- *Resource optimization*. Many existing models of task allocation and load balancing have been implemented based on the optimization of accessibility of required resources [11][27][28][29]. Then, how to measure the resource accessibility and how to optimize such a performance index are key problems. For example, the following two typical situations are always observed in reality: 1) if a node has plentiful resources, it may have high resource accessibility; 2) if a node does not have plentiful resources by itself, but it can obtain enough resources from other interacting nodes easily, then it can also be allocated many tasks. Therefore, the optimization for resource accessibility should not only consider the access performance of nodes' own resources but also that of nodes' contextual resources [11].
- *Reliability.* In open distributed systems, some nodes may be unreliable. Therefore, a key problem is how to guarantee reliable resource access as well as optimize the resource access performance [27]. Therefore, we should design some reliability-oriented approaches to find a task allocation result such that the reliability is maximized [30][31].
- *Coordination among heterogeneous nodes.* Heterogeneous nodes do not cooperate in making decisions while they execute tasks [32]; each node may optimize its own resource access performance independently of the others. Therefore, how to make these noncooperative nodes coordinate to improve the overall performance of tasks should be well-solved; moreover, how to incentivize self-interested nodes to contribute their resources to others is also a key problem [28].
- *Considering network structures.* The communication and interaction of nodes are constrained by the network structures. Then, how to measure the influence of network structures on the performance of tasks is a problem. The task allocation and load balancing should consider the localities c-

eTc.0 elap8(plb Tfeenanci)3(m fr)4s-2.1(ms oc0039 Tc())Tj-26]TJ13.8
p1.6(e t)-allforman.2152(itCs6 cs 19.300scn.7o)6.

scheme in heterogeneous distributed systems, and they derived a new distributed load balancing algorithm based on the structure of the Nash equilibrium; such approach can achieve the advantage of distributed structure, low complexity, and optimality of allocation for each user. Penmatsa and Chronopoulos [44] formulated the load balancing problem as a non-cooperative game among the users who try to minimize the expected response time of their own tasks, which can provide fairness to all the users in the system.

Auction is another used market-like technique for the task allocation and load balancing in distributed systems that describes the mechanism or set of negotiation rules for tasks. For example, Izakian et al. [45] introduced a continuous double auction method for grid resource allocation in which resources are considered as provider agents and users as consumer agents, which is efficient in terms of successful execution rates, resource utilization rates, and fair profit allocation. Choi et al. [46] presented the consensus-based auction algorithm to perform decentralized task allocation, which utilizes a market-based decision strategy as the mechanism. Garg et al. [47] proposed a novel meta-scheduler based on the double auction mechanism.

3.2.2 Society -Based Approaches

There are many societal mechanisms, such as strategy diffusion and coalition formation, which describe the collective decision making of actors to coexist.

Strategy diffusion is a common phenomenon in the collective motion of society, which denotes the large scale of strategy penetrations of certain agents on other agents [50]. In distributed systems, the negotiation between neighboring nodes for tasks can be described as a diffusion process; for example, Rotaru and Nägeli [48] introduced the diffusion mechanism into the dynamic load balancing in heterogeneous systems and gave a direct explicit expression of the balancing flow generated by a generalized diffusion algorithm. The advantage of the diffusion mechanism is that each node only knows the information of neighbor nodes and the global optimal result can be achieved finally.

A coalition is an alliance among individuals, in which individuals cooperate in joint action. In distributed systems, some nodes can form a coalition to execute a task if such a task cannot be performed by a single node [90]. For example, Shehory and Kraus [49] presented algorithms that enable the nodes to form coalitions and assign a task to each coalition, and these algorithms have low ratio bounds and low computational complexities. Low et al. [51] presented a task allocation scheme via self-organizing swarm coalitions for distributed mobile sensor network coverage, which uses the concepts of ant behavior to self-regulate the regional distributions of sensors in proportion to that of the moving targets to be tracked. Wang and Jiang [52] proposed a community-aware task allocation model for social network systems where a node's cooperation domain is constrained in a community and each node can negotiate only with its intracommunity member nodes; the proposed model performs very closely to the benchmark exponential brute-force optimal and the network flow-based greedy algorithm in terms of overall system profit.

Moreover, there are a few studies that utilize other typical social mechanisms for task allocation. For example, Jiang and Huang [1] used the Matthew effect (i.e., the rich get richer mechanism) for the task allocation of distributed systems with resource caching, where the nodes that were heavily burdened by tasks may have certain preferential rights to new tasks in the future. Through experimental validation, it was proven that such Matthew effect-based task allocation can effectively reduce tasks' execution time.

In summary, the aforementioned studies are still in the preliminary stage of purely applying some special societal mechanisms, and the final performance often depends on the matching degree between the societal mechanisms and the system environments. In the future, we think the learning and self-adaption in society should be introduced to implement a real self-adaptable task allocation for dynamic and complex systems. Moreover, a systematic methodology for applying societal mechanisms should be developed.

3.2.3 Agent-Based Approaches

In the agent-based approaches, the agents are distributed in the network and span across the nodes; the agents can interact with nodes and cooperate with other agents in order to solve complex problems for tasks [53]. Totally, autonomous agents can observe their surrounding environment and perform simple local computations leading to actions of task allocation and load balancing.

Liu et al. [4] provided an agent-based decentralized task allocation mechanism for minigrids. In the proposed model, an agent will be automatically dispatched to a task immediately after the task is randomly submitted to a node; then, the agent will carry the task to search for an appropriate node and the dispersion of agents can realize load balancing. To reduce the waiting time of the task, the agent will take the task to a node where the queue of waiting agents is small. In the model, agents' behaviors can be classified into two types, leaving and queuing; an agent carrying a task will leave a node if it finds there are too many agents queuing at such node; an agent carrying a task will decide to queue at a node if it finds the number of already queuing agents at such node acceptable.

Chow and Kwok [54] proposed a multiagent-based load balancing model for a cluster of workstations, where the agents can span across all the machines of a cluster. The proposed algorithm works by associating each agent with a credit value that depends on the agent's affinity to a machine, the agent's current workload, the agent's communication behavior, the agent's mobility, etc. To realize load balancing among workstations, the credits of agents are examined, and the agent with a lower credit value is migrated to a relatively lightly loaded workstation.

3.2.4 Other Alternative Approaches for Various Specific Objectives

There are some distributed control approaches that cannot be categorized into the above three types; now, we also introduce some representative studies of them here.

Di and Wang [35] proposed a novel autonomous resource allocation scheme to maximize the throughput of self-organizing P2P grid systems, which have high adaptability to dynamic environments by proactive and convex-optimal estimation of nodes' volatile states. Hong and Prasanna [55] considered the task allocation problem for computing large sets of equal-sized independent tasks on heterogeneous computing systems, and developed a decentralized adaptive algorithm for adaptive task allocation. Beaumont et al. [56] presented a distributed task scheduler to maximize the throughput of each application while ensuring a fair sharing of resources between applications.

Antonis et al. [57] proposed an adaptive distributed hierarchical scheme, the Virtual Tree Algorithm (VTA), which creates a virtual binary tree structure over the actual network topology. Shah et al. [58] proposed adaptive and decentralized load balancing algorithms to address several issues that are imperative to grid environments such as handling resource heterogeneity and sharing, and communication latency.

3.2.5 Summary for Distributed Control Model

In summary, the main characteristic of distributed control model is that the nodes can coordinate with each other autonomously for task allocation. With the autonomous and distributed task allocation, the control can be adapted for varying system scales and the failure of some nodes, thus the robustness is high. However, as each node can only know the information about neighbors, the results may be only locally optimal but not globally optimal. Moreover, the negotiation among nodes may produce additional computation costs.

3.3 Hybrid Control Model

It is generally recognized that the distributed control model is implicitly designed for infinitely large distributed systems, while small systems are regarded as being controllable using the traditional centralized model. However, some real systems do not fit conveniently into these “small” or “large” categories [39]; particularly, some systems may be dynamic, and the scales especially may be variable during operation. Therefore, it is now difficult to decide whether centralized or distributed control models are used. Moreover, totally centralized models and totally decentralized models both have their respective drawbacks.

To solve the above problems, we can consider integrating the centralized and distributed control models in a system. For example, Zaki et al. [59] presented four strategies based on control models (either centralized or distributed) and the required information (either global or local): global centralized load balancing, global distributed load balancing, local centralized load balancing, and local distributed load balancing. They integrated those four strategies and presented a customized load balancing scheme.

Jiang [11] provided a spectrum between a totally centralized approach and a totally decentralized approach for performing task allocation: the centralized heuristic is utilized to control the overall status information, and the distributed heuristic is utilized to achieve the flexibility of task allocation. The approach in [11] is realized as follows: a node is allocated as a manager for a task using a centralized heuristic, and, if the manager lacks the necessary resources, it negotiates with other nodes for contractors using a distributed heuristic; finally, the manager and the contractors constitute the allocated nodes. Then, in [27], Jiang et al. substantially extended the hybrid control architecture in [11] by taking into account the nodes’ negotiation reputations during task allocation, where tasks can obtain dependable resources in the least access time.

Moreover, Abdallah and Lesser [60] proposed a mediator-agent architecture for hybrid control of task allocation,

4 RESOURCE OPTIMIZATION IN TASK ALLOCATION AND LOAD BALANCING

In distributed systems, some resources are placed at the nodes within the networks and can be accessed and shared by users to execute tasks [14][32]; typical resources are computation resources, communication resources, data resources, and storage resources. Without loss of generality, task execution can be described through nodes' operations when accessing necessary resources distributed in the networks [1][5][6][27][28]; also, task allocation often means allocating tasks to resources [34][35].

To achieve good performance of tasks, many existing studies have been implemented based on resource optimization [11][27][28][29]. The related works of resource optimization mainly considered two types of resources: 1) *a self-owned resource-based approach* implemented based on nodes' self-owned resource statuses, i.e. a node's probability of being allocated tasks is determined only by its self-owned resources, and the performance of tasks is mainly related to the allocated nodes' own resource statues; and 2) *a contextual resource-based approach* implemented based on not only nodes' self-owned resource statuses but also their contextual resource statuses because nodes may cooperate with others within their contexts, and the performance of tasks are related not only to the allocated nodes' resource statuses but also their contextual resource statuses.

In existing related studies, there are many optimization objectives, such as minimizing the resource access time, maximizing the resource access reliability, and minimizing the resource access conflicts of different tasks.

4.1 Self-Owned Resource-Based Optimization

In some systems, the resources are often dynamic and of great heterogeneity; moreover, the resources may become unreliable due to disconnection of power or communication. To optimize the resource allocation for tasks in grids, Xu et al. [34] presented a Chemical Reaction Optimization algorithm inspired by the interactions between molecules in a chemical reaction, which can optimize the resources for tasks with three objectives: makespan, flowtime, and reliability. Letting *Makespan* be the completion time of the last finished task, *Flowtime* be the total time consumed by all tasks, and *T-aborted* represent the total wasted time caused by the aborted tasks in terms of reliability, the objective of resource optimization for tasks in [34] is to:

al resources in the systems with resource caching: if a node's contextual nodes have richer experiences of executing tasks, the node may have higher access to the resources (required by the tasks) even if it seldom accessed the resources by itself. Therefore, Jiang and Huang [1] proposed the following novel idea of task allocation based on contextual preferential attachment: contextually-experienced nodes receive more new tasks than less contextually-experienced nodes. With such model, the contextual resource access time can be optimized.

Moreover, in real-world situations, multiplex networks are often observed where there are multiple types of links between nodes, and each type of link may have a different relative bias in communicating different types of resources. Therefore, in the systems, there may be several network layers, where each network layer is composed of the same type of links and the involved nodes. By considering this new situation, Jiang et al. [63] proposed the network layer-oriented task allocation model, which considers not only the contextual nodes but also the contextual layers.

In summary, the contextual resource-based optimization methods consider the coordination among nodes, so they can be fitted for current cooperative distributed systems where many nodes often cooperate for executing tasks. However, these optimization methods need a large amount of computation, which may produce high costs for the large-scale systems.

4.3 Summary and Future Research Directions

In summary, the comparisons among the two resource optimization methods can be shown as Table 2. From the table, we can see that each method has its own advantages and disadvantages.

Next, we discuss some challenges in existing studies and present some insights on future research directions on the resource optimization of task allocation and load balancing in distributed systems, which are shown as follows.

- There are often many tasks that are executed concurrently in systems, and the concurrent tasks may compete for resources. Existing related studies mainly perform resource optimization according to the current resource situation in the system and cannot predict the future situation where other tasks may also consume resources. Therefore, how to make resource optimization by considering and predicting the resource usages of other concurrent tasks is a research challenge.
- The resource distribution in systems may be dynamic; more seriously, the resource optimization schemes may be disabled due to the drastic change of resource distribution. Therefore, in the future, the resource optimization schemes should adapt to the dynamic change of resource distribution.
- In some distributed systems, such as cloud computing systems, virtualization technology is often used to provide virtualized resources for higher-level applications. Virtualization technology may create challenges for resource optimization because the virtualization provides the capability of pooling computing resources from clusters of servers and dynamically assigning or reassigning virtual resources to applications on-demand [88]. Such issue should be investigated in the future research of resource optimization.

TABLE 2. The comparisons between the two resource optimization methods

Methods	Basic principle	Advantages	Disadvantages	Application Environments
Self-owned resource-based optimization	A node's probability of being allocated tasks is determined only by its self-owned resources; thus only the allocated nodes' own resources are optimized.	Easy to implement and the information for self-owned resources can be easily achieved	The resource access performance may be poor if the allocated nodes cannot finish the tasks by themselves	Nodes often execute the tasks autonomously
Contextual resource-based optimization	A node's probability of being allocated tasks is determined by not only its own resources but also the resources in its contexts; thus both the allocated nodes' own resources and their contextual resources are optimized.	Nodes contribute their resources for cooperating to execute tasks; the resource access time among allocated nodes can be optimized	A large amount of computations are needed and high costs may be produced	Cooperative distributed systems in which many nodes often cooperate for executing tasks

5 Reliability in Task Allocation and Load Balancing

Due to the openness, dynamics, and heterogeneity of distributed systems, the nodes or networks may be unreliable; even more, some users may use the systems to perform malicious actions. Reliability in task allocation and load balancing is of critical importance for distributed systems, especially for some mission critical applications, such as financial systems and military ad hoc networks.

The reliability of tasks is the probability that all tasks can be executed successfully in the system. The reliability-oriented task allocation problem is to find a task allocation result such that the reliability is maximized [30][31]. Typically, there are two classes of approaches: one is the redundancy-based approaches, which mainly implement redundancy of some elements so that the overall task execution is successful even if some parts of the elements fail; the other is the non-redundancy based approaches, which mainly adopt certain mechanisms to optimize the reliability for tasks.

5.1 Redundancy-Based Approaches

There are two types of redundancy to achieve the reliability for executing tasks: one is the redundancy of resources, and the other is the redundancy of tasks.

5.1.1 Redundancy of Resources

Redundancy of resources means that more resources will be provided for tasks than necessary, which can avoid the situation where the tasks may be executed unsuccessfully due to the unreliability of some resources. Generally, it is demonstrated that redundancy in resources can improve reliability. However, the reliability-oriented task allocation based on resource redundancy is NP-hard, so it is necessary to find some heuristic algorithms that can achieve near-optimal results efficiently and simply [31].

Moreover, redundancy may be an expensive approach and incurs higher related costs, thus some researchers aim to maximize the reliability of task allocation with less extra hardware and/or software costs. For example, Kang et al. [13] proposed a simple and effective iterative greedy algorithm to find the best possible solution within a reasonable amount of computation time; the algorithm first uses a constructive heuristic to obtain an initial allocation and iteratively improves it in a greedy way. Hsieh [64] proposed a hybrid genetic algorithm to determine the optimal resource redundancy policies so that the system cost is minimized. Elegbede et al. [65] proposed an algo-

[9]; for example, some nodes may fabricate their resource status information during the task allocation for achieving the tasks, but they do not really contribute all their free resources if the allocated tasks require those resources [27]. To measure the reliability of nodes for tasks, the concepts of trust and reputation are often used.

1) Trust-based approaches

Trust denotes that one party is willing to rely the actions on another part. Given the background that some nodes may not successfully complete their allocated tasks, the trust between nodes should be taken into account when allocating tasks. Trust in task allocation often denotes that the allocator is willing to rely on the actions of a node for providing reliable execution of such task.

Ramchurn et al. [75] developed a class of trust-based mechanisms that can take into account multiple subjective measures of the probability of a node succeeding at a given task and produce allocations that maximize social utility and ensure that no node obtains a negative utility. Shen et al. [76] developed a trust model to construct a novel mechanism that motivates sensor agents to limit their greediness or selfishness to make task allocation, and they modeled the sensor allocation optimization problem with trust-in-loop negotiation game and solved it using a subgame perfect equilibrium.

2) Reputation-based approaches

Reputation of a social actor means the opinion about such actor as defined by others [87]. Now, the concept of reputation can be introduced to measure the reliability in distributed systems. A node's reliability is not merely a binary property perceived by another node, but instead is often a statistical one based on such node's prior performance and behavior as perceived by other nodes [78].

To achieve reliable resources with the least access time to execute tasks in undependable social network systems, Jiang et al. [27] presented a task allocation model based on the negotiation reputation mechanism, where a node's past behaviors in the resource negotiation of task execution can influence its probability to be allocated new tasks in the future. In this model, the node that contributes more reliable resources with less access time during task execution is rewarded with a higher negotiation reputation, and may receive preferential allocation of new tasks. The task allocation model based on negotiation reputation in [27] is superior to the traditional resource-based allocation approaches and game theory-based allocation approaches in terms of both the task allocation success rate and task execution time and that it usually performs close to the ideal approach (in which deceptive nodes are fully detected) in terms of task execution time. In addition, the model in [27] considered the reputation of communication paths; the reputation of communication paths between two nodes can be adapted according to the two nodes' past negotiations in task execution; for example, if the two nodes can negotiate for finishing a task through the path, then the reputation of that path will be gained, and vice versa.

Another representative study is that Yahyaoui [79] presented a distributed reputation-based game theoretical model for task allocation: each node submits a cost for achieving a specific task and computes the reputation based cost, and the node with the minimal reputation-based cost will be allocated the task.

5.3 Summary and Future Research Directions

In summary, a comparison of the different approaches for ensuring reliability of task is shown as Table 3. Next, we discuss some challenges in existing studies and present some insights on future research directions on the reliability of task allocation and load balancing in distributed systems, which are shown as follows.

- In a distributed system, there may be more than one mechanism for ensuring reliability. For example, hybrid clouds are composed of private and public clouds, and the private cloud may have a reliability mechanism different from that of the public cloud. Therefore, how to coordinate different reliability mechanisms and ensure their consistency is a future research issue.
- In most existing related studies, the reliability mechanisms are often predesigned offline. However, the systems may be large and dynamic; thus, how to make the system learn to adopt a suitable reliability mechanism online should be solved in the future.
- Existing reliability mechanisms mainly aim to ensure the reliability during the stage of task allocation. However, how can we do if the perfect reliability cannot be ensured during the task allocation stage? We think that, in the future, the reliability should also be achieved during the task execution stage, i.e., a reliable task execution mechanism should be investigated.

TABLE 3. The comparison among the approaches for reliability

Approaches		Basic principle	Advantages	Disadvantages	Application Environments
Redundancy-based		Implement redundancy of some elements so that the overall task execution is successful even if some parts of the elements fail	Good failure-tolerance performance	Produce additional computation and resource costs	Small-scale systems
Non-redundancy-based	Optimization mechanisms	Use optimization mechanisms to find the optimal or near-optimal task allocation strategy	Can achieve a theoretically reliable performance with minimum costs	The optimization mechanism is not general and may be only useful for some special cases	Some special cases under certain constraints
	Trust/reputation mechanisms	Trust or reputation is used to ensure the reliability: the nodes should interact to construct trust relations for task allocation; and a node's reputation is determined by its past experiences for executing tasks.	There are many mature trust/reputation mechanisms that can be used. And this approach can adapt for different system scales	The construction of trust relation between nodes may need costs or infrastructure. The past experiences of a node may be undependable so that incorrect reputation may be achieved.	Relatively general environments

6 Coordination among Heterogeneous Nodes in Task

to cooperate with other nodes. Therefore, both the resources and locality of a node should be taken into account in task allocation; the experiment results in [15] showed that the locality-based task method can reduce the communication costs for a single task more effectively than the resource-based one, but the resource-based method can generally obtain better load balancing performance for parallel tasks than the locality-based one.

The second type of network structures are the virtual interaction networks, which are composed of the interaction relations among nodes; an example of this type is social network systems. Jiang et al. [27] made a series of experiments on the task allocation in some typical social network structures: small world networks, scale-free networks, random networks, regular ring networks, and the Jazz bands networks; they validated the influences of different social networks on the task allocation model.

In summary, some challenges in existing studies and future research directions on the effects of network structures in task allocation and load balancing of distributed systems can been shown as follows.

- In reality, the physical communication networks and virtual interaction networks may be correlated with each other. Therefore, the correlated effects between these two types of network structures in task allocation should be investigated in the future.
- In real distributed systems, especially large systems, there may be hybrid network structures; for example, ad hoc structure and P2P structure may coexist in the physical communication network in a system, or the small world structures and random structures may be hybrid in the virtual interaction network in a system. Therefore, the effects of hybrid network structures on task allocation will be investigated in the future.

8 Conclusions

Task allocation and load balancing have been intensively researched in past decades; a large number of related studies and results have been presented concerning this topic. However, there are many types of distributed systems, which means that the task allocation and load balancing models in different types of systems are also different; therefore, people may often be puzzled by the enormous amount of varying related studies.

To solve the above problem, in this survey, we perform a systematic review of task allocation and load balancing by analyzing the general characteristics of distributed systems. This survey summarizes the following general characteristics of distributed systems: 1) lack of a global control unit; 2) distribution and sharing of resources; 3) openness and unreliability of systems; 4) heterogeneous node*s;* and 5) constraints of network structures. Based on these typical characteristics, this paper reviews the existing studies on task allocation and load balancing with respect to the following aspects: control models, resource optimization, reliability, coordination, and considering network structures. For each aspect, we summarize the existing studies and discuss the future research directions. Through this survey, the related studies on task allocation and load balancing can be understood well by considering how they can satisfy the characteristics of distributed systems.

Currently, distributed systems are developing rapidly, and new distributed computing topics are continually presented. For example, big data and cloud computing may create many new problems, such as the need for high throughput, interoperability between hybrid clouds, security and privacy. Therefore, task allocation and load balancing should consider these new emerging problems in distributed systems. Moreover, current systems may often be dynamic, scalable, and evolutionary; and how to respond quickly to rapid changes in those systems may bring great challenges to task allocation and load balancing.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.61170164, and No. 61472079), the Funds for Distinguished Young Scholars of the Natural Science Foundation of Jiangsu Province (No.BK2012020), and the Program for Distinguished Talents of Six Domains in Jiangsu Province (No.2011-DZ023).

References

[1] Yichuan Jiang, Zhichuan Huang, “The Rich Get Richer: Preferential Attachment in the Task Allocation of Cooperative Networked Multiagent Systems with Resource Caching,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(5): 1040-1052, 2012.

[2] Yichuan Jiang, Jing Hu, Donghui Lin, “Decision Making of Networked Multiagent Systems for Interaction Structures,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(6): 1107-1121, 2011.

[3] Dayong Ye, Minjie Zhang, Danny Sutanto, “Self-Adaptation Based Dynamic Coalition Formation in A Distributed Agent Network: A Mechanism and A Brief Survey,” *IEEE Transactions on Parallel and Distributed Systems*, 24(5): 1042-1051, 2013.

[4] Jiming Liu, Xiaolong Jin, Yuanshi Wang, “Agent-based Load Balancing on Homogeneous Minigrids: Macroscopic Modeling and Characterization,” *IEEE Transactions on Parallel and Distributed Systems*, 16(7): 586-598, 2005.

[5] Weixiong Rao, Lei Chen, Ada Wai-Chee Fu, Guoren Wang, ”Optimal Resource Placement in Structured Peer-to-Peer Networks”, *IEEE Transactions on Parallel and Distributed Systems*, 21(7): 1011-1026, 2010.

[6] Yuan Xue, Baochun Li, Klara Nahrstedt, “Optimal Resource Allocation in Wireless Ad Hoc Networks: A Price-Based Approach,” *IEEE Transactions on Mobile Computing*, 5(4): 347-364, 2006

[7] Lazaros Gkatzikis, Iordanis Koutsopoulos, “Migrate or Not? Exploiting Dynamic Task Migration in Mobile Cloud Computing Systems,” *IEEE Wireless Communications*, 20(3): 24032, 2013.

[8] P.Lukowicz, A.Pentland, A.Ferscha, “From Context Awareness to Socially Aware Computing,” *IEEE Pervasive Computing*, 11(1): 32-41, 2012.

[9] Yichuan Jiang, J. C. Jiang, ‘Understanding Social Networks from a Multiagent Perspective,” *IEEE Transactions on Parallel and Distributed Systems*, 25(10): 2743-2759, 2014.

[10] Gamal Attiya, Yskandar Hamam, “Task Allocation for Maximizing Reliability of Distributed Systems: A Simulated Annealing Approach,” *Journal of Parallel and Distributed Computing*, 66(10): 1259-1266, 2006.

[11] Yichuan Jiang, Jiuchuan Jiang, “Contextual Resource Negotiation-Based Task Allocation and Load Balancing in Complex Software Systems,“ *IEEE Transactions on Parallel and Distributed Systems*, 20(5): 641-653, 2009.

[12] Y.C.Jiang, J.Jiang, “A Multi-Agent Coordination Model for the Variation of Underlying Network Topology,” *Expert Systems with Applications*, 29(2): 372-382, 2005.

[13] Qinma Kang, Hong He, Jun We, “An Effective Iterated Greedy Algorithm for Reliability-Oriented Task Allocation in Distributed Computing Systems,” *Journal of Parallel and Distributed Computing*, 73(8): 1106-1115, 2013.

[14] Bo Hong, Viktor K.Prasanna, “Adaptive Allocation of Independent Tasks to Maximize Throughput,” *IEEE Transactions on Parallel and Distributed Systems*, 18(10): 1420-1435, 2007.

[15] Yichuan Jiang, Zhaofeng Li, “Locality-Sensitive Task Allocation and Load Balancing in Networked Multiagent Systems: Talent Versus Centrality,” *Journal of Parallel and Distributed Computing*, 71(6): 822-836, 2011.

[16] Weijia Jia, Wanlei Zhou, *Distributed Network Systems: From Concepts to Implementations*, Springer, Boston, 2005.

[17] George Coulouris, Jean Dollimore, and Tim Kindberg, *Distributed Systems – Concepts and Design*, Addison-Wesley, second edition, 1994.

[18] S.J.Kim, G.B.Giannakis, “Optimal Resource Allocation for MIMO Ad Hoc Cognitive Radio Networks*,” IEEE Transactions on Information Theory*, 57(5): 3117-3131, 2011.

[19] I.F.Akyildiz, W.Su, Y.Sankarasubramaniam, E.Cayirci, “A Survey on Sensor Networks,” *IEEE Communications Magazine*, 40(8): 102-114, 2002.

[20] K.Krauter, R.Buyya, M.Maheswaran, “A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing,” *Software: Practice and Experience*, 32(2): 135-164, 2001.

[21] Donghui Lin, Toru Ishida, Yohei Murakami and Masahiro Tanaka, "QoS Analysis for Service Composition by Human and Web Services," *IEICE Transactions on Information and Systems*, E97-D(4): 762-769, 2014.

[22] B.Tan, L.Massoulie, “Optimal Content Placement for Peer-to-Peer Video-on-Demand Systems,” *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM-11)*, pp.694-702, Shanghai, China, April 10-15, 2011.

[23] L.Yin, G.Cao, “Supporting Cooperative Caching in Ad Hoc Networks,” *IEEE Transactions on Mobile Computing*, 5(1): 77-89, 2006.

[24] Ping Yi, Yue Wu, Futai Zou, Ning Liu, “A Survey on Security in Wireless Mesh Networks,” *IETE Technical Review*, 27(1): 6-14, 2010.

[25] Ting Zhu, A.Mohaisen, Ping Yi, T.Don, "DEOS: Dynamic Energy-Oriented Scheduling for Sustainable Wireless Sensor Networks," *Proceedings of The 31st Annual IEEE International Conference on Computer Communications (INFOCOM-12)*, pp.2363-2371, Orlando, Florida, USA, March 25-30, 2012.

[26] Yichuan Jiang, J.C. Jiang, “Diffusion in Social Networks: A Multiagent Perspective,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(2): 198 – 213, 2015.

[27] Yichuan Jiang, Yifeng Zhou, Wanyuan Wang, “Task Allocation for Undependable Multiagent Systems in Social Networks,” *IEEE Transactions on Parallel and Distributed Systems*, 24(8): 1671-1681, 2013.

[28] Mathijs M.Weerdt, Yingqian Zhang, Tomas Klos, “Multiagent Task Allocation in Social Networks,” *Autonomous Agents and Multi-Agent Systems*, 25(1): 46-86, 2012.

[29] Bo An, Victor Lesser, Kwang Mong Sim, “Strategic Agents for Multi-Resource Negotiation,” *Journal of Autonomous Agents and Multi-Agent Systems*, 23(1): 114-153, 2011.

[30] S.Kartik, C.Siva Ram Murthy, “Task Allocation Algorithms for Maximizing Reliability of Distributed Computing Systems,” *IEEE Transactions on Computers*, 46(6): 719-724, 1997.

[31] C.C.Chiu, Y.S.Yeh, J.S.Chou, “A Fast Algorithm for Reliability-Oriented Task Assignment in a Distributed System,” *Computer Communications*, 25(17): 1622-1630, 2002.

[32] Daniel Grosu, A.T.Chronopoulos, “Noncooperative Load Balancing in Distributed Systems,” *Journal of Parallel and Distributed Computing*, 65(9): 1022-1034, 2005.

[33] Thomas Kunz, “The Influence of Different Workload Descriptions on a Heuristic Load Balancing Scheme,” *IEEE Transactions on Software Engineering*, 17(7): 725-730, 1991.

[34] J. Xu, A. Y. S. Lam, and V. O. K. Li, “Chemical Reaction Optimization for Task Scheduling in Grid Computing,” *IEEE Transactions on Parallel and Distributed Systems*, 22(10): 1624–1631, 2011.

[35] Sheng Di, Cho-Li Wang, “Decentralized Proactive Resource Allocation for Maximizing Throughput of P2P Grid,” *Journal of Parallel and Distributed Computing*, 72(2): 308-321, 2012.

[36] Zhao Tong, Zheng Xiao, Kenli Li, Keqin Li," Proactive Scheduling in Distributed Computing—A Reinforcement Learning Approach," *Journal of Parallel and Distributed Computing*, in press.

[37] A.Y.Zomaya, Y.H.Teh, “Observations on Using Genetic Algorithms for Dynamic Load-Balancing,” *IEEE Transactions on Parallel and Distributed Systems*, 12(9): 899-911, 2001.

[38] T. Eymann, et al., “Decentralized vs. Centralized Economic Coordination of Resource Allocation in Grids,” *Grid Computing*, LNCS 2970: 9-16, 2004.

[39] J.van der Horst, J.Noble, “Distributed and Centralized Task Allocation: When and Where to Use Them,” *Proceedings of the Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshop (SASOW-10)*, pp.1-8, Budapest, Hungary, Sept.27-28, 2010.

[40] Hwa-Chun Lin, C.S.Raghavendra, “A Dynamic Load-Balancing Policy With a Central Job Dispatcher (LBC),” *IEEE Transactions on Software Engineering*, 18(2): 148-158, 1992.

[41] B.Godfrey et al., “Load Balancing in Dynamic Structured P2P Systems,” *Proceedings of Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-04)*, pp.2253-2262, Hong Kong, China, March 7-11, 2004.

[42] William E.Walsh, Michael P.Wellman, “A Market Protocol for Decentralized Task Allocation,” *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98)*, pp.325-332, Paris, France, July 3-7, 1998.

[43] P.Ghosh, N.Roy, S.K.Das, K.Basu, “A Game Theory Based Pricing Strategy for Job Allocation in Mobile Grids,” *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS-04)*, Santa Fe, New Mexico, April 26-30, 2004.

[44] S.Penmatsa, A.T.Chronopoulos, “Game-Theoretic Static Load Balancing for Distributed Systems,” *Journal of Parallel and Distributed Computing*, 71(4): 537-555, 2011.

[45] H.Izakian, A.Abraham, B.T.Ladani, “An Auction Method for Resource Allocation in Computational Grids,” *Future Generation Computer Systems*, 26(2): 228-235, 2010.

[46] H.L.Choi, L.Brunet, J.P.How, “Consensus-Based Decentralized Auctions for Robust Task Allocation,” *IEEE Transactions on Robotics*, 25(4): 912-926, 2009.

[47] S.K.Garg, S.Venugopal, J.Broberg, R.Buyya, “Double Auction-Inspired Meta-Scheduling of Parallel Applications on Global Grids,” *Journal of Parallel and Distributed Computing*, 73(4): 450-464, 2013.

[48] T.Rotaru, H.H. Nägeli, “Dynamic Load Balancing by Diffusion in Heterogeneous Systems,” *Journal of Parallel and Distributed Computing*, 64(4): 481-497, 2004.

[49] O.Shehory, S.Kraus, “Methods for Task Allocation via Agent Coalition Formation,” *Artificial Intelligence*, 101(1-2): 165-200, 1998.

[50] Yichuan Jiang, “Concurrent Collective Strategy Diffusion of Multiagents: The Spatial Model and Case Study,” *IEEE Transactions on Systems, Man and Cybernetics-Part C*, 39(4): 448 - 458, 2009.

[51] K.H.Low, W.K.Leow, M.H.Ang, Jr., “Task Allocation via Self-Organizing Swarm Coalition in Distributed Mobile Sensor Network,” *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pp.28-33, San Jose, California, July 25-29, 2004.

[52] Wanyuan Wang, Yichuan Jiang, “Community-Aware Task Allocation for Social Networked Multiagent Systems,” *IEEE Transactions on Cybernetics*, 44(9): 1529-1543, 2014.

[53] Ö.Babaoglu, H.Meling, A.Montresor, “Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems,” *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS-02)*, pp.15-22, Vienna, Austria, July 2-5, 2002.

[54] Ka-Po Chow, Yu-Kwong Kwok, “On Load Balancing for Distributed Multiagent Computing,” *IEEE Transactions on Parallel and Distributed Systems*, 13(8): 787-801, 2002.

[55] B.Hong, V.K.Prasanna, “Distributed Adaptive Task Allocation in Heterogeneous Computing Environments to Maximize Throughput,” *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS-04)*, Santa Fe, New Mexico, USA, April 26-30, 2004.

[56] Olivier Beaumont *et al*., “Centralized versus Distributed Schedulers for Bag-of-Tasks Applications,” *IEEE Transactions on Parallel and Distributed Systems*, 19(5): 698-709, 2008.

[57] K.Antonis, J.Garofalakis, I.Mourtos, P.Spirakis, “A Hierarchical Adaptive Distributed Algorithm for Load Balancing,” *Journal of Parallel and Distributed Computing*, 64(1): 151-162, 2004.[52]

[58] R.Shah, B.Veeravalli, M.Misra, “On the Design of Adaptive and Decentralized Load-Balancing Algorithms with Load Estimation for Computational Grid Environments*,” IEEE Transactions on Parallel and Distributed Systems*, 18(12): 1675-1686, 2007.

[59] M.J.Zaki, W.Li, S.Parthasarathy, “Customized Dynamic Load Balancing for a Network of Workstations,” *Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing (HPDC-96)*, pp.282-291, Syracuse, NY, USA, Aug.6-9, 1996.

[60] S.Abdallah, V.Lesser, “Modeling Task Allocation Using a Decision Theoretic Model,” *Proceedings of The 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-05)*, pp.719-726, Utrecht, Netherlands, July 25-29, 2005.

[61] Anubhav Das, Daniel Grosu, “Combinatorial Auction-Based Protocols for Resource Allocation in Grids,” *Proceedings of the 19th IEEE International Parallel and Distrib4.(e)(-(,″1(m)5 Twr(□)Tj49A368 -1.4Tc(□)T y|)5.9(5j-29.203h)-10.8.(w)1.9(06*

al Research, 147(2): 430-447, 2003.

[65] A.O.Charles Elegbede, C.Chu, K.H.Adjallah, F.Yalaoui, “Reliability Allocation Through Cost Minimization,” *IEEE Transactions on Reliability*, 52(1): 106-111, 2003.

[66] J.A. Bannister and K.S. Trivedi, “Task Allocation in Fault-Tolerant Distributed Systems,”*Acta Informatica*, 20(3): 261-281, 1983.

[67] V.Cherkassky, C.I.H.Chen, “Redundant Task-Allocation in Multicomputer Systems,” *IEEE Transactions on Reliabilit*y, 41(3): 336-342, 1992.

[68] Y.S.Dai, G.Levitin, “Optimal Resource Allocation for Maximizing Performance and Reliability in Tree-Structured Grid Services,” *IEEE Transactions on Reliability*, 56(3): 444-453, 2007.

[69] P.Ajith Tom, C.Siva Ram Murthy, “Algorithms for Reliability-Oriented Module Allocation in Distributed Computing Systems,” *Journal of Systems and Software*, 40(2): 125-138, 1998.

[70] Y.C.Jiang, Z.Y.Xia, Y.P.Zhong, S.Y.Zhang, “Defend Mobile Agent against Malicious Hosts in Migration Itineraries,” *Microprocessors and Microsystems*, 28(10): 531-546, 2004.

[71] H.R.Faragardi, R.Shojaee, M.A.Keshtkar, H.Tabani, “Optimal Task Allocation for Maximizing Reliability in Distributed Real-Time Systems,” *Proceedings of the IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS-13)*, pp.513-519, Niigata, June 16-20, 2013.

[72] P.Y.Yin, S.S.Yu, P.P.Wang, Y.T.Wang, “Task Allocation for Maximizing Reliability of a Distributed System using Hybrid Particle Swarm Optimization,” *Journal of Systems and Software*, 80(5): 724-735, 2007.

[73] Q.M.Kang, H.He, H.M.Song, R.Deng, “Task Allocation for Maximizing Reliability of Distributed Computing Systems using Honeybee Mating Optimization,” *Journal of Systems and Software*, 83(11): 2165-2174, 2010.

[74] S.Srinivasan, N.K.Jha, “Safety and Reliability Driven Task Allocation in Distributed Systems,” *IEEE Transactions on Parallel and Distributed Systems*, 10(3): 238-251, 1999.

[75] Sarvapali D.Ramchurn, Claudio Mezzetti, Andrea Giovannucii, Juan A.Rodriguez-Aguilar., Rajdeep K. Dash, Nicholas R.Jennings, “Trust-Based Mechanisms for Robust and Efficient Task Allocation in the Presence of Execution Uncertainty,” *Journal of Artificial Intelligence Research*, 35(1): 119-159, 2009.

[76] D.Shen, G.Chen, K.Pham, E.Blasch, “A Trust-based Sensor Allocation Algorithm in Cooperative Space Search Problems,” *Proc. SPIE 8044, Defense, Security, and Sensing. International Society for Optics and Photonics*, doi:10.1117/12.882904, 2011.

[77] A.A.Almutairi, M.I.Sarfraz, S.Basalamah, "A Distributed Access Control Architecture for Cloud Computing," *IEEE Software*, 29(2): 36-44, 2012.

[78] J.Sonnek, A.Chandra, J.B.Weissman, “Adaptive Reputation-Based Scheduling on Unreliable Distributed Infrastructures,” *IEEE Transactions on Parallel and Distributed Systems*, 18(11): 1551-1564, 2007.

[79] Hamdi Yahyaoui, “A Reputation-Based Game for Tasks Allocation,” *Enterprise Information Systems, Lecture Notes in Business Information Processing*, Vol.24: 728-736, 2009.

[80] R.Subrata, Albert Y.Zomaya, B.Landfeldt, “Game-Theoretic Approach for Load Balancing in Computational Grids,” *IEEE Transactions on Parallel and Distributed Systems*, 19(1): 66-76, 2008.

[81] E.Even-Dar, A.Kesselman, Y.Mansour, “Convergence Time to Nash Equilibrium in Load Balancing,” *ACM Transactions on Algorithm*, 3(3): Article 32, 2007.

[82] C.C.Hui, S.T.Chanson, “Allocating Task Interaction Graphs to Processors in Heterogeneous Networks,” *IEEE Transactions on Parallel and Distributed Systems*, 8(9): 908-925, 1997.

[83] B.Ucar, C.Aykanat, K.Kaya, M.Ikinci, “Task Assignment in Heterogeneous Computing Systems,” *Journal of Parallel and Distributed Computi1ng*, 66(1): 32-46, 2006.

[84] R.Bajaj, D.P.Agrawal, “Improving Scheduling of Tasks in a Heterogeneous Environment,” *IEEE Transactions on Parallel and Distributed Systems*, 15(2): 107-118, 2004.

[85] P.R.Ma, E.Y.S.Lee, M.Tsuchiya, “A Task Allocation Model for Distributed Computing Systems,” *IEEE Transactions on Computers*, C-31(1): 41-47, 1982.

[86] S.M.Shatz, J.P.Wang, M.Goto, “Task Allocation for Maximizing Reliability of Distributed Computer Systems,” *IEEE Transactions on Computers*, 41(9): 1156-1168, 1992.

[87] J.M. Pujol, R.Sangüesa, J.Delgado, "Extracting Reputation in Multi Agent Systems by Means of Social Network Topology," *Proceedings of the First International Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, pp.467-474, Bologna, Italy, July 15-19, 2002.

[88] Q.Zhang, L.Cheng, R.Boutaba, “Cloud Computing: State-of-the-Art and Research Challenges,” *Journal of Internet Services and Applications*, 1(1): 7-18, 2010.

[89] Yong Zhang, Zhuo Chang, Francis Y.L. Chin, Hing-Fung Ting and Yung H. Tsin, “Online Uniformly Inserting Points on Grid,” *Information Processing Letters*, 111(16): 773-779, 2011.

[90] Sarit Kraus, Onn Shehory, Gilad Taase, “Coalition Formation with Uncertain Heterogeneous Information,” *Proceedings of the Second International Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, Melbourne, Australia, July 14-18, 2003.

[91] Yawei Li, Zhiling Lan, “A Survey of Load Balancing in Grid Computing,” *Lecture Notes in Computer Science*, Volume 3314, pp 280-285, 2005.

[92] Raj, J.S. and Fiona, R, “Load Balancing Techniques in Grid Environment: A Survey,” *Proceedings of the 2013 International Conference on Computer Communication and Informatics (ICCCI-13)*, pp.1-4, Coimbatore, India, January 4-6 2013.

[93] Nuaimi, K.A., Mohamed, N. ; Nuaimi, M.A. ; Al-Jaroodi, J., “A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms,” *Proceedings of the Second Symposium on Network Cloud Computing and Applications (NCCA-12)*, pp.137-142 London, England, December 3-4, 2012.

[94] Tushar Desai, Jignesh Prajapati, “A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing,” *International Journal of Scientific & Technology Research*, 2(11): 158-161, 2013.

[95] Vishakha Patange, D.D.Gatade, “Survey of Load Balancing Approaches in Peer-To-Peer Network,” *International Journal of Soft Computing and Engineering*, 3(2): 422-424, 2013.

[96] Duong-Ba Thuan, Thinh Nguyen, Bella Bose, Duc A. Tran, “Distributed Client-Server Assignment for Online Social Networks Applications,” *IEEE Transactions on Emerging Topics in Computing*, 10.1109/TETC.2014.2358801, 2014, in press.

Yichuan Jiang (SM’13) received his PhD degree in computer science from Fudan University, Shanghai, China, in 2005. He is currently a full professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. His main research interests include distributed systems, multiagent systems, and social networks. He has published more than 80 scientific articles in refereed journals and conference proceedings, such as the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, the IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, the IEEE Transactions on Systems, Man, and Cybernetics: Systems, the IEEE Transactions on Cybernetics, the ACM Transactions on Autonomous and Adaptive Systems, the Journal of Parallel and Distributed Computing, the International Joint Conference on Artificial Intelligence (IJCAI), the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), and the IEEE International Conference on Tools with Artificial Intelligence (ICTAI). He won the best paper award from PRIMA06 and best student paper awards twice from ICTAI13 and ICTAI14. He is a senior member of IEEE, a member of editorial board of Advances in Internet of Things, an editor of International Journal of Networked Computing and Advanced Information Management, an editor of Operations Research and Fuzziology, and a member of the editorial board of Chinese Journal of Computers.