Chapter 2

State of the Art

Dependability in on-demand computing based transaction processing system has been the vision of computer science research community for more than four decades [38]. This chapter surveys the related work in this area in light of on-demand computing infrastructures and their requirements for transaction processing system. Our goal is to distill the key concepts and analyze their applicability, scope and challenges. A thorough understanding and a precise characterization of the dependability are essential to carry forward the lessons learned from the rich literature in scalable and distributed on-demand computing based transaction processing system. Then we discuss the detailed literature survey of work done in the area covering the attributes of the dependability. We also survey the soft computing based scheduling and task allocation approaches which are applicable in on-demand computing system. Along with them, we also consider that load of the system be balanced.

2.1 On-demand Computing

On-demand computing is an enterprise-level model of technology and computing in which resources are provided on an-needed and when-needed basis. Rather than all at once, on-demand computing allows large enterprises to provide their clients with access to computing resources as they become necessary. Currently large enterprises need to be agile and need the ability to scale resources easily and quickly as the market needs are

rapidly changing. That is why on-demand computing has emerged as the most widely used computing model for the enterprises such as IBM, PayPal, VisaNet etc. They are using this computing environment to ease the transaction processing execution. This type of enterprise model seeks cloud-based or grid-based services to increase efficiency from supply chain, distribution, and service standpoint. Thus on-demand computing is seen as the future of the Internet for the enterprises as the transaction processing is flourishing day by day.

This thesis focuses on dependability in on-demand computing based transaction processing system. But, there are several challenges in on-demand computing based transaction processing system [39, 40]. Dependability and performance enhancement are major challenges in this system. In the literature, there are a lot of research examples on the topic of dependability in cloud, grid and other computing based transaction processing system. In order to meat these challenges, this thesis presents the model, strategies and techniques. These techniques include soft computing based load balanced scheduling and task allocation. By using these techniques, availability, reliability, performability, throughput, resource utilization are to be improved and makespan and load on the system are to be minimized.

2.2 Load Balanced Transaction Scheduling

Scheduling in on-demand computing system is a tough problem due to its characteristics of heterogeneity, autonomy, scalability and adaptability [3, 20, 34]. This problem remains an open research issue. In the literature, there are several scheduling techniques available [41, 42, 43] which optimize one or more characteristic parameters such as makespan[1], turnaround time[2], reliability, availability. They do not accurately reflect load balancing and deadline-constrained tasks like transactions. In this system, the load burden and subsequently unavailability of resources within prescribed deadline, networks and deadline-miss faults are unavoidable and may have an adverse effect on the running applications [44]. Task scheduling is of two types; static scheduling and

[1]The makespan is the total length of the schedule (that is, when all the jobs have finished processing).

[2]In computing, turnaround time is the total time taken between the submission of a program/process/thread/task for execution and the return of the complete output to the customer/user.

dynamic scheduling. The first one is not suitable for on-demand computing based transaction processing system. The second one, i.e., dynamic scheduling imparts an immediate and direct decision without a priori knowledge, and also it hardly wastes processing time [45, 46, 35]. Most of the proposals such as ELISA (Estimated Information Scheduling Algorithm) [13] use a dynamic and decentralized approach. Li *et al.* proposed a hybrid load balancing strategy of sequential tasks for grid computing environments [47]. Therefore, heuristic and meta-heuristic methods have been found to be used for getting the optimal performance and dependability by task scheduling [15]. The problem becomes complicated by the fact that the resources of this system may fail at any point of time. Thus the load on certain resources may increase [48]. Therefore, load balancing also becomes an important issue in this situation to overcome. The problem becomes NP-hard and subsequently it requires a method which can care not only dependability and performance but also load balancing.

There are two distinct ways for load balancing in scheduling; firstly, after sending the transactions to the executing nodes and secondly at the time of transaction scheduling before sending them to the nodes. If the first way is chosen, it creates interprocessor overhead. Due to this excessive interprocessor communication, the performability of the system starts decreasing. Then the system most likely requires an optimal task allocation[3] approach to balance these conflicting factors [49]. But, this approach is not suitable for deadline constrained transactions, because it creates the delay in the system. On the other hand, if the second way of scheduling is chosen, the delay remains minimum. In both the cases, a good scheduling technique is one that adjusts its scheduling strategy according to its changing status of its environment as well as its jobs [36]. Thus, load balanced scheduling is defined as a mechanism which can perform the load balancing strategy before scheduling the transactions and then finds the appropriate nodes to which the transaction should be sent for processing.

The aim of load balanced scheduling in on-demand based transaction processing system should be to balance the load before dispatching the transactions to the available nodes or resources. Scheduling approaches like First Come First Serve (FCFS), Round Robin (RR), Priority Scheduling, Shortest Job First (SJF) [50, 51, 52] can be applied in this

[3]Task allocation and partitioning refers to the way that tasks are chosen, assigned, subdivided, and coordinated (here, within a single colony of social insects

environment. Due to convoy effect[4] in FCFS, indefinite blocking or starvation in Priority Scheduling and SJF, and long time taking in RR, some other scheduling algorithms like Max-min and Min-min [53] came into existence. Along with these methods for scheduling, researchers have been applying also some heuristic and meta-heuristic methods. But meta-heuristic optimization approaches have been the approximate algorithms which are basically used to obtain good enough solutions to hard CO problems [54]. Now the problem is to model the system and the load balanced scheduling algorithms which can improve availability, reliability, performability etc.

There are several studies which worked on scheduling for maximizing availability in computing systems [55, 56, 57]. In this regard, the handling of transaction processing is a challenging task and is an important means to protect the systems from various failures due to unavailability of resources and to complete the transaction by the deadline [7]. There are two reasons for resource unavailability. The first one is the load caused due to the uneven arrival of excessive transactions in the system [58, 59]. The second reason is the failure possibility. In this situation, tasks should be scheduled to those resources where they can be executed within their deadline along with achieving better resource utilization, maximizing performability, minimizing miss ratio and reducing the average response time [35]. Heuristic and meta-heuristic methods can also be used to get optimal performance [15].

Some heuristic algorithms such as Hierarchical Load Balanced Algorithm (HLBA) [34], Most Fit Task First, (MFTF) [60, 61], Dynamic and Decentralized Load Balancing (DLB) scheduling algorithm [35] for grid computing and Earliest Completion Load Balancing (ECLB) for real-time distributed transaction processing [10] have been proposed in the literature. Some artificial life techniques such as genetic algorithm (GA) [29, 30, 31, 32, 42, 1], Hybrid real-coded genetic algorithm [62], Extremal Optimization [2], and Tabu Search (TS) have also been used to solve this type of complex problem. Instead of these, swarm intelligence has also become a research interest to many research scientists of related fields in recent years. The example of a swarm is the bees which swarm around their hive; however, the mechanism can easily be implemented to other systems with a similar working architecture [63, 64, 65].

[4] Convoy Effect is phenomenon associated with the First Come First Serve (FCFS) algorithm, in which the whole Operating System slows down due to few slow processes.

In such environment, a good schedule is required which can adjust its scheduling strategy according to the changing status of the system [66]. Therefore, a dynamic algorithm in job scheduling such as Ant Colony Optimization (ACO) is appropriate for on-demand computing system [36]. Among meta-heuristic optimization approaches, the ACO has gained its popularity widely in solving combinatorial problems [54, 67, 68, 69, 70]. Scheduling using ACO algorithm has been studied extensively in many works. Ant colony optimization was first introduced by M. Dorigo and his colleagues in the early 1990s. ACO is an approximate algorithm which is used to obtain good enough solutions to hard CO problems in a reasonable amount of time [54]. The ACO has been successfully applied to diverse CO problems, which include load balancing [71], scheduling [72], telecommunication networks, reliability analysis [73], and traveling salesman [74]. ACO is an approximate algorithm which is used to obtain good enough solutions to hard CO problems in a reasonable amount of time [54]. The pheromone update is changed by adding encouragement, punishment coefficient and load balancing factor [75]. Multiple kinds of ants were used to find multiple paths for network routing [76]. The idea behind this approach was that each kind of ant sensed their kind of pheromone to find shortest paths. For load balanced scheduling in grid computing, Balanced Ant Colony Optimization (BACO) algorithm [36] was proposed. Load Balancing Ant Colony Optimization (LBACO) algorithm was proposed for cloud task scheduling [77]. ACO along with PSO (Particle Swarm Optimization) was also proposed for load balancing in the grid [71]. We also find Symbiotic Organism Search (SOS) optimization based task scheduling in cloud computing environment [78].

Some artificial life techniques such as GA, Hybrid real-coded genetic algorithm [62], EO, and TS also have been used to solve this type of complex problem. But these approaches for scheduling are unable to give the optimal performance for the transaction due to the deadline and real-time constraints [79, 80] of the transactions. Load balanced scheduling in such a heterogeneous and dynamic environment like a computational grid is difficult [72] and the research in this type of scheduling [37] for on-demand based transaction processing system is considerably rare.

The Honey Bee Optimization (HBO) is another approach which is widely used, works on the concept of a thoroughly integrated unit in gathering its food. They monitor the flower patches in the countryside surrounding their hive; they distribute their foraging activity among these patches so that nectar and pollen are collected efficiently, in sufficient

quantity, and in the nutritionally correct mix. They properly apportion the food they gather between present consumption and storage for future needs [81]. There are several algorithms which are based on this concept. Honey Bee Behavior inspired Load Balancing (HBB-LB) aims to achieve well-balanced load across virtual machines for maximizing the throughput of cloud computing system [82]. Bee Colony Optimization (BCO) was proposed [83] for scheduling independent tasks to identical processors in distributed systems and for Traveling Salesman Problem (TSP) [84]. Another algorithm

TABLE 2.1: Characteristics of the algorithms in the literature

Algorithms	Load balancing	Scheduling	Heuristic	Meta-heuristic	Dynamic	Real-time
ELISA [13]	√	√	√		√	
HLBA [34]	√	√	√		√	
MFTF [60, 61]	√	√	√		√	
DLB [35]	√	√	√		√	
ECLB [10]	√		√		√	√
GA [1]	√	√		√	√	
Hybrid real-coded GA [62]	√			√	√	√
Extremal Optimization [2]		√		√	√	
HBB-LB [82]	√			√	√	
BCO [83]		√		√	√	
MBO [85]		√		√	√	
BLA [86]		√		√	√	
ACO [36]	√	√		√	√	
Randomized Algorithm [36]	√	√	√		√	

Marriage in Honey-Bees Optimization algorithm (MBO) has also been proposed to find near-optimal solutions [85]. Bees Life Algorithm (BLA) has been presented which is used to schedule computation jobs among processing resources efficiently onto the cloud data centers [86]. Further, a bee colony optimization algorithm to job shop scheduling was also proposed [87]. We also have a description on the application of the Bees Algorithm to a combinatorial optimization problem, the scheduling of jobs with a

common due date for a machine to minimize the penalties associated with early or late completion [88]. We can see the characteristics of the related algorithms in **TABLE 2.1**.

But, in the case of a transaction processing in on-demand computing which emphasizes on deadline, we hardly find a solution of the load balanced scheduling problem [79, 80]. The research in load balanced scheduling [37] in on-demand computing based transaction processing is considerably rare.

2.3 Load Balanced Task Allocation

The balanced task allocation strategy in on-demand computing based transaction processing system which allocates the deadline-constrained transactions based on failure possibility of the distributed constituents becomes very important for the reliable execution of the transaction. The balanced task allocation is achieved by assigning the tasks to the balanced nodes. These nodes can be found by maximizing the resource availability in the system. The balanced task allocation in such distributed systems is not easy and is a non-deterministic polynomial-time hard (NP-hard) problem. The problem becomes complicated by the fact that the resources of these systems may fail at any point of time. Several number of research works have been done on the task allocation in different distributed computing environments. The balanced task allocation in an on-demand computing system deserves special attention. The method should be implemented by determining the node at which a specific transaction is to be executed within its specific deadline. Therefore, the balanced task allocation becomes important for transaction processing in on-demand computing system [89].

In this field, Shatz *et al.* proposed a mathematical model for task allocation in distributed system and computed cost of the system [90, 22]. On this same concept, many research has been seen in literature with the consideration of cost optimization or reliability optimization [25, 91, 92]. Yen *et al.* also proposed task allocation and assessed the reliability for multiple-agent cooperating systems [93]. Kang *et al.* proposed task allocation for maximizing reliability of distributed computing system using a meta-heuristic algorithm (honeybee mating optimization) [94]. By improving task allocation approach, Kartik *et al.* proposed a task allocation algorithm to maximize

reliability of redundant distributed computing system [91, 24]. Peroza *et al.* considered random node failure and proposed task allocation by maximizing reliability in distributed system [95]. Kishor *et al.* also discussed about task allocation technique to maximize the composite performance and dependability of distributed system [92]. Hsieh *et al.* also focussed on task allocation for maximizing reliability of distributed system [96]. Peng-Yeng *et al.* proposed a hybrid particle swarm optimization based task allocation algorithm for maximizing reliability of distributed system [97]. A large number of the solutions of task allocation problem in the literature are based on simple system models, which do not accurately reflect load balancing and deadline-constrained task allocation. But, Tripathi *et al.* proposed a model to overcome the problem of load balanced task allocation in distributed system [21, 42]. We see that the consideration of reliability is essential for the next generation of accurate and efficient task allocation.

The objective of the balanced task allocation is to minimize the load and to minimize the probability of failures of the transactions. There are mainly three terms which should be focussed while allocating the transactions; load, deadlines, and the probability of failures. It is a complex task to minimize all three objectives at the same time. Therefore, task allocation algorithm which can take into account all these three objectives must be devised. In this direction, Dogan and Ozgunar proposed a reliable dynamic level scheduling algorithm (RDLS) by incorporating three reliability cost functions into making dynamic level [98]. But the approach is not able to improve the system reliability. Fault-tolerant is another well-known technique which can improve the system reliability [99, 100, 101, 102]. When problem size increases, the computing time and memory requirements also increase. Thus, most of the exact approaches are limited to solve only moderately sized problem instances [23, 94]. Therefore, the balanced task allocation problem on the distributed computing systems has been proved to be NP-hard [91, 103].

The heuristic algorithms can derive near-optimal or optimal solutions within reasonable computing time [94]. In recent years, most research has been on developing meta-heuristic algorithms to solve the problem. For example, Tripathi *et al.* presented balanced task allocation solution based on genetic algorithm (GA) [42]. Although they did not consider the problem of reliability, the approach for balanced task allocation was a positive direction for the reliability evaluation of the distributed computing system including on-demand computing. Then, in 2001, Vidyarthi and Tripathi presented the

solution of the reliability problem using GA to quickly find a near-optimal balanced task allocation [1]. Other evolutionary algorithms such as SA [104], ACO [73], PSO [97], HBMO [94], Biogeography-based Optimization [105] were also used in grid environment for getting the optimal solution for the specified problem.

This thesis presents social spider optimization (SSO) based algorithm for finding near-optimal balanced task allocation. To achieve performance, availability, reliability and consistency, data must be readily accessible in a data warehouse, backup procedures must be in place and the recovery process must be in place to deal with system failure, human failure, computer viruses, software applications or natural disasters. The analysis and evaluation of throughput, availability, reliability and perforability in on-demand computing based transaction processing system become challenging issues and attract more and more attentions of researchers.

This dissertation presents the approaches with the published solutions to these problems. The load balanced scheduling problem is dealt in [26] by improving throughput, resource utilization and by minimizing the makespan and load in the system. The performability factor has been focused in [106] and reliability problem is dealt in [107]. All these researches use meta-heuristic approaches with scheduling or task allocation strategies. Apart from meta-heuristic approaches, heuristic algorithms are also proposed for finding the solution to the problem. The algorithm proposed in [89] is a heuristic based algorithm while the algorithm proposed in [19] is for adaptability of the system.