Chapter 6

Load Balanced Transaction Allocation considering Reliability

Balanced task allocation is one of the methods which can be used to maximize the performance and reliability in on-demand computing based transaction processing system. On-demand computing is an increasingly popular enterprise model. It provides computing resources to the user as needed which may be maintained within the user's enterprise, or made available by a service provider. The balanced task allocation in such environment is known to be an NP-hard. The reliability is a measure of trustworthiness of the system while executing the task. So, we derive the reliability formula for on-demand computing based transaction processing system considering resource availability. We propose the balanced task allocation based on social spider optimization methodfor this problem. The LBTA_SSO is based on the cooperative behavior of social-spiders to find a collection of task allocation solutions. We modified five existing algorithms to obtain the task allocation algorithms; Honey Bee Optimization (HBO), Ant Colony Optimization (ACO), Hierarchical Load Balanced Algorithm (HLBA), Dynamic and Decentralized Load Balancing (DLB), and Randomized Algorithm respectively. Then, we compared the proposed algorithm with these modified algorithms. The results show that our algorithm works better than the modified existing algorithms. We compared our algorithms on two different platform; grid [107] and cloud [117].

TABLE 6.1: Definitions

Decision Variables

$$r_i = \begin{cases} 1, & \text{if resources are available} \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ik} = \begin{cases} 1, & \text{if transaction } T_i \text{ is scheduled to execute on node } N_k \\ 0, & \text{otherwise} \end{cases}$$

6.1 Load Balanced Transaction Allocation Model

Our objective is to find a load balanced task allocation by maximizing the system reliability which is the probability for the successful completion of distributed programs with requirements that all the allocated processors and involved communication links are operational during the execution lifetime with no deadline-miss.

6.1.1 Assumptions

- Transactions arrive according to a Poisson process (i.e., exponentially distributed interarrival times).
- The failure, repair and transaction processing times are exponentially distributed.
- There is an infinite buffer space for queueing transactions in the system. In modern system, it is likely because memory is fairly cheap.
- The probability of having i transactions in on-demand computing system follows a simple M/M/c model. Because transaction buffer sizes are assumed to be quite large.
- The network topologies in the system are cycle-free. It means that there will be a unique path between any pair of edges.
- The system considers the steady-state user-perceived availability of the resources. It is strongly based on the performance (especially the response time) of the system.

6.2 Problem Formulation

The consistency of the balanced task allocation method used in any system can be measured by the reliability of the system. The reliability of the transaction processing in on-demand computing system is the probability that over a given time t, the entire transaction executes properly without failure. The failure of transaction in this system is caused not only by node and link fault but also by the deadline-miss fault. Therefore, this chapter also introduces the deadline-miss fault while formulating the reliability.

6.2.1 Reliability Model

Resource availability plays an important role when a transaction is executed in on-demand computing system within its deadline without any failure. The reliability formulation for distributed computing system (DCS) by Shatz [22] considered the failure caused by node and link fault only. We formulate the reliability by introducing the deadline-miss fault [44] and the steady-state user-perceived availability [18] of resources.

The reliability formulation expressed by [22] is given as follows:

$$R_{k,kb}(X) = \left[\prod_{k=1}^{n} R_k(X)\right] \cdot \left[\prod_{k=1}^{n-1}\prod_{b>k} R_{kb}(X)\right] \tag{1}$$

where the node reliability $R_k(X)$ of a node N_k during a time interval t has been computed as follows: $e^{-\gamma_k \sum_{i=1}^{m} x_{ik} e_{ik}}$ when $\sum_{i=1}^{m} x_{ik} e_{ik}$ is the total elapsed time t for executing the transactions assigned to N_k. While the communication link reliability $R_{kb}(X)$ at a time interval t has been computed as follows: $e^{-\sigma_{kb} \sum_{i=1}^{m} \sum_{g \neq i} x_{ik} x_{gb} (cost_{ig}/w_{kb})}$ where the total elapsed time for transmitting the transaction communication via l_{kb} is $\sum_{i=1}^{m} \sum_{g=1}^{m} x_{ik} x_{gb} (cost_{ig}/w_{kb})$.

The system reliability that there is no transaction deadline-miss fault in addition to nodes and communication links are operational during the elapsed time for the execution can be computed as follows:

$$R_{k,kb,DM_i}(X) = \left[\prod_{k=1}^{n} R_k(X)\right] \cdot \left[\prod_{k=1}^{n-1}\prod_{b>k} R_{kb}(X)\right] \cdot \left[\prod_{i=1}^{m} R_{DM_i}(X)\right] \tag{2}$$

where the transactions are the steady-state and follow queuing system model $M/M/c$[1] [114] and $R_{DM_i}(X)$, the probability that there is no deadline-miss with rate ψ_i when transaction T_i is scheduled on N_k can be computed by using the Markov model as follows:

$$R_{DM_i}(X) = e^{-\psi_i \cdot \left[\frac{1}{\mu} + \Pi_0 \cdot \frac{\rho (c\rho)^c}{c!(c\mu-\lambda)(1-\rho)}\right]}, \quad \forall c \in N \tag{3}$$

where Π_0 is given by

$$\Pi_0 = \left[\sum_{i=0}^{c-1} \frac{(c\rho)^i}{i!} + \frac{(c\rho)^c}{c!(1-\rho)}\right]^{-1} \tag{4}$$

where $\rho = \frac{\lambda}{c.\mu} < 1$.

Finally, the reliability of transaction in on-demand computing system considering the conditional steady-state user-perceived availability of resources [118], is computed as follows:

$$R_{k,kb,DM_i,A_\lambda}(X) = \left[\prod_{k=1}^{n} R_k(X).A_\lambda\right] \cdot \left[\prod_{k=1}^{n-1}\prod_{b>k} R_{kb}(X)\right] \cdot \left[\prod_{i=1}^{m} R_{DM_i}(X)\right] \tag{5}$$

where A_λ is the steady-state availability of the resources under the load λ. We take its formula from [18] expressed as follows:

$$A_\lambda = \sum_{c=1}^{n} A_{c,\lambda} Q_c \tag{6}$$

where $\forall c = 1,...,n$ and $A_{c,\lambda}$ of available servers which has been computed as $\sum_{i=0}^{K} r_i \Pi_i$ with the steady-state probabilities for the model which are given as follows:

$$\Pi_i = \frac{(c\rho)^i}{i!}\Pi_0, \quad 1 \le i \le c-1 \tag{7}$$

[1] $M/M/c$ represents the queue length in a system having c number of servers where jobs arrive following Poisson process and job service time have exponential distribution. Here first M represents memoryless with λ arrival rate of jobs, second M represents service rate of jobs with μ and c represents the number of servers.

$$\Pi_i = \frac{c^c \rho^i}{c!} \Pi_0, \quad i \geq c \tag{8}$$

In the above formulation, $\rho = \frac{\lambda}{c\mu}$ and Q_c has been expressed as: $Q_c = \frac{n!}{c!(n-c)!} q^c (1-q)^{n-c}$, where $q = \frac{\eta}{\gamma+\eta}$ be the availability of a single server.

The reliability has been expressed regarding $cost(X)$ in [22]. Similarly, the reliability can be expressed as follows:

$$R_{k,kb,DM_i,A_\lambda}(X) = e^{-cost(X)} \tag{9}$$

From **Eq.**(9) it is evident that minimizing the $cost(X)$ will maximize the reliability $R_{k,kb,DM_i,A_\lambda}(X)$, therefore, the addressed task allocation problem is formulated as:

$$\text{minimize} \quad cost(X) \tag{10}$$

$$\text{subject to} \quad \sum_{k=1}^{c} x_{ik} = 1, \; \forall i = 1, \ldots, m \tag{11}$$

$$\sum_{i=1}^{m} y_i x_{ik} \leq M_k, \; \forall k = 1, \ldots, c \tag{12}$$

$$\sum_{i=1}^{m} z_i x_{ik} \leq C_{N_k}, \; \forall k = 1, \ldots, c \tag{13}$$

$$x_{ik} \in [0,1] \; \forall i,k \tag{14}$$

Constraint 11 states that each transaction is assigned to exactly one processor. Constraint 12 ensures that the total memory y_i required by all transactions to node k does not exceed the available memory M_k of the node. Constraint 13 ensures that the total processing load z_i required by all the transactions assigned to node k does not exceed the available processing capacity C_{N_k} of the node. Constraint 14 guarantees that x_{ik} is binary variable.

Theorem 6. *If A_λ be the availability under load λ and $R_k(X)$ with task allocation X be the probability that k number of servers are running without failure, then the reliability of transactions in on-demand computing system will be computed as $\prod_{k=1}^{n} R_k(X).A_\lambda$.*

Proof. The probability that there are no failures of node k in time interval t is expressed as $R_k(X) = e^{-\int_0^t \gamma_k t}$. Now, the availability of resources is the prime factor for execution of transactions. Because the transactions have their respective deadlines within which they have to execute. If the resource is available within the deadline, the transactions are successfully executed. Therefore, the availability of the resources is important for a transaction execution and also to find the reliability of the system. Then the probability that the transaction has no failures when executing at node k available at time t is given by $\prod_{k=1}^{n} R_k(X).A_\lambda$.

□

Theorem 7. *If transaction T_i scheduled on nodes N_k is modeled as $M/M/c$ queuing system, then the probability that there is no deadline-miss is $e^{-\psi_i.\left[\frac{1}{\mu}+\Pi_0.\frac{\rho(c\rho)^c}{c!(c\mu-\lambda)(1-\rho)}\right]}$.*

Proof. The $M/M/c$ queuing system is identical to the $M/M/1$ system except that there are c servers. A transaction at the head of the queue is routed to any server that is available. According to queuing theory [114], the probability that there are m number of transactions waiting in the queue is given by

$$\Pi_i = \begin{cases} \Pi_0.\frac{(c\rho)^i}{i!}, & i \leq c \\ \Pi_0.\frac{c^c\rho^i}{c!}, & i > c \end{cases} \tag{15}$$

where ρ is calculated as $\rho = \frac{\lambda}{c\mu} < 1$. Here Π_0 (**Eq.**4) can be calculated by using **Eq.**(15) where the condition $\sum_{i=0}^{\infty} \Pi_m = 1$.

The probability that transactions are waiting in the queue is given by

$$\begin{aligned} P\{\text{Transaction Queuing}\} &= \sum_{i=c}^{\infty} \Pi_i = \sum_{i=c}^{\infty} \frac{\Pi_0 c^c \rho^i}{c!} \\ &= \frac{\Pi_0 (c\rho)^c}{c!} \sum_{i=c}^{\infty} \rho^{i-c} \end{aligned} \tag{16}$$

According to [114], $P\{\text{Transaction Queuing}\}$ in **Eq.**16 can be computed as $\frac{\Pi_0(c\rho)^c}{c!(1-\rho)}$. Therefore, the average delay by per transaction is computed as $\frac{1}{\mu} + \Pi_0.\frac{\rho(c\rho)^c}{c!(c\mu-\lambda)(1-\rho)}$.

Finally, if ψ_i is the deadline-miss rate, the probability that there is no deadline-miss of a transaction can be calculated as $e^{-\psi_i \cdot \left[\frac{1}{\mu} + \Pi_0 \cdot \frac{\rho(c\rho)^c}{c!(c\mu-\lambda)(1-\rho)}\right]}$.

□

Theorem 8. *If transactions T_i scheduled on nodes N_k is modeled as $M/M/1$ queuing system, then the probability that there is no deadline-miss is $R_{DM_i} = e^{\psi_i \cdot \frac{1}{\mu-\lambda}}$ where $\rho = \frac{\lambda}{\mu}$.*

Proof. In $M/M/1$ system, if $\rho < 1$, then the probability that there are i number of transactions waiting in the queue is given by

$$\Pi_{i+1} = \rho^{i+1}\Pi_0, \quad i = 0, 1, \ldots \tag{17}$$

As according to [114], the probabilities Π_i are all positive and when they are added up to unity, we find

$$1 = \sum_{m=0}^{\infty} \Pi_i = \sum_{i=0}^{\infty} \rho^i \Pi_0 = \frac{\Pi_0}{1-\rho} \tag{18}$$

When we add Eq.(17) and Eq.(18), we obtain

$$\Pi_i = \rho^i(1-\rho), \quad i = 0, 1, \ldots$$

In the steady-state system, the number of transactions can be calculated as $\sum_{i=0}^{\infty} i\Pi_i$ i.e., $\sum_{i=0}^{\infty} i(1-\rho)\rho^i = \rho(1-\rho)\frac{1}{(1-\rho)^2} = \frac{\rho}{1-\rho}$. Therefore, the average delay by per transaction (waiting time in the queue plus service time) is given by $\frac{\rho}{\mu(1-\rho)}$. Finally, the probability that there is no deadline-miss is calculated as $e^{\psi_i \cdot \frac{1}{\mu-\lambda}}$ using $\rho = \frac{\lambda}{\mu}$.

□

6.3 Proposed Algorithm

In this section, we propose the balanced task allocation based on social spider optimization (LBTA_SSO) algorithm which aims at finding an assignment of transactions in on-demand computing system to a set of balanced nodes subject to the resource constraints.

6.3.1 Social Spider Optimization

SSO which was proposed by Cuevas *et al.* in 2013 [119] is based on the cooperative behavior of social-spiders. SSO is inspired by the complex cooperating groups organized by social insect societies because cooperative groups can manipulate and exploit resources and brood in a better way by allowing the task specialization among group members [119]. Because of such behavior, SSO serves as a function optimizer by which a social insect colony functions as an integrated unit that not only possesses the ability to operate in a distributed manner, but also to undertake enormous construction of global projects [120].

In SSO, the search space of the optimization problem is formulated as a hyper-dimensional spider web. Each position on the web represents a feasible solution to the optimization problem. Each spider holds a position. The quality (fitness) of the solution is based on the objective function. When a spider moves to a new position, it generates vibration. The vibration holds the information of the spider. Other spiders get the information upon receiving the vibration.

There are three phases in SSO: initialization, iteration, and final.

- **Initialization:** In this phase, the algorithm defines the objective function and its solution. The values of parameters used in SSO are assigned. The positions of spiders are randomly generated in the search space, with their calculated fitness value.
- **Iteration:** In this phase, many iterations are performed by the algorithm. All spiders in the web move to a new position. Each iteration can be further divided into sub-steps: fitness evaluation, vibration generation, mask changing, random walk, and constraint handling.

- **Final:** The final phase is the constraint handling. There are many methods to handle the boundary constraints.

Algorithm 4 LBTA_SSO

Assign values to the required parameters
Create the population of nodes
Initialize Vib_{tar} for each node N_k ▷ Initialization
while transaction queue is not empty **do**
  for each Transaction T_i **do**
    for each node N_k **do**
      Calculate vibration of each node
      if $Vib_{tar} > Vib_{thres}$ **then**
        Assign T_i to N_{tar} ▷ Assignment of Transaction T_i to node N_j having vibration higher than threshold
      else
        Select Vib_{best} from Vib_k
        if $Vib_{best} > Vib_{tar}$ **then**
          $Vib_{tar} \leftarrow Vib_{best}$
          Assign T_i to N_{best} ▷ Assignment of Transaction T_i to node N_j the highest vibration
        else
          Assign T_i to N_{tar}
        end if
      end if
    end for
  end for
end while

6.3.2 Constraints Description

- ***Spider*:** The spiders are the agents of LBTA_SSO to perform optimization. The node represents the target of the spiders.

- ***Fitness assignation*:** Every node N_k has a load L_{ik} which represents the solution quality of node. The load of each node is calculated. The constraint 13 states that the minimization of load on the system will ensure the minimization of cost of the system. The fitness function of the problem with respect to LBTA_SSO is given by **Eq.**(10).

- *Vibration* We formulate the vibration of each node by using load perceived by the corresponding node. The formulation states that if the node is overloaded, the intensity of vibration emitted by that node to attract the other spiders is weak while lightly loaded node emits stronger vibration. The vibration perceived by the individual N_k is modeled according to the following equations:

$$Vib_{N_k} = e^{-L_{ik}} \tag{19}$$

The transactions T_i are randomly generated. A population of nodes are created by satisfying the fitness function of the problem.

In the iteration phase, all incoming transactions are assigned to the suitable nodes and fitness values of the respective nodes are calculated. Each node generates vibration using **Eq.**(19). The scheduler's decision for the node selection to assign the next transaction from the scheduling queue is based on the vibration produced by the node. In this process, each transaction will receive $|n|$ number of different vibrations generated by the nodes. After receiving the vibrations, the scheduler will select the strongest vibration Vib_{best} and compare it with Vib_{tar}. The transaction will be assigned to the node with the strongest vibration Vib_{best}. The vibration of the assigned node will become Vib_{tar}.

The node selection depends on the final movement of attraction or repulsion on several random phenomena. We model the selection of node as a stochastic decision. Consider Vib_{thres} as threshold value of vibration. The attraction or repulsion movement generation depends on the operator which is modeled as follows:

$$N_{k+1} = \begin{cases} select\ N_k, \text{if } Vib_{N_k} \geq Vib_{thres} \\ reject\ N_k, \text{if } Vib_{N_k} < Vib_{thres} \end{cases} \tag{20}$$

Vib_{thres} can be calculated as

$$Vib_{thres} = e^{-\frac{n}{2}} \tag{21}$$

How the LBTA_SSO works is described below.

Line 1 creates the population of nodes n. Line 2 initializes the target vibration of each node as Vib_{tar}. Until the transaction queue is not empty, lines $4-21$ run **while** loop repeatedly selecting the random nodes to search the optimal node for the requested transaction. In each iteration of this **while** loop, the algorithm performs the following operations:

Lines $5-20$ run the **for** loop for each T_i. Lines $6-19$ again run a **for** loop, but this time for each node N_k from the population of n nodes.

Line 7 calculates the vibration of each node as Vib_{N_k} which is calculated using **Eq.** (19). Line 7 checks whether $Vib_{tar} > Vib_{thres}$ (here $Vib_{thres} = 0.018$). If it is, then line 9 assigns the transaction T_i to N_{tar} node. Otherwise, line 11 starts searching the best vibration Vib_{best} from the set of node. Line 12 again checks whether $Vib_{best} > Vib_{tar}$. If it is then, line 14 assigns T_i to N_{best}. Otherwise, line 16 assigns the transaction T_i to N_{tar} node.

Then line 16 updates the fitness value to all nodes. Fitness value means the status of the load. We repeat the iterations of the **while** until all the transactions are not scheduled.

6.4 Applying the Algorithm

We applied the proposed algorithm on a complete undirected graph denoted by $G = (N, E)$ where N is the set of nodes; $E = N \times N$ is the set of edges between the nodes. **FIGURE 6.1** gives an illustrative example how the LBTA_SSO works. Suppose there are m number of transactions $(T_1, T_2, \ldots, T_m)$ which arrive at the system with available nodes (suppose set N has $n = 8$).

Initialization: In this phase of the algorithm, let us initialize the load of each node of the graph as: $L_1 = 2$, $L_2 = 5$, $L_3 = 4$, $L_4 = 3$, $L_5 = 7$, $L_6 = 5$, $L_7 = 8$, and $L_8 = 0$.

Iteration: In this phase, the algorithm selects the feasible solution at each iteration. Initially, at each node, a spider is randomly placed. Let the spider representing the transaction T_i is at N_1. Then the algorithm calculates the vibrations of spiders at all the nodes. Every spider at each iteration visits the nodes according to the vibration released by the nodes. Thus, in one iteration each spider has to traverse the graph to search the best node. This best node is a feasible solution. Then each node in the same iteration might have traversed the graph and have searched the best node. Among all the feasible solutions found in the same iterations by the spiders, the node which has the highest vibration is selected as the optimal solution.

Let us suppose the spider at node N_1 starts to traverse the graph to find the best node (lightly loaded node). At first, the vibration of the node is calculated using **Eq.** (19). We see $Vib_1 = 0.135335283$. Then threshold value is calculated using $Vib_{thres} = e^{-\frac{n}{2}}$ given in **Eq.** (21). Since $n = 8$, $Vib_{thres} = e^{-\frac{n}{2}}$ is calculated as 0.018315639. Vib_1 is greater than the threshold value 0.018315639. Hence, N_1 becomes the solution and T_i is assigned to N_1. Now, the load of the node N_1 is incremented by 1 and it becomes 3. Again the vibration of the node N_1 is calculated as 0.049787068 which is greater than Vib_{thres}. Then node N_1 is again selected as shown in the **FIGURE** (**6.1**a). Now the Vib_1 is 0.018315639 which is equal to Vib_{thres}. Then the algorithm starts searching the nodes with the maximum vibrations. The maximum vibration is shown by node N_8 which is equal to 1. The algorithm again compares the vibration of N_8 with Vib_{thres}. Since it is greater than Vib_{thres}, the node is selected four times consecutively until the load of N_8 becomes 4 (as shown in **FIGURE** (**6.1**b)). In a similar way, the algorithm selects N_4 two times as shown in **FIGURE** (**6.1**c) and **FIGURE** (**6.1**d). Then node N_3 and N_1 are selected as shown in **FIGURE** (**6.1**e) and **FIGURE** (**6.1**f) respectively.

Final: Finally the algorithm updates the fitness value of all the nodes.

The process continues until the queue for incoming transactions is not empty.

6.5 Simulation and Result Analysis

The balanced task allocation of transactions in on-demand computing system is evaluated through simulations with Colored Petri Nets (CPNs or CP-nets) [121, 122]. We use the Poisson process for modeling of various faults occurring in the system. In our simulations, the grid scenario is based on Czech National Grid Infrastructure Metacentrum project. The cloud scenario is based on Amazon's Elastic Compute Cloud (EC2). This cloud scenario is made up of 8 sites with homogeneous resource pool. The transaction traces used in the simulations specify a set of parameters such as the transaction identifier, associated transaction user priority, the set of properties to be met in the target resource and arrival time to the scheduler.

6.5.1 Result Evaluation

We simulated the proposed algorithm with two different scenarios; first on grid computing and second on cloud computing. We modified five known algorithms for the

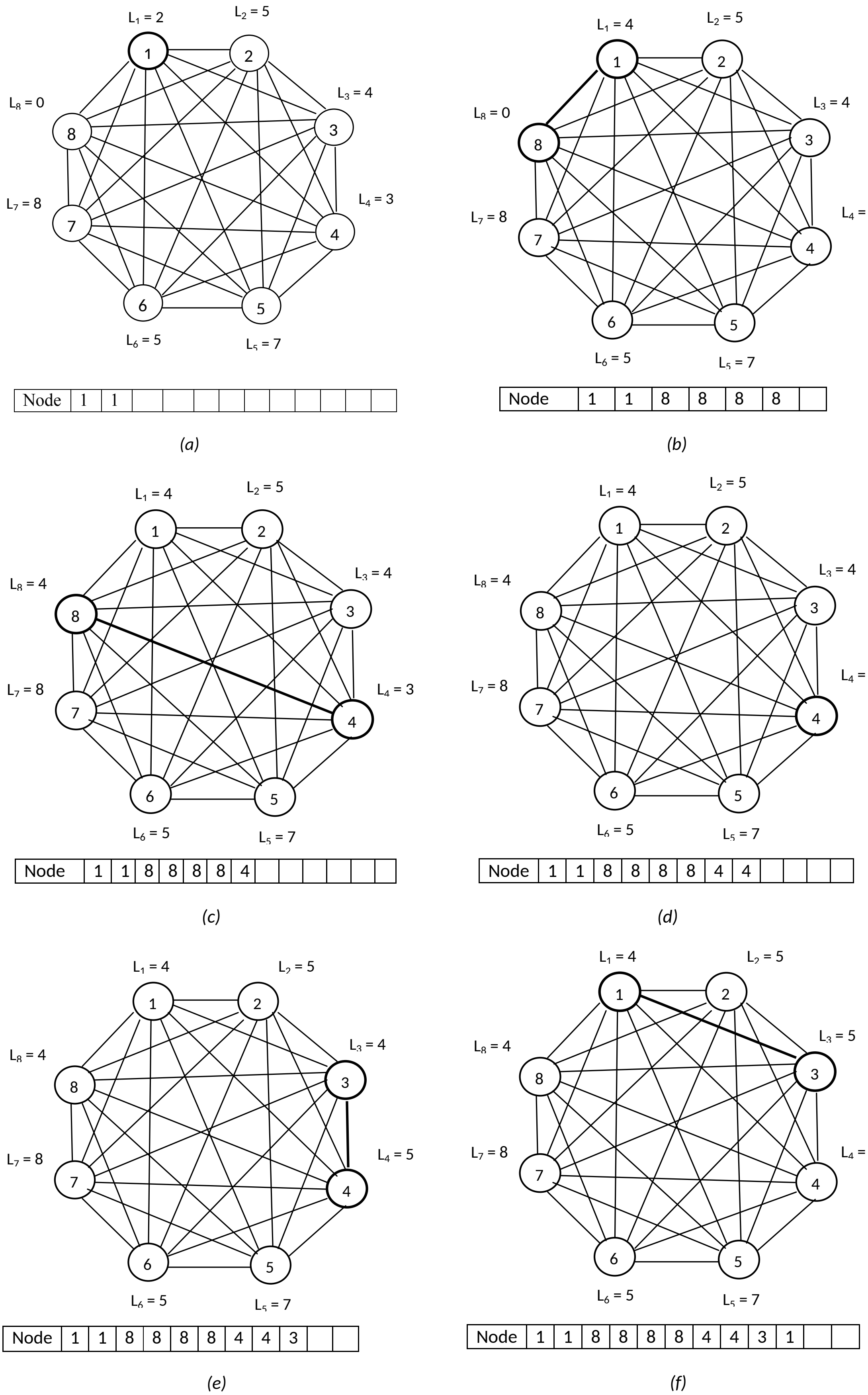

FIGURE 6.1: Working example of the LBTA_SSO when N=8

TABLE 6.2: Normality Shapiro-Wilk tests and Wilcoxon statistical tests for best results found for LBTA_SSO, HBO, ACO, HLBA, DLB, Randomized algorithms in grid computing and cloud computing scenarios

Data	Shapiro-Wik W	p-value
LBTA_SSO	0.9271	0.00200
HBO	0.8281	0.00240
ACO	0.8281	0.00250
HLBA	0.8728	0.00248
DLB	0.8568	0.00250
Randomized	0.8428	0.00250

purpose of comparison with the proposed algorithm. We compared the performances of the proposed algorithm with these five algorithms; HBO, ACO [36], Hierarchical Load Balanced Algorithm (HLBA) [34], Dynamic and Decentralized Load Balancing (DLB) algorithm [35], and Randomized algorithm with random selection method [36]. We ran each algorithm 40 times at each time unit value for every problem instances to get the result.

6.5.1.1 Resource availability

Resource availability is one of the objectives of this chapter. The maximum resource availability is needed to maximize the reliability of the system (as shown in **Eq.** (5) and **Eq.** (6)). We have the comparative results of resource availability in grid computing system along with several iterations from 100 to 1000 in 40 runs using the mentioned algorithms as shown in **TABLE 6.3** and **6.5** with p-values (as given in **TABLE 6.2**). Specifically, **TABLE 6.3** presents the mean result which is achieved by the populations (average) with the associated standard deviation, 95% confidence interval and the best result (Maximum). It is evident that the resource availability with the LBTA_SSO in grid environment obtains better average convergence results (iteration 1000) than the system with other algorithms.

Similarily we have the comparative results of resource availability in cloud computing system along with several iterations from 100 – 1000 in 40 runs using the mentioned algorithms as shown in **TABLE 6.4** and **6.6** with p-values (as shown in **TABLE 6.2**). **TABLE 6.4** presents the mean result achieved by the populations with the associated standard deviation, 95% confidence interval and the best result (Maximum). The

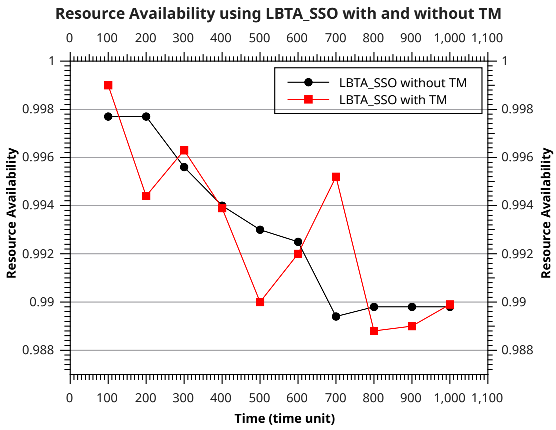

FIGURE 6.2: Resource Availability in LBTA_SSO with and without TM in grid computing based environment

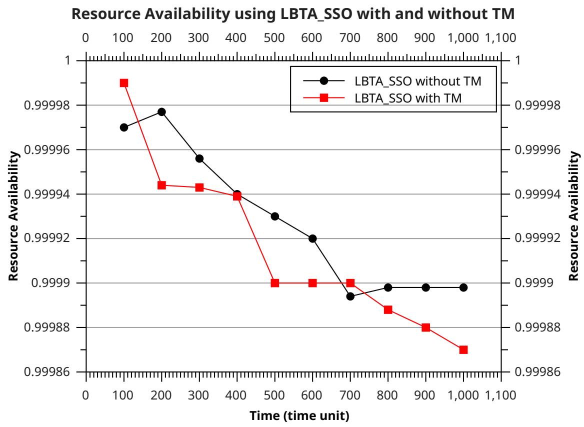

FIGURE 6.3: Resource Availability in LBTA_SSO with and without TM in cloud computing based environment

resource availability in the cloud computing system is higher than that in grid computing system. Also, the LBTA_SSO works better than other algorithms for both grid and cloud environments.

Here we have the comparison of resource availability of our algorithm when used with and without transaction management in on-demand computing based transaction processing system. We see in FIGURE 6.2 which shows the result of grid computing based transaction processing system while in FIGURE 6.3 we see the result of cloud computing based transaction processing system. In FIGURE 6.2, the proposed algorithm performs better for transaction processing in grid computing system than the computational grid without transaction processing. The improvement in resource availability is caused by the minimization of load on each node for deadline-constrained transaction. A maximum number of transactions can successfully get executed when they get completed within their deadlines. But in the case of computational grid, the jobs may not be time-bound (deadline). We see in FIGURE 6.3 which shows the same improvement that we get in the case of grid computing scenario.

Here we see the resource availability when we used the transaction management in the simulation as shown in FIGURE 6.4 which shows the comparative analysis of resource availability when all the algorithms are applied in grid computing based scenario. The result shows that the proposed algorithm LBTA_SSO outperforms the other algorithms. In

TABLE 6.3: Resource Availability in case of grid computing system for 40 simulations

Strategy	Iteration	Average	Standard deviation	Confidence Interval (95%)		Maximum
LBTA_SSO	1000	0.999	0.0062	0.9996185120	0.9996187600	0.999619
	900	0.9944	0.0063	0.9948444081	0.9948445860	0.994900
	800	0.9963	0.0064	0.9966686174	0.9966690000	0.996669
	700	0.9939	0.0065	0.9942242200	0.9942243500	0.994230
	600	0.992	0.0067	0.9922728699	0.9922729793	0.992200
	500	0.99	0.0066	0.9902944526	00.9902945707	0.991250
	400	0.9952	0.0068	0.9942242200	0.9942243500	0.995300
	300	0.9888	0.0069	0.9890433664	0.9890434639	0.988900
	200	0.989	0.007	0.9892327733	0.9892328667	0.989100
	100	0.9899	0.0071	0.9901239829	0.9901240727	0.989910
HBO	1000	0.9977	0.0062	0.9996185120	0.9996185120	0.997766
	900	0.9965	0.0063	0.9948444081	0.9948444081	0.996600
	800	0.9968	0.0064	0.9966686174	0.9966686174	0.996900
	700	0.9931	0.0068	0.9942242200	0.9954563990	0.993200
	600	0.9896	0.0067	0.9922728699	0.9922728699	0.999700
	500	0.9881	0.0066	0.9902944526	0.9902944526	0.988500
	400	0.9939	0.0065	0.9942242200	0.9942242200	0.994250
	300	0.9887	0.0069	0.9890433664	0.9890433664	0.988900
	200	0.989	0.007	0.9892327733	0.9892327733	0.989100
	100	0.9895	0.0071	0.9901239829	0.9901239829	0.989600
ACO	1000	0.99999	0.0062	0.9996185120	0.9996185120	0.999989
	900	0.9958	0.0063	0.9953556364	0.9962443636	0.996500
	800	0.9932	0.0064	0.9928314196	0.9935685804	0.994500
	700	0.9928	0.0065	0.9924758125	0.9931241875	0.993850
	600	0.9911	0.0067	0.9908271575	0.9913728425	0.992456
	500	0.9911	0.0066	0.9908055769	0.9913944231	0.990810
	400	0.9905	0.0068	0.9902436267	0.9907563733	0.990850
	300	0.99	0.0069	0.9890433664	0.9902433420	0.991250
	200	0.9894	0.007	0.9891672500	0.9896327500	0.989900
	100	0.9879	0.0071	0.9876760396	0.9881239604	0.988500
HLBA	1000	0.989	0.0062	0.9883815376	0.9896184624	0.989200
	900	0.989	0.0063	0.9885556275	0.9894443725	0.989100
	800	0.9882	0.0064	0.9878314122	0.9885685878	0.988500
	700	0.9906	0.0065	0.9902758060	0.9909241940	0.991200
	600	0.9887	0.0066	0.9884055710	0.9889944290	0.988900
	500	0.9885	0.0067	0.9882271520	0.9887728480	0.988600
	400	0.9906	0.0065	0.9902758060	0.9909241940	0.991200
	300	0.9855	0.0069	0.9852566532	0.9857433468	0.986500
	200	0.9842	0.007	0.9839672453	0.9844327547	0.985200
	100	0.9879	0.0071	0.9876760351	0.9881239649	0.988100
DLB	1000	0.9879	0.0062	0.9872815500	0.9885184500	0.988000
	900	0.9884	0.0063	0.9879556364	0.9888443636	0.988500
	800	0.9886	0.0064	0.9882314196	0.9889685804	0.989100
	700	0.9911	0.0065	0.9906758125	0.9913241875	0.992500
	600	0.9886	0.0066	0.9883055769	0.9888944231	0.989100
	500	0.9888	0.0067	0.9885271575	0.9890728425	0.9891
	400	0.9841	0.0068	0.9838436267	0.9843563733	0.984500
	300	0.9859	0.0069	0.9856566580	0.9861433420	0.986500
	200	0.984	0.007	0.9837672500	0.9842327500	0.985500
	100	0.9876	0.0071	0.9873760396	0.9878239604	0.988250
Randomized	1000	0.9877	0.0062	0.9870815996	0.9883184004	0.988800
	900	0.9863	0.0063	0.9858556721	0.9867443279	0.987250
	800	0.9903	0.0064	0.9877314491	0.9884685509	0.992500
	700	0.9939	0.0065	0.9899758385	0.9906241615	0.994500
	600	0.9879	0.0066	0.9876056005	0.9881943995	0.988500
	500	0.9881	0.0067	0.9878271793	0.9883728207	0.989100
	400	0.9836	0.0068	0.9833436473	0.9838563527	0.997500
	300	0.9856	0.0069	0.9853566776	0.9858433224	0.988500
	200	0.9839	0.007	0.9836672687	0.9841327313	0.985500
	100	0.9871	0.0071	0.9868760576	0.9873239424	0.988500

TABLE 6.4: Resource Availability in case of cloud computing system for 40 simulations

Strategy	Iteration	Average	Standard deviation	Confidence Interval (95%)	Maximum
LBTA_SSO	1000	0.99999	0.00612	0.999981851 0.999991876	0.9999925
	900	0.999984	0.00613	0.9999848444 0.9999894845	0.99999
	800	0.999963	0.00614	0.9999626174 0.999966652	0.99998
	700	0.999959	0.00625	0.999942242 0.9999594224	0.999975
	600	0.999952	0.00627	0.999952299 0.999962272	0.999965
	500	0.999945	0.00636	0.99994502 0.99994505	0.9999575
	400	0.999949	0.00628	0.999948125 0.99994822	0.999959
	300	0.9999388	0.00639	0.99938894 0.99938904	0.9999475
	200	0.999928	0.0072	0.999928232 0.999928923	0.999935
	100	0.9999189	0.00721	0.9999139829 0.999918947	0.9999205
HBO	1000	0.999977	0.00612	0.999981 0.999961851	0.99997576
	900	0.999965	0.00613	0.999948444 0.999948444	0.999975
	800	0.999968	0.00614	0.999966686 0.9966686	0.999978
	700	0.999931	0.00618	0.9999422422 0.9999545639	0.9999413
	600	0.999926	0.00617	0.9999227286 0.9999272869	0.999936
	500	0.9999881	0.00616	0.999902944 0.999902944	0.999989
	400	0.999939	0.00615	0.9999422422 0.9999422422	0.999955
	300	0.999887	0.00619	0.9998904336 0.9998904336	0.999899
	200	0.99989	0.0072	0.9998923277 0.999893375	0.999899
	100	0.999895	0.00721	0.9998912398 0.9998979829	0.999899
ACO	1000	0.99999	0.00612	0.9996185120 0.9996185120	0.999991
	900	0.999958	0.00613	0.9999535563 0.9999624436	0.999965
	800	0.999932	0.00614	0.9999283141 0.9999356858	0.999935
	700	0.999928	0.00615	0.9999247581 0.9999312418	0.999935
	600	0.99991	0.00617	0.9999082715 0.9999137284	0.999925
	500	0.99991	0.00616	0.9999080559 0.9999139442	0.999921
	400	0.999905	0.00618	0.9999024362 0.9999075637	0.9999125
	300	0.9999	0.00619	0.9998904336 0.9999024334	0.99991
	200	0.999894	0.0072	0.9998916725 0.9998963275	0.999899
	100	0.999879	0.00721	0.9998767603 0.9998812396	0.999888
HLBA	1000	0.99989	0.00612	0.9998838153 0.9998961846	0.999891
	900	0.99989	0.00613	0.9998855562 0.9998944437	0.999891
	800	0.99988	0.00614	0.9998783141 0.9998856858	0.99989
	700	0.99986	0.00615	0.999958060 0.999894194	0.999875
	600	0.99985	0.00616	0.999884055 0.999889944	0.99988
	500	0.999845	0.00617	0.999882271 0.999887728	0.999885
	400	0.999876	0.00615	0.999875806 0.999892419	0.999887
	300	0.999835	0.00619	0.999835256 0.999857433	0.999852
	200	0.999842	0.0072	0.999839672 0.999844327	0.9998944
	100	0.999837	0.00721	0.99983767603 0.9998512	0.999839
DLB	1000	0.99987	0.00612	0.9998728155 0.9998851845	0.9998718
	900	0.999884	0.00613	0.9998795563 0.9998884436	0.9998863
	800	0.999886	0.00614	0.9998823141 0.9998896858	0.999889
	700	0.99991	0.00615	0.9999067581 0.9999132418	0.99993
	600	0.999886	0.00616	0.9998830557 0.9998889442	0.9998898
	500	0.99988	0.00627	0.9998852715 0.9998907284	0.9998987
	400	0.999841	0.00618	0.99983843625 0.9998435637	0.999867
	300	0.99985	0.00619	0.9998565665 0.9998614334	0.99988
	200	0.99984	0.0072	0.9998376725 0.9998423275	0.999868
	100	0.999876	0.00721	0.9998737603 0.9998782396	0.999889
Randomized	1000	0.99987	0.00612	0.9998708159 0.999883184	0.99989899
	900	0.999863	0.00613	0.9998585567 0.999867443	0.9998989
	800	0.9999	0.00614	0.9998773144 0.999884685	0.99997584
	700	0.999939	0.00615	0.9998997583 0.999906241	0.9999888
	600	0.999879	0.00616	0.999876056005 0.9998819439	0.999885649
	500	0.99981	0.00617	0.9998782717 0.9998837282	0.99987589
	400	0.99983	0.00618	0.9998334364 0.99983856352	0.9999328975
	300	0.999856	0.00619	0.9998535667 0.9998584332	0.999895468
	200	0.999839	0.0072	0.999836672687 0.9998413273	0.9998567
	100	0.999871	0.00721	0.9998687605 0.9998732394	0.9998789

TABLE 6.5: Resource availability in case of grid computing system

Time	LBTA_SSO		HBO		ACO		HLBA		DLB		Randomized	
	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM
100	0.999	0.9977	0.9977	0.9977	0.9975	0.9911	0.989	0.9879	0.9879	0.9861	0.9877	0.9857
200	0.9944	0.9977	0.9935	0.9977	0.9928	0.9922	0.9890	0.9863	0.9884	0.9857	0.9863	0.9830
300	0.9963	0.9956	0.9958	0.9955	0.9932	0.9936	0.9882	0.9872	0.9881	0.9868	0.9881	0.9863
400	0.9939	0.994	0.9939	0.9912	0.9928	0.9903	0.9906	0.9836	0.99	0.9829	0.99	0.9822
500	0.99	0.993	0.9881	0.9903	0.988	0.99	0.9879	0.9858	0.9876	0.9857	0.9875	0.9836
600	0.992	0.9925	0.9896	0.9882	0.989	0.988	0.9885	0.9872	0.988	0.9867	0.9878	0.9845
700	0.9952	0.9894	0.9931	0.9884	0.9905	0.9882	0.9839	0.9843	0.9831	0.9839	0.983	0.9836
800	0.9888	0.9898	0.9887	0.9873	0.988	0.9872	0.9855	0.9849	0.985	0.9846	0.9846	0.9842
900	0.989	0.9898	0.989	0.989	0.9884	0.9884	0.9842	0.9863	0.984	0.986	0.9839	0.9857
1000	0.9899	0.9898	0.9895	0.9895	0.9879	0.989	0.9879	0.9868	0.9876	0.9863	0.9871	0.9854

TABLE 6.6: Resource availability in case of cloud computing system

Time	LBTA_SSO		HBO		ACO		HLBA		DLB		Randomized	
	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM
100	0.99999	0.99997	0.99997	0.99997	0.99996	0.99991	0.99989	0.99987	0.999875	0.99986	0.99987	0.99985
200	0.999944	0.999977	0.999935	0.999977	0.99995	0.9999	0.99985	0.999863	0.999874	0.999857	0.999863	0.99983
300	0.999943	0.999956	0.99996	0.999955	0.999932	0.9999	0.999848	0.999842	0.99987	0.999857	0.99979	0.999863
400	0.999939	0.99994	0.999939	0.999912	0.999928	0.99989	0.999836	0.999836	0.99986	0.999829	0.99978	0.999822
500	0.9999	0.99993	0.999881	0.9998	0.99981	0.999872	0.999825	0.99983	0.99986	0.99957	0.999818	0.99983
600	0.9999	0.99992	0.99989	0.99988	0.99981	0.999865	0.999785	0.99982	0.99985	0.999806	0.99958	0.999825
700	0.9999	0.999894	0.999831	0.999884	0.999805	0.99985	0.99978	0.99982	0.99975	0.999803	0.99948	0.999813
800	0.999888	0.999898	0.999887	0.999873	0.9998	0.99985	0.999755	0.999819	0.99975	0.9998	0.99938	0.999802
900	0.99988	0.999898	0.99989	0.99989	0.999784	0.99984	0.999742	0.999813	0.999735	0.9998	0.99935	0.9998
1000	0.99987	0.999898	0.99989	0.999895	0.99975	0.9998	0.999739	0.999808	0.9997	0.9998	0.99925	0.9998

FIGURE 6.5 we have the comparative analysis of resource availability simulated in cloud computing based scenario. We see that our proposed algorithm outperforms the other compared algorithms in both of the environments. We also see the comparative analysis of the resource availability when all the algorithms are simulated in grid computing as well as cloud computing based scenario without transaction management where **FIGURE 6.6** is for grid computing and **FIGURE 6.7** is for cloud computing.

We also compared the performance of grid and cloud computing using our proposed algorithm. Here we see the resource availability comparison using our algorithm between grid and cloud. We conclude that the resource availability is much better in cloud environment than grid environment either with transaction management (shown in **FIGURE 6.8**) or without transaction management (shown in **FIGURE 6.9**).

The LBTA_SSO performs much better in the cloud computing environment as compared to grid computing environment due to two main differences, namely configurability (homogeneity) and quality-of-service. Cloud computing generally have a configurable environment in terms of operating system. It offers a homogeneous resource pool. While, grid computing offers heterogeneous resource pool volunteerly. Cloud computing provides higher quality-of-service than grid computing. Because resources in cloud

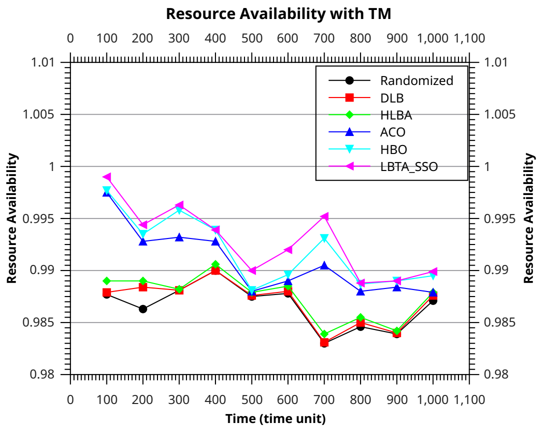

FIGURE 6.4: Resource Availability when TM is used in grid computing based environment

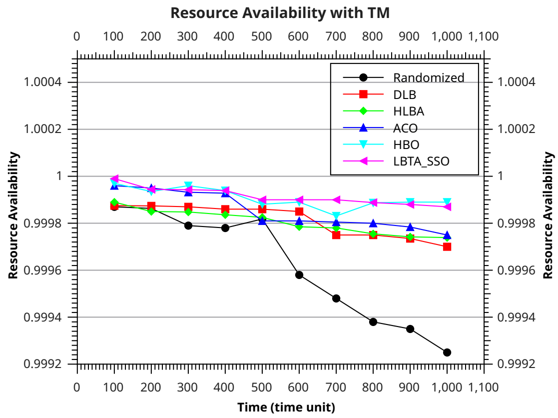

FIGURE 6.5: Resource Availability when TM is used in cloud computing based environment

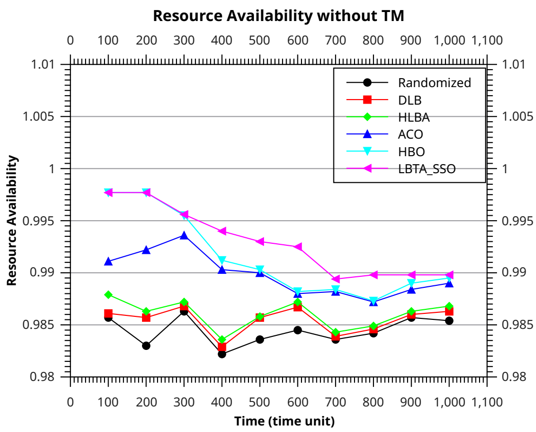

FIGURE 6.6: Resource Availability when no TM is used in grid computing based environment

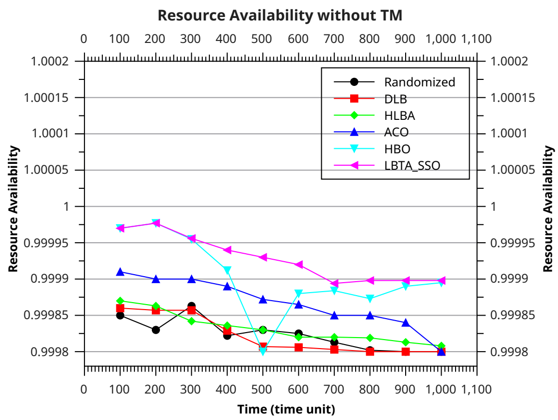

FIGURE 6.7: Resource Availability when no TM is used in cloud computing based environment

computing are dedicated and even there is no risk of preemption also. Hence, the resource availability is higher in cloud computing environment than grid computing.

6.5.1.2 Reliability

The reliability of on-demand computing based transaction processing system considering the conditional steady-state user-perceived availability of resources is computed using

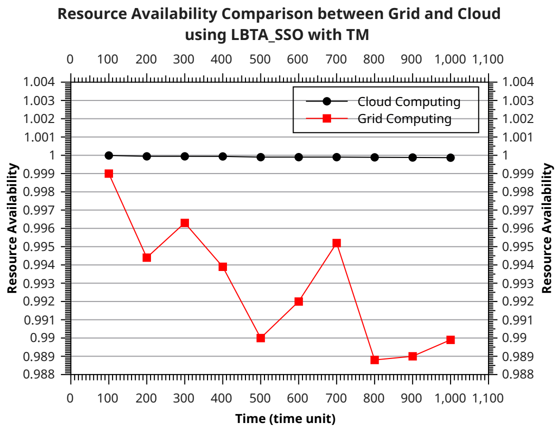

FIGURE 6.8: Resource availability comparison between grid and cloud computing system when transaction processing is used

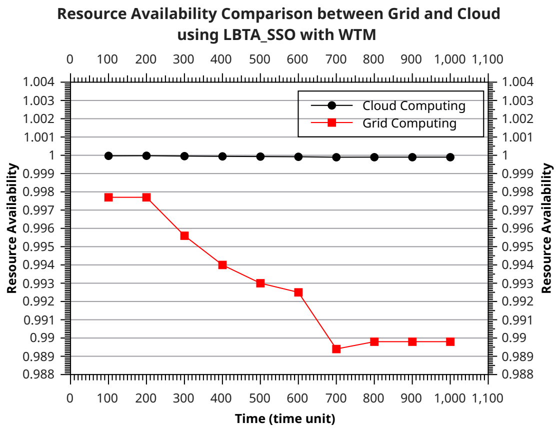

FIGURE 6.9: Resource availability comparison between grid and cloud computing system when no transaction processing is used

Eq. (5), **Eq.** (6) and **Eq.** (9). Since the reliability in this chapter is based on resource availability of the system, we needed the result of the resource availability to find the reliability of the system.

We have the comparative results of reliability using the mentioned algorithms with *p*-values given in **TABLE 6.2** as shown in **TABLE 6.7** and **6.9**. In a similar way, we have the comparative results of reliability using the mentioned algorithms with *p*-values given in **TABLE 6.2** as shown in **TABLE 6.8** and **6.10**.

In **TABLE 6.7**, the reliability of all the methods is based on the results given in **TABLE 6.3**. In a similar way, reliability in cloud computing environment shown in **TABLE 6.8** is based on **TABLE 6.4**. All the results are calculated by the populations with the associated standard deviation and 95% confidence interval and the best result (Maximum).

Firstly, we compared the performance of our proposed algorithm simulated with transaction management and without transaction management. We have the comparisons in both grid (as shown in **FIGURE 6.10**) and cloud (as shown in **FIGURE 6.11**) based environment. In cloud computing environment the reliability is approximately same for the tasks having transaction management and tasks having no transaction management. But in grid computing environment, the reliability is better for the tasks with transaction management as compared with the tasks without transaction management.

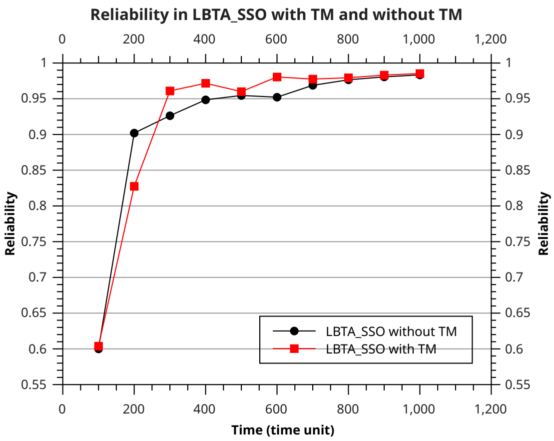

FIGURE 6.10: Reliability in LBTA_SSO with TM and without TM in grid computing based environment

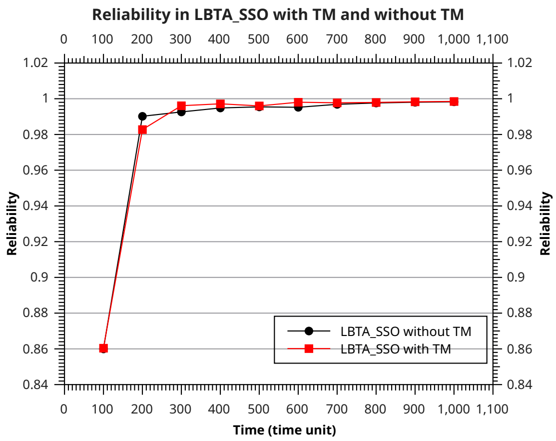

FIGURE 6.11: Reliability in LBTA_SSO with TM and without TM in cloud computing based environment

Then we compared the reliability of our proposed algorithm with other mentioned algorithms. Here we compare the reliability when we use the transaction management in the simulation where **FIGURE 6.12** shows the comparative analysis of reliability when all the algorithms are simulated in grid computing based scenario. The results show that the proposed algorithm LBTA_SSO outperforms the other algorithms. **FIGURE 6.13** shows the comparative analysis of reliability simulated in cloud computing based scenario. We see that our proposed algorithm outperforms the other compared algorithms in both of the environments. We also have the comparative analysis of the reliability when all the algorithms are simulated in grid computing as well as cloud computing based scenario without transaction management where **FIGURE 6.14** is for grid computing and **FIGURE 6.15** is for cloud computing.

We also compared the performance of grid and cloud computing using our proposed algorithm. Here we have the reliability comparison using our algorithm between grid and cloud. We conclude that the reliability is much better in cloud environment than grid environment either with transaction management (shown in **FIGURE 6.16**) or without transaction management (shown in **FIGURE 6.17**). The maximization of reliability is achieved because of the maximization in the steady-state user-perceived availability of resources as depicted in **Eq.** (9).

TABLE 6.7: Reliability when grid computing is used for 40 simulations

Strategy	Iteration	Average	Standard deviation	Confidence Interval (95%)		Maximum
LBTA_SSO	1000	0.9853	0.063	0.9851011750	0.9854988250	0.98506491
	900	0.9831	0.00629	0.9828907527	0.9833092473	0.98548
	800	0.9794	0.00628	0.9791784125	0.9796215875	0.97999
	700	0.9775	0.00627	0.9772634902	0.9777365098	0.9779867
	600	0.9805	0.00626	0.9802449477	0.9807550523	0.980993
	500	0.96	0.00625	0.9597210505	0.9602789495	0.96898
	400	0.9717	0.00624	0.971388624	0.972011376	0.97199
	300	0.961	0.00623	0.9606410302	0.9613589698	0.96121
	200	0.8275	0.00622	0.8270610592	0.8279389408	0.82999
	100	0.6038	0.00621	0.6031802420	0.6044197580	0.603989
HBO	1000	0.973	0.063	0.9728012546	0.97319	0.9719889
	900	0.969	0.00629	0.9687908365	0.9692091635	0.9708948
	800	0.96	0.00628	0.9597785013	0.9602214987	0.9689895
	700	0.97	0.00627	0.969763585	0.970236415	0.979879
	600	0.95	0.00626	0.9497450499	0.9502549501	0.95899553
	500	0.977	0.00625	0.9767211623	0.9772788377	0.97798989
	400	0.949	0.00624	0.9486887488	0.9493112512	0.9498989
	300	0.953	0.00623	0.9526411740	0.9533588260	0.954975
	200	0.865	0.00622	0.8645612352	0.8654387648	0.86589
	100	0.667	0.00621	0.6663804904	0.6676195096	0.66895
ACO	1000	0.955	0.063	0.9548012746	0.9551987254	0.958975
	900	0.95	0.00629	0.9497908575	0.9502091425	0.9575
	800	0.94	0.00628	0.9397785235	0.9402214765	0.955
	700	0.935	0.00627	0.9347636087	0.9352363913	0.939898
	600	0.927	0.00626	0.9267450755	0.9272549245	0.92953
	500	0.915	0.00625	0.9147211903	0.9152788097	0.916789
	400	0.895	0.00624	0.89468878	0.89531122	0.8965
	300	0.775	0.00623	0.77464121	0.77535879	0.77689
	200	0.77	0.00622	0.7695612791	0.7704387209	0.7725
	100	0.77	0.00621	0.7695612791	0.7704387209	0.77595
HLBA	1000	0.9892	0.063	0.9890012706	0.9893987294	0.9896
	900	0.9838	0.00629	0.9835908533	0.9840091467	0.9858
	800	0.9854	0.00628	0.9851785191	0.9856214809	0.98599
	700	0.9781	0.00627	0.977863604	0.978336396	0.97908
	600	0.9796	0.00626	0.9793450704	0.9798549296	0.97999
	500	0.975	0.00625	0.9747211847	0.9752788153	0.9758
	400	0.9651	0.00624	0.9647887738	0.9654112262	0.96525
	300	0.9426	0.00623	0.9422412028	0.9429587972	0.94271
	200	0.923	0.00622	0.9225612703	0.9234387297	0.9245
	100	0.6436	0.00621	0.6429805401	0.6442194599	0.64389
DLB	1000	0.9765	0.063	0.9763012746	0.9766987254	0.9775
	900	0.9704	0.00629	0.9701908575	0.9706091425	0.9712
	800	0.955	0.00628	0.9547785235	0.9552214765	0.9565
	700	0.9605	0.00627	0.9602636087	0.9607363913	0.9625
	600	0.955	0.00626	0.9547450755	0.9552549245	0.9565
	500	0.9228	0.00625	0.9225211903	0.9230788097	0.9235
	400	0.9105	0.00624	0.91018878	0.91081122	0.9125
	300	0.864	0.00623	0.86364121	0.86435879	0.8675
	200	0.65	0.00622	0.649562379	0.650437621	0.65623
	100	0.61	0.00621	0.609382105	0.610617895	0.6237
Randomized	1000	0.9702	0.063	0.9700012746	0.9703987254	0.9725
	900	0.9624	0.00629	0.9622908575	0.9627091425	0.9635
	800	0.9482	0.00628	0.9479785235	.9484214765	0.9489
	700	0.9503	0.00627	0.9500636087	0.9505363913	0.9525
	600	0.9432	0.00626	0.9429450755	0.9434549245	0.9445
	500	0.92	0.00625	0.9197211903	0.9202788097	0.9225
	400	0.889	0.00624	0.88868878	0.88931122	0.8899
	300	0.833	0.00623	0.83264121	0.83335879	0.835
	200	0.63	0.00622	0.629562379	0.630437621	0.645
	100	0.61	0.00621	0.60938111	0.61061889	0.625

TABLE 6.8: Reliability when cloud computing is used for 40 simulations

Strategy	Iteration	Average	Standard deviation	Confidence Interval (95%)	Maximum
LBTA_SSO	1000	0.998853	0.0613	0.9988510117 0.9988549882	0.9889
	900	0.998831	0.00629	0.998828907527 0.9988330924	0.9989
	800	0.998794	0.00628	0.9987917841 0.99879621587	0.9988
	700	0.99877	0.00627	0.998772634902 0.9987773650	0.998798
	600	0.9988	0.00626	0.9988024494 0.9988075505	0.99889
	500	0.9986	0.00625	0.9985972105 0.9986027894	0.9987
	400	0.998717	0.00624	0.998713886 0.998720113	0.99875
	300	0.99861	0.00623	0.9986064103 0.9986135896	0.99865
	200	0.98275	0.00622	0.98270610592 0.98279389408	0.98278
	100	0.86938	0.00621	0.86931802420 0.86944197580	0.86945
HBO	1000	0.99873	0.063	0.9987280125 0.9987319874	0.99875
	900	0.99869	0.00629	0.9986879083 0.9986920916	0.99875
	800	0.9986	0.00628	0.998597785 0.998602214	0.9988
	700	0.9987	0.00627	0.998697635 0.998702364	0.9989
	600	0.9985	0.00626	0.9984974504 0.9985025495	0.99895
	500	0.99877	0.00625	0.9987672116 0.9987727883	0.99889
	400	0.99849	0.00624	0.9984868874 0.9984931125	0.998579
	300	0.99853	0.00623	0.9985264117 0.9985335882	0.99855
	200	0.9865	0.00622	0.98645612352 0.8654387648	0.9875
	100	0.8667	0.00621	0.86663804904 0.86676195096	0.8695
ACO	1000	0.99855	0.063	0.998548012746 0.998551987254	0.99857
	900	0.9985	0.00629	0.9984979085 0.9985020914	0.99879
	800	0.9984	0.00628	0.9983977852 0.9984022147	0.99855
	700	0.99835	0.00627	0.998347636 0.9983523639	0.998498
	600	0.99727	0.00626	0.9792674507 0.9972725492	0.997659
	500	0.99715	0.00625	0.9971472119 0.997152788	0.997356
	400	0.9895	0.00624	0.989468878 0.989531122	0.98979
	300	0.8775	0.00623	0.877464121 0.877535879	0.87759
	200	0.877	0.00622	0.87695612791 0.87704387209	0.8771
	100	0.875	0.00621	0.873805525 0.8996194475	0.87545
HLBA	1000	0.997892	0.063	0.9978900127 0.9978939872	0.99789009
	900	0.997838	0.00629	0.9978359085 0.9978400914	0.99798
	800	0.997854	0.00628	0.9978517851 0.9978562148	0.99799
	700	0.997781	0.00627	0.997778636 0.997783363	0.99789
	600	0.997796	0.00626	0.9977934507 0.9977985492	0.997799
	500	0.99775	0.00625	0.9977472118 0.9977527881	0.997789
	400	0.997651	0.00624	0.9976478877 0.9976541122	0.9976993
	300	0.997426	0.00623	0.9974224125 0.9974295879	0.99745
	200	0.9923	0.00622	0.99225612703 0.99234387297	0.992424
	100	0.86436	0.00621	0.86429805401 0.86442194599	0.8644589
DLB	1000	0.9957	0.063	0.9957630127 0.9957669872	0.99575099
	900	0.9957	0.00629	0.9957019085 0.9957060914	0.995709
	800	0.9955	0.00628	0.9955477852 0.9955522147	0.99553
	700	0.9956	0.00627	0.9956026365 0.9956073639	0.995698
	600	0.99555	0.00626	0.9955474507 0.9955525492	0.995559
	500	0.994228	0.00625	0.9942252119 0.9942307885	0.9942345
	400	0.9941	0.00624	0.99410188 0.99410811	0.99493
	300	0.9864	0.00623	0.986364121 0.986435879	0.9867891
	200	0.865	0.00622	0.8649562379 0.8650437621	0.86562388
	100	0.861	0.00621	0.8609382105 0.8610617895	0.861378
Randomized	1000	0.9927	0.063	0.9927000127 0.9927039872	0.992709
	900	0.99262	0.00629	0.9926229085 0.9926270914	0.9926289
	800	0.992482	0.00628	0.992479785 .992484214	0.9924952
	700	0.992503	0.00627	0.992500636 0.9925053639	0.9925498
	600	0.992432	0.00626	0.9924294507 0.9924345492	0.992436
	500	0.9912	0.00625	0.9911972119 0.9912027885	0.99219
	400	0.9889	0.00624	0.988868878 0.988931122	0.98897
	300	0.9833	0.00623	0.983264121 0.983335879	0.983478
	200	0.863	0.00622	0.8629562379 0.8630437621	0.86375
	100	0.861	0.00621	0.860938111 0.861061889	0.862491

TABLE 6.9: Reliability in grid computing

Time	LBTA_SSO		HBO		ACO		HLBA		DLB		Randomized	
	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM
100	0.6038	0.6	0.667	0.667	0.66	0.567	0.6436	0.6153	0.61	0.619	0.61	0.62
200	0.8275	0.902	0.865	0.926	0.77	0.562	0.923	0.913	0.65	0.62	0.63	0.639
300	0.961	0.9263	0.953	0.93	0.775	0.823	0.9426	0.9456	0.864	0.816	0.833	0.778
400	0.9717	0.9485	0.949	0.893	0.895	0.874	0.9651	0.9551	0.9105	0.8641	0.889	0.833
500	0.96	0.9546	0.977	0.94	0.915	0.900	0.975	0.9691	0.9228	0.9097	0.92	0.886
600	0.9805	0.9522	0.95	0.942	0.927	0.932	0.9796	0.9782	0.955	0.9507	0.9432	0.938
700	0.9775	0.9688	0.97	0.943	0.935	0.93	0.9781	0.9818	0.9605	0.9648	0.9503	0.9557
800	0.9794	0.9765	0.96	0.954	0.94	0.937	0.9854	0.9831	0.955	0.9697	0.9482	0.962
900	0.9831	0.9806	0.969	0.964	0.95	0.94	0.9838	0.9872	0.9704	0.9746	0.9625	0.9678
1000	0.9853	0.9834	0.973	0.964	0.955	0.954	0.9892	0.9875	0.9765	0.9784	0.9702	0.9727

TABLE 6.10: Reliability in cloud computing

Time	LBTA_SSO		HBO		ACO		HLBA		DLB		Randomized	
	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM
100	0.86038	0.86	0.8667	0.8667	0.8999	0.82	0.86436	0.86153	0.861	0.8619	0.861	0.862
200	0.98275	0.9902	0.865	0.9926	0.977	0.9562	0.9923	0.9913	0.865	0.862	0.863	0.8639
300	0.9961	0.99263	0.9953	0.993	0.9775	0.9823	0.99426	0.99456	0.9864	0.9816	0.9833	0.9778
400	0.99717	0.99485	0.9949	0.9893	0.9895	0.9874	0.99651	0.99551	0.99105	0.98641	0.9889	0.9833
500	0.996	0.99546	0.9977	0.994	0.9915	0.9900	0.9975	0.99691	0.99228	0.99097	0.992	0.9886
600	0.99805	0.99522	0.995	0.9942	0.9927	0.9932	0.99796	0.99782	0.9955	0.99507	0.99432	0.9938
700	0.99775	0.99688	0.997	0.9943	0.9935	0.993	0.99781	0.99818	0.99605	0.99648	0.99503	0.99557
800	0.99794	0.99765	0.996	0.9954	0.994	0.9937	0.99854	0.99831	0.9955	0.99697	0.99482	0.9962
900	0.99831	0.99806	0.9969	0.9964	0.995	0.94	0.99838	0.99872	0.99704	0.99746	0.99625	0.99678
1000	0.9985	0.9983	0.9973	0.9964	0.9955	0.9954	0.9989	0.9987	0.9976	0.99784	0.997	0.997

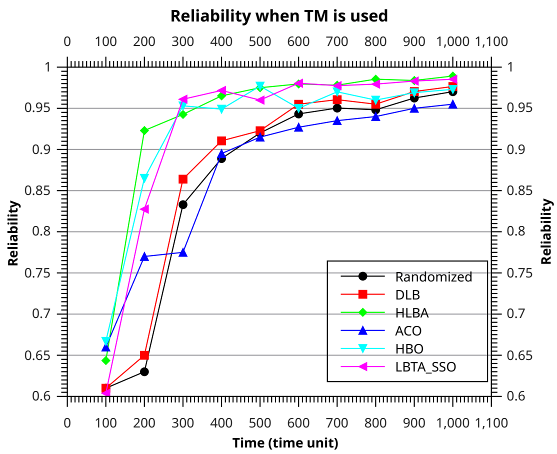

FIGURE 6.12: Reliability when TM is used in grid computing based environment

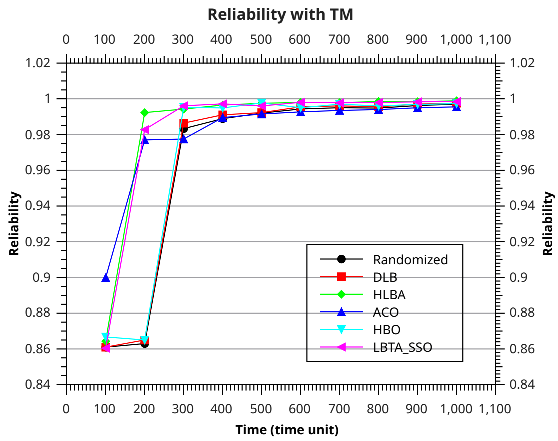

FIGURE 6.13: Reliability when TM is used in cloud computing based environment

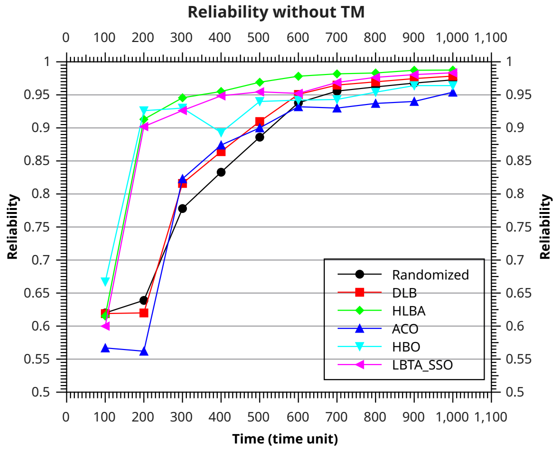

FIGURE 6.14: Reliability when no TM is used in grid computing based environment

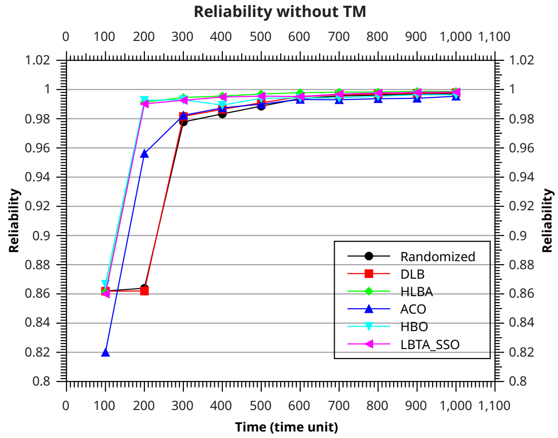

FIGURE 6.15: Reliability when no TM is used in cloud computing based environment

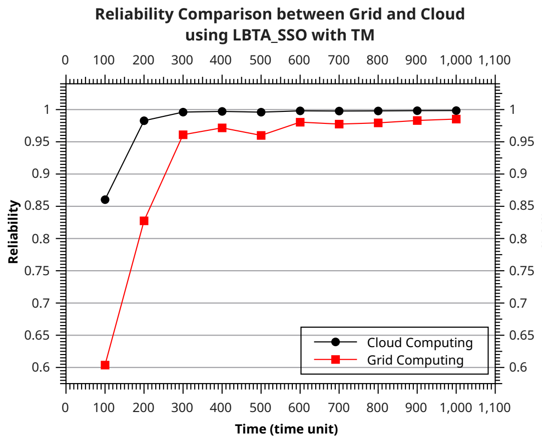

FIGURE 6.16: Reliability comparison between grid and cloud computing system when transaction processing is used

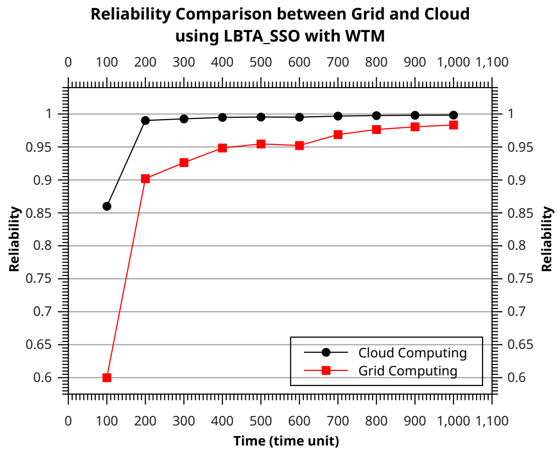

FIGURE 6.17: Reliability comparison between grid and cloud computing system when no transaction processing is used

6.5.1.3 Miss Ratio

Miss Ratio is one of the important performance parameters for a transaction processing system. For better performance of the system, the miss ratio should be minimum. This section illustrates the comparative analysis of miss ratio of transactions when the mentioned algorithms are applied for a transaction in grid computing system as well as in a computational grid without transaction processing.

Miss Ratio [7] can be calculated as

$$Miss\ Ratio = \frac{T_{miss}}{T_{total}} * 100\% \tag{22}$$

where T_{miss} is the number of transactions missing the deadlines and T_{total} is the total number of handled transactions.

We have the comparative results of the miss ratio of transactions using the mentioned algorithms with p-values (from **TABLE 6.2**) in **TABLE 6.11** and **6.13**. In a similar way, we have the comparative results of miss ratio of transactions using the mentioned algorithms with p-values (from **TABLE 6.2**) in **TABLE 6.12** and **6.14**. All the results are calculated by the populations with the associated standard deviation and 95% confidence interval and the best result (Minimum).

We compared the performance of our proposed algorithm simulated with transaction management and without transaction management. We have the comparisons in both grid (as shown in **FIGURE 6.18**) and cloud (as shown in **FIGURE 6.19**) based environment. In both grid and cloud computing environment, the miss ratio is approximately same for the tasks having transaction management and tasks having no transaction management.

Then we compared the miss ratio of transactions in our proposed algorithm with other mentioned algorithms. Here we see that the miss ratio when we used the transaction management in the simulation where **FIGURE 6.20** shows the comparative analysis of miss ratio when all the algorithms are simulated in grid computing based scenario. The result shows that the proposed algorithm LBTA_SSO outperforms the other algorithms. **FIGURE 6.21** shows the comparative analysis of miss ratio simulated in cloud computing based scenario. We see that our proposed algorithm outperforms the other compared algorithms in both of the environments. We also see that the comparative analysis of the miss ratio when all the algorithms are simulated in grid computing as well as cloud computing based scenario without transaction management where **FIGURE 6.22** is for grid computing and **FIGURE 6.23** is for cloud computing.

We also compared the miss ratio in grid and cloud computing using our proposed algorithm. Here we have the miss ratio comparison using our algorithm between grid and cloud. Here we conclude that the miss ratio is minimum in cloud environment than grid

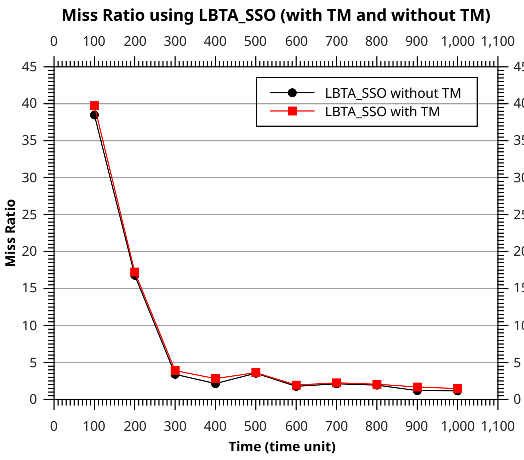

FIGURE 6.18: Miss Ratio in LBTA_SSO with and without TM in grid computing based environment

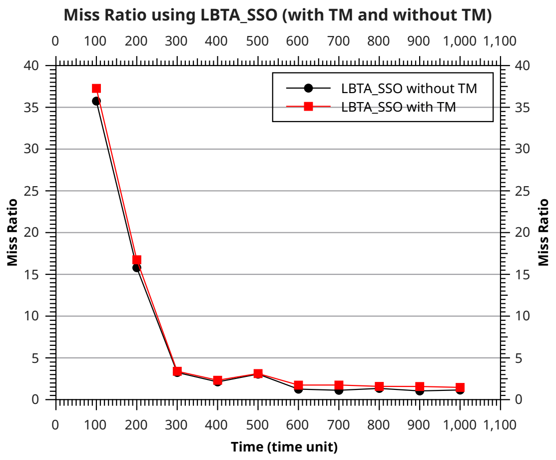

FIGURE 6.19: Miss Ratio in LBTA_SSO with and without TM in cloud computing based environment

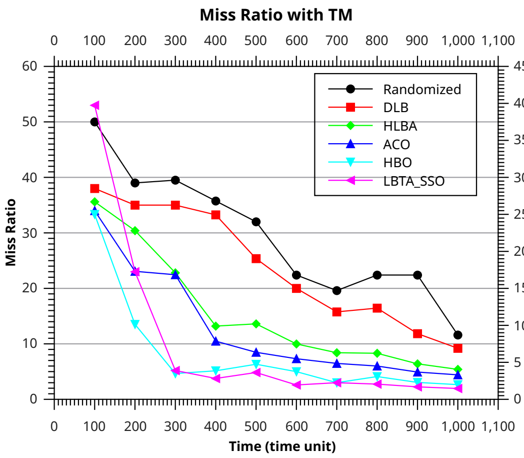

FIGURE 6.20: Miss Ratio when TM is used in grid computing based environment

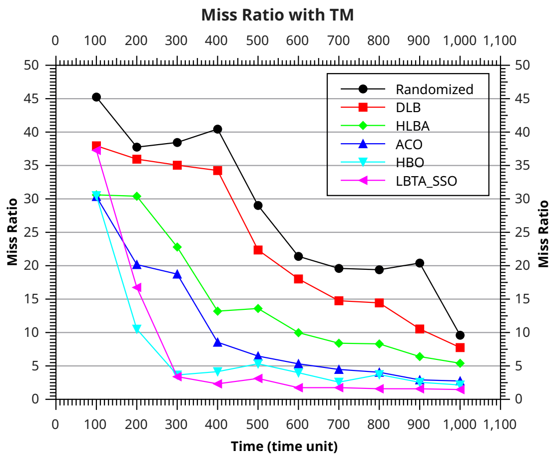

FIGURE 6.21: Miss Ratio when TM is used in cloud computing based environment

environment either with transaction management (shown in **FIGURE 6.24**) or without transaction management (shown in **FIGURE 6.25**).

The improvement in results is seen because of the balanced task allocation approach of the algorithm. When the nodes are balanced before assigning the transactions to them, the chances of transaction commit is increased. Because waiting time at each node is minimized. Thus, the miss ratio is also minimized.

TABLE 6.11: Miss Ratio in grid computing for 40 simulations

Strategy	Iteration	Average	Standard deviation	Confidence Interval (95%)		Minimum
LBTA_SSO	1000	1.4647	0.33	1.4542853548	1.4751146452	1.451275558
	900	1.6853	0.322	1.6746546667	1.6959453333	1.673535567
	800	2.0568	0.321	2.0458617652	2.0677382348	2.034759521
	700	2.2488	0.32	2.2374837437	2.2601162563	2.1263726326
	600	1.943	0.292	1.931184478	1.954815522	1.89893376
	500	3.6261	0.282	3.6136030633	3.6385969367	3.59950522
	400	2.8205	0.275	2.807027	2.833973	2.706016
	300	3.8955	0.265	3.8805189152	3.9104810848	3.7704088142
	200	17.241	0.255	17.2233576858	17.2586423142	17.1122465747
	100	39.75	0.247	39.726048	39.773952	39.615937
HBO	1000	2.65	0.333	2.639589529	2.6604104710	2.528478418
	900	3.05	0.322	3.0393589333	3.0606410667	2.989898989
	800	4.104	0.331	4.0930661493	4.1149338507	4.0820550382
	700	3	0.33	2.9886882793	3.0113117207	2.878787893
	600	4.987	0.292	4.9751892137	4.9988107863	4.8650781026
	500	6.323	0.283	6.3105080721	6.3354919279	6.200407061
	400	5.14	0.275	5.1265324	5.1534676000	5.015412344
	300	4.65	0.267	4.6350249197	4.6649750803	4.5249138086
	200	13.51	0.257	13.4923647569	13.5276352431	12.897654239
	100	33.33	0.248	33.3060576	33.3539424	33.01234786
ACO	1000	4.414	0.323	4.4244094275	4.4244094275	4.39867453
	900	4.914	0.322	4.92464	4.98464	4.89897654
	800	6	0.312	6.0109327547	6.2309327547	5.9898987677
	700	6.472	0.33	6.4833105869	6.5633105869	6.3722098764
	600	7.317	0.292	7.3288096024	7.5677123578	7.023458976
	500	8.475	0.283	8.4874906757	8.58985817868	8.0786756429
	400	10.484	0.275	10.49746625	11.0897654321	10.03987654
	300	22.454	0.262	22.4689735792	22.8676876545	21.346578692
	200	23.07	0.253	23.0876334754	23.9889765424	22.8989876544
	100	34	0.234	34.02394	35.10239	34.002395
HLBA	1000	5.4	0.313	5.3895903638	5.4104096362	5.2784802517
	900	6.4	0.312	6.3893597867	6.4106402133	6.2756437897
	800	8.3	0.312	8.2890670261	8.3109329739	8.1780560124
	700	8.4	0.321	8.3886891864	8.4113108136	8.1567876544
	600	9.98	0.291	9.9681901608	9.9918098392	9.8657898756
	500	13.6	0.282	13.5875090738	13.6124909262	13.443567865
	400	13.2	0.271	13.18653348	13.21346652	13.056436723
	300	22.8	0.262	22.7850261205	22.8149738795	22.089786756
	200	30.4	0.252	30.3823661711	30.4176338289	30.12456876
	100	35.6	0.242	35.57605952	35.62394048	35.045678543
DLB	1000	9.2	0.33	9.1895905725	9.2104094275	9.087654325
	900	11.84	0.312	11.82936	11.85064	11.82936
	800	16.44	0.312	16.4290672453	16.4509327547	16.32432311
	700	15.76	0.32	15.7486894131	15.7713105869	15.087674868
	600	20	0.292	19.9881903976	20.0118096024	19.12389765
	500	25.36	0.282	25.3475093243	25.3724906757	24.898987645
	400	36.2	0.227	36.18653375	36.21346625	35.897656432
	300	53	0.226	52.9850264208	53.0149735792	52.120087658
	200	35	0.225	34.9823665246	35.0176334754	34.0564328976
	100	38	0.224	37.97606	38.02394	37.856476
Randomized	1000	11.6	0.33	11.5895905725	11.6104094275	11.05623455
	900	22.4	0.322	22.38936	22.41064	22.0005676
	800	22.4	0.313	22.3890672453	22.4109327547	21.8338906724
	700	19.6	0.332	19.5886894131	19.6113105869	19.058868941
	600	22.4	0.292	22.3881903976	22.4118096024	22.0538819039
	500	32	0.282	31.9875093243	32.0124906757	31.0098750932
	400	44.4	0.275	44.38653375	44.41346625	44.00386533
	300	65	0.263	64.9850264208	65.0149735792	64.0098502642
	200	75.07	0.225	75.9823665246	75.0176334754	75.0098236652
	100	79	0.234	79.97606	80.02394	79.000976

TABLE 6.12: Miss Ratio in cloud computing system for 40 simulations

Strategy	Iteration	Average	Standard deviation	Confidence Interval (95%)	Minimum
LBTA_SSO	1000	1.4647	0.33	1.4542853548 1.4751146452	1.451275558
	900	1.5853	0.32	1.574654666 1.5959453332	1.573535565
	800	1.568	0.31	1.4586176525 1.677382348	1.347595215
	700	1.7488	0.3	1.74837437 1.7601162563	1.73726326
	600	1.743	0.29	1.731184478 1.754815522	1.729893376
	500	3.1261	0.28	3.136030633 3.385969367	3.19950522
	400	2.3205	0.27	2.307027 2.339735	2.306016
	300	3.3955	0.26	3.3805189152 3.4104810848	3.2704088142
	200	16.741	0.25	16.723357685 16.758642314	16.722465747
	100	37.25	0.24	37.226048 37.273952	37.115937
HBO	1000	2.15	0.33	2.139589525 2.1604104715	2.152847842
	900	2.55	0.32	2.539358934 2.5606410665	2.358989895
	800	3.704	0.31	3.7043066149 3.8149338507	3.7042055038
	700	2.575	0.3	2.4988688279 2.551131172	2.5748787893
	600	3.987	0.29	3.9751892137 3.9988107863	3.8650781026
	500	5.325	0.28	5.305080725 5.354919279	5.200407065
	400	4.14	0.27	4.1265324 4.153467605	4.015412375
	300	3.65	0.26	3.6350249175 3.7664975085	3.5249138075
	200	10.51	0.25	10.4923647575 10.5276352445	10.897654255
	100	30.53	0.24	30.53060576 30.63539424	30.401234786
ACO	1000	2.7415	0.33	2.74244094275 2.758987675	2.639867453
	900	2.914	0.32	2.92464 2.98464	2.89897654
	800	4.05	0.31	4.010932755 4.230932755	4.0489898765
	700	4.472	0.3	4.483310585 4.563310587	4.372209877
	600	5.317	0.29	5.328809623 5.567712358	5.02345898
	500	6.475	0.28	6.487490676 6.5898581787	6.07867564
	400	8.545	0.27	8.49746675 9.0897654321	8.0398766
	300	18.745	0.26	18.546897358 18.867687655	18.465786925
	200	20.07	0.25	20.187633485 20.988976557	20.189898765
	100	30.75	0.24	30.82394 30.910239	30.7002395
HLBA	1000	3.54	0.33	3.389590375 3.5410409637	3.5784802517
	900	4.45	0.32	4.3893597875 4.410640245	4.27564378974
	800	6.3	0.31	6.289067027 6.310932974	6.1780561
	700	6.4	0.3	6.388689185 6.4113108137	6.15678766
	600	7.98	0.29	7.968190175 7.991809845	7.865789876
	500	10.6	0.28	10.587509085 10.612490927	10.443567875
	400	11.2	0.27	11.18653348 11.21346652	11.056436723
	300	18.8	0.26	18.7850261205 18.8149738795	18.089786756
	200	26.54	0.25	26.3823661711 26.54176338289	26.12456876
	100	30.6	0.24	30.57605952 30.62394048	30.045678543
DLB	1000	7.752	0.33	7.7518959057 7.7521040942	7.747654325
	900	10.54	0.32	10.52936 10.585064	10.582936
	800	14.44	0.31	14.4290672325 14.4509327456	14.32432375
	700	14.76	0.3	14.7486894131 14.7713105869	14.087674868
	600	18.025	0.29	18.019881903 18.0275096024	18.012389765
	500	22.36	0.28	22.3475093243 22.3724906757	22.898987645
	400	34.25	0.27	34.25186533 34.72134662	34.1897656435
	300	35.05	0.26	35.009850264 35.01497357925	35.012008765
	200	35.95	0.25	35.4982366524 35.9501763347	35.564328975
	100	37.95	0.24	37.97606 38.023944	37.0564765
Randomized	1000	9.6	0.33	9.758959057 9.7610409427	9.75623455
	900	20.4	0.32	20.38936 20.41064	20.0005676
	800	19.4	0.31	19.3890672453 19.510932756	19.758338906
	700	19.6	0.3	19.5886894131 19.6113105869	19.058868941
	600	21.4	0.29	21.3881903976 21.4118096024	21.0538819039
	500	29.025	0.28	29.02579875 29.71249067	29.0098750932
	400	40.45	0.27	40.38756533 40.41375466	40.00375865
	300	60.95	0.26	60.8502642085 61.0149735792	60.9850264275
	200	69.07	0.25	68.9823665246 69.0176334754	68.0098236652
	100	75	0.24	75.7597606 75.02394	75.0075976

TABLE 6.13: Miss Ratio (%) in grid computing

Time	LBTA_SSO		HBO		ACO		HLBA		DLB		Randomized	
	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM
100	39.75	38.5	33.33	33.125	34	32.45	35.6	33.3	38	35.75	50	50
200	17.241	16.8039	13.51	12.407	23.07	22.75	30.4	29.6	35	34.5	39	37.75
300	3.8955	3.3985	4.65	4.05	22.454	17.645	22.8	21.6	35	34	39.5	36
400	2.8205	2.147	5.140	4.738	10.484	11.631	13.2	12.6	33.25	33.75	33.75	33.75
500	3.6261	3.5442	6.323	5.99	8.475	7.937	13.6	11.6	25.36	24.8	32	31.6
600	1.943	1.7715	4.987	4.76	7.317	6.785	9.98	8.6	20	19.75	22.4	21.4
700	2.2488	2.1192	3.00	2.66	6.472	5.963	8.4	7.5	15.76	14.4	19.6	17.6
800	2.0568	1.934	4.104	3.64	6.00	5.340	8.30	7.50	16.44	12.12	22.4	15.2
900	1.6853	1.1941	3.05	2.632	4.914	4.149	6.4	5.74	11.84	10.6	22.4	21.8
1000	1.4647	1.1655	2.65	2.1728	4.414	3.608	5.4	4.6	9.2	8.6	11.6	10.8

TABLE 6.14: Miss Ratio (%) in cloud computing

Time	LBTA_SSO		HBO		ACO		HLBA		DLB		Randomized	
	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM	TM	WTM
100	37.275	35.75	30.5	30.05	30.35	30	30.6	30.3	37.95	38.25	45.25	45.2
200	16.75	15.8039	10.51	7.407	20.17	18.75	30.4	28.6	35.95	34.5	37.75	36.5
300	3.3955	3.25	3.65	3.05	18.745	17.645	22.8	21.6	35.05	34.25	38.45	37.5
400	2.3205	2.125	4.145	4.05	8.545	8.26	13.2	12.6	34.25	33.25	40.45	38.575
500	3.1261	3.054	5.325	5.199	6.475	5.937	13.6	11.6	22.36	21.8	29.025	28.6
600	1.743	1.257	3.985	3.76	5.317	5.1785	9.98	9.16	18.025	17.75	21.4	20.4
700	1.7488	1.1192	2.575	1.66	4.472	3.963	8.4	7.5	14.76	13.4	19.6	17.6
800	1.5853	1.3437	3.704	3.64	4.05	3.340	8.30	7.50	14.44	12.12	19.4	15.2
900	1.575	1.0417	2.55	2.1632	2.914	2.149	6.4	5.4	10.54	9.6	20.4	18.8
1000	1.4647	1.1551	2.15	1.728	2.7415	2.608	5.4	5.16	7.752	7.6	9.6	8.8

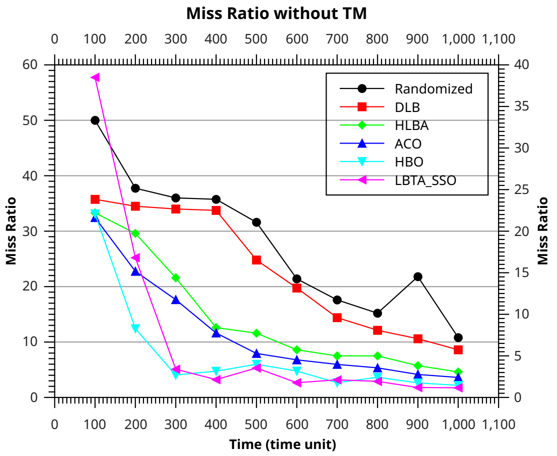

FIGURE 6.22: Miss Ratio when no TM is used in grid computing based environment

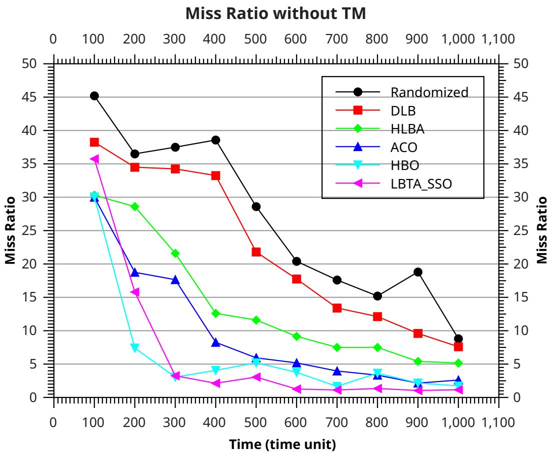

FIGURE 6.23: Miss Ratio when no TM is used in cloud computing based environment

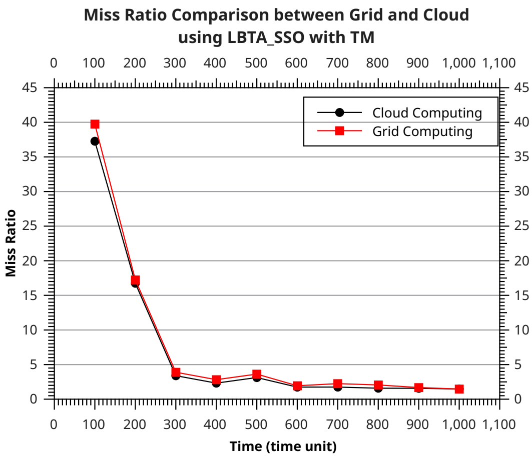

FIGURE 6.24: Miss Ratio comparison between grid and cloud computing when transaction processing is used

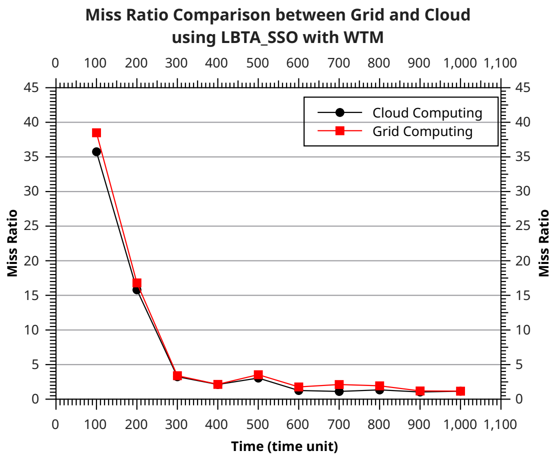

FIGURE 6.25: Miss Ratio comparison between grid and cloud computing when no transaction processing is used

6.6 Observations

In this chapter, we formulated reliability considering deadline-miss fault in on-demand computing based transaction processing system. We have simulated the proposed algorithms considering grid computing scenario and cloud computing scenario for case studies. We modified five known algorithms based on HBO, ACO, HLBA, DLB, and Randomized for the purpose of comparison. We ran each simulation 40 times for every problem instance to get the result. From the experimental results we see that our proposed algorithm LBTS_SSO outperforms other algorithms such as HBO, ACO, HLBA, DLB, and Randomized. We also conducted Normality Shapiro-Wilk tests and Wilcoxon statistical tests.

6.7 Summary

In this chapter, we proposed the balanced task allocation algorithm for on-demand computing based transaction processing system using social spider optimization. The algorithm first balances the load before it allocates the transaction to the appropriate node in the system. We also formulated the resource availability and reliability considering the load. We simulated the algorithm on two scenarios of on-demand computing system; grid and cloud. We compared the proposed algorithm with five modified algorithms. The results show that the resource availability and reliability are maximized. It also reduces the miss ratio. The proposed algorithm works well for a transaction in on-demand computing system. In future, we plan to extend this work to analyze the dependability of the system using stochastic algorithms.