International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 6 (2015) pp. 14029-14040

http://www.ripublication.com

Project Task Scheduling In Grid Computing Using Optimization Algorithm

M.Suguna[1]
[1]*Assistant Professor (SRG),*
Department of Computer Science and Engineering,
Kumaraguru College of Technology, Coimbatore -641049, Tamilnadu, India.
Email: msresearch14@gmail.com
Dr.D.Sharmila[2]
[2]*Professor and Head,*
Department of Electronics and Instrumentations Engineering,
Bannari Amman Institute of Technology, Sathyamangalam- 638 401. Tamilnadu, India

Abstract

Grid Computing is a technology that represents a kind of distributed Computing. Grid Scheduling is the process of scheduling jobs over grid resources. Make span and flow time are the two metrics for measuring the quality of a schedule in grid computing. This paper deals with a innovative task scheduling algorithm for scattered heterogenous computing environments. Considering computational intensive applications in the distributed and parallel computing system, efficient task scheduling is considered as one of the most essential and complex issues. This is an NP-Hard problem, and thus leads to failure of improved scheduling strategy. In order to solve the task scheduling problem and the scheduling strategy in heterogeneous environment, a new scheduling approach is designed named as Chemical Reaction Optimization Algorithm. Chemical Reaction Optimization (CRO) algorithm which is considered as the efficient task scheduling solution. CRO will perform better both in small scale and large scale applications. This algorithm combined with permutation and vector based representation. This algorithm will increase the performance result which is going to prove in the experimental evaluation by calculating the Makespan and Flowtime.

Keywords: Task Scheduling, Proxy Based Task Scheduling Algorithm, Chemical Reaction Optimization.

Introduction

A Computational grid is a hardware and software infrastructure which provides consistent and an inexpensive access for easy processing of high end computations. Grid environment is a distributed environment which consists of various processors with unique capabilities. The term grid computing refers to the combination of computer resources and multiple administrative domains to reach a common goal. Grid System are classified into two categories: Compute grid and data grids. In compute grids compute cycle plays a major role in resource management. In data grid the data's are spread in the geographical locations.

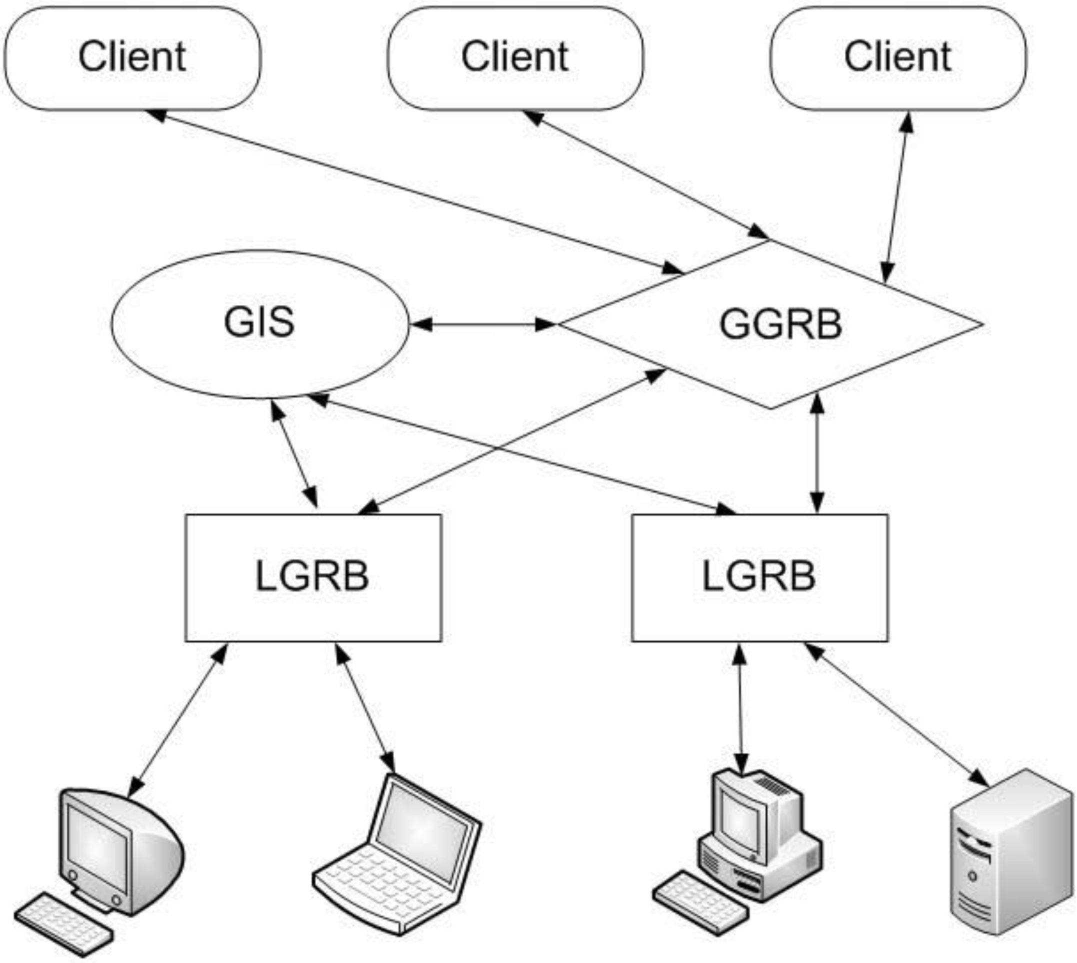

Figure 1: The architecture of the Grid System

A grid usually consists of five parts: Clients, the Global and Local Grid Resource Brokers, GIS (Grid Information Server), and resource nodes as shown in the fig1. Clients register their requests of processing their computational tasks at GGRB. Resource nodes register their donated resource at LGRB and process clients' tasks according to the instructions from LGRB. In practice, client and resource node can be the same computer. GIS collects the resource information from all the local grid resources then it transfers to GGRB.

GGRB is responsible for scheduling. It possesses all necessary information about the tasks and resources and acts like a database of the grid. A grid operates in time intervals. At the beginning of each interval, GGRB performs scheduling. During an interval, the registered resources process the tasks. Scheduling tasks for various heterogeneous nodes is an essential part of a grid system. Efficient scheduling (i.e., performing a scheduling scheme which minimizes the computational overheads and uses the resources effectively) is one of the most important problems in grid computing. A large number of heuristic algorithm is used for task scheduling on grid computing system. The commonly used algorithms are opportunistic load balancing, Genetic Simulated Annealing (GSA), Genetic Algorithm, Ant Colony Optimization. In Grid Computing PBSA Algorithm is proposed for task allocation. Based upon the functional behavior of the job the task is allocated to the resource.

Scope and Contribution

The main objective of the Chemical Reaction Optimization Algorithm is to improve scheduling strategy for the task allocation in parallel and distributed systems. The proposed system focussed on some factors such as Reduce Makespan and Flow time. Global Grid Resource Broker(GGRB) which mainly supports for decision making in order to group the jobs received based on size of the job, deadline of the job, functional service for the requested job. Job Global Information Server (GIS) which mainly supports for decision making in order to allocate the jobs to the donated Local Grid Resource Broker (LGRB). The above factors are going to compare with Operators in Chemical Reaction Optimization (CRO) and it will be displayed in the experimental evaluation result chart.

Related Works

In this section, many algorithms that were designed for the scheduling of Meta tasks in computational grids is discussed. One of the easiest techniques in grid scheduling is Opportunistic Load Balancing .It workflow tasks in Grid environments are difficult because resource availability often changes during workflow execution. Opportunistic Load Balancing attempts to improve the response time of user's submitted applications by ensuring maximal utilization of available resources. In a typical distributed system, there will be number of interconnected resources with different processing capability that operate independently or cooperatively. Each resource has a master workload which evenly distributes the workload among the resources to minimise the time for execution.

Heuristic Task Scheduling Algorithm in Grid computing environment based upon the predictive execution time of tasks. It is a scheduling strategy that employs mean load as heuristic information.. The tasks are reassigned between two machines to raise the load of the machine with lower-load and reduce that of the machine with higher-load under the mean load heuristic.

Minimum Execution Time (MET)

Some algorithm calculate the execution time of the tasks existing in the meta task on the resources, then assign the resource for each task with the minimum expected execution time for the task. It does not consider the current load and the availability of the machine. The algorithm with this mechanism is called as minimum execution time algorithm**.** Minimum completion time algorithm calculates the completion time for a job on all machine by adding the machines availability time and the expected execution time of the job on the machine. The machine having minimum completion time for the job is selected. This algorithm considers only one task at a time.

Min-Min Algorithm

Min-min algorithm begins with a set of all unmapped tasks. The completion time for each job on each machine is calculated. Select the machine that has minimum completion time for each job. Then the job with the overall minimum completion time is selected and mapped to the machine. Again, this process is repeated with the remaining unmapped tasks. Compared to MCT, Min-min considers all unmapped tasks at a time.

Ant Colony Algorithm

Ant Colony Optimization Algorithm is a graph based algorithm which is coming under the family of swarm intelligence method. Swarm intelligence represents the behaviour of insects like wasps, ants, honey bees etc. Ant colony algorithm was proposed by Dorigi et al in the year 1991.Ant colony algorithm is naturally observed by the activities of real ants. The ants move in a path for searching food source from the nest. There is an indirect communication among ants that is obtained through the deposit of their chemical substance.

When the ants move they follow the same path with the help of the chemical substance named pheromone. This is laid down by the ants on their moving path. The pheromone intensity of the ants is high in a path means then the forthcoming ants will follow that particular path, otherwise it follows another path.

The quality of pheromone on each path will affect the possibility of other ants to select the path. The main goal of the Ant colony algorithm is to find the optimal path. ACO algorithm is dynamic in nature. This converges in a very fast manner but it will produce bad suboptimal solution.

Chemical Reaction Optimization Algorithm

Chemical Reaction Optimization (CRO) algorithms are proposed for the grid scheduling problem which is a population-based Metaheuristics inspired by the interactions between molecules in a chemical reaction. They compare these CRO methods with four other acknowledged Metaheuristics on a wide range of instances. Simulation result denotes that the CRO methods generally perform better than existing methods and performance improvement is especially significant in large-scale applications. Even though they declared their future work can be carried forward in the following three directions. First, other models of grid scheduling (e.g., workflow model, priority model, etc.) can be studied with the CRO approach. Second, to understand CRO further, we can design more operators for CRO, develop some heuristic components dedicated to the grid scheduling problem, and test the performance. In addition, other ways solving multi objective optimization problem can be adopted.[8].

All the users submit the jobs to GGRB. It will receive jobs post by the client from the distributed system and parallel system. For each and every job, it assigns a primary key in order to separate the job from other jobs. Initially a token is allotted for each job. The resource brokers use scheduling algorithms or policies for mapping jobs to resources to optimize system or user objectives depending on their goals. Based on the molecule algorithm, the jobs are allowed to store in the Global Grid Resource Broker. From which each job is allocate to resources with the help of Grid Information Server. Resource can be a single processor or multi-processor with shared or distributed memory and managed by time or space shared schedulers. The processing nodes within a resource can be heterogeneous in terms of processing capability, configuration, and availability.

Here the performance comparison of the approaches based on the following parameter make span which represents the latest completion time among all the tasks. Only GIS allocate the resources directly to GGRB by sending the Local Grid Resource Broker's IP Address to the GGRB. The start time and the end time of each job as well as for each batch will be allowed to store in the centralized CRO database. The make span time of each job is stored in s the make span database. When the execution of each process gets completed the MakeSpan value gets updated in the database.

Algorithm for CRO

Input: molecule $M\omega$
$\omega \leftarrow N(\omega)$
$PE\omega \leftarrow f(\omega)$
$NumHit\omega \leftarrow NumHit\omega + 1$
if $PE\omega + KE\omega \geq PE\omega$ **then**
Generate $a \in [KE\ LossRate, 1]$
$KE\omega \leftarrow (PE\omega - PE\omega + KE\omega) \times a$
$buffer \leftarrow buffer + (PE\omega - PE\omega + KE\omega) \times (1 - a)$
$\omega \leftarrow \omega$
$PE\omega \leftarrow PE\omega$
$KE\omega \leftarrow KE\omega$_
if $PE\omega < MinPE\omega$ **then**
$MinStruct\omega \leftarrow \omega$
$MinPE\omega \leftarrow PE\omega$
$MinHit\omega \leftarrow NumHit\omega$
end if
end if

Operators In CRO

Two Operators are used in CRO namely permutation based representation and vector based representation.

Permutation Based Representation

In CRO both permutation based representation and vector based representation are compared.In the permutation based representation we transform two vector into a single vector.The vector length n is extended to n+m-1 where the added element m-1 are the delimiters are distributed to the task assigned to the nodes.

(i) Insertion Operator

In this operator two numbers are randomly selected from the vector, then the second number is inserted before the first number. In the below example the number 2 and 7 are randomly selected and the number 7 is placed ahead to number 2.

⇒[1,3,**2**,5,6,8,**7**,9,4] [1,3,**7,2**,6,8,9,4]

(ii) Position-Based Operator:

This operator is used for synthesis. It involves two molecules and combines two solutions w_1 and w_2 into a new one w^1. Each number in w1 has 50 percent probability to be chosen and passed to w^1 at the same position. Then, the numbers in the unfilled positions of w^1 are picked from w^2. For example, let w^1 be [1,3,7,5,6,8,2,9,4]. We randomly choose the numbers 7,6 and 2, then pass them to the same positions in w^1. The remaining numbers, 1, 3, 4, 5, 8, and 9 in w^1 are put in the vacant positions according to their orders in w^2, which is 4, 5, 3,1,8,9. Thus, we have

Strings Before Synthesis

w^1:[1,3,7,5,6,8,2,9,4]
w^2:[7,4,5,3,1,2,8,6,9]

String After Synthesis

w^1:[4,5,7,3,6,1,2,8,9]

(iii) Two Exchange Neighbourhood Operator

This operator is mainly used for quadratic assignment problem and resource constrained project scheduling problem. In this operator two numbers are randomly selected from the solution and the selected numbers position gets exchange. From the below example the positions of 2 and 6 are exchanged form the solution.

⇒[1,3,**2**,5,7,8,**6**,9,4] [1,3,**6**,5,7,8,**2**,9,4]

Vector Based Representation

Four operators are used in this vector based representation.

(i) One-Resource Change Operator

This operator is used for on-wall ineffective collision. It randomly choose one task and assign to the another resource. In the below example the fifth task is assigned from resource 3 to resource 2.

⟹[1,3,1,4,**3**,2] [1,3,1,4,**2**,3]

(ii) Pair wise Exchange Operator

This operator is mainly focused on on-wall ineffective collision and intermolecular ineffective collision. Two tasks are randomly selected and their resources are exchanged for the selected task. In the given example the second task and the fifth task are assigned with resources 3 and 2. After the change, the second task is now assigned to resource 2, and the fifth to resource 3.

⟹ [1,3,1,4,2,2] [1,2,4,3,1,2]

(iii) One-Position Exchange Operator

This operator combines two solution w1 and w2 into a new one w^1. An integer value k is randomly generated in the range of [1,n], where n is the total number of tasks. Then w^1 is generated by picking the first k values from w1 and the rest of the (n-k) values from w2. Below, w^1([1,3,2,3,4,1]) is formed by combining the first two numbers(1 and 3) from w1 and the last four (2,3,4 and 1) from w2. Thus, we have

Strings Before Synthesis

w^1:[1,3,1,4,2,2]
w^2:[3,2,2,3,4,1]

String After Synthesis

w^1:[1,3,2,3,4,1]

(iv) Half-Random Operator

We produce two new solutions $w1^1$ and $w2^1$ from one solution $w^{1.}$ We divide the values of the solution into two sets, depending on whether they are in the odd or even numbered positions of $w1^1$, while the even-position set of w is assigned to the even position of $w2^1$. The unassigned positions in these two new solutions are filled with random numbers. As the example shown below, the numbers 1,1 and 2 are given to $w1^1$ in the same odd places, while 3,4, and 2 are allocated to $w2^1$ in the even places. Then random numbers (2,3, and 1)is generated to fill in the vacant positions of these two new solutions. Thus, we have

Strings Before Decomposition

w:[1,3,1,4,2,2]

String After Decomposition

$w1^1$:[1,2,1,3,2,1]

w2[1]:[2,3,3,4,1,2]

Experimental Results

Grid computing aims to assign tasks to computing nodes and minimize the execution time of tasks as well as workload. It is an emerging computing model that provides the ability to perform higher throughput computing. The resources of many computers that are connected by a network is used to solve large-scale computation problems. The proposed algorithm is capable of finding the optimal resource for processing a job.Scheduling of jobs to resources as per their requirements and properties.

Major function is to find the optimal resources and to allocate them to each individual job. The objectives of a grid scheduler are overcoming heterogeneousness of computing resources, maximizing overall system performance, such as high resource, utilization rate and supporting various computing intensive applications. The service gets started for the task allocating to the resources. The start time and end time of the task is calculated. The time utilization for users' job and the budget utilization are calculated. Based upon the job execution time the task is allocated to the resource.

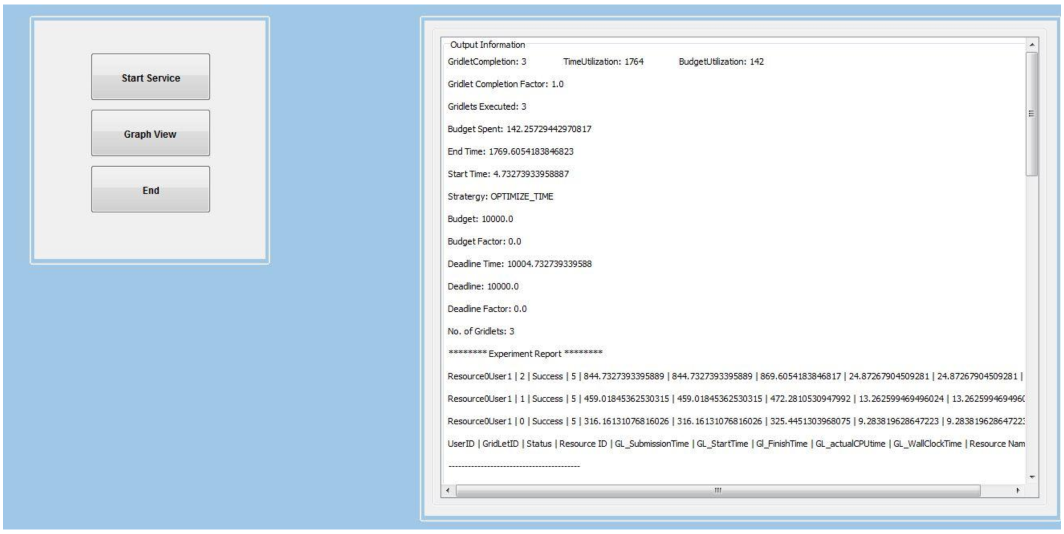

Figure 2: Starting CRO algorithm for Scheduler

The job which having low execution time having high priority to allocate resources. and a comparison between the makespan of existing algorithms with parameter specifications described earlier. As the experimental results show CRO is performing best compared to existing algorithm. This algorithm produce 80% but in existing algorithm it produce only 72%.This is expected as the CRO is keeping track of the state of all the resources at each point in time which makes it able to make more optimal decisions at each point in time.

Initially the grid network is initialized using the gridsim simulator. The Grid Sim toolkit allows modeling and simulation of entities in parallel. The service gets started for the task allocating to the resources. The start time and end time of the task is calculated. The time utilization for users job and the budget utilization are calculated. Based upon the job execution time the task is allocated to the resource. The job which having low execution time having high priority to allocate to the resource.

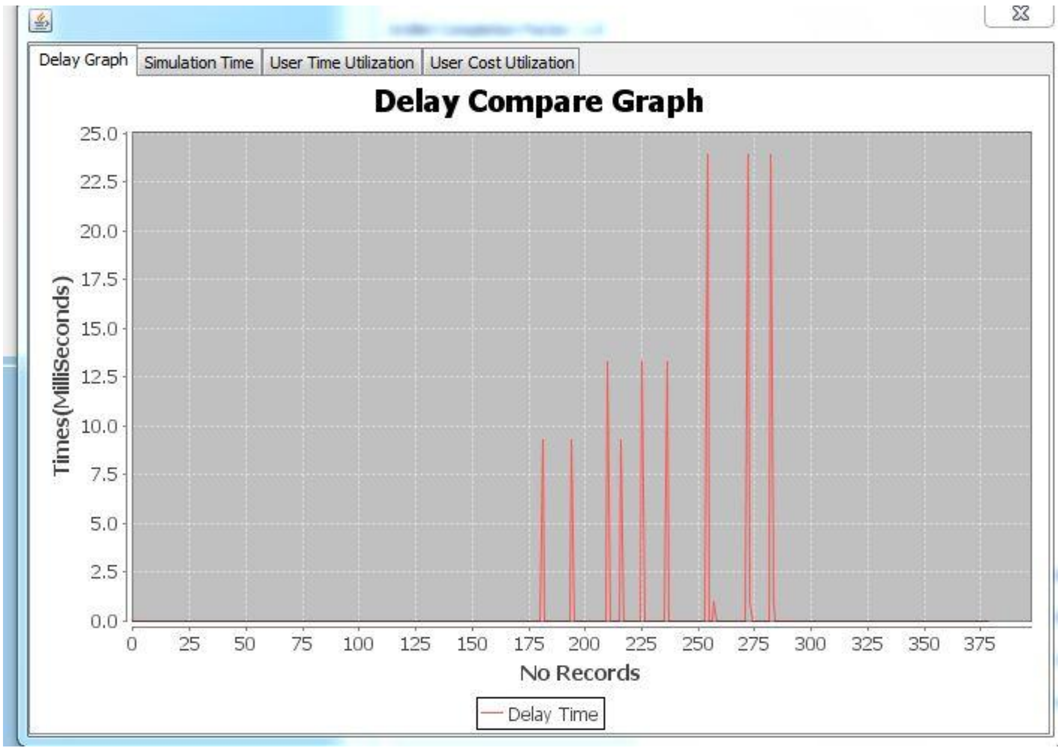

Figure 3: Simulation result of Comparison of the Delay between Different User

The time delay for the task allocated to different resource is give in the above graph.In this graph the X-axis represents the records and Y-axis represent the times in millisecond. The time delay between three users are plotted in the graph.

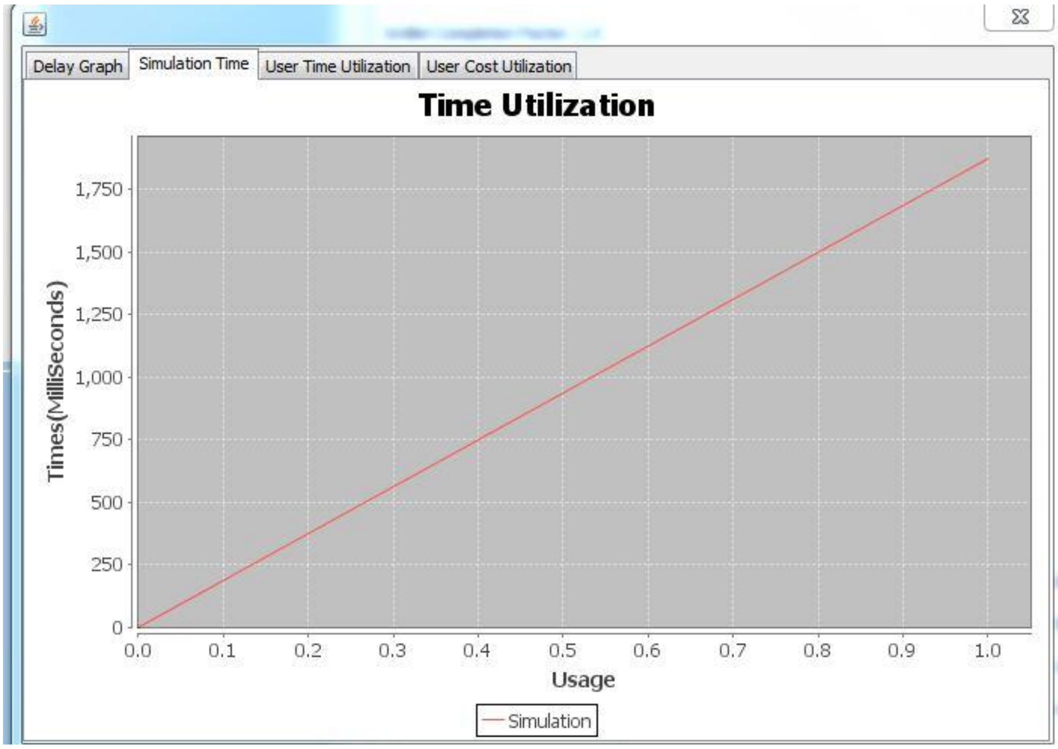

Figure 4: Simulation Time for Scheduling Jobs

In the above fig4 shows the overall simulation time for all task executed in the system is specified. In this graph the X-axis represent the usage of processor and Y-axis represent the time. The total time taken to complete all the task is represented in the graph. For task allocation the time taken for the completion of each task gets calculated.

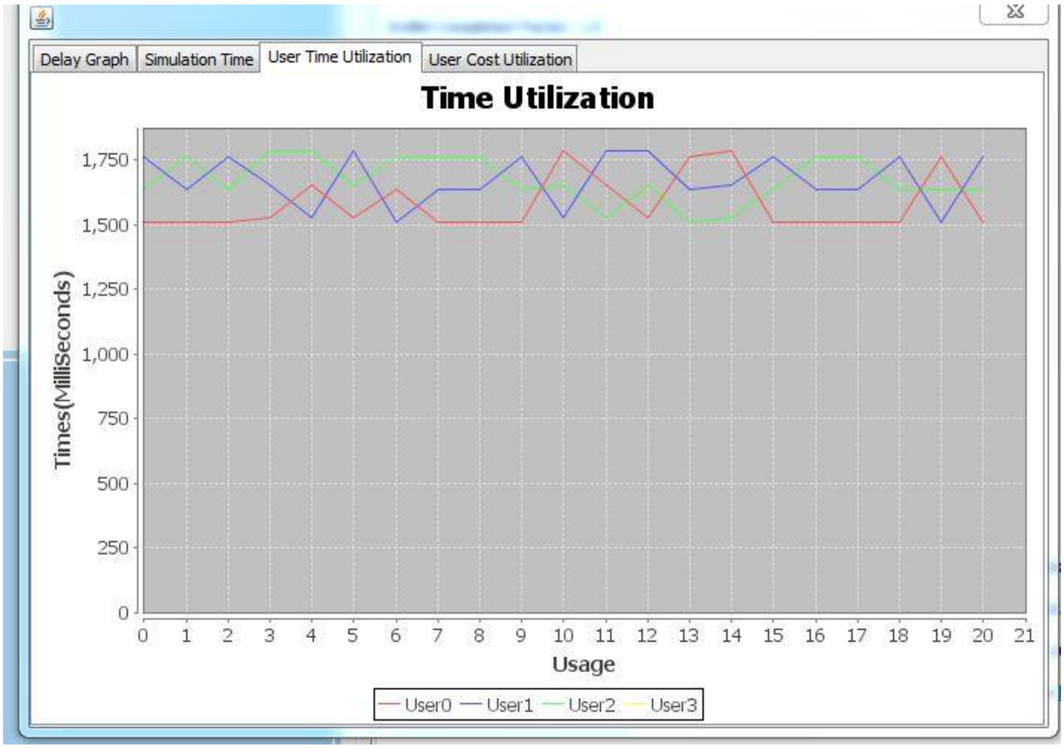

Figure 5: Simulation Time Utilization Graph

The total time utilized by different users for scheduling process is plotted in the above graph. The X-axis represent the usage of processor and Y-axis represent the time.For user 0 the total time taken to complete the task. For user 1 total time taken for the completion of the task. For user 2 how much time taken to complete the task is represented in the above graph.

Conclusion And Future Work

Recent researches have proved that scheduling on computational grids is best solved by Chemical Reaction Optimization Algorithm. Hence CRO is developed to allocate proper resources for each job. This algorithm guarantees efficient resource allocation of the machines. The experiment results states that the CRO algorithm solves the resource scheduling problem in the grid computing environment faster than the existing algorithm and results in reduced makespan.The future work will focus on implementing proxy based task scheduling algorithm. This algorithm compares with other algorithms to produce high throughput computing and reduce the Makespan and Flowtime.

Acknowledgments

This work is supported by bit lab. This above submitted work is true of our knowledge

References

[1]. Samaneh Hoseini Semnani, Kamran Zamanifar, Naser Nematbakhsh “New Heuristic in Ant Colony Optimization to Solve Job Scheduling Problem in Grid”, Dept. of Computer Science, University of Isfahan,Isfahan, Iran : 2009.

[2]. B Yagoubi and 2Y. Slimani “Task Load Balancing Strategy for Grid Computing”, Slimani University of Oran, 2007.

[3]. Foster, I., Voeckler,J., Wilde,M., Zhao, Y., Chimera, “A Virtual Data System for Representing, Querying, and Automating Data Derivation”, in 14th International Conference on Scientific and Statistical Database Management (SSDBM 2002), Edinburgh, July 2002.

[4]. Lina Ni1,2, Jinquan Zhang1,2, Chungang Yan1, Changjun Jiang1 1 “A Heuristic Algorithm for Task Scheduling Based on Mean Load”,Department of Computer Science, Tongji University, Shanghai, 92, China, 2000.

[5]. R.Armstrong, D.Hensgen, and T.Kidd, “The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-time Predictions”, In 7th IEEE Heterogeneous Computing Workshop(HCW’98), pp. 79-87, 1998.

[6]. R.F.Freund, and M.Gherrity, “Scheduling Resources in Multi-user Heterogeneous Computing Environment with Smart Net”, In Proceedings of the 7th IEEE HCW, 1998.

[7]. M. Dorigo, T. St¨utzle, "the ant colony optimization_metaheuristic: Algorithms, applications, and advances". In F. Glover, G. Kochenberger editors, Handbook of Metaheuristics, volume 57 of International Series in Operations Research and Management Science, pages 251–285. Kluwer Academic Publishers, 2002.

[8]. Jin Xu, Student Member, IEEE, Albert Y.S. Lam, Member, IEEE, and Victor O.K. Li, Fellow, IEEE, “Chemical Reaction Optimization for Task Scheduling in Grid Computing”, IEEE Transactions on Parallel And Distributed Systems Vol. 22, No. 10, October 2011.