Improving the Resource Allocation in Grid Computing using Fusion of SS-GA

S. Ramachandra[1], R. Praveen Sam[2]

[1]Assistant Professor, CSE, MTIET, Palamaner

[2]Professor of CSE, G. Pullareddy Engineering College (Autonomous), Kurnool

Abstract: ***Grid computing is the collection of resources from multiple locations to reach a standard goal. It interconnected the systems where the ma-chines utilized the same resources. Actually, the failures and high execution time depends on the resource allocation. The Sharing of resources of grid computing are related to the distributed computer systems. The sifted scheduler managed the work flow and Genetic Algorithm finds the applications and generates the solutions to optimization of the problems. The proposed method minimizes the response time and maximizes the utilization of work mode. The optimization factors are reduces the rate of failures and execution time of grid. This paper proposes the response time and improves resources providing of grid computing.***

Keywords: Grid Computing, Resource Allocation, Swift Scheduling, Genetic Algorithm, Optimization, Execution time, Computational time

1. Introduction

In grid computing, resource allocation is a critical problem, which seriously affects the capability of the entire grid system [2]. In computational grids, heterogeneous resources with different systems in different places are dynamically available and distributed geographically. The user's resources requirements in the grids vary depending on their goals, time constraints, priorities and budgets [3].Allocating their tasks to the appropriate resources in the grids so that performance requirements are satisfied and costs are subject to an extraordinarily complicated problem [4]. In the proposed algorithm, the idea of SWIFT scheduling-genetic algorithm is to use the Genetic algorithms are search algorithms [5].

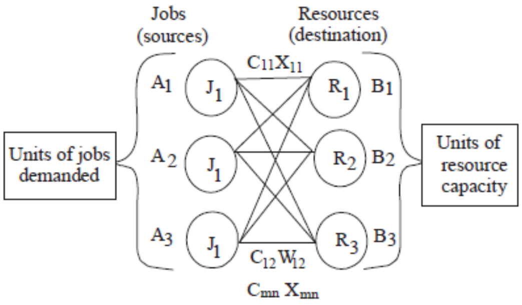
In the process of evolution, the individual is continuously changing genes to adapt to the environment of his life. The advantage of genetic algorithm is adaptive in nature, and reduce make span and flow time. In Grid re-source condition focuses on running track of resources in grid computing system by means of analyzing historical monitoring the data [6].

The Swift Scheduler (SS) in GridSim maps jobs from re-source queue as well as resources from job queue by the use of heuristics function.

According to Swift Scheduler, jobs allocations as well as resource selection process are executed using heuristic searchingalgorithm on Shortest Job First, which minimizes the average time waiting of jobs [1]. Therefore, the turnaround time is minimized and resource utilization is greater than before.

2. Related Work

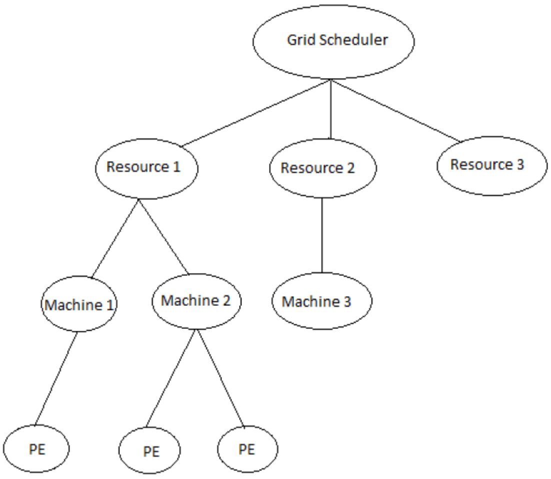
Over the past few years, the development of high speed networks, increasing overall computing capacity and low cost of storage devices have given rise to a new paradigm of distributed computation: grid computing [1].



Resource Allocation in Grid

Stages of genetic algorithms is:
1) The establishment of the initial.
2) The calculation of fitness values.
3) Selections.
4) Regeneration.
5) Creation of new population regenerated.

Despite of its advantages, Grid computing also entails many new challenges, like the adaptation of parallel programs previously developed for homogeneous resources clusters to the dynamic and heterogeneous Grid resources with minimal intrusion into the code or resource scheduling.



Scheduling Hierarchy

In a grid environment, the scheduler is the manager of the workflow, acting as intermediary between the user and the distributed resources, thus concealing the complexity of the grid system [2].

Equations

Let N be the number of jobs in Job Queue 'Jq' which is indicated as,
Jq = fJ1; J2; J3; ::::; JNg (1)
Jobs are allotted to M number of resources in Resource Queue'Rq'which is indicated as,
Rq = fR1; R2; R3; ::::; RMg (2)

Let F(Ji; Rj) be the overall job completion time for the ith job in jth resources can be calculated as,

$$\rangle \rangle F(Ji; Rj) = \rangle \rangle G(Ji; Rj) + \rangle \quad \rangle H(Ji; Rj) \quad (3)$$

Let G(Ji; Rj) be the expected job completion time of the ith job in jth resources which can be calculated as,

$$\rangle \rangle G(Ji; Rj) = \rangle \rangle (JLi = RCj) \qquad (4)$$

JLi be the Job length of ith Jobs and RCj be the capacity of the jth resources.

Let H(Ji; Rj) be the heuristic function of the ith job in jth resources which can be calculated as

$$\sum_{i=0}^{N}\sum_{j=0}^{M} H(Ji; Rj) = \sum_{i=0}^{N}\sum_{j=0}^{M} (JLi = RCj) + \text{COMMUNICATIONOVERHEAD}$$

3. Methods

3.1 Methods

SWIFT works as follows.
This Method works based on resource allocation.
Step 1:Initializing the n number of jobs in the pool
Step 2: Initializing the available M number of resource in Grid
Step 3: N number of jobs should be schedule in ascending order based on the job's length
Step 4: Search the resource by the meta-heuristic function based on less time and allocating the jobs of appropriate resources in the system.

3.1.1 Procedure for Swift Scheduler

```
Begin:
1.        Initialize job queue, resource queue
 Loop exec for 'N' jobs
2.        Begin:
          Initialize job service time and add jobs (N) to job queue.
End
3.        Loop exec for 'M' Resources
Begin: Add the Resources (R) to resource queue.
   End.
   Arrange jobs in the ascending order based on job length and maintained jobs Queue.
Loop exec for 'N' jobs
   Begin:
Loop exec for 'M' resources
   Begin:
      Calculate the processing time of 'N' in 'M' resources.
       End.
   End.
   Loop exec for 'N' jobs
Begin:
 Loop exec for 'M' resources
 Begin: Select the best processing time resource for the each jobs using heuristic function like allocate (n, m).
  End.
 End.
End.
```

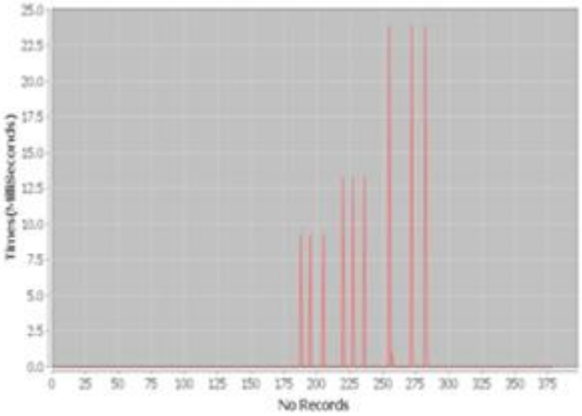
3.2 Genetic Method

Begin
Step 1: Initializing the population with randomly generated N number of chromosomes
Step 2: Initializing the available M number of resource in Grid.
Step 3: Start the loop for finding the appropriate resources in grid
Step 4: Fitness function should be evaluation for each chromosomes of grid.
Step 5: For next generation - selecting the chromosomes for mating
Step 6: The chromosomes between the random pairs
Step 7: Chang the chromosome with new chromosome
Step 8: When the chromosome finds the best resource in grid then stop the loop.
End

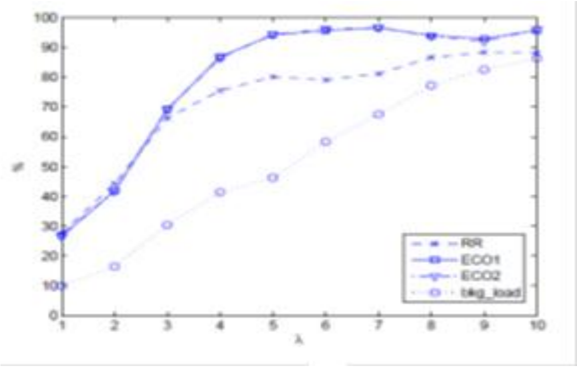
4. Techniques

SS-GA Technique:
Begin
Step 1: Initializing the jobs in the pool
Step 2: Initializing the available resources in resource job pool
Step 3: Scheduling the job with high priority in the increasing order in grid.
Step 4: If the job with no priority then it schedules the job in shortest job first order.
Step 5: Initializing the scheduled Population size P(s) based on the number of jobs in grid.
Step 6: Performing the function of fitness to minimize the following objective function
Step 7:Job Completion Time and Cost.
Step 8: Select the half of the individuals among the population P(s) having better the fitness values
Step 9: The selected individuals with a crossover rate
Step 10:The rate of the new better chromosomes to be carried by mutation
Step 11: Terminate the above process after 'n' iterations to achieve better results in case of objective function in resource allocation in grid
End.

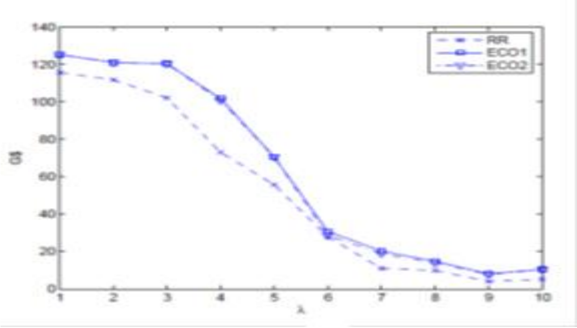
5. Simulation Results



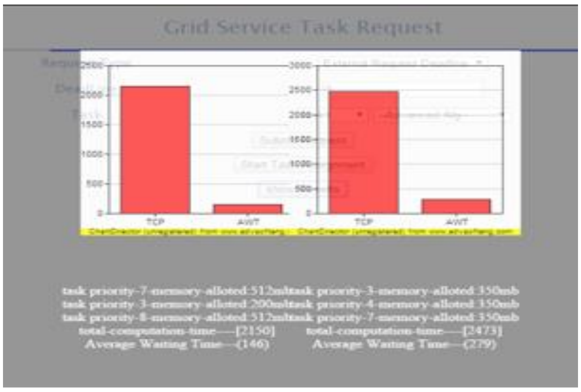
Delay Compare Graph



Computational Result 1



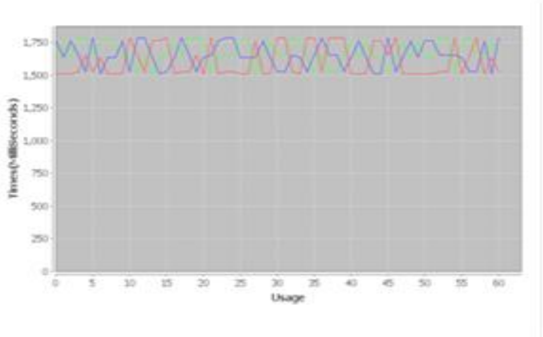
Computational Result 2



task priority-7-memory-alloted:512mbtask priority-3-memory-alloted:350mb
task priority-3-memory-alloted:200mbtask priority-4-memory-alloted:350mb
task priority-8-memory-alloted:512mbtask priority-7-memory-alloted:350mb
total-computation-time----[2150] total-computation-time----[2473]
Average Waiting Time---(146) Average Waiting Time---(279)

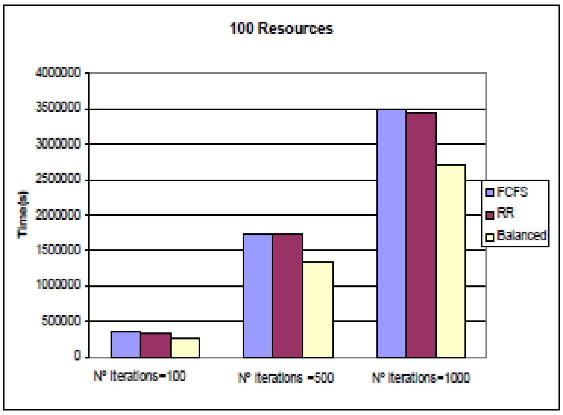
Graft of Time Comparisons

This Graph Show the different between the time Comparisons of the Grid.



Time Utilization

Figure 3: The above graph shows the time utilization and usage.



Graft shows the time of Execution in Genetic Algorithm

6. Conclusion

In this paper, we proposed Swift scheduler that completes a task by using highly utilized low cost resources with minimum computational time. Our scheduling algorithm uses the heuristic function to select the best resources to achieve a higher throughput while maintaining the desired success rate of the job completion. This algorithm is performing better than real time job parameters and suitable for different job sizes in real environment.

Reference

[1] SWIFT: Scheduling in Web Servers for Fast Response Time.

[2] A dynamic-balanced scheduler for Genetic Algorithms for Grid Computing.

[3] SonalNagariya, Mahindra Mishra *"Resource Scheduling in Grid Computing: A Survey"* International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 10, October 2013.

[4] MayankRawat and Ajay Kshemkalyani *"SWIFT: Scheduling in Web Servers for Fast Response Time"* Univ. of Illinois at Chicago, Chicago, IL 60607.

[5] SaurabhMandloi, Hitesh Gupta *"A Review of Resource Allocation and Task scheduling for Computational Grids based on Meta-heuristic Function"* International Journal of Research in Computer and Communication Technology, Vol 2, Issue 3, March-2013.

[6] K. Somasundaram, S. Radhakrishnan *"Task Resource Allocation in Grid using Swift Scheduler"* Int. J. of

Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. IV (2009), No. 2, pp. 158-166.

[7] U.Karthick Kumar “*A Dynamic Load Balancing Algorithm in Computational Grid Using Fair Schedulin*” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 1, September 2011.

[8] Haruna Ahmed Abba, Nordin B. Zakaria and NazleeniHaron*“Grid Resource Allocation: A Review”* Research Journal of Information Technology 4(2): 38-55, 2012.