Dynamic Resource Management, Reservation and Scheduling in Grid Computing

Brabulin J Machij[1], Ravi Maran. S[2]

[1]**Department of Computer Science and Engineering**
M.A.M College of Engineering, Siruganur, Trichy, Tamil Nadu, India

[2]**Department of Computer Science and Engineering**
M.A.M College of Engineering, Siruganur, Trichy, Tamil Nadu, India

Abstract

Grid computing is sharing of task over multiple computers. In the real world grid resources are graphically dispersed and different organization has their own access policy, cost and mechanism and also there arises the situation related to the dynamic job behavior. This may cause inefficient resource usage, job failures and breakdown of QoS contracts. The proposed system has a dynamic resource management which manages advanced reservation and dynamic scheduling. The new algorithm is based on advanced reservation and Ant Colony Optimization algorithm. This helps in availability of resource when needed, reduction of cost, QoS oriented and are reliable. The experimental result showed the scheduling algorithm can tackle the similar request better to other scheduling algorithm.

Keywords: *kAPAC Algorithm, Ant Colony Algorithm,Advanced Reservation, Resource pool*

1. Introduction

Grid [1] has become a most important way to share resources in a heterogeneous environment. Grid computing is the most distributed computing, and it makes use of computers communicating over the Internet to work on a given problem. Grid is used to share the most kinds of widespread resources in the distributed environment such as computing resources, storage resources, bandwidth resources, software resources, data resources, information resources, knowledge resources, etc. Grid can be used in resource sharing by means of virtual environment. Grid scheduling deals with how tasks are distributed on resources. Efficient task scheduling strategies and algorithms in grid can fully utilize the processing power of grid system and improve the performance of it.

One of the basic uses of grid computing is to run an existing application on a different machine. So scheduling should be needed to be done properly with the existing resources. The major drawbacks enrolled in the grid system are unavailability of resources, QoS constraints (in terms of delay), job failures .These constraints can be removed by using a proper scheduling mechanism in a grid environment.

To facilitate it, the grid resource is reserved by an advanced prediction mechanism and then scheduled properly by an ant colony scheduling algorithm. This enhances better QoS constraints, ability to handle multiple jobs, job contention is removed and resources are properly utilized to complete jobs.

The remainder of the paper is organized as follows. In Section II related work were presented. In Section III proposed system model were presented. In Section IV experimental setup and analysis were presented. In Section V conclusions were made.

2. Related Works

Resource management is needed to all the grid nodes. The past study with the grid computing in the context of resource reservation and scheduling can be briefly noted and the eagerness needed for the new scheduling system can be briefly explained below.

R. Venkatesan, K.Ramalakshmi [2] proposed a multiple queuing scheme for handling queuing mechanism and advance reservation is made based on the trust factor. The resource which has enough capacity can reserve new task otherwise try to reserve the new task in the next resource with a lower trust factor else cannot be reserved. In contrast to their work, different type of advanced reservation method were being enforced with the similar technique which can focus on better utilization of resources and also capable of meeting reservation request.

S. Gomati, Dr. D. Manimegalai [3] proposed scheduling mechanism for the group of resources. They were

grouped based on some properties . The grouped jobs are allocated to resources in dynamic grid environment taking into account memory constraint, processing capabilities and the bandwidth of the resources. soon as the jobs are put into a group with a matching selected resource, the grouped job is dispatched to the selected resource for computation. This made to group resources based on nearest neighboring distance and finally reserving the resources.
Zhiangwu, Jiecuo [4] proposed a new grid system architecture using resource pool. Then the Quality of service distance computation method for hybrid variable types is presented between various resources pool. Then, new scheduling algorithm based on k set adaptive prediction admission control(kAPAC) was implemented by first finding the QOS distance between the k set element and thereby then predicting the job to the respective resource pool.In contrast to this work the feature is enhanced by means of an scheduling algorithm.

Hui Yan, Xue-Qin Shien, Xing Li, Ming-Hui Wu [5] proposed the improved ant colony algorithm for the job to be scheduled. This algorithm proposed the new form of new adaptive mechanism for scheduling. It also made a new load balancing mechanism by which the the load balancing factor is not used and only based on the adaptation by the pheromone the load gets balanced. That had made the job finishing rate at different resource in quicker. In contrast to this work the feature is enhanced by reserving the resources before scheduling it.

Maria Chtepan, Mohd Noor Sap, Abdul Hanan Abdullah, and Chai Chompoo-inwai [6] proposed the dynamic scheduling which require specific job requirement, dependency and QoS contracts. This tells about the different characteristics in the dynamic grid environment. Siriluck Lorpunmanee, Mohd Noor Sap, Abdul Hanan Abdullah, and Chai Chompoo-inwai [7] proposed the problems addressed by developing a general framework of grid scheduling using dynamic information and anant colony optimization algorithm to improve the decision of scheduling.Incontrast to it job migration between user request and resource can be made easily.

3. System Model

3.1 Reservation Mechanism

When a user request a job, the job will be first reserved by a resource. In managing the grid system, the accepted request are bound with fixed resources. There are two significant limitations are noted. First, some secondary QOS attributes determine request whether to be accepted or not. The success rate may get decrease in extreme extend. Second, availability of resources are not considered. In fact, resources will be utilized by reservation requests in future time. Therefore resources with low availability lead to the failure of bound request in a high probability. In advanced reservation the resources are reserved before its need. So the resources cannot be used by other user till it finishes the job. This causes much time to be wasted and the similar resources which are needed by the other system needs to wait for the job completion of the first or it needs to be discarded. This causes much of the time to be wasted or the service time is much reduced. In this paper the better reservation mechanism is discussed.

kAPAC Algorithm Implementation

Admission control Manager will determine whether to accept requests from user to grid resource pool. Now the RSLA(Resource Service Level Agreement)is compared with TSLA(Task Service Level Agreement) .If the QoS requirements specified in TSLA are satisfied by RSLA, request can be accepted by system if exceeds the limit then reservation is not allowed which is described by the kAPAC algorithm and is being implemented by the Admission Control Manager which is explained in Table 1.

Table 1: kAPAC Algorithm

1	wait_request(q)//reservation request q arrives
2	compute the number of similar request with q in Queue,which is written $Index_{rej}$
3	if($Index_{rej}$ >=k)then// the similar request are enough to reject q
4	reject_request(q)
5	else
6	select k nearest neighbours(kNN) of q in resource set
7	compute the availability of q's kNN,which is written availability_k_set(q)
8	if(availability_k_set(q) >threshold)then
9	accept_request(q) and Queue.enqueue(q)
10	end of while

First request from the user is arrived on the job queue.Then the request will be placed on the Queue. If the similar request from different sources arrive then calculate the number of similar request.If the request is greater than the Index of the queue then reject the request or if the request is less than the maximum element of the index element find the QoS distance of the resource This Qos distance is based on response time, short time delay.After finding the Qos distance then find the availability in the resources.If the availability of the resources is greater than the threshold level then accept the request and reserve the request. By doing this, confirmed request were only allowed to reserve . After reserving enqueue the next request and repeat the above.

Computing the Qos Distance

For the better performance of reserving QoS distance should be calculated. Some QoS parameters are response time, bandwidth and delay.QoS distance can be calculated to reduce the latency between the system. It can be calculated using the formula given below. Let $T_i$and T_j denote two QOS vectors and distance d (T_i, T_j) is calculated as

$$d\,(T_i, T_j)=\frac{\Sigma_{n=1}^{p}\delta_{ij}^{n}d_{ij}^{n}}{\Sigma_{n=1}^{p}\delta_{ij}^{n}} \qquad (1)$$

The equation (1) can be used to find the QoS distance. Here ifT_i, T_j does not assign a variable to n_{th} variable, then $\delta_{ij}^{n} = 0\ else \delta_{ij}^{n} = 1$.The distance between the n_{th} variable between $T_i and T_j$ is written d_{ij}^{n}which relies on the variable type.The d_{ij}^{n} can be calculated using the formula given below.

$$d_{ij}^{n} = \frac{|x_{in}-x_{jn}|}{max\ x_{hn}-min\ x_{hn}} \qquad (2)$$

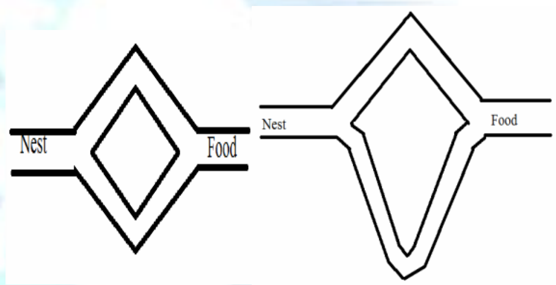
In this equation (2) the distance can be calculated using the $x_{in}, x_{jn}$are the distance between two attributes. x_{hn}, x_{hn} represents non missing x_{in}, x_{jn} values. Thus the QoS distance can be found out.

During the reservation mechanism the request from the user arrives and the resource from the most of the top k request are reserved based on two factors. The factors are as follows. In the beginning, recently used resources will be given more importance and its availability last till all the similar request are responded. Next if the idle time increases the availability of recently used resources were made less available and the new resources were made for next incoming request.

It is found that the recently used resources if not used immediately then the chance of scheduling decreases with increase in time since many resources wait for and it. Also the resources which is current available can be used for the longer time till its request were granted.

3.2 Scheduling using Ant Colony Algorithm

Ant colony optimisation algorithms are used for finding the shortest path. Usually ant lives in colonies in the nest. They take food by moving to their own desired path. So each ant migrates in different direction in search of food. While going on their path they excrete an substance called pheromone by which different ant which are coming backwards follow it. On finding the food it takes the food and comeback with a different shortest path. All ant follow up and the pheromone layer in the ground becomes dense and finally all ants will take the most shortest path. After few time the pheromone layer get disappears and better short path is made and all ants will continue the path. This principle can be experimentally proved by double bridge experiment which is shown in fig.3.1



(a)Branches have different lengths (b) Branches have different lengths

Fig 1Experimental setup for the double bridge experiment.

The experiment tells that two paths are connected from the same nest and finally combines at the food source. In the experiment the path of one has same length in both the direction which is shown in fig. 1(a). In this first one will take a probability of 50% of the ants will take the first way while the 50% others will take the other way, as they have no clue to conclude the ground configuration. The other one which is shown in fig. 1(b) has the unequal length of one path and it is found the bottom path has double the distance of the upper one. But in this the probability of getting both 50% chance will not occur. The short route is always preferred. The ants taking the shorter path will reach the food source before the others and leave behind them the trail of pheromones. After reaching the food, they will turn back and try to find the nest. Since the pheromone is more at the shorter path all ants will follow the same path and the long path is left completely. Thus the the ant which carries the food will take the path already explored, as it knows ti was the best path to the nest. Since the ant chooses the shortest way and will continue to deposit pheromones, the path will be a key way for other ants to take the same path..The ants which had taken the long way will have more choice to come back to the shortest way, and after some time, they will merge with the best path. By this way the ants will find the shortest path by themselves, without having a global view of the ground. Each Ant takes the decision by the pheromone level and amount that is needed to explore, find the food, and bring it back to the nest , in an optimized way. Each ant constructs its own solution by making decisions, using existing problem constraints and heuristics combined with experience which is analogous to a substance called pheromone. The colony then reinforces decisions in the construction process according to their successes by adding pheromone, which also decays to mitigate against poorer decisions. Thus ant colony algorithm is used to find the optimal solution for scheduling tasks.

Table 2.Ant Colony Algorithm

```
1.procedure ACO
2. begin
3. Initialize the pheromone
4. while stopping criterion not satisfied
do
5. Position each ant in a starting node
6. Repeat
7. for each ant do
8. Chose next node by applying the state transition
rate
9. end for
10.transition possibility from the next job to resource
is found
11. until every ant has build a solution
12. Update the pheromone
13. end while
14. end
```

3.3 Proposed Scheduling Architecture

In the scheduling architecture shown in fig 3.2,first the Admission Control Manager decides whether to reserve the user request or to neglect it.For this first request from user is taken.From this the similar request from all user ik taken as k,.Next it tries to find out the Queue Index.If this queue is not filled then enqueue the user request and also check the similar request to be lesser than k. After verification the QoS distance should be calculated.This ensures that the nearest neighbour resource distance is being calculated.Finally, the availability of the resource if greater than the threshold then reserve it.This method is very good because it can handle similar user request in a higher probability. Next after reserving the request the ant colony algorithm is used in scheduling. The grid system is composed of several computational resources. The resources may be in heterogeneous environment supporting different platforms. . The grid scheduler finds out the better resource of a particular job and submits that job to the selected host. In addition, the similar job can be easily scheduled using the algorithm. Since grid system works well for huge resources, scheduling can be done by sharing the same resources and subdividing a huge resource into so many sub task and the resources can be taken by many ants at the time and later each ant get combines the resources from each other or take the resources one by one from the resource pool. This causes much of the request to be scheduled so that the similar request can be handled properly and the time being spent by each ants also becomes lesser.This way of scheduling much important because job contention will never takes place because only after conforming after resource availability the resource is being reserved and scheduled.

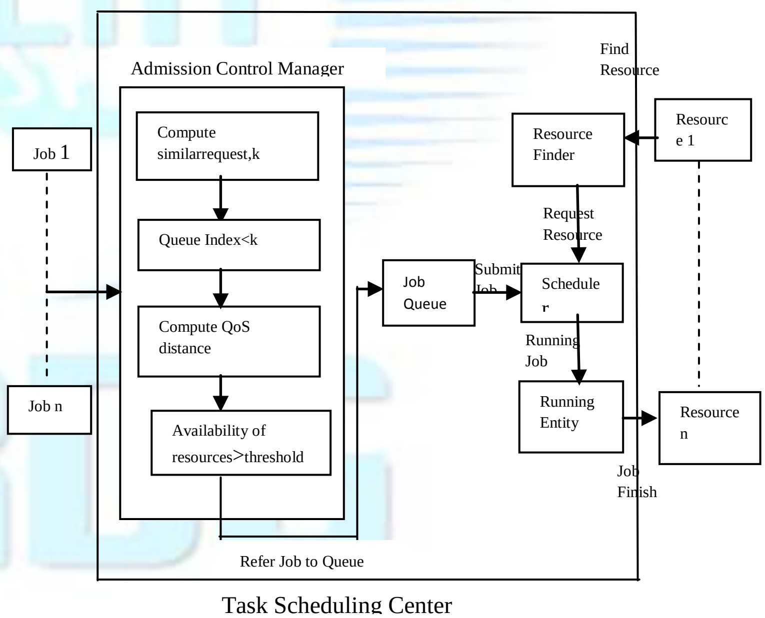


Fig 2 Scheduling Architecture

4. Experimental Setup And Analysis

GridSim [8] is used to simulate the grid computing environment. It simulates the scheduling mechanism in a effective manner.

4.1 Performance Analysis

The Performance of the scheduling algorithm can be easily reviewed by comparing with other scheduling algorithm. When user submits a job and relevant parameters such as computation workload, communication, quantity and deadline of the request is noted. By comparing the result with the exiting scheduling algorithm it was found good and also its scheduling mechanism guarantees much for similar request from different user. The response time of the request is also had reduced with less delay compared to ant algorithm.

5. Conclusion

A simulation system was developed to test the scheduling algorithm in a simulated grid environment. Experiment results evince its real potential for grid computing. This study is a good attempt to support improved scheduling algorithm in grid job scheduling. In the proposed model QoS factors like availability and delay were concerned and similar request are processed in a better and efficient manner. In the future this algorithm is enhanced for handling QoS factors like jitter.

References

[1] Ian Foster. What is the grid? A three point checklist, GridToday, vol.1, pp.6-12, 2002.

[2] R.Venkatesan, K.Ramalakshmi," Scheduling Framework for Queuing and Advanced Reservation Using TBRA in Grid Computing Environments: Trust Based Resource Agent", 2011 IEEE

[3] S. Gomati, Dr. D. Manimegalai," An adaptive grouping based Job Scheduling in Grid Computing", Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011)

[4] Zhiangwu, Jiecuo,"Dynamic advance reservation for grid system using resource pools",E.Altman and W.Shi(Eds.):NPC 2011,LNCS 6985, pp.123-134.Springer, 2011

[5] Hui Yan, Xue-Qin Shien, Xing Li ,Ming-HuiWu,"An improved ant algorithm for job scheduling in grid computing," Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005

[6] Maria Chtepen, Bart Doedt,Peter Vanrollegem, "Dynamic Scheduling in Grid Systems", Sixth Firw PHD Symposium, Faculty of engineering, Ghent University,30th November 2005,paper nr-110

[7] Siriluck Lorpunmanee, Mohd Noor Sap, Abdul Hanan Abdullah, and Chai Chompoo-inwai," An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment ",International Journal of Computer and Information Engineering 1:8 2007

[8] R. Buyya and M. Murshed, "GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for Grid
computing", The Journal of Concurrency and Computation: Practice and Experience (CCPE), 14:1175-1220, 2002.