Predictive Admission Control Algorithm for Advance Reservation in Equipment Grid

Jie Yin[1], Yuexuan Wang[2], Meizhi Hu[2], Cheng Wu[1]
1. National CIMS Engineering and Research Center, Tsinghua University, Beijing, 100084, China
2. Institute for theoretical computer science, Tsinghua University, Beijing, 100084, China
E-mail: (yinjie05, hmq02)@mails.tsinghua.edu.cn, (wangyuexuan, wuc)@tsinghua.edu.cn

Abstract

Experiments submitted to equipment grid have quality of service (QoS) requirements, and advance reservation is used to satisfy such requirements. Due to the dynamic behaviors and fluctuations of resources in equipment grid, some previously accepted advance reservations are unable to be fulfilled. In this paper, we present a predictive admission control algorithm to decide whether new advance reservation requests can be accepted according to their QoS requirements and prediction of future resource utilization. Historical data are used in this algorithm to predict future status of resources. Experiments demonstrate that our algorithm can reduce the number of accepted advance reservations that fail to be fulfilled and keep the resource utilization ratio at an acceptable level.

1. Introduction

Grid computing, originally designed to be the next generation parallel and distributed computing infrastructure via wide-area sharing of computational resources [1], has evolved to be a mainstream technology enabling large-scale virtual organizations [2]. Grid technologies have the potential to be utilized for cross-domain sharing of many computer integrated resources, e.g. telescopes, observatories [3], and WSNs (wireless sensor networks) [21].

The idea of equipment grid that supports remote access to scientific equipments for education and research has attracted many research interests. Grid technologies are used to connect expensive scientific equipments together for sharing. As a part of China's national grid for education and research, remote manipulation of geographically distributed scientific equipments and cross-organization sharing of high-quality education resources using grid technologies was discussed in [4, 5]. Advance reservation [6] is a mechanism for requesting QoS at present that will take effect in the future. QoS can be defined in terms of job execution time, queue waiting time, data transfer time, CPU workloads and quality of equipments. In some applications, such as video conferencing, multi-player games and remote immersion, advance reservation is used to ensure the required QoS will be available in the future as needed. Many research efforts in the areas like HPC and Internet have focused on advance reservation for resources like CPU, network bandwidth and instruments. Architecture for advance reservation in the Internet is provided in [7]. Negotiation mechanism for advance reservation is studied in [8] and [9]. Applying advance reservation to increase predictability and controllability of future behaviors are given in [10] and [11]. Analysis of the issues of advance reservation and its impact on queue scheduling system are discussed in [12], [13] and [14].

Different from the treatment in HPC in which a centralized resource manager is in charge of the status and availability of resources, advance reservation in grid is a more complex and challenging problem because of resource fluctuation. Heterogeneous resources in grid are accessible under different local resource management policies. In [15], the authors extended a simulation toolkit, GridSim [16], to support advance reservation. Some simulation results given in [17] show the importance of advance reservation in grid. Advance reservation with priorities in grid is studied in [18].

All the above work assumes that once an advance reservation request is accepted, it will definitely be fulfilled. But in practice, it is not always the case. In grid, especially equipment grid, certain special reasons may prevent a confirmed advance reservation from being fulfilled. Examples include resource malfunctions and preemption by more urgent tasks from local schedulers, which are often associated with economic benefits. When confirmed contracts can not be fulfilled, the reputation of the providers of reserved resources will be ruined and the claimed benefits will be affected. The unfulfilment of accepted advance reservations will cause damages both to the clients and to the equipment grid. We propose a predictive admission control algorithm in Section 3 to avoid such situation by refusing some advance reservation requests which may not be fulfilled according to QoS requirements and historical information.

The rest of this paper is organized as follows. In Section 2, resource management issues in equipment grid are introduced. In Section 3, a predictive admission control algorithm is proposed. Section 4 presents some simulation results on our proposed algorithm. In Section 5

978-0-7695-3283-7/08 $25.00 © 2008 IEEE
DOI 10.1109/SCC.2008.13

we present related work on advance reservation. Section 6 concludes this paper and discusses future work.

2. Advance Reservations in Equipment Grid

Equipment grid is an infrastructure that connects geographically distributed equipments together and provides services using grid technologies. The pool model for equipment grid [4] consists of the meta-equipment level, the equipment pool level and the equipment pool alliance level. The equipment pool level consists of equipments with similar functions. Different kinds of equipments pools constitute the equipment pool alliance. Equipments distributed geographically are connected by high-speed networks and agglutinated by specially designed middleware software.

Figure 1 is the illustration of equipment grid.

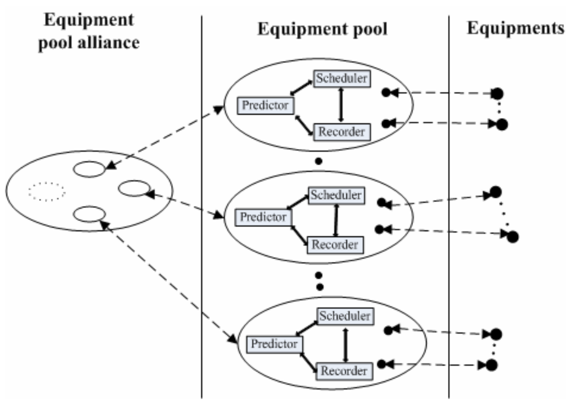

Figure.1 Structure of equipment grid

In figure 1, there is a scheduler, an information recorder and a predictor in every pool. The information recorder stores some important information for future use, such as which equipment had been used and its related scheduling time table. Scheduling time table indicates a future working plan. Predictor can forecast the future status of its pool according to the current status of the system and historical information recorded by the information recorder. The scheduler dispatches accepted advance reservation requests to certain equipments according to information provided by the information recorder and the predictor when the reserved time approaches.

2.1. Resources management

Different from resources like CPU, memory and network bandwidth which can be used in a shared way, resources being reserved in equipment grid are geographically distributed physical equipments which can only be used exclusively. An equipment resource can not run more than one experiment concurrently. The number of equipments that are available in the pools change dynamically. It is also true for WSN, for instance, due to the limitation of power, most sensors are in sleep mode. When users want to use these sensors to collect information, they should reserve these resources in advance. When the reserved time comes, the cluster heads will send activation signals to wake up those related sleeping nodes.

In the equipment grid, when a user wants to reserve equipment for an experiment, he submits his request to the pool alliance. The pool alliance queries the related pools and decides whether to accept or reject this request based on its QoS requirements, prediction of future availability and utilization of resources. The QoS requirements include the costs, the deadline of this experiment and precision of experimental result.

We make the following reasonable assumptions. When an experiment is terminated, intermediate results are not preserved and it will have to be restarted from the beginning. Equipments should not quit from their pool with jobs that are unfinished. An advance reservation request can only reserve one resource. To reserve multiple resources in different pools for more complicated experiments belongs to the co-allocation problem which will be addressed in our future work.

2.2 Advance reservation request

The reservation request consists of what kind of equipment to be reserved, the expected start time and estimated execution time. A user request can often be expressed as *Request* (E, T_s, T_e), where E means what kind of equipment the user needs, T_s is the expected start time and T_e is time duration that the user wants to keep this reservation. Reserved jobs have higher priority than jobs waiting in the queue.

If there are idle resources during the time period [T_s, T_s+T_e] and parameters such as cost and quality of equipment can be satisfied, the advance reservation will be accepted. The difference between traditional advance reservation [17] and predictive advance reservation is that the latter uses a predictive algorithm (introduced in Section 3) to predict the resources available in [T_s, T_s+T_e]. The probability of unfulfilment of accepted advance reservations caused by resource fluctuation will be lower than the former approach when the predictive algorithm is used.

The scheduler in the equipment pool will pre-allocate suitable equipment for an accepted advance reservation request and change the scheduling time table of this pre-allocated equipment.

When the reserved time comes, if the pre-allocated equipment is available, the scheduler will allocate the equipment to perform the experiment as has been requested and reserved. If however the pre-allocated equipment is not available or unwilling to fulfil the reserved request, the scheduler will find another candidate which can satisfy the time and QoS requirements to perform the experiment. If no alternate equipment is available, the reservation is not successful. The system either discards this request or reallocates another resource when there are some suitable equipments available later.

3 Predictive admission control

3.1 Prediction method

In our present design, we predict the future status of the equipment pool using data of the last *n* days. The status of the equipment pool includes the number of equipments providing services in the pool, scheduling tables of equipments, etc. The information recorder will record the status at constant time intervals. The predictor will use the historical information to predict status of equipment pool through equation (1).

$$u_{t,i} = \sum_{k=1}^{n} a_k \times u_{t,i-k} \qquad (1)$$

In equation (1), $u_{t,i}$ means the number of equipments that are providing services at time *t* in a future day *i*. $u_{t,i-k}$ means *k* days before the future day *i*, the number of equipments available at time *t*. a_k is the coefficient of $u_{t,k}$, which satisfies equation (2) and can be adjusted.

$$\sum_{k=1}^{n} a_k = 1 \qquad (2)$$

When the predicted time *t* is between two time intervals, equation (3) is used to estimate the number of equipments that are available.

$$u_{t,i} = \frac{t_1 + \Delta t - t}{\Delta t} u_{t_1,i} + \frac{t - t_1}{\Delta t} u_{t_1+\Delta t,i} \quad (t \in [t_1, t_1 + \Delta t]) \qquad (3)$$

In equation (3), Δt is the constant time interval and *t* is in the region $[t_1, t_1 + \Delta t]$.

The predictive admission control algorithm is used when a pool receives an advanced reservation request from the pool alliance.

3.2 Predictive algorithm

The predictive admission control algorithm (PACA) takes a user request as input parameter and outputs *accept* or *refuse* in response to the user request.

PACA (*Request*)

__Input__: User advance reservation request

__Output:__ Response to user request

// get reserved start time and current time

1) $T_s \leftarrow Request.T_s$
2) $T_e \leftarrow Request.T_e$
3) *currentHour* ← *now.h*
4) *currentDay* ← *now.d*
5) *reservedHour* ← $T_s.h$
6) *reservedDay* ← $T_s.d$

// latestInterval() is to get nearest sampling time

7) *t1* ← *latestInterval* (*currentHour*)
8) *t2* ← *latestInterval* (*reservedHour*)
9) *result* ← "*false*"
10) *temp* ← *t1*

//prediction according to equation (1)~(3)

11) **while** (*temp* < *t2* + Δt)
{
12) *temp* ← *temp* + Δt
13) $u_{temp,\ currentDay} \leftarrow \sum_{k=1}^{n} a_k \times u_{temp,\ currentDay-k}$
}
14) *temp* ← *currentDay*
15) **while** (*temp* < *reservedDay*)
{
16) *temp* ← *temp* + 1
17) $u_{t2,\ temp} \leftarrow \sum_{k=1}^{n} a_k \times u_{t2,\ temp-k}$
18) $u_{t2+\Delta t,\ temp} \leftarrow \sum_{k=1}^{n} a_k \times u_{t2+\Delta t,\ temp-k}$
}
19) $u_{reservedHour,reservedDay} \leftarrow \frac{t_2 + \Delta t - reservedHour}{\Delta t} u_{t_2,reservedDay} + \frac{t - t_2}{\Delta t} u_{t_2+\Delta t,reservedDay}$

// get idle resources in time spot T_s

20) **for** each $timeTable_k$ in *Scheduling_Time_Table*
{
21) **if** ($timeTable_k$ (T_s) == " *occupied* ")
{
22) *occupiedNum++*
}
}

// find available resources during [T_s, T_s+T_e*]*

23) **if** (*occupiedNum* < $u_{reservedHour,\ reservedDay}$)
{
24) **for** each $timeTable_k$ in *Scheduling_Time_Table*
{
25) **if** *timePeriod* ($timeTable_k$, T_s, T_e)=="*vacant*"
{
26) *result* ← "*true*"
27) *allocatedIndex* ← *k*
28) **break**
}
}
}

// decide whether to accept a request or not

29) **if** (*result* == "*true*")
{
30) $timeTable_{allocatedIndex}$ (T_s) ← "*occupied*"
31) *timePeriod* ($timeTable_k$, T_s, T_e) ← " *occupied* "
32) *pre_allocated* (*allocatedIndex*, T_s, T_e , *Request*)

33) *Response* ← "*accept*"
34) **return** *Response*
}
35) **else**
{
36) *Response* ← "*reject*"
37) **return** *Response*
}
End

Function *latestInterval* (*t*) in line 7) and 8) is to return the time when the pool collects its status information such as the number of equipments that are available. *t* is in the region of [*latestInterval* (*t*), *latestInterval* $(t)+\Delta t$] and Δt is the time interval.

Function *timePeriod* ($timeTable_k$, T_s, T_e) in 25) and 31) is to return whether equipment *k* can be used during period [T_s, T_s+T_e] according to $timeTable_k$. $timeTable_k$ is a table that records the future work plan of equipment *k*.

Function *pre_allocated* (*k*, T_s, T_e, *Request*) is to pre-allocate equipment *k* to the user reservation request from time T_s to T_s+T_e when *Request* has been accepted.

Line 1) and 2) are to get the start time and execution time respectively of the reserved request. Line 3) and 4) are to get current time and date. Line 5) and 6) are to get the date and hour of start time. Line 7) and 8) are to find the nearest time when the equipment collects its status information. From line 11) to 19), the status of the equipment pool in reserved time T_s is predicted using the method introduced in Section 3.2. Line 20) to 22) are to check whether there are available equipments in the reserved time. Line 23) to 37) pre-allocate vacant equipment for this reserved experiment.

To improve the precision of the predicted method used in **PACA**, algorithm 2 is used to adjust parameters $a_1 \sim a_n$ according to the difference between the predicted value and the real one.

At the initial phase, $a_1 = \cdots = a_n = 1/n$.

Algorithm 2:
Input: $a_1 \sim a_n$, $predictedU_{t,j}$, $realU_{t,j} \sim realU_{t,j-10}$
Output: $a_1 \sim a_n$
// to decide whether the error is acceptable
(1) **if** ($|predictedU_{t,j} - realU_{t,j}|) > e$
{
// choose a day with least error and increase its weight
(2) **for** each $realU_{t,j-k}$ (k = 1,2,.., n)
{
(3) select *k1* that $|predictedU_{t,j} - realU_{t,j-k1}|$ has the minimum value
}
(4) $s = 0$
// tuning all the weights
(5) **for** k = 1,2..,n
{
(6) **if** $k \neq k1$
{
(7) $a_{n-k} \leftarrow a_{n-k}/2$
(8) $s \leftarrow s + a_{n-k}$
}
}
(9) $a_{n-k1} = a_{n-k1} + s$
}
End

In this algorithm, *e* is a boundary. When the difference between the predicated value $predictedU_{t,j}$ and the actual value $realU_{t,j}$ is beyond the value of *e*, this algorithm will be used to adjust $a_1 \sim a_n$.

4 Simulation results

This section presents some results from a simulation study designed to evaluate the performance of our predicted algorithm as given in Section 3. We compare our algorithm with a traditional admission control algorithm. In the simulation, a process of 1,000 days is simulated with reservation requests arriving randomly according to a Poisson arrival process. For each request, attributes like reservation start time and execution time are defined. An equipment pool with 50 equipments participates in this simulation. The number of equipments fluctuates over time with different probabilities. The execution time of each request is uniformly distributed with a granularity of one hour and within a boundary which is set in the simulation. *n* in equation (1) is defined as 10. For every time interval, algorithm 2 is used to adjust $a_1 \sim a_n$. Up to 10 percent of the resources will quit the equipment pool for a random time between 1 and 10 days.

The metric we adopted for evaluation includes resource utilization, rejection probability and mis-accepted probability.

Figure 2 shows the variation of the rejection probability with the number of advance reservation requests when 10% of the resources in equipment pool fluctuate dynamically. The number of requests ranges from 10 to 100 with an increment of 10. Figure 3 shows the variation of the mis-accepted probability with the number of requests. Figure 4 shows the resource utilization with the number of requests.

In figures 2, 3 and 4, the rejection probability is a bit higher when the predictive admission control algorithm is used. This is because when an advance reservation is submitted, the pool will predict the future status before accept it. Often the resources available are less than the total service ability of the pool. The traditional admission control algorithm does not take this difference into account and may accept some requests that can not be fulfilled in the future, which is demonstrated in Figure 3 where higher mis-accepted probability can be seen. In Figure 4, the resource utilization is a little lower when our algorithm is applied.

Figures 5, 6 and 7 are similar to Figures 2, 3 and 4 except that the resource fluctuation probability is 50% instead of 10%. When the fluctuation probability increases from 10% to 50%, all measures in both algorithms would increase.

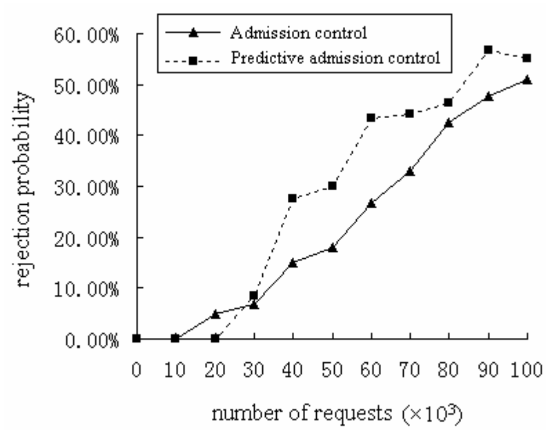

Figure 2. Rejection probability vs requests

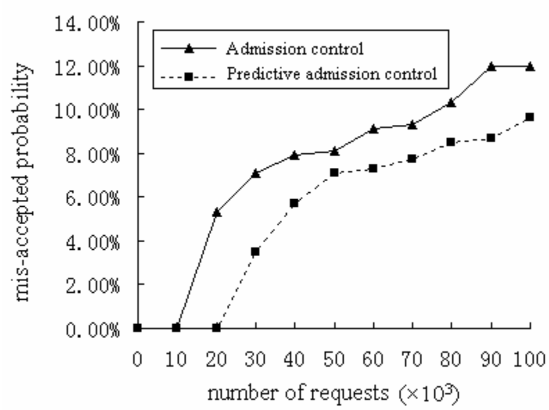

Figure 3. Mis-accepted probability vs requests

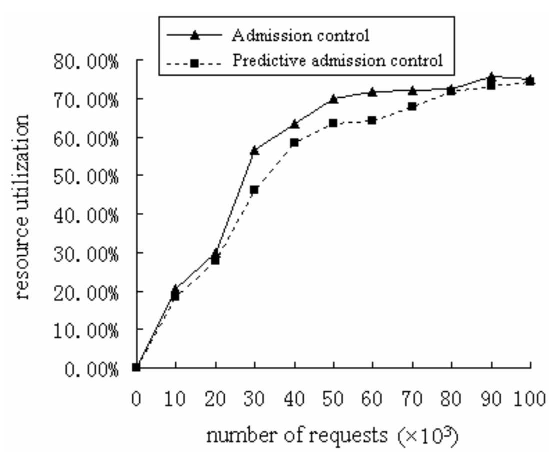

Figure 4. Resource utilization vs requests

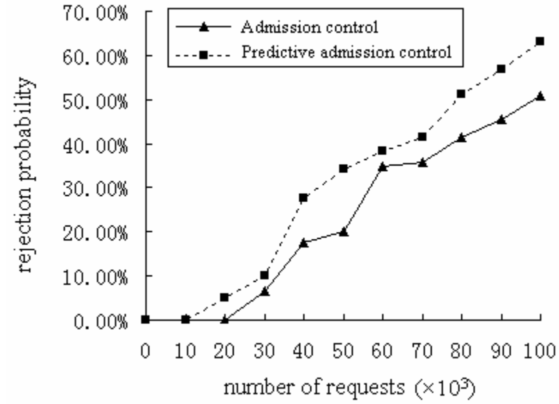

Figure 5. Rejection probability vs requests

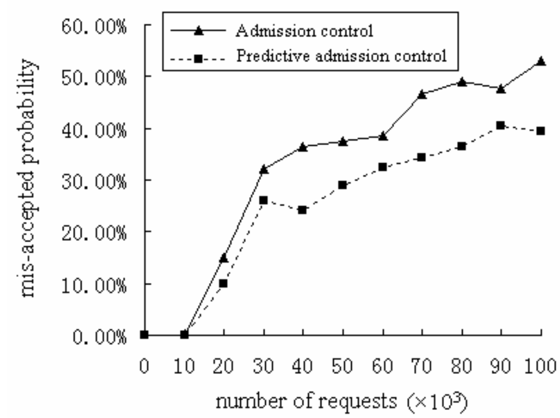

Figure 6. Mis-accepted probability vs requests

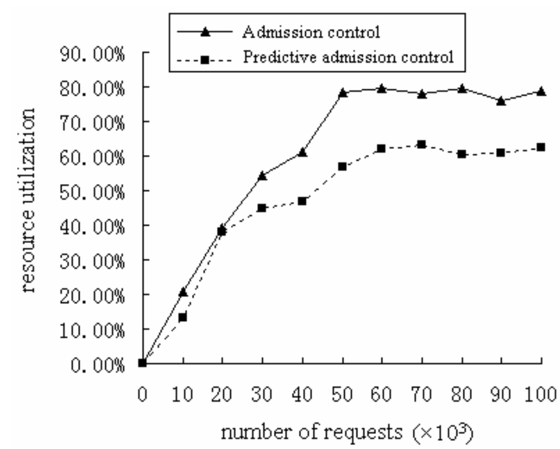

Figure 7. Resource utilization vs requests

Figures 8 is the variation of the mis-accepted probability when execution time ranges from 2 to 20 hours. Figure 9 is the variation between resource utilization with execution time. These results are obtained when 50,000 advanced requests are submitted and half of the resource fluctuates. When the execution time extends, both the mis-accepted probability and the resource utilization are increased to a balance level. The high mis-accepted probability is due to the fact that with the increased execution time, few resources can provide alternative services when the former pre-allocated resources are unavailable.

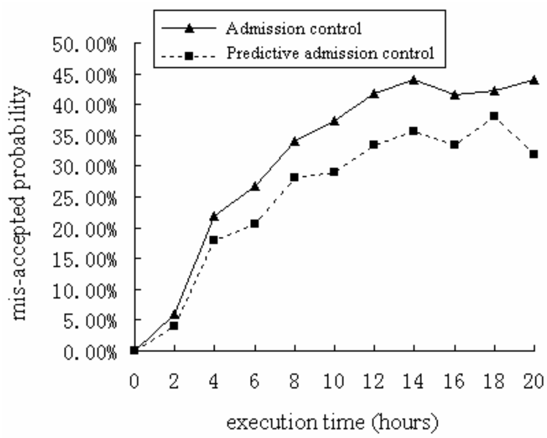

Figure 8. Mis-accepted probability vs requests

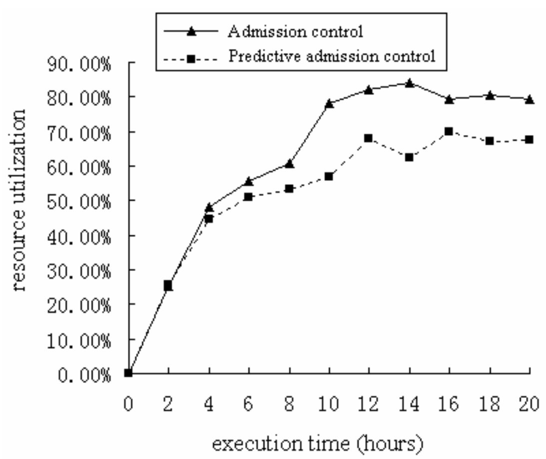

Figure 9. Resource utilization vs requests

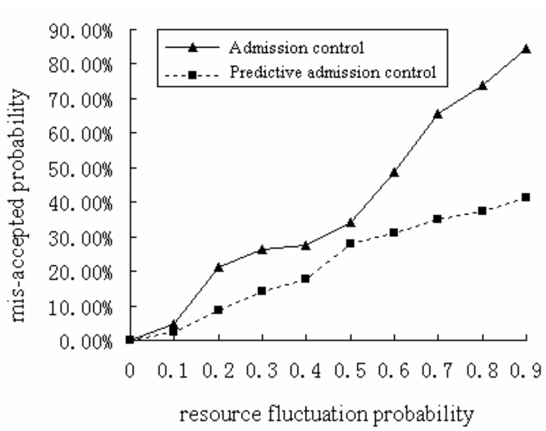

Figure 10. Mis-accepted probability vs resource fluctuation

Figure 10 shows the relationship between the mis-accepted probability and the resource fluctuation probability when 50,000 advanced requests with execution time uniformly distributed between 1 to 10 hours are submitted. With the increment of resource fluctuation, the mis-accepted probability increases sharply under the traditional admission control algorithm but stays within an acceptable scope with our algorithm. We can also see that when the resource fluctuation probability is high, which often occurs in equipment grid systems, our predictive algorithm has a much lower mis-accepted probability.

5. Related Work

In the past few years, some literature has discussed the need for advance reservation and admission control algorithm. However, to our knowledge, little work has been done on the problem of some accepted advance reservation requests being not fulfilled.

Wolf et al. [12] provide a model for resource reservation in advance. In their work, t is defined as the difference between $t_{reserve}$ and t_{start}. Only when t is larger than t_{min} and less than t_{max} can requests be accepted for the sake of minimizing the negative impact of advance reservation on the resource management system.

Cao and Zimmermann present in [14] the influence of advance reservation on FCFS queue scheduling policy. Their experiments show that a large percentage of reservation requests will sacrifice queue scheduling efficiency. Burchard analyzes in [13] problems that are caused by the import of advance reservation and suggests malleable reservation to prevent deteriorated performance.

Rui and Muthucumaru propose in [18] a benefit function based on a renegotiation mechanism. They introduce a novel way of incorporating QoS constraints and priority into an advance reservation scheduling algorithm in grid computing systems.

When users submit their advance reservation requests, the reserved time duration according to their job execution time should be explicitly reserved. But users may have no idea of how long their jobs will take. In [19] and [20], some methods to predict the application runtimes are provided.

6. Conclusions

This paper introduces a predictive admission control algorithm for advance reservation in equipment grid. We compare the performance of our proposed algorithm with a traditional admission control algorithm via simulation. The mis-accepted request problem in advance reservation can be quite serious when an experiment needs the participation of multiple resources. The simulation results show a tradeoff between the mis-accepted probability, the request rejection probability, and resource utilization.

Several directions can be identified for future investigation, including: 1) an economic model for the mis-accepted request problem and the method to minimize its impact, and 2) extension of this predicted algorithm to deal with the case where multiple resources are needed.

7. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China Grant 60553001, 60604033, the National Basic Research Program of China Grant 2007CB807900, 2007CB807901 and Hi-Tech Research & Development Program of China Grant 2006AA10Z216.

8. References

[1] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan-Kaufmann, 1998.

[2] I. Foster, C. Kesselman and S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”, *International Journal of Supercomputer Applications*, 15(3), 2001, pp.200-222.

[3] NSF Cyberinfrastructure Council, *NSF’s Cyberinfrastructure Vision for 21st Century Discovery*, Version 5.0, January 20, 2006.

[4] Y. Wang, C Wu, “Study on Equipment Interoperation Chain Model in Grid Environment”, Lecture Notes in Computer Science, Vol.3758, 2005, pp.588-595

[5] Y. Wang and C. Wu, “A Study on Education Resource Sharing Grid”, *International Journal of Information Technology, Special Issue on Grid Computing I*, 11(3), 2005, pp.73-80.

[6] Lars C. Wolf, R Steinmetz, “Concepts for Resource Reservation in Advance”, Journal of Multimedia and Applications 4(3), 1997, pp.255-278

[7] S Berson, R Lindell and R Braden, “An Architecture for Advance Reservations in the Internet”, Technical report, USC Information Science Institute, 1998, at http://citeseer.ist.psu.edu/cache/papers/cs/18982/http:zSzzSzwww.isi.eduzSz~bersonzSzadvance.pdf/berson98architecture.pdf

[8] Mz Siddiqui, A Villazon and T Fahringer, “Grid Capacity Planning with Negotiation-based Advance Reservation for Optimized QoS”, in SuperComputing Conference. 2006. Tampa, Florida.

[9] A. Hafid, G. Bochmann and R. Dssouli, A quality of service negotiation approach with future reservations (NAFUR): a detailed study. Computer Networks and ISDN Systems 30(8), 1998, pp.779–794.

[10] M Wieczorek, M Siddiqui and A Villazon et al. “Applying Advance Reservation to Increase Predictability of Workflow Execution on the Grid”. Second IEEE International Conference on e-Science and Grid Computing. 2006. Amsterdam.

[11] M. Degermark, T. Kohler, S. Pink and O. Schelen, “Advance reservations for predictive service in the Internet. ACM/Springer Journal of Multimedia Systems”, 5(3), 1997, pp.77–186

[12] L. C. Wolf, L. Delgrossi, R. Steinmetz and S. Schaller etc, “Issues of reserving resources in advance,” Proc. 5th Int. Workshop Network and Operating System Support for Digital Audio and Video Durham, NH, 1995, pp.27-37

[13] L.-O. Burchard, “On the Performance of Computer Networks with Advance Reservation Mechanisms,” Proc. 11th IEEE Int Conf Networks (ICON ’03), 2003, pp.449-454

[14] Cao, J. and F. Zimmermann. “Queue scheduling and advance reservations with COSY”. In Proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS 2004), 2004.

[15] A. Sulistio and R. Buyya, “A grid simulation infrastructure supporting advance reservation”, in Proc. of the 16th International Conference on Parallel and Distributed Computing and Systems (PDCS'04), Cambridge, USA, 2004.

[16] R. Buyya and M. Murshed, “GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing”, J. of Concurrency and Computation: Practice and Experience, 14(13–15), 2002, pp.1175–1220.

[17] W. Smith, I. Foster, and V. Taylor, “Scheduling with Advanced Reservations”, In Proc. of the Int. Parallel and Distributed Processing Symposium (IPDPS), 2000, pp.127–132.

[18] R Min, M Maheswaran, “Scheduling Advance reservation with priorities in grid computing system”, International Conference on Parallel And Distributed Computing Systems(PDCS'01), 2001, pp.172-176

[19] S. Krishnaswamy, A. Zaslasvky, and S.W. Loke, "Predicting Application Runtimes Using Rough Sets," Proc. 9th Int'l Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems, 2002

[20] W. Smith, I. Foster, and V. Taylor, "Predicting Application Runtimes Using Historical Information", LNCS 1459, 1998, pp.122-142

[21] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam etc, “Wireless sensor networks: a survey”, Computer Networks, 38, 2002, pp.393-422