A predictive admission control algorithm for user advance reservation in equipment grid

Yuexuan Wang* and Meizhi Hu

Institute for Theoretical Computer Science,
Tsinghua University,
Beijing, 100084, China
E-mail: wangyuexuan@tsinghua.edu.cn
E-mail: humeizhi@tsinghua.edu.cn
*Corresponding author

Jie Yin

National CIMS Engineering and Research Center,
Tsinghua University,
Beijing, 100084, China
E-mail: yinjie05@tsinghua.edu.cn

Abstract: In a service computing world, a user submits job requests to an equipment grid which often has quality of service (QoS) requirements. The advance reservations of equipment grid are used to satisfy such requirements. Due to the dynamic behaviours and fluctuations of resources in equipment grid, some accepted advance reservations are not able to be fulfilled in different contexts. Therefore, we propose a predictive admission control algorithm to decide whether the new advance reservation requests can be accepted according to their QoS requirements and prediction of future resource utilisation. Historical data are used in this algorithm to predict future status of resources in the equipment grid. The simulation experiments demonstrate that our algorithm can reduce the number of accepted advance reservations that fail to be fulfilled, and can keep the resource utilisation ratio at an acceptable level.

Keywords: equipment grid; predictive admission control; advance reservation; scheduling; wireless sensor network; WSN.

Reference to this paper should be made as follows: Wang, Y., Hu, M. and Yin, J. (2009) 'A predictive admission control algorithm for user advance reservation in equipment grid', *Int. J. Business Process Integration and Management*, Vol. 4, No. 4, pp.274–283.

Biographical notes: Yuexuan Wang received her PhD from Zhejiang University in March 2003. She was employed as a Postdoctoral Fellow in Department of Automation of Tsinghua University from April 2003 to February 2006. From March 2006 she began her research work as an Assistant Professor in Tsinghua University. In June 2006 she became an Associate Professor of Tsinghua University. In December 2009 she became Professor of Tsinghua University. She is also the Deputy Director of Institute for Theoretical Computer Science, Tsinghua University. Her research interests include grid computing, wireless sensor network, quantum computing, etc.

Meizhi Hu received her PhD from Tsinghua University in 2008; then joined the Institute for Theoretical Computer Science, Tsinghua University. Her research interests include distributed computing, grid monitoring, etc.

Jie Yin received his Masters degree from the Institution of Beijing Machinery in 2005, and PhD from Tsinghua University in 2009. Now he works at Shen Zhen Entry-Exit Inspection & Quarantine Bureau.

1 Introduction

Equipment grid is becoming more popular and important in supporting different business applications nowadays. Originally, the grid computing is designed to be the next generation parallel and distributed computing infrastructure via wide-area sharing of computational resources (Foster and Kesselman, 1998). It has evolved to be a mainstream technology for enabling large-scale virtual organisations (Foster et al., 2001). The grid technologies have the potential to be utilised for cross-domain sharing of many computer integrated resources, e.g., telescopes, observatories (NSF Cyberinfrastructure Council, 2006), and the wireless sensor networks (WSNs) (Akyildiz et al., 2002). The main objective of this paper is to improve the performance of advance reservation by reducing the system unfulfilment events according to the quality of service (QoS) requirements and the historical information in the equipment grid. Based on our observations, several research results (Roy and Sander, 2003; Foster et al., 2000) assume that once an advance reservation request is accepted, it will be fulfilled. Technically, this is not always the case. In the grid environment, especially equipment grid, the system unfulfilment situations may prevent an accepted advance reservation from being fulfilled. The reasons that cause system unfulfilment include the resource malfunctions and pre-emption by more urgent tasks from the local schedulers, which often are associated with the economic benefits. Furthermore, the needs of several resources and sharing of equipment among users at the same time must be organised. The occurrence of system unfulfilment situations divided by the total number of accepted user advance reservation requests is called unfulfilment probability of the equipment grid. When confirmed contracts cannot be fulfilled, the reputation of providers of such reserved resources will be ruined and their benefits will be affected. The unfulfilment of accepted advance reservations will cause damages both to the clients and the equipment grid. Thus, we propose a predictive admission control algorithm (PACA) in Section 3, which can estimate the priorities of those advance reservation requests and manage the resource sharing which may not be fulfilled according to the QoS requirements and historical information.

The QoS is extremely important for computing and collaboration grids because the service oriented architecture (SOA) and web service originated in the grids environment. The QoS of grid emerges due to a large amount of users (Plestys et al., 2007). The necessity of QoS of the grid is based on three main aspects: the concept of grid service, the multiplicity of users' demands and the heterogeneity of grid resources (Juhua et al., 2003). Thereby our algorithm attacks the problems of the availability of grid resource and also the advanced reservation of the equipment grid. The idea of building equipment grid that supports remote access to the scientific equipments for education and research has attracted much attention in China. The grid technologies are used to connect the expensive scientific equipments together for work and share. As a part of China's national grid for education and research, remote manipulation of geographically distributed scientific equipments and cross-organisation sharing of high-quality education resources using grid technologies was discussed in Wang and Wu (2005a, 2005b).

An advance reservation (Wolf and Steinmetz, 2004) is a mechanism for requesting the QoS at present that will take effect in the future. The QoS can be defined in terms of job execution time, queue waiting time, data transfer time, central processing unit (CPU) workloads and the quality of equipments. In some applications, such as video conferencing, multi-player games and remote immersion, an advance reservation is used to ensure the required QoS will be available in the future when needed. Many research efforts in the areas of high performance computing (HPC) and internet have focused on the advance reservation for resources like CPU, network bandwidth and so on. The architecture for the advance reservation in the internet is provided in Berson et al. (1998). The negotiation mechanism for the advance reservation is studied in Siddiqui et al. (2006) and Hafid et al. (1998). Applying advance reservation to increase predictability and controllability of the future behaviours is given in Wieczorek et al. (2006) and Degermark et al. (1997). The analysis of issues of the advance reservation and its impact on queue scheduling system are discussed in Wolf et al. (1995), Burchard (2003) and Cao and Zimmermann (2004).

Different from the treatment in HPC in which a centralised resource manager is in charge of the status and availability of resources, the advance reservation in grid is a more complex and challenging problem because of the resource fluctuation and the distributed resource management strategies. The heterogeneous resources in grid are accessible under the different local resource management policies. In Sulistio and Buyya (2004), the authors extended a simulation toolkit called the GridSim (Buyya and Murshed, 2002), to support the advance reservation. Some simulation results given in Smith et al. (2000) show the necessities of advance reservation in grid. The advance reservation with priorities in grid is studied in Min and Maheswaran (2001).

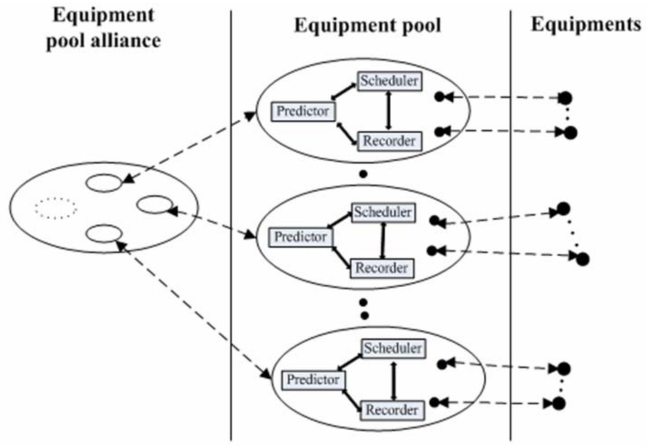
The details of our purposed algorithm and some related works are described in the rest of this paper which is organised as follows. In Section 2, resource management issues in equipment grid are introduced. In Section 3, a PACA is proposed. Section 4 presents simulation results on our proposed algorithm. In Section 5 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 to conduct experiments using grid technologies. In Wang and Wu (2005b), the equipment pool model, which consists of the meta-equipment level, the equipment pool level, and the equipment pool alliance level, is provided. The equipment pool level consists of the equipments with similar functions. For example, all kinds of microscopes that are available in the equipment grid form the microscope pool. The different kinds of equipments pools constitute the equipment pool alliance. The equipments distributed geographically are connected by high speed networks and agglutinated by specially designed middleware software.

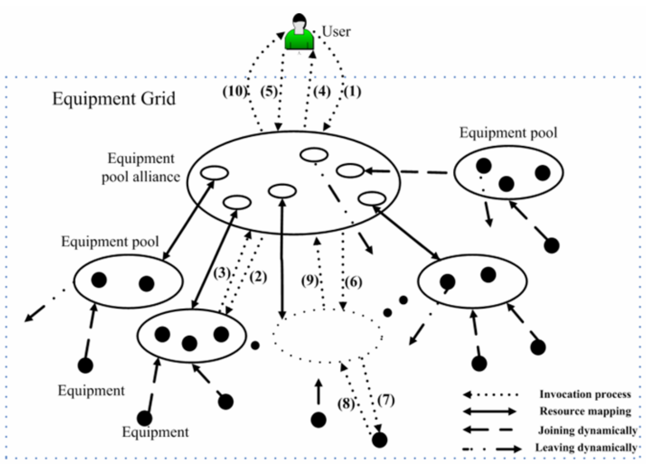
Figure 1 is the illustration of equipment grid, where there is a scheduler, an information recorder, and a predictor in every pool. The information recorder stores some important information. For instance, which equipment had been used, who used it, how many times it had been used, the regular execution time of it, and its related scheduling time table. The scheduling time table indicates the future work plan of equipments. A 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. A scheduler dispatches accepted the advance reservation requests to certain equipments according to information provided by the information recorder and the predictor when the reserved time comes.

Figure 1 The structure of equipment grid (see online version for colours)



Referring to Figure 1, the dash lines in the map indicate the distributed equipments to the equipment pool and equipment pools to the equipment pool alliance. The equipment can join and quit its belonging equipment pool dynamically. Every pool can also join and quit pool alliance dynamically. Figure 2 describes the detailed process when user submits an advance reservation request for his experiment to the end of his experiment.

Figure 2 Demonstration of user advance reservation process in equipment grid (see online version for colours)



In Figure 2, when a user wants to reserve an experiment in advance, the user first submits his/her advanced reservation request to the equipment pool alliance in the point (1). The equipment pool alliance analyses this request and submits it to the related equipment pool in the point (2). When the related equipment pool receives a request, the predictor in it will first estimate the status of this pool in the reserved period and decide whether to accept this advanced request with the historical information provided by the recorder. The equipment pool transfers this request to the equipment pool alliance in the point (3). If the reservation request is accepted, the scheduler in the pool will pre-allocate a resource for this request. In the point (4), the pool alliance forwards the point (3) to the user. If the user advanced request is accepted and when the reserved time comes, the user submits his experiment with necessary material in the point (5) and the pool alliance forwards this experiment to related pool in the point (6). The pool will allocate this experiment by scheduler in the point (7). After the experiment is finished, the equipment that running this task returns experimental result to its pool in the point (8). The related pool reports to the pool alliance in the point (9) and the pool alliance informs the user in the point (10).

The PACA is used when a pool receives an advanced reservation request from the pool alliance. In Section 3, we will discuss this algorithm in detail.

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 equipment that can only be used exclusively. A resource in an equipment pool cannot run more than one experiment concurrently. The number of resources that are available in the pools may change dynamically. It is also the case for WSN. For instance, due to the limitation of power, most wireless sensors are in sleep mode. When users want to wake up and use these nodes 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 the sleeping nodes. Then these working nodes transfer the data they collected to the cluster heads. After the data aggregation, the cluster heads transfer these data to user.

In the equipment grid, when a user wants to reserve resources for his/her experiment, the user submits his/her request to the pool alliance. The pool alliance queries the related pools and decides whether to accept or refuse this request based on the QoS requirements and prediction of future availability, and utilisation of resources. The QoS requirements include the costs, the deadline of the experiments and the precision of the results.

In this paper, we make the following assumptions. When an experiment is terminated, the intermediate results are not saved and it will have to be restarted from the beginning. The equipment should not withdraw from their pool with a job that is unfinished. An advance reservation request can only reserve one resource. To reserve multiple resources in the different pools for 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 be expressed as request (E, T_s, T_e), where E means what kind of equipment the user wants to reserve, T_s is the expected start time and T_e is the expected execution time. Reserved jobs often have a higher priority than the jobs waiting in the queue without an advance reservation.

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.

Different from traditional admission control algorithm for advance reservation (Smith et al., 2000), the proposed PACA for an advance reservation uses the predictive algorithm introduced in Section 3 to predict the number of resources that are available in the related equipment pool and their states during [Ts, Ts + Te]. The unfulfilment probability of accepted advance reservation caused by resources fluctuation will be lower than the traditional approach when the predictive algorithm is used.

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

When the reserved time comes and the pre-allocated equipment is online and available, the scheduler will allocate the pre-allocated resource to perform the experiment as has been requested and reserved. However if the pre-allocated equipment is not available or unwilling to fulfil the reserved request, this is the situation of the system unfulfilment. In this case, the scheduler will find another candidate resource which can satisfy the time and QoS requirements to perform this experiment. If there is no alternate resource available, the system either discards this experiment request, reallocates another resource when there is suitable resource available or asks the user to modify his former experiment request and submit another advance reservation request.

3 Predictive admission control

3.1 Prediction method

In the design proposed, 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 resources that are providing services in the pool, scheduling tables and trust tables of these resources, etc. The information recorder will record the status of its pool at constant time intervals. The predictor will use historical information to predict statuses through equation (1).

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

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

$$\sum_{k=1}^{n} a_k = 1 \tag{2}$$

When the time t is between two time intervals, equation (3) is used to predict the number of resources available.

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

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

$$u'_{t,i} = trust \times u_{t,i} \tag{4}$$

In equation (4), $u_{t,i}$ is the corrected to $u'_{t,i}$ by taking the reliability of resources into account. The parameter *trust* is number calculated from equation (5).

$$trust = \sum_{i=1}^{N} trust\_factor_i \tag{5}$$

In equation (5), $trust\_factor_i$ is a parameter that reflects the reliability of resource i. N is the number of resources that are available in the related equipment pool.

The PACA is used when equipment pool receives a user advanced reservation request dispatched by pool alliance.

3.2 Predictive algorithm

The PACA takes user request as input parameter and outputs accept or refuse in response to the request.

Algorithm 1: PACA *(Request)*

Input: User advance reservation request

Output: Response to user request

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 // get reserved start time and current time*

7) *t1* ← *latestInterval* (*currentHour*)

8) *t2* ← *latestInterval* (*reservedHour*) *// latestInterval() is to get nearest sampling time*

9) *result* ← *'false'*

10) *temp* ← *t1*

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) $u_{reservedHour,reservedDay} \leftarrow \frac{t_2 + \Delta t - reservedHour}{\Delta t} u_{t_2,reservedDay} + \frac{t - t_2}{\Delta t} u_{t_2+\Delta}$

//prediction

20) for each *i* < *N // N is the number of resources currently available.*

{

21) *trust* = *trust* + $trust\_factor_i$

}

22) *trust* = *trust* / *N*

23) $u_{reservedHour,reservedDay} = trust \times u_{reservedHour,reservedDay}$

24) for each $timeTable_k$ in *Scheduling_Time_Table*

{

25) if ($timeTable_k$ (T_s) == "*occupied"*)

{

26) *occupiedNum*++

}

} *// get idle resources in time spot* T_s

27) if (*occupiedNum* < $u_{reservedHour,\ reservedDay}$)

{

28) for each $timeTable_k$ in *Scheduling_Time_Table*

{

29) if *timePeriod* ($timeTable_k$, T_s, T_e) == "*vacant"*

{

30) *result* ← "*true*"

31) *allocatedIndex* ← *k*

32) break

}

}

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

33) if (*result* == "*true*")

{

34) $timeTable_{allocatedIndex}$ (T_s) ← "*occupied"*

35) *timePeriod* ($timeTable_k$, T_s, T_e) ← "*occupied"*

36) *pre_allocated* (*allocatedIndex,* T_s, T_e , *Request*)

37) *Response* ← "*accept*"

38) return *Response*

}

39) else

{

40) *Response* ← "*refuse*"

41) return *Response*

} *// decide whether to accept a request or not*

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*) + Δt] and Δt is the time interval.

Function *timePeriod* ($timeTable_k$, T_s, T_e) in 29) and 35) is to return whether equipment *k* can be used during period [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, *Te*, *Request*) is to pre-allocate equipment *k* to the user reservation request from time T_s to T_e when *Request* has been accepted.

Lines 1) and 2) are to get the start time and execution time respectively of the reserved request. Lines 3) and 4) are to get current time and date. Lines 5) and 6) are to get the date and hour of start time. Lines 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 23) is to corrected the predicted value using the reliability parameter of resources. The value of reliability parameter $trust\_factor_i$ is adjusted by Algorithm 3. Lines 24) to 26) are to check whether there are available equipments in the reserved time. Line 27) to 41) pre-allocate vacant equipment for this reserved experiment.

To improve the preciseness 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, $a1 = \ldots = a_n = 1/n$.

Algorithm 2:

Input: $a_1 \sim a_n$, $predictedU_{t,j}$, $realU_{t,j} \sim realU_{t,j-10}$

Output: $a_1 \sim a_n$

(1) if (| $predictedU_{t,j}$ - $realU_{t,j}$ |) > *e // to decide whether the error is acceptable*

{

(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

} *// choose a former day with least error and increase its weight*

(4) *s* = 0

(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) $a_{n-kl} = a_{n-kl} + s$ // *tuning all the weights*

}

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 this value, this algorithm will be used to adjust $a_1 \sim a_n$.

Algorithm 3:

if (*available* (*pre_allocatedResource* (*Request*)) == "*false*")

{

trust_factor [*pre_allocatedResource* (*Request*)] / = 2;

}

else

{

temp = *trust_factor* [*pre_allocatedResource* (*Request*)] + *increment*

if (*temp* > 1)

{

trust_factor [*pre_allocatedResource* (*Request*)] = 1

}

else

{

trust_factor [*pre_allocatedResource* (*Request*)] = *temp*

}

}

End

4 Simulation results

This section presents some results from simulation study designed to evaluate the performance of our predicted admission control algorithm given in Section 3. We compare our algorithm with a traditional admission control algorithm. In this simulation, a process of 1,000 days is simulated with user reservation requests arriving randomly that follow a Poisson arrival process. For each request, the attributes, such as reservation start time and expected execution time, are defined before the execution. An equipment pool with 50 equipments is participated 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 to 10. For every time interval, Algorithm 2 is used to adjust $a_1 \sim a_n$. Up to 10% of the resources will quit the equipment pool for a random time period between one and ten days.

The metrics we adopted for evaluation include resource utilisation, rejection probability and unfulfilment probability.

$$\textit{rejection probability} = \frac{\text{(number of rejection)}}{\text{(number of requests)}}$$

$$\textit{unfufilment probability} = \frac{\text{(number of accepted request not be fulfilled)}}{\text{(total number of accepted requests)}}$$

Figure 3 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 4 shows the variation of the unfulfilment probability with the number of requests. Figure 5 shows the resource utilisation with the number of requests.

Figure 3 Rejections probability vs. requests

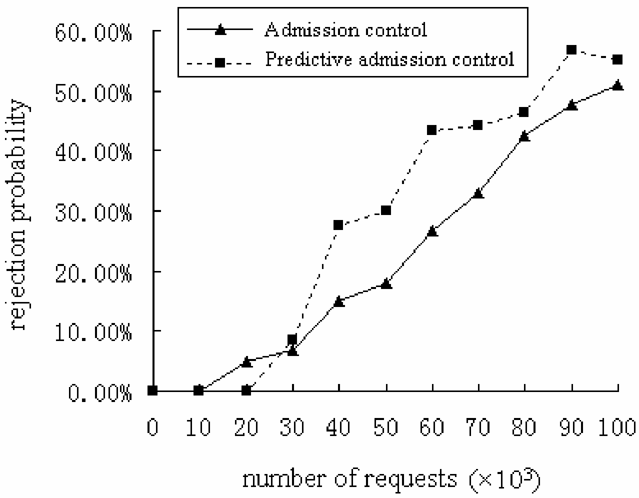
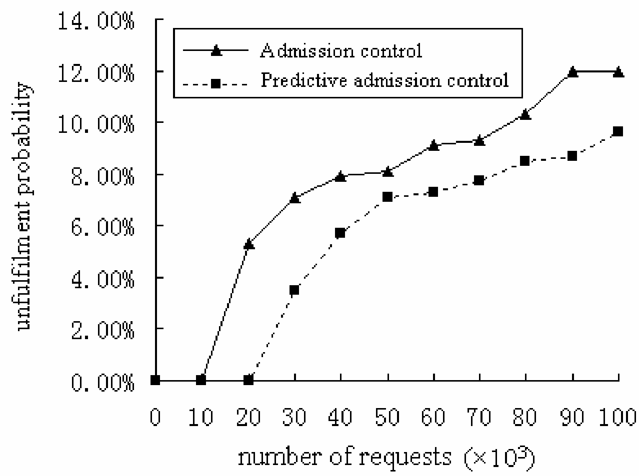


Figure 4 Unfulfilment probability vs. requests



In Figures 3, 4 and 5, the rejection probability is a bit higher when the PACA is used. This is because when an advance reservation is received, the pool will predict the future status before accepting 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 cannot be fulfilled in the future, which is demonstrated in Figure 3 where higher unfulfilment probability can be seen. Referring to Figure 4, the resource utilisation is a little lower when our algorithm is applied.

Figure 5 Resource utilisation vs. requests

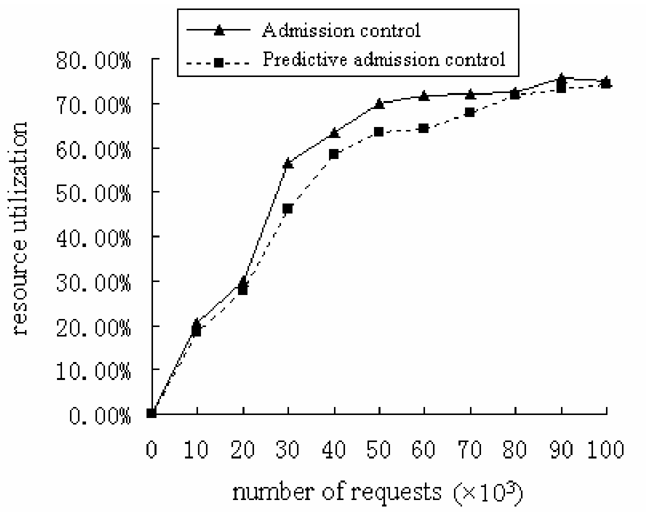


Figure 6 Rejections probability vs. requests

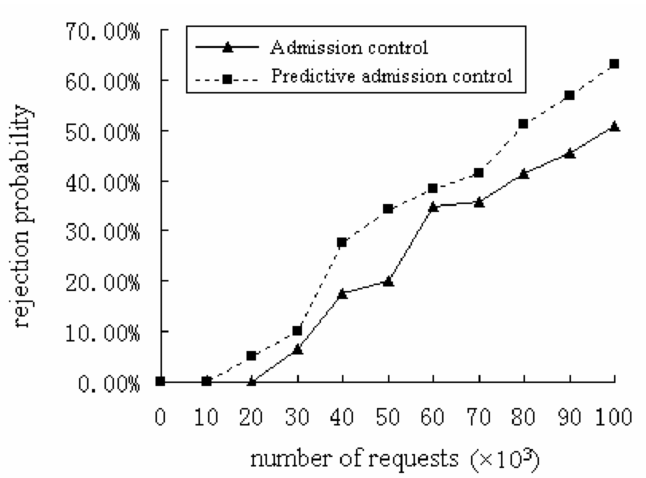
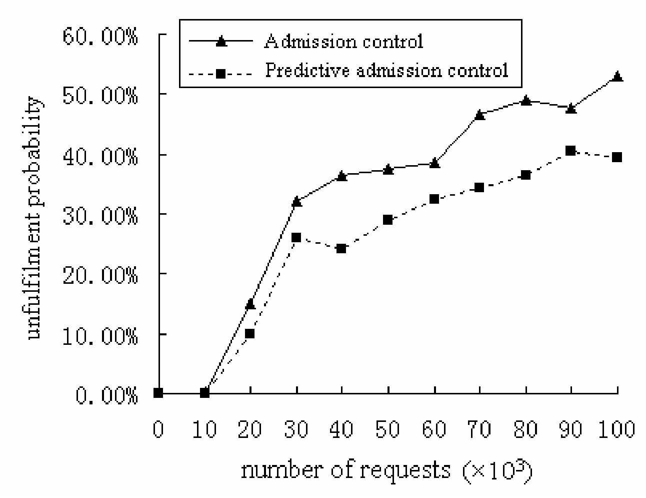


Figure 7 Unfulfilment probability vs. requests



Figures 6, 7 and 8 are similar to Figures 3, 4 and 5 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 8 Resource utilisation vs. requests

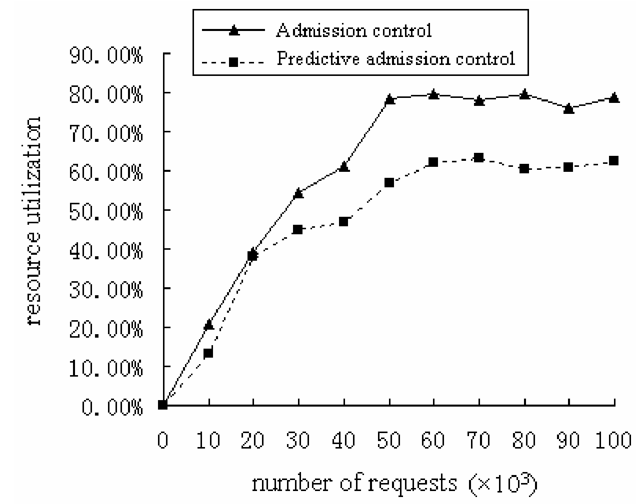


Figure 9 Unfulfilment probability vs. requests

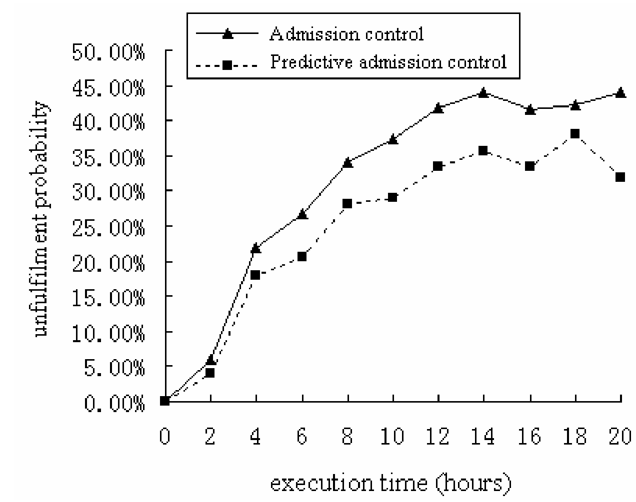
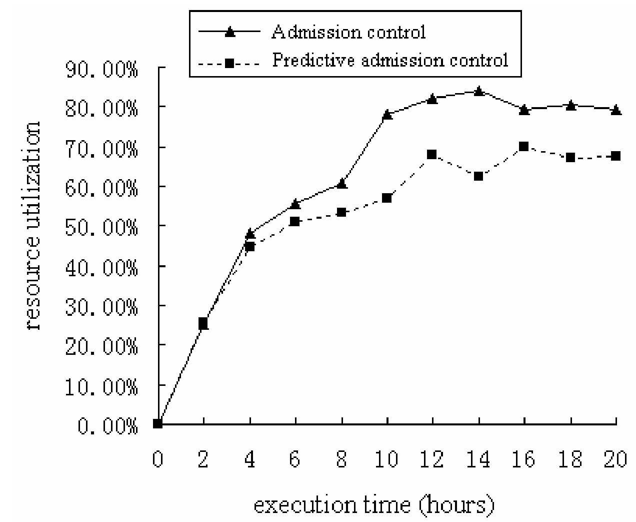


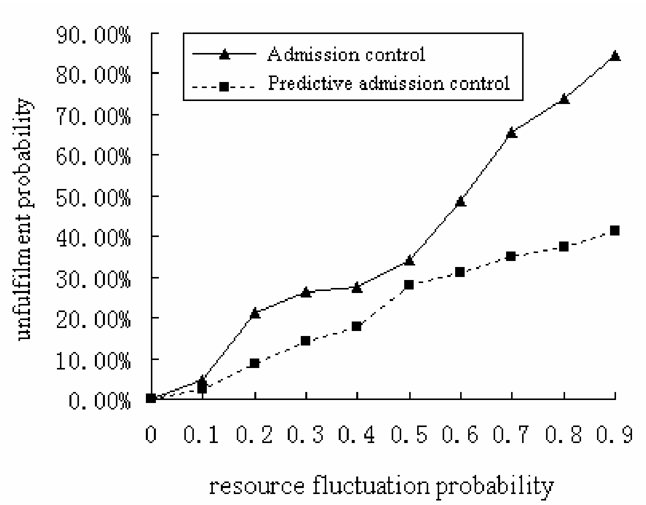
Figure 10 Resource utilisation vs. requests



Figures 9 and 10 show the variation of the unfulfilment probability and the resource utilisation with the execution time that ranges from two to 20 hours, respectively. These results are obtained when 50,000 advanced requests are submitted and half of the resource would fluctuate. When the execution time extends, both the unfulfilment probability and the resource utilisation are increased to a balance level. The high unfulfilment probability is due to the fact that with the increased execution time, few resources can provide alternative services when the pre-allocated resources are unavailable.

Figure 11 shows the relationship between the unfulfilment probability and the resource fluctuation probability when 50,000 advanced requests with execution time uniformly distributed between one to ten hours are submitted. It can be seen that with the increase of resource fluctuation, the unfulfilment probability increases sharply with the traditional admission control algorithm while 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 unfulfilment probability.

Figure 11 Unfulfilment probability vs. resource fluctuation probability



5 Related work

In the past few years, some literature has discussed the need for advance reservation and admission control algorithm. However, a small number of research works have been done on the problems of some accepted advance reservation requests not being fulfilled.

Wolf et al. (1995) propose a model for resource reservation in advance. In their work, *t* is defined as the difference between *treserve* and *tstart*. Only when *t* is larger than *tmin* and less than *tmax* can requests be accepted for the sake of minimising the negative impact of advance reservation on the resource management system.

Further, Curti et al. (2005) discuss the problem of network resource management on grids called the general-purpose architecture for reservation and allocation (GARA) (Roy and Sander, 2003; Foster et al., 2000). The GARA addresses the issue of advance reservation by proposing a framework which can be applied to the different resource types such as computing, storage and networking. Resource management is integrated with a variety of general-purpose grid services (e.g., for authentication and authorisation). However, the GARA does not support resource discovery. Therefore, a grid customer cannot dynamically discover the entire range of available network reservation services. This problem can be avoided by our proposed algorithm.

Cao and Zimmermann (2004) present 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 (2003) analyses problems that are caused by the import of advance reservation and suggests malleable reservation to prevent deteriorated performance. In addition, Rui and Muthucumaru (2001) propose 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. In Wieder et al. (2007), the scheduling technique for reservation and access management are proposed to compute and manage resources for the grid environments. Battestilli et al. (2007) also focus on the reservation and resource sharing problems by building an optical grid. The grid applications can request dynamically in-advance or on-demand any types of grid resources not only high-performance computers, but also deterministic, high-bandwidth network paths (Battestilli et al., 2007). Following these research works, the simultaneous coordination of all resources for advance reservation within a grid environment still is their challenge.

In Barz et al. (2007), the MetaScheduling Service (MSS) (Waldrich et al., 2006) is responsible for the negotiation agreements on resource usage with the local resource management systems. The agreements are made using the WS-agreement (Andrieux et al., 2007) developed by grid resource allocation agreement protocol (GRAAP) and these agreements are called the service level agreements. Extending this approach to network resources was tested in vertically integrated optical testbed (VIOLA, 2005) which allows user or application driven selection and reservation of network connections with dedicated QoS based on evolving network technologies (Barz et al., 2007). Our algorithm is possible to integrate with this approach such that it can improve the reservation and resource management system by considering not only the agreements but the historical information and the QoS requirements also. Later, Kravtsov et al. (2008) develop the QosCosGrid architecture for a quasi-opportunistic supercomputer and they reported the experimental results derived from studying and identifying the requirements a grid needs in order to facilitate a quasi-opportunistic supercomputing.

Regarding time constraints, when users submit their advance reservation requests, the reserved time duration according to their jobs' execution time should be explicitly reserved. However users may have no idea of how long their jobs will take. In Krishnaswamy et al. (2002) and Smith et al. (1998), some methods to predict the application runtimes are provided. In Dinda and Hallaron (1999, 2000), the prediction performance of the Box-Jenkins five models, which are autoregressive (AR), moving average (MA), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA) and autoregressive fractionally integrated moving average (ARFIMA), are studied. Through simulation, they found that the AR model over performs the other four not only in its simplicity but also in its prediction precision. In Wolski et al. (1999, 1997) and Wolski (1998), the AR model is used to predict the performances like delay and throughput from end to end. In this paper, we adopt an improved AR model to predict the available number of resources in equipment pool according to its historical information.

6 Conclusions and future work

This paper introduces a PACA for user advance reservation in equipment grid. The grid computing provides the infrastructure needed to access and benefit from distributed resources through the virtual organisations (Albodour et al., 2008). Since the grid is an important part in the execution of applications, vast data processing, and sharing resources which need large computational power. The QoS on grid aims to guarantee and maintain the availability and the access of services, resources, and equipments. Furthermore, widespread grid adoption also increases the need for automated distributed management of grids, as the number of resources offered on these grids rises dramatically (Volckaert et al., 2005). The dynamic configuration and optimisation techniques of grid resource usage can greatly minimise the management cost for a large-scale grid system, and at the same time achieve better resource efficiency and QoS support (Kephart and Chess, 2003; Ganek and Corbi, 2003). We compare the performance of our proposed algorithm with a traditional admission control algorithm via simulation. The unfulfilment request problem in advance reservation can be serious especially when experiments need the participation of multiple resources. The simulation results show a trade-off between the unfulfilment probability, the request rejection probability, and resource utilisation.

Several directions can be identified for future investigation, including:

1 an economic model for the unfulfilment request problem and the method to minimise its impact

2 extension of this predicted algorithm to deal with the case where multiple resources are needed.

All these ideas are listed in our future work in this research project.

Acknowledgements

This work is supported by the National 863 High-Tech R&D Program of the Ministry of Science and Technology of China under grant No. 2006AA10Z216, the National Science Foundation of China under grant No. 60604033 and the National Basic Research Program of China Grant 2007CB807900, 2007CB807901.

References

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) 'Wireless sensor networks: a survey', *Computer Networks*, Vol. 38, pp.393–422.

Albodour, R., James, A., Yaacob, N. and Godwin, A. (2008) 'Grid service QoS in medical environments', *Computer Supported Cooperative Work in Design, the 12th International Conference on CSCWD*, April, pp.16–18, pp.548–552.

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S. and Xu, M. (2007) *Web Services Agreement Specification (WSAgreement)*, March, available at http://www.ogf.org/documents/GFD.107.pdf).

Barz, C., Eickermann, T., Pilz, M., Wäldrich, O., Westphal, L. and Ziegler, W. (2007) 'Co-allocating compute and network resources – bandwidth on demand', *VIOLA Testbed CoreGRID Symposium, CoreGRID Series, Towards Next Generation Grids*, Springer, Rennes, France, August, pp.193–202.

Battestilli, L., Hutanu, A., Karmous-Edwards, G., Katz, D.S., MacLaren, J., Mambretti, J., Moore, J.H., Seung-Jong, P., Perros, H.G., Sundar, S., Tanwir, S., Thorpe, S.R. and Yufeng, X. (2007) 'EnLIGHTened computing: an architecture for co-allocating network, compute, and other grid resources for high-end applications,' *International Symposium on High Capacity Optical Networks and Enabling Technologies*, HONET 2007, November, pp.1–8.

Berson, S., Lindell, R. and Braden, R. (1998) *An Architecture for Advance Reservations in the Internet*, Technical Report, USC Information Science Institute, available at http://citeseer.ist.psu.edu/cache/papers/cs/18982/http:zSzzSz www.isi.eduzSz~bersonzSzadvance.pdf/berson98architecture.pdf.

Burchard, L.O. (2003) 'On the performance of computer networks with advance reservation mechanisms', *Proc. 11th IEEE Int. Conf Networks (ICON '03)*, pp.449–454.

Buyya, R. and Murshed, M. (2002) '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*, Vol. 14, Nos. 13–15, pp.1175–1220.

Cao, J. and Zimmermann, F. (2004) 'Queue scheduling and advance reservations with COSY', *Proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS 2004)*, pp.63–70.

Curti, C., Ferrari, T., Gommans, L., van Oudenaarde, S., Ronchieri, E., Giacomini, F. and Vistoli, C. (2005) 'On advance reservation of heterogeneous network paths', *Future Generation Computer. Syst.*, Vol. 21, No. 4, pp.525–538.

Degermark, M., Kohler, T., Pink, S. and Schelen, O. (1997) 'Advance reservations for predictive service in the internet', *ACM/Springer Journal of Multimedia Systems*, Vol. 5, No. 3, pp.77–186.

Dinda, P.A. and Hallaron, D.R.O. (1999) 'An evaluation of linear models for host load prediction', *Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing*, pp.87–96.

Dinda, P.A. and Hallaron, D.R.O. (2000) 'Host load prediction using linear models', *Cluster Computing*, Vol. 3, No. 4, pp.265–280.

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

Foster, I., Kesselman, C. and Tuecke, S. (2001) 'The anatomy of the grid: enabling scalable virtual organizations', *International Journal of Supercomputer Applications*, Vol. 15, No. 3, pp.200–222.

Foster, I., Roy, A. and Sander, V. (2000) 'A quality of service architecture that combines resource reservation and application adaptation', *Proceedings of the Eighth International Workshop on Quality of Service*, June, Pittsburgh, PA.

Ganek, A.G. and Corbi, T.A. (2003) 'The dawning of the autonomic computing era', *IBM Systems Journal*, Vol. 42, No. 1, pp.5–18.

Hafid, A., Bochmann, G. and Dssouli, R. (1998) 'A quality of service negotiation approach with future reservations (NAFUR): a detailed study', *Computer Networks and ISDN Systems*, Vol. 30, No. 8, pp.779–794.

Juhua, P., Zhang, X. and Zhenxing, W. (2003) 'The research on QoS for grid computing', *Communication Technology Proceedings, the ICCT 2003. International Conference on Communication Technology*, April, Vol. 2, pp.1711–1714.

Kephart, J.O. and Chess, D.M. (2003) 'The vision of autonomic computing', *IEEE Computer*, Vol. 36, No. 1, pp.41–50.

Kravtsov, V., Schuster, A., Carmeli, D., Kurowski, K. and Dubitzky, W. (2008) 'Grid-enabling complex system applications with QosCosGrid: an architectural perspective', *Proceedings of The International Conference on Grid Computing and Applications (GCA'08)*, Las-Vegas, USA, pp.168–174.

Krishnaswamy, S., Zaslavsky, A. and Loke, S.W. (2002) 'Predicting application runtimes using rough sets', *Proc. 9th Int'l Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp.455–462.

Min, R. and Maheswaran, M. (2001) 'Scheduling advance reservation with priorities in grid computing system', *International Conference on Parallel And Distributed Computing Systems(PDCS'01)*, pp.172–176.

NSF Cyberinfrastructure Council (2006) *NSF's Cyberinfrastructure Vision for 21st Century Discovery*, Version 5.0, 20 January.

Plestys, R., Vilutis, G., Sandonavicius, D., Vaskeviciute, R. and Kavaliunas, R. (2007) 'The measurement of grid QoS parameters', *29th International Conference on Information Technology Interfaces, 2007 (ITI 2007)*, June, pp.703–707.

Roy, A. and Sander, V. (2003) 'GARA: a uniform quality of service architecture', in J. Nabryski, J.M. Schopf and J. Weglarz (Eds.): *Grid Resource Management State of the Art and Future Trends*, June, pp.377–394, Kluwer Academic Publishers.

Siddiqui, M., Villazon, A. and Fahringer, T. (2006) 'Grid capacity planning with negotiation-based advance reservation for optimized QoS', *SuperComputing Conference*, Tampa, Florida.

Smith, W., Foster, I. and Taylor, V. (1998) 'Predicting application runtimes using historical information', *LNCS 1459*, pp.122–142.

Smith, W., Foster, I. and Taylor, V. (2000) 'Scheduling with advanced reservations', *Proc. of the Int. Parallel and Distributed Processing Symposium (IPDPS)*, pp.127–132.

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

Vertically Integrated Optical Testbed for Large Application in DFN (VIOLA) (2005) available at http://www.viola-testbed.de/.

Volckaert, B., Thysebaert, P., De Leenheer, M., De Turck, F., Dhoedt, B. and Demeester, P. (2005) 'A distributed resource and network partitioning architecture for service grids', *IEEE International Symposium on Cluster Computing and the Grid*, Vol. 1, pp.426–433.

Waldrich, O., Wieder, P. and Ziegler, W. (2006) 'A meta-scheduling service for co-allocating arbitrary types of resources', *Proc. of the Second Grid Resource Management Workshop (GRMWS'05) in conjunction with the Sixth International Conference on Parallel Processing and Applied Mathematics (PPAM 2005): Lecture Notes in Computer Science*, Poznan, Poland, 11–14 September, Vol. 3911, pp.782–791.

Wang, Y. and Wu, C. (2005a) 'A study on education resource sharing grid', *International Journal of Information Technology, Special Issue on Grid Computing I*, Vol. 11, No. 3, pp.73–80.

Wang, Y. and Wu, C. (2005b) 'Study on equipment interoperation chain model in grid environment', *Lecture Notes in Computer Science*, Vol. 3758, pp.588–595.

Wieczorek, M., Siddiqui, M., Villazon, A. et al. (2006) 'Applying advance reservation to increase predictability of workflow execution on the grid', *Second IEEE International Conference on e-Science and Grid Computing*, Amsterdam.

Wieder, P., Wäldrich, O. and Ziegler, W. (2007) 'Advanced techniques for scheduling, reservation and access management for remote laboratories and instruments', *TR-0078, Institute on Resource Management and Scheduling, CoreGRID – Network of Excellence*, March, available at http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0078.pdf.

Wolf, L.C. and Steinmetz, R. (2004) 'Concepts for resource reservation in advance', *Journal of Multimedia and Applications*, Vol. 4, No. 3, pp.255–278.

Wolf, L.C., Delgrossi, L., Steinmetz, R., Schaller, S. and Wittig, H. (1995) 'Issues of reserving resources in advance', *Proc. 5th Int. Workshop Network and Operating System Support for Digital Audio and Video*, Durham, NH, pp.27–37.

Wolski, R. (1998) 'Dynamically forecasting network performance using the network weather service', *Cluster Computing*, Vol. 1, No. 1, pp.119–132.

Wolski, R., Spring, N.T. and Hayes, J. (1999) 'The network weather service: a distributed resource performance forecasting service for metacomputing', *Journal of Future Generation Computing Systems*, Vol. 15, Nos. 5–6, pp.757–768.

Wolski, R., Spring, N.T. and Peterson, C. (1997) 'Implementing a performance forecasting system for metacomputing: the network weather service', *Proceedings of Supercomputing 1997*, pp.7–25.