Inf Syst Front (2013) 15:589–598
DOI 10.1007/s10796-013-9437-6

A reliable scheduling method in equipment grid using provenance information

Yuexuan Wang · Jie Yin

Published online: 21 June 2013
© Springer Science+Business Media New York 2013

Abstract Scheduling resources in grid is an open difficult problem due to resource fluctuations. A fuzzy scheduling method using provenance information is proposed. In this method, resource dispatch probability is dynamically adjusted according to user feedback information, which is user appreciation information represented by fuzzy variables. To minimize the influence of cheating, collusive and decrying of user appreciations, provenance information is used to estimate trust factor of each user appreciation during resource dispatch probability adjustment process. Simulation results confirm capability of the proposed method to effectively reduce impacts of malicious user appreciations and increase user satisfactions.

Keywords Equipment grid · Provenance · Scheduling

1 Introduction

Equipment grid is the grid that connects geographical distributed instruments, like microscopes, telescopes, mass spectrometers etc, together to provide resource coordination and cooperation for users. It consists of the following three components: equipment pool alliance, equipment pool, and geographically distributed physical instruments. Scheduling remote access of scientific instruments in equipment grid with consideration of QoS issues was studied in (Yin et al. 2007), in which a QoS feedback mechanism to show whether users are satisfied with their experiment results from equipment grid is provided. Feedback information which reflects QoS of instruments is fuzzy variable. The dispatch probability, defined as the probability that each online instrument in its equipment pool has to get a newly dispatched job, of instrument is dynamically adjusted according to user feedback information. The larger an instrument dispatch probability is, the more chance it has to run a new job. Thus utilization of instruments with higher QoS is increased. As a result, QoS of equipment grid as a whole is dramatically improved.

There are some problems unsolved. For example, simulation results are based on the assumption that all users are honest and provide authentic appreciations, but in reality some malicious behaviors like cheating, collusive and decrying are inevitable. To deal with such problem, a more reliable scheduling method using provenance information which can detect user malicious appreciations and try to reduce their impacts on instruments is proposed.

The rest of this paper is organized as follows. In Section 2, relevant background is examined. In Section 3, we introduce the proposed scheduling method. The results of simulation studies are discussed in Section 4. Section 5 presents some related work and Section 6 concludes this paper.

2 Background

In equipment grid, similar instruments are organized into equipment pool and different equipment pools constitute equipment pool alliance (Wang and Wu 2005). Instruments

Y. Wang (✉)
Institute for Theoretical Computer Science, Tsinghua University, Beijing 100084, China
e-mail: wangyuexuan@tsinghua.edu.cn

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

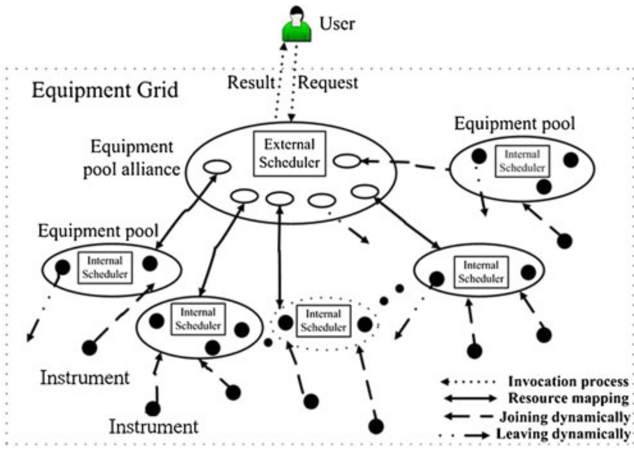


Fig. 1 Hierarchy model of equipment grid

are subordinated to certain equipment pools and can join and leave the pools dynamically. All equipment pools have their images in equipment pool alliance and can join and leave pool alliance dynamically, as is shown in Fig. 1.

When a request to conduct experiment arrives, it is first submitted to equipment pool alliance. Equipment pool alliance analyses this request and verifies whether it can be accepted with currently online instruments. If this job request can be accepted, equipment pool alliance submits it to related equipment pool or pools (in case the job needs the coordination of multiple instruments) by external scheduler. Suitable instruments will be selected by internal scheduler(s) in related equipment pool(s) to run the experiment according to their dispatch probabilities.

In the equipment grid, the target to be optimized is not a single index like makespan or cost but a comprehensive parameter, also known as QoS of instruments (Yin et al. 2007). This parameter includes all the following factors like, execution time of all sub-jobs of this experiment, cost for the whole experiments, system error and precision of experiment results etc. Our former work expressed this parameter as a fuzzy parameter, QoS, which can be reflected by user feedback information. To solve malicious user appreciation problem mentioned in Section 1, we extend our former work by using provenance information.

The definition of provenance in Oxford English Dictionary is that, *the fact of coming from some particular source or quarter; origin; derivation.* Currently provenance research in computer science focus on data provenance, like Lineage Information Program (Miles et al. 2005), Chimera (Foster et al. 2002), myGrid (Stevens et al. 2003) etc. The survey of data provenance is introduced in (Buneman et al. 2001).

3 Fuzzy scheduling algorithm using provenance information

When a user submits an experiment request in equipment grid, these steps will be followed.

Step 1. The user logs into the equipment grid system with his account and password. After successful login, system maps his account into his VO account.

Step 2. The user submits the experiment request with his JDL (Job Description Language) file. In a JDL file, the deadline for this experiment, the maximum cost he can offer, the instruments needed to fulfill this experiment etc, are specified.

Step 3. The equipment pool alliance will decide whether to accept this request or not according to the JDL file. If a request is accepted, equipment pool alliance will submit it to related equipment pool or pools by external scheduler.

Step 4. When an equipment pool receives a task dispatched by equipment pool alliance, the internal scheduler will dispatch this task to a suitable instrument according to JDL constrains and estimated instrument QoS. The QoS of instrument is a multi-factor influenced parameter that can be better expressed as fuzzy variable and reflected by user appreciation.

Step 5. The user gives an appreciation towards the experiment based on the process and result after experiment finished and result received.

Step 6. The internal scheduler adjusts dispatch probability according to user appreciations.

The detailed scheduling model is demonstrated in Fig. 2.

In Fig. 2, VO, external scheduler and internal scheduler store important information such as who enter the system, the affiliated VO the user belongs to, what kind of equipment pool(s) required, which instrument in the selected pool be scheduled to run the job, what the result and user appreciation is etc, into the provenance database.

According to resource dispatch probability, internal scheduler determines which resource to be selected when new request comes. Dispatch probability in equipment grid is modified by user appreciations and provenance information.

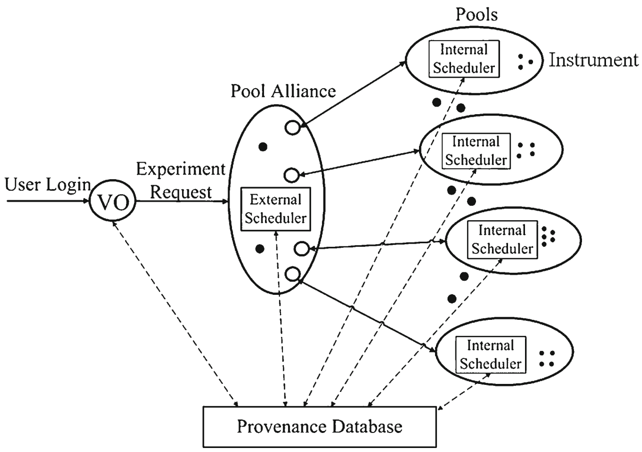


Fig. 2 Scheduling model of equipment grid

3.1 Mechanism of external scheduler and internal scheduler

When user submits a job request with its JDL file, the external scheduler will examine the job JDL and distribute the job to related equipment pool(s). Job request can be presented as *Request* (E, C, T), in which E, C and T means the kinds of instruments needed, total cost and total time, respectively. If following constrains can be satisfied, this request can be accepted.

$$E \in \{EP_1, EP_2, \ldots, EP_k\} \tag{1}$$

$$\sum_{i=1}^{m} Te_i + \sum_{i=1}^{m} Tw_i \le T \tag{2}$$

$$\sum_{i=1}^{m} C_i \le C \tag{3}$$

In (1), it means current online equipment pools contains the instrument(s) needed. EP_i ($i \in [1, k]$) is equipment pool i, and k is the total number of online equipment pools that are available. In (2), total execution time and waiting time is within a time limit T when there are m instruments working coordinately to finish the experiment. Te_i and Tw_i means execution time and queue waiting time of the ith step in the service chain respectively. The estimated Te_i and Tw_i can be calculated from provenance information. Equation (3) specifies that total cost is within the upper bound, C.

When an internal scheduler receives a job distributed by pool alliance, it will find a suitable resource and distribute this job to it. In FCFS algorithm when a new job comes, it will enter into a queue waiting to be served. When there are idle resources available, the queue head will be submitted to the instrument. In Min-min algorithm, the job in the queue with minimum T is scheduled to instrument with minimum Tw. Tw of each instrument is recorded in its pool. Different from FCFS and Min-min, our scheme dispatches jobs to instruments according to resource dispatch probability. There are many factors that may influence user appreciations, so a comprehensive fuzzy variable is appropriate to represent it. The linguistic values like, *very bad*, *bad*, *normal*, *good* and *very good* are used to reflect user appreciations towards their experiments. Normally, a *very good* appreciation means the instrument used is of *very good* quality and the user is satisfied with the experiment submitted. A *very bad* appreciations is usually associated with *very bad* of instrument QoS and reflects user dissatisfaction.

3.2 Scheduling algorithm

The following algorithm adjusts dispatch probability in equipment pool according to user appreciations. When several jobs come simultaneously, the job with minimum execution time is scheduled first. The instrument with more user satisfactory appreciations like *very good* or *good* has a higher dispatch probability while those with poor appreciations much lower. Taking user malicious behaviors like cheating, collusive and decrying into account, the following two measures are used to reduce the influence of these bad deeds.

1) Provenance information like, all the user appreciations for a specific instrument, how the current dispatch probability came etc, will be recorded. If a new appreciation is too different from an estimated appreciation $appraisal_{est}$ (estimated from past appreciations the considered instrument had received) the trust factor of this appreciation will decrease. Different here means that there are at least two rank difference between current user appreciation and estimated appreciation. For example, when user appreciation is *very good*, while the $appraisal_{est}$ is *normal* according to provenance database, because there are two rank differences between user appreciation and estimated appreciation, then we can say that they are different.
2) Compared with our form method proposed in (Yin et al. 2007), in which equipment pool adjusts dispatch probability at every user appreciation arrival. The scheduling algorithm in this proposed method will be triggered to adjust dispatch probability when instruments in a pool have received several appreciations, for example N_0.

The algorithm can be expressed as follows.

Algorithm
Constant number: c_1, c_2, $deltap_1$, $deltap_2$, $deltap_3$, $deltap_4$, $deltap_5$, N_0, N_1, N_2
Variables: *rank_vg*, *rank_g*, *rank_n*, *rank_b*, *rank_vb*, *appraisal_est*, *currentAppraisal*, $prep \in R^{i}$, $N \in R^{i \times j}$, *Trust*, $A \in R^{i \times j \times k}$
Input: *currentAppraisal*

1. **while** the number of user appreciations from VO_i to equipment j, $N(i,j) < N_0$
 1.1 $A(i,j, N(i,j)) \leftarrow$ *currentAppraisal*
 1.2 **if*difference***(*appraisal_est*,*currentAppraisal*)>2
  1.2.1 *trustfactor* $\leftarrow c_2$
 1.3 **else** trustfactor $\leftarrow c_1$
  endif
 1.4 $Trust(i,j, N(i,j)) \leftarrow$ *trustfactor*
 1.5 $N(i,j)$++
 endwhile
2 **for** every i, *// Select a user appreciation rank*
 2.1 **for** k= 1: N_1
 2.1.1 **switch**$(A(i,j,k))$:
  2.1.1.1 **case**: "*very good*": *rank_vg+ trustfactor*
  2.1.1.2 **case**: "*good*": *rank_g+ trustfactor*
  2.1.1.3 **case**: "*normal*": *rank_n+ trustfactor*
  2.1.1.4 **case**: "*bad*": *rank_b+ trustfactor*
  2.1.1.5 **case**: "*very bad*": *rank_vb+ trustfactor*
  endswitch
3 select the largest among *rank_vg*, *rank_g*, *rank_n*, *rank_b* and *rank_vb* and assign the related appreciation rank to *appraisal_est*

4 **switch**(*appraisal_est*)
 //Use *appraisal_est* to adjust dispatch probability
 4.1 **case**: "*very good*":
  4.1.1 $deltap = deltap_1$
 4.2 **case**: "*good*":
  4.2.1 $deltap = deltap_2$
 4.3 **case**: "*normal*":
  4.3.1 $deltap = deltap_3$
 4.4 **case**: "*bad*":
  4.4.1 $deltap = deltap_4$
 4.5 **case**: "*very bad*":
  4.5.1 $deltap = deltap_5$
 endswitch
 4.6 **for** k= 1: N_2
  4.6.1 $prep_k = prep_k * (1- deltap)$;
 4.7 $prep_j = prep_j + deltap$;
5 $N(i,j) \leftarrow 0$
6 $A(i,j,N(i,j)) \leftarrow$ null
End

In this algorithm, c_1, c_2 are trust factors assigned by program and $c_1 > c_2 > 0$. N_1 is the number of VO and N_2 is the current number of instruments in their equipment pool. *currentAppraisal* is the appreciation that user provided to his recent job. *appraisal_est* is a variable that calculated from historical appreciations that this resource had gain. $prep^{[i]}$ is the dispatch probability in the equipment pool. N is a two dimensional matrix and $N(i,j)$ represents the number of appreciations to resource j from users in VO_i. *Trust* and A are three dimensional matrix that record all trust factors and user appreciations. Function ***difference***(p_1, p_2) is to get difference of two user appreciations p_1 and p_2. For example, If p_1 is *very good* and p_2 is *very bad*, then ***difference***(p_1, p_2) is 4. Normally $deltap_1 > deltap_2 > deltap_3 > 0 > deltap_4 > deltap_5$

3.3 Malicious behavior categorization

User malicious appreciations can be classified into three types according to their behaviors. The first type (*T1*) refers to the situation that users always give high rank appreciations no matter what they actually think about their experiments. The second type (*T2*) is that users always send low rank appreciations. In the third type (*T3*), users give random appreciations regardless of their own true feelings. Some more complex user behaviors will not be discussed here.

We define $UAP_1 \sim UAP_5$ is the probability that user appreciation ranging from *very good* to *very bad* for a given experiment. Table 1 is the distribution of user appreciation probability for the three cases.

In the rest of this subsection, we will analyze the influence of user malicious appreciations to dispatch probability of equipment pool under the three circumstances.

We assume that when an instrument R received N_0 appreciations, in which r appreciations are malicious, from a same VO. $\Delta p_1 \sim \Delta p_5$ means $deltap_1 \sim deltap_5$ in the algorithm proposed in Section 3.2 and the value of Δp is chosen from Δp_1 to Δp_5 decided by QoS of R. p is the initial dispatch probability of R when there are no malicious behaviors.

Table 1 Distribution of malicious user appreciation

Type	Probability distribution
T1	$UAP_1=1,\ UAP_2=UAP_3=UAP_4=UAP_5=0$
T2	$UAP_1=UAP_2=UAP_3=UAP_4=0,\ UAP_5=1$
T3	$UAP_1=UAP_2=UAP_3=UAP_4=UAP_5=0.2$

Then according to the algorithm in (Yin et al. 2007), the dispatch probability of R becomes (4).

$$\begin{cases} p\ (1+\Delta p_1)^r\ (1+\Delta p)^{N_0-r} & (T1) \\ p\ (1+\Delta p_5)^r\ (1+\Delta p)^{N_0-r} & (T2) \\ \dfrac{p(1+\Delta p)^{N_0-3r/5}}{[(1+\Delta p_1)(1+\Delta p_2)(1+\Delta p_3)(1+\Delta p_4)(1+\Delta p_5)]^{r/5}} & (T3) \end{cases} \tag{4}$$

While for our algorithm, the dispatch probability of R becomes (5).

$$\begin{cases} \dfrac{p}{2}\left\{[1+\mathrm{sgn}(N_0-(1+c_2)r)](1+\Delta p)^{N_0-(1+c_2)r}+[1+\mathrm{sgn}((1+c_2)r-N_0)](1+\Delta p_1)^{(1+c_1)r-N_0}\right\} & (T1) \\ \dfrac{p}{2}\left\{[1+\mathrm{sgn}(N_0-(1+c_2)r)](1+\Delta p)^{N_0-(1+c_2)r}+[1+\mathrm{sgn}((1+c_2)r-N_0)](1+\Delta p_5)^{(1+c_1)r-N_0}\right\} & (T2) \\ \dfrac{p(1+\Delta p)^{N_0-3r/5}}{[(1+\Delta p_1)(1+\Delta p_2)(1+\Delta p_3)(1+\Delta p_4)(1+\Delta p_5)]^{r/5}} & (T3) \end{cases} \tag{5}$$

Divided (5) by (4), we have (6).

$$\begin{cases} 2(1+\Delta p_1)^r(1+\Delta p)^{N_0-r} \Big/ \left\{[1+\mathrm{sgn}(N_0-(1+c_2)r)](1+\Delta p)^{N_0-(1+c_2)r}+[1+\mathrm{sgn}((1+c_2)r-N_0)](1+\Delta p_1)^{(1+c_1)r-N_0}\right\} & (T1) \\ 2(1+\Delta p_5)^r(1+\Delta p)^{N_0-r} \Big/ \left\{[1+\mathrm{sgn}(N_0-(1+c_2)r)](1+\Delta p)^{N_0-(1+c_2)r}+[1+\mathrm{sgn}((1+c_2)r-N_0)](1+\Delta p_5)^{(1+c_1)r-N_0}\right\} & (T2) \\ 1 & (T3) \end{cases} \tag{6}$$

From (6), we can find that in $T3$, the newly proposed method have the same effect with former method. While in the other two situations, the new method has better performance which can be demonstrated by the following simulation results.

4 Simulations

In this Section, simulations are given to illustrate the algorithm introduced in Section 3.

In this simulation, an equipment pool that containing 500 instruments of different QoS, with 100 *very good*, 100 *good*, 100 *normal*, 100 *bad* and 100 *very bad*, are providing services in equipment grid. When user submits an experiment request in equipment grid, one of these instruments is scheduled for the job.

Figures 3, 4, 5, 6, and 7 are the simulation results of 100,000 user appreciations.

From Figs. 3, 4, 5, 6, and 7, the vertical axis is the number of user appreciations in the form of logarithmic coordinates.

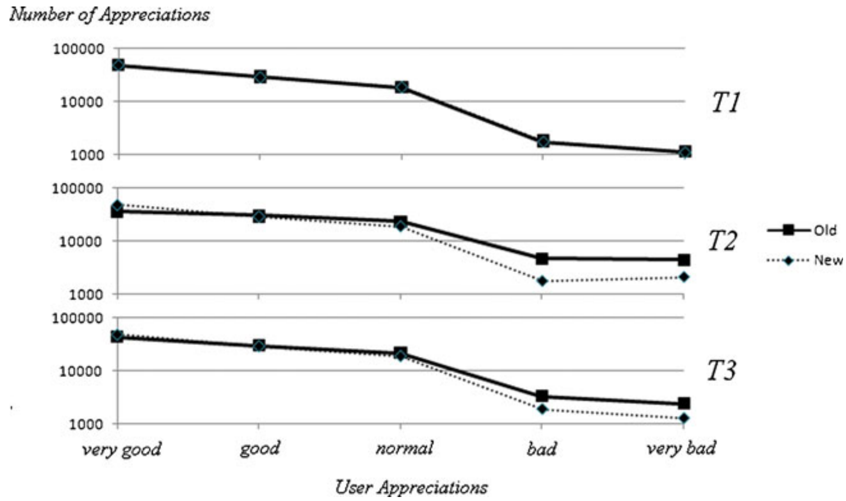


Fig. 3 One percent of malicious user appreciations

The horizontal axis is the user appreciations like *very good*, *good*, *normal*, *bad* and *very bad*. *T1*, *T2* and *T3* are the three user malicious appreciation types in Section3. The curve marked old is the simulation result of our former algorithm and the curve marked new means simulation result of this new algorithm.

In Fig. 3, when 1 % of user appreciations are malicious appreciations, the simulation results of our former algorithm and this newly presented method are illuminated. We can find in *T1*, there is little difference between these two approaches. In *T2* and *T3*, this new algorithm over performs the former in more appreciations of *very good* and less number in *very bad*.

In Figs. 4 and 5, when 10 % and 50 % of user malicious appreciations are proposed, the trend is similar to Fig. 3 except in *T*1. There are a bit more low rank appreciations like *bad* and *very bad* in the new algorithm than the old one. This is because in *T*1, the malicious appreciations are *very good*, which increased the number of high rank appreciations and decrease the number of low rank appreciations

In Fig. 6, when all user appreciations are malicious, there is no difference between these two approaches because all appreciations have no relationship with instrument selection.

We can find that in *T*1, there are much more *very good* user appreciations than *T*2 and *T*3 under the same conditions. This is because all malicious appreciations in *T*1 are *very good*, which increased the number of *very good* appreciations. It is the same reason why there are much more *very bad* appreciations in *T*2. In *T*3 (because users give their appreciations randomly) the percentage of malicious appreciations increased, the user appreciations distributed more evenly.

Figure 7 is the situation when 1/3 of the user appreciations belong to *T*1, 1/3 belongs to *T*2 and 1/3 belongs to *T*3. It is clear that in case of *T*1 and *T*2, the number of *bad* and *very bad* appreciations in the new method is a bit less than the old

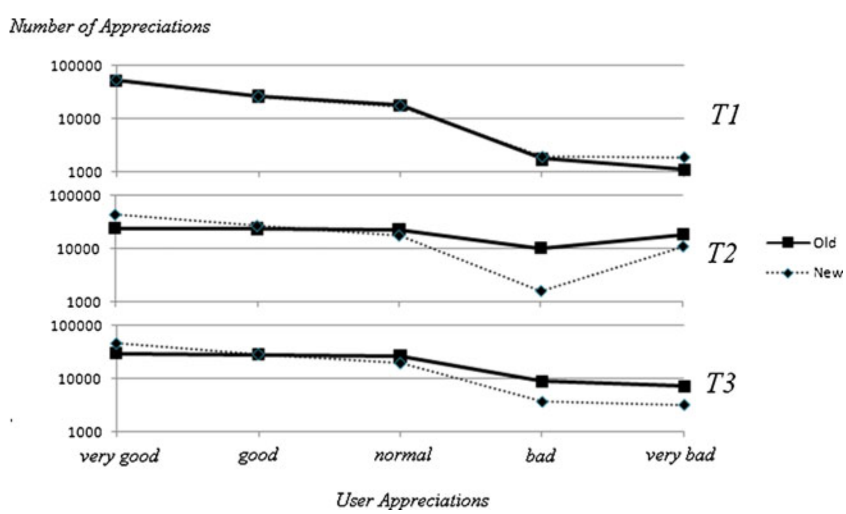
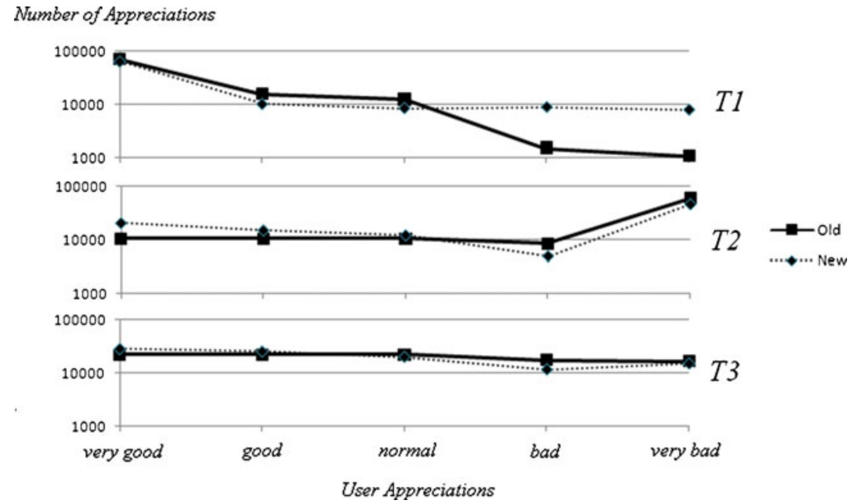


Fig. 4 Ten percent of malicious user appreciations

Fig. 5 Fifty percent of malicious user appreciations



method. In $T3$, there is little difference in the two algorithms because the users give their appreciations randomly.

More simulation results (like case 1, 50 %$T1$+50 %$T2$; case 2, 50 %$T2$+50 %$T3$ and case 3, 50 %$T1$ +50 %$T3$) support the conclusion draw from Fig. 7 and are not demonstrated here.

In the following, we will study the relationship of N_0, the number of user appreciations that a specific instrument received, and user appreciation distribution in such situation that 10 % of user appreciation belong to $T1$, 10 %$T2$, 10 %$T3$ and 70 % authentic appreciation.

In Fig. 8, the situations when N_0 equals to 1, 2, 5, 10, 100 and 1,000 is simulated. The upper part in Fig. 8 is the distribution of *very good* appreciations. The lower part is the distribution of *very bad* appreciations.

In Fig. 8, we can find that when N_0 is 10, the distribution of user appreciations is more acceptable. It has more *very good* user appreciations and less *very bad* user appreciations towards all the three cases of user malicious appreciations.

The reason lies in that when N_0 is small, the algorithm proposed in 3.2 is invoked frequently and some malicious appreciations will play important role in instrument dispatch probability adjustment process. While N_0 is too big, the proposed algorithm will be used less, and the dispatch probability is adjusted in a limited region with limited function.

Similar conclusions can be drawn from more simulation situations, like: (1 %$T1$, 1 %$T2$, 1 %$T3$ and 97 % authentic appreciations) and (25 %$T1$, 25 %$T2$, 25 %$T3$ and 25 % authentic appreciations), and detailed results will not be demonstrated.

In the following we will study the relationship between c_2 and user appreciation distribution. Figures 9 and 10 is the *very good* and *very bad* appreciation distribution when N_0 equals to 10 and c_2 varied from 0 to 1 with increment of 0.1. The horizontal axis is c_2 and the vertical axis is the ratio of the number of user appreciations to the number of user appreciations when c_2 equals to 1.

Fig. 6 One hundred percent of malicious user appreciations

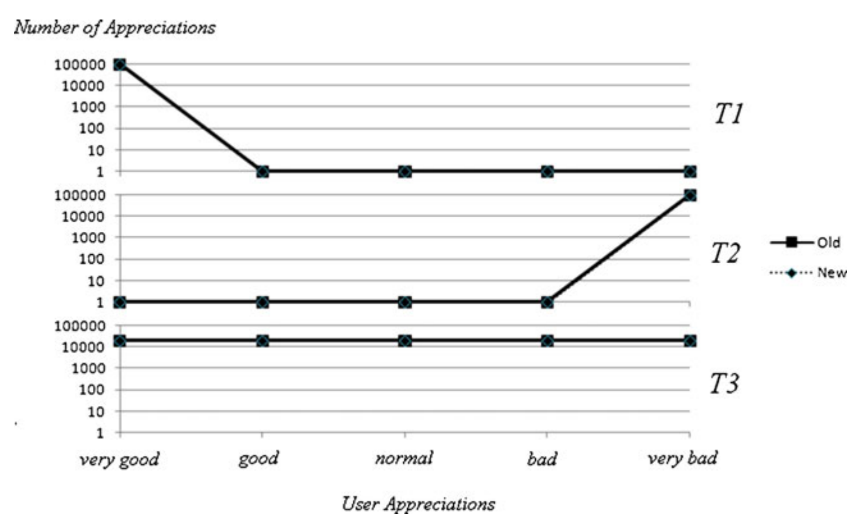
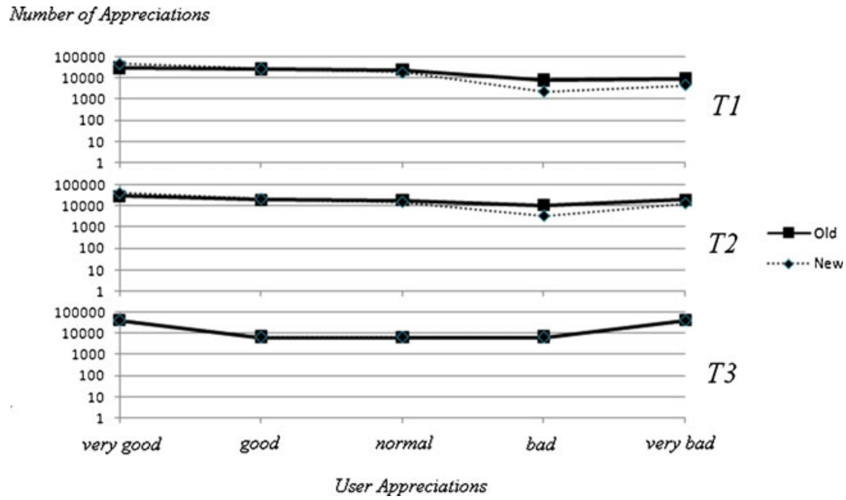


Fig. 7 Hybrid malicious user appreciations



The upper part of Fig. 9 is the user appreciation distribution of *very good*, and the lower part is *very bad*. The four curves in each part are the situations when hybrid situation (1/3 of $T1$, 1/3 of $T2$ and 1/3 of $T3$), 10 % of $T1$ malicious user appreciations, 10 % of $T2$ and 10 % of $T3$ occur.

Figure 10 is similar to Fig. 9, but the three curves are 50 % malicious appreciations exist which are $T1$, 50 %$T2$ and 50 %$T3$.

In Fig. 9 when the malicious user appreciation ratio is low, the value of c_2 has little effect on the user appreciation distribution. As we can see from Fig. 10 as the percentage of malicious users reaches 50 %. We see very little influence from c_2. In the lower part, it is clear that in case of $T2$ and $T3$, the user appreciation distribution is incentive to c_2. In case of $T1$, with the increment of c_2, the number of *very bad* appreciations decreased. The reason is that in $T1$, with the trust factor increase, more and more instruments especially those with low QoS get high appreciations and thus result in the fewer *very bad* appreciations.

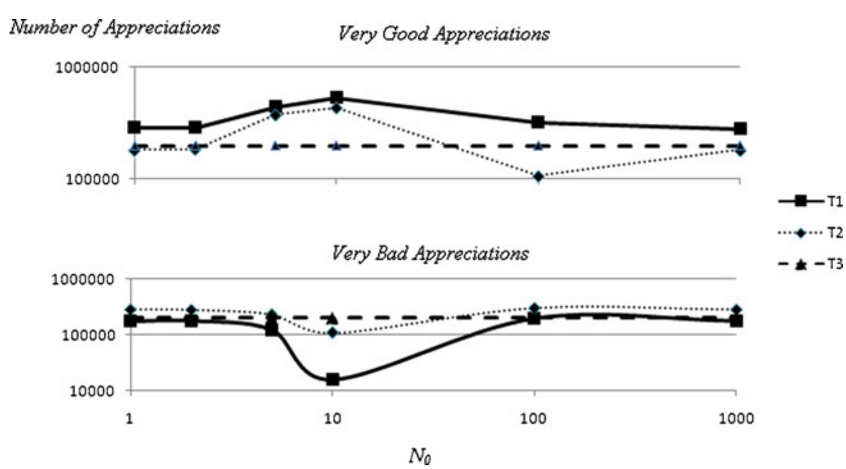
From all the simulation results, we can find that,

1) In case of $T1$, when the malicious appreciations ratio is low or medium, the new algorithm over performs the old one in fewer *very bad* user appreciations.
2) In case of $T2$ and $T3$, when the malicious appreciation ratio is low or medium, the new algorithm has more *very good* appreciations and less *very bad* appreciations.
3) When the malicious appreciation ratio is high or very high, both algorithms have similar performance.
4) The parameters like N_0, c_2 and so on can be better adjusted using some optimization method like GA, NN etc.

5 Related work

Traditional scheduling problem is to dispatch *m* workloads to *n* resources and optimize a certain criteria such as makespan, cost, utilization of resources etc. The problem of optimally mapping these tasks onto resources in a distributed heterogeneous environment had been shown, in general, to be NP-

Fig. 8 Relationship of N_0 and user appreciation distribution



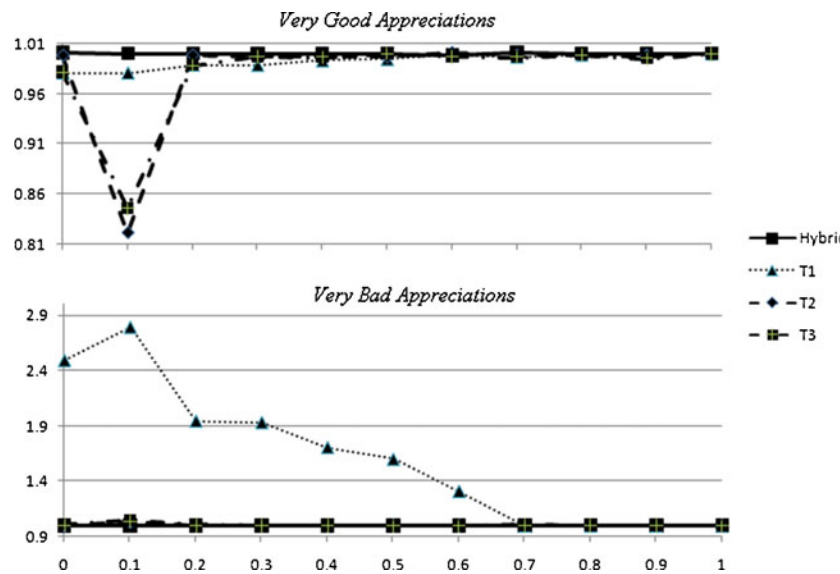


Fig. 9 Relationship of c_2 and user appreciation distribution when malicious rate is low

complete (Fernandez-Baca 1989). Some heuristic algorithms were developed to solve such problems. Eleven static heuristics, which are OLB, MCT, MET, Min-min, Max-min, Duplex, GA, SA, GSA, Tabu, A* respectively, had been studied in (Braun et al. 2001). Simulation results showed that GA has the best result but with too much calculation time while Min-min heuristic has best overall performance. While in grid environment, resources fluctuate dynamically. All the static heuristics need slightly adjusted. Eight dynamic scheduling methods which include five on-line mode heuristics and three batch model heuristics were compared in (Maheswaran et al. 1999) and KPB heuristic over performed than the other four on-line dynamic heuristics and Min-min heuristic had the best performance than the other two batch model heuristics.

Some prediction techniques (Spooner et al. 2003; Yang et al. 2003) for resource availability and resource advanced reservation techniques (Cao and Zimmermann 2004; Smith et al. 2000) were used to improve scheduling performance. While all these methods tried to optimized a certain parameter like make span, cost and resource utilization etc under such assumption that all jobs are irrelevant and there is no data dependence among them.

In (Mekouar et al. 2006), the authors proposed a reputation management scheme for P2P networks and analyzed some malicious behaviors that ruin the reputation in the system.

Our former work (Yin et al. 2007) presented a fuzzy parameter QoS to represent the multi objects to be optimized and this work provides an improved algorithm in case of user malicious appreciations.

6 Conclusions

We had proposed a fuzzy scheduling algorithm using provenance information to schedule instruments in equipment grid. In this algorithm, instruments with satisfactory appreciations like *very good* or *good* have a large dispatch probability. Using provenance information, this algorithm will detect some

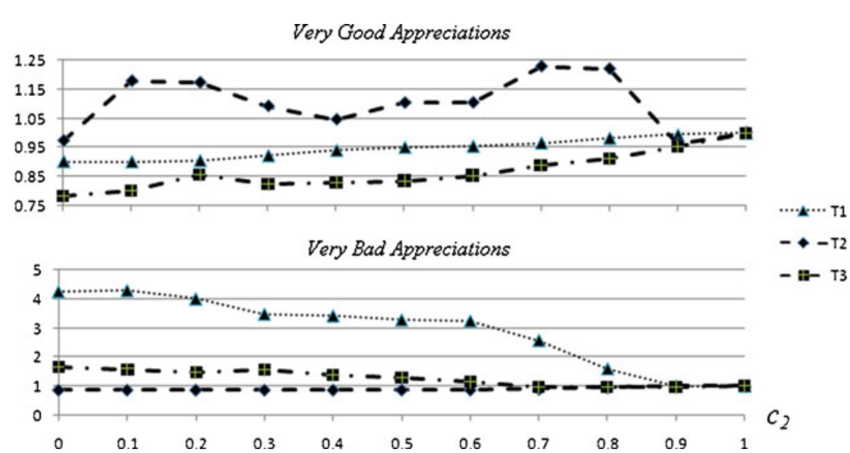


Fig. 10 Relationship of c_2 and user appreciation distribution when malicious rate is high

malicious appreciations and try to decrease their trust factors in dispatch probability adjustment process. Simulation results show that in comparison with our former work, this new algorithm is more robust when there are some malicious behaviors.

In future work, we hope to extend the scheduling algorithm into multiple resources scheduling scenario, in which an experiment may need the coordination of a service chain (Wang and Wu 2005) consisting of several instruments. In such scenario, factors like service combination should be taken into consideration. A second direction is to apply our algorithm to the test bed currently being developed.

Acknowledgments 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, 2007BC807901 and the Hi-Tech research & Development Program of China Grant 2006AA10Z216.

References

Braun, T. D., Siegel, H. J., Beck, N., et al. (2001). A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed Computing, 61*(6), 810–837.

Buneman, P., Khanna, S., & Tan, W. C. (2001). Why and where: a characterization of data provenance. *Lecture Notes in Computer Science, 1973*, 316–330.

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

Fernandez-Baca, D. (1989). Allocating modules to processors in a distributed system. *IEEE Transactions on Software Engineering, 15*(11), 1427–1436.

Foster, I. T., Vöckler, J. S., Wilde, M., Zhao, Y. (2002). Chimera: A virtual data system for representing, querying, and automating data derivation, SSDBM, 37–46.

Maheswaran, M., Ali, S., Siegel, H. J., et al. (1999). Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. Proceedings of the 8th IEEE Heterogeneous Computing Workshop (HCW99). IEEE Computer Society Press, 30–44.

Mekouar, L., Iraqi, Y., & Boutaba, R. (2006). Peer-to-peer's most wanted: malicious peers. *Computer Networks, 50*(4), 545–562.

Miles, S., Groth, P., Branco, M., Moreau, L. (2005). The requirements of recording and using provenance in e-Science experiments, Technical Report. Electronics and Computer Science, University of Southampton.

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

Spooner, D. P., Jarvis, S. A., Cao, J., et al. (2003). Local grid scheduling techniques using performance prediction. *IEE Proceedings of Computers and Digital Techniques, 150*(2), 87–96.

Stevens, R. D., Robinson, A. J., & Goble, C. A. (2003). myGrid: personalised bioinformatics on the information grid. *Bioinformatics, 19*, 302–304.

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

Yang, L., Schopf, J. M., Foster, I. (2003). Conservative scheduling: Using predicted variance to improve scheduling decisions in dynamic environments. Supercomputing 2003, Phoenix, November 2003.

Yin, J., Cao, J., Wang, Y. X., et al. (2007). Scheduling remote access to scientific equipments in cyberinfrastructure for education and research. *International Symposium on Cluster Computing and the Grid*.

Yuexuan Wang is a Full Professor of Tsinghua University. Lies in the areas of networking and algorithms, distributed computing, and quantum networks. She has a expertise in the field of networks, where she developed new distributed algorithms for several problems such as message broadcasting and energy saving, with applications to wireless networks and mobile aerial sensor networks. In particular, she used compressed sensing techniques to devise efficient data gathering algorithms, and she proposed two different methods called Major Coefficients Recovery and Compressed Sparse Functions. She has already published over 40 papers on computer science in high-ranking journals and international conferences, and has been member of program committee for MobiHoc and several other international conferences.

Jie Yin Ph.D. student of the Department of Automation, Tsinghua University 2005-2008. His research interests lie in the area of grid computing, intelligent control and adaptive control.