A Scheduling Method for Service Chain in Equipment Grid

Yuexuan Wang[1], Jie Yin[2], Meizhi Hu[1]
[1]*Institute for Theoretical Computer Science, Tsinghua University, Beijing, 100084, China*
[2]*National CIMS Engineering and Research Center, Tsinghua University, Beijing, 100084, China*
E-mail:wangyuexuan(yinjie05, humeizhi)@tsinghua.edu.cn

Abstract

Equipment grid aims to facilitate easy access to expensive scientific instruments by grid services and consists of the following three components: service pool alliance, service pool, and geographically distributed physical instruments. When users submit experiments to equipment grid, service pool alliance will allocate instruments in related service pools to conduct the experiments, which may need coordination and cooperation of several physical instruments that constitute service chains. After experiments have finished and results have returned, users evaluate the performance of related service chains. In this work, a scheduling algorithm using provenance information is proposed to enhance performance of equipment grid by increasing dispatch probability of instruments with high QoS. In this algorithm, we express QoS of instruments and user appraisals in fuzzy linguistic values, taking the vagueness of user opinions on experiment results and various criteria to evaluate instrument QoS into account. Simulation results show that with this algorithm, equipment grid can better satisfy the users.

1. Introduction

Grid computing, motivated by wide-area sharing of computational resources [1], has evolved to be a mainstream technology for enabling large-scale virtual organizations [2-4]. Equipment grid that supports remote access to scientific instruments, like microscopes, telescopes, mass spectrometers and so on, for education and research has attracted attention in China. To make expensive scientific instruments accessible for more applications by more people, grid technologies are used to connect instruments for resource sharing and coordination. All functions of instruments in equipment grid are published as services and categorized into service pools. As a part of the China national grid for education and research, remote manipulation of geographically distributed scientific instruments and cross-organization sharing of high quality education resources using grid technologies were discussed [5, 6].

Resource scheduling issues for clusters and grids have been discussed for many years. Especially, using historical QoS data to improve scheduling performance has been proven to be effective. In a parallel and distributed computing environment, QoS data can be defined easily using quantitative values, e.g. job execution time [7], queue waiting time [8], data transfer time [9], CPU workloads [10], which can be modeled and analyzed using performance prediction technologies [11] and be utilized to improve resource scheduling performance. However, it is difficult for users to characterize performances quantitatively since various criteria (e.g. time, cost, performance and location of instruments) may play different roles in different experiments and users can only provide overall appraisals for the experiments they submitted.

Scheduling remote access of scientific instruments with consideration of QoS issues was studied in [12]. A QoS feedback mechanism to reflect whether users are satisfied with their experiment results was provided. The dispatch probability that which instrument will be chosen for a given job is dynamically adjusted according to user feedback information. As a result, utilization of instruments providing high QoS is increased and QoS of equipment grid as a whole is dramatically improved.

There are still unsolved problems in [12]. For example, user feedback information is a comprehensive evaluation towards his experiment, which may involve participation of several instruments belonging to different service pools. QoS improvement in [12] is not obvious in situations when instruments of high quality work coordinate with those of poor quality, thus making QoS of the whole experiment

978-0-7695-3708-5/09 $25.00 © 2009 IEEE
DOI 10.1109/SERVICES-I.2009.74

unsatisfactory. As a result, all instruments that are involved in the service chain will decrease their dispatch probabilities. A new algorithm using provenance information is proposed in this paper. Through simulation, it shows significant increase of user satisfactory appraisals.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the background of scheduling problems in equipment grid for education and research. In Section 3, an algorithm to improve QoS of experiments which may involve participation of several instruments constituting a service chain is proposed. Simulation results are given in Section 4. In Section 5, related work in provenance is introduced and the paper concludes in Section 6.

2. Backgrounds

In equipment grid, instruments with similar functionalities are organized into service pools and different service pools constitute service pool alliance, which works as a UDDI server [21]. When an experiment is submitted to equipment grid, it is first submitted to pool alliance. Pool alliance will analyze the job and verifies whether it can be done with current available instruments. If the job can be fulfilled, service pool alliance will dispatch it to related service pool(s). Suitable instruments will be selected and assigned to the job by the pool(s).

The following 8 steps constitute a full service invoking process in equipment grid. When a user wants to conduct an experiment, he submits an experiment request with its JDF (Job Description File) to service pool alliance in Step 1. In JDF, parameters like the kinds of instruments needed, time and cost limitations, resolution requirements, etc are specified. Pool alliance will check whether it can satisfy the JDF of this experiment with current instruments available. If all requirements in JDF are not satisfied, pool alliance will reply the user with refusal information in Step 2. Otherwise it accepts this user request in Step 2 and analyzes the experiment, then submits related parts to corresponding pools in Step 3. All related pools will find suitable resources and submit part of this experiment to selected instruments in Step 4. In Step 5, selected instruments return results to their pools after they have finished the jobs assigned. Service pools send results to pool alliance in Step 6. Pool alliance composes all results received and gives a final result to the user in Step 7. In Step 8, user feedback is collected about the experiment.

The rest of this paper focuses on Step 4, which is a vital step for resource sharing and scheduling. The scheduling model in [12] takes user feedback information into account and tries to better satisfy the users.

Consider a service pool with N instruments, when a new job is submitted to this pool, the probability that this new job runs on instrument i is p_i ($i \in [1, N]$). It is obvious that the following Equation (1) holds:

$$\sum_{i=1}^{N} p_i = 1 \qquad (1)$$

There are many factors that have influence on user appraisals, for example, cost of experiment that an instrument charges for, execution time and waiting time, location of the instrument, resolution and reliability of the result. All these factors can be regarded as a virtual parameter q, which means QoS of instrument and q_i is the QoS of instrument i in the pool. The pool adjusts dispatch probability p_i according to user appraisal Q to the experiment. Both variables q and Q are fuzzy variables. Vague linguistic values like *terrible*, *bad*, *normal*, *good* and *excellent* are used to express user appraisals towards experiments in equipment grid.

When a user submits an experiment with detailed specifications in JDF to equipment grid, service pool alliance will dispatch this job to the corresponding service pool or pools. Every related pool selects a resource to run the experiment or part of it. If instrument i in a pool is selected, q_i has value as one of the following linguistic values, *very bad*, *bad*, *normal*, *good* and *very good*. In many cases, q_i with *very good* value has a large probability to receive *excellent* value of Q, *good* to *good*, *normal* to *normal*, *bad* to *bad* and *very bad* to *terrible*. We can represent this relationship as a function $\boldsymbol{f}$ mapping from fuzzy values to fuzzy values, which will be demonstrated in Section 3. Because the value q of a specific experiment is not known by service pool beforehand and can only be reflected by user appraisal towards his experiment. Service pool will adjust dispatch probability p_i to make instruments with *good* or *excellent* appraisal higher utilization ratio.

The value of p_i is proportional to the expected value of fuzzy random variable $preq_i$, which is an estimated value of fuzzy value q_i, as shown in Equation (2). We define QoS_i equals to the expected value of $preq_i$, that is $QoS_i = E[preq_i]$. The reason why $preq_i$ instead of q_i is used in Equation (2) is that service pool has no prior information about q_i and has to estimate what value it may be through former user appraisals.

$$p_i = E[preq_i] / \sum_{i=1}^{N} E[preq_i] \qquad (2)$$

$$preq_i = \begin{cases} "Verygood" & with\ \ probability\ \ {}^1prep_i \\ "Good" & with\ \ probability\ \ {}^2prep_i \\ "Normal" & with\ \ probability\ \ {}^3prep_i \\ "Bad" & with\ \ probability\ \ {}^4prep_i \\ "VeryBad" & with\ \ probability\ \ {}^5prep_i \end{cases} \quad (3)$$

In Equation (3), ${}^kprep_i$ ($1 \le k \le 5$) means the probability that $prep_i$ takes kth fuzzy value. The initial values of ${}^kprep_i$ equal to 0.2. $preq_i$ is a fuzzy random variable (see [22, 23]). In the following example, expected value of $preq_i$ can be calculated by Equation (4).

$$E[preq_i] = \sum_{k=1}^{5} {}^kprep_i \times c_i \quad (4)$$

In Equation (4), c_i is defined to be the center of $preq_i$´s membership function. When the membership function changed, the expected value of $preq_i$ will also be different. The method to adjust $preq_i$ is that when user appraisal is *excellent,* ${}^1prep_i$ will increase. If *good*, ${}^2prep_i$ will increase; *normal*, ${}^3prep_i$ will increase; *bad*, ${}^4prep_i$ will increase and *very bad*, ${}^5prep_i$ will increase.

When a service chain consisting of multiple instruments is needed to conduct an experiment, Q is user appraisal towards the whole service chain. This is a common case that QoS of instruments in a service chain are not consistent. Users can only give a comprehensive appraisal for the whole chain. We assume that user appraisal to a service chain is determined by the worst QoS of instruments in the chain, which can be expressed in (5).

$${}_1QoS \wedge {}_2QoS \wedge {}_3QoS \wedge \cdots\cdots \wedge {}_DQoS \quad (5)$$

In (5), ${}_i$QoS means QoS of the ith instrument in a service chain and subscript D means the total number of instruments in the chain for a given experiment. Symbol $\wedge$ means the worst QoS in two instruments.

System will use the worst QoS of all instruments in a service chain to adjust p_i in all pools that participated in this experiment. The dispatch probability adjustment algorithm in [12] does not make good use of provenance information. It may lead to instruments with high QoS to have less chance to be scheduled if these instruments worked coordinately with some poor QoS instruments and got poor appraisals. In the following section, an improved algorithm using provenance information is proposed.

3. Scheduling algorithm

In this section, we introduce an algorithm which uses provenance information to estimate QoS of instrument from past user appraisals. Figure 1 demonstrates the model in a service pool when provenance information is used.

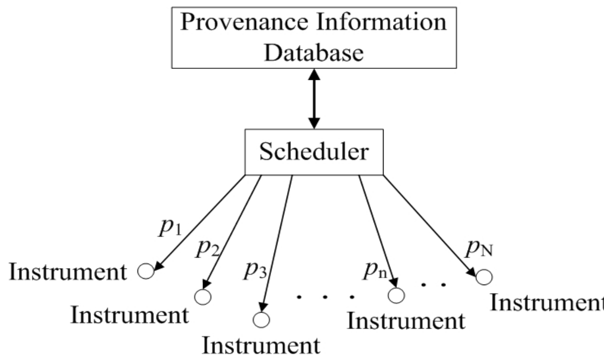

Figure 1. Scheduling resources in service pool using provenance information

In Figure 1, provenance information database is used to store provenance information, such as job submission time, job finish time, URI of service providers, user appraisals, etc. Typical provenance information can be represented in Equation 6.

Provenance(*R*) = {*T*, *C*, *P*, *input*/*Provenance*(*input*), *output*, *RS*, *UA*, *A*} (6)

In Equation (6), R is a result and *Provenance*(R) is the provenance information of R. T is the time spot that this provenance information is recorded. C is the client, P is the service provider, *input* and *output* are the input data and output data of this process that produced R. When the input is an intermediate result of another process, *Provenance*(*input*) is used. *RS* means the resource states and related operations. *UA* is user appraisal and A records some supplementary information.

How to deduce QoS of instruments from provenance information is the key contribution of this algorithm.

For the purpose of demonstration, we first define a partial order relationship among the following fuzzy linguistic values in (7) and (8):

$$\text{terrible} < \text{bad} < \text{normal} < \text{good} < \text{excellent} \quad (7)$$

$$f(\text{terrible}) < f(\text{bad}) < f(\text{normal}) < f(\text{good}) < f(\text{excellent}) \quad (8)$$

In (8), $\boldsymbol{f}$ is the function from QoS of instrument to user appraisal introduced in Section 2.

The flow chart of our proposed algorithm is shown in Figure 2.

The four steps in Figure 2 constitute the sketch of this scheduling algorithm. Details of these four steps are as follows.

Step 1: This step sets initial probability of all instruments to be scheduled in a service pool. The scheduler in the pool has no prior information about which instrument has more satisfactory QoS. Thus all instruments have equal probability to be scheduled. If there are N instruments in this service pool, then all these instruments have probability of $1/N$ to be scheduled.

Step 2: This step selects instrument i according to probability p_i. It is clear that instruments with larger p have more chance to be scheduled.

Step 3: This step estimates QoS of instruments from user appraisals. As mentioned, it is assumed that appraisals are determined by the worst QoS of all instruments involved. When a resource participated in several service chains and received different appraisals, the best appraisal is used to reflect QoS of this instrument, which can be easily proven by Proof 1.

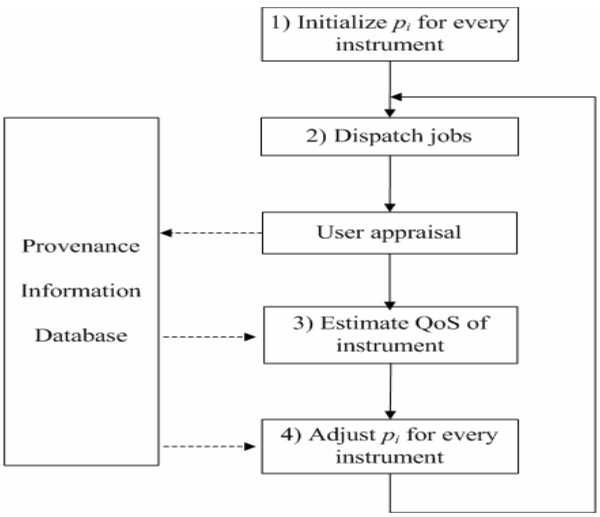

Figure 2. Flow chart of scheduling algorithm

Proof 1:

Suppose instrument k has participated in different service chains for m times. Service chain i consists of D_i (i□[1, m]) instruments and has received appraisal AP_i ($i \in [1, m]$).

According to the assumption that user appraisal for a given service chain is determined by the worst QoS of all instruments contained in the chain, we have the following (9).

$$AP_i = f(\min({}_1^iQoS_l, {}_2^iQoS_m, ..., {}_J^iQoS_k, ..., {}_{D_i}^iQoS_j)) \le f(QoS_k) \quad (i \in [1, m]) \tag{9}$$

In (9), AP_i is user appraisal to service chain i. QoS_k is QoS of instrument k, which is unknown. ${}^i{}_J QoS_k$ means QoS of instrument k as the Jth step in service chain i.

According to (9), we get following:

$$\max(AP_1, AP_2, ..., AP_i, ...AP_m) \le f(QoS_k) = \overline{AP(k)} \quad (i \in [1, m])$$

(10)

In (10), $\overline{AP(k)}$ is the appropriate appraisal that instrument k deserves. From (10) it is clear that instruments usually cannot get their appropriate appraisals.

End

Step 4: This step adjusts probability p_i in service pool according to the estimated QoS and provenance information. Instruments with better QoS have more chances to be scheduled. The algorithm to adjust p_i is demonstrated in Algorithm 1. In Algorithm 1, we suppose that instrument k in this service pool has been scheduled and received user appraisal AP.

Algorithm 1:

```
switch (AP)
  {
  case " excellent " :
        deltaP1 = dP11;
        Q = " very good " ; break;
  case " good " :
      deltaP1 = dP12; Q= " good " ; break ;
  case " normal " :
      deltaP1 = dP13 ; Q = " normal " ; break ;
  case " bad " :
      deltaP1 = - dP14; Q = " bad " ; break ;
  case " terrible ":
      deltaP1 = - dP15;
      Q = " very bad " ; break ;
  default :
     deltaP1 = 0 ;
  }
 if (Q > QoSk)
  {
   QoSk = Q ;
  }
switch (QoSk )
{
  case " very good " :
         deltaP2 = dP21;  break ;
  case " good " :
         deltaP2 = dP22 ;  break ;
  case " normal " :
         deltaP2 = dP23;  break ;
  case " bad " :
        deltaP2 = - dP24 ;  break ;
  case " very bad " :
        deltaP2 = - dP25;  break ;
  default :
        deltaP2 = 0 ;
```

```
    }
    deltaP = deltaP1 × deltaP2 ;
    for ( int i = 0 ; i < N ; i++)
      { pi = (1- deltaP ) × pi ; }
    pk = pk + deltaP ;
  Store AP into Provenance Information Database
End
```

In the above algorithm, instrument with high QoS and related service chain newly providing high quality service have more chance to be scheduled in the future. dP_1i (i=1,2,3,4,5) and dP_2i (i=1,2,3,4,5) are numerical values that affect the outcome of this algorithm. If these values are too large, some instruments in the pool will have no chance to be scheduled regardless of what QoS these instruments have. If they are too small, the probability for instruments with high QoS will be small, which leads to minor enhancement of user appraisals.

If a new instrument joins a service pool, it has the average probability to be scheduled. Algorithm 2 is used to adjust probabilities in a service pool when a new instrument joins in.

***Algorithm 2*:**

```
  for ( i = 1; i < N +1; i++)
   { pi = pi * N / (N + 1) ;}
     N = N + 1 ;
     pN = 1 / N ;
End
```

When an instrument stops providing services in its service pool, all dispatch probabilities in the pool are adjusted according to Algorithm 3. In Algorithm 3, instrument k is supposed to be leaving its pool.

Algorithm 3:

```
  totalP = 0 ;
  pk = 0 ;
  for ( i=1; i <=N ; i++ )
     { totalP = pi + totalP ; }
  for ( i = 1; i <= N ; i++ )
     { pi = pi / totalP ; }
  for (i = k ; i <N - 1 ; i++)
     { pi = pi+1 ; }
  N = N –1;
End
```

Algorithm 3 allows instruments without any jobs running on it to quit this system. Any instrument running jobs is not permitted to leave. If it leaves by some inevitable reason, the pool will record this instrument as unstable and it will have a much lower probability to be scheduled when next time it wants to join this pool.

Let's assume a service chain comprising of n instruments is needed to finish a specific experiment. The number of instruments providing different QoS in each pool is the same. Distribution of user appraisals can be expressed in Equation (11).

$$P(AP) = \begin{cases} 1-0.8^n & (\text{"terrible"}) \\ 0.8^n - 0.6^n & (\text{"bad"}) \\ 0.6^n - 0.4^n & (\text{"normal"}) \\ 0.4^n - 0.2^n & (\text{"good"}) \\ 0.2^n & (\text{"excellent"}) \end{cases} \qquad (11)$$

From appraisal distribution shown in (11), it is obvious that equipment grid provides relatively low QoS. To provide higher QoS for users, [12] provides an algorithm, but its deficiency lies in that many instruments with high quality might be scheduled with low probability. Table 1 shows distribution of what appraisal instruments with different QoS might have in an n-instruments service chain.

Table 1 Distribution of user appraisals to instruments of different QoS

	very bad	*bad*	*normal*	*good*	*very good*
terrible	1	$\frac{1-0.8^{n-1}}{1-0.6^{n-1}}$	$\frac{1-0.8^{n-1}}{1-0.4^{n-1}}$	$\frac{1-0.8^{n-1}}{1-0.2^{n-1}}$	$1-0.8^{n-1}$
bad	0	$\frac{0.8^{n-1}-0.6^{n-1}}{1-0.6^{n-1}}$	$\frac{0.8^{n-1}-0.6^{n-1}}{1-0.4^{n-1}}$	$\frac{0.8^{n-1}-0.6^{n-1}}{1-0.2^{n-1}}$	$0.8^{n-1}-0.6^{n-1}$
normal	0	0	$\frac{0.6^{n-1}-0.4^{n-1}}{1-0.4^{n-1}}$	$\frac{0.6^{n-1}-0.4^{n-1}}{1-0.2^{n-1}}$	$0.6^{n-1}-0.4^{n-1}$
good	0	0	0	$\frac{0.4^{n-1}-0.2^{n-1}}{1-0.2^{n-1}}$	$0.4^{n-1}-0.2^{n-1}$
excellent	0	0	0	0	0.2^{n-1}

From Table 1, we see that instruments with high QoS receive relatively low rank appraisals, being considered as low QoS and have less chance to be scheduled. In this proposed algorithm, resources will eventually have appropriate probability to be scheduled according to their QoS.

4. Experimental results

In this section three cases are given to illustrate the scheduling algorithm. Simulation platform is developed using JSP and Javascript. The Web server is Tomcat 5.5 and service container is Axis 1.4. Ten service pools with 500 instruments are simulated by services deployed in 10 computers of P4 2.4GHz CPU and 1GB memory.

In the first case, many simple experiments which require a single instrument are submitted to pool alliance. A specific service pool with N instruments can conduct these kinds of experiments. In this case, N equals to 50. 10 of the instruments have *very good* QoS and may receive user appraisal of *excellent*, 10 *good*, 10 *normal*, 10 *bad* and 10 *very bad*.

Figure 3 is the simulation result when user feedback information is used to adjust dispatch probabilities of instruments in 100,000 such experiments. The vertical axis represents the number of jobs and the horizontal axis represents user feedback information in terms of vague values. The results without probability adjustment using the proposed scheduling algorithm are also given in Figure 3.

In Figure 3, the parameters in parameter set 1 are $dP_1 1=0$, $dP_1 2=0$, $dP_1 3=0$, $dP_1 4=0.01$, $dP_1 5=0.02$, $dP_2 1=0$, $dP_2 2=0$, $dP_2 3=0$, $dP_2 4=0.001$, $dP_2 5=0.002$ and in parameter set 2 are $dP_1 1=0$, $dP_1 2=0$, $dP_1 3=0$, $dP_1 4=0.01$, $dP_1 5=0.02$, $dP_2 1=0$, $dP_2 2=0$, $dP_2 3=0$, $dP_2 4=0.01$, $dP_2 5=0.02$, respectively.

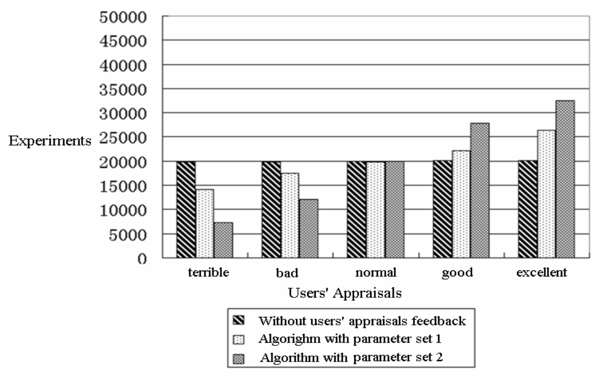

Figure 3. Results of user appraisals for 100,000 experiments

As shown in Figure 3, when user feedback information is considered, instruments with poor QoS have less chance to be scheduled. Owners of these instruments should improve QoS of their instruments by decreasing the price or shortening the execution time, if they want to have a high dispatch probability.

In the second case, three instruments in different pools constitute a service chain to complete an experiment. An application scenario is that, when a fossil's formation age and components need to be uncovered, the following experiment is required. This kind of experiment can be conducted with a service chain consisting of three instruments, which are a microscope, image processor and mass spectrograph. Figure 4 is the demonstration of this service chain. There are N_1 microscopes, N_2 image processors and N_3 mass spectrographs in three pools, respectively. Final user appraisal to this service chain is decided by ${}_1q_i \wedge {}_2q_j \wedge {}_3q_k$, when instrument i in one pool works coordinately with instruments j and k in the other two pools. In this case N_1 , N_2 and N_3 equal to 50.

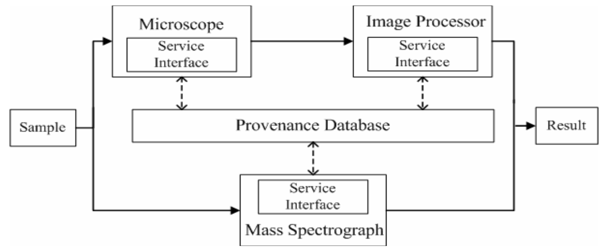

Figure 4. The service chain for a specific experiment

Figure 5 includes simulation results when feedback information is used to adjust dispatch probability in each service pools. Horizontal axis represents user appraisals to a service chain. Vertical axis means the number of user appraisals. Three different columns represent results from the following three conditions: simulation of random scheduling without user feedback information, simulation using feedback information with probability adjustment algorithm proposed in [12], and simulation using provenance information with probability adjustment algorithm proposed in this paper respectively.

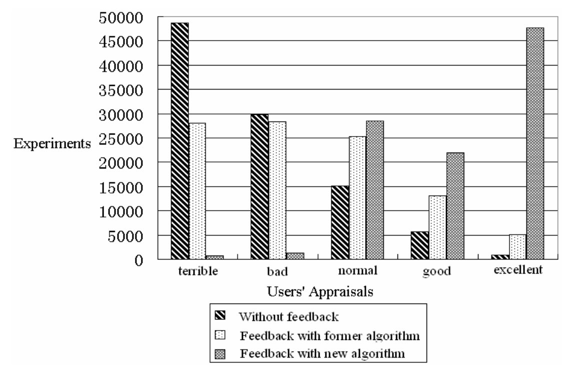

Figure 5. Simulation of user appraisals for service chains consisting of three instruments

In the third case, ten instruments constitute a service chain to provide service. The number of instruments in these ten pools is N_1, N_2,..., N_{10}, respectively and each equals to 50. Figure 5 is the simulation results in this case.

In Figure 5 and 6, the first circumstance does not take user feedback appraisals into account. In this case, distribution of user appraisals agrees with Equation (11). When feedback information is used, less appraisals ranking in *bad* or *terrible* appeared. In the second and third circumstances, many better user appraisals appeared when provenance information is used to schedule in service pools. The second circumstance in Figure 5 and 6 are situations when the algorithm in [12] is used with proper parameters chosen. Its improvement on user appraisals is not satisfactory while in the third case with this new algorithm, instruments with high QoS have more chances to be used and more user appraisals rise to *good* and *excellent*. As a whole, QoS of equipment grid is improved.

In Figure 5, there are more than 60 percent user appraisals in the *terrible* and *bad* level in the first circumstance. 35 percent of user appraisals rank in the *terrible* and *bad* level in the second circumstance, while only 1 percent in the third circumstance.

In Figure 6, the percentages of the three circumstances in *terrible* and *bad* level are 99, 87 and 2 percent. While in *good* and *excellent* level, they are 0, 0.3 and 85 percentage, respectively.

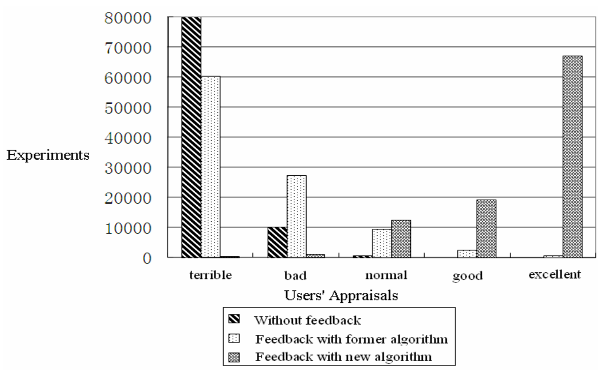

Figure 6. Simulation of user appraisals for service chains consisting of ten instruments

5. Related works

Currently, provenance is mainly used in some data intensive projects and related research interests focus on data provenance, like Lineage Information Program [13], Chimera [14], myGrid [15], Collaboratory for Multi-scale Chemical Science (CMCS) [16], Provenance Aware Service-oriented Architecture (PASOA) [17], Earth System Science Workbench (ESSW) [18] and Tioga [19]. The survey of data provenance is introduced in [20].

Traditional scheduling problems are 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 the resources in a distributed heterogeneous environment had been shown, in general, to be NP-complete [24]. 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 [25]. Simulation results showed that the GA has the best result but with too much calculation time while Min-min heuristic has best overall performance. While in grid environment, resources change dynamically, and all the static heuristics need slight adjustment. Eight dynamic scheduling methods, which include five on-line mode heuristics and three batch model heuristics, were compared in [26] and KPB heuristic outperformed the other four on-line dynamic heuristics and Min-min heuristic had the best performance than the other two batch model heuristics.

These methods are not effective when multiple objects are to be optimized. The work in [12] presented a fuzzy parameter, QoS to represent the multi objects and try to adjust resource dispatch probability using user feedback appraisals. In this work this method using provenance information got extended to service chain in equipment grid.

6. Conclusions

The main contribution of this paper is the proposal of a scheduling algorithm using provenance information, which takes user QoS feedback information into account and provides more satisfactory instruments for users in equipment grid. The algorithm provided in this work to increase scheduling probability of instruments with high QoS and decrease usage of those with low QoS, is demonstrated by simulation to be effective in equipment grid environment.

Ongoing work includes an information service providing detailed instrument and experiment data, a service acts as workflow enactor to manage experiments involving multiple instruments, a mechanism to ensure reliable and coherent of different user appraisals, management of provenance information and its storage and a layered security mechanism for authentication and authorization of remote resource access.

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, 2007BC807901 and the 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), 200-222.

[3] D. E. Atkins, K. K. Droegemeier, S. I. Feldman, H. Garcia-Molina, M. L. Klein, D. G. Messerschmitt, P. Messina, et. al., *Revolutionizing Science and Engineering through Cyberinfrastructure*, National Science Foundation Blue – Ribbon Advisory Panel on Cyberinfrastructure, January 2003.

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

[5] Y. Wang, L. Liu, C. Wu and W. Ni, "Research on Equipment Resource Scheduling in Grids", *Grid and Cooperative Computing,* LNCS Vol. 3251, Springer Verlag, 2004, 927-930.

[6] 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), 73-80.

[7] W. Smith, V. Taylor and I. Foster, "Predicting Application Run Times Using Historical Information", *Job Scheduling Strategies for Parallel Processing,* LNCS Vol. 1459, Springer Verlag, (1998), 122-142.

[8] W. Smith, V. Taylor and I. Foster, "Using Run-Time Predictions to Estimate Queue Wait Times and Improve Scheduler Performance", *Job Scheduling Strategies for Parallel Processing,* LNCS Vol. 1659, Springer Verlag, (1999), 202-219.

[9] M. Faerman, A. Su, R. Wolski and F. Berman, "Adaptive Performance Prediction for Distributed Data-Intensive Applications", in *Proceedings of ACM/IEEE Supercomputing Conference*, 1999.

[10] L. Yang, J. M. Schopf and I. Foster, "Conservative Scheduling: Using Predicted Variance to Improve Scheduling Decisions in Dynamic Environments", in *Proceedings of ACM/IEEE Supercomputing Conference*, 2003.

[11] D. P. Spooner, S. A. Jarvis, J. Cao, S. Saini and G. R. Nudd, "Local Grid Scheduling Techniques Using Performance Prediction", *IEE Proceedings – Computers and Digital Techniques*, 150(2), (2003), 87-96.

[12] J. Yin, J Cao, YX Wang etc, "Scheduling Remote Access to Scientific Equipments in Cyber-infrastructure for Education and Research", *International Symposium on Cluster Computing and the Grid 2007.*

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

[14] I. T. Foster, J. S. Vöckler, M. Wilde, and Y. Zhao, Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation, SSDBM, (2002), 37-46.

[15] R. D. Stevens, A. J. Robinson, and C. A. Goble, myGrid: personalised bioinformatics on the information grid, Bioinformatics, 19, (2003), 302 -304.

[16] J. Myers, C. Pancerella, C. Lansing, K. Schuchardt, and B. Didier, Multi-Scale Science, Supporting Emerging Practice with Semantically Derived Provenance, Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, 2003.

[17] P. Groth, M. Luck, and L. Moreau, A protocol for recording provenance in service-oriented Grids, OPODIS. Grenoble, France, 2004.

[18] J. Frew and R. Bose, Earth System Science Workbench: A Data Management Infrastructure for Earth Science Products, SSDBM, (2001), 180-189.

[19] A. Woodruff and M. Stonebraker, Supporting Fine-grained Data Lineage in a Database Visualization Environment, ICDE, (1997), 91-102.

[20] Buneman P, Khanna S, Tan WC, Why and where: A characterization of data provenance , LECTURE NOTES IN COMPUTER SCIENCE.1973, (2001), 316-330.

[21] Yuexuan. Wang, Cheng Wu: Study on Equipment Interoperation Chain Model in Grid Environment, Lecture Notes in Computer Science, 3758, (2005), 588-595.

[22] B. Liu and J. Peng, *A Course in Uncertainty Theory*, Tsinghua University Press, Beijing, 2005.

[23] Y. K. Liu and B. Liu, "Fuzzy Random Variables: a Scalar Expected Value Operator", *Fuzzy Optimization and Decision Making*, 2(2), (2003), 143-160.

[24] D. Fernandez-Baca, "Allocating modules to processors in a distributed system", IEEE Trans. Software Energy. 15(11), 1989, 1427-1436.

[25] T. D. Braun, H. J. Siegel, and N. Beck et al, "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), 2001, 810-837

[26] M. Maheswaran, S. Ali, H.J. Siegel, et al. "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, 1999, 30-44.