DOI : 10.3745/JIPS.2008.4.3.103

Policy Adjuster-driven Grid Workflow Management for Collaborative Heart Disease Identification System

Shengzhong Deng*, Chan-Hyun Youn*, Qi Liu*, Hoe Young Kim*, Taoran Yu* and Young Hun Kim*

Abstract: This paper proposes a policy adjuster-driven Grid workflow management system for collaborative healthcare platform, which supports collaborative heart disease diagnosis applications. To select policies according to service level agreement of users and dynamic resource status, we devised a policy adjuster to handle workflow management polices and resource management policies using policy decision scheme. We implemented this new architecture with workflow management functions based on policy quorum based resource management system for providing poincare geometry-characterized ECG analysis and virtual heart simulation service. To evaluate our proposed system, we executed a heart disease identification application in our system and compared the performance to that of the general workflow system and PQRM system under different types of SLA.

Keywords: *Grid Worfklow, Collaborative Healthcare Platform, SLA, Policy Adjuster*

1. Introduction

There have been many attempts to create healthcare system or platform by combining the existing medical service with IT such as sensor and communication technology. In other words, the general health clinical data and information are exchanged electronically, and the patient, doctor, hospital and laboratories can be connected through communication network to perform medical practices. Such healthcare system is enabled and supported by IT, yet the core technology will be high-speed information exchange technology and high performance computing technology in distributed environment. Grid technology can meet these requirements by bringing heterogeneous resources together and allocating them efficiently to applications.

Medical applications are no longer restricted to one medical institute. Some collaborated applications like heart disease simulator, an instance of workflow jobs, require several resources work together. So the healthcare platform requires collaborative job scheduling function based on resource status which is the core part of Grid resource management system [2][3]. However, the complexity of resource management for QoS guarantee in Grid significantly increases as the grid computing grows. One of the promising approaches trying to solve this problem is policy based resource management system [2] that is suitable to the complex management environments. To manage resource in Grid, it is required to have a sort of grid resource broker and scheduler that manage resources and jobs in the virtual Grid environment. Authors in [3] suggested policy-based resource management system (PQRM) architecture by consideration of Grid Service Level Agreement (SLA) and abstract architecture of grid resource management.

In medical healthcare environment, beside most of single-program applications, there exist many applications which require co-process of many programs following a strict processing order. Since these applications are executed on Grid, they are called Grid workflows. Current Grid resource management systems do not consider complexity of collaborative medical applications and service level guarantee at workflow level. In this paper, we propose Grid workflow-driven healthcare platform for collaborative applications.

In the sense of optimum management, PQRM [5] provides optimum resource quorum for applications, so we develop workflow-Integrated PQRM as one instance for this workflow-driven healthcare platform. PQRM's available resource quorum is optimum resource set in resource management point of view and our workflow-driven system can provide optimum resource set in workflow management point of view. Two management philosophies require a policy decision scheme to adjust based on user's SLA. We evaluate the proposed workflow-driven healthcare platform by measuring the completion time of collaborative heart disease simulator under different types of user SLA.

Manuscript received July 11, 2008; accepted September 17, 2008.
Corresponding Author: Chan-Hyun Youn
* Information and Communication University, Daejon, Korea ({deng, chyoun, liuqi, hoeyoung, taoran, mystical}@icu.ac.kr)

2. Previous Works

2.1 Collaborative Medical Platform

Computed-based medical system for physiological diagnosis has been attempted. The procedure from sensing or monitoring of the patient's condition to processing of the sensed result values is the main trend of medical system development. Additionally, combined with grid technologies, real-time processing and huge amount of data combination and management using high-performance computing power is also being researched. [6][7]. However, much of the existing development was limited to the analysis of the results of sensing the patient's conditions. There has been almost no research to create a synthetic platform that could diagnose through processing the sensed results and support combination of other diagnosis methods from the cooperation with hospital. Hence, Physio-Grid that supports integration of many diagnosis methods and supports data integration that is specific to medical system through web-service.

2.2 Virtual Heart Disease Simulator

Virtual hearts refers to computer-generated program and database that emulates the biological characteristics of real heart in figure and function. In terms shape, the real heart's geometrical data (MRI or CT data) are used, whilst electronic structure of cardiac muscle cells, ventricle and atrium cells are interpreted using bio-domain method generally[4]. In term of function, the solution for differential equation that expresses biological characteristics of cardiac cells and tissues that consist of heart is used to emulate the real heart's clinical change. Virtual aims to analyze with computer physiological or pathological characteristics of heart. Virtual experiments that cannot be done in clinic or in experiment may be carried out on simulation.

2.3 Policy Quorum based Resource Management

To manage resources in a Grid environment, it is required to have a sort of grid resource broker and scheduler that manages resources and jobs. The research in [3] suggested policy-based resource management system architecture based on grid SLA. For guarantying QoS, it defines the quorum of QoS parameter; the set based on guarantying QoS and suggested the resource allocation mechanism via well-defined algorithm. And the research in [5] suggested the scheme for the reset of quorum for guarantying QoS, coping with variation of resource status based on evaluation of quorum in time domain. In PQRM, resource reconfiguration performs the reshuffling of the current available resource set for maintaining the quality level of the resources.

3. Model Description of Grid Workflow-driven Healthcare Platform

To provide high quality of medical services related with heart disease, a collaborative healthcare platform is needed for heart disease identification with Poincare Geometry-Characterized ECG analysis and virtual heart simulation.

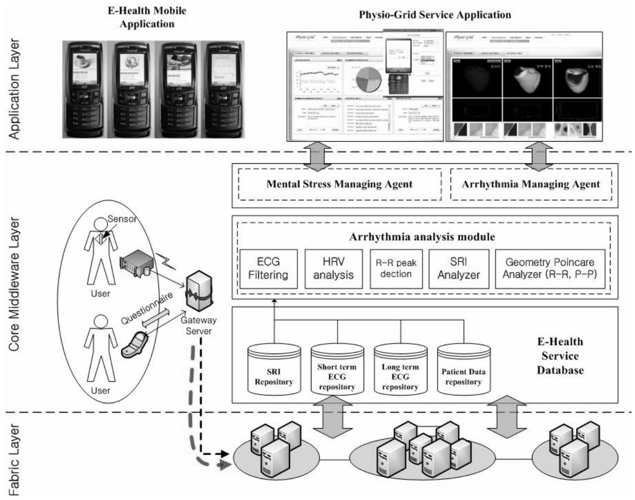


Fig. 1. Collaborative heart disease care service

Collaborative heart disease care services shown in figure 1 can be mainly classified to two service applications; mental stress management agent and arrhythmia syndrome management agent. For heart rate related mental stress identification, medical doctor needed to know heart rates and all Stress Response Inventory (SRI) scores at the same time. Then the medical doctor sought physiological relationships between heart rates and each SRI score to identify mental stress factors which may induce heart rate changes. However, mental stress level and heart diseases such as tachycardia or bradycardia are needed to care continuously with close patient observation. The causes of arrhythmia can be expressed by more than 30 reasons. Verifying the relationships between arrhythmia and causes in quantitative analysis requires deep understanding of its characteristic and cooperation of different analysis modules.

3.1 Architecture of PQRM based Workfow-driven Healthcare Platform

To provide optimum job scheduling solutions based on users' SLA for collaborative heart disease care services presented in figure 2, we propose a new architecture of Grid workflow-driven healthcare platform with policy adjuster integrated based on PQRM system [5]. To adjust different policies according to the requirements in SLA, policy adjuster is developed to manage resource mapping policies for sub-jobs in the workflow. To handle each application in environment processing huge amount of data and signal, we adopt policy quorum based resource management technologies and design this system to manage job execution and heterogeneous resources.

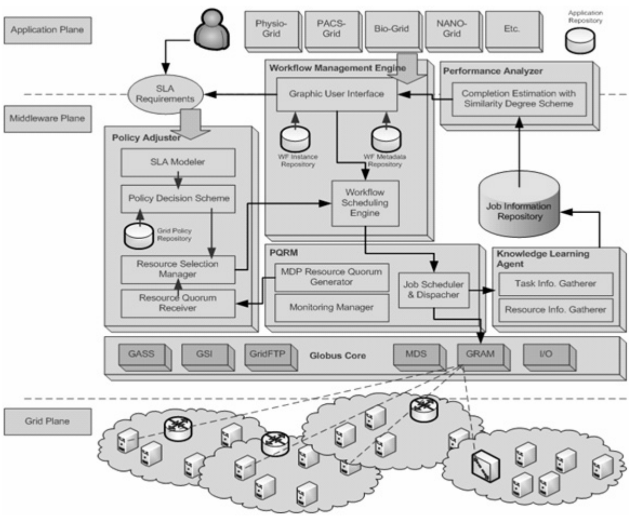


Fig. 2. Architecture of Grid workflow-driven healthcare platform with policy adjuster integrated

Figure 2 shows layered architecture of policy adjuster integrated workflow management system for collaborative healthcare services. Doctors and patients build their applications of collaborative healthcare services in the graphic user interface. Performance analyzer provides the prediction finish time for the collaborative healthcare services using completion estimation with similarity degree scheme by analyzing the historical data. According to the prediction information, users can set up their SLA requirements. The application requests are then submitted and parsed into tasks and dependencies in workflow management engine. Policy adjuster analyzes the input SLA and maps the tasks of workflow in time using policy decision algorithm to achieve the optimized mapping solution in terms of the SLA requirements. After receiving the mapping information from policy adjuster, workflow scheduling engine allocates tasks to resources and creates task managers to manage and monitor the tasks. PQRM components are responsible to monitor the resource status, generate the Available Resource Quorum (ARQ) and dispatch the jobs in Grid environment

Main functions in our system are listed as follows:

Workflow Scheduling Engine: is a manager for the tasks in the workflow to allocate each task to the specific resource selected by policy adjuster and monitor the job execution status in its lifetime.

Policy Decision Scheme: receives SLA constrains from users and determines a specific system policy using in this execution for all the workflow sub-jobs to find a best mapping solution according to users' intention.

Completion Estimation with Similarity Degree Scheme: receives the task name and the resource as input, analyzes the historical executions and predicts the probable runtime of the next execution.

Comparing with general workflow management systems [17], our proposing system is more advanced and efficient in the following aspects. Firstly, we provide many advanced functions for collaborative healthcare services such as geometry poincare analyzer and 3-D virtual heart simulator; secondly, future runtime of a job can be predicted in our system using completion estimation with similarity degree algorithm by analyzing the historical data; thirdly, the execution status of each task on Grid resource can be monitoring and shown to users; finally, SLA requirements of users can be guaranteed by adjusting policies. Users can select time optimized execution policy or cost optimized execution policy as their intentions.

3.2 Service Scenario of Collaborative Healthcare Platform

In this section, a medical scenario shows how a physician might treat a patient to identify the heart disease by collecting the information from ECG analysis and simulating the data using Virtual Heart Simulator. The service description of advanced ECG processing and Virtual Heart Simulation is shown in a scenario of figure 3. The application begins when a patient goes to the doctor with heart disease suspicion. Firstly, doctor acquires ECG signals by sensing the heart bit rate of the patient. If high (HR>110 [BPM]) or low heart rates (HR<60 [BPM]) is detected, medical doctor needs to ask the SRI questions to find heart rate related stress factors for more detailed and accurate diagnosis. He may also need to the collaborative diagnosis with other medical doctors on this symptom. The sensitivity analysis of poincare geometry-characteristics with PP and RR index ratio is processed. The transmitted SRI responses and ECG signals are stored at the databases in medical centers. Then if there are patients' or any customer's heart rate changes, medical doctors may start identification process whether major factor is mental stress, heart disease, or arrhythmia from regression analysis.

Based on the ECG processing, doctor decides to utilize the Virtual heart simulation to virtualize the ECG sensing data stored in the database of medical centers. The computing of virtual heart simulation would be separated in the distributed heterogeneous resources in the Grid environment.

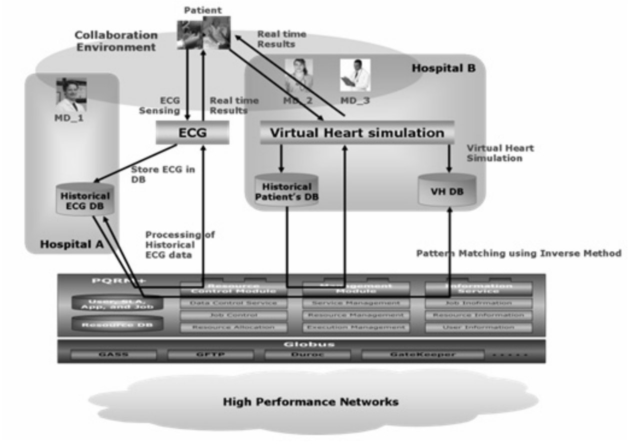


Fig. 3. Service description of advance ECG processing and Virtual Heart Simulation

Before the medical doctor executes collaborative healthcare application in our workflow-driven platform, he may need to set up the requirements for different service levels in SLA such as time optimized service or cost optimized service. In our workflow management system, we are now considering two main factors: finish time and cost. We set $SLA = \{ft, co\}$, where $0 \le ft, co \le 1$. Each factor has a real number value, and the closer one value means the higher intent factor in SLA for a user.

In this scenario, the doctor wants to get the visualized result of virtual heart as soon as possible, so he sets the ft to 1. Then a suitable policy is selected by policy adjuster to guarantee the fastest finish time. Finally, the result of simulation is provided to doctors. In this way, patients with heart disease can receive more visualized and accurate diagnosis in the minimal time.

3.3 Completion Estimation with Similarity Degree Scheme in Performance Analyzer

In order to provide the runtime prediction of collaborative healthcare services, completion estimation with similarity degree scheme is developed in the performance analyzer of our proposed system.

Definition 1. Tuple $I = \langle u_{cpu}, u_{mem}, b_{bd}, t_c \rangle$ is one of the execution instances of U, where u_{cpu} is the cpu usage, u_{mem} is the memory usage, b_{bd} is the bandwidth, and t_c is the completion time.

For the task i and resource j, we retrieve the instances set of all the past instances which have been executed in our system.

Definition 2. Let $\{z_1, ..., z_m\}$ be subsets of U, $z_1, ..., z_m \in U$, $z_1 \cup z_1 \cup ... \cup z_m = U$ and for every $z, z' \in S$, $z \cap z' = \phi$. Each $z \in U$ is called a zone in our system.

The maximum finish time, the minimum finish time and the total number of instances can be found. Then we divide the time period between $T_{\max}$ and $T_{\min}$ into M zones. All the instances will be distributed into zones in terms of their finish time values. The attribute of zone z is presented in the form of $\langle \hat{c}, \hat{m}, \hat{b} \rangle$, where $\hat{c}, \hat{m}, \hat{b}$ stand for average cpu usage, average memory size and average bandwidth of the instances in this zone z respectively.

Lemma 1. The property of zone $z(k)$ is calculated as:

$$\langle \hat{c}_k, \hat{m}_k, \hat{b}_k \rangle = \left\langle \frac{1}{|z|} \sum_{n=1}^{|z|} u_{cpu,n}, \frac{1}{|z|} \sum_{n=1}^{|z|} u_{mem,n}, \frac{1}{|z|} \sum_{n=1}^{|z|} b_{bd,n} \right\rangle$$

Lemma 2. The similarity degree $d(k)$ between the current resource status $\langle c_c, m_c, b_c \rangle$ and resource status $\langle \hat{c}_k, \hat{m}_k, \hat{b}_k \rangle$ in each zone is defined as:

$$d(k) = \sqrt{w_c \left(\hat{c}_k - c_c\right)^2 + w_m \left(\hat{m}_k - m_c\right)^2 + w_b \left(\hat{b}_k - b_c\right)^2}$$

where c_c, m_c, b_c are the current cpu usage, memory size and bandwidth of resource j and w_c, w_m, w_b are the weight values.

Theorem 1. The prediction time of a task *i* on resource *j* is the average finish time of instances in the zone *k*, which has the lowest similarity degree $d(k)$ among all the zones.

Proof. The runtime of task *i* is mainly determined by the capability and the status of resource *j*. For same resource, the capability is always same. The resource status is represented by the zone property $\langle \hat{c}_k, \hat{m}_k, \hat{b}_k \rangle$. If the similarity degree $d(k)$ of zone *k* is lower, which means the Euclid distance between current resource status and past resource status of this zone is smaller, the current resource status is more similar as the instances of the zone *k*, the probable runtime is closer to the average runtime of the instances in zone *k*.

3.4 Policy Adjuster Mechanism of Collaborative Healthcare Platform

When the SLA in a workflow side such as minimized completion time or minimized cost is considered, resources within the ARQ provided by PQRM may become infeasible. Thus, In order to guarantee users' requirements in SLA by adjusting system policies, we derive a policy adjuster mechanism for collaborative healthcare platform.

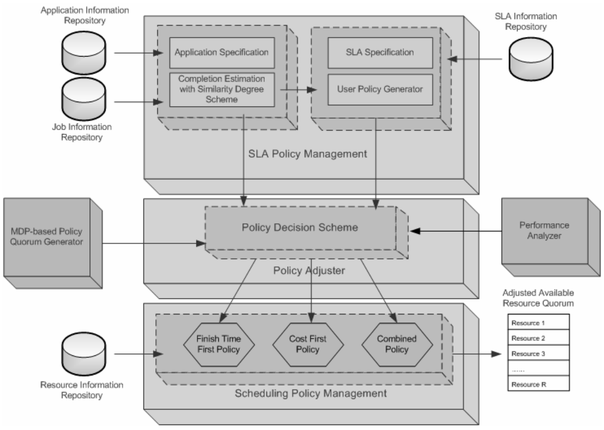


Fig. 4. Policy management architecture of policy adjuster

As shown in figure 4, in the user domain, application specification is composed and completion estimation with similarity degree scheme is invoked to predict the future runtime of the application on each resource. Based on the runtime prediction, users can construct their SLA specification as their intentions. According to user policy, policy decision scheme selects the corresponding scheduling policy, such as finish time first policy, cost first policy or combined policy, for mapping the job to the Grid resources. Finally, an adjusted available resource quorum is generated for the job scheduling.

Definition 3. Tuple $\langle ft, co\rangle$ is SLA constrains defined by users, where *ft* is the finish time factor and *co* is cost; $0 \le ft, co \le 1$ and $ft + co = 1$.

Definition 4. Cost of a resource is determined by the resource capability and resource status. Namely, the resource with better capability and better status would cost more. It is defined as follows:

$$c_k = w_c \cdot cpu_k \cdot I_{cpu_k} + w_m \cdot mem_k + w_s \cdot st_k + w_b \cdot bd_k$$

Where cpu_k, I_{cpu_k}, mem_k, st_k and bd_k are cpu capability, idle rate of cpu, memery, storage and bandwidth of k[th] resource respectively. w_c, w_m, w_s and w_b are weight valuses. Total cost of workflow is the sum of cost of all the sub-jobs.

When policies are applied to management system, the conditions to differentiate policies should be defined. In our system, the rank $r_k(t)$ of resource k can be calculated by different policies according to different SLA requirements.

$$r_k(t) = \begin{cases} t_k(t), & ft = 1 \\ \alpha \cdot t_k(t) + \beta \cdot c_k(t) \cdot \mu, & 0 < ft < 1 \\ c_k(t), & ft = 0 \end{cases}$$

where $r_k(t)$ is the rank of the k[th] resource in the ARQ list, $t_k(t)$ is the prediction finish time and $c_k(t)$ is the cost of resource. α equals to *ft* and β equals to *co*. μ is a coefficient which equals $\mu = \frac{\mathrm{E}[t_k(t)]}{\mathrm{E}[c_k(t)]}$, where $\mathrm{E}[t_k(t)]$ and $\mathrm{E}[c_k(t)]$ are expectation of the prediction completion time and cost of the resource k, respectively.

Three kinds of policies have been presented. The finish time first policy is valid when the factor *ft* in SLA equals to 1, which means user wants to finish the task as soon as possible. The cost first policy is used when user evaluate the cost factor as 1. When finish time factor *ft* is between 0 and 1, combined policy of finish time and cost is adopted for resource selection.

ALGORITHM Policy Adjuster Scheme

INPUT: $SLA = \{ft, co\}$ from user, ARQ from PQRM.
OUPUT: A feasible resource with smallest rank value.
BEGIN:
If *co* equals 1 **then**
While the next resource is not null **do {**
Calculate the cost $c_k(t)$ for resource k;
$r_k(t) = c_k(t)$;
break;
}
else if *ft* equals 1 **then**
While the next resource is not null **do {**
Get the finish time $t_k(t)$ for resource k from similarity degree based zone calculator;
$r_k(t) = t_k(t)$;
break;
}
else then
Calculate the cost and finish time for each resource;
If μ is null **then** $\mu = \frac{\mathrm{E}[t_k(t)]}{\mathrm{E}[c_k(t)]}$;
While the next resource is not null **do {**
Calculate the rank $r_k(t) = ft \cdot t_k(t) + co \cdot c_k(t) \cdot \mu$,
}
Set μ to null; get AARQ; sort the AARQ in terms of rank value;
end if
Provide the resource with smallest rank;
END

Fig. 5. Algorithm of policy adjuster scheme

We generate Adjusted Available Resource Quorum (AARQ) by calculating the rank value $r_k(t)$ of all the resources in ARQ, and assign the task on the resource which has the smallest rank value.

Based on the idea presented above, we propose a policy adjuster algorithm as shown in figure 5.

4. Implementation and Experiment

We evaluate policy adjuster-driven workflow management system's performance by using a collaborative health disease identification application. We evaluate the general system, PQRM system[5] and our proposed policy adjuster-driven workflow management under different user's SLA types. The general system submits the job randomly because it does not consider resource management. PQRM system use allocation cost to select the resources. Policy Adjuster-driven workflow management system has three submission strategies: Finish Time Optimization, Cost Optimization and mapping considering both Time and Cost together. Policy adjuster is used to differentiate policies for those different submission strategies. The sub jobs will be mapped to our resource pool based on different scheduling policies. We monitor the completion time of job and resource cost under these three different systems.

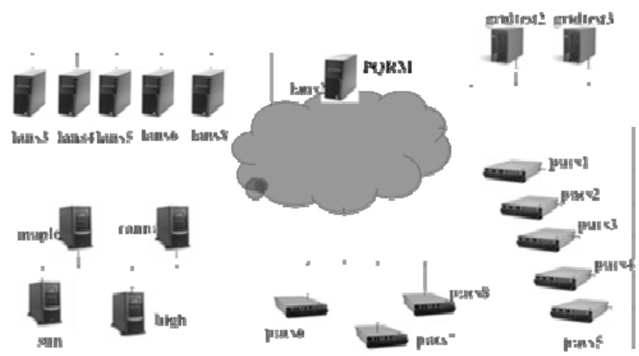


Fig. 6. Experiment Network Topology

As shown in figure 6, in our experiment environment there are several different types of resources. lans series resources are high performance HP cluster machines; pacs 1-5 series resources are general IBM servers; pacs 6-8 are high performance IBM servers with high price; gridtest resources are general desktop computers with low price; maple, canna, sun and high are high performance duo-core cluster machines. Table 1 shows the detail description of resources.

As listed in table 2, In general workflow system, there is no SLA guarantee mechanism and best effort mapping solution is used. In PQRM system, allocation cost is used as a metric to select the optimal resources.

In our proposed system, since the value of SLA factor is from 0 to 1, we use a form of SLA with these two factors: finish time and cost, which can represent three different resource mapping policies: finish time first policy, cost first policy and combined policy of finish time and cost.

Finish time first policy stands for time optimized mapping solutions; on the other hand, cost first policy stands for cost optimized mapping solutions; combined policy means that users consider both the completion time and cost at the same time.

Table 1. List of Resource Pool

Hostname	O.S.	CPU	RAM	I/F Card
lans1.icu.ac.kr	Red hat Enterprise 4 AS 32bit	Intel Pentium D 3.0 GHz	1GB	100Mbps
lans2.icu.ac.kr				
lans3.icu.ac.kr				
lans4.icu.ac.kr				
lans5.icu.ac.kr				
lans6.icu.ac.kr				
lans8.icu.ac.kr				
pacs1.icu.ac.kr	Red hat Enterprise 4 AS 32bit	Intel Pentium D 2.8 GHz	1GB	100Mbps
pacs2.icu.ac.kr				
pacs3.icu.ac.kr				
pacs4.icu.ac.kr				
pacs5.icu.ac.kr				
pacs6.icu.ac.kr		Intel Pentium D 3.2 GHz		
pacs7.icu.ac.kr				
pacs8.icu.ac.kr				
sun.icu.ac.kr	Red hat Enterprise 4 AS 32bit	Intel Core 2 Duo 2.20GHz, 2.20GHz	2GB	100Mbps
high.icu.ac.kr				
maple.icu.ac.kr				
canna.icu.kr				
gridtest1.icu.ac.kr	Red hat Enterprise 4 AS 32bit	Intel Celeron 2.80GHz	512MB	100Mbps
gridtest2.icu.ac.kr				
gridtest3.icu.ac.kr				

Table 2. List of SLA Types with related Policies

System	SLA strategy		Policy	Note
	Finish time	Cost		
General Workflow System	-	-	-	Best effort
PQRM System	-	-	-	Allocation cost optimization
Policy Adjuster integrated Workflow Management System	1	0	Finish Time First Policy	Time optimization
	0	1	Cost First Policy	Cost optimization
	0.1	0.9	Combined Policy of Finish time and cost	Care cost much more than finish time
	0.3	0.7		Care cost a little more than finish time
	0.5	0.5		Care cost and finish time as the same importance
	0.7	0.3		Care finish time a little more than cost
	0.9	0.1		Care finish time much more than cost

- Task monitoring

Task monitoring function is provided for each task running in our system. Users can be noticed about the running status in the graphic interface. To get the information about the running tasks, a communication mechanism between task managers and workflow scheduling engine should be determined. The figure 7 shows the communications between gridtest3 which is the host for workflow scheduling engine (WSE) and pacs6 which is running a task manager for a sub-job.

We can see that pacs6 (black line) interacts with gridtest3 periodically. Approximately, every 6s task manager should notice to WSE with the task status event. The first conversation package is the largest since the prerequisites for the execution on pacs6 should be sent to the task manager. Each time the job status changes, the color of the symbol will change to a corresponding one.

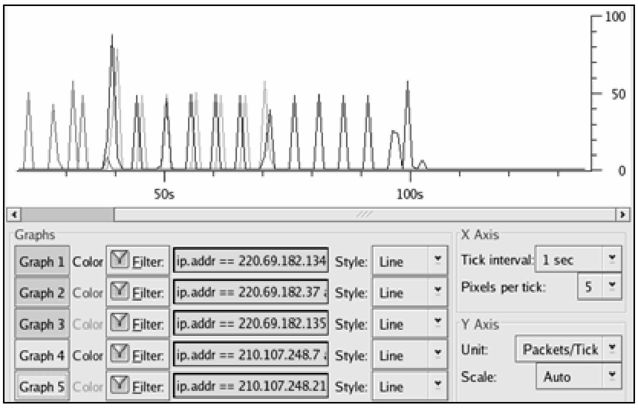


Fig. 8. Communications between WSE and task manager

- Time optimization by finish time first policy

Time optimization is an important service type that can provide shortest execution time for the applications. When the finish time factor in SLA is set to 1, the finish time first policy is selected to achieve time optimization execution for this application. As shown in figure 8, we measure the total execution time of ECG application under four different strategies.

In the experiment, we measure the total execution time of the application which runs on different systems. The broken line stands for the execution time in general workflow management system without any resource mapping policy. The green value is the finish time of PQRM system. Apparently, when we use finish time first policy to execution the application, we can achieve the shortest finish time comparing with other three cases. Cost first policy using in our system has the worst finish time because the costless resources are selected in the executions.

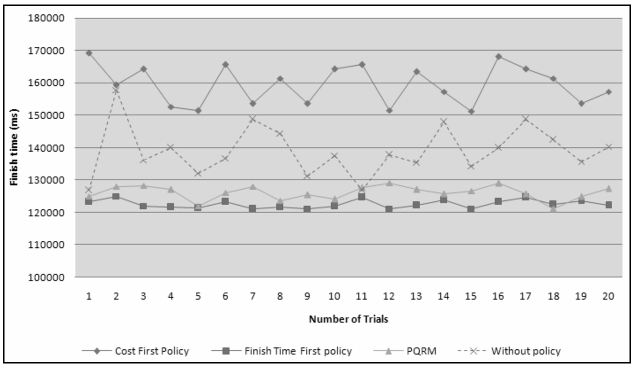


Fig. 9. Completion Time of different policies

- Cost optimization by cost first policy

Cost optimization is another service type in our proposed system. When the cost factor in SLA is set to 1, the cost first policy is selected by our system to achieve cost optimization execution for this application.

As shown in figure 9, we measure the total cost of the application which runs on different systems. Apparently, cost first policy achieves the lowest average cost for the applications. It can reduce approximate 24% of cost comparing to the PQRM system which consumes highest cost among all the situations. That is because, with cost first policy, users care about cost only, the costless resources in the resource pool are selected. On the other hand, the cost of finish time first policy is much higher than the case without policy.

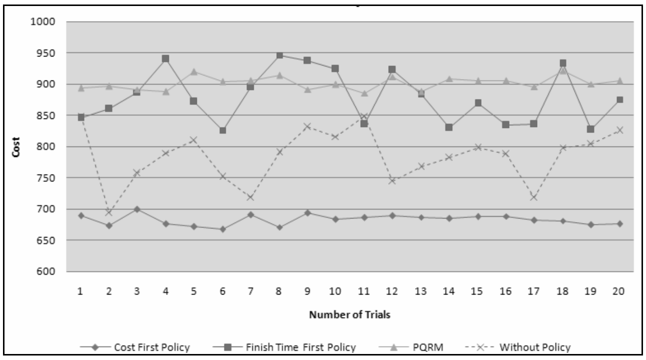


Fig. 10. Cost performance of different policies

- Consider both finish time and cost by combined policy

Besides time optimization and cost optimization strategies, users can construct other SLA types in our proposed system by setting the finish time factor and cost factor with other values expect 1 and 0. In these cases, combined policy of finish time and cost is valid for mapping the sub-jobs to the suitable resources to meet users' intention. The differences between SLA values are shown in figure 10.

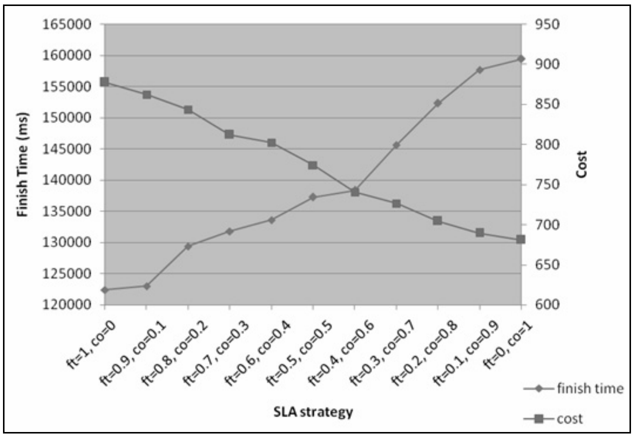


Fig. 11. Finish time and cost under different SLA strategies

Different values of *ft* and *co* stand for the way user evaluates the finish time and cost. The higher value of *ft* means user cares finish time more than cost, the higher value of *co* means user care cost more than finish time. As we can see, when the *ft* value in SLA is decreasing, the finish time of the application is increasing from 122000 to almost 160000. At the same time, the cost is decreasing since the value of *co* in SLA increases. That means users can control the finish time and cost by setting the value of *ft* and *co* in SLA, bigger *ft* brings shorter finish time and smaller *co* leads to higher cost of the application.

As we shown above, our proposed Grid policy adjuster-driven workflow management system can provide different policies according to different users' requirements so that service level agreement can be guaranteed. When users set finish time factor in SLA to 1, our system can select the finish time first policy to provide minimized completion time scheduling service; when users set cost factor in SLA to 1, our system chooses cost first policy to provide minimized cost resources set; when users consider both time and cost, our proposed system also can provide stable service that meets users' requirements using combined policy.

5. Summary

This paper proposes a policy adjuster-driven workflow management system for healthcare platform, which supports collaborative health disease identification applications. To guarantee the SLA of users' requirements, we implemented a new architecture that integrates workflow functions with policy adjuster based on policy quorum based resource management system. We derived a policy adjuster to handle workflow management polices and resource management policies. Based on policy decision scheme, resource selection can be controlled according to service level agreement of users and dynamic resource status. We evaluate this platform through a collaborative heart disease identification application and compare the performance to that of the general workflow system and PQRM system under different types of SLA. The result shows that time optimization can be achieved by finish time first policy and the cost can be reduced by approximate 24% comparing to PQRM system when the cost first policy is used. When the combined policy is adopted, users can control the finish time and cost by setting the factor values in SLA. Therefore the proposed management system can guarantee quality of service for workflow applications based on different user's requirement and dynamic resource status.

References

[1] K Krauter, R Buyya, M Maheswaran, "Taxonomy of Resource Management in Grid", Software Practice and Experience, 2002

[2] K Yang, A Galis, C Todd, "Policy-based active Grid management architecture", ICON 2002. 10th IEEE International

[3] Dong Su Nam et al., "QoS-constraint Quorum Management scheme for a Grid-based Electrocardiogram", LNCS 3043 pp.352-359, 2004.

[4] Eun Bo Shim et al., "Development of a 3D virtual heart including the mechanism of human cardiac cell physiology-electric conduction of cardiac tissue-heart mechanics", The Korean Society of Mechanical Engineers 60th anniversary fall conference 2005, The Korean Society of Mechanical Engineers, pp58-58, 2005.

[5] Sang-Il Lee, YoungJoo Han, Hyewon Song, Changhee Han, Chan-Hyun Youn, "Grid Policy Administrator with MDP-Based Resource Management Scheme", Proc. The 2006 Int. Conf. on Grid Computing & Applications, pp. 25-31, June 2006

[6] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver and K. Glover, M.R. Pocock, A. Wipat, and P. Li, "Taverna: a tool for the composition and enactment of bioinformatics workflows". Bioinformatics, 20(17):3045-3054, Oxford University Press, London, UK, 2004.

[7] Ian J. Taylor et al., "Workflows for e-Science", chap. 13. Oxford: Clarendon, 1892, pp.190–191.

[8] T. Tannenbaum, D. Wright, K. Miller, and M. Livny, "Condor - A Distributed Job Scheduler", Beowulf Cluster Computing with Linux, The MIT Press, MA, USA, 2002.
[9] J Yu, R Buyya 1, "A Taxonomy of Scientific Workflow Systems for Grid Computing", Workflow in Grid Systems Workshop, GGF-10, Berlin, March 9, 2004
[10] Christos G. Cassandras, "Discrete Event Systems: Modeling and Performance Analysis", Aksen Associates Incorporated Publishers, 1993.
[11] www.accessgrid.org
[12] http://www.anl.gov/Media_Center/logos21-2/grid.htm
[13] http://www.library.uams.edu/accessgrid/aghome.aspx
[14] http://cci.uchicago.edu/projects/abc/
[15] L.Altintas et al., "a framework for the design and reuse of grid workflows," Porc. Scientific Appications of Grid computing, LNCS 3458, Springer, 2005, pp. 119-132
[16] E. Neubauser et al., "Workflow-based grid applications, " Future Generation Computer System, Jan. 2006, pp. 6-15
[17] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, "Scientific Workflow Management and the KEPLER System". Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows, to appear, 2005

Shengzhong Deng
He received the BS degrees in Information Engineering from Beijing University of Posts and Telecommunications in 2006. He joined LANS Lab to continue his graduate study in ICU

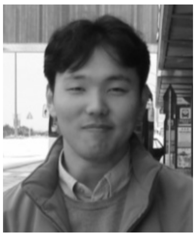
Chan-Hyun Youn
He received the BS and MS degrees in electronics engineering from Kyungpook National University, Daegu, Korea, in 1981 and 1985, respectively. He also received a PhD in electrical and communications engineering from Tohoku University, Japan, in 1994. He served in the Korean Army as a Communications Officer, First Lieutenant, from 1981 to 1983. Before joining the University, from 1986 to 1997, he was a Leader of High-Speed Networking Team at Korea Telecom (KT) Telecommunications Network Research Laboratories where he had been involved in the research and development of a centralized switching maintenance system, MOVE, and HAN/B-ISDN network testbed. Especially, he was a Principal Investigator of high-speed networking projects including an ATM technical trial between KT and KDD, Japan, Asia—Pacific Information and Communications University, Daejeon, Korea. He also was a Visiting Scholar at MIT, Cambridge, USA in 2004. He is a Vice President of Grid Forum Korea. Currently, he is interested in the Grid middleware, high performance routing, multicasting, optical Internet, and network performance measurement. He was a recipient of IEICE PAACS Friendship Prize, Japan, in 1994. He is a Member of IEEE, IEICE, KICS and KISS.

Qi Liu
He received the BS degrees in Information Engineering from Beijing University of Posts and Telecommunications in 2006. He joined Communication Network Lab to continue his graduate study in BUPT. At 2007, he came to Advanced Network and System Architecture Lab of ICU in Korea as exchange student.

Hoe Young Kim
He received the BS degrees in Computer Science from Chung-Ang Univ in 2007. He has been master student at Information and communications University since 2007. His research interests include Grid system, Service Oriented Architecture and eHealth System

Taoran Yu

She received the BS degrees in Information Engineering from Beijing University of Posts and Communications in 2006. She has been a master student at the same university since 2006. Her research interests include mobile IPv6, Next Step in Signaling and Grid computing.



Younghun Kim

He received the BS degrees in Computer Engineering from Korea University of Technology and Education in 2006. He has been master student at Information and communications University since 2007. His research interests include Grid Computing, and Medical Image oriented Data Grid middleware.