

QoS-Constrained Resource Allocation for a Grid-Based Multiple Source Electrocardiogram Application

Dong Su Nam^{1,5}, Chan-Hyun Youn^{1,3}, Bong Hwan Lee²,
Gari Clifford³, and Jennifer Healey⁴

¹ School of Engineering, Information and Communications University
103-6 Munji-dong, Yusong-gu, Daejeon 305-732, Korea
{dsnam, chyoun}@icu.ac.kr
<http://vega.icu.ac.kr/~highnet>

² Dept. of Information and Communications Engineering, Daejeon University
Daejeon 300-716, Korea
blee@dju.ac.kr

³ Harvard-MIT Division of Health Science Technology, MIT,
Cambridge, MA 02139, USA
{chyoun, gari}@mit.edu

⁴ Dept. of Translational Medicine, Harvard Medical School/BIDMC,
Boston, MA 02215, USA
jhealey@bidmc.harvard.edu

⁵ Dept. of Information Assurance, National Security Research Institute,
52-1 Hwaam-dong, Yusong-gu, Daejeon 305-348, Korea
dsnam@etri.re.kr

Abstract. QoS-constrained policy has an advantage to guarantee QoS requirements requested by users. Quorum systems can ensure the consistency and availability of replicated data despite the benign failure of data repositories. We propose a Quorum based resource management scheme, which resource Quorum includes middleware entity and network entity, both can satisfy requirements of application QoS. We also suggest a heuristic configuration algorithm in order to optimize performance and usage cost of Resource Quorum. We evaluate both simulations and experiments based on the electrocardiogram (ECG) application for health care, because this application requires transferring giga-bytes of data and analyzing complicated signal of ECG. Simulation results show that network capabilities are more important than computing capabilities, as both sizes of transferred data and computation task increases. Experimental results show that our scheme can reduce the total execution time of ECG application by using proposed heuristic algorithm compared to policy based management scheme.

1 Introduction

Grid Computing have proposed to extend distributed computing infrastructure for advanced science and engineering. Researchers have made much progress in constructing such an infrastructure and extending and applying it to a broad range of

computing problems. As a result, “grid” has entered the computer science vocabulary to denote middleware infrastructure, tools, and applications concerned with integrating geographically distributed computational resources. [1] Grid computing lets networked computers share and apply their resources toward solving a given problem [2]. Grid technology must therefore include a function that can divide pieces of a problem over multiple computers and then integrate the resulting partial solutions. It must also be able to provide naming, security, and data retrieval functions across multiple computers.

In this paper, we propose QoS-constrained Quorum configuration scheme for Reliable Resource Management. In order to apply the Quorum based model to our scheme, we define elements in Quorum as QoS constrained middleware entities and network entities, represented as a binary form. Proposed QoS Quorum means the condition including QoS entities required by user and Resource Quorum means the resource including entities that can satisfy the condition of QoS Quorum. Therefore Resource Quorum, mapped with QoS Quorum, can guarantee user’s QoS requirements. We also suggest a heuristic configuration algorithm in order to optimize performance and usage cost of Resource Quorum. We carry out both simulations and experiments based on the electrocardiogram (ECG) application for health care, because this application needs transferring lots of data and analyzing complicated signal of ECG.. Simulation results show that network capabilities are more important than computing capabilities, as both sizes of transferred data and computation task increases. Experiment results show that our scheme can reduce the total execution time of ECG application by using proposed heuristic algorithm compared to policy based management scheme,. We expect that this result can be contributed to take advantage of more reliable resources in Grid.

2 Model Description for QoS-Constrained Resource Entity

A variety of entities exist in heterogeneous environments such as computational Grids. The entities of resources can be divided as middleware resources and network resources. Middleware resource entities are related to a single computer system and include the CPU speed, memory size, storage capacity and I/O devices. Network resource entities, such as bandwidth, delay and jitter, are based on a source-destination pair. We define the entity of both QoS requirements and resources, are considered system middleware and network elements from various Grid resources. Our resource management model is based on some assumptions that simplify the problem formulation:

- ♣ QoS dimensions have a one to one correspondence to system resources metrics.
- ♣ System resources are modeled as limited buckets of capacity. The total resource utilization in the system cannot exceed the available amount.
- ♣ A network resource is modeled as a limited bucket associated to a pair of between broker or client users and system resources.
- ♣ Network links are bi-directional. Connections in both directions share the same network resources.

- ♣ Resources are independent of each other.
- ♣ Resources are not probabilistic and the system guarantees the contracted QoS.

2.1 Optimization Problem of Application QoS Constraints

An application, \mathbf{A} , can be represented by an undirected graph as a function of the computing tasks and communication relations as comprising a number S , of tasks, T , such that $\mathbf{A} = \{T_1, \dots, T_s\} = \{ \langle V_r^k, E_r^{kl} \rangle | V_r^k \text{ and } E_r^{kl} = T^k \times T^l, k \neq l \text{ and } k, l = 1, \dots, s \}$.

V_r^k means the vertices of each computing node and E_r^{kl} means the edge for the communication between V_r^k and E_r^{kl} . l means all communication peers related to the V_r^k .

We similarly define an available resource universe comprising n resource entities, $\mathbf{R} = \{R_1, \dots, R_n\}$. Which can also be represented as a undirected graph

$$R = \{ \langle V_r^k, E_r^{kl} \rangle | V_r^k \text{ and } E_r^{kl} = R^k \times R^l, k \neq l \text{ and } k, l = 1, \dots, n \}$$

The QoS request is described by vector matrices \mathbf{MQ}^M and \mathbf{MQ}^m for the middleware constraints, and also \mathbf{NQ}^M and \mathbf{NQ}^m define the maximum, respectively, minimum network resource requirements for each connection (i, j) , for which tasks T_i and T_j communicate. For the middleware constraints:

$$\mathbf{MQ}^M = (mq_{ij}^M)_{i=1..k, j=1..s}, \mathbf{NQ}^M = (nq_{ij}^M)_{i=1..k, j=1..s} \tag{1}$$

$$\mathbf{MQ}^m = (mq_{ij}^m)_{i=1..k, j=1..s}, \mathbf{NQ}^m = (nq_{ij}^m)_{i=1..k, j=1..s} \tag{2}$$

where i is the number of entities and j is the number of resource nodes. This resource model assumes that there is a path in the network between any two nodes and that all resource allocations for connections are independent.

The optimal allocation is designed to maximize the application QoS requirements defined as a function of the combined QoS ratio for all tasks. The overall application QoS requirement is a linear combination of the middleware task utility and the network task utility. The middleware utility of task T_s for resource R_i is mu_i^s and the network utility of task T_s between resource R_i and R_j is $nu_i^s \cdot nu_j^s$. Maximizing the followed function satisfied application QoS is the optimization of the middleware entity utility and the network entity utility, where s is the number of tasks

$$Q(f) = \sum_{i=1}^n mu_i^s + \sum_{i=1}^n nu_i^s \cdot \sum_{j=1}^n nu_j^s, i \neq j \tag{3}$$

2.2 Quorum Vectors

An assertion is a set of QoS attributes that are required for service delivery and agreed on with the service requester. Each service requester must specify its QoS requirements for the resource manager. Therefore QoS Quorum can be determined from SLAs containing user’s resource on demand that is composed of middleware and

network resource requirements. The QoS Quorum is configured of binary forms that mean “must satisfied minimum QoS requirements for resource entities = 1” or “don’t care satisfied minimum QoS requirements for resource entities = 0” [8]. The QoS Quorum can be defined as: a QoS Quorum set $Q_{QoS} = \{Q_{MQ}, Q_{NQ}\}$ is a collection of subsets $Q_{MQ}, Q_{NQ} \subseteq U$ of a finite universe U . Both QoS quorum Q_{MQ} and Q_{NQ} consists of middleware QoS vectors network QoS vectors.

Similarly we define a resource Quorum as the subset of resources satisfying the conditions required by the QoS Quorum. The resource manager allocates this subset of Grid resources to the application to allow the application tasks to be executed. Resource Quorum can be defined as, $Q_R = \{Q_{MR}, Q_{NR}\}$ consisting of both middleware resource vectors (m) and network vectors (N). If a specified entity can satisfy the correspondent QoS requirement it can be “1” otherwise “0”.

3 Heuristic Optimization Scheme in Resource Allocation

Basically, Resource Quorum set has the characteristics to guarantee minimal QoS requirements as defined in the previous section. We can select the specified available resources in Resource Quorum set. First of all, minimal QoS constraints created by the SLAs make up two groups of vectors in QoS Quorum and Resource Quorum. QoS Quorum is made for the service class correspondent with one of QoS services. Simultaneously, Resource Quorum is determined from whether satisfying QoS constraints or not. After the configuration of two types of Quorum, we can create Resource Quorum sets for guaranteeing each QoS service. We suggest a heuristic algorithm to maximize performance through capabilities of resources and to minimize the usage cost of resources. The sum of both entities middleware and network in a resource can be represented as either the capability or the usage cost of the resource. As the sum of entities is large, the resource can improve its capacity. Also, as the sum of entities is small, the usage cost of the resource is less. When we allocate superior resources in a Resource Quorum set to Grid applications, we can expect resultant reductions in both computation time and communication time. Therefore we can optimize the performance and the usage cost, by using the sum of entities in resources allocated in Resource Quorum set. At this time, either the middleware entity or network entity is given precedence. We will show the impact as the precedence of entities changes using simulation. We assume that all Grid applications could be downloaded from the application repository. Each task of an application is executed at distributed resources. To minimize the usage cost of resources, we select the resource having the lowest value of summation. Likewise to maximize the performance of computing, we select the resources having the highest value of summation. Note that all of resources satisfy the minimum QoS requirements. We assume that all Grid applications could be downloaded from the application repository. Each task of an application is executed at distributed resources. To minimize the usage cost of resources, we select the resource having the lowest value of summation. Likewise to maximize the performance of computing, we select the resources having the highest value of summation. Note that all of resources satisfy the minimum QoS requirements. The performance maximization procedure could

minimize the execution time by allocating application tasks to resources having superior capabilities. When users want to execute their applications in the deadline time, we can apply this scheme for the task scheduling. In order to minimize execution time of tasks, we select sequentially the resources by sorting in order of the maximum values of vector sum and then we configure Resource Quorum. The usage cost minimization scheme is to reduce usage cost of resources by allocating application tasks to resources having poor capabilities. When users want to execute their applications with the lowest budget, we can apply this scheme for the task scheduling. In order to minimize the usage cost of resources, we select sequentially the resources by sorting in order of the minimum values of vector sum and configure Resource Quorum. Although these resources have some inferior middleware and network capabilities, these resources are able to satisfy the minimal QoS requirements.

4 Performance Evaluations

To evaluate the proposed reliable resource management scheme, we simulate it with Simgrid and discuss some measurement results in international Grids.

4.1 Simulation Using Simgrid

To analyze the proposed scheme, we have used the useful tool, Simgrid [8], which provides core functionalities that can be used to build simulators for the study of application scheduling in distributed environments. The scenario 1 shows that the randomly selected Quorum satisfying the minimum QoS requirements is allocated to Grid application and the scenario 2 shows that the Quorum both satisfying the minimum QoS requirements and having better CPU capabilities than network capabilities is allocated the Grid application. The scenario 3 shows that the Quorum both satisfying the minimum QoS requirements and having better network capabilities than CPU capabilities is allocated Grid application. In the simulation results, we have known that the computation capabilities affected the execution time only for the small size of data such as 3Mbytes or 30Mbytes. However, as the size of data became larger such as 300Mbytes, the effect of network capabilities became more important. Figure 1 shows that network capabilities more important than computing capabilities as the size of data and the amount of computation tasks are large. The result also showed that the proposed Quorum configuration scheme could improve the performance, compared with the random configuration of resources.

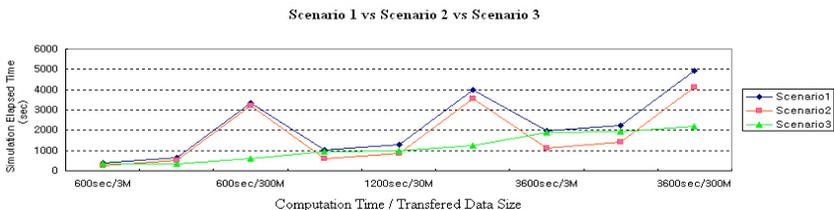


Fig. 1. Total elapsed time (sec) of scenario 1, 2, and 3

4.2 Experiment Results

The testbed to implement for this experiment have shown in Figure 2, which consists of five systems at Information and Communications University (ICU) and each one system at Hanyang University, Korea University and MIT in USA. Also, we have used nine DBs for sub tasks such as $\text{subjob1}=\{\text{DB1, DB2, DB3}\}$, $\text{subjob2}=\{\text{DB4, DB5, DB6}\}$, and $\text{subjob3}=\{\text{DB7, DB8, DB9}\}$ among the 72 MIMIC DBs which contains ECG signals [7].

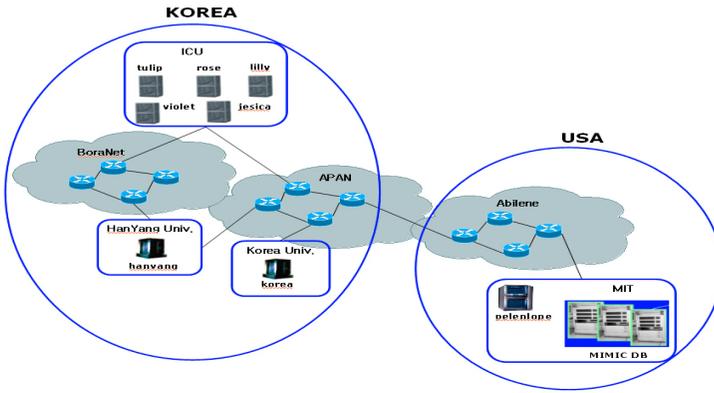


Fig. 2. Grid testbed for the ECGs experiments

Minimum QoS requirement is obtained from the SLAs requested by user. The PQRM generates the service classes of QoS Quorum such as QoS-1, QoS-2, and QoS-3. Three of QoS service classes are assumed as the Guaranteed Service, the Controlled load Service and the Best effort Service, respectively. In the QoS Quorum, the network requirement is classified into two parts since it needs two links to perform the ECG applications, one is the link between PQRM and resources and the other is the link between the resources and MIMIC DB. Figure 3 shows the creation of QoS Quorum. Minimum QoS requirement determines the Available Resources Vector for each resource. The measurement was performed with the Network Weather Service (NWS) [3]. 8 systems with Linux O/S were used in this experiment, which are 5 systems located in the Local Area, and 3 systems in the Wide Area. Each value of vectors is determined owing to the satisfaction of the minimum QoS requirement. After the creation of the Resource vector, Resource Quorum set is generated from QoS Quorum. In Figure 3, we can see that there is no resource to satisfy QoS-1 service. Instead, we knew the resources satisfying QoS-2 and QoS-3 service. In our experiments, the resources satisfying QoS-2 service have become four systems such as ICU1, MIT, ICU4, and ICU5. Also, the resources satisfying QoS-3 service have become eight systems such as HYU, KU, MIT, ICU1, ICU2, ICU3, ICU4, and ICU5.

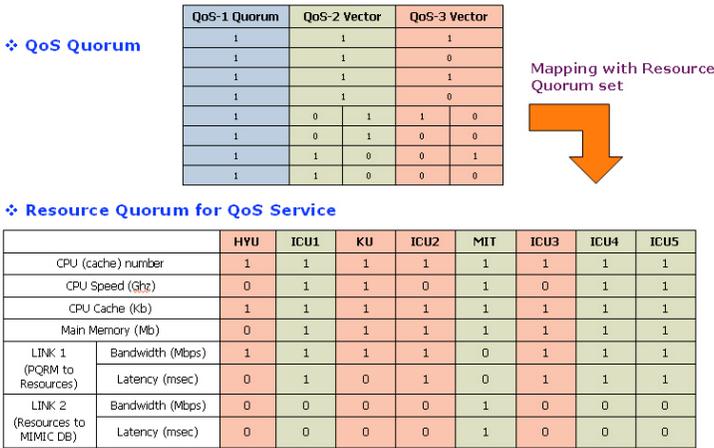


Fig. 3. The creation of Resource Quorum set for each QoS Quorum

To guarantee QoS-3 service we also have configured ten Resource Quorums and each Resource Quorum consists of three resources selected from eight resources such as ICU1, ICU2, ICU3, ICU4, ICU5, HYU, KU, and MIT. Resource Quorums from Quorum-1 to Quorum-8 were configured randomly. The Quorum-9 (Min_Cost Quorum) was configured by cost minimization configuration scheme that select in order of the minimum value of resource vectors and the Quorum-10 (Max_Perf Quorum) was configured by performance maximization configuration scheme that select in order of the maximum value of resource vectors. The total execution time of the ECGs application at the Quorum-10 is smaller than at other Quorums. Also Quorum-9, the Min_Cost Quorum, reduce total execution time better than Q-4, Q-5, Q-6, Q-7, and Q-8. Note that all of the Quorums satisfy the minimum QoS requirements. Figure 4 shows the comparison of execution time for the ECGs application.

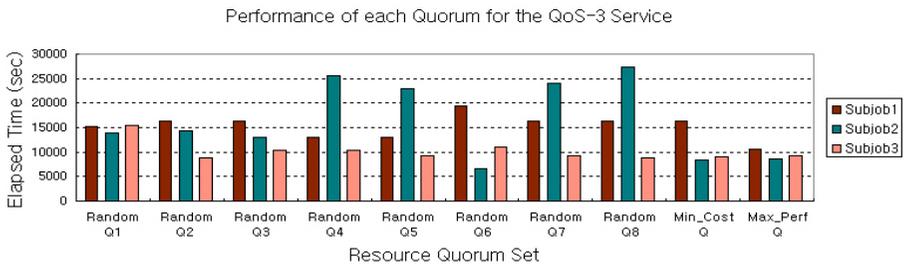


Fig. 4. Experiment results for QoS-3 service

5 Conclusions

The management of the overall Grid system itself in a flexible way is becoming more and more important. However, this is a relatively new field that has so far been understudied. Especially, policy based the SLAs can provide Quality of Service to Grid application users and a flexible way.

In this paper, we proposed QoS constrained Quorum configuration scheme for reliable resource management and we also applied the traditional Quorum system to resource management scheme in Grid, since the Quorum system can improve availability and reliability of resources. Initially, we defined Quorum entities consisting of middleware resources and network resources. We also defined a QoS Quorum as a set of conditions for including QoS entities required by the user and a resource Quorum as the set of resource entities that can satisfy the conditions of the QoS Quorum. Secondly we proposed two kinds of Resource Quorum configuration scheme for optimizing the performance and the usage cost. In this scheme, we configured a Resource Quorum set to adequately allocate resources to the Grid application after sorting in order of the summation of both entities of middleware and network. Finally, we carried out simulations using Simgrid and experiments based on the ECG application for health care since this application needs transferring lots of data and analyzing complicated signal of ECG. Simulation results showed that network capabilities are more important than computing capabilities, as both sizes of transferred data and computation task increase. Experiment results showed that our scheme could reduce the total execution time of ECG application by using proposed heuristic algorithm, compared to a policy based management scheme. We expect that the proposed resource entities model for management can be applied for more reliable resources allocation in Grid.

References

1. F.Douglis, I.Foster, "The Grid Grows Up" Internet Computing IEEE 2003, pp. 24-26
2. I. Foster and C. Kesselman, "Globus : A Metacomputing Infrastructure Toolkit," International J. Supercomputing Applications, vol. 11, no. 2, 1997, pp. 115-128
3. R. Wolski. "Dynamically Forecasting Network Performance to Support Dynamic Scheduling Using the Network Weather Service" In Proceedings of the 6th High-Performance Distributed Computing Conference, August 1997.
4. K.Yang, A. Galis, C. Todd "A Policy-based Active Grid Management Architecture", Proceedings of 10th IEEE International Conference on Networks (ICOIN02), pp 243-248, IEEE Press. August 2002.
5. I.Liabotis, et al, "Self-organising management of Grid environments"
6. D. Malkhi and M. Reiter. Byzantine quorum systems. In Proceedings of the 29th ACM Symposium on Theory of Computing (STOC), May 1997.
7. A. Golberger, L. Amaral, L. Glass, J.M. Hausdorff et al, "PhysioBank, PhysioToolkit, and PhysioNet :Component of a New Research Resource for Complex Physiologic Signals," Circulation 101 (23), June,2000
8. H. Casanova, Simgrid: A Toolkit for the Simulation of Application Scheduling , Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001), May 15-18, 2001, Brisbane, Australia.