Resource Reconfiguration Scheme Based on Temporal Quorum Status Estimation in Computational Grids

Chan-Hyun Youn[1,2], Byungsang Kim[2], Dong Su Nam[2], Bong-Hwan Lee[3],
Eun Bo Shim[4], Gary Clifford[1], and Jennifer Healey[5]

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

[2] School of Engineering, Information and Communications University
58-4 Hwaam-dong, Yousong-gu, Daejeon 305-732, Korea
{dsnam,bskim}@icu.ac.kr

[3] Dept. of Information and Communications Engineering, Daejeon University
96-3 Yongun-dong, Dong-gu, Daejeon 300-716, Korea
blee@dju.ac.kr

[4] Dept. of Mechanical Engineering, Kwangwon National University
192-1 Hyoja 2-dong, Chunchon 200-701, Kwangwon-do, Korea
ebshim@kangwon.ac.kr

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

Abstract. Quality of Service (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 includes a system resource and network resource, both of which can satisfy the requirements of application QoS. We also propose the resource reconfiguration algorithm based on temporal execution time estimation method. Resource reconfiguration performs the reshuffling of the current available resource set for maintaining the quality level of the resources. We evaluate the effectiveness of resource reconfiguration mechanism in a Heart Hemodynamics analysis. Our approach increases the stability of execution environment as well as decreases the completion time compared to the method that does not adopt the proposed reconfiguration scheme.

1 Introduction

Recently, Grid computing has distinguished itself from the conventional distributed computing by focusing on large-scale resource sharing and innovative applications under the environment of widely-connected network. There are a number of architectures known for the management of Grid networks. However, most of them are mainly focused on concrete aspects which do not cover

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 699–707, 2004.

© Springer-Verlag Berlin Heidelberg 2004

the management of the Grid as a whole. Examples are the Condor-G system [3] and Nimrod-G Grid resource broker [4]. Due to the complexity of the Grid system, and the trend towards increased complexity in both hardware/software and service requirements, the flexible management of the overall Grid system itself is becoming more and more important. Policy-based management (PBM)[5] shows a prominent approach and management architecture over previous traditional network management systems. Policy-based Management can guide the behavior of a network or distributed system through high-level declarative directives that are dynamically introduced, checked for consistency, refined and evaluated, resulting typically in a series of low-level actions [6]. The current PBM architecture has problems in the sense that it can only address a fairly limited set of issues and usually requires human intervention. Grid systems unfortunately suffer the same problems in terms of the scalability and autonomic management. It is time consuming and error-prone for Grid administrator, resource manager or broker to configure their system manually. Furthermore it is extremely hard to configure their local resource while considering other domains in the whole Grid system. In this paper, we propose a resource Quorum model for supporting QoS of resource status. Quorum set is a subset of the resource universe which can be obtained. Quorum set guarantees the reliable resource allocation because it is a set of resource collection that is selected according to the QoS of the resource. Also, we propose resource reconfiguration scheme in order to maintain the integrity of the Quorum set. Resource reconfiguration provides reshuffle of the current resources set and reallocates resources to tasks in order to support QoS. Execution time of an application reflects the current resource status. We use temporal variation of the execution time in order to estimate the resource status. Resource reconfiguration provides a good solution for degraded resources in an unpredictable environment. The remainder of this paper is structured as follows. In Section 2, we suggest the resource Quorum model and resource allocation scheme for reliable scheduling. Section 3 presents the proposed temporal Quorum status estimation method and resource reconfiguration algorithm. Section 4 shows the experimental results based on our reconfiguration algorithm using Heart Hemodynamics analysis. This paper is concluded in Section 5.

2 Resource Quorum Model

To guarantee QoS for user's application, the Grid manager should assign the resources to each application. The Quorum based management system defines the QoS vector to each application from the input profile. We utilize two different resource items: system resource and network resources. Each service requester must specify his QoS requirements for the resource manager in terms of the minimum QoS properties for both the system and network resources. All resources are distinguished from the others and ranked as a group of QoS level in terms of those resource descriptions. We represent the resource descriptions as a system resource vector $\boldsymbol{\theta_s}$ and a network resource vector $\boldsymbol{\theta_n}$.

2.1 Quorum Model for QoS Support

We define the Resource Quorum as $\boldsymbol{Q_R}$, which represents the current status of the resource. Each resource has its resource status vector which is represented by both invariable and variable elements [8]. System resources can take the processor specification or memory size as invariable elements and processor utilization or available memory space as variable elements. Also, network resource would have the link capacity with an invariable elements and end-to-end available bandwidth, delay, data loss rate as variable elements, such that

$$\boldsymbol{Q_R} = \{\langle \boldsymbol{\theta}_i, \boldsymbol{\theta}_{jk} \rangle | i, j, k = 1, \ldots, n\}, \tag{1}$$

where $\boldsymbol{\theta_i}$ denotes the current available resource level of the system resource i and $\boldsymbol{\theta_{jk}}$ represents the current available resource level of the network between system resource j and k. A resource universe $R = \{r_1, \ldots, r_u\}$ assumes a collection of resources that can be taken in the administration domain. A resource, $r = \langle S, N \rangle$, can be represented as undirected graph in the system, S and their communication networks, N. Thus,

$$\boldsymbol{R} = \{r_i, \ldots, r_u\} = \{\langle S_i, N_{ij} \rangle | S_i = r_i, N_{ij} = r_i \times r_j \;\; i \neq j \;\; i, j = 1, \ldots, u\} \tag{2}$$

R is an available resource universe which is a subset of resource universe U, and i and j are elements in the R. S_i is therefore a system resource that represents a computation node i and N_{ij} is a network resource that represents a communication network from system i to system j. An application can be represented by undirected graph with tasks and their communication relation. The application is viewed as composed of the number of m tasks which represent problem size of the application $\boldsymbol{A}$. $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \{\nu^1, \ldots, \nu^m\} = \{\langle \nu^k, e^{kl} \rangle | e^{kl} = \nu^k \times \nu^l, k \neq i, k, l = 1, \ldots, m\}, \tag{3}$$

where m means the number of tasks. ν^k means the vertices that represent each task and e^{kl} means the edge that represents communication between ν^k and ν^l. Thus, l represents all communication peers that related to the ν^k.

If we assume that each task has its QoS requirement issued from SLAs, all resources can be ranked by the performance or be grouped by its attribute in terms of the QoS level. A required QoS level represents the vector of the resource description and has the range between the minimum and maximum requirement. We denote them by $\boldsymbol{q}$ and $\boldsymbol{Q}$, respectively. We define the QoS-Quorum,$\boldsymbol{Q_A}$, which represents the required quality level for the application set $A = \{1, \ldots, m\}$.

$$\boldsymbol{Q_A} = \left\{ \left\langle \left[\boldsymbol{q}_i^k, \boldsymbol{Q}_i^k\right] \left[\boldsymbol{q}_{ij}^{kl}, \boldsymbol{Q}_{ij}^{kl}\right] \right\rangle | i \neq j, i, j = 1, \ldots, \mu \;\; k \neq l, k, l = 1, \ldots, m \right\}, \tag{4}$$

where $\boldsymbol{q}_i^k$ and $\boldsymbol{Q}_i^k$ denote the minimum and maximum QoS level required for task k on the system resource i. $\boldsymbol{q}_{ij}^{kl}$ and $\boldsymbol{Q}_{ij}^{kl}$ represent the minimum and maximum QoS level required for communicating the task k in system resource i and task l in system resource j.

2.2 Available Resource Quorum and Resource Configuration

To achieve the reliability in resource management, we define an available resource Quorum,$\boldsymbol{Q}_{AR}$, which is selected from a resource universe. $\boldsymbol{Q}_{AR}$ satisfies the QoS requirement from SLAs.

$$\boldsymbol{Q}_{\boldsymbol{AR}} = \{\langle S_i, N_{ij}\rangle \subseteq R | \boldsymbol{q}_i^k \leq \boldsymbol{\theta}_i \leq \boldsymbol{Q}_i^k, \boldsymbol{q}_{ij}^{kl} \leq \boldsymbol{\theta}_{ij} \leq \boldsymbol{Q}_{ij}^{kl}\}, \tag{5}$$

where $i, j = 1, \ldots, n' \leq \mu$ and $k, l = 1, \ldots, m$, and $\boldsymbol{Q}_{\boldsymbol{AR}}$ is a set that satisfies a desired minimum QoS level of the application. Then, we define the Resource Configuration Function, $F(A, \boldsymbol{AR})$,as follows:

$$F(A, \boldsymbol{Q}_{\boldsymbol{AR}}) = \{\langle S_i^k, N_{ij}^{kl}\rangle\} = \{\langle V^k, E^{kl}\rangle \overset{Q}{\rightarrow} \langle S_i, N_{ij}\rangle\}, \tag{6}$$

where

$$S_i^k = \begin{cases} 1, \text{ if the } \nu^k \in A \ \text{ and allocated on } S_i \\ \phi, \text{ otherwise,} \end{cases}$$

$$N_{ij}^{kl} = \begin{cases} 1, \text{ if the } e^{kl} \in A \ \text{ and existed in communication BW of } r_i \ and \ r_j \\ \phi, \text{ otherwise} \end{cases}$$

Basically, Available Resource Quorum set $\boldsymbol{Q}_{\boldsymbol{AR}}$ has the characteristics to guarantee the minimal QoS requirement. First of all, the minimal QoS constraints created by the SLAs make up two groups of vectors in the QoS Quorum and the Resource Quorum. The QoS Quorum is made for the service class correspondent with one of the QoS services. Simultaneously, the Resource Quorum is determined depending on whether the QoS constraints are satisfied or not.

3 Temporal Quorum Reconfiguration

Resource configuration function is different from general scheduling in terms of QoS support. Resource configuration provides a more reliable scheduling because it assumes that the elements of the available resource Quorum set $\boldsymbol{Q}_{\boldsymbol{AR}}$ satisfy the user's desired quality level. But Quorum status varies as changing of the status of resources which are included in Quorum set. In order to maintain the integrity of QoS, it is necessary to reconfigure the current Quorum set. By monitoring resource status, we can validate the current Quorum status. Execution time of an application reflects the resource status. Now we present the reconfiguration scheme based on variation of execution time of the application.

3.1 Estimation of the Resource Status

Since the execution time of an application is defined as summation of the computation time on the system, $\hat{S_i^k}(\theta_i)$ and communication time, $\hat{N_{ij}^{kl}}(\theta_{ij})$, we can estimate the current resource utilization by each application activity. If an application represents its previous execution progress, we can predict the temporal variation of the resource utilization on target application. Equation 7 shows the sensitivity of the estimates of the execution time, $Z(\bullet)$, according to resource Quorum.

$$\hat{S}_i^k(\theta_i) = \frac{dS_i^k(\theta_i)}{dZ(S_i^k(\theta_i))} \triangle Z(S_i^k(\theta_i)), \quad \hat{N}_{ij}^{kl}(\theta_{ij}) = \frac{dN_{ij}^{kl}(\theta_{ij})}{dZ(N_{ij}^{kl}(\theta_{ij}))} \triangle Z(N_{ij}^{kl}(\theta_{ij})) \quad (7)$$

Based on resource status which is obtained by Equation 7, we define the utility functions, such as a system utility function and a network utility function. Utility functions of the resource provide the current system and network performance compared to the previous status. Previously, we denoted the minimum QoS of each task required for computation and communication as $\boldsymbol{q}_i^k(\theta_i)$ and network, $\boldsymbol{q}_{ij}^{kl}(\theta_{ij})$, respectively. If a current estimates of system resource is $\hat{S}_i^k(\theta_i)$, then the utility function is represented by

$$\mu_i^k = \frac{\hat{s}_i^k(\theta_i) - q_i^k(\theta_i)}{q_i^k(\theta_i)}. \quad (8)$$

If $\mu_i^k < 0$, we assume that the system does not meet the QoS level. Similarly, if a current estimates of network status is $\hat{N}_{ij}^{kl}(\theta_{ij})$, the utility function is represented by

$$\mu_{ij}^{kl} = \frac{\hat{s}_{ij}^{kl}(\theta_{ij}) - q_{ij}^{kl}(\theta_{ij})}{q_{ij}^{kl}(\theta_{ij})} \quad (9)$$

Furthermore if $\mu_{ij}^{kl} < 0$, we recognize that the network does not guarantee the QoS requirements. Status of the current resource configuration can be presented as the accumulated value of the utility function. Therefore, on the current time instant T_c, the temporal status of the resource configuration,$sRC(A, \boldsymbol{Q_{AR}}, T_c)$, is defined as follows.

$$sRC(A, \boldsymbol{Q_{AR}}, T_c) = \left\{ \langle \mu_i^k, \mu_{ij}^{kl} \rangle \right\}, \quad (10)$$

where

$$\mu_i^k = \begin{cases} \sum_{t=0}^{T_c} \mu_i^k(T_c), & \text{if the } e^{kl} \in A \text{ and existed in communicaion BW of } r_i \text{ and } r_j \\ \phi, & \text{otherwise} \end{cases}$$

$$\mu_{ij}^{kl} = \begin{cases} \sum_{t=0}^{T_c} \mu_{ij}^{kl}(T_c), & \text{if the } e^{kl} \in A \text{ and existed in communicaion BW of } r_i \text{ and } r_j \\ \phi, & \text{otherwise} \end{cases}$$

The utility functions, μ_i^k and μ_{ij}^{kl}, are accumulated QoS reward values that are system and network resources. If the estimate of utility function is smaller than 0, it means that the resource does not match the desired QoS level. We should then identify the triggering condition for the resource reconfiguration.

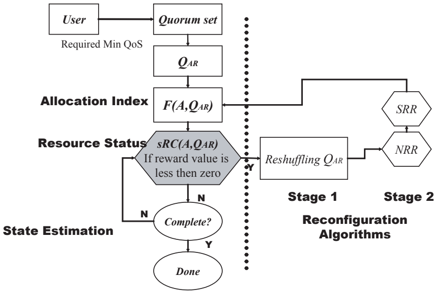


Fig. 1. Operation procedure of the proposed a resource Quorum reconfiguration scheme

3.2 Two-Stage Resource Reconfiguration Algorithm

The reconfiguration steps consist of System Resource Reconfiguration (SRR) when $\mu_i^k < 0$ and Network Resource Reconfiguration (NRR) when $\mu_{ij}^{kl} < 0$, respectively. Once triggered, the resource Quorum management system invokes the two-stage resource Quorum reconfiguration procedure on $sRC(A, \boldsymbol{Q_{AR}}, T_c)$ as shown in Figure 1.

In reconfiguration step, after generating the initial $\boldsymbol{Q_{AR}}$, it should be changed with the time. Before creating alternative configurations, we should obtain a new $\boldsymbol{Q_{AR}}$ by updating itself.

4 Experimental Results

We evaluate the effectiveness of resource reconfiguration mechanism with a Heart Hemodynamics analysis application that is parallel application linked with each task. The parallel applications are connected to each other by the Grids. End-to-end communication quality of service in case of linked chains is more important issue because it has an effect on the entire performance of the application execution. Moreover the communication status has various reasons that cause degradation. In order to point out the communication quality of service, we examine the end-to-end bandwidth between each task. Blood flow in the sac of the Korean Artificial Heart (KAH) is numerically simulated by finite element analysis. A distributed computing algorithm is employed to compute the hemodynamics of the sac using Globus-based MPI programming. Each sub-job of the KAH communicates with its neighbors in order to exchange the message for satisfying the boundary condition. Figure 2 shows the boundary condition of the KAH and the flow chart for simulating the blood sac in the KAH. The program has the iteration characteristics to solve the velocity and pressure at each time frame. The Grid testbed for collaborating among multi-domain environments is comprised of Linux-based Globus platforms for Grid application. The Grid testbed for modeling the KAH is run on three domains (Information and Communications University (ICU), MIT and Hanyang Univeristy (HYU))

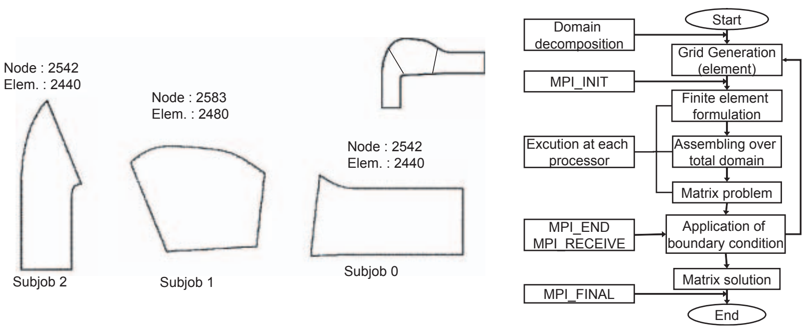
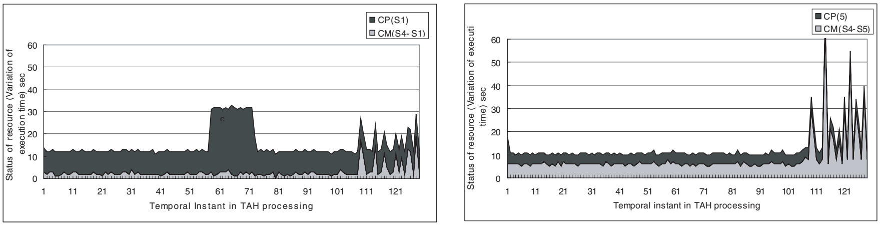


Fig. 2. Computing model for the KAH



(a) Execution time of system S1

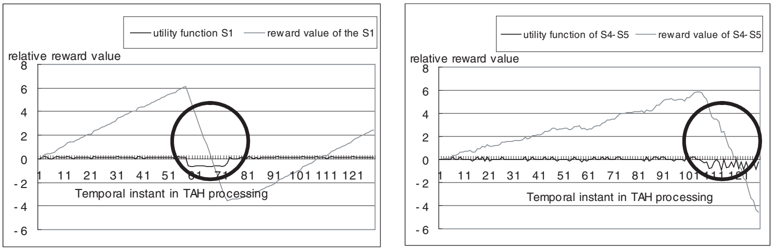
(b) Execution time of system S5

Fig. 3. Performance comparison of systems S1 and S5

and five systems (S1, S2, S3, S4, and S5 with available bandwidth of 0.1 Mbps). They are connected to the Korea Research and Education Network (KOREN) and Science Technology And Research Transit Access Point (STAR-TAP).

We conducted an experiment with offered load at system S1 at the time instant T2 (time instant 50) and network S4-S5 at the time instant T3 (time instant 100). T2 and T3 are the points of the time instant 50 and time instant 100 in Figure 3, respectively. Figure 3 shows the execution time of system S1 and S5. In Figure 3 (a), system S1 increases the computation time on the time instant 50 to 75. Also, system S5 increases the communication time irregularly and rapidly between system S4 and S5 between time instant 105 to 125 shown in Figure 3 (b). The performance was evaluated according to each computation time (CP) and communication time (CM).

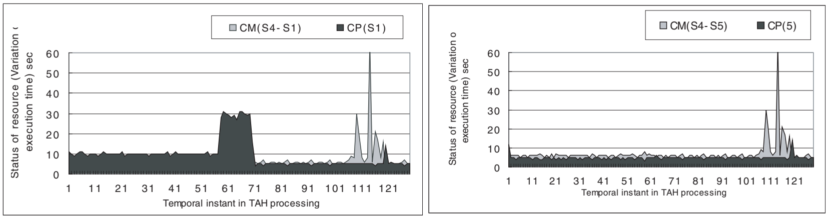
We calculate the reward value for the offered workload situation. Figure 4 shows the characteristics of the utility function and reward value of the system S1 and S5. The reward value of the system S1 has a negative value around at time instance 70 as depicted in Figure 4 (a). Also, the reward value of the systems S4-S5 has negative value at time-instance 115 as shown in Figure 4 (b). The zero point of the reward value of the resource is the reconfiguration triggering point. We take two reconfiguration points at time instance 70 and time instance 115 based on reward value from Figure 4. Let's consider the recon-



(a) Utility function
and relative reward value at system S1

(b) Utility function
and relative reward value at system S4-S5

Fig. 4. Characteristics comparison of the utility function and the reward values in target systems



(a) Execution time of system S1 after
committing the resource reconfiguration

(b) Execution time of system S5 after
committing the resource reconfiguration

Fig. 5. Performance comparison of the proposed reconfiguration scheme

figuration policies that are generated by our reconfiguration algorithm. Initial topology has the system S1, S4, S5. After the reconfiguration of system S1 at T2, the system S1 replaced with S3. Also, on the reconfiguration of network S4-S5, the system S4 replaced with S1. The network topology has been changed by the reconfiguration of the system. Figure 5 illustrates the execution time after committing the resource reconfiguration. Figure 5(a) shows the execution time of the system S1 when committing reconfiguration at T2, whereas Figure 5(b) shows the execution time of the system S5 when committing reconfiguration at T3. Figure 5 shows that proposed reconfiguration scheme increases the stability of the execution time compared to Figure 3.

Heart Hemodynamics Analysis Application shows that the effectiveness of system and network reconfiguration of the parallel application. From the experimental results, we can see that the proposed reconfiguration algorithm provides more stability during the execution time. In addition, the proposed algorithm decreases the completion time of the application.

5 Conclusions

Computational Grids are focused primarily on high-performance distributed computing. In a wide area Grid environment, it is most important to guarantee the user desired resources. Reliable resource allocation should be maintained on the temporal and spatial change. Quorum based resource modeling and resource configuration scheme provides more reliable resource scheduling. The proposed resource reconfiguration algorithm has two phases. One is the status estimation using temporal deviation. The other is resource reconfiguration based on estimated status. After triggering based on estimation, the reconfiguration algorithm tries to optimize the current system and network resource. Our approach provides both increase in the stability of the execution environment and decrease in the completion time compared to the methods that do not adopt the resource reconfiguration mechanism.

Acknowledgements

This work was supported in part by the Ministry of Information and Communication of Korea.

References

[1] Foster, I. and Kesselman, C. (eds.). *The Grid: Blueprint for a New Computing Infrastructure.* Morgan Kaufmann, 1999.

[2] Foster, I. and Kesselman, C. The Anatomy of the Grid:Enabling Scalable Virtual Organizations. *Intl J. Supercomputer Applications*, 2001.

[3] Frey, J., Foster, I., Livny, M., Tannenbaum, T. and Tuecke, S. Condor-G: A Computation Management Agent for Multi-Institutional Grids. University of Wisconsin Madison,2001. 700

[4] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger. Economic Models for Resource Management and Scheduling in Grid Computing.*Journal of Concurrency and Computation.*Wiley Press, 2002. 700

[5] K.Yang, A. Galis, C. Todd. A Policy-based Active Grid Management Architecture. *In Proceedings of the 10th IEEE International Conference on Networks (ICOIN02)*, pages 243-248, IEEE Press, 2002. 700

[6] P. Flegkas, P. Trimintzios, G. Pavlou, A. Liotta. Design and Implementation of a Policy-based Resource Management Architecture. *In Proceedings of the IEEE/IFIP Integrated Management Symposium (IM'2003)*, pages 215-229, Colorado Springs, USA, 2003. 700

[7] Franken, L. J. N. and Haverkort, B. R. The performability manager.*IEEE Network*, 1994.

[8] A.Leff, J. T.Rayfield, and D. M.Dias. Service-Level Agreements and Commercial Grids. pages 44-5, *IEEE Internet Computing*, 2003. 701

[9] I. Cardei, S. Varadarajan, M. Pavan, M. Cardei, and M. Min. Resource Management for Ad-hoc wireless networks with cluster organization. *Journal of Cluster Computing in the Internet*, 2004.

[10] Christos G, Cassandras. *Discrete Event Systems, Modeling and Performance Analysis*, IRWIN Press, 1993.