

An Optimal Resource Sharing in Hierarchical Virtual Organizations in the Grid

Kyong Hoon KIM^{†a)}, *Member*, Guy Martin TCHAMGOUE[†], Yong-Kee JUN[†],
and Wan Yeon LEE^{††}, *Nonmembers*

SUMMARY In large-scale collaborative computing, users and resource providers organize various Virtual Organizations (VOs) to share resources and services. A VO organizes other sub-VOs for the purpose of achieving the VO goal, which forms hierarchical VO environments. VO participants agree upon a certain policies, such as resource sharing amount or user accesses. In this letter, we provide an optimal resource sharing mechanism in hierarchical VO environments under resource sharing agreements. The proposed algorithm enhances resource utilization and reduces mean response time of each user.

key words: resource sharing, virtual organization, optimal allocation, grid computing

1. Introduction

As computing and network techniques have been developed, new computing paradigms have been emerged, such as Grids, P2P, Clouds, autonomic computing, and so on. One of the common properties of these new computing paradigms is to aiming at collaboration among participants. A Virtual Organization (VO) enables participants of computing environments to share their resources and achieve their common goals. A virtual organization is defined as a set of individuals and institutions forming an ad-hoc partnership to solve a common problem by sharing resources [1]. Recent research has focused on VO-based services, including VO formation, operation, and resource allocation. Thus, large-scale and next-generation computing research projects provide VO services and organize various VOs to utilize distributed resources efficiently [2], [3].

As the number of VOs increases in the Grid, efficient VO management is required. For example, Data Grids can be classified into four models in terms of organizations: monadic, hierarchical, federated, and hybrid [4]. Among various VO models, this work focuses on the hierarchical VO model in which a VO can organize its own sub-VOs for the purpose of achieving the VO goal. Many national large-scale systems and projects have been established based on a consortium following the hierarchical VO model.

In VO-enabled computing environments, the VO-wide resource allocation problem becomes an emerging research topic, which enables a user to access several resources

throughout VOs. Most research has been conducted on policy-based resource allocation and fair resource sharing in VOs [5]–[9]. Dumitrescu and Foster [5] propose a usage policy-based scheduling in VOs and evaluate both aggregate resource utilization and aggregate response time. Elmroth and Gardfjall [6] have presented a decentralized architecture for a Grid-wide fair scheduling system, where each local scheduler enforces Grid-wide hierarchical sharing policies using global resource usage data. In [9], they define the fair resource sharing problem in hierarchical VOs and propose a heuristic algorithm. However, in this work, we provide an optimal resource sharing method in hierarchical VO environments for the purpose of minimizing the average response time.

The remainder of this letter is organized as follows. In Sect. 2, we present the hierarchical VO system model and define the resource sharing problem. Section 3 provides the optimal resource sharing by solving the problem. We show numerical analysis in Sect. 4 and finally conclude the letter.

2. System Model and Problem Definition

2.1 Hierarchical Virtual Organizations

Components in large-scale and collaborative computer systems are users, resource providers, and VOs. A *user* is an end-entity who requests services to the computing systems. A *resource provider* assigns different shares of resources to users in VOs that he or she has joined in. A *VO* is an organization of users, resource providers, and sub-VOs to meet the goal of that organization. Furthermore, a VO can be organized hierarchically based on certain policies, such as roles, tasks, teams, regions, and so on, in order to achieve the common goal of the VO efficiently. Thus, a VO is defined as a set of users, resource providers, and sub-VOs, as in [2], [9].

Members in a VO are associated with a certain agreements, such as resource sharing policies, user roles, and so on. In this work, we consider VO policies between resource providers and their VOs in terms of resource sharing. The resource sharing policy of a resource provider r indicates the maximum amount of resource share to a VO v , which is denoted as $share(r, v)$. This sharing policy is a kind of SLA (Service-Level Agreement) established between a resource provider and a VO. For example, R3 in Fig. 1 provides 25% of resource to AC3, 25% to GRIDS Lab, and 25% to Kidney Model VOs.

Manuscript received January 4, 2012.

[†]The authors are with the Department of Informatics, Gyeongsang National University, Jinju, South Korea.

^{††}The author is with the Department of Computer Science, Dongduk Women's University, Seoul, South Korea.

a) E-mail: khkim@gnu.ac.kr

DOI: 10.1587/transinf.E95.D.2948

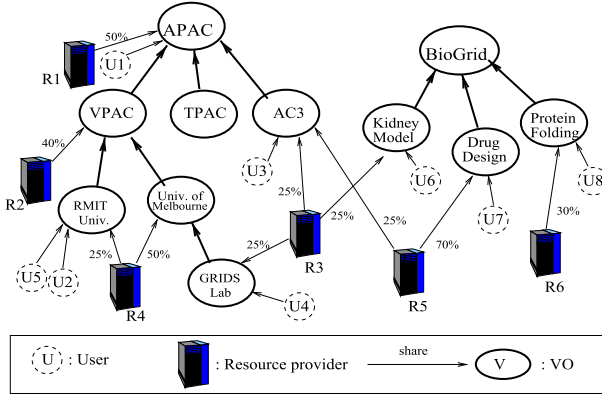


Fig. 1 An example of hierarchical VOs.

The resource share amount indicates the percentile of the total resource in a resource provider. It has different meaning according to the resource provider's sharing policy. For the space-shared scheduling policy, the share amount implies the number of processors provided to a VO. For the time-shared policy, it denotes the proportion in the total processing power of the resource provider assigned to a VO. In either case, users in a VO can access or use the resources under the resource sharing policies.

2.2 Problem Definition

In hierarchical VO environments, resource providers in a VO allow users in descendent VOs to use their resources as long as they do not violate the sharing policy. In other words, users can access resources in ancestor VOs as well as those in their own VOs. The resource broker should take this into consideration for resource allocation. For example, when the resource capacity in a higher-level VO is better than the lower-level one, users in lower-level are willing to use the higher-level VO resource due to its better performance. This can degrade the QoS provided to users in the higher-level VO and also lead to inefficient resource utilization. Thus, an efficient resource sharing mechanism is required in hierarchical VOs.

Jobs are generated by users and arrive at VOs. We assume that user i submits jobs according to a Poisson process with the arrival rate $\lambda^{(i)}$. Each resource provider j is modeled as an M/M/1 queueing system with the average processing rate $\mu^{(j)}$.

A VO's job arrival rate is derived from the participating users' job arrival rates by summing all the arrival rates. For a given set of VO users, U_v , a VO v 's job arrivals are modeled as a Poisson process with the arrival rate λ_v in Eq. (1).

$$\lambda_v = \sum_{u \in U_v} \lambda^{(u)} \quad (1)$$

Similarly, the resource processing rate of a VO v is defined by Eq. (2). Only the shared amount of a resource provider is available to a VO, so that $\mu^{(r)}$ is multiplied by $share(r, v)$.

$$\mu_v = \sum_{r \in R_v} share(r, v) \times \mu^{(r)} \quad (2)$$

The problem of efficient resource sharing considers how to allocate a VO's given processing rate to its descendent VOs for the purpose of minimizing the average response time. We denote the proportion of a VO i 's service processing for a descendent VO j as $p_{i,j}$. Then, the service processing rate of the descendent VO j is increased by $p_{i,j} \cdot \mu_i$. The actual service processing rate of a VO i is defined by Eq. (3).

$$\mu_i^a = \sum_{j \in \{i\} \cup ancs(i)} p_{j,i} \cdot \mu_j \quad (3)$$

where $ancs(i)$ is the set of ancestor VOs of VO i .

We decide the resource allocation $p_{i,j}$ from leaf nodes to the root node in a hierarchy. For a leaf VO v , $p_{v,v}$ should be 1.0 since there is no sub-VOs to share the resource. Now, let us assume that the actual service rate of each descendent VO j is known as μ_j^a . The problem is to decide $p_{i,i}$ and each $p_{i,j}$ where $j \in desc(i)$ to minimize the total response time ($\mu_i^a = 0$). Let us denote $desc(i)$ as the set of descendent VOs of VO i . Thus, the problem **SHARE_i** [9] is:

To minimize

$$\sum_{j \in \{i\} \cup desc(i)} \frac{1}{p_{i,j} \cdot \mu_i + \mu_j^a - \lambda_j} \quad (4a)$$

subject to

$$\sum_{j \in \{i\} \cup desc(i)} p_{i,j} = 1, \quad (4b)$$

$$p_{i,j} \geq 0. \quad (4c)$$

3. Optimal Resource Sharing

In order to provide an optimal solution to problem **SHARE_i**, we first ignore the inequality constraint given by Eq. (4c). We provide solution for the sub-problem given by Eqs. (4a) and (4b) and use the constraint in Eq. (4c) to refine the obtained solution.

An instance of the sub-problem of the non-linear problem **SHARE_i** denoted by **SUB-SHARE_i** is given by

$$\begin{aligned} & \text{Minimize} \quad \sum_{j \in \{i\} \cup desc(i)} \frac{1}{p_{i,j} \cdot \mu_i + \mu_j^a - \lambda_j} \\ & \text{Subject to} \quad \sum_{j \in \{i\} \cup desc(i)} p_{i,j} = 1. \end{aligned} \quad (5)$$

The application of the Lagrange multipliers [10] to the problem **SUB-SHARE_i** yields the following conditions:

$$-\frac{\mu_i}{(p_{i,j} \cdot \mu_i + \mu_j^a - \lambda_j)^2} - \alpha = 0, \quad j = 1, 2, \dots, n \quad (6)$$

where α is the common Lagrange multiplier and $n = |\{i\} \cup$

$desc(i)|$.

Equation (6) clearly shows that all VO j in the optimal solution to **SUB-SHARE** _{i} should have the same value of α and thus, contribute equally in minimizing the total response time. Therefore, for any pair of nodes $j, k, j \neq k$, we can write

$$-\frac{\mu_i}{(p_{i,j} \cdot \mu_i + \mu_j^a - \lambda_j)^2} = -\frac{\mu_i}{(p_{i,k} \cdot \mu_i + \mu_k^a - \lambda_k)^2}. \quad (7)$$

There are two possible solutions to Eq. (7) but only one is feasible since we assume all parameters to be positive. Given the relation between nodes j and k obtained from Eq. (7) and considering that the constraint in Eq. (5) should hold also, the optimal solution to problem **SUB-SHARE** _{i} is obtained by.

$$\begin{aligned} p_{i,k} &= \frac{1 - \sum_{j=1, j \neq k}^n C_j^k(i)}{n} \\ p_{i,j} &= C_j^k(i) + p_{i,k}, \quad \forall j, j \neq k \\ \text{with } C_j^k(i) &= \frac{\mu_k^a - \lambda_k - \mu_j^a + \lambda_j}{\mu_i} \end{aligned} \quad (8)$$

Now, we present the general solution to the problem **SHARE** _{i} . In addition to constraints given by Eqs. (4b) and (4c), the following necessary and sufficient Kuhn-Tucker conditions [10] are satisfied by any optimal solution to problem **SHARE** _{i} .

$$\frac{\mu_i}{(p_{i,j} \cdot \mu_i + \mu_j^a - \lambda_j)^2} + \alpha - \beta_j = 0, \quad j = 1, 2, \dots, n \quad (9a)$$

$$-\beta_j \cdot p_{i,j} = 0, \quad j = 1, 2, \dots, n \quad (9b)$$

$$\beta_j \geq 0, \quad j = 1, 2, \dots, n \quad (9c)$$

Note that $\alpha, \beta_1, \beta_2, \dots, \beta_n$ represent the Lagrange multipliers [10] for a given instance of the problem **SHARE** _{i} and $n = |i| \cup desc(i)|$. From the constraint (4c), we know that for a given VO i , the service processing of its descendent VO j denoted by $p_{i,j}$ can only be either zero or positive. Let us look separately at the two cases.

Case 1: A given descendent VO j receives no service processing i.e. $p_{i,j} = 0$. From Eqs. (9b) and (9c), we know that $\beta_i \geq 0$. Hence, from Eq. (9a), we obtain

$$-\frac{\mu_i}{(\mu_j^a - \lambda_j)^2} \leq \alpha, \quad j = 1, 2, \dots, n, \quad p_{i,j} = 0 \quad (10)$$

In this case, the value of α will be the lowest, which means that the response time will be the longest. So, the lower the value of α , the longer the response time. Therefore, this case cannot be a solution.

Case 2: VO j is provided with a positive service processing time i.e. $p_{i,j} > 0$. From Eqs. (9b) and (9c), we know that $\beta_i = 0$. Hence, from Eq. (9a), we obtain

$$-\frac{\mu_i}{(p_{i,j} \cdot \mu_i + \mu_j^a - \lambda_j)^2} = \alpha, \quad j = 1, 2, \dots, n, \quad p_{i,j} > 0 \quad (11)$$

Algorithm Optimal_Resource_Share(i)

Input: $\mu_i, \mu_j^a, \lambda_j$

Output: $FS = (p_{i,1}^*, p_{i,2}^*, \dots, p_{i,n}^*)$

1. Initialization

Order nodes into the list L in a decreasing value of $-\frac{\mu_i}{(\mu_j^a - \lambda_j)^2}$

2. Optimization

While ($L \neq \emptyset$)

 Compute the solution S_{int} to the corresponding problem **SUB-SHARE** _{i} as presented in Eq. (8)

 If all the inequality constraints in Eq. (4c) are satisfied

 Return $FS = S_{int}$

 Else

 Remove from L the node j that violates the constraint

 Set $p_{i,j}^* = 0$

 End if

End While

Return $FS = (p_{i,1}^*, p_{i,2}^*, \dots, p_{i,n}^*)$

Fig. 2 Algorithm for optimal resource sharing.

In this case, the solution to problem **SHARE** _{i} is exactly the same with the one of **SUB-SHARE** _{i} .

Equation (10) shows that a VO j receiving no service processing time may produce lower values of α and higher response time. Our algorithm to solve **SHARE** _{i} presented in Fig. 2 is based on analysis presented above. Thus, an optimal solution $FS = (p_{i,1}^*, p_{i,2}^*, \dots, p_{i,n}^*)$ to **SHARE** _{i} can be obtained in time $O(n^2)$, where n is the number VOs in VO i .

4. Numerical Analysis

We evaluate the proposed resource allocation scheme numerically for the example VO environment of Fig. 3. All resources are assumed to provide the same processing capacity of 20 PEs with 1,000 MIPS each. Each VO user continuously generates and submits jobs to the VO at the rate of 0.005 ($= \lambda_i$). The average job length is given by 1,500,000 MIs.

We compare the proposed scheme with two other resource allocation schemes, **Random** and **Dedicated**. In **Random** scheme, a user selects a resource provider randomly among available resources to be accessed. **Dedicated** scheme limits a user's access to his or her VO resource only. Table 1 shows the results of mean response times of VO users and the average values.

As shown in Table 1, the average response time of the proposed scheme shows the lowest because it distributes the resource in upper layers to lower-layer users optimally. The response times of users except VO3 are the same, which implies that the resource sharing is fair and efficient among all

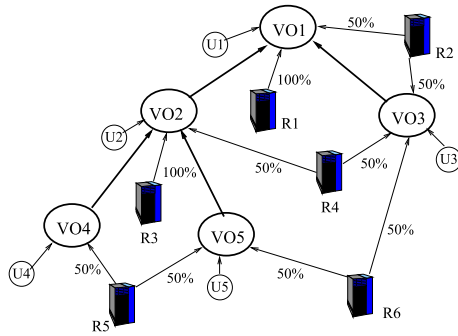


Fig. 3 Evaluated VO environments.

Table 1 Response time analysis (sec.).

	User1	User2	User3	User4	User5	Average
Dedicated	66.7	66.7	66.7	600.0	120.0	184.0
Random	150.0	110.3	103.6	140.2	102.1	121.2
Proposed	100.0	100.0	66.7	100.0	100.0	93.3

related users. The reason of low response time of VO3 user is because the resource capacity available to VO3 is high enough to be used for VO3 user. **Dedicated** scheme shows much difference among VO users according to the capacity of resources of each VO. Thus, the proposed scheme is required to utilize resources efficiently among VOs under the sharing policies.

5. Conclusions

In this letter, we defined a resource allocation sharing problem in hierarchical VOs and derived an optimal sharing under VO resource agreements. When the optimal sharing amount is determined, resource providers give higher priorities to users who request resources under the pre-determined sharing proportion. The time complexity of the proposed algorithm is given by $O(n^2)$, where n is the number VOs in a VO. The proposed algorithm enhances resource utilization and reduces mean response time of each user.

Acknowledgements

This research was supported in part by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2012-H0301-12-3003) supervised by the NIPA (National IT Industry Promotion Agency), and the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea (NRF) through the Human Resource Training Project for Regional Innovation.

References

- [1] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *International Journal of Supercomputer Applications*, vol.15, no.3, pp.200–222, 2001.
- [2] Open Science Grid project. <http://www.opensciencegrid.org>
- [3] The Enabling Grids for E-science project. <http://www.eu-egee.org>
- [4] S. Venugopal, R. Buyya, and K. Ramamohanarao, "A taxonomy of data grids for distributed data sharing, management and processing," *ACM Computing Surveys*, vol.38, no.1, pp.1–53, 2007.
- [5] C. Dumitrescu and I. Foster, "Usage policy-based CPU sharing in virtual organizations," *Proc. 5th IEEE/ACM International Workshop on Grid Computing (GRID 2004)*, pp.53–60, Pittsburgh, USA, Nov. 2004.
- [6] E. Elmroth and P. Gardfjall, "Design and evaluation of a decentralized system for Grid-wide fairshare scheduling," *Proc. 1st IEEE Conference on e-Science and Grid Computing*, Melbourne, Australia, Dec. 2005.
- [7] T.J. Norman, A. Preece, S. Chalmers, N.R. Jennings, M. Luck, V.D. Dang, T.D. Nguyen, V. Deora, J. Shao, W.A. Gray, and N.J. Fiddian, "Agent-based formation of virtual organizations," *Knowledge-Based Systems*, vol.17, pp.103–111, 2004.
- [8] K.H. Kim and R. Buyya, "Policy-based resource allocation in hierarchical virtual organizations for global grids," *Proc. 18th International Symposium on Computer Architecture and High Performance Computing*, Ouro Preto, Brazil, Oct. 2006.
- [9] K.H. Kim and R. Buyya, "Fair resource sharing in hierarchical virtual organizations for global grids," *Proc. 8th IEEE/ACM International Conference on Grid Computing (Grid 2007)*, Austin, USA, Sept. 2007.
- [10] D.G. Luenberger, Y. Ye, 2008. *Linear and Nonlinear Programming*, 3rd Edition. Springer, International Series in Operations Research and Management Science.