Robust Resource Allocation for Large-scale Distributed Shared Resource Environments

Yang-Suk Kee, Ken Yocum, Andrew A. Chien
Computer Science & Engineering, University of California San Diego
yskee@csag.ucsd.edu, kyocum@cs.ucsd.edu, achien@ucsd.edu
Henri Casanova
Information and Computer Sciences Department, University of Hawaii Manoa
henric@hawaii.edu

1. Introduction

Conceptually, resource allocation can be divided into resource selection (discovery) and binding (acquiring) processes. The architecture of traditional resource brokering systems separates resource selection from resource binding. In fact, these systems mostly focus on resource selection and provide sophisticated resource specification languages and resource selection algorithms. Typically, given a resource collection specification, a resource selection algorithm first identifies a matching collection based on available resource information and then attempts to acquire these collections by negotiating with local resource managers.

However, this separate approach cannot deal with binding failures efficiently. In the real world, resource binding can fail because of inaccurate resource information, authentication failure, and perhaps most importantly competition between applications. The key issue here is that these binding failures cannot be resolved until the system attempts to bind the whole resource collection after all resources have been selected. To address this issue, some systems consider multiple candidate resource collections. However, this approach still does not work well in shared resource environments because binding will succeed only when all the resources in a collection are acquired.

This paper presents a new formulation of the resource selection and binding problem and proposes a new algorithm called **integrated selection and binding** to solve this problem. Our insight is that a resource selection algorithm should consider binding failures. Consequently, the key idea of the integrated selection and binding approach is to decompose a resource collection request into components that can be bound and composed independently and to select multiple sets of resources for each component. The integrated approach is more efficient and effective than the separate approach for competitive access to federated resources.

2. Resource Allocation Approaches for Large-scale Federated Resources

2.1. Resource Description Language

We have designed a resource description language, vgDL [1]. It builds upon several existing constraint-based languages that provide constructs for expressing resource constraints on static and dynamic resource attributes and for expressing constraints on pair-wise inter-resource attributes. The key advance of vgDL for the purpose of selection and binding is the capability for applications to specify hierarchical resource aggregates and qualitative notions of network proximity between these aggregates. These enable hierarchical composition and allow us to decompose the request into a set of components.

vgDL contains three resource aggregates, distinguished based on homogeneity and network connectivity:

- **LooseBag** - a collection of heterogeneous nodes with possibly poor connectivity
- **TightBag** - a collection of heterogeneous nodes with good connectivity
- **Cluster** - a set of well-connected nodes with near identical individual resource attributes.

In addition, vgDL provides four operators that define network connectivity between aggregates: **close**, **far**, **highBW**, and **lowBW**. These composers indicate coarse notions of network proximity in terms of latency and bandwidth. We define “good” connectivity in terms of thresholds either fixed or set by examining the current dynamic inter-node attributes. This means the vgDL expression may be written oblivious to current network conditions or technologies.

2.2. Separate Selection and Binding with N-solutions

In systems such as Condor [2], RedLine [3], SWORD [4], application users can use a structured constraint language to specify complex resource requirements on single resources and the relationship amongst them. These systems all construct a set of resources and return it to the application. Then, the application is responsible for binding and accessing the resources of interest in the solution. Of course, if the resources are shared, such binding attempts may fail. SWORD averts the need to make repeated selection requests by returning multiple independent candidate solutions.

Typical selection algorithms for structured resource sets build solutions with bottom-up approach. For each component of a request, the algorithm collects a set of partial solutions. To compose a global solution, the algorithm must then select partial solutions for each component of the request and put them together. One or more of these global solutions are then returned to the application for binding.

2.3. Integrated Selection and Binding with N-solutions

To solve the problem of selecting and obtaining good resources in shared federated resource environments, we propose the idea of integrated selection and binding. In short, the idea is first to exploit the design of vgDL to make the compositional structure of resource requests explicit and malleable to identify independent resource allocation components. Second, select resources for each component in a vgDL request based on this structure and then bind resources for the components until sufficient resources to compose a global solution have been collected.

One critical aspect of this approach is the ability to decompose requests flexibly – an intentional design element of vgDL where loosely-coupled resources form natural bases of decomposition. Successful decomposition enables pre-selection of composable solutions, which of course make it possible to compose the bound resources into a good global solution. The ability of component-based decomposition, selection, and binding enables N selections for each component to correspond to $N^{components}$ global solutions. Without composition flexibility, N selections for each component would correspond to only N global solutions. Our hypothesis is that this flexibility will prove critical to succeed in meeting large, complex resource requests in highly-contested shared resource environments.

3. Conclusion

Understanding interplay between resource description, algorithm, and data representation is critical to efficient and effective resource allocation in shared distributed resource environments. We have realized this idea in the framework of the Virtual Grid and its execution system (vgES) [1]. Through an efficient implementation using relational database technologies, we can provide a robust resource allocation services to the applications. The integrated approach can provide more efficient and effective than the traditional separate selection and binding approach. The composable feature of the integrated approach fully exploits the extra solutions with small overhead.

4. Acknowledgements

The authors and research described here are supported in part by the National Science Foundation under NSF Cooperative Agreement NSF CCR-0331645, NSF NGS-0305390, and NSF Research Infrastructure Grant EIA-0303622. Support from Hewlett-Packard, BigBangwidth, Microsoft, and Intel is also gratefully acknowledged. Support from the Ministry of Information and Communication, Republic of Korea is also gratefully acknowledged.

References

[1] Y.-S. Kee, D. Logothetis, R. Huang, H. Casanova, A. A. Chien, Efficient Resource Description and High Quality Selection for Virtual Grids, ACM/IEEE International Symposium on Cluster Computing and the Grid, May 2005.

[2] R. Raman, M. Livny, and M. Solomon, Matchmaking: Distributed Resource Management for High Throughput Computing, IEEE Symposium on High Performance Distributed Computing, pp. 140-147, Jul. 1998.

[3] C. Liu and I. Foster, A Constraint Language Approach to Matchmaking, IEEE International Workshop on Research Issues on Data Engineering: Web Services for E-Commerce and E-Government Applications, pp. 7-14, Mar. 2004

[4] D. Oppenheimer, J. Albrecht, D. Patterson, and A. Vahdat. Design and Implementation Tradeoffs for Wide-Area Resource Discovery, IEEE Symposium on High Performance Distributed Computing, pp. 113-124, July 2005.