

Towards Adaptive, Scalable, and Reliable Resource Provisioning for WSRF-compliant Applications

Eun-Kyu Byun, Jae-Wan Jang and Jin-Soo Kim
Division of Computer Science

Korea Advanced Institute of Science and Technology (KAIST)
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, South Korea

ekbyun@camars.kaist.ac.kr, jwjang@camars.kaist.ac.kr, jinsoo@cs.kaist.ac.kr

ABSTRACT

Although WSRF (Web Services Resource Framework) and Java-based hosting environment have been successful in dealing with the heterogeneity of resources and the diversity of applications, the current Grid middleware has several limitations to support on-demand resource provisioning effectively.

This paper proposes DynaGrid, a new framework for WSRF-compliant applications. Many new components, such as ServiceDoor, Dynamic Service Launcher (DSL), Client Proxy, and PartitionManager, have been introduced to offer adaptive, scalable, and reliable resource provisioning. All of these components are implemented as standard WSRF-compliant Web services, hence DynaGrid is complementary to the existing Grid middleware. The experimental evaluations are performed on a real testbed with practical applications including the MapReduce application. The results indicate that DynaGrid effectively utilizes Grid resources by allocating only the required number of resources adaptively according to the amount of incoming requests, providing both the scalability and the reliability at the same time.

Categories and Subject Descriptors

C.2.4 [Computer-Communication Networks]: Distributed Systems

General Terms

Management, Design, Reliability

Keywords

Grid computing, WSRF, Globus Toolkit, Fault tolerant, Scalability

1. INTRODUCTION

Grid computing is the technology for building Internet-wide computing environment integrating distributed and heterogeneous resources [6]. The recently proposed OGSA (Open Grid Services Architecture) [5] and WSRF (Web Services Resource Framework) [2] are important steps towards Service Oriented Architecture (SOA), which provide the base architecture and interfaces for Grid.

WSRF is a group of specifications which define a generic and open framework for modeling and accessing stateful resources (will be denoted as *ServiceResource* in this paper) using Web services. Globus Toolkit version 4 (GT4) [4] is a representative WSRF-based Grid middleware which enables to execute or access WSRF-compliant Web services in a standard way.

Grid system should meet the following requirements for effective on-demand resource provisioning. The first is *adaptive resource provisioning*. In order to handle dynamically changing resource demand of each application, new resources should be allocated adaptively to the application that requires more computing power or storage. The second is the *scalability*. dynamic resource allocation usually necessitate a service-specific centralized manager which maintains the locations of currently allocated resources. Such a centralized manager easily becomes a performance bottleneck. Final requirement is the *reliability*. In the large-scale Grid system, the availability of resources also tends to vary dynamically as each resource may leave the system or crash unpredictably at any time. Unfortunately, existing WSRF framework including GT4 does not support such requirements.

This paper presents DynaGrid, a new framework which offers adaptive, scalable, and reliable resource provisioning for WSRF-compliant applications. Components of DynaGrid are implemented as standard Web service so that DynaGrid complements the limitation of GT4.

2. ARCHITECTURE OF DYNAGRID

DynaGrid is composed of four components: *ServiceDoor*, *Dynamic Service Launcher (DSL)*, *PartitionManager*, and *Client Proxy* as in Figure 1.

In DynaGrid, any Web service should deploy the corresponding ServiceDoor on a trusted resource. Only through the ServiceDoor, clients can access the Web services and ServiceResources. ServiceDoor hides complex resource management mechanisms from clients. Dynamic Service Launcher (DSL) is a Web service running on every resource in DynaGrid. DSL provides a new dynamic service deployment mecha-

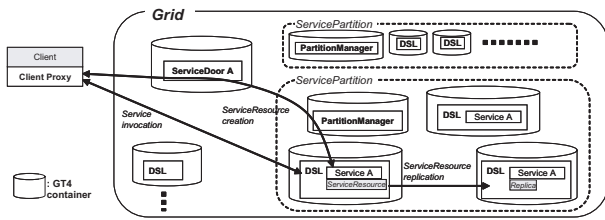


Figure 1: The overall architecture of DynaGrid

nism. The mechanism is generally applicable on any WSRF-based hosting environment, different to the mechanism used in the current GT4 [7]. DSLs also carry out ServiceResource replication and request logging. DynaGrid groups all the DSLs into several ServicePartitions managed by a PartitionManager. PartitionManager performs creation, replication, load balancing, recovery of ServiceResources and monitoring the status of DSLs and ServiceResources. Client Proxy transparently redirects the incoming service execution request to the actual location of ServiceResource.

To enhance the scalability, our adaptive resource provisioning mechanism does not rely on any centralized manager. DynaGrid achieves this by providing Client Proxy on the client side, a modified client stub which transparently redirects the client's request to the actual location of the corresponding ServiceResource. DynaGrid also distributes the cost of management and recovery by partitioning resources allocated for the service.

For reliability, every ServiceResource in DynaGrid is replicated to another resource and all the execution requests for the ServiceResource are logged within the replica. In case the original resource crashes, DynaGrid recovers the ServiceResource by replaying logged requests on the replica. DynaGrid can also recover DSL and PartitionManager failure since all information about ServicePartitions, allocated DSLs and ServiceResources are duplicated over ServiceDoor, PartitionManagers and DSLs.

3. EVALUATION

Figure 2 illustrates the throughput achieved with the simple Add service in several experimental settings. The result presents that, without direct Web service invocation, DynaGrid can not use the whole resources effectively since ServiceDoor becomes a bottleneck. The result also shows that the throughput with Client Proxy is scalable according to the number of DSLs hosting the service.

According to our experiment, the recovery of a ServiceResource takes at most 30 seconds. Assuming that the survival time of resource is 60 minute, the probability for loss of ServiceResource is estimated to only 0.8%.

Table 1: The execution time (seconds) of Nutch with GridMR on DynaGrid

Environment	Input 1	Input 2	Input 3
Original (Single server)	591	1435	2862
GridMR (5 DSLs)	630	1386	2497
GridMR (11 DSLs)	383	997	1832
GridMR (11 DSLs, 1 failure)	407	1059	1906

We practically execute Apache Nutch [1] on out DynaGrid

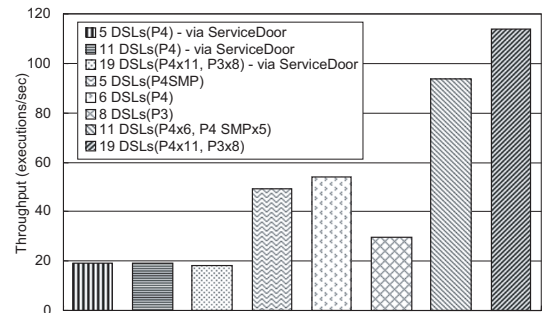


Figure 2: The effect of Client Proxy for the scalability through the comparison of aggregated throughput

testbed. We also develop GridMR, a WSRF-based Web service which implements the MapReduce framework [3] on DynaGrid and execute Nutch on it. Table 1 exhibits that the performance of Nutch improves as the number of DSLs increases and that GridMR successfully finishes the data processing only with the ignorable performance degradation even though a failure occurs in one of DSLs.

4. CONCLUSIONS

In this paper, we propose DynaGrid, a new framework which offers adaptive, scalable, and reliable resource provisioning for WSRF-compliant applications. We implement new dynamic service deployment mechanism and designed new mechanism for more scalable and reliable Grid platform. DynaGrid is complementary to the existing Grid middleware such as GT4, and can be used with any Java-based WSRF-compliant hosting environments.

5. REFERENCES

- [1] Apache Nutch Project. <http://nutch.apache.org/nutch>.
- [2] OASIS Web Services Resource Framework (WSRF) TC. <http://www.oasis-open.org/committees/wsrfl/>.
- [3] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI)*, pages 137–150, 2004.
- [4] I. Foster. Globus Toolkit Version 4: Software for Service-Oriented Systems. *Lecture Notes in Computer Science*, 3779:2–13, 2005.
- [5] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, 2002. <http://www.globus.org/research/papers/ogsa.pdf>.
- [6] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Lecture Notes in Computer Science*, 2150, 2001.
- [7] L. Qi, H. Jin, I. Foster, and J. Gawor. HAND: Highly Available Dynamic Deployment Infrastructure for Globus Toolkit 4. <http://www.globus.org/alliance/publications/papers/HAND-Submitted.pdf>, 2006.