A Service-oriented Resource Monitoring Architecture in a Distributed Environment

Ping Zhang
Nanjing Research Institute of Electronics Engineering
Nanjing, China
lyneausten@yahoo.com.cn

Hui Xu
Nanjing Research Institute of Electronics Engineering
Nanjing, China
mini_xh@hotmail.com

***Abstract*—This paper analyzes two common resource monitoring architectures, GMA and MDS, in distributed environment. Based on the strengths and weaknesses of both two architectures, a service-oriented resource monitoring architecture (RMA) was proposed, which can be seen as a GMA mode monitoring system constructed on the basis of MDS. The key technologies of the architecture were also been analyzed and implemented. RMA affords uniform, standards-based, scalable, service-oriented interface to provide interaction that is compatible and independent of operating platform. So the system has obvious advantages in distributed environment.**

Keywords- service-oriented; resource monitoring; service combination; GMA; MDS

I. INTRODUCTION

With the development of distributed computing technology, internet and the World Wide Web (WWW), local computing platform or infrastructure has been difficult to meet people's needs. People want to be able to use more widely distributed resources through effective equipment and tools in order to meet a wider range of application needs, without having to pay too much attention to their physical location.

The ensuing question is how to provide a scalable resource monitoring mechanism to master the status of these broad, heterogeneous, dynamic resources in real time, furthermore, to achieve accurate access to these state information, as well as orderly management and efficient retrieval. This will be the foundation of resource access, scheduling, control, fault detection and localization, as well as the premise of efficient use of the wide-area distributed resources. Therefore, resource monitoring is one of the key technologies to support resource sharing collaboration, reorganization and invulnerability.

For the above problem domain, this paper analyzes two resource monitoring architectures in a distributed environment. Based on the strengths and weaknesses of both two architectures, a service-oriented resource monitoring architecture was proposed. The key technologies of the architecture were also been analyzed and implemented.

II. DISTRIBUTED RESOURCE MONITORING ARCHITECTURE

In a distributed environment, there are two common monitoring architectures: GMA (Grid Monitoring Architecture) and MDS (Monitoring and Discovery Service). The former one is a theoretical model, the latter one is an important information service monitoring components of Globus toolset.

GMA is the standardization of grid monitoring system architecture which is promoted and committed by the Global Grid Forum. GMA's aim is to establish effective monitoring of distributed components, for example, allows for error detection and performance prediction. GMA processes performance data which is delivered as event with a time stamp. It consists of three types of components, namely, directory services, producer and consumer. Producers and consumers publish themselves to a centralized directory service, consumers can use a directory service to find interested producers, and vice versa. GMA briefly depicted three data transfer interactions between producers and consumers, which are publish / subscribe, query / response and notification as in [1] and [2].

GMA is just a specification, which does not require specific implementation details, neither defines a query language, data model or network protocol. There is no consideration of multi-hop peer-to-peer network, neither the synchronous message exchange nor routing. In addition, there is no loop detection, scope determining, overtime and temporary node selection.

MDS is an information service component of the Globus Toolkit, which is mainly used to detect, characterize and monitor the services and computing resources in grid, including static information and dynamic information of the resources. MDS has developed to MDS4, which is achieved based on WSRF (Web Service Resource Framework) specification. MDS4 is defined as a set of web services, also known as WS-MDS as in [3]. It provides standard web service interface to different detection tools and information sources, as well as a standard message formats and protocols for information access and transfer. It provided two senior services: Index Service and Trigger Service. Index service gathers data from different information sources, and provides data query and subscribe interfaces; Trigger Service collects data from grid resources, and performs various operations when certain conditions satisfied.

Globus resources are mainly oriented supercomputers, workstations, clusters, but it is not suited to widely-spread resources in a broader grid computing environment on the Internet as in [4] and [5]. It is relatively poor of flexibility in resource discovery, monitoring, access permissions management and information query. Because of the use of

Published by Atlantis Press, Paris, France.

end-to-end architecture, interfere with each other in end-to-end active testing makes a decrease in performance.

III. A SERVICE-ORIENTED RESOURCE MONITORING ARCHITECTURE BUILT IN JAVA WEB SERVICES

Reference [6] indicates that the software industry gone through a spiral curve, from the initial process-oriented to object-oriented, from component oriented to later integration-oriented and now service-oriented. So-called "service" is only related to the business, it is a kind of business interface independent of technology. The so-called "service-oriented", that is, how to implement the service interface independent of the technology as in [7].

Though MDS and GMA have different expectations and design in many aspects in information discovery and monitoring data transmission, the service-oriented resource monitoring architecture (RMA) in this section can be seen as their binding, i.e. a GMA mode monitoring system constructed on the basis of MDS.

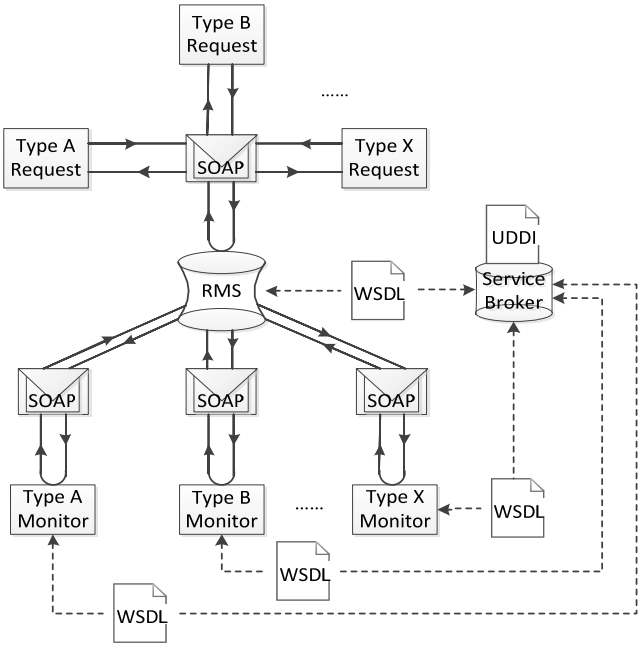

Figure 1. Service-oriented resource monitoring architecture.

Web service is service based on XML and HTTP as in [8], it is built on SOAP, WSDL and UDDI, which have become widely accepted as industry standard, RMA in this paper also adopted web services technology to build the resource monitoring framework. In this architecture, all the RMA components, including the RMS (Resource Monitoring Service) and various different types of MA (monitoring agent) are designed to be a web service. As shown above.

In the resource monitoring framework above, the various types of monitor is web services provider, using WSDL to describe a web service, and submitted service description information to the Service Broker.

Service broker use UDDI (Universal Description and Discovery Interface) specification to help web service provider publishing web service information online, as well as the service discovery for web service users.

RMS as web services user, gets Web services description from service broker. RMS packages method name, parameter values follow the service description information using SOAP protocol, and passes SOAP message to the service providers. Service provider handles the service request and completes a service call by returning the service response using SOAP format.

Been different with GMA, RMA has been designed as a hierarchical architecture of monitoring and information management, which is borrowed from MDS. RMS is not only a service consumer, but also a service provider. RMS could response different types of service requests, which could be from either resources monitoring service instance like RMS itself, or services outside RMA framework, such as resource scheduling service, fault detection and location services, etc. Applications can query and get the resource status information collected to build different tools and applications by taking advantage of the unified Web service inquiries, subscription, and notification interface, such as command line, browser and web service call.

IV. KEY TECHNOLOGIES

A. *Monitor Agent*

Distributed computing environment based on the Internet is a heterogeneous environment, which may be different from hardware to software, from operating system to programming language [9]. In order to guarantee the effectiveness of monitoring, all components in the application side must be monitored, including application software, hardware and network, etc.

Specifically, the paper argues that resources should be monitored in accordance with its hierarchy, as the following classification: hardware layer resources, operating system layer resources, software layer resources, service layer resources and the logical layer resources, as shown in the figure below.

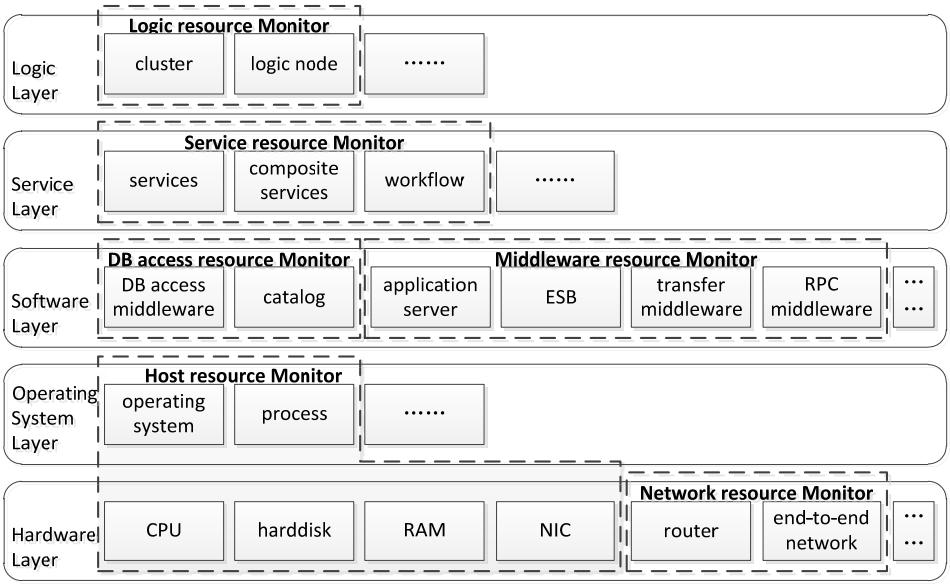

Figure 2. Monitor agent.

The hardware layer refers to the state of hardware resource and environment, such as the state of CPU, memory status, disk status, network card status, router state and end-to-end network status and so on.

The operating system layer is the state of the operating system; meanwhile, the processes should be included in the operating system state.

The software layer refers to the application or program that can run independently, especially middleware, such as database access middleware, messaging middleware, remote procedure call middleware, ESB service bus, directory services, application servers, and so on.

The service layer refers to the resources encapsulated into Web services, contains separated Web services and Web service combinations. Although workflow does not involve any specific service implementations, it describes the relationship of collaboration and coordination among multiple service interactions, it is also regarded as a service layer resources.

Logical layer refers to resource of work domain or manage domain especially in order to facilitate the understanding or management, such as clustering, logical nodes, and so on.

After completion of the division of resources, the next question is how to collect these heterogeneous resource statuses. As shown in the dashed box in the figure above, one single monitor agent can collect and manage one or more different types of resources for facilitate of implementation and management. Host Resource Monitor is the representative one, which can monitor CPU status, memory status, disk status, NIC state, operating system state and process state.

The other Monitor agent will not referred here due to limited space, however, convenient for later reads, three types of monitoring data are pointed out following in particular.

Services Resource Monitor:

TABLE I. SERVICE STATUS

Status Name	State content	Remark
serviceId	Service ID	
serviceName	Service Name	
state	Service Status	Run, stop, error,
stateMsg	Status Description	
baseTime	Reference time	Service start time, or start after artificially reset time
lastCallTime	Last call time	
recentNumber	Number of calls recently	Based on the most recent number of samples
recentTime	Sum of response recently	Based on the most recent number of samples
recentAvgTime	Average response recently	Based on the most recent number of samples
recentTps	TPS recently	Based on the most recent number of samples
recentSuccessRate	Successful rate recently (percentage)	Based on the most recent number of samples
totalNumber	Total number of calls	Since the reference based-time
totalFinishNumber	Total number of succeeded calls	Since the reference based-time
totalErrorNumber	Total number of failed calls	Since the reference based-time
totalTime	Sum of response time	Based on the total number
totalAvgTime	Average response time	Based on the total number
totalSuccessRate	Successful rate called	Based on the total number
totalTps	Average TPS	Based on the total number