SORBA : SHARED OBJECT REQUEST BROKER ARCHITECTURE FOR PEER COMPUTING

UNGU KANG*
HEE JOUNG HWANG*
SEI HOON LEE**
CHANGJONG WANG***
**Dept. of Newmedia, Gachon Gil College.*
***Dept. of ComputerEngineering, Inha Techinical College.*
****Dept. of Computer Science & Engineering,*
Inha University, Inchon, Republic of Korea
ugkang@gcgc.ac.kr, hwanghj@gcgc.ac.kr, seihoonlee@true.inhatc.ac.kr, cjwangse@dragon.inha.ac.kr

Abstract : In this paper, we proposed a SORBA (Shared Object Request Broker Architecture) model, an object-based P2P model, which complies with an open architecture, and manages resources afforded through a new object called SOC(Shared Object Class) in order to solve problems attendant upon a P2P(Peer-to-Peer) model. Also, it is so designed as to enable it to be extensible in a variety of ways under a P2P environment through a distributable pluggable function. The SORBA comes up with a new model featuring in particular efficient distribution of computing power and resources and automated management, thus overcoming weaknesses encountered in building an enterprise information system on the existing P2P models.

Keyword : Peer to Peer, Distribited Object Management, Framework Techniques, Object Request Broker

1. INTRODUCTION

Advancement in the Internet and PCs has been accelerating distribution and sharing of information, and existing client-server computing environments have been transformed into a distributed multiple server environment. However, with the source of information being expanded from the web to cover databases and even PCs, and due to the limit of search portals that includes limitation to the scope of information retrieval, imprecision, insufficient catering to individual needs, diversity of formats, and the like, users started to seek alternatives, thus bringing attention to a peer-to-peer model where a number of PCs (peers) act as a server. [10, 11]

For one thing, the Napster service intended to share MP3 music files, a model where the central server is not provided with actual files, but offers information on distributed files spread out over PCs on the Internet, while enabling one to share information among individual PCs, is becoming exceedingly popular. [10, 11]

However, the problem with a P2P service is that it cannot guarantee the reliability of data, and that it cannot come to grips positively with the issue of copyright to the information due to its inability to control at the center. Other problems are that network overload can be worsened for certain PCs that house specialized, specific and repetitive resources like MP3, rather than general data, and the inability to download data while power is switched off. In addition, it provides a new avenue for viruses to spread over.

Therefore, it is assessed that such a set of problems makes it unsuitable to build a P2P-based service in enterprise information system. Besides such problems, actual operation of P2P-based network points to a need for a standardized model that makes it possible to share resources in an efficient way. [8, 9]

In this paper, we proposed the SORBA (Shared Object Request Broker Architecture) model, an object-based P2P model, which makes it possible to solve the foregoing problems while taking advantage of strengths of the P2P mode. The proposed model enables a variety of applications and management to be made among distributed computers spread over a network by managing objects other than a file or a URL as a unit, and makes it applicable to enterprise information systems. At the same time, the proposed model offers a service model designed to run P2P networks more efficiently and operational environments that make it possible to share diverse resources like hardware, off-line relative resource, contents and the like..

2. RELATED WORKS

While there are quite a few P2P-related research projects now under way, among those projects proposing architecture are Legion and Magi™ Open-Source Infrastructure proposed by the members of the Intel' s P2P Working Group, the ' e-speak' , an open source project now under way in HP. [8, 9]

Legion is an extensible, integrated architecture for secure resource sharing in distributed, heterogeneous, multi-organizational context, engineered from first principles to meet the challenges of a P2P environment[9]. While Legion is a model that meets requirements under a variety of environments that include object model, security, scalability, extensibility, site autonomy, heterogeneity, NFS & Samba support and distributed events, it was not intended for the purpose of P2P environments, its key function being storage management under distributed environments. However, the possibility of its application under a P2P environment was proposed recently.

Magi are an architectural framework that explicitly addresses the coordination of e-business messaging and deployment across a range of computing platforms. It is an open-source interoperability specification consisting of complementary protocol standards, formats, and implementations.

In spite of various strengths it has, Magi being an architectural framework designed to perform projects related with the processing of objects on the level of containers, it is difficult to maintain consistency relative with peers in case specifications of containers have to be altered. Also, without reference to specific models or types of resources being shared, it is realized with respect to the processing of events utilizing a Servlet engine through HTTP alone.

An open source project under development by HP, the e-speak is an architecture that enables a variety of P2P-based services to be operated through the standardized API and e-speak engine.

A sort of service container, the e-speak engine is now getting under way, while the target of P2P services is being expanded on the basis of B2B to include applications, computing resources, business processes and the like.

The e-speak features its ability to support resources aimed at P2P services in a variety of ways, and its features being a P2P model aimed at B2B. At the same time, the e-speak proposes a network object model concerning exchanges of component models of the P2P network and shared objects.

3. DESIGN OF THE SORBA

3.1 OVERVIEW OF SORBA ARCHITECTURE

The purpose of SORBA(Shared Object Request Broker Architecture) is to provide ways of processing a variety of services (for example, ASP, etc. supporting SORBA service) including file sharing as a concept of object-based resources without being affected by operating systems, protocols, databases, etc. under P2P environments. To this end, SORBA defines resources in terms of SOC (Shared Object Class), a new form of class, and provides a variety of additional functions that include reading, revising, redistributing and support for monitoring and security. At the same time, SORBA proposes distributive P2P service operation framework and SORBA service, a network model, and API so that it may support the SOC and SORBA service under diverse applications.

As shown in Figure 1, SORBA is composed of SOC and Object Manager intended to manage SOC, Module Container providing operation environments intended to handle service and SOC Request Manager which is a messaging sub system. The proposed model supports CORBA(Common Object Request Broker Architecture) as a middleware for inter-operability among objects, and the Java-based SORBA API in order to provide easy programming environments under diverse hardware and operation systems. For mutual communications, the RMI and RMI over IIOP are employed.

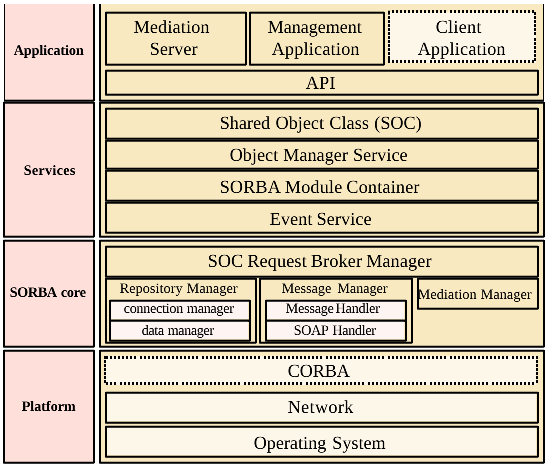

Figure 1. Internal Architecture of SORBA

3.2 SORBA resource

The resources SORBA deals with are files distributed over the network, on-line and off-line relative services, and are defined as featuring:

- ✓ There are numerous resources.
- ✓ These resources are owned and managed by different, potentially mutually distrustful organizations and individuals.
- ✓ The resources and the collection are potentially faulty.
- ✓ There are different security requirements and policies for different resources.
- ✓ The resources are heterogeneous. I.e., they have different CPU architectures, are running different operating systems, and have different amounts of memory and disk.
- ✓ Heterogeneous, multilevel networks connect them.
- ✓ They have different resource management policies.
- ✓ They are likely to be geographically separated (on a campus, in an enterprise, on a continent, etc.).
- ✓ At the same time, SORBA resources are classified under 4 categories in total as shown in Figure 2.

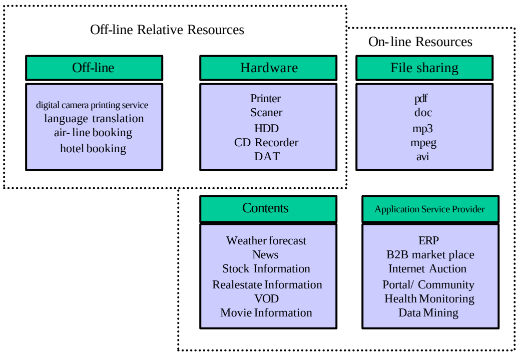

Figure 2. SORBA Resource Categories

SORBA resources of these sorts should be registered as SORBA services, and the registration of service is easily realizable through the SORBA API. Also, resources made without regard for SORBA can be registered and operated through the SORBA gateway and External service interface.

3.3 Shared Object Class

The Shared Object Class (SOC), a key element of the SORBA, is composed of resources to be shared with, pluggable functions and objective descriptor concerning the SOC as shown in Figure 3.

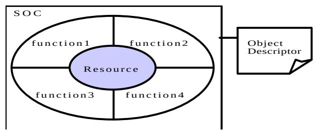

Figure 3. Structure of SOC

The purpose of redefining resources as a new object called SOC is to enhance reliability by preventing computing power from being lowered in order to manage resources and solve problems of resource security through distribution of functions for the management of resources after they are included in the resources. This can also enhance the worth of services by including not only simple files but also diverse functions designed for various purposes.

The functions included in the SOC could be embedded functions or it could be managed as a small object that does not exceed the size of the original file too much when the current model has been applied by designing in such a way that functions present in a remotely located object may be designated. The SOC is managed by the Object Manager, and resources that befit services as a class that can be transformed into a paralleled system can be attached to the SOC, and can be remotely located.

Accordingly, the information on the location of resources is described in terms of URI(Uniform Resource Identifier), which indicates the information on the access of objects on the Object Descriptor. The purpose of managing resources through the URI at the SOC is to operate resources in a distributive manner depending upon computing and network environments while maintaining the information on the proprietor, generator and computer.

3.4 SOC Request Broker Manager

SOC Request Broker Manager exchanges all SORBA messages among SORBA services as an interface to receive SOC requests from remote peers. The SOC Request Broker Manager uses the Outgoing Queue to receive messages from the user. The SORBA Message Service brings up the Naming Resolution and Message Handler.

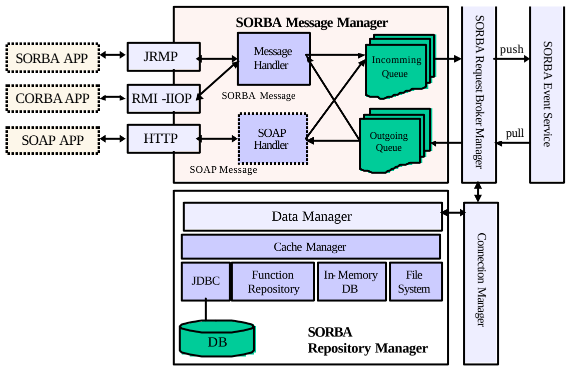

Figure 4. SORBA Request Broker Manager

And the Message is written down on the recipient's Incomming Queue by means of the SORBA Message Service. While the majority of SORBA service handlers are located in the exterior of the SORBA (or application), the Account Manager, Remote Resource Manager, Meta Data Manager, Incomming Queue, Function Repository, Naming Service, etc. as interior modules are located in the Module Container.

Figure 4 shows basic composition of the Request Broker Manager, which determines if the request from the peer is simply a reference or a duplicate of resources, and provides supports that the SOC can be readily handled at the Application through the Remote Resource Manager, Function Manager, MetaData Manager located in the Module Container. At the same time, it plays the role of registering services and providing information on the services through conjunction with the SORBA Repository Manager.

3.5 Network Architecture Model and Network Distributive Algorithm of SORBA

In an effort to solve problems related with a pure P2P network environment, and offer a variety of services, the current dissertation proposes an Enhanced Broker Mediated Model. With a mediation server of various functions mounted, this model is designed in such a way that it can solve problems attendant upon the existing genuine P2P models, while maintaining the strengths of pure P2P systems. The management of the information on shared resources on the mediation servers spread out rather than concentrated on a central server makes it possible to reduce network overload attendant upon increases in the number of users and inefficiency of costs caused by resource databases being enlarged to an extraordinary size.

The current dissertation proposes a distributed algorithm as shown below, with its SORBA network which is designed to reduce network overload to a minimum as well as maximize efficiency in operation.

Algorithm 1. Distributed Mediation Server Management

```
begin Method
 search query on local database;
 if not found then
     broadcast query to all alive Medation Servers;
     create asyncronous receiver thread;
     receive query results;
     if not exist Mediation Server on same subnet then
        regist this server to Mediation Server list;
      braodcast server information on subnet;
   end if
  else
     return found result;
     if client request more result then
        broadcast query to all live Mediation Servers;
        create asyncronous receiver thread;
    end if
  end if
 end Method
```

3.6 Example of SORBA Applications

SORBA that is realized in a Java language including the RMI is oriented toward an open architecture, and supports the CORBA for the purpose of inter-operability with objects on a network. As an experiment designed to validate the proposed models, a file-sharing P2P client has been developed which makes it possible to establish authorities with the aid of SORBA API and ACL functions under the assumption that there exists a SORBA server realized for specific purposes on a basic network model. The SORBA API supports the two programming models as shown in the following, and the current dissertation introduces a model utilizing a Standard Application Model, applications programs being Swing Application, and SORBA ClientAppIntf Interface being implemented. Figure

6 is a picture that shows a new SOC being generated by means of a SORBA client realized. As the picture shows, it is designed in such a way that authorities as to specific files may be set up and the binding done by means of the SOC.

The results of the experiment shows that the SOC is normally generated and distributed, and that those SOCs distributed could cope with changes in the file like infection with viruses. At the same time, the use of ACL functions made it possible to control over the user, computer, reading and writing.

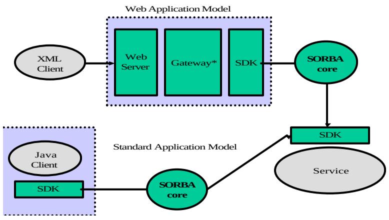

Figure 5. SOC Programming model

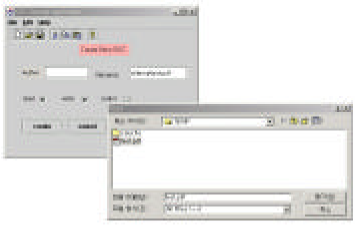

Figure 6. File-sharing P2P Client that makes it possible to set up authorities

4. Experiments and Research

The experiment evaluated the performance of the operation of SORBA in the network environment and the performance of SORBA by means of the realization of the application which generate and set the authority by using SORBA API.

4.1 Experiments on response time according to the number of simultaneous queries

The purpose of this experiment is to evaluate the reliability of successful requests and responses attendant upon increase in the number of queries with regard to distributed SORBA resources. Accordingly, this experiment is designed in order to touch off requests and responses of differing network sizes on an adjustable network, and evaluate the effects. Requests were transmitted by nodes on the network to 500 900 at random, resulting in the following:

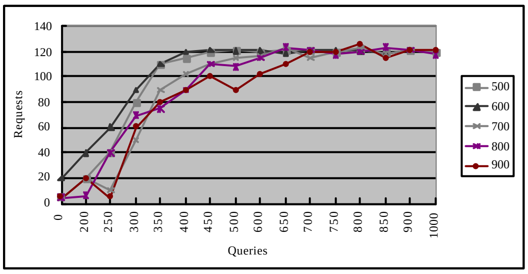

Figure 7. Measurement of requests made by nodes according to increases in the number of queries

The measurement indicates that requests remained constant on a certain level generated depending upon changes in nodes with 1000 as the number of queries, enabling given network resources to be efficiently utilized.

4.2 Evaluation of Network Efficiency

Designed to measure the information retrieval time and network efficiency according to increases in clients, this experiment compared a pure P2P and broker mediated server model and the advanced broker mediated server model proposed by the current dissertation. Here are signs for the experiment.

C_n : Number of clients (node)

M_n : Number of mediation server

bw : Network Bandwidth

P_s : Message packet size

bw_{rate} : Weighted value resulting from uses of network

N_{start} : Number of initial information retrievals

N_{after} : Number of second retrievals after initial information retrievals

N_{tot} : Total number of retrievals

ti_{trans} : Time during which requests are transmitted to the SORBA server

ti_{work} : Time the entire requests are carried out

Here are variables set up for the experiments.

C_n = 50,100,200,500,1000,2000,3000,4000,5000

M_n = 4

bw = 256k

bw_{rate} = 100

P_s = 10k

Time each host is registered on the SORBA server

$ti_{reg} = (ti_{start} + ti_{due}) \log_2^m$

Time during which all the work requests are transmitted to SORBA server

$ti_{all} = n(1/2 + 1/4 + 1/8 \cdots + 1/(2^{\log 2m})) * t_{trans}$

Time during which processed outcomes are received and integrated

$ti_{final} = (ti_{ret} + ti_{sum})\log_2^m$

The number of communications required for information retrieval N_{start} is expressed as follows:

$$N_{start} = (C_n / M_n + (M_n - 1)) * M_n$$

Since a client can retrieve information through the closest mediation server after an initial information retrieval has been made $N_{after} = C_n$ follows as to N_{after}.

Under an assumption that only one single retrieval was made for all the Cn to locate resources for the experiments, the network bandwidth bw was set at 256k and the message size Ps needed for retrieval at 10k., measurements were made up to Cn=500 in the case of pure P2P models. At the same time, in case the bandwidth was exceeded, an assumption was made that a weighted value bwrate was 100.

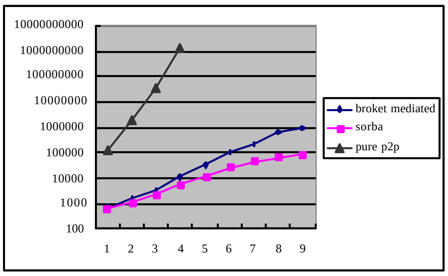

Figure 8. Network performance test

The experiment shows that when a certain level is exceeded general broker mediated type increased in the width of increments in the case of a broker mediated type, while the proposed type showed a gently-sloping increasing curve according to increases in C_n, indicating that compliance with enhanced performance with regard to information retrieval is made. Accordingly, it is expected that this will serve to reduce network overload under a P2P environment.

Another experiment, carried out with Mn, a number of the mediation server, at an average of 2% of Cn, to measure changes in performance attendant upon increases in the mediation server has resulted in the following graph.

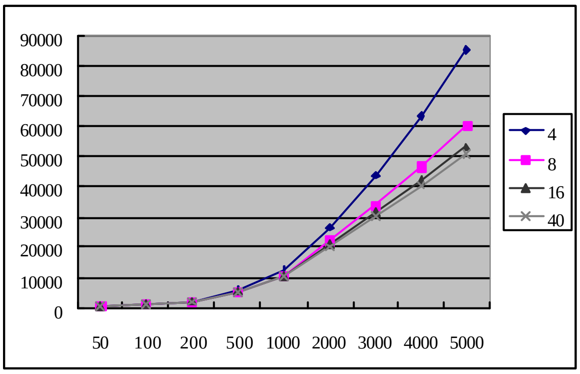

Figure 9. Performance test according to increase in mediation server

The experiment shows that while no enhancement in performance attendant upon increases in Mn is noted until C_n has reached a certain level, but that once a certain level has been reached, enhancement in performance was noted. However, when C_n has reached a certain point again, enhancement in performance starts to slow down attendant upon increases in M_n, requiring maintenance of M_n at an optimum level to be made. This remains a project this model has to solve in the days to come..

5. Conclusions and Future Work

In this paper, we proposed the SORBA, an object-based P2P architecture. The proposed model can manage the resources afforded through a class by the name of SOC as objects, is so designed as to be extensible in diverse ways under P2P environments by means of distributable and pluggable functions.

Accordingly, it is applicable as a new model to computing power and efficient, distributive operation and management in building P2P in the enterprise information system. In addition, it shows a new possibility of P2P by expanding the scope of shared resources that includes hardware on a P2P network and services as associated with off-line, ASP, contents, and the like. The current dissertation also opened a new possibility that it is possible to operate P2P networks more efficiently by proposing a new algorithm for distributive network management. As a further study and research project, a continuous study should be made on the ways of enhancing performance of SORBA service, ways of carrying out configuration of diverse agents for operation of more dynamic services, and on the expansion of pluggable functions offered as a basic requirement.

References

[1] Roy T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures", Ph.D. dissertation, Information and Computer Science, University of California, Irvine, 2000.

[2] Ki, R., Figueira, S. Schopf, J., and Shao, G. "Application-Level Scheduling on Distributed Heterogeneous Networks", Proceedings of Supercomputing '96, November 1996.

[3] Grimshaw, A. S., Nguyen-Tuong, A., Viles, C., and M. Lewis, "Using Reflection for Flexibility and Extensibility in a Metacomputing Environment"

[4] Karpovich, J., "Support for object placement in wide-area heterogeneous distributed systems," University of Virginia Computer Science Technical Report CS-96-03, January 1996.

[5] Roy T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures", Ph.D. dissertation, Information and Computer Science, University of California, Irvine, 2000.

[6] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, D. Winer. "Simple Object Access Protocol (SOAP) 1.1", W3C Note 08 May 2000, http://www.w3.org/TR/SOAP/

[7] D. B. Lange / M. OSHIMA, Programming And Deploying Java Mobile Agents With Aglets, Addison Wesley, 1998.

[8] Gregory Alan Bolcer, Michael Gorlick, Arthur S. Hitomi, Peter Kammer,Brian Morrow, Peyman Oreizy, Richard N. Taylor. "Peer-to-Peer Architectures and the Magi™ Open-Source Infrastructure", December 6, 2000, http://www.peer-to-peerwg.org.

[9] Applied MetaComputing, "Legion– An Integrated Architecture for Secure Resource Sharing", 2000, http://www.peer-to-peerwg.org.

[10] Endeavor Technology, "Introducing Peer-to-Peer", 2000, http://www.peer-to-peerwg.org.

[11] Andy Oram, "Peer-To-Peer", March 2001, O' reilly.