GSEE: A Grid-enabled Value-added Service Platform in NGN

LI Li, YANG Fangchun

State Key Lab. of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing 100876 P.R.China
vvllee@gmail.com
fcyang@bupt.edu.cn

Abstract: GSEE - a grid-enabled value-added service platform in Next Generation Network (NGN) - is proposed to solve the performance problem with traditional service platform of NGN. The value-added services hosted by GSEE are wrapped into grid services. GSEE and other nodes with spare processing capability are organized into virtual organization (VO). A task scheduling algorithm is proposed to overcome the performance bottleneck problem when system gets overload by leveraging existing resources in the same VO. GSEE is modeled by Stochastic Petri Net (SPN) in this paper. And simulation experiment is done based on this model, with the help of SPNP (stochastic Petri net package) to test the performance of GSEE. The simulation result verifies GSEE can really improve system performance.

1 Introduction

These years have witnessed the speedy developments of computing technology and IP based data networks. With the convergence of the telecommunication networks and packet-switch data networks, the so-called Next Generation Networks (NGN) comes into being. NGN makes possible the convergence of existing fixed, mobile, wire and wireless access networks [SM00][DH01]. Application Server (AS) is a key component of the service layer in NGN, which provides support for the creation, execution, and management of the value-added service.

NGN is a service-driven network. The number of services and service type in NGN are much larger than traditional telecommunication network. With the emergence of more and more new services in NGN, the requirement for the computing capability of AS is much higher.

In order to promote system performance, some methods have been adopted in the design of AS [Lyp03][WYZ04]. These methods include the adoption of distributed computing technology, the design of appropriate load balance algorithm and overload control algorithm. The basic idea of these methods is to making the full use of the resources of AS which is a distributed system. These methods can promote the system performance in some extent. But from the viewpoint of the entire network, due to the disequilibrium of the service flow in the network, there are still quite a lot of spare processing capabilities which are not been used efficiently. Here comes the contradiction between the spareness and insufficiency of the process capability. The current research about AS has not solved this problem properly.

Grid technology makes it possible to solve the contradiction discussed above. Grid computing [FKT01] is concerned with coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. Resources in grid can be viewed as a coordinated large-scale virtual pool of resources. Web services emerge as a standard interoperable technology for grid systems. Open Grid Services Architecture (OGSA), oriented service grids, discussed in Global Grid Forum (GGF) has been evolving since it was proposed in the early 2002. According to [Fi02a], the service abstraction may be used to specify access to computational resources, storage resources, and networks, in a unified way. With service interface, how the actual service is implemented is hidden from the user through the service interface.

According to the above thinking, we propose a grid-enabled value-added service platform - GSEE. In GSEE, all the value-added services in NGN will take the form of grid service. With the help of grid technology, GSEE aggregates all the resources existing in telecom operator to provide powerful and extensible resources for the value-added service execution platform in NGN. Thus GSEE can solve the contradiction mentioned above, and promote the system performance further.

The rest of the paper is organized as follows: Section II describes the related work. Section III presents GSEE in details. In order to analysis the performance of GSEE, the Stochastic Petri Net (SPN) model of GSEE is proposed in Section IV, and the SPN-based performance simulation experiment of GSEE is discussed in Section V. Section VI concludes the paper.

2 Related work

Many open network capability standards have emerged in recent years: Parlay/OSA [MK03], JAIN [KTG00], etc. JAIN bases on Java technology, whereas Parlay is protocol neutral. The goal of Parlay is to abstract network capabilities into a set of interfaces to allow service providers including third parties to use and control network resources in a standard manner.

Many Application Servers in NGN are developed based on Parlay. Parlay makes heterogeneous network capabilities homogeneous. The aim of Parlay is not to implement distributed services management, discovery and integration. So an Application Server is necessary for services based on Parlay to be converged in a distributed network environment. The general distributed technologies such as Common Object Request Broker Architecture (CORBA) and Distributed Component Object Model (DCOM) don't fit for loose-coupled distributed objects in the large network scope. As an important emerging distributed computing paradigm, Web services differs from other approaches such as CORBA and Java Remote Method Invocation (RMI) in its focus on simple, Internet-based standards (e.g. eXtensible Markup Language: XML) to address heterogeneous distributed computing [Fi02a].

Based on Grid and Web Services, Open Grid Services Architecture (OGSA) is an important progress. A basic premise of OGSA is that everything is represented by a Grid service. It defines standard mechanism for creating, naming and discovering Grid Service instances. OGSA defines the semantics of Grid Service. Its service model emphasizes the service virtualization, service integration ability and message based communication structure [Fi02a][Fi02b]. All these have the same view with NGN value-added service characteristics. So how to apply grid technology on NGN is worth studying.

3 A Grid-enabled Value-added Service Platform in NGN

As described previously, the introduction of the open API such as Parlay makes the network capabilities in NGN open. The value-added service in NGN (like the Click to Call service) access the network capability by these API. The current version of GSEE is based on Parlay.

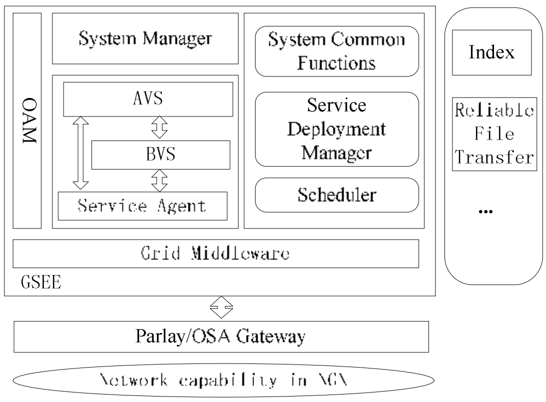

Fig. 1. The Architecture of GSEE

GSEE wrapped the value-added service of NGN into the grid service. There are two types of service defined in GSEE: Basic Value-added Service (BVS) and Advanced Value-added Service (AVS). An AVS is a high-level service which composite some BVSes to achieve some more complex function. The architecture of GSEE is depicted in Fig. 1.

The components of GSEE are described as follows:

- Grid Middleware GSEE is built based on the grid middleware - Globus toolkit.
- Service Agent is the advanced agent of the value-added service hosted in GSEE. All the service requests will be sent to Service Agent firstly, and the Service Agent will transmit the request to the proper service node for real processing.
- Scheduler is in charge of the scheduling the service request to the corresponding node. Scheduler is the core module of GSEE which decides which service node the Service Agent should transmit request to.
- Service Deployment Manager is in charge of the deployment of value-added services in GSEE.
- System Common Functions provides the common functions such as logging, alarm, timer, statistics, etc.
- System Manager provides the management function such as service management, statistics management, charge management, etc.
- OAM (Operation, Administration, and Maintenance) provide convenient management interface for the manager of GSEE including WEB, command line and GUI (Graphic User Interface). All the management command OAM received will be transferred to the corresponding management function in System Manager for the real processing.

3.1 Service Model of GSEE

The service model of GSEE defines the relationship of the value-added services, including BVS and AVS. The service model includes static and dynamic model.

3.1.1 Static Service Model

In GSEE, all the value-added services in NGN are wrapped into grid service. So as a composite value-added service, AVS in GSEE can access BVS and Internet grid service in the consistent way. The BVS can access the network capability by the Parlay Gateway. The static service model of GSEE is depicted in Fig. 2.

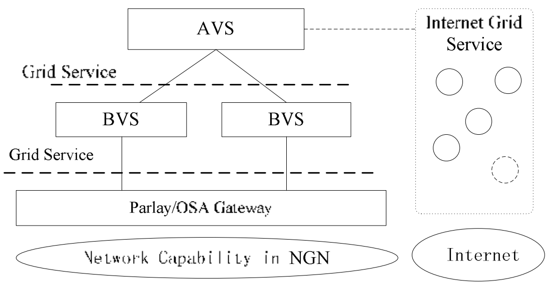

Fig. 2. The Static Service Model of GSEE

From Fig.2, we can see that GSEE realizes the service convergence of NGN and Internet by wrapping all the value-added service into grid service.

3.1.2 Dynamic Service Model

The dynamic service model describes the interaction between the services in GSEE. According to the initiator of the interaction, the dynamic service model is divided into two sub-models: Network Initiate Interaction Model and Application Initiate Interaction Model. These two sub-models define two different interaction scenarios (Basic Interaction and Advanced Interaction) according to whether AVS is the main service flow controller. The dynamic service model is depicted in Fig. 3 and Fig. 4 separately.

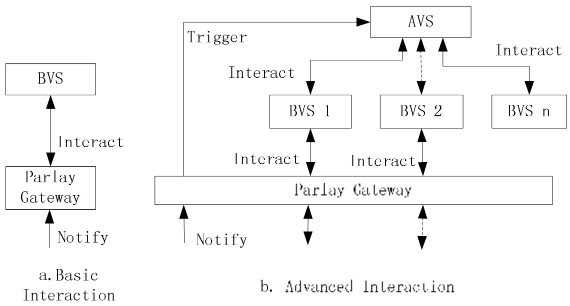

Fig. 3. Network Initiate Interaction Model

In the Network Initiate Interaction Model, value-added service (BVS or AVS) will set the event criteria to the Parlay Gateway firstly. Whenever the event occurs, network will notify the Parlay Gateway, and Parlay Gateway will trigger the value-added service (BVS or AVS) to execute to control the following service flow.

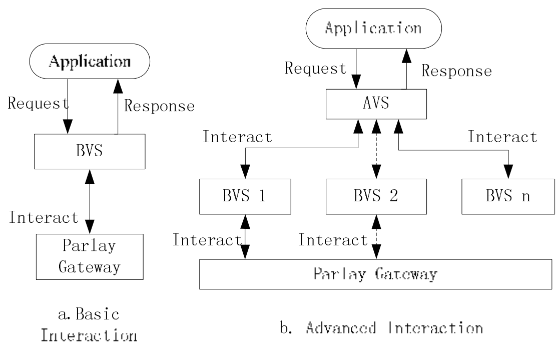

Fig. 4. Application Initiate Interaction Model

In the Application Initiate Interaction Model, the application can send its request directly to the value-added service (BVS or AVS) running on the GSEE. On receiving the request from application, the value-added service will execute to meet the requirement of application. The value-added service in GSEE uses the network capability by the Parlay Gateway.

3.2 Task Scheduling of GSEE

With the help of grid technology, GSEE can leverage the spare resources in the telecom operator to solve the process capability bottleneck problem with traditional Application Server in NGN. In the grid environment, several GSEE systems can construct a virtual organization (VO) according to the specific sharing policy. The GSEE in the same VO can share their processing capability.

As depicted in Fig. 5, GSEE is a distributed system which is composed by several independent nodes. The GSEE software is deployed on these nodes. For the convenience, we name the nodes in the specific GSEE as Gis (GSEE Inner Server); and name the other GSEEs which locate in the same VO and share their resources with the specific GSEE as Gos (GSEE Outer Server).

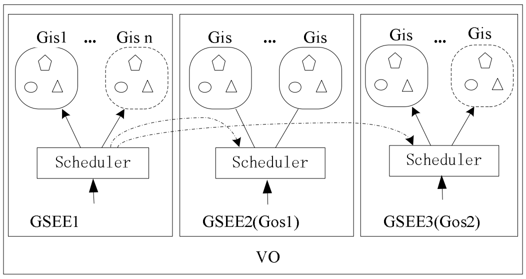

Fig. 5. Task scheduling in GSEE

GSEE is built on grid middleware, and defines Service Agent, scheduler and service deployment manager to enable better resource sharing. With the help of Service Agent, flexible virtual running environment [Fi02a] can be built in GSEE. All service requests sent to value-added service will firstly processed by the corresponding service agent. The service agent will interact with the scheduler to select proper nodes to process the service request. The scheduler is the core component of GSEE. The scheduling algorithm of scheduler is described as follows:

Scheduling algorithm of GSEE

```
define (loadlevel == 3) as the overload state;
define (loadlevel < 3) as the underload state;
while(comes a service request){
  if(exists a Gis, and Gis.loadlevel < 3){
    select this Gis to process the request;
    break;
```

```
  }
  if(for all of the Gis, Gis.loadlevel == 3)
  &&(exists a Gos, and Gos.loadlevel < 3){
    select this Gos to process the request;
    break;
  }
  if(for all Gos, Gos.loadlevel == 3){
    filter this request according to the predefined policy;
  }
}//end of while
```

In the scheduling algorithm, the *loadlevel* of Gis is computed by the system performance parameters such as the CPU Utilization, Memory Utilization, etc; the *loadlevel* of Gos is defined as the minimum *loadlevel* of all the Gis belongs to the Gos. The load information of Gos can be acquired from the Index Service of Grid.

3.3 An Example

Here gives an example of the request processing of GSEE. In this example, the application wants to access a BVS-BVS1, as depicted in Fig.6. This example is based on WSRF [Sb05]. In WSRF, a service instance is defined as a WS-Resource. A WS-Resource is composed of two parts: the Instance Service and the resource. The resource keeps the state data of service instance. A WS-Resource is referred by its endpoint reference. In order to access the specific WS-Resource, the endpoint reference must be obtained.

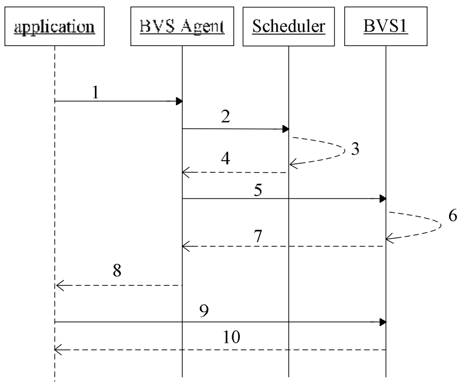

Fig. 6. workflow of an example.

When the application wants to access a BVS, it will firstly request the BVS Agent to create a WS-Resource of the specified BVS (Step 1). On receiving the request, the BVS Agent will request the scheduler to select the proper node to process the request (Step 2). The scheduler selects the proper BVS1 running on the proper node, and notify the distribution result to BVS Agent (Step 3, 4). BVS Agent sends the request to BVS1 (Step 5). BVS1 creates the actual resource and constructs the endpoint reference to the actual WS-Resource back to the application (Step 6, 7, 8). The application sends the request to the WS-Resource according to the returned endpoint reference (Step 9). The WS-Resource of BVS1 executes and returns the result (Step 10).

When the specified GSEE is overload, the scheduler will distribute the request to the node with spare processing capability according to the scheduling algorithm described previously in step 3. From this, we can see that with the help of grid in the resource sharing and coordination support, when GSEE is overload, it will distribute the requests to the outer spare node in the same VO, thus to enlarge the processing capability of GSEE dynamically.

4 SPN Modeling of GSEE

4.1 SPN Introduction

We briefly introduce the main features of SPN and the fundamental concepts in SPN and extensions needed for understanding the model presented in this paper. The reader is assumed to have some basic knowledge of Petri nets such as that given in [Mt89].

SPNs are Petri nets augmented with the set of average (possibly marking dependent) transition rates for the exponentially distributed transition-firing times. A transition represents a class of possible changes of markings. Such a change, also called transition firing, consists of removing tokens from the input places of the transition and adding tokens to the output places of the transition according to the expressions labeled on the arcs. A transition may be associated with an enabling predicate which can be expressed in terms of the place marking expressions. If the predicate of a transition evaluates to be false, the transition is disabled. In SPN models, transitions can be categorized into two classes: Class 1 transitions are used to represent logical relations or determine if some conditions are satisfied. This class of transitions is called immediate transition with zero firing time. Class 2 transitions are used to represent the operations on the tasks or information processing. This class of transitions is called timed transition with exponential distributed firing time.

A marking in a SPN model represents a distribution of tokens in each place of the model. The state space of a model consists of the set of all markings reachable from the initial marking through the occurrence of transition firing. A SPN is homomorphic to a continuous time Mar-kov Chain (MC), and there is a one-to-one relationship between markings of the SPN and states of the MC [CT93].

4.2 System Model

The GSEE system was described in Section 3. The following assumptions are made in our SPN modeling:

- There are m Gises in a GSEE, the j-th Gis is denoted by s_j. Its mean service rate is at μ_j.
- Each Gis has an internal task queue to buffer the request. GSEE has m internal queues, the j-th queue in the j-th Gis is denoted by q_j. The capacity of q_j is $buff_j$.
- There are w Goses, the l-th Gos is denoted by t_l. Its mean service rate is at v_l.
- Each Gos has a buffer queue, the l-th queue in l-th Gos is denoted by r_l. The capacity of r_l is $length_l$.
- The service requests are classified into n categories. The i-th requests are denoted by c_i. The i-th requests are submitted according to the Poisson process with mean rate λ_i. These requests can be processed by Gis or Gos.

Fig. 7 shows the SPN model of GSEE. In this model, the rectangles denote timed transitions and the black bars denote immediate transitions. Requests' arrival and being executed are represented by timed transitions associated with exponential distributed firing time. Requests' being dispatched to the computing nodes and competing for server resource are represented by immediate transitions with zero firing time. The circles denote places, and they contain tokens denoted by black dots. Tokens in the model can represent either service requests or server resources. Request queues are represented by places, and the numbers of waiting quests in the queues are represented by the marking of those places.

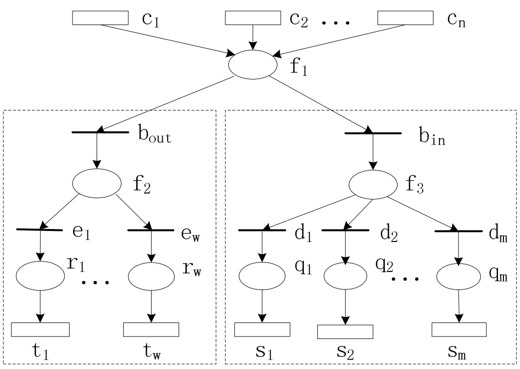

Fig. 7. The SPN Model of GSEE

The meanings of the transitions and places in Fig. 7 are described in the following ($1 \le j \le m$, $1 \le i \le n$, $1 \le l \le w$):

- c_i : models the arrival of i-th requests at rate λ_i.
- f_1, f_2, f_2 : model the judgment function of scheduler to determine which queue should the request be put into. f_1 makes its decision according to the enabling predicate and firing probability of b_{in} and b_{out}. f_2 makes its decision according to the enabling predicate and firing probability of e_l. f_3 makes its decision according to the enabling predicate and firing probability of d_j.
- b_{in}, b_{out}, e_l, d_j : model the dispatching of request, they can be associated with the enabling predicate and firing probability.
- q_j : models the request queue in j-th Gis (Gis_j), its capacity is $buff_j$.
- s_j : models executing requests by Gis_j. Its mean service rate is at μ_j.
- r_l : models the request queue in l-th Gos (Gos_l), its capacity is $length_l$.
- t_l : models executing requests by Gos_l. Its mean service rate is at v_l.

5 Simulation Experiment

In order to verify the performance promotion of GSEE compare to traditional Application Server (AS) in NGN, we use the software package SPNP [CMT89] to analyze the SPN model in Fig. 7.

5.1 Experiment Design

The simulation experiment assumes the quantity of Gis is 2 (m=2). There is 1 Gos in the same VO sharing its processing capability (w=1). The simulation experiment assumes the number of service requests category is 1 (n=1), and the arrival rates of the requests monotonically increase from 2.0 to 32.0 calls/s. The parameters of the experiment are listed in Table 1. The unit of place in SPN is calls; the unit of transition is calls/s.

Table 1. Parameters of the simulation experiment of GSEE

Parameter	m	w	μ_1	μ_2	v_1
GSEE	2	1	10	10	8
Parameter	$buff_1$	$buff_2$	n	$length_1$	
GSEE	5	5	1	5	

The enabling predicate and firing probability of transition b_{in} , b_{out} , e_l , d_j are described according to the scheduling algorithm discussed in the section 3.2. The enabling predicate and firing probability of d_j and e_l are defined according to the Random Selection Algorithm.

- Enabling predicate and firing probability of d_j

The enabling predicate of d_j is $y_j : M(q_j) < buff_j$

The firing probability of d_j is $g_j(M) = \begin{cases} \dfrac{1}{\|RS_1(M)\|} & if\ j \in RS_1(M) \\ 0 & else \end{cases}$, where $RS_1(M) = \{k \mid M(q_k) < buff_k\}$

- Enabling predicate and firing probability of e_l

The enabling predicate of e_l is $y_l : M(r_l) < length_l$

The firing probability of e_l is $g_l(M) = \begin{cases} \dfrac{1}{\|RS_2(M)\|} & if\ l \in RS_2(M) \\ 0 & else \end{cases}$, where $RS_2(M) = \{k \mid M(r_k) < length_k\}$

- Enabling predicate and firing probability of b_{in}

The enabling predicate of b_{in} is $y_{b_{in}} : \exists j, 1 \le j \le m, M(q_j) < buff_j$

The firing probability of b_{in} is $g_{in}(M) = \begin{cases} 1 & if\ \exists j, 1 \le j \le m, M(q_j) < buff_j \\ 0 & else \end{cases}$

- Enabling predicate and firing probability of b_{out}

The enabling predicate of b_{out} is

$$y_{b_{out}} : \forall j, 1 \le j \le m, \exists l, 1 \le l \le w, (M(q_j) \ge buff_j) \wedge (M(r_l) < length_l)$$

The firing probability of b_{out} is

$$g_{out}(M)=\begin{cases}1 & if\ \forall j,1\le j\le m,\exists l,1\le l\le w,((M(q_j)\ge buff_j) \\ & \qquad \wedge(M(r_l)<length_l)) \\ 0 & else\end{cases}$$

We have defined the SPN model of traditional AS in NGN, but due to the length of this paper, the details of the SPN model of AS are not described here. The parameters of the simulation experiment of AS are the same as GSEE which are listed in Table 2. We assumes that AS is composed by 2 nodes (m=2), the mean service rate of each node is 10 calls/s ($\mu_1=\mu_2$ =10). The request buffer capacity in each node is 5 ($buff_1=buff_2$ =5).

Table 2. Parameters of the simulation experiment of AS

Parameter	m	μ_1	μ_2	$buff_1$	$buff_2$
AS	2	10	10	5	5

5.2 Simulation Result

Fig. 8 plots the throughput of service requests in calls/s as a function of the total request arrival rate.

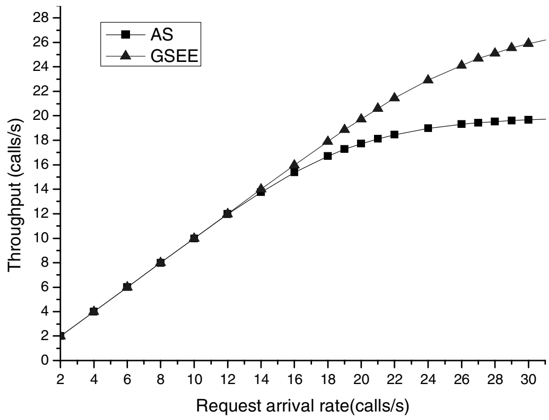

Fig. 8. the throughput comparison between the GSEE and AS

As shown in Fig. 8, the throughput of GSEE has promoted much more than traditional AS in NGN. At low request arrival rate (less than the 17 calls/s), the throughput of GSEE is almost equal to that of AS. As the arrival rates increases, the AS' capacity is exhausted at 20 calls/s. while the throughput of GSEE keeps increasing corresponding. The throughput of GSEE is 1.11 times than that of AS at the 20 calls/s, and the throughput of GSEE is 1.32 times than that of AS at the 30 calls/s. The reason for that is when the capacity of Gises in GSEE is exhausted, the newly incoming service requests will be scheduled to the spare Gos in the same VO as GSEE. The throughput of GSEE is promoted by leveraging the spare resources in the same VO with the help of grid. But traditional AS based on traditional distributed technology such as CORBA cannot achieve this goal.

6 Conclusion and Future Work

With the development of NGN, the number of value-added services in NGN will be much larger than traditional telecom network. And there are much more requirement for powerful processing capability of Application Server in NGN. On the other hand, there are a lot of spare processing resources in the network. Traditional Application Server can not leverage the spare processing capabilities to overcome the processing capability bottleneck problem.

In this paper, we proposed a grid-enabled value-added service platform in NGN (GSEE). GSEE wraps all the value-added service in NGN into the grid service and provides the running environment for these value-added services. GSEE is designed based on grid middleware, and with the help of grid, it can use the spare processing capabilities in the same VO to promote its performance. This paper also proposes a SPN model for GSEE which can be used to do the performance simulation for GSEE. At the end of this paper, the simulation experiment is done based on the SPN model of GSEE. The result of experiment shows that the performance of GSEE can be promoted evidently when compared with traditional AS.

There are some related works to be perfected at present. In the future, we intend to take steps to (1) create more value-added services based on GSEE, especially some resource-consuming services such as mobile and multi-media value-added services, and test the performance, (2) study the issues related to the management service of GSEE, and (3) design a service creation platform based on GSEE to improve the service creation speed.

Reference

[CMT89] G. Ciardo, J. Muppala, K.S. Trivedi. SPNP: stochastic Petri net package. Proceedings of the Petri nets and performance models. Kyoto: IEEE Computer Society, 1989. 142-151.

[CT93] G. Ciardo, and K. S. Trivedi, A Decomposition Approach for Stochastic Reward Net Models, Performance Evaluation, 1993, Vol. 18, pp. 37-59.

[DH01] Y. De Serres and L. Hegarty, Value-Added Services in the Converged Network, IEEE Communication Magn., vol. 39, pp. 146-154, 2001.

[Fi02a] I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Downloadable at:http://www.globus.org/research/papers/ogsa.pdf, 2002.

[Fi02b] I Foster, C. Kesselman, J. M. Nick, and S. Tuecke, Grid services for distributed system integration, IEEE Computer, vol. 35, pp. 37-46, 2002.

[FKT01]. I. Foster, C. Kesselman, and S. Tuecke, The Anatomy of the Grid, Intl. Journal of Supercomputer Applications, vol. 15, pp.200-222, 2001.

[KTG00] J. de Keijzer, D. Tait, and R. Goedman, JAIN: a new approach to services in communication networks, IEEE Communications Magn., vol. 38, pp. 94-99, 2000.

[Lyp03] LI Yong ping. Study of architecture, load balance and fault tolerant technology for the application server in NGN [D]. Beijing: School of Computer Science and Technology,Beijing University of Posts and Telecommunications 2003.

[MK03] A.-J.Moerdijk and L. Klostermann, Opening the networks with Parlay/OSA: standards and aspects behind the APIs, IEEE Network, vol. 17, pp. 58-64, 2003.

[Mt89] T. Murata, Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE, 1989, Vol. 77, No. 4, pp. 541-580.

[Sb05] Borja Sotomayor. The Globus Toolkit 4 Programmer's Tutorial. http://gdp.globus.org/gt4-tutorial/. 2005-11-26.

[SM00] A. R. Seshadri and S. Mohan, Control and management in next generation networks: challenges and opportunities, IEEE Communications Magn., vol. 38, pp. 94-102, 2000.

[WYZ04] WU Wei, YANG Fang chun, ZOU Hua. An overload control algorithm for application server in Next Generation Networks. Acta Electronica Sinica, 2004, 2(7):1112-1115.