私立東海大學

資訊工程研究所

碩士論文

指導教授：楊朝棟　博士

實作 SOA 為基礎的入口網站與服務使用配發機制

Implementation of a SOA-based Service Deployment Platform with Portal

研 究 生：尤士齊

中華民國九十九年七月

摘要

格網提供了強大的計算能力與儲存空間，在網格上面發展了資源監控與工作排程是為了讓格網本身的效能提高，可以將資源利用的效率更好，而後又有人提出了多個網格之間資源的共享與分配的機制，資源仲介作為中介層在不同資源中為開發者提供了更簡便的方式來存取，我們有了越來越多得資源時，開始重新去探討資源的需求來源，如何將這些資源讓使用者可以使用而且是更有效率的使用，所以才發展出了服務導向這樣的概念，原先這樣的需求是從商業中被挖掘出來的，並用將近十年間的技術，嘗試在不同平台不同開發語言環境不同使用者之中尋找一溝通的橋樑，服務導向提供了這樣的機制，讓服務提供者與服務使用者之間有一個標準可以遵循，解決溝通的問題，就會探討到效能問題，在之後就會提到可靠度的問題，這些都解決了那才是應用問題，在這篇論文中使用了服務導向架構來做服務導向，在格網上的資源仲介前端部屬服務，用來存取格網資源，另外使用服務導向的特性也在 Hadoop 上面做了其他服務介面，提供 Hadoop 本身副本的特性，提供可靠的檔案存取服務，以上兩種都是集中式的服務，那使用者端方面的服務採用的也是服務導向介面，讓使用者也是服務提供者這樣的概念去開發的，最後將這些服務註冊在使用者入口網站，使用者可以在網站上直接去使用這些服務，而服務提供者可以在網站上設計服務流程並且管理。

關鍵字：格網計算、服務導向架構、資源仲介、服務流程、雲端

Abstract

In recent years, Grid grows rapidly, it support huge computing capability and storage, it has two important function which is concentrate on get more performance, that is resource monitor and job schedule, by the way it's also combine multiple grid to share each resource and design a algorithm of job submit that comes out a Resource Broker to manage grid. When we get more resources, we need to think about how can we use these, and how can user use resources more efficient, for this reason we use Service Oriented Architecture to provide flexible and usable environment. SOA come up with commercial requirement, it integration many technique cross over ten years to find the solution in different platform, different programming language and different user. The SOA provide the connection with a protocol between service provider and service user. After this, the performance problem and the reliability problem are reviewed. Finally we apply SOA into our Grid and Hadoop platform. Service is like an interface in front of Resource Broker in Grid, Resource Broker is a middleware that provide function for developer. The Hadoop has a characteristic of file replication, it provide a reliability of file. Service provided on Grid and Hadoop are centralized. We design a portal, user can use services on it or service provider can register his service for user, and portal also provide service workflow design, that user can use services each by each job's result and manage it.

Keywords: Grid Computing, SOA, Hadoop, Service Workflow

Acknowledgements

寫這篇論文有太多要感謝的人，在學校要感謝的是我的指導教授　楊朝棟教授，引領我進高效能運算的領域，磨練我做報告與投影片的能力，讓我了解到多個領域要如何整合，雖然有時有點嚴格，但都能讓學生去做自己想做的東西，尤其是在許多新領域上面能讓學生自由發展，創意由此而生，一直到實踐都能有硬體上的資源可以使用，另外要感謝口試委員許慶賢教授、賴冠州教授、李冠憬教授、呂芳懌教授對於格式的叮嚀，嚴謹的精神不管是做研究還是做人處事乃至於時雨春風皆要面面俱到才行，學生自當時時提醒自己。

在實驗室能了解我部分內在的人，陪我一起享受人生甘苦，到最後這個階段還能給我安慰的人，你們真的是太好了，在研究領域上我要感謝在計畫上合作的人，一個是能了解 SOA 架構到底是甚麼，用在甚麼地方，為什麼需要用，沒有你我大概也會懷疑我在 SOA 上面努力的方向，感謝你在醫院方面提供的資訊，另一個是在計畫上幫助我了解整個計畫的走向與目標，因為做了計劃才能感受到如人飲水　冷暖自知，雖不貼切但卻也不遠矣。

最後我的父母在許多層面讓我不用付出太多額外的精神，讓我專心在專業領域研究，某位醫生長輩分享了許多在人生上觀念，讓我有更深的體會，可以在追求技術這個課題上去思考不同的發展，在網路上我要感謝我的精神支柱，那就是跟我一樣同是天崖淪落人，相逢何必曾相識的諸君。

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

According to the Grid Computing developing, we improve the efficient in Grid by resource monitor and job schedule, and also distributed resouce in multiple grid[1,6], but there is more complicated configuration and more difficult to manage resource when Grid grows rapidly, so we use Servcie Oriented Architecture[2] which provide more flexible to deploy our resources.

User side, they can use service just like a function call, and also combine services to complete their work[3] by our portal.When we get more and more servcie, the problem of manage service is comes out, and more and more data format in our portal, so to solve this problem, we develope a service to convert data from one format to other, converter just like a componet in our system, example in medically we convert the clinical data to internationl data format[4], finally we combine these and complete our work.

To get more compatibly with BPEL we also implementation this one, we can get the information about the service provider node's loading by grid monitor, and to adjust our workflow to complete our job. Intraditional the security is design for a group, someone propose a schema can o access control in each machine node[5], in our portal we use a security token schema do this.

1.2 Contribution

In this work, we provide a Service Oritented Architecture platform, service provider can register their service in our portal, design service workflow, user can user their service by their browser in our portal, or get service description document to call

servcie, our portal also can dynamic generate BPEL document by machines's loading in Grid.

1.3 Thesis Organization

In Chapter 2 we first introduce all of the technique and background , in chapter 3 we will introduce the architecture and flowchart, in chapter 4 is experiment and result, finally chapter 5 is conclusion.

Chapter 2

Background Review and Related Work

2.1 SOA

Service Oritented Architecture has three main component, WSDL(Web Service Description Language) is description service attributes、address,etc..user get this document and then access service by standard protocol, to regist service in our portal need use this wsdl document, and then we parser this to get operation、attributes、service's namespace,etc..

If we has these we can call service and design a workflow, to parser wsdl document we use the package WOM and XSOM[10,11], the second is SOAP, this is a protocol over http so it can cross platform, user use this protocol to call service, there is a lot of platform support this by different library.

For example in mobile device there is a ksoap2 library[12], we can develope a program to access our Grid, the third one is UDDI, user can search servcie and get wsdl document location from it, when user get wsdl document to access servcie, we use juddi[13] to provicde this function, and it also has a hierarchy architecture like DNS to support more connection request.

In this work we combine uddi function to our portal. In our portal we use SUN's solution, Netbeans and glssfish to develope our Service Oritented Architecture, it's easy to use and reduce developement time, and also support openESB which is Service Oritented Architecture, otherwise also can use Apache Axis[14] to build the Service Oriented Architecture.

After build the SOA, we design a portal to let user to access servcie directly, and also need the security authentication to each servcie.

2.1.1 SOAP

SOAP is the Web Services used to transfer data encoding format. W3C has developed the SOAP 1.2 specification currently. The main purpose of SOAP is to provide the RPC (Remote Procedure Call) function. SOAP is a XML-based format, the xml parsing performance is concern with the algorithm of the parser. A complete packet of SOAP will be used the “Header + Body” to mark the part of the contents of each message. Header block which carries a representation of a Web resource, which is needed for processing a SOAP message but which a receiver would prefer not to or cannot obtain by dereferencing the URI for the resource carried within the message. In addition to using HTTP, SMTP and other network protocol for data binding and exchange.

The SOAP body is the mandatory element within the SOAP env:Envelope, which implies that this is where the main end-to-end information conveyed in a SOAP message must be carried.

There is no relevant with SOAP standard and the underlying network protocol. Therefore, when we accorded SOAP specifications to produce SOAP documents, regardless of the use of any network protocol can be exchanged SOAP documents. Then both sides can be selected in accordance with the protocol to exchange information.

2.1.2 WSDL

WSDL[24] is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related

concrete endpoints are combined into abstract endpoints (services).

WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME.

A WSDL document defines services as collections of network endpoints, or ports. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions: messages, which are abstract descriptions of the data being exchanged, and port types which are abstract collections of operations. The concrete protocol and data format specifications for a particular port type constitutes a reusable binding. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service.

WSDL four style/use models:

- RPC/encoded
- RPC/literal
- Document/encoded
- Document/literal

The terminology here is very unfortunate: RPC versus document. These terms imply that the RPC style should be used for RPC programming models and that the document style should be used for document or messaging programming models. That is not the case at all. The style has nothing to do with a programming model. It merely dictates how to translate a WSDL binding to a SOAP message.

Likewise, the terms encoded and literal are only meaningful for the WSDL-to-SOAP mapping, though, at least here, the traditional meanings of the words make a bit more sense.

2.1.3 UDDI

The Universal Description, Discovery and Integration (UDDI)[25] specifications define a registry service for Web services and for other electronic and non-electronic services. A UDDI registry service is a Web service that manages information about service providers, service implementations, and service metadata. Service providers can use UDDI to advertise the services they offer. Service consumers can use UDDI to discover services that suit their requirements and to obtain the service metadata needed to consume those services.

The UDDI specifications define:

- SOAP APIs that applications use to query and to publish information to a UDDI registry
- XML Schema schemata of the registry data model and the SOAP message formats
- WSDL definitions of the SOAP APIs
- UDDI registry definitions (technical models - tModels) of various identifier and category systems that may be used to identify and categorize UDDI registrations

jUDDI[13] (pronounced "Judy") is an open source Java implementation of the Universal Description, Discovery, and Integration (UDDI v3) specification for Web Services.

The OASIS UDDI Spec TC also develops technical notes and best practice documents that aid users in deploying and using UDDI registries effectively.

JUDDIv3 feature:

- UDDI Specification version 3.0 compliant
- JDK 1.6 Recommended, but supports for JDK 1.5 and later

- Build on JAXB and JAX-WS standards, tested on CXF
- Build on JPA standard, tested with OpenJPA and Hibernate
- Pre-configured bundle deployed to Tomcat
- UDDI portlets
- Pre-configured portal bundle download
- User and Developer Documentation

2.1.4 WOM

WOM[10] stands for WSDL Object Model. WOM is a Java library to parse a WSDL document and navigate, inspect each of it's element. WOM is based largely on XSOM (XML Schema Object Model).

The current WOM implementation supports WSDL 1.1.

2.1.5 XSOM

XML Schema Object Model (XSOM) [11]is a Java library that allows applications to easily parse XML Schema documents and inspect information in them. It is expected to be useful for applications that need to take XML Schema as an input.

The library is a straight-forward implement of "schema components". Refer to this specification of how this object model works. In particular, this diagram might be helpful.

XSOMParser works like this; You create a new instance of XSOMParser, set it up, then call the parse method to parse a schema file. When a schema is parsed, all the schemas that are refered from it (<include>and <import>) will also be parsed. If you have multiple separate schema files that need to be compiled together, just call the parse method multiple times.Once you parse all the schemas you want, call the

getResult method to obtain the XSSchemaSet object, then use that object to obtain information about the schema.

The primary design goals of XSOM are:

1. Expose all the information defined in the schema spec
2. Provide additional methods that help simplifying client applications.

Providing mutation methods was a non-goal for this project, primarily because of the added complexity.

XSOM consists of roughly three parts. The first part is the public interface, which is defined in the com.sun.xml.xsom package. The entire functionality of XSOM is exposed via this interface. This interface is derived from a draft document submitted to W3C by some WG members. The second part is the implementation of these interfaces, the com.sun.xml.xsom.impl package. These codes are all hand-written. The third part is a parser like figure 2-2-1 that reads XML representation of XML Schema and builds XSOM nodes accordingly. The package is com.sun.xml.xsom.parser. This part of the code is mostly generated by RelaxNGCC.

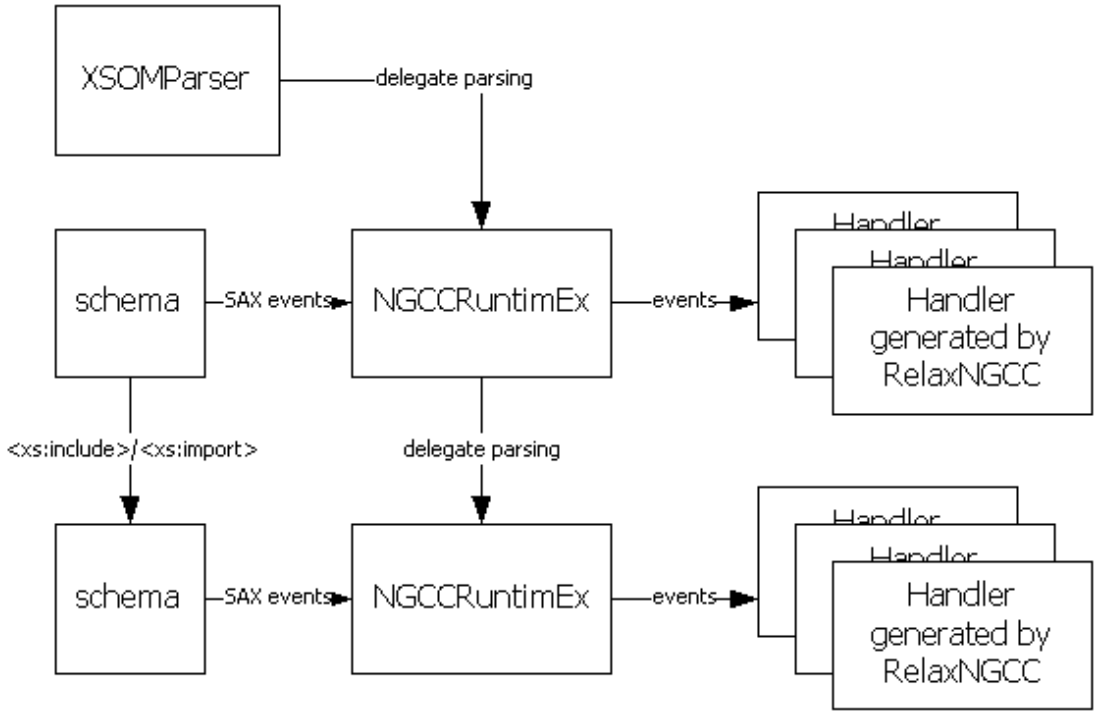

Figure 2-2-1 xsom xml parser

2.2 Resource Broker

The main layers of our system architecture include SOA web portal, Services, Grid portal, Resource broker, Grid nodes.

We use a Resource Broker for Computational Grids. It discovers and evaluates Grid resources, and makes job submission decisions by comparing job requirements with Grid resources. Each rectangle represents a discrete system component. The Resource Broker’s primary task is to compare user requests and resource information provided by the Information Service.

After an appropriate assignment scheme is chosen, Grid resources are assigned and the Scheduler submits the job for execution. The results are then collected and returned to the Resource Broker, which records them in the Information Center database via the Information Service Agent. Users can view the results through the Grid portal.

The system architecture includes five layers: SOA web portal, Service, Grid portal, Resource broker, Grid nodes. The following sections will be described in detail.

ResouceBroker is used to distributed resource and job scheduled, it is imporatl to monitor and distribute, and resouce broker use these information and their algorithm to job scheduled, in grid we use NWS[26,27] to aggregate each node's CPU loading, how much memory used and network bandwidth. We use globus to do job scheduled, globus toolkit[28] is a middleware which implementation WSRF, it's include OGSA OGSI integration, globus toolkit's function has resouces monitor manager, file transfer(GridFTP), file manager, data security(GIS), and also MDS(monitoring and discovery system) to aggregate information、register service[28], etc..it's similarly uddi. globus also can work with MPICH to run MPI program, in our portal also do this, and the resouce broker monitor this.

2.3 Medical Grid

Medicare Grid, a grid based on E-health system[19,20], to fulfill a practical future medical scenario for next twenty years. It uses grid and peer-to-peer technology to integrate all the computing and storage resources and develop medical data exchange mechanisms base on it. Those data form a data warehouse to support the data mining techniques that extract valuable information for medical decision support and future research.

Medical Grid is to establish from individuals, families, hospital physicians to complete medical service mechanism, in order to improve the way we have to be treated in a hospital nowadays. It can not only take care of the old, handicapped persons or anyone who need long-term medical care but also decrease the opportunity of accidents and prevent from cardiovascular diseases taking place. Moreover, it would reduce the expense of medical insurance system and create a better life for all human beings.

This system construct of one medical data exchange and sharing platform. Allow any person in any place for medical treatment, keep to have a complete patient record. To help physicians predominate the patient's medical history in order to provide complete medical services. It uses data mining technology and methods to provide the provision of medical decision in the grid platform. We depend personal medical information to carry out follow-up medical (such as medication, medical diagnosis, care, etc.) analysis, to reduce medical expenditures. To medical quality assurance, RFID technology will be integrated into the emergency care, surgery safety, patient safety and medication management. According to community as a unit, the development of wearable physiological signal flow measurement system.

Nowadays the application of grid computing instrument, and did not have a set of comprehensive use of a mechanism for users to more easy way, just in their own

computer to install the public on the widespread use of the software, you can network through the Internet on the specific scope of Grid resources to carry out access, efficient computing and control platform. First of all, we want to develop the Resource Broker will make use of Globus Toolkit as the future of the Grid software agent.

The main objective of this program is to develop a Grid Monitor, used to monitor the computing platforms on all available computing resources of the current status, such as machine load on CPU, Memory size available, the node between the current instantaneous frequency wide, and the network flows in each node shown and job monitor shown. This Grid Monitor is NWS (Network Weather Service) as the main core, by the NWS is responsible to retrieve the information (bandwidth, flow rate, CPU load, etc.), and then use RRD Tool with MRTG to carry out mapping to produce page, supply users browser and Jane transaction. It show the Ganglia snapshot from grid nodes. Then a set of questions future prediction model at the grid on the network flow and execution time for job applications.

2.4 Integration Resource Broker with Serivce

SOA combination of elements typically include: software components, services and flow. When the hospital or enterprises face an external request, the external flow is responsible for the definition of the requirements of treatment steps; Services of specific steps, including all program components and software components are responsible for the implementation of programs. We use the characteristics of SOA, and resource brokers (Resource Broker) for the combination of resource brokers go through the medical access grid (Medicare Grid) system. The High-level interactions between various entities, components, resources, and human participants that the channels of control as shown. It is described below:

1. SOA components will be imported from time to time updates on web services,

the service's Web Services Description Language (WSDL) to release to be published on the SOA web portal for users to search.

2. User initiates session with portal by selecting modeling resources required and ‘planning’ experiment with the resources. .On the SOA web portal, the user can search for suitable services and resources.
3. User obtains authorization through VO and authorization service.
4. User to get the service of WSDL document.
5. WSDL of the service allows users to be connected to the Internet through the SOA components, and will invoke service.
6. SOA portal through the SOA technology contacts Resource Broker to request resources (remote/local services or database). Medical Grid and P2P network. SOA component will take submission of tasks to resource broker.
7. Resource broker submitted the tasks to Medical Grid or P2P network in order to achieve the goal.

8-10. Results are visualized on the users’ desktop through the Web browser.

2.5 Resource Broker Providing Serivces

We identify the following modules in our paper:

Resource Status: The main function was provided users to view the machines status when user create job, then they can choose the correct machines. The user chooses one or more nodes and sends a request to the Information Service, which finds the information and returns it in a sorted list. Information type include hostname and machine status.

Job Submit: The users can submit job into the Workflow Engine by the SOA middleware, and then will be allocated to the grid environment for computing machines by the resource broker (RB).

TMT/CDA Convert: the general public of the personal health record information, because of differences in medical situations may be scattered in a variety of medical institutions; to make the information on to the network for the exchange and integration, we need the same information in a standard format will be edited. We use international standards as the Health Level 7 as the exchange of information integration.

Handset service: We use SOAP protocol as intermediate. kSOAP libraries a SOAP web service that can connect with handset devices.

2.6 TMT/CDAR2

TMT is a clinical data, it's used in most taiwan's hospital, and CDA r2 is internationl standard, these are xml format document.In our platform, we put these data into Hadoop system[21], it's a distributed system and per data has three backup copy, we also write a Map&Reduce program to analyze and statistic clinical data, and generate a trend module.

Since a variety of time and space factors, the general public may be at different hospitals for treatment, and various medical institutions to use the hospital information system (HIS) is not necessarily the same as a result of patient medical information can't be at each inter-hospital transfer, often creating the need for patients to check the duplicate, or duplicate medical acts, resulting in unnecessary waste of medical resources. By the HL7 Organization for medical information exchange standards, the role of medical information is structured, standardized so that they can through the network, at different medical institutions between the transmissions, to achieve the purpose of information sharing.

Health Level Seven (HL7) [28, 29, 30] is one of several American National Standards Institute (ANSI) -accredited Standards Developing Organizations (SDOs)

operating in the healthcare arena. Most SDOs produce standards (sometimes called specifications or protocols) for a particular healthcare domain such as pharmacy, medical devices, imaging or insurance (claims processing) transactions.

Health Level Seven's domain is clinical and administrative data. HL7 standard interface to pass through different borders and cultural messages to promote health messages also have permeability and dilated. HL7 Organization founded in 1987. Its main purpose is to develop various types of medical information systems, such as clinical, banking, insurance, management, administration and inspection, such as electronic data standards.

HL7 protocol brings together different vendors application software used to design the interface between a standard formats, which allows the various medical institutions of different application systems, carry out some important information to communicate.

The protocol design while retaining a considerable degree of flexibility, making some information on the specific needs of a deal to maintain compatibility. The HL7 Organization is based on ISO (International Standards Organizations, ISO), the use of open systems architecture (Open System Interconnection, OSI) communications model as shown**.** The HL7 is satisfied for the highest level, that is, the application layer. It provides norms such as: the relevance of the classification, the emergence of an effective inspection, the structural mechanism for the exchange of information and consultation functions.

Taiwan health information exchange agreements seventh layer Association (HL7 Taiwan) was established in 2001, and at the Department of Health under the guidance of continuing education to organize various activities such as training and seminar, and actively promote the HL7 standards in the domestic [5].

2.7 BPEL

WS-BPEL is an XML-based language defining several constructs to write business

processes. It defines a set of basic control structures like conditions or loops as well as elements to invoke web services and receive messages from services. It relies on WSDL to express web services interfaces. Message structures can be manipulated, assigning parts or the whole of them to variables that can in turn be used to send other messages.

More about the specification can be found on our BPEL page. In our system, This thesis use Apache ODE(Orchestration Director Engine) [16]to generate BPEL document, and read it to our portal[15].

Apache ODE (Orchestration Director Engine) executes business processes written following the WS-BPEL standard. It talks to web services, sending and receiving messages, handling data manipulation and error recovery as described by your process definition. It supports both long and short living process executions to orchestrate all the services that are part of your application.

WS-BPEL is an XML-based language defining several constructs to write business processes. It defines a set of basic control structures like conditions or loops as well as elements to invoke web services and receive messages from services. It relies on WSDL to express web services interfaces. Message structures can be manipulated, assigning parts or the whole of them to variables that can in turn be used to send other messages.

ODE v1.3.4 Features:

Side-by-side support for both the WS-BPEL 2.0 OASIS standard and the legacy BPEL4WS 1.1 vendor specification.

- Supports 2 communication layers: one based on Axis2 (Web Services http transport) and another one based on the JBI standard (using ServiceMix).
- Support for the HTTP WSDL binding, allowing invocation of REST-style web

services.

- Possibility to map process variables externally to a database table of your choice.
- High level API to the engine that allows you to integrate the core with virtually any communication layer.
- Hot-deployment of your processes.
- Compiled approach to BPEL that provides detailed analysis and validation at the command line or at deployment.
- Management interface for processes, instances and messages.

Some paper[32] use bpel to integrate grid resource, and use grid's middleware which provided algorithm to send job with workflow. It use four layers, each layer has service which can be used by workflow.

Four layers:

- Processes (workflows)
- Scientific Serivces
- Infrastructural Serivces
- Resources & Data

Processes is a workflow layer with bpel technique, Scientific serivces layer is a component of infrastructural services, we know to complete work that we always need use more than one service, and in different scientific domains needs different data format.

Infrastructural Service provide the different platform like grid, in this layer use the middleware's api to interact with the grid, and follow the standard OGSA, WSRF to developing the service.

There is some component in the paper:

- Workflow
- Service interface
- Component in different science domains
- Data access
- Service entry points
- Data structure (format)

and use these componet to integrate with bpel.

Chapter 3

User Portal

SOA Portal provide user authentication, after that and user can search servcie, portal parser the wsdl document to get the service operation node, service provider can register their service on the portal[17,18], user can access service throught portal or get the service information and access the service by itself in user's browser, this technique is implementaion by javascript,figure 3-1 shows the portal's flowchart.

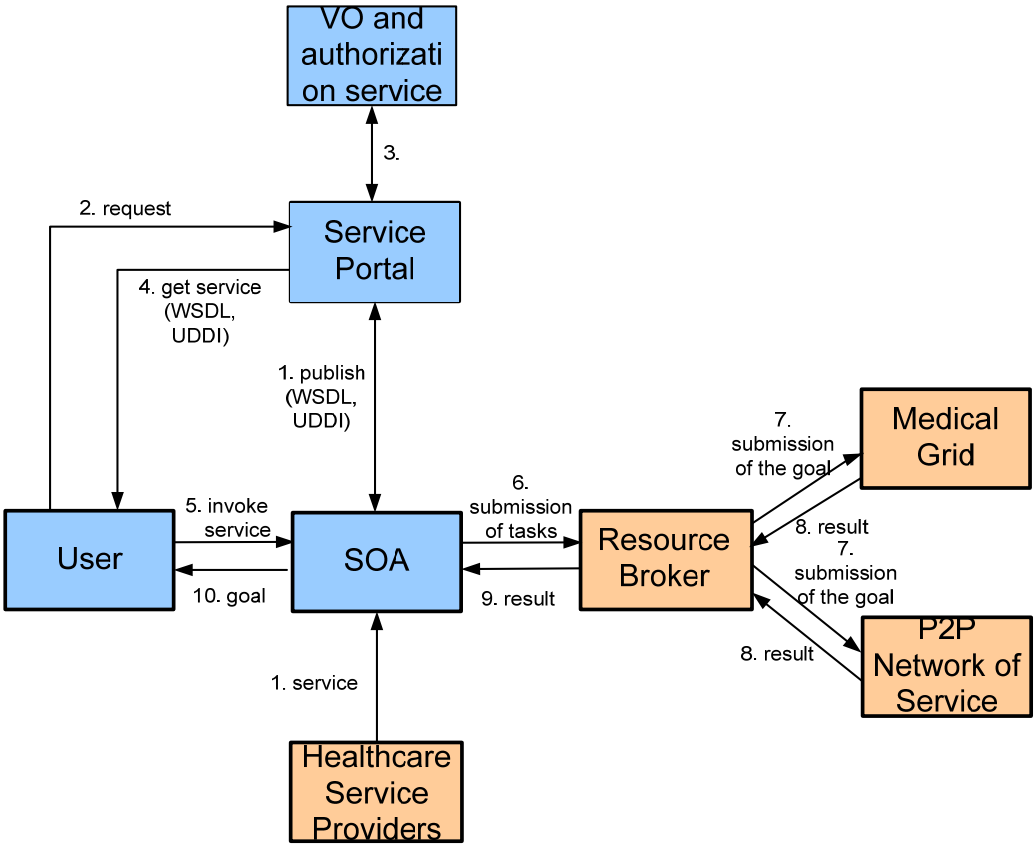

Figure 3-1 portal flowchart

There are steps of use service.

In Step 1, Service provider registry their services into portal.

In Step 2,3,4, User send request to portal, and get WSDL document which can used to access service.

In Step 5,6,7,8, Portal according to service's request and send the request to our resource broker or the p2p network.

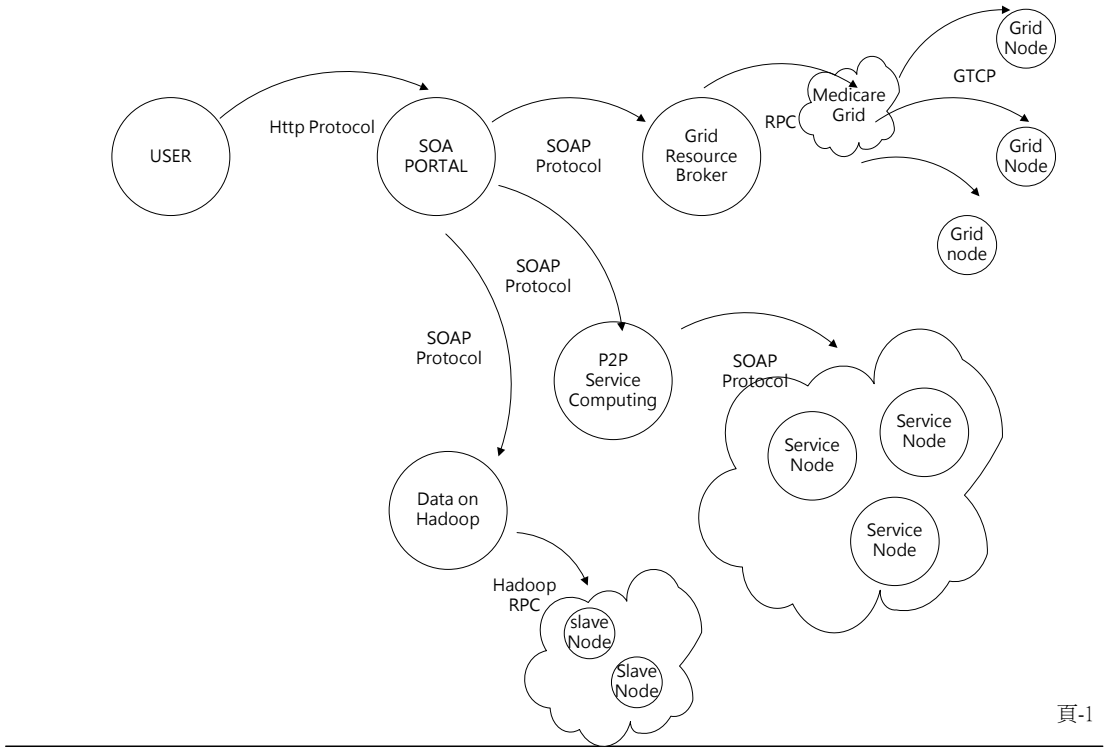

Figure 3-2 Architecture

Expand the service model into a large scale; you can see the architecture in figure 3-2, the portal use Grid by resource broker, access p2p service network, and Hadoop. To combine these, first we need design an interface use SOA, the portal can access these resource by call service with SOAP.

For example user want to access grid service, the portal will call Resource Broker which manage Medicare Grid, and resource broker use grid protocol to complete the job[22,23]. In p2p service network, each client has a SOA interface, the portal can access directly, and user is not only the user of the service but also the service provider. In Hadoop, there is also a interface provide TMT clinical data service, it has three backup, and a small load balance technique.

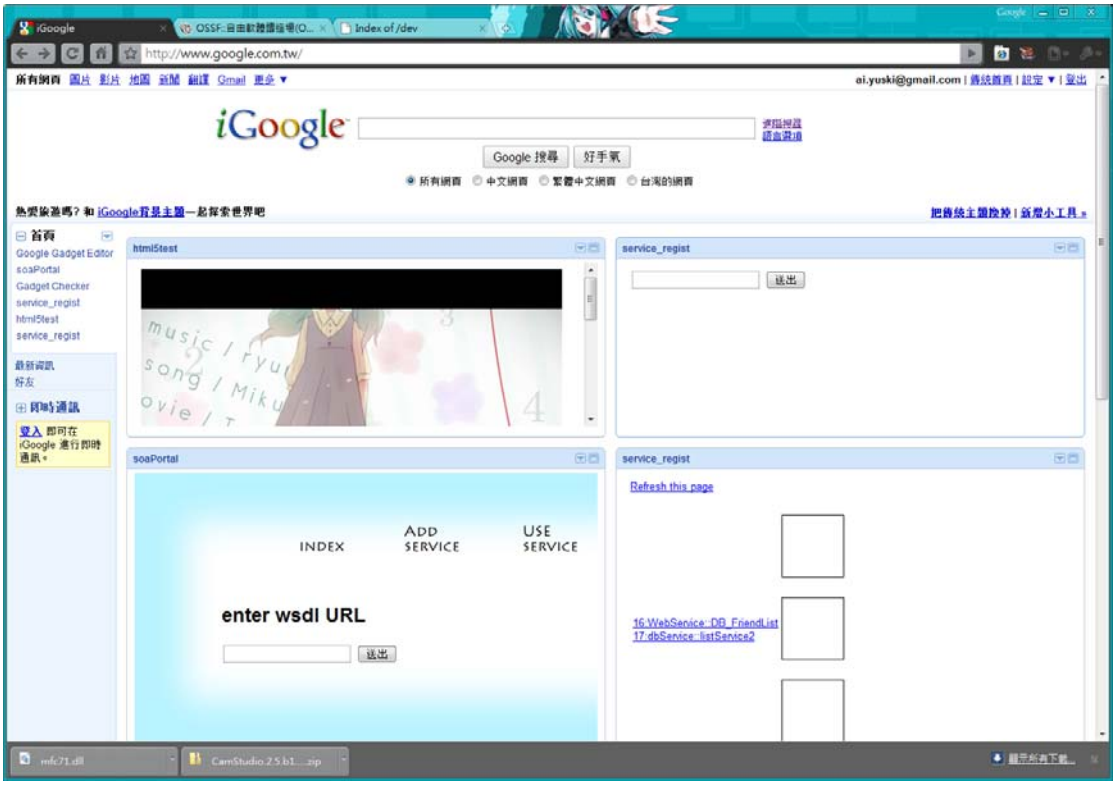

Figure 3-3 igoogle portal

User interface of portal use igoogle gadget like figure 3-3, this portal use HTML5 to implementation Drag&Drop and H.264 video play. By the way, this portal also uses web os to implement these, but didn't deploy finally, why? It's a long story. :(

3.1 Igoogle and EyeOS

iGoogle[31] provides users with their own private page to personalize with content that they find most interesting and useful. Each iGoogle page is as unique as the person who creates it. Some use iGoogle to keep a pulse of the day, frequently checking news headlines, stocks and weather. Others track their favorite blogs, play games, or keep a list of their to-do's. You can develop a gadget to satisfy any of these needs and preferences.

The integration of OpenSocial with gadgets gives you an opportunity to enhance your content for users by incorporating social features. For example, a books gadget could display what a user's friends are reading, allow users to request to borrow books

from friends' libraries, and show users books that their friends recently rated. As users share content with their friends, your gadget will naturally build a broad audience for distributing content and driving traffic.

Eyeos is a disruptive desktop entirely usable from a web browesr. It includes an office suite and some sollaboration applications, as well as full framework to develop new web apps. The execution screen is like figure 3-4.

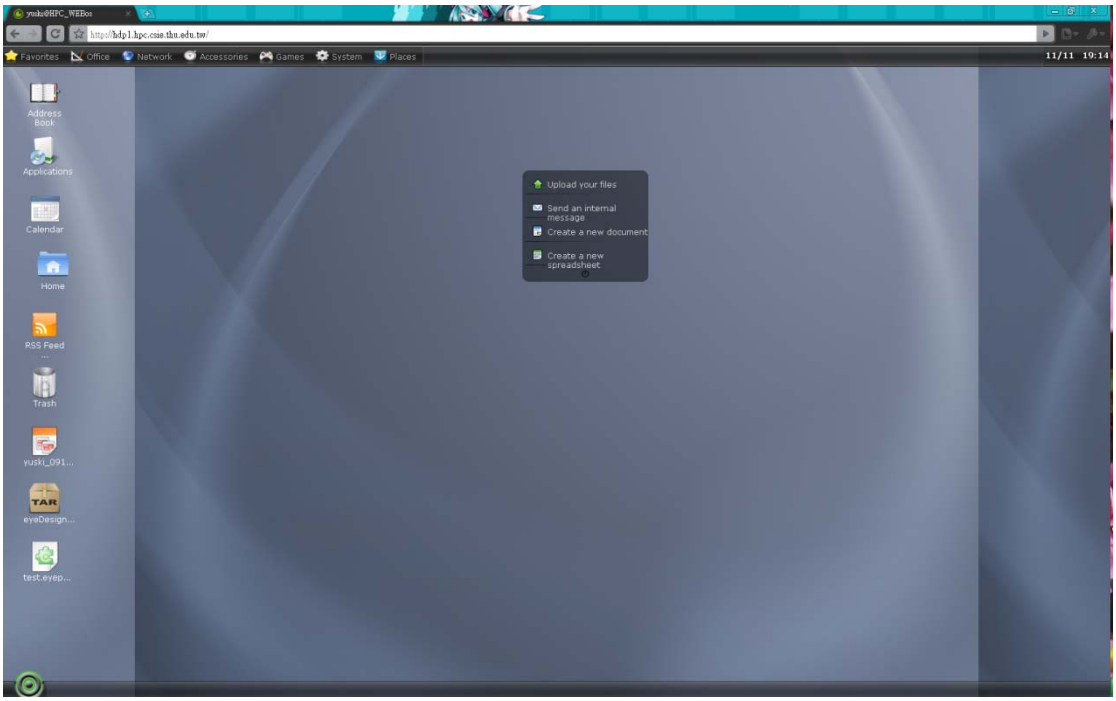

Figure 3-4 eyeos run in web browser

3.2 Service Registry and Using Service

In this thesis implementaion such service, TMT convert Service to convert medical data, jobsubmit service to send job to resoucrbroker, register service to register service into our portal, wsdl parser service to parser wsdl document for register service like figure 3-5, database access service to save or read data into database for register service, user authentication service to authenticate, monitor service to monitor grid's node and we can use this for load balancing. All these service to compose portal, we use these service to complete our work.

Figure 3-5 Service Register

If we want register a service, there is a flowchart:

User authentication service -> register service -> wsdl parser service -> database access service

User can combine services to complete their work, portal provide this service, in figure 3-6 we can drag&drop service that register in our portal into the block, and it also generate the bpel document for compatibility.

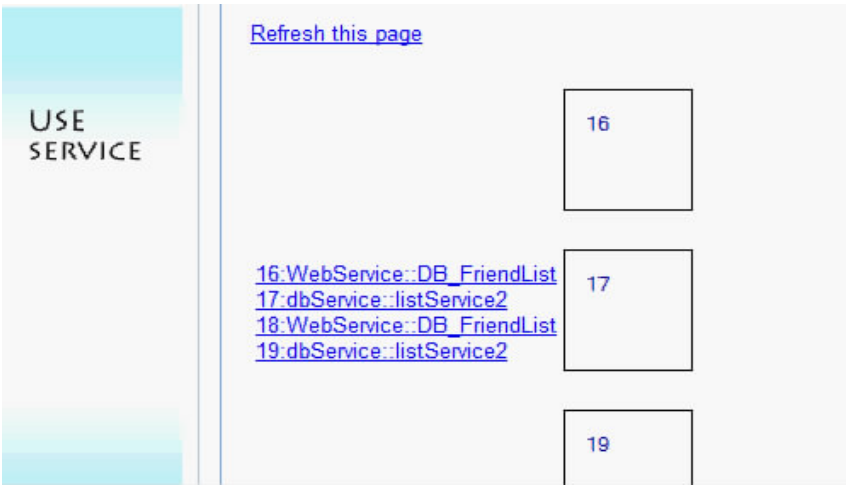

Figure 3-6 Service Workflow

The service Hadoop provide is the TMT clinical data service, it store data in the Hadoop, there clinical document list in figure 3-7, choose Hadoop because it has a

nice backup schema you can see the backup in figure 3-8, also can use the map&reduce to statistics information of the clinical data. In this work is parsing the document which is xml format and display html format like figure 3-9.

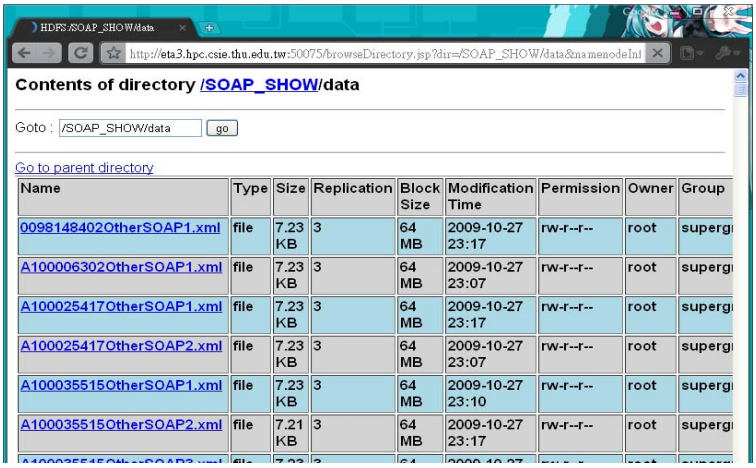

Figure 3-7 TMT on Hadoop

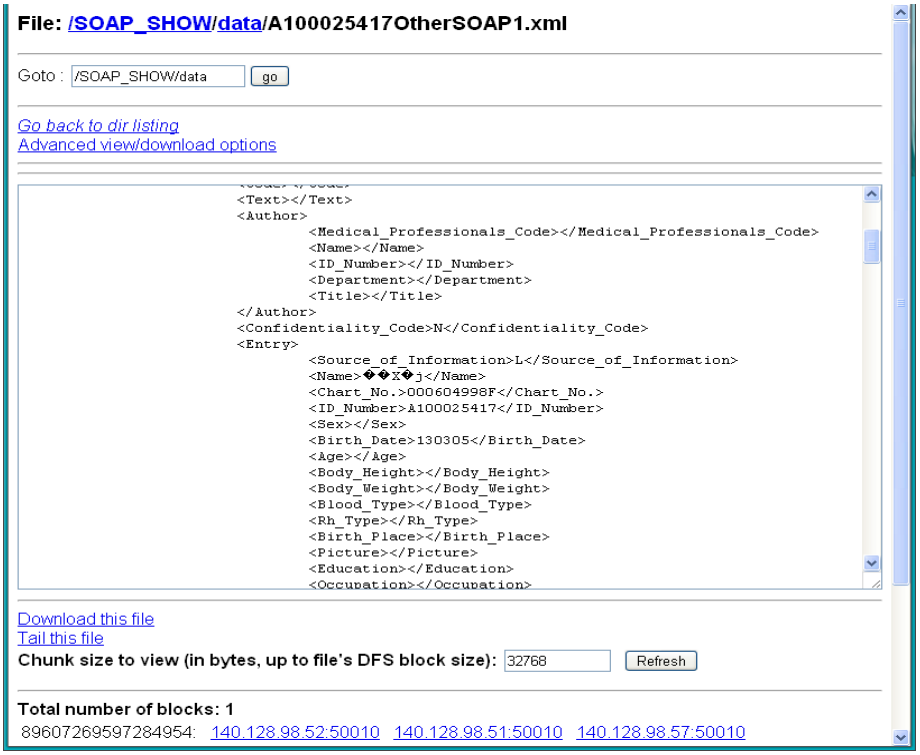

Figure 3-8 multiple backup copy

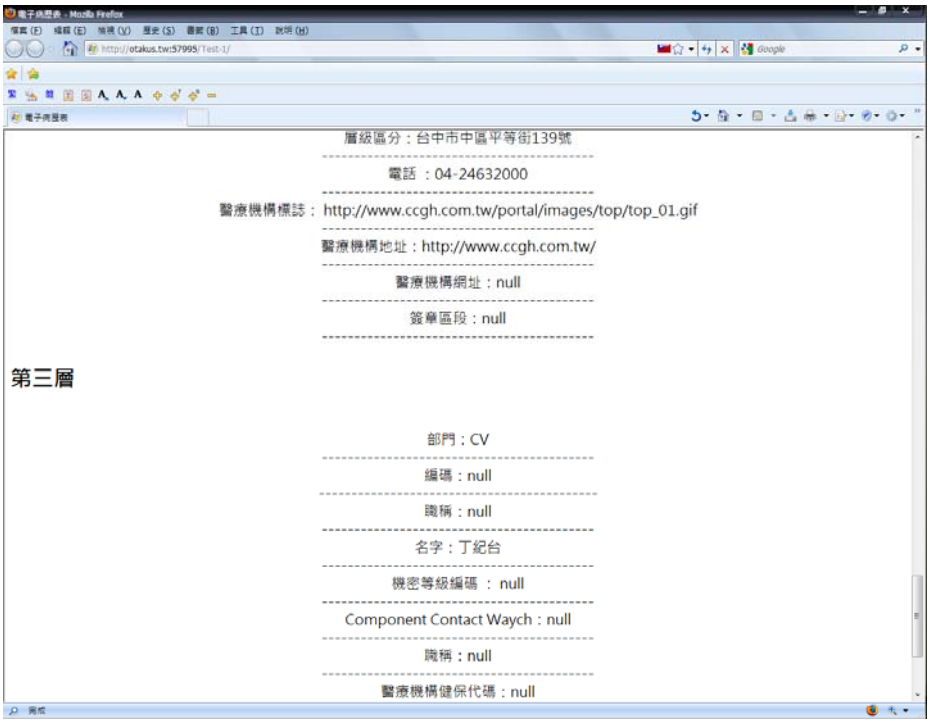

Figure 3-9 TMT parser

To use service in this thesis use ksoap2 library in our portal, and in portal also has workflow schema, to compatibly portal will auto generate BPEL document.

3.3 Load Balancing Access And Dynamic Deploy Servcie

In workflow, our portal can get the machine loading data, automatically choose the service which can work, if thereis no machine can work, and the service can deploy independly, the portal will do this, and generate BPEL document.

The portal deploys service according to the machines computing ability and the CPU loading in service request time. First the portal check the machines CPU loading, if all machines loading is 90 percents up, it will send the request to the fastest machine which calculate by compute pi before.

Chapter 4

Experimental Environments and Result

4.1Testing the latency of job submit in Grid

The environments has two, one is in Grid, the other is in Hadoop, in Hadoop has 5 nodes, you can see it in figure 4-1

Live Datanodes : 5

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
eta1	2	In Service	65.13	0.22	12.45	52.46	0.34		80.55	28985
eta2	0	In Service	65.03	0.36	9.11	55.56	0.55		85.44	28253
eta3	0	In Service	64.2	0.27	9.55	54.38	0.43		84.7	28194
eta4	0	In Service	64.2	0.31	8.79	55.11	0.49		85.83	28689
hdp2	1	In Service	48.43	0.37	5.77	42.3	0.75		87.34	28730

Figure 4-1 Hadoop node

In Grid we send the job into resource broker by our job submit service, you can see it has a job in the figure 4-2. We send the job from handset devices, a android platform system mobile phone.

Figure 4-2 jobSubmit

Calculate the latency time of job submission in personal computer (PC) and handset device. The latency time is denote by a job submit to the resource broker until it return “true” message to user. We can see the result in figure 4-3. The latency time of handset device is higher than PC about 3 times. Because the handset devices use wireless network and android system use SOAP protocol.

It spent more time to use XML parser to explain the SOAP message. The SOAP encoding rules include some type of additional information. It will take up some storage space and increase system encoding and decoding of the additional processing time. But the highest latency time of handset device is close to 1.5 seconds. The time can be ignored for user.

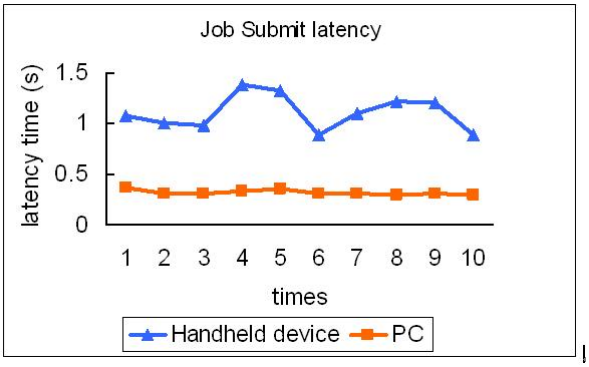

Figure 4-3 Job submit latency comparison.

4.2 Testing Service Computing with Portal Load Balancing

In this thesis also test a service load balancing with portal, analysis the relation between the machines when the amount of request happening, Testing Program is calculate PI, use formula Gregory's Series. The Formula is pi/4 = 1/1 - 1/3 + 1/5 - 1/7 +...

$$\pi = 4\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1}$$

First deploy this program into each machines to test their ability, you can see the

service interface display with html format in figure 4-4.The precision is only 8 digits, because the formula is hundreds years old, and the calculate loop cycle is 1 billion, the execution time is 9 seconds.

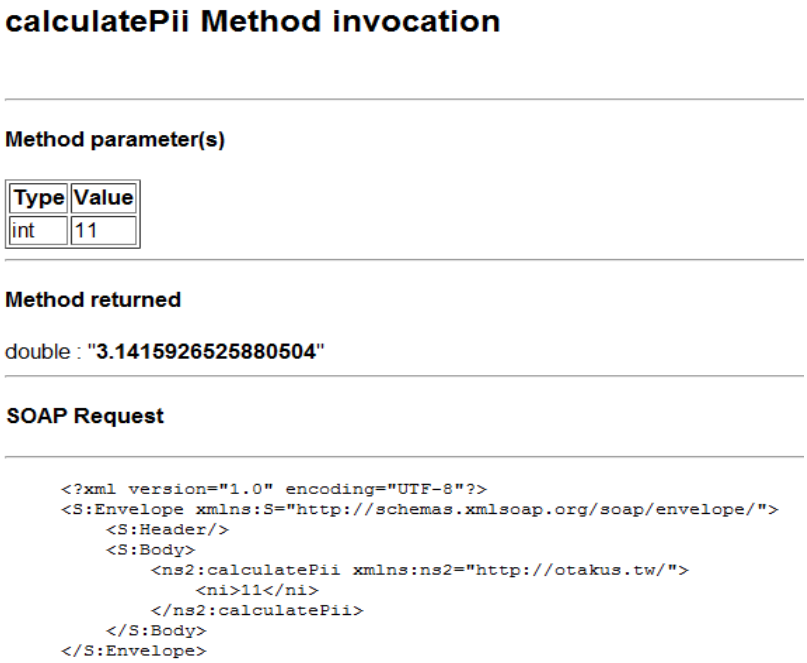

Figure 4-4 program of calculate pi

Machine name	CPU	Cpu clock	cores
Otakus	Core2 quad Q9550	2.83GHz	4
Beta3	Opteron 246	2GHz	2
Beta4	Opteron 246	2GHz	2
Beta5	Opteron 2212	2GHz	4

Table 4-1 machines list

The first experiment is test each machine's ability, all these data will be used for service load balancing by portal, the environment include 4 machines you can see in table 4-1, one for service workflow, portal and dispatch service request, others will compute pi. Otakus is a portal dispatch service request to others, and get each machines loading. You can see figure 4-5, beta5 and otakus are quad core machine,

their execution time is less than the two core's machine, in computing mode, the beta5 is the fastest, the portal service request dispatch will send more job to beta5.

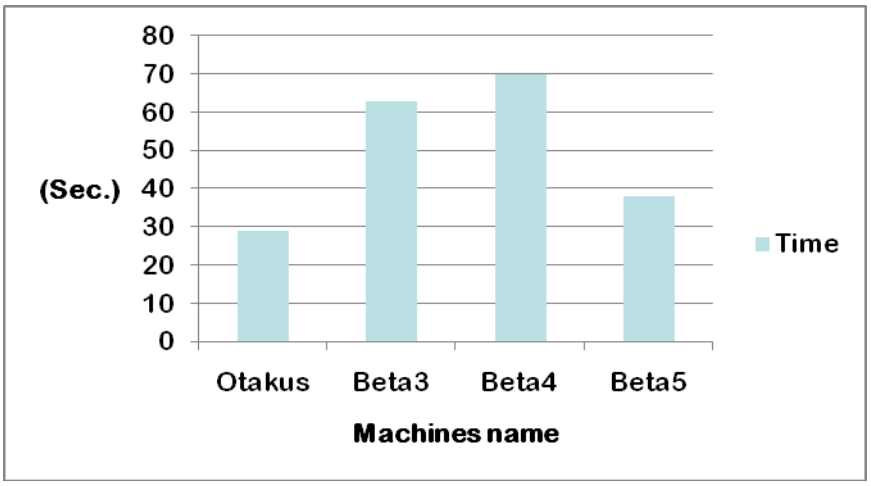

Figure 4-5 calculate each machine's ability

The second experiment will test one machine versus three machines, the computing time will reduce dramatically or predictability. The test setting is Beta5 machines versus Beta3+Beta4+Beta5 and sends 50 calculate pi request. In figure 4-6 you can see three machines is fast than one.

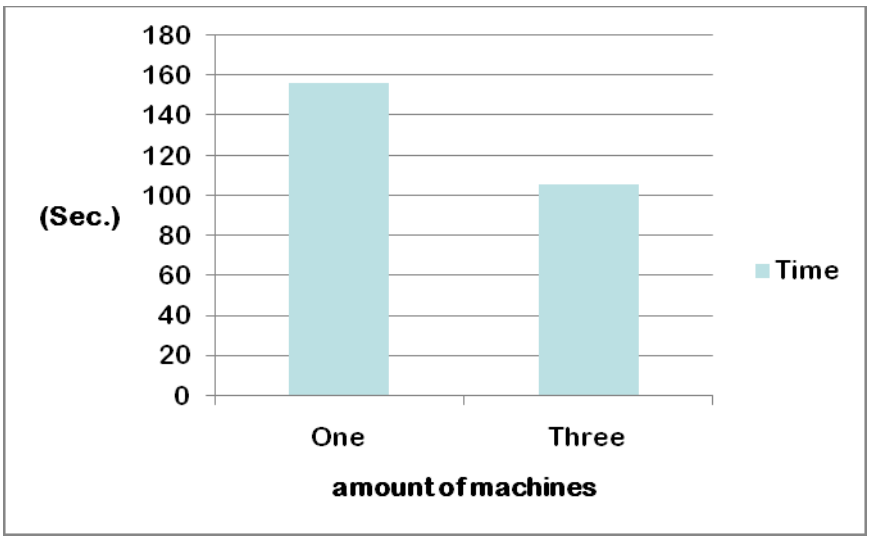

Figure 4-6 one machine versus three machines

The third experiment send amount of request to test the portal, in figure 4-7, although the request is 8 multiplying but the execution time is less than it.

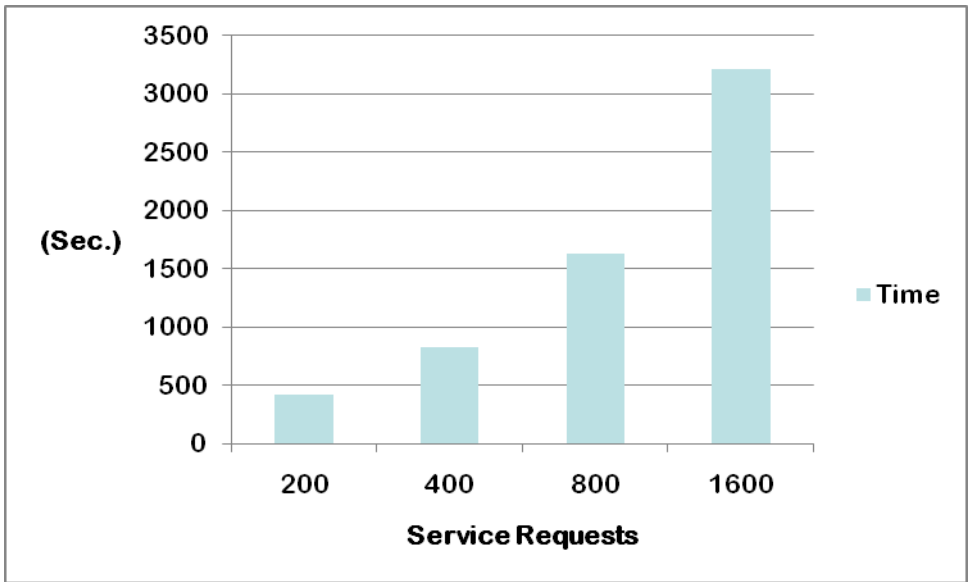

Figure 4-7 amount of request test

The forth experiment is analysis the service request distributed, you can see the figure 4-8, the Beta5 machine is fastest so half of service will send to it, the portal do this load balancing and it looks work perfectly.

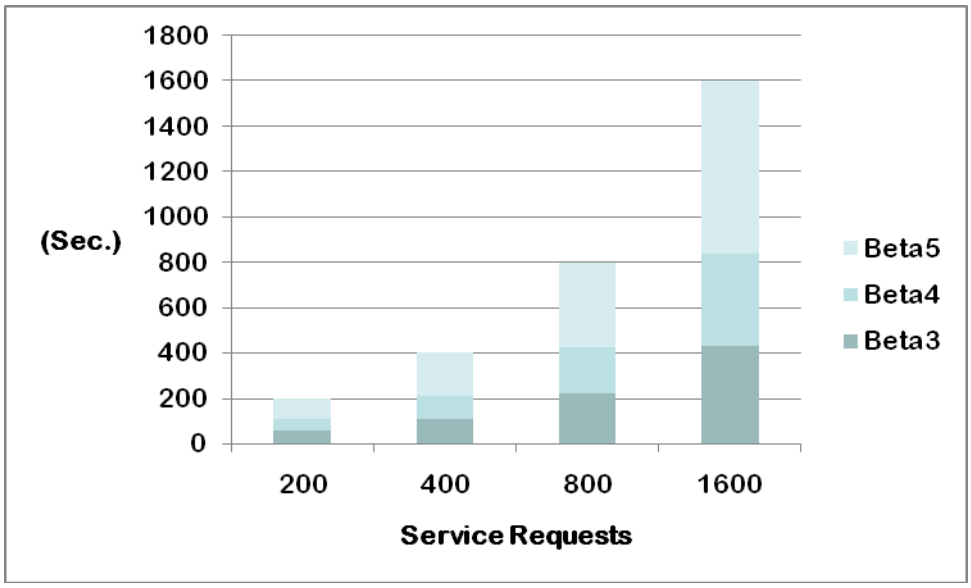

Figure 4-8 request distributed

Chapter 5

Conclusions and Future Work

We create a portal, user can use this portal to access grid's resource or Hadoop resource, and also a service workflow which combines service to complete user's work, in the future our portal will support semantic analysis to make workflow better. The paper provided an analysis of use of the SOA paradigm in the context of an e-health Web-based system.

We design and implementation a SOA web portal. The web portal provides the services that combine resource broker. The resource broker enables users to submit job, monitoring grid nodes and viewing patient records, etc. via a SOA web portal. So, user will be able to take care of their own will be more convenient.

Our future work will focus on handset device API and the service from hospital. We hope the patient records can follow user all the time. Whenever the user got sick, they can be anywhere for medical treatment. If the clinical data is authorization, we can use these data to statistical to predict the trend of infection disease.

In the future will developing browser based computing, each user can share their resource use their browser, maybe in the future flash will support gpu hardware s acceleration that can be used for gpu computing or use web GL to do this. Web browser computing can run in different device.

About the computing in different platform, our portal can access service in different platform, in the future we will developing the module separate data and computing service by type or function, first need to defined data format and different data converting service, computing part needs to developing a lot of math function or science function, and use our portal to combine these, finally we have a general

computing platform cross platform and has a quality of service.

Bibliography

[1] Yang, Chao-Tung, and Sung-Yi Chen. "A Multi-Site Resource Allocation Strategy in Computational Grids." *In Advances in Grid and Pervasive Computing, Lecture Notes in Computer Science,* Volume 5036, pp. 199-210, 2008.

[2] Firat Kart, Gengxin Miao, L.E. Moser, P.M.Melliar-Smith. "A Distributed e-Healthcare System Based on the Service Oriented Architecture." *IEEE International Conference on Services Computing* , pp.652-659, 2007.

[3] Seog-Chan Oh, Dongwon Lee, Soundar R.T. Kumara. "Effective Web Service Composition in Diverse and Large-Scale Service Networks." *IEEE Transactions on Service Computing January-March*, Volume 1, pp. 15-32, 2008.

[4] Hong, Shang-Pei. "An exchange system for medical information based on hl7/cda."

[5] P. Mazzoleni, B.Crispo, S.Sivasubramanian, E. Bertino. "Efficient integration of fine-grained access control and resource brokering in grid." *The Journal of Supercomputing archive*, Volume 49, pp. 108-126, 2009.

[6] Yang, Chao-Tung, Sung-Yi Chen, and Tsui-Ting Chen. "A Grid Resource Broker with Network Bandwidth-Aware Job Scheduling for Computational Grids." *In Advances in Grid and Pervasive Computing*, Volume 4459, pp. 1-12, 2007.

[7] Yang, Chao-Tung, Po-Chi Shih, Sung-Yi Chen, and Wen-Chung Shih. "*An Efficient Network Information Model Using NWS for Grid Computing Environments." In Grid and Cooperative Computing - GCC 2005,* pp. 287-99, 2005.

[8] Yang, Chao-Tung, Po-Chi Shih, Cheng-Fang Lin, and Sung-Yi Chen. "*A Resource Broker with an Efficient Network Information Model on Grid Environments." The Journal of Supercomputing,* Volume 40, pp. 249-67, 2007.

[9] C.T. Yang, C.F. Lin, and S.Y. Chen, “A Workflow-based Computational Resource Broker with Information Monitoring in Grids,” *in Fifth International Conference on Grid and Cooperative Computing (GCC'06),* pp. 199-206, 2006.

[10] https://wom.dev.java.net/

[11] https://xsom.dev.java.net/

[12] http://ksoap2.sourceforge.net/

[13] http://ws.apache.org/juddi/

[14] http://ws.apache.org/axis/

[15] Luciano Baresi1, Sam Guinea, Olivier Nano, and George Spanoudakis. "Comprehensive Monitoring of BPEL Processes." *IEEE Internet Computing archive,* Volume 14,pp. 50-57, May 2010.

[16] http://ode.apache.org/

[17] Jaewook Kim, Sookyoung Lee, Milton Halem and Yun Peng."Semantic Similarity Analysis of XML Schema Using Grid Computing." *Information Reuse & Integration, 2009. IRI '09. IEEE International Conference,* pp. 57-62, 2009.

[18] Ian Cooper and Coral Y. Walker. "The Design and Evaluation of MPI-Style Web Services." IEEE Transactions on Services Computing, Volume 2, pp. 197-209, 2009.

[19] Mangal Sain, Sachin Bhardwaj, HoonJae Lee and Wan-Young Chung. "Architecture of Personal Healthcare Information System in Ubiquitous Healthcare." *Communication and Networking International Conference, FGCN/ACN 2009, Held as Part of the Future Generation Information Technology Conference, FGIT 2009*, Volume 56, pp.157-164, 2009.

[20] Lenuta ALBOAIE, Sabin C. BURAGA, Victor FELEA. "TELEMON – an SOA-based e-Health System.Designing the Main Architectural Components." *waina, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops,* pp. 557-56, 2010.

[21] Jianwu Wang, Daniel Crawl, Ilkay Altintas. "Kepler + Hadoop : A General Architecture Facilitating Data-Intensive Applications in Scientific Workflow Systems." *Conference on High Performance Networking and Computing Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science, 2009.*

[22] Carolina Fortuna, Mihael Mohorcic. " Dynamic composition of services for end-to-end information transport." *IEEE Wireless Communications,* Volume 16, pp. 56-62, August 2009.

[23] Gulnoza Ziyaeva, Eunmi Choi, Dugki Min. "Content-Based Intelligent Routing and Message Processing in Enterprise Service Bus." *Convergence and Hybrid Information Technology, ICHIT '08. International Conference*, pp.245-249, 2008.

[24] http://www.w3.org/TR/wsdl

[25] http://www.oasis-open.org/committees/uddi-spec/faq.php

[26] Laura E. Grit , “Broker Architectures for Service-oriented Systems”, 2005.

[27] Dagmar Krefting, Julian Bart, Kamen Beronov, Olga Dzhimova, Jürgen Falkner, Michael Hartung, Andreas Hoheisel, Tobias A. Knochf, Thomas Lingner, Yassene Mohammed, Kathrin Peter, Erhard Rahm, Ulrich Sax, Dietmar Sommerfeld, Thomas Steinke, Thomas Tolxdorff, Michal Vossberg, Fred Viezens, Anette Weisbecker, “MediGRID: Towards a user friendly secured grid infrastructure,” *Future Generation Computer Systems* , Volume. 25, pp. 326-336, 2009.

[28] Mercedes Argüello Casteleiro, Julio Des, Maria Jesus Fernandez Prieto, Rogelio Perez, Hilary Paniagua, "Executing medical guidelines on the web: Towards next generation healthcare.", Volume 22, pp. 545-551, 2009.

[29] HL7, “Health Level 7 Version 2.3.1 Final Standard”, 1999.

[30] Bojan Blazona, Miroslav Koncar, “HL7 and DICOM based integration of radiology departments with healthcare enterprise information systems,” *International Journal of Medical Informatics*, Volume. 76, pp. 425-432, 2007.

[31] http://code.google.com/intl/zh-TW/apis/igoogle/docs/users.html

[32] Andrea Bosin, Nicoletta Dess`ı, Barbara Pes, "Extending SOA paradigm to E-Science environments." *Future Generation Computer Systems (2010), doi:10.1016/j.future.2010.07.003*