SOVoIP Prototype System

User Manual

Author: Jin Li

Contact: auswer@hotmail.com

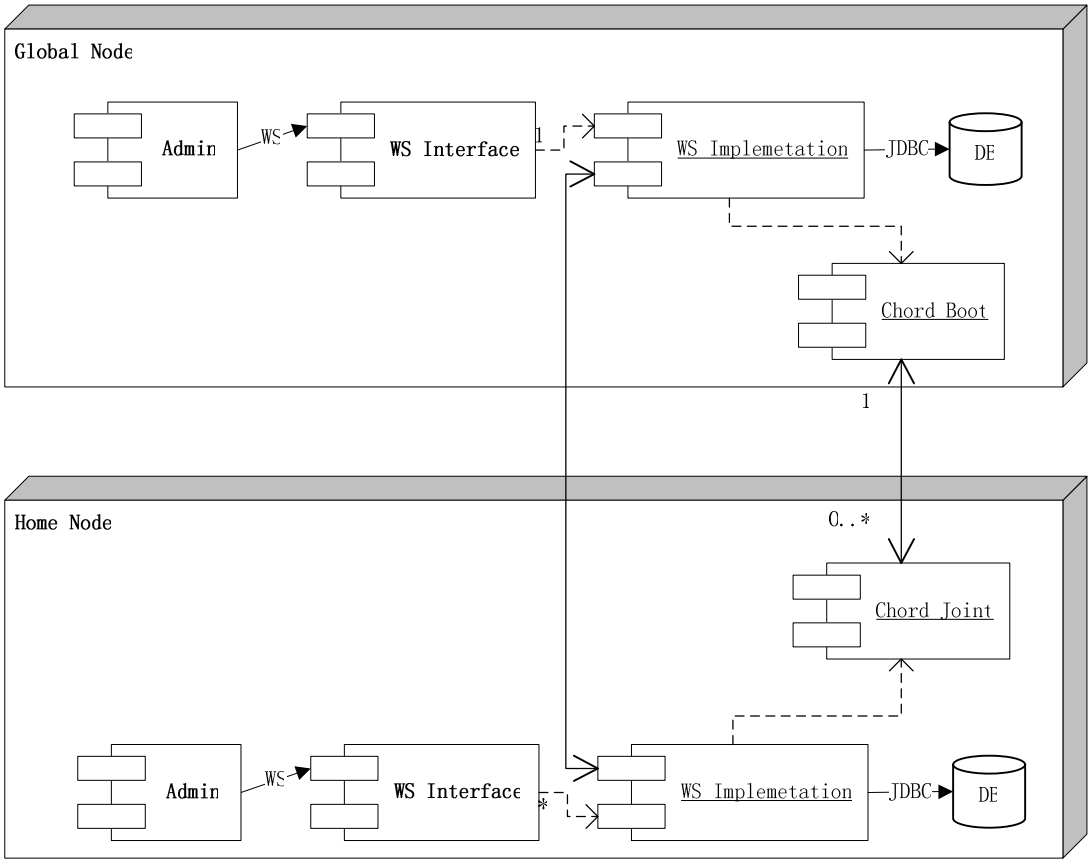
Table of Contents

1 Introduction

The SOVoIP (Service-Oriented Voice over IP) project is the demonstration for the working scheme of a prototype system designed by Dr. Arif Jubaer, aiming to provide a light weight, robust solution to VoIP. This project is a part of the Sun Academic Initiative program, sponsored by University of Melbourne and Sun Microsystem.

To design and implement this P2P-based servers for a service-oriented VoIP software, **Java**, **MySQL server** and **Glassfish server** are used to realize the features in the original proposal. The development environment is based on **NetBeans IDE.**

1.1 System Overview



Deployment diagram, showing all the components in this system

As illustrated in the deployment diagram, the system has two kinds of servers: Global Public Nodes and normal Public Nodes. Global Public Nodes are referred as **Global Nodes** in the future. Since the normal Public Nodes are all used as “Home Node” which maintains information of users assigned to it, we refer to this kind of servers as **Home Node**. It is necessary that the nearest nodes in the prototype are all avatars of home nodes, even if they can be just a “passing-through” server. The deployment are different for these two kinds of servers, for each of them, a dedicated war file containing an

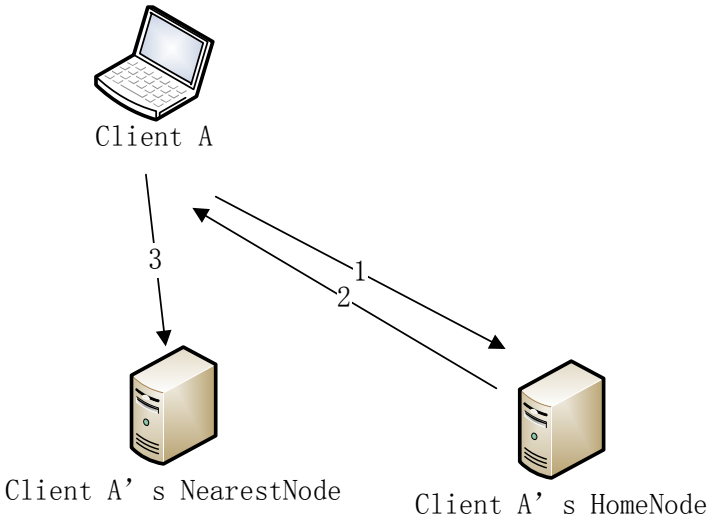
administration panel will need to be deployed separately on the machine on which that type of Node is deployed. We also define the Home Node which is nearest to a client as this client's **Nearest Node.**

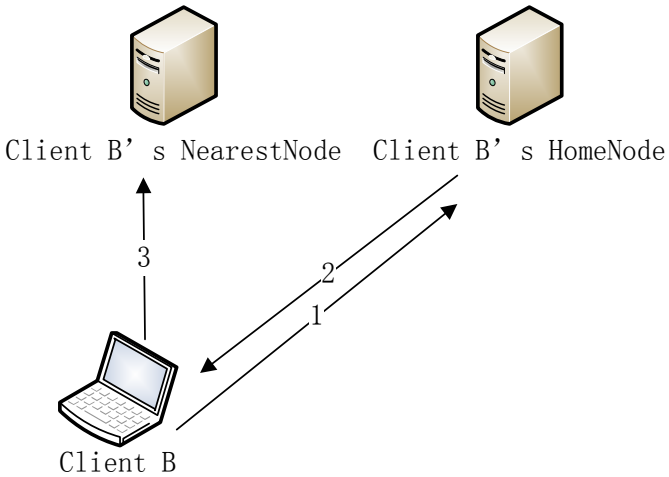
The behaviors of inter-servers and server-client communication are helpful for understanding the working scheme of the entire system. For further information, please refer to next section "1.2 Online support and resources".

1.1.1 Inter-servers Communication

Take two typical workflows in the system as examples.

A. Workflow of Authentication and Registration

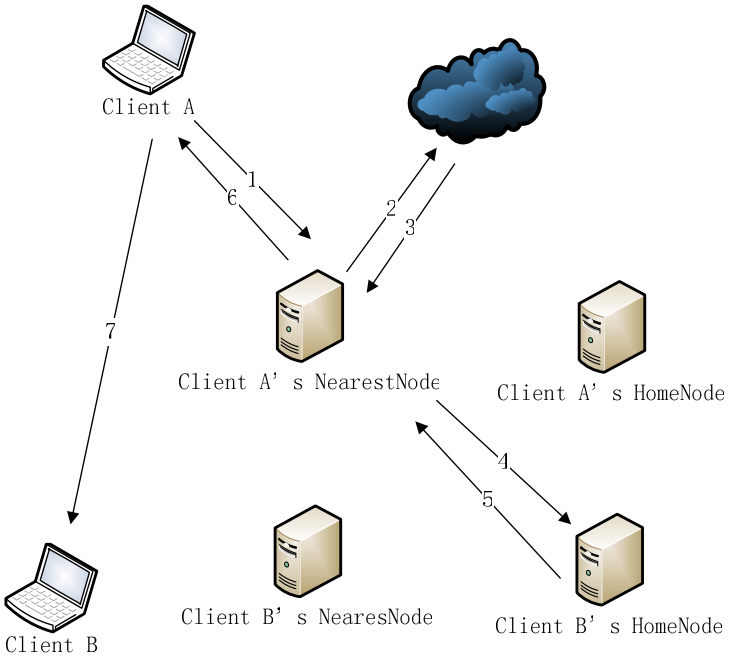




1. ClinetA logs into A's HomeNode, same to B;
2. A's HomeNode assign a Node to A as its NearestNode, same to B;
3. Finally, ClientA's NearestNode takes care of A, same to B.

(NOTE: in this scenario, Home Nodes maintain the registration information of corresponding Clients in their databases, not the NearestNodes.)

B.Workflow of UserSearch

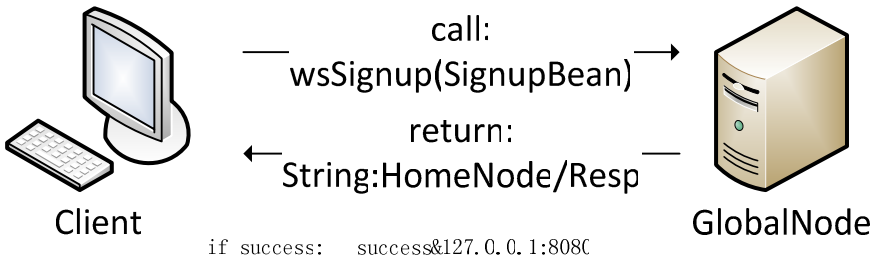


1. ClinetA uses "UserSearch()" to ask for ClientB's IP and port, via ClientA's NearestNode;
2. ClientA's NearestNode uses Chord to retrieve ClientB's HomeNode;
3. Chord returns ClientB's HomeNode;
4. A's NearestNode asks B's NearestNode for B's info (Web Service to Web Service);
5. B's NearestNode returns B's IP and port to A's NearestNode;
6. A's NearestNode return B's IP and port to A
7. A starts to contact B.

1.1.2 Client-Server Communication

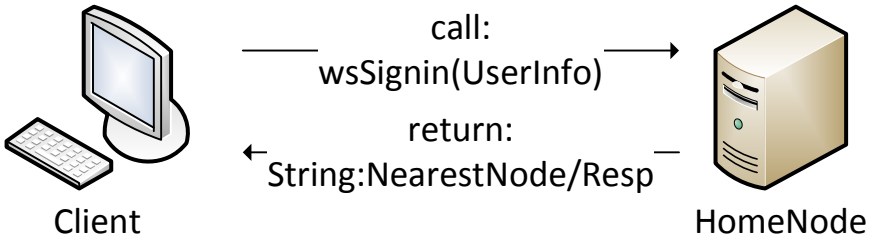
There are 8 types of interactions between a client and the server side:

1. The Process of Sign-Up



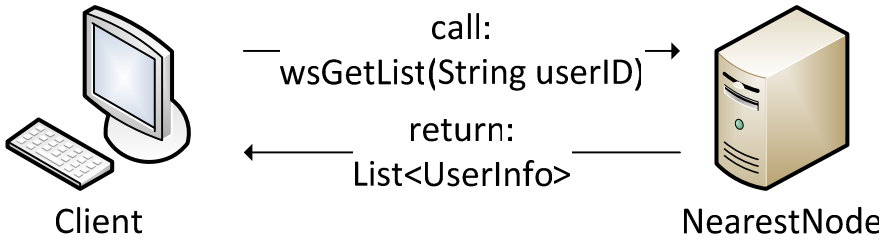
if success: success&127.0.0.1:8080
if fail: fail&message_for_you_to_print_out

2. The Process of Sign-In



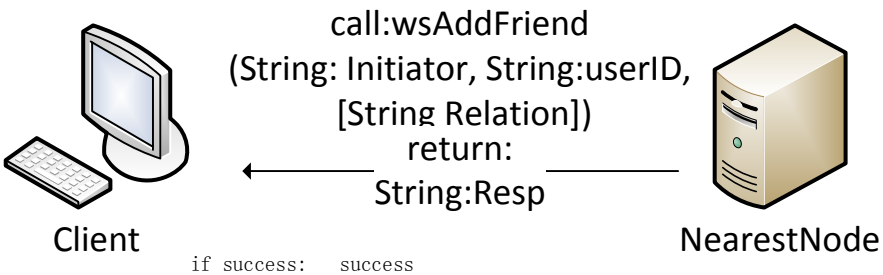

```
if success:    success&127.0.0.1:8080
if fail:       fail&message_for_you_to_print_out
```

3. The Process of Retrieving Friend List



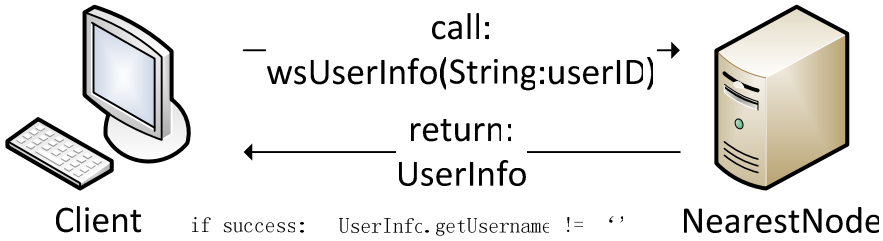

```
if success: List.getElement(0).getUserID != ‘’
if fail:    List.getElement(0).getUserID = ‘’
```

4. The Process of Adding a Friend



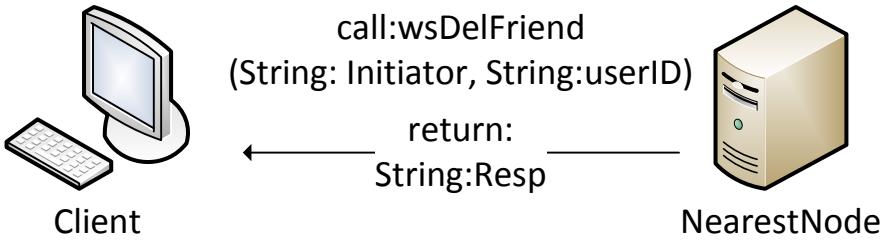

```
if success:    success
if fail:       fail&message_for_you_to_print_out
```

5. The Process of User Search



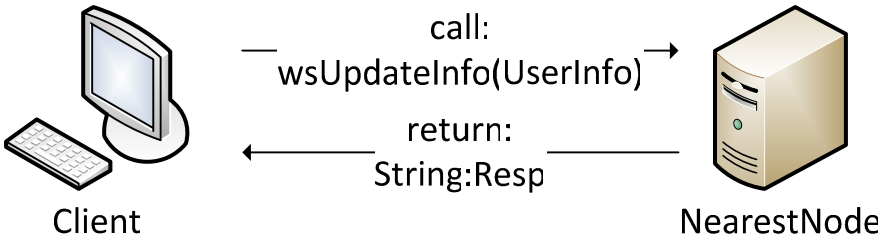

```
if success:    UserInfo.getUsername != ‘’
if fail:
 UserInfo.getUsername = ‘’ (empty string)
```

6. The Process of Deleting a Friend




```
if success:   success
if fail:      fail&message_for_you_to_print_out
```

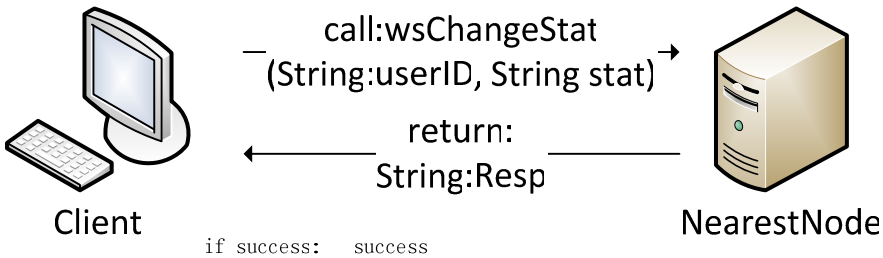
7. The Process of Updateing Personal Information




```
if success:   success
if fail:      fail&message_for_you_to_print_out
```

8. The Process of Changing Status

```
status:
online
offline
busy
```




```
if success:   success
if fail:      fail&message_for_you_to_print_out
```

1.2 Online support and resources

There are several ways for obtaining the source codes and detailed documents:

1.2.1 Original Papers

The papers about the SOVoIP can be found at:
http://p2p.cs.mu.oz.au/software/SOVoIP/

1.2.2 Hosted Project

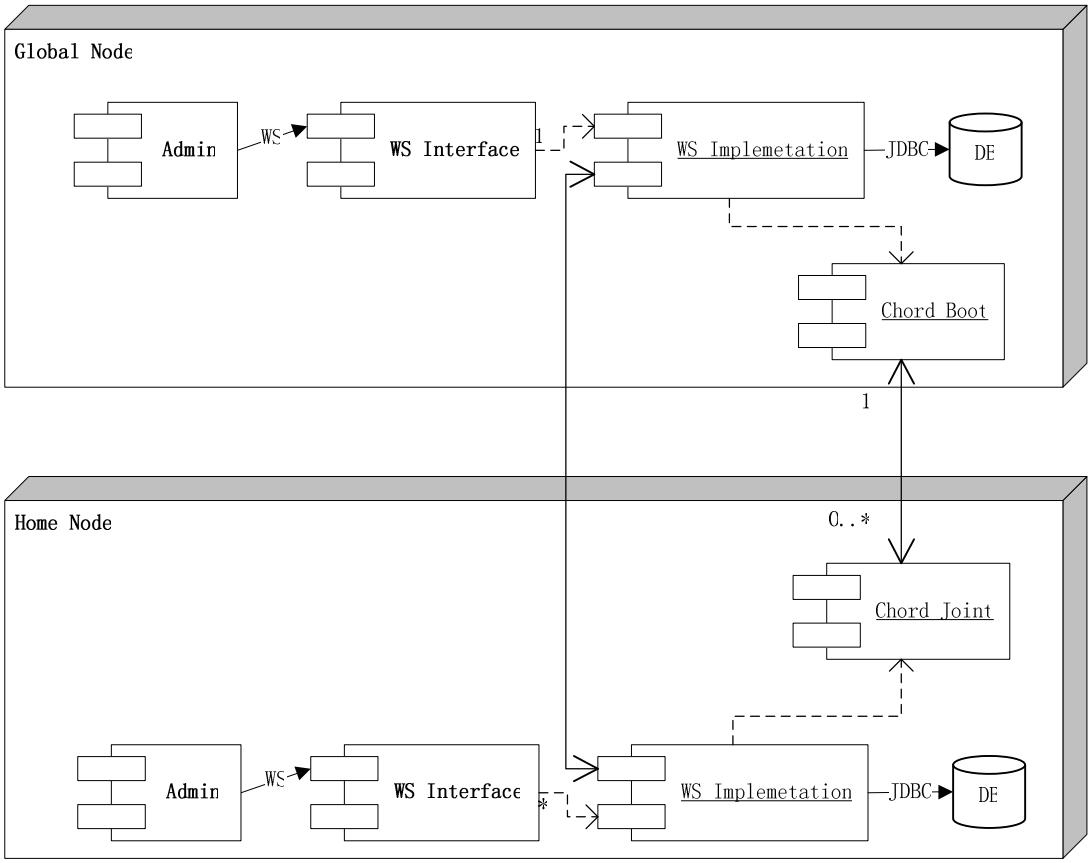
The project is hosted on Sun's Kenai website:
http://kenai.com/projects/sovoip/

which includes the Issue tracking, wiki and mail list.

1.2.3 SVN

All the source codes and database scripts involved in this manual can be found in the SVN.
Subversion URI is:
https://svn.kenai.com/svn/sovoip~sovoiprototype

2 Deployment Guide (Single machine)



Deployment diagram, showing all the components in this system

As illustrated in the deployment diagram, the system has two kinds of servers: Global Public Nodes and normal Public Nodes. Global Public Nodes are referred as **Global Nodes** in the future. Since the normal Public Nodes are all used as “Home Node” which maintains information of users assigned to it, we refer to this kind of servers as **Home Node**. It is necessary that the nearest nodes in the prototype are all avatars of home nodes, even if they can be just a “passing-through” server. The deployment are different for these two kinds of servers, for each of them, a dedicated war file containing an administration panel will need to be deployed separately on the machine on which that type of Node is deployed.

The procedure of deploying the system involves preparation, distribution and configuration.

2.1 Deployment: Preparation

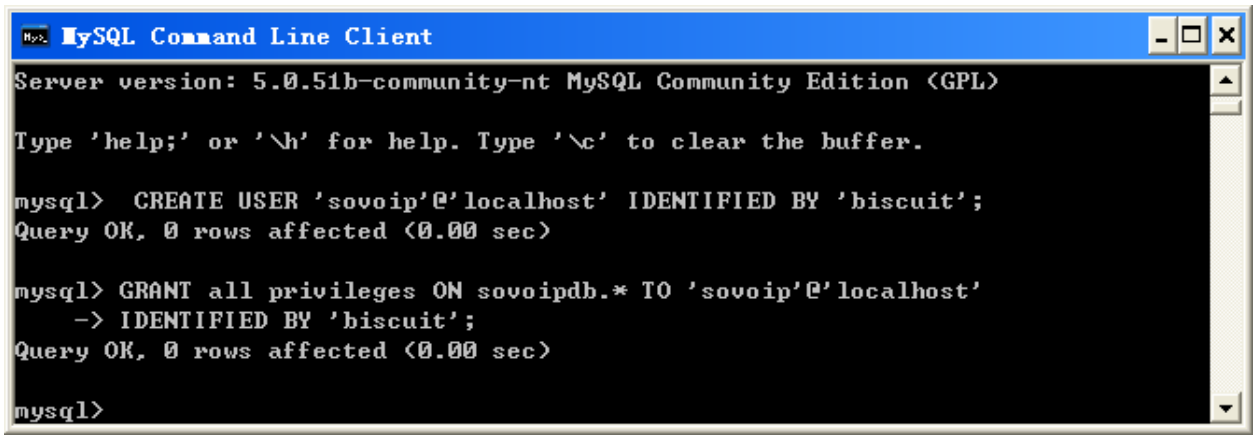
Now we have one machine working as both Global Node and Home node, with IP **192.168.1.101** .

The softwares we install on both machines are:

1. **MySQL Server 5.0** (as DB server);
2. **PHPMyAdmin 2.11.7** (as DB setting tool);
3. **Sun GlassFish Server v2.1** (as application server);
4. **Mozilla Firefox 3.0.14** (as Explorer for running admin panel);
5. **WinRAR** (to manip)

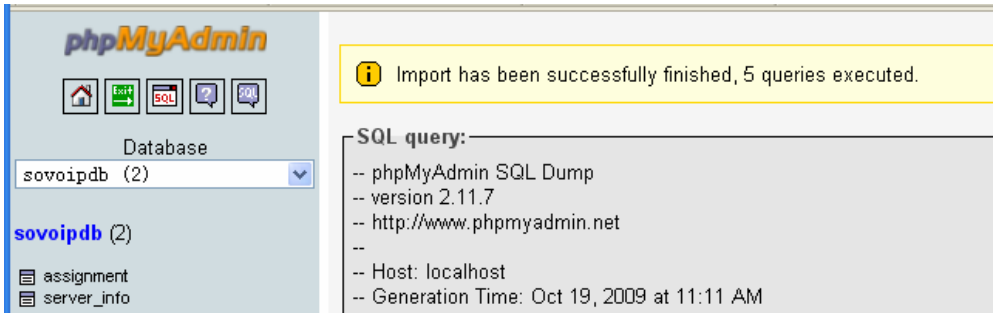
2.1.1 Setup database

On both machines, in the MySQL console, create a user for the database:

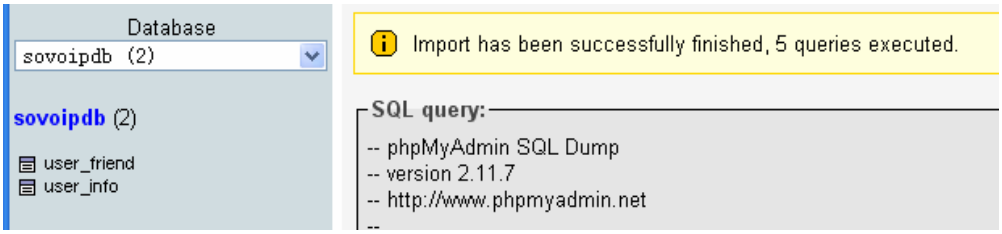


mysql > CREATE USER 'sovoip'@'localhost' IDENTIFIED BY 'biscuit';
mysql > GRANT all privileges ON sovoipdb. TO 'sovoip'@'localhost'*
> IDENTIFIED BY 'biscuit';

In the PHPMyAdmin panel(or you can use MySQL console itself) of both machines, create a database called "sovoipdb". Firstly for Global Node functions, run the script ".db/global_node_structure.sql" under the database "sovoipdb"; secondly, for Home Node functions, run the script ".db/ public_node_structure.sql" under the database "sovoipdb". After that, all the table structures are created.



After creating tables of the first step



After creating tables of the second step

The last thing to do is to check the port for MySQL server. (**By default it is 3306**)

2.1.2 Manipulate war file

If you follow the previous step and the default MySQL port is not changed, the war files provided in the "deployment" folder do not need to be changed, you can jump over this step to distribution. If not, you need to modify the properties files in the war file before you head to the next step.

Use WinRAR to open the **GlobalNode.war** which is under the "deployment" folder. Extract the file *GlobalNode.war\WEB-INF\classes\utility\config.properties* and edit it. There some fields you might need to change:

Fields		Explanations
Key	**Default Value**	**(Conditions)**
username	sovoip	If you have another username
password	biscuit	If you have another password
url	jdbc:mysql://localhost:3306/sovoipdb	if the JDBC URI is changed
INITIAL_PORT	9820	if the system can't use it as chord port

After modification, you can replace the properties file in the war file.

The same to **HomeNode.war,** which is under the "deployment" folder. Extract the file *HomeNode.war\WEB-INF\classes\utility\config.propertie* and edit it. There some fields you might need to change:

Fields		Explanations
Key	**Default Value**	**(Conditions)**
username	sovoip	If you have another username
password	biscuit	If you have another password
url	jdbc:mysql://localhost:3306/sovoipdb	if the JDBC URI is changed
INITIAL_PORT	9900	If can't use it as chord port

After modification, you can replace the properties file in the war file.

2.2 Deployment: Distribution

Manually deploying the war files on GlassFish server:
The **GlobalNode.war** and **GlobalNodeAdmin.war** ;
The **HomeNode.war** and **HomeNodeAdmin.war** ;

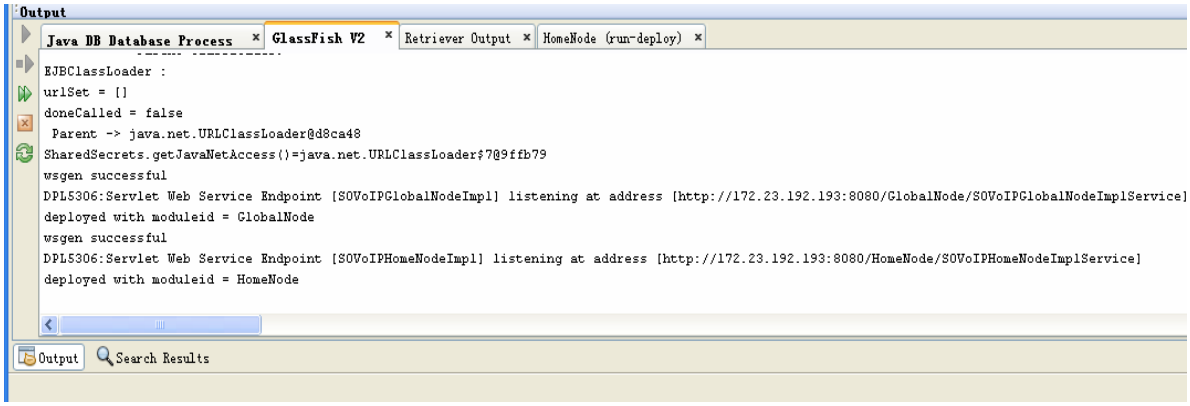
2.2.1 Deploy in GlassFish

After the deployment of the servers

In GlassFish server's console (*http://localhost:4848/*), go to Applications->Web Applications, press "deploy" button, then choose the war file to deploy.

2.2.2 Diagnose

The server's console will give helpful information to diagnose the deployment.

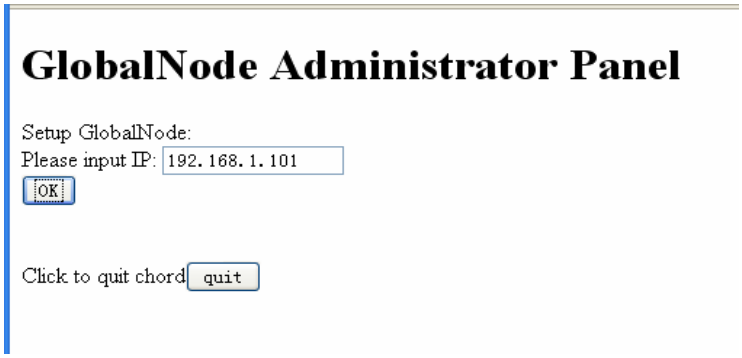


After the deployment of the servers

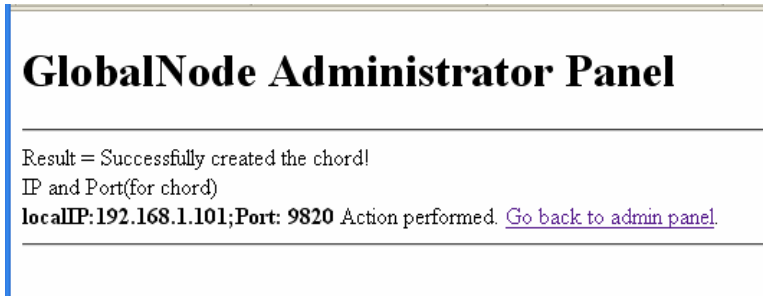
2.3 Deployment: Configuration

Now the servers are deployed, they need to be configured for initialization and connection. The outlooks of the admin panels are tatty; you may need to bear with them.

2.3.1 Configure The Global Node



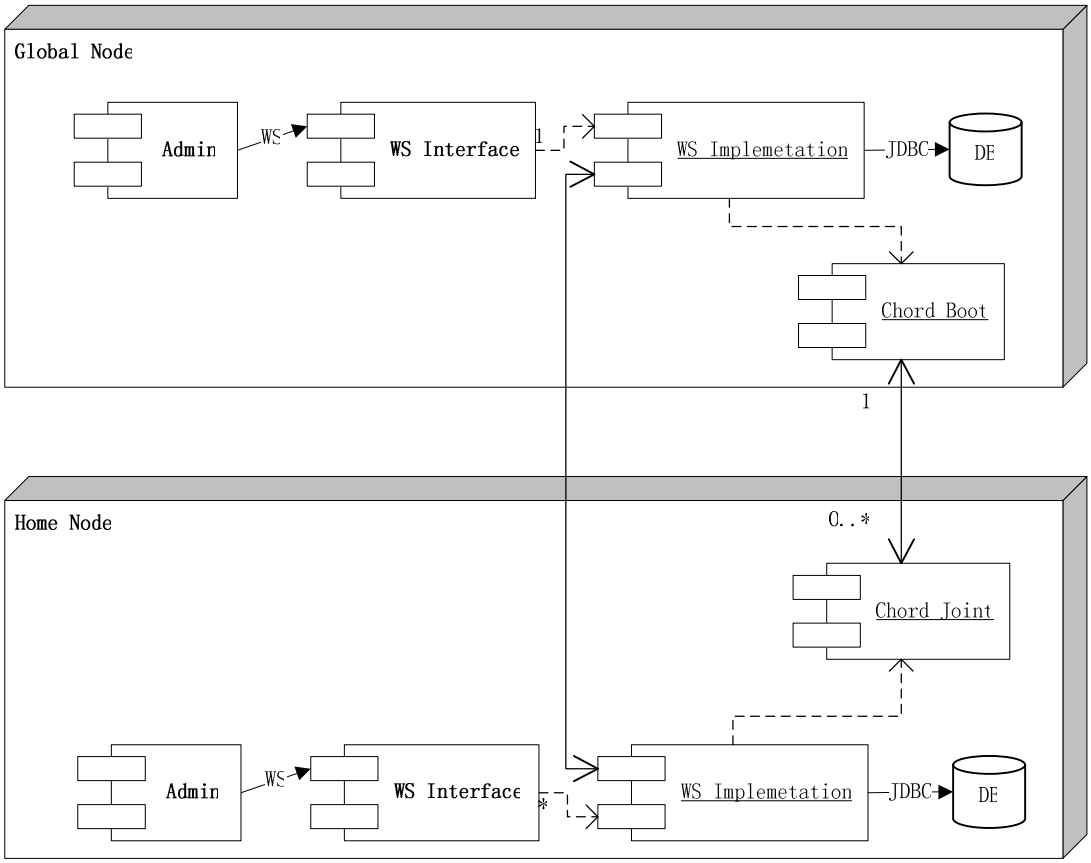
To configure the Global Node, after knowing the IP address exposed to the outside world, in this case it is **192.168.1.101**, go to (*http://localhost:8080/GlobalNodeAdmin/index.jsp*) GlobalNode Administrator Panel. Type in **192.168.1.101,** then press OK. Then the Global Node is started. The port number should be remembered for the next step.



2.3.2 Configure The Home Node

On machine #2, after knowing the IP address exposed to the outside world, in this case it is **192.168.1.101**, go to (*http://localhost:8080/HomeNodeAdmin/index.jsp*) Home Node Administrator Panel. Type in **192.168.1.101** as Global Node's IP and **8080** as Global Node's port for web service**,** so that this Home Node knows where to find the Global Node; then type in **192.168.1.101** as this Home Node's IP and **8080** as port, the Global Node knows where to find this Home Node; The type in the chord port **9820** for Global Node that we obtained from last step; then press OK. Please note that you need to consider the IP and port for Web Service in your case, such as city name.

3 Deployment Guide (Multiple machines)



Deployment diagram, showing all the components in this system

As illustrated in the deployment diagram, the system has two kinds of servers: Global Public Nodes and normal Public Nodes. Global Public Nodes are referred as **Global Nodes** in the future. Since the normal Public Nodes are all used as "Home Node" which maintains information of users assigned to it, we refer to this kind of servers as **Home Node**. It is necessary that the nearest nodes in the prototype are all avatars of home nodes, even if they can be just a "passing-through" server. The deployment are different for these two kinds of servers, for each of them, a dedicated war file containing an administration panel will need to be deployed separately on the machine on which that type of Node is deployed.

The procedure of deploying the system involves preparation, distribution and configuration.

3.1 Deployment: Preparation

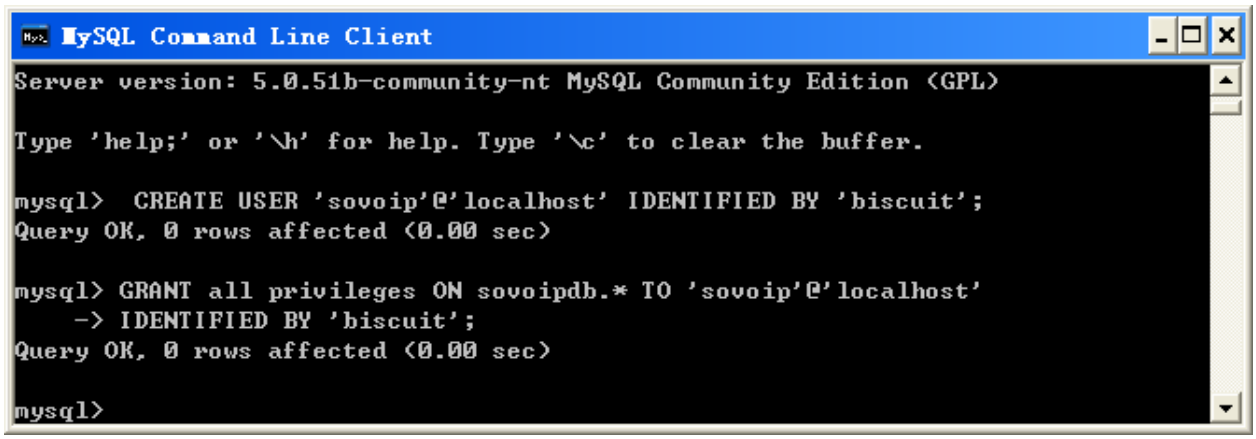
Now we have two machines connected in LAN. We will use the machine #1 with IP **192.168.1.101** as the Global Node, and machine #2, **192.168.1.102,** as a Home Node. (Please make sure all the firewalls are closed)

The softwares we install on both machines are:

6. **MySQL Server 5.0** (as DB server);
7. **PHPMyAdmin 2.11.7** (as DB setting tool);
8. **Sun GlassFish Server v2.1** (as application server);
9. **Mozilla Firefox 3.0.14** (as Explorer for running admin panel);
10. **WinRAR** (to manip)

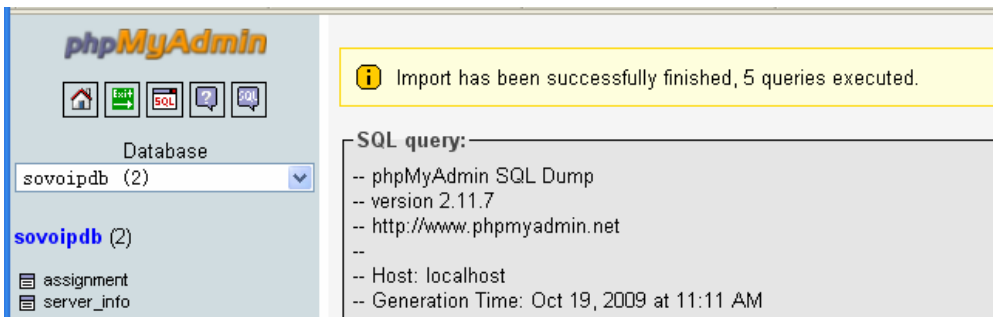
3.1.1 Setup database

On both machines, in the MySQL console, create a user for the database:

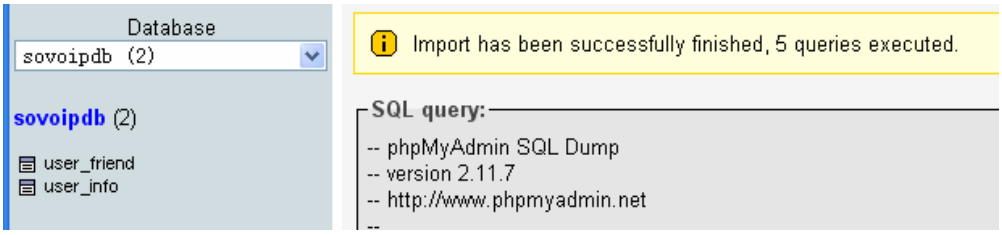


mysql > CREATE USER 'sovoip'@'localhost' IDENTIFIED BY 'biscuit';
mysql > GRANT all privileges ON sovoipdb. TO 'sovoip'@'localhost'*
> IDENTIFIED BY 'biscuit';

In the PHPMyAdmin panel(or you can use MySQL console itself) of both machines, create a database called “sovoipdb”. On machine #1, run the script “.db/global_node_structure.sql” under the database “sovoipdb”; on machine #2, run the script “.db/ public_node_structure.sql” under the database “sovoipdb”. After that, all the table structures are created.



After creating tables in machine #1



After creating tables in machine #2

The last thing to do is to check the port for MySQL server. (**By default it is 3306**)

3.1.2 Manipulate war file

If you follow the previous step and the default MySQL port is not changed, the war files provided in the "deployment" folder do not need to be changed, you can jump over this step to distribution. If not, you need to modify the properties files in the war file before you head to the next step.

Use WinRAR to open the **GlobalNode.war** which is under the "deployment" folder. Extract the file *GlobalNode.war\WEB-INF\classes\utility\config.properties* and edit it. There some fields you might need to change:

Fields		Explanations
Key	**Default Value**	**(Conditions)**
username	sovoip	If you have another username
password	biscuit	If you have another password
url	jdbc:mysql://localhost:3306/sovoipdb	if the JDBC URI is changed
INITIAL_PORT	9820	if the system can't use it as chord port

After modification, you can replace the properties file in the war file.

The same to **HomeNode.war,** which is under the "deployment" folder when you are the administrator of machine #2. Extract the file *HomeNode.war\WEB-INF\classes\utility\config.propertie* and edit it. There some fields you might need to change:

Fields		Explanations
Key	**Default Value**	**(Conditions)**
username	sovoip	If you have another username
password	biscuit	If you have another password
url	jdbc:mysql://localhost:3306/sovoipdb	if the JDBC URI is changed
INITIAL_PORT	9900	If can't use it as chord port

After modification, you can replace the properties file in the war file.

3.2 Deployment: Distribution

Manually deploying the war files on GlassFish server:
The **GlobalNode.war** and **GlobalNodeAdmin.war** should be in **machine #1**.
The **HomeNode.war** and **HomeNodeAdmin.war** should be in **machine #2**.

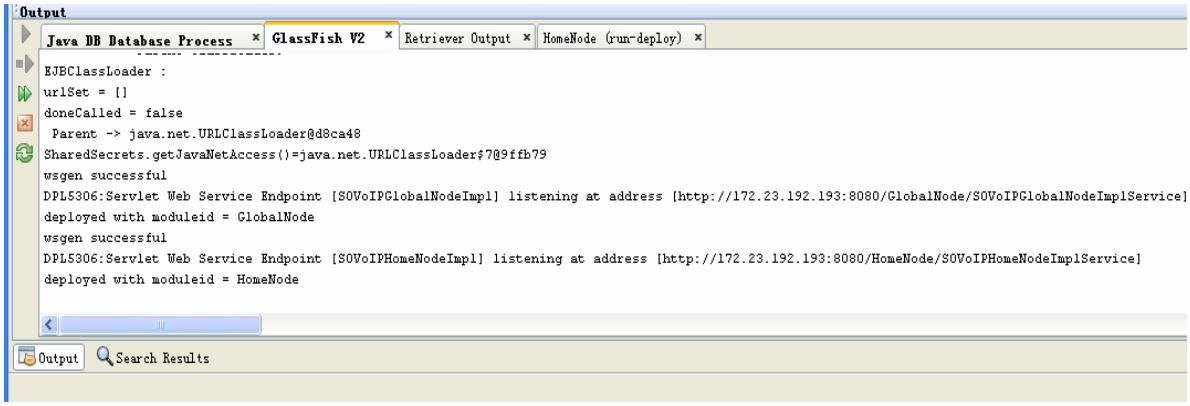
3.2.1 Deploy in GlassFish

After deployment of the servers
(using one machine, which is not the case of multiple machines).

In GlassFish server's console (*http://localhost:4848/*), go to Applications->Web Applications, press "deploy" button, then choose the war file to deploy.

3.2.2 Diagnose

The server's console will give helpful information to diagnose the deployment.

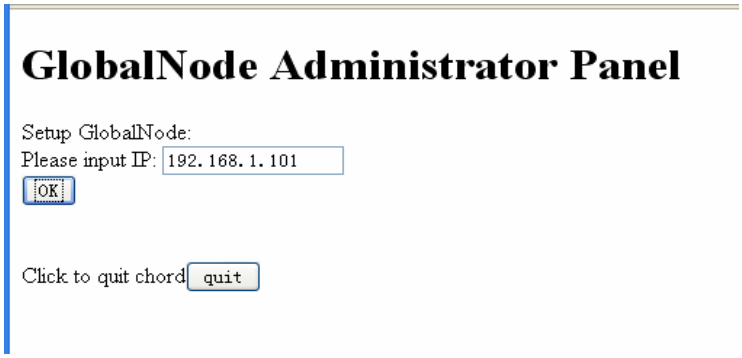


After deployment of the servers
(using one machine, which is not the case of multiple machines).

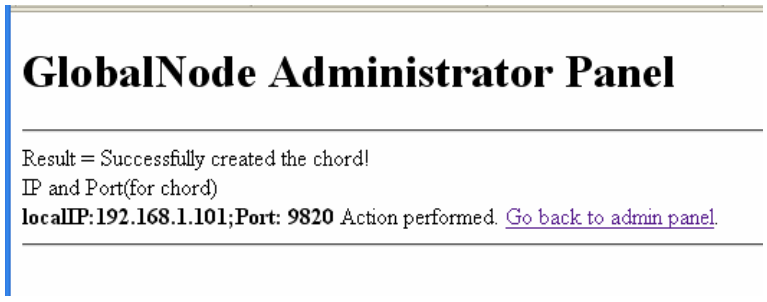
3.3 Deployment: Configuration

Now the servers are deployed, they need to be configured for initialization and connection. The outlooks of the admin panels are tatty; you may need to bear with them.

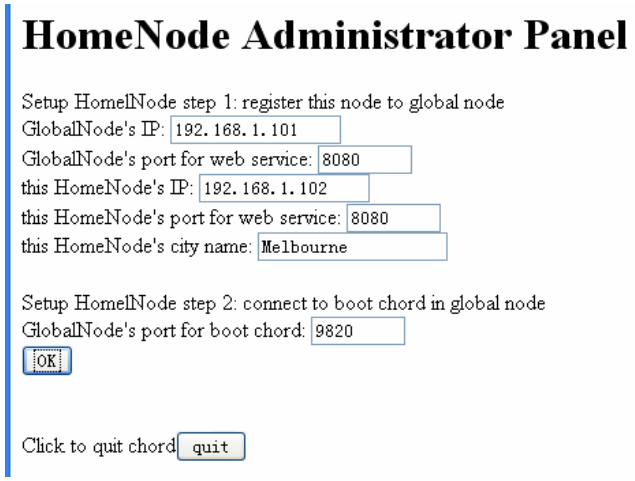
3.3.1 Configure The Global Node



On machine #1, after knowing the IP address exposed to the outside world, in this case it is **192.168.1.101**, go to (*http://localhost:8080/GlobalNodeAdmin/index.jsp*) GlobalNode Administrator Panel. Type in **192.168.1.101,** then press OK. Then the Global Node is started. The port number should be remembered for the next step.



3.3.2 Configure The Home Node



On machine #2, after knowing the IP address exposed to the outside world, in this case it is **192.168.1.102**, go to (*http://localhost:8080/HomeNodeAdmin/index.jsp*) Home Node Administrator Panel. Type in **192.168.1.101** as Global Node's IP and **8080** as Global Node's port for web service**,** so that this Home Node knows where to find the Global Node; then type in **192.168.1.102** as this Home Node's IP and **8080** as port, the Global Node knows where to find this Home Node; The type in the chord port **9820** for Global Node that we obtained from last step; then press OK. Please note that you need to consider the IP and port for Web Service in your case, such as city name.