August 2006

IBM Information Management software

High Availability with IBM DB2® 9 and xkoto GRIDSCALE

Step-by-Step Setup Guide

David Tam,
Belal Tassi

IBM Toronto Lab

1. Introduction

1.1 What is xkoto GRIDSCALE?

The xkoto GRIDSCALE Database Load Balancer is a database management solution which distributes and optimizes your database workload among a pool of available database servers.

Traditional database management high availability solutions use an Active – Passive configuration where applications only interact with the active database server. This single database server must process all requests from the application while the passive database server's sole purpose is to constantly replicate the active server to provide disaster recovery in case of a failure.

The xkoto GRIDSCALE configuration, on the other hand, uses an Active – Active configuration. By allowing all servers to contribute to read transactions, performance is increased significantly, while reliability and availability are also increased by allowing each database server to process every write transaction to maintain its own copy of the database.

Furthermore, these database servers do not have to reside in the same geographic location, they may be distributed anywhere around the world with no distance limitation. The xkoto GRIDSCALE Database Load Balancer also supports online upgrades from version to version distinguishing the xkoto GRIDSCALE Database Load Balancer from any other high available, disaster recovery solution on the market.

In this paper, you will learn how to set up and implement the xkoto GRIDSCALE high availability solution with IBM® DB2® 9 - in an easy step-by-step fashion.

1.2 Three-Tier Applications

The three-tier setup used in this paper consists of an application server, xkoto GRIDSCALE server, and a number of redundant database servers on the bottom tier. Users will interact with the application server on a day-to-day basis to perform transactions on their database. The purpose of the xkoto GRIDSCALE server is to accept these transactions created by the user and apply them to the database that is stored locally on each of the database servers.

1.3 Hardware

The xkoto GRIDSCALE setup requires a minimum of thee physical machines:

- 1 Application Server – The machine that hosts the client
- 1 xkoto GRIDSCALE Server – The xkoto load balancer
- 1 Database Server – The machine that hosts the database itself

However, with the use of three machines, there is no high availability (HA) or disaster recovery (DR) benefit. For any increase in performance and reliability, it is recommended to have at least three database servers, which will ultimately be the final configuration at the end of this paper.

This paper will walk you through step by step and explain how to set up the three different types of machines. The hardware used in this paper are all identical, specifically:

- 5 X IBM System x™ HS20 Blades, dual 64-bit Intel® processors running 64-bit SLES 9 SP3

We will be using a JDBC-based Java™ application running on our application server to generate read and write transactions, one xkoto GRIDSCALE server to manage these transactions, and three database servers, two local for HA purposes and one remote for DR, each of which contains its own full copy of the database--identical on all the database servers. Note that the remote database server does not have to be in the same geographical location as the xkoto GRIDSCALE server; this provides the most effective DR possible.

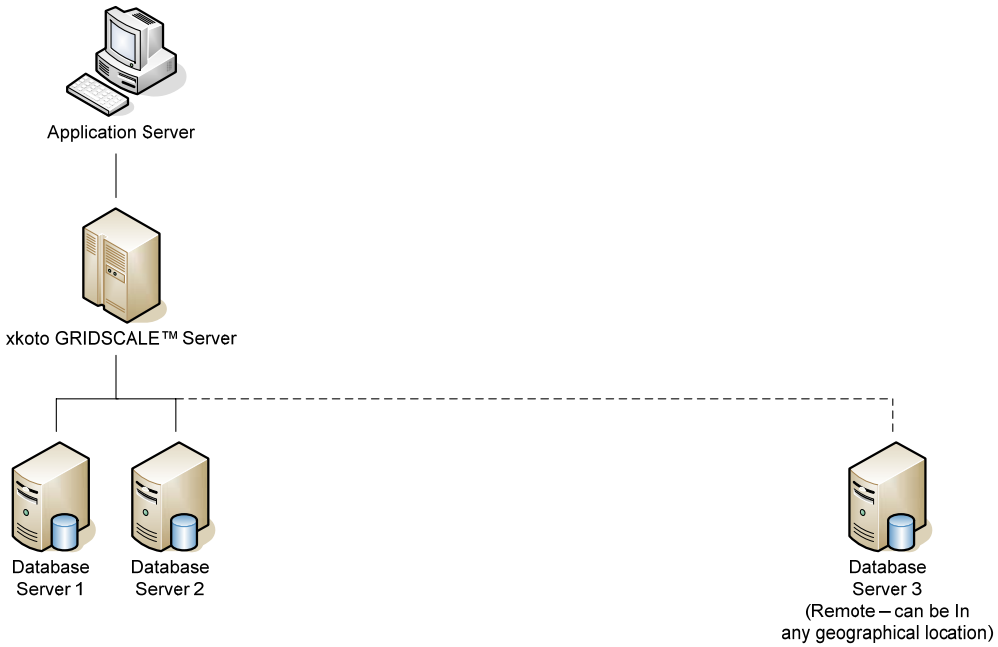


Figure 1: Hardware Architecture used in HA setup

1.4 Software

Here is a more in-depth view of how the xkoto GRIDSCALE HA setup works from a software perspective:

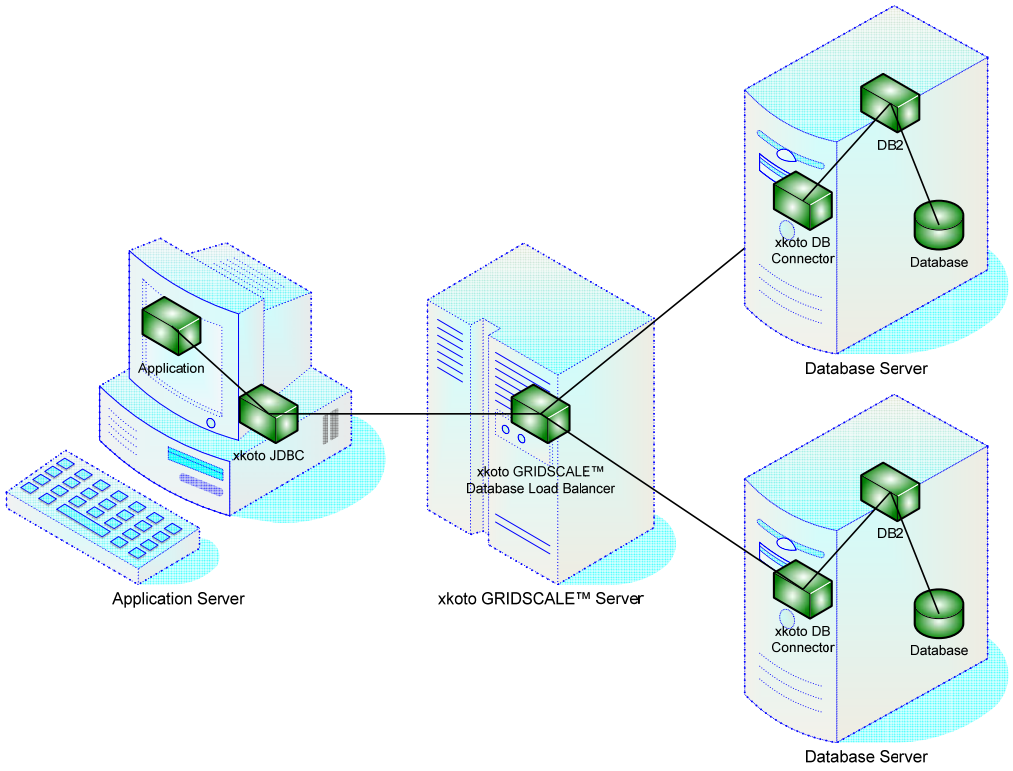


Figure 2: Software Architecture used in HA setup

For our application, we will be using the DB2 Virtual Production System (DB2VPS) to simulate a real OLTP workload against our sample database. This application will communicate using the installed xkoto JDBC driver to send the transaction requests to the xkoto GRIDSCALE Database Load Balancer. The xkoto GRIDSCALE Database Load Balancer will then distribute the transactions among all active database servers. (Only two are shown in the diagram above for simplicity.) This task is achieved by having the xkoto GRIDSCALE Database Load Balancer communicate with the xkoto GRIDSCALE DB Connector, which resides on each of the database servers.

On the database server, the xkoto GRIDSCALE DB Connector bridges the gap between the xkoto GRIDSCALE Database Load Balancer and the DB2 application. DB2 will then perform the transactions on the database that were generated by the DB2VPS application.

2. Preparing for the Installation

Your first step is to install an operating system on all five machines. As mentioned previously in our setup, we will be using SLES 9 SP3 x86 64-bit editions.

Next, write down the host names for all five machines and test the network connection to ensure that each machine can communicate with the machine that you intend to use as the xkoto GRIDSCALE server. The chart in Figure 3 below shows the host names that we will be using for our five servers. From this point forward, we will be using these host names to refer to a specific server.

	Application Server	xkoto GRIDSCALE Server	Database Server
Hostname	madrox1	madrox2	madrox3 madrox4 madrox5

Figure 3: Machine Host names

After you have all your machines and their respective networking up and running, you must now copy all of the required software onto each machine. Figure 4 below shows what software should be copied to each machine before starting the installation.

	Application Server	xkoto GRIDSCALE Server	Database Server
Required Software	JDBC-based Application xkoto GRIDSCALE	xkoto GRIDSCALE license.dat	DB2 V9.1 or DB2 Universal Database™ Version 8.2 (DB2 UDB V8.2) xkoto GRIDSCALE

Figure 4: Required Software for each Machine

We will be using DB2 V9.1 in our xkoto GRIDSCALE setup but you may use DB2 UDB V8.2 which has also been verified if you wish.

It is strongly recommended that you follow this installation order:

Section 3 – Install Application Server (madrox1)
Section 4 – Install xkoto GRIDSCALE Server (madrox2)
Section 5 – Install Database Server (madrox3)
Section 6 – Complete the xkoto GRIDSCALE Setup

Section 7 – Add Additional Database Servers
Section 8 – Perform High Availability Test

In Sections 3 – 6, we will show you how to set up the simplest xkoto GRIDSCALE configuration possible using three servers: an application server, a xkoto GRIDSCALE server, and a database server. By the end of these sections, you will have a fully functional xkoto GRIDSCALE setup.

In Section 7, we will demonstrate how to increase the performance, scalability, and reliability of your xkoto GRIDSCALE setup by adding additional database servers. This section is also intended for users who would like to start small, but still have the flexibility to upgrade their system to meet their growing business demands.

In Section 8, we will challenge xkoto GRIDSCALE by simulating a real-life disaster scenario – a full unexpected failure of one of the database servers. See how xkoto GRIDSCALE holds up and how you can recover from this sticky situation.
Now let's begin…

3. Install Application Server

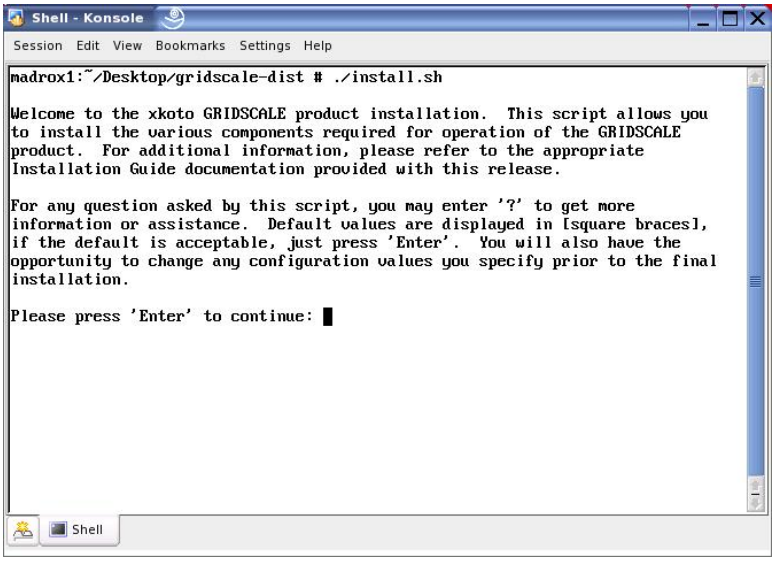
3.1 Install xkoto JDBC Driver

In order for the xkoto GRIDSCALE setup to function with our JDBC-based application, it must communicate with the xkoto GRIDSCALE server using the xkoto JDBC driver.

.

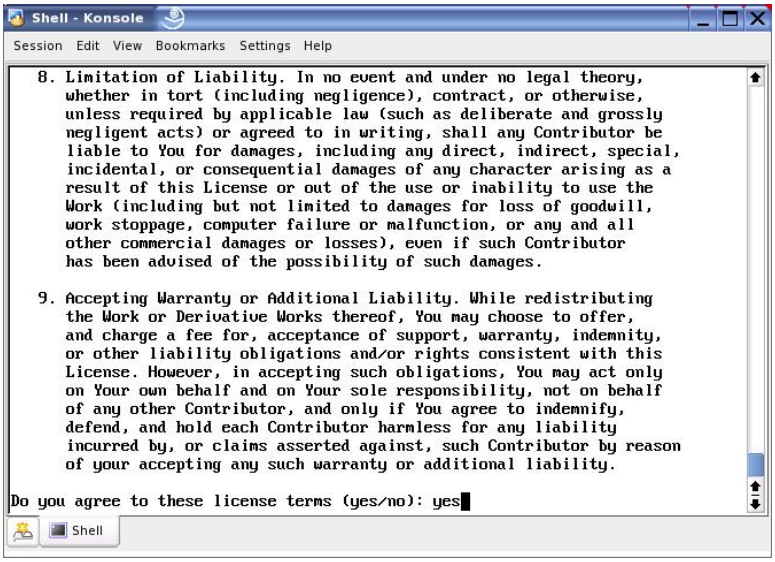
To install the xkoto JDBC driver, change your current directory to where xkoto GRIDSCALE was extracted and run the installation file by typing:

```
# ./install.sh
```

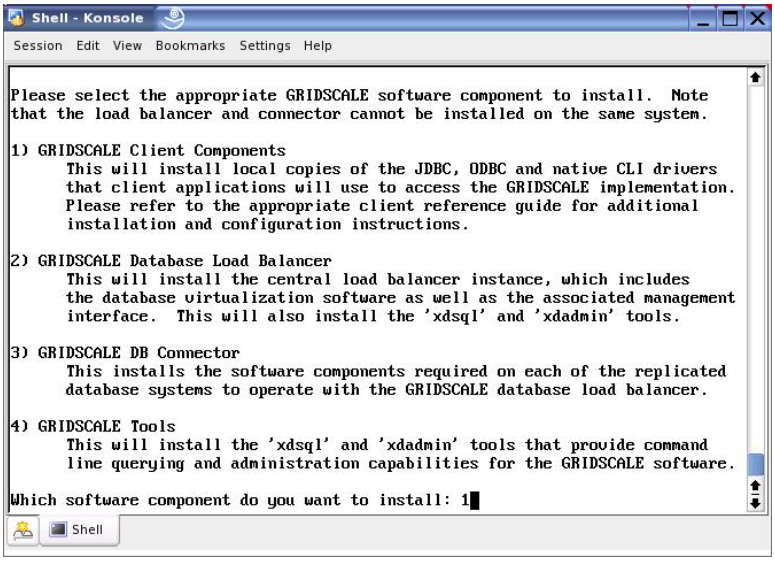


Press *Enter* to continue.

Read and agree to the license terms by typing '`yes`' and pressing *Enter*.

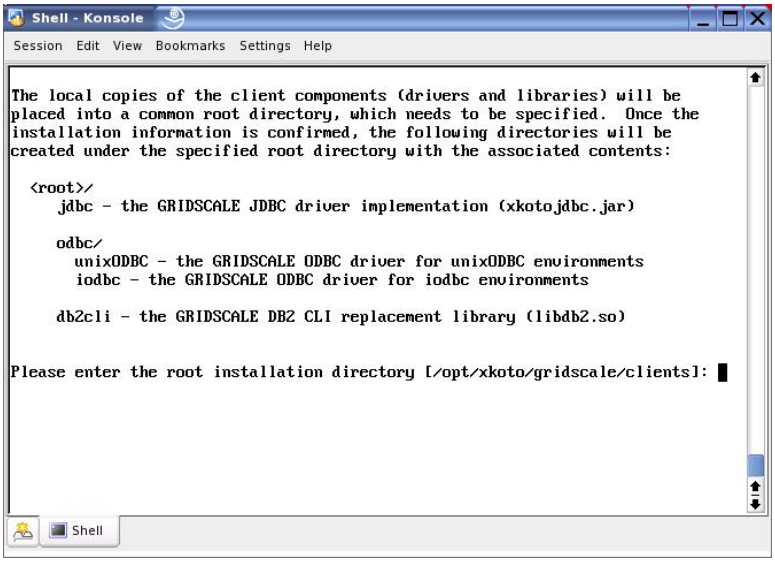


Install the xkoto JDBC Driver by typing '`1`' and pressing *Enter*.

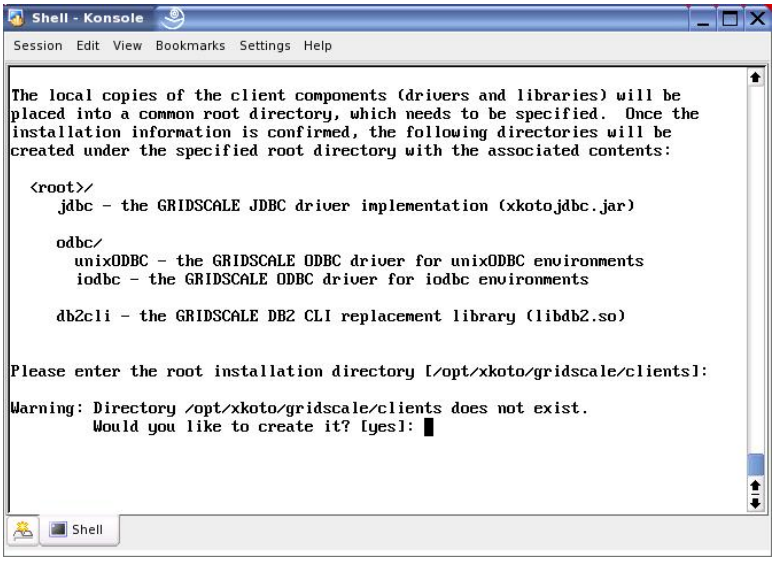


A description of the xkoto GRIDSCALE Client Components will be displayed. Notice that the xkoto GRIDSCALE is also compatible and includes drivers for ODBC- and CLI-based applications. When you have finished reading, press *Enter* to continue.

You may change the installation directory by entering a new path or keep the default path by leaving the field blank. Press *Enter.*

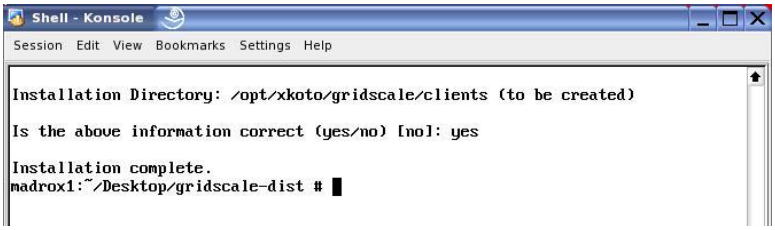


Press *Enter* once more to create the directory.



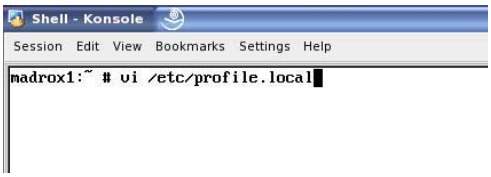
Review the installation options presented. If you are satisfied with the current settings, type '`yes`' and press *Enter* to complete your installation; otherwise, enter '`no`' to start again.

When the screen below appears, you have successfully installed the xkoto JDBC driver.



Now that the xkoto JDBC driver has been installed on the application server, you must add the location of the xkoto JDBC driver to Java's class path. Since this path needs to be set every time the application server starts, we will make this permanent by modifying the 'profile.local' file. You can do so by entering the following commands.

```
# vi /etc/profile.local
```



Enter the line '`export CLASSPATH=$CLASSPATH:/opt/xkoto/gridscale/clients/jdbc/xkotojdbc.jar`'

When you have finished editing the file, press *Esc* and then type '`:wq`' followed by *Enter* to save the file and exit.

Source your new 'profile.local' file for the class path changes to take effect by going into the /etc directory and entering the command:

```
# ./profile.local
```

3.2 Tell Your Application to Use the xkoto JDBC Driver

For JDBC applications that use DataSources and WebSphere® connection pooling, refer to the 'Configuring Your WebSphere Application for xkoto GRIDSCALE' paper and skip this section.

For applications that use a JDBC driver and create their own connection to the database, you must modify the source code in order to use the xkoto GRIDSCALE configuration. Since your application is already using some variation of the JDBC driver, modifying your application to support the xkoto JDBC should only require a few simple steps.

For us, our DB2VPS application currently supports both DB2 Type 2 and Type 4 JDBC drivers. Looking at our source code, we see the following two lines, which load the DB2 Type 2 JDBC driver:

```
url = "jdbc:db2:" + dbName;
Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
```

And the following two lines load the DB2 Type 4 JDBC driver.

```
url = "jdbc:db2://" + serverName + ":" + portNumber + "/" + dbName;
Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
```

All we need to do is modify these two lines so that it uses the xkoto JDBC driver rather than the default DB2 JDBC driver. Change these two lines in your application into the two lines shown below:

```
url = "jdbc:xkoto://" + serverName + ":" + portNumber + "/" +
dbName;
Class.forName ("xkoto.GRIDSCALE.sql.Driver").newInstance ();
```

Now that you have modified your code to support the xkoto JDBC driver, recompile your application to pick up your changes.

3.3 Install Your Application

Depending on the Java application you are using, you may need to install a Java application server such as the IBM Websphere application server. For more information, visit www.ibm.com/websphere.

Once you have installed your application, and potentially your Java application server, your client application server is complete.

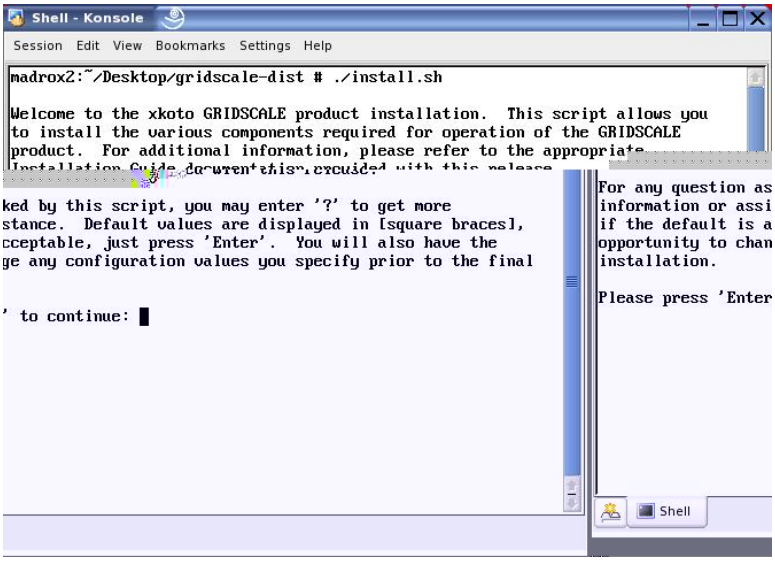
4. Install xkoto GRIDSCALE Server

4.1 Install xkoto GRIDSCALE Database Load Balancer

The xkoto GRIDSCALE Database Load Balancer's function is to distribute the workload equally among all database servers to provide a high-performance and highly available system.

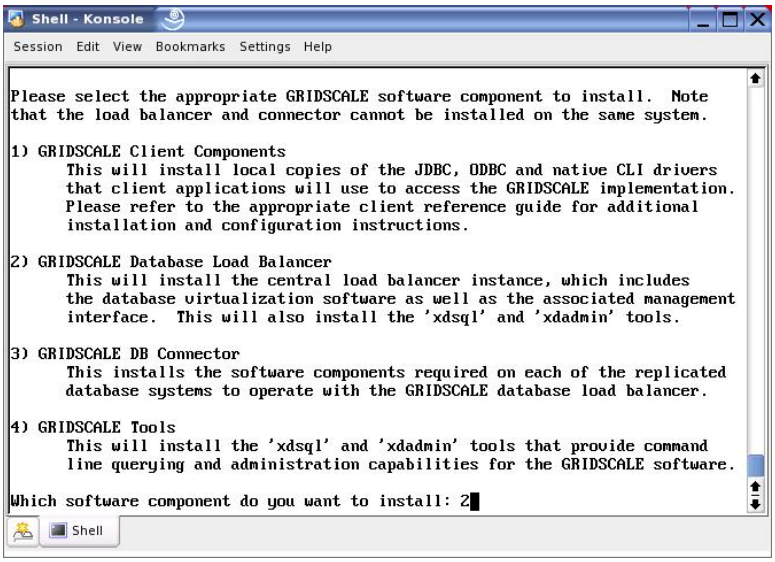
Before we begin this step, make sure that the xkoto GRIDSCALE 'license.dat' file is inside the same directory as the 'install.sh' file. To begin installation of the xkoto GRIDSCALE Database Load Balancer on the xkoto GRIDSCALE server, type:

```
# ./install.sh
```



Press *Enter* to continue. Again, read and agree to the xkoto license terms by typing '`yes`' and pressing *Enter*.

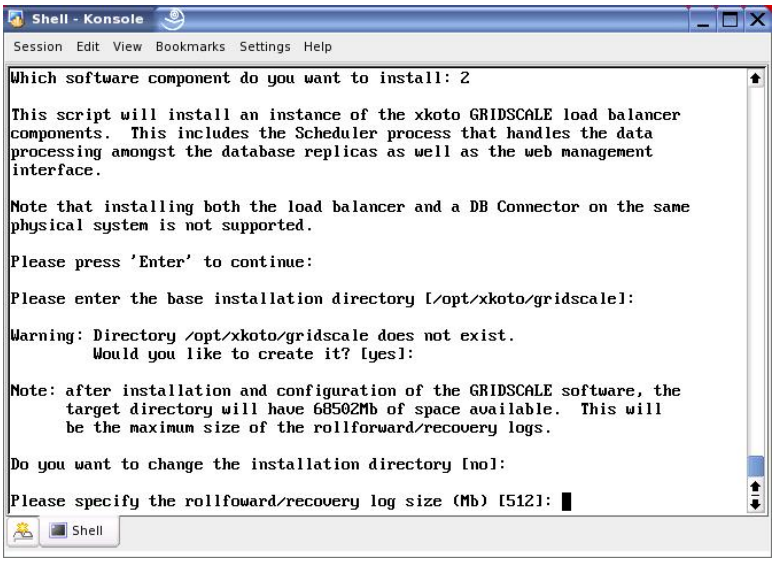
Install the xkoto GRIDSCALE Database Load Balancer by entering '`2`'.



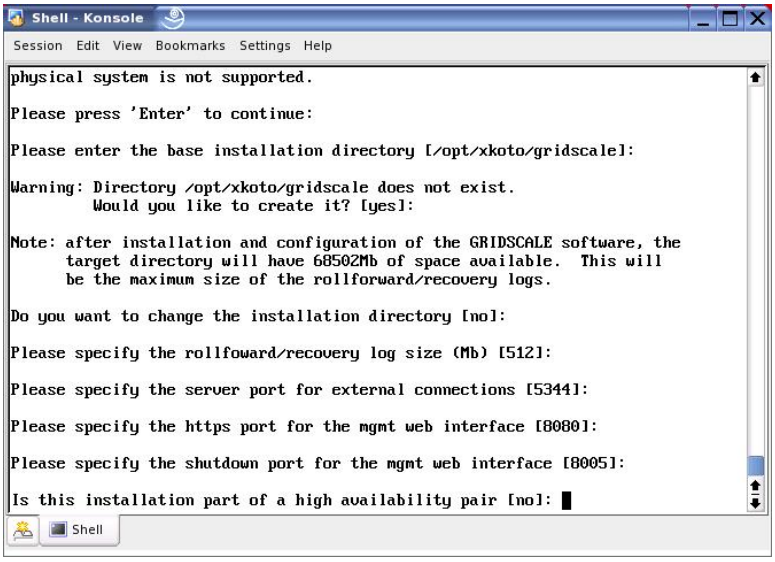
Read the xkoto GRIDSCALE Database Load Balancer description and press *Enter* again to continue.

You can manually enter an installation path or leave it blank to accept the default settings. Press *Enter* to continue and press *Enter* a second time to create the directory, and *Enter* a third time to keep the new directory.

Now press *Enter* four more times to accept the default log size of 512 MB and accept all the default port settings.

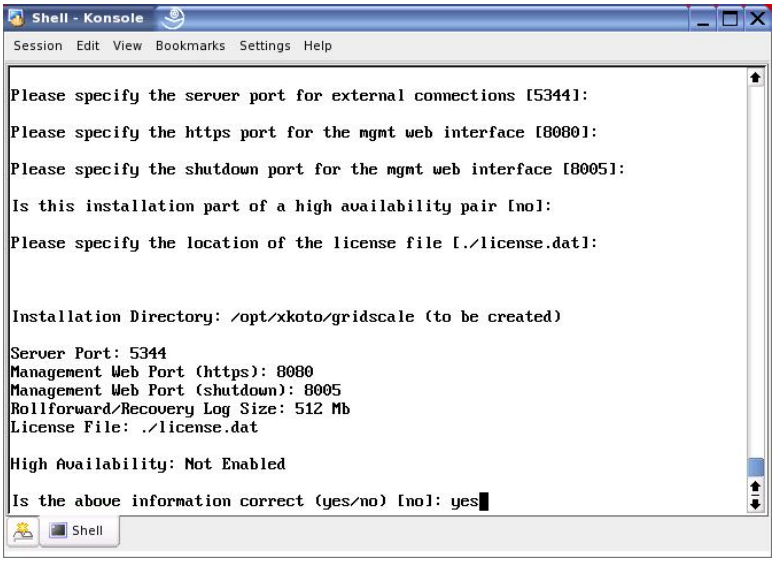


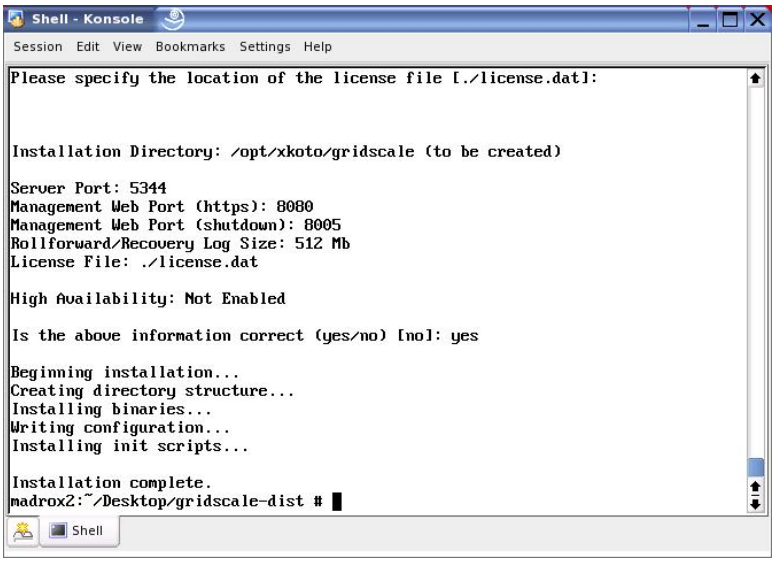
The installer now asks if this installation is part of a xkoto GRIDSCALE highly available pair. This is an advanced feature that will allow you to run multiple redundant xkoto GRIDSCALE Database Load Balancers together. For this setup, we will only be using one xkoto GRIDSCALE server so press *Enter* to select *No*.



The file 'license.dat' contains your license for the xkoto GRIDSCALE Database Load Balancer and installation will not continue without it. Since this file should already be in the same directory as the xkoto GRIDSCALE install files, press *Enter*.

Review the installation options. If you are satisfied with these settings, enter '`yes`' to proceed.





Installation of xkoto GRIDSCALE Database Load Balancer is now complete.

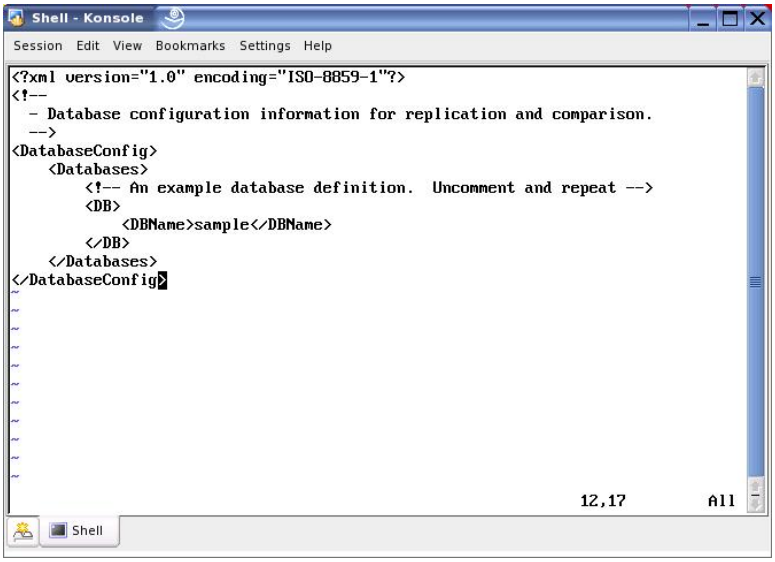
4.2 Modify databases.xml

In order for the xkoto GRIDSCALE Database Load Balancer to communicate with a specific DB2 9 database, you must modify the 'databases.xml' file. The purpose of this XML file is to keep the xkoto GRIDSCALE server informed about which databases you want it to manage. This file must be updated every time a database name is created, modified, or deleted.

If you have installed the xkoto GRIDSCALE Database Load Balancer using the default path, the 'database.xml' file will be located in '/opt/xkoto/gridscale/config'. You can use vi to edit the file:

```
# vi databases.xml
```

Modify the text so that it is identical to the sample below. In this example, our database is named 'sample'.



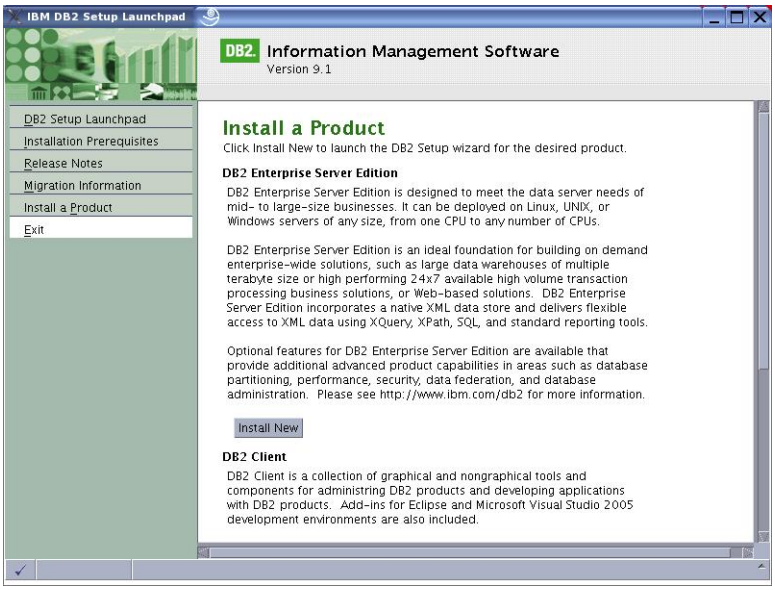
When you are finished, press Esc and type ':wq' to save and close the 'databases.xml' file.

5. Install the Database Server

Note: This section is to be preformed on only one of your database servers. The two remaining database servers will be added in Section 7.

5.1 Install IBM DB2 Enterprise Edition

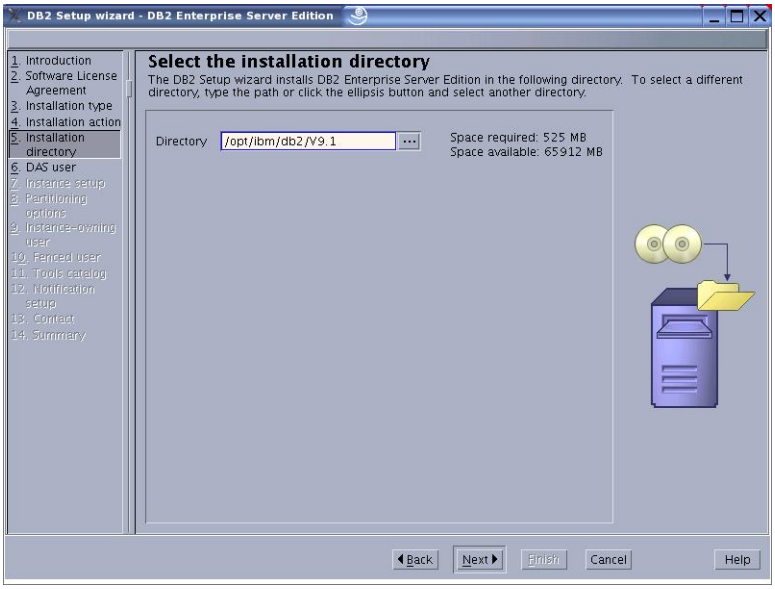
Open the folder that contains DB2 9 and click **db2setup** to begin. In the new window that opens, click **Install a Product** and then under DB2 Enterprise Server Edition click **Install New**.



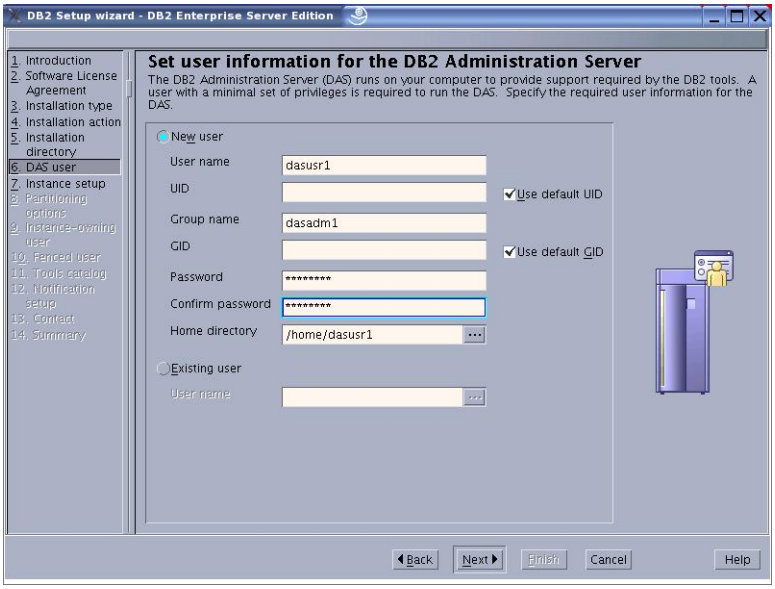
Click **Next** to continue. Read and agree to the license terms by clicking **Accept** and then **Next**.

Click **Next** to accept the Typical Install and **Next** again to install DB2 9 on the current computer.

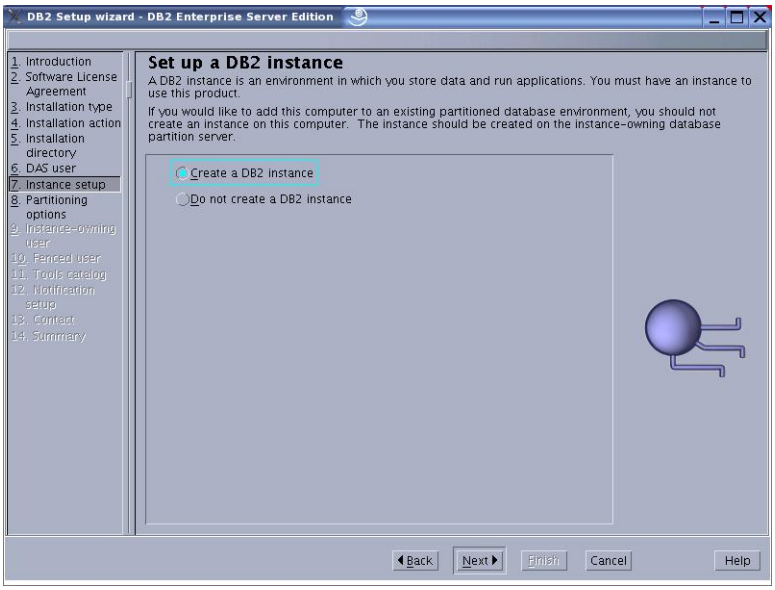
You can manually enter an installation directory or leave the default path as is. Click **Next** to continue.



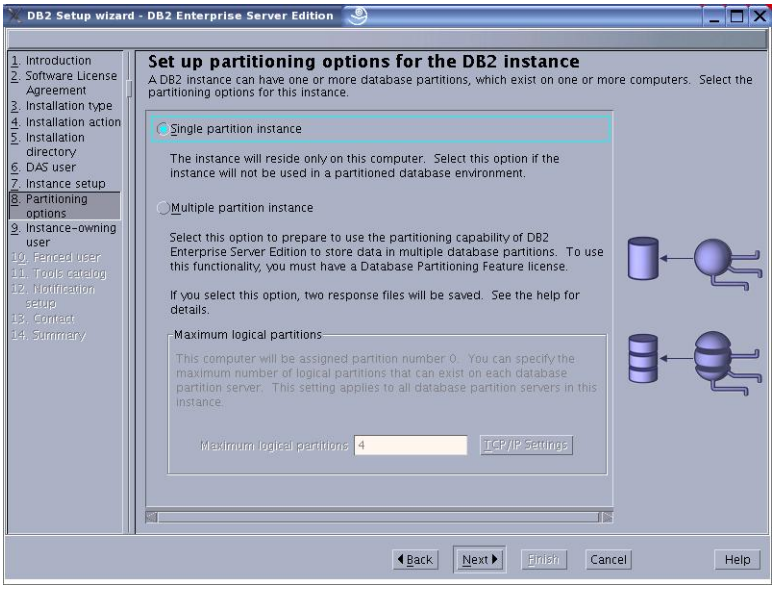
Now you must create a DAS user for DB2 9. You can accept the default name or change it if you want to do so. When you are finished, enter a password for the user and click **Next**.



Leave **Create a DB2 instance** selected and click **Next**.

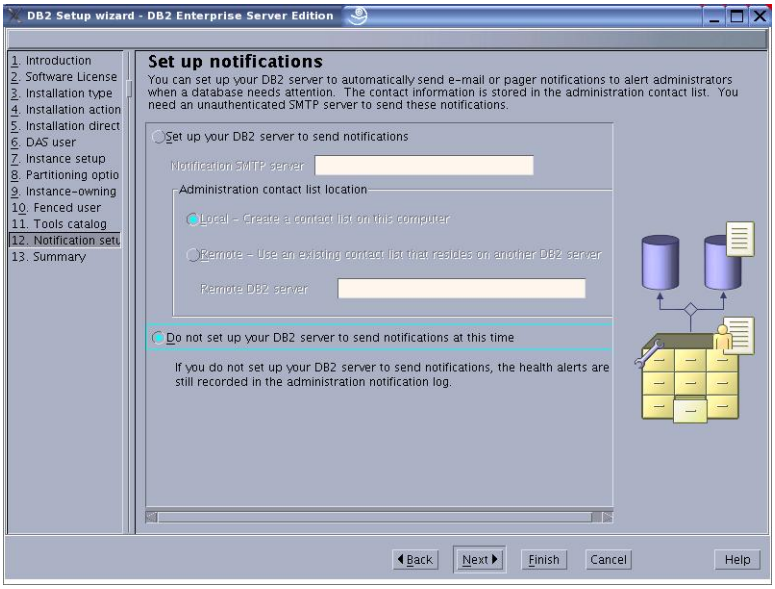


Leave the setting on **Single Partition** and click **Next**.

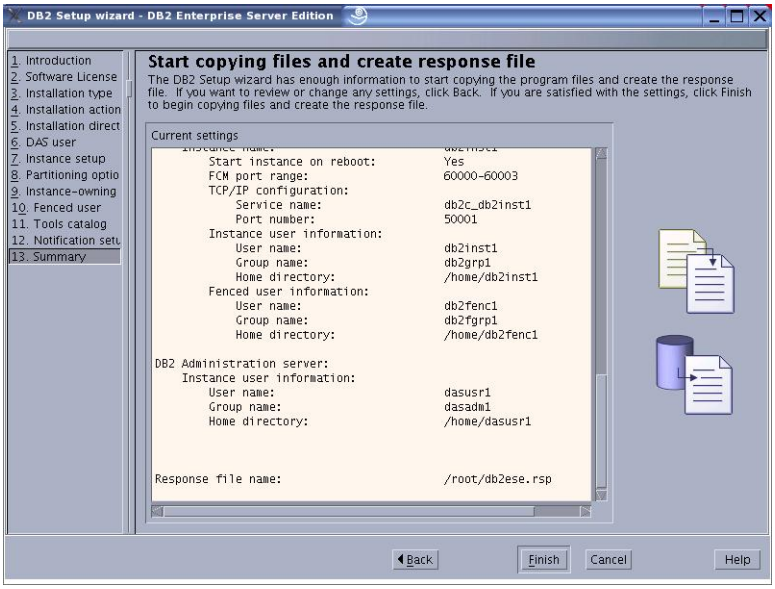


Similar to the way you create the DAS user, create an Instance User and a Fenced User. Again, you may change the names if you want, then enter a password and click **Next** to continue.

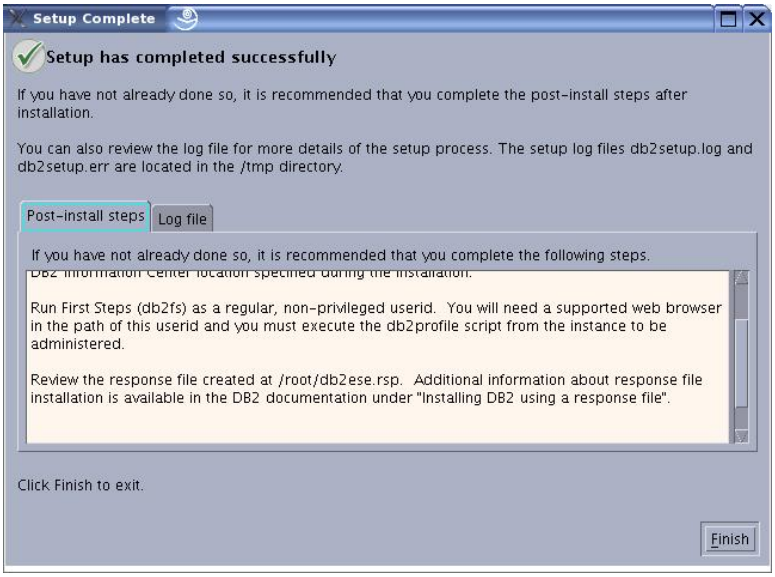
Leave the DB2 tools page as is and click **Next**, and then select **Do not set up your DB2 server to send notifications** and click **Next**.



Review the installation settings. If you want to make a change, you can click **Back**; otherwise, click **Finish** to begin the installation.



DB2 Enterprise Server Edition 9 is now installed.



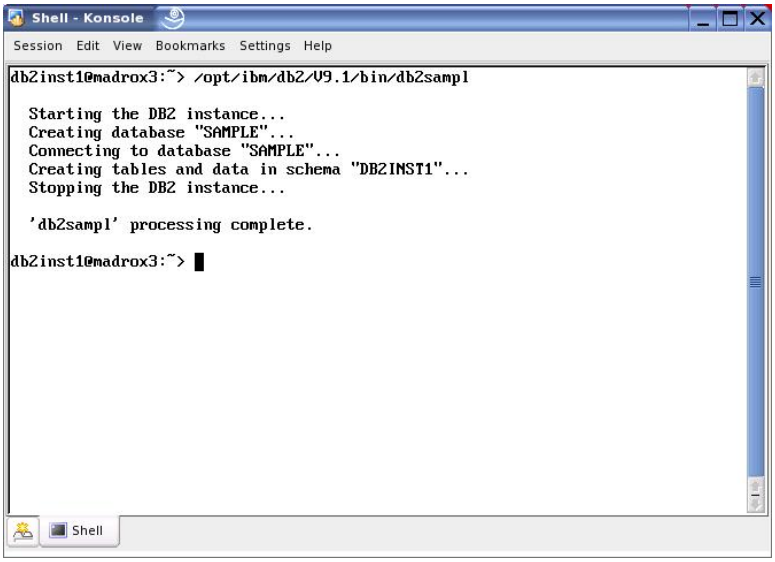
5.2 Create a Sample Database

We will now create the sample database for our application to work with.

Begin by logging in as the DB2 instance owner, which, if you accepted the default, is user ‘db2inst1’.

Next, enter the command:

```
# db2sampl
```



When you see ‘processing complete, you now have a new database named ‘SAMPLE’, which is prepopulated with sample tables and data.

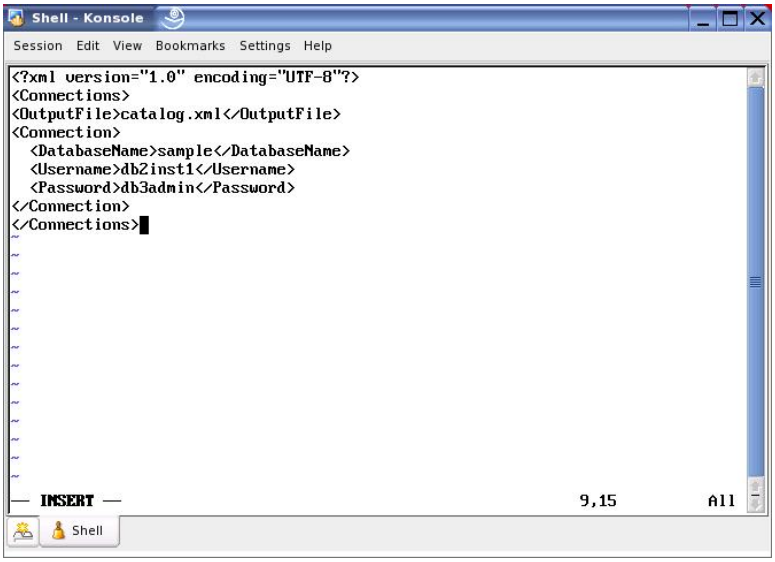
5.3 Run xkoto Catalog Scanner

The xkoto Catalog Scanner will search though the DB2 catalog and create a file called ‘catalog.xml’. This file will be used by the xkoto GRIDSCALE Database Load Balancer to identify stored procedures, aliases, functions, foreign key references, triggers, and views used in a particular database. For this reason, whenever a new database is created on the database server, a new ‘input.xml’ and ‘catalog.xml’ must subsequently also be created.

Ensure that you are logged in as the instance owner and begin by creating an XML file called ‘input.xml’. This file will contain information about the DB2 server that is needed by the db2CatalogScanner to determine which database to scan. We are going to create an ‘input.xml’ file using the vi text editor. Switch to the same directory that contains the db2CatalogScanner file, and then enter the following command:

```
# vi input.xml
```

To tell the db2CatalogScanner to create an output file called ‘catalog.xml’ that will contain information from a database named ‘sample’ with instance owner ‘db2inst1’ and password ‘db3admin’, enter the following text into the ‘input.xml’ file:

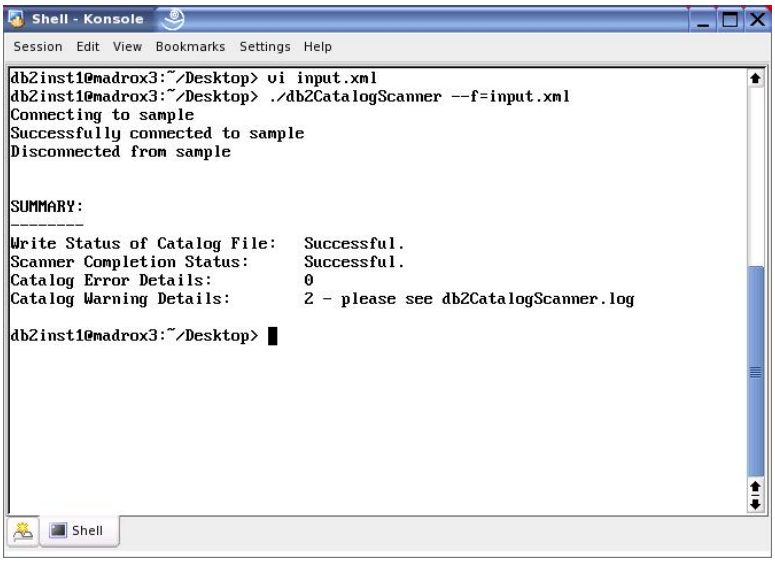


Once you have finished editing the file you can save the 'input.xml' file by pressing *Esc* and typing '`:wq`' to save the file and close the editor.

Now that the 'input.xml' has been created, run the db2CatalogScanner on your database server so that it creates a 'catalog.xml' file.

You should still be in the same directory as the db2CatalogFile, so enter the following command to run the Catalog Scanner:

```
# ./db2CatalogScanner --f=input.xml
```



After running the db2CatalogScanner, your 'catalog.xml' file can be found in the same directory as your 'input.xml' file. Now you must transport the 'catalog.xml' file from the database server madrox3 to the xkoto GRIDSCALE server madrox2. If you have installed the xkoto GRIDSCALE Database Load Balancer on the xkoto GRIDSCALE server using the default path, you will find the 'catalog.xml' that needs to be replaced in '/opt/xkoto/gridscale/config'. One method of transporting this file is to use the scp utility:

```
# scp catalog.xml root@madrox2:/opt/xkoto/gridscale/config
```

When you have replaced this file, this step is complete.

5.4 Install xkoto GRIDSCALE Connector

Prior to installing the xkoto GRIDSCALE DB Connector on madrox3, you must ensure that the xkoto GRIDSCALE Database Load Balancer on madrox2 is running. You can do so by entering the following commands on the xkoto GRIDSCALE server madrox2:

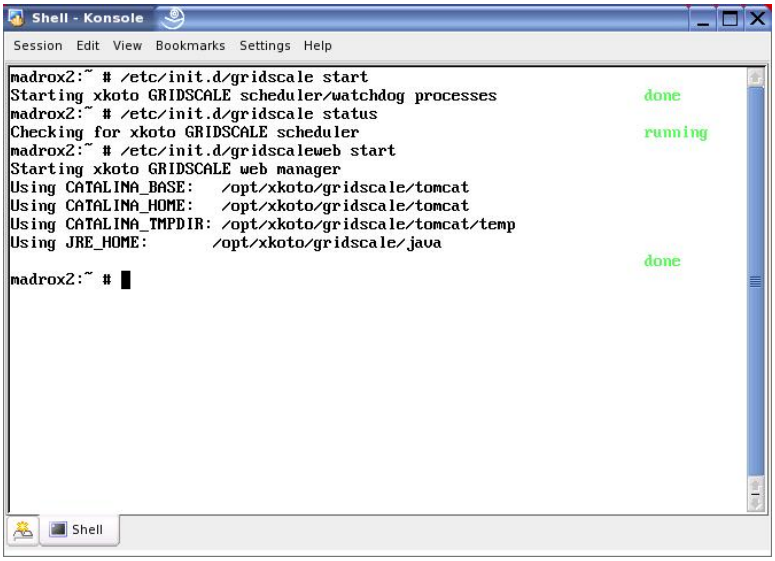
```
# /etc/init.d/gridscale start
```

To confirm that it is running, enter the command:

```
# /etc/init.d/gridscale status
```

Now start the xkoto GRIDSCALE Web server so that we can access the xkoto GRIDSCALE Administrator Console, by entering:

```
# /etc/init.d/gridscaleweb start
```



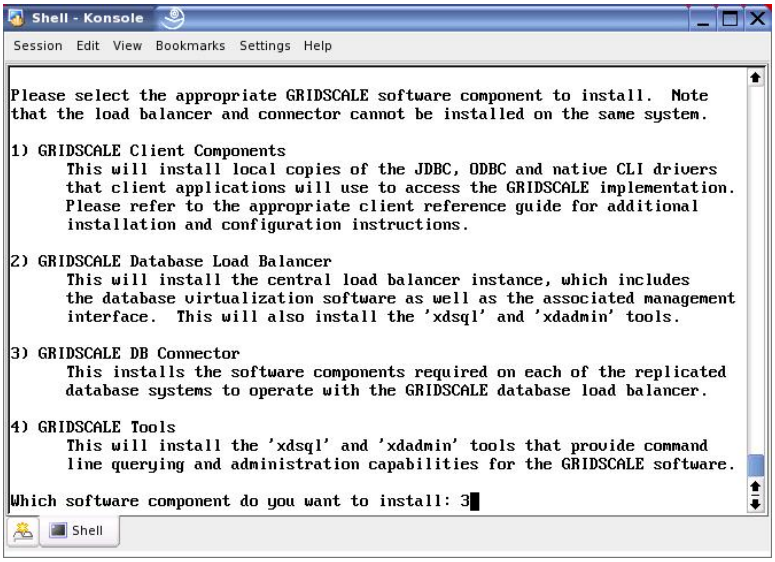
Now that the xkoto GRIDSCALE Database Load Balancer is running, we can install the GRIDSCALE DB Connector on madrox3. The xkoto GRIDSCALE DB Connector acts as a communication bridge between the xkoto GRIDSCALE Database Load Balancer and the IBM DB2 9 database server.

To install the xkoto GRIDSCALE DB Connector on madrox3, switch to the directory where the xkoto GRIDSCALE distribution was extracted and type:

```
# ./install.sh
```

Press *Enter* to continue, and then read and agree to the license terms by typing '`yes`' and pressing *Enter*.

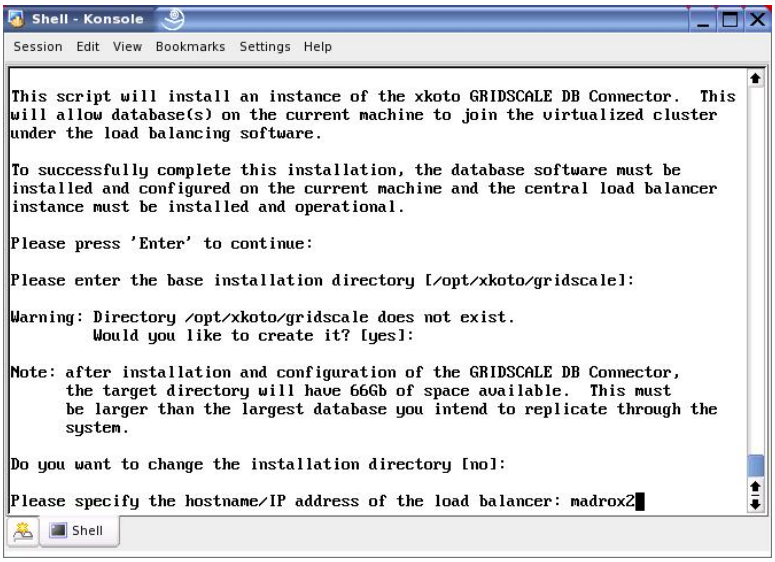
We need to install xkoto GRIDSCALE DB Connector so type '`3`' and press *Enter* to continue.



Press *Enter* to continue.

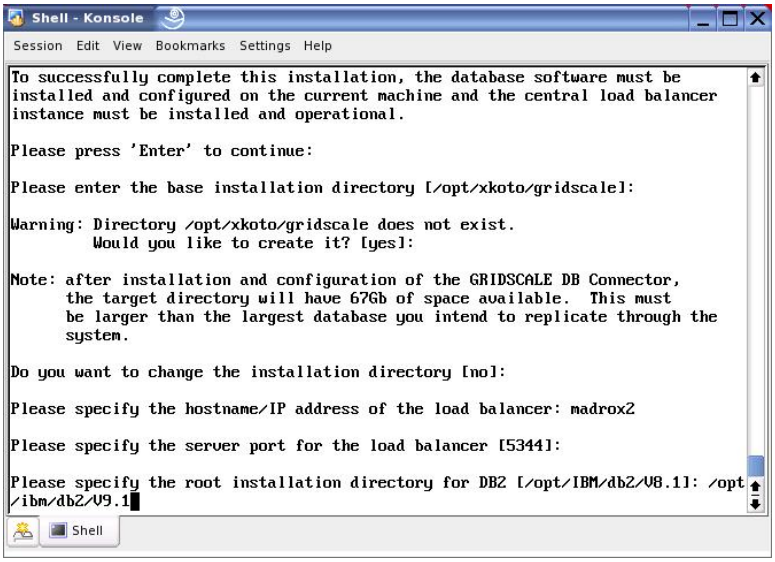
Accept the default installation path for xkoto GRIDSCALE DB Connector by pressing *Enter;* press *Enter* a second time to create the directory and a third time, to say *No* that you don't want to change the directory.

Enter the host name of xkoto GRIDSCALE server and press *Enter*. In our example, it is called madrox2.

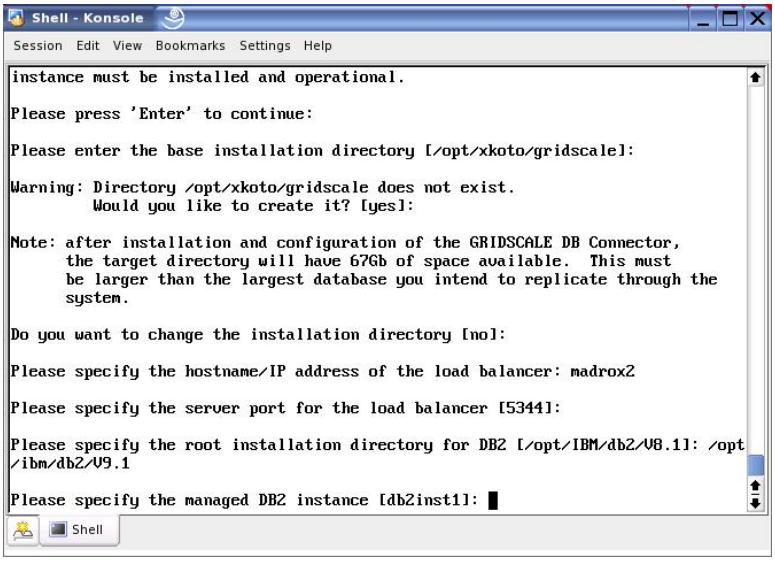


By default, xkoto GRIDSCALE uses port 5344. If you did not change this during the installation of the xkoto GRIDSCALE Database Load Balancer, press *Enter*. If you did, you must enter the new port number.

Shown in the window is the default path where DB2 is installed. Since we are using IBM DB2 9, the default path must be changed. Type in your DB2 9 installation path '`/opt/ibm/db2/V9.1`' and press *Enter*.

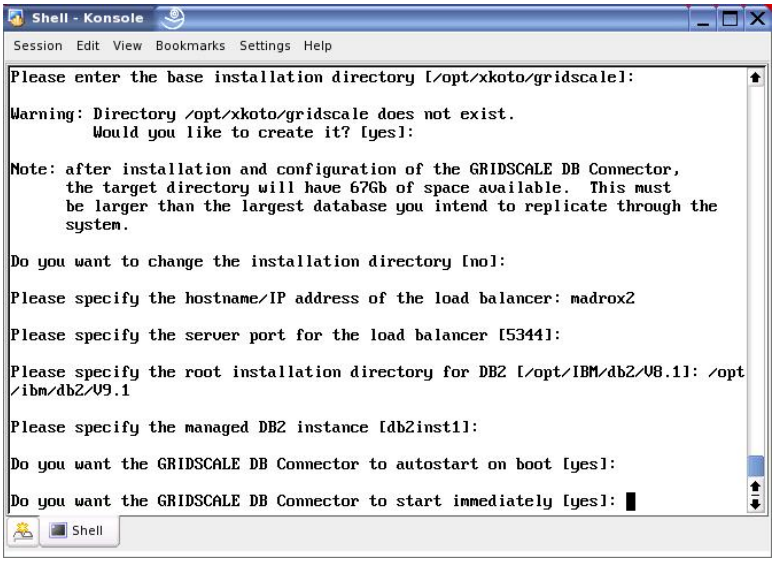


By default, DB2 creates an instance called 'db2inst1'. If you did not change the name of this instance, press *Enter*; otherwise, enter the new name here.

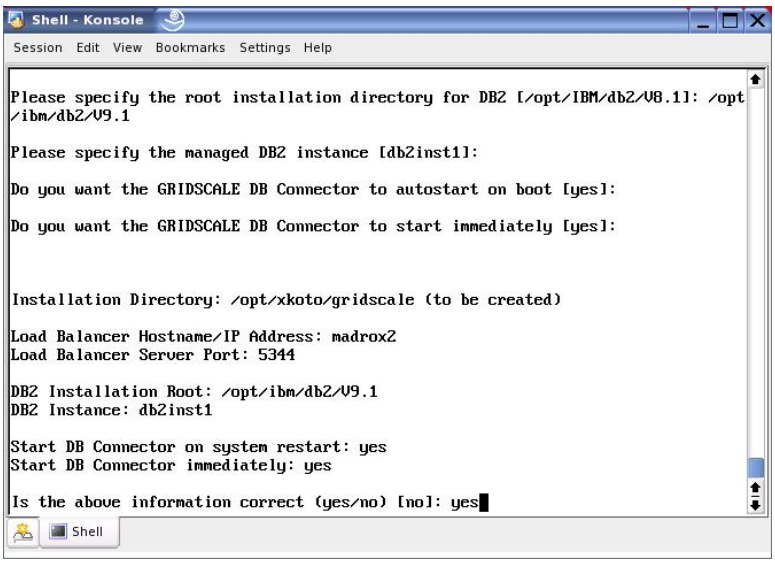


Press *Enter* to continue. This will allow xkoto GRIDSCALE DB Connector to automatically run every time the database server boots up or restarts such that the xkoto GRIDSCALE Database Load Balancer will be able to detect this node.

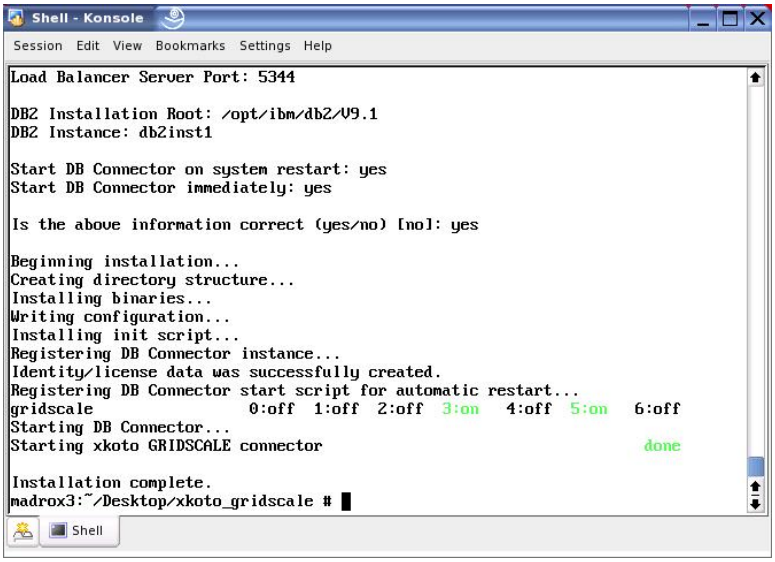
Press *Enter* again to start the xkoto GRIDSCALE DB Connector now.



Review your installation settings. If you are satisfied with the settings, type '`yes`' and press *Enter* to continue.



When xkoto GRIDSCALE DB Connector has finished installing on madrox3, your database server is ready for use.



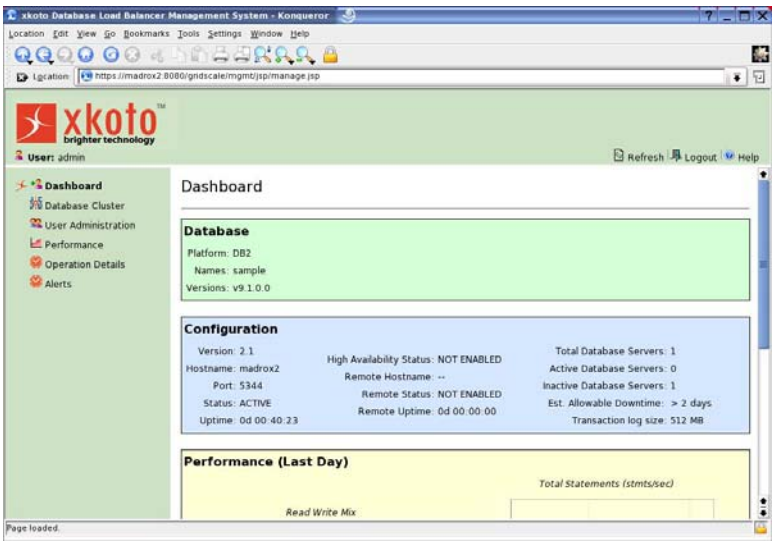
6. Completing the xkoto GRIDSCALE Setup

6.1 Activating the Database Server

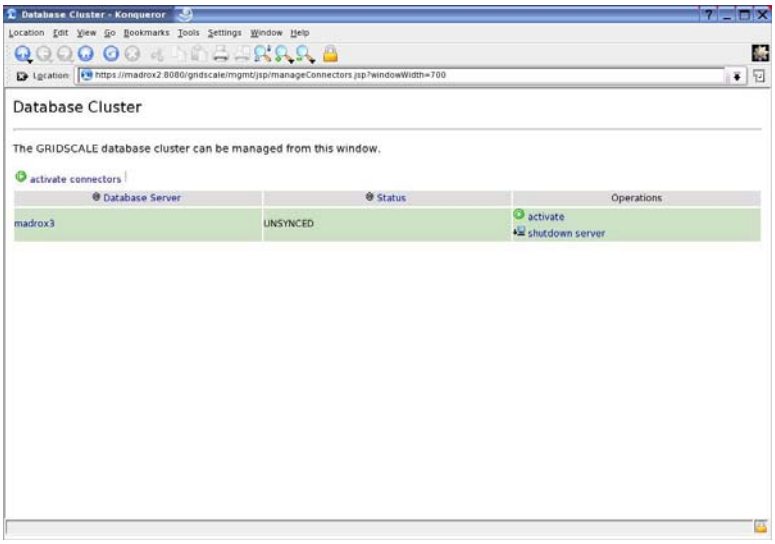
Now that all the required components have been installed, we can activate the database server so that we may begin using our database. This step can be performed from any machine in any location as long as it has access to the xkoto GRIDSCALE server.

Log into the xkoto Administrator Console by entering '`https://madrox2:8080/gridscale`' in your browser. Use the default username and password 'admin' and 'admin', respectively.

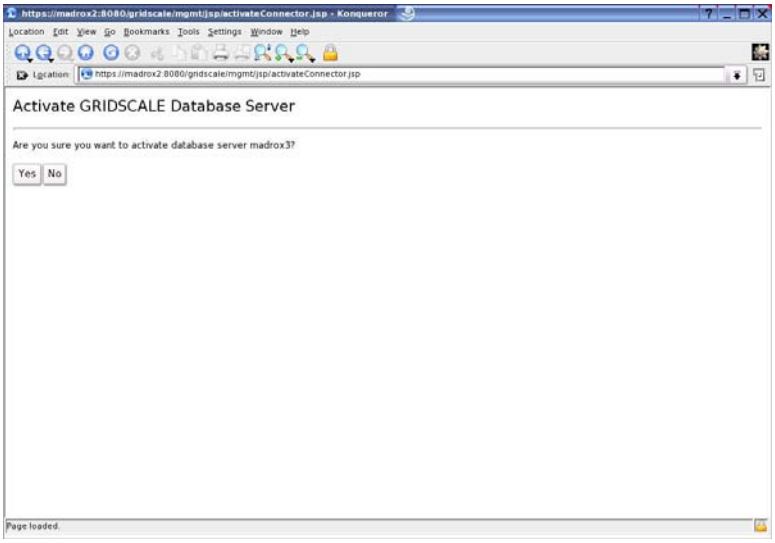
Under the configuration box, you will see all database servers that are available. At this point, we have one database server that is inactive. Click **Database Cluster** on the left.



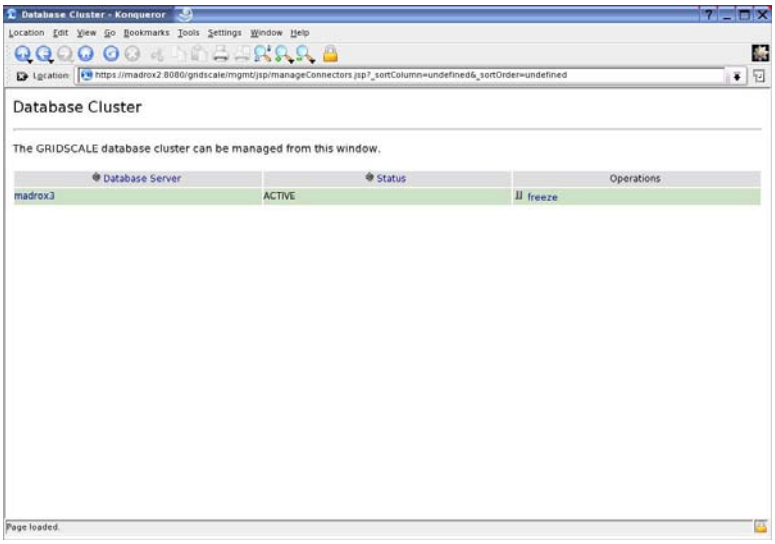
Click **activate** under the Operations column.



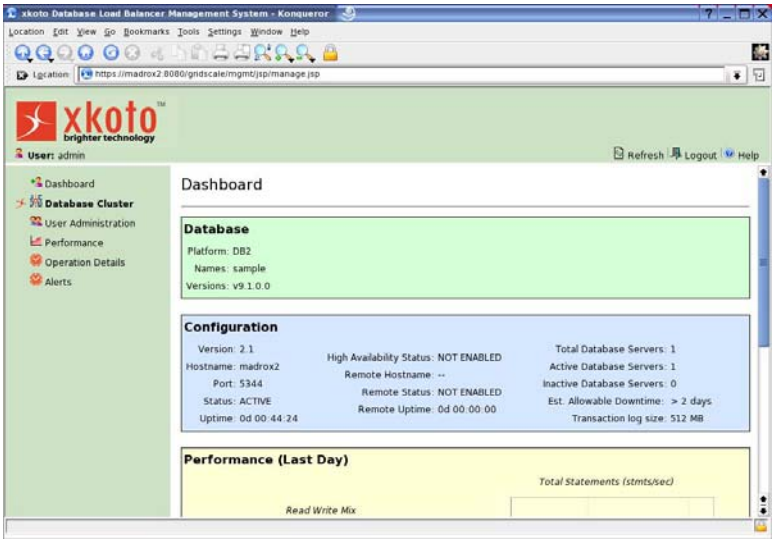
When asked to confirm, click **Yes** to activate the database server.



The database server will now activate and synchronize with the xkoto GRIDSCALE server. Once the database server has successfully activated, the status will become “Active” and the database is ready for use.



If you return to the Dashboard you should see one active database server in the Configuration box.



Your database is now active and ready to be used.

6.2 Start Your Application

We can now start our application and begin generating transactions on the database. We will be using DB2VPS for our application to generate a realistic workload containing both reads and writes. Note that we do not connect directly to the database server, but rather to the xkoto GRIDSCALE server. In our case, we would enter the following information in our application:

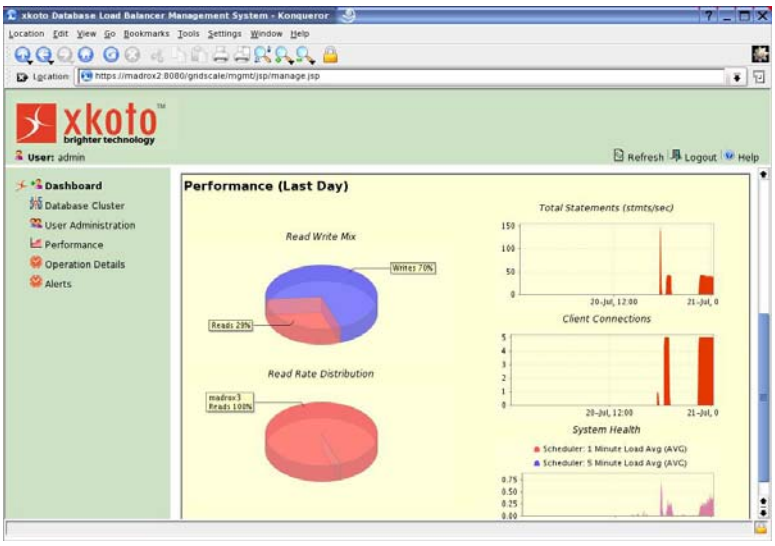
```
Driver: xkoto JDBC
Server: madrox2
Port: 5344
Database name: sample
Username: db2inst1
Password: db3admin
```

6.3 Monitoring the Performance

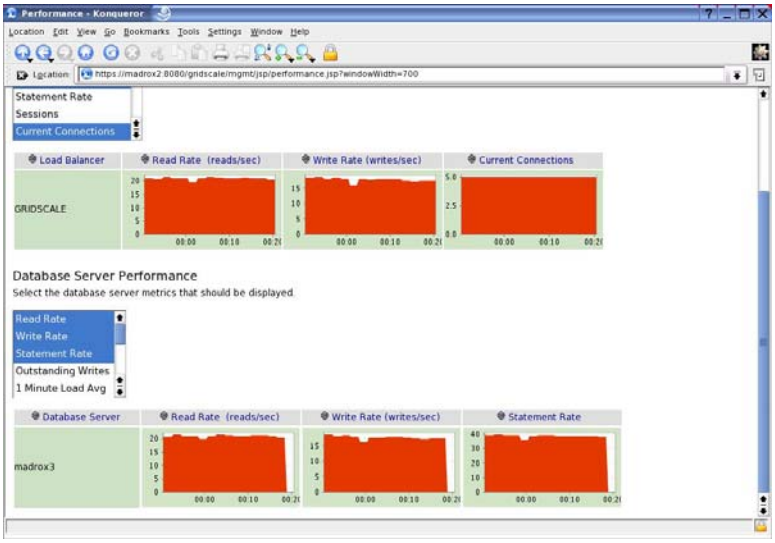
To obtain a more precise estimate of your xkoto GRIDSCALE system's performance, you may want to let your application run for a few hours. This will average out most read and write transactions to provide more accurate graphs.

To check the performance, log into the xkoto Administrator Console as described in Section 6.1 above.

Immediately after you log in, you will see the GRIDSCALE Dashboard. This page will show you the Read / Write Mix as well as the read contribution by each of the database servers. At this point, you will only have one database server so 100% of read will come from madrox3.



If you click Performance on the left menu, you will see the screen below. In the performance section, you can see live read and write transactions that are occurring.



Looking closely at the transaction graphs, you will notice that the read requests sent to the xkoto GRIDSCALE Database Load Balancer are equal to the read transactions preformed by our single database server madrox3. This reaffirms what we already know: there are no performance or scalability gains with the use of one database server.

Now let's proceed and see how you can improve the performance by adding additional database servers.

7- Adding Additional Database Servers

7.1 Activating Additional Database Servers

In the previous section, we explained how to set up a typical database system commonly used in small and medium-sized business. As business grows, a company may soon find their database capabilities unable to keep up with their increasing demands. Database sizes multiply, transactions increase, and 24X7 access to the information becomes increasingly valuable.

In this section, we will now demonstrate how you can upgrade your current xkoto GRIDSCALE configuration to increase the performance and reliability by adding two additional database servers.

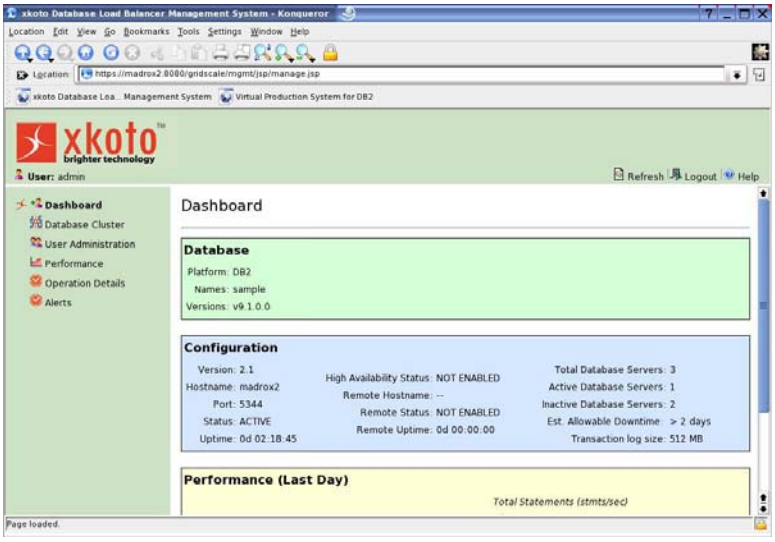
Begin by setting up these two machines–madrox4 and madrox5–as described in Sections 5.1 and 5.4–Installing the Database Server. Even though our remote database server, madrox5, is in a different geographical location, the setup and functionality is identical to any other database server. Note that it is only necessary to install DB2 9 and xkoto GRIDSCALE DB Connector on these two machines–Sections 5.2 and 5.3 do not have to be repeated. You do not need to create a database because we will use the xkoto GRIDSCALE replication process described later to copy our pre-existing database from madrox3 to your additional database servers.

After IBM DB2 9 and xkoto GRIDSCALE DB Connector have been installed on both database servers, you are ready to replicate the SAMPLE database from madrox3.

Before you can begin the replication of the database, you must stop your application. This is because during the replication process the current database must be put into the "frozen" state–and since no other database servers exist yet, transactions will be lost if the application is running. This is one reason why it is recommended to have at least three database servers. In the event that the first database server fails, you can keep the second active, and freeze the third to use as a base to replicate the database back to the first machine. Refer to xkoto's Web site for more information regarding this topic.

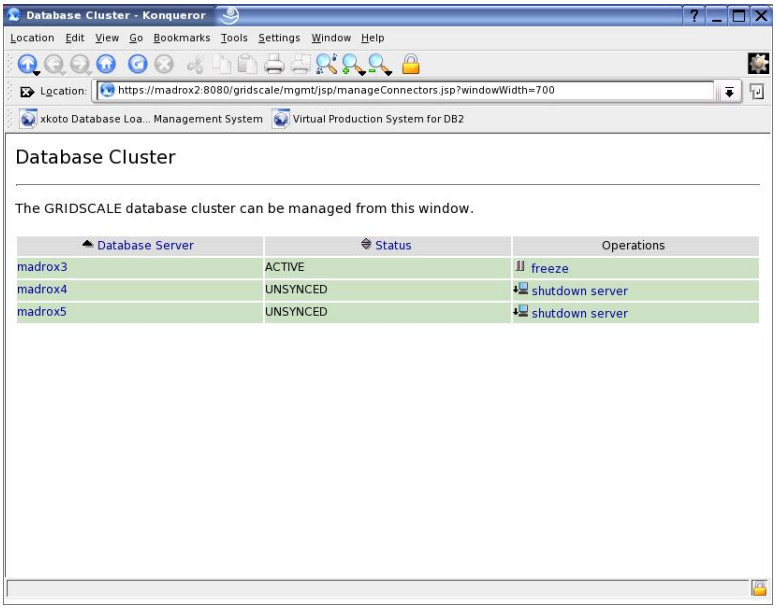
After stopping your application, log in to the xkoto GRIDSCALE Administrator Console, as described in Section 6.1.

After you have logged in, you will see the Dashboard. Under the Configuration section, you should see all three database servers, one active and two inactive. Click **Database Cluster** to continue.

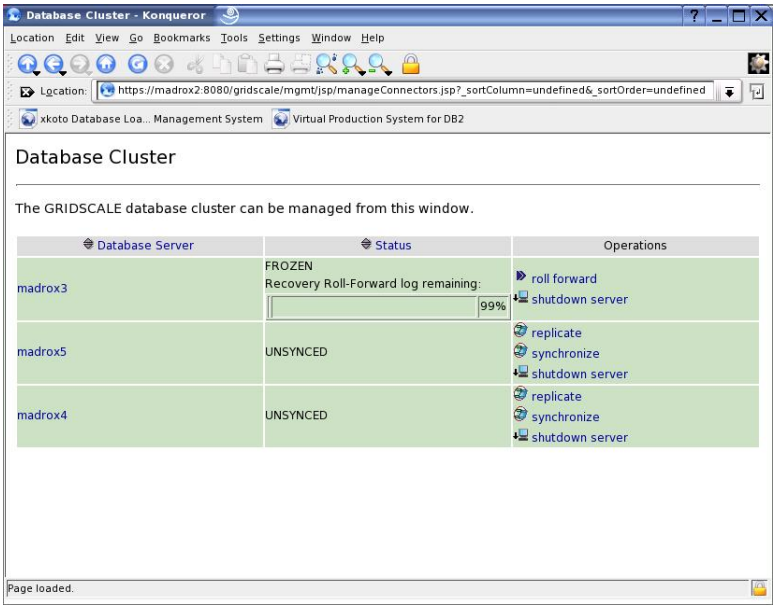


You will see a list of all database servers but only active servers contain up-to-date information on their database. For this setup, the objective is to copy the entire database from the one

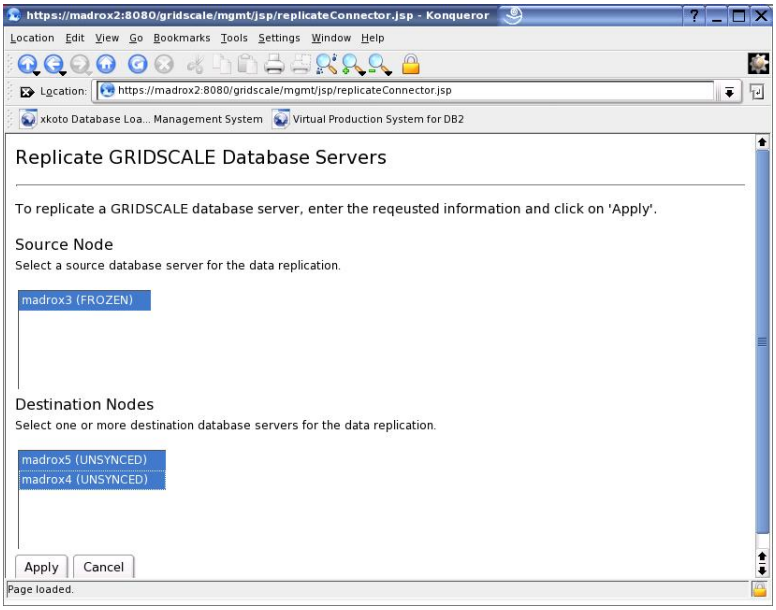
active database server to the two other database servers such that we have a cluster of database servers with identical databases on each one. To start the process, click **Freeze** beside the active server. When asked to confirm the freeze, click **Yes**.



Now that the database server has been *frozen*, no changes can be made to its database. This is done to maintain consistency between the original database and the destination during the replication process. Click **Replicate** beside any of the Unsynched databases.

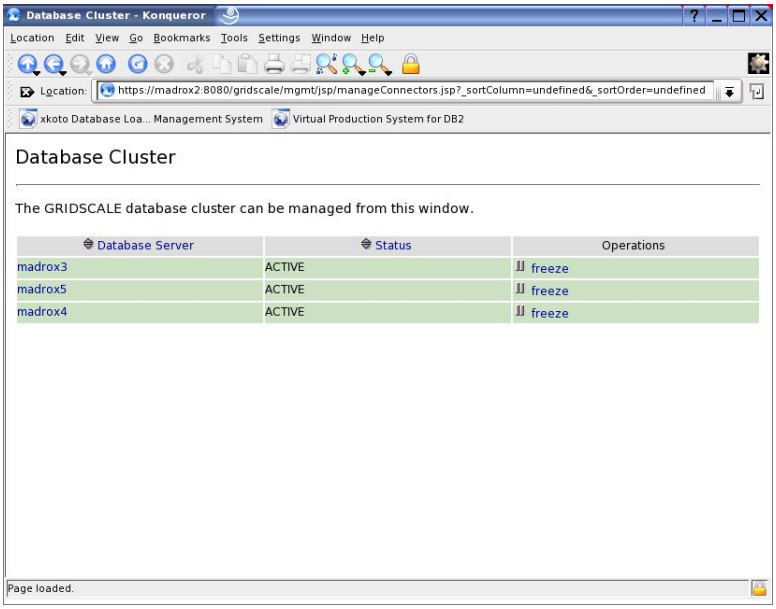


Select the **Source Node** where you want to copy the database from and select the **Destination Nodes** where you want to copy the database to. If you want to select multiple nodes, you can do so with the *Shift* or *Ctrl* key. Click **Apply** when you are finished and **Apply** again when you see the confirmation page.

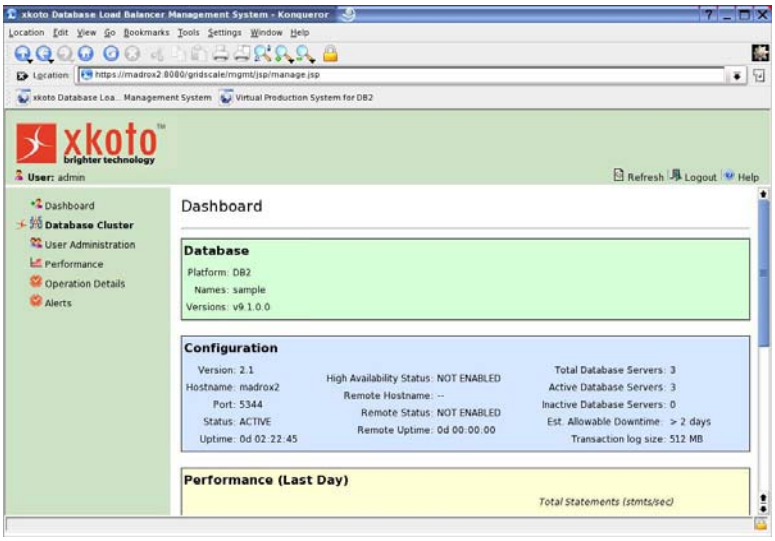


xkoto will now automatically replicate and synchronize the databases. Depending on the size of the original database, this may take some time.

Once all the database servers have been synchronized, their status will become ”Active”.



If you go back to the Dashboard, you should see three active database servers.



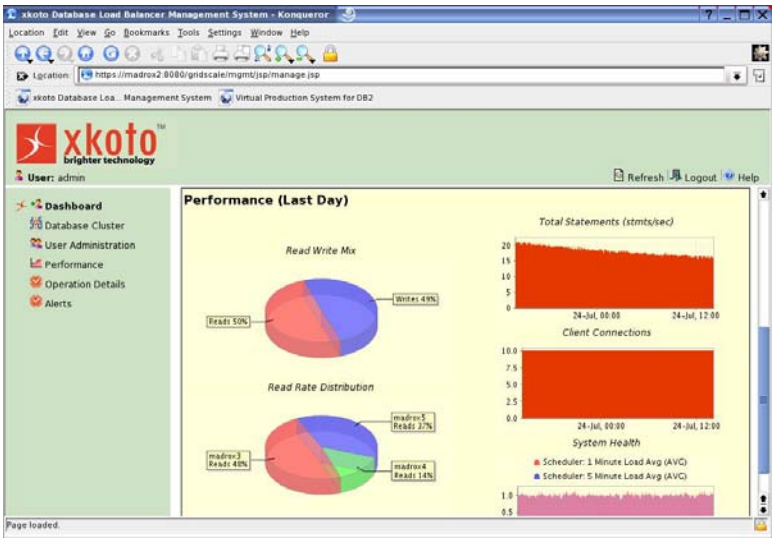
Once you have all three database servers active, you have completed this step and your highly available DB2 9 database running on xkoto GRIDSCALE is ready to be used. You can now start your application and begin generating transactions.

7.2 Monitoring Performance

It is recommended to leave your application running for at least a day in order to generate accurate performance statistics.

To monitor the performance, log in to the xkoto GRIDSCALE Administrator Console as mentioned earlier.

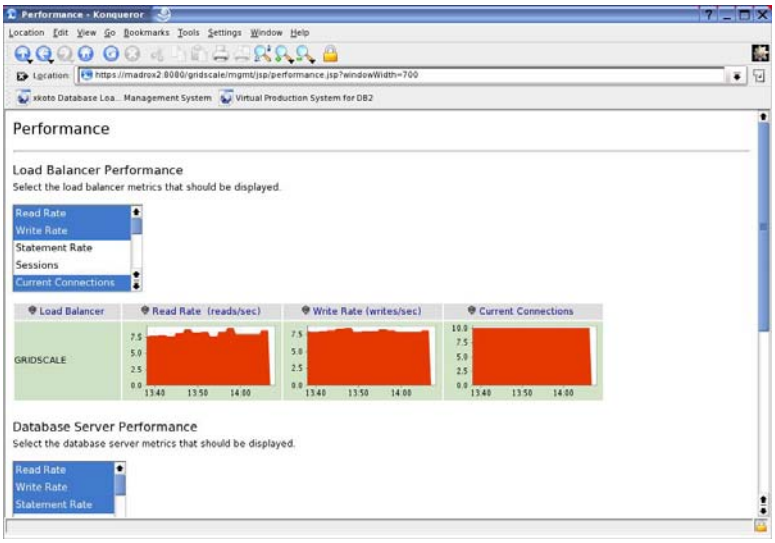
You will now see the Dashboard. If you look inside the Performance box, you should notice the *Read Rate Distribution* being divided among the three database servers. Since we have three database servers, in theory each should contribute 1/3 of all reads. Depending on how long your application has been running, this may or may not be the case because it requires time to average out the read contributions.

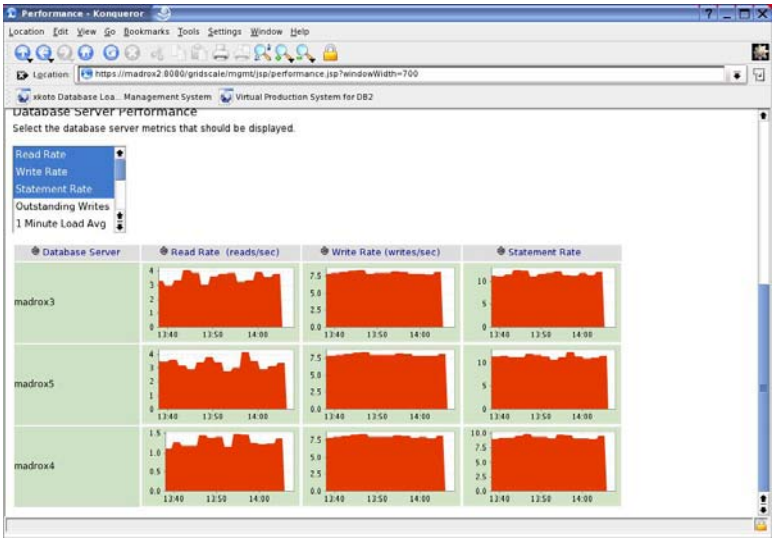


If you click **Performance** in the left menu, you should see the screen below. Begin by observing the read transactions, which are sent to the xkoto GRIDSCALE Database Load

Balancer. If you scroll down, you will see how these read transactions are distributed among the three database servers.

As can be seen by the two following screenshots, you will notice that each of the read transactions preformed by a single database sever is about even and, in fact, when added together are equal to the total read requests sent to the xkoto GRIDSCALE Database Load Balancer.



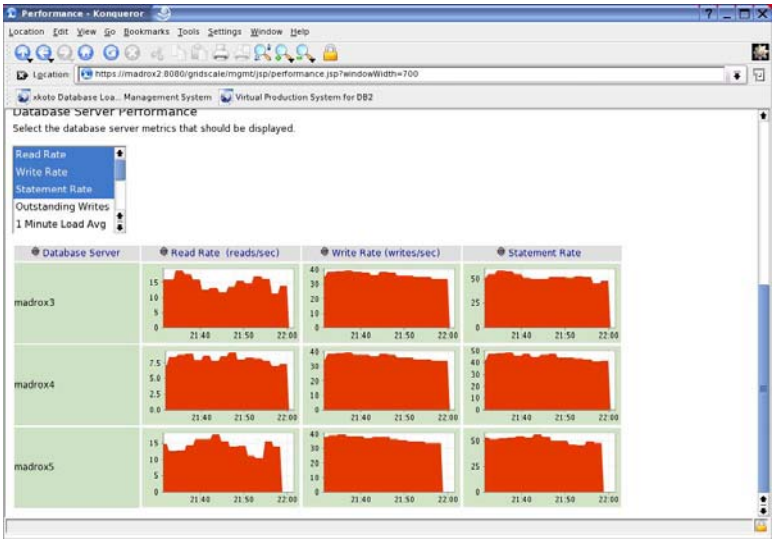


There is a lot more to functionality in the xkoto GRIDSCALE Administrative Console that we will not cover. Refer to the xkoto user manuals for more information.

8. High Availability Test

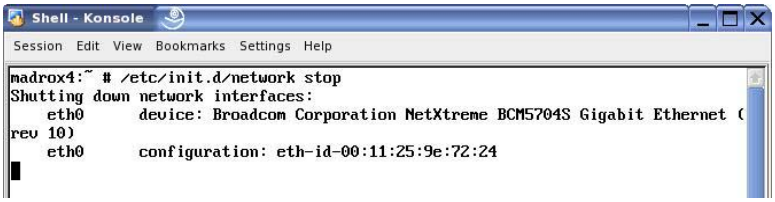
In this section we will demonstrate how xkoto GRIDSCALE provides a highly available DB2 9 database solution. Consider our current setup, which contains three database servers: two local and one remote. Everything has been running smoothly up until this point, but we will now simulate a severe network failure on one of the database servers by prematurely shutting down its network.

Shown below is the Performance graph with three database servers running. Notice how the read transactions are relatively evenly distributed on all three servers.



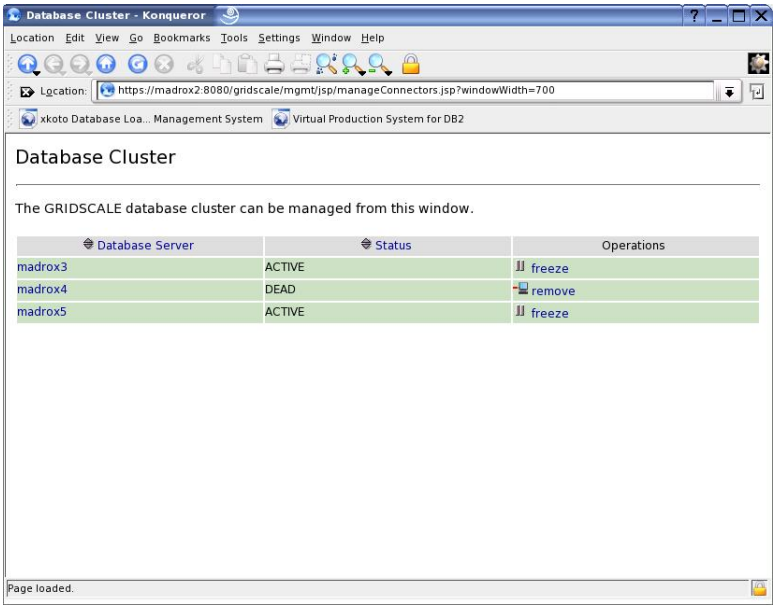
Now select a database server to shut down--we have decided to shut down our second server madrox4. Issue the shutdown command:

```
# /etc/init.d/network stop
```

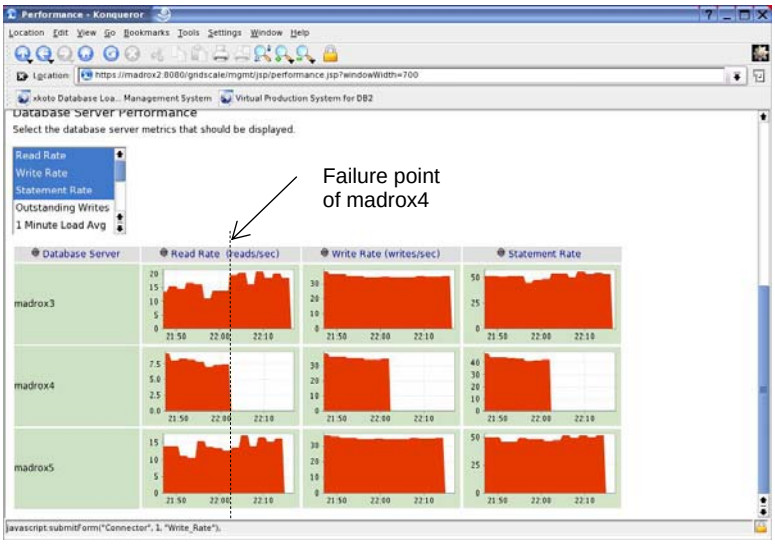


Note that this will disable the machines network and you will need physical access to fix this.

In the xkoto GRIDSCALE Administration Console, madrox4 should appear DEAD in the Database Cluster page.



Now that one of your database servers has “failed”, continuously refresh the Performance page by pressing F5. You will eventually see graphs similar to the one below. Notice how the read transactions for madrox3 and madrox5 have increased after the failure point of madrox4 to compensate for our dead server.



This quick demonstration shows how the xkoto GRIDSCALE Database Load Balancer offers effective HA with your DB2 9 database since the transactions remain uninterrupted with the failure of a database server. For more detailed information about xkoto GRIDSCALE HA, refer to the GRIDSCALE user manual.

9. Troubleshooting

9.1 While running db2CatalogScanner, it says libdb2.so.1 cannot be found.

Begin by using switching to the directory that contains the db2CatalogScanner file. Then use the command ‘ldd db2CatalogScanner’, which should show the output similar to:

```
# ldd db2CatalogScanner
linux-gate.so.1 =>  (0xffffe000)
libdb2.so.1 => not found
libc.so.6 => /lib/tls/libc.so.6 (0x55582000)
libpthread.so.0 => /lib/tls/libpthread.so.0(0x5569c000)
libm.so.6 => /lib/tls/libm.so.6 (0x556ac000)
/lib/ld-linux.so.2 (0x55555000)
```

If you see that libdb2.so.1 is pointing to ‘not found’, that means the path to this file is not properly set up. Add the correct path by using the command `‘export LD_LIBRARY_PATH=/opt/IBM/db2/V8.1/lib’`

9.2 GRIDSCALE Administrator Console cannot detect a database server

Ensure that xkoto Connector is currently running on the database server that cannot be detected. Use the command:

```
# /etc/init.d/GRIDSCALE status
```

If it does not respond with ‘running’, then you must start the xkoto Connector by entering:

```
# /etc/init.d/GRIDSCALE start
```

9.3 GRIDSCALE Connector cannot start

Before attempting to start the xkoto GRIDSCALE Connector, DB2 9 must be started first. Log in as the instance owner and run the command `'db2start'`. Then log in as the root user and proceed to start the connector by entering the command:

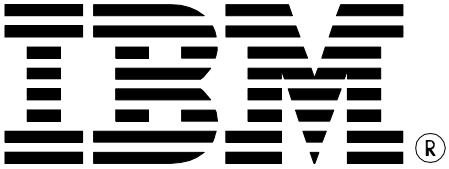
```
# /etc/init.d/GRIDSCALE start
```

9. Additional Info

xkoto
http://www.xkoto.com

IBM WebSphere
http://www.ibm.com/websphere

IBM DB2 9
http://www-306.ibm.com/software/data/db2/



IBM Canada
8200 Warden Avenue
Markham, ON
L6G 1C7
Canada

Printed in United States of America
08/06