Server Installation Guide

BigWorld Technology 2.0. Released 2010.

Software designed and built in Australia by BigWorld.

Level 2, Wentworth Park Grandstand, Wattle St
Glebe NSW 2037, Australia
www.bigworldtech.com

Table of Contents

1. Overview 5
2. Hardware Requirements 7
    2.1. CPU Specifications 7
    2.2. Duals/Single/Quad/Blades 7
    2.3. Network Interface Cards 8
    2.4. Disk Storage 8
    2.5. Power Supplies 8
    2.6. Memory 8
    2.7. NOC Bandwidth 8
    2.8. VMWare 8
3. Installing CentOS 5 9
    3.1. Installing 9
    3.2. Post-installation Setup 11
        3.2.1. Install updates 11
        3.2.2. Configure services 11
        3.2.3. Install build tools 11
        3.2.4. MySQL server and development libraries 11
        3.2.5. Changing UIDs 12
4. Installing the BigWorld Server 15
    4.1. Obtaining the Package 15
    4.2. Installing BWMachined 15
        4.2.1. BWMachined RPM Package 15
        4.2.2. Run the Daemon Installation Script 15
    4.3. Configuring BWMachined 16
        4.3.1. Creating `~/.bwmachined.conf` 16
        4.3.2. Creating `/etc/bwmachined.conf` 17
    4.4. Check That BWMachined is Running 19
    4.5. Configure MySQL Server 19
        4.5.1. MySQL Tweaks 21
    4.6. Installing Server Tools 21
5. Cluster Configuration 23
    5.1. Security 23
    5.2. Routing 24
    5.3. Disabling cron jobs 24
6. Development Environment Configuration 27
    6.1. Setting up shared resources 27
    6.2. Configure bwmachined 27

Chapter 1. Overview

This document describes how to properly install Linux for the BigWorld Server and how to install the BigWorld Server.

CentOS 5 64 bit is the recommended platform for development and production environments.

Chapter 2. Hardware Requirements

The quick summary: as a minimum, we require the following:

- 1GHz CPU (non-mobile CPU preferred)
- 256MB RAM
- 8GB Hard Disk
- 100Mbps Network Interface Card

See below for recommended hardware requirements.

2.1. CPU Specifications

In general, use the fastest CPU's that you can. Faster CPU's mean more entities per CPU and fewer machines.

As far as which CPU to buy, look at the normal kinds of things for servers: big L1 and L2 cache sizes, and fast front-side buses are always better than small L1 and L2 cache sizes and slow front-side buses.

Multiple processors will help you out due to lower network traffic and fewer machines, until such time as the processors are generating too much data to get to the network card (or over the PCI bus to the network card).

If you have multiple processors, then make sure that you have one server component running on each CPU.

2.2. Duals/Single/Quad/Blades

In general CPU density in each box is a price decision. Blade machines are expensive, but if you are paying expensive rates for your NOC it may work out to be cheaper. If we ignored the NOC cost, we recommend dual CPU machines.

An example Blade setup would be as follows:

BladeCenter LS20 885051U

- Processor: Low Power AMD Opteron Processor Model 246 (Standard)
- Memory: 4 GB PC3200 ECC DDR RDIMM (2 x 2 GB Kit) System Memory
- IBM eServer BladeCenter ™ Gigabit Ethernet Expansion Card
- SCSI Hard disk drive 1 : 73GB Non Hot-Swap 2.5" 10K RPM Ultra320 SCSI HDD

BladeCenter 86773XU

- Optical device: IBM 8X Max DVD-ROM Ultrabay Slim Drive (Standard)
- Diskette drive: IBM 1.44MB 3.5-inch Diskette Drive (Standard)
- Power supply modules 1 and 2: BladeCenter 2000W Power Supplies one and two (Standard)
- Management modules: BladeCenter KVM / Management Module (Standard)
- Switch module bay 1: Nortel Networks Layer 2/3 Copper GbE Switch Module for IBM eServer BladeCenter
- Switch module bay 2: Nortel Networks Layer 2/3 Copper GbE Switch Module for IBM eServer BladeCenter

A setup with 10 LS20 blades in a 86773XU BladeCenter would cost approx $60k USD.

2.3. Network Interface Cards

1Gbps NICs are recommended. The accurate way to determine what NIC is required is to measure the inter-server traffic for your game. If the traffic is reaching 25% of the cards capacity we recommend using a faster card (i.e. if the traffic is more than 25Mbps use a 1Gbps NIC). Note most 100Mbps NIC cards cannot handle more than 50Mbps sustained throughput.

2.4. Disk Storage

For the general machines (CellApps, BaseApps, and the various managers) RAID disk setups are not recommended. None of these machines use the disk sub-system extensively (and are certainly not disk bound). Use standard drives, that are big enough to store the entire world data (typically in the order of 1 to 10G). If a drive dies it can be replaced and the data copied from the master.

The machine with the master copy of the data should use a RAID 5 system for speed and data integrity. Hot-swap drives will facilitate easy replacement when drives fail. The database server also needs to use RAID 5 for data integrity and Logical Volume Management for snapshotting. We also recommend using 10k or 15k RPM drives (SATA or SCSI). The database for a game stores backup copies of all entities. This database can be large, potentially 10G to 1TB, but this should be measured during development. This can be estimated by multiplying the number of entities by their size.

2.5. Power Supplies

We recommend the use of dual redundant PSUs for the database machine, and the master data server. All other machines can use standard single PSU's, since a failure of these machines is not critical. It is cheaper to let the BigWorld fault tolerance system do the work on the software side.

2.6. Memory

To calculate memory requirements for the CellApp we recommend the following:

- Around 32MB to 128MB, for Linux to run comfortably (depends on how well you have stripped back the kernel and system services).
- Around 32MB for a CellApp or BaseApp to run comfortably with no entities on them and no spaces loaded.
- Enough RAM for your entities, and for your world geometry (remember that the cell needs to load up enough geometry to cover for the AoI for all entities it is supporting). This amount will depend on how dense your meshes are, and how much data is stored with each entity. 2GB of RAM would not be unusual for an average game.

The BaseApps would typically require around 512M, depending on how much entity data is stored. All other machines require around 512M.

2.7. NOC Bandwidth

This is easy to calculate. Multiply the number of players by the desired bandwidth per player. Outgoing bandwidth is generally higher than the incoming bandwidth.

2.8. VMWare

It is possible to use VMWare for single developer testing purposes, however VMWare is not recommended as a scaleable production configuration due to the timing latency that can be introduced. It is also important to note that if you are intending to use a VMWare image that the architecture of the package you are creating should be same as the intended machine you will run the image on. While this may seem counter-intuitive, this is important as the cross-architecture emulation significantly slows down the server and will generally causing process deaths due to a lack of responsiveness.

Chapter 3. Installing CentOS 5

Note

Even experienced users should skim the following sections to make sure that required packages are installed.

You may wish to refer to the CentOS documentation for additional notes and guidelines on installing and configuring CentOS.

3.1. Installing

- Boot the computer using the installation DVD, or other media (for example PXE boot). See the CentOS documentation for further details.

  You may need to select the CD/DVD ROM drive as a bootable device in the BIOS when installing from DVD.

- This installation guide is based on the graphical installer. Press ENTER at the first boot screen to select `Install in graphical mode`.

  If you are having trouble with video card drivers, then you can reboot and try the text-only installer.

- If this is the first time that the CD/DVD has been used, then it is worthwhile to use the built-in test option.

  The test will take around 15 minutes. If you do not want to test, just select Skip.

- **Language and keyboard type.**

  The language selected will be used for the installation procedure as well as being the default language of the installed system.

- **Installation method**

  Choose `Local CDROM` if you are using DVD to install, or you can choose your local CentOS mirror.

- **Disk partitioning**

  You can partition the disk as you see fit. The `Remove all partitions on selected drives and create default layout` option should work for most situations. You can modify the default layout by checking the `Review and modify partitioning layout` checkbox.

  Note

  If the machine will host the database server (running MySQL and DBMgr), and you are using secondary databases, you will need to use LVM partitions and you will also need to allocate some free space for the LVM snapshot. You can add unallocated space on one of your logical drives when reviewing the partitioning layout. See "Database Snapshot Tool" for more details about the snapshotting tool.

- **Boot loader configuration**

  Select the appropriate options for you machine (by default, the bootloader will be installed to the MBR).

- **Network configuration**

  Ensure that you have at least one network device listed, and that IPv4 is enabled for it.

In production, for BaseApp and LoginApp machines, there should be two network interfaces, one for external traffic, and the other for internal server traffic. In development, these can be the same.

The hostname can be specified manually or it can be set from DHCP.

If you are not using DHCP, you will need to enter the default gateway and DNS addresses.

- **Time zone selection**

  Select your time zone.

  We recommend that you leave the system clock as UTC as per the default.

- **Setting the `root` password**

  You will need access to the `root` account to install some BigWorld Server components, make sure you remember this password.

- **Package selection**

  If this is a production machine, we recommend that you have `Desktop - Gnome` unchecked. You can leave the other options unchecked, the specific packages that the BigWorld server requires will be installed later on in this guide.

  In development, you may wish to use the machine as a desktop development machine, in which case you can choose to install whichever packages you require for development, such as the `Desktop - Gnome` package group.

- **First boot configuration**

  The installation program will format the disk partitions, install the base system and system packages. After this process is complete, you will be asked to reboot the machine. On first boot, you will be prompted for further configuration.

- **Authentication**

  This tool sets up how your OS will look up user account information.

  BigWorld components assume that the username to UID mapping is unique across the network i.e. two users with the same name on two different machines will have the same UID and vice versa. If you are creating accounts with the same name on multiple machines, please ensure that they all have the UID by manually specifying their UID.

  Furthermore, the BigWorld server also assumes that server components started by the same user (as identified by their UID) belongs to the same server instance, even when those components are running on different machines. To run multiple BigWorld Server instances, multiple user accounts are needed.

  You can set up remote account information servers such as LDAP. We recommend using LDAP during development to ensure that every machine in the cluster has the same user set. Typically, each developer user has an account where they can run their own servers independent of other users.

  Refer to the OpenLDAP documentation for further information on how to configure an LDAP service to authenticate users.

- **Firewall configuration**

  For a development machine, the firewall should be disabled. The default firewall blocks all UDP traffic, which prevents the BigWorld Server from operating. You can disable the firewall by setting the `Security Level` option to `Disabled`.

For a production machine, you will need to setup specialised firewall rules for your specific security requirements. Guides for BigWorld Server specific firewall settings are given in the *Cluster Configuration* on page 23 section of this document.

BigWorld Server is known to work with the default SELinux settings (enforcing).

- **System services**

  For production machines, in order to avoid unexpected load spikes on your system from background services, we recommend that you disable any non-essential services. You can configure which services are started up at boot time.

  Services that are recommended to be disabled include:

  - cups
  - bluetooth
  - yum-updatesd

- **Finishing the installation**

  Once exiting the first boot configuration screen, you will be presented with a login prompt. Login as the root user to continue the installation.

3.2. Post-installation Setup

3.2.1. Install updates

Although not strictly required, it is a good idea to install the latest updates. You can update the packages by running the following command:

```
# yum update
```

3.2.2. Configure services

In order to avoid unexpected load spikes on your system from background services, non-essential services should be disabled. The service configuration can be modified by running:

```
# firstboot --reconfig
```

This will bring up the same configuration menu that appears after the OS has been installed. Select the option `System services`, and uncheck those services that you don't wish to start at boot time. See above in *Installing CentOS 5* on page 9 for a list of recommended services to disable.

3.2.3. Install build tools

To build the BigWorld Server, GCC and Make must be installed. These should be the default compiler and make utility on your Linux installation.

```
# yum install gcc-c++ make
```

3.2.4. MySQL server and development libraries

This step is required if you wish to use BigWorld WebConsole and StatLogger, or use DBMgr with MySQL support.

BigWorld Server is compatible with MySQL 5.0 and MySQL 5.1. Support for MySQL 4.x has been deprecated.

If you have not already installed the MySQL server, you can do so by running the following command as the root user:

```
# yum install mysql-server
```

If you intend on using DBMgr with MySQL support, also ensure you have the MySQL client development package as this is required to rebuild DbMgr with MySQL support.

Install the MySQL client development package by running the following command as the root user:

```
# yum install mysql-devel
```

3.2.5. Changing UIDs

A requirement of BWMachined is that all machines in your cluster must have the same user account information, in particular, that the numerical user IDs for a particular username are the same on every machine.

When setting up your cluster, one system in your cluster may end up with a different mapping of UID's to usernames to other systems in your cluster. This is especially if you are not using LDAP or other such tool to synchronise login names.

If you are using the GNOME desktop environment on these machines, there can be problems when you change the UID for a username. This section outlines the steps to change the UID of a username and avoid these problems.

1. Make sure no one is logged in graphically.
2. If you are in graphical mode, press `CTRL + ALT + F1` to switch to a text console.
3. Log in as `root`.
4. Choose the new user ID and group ID for your user, making sure that the new user ID is not being used by any other user, and similarly that the group ID is not used by another group. You can check by looking through `/etc/passwd` and `/etc/group`. By convention, the user's primary group ID and user ID are the same, though this need not be the case.
5. Change the user ID and group ID of the user and the user's primary group by invoking the following commands:

   ```
   # groupmod -g <new GID> <username>
   # usermod -u <new UID> -g <new GID> <username>
   ```

6. Confirm that the new user has the new UID and GID:

   ```
   # id <username>
   ```

7. Issue the commands below to remove any invalidated GNOME state:

   ```
   # rm -rf /tmp/*<username>*
   ```

8. You will now need to change the ownership of the home directory to the new UID and GID.

```
# chown -R <username>:<username> /home/<username>
```

9. Press `CTRL + ALT + F7` to return to the graphical login if required.

Chapter 4. Installing the BigWorld Server

4.1. Obtaining the Package

To obtain the BigWorld server from the SVN patch repository, use your provided username / password to access instructions on how to use the repository and upgrade releases. These can be found at https://svn01.bigworldtech.com/docs/

4.2. Installing BWMachined

BWMachined is a daemon used in the BigWorld Server environment to start and track processes, and publish information about BigWorld servers.

It must be running before any other BigWorld processes are started. It is strongly recommended to install it as a daemon, so it is always running (although this is not strictly necessary).

There are two installation methods for BWMachined. The first installation method, described in the section "BWMachined RPM Package" on page 15, uses a BWMachined RPM Package. The second installation method, described in the section "Run the Daemon Installation Script"on page 15, uses the `bwmachined2.sh` script.

For more information on how to configure BWMachined, please see the section "Configuring BWMachined" on page 16 .

4.2.1. BWMachined RPM Package

This section discusses how to install BWMachined using a BWMachined RPM package. For more information on BigWorld RPM implementation, such as how to generate the BWMachined RPM package, see the Server Operations Guide's chapter *RPM*.

BWMachined RPM package can be found in the `bigworld/tools/server/rpm/binary_rpms` directory.

To install BWMachined using a RPM package directly, run the following command as root:

```
# yum --nogpgcheck install bigworld-bwmachined-<version>-<release>.x86_64.rpm
```

where `<version>` and `<release>` are replaced by the actual version and release number.

For a large environment where BWMachined needs to be installed on many machines, we recommend that a yum repository to be set up to provide the BWMachined RPM package. This allows the BWMachined to be installed by running the yum command on a machine without the need to manually copy the BWMachined RPM package to that machine first. For more information on how to set up a yum repository, see the Server Operations Guide's chapter *RPM*.

If a yum repository has been set up and the BWMachined package is made available in the repository, then the BWMachined can be installed using the following command:

```
# yum --nogpgcheck install bigworld-bwmachined
```

4.2.2. Run the Daemon Installation Script

To install BWMachined, you need to run `bwmachined2.sh` as the root user in the `bigworld/tools/server/install` directory. For example, issue the commands below:

```
# ./bwmachined2.sh install
```

These commands will do the following:

- Stop any existing installed BWMachined daemon, and uninstall it if it does exist.
- Create the BWMachined init script in `/etc/rc.d/init.d/`.
- Create the symbolic links in `/etc/rc[1-5].d/`. It is setup to stop in `rc1.d/` and start in `rc[2-5].d/`. These can be change manually if desired.
- Copy the BWMachined executable to the directory `/usr/local/sbin/`.
- Launch the executable in daemon mode as if the init script were called with the `start` argument.

4.3. Configuring BWMachined

BWMachined plays a crucial role in the ongoing operation of your cluster environment, so it's important to understand how it can be configured and which configuration options are relevant for your server environment.

There are two configuration files which are relevant to the operation of BWMachined:

- `~/.bwmachined.conf`

  This file is used to specify options relating to how an individual user working within a BigWorld cluster should find the server resources required to operate on any available cluster machines.

- `/etc/bwmachined.conf`

  This file is primarily used to specify settings relating to how the machine that BWMachined is running on should operate within the cluster environment.

4.3.1. Creating `~/.bwmachined.conf`

When starting a server and related components there are two important pieces of information required to facilitate this:

- Where to find the BigWorld server executable files.
- Which directories contain the game resources to use with the server.

The preferred way to specify these settings is in a file `~/.bwmachined.conf`.

Note

For users not familiar with Linux:

- The ~ (tilde) indicates the user's home directory, for example a user called `johns` would generally have a home directory located at `/home/johns`
- The period character before the filename indicates it is a hidden file which causes it to not be displayed by many directory listing applications including ls (unless the `-a` option is specified).

Below is an example of a `~/.bwmachined.conf` file for a user `johns`. This file needs to be manually created when the user account is created. Note the different places where a semi-colon and a colon are used.

```
# .bwmachined.conf
# Format: BW_ROOT;BW_RES_PATH:[BW_RES_PATH] ...
/home/johns/mf;/home/johns/mf/fantasydemo/res:/home/johns/mf/bigworld/res
```

The path before the semi-colon should point to the root directory of the installed files. You will generally have the `bigworld` directory underneath this root directory. The paths after the semi-colon (separated by colon characters) specify the resource paths that will be used to find resources used by your game.

When starting a BigWorld server using the standard server tools, such as **control_cluster.py** or WebConsole, BWMachined is responsible for launching the server binaries and uses the information located in `~/.bwmachined.conf` as well as the architecture of the host system to determine how to launch the server for the requesting user.

Note

Each user that needs to run a server within your cluster environment will need a `~/.bwmachined.conf` file created and configured for them.

4.3.2. Creating `/etc/bwmachined.conf`

The global configuration file `/etc/bwmachined.conf` is used for setting options that define how BWMachined will operate on the host it is running on. For example, if you have multiple machines in your cluster, and during development you wish to isolate certain machines into groups for developer usage, this would be applied in the global configuration file. The following list provides a quick summary of host based configuration options that may be applied in the file `/etc/bwmachined.conf`:

- User defined categories
- Reviver configuration
- BigWorld server timing method
- Interface configuration for multi-interface hosts

Note

The configuration file is only read when BWMachined starts. It will have to be restarted if you want it to acknowledge your changes.

4.3.2.1. Reviver Configuration

When the Reviver process starts, it queries the local BWMachined process, and will only support the components that have an entry in a special user defined category called `[Components]`. An example configuration specifying that the Reviver should support all server components would be defined as below:

```
[Components]
baseApp
baseAppMgr
cellApp
cellAppMgr
dbMgr
loginApp
```

Note

BaseApp and CellApp will not be restarted by Reviver, the `[Components]` entries are used by the WebConsole and **control_cluster.py** to determine which processes should be started by BWMachined on that host. This list however is only a hint for the server tools and the processes may still be started on that host if required.

Note

If the `[Components]` category does not contain any entries, then Reviver will support all server components.

4.3.2.2. Timing Method

By default, time services is provided by the clock_gettime system call. The default is the recommended timing method and works for all supported platforms, for further options, please refer to *Clock*.

4.3.2.3. Machine groups

The membership of the machine running BWMachined can also be optionally specified in `/etc/bwmachined.conf`. For more details on what machine groups are used for, and how to specify them, refer to the *Machine Groups and Categories*.

4.3.2.4. Internal Interface Configuration for Multi-interface Hosts

The BigWorld machine daemon is used for both internal machines and out-facing machines such as those that BaseApps and LoginApps are run on. The protocol that BigWorld components use for discovery of server processes, and process startup registry involves a UDP broadcast to the machine daemons.

The machine daemon must determine the interface to receive these broadcasts from. By default, the machine daemon will determine which interface is the internal interface by sending a broadcast packet on each interface and waiting for this broadcast packet to be returned. The interface which receives the first broadcast packet is assumed to be the internal network.

For the out-facing machines with more than one interface, this may result in the incorrect interface being chosen. In these situations, it is best to check your broadcast routing rules[1] and consider adding firewall rules to block receiving from bwmachined ports (20018 and 20019) on these other interfaces.

For example, if `eth1` is not your internal interface:

```
# /sbin/iptables -I INPUT 1 -p udp -i eth1 -m multiport --destination-ports
 20018:20019 -j DROP
```

Note

In most circumstances it is better to block all ports and then only open the ports that are needed. See “Security” on page 23 .

In rare cases, the `[InternalInterface]` configuration option can be set with either the dotted-decimal address or the name of the interface that is connected to the internal network.

For example, using the name of the interface:

```
[InternalInterface]
eth0
```

Using dotted-decimal notation:

```
[InternalInterface]
```

[1] For more details, see “Routing” on page 24 .

```
192.168.0.1
```

If this option is not specified, auto-discovery of the internal interface will be performed. If the option is specified, but no interface is found that matches the value set in `[InternalInterface]`, then an error will be logged to syslog and the machine daemon process will terminate.

This option may now also be specified in the `[bwmachined]` section of `/etc/bigworld.conf` as follows:

```
[bwmachined]
internal_interface = <value>
```

4.4. Check That BWMachined is Running

- To check if the daemon is running, issue the command below:

  ```
  # ps -C bwmachined2
  ```

  The output should be one line listing the details of the daemon process.

- Another way to verify if the daemon is running, is to run the **control_cluster.py**[2] utility. To use **control_cluster.py** to check the status of BWMachined, issue the following command:

  ```
  $ bigworld/tools/server/control_cluster.py cinfo
  ```

  The machines correctly running BWMachined should be displayed, as in the example below:

  ```
  shire       10.40.3.37      0 processes   0%, 0% of 2000MHz (4% mem)
  ```

  Make sure that the process has an address that is on the internal network. If it does not, make sure that your broadcast route is set correctly.

  If your machine is not listed, but has BWMachined running, then you might need to check your firewall rules. For more details, see section “Security” on page 23 in this document.

  You can also run the following command to check the relationship between bwmachined processes in the cluster.

  ```
  $ bigworld/tools/server/control_cluster.py checkring
  ```

4.5. Configure MySQL Server

This step is required if you wish to use BigWorld WebConsole and StatLogger, or use DBMgr with MySQL support.

At this point it is assumed you have installed the MySQL server and the MySQL client development packages. See “Post-installation Setup” on page 11 for more information.

- The MySQL daemon is not configured to run by default in CentOS. You can check whether or not it is running by issuing the following command as the root user:

  ```
  # /sbin/service mysqld status
  ```

If (as expected) the MySQL daemon is not running, you can manually start it by running the following command as the root user:

```
# /sbin/service mysqld start
```

To ensure that MySQL starts automatically each time you boot the machine, you can use the **chkconfig** utility. To see what runlevels[3] MySQL is enabled for, run the following command as the root user:

```
# /sbin/chkconfig --list mysqld
```

If MySQL is disabled for all runlevels, you can enable it to the network-enabled runlevels by running the following command as the root user:

```
# /sbin/chkconfig --level 35 mysqld on
```

- The default MySQL installation is configured with one user called `root`. This is not the same as the system root user. Additional users may be created using the **mysql** command-line utility. A user is required for DBMgr's database and for the WebConsole and StatLogger server tools. In order to create a new user, log in to MySQL as the `root` user:

  ```
  $ mysql -u root
  ```

  Then create an additional user with the required privileges. For example:

  ```
  mysql> GRANT SELECT, INSERT, UPDATE, DELETE, ALTER, CREATE, DROP, INDEX ON
   fantasydemo.* TO 'bigworld'@'localhost' IDENTIFIED BY 'bigworld_passwd';
  mysql> GRANT SELECT, INSERT, UPDATE, DELETE, ALTER, CREATE, DROP, INDEX ON
   fantasydemo.* TO 'bigworld'@'%' IDENTIFIED BY 'bigworld_passwd';
  mysql> GRANT RELOAD ON *.* TO 'bigworld'@'localhost' IDENTIFIED BY
   'bigworld_passwd';
  mysql> GRANT RELOAD ON *.* TO 'bigworld'@'%' IDENTIFIED BY
   'bigworld_passwd';
  ```

  The above commands create a user called `bigworld` with a password of `bigworld_passwd` that has `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `ALTER`, `CREATE`, `DROP`, `INDEX` and `RELOAD` privileges on the fantasydemo database.

  You can restrict the machines from which the user can connect by specifying a more restrictive pattern after the `@` symbol.

  For more information on MySQL account management, please refer to the MySQL online documentation, specifically the *MySQL Server Administration* document.

- Create a new game database using the **mysql** command-line utility. For example, for FantasyDemo:

  ```
  mysql> CREATE DATABASE fantasydemo;
  ```

  This must match the database connection details specified in your game's `bw.xml` configuration file.

- The DBMgr binary provided in your package does not have MySQL support built-in. You will need to rebuild DBMgr with MySQL support enabled. Please refer to the Server Operations Guide's section

"Compiling DBMgr with MySQL Support" for instructions on how to rebuild and configure your game environment for MySQL support.

4.5.1. MySQL Tweaks

MySQL documentation suggests that the `innodb_buffer_pool_size` system variable can be set to 80% of the physical memory size.

4.6. Installing Server Tools

A suite of utilities collectively referred to as *Server Tools* are provided to monitor and query BigWorld Server status, both live and post event. These tools include WebConsole, MessageLogger and StatLogger.

A typical deployment of the server tools is to install one instance of the WebConsole, MessageLogger and StatLogger in the network. The WebConsole can be used to control multiple BigWorld servers. Each server is running under different Unix account. A WebConsole account will need to be created for each user (e.g. server operator / developer) that will control a BigWorld server in the network. Each Unix user can only run one BigWorld server at a time.

To install these server tools, refer to the Server Tools Installation Guide.

Chapter 5. Cluster Configuration

5.1. Security

In a BigWorld Server cluster, not all machines are connected to the public internet, but those that are need to be secured well.

This is most easily achieved by using a firewall to block incoming packets. In general, the approach for machines with public IP addresses should be to block all incoming packets on the interface with the public IP.

The exception is that LoginApps and BaseApps need to allow UDP traffic to the ports they listen on. The ports to be used are defined in the `res/server/bw.xml` file using the options `loginApp/externalPorts/port` and `baseApp/externalPorts/port`. For details on these options, see the Server Operations Guide's chapter *Server Configuration with `bw.xml`*, sections "BaseApp Configuration Options" and "LoginApp Configuration Options".

Using the Linux firewall configuration tool, **iptables**, we can add a rule to drop all incoming traffic on the external interface. In the following examples, we assume the external interface is `eth1`. For example:

```
# /sbin/iptables -A INPUT -i eth1 -j DROP
```

On the machines running LoginApps, we can add a rule to allow traffic through on the login port, using the default port of `20013`, as illustrated below:

```
# /sbin/iptables -I INPUT 1 -p udp -i eth1 --destination-port 20013 -j ACCEPT
```

Similarly, for machines running BaseApps, we add similar rules to allow traffic through on the BaseApp external port as specified in the `baseApp/externalPorts/port` options.

We use `-I INPUT 1` in the new rule instead of `-A INPUT` because **iptables** applies the first rule in the chain that matches an incoming packet. Therefore, we need to insert the rule for accepting login packets before the rule for rejecting all UDP traffic on `eth1`.

For a production server, you should disable all networking services apart from SSH from trusted IP addresses.

BWMachined requires the ability to broadcast on the internal interface and receive back its own replies on UDP ports `20018` and `20019`. The internal interface is denoted here as `eth0`. The firewall rules should accommodate this requirement.

For example:

```
# /sbin/iptables -I INPUT 1 -p udp -i eth0 -m multiport --destination-ports
 20018:20019 -j ACCEPT
```

For details about the ports used by a BigWorld server see *Security*.

You must ensure that the firewall rules are restored each time the machine boots. You can use the following command to save the **iptables** configuration:

```
# /etc/init.d/iptables save
```

5.2. Routing

A large number of tools and server components in BigWorld Technology rely on being able to send IP broadcast packets to the default broadcast address (255.255.255.255), and for them to be routed correctly.

This will happen by default on a machine with only one network interface (i.e., machines on the internal network only, such as CellApp machines, DBMgr machines, etc...).

For machines with two network interfaces (i.e., BaseApp and LoginApp machines), we need to make sure that packets sent to the broadcast address are routed via the internal interface.

We can make sure this is done correctly by making an entry in the kernel routing table with the **ip** command. This command may not be installed by default. You can install this utility running the following command as root:

```
# yum install iproute
```

In the example below, we once again assume that `eth0` interface is the internal network. To add a default broadcast route, run the following command as the root user:

```
# /sbin/ip route add broadcast 255.255.255.255 dev eth0
```

This command however will only add the route to the current routing table, and will not apply after rebooting your machine. In order to ensure this route is applied whenever the `eth0` interface is brought online run the following command as the root user:

```
# echo "broadcast 255.255.255.255 dev eth0" >
 /etc/sysconfig/network-scripts/route-eth0
```

This command will create the file `/etc/sysconfig/network-scripts/route-eth0` if it doesn't already exist.

5.3. Disabling cron jobs

Cron is a system daemon which enables tasks to be scheduled for running at pre-determined intervals, such as hourly, daily, weekly, etc. Cron refers to these periodic tasks as *jobs*. These jobs may adversely affect the performance of a running server due to the the kind of functionality they perform. For example it is quite common for cron jobs to update the **locate** database which involves performing a recursive directory listing on the entire machine. These kinds of jobs can involve reading from each part of a hard drive, effectively causing a flush of the disk cache Linux has in memory. This can cause a momentary lapse in performance of server processes, as disk swapping starts to occur on the host machine.

We recommend that BigWorld Server machines have resource-intensive (CPU, memory or disk) cron jobs disabled in production environments. You can achieve this (with various levels of granularity) by disabling these cron jobs.

Cron jobs can be disabled by clearing the executable bit on the relevant job. For example, to disable the job run by `/etc/cron.daily/makewhatis.cron`:

```
# chmod -x /etc/cron.daily/makewhatis.cron
```

Re-enabling cronjobs can be done by setting the executable bit using the reverse operation:

```
# chmod +x /etc/cron.daily/makewhatis.cron
```

- System cron jobs are stored in the following locations:
  - `/etc/cron.d` (contains cron jobs for system services)
  - `/etc/cron.hourly` (for hourly cron job scripts)
  - `/etc/cron.daily` (for daily cron job scripts)
  - `/etc/cron.weekly` (for weekly cron job scripts)
  - `/etc/cron.monthly` (for monthly cron job scripts)

Also remove any unnecessary user-level cron jobs. These can be listed per-user using:

```
$ crontab -l
```

Note

We do not recommend completely disabling the cron service as facilities such as log rotation and some security mechanisms may rely on the cron service being active.

Chapter 6. Development Environment Configuration

The server side development environment requires a little bit of configuration to enable fluid development. This chapter aims to instruct the reader on how to successfully configure the server environment.

6.1. Setting up shared resources

Developing a game using BigWorld inevitably means managing code and assets developed by a number of people on both Windows and Linux machines. With this in mind BigWorld has a set of recommendations for working in this kind of environment. Please refer to the Server Programming Guide section *Shared Development Environments* for more information.

6.2. Configure bwmachined

Each user wishing to run a server must have a valid BWMachined configuration script which points to the server binaries and game resources to be used when starting the server. A script called **bw_configure** is provided to assist in this process. When installing the server using RPM packages, this script should be automatically placed in your environment path and available for use.

Note

If this is your first time installing and running a BigWorld server and are uncertain of what to input here, please refer to the Tutorial chapter *A basic client-server game (`CLIENT_SERVER`)*, which runs through this process step by step.

To create or update your bwmachined configuration file (`$HOME/.bwmachined.conf`) simply run the script as follows:

```
$ bw_configure
Game resource path [~/my_game/res]: ~/mf/my_game/res
Writing to /home/alice/.bwmachined.conf succeeded

Installation root : /opt/bigworld/current/server
BigWorld resources: /opt/bigworld/current/server/res
Game resources    : /home/alice/mf/my_game/res
```