Configuring, Tuning, and Using Oracle9i in a Red Hat Linux Environment

Edward Whalen
November 2002

In the past year the success of Linux has grown, even in tough economic times. Support from hardware vendors such as Dell, IBM and HP have helped its popularity, but without the support from application vendors such as Oracle, Linux would not make it. Fortunately for all of us, Oracle has taken on Linux with enthusiasm and commitment. It is this commitment from Oracle and others that will allow Linux to become a significant database platform.

Background

Linux is very similar to the UNIX operating system, but due to several factors, Linux is not the same as UNIX. Let's look at a brief history of UNIX and Linux.

A Brief History of Linux and UNIX

- 1969 – AT&T Bell Labs - Ken Thompson & Dennis Ritchie started development using a PDP-7
- 1971 – 1st edition is released
- 1973 – 4th edition rewritten in "C"
- 1975 – 6th edition released outside of Bell Labs
- 1980 – Microsoft releases Xenix
- 1982 – AT&T's Unix System Group (USG) releases System III
- 1983 – UNIX System V, the first supported UNIX is released
- 1984 – University of California Berkely releases 4.2BSD
- 1989 – UNIX system V release 4 is released, unifying System V, BSD & Xenix
- 1991 – USL (Unix Systems Lab) is spun off
- 1991 – Linus Torvalds begins work on Linux at the University of Helsinki in Finland and the first version was released
- 1993 – Novell buys USL
- 1993 – Novell transfers rights to UNIX trademark and specification to X/Open
- 1999 – Linux version 2.2 kernel released
- 2001 – Linux version 2.4 kernel released

Linux is very similar to UNIX in several ways. Let's look at some of the attributes of Linux.

Operating system concepts

- Linux is an operating system with the following attributes
    - Multi-user
    - Multi-tasking
    - Virtual Memory system

    - X-Windows
    - Compatible with the IEEE POSIX.1 standard
    - Non-proprietary source code
    - Source code is available
- Linux is based on a microkernel architecture (similar to Windows 2000)
- Linux is multi-tasking
- Linux is a virtual-memory OS
- Drivers can be dynamically loaded/unloaded
- Some kernel parameters can be modified on the fly
- Each user logs into their own environment with a username and password
- User programs cannot modify sensitive system information (hence less crashes)
- Each user may be given permission to access directories other than their own

In many ways Linux is very similar to UNIX. In other ways Linux is different from UNIX. A great deal of growth in the Linux market has come in the commodity hardware space, i.e. i386 PCs. Because of this, the major Linux brands that run on PC hardware supports a large number of different hardware components. This is different from many of the proprietary UNIX brands that only support that company's hardware.

In fact, when one compares Linux to UNIX one might say:

It looks like a duck.
It walks like a duck.
It quacks like a duck.
But, it's a penguin.

Because of the variety of hardware and the pervasiveness of Linux, the install and management process has been assisted by a number of tools and utilities. There is a tendency for a large number of end users to administer their own systems rather than relying on the system administrator, therefore many of the administrative tasks can be done via a GUI.

Configuring Linux Systems for Oracle

The installation documentation that Oracle provides is very good, but must be carefully followed. There are several steps that must be taken prior to installing Oracle in order for the installation process to be successful. I myself have had to repeat the installation process because I forgot a specific step. Documenting the problems you have and the solution to those problems can save you time in the future.

Install the Necessary Components

The Oracle installation process installs object libraries and then links those object libraries into executables based on the options that you have selected. Therefore it is necessary to install the development tools for Linux. Without these tools you will not be able to install Oracle. Unfortunately whereas the Linux installation allows you to select development tools as part of it, in order to install the development kit after installation

requires you to know all of the individual components and to install those components using RPM.

Perform Pre-Installation Tasks

As part of the installation process it is necessary to create an Oracle account. In addition you must set up several groups and tune the kernel in order to have sufficient shared memory support for the Oracle instance. Tuning Linux is covered later.

Oracle RAC on Linux

Oracle Real Application Clusters (RAC) is the follow-on product to Oracle Parallel Server (OPS). RAC allows multiple instances to access the same database simultaneously. RAC provides both fault tolerance and performance benefits by allowing the system to scale out, and at the same time since all nodes access the same database, the failure of one instance will not cause the loss of access to the database.

In order for RAC to work you must have a shared disk subsystem. All nodes in the cluster must be able to access all of the data, redo log files, control files and parameter files for all nodes in the cluster. The data disks must be globally available in order to allow all nodes to access the database. Each node has its own redo log and control files, but the other nodes must be able to access them in order to recover that node in the event of a system failure.

Probably the biggest difference between RAC and OPS is the addition of *Cache Fusion*. With OPS a request for data from one node to another required the data to be written to disk first, then the requesting node can read that data. With cache fusion, data is passed along with locks.

Oracle RAC solutions are available from vendors such as Dell, IBM and HP. You can even put together your own RAC system for development and testing by using a SCSI or other low cost shared disk solution.

Tuning Linux

Tuning Linux can be done via sever different methods, some of which are easy and another that is very difficult. The first method involves echoing values to the proper /proc file that controls the tuning parameter. The second involves using the /etc/sysctl.conf file and the third involves relinking the kernel.

Echoing a value to the proper /proc pseudo-file will dynamically change the current running value of that parameter. This change will take place immediately and is very easy to use. These changes are dynamic, but are only valid until the system is rebooted. In order to have these changes take place each time the OS reboots you can place these echo commands into the /etc/rc.local file. An example is shown here:

```
echo 2147483649 > /proc/sys/kernel/shmmax
echo 1024 65000 > /proc/sys/net/ipv4/ip_local_port_range
```

In addition, these parameter values can be put into the /etc/sysclt.conf file. The syntax is slightly different, but the effect is the same. Here is an example of the /etc/sysctl.conf file.

```
# Disables packet forwarding
net.ipv4.ip_forward = 0
# Enables source route verification
net.ipv4.conf.default.rp_filter = 1
# Enables the magic-sysrq key
kernel.sysrq = 1
kernel.sem=250 32000 100 128
kernel.shmmax=2147483648
kernel.shmmni=4096
fs.file-max=409200
```

This is also a very easy and convenient method of setting the kernel tuning parameters to be set each time the OS is booted.

The final method is to rebuild the Linux kernel. This is only necessary for a few parameters such as the parameter to change the SCSI timeout value. This is typically not necessary and is not an easy task. I do not recommend rebuilding the Linux kernel unless it is absolutely necessary.

Monitoring Linux

There are several methods of monitoring the Linux operating system. You can view OS data from within the /proc pseudo-filesystem, you can use the Linux tools that are provided or you can use third party tools. Which method you choose is based on your personal preference.

Viewing OS statistics via the /proc pseudo filesystem is fairly difficult since you are only provided with values that must be interpreted. There are a number of files within this pseudo-filesystem that you can view, such as /proc/sysinfo, /proc/meminfo, /proc/partitions, etc. An example of viewing /proc/meminfo is shown here:

```
[oracle@localhost 2500w]$ cat /proc/meminfo
        total:    used:    free:  shared: buffers:  cached:
Mem:  4022009856 4016812032  5197824 220852224 57868288
3519877120
Swap: 4342824960        0 4342824960
MemTotal:      3927744 kB
MemFree:          5076 kB
..
..
LowTotal:       847616 kB
LowFree:          3048 kB
SwapTotal:     4241040 kB
SwapFree:      4241040 kB
```

```
BigPagesFree:         0 kB
```

This provides you with a lot of good information, but is not in a very pleasing format. There are several Linux commands that can be used to view performance data. This includes top, vmstat, iostat and free. These applications read data out of the /proc pseudo-filesystem and display them in a more human readable format. These applications also update the data regularly. Let's look at an example of top.

```
11:41am  up 32 min,  2 users,  load average: 1.40, 0.98, 0.59
89 processes: 87 sleeping, 2 running, 0 zombie, 0 stopped
CPU0 states: 13.1% user, 33.3% system,  0.0% nice, 53.0% idle
CPU1 states:  0.0% user,  6.4% system,  0.0% nice, 93.0% idle
CPU2 states:  0.0% user,  0.4% system,  0.0% nice, 99.1% idle
CPU3 states:  0.1% user,  1.1% system,  0.0% nice, 98.2% idle
Mem:  3927744K av, 3922648K used,    5096K free,  216432K shrd,   59680K buff
Swap: 4241040K av,       0K used, 4241040K free                3453976K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
 1515 oracle    17   0 23344  22M 17404 R    46.7  0.5   1:22 oracle
   12 root      15   0     0    0     0 SW    5.7  0.0   0:36 bdflush
 1522 oracle    15   0  1072 1072   840 R     0.5  0.0   0:00 top
    1 root      15   0   512  512   444 S     0.0  0.0   0:04 init
    2 root      15   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root      15   0     0    0     0 SW    0.0  0.0   0:00 keventd
    4 root      15   0     0    0     0 SW    0.0  0.0   0:00 keventd
    5 root      15   0     0    0     0 SW    0.0  0.0   0:00 keventd
    6 root      34  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    7 root      34  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU1
    8 root      34  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU2
    9 root      34  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU3
   10 root      15   0     0    0     0 SW    0.0  0.0   0:22 kswapd
   11 root      16   0     0    0     0 SW    0.0  0.0   0:00 kreclaimd
   13 root      15   0     0    0     0 SW    0.0  0.0   0:00 kupdated
```

Top includes both overall OS information and process information. The process information is similar to that which you would get from sar and the process information is a sorted representation of data that you would get from the ps command.

In addition to the data that you can get from the Linux tools, there are a number of very good third party tools such as Bradmark's NORAD surveillance that can be used to view Linux OS data.

Issues and Gotchas

There are a number of issues that can affect the installation and configuration of Oracle9i on Linux. The most common issues are listed here:

Sufficient Temp Space	/tmp must be at least 400M, if you don't have sufficient tmp space configured, Oracle will not be able to complete the installation process. Remember that /tmp might be configured to be part of the / filesystem.
Sufficient Swap Space	There must be at least 1GB of swap space for the Oracle installation process to work. With less than 1GB of swap space the installation might fail.

Dev Environment	Oracle will install, but will fail to link without the Linux development system. This is a problem that I have run into occasionally. If you choose Advanced Server during the Linux install, and not custom, the development environment won't be installed.

Conclusions

In conclusion, I feel that the Linux Operating System is a stable and robust platform for running Oracle. It is becoming much more popular, and many vendors are supporting it. It is easy to work with, easy to tune, and performs well. In addition, we have configured a number of Oracle RAC systems on Linux with great success. You can have confidence in running Oracle on Linux.