RAW PARTITIONS ON A UNIX SYSTEM - OR - DON'T BE AFRAID OF GOING RAW

Mary Ann Hale and Irma Fisher *
Internal Revenue Service

Abstract

This paper will explore using raw partitions on Unix systems for Oracle database files. We will discuss how Unix accesses the raw device and how Oracle uses the raw disk partition.

A step by step approach will be used to illustrate setting up raw partitions on a Unix system. Effected system files will be discussed and examples will be provided. This discussion will include converting from Unix file systems for ORACLE database files to using raw partitions. We will compare and contrast the use of both types of files.

We will discuss system backup procedures, changing buffer cache parameters and performance issues.

Background - The File System

The Unix File System consists of:

ordinary files	no particular structuring is expected by the system
directories	induces a structure on the file system
special files	each supported I/O device is associated with at least one such file. They are read and written just like ordinary disk files, but requests to read or write result in activation of the associated device. An entry for each special file resides in directory /dev. Special files exist for each communication line, each disk, [each disk partition], each tape drive, and for physical main memory.

The advantages to this structure are:

1. file and device I/O are as similar as possible
2. file and device names have the same syntax and meaning (programs expecting a file name as a parameter can be passed a device name)
3. special files are subject to the same protection mechanism as regular files (Unix permissions).

Although the root of the file system is always stored on the same device, it is not necessary that the entire file system hierarchy reside on this device. The "mount" system request causes references to an ordinary file to refer instead to the root directory of the file system on the new volume. "Mount" replaces a leaf of the hierarchy tree (the ordinary file) by a new subtree (the hierarchy stored on the removable volume). After the "mount", there is no distinction between files on the new volume and those in the permanent file system.

The system calls to do I/O are designed to eliminate the differences between the various devices and styles of access. There is no distinction between "random" and "sequential" I/O, nor is any logical record size imposed by the system.

Device Special Files

A directory entry contains only a name for the associated file and a pointer to the file itself. This pointer is an integer called the i-number of the file. When the file is accessed, its i-number is used as an index into a system table (the i-list) stored in a known part of the device on which the directory resides. The entry found thereby (the file's i-node) contains the description of the file:

- the user and group ID of its owner
- its protection bits
- the physical disk or tape addresses for the file contents
- its size
- time of creation, last use, and last modification
- the number of links
- a code indicating whether the file is a directory, an ordinary file, or a special file

*This paper was originally presented at the 1992 International Oracle User Week Conference in San Francisco, California, September 14-18, 1992.

When an I/O request is made to a file whose i-node indicates that it is a special file, the first device address word specifies an internal device name, which is interpreted as a pair of numbers representing, respectively, a device type and subdevice number. The device type indicates which system routine will deal with I/O on that device; the subdevice number selects, for example, a disk drive attached to a particular controller.

Figure 1
Example of Device Special Files

```
crw------- 1 root   56,  0 Apr 5 10:06 /dev/rrz0a
crw------- 1 root   56,  1 May 2 06:30 /dev/rrz0b
crw------- 1 root   56,  2 Apr 5 10:06 /dev/rrz0c
crw------- 1 root   56,  3 Apr 5 10:06 /dev/rrz0d
crw------- 1 root   56,  4 Apr 5 10:06 /dev/rrz0e
crw------- 1 root   56,  5 Apr 5 10:06 /dev/rrz0f
crw------- 1 root   56,  6 May 2 06:24 /dev/rrz0g
crw------- 1 root   56,  7 Apr 5 10:06 /dev/rrz0h
brw------- 1 root   21,  0 Apr 5 10:06 /dev/rz0a
brw------- 1 root   21,  1 Apr 5 10:06 /dev/rz0b
brw------- 1 root   21,  2 Apr 5 10:06 /dev/rz0c
brw------- 1 root   21,  3 Apr 5 10:06 /dev/rz0d
brw------- 1 root   21,  4 Apr 5 10:06 /dev/rz0e
brw------- 1 root   21,  5 Apr 5 10:06 /dev/rz0f
brw------- 1 root   21,  6 Apr 5 10:06 /dev/rz0g
brw------- 1 root   21,  7 Apr 5 10:06 /dev/rz0h
```

What is a Raw Partition?

RE: 4.2 BSD (Berkeley Software Distribution)

The file system organization partitions the disk into one or more areas called cylinder groups. A cylinder group is comprised of one or more consecutive cylinders on a disk. Associated with each cylinder group is some bookkeeping information such as a redundant copy of the super-block, space for inodes, and a bit map describing available blocks in the cylinder group. A static number of inodes is also allocated for each group when the file system is created.

The UNIX* system provides various levels of interfaces to the hardware mass storage system. At the lowest level is a raw disk. This interface provides access to the disk as a linear array of sectors. This interface is used by programs that need to do disk to disk copies or that wish to dump file systems. User programs with proper access rights can also access this interface. The raw devices have names like "/dev/rrz#a", the "r" standing for "raw".

A disk is usually formatted with a file system that in interpreted by the UNIX system to provide a directory hierarchy and files. The block level interface is used for this level. The disks with the names "/dev/rz#a", etc are block devices. The UNIX system interprets and multiplexes requests from user programs to create, read, write, and delete files by allocating and freeing inodes and data blocks.

File system input/output is buffered by the kernel. The data is transferred from the disk into buffers in the kernel address space and then 40% of the processor cycles are spent copying these buffers to user address space. Each UNIX* physical disk is divided into a [finite] number of disk partitions, each of which may occupy any consecutive cylinder range on the physical device. The cylinders occupied by the partitions for each drive type are specified in the disk description file /etc/disktab.

Figure 2
Example of /etc/disktab File

```
rz57|RZ57|DEC RZ57 Winchester:\
     :ty=winchester:ns#71:nt#15:nc#1925:\
     :pa#32768:ba#8192:fa#1024:\
     :pb#184320:bb#4096:fb#1024:\
     :pc#2025788:bc#8192:fc#1024:\
     :pd#299008:bd#8192:fd#1024:\
     :pe#299008:be#8192:fe#1024:\
     :pf#596284:bf#8192:ff#1024:\
        :pg#614400:bg#8192:fg#1024:\
        :ph#1194300:bh#8192:fh#1024:
```

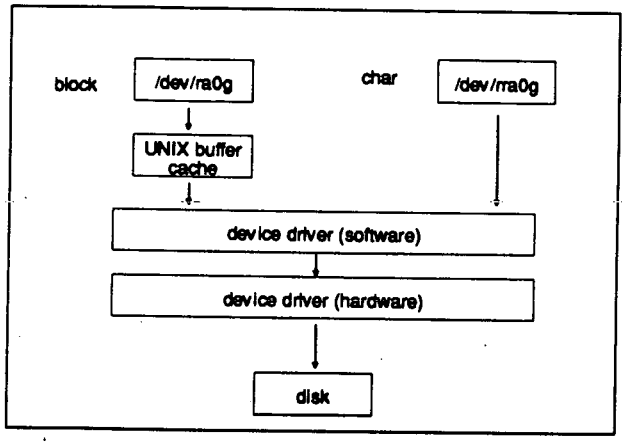
Each partition may be used for either a raw data area such as a paging [swap] area, [or raw data area], or to store a UNIX file system. It is conventional for the first partition on a disk to be used to store a root file system, from which UNIX may be bootstrapped. The second partition is traditionally used as a paging [swap] area, and the rest of the disk is divided into spaces for additional "mounted file systems" [or raw partitions]. The third logical partition of each physical disk also has a conventional usage: it allows access to the entire physical device, including the bad sector forwarding information recorded at the end of the disk. It is occasionally used to store a single large file system, [a raw data area], or to access the entire pack when making a copy of it on another. Care must be taken to not overwrite the last few tracks and thereby clobber the bad sector information.

How Do I Access a Raw Partition?

Raw partitions are accessed via device drivers. A

device driver is a software program that translates requests to a block address. The requests are then turned over to the disk controller which does the physical access.

Figure 3.
Accessing the Disk



Introduction

Now that you have an understanding of the UNIX file system, character special files, what a raw partition is and how it is accessed, you are ready to implement the use of raw partitions for ORACLE:

Why Use Raw Partitions for ORACLE?

Raw (character) devices are not buffered by the Unix kernel. Data is transferred directly from the SGA (Oracle buffer cache) to the disk. Avoiding the UNIX kernel buffers will eliminate the overhead associated with them. This results in an increase in performance.

How Does ORACLE Use a Raw Disk Partition?

Oracle can use raw disk partitions anywhere a file specification is necessary. The file specification may identify a logfile or datafile. The raw partition may be used for any tablespace including the system tablespace. Use the character special file to identify the raw partition to use.

The Role of Your System Administrator.

The System Administrator of your system plays a major role in enabling you to use raw partitions for Oracle. They know your system. They know how it is set up, what disk drives are installed, how they are configured, and what disk partitions are mounted as UNIX file systems. They will help you determine the character devices that identify the available raw partitions. The first few steps in setting up raw partitions for Oracle use requires superuser privileges. Since the System Administrator has the ability to login as superuser, you need their assistance. Working as a team, you and the System Administrator can determine the best combination of UNIX file system files and raw partitions for your site.

Steps to Implement Raw Partitions

1. Determine the available raw partitions.

   Work with your System Administrator to determine the raw partitions that are available. The first 907K on the disk and the last 1782K of the drive contain information about that drive and bad block replacements. Care must be taken to ensure you do not overwrite this information. Sizes depend on the disk configuration and may differ for your site. Check the documentation for your specific disk. If necessary, change the disk table to work around any conflicts.

   You may also want to have the System Administrator create customized disk partitions for you. Depending on the release of UNIX you are running, you may have from 1 - 255 disk partitions on any drive. DYNIX/ptx provides the ability to customize up to 255 partitions.

2. Oracle must own the character special file.

   Make 'oracle' the owner of the raw disk partition. This is the character special file. Use the chown command.

   ```
   chown oracle /dev/rrz0g
   ```

3. Restrict access.

   Restrict the permissions of the character special file for 'oracle' use only. Use the chmod command

   ```
   chmod 600 /dev/rrz0g
   ```

4. Determine the size of the raw partition.

   The size can be computed by using the sector information available in the /etc/disktab file. Referring to the /etc/disktab in figure 2, for partition 'g' you can compute the size as follows:

   ```
   partition sector size (pg) = 614400
   sector / 2 = size partition in bytes
   614400 / 2 = 307200
   ```

   The size used for the datafile of the raw partition must be at least two Oracle blocks smaller than the size of the actual raw

partition computed above. You may specify any size less than the actual size, but space not used by the datafile is not accessible for other purposes. Computing the size for Oracle does not have to be an exact science. Try to estimate the size as close to the complete partition as possible.

5. Identify the raw partition to Oracle.

   Specify the raw partition in the appropriate SQL statement.

```
create database abc
   logfile '?/dbs/log1.dbf' size 100K,
           '?/dbs/log2.dbf' size 100K
   datafile '/dev/rra0g' size 295M reuse;

create tablespace tbsp1
datafile '/dev/rra0g' size 295M reuse;

alter tablespace tbsp1
add datafile '/dev/rra0g' size 295M reuse;

(i.e. 295M = 304087000 bytes)
```

   You can use any combination of raw partition and file system files for a tablespace.

6. Change your backup procedure.

   The system utilities dump and restore are not valid for backing up raw disk partitions. I recommend using the 'dd' command. Experiment with different blocksizes to get the best performance. We use the command:

```
dd if=/dev/rra0g of=/dev/rmt1h bs=20K
if = input file (raw partition)
of = output file (9-track tape)
bs = input/output block size
```

   Some releases of the 'dd' command do not support multi-volume output. If the size of the raw partition will not fit on one tape, you can write to multiple tapes by specifying the count and skip parameters.

7. Change the Buffer Caches.

   Since the use of raw partitions bypass the UNIX buffer cache, you need to adjust the size of your buffer caches. The Oracle buffer cache stores the database buffers for your Oracle data. The UNIX buffer cache hold blocks of file system data. If you use more raw partitions for your datafiles than file systems, you should increase the Oracle buffer cache and decrease the UNIX buffer cache. In tuning either cache you want the block of data needed to be found in the cache, thus eliminating the need for disk I/O.

   To increase the Oracle buffer cache, start with the db_block_buffers. You can determine your hit ratio by using the following query.

```
select 'Hit Ratio' Cache,
    round (((( sum (decode (statistic#,26,value,0)) +
          sum (decode (statistic#,27,value,0)) -
          sum (decode (statistic#,28,value,0)) ) /
         ( sum (decode (statistic#,26,value,0)) +
          sum (decode (statistic#,27,value,0)) )) *
         100,2) Value
    from v$sysstat;
```

   To decrease the UNIX buffer cache you need to adjust the operating system parameter that configures the buffers (NBUF for DYNIX/ptx or bufcache for Ultrix). Be cautious not to cause excessive paging and swapping by increasing your caches too much. Check the system with the operating system commands that provide information about swapping, paging and disk activity. A few suggested commands are:

```
BSD Unix ---    vmstat -S -- memory status
        including swapping,
        iostat -- input/output statistics;
ATT UNIX ---    sar -b -- cache hit ratios,
        sar -w -- swap,
        sar -p -- paging.
```

Changing from UNIX file system files to ORACLE raw partition files.

If you want to change your UNIX file system to an Oracle raw partition follow these steps:

1. move the data contained in the mounted file system elsewhere
2. unmount the file system
3. perform steps 2 and 3 above referencing the raw partition
4. take the entry for that device out of the /etc/fstab file
5. specify the raw partition in the appropriate SQL statement.

Performance Issues

Performance of your system should increase with the use of raw partitions. This increase is effected by many factors. These include the efficiency of your hardware, how your data is distributed on your disk drives, and processes running concurrently with ORACLE.

On-Going Notes

When your operating system is upgraded the special files may be overwritten. If this happens you need to change the owner and permission on the character special files as describes in step 2 and 3 above.

Conclusion

Once you have an understanding of the UNIX file system, character special files, what a raw partition is and how it is accessed, the task of implementing raw partitions for Oracle database files is efficient and an effective use of available resources. Oracle provides many options when configuring a database. Using raw partitions is an option that is rewarding and well worth looking into.

Bibliography

"A Fast File System for UNIX*" by McKusick, Joy, Leffler, Fabry, 1983

"The UNIX* Time-Sharing System" by D.M. Ritchie and K.Thompson, 1974

"Installing and Operating 4.2BSD on the VAX^" by Leffler and Joy, 1983

References

- ORACLE Installation and User's Guide V6.0.30
- ORACLE SQL*Language Reference Manual V6.0