How much space is there?

This is one of the questions that is most frequently asked of capacity management people. While it is easy to give a simple answer, the information provided is often misleading. Here we'll see that there are at least three ways of approaching the question – the answer you want will depend on what your interests are.

1.) Filesystems

Take for example a production server that runs an Oracle database. The first answer to the above question is obtained by running the *df* command[1] on the machine in question, for example:

```
p137-perfmon: bdf
Filesystem              kbytes      used     avail %used  Mounted on
/dev/vg00/lvol3         143360    129123     13384  91%   /
/dev/vg00/lvol1          83733     44013     31346  58%   /stand
/dev/vg00/lvol8        2560000   1479135   1013952  59%   /var
/dev/vg00/lvol7        1536000    712994    771578  48%   /usr
/dev/vg00/lvol9        1048576     84534    904486   9%   /usr/openv
/dev/vg00/lvol6          65536      3138     60018   5%   /tmp
/dev/vg00/lvol5        1024000    541920    451991  55%   /opt
/dev/vg00/lvol14        409600     64321    327913  16%   /opt/ultracomp
/dev/vg00/lvol15        102400     17209     81520  17%   /opt/scripts
/dev/vg00/lvol17       1048576    301785    700256  30%   /opt/product/maestro
/dev/vg00/lvol16        102400     79128     21829  78%   /opt/mon
/dev/vg00/lvol18       1048576    496054    518269  49%   /opt/bdc
/dev/vg00/lvol4         258048     35111    209093  14%   /home
/dev/xpvg01_n/lvol1    7340032   2822251   4235428  40%   /u01
/dev/xpvg01_n/lvol2   14336000  13847410    473326  97%   /u02
/dev/xpvg01_n/lvol3   15179776  10609859   4284301  71%   /u03
/dev/xpvg01_n/lvol4   10485760   4810316   5498200  47%   /u04
/dev/xpvg01_n/lvol5    8388608   4855426   3446268  58%   /jdf/jdfprod
/dev/xpvg01_n/lvol8    5242880   3628134   1513842  71%   /gb
/dev/xpvg01_n/lvol10    102400      1206     94939   1%   /jdf/jdfpr/mcs/cgr/in
/dev/xpvg01_n/lvol11    131072      1134    121824   1%   /jdf/jdfpr/jdftmp2
/dev/xpvg01_n/lvol12  14655488  13039896   1565110  89%   /u05
/dev/xpvg01_n/lvol13    131072      1134    121824   1%   /jdf/jdfpr/jdftmp1
/dev/xpvg01_n/lvol14   7081984   3690664   3179424  54%   /u06
/dev/xpvg01a_n/lvol1  15360000   4198490  10812720  28%   /u07
/dev/xpvg01a_n/lvol2  15360000   9965892   5225612  66%   /u08
/dev/xpvg01a_n/lvol3  15360000   4198554  10812722  28%   /u09
/dev/xpvg01a_n/lvol4   4096000      2104   3838034   0%   /u10
```

When all the numbers are added up, it turns out that there is 136.6GB of storage "seen" by this system. It has used 77.8GB and has 58.8GB free.

The first complication is to do with all the various lines returned by the command. This shows that there are, in fact, 28 different locations that contain disk space. If an application requires space in one location, it can only use what is free there. So for example the highlighted line, which is a filesystem used by a mailing application, has 1.5GB available. If this application needed

[1] Or "bdf" on HP-UX systems. The command here reports disk sizes in KB

2GB of space its requirements could not be satisfied without some reorganisations or housekeeping (see part 3 below).
So far as Oracle is concerned, it uses the filesystems whose names start /u … There are 10 of these and they contain 50GB of free space. This means the Oracle database could be expanded to use this extra space but not the 8 or 9 GB that is available in the non-Oracle filesystems.

2.) Database
A second consideration is that the database itself contains unused space. Oracle has a mechanism of creating database *tables* of a pre-determined size in a given *tablespace* and using them to store data. When a tablespace becomes full it can be extended by adding extra files from the pool of free filesystem space we discussed above.
This means that as well as the free space on the disks, there will be free space inside the database, too. Running an Oracle query to report on the space inside the database yields:

```
p137-perfmon: sqlplus / @db_usage.sql

SQL*Plus: Release 8.1.7.0.0 - Production on Tue Dec 21 10:13:41 2004

(c) Copyright 2000 Oracle Corporation.  All rights reserved.


Connected to:
Oracle8i Enterprise Edition Release 8.1.7.4.0 - Production
With the Partitioning option
JServer Release 8.1.7.4.0 - Production


TABLESPACE_NAME                MB Total MB Free MB Largest PERCENT_USED
------------------------------ -------- ------- ---------- ------------
IND_GAS_ACTUALS                    8992     650        330        92.77
USERS                               250      25          5        90.14
IND_ELEC_ACTUALS                   7680     816        736        89.38
TAB_ELEC_ACTUALS                   4096     592        304        85.55
TAB_GAS_ACTUALS                    5120     768        384           85
TAB_EXPECTATION                    4096     656        336        83.98
TAB_MPR_REGISTER                   3000     496        496        83.47
IND_MPR_REGISTER                   2048     368        368        82.03
TAB_DATA_2                         4223     928        928        78.03
IND_EXPECTATION                    8192    2176        448        73.44
TAB_DATA                           1536     425        190        72.33
IND_DATA                           1024     369        369        63.97
SYSTEM                              200     150        148         25.2
RBS                                4096    3840       1866         6.25
Disconnected from Oracle8i Enterprise Edition Release 8.1.7.4.0 -
Production
With the Partitioning option
JServer Release 8.1.7.4.0 - Production
```

This shows that there are various amounts of free space available within the individual *tablespaces* of the database. Each one will grow at a different rate, depending on the type of data it stores. When a particular tablespace runs out

of space, the applications that use it will fail until more space is allocated from the free space on the disks.

For example, the *users* tablespace is pretty short of space. If an application used up the 25MB that it has free, it would be unable to continue until some of the 50GB of Oracle space (see above) is made available to it. At that time, the amount of free space in the *users* tablespace would be increased by a few GB and the amount of free disk space would decrease by a slightly larger amount. One subtlety is the column labeled “MB Largest”. This is the largest single chunk of free space. Even though there is 25MB free for users, if one of them tried to use a single 6MB chunk of space, they would get an error message as the database could not find a piece of space large enough for them.

3.) Physical disk

There is one final and hidden aspect to storage. Not all the actual space on a disk is necessarily assigned for files. The nature of this implementation means that there is a layer of abstraction between the actual hardware and the filesystems used to hold data. This layer defines entities called *Physical Extents,* which are 4MB blocks of disk space. These are assigned at will to various *Volume Groups* - which are what the user sees as part of the filesystem names in the first table, e.g. /dev/**vg00**/lvol07.

A report of the first volume group looks like this:

```
p137-perfmon: vgdisplay
--- Volume groups ---
VG Name                     /dev/vg00
VG Write Access             read/write
VG Status                   available
Max LV                      255
Cur LV                      15
Open LV                     15
Max PV                      16
Cur PV                      2
Act PV                      2
Max PE per PV               4350
VGDA                        4
PE Size (Mbytes)            4
Total PE                    8680
Alloc PE                    7166
Free PE                     1514
Total PVG                   0
Total Spare PVs             0
Total Spare PVs in use      0
```

Here there are 8680 physical extents (or 34.7GB) available to all the filesystems in the vg00 group. However, only 7166 (28.7GB) are allocated, leaving 6GB of disk space that is invisible. This space could be assigned to any of the filesystems that are in the vg00 group.

Conclusion

From this we can see that there are 3 answers to the question “how much free space is there?” One answer comes from looking at the amount of free space within the database itself. The second comes from looking at the amount of space available to the applications and that can be assigned to the database if, or when, it runs short. The third answer is the absolute amount of space on the disks, of which a proportion is often kept unallocated.

In this particular case, the database is sized at a total of 54.6GB usable space, which takes up 66.4GB on disk, with 50.7GB that could be assigned later. There is a total of 11.4GB of disk space used by the operating system and other applications and an additional 8.1GB of free space they could use.

This gives a grand total of 136.6GB of disk space that is visible to everything on this machine. However when the hidden space is taken into account, the total amount of purchased disk space is173.6GB.

Afterword.
This example is taken from an HP-UX Unix server. Although the details here are specific to this version of Unix, the principles apply to most operating systems that work with modern, enterprise class, storage. In fact, once you start to dissect the inner workings of an intelligent storage array, you could well find that there are even more layers. Some products have each physical disk divided into Logical Units (LUNs), associated with RAID groups and possibly have their own Logical Volme Manager defining which parts of disks get associated with which uses.
The moral is storage is no longer the simple 1-disk:1-filesystem arrangement that the models and monitoring packages assume. To fully manage disk performance you need to both configure your storage in a manageable way and then set up your monitoring to account for the physical layout.