

Performance Testing & Tuning Your Disk Drives

Stan Sieler

Michael Hensley

Allegro Consultants, Inc.
www.allegro.com

2001-06-22

In the last 20 years, hard disk drives have gone from 120 MB in a washing machine size cabinet (about 6,000 bytes per cubic inch) to 181 GB in a box the size of a paperback book (about 3.7 billion bytes per cubic inch) ... about 1/2 million times more compact!

Data transfer rates have gone from about 100 KB/second to about 50 MB/second ... about 500 times faster!

The average "seek" time has dropped from about 90 ms to 4 ms.

The cost per megabyte has gone from about \$150/MB to \$0.003/MB (one third of a penny per MB), about 1/45,000th the price!

Choices for hard drives have multiplied, from having two different drives available from HP (7925 vs. 7935) to having hundreds of different drives from half a dozen manufacturers, sold by thousands of retailers.

Yet, many sites still use disk drives they bought in the 1980s.
And, many sites still use disk drives the way they did in the 1980s.

How do disk drives work, how can we test them, and how can we take advantage of all of the above changes?

1. Hard Drive Operation

A hard disk drive contains one or more rigid circular platters, coated with a magnetic media, on which data is recorded.

The drive has two magnetic read/write "heads" for each platter (assuming all platters use both sides). All of the heads move in unison, because they're mounted onto the same "arm", which pivots to move the heads in (to read data closer to the center of a platter) and out (to read data closer to the outside of a platter).

Each disk head is used to read/write data on one side of one platter. The data a single head can access is stored in concentric rings, called tracks, on one side of a platter. If you consider the same track on both sides of the platter, and the same track from both sides of every other platter, it's called a cylinder. The track at the outside of a platter is track 0.

Each track is composed of some number of sectors. A sector is generally 512 bytes of user data. (Many older disk drives have sectors of 256 bytes of user data, and with MPE, we're used to considering sectors as blocks of 256 bytes ... MPE presents that view to the user, to remain consistent across time and across multiple kinds of disk drives. For this paper, a sector will be 512 bytes.)

The number of sectors per track usually varies from one value at the outside of a platter to a smaller number at the inside of a platter. (Since an outside track is longer than an inside track, there's more physical room for data). Because of this, disk drive manufacturers usually specify the average number of sectors per track.

In addition to the user data of 512 bytes, each sector has some overhead data, including a header and trailer, and a checksum (or CRC).

What factors influence the time it takes to read a sector?

- command overhead (how fast can the electronics/software in the disk controller process the read command, once it has been received by the drive?)
- seek time (moving the arm holding the head to the right track)
- head settle time (when the heads move, it takes some time for them to come to a complete stop ... this is very short, and seems to usually be included in the seek time specifications, so we'll ignore it for this paper)
- average rotational latency (when will the sector be under the head)
- data transfer time (how fast do the bits come off the surface of the disk)
- cache hit rate (how likely is the requested sector to be in the cache memory of the controller?)

and, once the drive has read a sector, it must send it to the mainframe (an HP 3000 or HP 9000 for this paper):

- channel transfer time (e.g., moving the data over a SCSI cable to the mainframe)

Put another way, the above says:

- figure out what to do
- wait for the arm to move the head to the correct track
- wait for the disk to rotate to the correct sector
- read the bits from the disk into the controller's buffer
- send the bits to the mainframe

While looking at those factors, we'll use a current (in June, 2001) 36 GB SCSI drive as an example: the IBM Ultrastar 36Z15

(<http://www.storage.ibm.com/hardsoft/diskdrdl/ultra/36z15data.htm>):

```
Capacity:                36.7 GB
Interface:                Ultra 160 and Ultra 320 SCSI
Rotational Speed:        15,000 RPM
Average seek time:       4.2 ms
Sector size:              512 bytes (usually)
Sectors per track:       322 to 465
Heads:                    12
Platters:                 6
Sectors per cylinder:    3864 to 5580
Cylinders:                14533
Cache:                    4 MB
Latency (average):       2.00 ms
Media transfer rate:     56.6 MB/sec - 80.8 MB/sec
Interface transfer rate: 160 MB/sec - 320 MB/sec
Sustained data rate:     36.6 MB/sec - 52.8 MB/sec
Seek time:
  Average                  4.2 ms
  Track to track           0.65 ms
  Full stroke              8.9 ms
Error rate (nonrecoverable): 10 in 10E16 bits read
  (this means that if you read the entire 40 GB disk 300,000 times, you're likely to get 10 errors ... this is
  much better than older disk drives)
```

Command Overhead

Command overhead is the amount of time it takes the electronics/software in the disk controller to process the command. On the Ultrastar drive, that's about 0.4 milliseconds.

Seek Time

Seek times are generally specified by three numbers: an average, a track-to-track time (the time required to move the disk head from one track to the next track), and full stroke (the time required to move the head the farthest: from the outside track to the inner track (or vice versa)).

The Ultrastar's seek times:

Average	4.2 ms
Track to track	0.65 ms
Full stroke	8.9 ms

Average Rotational Latency

The disk platters spin at high rates...typically 5,400, or 7,200, or 10,000, or even 15,000 RPM. A combination of spin rate, head seek time, and data density affect the performance of a disk drive. Faster spin rates usually mean better performance (and more heat).

Rotational latency is the time it takes for the disk to rotate so that the target sector on the disk is under the read/write head. The average rotational latency is 1/2 of the time it takes to do one rotation of the platter. For a 7,200 RPM disk drive, each rotation takes 8.3 milliseconds (since we have 60 seconds per minute, or 60,000 milliseconds per minute, and we have 7,200 rotations per 60,000 milliseconds, then each rotation is $60,000 / 7200$ milliseconds.) Thus, the average rotational latency for 7,200 RPM disks is 4.15 milliseconds.

<u>RPM</u>	<u>Average Rotational Latency</u>
5,400	5.55 milliseconds
7,200	4.15 milliseconds
10,000	3.00 milliseconds
15,000	2.00 milliseconds

The above table shows that moving from a 5,400 RPM disk to a 15,000 RPM disk would immediately cut 3.55 milliseconds off an average I/O.

Data Transfer Time

The data transfer time is the time it takes to actually read the data as it passes under the head. For the Ultrastar, the value we want is the "sustained data rate", which varies from 52.8 MB/sec for the outside track (track 0) to 36.6 MB/s for the inside track. This speed is closely linked to the RPM.

The variation between the outer and inner tracks means that files placed on the outside tracks will perform better.

Controller Cache

Most modern disk drives have some form of "cache" memory, which they use to accelerate the performance of the disk drive.

Read cache is memory in the disk controller used to hold data that was read recently (and therefore might be read again), and to hold data that is near recently read data on disk (and therefore might be requested soon).

Write cache is memory in the disk controller used to hold data that was recently sent to the disk by the mainframe. In some cases, this data may have already been written to the disk, but is being held in case the mainframe happens to try to read it soon. In other cases, the data has been accepted by the drive as "written", but it has not been fully written to the disk platter(s) yet.

The internal implementation of the cache for many disk drives seems to be hard to determine. In the case of the Ultrastar, it is documented as breaking the cache into "segments", where the size of the segment is user controllable:

- 27 segments of 128 KB (or, 32 pages)
- 13 segments of 256 KB (or, 64 pages)
- 6 segments of 512 KB (or, 128 pages)

Channel Transfer Time

This is the time it takes to send bytes of data over the SCSI (or whatever interface you are using) bus from the disk drive to the computer. SCSI (Small Computer System Interface) comes in a variety of standards:

<u>Width</u>	<u>MB/sec</u>	<u>SCSI interface type</u>
8	5	SCSI-1
8	10	Fast SCSI (aka SCSI-2)
8	20	Ultra SCSI (aka SCSI-3)
8	40	Ultra2 SCSI
16	20	Fast Wide SCSI (aka Wide SCSI-2)
16	40	Wide Ultra SCSI (aka Wide SCSI-3)
16	80	Wide Ultra2 SCSI
16	160	Ultra 160 SCSI

(The "Width" column is the number of data bits sent at a time)

A "Wide" interface sends data 16 bits at a time, a normal (or non-wide) interface sends data 8 bits at a time.

The most common SCSI interface on an HP 3000 or HP 9000 is probably the SCSI-2, with the 50-pin High Density connector. (This is the connector on the multi-function card used in the 3000/9x7, 3000/9x8, 9000/Ex5, 9000/8x7, and one of the two standard connectors on A-Class systems.)

(more information on SCSI cables / connectors can be found in appendix A)

Calculating the Cost of a Read

Now that we've looked at the factors, we can calculate I/O time:

command overhead + seek time + average rotational latency + transfer time

(note: the above disregards controller caching)

Using numbers similar to the Ultrastar, we can calculate the time it might take to read one page (4,096 bytes) from track 0:

<u>RPM</u>	<u>Cmd</u>	<u>Seek</u>	<u>Latency</u>	<u>Transfer</u>	<u>4,096 byte read</u>
5,400	0.4	4.2	5.55	(19 MB/sec)	10.36
7,200	0.4	4.2	4.15	(25 MB/sec)	8.91
10,000	0.4	4.2	3.00	(34 MB/sec)	7.71
15,000	0.4	4.2	2.00	(52 MB/sec)	6.68

(all times in milliseconds)

Note that the actual speed we'll see from the mainframe will be lower ... bus contention, the speed of the bus, and the speed of the CPU all conspire to further limit performance.

If the sector had been found in cache, the cost of the read would be:

command overhead + transfer time

which would give us:

<u>RPM</u>	<u>Cmd</u>	<u>4,096 byte read</u>
any	0.4	0.4

2. Data Access Patterns

We can see from the formula to calculate the time for an I/O that the pattern of data access will directly affect the overall performance. Reading sequential data will minimize the seek time. Depending upon the elapsed time between reads, the rotational delay will affect performance in a hard to predict manner. Random reads will be affected by seek time, and longer reads will better amortize the cost of the setup, seek, and rotational delay.

The reason that rotational delay is hard to computer for two consecutive accesses is that we don't know how much time will elapsed between the two disk I/O requests. If the second request comes just after the target sector has gone past the disk head, the drive will have to wait for one full revolution of the platter before it can read the sector.

3. Benchmarks

So, how can we tell how well our disk drives can perform? How can we compare one drive to another? The answer is "benchmarks", tests designed to provide performance information.

There are several approaches to benchmarking the performance of a disk drive (or a set of disk drives):

- FOS-based tools/commands
- ISL-based tools
- third party products

The FOS-based tools will be discussed only for MPE/iX, because HP-UX seems to lack the ability for a user to flush the mainframe cache / file-buffers. This flushing is necessary for repeatable benchmarks.

The ISL-based and third-party tools will be discussed for both HP-UX and MPE/iX.

3.1 FOS-based tools/commands (MPE/iX)

We can test the performance of a disk drive by timing commands like "COPY". By putting large files on different drives, we can compare different disk drives against each other. By placing the file in different locations on the same drive, we can see the effect of file placement (e.g., outside vs. inside tracks) on performance. The advantage of FOS-based tools/commands is that they're free, and they're on all HP 3000s.

The key to doing FOS-based testing is to be able to flush data out of memory, to avoid MPE's "caching" influencing the results. The free utility, FLUSH, from:

<http://www.allegro.com/software/hp3000/allegro.html#FLUSH>

will flush all pages associated with closed disk files from memory.

Let's do some performance testing on a 10 MB file.

First, we'll build the data file:

```
comment We'll assume this command file is called MAKE10.
comment Build a 10 MB file (by copying data into a new file).
comment The file is a single extent!

build tenmb; rec=128,1,f,binary; disc=40960, 1

comment Put some data into it (10 MB worth):

fcopy from=nl.pub.sys; to=tenmb
comment The above takes about 15 seconds on a 968.
```

We aren't trying to time the above, because there are far too many uncontrollable variables affecting the performance.

Now, let's time read access to the file.

Here is a sample command file to time read access for the 10 MB file:

```
comment Assume this command file is called TIME10

run flush.pub.allegro
comment The above flushes all closed files from memory

if finfo ("tenmb", "accessors") > 0 then
  echo Sorry, someone has the file open
  return
endif

if not finfo ("yes", "exists") then
  echo YES > yes
  save yes
endif

file null = $null

setvar _cpu hpcpumsecs
setvar _elapsed hpconnsecs

fcopy from = tenmb; to = $null < yes

setvar _cpu (hpcpumsecs - _cpu)
setvar _elapsed (hpconnsecs - _elapsed)

echo FCOPY of TENMB to $NULL took !_elapsed secs, !_cpu CPU millisecs
```

Note that the file builds a file called "YES" which contains "YES", to make FCOPY happy if this command file is run interactively.

Sample:

```
:make10
:time10
...
FCOPY of TENMB to $NULL took 13 secs, 12114 CPU millisecs
```

While we can do benchmarking from FOS, we cannot do it really well. The FOS-based tools are not designed for doing benchmarking, and introduce significant overhead.

3.2 ISL-based benchmarks

From the ISL> prompt, ODE has a command that can read an entire disk drive. This can be used to calculate serial disk performance.

```
ISL> ODE
ODE> DISKUTIL
DISKUTIL> RUN
```

This will invoke three tests of the disk drive...two are very short, and the other will read the entire disk. If you use a stopwatch, you can time how long it takes.

Note: DISKUTIL is picky, and only works with some disk drives.

Running DISKUTIL's "RUN" for two different disk drives will provide a comparison of their serial access performance. Unfortunately, there is no way to tell DISKUTIL to stop after a specified amount of time (or a specified number of sectors). This can lead to lengthy tests.

3.3 Third-party Products

DiskPerf, from Allegro Consultants, provides the ability to do non-destructive tests of any disk drive attached to an HP 3000 (and configured as a disk drive) or an HP 9000.

Sample run:

```
:diskperf
readdisk 3 mbs 10 quiet
  (this will read ldev 3, starting at sector 0, for 10 MBs)

Took 10 CPU; 2,233 elapsed millisecs
```

We will use DiskPerf for benchmarks for the rest of this paper.

Here we see that reading 10 MB from ldev 3 took 2.2 seconds, which is 1/6 the time it took our FOS-based test (FCOPY).

Some other HP-UX disk benchmark programs exist, including diskbench (an unsupported tool at <http://devresource.hp.com/devresource/Tools/Diskbench/Diskbench.html>).

However, except for DiskPerf, none will also run on MPE/iX, so we will use DiskPerf in this paper (<http://www.allegro.com/products/diskperf/>).

4. Performance Moves

There are seven kinds of performance moves we can make on an HP 3000:

- replace old technology, slow disk drives;
- move small frequently-accessed files to disks with high performing seek mechanisms;
- group small files together;
- move large serially-accessed files to the outer tracks of fast serial transfer disk drives;
- optimize same-sector access (if we use this);
- use more disk drives;
- use a faster CPU.

Each of these will be examined, in turn.

Note: the majority of the tests in this section were run on a standalone HP 3000/968, using DiskPerf [20000620], on MPE/iX 6.0.

4.1 Replace old technology, slow disk drives

4.1.1 HP-IB

Many HP 3000 and HP 9000s are using disk drives that were designed and built in the 1980s, in the early-to-mid 1990s.

If you are using any HP-IB disk drives, replace them. (This assumes, of course, that your HP 3000 / HP 9000 can boot from a SCSI disk drive. If yours cannot, consider contacting HP for a firmware upgrade to allow a SCSI boot.)

HP-IB drives are extremely slow.

The following compares the performance of an HP-IB disk (7951, inside an HP 7963B disk subsystem) against two SCSI disks ("HP s" is an HPC2490AM (2 GB), "ST s" is an ST19171N (9 GB)):

```

--- Reads/sec ---    (#pages = 1)

```

DISK	Random		Serial	
	Best	Average	Best	Average
HP s	79.9 r	79.5 r	97.6 r	96.9 r
ST s	103.0 r	102.1 r	599.2 r	596.8 r
HPIB	27.8 r	27.5 r	42.1 r	41.8 r

```

--- MBs/sec ---    (#pages = 1)

```

DISK	Random		Serial	
	Best	Average	Best	Average
HP s	0.3 M	0.3 M	0.4 M	0.4 M
ST s	0.4 M	0.4 M	2.3 M	2.3 M
HPIB	0.1 M	0.1 M	0.2 M	0.2 M

The HP-IB drive is about 1/3 the speed of the five year old HP SCSI drive, and about 1/4 the speed of a two year old Seagate SCSI drive for random access, and less than one tenth the speed of the Seagate drive for serial access.

If we test for the kinds of serial access the file system will do when it knows you're accessing a file serially (i.e., read 8 pages at a time) we get the following:

```

--- Reads/sec ---    (#pages = 8)

```

DISK	Serial	
	Best	Average
HP s	62.3 r	62.1 r
ST s	127.3 r	126.4 r
HPIB	15.3 r	15.3 r

```

--- MBs/sec ---

```

DISK	Serial	
	Best	Average
HP s	1.9 M	1.9 M
ST s	4.0 M	4.0 M
HPIB	0.5 M	0.5 M

We see that the tested HP-IB drive is extremely slow compared to older SCSI drives! We also see a big difference in the two SCSI drives.

4.1.2 SCSI

The performance of SCSI drives has changed dramatically over the last 10 years.

The following test compares an ST15230N (manufactured in 1994) against a Seagate ST19171N (manufactured in 1996):

```
--- Reads/sec --- (#pages = 1)
```

Disk	Random		Serial		Year
	Best	Average	Best	Average	
ST19171N	102.6 r	102.5 r	599.9 r	598.1 r	1996
ST15230N	87.5 r	87.2 r	552.3 r	535.3 r	1994

The newer drive is 17% faster!

If we test for the kinds of serial access the file system will do when it knows you're accessing a file serially, we get the following:

```
--- Reads/sec --- (#pages = 8)
```

Disk	Serial		Year
	Best	Average	
ST19171N	131.9 r	131.0 r	1996
ST15230N	127.6 r	127.0 r	1994

Not as much a difference as for the random access performance.

4.2 Small Files & Fast Seeks

Small files will benefit from being put on disk drives that have superior seek performance. This is because the biggest factor in opening and reading a small file is the time it takes to move the disk head to the first page of the file.

DiskPerf can be used to show the random I/O performance of a set of disk drives.

You may be able to move files from one disk to another (in the same volume set) with carefully planned STORE/purge/RESTORE operations, or with various third party tools, including De-Frag/X from Lund Performance Solutions, and MPEX from VESoft.

4.3 Grouping small files together

If your small files are scattered across an entire disk drive, with extents of large files between them, then the cost of reading any two small files is likely to be more than if the files were next to each other, due to the increased motion of the disk head (seek times).

You may be able to change the position of files on a disk drive with carefully planned STORE/purge/RESTORE operations, or with various third party tools, including De-Frag/X from Lund Performance Solutions, and MPEX from VESoft.

4.4 Using Fast Serial access

Large, serially accessed files will benefit from being put on disk drives that have superior serial I/O performance.

DiskPerf can be used to show the serial I/O performance of a set of disk drives, but the basic recommendation is to install new disk drives, running at 15,000 RPM. The fast rotation speed means good serial I/O performance.

Testing shows that the serial read performance will increase as the read size increases:

ST19171N

#Pages/Read	MB/sec
1	2.3
4	3.7
8	4.1
16	4.3

DiskPerf's "buffer" test shows the performance difference caused by using different I/O sizes. The "#Pages" column is the size of the disk read, in pages (one page = 4,096 bytes):

#Pages	Reads/sec	MBs/sec	#Pages	Reads/sec	MBs/sec
1	303.0 r	1.2 M	2	314.5 r	2.5 M
4	195.3 r	3.1 M	6	163.9 r	3.8 M
8	124.7 r	3.9 M	16	67.0 r	4.2 M
32	34.9 r	4.4 M	64	17.6 r	4.4 M
128	9.0 r	4.5 M	192	5.9 r	4.5 M
256	4.4 r	4.4 M			

(The above is for a Seagate ST19171N disk drive)

We can see a definite benefit in increasing read sizes up to about 16 or 32 pages. Since MPE/iX generally never reads more than 8 or 12 pages at a time, we can see a possible MPE/iX enhancement request here.

4.5 Same-Sector Access

Some disk drives exhibit poor performance when the same disk address is accessed repeatedly. This is probably because their cache control logic says "I've just read sector X, so I won't keep it in my cache ... I'd rather keep sector X+1 in cache".

Unfortunately, some applications do frequent access to the same area of disk (e.g., an application with a home-grown transaction number updating the disk for each new transaction). The (usually very small) data files used in this manner may want to be moved to a disk drive that does not penalize this kind of access.

It seems that Seagate SCSI drives handle "same sector" relatively poorly, at about 1/5 the speed of serial access. This is clearly the result of inopportune (for this test) choices in cache management. However, even at that cost, the Seagate drive we tested outperformed the HP C2490AM drive (which is one year older): (Serial performance shown for comparison.)

Disk	Reads/second (one page)	
	Same	Serial
HPC2490AM	105.7	97.7
ST19171N	118.7	591.2
79591 (HP-IB)	55.4	41.7

The HP C2490AM disk drive was actually faster at "same sector" access than at serial access!

4.6 Faster CPU

The speed of the mainframe CPU can dramatically affect the actual disk performance.

The following two tests are on identical hardware, with only the CPU different.

32 MHz CPU (3000/918)

--- Reads/sec --- (#pages = 1)

Disk	Random		Serial	
	Best	Average	Best	Average
ST34573N	71.8 r	71.1 r	278.2 r	277.2 r
ST39140N	61.1 r	60.9 r	222.7 r	209.0 r

48 MHz CPU (3000/928)

--- Reads/sec --- (#pages = 1)

Disk	Random		Serial	
	Best	Average	Best	Average
ST34573N	73.7 r	72.4 r	395.9 r	360.1 r
ST39140N	65.5 r	64.6 r	258.2 r	257.4 r

The random I/O performance is marginally better on the faster CPU, but the serial I/O performance is significantly better (42% better for the ST34573N drive).

This, of course, implies that disk performance on the HP 3000 A-Class will suffer when compared to an equivalent HP 9000 A-Class, because of the difference in CPU rates.

4.7 More Disk Drives

One performance improvement that we can do is to add more disk drives. This is a definitely a concept that would have been foreign to system managers in the 1980s: disk drives were expensive then! However, with the low cost of disk drives and external cabinets, it is an approach worth considering today.

If a system is using two 4 GB disk drives, adding two more and then "balancing" the disk usage across all four drives would mean that all of the data on each drive would be stored in the outside half of the tracks ... the faster half. This means that data transfer times decrease, and seek times decrease. In addition, having four spindles instead of two means that we can generally get nearly twice as many disk I/Os per second.

5. Replacing a Disk Drive

So...you've decided to replace your boot disk (LDEV 1 for an HP 3000, the "root" disk for an HP 9000). How can you do this without "reloading" (reinstalling the operating system and restoring all of the files from tape)?

The basic answer is: cloning. Cloning a disk drive means copying every bit of data from the original drive to a new ("clone") drive. (The new drive has to be at least the same size as the original drive.)

On MPE, the standard tools supplied with FOS include an offline (ISL-based) utility, ODE, that can clone a disk drive. On HP-UX, the offline diagnostics disk should contain the same utility, ODE.

ODE has a module called "COPYUTIL", which can clone a disk drive with the following steps:

- a) run ODE
- b) enter the command: COPYUTIL
- c) COPY the disk to a tape (or set of tapes)
- d) COPY the tape(s) to the new disk

Once you've cloned the disk drive, you can boot from the new disk.

Note that you have several options with the above approach:

- connect both the source and clone disk drives to the computer at the same time.

If you do this, be sure to disconnect (logically or physically) the source drive before trying to boot from the clone drive. Also, you will probably have to change the address of the clone drive to match the original address of the source drive.

- connect the source drive, copy it to tape, disconnect it, connect the clone drive at the same address the source drive had been, and copy the tape to it.

The ODE/COPYUTIL approach can be used to clone (and replace) any disk drive on the system, not just the boot disk.

There are two major limitations to the ODE/COPYUTIL approach:

- finickiness :: COPYUTIL refuses to deal with disk drives that it does not recognize as officially supported disks.
- performance :: the need to go to tape as an intermediate storage mechanism drastically slows the clone.

On HP-UX, an online clone can be done with the "dd" command. Note, however, that you are well advised to unmount all file systems from the source disk drive before starting the dd. Otherwise, the result may be unusable. To clone the root disk, you may be able to boot to a RAM-resident version of HP-UX to do the dd.

On the HP 3000, De-Frag/X provides an online CLONE command, which can clone idle ("LONER") disks, and can clone in-use disks. Note that cloning an in-use disk must be done carefully, to minimize the chance that data will change during the clone operation, and render the clone invalid.

"dd" and De-Frag/X both allow the clone disk to be larger than the source disk (the same size is also acceptable, of course). In both cases, the result will be a cloned disk that looks like the original disk! This means that if you clone a 4 GB drive onto a 9 GB drive, the result will be a disk has 4 GB usable on it... because all of the on-disk data structures that described the disk as a 4 GB disk were carefully cloned onto the 9 GB disk! I.e., you won't be able to access those extra 5 GB! For the HP 3000, an enhancement proposal to allow access to that "new" area on such a disk has been submitted by SIGMPE (the "disk partitioning" enhancement).

5.1 Replacing a non-boot disk

In general, replacing a disk other than a boot disk is simpler. The software mentioned above can be used, but the concern about the data on the source disk changing during the clone is reduced because you can generally "isolate" the source disk from changes.

On an HP 3000, if the source disk is part of a "user volume set" (i.e., not part of MPEXL_SYSTEM_VOLUME_SET), you can VSCLOSE the volume set, and then clone the now idle source disk. If the source disk is part of MPEXL_SYSTEM_VOLUME_SET (but not

LDEV 1), you can boot with "START NORECOVERY SINGLE-DISC SINGLE-USER", which should bootup using only LDEV 1. Note: if you intend to use the SINGLE-DISC bootup, first ensure that the cloning software you intend to use is located entirely on LDEV 1!

6. Summary

In the 1980s, we learned that disk drives were very expensive, bulky, and relatively slow. In 2001, disk drives are dirt cheap, tiny, and very fast. These changes should encourage us to look at the disk drives we use, and how we use them, and consider approaching disk drive usage with a 21st Century approach.

Appendix A :: SCSI cables / connectors

In addition to the width (normal or wide) and the SCSI interface type, a related topic is type of connector used for SCSI devices.

50 Pin Centronics

Often seen on external devices, very common.
Designed for SCSI-1 and SCSI-2.

50 Pin SCSI-2 (aka "High Density SCSI-2")

Seen on the back of HP 3000s and HP 9000s (e.g., on the multifunction card of a 9x7, 9x8, or Ex5).

68 Pin SCSI-2 Wide

The "Wide" equivalent of the High Density SCSI-2 connector.
Seen on HP's fast/wide SCSI adapters.

80 Pin SCA

This essentially combines a 68 PIN SCSI-2 Wide connector with the SCSI ID lines and power lines, all in one cable.

You can often use an adapter to convert from one kind of cable to another.

In all cases, I highly recommend using high quality SCSI cables and high quality terminators and high quality adapters. This is not the area to economize in! Granite Digital, at www.scsipro.com, offers very high quality SCSI products.

Appendix B :: Seagate Model Numbers

Many HP 3000 and HP 9000 computers have Seagate disk drives. Seagate has a very good convention for naming their drives, and understanding it means that you can look at a Seagate model number and instantly know the drive form factor, capacity, and interface type! The following is derived from <http://www.seagate.com/support/kb/disc/discmodelinter.html>

Format: ST <Form Factor> <# MBs> <Interface>

Example: ST15150N (ST 1 5150 N)

Form Factor: 1 = 3.5-inch, half-height, 1.6-inch high

MBS = 5150 (i.e., about 5 GB)

Interface: N = SCSI, 50-pin Narrow SCSI Connector

Form Factor

1 = 3.5-inch, half-height, 1.6-inch high

2 = 5.25-inch, half-height, 1.6-inch high

3 = 3.5-inch, low-profile (LP), 1-inch high

4 = 5.25-inch, full height

5 = 3.5-inch, 0.75-inch high

6 = 9-inch

7 = 1.8-inch

8 = 8-inch

9 = 2.5-inch, 0.75-inch high

Interface:

DC = Differential, 80-pin Single Connector Attachment (SCA)

FC = Fibre Channel, 40-pin SCA

FCV = Same as FC, but with an increased cache size
(Recommended for Video applications and applications with large file formats)

LC = Low Voltage Differential, 80-pin SCA

LCV = Same as LC, but with an increased cache size
(Recommended for Video applications and applications with large file formats)

LW = Low Voltage Differential, 68-pin Wide SCSI Connector

LWV = Same as LW, but with an increased cache size
(Recommended for Video applications and applications with large file formats)

N = SCSI, 50-pin Narrow SCSI Connector

ND = Differential, 50-pin Narrow SCSI Connector

W = SCSI, 68-pin Wide SCSI Connector

WC = SCSI, 80-pin SCA (Hot Swappable)

WD = Differential, 68-pin Wide SCSI Connector

older models might have:

NM = SCSI, 50-pin Narrow SCSI Connector, Macintosh Compatible

NV = SCSI, 50-pin Narrow SCSI Connector, Netware Ready

non-SCSI models might have:

- A = ATA, Ultra ATA (IDE/EIDE)
- AG = ATA w/SafeWrite Shock Protection
- E = ESDI Interface
- J = SMD/SME-E Interface
- K = IPI-2 Interface or Part of a Kit
- P = PCMCIA (PC Card) Interface
- R = ST412/RLL Interface
- S = Synchronized Spindle or Synchronous SCSI
- X = XT Bus (IDE)
- Unspecified = ST412/MFM Interface

Note: Seagate uses "disc" instead of the (now) standard "disk". Their support line reports that this spelling was required by one of their original investors, for reasons of their own.