

# Trends in disk drives and data transfer speed.

Steven Branigan  
[sb@cyanline.com](mailto:sb@cyanline.com)

## Abstract:

*The increases in disk drives capacities are exceeding the advances in disk data movement. This has started to impact the computer forensic data acquisition process, resulting in exceedingly long times required to make forensic disk image on disk drives that are 300 GBs or larger.*

*Below, I present some of trends in the disk storage industry to help predict where the market is going. I also examine the hardware architecture to determine the optimal configuration for minimizing the time necessary to make forensic images in spite of the trend.*

In just the past 12 years, the speed that data are moved to and from a PC disk drive has increased an amazing 3,093%. Back in 1992, a 43 Megabyte (MB) disk drive could transfer data at the rate of 1.875 MB/second. Now, in a very impressive speedup, disks can move 58 MB/second. A few factors have contributed to this speed-up<sup>1</sup>.

- Internal disk data transfer rates, or the amount of bits the disk can move from the platter to the disk controller have dramatically increased.
- Internal computer processor and data bus speeds have increased greatly as we moved from ISA to PCI.
- Data transfer rates between the disk controller and the computer have dramatically increased from the slow ATA interface to the speedy SATA (150MB/s) interface.

While this speedup is impressive, it is nothing compared to the increase in disk capacity, which has also made even more amazing progress over the same period of time. It has increased over 460,000% since 1992, with today's 300 Gigabyte (GB) disks dwarfing the 43 MB drives of the past.

We see that while disk data transfer speeds and disk capacities are growing, they are not growing at the same rate. This will create new challenges for us in the future. Figure 1 (where the y-axis is logarithmic) displays the disk capacity changes over the years as compared to the data transfer rates – this graph shows that the gap between them is large and widening. As we will see, it is beginning to have unexpected impacts on the acquisition portion of the computer forensic process.

---

<sup>1</sup> The careful reader will notice that the bottleneck on data transfers is the internal data movement, not the interface.

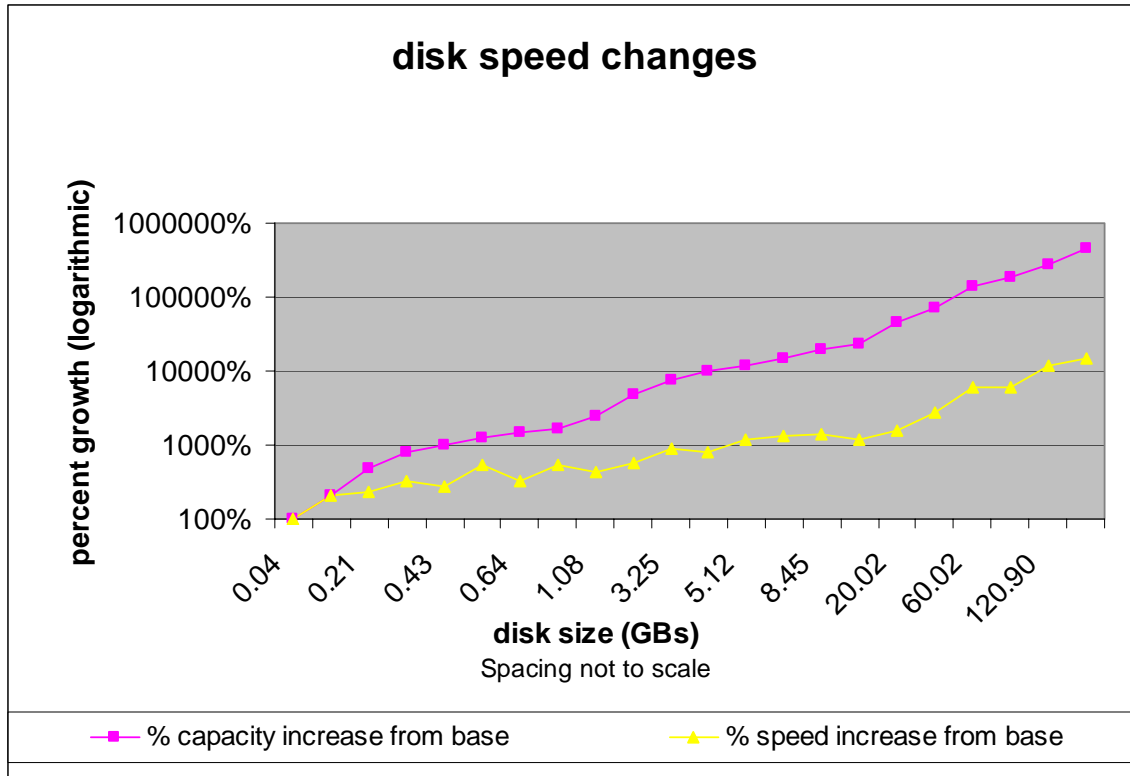
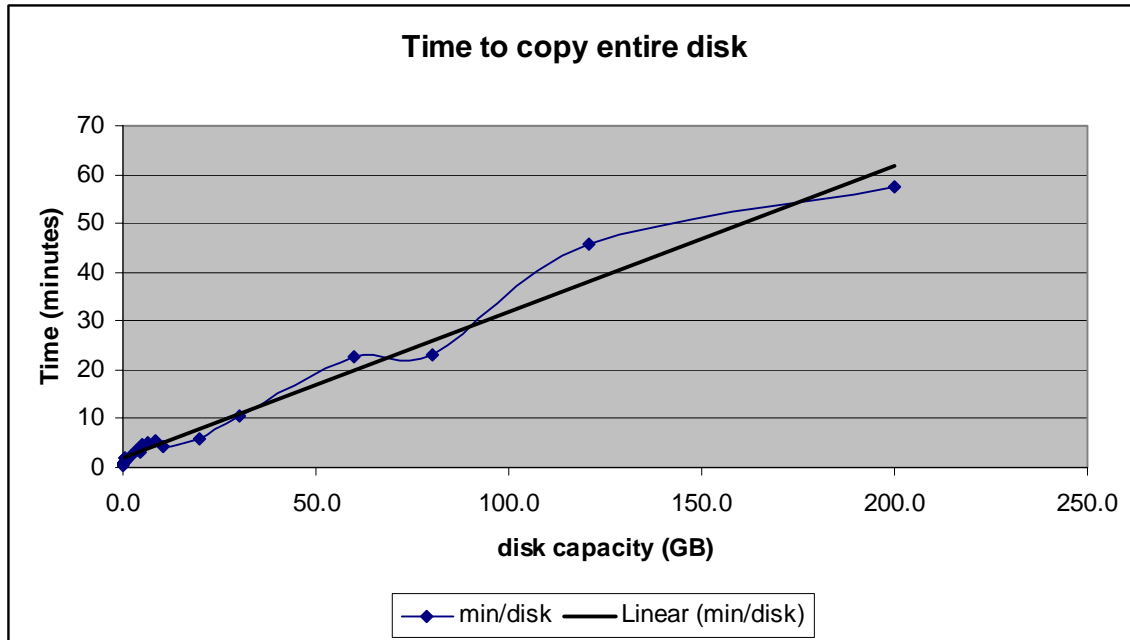


Figure 1

## Getting to all the Data

Current state-of-the-art computer forensics involves making an exact, sector by sector copy of a subject's disk drive. Here, it does not matter how much information has been stored on the disk drive because every single disk sector must be copied, whether it is currently allocated to a file or not. This process is well understood, and specialized devices have been created that can copy an entire disk drive of less than 20 GBs in only 10 minutes. Standard computers could make a copy of the disk drive in about twice the time.

Starting with the introduction of the 60 GB disk drives, those days of less than ten minute data acquisitions are over. Capacity, having raced past the data transfer speed is starting to impact the amount of time required to perform a disk image. The introduction of the 60 GB disk drive marks the first time that a single disk drive cannot be completely copied in less than 10 minutes no matter the hardware. In fact, if everything went right, it would take over 20 minutes to get all of the data off of this disk drive. And, with the introduction of the 120 and 200 GB disk drives, the amount of time necessary has grown even larger under the best circumstances. In Figure 2 you can see how the times required for a disk to transfer all of its data grows with the size of the disk drive. Notice that the 200 GB disk drives require nearly 60 minutes to transfer all of their data.



**Figure 2**

*The data for the above chart has been drawn from disk drive manufacturers' supplied data that I compiled for disk drives from 1992 to present. That chart is listed in Appendix A.*

A key value that I derived is "disk drive copy time" which was calculated by finding the minimum of the maximum disk drive supported data transfer speed. For example, let's look at the row for the ST380013A, listed below in Figure 3.

disk	size (MB)	Size (GB)	rpm	avg read time	sustained xfer rate	Max Internal transfer rate (MB/s)	Max interface xfer rate (MB/s)	Max possible rate (MB/s)	Minutes to copy entire disk
ST380013A	80,000.0	80.000	7200	8.5	58	85.4	100	58	22.99

**Figure 3**

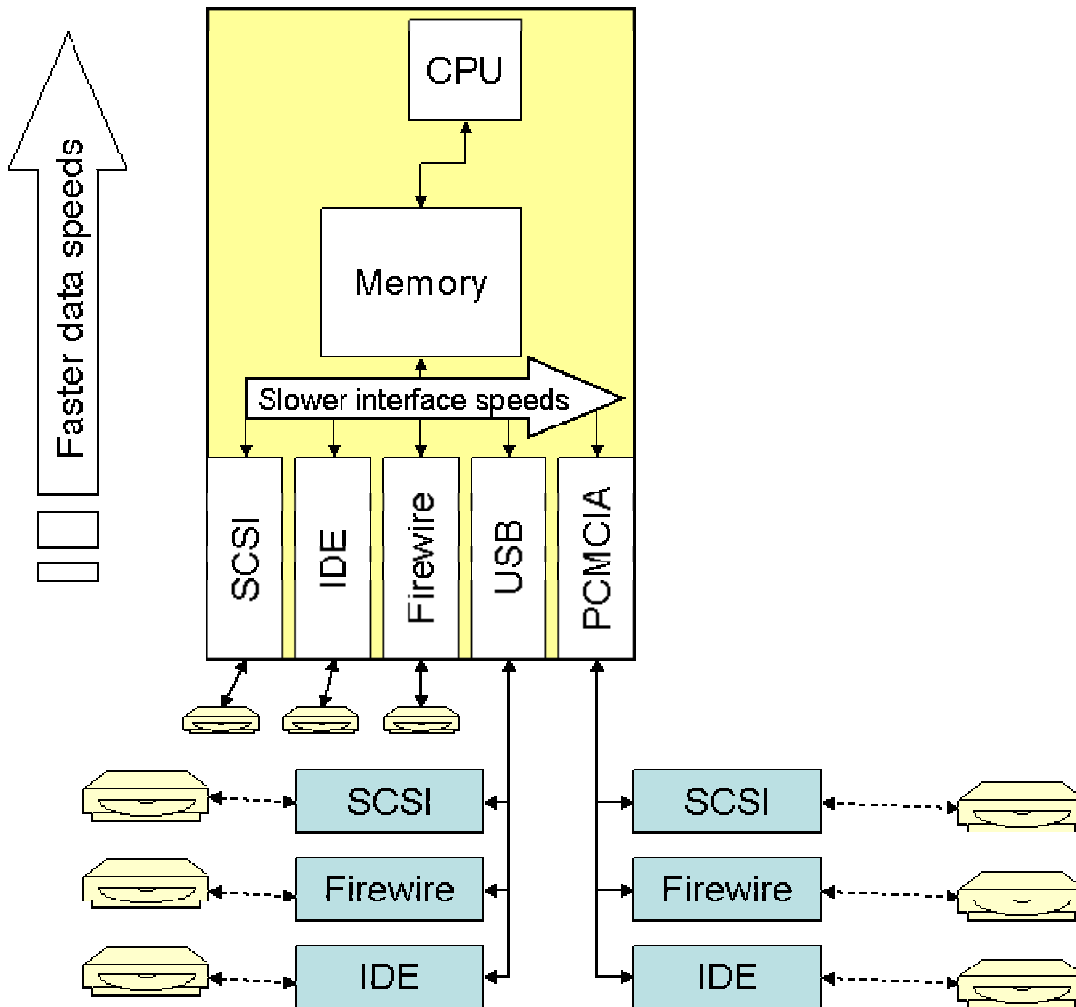
For this drive, we see that it can be copied at a speed limit of 58 MB/s. It has a maximum sustained data transfer rate of 58 MB/s, a maximum internal transfer rate of 85.4 MB/s and a maximum interface rate of 100 MB/s. Since the maximum internal transfer rate is the bottleneck, the disk drive cannot support a sustained data bandwidth of more than 58 MB/s.

It appears, based upon the graph in Figure 2 that we are starting to see a linear relationship between the capacity of a disk drive and the amount of time required to copy all of the data off of the drive.

So, we see that in the best case, very large disk drives can not be copied at a rate faster than 3.6 GB/s under the best scenario. We also see, as a corollary, that the method chosen to forensically copy a disk drive can have a strong impact on the time needed to complete that task.

## How data moves around

Now that we know the best disk drive performance possible, what can we do to achieve it? Disk drives can be connected to computers in a few different ways. A brief review of how disks drives interface with a computer displayed in Figure 4.



**Figure 4**

*The CPU is capable of moving data fastest, followed by the computer's memory and ultimately the interface cards. The primary computer-to-disk interface cards are SCSI, IDE, Firewire, USB and PCMCIA, each with varying speeds of their own. Note that unlike the other interfaces, USB and PCMCIA are special because they are actually bridges, connecting one bus with another.*

Figure 4 implies that SCSI would be the fastest disk drive interface, while Firewire over PCMCIA would be the slowest. At a pure interface level, where Programmed Input/Output is being used this statement is true. Programmed IO requires that the CPU be involved in every data movement from one disk to another. However, some

disk drive interfaces have intelligence of their own and can perform Direct Memory Access (DMA) transactions. DMA transfers require very little service from the CPU, resulting in much faster transfers.

## Striving for the optimal configuration

In an attempt to determine the optimal configuration for performing field acquisitions, I have performed some simple benchmark testing. In some cases, the results are what we would expect, that performing an acquisition without compression would provide for the fastest data transfer. However, I did encounter a configuration where some compression actually increased the overall speed of the task. Below, I present some of the data from my findings along with a recommended optimal configuration.

### *Effects of compression on acquisition*

Let's look at the slowest configuration, the single card firewire to firewire copy in some more detail. For test sets 1 and 2, I used the following configuration.

#### CPU

- Sager Laptop with a 2.8GHz Intel Pentium 4 processor.

#### External Storage

- Western Digital
  - 40GB, 120GB and 200GB disk drives,
- IOMega
  - 40GB FW and 30GB USB2.0 disk drives.

#### External Disk connectivity

- A dual-slot 800Mbps Firewire PCMCIA card
- Internal 4 pin Fireware connector
- Internal USB 2.0 connector

#### Acquisition Software

- Encase 4.19

To create uniform disks, all drives were zeroed using the following command:

```
dd if=/dev/zero of=target disk bs=4096 count=(total bytes/4096)
```

Zeroing the disk drives results in a highly compressible media, of course.

### *Test set 1*

Using a disk with zeroed data

Read device: Firewire PCMCIA

Write device: same Firewire PCMCIA

	<b>40 GB (37.3 effective)</b>	<b>120 GB (111.7 effective)</b>	<b>200 GB (186.2 effective)</b>
<b>No compression</b>	60 min (10.34 MB/s)	172 min (10.82 MB/s)	295 min (10.52 MB/s)
<b>Standard compression</b>	37 min (16.77 MB/s)	110 min (16.93 MB/s)	180 min (17.24 MB/s)
<b>Best Compression</b>	51 min (12.17 MB/s)	140 min (13.3 MB/s)	228 min (13.61 MB/s)

Test Set 1 reveals that for a highly compressible disk drive, using compression can actually reduce to total time needed to perform a full disk acquisition. This indicates that compression actually decreased the time necessary to perform a complete disk copy versus no compression. This finding suggests that the disk writes are the bottle-neck, and anything that reduces the number of disk writes, as compression would, speeds up the overall copy process. Not surprising since one Firewire card was responsible for handling the reads and the writes. Note that there appears to be an optimal amount of compression for certain disk drives configurations.

Test Set 1 also reveals that the time required for a 200GB disk drive is large, nearly 5 hours for a no-compression acquisition using this architecture.

Next, I wanted to discover the effects of compressibility of the source media on the disk copy times. Let's review data collected using highly compressible data (all zeros) versus highly uncompressible data (sourced from /dev/random). This is covered in Test Set 2.

## **Test Set 2**

Let's look a little closer at the compressibility of the disk drive as it relates to acquisition time. We used a highly compressible disk drive in Test Set 1. In Test Set 2, I compare the acquisition time for a 40GB disk drive that is highly compressible versus a disk drive that is highly random. To create a disk filled with highly random data, I used the following command:

```
dd if=/dev/urandom of=target disk bs=4096 count=(total bytes/4096)
```

	<b>40 GB (zeroed disk)</b>	<b>40 GB (random data)</b>
<b>No compression</b>	60 min (10.34 MB/s)	60 min (10.34 MB/s)
<b>Standard compression</b>	37 min (16.77 MB/s)	125 min (4.97 MB/s)

<b>Best Compression</b>	51 min (12.16 MB/s)	133 min (4.67 MB/s)
-------------------------	------------------------	------------------------

Here we see that compression greatly slows the acquisition time for a drive full of highly random data, suggesting that the CPU is the bottleneck for these acquisitions.

Test Sets 1 and 2 are the two extreme cases. Actual disk drives in the field should be somewhere in between the two, since most user data is text, and text is highly compressible. It is best, going forward, that we ignore the effects of compression on hardware testing, since the effects of compression are directly related to the type of data being acquired.

### ***Test Set 3***

A comparison of different data acquisition techniques and their effective throughputs. In all cases, including the IDE to IDE, these tests were run using only a single disk drive attached to an interface. (For the IDE case, it required two cards.) Notice that the times for all but the IDE-> USB2.0 acquisition are faster than running two disk drives off of the same PCMCIA card.

40GB no compression	<b>IDE -&gt; IDE</b>	<b>IDE -&gt; Firewire</b>	<b>USB 2.0 -&gt; IDE</b>	<b>Firewire -&gt; IDE</b>	<b>IDE-&gt; USB 2.0</b>
Time	22 minutes	30 minutes	35 minutes	37 minutes	65 minutes
Throughput MB/s	28.94	21.22	18.19	17.21	9.79
Extrapolation to 200GB	110 minutes	150 minutes	175 minutes	185 minutes	325 minutes

None of these times gets us to the theoretical throughput that the disks are capable of performing. Only the IDE-> IDE copy is even close. More research should be done here to examine if two disks on the same IDE controller card can result in a faster acquisitions than the 22 minutes required to copy a 40GB disk drive.

The other hardware architectures have a strong impact on the overall performance of the copy. This indicates that not all interfaces are efficient with writing to disk drives. Therefore, anything that we can do in the architecture to increase the write throughput will ultimately decrease the time necessary to perform a disk copy.

And, as the industry has already made 200GB disk drives available, it is merely a matter of months before we will need copy a drive of this size or larger. With an improper configuration, one of these operations could take up to 6 hours. However,

by following some architectural guidelines, we can safely reduce this acquisition to merely 2-3 hours.

How? By ensuring that the disk that we are writing to is on the fastest bus possible. The best configuration that I have encountered so far is having the write disk be on an IDE bus. The read-disk can then easily be accessed using either firewire or USB.

## **Conclusions**

So, why might this matter? This has a very large impact to the data acquisition portion of the computer forensics process. Data acquisitions for the new, large disk drives will require much more time than they have before, even under the best of circumstances. Under the best scenario, a 200 GB disk drive can not be imaged in less than 60 minutes due to the data transfer limitations.

So computer forensics personnel will need to allocate substantially more time when needing to make forensic images. No hardware will be able to make imaging disk drives of this size faster. The limitation factors are the disks themselves.

It is more important as we move forward that forensic examiners concentrate on doing disk acquisitions in parallel so that the overall time required for a "job" is kept to a manageable level.

## Appendix A

### Disk drive statistics over the years.

disk	bus	year	size (MB)	Size (GB)	rpm	avg read time	sustained xfer rate	Max internal transfer rate (MB/s)	Max interface xfer rate (MB/s)	Max possible rate (MB/s)	Minutes to copy entire disk
ST3051A	ATA	1992	43.1	0.043	3200	16	N/A	1.875	5	1.875	0.38
ST3096A	ATA	1992	89.2	0.089	3200	14	N/A	1.875	5	1.875	0.79
ST3240A	ATA	1994	210.9	0.211	3800	15	N/A	4	5	4	0.88
ST3390A	ATA	1993	341.3	0.341	4500	12	N/A	4.57	11.1	4.57	1.24
ST3425A	ATA	1995	425.0	0.425	3600	14	N/A	6.9	15.5	6.9	1.03
ST3543A	ATA	1995	541.9	0.542	3600	15	N/A	4.4125	13	4.41	2.05
ST3636A	ATA	1996	640.5	0.640	4500	12.5	N/A	8.4	16.7	8.4	1.27
ST3780A	ATA	1994	722.0	0.722	4500	12	N/A	5.905	16.6	5.91	2.04
ST31010A	ATA	1997	1,080.0	1.080	4500	12.5	N/A	10.975	16.7	10.98	1.64
ST32111A	ATA Ultra-	1998	2,110.0	2.110	4500	13	N/A	15.75	33.3	15.75	2.23
ST33223A	ATA Ultra-	1998	3,250.0	3.250	4500	13	N/A	15.75	33.3	15.75	3.44
ST34312A	ATA Ultra-	1999	4,310.0	4.310	5400	9	N/A	23.5	66.6	23.5	3.06
ST35120A	ATA Ultra-	1998	5,120.0	5.120	5400	12.5	N/A	18.5	33.3	18.5	4.61
ST36422A	ATA Ultra-	1998	6,400.0	6.400	5400	12	N/A	21.375	33.3	21.38	4.99
ST38421A	ATA Ultra-	1999	8,450.0	8.450	5400	10.5	N/A	25.75	66.6	25.75	5.47
ST310212A	ATA Ultra-	1999	10,240.0	10.240	5400	8.9	N/A	38.5	66.6	38.5	4.43
ST320014A	ATA Ultra-	2002	20,020.0	20.020	5400	18	N/A	55.625	100	55.6	6.00
ST330621A	ATA Ultra-	2000	30,020.0	30.020	5400	9.9	N/A	47	100	47	10.65
ST360015A	ATA Ultra-	2002	60,020.0	60.020	7200	9	44	71.25	100	44	22.73
ST380013A	ATA Serial	2003	80,000.0	80.000	7200	8.5	58	85.4	100	58	22.99
ST3120023AS	ATA Serial	2001	120,900.0	120.900	7200	9.4	44	71.25	150	44	45.80
ST3200822AS	ATA	2003	200,000.0	200.000	7200	8.5	58	85.4	150	58	57.47