# Terabyte IDE RAID-5 Disk Arrays

D. A. Sanders, L. M. Cremaldi, V. Eschenburg, R. Godang, C. N. Lawrence, C. Riley, D. J. Summers
*University of Mississippi, Department of Physics and Astronomy, University, MS 38677, USA*
D. L. Petravick
*FNAL, CD-Integrated Systems Development, MS 120, P.O. Box 500, Batavia, IL 60510, USA*

Fermilab

High energy physics experiments are currently recording large amounts of data and in a few years will be recording prodigious quantities of data. New methods must be developed to handle this data and make analysis at universities possible. We examine some techniques that exploit recent developments in commodity hardware. We report on tests of redundant arrays of integrated drive electronics (IDE) disk drives for use in offline high energy physics data analysis. IDE redundant array of inexpensive disks (RAID) prices now are less than the cost per terabyte of million-dollar tape robots! The arrays can be scaled to sizes affordable to institutions without robots and used when fast random access at low cost is important.

FERMILAB-Conf-03/150-E September 2003

## 1. Introduction

We report tests, using the Linux operating system, of redundant arrays of integrated drive electronics (IDE) disk drives for use in particle physics Monte Carlo simulations and data analysis [1]. Parts costs of total systems using commodity IDE disks are now at the $2000 per terabyte level. A revolution is in the making. Disk storage prices have now decreased to the point where they are lower than the cost per terabyte of 300 terabyte Storage Technology tape silos. The disks also offer far better granularity; even small institutions can afford to deploy systems. The faster random access of disk versus tape is another major advantage. Our tests include reports on software redundant arrays of inexpensive disks – Level 5 (RAID-5) systems running under Linux 2.4 using Promise Ultra 133 disk controllers that allow disks larger than 137 GB. The 137 GB limit comes from 28-bit logical block addressing, which allows $2^{28}$ 512 byte blocks on IDE disks. Recently 48-bit logical block addressing has been implemented. RAID-5 protects data in case of a catastrophic single disk failure by providing parity bits. Journaling file systems are used to allow rapid recovery from system crashes.

Our data analysis strategy is to encapsulate data and CPU processing power together. Data is stored on many PCs. Analysis of a particular part of a data set takes place locally on, or close to, the PC where the data resides. The network backbone is only used to put results together. If the I/O overhead is moderate and analysis tasks need more than one local CPU to plow through data, then each of these disk arrays could be used as a local file server to a few computers sharing a local ethernet switch. These commodity 8-port gigabit ethernet switches would be combined with a single high end, fast backplane switch allowing the connection of a thousand PCs. We have also successfully tested using Network File System (NFS) software to connect our disk arrays to computers that cannot run Linux 2.4.

RAID [2] stands for Redundant Array of Inexpensive Disks. Many industry offerings meet all of the qualifications except the inexpensive part, severely limiting the size of an array for a given budget. This is now changing. The different RAID levels can be defined as follow:

- RAID-0: "Striped." Disks are combined into one physical device where reads and writes of data are done in parallel. Access speed is fast but there is no redundancy.
- RAID-1: "Mirrored." Fully redundant, but the size is limited to the smallest disk.
- RAID-4: "Parity." For $N$ disks, 1 disk is used as a parity bit and the remaining $N-1$ disks are combined. Protects against a single disk failure but access speed is slow since you have to update the parity disk for each write. Some, but not all, files may be recoverable if two disks fail.
- RAID-5: "Striped-Parity." As with RAID-4, the effective size is that of $N-1$ disks. However, since the parity information is also distributed evenly among the $N$ drives the bottleneck of having to update the parity disk for each write is avoided. Protects against a single disk failure and the access speed is fast.

RAID-5, using enhanced integrated drive electronics (EIDE) disks under Linux software, is now available [3]. Redundant disk arrays do provide protection in the most likely single disk failure case, that in which a single disk simply stops working. This removes a major obstacle to building large arrays of EIDE disks. However, RAID-5 does not totally protect against other types of disk failures. RAID-5 will offer limited protection in the case where a single disk stops working but causes the whole EIDE bus to fail (or the whole EIDE controller card to fail), but only temporarily stops them from functioning. This would temporarily disable the whole RAID-5 array. If replacing the bad disk solves the problem, i.e. the failure did not permanently damage data on other disks, then

Table I Comparison of Large EIDE Disks for a RAID-5 Array

| Disk Model | Size (GB) | RPM | Cost/GB | GB/platter | Cache Buffer | Warranty |
|---|---|---|---|---|---|---|
| Maxtor D540X [4] | 160 | 5400 | $1.03 | 40 | 2 MB | 3 year |
| Maxtor DiamondMax 16 [5] | 250 | 5400 | $1.09 | 83 | 2 MB | 1 year |
| Maxtor MaXLine Plus II [6] | 250 | 7200 | $1.52 | 83 | 8 MB | 3 year |
| Western Digital WD2500JB [7] | 250 | 7200 | $1.31 | 83 | 8 MB | 3 year |
| IBM-Hitachi 180GXP [8] | 180 | 7200 | $1.00 | 60 | 8 MB | 3 year |

the RAID-5 array would recover normally. Similarly if only the controller card was damaged then replacing it would allow the RAID-5 array to recover normally. However, if more than one disk was damaged, especially if the file or directory structure information was damaged, the entire RAID-5 array would be damaged. The remaining failure mode would be for a disk to be delivering corrupted data. There is no protection for this inherent to RAID-5; however, a longitudinal parity check on the data, such as a checksum record count (CRC), could be built into event headers to flag the problem. Redundant copies of data that are very hard to recreate are still needed. RAID-5 does allow one to ignore backing up data that is only moderately hard to recreate.

## 2. Large Disks

In today's marketplace, the cost per terabyte of disks with EIDE interfaces is about half that of disks with SCSI (Small Computer System Interface). The EIDE interface is limited to 2 drives on each bus and SCSI is limited to 7 (14 with wide SCSI). The only major drawback of EIDE disks is the limit in the length of cable connecting the drives to the drive controller. This limit is nominally 18 inches; however, we have successfully used 24 inch long cables [9]. Therefore, one is limited to about 10 disks per box for an array (or perhaps 20 with a "double tower"). To get a large RAID array one needs to use large capacity disk drives. There have been some problems with using large disks, primarily the maximum addressable size. We have addressed these problems in an earlier papers [10, 11]. Because of these concerns and because we wanted to put more drives into an array than could be supported by the motherboard we opted to use PCI disk controller cards. In the past we have tested both Promise Technologies ULTRA 66 and ULTRA 100 disk controller cards in RAID-5 disk arrays consisting of either 80 or 100 GB disks[11]. Each of the PCI disk controller cards support four drives. We now report on our tests of the Promise Technologies ULTRA 133 TX2 [12] that supports disk drives with capacity greater than 137 GB.

Using arrays of disk drives, as shown in Table I, the cost per terabyte is similar to that of cost of Storage Technology tape silos. However, RAID-5 arrays offer a lot better granularity since they are scalable down to a terabyte. For example, if you wanted to store 10 TB of data you would still have to pay about $1,000,000 for the tape silo but only $20,000 for a RAID-5 array. Thus, even small institutions can afford to deploy systems. And the Terabyte disk arrays can be used as caches to take full advantage of Grid Computing [13].

## 3. RAID Arrays

There exist disk controllers that implement RAID-5 protocols right in the controller, for example 3ware's Escalade 7500 series [14], which will handle up to 12 EIDE drives. These controllers cost $600 and, at the time that we built the system shown in Table III, did not support disk drives larger than 137 Gigabytes [15]. Therefore, we focused our attention on software RAID-5 implementations [3, 16], which we tested extensively.

There are also various commercial RAID systems that rely on a hardware RAID controller. Examples of these are shown in Table II. They are typically 3U or larger rack mounted systems. However, commercial systems have not been off-the-shelf commodity items. This is changing and the only drawback is that, even allowing for cost of assembly, they are anywhere from twice to over twenty-five times as expensive.

Table II Some Commodity Hardware RAID Arrays.

| System | Capacity | Size | Price/GB [a] |
|---|---|---|---|
| Apple Xserve RAID | 2.52 TB | 3U | $4.36 |
| Dell EMC CX200 | 2.2 TB | 3U | $13.63 |
| HP 7100 | 2.2 TB | 2×3U | $50.21 |
| IBM DF4000R | 2.2 TB | 2×3U | $20.08 |
| Sun StorEdge T3 | 2.64 TB | 3×3.5U | $54.66 |

[a]Based on suggested retail Prices on February 7, 2003[17]

## 3.1. Hardware

We now report on the use of disks with capacity greater than 137 GB. The drives we consider for use with a RAID-5 array are compared in Table I. The disk we tested was the Maxtor D540X 160 GB disk [4]. In general, the internal I/O speed of a disk is proportional to its rotational speed and increases as a function of platter capacity. One should note that the "spin–up" of these drives takes 1.8-2.5 Amps at 12 Volts (typically 22 W total for both 12 V and 5V).

When assembling an array we had to worry about the "spin-up" current draw on the 12V part of the power supply. With 8 disks in the array (plus the system disk) we would have exceeded the capacity of the power supply that came with our tower case, so we decided to add a second off-the-shelf power supply rather than buying a more expensive single supply. By using 2 power supplies we benefit from under loading the supplies. The benefits include both a longer lifetime and better cooling since the heat generated is distributed over 2 supplies, each with their own cooling fans. We used the hardware shown in Table III for our array test. Many of the components we chose are generic; thus, components from other manufacturers also work. We have measured the wall power consumption for the whole disk array box in Table III. It uses 276 watts at startup and 156 watts during normal sustained running.

Table III Components used in our 1 Terabyte RAID-5 disk array

| System Component | Unit Price |
|---|---|
| 40 GB IBM system disk [18] | $65 |
| 8 – 160 GB Maxtor RAID-5 disks [4] | $170 |
| 2 – Promise ATA/133 PCI cards [12] | $32 |
| 4 – StarTech 24" ATA/100 cables [9] | $3 |
| AMD Athlon 1.9 GHz/266 CPU [19] | $77 |
| Asus A7M266 motherboard, audio [20] | $67 |
| 2 – 256 MB DDR PC2100 DIMMs | $33 |
| In-Win Q500P Full Tower Case [21] | $77 |
| Sparkle 15A @ 12V power supply [22] | $34 |
| 2 – Antec 80mm ball bearing case fans | $8 |
| 110 Alert temperature alarm [23] | $15 |
| Pine 8 MB AGP video card [24] | $15 |
| SMC EZ card 10/100 ethernet [25] | $12 |
| Toshiba 16x DVD, 48x CDROM | $36 |
| Sony 1.44 MB floppy drive | $12 |
| KeyTronic 104 key PS/2 keyboard | $7 |
| DEXXA 3 button PS/2 mouse | $4 |
| Total | $1922 |

To install the second power supply we had to modify our tower case with a jigsaw and a hand drill. We also had to use a jumper to ground the green wire in the 20-pin block ATXPWR connector to fake the power-on switch.

When installing the two disk controller cards care had to be taken that they did not share interrupts with other highly utilized hardware such as the video card and the ethernet card. We also tried to make sure that they did not share interrupts with each other. There are 16 possible interrupt requests (IRQs) that allow the various devices, such as EIDE controllers, video cards, mice, serial, and parallel ports, to communicate with the CPU. Most PC operating systems allow sharing of IRQs but one would naturally want to avoid overburdening any one IRQ. There are also a special class of IRQs used by the PCI bus, they are called PCI IRQs (PIRQ). Each PCI card slot has 4 interrupt numbers. This means that they share some IRQs with the other slots; therefore, we had to juggle the cards we used (video, 2 EIDE controllers, and an ethernet).

When we tried to use a disk as a "Slave" on a motherboard EIDE bus, we found that it would not run at the full speed of the bus and slowed down the access speed of the entire RAID-5 array. This was a problem of either the motherboard's basic input/output system (BIOS) or EIDE controller. This problem was not in evidence when using the disk controller cards. Therefore, we decided that rather than take a factor of 10 hit in the access speed we would rather use 8 instead of 9 hard disks.

## 3.2. Software

For the actual tests we used Linux kernel 2.4.17 with the RedHat 7.2 (see http://www.redhat.com/) distribution (we had to upgrade the kernel to this level) and applied a patch to allow support for greater than 137 GB disks (see http://www.kernel.org/ and see http://www.linuxdiskcert.org/). The latest stable kernel version is 2.4.20 (see http://www.kernel.org/). We needed the 2.4.x kernel to allow full support for "Journaling" file systems. Journaling file systems provide rapid recovery from crashes. A computer can finish its boot-up at a normal speed, rather than waiting to perform a file system check (FSCK) on the entire RAID array. This is then conducted in the background allowing the user to continue to use the RAID array. There are now 4 different Linux Journaling file systems: XFS, a port from SGI [26]; JFS, a port from IBM [27]; ext3 [28], a Journalized version of the standard ext2 file system; and ReiserFS from namesys [29]. Comparisons of these Journaling file systems have been done elsewhere [30]. When we tested our RAID-5 arrays only ext3 and the ReiserFS were easily available for the 2.4.x kernel; therefore, we tested 2 different Journaling file systems; ReiserFS and ext3.

We opted on using ext3 for two reasons: 1) At the time there were stability problems with ReiserFS and NFS (this has since been resolved with kernel 2.4.7) and 2) it was an extension of the standard ext2fs (it was originally developed for the 2.2 kernel) and, if synced properly could be mounted as ext2. Ext3 is the only one that will allow direct upgrading from ext2, this is why it is now the default for RedHat since 7.2.

NFS is a very flexible system that allows one to manage files on several computers inside a network as if they were on the local hard disk. So, there's no need to know what actual file system they are stored under nor where the files are physically located in order to access them. Therefore, we use NFS to connect these disks arrays to computers that cannot run Linux 2.4. We have successfully used NFS to mount disk arrays on the following types of computers: a DECstation 5000/150 running Ultrix 4.3A, a Sun UltraSparc 10 running Solaris 7, a Macintosh G3 running MacOS X, and various Linux boxes with both the 2.2 and 2.4 kernels.

As an example, in Spring 2002 we built a pair of one Terabyte Linux RAID-5 arrays, as described in section 3.1, to store CMS Monte Carlo data at CERN. They were mounted using NFS, via gigabit ethernet. They remotely served the random background data to the CMS Monte Carlo Computers, as if it was local. While this is not as efficient as serving the data directly, it is clearly a viable technique [31]. We also are currently using two, NFS mounted, RAID-5 boxes, one at SLAC and one at the University of Mississippi, to run analysis software with the BaBar KANGA and CMS CMSIM/ORCA code.

We have performed a few simple speed tests. The first was "hdparm -tT /dev/xxx". This test simply reads a 64 MB chunk of data and measures the speed. On a single drive we saw read/write speeds of about 30 MB/s. The whole array saw an increase to 95 MB/s. When we tried writing a text file using a simple FORTRAN program (we wrote "All work and no play make Jack a dull boy" $10^8$ times), the speed was about 95 MB/s While mounted via NFS over 100 Mb/s ethernet the speed was 2.12 MB/s, limited by both the ethernet speed and the NFS communication overhead. In the past [1], we have been able to get much higher fractions of the rated ethernet bandwidth by using the lower level TCP/IP socket protocol [32] in place of the higher level NFS protocol. TCP/IP sockets are more cumbersome to program, but are much faster.

We also tested what actually happens when a disk fails by turning the power off to one disk in our RAID-5 array. One could continue to read and write files, but in a "degraded" mode, that is without the parity safety net. When a blank disk was added to replace the failed disk, again one could continue to read and write files in a mode where the disk access speed is reduced while the system rebuilt the missing disk as a background job. This speed reduction in disk access was due to the fact that the parity regeneration is a major disk access in its own right. For more details, see reference [16].

The performance of Linux IDE software drivers is improving. The latest standards [33] include support for command overlap, READ/WRITE direct memory access QUEUED commands, scatter/gather data transfers without intervention of the CPU, and elevator seeks. Command overlap is a protocol that allows devices that require extended command time to perform a bus release so that commands may be executed by the other device on the bus. Command queuing allows the host to issue concurrent commands to the same device. Elevator seeks minimize disk head movement by optimizing the order of I/O commands. The Hitachi/IBM 180GXP disk [8] supports elevator seeks under the new ATA6 standard [33].

We did encounter a few problems. We had to modify "MAKEDEV" to allow for more than eight IDE devices, that is to allow for disks beyond "/dev/hdg". For version 2.x one would have to actually modify the script; however, for version 3.x we just had to modify the file "/etc/makedev.d/ide". This should no longer be a problem with newer releases of Linux.

Another problem was the 2 GB file size limit. Older operating system and compiler libraries used a 32 bit "long-integer" for addressing files; therefore, they could not normally address files larger than 2 GB ($2^{31}$). There are patches to the Linux 2.4 kernel and glibc but there are still some problems with NFS and not all applications use these patches.

We have found that the current underlying file systems (ext2, ext3, reiserfs) do not have a 2 GB file size limit. The limit for ext2/ext3 is in the petabytes. The 2.4 kernel series supports large files (64-bit offsets). Current versions of GNU libc support large files. However, by default the 32-bit offset interface is used. To use 64-bit offsets, C/C++ code must be recompiled with the following as the first line:

```
#define _FILE_OFFSET_BITS 64
```

or the code must use the *64 functions (i.e. open becomes open64, etc.) if they exist. This functionality is not included in GNU FORTRAN (g77); however, it should be possible to write a simple wrapper C program to replace the OPEN statement (perhaps called open64). We have succeeded in writing files larger than 2 GB using a simple C program with "#define _FILE_OFFSET_BITS 64" as the first line. This works over NFS version 3 but not version 2.

While RAID-5 is recoverable for a hardware failure, there is no protection against accidental deletion of files. To address this problem we suggest a simple script to replace the "rm" command. Rather than deleting files it would move them to a "/raid/Trash" or better yet a "/raid/.Trash" directory on the RAID-5 disk array (similar to the "Trash can" in the Macintosh OS). The system administrator could later purge

them as space is needed using an algorithm based on criteria such as file size, file age, and user quota.

## 4. High Energy Physics Strategy

We encapsulate data and CPU processing power. A block of real or Monte Carlo simulated data for an analysis is broken up into groups of events and distributed once to a set of RAID disk boxes, which each may also serve a few additional processors via a local 8-port gigabit ethernet switch (see Figure 1).

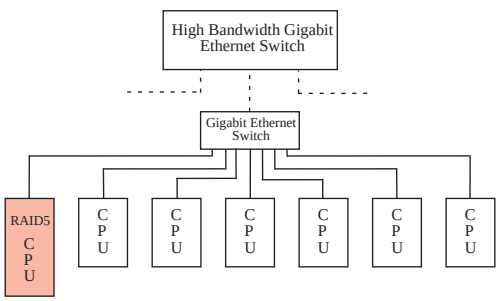


Figure 1: An example of a RAID-5 disk array mounted on several local CPUs via a 8-port gigabit switch.

Examples of commodity gigabit ethernet switches and PCI adapters are seen in Table IV. Dual processor boxes would also add more local CPU power. Events are stored on disks close to the CPUs that will process them to minimize I/O. Events are only moved once. Event parallel processing has a long history of success in high energy physics [1, 40, 41]. The data from each analysis are distributed among all the RAID arrays so all the computing power can be brought to bear on each analysis.

Table IV Examples of Commodity Gigabit Ethernet Switches and Adapters.

| Company | Model | Type | Cost |
|---|---|---|---|
| Linksys [34] | EG008W | 8-port switch | $162 |
| D–Link [35] | DGS–1008T | 8-port switch | $312 |
| Netgear [36] | GS508T | 8-port switch | $502 |
| Netgear [37] | GS524T | 24-port switch | $1499 |
| D–Link [38] | DGE500T | PCI adapter | $46 |
| Intel [39] | 82540EM | PCI adapter | $41 |

For example, in the case of an important analysis (such as a Higgs analysis), one could put 50 GB of data onto each of 100 RAID arrays and then bring the full computing power of 700 CPUs into play. Instances of an analysis job are run on each local cluster in parallel. Several analyses jobs may be running in memory or queued to each local cluster to level loads. The data volume of the results (e.g. histograms) is small and is gathered together over the network backbone. Results are examined and the analysis is rerun. The system is inherently fault tolerant. If three of a hundred clusters are down, one still gets 97% of the data and analysis is not impeded.

RAID-5 arrays should be treated as fairly secure, large, high-speed "scratch disks". RAID-5 just means that disk data will be lost less frequently. Data which is very hard to re-create still needs to reside on tape. The inefficiency of an offline tape vault can be an advantage. Its harder to erase your entire raw data set with a single keystroke, if thousands of tapes have to be physically mounted. Someone may ask why all the write protect switches are being reset before all is lost. Its the same reason the Air Force has real people with keys in ICBM silos.

The granularity offered by RAID-5 arrays allows a university or small experiment in a laboratory to set up a few terabyte computer farm, while allowing a large Analysis Site or Laboratory to set up a few hundred terabyte or a petabyte computer system. For a large site, they would not necessarily have to purchase the full system at once, but buy and install the system in smaller parts. This would have two advantages, primarily they would be able to spread the cost over a few years and secondly, given the rapid increase in both CPU power and disk size, one could get the best "bang for the buck".

What would be required to build a 1/4 petabyte system (similar size as a tape silo)? Start with eight 250 GB Maxtor disks in a box. The Promise Ultra133 card allows one to exceed the 137 GB limit. Each box provides $7 \times 250\,\text{GB} = 1750\,\text{GB}$ of usable RAID-5 disk space in addition to a CPU for computations. 280 terabytes is reached with 161 boxes. Use 23 commodity 8-port gigabit ethernet switches ($170 each) to connect the 161 boxes to a 24-port commodity gigabit ethernet switch. See Figure 2. This could easily fit in a room that was formerly occupied by a few old Mainframes, say an area of about a hundred square meters. The power consumption would be 25 kilowatts, 45 kilowatts if they all start up at once. One would need to build up operational experience for smooth running. As newer disks arrive that hold yet more data, even a petabyte system would become feasible. If one still needed more processing power per RAID array you could substitute for each RAID-5 CPU shown in Figure 2, 6 CPUs plus 1 RAID-5 CPU connected by an 8-port gigabit ethernet switch as described in Figure 1. Multiple CPUs per motherboard provide another alternative to adjust the disk space to processing power ratio.

Grid Computing [13] will entail the movement of large amounts of data between various sites. RAID-5 arrays will be needed as disk caches both during the transfer and when it reaches its final destination. Another example that can apply to Grid Computing is the Fermilab Mass Storage System, Enstore [42],

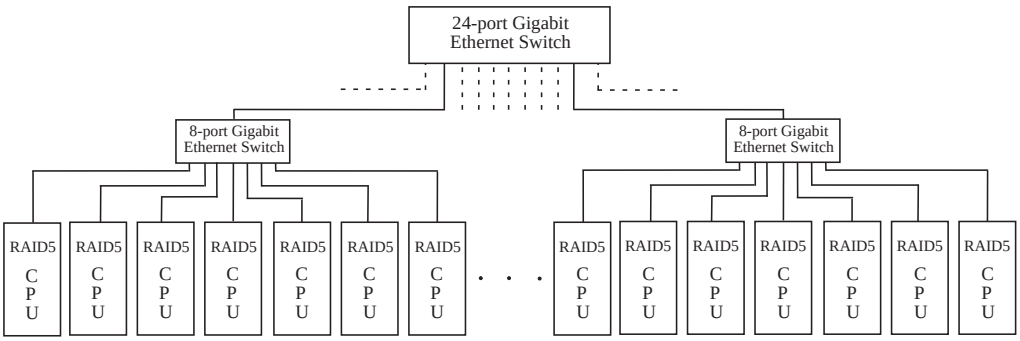


Figure 2: A schematic of a 1/4 petabyte (or larger) system.

where RAID arrays are used as a disk cache for a Tape Silo. Enstore uses RAID arrays to stage tapes to disk allowing faster analysis of large data sets.

## 5. Conclusion

We have tested redundant arrays of IDE disk drives for use in offline high energy physics data analysis and Monte Carlo simulations. Parts costs of total systems using commodity IDE disks are now at the $2000 per terabyte level, a lower cost per terabyte than Storage Technology tape silos. The disks, however, offer much better granularity; even small institutions can afford them. The faster access of disk versus tape is a major added bonus. We have tested software RAID-5 systems running under Linux 2.4 using Promise Ultra 133 disk controllers. RAID-5 provides parity bits to protect data in case of a single catastrophic disk failure. Tape backup is not required for data that can be recreated with modest effort. Journaling file systems permit rapid recovery from crashes. Our data analysis strategy is to encapsulate data and CPU processing power. Data is stored on many PCs. Analysis for a particular part of a data set takes place locally on the PC where the data resides. The network is only used to put results together. Commodity 8-port gigabit ethernet switches combined with a single high end, fast backplane switch [43] would allow one to connect over a thousand PCs, each with a terabyte of disk space. Some tasks may need more than one CPU to go through the data even on one RAID array. For such tasks dual CPUs and/or several boxes on one local 8-port ethernet switch should be adequate and avoids overwhelming the backbone switching fabric connecting an entire installation. Again the backbone is only used to put results together.

Current high energy physics experiments, such as BaBar at SLAC, feature relatively low data acquisition rates, only 3 MB/s, less than a third of the rates taken at Fermilab fixed target experiments a decade ago [1]. The Large Hadron Collider experiments CMS and Atlas, with data acquisition rates starting at 100 MB/s, will be more challenging and require physical architectures that minimize helter skelter data movement if they are to fulfill their promise. In many cases, architectures designed to solve particular processing problems are far more cost effective than general solutions [1, 40]. As Steve Wolbers in his talk at CHEP03 [44] reminded us, all data processing groups can not depend on Moore's Law to save them. Data acquisition groups want to write out additional interesting events. Programmers like to adopt new languages that are further abstracted from the CPUs running them. Small objects and pointers seem to find their way into code. Machines hate to interrupt pipelines and love direct addressing. Universities want networks to transfers billions of events quickly. Even Gordon Moore may not be able to do all of this simultaneously. Efficiency may still be useful. Designing time critical code [45], regardless of the language chosen, to fit into larger blocks without pointers can increase speed by a factor of 10 to 100. Code to methodically bit-pack events into the minimum possible size may be worth writing [46]. If events are smaller, more can be stored on a given disk and more can be transferred over a given network in a day. All of this requires planning at an early stage. No software package will generate it automatically.

Techniques explored in this paper, physically encapsulating data and CPUs together, may be useful. Terabyte disk arrays at small institutions are now feasible. Computing has progressed since the days when science was done by punching a few kilobytes into paper tape [47].

## Acknowledgments

Many thanks to S. Bracker, J. Izen, L. Lueking, R. Mount, M. Purohit, W. Toki, and T. Wildish for their help and suggestions. This work was supported in part by the U.S. Department of Energy under Grant Nos. DE-FG05-91ER40622 and DE-AC02-76CH03000.

## References


[1] For example, a decade ago the Fermilab E791 collaboration recorded and reconstructed 50 TB of raw data in order to generate charm physics results. For details of the saga, in which more data was written to tape than in all previous HEP experiments combined, see:
S. Amato, J. R. de Mello Neto, J. de Miranda, C. James, D. J. Summers and S. B. Bracker, Nucl. Instrum. Meth. A **324**, 535 (1993) [arXiv:hep-ex/0001003];
S. Bracker and S. Hansen, [arXiv:hep-ex/0210034];
S. Hansen, D. Graupman, S. Bracker and S. Wickert, IEEE Trans. Nucl. Sci. **34**, 1003 (1987);
S. Bracker, K. Gounder, K. Hendrix and D. Summers, IEEE Trans. Nucl. Sci. **43**, 2457 (1996) [arXiv:hep-ex/9511009];
E. M. Aitala *et al.* [E791 Collaboration], Phys. Rev. Lett. **77**, 2384 (1996) [arXiv:hep-ex/9606016];
E. M. Aitala *et al.* [E791 Collaboration], Phys. Rev. Lett. **80**, 1393 (1998) [arXiv:hep-ph/9710216];
E. M. Aitala *et al.* [E791 Collaboration], Phys. Rev. Lett. **83**, 32 (1999) [arXiv:hep-ex/9903012];
E. M. Aitala *et al.* [E791 Collaboration], Phys. Lett. B **403**, 377 (1997) [arXiv:hep-ex/9612005];
E. M. Aitala *et al.* [E791 Collaboration], Phys. Lett. B **462**, 401 (1999) [arXiv:hep-ex/9906045];
E. M. Aitala *et al.* [E791 Collaboration], Phys. Rev. D **57**, 13 (1998) [arXiv:hep-ex/9608018].

[2] D. A. Patterson, G. Gibson and R. H. Katz, Sigmod Record **17**, 109 (1988).

[3] M. de Icaza, I. Molnar, and G. Oxman, "The linux raid-1,4,5 code," in *3rd Annu. Linux Expo'97*, (April 1997).

[4] Maxtor. (2001) DiamondMax D540X. http://www.maxtor.com/en/documentation/data_sheets/d540x_datasheet.pdf [2003].

[5] Maxtor. (2003) DiamondMax 16. http://www.maxtor.com/en/documentation/data_sheets/diamondmax_16_data_sheet.pdf [2003].

[6] Maxtor. (2003) Maxline ATA. http://www.maxtor.com/en/documentation/data_sheets/maxline_data_sheet.pdf [2003].

[7] Western Digital Corp. (2003) Specifications for the WD Caviar WD2500JB. http://www.wdc.com/en/products/current/drives.asp?Model=WD2500JB [2003].

[8] Hitachi Global Storage Technologies. (2003) Deskstar 180GXP. http://www.hgst.com/hdd/desk/ds180gxp.htm and http://ssddom01.hgst.com/tech/techlib.nsf/techdocs/CF02BAB6EA8E3B7F87256C16006B1CFA/$file/D180GXP_ds.pdf [2003].

[9] StarTech.com. (2002) 24 In. Dual Drive Ultra ATA/66/100 Cable. http://www.startech.com/ststore/itemdetail.cfm?product_id=IDE66_24 [2003].
The ATA/100 standard uses an 80 wire cable to transmit up to 133 Megabytes per second.

[10] D. Sanders, C. Riley, L. Cremaldi, D. Summers and D. Petravick, in *Proc. Int. Conf. Computing in High- Energy Physics (CHEP 98)*, Chicago, IL, (Aug. 31 - Sep 4 1998) [arXiv:hep-ex/9912067].

[11] D. A. Sanders, L. M. Cremaldi, V. Eschenburg, C. N. Lawrence, C. Riley, D. J. Summers and D. L. Petravick, IEEE Trans. Nucl. Sci. **49**, 1834 (2002) [arXiv:hep-ex/0112003].

[12] Promise Technologies, inc. (2001) Ultra133 TX2 – Ultra ATA/133 Controller for 66 MHz PCI Motherboards. http://www.promise.com/marketing/datasheet/file/U133_TX2_DS.pdf [2003] and http://www.promise.com/marketing/datasheet/file/Ultra133tx2DS_v2.pdf [2003].
Each ATA/PCI Promise card controls four disks.

[13] P. Avery, Phil. Trans. Roy. Soc. Lond. **360**, 1191 (2002);
L. Lueking *et al.* , Lect. Notes Comput. Sci. **2242**, 177 (2001).

[14] 3ware. (2003) Escalade 7500 Series ATA RAID Controller. http://www.3ware.com/products/pdf/Escalade7500SeriesDS1-7.qk.pdf [2003];
3ware. (2003) Escalade 7500-12 ATA RAID Controller. http://www.3ware.com/products/pdf/12-PortDS1-7.qk.pdf [2003].

[15] K. Abendroth, "personal communication," email: kent.abendroth@3ware.com.

[16] J. Østergaard. (2000) The software-RAID HOWTO. http://www.linuxdoc.org/HOWTO/Software-RAID-HOWTO.html.

[17] Apple Computers. (2003) Xserve RAID. http://www.apple.com/server/pdfs/L26325A_XserveRAID_TO.pdf [2003].

[18] IBM. (2002) IBM Deskstar 60GXP hard disk drive. http://ssddom01.hgst.com/tech/techlib.nsf/techdocs/85256AB8006A31E587256A7600736475/$file/D60GXP_ds.pdf [2003].

[19] AMD. (2002) AMD Athlon Processor Product Brief. http://www.amd.com/us-en/Corporate/VirtualPressRoom/0,,51_104_543~24415,00.html [2003]
We bought our AMD CPU boxed with a fan.

[20] ASUS. (2002) ASUS A7M266. http://www.asus.com/mb/socketa/a7m266/overview.htm [2003].

[21] In-Win Development, inc. (2002) IW-Q500 ATX Full Tower Case. http://www.in-win.com.tw/home/detail.php?show=features&event=ATX&class=Full_Tower&type=Q-Series&model=IW-Q500 [2003]
Note: the Q500P case comes with a 300 Watt power supply.

[22] Sparkle Power Inc. (2002) FSP300-60BTV For P4 and Athlon. http://www.sparklepower.com/pdf/FSP300-60BTV.pdf [2003]
The Sparkle FSP300-60BTV is used as a second 300 watt supply. At 12 volts it gives 15 amps.

[23] PC Power & Cooling, Inc. (2000) 110 ALERT Computer Over-Temperature Alarm. http://www.pcpowercooling.com/pdf/110Alert_ds.pdf [2003].

[24] PINE Technology. (2001) VGA Card Specification. http://www.pinegroup.com/pdf/S2/L1/product_list.pdf [2003];
nVIDIA. (2002) nVIDIA Consumer Desktop Solutions. http://www.nvidia.com/docs/lo/962/SUPP/NV_LC_02.05.02B.pdf [2003]
The Pine 8MB AGP NVIDIA VANTA LT video card is used to run a monitor for diagnostics.

[25] SMC Networks. (2001) EZ Card 10/100. http://www.smc.com/drivers_downloads/library/SMC1211.pdf [2003];
D. Becker. (1999) A RealTek RTL8129/8139 Fast Ethernet driver for Linux. http://www.smc.com/drivers_downloads/library/rtl8139.c [2003]
The SMC 1211TX EZ PCI Card uses an rtl8139 10/100 ethernet software driver.

[26] SGI. (2001) XFS: A high-performance journaling file system. http://oss.sgi.com/projects/xfs/ [2003].

[27] S. Best. (2002) Journaled File System Technology for Linux. http://www-124.ibm.com/developerworks/oss/jfs/ [2003].

[28] A. Morton. (2002) ext3 for 2.4. http://www.zip.com.au/∼akpm/linux/ext3/ [2003].

[29] H. Reiser. (2001) Three reasons why ReiserFS is great for you. http://www.reiserfs.org/ [2003].

[30] R. Galli. (2001) Journal File Systems in Linux. *Upgrade*. Vol. II(6), pp. 1-8, http://www.upgrade-cepis.org/issues/2001/6/up2-6Galli.pdf [2003].

[31] V. Lefebure and T. Wildish, "The spring 2002 DAQ TDR production," CERN-CMS-NOTE-' 2002-034. See http://cmsdoc.cern.ch/documents/02/note02_034.pdf [2003].

[32] S. Feit, *TCP/IP: Architecture, Protocols, and Implementation*. NewYork: McGraw–Hill, (1993).

[33] P. McLean. (2002) Technical Committee T13 AT Attachment. http://www.t13.org/ [2003].

[34] Linksys. (2003) Gigabit 8-Port Workgroup Switch. ftp://ftp.linksys.com/datasheet/eg008w_ds.pdf [2003].

[35] D-Link Systems. (2001) DGS - 1008T. http://www.dlink.com/products/switches/dgs1008t/dgs1008t.pdf [2003].

[36] NETGEAR Inc. (2001) 8 port 10/100/1000 Mbps Copper Gigabit Switch. http://www.netgear.com/pdf_docs/gs508t.pdf [2003].

[37] NETGEAR Inc. (2001) 24 port 10/100/1000 Mbps Copper Gigabit Switch. http://www.netgear.com/pdf_docs/GS524T_Product_Sheet.pdf [2003].

[38] D-Link Systems. (2001) DGE - 550T. http://www.dlink.com/products/gigabit/dge550t/dge550t.pdf [2003].

[39] Intel Corporation. (2002) Intel PRO/1000 MT Desktop Adapter. http://www.intel.com/network/connectivity/resources/doc_library/data_sheets/pro1000mt_da.pdf [2003].

[40] For a description of Fermilab's first UNIX farm: C. Stoughton and D. J. Summers, Comput. Phys. **6**, 371 (1992), [arXiv:hep-ex/0007002];
C. Gay and S. Bracker, IEEE Trans. Nucl. Sci. **34**, 870 (1987).

[41] P. F. Kunz, R. N. Fall, M. F. Gravina, J. H. Halperin, L. J. Levinson, G. J. Oxoby, and Q. H. Trang, IEEE Trans. Nucl. Sci.**27** 582 (1980).

[42] Fermilab (2002) Fermilab Mass Storage System – Enstore. http://www.fnal.gov/docs/products/enstore/html/intro.html [2003];
D. Petravick, in *Proc. Int. Conf. Computing in High Energy and Nuclear Physics (CHEP 2000)*, Padova, Italy, (7-11 Feb 2000) 630-633.

[43] For example, the Cisco Catalyst 3750 Switch has a 32 Gigabits per second backplane capacity. Up to nine cards may be installed in this switch. One option, the 3750G-24T card, has 24 full duplex 10/100/1000 Base-TX ports. See:
Cisco Systems. (2003) Cisco Catalyst 3750 Series Switches. http://www.cisco.com/warp/public/cc/pd/si/casi/ps5023/prodlit/cat50_ds.pdf [2003].

[44] S. Wolbers, "Computing exprience from CDF and D0," in *Proc. Int. Conf. Computing in High Energy and Nuclear Physics (CHEP 2003)*, La Jolla, California, (March 24-28 2003), these proceedings. http://chep03.ucsd.edu/files/10002.pdf

[45] M. Metcalf, "FORTRAN Optimization," San Diego, Academic Press, ISBN 0124924808 (1985).

[46] S. Bracker, "Proposed data format for B factory events," SLAC-BABAR-NOTE-129 (1994).

[47] B. M. Lasker, S. B. Bracker and W. E. Kunkel, Publ. Astron. Soc. Pac. **85**, 109 (1973).