Putting all that (HEP-) Data to work – a REAL Implementation of an Unlimited Computing and Storage Architecture

Michael Ernst

Deutsches Elektronen - Synchrotrons DESY, Notkestrasse 85, 22607 Hamburg, Germany

Since Computing in HEP left the 'Mainframe-Path', many institutions demonstrated a successful migration to workstation-based computing, especially for application requiring a high CPU-to-I/O ratio. However, the difficulties and the complexity starts beyond just providing CPU-Cycles. Critical applications, requiring either sequential access to large amounts of data or to many small sets out of a multi 10-Terabyte Data Repository need technical approaches we haven't had so far. Though we felt that we were hardly able to follow technology evolving in the various fields, we recently had to realize that even politics overtook technical evolution - at least in the areas mentioned above. The USA is making peace with Russia. DEC is talking to IBM, SGI communicating with HP. All these things became true, and though, unfortunately, the Cold War lasted 50 years, and -in a relative sense- we were afraid that '50 years' seemed to be how long any self respecting high performance computer (or a set of workstations) had to wait for data from its 'Server', fortunately, we are now facing a similar progress on friendliness, harmony and balance in the former problematic (computing-) areas. Buzzwords, mentioned many thousand times in talks describing today's and future requirements, including Functionality, Reliability, Scalability, Modularity and Portability are not just phrases, wishes and dreams any longer.

At DESY, we are in the process of demonstrating an architecture that is taking those five issues equally into consideration, including Heterogeneous Computing Platforms with ultimate file system approaches, Heterogeneous Mass Storage Devices and an Open Distributed Hierarchical Mass Storage Management System.

This contribution will provide an overview on how far we got and what the next steps will be.

1 Introduction

To cope with the increasing computing power required for HEP experiments, High-Energy Physics Laboratories started around 1990 to set up Workstation Clusters. In the early days many of those were set up 'dataless', meaning that basically no local Mass Storage Devices were directly connected. Instead, the architects decided to get those machines connected to a mainframe environment where the were able to find appropriate space, efficient management and most important, a reliable place for the data. The main user community was still using the mainframe as their primary computing platform, hence, our architects were trying to design the interface to the batch-environment related to the cluster as transparent as possible. On completion of the jobs the results were spooled back onto the mainframe disks, of course. Reasons for such hybrids were therefore simplicity (user perspective), and probably the high confidence level concerning data integrity.

However, the world moved much faster than many might have expected. Due to the progressively increasing amount of data and the fact that CPU technology offered a 2X every 18 months, the network interconnect between both worlds became the major bottleneck. Also, things were fine as long as the data could fit onto the

existing mainframe disks, because even in the early '90s the price-capacity ratio between mainframe and SCSI disks was at least in the order of 3. Not to mention the enormous maintenance cost for mainframe storage peripherals.
So, things worked pretty sufficient and kept scientists happy as long as they had especially CPU-intensive jobs, like any kind of simulation, running for days on the workstations and just producing a 50 MByte dataset.

2 Requirements and the resulting Dilemma

Today, the mainframes basically disappeared from the relevant computing environment for the physicists. Most of the accelerator laboratories moved to Distributed Heterogeneous Computing involving multiple Workstation Vendors products. At small scale installations Mass Storage Devices, like Disk and Tape Drives, are locally attached to one or multiple machines and NFS crossmounts give a homogeneous file system view. However, this setup might be adequate for simulation farms, but it is certainly not for Physics Analysis.
Though physics related questions have moved to different topics over time the overall data characteristics stay the same. Events are analyzed in any order, in many cases in parallel. Historically, HEP data sets have been orders of magnitude too large for affordable random access storage, so the HEP analysis principle is still oriented towards large, sequential files stored on magnetic tapes.
However, at the event level, access to HEP data is intrinsically random; even within events, particular studies may make very selective use of data. The ideal data storage system for a year >2000 experiment would offer random access to multi 100 terabytes of structured data selected from a multi petabyte tape store, with the granularity going down to to the level of a few bytes. The bandwidth required by analysis processes will be in the order of 100 MBytes/s at minimal latency.
Looking at these terrifying numbers only the blue-eyed believes that we can handle that with todays approaches, like file based access through unrelated stage requests, initiated by the average physicist out of a group of 200 others working basically on the same set of data.

3 Architecture and Topology

During the past 5 years experience, gathered in the major accelerator labs (CERN, Fermilab, DESY), has shown that the new computing era needs concepts and models allowing for service distribution. The reason is a certain architecture might be excellent on CPU-cycle delivery, but rather limited on connection of Mass Storage Peripherals and Data Access, respectively. Another architecture, on the other hand, has a very powerful 1/0 Subsystem, but can't get us the CPU we are looking for.
So, the natural conclusion was to allocate functionality were its appropriate, however, as there is no free lunch, on the expense of an additional functional block, the Network, which in the past had only limited importance, especially in batch computing.
Functional blocks are:

- CPU Servers

. Data Servers (Tape and Disk)

. Backplane (Network)

While there is nothing special on CPU Servers (main point here is an adequate network interface), we need to elaborate on the architecture of the Data Servers, both Disk and Tape Servers.

3.1 Disk and Tape Servers

When thinking of remote access to disk based file systems, the Network File System (NFS) comes to many people's mind. No question, this is a very convenient solution; on the other hand, current practise proves that raw data rates of 5 to 7 MBytes/s per Disk Drive decreases to 1 MByte/s by the time the data has traversed the protocol stack labyrinth (both disk access and network related).
Hence, network based transport protocol alternatives, based upon TCP/IP sockets were chosen in order to avoid performance penalties. Still the remote file systems are NFS-mounted, however, this path is just used for administrative purposes, strictly not for data transfer. A package offering this kind of functionality is called RFIO, also part of CERN's SHIFT software.
There is a very important detail concerning the backplane. As the backplane is not necessarily homogeneous, meaning it's *not* made out of a single network technology, like FDDI, HIPPI, ATM or FCS, a method had to be selected allowing to transparently forward data through different technology approaches. The answer was, as mentioned already, the Internet Protocol Stack 1P. The described architecture was implemented for the ZEUS experiment, including a Data Server, based on a SGI Challenge DM with a HIPPI interface. The NetStar GigaRouter [1], a Mixed Media Router, transparently forwarding packets between HIPPI, FDDI and ATM, forms the backplane. Sustained data rates of more than 30 MBytes/s (multiple streams) into the ZEUS batch machines are deliverd at a cost of 30% of the available CPU on the Challenge DM. Again, the DM is nothing but an '1/0 Crossbar' between a large disk farm (> 200 Disk Drives) and the network. Similar approaches were taken in order to achieve Network Attached Tape Devices. We choose FDDI attachment for our STK 3490s by means of a RS/6000 based 'Controller', because the limited data transfer rate of 3 MBytes/s makes FDDI perfectly suitable. On the other hand, connecting a number of high performance D2 or D3 helical tape drives to an '1/0 Crossbar', HIPPI attachment gives a good balance on the network side.

Besides Disk 1/0 related protocol (Software) issues we in HEP get often hurt by internal limitations of current 32 bit UNIX File Systems (UFS) in terms of maximum size and whenever the file system gets corrupted. The size is currently limited to 2 GByte, unless there is a Volume Manager which allows to group a number of, let's say 4, physical file systems to a logical volume. Even 8 GByte is by far too small for us. Another problem, especially at large installations with multi 100 TByte of connected disk capacity, is the time the salvaging process needs once inconsistencies show up. Though vendors, like SGI, allow to check several file systems in parallel, it was very likely at DESY that this process lasts more than 1 hour. Though we were able to get around the size-problem (for example by using the 'Disk Pool

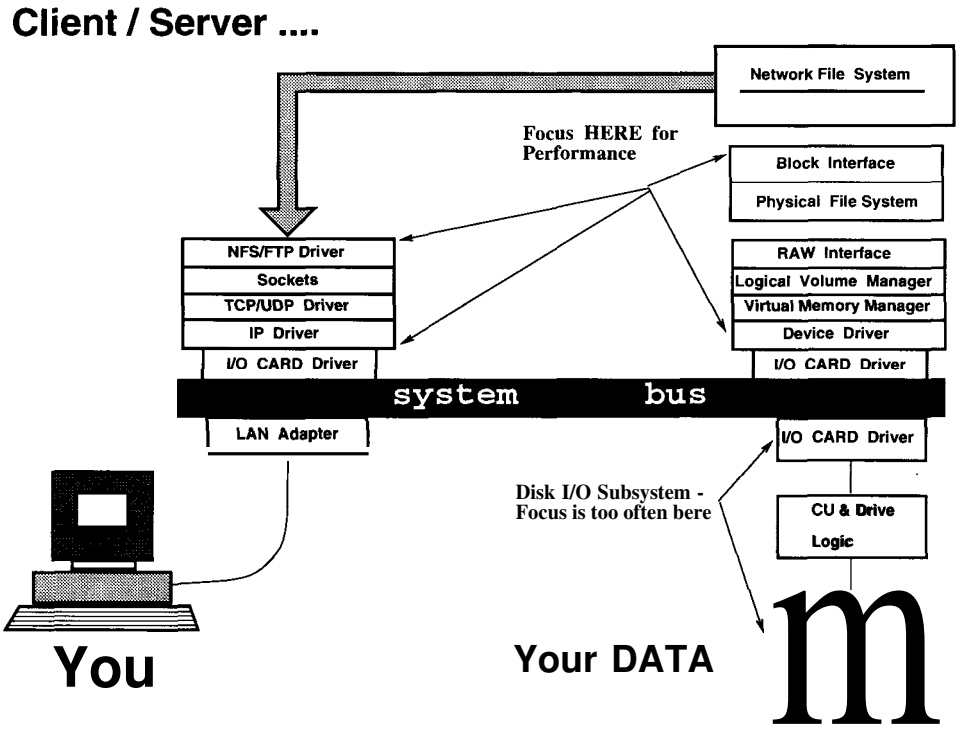


Figure 1: Disk and Network Protocol Stacks

Manager', another component from the SHIFT package), we had long down times during reboot due to 'fsck'. Fortunately, since CPU technology is moving rapidly to 64-bit architecture, giving room for much larger data files, Operating Systems and File Systems must follow. With IRIX 5.3 SGI introduced XFS, a true 64-bit file system, which not only allows for terabyte-size files, but also uses journalling to overcome the file system check problem. Our current experience proves that XFS is already pretty mature (though quotas are missing today, but is announced for 1/96; and there is a lack of intermediate problem recovery mechanisms), and whenever the filesystem becomes corrupt we get a 500 GByte Server back to work in less than 10 minutes (assuming it crashed and reboots without having other problems). Another important point is performance. Due to a large number of directly attached ufs/efs-filesystems, handling of the huge amount of blocks results in long block lists, and hence, the system behaved sometimes very sticky. With xfs these lists don't exist any longer. Therefore, the XFS-Filesystem performs much better, and as a

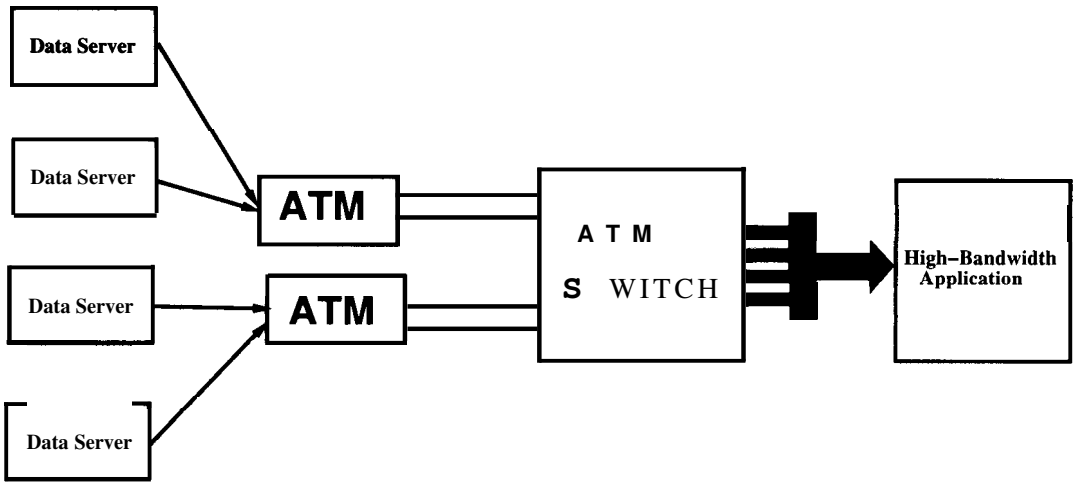


Figure 2: Distributed Data Servers using ATM Networks

consequence, saves lots ofCPU resources.

3.2 Backplane

For a number of reasons, including efficiency and scalability, one should consider dedicated servers, reserved to run batch jobs or deliver data to them. This is not new at all, since it was proposed by CERN in the SHIFT project long time ago. However, the critical point in the past was always the Backplane, interconnecting CPU- and Data Servers. We discovered very early that standard network technology, like FDDI, wasn't sufficient (small packets = large protocol processing overhead, limited bandwidth of < 10 MBytes/s) and that only special purpose hardware, like Ultranet, came close to what we needed. Though we would have been in bad shape without Ultranet, we were facing a lot of compatibility and reliability issues, unavoidable for a small company dealing with 5 to 10 computer vendor's un-

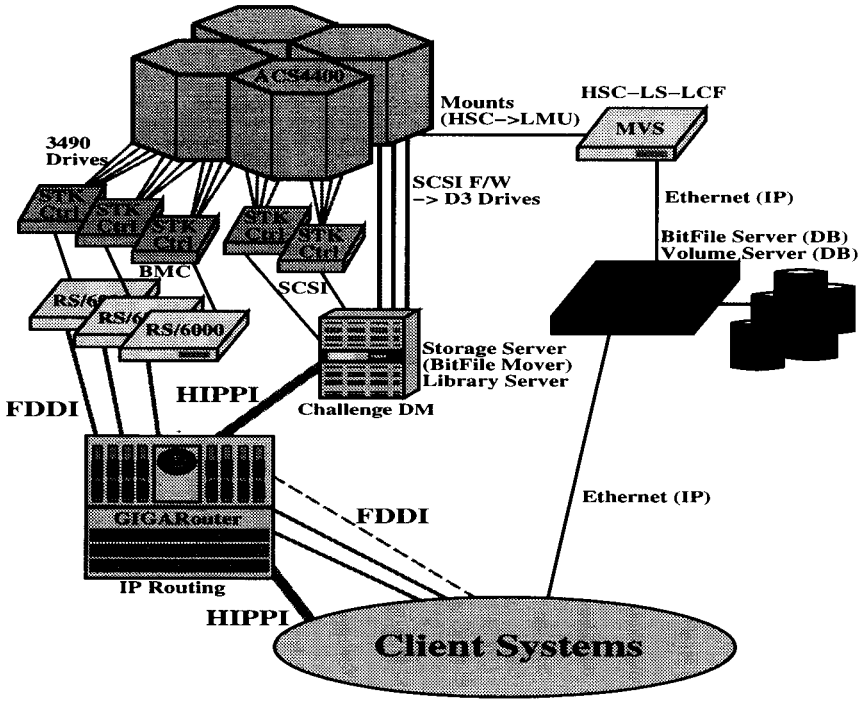
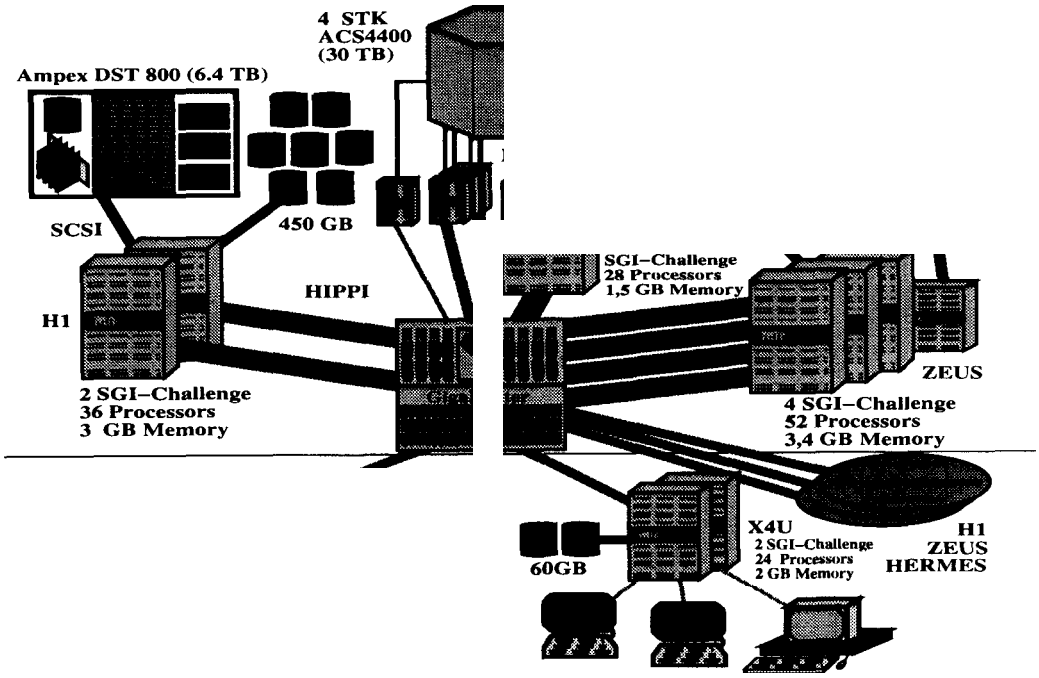


Figure 3: OSM based Storage Architecture at DESY

derwear (1/0 subsystems). Also, the internal limits of Ultranet (< 32 concurrent streams) forced us to look for alternatives, preferably standard network technology supported by the vendors envolved in the computer and data storage business. Just to mention their names, there is HIPPI (parallel 32/64 bit channel architecture, 100/200 MByte/s, a *very* pragmatic approach, believed to have a limited lifetime), ATM and FCS providing for a variety of speeds (ranging from 155 megabits/s to gigabits/s). Common to them is they were or are going to be approved by standard committees, like ANSI. Not all of them are available on any of our favourite computing platforms today, however, especially true for ATM, the number of vendor supported interfaces, including drivers, high level APIs and 1P is growing daily.


4 STK
ACS4400
(30 TB)
Ampex DST 800 (6.4 TB)
SCSI
450 GB
HIPPI
H1
2 SGI–Challenge
36 Processors
3 GB Memory
SGI–Challenge
28 Processors
1,5 GB Memory
ZEUS
4 SGI–Challenge
52 Processors
3,4 GB Memory
X4U
2 SGI–Challenge
24 Processors
2 GB Memory
60GB
H1
ZEUS
HERMES