Large Scale Cluster Computing Workshop

Fermilab, IL, May 22nd to 25th, 2001

Proceedings

1.0 Introduction

Recent revolutions in computer hardware and software technologies have paved the way for the large-scale deployment of clusters of off-the-shelf commodity computers to address problems that were previously the domain of tightly-coupled SMP[1] computers. As the data and processing needs of doing physics research increases while budgets remain stable or decrease and staffing levels only incrementally increase, there is a fiscal and computational need that must be and that can probably only be met by large scale clusters of commodity hardware with Open Source or lab-developed software. Near-term projects within high-energy physics and other computing communities will deploy clusters of some thousands of processors serving hundreds or even thousands of independent users. This will expand the reach in both dimensions by an order of magnitude from the current, successful production facilities.

A Large-Scale Cluster Computing Workshop was held at the Fermi National Accelerator Laboratory (Fermilab, or FNAL), Batavia, Illinois in May 2001 to examine these issues. The goals of this workshop were:

1. To determine from practical experience what tools exist that can scale up to the cluster sizes foreseen for the next generation of HENP[2] experiments (several thousand nodes) and by implication to identify areas where some investment of money or effort is likely to be needed;
2. To compare and record experiences gained with such tools;
3. To produce a practical guide to all stages of designing, planning, installing, building and operating a large computing cluster in HENP;
4. To identify and connect groups with similar interest within HENP and the larger clustering community.

Computing experts with responsibility and/or experience of such large clusters were invited, a criterion for invitation being experience with clusters of at least 100-200 nodes. The clusters of interest were those equipping centres of the sizes of Tier 0 (thousands of nodes) for CERN's LHC project[3] or Tier 1 (at least 200-1000 nodes), as described in the MONARC (Models of Networked Analysis at Regional Centres for LHC) project at http://monarc.web.cern.ch/MONARC/. The attendees came not only from various particle physics sites worldwide but also from other branches of science, including biophysics and various Grid projects, as well as from industry.

The attendees shared freely their experiences and ideas and proceedings are being currently edited, from material collected by the convenors and offered by the attendees. In addition, the convenors, again with the help of material offered by the attendees, are in the process of producing a "Guide to Building and Operating a Large Cluster". This is intended to describe all phases in the life of a cluster and the tools used or planned to be used. This guide should then be publicised (made available on the web, presented at

[1] Symmetric Multi-Processor

[2] High-energy and Nuclear Physics

[3] LHC, the Large Hadron Collider, is a project now in construction at CERN in Geneva. Its main characteristics in computing terms are described below, as are the meanings of Tier 0 and Tier 1. It is expected to come online during 2006.

appropriate meetings and conferences) and regularly kept up to date as more experience is gained. It is planned to hold a similar workshop in 18-24 months to update the guide.

All the material for the workshop is available at the following web site:
http://conferences.fnal.gov/lccws/
In particular, we shall publish at this site various summaries including a full conference summary with links to relevant web sites, a summary paper to be presented to the CHEP conference in September and the eventual Guide to Building and Operating a Large Cluster referred to above.

2.0 Opening Session (Chair, Dane Skow, FNAL)

The meeting was opened by the co-convenors – Alan Silverman from CERN in Geneva and Dane Skow from Fermilab. They explained briefly the original idea behind the meeting (from the HEPiX[4] Large Cluster Special Interest Group) and the goals of the meeting, as described above.

2.1 Welcome and Fermilab Computing

The meeting proper began with an overview of the challenge facing high-energy physics. Matthias Kasemann, head of the Computing Division at Fermilab described the laboratory's current and near-term scientific programme covering a myriad of experiments, not only at the Fermilab site but world-wide, including participation in CERN's future LHC programme notably in the CMS experiment, in NuMI/MINOS, MiniBoone and the Pierre Auger Cosmic Ray Observatory in Argentina. He described Fermilab's current and future computing needs for its Tevatron Collider Run II experiments, pointing out where clusters, or computing 'farms' as they are sometimes known, are used already.

He laid out the challenges of conducting meaningful and productive computing within worldwide collaborations when computing resources are widely spread and software development and physics analysis must be performed across great distances. He noted that the overwhelming importance of data in current and future generations of high-energy physics experiments had prompted the interest in Data Grids. He posed some questions for the workshop to consider over the coming 3 days:

- Should or could a cluster emulate a mainframe?
- How much could particle physics computer models be adjusted to make most efficient use of clusters?
- Where do clusters not make sense?
- What is the real total cost of ownership of clusters?
- Could we harness the unused CPU power of desktops?
- How to use clusters for high I/O applications?
- How to design clusters for high availability?

2.2 LHC Scale Computing

Wolfgang von Rueden, head of the Physics Data Processing group in CERN's Information Technology Division, presented the LHC experiments' computing needs. He described CERN's role in the project, displayed the relative event sizes and data rates expected from Fermilab RUN II and LHC experiments, and presented a table of their main characteristics, pointing out in particular the huge increases in data expected at LHC and consequently the huge increases in computing power that must be installed and operated for the LHC experiments.

The other problem posed by modern experiments is their geographical spread, with collaborators throughout the world requiring access to data and to computer power. He noted that typical particle

[4] HEPiX is a group of UNIX users in the High-energy Physics community who meet regularly to share experiences and occasionally undertake specific projects.

physics computing is more appropriately characterised as High Throughput Computing as opposed to High Performance Computing.

The need to exploit national resources and to reduce the dependence on links to CERN has produced the (MONARC) multi-layered model. This is based on a large central site to collect and store raw data (Tier 0 at CERN) and multiple tiers (for example National Computing Centres, Tier 1[5], down to individual user's desks, Tier 4) each with data extracts and/or data copies and each one performing different stages of physics analysis.

Von Rueden showed where Grid Computing would be applied. He ended by expressing the hope that the workshop could provide answers to a number of topical problem questions such as cluster scaling and making efficient use of resources, and some good ideas to make progress in the domain of the management of large clusters

2.3 IEEE Task Force on Cluster Computing

Bill Gropp of Argonne National Laboratory (ANL) presented the IEEE Task Force for Cluster Computing. This group was established in 1999 to create an international focal point in the areas of design, analysis and development of cluster-related activities. It aims to set up technical standards in the development of cluster systems and their applications, sponsor appropriate meetings (see web site for the upcoming events) and publish a bi-annual newsletter on clustering. Given that an IEEE Task Force's life is usually limited to 2-3 years, the group will submit an application shortly to the IEEE to be upgraded to a full Technical Committee. For those interested in its activities, the Task Force has established 3 mailing lists – see overheads. One of the most visible deliverables thus far by the Task Force is a White Paper covering many aspects of clustering.

2.4 Scalable Clusters

Bill Gropp from Argonne described some issues facing cluster builders. The http://www.top500.org/ web site list of the 500 largest computers in the world includes 26 clusters with more than 128 nodes and 8 with more than 500 nodes. Most of these run Linux. Since these are devoted to production applications, where do system administrators test their changes? For low values of N, one can usually assume that if a procedure or tool works for N nodes then it will work for N+1. But this may no longer stay true as N rises to large values. A developer needs access to large-scale clusters for realistic tests, which often conflicts with running production services.

How to define scalable? One possible definition is that operations on a cluster must complete "fast enough" (for example within 0.5 to 3 seconds for an interactive operation) and operations must be reliable. Another issue is the selection of tools – how to choose from a vast array, public domain and commercial?

One solution is to adopt the UNIX philosophy, build from small blocks. This is what the Scalable UNIX Tools project in Argonne is based on – basically parallel versions of the most common UNIX tools such as ps, cp, and ls and so on layered on top of MPI[6] with the addition of a few new utilities to fill out some gaps. An example is the ptcp command which was used to copy a 10MB file to 240 nodes in 14 seconds. As a caveat, it was noted that this implementation relies on accessing trusted hosts behind a firewall but other implementations could be envisaged based on different trust mechanisms. There was a lengthy discussion on when a cluster could be considered as a single "system" (as in the Scyld project where parallel ps makes sense) or as separate systems where it may not.

[5] Examples of Tier 1 sites are Fermilab for the US part of CMS and BNL for the US part of ATLAS.

[6] Message Passing Interface

3.0 Usage Panel (Chair, Dane Skow, FNAL)

Delegates from a number of sites presented short reviews of their current configurations. Panellists had been invited to present a brief description of their cluster, its size, its architecture, its purpose; any special features and what decisions and considerations had been taken in its design, installation and operation.

3.1 RHIC Computing Facility (RCF) (Tom Yanuklis, BNL)

Most CPU power in BNL serving the RHIC experiments is Linux-based, including 338 2U high, rack-mounted dual or quad CPU Pentium Pro, II and Pentium III PCs with speeds ranging from 200 to 800 MHz for a total of more than 18K SPECint95 units. Memory size varies but the later models have increasingly more. Currently Redhat 6.1, kernel 2.2.18, is used. Operating System upgrades are usually performed via the network but sometimes initiated by a boot floppy which then points to a target image on a master node. Both OpenAFS[7] and NFS are used.

There are two logical farms – Central Reconstruction System (CRS) and Central Analysis System (CAS). CRS uses a locally designed software for resource distribution with an HPSS[8] interface to STK tape silos. It is used for batch only, no interactive use; it is consequently rather stable.

The CAS cluster permits user login including offsite access via gateways and ssh, the actual cluster nodes being closed to inbound traffic. Users can then submit jobs manually or via LSF for which several GUIs[9] are available. There are LSF queues per experiment and priority. LSF licence costs are an issue. Concurrent batch and interactive use creates more system instability than is seen on the CRS nodes.

BNL built their own web-based monitoring scheme to provide 24-hour coverage with links to e-mail and to a pager in case of alarms. They monitor major services (AFS, LSF[10], etc) as well as gathering performance statistics. BNL also makes use the VACM tool developed by VA Linux, especially for remote system administration.

They are transitioning to open SSH and the few systems upgraded so far have not displayed any problems. An important issue before deploying Kerberos on the farm nodes concerns token passing such that users do not have to authorise themselves twice. BNL uses CTS for its trouble-ticket scheme.

3.2 BaBar (Charles Young, SLAC)

At its inception, BaBar had tried to support a variety of platforms but had rapidly concentrated solely on Solaris although recently Linux has been added. They have found that having two platforms has advantages but more than two does not. BaBar operated multiple clusters at multiple sites around the world; this talk concentrates uniquely on the ones in SLAC where the experiment acquires its data. The reconstruction farm, a 200 CPU cluster, is quasi-real time with feedback to the data taking and so should operate round the clock while the experiment is operational. There is no general user access, it is a fully controlled environment. The application is customised for the experiment and very tightly coupled to running on a farm. This cluster is today 100% SUN/Solaris but will move to Linux at some future time because Linux-based dual CPU systems offer a much more cost-effective solution to the problems, running on fewer nodes and with lower infrastructure costs, especially network interfacing costs.

[7] AFS is a distributed file system developed by Transarc and now owned and marketed by IBM. OpenAFS is an open source derivative of AFS where IBM is a (non-active) participant.

[8] HPSS is hierarchical storage system software from IBM designed to manage and access 100s of terabytes to petabytes of data. It is used at a number of HEP and other sites represented at this workshop.

[9] Graphical User Interface.

[10] LSF is a workload management software suite from Platform Corp that lets organizations manage their computing workloads. Its batch queuing features are widely used in HEP and other sites represented at this workshop.

There is also a farm dedicated to running Monte Carlo simulations; it is also a controlled environment, no general users. This one has about 100 CPUs with a mixture of SUN/Solaris and PC/Linux. Software issues on this farm relate to the use of commercial packages such as LSF (for batch control), AFS for file access and Objectivity[11] for object-oriented database use. The application is described as "embarrassingly parallel".

Next there is a 200 CPU data analysis cluster. Data is stored on HPSS and accessed via Objectivity. This cluster offers general access by users who have widely varying access patterns with respect to the amount of CPU load and data accessed. This cluster is a mixture of Solaris and Linux and uses LSF.

Still at SLAC there is a smaller, 40 CPU, offline cluster with a mixture of SUN/Solaris and PC/Linux nodes.

One of the issues facing the experiment is the licensing cost of the commercial software, which is becoming a significant fraction compared to the hardware acquisition cost. LSF also is showing strain under scaling with many jobs to schedule requiring a complex priority algorithm.

Finally the speaker noted that BaBar computing model was already split into tiers, in their case Tiers A, B and C, where Tier A Centres are established already at SLAC, IN2P3 in Lyon and RAL in England. He ended by making reference to Grid computing and how it could possibly help BaBar in the future.

3.3 FNAL Offline Computing Farms (Steve Wolbers)

These are based on 314 dual-CPU Pentium PCs running Linux, with 124 more on order. The latest PCs are rack-mounted (the previous incarnations were in boxes). The farms are logically divided by experiment and are designed to run a small number of large jobs submitted by a small number of expert users. There are currently 314 PCs with 124 more on order. The CDF experiment has two I/O nodes (large SGI systems) as front ends to the farms and a tape system directly attached to one of these. Stephen anticipates that the data volume per experiment per year will approximately double every 2.4 years and this is taken into consideration during farm design. In the future disk farms may replace tapes and analysis farms may be on the horizon.

The primary goal of today's clusters is to return event reconstruction results quickly and provide cost-effective CPU power and not necessarily to achieve 100% CPU usage. The batch scheme in use is the locally developed FBSng. Rare among HEP labs, FNAL executes a 30-day burn-in on new PCs. Their expansion plans are to acquire more of the same since it appears to work and does not create a heavy support load.

3.4 Sanger Centre (UK) (James Cuff)

The Sanger Centre is a research centre funded primarily by The Wellcome Trust and performing high performance computing for the Human Genome Project. It was founded in 1993 and has approximately 570 staff members now. They have a total of some 1600 computing devices, nearly half of them Compaq Alpha systems but there are also 300 PC desktops as well as a few vector systems. The Alpha clusters are based on Compaq's FibreChannel Tru64 Clustering and use a single system image. The data volume for the human genome is 3000 MB. The central backbone is ATM.

The largest cluster is the Node Sequence Annotation Farm consisting of 8 racks of 40 Alpha systems; each Alpha has 1GB memory, plus a number of PCs for a total of 440 nodes, a configuration driven by the needs of the applications. There is also 19.2 Terabytes of spinning disk and the total system corresponds to 1000KW of power.

[11] Objectivity is an object oriented database product from the Objectivity Corporation.

The batch scheme in place is LSF. Such tightly coupled clusters offer good territory for NFS and the applications use sockets for fast inter-node data transfer. Unlike typical HEP applications, Sanger's applications are better suited to large memory systems with high degrees of SMP. However, in the longer term, they are considering moving towards wide-area clusters, looking perhaps towards Grid-type solutions.

3.5 **H1** (Ralf Gerhards, DESY)

For historical reasons, the H1 experiment at DESY operate a mixed cluster, originally based on SGI workstations but now also containing PCs running Linux. Today the SGI Origin 2001 operates as one of the disc servers, the others being PCs serving IDE discs based on a model developed by CERN. The farm, one of several similar farms for the various major experiments at DESY, is arranged as a hierarchical series of sub-farms; within the 10-20 node sub-farms the interconnect is Fast Ethernet and the sub-farms are connected together by Gigabit Ethernet. These sub-farms are assigned to different activities such as high-level triggering, Monte Carlo production and batch processing. The node allocation between these tasks is quite dynamic. The farm nodes, 50 dual-CPU nodes in total, are installed using the same procedures developed for DESY's desktops, thus offering a site-wide SuSE Linux environment for the H1 experiment users and permitting them to benefit from the general DESY Linux support services. Over and above this is an H1-developed framework based on CORBA[12] for event distribution. The batch system in use is PBS[13], integrated to AFS, with difficulty. DESY is also working on a Data Cache project in conjunction with other HEP labs; this project should offer transparent access to data whether on disc or on tape.

3.6 **Belle** (Atsushi Manabe, KEK)

Historically KEK experiments have been based on RISC systems (UNIX workstations) and movement to PCs has been slowed by in-place lease agreements. Nevertheless, Belle has 3 PC-based farms:

- 158 CPUs housed in mainly quad-CPU boxes for event reconstruction and using SUN servers for I/O. The nodes use home-built software to make them look like SMP systems. There are only 5 users accessing this cluster. Installation and cabling was done by in-house staff and physicists.
- The second cluster, with 88 CPUs, has dual Pentium III PCs; it was fully installed (hardware and software) by Compaq
- The third 60-node cluster again has quad-CPU systems. It was acquired under a government-funded 5 year lease agreement within which architecture upgrades will be possible

All the clusters use NFS for file access to the data. There are 14TB of disk (10TB of RAID and 4TB local) on 100BaseT network. The data servers consist of 40 SUN servers each with a DTF2 tape drive. The tape library has a 500TB capacity. The farm is used for PC farm I/O R&D, HPSS (Linux HPSS client API driver by IBM), SAN (with Fujitsu), and GRID activity for ATLAS in Japan. There is no batch system in place but the batch jobs are long running and so a manual scheme is thought sufficient.

3.7 **Lattice QCD Clusters** (Jim Simone, Fermilab)

This theoretical physics application needs sophisticated algorithms and advanced computing power, typically massively parallel computing capabilities where each processor can communicate easily with its neighbours and near-neighbours. In previous implementations, such systems were based on commercial or specially built supercomputers. The current project is to build a national infrastructure across the USA

[12] CORBA is the Common Object Request Broker Architecture developed by the Object Management Group (OMG). It is a vendor-independent scheme to permit applications to work together over networks.

[13] PBS is the Portable Batch System, developed at Veridian Systems for NASA but now available as open source.

with three 10 Tflop/s[14] facilities by 2005, two (Fermilab and Jefferson Lab) with commodity PCs and the third (BNL/Columbia) using custom-built chips. Moving from super computers to clusters has allowed the project to benefit from fast-changing technology, open source Linux and lots of easy-to-use tools.

The FNAL-based 80-node cluster was installed in Fall 2000 with the intention to ramp up by 300 more nodes per year. This is a true Beowulf cluster system whereas many of the farms described above are not Beowulf. This is because Lattice QCD requires tight communication between processes. Myrinet 2000 is used for networking as it is much more efficient than Ethernet for TCP traffic in terms of CPU overhead at high throughput.

The PCs' BIOS is PXE-enabled[15] for remote boot over Ethernet and it has been modified to permit access to the serial port for remote health checking and certain management tasks. The software installed includes PBS with the Maui scheduler for batch queuing and MPI for inter-node communication. The choice of which MPI architecture was made after some tests and mpich/vmi was chosen with the virtual driver layer provided by NCSA but there is a lingering doubt on the final choice with respect to its ability to scale.

3.8 Discussion

Given that current-day PC configurations usually include very large disc systems (typically 20-40GB), how to make use of this space? One suggestion was CacheFS which is a scheme using a local cache in synch with the original file system. There was a mixed reaction to this: SLAC had been using it indirectly[16] but has stopped using it but the Biomed team from the University of Minnesota reported good experiences. Another alternative is to use the local disc for read-only storage of input files for the application, copied automatically on demand or manually from a central data store, essentially creating a local data replica.

The next topic was how to use unused CPU cycles, especially on desktops (this would come up later in the meeting also). Although there are clearly solutions (see later, Condor, Seti@Home, etc) there is the question of resource competition and management overhead. Also, the fact that users can alter the local setup without warning can make this a too chaotic environment. However, there are many successful examples where Condor in particular is used including some US government agencies and, within the HEP world, INFN in Italy and a particular CERN experiment (CHORUS). Sanger also makes a little use of Condor. On the other hand, DESY had decided that it is cheaper to invest in a few more farm nodes than to dedicate one person to administer such a cycle-stealing scheme.

Talking of management overhead, it was noted that one reason to upgrade old CPUs for more recent faster chips is precisely to decrease such overhead. Management overhead typically increases with the number of boxes and a single 800MHz chip PC can approximately replace 3-4 200MHz systems in power.

A poll was taken on which sites used asset management packages. BNL has developed a database scheme for this as has H1 at DESY. The BNL scheme uses manual input of the initial data with subsequent auto-update with respect to a central database. CERN too has a database scheme based on the PCs MAC address[17] with facilities for node tracking and location finding (where a given node is located physically in the computer room). They are seriously considering the use of bar codes in the future.

[14] 1 Tflop/s is a Tera (1000 million) Floating Point operations per second

[15] PXE is Preboot Executive Environment, a method to boot a mini-kernel

[16] They had been using SUN's autoclient, which uses cacheFS but had stopped because of bottlenecks and the risks of a single point of failure for the cluster.

[17] The MAC address (short for Media Access Control address) is the hardware address that uniquely identifies each node of a network.

How should asset data for new systems be collected when they start arriving in the sort of bulk numbers expected in the future – for example 100+ nodes per week when we consider clusters of 5000 nodes. Inputting asset data for this number of PCs by hand does not scale. We really want something more automatic – something in the firmware for example. It was noted that IBM has some such a scheme with embedded data, which can be read out with a hand-held scanner.

Turning to system administration tools, it was remarked that no site represented had mentioned using cfengine, a public domain system admin tool commonly found in the wider UNIX community. It was questioned if there is experience on using it on clusters of up to 1000 nodes and beyond.

Lastly, the question of file systems in use was brought up. There is heavy use of NFS and AFS at various sites but much less of so-called global file systems (GFS for example). Inside HEP at least, AFS appears to have a dominant position although there is a clear move towards OpenAFS.

4.0 Large Site Session (Session Chair, Steve Wolberg, FNAL)

4.1 CERN Clusters (Tim Smith)

Until recently, CERN's central computer center was comprised of a large number (37) of small to medium sized clusters in a variety of architectures, each dedicated to a given experiment. This clearly does not scale and there is currently a migration to a small number (three) of much larger clusters made up from two architectures (Linux and Solaris) where the clusters are distinguished by usage rather than by experiment and the different experiments share the clusters. This also involves reducing the variety of hardware configurations in place. The three clusters will be heterogeneous which it is accepted will add extra complications in configuration and system administration and possibly means that system administration tools will have to be largely homegrown.

There is a 50-node dual-CPU Interactive Cluster: user access is balanced by using DNS[18] where the user connects to a generic DNS name and an algorithm discovers the least loaded node to which the user is then connected. The load algorithm considers not only CPU load but also other factors such as memory occupation and others.

There is a 190 node dual-CPU scheduled batch cluster which is dedicated at any given time to one or a small number of experiments – for example for dedicated beam running time; or to defined data challenge periods. Dedicating all or some nodes of this cluster to given experiments is done simply by declaring only experiment-specific LSF queues on these nodes. Then rearranging the split between experiments or replacing one experiment by another is simply a matter of redefining the LSF queue assignment, although it is accepted this is a rather mechanical process today.

Lastly, there is a so-called 280-node dual-CPU "chaotic" batch cluster, which offers general and unscheduled time.

There is also a tape and disc server cluster, which does not offer direct user access.

Typically the user connects to the Interactive Cluster and launches an LSF job to one of the two batch clusters. There is virtually no user login permitted on the batch clusters but exception can be made for debugging if it appears that there is no alternative to fixing some particular problem. For this purpose LSRUN (from the LSF suite) is run on a few batch nodes in each cluster.

Among the tools in use are

[18] DNS is the Domain Name Scheme used to translate a logical network name to a physical network address.

- System installation – Kickstart[19] for Linux nodes and Jumpstart[20] for Solaris
- Automation of installation – ANIS (CERN developed)
- Post-installation and system configuration – SUE (CERN developed)
- Application installation – ASIS (CERN developed). But the speaker noted that both SUE and ASIS were originally written to ease system admin tasks when there were a wide range of UNIX flavours to support. With this decreasing to only 2 platforms over time, it is perhaps an opportune moment to reconsider both of these tools. Can they be simplified? Also, for batch nodes, only a very few applications from the ASIS suite are needed.
- Console management – PCM (DEC's Polycenter Console Manager[21]) and a console concentrator have been used up to now but they are moving to cross-wiring of the serial ports to better cope with the rapidly-rising number of nodes, although cross-wiring such a large number of nodes is a headache to manage. For the future they are investigating public domain tools such as VACM and Etherlite.
- Power management – none
- Monitoring – currently using a CERN-built alarm scheme known as SURE plus simple home-written tools for performance monitoring. Currently working on a larger project to build a performance and alarm scheme, which will monitor services rather than objects within the servers (the so-called PEM project, described later in the meeting).

4.2 **VA Linux** (Steve DuChene)

VA Linux install and support clusters up to quite large number of nodes including installations for two sites represented at the meeting – BNL and SLAC. They have noticed a marked trend in increasing CPU power per floor space by moving from 4U to 2U to 1U systems and they expect this to continue – several PCs in a 1U high unit shortly. But sites should be aware that such dense racking leads inevitably to greater challenges in power distribution and heat dissipation.

For each configuration, a cluster must have a configuration and management structure. For console management for example, VA Linux recommends the VACM tool, which they developed and which is now in the public domain. In VACM configurations, there is a controller PC in each rack of up to 64 nodes on which VACM runs; from there VACM then access the BIOS directly or a special (from Xircom) board connected to the motherboard to get its monitor data. VACM also supports remote power cycling and BIOS serial console port redirection. It can access sensors on the motherboard – fan speeds, temperature, etc. Further, the code is not x86-specific so, being open source, is portable to other architectures and there is an API[22] to add your own modules. The source can be found on sourceforge.org.

Another tool which they wrote and have made available is SystemImager: the system administrator configures a master image on a typical node, stores the image on a server and loads that to client nodes via a network bootstrap or on demand from a floppy boot. Obviously this offers most advantages on a homogeneous cluster. Both the initial load of the clients and their subsequent consistency depend on the standard UNIX rsynch protocol with the originally configured node as the master image. It was noted however that at least the current version of rsynch suffers from scaling problems. In the current scheme, it is recommended to base no more than 20 to 30 nodes on a single master but larger configurations can be arranged in a hierarchical manner – image a set of masters from a super-master and then cascade the image down to the end nodes. Other effects of scaling could be offset by using faster cluster interconnects.

[19] Kickstart is a tool from Redhat which lets you automate most/all of a RedHat Linux installation
[20] JumpStart is Sun's method of providing a turnkey, hands-off solution to installing Solaris
[21] DEC Polycenter replaces the PC console terminal or monitor. It connects to the serial port of clients via terminal servers and collects the console output from the RS232 serial lines. The host system provides thus a central point for monitoring the console activity of multiple systems, as well as for connecting directly to those systems to perform management tasks.
[22] API – Application Programming Interface – a defined method to write a new module and interface it to the tool.

The next version of this tool should offer push updates and it may eventually use the multicast protocol for yet faster on-demand updates by performing the updates in parallel. Once again the source can be found on sourceforge.

Citing perhaps an extreme case of redundancy, one VA Linux customer has a policy of purchasing an extra 10% systems. Their scheme for reacting to a system problem is first to reboot; if that fails to re-install; and if that fails to replace the node and discuss offline with the supplier for repair or replacement of the failed node.

4.3 SLAC Computer Centre (Chuck Boeheim)

There is a single large physical cluster although viewed from a user point of view, there are multiple logical ones. This is achieved by the use of LSF queues. The hardware consists of some 900 single-CPU SUN workstations running Solaris and 512 dual CPU PCs running Linux (of which the second 256 nodes were about to be installed at the time of the workshop). There are also dedicated servers for AFS (7 nodes, 3TB of disc space), NFS (21 nodes, 16TB of data), Objectivity (94 nodes, 52TB of data) plus LSF and HPSS (10 tape movers and 40 tape drives). Objectivity manages the disk pool and HPSS manages the tape data. Finally there are 26 systems (a mixture of large and small Solaris servers and Linux boxes) dedicated to offer an interactive service. All these are interconnected via Gigabit Ethernet to the servers and 100Mbit Ethernet to the farm nodes, all linked by 9 CISCO 6509 switches.

Tha major customer these days at SLAC is the BaBar experiment. For this experiment there is a dedicated 12 node (mixed Solaris and Linux) build farm. The BaBar software consists of 7M SLOCS[23] of C++ and the builds, which take up to 24 hours, are scheduled by LSF although privileged core developers are permitted interactive access.

The batch nodes do not accept login access except for a few designated developers requiring to debug programs. NFS is used with automounter on all 1400 batch nodes, controlled by the use of net groups. This has been occasionally been plagued by mount storms and needs to be carefully monitored although there seems to be fewer problems using the TCP implementation of NFS as opposed to the more common UDP one. [It was noted later in the discussion that a similar-sized CERN cluster downplays the use of NFS, preferring to adopt a staging scheme based on the RFIO protocol.]

The Centre at SLAC is staffed by some 18 persons some of whom also have a role in desktop support for the Lab. The ratio of systems supported by staff members has gradually risen: in 1998 they estimated 15 systems per staff person; today it appears to be closer to 100 systems per person. One possible reason for this improvement is the reduction in the variety of supported platforms and a reduction in complexity.

Asked to explain the move from SUN to PC, the speaker explained that maximising the use of floor space was an important aspect: PCs can be obtained in 1U boxes, which SUN cannot supply today. As elsewhere, limited floor space is an issue but one aspect that may be relatively unique to SLAC, or perhaps in California generally, is the risk of seismic activity – physical rack stability is important!

In their PC systems. SLAC has enabled remote power management and remote management, the combination of which permits a fully lights-out operation of the centre. They use console servers with up to 500 serial lines per server. As regards burn-in testing, their users never permit them enough time for such "luxuries"! Further, they have noticed that when systems are returned to a vendor under warranty, sometimes a different configuration is returned! Like CERN and other sites, with so many nodes, physical tracking of nodes is an issue, a database is required with bar codes on the nodes of the clusters.

[23] SLOC – Significant (non-comment) Lines Of Code

Among the tools in use are

- Network installation – using locally-developed scripts wrapped around Kickstart and Jumpstart they have managed to install some 256 nodes in an hour
- Patch management – a local tool
- Power management – using tools from VA Linux (e.g. VACM) they can power up or down a cluster taking account of inter-node sequence dependencies
- Monitoring – the Ranger tool developed by C.Boeheim
- Reporting – a local tool which gathers reports across the cluster and produces short summaries (who wants to read 512 times the same error?)

For development purposes, the support team have established a small test bed where they can test scaling effects.

The speaker closed with the memorable quote that “a cluster is an excellent error amplifier”.

5.0 Hardware Panel

This panel was led by Lisa Giachetti of Fermilab. The panel and audience were asked to address the following questions:

- From among the criteria are used to select hardware - price, price performance, compatibility with another site, in-house expertise, future evolution of the architecture, network interconnect, etc. - which are the 3 most important in order of significance?
- Do you perform your own benchmarking of equipment?
- How do you handle life cycles of the hardware, for example, the evolution of Pentium processors where later configurations and generations may need a new system image?
- Have you experience, positive or negative, with heterogeneous clusters?

5.1 BNL (Tom Yanuklis)

Typically BNL consider that PCs have a 3-year lifecycle. In this respect, it is important to understand for how long a vendor will support a particular configuration and what effect future vendor changes might have on the ongoing operation of your farm. Their primary purchase criteria are price performance, manageability and compatibility with their installation image. Like many labs, BNL do not perform rigorous benchmarking of proposed configurations. But they do negotiate with vendors to obtain loaned systems that they then offer to end-users for evaluation with the target applications. They have noted that with increasing experience, users can better specify the most suitable configuration (memory, I/O needs, etc) for their application.

BNL prefer to install homogeneous clusters and declare each node either dedicated to batch or to interactive logins, although they reserve the right to modify the relative numbers of each within the cluster. As mentioned earlier by others, they have seen heat and power effects caused by ever-denser racking and they have had to install extra power supplies and supplementary cooling as their clusters have been expanded.

Over time, as the current experiments got underway and they built up processing momentum, they were very glad to have had the flexibility to change their processing model: instead of remotely accessing all the data, they were able to move to a model with local caching on the large system discs which are delivered in today’s standard configurations.

Once installed and running, getting changes to a cluster approved by the various interested groups can be an issue. This brings in the question of who proposes such changes – users or administrators – and what consensus must be reached among all parties before implementation of significant changes.

BNL certainly consider the use of installation tools such as Kickstart and SystemImager very important. Also remote power management and remote console operation are absolute essentials for example VACM and IBM's latest Cable Chain System which uses PCI cards with onboard Ethernet allowing commands to be issued to the cards over a private sub-network.

5.2 PDSF at LBNL (Thomas Davis)

The NERSC (National Energy Research Scientific Computing Center) at the Lawrence Berkeley National Laboratory has a long-standing tradition of operating supercomputers and the next iteration of this is likely to be a PC cluster so this is where current research is concentrated. PDSF (Parallel Distributed Systems Facility) originated at the US super-collider (SSC) which was planned to be built in Texas but was abandoned around 1995. It originally consisted of a mixture of SUN and HP workstations but recent incarnations are based on PCs running Linux.

Users, which include a number of major HEP experiments, purchase time on the facility with real funds and customers include many HEP experiments. But despite buying time on the cluster, clients do not own the systems, but rather rent computing resources (CPU time and disc space). They may actually get more resources than they paid for if the cluster is idle. LSF Fairshare is used to arbitrate resources between users. Customers often want to specify their preferred software environment, down to Linux versions and even compiler versions so one of PDSF's most notable problems is software configuration change control, not surprising faced with a large variety of user groups on the cluster.

PC hardware is purchased without ongoing vendor support, relying on the vendor's standard warranty: when the warranty period ends, the PCs are used until they die naturally or until they are 3 years old. Another reason given to replace PCs is the need for memory expansion. In general, LBNL purchase systems as late as possible before they are needed in order to benefit from the ever-rising best price/performance of the market, although they tend to purchase systems on the knee of the curve rather than the newest, fastest chip. For example, when the fast chip is 1000 MHz, they will buy 2 650MHz chip systems for a similar price.[24] Their memory configurations are always comprised of the largest DIMMs available at the moment of purchase.

They have noticed that disc space per dollar appears to be increasing faster than Moore's Law[25]; they estimate a doubling every 9 months. [This estimate is supported by CERN's PASTA review, which was performed to estimate the evolution of computing systems in the timescale of the LHC (until 2005/6).]

5.3 FNAL

FNAL have developed a qualification scheme for selecting PC/Linux vendors. They select a number of candidate suppliers (up to 18 in their first qualification cycle) who must submit a sample farm node and/or a sample desktop. FNAL provide the required Linux system image on a CD. FNAL then subject the samples to various hardware and software tests and benchmarking. They also test vendor support for the chosen hardware and for integration although no direct software support is requested from vendors.

The selection cycle is repeated approximately every 18-24 months or when there is a major technology change. After the last series, Fermilab selected 6 agreed suppliers of desktops and 5 possible suppliers of servers from which they expect to purchase systems for the coming year or so at least.

5.4 Discussion

[24] Another tip described is to be aware of the vendor's business cycle; the possibility of obtaining a good deal increases in the final days of a quarter.

[25] Moore's Law states that the number of transistors on a chip doubles every 18-24 months.

Upgrades: at most sites, hardware upgrades are rather uncommon; does this follow from the relative short life of today's PC configurations? PDSF have performed a major in-place disc upgrade once and they also went through an exercise of adding extra memory. BNL have found it sometimes necessary also to add extra memory, sometimes as soon as 6 months after initial installation; this had been provoked by a change of programming environment. BNL had negotiated an agreement with their suppliers whereby BNL installed the extra memory themselves without invalidating the vendor's warranty. It was noted that it might not always be possible to buy compatible memory for older systems.

Benchmarking: it was generally agreed that the best benchmark is the target application code. One suggestion for a more general benchmark is to join the SPEC organisation as an Associate Member and then acquire the source codes for the SPECmark tests at a relatively inexpensive price. Jefferson Lab use a test based on prime numbers (see for example, Mprime), which exercises the CPU. VA Linux has a tool (CTCS), which tests extensively the various parts of the CPU.

Acceptance Tests: it appears that these are relatively untypical among participating sites although Fermilab performs burn-in tests using the same codes as for the selection evaluations. They also use Seti@Home, which has the advantage to fit on a single floppy disc. NERSC perform some tests at the vendor site to validate complete systems before shipment.

On a related issue, when dealing with mixed vendor configurations (hardware from a given supplier and software from another) it is important to define clearly the responsibilities and the boundaries of support.

Vendor relations: it was suggested to channel all contacts with a vendor through only a few named and competent individuals on each side. They serve as filters of problems (in both directions) and they can translate the incoming problems into a description that each side can understand. Even with this mechanism in place, it is important to establish a relationship at the correct level – technical as well as strategic depending on the information to be exchanged. Until long-standing relationships can be built, it can be hard to reach a correct level within the supplier organisation (deep technical expertise for technical problems, senior managerial for strategic).

6.0 Panel A1 - Cluster Design, Configuration Management

The questions posed to this panel, chaired by Thomas Davis, were ----

- Do you use modeling tools to design the cluster?
- Do you use a formal database for configuration management?

6.1 Chiba City (John-Paul Navarro, Argonne)

Chiba City has been built up to 314 nodes since 1999 to be used as a test bed for scalability tests for High Performance Computing and Computer Science studies. The eventual goal is to have a multi-thousand node cluster. The configurations are largely Pentium III systems linked by 64-bit Myrinet[26]. One golden rule is the use of Open Source software and there is only a single commercial product in use.

The cluster is split logically into "towns" of up to 32 nodes each with a "mayor" managing each town. The mayors themselves are controlled by a "city mayor". Installation, control and management can be considered in a hierarchical manner. The actual configurations are stored in a configuration database. Sanity checks are performed at boot time and then daily to monitor that the nodes run the correct target environment. Mayors have direct access to the consoles of all nodes in their towns. Remote power management is also used and both this and the remote management are considered essential.

[26] Myrinet is a high-performance, packet-communication and switching technology from Myricom Inc. that is widely used to interconnect clusters.

The operating system is currently Redhat 6.2 with the 2.4.2 Linux kernel but all in-house software is developed so as to be independent of the version of Linux. The programming model is MPI-based and job management is via the PBS resource manager and the Maui scheduler.

The initial installations and configurations were performed by in-house staff but if they have the choice, they will not repeat this! Myrinet was difficult to configure and the network must be carefully planned; it was found to be sensitive to heavy load situations. Since the original installation they have had to replace the memory and upgrade the BIOS, both of which were described as a "pain". The environment is overall stressful: for example they suffer power fluctuations and they make the point that a problem, which occurs on 6 nodes, scales to a disaster when it occurs on 600!

They have built their own management tools (which they have put into the public domain) and they make use of rsh in system management tasks but find that it does not scale well so work is in progress to get round rsh's maximum 256 node restriction. NFS is used but it does not scale well in conditions of heavy use, especially for parallel applications so a parallel file system, PVFS is under investigation.

They are developing an MPI-based multi-purpose daemon (MPD) as an experiment in job management (job launching, signal propagation, etc). Work is also going on with the Scyld Beowulf system – use of a single system image on a cluster and emulation of a single process space across the entire cluster. The Scyld tests at Chiba City are on scalability and the use of Myrinet. Currently this is limited to 64 nodes but tests are being carried out using 128 nodes.

In the course of their work they have produced two toolkits, one for general systems admin and the second specifically for clusters. Both are available from their web site at http://www.mcs.anl.gov/systems/software.

Finally some lessons learned include

- Use of remote power and console is essential
- Many UNIX tools don't scale well – how would you build, operate and program a million node cluster? Could you?
- A configuration database is essential
- Change management is hard
- Random failures scale into nightmares on a cluster

6.2 PDSF and the Alvarez Clusters (Shane Cannon, LBNL)

NERSC at LBNL[27] has a number of very large clusters installed. For example they have an IBM SP cluster with more than 2000 nodes, a configuration which is rated fifth in the Top 500 Clusters list. There is also a 692 node Cray T3E. In general the applications in most use at NERSC are embarrassingly parallel and this is reflected in the cluster design. For a new configuration, rather than perform detailed modelling, they find that they can get the best value for money by simply using the principles of buying commodity processors and configurations.

The PDSF and Alvarez clusters are planned to offload and perhaps eventually the IBM and Cray systems and are directly targeted at parallel applications. In the Alvarez cluster, a high-speed network was specified in the Request For Prices (RFP) and Myrinet was selected.

Among the issues they face are maintaining consistency across a cluster and the scalability not only of the cluster configurations but also of the human resources needed to manage these.

[27] See Section 5.2 above for an explanation of these sites.

6.3 Linux Networx (Joshua Harr)

For installing new systems, Linux NetworX makes use of many tools including SystemImager (good for homogeneous clusters) and LUI (better for heterogeneous clusters). They find that neither of these meets all the needs today but the OSCAR project inside IBM is reputed to be planning to merge the best features. They have found however, as mentioned by others, that NFS and rsync by themselves do not scale.

They use LinuxBIOS – a Linux micro-kernel that can control the boot process but they agree with previous speakers that a remotely accessible BIOS is really desirable and ought be supplied by all vendors and properly supported on the PC motherboard.

6.4 Discussion

From a poll of the audience, in-house formal design processes in cluster planning are at best rare. More common is to use previous experience, personal or from colleagues or conference presentations. On the other hand, cluster vendors do indeed perform configuration planning, especially with respect to power distribution, cooling requirements and floor space.

The use of Uninterruptible Power Supplies (UPS) varies. NERSC does not have a global UPS, the main argument given that a UPS for a Cray would be prohibitively expensive; they do however protect key servers. SLAC has a UPS for the building, justified by comparing the cost with the clock time, which would be needed to restart an entire building full of computer systems[28]. Likewise, CERN's main computer centre is protected. FNAL at the current time has UPS on certain racks with vital servers, for example AFS, but it is currently considering adding a UPS for the whole centre. They have to perform two complete power cycles per year for safety checks and they find it sometimes takes weeks for stable service to be resumed!

Consistency of the configurations: it was agreed that hardware consistency can only be obtained by purchasing an entire cluster as a single unit. Software consistency can be obtained by several methods, described more fully elsewhere in this meeting. For example by cloning a single system image; by using a form of synchronisation against a master system or by using a consistent set of RPMs[29]. In this respect Redhat are reputed to be working on a tool based on the use of RPMs, which should take care of inter-RPM dependencies. Debian, another Linux distribution, has a similar tool.

Monitoring tools: described in detail in Panel A3 (section 10) but it was noted here that as clusters increase in size, it is important to set correct thresholds. In a thousand node cluster, who if anyone should be alerted at 3 o'clock in the morning that 5 nodes have crashed?

Should one buy 1000 systems or a cluster of 1000 nodes? The former may be cheaper because the second is a "solution" and often costs more. But don't forget the initial and ongoing in-house support costs. On the other hand, if only the vendor-supplied hardware will be used and not the software (as reported elsewhere), is it worth the extra expense to buy the "solution"?

Tools – develop or re-use? A number of sites have built various installation utilities which will install a new PC with the system administrator being asked only a very few questions (see Panel A2, section 7). In general there is a trade-off of adaptability versus flexibility when deciding to use or adapt an existing tool

[28] SLAC has estimated that recovering from a power down takes up to 8 to 10 hours with several people helping.

[29] RPM is the Red Hat Package Manager. While it does contain Red Hat in the name, it is completely intended to be an open packaging system available for anyone to use. It allows users to take source code for new software and package it into source and binary form such that binaries can be installed and tracked and source can be rebuilt. It also maintains a database of all packages and their files that can be used for verifying packages and querying for information about files and/or packages.

or producing one's own. There is a general desire at some level not to reinvent the wheel but it is often accepted to be more intellectually challenging to produce one's own.

Configuration Management: apart from the tools already mentioned, one should add the possibility to use cfengine. PDSF has begun to investigate this public domain tool, which is frequently used in the wider UNIX community if not often in the HENP world. One major question concerns its scalability although other sites report no problems in this respect. Apart from such freely available tools, many vendors are rumoured to be working on such tools.

Cluster design: what tools exist? In particular, help would be appreciated in selecting the best cluster architecture and cluster interconnect. However, this only seems possible if the application workload can be accurately specified, and modelled. MPI applications should be an area where this could be possible.

7.0 Panel B1 - Data Access, Data Movement

The questions posed to this panel, chaired by Don Petravick, were ----

- What is the size of the data store and which tools are in use?
- How to deal with "free" storage (large local discs on modern PCs for example)?
- What protocol and software stacks are used to access this data across the LAN/WAN?

7.1 Sanger Institute (James Cuff)

ENSEMBL is a $14M scheme for moving data in the context of a programme for automatically annotating genomes. The users require 24-hour turnaround for their analyses. The files concerned are multi-GB in size. For example, using mySQL as the access tool, the databases may be more than 25GB and single tables of the order of 8-10GB. Sanger has found that large memory caches (for example 3GB) can significantly increase performance.

They experience trouble with NFS across the clusters so all binaries, configuration files and databases are stored locally and only the home directories are NFS-mounted because they're not high throughput or too volatile. They use MDP[30] for reliable multicast. It scaled to 40 MB/s in tests over 100BateT links (40 times 1MB/s) but it failed in production – incomplete files and file corruption! They have found that it is only possibly to multi-cast at speeds of up to 64KB/s reliably. The problem appears to be with "No Acknowledge" (NAKs) messages for dropped bits and the error follow-up of these.

7.2 **Condor I/O** (Doug Thain)

The University of Wisconsin has 640 CPUs in its Computer Science building; among other uses, these are used for computer architecture development on Condor. The 264 CPUs at INFN is largest single user community. It seems that one of the attractions for particle physics sites is the real case usage need. Is it a funding niche to ask for "integration" projects to take computer science projects to a "phase 3" with early usage testers. Do the funding agencies see value in these pilot projects or do they expect the results to be attractive enough to end users (commercial or academic) to where they will volunteer their own effort to integrate the research results?

Condor has as common denominator the principle to hide errors from running jobs but rather propagate failures back to the scheduler, eventually the end user, for recovery. This, along with remote I/O, is easily accommodated by linking against the Condor C libraries.

[30] The Multicast Dissemination Protocol (MDP) is a protocol framework and software toolkit for reliable multicasting data objects. It was developed by the US Navy.

A recent development is DAGMan[31], which could be thought of as a distributed UNIX "make" command with persistency.

Another development is Kangaroo. Under this scheme, the aggregate ensemble of network, memory and disc space is considered as large buffer for the full transfer. The goal is to allow overlaps of the computation and data transport. This presumes there is no (or acceptable) contention for CPU between computation and data transfer. Kangaroo is explicitly trading aggregate throughput for data consistency.

In summary, correctness is a major obstacle to high-throughput computing. Jobs must be protected from all possible errors in data access.

The question was asked on whether there had been any game theory analysis on where the optimization is in the "abort and restart" philosophy of simple checkpoint/restart. Experimentally users universally chose this method over checkpoint. Where is the breakeven point on job size/error frequency space?

7.3 **NIKHEF Data Management** (Kors Bos)

What is the current HEP problem size? The data rate is 10**9 events/year and there is a simulation requirement of a minimum of 10% of this (10**8 events/year). A typical CPU in NIKHEF's cluster typically requires 3 minutes to simulate an event, equals 10**5 events per year per CPU. Multiply that by 100 CPUs and there is still a factor of 10 too few nodes for the required simulation. Hence the need to aggregate data from different remote sites.

For the D0 experiment, the generated information is 1.5 GB of detector output per CPU plus 0.7GB simulation data. [A very important question relating to the physics of such simulations is how do they deal with the distribution of the random numbers/seeds.] They thus generate some 200GB of data per day overall and transmit half of this to FNAL.

Seeing that ftp throughput is limited by roundtrip latency across the network. They use bbftp[32] to bring this up to 25 Mb/s. With multiple bbftps they can get 45 Mb/s, limited mostly by the network connection between the Chicago access point of the transatlantic link and FNAL itself.

The conclusion is that producing and storing data is rather simple; moving it is less simple but still easy. The real issue is managing the data.

The bulk of the data is in storage of intermediate results, which are never used in reality. There are advocates of not storing these intermediate results but rather to recalculate them. Is there a market in less reliable storage for this bulk data?

7.4 **Genomics and Bio-Informatics in Minnesota** (Chris Dwan)

Their planning horizon is much compressed compared to the HEP presentations. They have the genome analysis in hand now that was originally planned for 2008. They are living with the bounty of the funding wave/success.

Their computing model is pipelined data processing. The problem size is 100KB and less than 5 seconds processing per "event". They run 10,000 jobs/month so the major problem is the job management and job dependency for users with lots of distributed jobs.

[31] DAGman stands for Directed Acyclic Graph Manager

[32] bbftp is described below, in the NIKHEF talk in the Small Site Session.

The users perform similarity searches, searches against one or more 10GB databases where the problem is data movement. They use servers for this. [Could an alternative be to put these on PCs with many replicas of the database, which seems to be the Sanger approach?]

The Bio-Informatics component is in data warehousing, including mirroring and crosschecking other public resources. They have local implementations of various public databases stored in ORACLE. They must store large databases of raw information as well as information processed locally with web-based display of results and visualization tools.

They have a 100-node Condor pool[33] using desktop PCs, shared with other users. The algorithms are heuristic and so there is a fundamental question about whether different results are erroneous or just produced on different hardware, for example some from 32 bit, some from 64 bit architectures.

Biophysics teams experience the same phenomenon as some particle physicists (see previous talk) that it may be easier to recollect the data than to migrate forward from old stored copies, at least for the next decade.

Is there a basis for the presumption that there will always be the need for local storage? What about the option of cataloging where the data is? What are the overheads of handling concurrent data accesses? What would be the result of comparing the growth curves of network speed with the local disk speeds? From pure performance reasons, which is likely to be the winner?

The question was raised about interest in iSCSI[34] for accessing disks remotely. Core routers are at the 6GB/s fabric level now. What is the aggregate rate of the local disk busses and how does it change?

There are concerns with the model of splitting data across multiple CPUs and fragmenting jobs to span the data. These relate especially to the complexity of the data catalog and with the job handling to make sure the sub-portions are a complete set and restart.

7.5 Discussion

100TB stores are handled today. Stores at the level of PB are still daunting. Network capabilities still aggregate well into a large capability. FibreChannel is used only in small scale between large boxes for redundancy. Local disk is still used for local disk buffer for a remote transfer.

There is lots of interest in multicast transmission of data, but current implementations have reliability problems. Parallelization of normal tools seems to be serving people well enough in clusters up to 100-200 nodes. This brings up the question of “do the sites managing clusters in units of 1000+ (SLAC, CERN, FNAL, etc) have any insights into mental limits that one hits?[35] It's relatively easy to create another layer of hierarchy in systems that already have at least one intermediate layer, but getting that abstraction in the first place is often quite difficult.

8.0 Panel A2 - Installation, Upgrading, Testing

The questions posed to this panel Steven Timm, chaired by, were ----

- Do you buy-in installation services? From the supplier or a third-party vendor?
- Do you buy pre-configured systems or build your own configuration?

[33] A cluster or farm in Condor terms is called a pool

[34] iSCSI transmits the native SCSI disc/tape access protocol over a layer of the IP stack.

[35] For example, consider the tool described above by Chuck Boeheim of SLAC that aggregates error reports across a large cluster into an easily digested summary.

- Do you upgrade the full cluster at one time or in rolling mode?
- Do you perform formal acceptance or burn-in tests?

8.1 Dolly+ (Atsushi Manabe, KEK)

Faced with having to install some 100 new PCs, they considered the various options ranging from buying pre-installed PCs to performing the installation on the nodes by hand individually. Eventually they came across a tool for cloning disk images developed by the CoPs (Clusters of PCs) project at ETH in Zurich and they adapted it locally for their software installations. Their target was to install Linux on 100 PCs in around 10 minutes.

The PXE bootstrap (Preboot Executive Environment) of their particular PC model starts a pre-installer, which in turn was modified to fire up a modified version of Redhat's Kickstart. This formats the disc, sets up network parameters and calls Dolly+ to clone the disc image from the master. The nodes are considered as connected logically in a ring and the software is propagated from one node to the next with shortcuts in the event of failure of a particular node. Such an arrangement reduces contention problems on a central server. Using SCSI discs, the target was achieved, just over 9 minutes for 100 nodes with a 4GB disc image, although the times are doubled if using IDE discs or an 8GB disc image. A beta version is available.

8.2 Rocks (Philip Papadopoulos, San Diego Supercomputer Center)

The San Diego Supercomputer Center integrates their clusters themselves, especially since up to now their clusters have been relatively modest in size. They tend to update all nodes in a cluster in one cycle since this is rather fast with Rocks. They have performed rolling upgrades, although this has caused configuration problems. They do not perform formal acceptance tests; they suggest that this would be more likely if there were more automation of such tests.

Installing clusters by hand has the major drawback of having to keep all the cluster nodes up to date. Disk imaging also is not sufficient, especially if the cluster is not 100% homogeneous. Specialised installation tools do exist, LUI (Linux Utility for cluster Installation) from IBM was mentioned again but San Diego believe that they should trust the Linux vendors and use their tools where possible – thus the use of Redhat's Kickstart for example. But they do require automation of the procedures to generate the configuration needed by Kickstart, which Rocks does.

They have evolved two cluster node types. One, front-end, for login sessions, the other for batch queue execution where the operating system image is considered disposable – it can be re-installed as needed. The Rocks toolkit consists of a bootable CD and a floppy containing all the required packages and the site configuration files to install an entire cluster the first time. Subsequent updates and re-installations are from a network server. It supports heterogeneous architectures within the cluster(s) and parallel re-installations – one node takes 10 minutes, 32 nodes takes 13 minutes; adding more nodes implies adding more HTTP servers for the installation. It is so trivial to re-install a node that if there is any doubt about the running configuration, the node is simply re-installed. Apart from re-installations to ensure a consistent environment, a hard power cycle triggers a re-installation by default.

The cluster-wide configuration files are stored in a mySQL database and all the required software is packaged in RPMs. They are currently tracking Redhat Linux 7.1 with the 2.4 kernel. They have developed a program called insert-ethers which parses the /var/log/messages for DHCPDISCOVER messages and extracts the MAC addresses discovered.

There is no serial console on their PCs so BIOS messages are not seen and this gives a problem to debug very tricky problems. They have considered various monitoring tools, including those based on SNMP, Ganglia (from UCB), NGOP from Fermilab, PEM from CERN, the Chiba City tools, etc but they feel more investigations are needed before coming to a decision.

8.3 European DataGrid Project, WP4 (Jarek Polok, CERN)

WP4, or Fabric Management, is the work package of the European DataGrid project concerned with the management of a computer centre for the DataGrid. There are many fabric management tools in common use today but how well do they scale to multi-thousand node clusters? Tools at that level must be modular, scalable and, above all, automated. One of the first tasks of WP4 therefore has been a survey of non-commercial tools on the market place.

Looking longer term, WP4's work has been split into individual tasks in order to define an overall architecture: -

- Software packages which consist of "data", dependency information and methods (of installation for example)
- Software repository scheme, which should include an interface for system administrators and which interfaces to the WP4 configuration system
- Node management scheme which performs operations on the software packages and also consults the WP4 configuration system
- Bootstrap service system
- Gridification (including user authentication and security issues)

The final installation scheme must be scalable to thousand+ node clusters and should not depend on software platform although at lower levels there may well need to be different implementations.

The Datagrid project has set month 9 (September) for a first demonstration of what exists and WP4 have decided for this first testbed to use –

- LCFG from Edinburgh University for configuration management
- Updaterpms, also from Edinburgh University, (and maybe ASIS from CERN) for environment tailoring
- SystemImager for installation

This is not a long-term commitment to these tools but rather to solve an immediate problem (to have a running testbed in month 9) and also to evaluate these tools further. The actual installation scheme will use Kickstart for Linux, JumpStart for Solaris.

8.4 Discussion

The various sites represented use virtually all possible schemes for system installation – some use the standard vendor-supplied method for the chosen operating system, others develop their own and use it; some develop their own environment and request the box supplier to install it before shipment; a few require the hardware supplier to install the chosen environment on site and some leave the entire installation to the supplier. However, there does seem to be a trend more and more towards network-based installation schemes of one type or another.

The LinuxBIOS utility from Los Alamos National Laboratory is used in some sites but there are reports that at least the current version is difficult to setup although once configured it works well. And since it is open source software, it permits a local site to make changes to the BIOS, for example adding ssh for secure remote access. A similar tool is BIOSwriter (available on sourceforge.org), which permits to clone BIOS settings across a cluster. It has several weaknesses, the most serious of which is that the time written into all the cloned BIOS's is the one stored when the master BIOS is read; thus all the times written are wrong. It also appears to fail on certain BIOS's.

Software upgrades too are handled differently: for example CERN performs rolling upgrades on its clusters; Kek performs them all at once, at least on homogeneous clusters; BNL environments are “locked” during a physics run except for security patches and this tends to concentrate upgrades together; SanDiego (with Rocks) performs the upgrade in batch mode all at once. SystemImager permits the upgrade to take place on a live system, including upgrading the kernel, but the new system is only activated at the following bootstrap.

VA Linux uses the CTCS tool (Cerberus Test Control System) for system burn-in. This checks the system very thoroughly, testing memory, disc access, CPU operation, CPU power consumption, etc. In fact if certain CPU flags are wrongly set, it can actually damage the CPU.

Finally, when asked how deeply various sites specify their desired system configurations, LBNL/NERSC and CERN noted that they go down to the chip and motherboard level. Fermilab used to do this also but have recently adopted a higher-level view.

9.0 Panel B2 - CPU and Resource Allocation

The questions posed to this panel, chaired by Jim Amundson, were ----

- Batch queuing system in use?
- Turnaround guarantees?
- Pre-allocation of resources?

9.1 BaBar (Charles Young)

Should one buy or develop a batch scheduling system? Development must take account of support and maintenance issues. Purchase costs, including ongoing support licences, start to be comparable to the costs of the latest cheap PC devices. Can one manage a single unit of 10,000 nodes? Will we need to? How does one define queues, by CPU speed; by the expected output file size; by I/O bandwidth; make them experiment-specific; dependent on node configuration; etc. Are these various alternatives orthogonal?

Pre-allocation of nodes reduces wait time for resources but it may reduce CPU utilization and efficiency. What level of efficiency is important?

What kind of priority guarantees must be offered for job turnaround? The speaker stated that "users are damn good at gaming the system".

9.2 LSF (David Bigagli, Platform)

The challenge in data movement was posed as "can we run the same job against multiple pools of data and collect the output into a unique whole?" Can LSF handle a list of resources for each machine, being a (dynamic) list of files located on the machine itself? Can it handle cases of multiple returns of the same data.

How can/should we deal with the "scheduler" problems? There is traditionally the tension between desires to let idle resources be used, but still provide some guarantee of turnaround.

Apart from LSF there are at least 4 other batch systems in use in HEP. What are the reasons? Answers from the audience include cost, portability, customization of scheduler/database integration, etc.

9.3 Discussion

The challenges for systems administrators are

- For network I/O, to use TCP or UDP
- How to efficiently maintain host/resource information
- Daemon initialization and the propagation of configurations changes

- Operating System limitations (file descriptions)
- Administrative challenges.

LSF has a large events file that must be common to the fallback server in order to take over the jobs. Other than that, failover works smoothly (as indicated by SLAC and Sanger). Problems with jobs finishing in the window during failover have to be dealt with. Typically, the frequency of failover is the hardware failure frequency of the master machine.

Most sites are using quad/dual Suns server for the master batch servers. FNAL for example has a single Sun. CERN adds disk pool space to the scheduling algorithm but that is the only element that people have added to the scheduling algorithms. On very restricted pools of machines of size 50-200 machines experiments are successfully submitting jobs without a batch system.

An audience census demonstrated that 30% use LSF and 70% use something else, made up of 10% using Condor (opportunistic scheduling. DAGMan component, match-making); 30% use PBS (price, sufficient to the needs, external collaboration, ability to influence scheduler design) impact of commercialization is unknown and worrisome; and 20% have local custom-built tools (historical expertise, sufficient to the needs, guaranteed access to developer, costs, reliability, capabilities to manage resources). Are there any teeth in the POSIX batch standard for batch? How many people even know there is such a thing?

Turning to AFS support in LSF, is the data encrypted in transfer? In LSF 4.0 the claim is that it is possible for root on the client to get the user's password via the token that was transferred. No one in the audience, nor the LSF speaker, recognized the problem.

On clusters of about 100 machines, DNS round robin load balancing works for interactive logins. MOSIX[36] is used by one site for ssh gateway redundancy to allow clean failover to another box. CERN finesses this by forcing the ssh key to be the same.

What is the most effective way of dealing with queue abusers? One site relies on user liaison to get the social pressure to change bad habits. Public posting of monitoring information gets peer pressure to reform abusers. However, in shared pools, people are adamant about pointing out "other" abusers.

How do sites schedule downtime?

- Train people that jobs longer than 24 hours are at risk.
- CERN posts a future shutdown time for the job starter (internal)
- BQS[37] has this feature inside.
- Condor has a daemon (eventd) for draining queues.
- Some labs reboot and have maintenance windows.

10.0 Small Site Session (Session chair: Wolfgang von Rueden (CERN)

10.1 NIKHEF (Kors Bos)

The NIKHEF data centre includes a 50 node farm with a total of 100 Pentium III CPUs dedicated to Monte Carlo event production for the D0 experiment at the Fermilab Tevatron; it is expected to double in size each year for the next few years. There is also a small, 2 node, test farm with the same configuration; this system is considered vital by the support team and is used for 90% of development and testing.

[36] MOSIX is a software package that enhances the Linux kernel with cluster computing capabilities. See the Web site at http://www.mosix.cs.huji.ac.il/txt_main.html.

[37] BQS is a batch scheduler developed and used at CCIN2P3. See talk from this site in the Small Site Session.

NIKHEF is connected to SURFNET – a 20Gbps backbone in Holland. NIKHEF are also members of a national Grid project now getting underway in the Netherlands.

The nodes run Redhat Linux version 6.2, booted via the network with tools and applications being made available via the UPS/UPD system developed by Fermilab. Also much appreciated from Fermilab, the batch scheme is FBSng. Jobs to be run are generated by a script and passed to FBSng; typically the script is setup to generate enough jobs to keep the farm busy for 7 days, lowering the administration overhead.

The data for D0 is stored in SAM – a database scheme developed by D0. SAM, installed at all major D0 sites, knows what data is stored where and how it should be processed. It could be considered an early-generation Grid. [In fact NIKHEF participate in both the European DataGrid project and a local Dutch Grid project.

Data is passed from the local file system back to Fermilab using ftp so some research has taken place on this particular application. Native UNIX ftp transfers data at up to 4Mbps, a rate partially governed by ftp's handshake protocol. Using bbftp developed at IN2P3, a maximum rate of about 20 Mbps has been achieved, with up to 45 Mbps using 7 streams in parallel. A development called grid-ftp has achieved 25 Mbps. Increasing ftp parallelism gives better performance but so far the absolute maximum seen is 60 Mbps and NIKHEF believes that ultimately 100 Mbps will be needed. However, at the current time, the bottleneck is not ftp but rather the line speed from Fermilab itself to the Chicago access point of the transatlantic link, a problem which is acknowledged and which it is planned to solve as soon as possible.

A current development in NIKHEF is research into building their own PC systems from specially chosen motherboard and chips. Using in-house skills and bought-in components, they expect to be able to build suitable farm nodes for only $2K (parts only) as compared to a minimum of $4K, from Dell for example for the previous generation of farm nodes. Once designed, it takes only an hour to build a node. So far, there is motivation among the group for this activity but will this last over a longish production run?

10.2 Jefferson Lab (Ian Bird)

The main farm at JLab is a 250 PC node cluster (soon to expand to 320 nodes) used mainly for reconstruction and some analysis of the data from their on-site experiments, now producing about 1TB of data per day. There is also some simulation and a little general batch load. Access is possible to the farm and its mass storage from anywhere on the JLAB site via locally-written software.

Jefferson specifies a particular motherboard and their desired chip configuration and then they get a supplier to build it. Their storage scheme is based on storage racks in units of 1TB and they can fit up to 5 such units in a rack. It used to be controlled by OSM but they have moved to a home-written software. On their backbone, they have had good experience with Foundary Gigabit switches, which give equivalent performance as the more common Cisco switches.

Jefferson also participate in the Lattice QCD project mentioned above; in partnership with MIT, they have a 28 dual node Compaq Alpha cluster installed, connected with Myrinet and front-ended by a login server and a file server. The first implementation of this cluster was with stand-alone box systems but the most recent version has been rack mounted by an outside vendor. A second, independent, cluster will be added in the near future, probably with 128 Pentium 4 PCs and this is expected to at least double later. The two clusters – Alpha and Pentium – cannot realistically be merged because of the parallel nature of the application, which virtually demands that all nodes in a cluster be of the same architecture.

Jefferson has developed its own UNIX user environment (CUE). This includes files accessed via NFS or CIFS from a Network Appliance file server. For batch production, Jefferson uses both LSF and, because of the licence costs of LSF, PBS and it has devised a scheme (Jsub) on the interactive nodes which is effectively a thin wrapper to permit users to submit jobs to either in a transparent manner. They have devised a local scheduler for PBS. Actual resource allocation is governed by LSF and LSF Fairshare. LSF is also used to publish regular usage plots. They have also developed a web interface to permit users to monitor progress of their jobs. There are plans to merge these various tools into a web application based on XML and including a web toolkit, which experiments can use to tailor it for their own use.

The Lattice QCD cluster uses PBS only but with a locally developed scheduler plug-in which mimics LSF's hierarchical behaviour. It has a web interface with user authentication via certificates. Users can submit jobs either to the QCD cluster in Jefferson or the one in MIT.

For installing PCs, Jefferson currently use Kickstart complemented by post-installation scripts but they are looking at PXE-equipped (Preboot Executive Environment) motherboards and use of the DHCP protocol in the future. For upgrades, the autoRPM tool is used but new kernels are installed by hand. Redhat version upgrades are performed in a rolling manner. There is no significant use yet of remote power management or remote console management.

For checkout of new systems they use Mprime, a public domain tool; system monitoring is done also with a public domain tool (mon), which they accept is perhaps simplistic but is considered sufficient for their needs. Statistics produced by LSF are collected and used for performance measurements.

A major issue is the pressure on floor space for new systems, especially with their need to add yet more tape storage silos. Operation of their centre is "lights-out".

Having covered the broad range of work done at the lab, Ian offered the definition of a "small site" as one where "we just have 2 to 3 times fewer people to do the same jobs" as at the larger sites.

10.3 CCIN2P3, Lyon (Wojciech Wojcik)

The computer centre at IN2P3 operates a multi-platform, multi-experiment cluster. It has to support a total of 35 different experiments spread across several scientific disciplines at sites in several continents. The centre has multiple platforms to support, currently 4 but reducing shortly to 3. The reasons for this are largely historical (the need to re-use existing equipment) but also partly dictated by limited staff resources. The architectures supported are AIX, Solaris and Linux with HP-UX being the one planned for near-term phase-out. There are three clusters – batch, interactive and data – all heterogeneous. The large central batch farm is shared by all the client experiments and users are assigned to a given group or experiment by executing a particular profile for that group or experiment.

Home directories are stored on AFS as are many standard utilities accessed in /usr/local. Data is accessed from local disc, staged there on request. Many experiments use RFIO for data access (originally from CERN, augmented locally in IN2P3); some experiments (for example BaBar) use HPSS and Objectivity for data access; others use Xtage to insulate the user from the tape technology in use (this is common at most sites now although many have their own locally-developed data stager). The batch scheduler used is BQS, a local development some time ago, still maintained and available on all the installed platforms. Jobs are allocated to a node according to CPU load, the architecture demanded, group allocations and so on.

The latest acquisitions at IN2P3 included a major PC purchase where the contract went to IBM who were responsible also for the physical installation in racks in the centre. IN2P3 is also now gaining experience with SANs[38]. They currently have some 35TB of disc space connected.

[38] SAN – Storage Attached Network

Being only a data processing site and supporting multiple experiments, IN2P3 see many problems in exchanging data, not only with respect to network bandwidth for online data transfer but also the physical expert and import of data. CERN was using HPSS and is now switching to CASTOR. SLAC (BaBar) uses HPSS. Fermilab (D0) uses Enstore. How can a small centre expected to cope with this range? Could we, should we, develop a higher level of abstraction when dealing with mass storage? In addition, they are expected to licence commercial products such as Objectivity in order to support their customers. Apart from the financial cost of this, it only adds to the number of parameters to be considered in selecting a platform or in upgrading a cluster. Not only must they minimise disruption across their client base, they must consider if a particular product or new version of a product is compatible with the target environment.

A similar challenge is the choice of UNIX environment to be compatible with their multiplicity of customers, for example, which version of a given compiler should be installed on the cluster (in fact they are currently required to install 3 versions of C).

11.0 Software Panel

The questions posed to this panel, chaired by Ian Bird, were ----

- How do you select software tools? By reputation, from conference reports, after in-house evaluation, by personal experience, etc. Obviously all of these may play a role – which are the 3 most important in order of significance
- Do you trade-off personnel costs against the cost of acquiring commercial tools?

11.1 CONDOR (Derek Wright, University of Wisconsin)

Condor is a system of daemons and tools that harness desktop machines and computing resources for high throughput computing. Thus it should be noted that Condor is not targeted at High Performance Computing but rather at making use of otherwise unused cycles. It was noted that the average user, even at peak times during the working day, seldom uses more than 50% of his or her CPU as measured on an hourly basis. Condor has a central matchmaker that matches job requirements against resources that are made available via so-called Class Ads published by participating nodes. It can scale to managing thousands of jobs, including inter-job dependencies via the recently developed DAGMan. There is also centralised monitoring of both jobs and nodes and a degree of fault tolerance. Making jobs checkpointable (for example by linking against Condor libraries) helps Condor but this is not essential. Owners of participating nodes can “vacate” their systems of current jobs at any time and at no notice.

Condor is easy to set-up and administer since all jobs and resources are under the control, or at least responsibility, of a central node. There are no batch queues to set-up and administer. Job control is via a set of daemons.

Examples of Condor pools[39] in current use include INFN HEP sites (270 nodes), the CHORUS experiment at CERN (100 nodes), Nasa Ames (330 nodes) and NCSA (200 nodes). At its main development site in the University of Wisconsin at Madison, the pool consists of some 750 nodes of which 350 are desktop systems. The speaker noted that the Condor development team consists largely of ex-system administrators and this helps focus development from that viewpoint.

One interesting feature of Condor is the eventd feature. This can be used to rundown a Condor node before a scheduled event takes place such as a reboot. For MPI applications, Condor can gather together a required number of nodes before scheduling the jobs in parallel.

[39] A cluster or farm in Condor terms is called a pool.

Current work in Condor includes the addition of Kerberos and X509 authentication, now in beta test, and the use of encrypted channels for secure data movement (this could include AFS tokens). A lot of work is going on to study how best to schedule I/O. Another issue for the future is Condor's scalability beyond 1000 nodes, which is (relatively at least) untested. Condor will shortly be packaged in RPM format for Linux but it should be noted that Condor is **NOT** Open Source. Binaries are freely available for about 14 platforms but the sources must be licensed. Another option is to contract for a support contract from the Condor team with guaranteed response times to submitted problems.

11.2 **Meta-Processor Platform** (Robert Reynolds)

This is based on the seti@home[40] scheme and is the largest distributed computing project in history. It currently has some 3 million users with over 1 million nodes for a total of 25 Teraflops! It can be packaged to gather resources across the Internet or within an Intranet. Current applications include cancer research as well as the more renowned search for extra-terrestrial intelligence (SETI). One particular application (cancer research) has more than 400,000 members on 600,000 nodes scanning more than 25,000 molecules per second; the sustained throughput is 12 Tflops per day with peaks reaching 33 Tflops.

The scheme is based on aggregating resources and measured in millions of devices: there is a small (the suggested maximum is 3MB of code plus a maximum of 3MB resident data), unobtrusive, self-updating agent on the node running at low priority just above the idle loop; indeed it can replace the screen saver. It communicates with a central master by way of an encrypted channel sharing code that has been authenticated by electronic signatures. The node owner or user has control over when this agent runs and when and how it should be pre-empted.

Agents and code are available for Linux (the initial platform) and Windows (where most of the current resources come from). Some 80% of the nodes or more have Internet connection speeds equivalent to ISDN or better. The scheduling involves sufficient redundancy, running the same job N times on different clients, to allow for the fact that the client owner may switch his or her system off at any moment.

For people interested to port their applications to this tool there is a software development kit for code written in C, C++ and Fortran. Apart from the above-mentioned applications, another example is to use this to stress-test a web site – how many simultaneous accesses can it cope with. This typical of the ideal application – course-grained parallelism, small application code footprint, small data transfer.

12.0 Panel A3 – Monitoring

The question posed to this panel, chaired by Olof Barring, was ----

- Do you monitor services or servers? In other words, do you monitor that a service is being delivered or that a particular hardware or software status is faulty

12.1 **BNL** (Tony Chan)

BNL uses a mixture of commercial and locally developed tools for monitoring. They monitor both hardware and software as well as the health of certain services such as AFS and LSF; for LSF they use tools provided by the vendor of the product. They also monitor the state of what might be called infrastructure – UPS state, cooling system, etc. Alarms are signalled by beeper and by electronic mail.

The VA Linux-developed tool VACM is used to monitor hardware system status which permits a limited number of actions to be performed, including a power cycle.

[40] Seti@Home is a scheme whereby users volunteer to run an application on their nodes, effectively using up idle time..

From open source modules they have produced a scheme whereby users have access via the web to the current state of CPU load and some similar metrics.

12.2 NGOP (Tanya Levshina, FNAL)

The NGOP project at Fermilab is targeted at monitoring heterogeneous clusters running a range of services for a number of groups. Its goals are to offer "active" monitoring, to provide problem diagnostics with early error detection and correction and also to display the state of the services. Before starting the project, the team looked at existing tools, public domain and commercial, and decided that none offered the flexibility or adaptability they felt they needed. Reasons cited included limited off-the-shelf functionality of some tools, anticipated difficulties of integrating new packages into the tools, high costs, both initial and ongoing support, and doubts about scalability without yet further investment. Instead they would have to develop their own.

The current, first, implementation targets exceptions and diagnostics and it monitors some 6500 objects on 512 nodes. Objects monitored include checking for the presence of certain critical system daemons and file systems, the CPU and memory load, the number of users and processes, disk errors and NFS timeouts and a few hardware measures such as baseboard temperatures and fan speeds. It stores its information in an ORACLE database, is accessed via a GUI and there is a report generator. It is mostly written in Python with a little C code. XML (partially MATHML) is used to describe the configurations. At the present time it monitors objects rather than services although NGOP users can use the framework to build deductive monitors and some system administrators have indeed done this, for example the mail support team.

Plans for the future include support for scaling up to 10,000 nodes, to provide a callable API for monitoring a client and to add historical rules with escalating alarms.

12.3 PEM (Olof Barring, CERN)

Like Fermilab, the CERN PEM team performed a tools survey and like Fermilab they decided that they required to build something tailored to their needs and available resources. Like the NGOP team, they designed an agent-based scheme but they decided that from the beginning they would target services, a correlation engine being part of the original plan. Scalability would be built in by incorporating brokers who would interface to the front-end monitoring agents, currently assigning a broker for every 50 agents. In the current, first, version of PEM, some 400 nodes are monitored with about 30 measurements taken every 30 seconds. It is planned soon to extend this test scheme to 1000 nodes.

Like NGOP, PEM is designed to be "active" – it should react and take corrective action if it recognises a problem such as a measurement out of defined limits. The recovery action can be implemented via a broker and/or signalled by an alarm to a registered listener.

PEM has a measurement repository to store all data from the brokers. The data is physically stored via JDBC into an ORACLE database; initially this gave scaling problems but work on the interfacing, especially a better implementation of JDBC and better threading of the code, solved this.

The correlation engine, today still only a plan, will transform simple metrics into service measurements. Currently, most monitoring code is written in scripts, which permits fast prototyping and easy access to protocols. In the future, PEM plans to duplicate all modules to allow for scaling to very large numbers of nodes.

The PEM prototype will be used for the first European DataGrid fabric monitoring tests (see below) with various constituent parts (for example, the configuration manager and the underlying data transport layer) being later replaced by those from the DataGrid project.

12.4 Discussion

There was a discussion about how much measurement data should be stored and for how long. CERN considers that saving historical data is essential in order to be able to plot trends. BNL agreed; they save all monitored data although in condensed mode (partially summarised). With the number of nodes eventually planned for the CERN centre, a PEM Database Cluster might become necessary.

For PEM, the current assigned resources are 14 people part-time, totalling about 4-5 FTE[41]. FNAL have about the same number of people contributing to NGOP but for a total of about 1 FTE only.

One of the most basic questions to answer is whether to buy a commercial product or to develop something in-house. In favour of the former is the sheer amount of local resources needed to develop a viable tool. IN2P3 for example had started a monitoring project last year with a single person and had been forced recently to abandon it (they are currently evaluating the NGOP and also discussing with the PEM team). On the other hand, commercial tools are typically very large, range from expensive to very expensive and need a lot of resources to adapt to local requirements on initial installation. However, their ongoing resource needs are usually much less than in-house projects, which often need much support over their lifetimes.

According to some of the audience, building the framework is not where most of the work is. The difficult part is to decide what to measure. Others disagreed: a good, scalable architecture is very important. And a subscription driven correlation engine is a good step. What communities is monitoring for?

- System administrators need it for monitoring status, performance and for alarms.
- Application writers need performance data.
- Everyone wants to know about exceptional performance states.

Apart from the sites listed above, other sites reported on various monitoring tools: SLAC has a locally-developed tool (Ranger) for relatively simple tasks such as to monitor daemons and restart those which die or hang. San Diego SuperComputer Centre use Ganglia from Berkeley, chosen because it was freely available and appeared to satisfy their immediate needs. LBNL (NERSC) looked at various public domain products (including mon and big brother) but found they did not scale well; they have now (in the last few weeks) settled on netsaint which seems better and more extensible. It relies on SNMP daemons. Another tool mentioned was SiteAssure from Platform: it is rule-based – if this condition is true, perform that action.

As described above, VACM relies on so-called “mayors” to control 30-40 nodes and scalability can be achieved by designating “super-mayors” to control mayors. But VACM only monitors node availability; how to measure a service which might depend on multiple nodes?

13.0 Panel B3 – User issues, security

The questions posed to this panel, chaired by Ruth Pordes, were ----

- Do you have written policies for users - non-abuse of the system, the right to check e-mail, the right to enforce password rules
- Do you have a dedicated security team?
- Do you permit access from off-site; do you enforce rules for this?

[41] FTE – Full Time Equivalent, a measure of human resources allocated to a task or project.

13.1 Fermilab Strong Authentication Project (Mark Kaletka)

As other large labs, Fermilab has established a set of rules that users must accept in order to use computer services. These include rules relating to security and Fermilab has created a security incident response team with well-practised response processes. A major concern of the team is data and file backup since the most damaging incidents are those that destroy or make data unavailable. There is rather less concern, at least among the users, about data privacy. Probably the largest concern however is that hacked systems can be used to launch further attacks both inside the lab and towards the world-wide Internet

A project is underway to implement strong authentication, which is defined as "using techniques that permit entities to provide evidence that they know a particular secret without revealing the secret." The project addresses approximately half of the problems that have been determined to be root causes of incidents. One of the basic tools being used is Kerberos version 5 with some local enhancements[42] and CryptoCard challenge and response one-time passwords. Rollout of the project has begun with the goal of creating a secure realm across the site by the end of 2001, with a few declared exceptions.

Strong authentication for access to computer farms brings extra issues such as the secure handling of host and user service keytabs[43] across the farm, authenticating processes which do not belong to an individual and so on.

In the discussion, it was remarked that encryption and security options could very badly affect the performance of data transfer. A major issue is how to deal with trust relationships of other realms: The project team are technically comfortable that this works because of internal tests with migrations but they have not yet gone through the steps of negotiating a trust with another HEP realm.

To the question on whether anyone has tested the Java ssh[44] client for the web access, apparently there was some initial interest but no production use.

13.2 Security at the University of Minnesota (Mike Karo)

The support team is charged with looking after the security of about 100 computers. Most of these have non-dedicated functions so any user can logon and do what they chose. This drives the need for any of the machines to accept logons.

The first obvious security method is to disable as many unnecessary services as possible. For example, only one machine accepts incoming telnet or ftp. How to enforce security when using the batch features? The simplest is to issue warning calls to the users to ask not to exhibit bad behaviour.

Unfortunately for them, the university mandates that they should not use firewalls because of the public funding for the university and the desire to keep broad public access.

They are looking at SunRays as a method of privacy and simplicity of administration but they discovered that the IP/UDP traffic for these machines does not handle network congestion well. They are using smartcards as physical authorization.

Another area of study is to understand best practices in the realm of human interface (GUI).

- Do you have to select/customize window managers?

[42] The code is available on request from Fermilab (contact M.Kaltetka) but subject to certain DoE restrictions on export.

[43] For each service using Kerberos, there must be a service key known only by Kerberos and the service. On the Kerberos server, the service key is stored in the Kerberos database. On the server host, these service keys are stored in key tables, which are files known as keytabs

[44] ssh – secure shell in UNIX

- What value have people found in developing the GUI/custom interfaces to, for example, batch services?
- Is this a support albatross for limited gain or is this an interface level that allows for the insertion of local necessary customizations/migration protection.

It was reported that Jefferson Lab users are looking for a common GUI to aid the training of new users and University of Minnesota users are looking for the same benefits. Users at Wisconsin require access from Palm Pilots and hence a more “esoteric” access method and format.

The question was raised on whether Globus allows for an abstraction layer above the various batch systems. There appears to be very limited experience with it yet.

The speaker was asked if he had looked at GCG[45]. It is apparently free to academic sites but it is heavyweight and rather keyed to the biophysics community.

13.3 Discussions

Do people have experience with LSTcsh[46] from Platform Inc.? It could be a useful method for novices to access the broader resources of an LSF cluster.

About 30% of the attendees were using Kerberos. One drawback is the effort in the creation of a centralized registry of accounts. NERSC has the additional problem of being a global resource provider.

About 30% people regularly run crack[47] on their clusters.

Wisconsin appears to have deployed a method of generation of Kerberos tickets from PKI [48]certificates. They offered the conclusion that Kerberos turns out to be not useful on a cluster and even impossible between clusters.

Question: Is anyone worrying about application authentication? It appears that the answer is no. Instead, people are currently trying to address the problem by restricting sensitive data to private networks, internal to the sensitive applications.

How will we protect against people using their Kerberos passwords for Web passwords?

What are people looking toward for the registry of large user databases. Globus will require a mapping interface at each site to present a list of users. We need to distinguish between authentication and authorization.

To the question of whether people are clear on the distinction between security policy and usage policy there was rather a lot of silence!

What do different sites do about dealing with the creating secure environments? Most farms are behind a firewall but, for example, the University of Wisconsin and FNAL are on the public net.

[45] The Genetics Computer Group (or 'Wisconsin') package is a collection of programs used to analyse or manipulate DNA and protein sequence data

[46] The tcsch shell (called tc-shell, t-shell, or trusted shell) has all of the features of csh plus many more. LSTcsh takes this one step further by building in LSF load balancing features directly into shell commands.

[47] Crack is a password-guessing program that is designed to quickly locate insecurities in Unix (or other) password files by scanning the contents of a password file, looking for users who have misguidedly chosen a weak login password. See the appendix from the previous version for more details.

[48] Public Key Infrastructure

Are there centralized methods of dealing with patch selection and installation in relation to security issues? There is usually someone charged with watching the lists and spreading the word. NERSC has 4-5 FTEs, Sanger has 1 dedicated security person, SLAC has 3, JLab has 1, and FNAL has 2. Some university colleges have a floor warden for scans and audits. The University of Wisconsin largely accepts centralized administration. The general theme in the audience seems to be that user administration of machines is a slippery slope to chaos. 100% of the people present administered their own machines.

In biophysics, the main concern is the loss of results, which translates directly into a loss of dollars. In addition, bio-physicists are concerned about being a target of upset people.

What sort of training is done on security issues? SLAC requires mandatory training for all users. Sanger has a public training series.

Are there scaling problems anticipated with 1000 node clusters? Uniformity of a cluster is a big advantage as is limited direct login to the machines. This helps immensely in dealing with the clusters. Getting the time for maintenance and reconfiguration for (urgent) security patches is probably the biggest question. Scalability of the administration tools will necessarily provide the capabilities for the turnaround of machine update.

14.0 Panel A4 – Grid Computing

This panel was chaired by Chuck Boeheim.

14.1 European DataGrid (Olof Barring, CERN)

The European DataGrid project is funded for 3 years by the European Commission for a total of some 10M Ecus. There are 6 principal partners and 15 associate partners. The project is split into 12 work packages, 5 for middleware, 3 for applications (HEP, Earth Observation and Bio-informatics), 2 concerned with networking and establishing a testbed and 2 for administration. The work package of most concern to the provision of clustering is Work Package 4, Fabric Management.

WP4 is charged with delivering a computing fabric comprised of the tools necessary to manage a centre providing grid services on large clusters. There are some 14 FTEs available spread across 6 partners. WP4 has identified 6 sub-tasks and the interfaces between these –

- Configuration management: databases of permitted configurations
- Software installation: including software repositories, bootstrap procedures and node management services
- Monitoring
- Fault tolerance
- Resource management
- Gridification, including security

The DataGrid project started in January 2001 and much of the current effort is gather the detailed requirements of the work packages, especially the interfaces between them, and to define a global architecture. The overall timetable specifies major milestones with testbed demonstrations of current grid functionality at 9, 21 and 33 months. The first prototype, due this September, will be based largely on Globus but in the longer term the work packages will match the user requirements (now being gathered) with what is available and what can be developed.

For this first prototype, WP4 will be able to provide an interim installation scheme and will make available only low-level queries. It will use LCFG from the University of Edinburgh for software maintenance, SystemImager from VA Linux for the initial software installation and VACM, also from VA Linux for console control.

It is assumed that a fabric will execute local jobs as well as jobs submitted via the Grid and a means must be found for these to co-exist. WP4 must take account of multiple operating systems and compiler combinations and the question is whether to store these centrally and replicate them at Grid sites or request them on demand or store them on local discs or send them along with client jobs.

14.2 **PPDG/GriPhyn** (Ruth Pordes, FNAL)

PPDG[49] is a US project involving 6 High-energy and Nuclear Physics experiments and 4 Computer Science groups as well as 4 US scientific laboratories. Among the partners are the Globus team, the Condor team and the producers of SRB (Storage Request Broker from the San Diego SuperComputer Center). In all there are some 25 participants. It is expected soon to be funded by the US Department of Energy (DoE) for 3 years. It is a follow-on to a similarly named smaller project, which emphasised networking aspects only. In this incarnation the proposal is to build an end-to-end integrated production system for the named experiments (see overhead).

To date, PPDG work has achieved 100 MB/sec point-to-point file transfers using common storage management interfaces to various mass storage systems. It has also developed a file replication prototype (GDMP – Grid Data Mirroring Package). The proposal to the DoE lists a number of milestones and deliverables and, in detail, many of these and the tasks required to achieve them resemble those of the European DataGrid, with the notable exception that the US project has no equivalent of the DataGrid's Work Package 4, Fabric Management.

The project is in an early stage but already there are concerns that the various client experiments need to resolve their differences (for example in data handling models) and work together, especially since they are at different stages of their lifecycles (BaBar in production; FNAL Run II just getting started; the LHC experiments in development and will be for some time).

Going into some detail on PPDG activities, the first major one is to develop a prototype to replicate Monte Carlo data for the CMS experiment between CERN and some remote sites. There is a first prototype for Objectivity files with flat file support to be added next.

A second activity is to provide job definition and global management facilities for the analysis of Fermilab's D0 experiment. The starting point is D0's existing SAM "database" (see the Nikhef cluster talk in Section 10 above) which offers file replication and disc caching). Condor services will be used in this activity.

GriPhyN[50] is a 3-year project, funded by the National Science Foundation (NSF). It is essentially a computer science research project involving 17 university departments plus the San Diego SuperComputer Center, 3 US science labs and 4 physics and astrophysics experiments. It started formally in September 2000 and has some 40 participants. GriPhyn addresses the concept of virtual data – is it "cheaper" to access the processed data from a remote site or recalculate it from the raw data? The project should develop a virtual data toolkit offering transparency with respect to location but also with respect to materialization of the data. The results should be applicable to a Petascale Datagrid.

University computer science departments will do much of the work and the developed software should be stored in a central repository and made available in the public domain. This differs from the PPDG model

[49] PPDG stands for Particle Physics Data Grid
[50] GriPhyn stands for Grid Physics Network

where work will be done across the collaboration, should transition to the participating computer science departments and experiments would get the software from there.

In summary, the high-level challenge facing all the Grid projects is how to translate the technology into something useful by the clients, how to deliver the G word? An added complication is the mushrooming of grid projects, European and US. How to make them communicate and interact, especially since many have common user groups and experiments?

15.0 Panel B4 -- Application Enviroment, Load Balancing

The questions posed to this panel, chaired by Tim Smith, were ----

- What kinds of applications run on the cluster?
- Does the cluster support both interactive and batch jobs
- Is load balancing automatic or manual?

15.1 CDF Online Cluster (Jeff Tseng, MIT)

The principle application for this cluster is real-time event filtering and data recording. The movement from mainframe to clusters is a phenomenon seen in the experimental online world also. [In fact, can this not be extended to the general question of instrumentation and business?] The CDF online cluster consists of some 150 nodes, dual CPU Pentium IIs and IIIs running Linux (expected to expand to 250 nodes by the end of 2001). Their target configuration was one costing less than $2000 per node and they have usually been able to pay less than this. The systems are packaged in commodity boxes on shelves. They are interconnected by a bank of 9 3COM Fast Ethernet switches.

Maximum data logging rate at CDF under Run II conditions is 20 MB/s output and this is the current limiting bottleneck[51]. Availability of the cluster must be 100% of the accelerator operation to avoid losing data so there is no regular schedule for maintenance possible. The data taking is subject to a 5 minute start/stop warnings and long uptimes ("data waits for no one"). CORBA is used to ensure truly parallel control operations. LAN performance is crucial to operation and this has driven them to install a private network for the online cluster. In short, data taking down time must be determined by the Fermilab Collider, not by the cluster. The nodes are largely interchangeable; they are tested on arrival as follows:

- against the Fermilab standard benchmark ("tiny")
- the CPUs are tested under load for heating effects;
- the disc I/O rates are tested against the specifications

On the cluster they are running the full offline environment and they are using this for the trigger. This raises the problem of delivering an executable to an arbitrary machine and more work is needed because the problem is not solved yet. Filter executable and libraries are ~100 MB. Trigger table and databases are also ~100 MB and they must regularly distribute 100MB files to 150 machines within 5 minutes. For this they have developed a pipelined copy program, which is MPI-like without being MPI[52].

A major effort has gone into online monitoring of the data because the discovery of data corruption further down the data chain causes great problems. There is a high-throughput error reporting scheme with message collation dealing of periodic status reports every 4 seconds per node!

15.2 Application Environment Load Balancing (Tim Smith, CERN)

Question: how do you ensure that the user environment is the same across the production machines?

[51] The input rate from the data acquisition system to the ATM switch peaks at 260 MBps.

[52] MPI (Message Passing Interface) is a library specification for message passing.

- FNAL uses group accounts that they maintain for the login accounts; builds are done by the users on a common build platform or using their own tools. Users are expected to provide specialized tools wherever possible.
- SLAC uses AFS for home directories and binaries/products. Users pre-compile their codes interactively. They are encouraged to use dedicated build machines available through the batch build machines.
- The Sanger Institute's Pipeline has its own dedicated user that the jobs run under so they get a common environment. Direct logins on the worker nodes are administered by Tru64 UNIX directly.
- CERN distributes home directories with AFS and uses SUE/ASIS for making the same products available.

Given access to your personal home directory do people not need the group account structure to keep things together. Most sites said individual accounts are sufficient and they didn't need group accounts.

What tools do people use to keep track of the environment? We see occasionally that dynamic libraries are changed underneath the executables and that there is an unknown dependency. One technique seems to be to statically link the executable. However, there are some reasons not to statically link codes:

- improvements in the library may be desired and would be unavailable
- sometimes 3rd party libraries are only available as shared libraries.
- static links don't allow for dynamic changes of infrastructure without redistributing the binaries.

On the other, one argument in favour of static linking is to want to use variations on the machine configuration (Operating System versions.

Can we merge the static and dynamic linking? Only the infrastructure things independent of the OS could be dynamic.

On the subject of load balancing, attention was drawn to ISS[53], which IBM has made into a product and CERN has developed further. ISS is the domain name system (DNS[54]) component of the Network Dispatcher that can be used for wide-area load balancing. ISS balances the load on servers by communicating with load-monitoring daemons installed on each server machine, and then it alters the IP address returned to the client via DNS. Creating too many IP domains could create its own problems.

15.3 Discussion

Question: who uses over-subscription techniques or dynamic load level dependent on the job type. The most common technique appears to be to allocate more concurrent processes than the total number of processors on large SMPs, but to operate with tight restrictions on the worker machines.

Lsbind, authored by a Washington University professor, is a program that can submit to nodes that run LSF directly.

LBNL uses the technique of putting batch queues on interactive machines to utilise
Background cycles on lightly loaded machines.

SLAC has a DNS daemon that performs round robin scheduling and is run on all the members of the cluster.

The Sanger Institute reports that the DEC TruCluster software takes care of selecting machines within the cluster. They believe that the algorithm is not sophisticated and this works reasonably well.

[53] ISS – Interactive Session Support

[54] DNS is the Domain Name Scheme used to translate a logical network name to a physical network address.

One possible disadvantage of some load balancing techniques is that finding the actual node names on which jobs are running is very useful for retrieving data from partially completed jobs and for diagnosing errors.

Sanger uses technique of wrapping the critical executables with a maximum CPU-time script to prevent runaway. There are a number of sites that run watcher daemons that renice[55] or kill processes that run amok.

There are differences among sites on whether they allow mixing of interactive and batch processing on the same machine.

djns is a name server from David Bernstein that has alternate scheduling algorithms.

The next subject discussed was job and queue management. SLAC and FNAL both delegate the queue management to the users. SLAC does this queue by queue from a common cluster and this works fine. How does this work with Condor? CONDOR delegation would be done by access control with the configuration files. SLAC use the ranking system for determining whose jobs they prefer. Uni Minnesota does this by gang-editing of farm machine configuration files. Is there another method? Is this a hole in dealing with dynamic priorities within stable group definitions?

What to do with the "pre-exec" scripts to trap and react to job failures? How do you distinguish between single rogue nodes and problem cases where a failed server can blacklist a whole cluster? The feeling is that declaring a node "bad" is an expensive failure mode for transient errors (which can be the class of problem that can most rapidly scan the whole cluster). Condor jobs are by default restarted and can be flagged to leave the queue on failure.

Queuing in Condor at the level of about 100,000 jobs sees to cause trouble. LSF seems to not have experience here against lists of more than 10K jobs.

Job chunking in LSF is a method of reducing the demands on the central scheduler when there are lots of small jobs. This sets up an automatic chain particularly useful for small jobs.

One needs good tools for finding the hotspots and the holes in execution so that people can find when the systems run amok. Effort is underway in Condor to determine the rogue machines and to provide methods for jobs to "vote" against machines and find the anti-social machines. Will this all scale up to clusters of 5000 machines?

New question: given a 10K node cluster with 100K jobs a day, how do you digest the logs to present useful information to system administrators, let alone the individual users? What sort of reports are you going to need to show to the management and the funding agencies in order to justify decisions on the cluster? This seems to be a fruitful area for exploration of needs and capabilities. “Digesting such reports is going to be a high performance computing problem itself in these type of cluster" according to Randy Melen of SLAC. What kind of trend reporting is needed? Do we need to start digesting all these logs into a database and using database tools against the information pool? What sort of size of a problem does this end up dealing with, or creating?

Given the usual techniques of dealing with log file growth (roll over on size) there will be a very small event horizon on knowing what is going on with these machines. Will it even be possible given the rate of growth and will there need to be high performance logging techniques necessary to bring to bear?

[55] renice is a UNIX command to adjust the priority of a running job.

16.0 Panel Summaries (Session chair: Alan Silverman)

16.1 Panel A1 - Cluster Design, Configuration Management

Some key tools[56] that were identified during this panel included --

- cfengine – apparently little used in this community (HENP) but popular elsewhere. Does this point to a difference in needs? Or just "preferences"; doubts were expressed as to its scalability but these are thought to be unfounded
- SystemImager and LUIS (Linux Unique Init System)– neither does the full range of tasks for configuration management but there are hopes that the OSCAR project will merge the best features of both
- NFS – warnings about poor scaling in heavy use
- Chiba City tools – these do appear to be well-designed and to scale up well

Variations in local environments make for difficulties in sharing tools, especially where the support staff may have their own philosophy of doing things. However, almost every site agreed that having remote console facilities and remote control of power was virtually essential in large configurations. On the other hand, modeling of a cluster only makes sense if the target application is well defined and its characteristics well known – this seems to be rarely the case among the sites represented.

16.2 Panel A2 - Installation, Upgrading, Testing

Across the sites there are a variety of tools and methods in use for installing cluster nodes, some commercial, many locally-developed. Some of the key tools identified for software distribution were:

- Dolly+, used at Kek as a mechanism that uses a ring topology to support scalable system builds -- it is still in its testing phase and has not been tested at scale. It appears to resolve contention problems where too many clients access a central server at one time.
- Rocks (http://rocks.npaci.edu) is a cluster installation/update/management system, which automates the process of creating a Kickstart file. It is based on standard RedHat 7.1, has been tested on clusters up to 96 nodes (< 30 minutes to install all 96 nodes), is currently used on more than 10 clusters and is freely available now. In addition, Compaq references Rocks as their preferred cluster integration for customers wanting a 100% freeware solution.
- CERN has an installation strategy currently being employed as part of their Grid project that leverages Redhat tools and basic shell scripts. They must have something working by September of this year.

Many sites purchase hardware installation services but virtually all perform their own software installation. Some sites purchase pre-configured systems, others prefer to specify down to the chip and motherboard level.

Three sites gave examples of burn-in of new systems: FNAL performs these on site and BNL and NERSC requires the vendor to perform tests at the factory before shipment. VA Linux has a tool, CTCS, which tests CPU functions under heavy load; it is available on Sourceforge. On the other hand, cluster benchmarking after major upgrades is uncommon – the resources are never made available for non-production work! Some interest was expressed that the community could usefully share such burn-in tests as exist. But finally the actual target applications are the best benchmarks.

[56] This summary, indeed this workshop, does not attempt to describe in detail such tools. The reader is referred to the web references for a description of the tools and to conferences such as those sponsored by Usenix for where and how they are used.

Most sites upgrade clusters in one operation but the capacity for performing rolling upgrades was considered important and even vital for some sites.

16.3 Panel A3 – Monitoring

Of the three sites that gave presentations, BNL use a mixture of commercial and home-written tools; they monitor the health of systems but also the NFS, AFS and LSF services. They have built a layer on top of the various tools to permit web access and to archive the data. FNAL performed a survey of existing tools and decided that they required to develop their own, focusing initially at least on alarms. The first prototype, which monitors for alarms and performs some recovery actions, has been deployed and is giving good results.

The CERN team decided from the outset to concentrate on the service level as opposed to monitoring objects but the tool should also measure performance. Here also a first prototype is running but not yet at the service level.

All sites represented built tools in addition to those that come “free” with a particular package such as LSF. And aside from the plans of both Fermilab (NGOP) and CERN (PEM) ultimately to monitor services, everyone today monitors objects – file system full, daemon missing, etc.

In choosing whether to use commercial tools or develop one’s own it should be noted that so-called “enterprise packages” are typically priced for commercial sites where downtime is expensive and has quantifiable cost. They usually have considerable initial installation and integration costs. But one must not forget the often-high ongoing costs for home-built tools as well as vulnerability to personnel loss/reallocation.

There was a discussion about alarm monitoring as opposed to performance monitoring. System administrators usually concentrate first on alarms but users want performance data.

A couple of other tools were mentioned – netsaint (public domain software used at NERSC) and SiteAssure from Platform.

16.4 Panel A4 – Grid Computing

A number of Grid projects were presented in which major HENP labs and experiments play a prominent part. In Europe there is the European DataGrid project, a 3-year collaboration by 21 partners. It is split into discrete work packages of which one (Work Package 4, Fabric Management) is charged with establishing and operating large computer fabrics. This project has a first milestone at month 9 (September this year) to demonstrate some basic functionality but it is unclear how this will relate to a final, architectured, solution.

In the US there are two projects in this space – PPDG and GriPhyN. The first is a follow-on to a project that had concentrated on networking aspects; the new 3-year project aims at full end-to-end solutions for the six participating experiments. The second, GriPhyN incorporates more computer-science research and education and includes the goal of developing a virtual data toolkit – is it more efficient to replicate processed data or recalculate from raw data. It is noted that neither of these Grid projects has a direct equivalent to the European DataGrid Work Package 4. Is this a serious omission? Is this an appropriate area where this workshop can provide a seed to future US collaboration in this area?

All these projects are in the project and architecture definition stage and there are many choices to be made, not an easy matter from among such wide-ranging groups of computer scientists and end-users.

How and when will these projects deliver ubiquitous production services? And what are the boundaries and overlaps between the European and US projects?

It is still too early to be sure how to translate the “G word” into useful and ubiquitous services.

16.5 Panel B1 - Data Access, Data Movement

A number of sites described how they access data. Within an individual experiment, a number of collaborations have worldwide “psuedo-grids” operational today. These readily point toward issues of reliability, allocation, scalability and optimization for the more general Grid. Selected tools must be available free for collaborators in order to achieve general acceptance. For the distribution of data, multicast has been used but difficulties with error rates increasing with data size have halted wider use.

The Condor philosophy for the Grid is to hide data access errors as much as possible from reaching the jobs.

Nikhef are concerned about network throughput and they work hard to identify each successive bottleneck (just as likely to be at the main data center as the remote site). Despite this however, they consider network transfers much better than those transporting physical media. They note for future experiments that a single active physics collaborator can generate up to 20 TB of data per year. Much of this stored data can be recreated, so the challenge was made: “Why store it, just re-calculate it instead.”

The Genomics group at the University of Minnesota reported that they had been forced to use so-called “opportunistic cycles” on desktops via Condor because of the rapid development of their science. The analysis and computation needs expected for 2008 had already arrived because of the very successful Genome projects.

Turning to storage itself as opposed to storage access, one question is how best to use the capacity of the local disc, often 40GB or more, which is delivered with the current generation of PCs. Or, with Grid coming (but see the previous section), will we need any local storage?

16.6 Panel B2 - CPU and Resource Allocation

This panel started with a familiar discussion - whether to use a commercial tool, this time for batch scheduling, or develop one’s own. Issues such as vendor licence and ongoing maintenance costs must be weighed against development and ongoing support costs. A homegrown scheme possibly has more flexibility but more ongoing maintenance.

A poll of the sites represented showed that some 30% use LSF (commercial but it works well), 30% use PBS (free, public domain), 20% use Condor (free and good support) and 2 sites (FNAL and IN2P3) had developed their own tool. Both IN2P3 (BQS) and FNAL (FBSng) cited historical reasons and cost sensitivity. CERN and the CDF experiment at FNAL are looking at MOSIX but initial studies seem to indicate a lack of control at the node level. There is also a low-level investigation at DESY of Codine[57], recently acquired by SUN and offered under the name SUN Grid Engine.

16.7 Panel B3 – Security

Large sites such as BNL, CERN and FNAL have formal network incident response teams. Both of these have looked at using Kerberos to improve security and reduce the passage of clear text passwords. BNL and Fermilab have carried this through to a pilot scheme. On the other hand, although password security is

[57] Codine is a job queuing system made by GENIAS Software from Neutraubling, Germany

taken very seriously, data security is less of an issue; largely of course because of the need to support worldwide physics collaborations.

Other security measures in place at various sites include --

- Disabling as many server functions as possible
- Firewalls in most sites, but with various degrees of tightness applied according the local environment and the date of the most recent serious attack!
- Increasing use of smartcards and certificates instead of clear-text passwords
- Crack for password checking, used in some 30% of sites

Many sites have a security policy; others have a usage policy, which often incorporates some security rules. The Panel came to the conclusion that clusters do not in themselves change the issues around security and that great care must be taken when deciding which vendor patches should be applied or ignored. It was noted that a cluster is “a very good error amplifier” and access controls help limit “innocent errors” as well as malicious mischief.

16.8 Panel B4 - Application Environment, Load Balancing

When it comes to accessing system and application tools and libraries, should one use remote file sharing or should the target client node re-synchronise with some declared master node. One suggestion was to use a pre-compiler to hide these differences. Put differently, the local environment could access remote files via a global file system or issue puts and gets on demand; or it could access local files, which are shipped with the job or created by resynchronisation before job execution.

The question “dynamic or statically-linked libraries” was summarised as – system administrators prefer static, users prefer dynamic ones. Arguments against dynamic environments are increased system configuration sensitivity (portability) and security implications. However, some third-party applications only exist in dynamic format by vendor design.

As mentioned in another panel, DNS name lookup is quite often used for simple load balancing. Algorithms used to determine the current translation of a generic cluster name into a physical node range from simple round-robin to quite complicated metrics covering the number of active jobs on a node, its current free memory and so on. However, most such schemes are fixed – once a job is assigned to a job, it does not move, even if subsequent job mixes make this node the least appropriate choice for execution. These schemes worked well, often surprisingly well, at spreading the load.

Where possible, job and queue management are delegated to user representatives and this extends as far as tuning the job mix via the use of priorities and also host affiliation and applying peer pressure to people abusing the queues. Job dispatching in a busy environment is not easy – what does “pending” mean to a user, how to forecast future needs and availability.

17 Summary of the SuperComputing Scalable Cluster Software Conference

This conference in New England ran in parallel to the Workshop and two attendees, Neil Pundit and Greg Lindahl) were kind enough to rush back from it to report to us. The conference is aimed at US Department of Energy sites; in particular the ASCI super computer sites[58]. These sites are much larger in scale to those represented in the workshop, often having above 1000 nodes already. Indeed some sites already have 10,000 node clusters and their questions are how to deal with 100,000 nodes. Clearly innovative

[58] ASCI is the Accelerated Strategic Computing Initiative funded by the US Department of Energy.

solutions are required but they usually have access to the latest technology from major suppliers and, very important, they have the resources to re-architecture solutions to their precise needs.

Among the architectures discussed were QCDOC (QCD on a chip using custom silicon) and an IBM-developed ASIC using a PowerPC chip with large amounts of on-board memory. It must be noted that such architectures may be good for some applications (QCDOC is obviously targeted at QCD lattice gauge calculations) but quite unsuited to more general loads.

A new ASCI initiative is called Blue Lite and based at Lawrence Livermore National Laboratory. A 12,000-node cluster is now in prototype and the target is for a 64,000 nodes to deliver 384 TeraFlops in 2003.

The DoE has launched an initiative to try to get round software barriers by means of a multi-million dollar project for a Scaleable System Software Enabling Technology Center. Groups concerned are those in computer science and mathematics. One site chosen to work on this is the San Diego SuperComputing Centre and other sites should be known shortly.

As stated above, most of the work described at the conference was commercial or proprietary but one of the most interesting talks came from Rich Ferri of IBM who spoke about Open Source software. He noted that there are many parallel projects in many computing fields with a high degree of overlap, some collaboration but a lot of competition. This results in many projects dying. He noted that when such projects are aimed at our science, we are rather prone at winnowing out overlapping projects.

18 Cplant (Neil Pundit, Sandia National Laboratories)

Cplant is a development at Sandia Labs, home of ASCI Red as well as others of the world's largest clusters. In fact, Cplant is based on ASCI Red and attempts to extend the features of it.

Cplant is variously described as a concept, a software effort and a software package. It is used to create Massively Parallel Processors (MPPs) at commodity prices. It is currently used at various sites and projects and is licensed under the GPL (Gnu Public Licence) open source licence but it has also been trade-marked and licensed to a firm for those sites wishing guaranteed support.

It can in principle scale to 10,000 node clusters although thus far it has only been used on clusters ranging from 32 to 1024 nodes[59]. Clusters are managed in "planes" of up to 256 nodes with I/O nodes kept separate and worker nodes being diskless. The configuration consists of Compaq Alpha workstations running a Linux kernel, portals for fast messaging, home-grown management tools and a parallel version of NFS known as ENFS (Extended NFS) where special techniques are added to improve file locking semantics above those of standard NFS. ENFS offers high throughput with peaks of 117 MBps using 8 I/O servers connected to an SGI Origin 2000 data feeder. Other tools in use include PBS for batch queues with an improved scheduler, Myrinet for high speed interconnects and various commercial software tools such as the TotalView debugger[60]. Myrinet has been the source of a number of incidents and they have worked with the supplier to solve these. Over the past 2 years, it has required some 25 FTE years of effort of which ENFS alone required 3 FTE years.

During the building-up phase of the cluster, there appears to be a transition at around 250 nodes to a new class of problem. Many errors are hidden by (vendor) software stacks; often, getting to the root cause can be difficult. Among other lessons learned during the development was not to underestimate the time needed for testing and releasing the package. Not enough time was allowed for working with real

[59] This cluster is 31st in the November edition of the Top500 SuperComputer list.

[60] from Etmus Inc.

applications before release. A structured, 5 phase test scheme is now being formalised with phases ranging from internal testing by developers and independent testers to testing by external “friendly testers”.

19 Closing

The workshop ended with the delegates agreeing that it had been useful and should be repeated in approximately 18 months. No summary was made, the primary goal being to share experiences, but returning to the questions posed at the start of the workshop by Matthias Kasemann, it is clear that clusters have replaced mainframes in virtually all of the HENP world, but that the administration of them is far from simple and poses increasing problems as cluster sizes scale. In-house support costs must be balanced against bought-in solutions, not only for hardware and software but also for operations and management. Finally the delegates agreed that there are several solutions for, and a number of practical examples of, the use of desktops to increase overall computing power available.

It was agreed to produce a proceedings (this document) and also to fill in sections of the Guide to Cluster Building outline circulated earlier in the week. Both these will be circulated to all delegates before publication and presented at the Computing in High-energy Physics conference in Beijing in September (CHEP01) and at the Fall HEPiX meeting in LBNL (Lawrence Berkeley National Laboratory) in October.

Having judged the workshop generally useful, it was agreed to schedule a second meeting in about 18 to 24 months time.

Alan Silverman
18 January 2002