;login:

THE MAGAZINE OF USENIX & SAGE

August 2001 • Volume 26 • Number 5

Special Focus Issue:Clustering

Guest Editor: Joseph L. Kaiser

inside:

CLUSTERS

PICKING CLUSTER PARTS: CLUSTER CONSTRUCTION AT THE GENOME SEQUENCE CENTRE

by Martin Krzywinski

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

picking cluster parts: cluster construction at the genome sequence centre

by Martin Krzywinski

Martin Krzywinski's background is in biochemistry and physics. He has been involved in medical imaging (M.Sc. PET imaging, UBC), stochastic simulation and web programming and design. He is currently the bioinformatics coordinator at the Genome Sequence Centre in Vancouver.

martink@bcgsc.bc.ca

When we decided to purchase our cluster, it seemed that everyone had a different idea of what one should look like. Although the hardware budget largely influences the specifications of a cluster, even those built with similar budgets, there were many solutions. Whether or not to go with a turn-key solution from a vendor was a difficult choice – again largely motivated by the total cost. We had enough expertise to build the nodes from parts and assemble the cluster, but the idea of a pre-assembled solution held its own attraction. If we could have a system shipped to us, already in a rack, much of the tedious assembly time would be saved. That is, if the pre-assembly did not cost too much. On the other hand, better cluster price-performance was probably found in commodity off-the-shelf (COTS) units.

Our cluster had an additional important restriction: physical space. The lab does not have a spacious computer room, and the network room which would house the cluster is approximately 100 sq. ft. – basically large enough to fit three computer racks and a small chair and table. We knew right away that cooling would be of significant concern in such a small space. We had to purchase slim components, since every 1U in the rack counted.

This article describes steps and decisions we made in choosing the hardware for the cluster.

It should be noted that the mention of vendor-specific information is done purely for added information and reference. At the outset, we had no particular vendors in mind and arrived at our final solution based largely on the functionality and value of components rather than the available service options.

LINKS:

Genome Sequence Centre:
http://www.bcgsc.bc.ca

Raidion storage:
http://www.raidionsystems.com/

VA Linux:
http://www.valinux.com

Linux:
http://www.linux.com

Overview and Computing Facilities

The Genome Sequence Centre is a high-throughput facility for carrying out research in bioinformatics and genomics, as well as mapping mammalian genomes (mouse, rat, cow). The Centre is also involved in sequencing projects such as the full-length human cDNA project. The Centre was founded in early 1999 by the late Nobel laureate Dr. Michael Smith. We moved into our new facility at the British Columbia Cancer Agency in December of 1999.

The Centre's computing platform was built around a dual Xeon VA Linux server with 200GB of RAID5 storage provided by a redundant, external Raidion controller. The philosophy behind designing the computing environment was based on the concept of a powerful workstation which did not have to rely on a central server for CPU power, but only for files through NFS. So, while CPU resources are decentralized, much like in a logically managed set of computers, the disk storage and other resources, such as mail and Web service are centralized. All workstations are dual Pentium III platforms (500MHz or higher) with 512MB of RAM and a local system disk. Currently, there are several file servers with total online storage of nearly 3TB. The network is 100Mbps, fully switched, with a gigabit backbone.

With our wide use of Linux in the lab, in-house expertise and widely available binaries, it was natural to pick Linux as the operating system for the cluster.

Motivation for a Cluster

The field of bioinformatics is famous for its large data sets and diverse, numerous databases. There are hundreds of public databases of biological information. Attempts have been made to centralize some of this information. To be sure, not only do such data sets require large and relatively fast storage solutions, analysis and indexing of this data is a time-consuming task – ideal for clusters.

One example of cluster application is in the building of a physical map of a genome. The process requires *N* x *N* comparisons of objects, and the comparison function is very expensive. With *N* being on the order of 3 million, the process is exhausting on a single computer. In early 2000, the physical map of the human genome took about two weeks to assemble on a 900MHz K7 system (Kryotech's super-cooled 900MHz system). Luckily the process is amenable to being parallelized, since the comparisons are independent. By using existing workstations and parallelizing the software used to build the map (FPC), we were able to reduce the time of a build to less than 24 hours.

The drastic speed increase meant that we could not only assemble the map very quickly, but we could also assemble many maps with different build parameters within a reasonable amount of time. Distributing the code over the workstations in the lab allowed us to gain the expertise to develop similar approaches to other computing problems in bioinformatics. As CPUs very quickly became cheaper and faster, the 500–600MHz workstations no longer represented the fastest white-box computing platforms. In addition, ensuring the constant availability of workstations was a challenge – the owners of the boxes often stressed the machine's capacity, at times to the point of requiring a reboot. Interactive user sessions were being affected by distributed tasks, slowing personnel efficiency.

It became clear that the Centre would greatly benefit from a central source of CPU power.

LINKS:
Kryotech:
http://www.kryotech.com/index.html
FPC:
http://www.sanger.ac.uk/Software/fpc/faq.shtml
Human Genome Project:
http://www.ornl.gov/TechResources/Human_Genome/home.html
Human map build:
http://carbon.wi.mit.edu:8000/cgi-bin/contig/phys_map
Mouse map build:
http://www.bcgsc.bc.ca/projects/mouse_mapping
Genomic databases:
http://ihg.gsf.de/ihg/databases.html
SRS:
http://srs.ebi.ac.uk
SGI bioinformatics cluster resource:
http://www.sgi.com/solutions/sciences/chembio/linux_clusters.html
YAC bioinformatics cluster:
http://bioinfo.mshri.on.ca/yac/
The Collective cluster:
http://www.cs.uidaho.edu/~beowulf/

Cluster Construction

Initial Considerations

As soon as we knew that we needed a cluster, typical issues presented themselves: to build or to buy? Gigabit connectivity to the nodes or not? Do we buy the fastest CPU available or hunt for the best price-performance, the so-called price sweet-spot.

The cluster had to be made flexible to address all types of computing problems: those included CPU, memory, and I/O bound. We could not list all the problems that the cluster was going to help us solve. Ideally, answers we obtained with the cluster would give rise to new and interesting problems.

The Centre was awarded a grant from CFI to construct the cluster. Our budget was large enough to entertain sourcing the hardware and construction to vendors such as VA or IBM, both supporting Linux. After communicating with these companies, it became clear that we could build our own hardware significantly cheaper. Some of the management features of the VA and IBM solutions appeared very attractive, although it was not clear exactly whether this additional monitoring and management hardware provided good value.

LINKS:

IBM high-availability clusters:
http://www1.ibm.com/servers/eserver/iseries/ha/
VA clusters:
http://www-unix.mcs.anl.gov/chiba/
CFI:
http://www.innovation.ca/index.cfm

LINKS:

IBM 4000R:
http://www.pc.ibm.com/us/netfinity/4000r.html
IBM x330:
http://www.pc.ibm.com/us/eserver/xseries/x330.html
VA 1221:
http://www.valinux.com/systems/productinfo.html?product=1221
Rack mount equipment RC0101 1U case:
http://www.rackmountequipment.com/products/rackchassis/RC0101/RC0101.htm
Rack mount RM0120 1U case:
http://www.rackmount.com/Rackmt/RM-01-20TEST7.htm

Broadly, the questions that faced us fell into these categories:

1. Profile of nodes (1U, 2U, or ATX/micro)
2. Node components (CPU, memory, storage, power supply)
3. Cluster networking and topology
4. Electrical and UPS
5. Node management
6. Cooling
7. Vendor-bought or assembled

Profile of Nodes

Being heavily restricted by space, the 1U node solution was the obvious one. At the time, good quality dual CPU 1U cases were becoming available. Internal cooling was a very important factor – the 1U server market was just now opening and it seemed that each vendor had a unique solution. At the time, in the first quarter of 2000, even VA Linux proposed that clusters be built with 2U nodes out of cooling considerations. Later, they designed a small case and handled the cooling effectively and are now proponents of building clusters using 1U nodes exclusively.

If the use of the cluster helped us to solve complicated computational challenges, additional funding for more nodes could be expected. The 1U solution was attractive because it allowed us to expand the cluster with minimal use of additional space. With our space restrictions, it was important to minimize empty space within a node as much as possible, without sacrificing cooling or the quality of the power supply of the case.

Node Components

STORAGE

To keep the cluster flexible, we wanted each node to have its own local storage space. Installs, and possibly booting, would be done over the network, but each node would house its own copy of the required system files and libraries. Anticipating at least 512MB in each node, the memory overhead in caching these files would be insignificant. Local storage would also remove the stress on the cluster NFS server placed by nodes having to retrieve any information from a centralized disk. Local storage would also allow nodes to write large temporary files without causing an NFS bottleneck. Finally, if some jobs required very large input files, these files could be copied to each of the nodes ahead of time. SCSI was felt to be too expensive, especially for large, fast disks, and therefore, we decided at the time to stay with the EIDE interface. We could easily give each node 20 or 30GB of local storage at a fraction of the price of SCSI disks. With the cost savings, we could always purchase additional disks for the cluster RAID stack.

MEMORY

Many of the bioinformatics tasks require large amounts of memory, upwards of 1GB or more. We decided that each node would therefore have 1GB of fast memory, PC133 ECC Registered. If the motherboard was chosen to hold up to 4GB, we could always add memory later. Deciding to skip ahead from 512MB to 1GB for the nodes meant that the cost would be higher, but we felt that we could address a broader range of problems this way.

POWER SUPPLY

The power supply was a significant concern. In the management of about 50 computers in the Centre, power supply faults are probably the most common hardware failure.

Low-quality power supplies lose capacitance over time and put undue stress on node components, decreasing their life span. As much as internal node cooling, choosing a case with a powerful power supply was important. If we were to go to a vendor to purchase the cluster, the power supply choice would be made for us.

CPU

The last and important key to node performance is the CPU. We quickly discovered that measuring the value of CPU speed in cluster nodes did not necessarily follow the same rules as doing so for a single workstation. For example, at the time of writing this article, local CPU prices were about C$200 for 700MHz Pentium III and C$360 for 1GHz Pentium III. The latter price per MHz is 1.3 times higher. When we were building the specification list for the cluster, however, the difference between the fastest processor and the best-valued processor was much larger – typically five times or more.

When building a single workstation, one could argue that paying a premium for the fastest processor does not equate to good price-performance. This simple consideration motivates most workstation purchasers, who typically like to buy the second or third-fastest CPU. However, once one factors in the price of the memory and other components in the system, the CPU represents only a small fraction of the total cost of the computer. Consequently, the relative price difference between, for example, a 700MHz or 1GHz processor is small. At this point one might think that when already spending many times the cost of a single processor on a workstation, the cost of an additional half-processor can be justified.

This can be illustrated by the following example. Suppose that the cost of the node without the CPUs is $4000. Let the price of 700MHz CPUs be $200. Using the 700MHz CPUs, with a budget of $100,000 for nodes alone, one can purchase 23 nodes. The speed of such a cluster would be 32Gflops. Going to the 1GHz CPUs at $360 each, we can afford 21 nodes but the cluster speed is 42Gflops. We have purchased two fewer nodes and the cost of the node components was transferred into the additional CPU cost. Interestingly, the GHz CPUs can cost as much as $1140 and we would still be well advised to use them in the cluster. In this extreme case, we could afford only 16 nodes but the speed would be 32Gflops, just like for the "value-priced" 700MHz-based cluster. The benefit of this scenario is that there are now seven fewer nodes to maintain and support with UPS and networking components. Thus, while the node costs are the same, we can probably save money on the supporting hardware.

One could argue that by buying fewer nodes, you lose proportionately more cluster cycles when a node goes down. This may be important in environments where node availability is low and for highly parallelized, robust tasks which can suffer a node loss without a break in the computation.

For us, taking advantage of the fastest CPU meant that we could use our space more effectively. Specifically, we would not have to purchase as many UPS units, saving room in the rack for more nodes.

These ideas are illustrated in the figures. In Figure 1 we let ks be the ratio of speeds of the most expensive CPU to the best-value CPU. For example, if the most expensive CPU is 1GHz and the best-value CPU is 700MHz, ks = 1.4. With a fixed budget we buy as many of the best-value nodes as possible in one case and as many nodes with the most expensive CPU in another. Let the ratio of the speeds of these clusters be S. The figure uses the arbitrary value of $200 for the cost of the best-value CPU. Notice that when S = 1 the speeds of

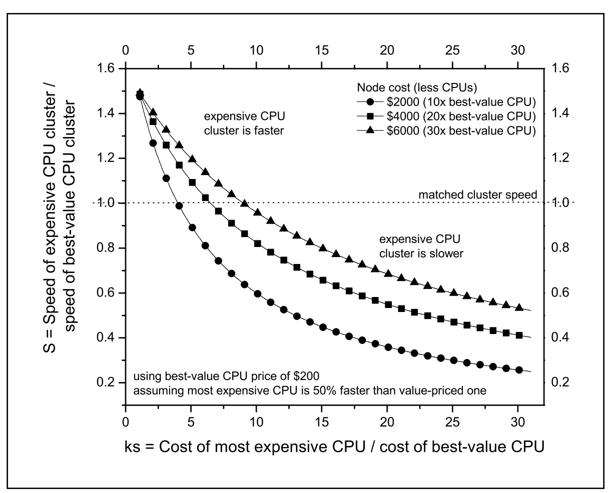

Figure 1

LINKS:
PC magazine price-performance Java viewer:
http://www.zdnet.com/pcmag/features/cpu/

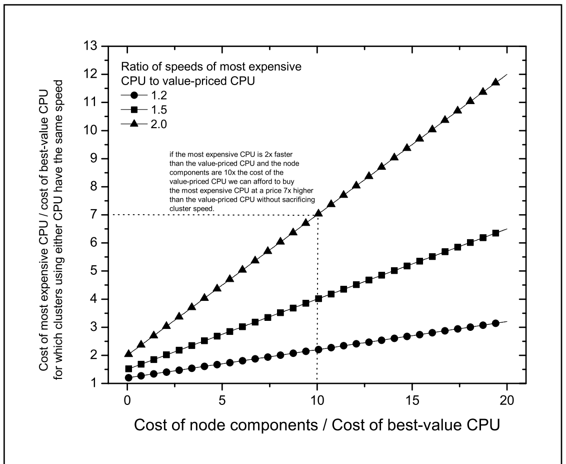

Figure 2

LINKS:
Myrinet:
http://www.myri.com/myrinet/performance/index.html

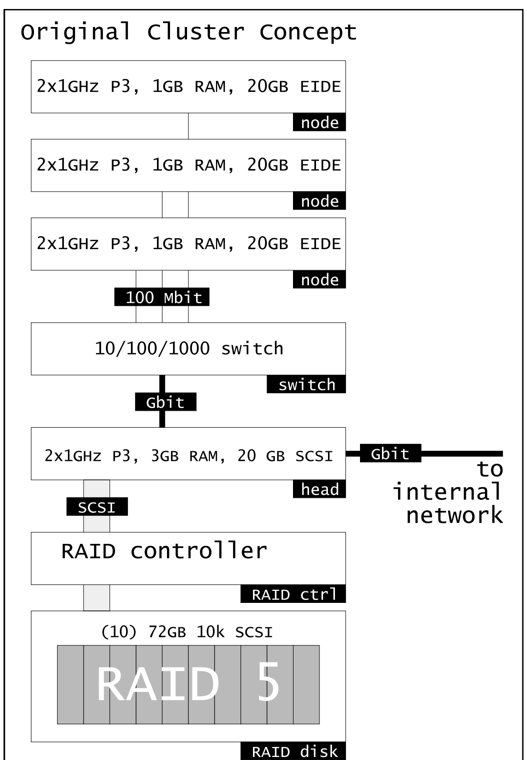

Figure 3

the two types of cluster configurations are the same. In other words, we can afford fewer nodes, but their CPUs are faster and the overall speed is the same. If our node components (memory, case, hard drive, etc.) cost 20x the price of the value-priced CPU ($4000) then we can afford to pay as much as 6.5 times more for the most expensive CPU, assumed to be only 50% faster.

Figure 2 shows the relative price of the fastest CPU vs. relative node component cost at which the two clusters have the same speed.

Even if one cannot afford as many nodes as would make the fast-CPU cluster have the same overall speed, the fact that each node is faster can be of significant advantage, particularly when the cluster is never fully utilized. These calculations were much more meaningful in the early parts of 2000, when the cost ratio between CPUs was much larger than their speed ratio. During May 2000, a 650MHz CPU was $130 and a 866MHz CPU was $1200.

Cluster Networking and Topology

Topology

Our internal network is a private class C subnet. To avoid filling the IP space, the cluster will be relegated to its own class C subnet. The nodes will be switched within the subnet. The main cluster management node will serve as an NFS server as well as a router for the nodes, connecting the cluster switch and the Centre's subnet switches using gigabit Ethernet. The topology is shown in Figure 3.

To keep the cost down and the cluster flexible, we decided to go with an Ethernet interconnect. Vendors such as IBM use Myrinet technology to connect nodes. This fabric provides lower latency than TCP/IP. Anticipating that we will not have many applications which will require numerous small messages to be passed between nodes, the cost of this technology, in our case, was better transferred into additional CPUs.

Networking

The choice of the switch for the cluster was motivated by our prior experience with managing a stack of 3Com Superstack II 3300 units which we use for the Centre's internal network. The drawback of these units is the four-unit limit on a logically managed stack. Furthermore, the stack is controlled by a central management unit whose failure brings down the entire stack. More sophisticated switches are circularly daisy chained in a stack so the failure of any one switch does not affect another. Secondly, the Superstack II units only have a single expansion slot. The stack master requires a matrix module to control the stack, and therefore one is left with only three expansion slots per stack. Using these for gigabit expansion modules, the maximum number of gigabit ports is three ports per 96 100Mb ports.

We wanted a switch which would stack more flexibly and contain more expansion modules. Not anticipating very heavy network loads, a switch with a full wire-speed backbone is not necessary, although the requirement for multiple gigabit connections to a single switch, to support multiple centralized servers and reduce the network bottleneck at these machines, exists. A chassis switch, which has multiple slots for various modules, is too expensive and bulky for our needs. However, the HP Procurve 4000M switch is a well-priced, 5U switch with 10 expansion slots which can be filled by a number of modules. It can receive up to 10x8 Mb ports, making it a very dense 80-port 100Mb switch. It can also receive multiple gigabit expansion modules, making the network topology flexible. We can always add additional gigabit-connected servers without the worry of running out of expansion slots on the switch. This particular unit is a good solution when the traffic on the network does not saturate the backbone of the switch and when there is a need for multiple gigabit connections.

LINKS:

HP Procurve 4000M:

http://www.hp.com/rnd/products/switches/switch4000/summary.htm

3Com Superstack II 3300:

http://www.3com.com/products/en_US/detail.jsp?tab=features&pathtype=purchase&sku=3C16981A-US

ELECTRICAL SYSTEMS AND UPS

Powering the cluster using UPS units was probably one of the most challenging aspects of constructing the specs. We cannot afford the space for a centralized, hard-wired UPS solution. All UPS units beyond 5000 VA must be hardwired. Multiple UPS units would have to be bought. Supplying power drops in the network room was a project for the building electricians. At the time, we had only four 15A/120V circuits in the network room.

When considering non-hardwired UPS units to power a large number of computers, the main requirement is that the number of UPS units be kept small and as much of the available electrical power as possible be transferred to the nodes.

For example, APC's 5000RM UPS requires a 30A/208V circuit. While the circuit can deliver 6000VA of power, the UPS only uses 5000VA. Furthermore, to deliver power to the cluster nodes, which require 15A/120V input, a transformer would have to be purchased to step-down the UPS voltage. The transformer, however, can deliver only 4500VA through its NEMA 5-15 outlets. This particular UPS and transformer together take up 7U in the rack. This is a comparatively bulky solution and does not optimally deliver all power coming into the UPS to the nodes.

Our requirements to deliver as large a fraction as possible of the incoming power into the nodes is due to the management of generator power in the building. All plugs for computers have generator backups, and it is important to maximize the use of all circuits to facilitate the plant operations staff in load balancing and expansion planning. The Centre has its own distribution panel within the building. The number of drops on the distribution panel is limited. Considering that a 208V circuit requires two drops and a 120V circuit requires a single drop, we could more efficiently use the drops if a 120V UPS unit could be found. Ideally, the Tripplite SMART3000RM2U is a 2U unit which uses a 30A/120V circuit. It can deliver 3000VA of power to the workstations and has nine NEMA 5–15 outlets. Tripplite is currently unique in being able to provide a slim, powerful UPS system designed specifically to power 15A/120V components. We anticipate that other vendors will follow in marketing similar products.

We estimate the average load of our nodes to be 130W (220VA). Each UPS could therefore support about 9–10 nodes at 70% load. Ideally, no power strips or transformers would have to be purchased. Looking ahead, in a 42U rack in which 5U are used for the switch, 2U for the head node and 3U for a disk tray, we have room for 32U of UPS/nodes. Given that a 2U UPS can support 10 1U nodes, the effective node width is 1.2U and the rack capacity is 26 nodes with three 2U UPS units. Using the APC UPS, for example, would require 2 UPS units which would take 14U, leaving only 18U for the nodes!

As most vendors do, Tripplite offers an environmental sensor which can be used to monitor the temperature in the room – a very useful backup thermometer for our enclosed space. Tripplite offers Linux drives and is considered one of the more Linux-friendly vendors.

The nodes were placed in an HP Rack System/e. This rack has the advantage of an available vertical extension kit which expands the rack from 41U to 49U. This rack system is

LINKS:

Tripplite SMART3000RM2U:
http://www.tripplite.com/products/family/ups/smart_pro/2u/index.cfm

APC 5000 RM:
http://www.apc.com/resource/include/techspec_index.cfm?base_sku=SU5000RMT5U&language=en&LOCAL.APCCountryCode=US

Powerware 9125:
http://www.powerware.com/Products/9125/product.asp

HP rack systems:
http://www.hp.com/racksolutions/p_erack.html

Great Lakes rack equipment:
http://www.greatcabinets.com/Standard_Cabinets/gl720-32.htm

extremely sturdy, supporting 2000-pound static loads, and is also available in a very minimalist style. Panels and doors may be added. A fan tray of nine 75 cfm fans was bought to facilitate air circulation through the rack. Our cooling requirements call for a high air flow in the room, probably making the fan tray unnecessary.

NODE MANAGEMENT

One could say that any number of networked computers is a cluster. With added management tools and software, a group of computers can be managed and used as a single entity. Many of the vendor solutions package some cluster-management software.

Cluster management can be broadly divided into two areas: system maintenance and administration; job queueing and monitoring. Often two separate packages or sets of programs would be used for these distinct tasks. Chances are, regardless of what the purpose of the cluster is, given an operating system there will be some cluster administration software available for it. There are a number of open source programs that will mirror, maintain, and logically administer multiple machines.

Because our cluster is a research cluster, we expect that its use will range from job distribution by simple constructs like `rsh` to more complicated job queueing using dedicated packages. Very likely, a number of in-house parallel applications will be developed using PVM- or MPI-style libraries. Currently, we see the cluster job management challenge as an ongoing effort to logically address the nodes in a transparent and efficient fashion.

It will be up to the system-administration scripts and applications to ensure node homogeneity and availability and to keep monitoring cluster statistics and utilization. One option is to boot from a floppy. During this process, nodes would be instructed to fetch specific system files and libraries from the head node. Multiple floppies could be used to load various node configurations. Once the nodes are appropriately configured, it will be important that they form a coherent collective which is not affected by the loss of one or more nodes. Both KickStart, which comes with RedHat distributions, or SystemImager can do this well.

LINKS:

Cluster management software:
http://www.npac.syr.edu/techreports/hypertext/sccs-748/cms-table.html; http://www.nhse.org/NHSEreview/CMS/index.html

Beowulf:
http://www.beowulf.org/software/software.html

Systemimager:
http://www.systemimager.org

TORC:
http://www.epm.ornl.gov/torc

One of our initial attempts to create some low-level structure in the cluster is through the use of hostnames. Each node is named XofY where X,Y are integers in the range 0–9. We split the cluster up into groups of 10 nodes (e.g., 0of0, 1of0 ... 9of0). This simple partitioning, in which hostnames depend on incremental group index and node index within the group, allows users to address nodes in an automated way through their scripts — they can automatically construct hostnames rather than remembering IP ranges. We expect that while we are deploying and testing higher-level management packages, users will be able to take advantage of this. Developing tools for user management will be an ongoing and evolving effort.

COOLING

By our estimation of a load of 130W per node, and using a general load of 130W per 1U of rack equipment, a 42U rack will produce about 19,000Btu/hr of heat. It is important to provide fully redundant cooling for a space as small as ours. In the case when cooling fails, a small room such as ours will significantly heat up, leading to damage to the equipment.

We chose to contract the cooling out to an engineering consulting firm to assist us in choosing the correct solution. This work is ongoing. Some of the considerations were as follows:

First, it is important that the temperature be as constant as possible. Some cooling systems cycle (turn on and off) in an effort to keep the temperature constant. When the load is much smaller than the capacity of the system (which can typically cool at a fixed rate) the system can cycle frequently, leading to increased wear. Some units have a minimum cycle delay, which can cause the temperature to rise significantly between cooling cycles. Consequently, choosing a cooling system which can adapt to varying loads is crucial. A typical computer room fills up with heat generating equipment slowly over time but cooling is purchased at the outset. It is wise to purchase sufficient cooling right away to handle the anticipated future loads. For example, our small room can at most house three computer racks. Using the numbers above and adding in 3 KW of power drawn by switches and other equipment, we find that the entire room will never require more than 66,000 Btu/h of cooling, assuming that 130W per 1U can be used.

Choosing a cooling system which can adapt to varying loads is crucial

Generally the controls of the cooling unit can be adjusted to match the requirements of the environment. Ideally, one should monitor the temperature and humidity in the room with an external, possibly redundant, probe. Units can be bought which attach to UPS devices, or to serial ports of nodes. Linux kernels can have LMSensors compiled into them which gives access to motherboard and CPU temperatures for some motherboards.

Secondly, the volume of air flow in the room should be considered. If the room is very small, the air circulation can make working in the room uncomfortable. Our cooling requirements call for some 4000 cfm of air flow. For a 100 sq. ft. space this makes for a breezy room.

Finally, it is important to plan the air flow to facilitate the existing cooling dynamics of the nodes. The cases will generally draw air from the front and exhaust out the back. Therefore, cool air should be delivered in front of the racks, and warm air should exhaust from behind the racks.

Monitoring the temperature of each node in the cluster can provide not only valuable environmental information but also give some measure of how temperature varies from node to node in a rack. Possibly, if a node's temperature reached some critical limit an automated script could shut down the node or send an email alert.

VENDOR-BOUGHT OR ASSEMBLED

Our original concept was to build our cluster using the rackmountequipment.com RC0101 cases and locally purchased components. We chose the Tyan Thunder LE server board for its compatibility with LMSensors, the ability to house 4GB of RAM, and the two 64-bit slots. Integrated Ethernet is useful for us, especially considering that the networking loads will be low. Alternatively, to minimize packet loss and maximize throughput reliability of a 100Mb connection, a server-class network card such as the 3Com 980 TXM can be used.

Very soon after finalizing the cluster specifications, we were approached by IBM to partner with them in building a cluster. We were offered their x330 nodes at a significant educational discount, paralleling our own costs to construct the nodes in-house.

The x330 node embodies many of the qualities that we were seeking through various vendors. The case is ultra-cooled with six internal fans, giving redundant cooling if any one fan should fail.

The problem of connecting all the nodes to a monitor is addressed by daisy chaining keyboard, video, and mouse for as many as 42 nodes to a single keyboard, mouse, and

LINKS:
IBM x330 review:
http://techWeb.techreviews.com/sections/topics/article/TT20001114S0002/1
IBM ASM guide:
ftp://ftp.pc.ibm.com/pub/pccbbs/pc_servers/25p2574.pdf
Tyan Thunder LE:
http://www.tyan.com/products/html/thunderle_p.html 3com 980
TXM:
http://buydirect.3com.com/iom_dcms/b2c/catalog/detail.html?SKU=3C980C%2DTXM&SM=

monitor. This is done via a special console port which combines the three signals, requiring a single cable between the nodes. Each node has a selection button on the front which connects it to the input devices and monitor. We are already finding that this is an invaluable tool. Additionally, it saves the space of having to manage cables and KVM switches.

The built-in SCSI 160 and hot-swappable 18GB 10,000 RPM disks make for crisp I/O. The facility to add another disk means that the disks can be mirrored with RAID to enhance availability.

The power consumption of an x330 node is very close to 130W when CPUs, memory, and disk are fully utilized. The draw spikes to about 160W during booting.

We have received nearly all of the parts for our cluster and are in the process of building and testing the components. We expect that the final construction will be complete in the next month or so.