Architecting, Building & Managing Production Bioclusters

February 4, 2003
O'Reilly Bioinformatics
Technology Conference

The BioTeam, www.bioteam.net

The BioTeam™, Inc.

- Independent, objective Bio-IT consulting
- Vendor & technology agnostic
- Scientists / IT professionals
- “Lifestyle entrepreneurs”
- Bridge the gap between Science and IT
- Partners
  - Michael Athanas
  - Chris Dagdigian
  - William Van Etten
  - Stan Gloss

Disclaimer(s)

- One downside of being a generalist is never being an expert in anything.
- Biologists be warned: This talk will mostly be about IT and engineering tips
- We are really going to be talking abut "compute farms" – not clusters

If this sounds familiar…

- Similar talk given at 2002 O'Reilly Bioinformatics Technology Conference
- Revised & expanded for this meeting

Lots of material to cover

- Presentation will be fast paced
- I may skip or gloss over some slides
- Please ask questions!
- Full presentation will be downloadable on the O'Reilly conference site
  - Or just give me your email address

Extra stuff if time permits…

- Clustering with Apple Xserve and OS X
- Grid Computing - hype vs. reality
- Blades, blades, blades
- Advanced cluster applications
  - Systems Biology @ Beyond Genomics
  - GridEngine + PISE portal at http://xblast.tamu.edu
- Sun's new LX50 Linux compute node
- How Sun Grid Engine works

What's new…

- Clusters are mainstream & even boring now
  - Real fun is in the science, software & workflow
  - Hardware vendors still horrible at this
  - Software vendors still pushing their "total solutions" at expense of interoperability, cost and usefulness
- Interesting vendor moves
  - Dell, Sun, Apple, etc.
- Blades
  - Actually become usable
- Sun GridEngine gains momentum
- Grid Computing is (still) hype and vapor
- The bioclusters mailing list hits critical mass

Why this talk in 2003?

- Available references *still* provide info that is not appropriate for life science clusters
  - Tremendous bias towards DIY parallel/beowulf style computing
  - Too much focus on raw price/performance vs. operations and administrative burden

Clusters & Bioinformatics

Why?

Simple.

It's all about power and money.

Clusters & Bioinformatics

- The impressive price/performance ratios gained from clustering piles of commodity computer hardware are real enough to convince even the most conservative CTOs and CIOs
- Especially in tight financial times
- There are three main benefits (IMHO)

Clusters & Bioinformatics

1 Existing in-house informatics techniques can be applied on a scale previously unconsidered

2 Areas of scientific inquiry previously discarded as impossible or impractical are now feasible

Clusters & Bioinformatics

3 IT dollars are saved by slowing the upgrade cycle and extending useful lifespan of existing 'big iron'

- Clusters can soak up load from large servers -- freeing them for more specialized tasks
- Use your large memory / 64 bit / SMP systems only for applications that require them!
  - RDBMS, sequence assembly, data mining, novel science

What we *not* talking about

High Availability (HA) Clusters

Application Clusters

Beowulf Clusters

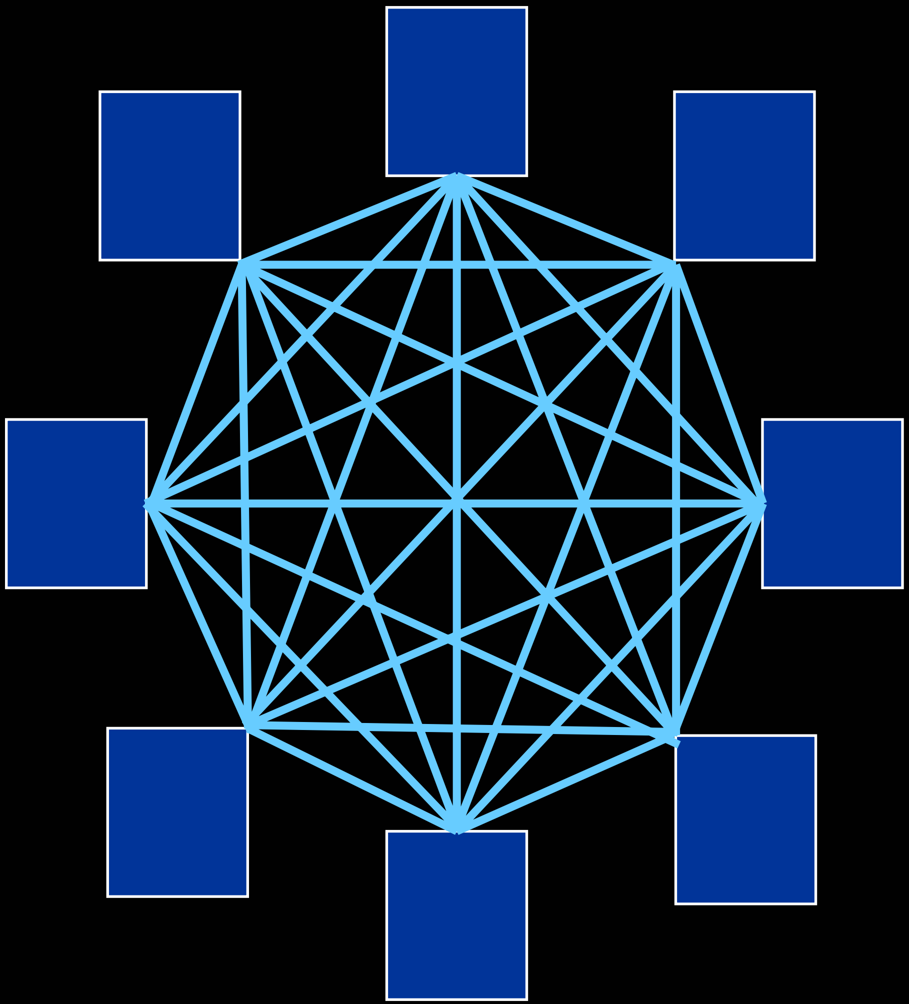
Huh? Why not beowulf clusters?

Simple really

Biologists don’t need them

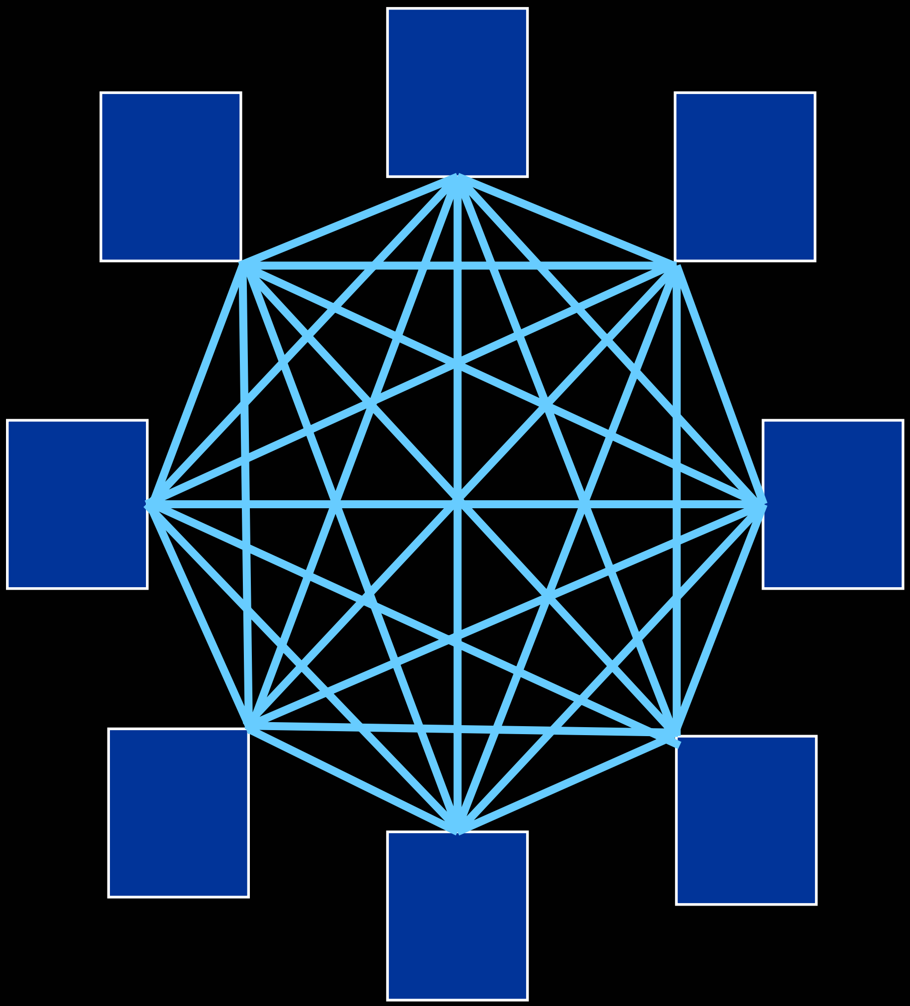
* *Some exceptions*

Beowulf systems



- Purpose built for parallel software execution
- Old topologies designed to avoid (expensive) ethernet switches
- New systems may use high speed / low latency interconnects for message passing

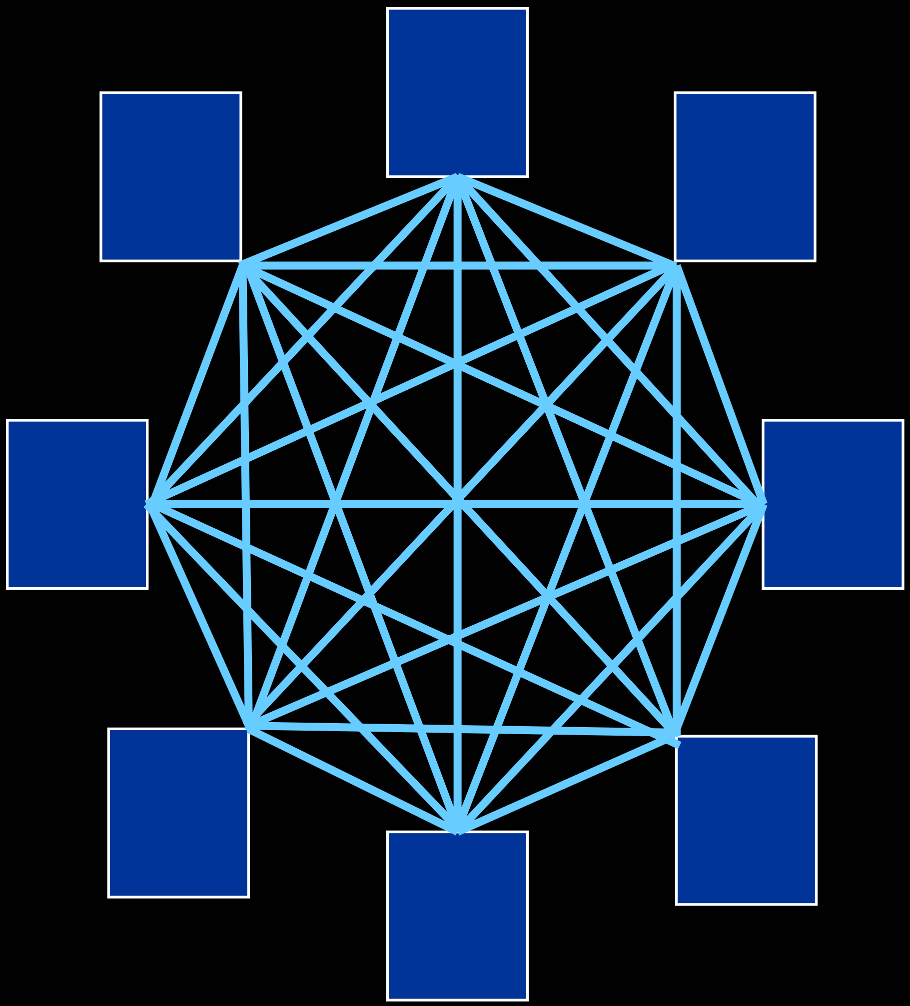
Biologists don't need beowulf



- Very little parallel code in life science informatics*
- Very few hardcore HPC hackers
- Lots of serial computing use cases

** Exceptions: modeling, chemistry & structural fields*

Why this matters

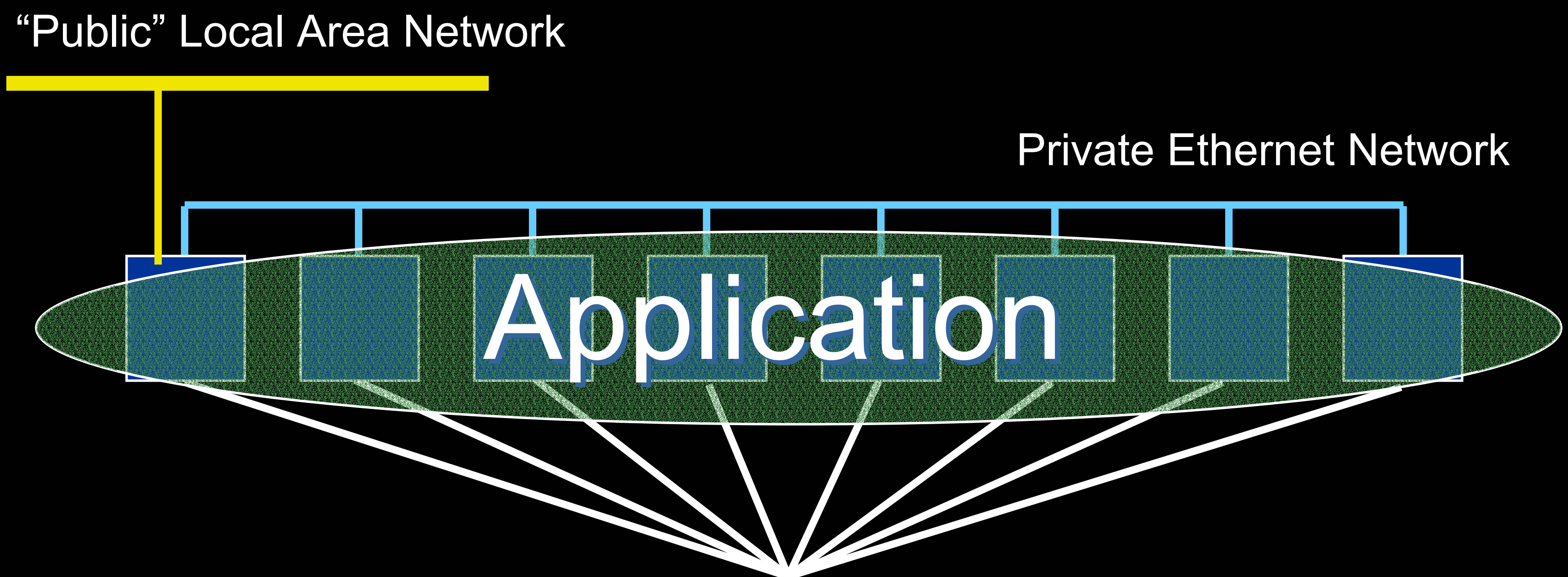


- A ‘biocluster’ benefits from different hardware, software, architecture and operational methodologies
- Following a ‘beowulf cookbook’ may not be optimal

Types of problems

Tightly Coupled	Embarrassingly Parallel
1 + 1 = X	1 + 1 = X
X + 2 = Y	2 + 2 = Y
Y + 3 = Z	3 + 3 = Z
N + Z = W	N + N = W

Tightly Coupled / Parallel



{Optional} High Speed/Low Latency Switching Fabric
Allows for message passing and shared memory operation

generic bioinformatics use case

Embarrassingly Parallel

1 + 1 = X

2 + 2 = Y

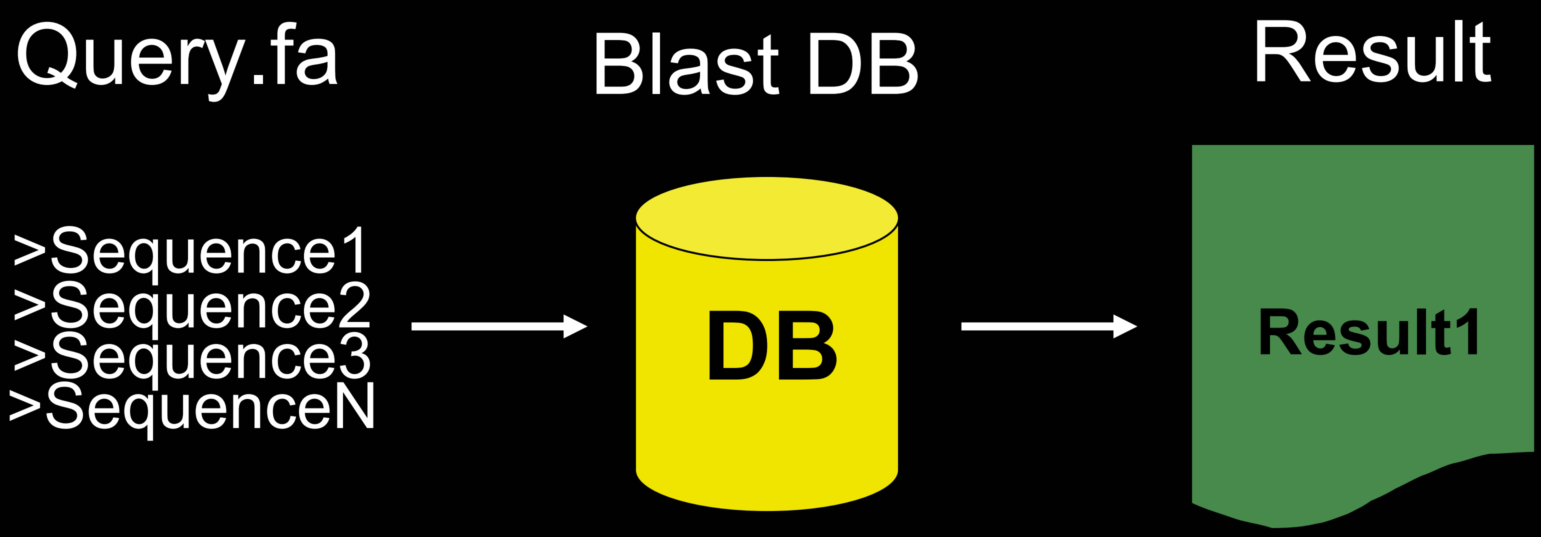
3 + 3 = Z

N + N = W

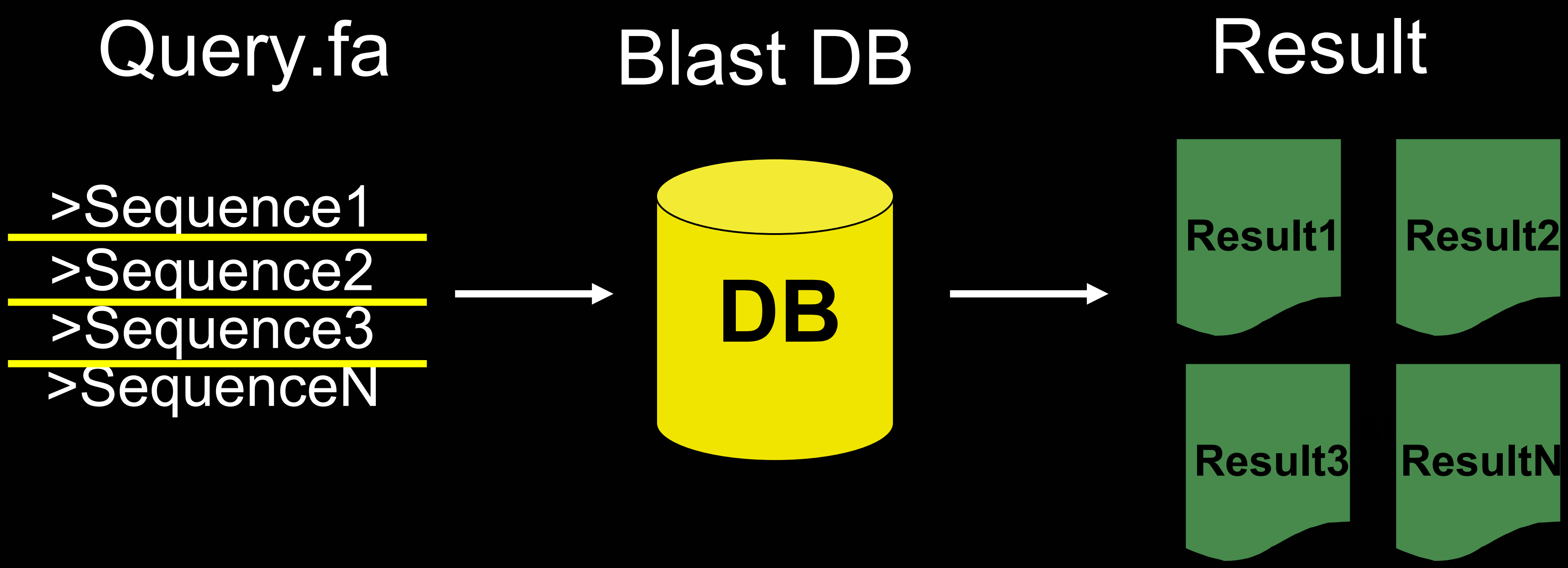
- Scientist has a problem
- She does not want or need to run ONE parallel application stretching across a 100 CPU cluster
- She wants to run one standalone program 100,000 or 1,000,000 times with slightly different input and output values

The BioTeam, www.bioteam.net

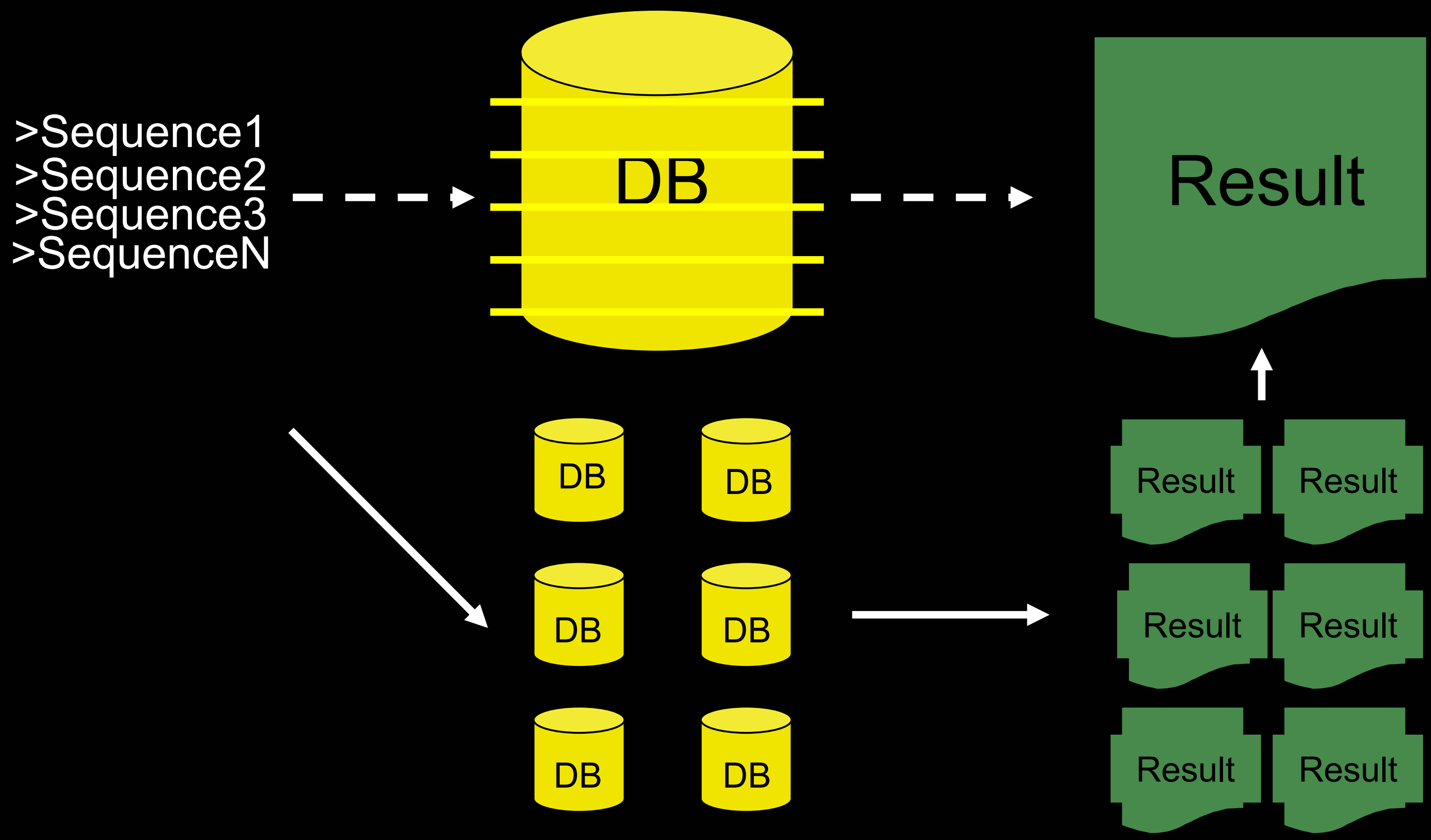
Obligatory BLAST use case



Split BLAST by query sequence

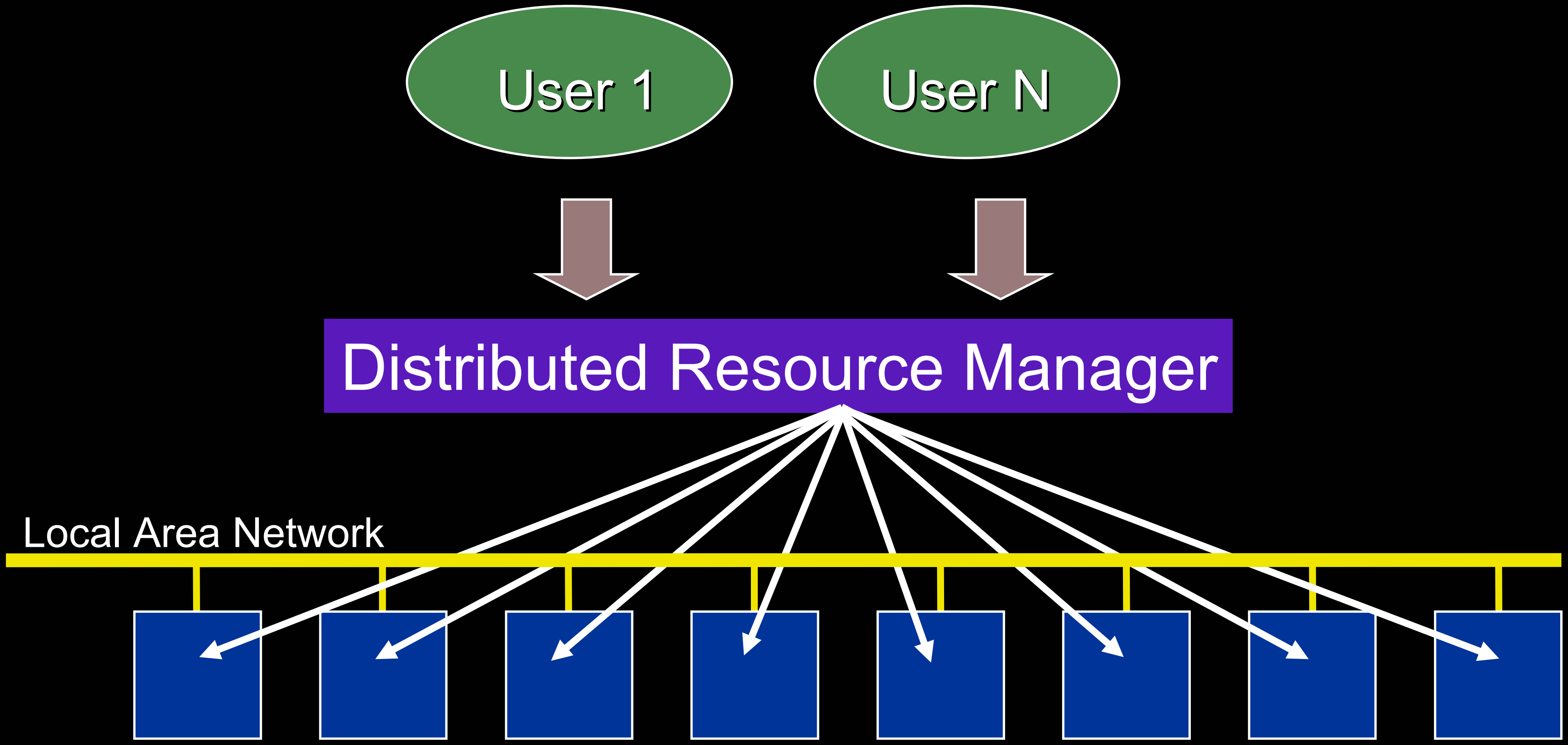


Split target databases & by query

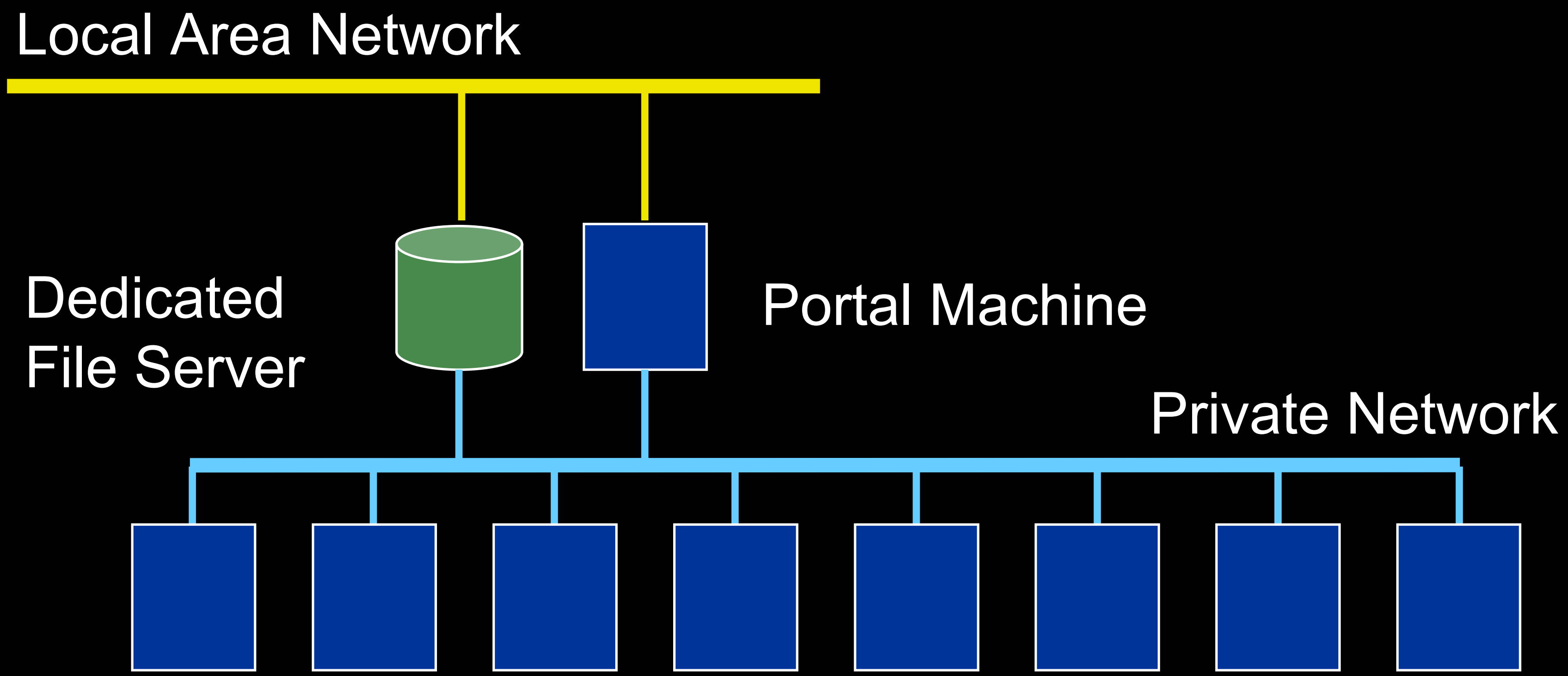


Embarrassingly parallel tasks lend themselves well to "compute farming"

Compute Farm Logical View



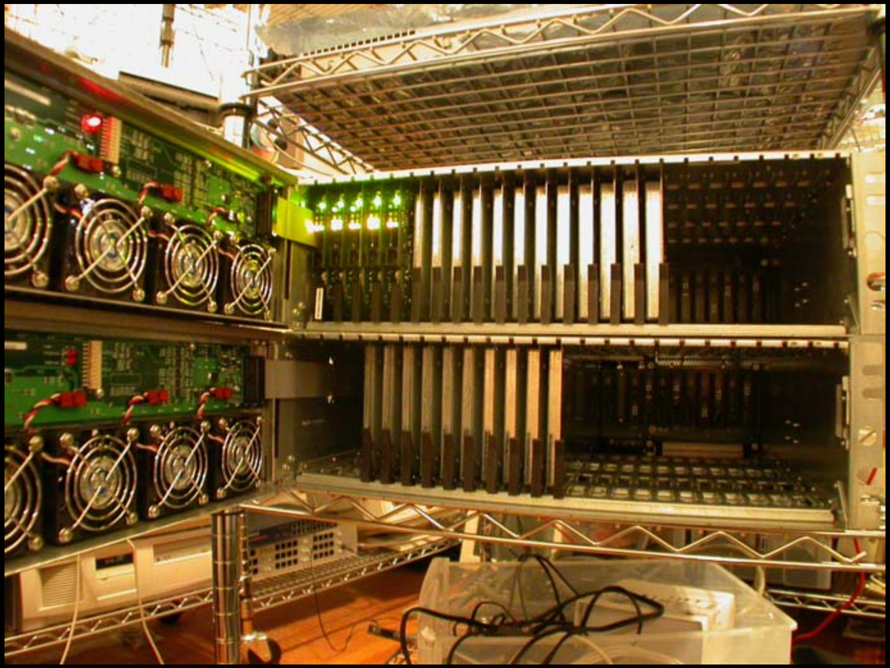
Compute Farm
'Portal' Architecture



Compute Farms in the real world

Dell system
@ Harvard
CGR

Compute Farms in the real world



RLX blades in a home office

The BioTeam, www.bioteam.net

Compute Farms in the real world

10 CPU AMD personal/pipeline cluster used at a biotech site

The BioTeam, www.bioteam.net

Compute Farms in the real world

The biggest Apple
Xserve cluster
in the world is
a biocluster!

Building “Production” BioClusters

Trends: 2002-2003

- Clusters have become core components of research computing infrastructures
- Replacing or augmenting development-scale single-user/single-algorithm systems managed by end users
- Supporting multiple competing users, groups and projects and workflows
- Officially sanctioned; managed by IT
  - For better or for worse!

Trends: 2002-2003

- Life science clustering may be old news but it is still amazingly easy to screw up
  - Lots of hardware vendors seeking ‘bio-IT’ dollars for ‘solutions’ developed for other markets. Buyer beware.
  - Very powerful hardware available at great pricing
  - Integration, optimization and operational issues sill exist

Essential to understand…

- Maximizing price/performance is secondary to other more important considerations

What can be more important ?

- **Total Administrative Burden**
  - Hardware, software and staff!

- Ease of scaling
- Flexibility & Usability
- Physical system footprint
- Heat, cooling and electrical concerns

BioCluster Design Goals

- Reliability, Availability & Security
  - Redundancy and abstraction of key components
- Flexible & Scalable
  - Support multiple competing users, groups, projects and workflows simultaneously
  - Add/remove/maintain nodes without visible downtime
- Manageable
  - Goal: Run 1,000 CPUs with one admin FTE
  - Business or Scientific priorities get reflected in resource allocation

Pre-Purchase Research

- User / Application Requirements
- Compilers, IDE’s & debuggers
- Physical & environmental constraints
- Initial size / future size
- Network & interconnects
- Storage
- Management
- Maintenance
- Architecture
- Distributed Resource Mangement (DRM)

User model & Use Cases

- Single User, Few, Many Users
- Groups of users
- Some more equal than others
- Batch/bulk vs. singleton jobs
- High throughput or fast turnaround
- Direct vs. Portal node access

Application Requirements

- Many short running processes
- Few long running processes
- High/Max RAM requirement
- CPU and/or IO bound
- Single/Multi-threaded
- MPI/PVM/Linda parallel aware
- * What will not run under Linux

Physical & Environmental Constraints

- Available Power
- Available Cooling
- Density (Blades/1U/2U/Tower)
- DIY Staging space
- Raised floor or ceiling drops
- Height & width surprises
- Fire code
- Organizational standards

This could be you…

The BioTeam, www.bioteam.net

Operation & Management: General

Reducing administrative burden via automation and intelligent design is critical to avoiding future nightmares.

- Hands-off OS installation and re-installation
- KVM or Crash Cart? *** IMPORTANT ***
- Serial console access?
- Cluster monitoring & event reporting
- Remote power control
- "LOM" or out of band management features
  - License costs for same…

Operation & Management: Linux
www.systemImager.org

"SystemImager is software that automates Linux installs, software distribution, and production deployment."

- True unattended disk partitioning and remote Linux installation
  - PXE network boot or local media
  - Can play nicely with Redhat Kickstart (even with PXE)
- Incremental updates of active systems
- Based on open standards and tools
  - RSYNC, SSH, DHCP, TFTP, PXE
- Totally free, open source
- Not without its issues

Operation & Management: OS X

- Carbon Copy Cloner & NetRestore
  - For full OS installations
  - Can boot and automagically provision an Apple Xserve via your trusty iPOD!
- ServerMonitor
- Apple Remote Desktop
- RADMIND
  - From University of ??
  - For incremental updates to running systems

Operation & Management: Cluster Console Access

- Required for debugging hardware and boot issues
- Large KVM or Serial Console systems may not be essential
- Universal console access may perpetuate “bad” operational habits
- Crash carts with VGA and keyboard cable extenders are cheap and very effective.

Console Access

Monitor key cluster servers with a 4 or 8 port KVM

Use inexpensive video and keyboard extension cables to reach systems that don't require constant console access

Operation & Management: Monitoring & Reporting

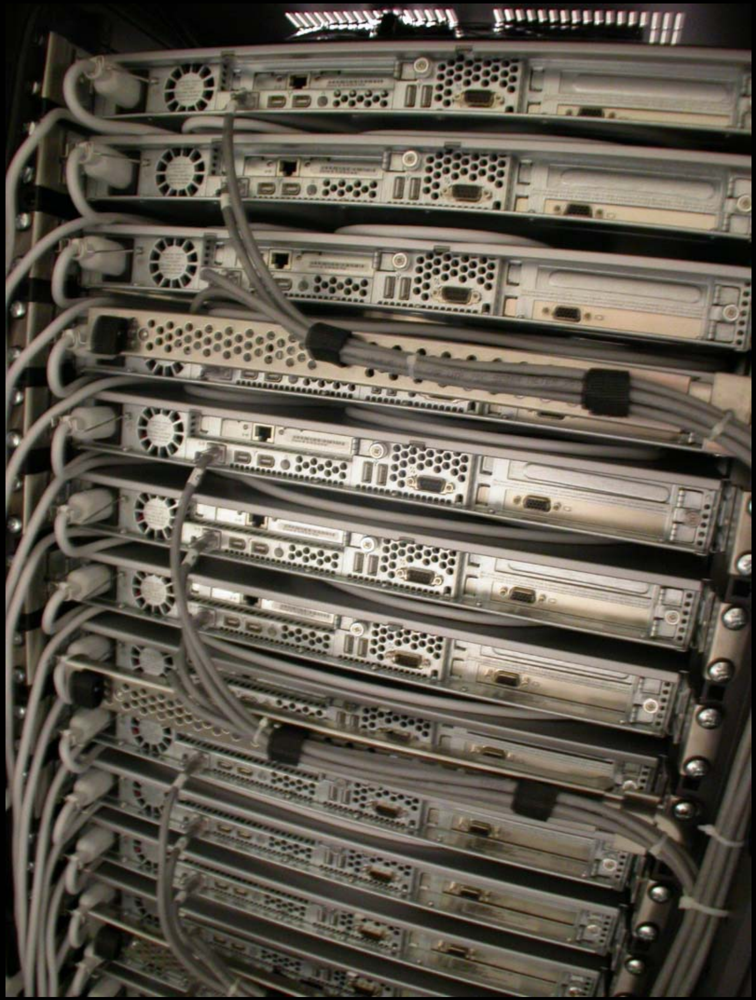
- Many high quality free tools
  - BigBrother, RRDTool, MRTG, Ganglia
  - Sar, Ntop, etc. etc.
- SNMP is pervasive now
  - Hook into your enterprise tools
- Keep system audit & log files around
  - Poor man's trending tool
    - System, daemon & accounting
    - DRM

Compilers

- Think about them
  - Often overlooked
  - GNU tools are great but consider commercial compiler options if you are:
    - A performance freak
    - Writing SMP or parallel apps
    - Doing serious scientific programming in C, C++ or Fortran

Network & Interconnects

- Need sufficient bandwidth for IO/interprocess communication
- High speed interconnect(s)
  - Do you have enough 66mhz PCI slots?
- How many networks do you need?
- Effect of interconnect topology on:
  - Scaling and future growth
  - Wire management
  - Access to external networks

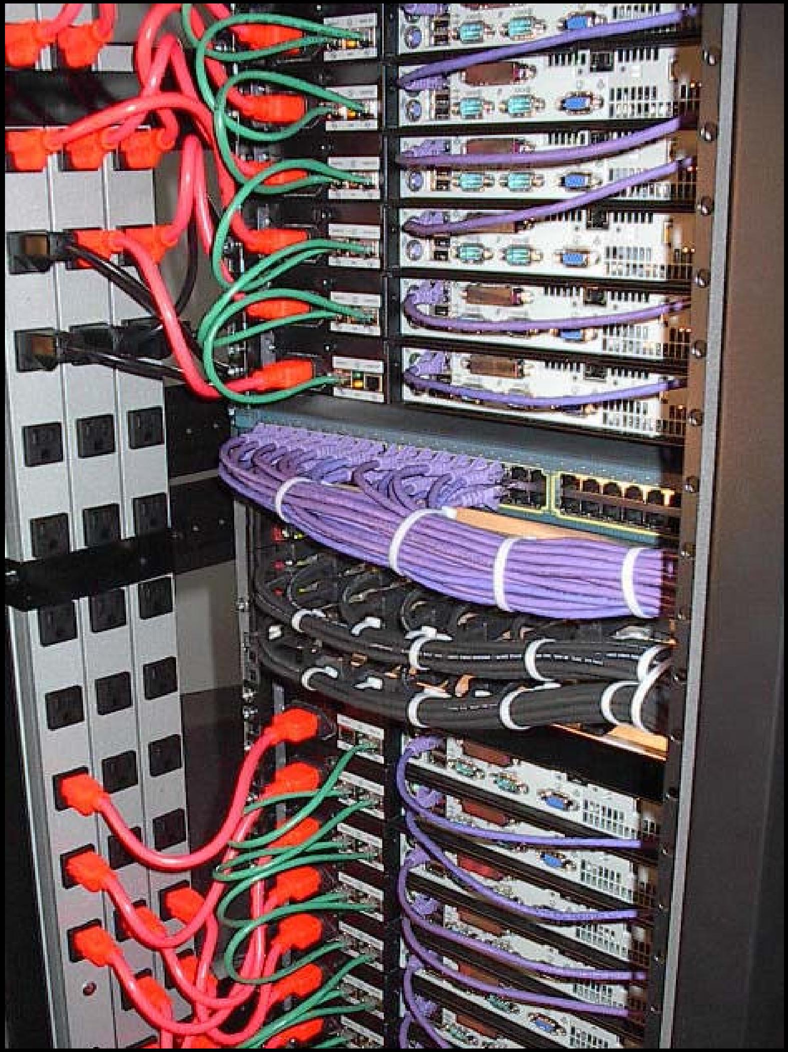


Network & Interconnects: High Speed Interconnects

- When low latency message passing is critical
  - Massively parallel applications
- Not generally needed in BioClusters (yet)
- Expensive
  - Can add 50% or more to cost of each server
- No magic, must be planned for
  - Applications, APIs, code, compilers, PCI slots, cable management & rack space
- Commonly seen products
  - Myrinet (www.myricom.com)
  - Dolphin SCI (www.dolphinics.com)

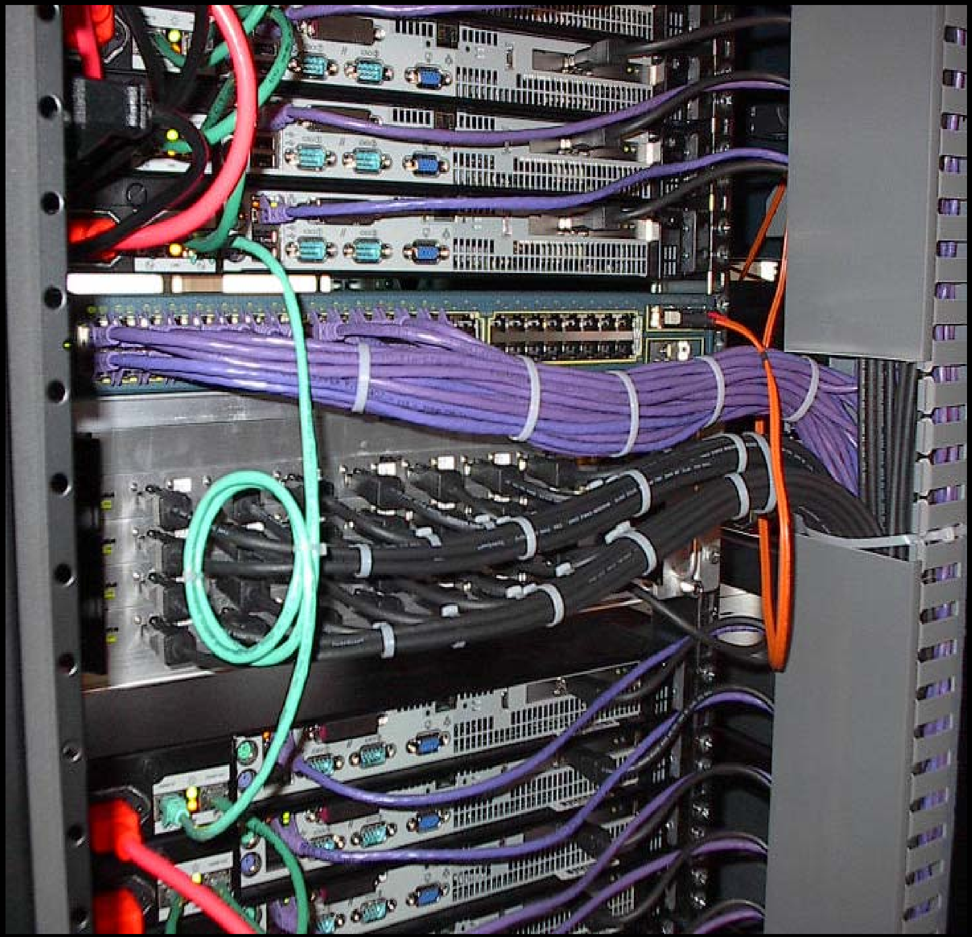
Network & Interconnects: Custom cable lengths

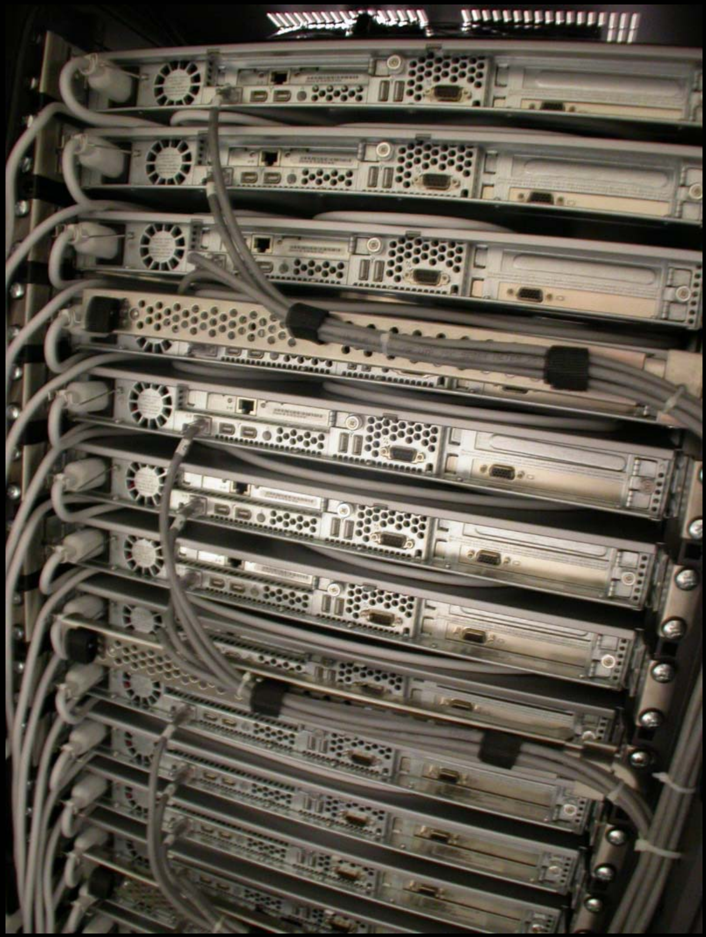
- Strongly recommended
- Modest cost; real gain
- Don't make your own
  - Savings offset by quality problems and testing overhead
- Significant advantages:
  - Cleaner install
  - Less wiring bulk
  - Color schemes
  - Aesthetics can be important (especially to management)



The BioTeam, www.bioteam.net

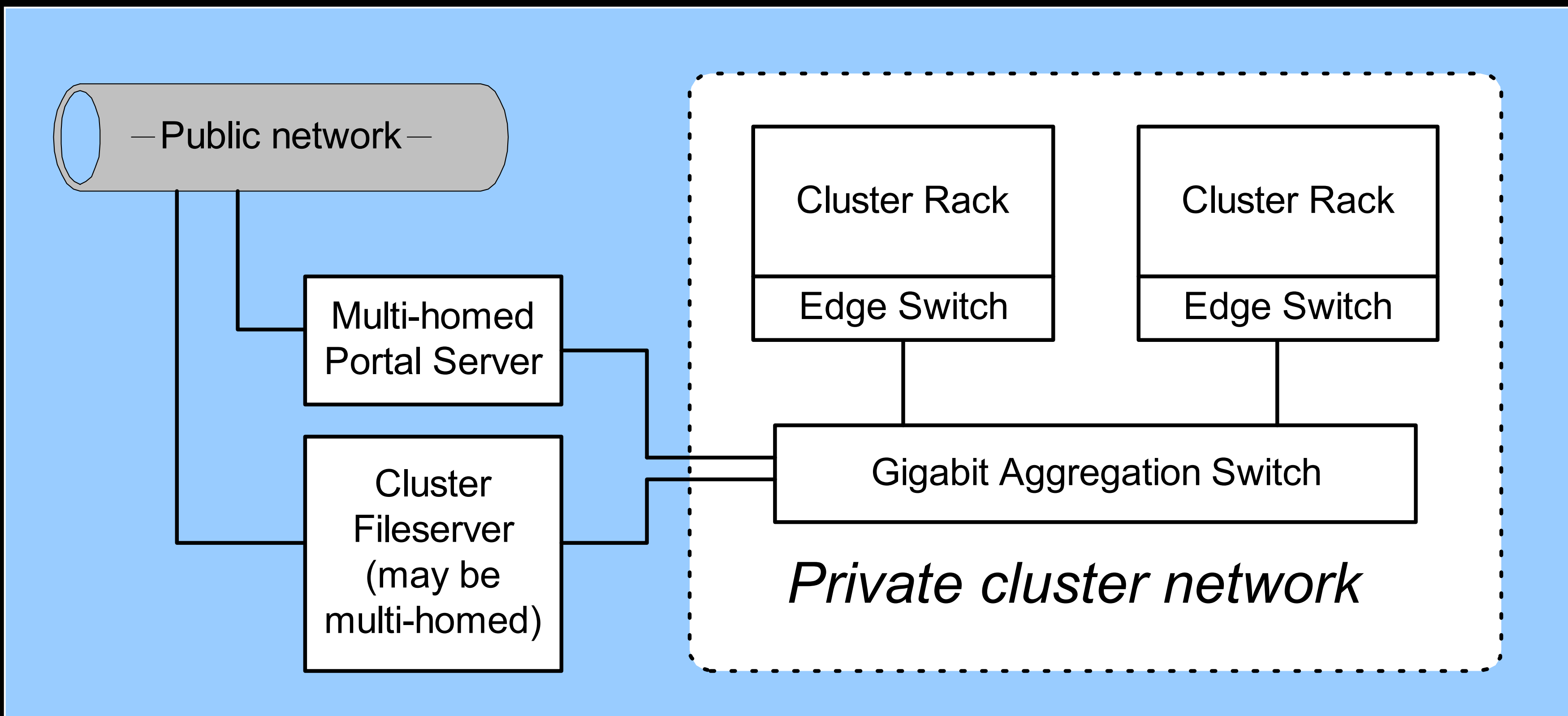
Wiring complexity:
Ignore FUD from high density server vendors…





The BioTeam, www.bioteam.net

Network & Interconnects: Edge switching and private subnets



Edge & aggregation network switching

- Cable management becomes reasonable
  - Minimal intra-rack cable runs
  - Power goes in; Fiber comes out
- Modular cluster architecture
  - Racks are self contained
  - Easy scaling by rack, not individual servers
- Logical place to direct connect core cluster elements
  - SAN Gateway / NAS Server
  - Cluster "Head" node(s), misc. servers

Network & Interconnects: Private Cluster Subnet

- Keep cluster traffic as local as possible
- Freedom to run customized network services
  - DHCP, DNS, NIS, NTP, TFTP, PXE
  - No chance of clobbering someone else
- Security
  - Cluster nodes not directly accessible from outside
  - Cluster nodes can reach external nets via NAT
- Key component in lowering the total cluster administrative burden

Storage

- Most BioClusters are I/O bound
- Not an area to pinch pennies
- NAS vs. SAN vs. Hybrid vs. local disk
- Picking the right RAID levels
- Heterogeneous storage
- Plan for data staging and caching

Storage: Know your bottlenecks

- Sequence analysis is I/O bound
  - Research will be rate limited by the speed of your disks and fileserver
- Modeling, chemistry & structure work
  - Generally CPU bound but lots of exceptions
- Some apps bound by speed of memory
  - Gaussian

Storage: Don't pinch pennies

- Slow storage means slow science
- Overloaded fileserver = useless cluster
  - Systems sit idle as CPUs wait for I/O
- Prepare for sticker shock
  - Commodity pricing has not reached the midrange and high-end storage markets
  - Backup is even worse

Storage: NAS vs. SAN vs. Local Disk

- SAN approach not appropriate; some exceptions
- NAS or Hybrid NAS/SAN is best
  - Overwhelming need for multiple clients with concurrent read/write access to the same file system or volume
  - Don't let vendors fool you into thinking SCSI disks are necessary in compute elements
  - Large, cheap IDE drives in compute elements allow for clever data staging & caching
    - Best way to avoid thrashing your fileserver

Storage: RAID levels

- Mirrored stripe sets are great if you can afford them
- RAID 5 or equiv is the best compromise
  - Excellent read performance; small performance hit on write operations
  - BioClusters tend to read far more data than they write
    - Example: Sequence similarity searching

Storage: Software RAID

- Test for yourself; Linux software RAID is amazing
  - Especially with cheap IDE drives on dedicated controllers
- Tests: A compute node with 2x 60 GB drives striped with software RAID0 testing a mySQL application
  - Faster than fibrechannel connection to Hitachi SAN
  - Faster than gigabit ethernet connection to a NetApp
  - YMMV of course; do your own testing

Storage: Heterogeneous storage approach

- Why store low value data on incredibly expensive disk arrays?
- Mix and match high and low end storage subsystems as needed
- IDE RAID based NAS appliances are great
  - $7-10K per terabyte
  - Great for scratch space and those nightly NCBI downloads and Genbank builds
- "Enterprise" IDE RAID chassis for $12K per terabyte
  - Hotswap everything
  - Fibrechannel or SCSI HBA's

Storage:
Plan for data staging to local disk

- A $250K NAS server can be brought to its knees by a few large BLAST searches
- Active clusters will tax even the fastest arrays. This is mostly unavoidable.
- Plan for data staging
  - Move data from fileserver to cheap local disk inside cluster compute elements
  - No magic; users and software developers need to explicitly do this in their workflows and pipelines

Storage: Areas for experimentation

- Intelligent use of local disk caches in a NAS environment
- software RAID techniques
- Ext3, Reiserfs and XFS filesystem tweaking
- Evaluate AFS or Coda in place of NFS
- Parallel filesystems
  - Are they stable? Can we use GigE instead of fibrechannel?

Storage: Data grids

- Forget compute grids; I need a data grid
  - Far more useful to me than a compute grid
- Need:
  - Single global namespace and URL type locater
  - Intelligent data caching, replication, migration, version control and invalidation
- Various people working on this
  - Avaki (with software)
  - Storage vendors (with hardware & software)

The Holy Grail of BioClusters: Data-centric process distribution

- Transparently handle
  - Data distribution
  - Data invalidation
- Data-centric job scheduling
  - Node with data in RAM
  - Node with data on disk
  - Peer Node
  - Network file server
- Observed: Linear scaling of I/O heavy workflows without thrashing networks or fileservers

Maintenance Philosophy:

- Prepare for high initial failure rates (regardless of vendor)
- Identify potential support contract and warrantee issues
- Some questions to consider:
  - Debug or wipe OS problems?
  - Diagnose hardware or swap out?
  - Replace by unit or components?
  - Internal staff or outsource?

Maintenance Recommendations

- Cluster compute elements must be
  - Anonymous
  - Interchangeable
  - Disposable
- Automation & remote management is key
  - Don't waste valuable admin resources messing around with individual servers and configuration settings
- 3 Possible states
  - Running / online
  - Faulted / Re-imaging
  - Failed / offline / marked for replacement whenever convenient

Distributed Resource Management (DRM)

- Batch queuing & scheduling
- Load balancing & remote job execution
- Resource allocation
- Detailed job accounting statistics
- Fine-grained user specifiable resources
- Suspend/resume/migrate jobs
- Tools for reporting Job/Host/Cluster status
- Job Arrays (*)
- Integration & control of parallel jobs

Distributed Resource Management: If you make the wrong choice…

- Best case scenario
  - Far too much time spent with installation, configuration and debugging
  - Significant administrative effort spent merely keeping things running day to day
  - You have to write scripts to “meta monitor” the state of your DRM and job queues

- Worst case scenario
  - Angry users; Angry boss
  - Crashes, disappearing jobs & data
  - Problems with high load or large numbers of jobs
  - Significant downtime

Distributed Resource Management: Commonly seen DRM suites

- Open source and/or freely licensed
  - OpenPBS
  - Sun Grid Engine
- Commercially available
  - PBS Pro
  - Platform LSF
  - Sun Grid Engine

Grid Engine

- Two versions are available
  - Free version
    - http://gridengine.sunsource.net
  - Commercial version
    - http://wwws.sun.com/software/gridware/sge.html
  - Each version has two flavors
    - Sun Grid Engine (SGE)
    - Sun Grid Engine Enterprise Edition (SGEE)

Different Versions of Grid Engine

- The free version of Grid Engine is identical to the commercial variant
  - Sun sales reps have been known to say otherwise. Don't believe them.
- Sun actually builds their commercial product off of the open source code repository
- The ONLY differences are:
  - Sun does extra quality control testing prior to release
  - Sun adds localization support for additional languages
  - Sun provides formal, enterprise level support

SGE vs. SGEE

- SGEE adds enhanced scheduling and policy features
- Additional policy based allocation features include
  - Share Tree
  - Fixed
  - Deadline
  - Override
- These features are powerful but complicated and generally SGEE is not needed for department level clusters
- The default FIFO and priority based scheduling that SGE provides are usually satisfactory for first time SGE adopters

Grid Engine

- Pro
  - Took life science cluster market by storm in 2003
  - Excellent product; improving rapidly
  - Serious competitor to Platform LSF
  - Will play nicely in future 'grid enabled' world
- Con
  - No API for cluster software developers
  - LSF still has the edge in administrative & operational burden
  - Docs are OK; Training is essentially nonexistent
  - Limited commercial support for architectures other than Solaris/Sparc and Linux/X86

Distributed Resource Management: OpenPBS / PBS Pro

- Portable Batch System; developed for NASA
- Open source version at www.openpbs.org
- Commercial version at www.pbspro.com

- Use PBS Pro, not OpenPBS
  - Non trivial differences between the two
  - You can get the source code
  - Reasonable pricing; free for most academics
- Purchase support and consider installation assistance from Veridian Systems

Distributed Resource Management: OpenPBS / PBS Pro

- Pro
  - Decent academic and commercial userbase
  - Solid professional services, support and training from Veridian Sysems
- Cons
  - No Job Array functionality
  - Higher administrative burden than Grid Engine or LSF
  - Minor but persistent complaints about OpenPBS
  - Most serious PBS sites have patched or otherwise significantly dug around in the internals of the system

Distributed Resource Management: Platform LSF

- Platform Computing, www.platform.com
- "Tight" integration with parallel environments
  - Including Myrinet topologies
- Used in very large life science settings
  - Celera, Sanger Center, Whitehead, many others
- Ensembl built with hooks for LSF
- Excellent support for simple workflows
  - Job arrays, Job arrays, Job arrays
- Official support for Mac OS X
- Not cheap
  - Pricing is getting better; especially for Linux
  - Academics can get special per-CPU subscription pricing

Distributed Resource Management: Platform LSF

- Pro
  - Rock solid stable & scalable -- the biggest clusters in the world run LSF
  - Commercial support, training & professional services available worldwide
  - Tech Support is very good from my experience
  - The only game in town for ISV support and commercial application integration
  - Easy to install, manage, configure and use
- Con
  - Expensive

Distributed Resource Management: What I said in 2002…

- At this time Platform LSF is still technically the best choice for serious production BioClusters
  - Lowest administrative/operational burden
  - Fault tolerance features are unmatched

- 2nd choice(s)
  - PBS Pro w/ support and professional services from the good folks at Veridian Systems
  - GridEngine rocks if you can support it internally

Distributed Resource Management: My $.02 for 2003

- My DRM of choice for new cluster projects is usually Grid Engine. Platform LSF is used for advanced systems and special situations

- For large production systems with complex requirements Platform LSF is still the best choice for serious production BioClusters
    - Lowest administrative/operational burden
    - Scaling & fault tolerance features are unmatched

- For small to midrange clusters or for groups that just need a rock solid basic DRM suite for their cluster
  - Sun Grid Engine is great for this
    - Open source version of SGE to be specific

Case Study 1

Harvard University
Bauer Center for Genomics Research

Bauer CGR @ Harvard

- New building opens April, 2002
- No IT director yet
- Technical computing infrastructure must be ready
  - Support researchers who have yet to be recruited
- Problems with the uncompleted server room
  - Too small
  - Bad network and electrical distribution
  - Not enough cooling
  - Construction deadlines mean major structural changes are impossible

Bauer CGR: needs

- A total research and technical computing infrastructure
- Quickly scalable to meet unforeseen demand
- Designed in January, purchased in February and operational in April or May
- Rapid deployment of dedicated servers
  – RDBMS, Ensembl, app servers, etc.

Bauer CGR: needs

- Usage patterns, user requirements and IT use cases largely unknown
- Key CGR Fellows and researchers just being recruited
- Known:
  - Large Rosetta Resolver installation
  - Would like to have a “linux cluster”

Bauer CGR: constraints

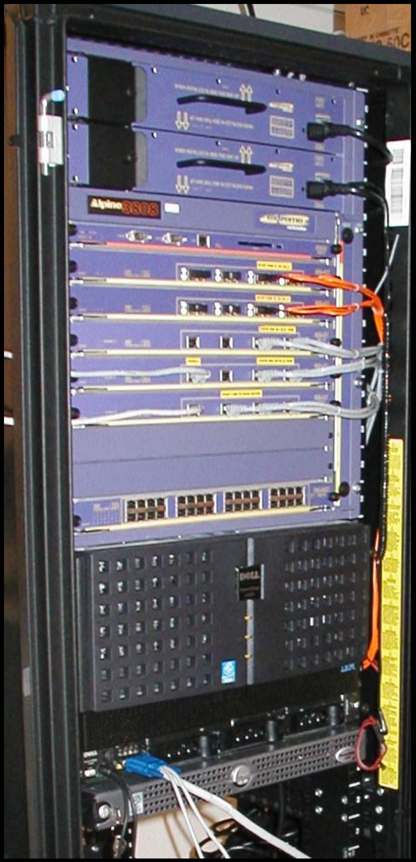
- No radical technology decisions
  - No IT director!
- Don't waste money
  - Start small but enable massive scaling in all directions
    - Compute power
    - Storage
    - Network & backup
- Oh yeah -- A staging area to build systems while server room is under construction

Bauer CGR: design

- Ignore the cluster initially
  - Cheap & pretty simple in overall scheme
- Start with the foundation items that have to support everything
  - NETWORK
  - STORAGE
  - BACKUP
  - (HVAC/Power)
- If we screw this up…
  - Expensive bits have to be thrown away or repurposed when it is time to grow

Bauer CGR: network issues

- Firewall/IDS is essential
  - Harvard is a big target; most campus IP address are directly reachable via the internet
  - Need DMZ for CGR hosted services
- Private network & subnet a must
  - Safe to predict cluster and CGR systems will be slinging lots of data back and forth
  - Eases cluster operational issues
- Big core switch needed
  - Extreme Networks Apine 3808
  - Large bladed chassis, mix and match ports as needed
  - Trunk to NAS and edge switches



Bauer CGR: Storage

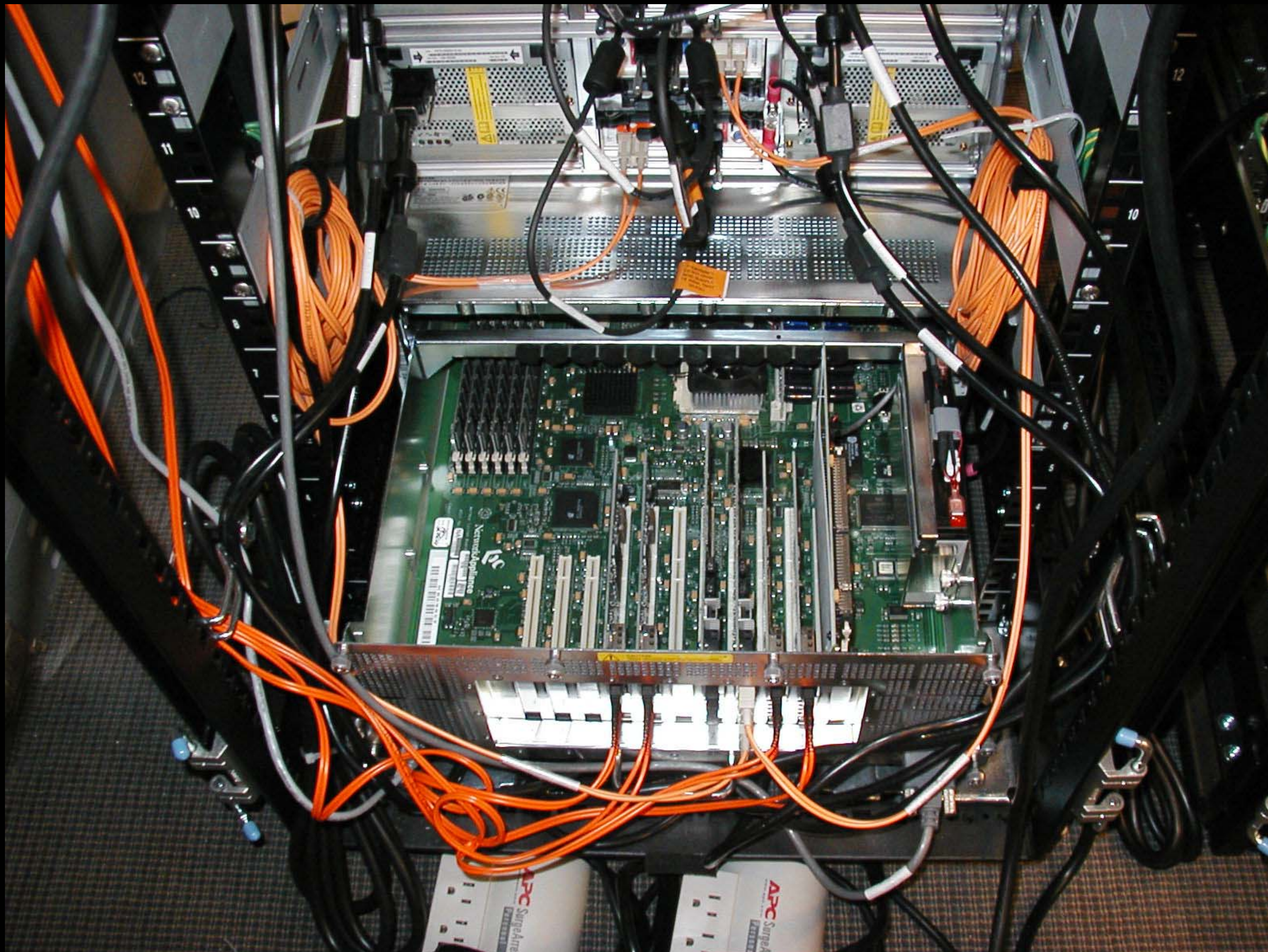
- Best guess on capacity issues
  - Need ~3TB usable disk in year 1
  - System should scale past 10TB raw capacity
- Many needs
  - Cluster storage
  - Lab data
  - User home directories
- Chosen
  - Network Appliance NAS
  - F840 Filer
  - 4TB raw capacity

Bauer CGR: Why NAS?

- Significant percentage of stored Bauer data must be readable/writeable by many systems and clients
  - Genetic sequence database
  - Microarray data
  - User home directories
    - Personal files visible from Unix, Windows or Apple systems
  - Platform LSF and other cluster related files
- No critical use cases justified the costs and complexities associated with a SAN or anything requiring fibrechannel interconnects
- Other possible solutions considered
  - Considered too radical

Bauer CGR: Why NetApp?

- Lots of cheaper NAS solutions out there
- Bauer could have built their own Linux fileservers
- Lots of storage startups building ‘NetApp killers’
- Network Appliance chosen because:
  - Amazingly low administrative burden
  - Great NFSv3 performance for the cluster
  - Fast to set up, easy to expand & trunk network connections
  - With ‘snapshot’ functionality users can recover their own deleted files
  - Simultaneous sharing of user home directories between Unix and Windows systems



Bauer CGR: Backup

- Large AIT tape library
  - 12 tape drives (6 populated today)
  - 364 internal tape capacity
  - Fibrechannel interfaces
- LAN free backups
  - Direct from NetApp via NDMP over fiber
  - Some tape drives shared via fiber with backup server
  - Legato software used

Bauer CGR: Cluster

- Modular & scalable cluster architecture
- "Compute Rack" contains
  - 30 dual processor compute nodes
  - Power distribution
  - 2 Ethernet aggregation switches
    - Each connected to 15 compute nodes
    - Gigabit uplinked to core cluster switch

Bauer CGR: Cluster

- Modular & scalable cluster architecture
- "Admin Rack" contains
  - Alpine 3808 core switch
  - Firewall and IDS system
    - 2 systems; 1 is a cold spare
  - Linux management & monitoring server
    - Dell Poweredge 6450
  - Network Appliance F840 NAS system

Bauer CGR: Cluster

- Modular & scalable cluster architecture

- “App Rack” contains
  - Cluster head & Platform LSF master node
    - Dell Poweredge 6450
  - 2 additional Poweredge 6450 servers
    - For use as database or application servers

Bauer CGR @ Harvard

Case Study 2

Harvard University
Department of Statistics

Stats Department @ Harvard

- Single department; multiple user groups
  - Undergrad and Graduate level research
- Focus on interdisciplinary informatics
  - Especially bioinformatics
- Significant amount of internally developed code
  - Researchers own software and algorithms being tested
- Significant interest in developing parallel software with MPI or PVM

Stats Department @ Harvard

- Fixed budget, very tight
- One shot purchase w/ grant money
  - Want price/performance
  - Will likely not be scaling much
  - Every dollar counts
- Existing Linux desktops and servers are overwhelmed
- Small office-sized server room

Stats @ Harvard: Needs

- "as many CPUs as we can afford"
- ~150 GB NAS storage
- Fast local I/O
- Fast network topology
- Every dollar counts!

Stats @ Harvard: design

- Compute Nodes
  - 10x Dell Poweredge 1650’s
    - Dual 1.4ghz Pentium III’s w/ 2GB RAM
    - Dual 1000TX Gigabit Ethernet NICs
    - Dual 80gb IDE disks
      - Software RAID for fast local I/O
- Network
  - Dell Powerconnect 5224 switch
    - 24 copper GigE ports for under $2,000

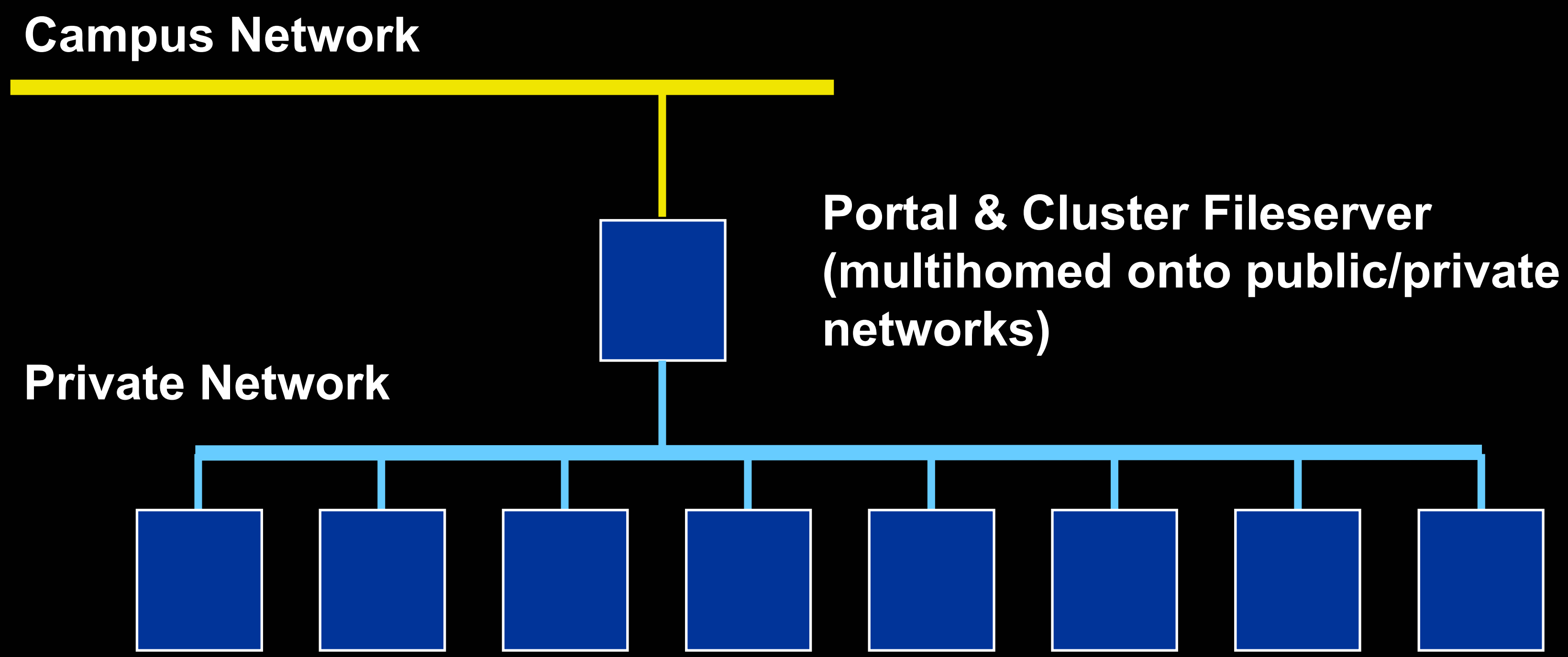
Stats @ Harvard: design

- Cluster head node & fileserver
  - Dell Poweredge 1650
    - Redundant power
    - 3x 72gb SCSI drives w/ hardware RAID5
      - Operating system only
  - Dell Powervault 220S storage enclosure
    - SCSI attached to the 1650 server
    - Space for 14x 10K RPM Ultra 160 SCSI drives
    - Installed with 4x 72 GB drives
      - 3x @ RAID5 + 1 hot spare

Stats @ Harvard: Sun Grid Engine

- Why GridEngine?
  - Open source; no licensing cost
  - Given tight budget even Platform's new low subscription pricing for LSF was not possible
  - SGE will easily support the batch queuing and resource allocation needs of the departments users
  - SGE also allows 'out of the box' both loose and tight integration with PVM or MPI parallel environments

silex.stat.harvard.edu



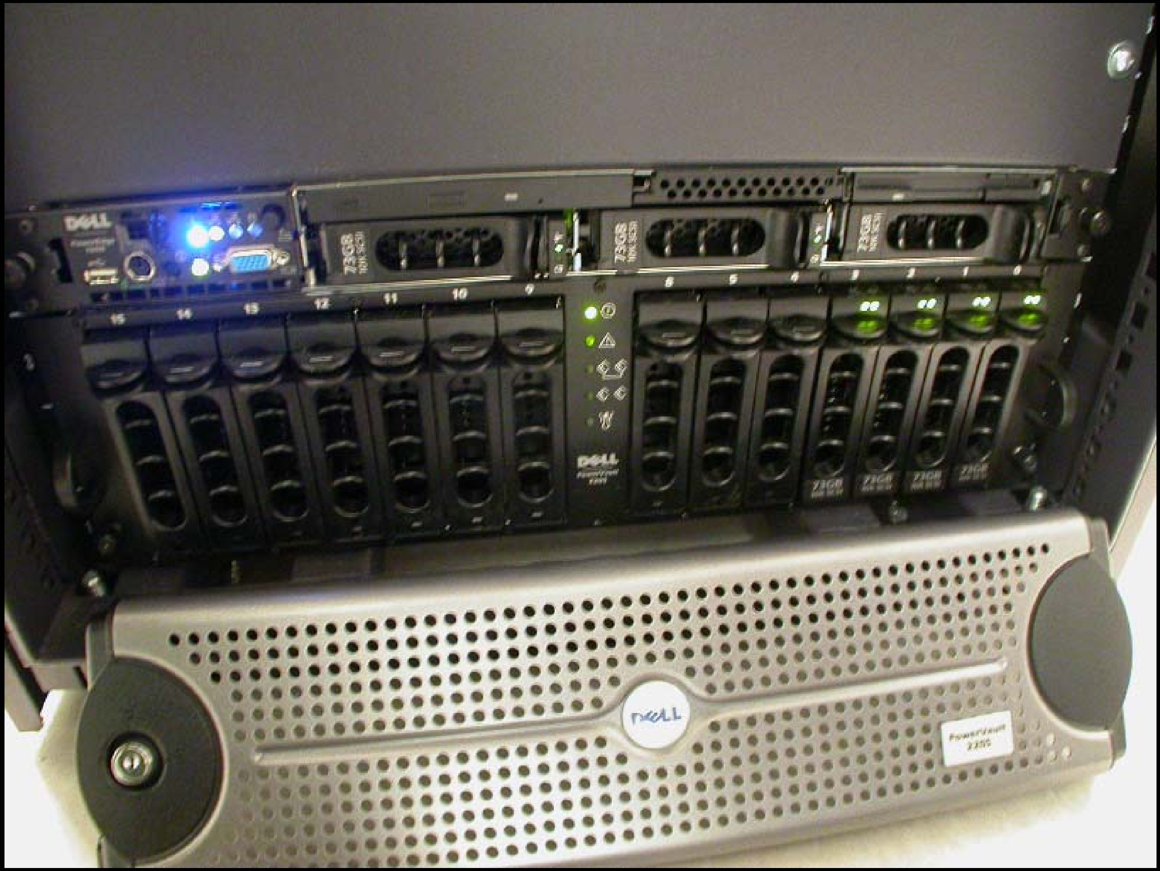
Stats @ Harvard

10x compute nodes /
20 CPUs

{ fileserver expansion space}

Head node
External disk enclosure

Stats @ Harvard



Stats @ Harvard

Stats @ Harvard: design wins

- Overall the compute nodes are nicely configured
  - Dual IDE drives allow for flexibility
    - Capacity w/ software RAID striping
    - Performance w/ software RAID mirroring
- Dell Poweredge 1650 servers are cool
  - Choice of IDE or hotswap SCSI drive enclosures
  - Onboard gigabit ethernet
  - Low failure rate compared to other 1U servers we've worked with
  - Aggressive pricing
- Dell Powerconnect 5224 switch is a great deal
  - 24 copper gigabit ethernet ports for less than $2,000

Stats @ Harvard: design wins

- The configuration of head node and fileserver has worked out well
- The Poweredge 1650 w/ external 14 bay drive shelf was a good use of limited dollars
  - Bigger Dell servers are much more expensive and only have internal capacity for a few more drives
  - Can't give up redundant power or hardware SCSI RAID in a head node
  - The storage enclosure can be moved to a dedicated fileserver if needed in the future

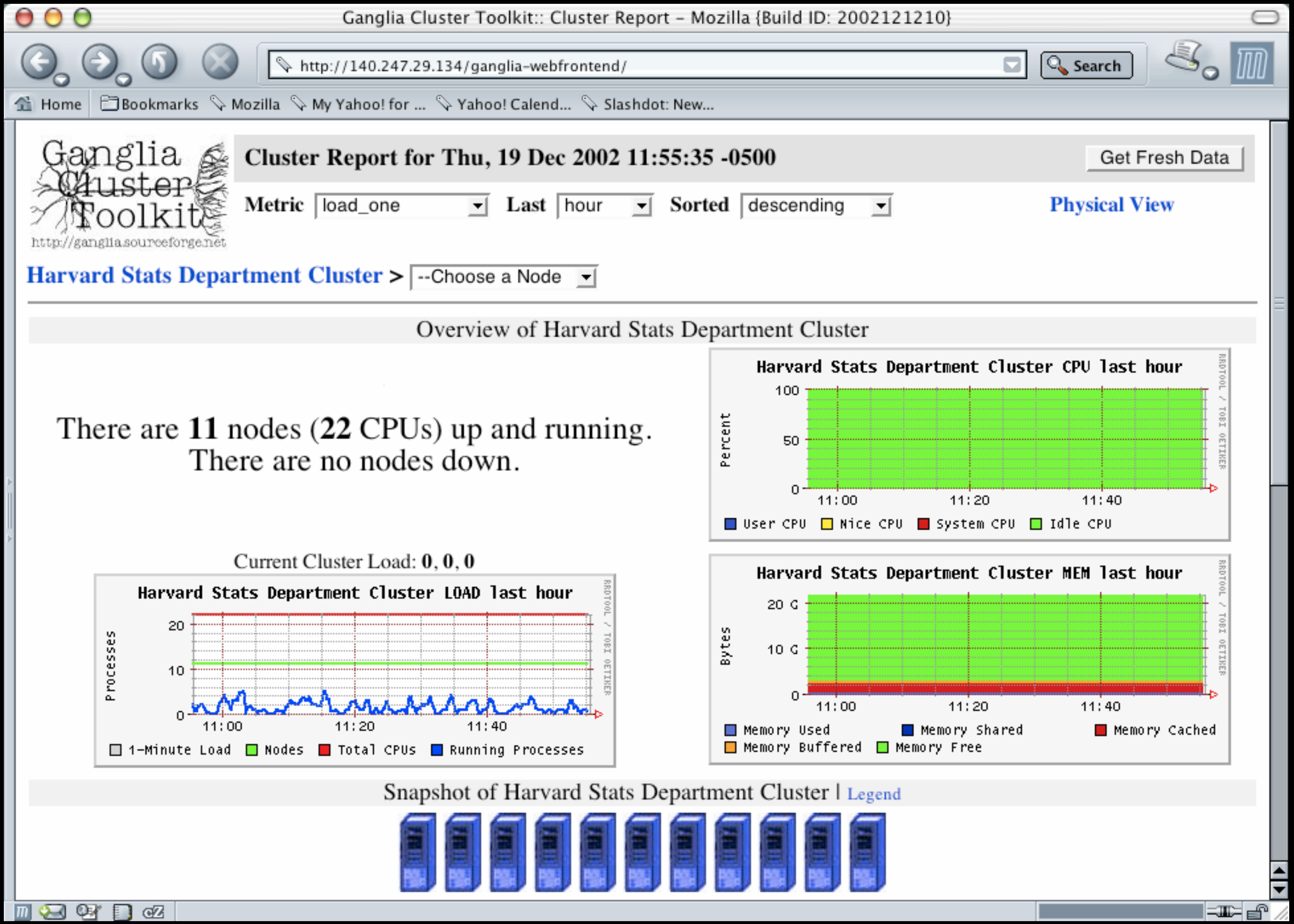
Stats @ Harvard: design tradeoffs

- Head node (PowerEdge 1650) is pretty small
  - Not bad though; we have SCSI hardware RAID5 and redundant power supplies
- Never a good idea to have your head node do double duty as a fileserver
  - Potential bottleneck
- Scaling
  - Existing fileserver and head node can only support a modest amount of additional cluster growth

Stats @ Harvard: design tradeoffs

- Backup
  - There is none
  - Budget did not allow for a single AIT or DLT tape drive let alone a changer, media (~100 per tape) or backup software licensing
- The cheap fix:
  - IDE 180 GB external USB disk(s)
  - Attached to head node / fileserver as needed
  - Files & data are rsync'ed to external disk
  - External drive(s) kept in safe place

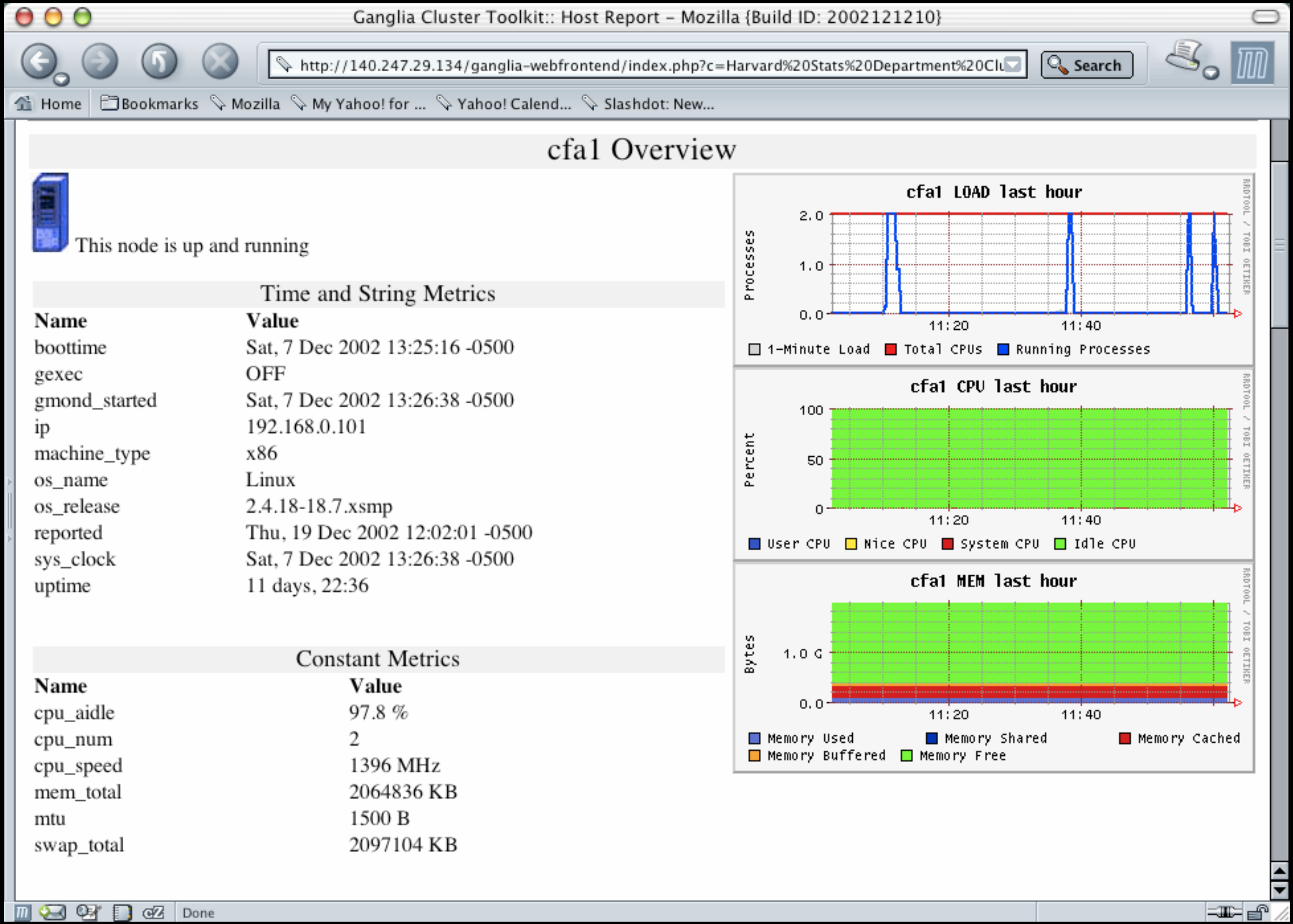
Stats @ Harvard: monitoring



http://silex.stat.harvard.edu/ganglia-webfrontend/

The BioTeam, www.bioteam.net

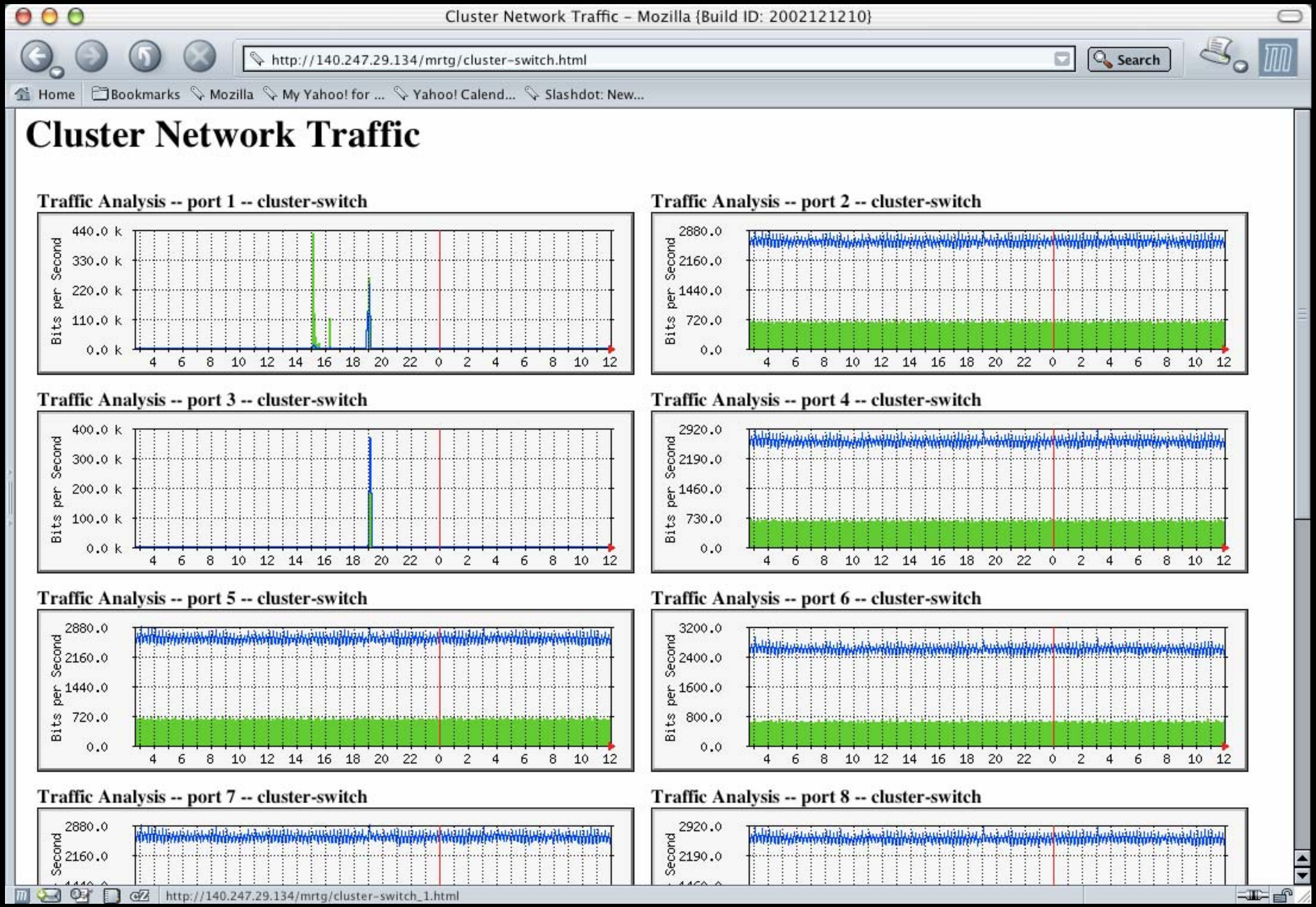
Stats @ Harvard: monitoring



http://silex.stat.harvard.edu/ganglia-webfrontend/

The BioTeam, www.bioteam.net

Stats @ Harvard: monitoring



http://silex.stat.harvard.edu/mrtg/

Mini Case Study -- Boston College

- Boston College Dept. of Biology
- Scaling up to hire bioinformatics faculty
  - Each will arrive with lab startup $$
- Want foundation in place so faculty can add to infrastructure rather than roll their own
- Department is partial to Apple but wants to support Linux as a runtime and application development environment
- Unsure about storage; needs to be scalable
- Limited initial budget & IT staff

Bio @ BC design: Hybrid cluster

- 10 dual-CPU Linux nodes
- 4x dual CPU Apple Xserve
- Gigabit network core
- Platform LSF for DRM layer
- DIY Hybrid SAN/NAS storage
  - Chosen for scaling & performance; Low initial costs as no need for FC switching or many HBAs;
  - Fibrechannel disks w/ Linux fileserver
  - Small tape changer & Legato sw for backup

What's new

Clustering with Xserve

- Apple has a rack mount!
- Several Xserve bioclusters already exist
  - 85 node system at TLL/Singapore
    - Platform LSF
    - Used for genome annotation pipeline
  - 4 node infx portal at xblast.tamu.edu
    - Sun Grid Engine 5.3p2
    - PISE framework + SGE hooks
      - Web portal to hundreds of web apps including EMBOSS and the altivec-enhanced binaries
  - Various others in academia
  - BioTeam is building a hybrid Linux/Xserve cluster for Boston College Dept. of Biology

Who would buy one?

- People looking for a small or midrange system with low administrative burden
  - Management & monitoring tools are fantastic
  - Automated OS installation & updating tools are finally here with OS X Server 10.2.3
- Existing Apple-friendly labs & departments
  - Powerful packages available within 'sign-off' limits of lab and department heads
  - This is how Xserve will sneak into industry
- Performance freaks with specific needs
  - Altivec-enhanced algorithims are very fast
    - HMMER, GROMACS, SW, etc.
    - Coders finding it easy to write altivec enhanced code

The BioTeam, www.bioteam.net

The BioTeam, www.bioteam.net

Should you buy one?

- Depends
- Price is not unreasonable for what you get; within Linux-on-X86 league
- Do your own eval and benchmarks
- If your application mix has a good union with the list of altivec-enabled algorithms you will be well served by taking an Xserve for a spin
- Hybrid Xserve/Linux clusters are on the way

The good…

- Solid hardware; easy to maintain on your own with spare parts kit
- Management/monitoring software that comes with OS X Server is very good
- Fast local disk, onboard GigE
- You can boot & provision them with a firewire drive or iPOD

The bad…

- 2 GB physical memory max
- Apple's NetBoot & NetInstall installation tools still require keyboard input; must use 3rd party tools to achieve 'hands off' system deployment
- Until very recently (10.2.3) there were still core things (like configuring RAID) that HAD to be done via a GUI
- Xserve is a great compute node; still waiting on Xraid and a suitable 'head node' chassis
- Only Platform LSF commercially supported
  - Several ports of Grid Engine exist

Grid Computing

- *IMHO* grid computing in the life sciences is nothing but hype and empty press releases
- There are few if any _real_ grids in the life sciences today; despite what vendors say
  - A cluster is not a ‘grid’ just because you say it is
  - Neither is a ‘cluster-of-clusters’ deployment
- Real world biogrids are on the way; several promising projects in academia and industry settings
- A real trend for sure; but over promised and under delivered today

Grid Computing - 2 big issues

- The definition of "grid computing" has been co-opted and twisted so far as to become essentially meaningless
  - Sun's SGE marketing is the easy example; there are many more
  - Any virtualization of resources is now a 'grid'
  - It is essential to compare definitions before you talk grids with colleagues or vendors!

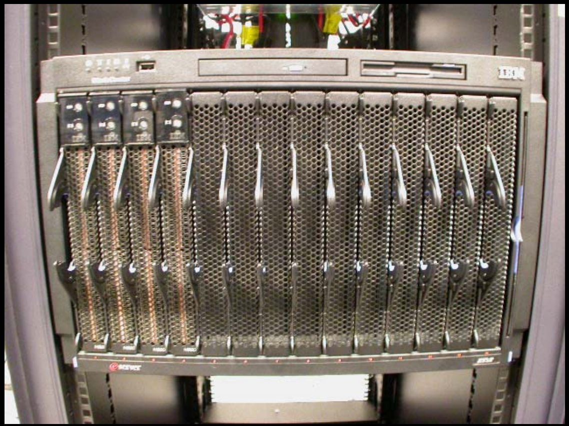
Grid Computing - 2 big issues

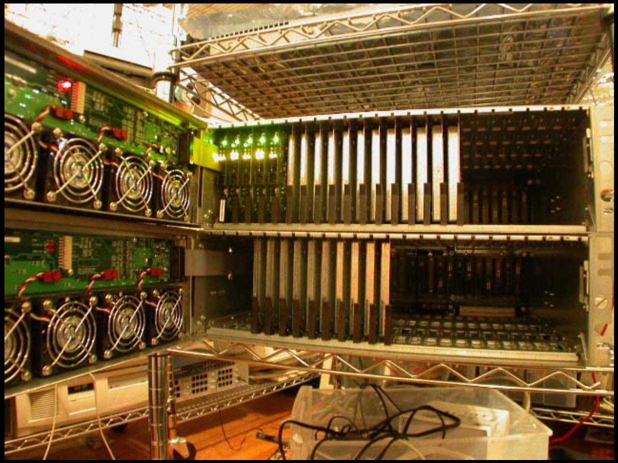
- The grid computing 'vision' that is being sold to you is not what we can actually deploy and implement today
  - What is promised:
    - 'Utility Style' computing on demand!
    - Seamless virtualization of resources across an enterprise!
    - Robust and manageable resource allocation & security policies!
    - Easy to use and easy to deploy!
  - What is delivered:
    - Heterogeneous cluster of clusters with job migration and resource brokering
      - Old news
    - Holes punched through your firewall

Grid Computing - Sanity check

- Figure out your real costs in time, software, security and operational burden with respect to what you gain with a grid
- Until grid deployment and policy based management and credential tools get better and easier to use you may find:
  - Easier and far faster (if not cheaper) to toss the grid and either
    - Purchase and deploy the CPU power you need locally
    - Invest in a fast VPN connection to a centralized computing resource
- Non scientific drivers may help justify complex grid deployments
  - 24/7 computing worldwide
  - Meeting peak computing demand spikes
  - Disaster tolerance & failover

Blade servers





The BioTeam, www.bioteam.net

Blade Servers

The BioTeam, www.bioteam.net

Blade Servers

- Price and proprietary hardware are not justified by wiring consolidation or management features
  - Proprietary is not bad; just not mass manufactured in enough quantity to get real price advantages
  - In many cases the management/monitoring methods and tools are very nice; just not nice enough to justify the price
- The key benefit is in environmental issues and physical system footprint
  - For some customers this alone will more than justify the acquisition costs

Blade servers - My $.02

- I'd use RLX blades as compute nodes if the price was right and I had benchmarked them for my specific use cases
- I probably would not use Compaq, IBM or Dell blades as cluster nodes
  - We can beat their density for far less $$ in many cases with mass market 1U half depth server chassis
- The Compaq/Dell/IBM blades may complement a whitebox cluster or general research computing infrastructure
  - Quick way to easily provision, deploy, maintain various web, database and application servers

Interesting Use Cases

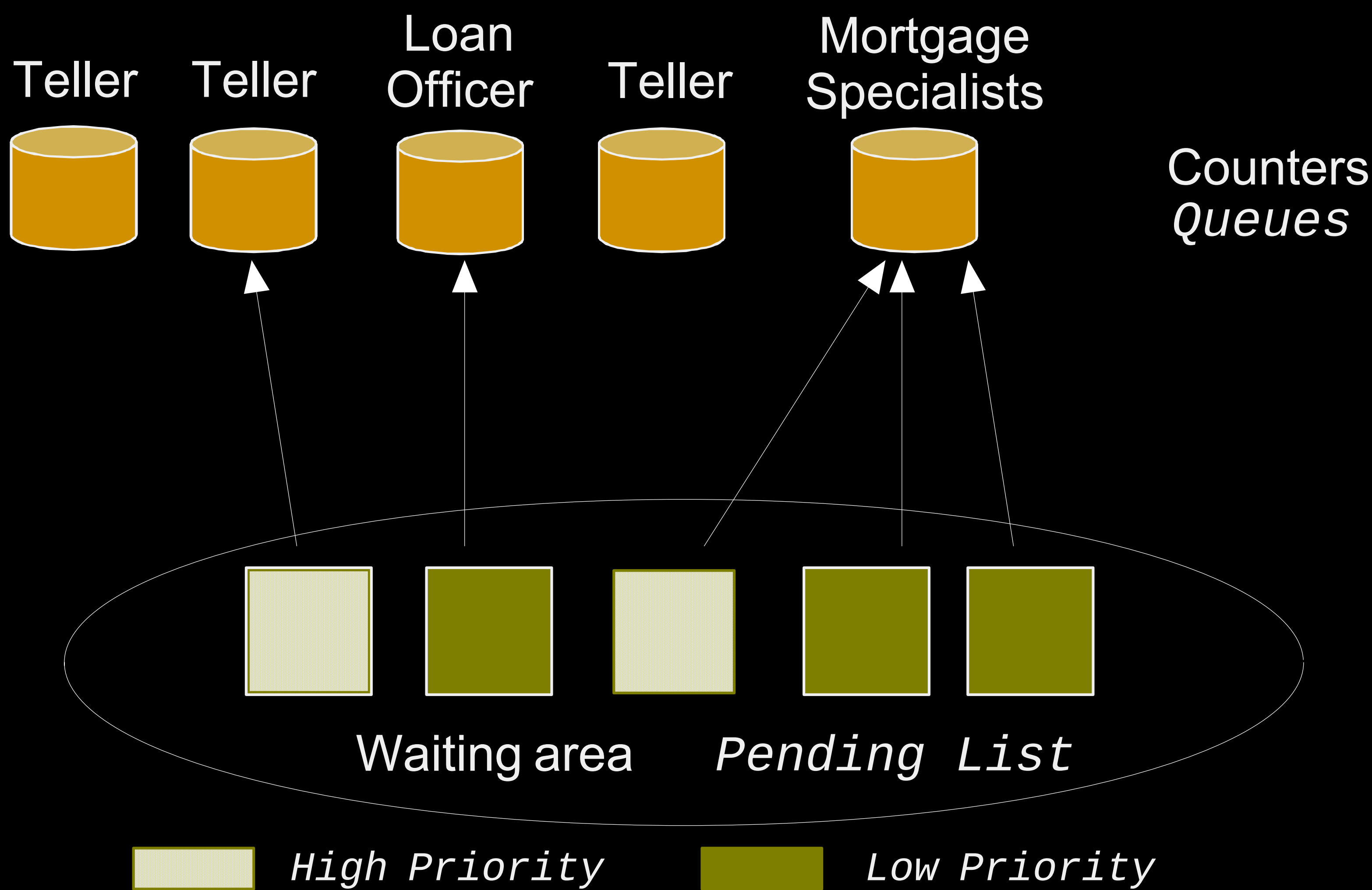
- You have a cluster; blast is running fine; Now what?
- Example: Beyond Genomics
  - I/O performance tuning
  - Visualization with MPI/povray
  - Cluster enabling matlab scripts
- Example: xblast.tamu.edu
  - Cool stuff
  - PISE framework on top of LSF or SGE
  - Instant web portal to hundreds of infx apps
- Workflows, pipelines and scientific application enabling is where the real fun is at

DRM Internals Example: How Grid Engine Works

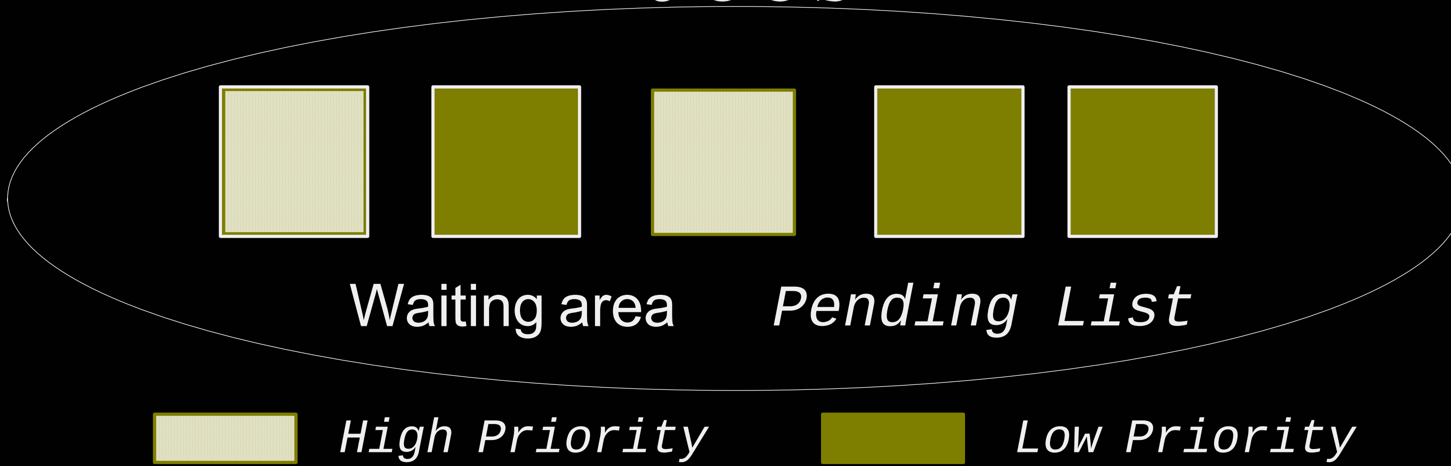
How Grid Engine Works

- If you have ever used OpenPBS, PBSPro or Platform LSF then prepare to be confused!
- SGE 'queues' are very different from 'queues' within LSF or PBS
  - Users do not submit jobs to any particular queue. Instead, users describe the resources needed by the job and SGE works to find the best possible queue

The Bank example



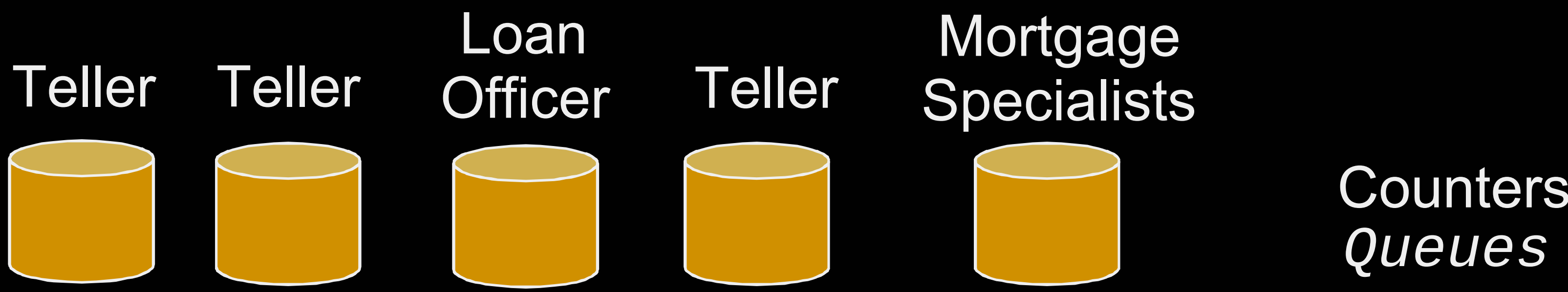
Jobs



- Cluster jobs are the bank customers. Jobs wait (‘pending’) in a holding area while clerks at counters provide service to bank customers, queues, located on servers and exec hosts, provide services for jobs.

- Like in the case of bank customers, the requirements for a particular job might be very different, and only certain queues might be able to provide specific services. Examples of requirements could include available memory, A certain CPU or machine architecture, free disk space, etc. etc.

Queues



- A Sun Grid Engine queue is a container for jobs that are allowed to execute on a particular host concurrently. A queue determines certain job attributes, for example, whether it might be migrated.

- Throughout its lifetime, a running job is associated with its queue. Association with a queue affects some of the things that can happen to a job. For example, if a queue is suspended, all the jobs associated with that queue are also suspended.

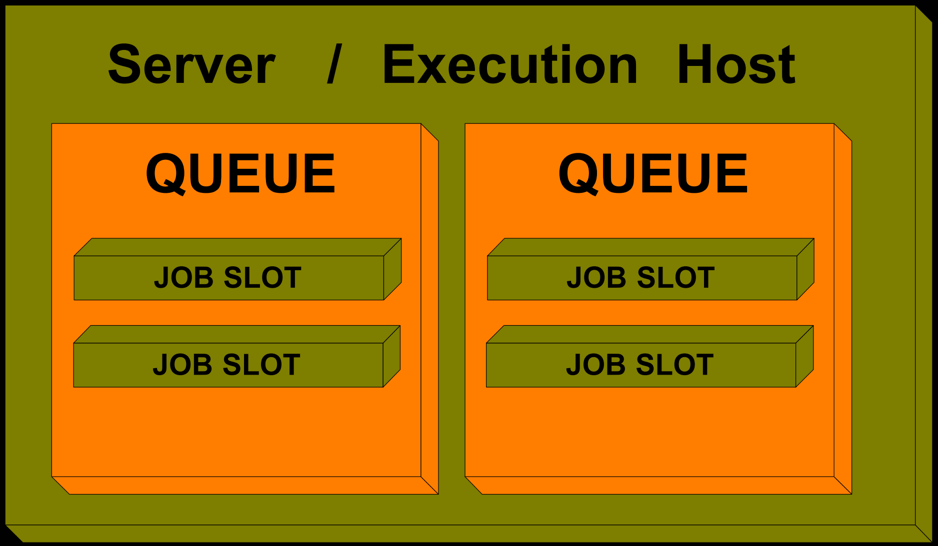
- In Sun Grid Engine, there is no need to submit jobs directly to a queue. You only need to specify the requirement profile of the job, which includes memory, operating system, and available software. Sun Grid Engine dispatches the job to a suitable queue on a low-loaded host automatically. If a job is submitted to a particular queue, the job is bound to this queue and to its host, and thus Sun Grid Engine is unable to select a lower-loaded or better-suited device.

Queues

- Queue: Virtual or real host

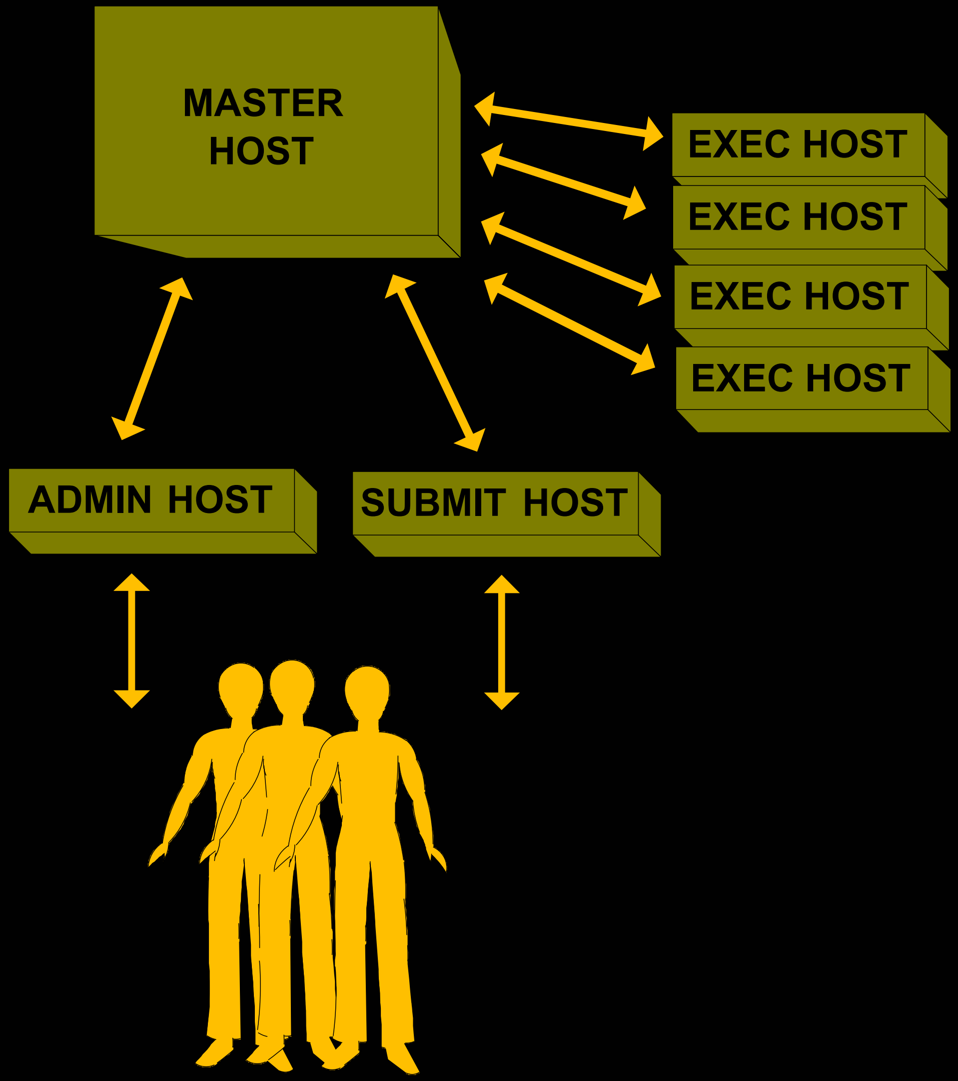
- Jobs acquire the attributes of the queue, e.g:
  - CPU & Memory Limits
  - Access permissions
  - Suspend/resume thresholds

- Job Slots: allow multiple jobs with the same characteristics to run on the same host

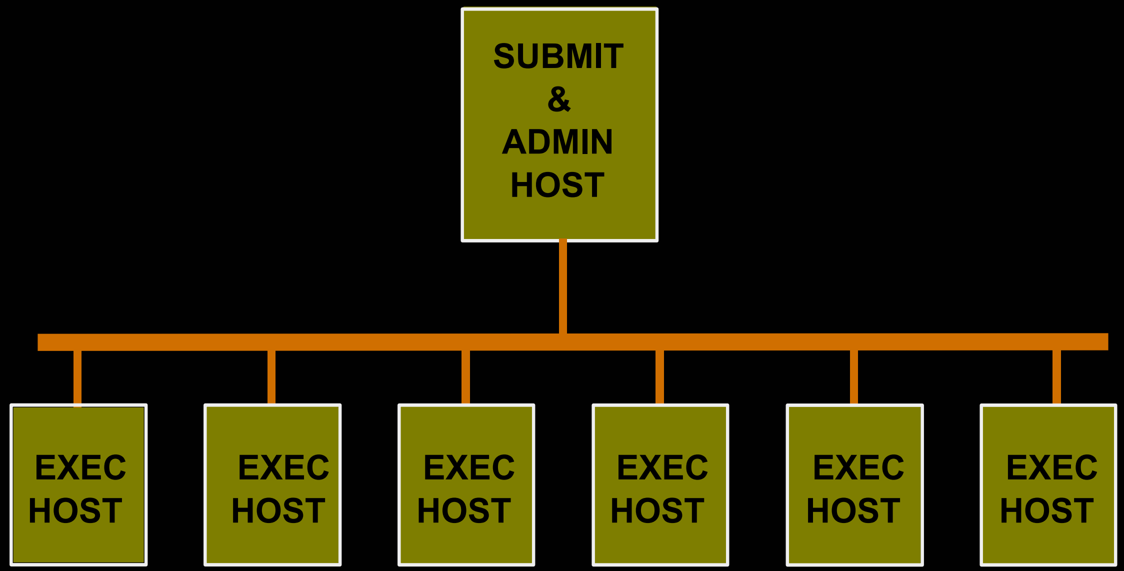


Scheduling

- Job Selection
  - Round-robin
  - FIFO
  - Priority
  - *SGEE adds policy based*
- Queue Selection
  - Queue availability
  - Access rights
  - Resource matching
  - Queue preferences
    - Load-balanced
    - Best-fit
    - Most free memory
    - Fastest CPU

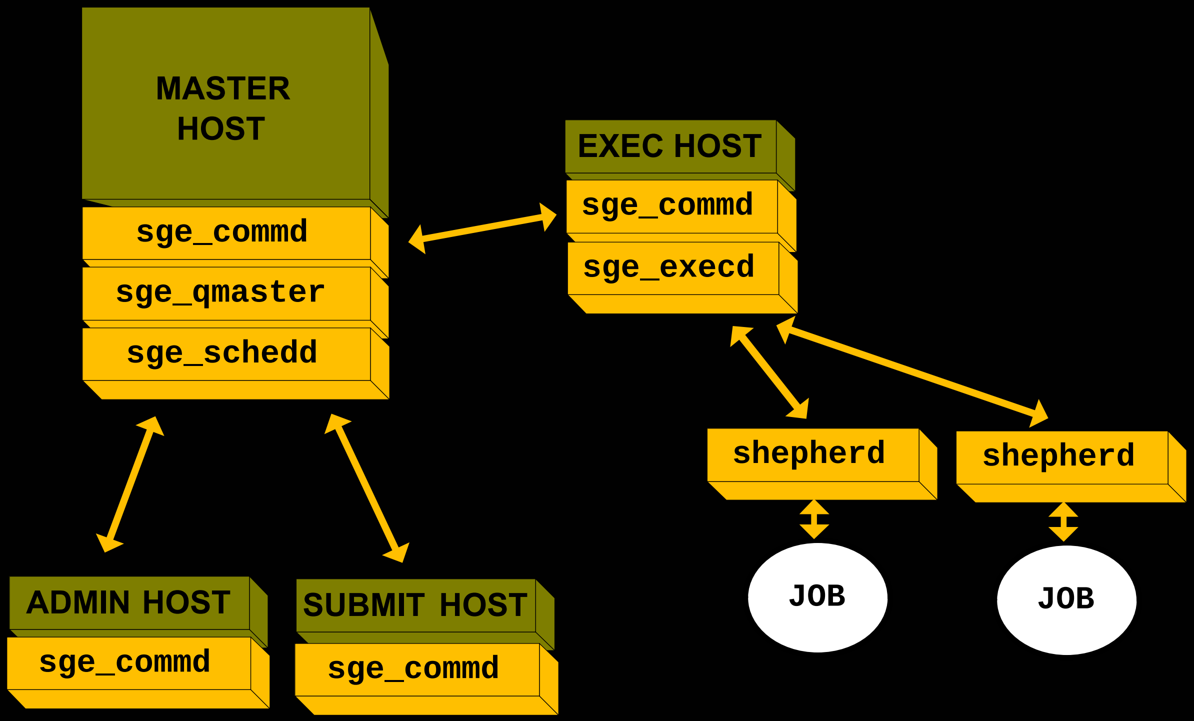


Generic compute farm

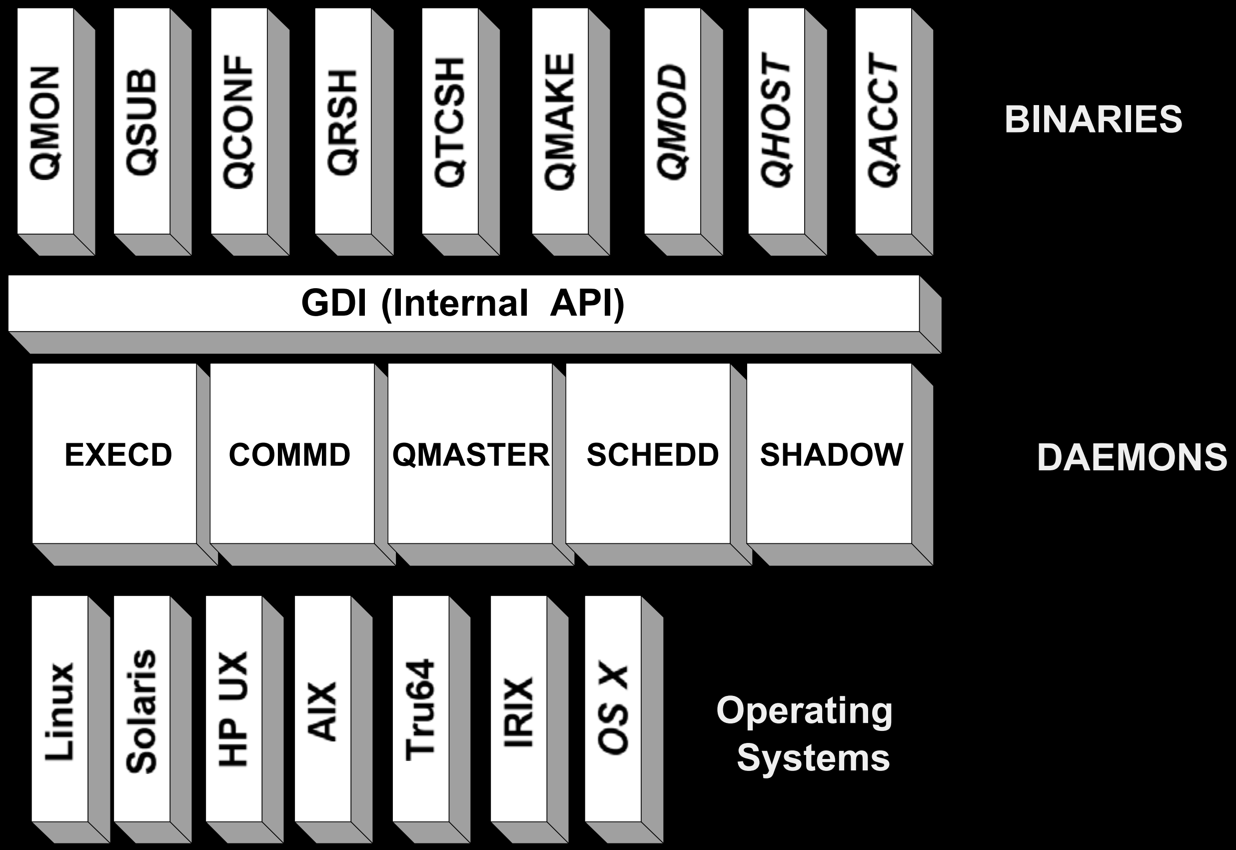


Host Types & SGE Daemons

- All machines run 'sge_commd'
- Exec hosts run 'sge_execd'
- Master runs 'sge_qmaster' 'sge_schedd'
- Master optionally can run 'sge_execd'



Grid Engine Architecture



sge_qmaster daemon

- Controls overall behavior of the cluster
- Holds most current state about jobs, queues and other SGE objects
- Interoperates with sge_schedd for scheduling activities
- Answers GDI requests from clients
- Delivers dispatched jobs to proper sge_execd's

sge_schedd daemon

- Dedicated scheduling process
- Event driven via GDI calls
- During startup:
  - Contacts sge_qmaster; gets full state info
- After startup:
  - Gets new state changes from qmaster
  - Updates internal state information
  - Performs a scheduling run
  - Sends job ‘order list’ back to qmaster
  - Rinse, repeat …

sge_execd daemon

- 1 daemon per execution host (compute node)
- Controls all running jobs
  - Including suspend/resume & reprioritization
- Collects job information
  - Resource consumption, exit code, etc.
- Collects & reports host information to qmaster
  - System load, free memory, etc.

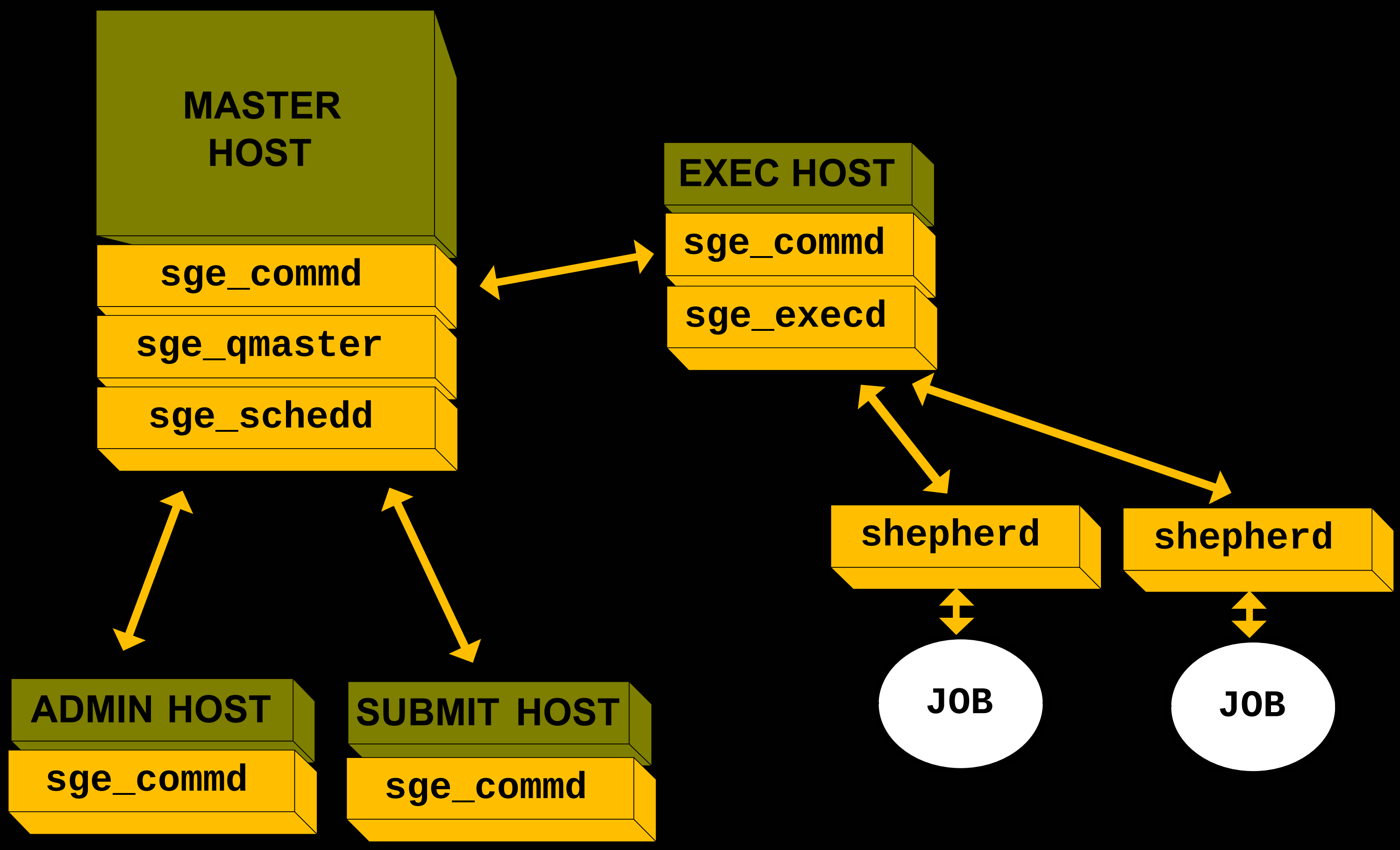
sge_execd daemon

- During startup:
  - Initializes
  - Connects to sge_commd on local host
  - Tries to connect/register with sge_qmaster
    - If failure, loop & retry
  - Looks for old jobs
    - If found, establishes process control again
  - Cleanup finished jobs, report to qmaster
- After startup:
  - Loop
    - Receive request
    - Process request
    - Report to qmaster

shepherd process

- Invoked by sge_execd to handle a single job
- Sets up resources before jobs begin
  - Prolog script, Parallel environment startup, etc.
- Starts the job as a child process
- Controls the job
  - Suspend/resume/terminate signals
  - Handles checkpointing
- Cleanup after job termination
  - Epilog scripts, Parallel environment shutdown, etc.

Putting it all together



Step by step

- Job submitted from submit host
- User identity, cwd, environment preserved
- Master host receives job request, places entry in database and notifies the scheduler
- Scheduler attempts to dispatch job to best possible queue. If successful queue is returned to the qmaster daemon
- Qmaster sends job to the proper sge_execd
- Execd launches a shepherd process
- Shepherd sets up job, runs job as user and cleans up
- Exit status & accounting data passed back to sge_execd and sge_qmaster

end ;

- Questions/Comments/Rants
  - chris@bioteam.net
- BOF Meeting Tonight
  - Clusters & Grids
- Join the BioClusters Mailing List
  - bioclusters@bioinformatics.org
  - http://www.bioinformatics.org
    - Look under “Mailing Lists” topic