Pink
A 1024 Node Science Appliance

Sung-Eun Choi, Erik Hendriks,
Ron Minnich, Matt Sottile,
Greg Watson

LA-UR-02-7246

S

Cluster state-of-the-art: *ca.* 1990

- Scripts for install, maintenance, scheduling, etc.
- 1 FTE required (at least) per 128 nodes*
- NFS for file systems, which works poorly in clusters
- Console is either glass tube or serial port
  - Impractical on large scale
- Have to "rsh" to node to do things
- Proprietary Unix makes changing things hard

*informal survey of research and industry systems

Cluster state-of-the-art: 2003

- Scripts for install, maintenance, scheduling, etc.
- 1 FTE required (at least) per 128 nodes*
- NFS for file systems, which works poorly in clusters
- Console is either glass tube or serial port
  - Impractical on large scale
- Have to “rsh” to node to do things

*informal survey of DOE labs

General Goals

- To provide users with a single logon cluster:
  - That provides SSI capabilities
  - That scales to 1024 or more nodes
  - That is compatible with existing practice
  - With O(1) software failure points instead of O(n)
  - With software that anticipates or tolerates failure
  - As a consequence: single point for management, console, and use

But … we also want:

- .01 FTE per 128 nodes (current is 1)
- 5 seconds to install all software on all nodes
- 5 seconds to upgrade
- 5 seconds to boot
- Eliminate Ethernet and serial port networks
- Reduce all the significant measures by at least O(100)

Science Appliance

- Culmination of a research program started in 2000
- Our goal was to address a simple problem:
  - clusters are too hard to build, install, manage, and use
- Hardware problems are easy to solve
  - if you buy unreliable hardware, it will be unreliable
  - so don’t buy unreliable hardware
- The issue is software architecture
  - software requiring disks on each node requires unreliable hardware

Science Appliance Architecture

- Fast, reliable, maintainable firmware
- Single process space
- Private name spaces
- Parallel file system
- Scalable monitoring
- In short, very little in common with older Linux cluster techniques
- Few moving parts

Science Appliance Software

Available as "Clustermatic"

- LinuxBIOS (firmware)
- BProc (single process space)
- V9fs (private name spaces)
- Panasas or Lustre (parallel file system)
- Supermon (scalable monitoring)

LinuxBIOS

- Replaces existing BIOS
- Boots to Linux in a few seconds
- Eliminates BIOS setup screens and stupid failure modes
  - “No keyboard hit F1 to continue”
- Configures hardware correctly
- Allows maintenance to be done from Linux

Beoboot

- Loads Linux kernel from network
- Then boots Linux from Linux
- Eliminates need for PXE, Etherboot, etc.
- Puts the full capabilities of Linux in use for booting
  - Can boot over Myrinet, QSW, etc.
  - Can use IPV4, IPV6, NFS[2,3,4], etc.

Global PID space

- Global process state maintained from one node only (BProc master)
- Applications that use standard process system calls work no matter what node they are on
- Problem of PID management reduced from $O(N^2)$ to $O(N)$ (*N is # nodes*)

BProc

- Single system image for process space
  - i.e. ALL process system calls work transparently across the cluster
- Allows a cluster to have single login, single IP address, single point of control
  - rsh is dead
- Tested to 15,000 procs
  - Started up in 3 seconds over TCP/IP
- Uses *asymmetric* single system image as opposed to *symmetric* single system image

Symmetric single system image

- *System-centric*
- The entire system shares all resources transparently
- Which means all nodes have resources from all nodes
- If anything fails, you fail. If you fail, everyone fails.
- Actually, worse than that …
- Add in all file servers, etc.

Asymmetric single system image

- *Application-centric*
- Application sees unified
  - PID space
  - File system space
  - Message-passing space
- Global process space is visible from front-end node only
- File system provided by *private name spaces*
- Message-passing space is TCP/IP or other network – solved problem

Asymmetric SSI Advantages

- Scaling
  - conventional SSI scaling is measured in 10s
  - our system scaling is measured in 1000s
- Much less complex
  - no distributed lock manager, consensus protocols, message journals, etc.
  - in this case, simpler means faster and more reliable

Private name spaces

- Application-centric, not node-centric
- With global file systems, *node* reliability is determined by reliability of *all* servers
- With private name spaces, *application* reliability is determined by reliability of servers used *by that application*
- Greatly reduces the reliability requirements

V9fs private name spaces

- Process-group-private mounts *a la* Plan 9
- Eliminates global NFS mounts and attendant performance and reliability problems
- Performs as well as NFS in tests
- Processes carry mounts as they migrate

BJS: BProc Job Scheduler

- Exploits BProc capabilities for performance
- Single scheduling process for entire cluster
- Framework for scheduling policy is a pluggable module
- Being integrated to LSF/Maui

Supermon

- Monitors OS, application, and hardware info
  - many systems can’t do hardware monitoring from Linux!
  - example: Compaq DS-10 cluster requires serial ports for hardware monitoring
- Scalable job monitoring with support for hierarchy
- Data stream based on symbolic expressions (LISP)
- Allows stream composition and filtering based on symbolic rules
- Real-time visualization

Other work

- ZPL
  - automatic check pointing
- Debuggers
  - parallel
  - relative debugging (Guard)
- Latency tolerant applications

Previous LANL Science Appliance systems

- 13-node cluster at SC 2000
- 128-node Alpha cluster assembled June 2001
- 128-node dual P4 cluster

Systems outside LANL

- Full “Science Appliance” installations
  - SNL/CA
- Partial installations
  - Academic (eg. ENIAC cluster at U. Penn, Clemson U.)
  - Industry (eg. Cluster at Pharmaceutical company that processes 25 jobs/second)

Pink: Newest LANL Science Appliance

- 1024 nodes
  - Dual P4, 2.4 Ghz, SMT (Hyper Threading)
- Myrinet interconnect
- Panasas file systems
- Science Appliance software

Pink Vital Stats

- 1024 nodes
- 2048 P4 CPUs
- 1024 Myrinet cards
- 2048 fiber (8.8 miles)
- 3072 switch ports
- 4096 RAM sticks (~2TB)
- 7000+ fans
- 1 disk
- 1 CDROM

The nodes

- 2- 2.4 Ghz CPUs
- 1 GByte per CPU
- Mounted in a .8U rack mount

The interconnect

- Myrinet “C” cards
  - 200 Mhz CPU
  - 64/66 PCI
- Fiber
- Clos network with full bisection bandwidth
- Only network for all nodes
  - This is somewhat novel
  - Every other cluster needs Ethernet for maintenance and/or booting

Parallel file systems

- Panasas
- Lustre
- NFS V4 (via Panasas or Lustre)
  - Not strictly a parallel file system

What's missing?

- Parallel debugger
  - i.e. TotalView for the small scale (128 nodes)
  - something else for the large scale (256 and up)
- New programming paradigm (e.g. ZPL)
  - SPMD does not hack it at this scale
- Failure anticipation
  - although failure rates on Pink are so far very low

Short-term tasks (Linux)

- Fix NFS
- Processor affinity
- Improve VM system for large contiguous areas
- Let apps know the state of their pages
  - Dirty, in-memory, etc.
  - Previously done on FreeBSD
  - Runtime systems could use this

Short-term tasks (Net)

- Improve map times for source-route net
  - Myrinet work will apply to Infiniband
- “Network serial port”
  - Minimal Myrinet Control Program that gives us poll-based NIC interface
- Fix route computation
  - Way too ad-hoc, way too slow, way too fragile
- Fix route loading
  - Done!

Future Opportunities

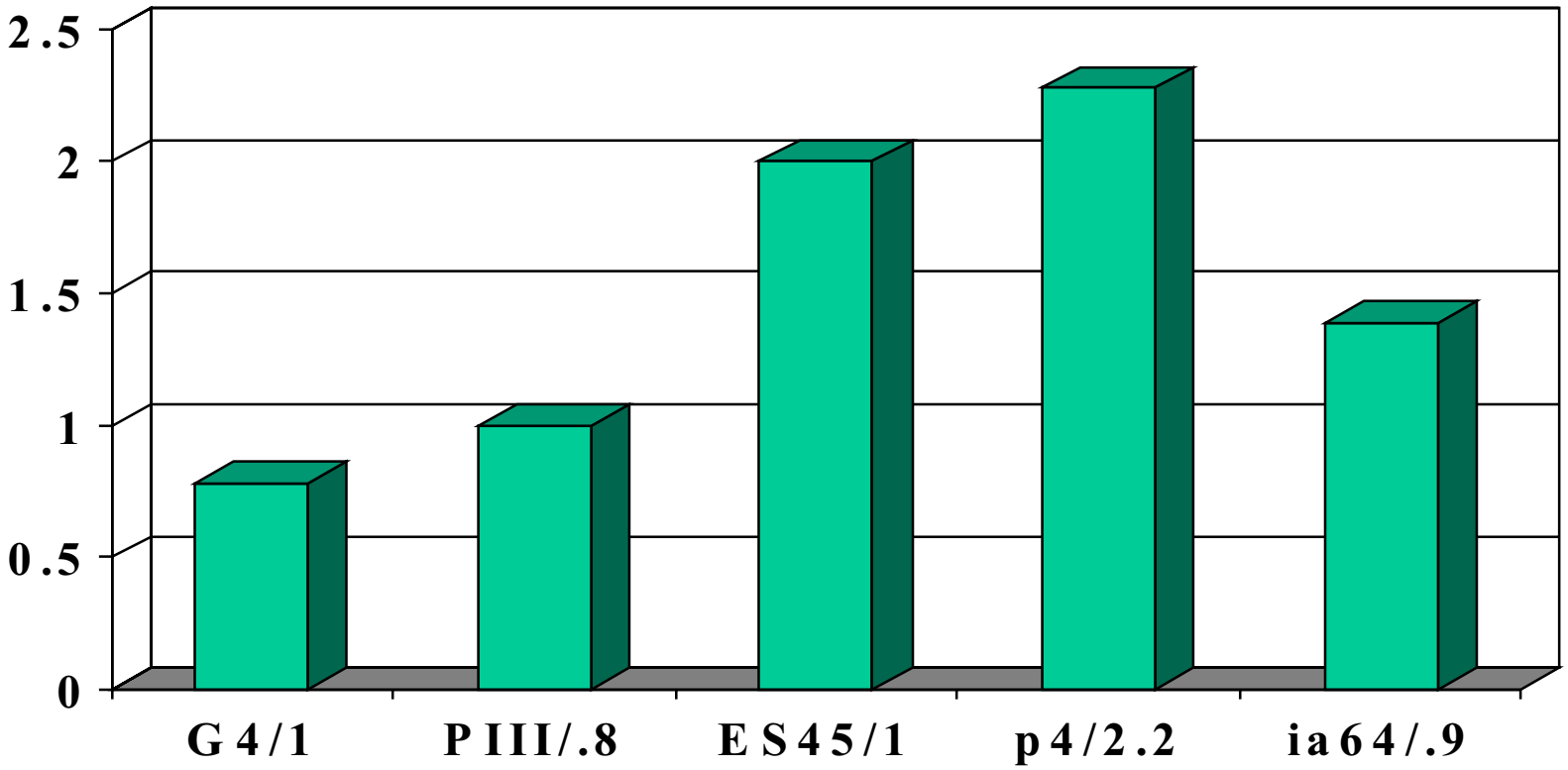
- Faster boot
- Application-specific configuration of node
  - e.g. hyper threading on/off per application
- Failure anticipation

Conclusion

- Science Appliance is a dramatically new cluster software architecture
- Eliminate as much software as possible
  - especially any and all Perl and other such guck
- The software architecture impacts the hardware architecture:
  - eliminate as much hardware as possible
  - especially if it has motors
- Provide users with a *single system image* environment

End of main talk slides (this slide added by matt)

Protein folding application

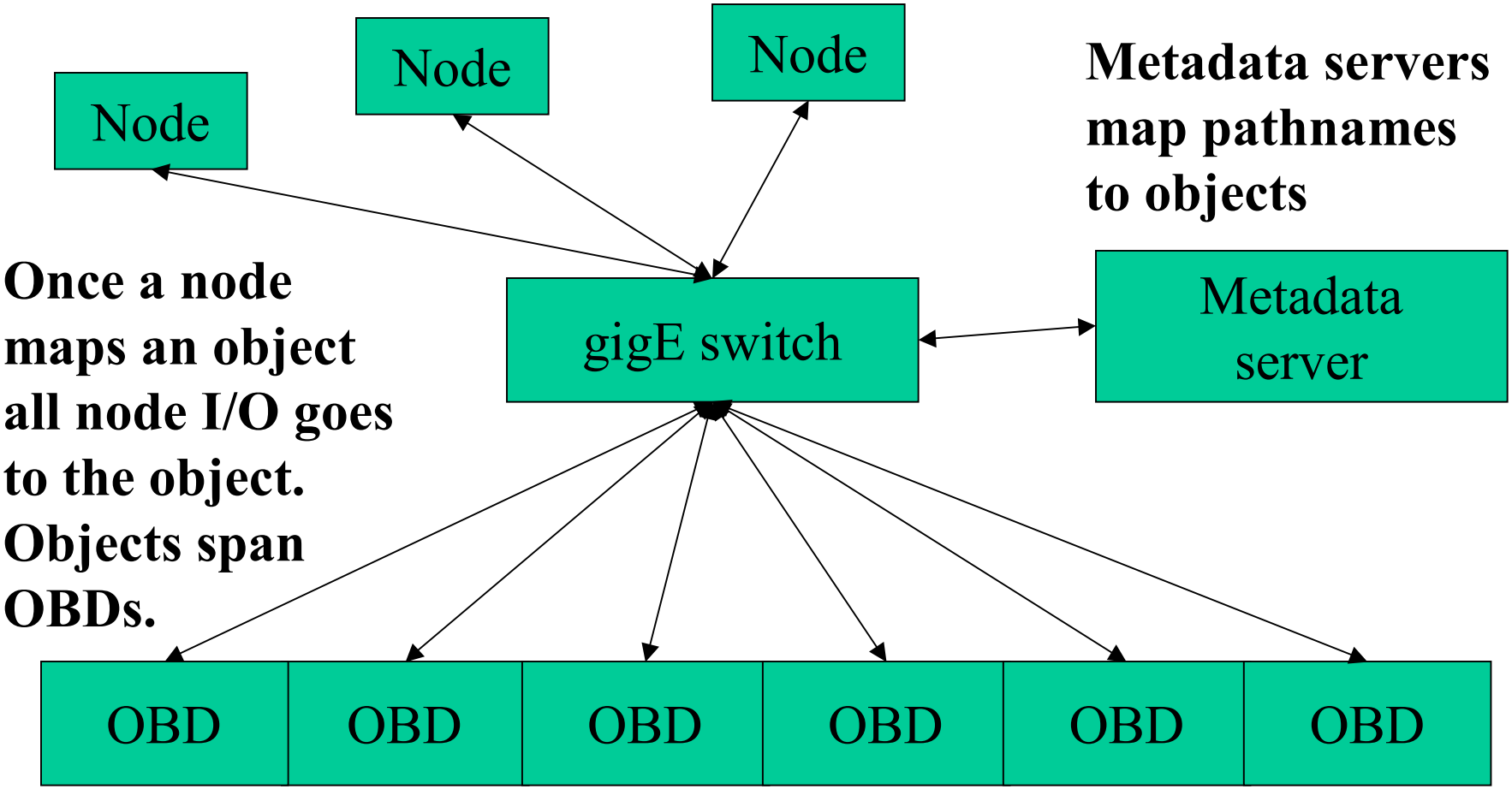


Performance relative to PIII 800Mhz

Panasas

- 128 Panasas Object-Based Disks
  - 2 disks in a cabinet with a power cord and gigE
- 64 I/O nodes in Pink connect to gigE switch
- Bandwidth is pretty good
- Still in a rough state but very promising

Panasas structure



Lustre

- Lustre Cluster File System
- DOE Tri-Lab funded, GPL, developed by Braam et. al.
- Similar scale to Panasas
- Architecture is pretty much the same as Panasas
  - With one advantage; server is Linux-based, GPL

File systems general comments

- An early, and failed, goal was to put the file system into the disk
- There is enough CPU in there to do it
- Now the systems all look like Linux-based motherboards running a server
- So why is it called OBD?
  - Marketing …

Compiler technology

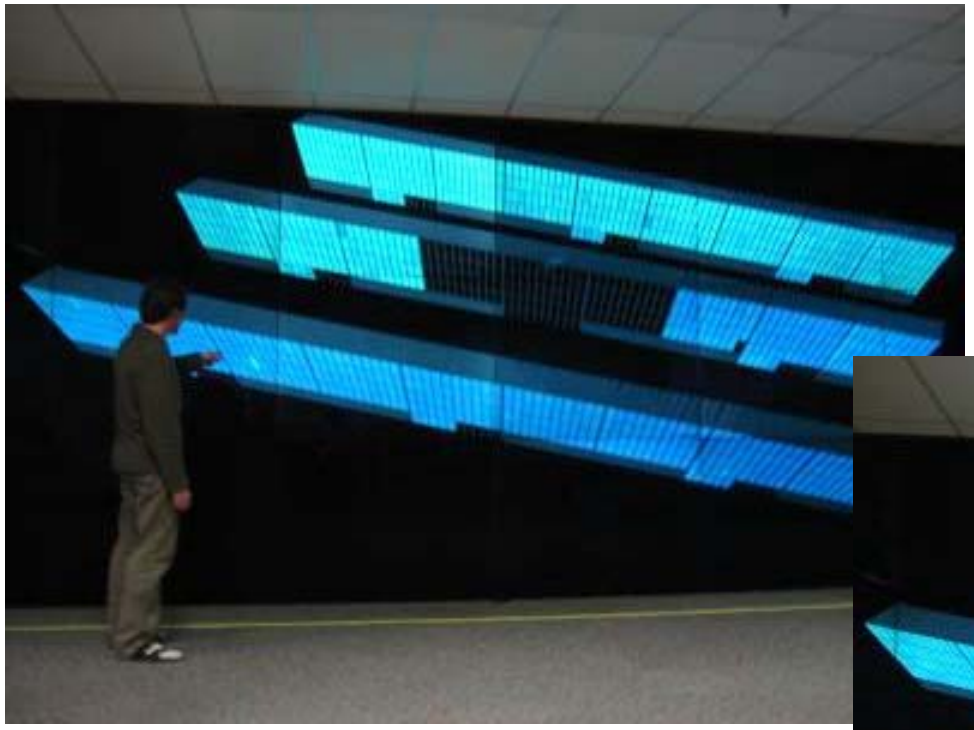
- Compiler support for check pointing
- Job reallocation in the event of failure
- Early results show:
  - Up to 75% reduction in checkpoint size
  - Can take a job running on □ processors and restart on □ processors

So just fix the BIOS!

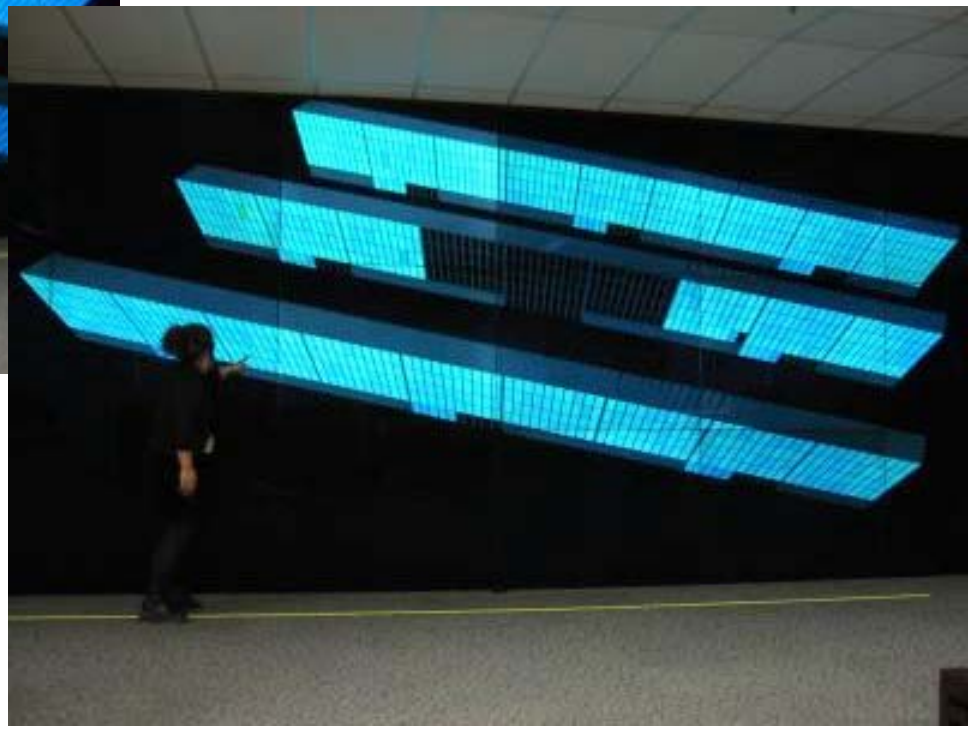
- Not fixable
- Misconfigures memory
- Misconfigures PCI
- Full of bugs (busted tables, etc.)
  - This recently broke Linux 2.4.19 …
- “no keyboard hit <F1> to continue”
- Windows-based configuration
- No configuration from Linux

Short Form

Seeing Pink

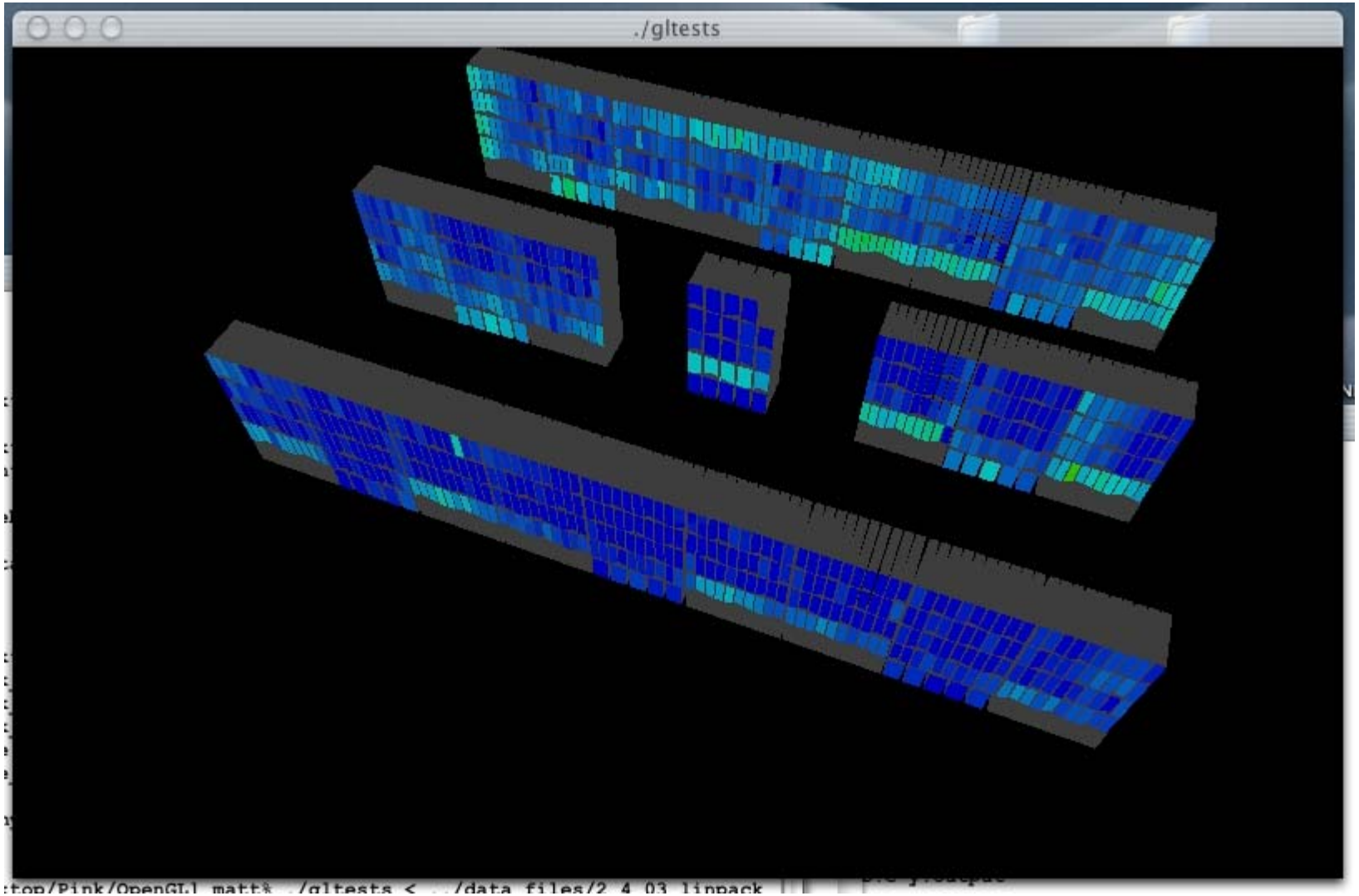


Data from ICE boxes
CPU temp is coming



Integrated Supermon data
With OpenGL imagery

Hot Pink



Pink Vital Stats

- 1024 nodes
- 2048 P4 CPUs
- 1024 Myrinet cards
- 2048 fibers (8.8 miles)
- 3072 switch ports
- 4096 sticks of RAM
- 7000+ fans
- 1 disk
- 1 CDROM

- LinuxBIOS
- Linux in FLASH
- Linux boots Linux over Myrinet
- No Ethernet
- Single Process Space
- Private Name Spaces
- Scalable Monitoring

What’s Unique?

- Just about everything except the nodes
- Oh, and except LinuxBIOS
- We used to think that was unique, but many people are using it

What’s Unique: Linux in FLASH

- Boot kernel in FLASH
- Boot init ram disk in FLASH
- FLASH ain’t FLASH; really, it’s an IDE-FLASH
  - But not for long: future Intel FLASH can grow to 256 MB (!) in Low-Pin-Count parts
- This is going to become more common

What’s unique: Linux is our bootstrap

- This is also less and less unique
- We use it in a unique way: there is no Ethernet
- We boot and run over Myrinet
- We don’t need limited, unreliable UDP-based protocols
- Our boot protocols are TCP/IP-based

What's unique: Single Process Space at this scale (bproc 3)

Boot 1022 nodes	222 secs		
Raw Process	1022 nodes	1.27 MB	.4s
	1022 nodes	4 MB	.7s
	1022 nodes	12 MB	1.6s
	1022 nodes	64 MB	7s
MPI no-op	1022 nodes	Small	1.6s
	2044 CPUs	Small	2.8s

What's unique: Monitoring that is in-band, and fast

- Can monitor single node over net at 11 Khz.
- Monitor whole cluster at 6 hz.
- Slow, right? Many systems are 1/30 hz. or less (see: Ganglia)
- No out-of-band serial network for monitoring

What’s unique: No Damn Disks

- Don’t need them, don’t want them
- But don’t have NFS root, either
- Root is in a RAM disk
- Footprint is 20M (I.e. who cares)
- Footprint is bounded, unlike NFS root footprint
- Parallel FS is Panasas and/or Luster

What’s unique: no serial port, no ethernet, no cd, no floppy, …

- Compute nodes are compute nodes: network-attached CPUs that do nothing but compute and send network packets
- They don’t need to be PCs and they don’t need to run full Redhat distros

What's unique: private name spaces a la Plan 9

- File system mounts are attached to an *application*, not a *node*
- With global mounts (e.g. NFS), *nodes* are as unreliable as *all file servers*
- With application mounts, *applications* are only as unreliable as *their own file servers*
- As we get more file servers, this gets more important

What's unique: we fixed Myrinet

- The Myrinet mapper is so slow it was killing our boot-time goals
- So we fixed things
- Route from compute node to master stored in CMOS
- Contact is made instantly, routes loaded instantly
- Minutes become seconds

What's unique: so far, most of it is unique

- But that won't last long
- We have people using this software in academia, industry, and gov't
- Also some commercial distros are using bits:
  - TerraSoft PPC distro uses Bproc 3 and Supermon
- LANL is building more of these

Best and worst machines

- Worst machine:
  - One size fits all
  - We should know this: we didn’t all get 3XL jackets
- Best machine:
  - Best match to the app
  - Then buy different machines for different apps
- Speciation is here
- You can give a user a dedicated machine and reduce O&M costs

What guided us: General Goals

- To provide users with a single logon cluster:
  - That provides SSI capabilities
  - That scales to 1024 or more nodes
  - That is compatible with existing practice
  - With O(1) failure points instead of O(n)
  - With software that anticipates or tolerates failure
  - As a consequence: Single Point for management, console, and use

But … we also want:

- .01 FTE per 128 nodes (current is 1)
- 5 seconds to install all software on all nodes
- 5 seconds to upgrade
- 5 seconds to boot
  - Obviously, more work to do there
  - Hardware limits how much better we can make it
- Eliminate Ethernet and serial port networks
- Reduce all the significant measures by at least O(100)

Applications

- Only time will tell
- We have early candidates in BIO and Climate
- There are learning curve issues with this environment which will have a bearing on what gets run
- Even an app which might run well, may not get run if the programmer can’t/won’t adapt
- Which is to say: still not sure

Kollaboration is Key

- CCN-5,7,8
- NIS-3
- P-23
- B-2
- CCS-2
- CCS-3

External Academic Collaboration

- U. Maryland (DARPA secure boot)
- U. Texas (Compilers)
- U. Oregon (Performance tools)
- U. Washington (Languages)
- U. Monash (Debuggers)
- U. Calgary (Operating systems)

External DOE collaboration

- Sandia/Ca. (LinuxBIOS, bproc, Infiniband)
- Sandia/NM (LinuxBIOS)
- LLNL (LinuxBIOS, cluster tools)
- Argonne (LinuxBIOS, Supermon)

Industry Collaboration

- IBM
- AMD
- Intel
- Linux NetworX
- Linux Labs
- Suse
- Sis
- VIA
- Acer
- HP (possible)
- TerraSoft (PPC Linux)
- Mellanox
- SuperMicro
- Tyan
- Appro (K8)

Conclusion

- A year ago we showed you a toy Science Appliance
- This year it’s real
- We are meeting many of our admin goals
- System was built in a few days, came up immediately