

# Scalable Cluster Architectures and Software

Matt Leininger and Curtis Janssen

Sandia National Laboratories  
Livermore, CA  
Scalable Computing R&D Group

12 December 2005



# Possible Areas of Collaboration

- LinuxBIOS
- Performance Tools
- Parallel Graph Algorithms
- Scalable InfiniBand Cluster Architectures / OpenIB
- FPGA, MTA, ClearSpeed (Auxiliary Computing Devices)
- Programming Models
- Linux Kernel Enhancements
- Parallel PDE's / Sundance
- Sensitivity Analysis and Uncertainty Quantification
- Agent Based Modeling

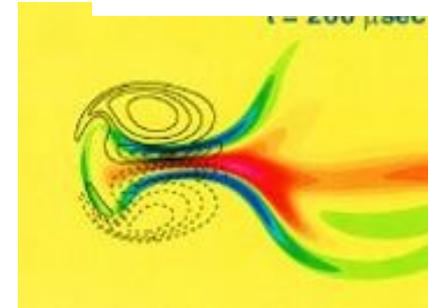
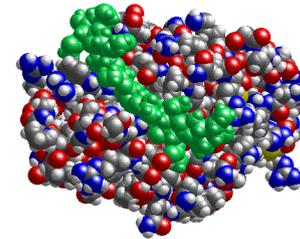
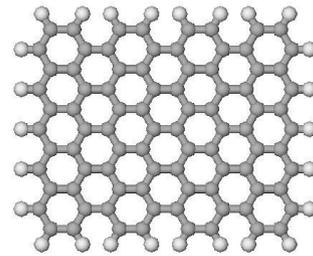


# Possible Areas of Collaboration

- LinuxBIOS
- Performance Tools
- Parallel Graph Algorithms
- Scalable InfiniBand Cluster Architectures / OpenIB
- FPGA, MTA, ClearSpeed (Auxiliary Computing Devices)
- Programming Models
- Linux Kernel Enhancements
- Parallel PDE's / Sundance
- Sensitivity Analysis and Uncertainty Quantification
- Agent Based Modeling

# Commodity HPC is Becoming Universal

- Financial
- Oil & Gas
- Pharmaceutical
- Combustion Science
- Homeland Security
- Nanotechnology
- Biotechnology
- Materials Science
- Engineering Sciences



- Leverage commodity multi-vendor solutions
- Move away from proprietary big-iron
- Commodity compute cycles are economic solutions
- Now need to scale out to 100's of Teraflops today and Petaflops soon
- **Sandia has a strong role in many of these areas for both software and hardware solutions**



# An interdisciplinary approach is used to maximize impact of research platforms

- All HPC elements are coupled to guide development of commodity technologies

HPC	Hardware	FPGAs, InfiniBand	Sandia
	Middleware	MPI	
	Applications	S3D, FLUQ, CTH MPQC, LAMMPS	
	Performance Tools, Benchmarks	VProf, mpiP, Vampir, ...	
	Production Computing	ICC Staff	



# Our interdisciplinary team seeks to maximize impact of research platforms

- Our unique approach is a tight vertical coupling of all HPC elements to guide development of commodity technologies
  - Hardware — Vendors provide much of this expertise. We are also investigating use of auxiliary computing devices such as FPGA's
  - Middleware — Have internal MPI developers as well as interactions with middleware developers at OSU, LANL, and IU.
  - Applications — Application developers are involved in configuration of testbeds, benchmarking, debugging
  - Performance tools — Understand coupling of app. with hardware
  - Production computing — Institutional computing staff involved
- LANL, LLNL, and other DOE labs are important partners



# LinuxBIOS

- Collaborate Ron Minnich and crew @ LANL
- LinuxBIOS summit @ LACSI this past October
- 50 attendees from Gov't Labs, AMD, Intel, Google, FSF, Tyan, LNXI, Appro, HP, SilverStorm
- AMD fully supporting LinuxBIOS (team ranges from 2-5)
- LinuxBIOS on Dell 1850 in progress (But will Dell support it?)
- Have chipset info for Intel E7520 and E7525, but Intel does not have LinuxBIOS developers
- Need LinuxBIOS RoundTable to take the next big step



# Parallel Tools

## Scalable Program Development Software

- Tools
  - Debuggers
  - Processor performance measurement
  - Message passing performance measurement
- Runtime
  - Message passing
  - Multi-threading
  - Memory layers



# Means

- Work with hardware vendors
- Support third party tool development
  - ASC PathForward contracts with Etnus and others
  - Other contracts
- Research and development
- Encouragement of standards support



# Issues

- Proprietary software is risky
  - Withdrawal of support possible and has happened
  - Ownership of KAI, Pallas, MSTI has changed
- Dependent on funding of Software PathForward projects to provide 3rd party software



# Debuggers

- TotalView is very popular and important to developers
  - Etnus training and one-on-one sessions with code teams
  - Development support by an ASC PathForward contract
- TotalView does not provide detection of memory areas until program state is damaged (SEGV) or not all (wrong answer)
- Memory access errors are some of the most time consuming bugs to track down
- Two approaches can be taken to provide memory debugging
  - Perform more run-time checks in a traditional debugging environment, such as TotalView (low overhead, low accuracy)
  - Do detailed load/store tracking to immediately detect invalid memory usage patterns (high overhead, high accuracy)



# Memory Use Debugging: Low Overhead, Low Accuracy

- Continuation of PathForward with Etnus
- Replace memory allocation primitives in executables
- Presents users with a familiar TotalView interface
- Can detect (with limitations):
  - Dangling pointers
  - Memory leaks



# Memory Use Debugging: High Overhead, High Accuracy

- Keeping track of individual loads and stores provides more complete information and captures errors at the precise instruction they occur
- Requires high CPU overhead ( $\sim x20$ )
- Requires high memory overhead ( $> x1.25$  to  $x2$ )
- Could not be easily implemented in TotalView—a new tool would have to be developed
- Contracting with Open-Works to port Valgrind to new architectures, and to make it MPI aware.



# Value of Performance Tuning

- Very pessimistic analysis of value of performance tuning:
  - Improve speed of app that uses 20% of RS by 10%
  - Effectively get more processing time. At \$1M/TFLOP saved \$800K on purchase price of new processors
  - Excludes maintenance costs
- At larger scales potential payoff is larger
- Performance analysis will be the only way to effectively utilize resources at very large scale
- It doesn't cost much to obtain 10% (or much more) savings on codes that have not been tuned before
- Difficult to estimate actual payoff since codes are in flux: as performance enhancements go into the code other changes occur as well



# MPI Performance Tools

- Two classes of tools—tracing and statistical
- MPI statistics—min, max, and mean times for MPI function calls for each MPI call site
  - Very low overhead, scales well
  - mpiP is an open-source solution (originated at LLNL)
  - We have ported mpiP to AMD64; Red Storm port in progress
- MPI tracing—collection, storage, and visualization of an application time line
  - Fairly low overhead, but scaling issues exists
  - Vampir is the most commonly used tool
  - Vampir developed by Pallas, which was bought by Intel



# Processor Performance Tools

- Again several approaches are used
- Instrumentation
  - Manual: Programmer inserted calls
  - Automatic: Compiler/preprocessor inserted calls
- Statistical
  - Running program is periodically interrupted and data is collected
  - Low overhead and no program modification is required
  - Certain types of information are difficult to collect this way—for example, object profiling



# Statistical Processor Profiling with VProf

- VProf is a statistical profiling tool developed at SNL
- Open-source, deployed on ICC, ASC White, and externally
- Have integrated mpiP with performance counter displays
- Migrating to SourceForge to better accommodate external contributors and gain access to SourceForge project management capabilities.



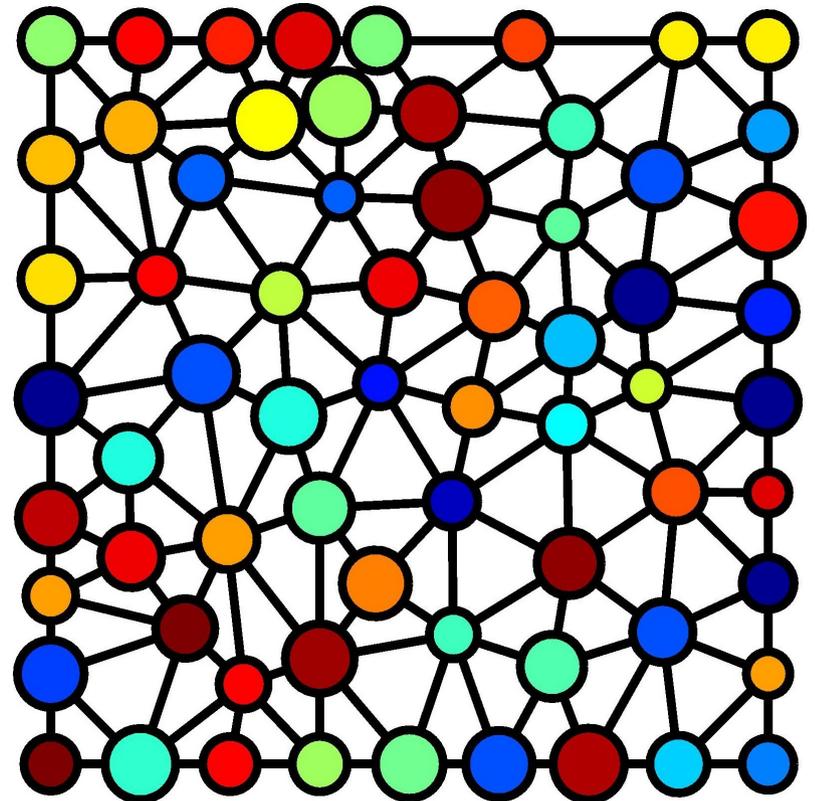
# Next Generation Performance Tool: OpenSpeedShop

- Open-source software PathForward contract with SGI
- Will provide data acquisition and presentation capabilities
  - Can dynamically attach to and instrument programs
  - No need to relink
- Modular architecture for future performance tool work
- Would allow display of many types of performance data side-by-side
- VProf work will be migrated to be compatible with OISS

# Parallel Graph Algorithms

## Why Graphs?

- Exemplar of memory-intensive application
- Widely applicable and can be very large scale
  - Scientific computing
    - sparse direct solvers
    - preconditioning
    - radiation transport
    - mesh generation
    - computational biology, etc.
  - Informatics
    - data-centric computing
    - encode entities & relationships
    - look for patterns or subgraphs

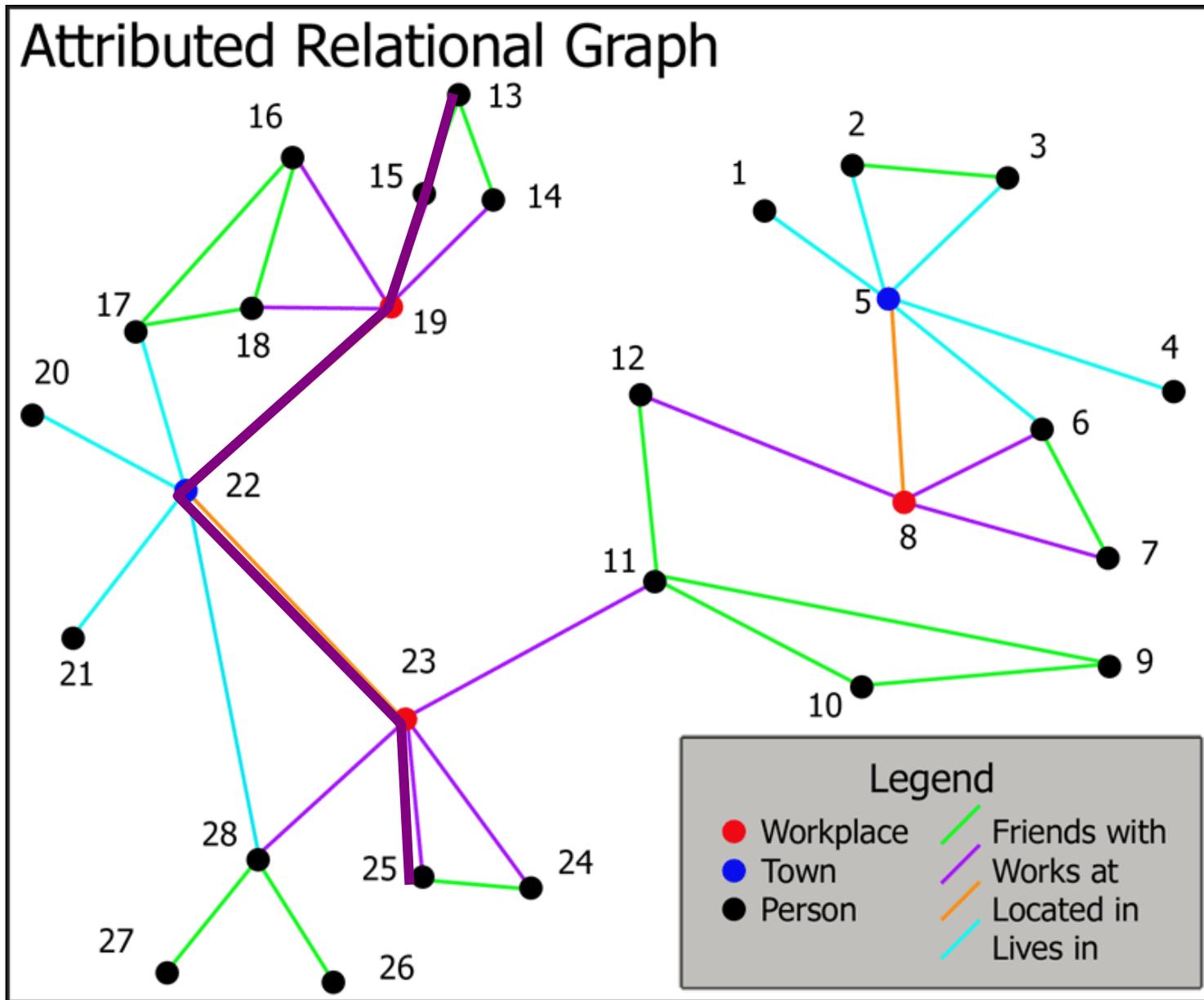




# Characteristics

- Data patterns
  - Moderately structured for scientific applications
    - Even unstructured grids make "nice" graphs
    - Good partitions, lots of locality on multiple scales
  - Highly unstructured for informatics
    - Similar to random, power-law networks
    - Can't be effectively partitioned
- Algorithm characteristics
  - Typically, follow links of edges
    - Maybe many at once - high level of concurrency
    - **Highly memory intensive**
      - Random accesses to global memory - small fetches
      - Next access depends on current one
      - Minimal computation

# Shortest Path Illustration

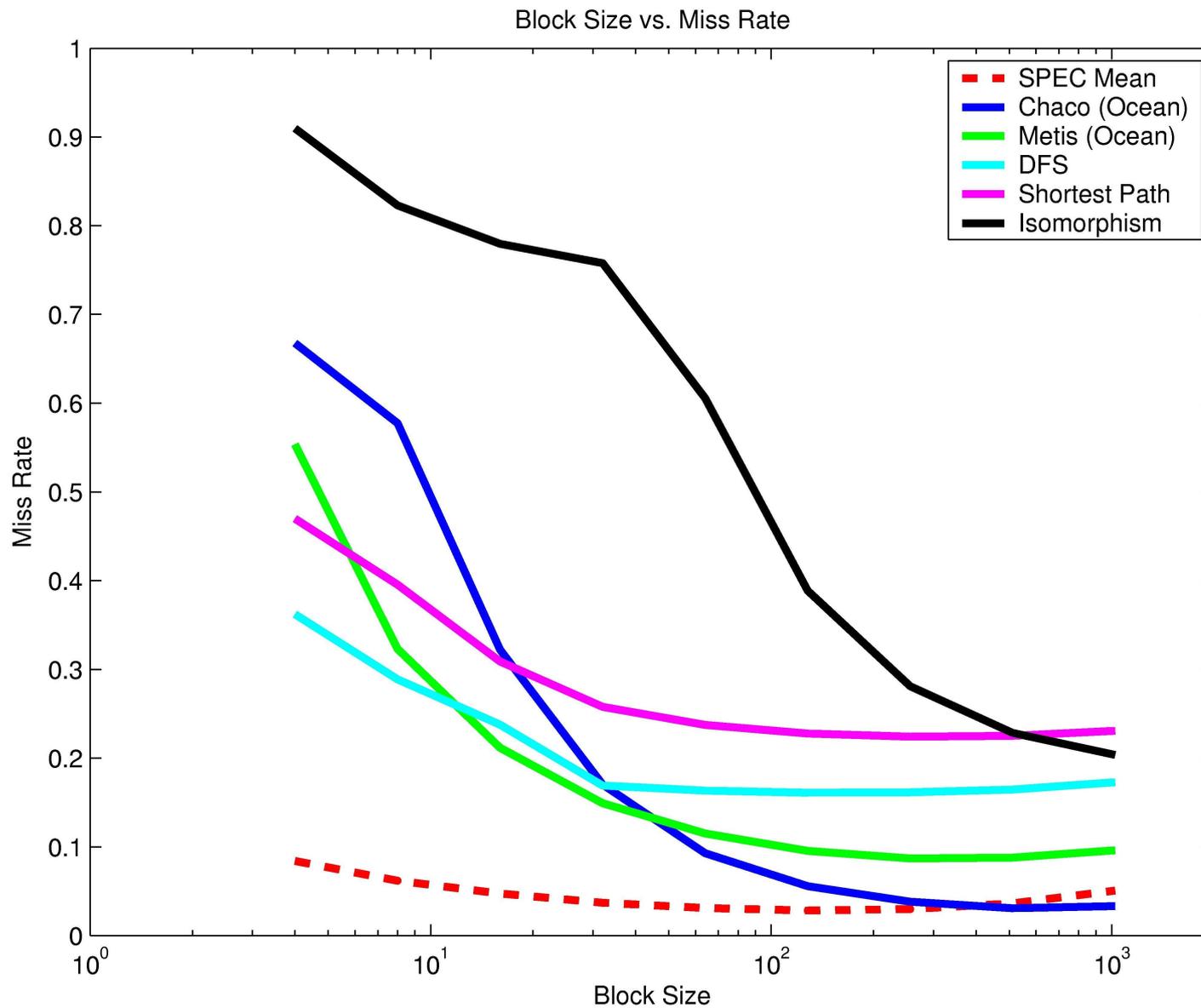




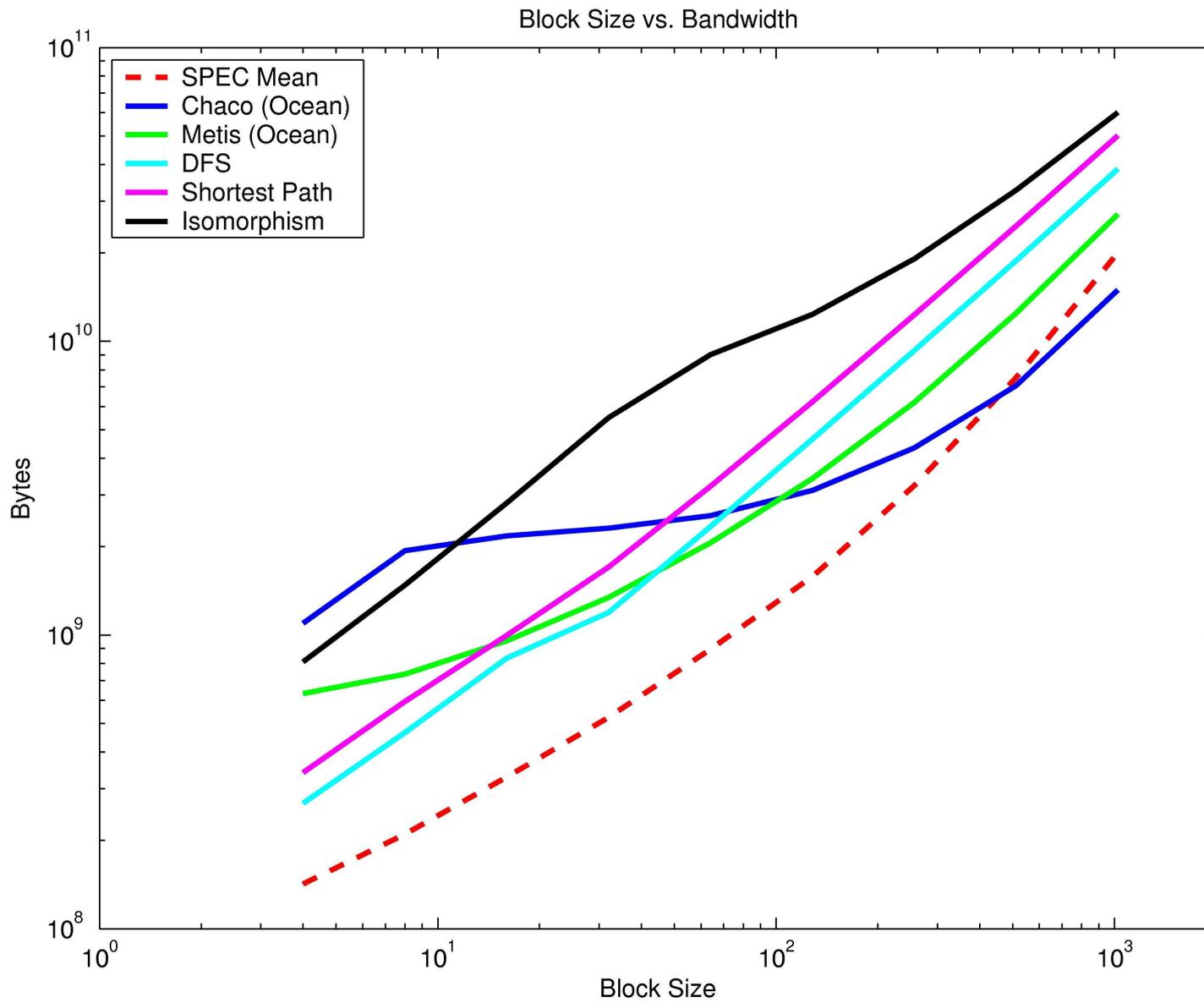
# Architectural Challenges

- Runtime is dominated by latency
- Essential no computation to hide memory costs
- Access pattern is data dependent
  - Prefetching unlikely to help
  - Often only want small part of cache line
- Potentially abysmal locality at **all** levels of memory hierarchy

# Caching Futility



# Larger Blocks are Expensive



# Properties Needed for Good Graph Performance

- Low latency / high bandwidth
  - For small messages!
- Latency tolerant
- Light-weight synchronization mechanisms
- Global address space
  - No graph partitioning required
  - Avoid memory-consuming profusion of ghost
- **These describe Burton Smith's MTA!**





# MTA Introduction

- Latency tolerance via massive multi-threading
  - Each processor has hardware support for 128 threads
  - Context switch in a single tick
  - Global address space, hashed to reduce hot-spots
  - No cache. Context switch on memory request.
  - Multiple outstanding loads
- Good match for applications which:
  - Exhibit complex memory access patterns
  - Aren't computationally intensive (slow clock)
  - Have lots of fine-grained parallelism
- Programming model
  - Serial code with parallelization directives
  - Code is cleaner than MPI, but quite subtle
  - Support for “future” based parallelism

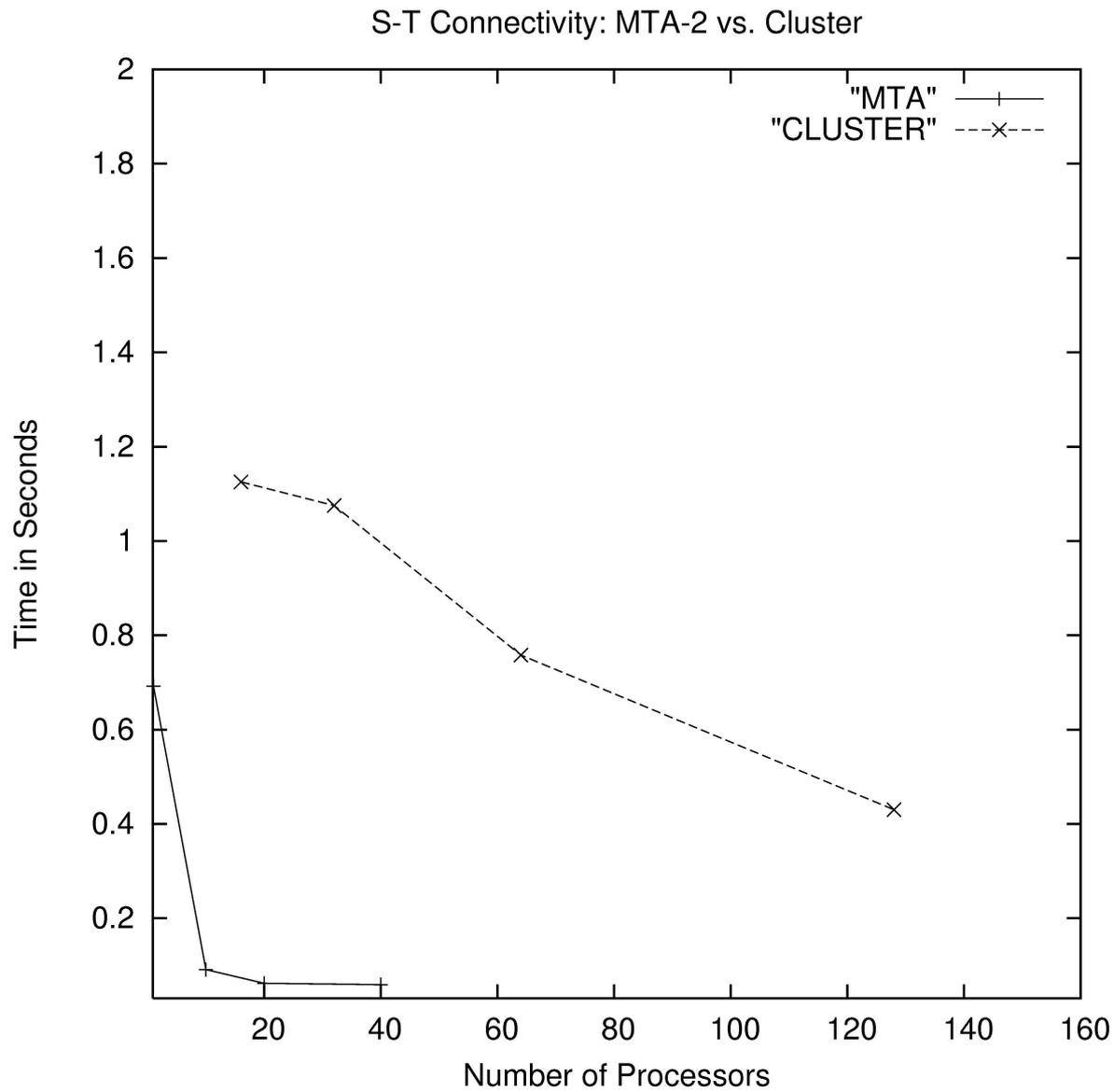




# Case Study – Shortest Path

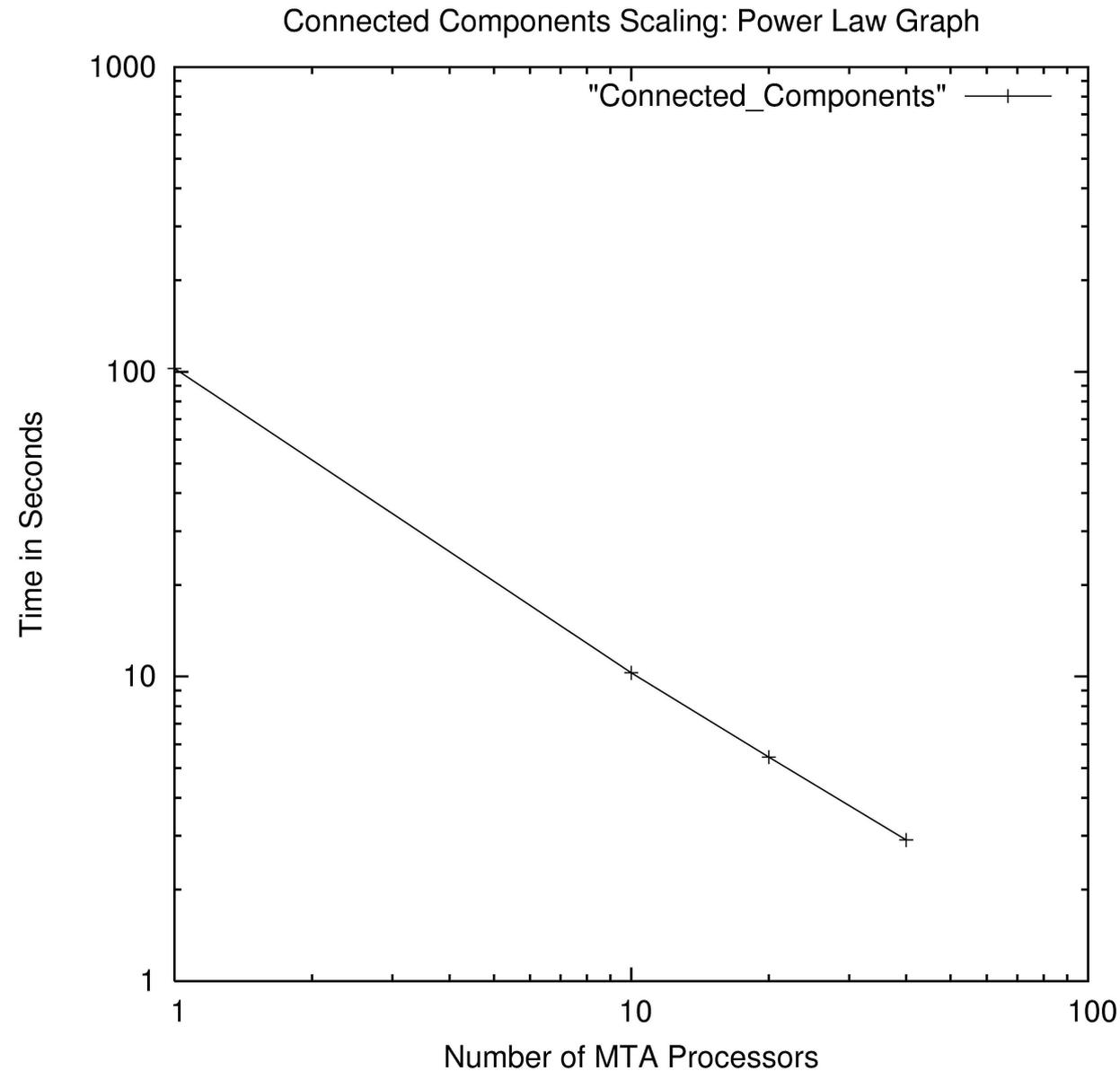
- Compare codes optimized for different architectures
- Option 1: Distributed Memory CompNets
  - Run on Linux cluster: 3GHz Xeons, Myrinet network
  - LLNL/SNL collaboration – **just for short path finding**
  - Finalist for Gordon Bell Prize on BlueGene/L
  - About 1100 lines of C code
- Option 2: MTA parallelization
  - **Part of general-purpose graph infrastructure**
  - About 400 lines of C++ code

# Short Paths on Erdos-Renyi Random Graphs ( $V=32M$ , $E=128M$ )



# Connected Components on MTA-2 Power-Law Graph $V=34M$ , $E=235M$

procs	time
1	102.7
10	10.29
20	5.44
40	2.91





# Remarks

- Single processor MTA competitive with current micros, despite 10x clock difference
- Excellent parallel scalability for MTA on range of graph problems
  - Identical to single processor code
- Eldorado is coming next year
  - Hybrid of MTA & Red Storm
  - Less well balanced, but affordable



# Broader Lessons

- Space of important apps is broader than PDE solvers
  - Data-centric applications may be quite different from traditional scientific simulations
- Architectural diversity is important
  - No single architecture can do everything well
- As memory wall gets steeper, latency tolerance will be essential for more and more applications
- High level of concurrency requires
  - Latency tolerance
  - Fine-grained synchronization

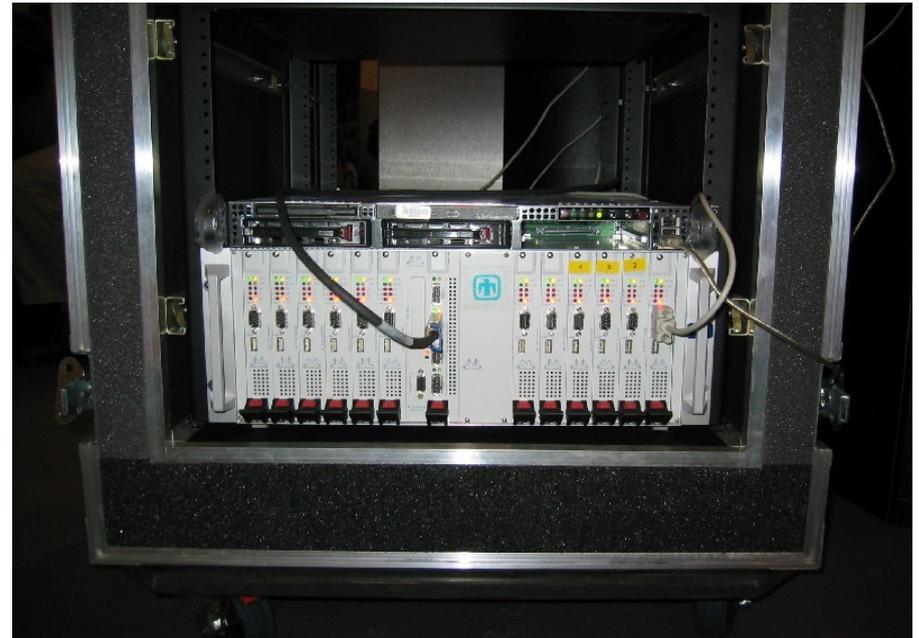


**Sandia's History of InfiniBand:**

**Scalable InfiniBand Cluster  
Architectures**

# History of InfiniBand Evaluation at Sandia

- Sandia has been evaluating IB since 2001 (early 1X equipment)
- Mellanox Nitro 1 (1X) and Nitro 2 (4X) blade systems
- Dell 1650 16 nodes, 1X Intel, 4X IBM, 4X Mellanox, Myrinet 2000
- Funded MVAPICH development
- 128 node Testbed in 2003 expanded to 220 nodes in 2004
  - Dual Intel Xeon, 1.076 TFlops (1.57 TFlops theoretical, 111 Top500)





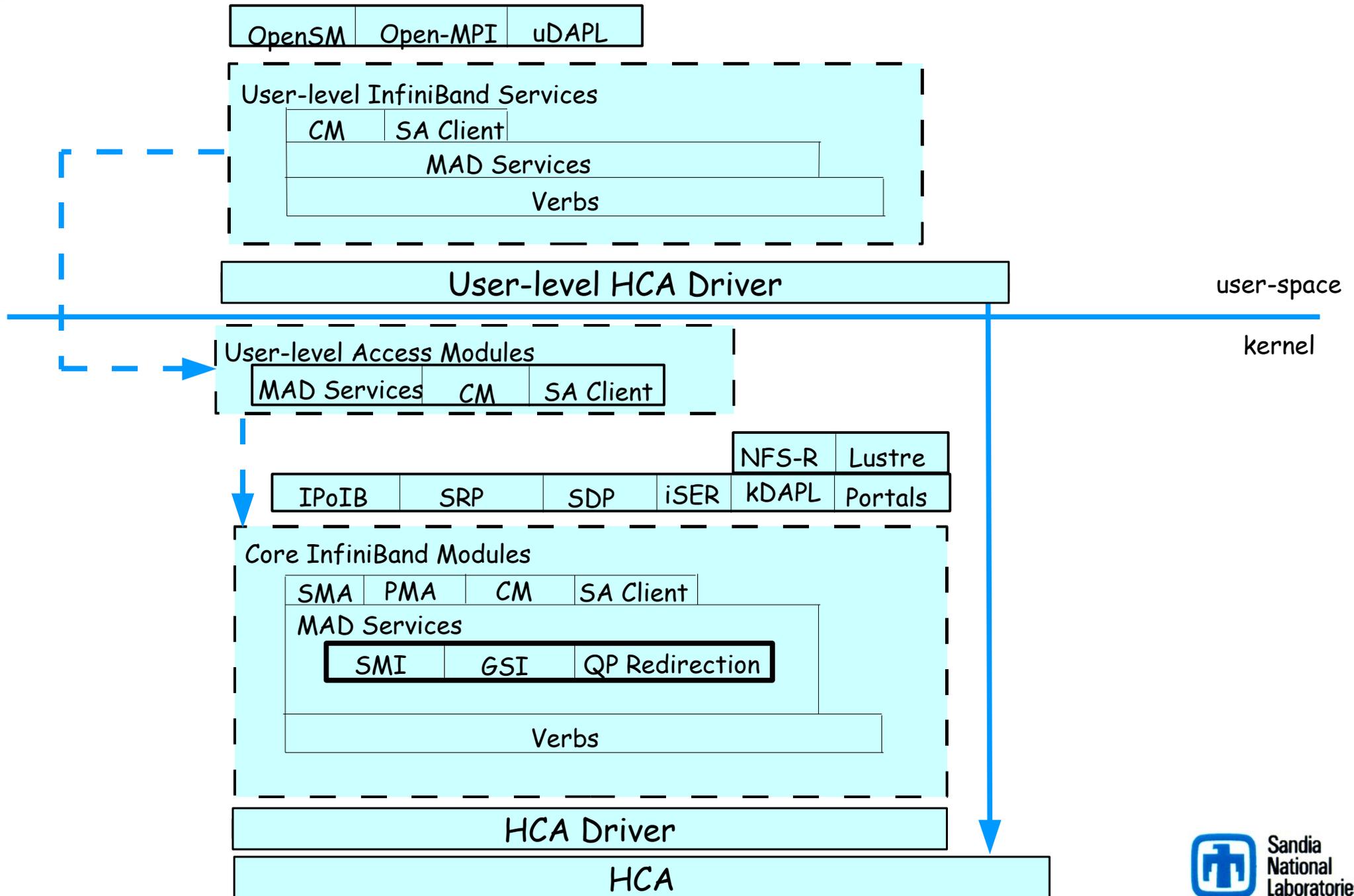
# United HPC and Industry by Forming OpenIB Alliance

- Tri-labs are founding members of the OpenIB Alliance
- Align Industry behind a single Open Source IB SW stack that meets the needs for HPC, data center, and scalable I/O
- Lead to DoE (ASC program) funds part of the work on the OpenIB stack that is focused on HPC - Voltaire, Cisco, SilverStorm, and Intel
- OpenIB currently has 23 members and continuing to expand
- OpenIB kernel space flows to kernel.org and then to Linux distributions
- OpenIB user-space flows to Linux distributions

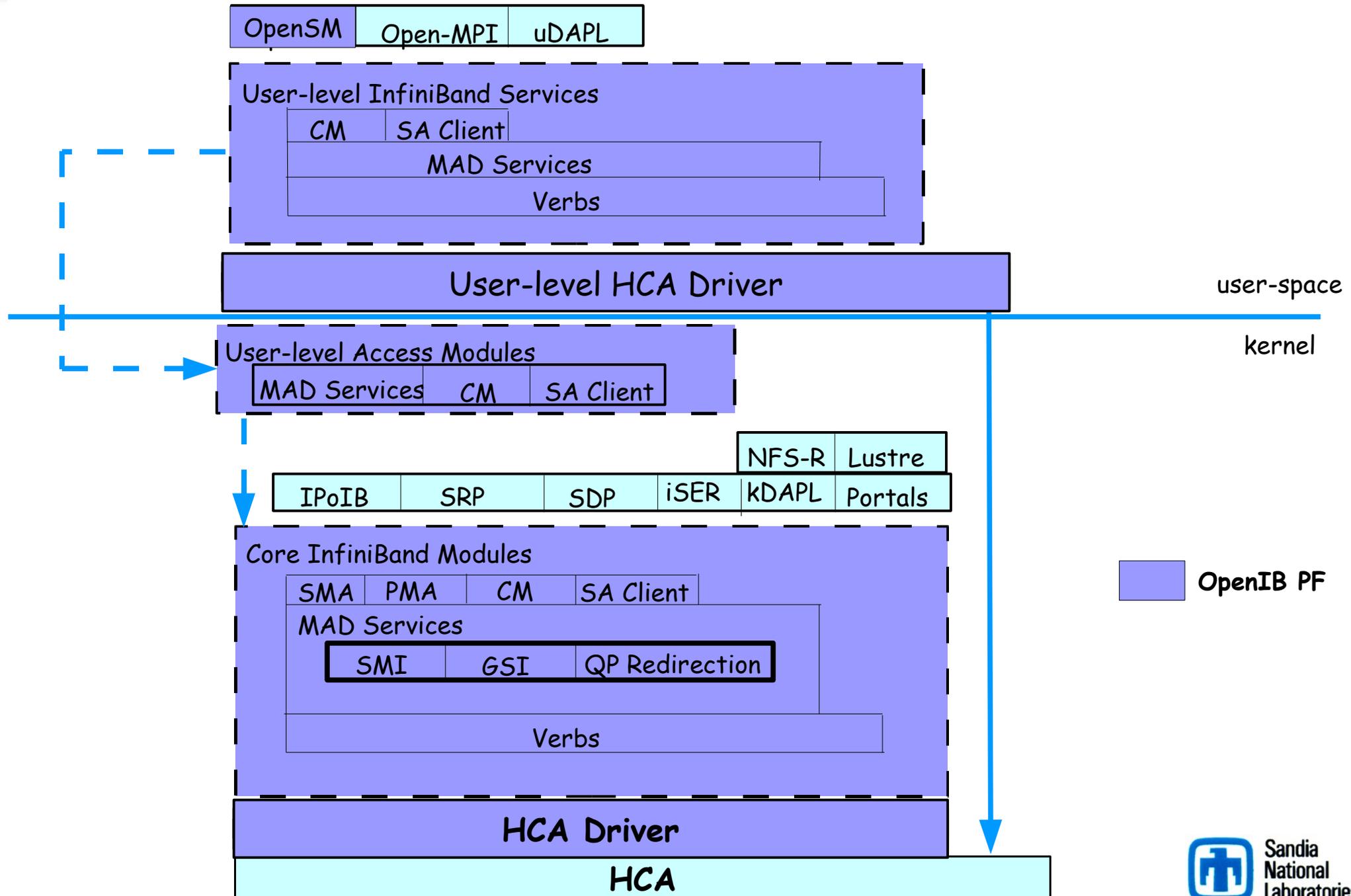
[www.openib.org](http://www.openib.org)



# OpenIB Stack Architecture



# OpenIB Stack Architecture



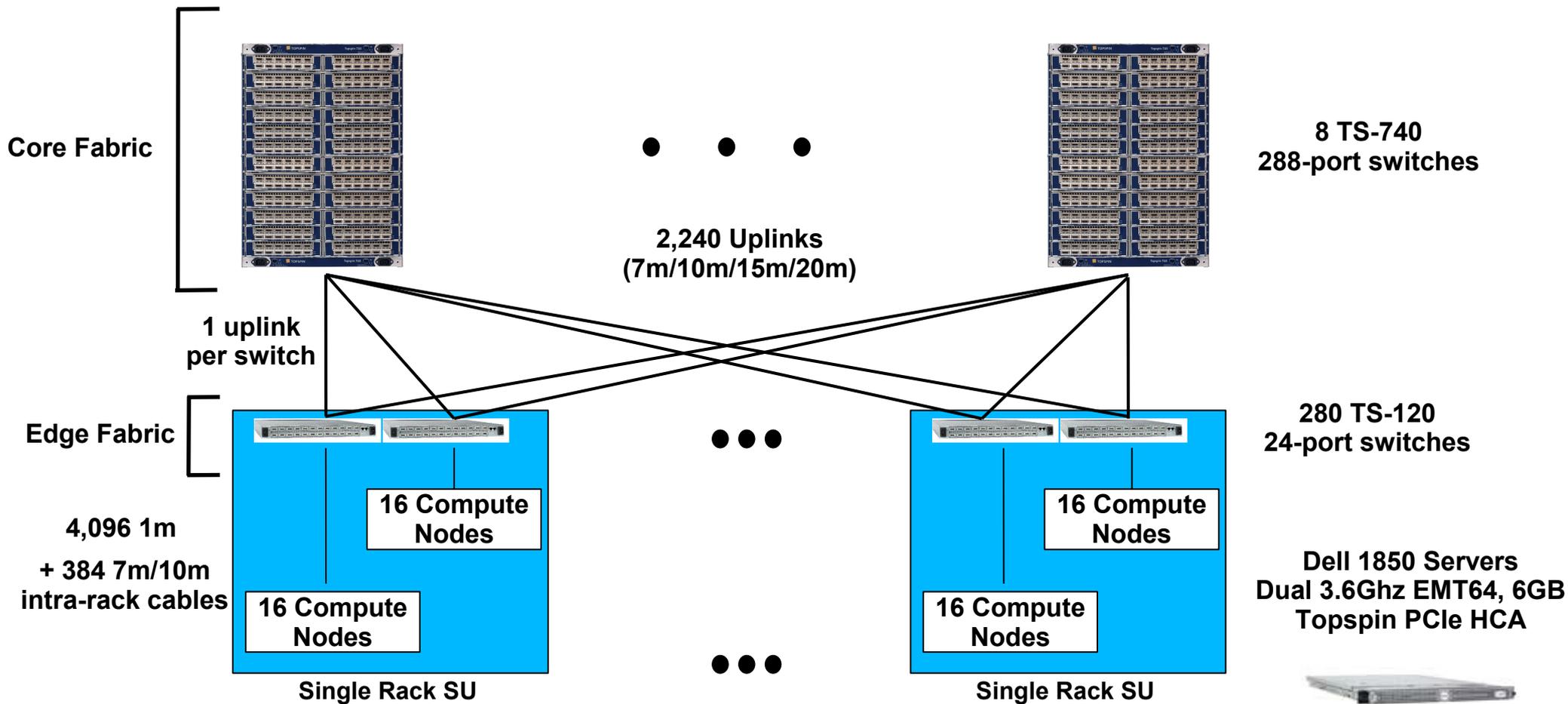


# Cluster computing strategy must include and learn from research vehicles

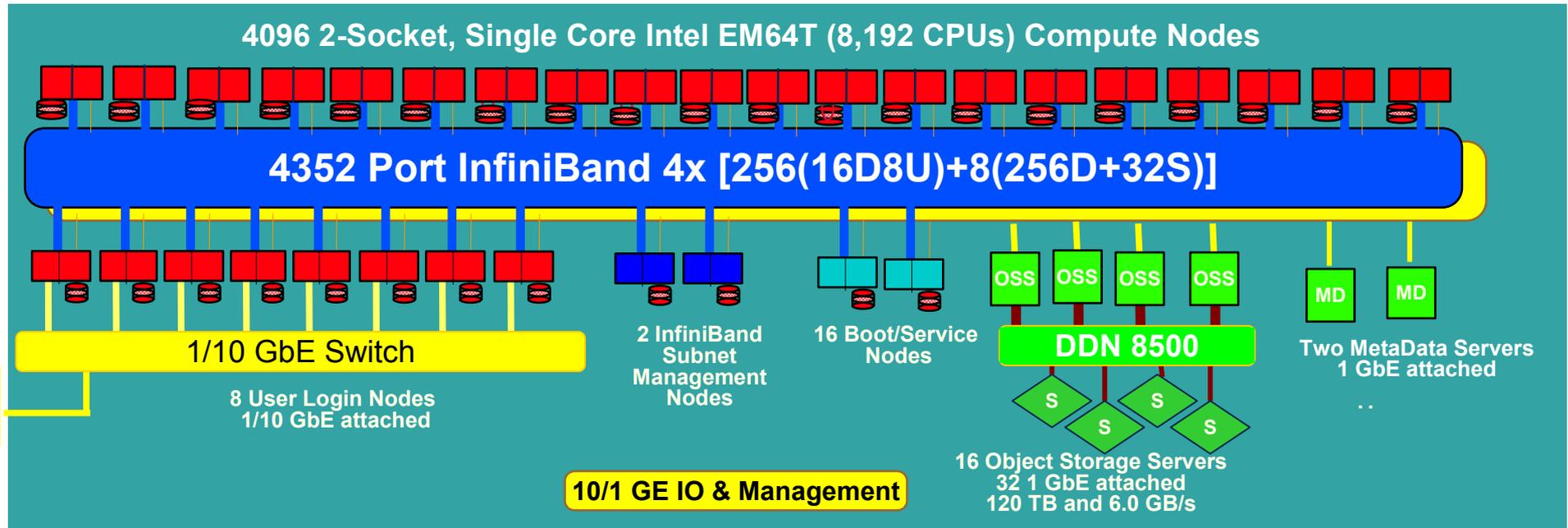
- Off-the-shelf technology doesn't appear from nowhere
  - Early adopters must try, benchmark, and guide development of prospective commodity technologies to ensure readiness for HPC
- Testbed machines are needed to develop and exercise the technology
  - Scalability testing requires machines of significant size
  - "Friendly users" provide feedback while getting much time on many nodes—enabling them to do interesting new science along the way
- Once the testbed stabilizes the technology is available
  - New machines can be purchased with confidence
  - Now stable testbed can be dedicated to "friendly users" full time

# Sandia Thunderbird Cluster

8,960 Processor, 64.5TF/s



# Sandia Thunderbird Architecture

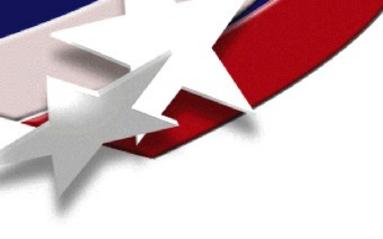


## System Parameters

- 14.4 GF/s dual socket 3.6 GHz single core Intel SMP nodes with 6.4 GB/s memory BW, DDR-2 400 SDRAM (memory B:F=0.42, BW B:F=0.44)
- ~4.0  $\mu$ s, 1.8 GB/s MPI latency and Bandwidth over 4X InfiniBand
- Archive support - Support 800 MB/s transfers to Archive over quad Jumbo Frame Gb-Enet and IB links from each Login node. Some solutions have 10GbE
- Local disk for swap and root partitions, fallback OS image (300 TB total)
- Remote/Network boot
- Serial over LAN
- Lustre storage ~120 TB capacity and 6.0 GB/s BW
- Disk Capacity 20 B:F = 300 TB local file system in multiple RAID5
- IO Bandwidth 0.001 B:F = 20 GB/s delivered parallel I/O performance

**Increases institutional capacity computing from ~ 24 Tflop/s to 84 TFlop/s.**

128 node Dell 1850 + Cisco Testbed at Sandia/CA



## Take Scalable Clusters to the Next Level By Working With Other Communities

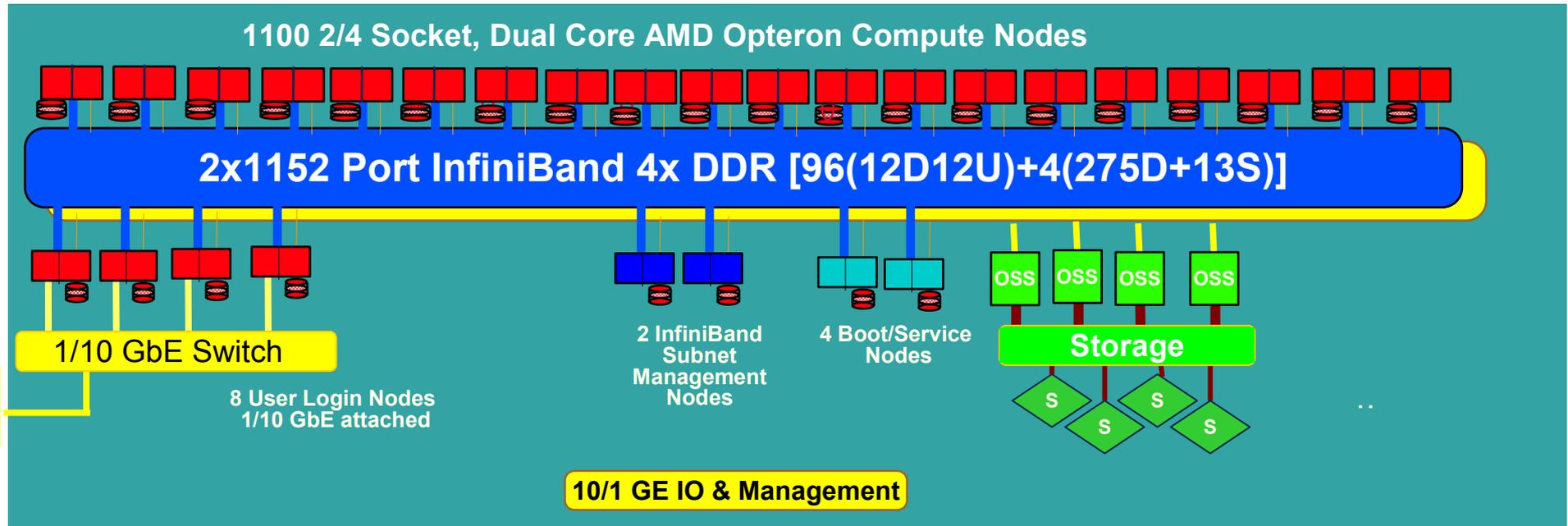
- Bring “customers” together to push limits of technology
- DoE HPC requirements overlap well with industries computing requirements
- HPC is moving away from proprietary systems to commodity solutions
- Commodity has reduced cost and provided more compute cycles
- Need to “scalable solutions” to transform industries
  - Financial, Oil&Gas, Pharma, etc.
- DoE has a history of building large clusters (IB, Myrinet, Quadrics)
  - Sandia: Thunderbird (4500), Cplant (1700), ICC (800), NWCC (1200), ...
  - LANL: Pink (1024), Lightning (2000+), Gordan (1000+), Flash (1000+), ...
  - LLNL: MCR (1152), Thunder (1024), Lilac (768), ALC (960), ...
- Scalability, Reliability, Performance, and Manageability are the key requirements for scaling out



# What barriers Need to be Overcome in Order for IB Architectures to Transform Industry Computing?

- Barriers are software, software, and software
  - IB HW has a rich set of features (IPC, Storage, Management, Congestion Control, etc.)
  - Need open source software to take full advantage of multiple vendors
  - Require storage I/O protocols that can be used for IB or Ethernet (iSCSI/iSER)
- DoE has struggle with past clusters (IB and non-IB) due to:
  - Immature software
  - New SW bugs typically occur at every doubling of cluster size
  - All large clusters are for "production work" not R&D
  - SW issues end up being solved by SW hacks, quick work arounds, or not fixed at all
  - The short term need for compute cycles overrides any R&D that could fix issues
- R&D required for longer term solutions that provide effective scaling
- Solution: Large-scale InfiniBand Testbed (~1000 nodes) solely dedicated to solving scalability, reliability, performance, and manageability at large-scale
  - Also provides large-scale testbed for application profiling, development, and testing
- DoE came to this conclusion several years ago.

# Sandia Thunderbird Architecture



## System Parameters

- 20.8 – 41.6 GF/s per node 8/16/32 GB RAM
- < 2.0  $\mu$ s, 1.8-3.6 GB/s MPI latency and Bandwidth over 4X DDR InfiniBand
- Diskless compute nodes boot over IB
- LinuxBIOS
- Mellanox Hermon HCA + PathScale HCA
- Storage – Panasas or Lustre?
- OpenIB gen2

## 23-46 Tflop InfiniBand Testbed

IB HW, OpenIB, Storage, Gateways,  
Applications, multi-vendor diagnostics

Moore's Law is slowing down leading to new computational barriers that will drive increased parallelism at the CPU, node, and network levels



# Extra Slides

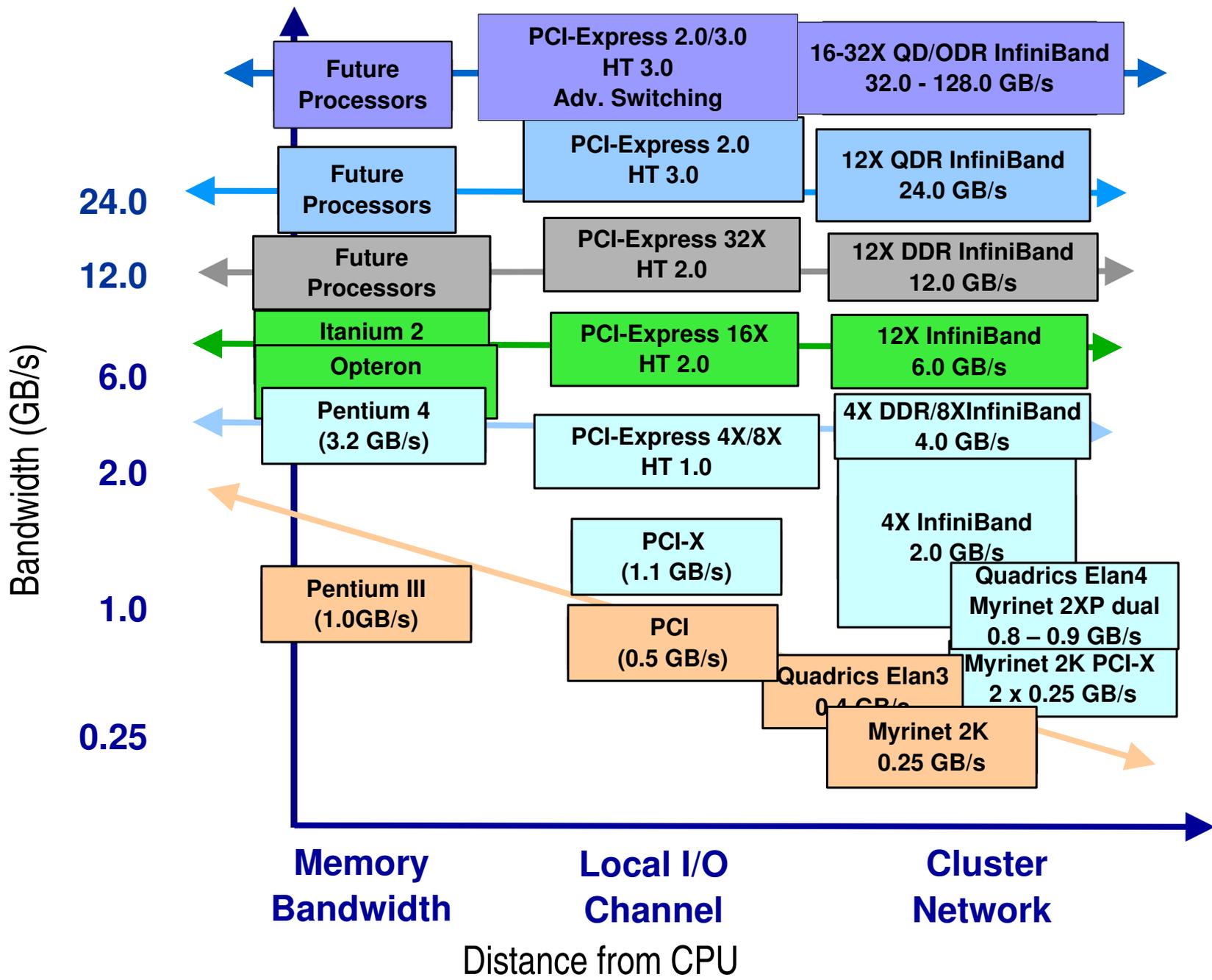
from Cluster 2005 Symposium and Panel



# What barriers need to be overcome in order for IB Architectures to transform industry computing?

- Ethernet will always be used in some HPC environments
- The issues facing the high-end HPC community are latency and affordable bandwidth
- IB is still less expensive and better performance than 10 GbE
- When 10 GbE is ubiquitous IB will have 12X DDR/QDR
- Longer term goal is to push for 24/32X QDR/ODR IB
- Barriers are software, software, software
  - Require storage I/O protocols that can be used for IB or Ethernet (iSCSI/iSER)
  - Common Linux RDMA software stack for IB/iWarp in mainline kernel (kernel.org)
- Switch and host side adaptive routing for improved scalability
- OpenIB

# InfiniBand Roadmap Tracks Future Processor and I/O Performance





## Will solutions support the trend towards higher radix switches and cross-section BW with a scalable cost model?

- The trend of building over 1k node clusters is moving outside of typical HPC centers
- High radix switches move cost effective solutions to larger node counts
  - Linear cost to go up to 288 nodes (and beyond with future IB switch silicon)
- Fat-tree high radix switches and network topologies
  - Managable up to 4k nodes
  - Beyond single high radix switch 3D mesh/torus topologies will likely provide better cost effective solutions



# What fraction of today's HPC workloads need the BW that IB offers?

- Worrying about just today's applications is a certain recipe for disaster
- BW to Flops ratios between 0.05 to  $<1.0$  for capacity to capability platforms
- ASC apps today:
  - On large SMP will outpace the current IB 4X DDR (4.0 GB/s) technology
  - Dual core CPUs increase network BW requirements
- ASC apps future:
  - Multi-core CPUs will outpace current IB roadmap leading to multi-rail systems
- ASC workloads sensitive to network latency and MPI collective scalability
- Simulations requirements are for 100s-1000s of high res 2D and low res 3D runs and 10s of medium to high res 3D runs
- Algorithms incorporating more complex physics
  - non-local/global effects which lead to more stress on the network and MPI collective operations
  - Longer term: modern algorithms need to be latency tolerant (multi-threaded)
- A large fraction of our workloads require high performance interconnects



# Will IB enable cluster solutions to have a single network for compute and storage?

- Current ASC clusters have at least two networks
  - One for compute (MPI)
  - Others for management and storage
- Future ASC clusters
  - Sandia and LANL are working towards single network
  - Use IB network for booting, compute, I/O (parallel and NFS), and management
  - Utilize IB-to-GbE in vendors switches
  - Push storage vendors to move in the IB direction
  - NFS/RDMA, iSER, iSCSI, and QoS are required to make this a reality



## How will IB deployment/reliability be accomplished in a price friendly manner for clusters with 1000's of nodes?

- Copper and optical cables have reliability problems
  - Copper difficult to install and maintain
  - Optical has had laser problems in the past
- Optical is the right answer as we move to 12X IB
  - needs to be more cost effective ~\$50 more per link or less
- Use 12X IB cables for switch to switch links
  - 12X physical uplinks could be configured as three 4X links to eliminate store and forward
- Increasing single rate to DDR/QDR and beyond pose engineering challenges for every system component included cabling
- Topologies other than fat-tree: 3D torus/mesh



## Do approaches that adopt the IB physical layer but use custom messaging layers portend changes to the IB standard?

- Open standards for high performance solutions are preferred, but standards need to evolve as new technologies or requirements are discovered
- Expect to see changes in the IB spec/hardware to meet more HPC req.
  - Improved UD performance
  - Reliable multicast
  - ~1 us latency
  - Improved BW/latency for small to medium sized messages
  - MPI collectives or primitives in HW
  - InfiniBand can learn from the success/failures of other HPC interconnects
  - Expand LID space
  - Expand number of service levels
- Software must remain the same/backward compatible, open source, and open development



# Parallel PDE's with Sundance

- Sandia Sundance PDE package
- Advantages of MPI w/o having to code to MPI
- Code PDE's in formula representation and Sundance will parallelize
- Finite element based



# Sensitivity Analysis and Uncertainty Quantification

- Monica



# Auxiliary Computing Devices

- Application accelerator
- Why use custom architectures - start with working cluster and add accelerators
- Devices connected to PCI-Express, HT, or PCI-X
- FPGAs
- DSPs
- ClearSpeed
- Cray MTA



# Agent Based Modeling

- Need info here



# Linux Kernel Enhancements

- Linux synchronized kernel scheduler
- Dynamic co-scheduler
- Other



# Programming Models

- Sockets
- MPI
- Threading
- Other



# What tools do we need for management of large-scale IB clusters?

- We expect to see many of the same tools that are currently used in the Ethernet world
- The ASC OpenIB PathForward is funding a good deal of work in this area
- OpenIB PathForward tools/diagnostics
  - User space firmware flash
  - HCA self tests
  - IB network diagnostics for reporting and tracking port errors, bad cables, etc.
  - Ibping, ibstatus, ibroute, ibtracetr, ibnetdiscover, smpquery, perfquery, ibnetverify, etc.
  - See OpenIB code repository at [www.openib.org](http://www.openib.org)
  - OpenIB tools are command line - working with vendors for visual tools
- The tools need to be open source, open development, and support multiple vendor solutions. Submit tools to OpenIB community!



## How would the development of a new I/O specification effect the progress of IB deployment?

- If new specification is well thought out then it may have a positive impact
- If not, then it might be ignored as other past techs/specs have been
- FWIW
  - IB has learned many lessons about interacting with OSS community
  - Lessons from IBAL, OpenIB, etc.
  - Open Source community does **not** like specs
  - Why not develop new I/O code base off OpenIB and submit it to the entire OSS community for review before writing the spec?
  - This would generate discussion, feedback, and generate stronger ties to OSS community
  - Remember in open source open development the code is the spec