Oak Ridge National Laboratory

Lustre Scalability Workshop

Presented by:
Galen M. Shipman

Collaborators:
David Dillow
Sarp Oral
Feiyi Wang

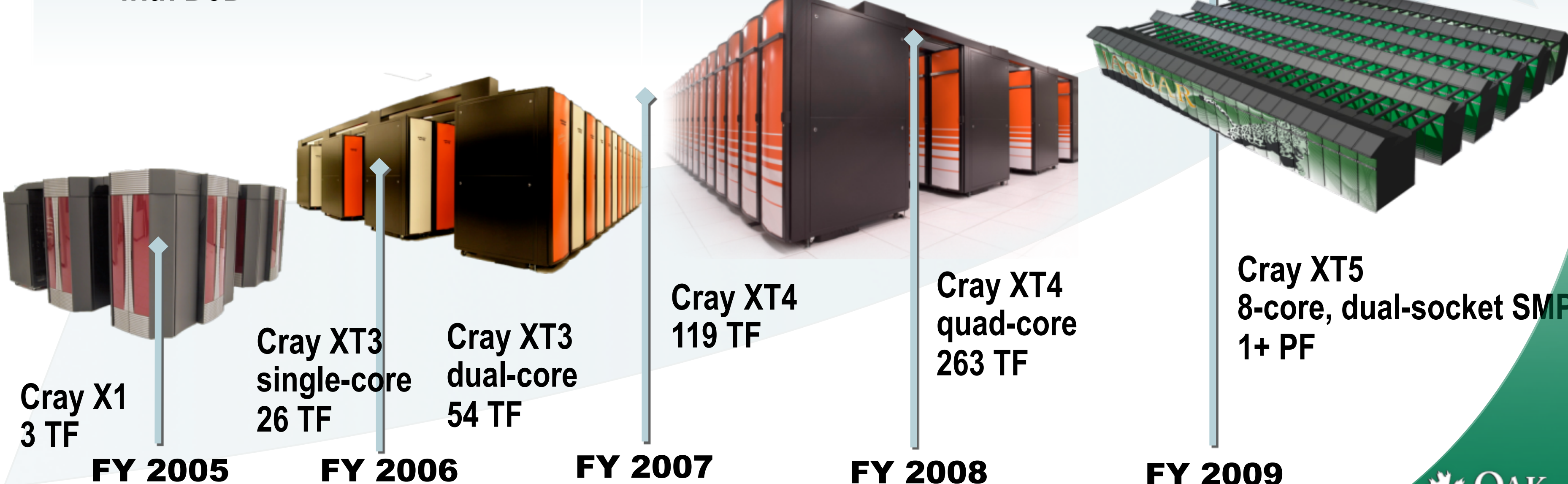
February 10, 2009

Managed by UT-Battelle for the
U. S. Department of Energy

We have increased system performance 300 times since 2004

Hardware scaled from single-core through dual-core to quad-core and dual-socket SMP nodes	Scaling applications and system software is the biggest challenge

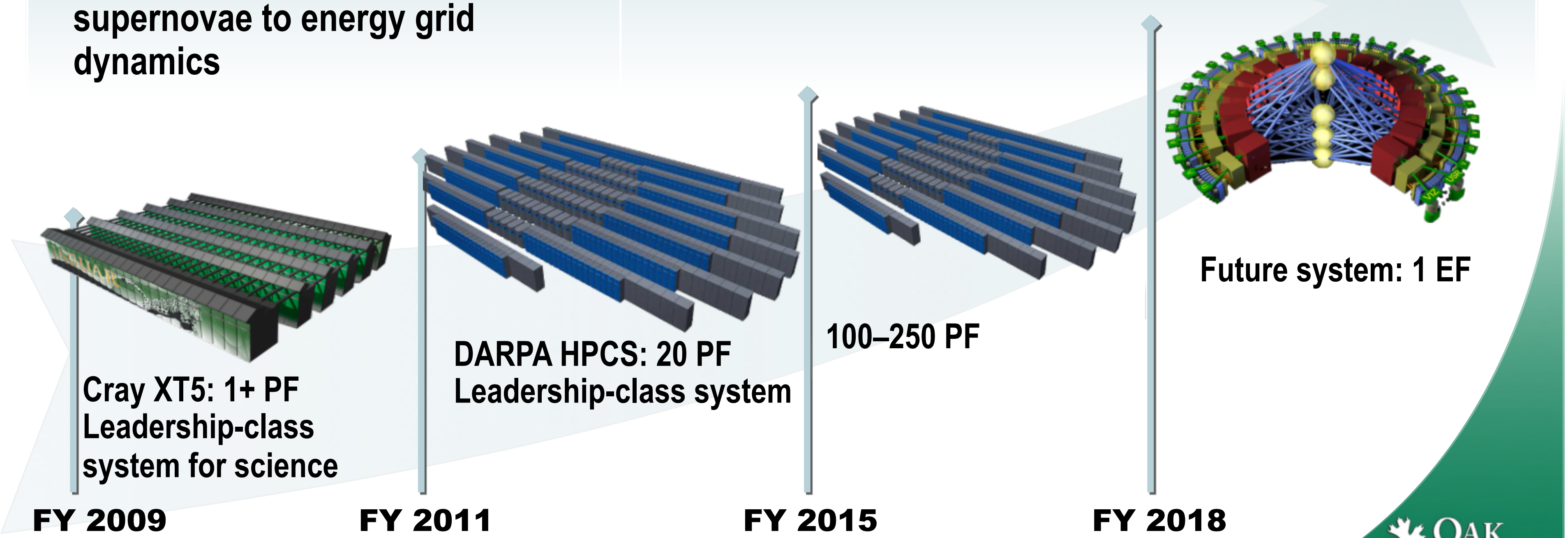
• NNSA and DoD have funded much of the basic system architecture research
  • Cray XT based on Sandia Red Storm
  • IBM BG designed with Livermore
  • Cray X1 designed in collaboration with DoD
	• SciDAC program is funding scalable application work that has advanced many science applications
	• DOE-SC and NSF have funded much of the library and applied math as well as tools
	• Computational liaisons are key to using deployed systems



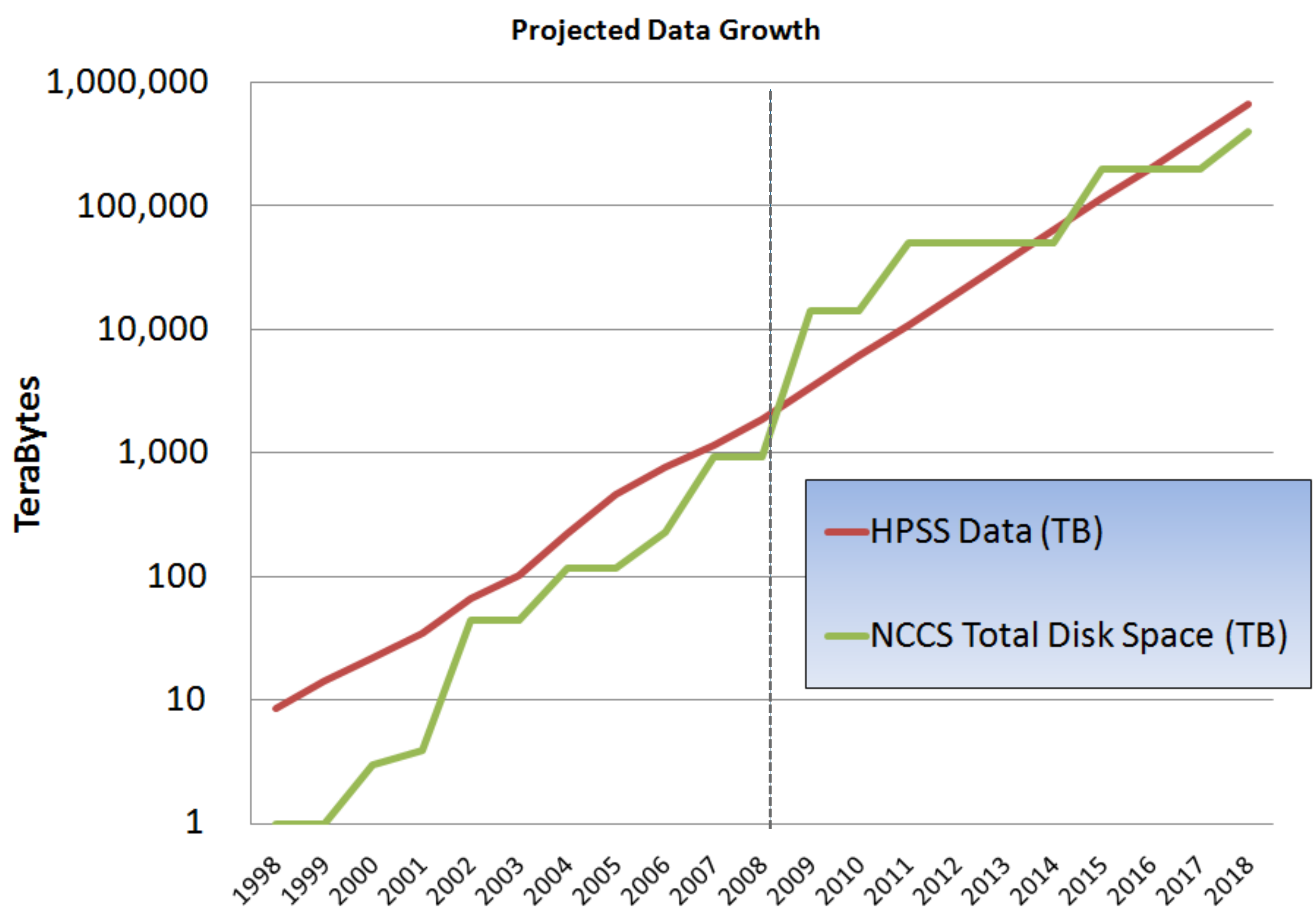
OAK RIDGE National Laboratory

We will advance computational capability by 1000× over the next decade

Mission: Deploy and operate the computational resources required to tackle global challenges	Vision: Maximize scientific productivity and progress on the largest-scale computational problems
• Deliver transforming discoveries in materials, biology, climate, energy technologies, etc. • Ability to investigate otherwise inaccessible systems, from supernovae to energy grid dynamics	• Providing world-class computational resources and specialized services for the most computationally intensive problems • Providing stable hardware/software path of increasing scale to maximize productive applications development



Explosive Data Growth



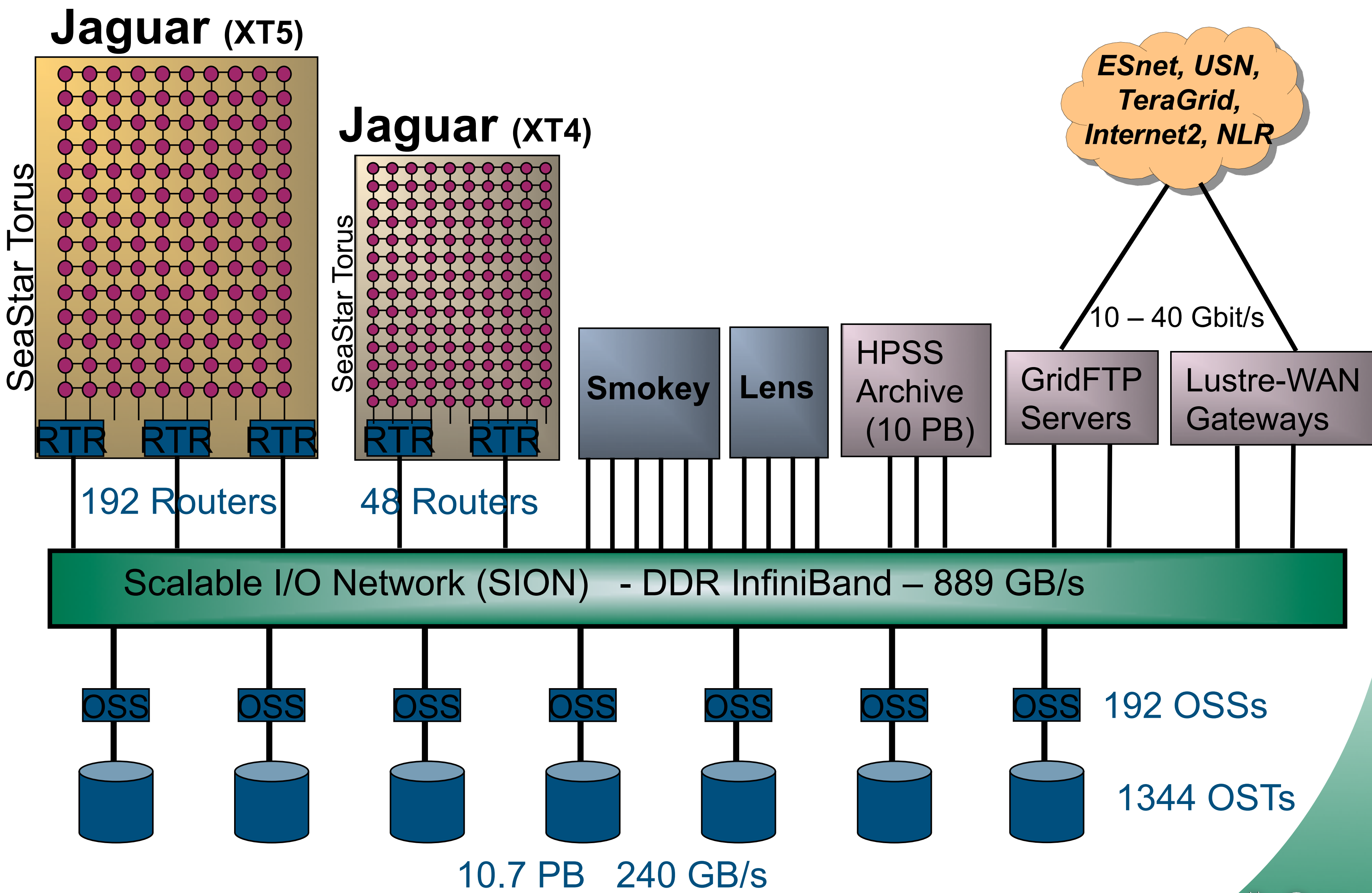
Parallel File Systems in the 21st Century

- Lessons learned from deploying a Peta-scale I/O infrastructure
- Storage system hardware trends
- File system requirements for 2012

The Spider Parallel File System

- ORNL has successfully deployed a direct attached parallel file system for the Jaguar XT5 simulation platform
  - Over 240 GB/sec of raw bandwidth
  - Over 10 Petabytes of aggregate storage
  - Demonstrated file system level bandwidth of >200 GB/sec (more optimizations to come)
- Work is ongoing to deploy this file system in a router attached configuration
  - Services multiple compute resources
  - Eliminates islands of data
  - Maximizes the impact of storage investment
  - Enhances manageability
  - Demonstrated on Jaguar XT5 using ½ of available storage (96 routers)

Spider

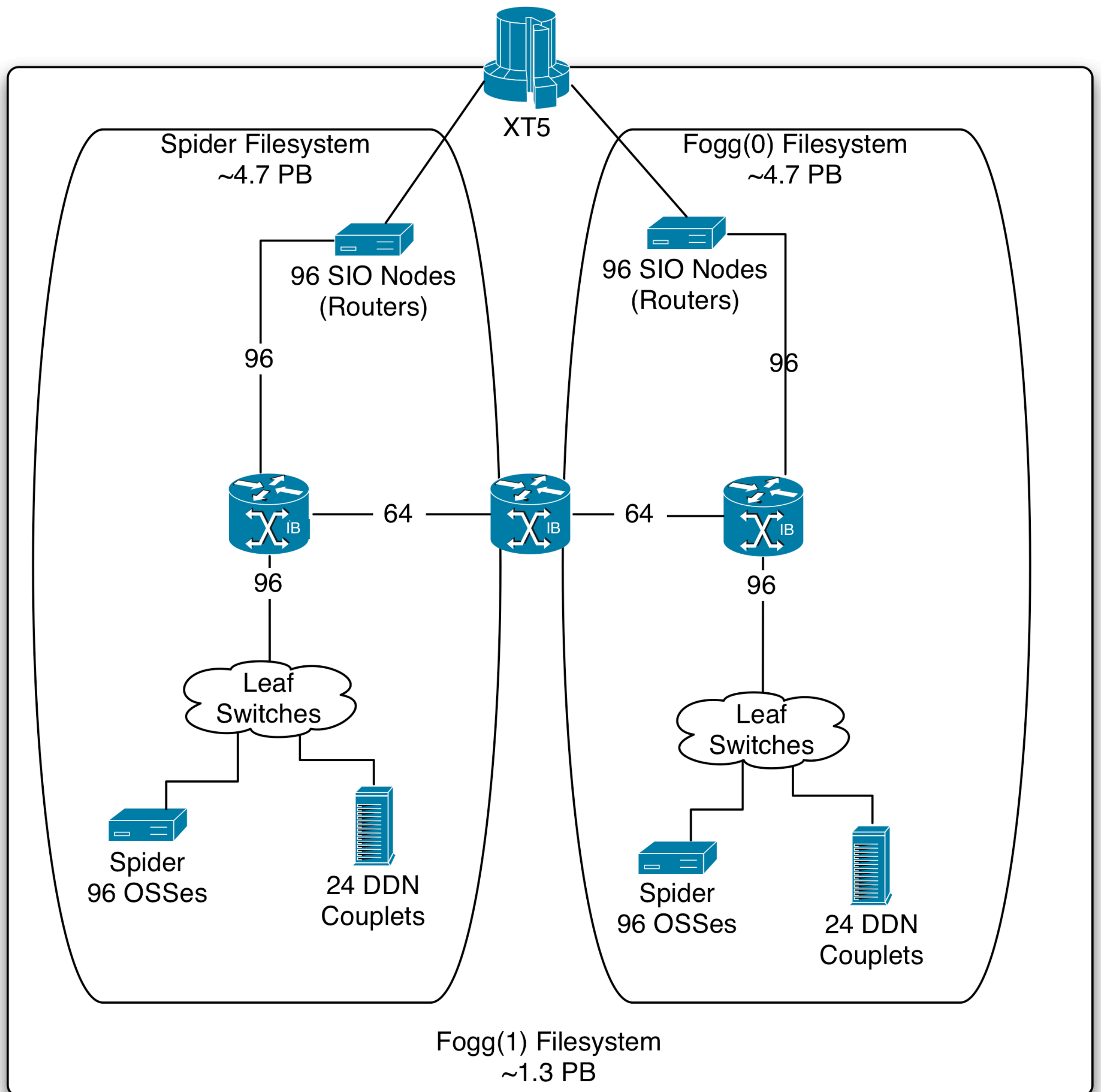

Jaguar (XT5)
SeaStar Torus
RTR
RTR
RTR
192 Routers
Jaguar (XT4)
SeaStar Torus
RTR
RTR
48 Routers
Smokey
Lens
HPSS
Archive
(10 PB)
ESnet, USN,
TeraGrid,
Internet2, NLR
10 – 40 Gbit/s
GridFTP
Servers
Lustre-WAN
Gateways
Scalable I/O Network (SION) - DDR InfiniBand – 889 GB/s
OSS
OSS
OSS
OSS
OSS
OSS
OSS
192 OSSs
1344 OSTs
10.7 PB 240 GB/s

Spider facts

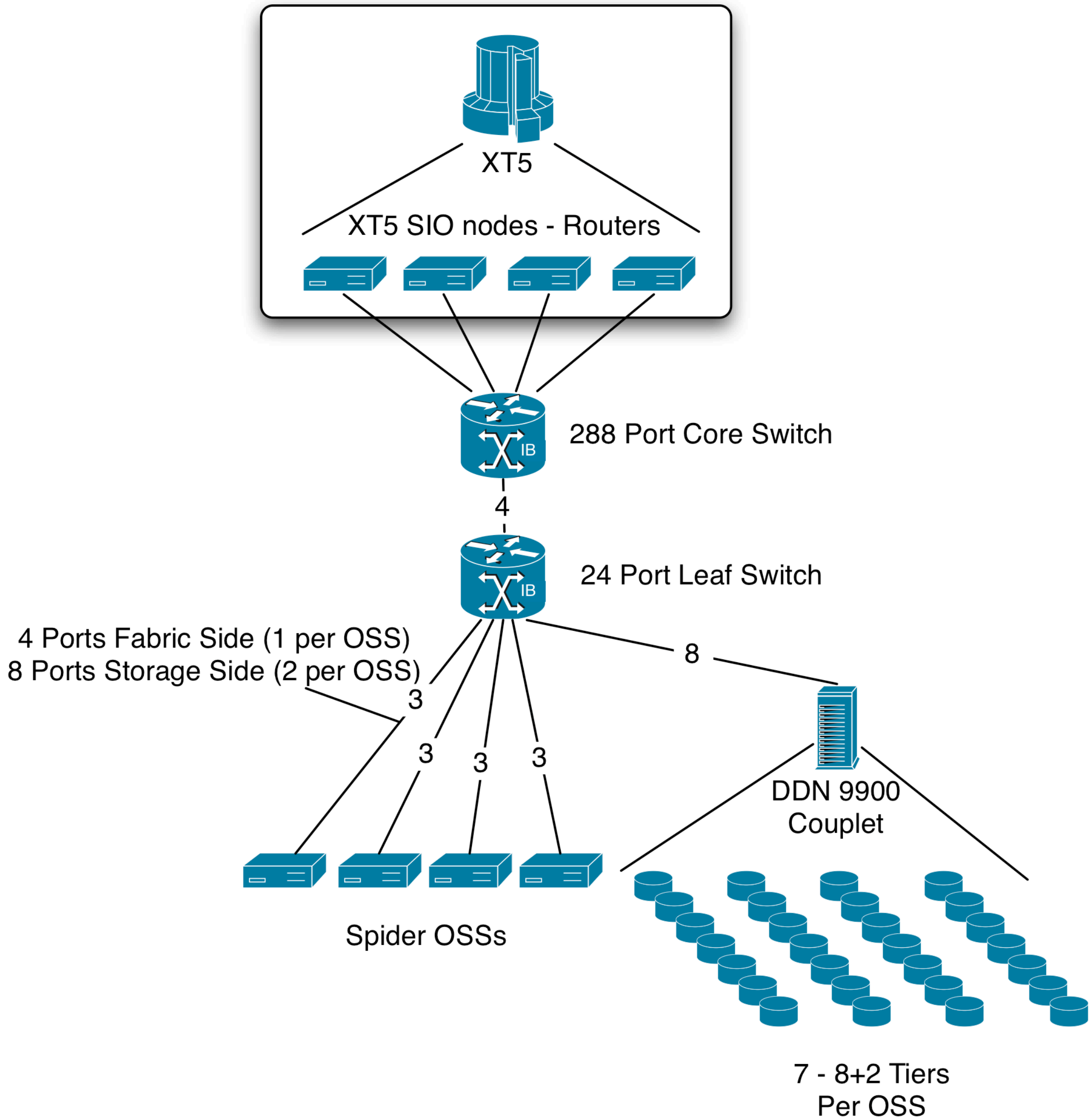
- 240 GB/s of Aggregate Bandwidth
- 48 DDN 9900 Couplets
- 13,440 1 TB SATA Drives
- Over 10 PB of RAID6 Capacity
- 192 Storage Servers
- Over 1000 InfiniBand Cables
- ~0.5 MW of Power
- ~20,000 LBs of Disks
- Fits in 32 Cabinets using 572 ft^2

OAK RIDGE National Laboratory

Spider Configuration



Spider Couplet View



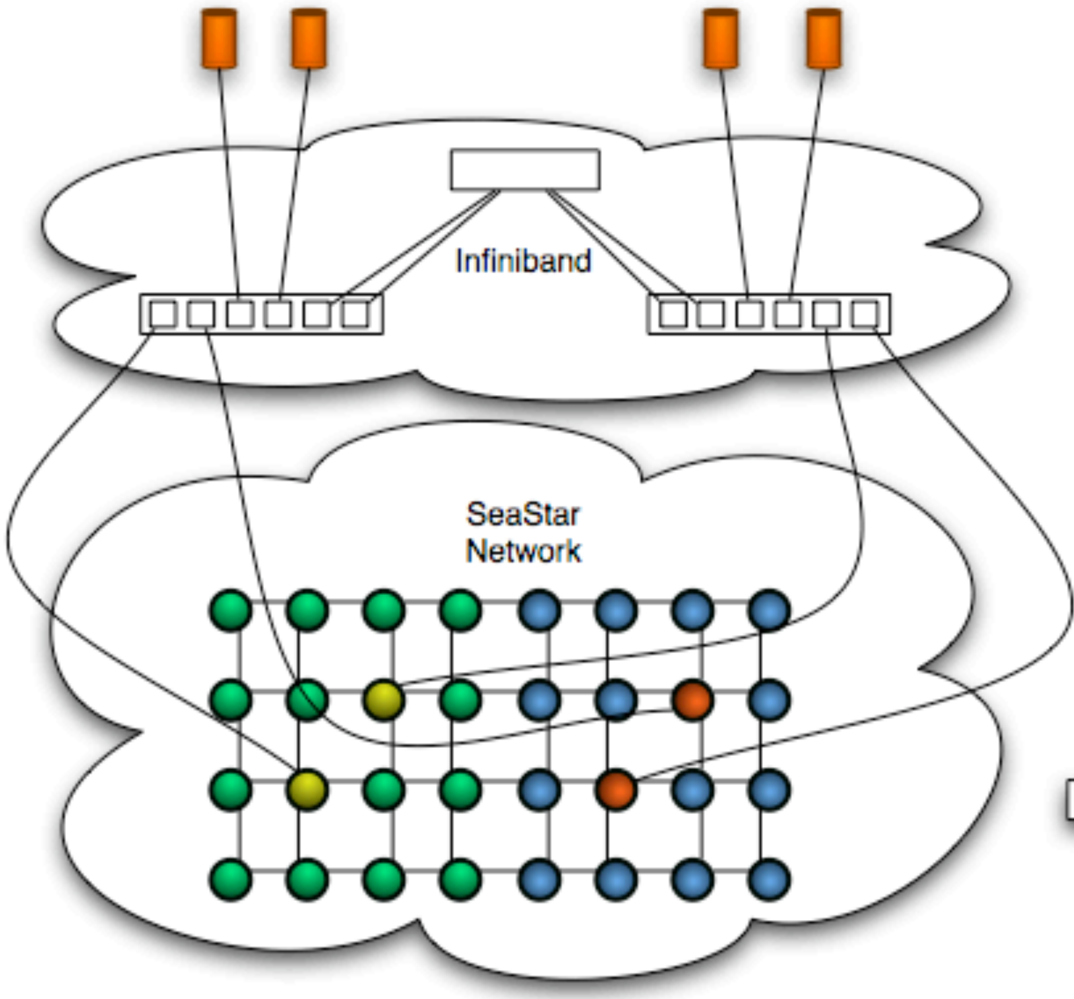
Lessons Learned: Network Congestion

- I/O infrastructure doesn't expose resource locality
  - There is currently no analog of nearest neighbor communication that will save us
- Multiple areas of congestion
  - Infiniband SAN
  - SeaStar Torus
  - LNET routing doesn't expose locality
    - May take a very long route unnecessarily
- Assumption of flat network space won't scale
  - Wrong assumption on even a single compute environment
  - Center wide file system will aggravate this
- Solution - Expose Locality
  - Lustre modifications allow fine grained routing capabilities

Design To Minimize Contention

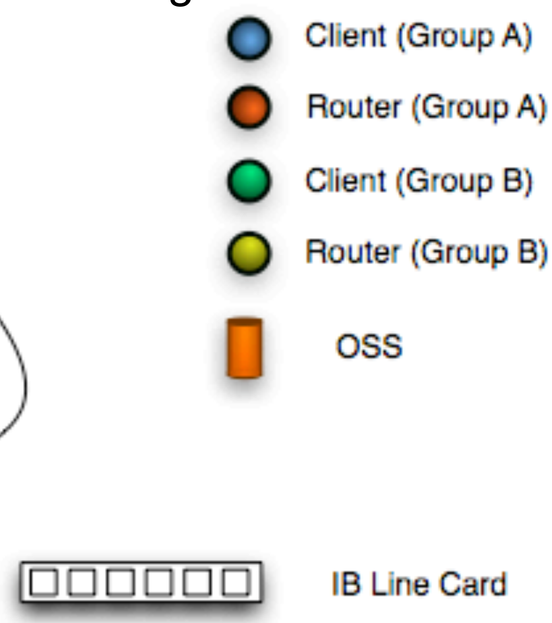
- Pair routers and object storage servers on the same line card (crossbar)
  - So long as routers only talk to OSSes on the same line card contention in the fat-tree is eliminated
  - Required small changes to Open SM
- Place routers strategically within the Torus
  - In some use cases routers (or groups of routers) can be thought of as a replicated resource
  - Assign clients to routers as to minimize contention
- Allocate objects to "nearest" OST
  - Requires changes to Lustre and/or I/O libraries

Intelligent LNET Routing



Clients prefer specific routers to these OSSes - minimizes IB congestion (same line card)

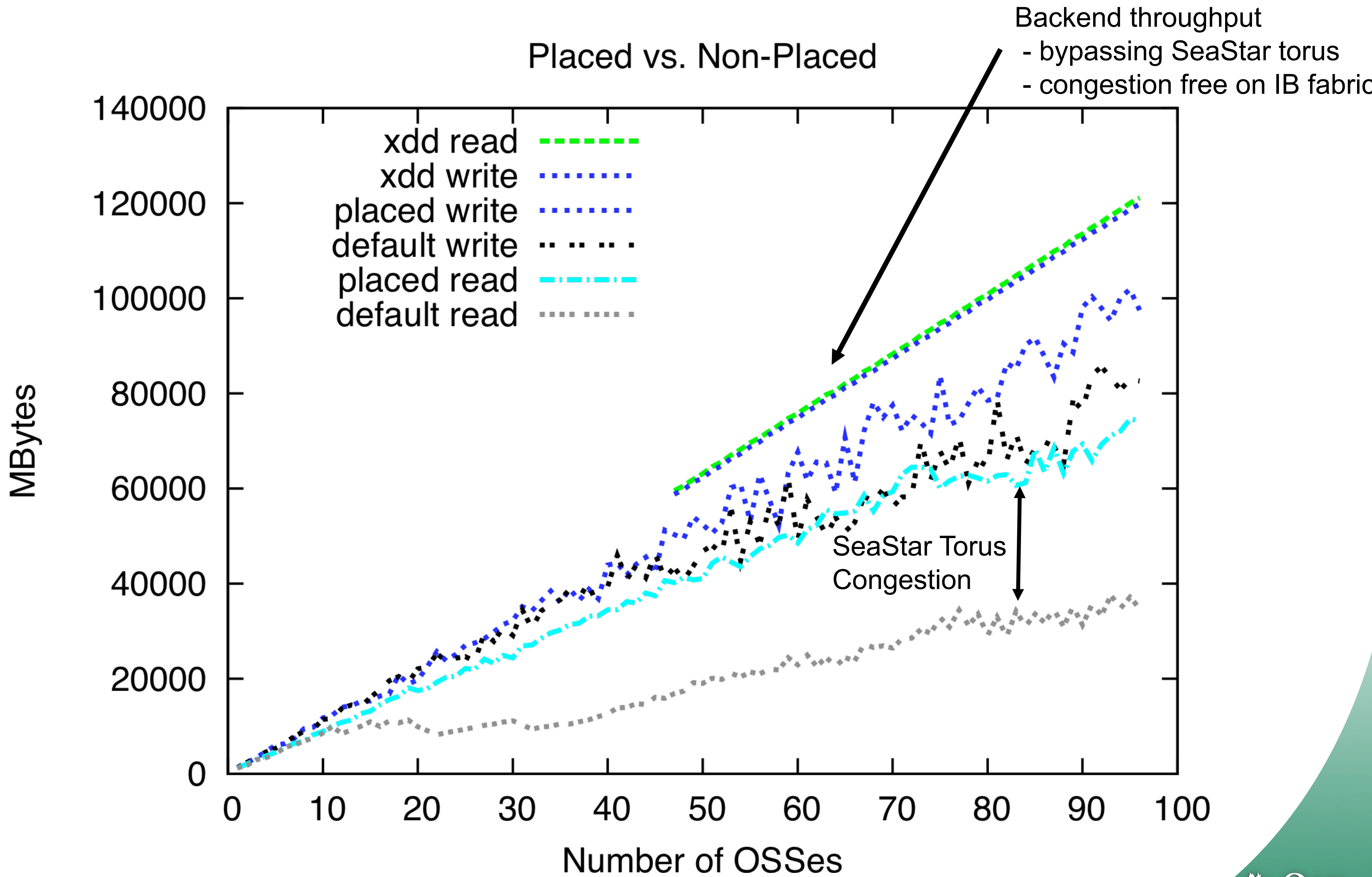
Assign clients to specific Router Groups - minimizes SeaStar Congestion



Performance Results

- Even in a direct attached configuration we have demonstrated the impact of network congestion on I/O performance
  - By strategically placing writers within the torus and pre-allocating file system objects we can substantially improve performance
  - Performance results obtained on Jaguar XT5 using ½ of the available backend storage

Performance Results (1/2 of Storage)



Lessons Learned: Journaling Overhead

- Even "sequential" writes can exhibit "random" I/O behavior due to journaling
- Special file (contiguous block space) reserved for journaling on ldiskfs
  - Located all together
  - Labeled as "journal device"
  - Towards the beginning on the physical disk layout
- After the file data portion is committed on disk
  - Journal meta data portion needs to committed as well
- Extra head seek needed for every journal transaction commit

Minimizing extra disk head seeks

- External journal on solid state devices
  - No disk seeks
  - Trade off between extra network transaction latency and disk seek latency
- Tested on a RamSan-400 device
  - 4 IB SDR 4x host ports
  - 7 external journal devices per host port
  - More than doubled the per DDN performance w.r.t. to internal journal devices on DDN devices
    - internal journal 1398.99
    - external journal on RAMSAN 3292.60
- Encountered some scalability problems per host port inherent to RamSan firmware
  - Reported to Texas Memory Systems Inc. and awaiting a resolution in next firmware release

Minimizing synchronous journal transaction commit penalty

- Two active transactions per ldiskfs (per OST)
  - One running and one closed
  - Running transaction can't be closed until closed transaction fully committed to disk
- Up to 8 RPCs (write ops) might be in flight per client
  - With synchronous journal committing
    - Some can be concurrently blocked until the closed transaction fully committed
  - Lower the client number, higher the possibility of lower utilization due to blocked RPCs
    - More writes are able to better utilize the pipeline

Minimizing synchronous journal transaction commit penalty

- To alleviate the problem
  - Reply to client when data portion of RPC is committed to disk
- Existing mechanism for client completion replies without waiting for data to be safe on disk
  - Only for meta data operations
  - Every RPC reply from a server has a special field in it that indicates “id last transaction on stable storage”
    - Client can keep track of completed, but not committed operations with this info
    - In case of server crash these operations could be resent (replayed) to the server once it is back up
- Extended the same concept for write I/O RPCs
- Implementation more than doubled the per DDN performance w.r.t. to internal journal devices on DDN devices
  - internal, sync journals 1398.99 MB/s
  - external, sync to RAMSAN 3292.60 MB/s
  - internal, async journals 4625.44 MB/s

Overcoming Journaling Overheads

- Identified two Lustre journaling bottlenecks
  - Extra head seek on magnetic disk
  - Blocked write I/O on synchronous journal commits
- Developed and implemented
  - A hardware solution based on solid state devices for extra head seek problem
  - A software solution based on asynchronous journal commits for the synchronous journal commits problem
- Both solutions more than doubled the performance
  - Async journal commits achieved better aggregate performance (with no additional hardware)

Lessons Learned: Disk subsystem overheads

- SATA IOP/s performance substantially degrades even "large block" random performance
  - Through detailed performance analysis we found that increasing I/O sizes from 1 MB to 4MB improved random I/O performance by a factor of 2.
  - Lustre level changes to increase RPC sizes from 1MB to 4MB are prototyped
  - Performance testing is underway, expect full results soon

Next steps

- Router attached testing using Jaguar XT5 underway
  - Over 18K Lustre clients
  - 96 OSSes
  - Over 100 GB/s of aggregate throughput
  - Transition to operations in early April
- Lustre WAN testing has been schedule
  - Two FTEs allocated to this task
  - Using Spider for this testing will allow us to explore issues of balance (1 GB/sec of client bandwidth vs. 100 GB/s of backend throughput)
- Lustre HSM development
  - ORNL has 3 FTEs contributing to HPSS who have begun investigating the Lustre HSM effort
  - Key to the success of our integrated backplane of services (automated migration/replication to HPSS)

Testbeds at ORNL

- Cray XT4 and XT5 single cabinet systems
  - DDN 9900 SATA
  - XBB2 SATA
  - RamSan-400
  - 5 Dell 1950 nodes (metadata + OSSes)
  - Allows testing of routed configuration and direct attached
- HPSS
  - 4 Movers, 1 Core server
  - DDN 9500

Testbeds at ORNL

- **WAN testbed**
  - OC192 Loop
    - 1400, 6600 and 8600 miles
  - 10 GigE and IB (Longbow) at the edge
  - Plan is to test using both Spider and our other testbed systems

A Few Storage System Trends

- Magnetic disks will be with us for some time (at least through 2015)
  - Disruptive technologies such as carbon nanotubes and phase change memory need significant research and investment
    - Difficult in the current economic environment
  - Rotational speeds are unlikely to improve dramatically (been at 15K for some time now)
  - Arial density becoming more of a challenge
  - Latency likely to remain nearly flat
- 2 ½ inch enterprise drives will dominate the market (aggregation at all levels will be required as drive counts continues to increase)
  - Examples currently exist: Seagate Savvio 10K.3

A Few Storage System Trends

- *Challenges for maintaining areal density trends
  - 1 TB per square inch is probably achievable via perpendicular grain layout, beyond this…
  - Superparamagnetic effect $K_u V \approx 60 k_b T$
  - Solution: store each bit as an exchange-coupled magnetic nanostructure (patterned magnetic media)
    - Requires new developments in Lithography
  - Ongoing research is promising, full scale manufacturing in 2012?

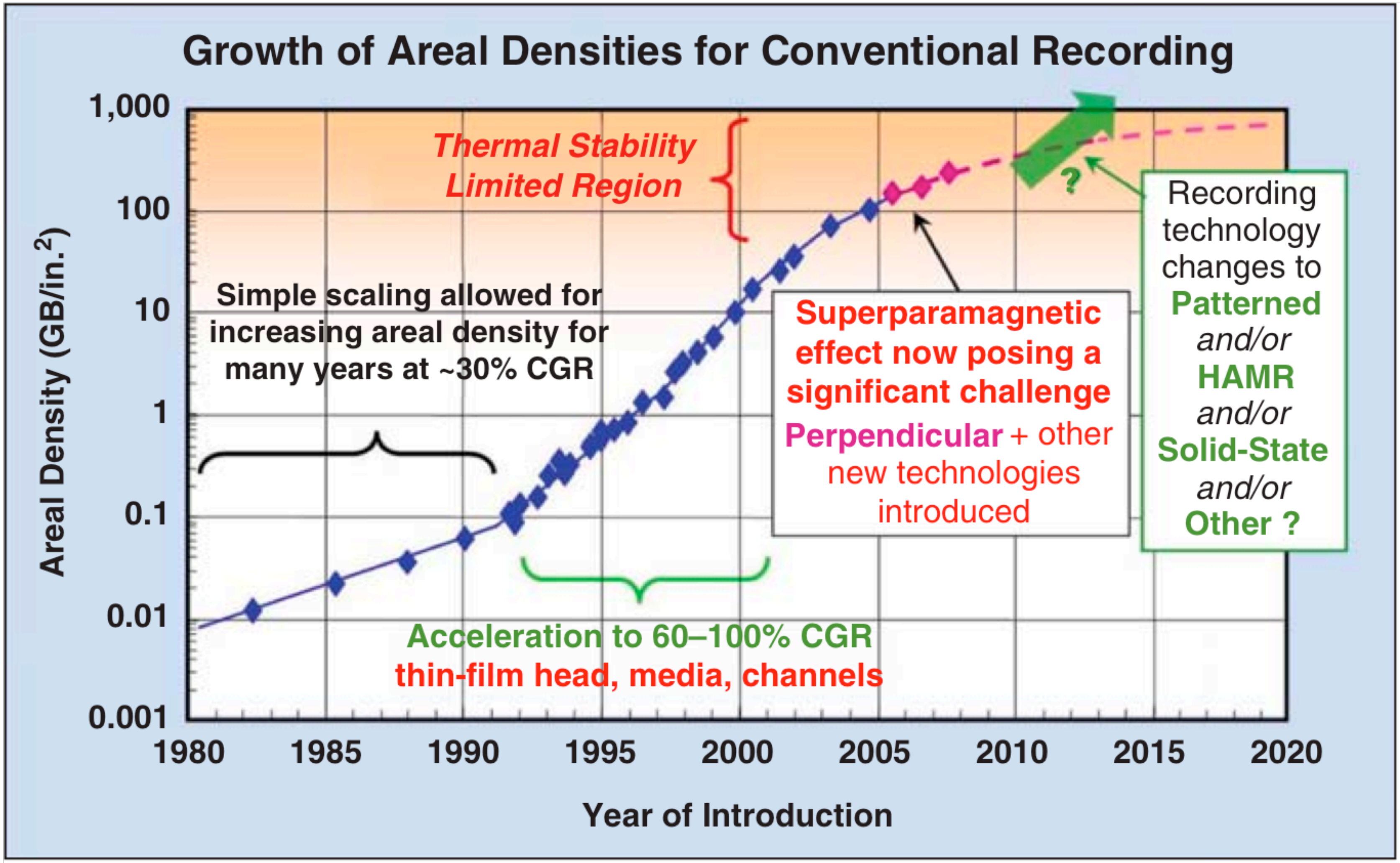
*MRS, September 2008: Nanostructured Materials in Information Storage

A Few Storage System Trends

- **Flash based devices will compete only at the high end**
  - **Ideal for replacing high IOP SAS drives**
  - **Cost likely to remain high relative to magnetic media**
  - ***Manufacturing techniques will improve density but charge retention will degrade at 8nm (or less) oxide thickness**
    - **Oxide film used to isolate a floating-gate**
    - **Will likely inhibit the same density trends seen in magnetic media**

*MRS, September 2008: Nanostructured Materials in Information Storage

Areal Density Trends



*MRS, September 2008: Nanostructured Materials in Information Storage

File system features to address storage trends

- Different storage systems for different I/O
  - File size
  - Access patterns
- SSDs for small files accessed often
- SAS based storage with cache mirroring for large "random" I/O
- SATA based storage for large "contiguous" I/O
- Log based storage targets for "write once" checkpoint data
- Offload object metadata – SSD for object description, magnetic media for data blocks
  - Implications for ZFS?

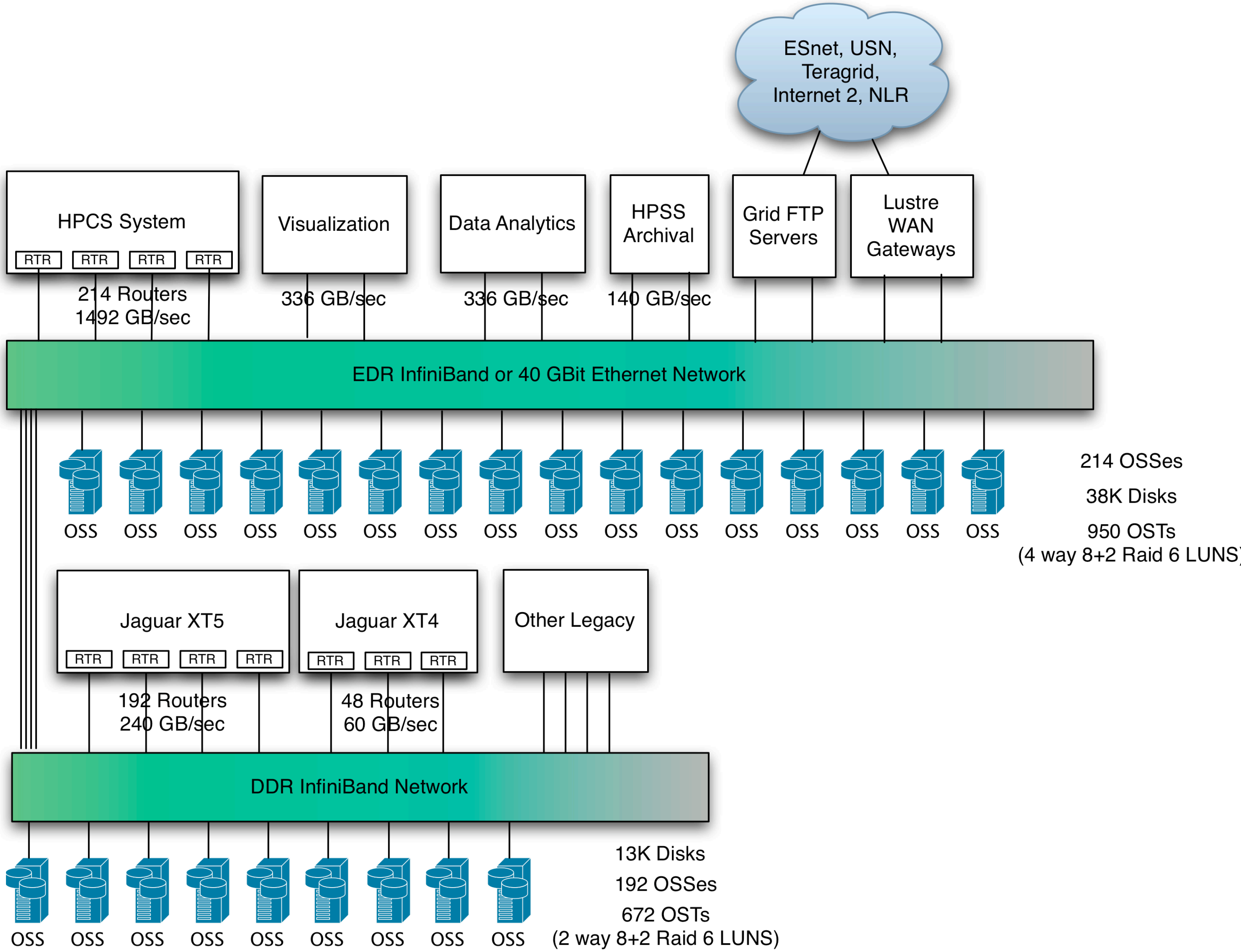
File system features to address storage trends

- Topology awareness
- Storage system pools
  - Automated migration policies
  - Much to learn from systems such as HPSS
- Ability to manage 100K+ drives
- Caching at multiple levels
  - Impacts recovery algorithms
- Alternatives to Posix interfaces
  - Expose global operations, I/O performance requirements and semantic requirements such as locking
  - Beyond MPI-I/O, a unified light weight I/O interface that is portable to multiple platforms and programming paradigms
    - MPI, Shmem, UPC, CAC, X10 and Fortress

2012 File System Projections

	Maintaining Current Balance (based on full system checkpoint in ~20 minutes)		Desired (based on full system checkpoint in 6 minutes)	
	Jaguar XT5	HPCS -2011	Jaguar XT5	HPCS -2011
Total Compute Node Memory (TB)	298	1,852	288	1,852
Total Disk Bandwidth (GB/s)	240	1,492	800	5,144
Per Disk Bandwidth (MB/sec)	25	50	25	50
Disk Capacity (TB)	1	8	1	8
Time to checkpoint 100% of Memory	1242	1242	360	360
Over Subscription of Disks (Raid 6)	1.25	1.25	1.25	1.25
Total # disks	12,288	38,184	40,960	131,698
Total Capacity (TB)	9,830	244,378	32,768	842,867
OSS Throughput (GB/sec)	1.25	7.00	1.25	8.00
OSS Nodes needed for bandwidth	192	214	640	644
OST disks per OSS for bandwidth	64	179	64	205
Total Clients	18,640	30,000	18,640	30,000
Clients per OSS	97	140	29	47

2012 Architecture



2012 file system requirements

- 1.5 TB/sec aggregate bandwidth
- 244 Petabytes of capacity (SATA - 8 TB)
  - 61 Petabytes of capacity (SAS – 2TB)
  - Final configuration may include pools of SATA, SAS and SSDs
- ~100K clients (from 2 major systems)
  - HPCS System
  - Jaguar
- ~200 OSTs per OSS
- ~400 clients per OSS

2012 file system requirements

- Full integration with HPSS
  - Replication, Migration, Disaster Recovery
  - Useful for large capacity project spaces
- OST Pools
  - Replication and Migration among pools
- Lustre WAN
  - Remote accessibility
- pNFS support
- QOS
  - Multiple platforms competing for bandwidth

2012 File System Requirements

- Improved data integrity
  - T10-DIF
  - ZFS (Dealing with licensing issues)
- Large LUN support
  - 256 TB
- Dramatically improved metadata performance
  - Improved single node SMP performance
  - Will clustered metadata arrive in time?
  - Ability to take advantage of SSD based MDTs

2012 File System Requirements

- Improved small block and random I/O performance
- Improved SMP performance for OSSes
  - Ability to support larger number of OSTs and clients per OSS
- Dramatically improved file system responsiveness
  - 30 seconds for “ls -l” ?
  - Performance will certainly degrade as we continue adding additional computational resources to Spider

Good overlap with HPCS I/O Scenarios

- 1. Single stream with large data blocks operating in half duplex mode
- 2. Single stream with large data blocks operating in full duplex mode
- 3. Multiple streams with large data blocks operating in full duplex mode
- 4. Extreme file creation rates
- 5. Checkpoint/restart with large I/O requests
- 6. Checkpoint/restart with small I/O requests
- 7. Checkpoint/restart large file count per directory - large I/Os
- 8. Checkpoint/restart large file count per directory - small I/Os
- 9. Walking through directory trees
- 10. Parallel walking through directory trees
- 11. Random stat() system call to files in the file system – one (1) process
- 12. Random stat() system call to files in the file system - multiple processes