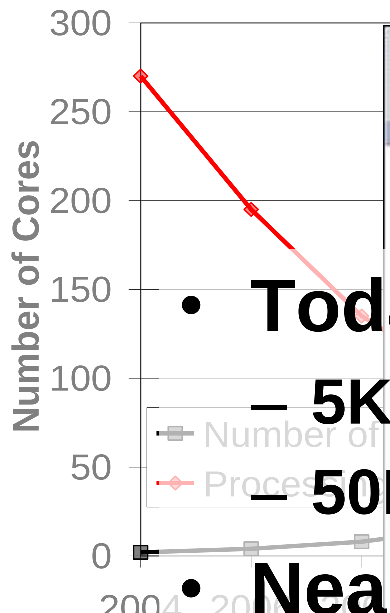
Rethinking Storage Systems for Exascale Computing

Ioan Raicu
Berkin Ozisikyilmaz, Chen Jin, Arefin Huq, Alok Choudhary
Center for Ultra-scale Computing and Information Security
Department of Electrical Engineering & Computer Science
Northwestern University

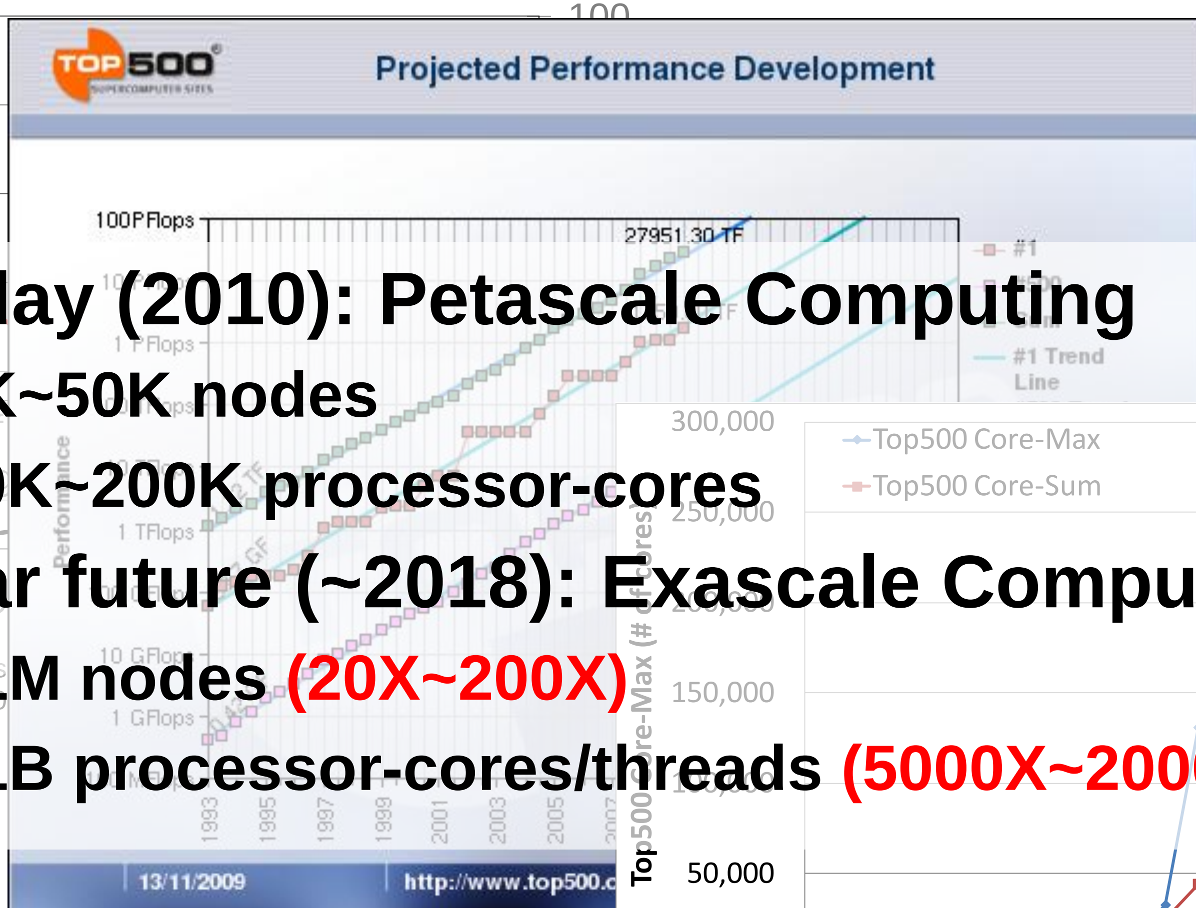
HPDC 2010 - Wild and Crazy Ideas Session
June 25th, 2010

Projected Growth Trends

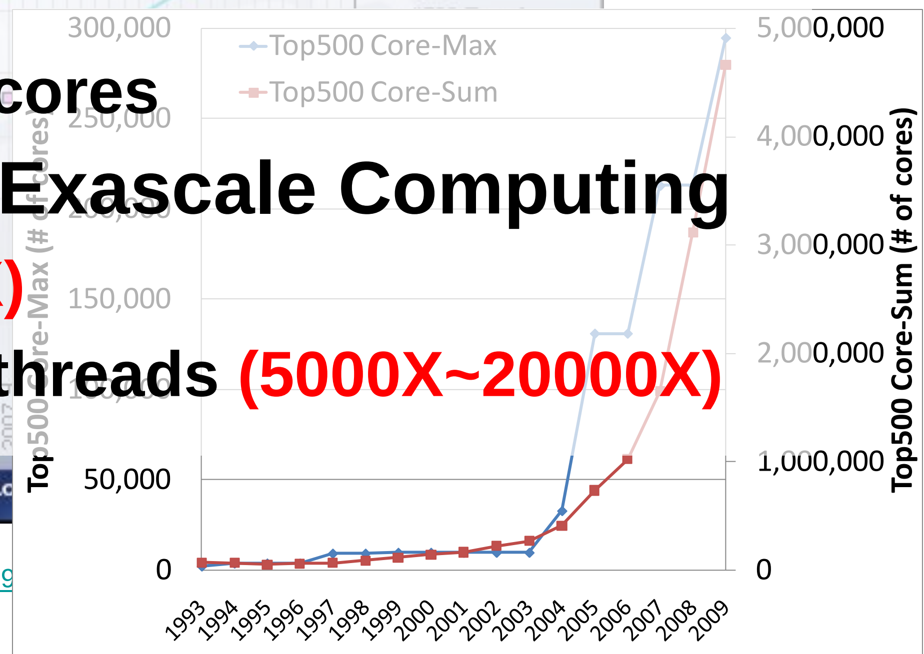
- **Today (2010): Petascale Computing**
  - **5K~50K nodes**
  - **50K~200K processor-cores**
- **Near future (~2018): Exascale Computing**
  - **~1M nodes (20X~200X)**
  - **~1B processor-cores/threads (5000X~20000X)**



Pat Helland, Microsoft, The Ir... Objects, November 9th, 20...



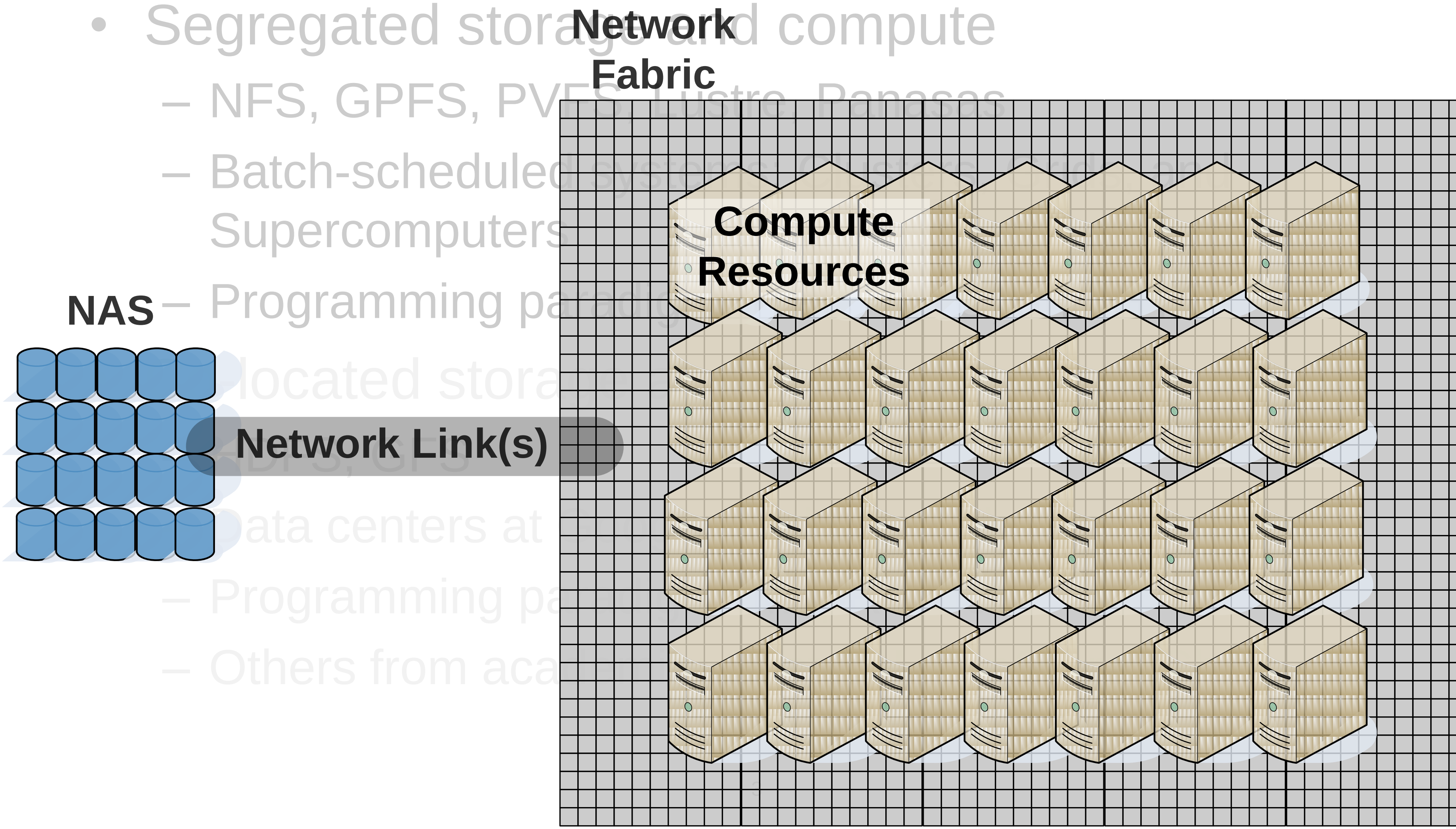
Top500 Projected Development, http://www.top500.org/lists/2009



http://www.top500.org/

State-of-the-Art Storage Systems in HEC Parallel File Systems

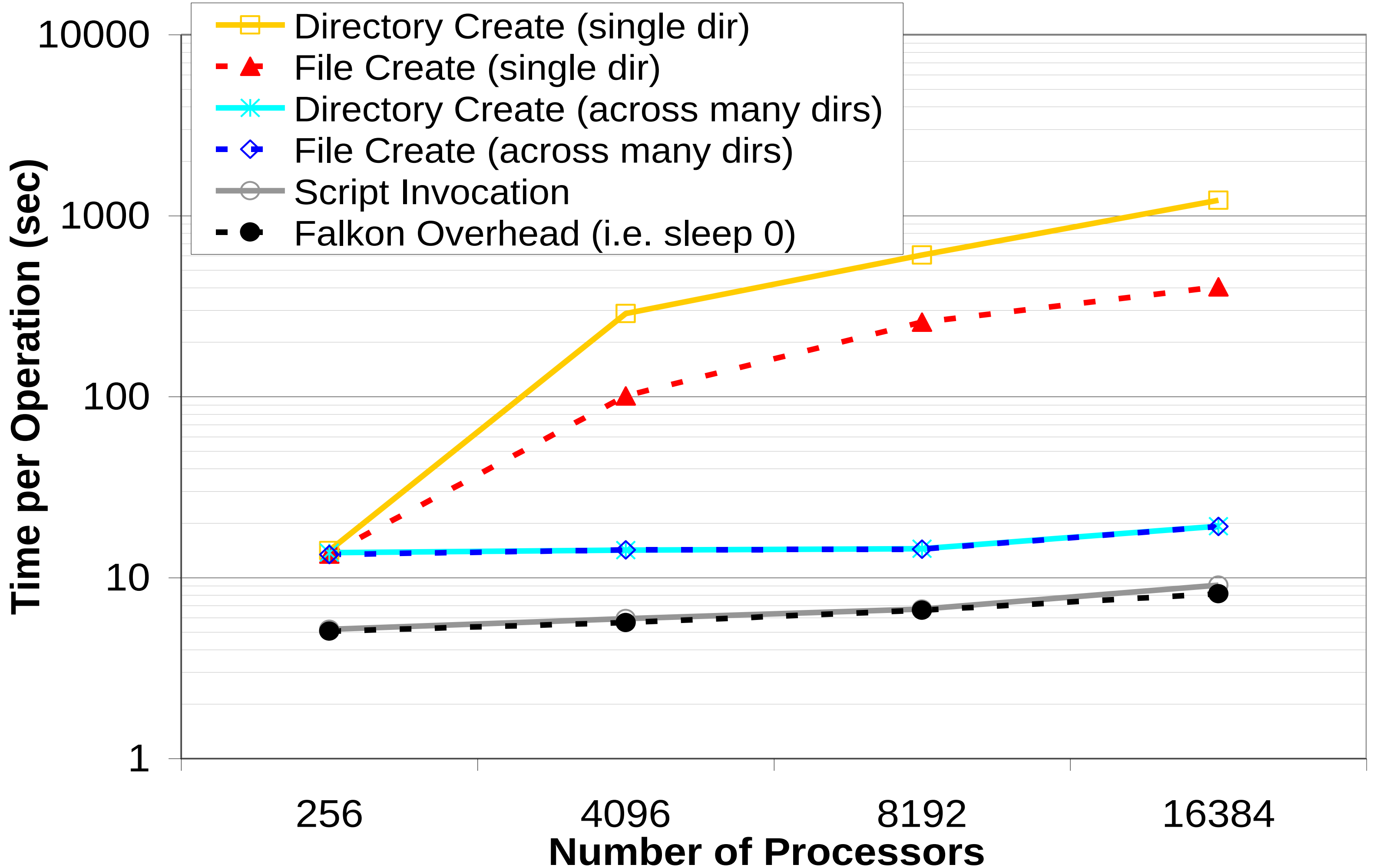
- Segregated stora... d compute
  - NFS, GPFS, PV... Lustre, Panasas
  - Batch-scheduled ... Supercomputers
  - Programming pa...
- ...located stora...
  - HDFS, GFS
- ...ata centers at ...
  - Programming pa...
  - Others from aca...



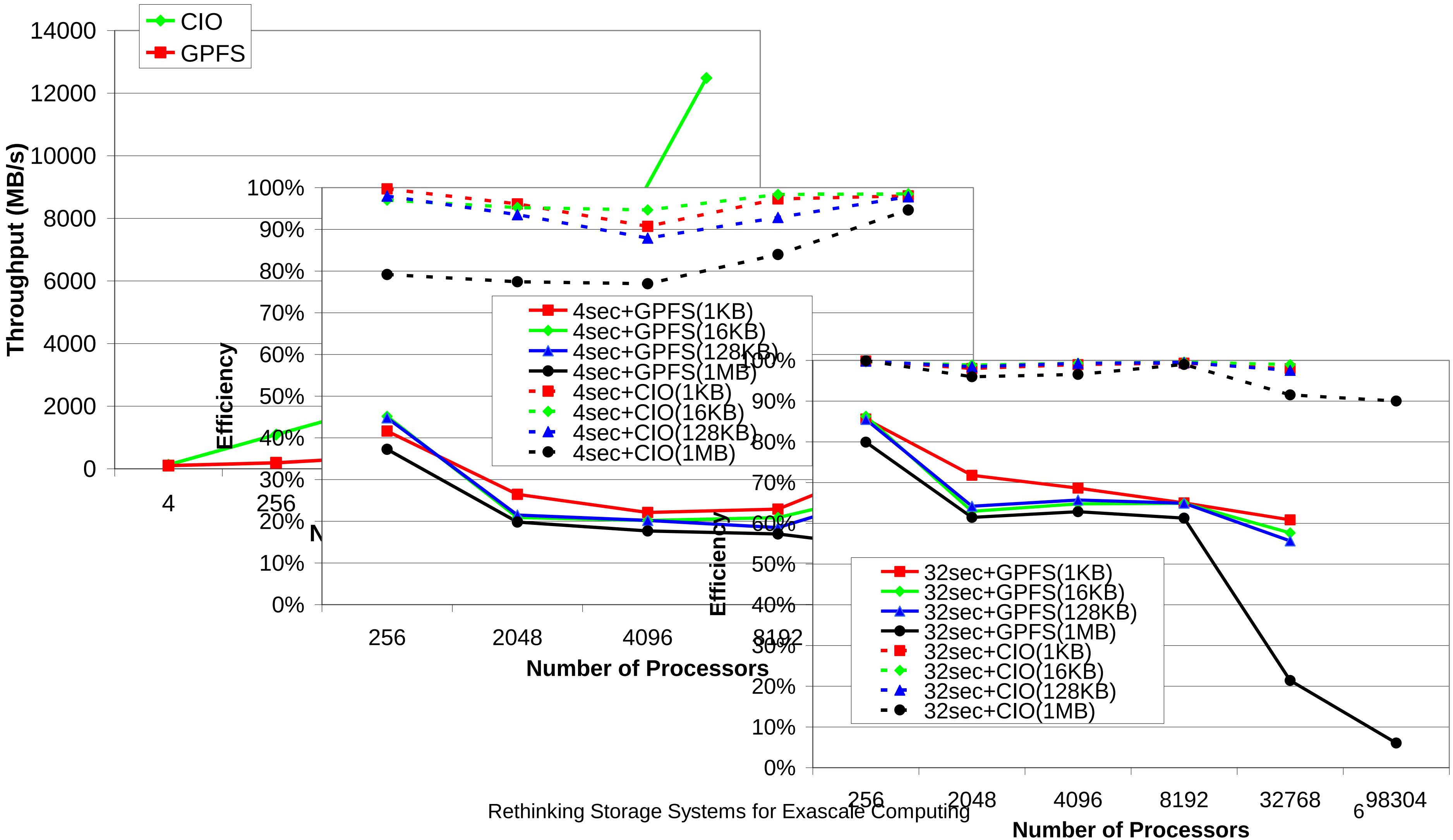
What are the key challenges?

- MTTF is likely to decrease with system size
- Support for data intensive applications/operations
  - Fueled by more complex questions, larger datasets, and the many-core computing era
  - ***HPC***: OS booting, application loading, check-pointing
  - ***HTC***: Inter-process communication
  - ***MTC***: Metadata intensive workloads, inter-process communication

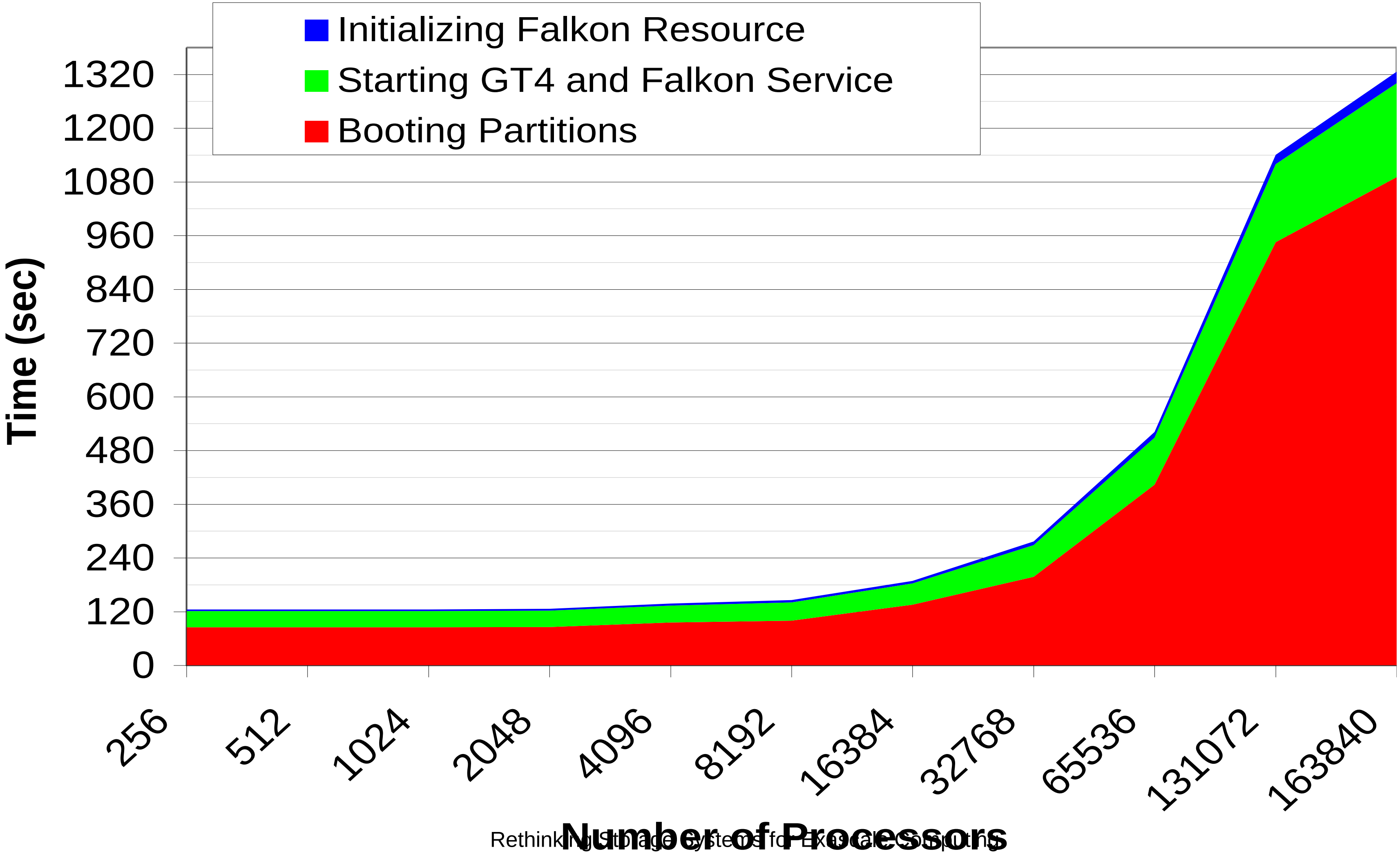
Some Challenges
Meta-data Operations on GPFS



Some Challenges Reading/Writing on GPFS



Some Challenges
Booting a IBM BlueGene/P



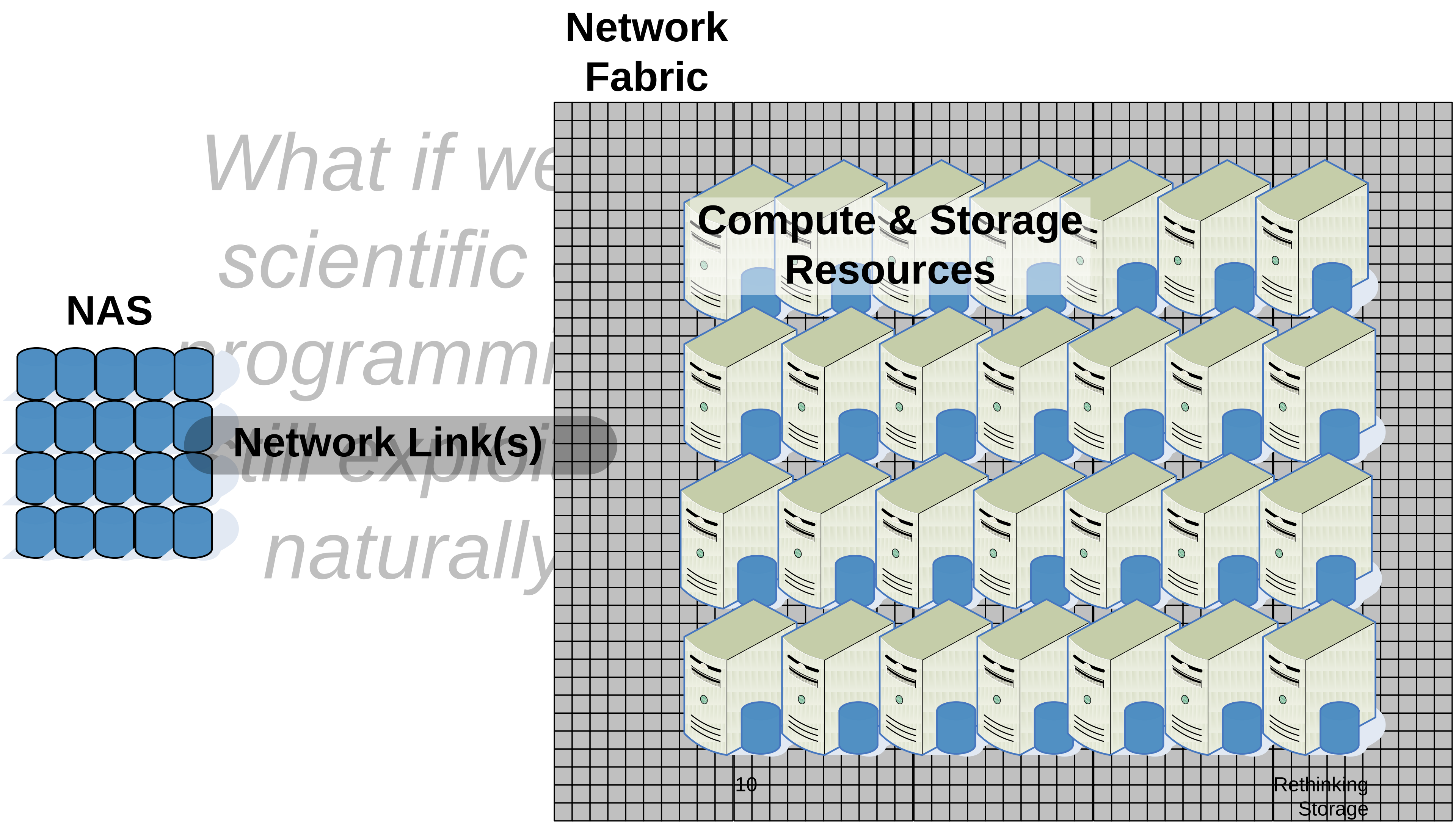
Exascale Supercomputing Architecture

- Compute
  - ~1M nodes
  - ~1K threads/cores per node
- Networking
  - N-dimensional torus
  - Meshes
- Storage
  - SANs with spinning disks will replace today's tape
  - SANs with SSDs might exist, replacing today's spinning disk SANs
  - SSDs will exist at every node

Proposed Work Directions

- ***Decentralization is critical***
  - Computational resource management (e.g. LRMs)
  - Storage systems (e.g. parallel file systems)
- ***Data locality must be maximized, while preserving I/O interfaces***
  - POSIX I/O on shared/parallel file systems ignore locality
  - Data-aware scheduling coupled with distributed file systems that expose locality is the key to scalability over the next decade

Proposed Storage System Architecture



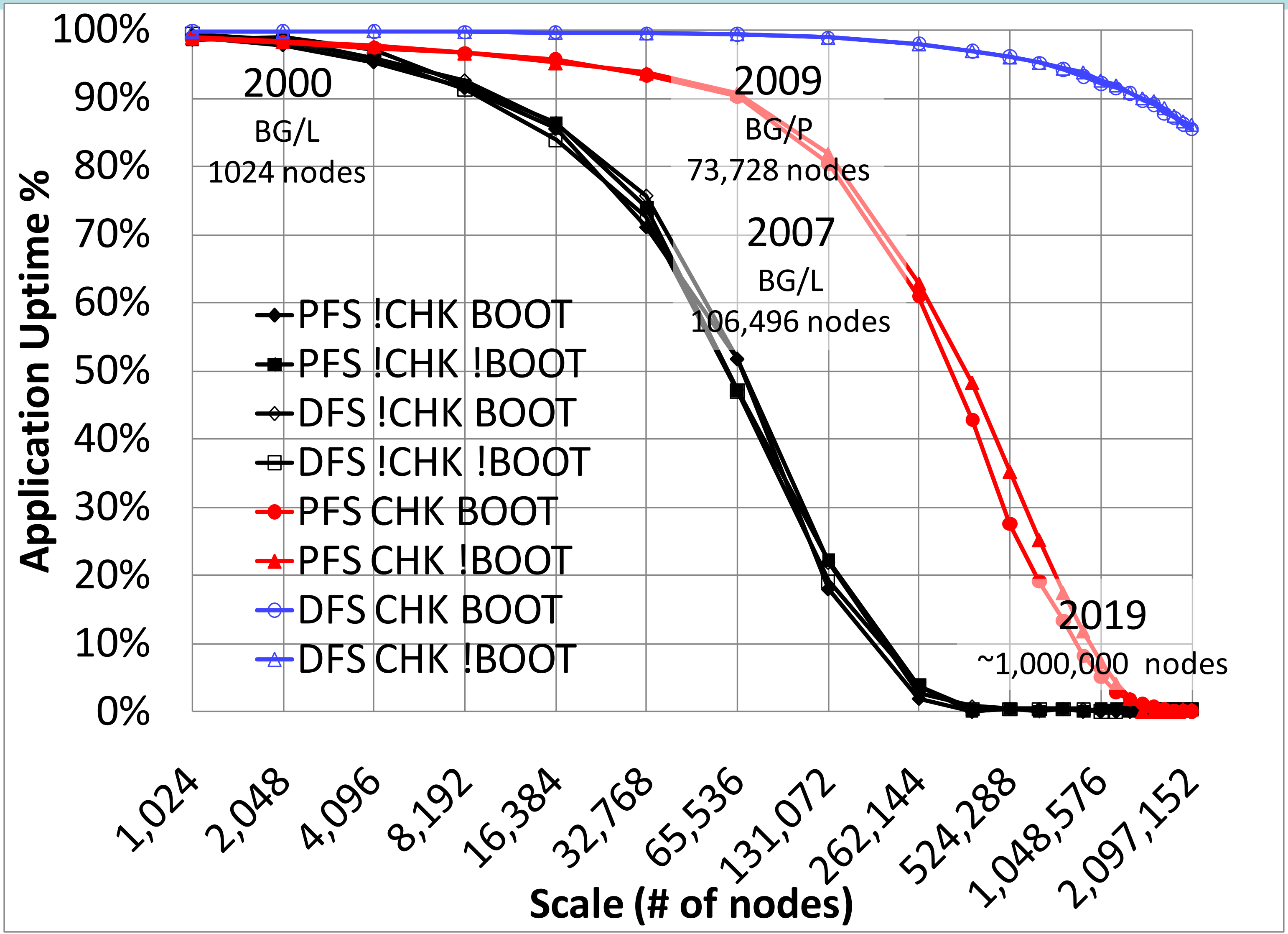
Proposed Work (cont)

- ***Building on my own research (e.g. data-diffusion), parallel file systems (PVFS), and distributed file systems (e.g. GFS)***
- Build a distributed file system for HEC
  - It should complement parallel file systems, not replace them
- Critical issues:
  - Must mimic parallel file systems interfaces and features in order to get wide adoption
  - Must handle some workloads currently run on parallel file systems significantly better

Proposed Work (cont)

- Access Interfaces and Semantics
  - POSIX-like compliance for generality (e.g. via FUSE)
  - Relaxed semantics to increase scalability
    - Eventual consistency on data modifications
    - Write-once read-many data access patterns
- Distributed metadata management
  - Employ structured distributed hash tables like data-structures
  - Must have O(1) put/get costs
  - Can leverage network-aware topology overlays
- Distribute data across many nodes
  - Must maintain and expose data locality in access patterns

Exascale Computing is Feasible!



More Information

- More information:
  - http://www.eecs.northwestern.edu/~iraicu/
  - iraicu@eecs.northwestern.edu
- Funding:
  - **NSF**: Computing Research Innovation Fellow Program