

# **Making a Case for Distributed File Systems at Exascales**

**Ioan Raicu**

Computer Science Department, Illinois Institute of Technology  
Math and Computer Science Division, Argonne National Laboratory

Indiana University  
April 27<sup>th</sup>, 2011

# Who am I?

- **Current position:**
  - Assistant Professor at Illinois Institute of Technology (CS)
  - Guest Research Faculty, Argonne National Laboratory (MCS)
- **Education:** PhD, University of Chicago, March 2009
- **Funding/Awards:**
  - NASA GSRP, 2006 – 2009 (\$84K)
  - NSF/CRA CIFellows, 2009 – 2010 (\$140K)
  - NSF CAREER, 2011 – 2015 (\$450K)
- **Over 70+ Collaborators:**
  - Ian Foster (UC/ANL), Rick Stevens (UC/ANL), Rob Ross (ANL), Marc Snir (UIUC), Arthur Barney Maccabe (ORNL), Alex Szalay (JHU), Pete Beckman (ANL), Kamil Iskra (ANL), Mike Wilde (UC/ANL), Douglas Thain (ND), Yong Zhao (UEST), Matei Ripeanu (UBC), Alok Choudhary (NU), Tevfik Kosar (SUNY), Yogesh Simhan (USC), Ewa Deelman (USC), and many more...
- **More info:** <http://www.cs.iit.edu/~iraicu/index.html>

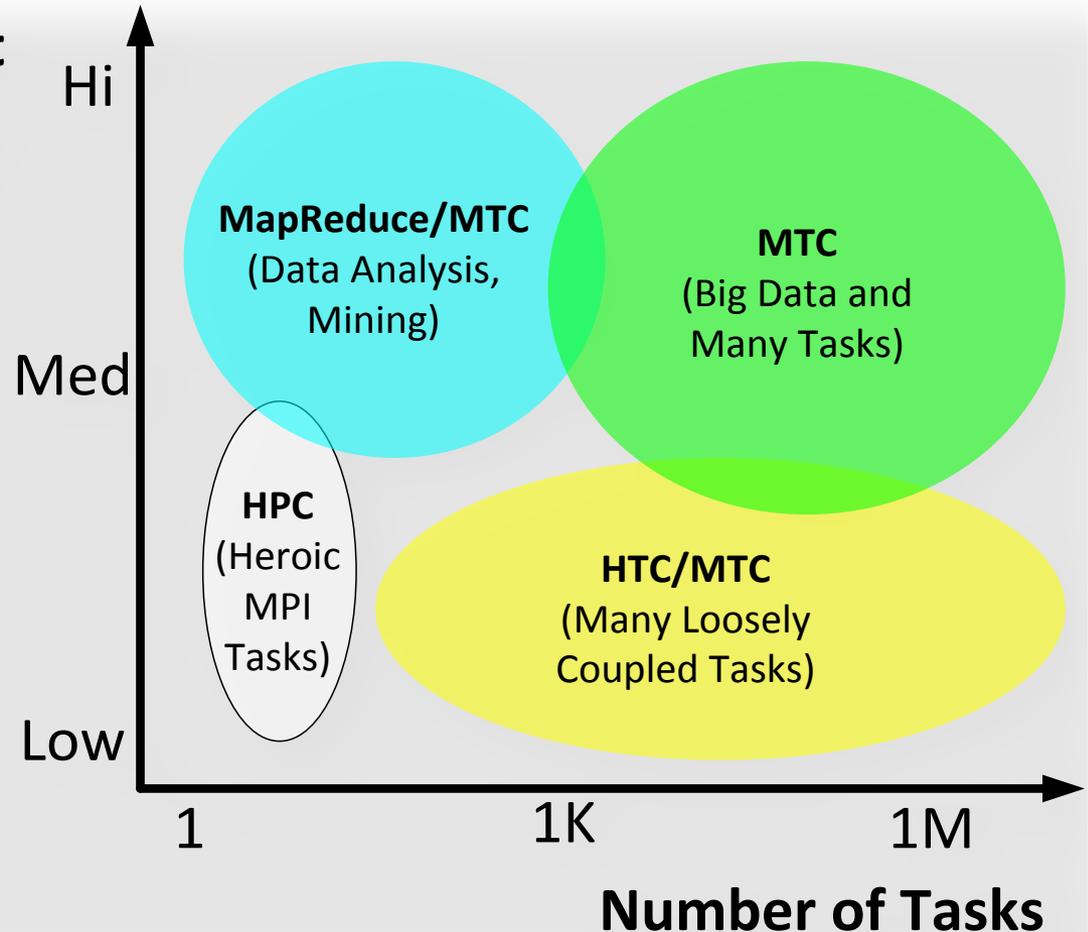


# Best Known For

- **MTC: Many-Task Computing**

- Bridge the gap between HPC and HTC
- Applied in clusters, grids, and supercomputers
- Loosely coupled apps with HPC orientations
- Many activities coupled by file system ops
- Many resources over short time periods

Input  
Data  
Size



# Best Known For

- **Falkon**
  - **Fast and Lightweight Task Execution Framework**

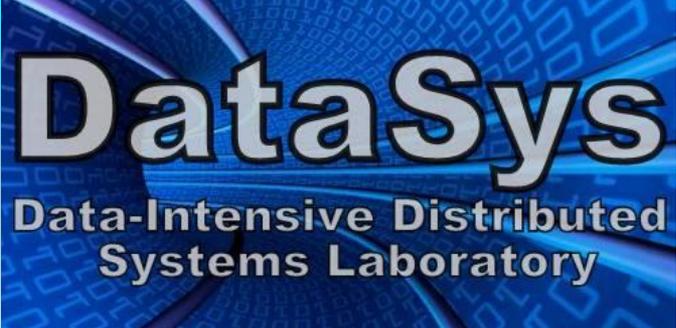
- <http://dev.globus.org/ubator/Falkon>

- **Swift**
  - **Parallel Programming System**

- <http://www.ci.uchicago.edu/swift/index.php>

| Field              | Description                                                                                              | Characteristics                                                                               | Status         |
|--------------------|----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|----------------|
| Astronomy          | Creation of montages from many digital images                                                            | Many 1-core tasks, much communication, complex dependencies                                   | Experimental   |
| Astronomy          | Stacking of cutouts from digital sky surveys                                                             | Many 1-core tasks, much communication                                                         | Experimental   |
| Biochemistry*      | Analysis of mass-spectrometer data for post-translational protein modifications                          | 10,000-100 million jobs for proteomic searches using custom serial codes                      | In development |
| Biochemistry*      | Protein structure prediction using iterative fixing algorithm; exploring other biomolecular interactions | Hundreds to thousands of 1- to 1,000-core simulations and data analysis                       | Operational    |
| Biochemistry*      | Identification of drug targets via computational docking/screening                                       | Up to 1 million 1-core docking operations                                                     | Operational    |
| Bioinformatics*    | Metagenome modeling                                                                                      | Thousands of 1-core integer programming problems                                              | In development |
| Business economics | Mining of large text corpora to study media bias                                                         | Analysis and comparison of over 70 million text files of news articles                        | In development |
| Climate science    | Ensemble climate model runs and analysis of output data                                                  | Tens to hundreds of 100- to 1,000-core simulations                                            | Experimental   |
| Economics*         | Generation of response surfaces for various economic models                                              | 1,000 to 1 million 1-core runs (10,000 typical), then data analysis                           | Operational    |
| Neuroscience*      | Analysis of functional MRI datasets                                                                      | Comparison of images; connectivity analysis with structural equation modeling, 100,000+ tasks | Operational    |
| Radiology          | Training of computer-aided diagnosis algorithms                                                          | Comparison of images; many tasks, much communication                                          | In development |
| Radiology          | Image processing and brain mapping for neuro-surgical planning research                                  | Execution of MPI application in parallel                                                      | In development |

Note: Asterisks indicate applications being run on Argonne National Laboratory's Blue Gene/P (Intrepid) and/or the TeraGrid Sun Constellation at the University of Texas at Austin (Ranger).



# DataSys: Data-Intensive Distributed Systems Laboratory

- **Research Focus**

- Emphasize designing, implementing, and evaluating systems, protocols, and middleware with the goal of supporting **data-intensive applications on extreme scale distributed systems**, from many-core systems, clusters, grids, clouds, and supercomputers

- **People**

- Dr. Ioan Raicu (Director)
- Tonglin Li (PhD Student)
- Xi Duan (MS Student)
- Raman Verma (Research Staff)
- 3 PhD and 1 UG students joining over the next several months

- **More information**

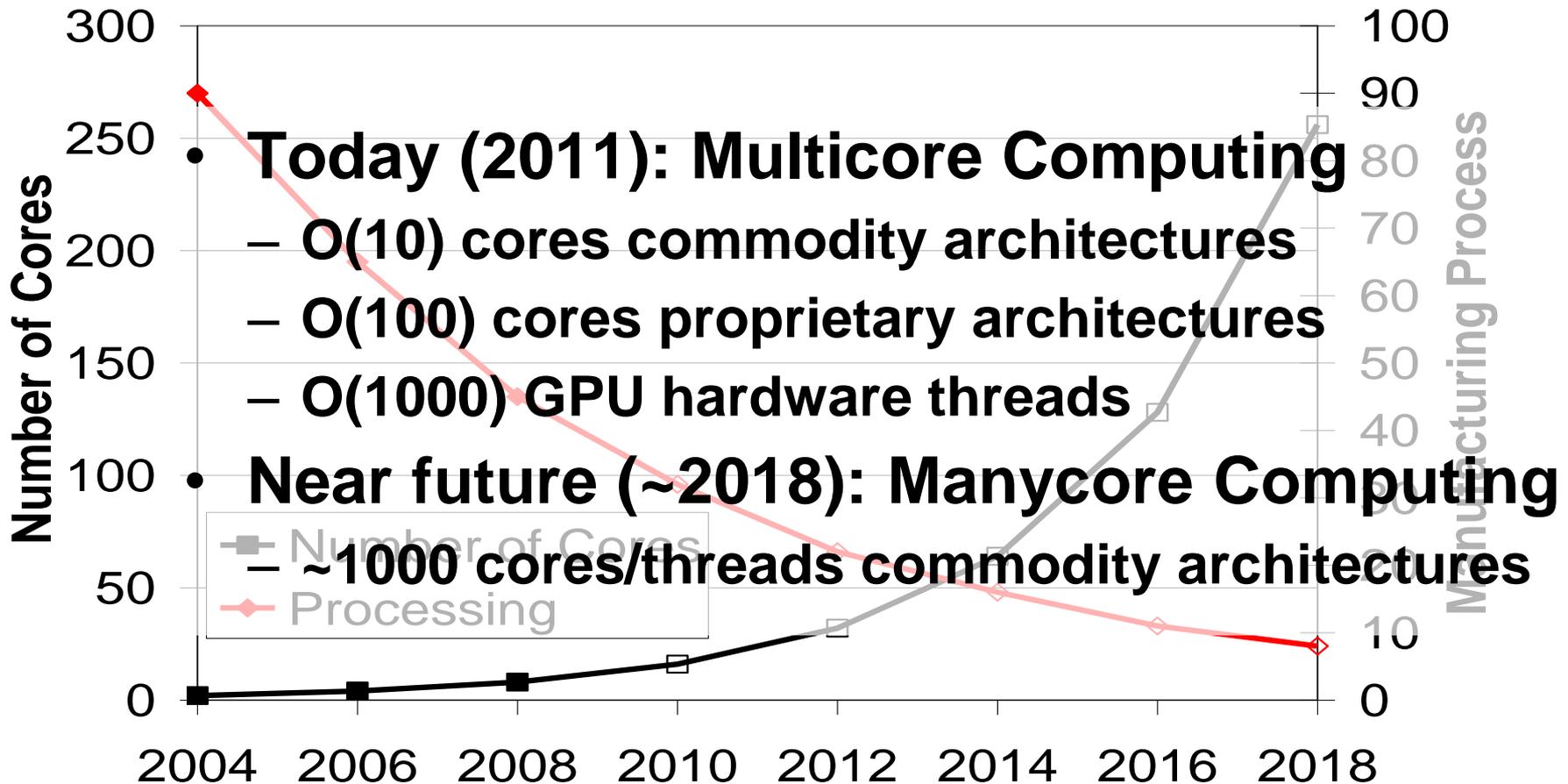
- <http://datasys.cs.iit.edu/>



# Overview

- **This talk covers material from my NSF CAREER award:**
  - Ioan Raicu, Arthur Barney Maccabe, Marc Snir, Rob Ross, Mike Wilde, Kamil Iskra, Jacob Furst, Mary Cummane. “Avoiding Achilles’ Heel in Exascale Computing with Distributed File Systems”, NSF OCI CAREER Award #1054974
- **And from a recent invited position paper (to appear):**
  - Ioan Raicu, Pete Beckman, Ian Foster. “[Making a Case for Distributed File Systems at Exascale](#)”, ACM Workshop on Large-scale System and Application Performance (LSAP), 2011

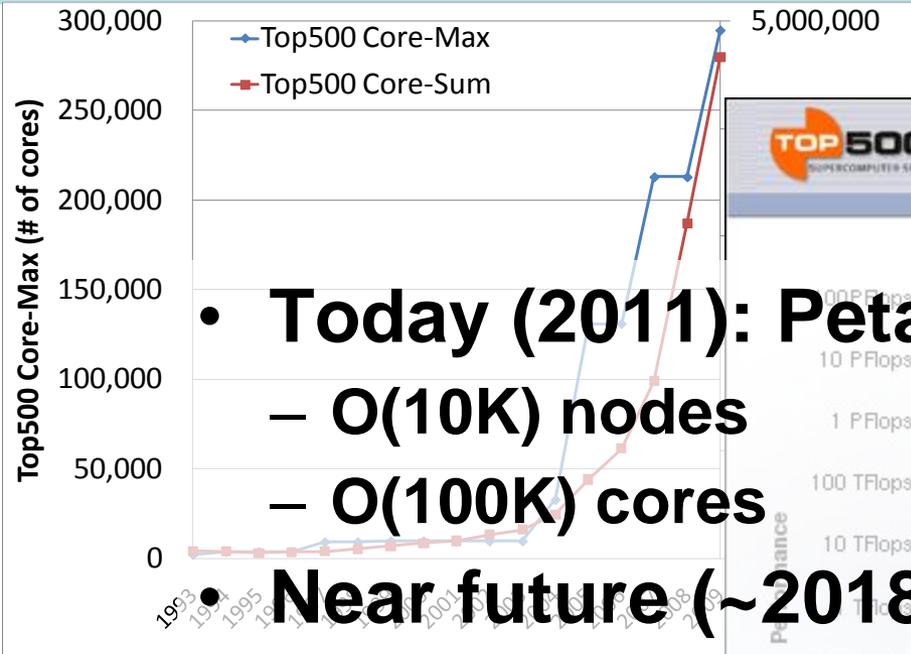
# Manycore Computing



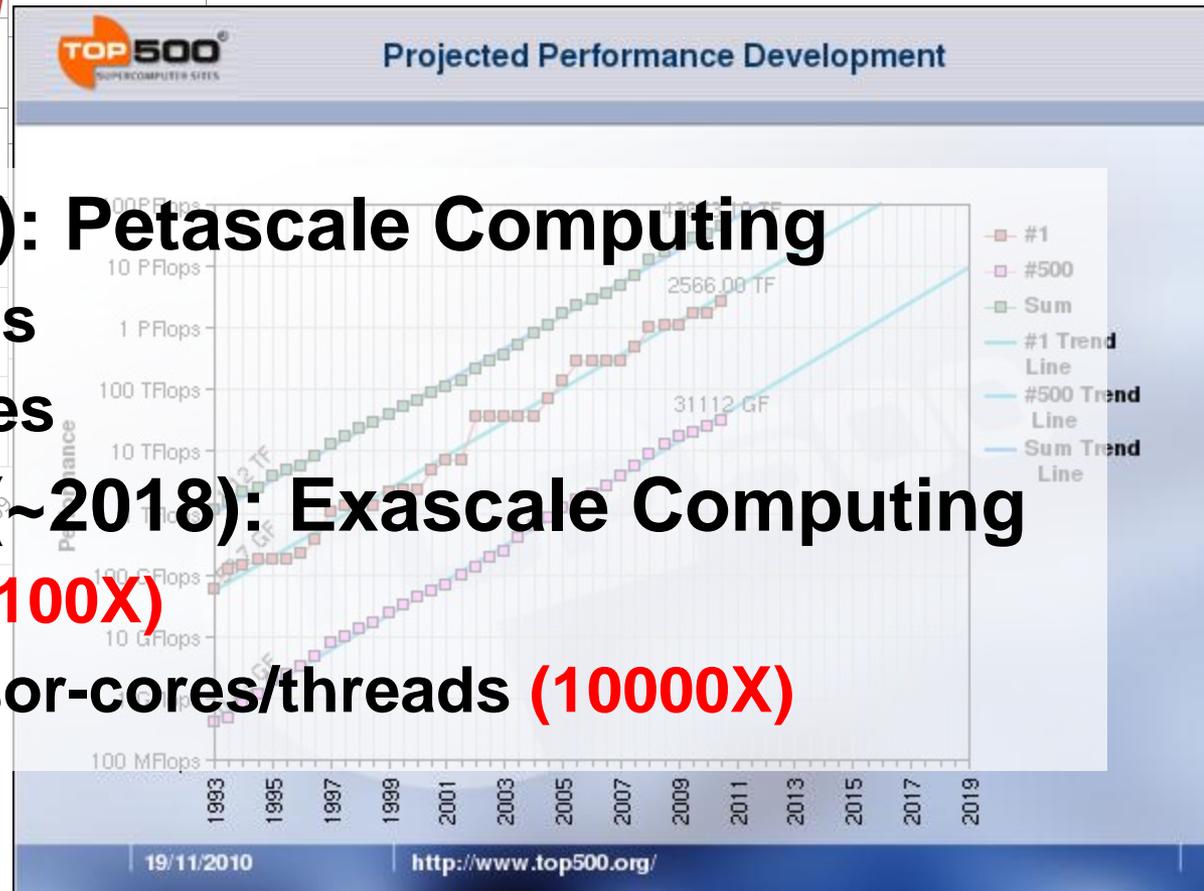
Pat Helland, Microsoft, The Irresistible Forces Meet the Movable

Objects, November 9<sup>th</sup>, 2007  
 Making a Case for Distributed File Systems at Exascales

# Exascale Computing



- **Today (2011): Petascale Computing**
  - O(10K) nodes
  - O(100K) cores
- **Near future (~2018): Exascale Computing**
  - ~1M nodes (100X)
  - ~1B processor-cores/threads (10000X)



Top500 Projected Development,

[http://www.top500.org/lists/2010/11/performance\\_development](http://www.top500.org/lists/2010/11/performance_development)

# Cloud Computing

- Relatively new paradigm... 3~4 years old
- Amazon in 2009
  - 40K servers split over 6 zones
    - 320K-cores, 320K disks
    - \$100M costs + \$12M/year in energy costs
    - Revenues about \$250M/year
    - [http://www.siliconvalleywatcher.com/mt/archives/2009/10/measuring\\_amaz.php](http://www.siliconvalleywatcher.com/mt/archives/2009/10/measuring_amaz.php)
- Amazon in 2018
  - Will likely look similar to exascale computing
    - 100K~1M nodes, ~1B-cores, ~1M disks
    - \$100M~\$200M costs + \$10M~\$20M/year in energy
    - Revenues 100X~1000X of what they are today

# Common Challenges

- Power efficiency
  - Will limit the number of cores on a chip (Manycore)
  - Will limit the number of nodes in cluster (Exascale and Cloud)
  - Will dictate a significant part of the cost of ownership
- Programming models/languages
  - Automatic parallelization
  - Threads, MPI, workflow systems, etc
  - Functional, imperative
  - Languages vs. Middleware

# Common Challenges

- Bottlenecks in scarce resources
  - Storage (Exascale and Clouds)
  - Memory (Manycore)
- Reliability
  - How to keep systems operational in face of failures
  - Checkpointing (Exascale)
  - Node-level replication enabled by virtualization (Exascale and Clouds)
  - Hardware redundancy and hardware error correction (Manycore)

# State-of-the-Art Storage Systems in HEC

## Parallel File Systems

- Segregated storage and compute

- NFS, GPFS, PVFS, Lustre, Panasas

- Batch-scheduled Supercomputers

- Programming pa

- Colocated storage

- Network Link(s)

- Data centers at

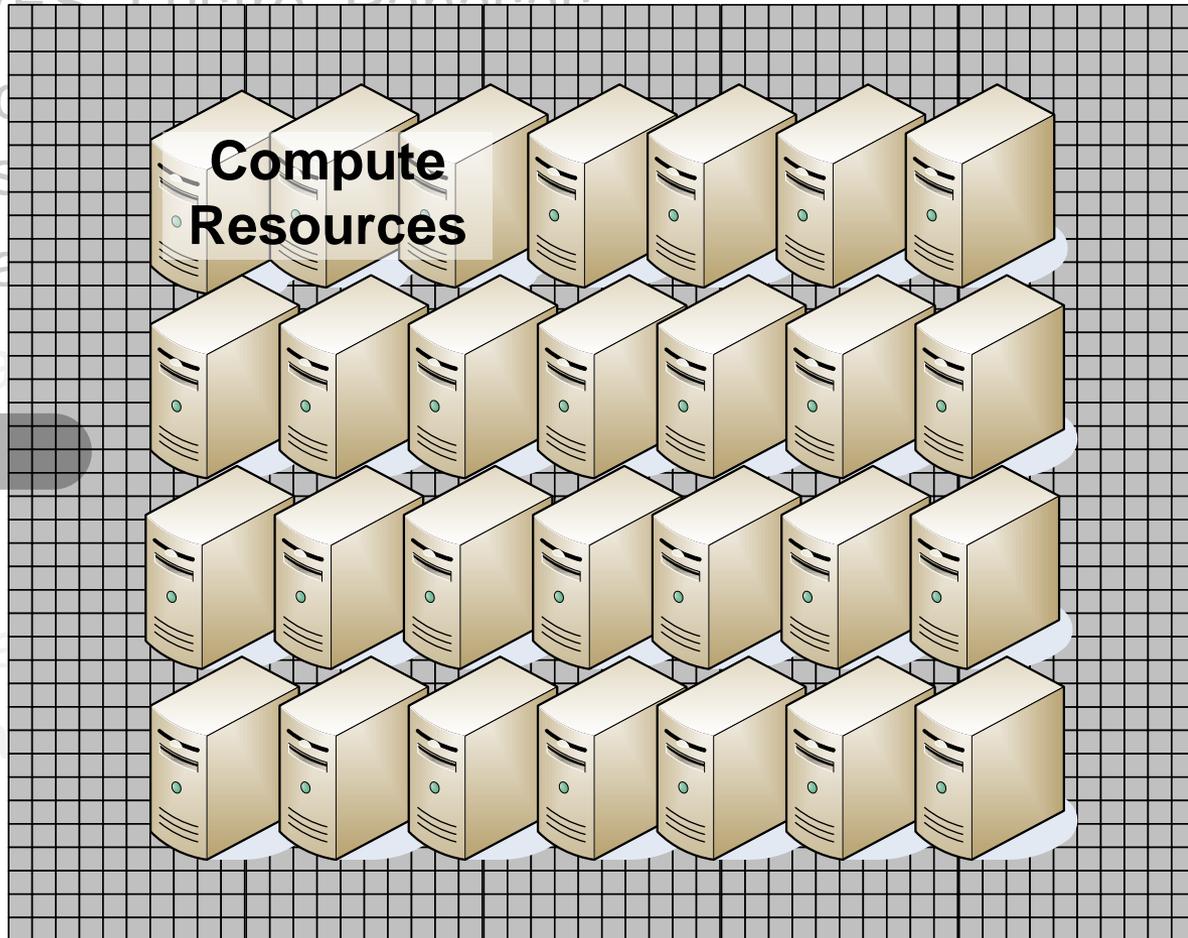
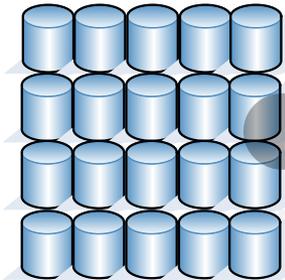
- Programming pa

- Others from aca

**Network Fabric**

**Compute Resources**

**NAS**

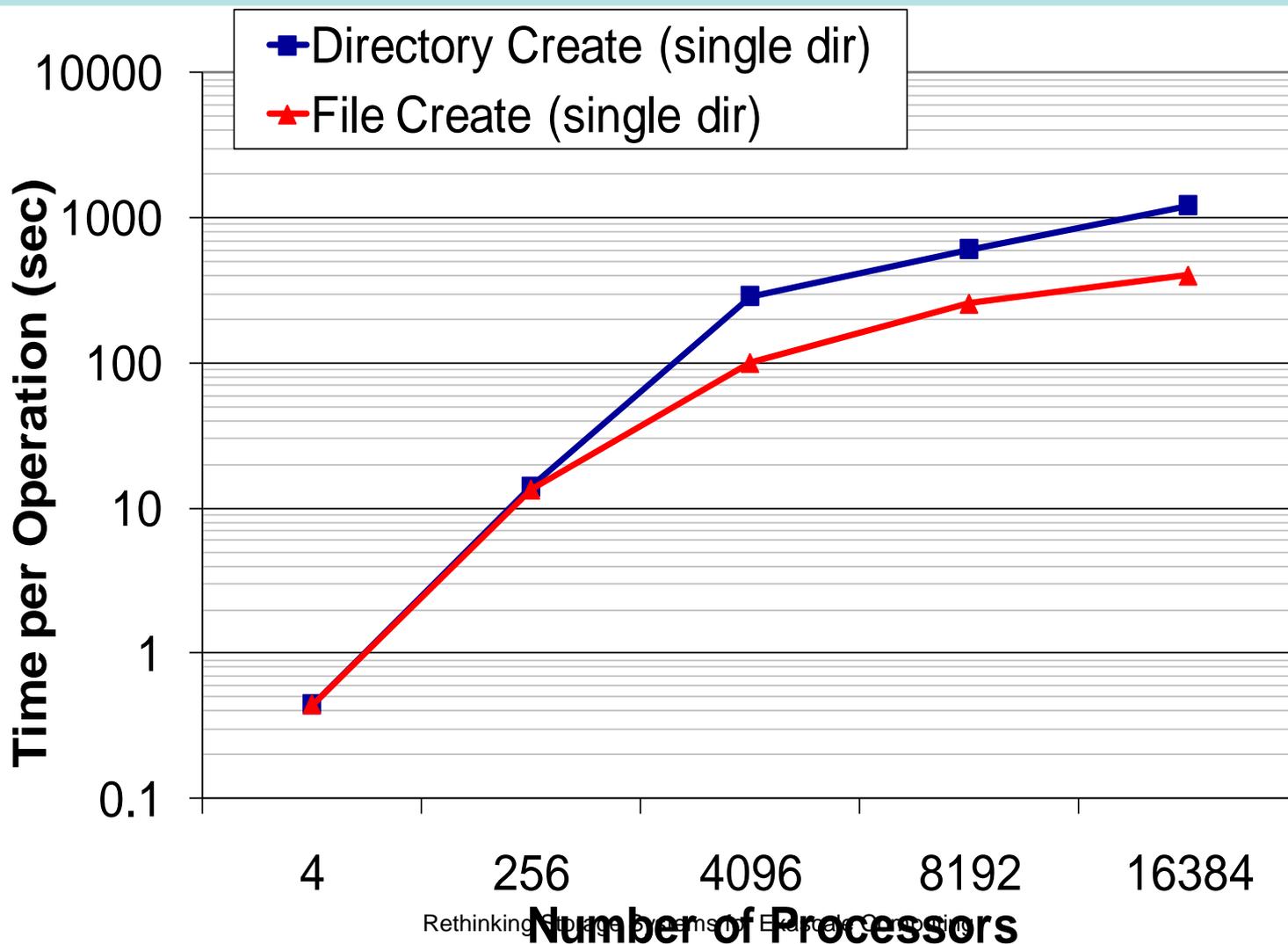


# What are the key challenges?

- 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

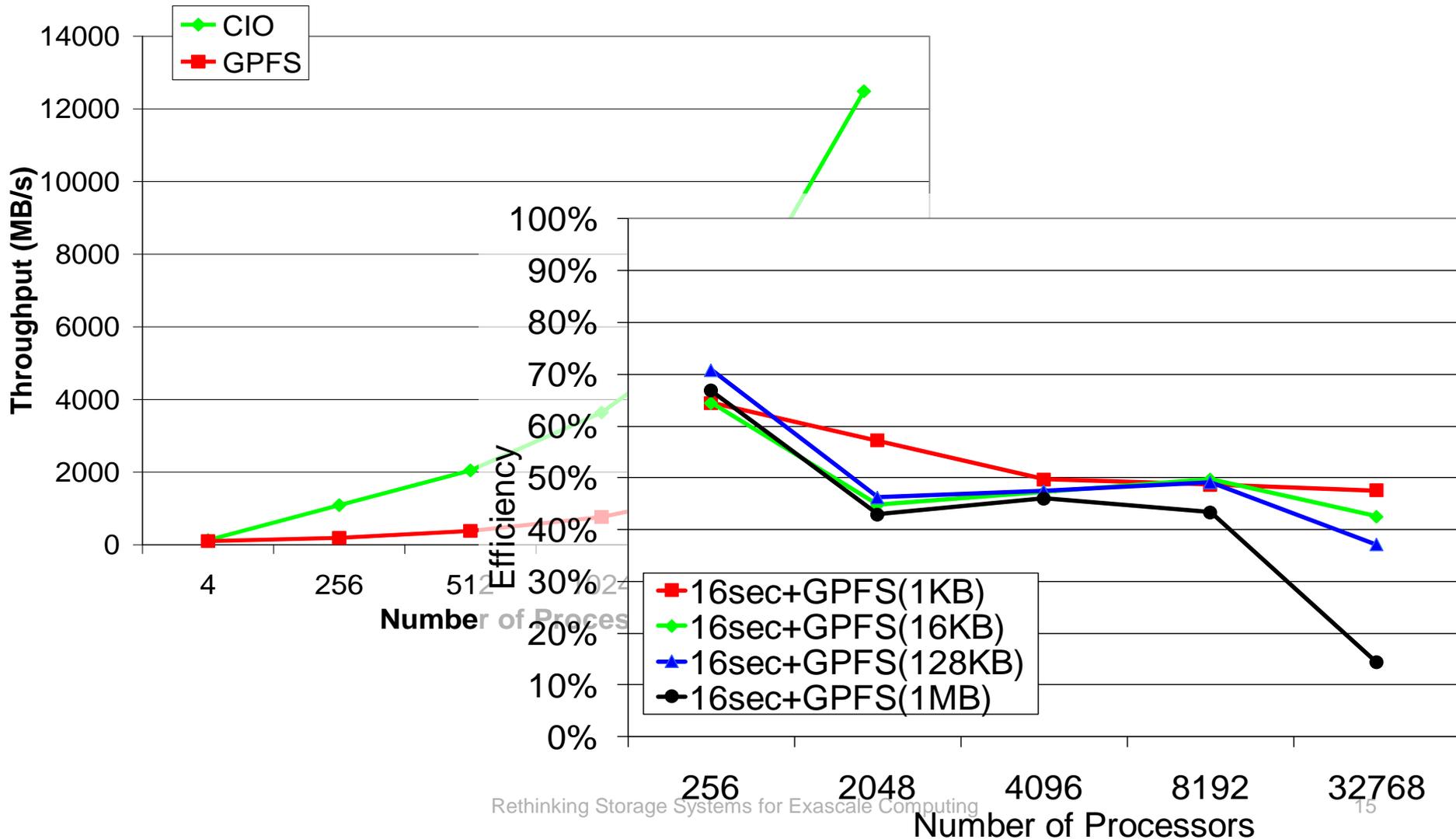
# Some Challenges

## Meta-data Operations on GPFS



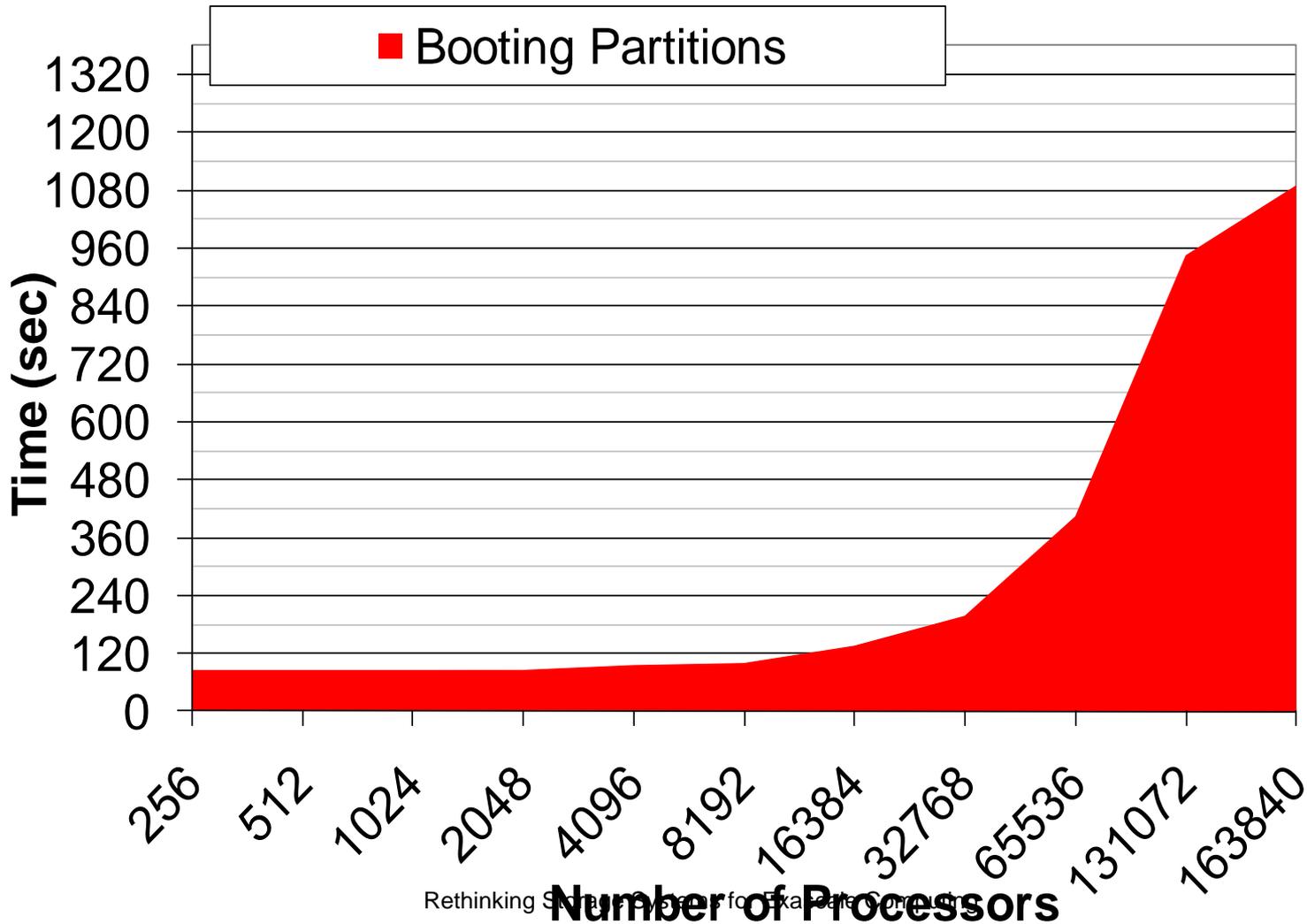
# Some Challenges

## Reading/Writing on GPFS



# Some Challenges

## Booting a IBM BlueGene/P



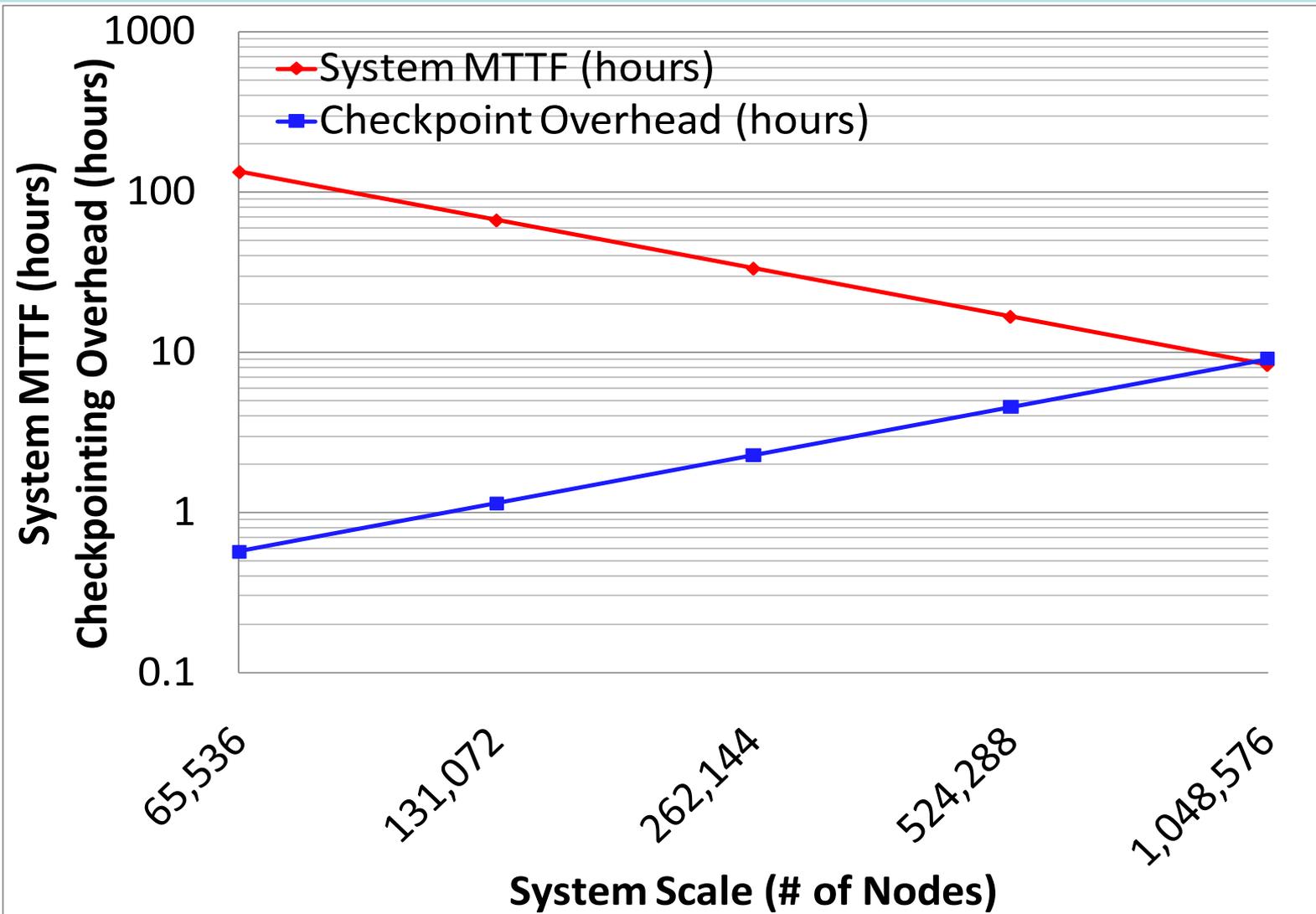
# Exascale Supercomputing Architecture

- Compute
  - 1M nodes, with ~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 might exist at every node

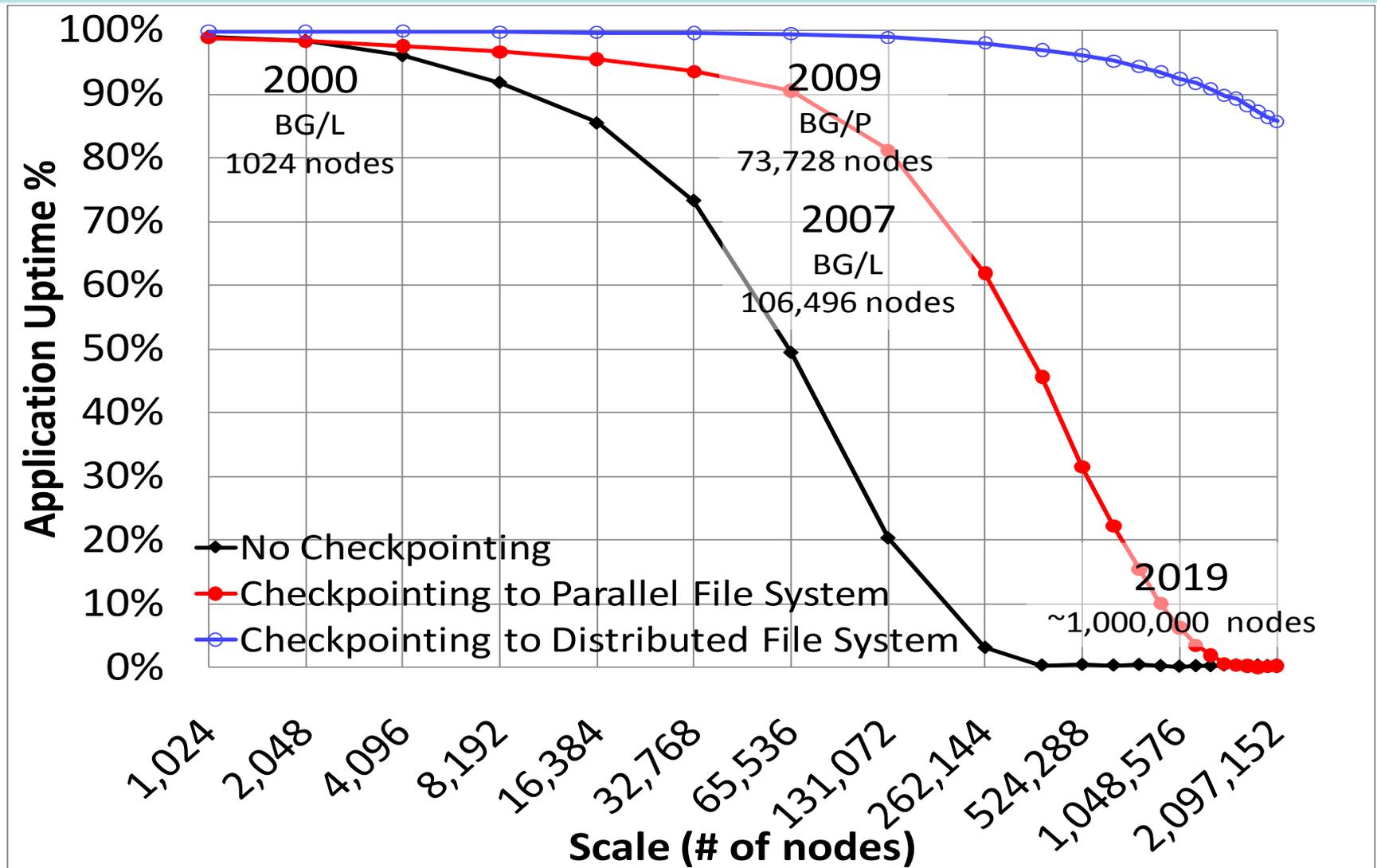
# Some Challenges to Overcome at Exascale Computing

- Programming paradigms
  - HPC is dominated by MPI today
  - Will MPI scale another 4 orders of magnitude?
  - MTC has better scaling properties (due to its asynchronous nature)
- Network topology must be used in job management, data management, compilers, etc
- Storage systems will need to become more distributed to scale

# Expected checkpointing cost and MTTF towards exascale



# Simulation application uptime towards exascale



# HEC FSIO 2008 Workshop Report

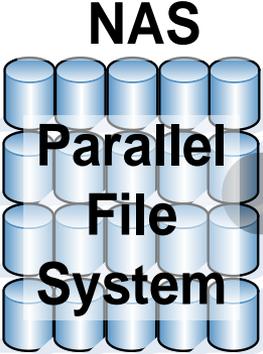
*Data path today is the same as the data path 20 years ago. There is a need for new technologies that will offer greater scalability for file system metadata... Novel approaches to I/O and file systems also need to be explored including redistribution of intelligence, user space file systems, data-aware file systems, and the use of novel storage devices... Most, if not all progress to date in parallel file systems in HEC has been evolutionary; what is lacking is revolutionary research; no fundamental solutions are being proposed. Revolutionary I/O technologies developments (both software and hardware) are needed to address the growing technology gap.*

# Research Directions

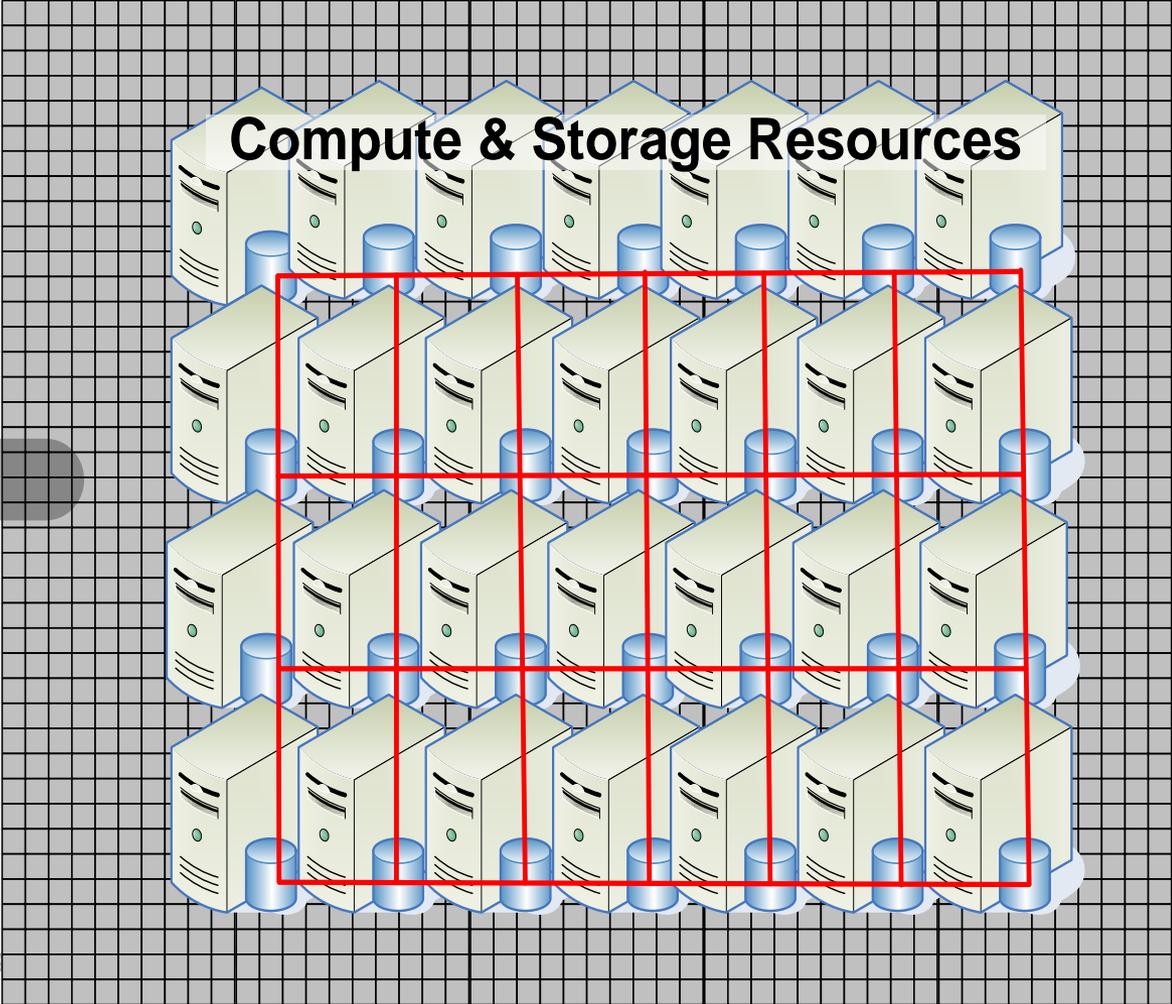
- ***Decentralization is critical***
  - Computational resource management (e.g. LRMs)
  - Storage systems (e.g. parallel file systems)
- ***Data locality must be maximized, while preserving I/O interfaces***
  - POSIX I/O on shared/parallel file systems ignore locality
  - Data-aware scheduling coupled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

# Storage System Architecture

## Network Fabric



Network Link(s)

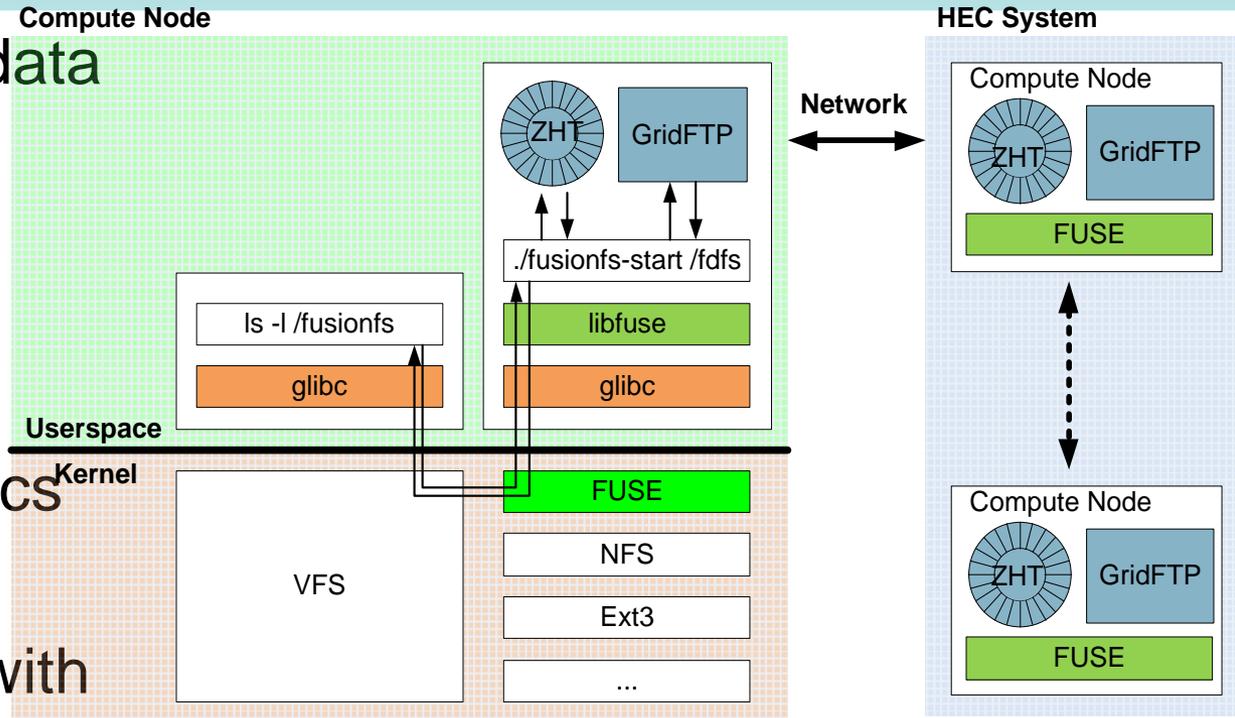


# FusionFS: Fusion Distributed File System

- ***Building on my own research (e.g. data-diffusion), parallel file systems (PVFS), and distributed file systems (e.g. GFS) → FusionFS, 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 (e.g. POSIX)
  - Must handle some workloads currently run on parallel file systems significantly better

# FusionFS Details

- Distributed Metadata Management
- Distributed Data Management
- Data Indexing
- Relaxed Semantics
- Data Locality
- Overlapping I/O with Computations
- POSIX



# FusionFS: Access Patterns

- **1-many read** (all processes read the same file and are not modified)
- **many-many read/write** (each process read/write to a unique file)
- **write-once read-many** (files are not modified after it is written)
- **append-only** (files can only be modified by appending at the end of files)
- **metadata** (metadata is created, modified, and/or destroyed at a high rate).

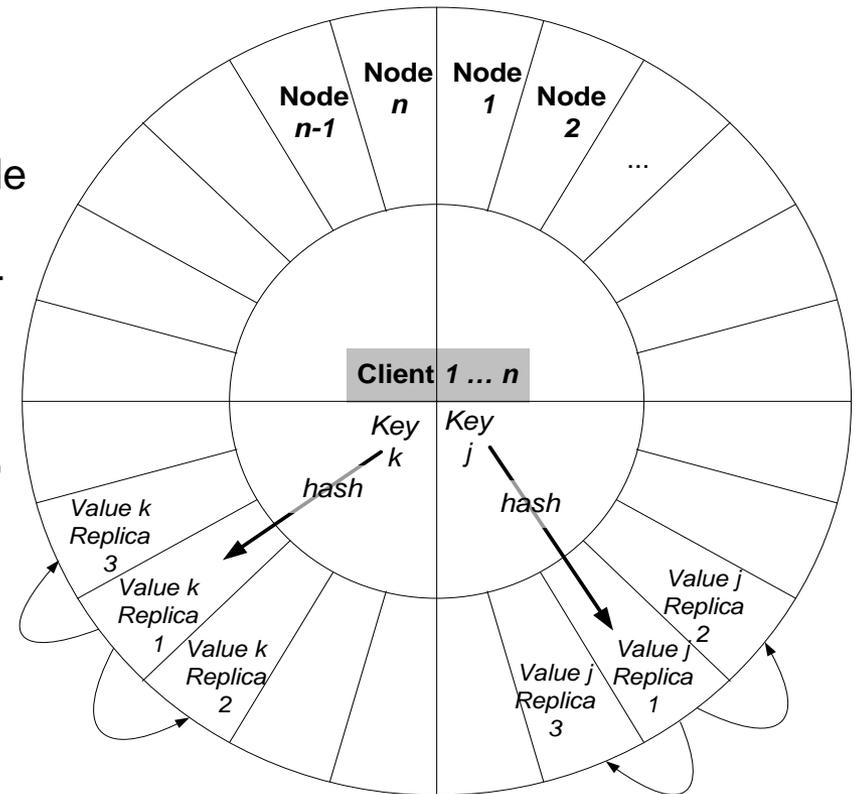
# FusionFS: Usage Scenarios

- **machine boot-up** (e.g. reading OS image on all nodes)
- **application loading** (e.g. reading scripts, binaries, and libraries on all nodes/processes)
- **common user data loading** (e.g. reading a common read-only database on all nodes/processes)
- **checkpointing** (e.g. writing unique files per node/process)
- **log writing** (writing unique files per node/process)
- **many-task computing** (each process reads some files, unique or shared, and each process writes unique files)

# ZHT:

## Zero Hop Distributed Hash Table

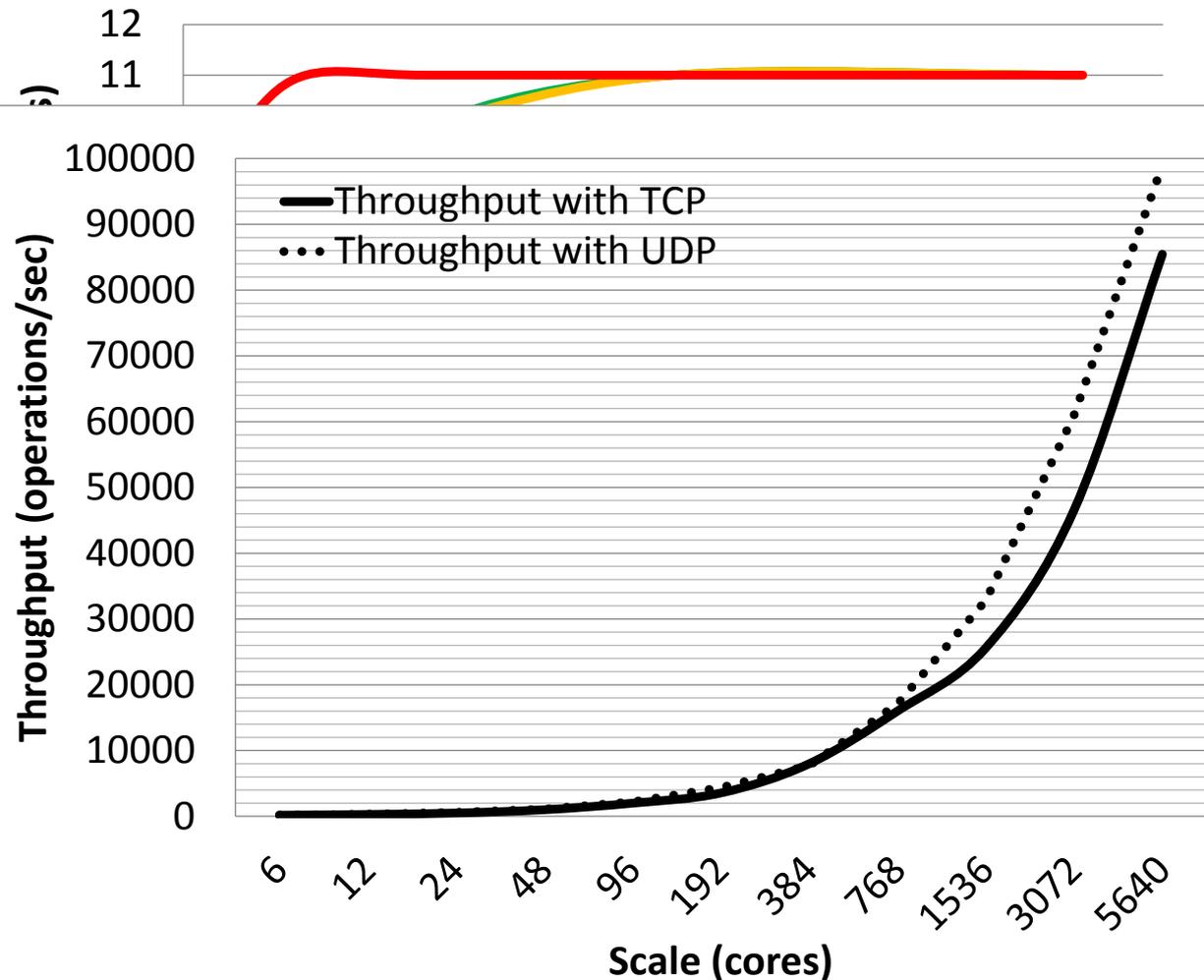
- Simplified distributed hash table tuned for the specific requirements of HEC
- Emphasized key features of HEC are:
  - Trustworthy/reliable hardware, fast network interconnects, non-existent node "churn", the requirement for low latencies, and scientific computing data-access patterns
- Primary goals:
  - Excellent availability and fault tolerance, with low latencies
- ZHT details:
  - Static membership function
  - Network topology aware node ID space
  - Replication and Caching
  - Efficient 1-to-all communication through spanning trees
  - Persistence



# ZHT Prototype

## Implementation and Performance

- C++/Linux
- Simple API
  - Insert, Find, Remove
- Communication
  - TCP & UDP
  - Evaluating MPI & BMI
- Hashing functions
  - SuperFastHash, FNVHash, alphaNumHash, BobJenkins, SDBM, CRC32, OneAtATimeHash
- Leverages other work
  - Google Buffer
  - Kyoto Cabinet



# Conclusions

- We MUST depart from traditional HEC architectures and approaches to reach exascales
- Scalable storage will open doors for novel research in programming paradigm shifts (e.g. MTC)
- This work has potential to touch every branch of computing, enabling efficient access, processing, storage, and sharing of valuable scientific data from many disciplines
  - Medicine, astronomy, bioinformatics, chemistry, aeronautics, analytics, economics, and new emerging computational areas in humanities, arts, and education
- Solutions for extreme scale HEC should be applicable to data centers and cloud infrastructures of tomorrow

# Main Message

- ***Preserving locality is critical!***
- *Segregating storage from compute resources is **BAD***
- *Parallel file systems + distributed file systems + distributed hash tables + nonvolatile memory*  
**→ new storage architecture for extreme-scale HEC**
- *Co-locating storage and compute is **GOOD***
  - *Leverage the abundance of processing power, bisection bandwidth, and local I/O*

# More Information

- **More information:**
  - <http://www.cs.iit.edu/~iraicu/index.html>
  - <http://datasys.cs.iit.edu/>
- **Relevant upcoming workshops and journals**
  - [DataCloud: IEEE Int. Workshop on Data-Intensive Computing in the Clouds](#) (at IPDPS), 2011
  - [HPDC/SigMetrics: HPDC/SIGMETRICS 2011 Student Poster Session](#), 2011
  - [JGC: Springer Journal of Grid Computing, Special Issue on Data Intensive Computing in the Clouds](#), 2011
  - [MTAGS: ACM Workshop on Many-Task Computing on Grids and Supercomputers \(at SC\)](#), 2011
  - [ScienceCloud: ACM Workshop on Scientific Cloud Computing \(at HPDC\)](#), [2010](#), [2011](#)
  - [SPJ: Scientific Programming Journal, Special Issue on Science-driven Cloud Computing](#), 2011
  - [TPDS: IEEE Transactions on Parallel and Distributed Systems, Special Issue on Many-Task Computing](#), 2011