Resource Management in Extreme Scales Distributed Systems

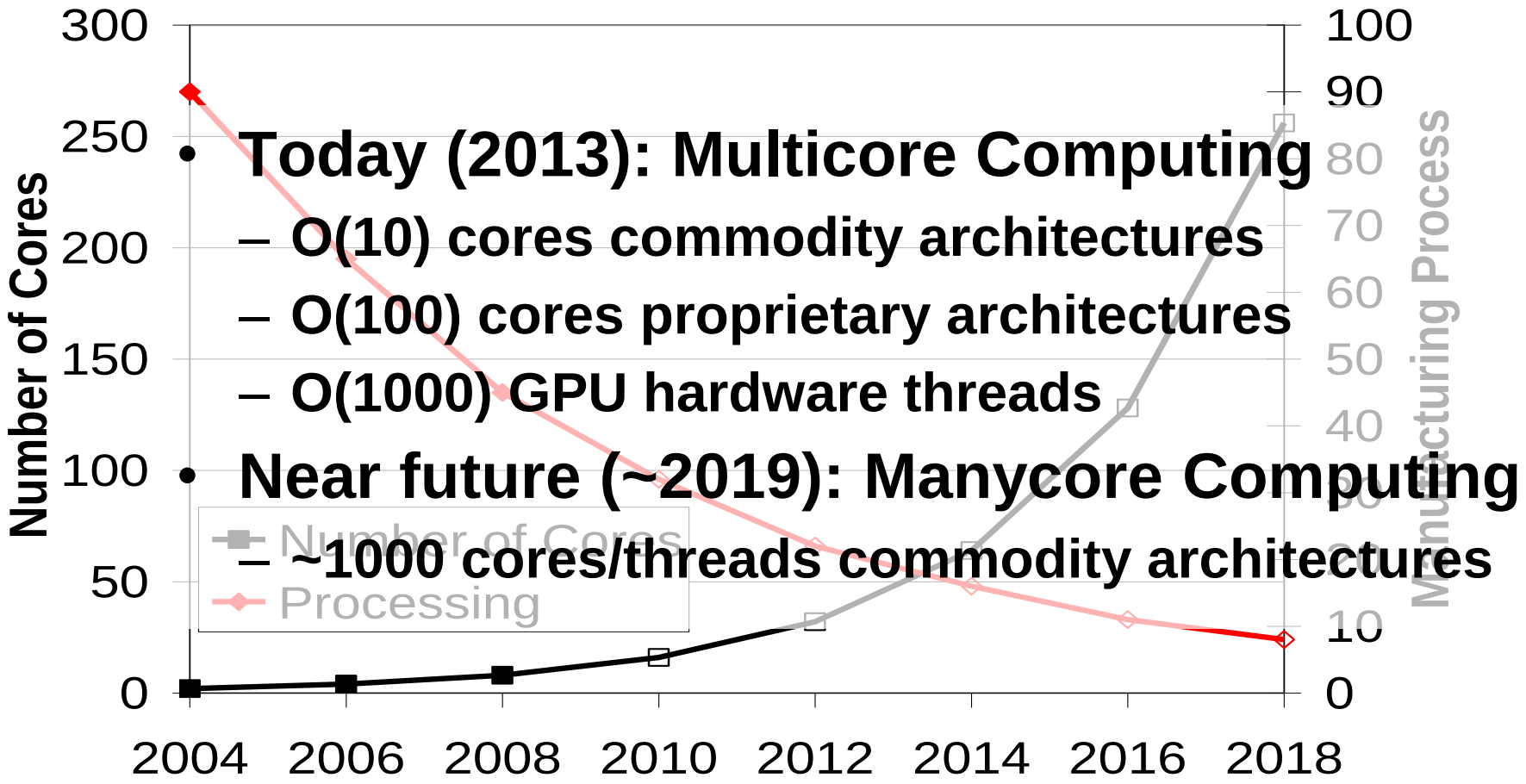
Ioan Raicu
Computer Science Department, Illinois Institute of Technology
Math and Computer Science Division, Argonne National Laboratory

August 12th, 2013
Los Alamos National Laboratory

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)
  - 6 PhD Students
  - 2 MS Students
  - 4 UG Students
- **Contact**
  - http://datasys.cs.iit.edu/
  - iraicu@cs.iit.edu

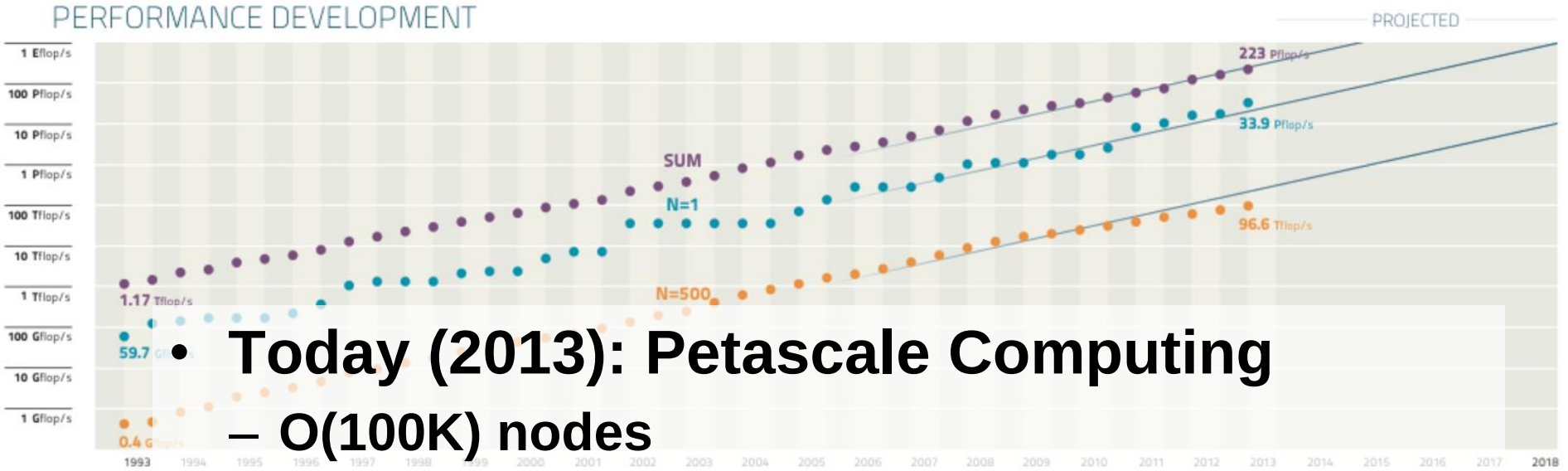
Manycore Computing



- **Today (2013): Multicore Computing**
  - **O(10) cores commodity architectures**
  - **O(100) cores proprietary architectures**
  - **O(1000) GPU hardware threads**
- **Near future (~2019): Manycore Computing**
  - **~1000 cores/threads commodity architectures**

Pat Helland, Microsoft, The Irresistible Forces Meet the Movable Objects, November 9th, 2007

Exascale Computing



- **Today (2013): Petascale Computing**
  - **O(100K) nodes**
  - **O(1M) cores**

http://www.top500.org/

- **Near future (~2018): Exascale Computing**
  - **~1M nodes (10X)**
  - **~1B processor-cores/threads (1000X)**

http://s.top500.org/static/lists/2013/06/TOP500_201306_Poster.png

Exascale Computing 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 3 orders of magnitude?
  - Other paradigms (including loosely coupled ones) might emerge to be more flexible, resilient, and scalable
- Storage systems will need to become more distributed to scale ➔ Critical for resilience of HPC
- Network topology must be used in job management, data management, compilers, etc
- Power efficient compilers and run-time systems

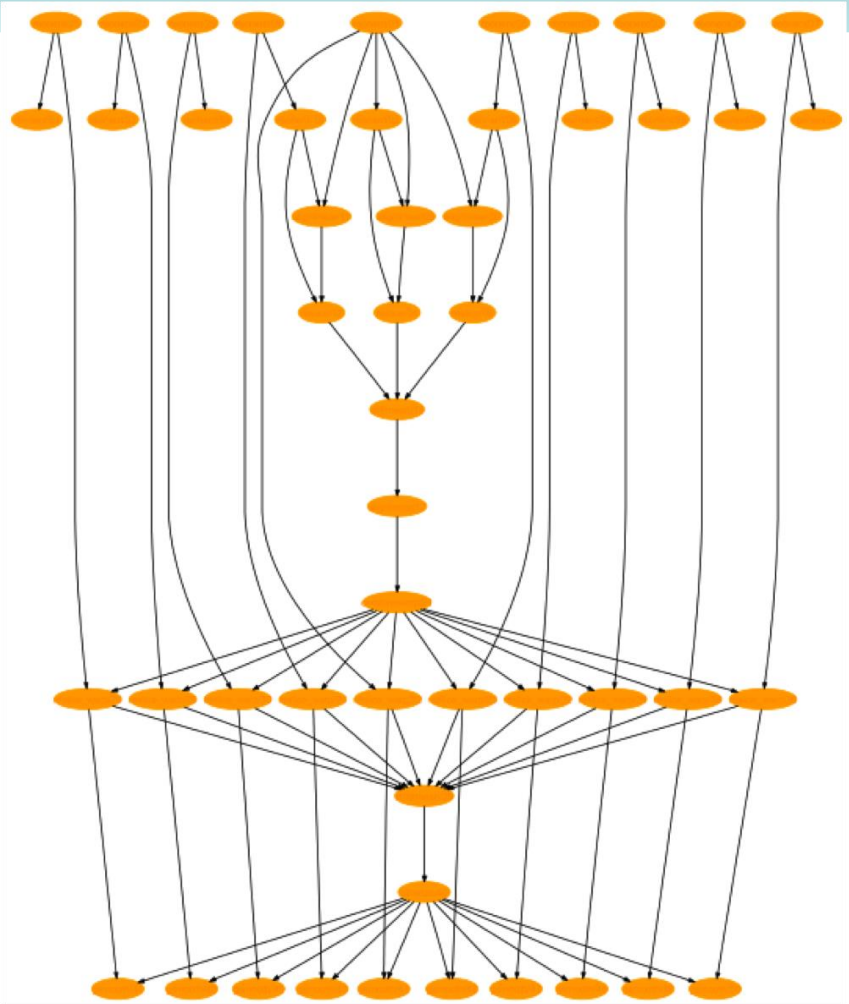
Main Message

- ***Decentralization is critical***
  - Computational resource management (e.g. LRMs)
  - Storage systems (e.g. parallel file systems)
- ***Preserving locality is critical!***
  - 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
- *Co-locating storage and compute is **GOOD***
  - Leverage the abundance of processing power, bisection bandwidth, and local I/O

Critical Technologies Needed to achieve Extreme Scales

- Fundamental Building Blocks (with a variety of resilience and consistency models)
  - Distributed hash tables (aka NoSQL data stores)
  - Distributed Message Queues
- Deliver future generation distributed systems
  - Global File Systems, Metadata, and Storage
  - Job Management Systems
  - Workflow Systems
  - Monitoring Systems
  - Provenance Systems
  - Data Indexing

Many-Task Computing (MTC)



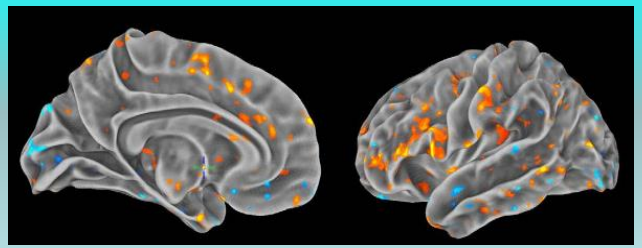
MTC emphasizes:

- bridging HPC/HTC
- many resources
  - short period of time
- many computational tasks
- dependent/independent tasks
- tasks organized as DAGs
- primary metrics are seconds

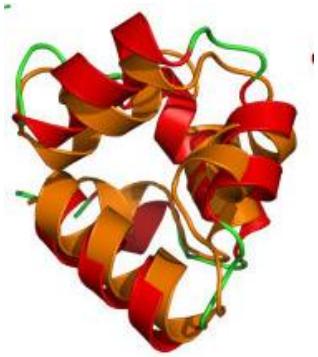
Advantages:

- Improve fault tolerant
- Maintain efficiency
- Programmability & Portability
- support embarrassingly parallel and parallel applications

Swift/T and Applications



- Swift/T
  - Active research project (CI UChicago & ANL)
  - Parallel Programming Framework
  - Throughput ~25k tasks/sec per process
  - Shown to scale to 128k cores



- Application Domains Supported
  - Astronomy, Biochemistry, Bioinformatics, Economics, Climate

Swift lets you write parallel scripts that run many copies of ordinary programs concurrently, using statements like this:

```
foreach protein in proteinList {
  runBLAST(protein);
}
```

Images from Swift Case Studies - http://www.ci.uchicago.edu/swift/case_studies/

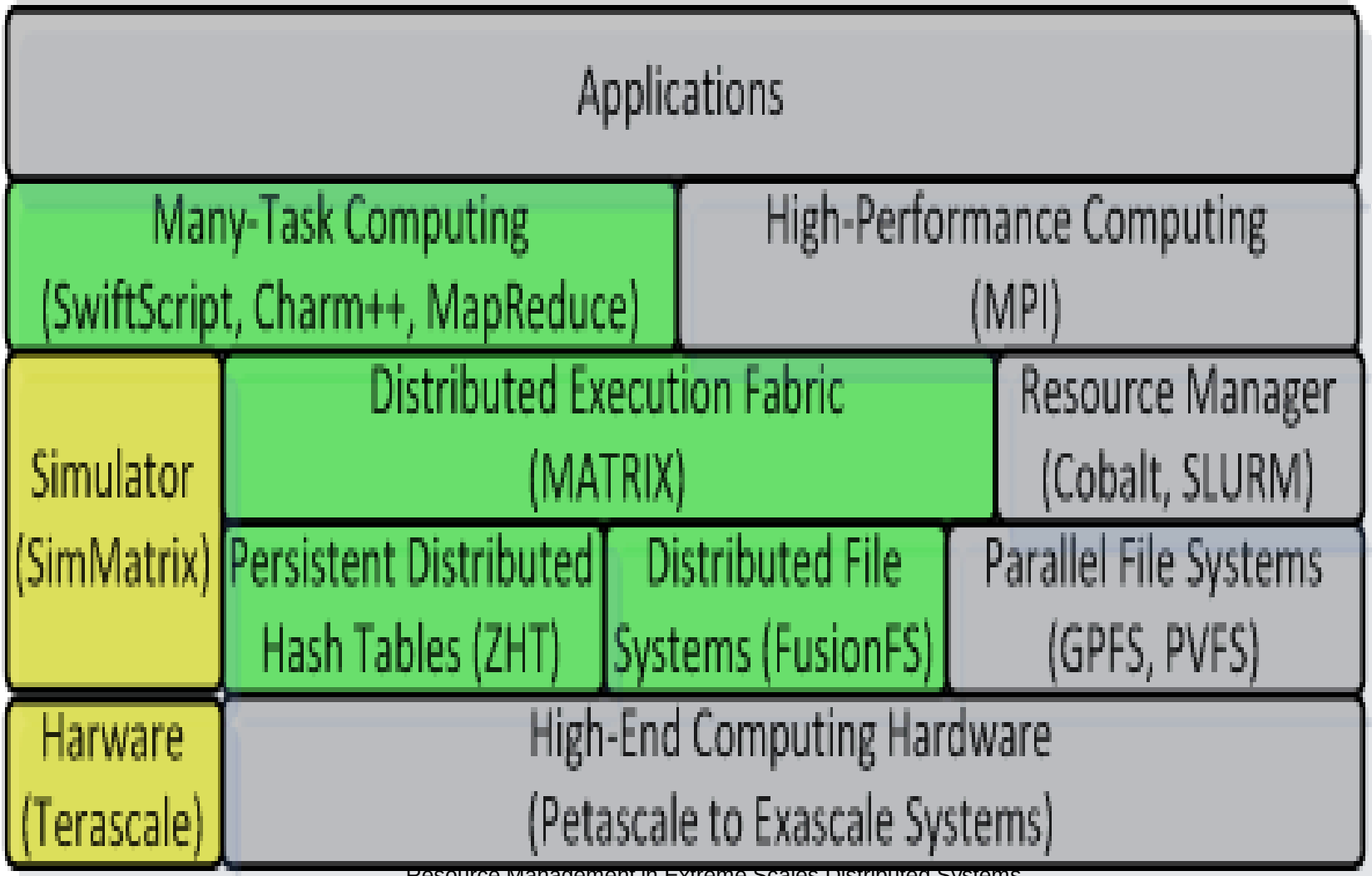
Swift Applications

Field	Description	Characteristics	Status
Astronomy	Creation of montages from many digital images	Many 1-core tasks, much communication, complex dependencies	E
Astronomy	Stacking of cutouts from digital sky surveys	Many 1-core tasks, much communication	E (Falkon)
Biochemistry	Analysis of mass-spec data for post-translational protein modifications	10,000 – 100,000 K jobs for proteomic searches using custom serial codes	D
Biochemistry	Protein folding using iterative fixing algorithm, also exploring other biomolecule interactions	100s to 1000s of 1-1000 core simulations & data analysis	O
Biochemistry	Identification of drug targets via computational screening	Up to 1M x 1 core	O (Falkon)
Bioinformatics	Metagenome modeling	1000's of 1-core integer programming problems	D
Business economics	Mining of large text corpora to study media bias	Analysis and comparison of 70M+ text files of news articles	D
Climate	Ensemble climate model runs and analysis of output data	10s to 100s of 100-1000 core simulations	E
Economics	Generation of response surfaces for various economic models	1K to 1M 1-core runs (10K typical), then data analysis	O
Neuroscience	Analysis of functional MRI datasets	Comparison of images; connectivity analysis with SEM, many tasks (100K+)	O
Radiology	Training of computer aided diagnosis algorithms	Comparison of images; many tasks, much communication	D
Radiology	Image processing and brain mapping for neurosurgical planning research	1000's of MPI application executions	O

Active Projects

- **Storage**
  - FusionFS: Fusion distributed File System
    - HyCache, FusionProv, IStore, RXSim
  - ZHT: Zero-Hop Distributed Hash Table
    - NoVoHT
- **Computing**
  - **Many-Task Computing**
    - MATRIX: MAny-Task computing execution fabRIc at eXascales
      - SimMatrix
    - Falkon: Fast and Light-weight tasK executiON framework
      - FalkonCloud
    - Swift: Fast, Reliable, Loosely Coupled Parallel Computation
  - **Many-Core Computing**
    - GeMTC: Virtualizing GPUs to Support MTC Applications
  - **Cloud Computing**
    - CloudBench: Optimizing Cloud Infrastructure for Scientific Computing Applications

Proposed Software Stack in Large-Scale Distributed Systems



ZHT Project

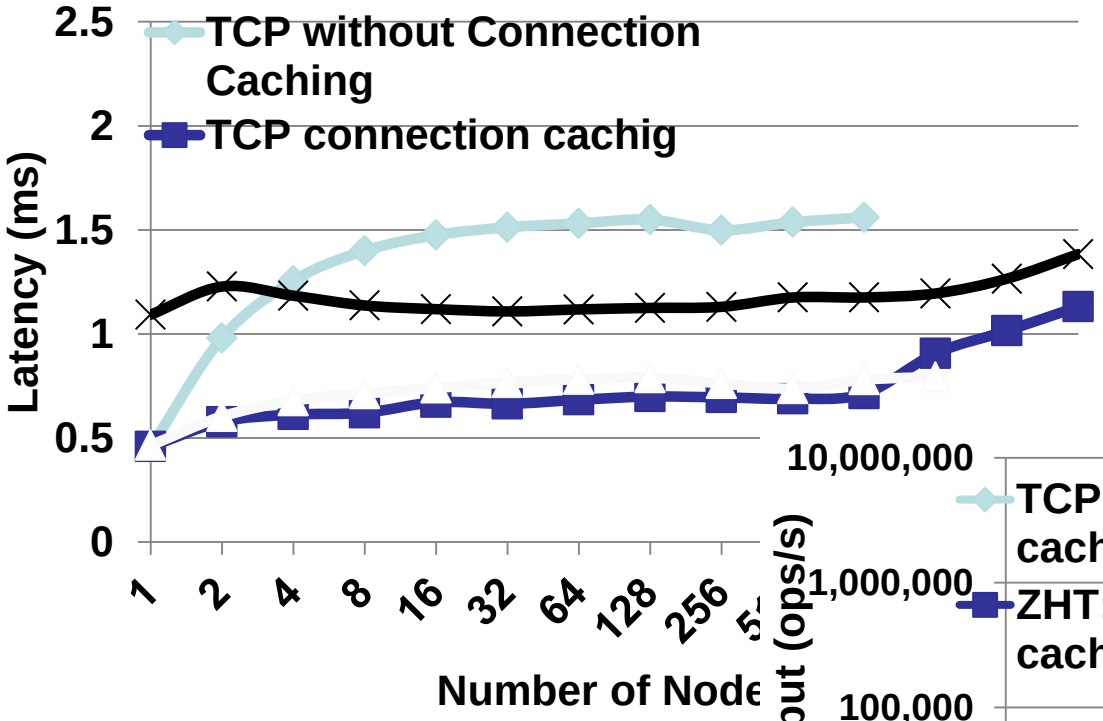
- ZHT: A distributed Key-Value store
  - Light-weighted
  - High performance
  - Scalable
  - Dynamic
  - Fault tolerant
  - Strong Consistency
  - Persistent
  - Versatile: works from clusters, to clouds, to supercomputers

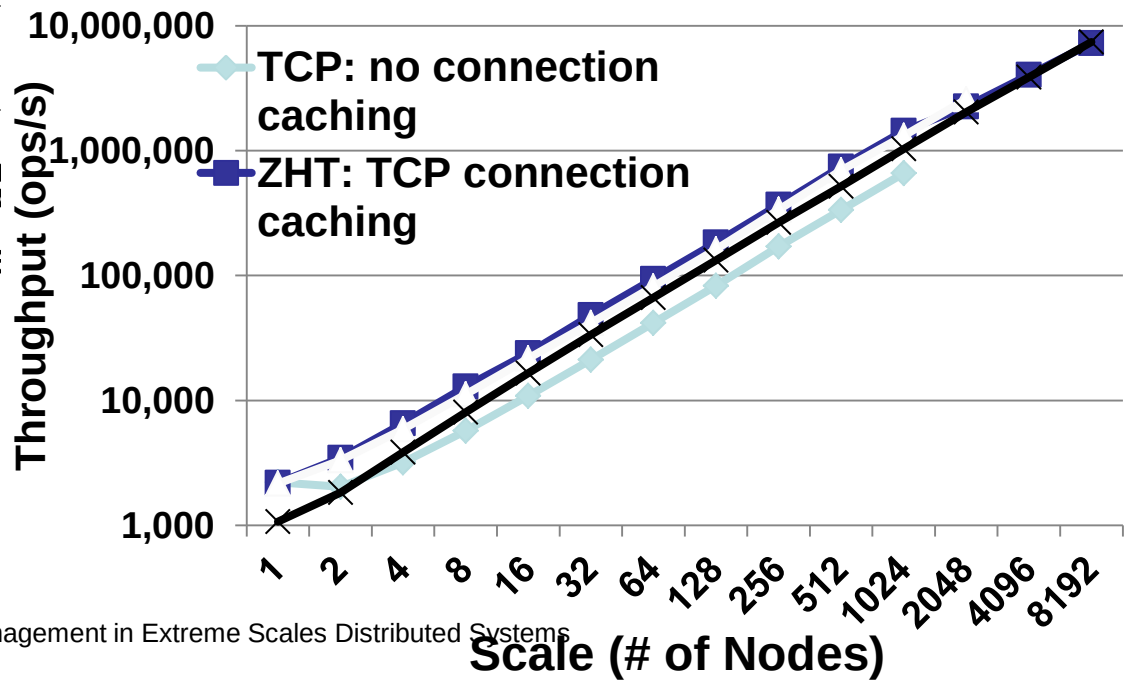
ZHT Project

- Many DHTs: Chord, Kademlia, Pastry, Cassandra, C-MPI, Memcached, Dynamo
- Why another?

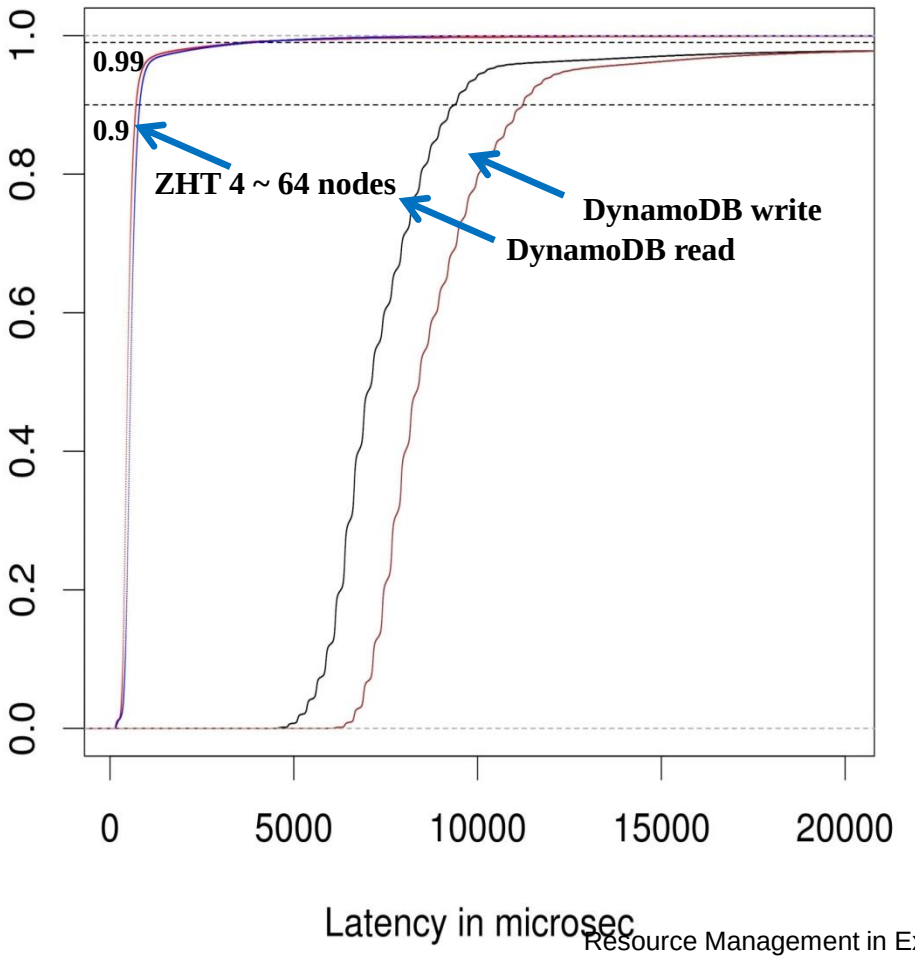
Name	Impl.	Routing Time	Persistence	Dynamic membership	Append Operation
Cassandra	Java	Log(N)	Yes	Yes	No
C-MPI	C	Log(N)	No	No	No
Dynamo	Java	0 to Log(N)	Yes	Yes	No
Memcached	C	0	No	No	No
ZHT	C++	0 to 2	Yes	Yes	Yes

ZHT Project


2.5
2
1.5
1
0.5
0
Latency (ms)
TCP without Connection Caching
TCP connection cachig
1
2
4
8
16
32
64
128
256
Number of Node


10,000,000
1,000,000
100,000
10,000
1,000
Throughput (ops/s)
TCP: no connection caching
ZHT: TCP connection caching
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
Scale (# of Nodes)

ZHT Project



ZHT on cc2.8xlarge instance
8 s-c pair/instance

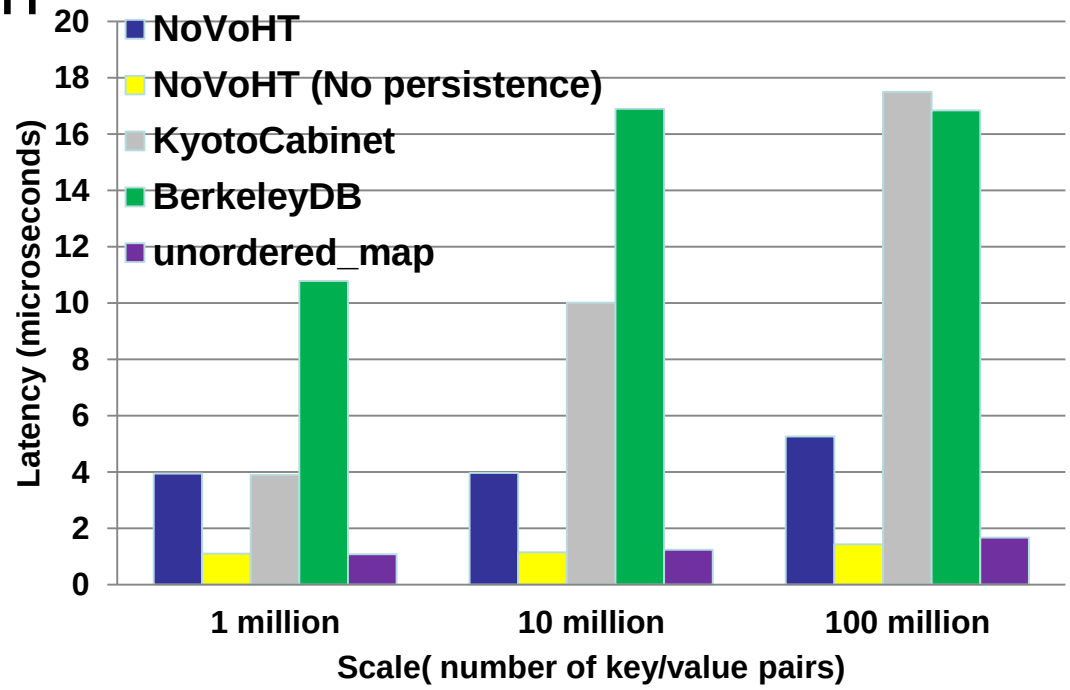
Scales	75%	90%	95%	99%	Avg	Throughput
8	186	199	214	260	172	46421
32	509	603	681	1114	426	75080
128	588	717	844	2071	542	236065
512	574	708	865	3568	608	841040

DynamoDB: 8 clients/instance

Scales	75%	90%	95%	99%	Avg	Throughput
8	11942	13794	20491	35358	12169	83.39
32	10081	11324	12448	34173	9515	3363.11
128	10735	12128	16091	37009	11104	11527
512	9942	13664	30960	38077	28488	Error

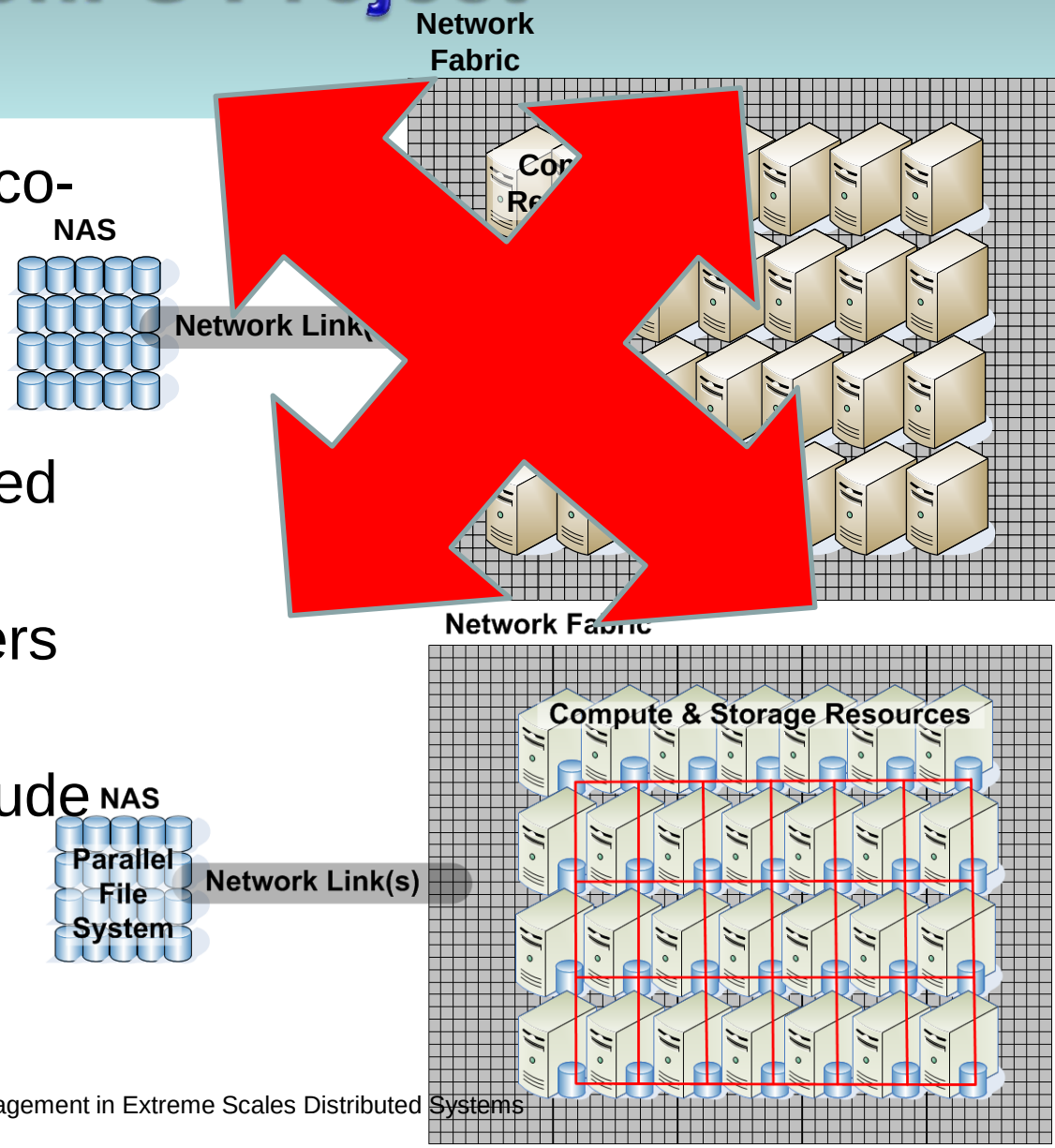
NoVoHT Project

- NoVoHT
  - Persistent in-memory hash map
  - Append operation
  - Live-migration

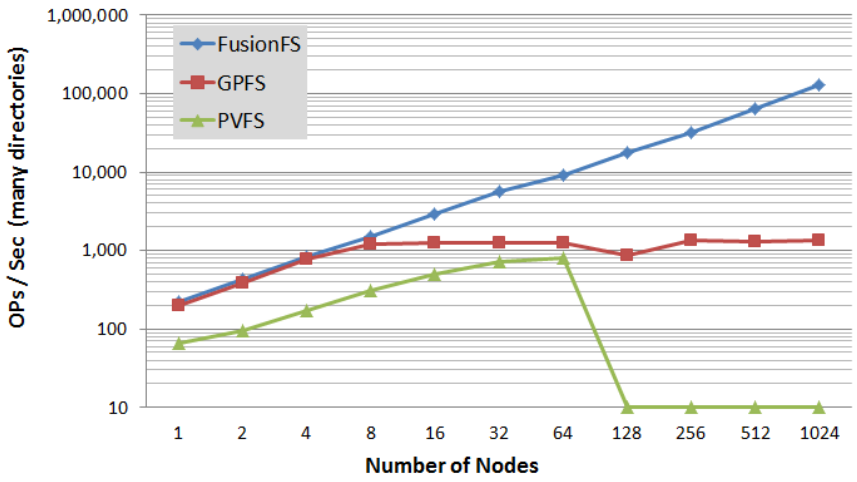


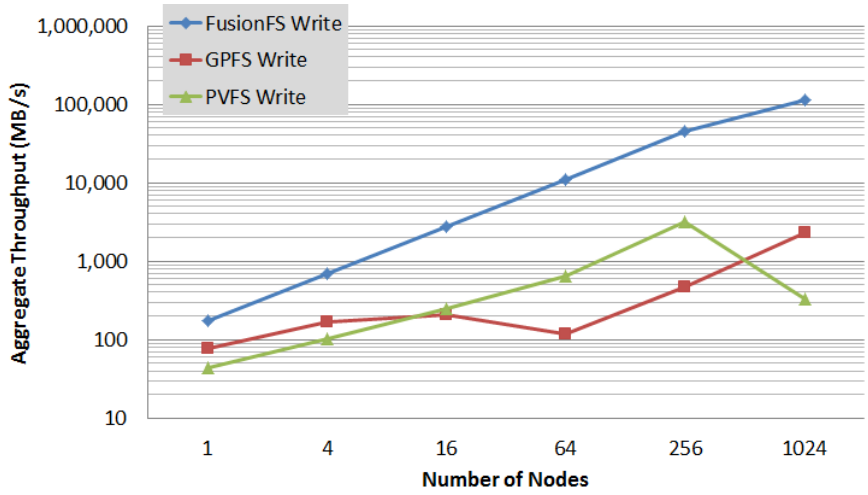
FusionFS Project

- A distributed file system co-locating storage and computations, while supporting POSIX
- Everything is decentralized and distributed
- Aims for millions of servers and clients scales
- Aims at orders of magnitude higher performance than current state of the art parallel file systems



FusionFS Project





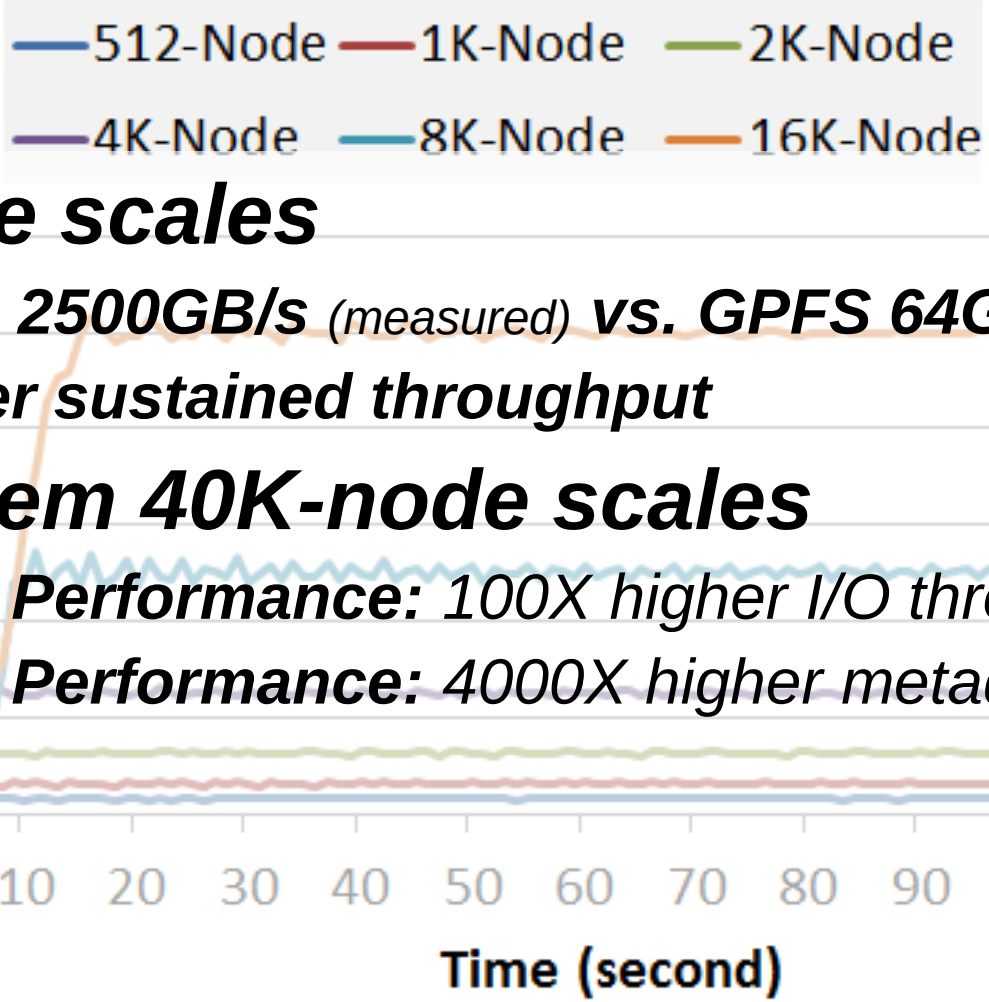
^ ~2 orders of magnitude faster metadata

This gap will grow even larger as parallel filesystems saturate external network – expected gap will be ~4 orders of magnitude faster performance

^ ~1.5 order of magnitude faster I/O

< ~1.5 order of magnitude faster runtime for real application

FusionFS Project



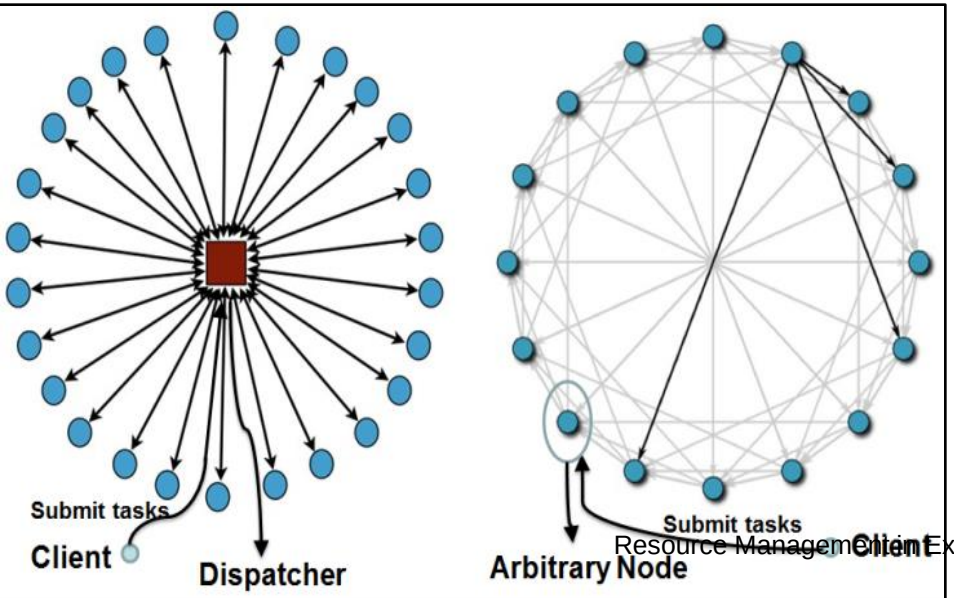
- ***16K-node scales***
  - ***FusionFS 2500GB/s*** *(measured)* ***vs. GPFS 64GB/s*** *(theoretical)*
  - ***39X higher sustained throughput***
- ***Full system 40K-node scales***
  - ***Expected Performance:*** *100X higher I/O throughput*
  - ***Expected Performance:*** *4000X higher metadata ops/sec*

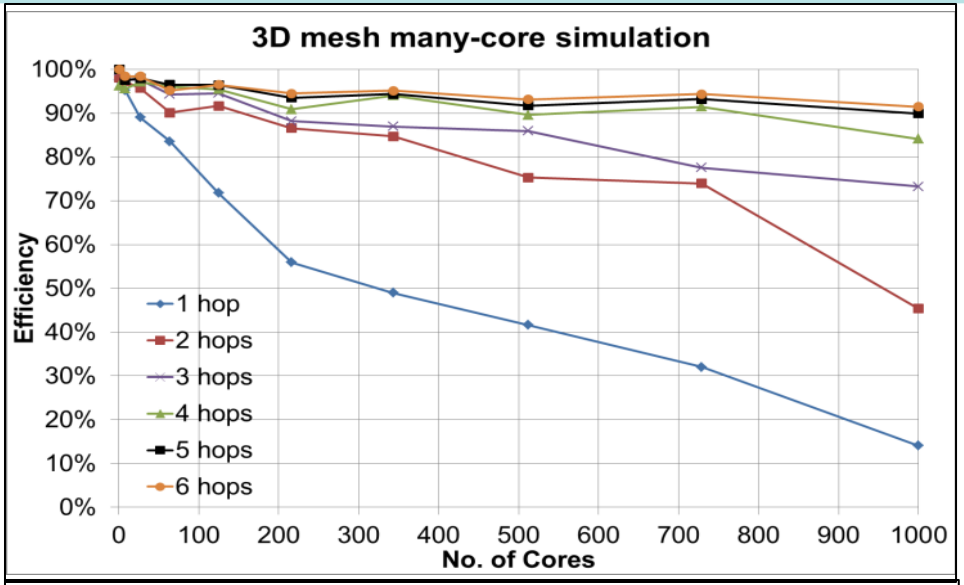
FusionFS Project

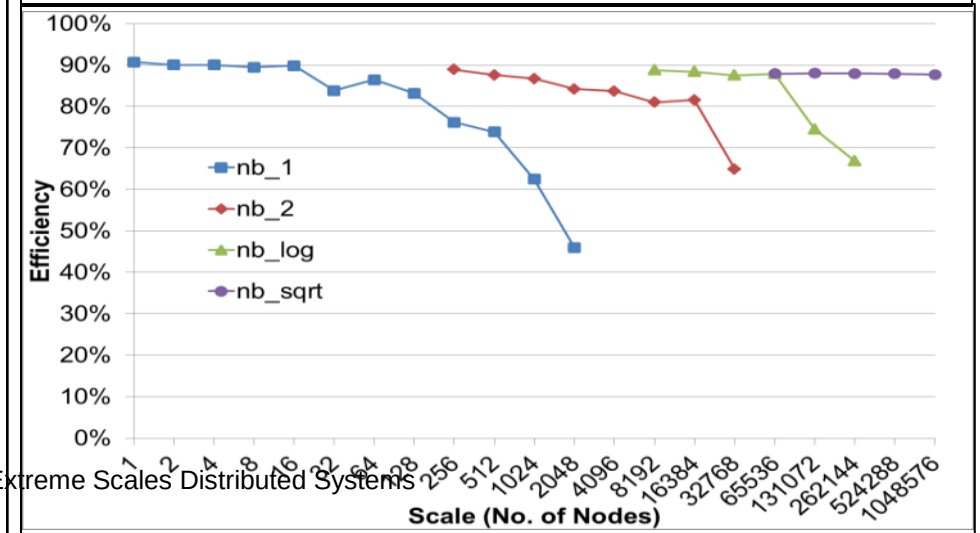
- Many sub-projects
  - Provenance (FusionProv) – uses ZHT
  - Information Dispersal Algorithms (IStore) – uses GPUs
- Other relevant Projects (planning to integrate into FusionFS)
  - SSD+HHD hybrid caching (HyCache)
  - Data Compression
- Improvements on the horizon
  - Non-POSIX interfaces (e.g. Amazon S3)
  - Explore viability of supporting HPC checkpointing
  - Deep indexing and search

SimMatrix Project

- Light-weight simulator to study MTC scheduling algorithms at exascale levels and on many-core architectures





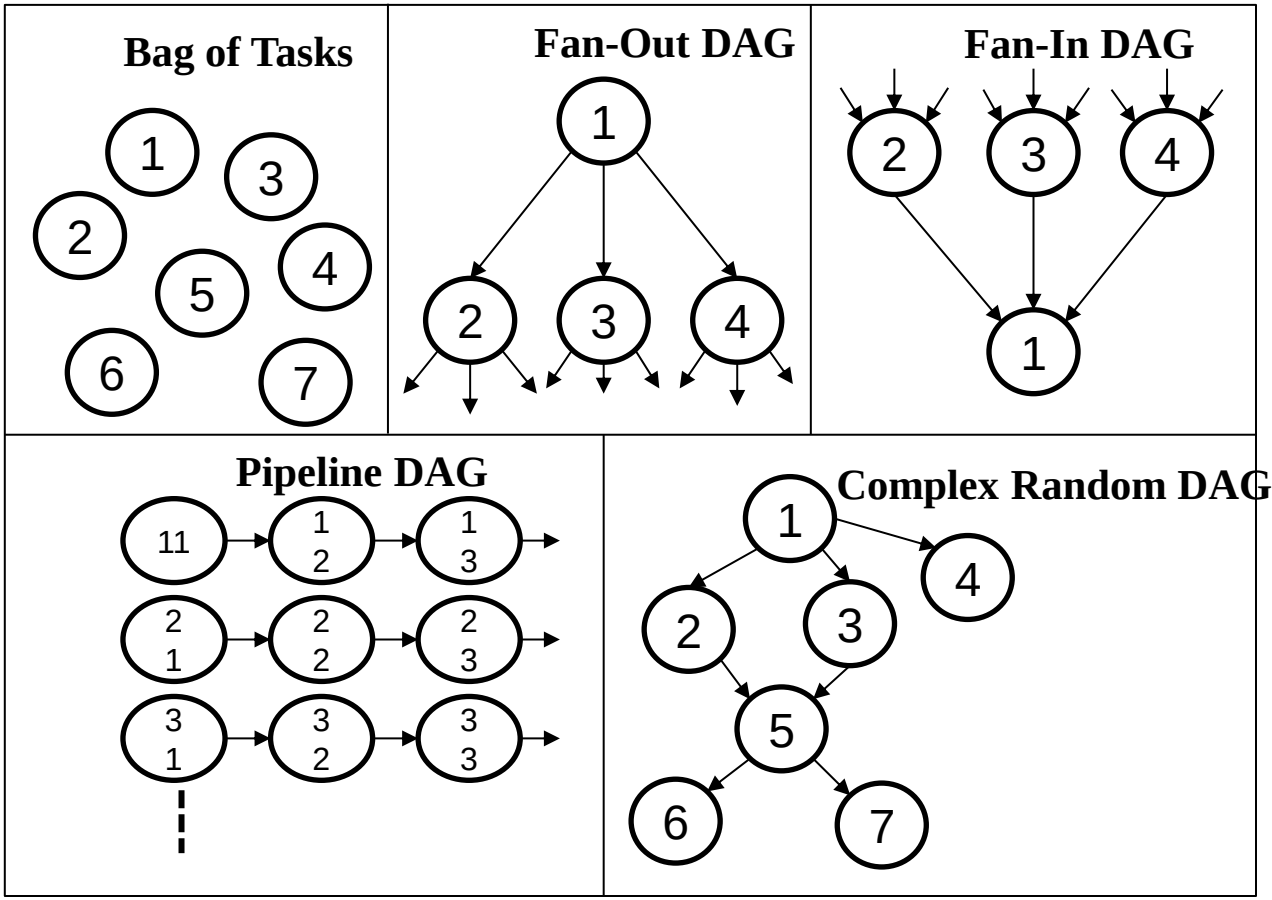


MATRIX Project

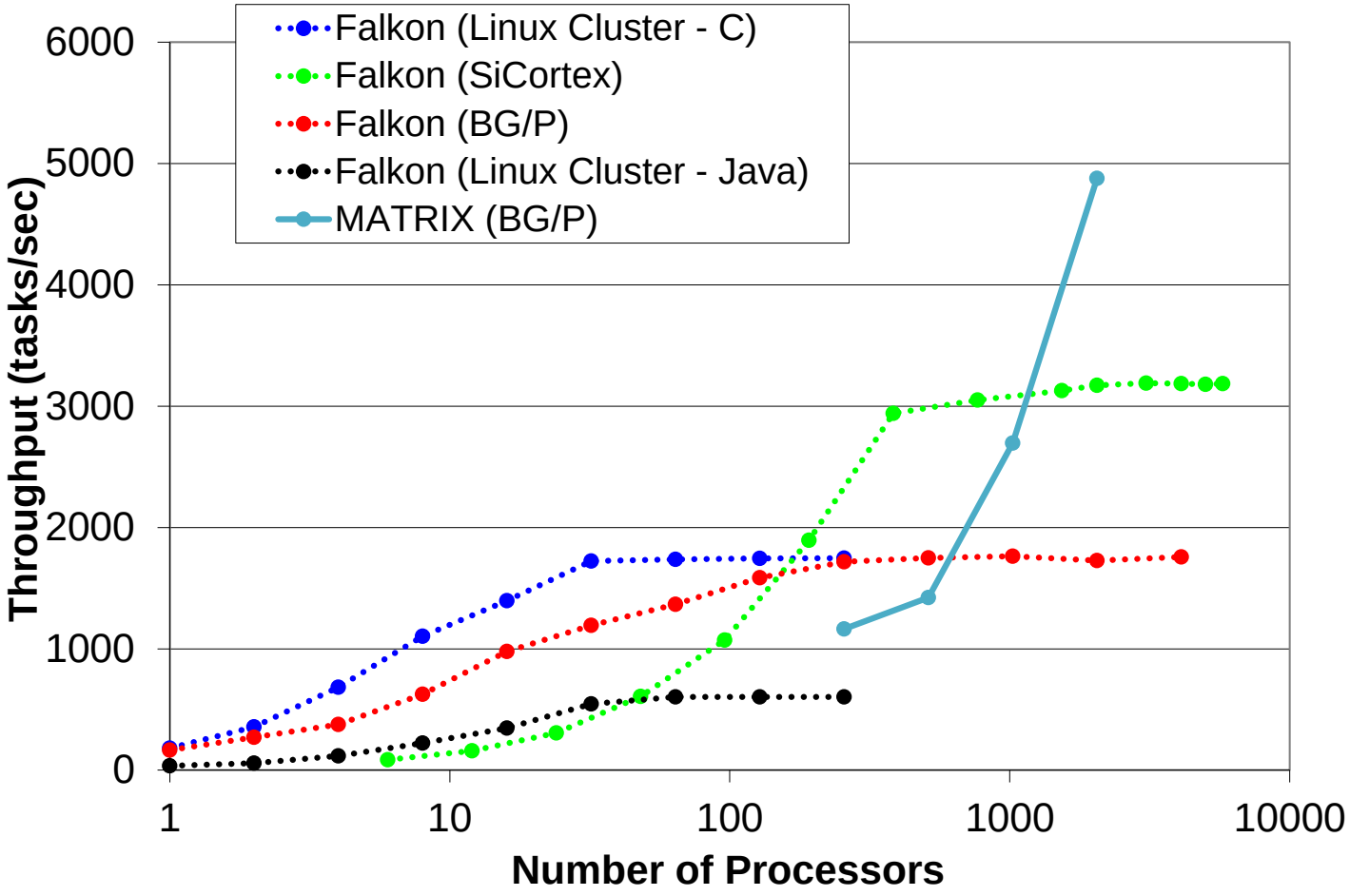
- MATRIX - distributed MTC execution framework for distributed load balancing using Work Stealing algorithm
  - Distributed scheduling is an efficient way to achieve load balancing, leading to high job throughput and system utilization
  - Dynamic job scheduling system at the granularity of node/core levels for extreme scale applications

MATRIX Project: Workloads

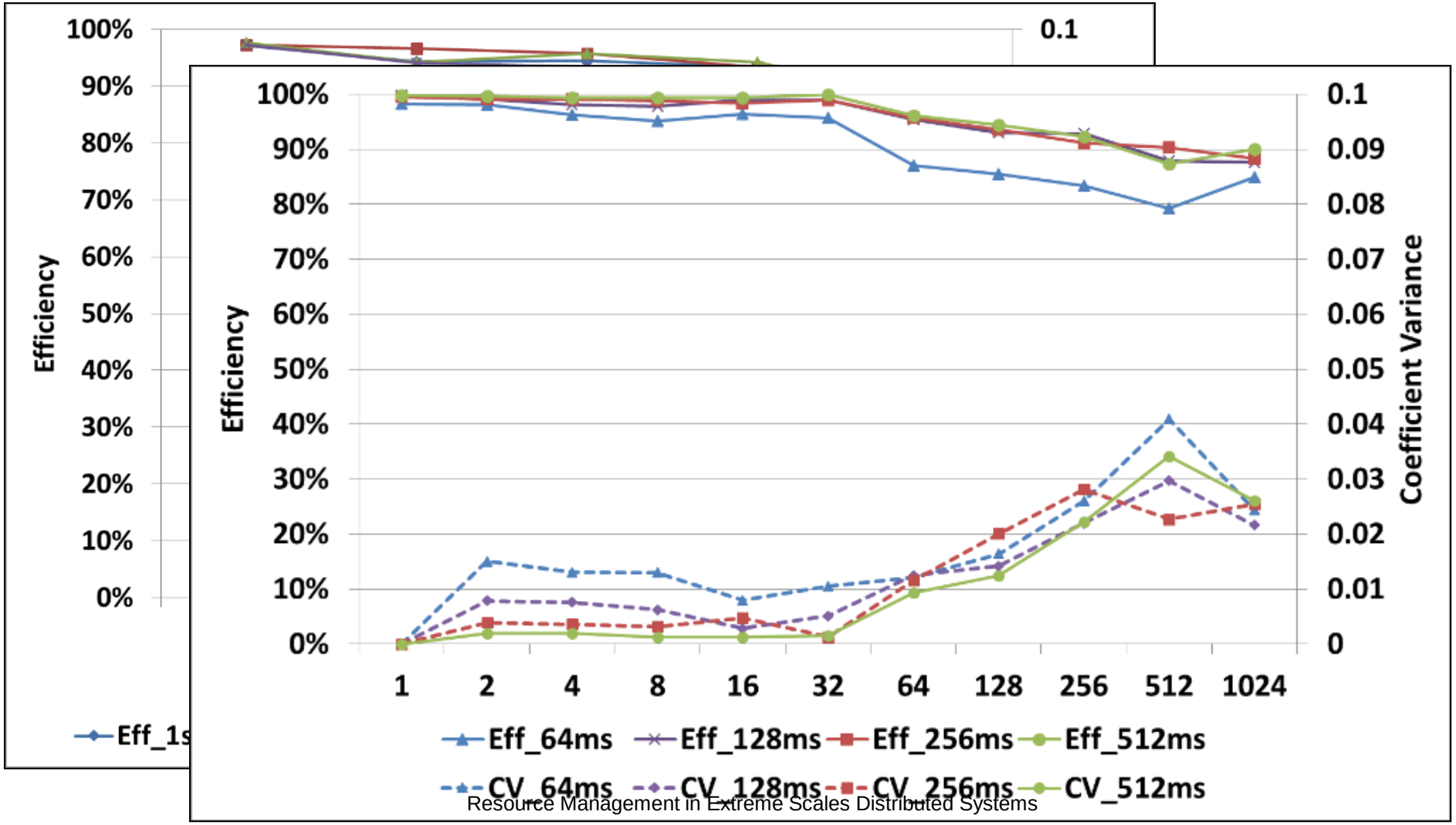
- Bag of Tasks
- Fan-In DAG
- Fan-Out DAG
- Pipeline DAG
- Complex Random DAG



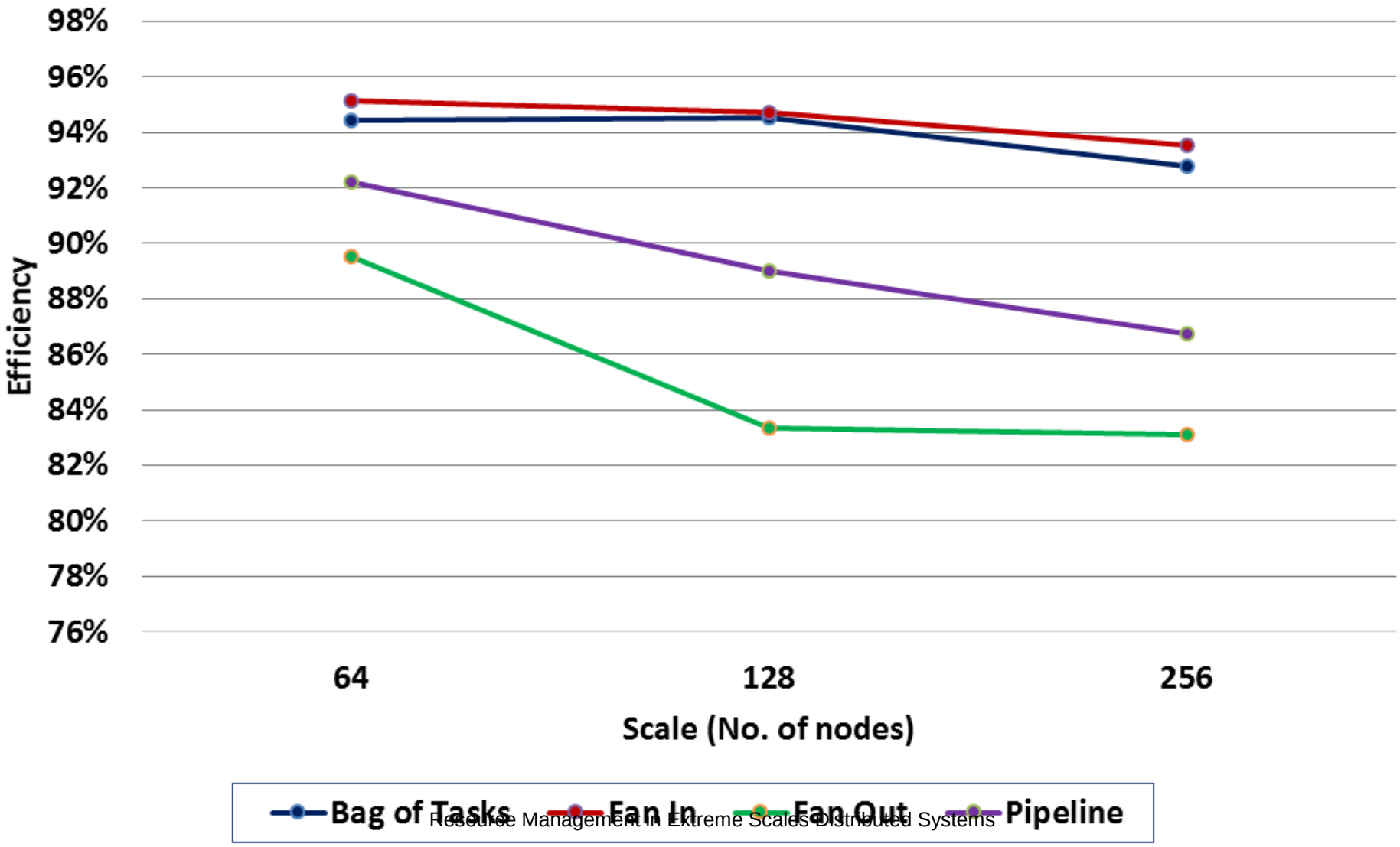
MATRIX Project



MATRIX Project



MATRIX Project



GeMTC Project

GPU

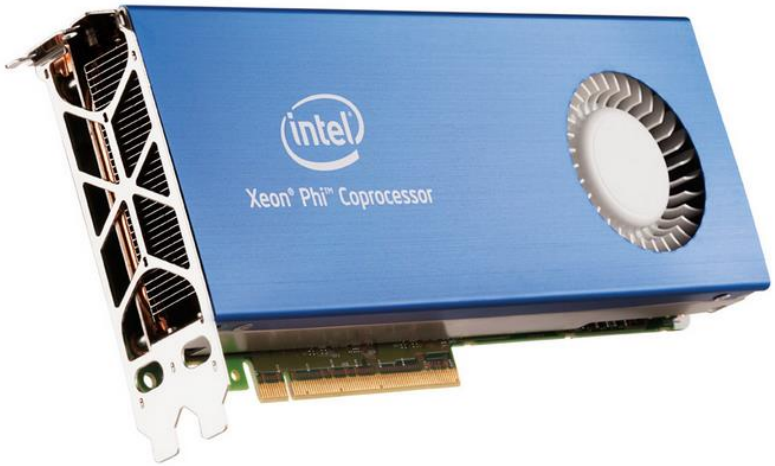
- Streaming Multiprocessors (15 SMXs on Kepler K20)
- 192 warps * 32 threads

Coprocessors

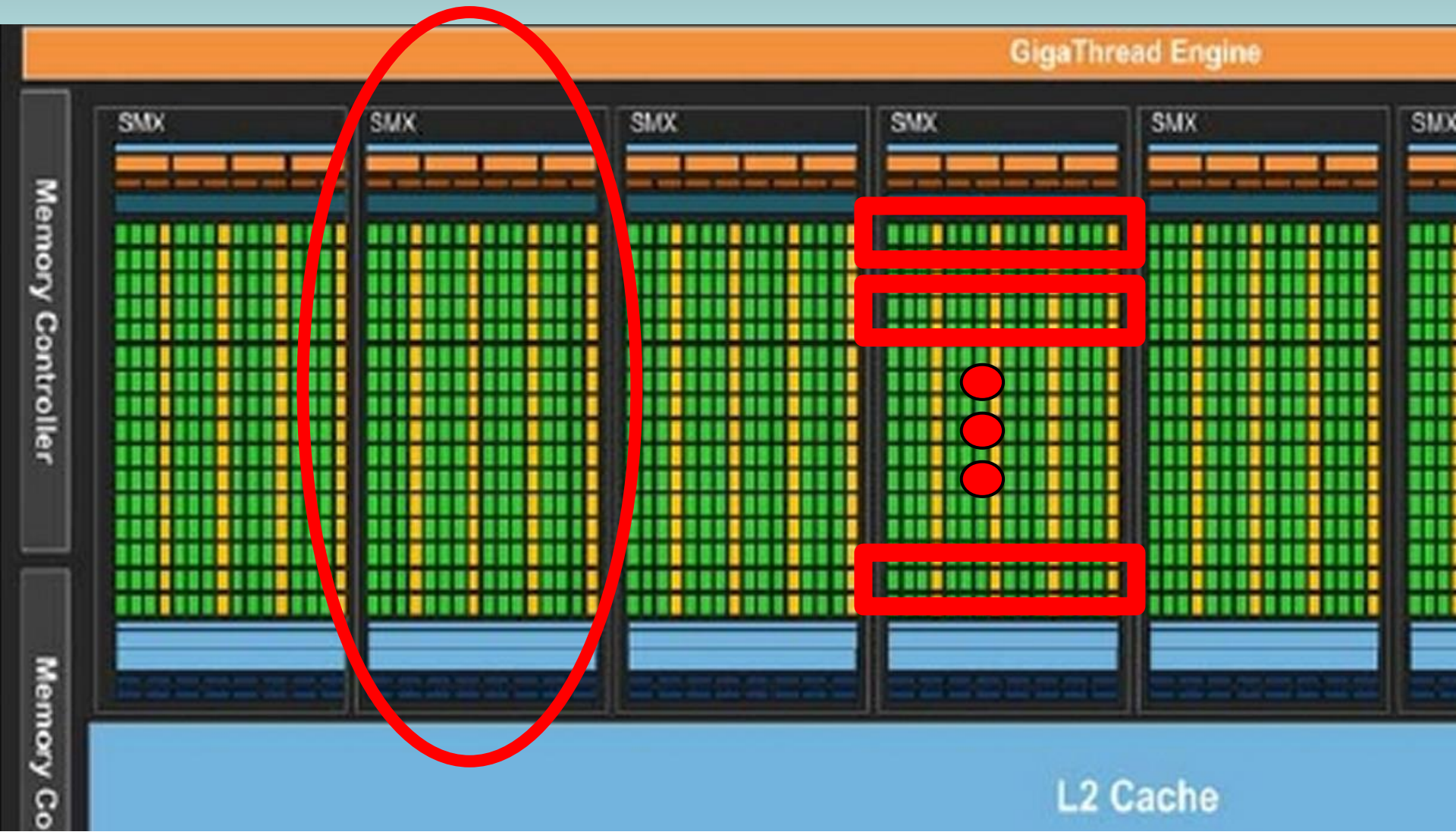
- Intel Xeon Phi
- 60 cores * 4 threads per core = 240 hardware threads

GeMTC

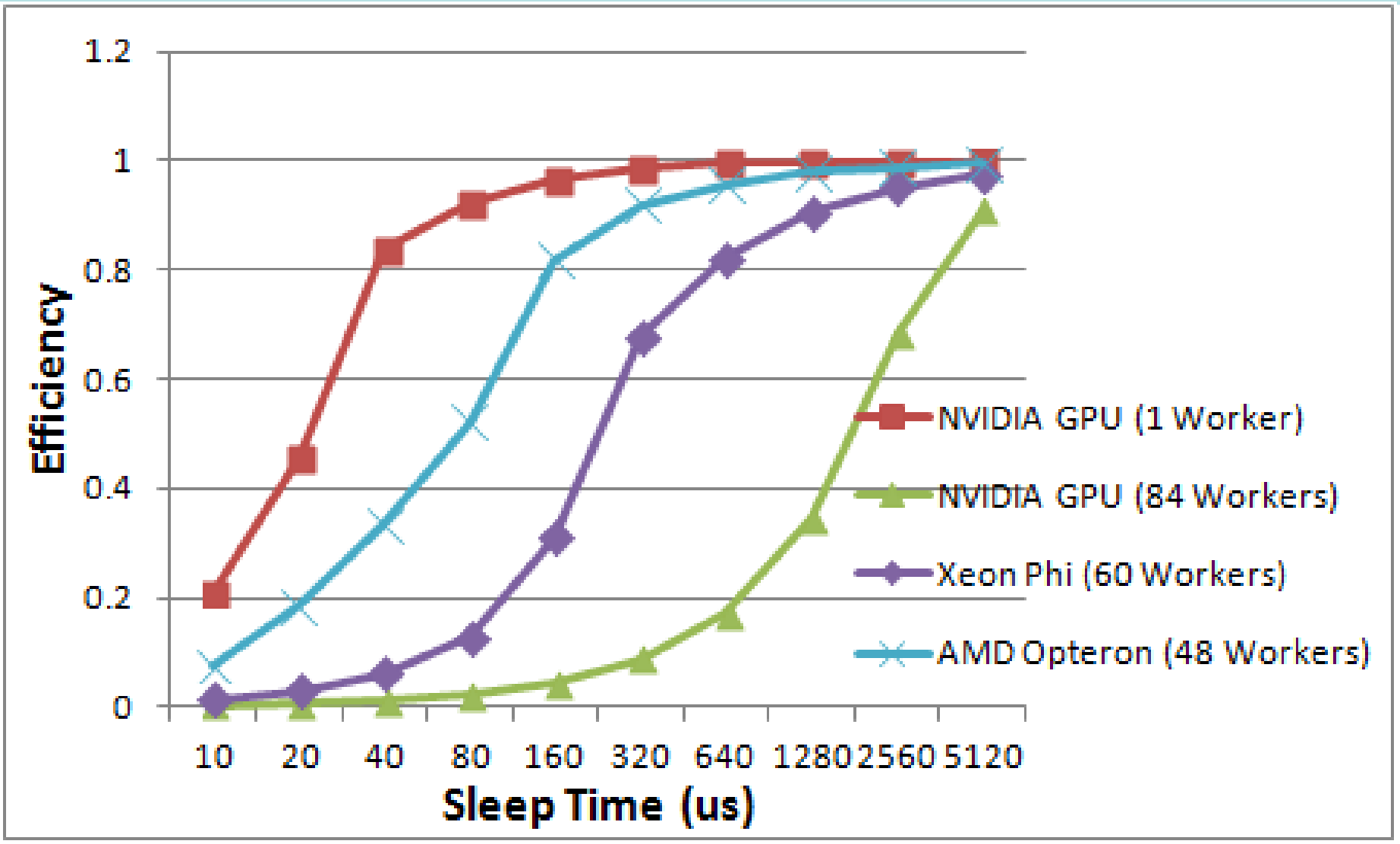
- Efficient support for MTC on accelrators



GeMTC Project



GeMTC Project



Active Collaborations
National Labs and Industry

- **National Laboratories**
  - **ANL**: Kamil Iskra, Rob Ross, Mike Wilde, Marc Snir, Pete Beckman, Justin Wozniak
  - **FNAL:** Gabriele Garzoglio
  - **LANL**: Mike Lang
  - **ORNL**: Arthur Barney Maccabe
  - **LBL**: Lavanya Ramakrishnan
- **Industry**
  - **Cleversafe:** Chris Gladwin
  - **EMC:** John Bent
  - **Accenture Technology Laboratory:** Teresa Tung
  - **Microsoft:** Roger Barga
  - **SchedMD:** Morris Jette, Danny Auble
  - **Oracle:** Hui Jin
  - **INRIA:** Gabriel Antoniu
  - **IBM:** Bogdan Nicolae

Active Collaborations Academia

- **Academia**
  - **IIT:** Xian-He Sun, Zhiling Lan, Shlomo Argamon
  - **UChicago:** Ian Foster, Tanu Malik, Zhao Zhang, Kyle Chard
  - **UEST China:** Yong Zhao
  - **SUNY:** Tevfik Kosar
  - **WSU:** Shiyong Lu
  - **USC:** Yogesh Simmhan
  - **Georgia Tech:** Jeffrey Vetter
  - **Columbia:** Glen Hocky

Active Funding ($)

- **NSF CAREER 2011 – 2015: $486K**
  - "*Avoiding Achilles' Heel in Exascale Computing with Distributed File Systems*", NSF CAREER
- **DOE Fermi 2011 – 2013: $84K**
  - "Networking and Distributed Systems in High-Energy Physics", DOE FNAL
- **DOE LANL 2013: $75K**
  - "Investigation of Distributed Systems for HPC System Services", DOE LANL
- **IIT STARR 2013: $15K**
  - **"***Towards the Support for Many-Task Computing on Many-Core Computing Platforms***",** IIT STARR Fellowship
- **Amazon 2011 - 2013: $18K**
  - **"***Distributed Systems Research on the Amazon Cloud Infrastructure***",** Amazon
- **NVIDIA 2013 – 2014: $12K**
  - "CUDA Teaching Center", NVIDIA

Funding (Time)

- **DOE 2011 – 2013: 450K hours**
  - "*FusionFS: Distributed File Systems for Exascale Computing*", DOE ANL ALCF; 450,000 hours on the IBM BlueGene/P
- **XSEDE 2013: 200K hours**
  - "*Many-Task Computing with Many-Core Accelerators on XSEDE*", NSF XSEDE; 200K hours on XSEDE
- **GLCPC 2013: 6M hours**
  - "*Implicitly-parallel functional dataflow for productive hybrid programming on Blue Waters*", Great Lakes Consortium for Petascale Computation (GLCPC); 6M hours on the Blue Waters Supercomputer
- **NICS 2013: 320K hours**
  - "Many-Task Computing with Many-Core Accelerators on Beacon", National Institute for Computational Sciences (NICS); 320K hours on the Beacon system

Service Activities

- IEEE Transactions on Cloud Computing
  - Special Issue on Scientific Cloud Computing
- Springer's Journal of Cloud Computing: Advances, Systems and Applications
- IEEE/ACM MTAGS 2013 @ SC13
- IEEE/ACM DataCloud 2013 @ SC13
- ACM ScienceCloud 2014 @ HPDC14
- IEEE CCGrid 2014 in Chicago
- GCASR 2014 in Chicago
- Others:
  - IEEE/ACM SC 2013, ACM HPDC 2014, IEEE IPDPS 2014, IEEE ICDCS 2014, IEEE eScience 2014

More Information

- More information:
  - http://www.cs.iit.edu/~iraicu/
  - http://datasys.cs.iit.edu/
- Contact:
  - iraicu@cs.iit.edu
- Questions?