Many-Task Computing on Grids, Clouds, and Supercomputers

Ioan Raicu
Center for Ultra-scale Computing and Information Security
Department of Electrical Engineering & Computer Science
Northwestern University

Computer Science, Illinois Institute of Technology
January 22nd, 2010

Acknowledgements

- Funding and Support (2003 – 2010)
  - CRA/NSF CIFellows Program
  - University of Chicago
    - Computer Science
    - Computational Institute
  - Argonne National Laboratory
    - Math and Computer Science Division
    - Argonne Leadership Computing Facility
  - NASA: Ames Research Center, GSRP Program

Ian Foster

Alok Choudhary

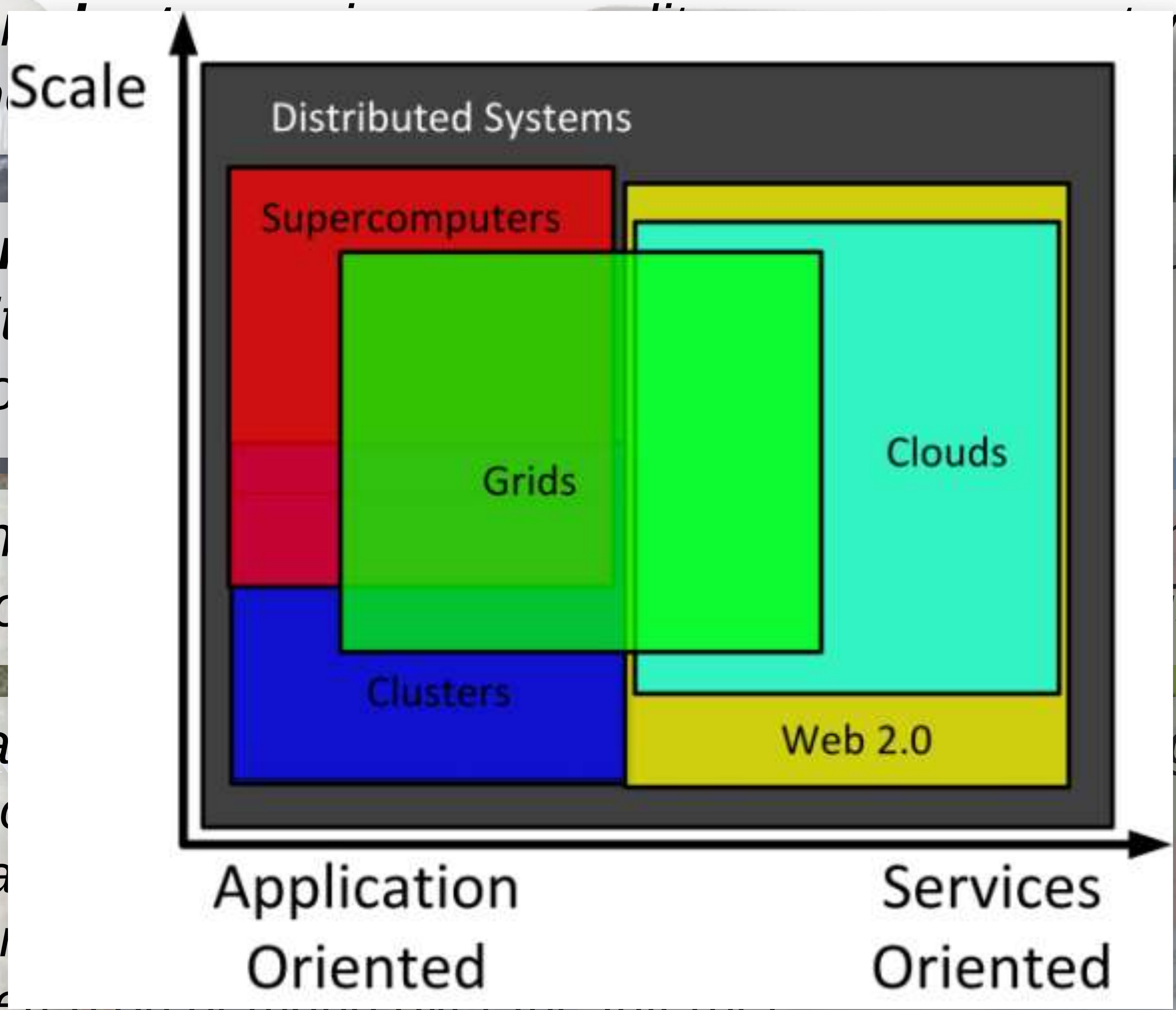
- Over 60 Collaborators
  - Ian Foster (UC/ANL), Rick Stevens (UC/ANL), Alex Szalay (JHU), Jim Gray (MSR), Pete Beckman (ANL), Jerry Yan (NASA ARC), Mike Wilde (UC/ANL), Douglas Thain (ND), Amitabh Chaudhary (ND), Yong Zhao (MS), Zhao Zhang (UC), Catalin Dumitrescu (FNAL), Matei Ripeanu (UBC), Alok Choudhary (NU), and many more…

Outline

- **Defining Many-Task Computing**
- **Motivation: Scalability Challenges**
- **Novel Resource Management Techniques**
- **Performance Evaluation: Micro-benchmarks**
- **Performance Evaluation: Applications**
- **Contributions**
- **Future Work**

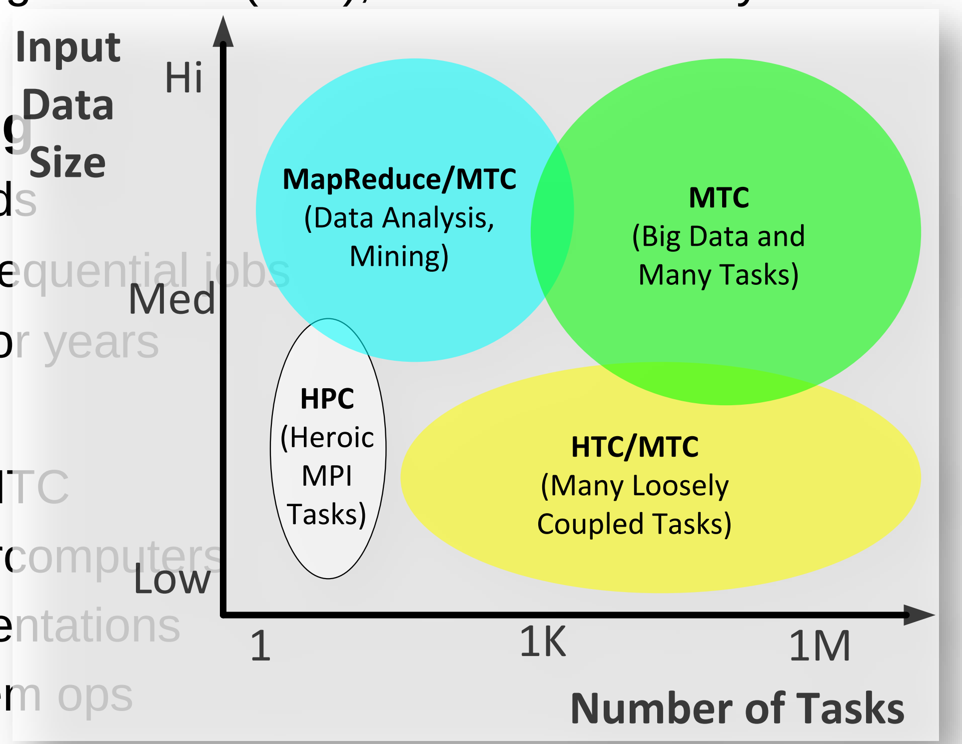
Clusters, Grids, Clouds, and Supercomputers

- *Compute... ...ork interconn...*
- ***Supercom**... ...using commodit... ...interconnects and custo...*
- ***Grids** ten... ...re typically loosely co... ...ispersed*
- ***Clouds** a... ...gm driven by:*
  1. *econo...*
  2. *virtua...*
  3. *dyna...*
  4. *delivered on demand over the Internet*



HPC, HTC, MTC

- **HPC: High-Performance Computing**
  - Synonymous with supercomputing
  - Tightly-coupled applications
  - Implemented using Message Passing Interface (MPI), needs low latency networks
  - Measured in FLOPS
- **HTC: High-Throughput Computing**
  - Typically applied in clusters and grids
  - Loosely-coupled applications with sequential jobs
  - Measured in operations per month or years
- **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

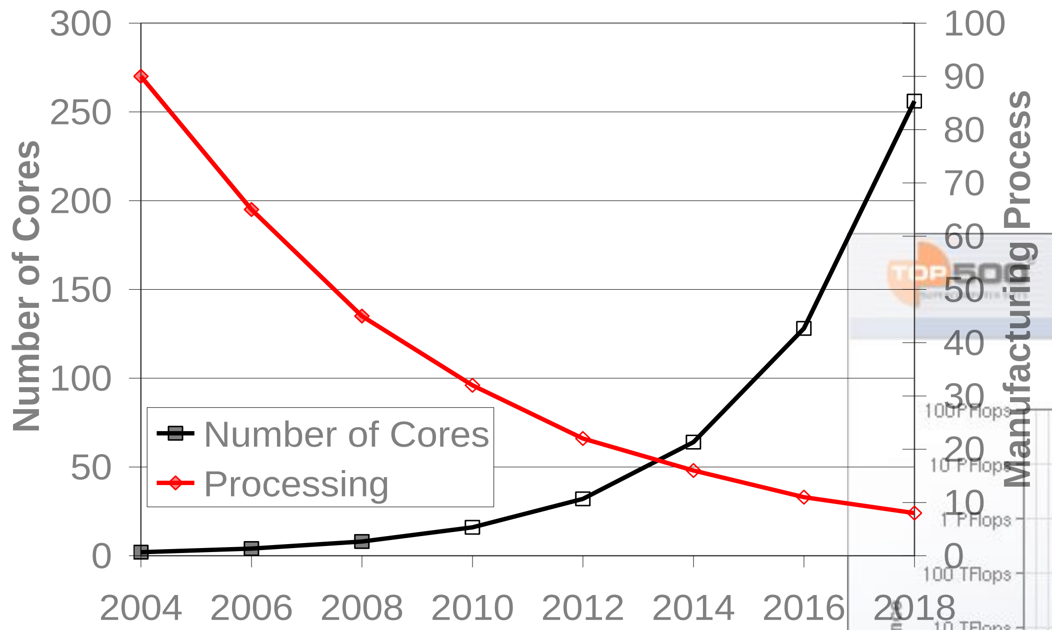


[MTAGS08] “Many-Task Computing for Grids and Supercomputers”

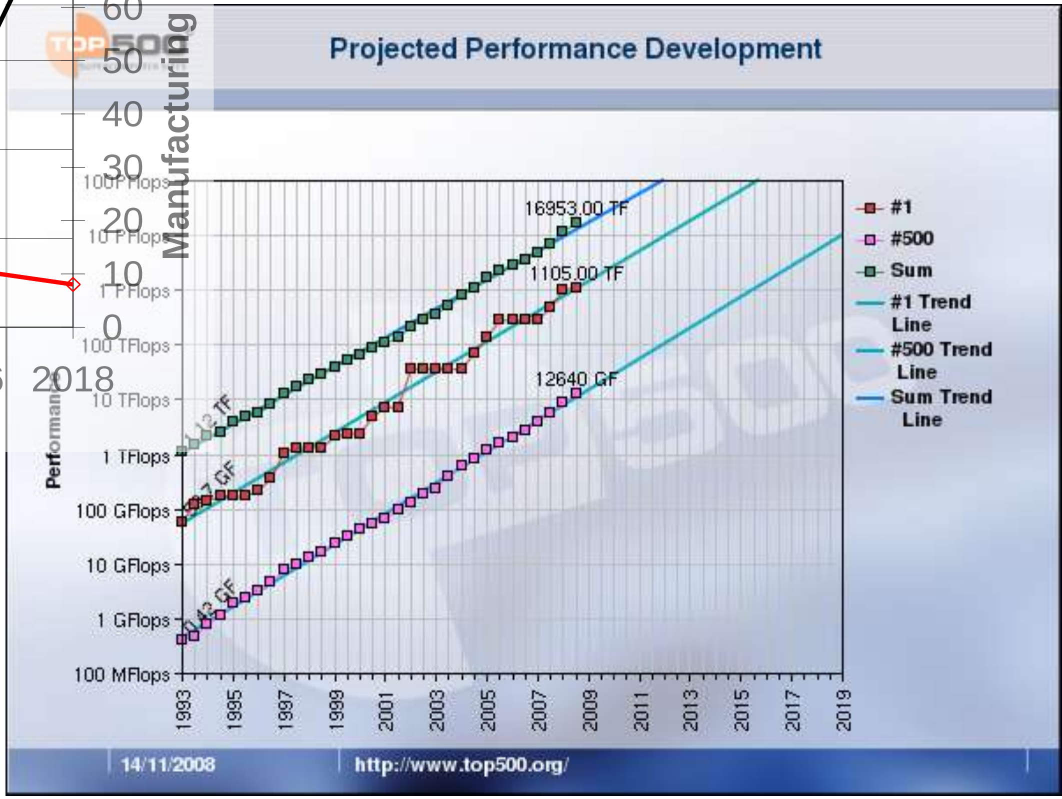
Outline

- Defining Many-Task Computing
- **Motivation: Scalability Challenges**
- Novel Resource Management Techniques
- Performance Evaluation: Micro-benchmarks
- Performance Evaluation: Applications
- Contributions
- Future Work

Projected Growth Trends



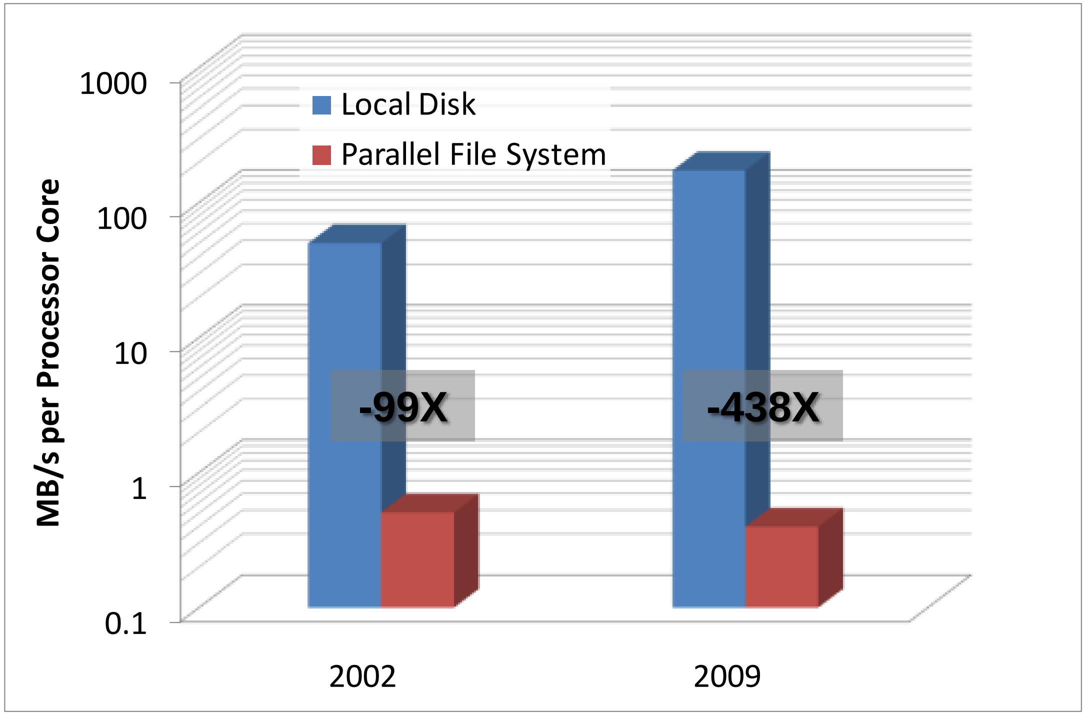
Pat Helland, Microsoft, The Irresistible Forces Meet the Movable Objects, November 9th, 2007



Top500 Projected Development, http://www.top500.org/lists/2008/11/performance_development

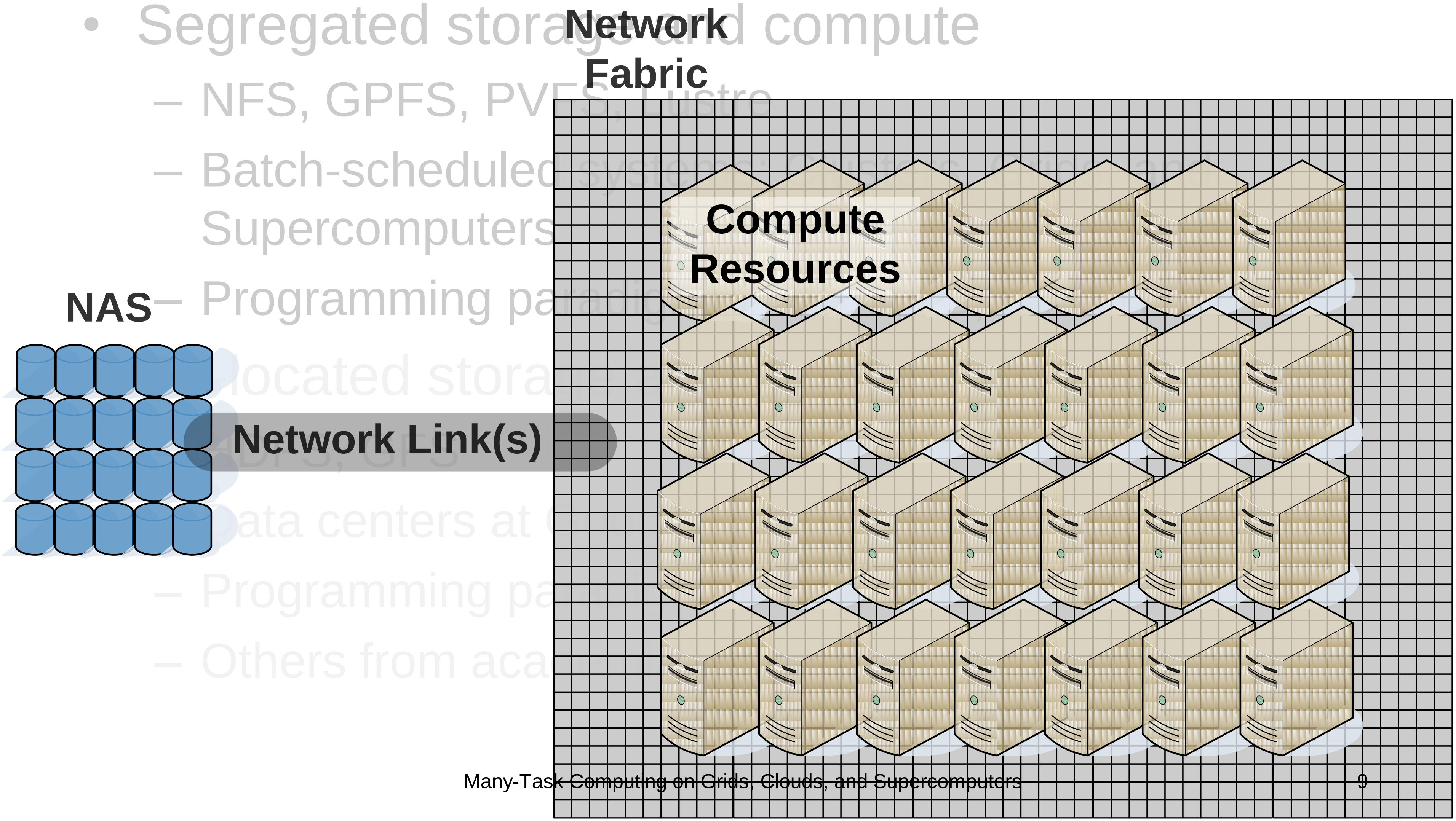
Growing Storage/Compute Gap

- 2002:
  - Local disk
    - ANL/UC TG Site (70GB SCSI)
  - Parallel File System
    - 2002: IBM Blue Gene/L (GPFS, 1GB/s)
- 2009:
  - Local disk
    - PADS (RAID-0, 6 drives 750GB SATA)
  - Parallel File System
    - IBM Blue Gene/P (GPFS, 65GB/s)



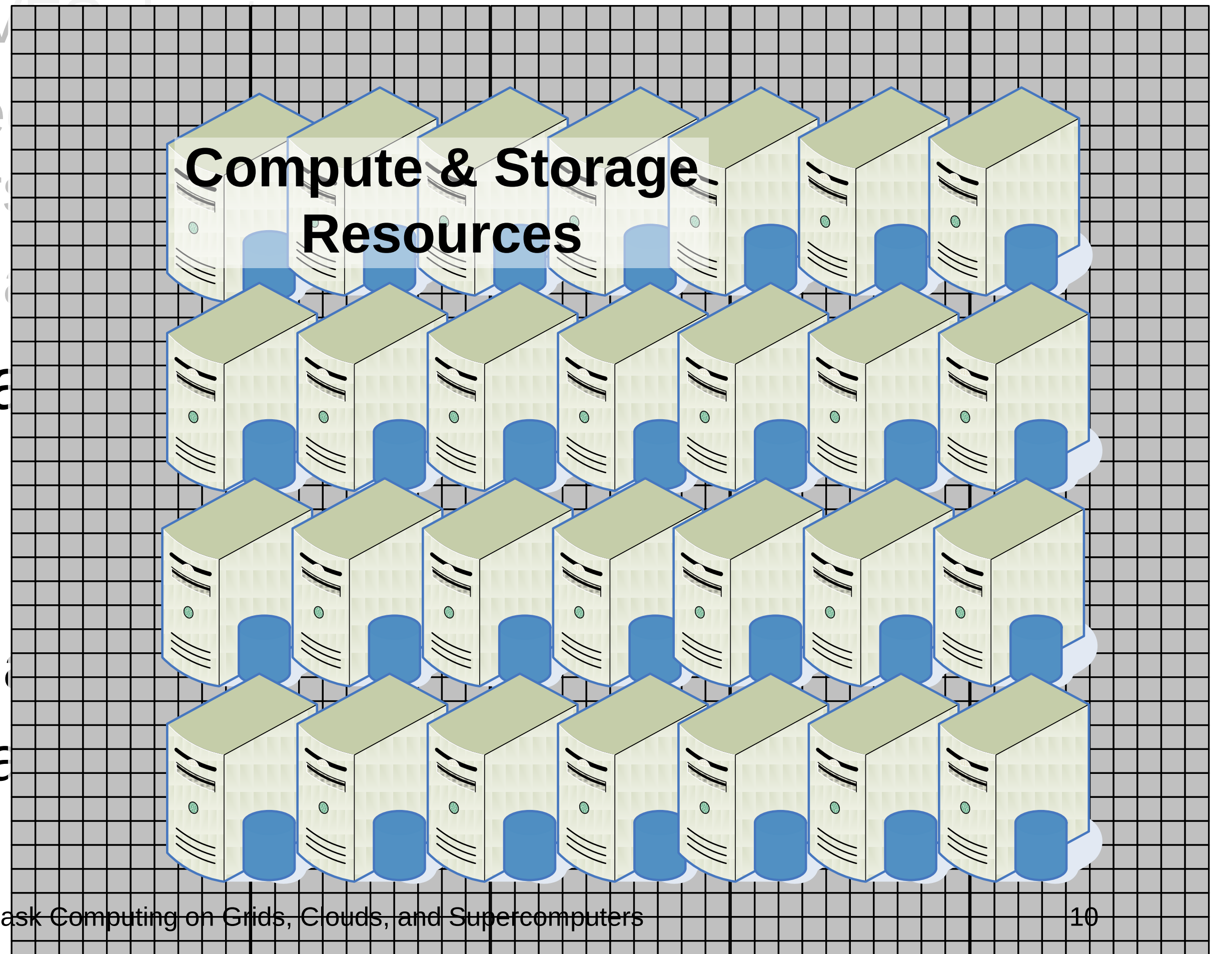
State of the Art: Storage Systems

- Segregated storage and compute
  - NFS, GPFS, PVFS, Lustre
  - Batch-scheduled
    Supercomputers
  - Programming pa
- located stora
  - DFS, GFS
  - ata centers at
  - Programming pa
  - Others from aca

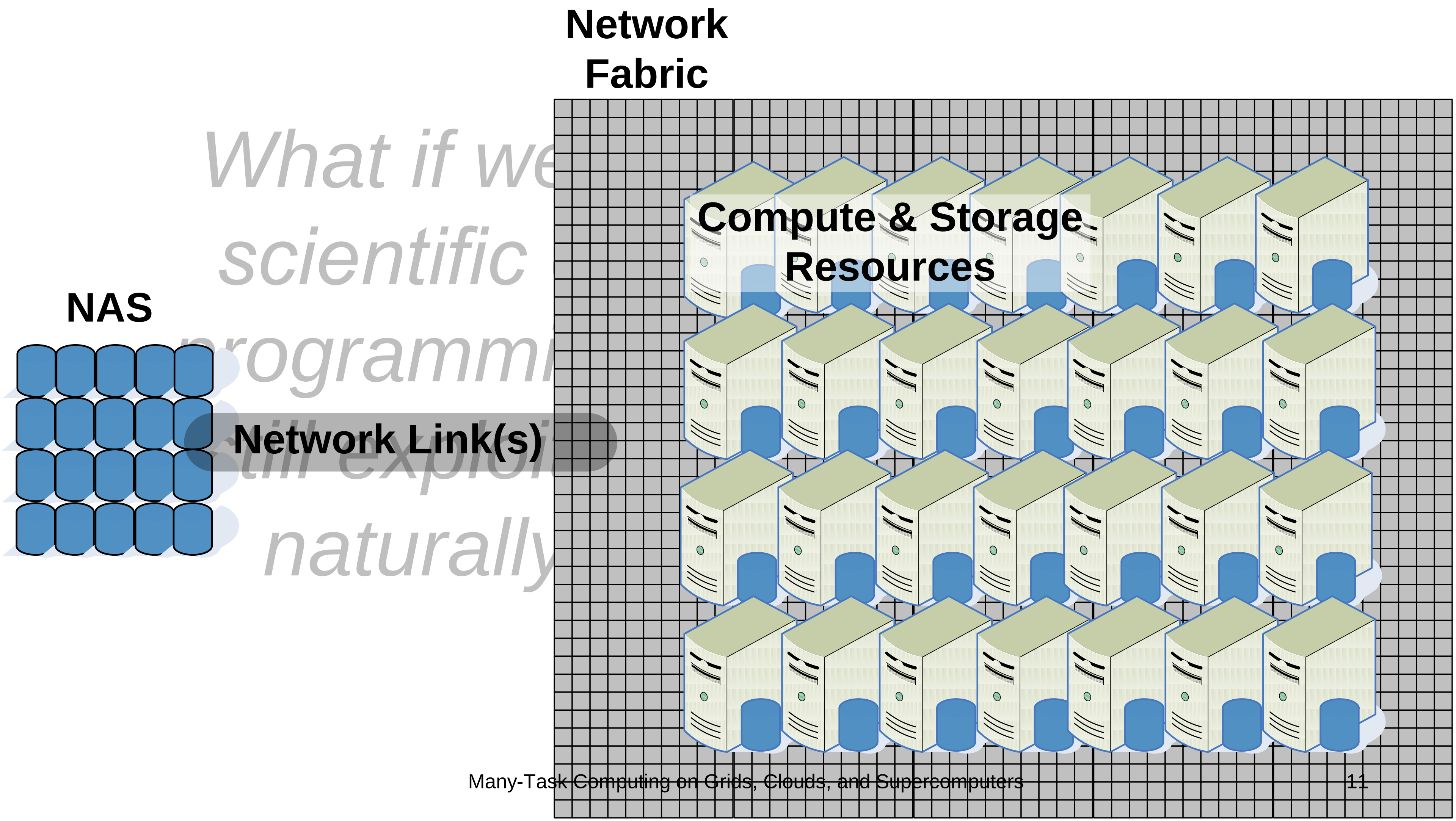


State of the Art: Storage Systems

- Segregated storage and compute
  - NFS, GPFS, PV
  - Batch-schedule
    Supercomputer
  - Programming p
- Co-located stora
  - HDFS, GFS
  - Data centers at
  - Programming p
  - Others from aca



Combine State of the Art Systems



Outline

- Defining Many-Task Computing
- Motivation: Scalability Challenges
- **Novel Resource Management Techniques**
- Performance Evaluation: Micro-benchmarks
- Performance Evaluation: Applications
- Contributions
- Future Work

Techniques to Support MTC

- Streamlined task dispatching
- Dynamic resource provisioning
  - Multi-level scheduling
  - Resources are acquired/released in response to demand
- Data diffusion
  - Data diffuses from archival storage to transient resources
  - Resource "caching" allows faster responses to subsequent requests
  - Co-locate data and computations to optimize performance

[HPDC09] "The Quest for Scalable Support of Data Intensive Workloads in Distributed Systems"
[DIDC09] "Towards Data Intensive Many-Task Computing"
[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"
[DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"
[UC07] "Harnessing Grid Resources with Data-Centric Task Farms"
[SC07] "Falkon: a Fast and Light-weight tasK executiON framework"
[TG07] "Dynamic Resource Provisioning in Grid Environments"

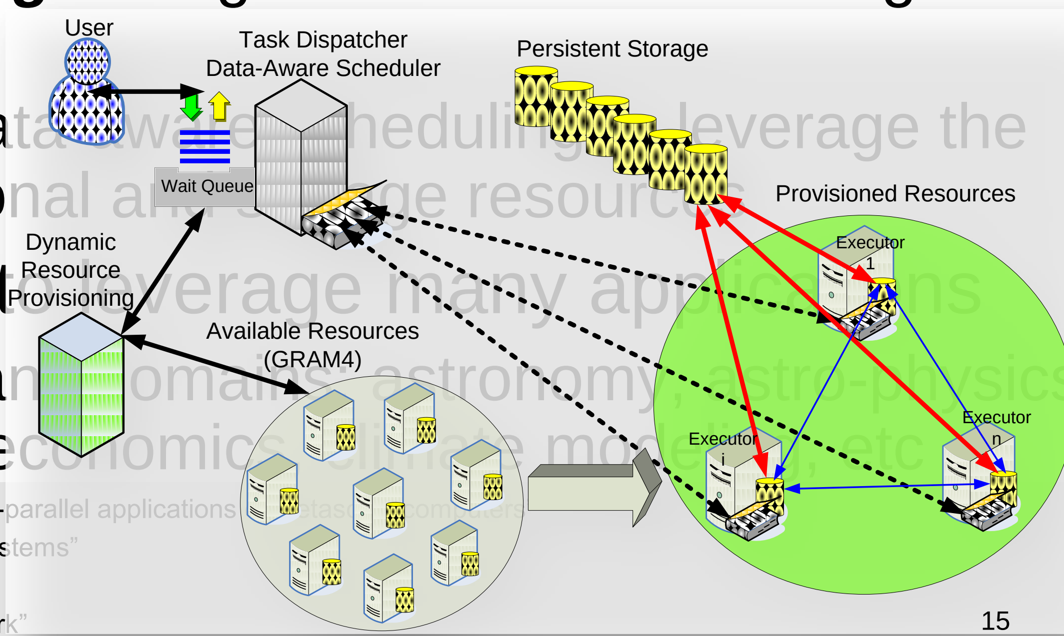
Theoretical and Practical Exploration

- Abstract model
  - Models the efficiency and speedup of entire workloads
  - Captures techniques to support MTC
    - Streamlined task dispatching, dynamic resource provisioning, data diffusion
  - Lead to proof of O(NM) competitive caching
- Middleware to support MTC
  - Falkon: a fast a light-weight execution framework
    - Reference Implementation of the abstract model
  - Swift: A Parallel Programming System

[TPDS10] "Middleware Support for Many-Task Computing", under preparation
[DIDC09] "Towards Data Intensive Many-Task Computing"
[SC07] "Falkon: a Fast and Light-weight tasK executiON framework"

Middleware Support: Falkon

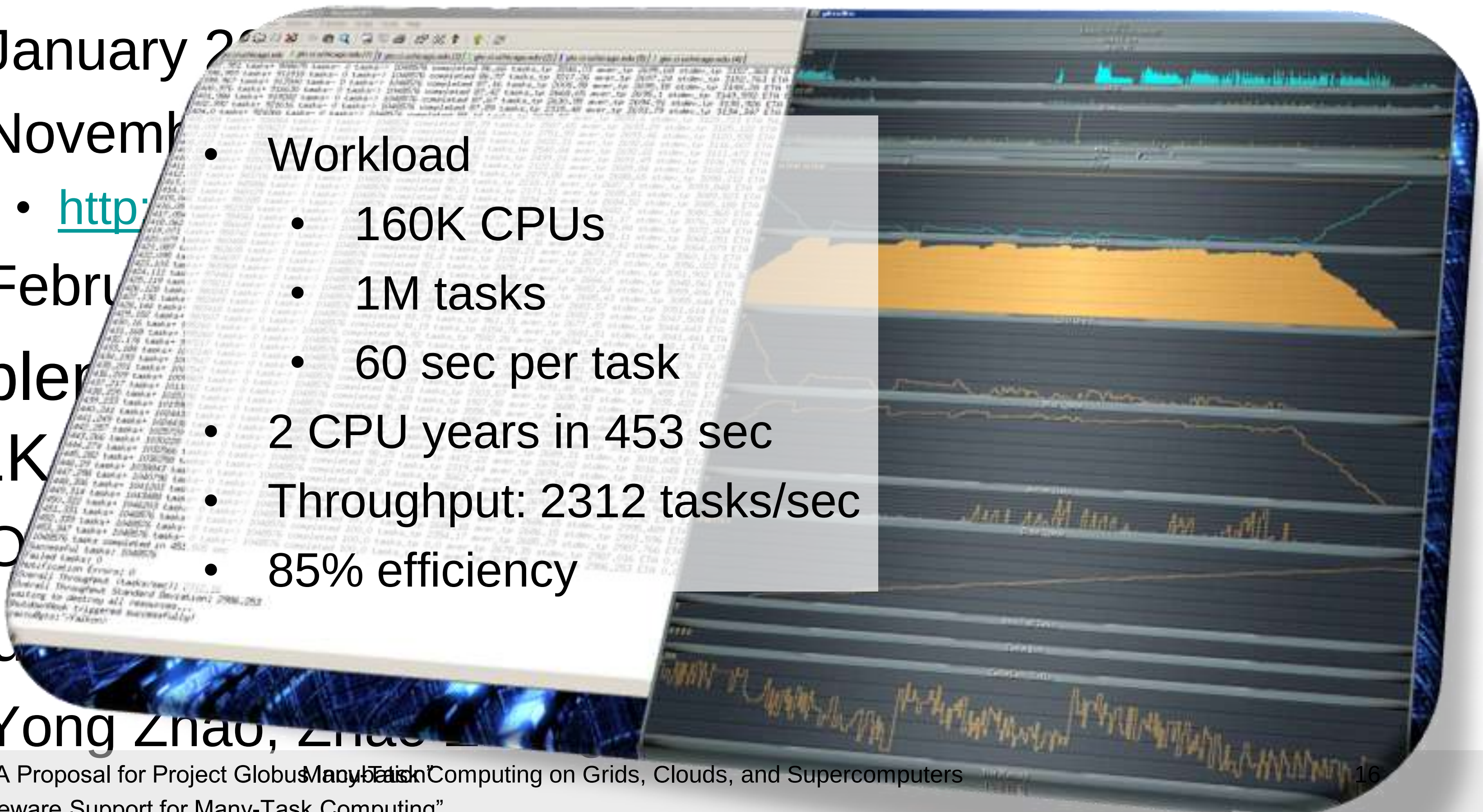
- ***Goal:*** enable the ***rapid and efficient*** execution of many independent jobs on large compute clusters
- Combines three components:
  - a ***streamlined task dispatcher***
  - ***resource provisioning*** through multi-level scheduling techniques
  - ***data diffusion*** and data-aware scheduling to leverage the co-located computational and storage resources
- Integration into Swift to leverage many applications
  - Applications cover many domains: astronomy, astro-physics, medicine, chemistry, economics, climate modeling, etc



[SciDAC09] "Extreme-scale scripting: Opportunities for large task-parallel applications on petascale computers"
[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"
[Globus07] "Falkon: A Proposal for Project Globus Incubation"
[SC07] "Falkon: a Fast and Light-weight tasK executiON framework"
[SWF07] "Swift: Fast, Reliable, Loosely Coupled Parallel Computation"

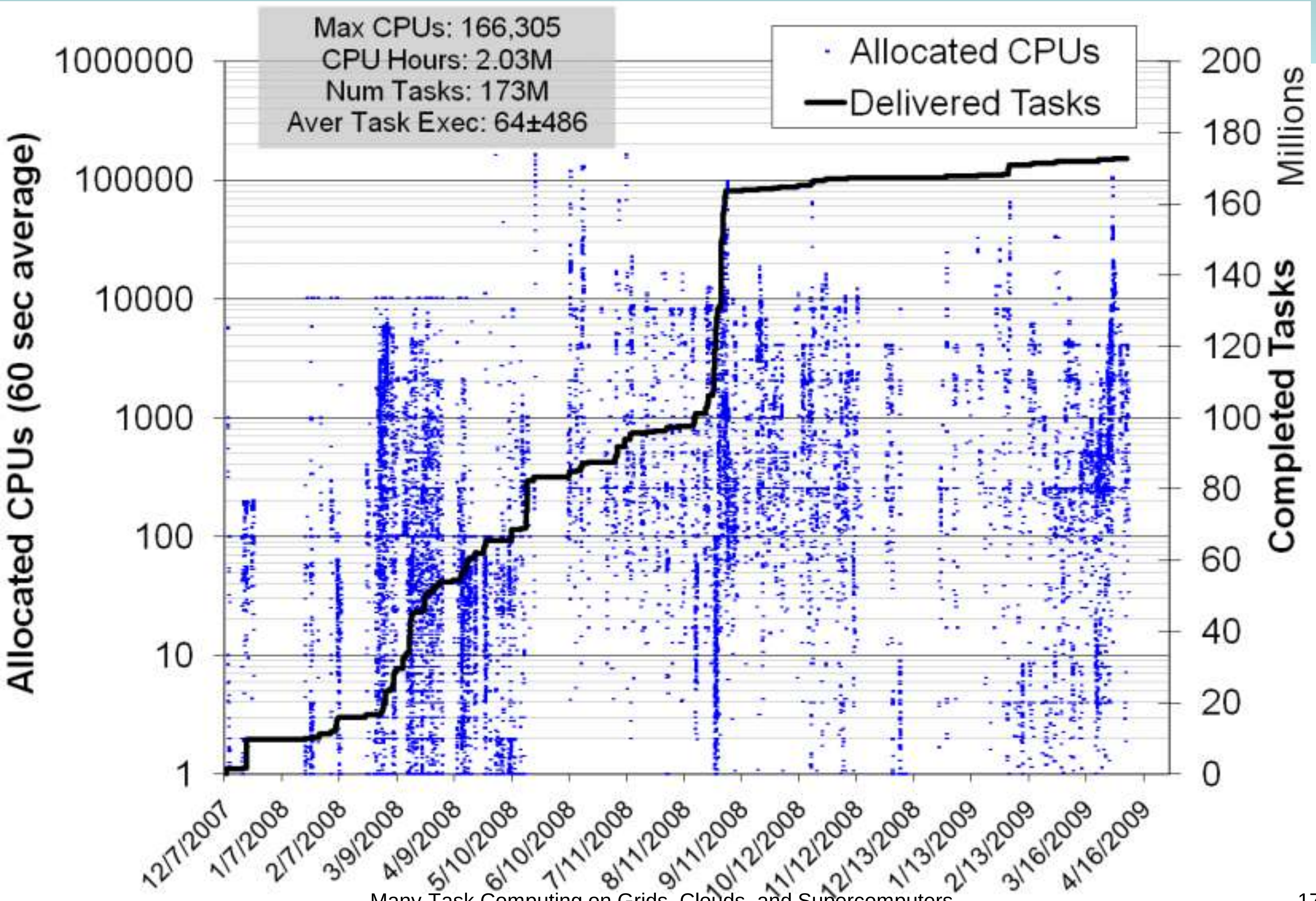
Falkon Project

- Falkon is a real system
  - Late 2005: Initial prototype, AstroPortal
  - January 2
  - Novemb
    - http:
  - Febru
- Implem
(~1K
  - O
- Sou
  - Yong Zhao, Zhao



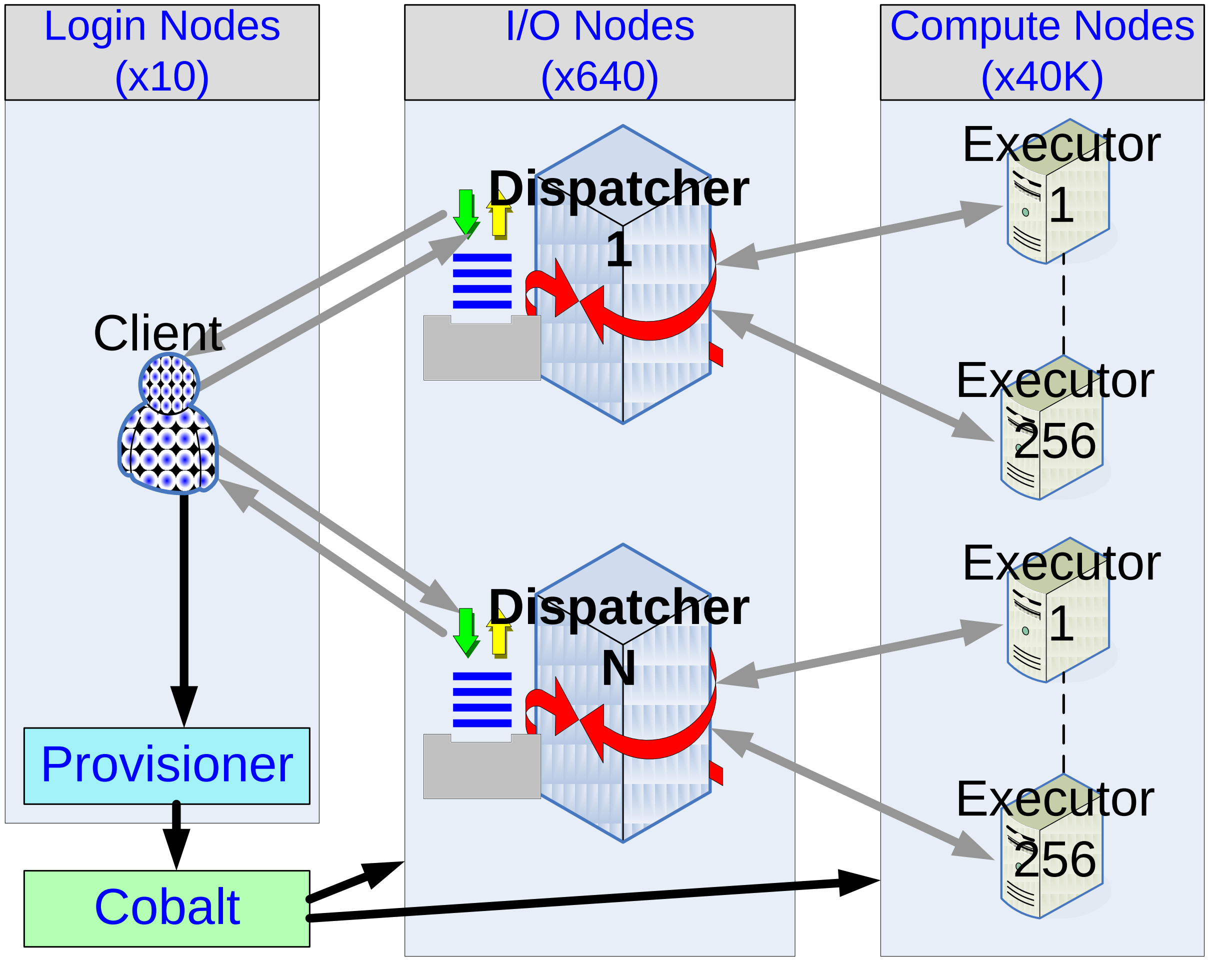
[Globus07] "Falkon: A Proposal for Project Globus Incubation"
[CLUSTER10] "Middleware Support for Many-Task Computing"

Falkon Activity History (16 months)



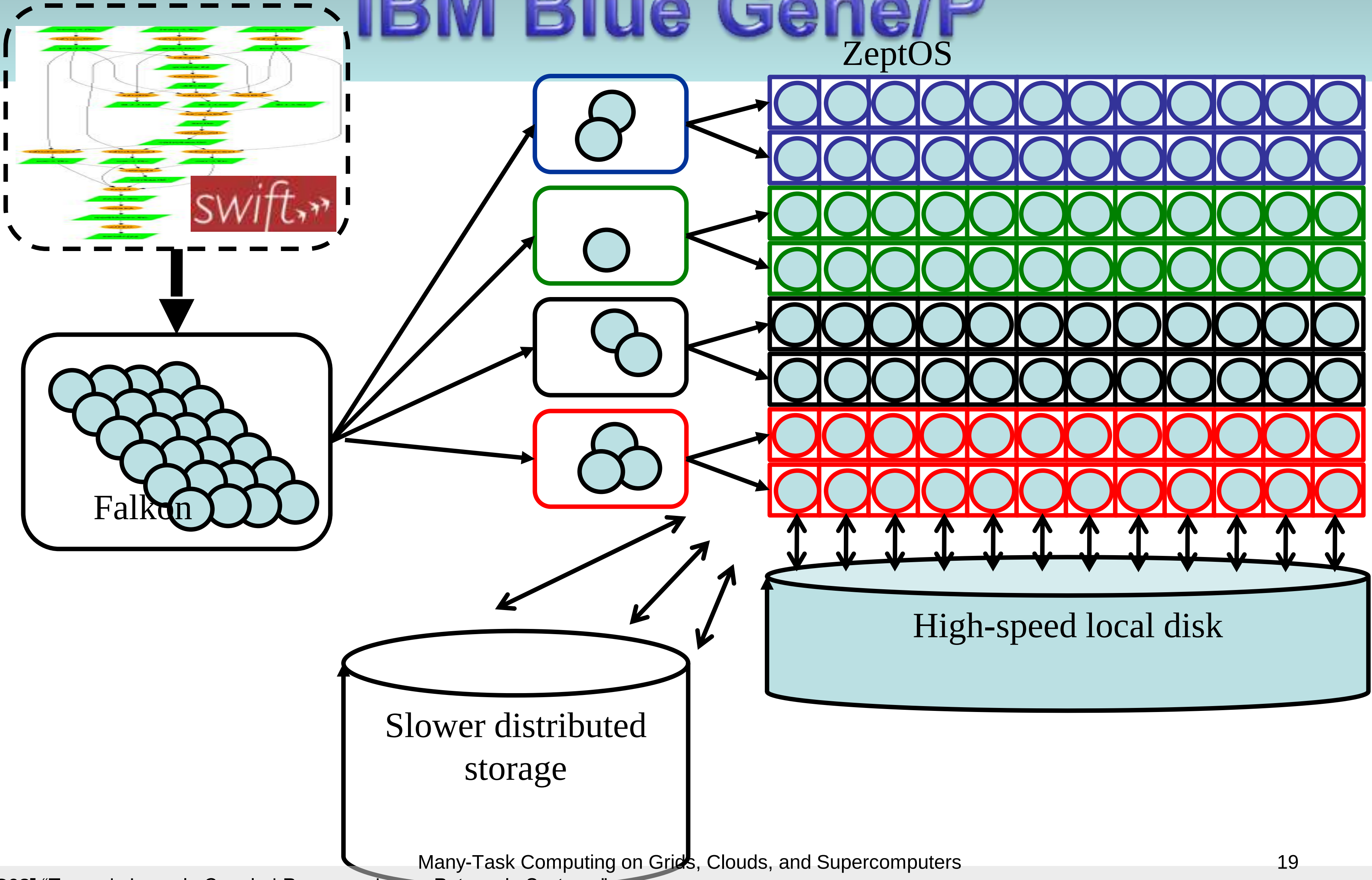
[TPDS09] "Middleware Support for Many-Task Computing", under preparation

Distributed Falkon Architecture



[SC08] “Towards Loosely-Coupled Programming on Petascale Systems”

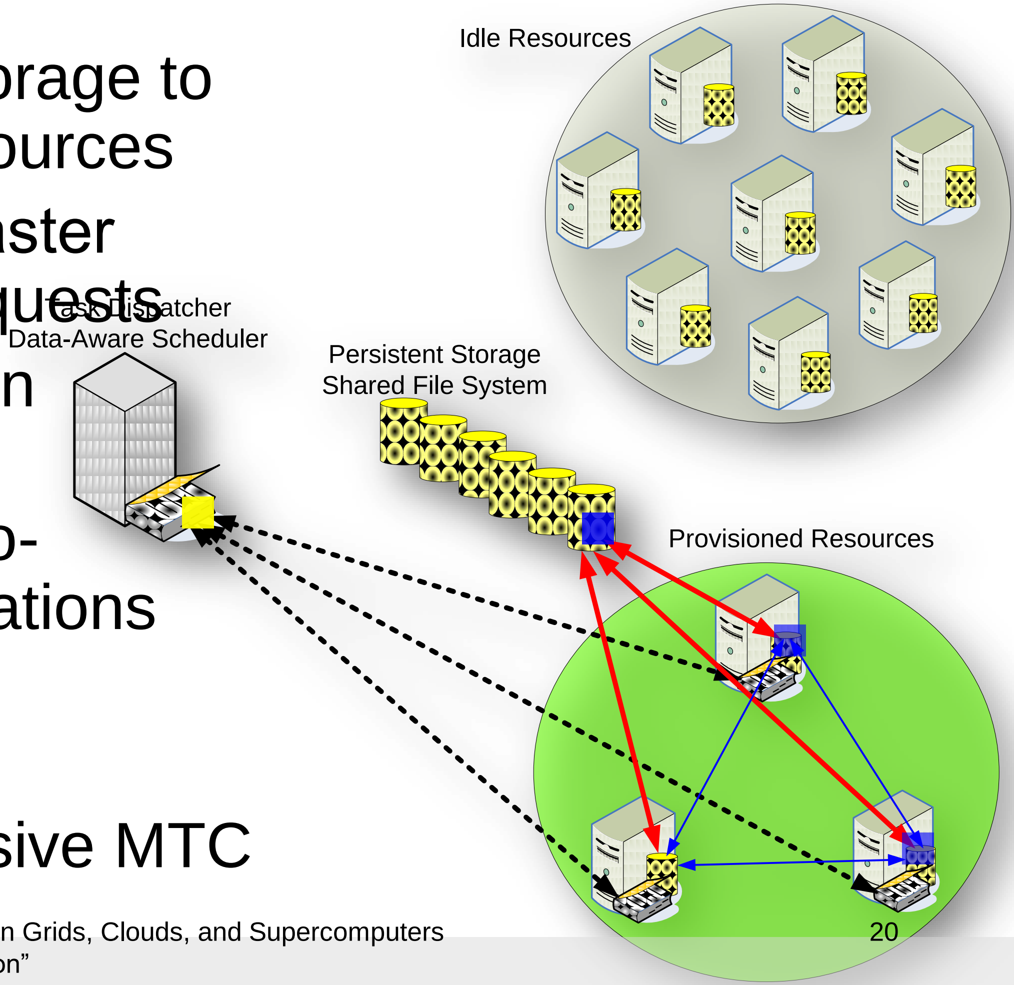
Managing 160K CPUs
IBM Blue Gene/P



[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

Data Diffusion

- Resource acquired in response to demand
- Data diffuse from archival storage to newly acquired transient resources
- Resource "caching" allows faster responses to subsequent requests
- Resources are released when demand drops
- Optimizes performance by co-scheduling data and computations
- Decrease dependency of a shared/parallel file systems
- Critical to support data intensive MTC



[DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"

Scheduling Policies

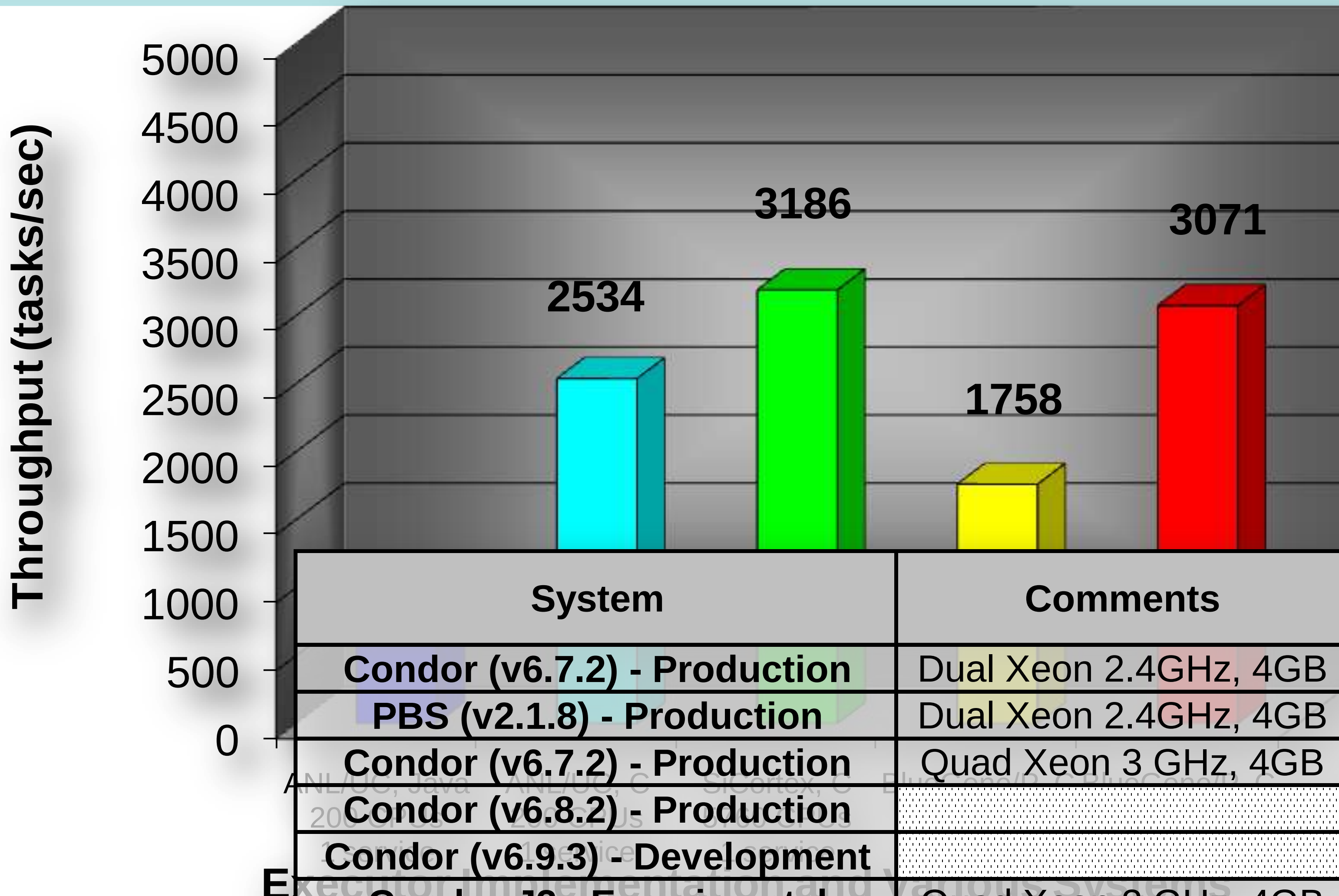
- FA: first-available
  - simple load balancing
- MCH: max-cache-hit
  - maximize cache hits
- MCU: max-compute-util
  - maximize processor utilization
- GCC: good-cache-compute
  - maximize both cache hit and processor utilization at the same time

[DADC08] “Accelerating Large-scale Data Exploration through Data Diffusion”

Outline

- Defining Many-Task Computing
- Motivation: Scalability Challenges
- Novel Resource Management Techniques
- **Performance Evaluation: Micro-benchmarks**
- Performance Evaluation: Applications
- Contributions
- Future Work

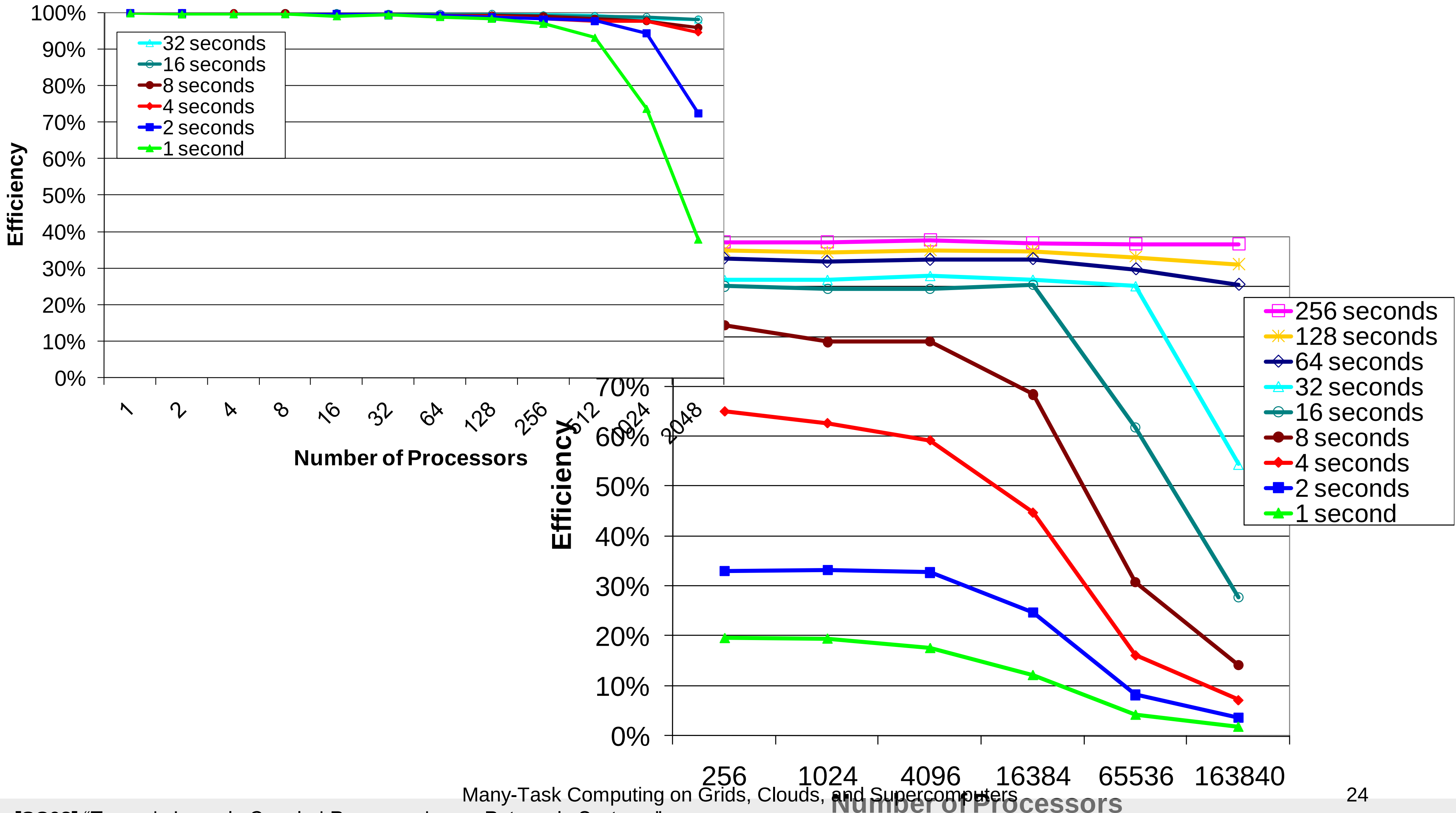
Dispatch Throughput



System	Comments	Throughput (tasks/sec)
Condor (v6.7.2) - Production	Dual Xeon 2.4GHz, 4GB	0.49
PBS (v2.1.8) - Production	Dual Xeon 2.4GHz, 4GB	0.45
Condor (v6.7.2) - Production	Quad Xeon 3 GHz, 4GB	2
Condor (v6.8.2) - Production		0.42
Condor (v6.9.3) - Development		11
Condor-J2 - Experimental	Quad Xeon 3 GHz, 4GB	22

[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

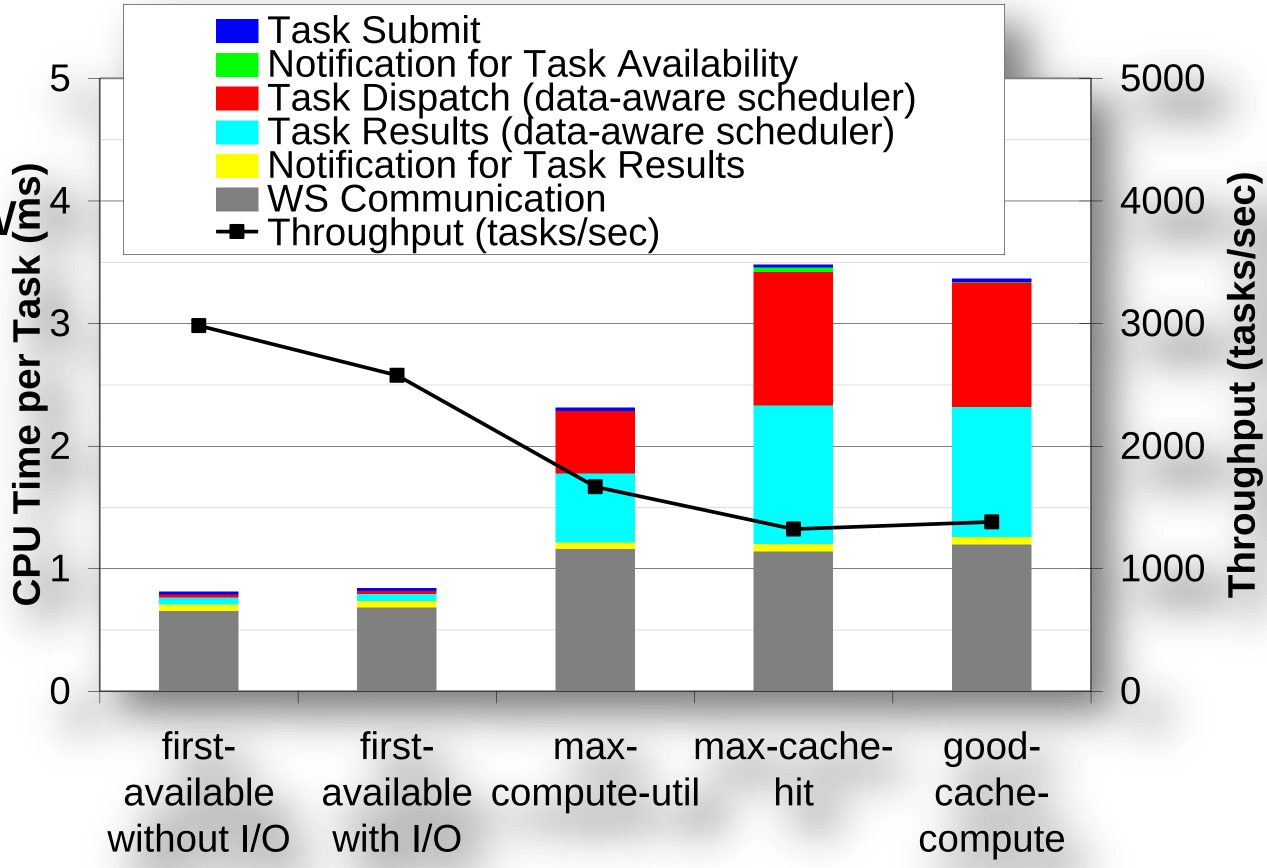
Execution Efficiency



[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

Data-Aware Scheduler Profiling

- 3GHz dual CPUs
- ANL/UC TG with 128 processors
- Scheduling window 2500 tasks
- Dataset
  - 100K files
  - 1 byte each
- Tasks
  - Read 1 file
  - Write 1 file



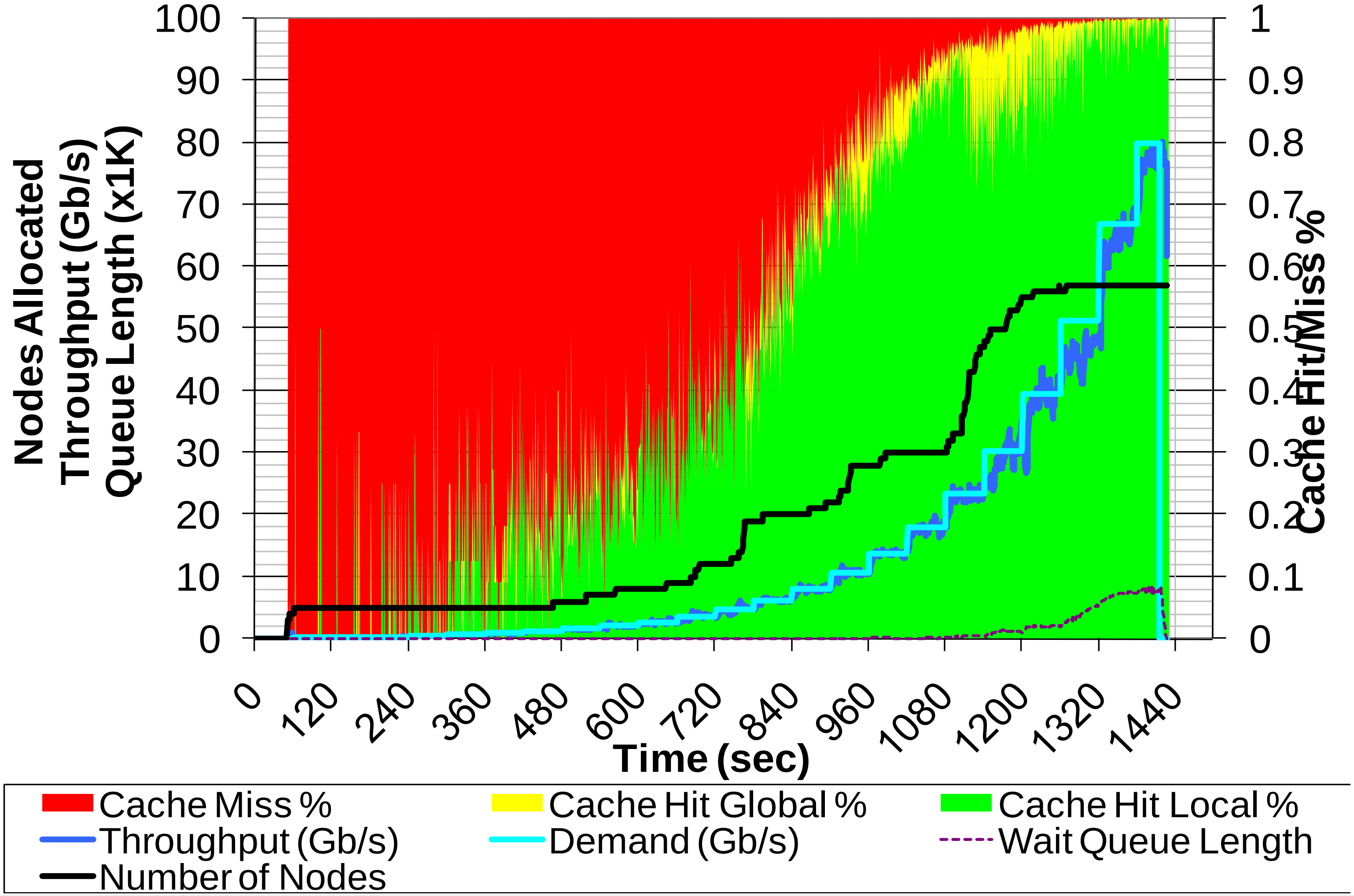
[DIDC09] “Towards Data Intensive Many-Task Computing”, under review

Synthetic Workloads

- Monotonically Increasing Workload
  - Emphasizes increasing loads
- Sine-Wave Workload
  - Emphasizes varying loads
- All-Pairs Workload
  - Compare to best case model of active storage
- Image Stacking Workload (Astronomy)
  - Evaluate data diffusion on a real large-scale data-intensive application from astronomy domain

[DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems"
[DIDC09] "Towards Data Intensive Many-Task Computing"

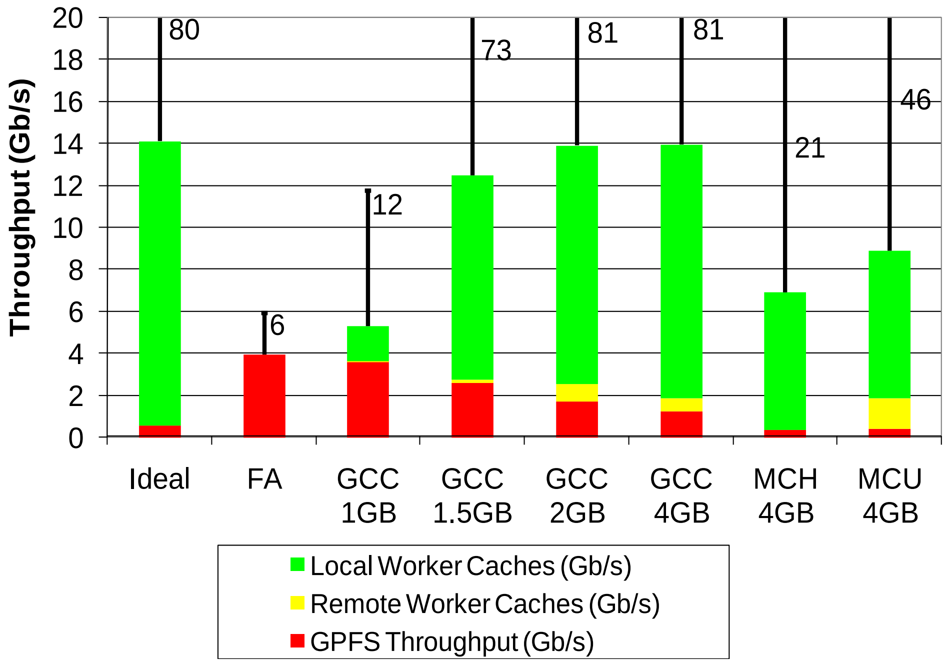
Data Diffusion
Monotonically Increasing Workload



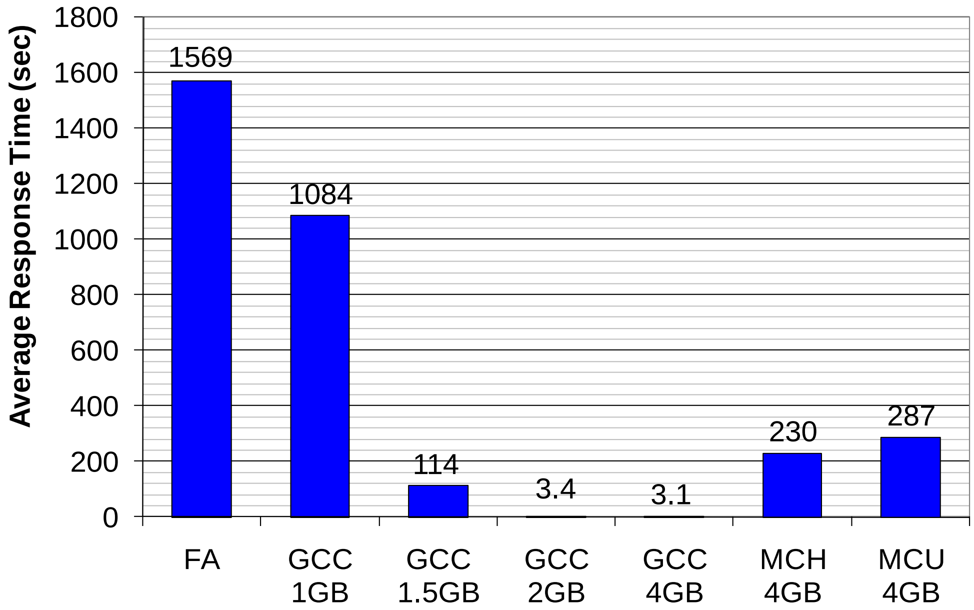
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems"

[DIDC09] "Towards Data Intensive Many-Task Computing"

Data Diffusion
Monotonically Increasing Workload



- ← Throughput:
  - Average: 14Gb/s vs 4Gb/s
  - Peak: 81Gb/s vs. 6Gb/s



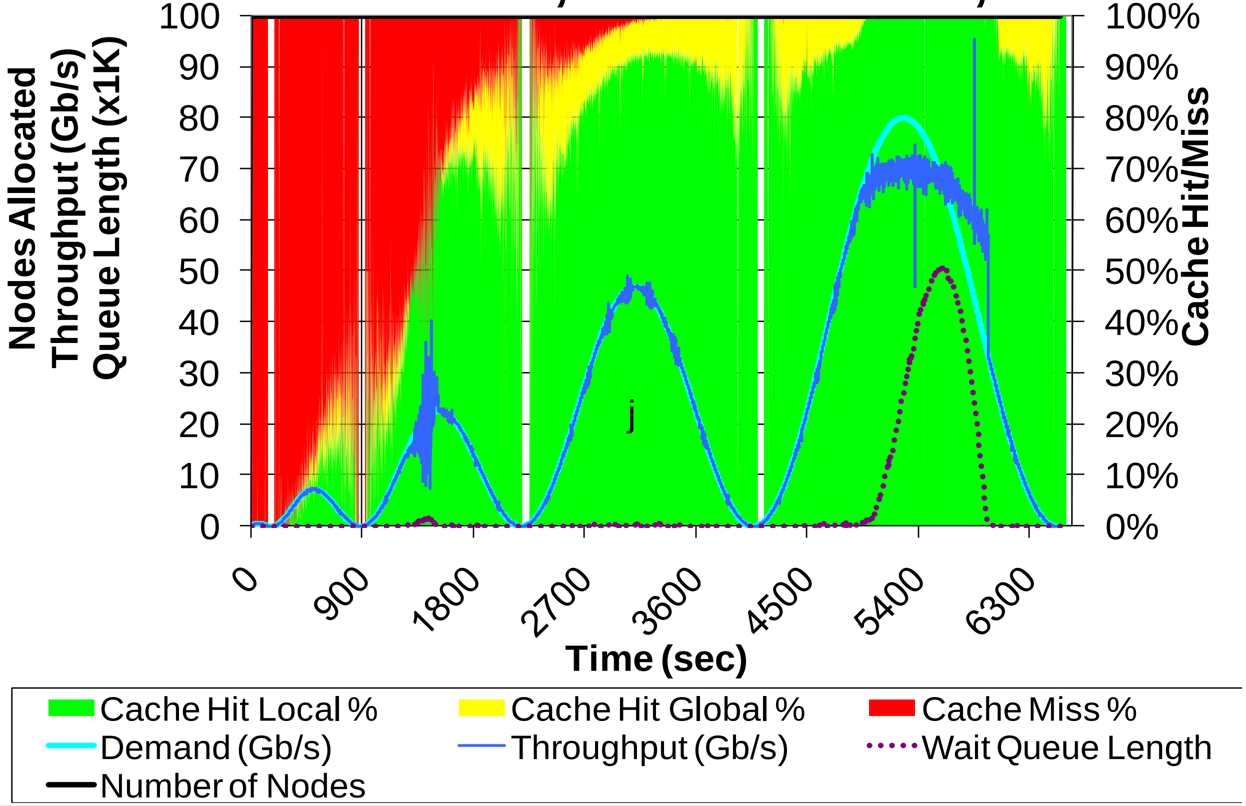
- Response Time →
  - 3 sec vs 1569 sec → 506X

[HPDC09] “The Quest for Scalable Support of Data Intensive Applications in Distributed Systems”

[DIDC09] “Towards Data Intensive Many-Task Computing”

Data Diffusion
Sine-Wave Workload

- GPFS ➔ 5.7 hrs, ~8Gb/s, 1138 CPU hrs
- GCC+SRP ➔ 1.8 hrs, ~25Gb/s, 361 CPU hrs

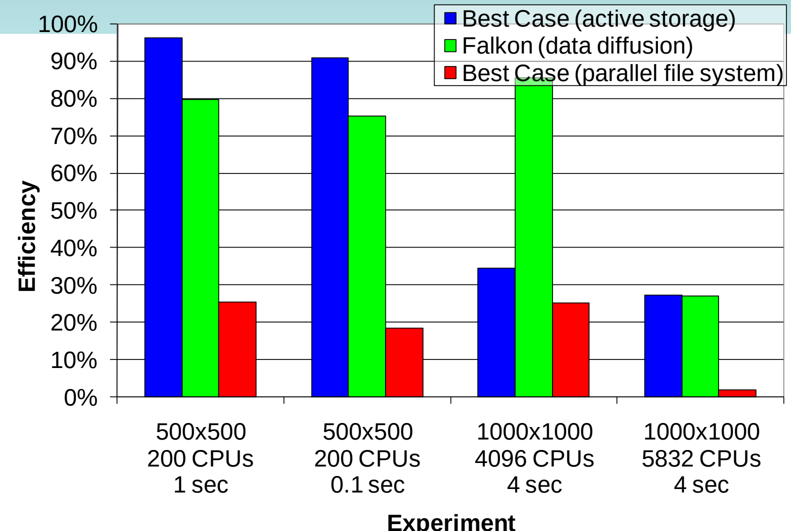


[HPDC09] “The Quest for Scalable Support of Data Intensive Applications in Distributed Systems”, under review
[DIDC09] “Towards Data Intensive Many-Task Computing”, under review

Data Diffusion vs. Active Storage All-Pairs Workload

- Pull vs. Push
  - Data Diffusion
    - Pulls ***task*** working set
    - Incremental spanning forest
  - Active Storage:
    - Pushes ***workload*** working set to all nodes
    - Static spanning tree

Christopher Moretti, Douglas Thain, University of Notre Dame



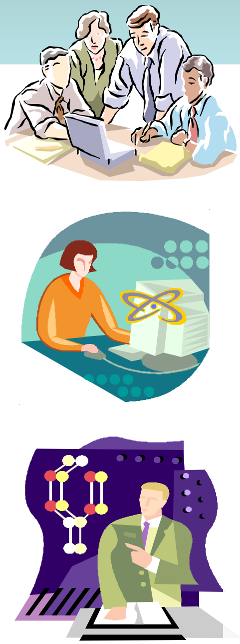
Experiment	Approach	Local Disk/Memory (GB)	Network (node-to-node) (GB)	Shared File System (GB)
500x500 200 CPUs 1 sec	Best Case (active storage)	6000	1536	12
	Falkon (data diffusion)	6000	1698	34
500x500 200 CPUs 0.1 sec	Best Case (active storage)	6000	1536	12
	Falkon (data diffusion)	6000	1528	62
1000x1000 4096 CPUs 4 sec	Best Case (active storage)	24000	12288	24
	Falkon (data diffusion)	24000	4676	384
1000x1000 5832 CPUs 4 sec	Best Case (active storage)	24000	12288	24
	Falkon (data diffusion)	24000	3867	906

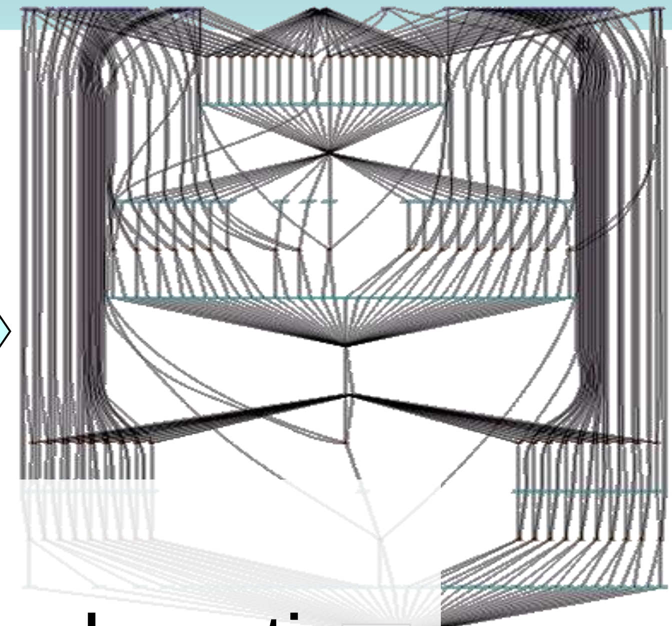
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

Outline

- Defining Many-Task Computing
- Motivation: Scalability Challenges
- Novel Resource Management Techniques
- Performance Evaluation: Micro-benchmarks
- **Performance Evaluation: Applications**
- Contributions
- Future Work

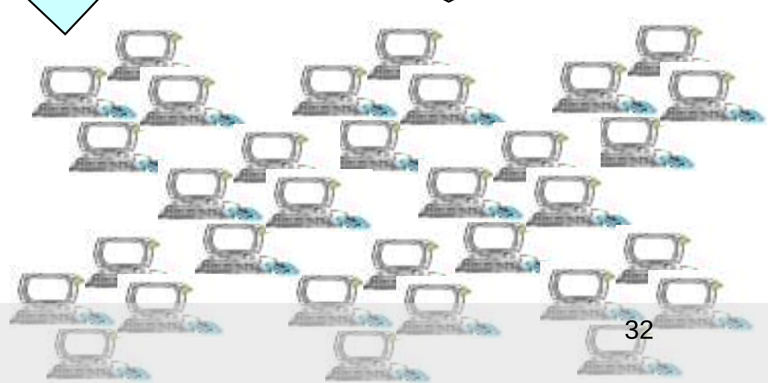
Applications
Medical Imaging: fMRI





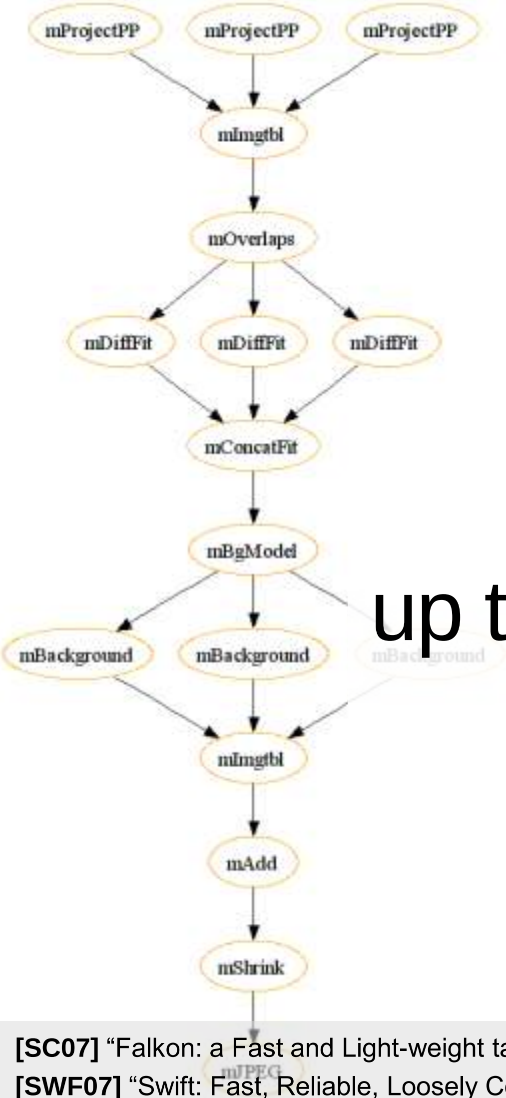
Improvement:
up to **90%** lower end-to-end run time

- Wide range of analyses
  - Testing, interactive analysis, production runs
  - Data mining
  - Parameter studies



[SC07] “Falkon: a Fast and Light-weight tasK executiON framework”
[SWF07] “Swift: Fast, Reliable, Loosely Coupled Parallel Computation”

Applications
Astronomy: Montage

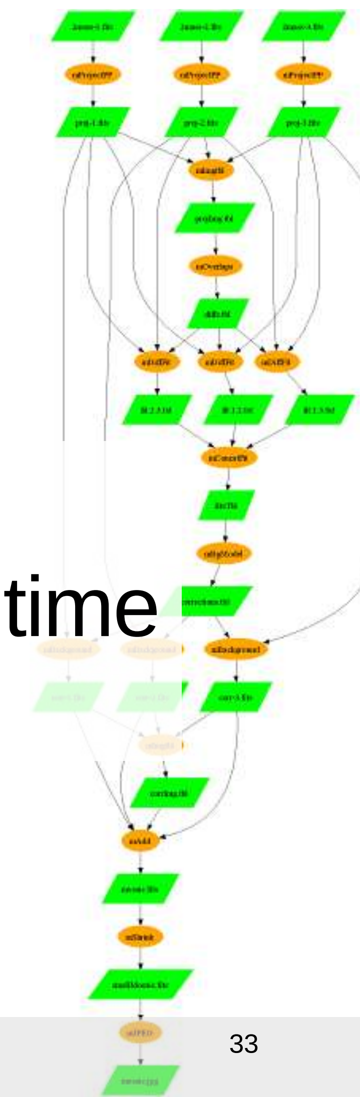


Improvement:

up to 57% lower end-to-end run time

Within 4% of MPI

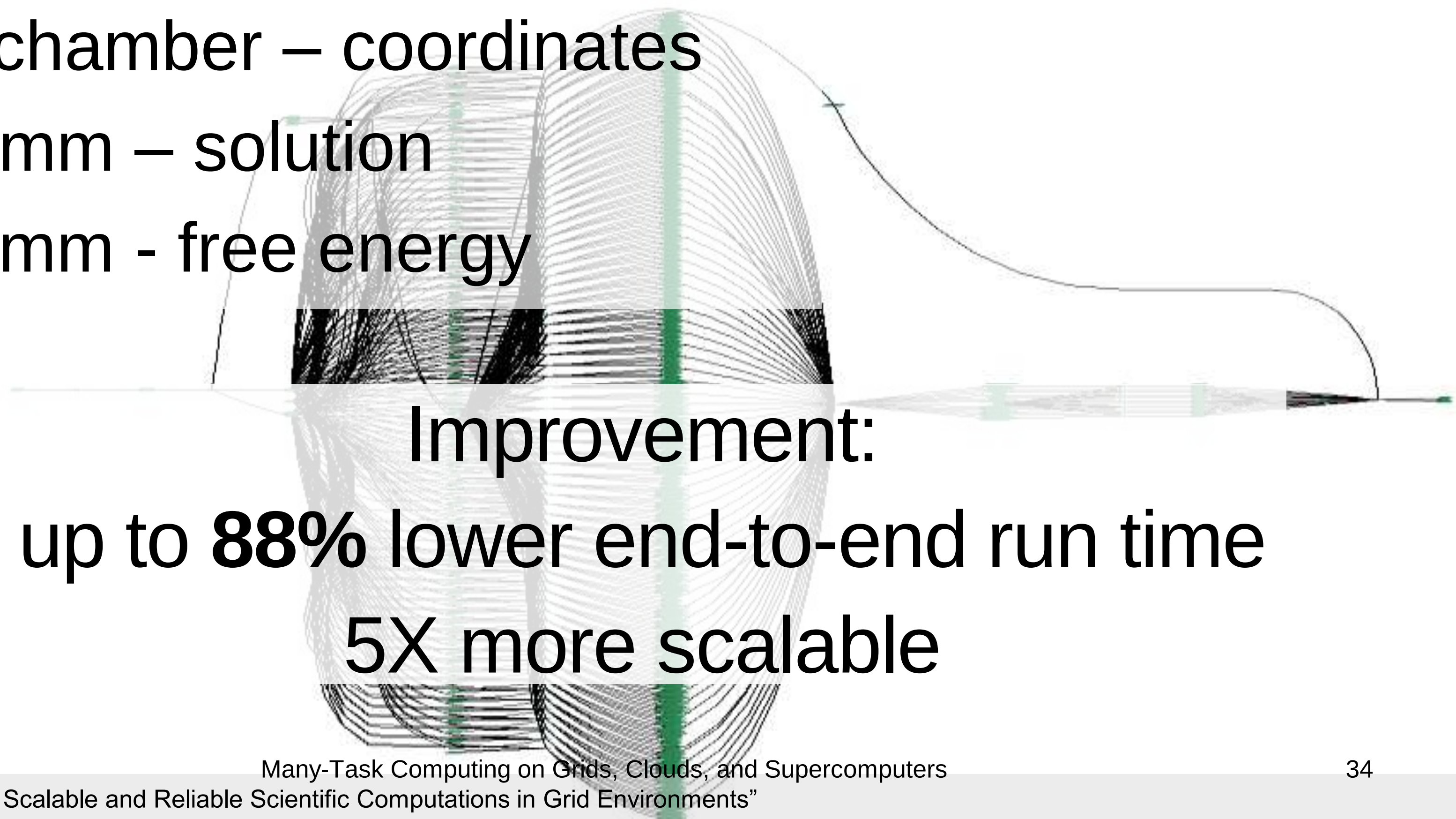
B. Berriman, J. Good (Caltech)
J. Jacob, D. Katz (JPL)



[SC07] “Falkon: a Fast and Light-weight tasK executiON framework”
[SWF07] “Swift: Fast, Reliable, Loosely Coupled Parallel Computation”

Applications
Molecular Dynamics: MolDyn

- Determination of free energies in aqueous solution
  - Antechamber – coordinates
  - Charmm – solution
  - Charmm - free energy



Improvement:

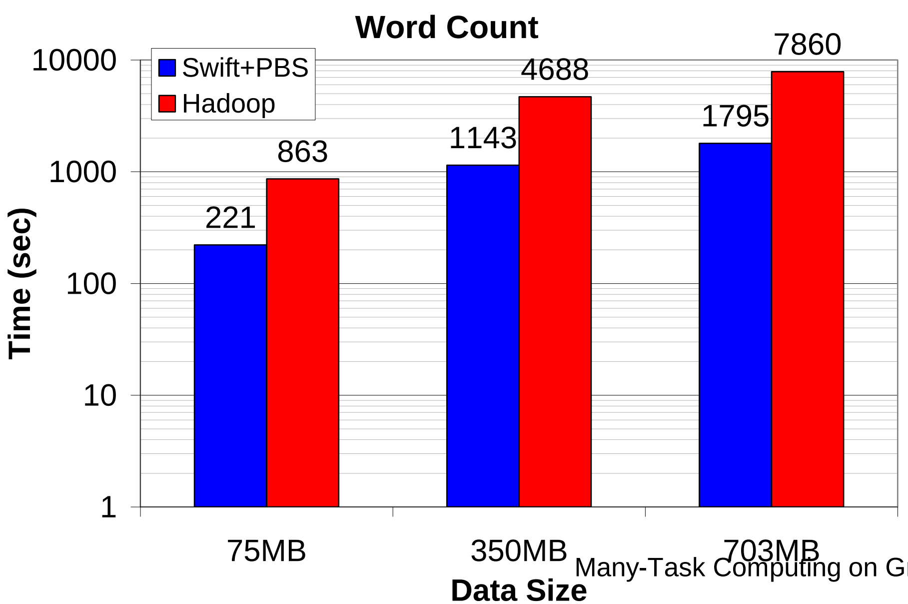
up to **88%** lower end-to-end run time

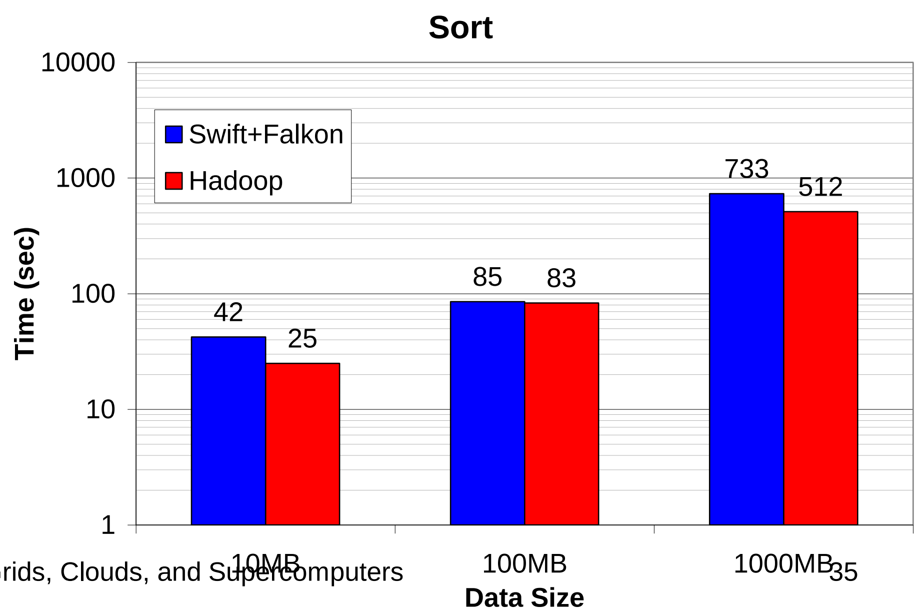
5X more scalable

[NOVA08] "Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments"

Applications
Word Count and Sort

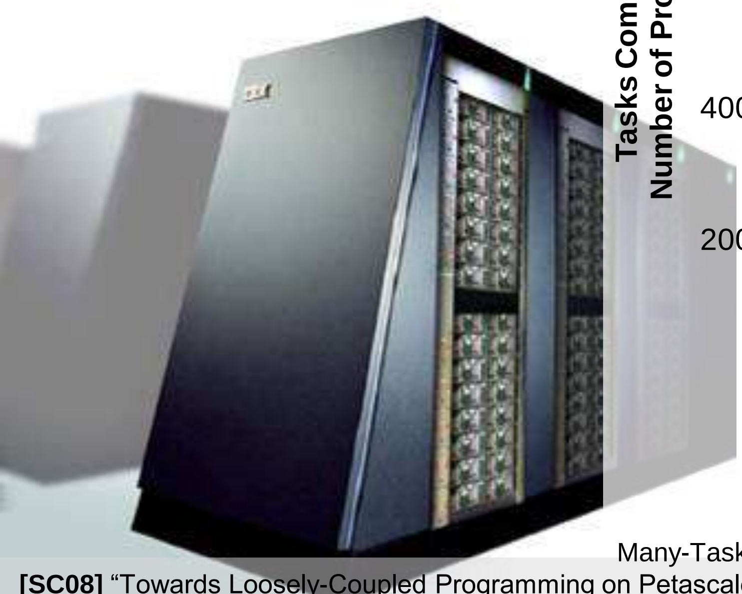
- Classic benchmarks for MapReduce
  - Word Count
  - Sort
- Swift and Falkon performs similar or better than Hadoop (on 32 processors)

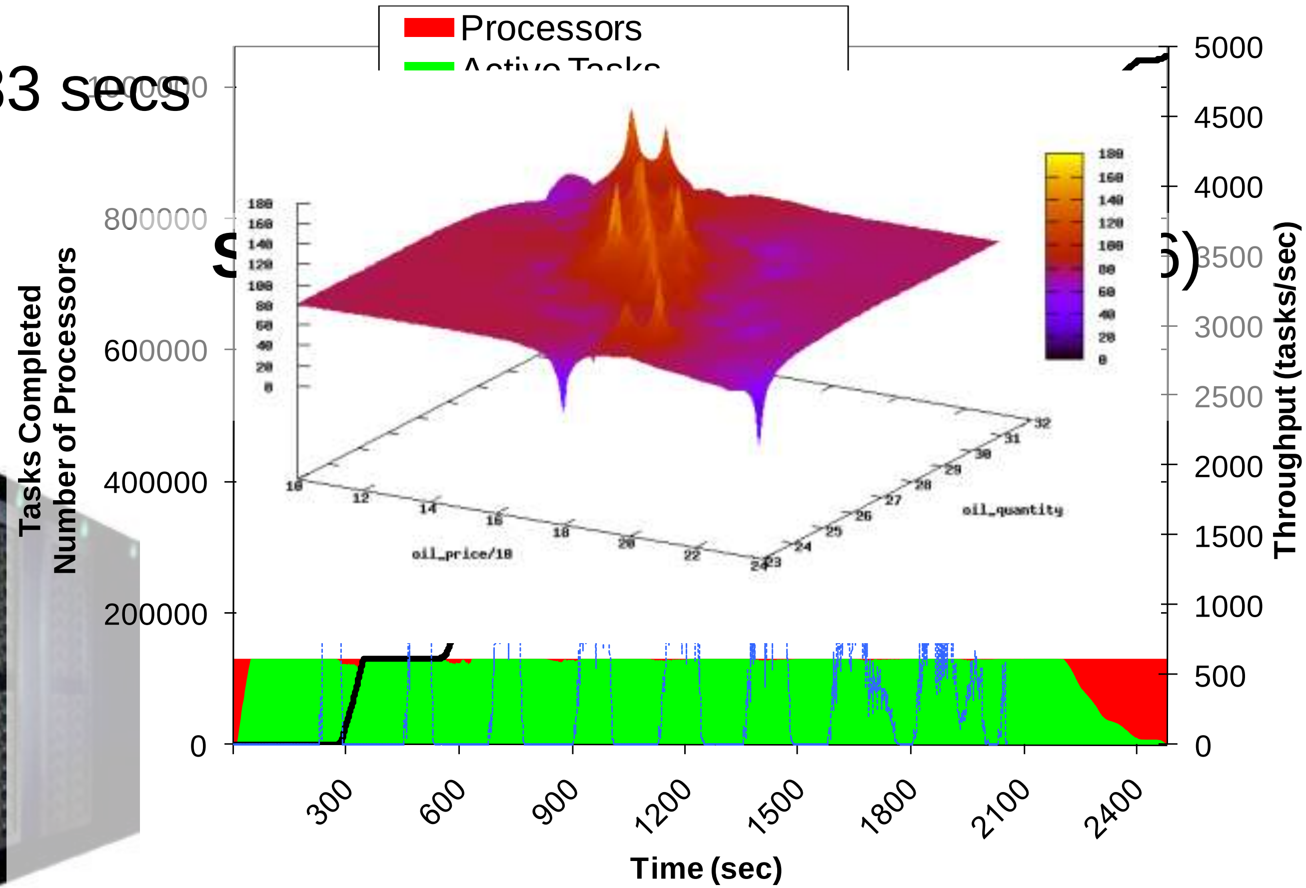




Applications
Economic Modeling: MARS

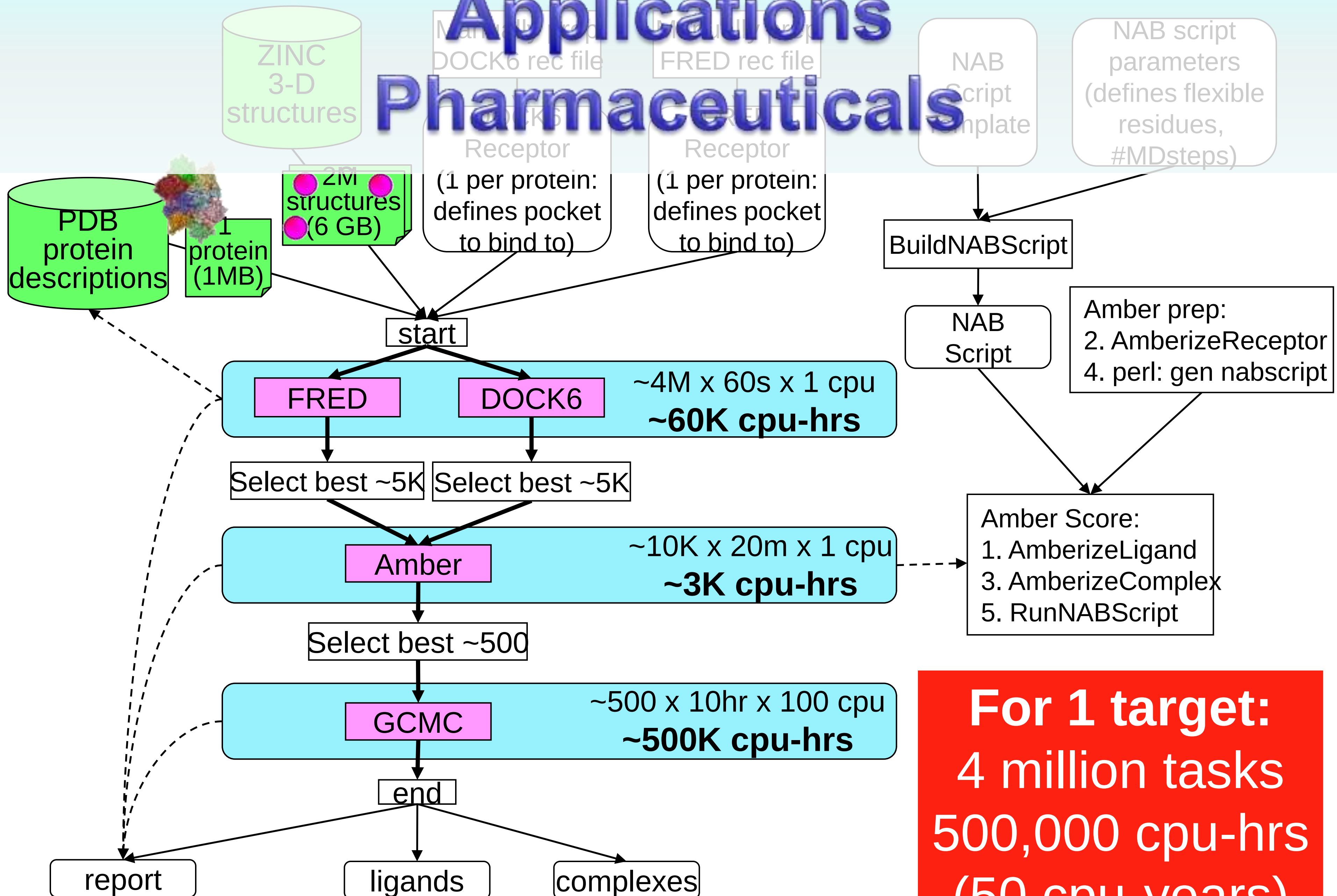
- CPU Cores: 130816
- Tasks: 1048576
- Elapsed time: 2483 secs
- CPU Years: 9.3





[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

Applications: Pharmaceuticals



For 1 target:
4 million tasks
500,000 cpu-hrs
(50 cpu-years)

[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

Applications
Pharmaceuticals: DOCK

CPU cores: 118784

Tasks: 934803

Elapsed time: 2.01 hours

Compute time: 21.43 CPU years

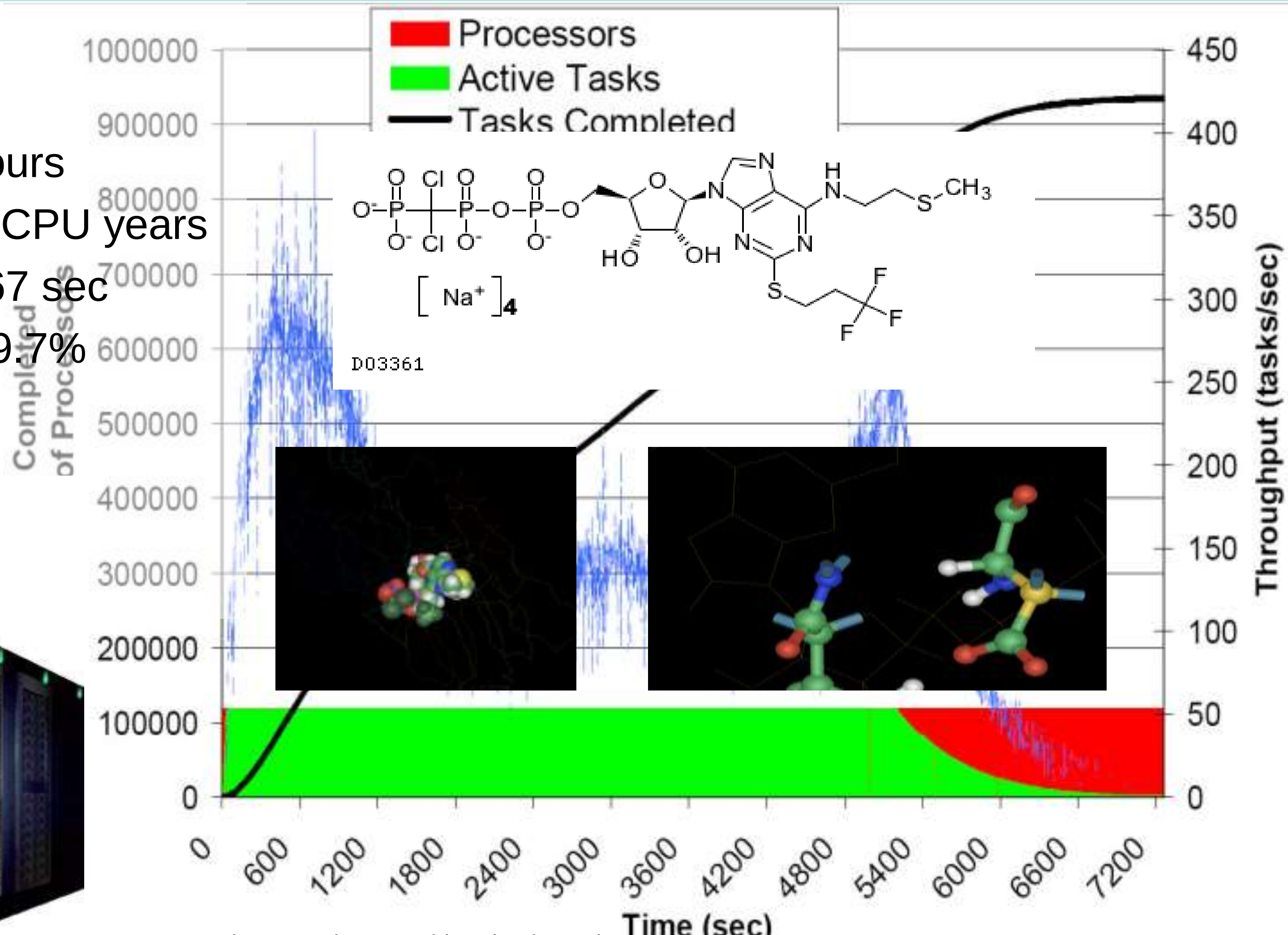
Average task time: 667 sec

Relative Efficiency: 99.7%

(from 16 to 32 racks)

Utilization:

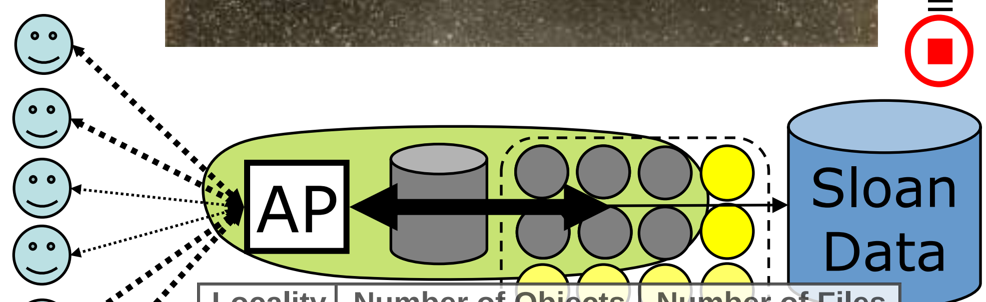
- Sustained: 99.6%
- Overall: 78.3%



[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

Applications
Astronomy: AstroPortal

- Purpose
  - On-demand “stacks” of random locations within ~10TB dataset
- Challenge
  - Processing Costs:
    - O(100ms) per object
  - Data Intensive:
    - 40MB:1sec
  - Rapid access to 10-10K “random” files
  - Time-varying load

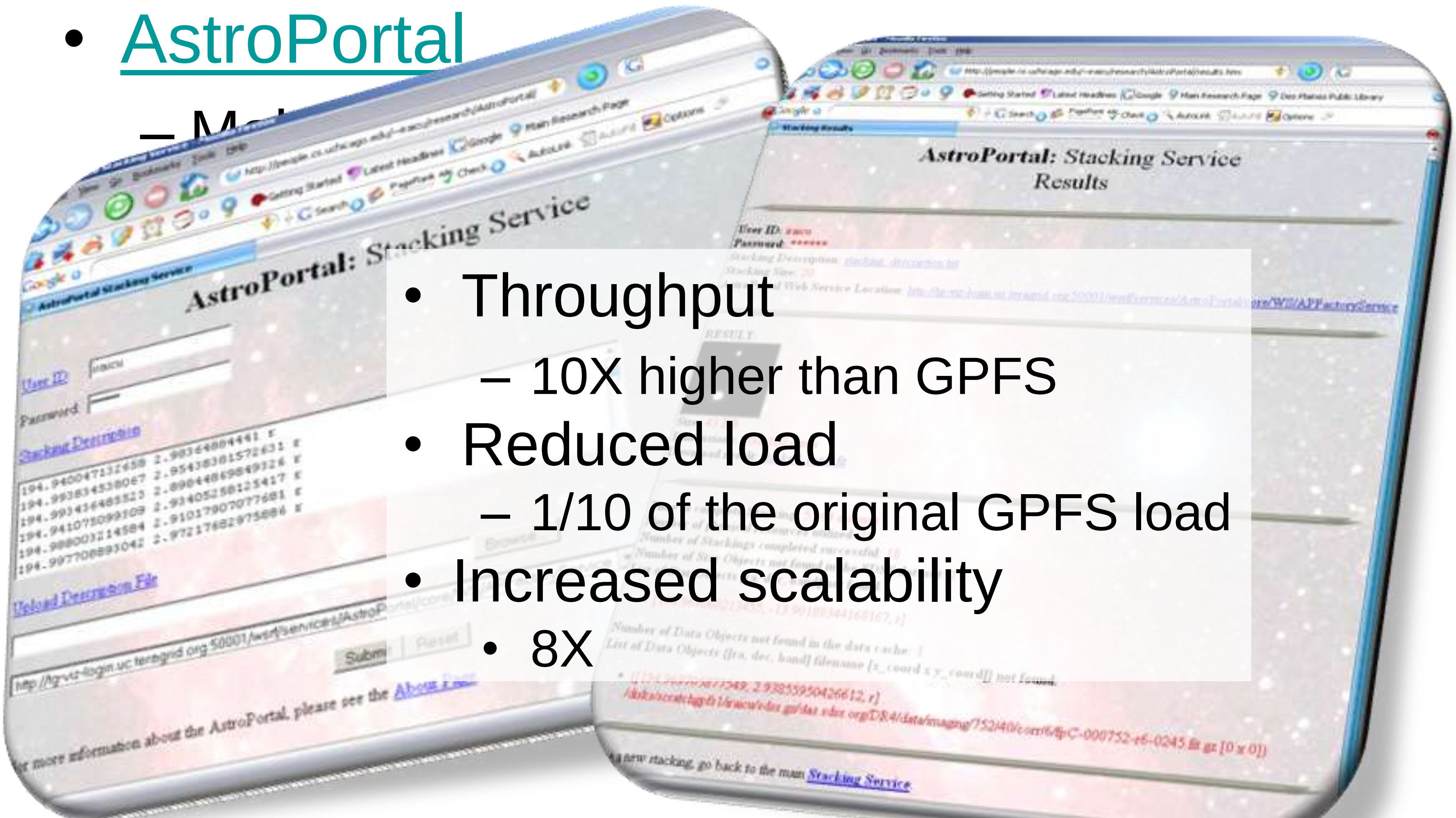


Locality	Number of Objects	Number of Files
1	111700	111700
1.38	154345	111699
2	97999	49000
3	88857	29620
4	76575	19145
5	60590	12120
10	46480	4650
20	40460	2025
30	23695	790

[DADC08] “Accelerating Large-scale Data Exploration through Data Diffusion”
[TG06] “AstroPortal: A Science Gateway for Large-scale Astronomy Data Analysis”

Applications
Astronomy: AstroPortal

- AstroPortal



- Throughput
  - 10X higher than GPFS
- Reduced load
  - 1/10 of the original GPFS load
- Increased scalability
  - 8X

Outline

- Defining Many-Task Computing
- Motivation: Scalability Challenges
- Novel Resource Management Techniques
- Performance Evaluation: Micro-benchmarks
- Performance Evaluation: Applications
- **Contributions**
- Future Work

Contributions

- There is more to HPC than tightly coupled MPI, and more to HTC than embarrassingly parallel long jobs
  - MTC: Many-Task Computing
  - Addressed real challenges in resource management in large scale distributed systems to enable MTC
  - Covered many domains (via Swift and Falkon): astronomy, medicine, chemistry, molecular dynamics, economic modelling, and data analytics

- Identified that data locality is critical at large-scale → data diffusion
  - Integrated streamlined task dispatching with data aware scheduling
  - Heuristics to maximize real world performance
  - Suitable for varying, data-intensive workloads
  - Proof of O(NM) Competitive Caching

Publications and Service

- Publications
  - 51 articles and proposals
  - 66 formal presentations
  - 854 citations ➔ H-Index 14
- Activities for broader community engagement
  - **ScienceCloud**: ACM Workshop on Scientific Cloud Computing, 2010
  - **TPDS**: IEEE Transactions on Parallel and Distributed Systems, Special Issue on Many-Task Computing, 2010
  - **MTAGS**: ACM Workshop on Many-Task Computing on Grids and Supercomputers, 2009
  - **MTAGS**: IEEE Workshop on Many-Task Computing on Grids and Supercomputers, 2008
  - **BegaJob**: Bird of Feather Session – "How to Run One Million Jobs", at IEEE/ACM SC08, 2008
  - 53 more activities as a program committee member and/or reviewer

Teaching

- **Introduction to Programming (Spring 2010)**
  - Northwestern University, Instructor
- **Hot Topics in Distributed Systems: Data-Intensive Computing (Winter 2010)**
  - Northwestern University (EECS495), Instructor
  - http://www.eecs.northwestern.edu/~iraicu/teaching/EECS495-DIC/index.html
- Networks and Distributed Systems (2006)
  - University of Chicago (CMSC 33300), Lead TA
  - http://dsl.cs.uchicago.edu/Courses/CMSC33300/index.html
- Grid Computing (2005)
  - University of Chicago (CMSC 33340), TA
  - http://www.mcs.anl.gov/~itf/CMSC23340/
- Introduction to Networking (2003)
  - Purdue University, Lab TA
- **Data Structures and Algorithm Analysis in C++ (2002)**
  - University of Michigan, Adjunct Assistant Professor

- 9 more courses at University of Chicago and Wayne State University
  - Advanced Network Design, Introduction to Programming for the World Wide Web I, Introduction to Computer Science 1 & 2, Introduction to Computer Systems , Fundamentals of Computer Programming I in Scheme, Problem Solving & Programming in C++, Data Structures & Abstraction in C++

Mythbusting

- ~~Embarrassingly~~ Happily parallel apps are trivial to run
  - Logistical problems can be tremendous
- Loosely coupled apps do not require "supercomputers"
  - Total computational requirements can be enormous
  - Individual tasks may be tightly coupled
  - Workloads frequently involve large amounts of I/O
  - Make use of idle resources from "supercomputers" via backfilling
  - Costs to run "supercomputers" per FLOP is among the best

"Impossible only means that you haven't found the solution yet."

Anonymous

- Loosely coupled apps do not require specialized system software
  - Their requirements on the job submission and storage systems can be extremely large
- Shared/parallel file systems are good for all applications
  - They don't scale proportionally with the compute resources
  - Data intensive applications don't perform and scale well
  - Growing compute/storage gap

Outline

- Defining Many-Task Computing
- Motivation: Scalability Challenges
- Novel Resource Management Techniques
- Performance Evaluation: Micro-benchmarks
- Performance Evaluation: Applications
- Contributions
- **Future Work**

Future Work

- Many-Task Computing and Data Intensive Computing
  - Develop theoretical and practical aspects of building efficient and scalable support for MTC
  - Build a new distributed data-aware execution fabric that will support HPC, MTC, and HTC
  - Support interactive HPC applications
- Cloud Computing
  - Enable scientific computing on clouds (e.g. Magellan project)
  - Enable HPC through novel networking
- Many-Core Computing
  - Apply data-aware scheduling of jobs from distributed systems ➔ OS scheduling of threads
  - Parallel programming models ➔ automatic parallelization using DAG-based data-flow techniques

Future Work (cont)

- Maintain and expand collaborations
  - **Government Labs:** ANL, FNAL, NASA, LBNL,
  - **Academia:** UC, CI, UIC, DePaul, NU, ND, Purdue, UIUC, IU, UWM, WSU, LSU, UNC, USF, UBC, DU, MU
  - **Industry:** Accenture, IBM, MS, Google, Yahoo, Amazon, Intel
- Interdisciplinary research
  - Bring HEC to the science domain
  - Many applicable domains: *astronomy, astrophysics, economic modeling, pharmaceuticals, chemistry, bioinformatics, neuroscience, data analytics, data mining, biometrics, molecular docking, structural equation modeling, posttranslational protein modification, climate modeling*
- Combining Education with Research
  - NSF CDI, NSF REU, NSF CPATH
- Apply for funding from NSF, DOE, NIH, and industry
  - DOE INCITE, NSF PetaApps, NSF HECURA, NIH R01

More Information

- More information: http://www.eecs.northwestern.edu/~iraicu/
- Related Projects:
  - Falkon: http://dev.globus.org/wiki/Incubator/Falkon
  - Swift: http://www.ci.uchicago.edu/swift/index.php
- People contributing ideas, slides, source code, applications, results, etc
  - Ian Foster, Alex Szalay, Rick Stevens, Mike Wilde, Jim Gray, Catalin Dumitrescu, Yong Zhao, Zhao Zhang, Gabriela Turcu, Ben Clifford, Mihael Hategan, Allan Espinosa, Kamil Iskra, Pete Beckman, Philip Little, Christopher Moretti, Amitabh Chaudhary, Douglas Thain, Quan Pham, Atilla Balkir, Jing Tie, Veronika Nefedova, Sarah Kenny, Gregor von Laszewski, Tiberiu Stef-Praun, Julian Bunn, Andrew Binkowski , Glen Hocky, Donald Hanson, Matthew Cohoon, Fangfang Xia, Mike Kubal, Alok Choudhary…
- Funding:
  - **NASA**: Ames Research Center, Graduate Student Research Program
  - **DOE**: Office of Advanced Scientific Computing Research
  - **NSF**: TeragGrid and Computing Research Innovation Fellow Program