Many-task Computing:

Bridging the Gap between High-Throughput Computing and High-Performance Computing

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

CUCIS Seminar
Northwestern University, EECS
September 21st, 2009

Acknowledgements

- Funding and Support (2003 – 2009)
  - University of Chicago
    - Computer Science
    - Computational Institute
  - Argonne National Laboratory
    - Math and Computer Science Division
    - Argonne Leadership Computing Facility
  - NASA
    - Ames Research Center
- 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)

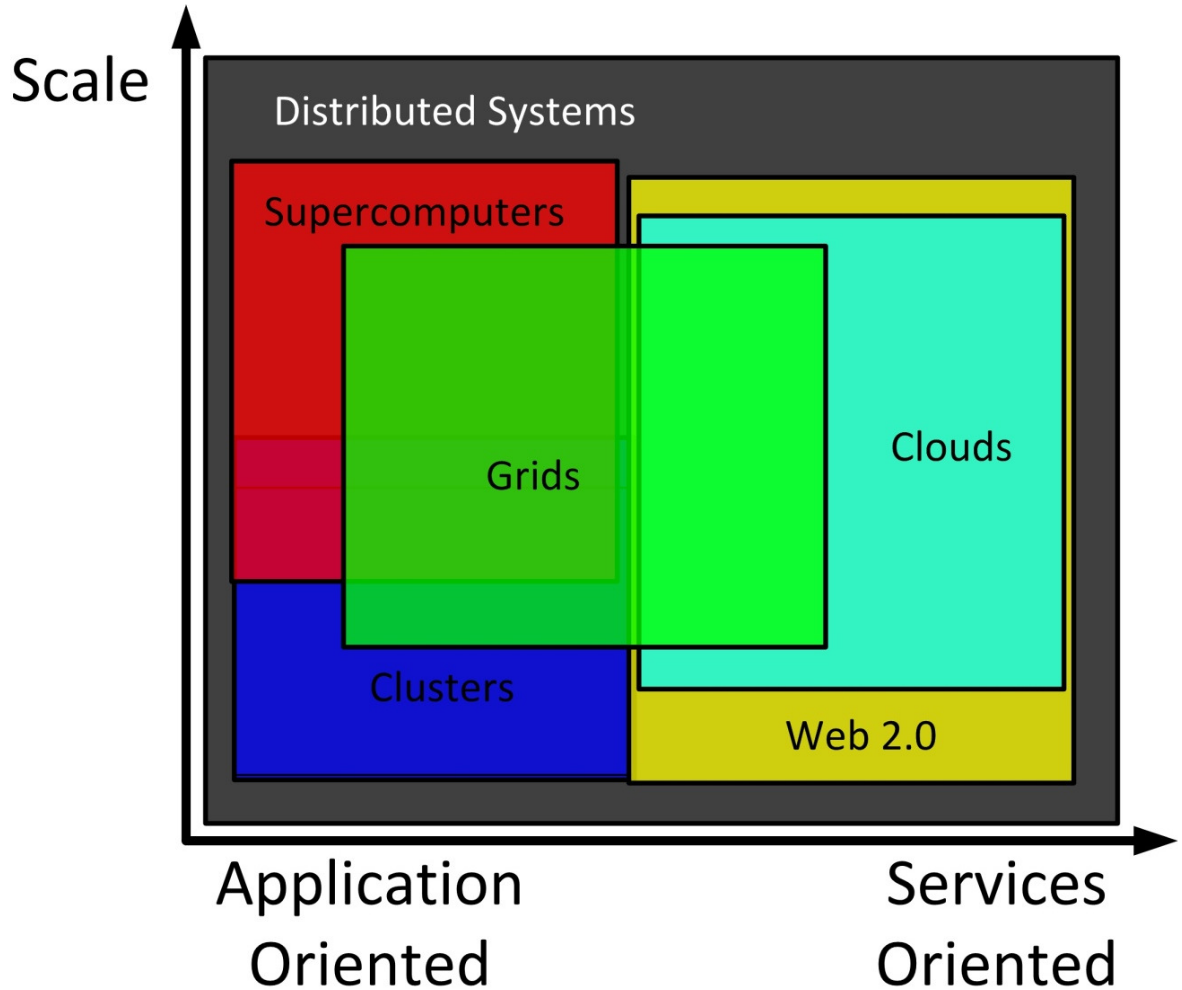
University of Chicago
Argonne National Laboratory

- Worked under Professor Ian Foster
- Large Group
  - Distributed Systems Laboratory, University of Chicago
    - http://dsl-wiki.cs.uchicago.edu/index.php/Main_Page
  - Computational Institute, University of Chicago
    - http://www.ci.uchicago.edu/index.php
  - Math and Computer Science Division, Argonne National Laboratory
    - http://www.mcs.anl.gov/index.php
  - Argonne Leadership Computing Facility
    - http://www.alcf.anl.gov/
- Research Areas:
  - Distributed systems, Grid middleware, Grid applications, Systems Design and Implementation, Data-intensive Computing, Deep Supercomputing, Next Generation Cybertools, Parallel Tools, Collaborative and Virtual Environments, Computational Science
- Many High Impact Projects:
  - Open Science Grid, TeraGrid, Globus, National Microbial Pathogen Research Center, Social Informatics Data Grid, Chicago Biomedical Consortium, Globus Toolkit, MPI, PVFS, IBM Blue Gene/P Supercomputer

Experience with a Variety of Large-Scale Systems

- PlanetLab (912 nodes at 470 sites all over the world)
- ANL SiCortex 5832 (6TF, 5832-cores)
- IBM Blue Gene/P Supercomputer at ANL (~557TF, 160K-cores)
- Sun Constellation Supercomputer (~579TF, 62K-cores)
- Cray XT5 (~1381TF, 150K-cores)
- Open Science Grid (43K-cores across 80 institutions in the US)
- TeraGrid (161K-cores across 11 institutions and 22 systems over the US)

Clusters, Grids, Supercomputers

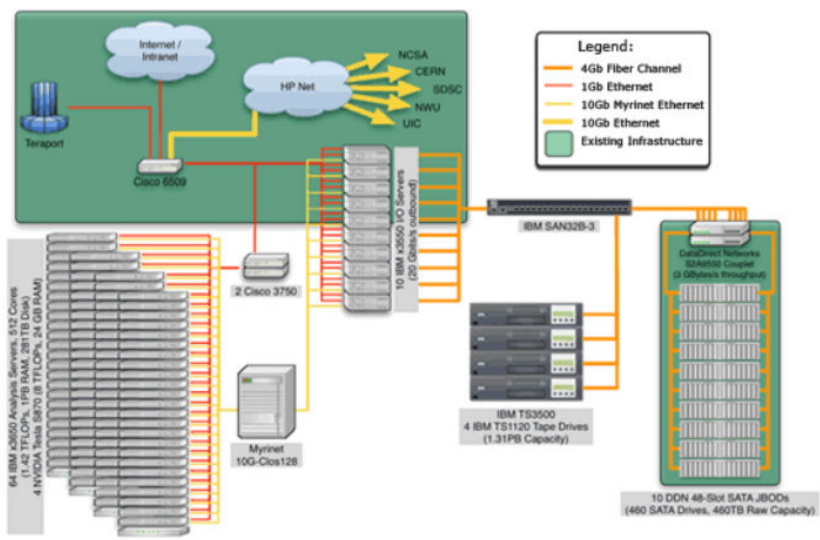


[GCE08] “Cloud Computing and Grid Computing 360-Degree Compared”

Cluster Computing: PADS

PADS

Computer clusters using commodity processors, network interconnects, and operating systems.



System diagram of the PADS environment, showing links to other resources.

PADS is a petabyte (10^{15}-byte)-scale online storage server capable of sustained multi-gigabyte/s I/O performance, tightly integrated with a 9 teraflop/s computing resource and multi-gigabit/s local and wide area networks. Its hardware and associated software enables the reliable storage of, access to, and analysis of massive datasets by both local users and the national scientific community.

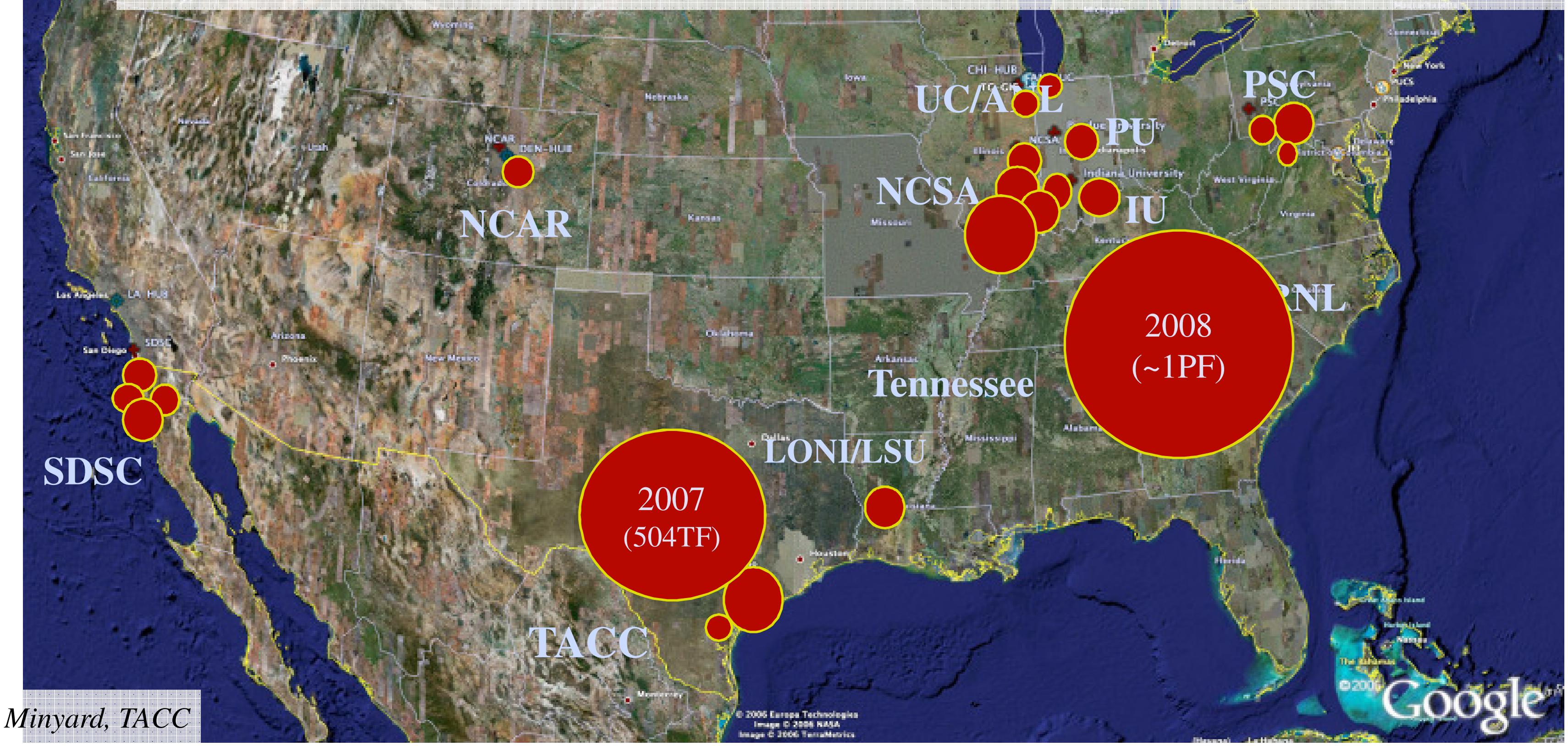
The PADS design results from a study of the storage and analysis requirements of participating groups in astrophysics and astronomy, computer science, economics, evolutionary and organismal biology, geosciences, high-energy physics, linguistics, materials science, neuroscience, psychology, and sociology. For these groups, PADS represents a significant opportunity to look at their data in new ways, enabling new scientific insights. The infrastructure also encourages new collaborations across disciplines. PADS is also a vehicle for computer science research into active data store systems, and provides rich data on which to investigate new techniques. Results will be made available as open source software.

The PADS project is supported in part by the National Science Foundation under grant OCI-0821678 and by The University of Chicago.

PADSstatus myPADS

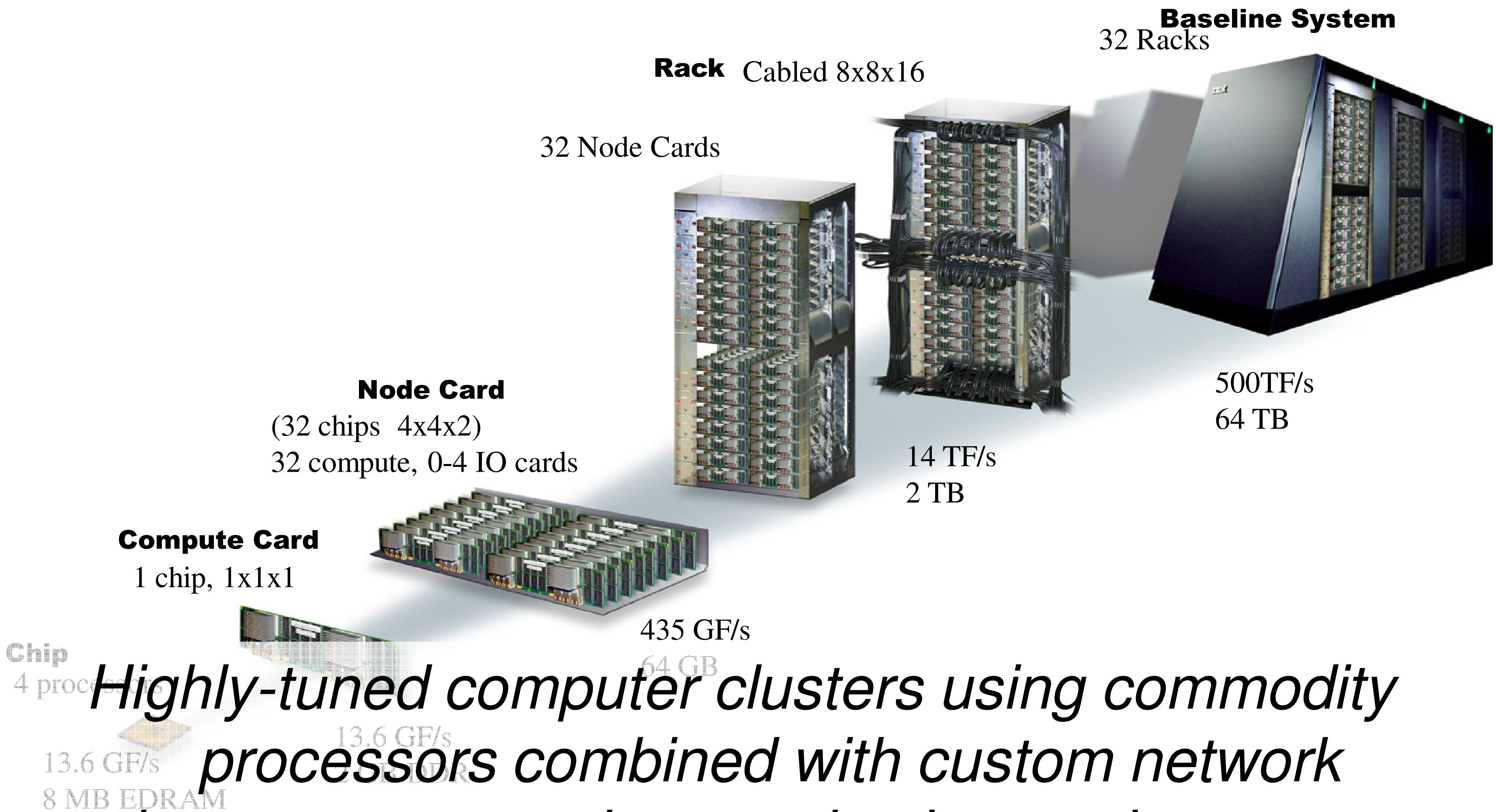
Grid Computing: TeraGrid

Grids tend to be composed of multiple clusters, and are typically loosely coupled, heterogeneous, and geographically dispersed



Tommy Minyard, TACC

Supercomputing: IBM Blue Gene/P



Highly-tuned computer clusters using commodity processors combined with custom network interconnects and customized operating system

High-Throughput Computing & High-Performance Computing

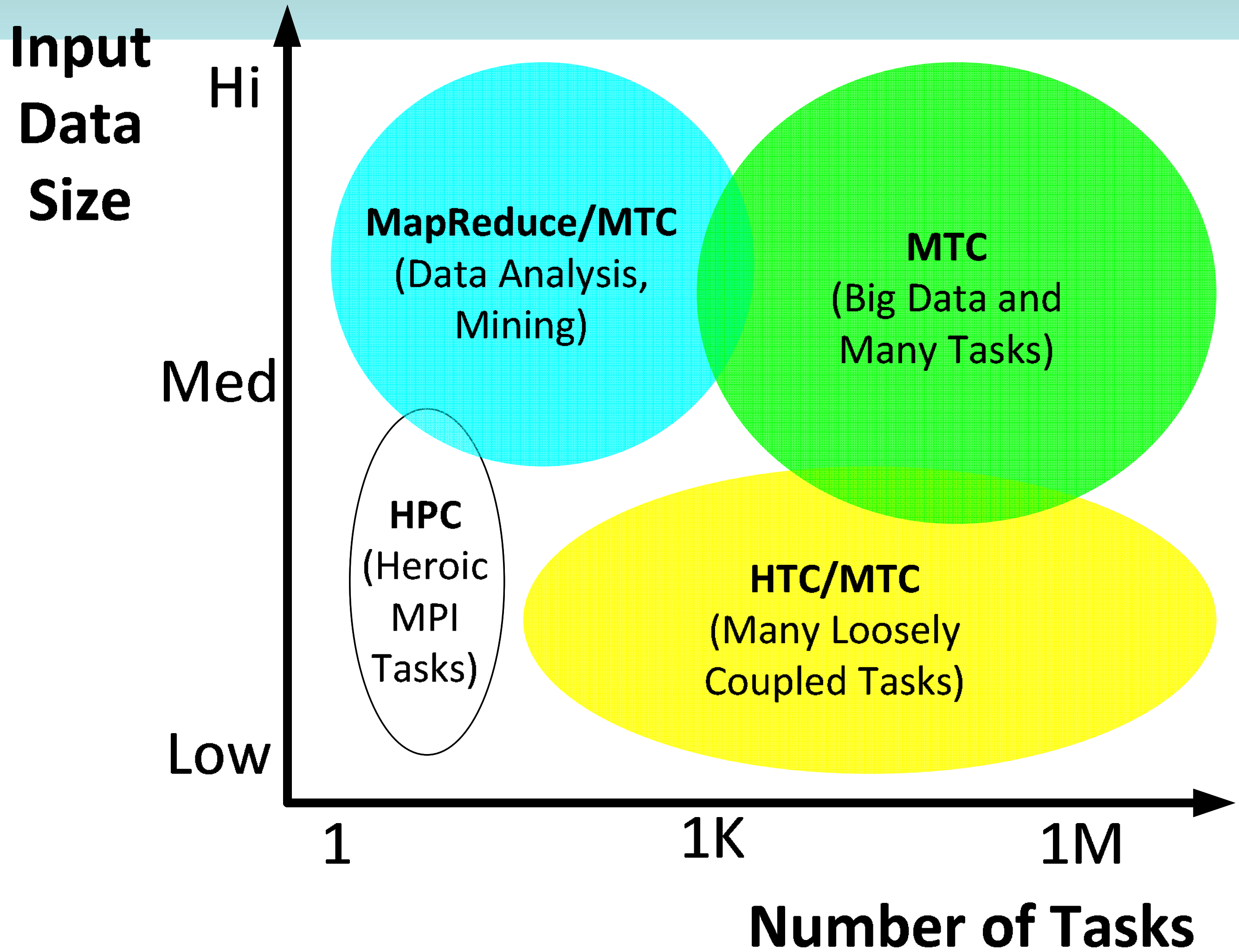
- **HTC: High-Throughput Computing**
  - Typically applied in clusters and grids
  - Loosely-coupled applications with sequential jobs
  - Large amounts of computing for long periods of times
  - Measured in operations per month or years
- **HPC: High-Performance Computing**
  - Synonymous with supercomputing
  - Tightly-coupled applications
  - Implemented using Message Passing Interface (MPI)
  - Large of amounts of computing for short periods of time
  - Usually requires low latency interconnects
  - Measured in FLOPS

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
  - Large number of tasks, large quantity of computing, and large volumes of data

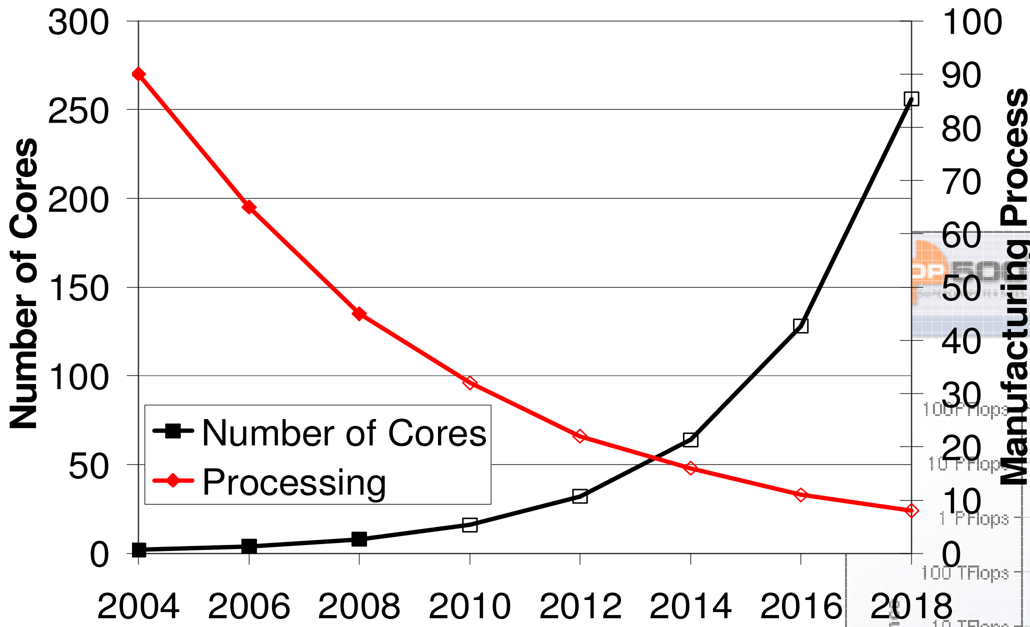
[MTAGS08 Workshop] Workshop on Many-Task Computing on Grids and Supercomputers 2008
[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"
[MTAGS08] "Many-Task Computing for Grids and Supercomputers"

Problem Space

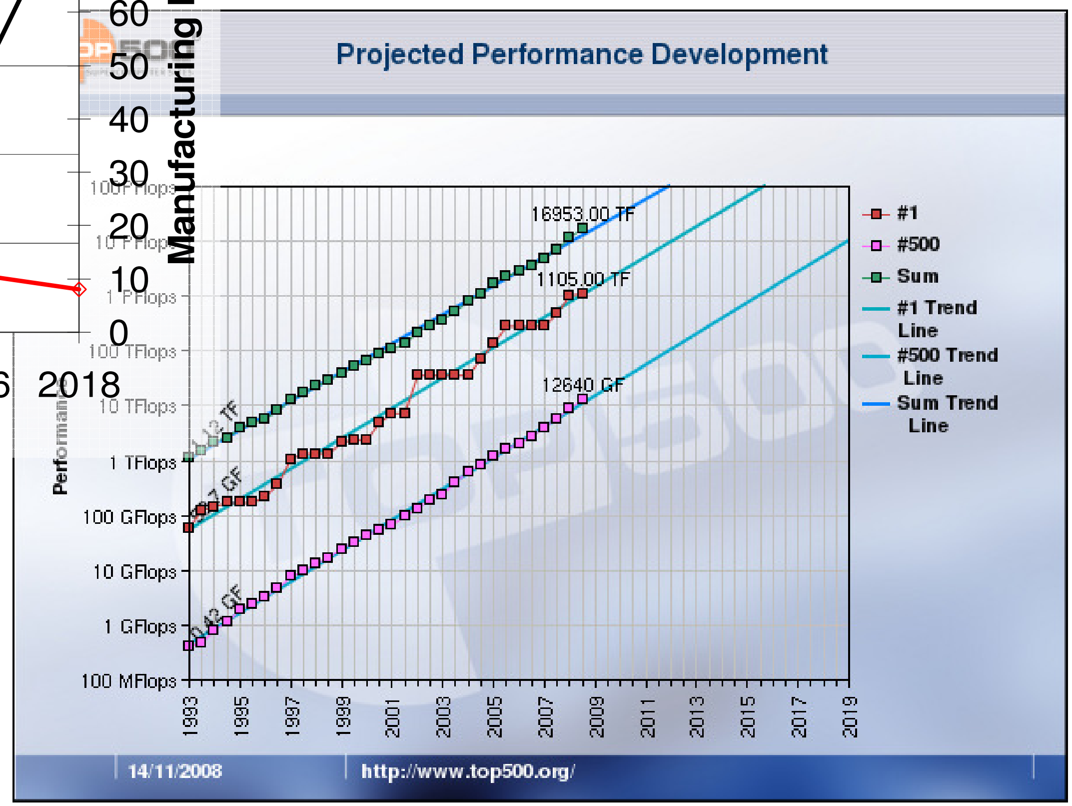


[MTAGS08] "Many-Task Computing for Grids and Supercomputers"

Projected Growth Trends



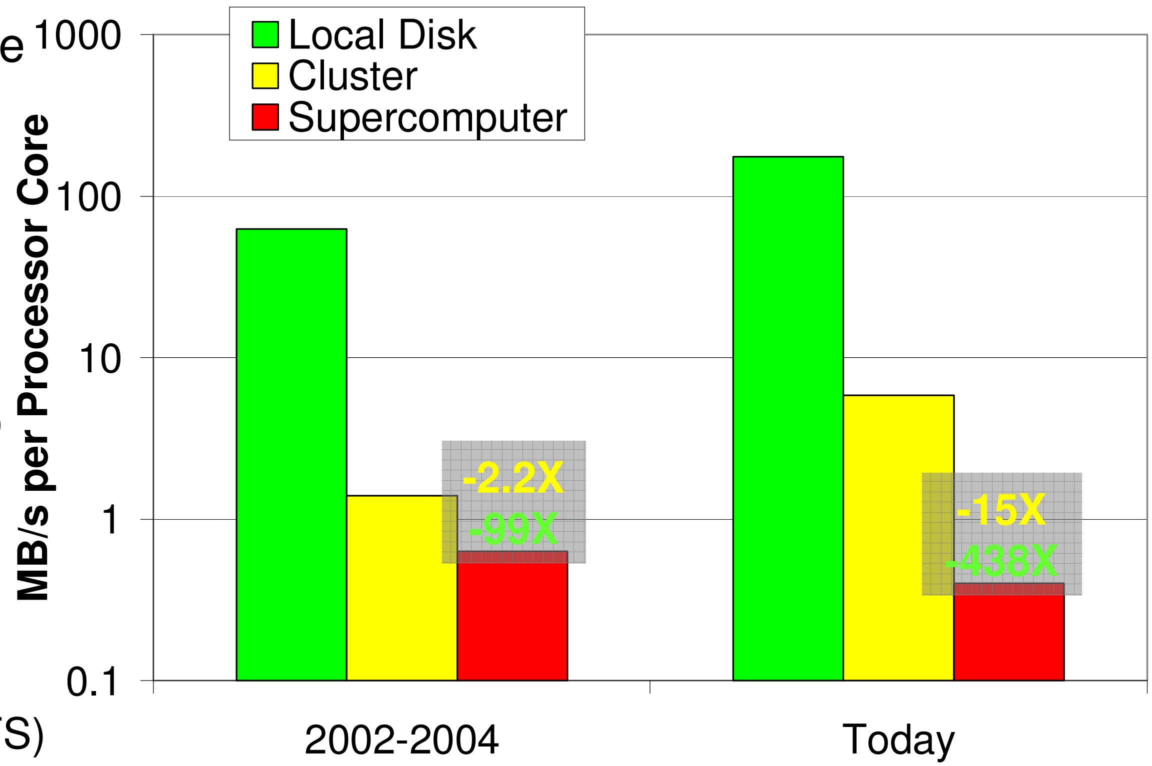
Pat Helland, Microsoft, The Irresistible Forces Meet the Movable Objects, November 9th, 2007



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

Growing Storage/Compute Gap

- Local Disk:
  - 2002-2004: ANL/UC TG Site (70GB SCSI)
  - Today: PADS (RAID-0, 6 drives 750GB SATA)
- Cluster:
  - 2002-2004: ANL/UC TG Site (GPFS, 8 servers, 1Gb/s each)
  - Today: PADS (GPFS, SAN)
- Supercomputer:
  - 2002-2004: IBM Blue Gene/L (GPFS)
  - Today: IBM Blue Gene/P (GPFS)

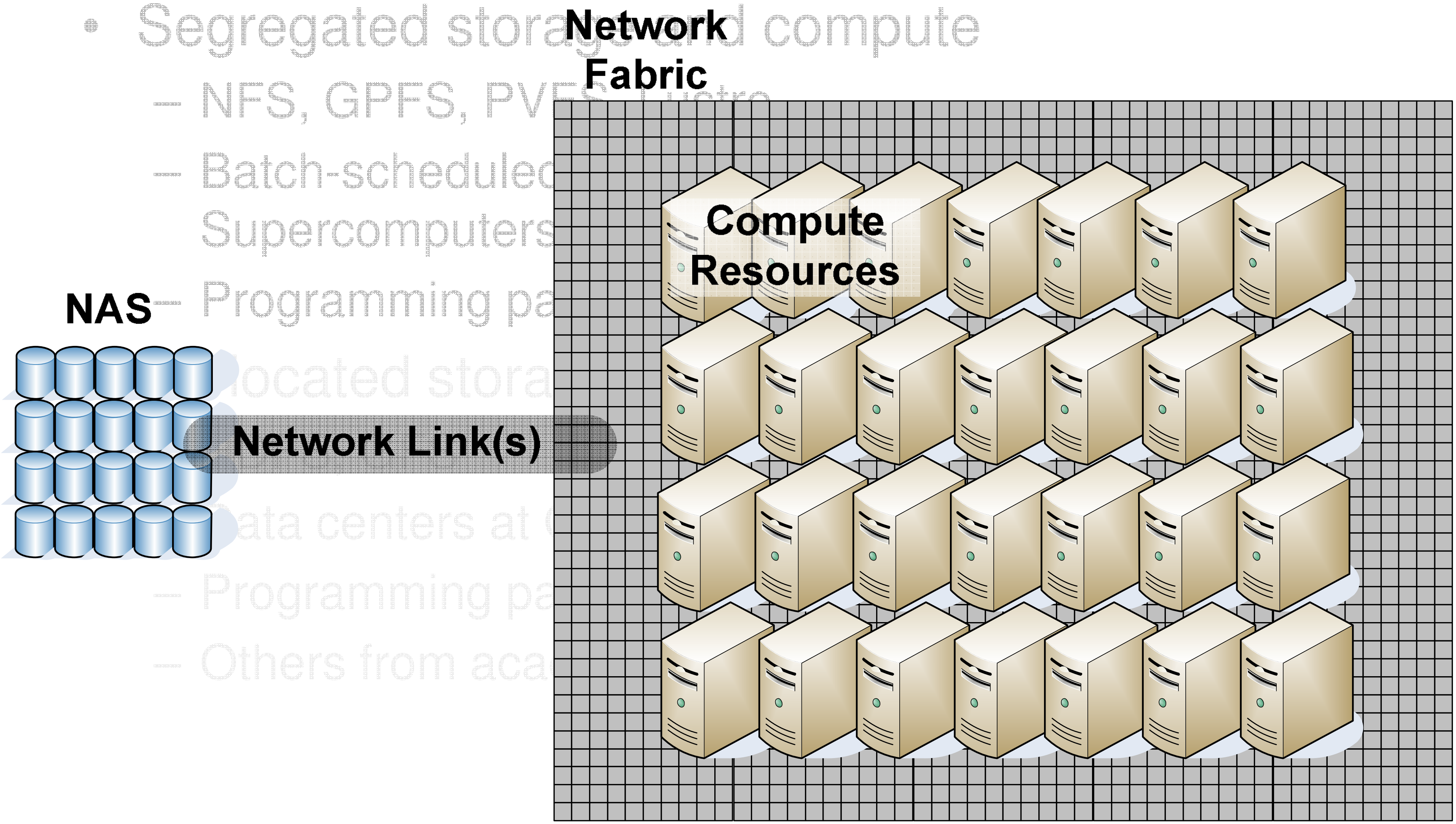


State of the Art: Storage Systems

- Segregated storage and compute
  - NFS, GPFS, PVFS, Lustre
  - Batch-scheduled systems: Clusters, Grids, and Supercomputers
  - Programming paradigm: HPC, MTC, and HTC
- Co-located storage and compute
  - HDFS, GFS
  - Data centers at Google, Yahoo, and others
  - Programming paradigm: MapReduce
  - Others from academia: Sector, MosaStore, Chirp

State of the Art: Storage Systems

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

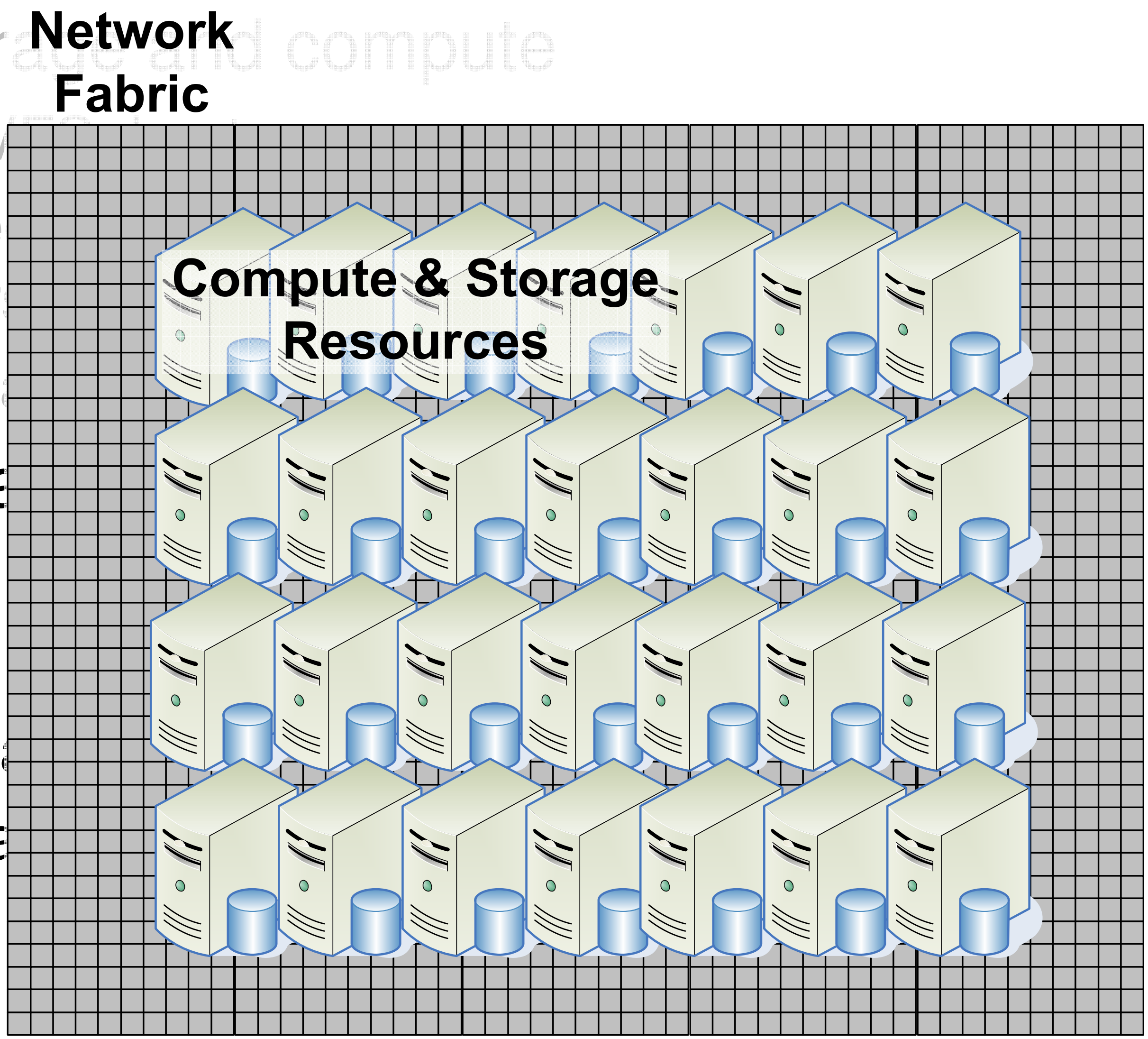


State of the Art: Storage Systems

- Segregated storage and compute
  - NFS, GPFS, PVFS, Lustre
  - Batch-scheduled systems: Clusters, Grids, and Supercomputers
  - Programming paradigm: HPC, MTC, and HTC
- Co-located storage and compute
  - HDFS, GFS
  - Data centers at Google, Yahoo, and others
  - Programming paradigm: MapReduce
  - Others from academia: Sector, MosaStore, Chirp

State of the Art: Storage Systems

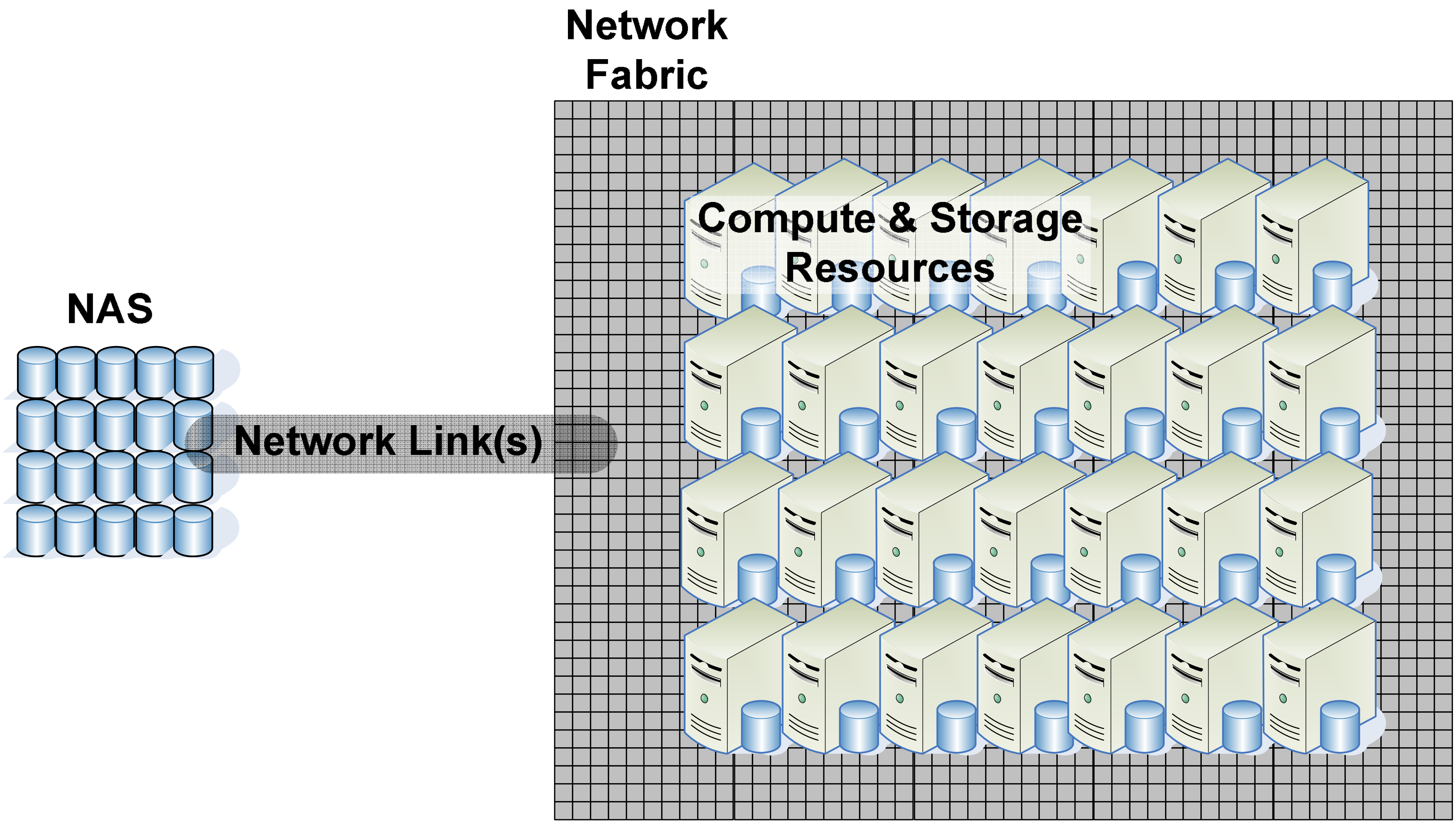
- Segregated storage and compute
  - NFS, GPFS, PV
  - Batch-schedule
    Supercomputer
  - Programming p
- Co-located stora
  - HDFS, GFS
  - Data centers at
  - Programming pa
  - Others from aca



Question

What if we could combine the scientific community's existing programming paradigms, but yet still exploit the data locality that naturally occurs in scientific workloads?

Combine State of the Art Systems



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

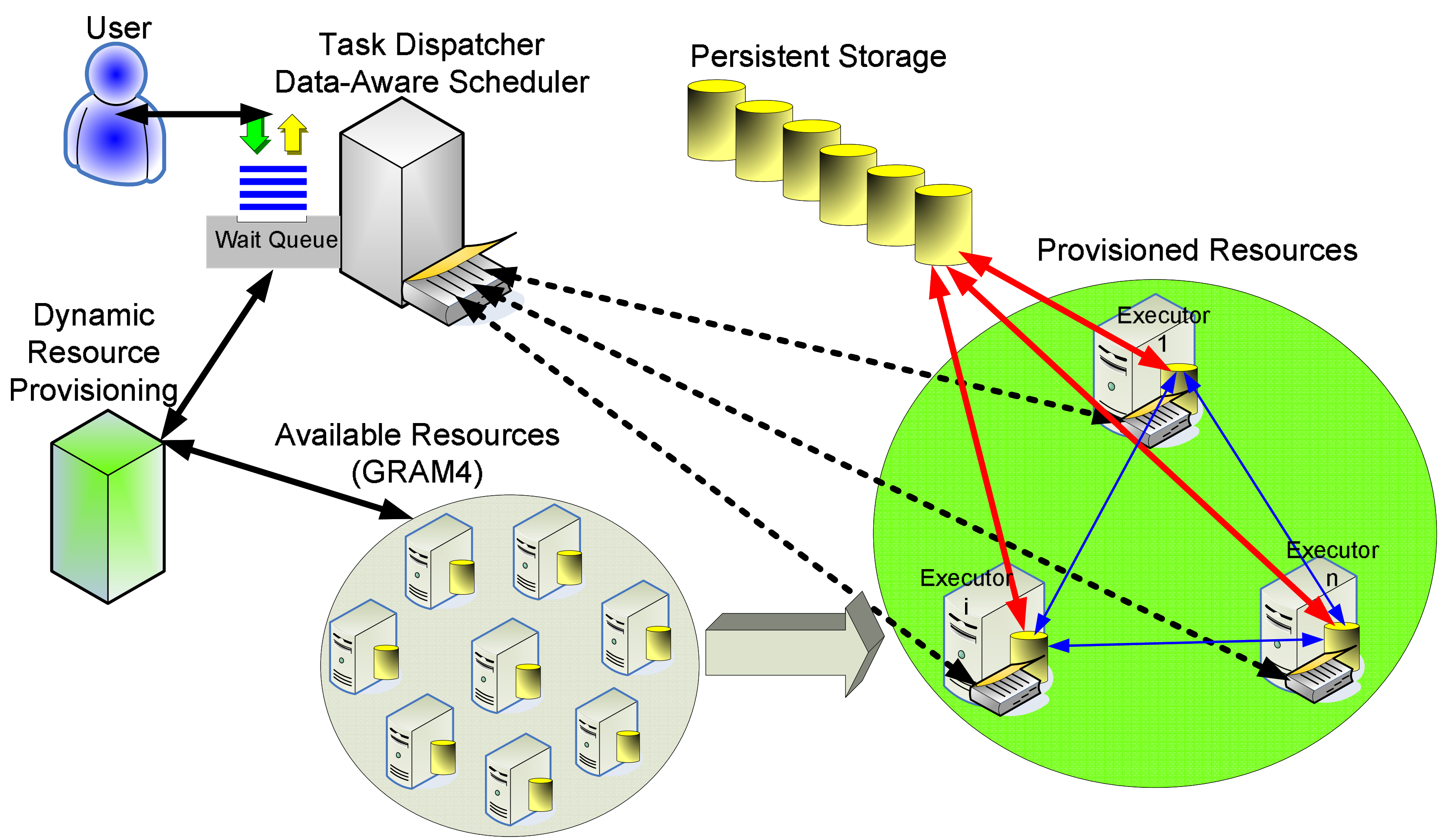
[TPDS10] "Middleware Support for Many-Task Computing", under preparation
[DIDC09] "Towards Data Intensive Many-Task Computing"
[SC07] "Falkon: a Fast and Light-weight tasK executiON framework"

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 Architecture



[SC07] "Falkon: a Fast and Light-weight tasK executiON framework"

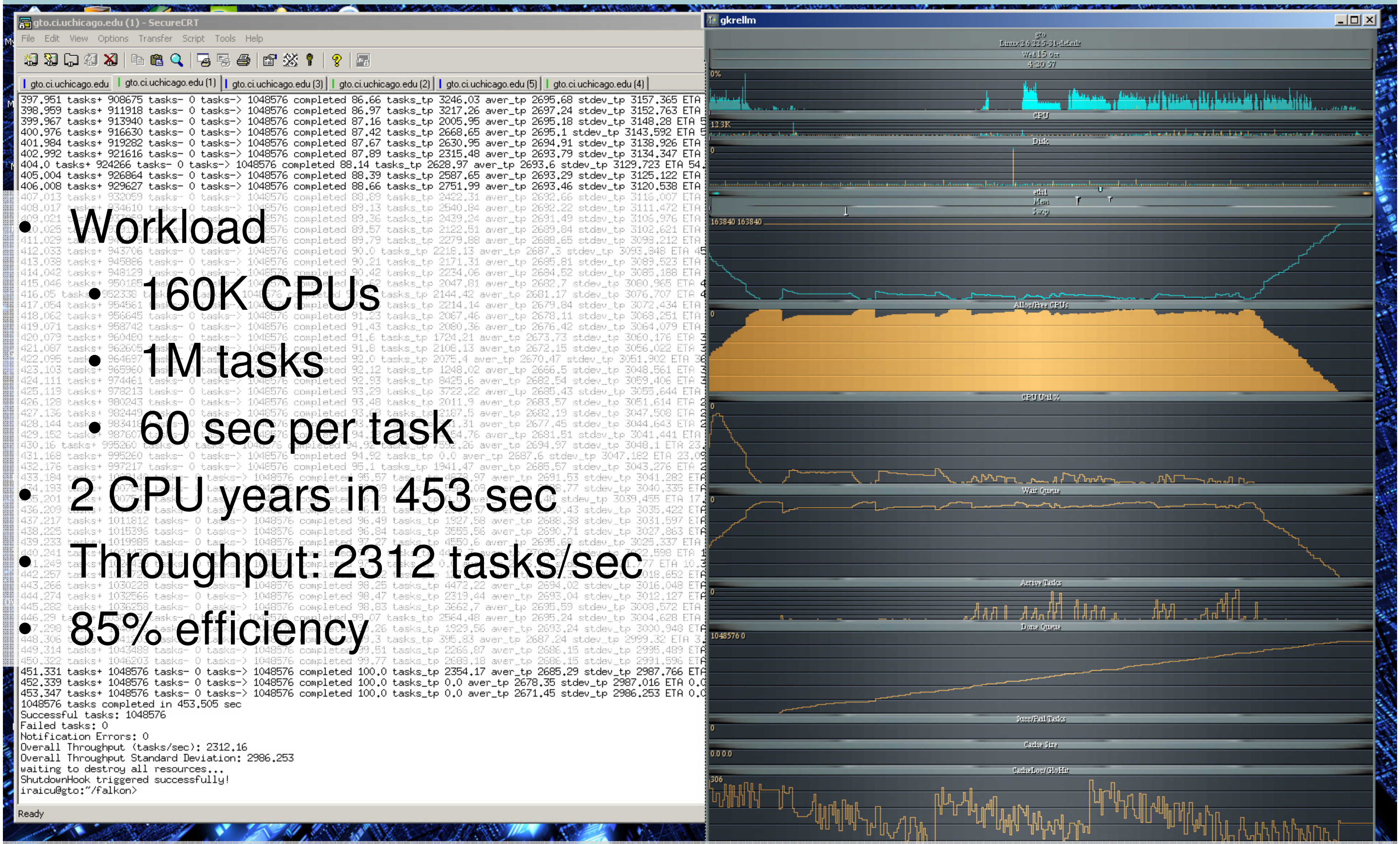
Falkon Project

- Falkon is a real system
  - Late 2005: Initial prototype, AstroPortal
  - January 2007: Falkon v0
  - November 2007: Globus incubator project v0.1
    - http://dev.globus.org/wiki/Incubator/Falkon
  - February 2009: Globus incubator project v0.9
- Implemented in Java (~20K lines of code) and C (~1K lines of code)
  - Open source: svn co https://svn.globus.org/repos/falkon
- Source code contributors (beside myself)
  - Yong Zhao, Zhao Zhang, Ben Clifford, Mihael Hategan

[Globus07] "Falkon: A Proposal for Project Globus Incubation"

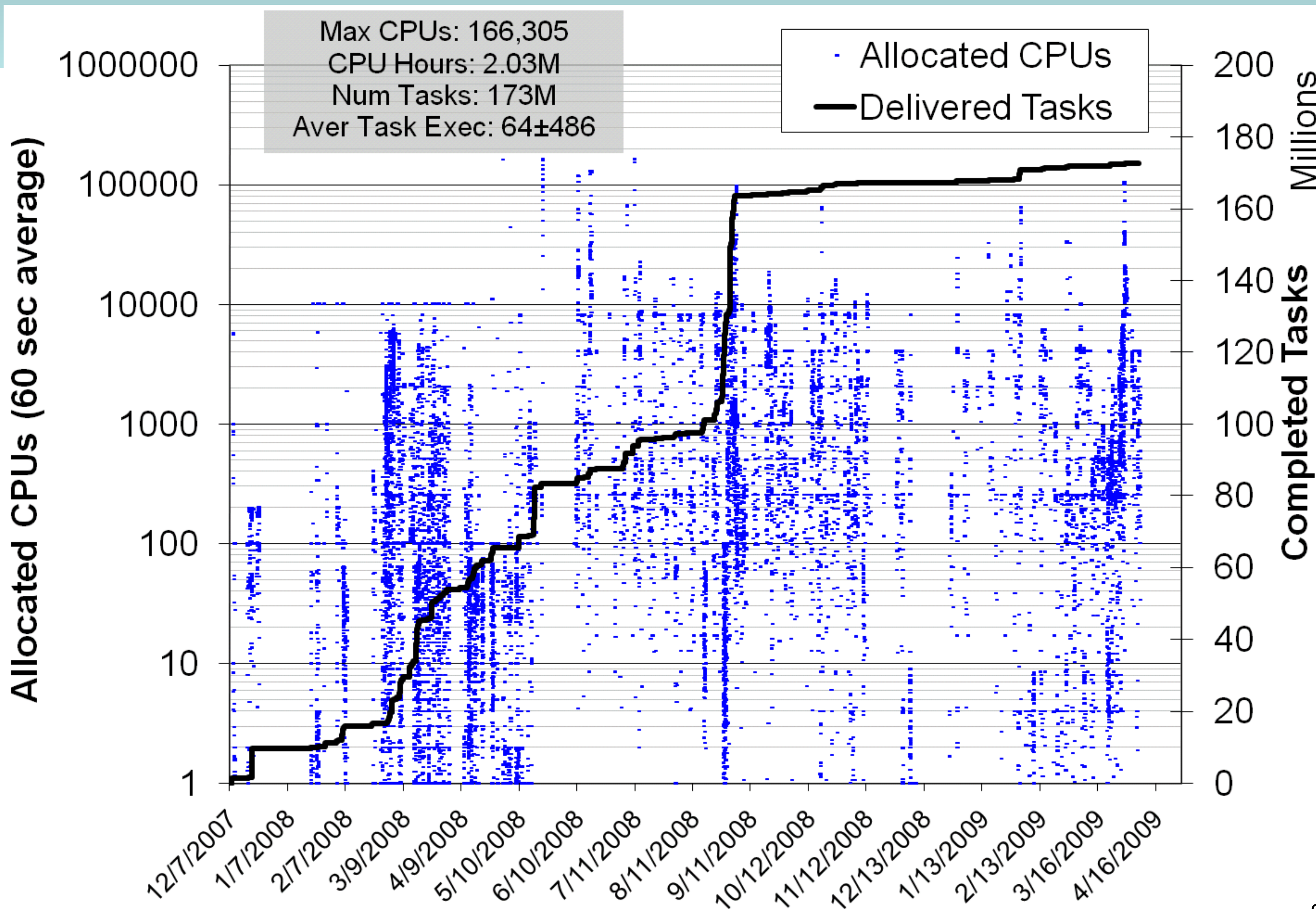
Falkon Monitoring

- Workload
  - 160K CPUs
  - 1M tasks
  - 60 sec per task
- 2 CPU years in 453 sec
- Throughput: 2312 tasks/sec
- 85% efficiency



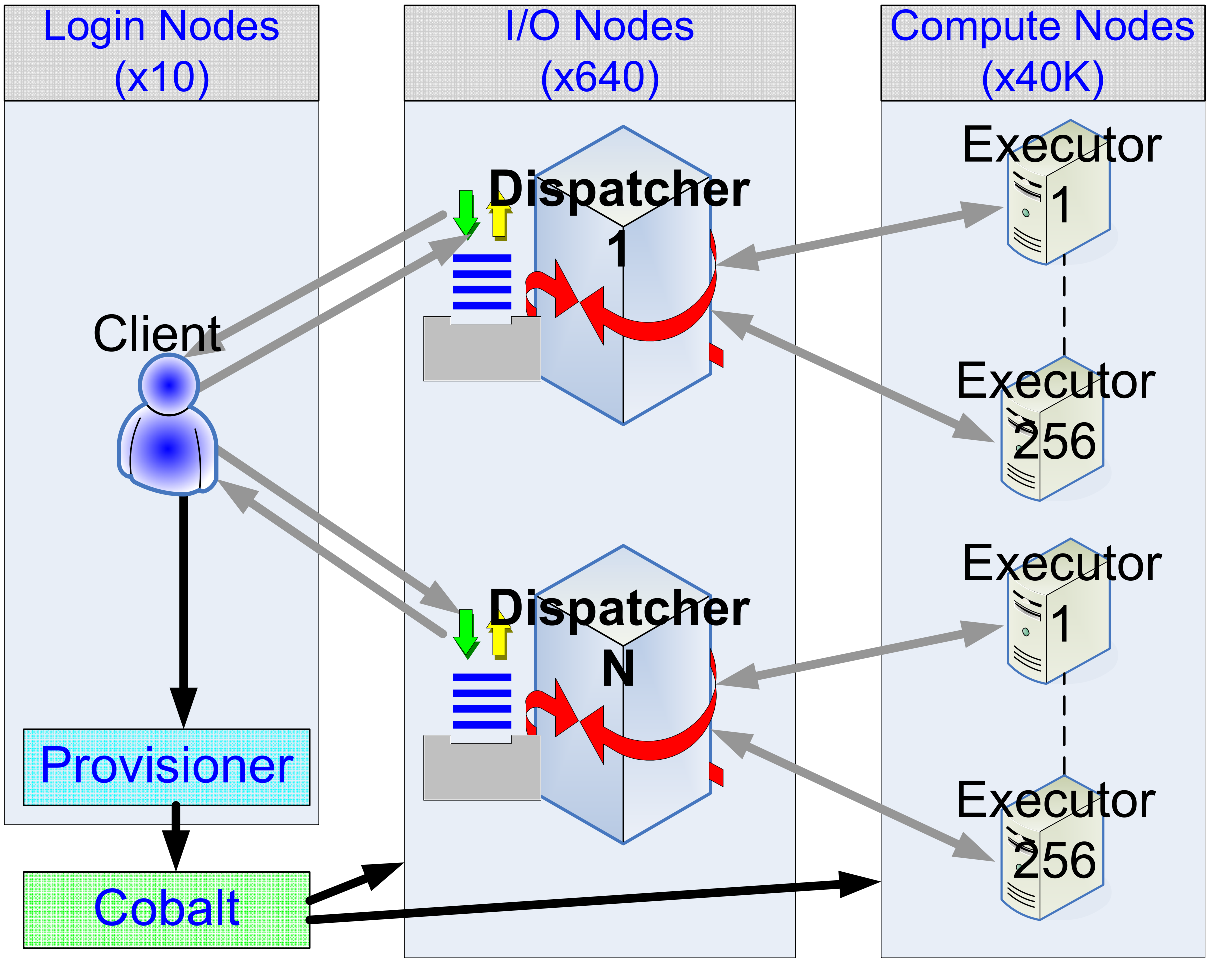
[TPDS09] "Middleware Support for Many-Task Computing", under preparation

Falkon Activity History (16 months)



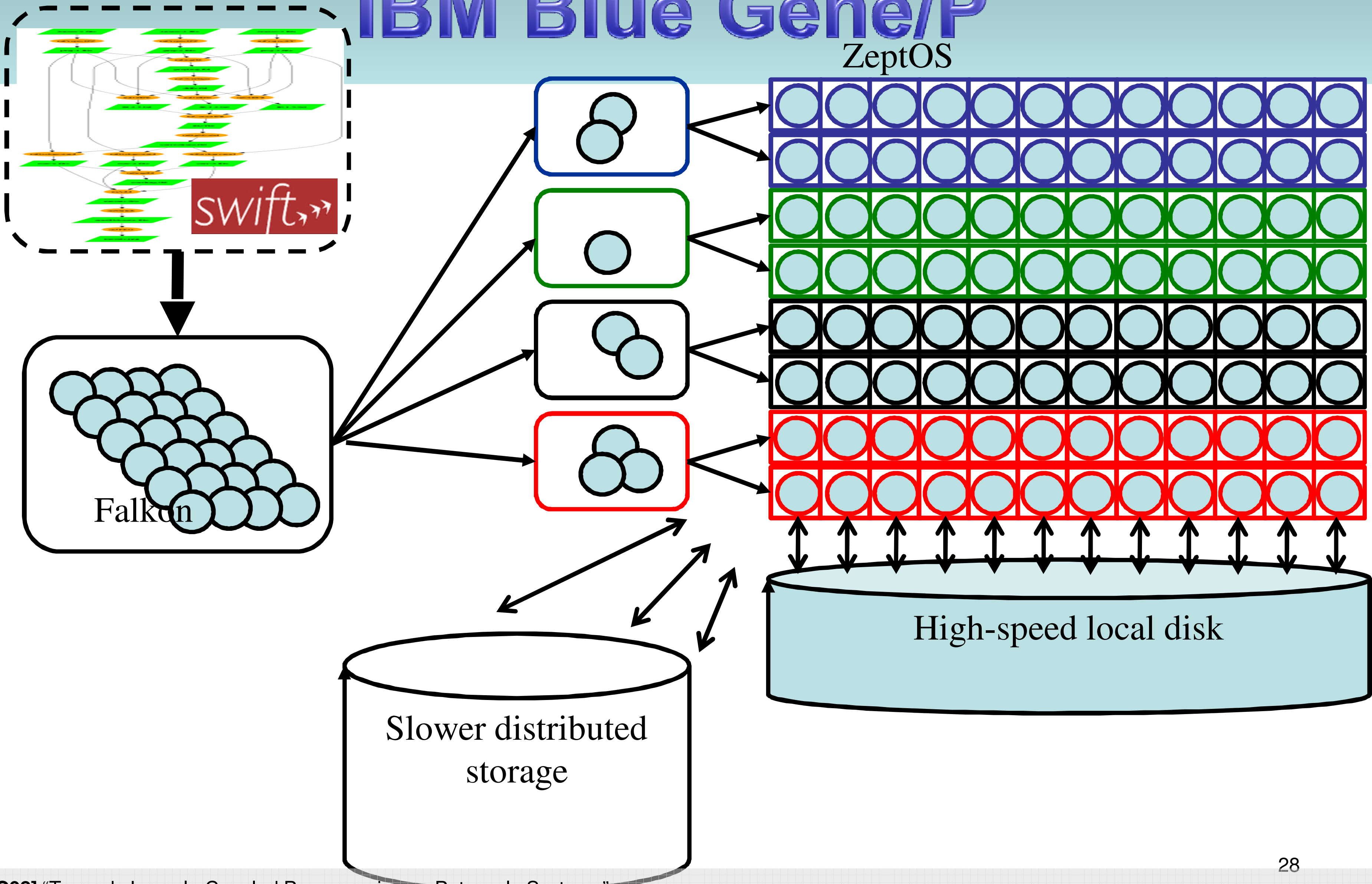
[TPDS09] "Middleware Support for Many-Task Computing", under preparation

Distributed Falkon Architecture



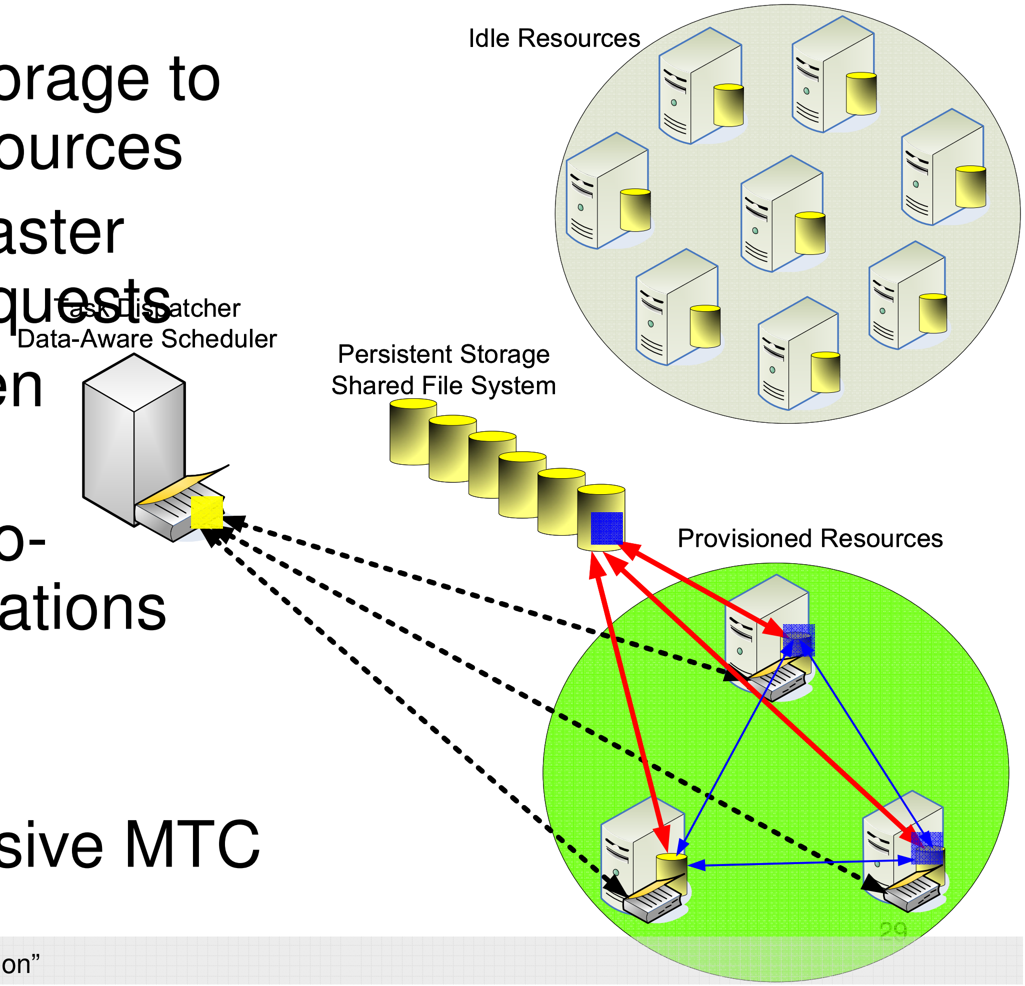
[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

Managing 160K CPUs
IBM Blue Gene/P



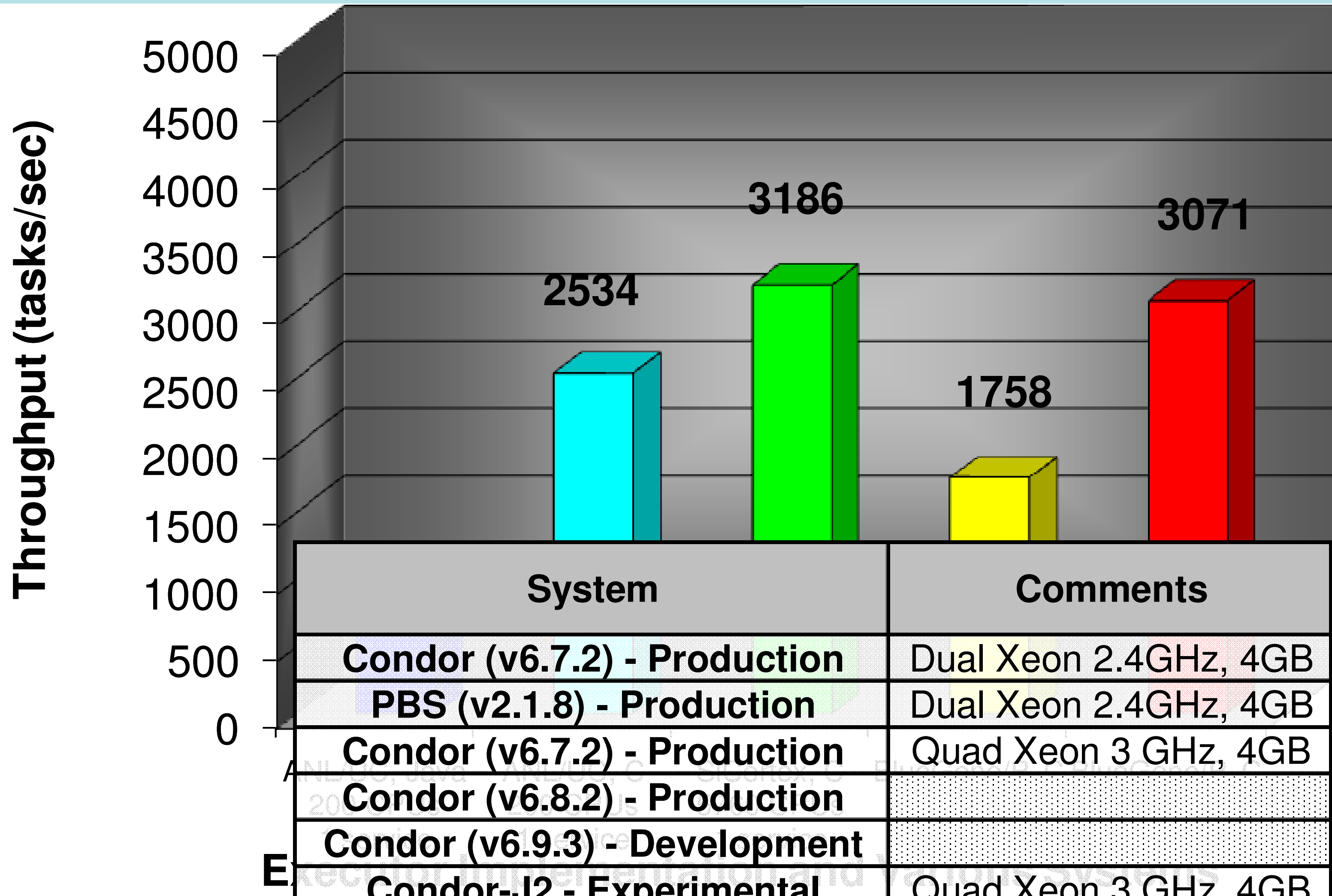
Data Diffusion

- 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"

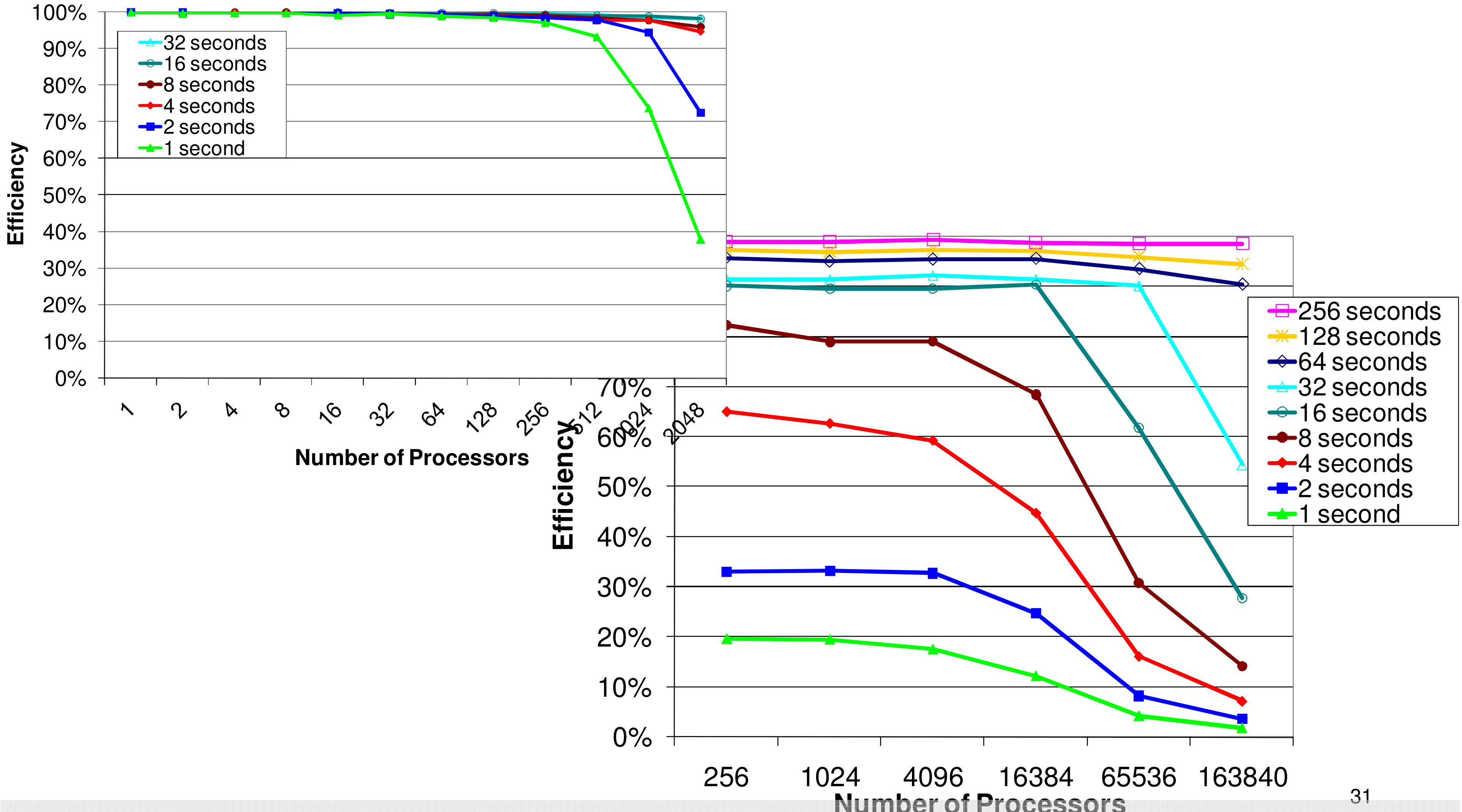
Dispatch Throughput



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

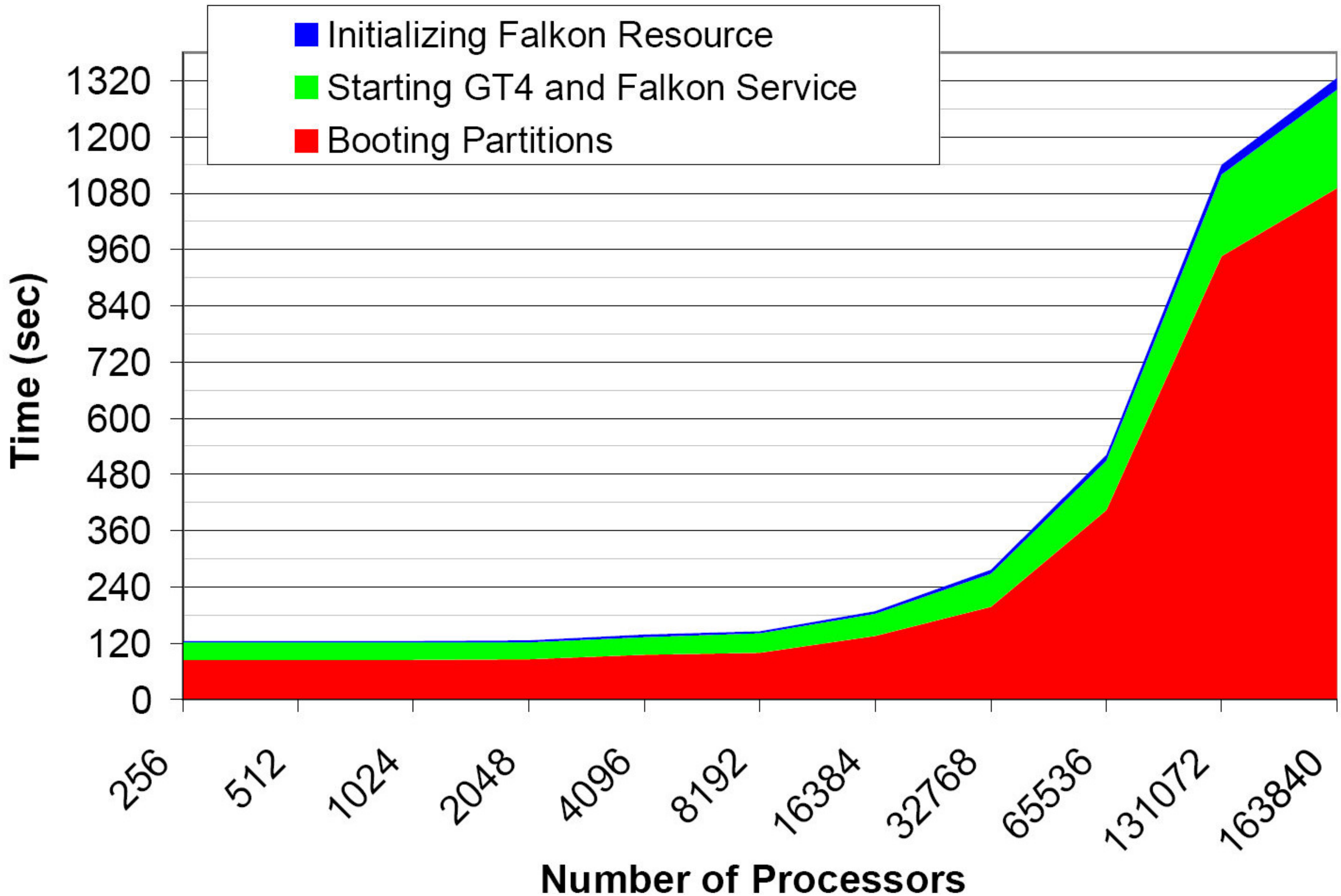
[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

Execution Efficiency

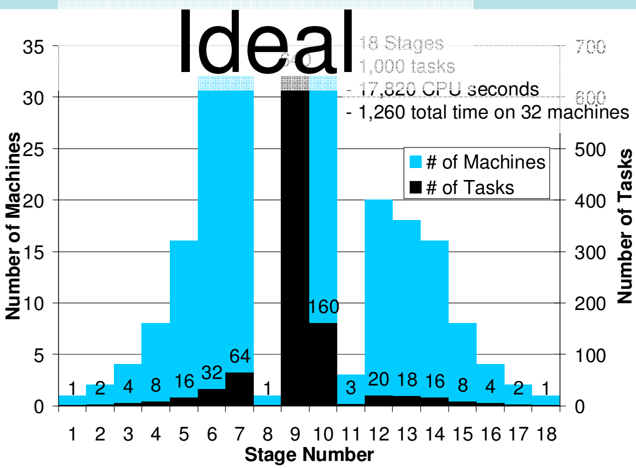


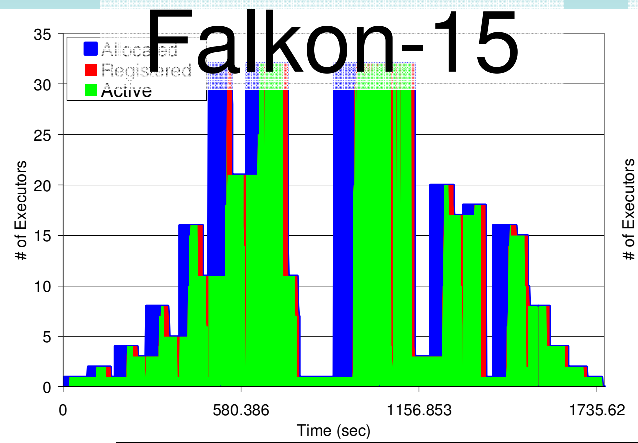
[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

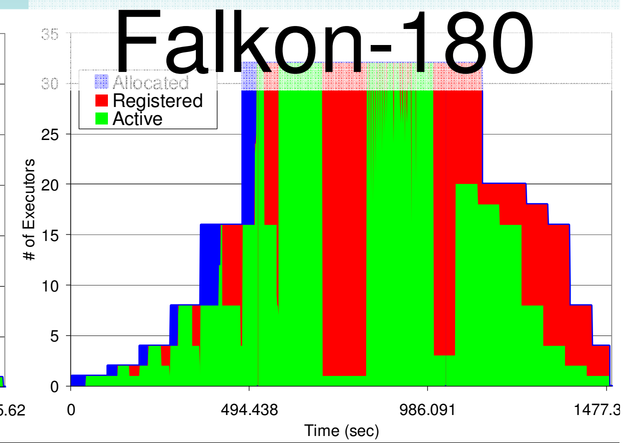
Resource Provisioning Overheads IBM Blue Gene/P



Dynamic Resource Provisioning







- End-to-end execution time:
  - 1260 sec in ideal case
  - 4904 sec → 1276 sec
- Average task queue time:
  - 42.2 sec in ideal case
  - 611 sec → 43.5 sec
- Trade-off:
  - Resource Utilization for Execution Efficiency

	GRAM +PBS	Falkon-15	Falkon-60	Falkon-120	Falkon-180	Falkon-∞	Ideal (32 nodes)
Queue Time (sec)	611.1	87.3	83.9	74.7	44.4	43.5	42.2
Execution Time (sec)	56.5	17.9	17.9	17.9	17.9	17.9	17.8
Execution Time %	8.5%	17.0%	17.6%	19.3%	28.7%	29.2%	29.7%

	GRAM +PBS	Falkon-15	Falkon-60	Falkon-120	Falkon-180	Falkon-∞	Ideal (32 nodes)
Time to complete (sec)	**4904**	**1754**	**1680**	**1507**	**1484**	**1276**	**1260**
Resouce Utilization	30%	89%	75%	65%	59%	44%	100%
Execution Efficiency	26%	72%	75%	84%	85%	99%	100%
Resource Allocations	1000	11	9	7	6	0	0

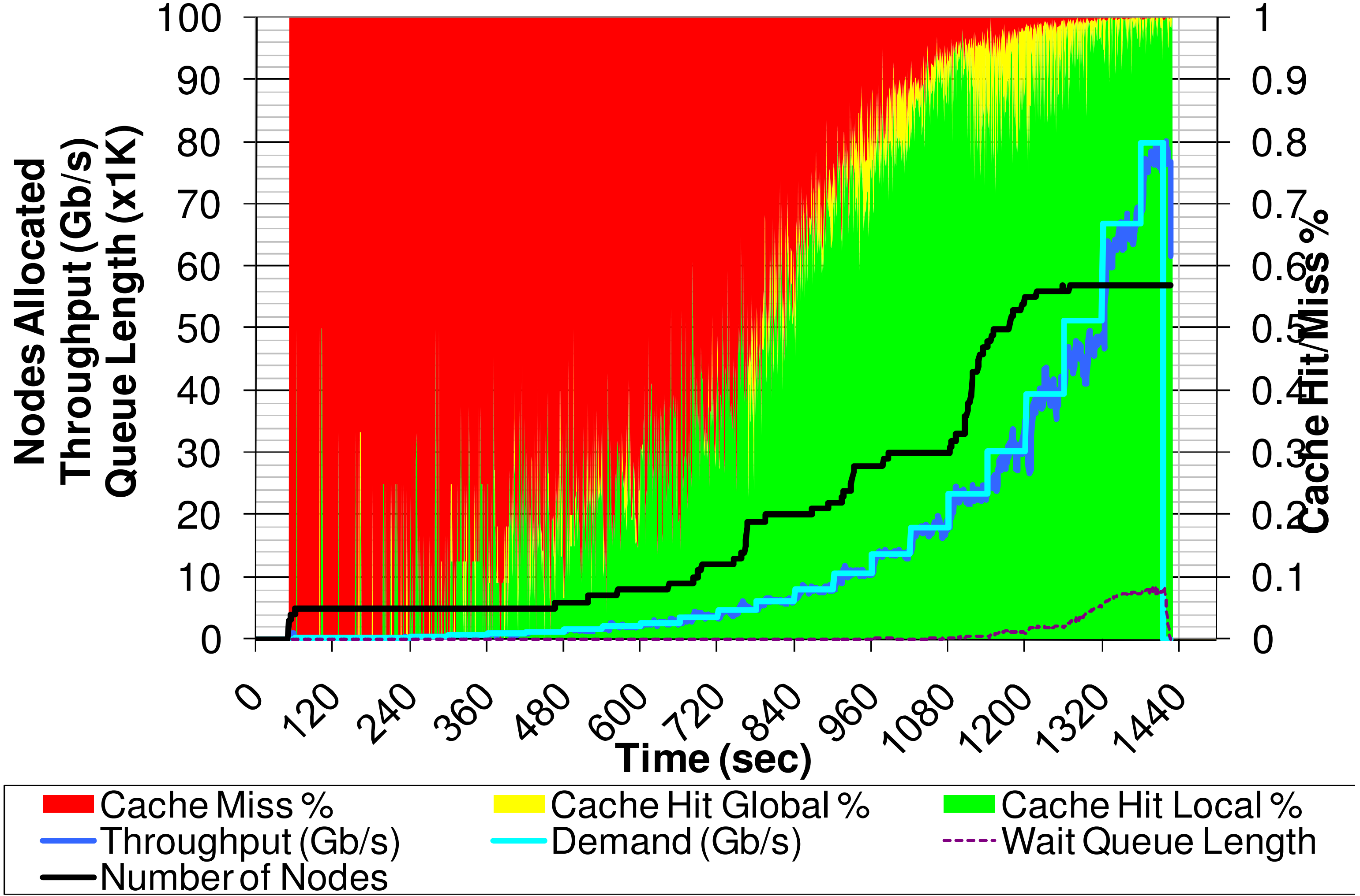
[SC07] "Falkon: a Fast and Light-weight tasK executiON framework"

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"

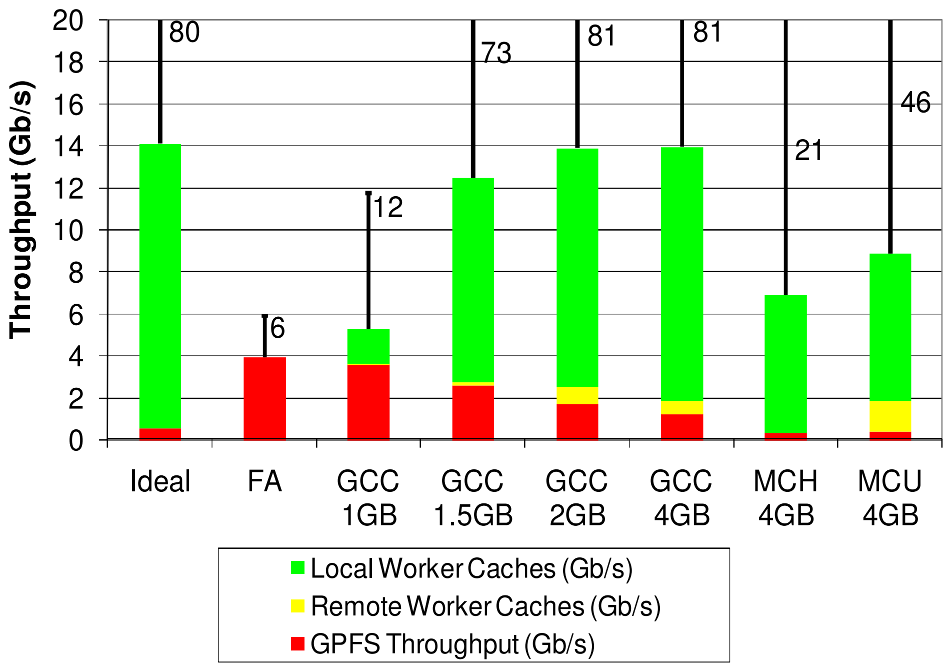
Data Diffusion
Monotonically Increasing Workload



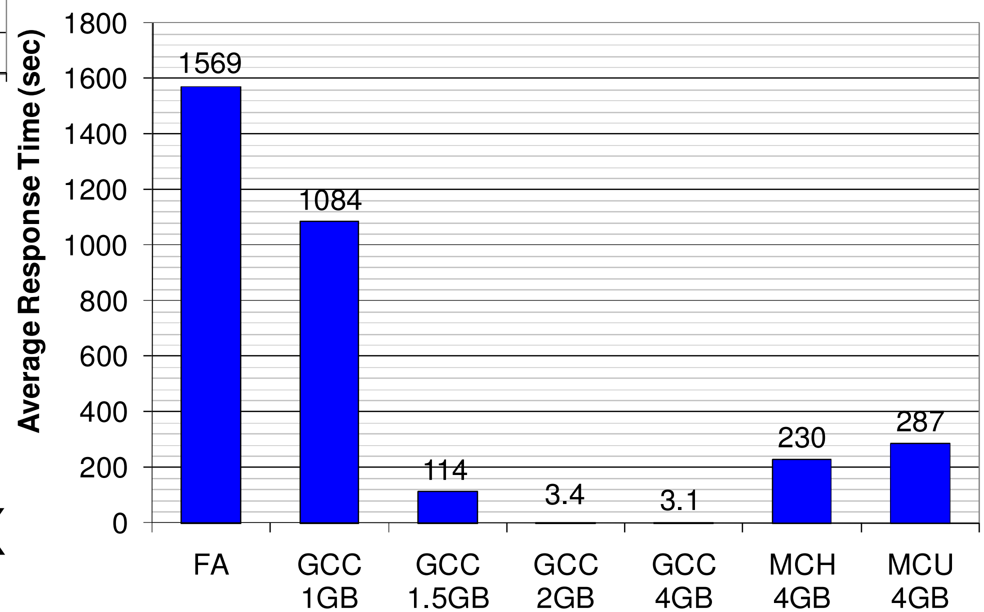
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems"

[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 →

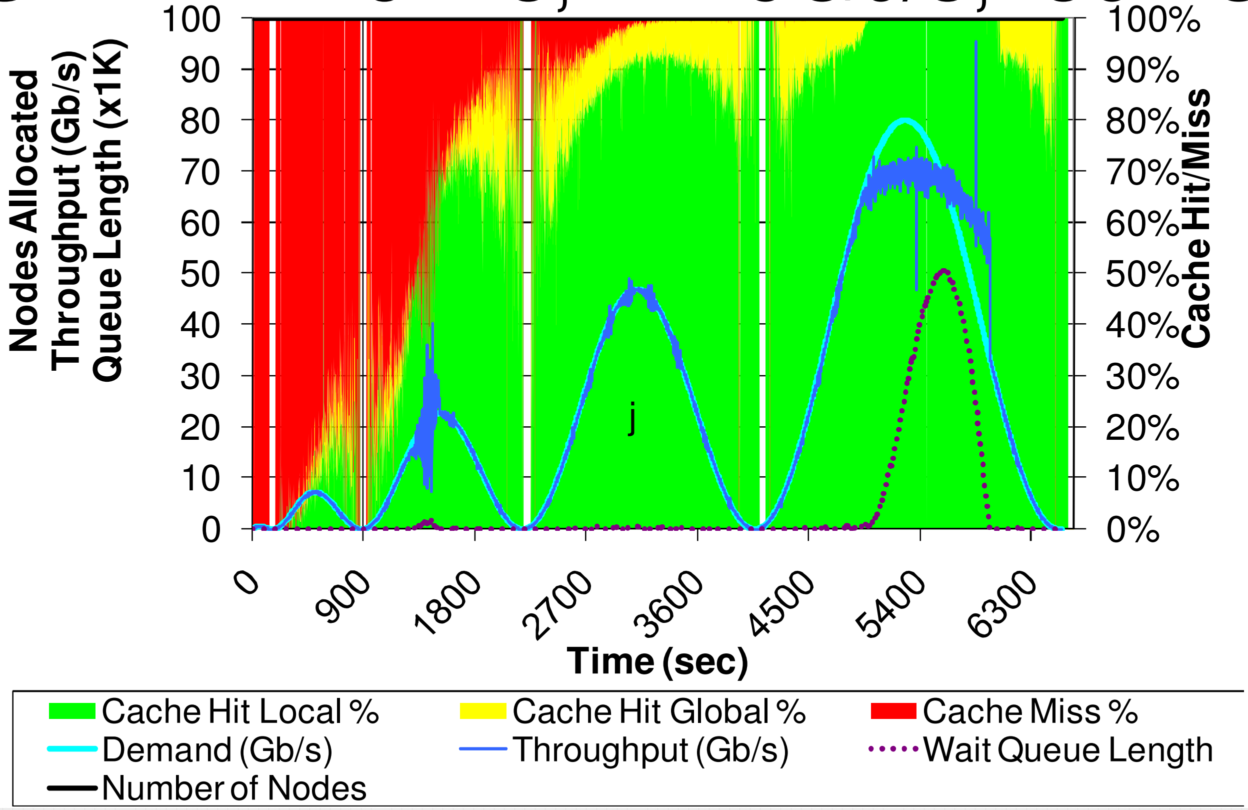
- 3 sec vs 1569 sec → 506X

[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

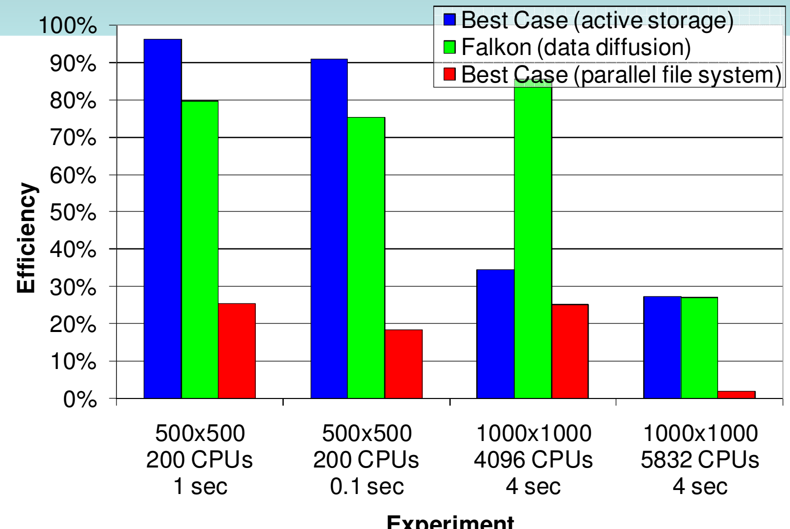


[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

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

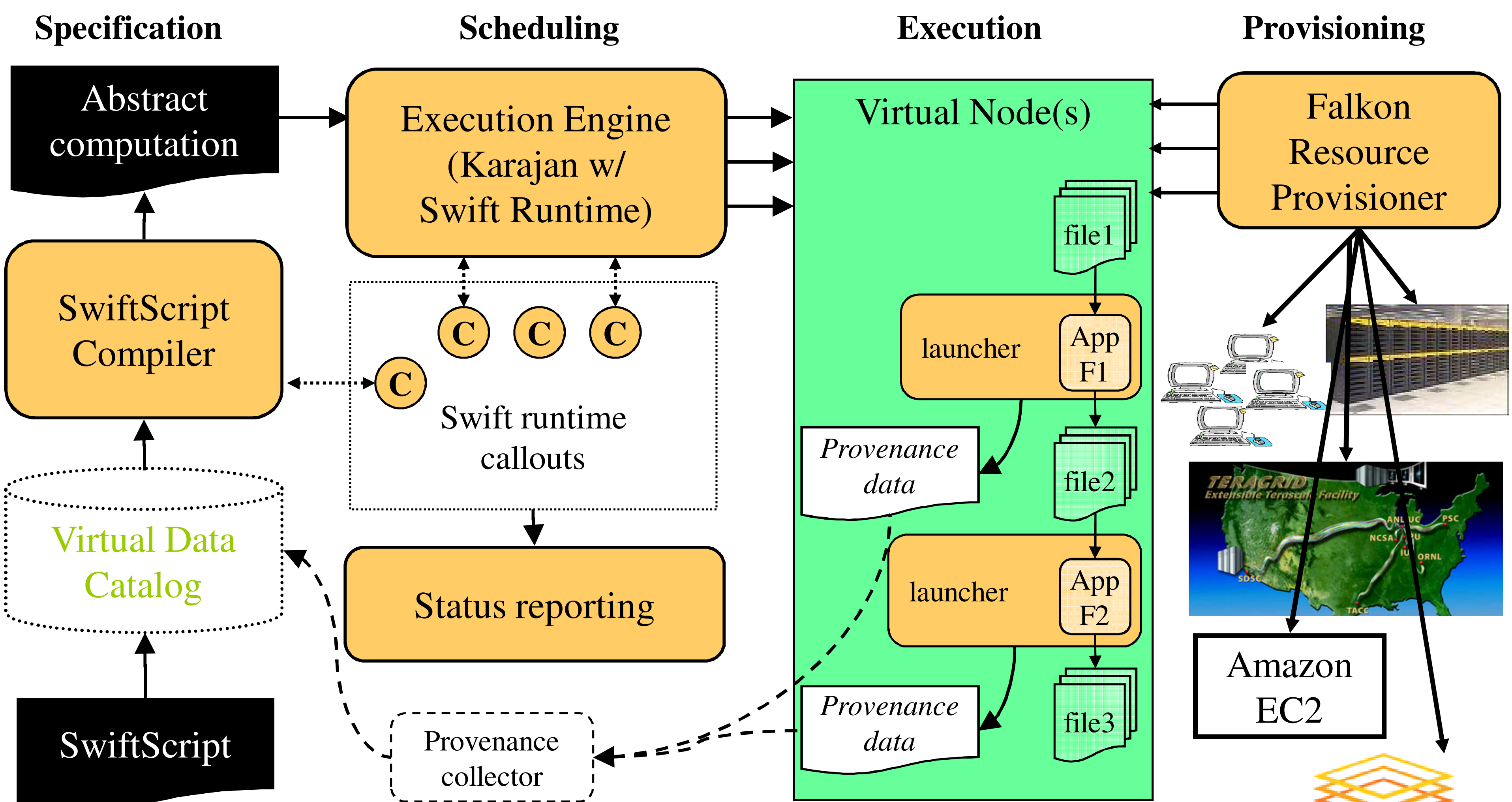
Christopher Moretti, Douglas Thain, University of Notre Dame



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

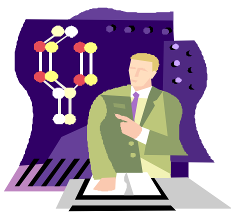
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

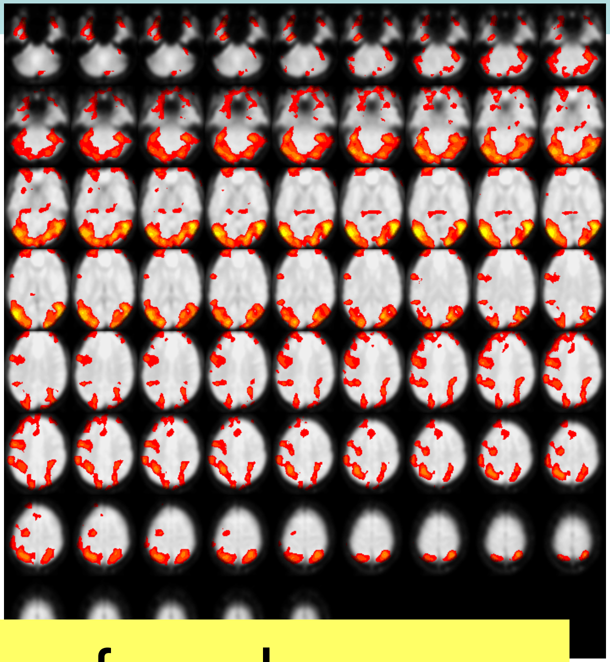
Applications
Swift Architecture

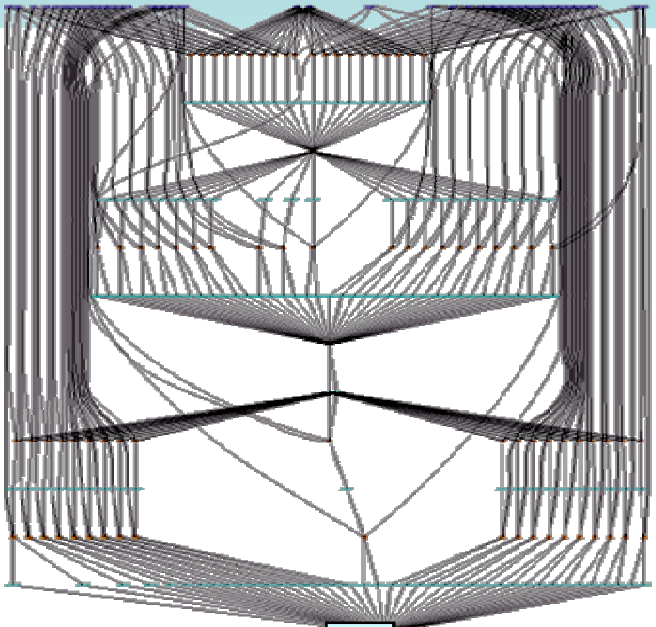


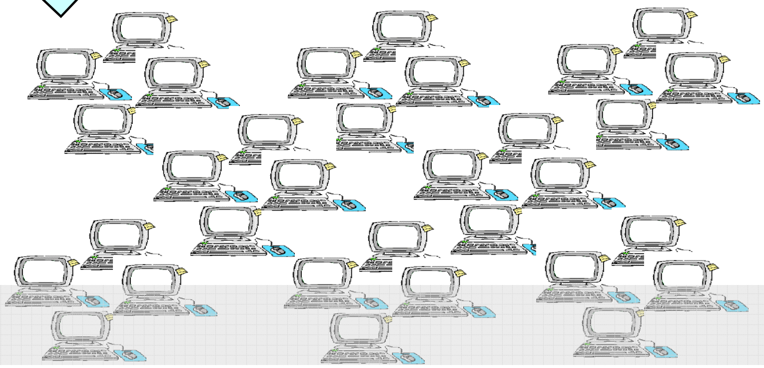
[NOVA08] "Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments"
[SWF07] "Swift: Fast, Reliable, Loosely Coupled Parallel Computation"

Applications
Medical Imaging: fMRI







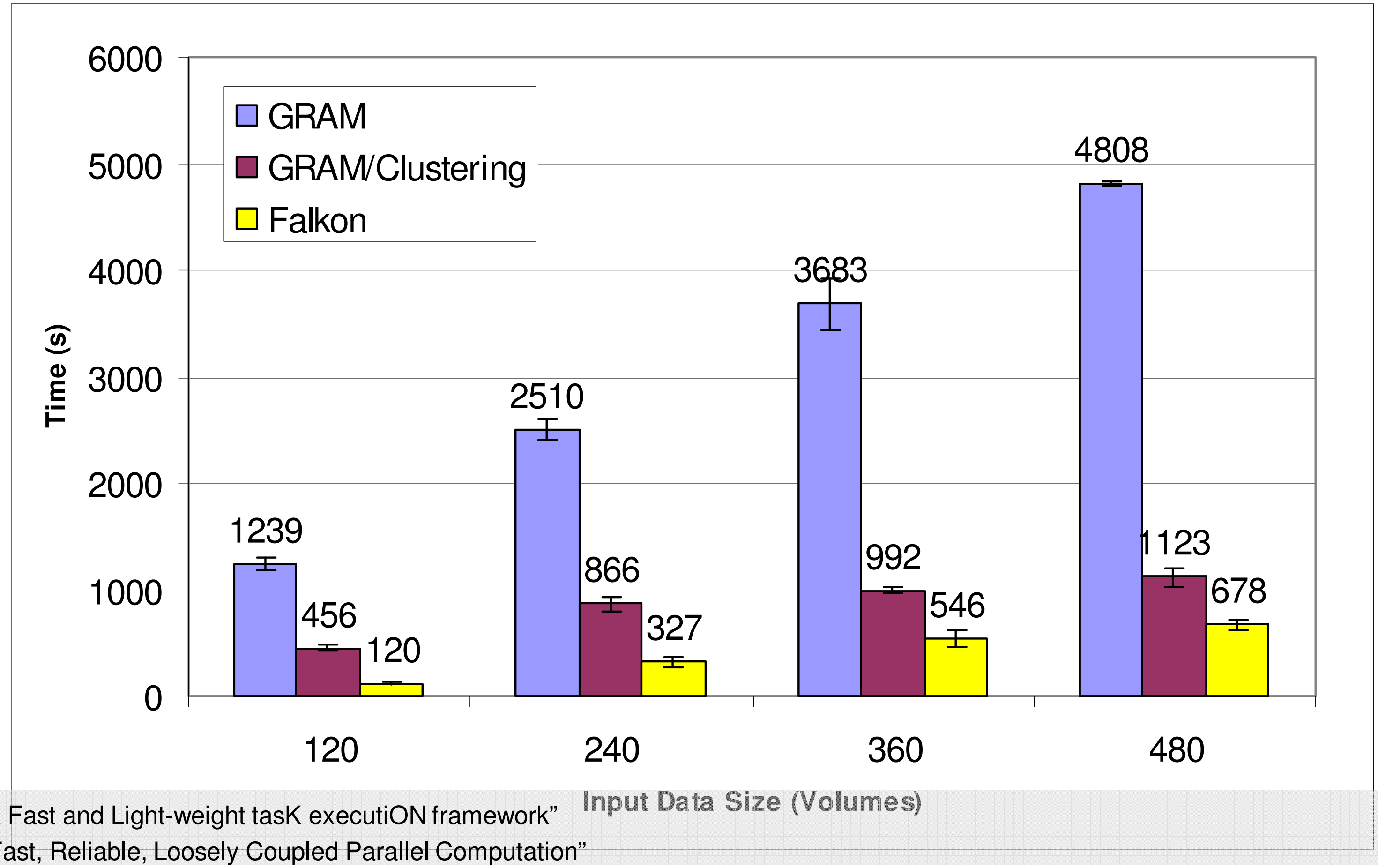


- Wide range of analyses
  - Testing, interactive analysis, production runs
  - Data mining
  - Parameter studies

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

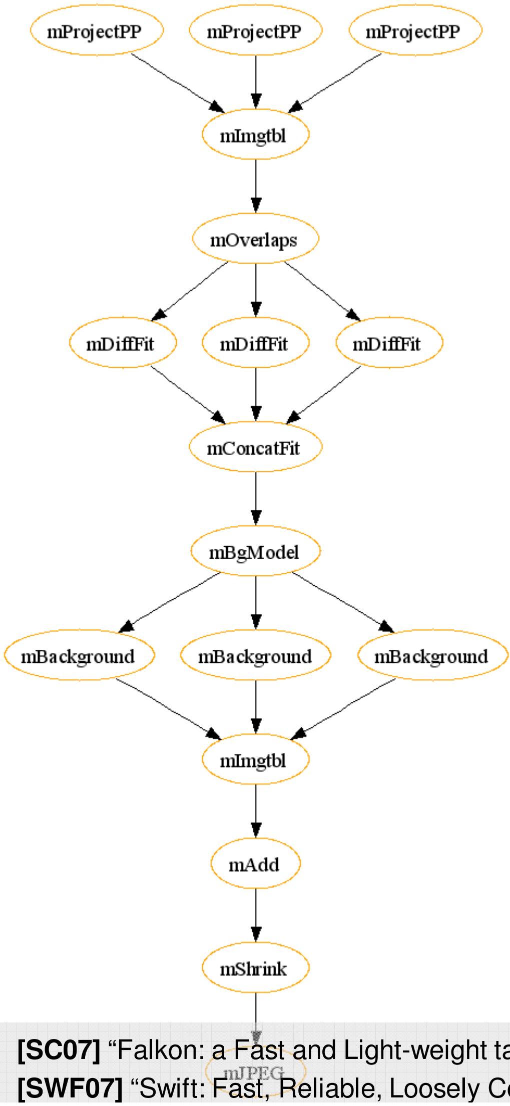
Applications
Medical Imaging: fMRI

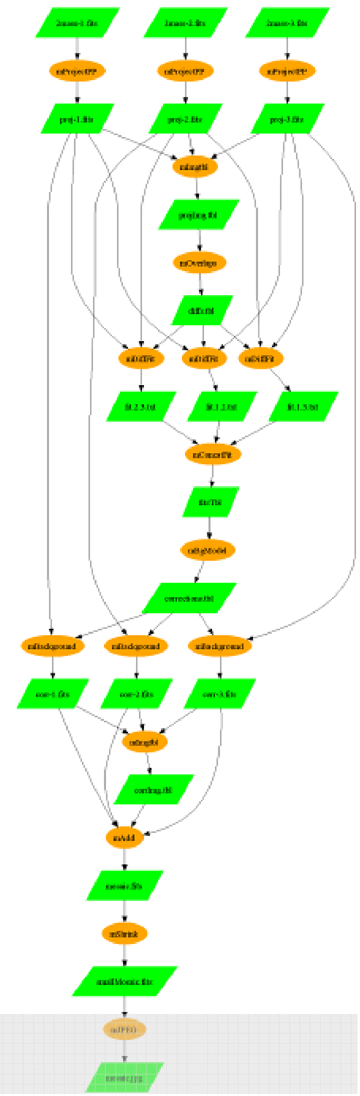
- GRAM vs. Falkon: 85%~90% lower run time
- GRAM/Clustering vs. Falkon: 40%~74% lower run time



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

Applications
Astronomy: Montage



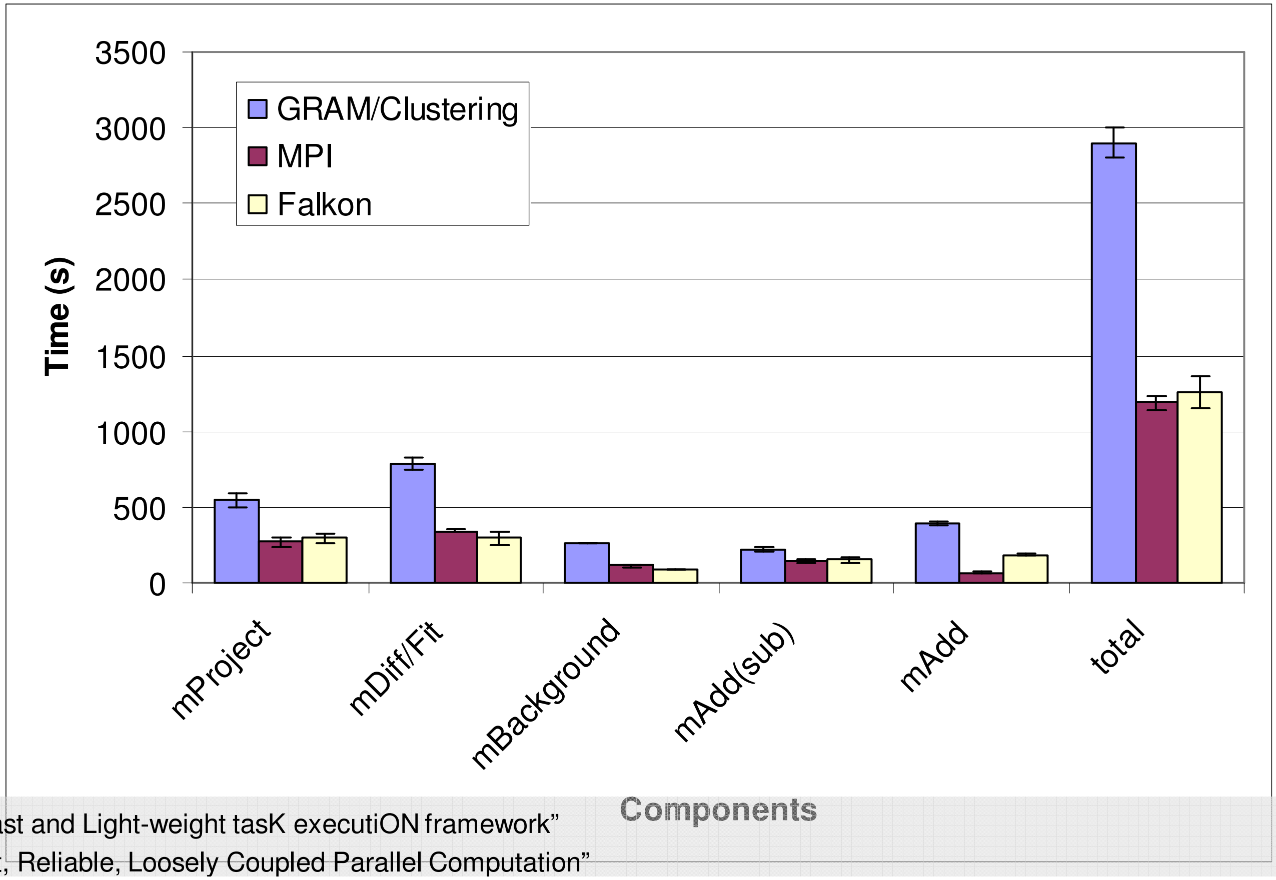


B. Berriman, J. Good (Caltech)
J. Jacob, D. Katz (JPL)

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

Applications
Astronomy: Montage

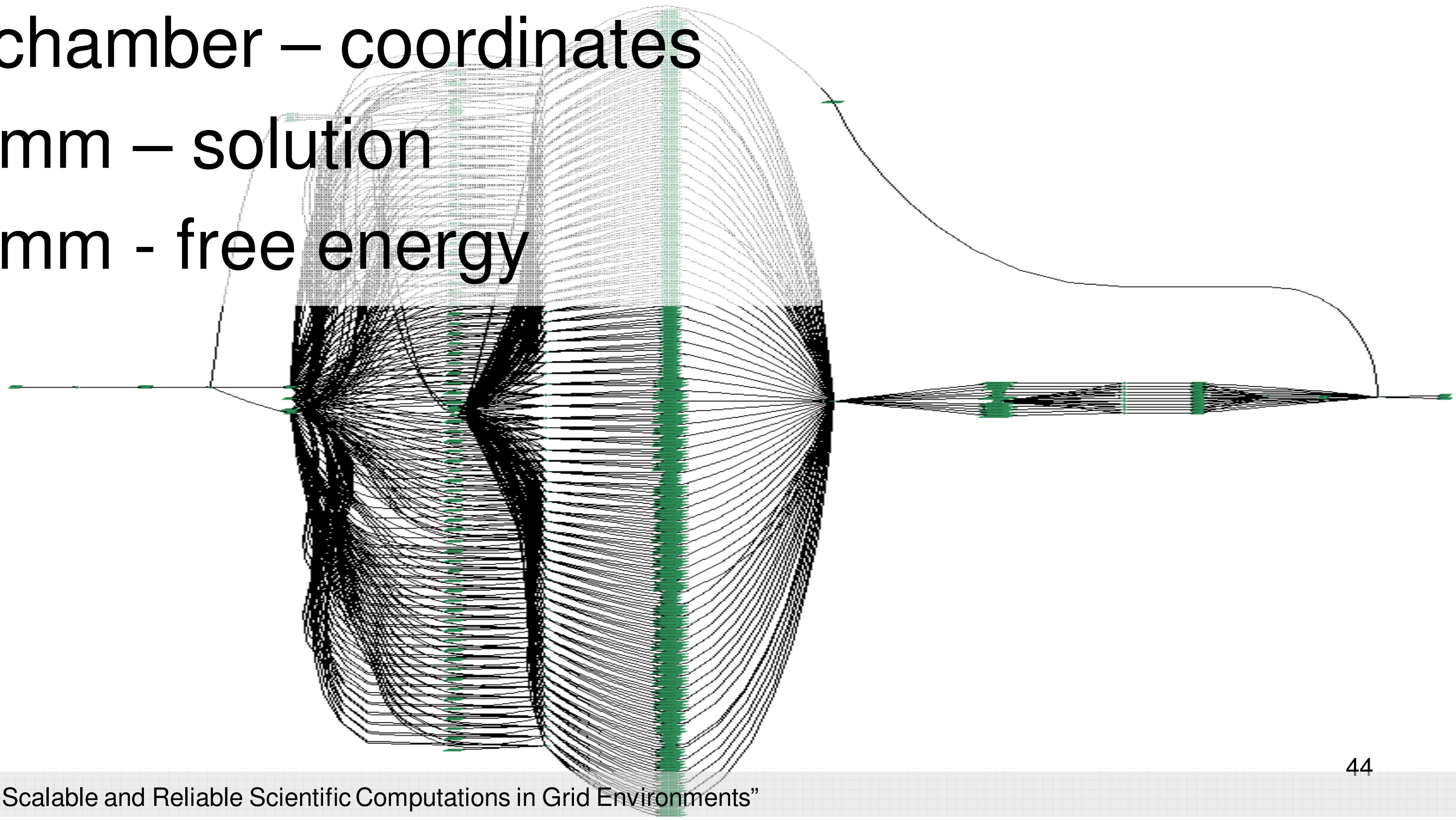
- GRAM/Clustering vs. Falkon: 57% lower application run time
- MPI* vs. Falkon: 4% higher application run time
- * MPI should be lower bound



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

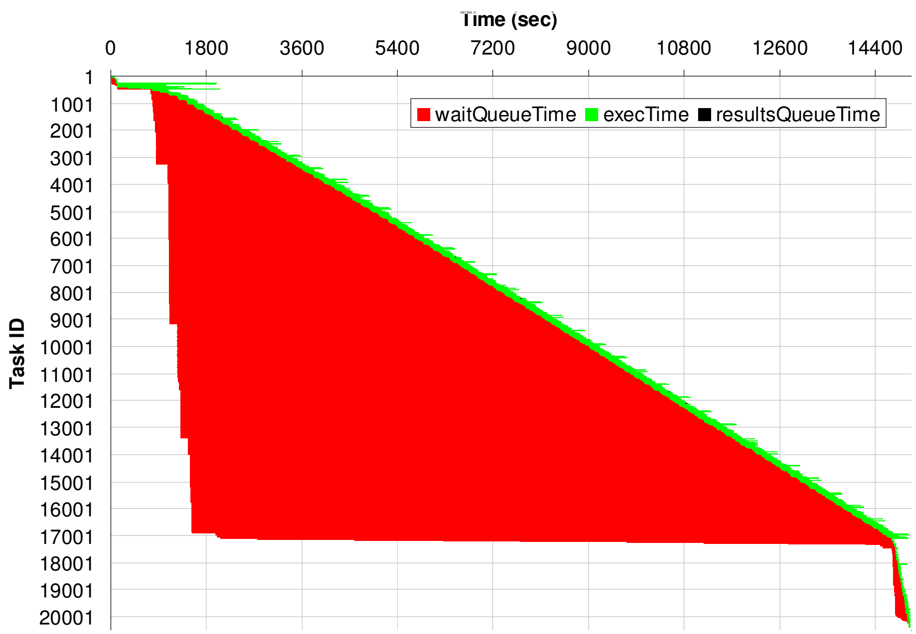
Applications
Molecular Dynamics: MolDyn

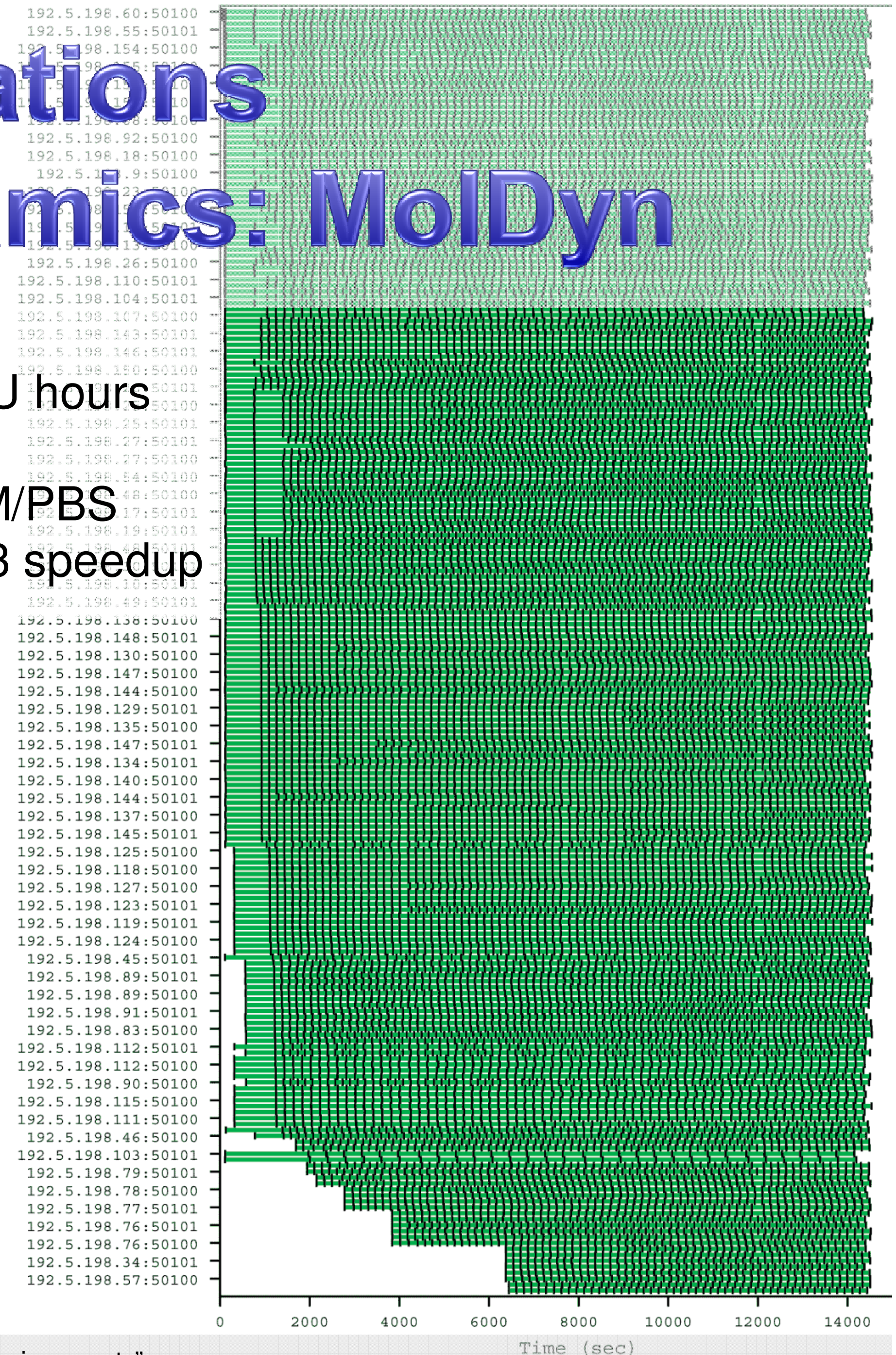
- Determination of free energies in aqueous solution
  - Antechamber – coordinates
  - Charmm – solution
  - Charmm - free energy



Applications
Molecular Dynamics: MolDyn

- 244 molecules → 20497 jobs
- 15091 seconds on 216 CPUs → 867.1 CPU hours
- Efficiency: 99.8%
- Speedup: 206.9x → 8.2x faster than GRAM/PBS
- 50 molecules w/ GRAM (4201 jobs) → 25.3 speedup

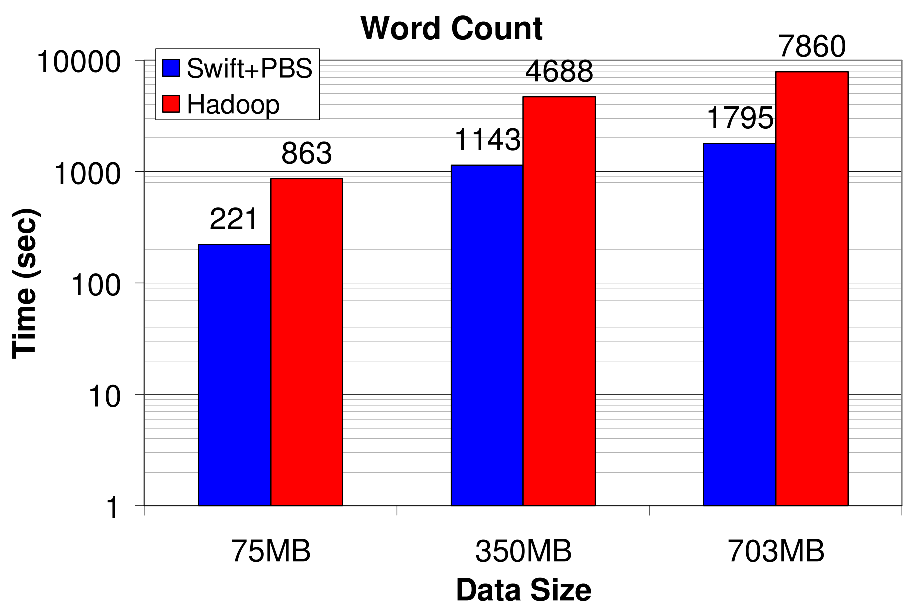


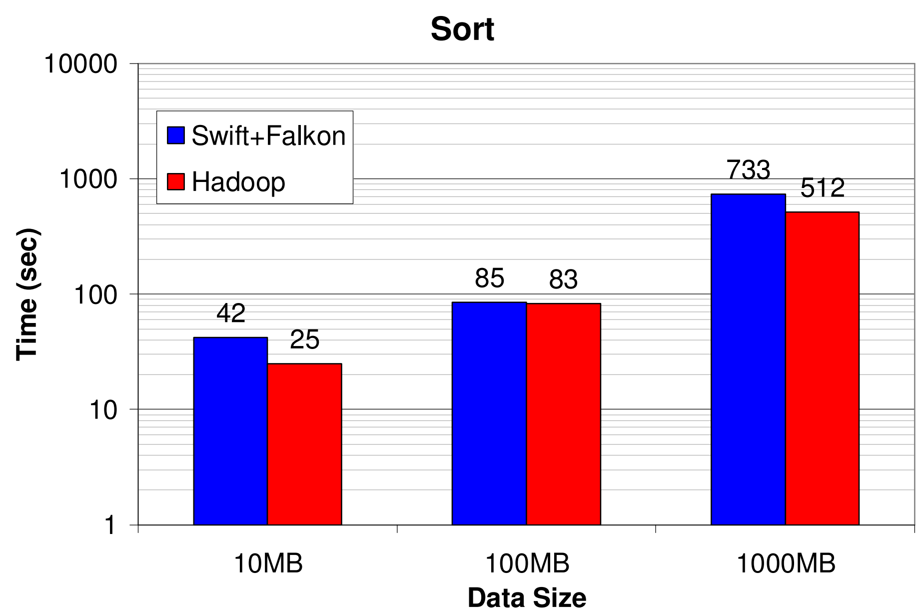


[NOVA08] "Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments"

Applications
Word Count and Sort

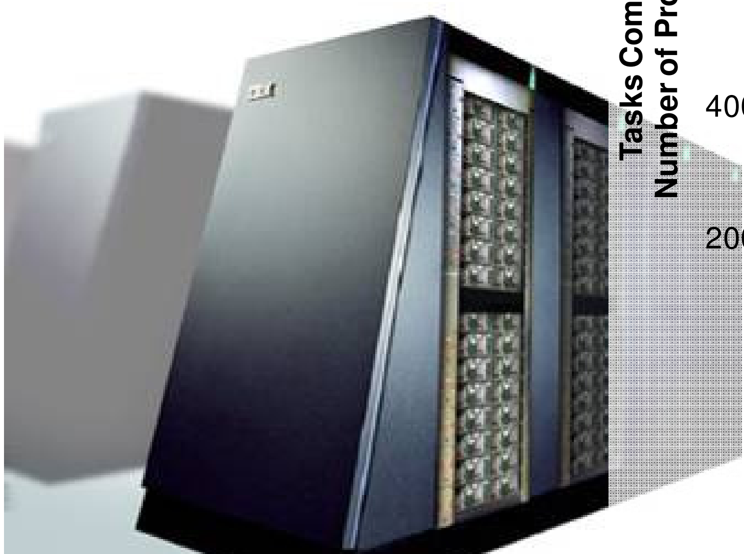
- Classic benchmarks for MapReduce
  - Word Count
  - Sort
- Swift and Falkon performs similar or better than Hadoop (on 32 processors)

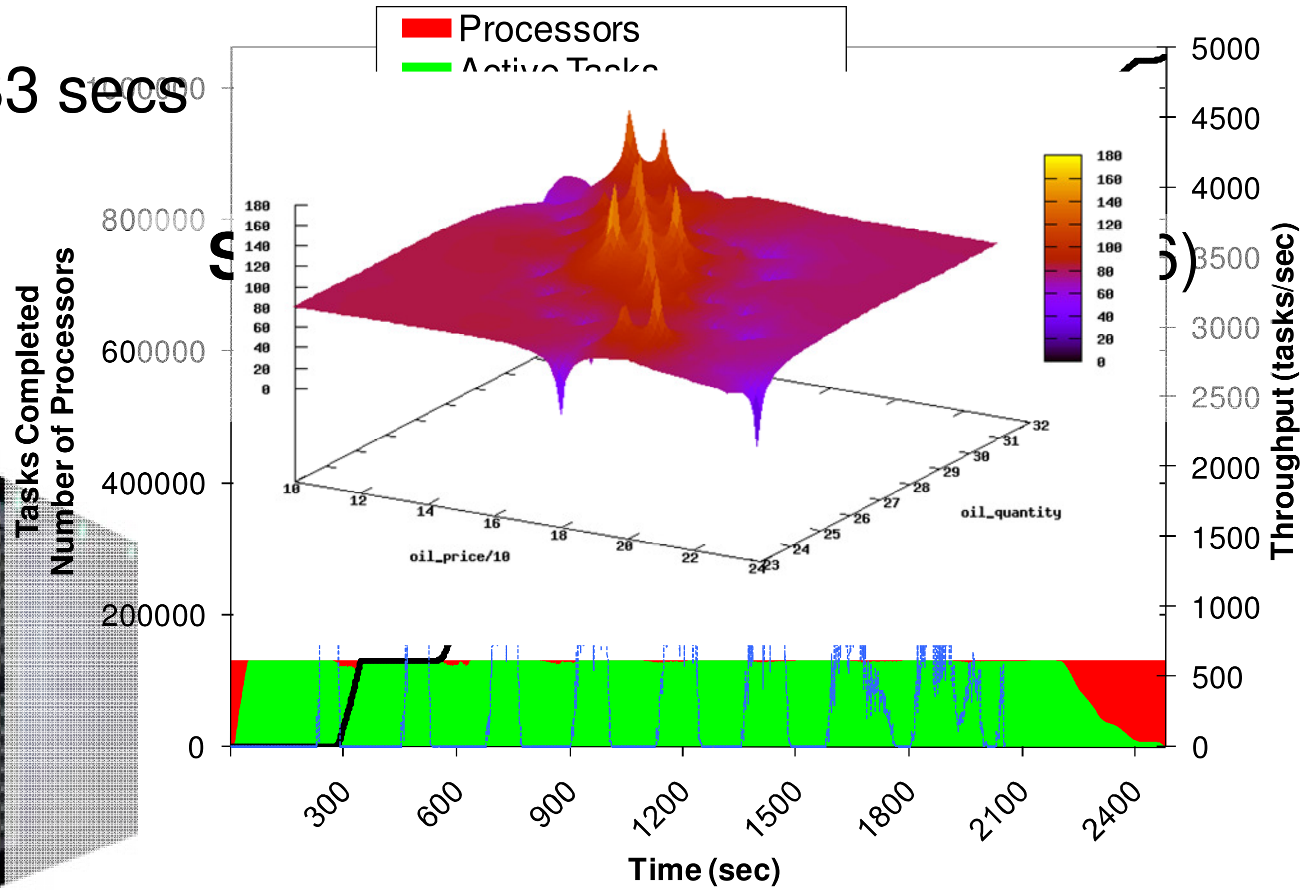




Applications
Economic Modeling: MARS

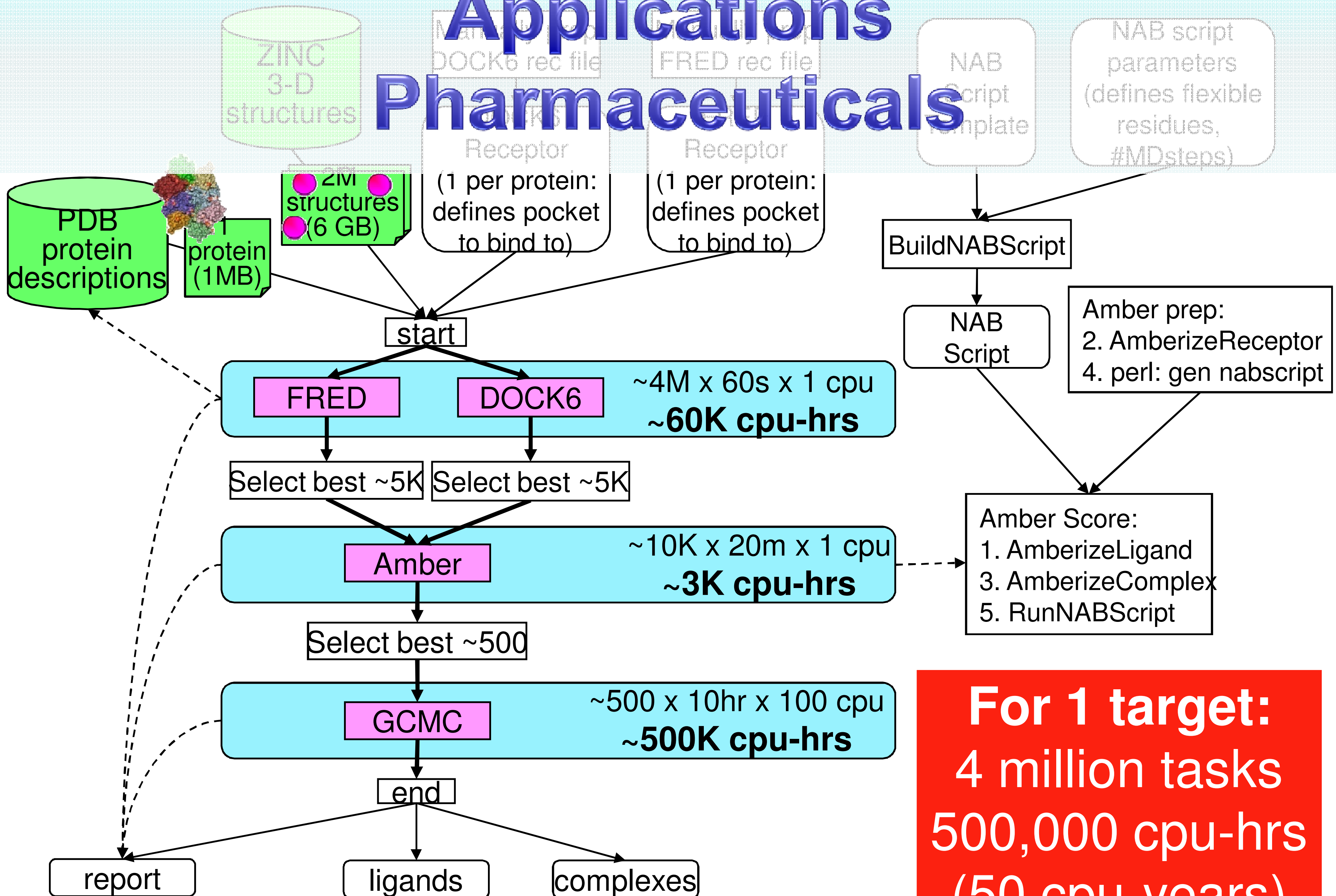
- CPU Cores: 130816
- Tasks: 1048576
- Elapsed time: 2483 secs
- CPU Years: 9.3





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

Applications
Pharmaceuticals



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

Applications
Pharmaceuticals: DOCK

CPU cores: 118784

Tasks: 934803

Elapsed time: 2.01 hours

Compute time: 21.43 CPU years

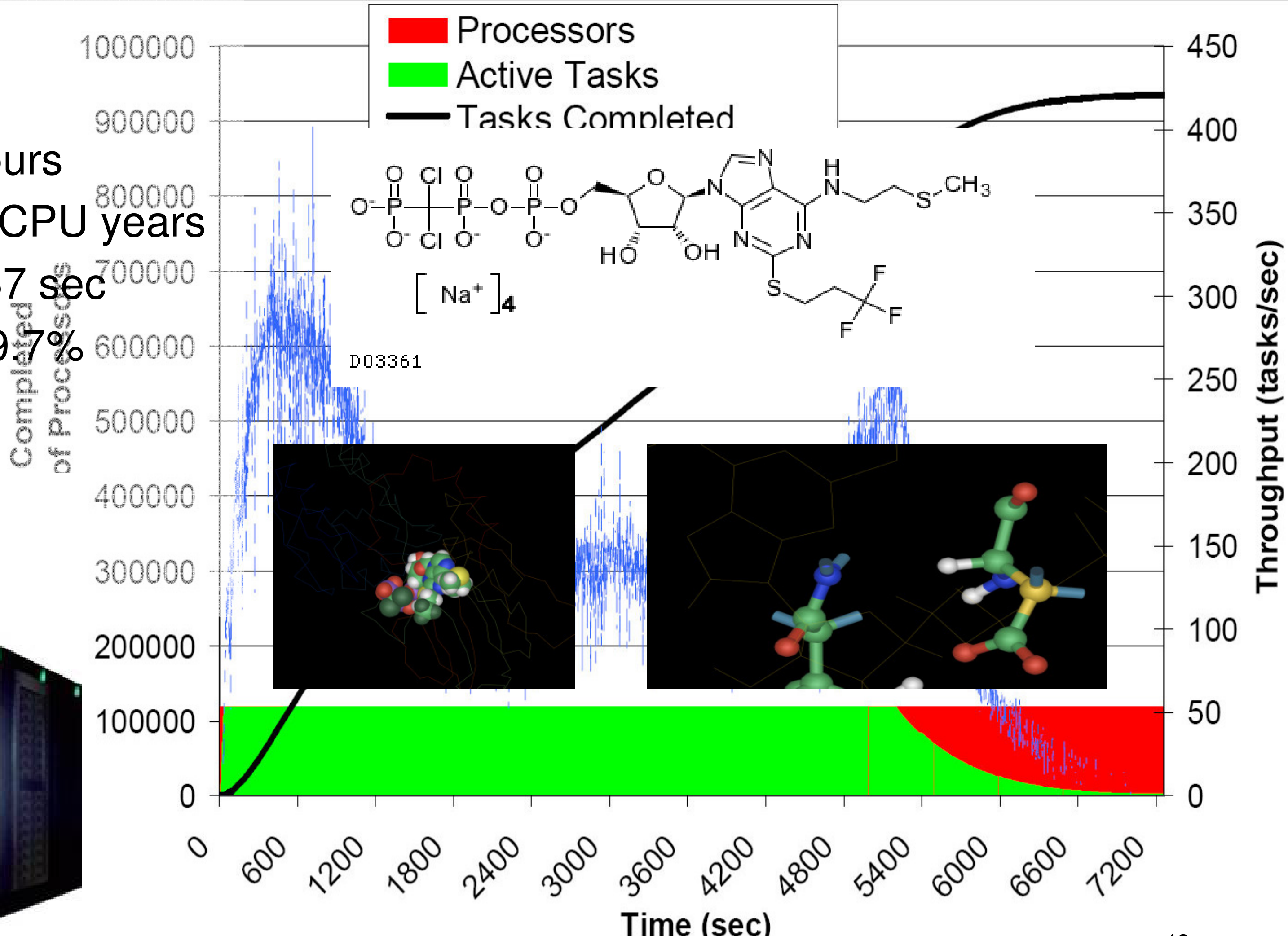
Average task time: 667 sec

Relative Efficiency: 99.7%

(from 16 to 32 racks)

Utilization:

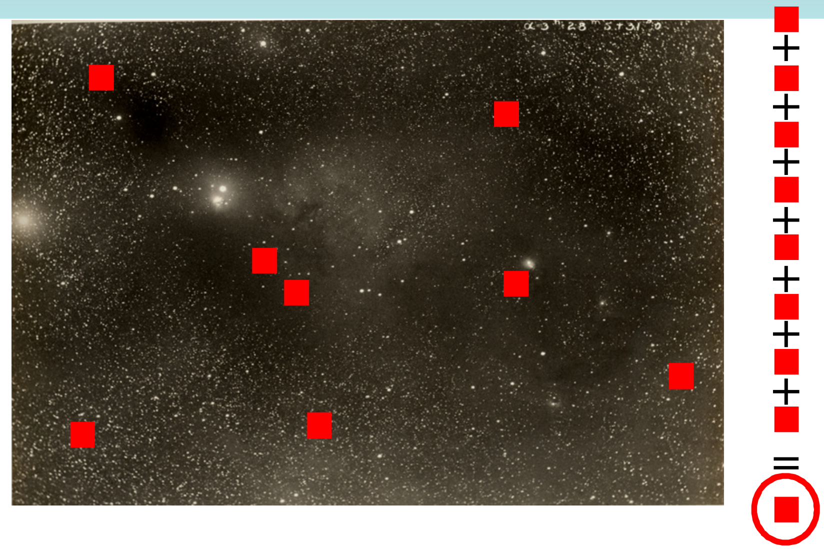
- Sustained: 99.6%
- Overall: 78.3%

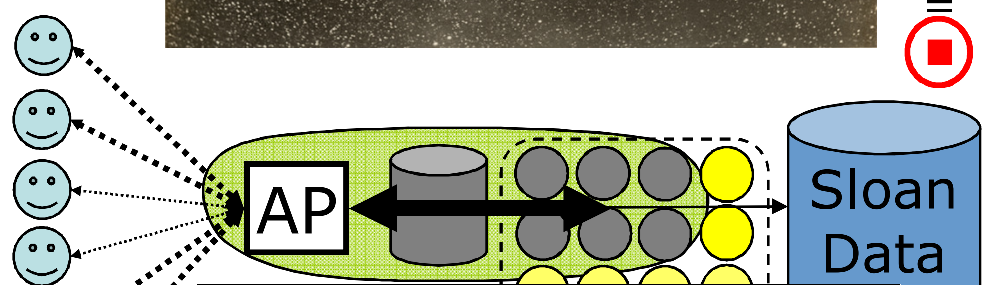


[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

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



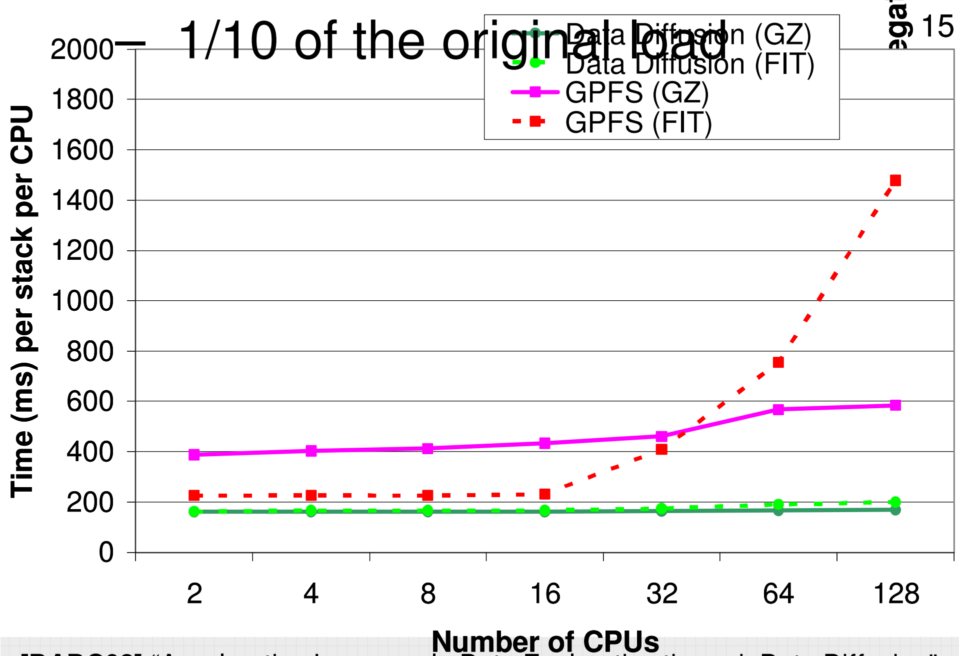


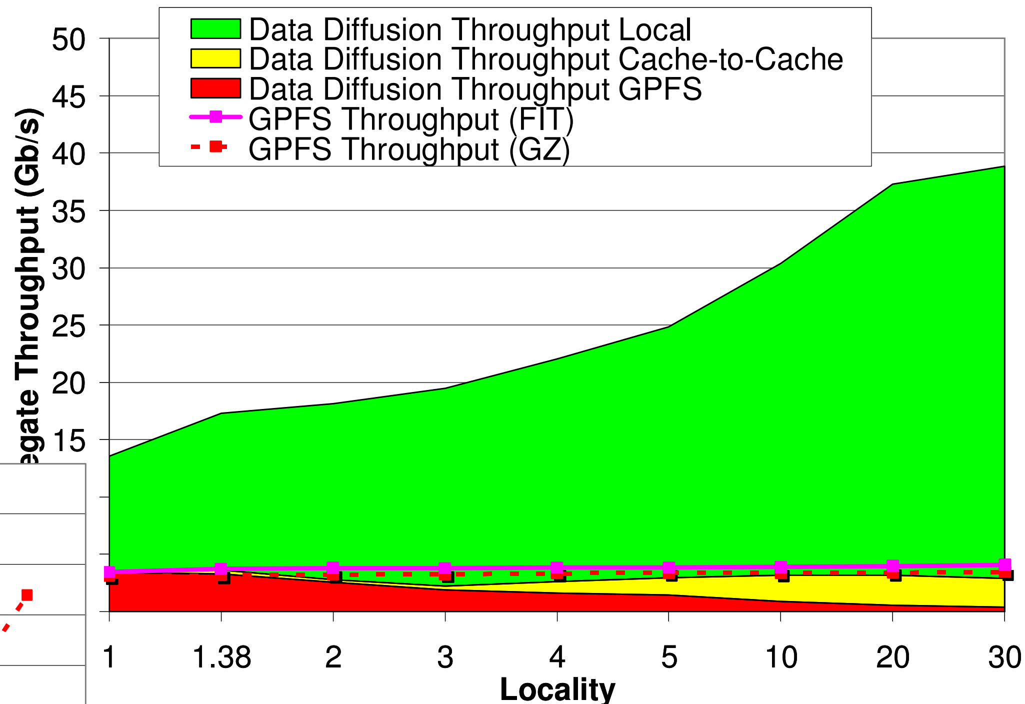
Locality	Number of Objects	Number of Files
1	111700	111700
1.38	154345	111699
2	97999	49000
3	88857	29620
4	76575	19145
5	60590	12120
10	46480	4650
20	40460	2025
30	23695	790

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

Applications
Astronomy: AstroPortal

- Aggregate throughput:
  - 39Gb/s
  - 10X higher than GPFS
- Reduced load on GPFS
  - 0.49Gb/s
  - 1/10 of the original load





- ←High data locality
  - Near perfect scalability

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

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

Contributions

- Identified that data locality is crucial to the efficient use of large scale distributed systems for data-intensive applications ➔ Data Diffusion
  - Integrated streamlined task dispatching with data aware scheduling policies
  - Heuristics to maximize real world performance
  - Suitable for varying, data-intensive workloads
  - Proof of O(NM) Competitive Caching

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

Summary

- My publications directly related to MTC
  - 27 articles and proposals
  - 40+ formal presentations
  - 250+ citations
- Activities for broader community engagement
  - IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS) 2008, co-located with SC08
  - MegaJob08 BOF at SC08
  - ACM MTAGS09, co-located with SC09
  - IEEE Transactions on Parallel and Distributed Systems (TPDS), Special Issue on Many-Task Computing, November 2010
- Courses
  - "Big Data" at University of Chicago (Ian Foster)
  - "Data-Intensive Computing" at Northwestern Univ. (Ioan Raicu)

Summary (cont)

- Open source project
  - Falkon Incubator Project with Globus
  - System wide installs on a variety of large systems
  - Dozens of users, 100s of millions of jobs, millions of CPU hours
- Other people's work
  - 2 PhD students at University of Chicago
  - Multiple grant proposals to NSF
- New Science
  - Astronomy: faint and transient object discovery
  - Pharmaceuticals: drug screening and discovery
  - Chemistry: predicting protein structure and recognizing docking partners
  - Economic modeling: study economic model sensitivities
  - Other domains: Astrophysics, bioinformatics, neuroscience, cognitive neuroscience, data analytics, data mining, biometrics

Future Work
Understanding the current limitations

- Falkon
  - Needs Java (not portable to the largest supercomputers)
  - Needs IP connectivity (an issue in the largest systems)
  - Naïve decentralized scheduler
  - No support for HPC workloads (e.g. MPI applications)
- Data Diffusion
  - Data access patterns: write once, read many
  - Task definition must include input/output files metadata
  - Per task working set must fit in local storage
  - Requires local storage (disk, memory, etc)
  - Centralized data-aware scheduler

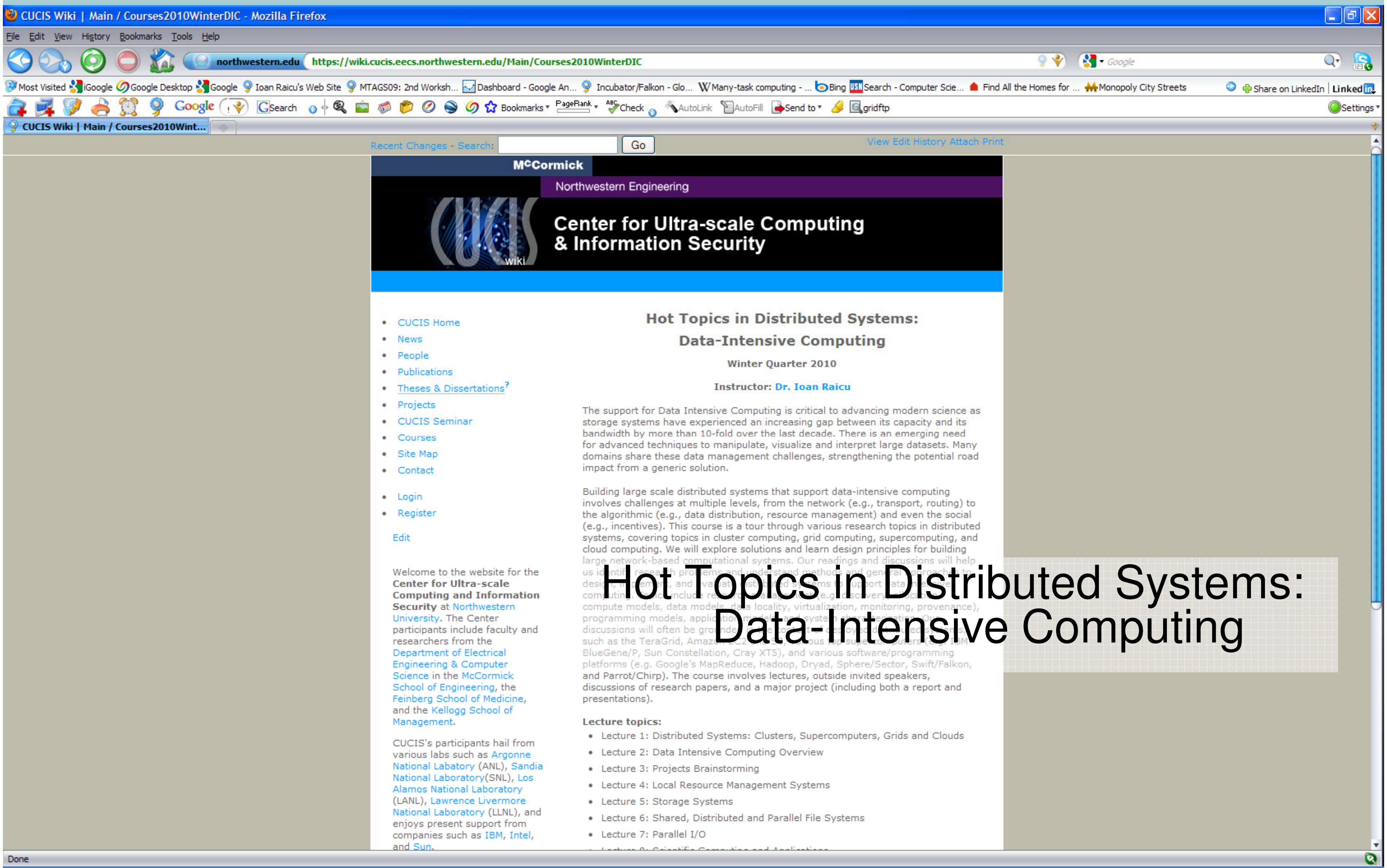
Future Work

- Distributing Falkon architecture
  - Distributed queuing system
  - Distributed metadata management
  - Scalable distributed data-aware scheduling
  - Distributed file storage system
- Interactive HPC
  - Ensemble MPI applications
  - Computational steering
- Computational and I/O Benchmarks
  - Workflow-based benchmarks
  - Characterizing capabilities of I/O systems
  - Application-oriented I/O benchmarks
- Generalizing, transparency, and alternative technologies

Other Ideas

- Cluster Computing on GPUs
- Distributed file/storage systems
- Distributed Operating Systems
- Data-intensive computing in Cloud Computing
- HPC in Cloud Computing
- Parallel programming systems/languages

Course on Data-Intensive Computing



Hot Topics in Distributed Systems:
Data-Intensive Computing

MTAGS09: 2nd Workshop on Many-Task Computing on Grids and Supercomputers - Mozilla Firefox

http://dsl.cs.uchicago.edu/MTAGS09/

MTAGS

2nd Workshop on Many-Task Computing on Grids and Supercomputers

co-located with ACM/IEEE SC09 (International Conference for High Performance, Networking, Storage and Analysis)
Portland, Oregon -- November 16th, 2009

ACM MTAGS09 Workshop @ SC09

Home

Call for Papers (TXT, PDF)

Program Committee

Important Dates

Paper Submission

Venue

Registration

Workshop Program

Important Dates

Abstract Due:	August 1st, 2009
Papers Due:	September 1st, 2009
Notification of Acceptance:	October 1st, 2009
Camera Ready Papers Due:	November 1st, 2009
Workshop Date:	November 16th, 2009

Committee Members

Workshop Chairs

Ioan Raicu, University of Chicago
Ian Foster, University of Chicago & Argonne National Laboratory
Yong Zhao, Microsoft

Technical Committee

* David Abramson, Monash University, Australia
* Pete Beckman, Argonne National Laboratory, USA
* Ian Foster, University of Chicago & Argonne National Laboratory, USA
* Bob Grossman, University of Illinois at Chicago, USA
* Indranil Gupta, University of Illinois at Urbana Champaign, USA
* Alexandru Iosup, Delft University of Technology, Netherlands
* Zhou Lei, Shanghai University, China
* Shiyong Lu, Wayne State University, USA
* Reagan Moore, University of North Carolina at Chapel Hill, USA
* Marlon Pierce, Indiana University, USA
* Ioan Raicu, University of Chicago, USA
* Matei Ripeanu, University of British Columbia, Canada
* Greg Thain, Univeristy of Wisconsin, USA
* Matthew Woitaszek, The University Corporation for Atmospheric Research, USA
* Sherali Zeadally, University of the District of Columbia, USA
* Yong Zhao, Microsoft, USA

IEEE Transactions on Parallel and Distributed Systems: Special Issue on Many-Task Computing - Mozilla Firefox

http://dsl.cs.uchicago.edu/TPDS_MTC/

IEEE Transactions on Parallel and Distributed Systems

Special Issue on Many-Task Computing

Home

Call for Papers (TXT, PDF)

Important Dates

Paper Submission

Abstract Due:	December 1st, 2009
Papers Due:	December 21st, 2009
First Round Decisions:	February 22nd, 2010
Major Revisions if needed:	April 19th, 2010
Second Round Decisions:	May 24th, 2010
Minor Revisions if needed:	June 7th, 2010
Final Decision:	June 21st, 2010
Publication Date:	November, 2010

IEEE TPDS Journal
Special Issue on MTC
Due Date: December 1st, 2009

Special Issue Guest Editors

Ian Foster, University of Chicago & Argonne National Laboratory
Ioan Raicu, University of Chicago
Yong Zhao, Microsoft

Dr. Ian Foster is the Associate Division Director and a Senior Scientist in the Mathematics and Computer Science Division at Argonne Na Distributed Systems Laboratory, and he is an Arthur Holly Compton Professor in the Department of Computer Science at the University o Grid Forum and with the Globus Alliance as an open source strategist. In 2006, he was appointed director of the Computation Institute, a and Argonne. An earlier project, Strand, received the British Computer Society Award for technical innovation. His research resulted in the algorithms for high-performance distributed computing and parallel computing. As a result he is denoted as "the father of the Grid". Fost the I-WAY wide-area distributed computing experiment, which connected supercomputers, databases and other high-end resources at 1 labs, the Distributed Systems Laboratory is the nexus of the multi-institute Globus Project, a research and development effort that encou advances necessary for engineering, business and other fields. Furthermore the Computation Institute addresses many of the most cha problems facing Grid implementations today. In 2004, he founded Univa Corporation, which was merged with United Devices in 2007 ar honors include the Lovelace Medal of the British Computer Society, the Gordon Bell Prize for high-performance computing (2001), as we American Association for the Advancement of Science in 2003. Dr. Foster also serves as PI or Co-PI on projects connected to the DOE g Computational Science Alliance, the NASA Information Power Grid project, the NSF Grid Physics Network, GRIDS Center, and Internation other DOE and NSF programs. His research is supported by DOE, NSF, NASA, Microsoft, and IBM.

Dr. Ioan Raicu holds a Ph.D. in Computer Science from University of Chicago under the guidance of Dr. Ian Foster. He is a 3-year award win Research Center. His research work and interests are in the general area of distributed systems. His dissertation work focused on this rela (MTC), which aims to bridge the gap between two predominant paradigms from distributed systems, High-Throughput Computing (HTC) an for the last five years focused on defining and exploring both the theory and practical aspects of realizing MTC across a wide range of large-s interested in efficient task dispatch and execution systems, resource provisioning, data management, scheduling, and performance evaluat funded by the NASA Ames Research Center Graduate Student Research Program, as well as the DOE Office of Advanced Scientific Comput resource management in large scale distributed systems with a focus on many-task computing, data intensive computing, cloud computing

Done

More Information

- More information: http://people.cs.uchicago.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, …

- Funding:
  - **NASA**:
    - Ames Research Center, Graduate Student Research Program
    - Jerry C. Yan, NASA GSRP Research Advisor
  - **DOE**:
    - Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy
  - **NSF**:
    - TeraGrid
    - CRA/NSF Computation Innovation Fellow