Scalable Resource Management in Cloud Computing, Grid Computing and Supercomputing

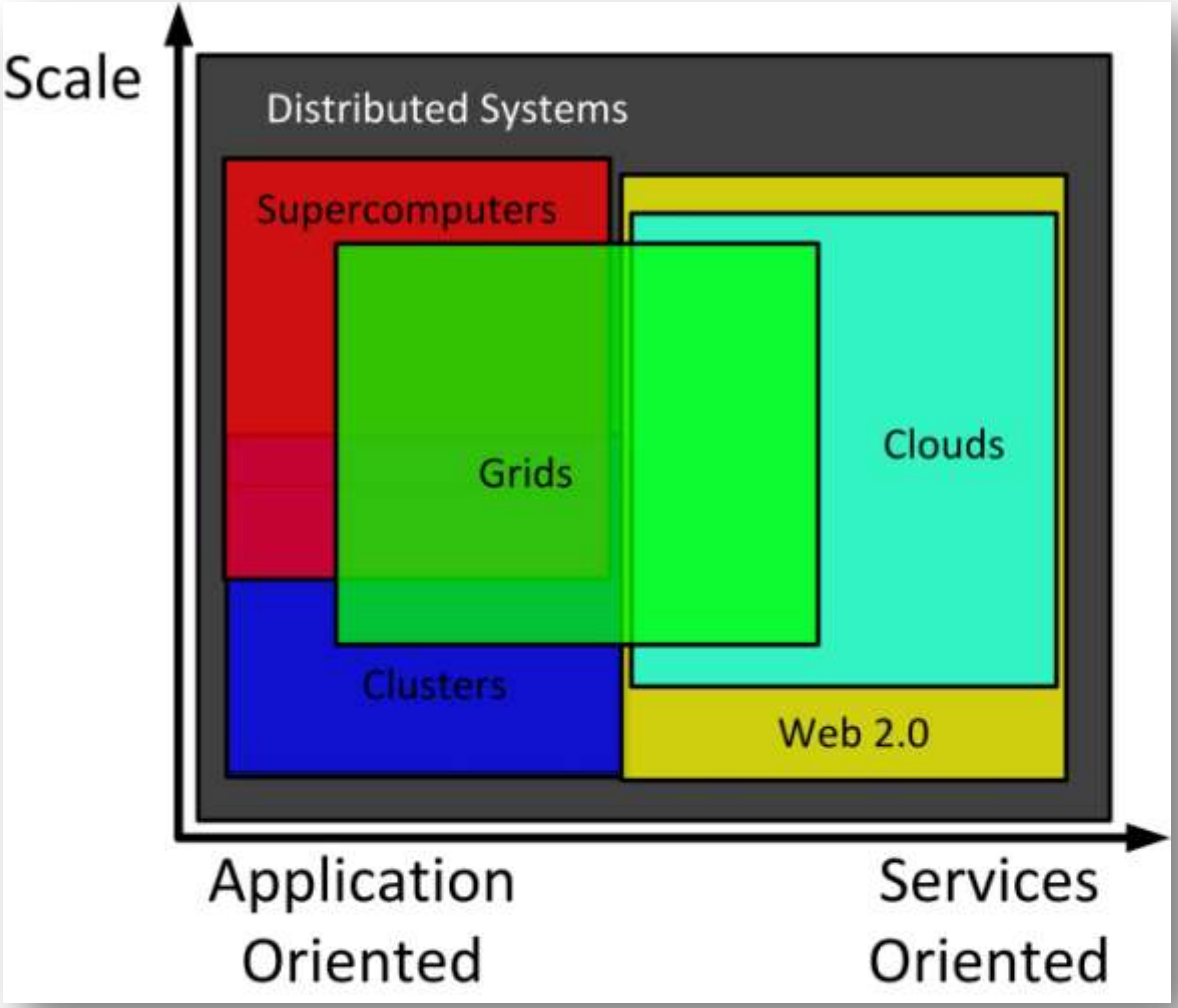
Ioan Raicu
Center for Ultra-scale Computing and Information Security
Department of Electrical Engineering & Computer Science
Northwestern University

College of Computing and Digital Media, DePaul University
January 20th, 2010

Outline

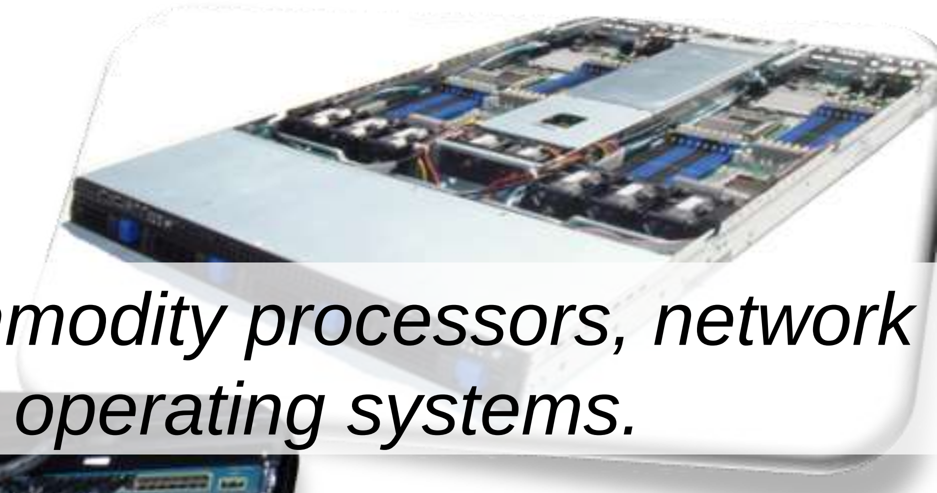
- **Overview**
- Contributions
- Applications
- Conclusions

Clusters, Grids, Clouds, and Supercomputers



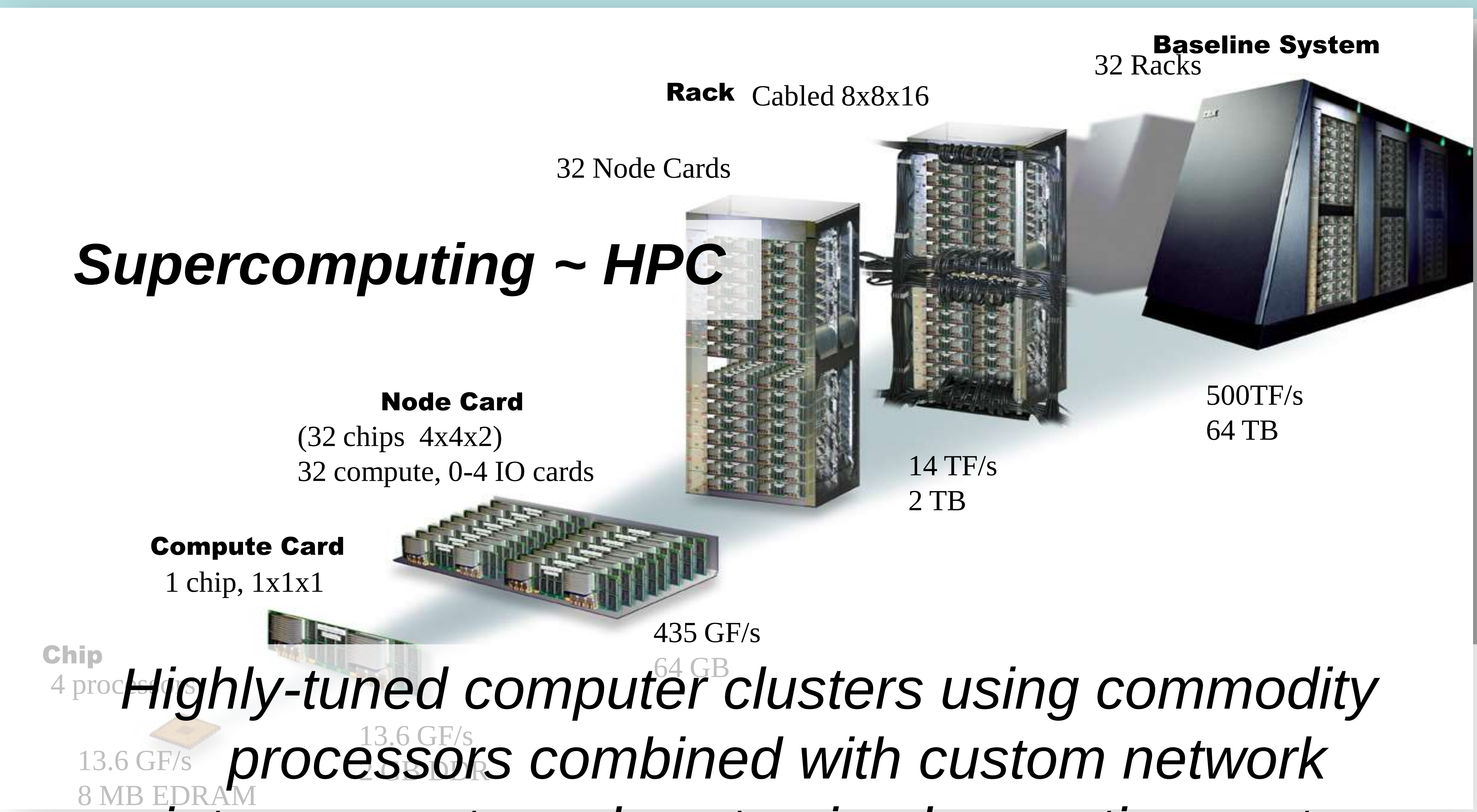
[GCE08] "Cloud Computing and Grid Computing 360-Degree Compared"

Cluster Computing



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

Supercomputing



Supercomputing ~ HPC

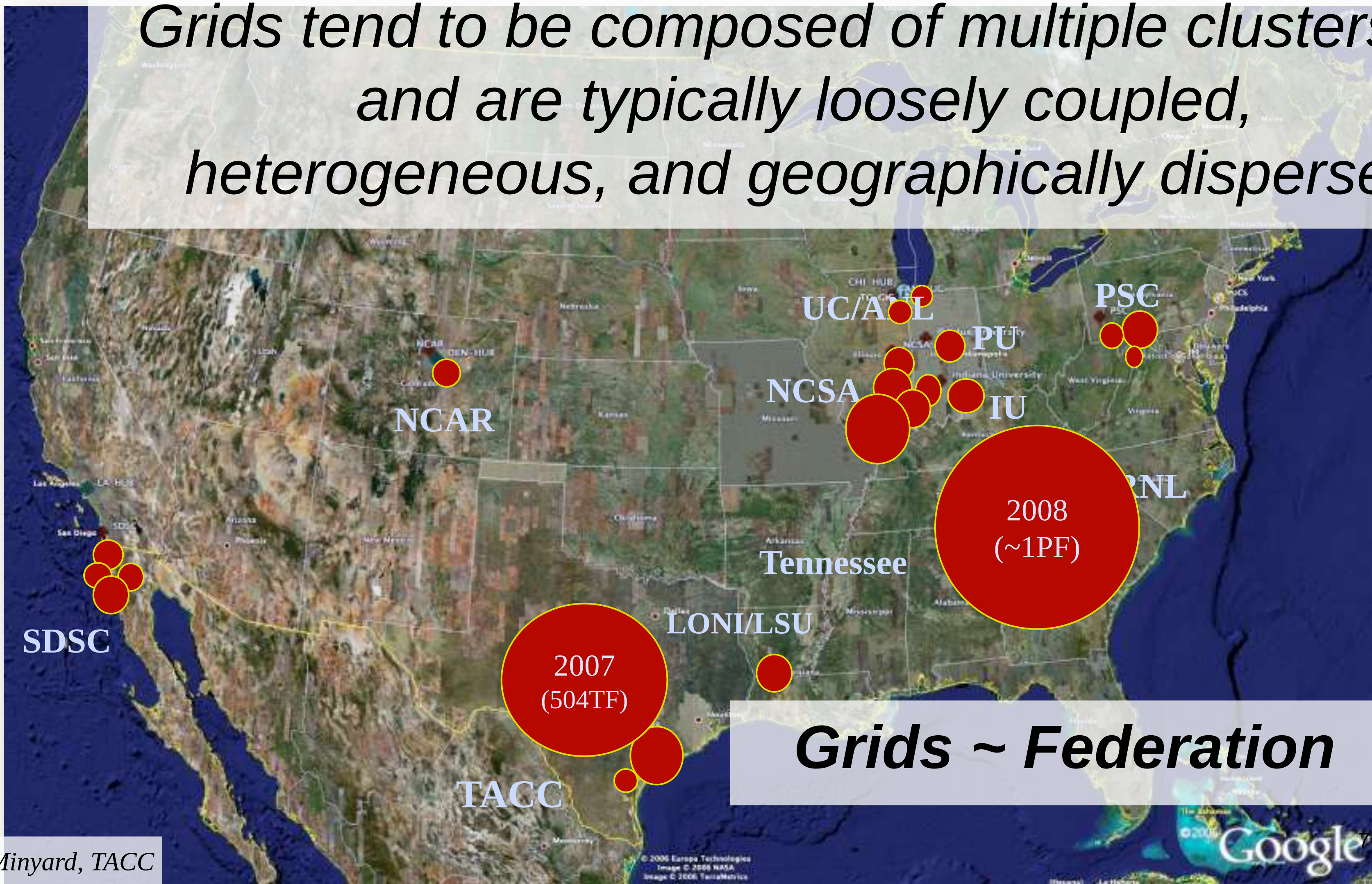
Highly-tuned computer clusters using commodity processors combined with custom network interconnects and customized operating system

Top 10 Supercomputers from Top500

- Cray XT4 & XT5
  - Jaguar #1
  - Kraken #3
- IBM BladeCenter Hybrid
  - Roadrunner #2
- IBM BlueGene/L & BlueGene/P
  - Jugene #4
  - Intrepid #8
  - BG/L #7
- NUDT (GPU based)
  - Tianhe-1 #5
- SGI Altix ICE
  - Plaiedas #6
- Sun Constellation
  - Ranger #9
  - Red Sky #10

Grid Computing

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

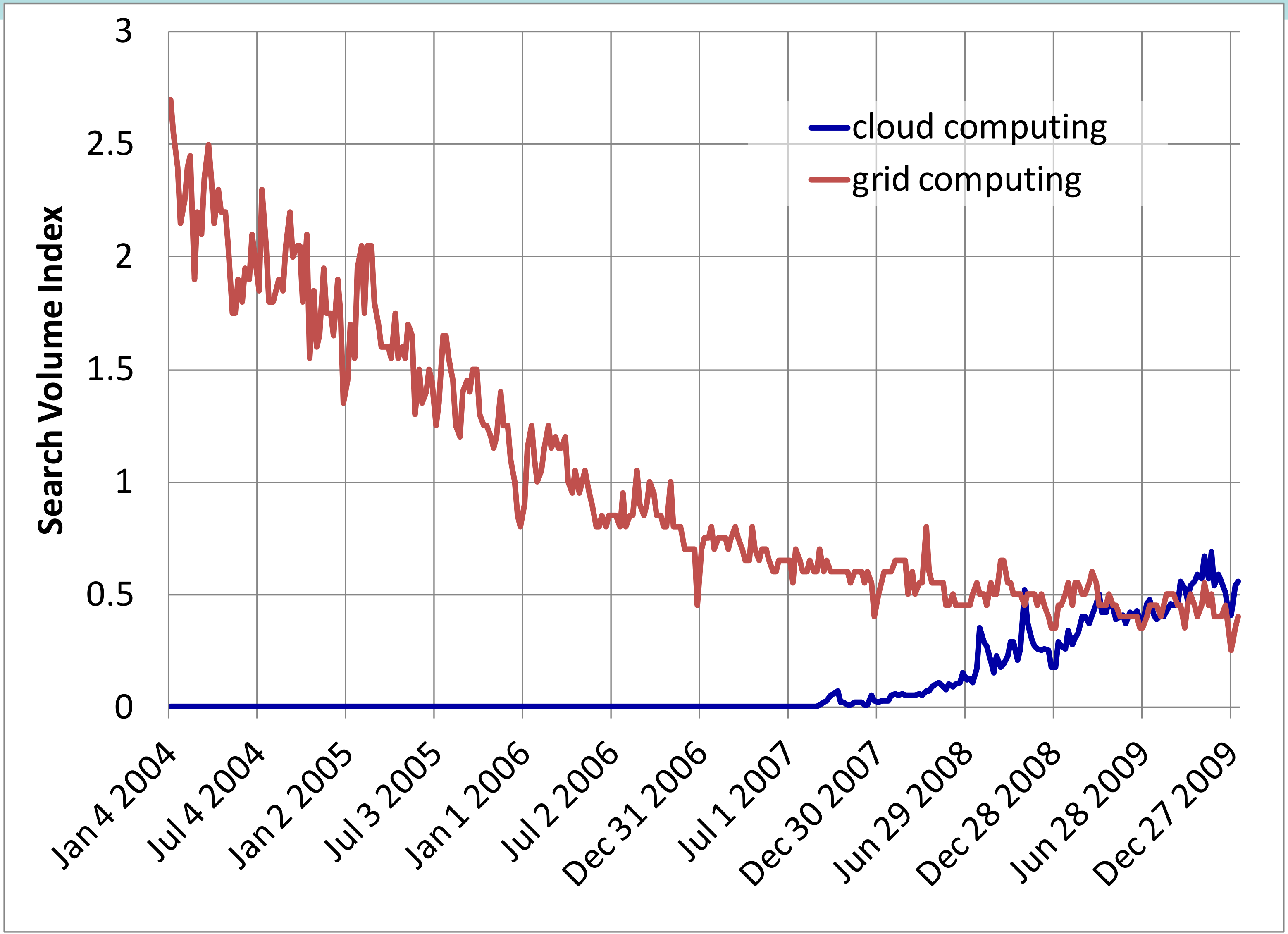


Grids ~ Federation

Major Grids

- TeraGrid (TG)
  - 200K-cores across 11 institutions and 22 systems over the US
- Open Science Grid (OSG)
  - 43K-cores across 80 institutions over the US
- Enabling Grids for E-sciencE (EGEE)
- LHC Computing Grid from CERN
- Middleware
  - Globus Toolkit
  - Unicore

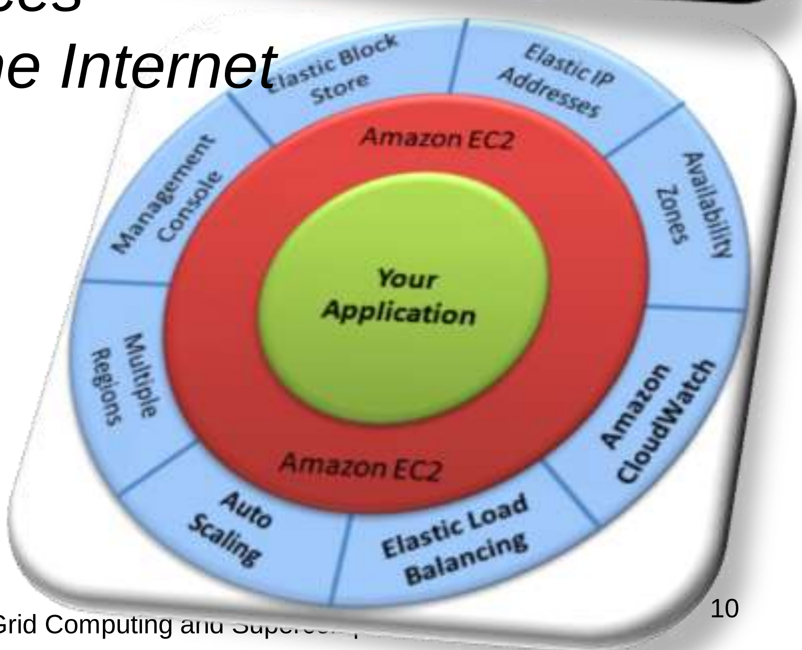
Cloud Computing: An Emerging Paradigm



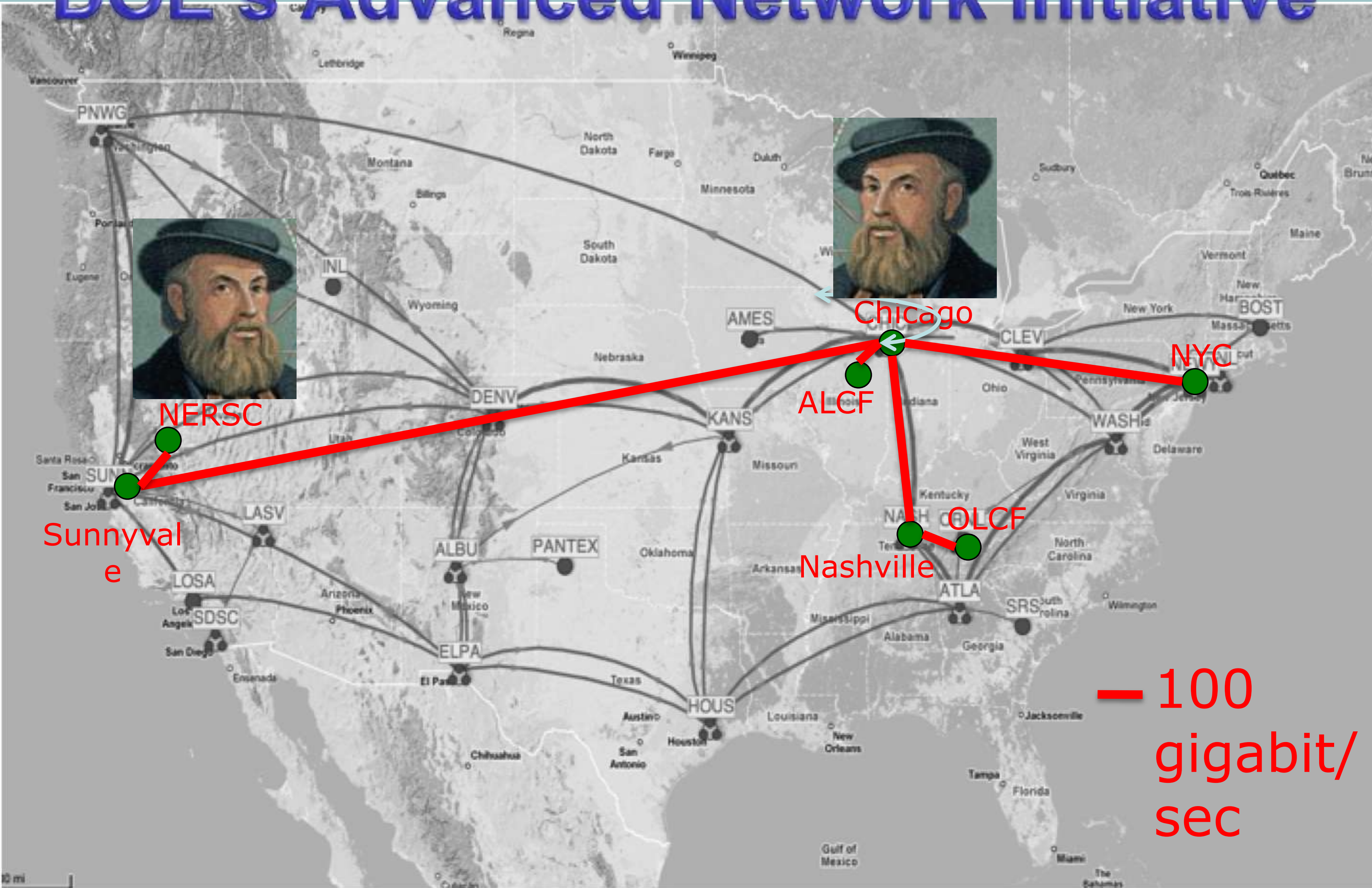
Cloud Computing

- *A large-scale distributed computing paradigm driven by:*
  1. *economies of scale*
  2. *virtualization*
  3. *dynamically-scalable resources*
  4. *delivered on demand over the Internet*

Clouds ~ hosting



Magellan + DOE's Advanced Network Initiative



Major Clouds

- Industry
  - Google App Engine
  - Amazon
  - Windows Azure
  - Salesforce
- Academia/Government
  - Magellan
  - FutureGrid
- Opensource middleware
  - Nimbus
  - Eucalyptus
  - OpenNebula

So is "Cloud Computing" just a new name for Grid?

- IT reinvents itself every five years
- The answer is complicated…

- **YES**: the vision is the same
  - to reduce the cost of computing
  - increase reliability
  - increase flexibility by transitioning from self operation to third party

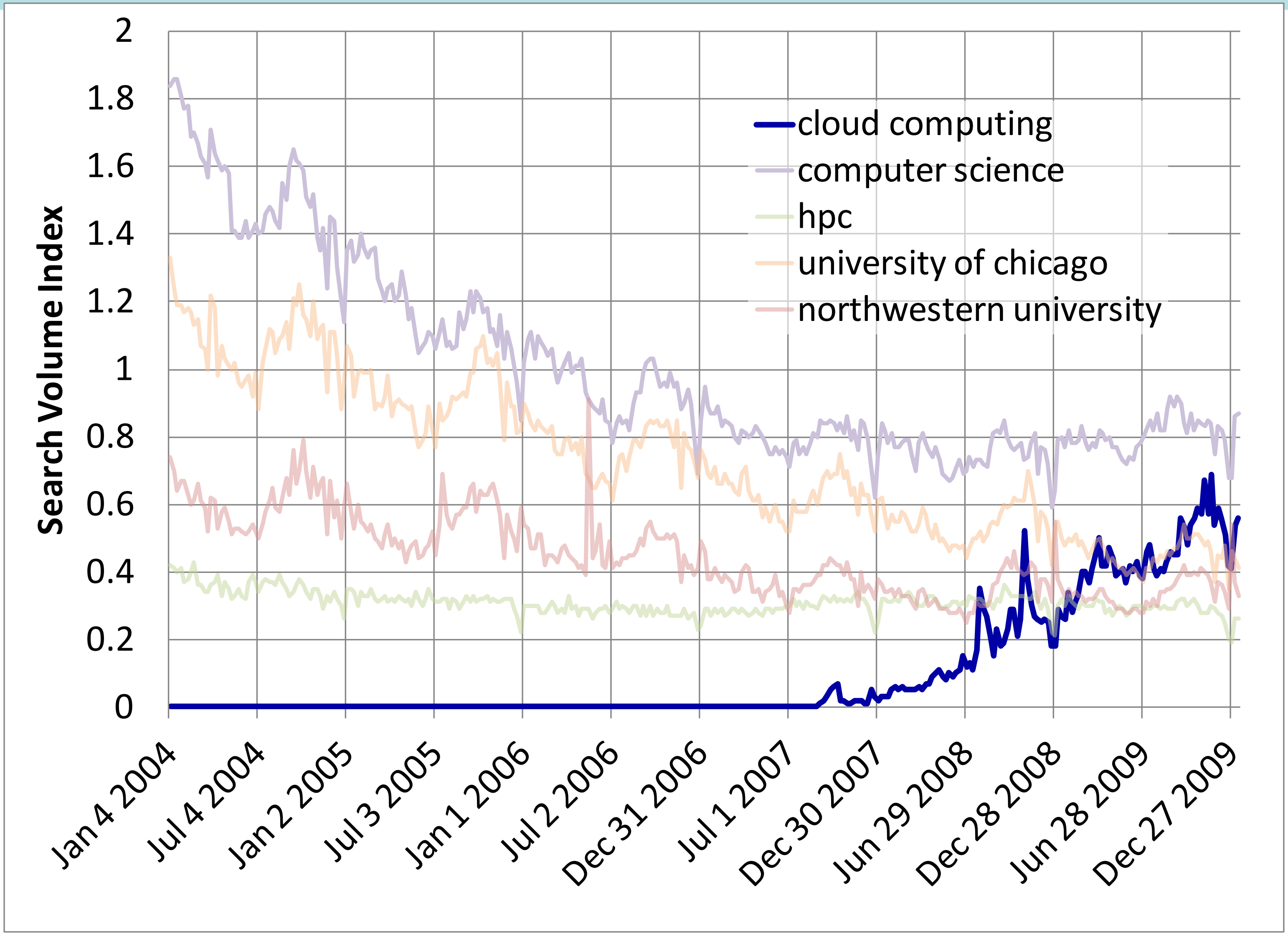
So is “Cloud Computing” just a new name for Grid?

- **NO**: things are different than they were 10 years ago
  - New needs to analyze massive data, increased demand for computing
  - Commodity clusters are expensive to operate
  - We have low-cost virtualization
  - Billions of dollars being spent by Amazon, Google, and Microsoft to create real commercial large-scale systems with hundreds of thousands of computers
  - The prospect of needing only a credit card to get on-demand access to *infinite computers is exciting; *infinite<O(1000)

So is "Cloud Computing" just a new name for Grid?

- **YES:** the problems are mostly the same
  - How to manage large facilities
  - Define methods to discover, request, and use resources
  - How to implement and execute parallel computations
  - Details differ, but issues are similar

How does Cloud Computing Compare?

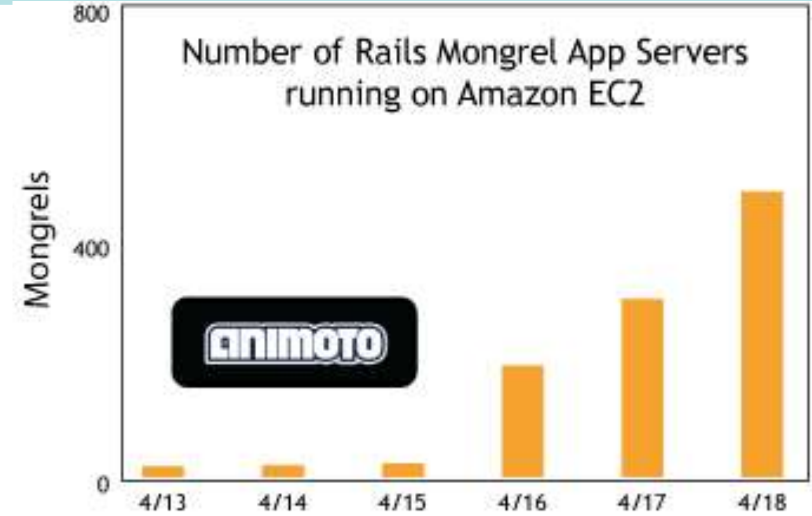


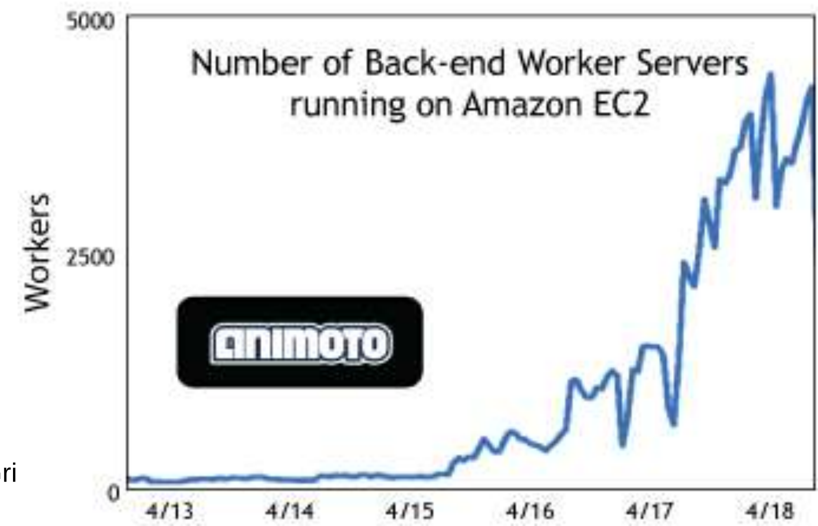
An Example of an Application in the Cloud

- [Animoto](#)
  - Makes it r … te videos with …

An Example of an Application in the Cloud

- Why is this a big deal?
  - No owned infrastructure
  - All resources rented on demand
- Critical for startups with risky business plans
- Not possible without Cloud Computing and a credit card
  - Launched in 2007/2008 timeframe



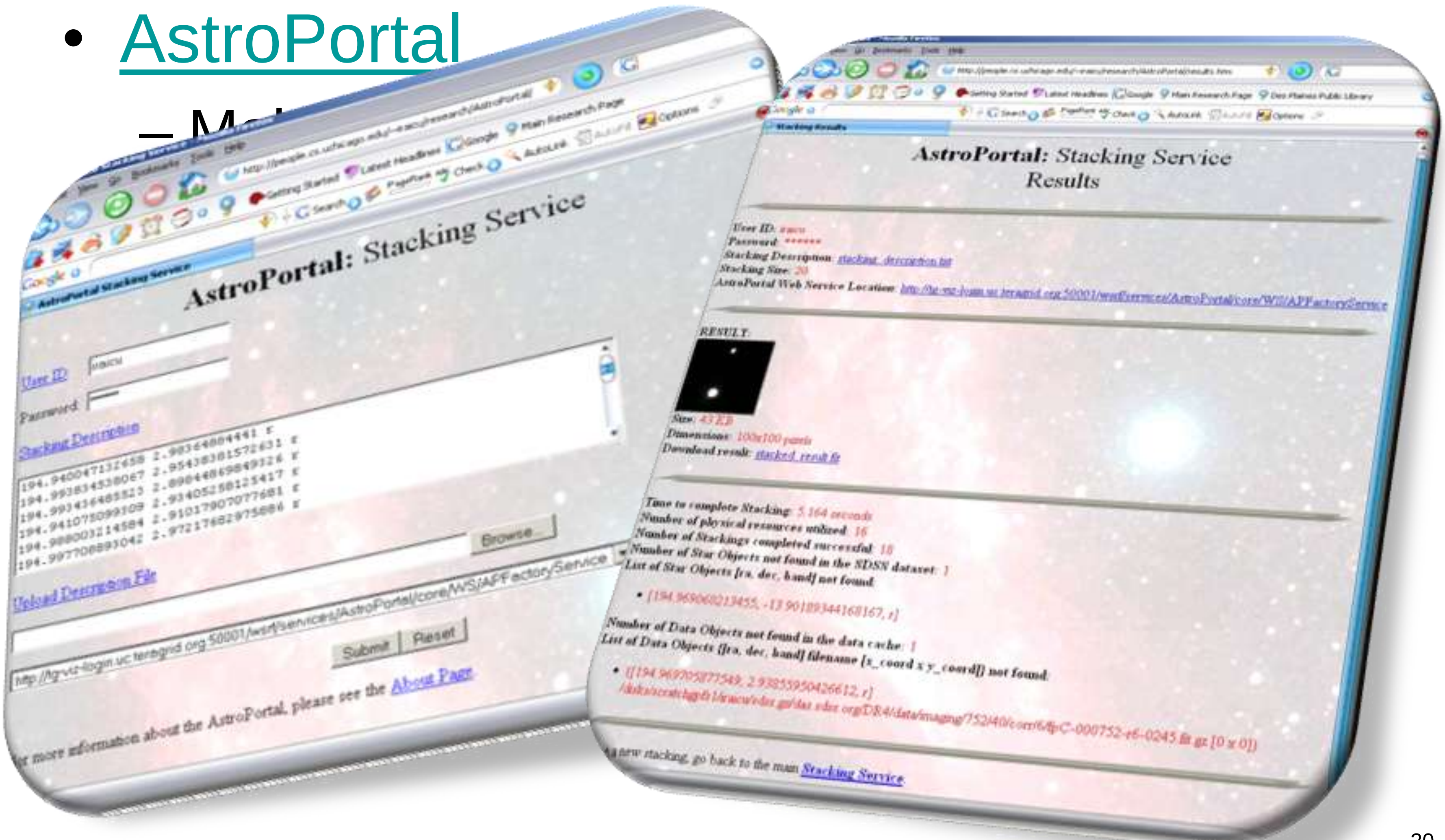


Outline

- Overview
- **Contributions**
- Applications
- Conclusions

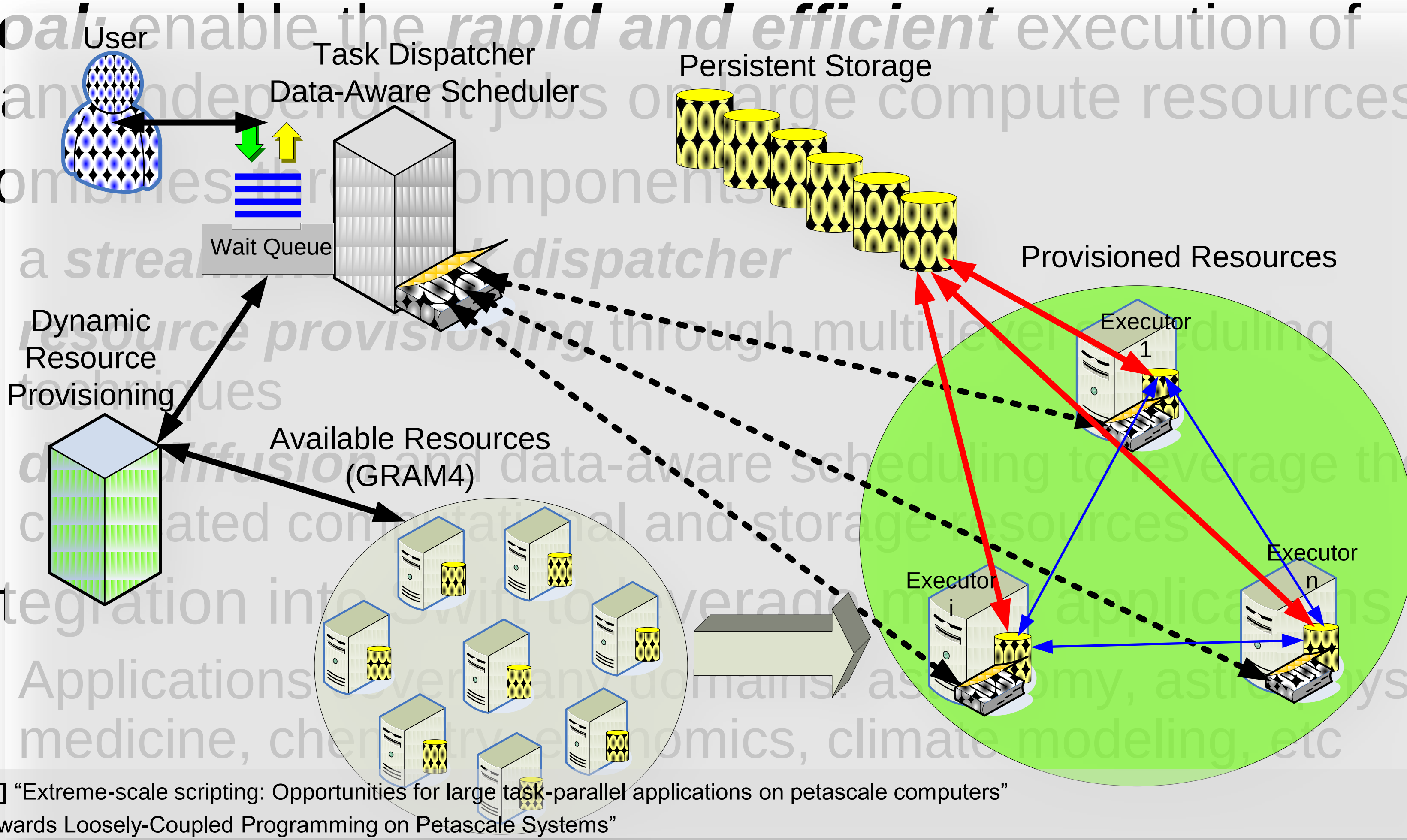
An Example of an Application in the Grid

- AstroPortal
  - M



Novel Resource Management Approach: Falkon Architecture

- ***Goal:*** enable the ***rapid and efficient*** execution of many independent jobs on large compute resources
- 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"

High-Throughput Computing & High-Performance Computing

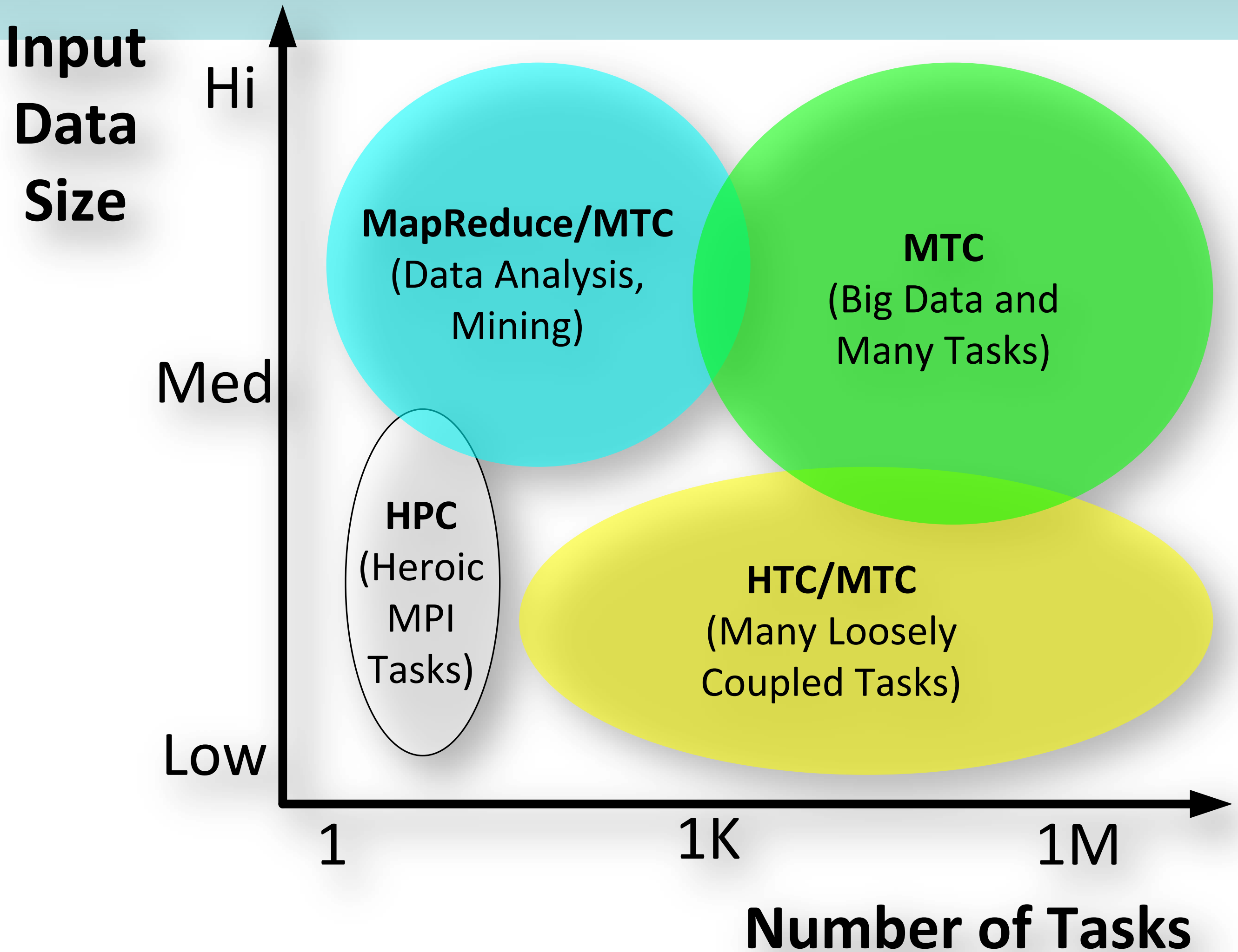
- **HTC: High-Throughput Computing**
  - Typically applied in clusters and grids
  - Loosely-coupled applications with sequential jobs
  - Large amounts of computing for long periods of times
  - Measured in operations per month or years
- **HPC: High-Performance Computing**
  - Synonymous with supercomputing
  - 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

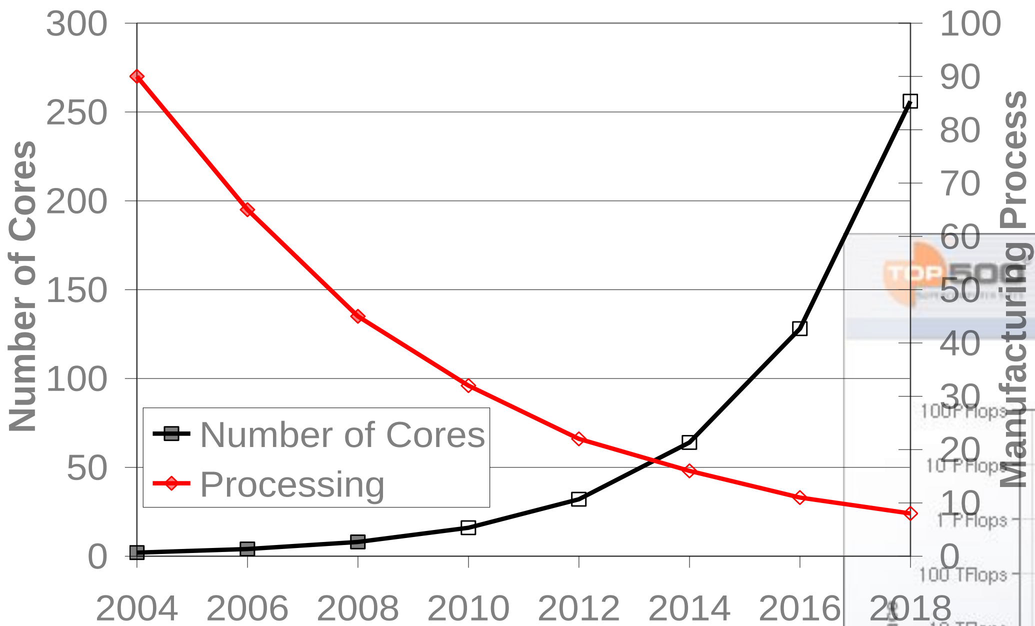
[MTAGS08 Workshop] Workshop on Many-Task Computing on Grids and Supercomputers 2008
[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"
[MTAGS08] "Many-Task Computing for Grids and Supercomputers"

Problem Space

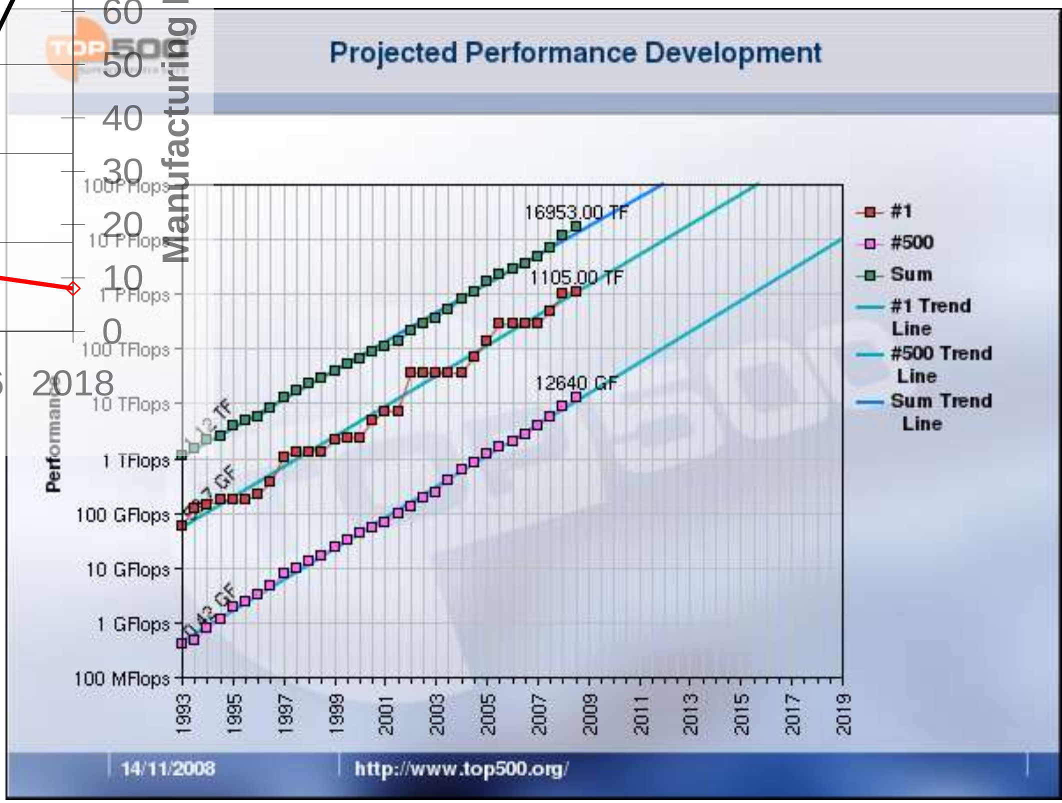


[MTAGS08] "Many-Task Computing for Grids and Supercomputers"

Projected Growth Trends



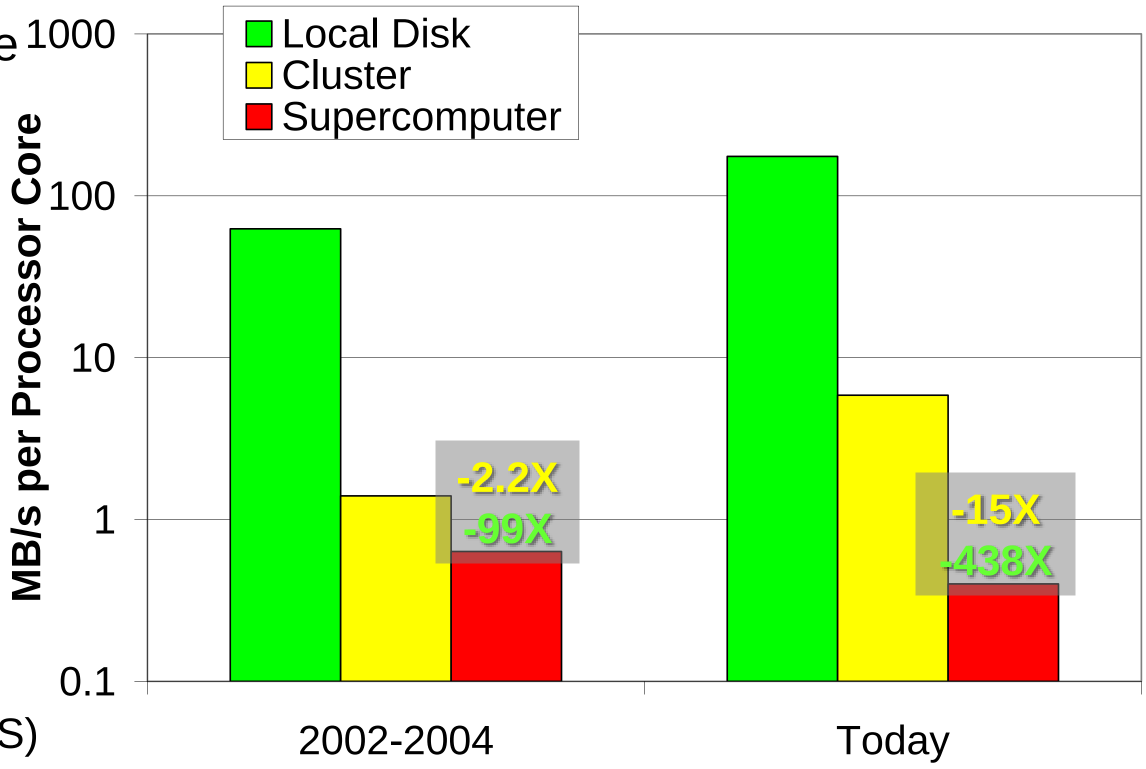
Pat Helland, Microsoft, The Irresistible Forces Meet the Movable Objects, November 9th, 2007



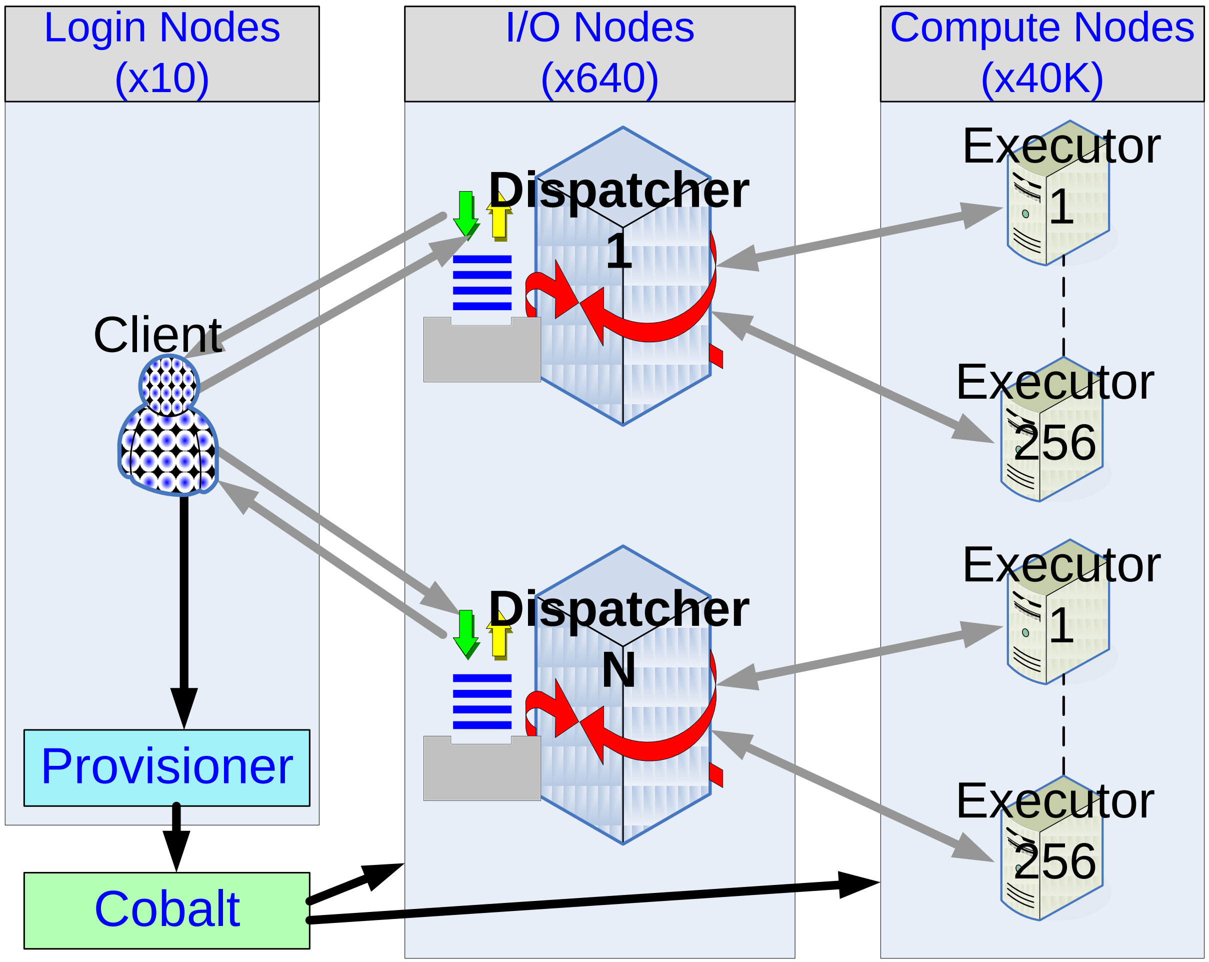
Top500 Projected Development, http://www.top500.org/lists/2008/11/performance_development

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)

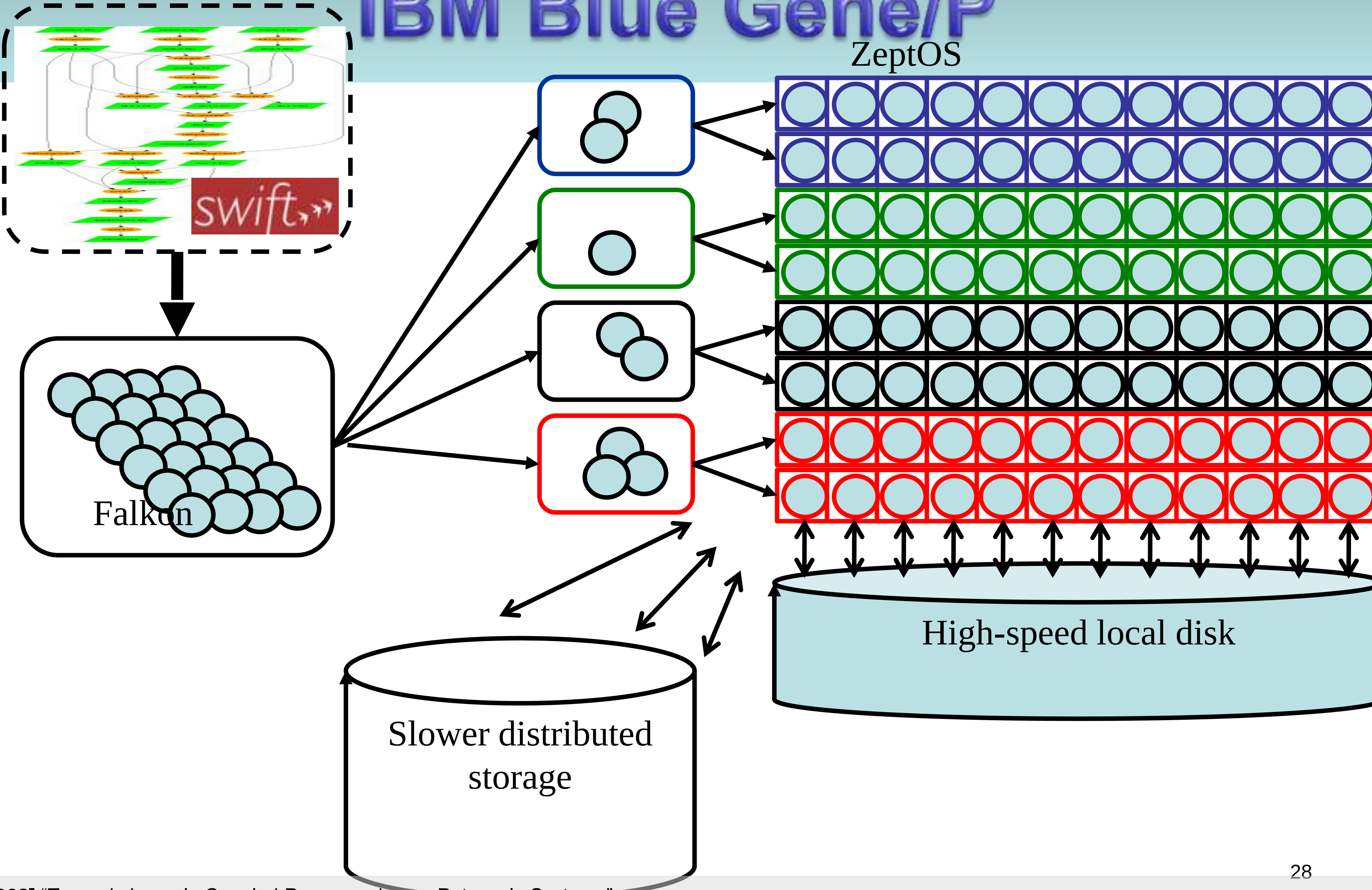


Distributed Falkon Architecture



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

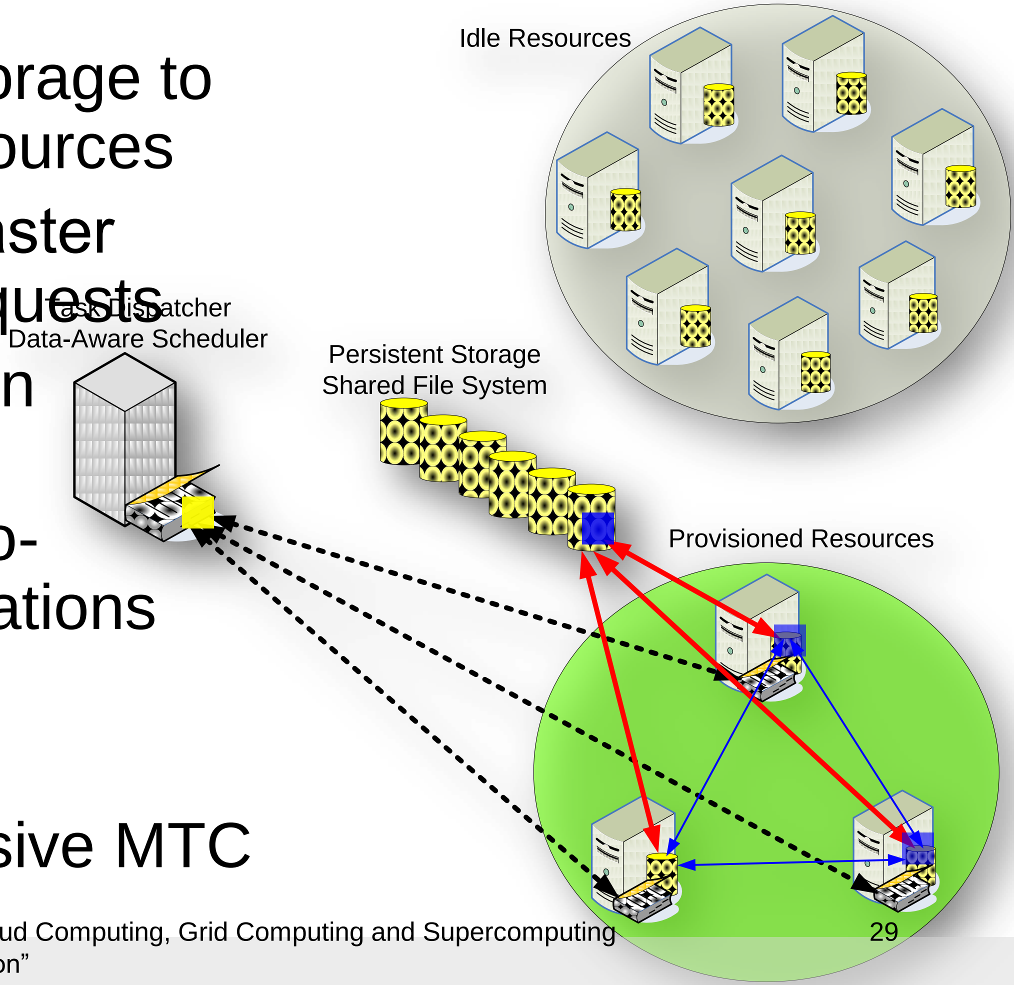
Managing 160K CPUs IBM Blue Gene/P



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

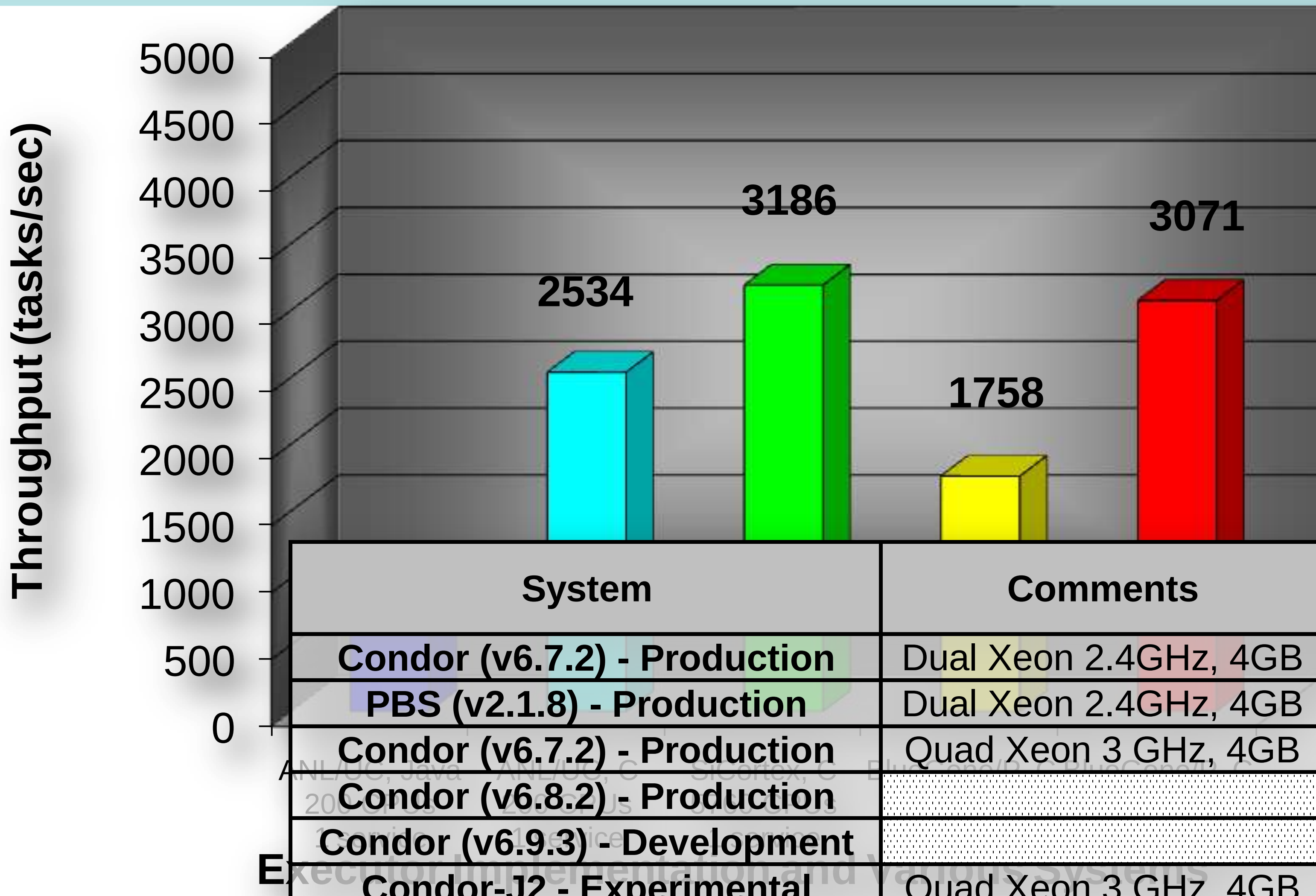
Data Diffusion

- 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"

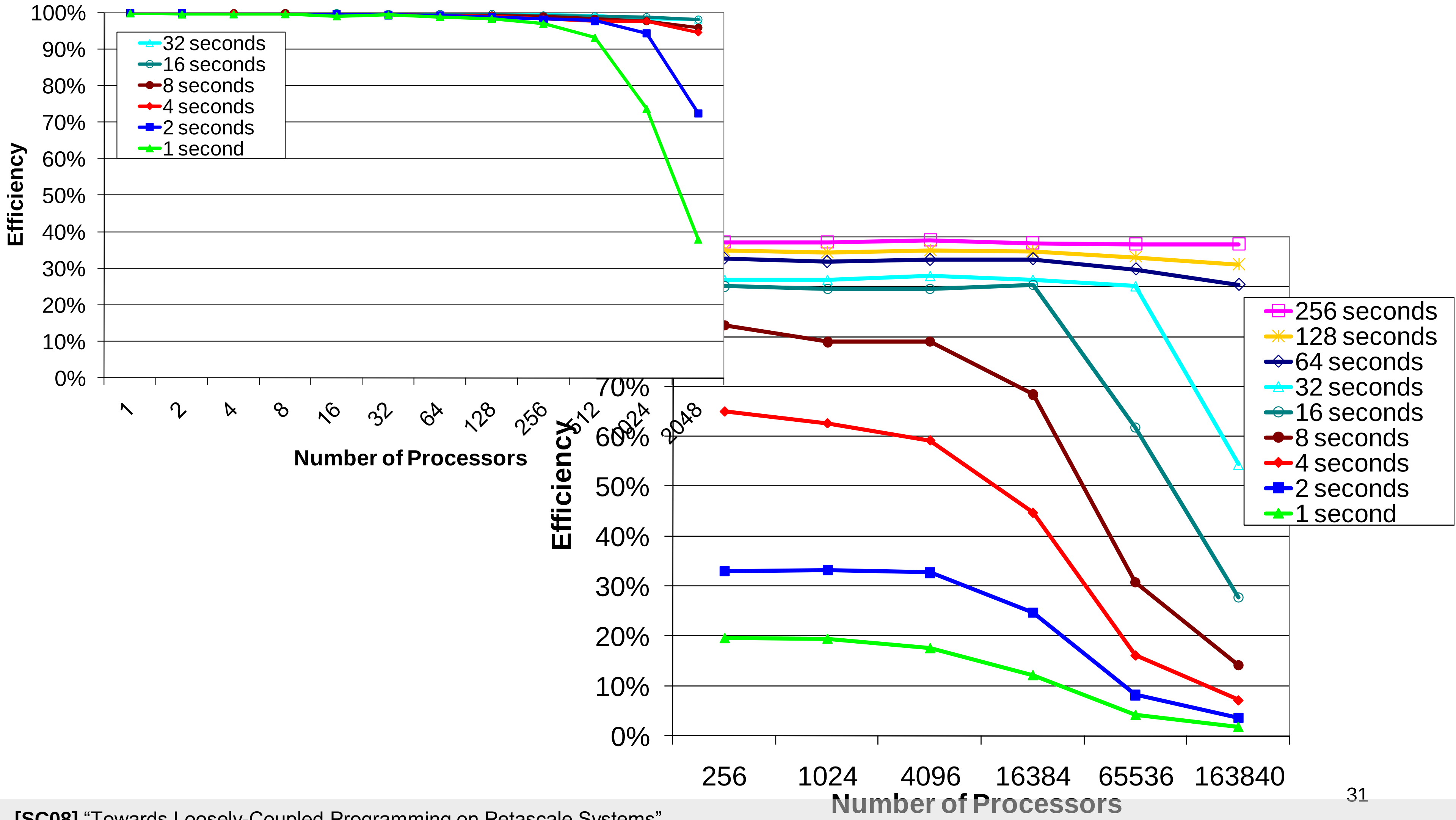
Dispatch Throughput



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

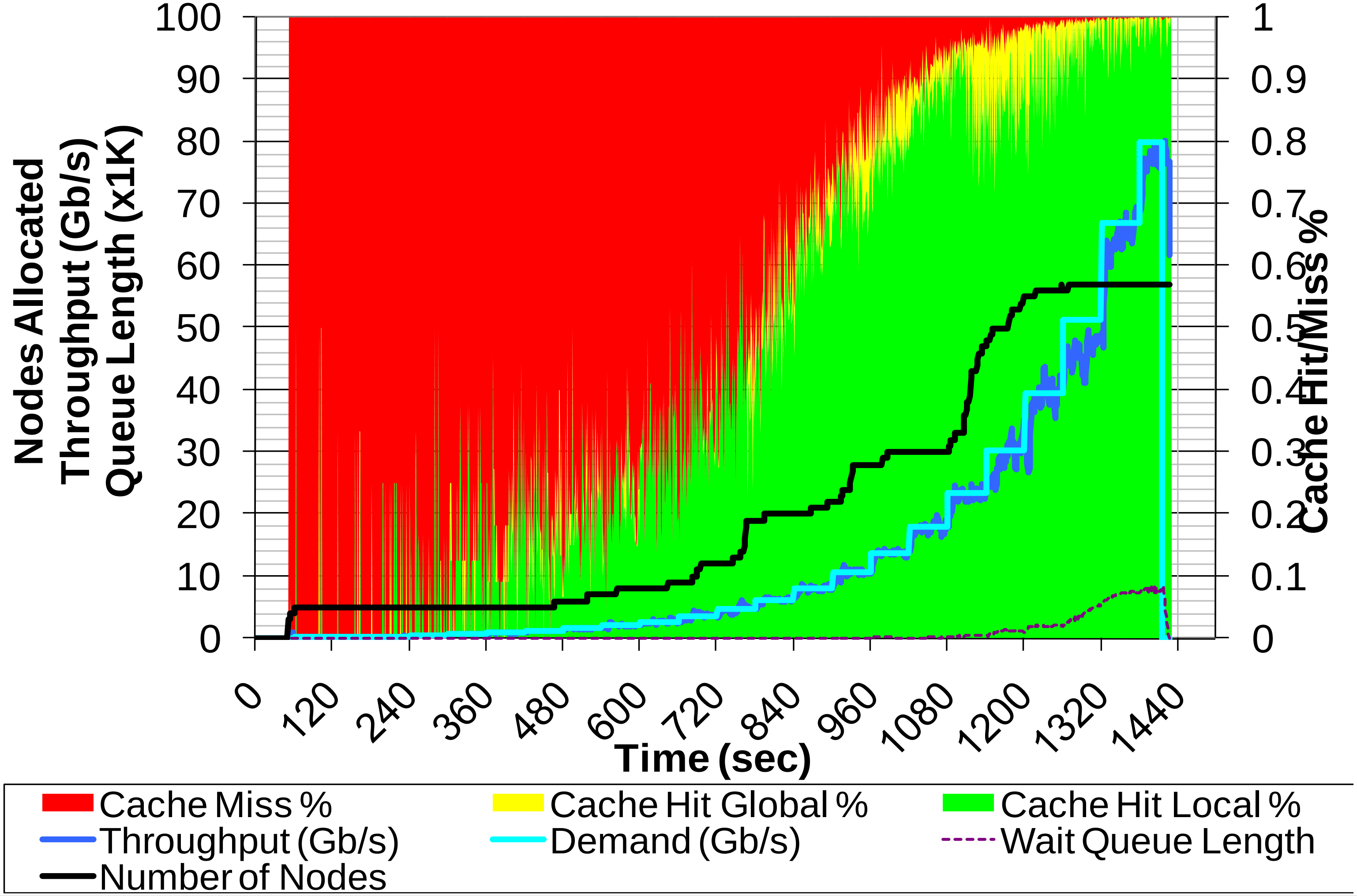
[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

Execution Efficiency



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

Data Diffusion
Monotonically Increasing Workload



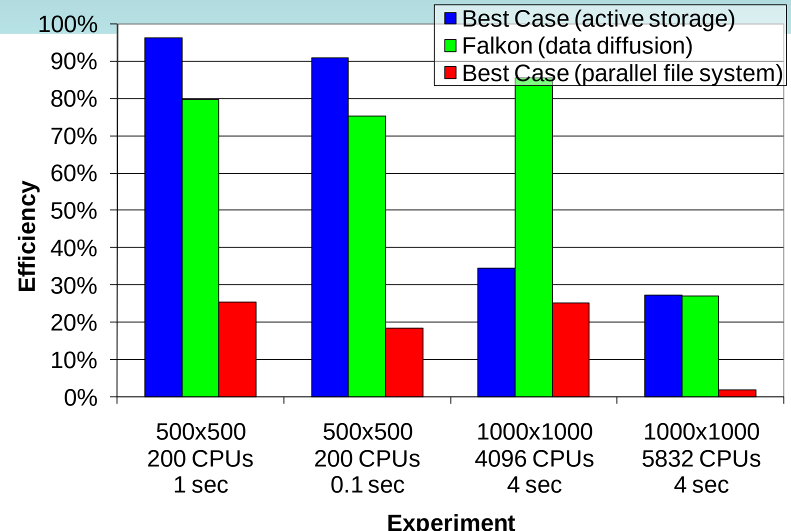
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems"

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

Data Diffusion 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



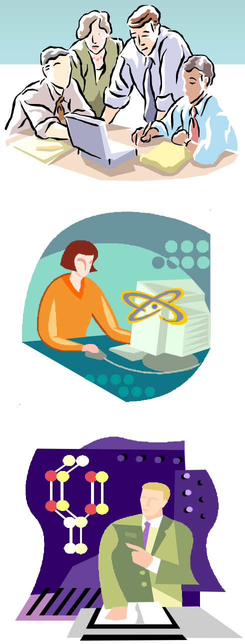
Experiment	Approach	Local Disk/Memory (GB)	Network (node-to-node) (GB)	Shared File System (GB)
500x500 200 CPUs 1 sec	Best Case (active storage)	6000	1536	12
	Falkon (data diffusion)	6000	1698	34
500x500 200 CPUs 0.1 sec	Best Case (active storage)	6000	1536	12
	Falkon (data diffusion)	6000	1528	62
1000x1000 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

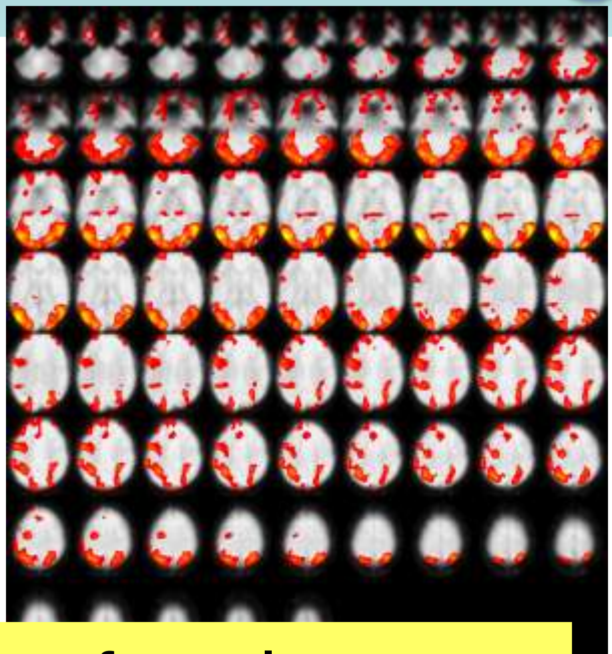
[HPDC09] “The Quest for Scalable Support of Data Intensive Applications in Distributed Systems”, under review
[DIDC09] “Towards Data Intensive Many-Task Computing”, under review

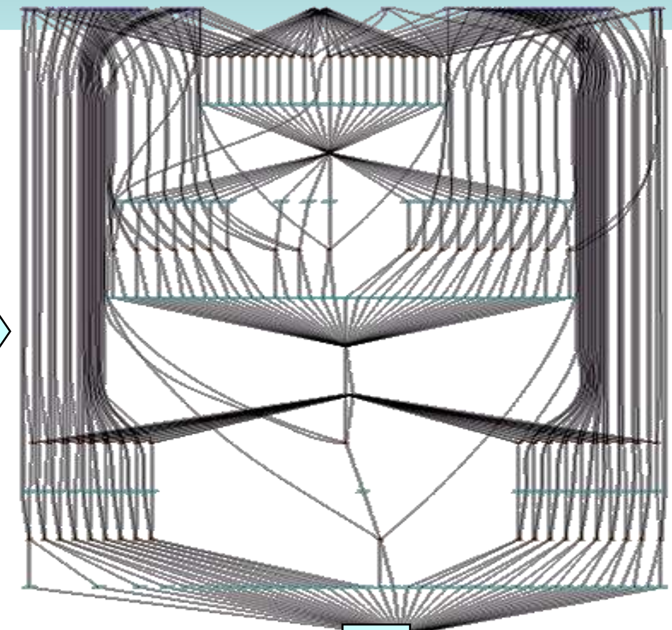
Outline

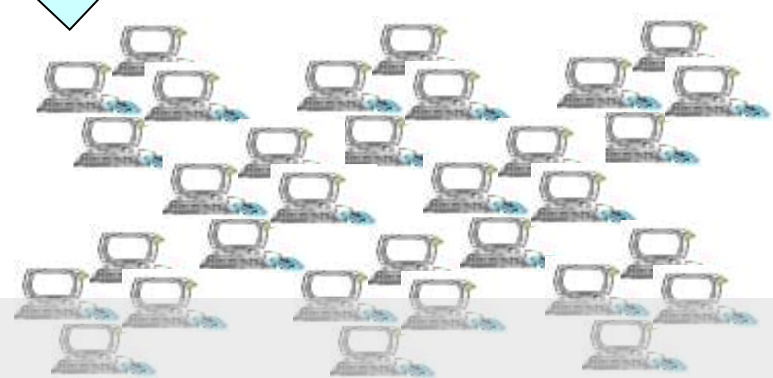
- Overview
- Contributions
- **Applications**
- Conclusions

Applications
Medical Imaging: fMRI







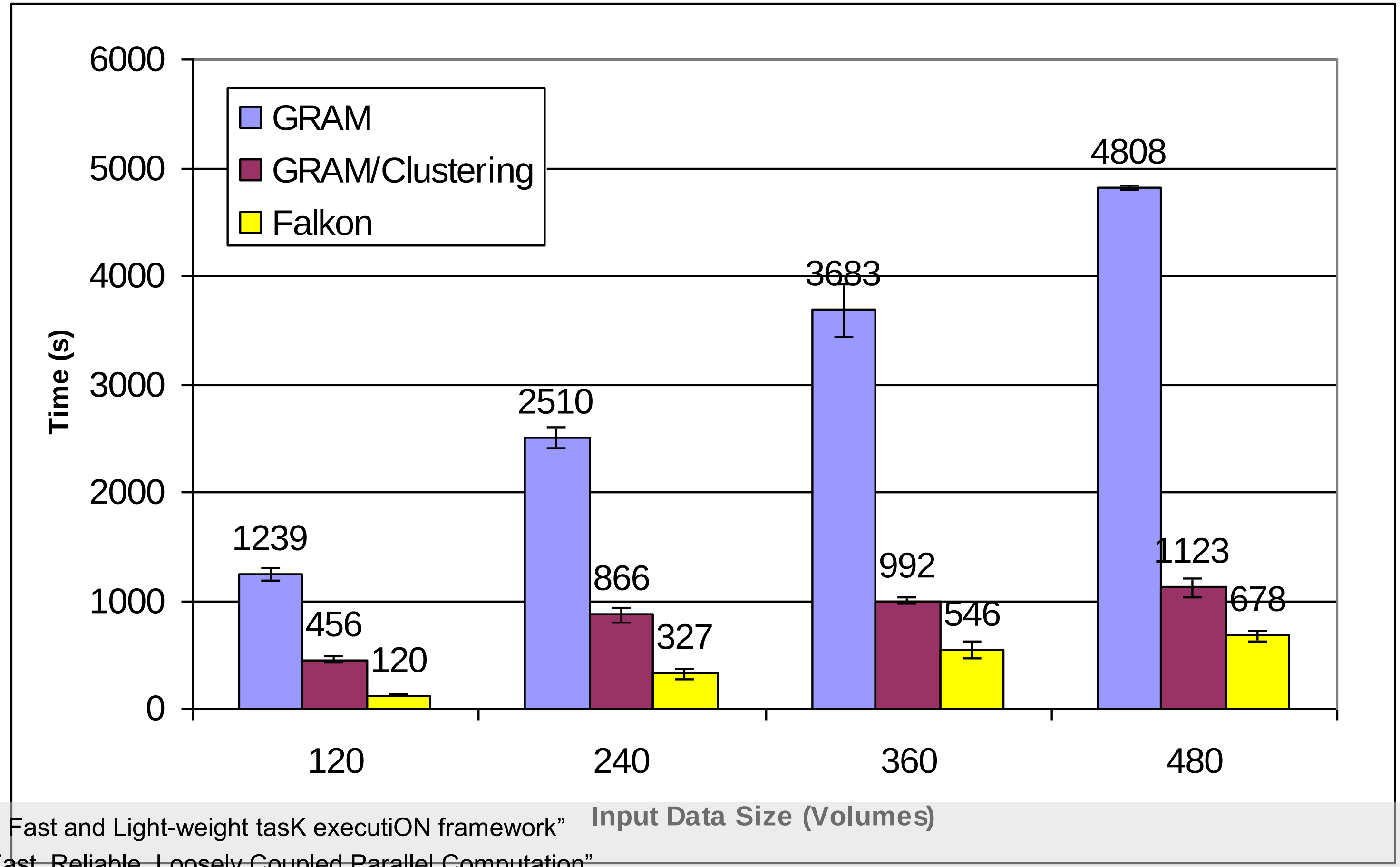


- Wide range of analyses
  - Testing, interactive analysis, production runs
  - Data mining
  - Parameter studies

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

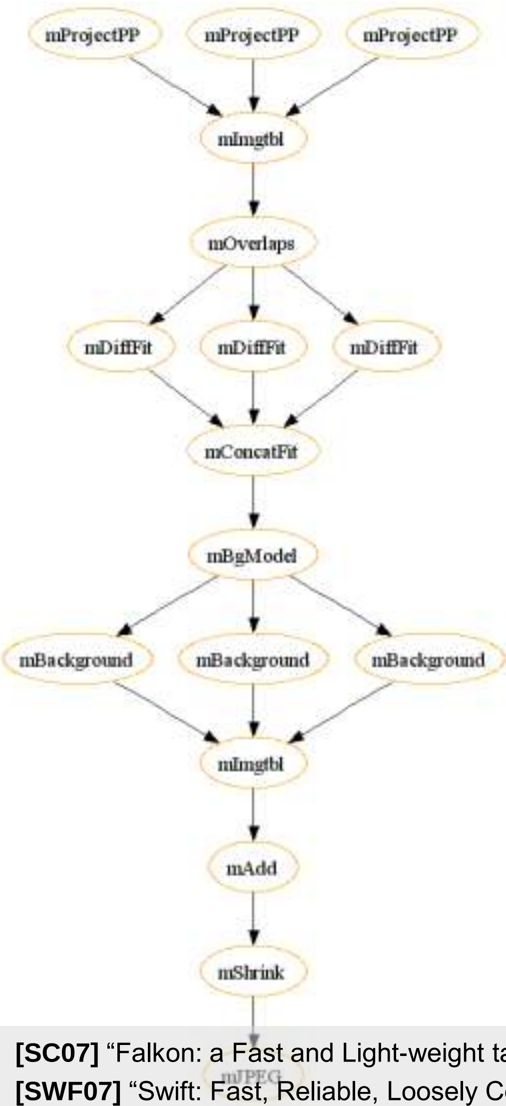
Applications
Medical Imaging: fMRI

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

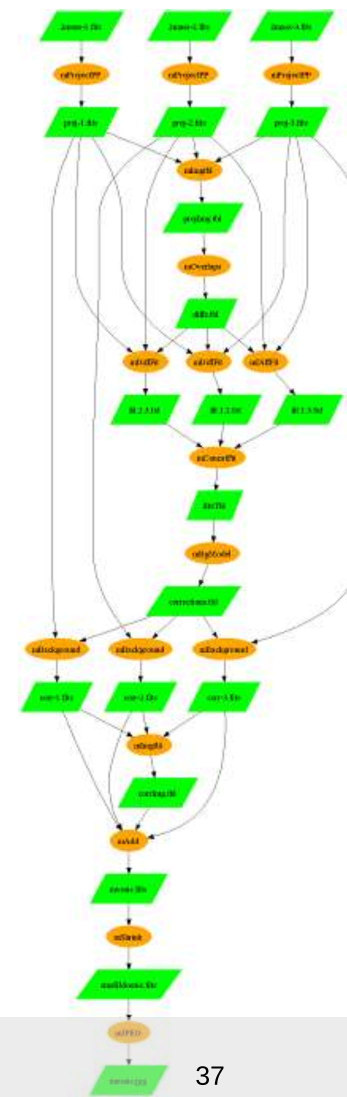


[SC07] "Falkon: a Fast and Light-weight tasK executiON framework"
[SWF07] "Swift: Fast, Reliable, Loosely Coupled Parallel Computation"

Applications
Astronomy: Montage



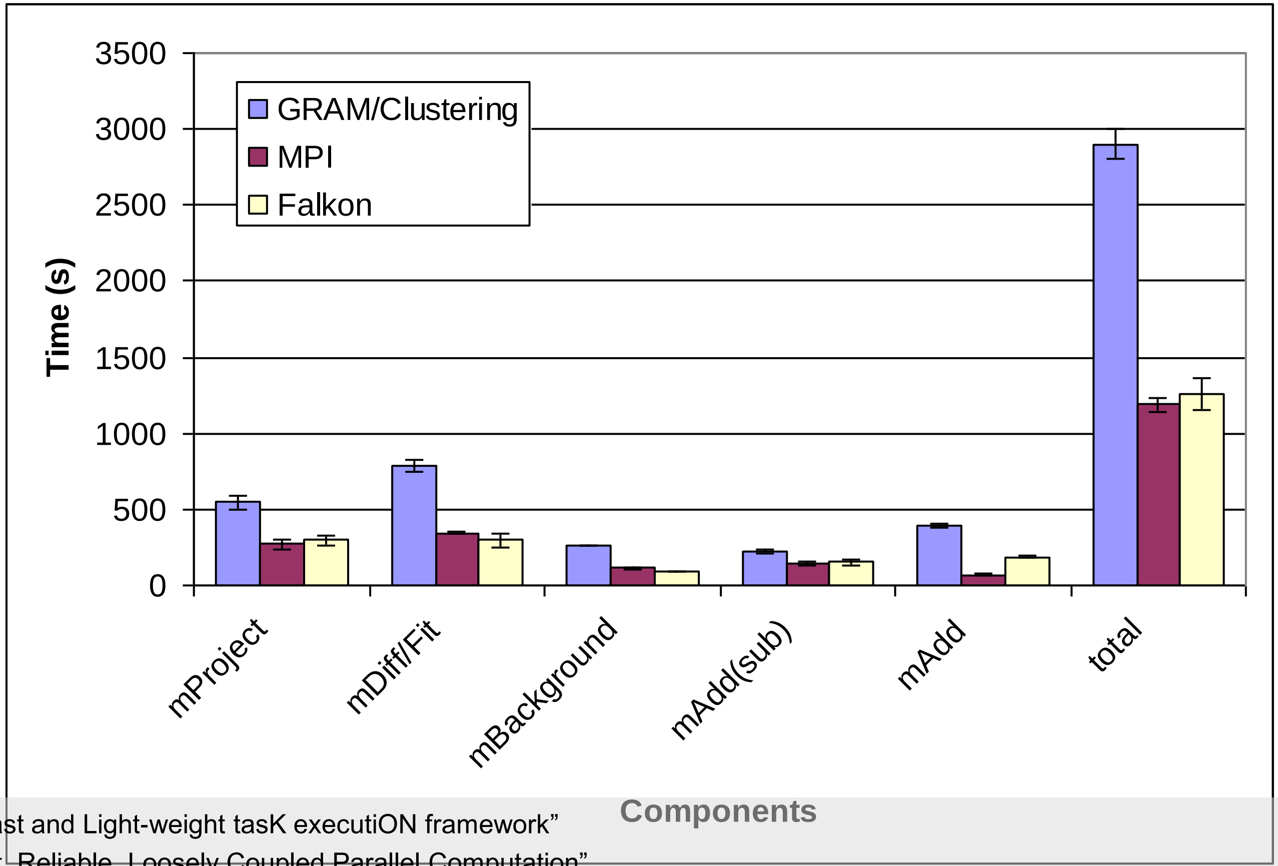
B. Berriman, J. Good (Caltech)
J. Jacob, D. Katz (JPL)



[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

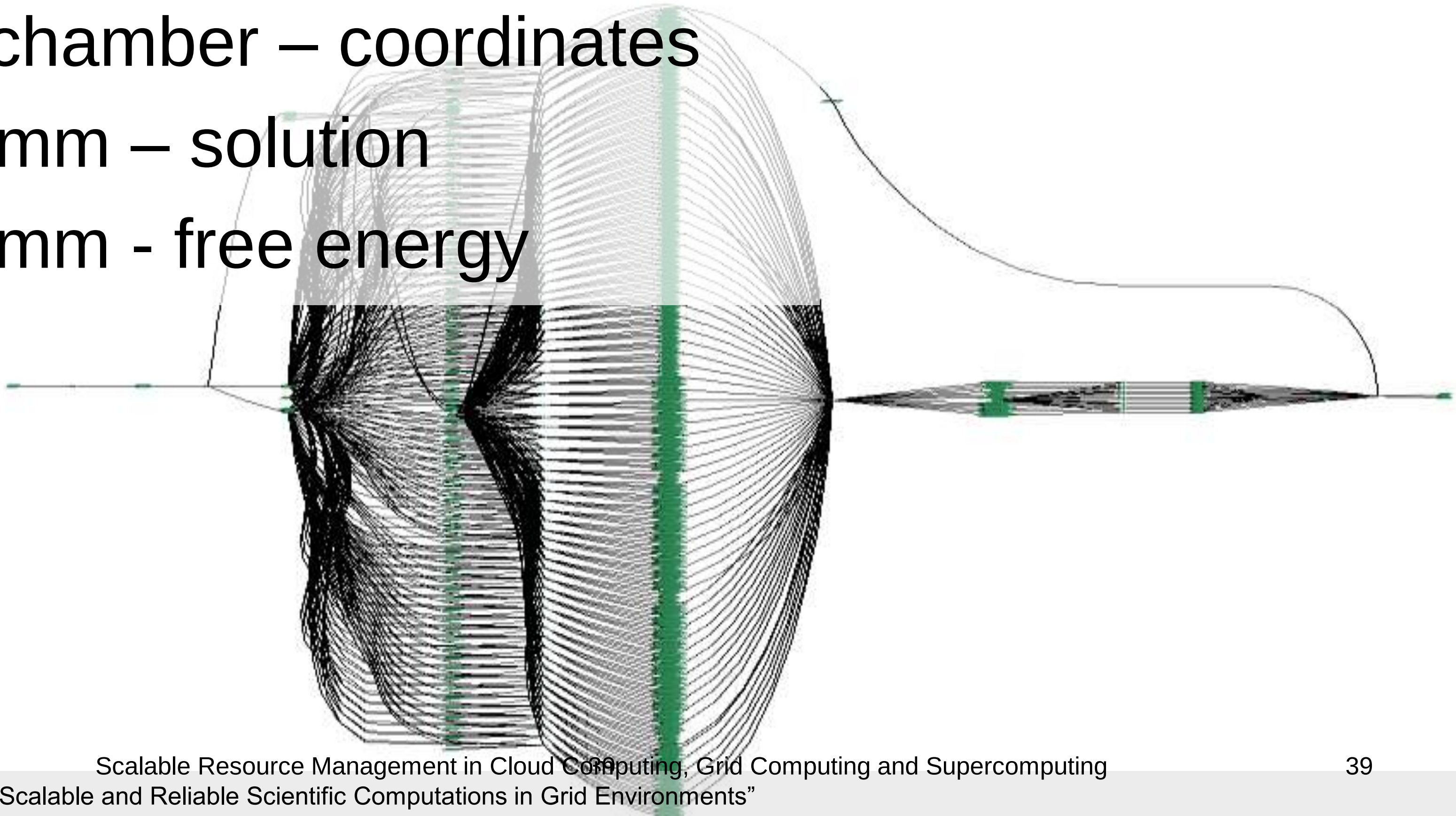


[SC07] "Falkon: a Fast and Light-weight tasK executiON framework"

[SWF07] "Swift: Fast, Reliable, Loosely Coupled Parallel Computation"

Applications
Molecular Dynamics: MolDyn

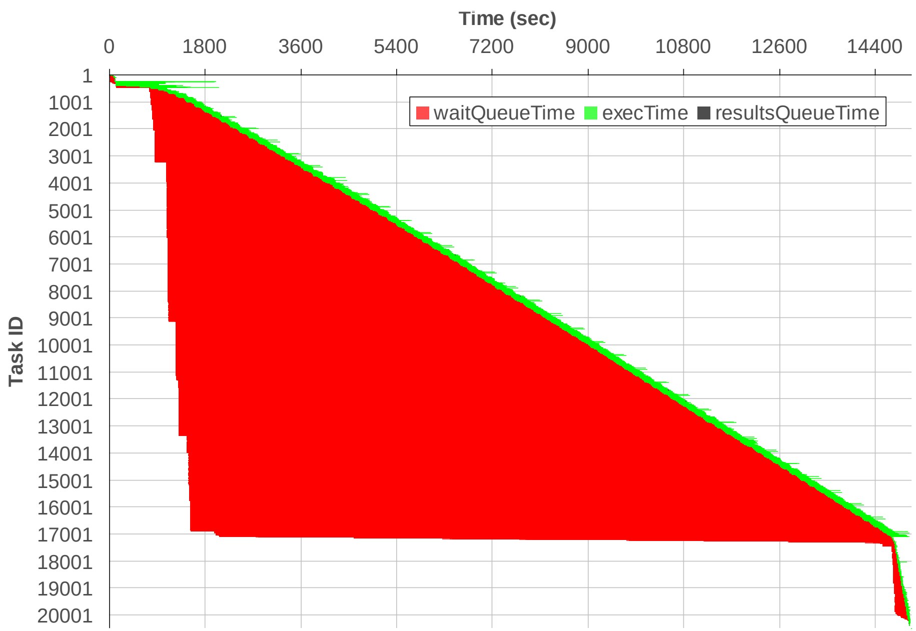
- Determination of free energies in aqueous solution
  - Antechamber – coordinates
  - Charmm – solution
  - Charmm - free energy

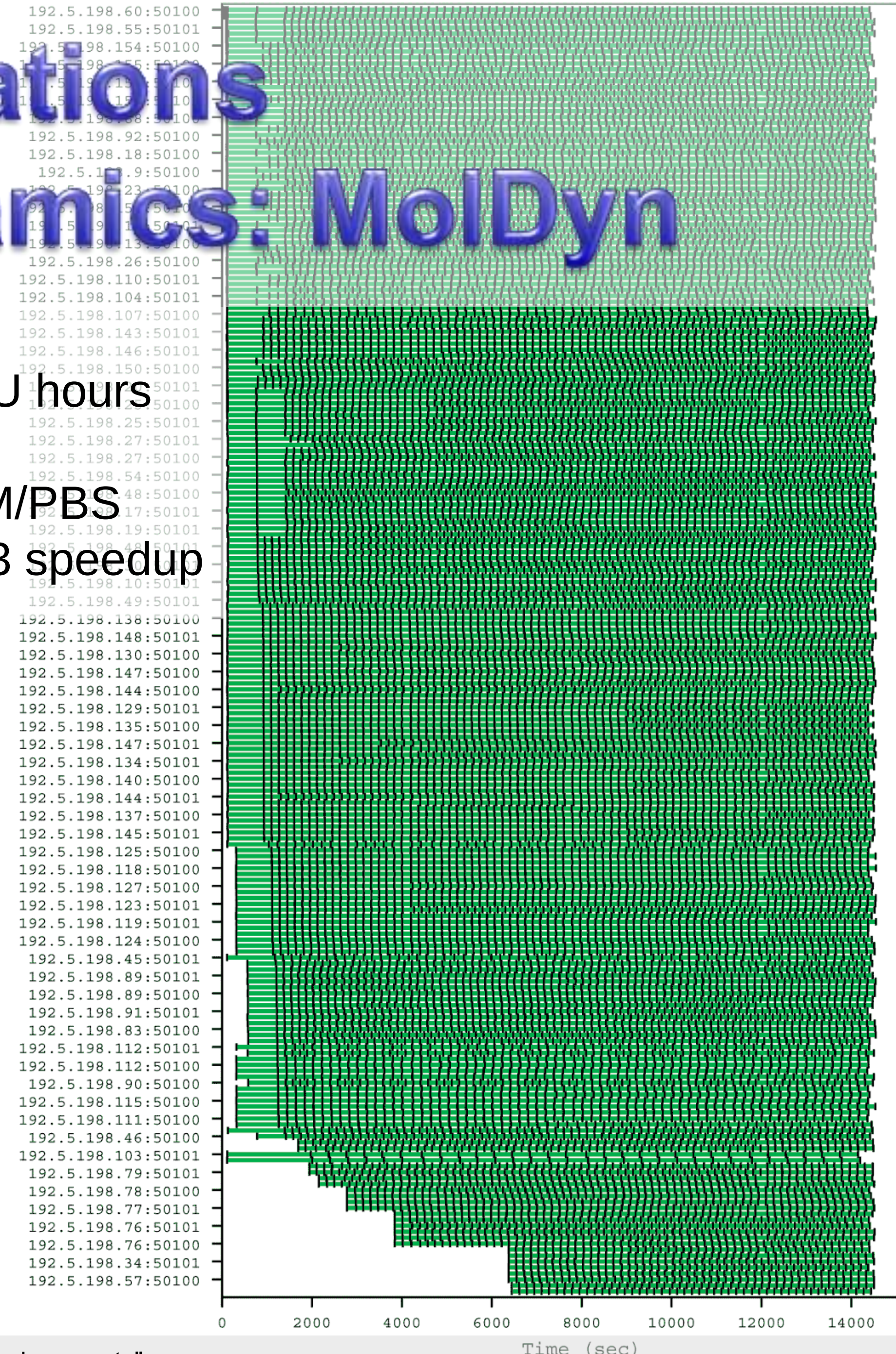


[NOVA08] “Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments”

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

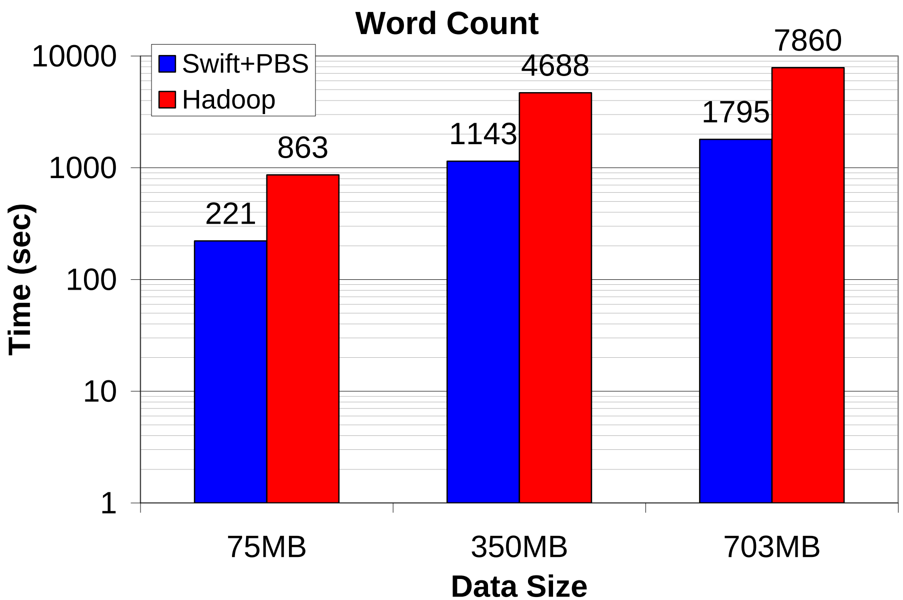


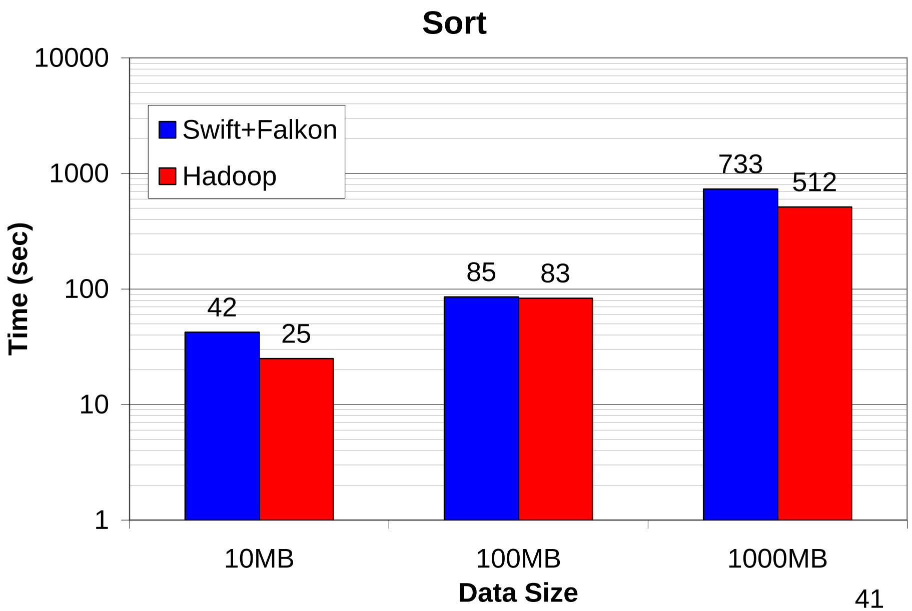


[NOVA08] "Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments"

Applications
Word Count and Sort

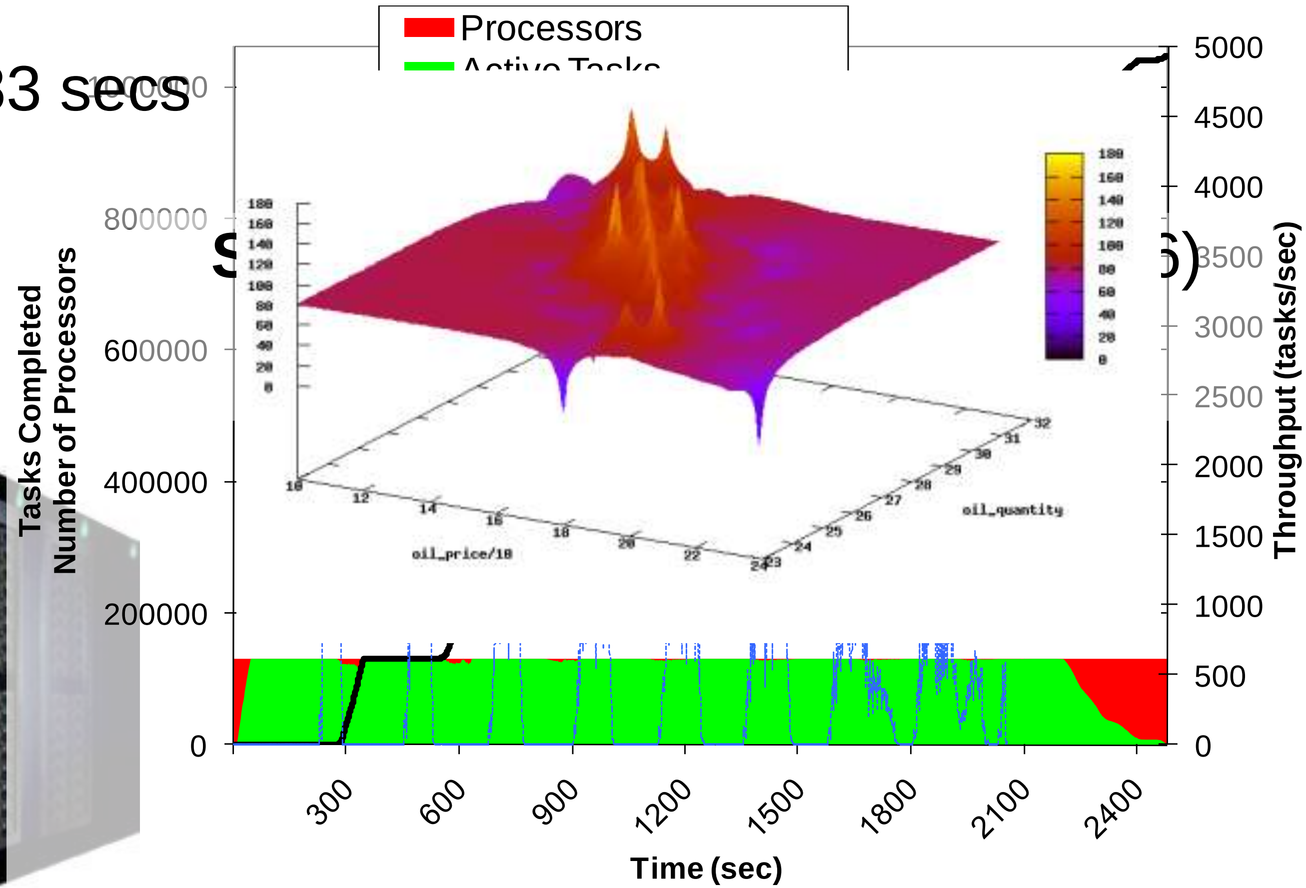
- Classic benchmarks for MapReduce
  - Word Count
  - Sort
- Swift and Falkon performs similar or better than Hadoop (on 32 processors)





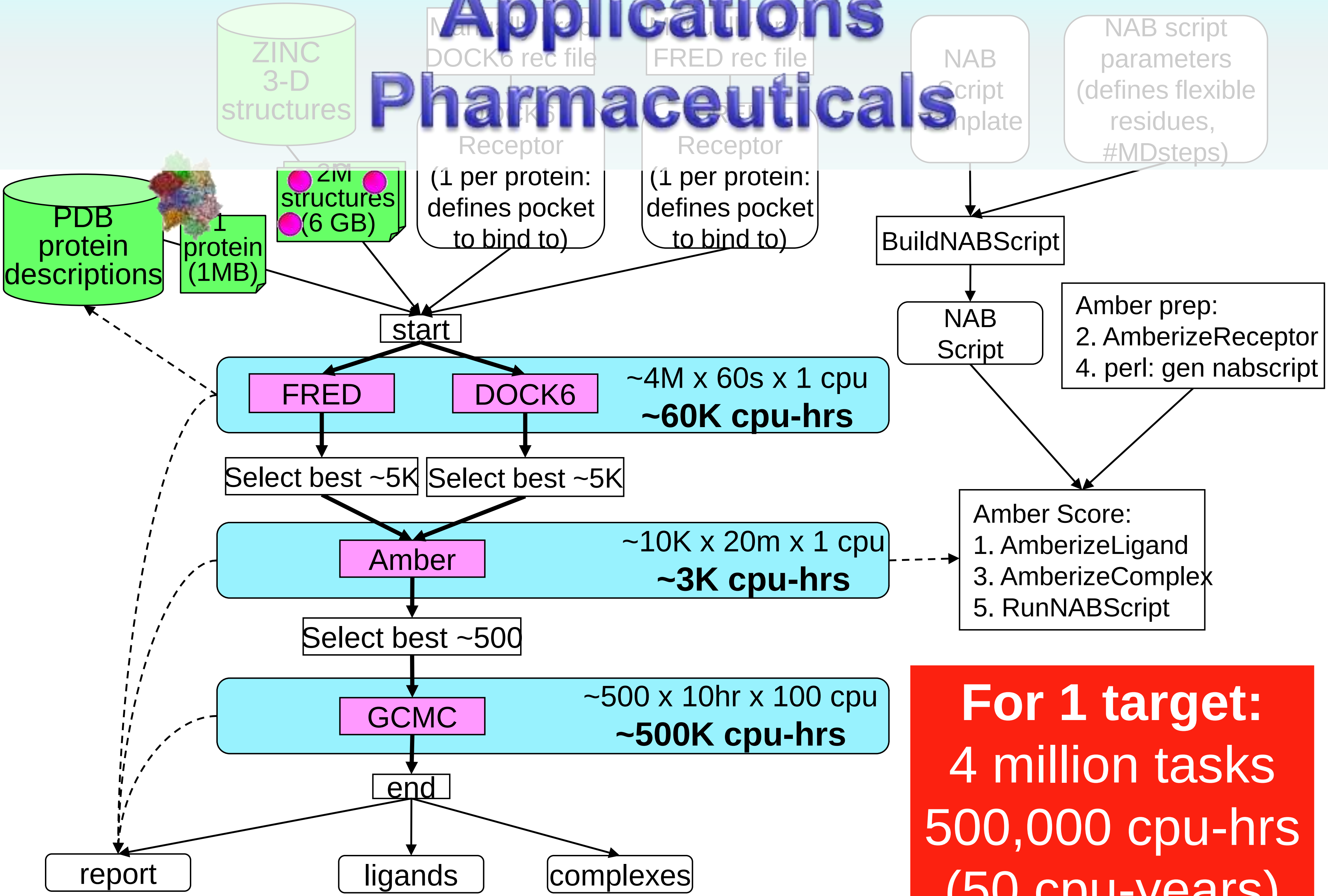
Applications
Economic Modeling: MARS

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



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

Applications
Pharmaceuticals



Applications
Pharmaceuticals: DOCK

CPU cores: 118784

Tasks: 934803

Elapsed time: 2.01 hours

Compute time: 21.43 CPU years

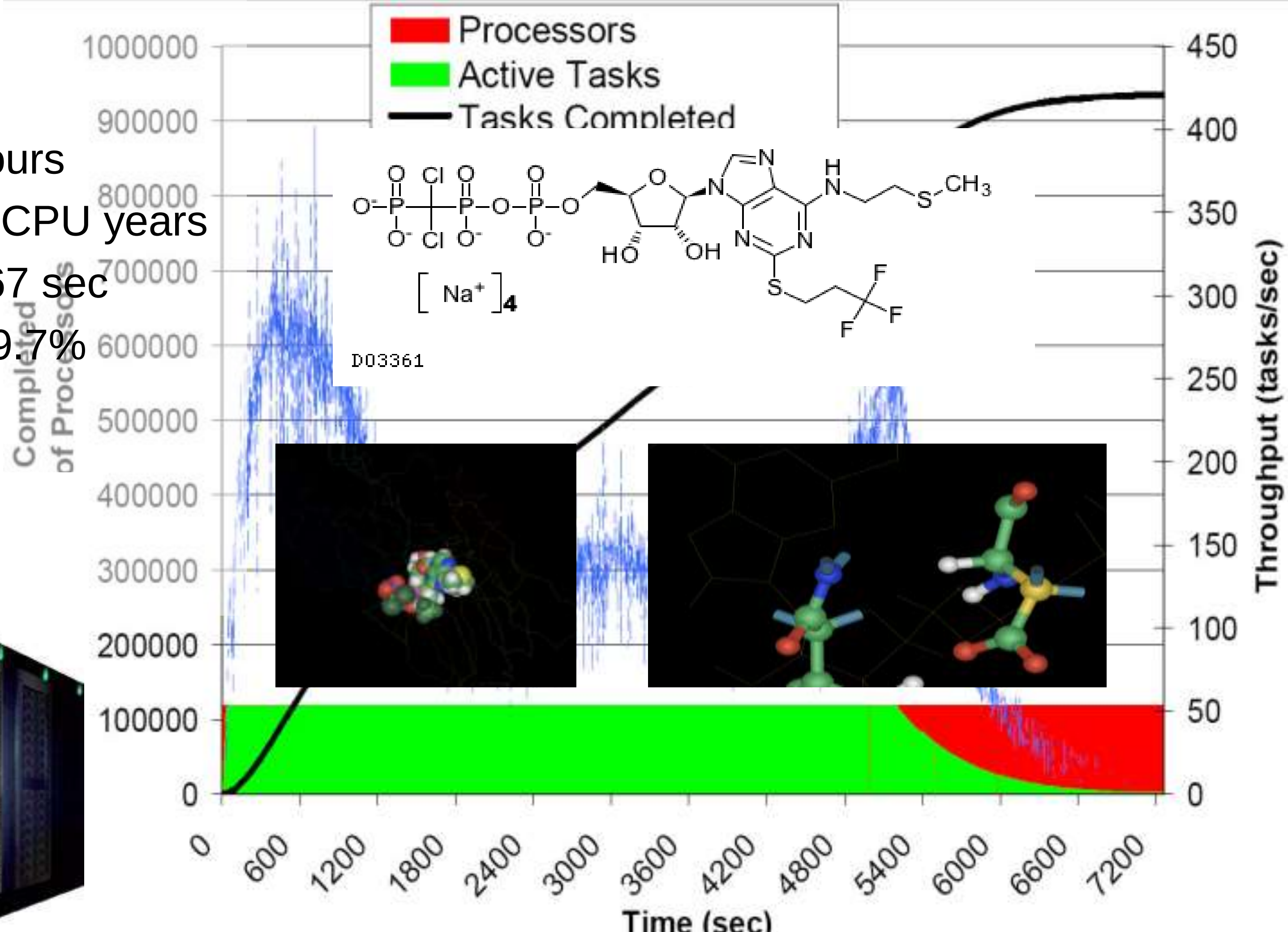
Average task time: 667 sec

Relative Efficiency: 99.7%

(from 16 to 32 racks)

Utilization:

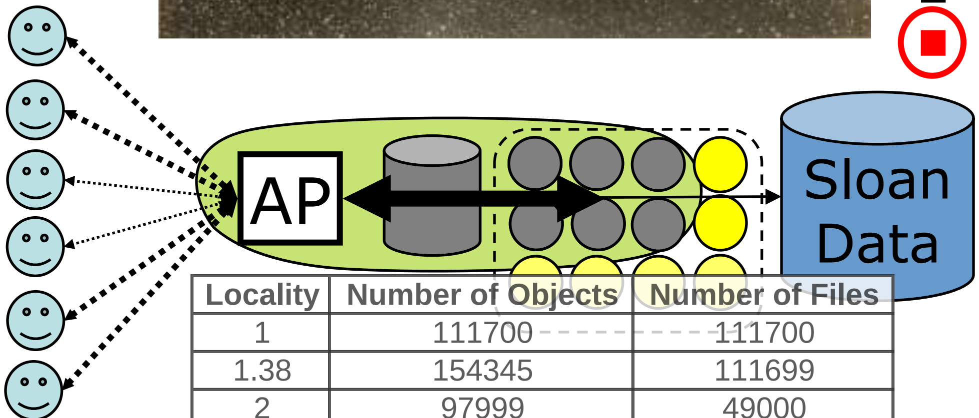
- Sustained: 99.6%
- Overall: 78.3%



[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

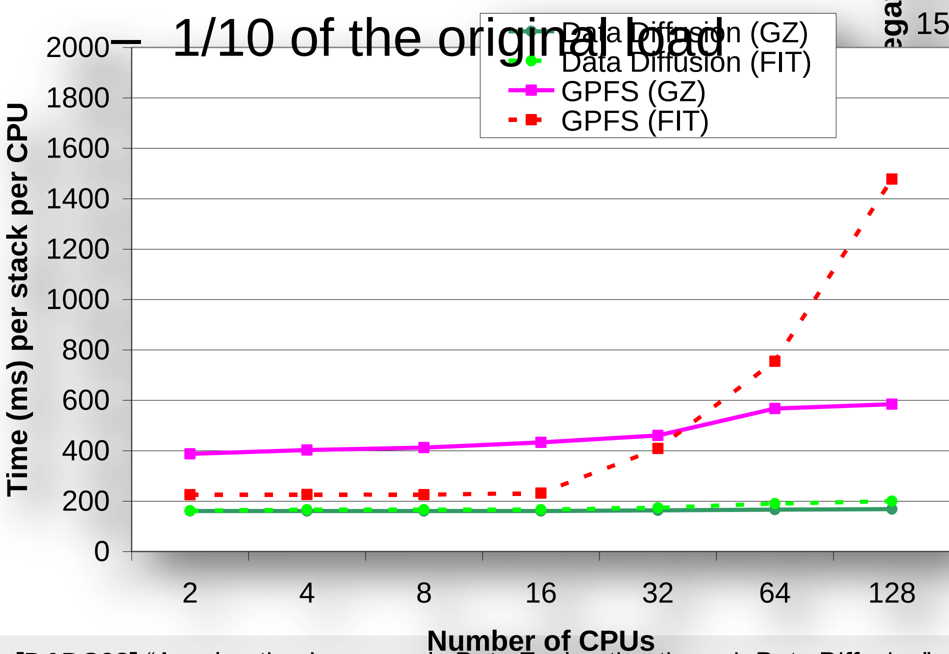


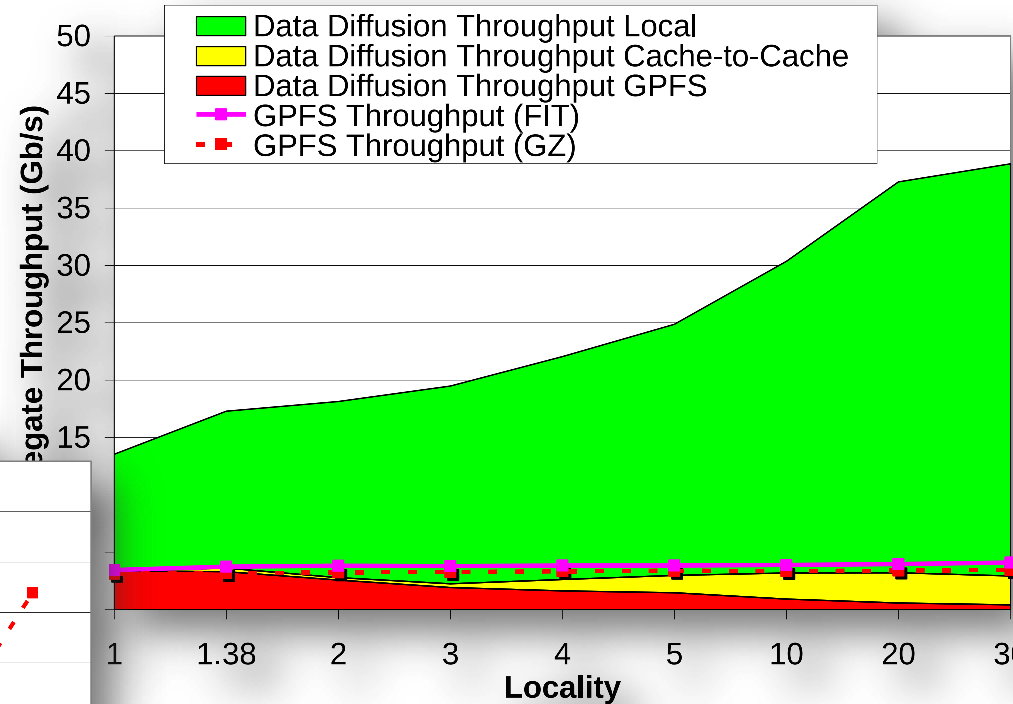
Locality	Number of Objects	Number of Files
1	111700	111700
1.38	154345	111699
2	97999	49000
3	88857	29620
4	76575	19145
5	60590	12120
10	46480	4650
20	40460	2025
30	23695	790

[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"

Outline

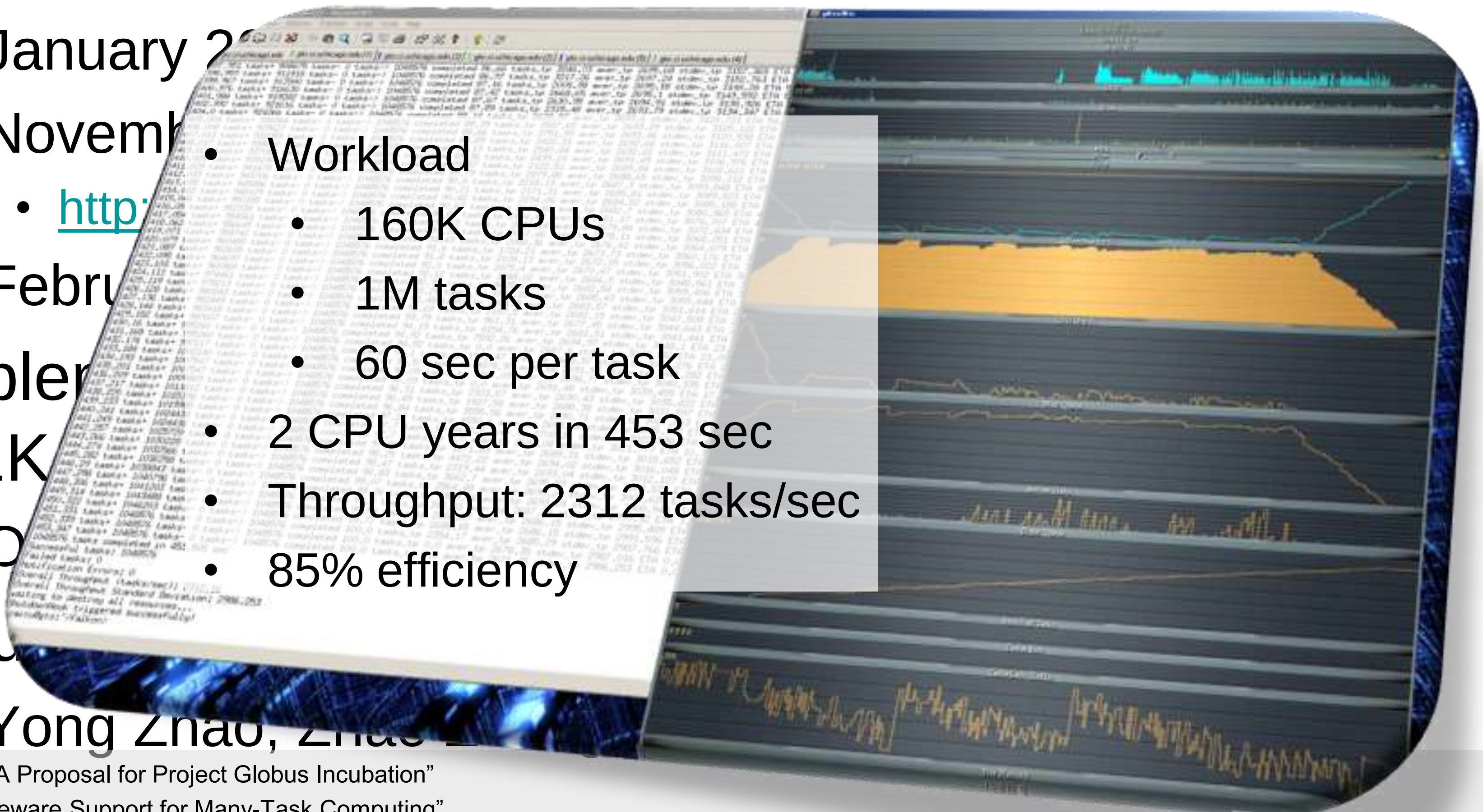
- Overview
- Contributions
- Applications
- **Conclusions**

Conclusions

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

Falkon Project

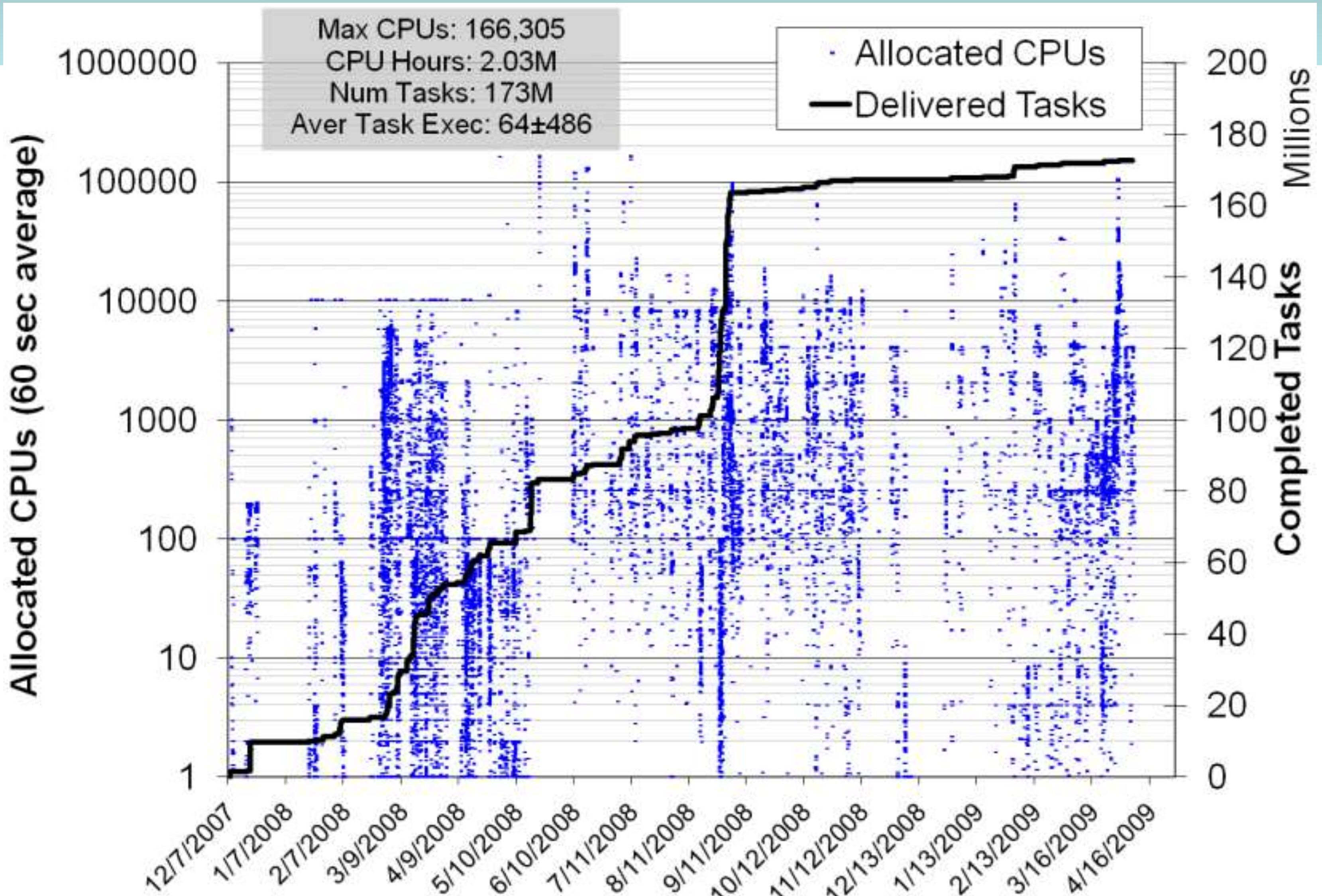
- Falkon is a real system
  - Late 2005: Initial prototype, AstroPortal
  - January 2
  - Novemb
    - http:
  - Febru
- Impler
  (~1K
  - O
- Sou
  - Yong Zhao, Zhao



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

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

Falkon Activity History (16 months)



[CLUSTER10] “Middleware Support for Many-Task Computing”

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
- 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

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

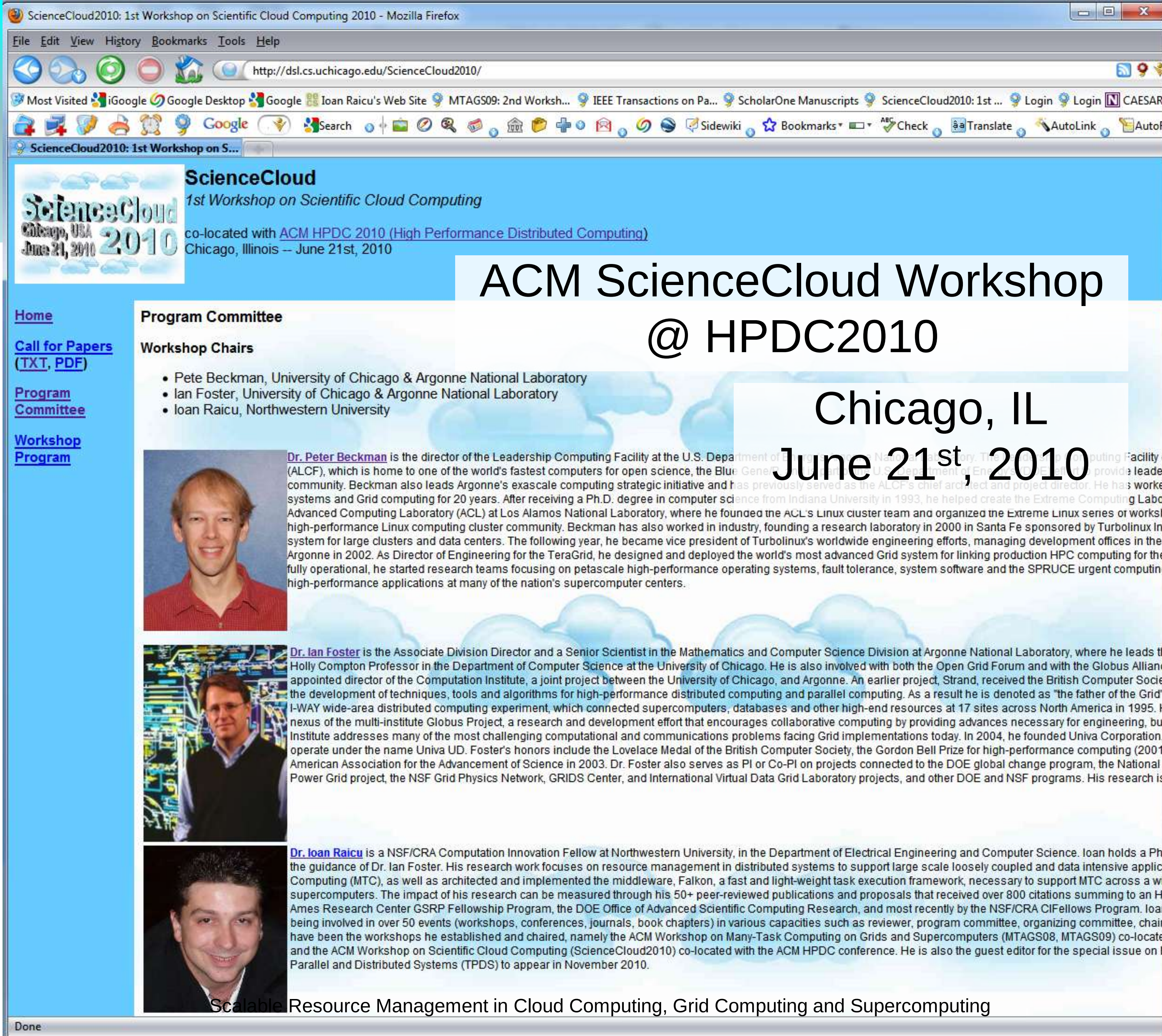
Anonymous

Where can you learn more about Distributed Systems?

- Hot Topics in Distributed Systems: Data-Intensive Computing
  – Northwestern University (EECS495), Instructor
  – http://www.eecs.northwestern.edu/~iraicu/teaching/EECS495-DIC/index.html
- Big Data: Data-intensive Computing Methods, Tools, and Apps
  – University of Chicago (CMSC 34900), Dr. Ian Foster
  – http://dsl-wiki.cs.uchicago.edu/index.php/BigData09
- Networks and Distributed Systems (2006)
  – University of Chicago (CMSC 33300), 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/

ACM ScienceCloud Workshop @ HPDC2010

Chicago, IL
June 21^{st}, 2010



Where can you learn more about Distributed Systems?

- ScienceCloud: ACM Workshop on Scientific Cloud Computing, 2010
- TPDS: IEEE Transactions on Parallel and Distributed Systems, Special Issue on Many-Task Computing, 2010
- HPDC: ACM International Symposium on High Performance Distributed Computing, 2010
- SWF: IEEE International Workshop on Scientific Workflows, 2010
- TG: TeraGrid Conference, 2010
- SC: IEEE/ACM Supercomputing Conference, 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

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