Scalable Resource Management in Clouds and Grids

Ioan Raicu
Distributed Systems Laboratory
Computer Science Department
University of Chicago

In Collaboration with:
Ian Foster, University of Chicago and Argonne National Laboratory
Rick Stevens, University of Chicago and Argonne National Laboratory
Alex Szalay, The Johns Hopkins University
+many more, see "Recent Collaborators" slide…

Accenture Technology Labs
October 24th, 2008

Distributed Systems Laboratory University of Chicago

http://dsl-wiki.cs.uchicago.edu/index.php/Main_Page

- Lead by Dr. Ian Foster
- Research Areas:
  - Distributed systems
  - Grid middleware
  - Grid applications
  - Designing, implementing, and evaluating systems, protocols, and applications
  - Data-intensive scientific computing
- People:
  - 1 faculty (Dr. Ian Foster)
  - 12 students
  - 2 research staff
  - 13 alumnis

Computation Institute
University of Chicago

http://www.ci.uchicago.edu/index.php

- People:
  - Director: Ian Foster
  - 70 faculty and scientists
  - 30 full-time professional staff
  - 14 graduate students
- Focus
  - Deep Supercomputing
  - Data Intensive Computing
  - Next Generation Cybertools
- Many high-impact projects
  - Open Science Grid
  - TeraGrid
  - Globus
  - National Microbial Pathogen Research Center
  - Social Informatics Data Grid
  - Chicago Biomedical Consortium

Math and Computer Science Div. Argonne National Laboratory

http://www.mcs.anl.gov/index.php

- People:
  - Associate Director: Ian Foster
  - 188 staff, researchers, scientists, developers
- Research Areas
  - Algorithms, Software, and Applications
  - Parallel Tools
  - Distributed Systems Research
  - Collaborative and Virtual Environments
  - Computational Science

About Ian Foster

http://www-fp.mcs.anl.gov/~foster/

- Many awards and titles:
  - 1995: "*father of grid computing*"
  - 1996: Globus Toolkit is released
  - 2001: Gordon Bell Award
  - 2002: R&D Magazine awards Globus "most promising new technology" of the year
  - 2003: Infoworld Magazine awards "*top 10 technology inovators*"
  - 2004: co-founder of Univa Corporation
  - 2005: Network World: "*The 50 most powerful people in networking*"
  - 2007: "*top three most influential computer scientists worldwide*" ➔ h-index 67
- Funding
  - NSF: $133M since 1999
  - Others: DOE, NASA, Microsoft, IBM

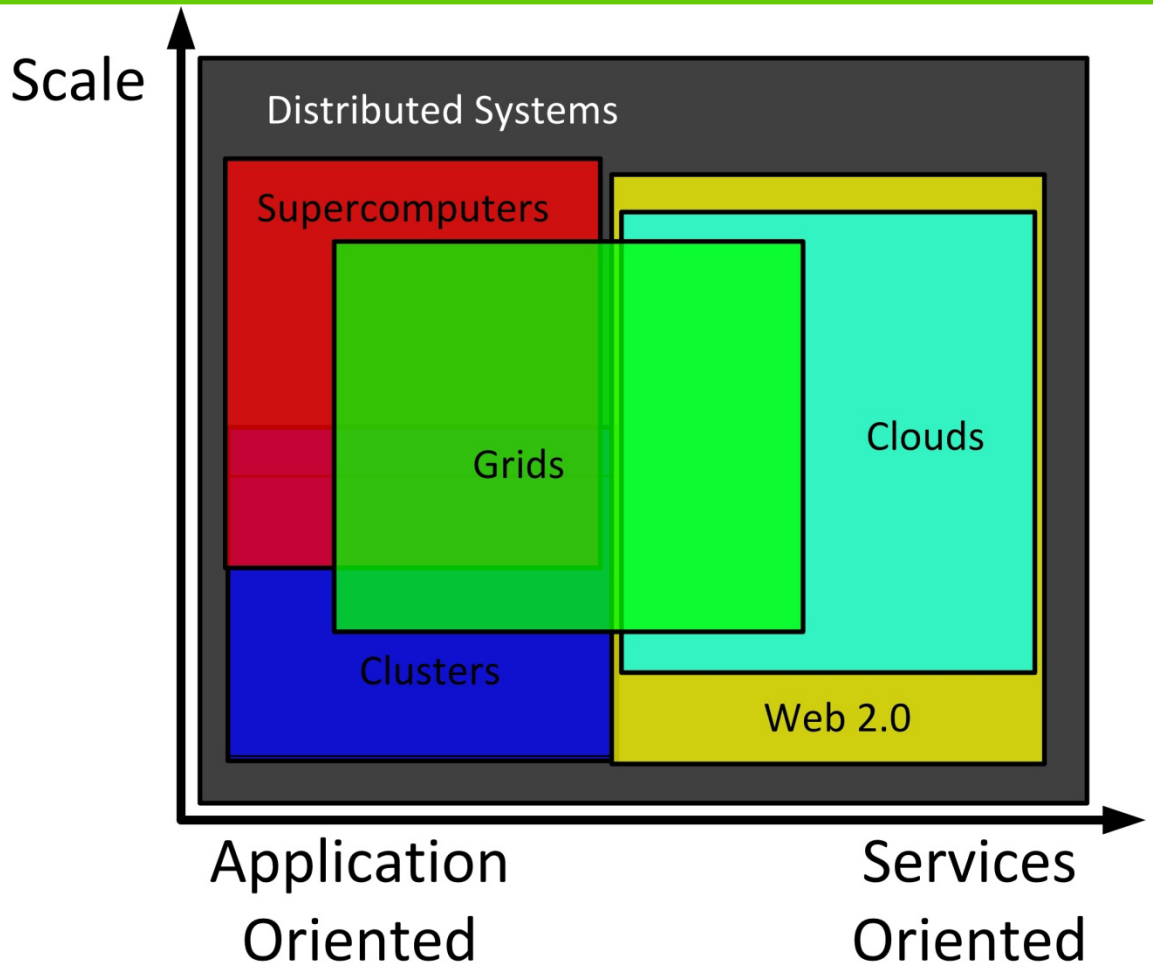
Projects

- GT4: Globus Toolkit 4
  - http://www.globus.org/
- Falkon: a Fast and Light-weight tasK executiON framework
  - http://dev.globus.org/wiki/Incubator/Falkon
- Swift: Fast, Reliable, Loosely Coupled Parallel Computation
  - http://www.ci.uchicago.edu/swift/
- AstroPortal: A Science Gateway for Large-scale Astronomy Data Analysis
  - http://people.cs.uchicago.edu/~iraicu/projects/Falkon/astro_portal.htm
- Haizea: a VM-based Lease Management Architecture
  - http://haizea.cs.uchicago.edu/
- AG: Access Grid
  - http://www.mcs.anl.gov/research/fl/research/index.php?p=proj_detail&id=1
- Collaborative Visualization and the Analysis Pipeline
  - http://www.mcs.anl.gov/research/fl/research/index.php?p=proj_detail&id=28
- Flash Center Visualization
  - http://www.mcs.anl.gov/research/fl/research/index.php?p=proj_detail&id=14
- TeraGrid: Visualization and Data Analysis Resource
  - http://www.mcs.anl.gov/research/fl/research/index.php?p=proj_detail&id=34

Resources

- UChicago CS (50+ machines over the UChicago campus)
  - http://tools.cs.uchicago.edu/find_cs_hosts/find.cgi
- UChicago TeraPort (274 processors)
  - http://teraport.uchicago.edu/
- UC/ANL Cluster (316 processors)
  - http://www.uc.teragrid.org/
- PlanetLab (912 nodes at 470 sites all over the world)
  - http://www.planet-lab.org/
- UChicago PADS (7TF, O(1000-cores))
  - http://www.ci.uchicago.edu/pads/
- ANL SiCortex 5832 (5832 processors)
  - http://www.mcs.anl.gov/hs/hardware/sicortex.php
- Open Science Grid (43K-cores across 80 institutions over the US)
  - http://www.opensciencegrid.org/
- IBM Blue Gene/P Supercomputer at ANL (160K processors)
  - https://wiki.alcf.anl.gov/index.php/Main_Page
- TeraGrid (161K-cores across 11 institutions and 22 systems over the US)
  - http://www.teragrid.org/

Clusters, Grids, Clouds, …



Supercomputing

Highly-tuned computer clusters using commodity processors combined with custom network interconnects

e.g. IBM Blue Gene/P

Grid Computing

A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities

e.g. TeraGrid

Cloud Computing

A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.

e.g. Amazon EC2

Cloud Computing and Grid Computing 360-Degree Compared

- Business model
- Architecture
- Resource management
- Programming model
- Application model
- Security model

Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu. "*Cloud Computing and Grid Computing 360-Degree Compared*", IEEE Grid Computing Environments (GCE08) 2008

Business Model

- Grids:
  - Largest Grids funded by government
  - Largest user-base in academia and government labs to drive scientific computing
  - Project-oriented
- Clouds:
  - Industry (i.e. Amazon) funded the initial Clouds
  - Large user base in common people, small businesses, large businesses, and a bit of openn science research
  - Utility computing

Architecture

- Grids:
  - Application: *Swift, Grid portals (NVO)*
  - Collective layer: *MDS, Condor-G, Nimrod-G*
  - Resource layer: *GRAM, Falkon, GridFTP*
  - Connectivity layer: *Grid Security Infrastructure*
  - Fabric layer: *GRAM, PBS, SGE, LSF, Condor, Falkon*
- Clouds:
  - Application Layer: *Software as a Service (SaaS)*
  - Platform Layer: *Platform as a Service (PaaS)*
  - Unified Resource: *Infrastructure as a Service (IaaS)*
  - Fabric: *IaaS*

Resource Management

- Compute Model
  - batch-scheduled vs. time-shared
- Data Model
  - Data Locality
  - Combining compute and data management
- Virtualization
  - Slow adoption vs. central component
- Monitoring
- Provenance

Programming and Application Model

- Grids:
  - Tightly coupled
    - High Performance Computing (MPI-based)
  - Loosely Coupled
    - High Throughput Computing
    - Workflows
  - Data Intensive
    - Map/Reduce
- Clouds:
  - Loosely Coupled, transactional oriented

Security Model

- Grids
  - Grid Security Infrastructure
  - Stronger, but steeper learning curve and wait time
- Clouds
  - Weaker, can use credit card to gain access, can reset password over plain text email, etc

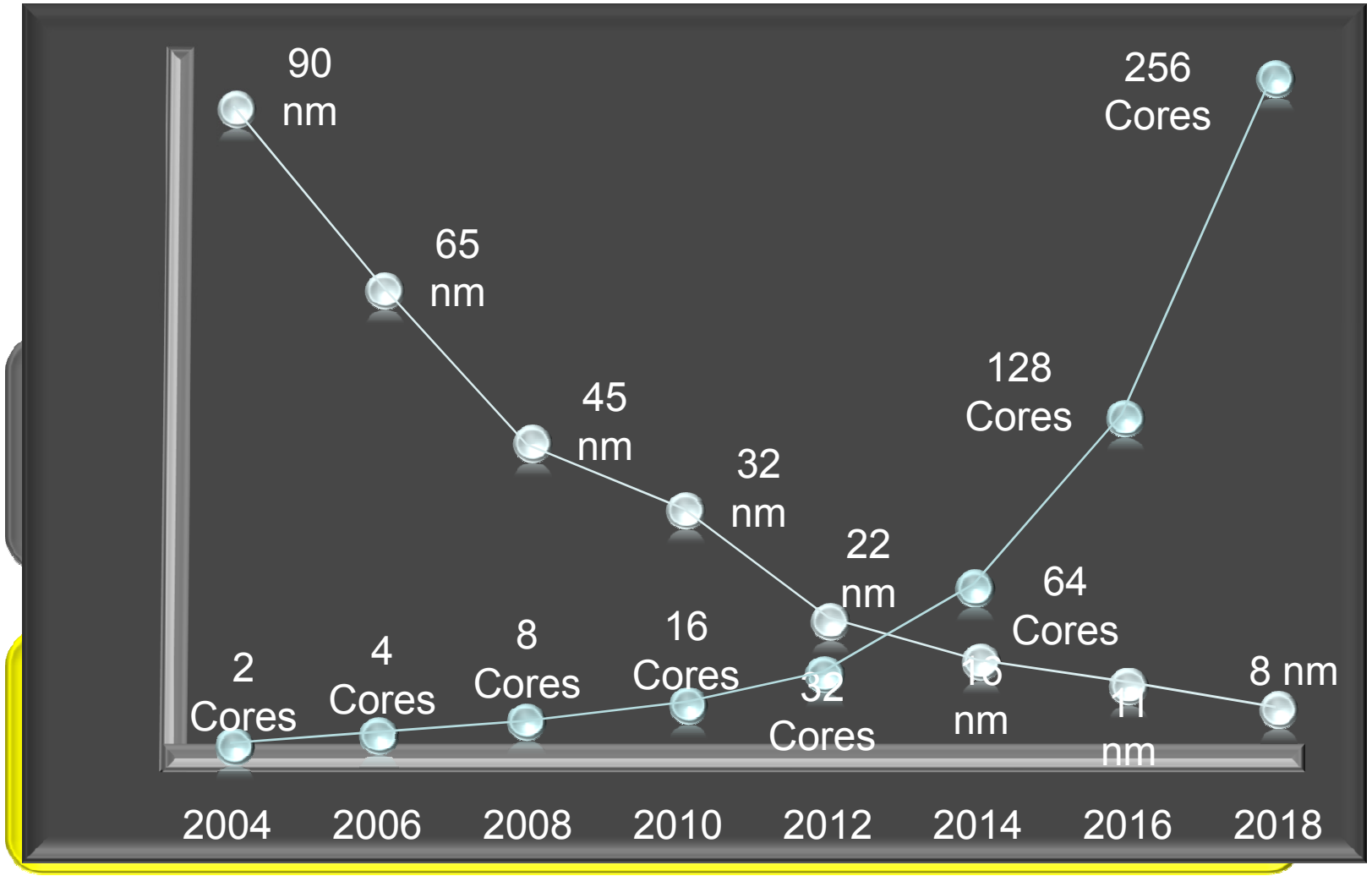
Grids vs. Clouds Conclusion

- Need support for on-demand provisioning
- Define protocols that allow users and service providers to discover, monitor and manage their reservations payments
- Need tools for managing both the underlying resources and the resulting distributed computations
- Need the centralized scale of today's Cloud utilities, and the distribution and interoperability of today's Grid facilities

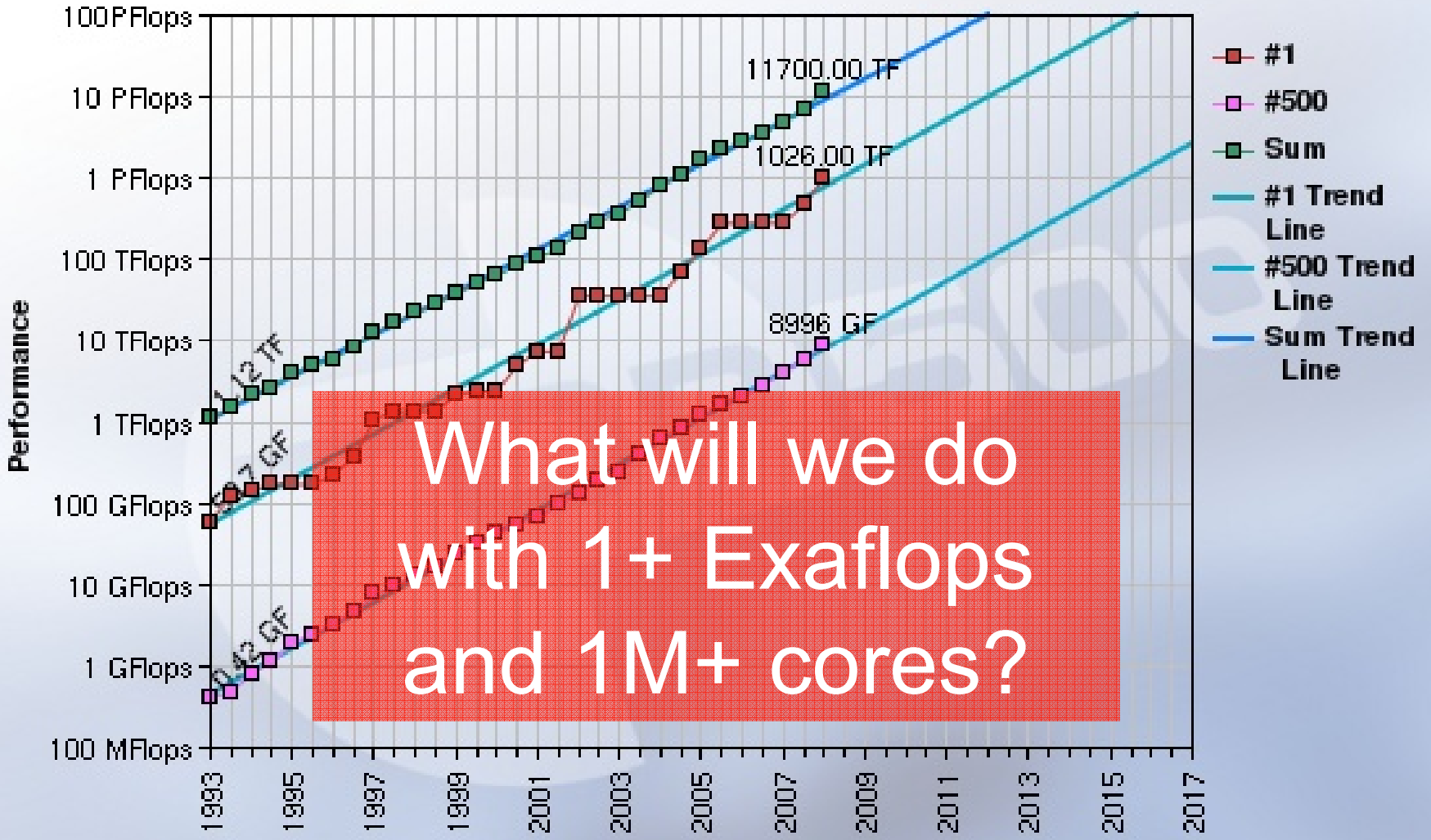
Grids vs. Clouds Conclusion

- **Need support for on-demand provisioning**
- Define protocols that allow users and service providers to discover, monitor and manage their reservations payments
- **Need tools for managing both the underlying resources and the resulting distributed computations**
- Need the centralized scale of today's Cloud utilities, and the distribution and interoperability of today's Grid facilities

Many-Core Growth Rates



Projected Performance Development



What will we do with 1+ Exaflops and 1M+ cores?

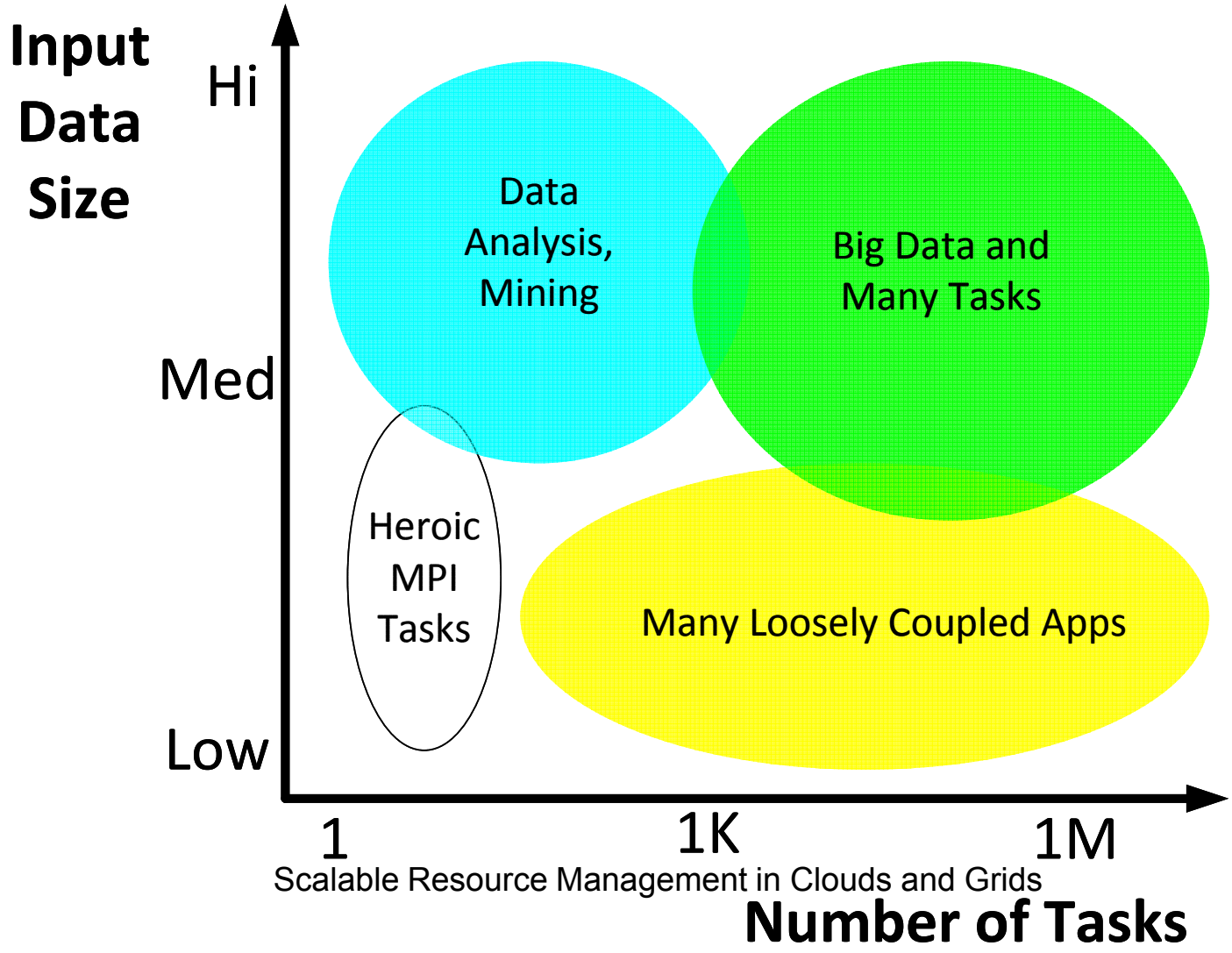
Programming Model Issues

- **Multicore** processors
- Massive **task parallelism**
- Massive **data parallelism**
- Integrating **black box applications**
- Complex **task dependencies** (task graphs)
- **Failure**, and other execution management issues
- **Dynamic task graphs**
- Documenting **provenance** of data products
- **Data management**: input, intermediate, output
- **Dynamic data access** involving large amounts of data

Programming Model Issues

- **Multicore** processors
- Massive **task parallelism**
- Massive **data parallelism**
- Integrating **black box applications**
- Complex **task dependencies** (task graphs)
- **Failure**, and other execution management issues
- **Dynamic task graphs**
- Documenting **provenance** of data products
- **Data management**: input, intermediate, output
- **Dynamic data access** involving large amounts of data

Problem Types



An Incomplete and Simplistic View of Programming Models and Tools

Single task, modest data
MPI, etc…

Many Tasks
DAGMan+Pegasus
Karajan+Swift+Falkon

Much Data
MapReduce/Hadoop
Dryad

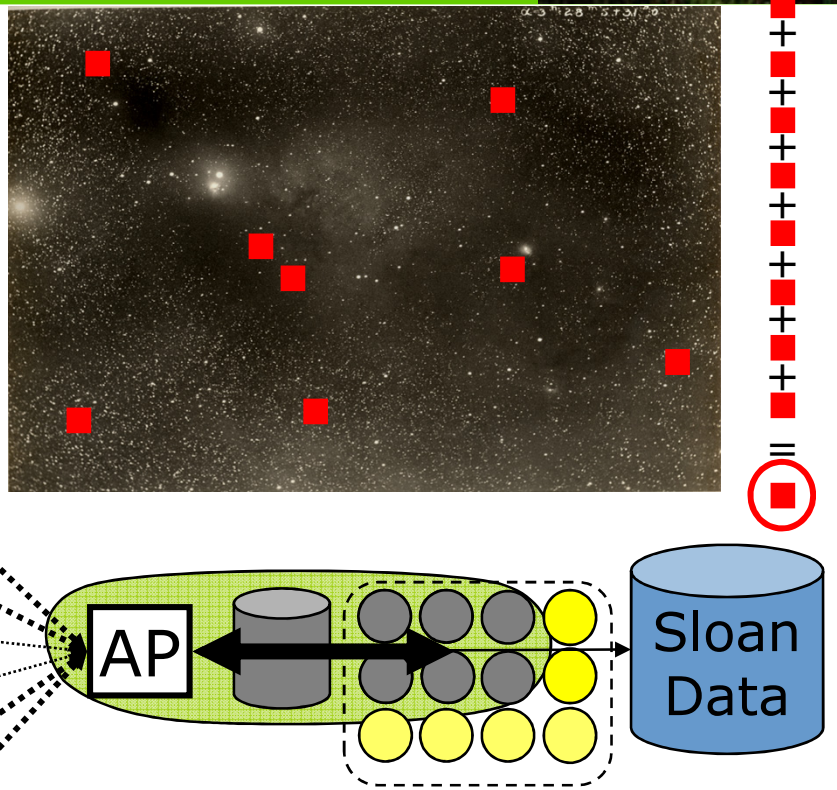
Complex Tasks, Much Data
Dryad, Pig, Sawzall
Swift+Falkon (using data diffusion)

MTC: Many Task Computing

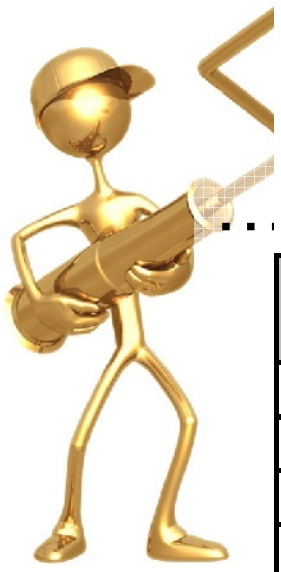
- Loosely coupled applications
  - High-performance computations comprising of multiple distinct activities, coupled via file system operations or message passing
  - Emphasis on using many resources over short time periods
  - Tasks can be:
    - small or large, independent and dependent, uniprocessor or multiprocessor, compute-intensive or data-intensive, static or dynamic, homogeneous or heterogeneous, loosely or tightly coupled, large number of tasks, large quantity of computing, and large volumes of data…

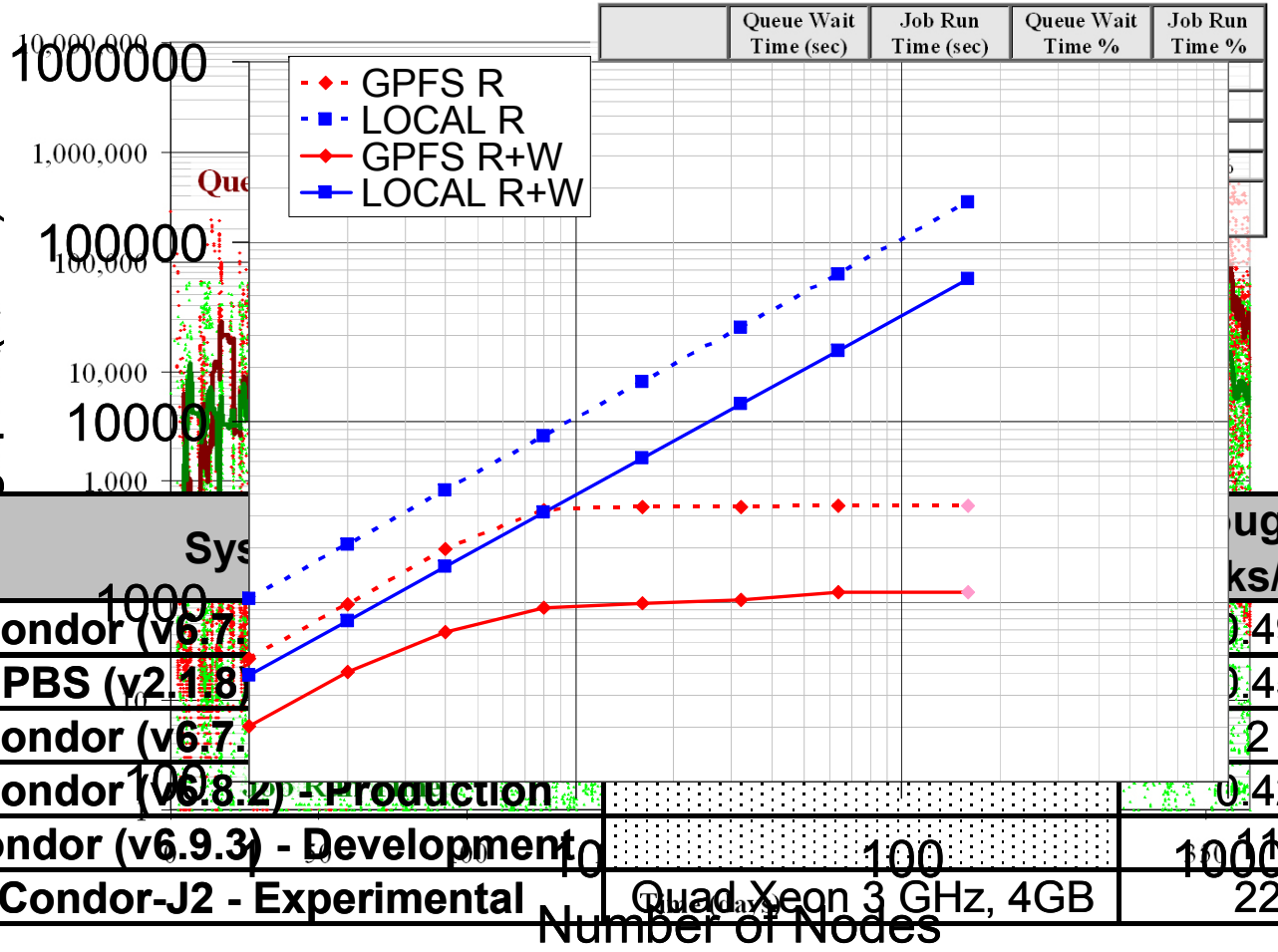
Motivating Example: AstroPortal Stacking Service

- 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



Obstacles and Solutions





Sys	ughput ks/sec)
Condor (v6.7.	.49
PBS (v2.1.8)	.45
Condor (v6.7.	2
Condor (v6.8.2) - Production	0.42
Condor (v6.9.3) - Development	11
Condor-J2 - Experimental	22

ght

Hypothesis

"Significant performance improvements can be obtained in the analysis of large dataset by leveraging information about data analysis workloads rather than individual data analysis tasks."

- **Important concepts related to the hypothesis**
  - **Workload**: a complex query (or set of queries) decomposable into simpler tasks to answer broader analysis questions
  - **Data locality** is crucial to the efficient use of large scale distributed systems for scientific and data-intensive applications
  - Allocate computational and caching storage resources, **co-scheduled** to optimize workload performance

Abstract Model

- AMDASK: An Abstract Model for DAta-centric taSK farms
  - Task Farm: A common parallel pattern that drives independent computational tasks
- Models the efficiency of data analysis workloads for the split/merge class of applications
- Captures data diffusion properties
  - Resources are acquired in response to demand
  - Data and applications diffuse from archival storage to new resources
  - Resource “caching” allows faster responses to subsequent requests
  - Resources are released when demand drops
  - Considers both data and computations to optimize performance

AMDASK: Base Definitions

- **Data Stores:** Persistent & Transient
  - Store capacity, load, ideal bandwidth, available bandwidth
- **Data Objects:**
  - Data object size, *data object's storage location(s),* copy time
- **Transient resources:** compute speed, resource state
- **Task:** application, input/output data

AMDASK: Execution Model Concepts

- Dispatch Policy
  - next-available, first-available, max-compute-util, max-cache-hit
- Caching Policy
  - random, FIFO, LRU, LFU
- Replay policy
- Data Fetch Policy
  - Just-in-Time, Spatial Locality
- Resource Acquisition Policy
  - one-at-a-time, additive, exponential, all-at-once, optimal
- Resource Release Policy
  - distributed, centralized

AMDASK: Performance Efficiency Model

- B: Average Task Execution Time:
  - K: Stream of tasks
  - μ(k): Task k execution time

$$B = \frac{1}{|K|}\sum_{k \in K} \mu(\kappa)$$

- Y: Average Task Execution Time with Overheads:
  - o(k): Dispatch overhead
  - ς(δ,τ): Time to get data

$$Y = \begin{cases} \frac{1}{|K|}\sum_{\kappa \in K}[\mu(\kappa)+o(\kappa)], & \delta \in \phi(\tau), \delta \in \Omega \\ \frac{1}{|K|}\sum_{\kappa \in K}[\mu(\kappa)+o(\kappa)+\zeta(\delta,\tau)], & \delta \notin \phi(\tau), \delta \in \Omega \end{cases}$$

- V: Workload Execution Time:
  - A: Arrival rate of tasks
  - T: Transient Resources

$$V = \max\left(\frac{B}{|T|}, \frac{1}{A}\right) * |K|$$

- W: Workload Execution Time with Overheads

$$W = \max\left(\frac{Y}{|T|}, \frac{1}{A}\right) * |K|$$

AMDASK: Performance Efficiency Model

- **Efficiency**

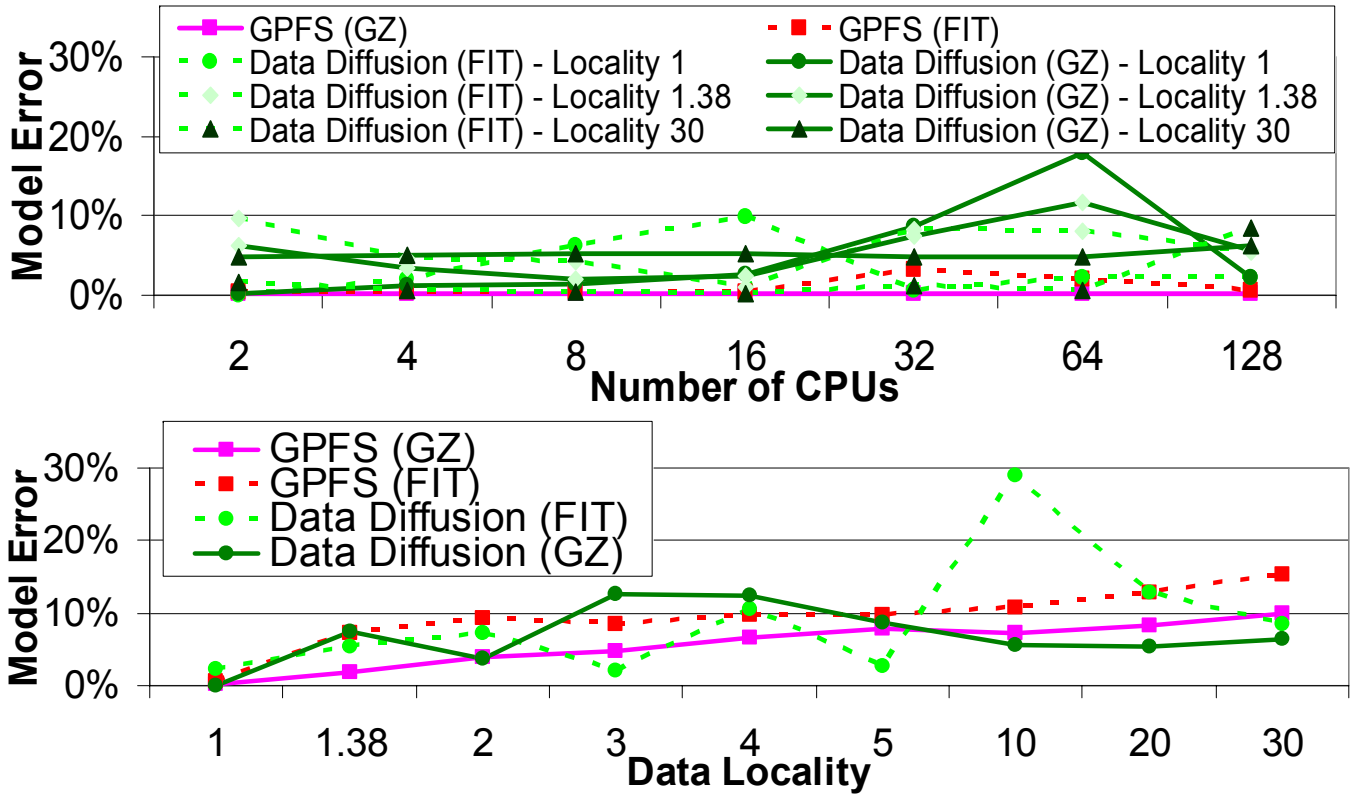
$$\mathrm{E} = \frac{V}{W} \longrightarrow E = \begin{cases} 1, & \frac{Y}{|T|} \le \frac{1}{A} \\ \max\left(\frac{B}{Y}, \frac{|\mathrm{T}|}{\mathrm{A} * Y}\right), & \frac{Y}{|T|} > \frac{1}{A} \end{cases}$$

- **Speedup**

$$S = E^* |T|$$

- **Optimizing Efficiency**
  - Easy to maximize either efficiency or speedup independently
  - Harder to maximize both at the same time
    - Find the smallest number of *transient resources* |T| while maximizing speedup*efficiency

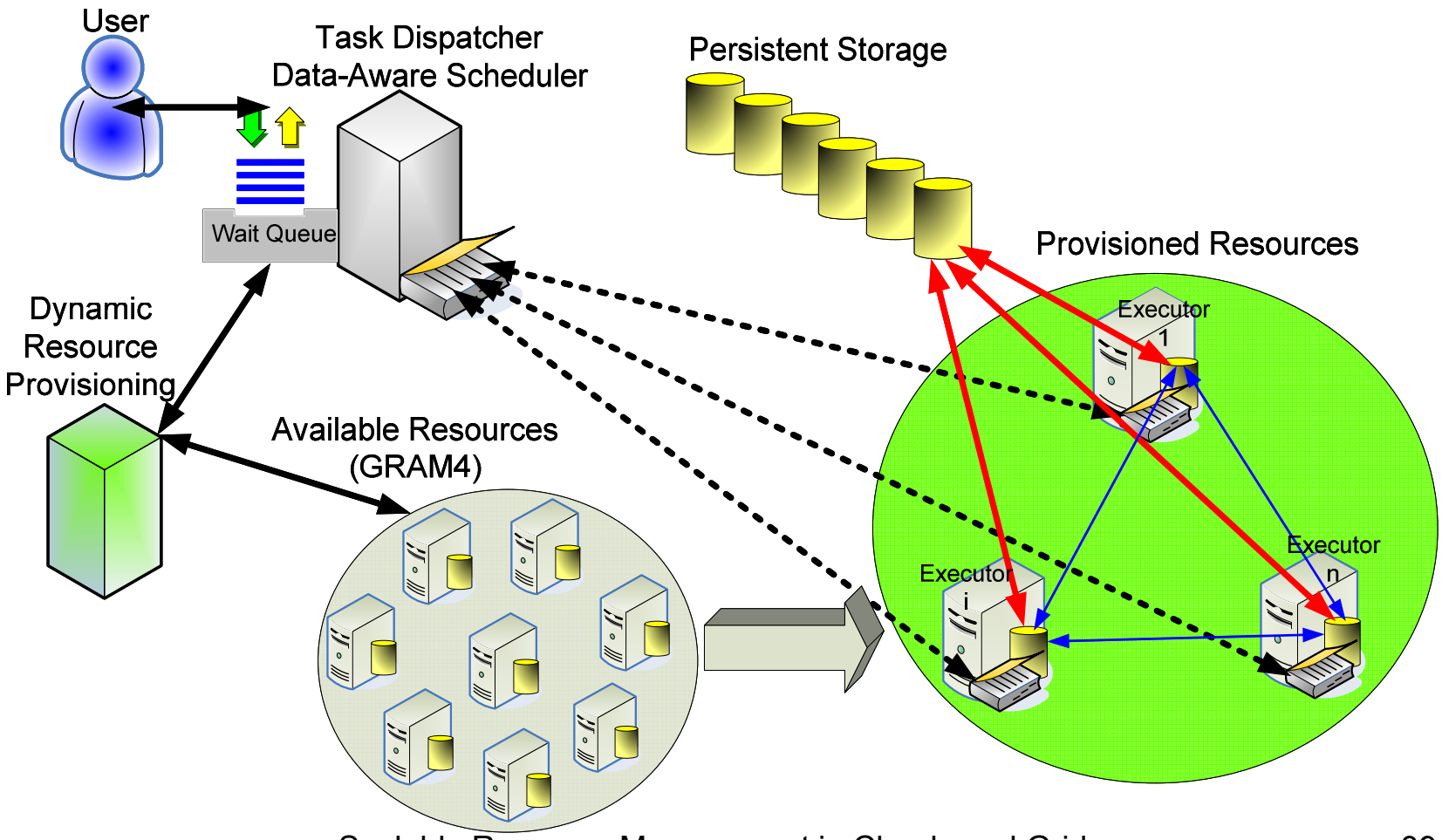
Model Validation



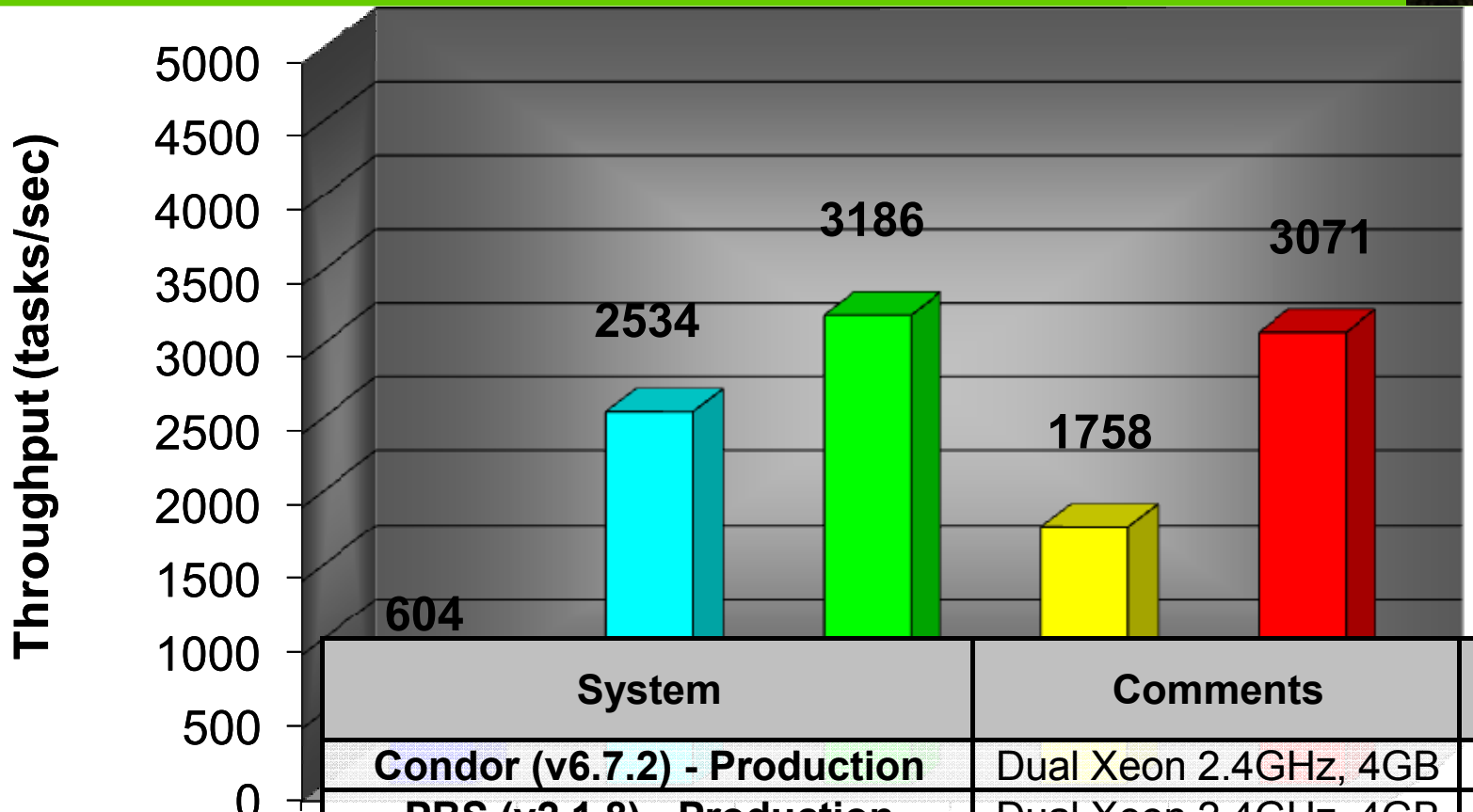
Falkon: a Fast and Light-weight tasK executiON framework

- ***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

Falkon Overview

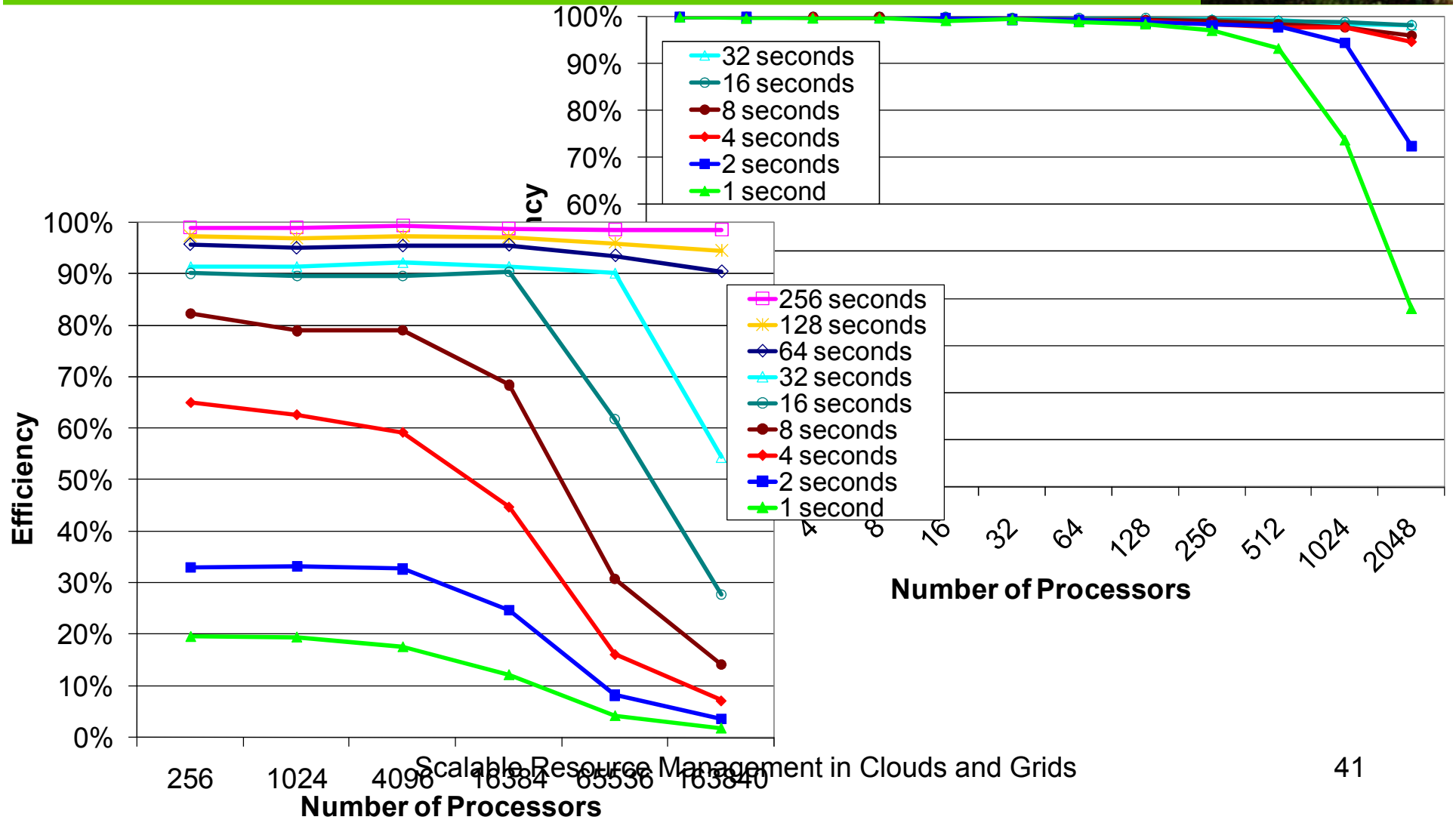


Dispatch Throughput



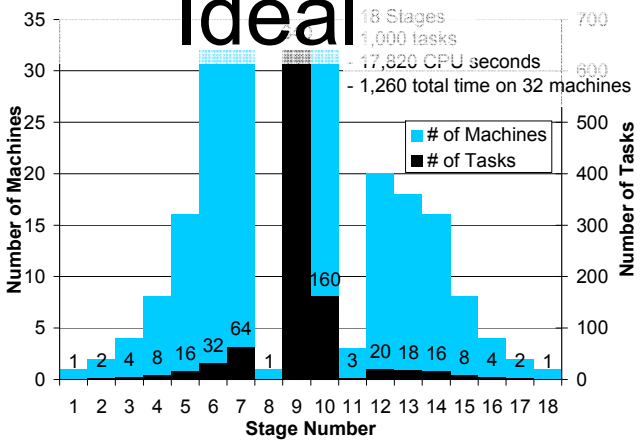
System	Comments	Throughput (tasks/sec)
Condor (v6.7.2) - Production	Dual Xeon 2.4GHz, 4GB	0.49
PBS (v2.1.8) - Production	Dual Xeon 2.4GHz, 4GB	0.45
Condor (v6.7.2) - Production	Quad Xeon 3 GHz, 4GB	2
Condor (v6.8.2) - Production		0.42
Condor (v6.9.3) - Development		11
Condor-J2 - Experimental	Quad Xeon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

Efficiency

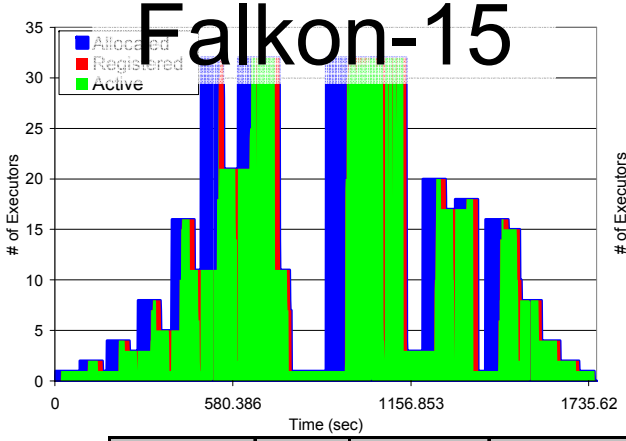

100%
90%
80%
70%
60%
32 seconds
16 seconds
8 seconds
4 seconds
2 seconds
1 second
4
8
16
32
64
128
256
512
1024
2048
Number of Processors
Efficiency
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
256 seconds
128 seconds
64 seconds
32 seconds
16 seconds
8 seconds
4 seconds
2 seconds
1 second
256
1024
4096
16384
65536
163840
Number of Processors

Resource Provisioning

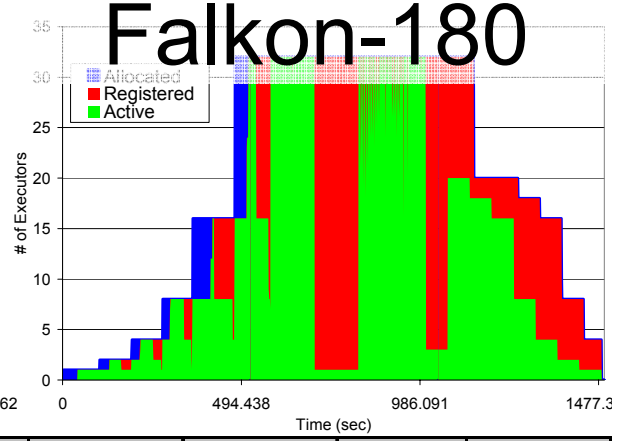
Ideal



Falkon-15



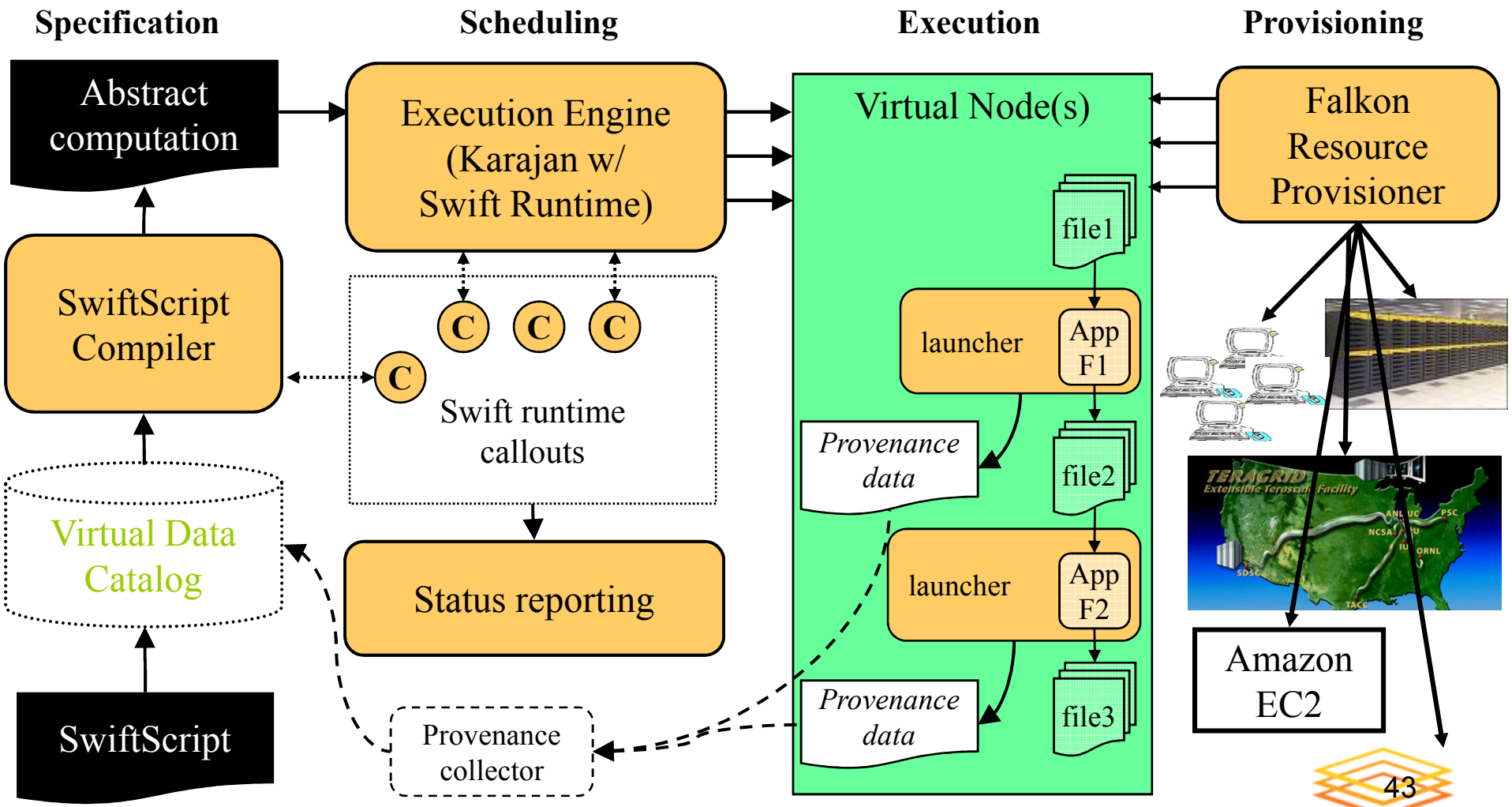
Falkon-180



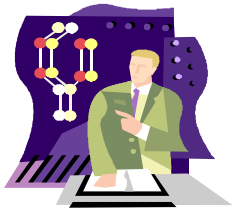
- End-to-end execution time:
  - 1260 sec in ideal case
  - 4904 sec → 1276 sec
- Average task queue time:
  - 42.2 sec in ideal case
  - 611 sec → 43.5 sec
- Trade-off:
  - Resource Utilization for Execution Efficiency

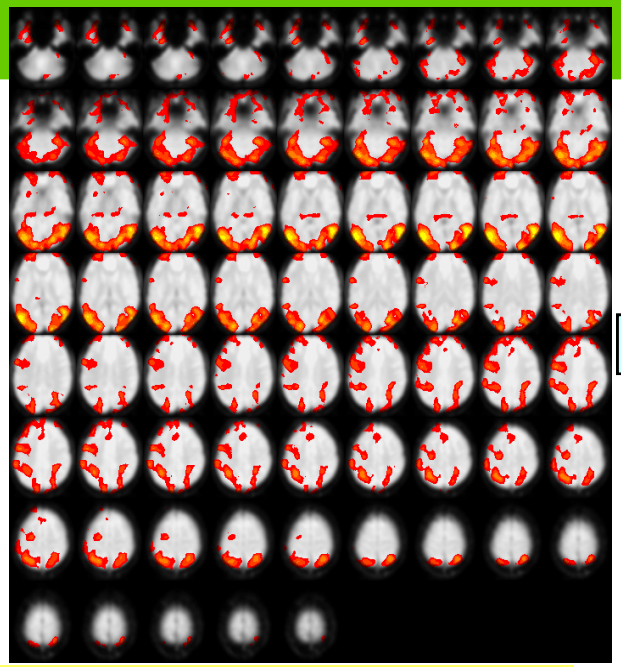
	GRAM +PBS	Falkon-15	Falkon-60	Falkon-120	Falkon-180	Falkon-∞	Ideal (32 nodes)
Queue Time (sec)	611.1	87.3	83.9	74.7	44.4	43.5	42.2
Execution Time (sec)	56.5	17.9	17.9	17.9	17.9	17.9	17.8
Execution Time %	8.5%	17.0%	17.6%	19.3%	28.7%	29.2%	29.7%
	GRAM +PBS	**Falkon-15**	**Falkon-60**	**Falkon-120**	**Falkon-180**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

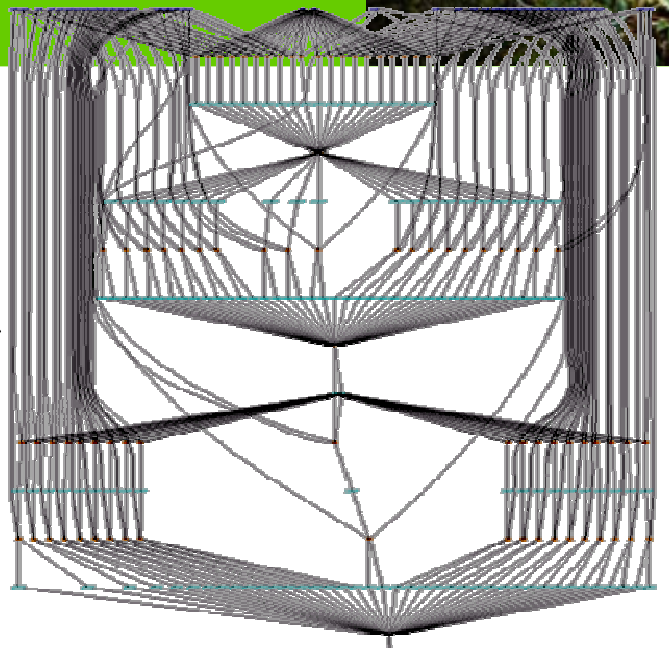
Swift Architecture

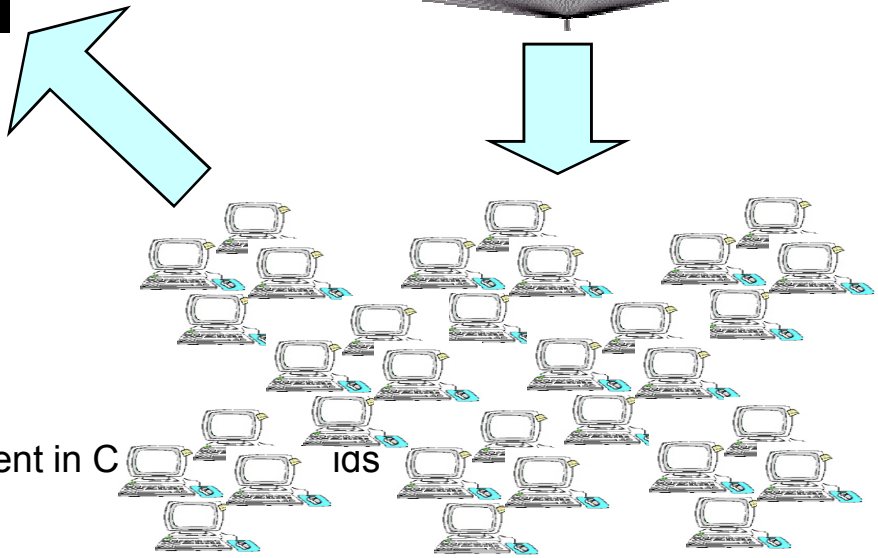


Functional MRI (fMRI)





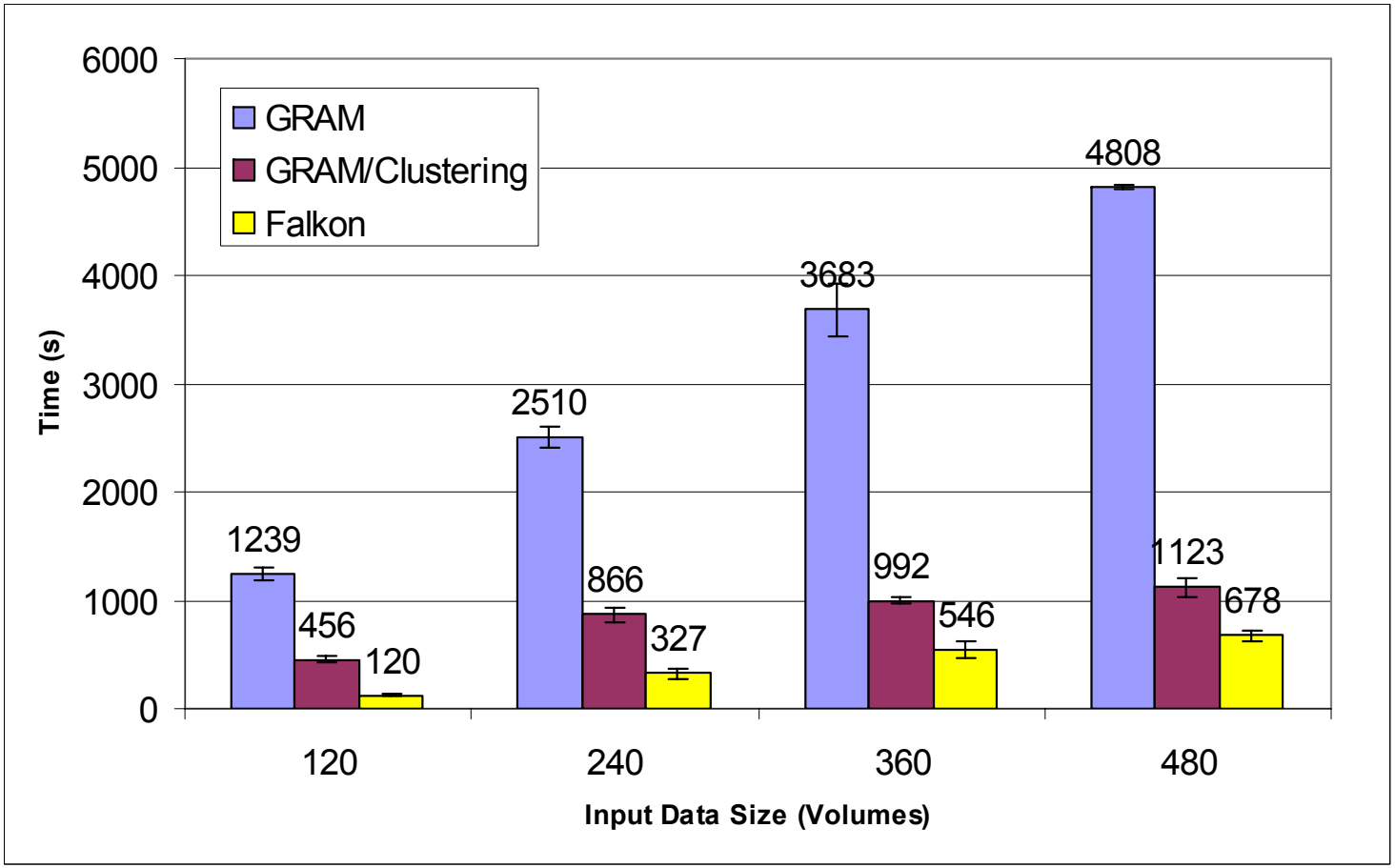


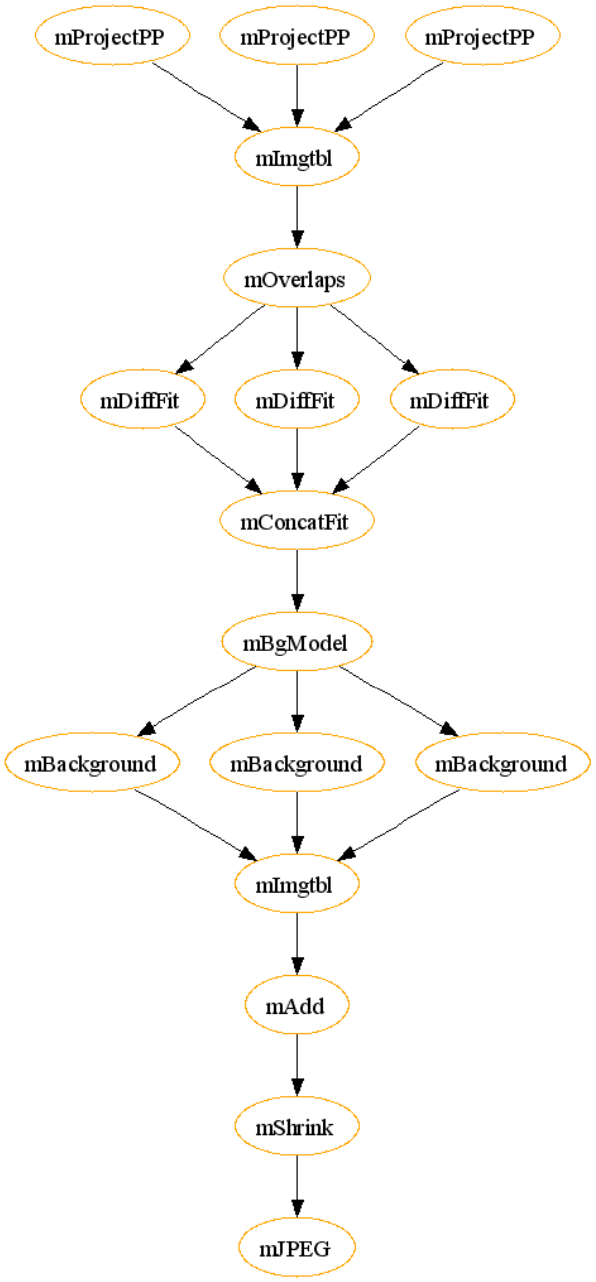


- Wide range of analyses
  - Testing, interactive analysis, production runs
  - Data mining
  - Parameter studies

Completed Milestones: fMRI Application

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



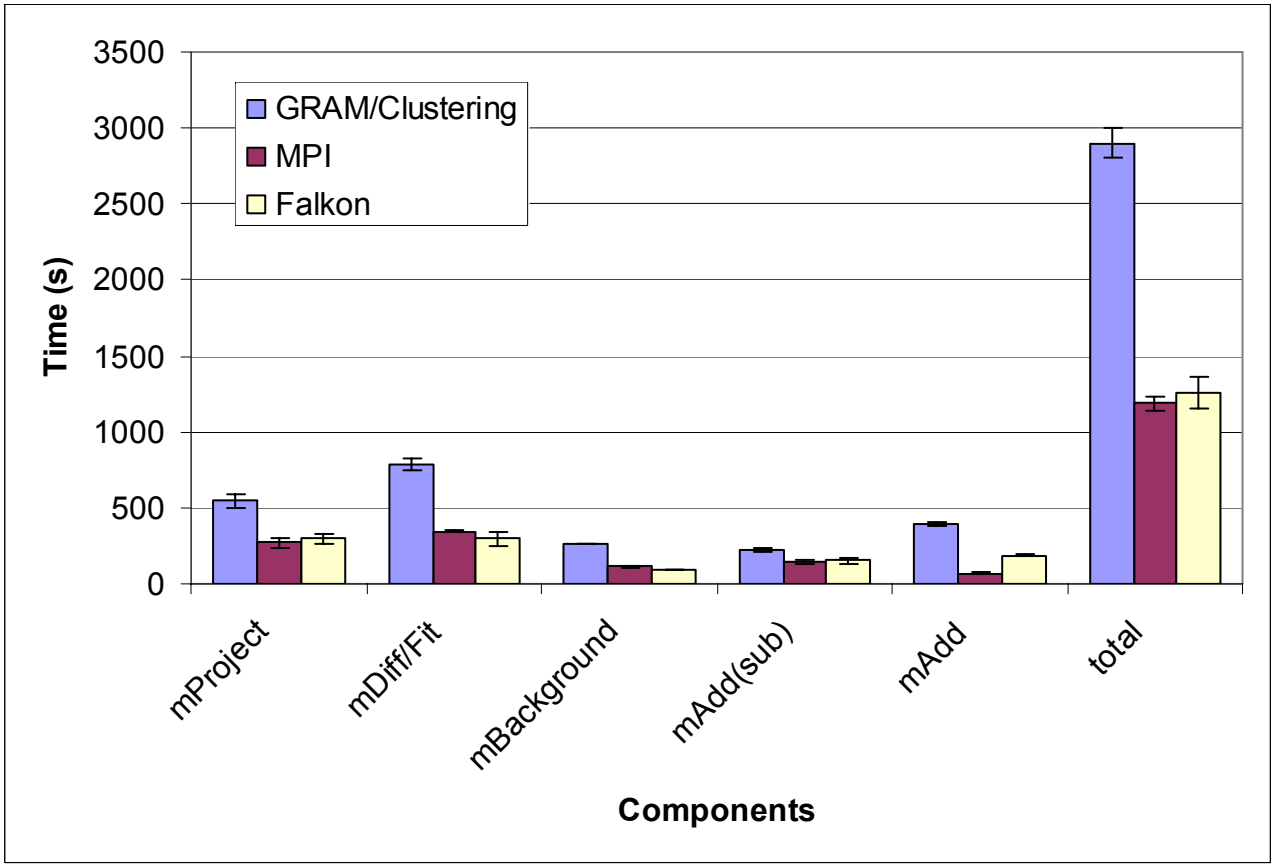


Montage

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

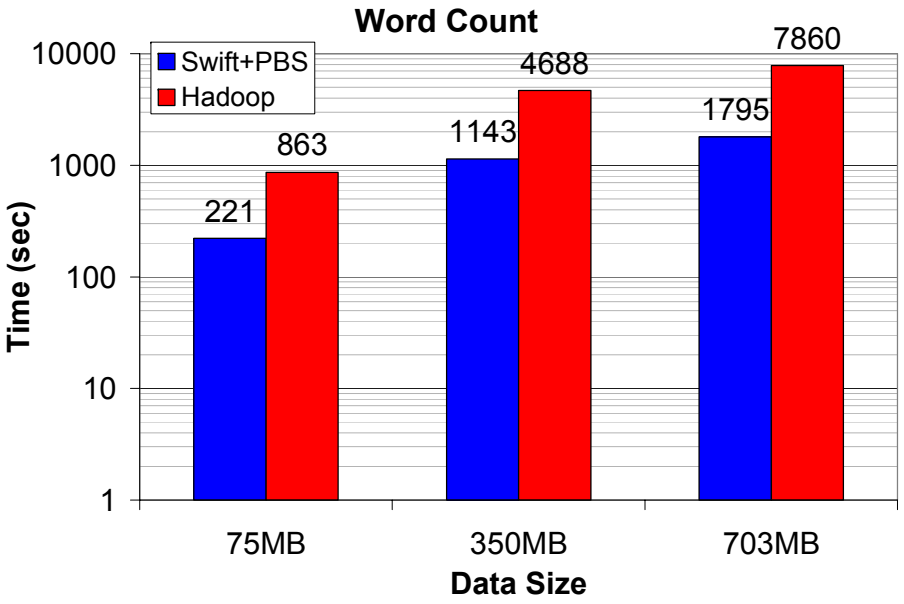
Completed Milestones: Montage Application

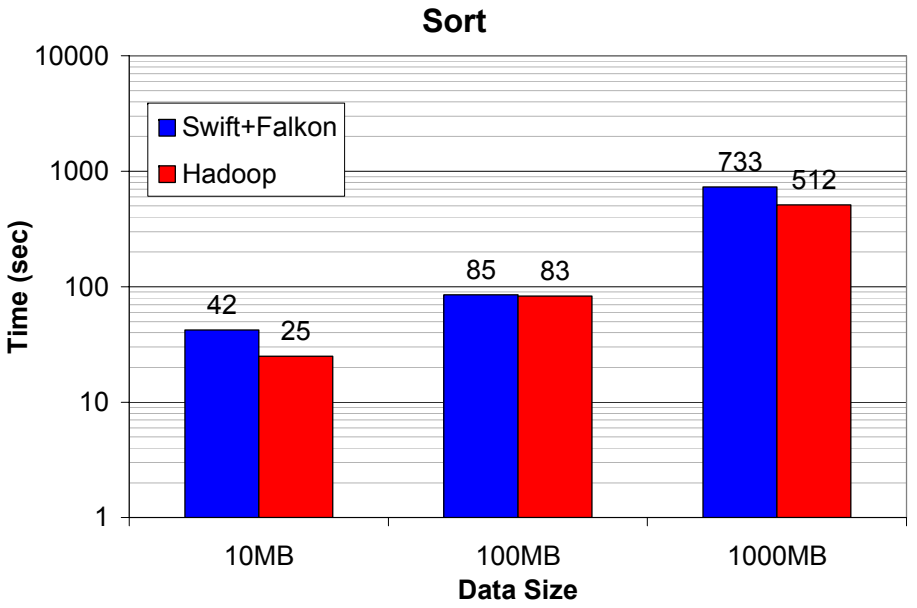
- GRAM/Clustering vs. Falkon: 57% lower application run time
- MPI* vs. Falkon: 4% higher application run time
- * MPI should be lower bound



Hadoop vs. Swift

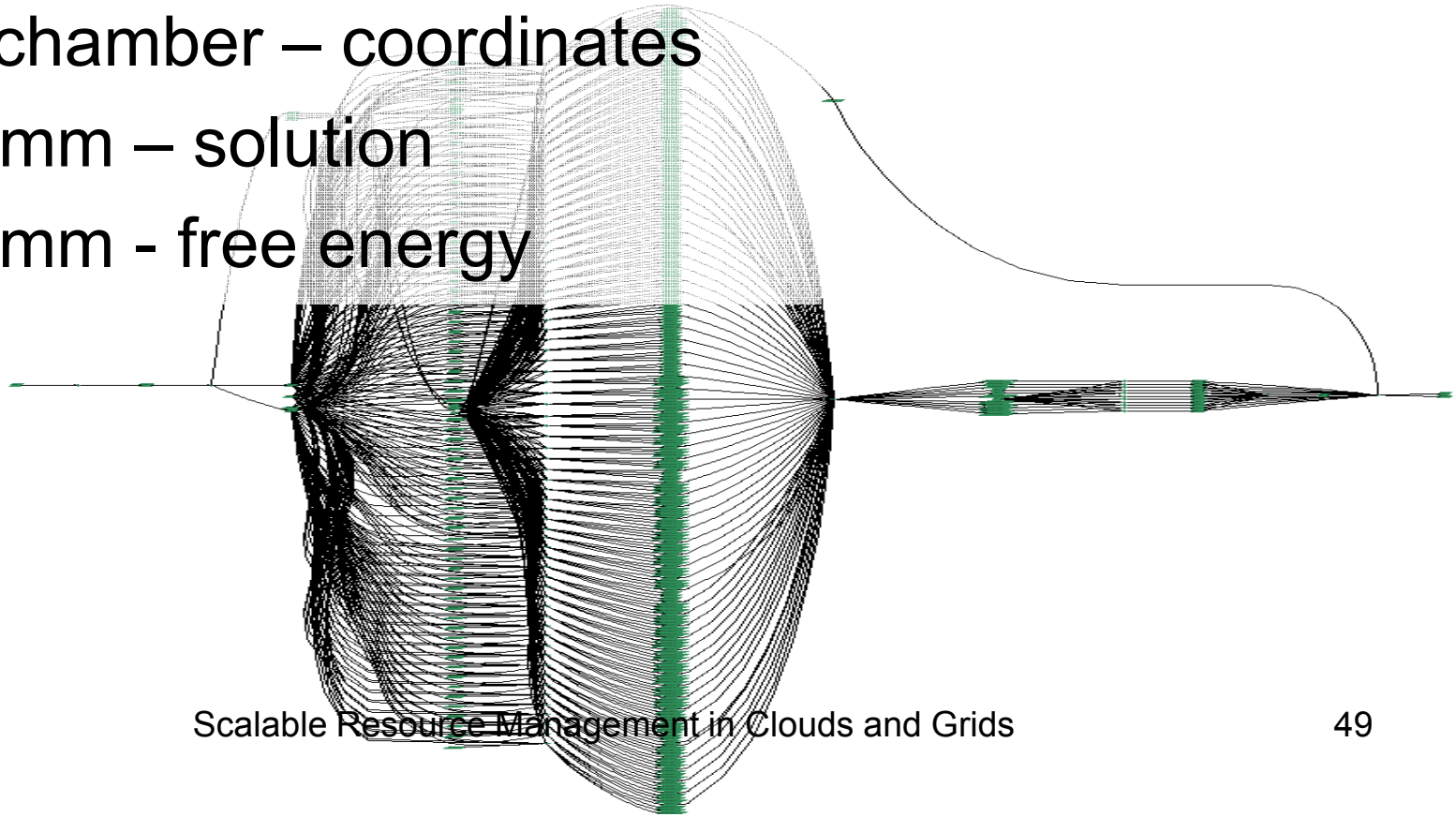
- Classic benchmarks for MapReduce
  - Word Count
  - Sort
- Swift performs similar or better than Hadoop (on 32 processors)





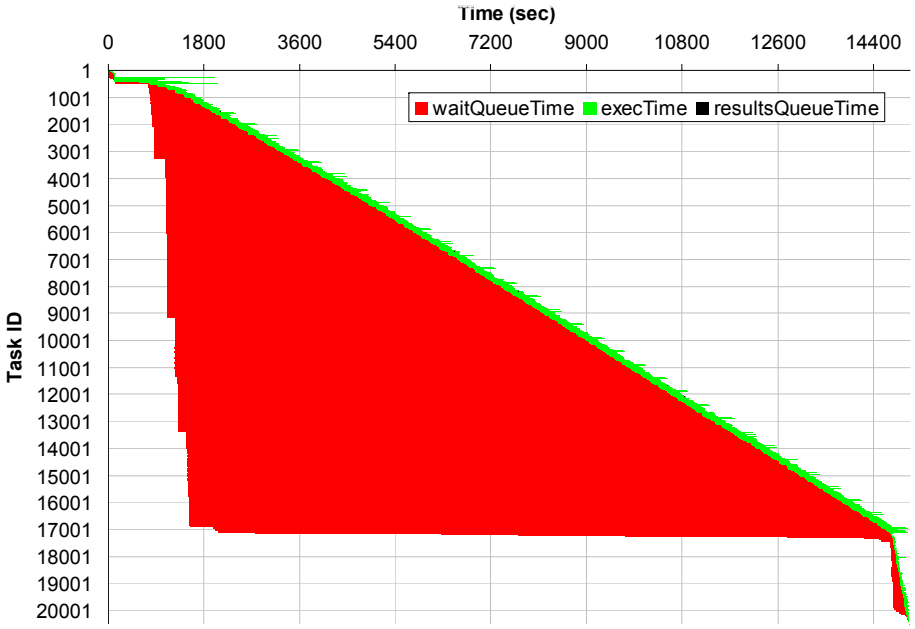
Molecular Dynamics

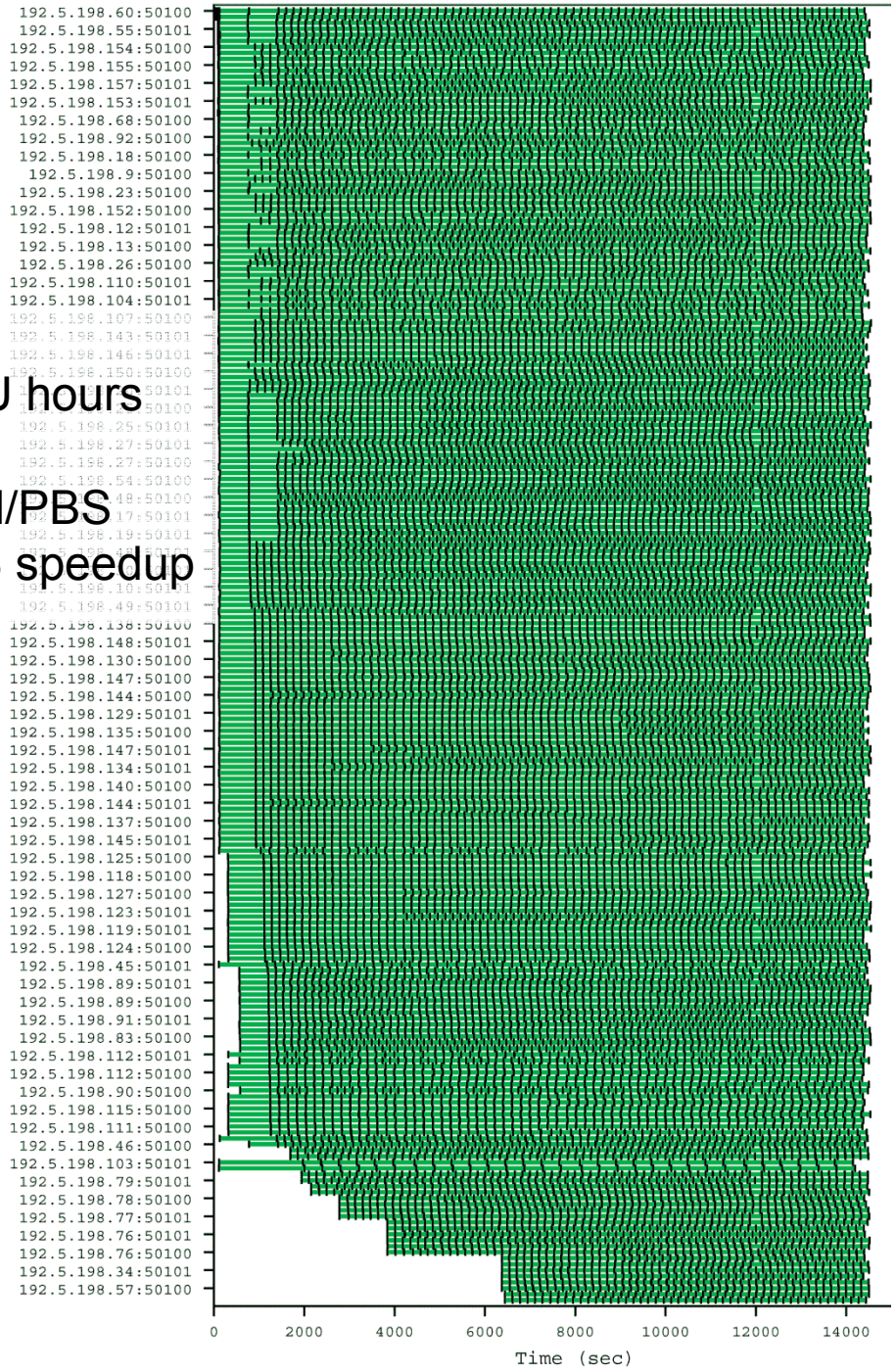
- Determination of free energies in aqueous solution
  - Antechamber – coordinates
  - Charmm – solution
  - Charmm - free energy



MolDyn Application

- 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

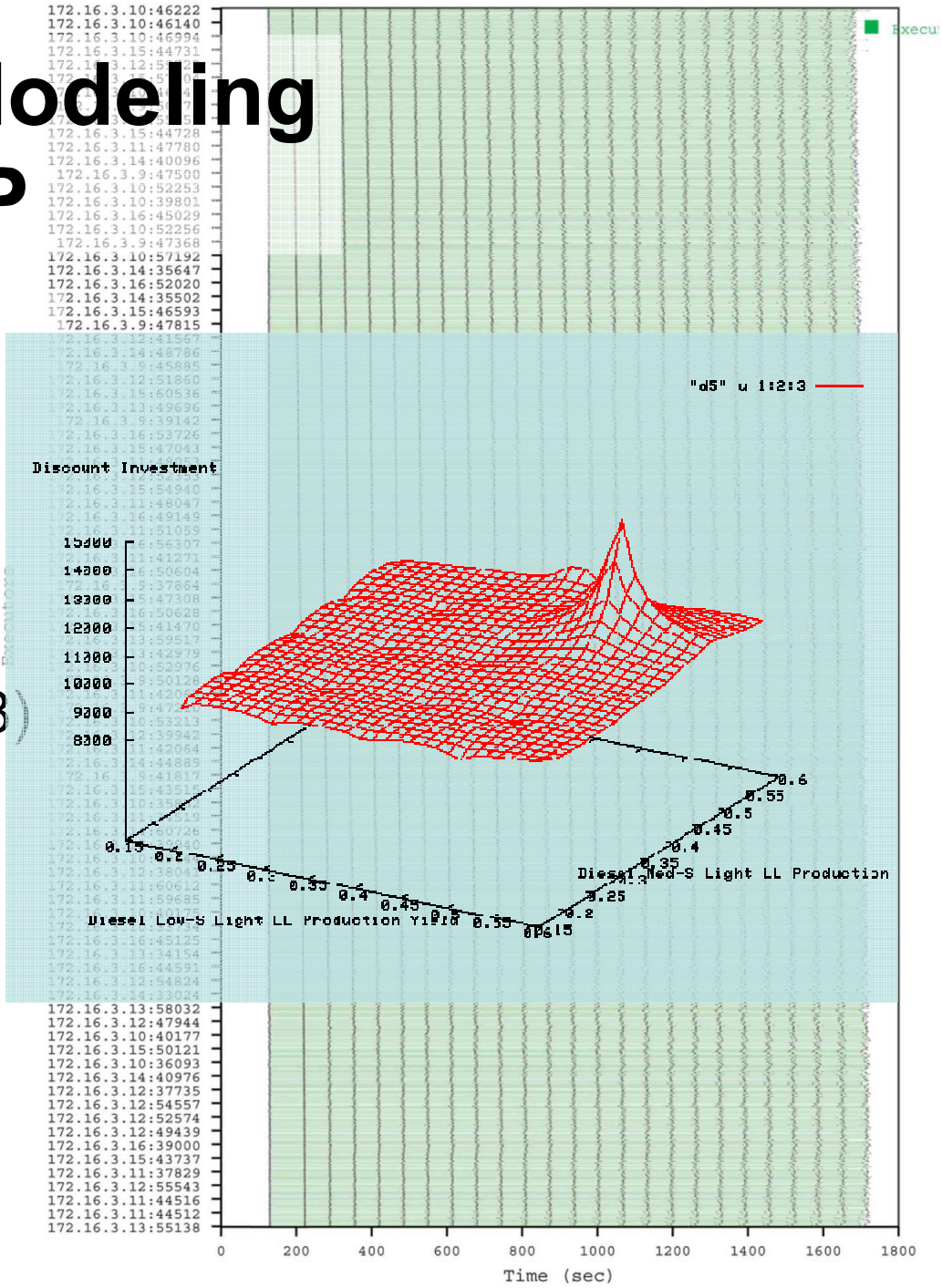




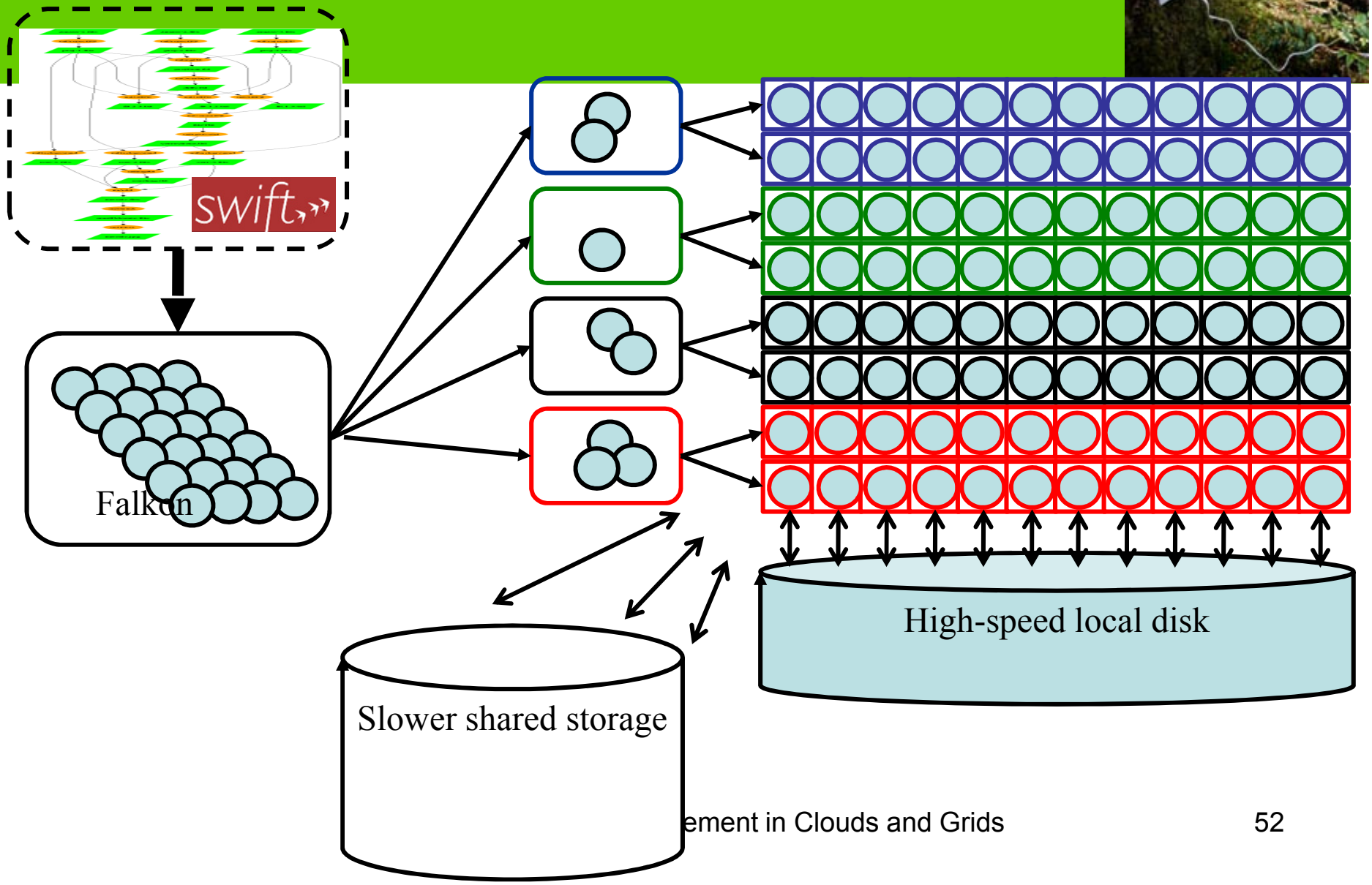
Scalable Resource Manage

MARS Economic Modeling on IBM BG/P

- CPU Cores: 2048
- Tasks: 49152
- Micro-tasks: 7077888
- Elapsed time: 1601 secs
- CPU Hours: 894
- Speedup: 1993X (ideal 2048)
- Efficiency: 97.3%

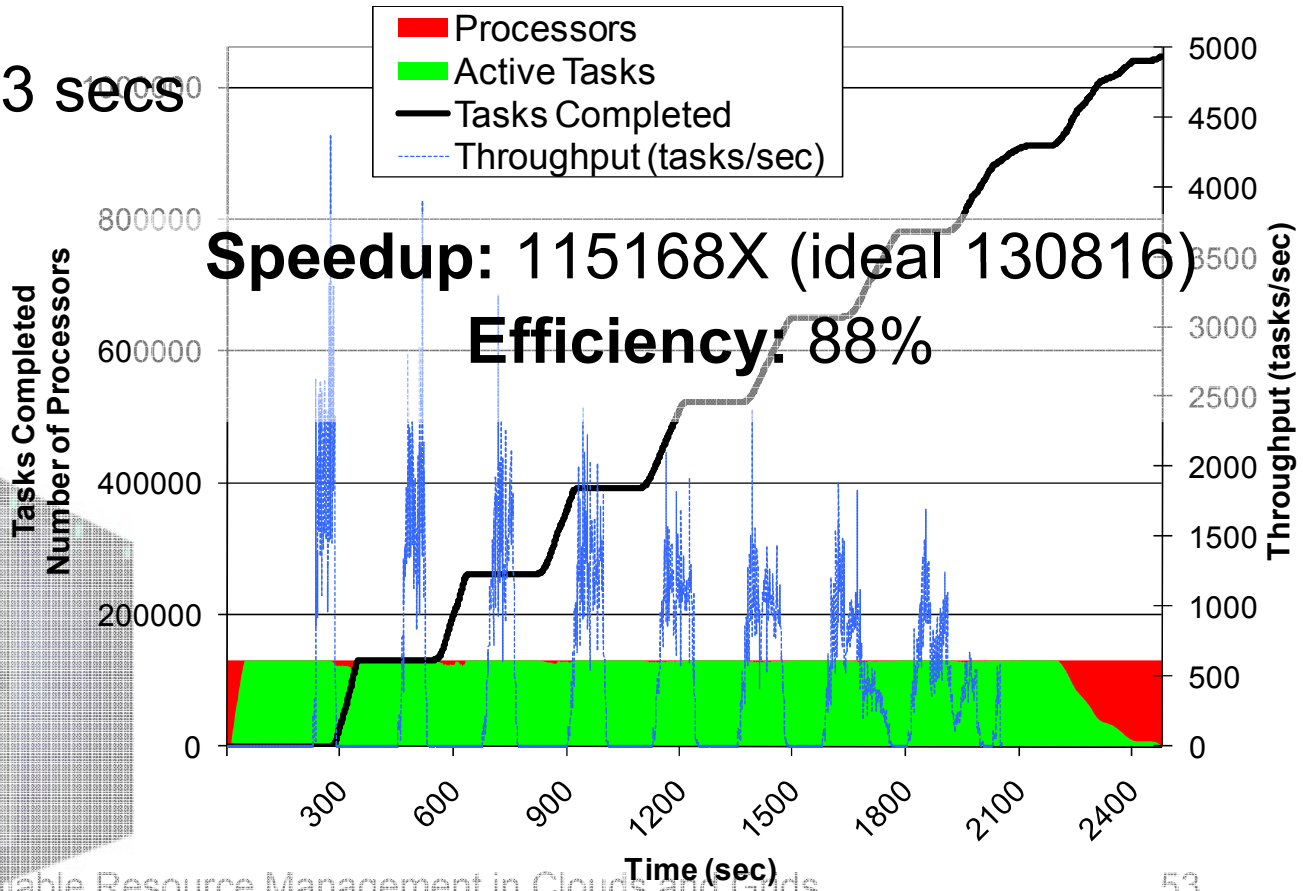


Managing 160K CPUs



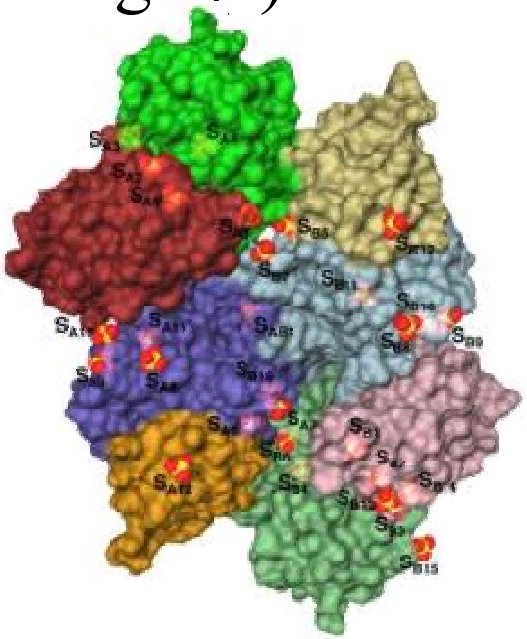
MARS Economic Modeling on IBM BG/P (128K CPUs)

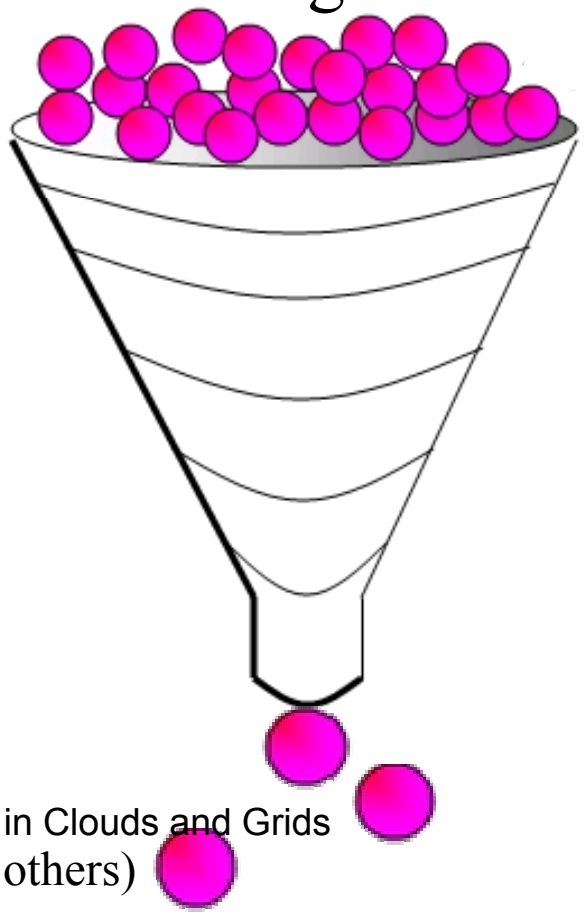
- CPU Cores: 130816
- Tasks: 1048576
- Elapsed time: 2483 secs
- CPU Years: 9.3



Many Many Tasks: Identifying Potential Drug Targets

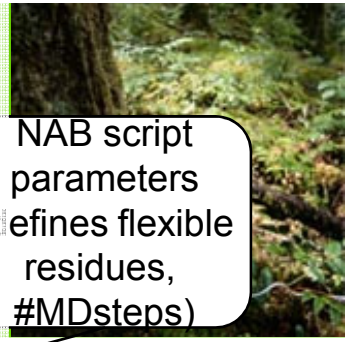
Protein target(s) x 2M+ ligands

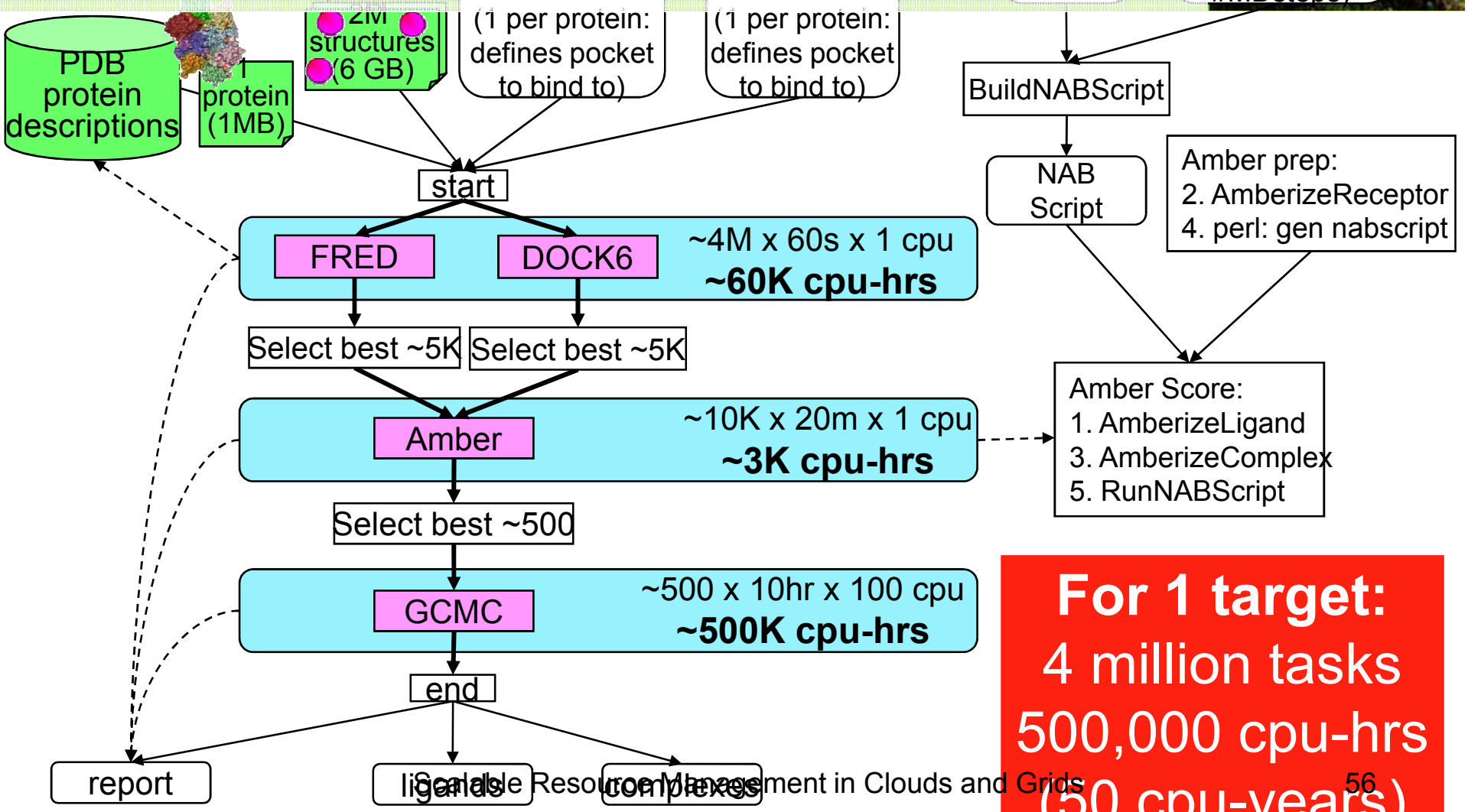




(Mike Kubal, Benoit Roux, and others)

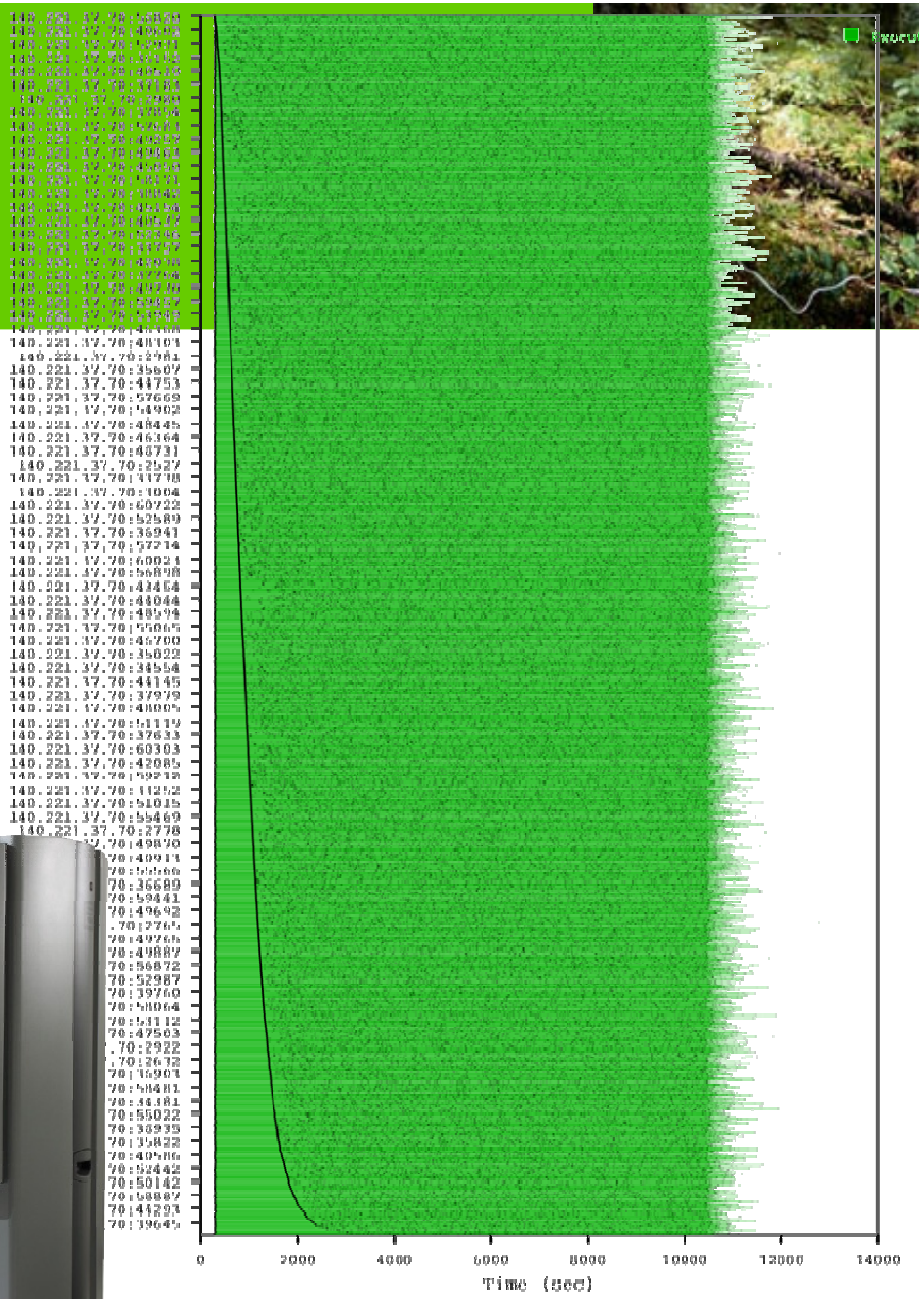
Many Many Tasks: Identifying Potential Drug Targets





DOCK on SiCortex

- CPU cores: 5760
- Tasks: 92160
- Elapsed time: 12821 sec
- Compute time: 1.94 CPU years
- Average task time: 660.3 sec
- Speedup: 5650X (ideal 5760)
- Efficiency: 98.2%



DOCK on the BG/P

CPU cores: 118784

Tasks: 934803

Elapsed time: 2.01 hours

Compute time: 21.43 CPU years

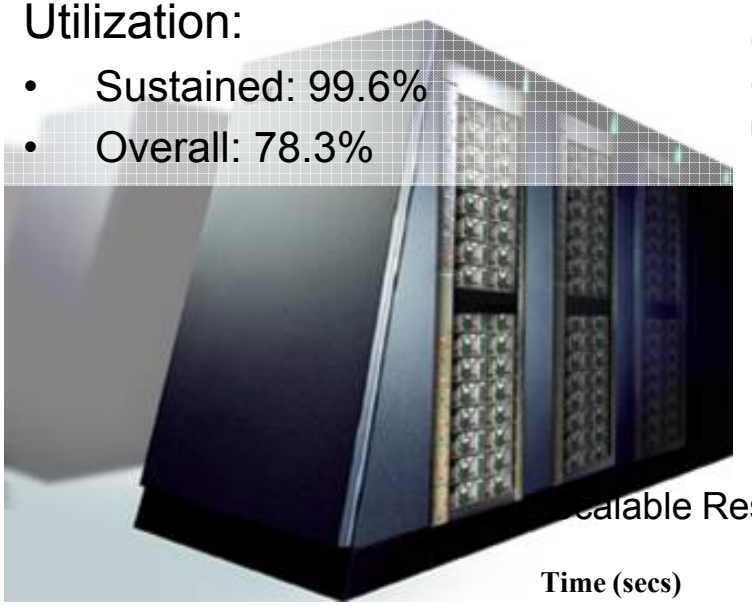
Average task time: 667 sec

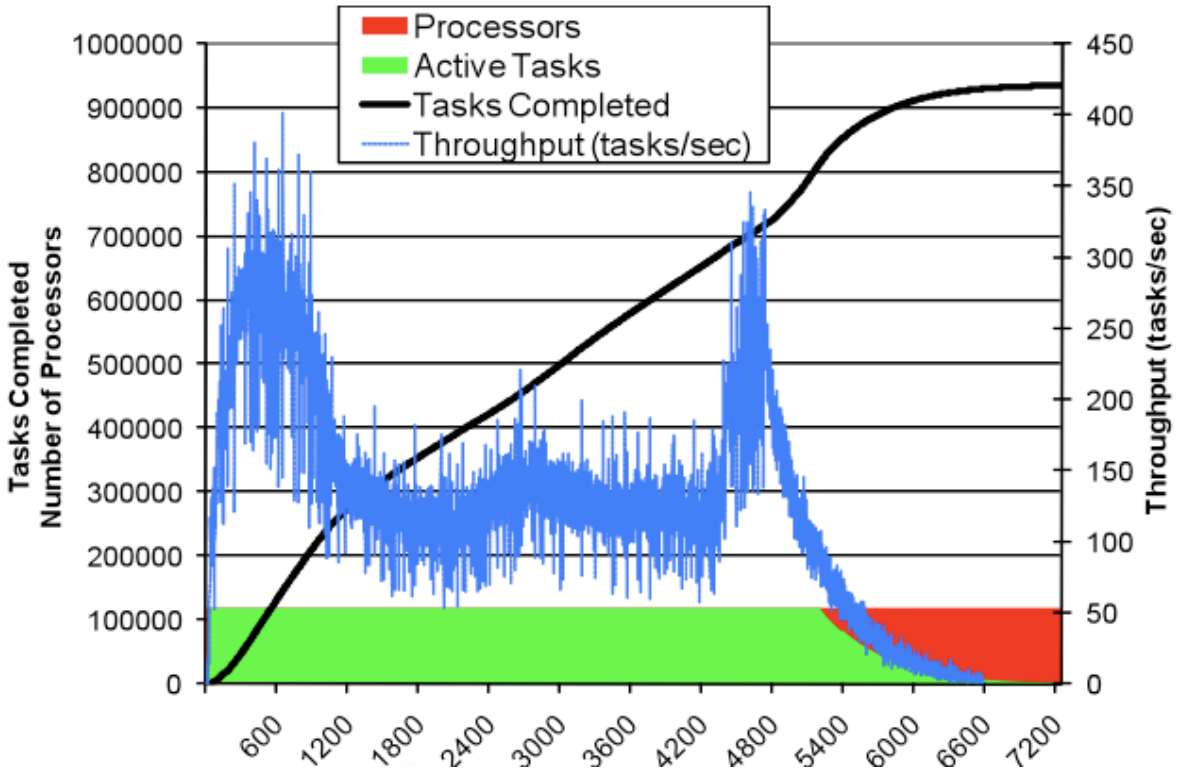
Relative Efficiency: 99.7%

(from 16 to 32 racks)

Utilization:

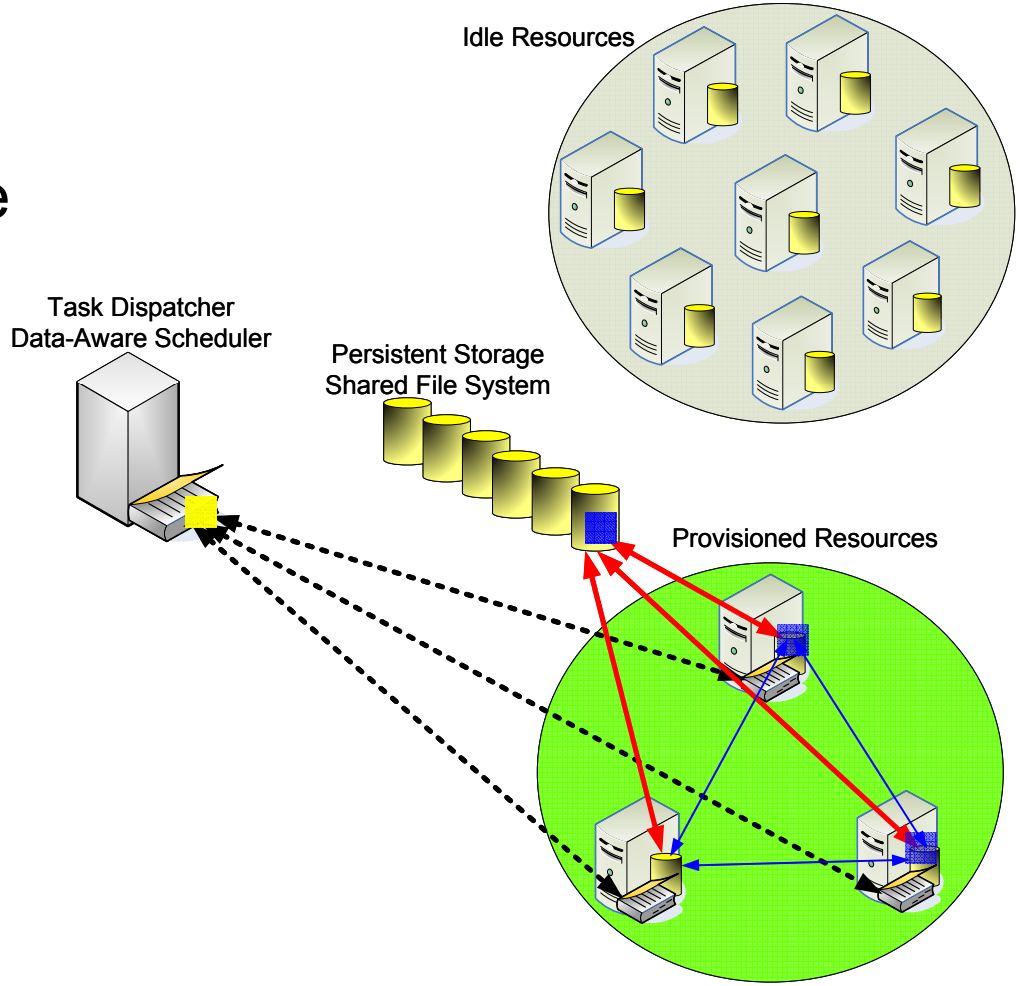
- Sustained: 99.6%
- Overall: 78.3%





Data Diffusion

- Resource acquired in response to demand
- Data and applications diffuse from archival storage to newly acquired resources
- Resource “caching” allows faster responses to subsequent requests
  - Cache Eviction Strategies: RANDOM, FIFO, LRU, LFU
- Resources are released when demand drops



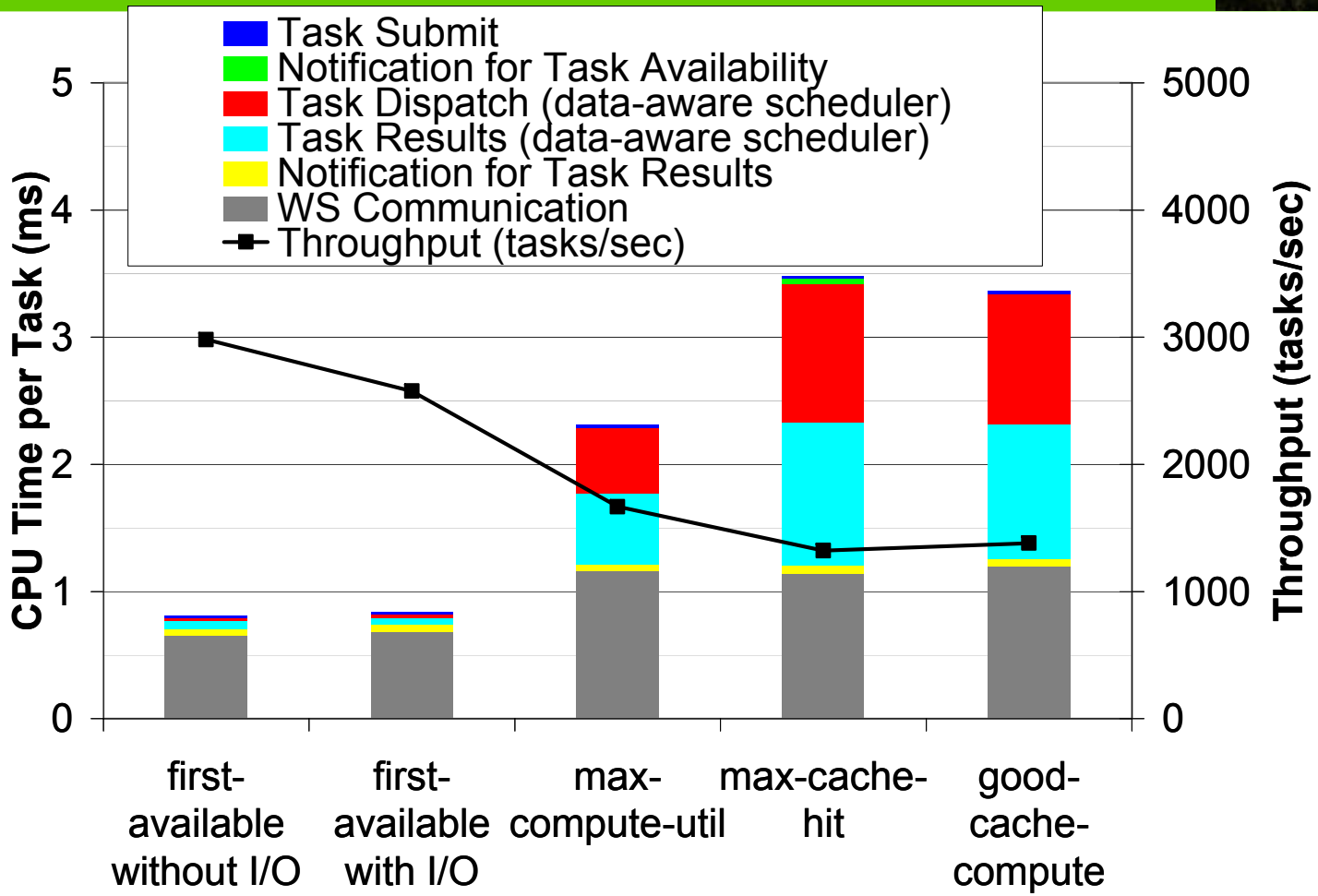
Data Diffusion

- Considers both data and computations to optimize performance
  - Supports data-aware scheduling
  - Can optimize compute utilization, cache hit performance, or a mixture of the two
- Decrease dependency of a shared file system
  - Theoretical linear scalability with compute resources
  - Significantly increases meta-data creation and/or modification performance
- Central for “data-centric task farm” realization

Scheduling Policies

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

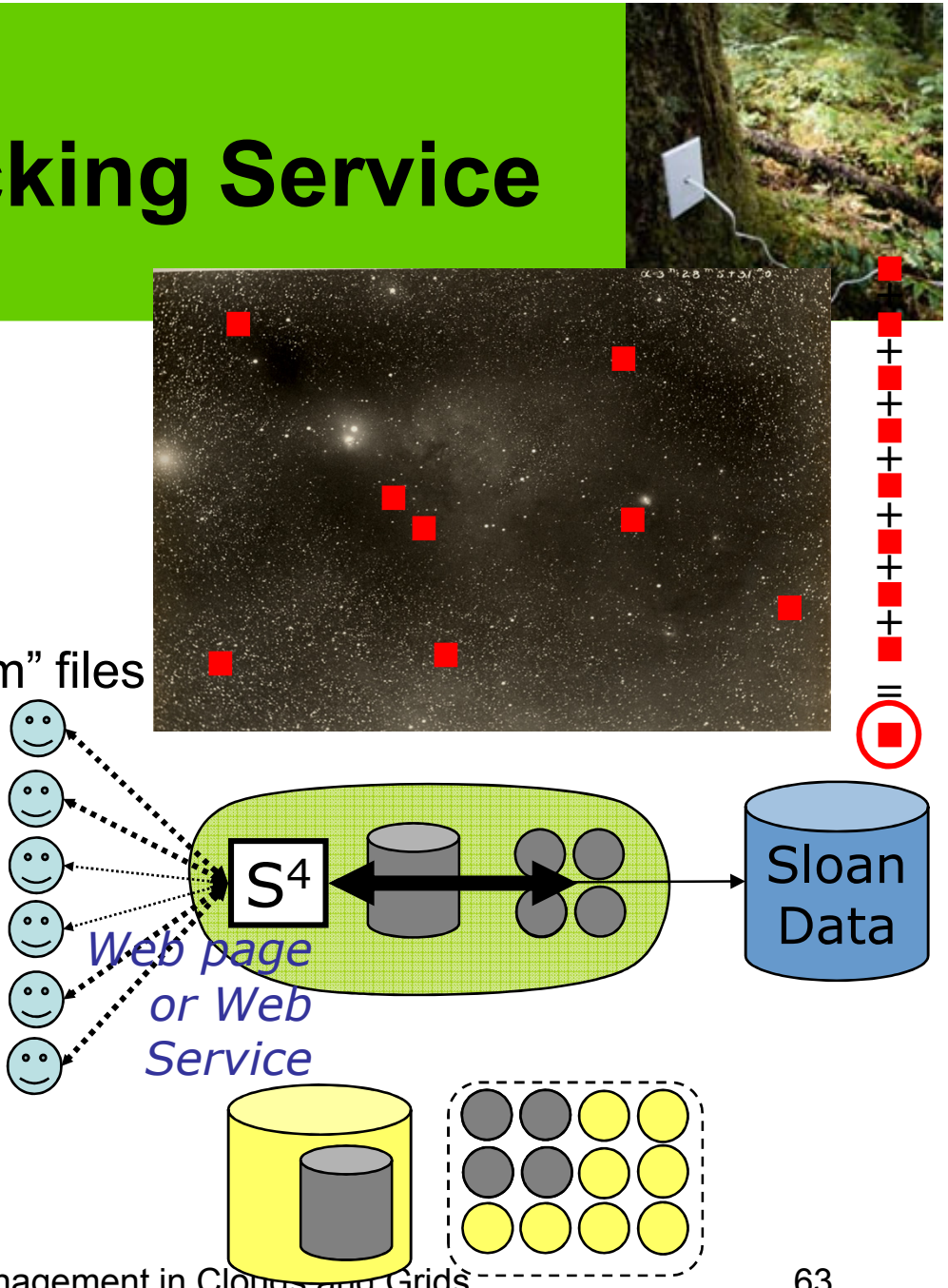
Data-Aware Scheduler Profiling



AstroPortal Stacking Service

- Purpose
  - On-demand “stacks” of random locations within ~10TB dataset
- Challenge
  - Rapid access to 10-10K “random” files
  - Time-varying load
- Sample Workloads

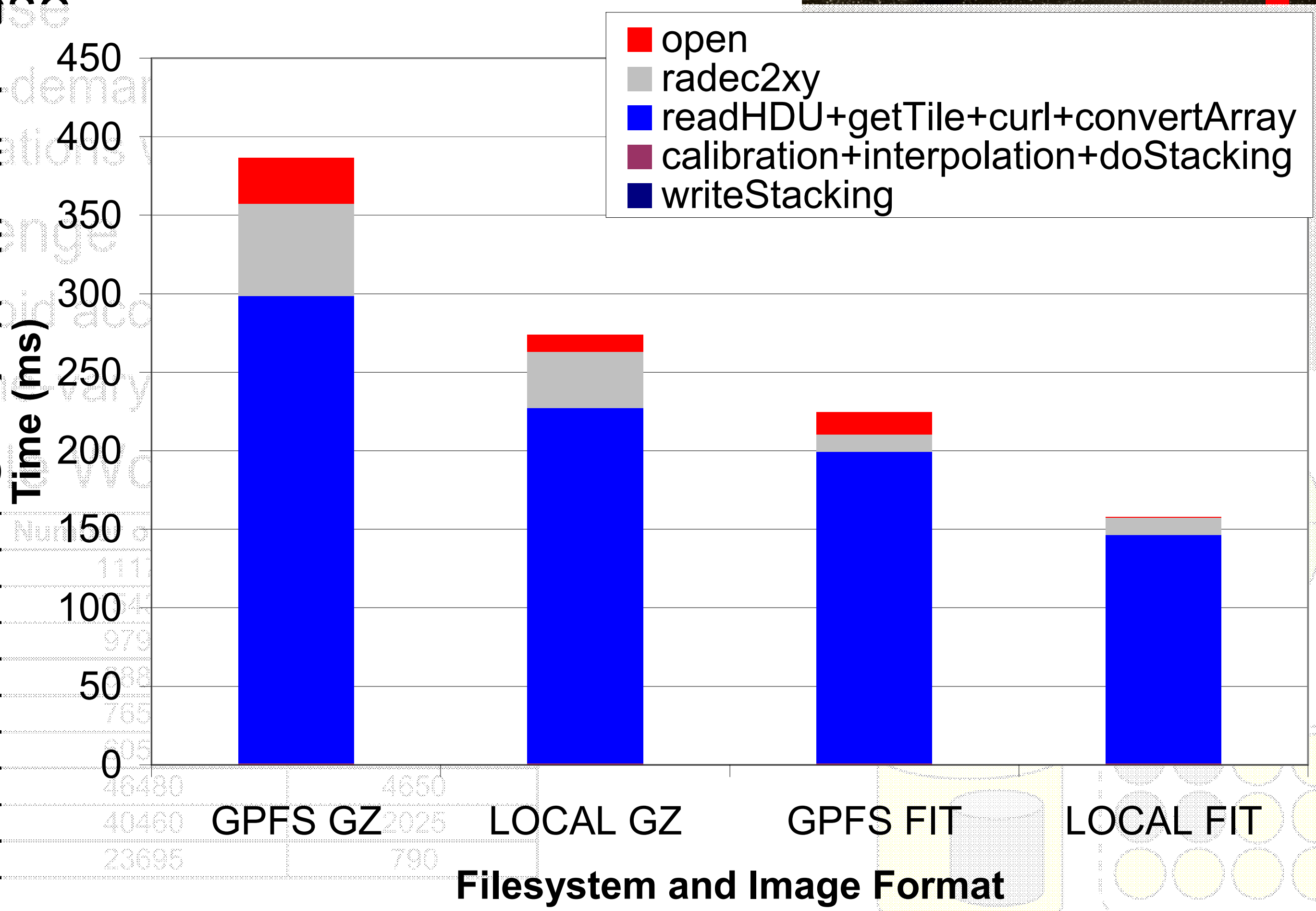
Locality	Number of Objects	Number of Files
1	111700	111700
1.38	154345	111699
2	97999	49000
3	88857	29620
4	76575	19145
5	60590	12120
10	46480	4650
20	40460	2025
30	23695	790

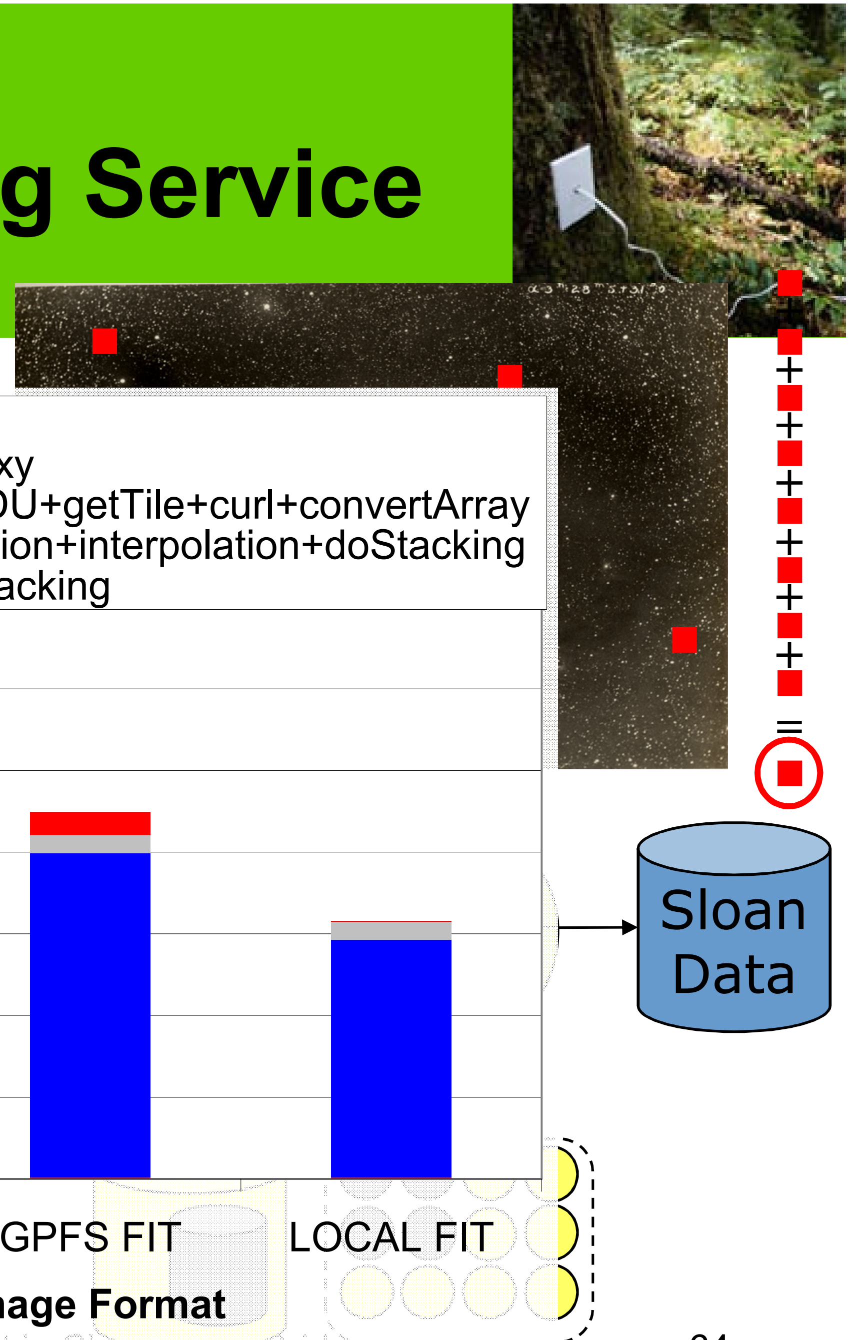


AstroPortal Stacking Service

- Purpo
  - On-
    loc
- Challe
  - Ra
  - Tim
- Samp

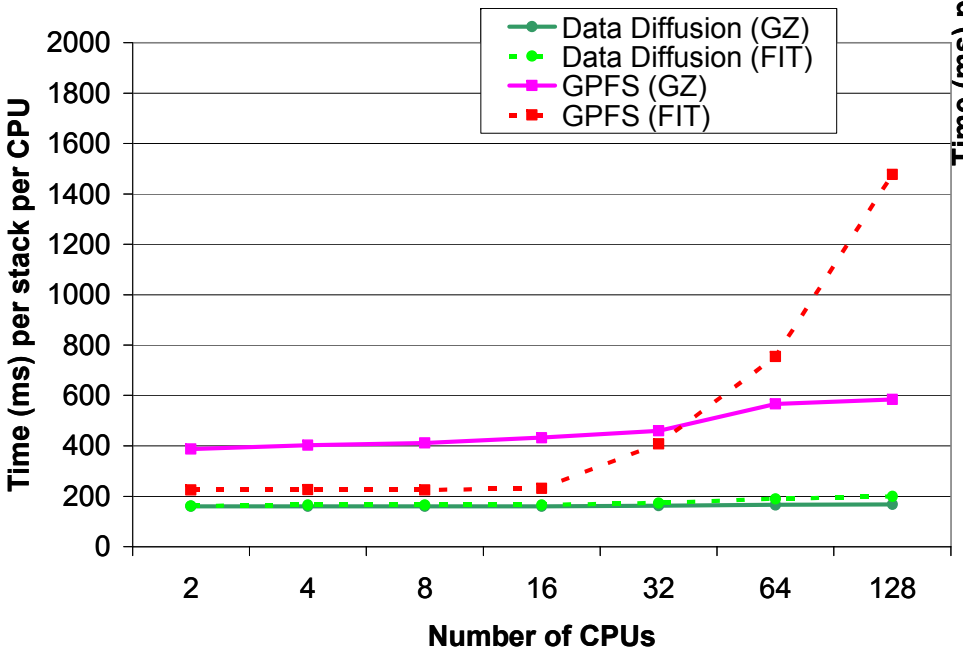
Locality
1
1.38
2
3
4
5
10
20
30

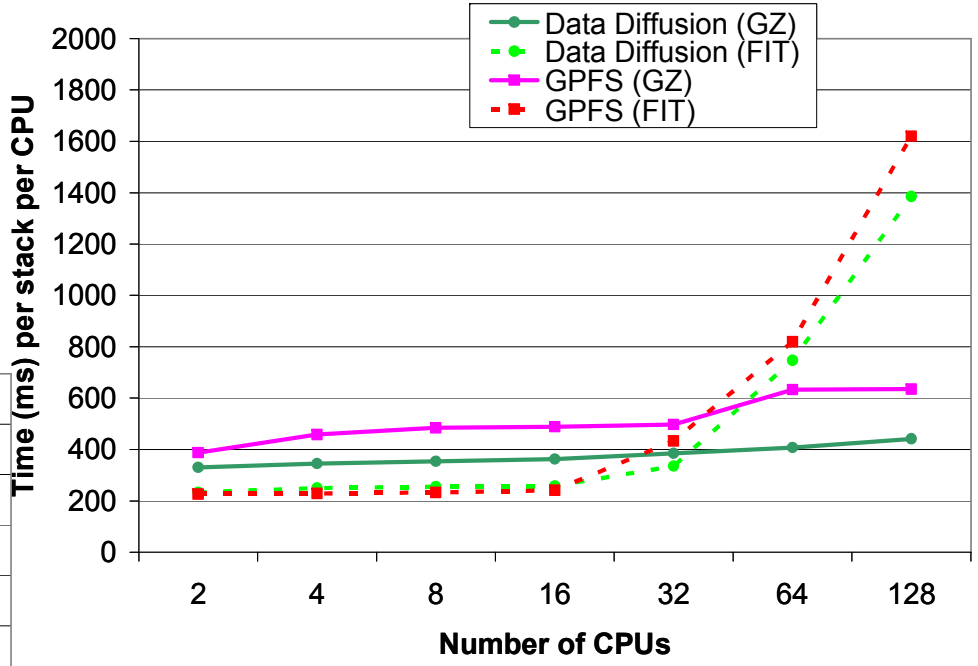




AstroPortal Stacking Service with Data Diffusion

Low data locality ➔
- Similar (but better) performance to GPFS

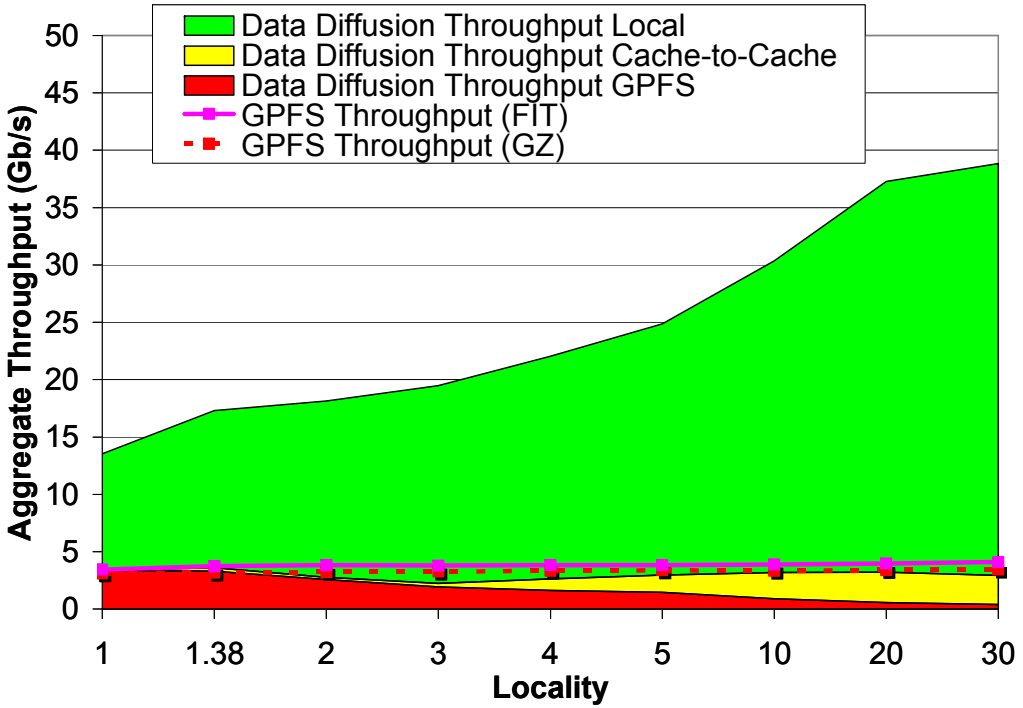


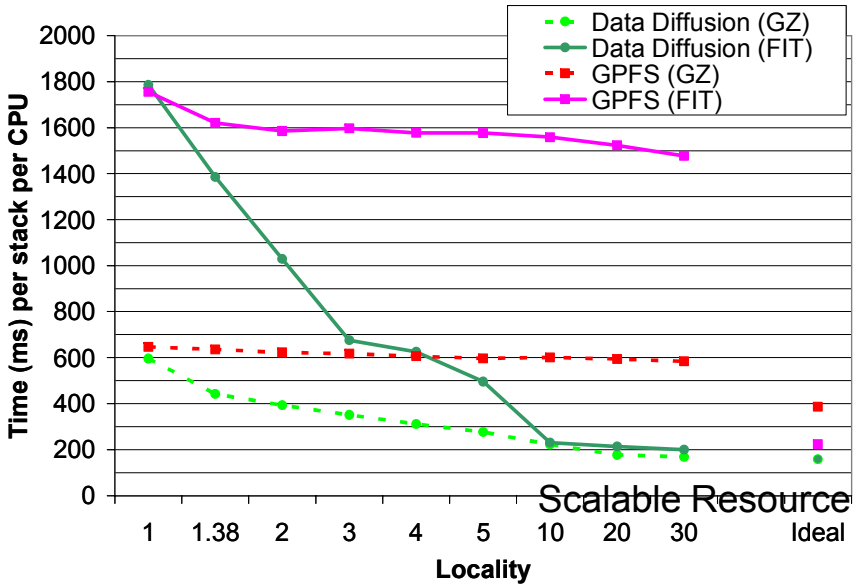


➔High data locality
- Near perfect scalability

AstroPortal Stacking Service with Data Diffusion

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

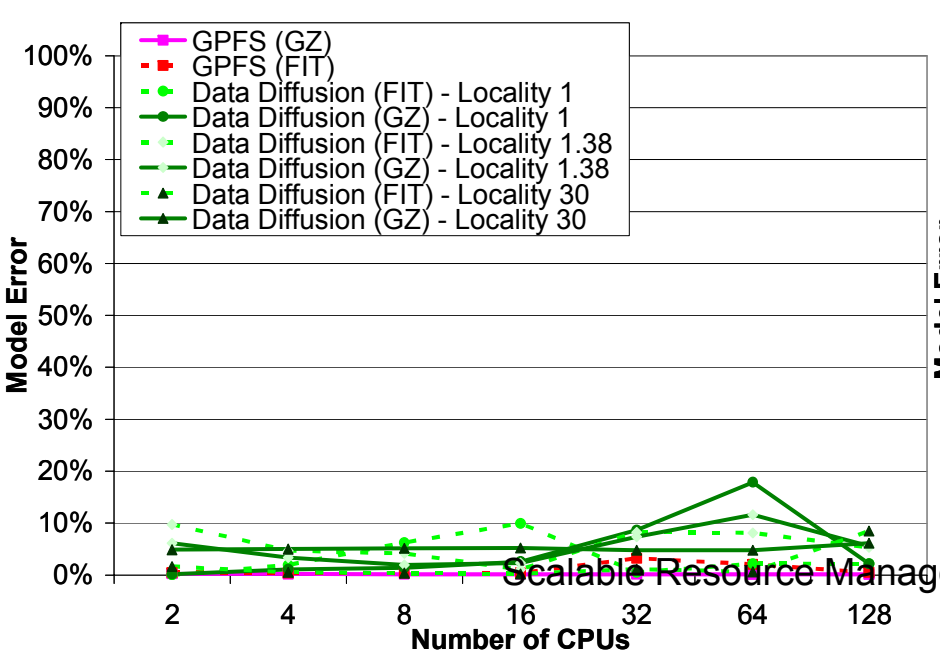


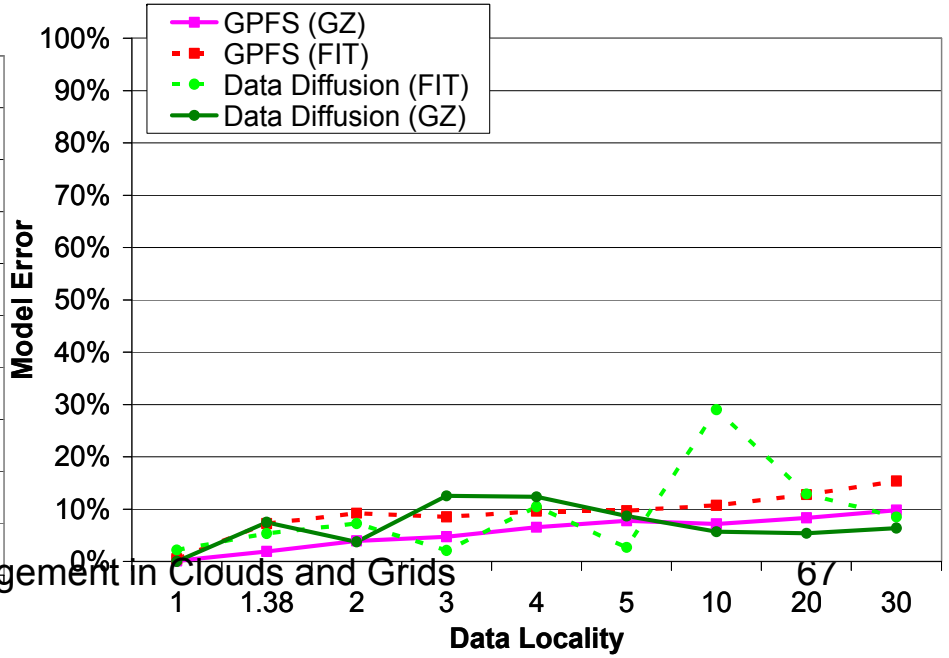


- Big performance gains as locality increases

AMDASK Model Validation

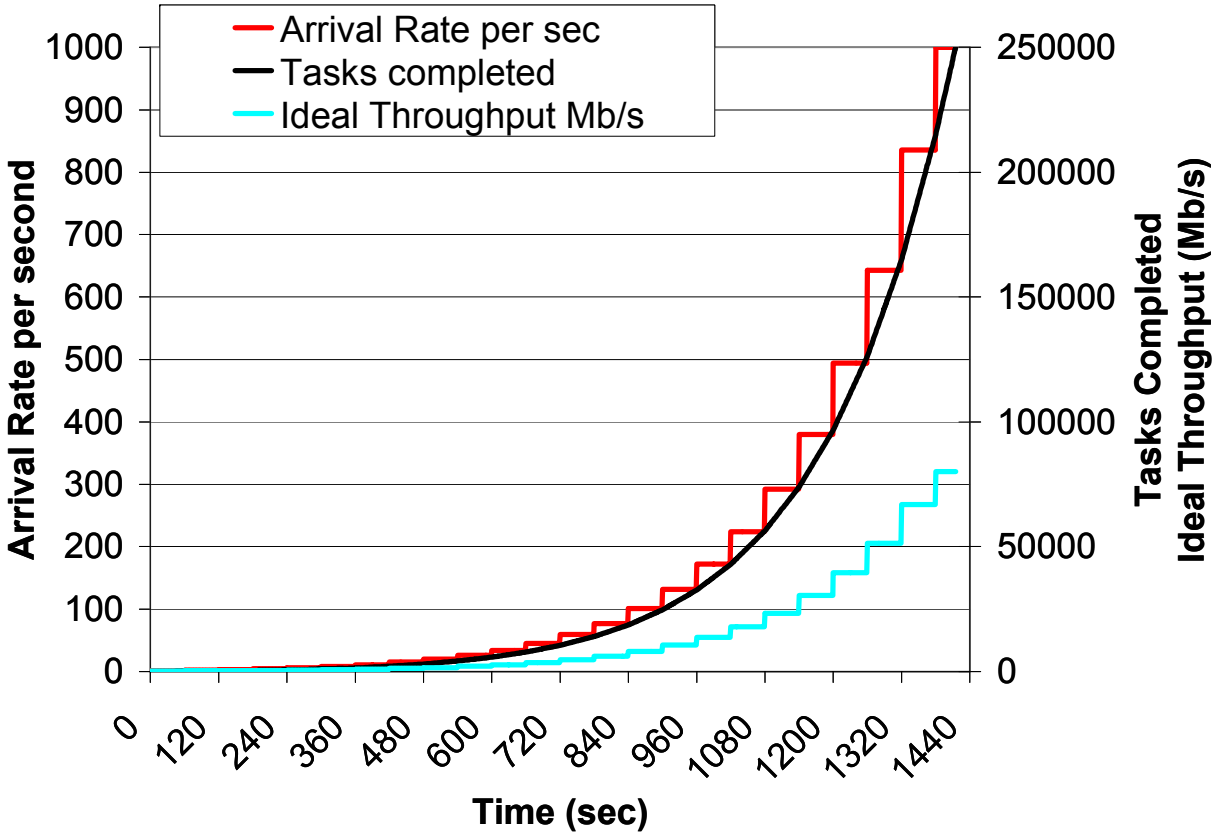
- Stacking service (large scale astronomy application)
- 92 experiments
- 558K files
  - Compressed: 2MB each ➔ 1.1TB
  - Un-compressed: 6MB each ➔ 3.3TB





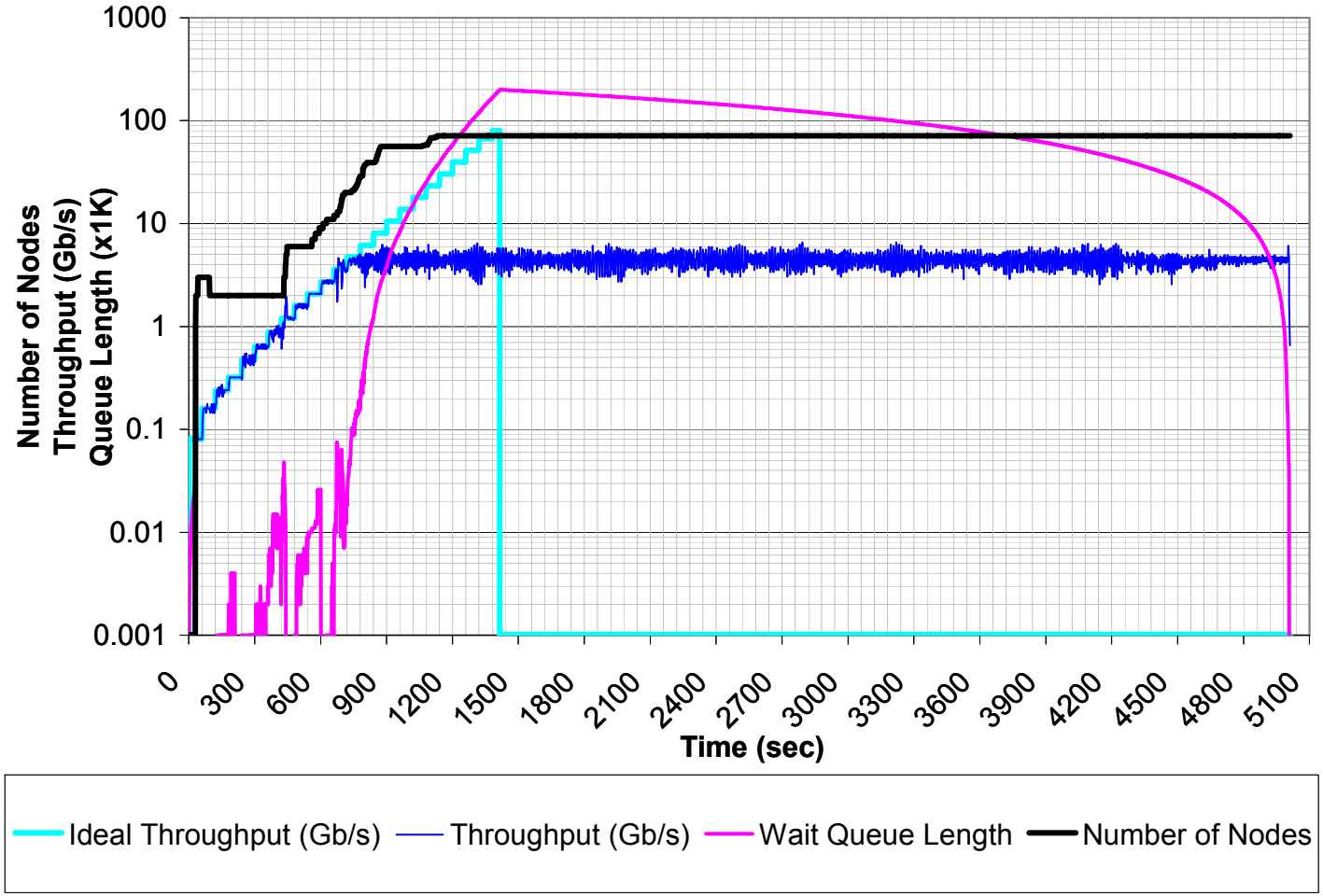
Data Diffusion: Data-Intensive Workload

- 250K tasks
  - 10MB reads
  - 10ms compute
- Vary arrival rate:
  - Min: 1 task/sec
  - Increment function: CEILING(*1.3)
  - Max: 1000 tasks/sec
- 128 processors
- Ideal case:
  - 1415 sec
  - 80Gb/s peak throughput



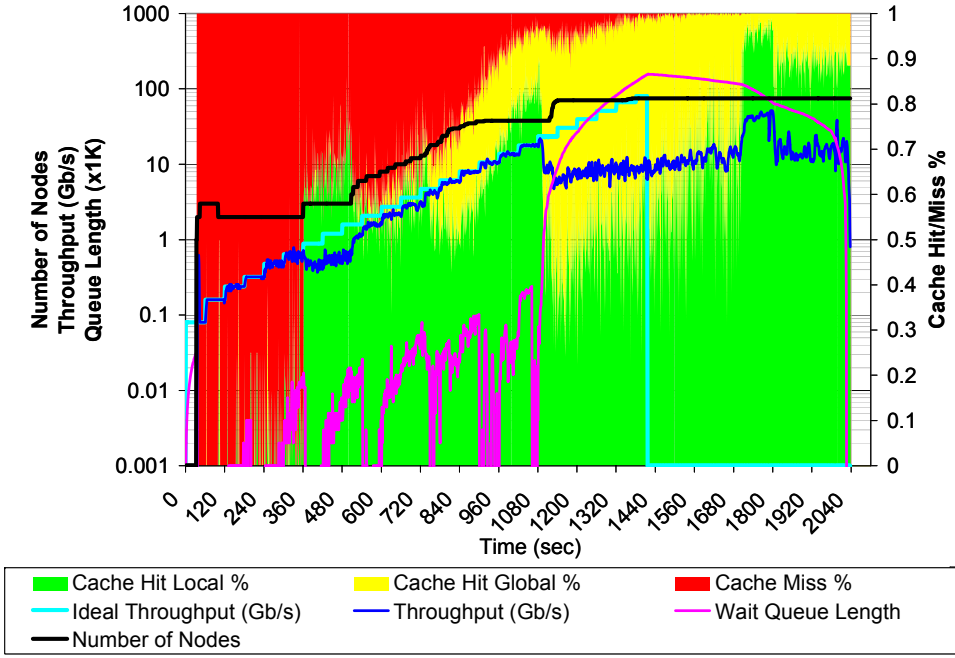
Data Diffusion: First-available (GPFS)

- **GPFS vs. ideal**: 5011 sec vs. 1415 sec

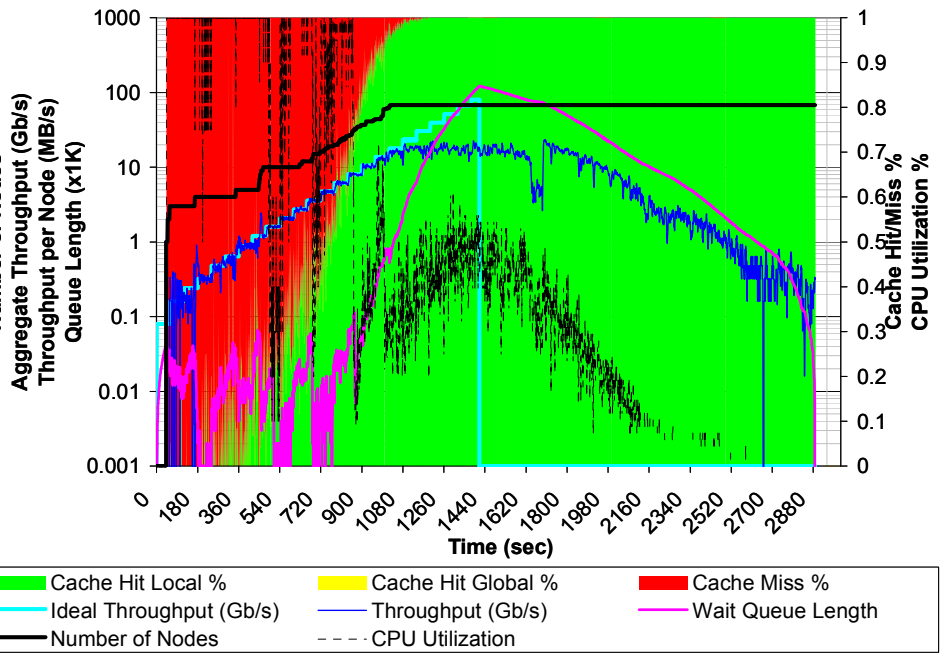


Data Diffusion: Max-compute-util & max-cache-hit

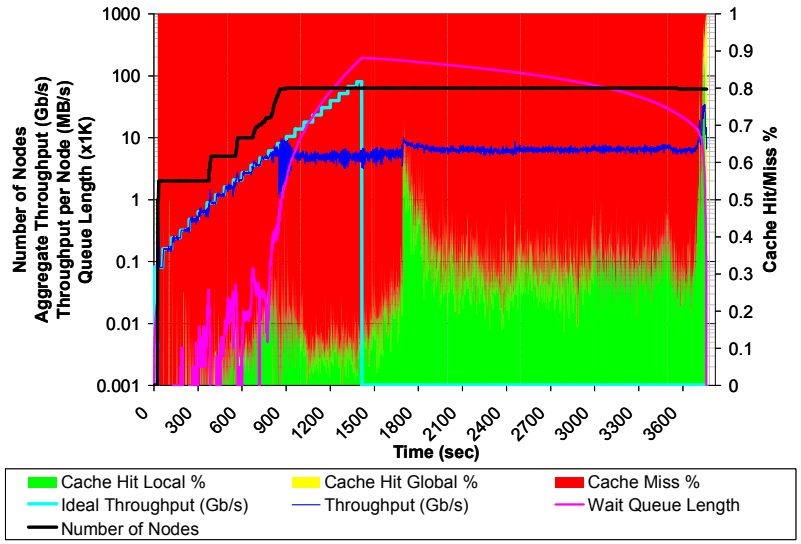
Max-compute-util



Max-cache-hit

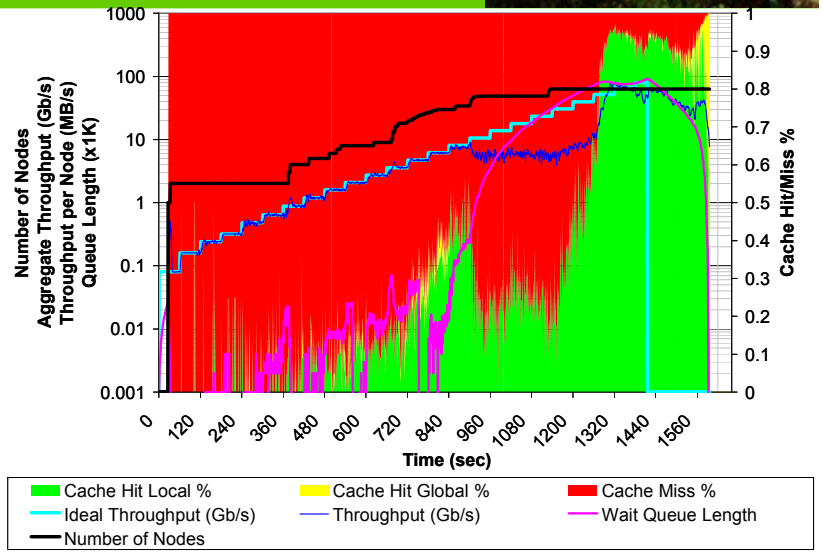


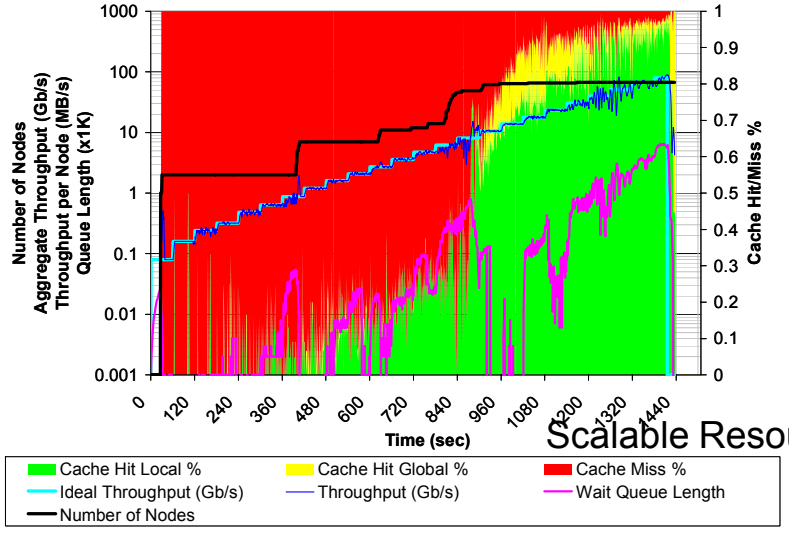
Data Diffusion: Good-cache-compute



← 1GB

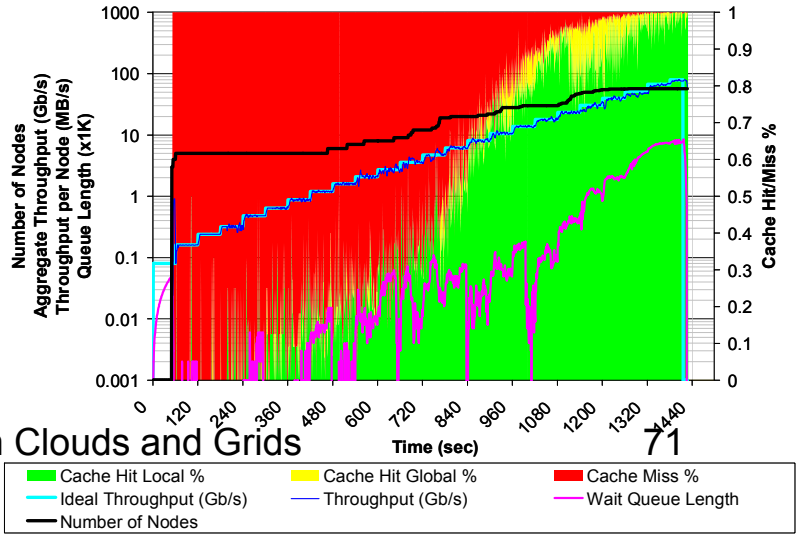
1.5GB →



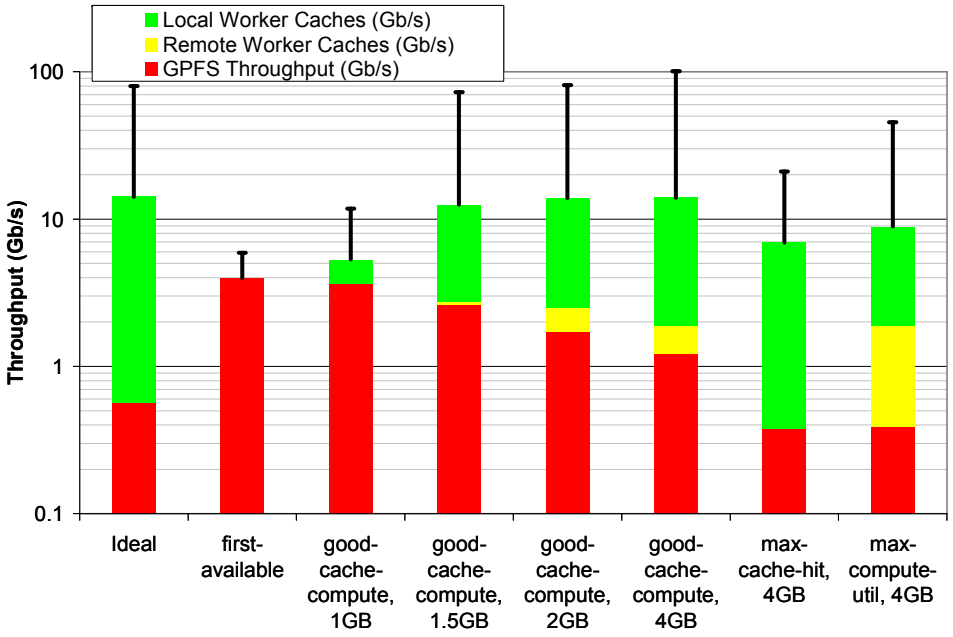


← 2GB

4GB →

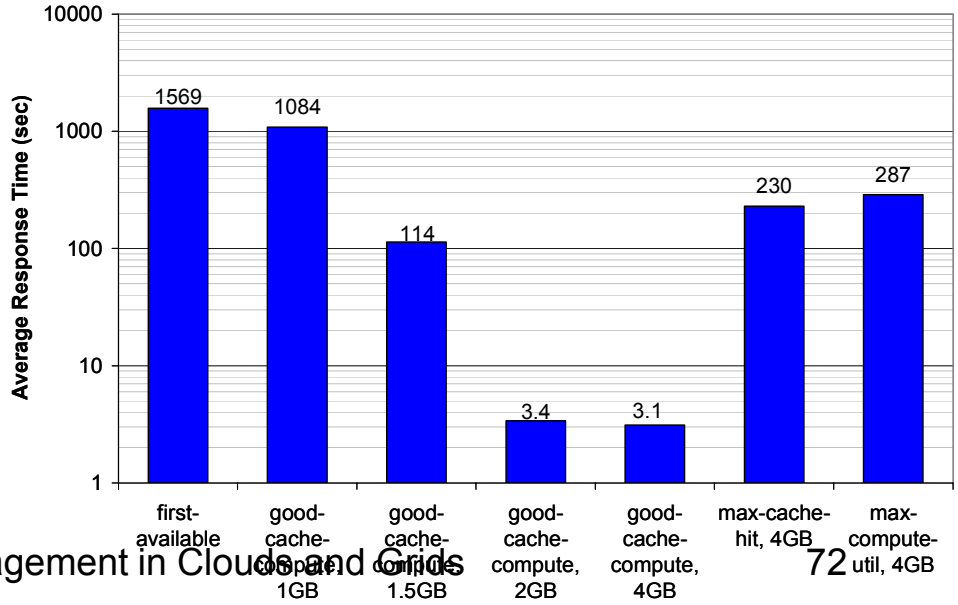


Data Diffusion: Throughput and Response Time



← Throughput:

- Average: 14Gb/s vs 4Gb/s
- Peak: 100Gb/s vs. 6Gb/s

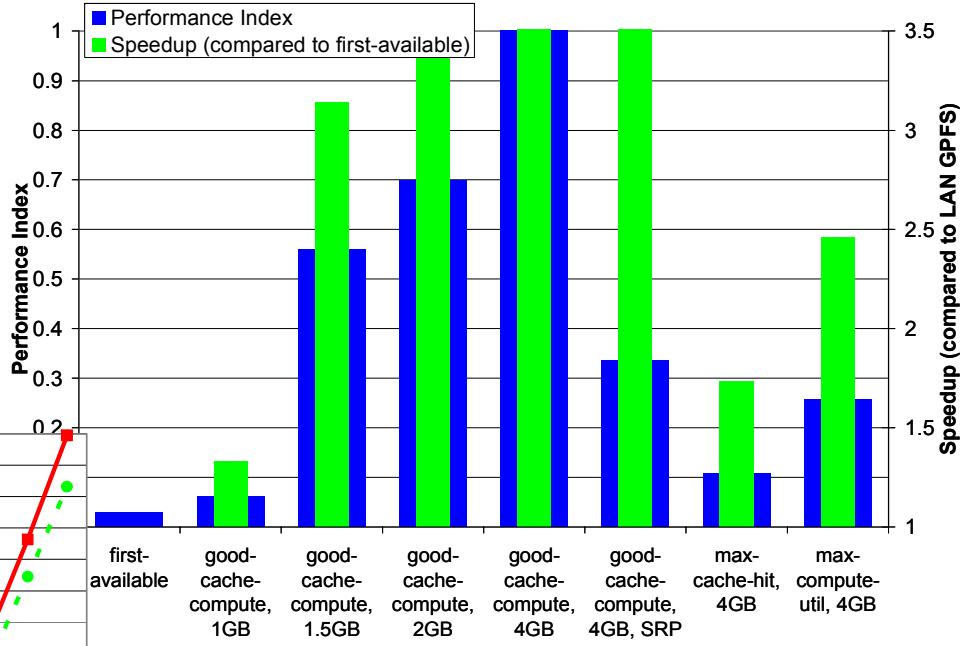


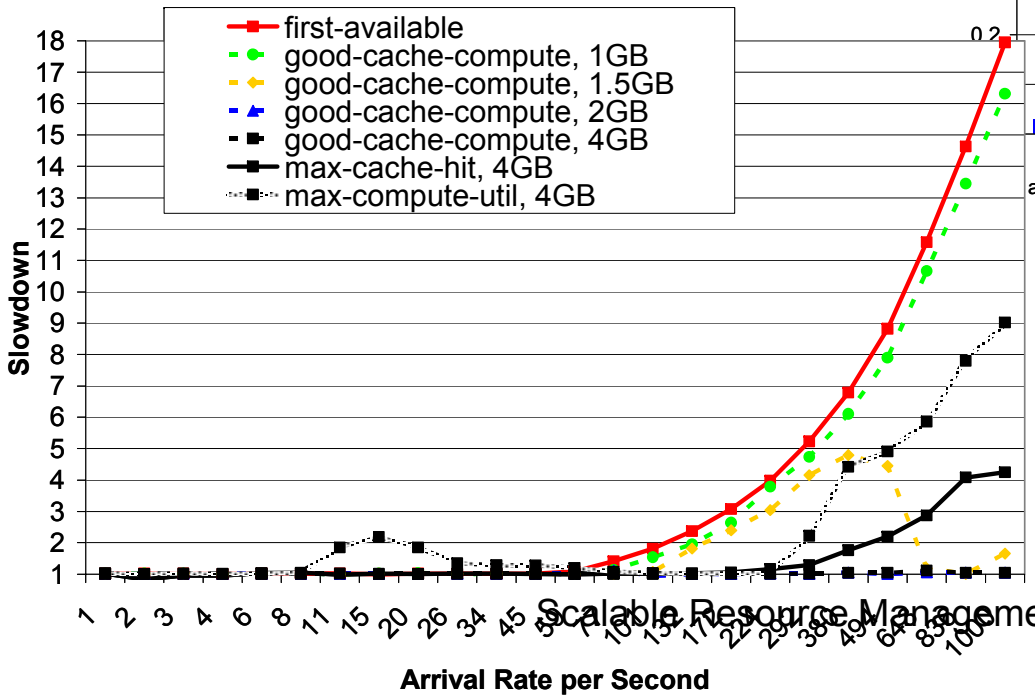
Response Time →

- 3 sec vs 1569 sec → 506X

Data Diffusion: Performance Index, Slowdown, and Speedup

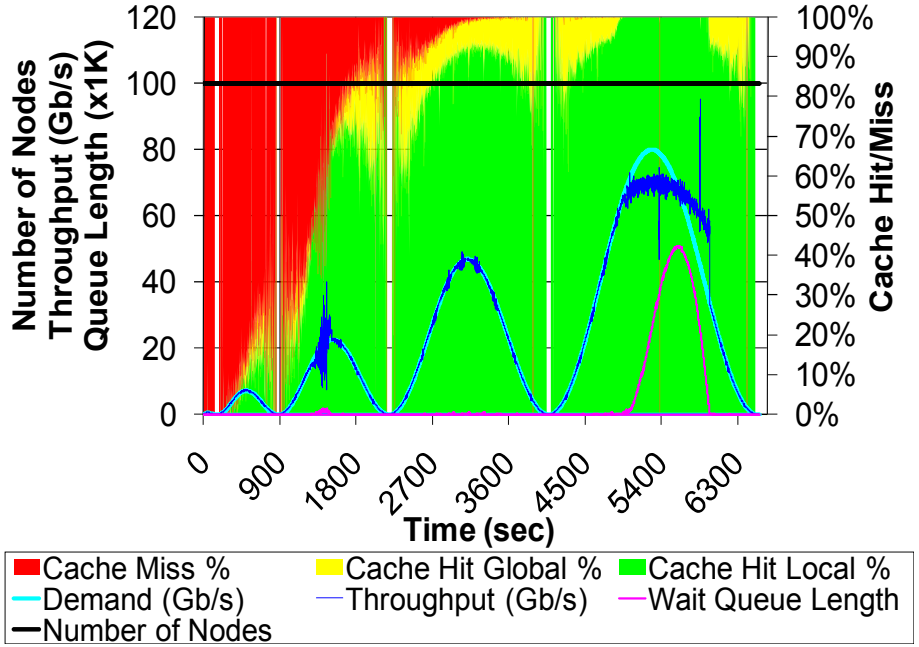
- Performance Index:
  - 34X higher
- Speedup
  - 3.5X faster than GPFS

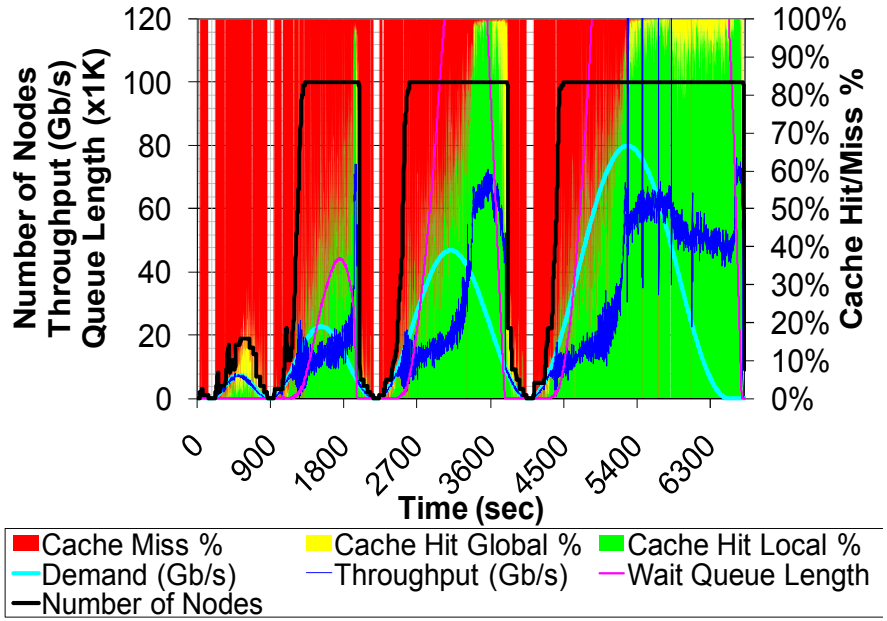




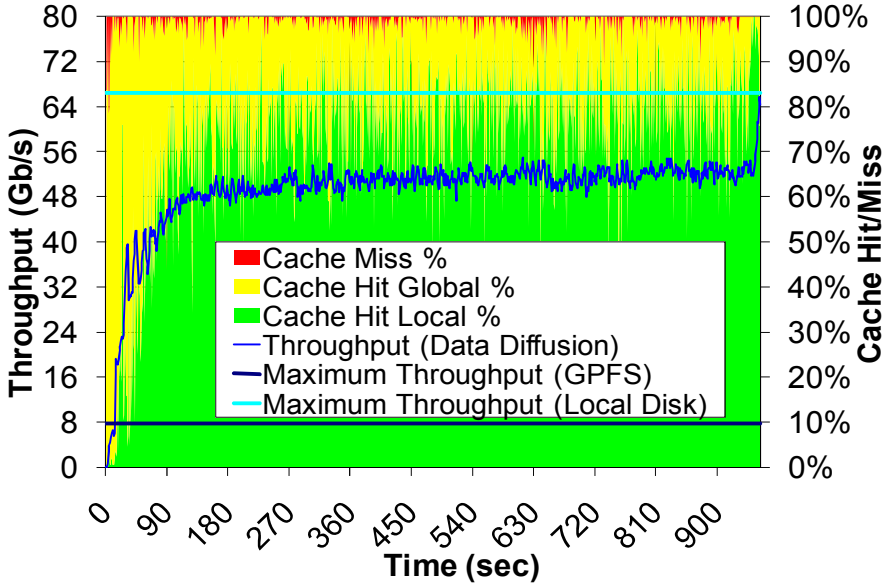
- Slowdown:
  - 18X slowdown for GPFS
  - Near ideal 1X slowdown for large enough caches

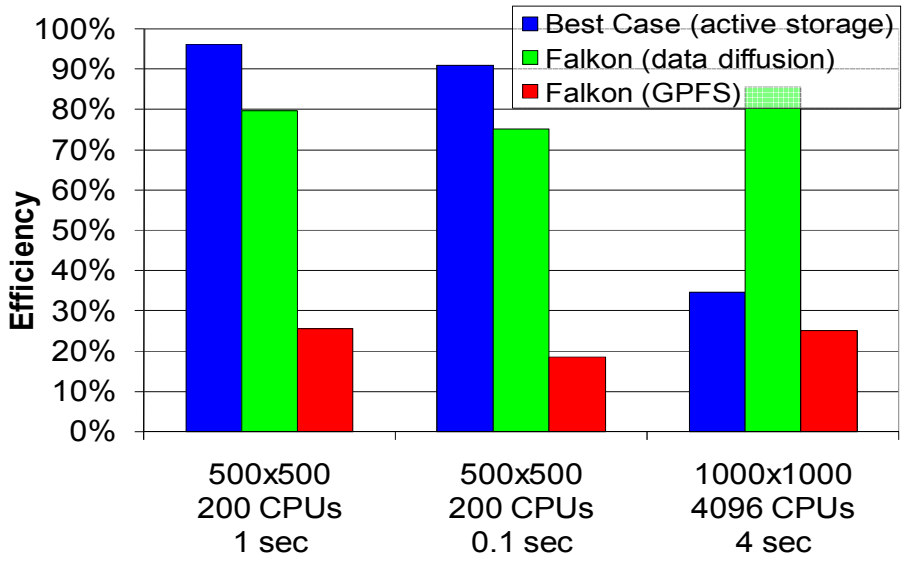
Data Diffusion
Sin-Wave Workload





Data Diffusion
All-Pairs Workload





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

Related Work: Task Farms

- [*Casanova99*]: Adaptive Scheduling for Task Farming with Grid Middleware
- [*Heymann00*]: Adaptive Scheduling for Master-Worker Applications on the Computational Grid
- [*Danelutto04*]: Adaptive Task Farm Implementation Strategies
- [*González-Vélez05*]: An Adaptive Skeletal Task Farm for Grids
- [*Petrou05*]: Scheduling Speculative Tasks in a Compute Farm
- [*Reid06*]: Task farming on Blue Gene

Conclusion: none addressed the proposed "data-centric" part of task farms, and the implementations were not as light-weight as ours

Related Work: Resource Provisioning

- [*Appleby01*]: **Oceano** - SLA Based Management of a Computing Utility
- [*Frey02, Mehta06*]: **Condor glide-ins**
- [*Walker06*]: **MyCluster** (based on Condor glide-ins)
- [*Ramakrishnan06*]: Grid Hosting with Adaptive Resource Control
- [*Bresnahan06*]: Provisioning of bandwidth
- [*Singh06*]: Simulations

Conclusion: None allows for dynamic resizing of resource pool (independent of application logic) based on system load

Related Work: Data Management

- [*Beynon01*]: **DataCutter**
- [*Ranganathan03*]: **Simulations**
- [*Ghemawat03,Dean04,Chang06*]: **BigTable**, **GFS**, **MapReduce**
- [*Liu04*]: **GridDB**
- [*Chervenak04,Chervenak06*]: **RLS** (Replica Location Service), **DRS** (Data Replication Service)
- [*Tatebe04,Xiaohui05*]: **GFarm**
- [*Branco04,Adams06*]: **DIAL/ATLAS**
- [*Kosar06*]: **Stork**
- [*Thain08*]: **Chirp/Parrot**

Conclusion: None focused on the co-location of storage and generic black box computations with data-aware scheduling while operating in a dynamic environment

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
    - BG/P: 0.35 gigaflops/watt **(higher is better)**
    - SiCortex: 0.32 gigaflops/watt
    - BG/L: 0.23 gigaflops/watt
    - x86-based HPC systems: an order of magnitude lower
- Loosely coupled apps do not require specialized system software
- Shared 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

Conclusions & Contributions

- Defined an *abstract model for performance efficiency of data analysis workloads* using data-centric task farms
- Provide a reference implementation (Falkon)
  - Use a streamlined dispatcher to increase task throughput by several orders of magnitude over traditional LRMs
  - Use multi-level scheduling to reduce perceived wait queue time for tasks to execute on remote resources
  - Address data diffusion through co-scheduling of storage and computational resources to improve performance and scalability
  - Provide the benefits of dedicated hardware without the associated high cost
  - Show effectiveness of data diffusion:
    - real large-scale astronomy application and a variety of synthetic workloads
  - Show effectiveness of streamlined task dispatching and dynamic resource provisioning:
    - astronomy, medicine, chemistry, molecular dynamics, economic modelling, and data mining

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
- Dissertation Committee:
  - Ian Foster, The University of Chicago & Argonne National Laboratory
  - Rick Stevens, The University of Chicago & Argonne National Laboratory
  - Alex Szalay, The Johns Hopkins University
- 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

Recent Collaborators (2005 – Present)

- **University of Chicago and/or Argonne National Laboratory**
  - William Allcock
  - Pete Beckman
  - John Bresnahan
  - Ian Foster
  - Kamil Iskra
  - Kate Keahey
  - Michael Papka
  - Rick Stevens
  - Mike Wilde
- **Cisco**
  - Petre Dini
- **Delft University of Technology**
  - Dick Epema
  - Alexandru Iosup
- **Fermi National Laboratory**
  - Catalin Dumitrescu
- **Indiana University**
  - Marlon Pierce
- **Marine Biological Laboratory**
  - Jennifer Schoph
- **Microsoft**
  - Jim Gray
  - Yong Zhao
- **NASA Ames Research Center**
  - Jerry C. Yan
- **The Johns Hopkins University**
  - Alex Szalay
- **University of British Columbia**
  - Matei Ripeanu
- **University of Notre Dame**
  - Amitabh Chaudhary
  - Douglas Thain
- **University of Southern California**
  - Carl Kesselman
  - Laura Pearlman
- **Wayne State University**
  - Shiyong Lu
  - Loren Schwiebert

Publications/Proposals
Central to Dissertation
(2005 – Present)

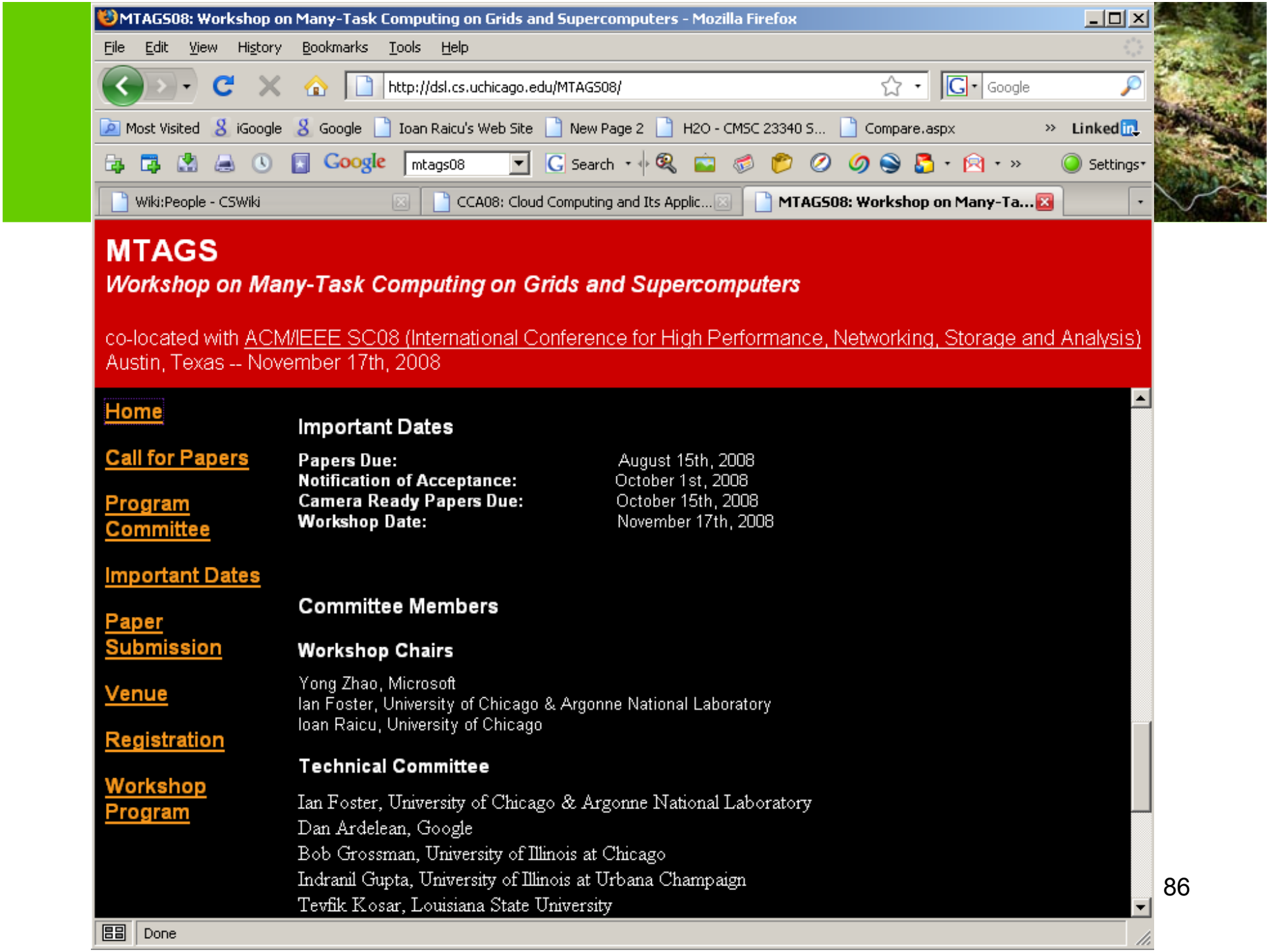
1. **Ioan Raicu**, Ian Foster, Yong Zhao, Philip Little, Christopher Moretti, Amitabh Chaudhary, Douglas Thain. "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review at USENIX NSDI09
2. Ian Foster, Yong Zhao, **Ioan Raicu**, Shiyong Lu. "Cloud Computing and Grid Computing 360-Degree Compared", to appear at IEEE Grid Computing Environments (GCE08) 2008, co-located with IEEE/ACM Supercomputing 2008.
3. Zhao Zhang, Allan Espinosa, Kamil Iskra, **Ioan Raicu**, Ian Foster, Michael Wilde. "Design and Evaluation of a Collective I/O Model for Loosely-coupled Petascale Programming", to appear at IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS08) 2008, co-located with IEEE/ACM Supercomputing 2008.
4. **Ioan Raicu**, Zhao Zhang, Mike Wilde, Ian Foster, Pete Beckman, Kamil Iskra, Ben Clifford. "Towards Loosely-Coupled Programming on Petascale Systems", to appear at IEEE/ACM **Supercomputing 2008**.
5. **Ioan Raicu**, Zhao Zhang, Mike Wilde, Ian Foster. "Enabling Loosely-Coupled Serial Job Execution on the IBM BlueGene/P Supercomputer and the SiCortex SC5832", Technical Report, Department of Computer Science, **University of Chicago, April 2008**.
6. Ioan Raicu, Ian Foster. "*Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets: Year 2 Status and Year 3 Proposal*", GSRP, Ames Research Center, NASA, March 2008 -- Award funded 10/1/08 - 9/30/09.
7. Quan T. Pham, Atilla S. Balkir, Jing Tie, Ian Foster, Mike Wilde, **Ioan Raicu**. "Data Intensive Scalable Computing on TeraGrid: A Comparison of MapReduce and Swift", Poster Presentation, **TeraGrid Conference 2008**.
8. **Ioan Raicu**, Yong Zhao, Ian Foster, Mike Wilde, Zhao Zhang, Ben Clifford, Mihael Hategan, Sarah Kenny. "Managing and Executing Loosely Coupled Large Scale Applications on Clusters, Grids, and Supercomputers", Extended Abstract, **GlobusWorld08**, part of Open Source Grid and Cluster Conference 2008.
9. Yong Zhao, **Ioan Raicu,** Ian Foster. "*Scientific Workflow Systems for 21st Century e-Science, New Bottle or New Wine?*", Invited Paper, **IEEE Workshop on Scientific Workflows 2008**, co-located with IEEE International Conference on Services Computing (SCC) 2008.
10. **Ioan Raicu,** Yong Zhao, Ian Foster, Alex Szalay. "*Accelerating Large-scale Data Exploration through Data Diffusion*", **International Workshop on Data-Aware Distributed Computing 2008**, co-locate with ACM/IEEE International Symposium High Performance Distributed Computing (HPDC) 2008.
11. **Ioan Raicu**, Ian Foster. "*Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets: Year 2 Status and Year 3 Proposal*", **GSRP, Ames Research Center, NASA**, February 2008.
12. **Ioan Raicu**, Ian Foster. "*Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets: Year 1 Final Report*", **GSRP, Ames Research Center, NASA**, February 2008.
13. Yong Zhao, **Ioan Raicu**, Ian Foster, Mihael Hategan, Veronika Nefedova, Mike Wilde. "*Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments*", to appear as a book chapter in Grid Computing Research Progress, ISBN: 978-1-60456-404-4, **Nova Publisher 2008**.
14. **Ioan Raicu**. "Harnessing Grid Resources with Data-Centric Task Farms", **University of Chicago, Computer Science Department,** PhD Proposal, December 2007, Chicago, Illinois.
15. **Ioan Raicu**, Yong Zhao, Catalin Dumitrescu, Ian Foster and Mike Wilde. "*Falkon: A Proposal for Project Globus Incubation*", **Globus Incubation Management Project**, 2007 – Proposal accepted 11/10/07.
16. **Ioan Raicu**, Ian Foster. "*Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets: Year 1 Status and Year 2 Proposal*", **GSRP, Ames Research Center, NASA**, February 2007 -- Award funded 10/1/07 - 9/30/08.
17. **Ioan Raicu**, Yong Zhao, Ian Foster, Alex Szalay. "*A Data Diffusion Approach to Large Scale Scientific Exploration*", **Microsoft Research eScience Workshop 2007**.
18. **Ioan Raicu**, Yong Zhao, Catalin Dumitrescu, Ian Foster, Mike Wilde. "*Falkon: a Fast and Light-weight tasK executiON framework*", **IEEE/ACM SuperComputing 2007**.
19. **Ioan Raicu**, Catalin Dumitrescu, Ian Foster. "*Dynamic Resource Provisioning in Grid Environments*", **TeraGrid Conference 2007**.
20. Yong Zhao, Mihael Hategan, Ben Clifford, Ian Foster, Gregor von Laszewski, **Ioan Raicu**, Tiberiu Stef-Praun, Mike Wilde. "*Swift: Fast, Reliable, Loosely Coupled Parallel Computation*", **IEEE Workshop on Scientific Workflows 2007**.
21. **Ioan Raicu**, Ian Foster. "*Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets*", **GSRP, Ames Research Center, NASA,** February 2006 -- Award funded 10/1/06 - 9/30/07.
22. **Ioan Raicu**, Ian Foster, Alex Szalay. "*Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets*", poster presentation, **IEEE/ACM SuperComputing 2006**.
23. **Ioan Raicu**, Ian Foster, Alex Szalay, Gabriela Turcu. "*AstroPortal: A Science Gateway for Large-scale Astronomy Data Analysis*", **TeraGrid Conference 2006**, June 2006.
24. Alex Szalay, Julian Bunn, Jim Gray, Ian Foster, **Ioan Raicu**. "*The Importance of Data Locality in Distributed Computing Applications*", **NSF Workflow Workshop 2006**.

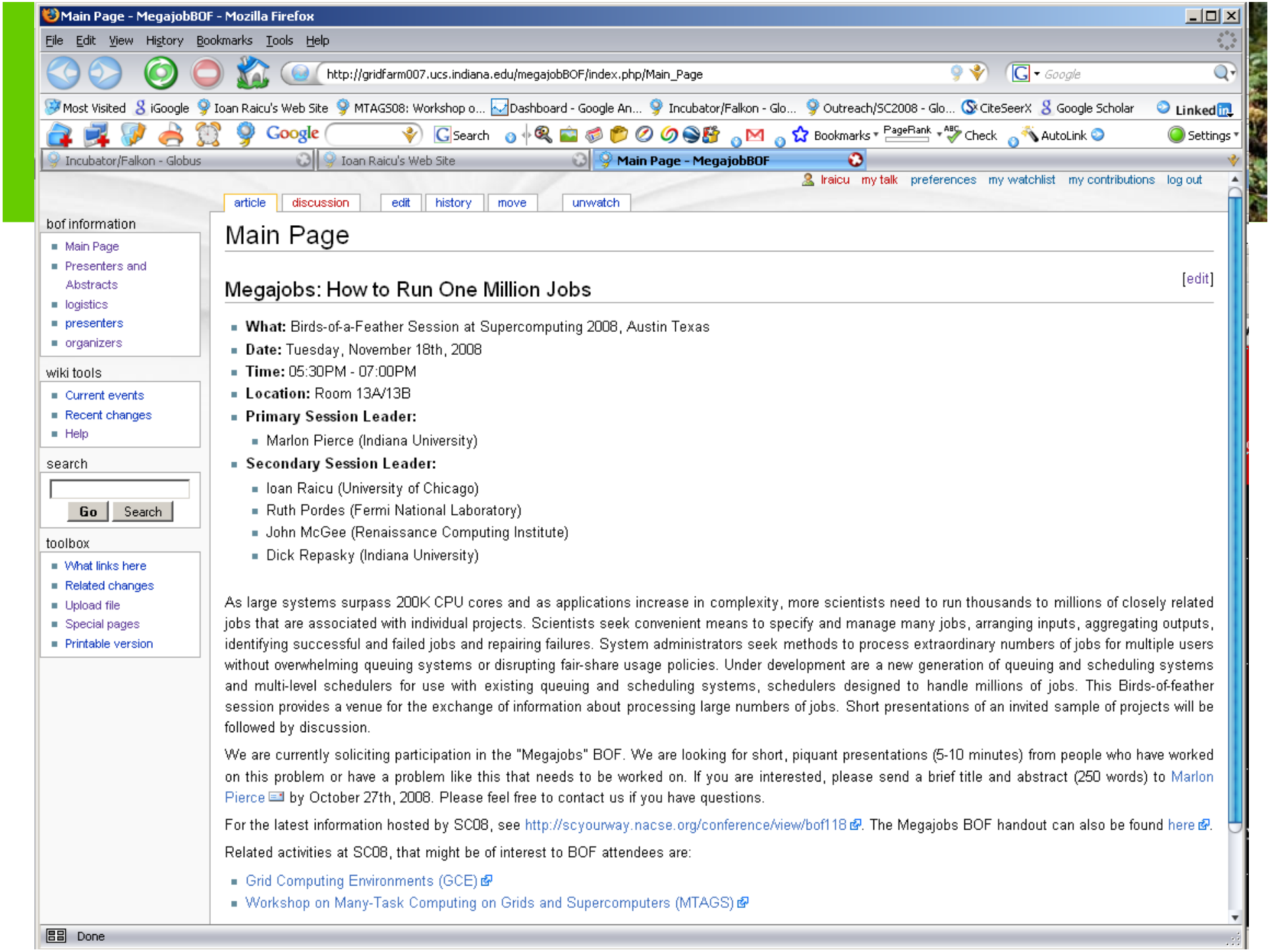
Other Publications
(2002 – 2007)
Disjoint Set from Previous Slide

1. Catalin Dumitrescu, Jan Dünnweber, Philipp Lüdeking, Sergei Gorlatch, Ioan Raicu and Ian Foster. *Simplifying Grid Application Programming Using Web-Enabled Code Transfer Tools*. *Toward Next Generation Grids,* Chapter 6, Springer Verlag, 2007.
2. Catalin Dumitrescu, Alexandru Iosup, H. Mohamed, Dick H.J. Epema, Matei Ripeanu, Nicolae Tapus, Ioan Raicu, Ian Foster. "*ServMark: A Framework for Testing Grids Services*", IEEE Grid 2007.
3. Catalin Dumitrescu, Ioan Raicu, Ian Foster. "*The Design, Usage, and Performance of GRUBER: A Grid uSLA-based Brokering Infrastructure*", International Journal of Grid Computing, 2007.
4. Catalin Dumitrescu, Ioan Raicu, Ian Foster. "*Usage SLA-based Scheduling in Grids*", Journal on Concurrency and Computation: Practice and Experience, 2006.
5. Ioan Raicu, Catalin Dumitrescu, Matei Ripeanu, Ian Foster. "*The Design, Performance, and Use of DiPerF: An automated DIstributed PERformance testing Framework*", International Journal of Grid Computing, Special Issue on Global and Peer-to-Peer Computing, 2006; 25% acceptance rate.
6. Catalin Dumitrescu, Ioan Raicu, Ian Foster. "*Performance Measurements in Running Workloads over a Grid*", The 4th International Conference on Grid and Cooperative Computing (GCC 2005); 11% acceptance rate
7. Catalin Dumitrescu, Ioan Raicu, Ian Foster. "*DI-GRUBER: A Distributed Approach for Grid Resource Brokering*", IEEE/ACM Super Computing 2005 (SC 2005); 22% acceptance rate.
8. William Allcock, John Bresnahan, Rajkumar Kettimuthu, Michael Link, Catalin Dumitrescu, Ioan Raicu, Ian Foster, "*The Globus Striped GridFTP Framework and Server*," sc, p. 54, ACM/IEEE SC 2005 Conference (SC'05), 2005; 22% acceptance rate.
9. Ioan Raicu. "*A Performance Study of the Globus Toolkit® and Grid Services via DiPerF, an automated DIstributed PERformance testing Framework*", University of Chicago, Computer Science Department, MS Thesis, May 2005, Chicago, Illinois.
10. Ioan Raicu, Loren Schwiebert, Scott Fowler, Sandeep K.S. Gupta. *"Local Load Balancing for Globally Efficient Routing in Wireless Sensor Networks"*, International Journal of Distributed Sensor Networks, 1: 163–185, 2005.
11. Ioan Raicu, Loren Schwiebert, Scott Fowler, Sandeep K.S. Gupta. *"e3D: An Energy-Efficient Routing Algorithm for Wireless Sensor Networks"*, IEEE ISSNIP 2004 (The International Conference on Intelligent Sensors, Sensor Networks and Information Processing), Melbourne, Australia, December 2004; top 10% of conference papers, extended version published in International Journal of Distributed Sensor Networks 2005.
12. Catalin Dumitrescu, Ioan Raicu, Matei Ripeanu, Ian Foster. "*DiPerF: an automated DIstributed PERformance testing Framework*", IEEE/ACM GRID2004, Pittsburgh, PA, November 2004, pp 289 - 296; 22% acceptance rate
13. Sheralli Zeadally, R. Wasseem, Ioan Raicu, "*Comparison of End-System IPv6 Protocol Stacks*", IEE Proceedings Communications, Special issue on Internet Protocols, Technology and Applications (VoIP), Vol. 151, No. 3, June 2004.
14. Sherali Zeadally, Ioan Raicu. "*Evaluating IPV6 on Windows and Solaris*", IEEE Internet Computing, Volume 7, Issue 3, May June 2003, pp 51 – 57.
15. Ioan Raicu, Sherali Zeadally. "*Impact of IPv6 on End-User Applications*", IEEE International Conference on Telecommunications 2003, ICT'2003, Volume 2, Feb 2003, pp 973 - 980, Tahiti Papeete, French Polynesia; 35% acceptance rate.
16. Ioan Raicu, Sherali Zeadally. "*Evaluating IPv4 to IPv6 Transition Mechanisms*" , IEEE International Conference on Telecommunications 2003, ICT'2003, Volume 2, Feb 2003, pp 1091 - 1098, Tahiti Papeete, French Polynesia; 35% acceptance rate.
17. Ioan Raicu. "*Efficient Even Distribution of Power Consumption in Wireless Sensor Networks*", ISCA 18th International Conference on Computers and Their Applications, CATA 2003, 2003, Honolulu, Hawaii, USA.
18. Ioan Raicu. "*An Empirical Analysis of Internet Protocol version 6 (IPv6)*", Wayne State University, Computer Science Department, MS Thesis, May 2002, Detroit, Michigan.
19. Ioan Raicu. "*Routing Algorithms for Wireless Sensor Networks*" Grace Hopper Celebration of Women in Computing 2002, GHC2002, 2002, British Columbia, Canada.
20. Ioan Raicu, Owen Richter, Loren Schwiebert, Sherali Zeadally. "*Using Wireless Sensor Networks to Narrow the Gap between Low-Level Information and Context-Awareness*", Proceedings of the ISCA 17th International Conference, Computers and their Applications, San Francisco, CA, 2002.

Service
(2002 – Present)

- Megajobs BOF: How to Run One Million Jobs, at IEEE/ACM Supercomputing 2008
- IEEE/ACM Workshop on Grid Computing Portals and Science Gateways (GCE08)
- IEEE International Conference on Internet and Web Applications and Services (ICIW 2009)
- IEEE/ACM Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS), co-located with IEEE/ACM Supercomputing 2008
- TeraGrid Conference (TG09)
- IEEE International Conference on Networks (ICN 2009)
- IEEE International Conference on Networking and Services (ICNS 2009)
- Distributed Systems Laboratory Workshop (DSLW08)
- IEEE International Conference on Internet and Web Applications and Services (ICIW08)
- Sixth Annual Conference on Communication Networks and Services Research (CNSR08)
- The Handbook of Technology Management (book to appear in 2008)
- TeraGrid Conference (TG08)
- ACM/IET/ICST International Workshop on Performance and Analysis of Wireless Networks (PAWN08)
- IEEE International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP08)
- IEEE International Conference on Systems and Networks Communications (ICNSC08)
- IEEE International Conference on Networking and Services (ICNS08)
- IEEE International Conference on Networking (ICN08)
- IEEE Internet Computing, Special Issue on Virtual Organizations, 2007
- IEEE/ACM Workshop on Grid Computing Portals and Science Gateways (GCE07)
- IEEE/ACM Grid Conference (SC07)
- Distributed Systems Laboratory Workshop (DSLW07)
- IEEE Internet Computing (IC07)
- The Handbook of Computer Networks (2007)
- IEEE/ACM SuperComputing (SC06)
- Distributed Systems Laboratory Workshop (DSLW06)
- IEEE Transactions on Computers (TC06)
- Journal of Concurrency and Computation: Practice and Experience 2006
- IEEE Communication Letters (CL05)
- High Performance Computing Symposium (HPCC05)
- IEEE Intelligent Sensing and Information Processing (ICISIP05)
- ARC Research Network on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP05)
- IEEE International Conference on Computer Communications and Networks (IC3N02)
- IEEE International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS02)


MTAGS08: Workshop on Many-Task Computing on Grids and Supercomputers - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://dsl.cs.uchicago.edu/MTAGS08/
Google
Most Visited iGoogle Google Ioan Raicu's Web Site New Page 2 H2O - CMSC 23340 S... Compare.aspx LinkedIn
mtags08 Search Settings
Wiki:People - CSWiki
CCA08: Cloud Computing and Its Applic...
MTAGS08: Workshop on Many-Ta...
MTAGS
Workshop on Many-Task Computing on Grids and Supercomputers
co-located with ACM/IEEE SC08 (International Conference for High Performance, Networking, Storage and Analysis)
Austin, Texas -- November 17th, 2008
Home
Call for Papers
Program Committee
Important Dates
Paper Submission
Venue
Registration
Workshop Program
Important Dates
Papers Due: August 15th, 2008
Notification of Acceptance: October 1st, 2008
Camera Ready Papers Due: October 15th, 2008
Workshop Date: November 17th, 2008
Committee Members
Workshop Chairs
Yong Zhao, Microsoft
Ian Foster, University of Chicago & Argonne National Laboratory
Ioan Raicu, University of Chicago
Technical Committee
Ian Foster, University of Chicago & Argonne National Laboratory
Dan Ardelean, Google
Bob Grossman, University of Illinois at Chicago
Indranil Gupta, University of Illinois at Urbana Champaign
Tevfik Kosar, Louisiana State University
Done


Main Page - MegajobBOF - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://gridfarm007.ucs.indiana.edu/megajobBOF/index.php/Main_Page
Most Visited iGoogle Ioan Raicu's Web Site MTAGS08: Workshop o... Dashboard - Google An... Incubator/Falkon - Glo... Outreach/SC2008 - Glo... CiteSeerX Google Scholar LinkedIn
Incubator/Falkon - Globus
Ioan Raicu's Web Site
Main Page - MegajobBOF
Iraicu my talk preferences my watchlist my contributions log out
article discussion edit history move unwatch
bof information
Main Page
Presenters and Abstracts
logistics
presenters
organizers
wiki tools
Current events
Recent changes
Help
search
Go Search
toolbox
What links here
Related changes
Upload file
Special pages
Printable version
Main Page
Megajobs: How to Run One Million Jobs
[edit]
What: Birds-of-a-Feather Session at Supercomputing 2008, Austin Texas
Date: Tuesday, November 18th, 2008
Time: 05:30PM - 07:00PM
Location: Room 13A/13B
Primary Session Leader:
Marlon Pierce (Indiana University)
Secondary Session Leader:
Ioan Raicu (University of Chicago)
Ruth Pordes (Fermi National Laboratory)
John McGee (Renaissance Computing Institute)
Dick Repasky (Indiana University)
As large systems surpass 200K CPU cores and as applications increase in complexity, more scientists need to run thousands to millions of closely related jobs that are associated with individual projects. Scientists seek convenient means to specify and manage many jobs, arranging inputs, aggregating outputs, identifying successful and failed jobs and repairing failures. System administrators seek methods to process extraordinary numbers of jobs for multiple users without overwhelming queuing systems or disrupting fair-share usage policies. Under development are a new generation of queuing and scheduling systems and multi-level schedulers for use with existing queuing and scheduling systems, schedulers designed to handle millions of jobs. This Birds-of-feather session provides a venue for the exchange of information about processing large numbers of jobs. Short presentations of an invited sample of projects will be followed by discussion.
We are currently soliciting participation in the "Megajobs" BOF. We are looking for short, piquant presentations (5-10 minutes) from people who have worked on this problem or have a problem like this that needs to be worked on. If you are interested, please send a brief title and abstract (250 words) to Marlon Pierce by October 27th, 2008. Please feel free to contact us if you have questions.
For the latest information hosted by SC08, see http://scyourway.nacse.org/conference/view/bof118. The Megajobs BOF handout can also be found here.
Related activities at SC08, that might be of interest to BOF attendees are:
Grid Computing Environments (GCE)
Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS)
Done