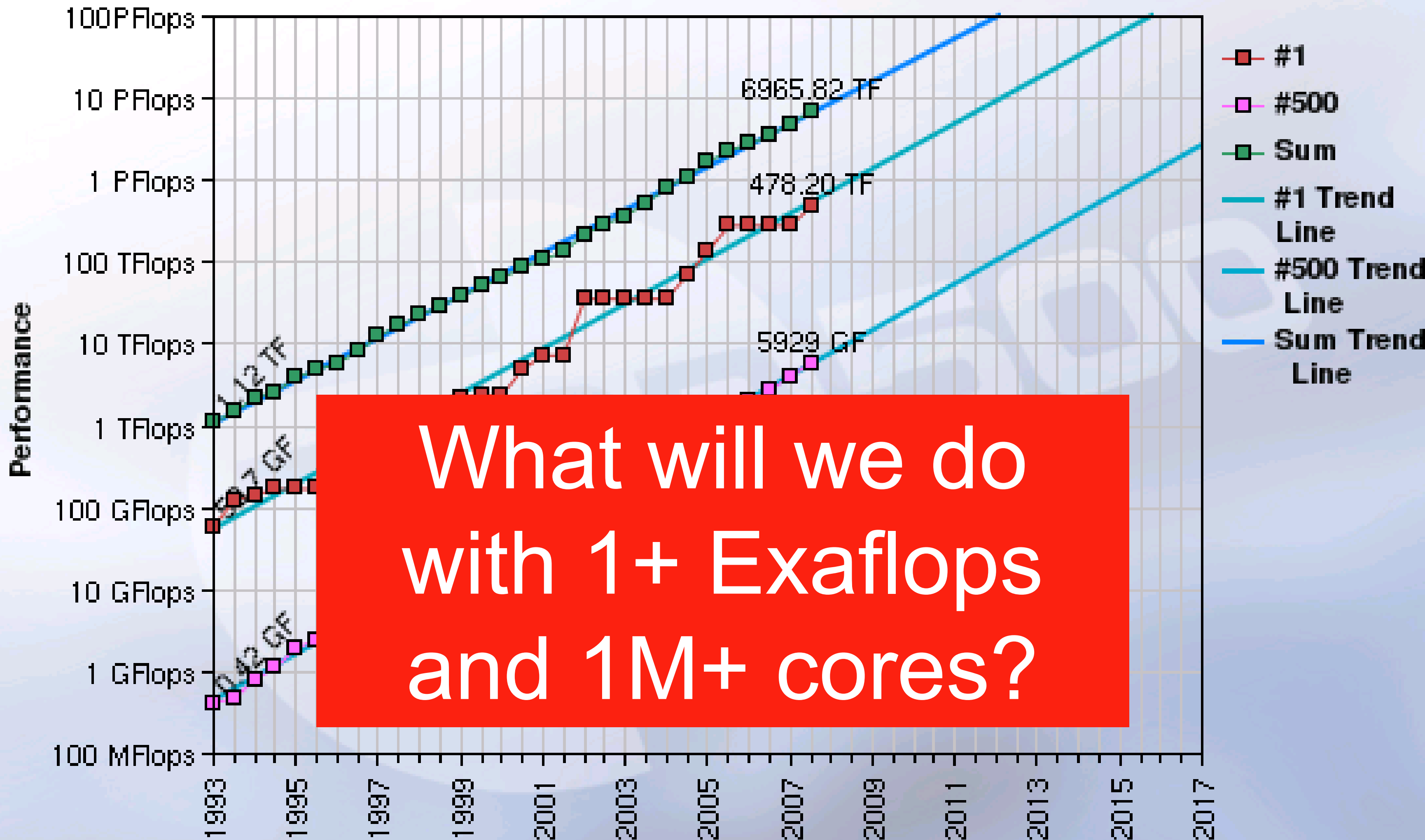
Harnessing Grid Resources with Data Data-Centric Task Farms

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
+24 more people, see "Most Recent Collaborators Relevant to Dissertation " slide at the end…

August 20th, 2008

Projected Performance Development



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

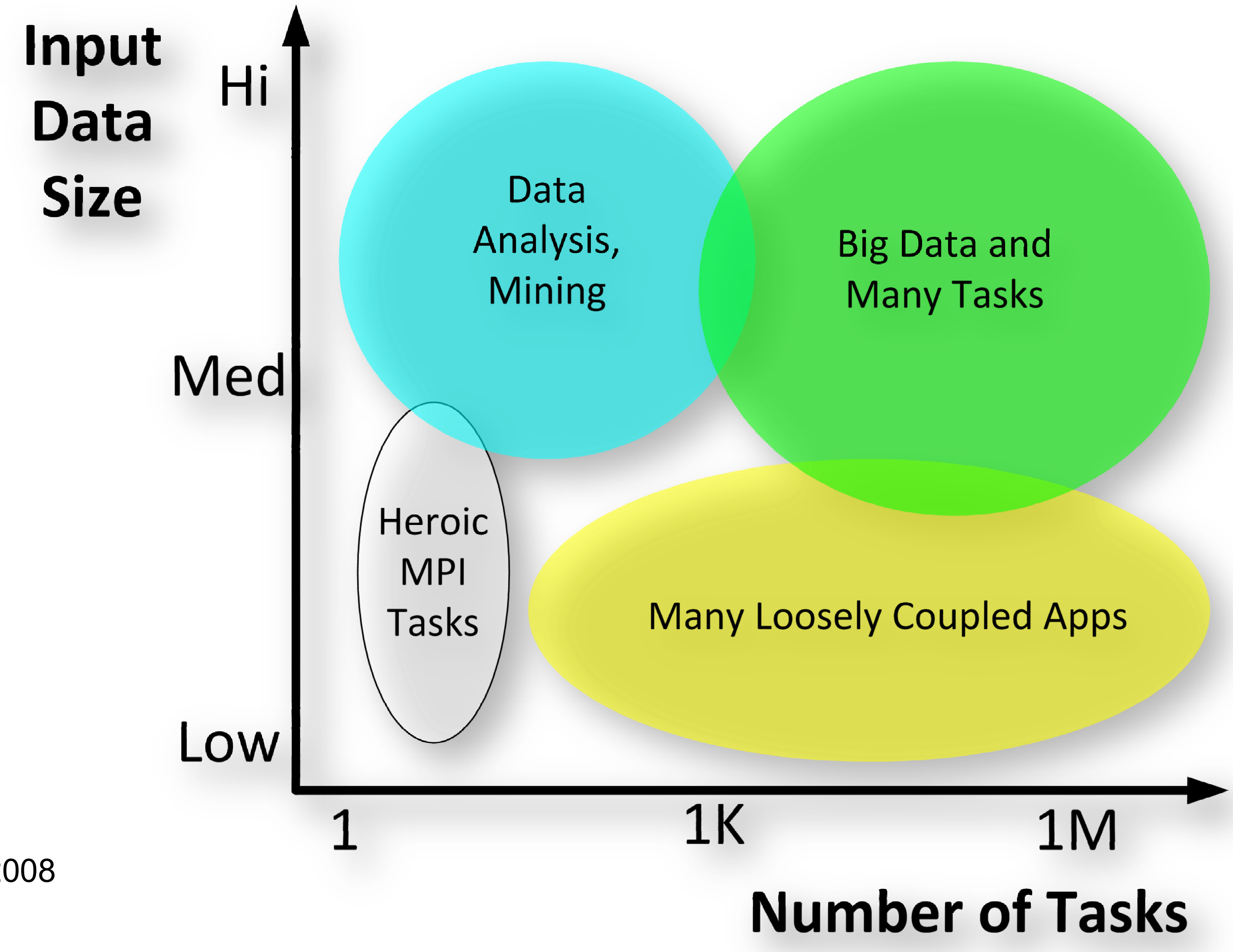
Programming Model Issues

- **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

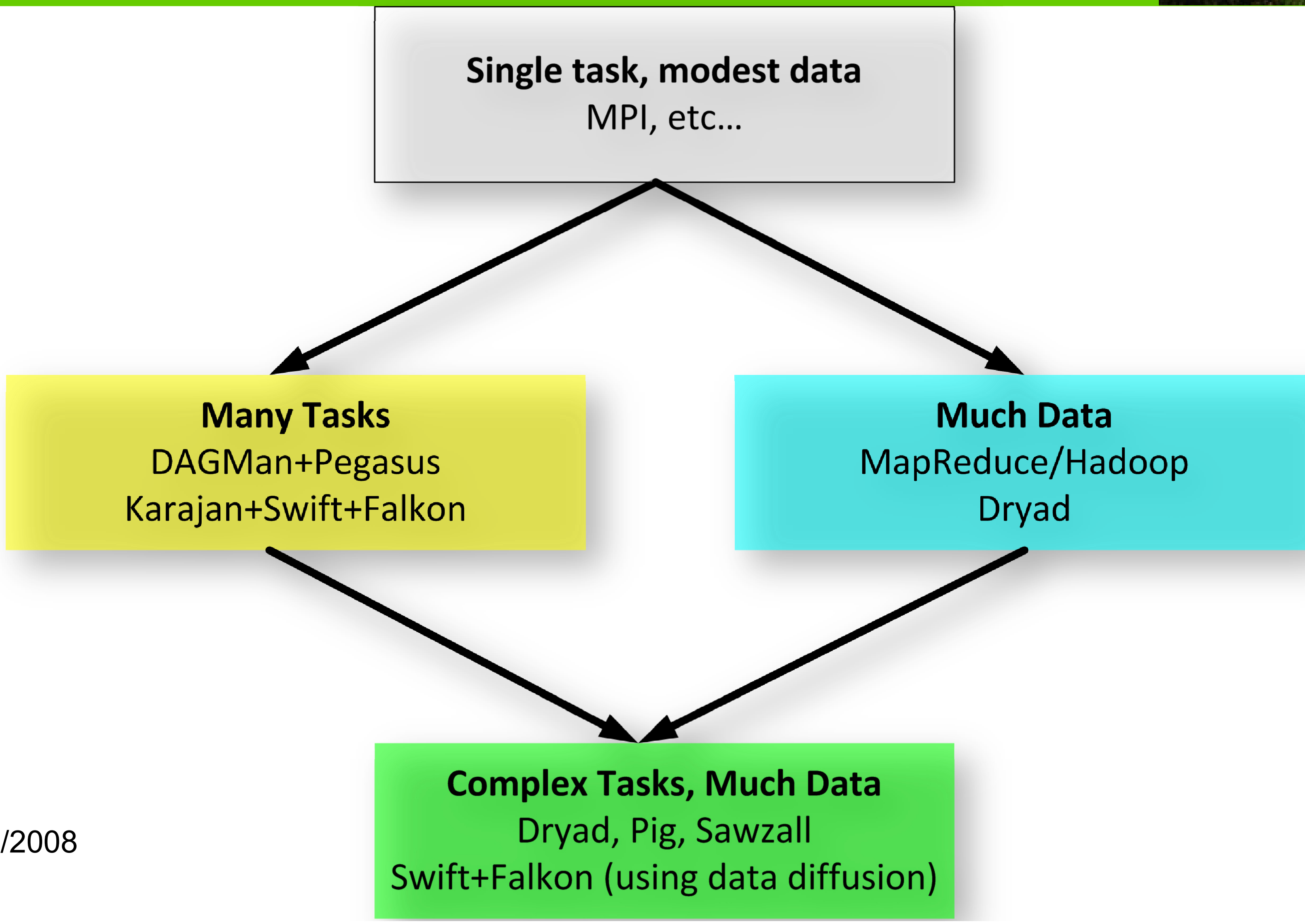
Programming Model Issues

- **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



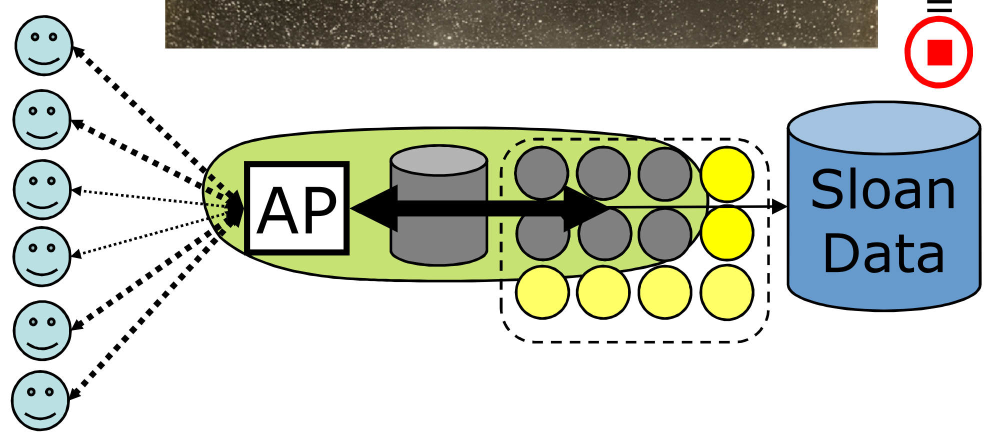
MTC: Many Task Computing

- 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…

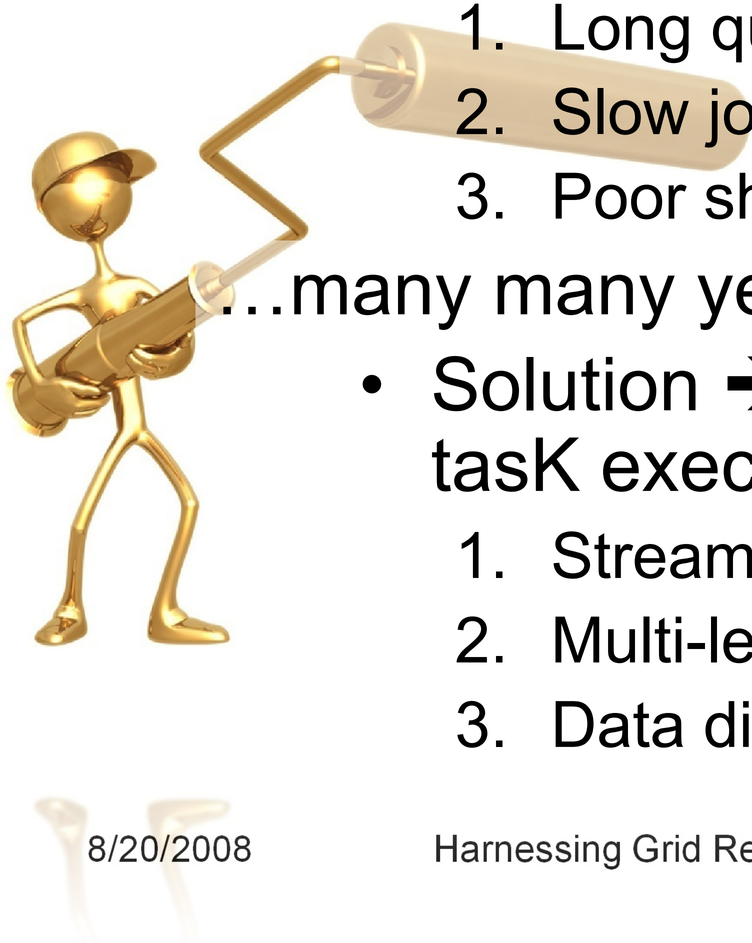
Motivating Example: AstroPortal Stacking Service

- 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



- Obstacles:
  1. Long queue times
  2. Slow job dispatch rates
  3. Poor shared file system scaling

…many many years of hard work…

- Solution ➔ Falkon: a Fast and Light-weight tasK executiON framework
  1. Streamlined dispatching
  2. Multi-level scheduling
  3. Data diffusion

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**

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

- **Speedup**

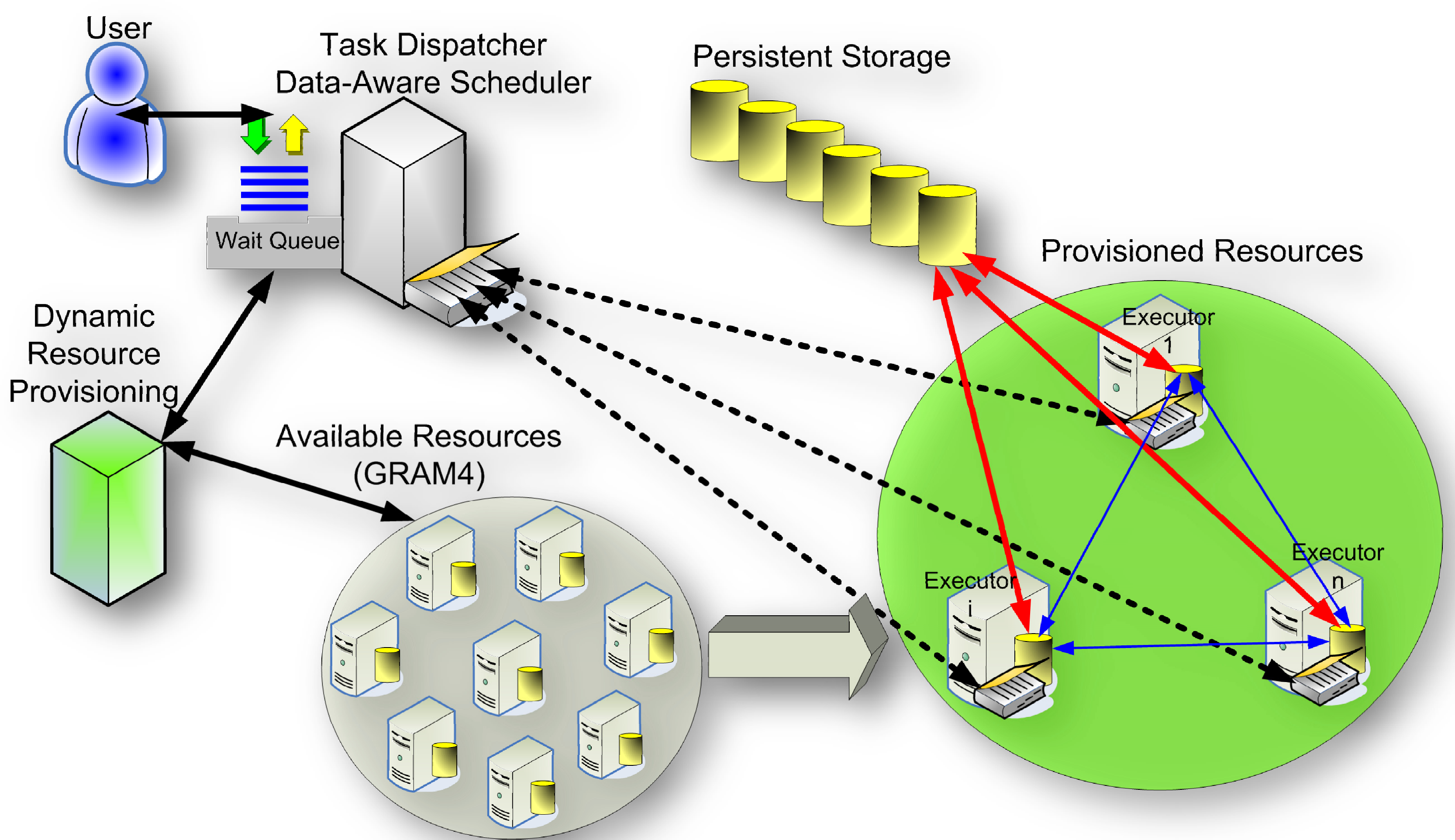
$$S = E*|T|$$

- **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

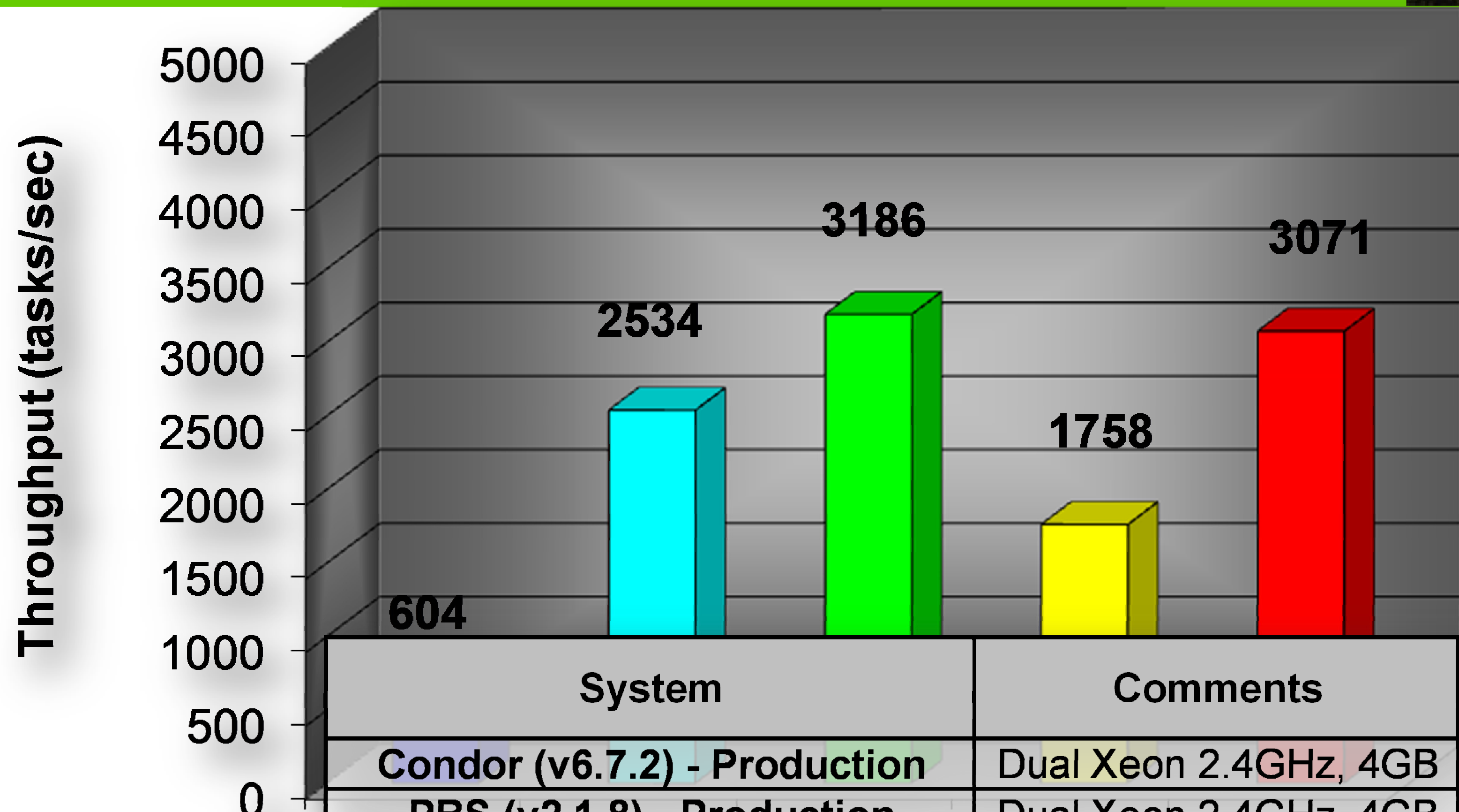
Falkon: a Fast and Light-weight tasK executiON framework

- ***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

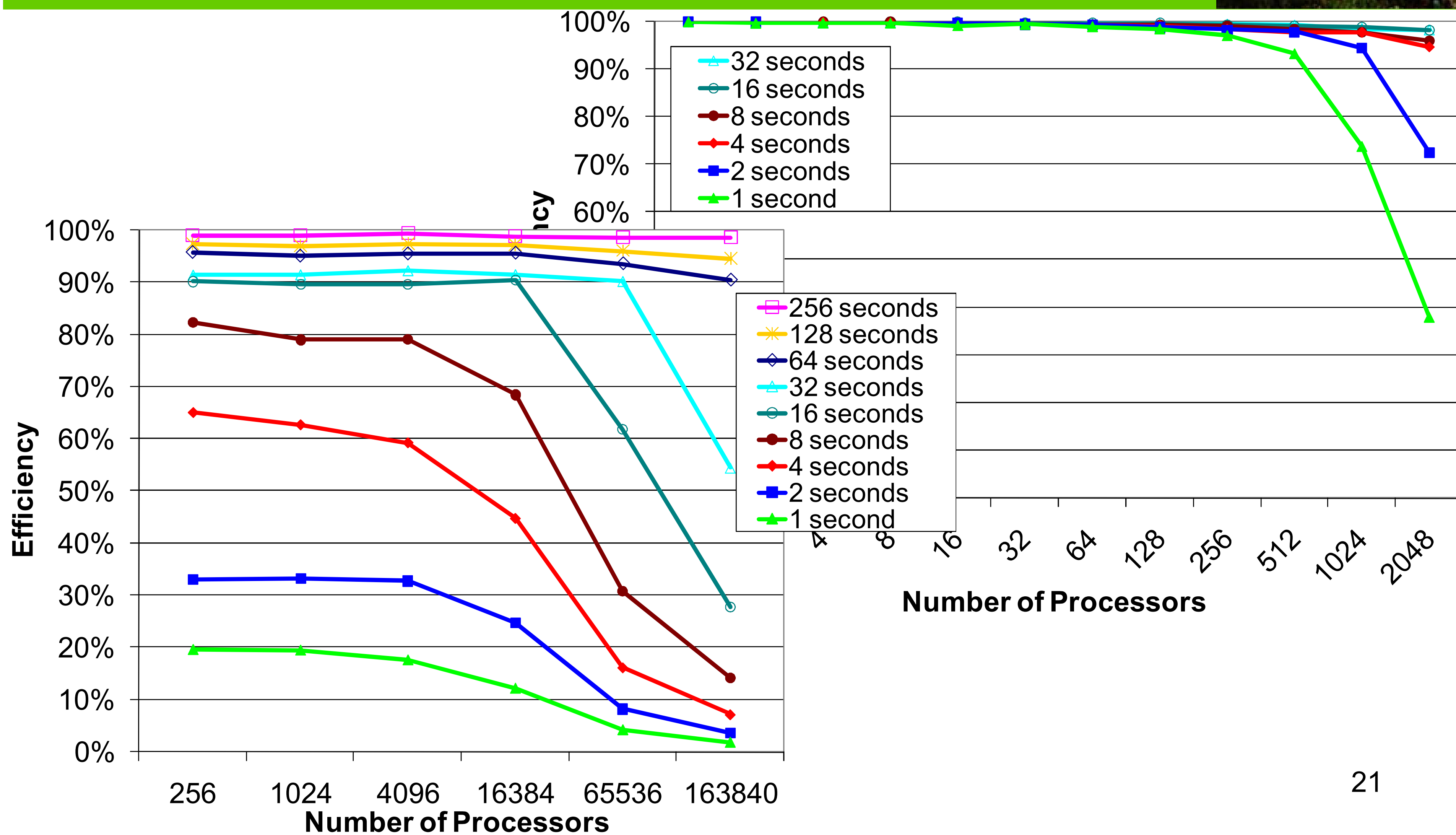


Dispatch Throughput

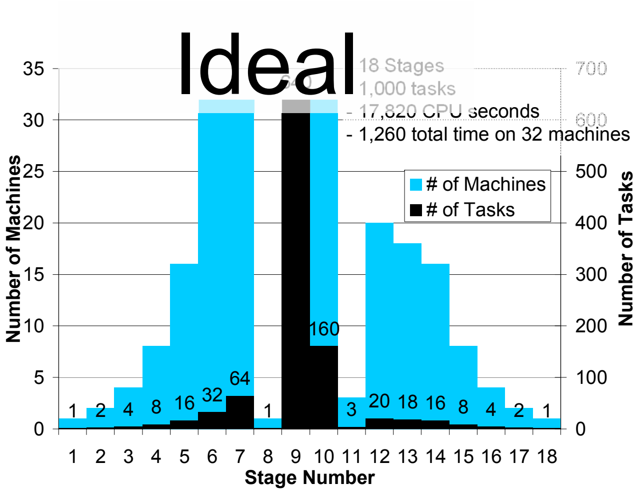


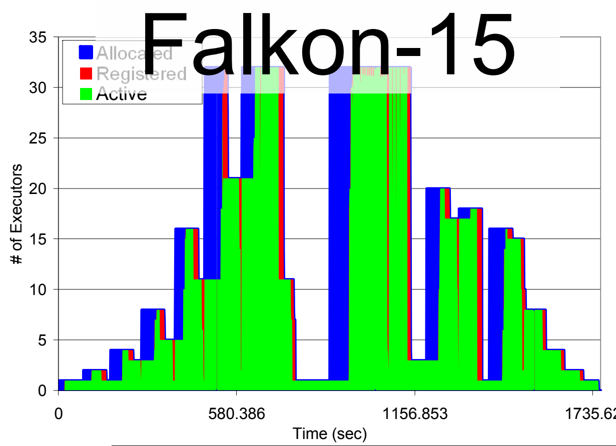
System	Comments	Throughput (tasks/sec)
Condor (v6.7.2) - Production	Dual Xeon 2.4GHz, 4GB	0.49
PBS (v2.1.8) - Production	Dual Xeon 2.4GHz, 4GB	0.45
Condor (v6.7.2) - Production	Quad Xeon 3 GHz, 4GB	2
Condor (v6.8.2) - Production		0.42
Condor (v6.9.3) - Development		11
Condor-J2 - Experimental	Quad Xeon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

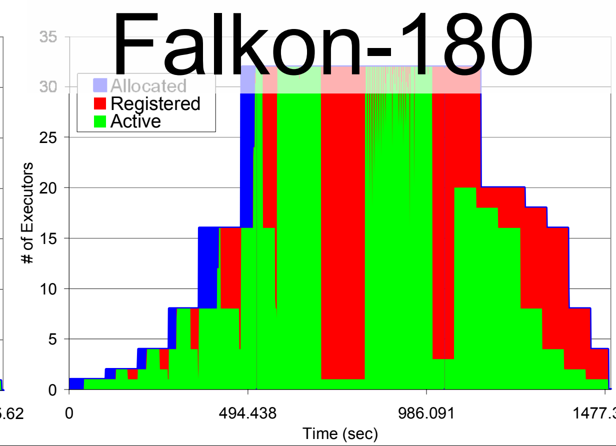
Efficiency


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





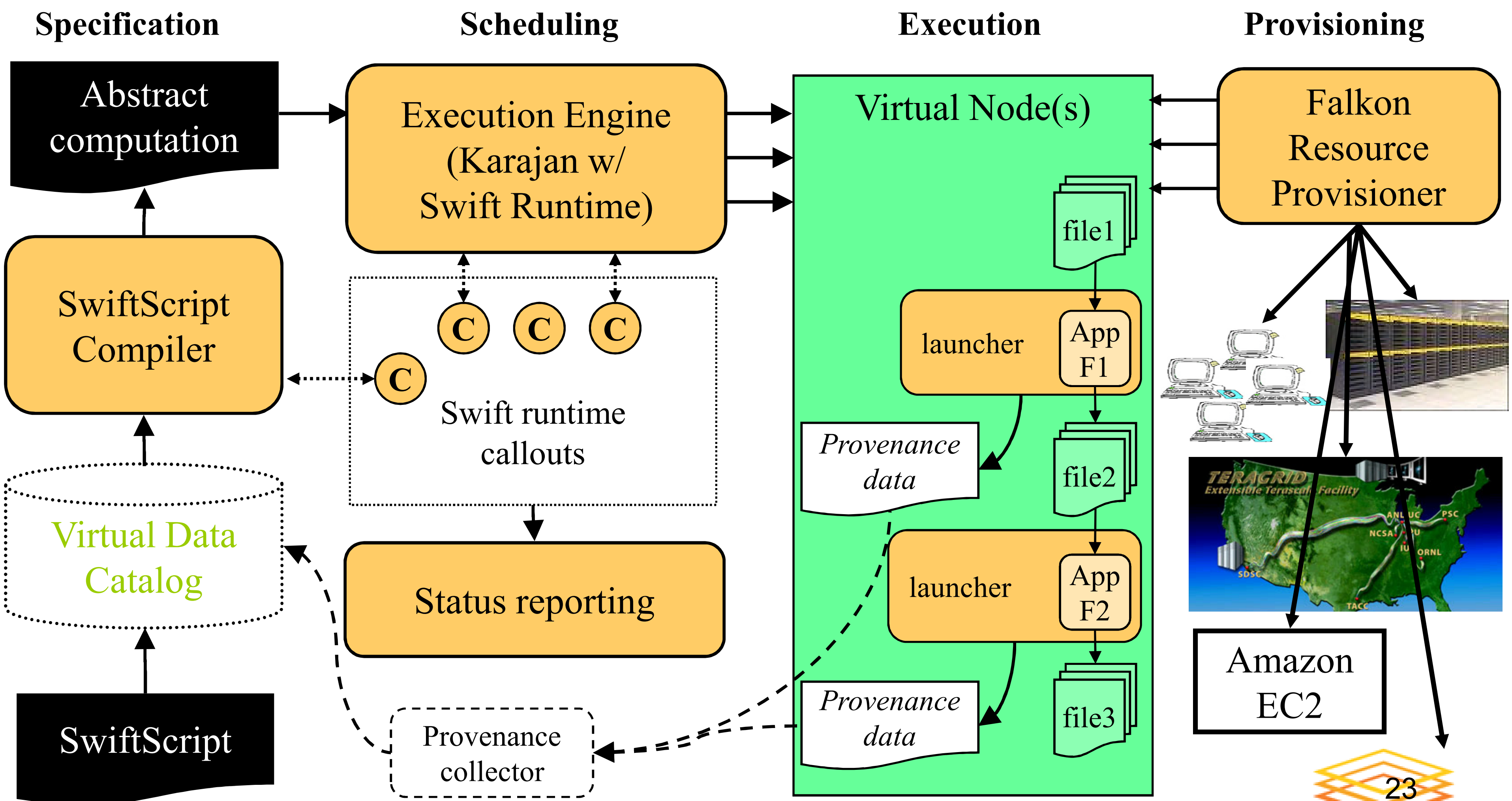


- End-to-end execution time:
  - 1260 sec in ideal case
  - 4904 sec → 1276 sec
- Average task queue time:
  - 42.2 sec in ideal case
  - 611 sec → 43.5 sec
- Trade-off:
  - Resource Utilization for Execution Efficiency

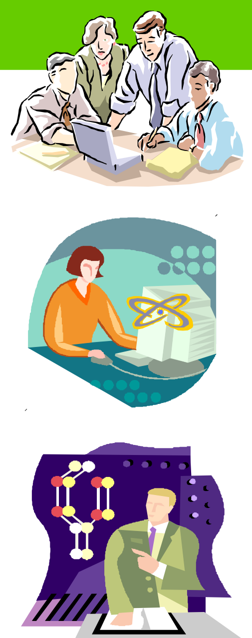
	GRAM +PBS	Falkon-15	Falkon-60	Falkon-120	Falkon-180	Falkon-∞	Ideal (32 nodes)
Queue Time (sec)	611.1	87.3	83.9	74.7	44.4	43.5	42.2
Execution Time (sec)	56.5	17.9	17.9	17.9	17.9	17.9	17.8
Execution Time %	8.5%	17.0%	17.6%	19.3%	28.7%	29.2%	29.7%

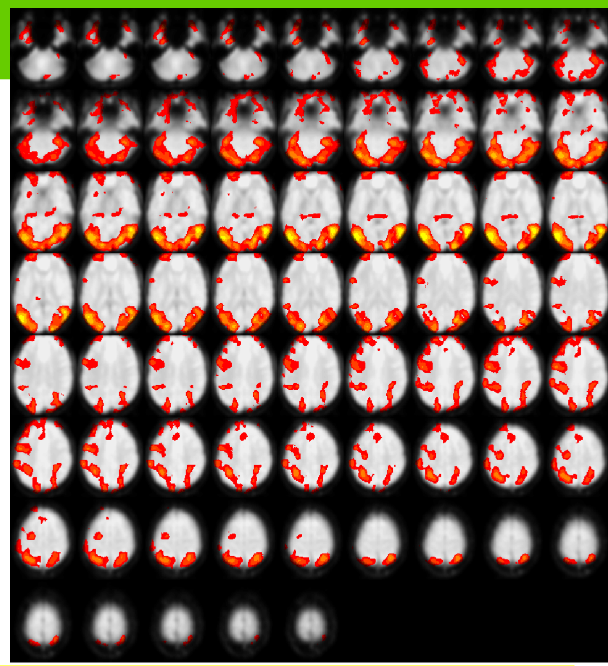
	GRAM +PBS	Falkon-15	Falkon-60	Falkon-120	Falkon-180	Falkon-∞	Ideal (32 nodes)
Time to complete (sec)	**4904**	**1754**	**1680**	**1507**	**1484**	**1276**	**1260**
Resouce Utilization	30%	89%	75%	65%	59%	44%	100%
Execution Efficiency	26%	72%	75%	84%	85%	99%	100%
Resource Allocations	1000	11	9	7	6	0	0

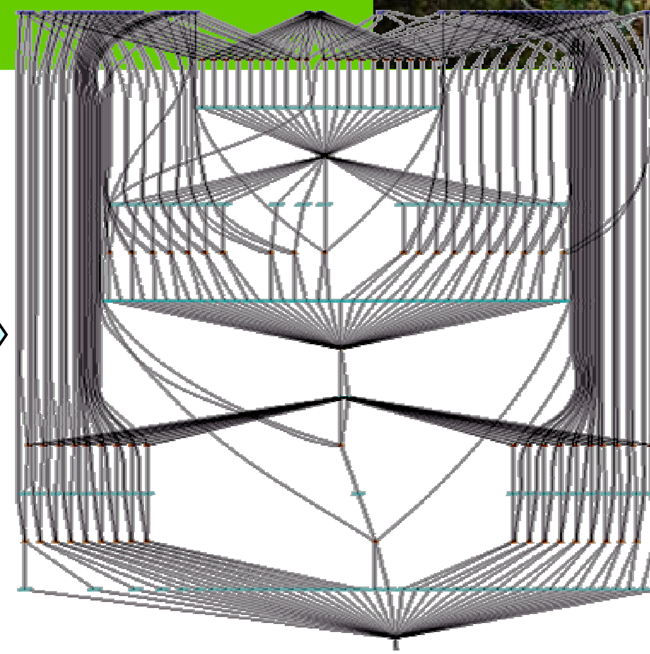
Swift Architecture

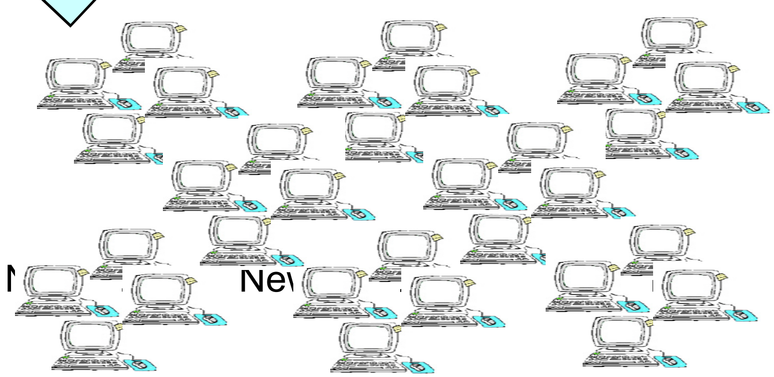


Functional MRI (fMRI)





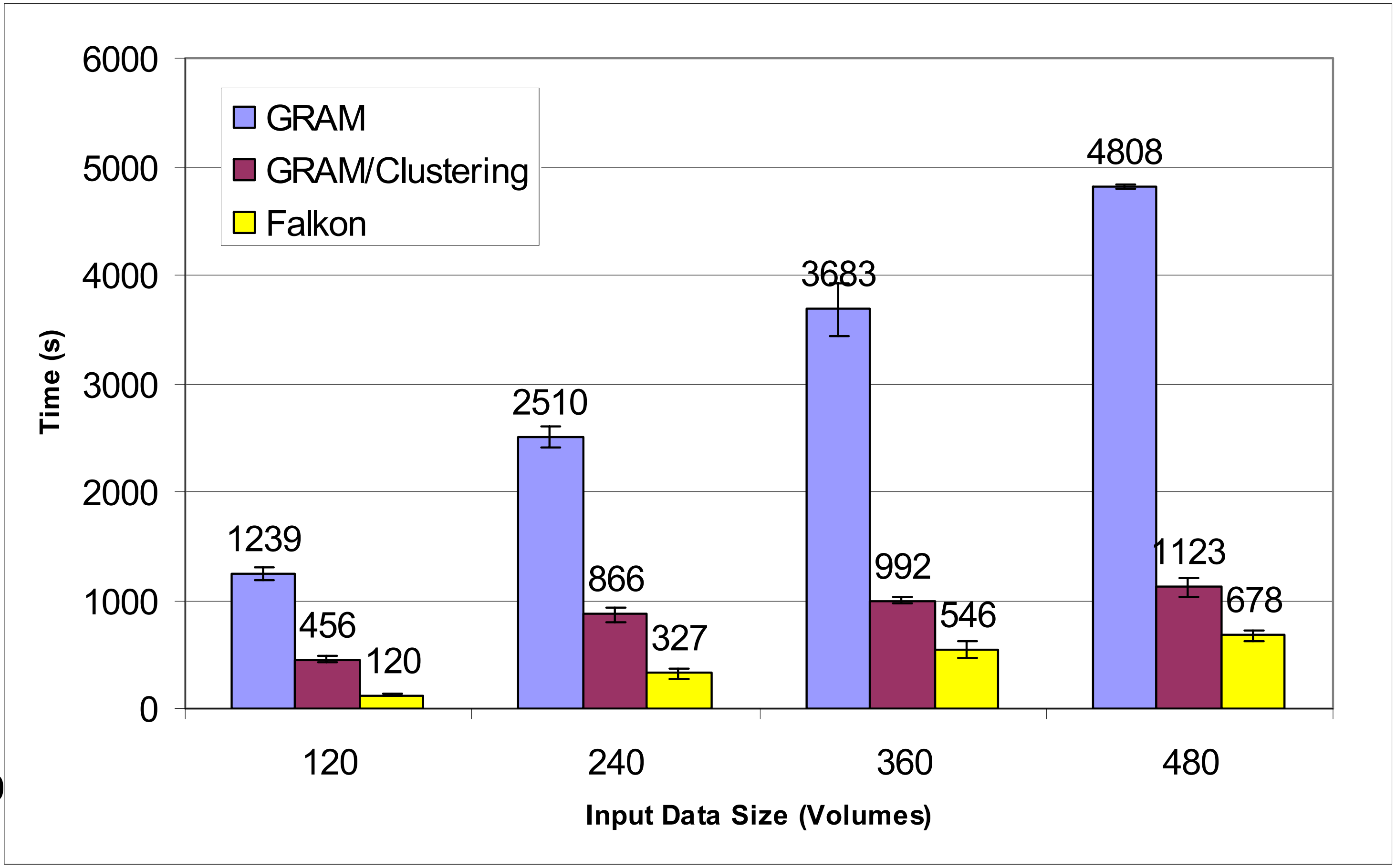


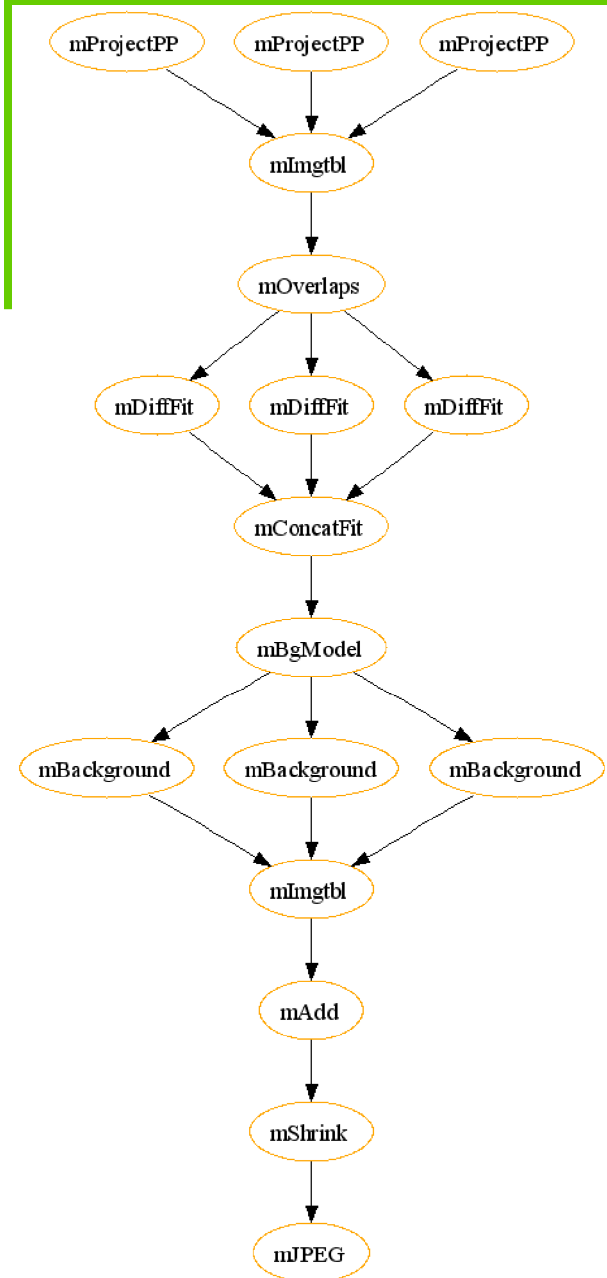


- Wide range of analyses
  - Testing, interactive analysis, production runs
  - Data mining
  - Parameter studies

Completed Milestones: fMRI Application

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



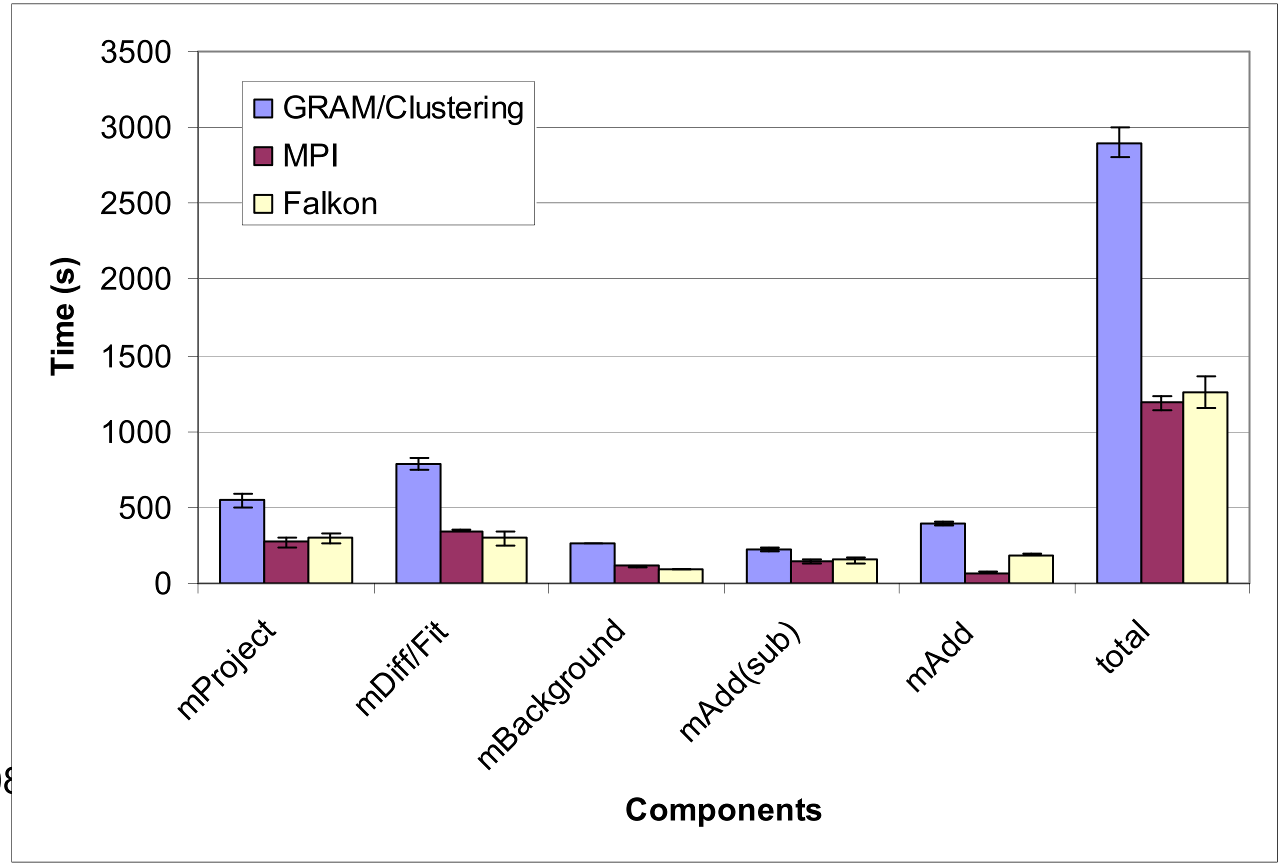


Montage

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

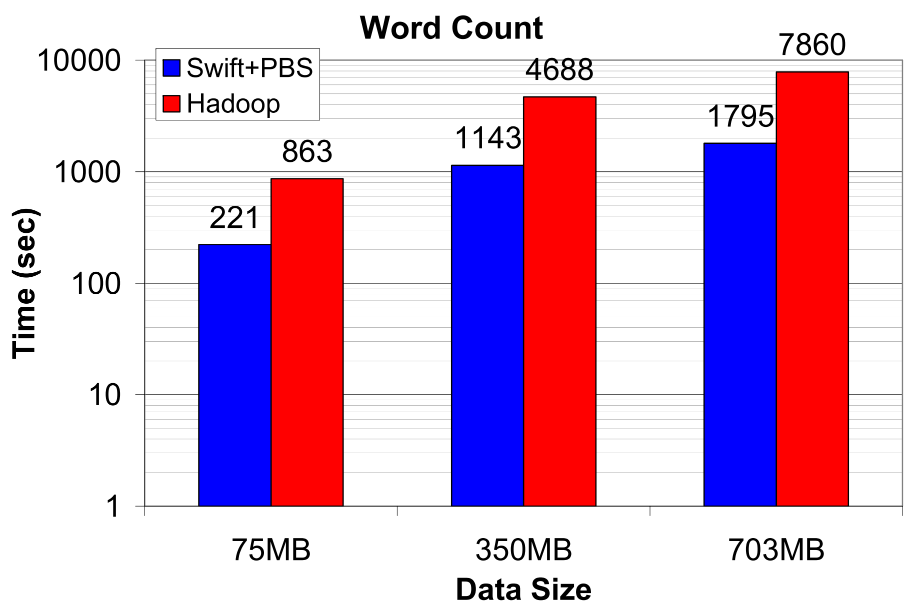
Completed Milestones: Montage Application

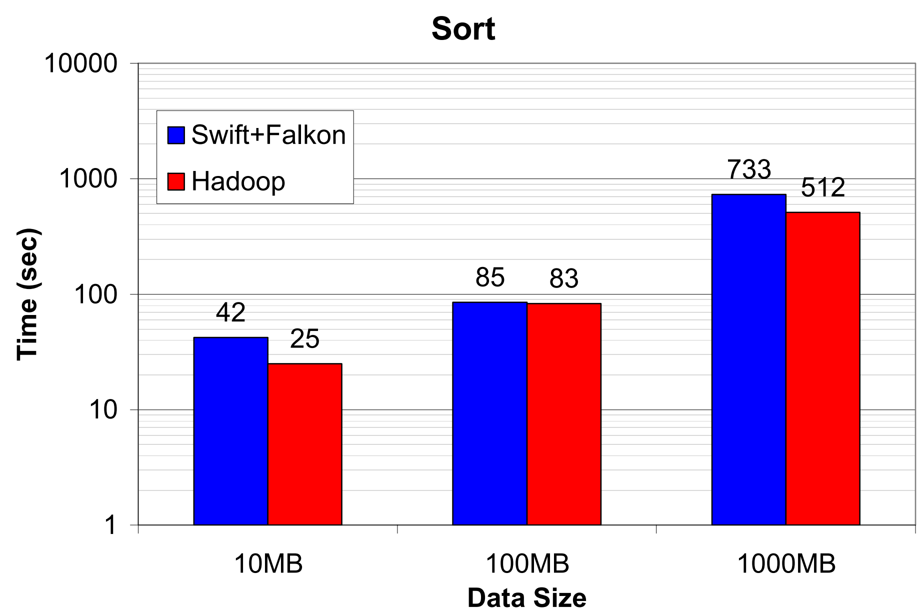
- GRAM/Clustering vs. Falkon: 57% lower application run time
- MPI* vs. Falkon: 4% higher application run time
- * MPI should be lower bound



Hadoop vs. Swift

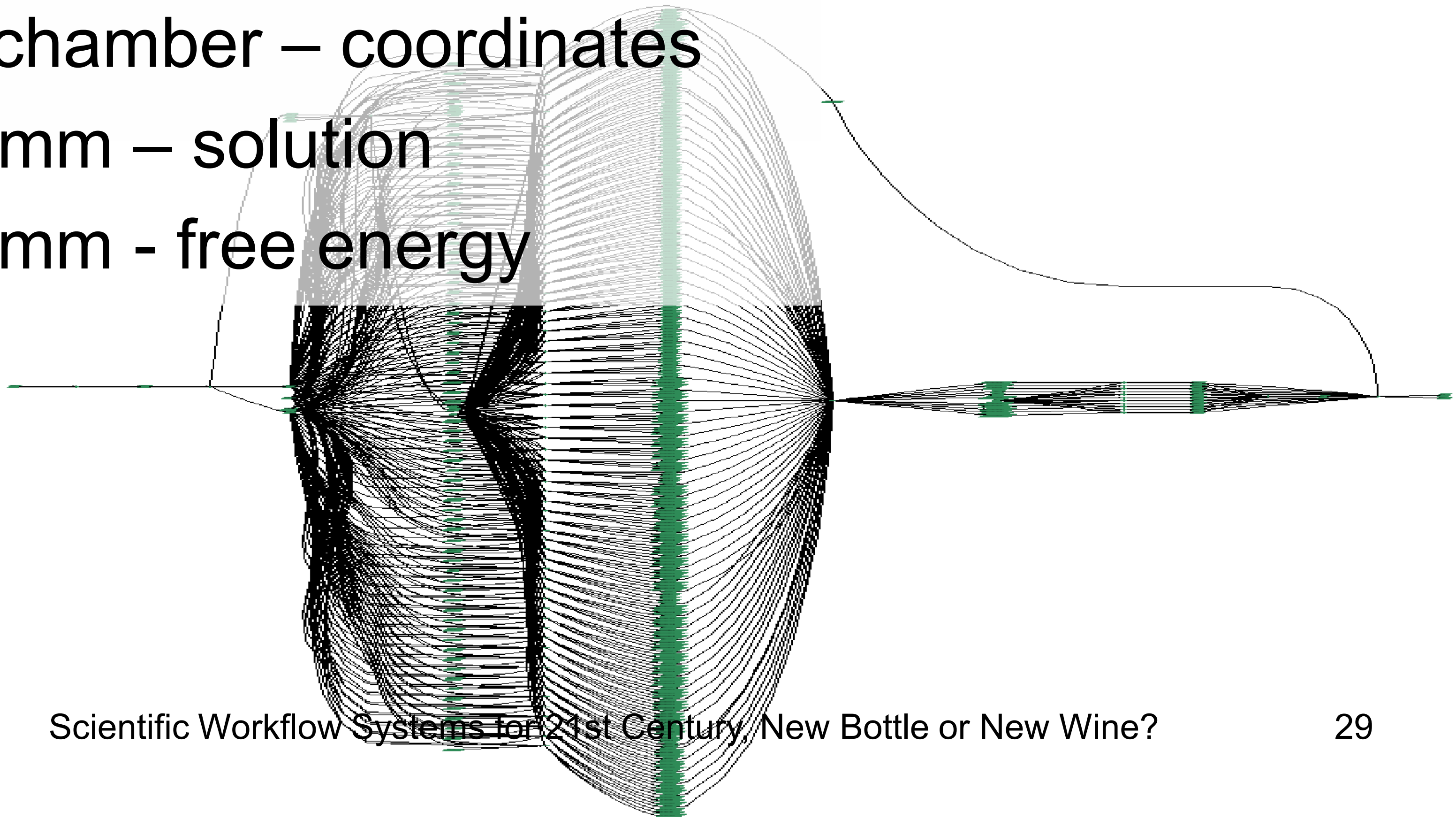
- Classic benchmarks for MapReduce
  - Word Count
  - Sort
- Swift performs similar or better than Hadoop (on 32 processors)





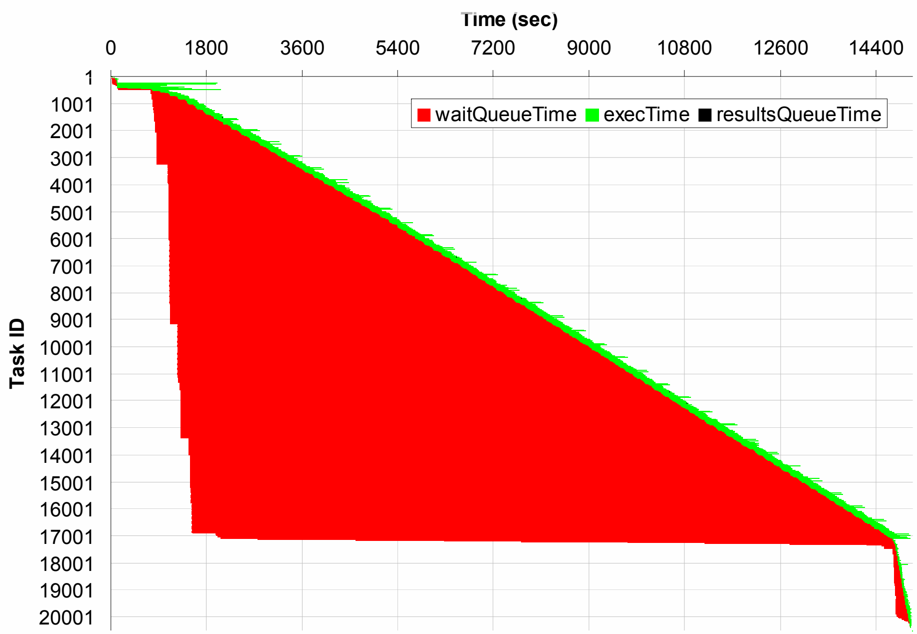
Molecular Dynamics

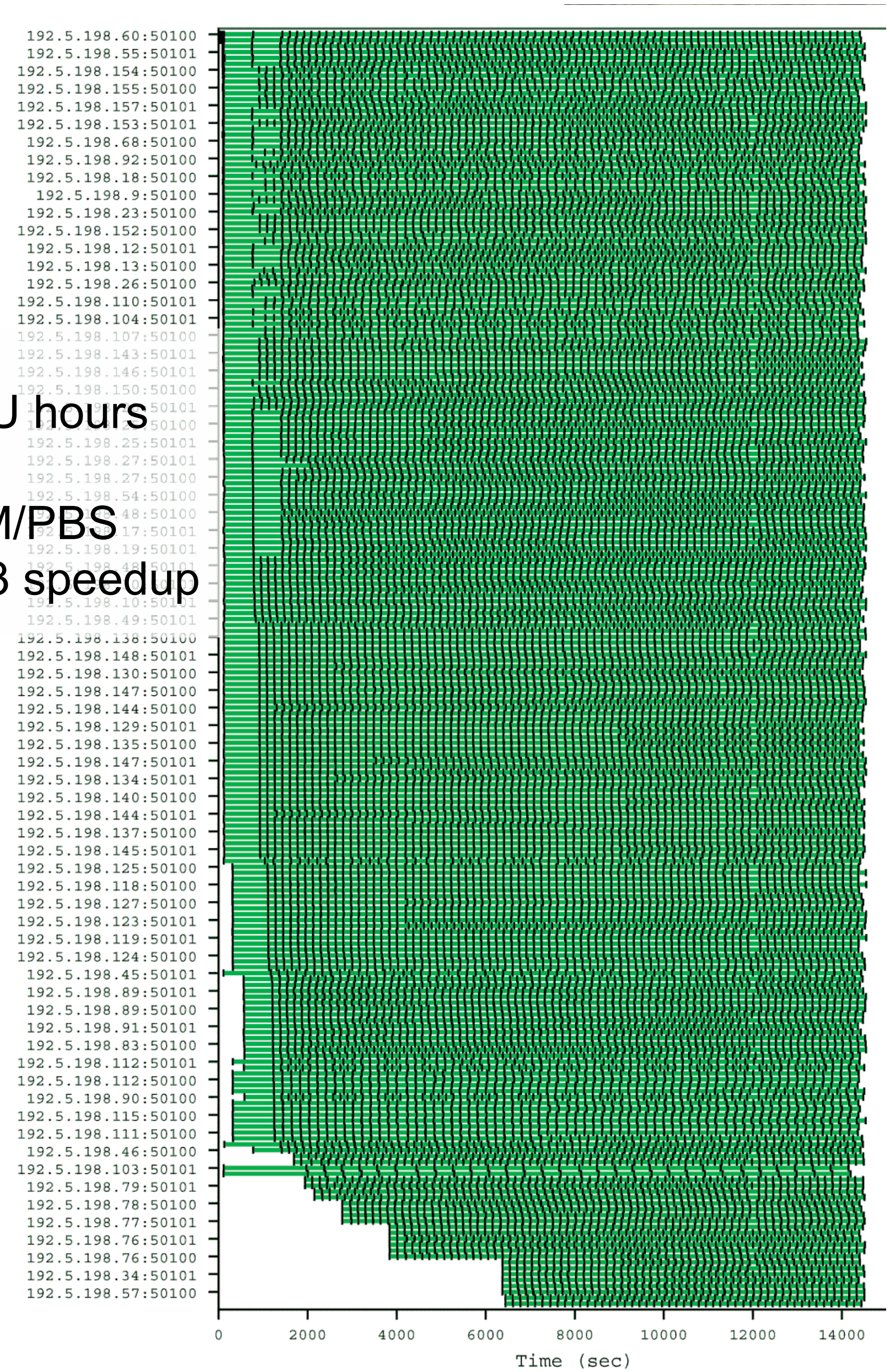
- Determination of free energies in aqueous solution
  - Antechamber – coordinates
  - Charmm – solution
  - Charmm - free energy



MolDyn Application

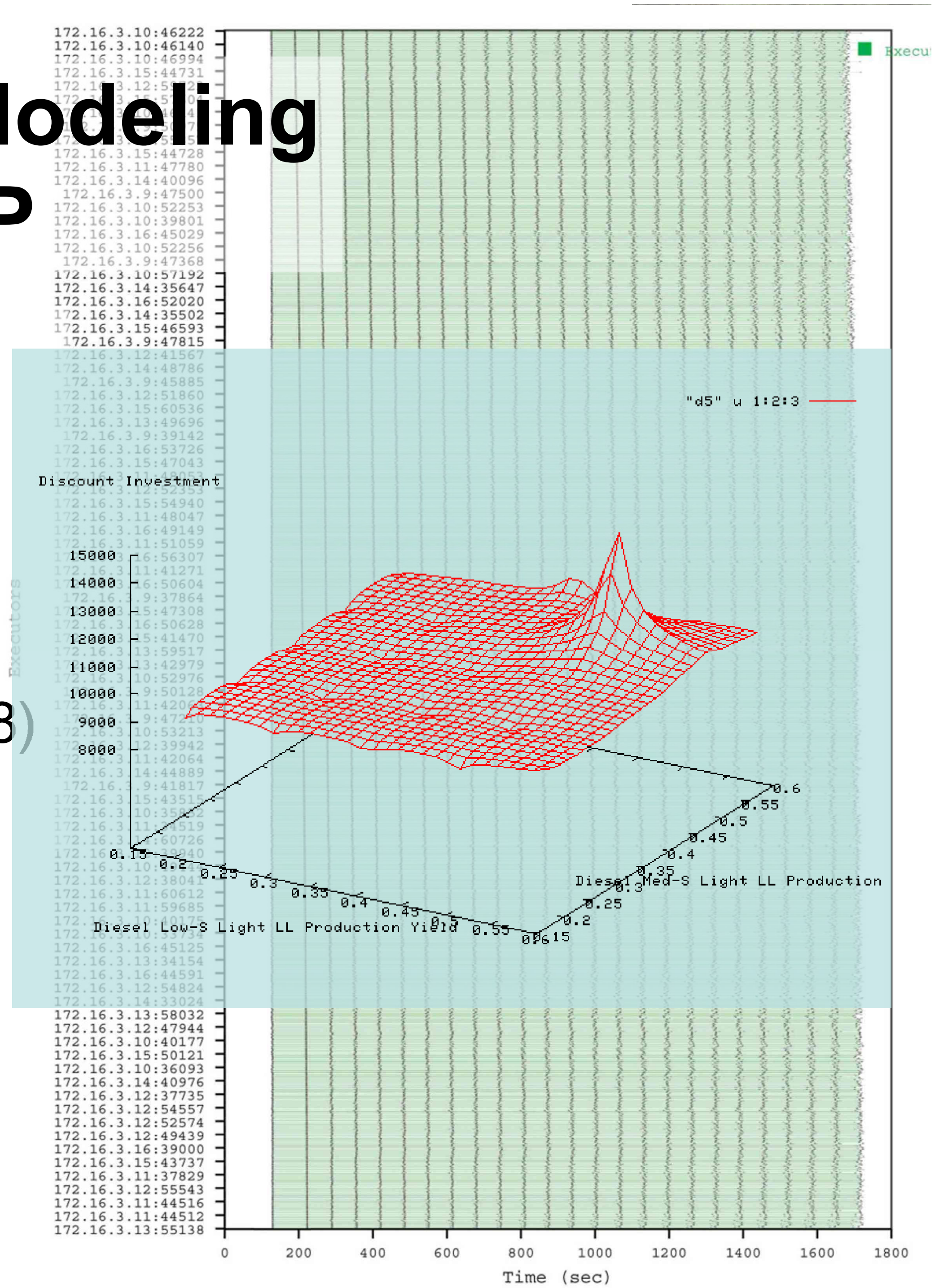
- 244 molecules → 20497 jobs
- 15091 seconds on 216 CPUs → 867.1 CPU hours
- Efficiency: 99.8%
- Speedup: 206.9x → 8.2x faster than GRAM/PBS
- 50 molecules w/ GRAM (4201 jobs) → 25.3 speedup





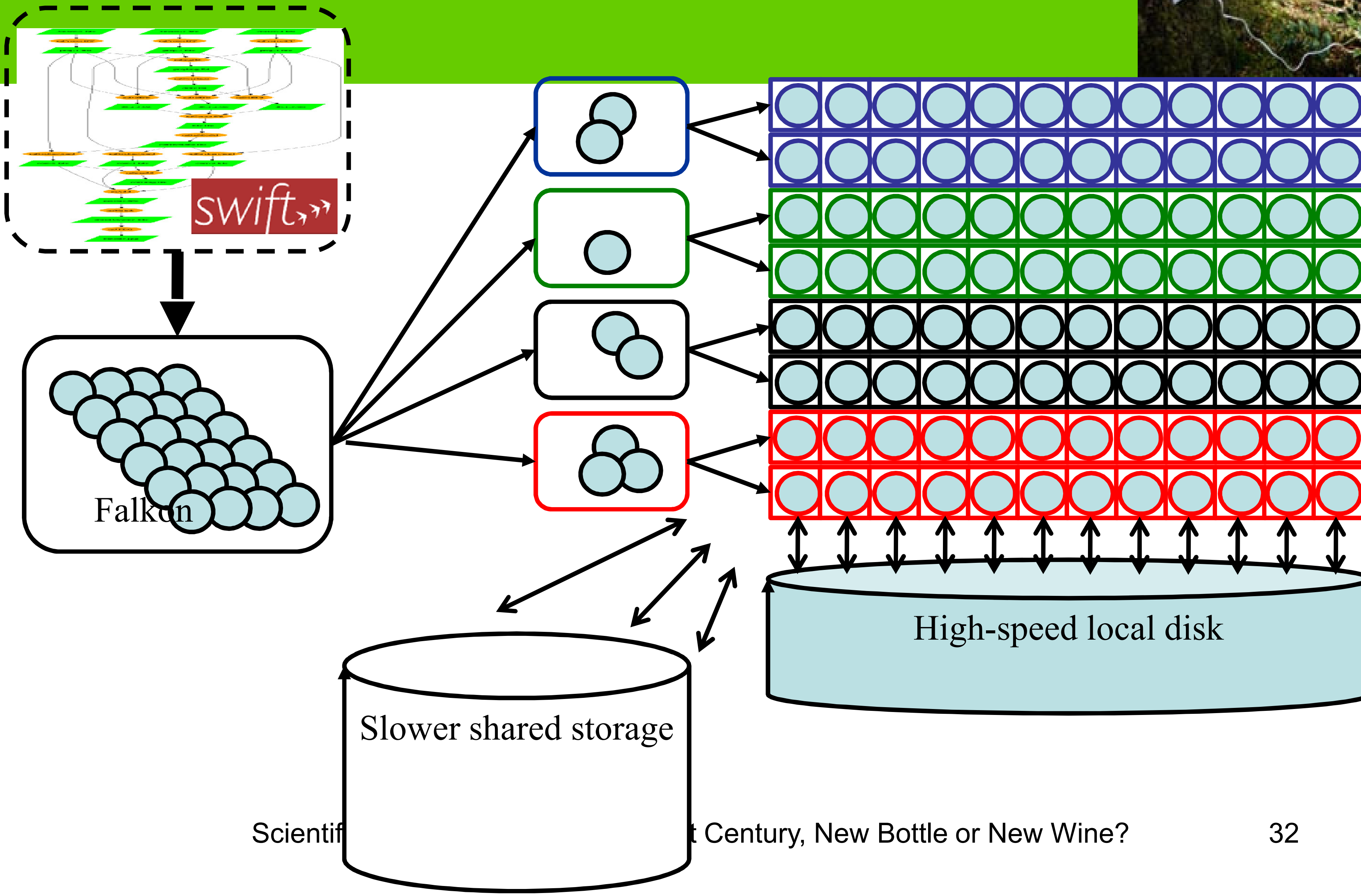
MARS Economic Modeling on IBM BG/P

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



ystems f

Managing 120K CPUs

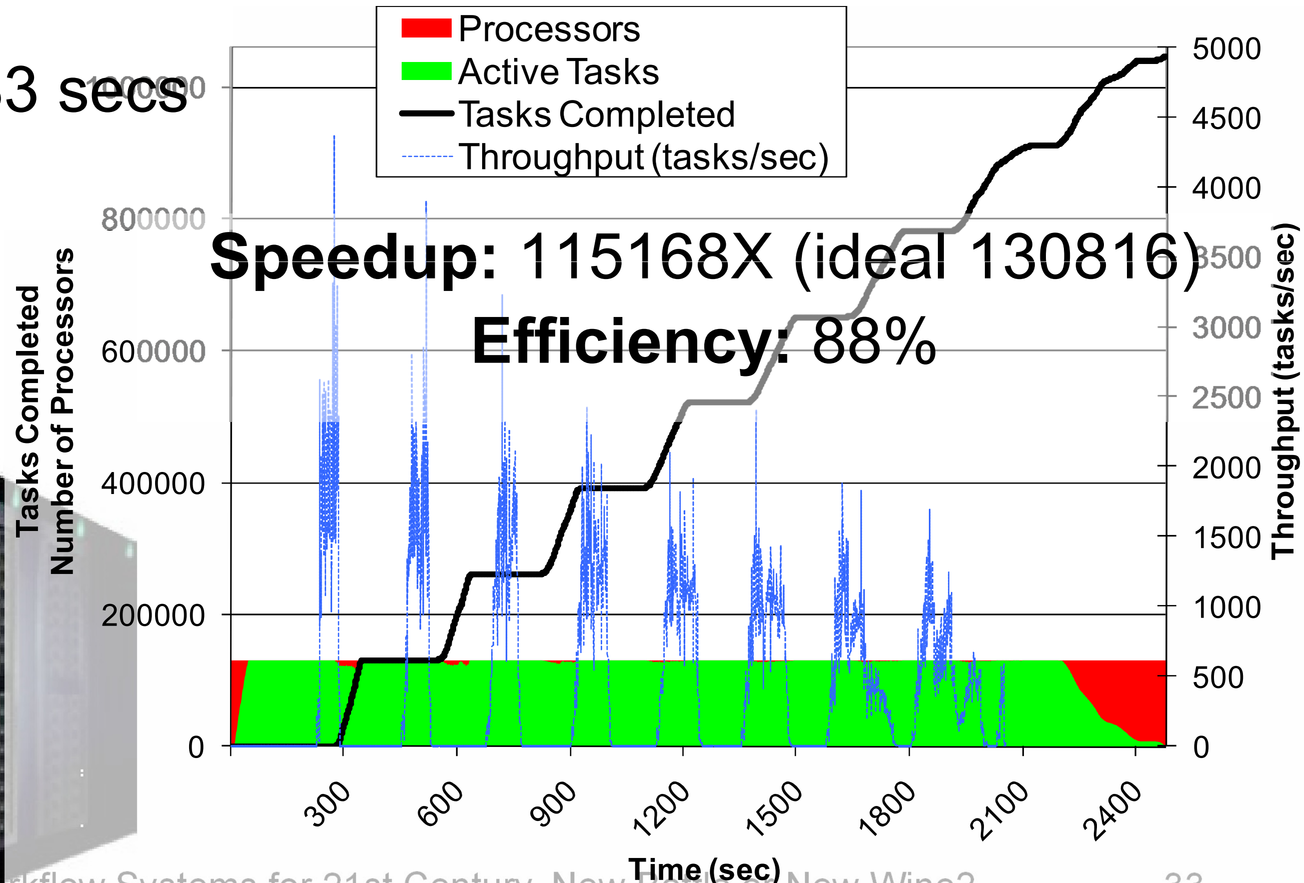


MARS Economic Modeling on IBM BG/P (128K CPUs)

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

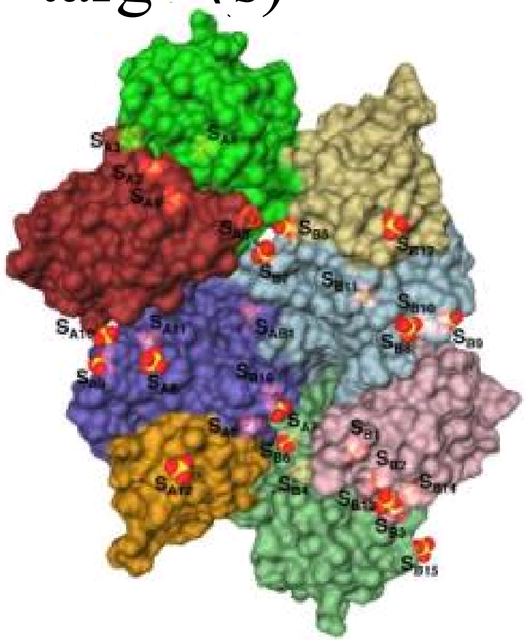
Speedup: 115168X (ideal 130816)

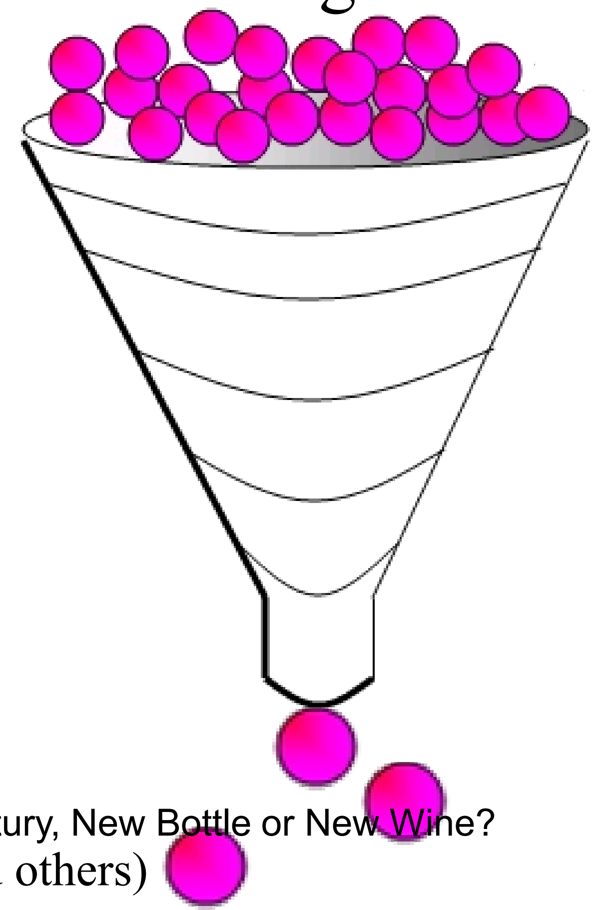
Efficiency: 88%



Many Many Tasks: Identifying Potential Drug Targets

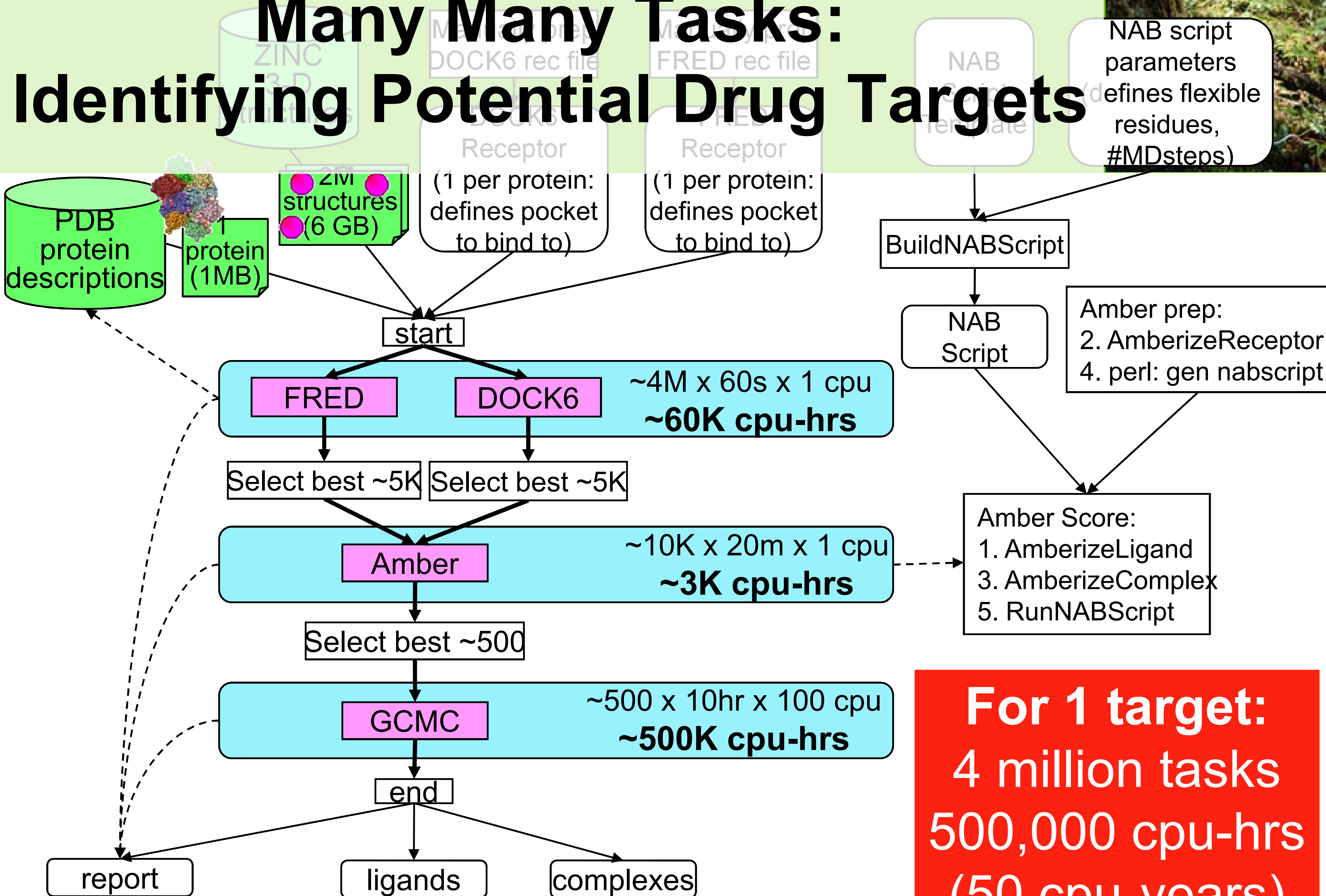
Protein target(s) x 2M+ ligands





(Mike Kubal, Benoit Roux, and others)

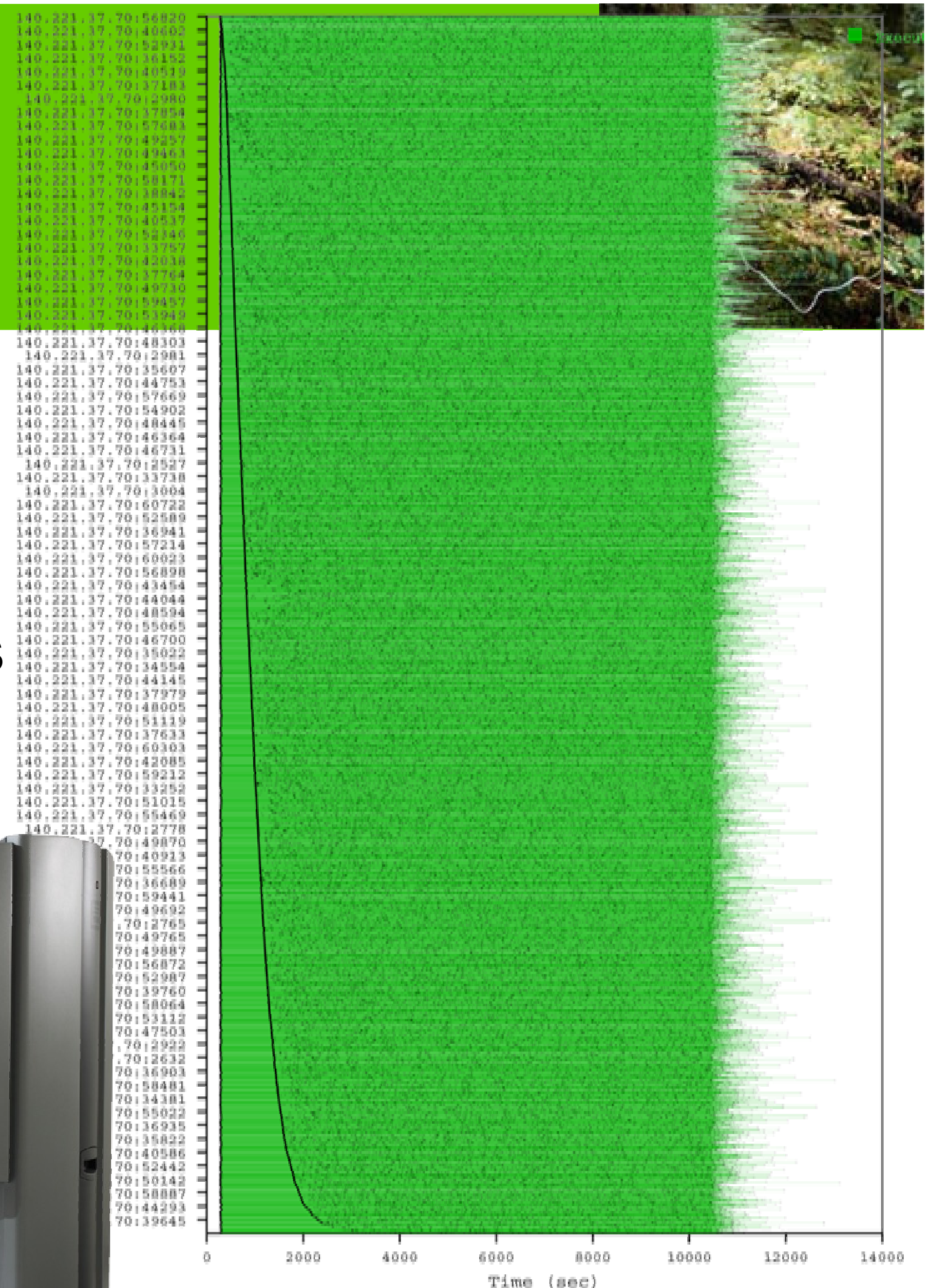
Many Many Tasks: Identifying Potential Drug Targets



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

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

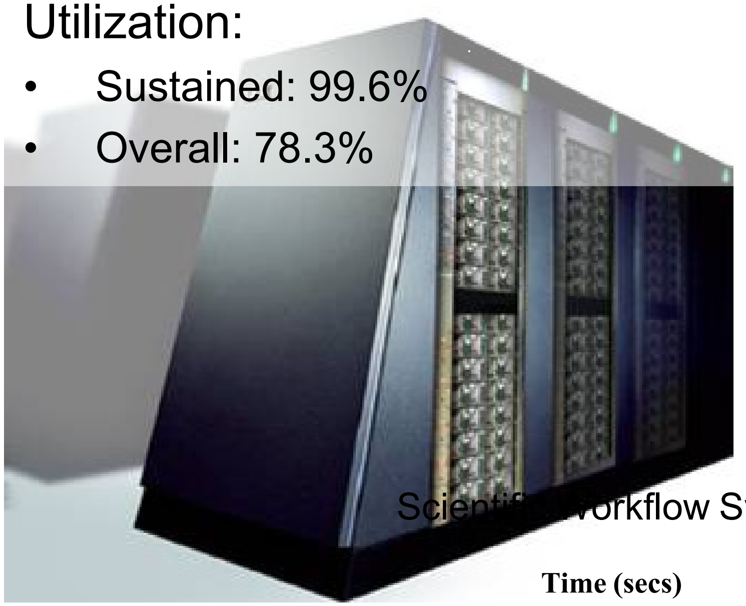
Average task time: 667 sec

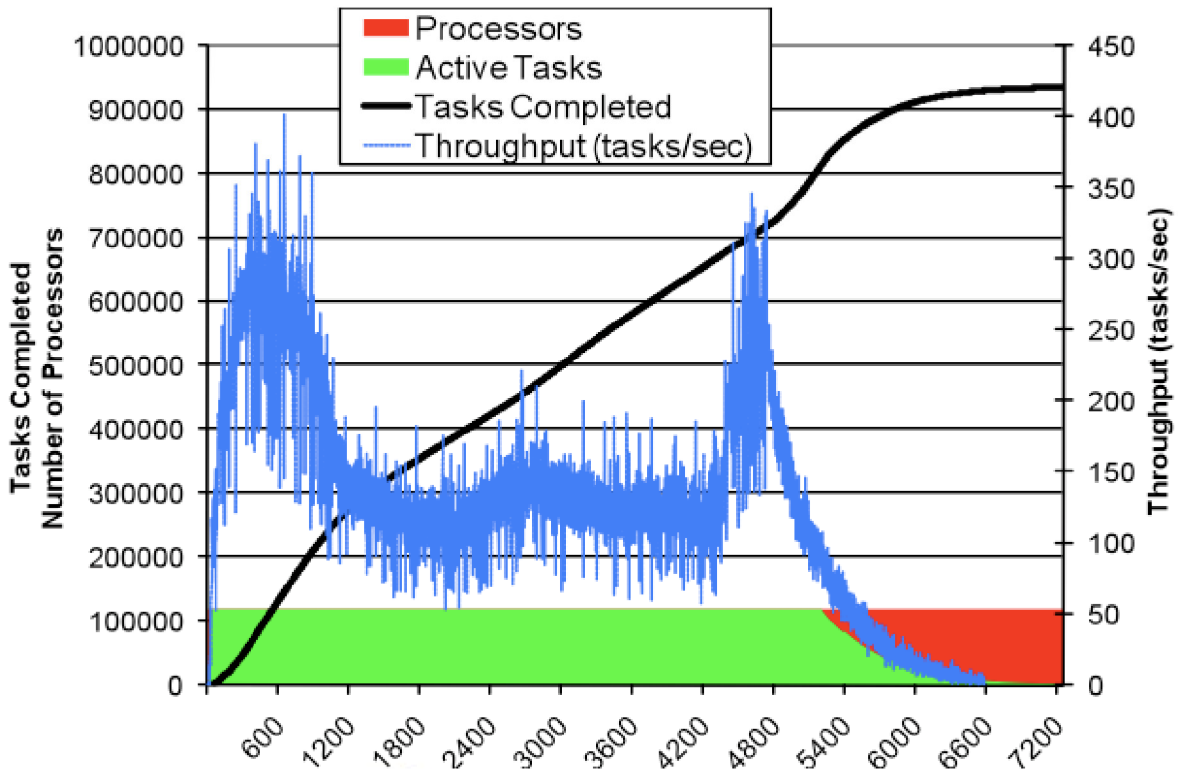
Relative Efficiency: 99.7%

(from 16 to 32 racks)

Utilization:

- Sustained: 99.6%
- Overall: 78.3%

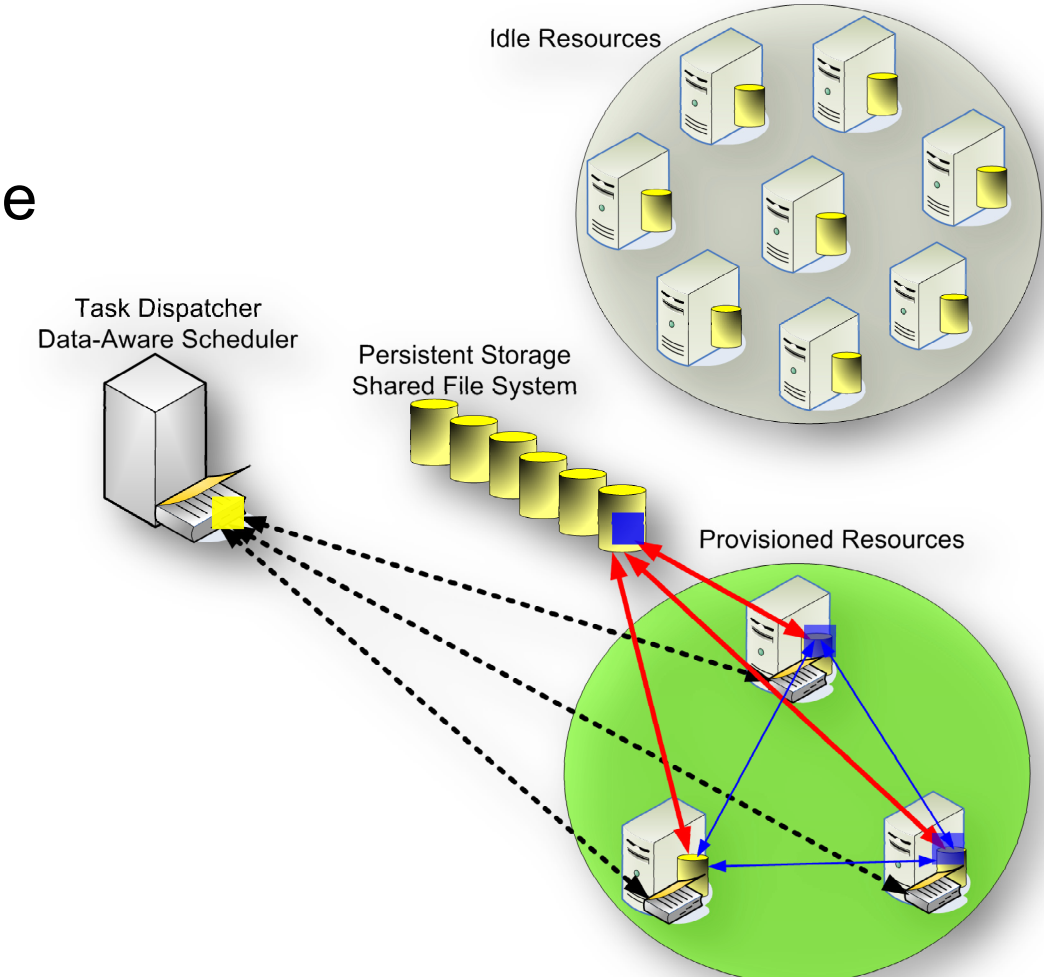




Time (secs)

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



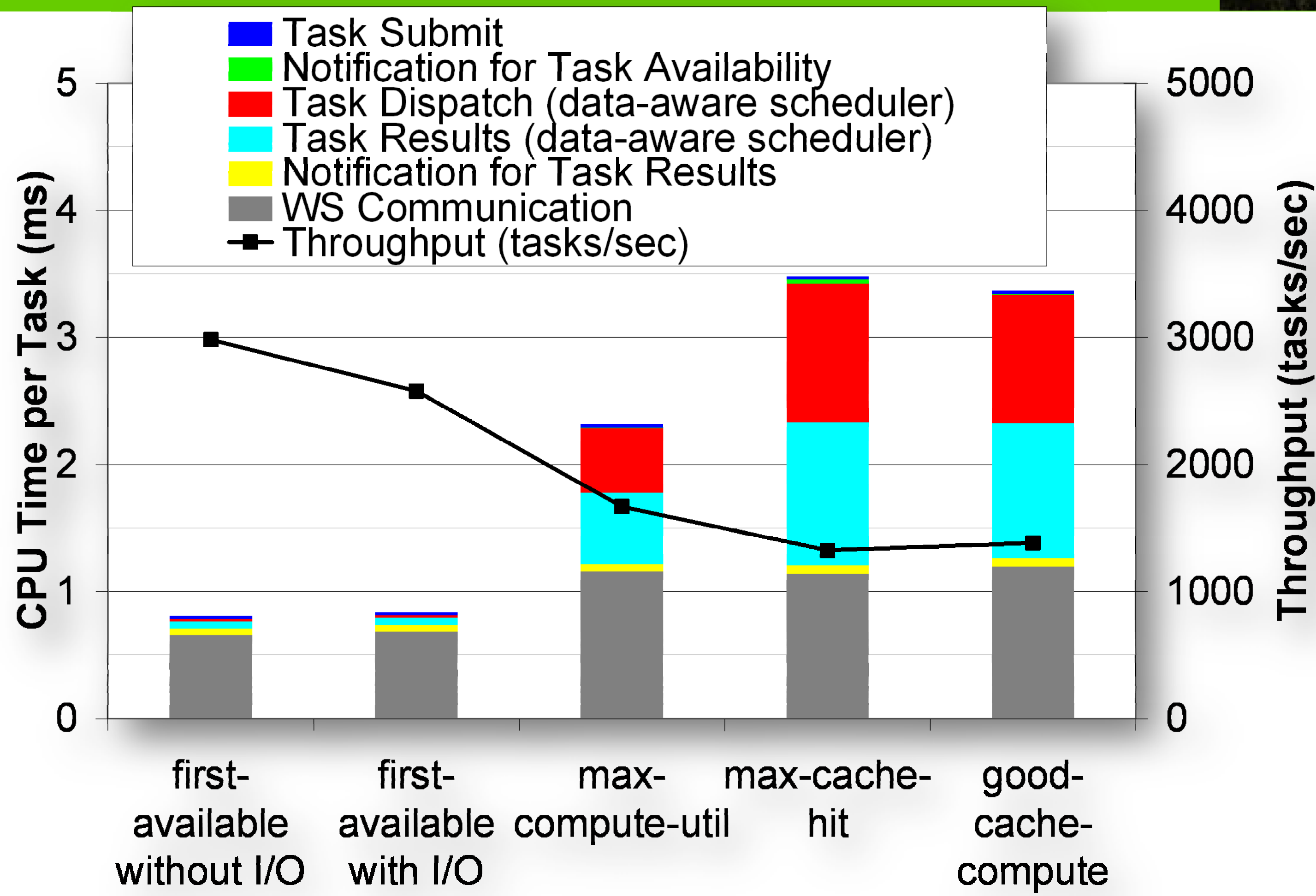
Data Diffusion

- 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

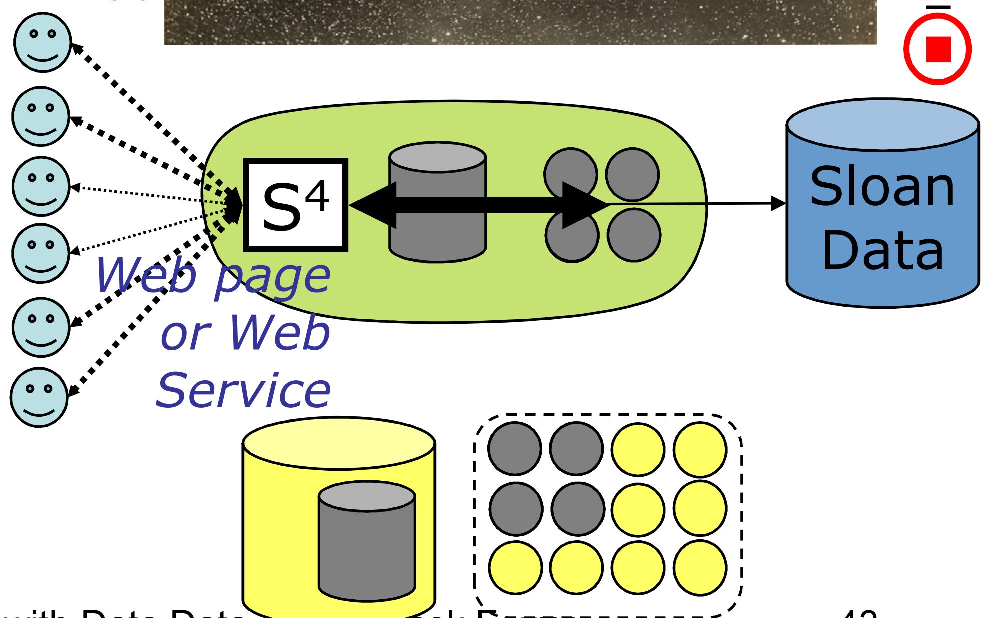
Data-Aware Scheduler Profiling



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

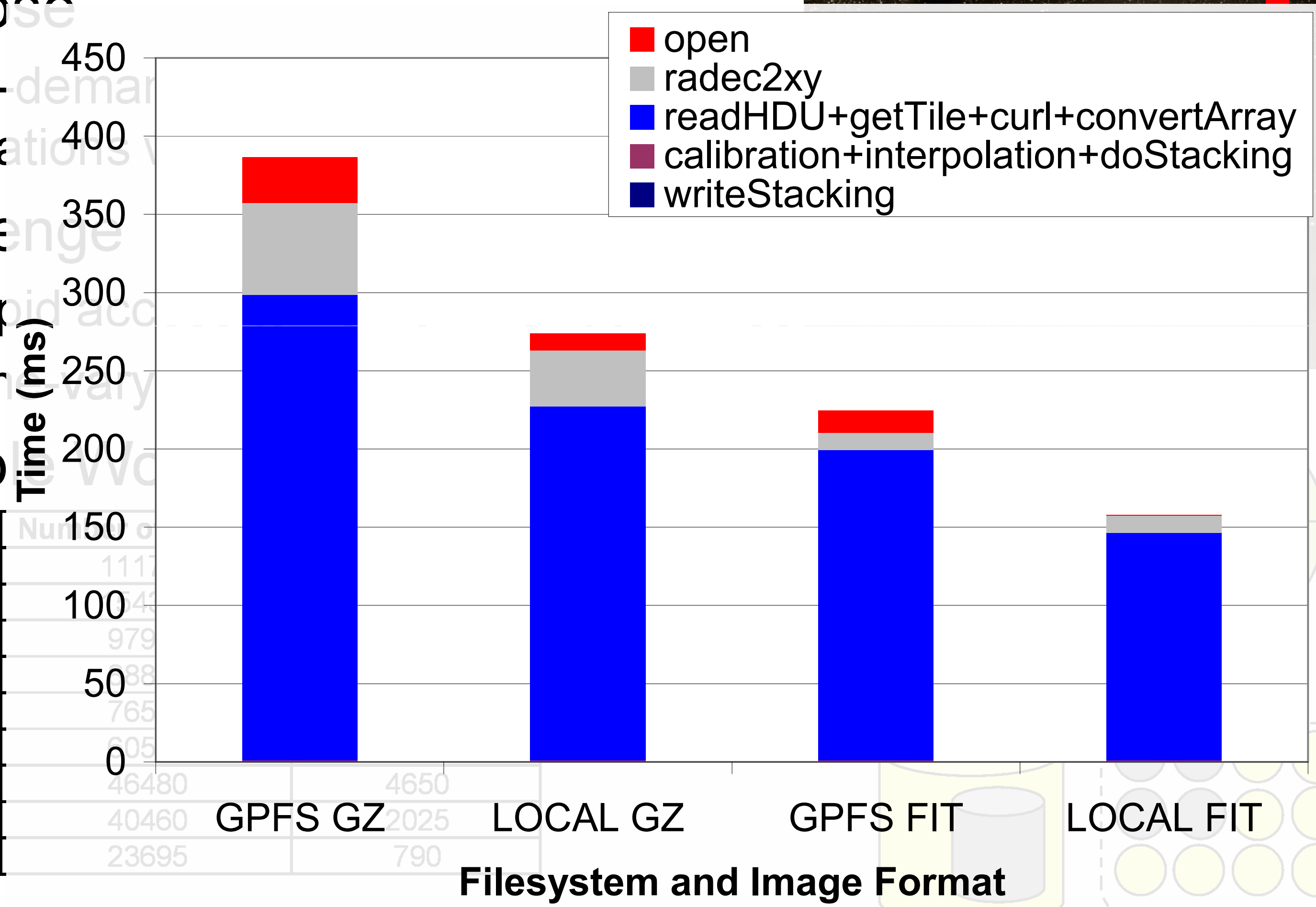
Locality	Number of Objects	Number of Files
1	111700	111700
1.38	154345	111699
2	97999	49000
3	88857	29620
4	76575	19145
5	60590	12120
10	46480	4650
20	40460	2025
30	23695	790



AstroPortal Stacking Service

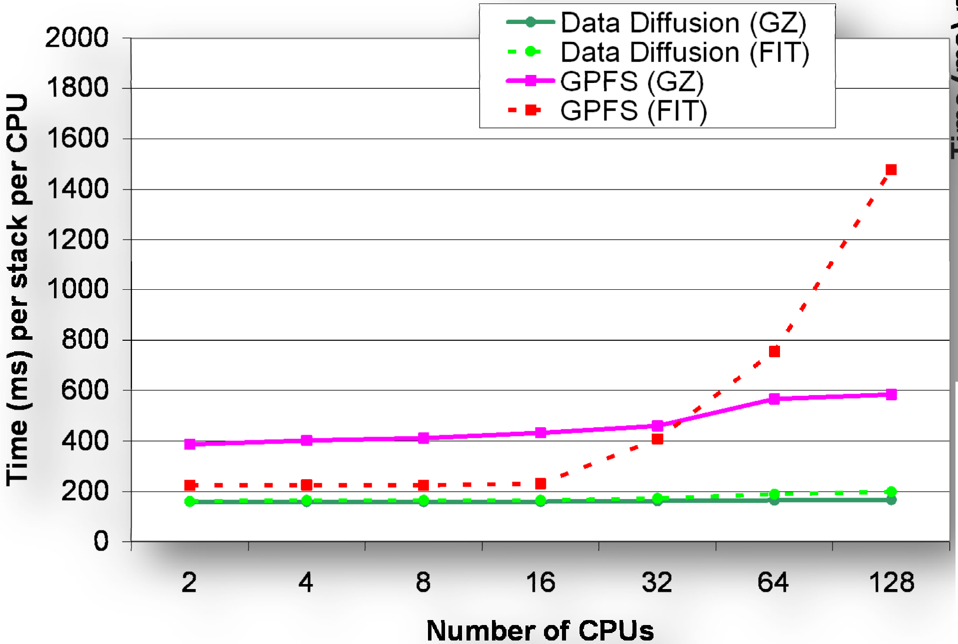
- Purpose
  - On-
    loc
- Challe
  - Ra
  - Ti
- Samp

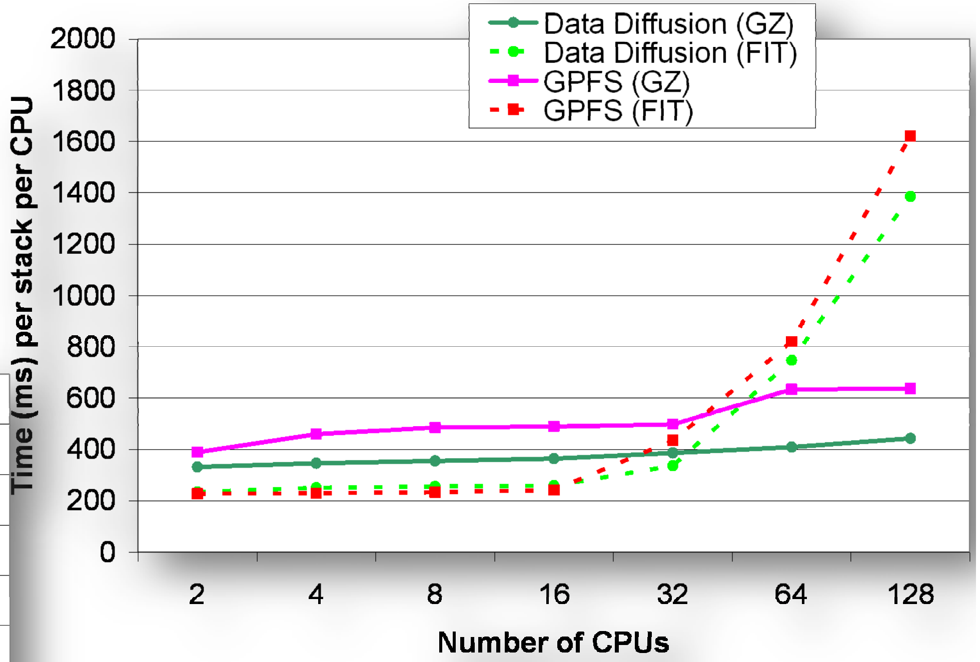
Locality	[illegible]	[illegible]
1	[illegible]	[illegible]
1.38	[illegible]	[illegible]
2	[illegible]	[illegible]
3	[illegible]	[illegible]
4	[illegible]	[illegible]
5	[illegible]	[illegible]
10	46480	4650
20	40460	2025
30	23695	790



AstroPortal Stacking Service with Data Diffusion

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

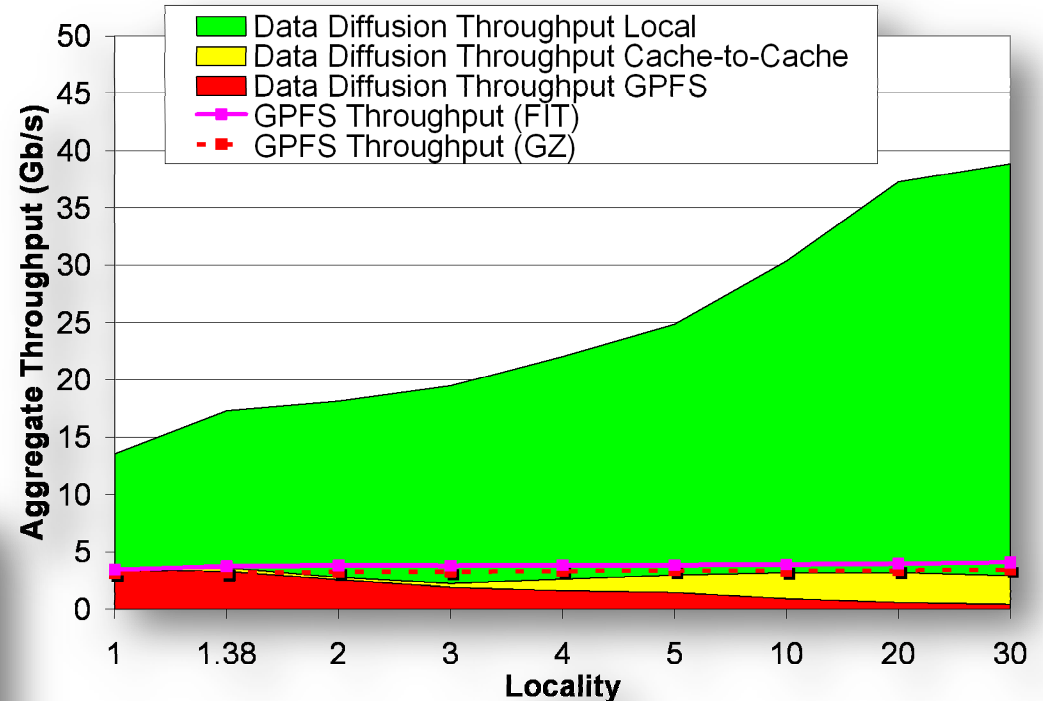


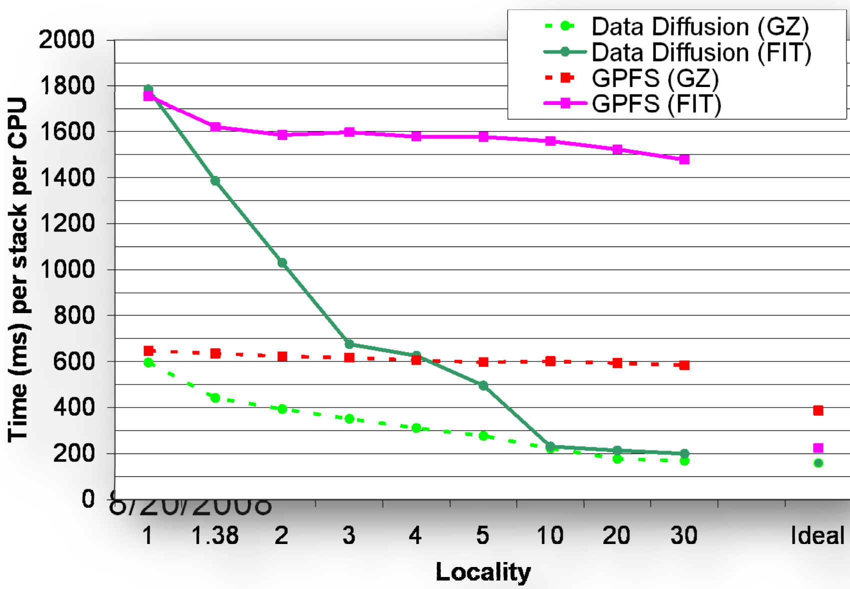


- ← High data locality
  - Near perfect scalability

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

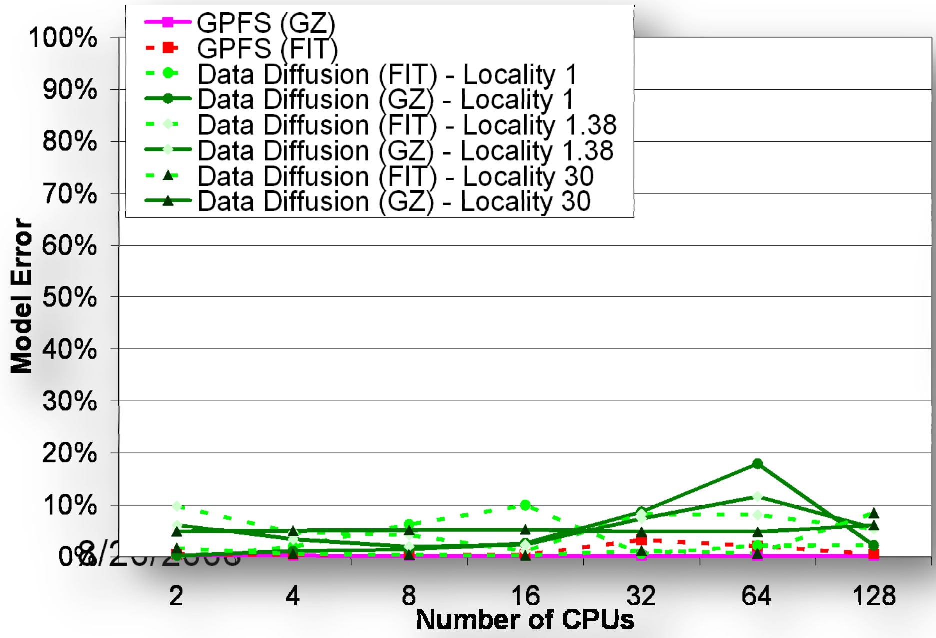


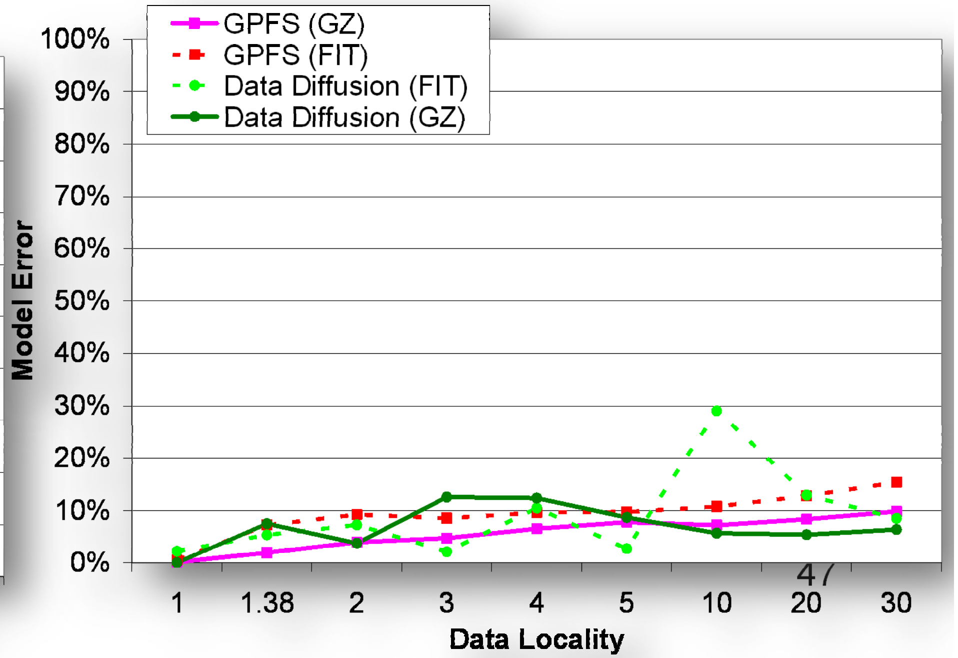


- Big performance gains as locality increases

AMDASK Model Validation

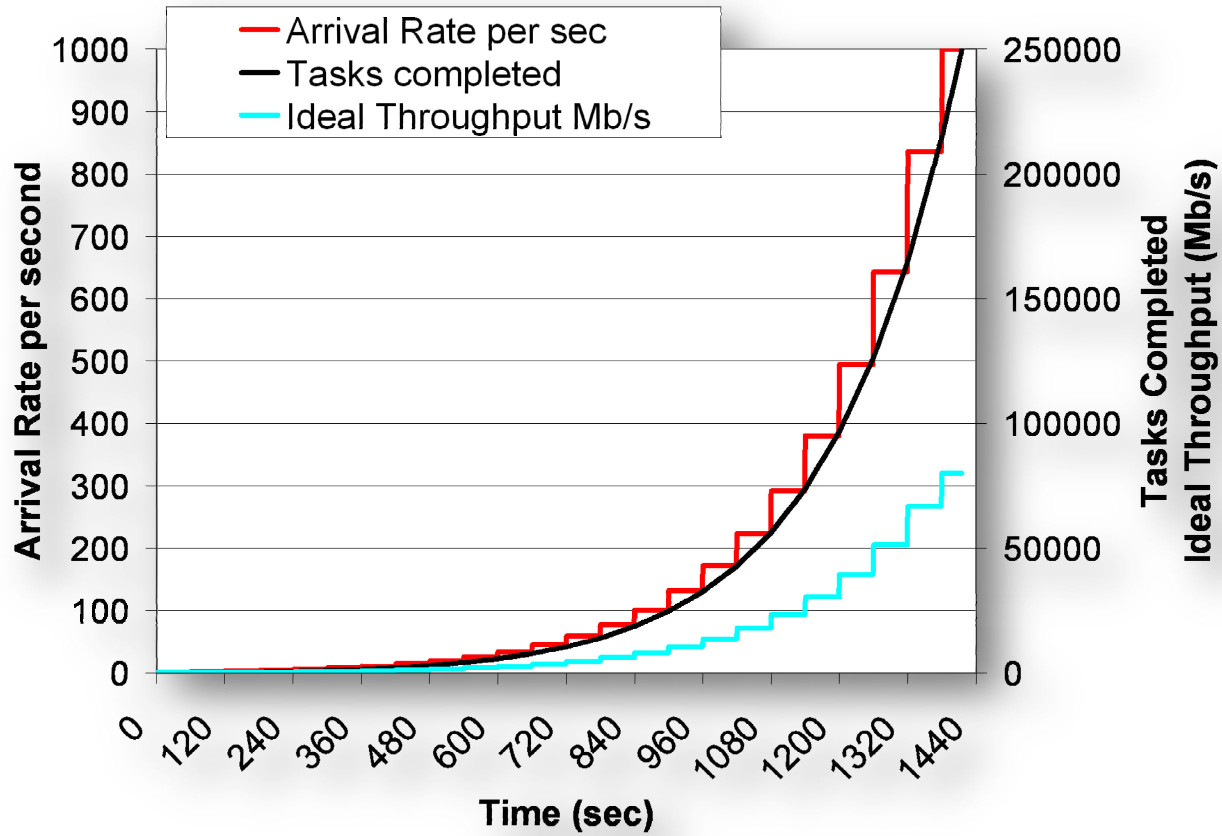
- Stacking service (large scale astronomy application)
- 92 experiments
- 558K files
  - Compressed: 2MB each ➔ 1.1TB
  - Un-compressed: 6MB each ➔ 3.3TB





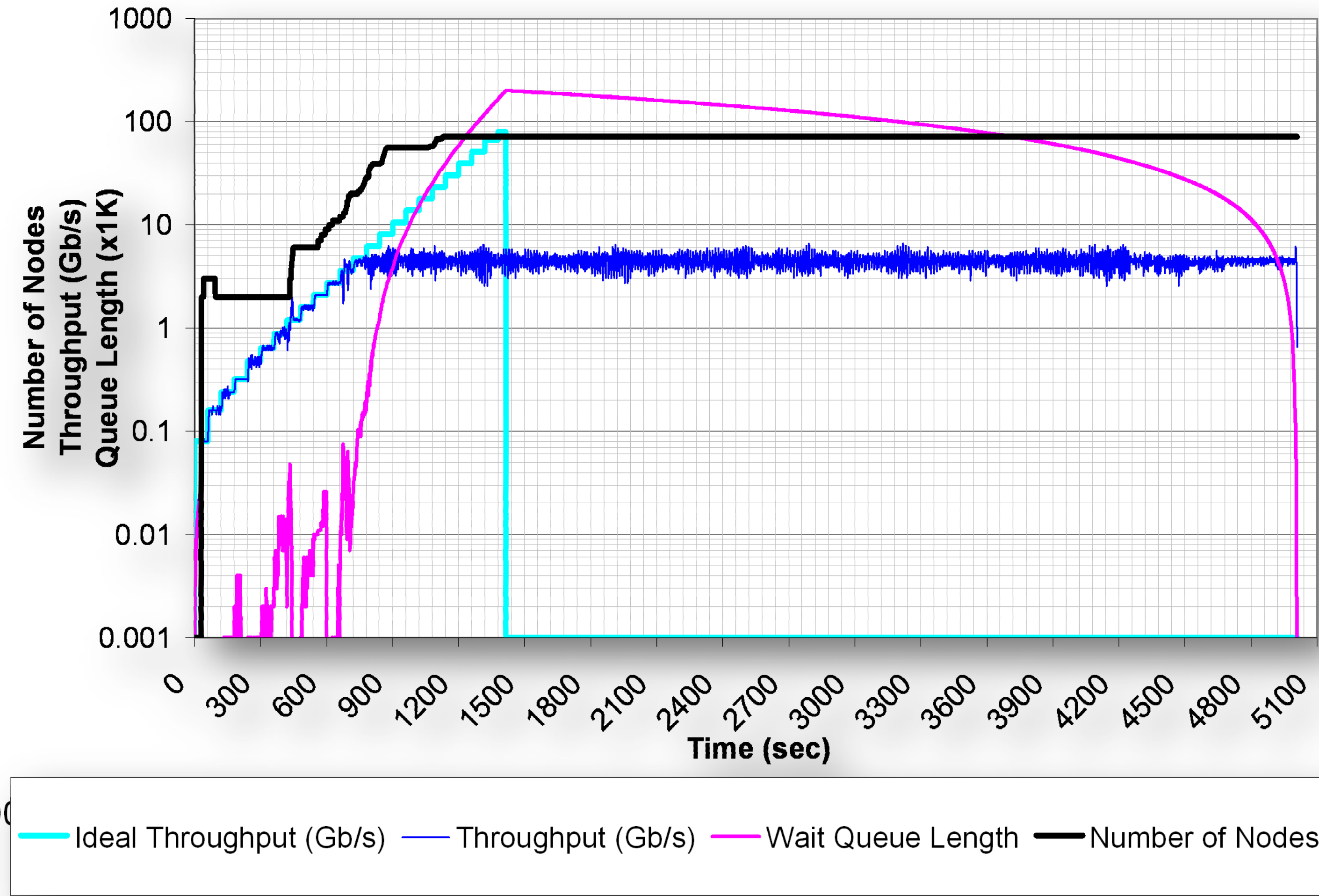
Data Diffusion:
Data-Intensive Workload

- 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



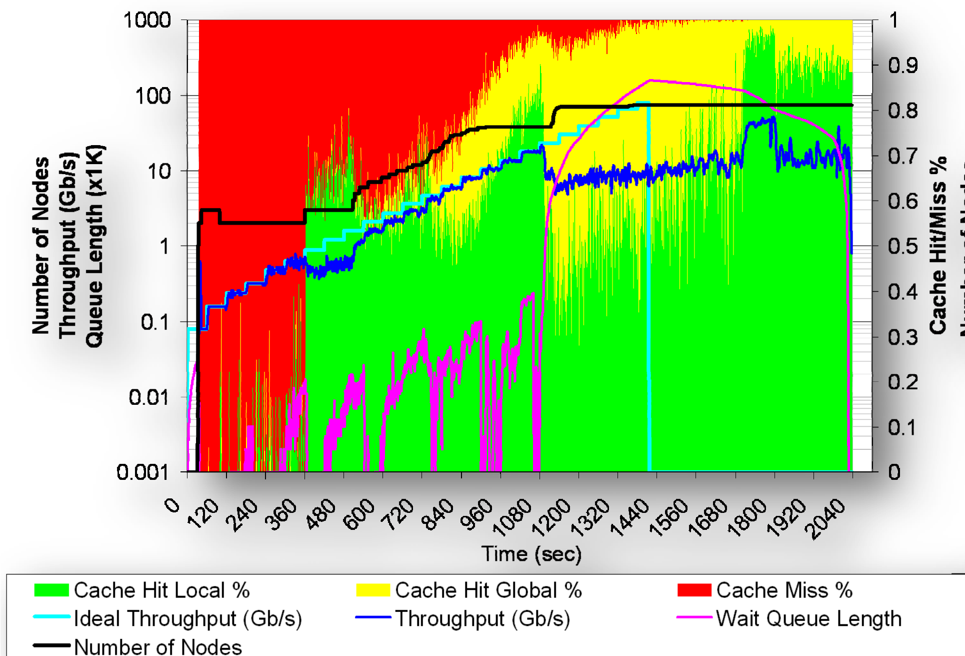
Data Diffusion: First-available (GPFS)

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

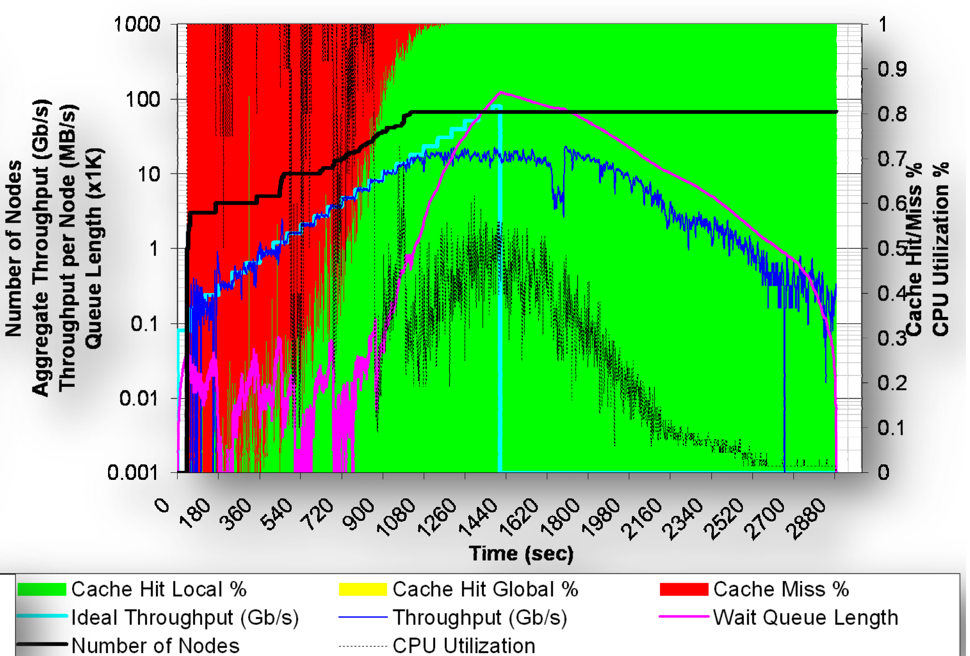


Data Diffusion: Max-compute-util & max-cache-hit

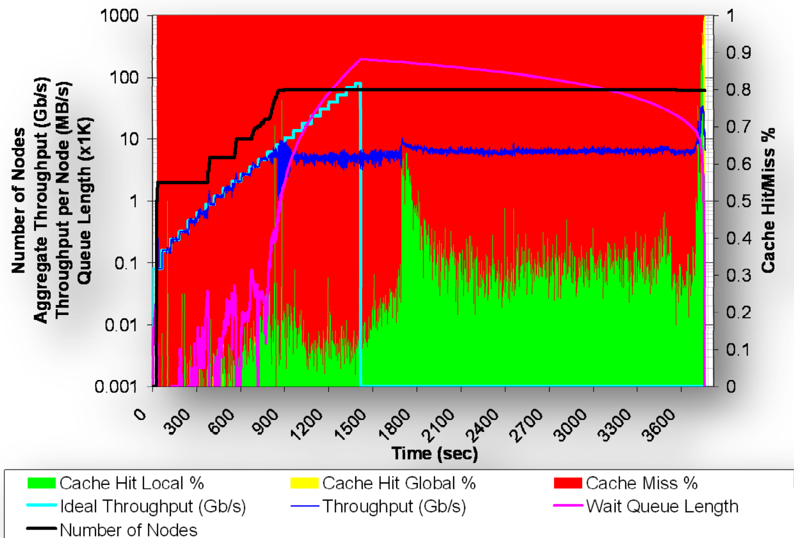
Max-compute-util



Max-cache-hit

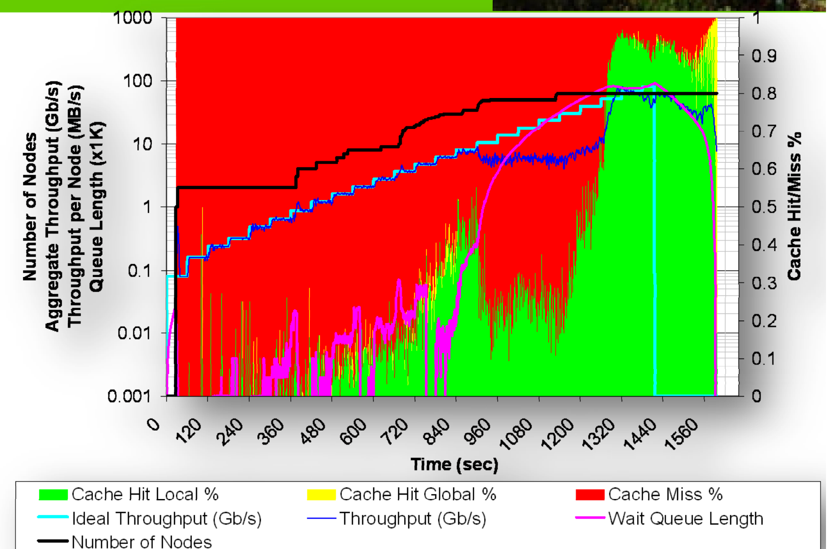


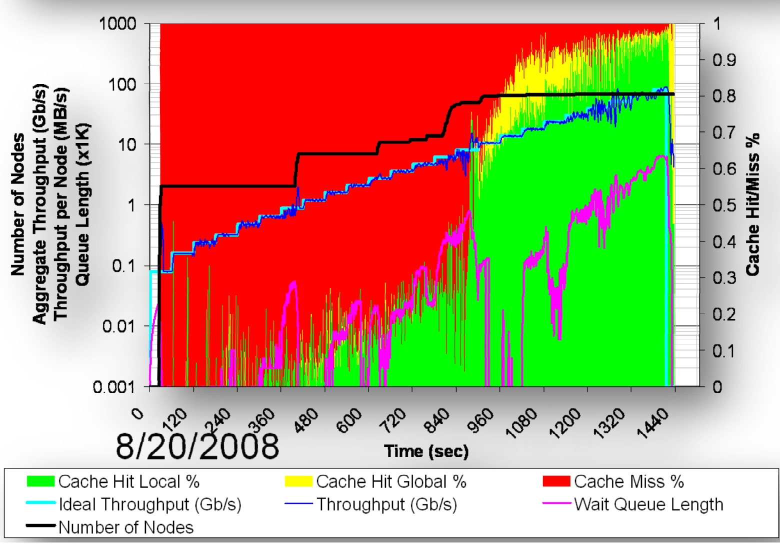
Data Diffusion: Good-cache-compute



← 1GB

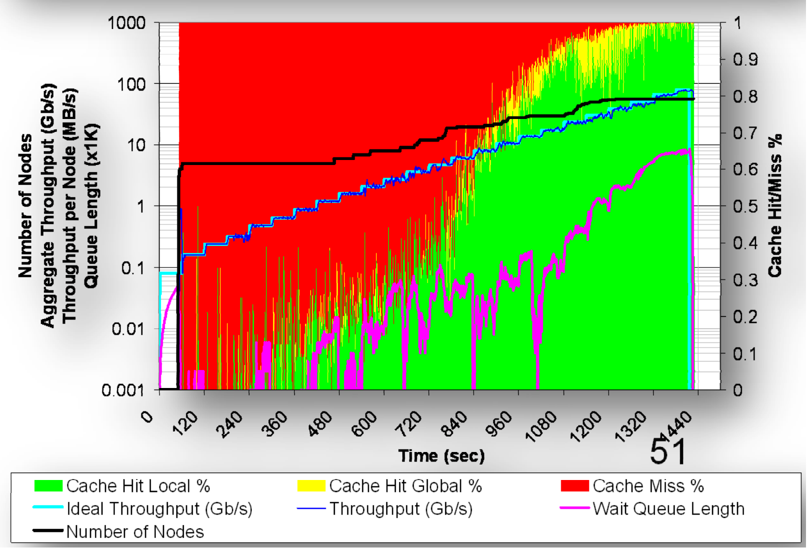
1.5GB →



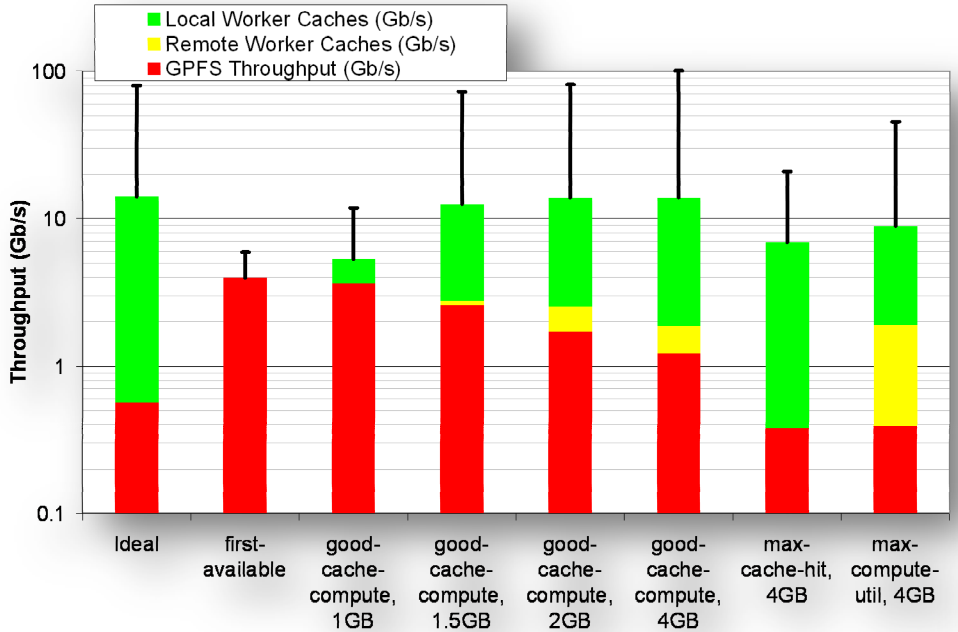


← 2GB

4GB →

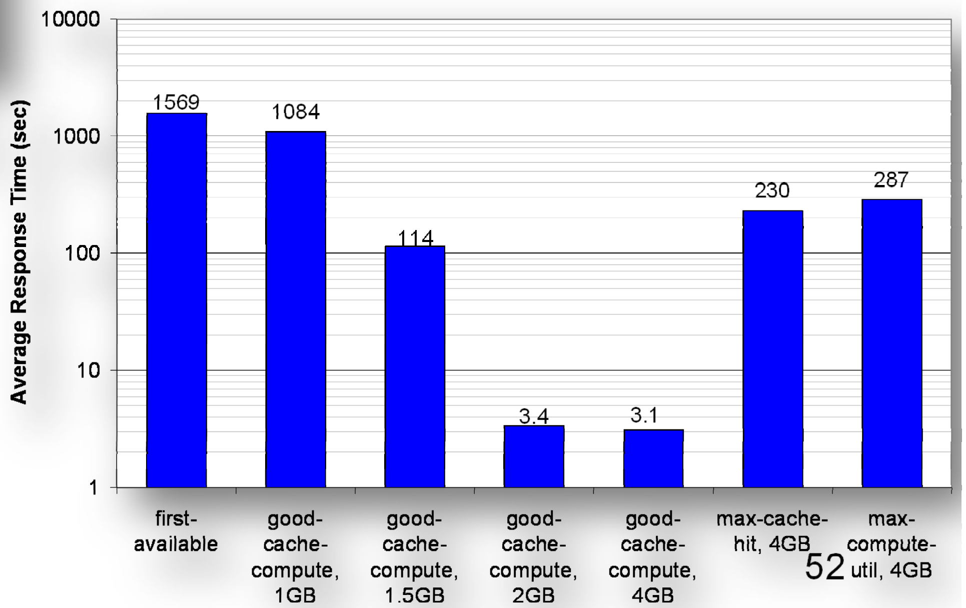


Data Diffusion: Throughput and Response Time



← Throughput:

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

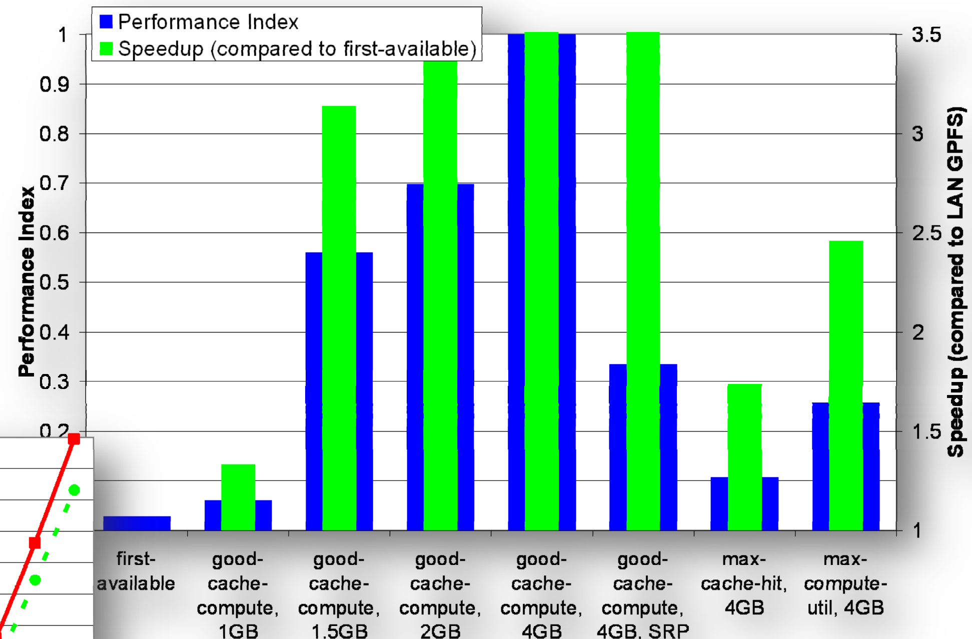


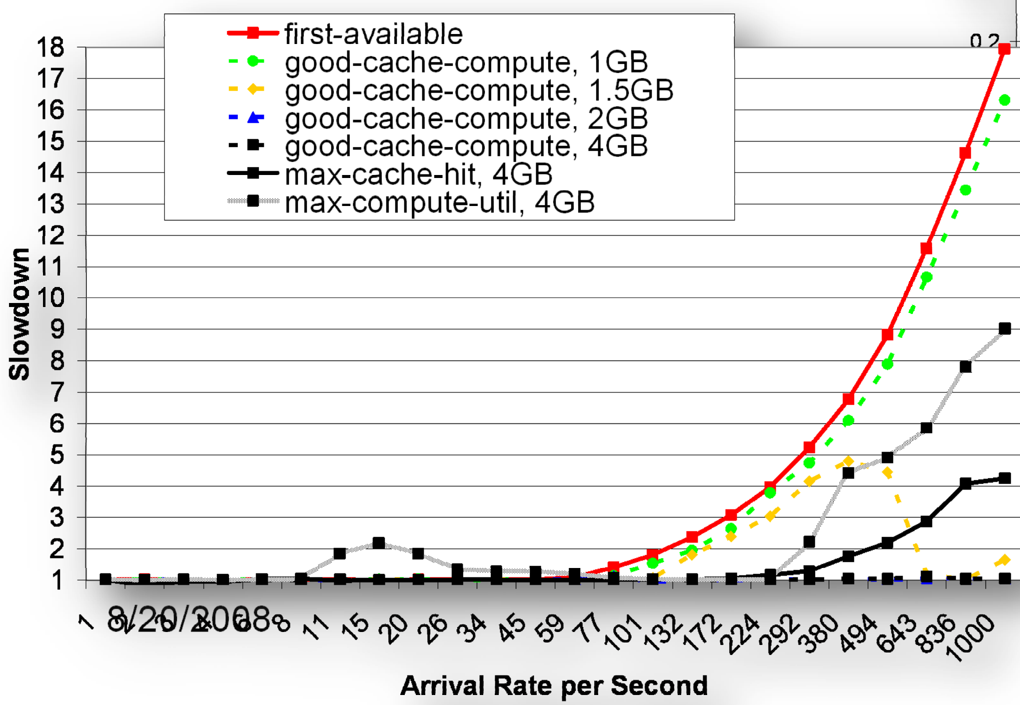
Response Time →

- 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

All-Pairs Synthetic Workload Falkon and Data Diffusion

500x500 ~ 250K tasks, 12MBx2 in, 8B out, 1 sec compute

	Number of CPUs	Time (sec)	Average Throughput (GB/s)	Average Throughput per CPU (MB/s)
Best Case Model	<500	~10800	~0.56	~1.11
Condor + Chirp (Active Storage)	<500	~21600	~0.28	~0.56
Condor + Shared File System (Demand Paging)	<500	~37800	~0.16	~0.32
Idea Case (with I/O to Local Disk)	128	1953	3.07	24.00
Falkon (Data Diffusion)	128	3056	1.96	15.34
Falkon (GPFS)	128	5438	1.10	8.62

500x500 ~ 250K tasks, 12MBx2 in, 8B out, 100ms compute

	Number of CPUs	Time (sec)	Average Throughput (GB/s)	Average Throughput per CPU (MB/s)
Idea Case (with I/O to Local Disk)	128	938	6.4	50.0
Falkon (Data Diffusion)	128	1033	5.8	45.4
Falkon (GPFS)	128	5416	1.1	8.7

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 on a real large-scale astronomy application

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

Relevant to Dissertation (2006 – Present)

- **Ian Foster**, The University of Chicago & Argonne National Laboratory
- **Alex Szalay**, The Johns Hopkins University
- **Rick Stevens**, The University of Chicago & Argonne National Laboratory
- **Yong Zhao**, Microsoft
- **Mike Wilde**, Computation Institute, University of Chicago & Argonne National Laboratory
- **Catalin Dumitrescu**, Fermi National Laboratory
- **Zhao Zhang**, The University of Chicago
- **Jerry C. Yan**, NASA, Ames Research Center
- **Kamil Iskra**, Argonne National Laboratory
- **Pete Beckman**, Argonne National Laboratory
- Mihael Hategan, The University of Chicago
- Ben Clifford, The University of Chicago
- Shiyong Lu, Wayne State University
- Veronika Nefedova, Argonne National Laboratory
- Tiberiu Stef-Praun, The University of Chicago
- Gabriela Turcu, The University of Chicago
- Atilla S. Balkir, The University of Chicago
- Jing Tie, The University of Chicago
- Quan T. Pham, The University of Chicago
- Sarah Kenny, The University of Chicago
- Dick Repasky, Indiana University
- Gregor von Laszewski, Rochester Institute of Technology
- Jim Gray, Microsoft Research
- Ruth Pordes, Fermilab National Accelerator Laboratory
- John McGee, Renaissance Computing Institute
- Julian Bunn, California Institute of Technology
- Marlon Pierce, Indiana University

All Collaborators
(2000 – Present)

- Ian Foster
- Catalin Dumitrescu
- Yong Zhao
- Mike Wilde
- Alex Szalay
- Sherali Zeadally
- Matei Ripeanu
- Mihael Hategan
- Zhao Zhang
- Alexandru Iosup
- Ben Clifford
- Dick Epema
- Jerry Yan
- Loren Schwiebert
- John Bresnahan
- Kamil Iskra
- Nicolae Tapus
- Owen Richter
- Rick Stevens
- Sandeep Gupta
- Scott Fowler
- Ahmad Naveed
- Anne-Marie Bosneag
- Atilla Balkir
- Carl Kesselman
- Davis Ford
- Dick Repasky
- Douglas Comer
- Gabriela Turcu
- Gohar Margaryan
- Gregor von Laszewski
- H Mohamed
- Jan Dünnweber
- Jennifer M Schoph
- Jim Gray
- Jing Tie
- John McGee
- Julian Bunn
- Laura Pearlman
- Liqiang Zhang
- Marlon Pierce
- Michael Link
- Mike D'Arcy
- Mugurel Andreica
- Neill Miller
- Pete Beckman
- Pham Quan
- Philipp Lüdeking
- R Wasseem
- Rajkumar Kettimuthu
- Ruth Pordes
- Sarah Kenny
- Sergei Gorlatch
- Shiyong Lu
- Tiberiu Stef-Praun
- Veronika Nefedova
- William Allcock

Publications/Proposals
Central to Dissertation (2005 – Present)

- **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**.
- **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**.
- 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.
- 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**.
- **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.
- 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.
- **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.
- **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.
- **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.
- 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**.
- **Ioan Raicu**. "Harnessing Grid Resources with Data-Centric Task Farms", **University of Chicago, Computer Science Department,** PhD Proposal, December 2007, Chicago, Illinois.
- **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.
- **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.
- **Ioan Raicu**, Yong Zhao, Ian Foster, Alex Szalay. "*A Data Diffusion Approach to Large Scale Scientific Exploration*", **Microsoft Research eScience Workshop 2007**.
- **Ioan Raicu**, Yong Zhao, Catalin Dumitrescu, Ian Foster, Mike Wilde. "*Falkon: a Fast and Light-weight tasK executiON framework*", **IEEE/ACM SuperComputing 2007**.
- **Ioan Raicu**, Catalin Dumitrescu, Ian Foster. "*Dynamic Resource Provisioning in Grid Environments*", **TeraGrid Conference 2007**.
- 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**.
- **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.
- **Ioan Raicu**, Ian Foster, Alex Szalay. "*Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets*", poster presentation, **IEEE/ACM SuperComputing 2006**.
- **Ioan Raicu**, Ian Foster, Alex Szalay, Gabriela Turcu. "*AstroPortal: A Science Gateway for Large-scale Astronomy Data Analysis*", **TeraGrid Conference 2006**, June 2006.
- 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

- 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.
- 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.
- 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.
- Catalin Dumitrescu, Ioan Raicu, Ian Foster. "*Usage SLA-based Scheduling in Grids*", Journal on Concurrency and Computation: Practice and Experience, 2006.
- 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.
- 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
- 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.
- 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.
- 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.
- 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.
- 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.
- 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
- 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.
- Sherali Zeadally, Ioan Raicu. "*Evaluating IPV6 on Windows and Solaris*", IEEE Internet Computing, Volume 7, Issue 3, May June 2003, pp 51 – 57.
- 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.
- 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.
- 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.
- Ioan Raicu. "*An Empirical Analysis of Internet Protocol version 6 (IPv6)*", Wayne State University, Computer Science Department, MS Thesis, May 2002, Detroit, Michigan.
- Ioan Raicu. "*Routing Algorithms for Wireless Sensor Networks*" Grace Hopper Celebration of Women in Computing 2002, GHC2002, 2002, British Columbia, Canada.
- 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)