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
+many more, see "Recent Collaborators" slide…

Motorola Labs
December 5th, 2008

Talk Overview

I. Introductions
   – University of Chicago, DSL
   – University of Chicago, CI
   – Argonne National Laboratory, MCS
II. Comparing Grids and Clouds
III. Scalable resource management challenges and solutions
   – Dispatch
   – Provisioning
   – Data Management

Talk Overview

I. Introductions
   - University of Chicago, DSL
   - University of Chicago, CI
   - Argonne National Laboratory, MCS

II. Comparing Grids and Clouds

III. Scalable resource management challenges and solutions
   - Dispatch
   - Provisioning
   - Data Management

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/

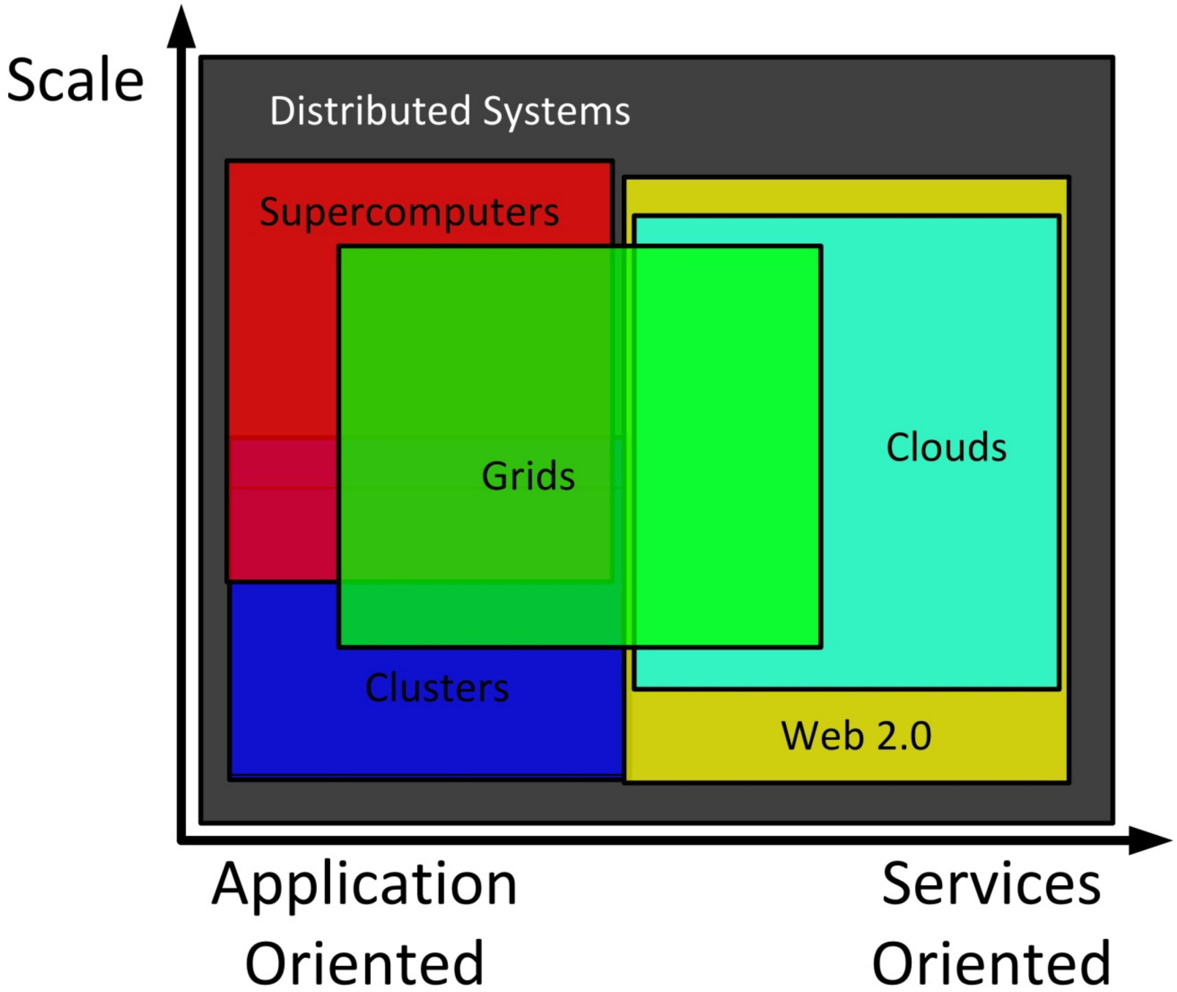
Talk Overview

I. Introductions
   - University of Chicago, DSL
   - University of Chicago, CI
   - Argonne National Laboratory, MCS

II. Comparing Grids and Clouds

III. Scalable resource management challenges and solutions
   - Dispatch
   - Provisioning
   - Data Management

Clusters, Grids, Clouds, …

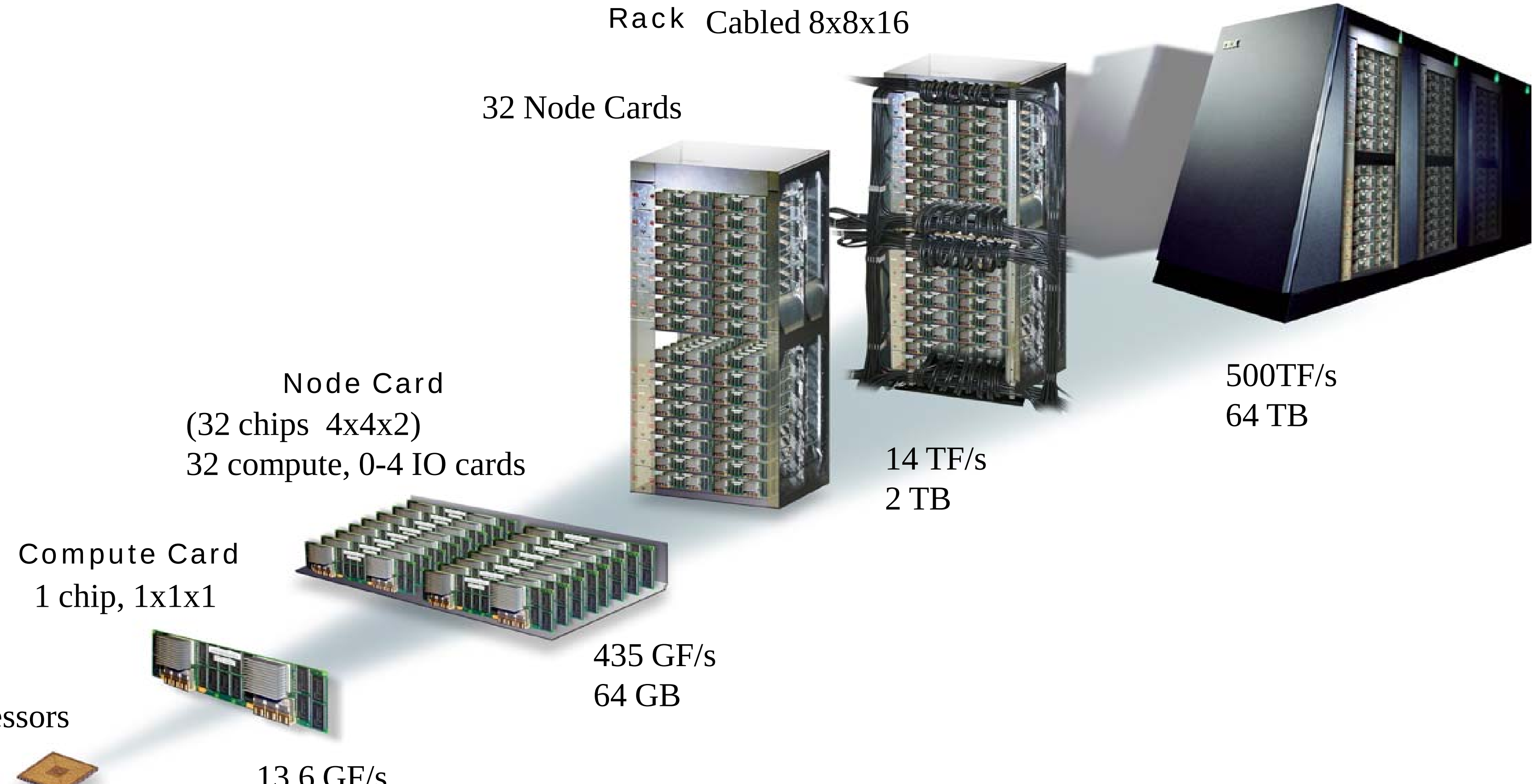


Supercomputing

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

e.g. IBM Blue Gene/P

IBM Blue Gene/P at ANL ALCF



Cluster Computing

Computer clusters using commodity processors, network interconnects, and operating system

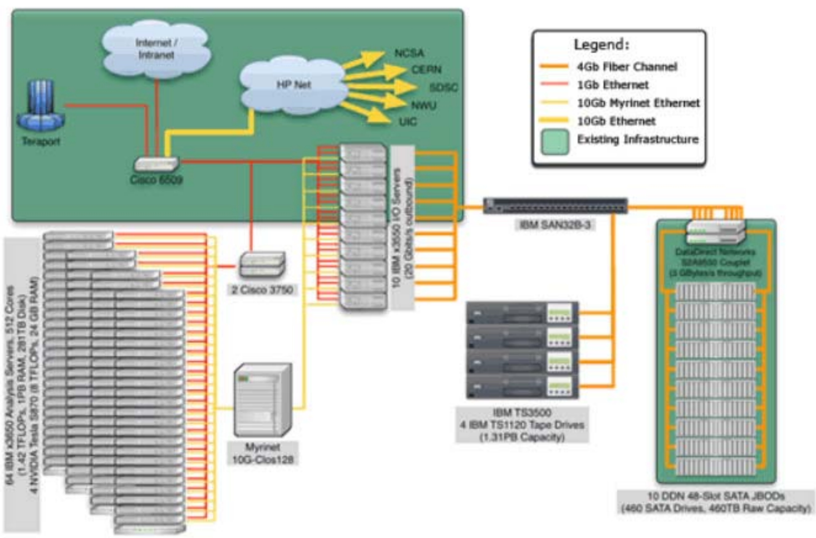
e.g. PADS

Petascale Active Data Store (PADS)

PADS

home

website in development



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

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

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

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

PADSstatus

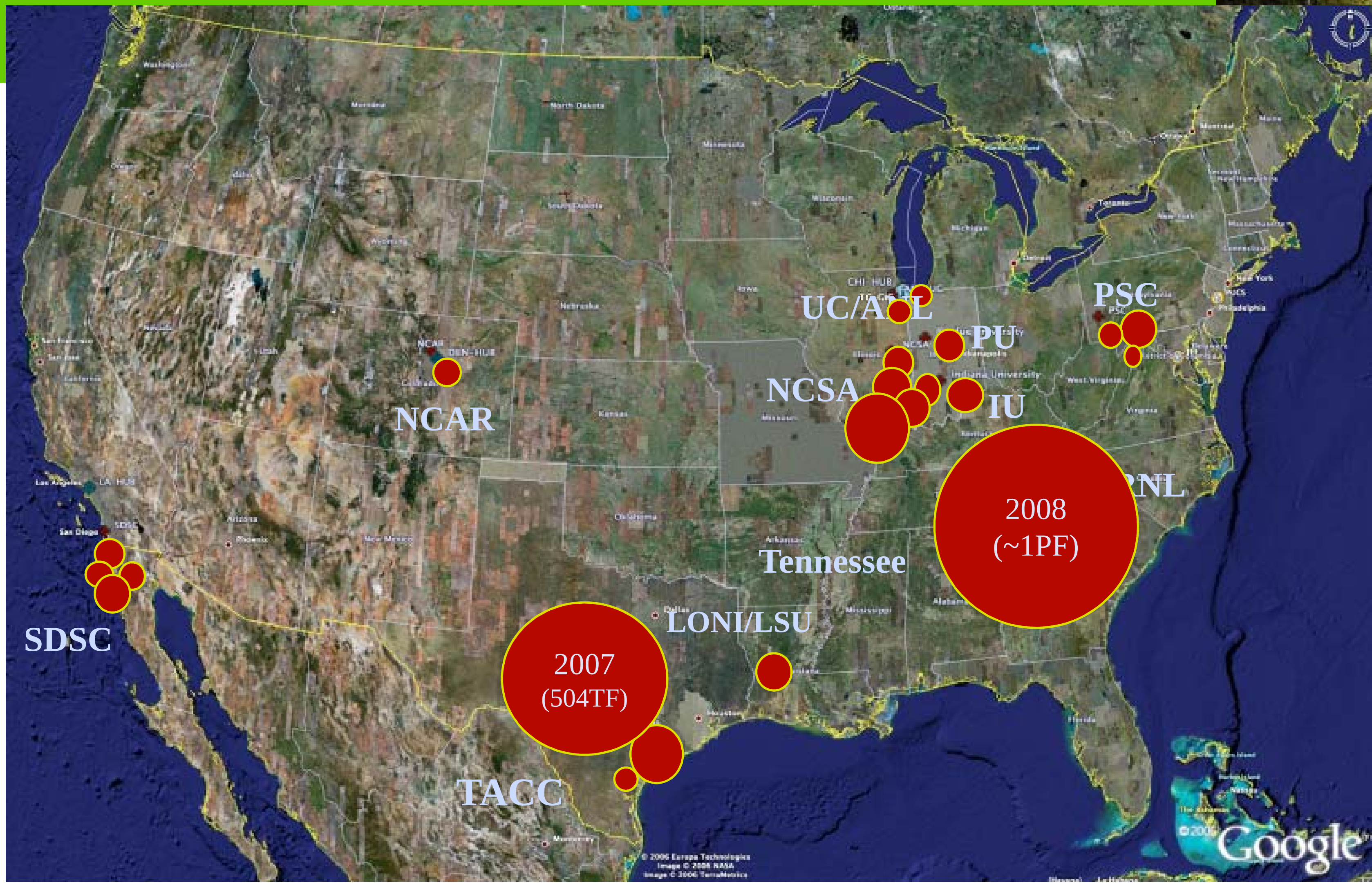
myPADS

Grid Computing

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

e.g. TeraGrid

TeraGrid High Performance Computing Systems 2007-8



Computational Resources
(size approximate - not to scale)

Tommy Minyard, TACC

What is the TeraGrid?

- An instrument (cyberinfrastructure) that delivers high-end IT resources - storage, computation, visualization, and data/service hosting - almost all of which are UNIX-based under the covers; some hidden by Web interfaces
  - 20 Petabytes of storage (disk and tape)
  - over 100 scientific data collections
  - 750 TFLOPS (161K-cores) in parallel computing systems and growing
  - Support for Science Gateways
- The largest individual cyberinfrastructure facility funded by the NSF, which supports the national science and engineering research community
- Something you can use without financial cost - allocated via peer review (and without double jeopardy)

Major Grids

- TeraGrid (TG)
- Open Science Grid (OSG)
- Enabling Grids for E-sciencE (EGEE)
- LHC Computing Grid from CERN
- Grid Middleware
  - Globus Toolkit
  - Unicore

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

Major Cloud Middleware

- Google App Engine
  – Engine, Datastore, memcache
- Amazon
  – EC2, S3, SQS, SimpleDB
- Microsoft Azure
- Nimbus
- Eucalyptus
- Salesforce

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

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

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

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

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

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

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

Outline

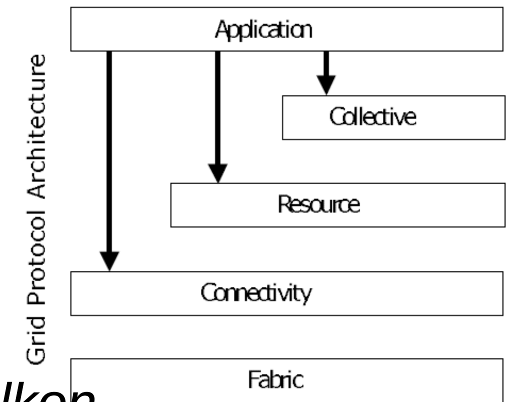
- Business model
- Architecture
- Resource management
- Programming model
- Application model
- Security model

Business Model

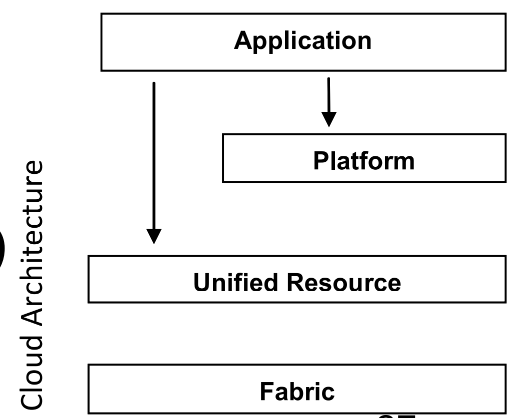
- Grids:
  - Largest Grids funded by government
  - Largest user-base in academia and government labs to drive scientific computing
  - Project-oriented: service units
- Clouds:
  - Industry (i.e. Amazon) funded the initial Clouds
  - Large user base in common people, small businesses, large businesses,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

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

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

Gateways

- Aimed to simplify usage of complex resources
- Grids
  - Front-ends to many different applications
  - Emerging technologies for Grids
- Clouds
  - Standard interface to Clouds

Gateway to Grids

Gateway to Clouds

Security Model

- Grids
  - Grid Security Infrastructure (GSI)
  - Stronger, but steeper learning curve and wait time
    - Personal verification: phone, manager, etc
- Clouds
  - Weaker, can use credit card to gain access, can reset password over plain text email, etc

Conclusion

- Move towards a mix of micro-production and large utilities, with load being distributed among them dynamically
  - Increasing numbers of small-scale producers (local clusters and embedded processors—in shoes and walls)
  - Large-scale regional producers
- Need to define protocols
  - Allow users and service providers to discover, monitor and manage their reservations and payments
  - Interoperability
- Need to combine the centralized scale of today's Cloud utilities, and the distribution and interoperability of today's Grid facilities
- Need support for on-demand provisioning
- Need tools for managing both the underlying resources and the resulting distributed computations
- Security and trust will be a major obstacle for commercial Clouds by large companies that have in-house IT resources to host their own data centers

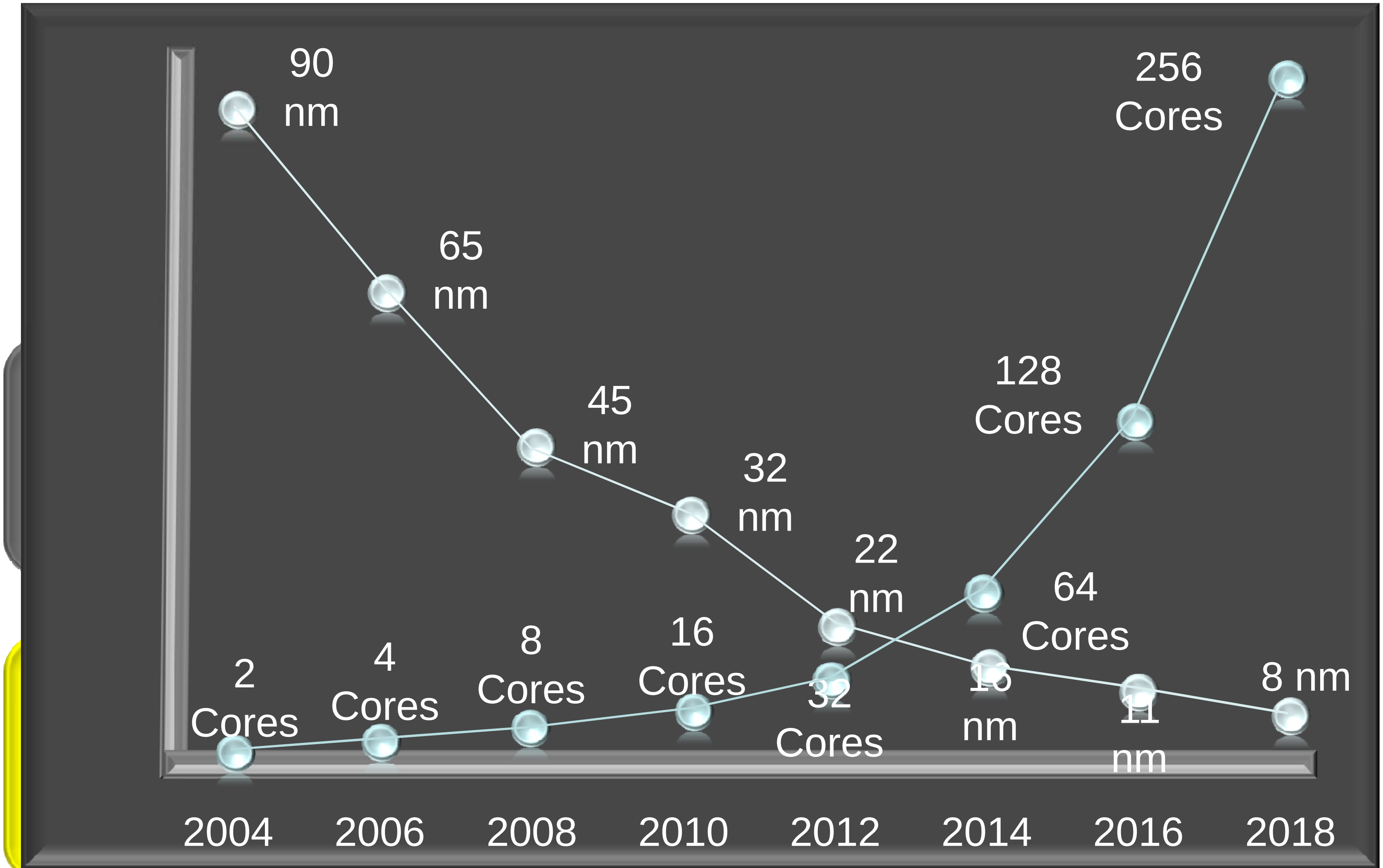
Talk Overview

I. Introductions
   - University of Chicago, DSL
   - University of Chicago, CI
   - Argonne National Laboratory, MCS

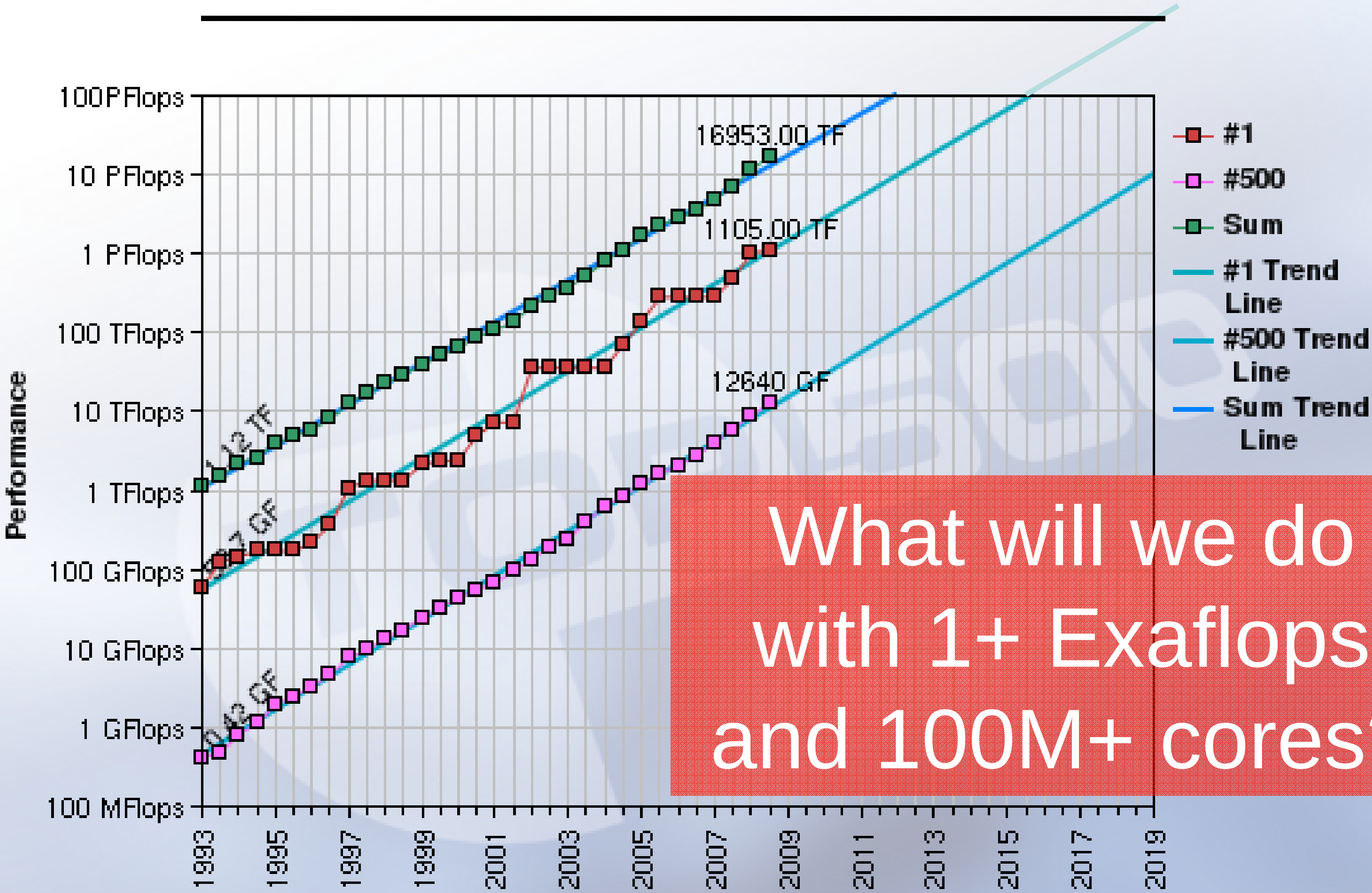
II. Comparing Grids and Clouds

III. Scalable resource management challenges and solutions
   - Dispatch
   - Provisioning
   - Data Management

Many-Core Growth Rates



Projected Performance Development



What will we do with 1+ Exaflops and 100M+ cores?

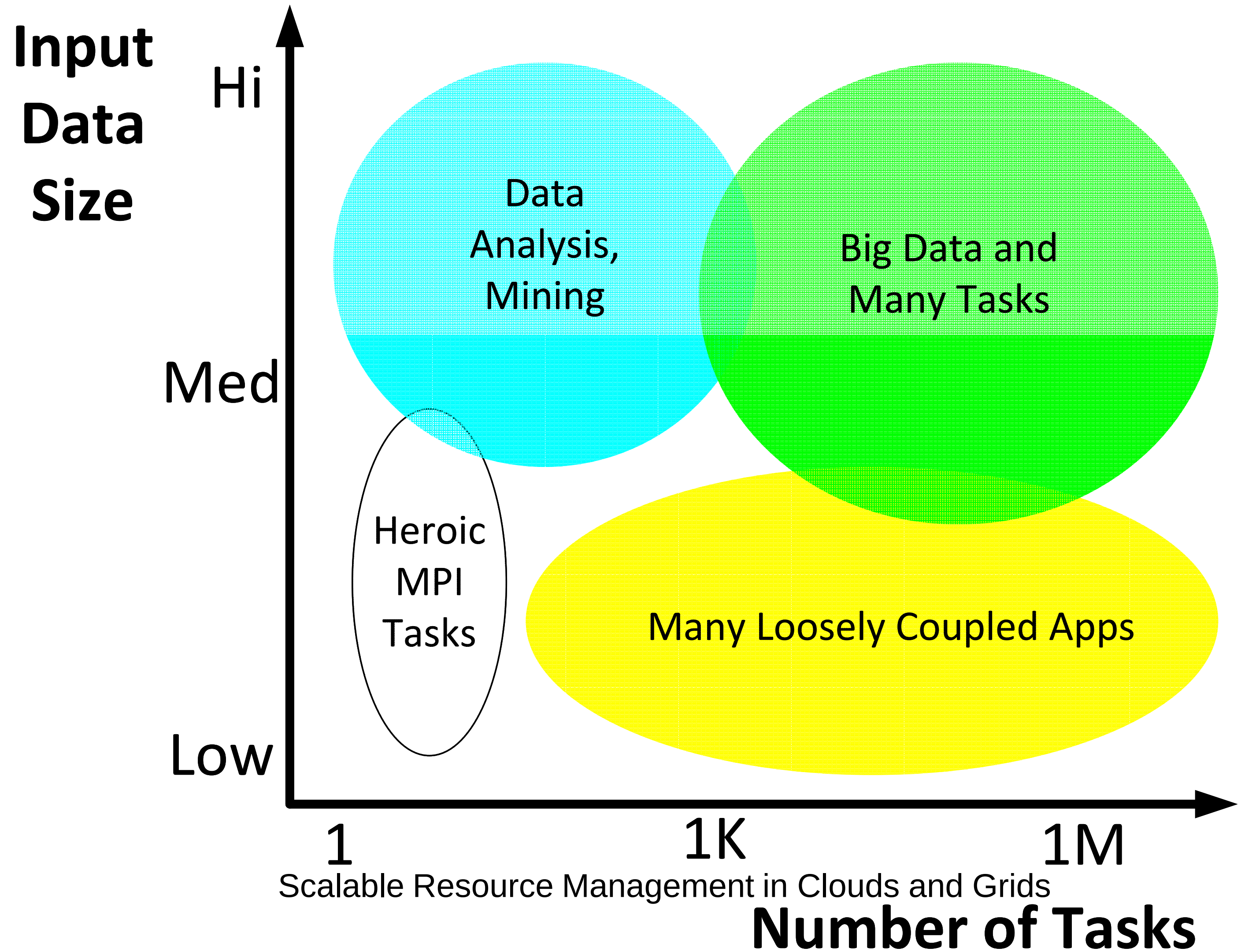
Programming Model Issues

- **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

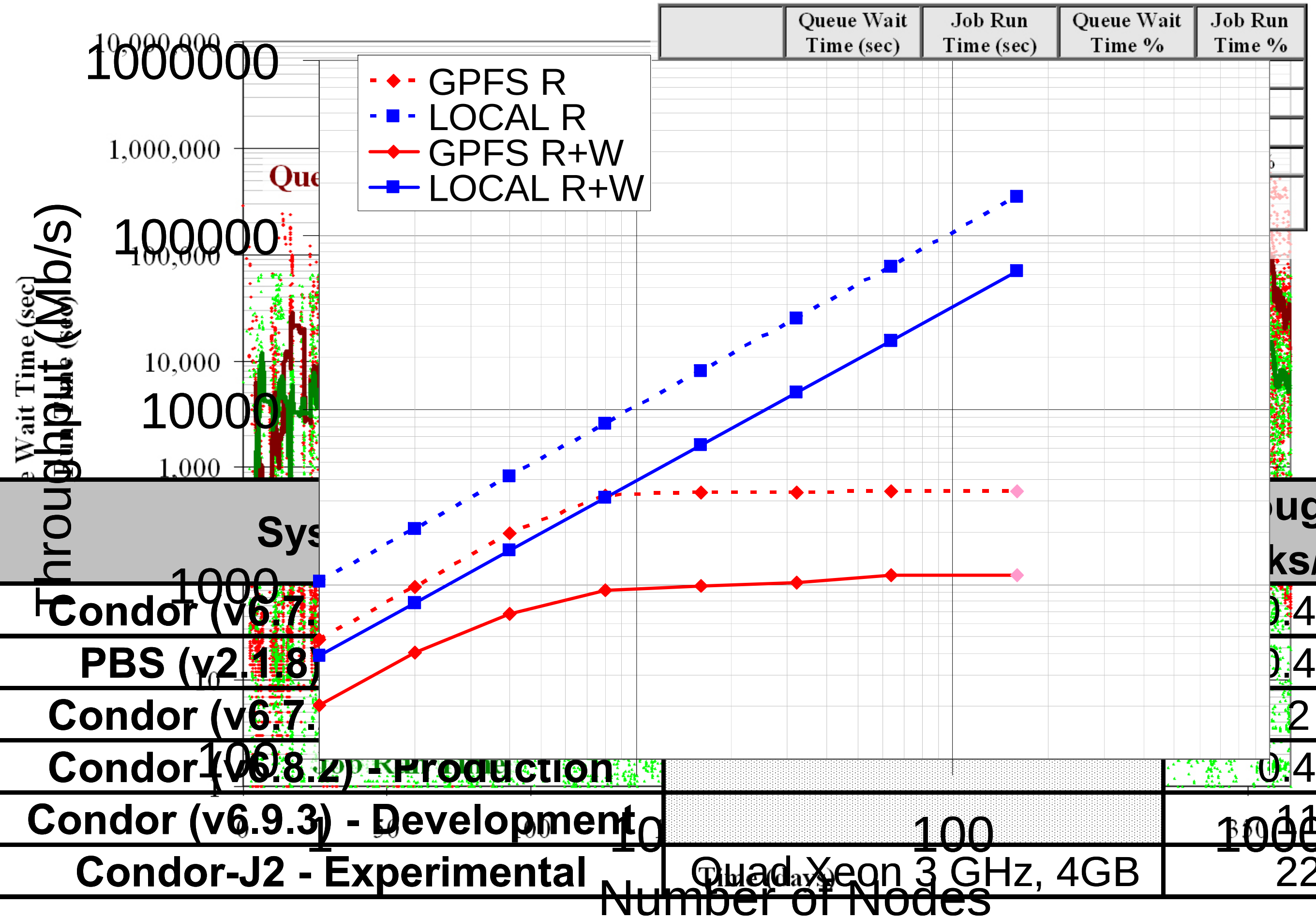
Much Data
MapReduce/Hadoop
Dryad

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

MTC: Many Task Computing

- Bridge the gap between HPC and HTC
- Loosely coupled applications with HPC orientations
- HPC comprising of multiple distinct activities, coupled via file system operations or message passing
- Emphasis on 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…

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	Quad Xeon 3 GHz, 4GB	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**

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

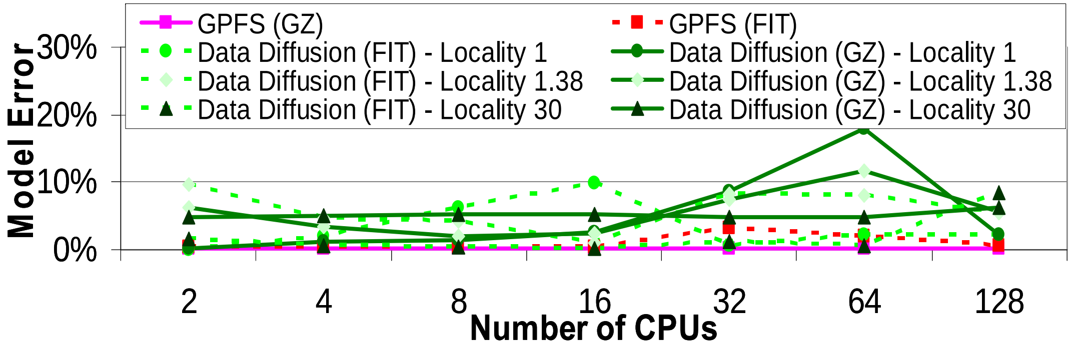
- **Speedup**

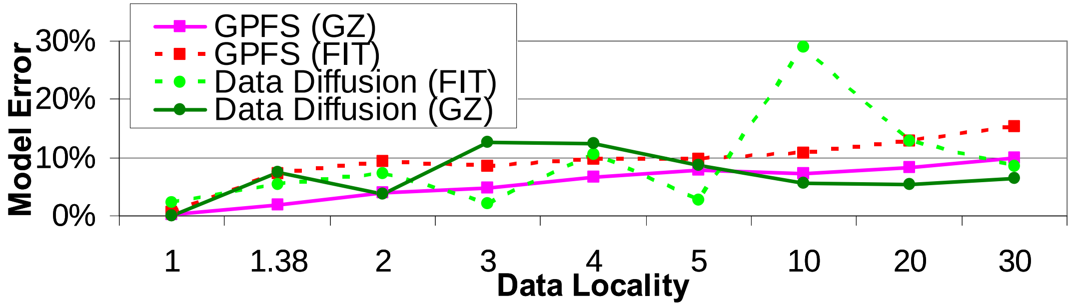
$$S = E * |T|$$

- **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

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





Talk Overview

I. Introductions
   - University of Chicago, DSL
   - University of Chicago, CI
   - Argonne National Laboratory, MCS

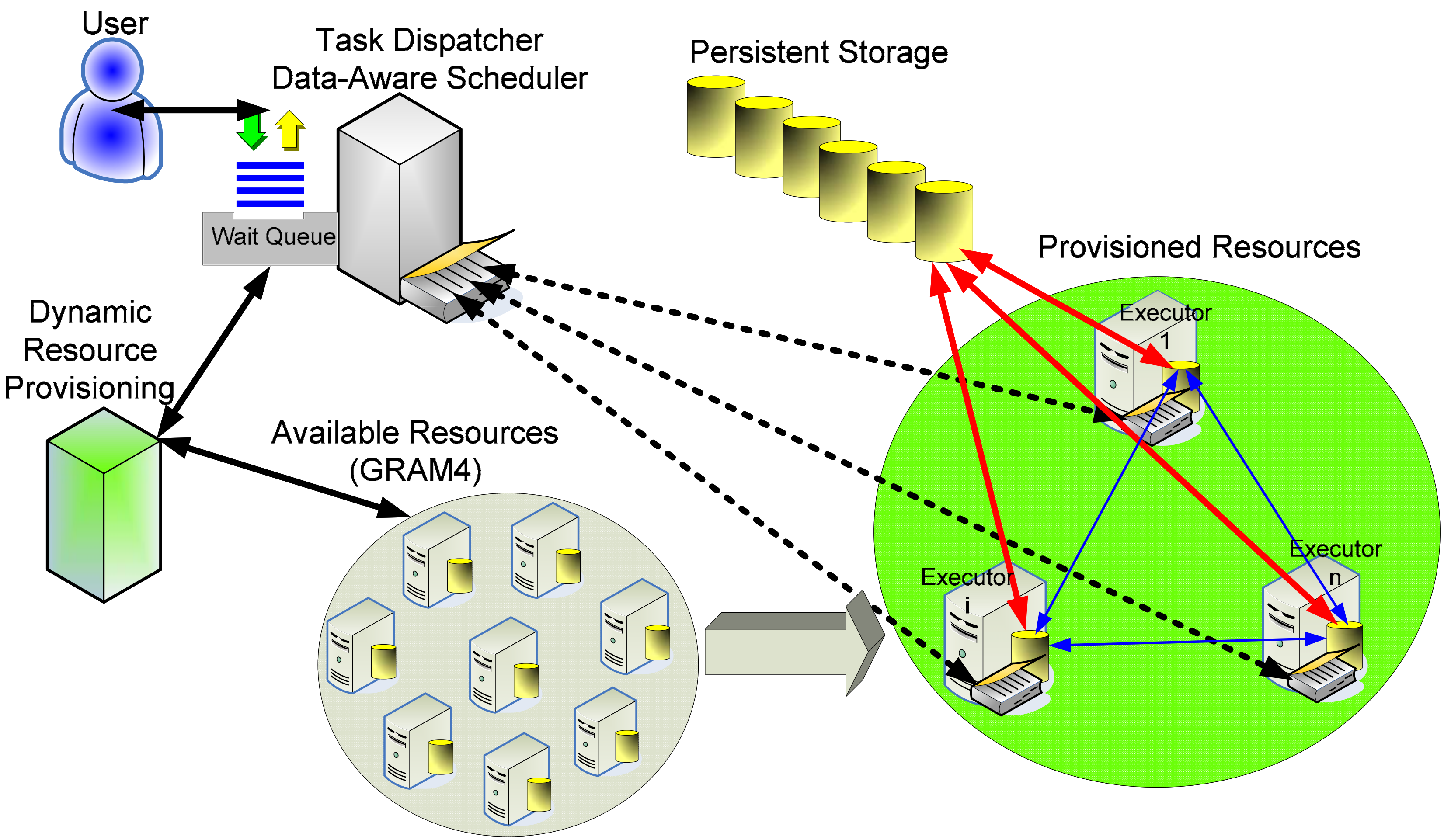
II. Comparing Grids and Clouds

III. Scalable resource management challenges and solutions
   - Dispatch
   - Provisioning
   - Data Management

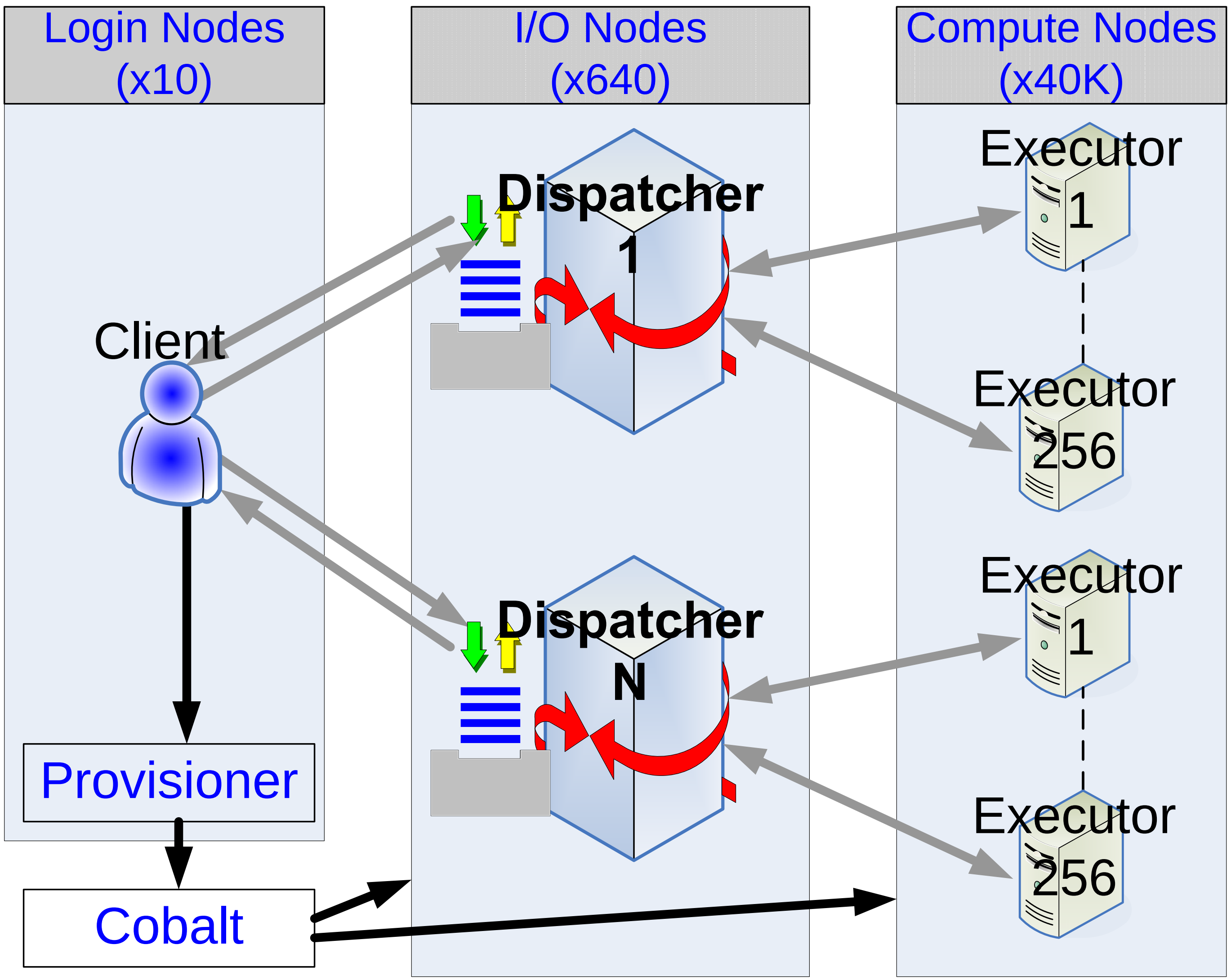
Falkon: a Fast and Light-weight tasK executiON framework

- ***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

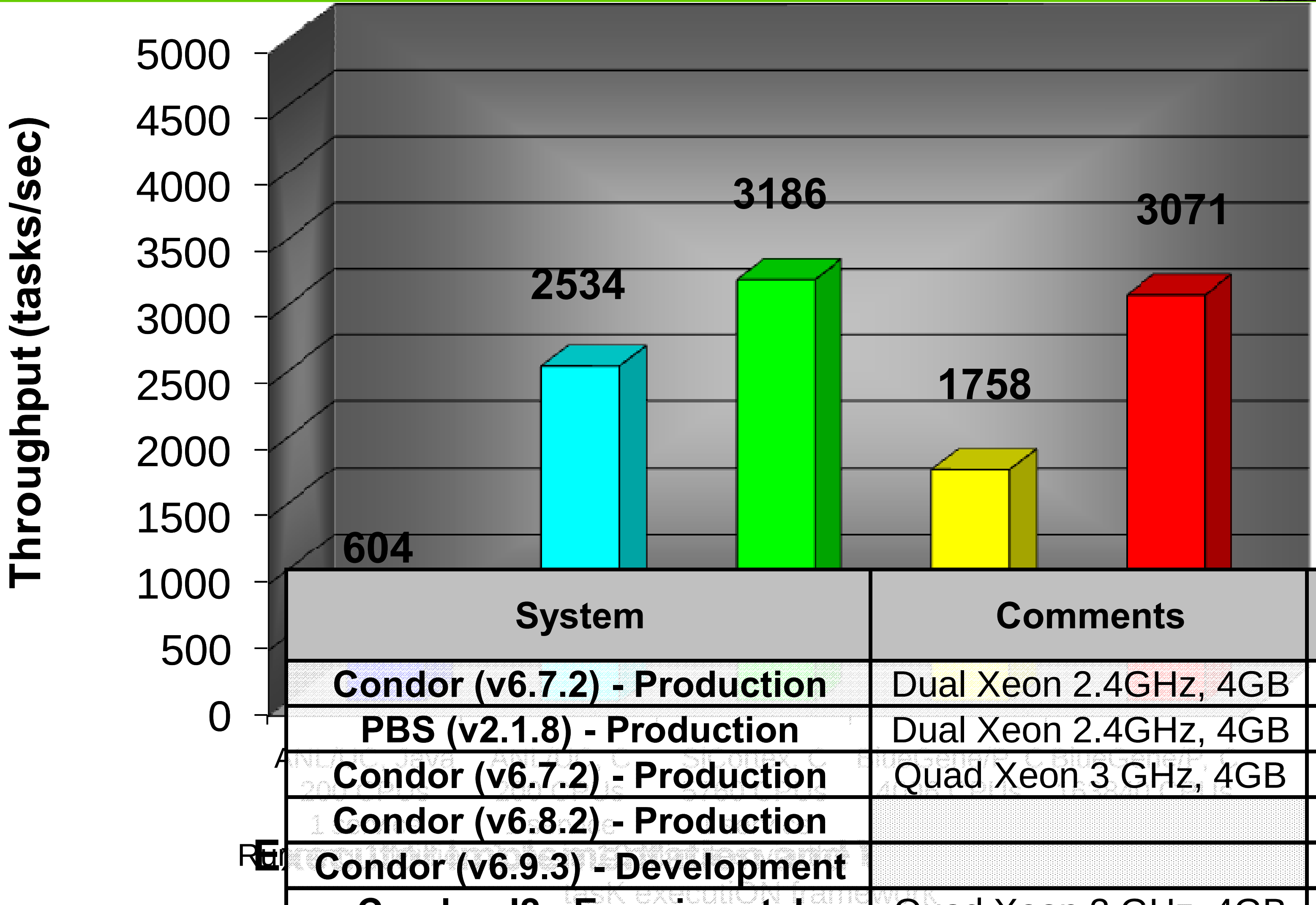
Falkon Overview



Distributed Falkon Architecture

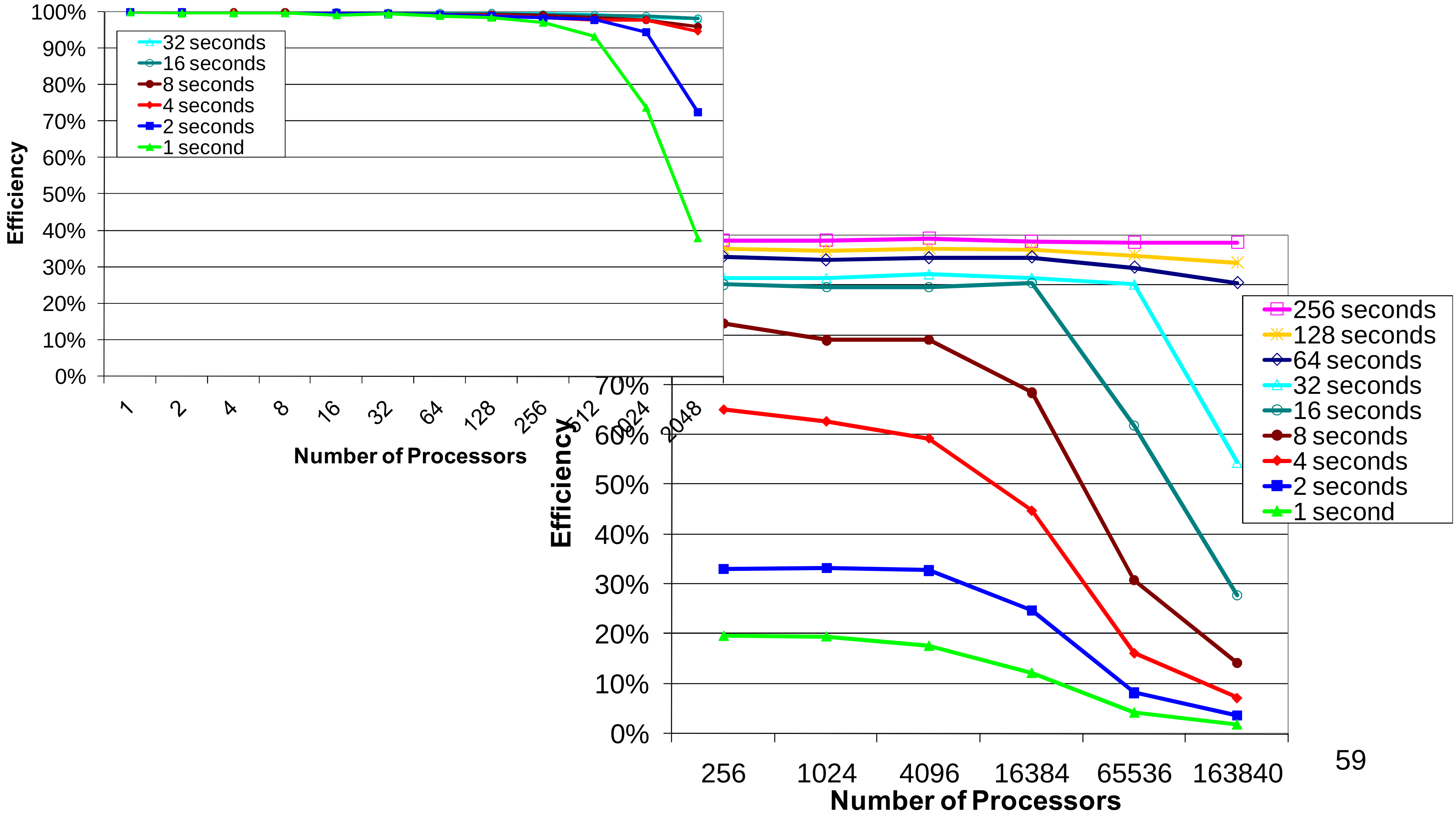


Dispatch Throughput



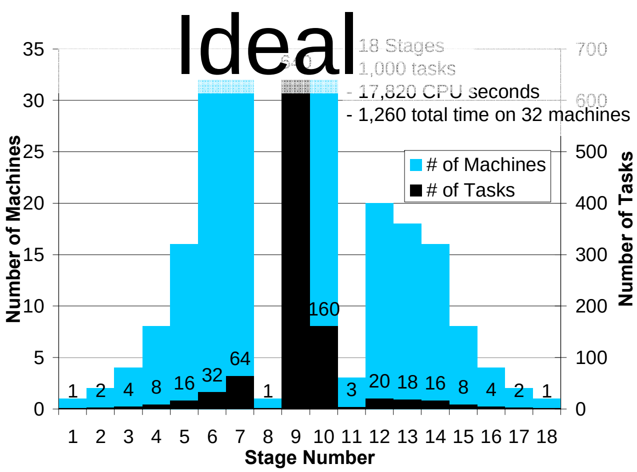
System	Comments	Throughput (tasks/sec)
Condor (v6.7.2) - Production	Dual Xeon 2.4GHz, 4GB	0.49
PBS (v2.1.8) - Production	Dual Xeon 2.4GHz, 4GB	0.45
Condor (v6.7.2) - Production	Quad Xeon 3 GHz, 4GB	2
Condor (v6.8.2) - Production		0.42
Condor (v6.9.3) - Development		11
Condor-J2 - Experimental	Quad Xeon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

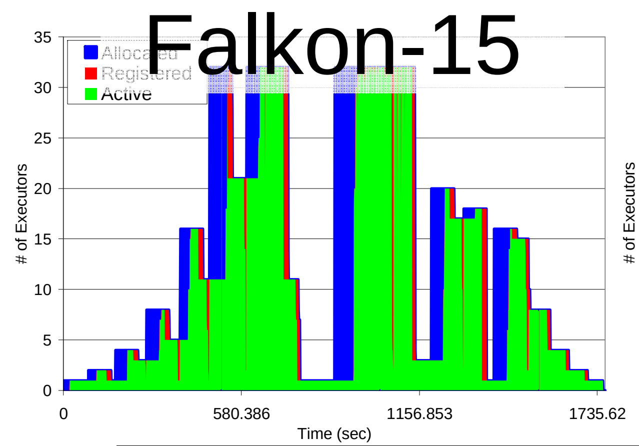
Efficiency

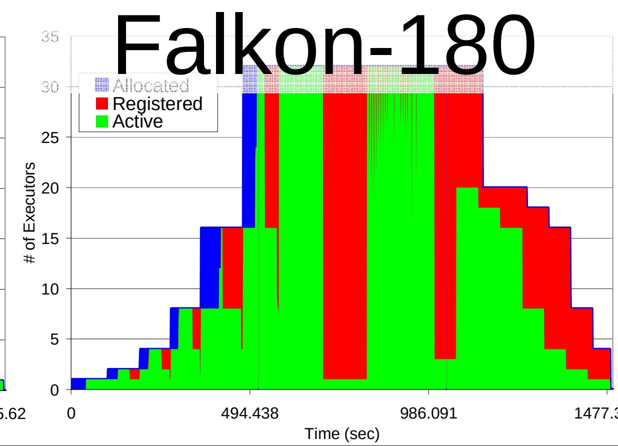


100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Efficiency
32 seconds
16 seconds
8 seconds
4 seconds
2 seconds
1 second
1
2
4
8
16
32
64
128
256
512
1024
2048
Number of Processors
Efficiency
256 seconds
128 seconds
64 seconds
32 seconds
16 seconds
8 seconds
4 seconds
2 seconds
1 second
256
1024
4096
16384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

Resource Provisioning



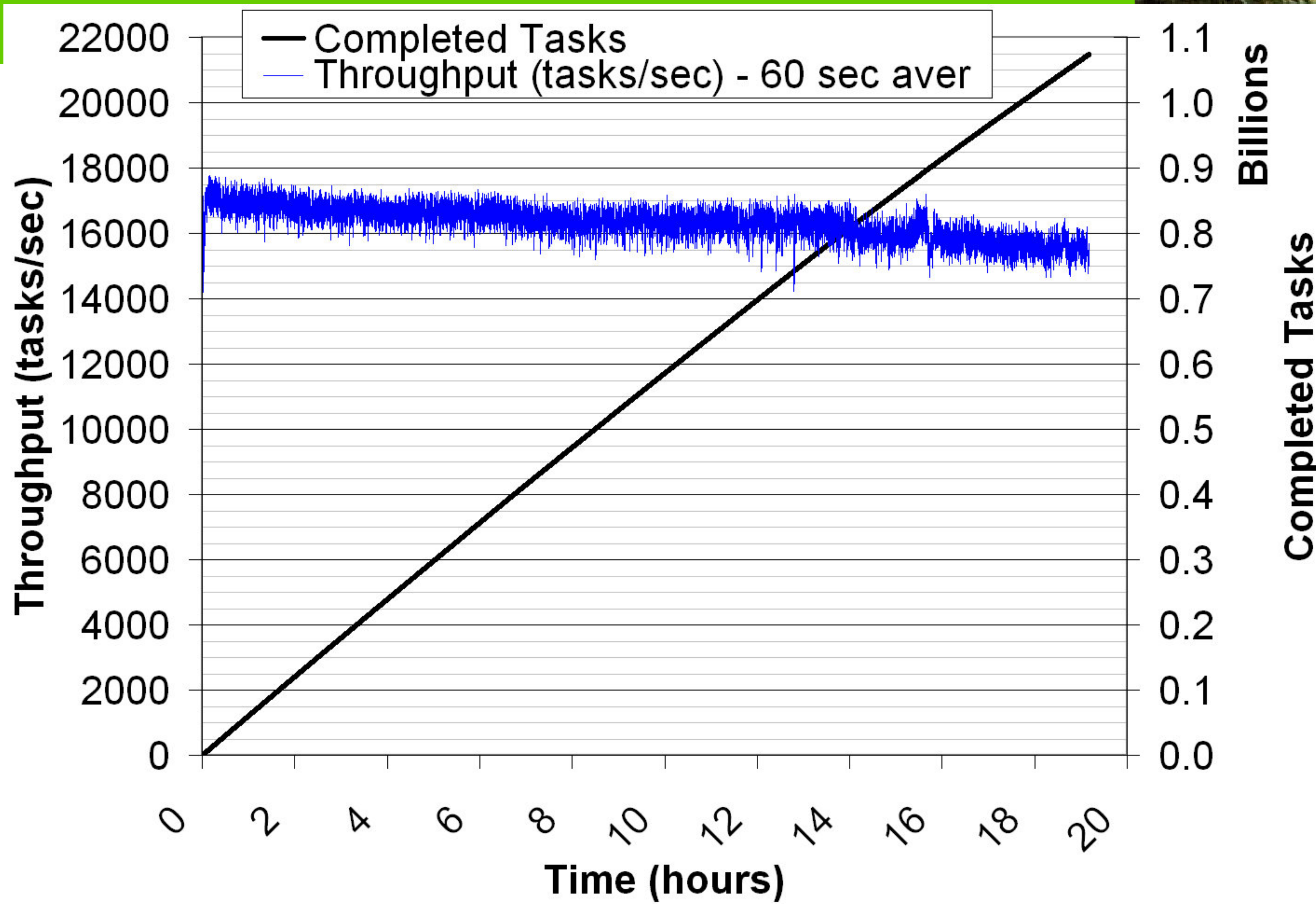




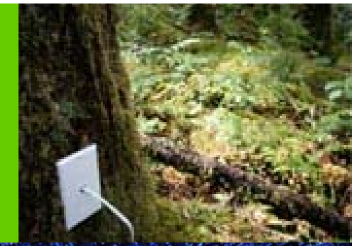
- End-to-end execution time:
  - 1260 sec in ideal case
  - 4904 sec → 1276 sec
- Average task queue time:
  - 42.2 sec in ideal case
  - 611 sec → 43.5 sec
- Trade-off:
  - Resource Utilization for Execution Efficiency

	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

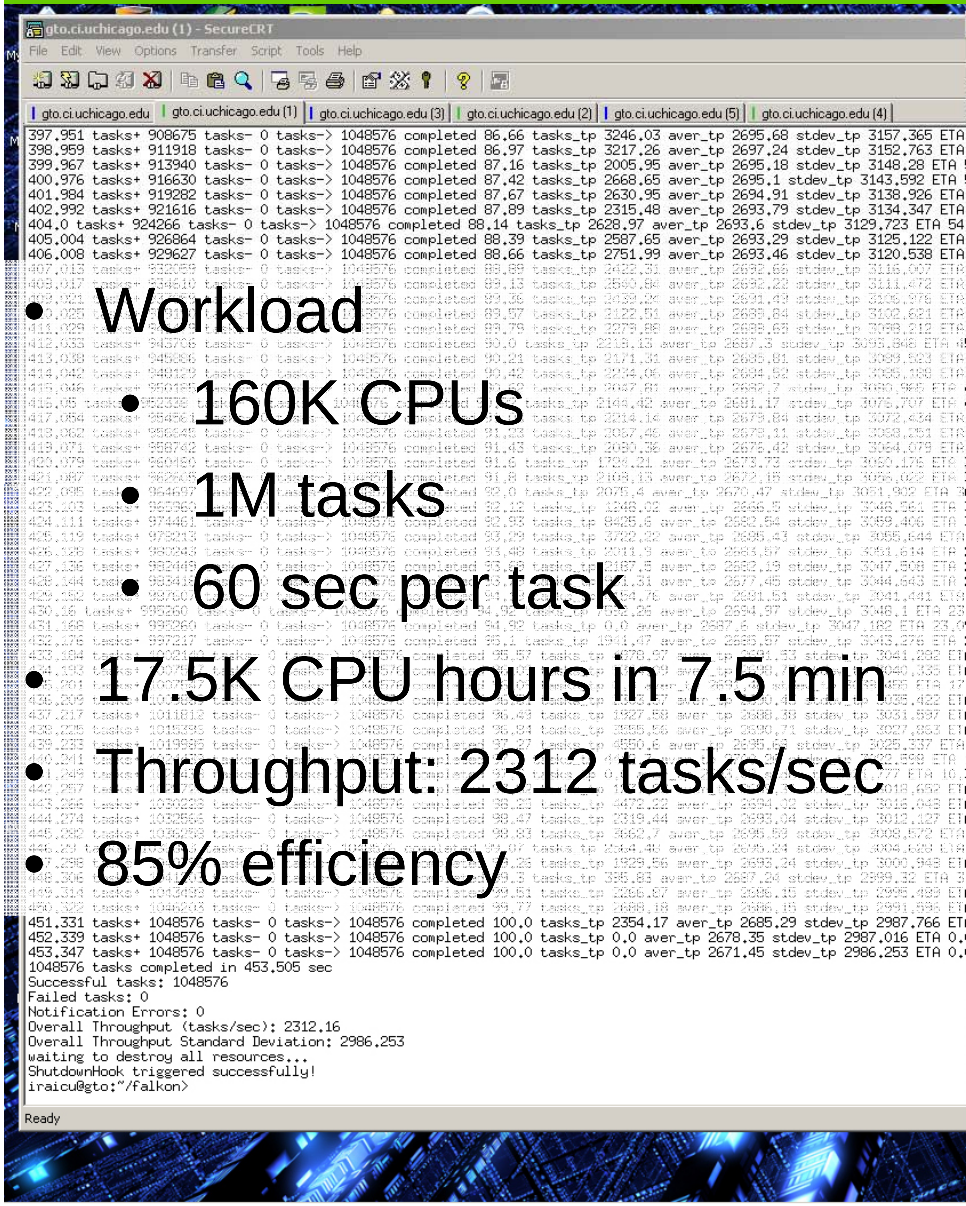
Falkon Endurance Test

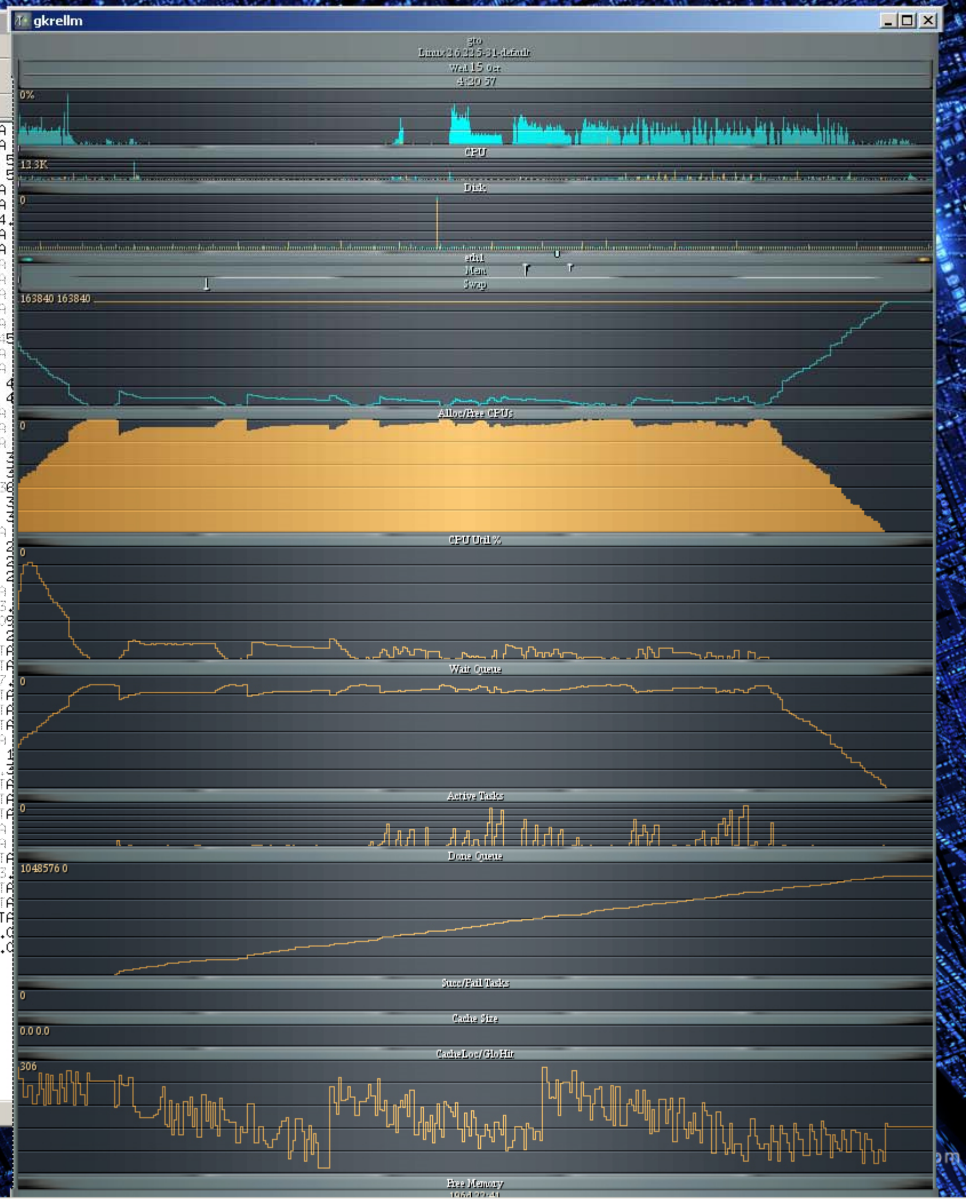


Falkon Monitoring

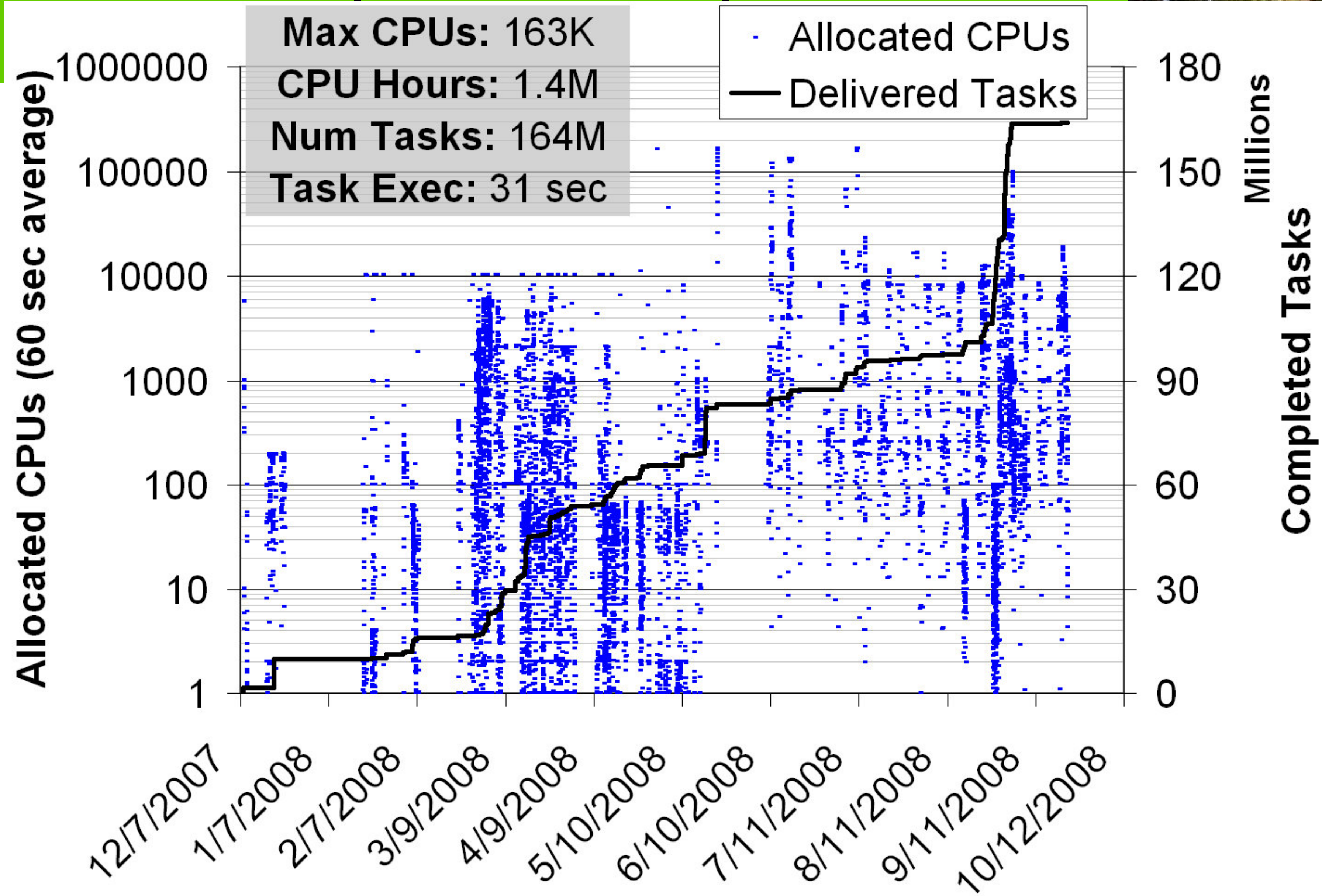


- Workload
  - 160K CPUs
  - 1M tasks
  - 60 sec per task
- 17.5K CPU hours in 7.5 min
- Throughput: 2312 tasks/sec
- 85% efficiency

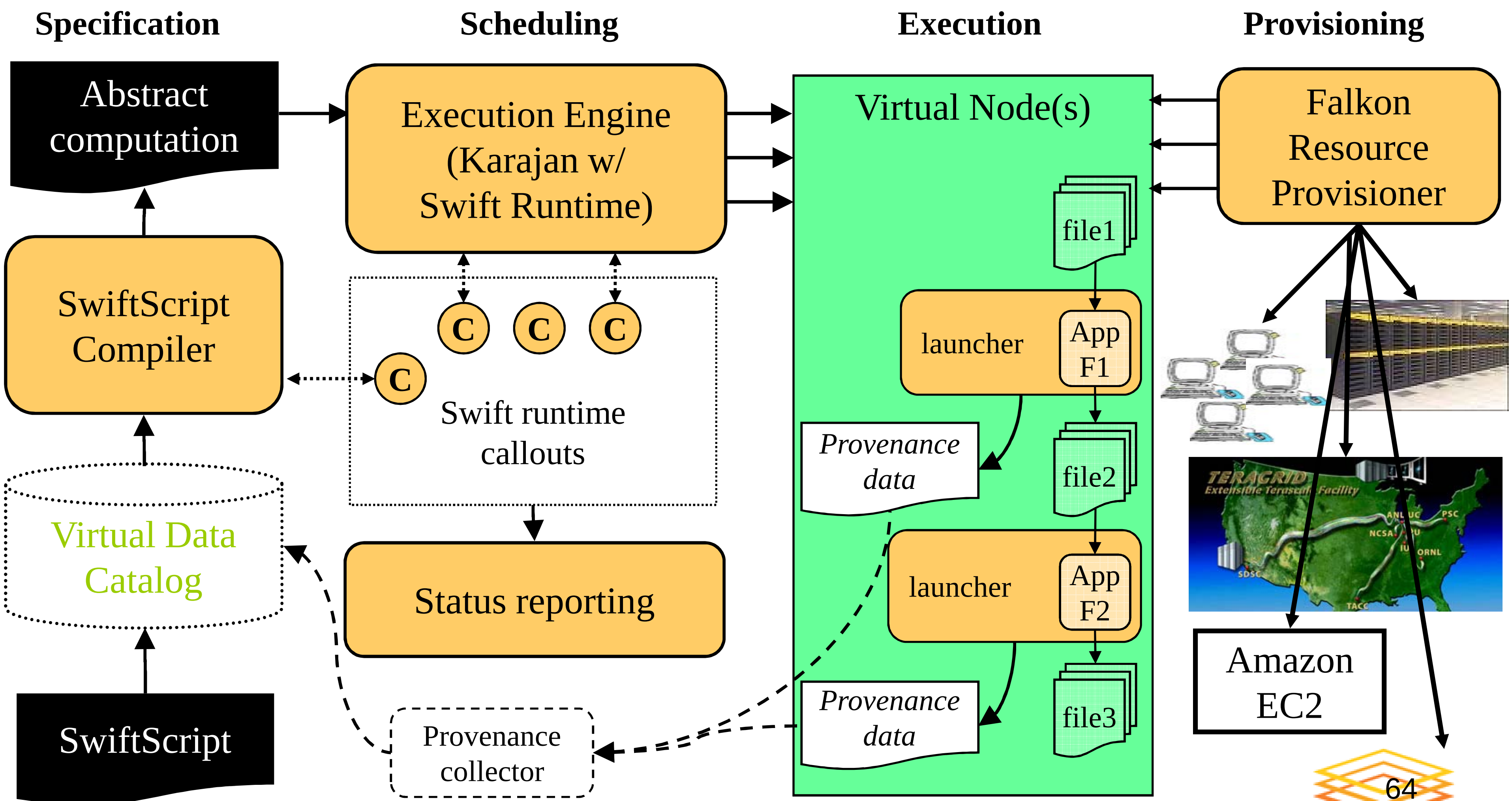




Falkon Activity History (10 months)

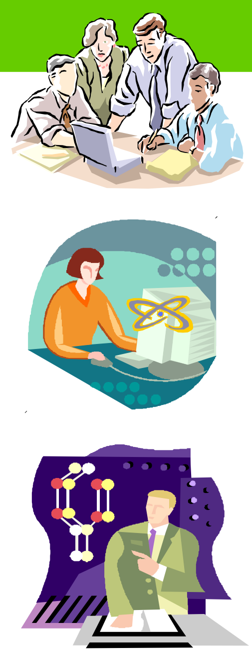


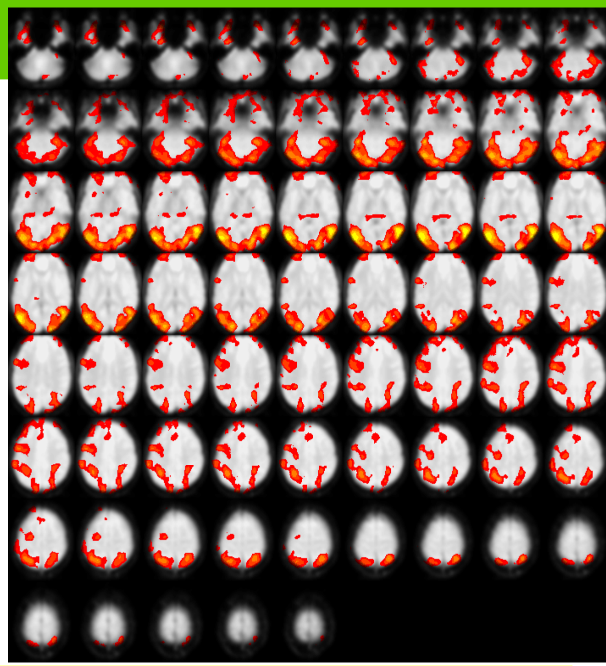
Swift Architecture

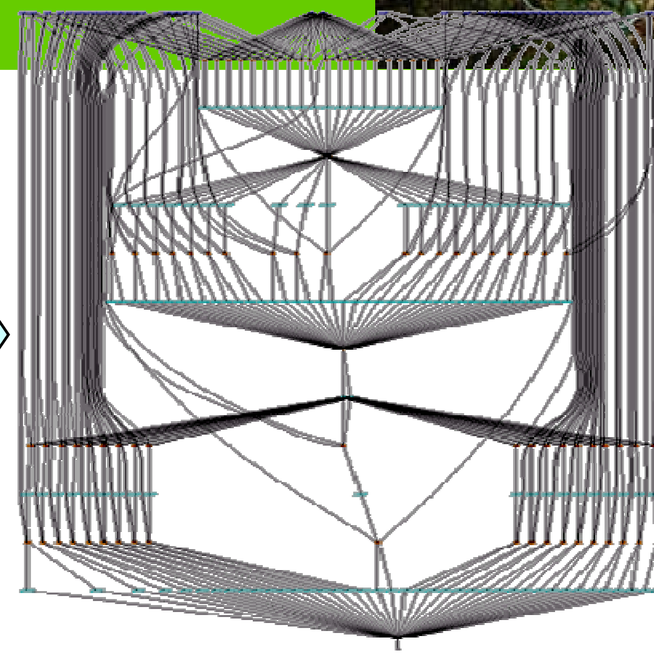


Open Science Grid

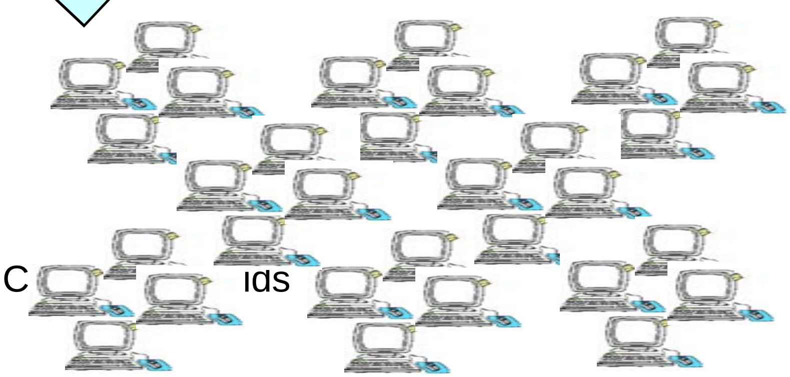
Functional MRI (fMRI)





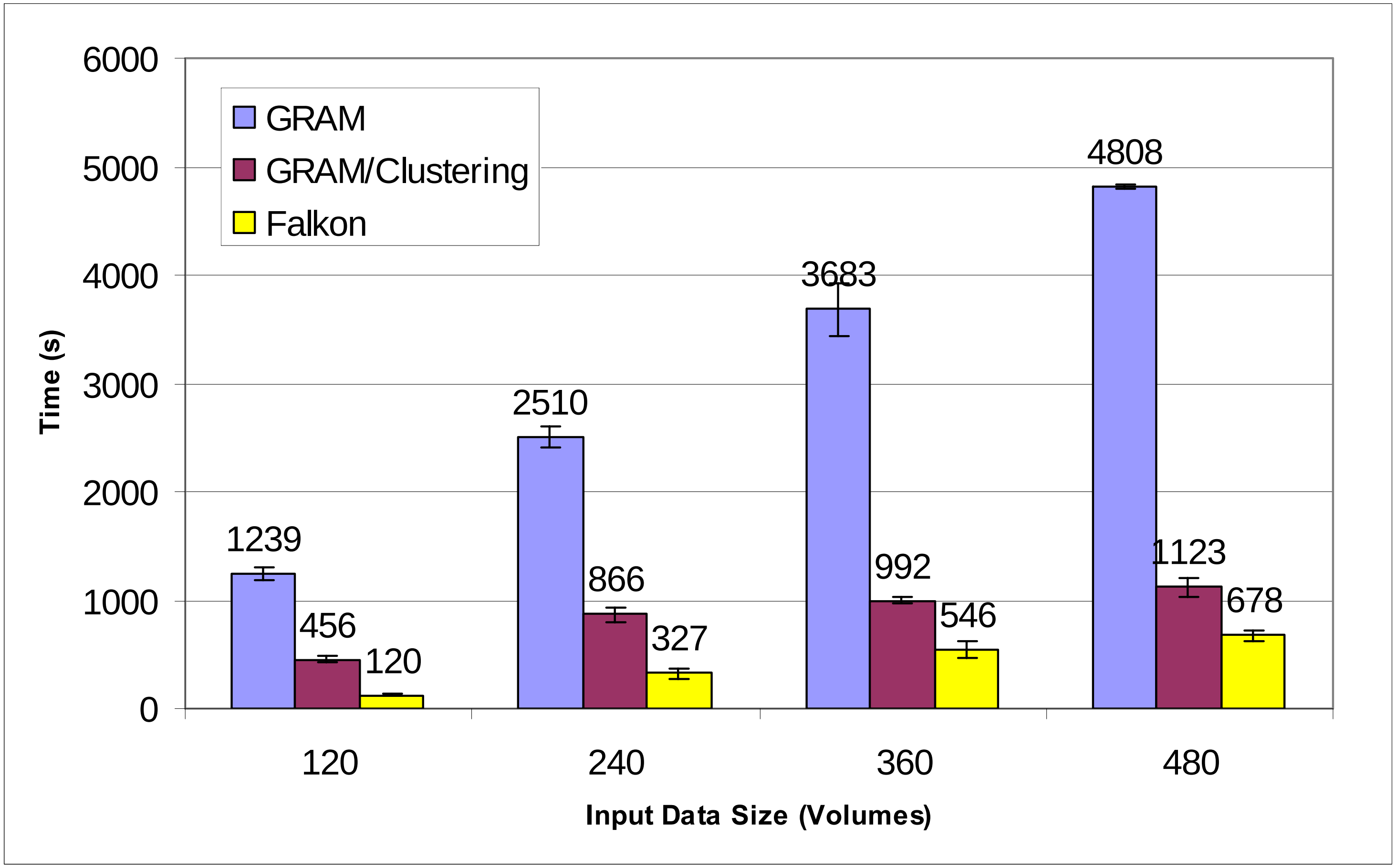


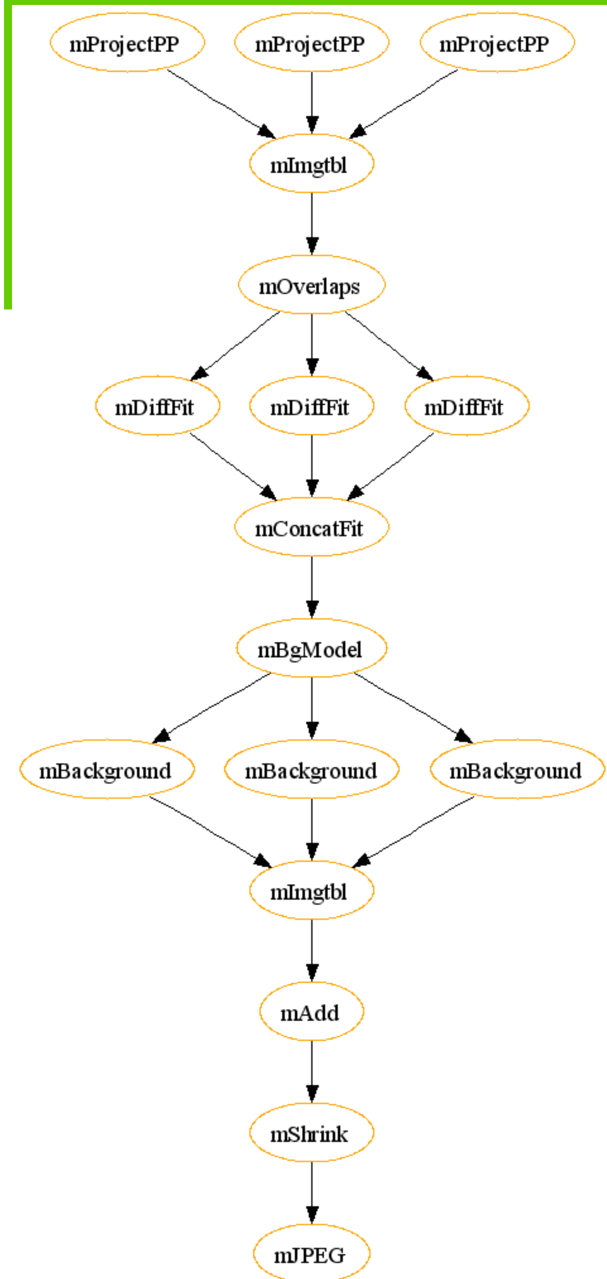
- Wide range of analyses
  - Testing, interactive analysis, production runs
  - Data mining
  - Parameter studies



Completed Milestones: fMRI Application

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



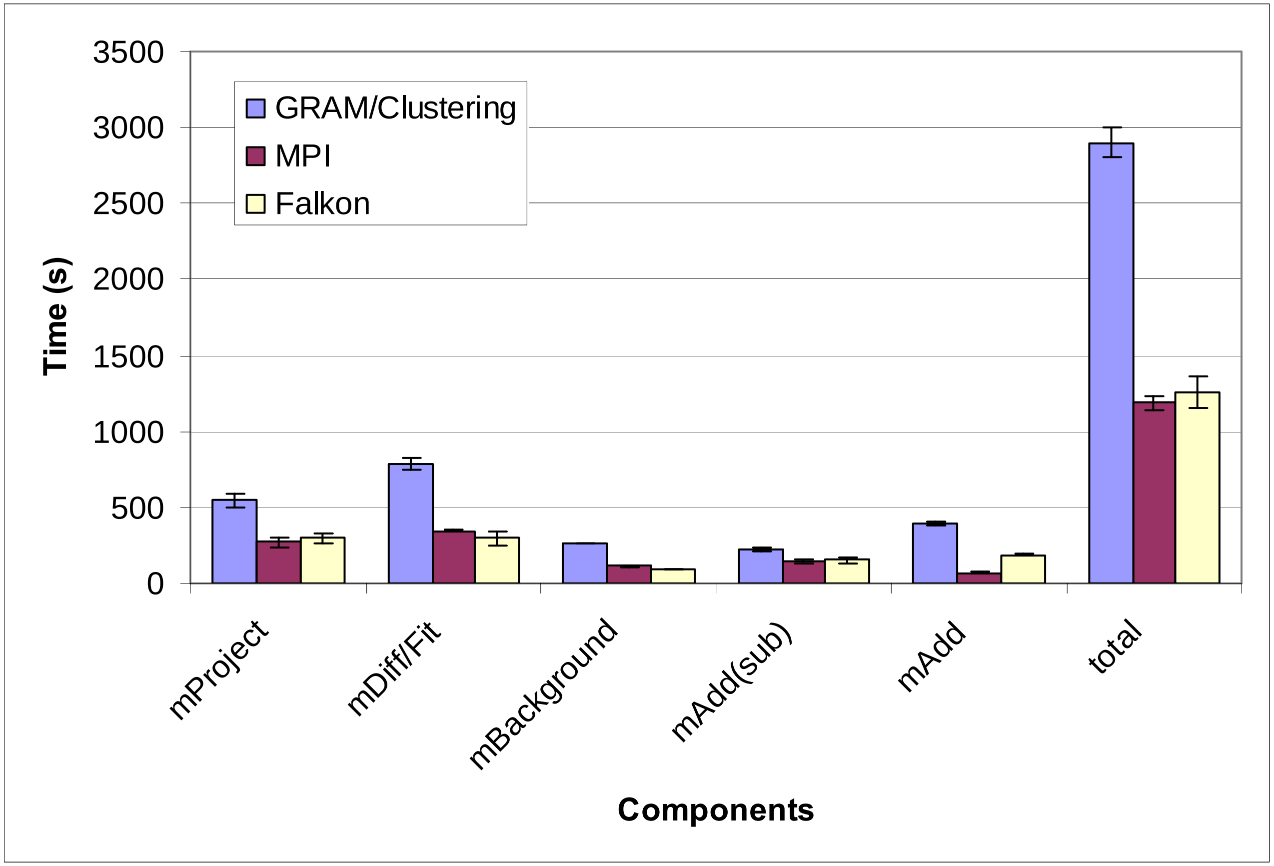


Montage

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

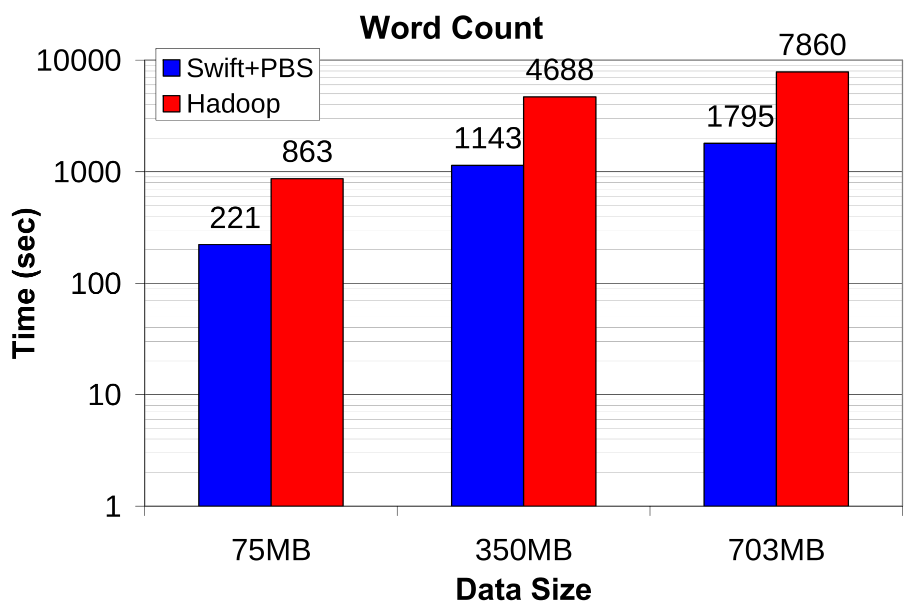
Completed Milestones: Montage Application

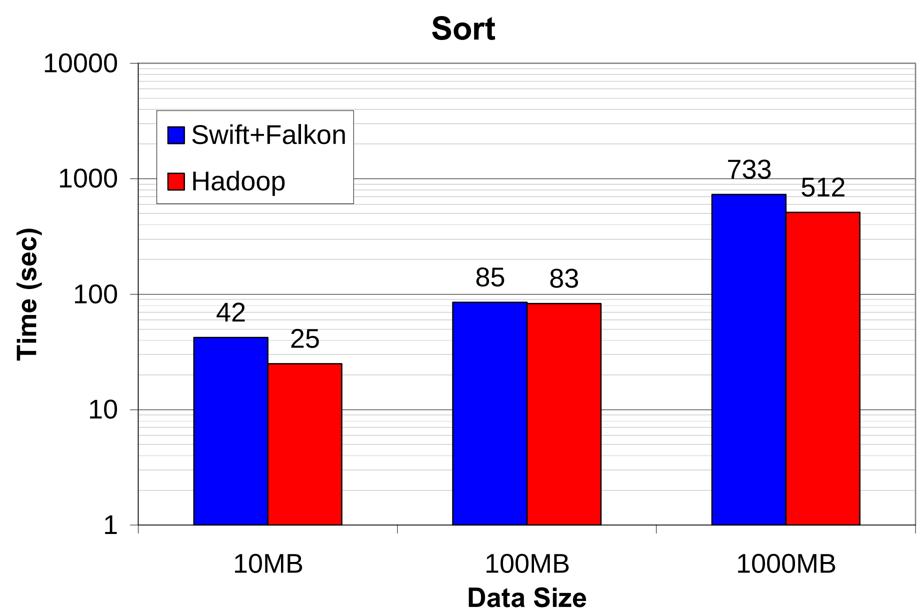
- GRAM/Clustering vs. Falkon: 57% lower application run time
- MPI* vs. Falkon: 4% higher application run time
- * MPI should be lower bound



Hadoop vs. Swift

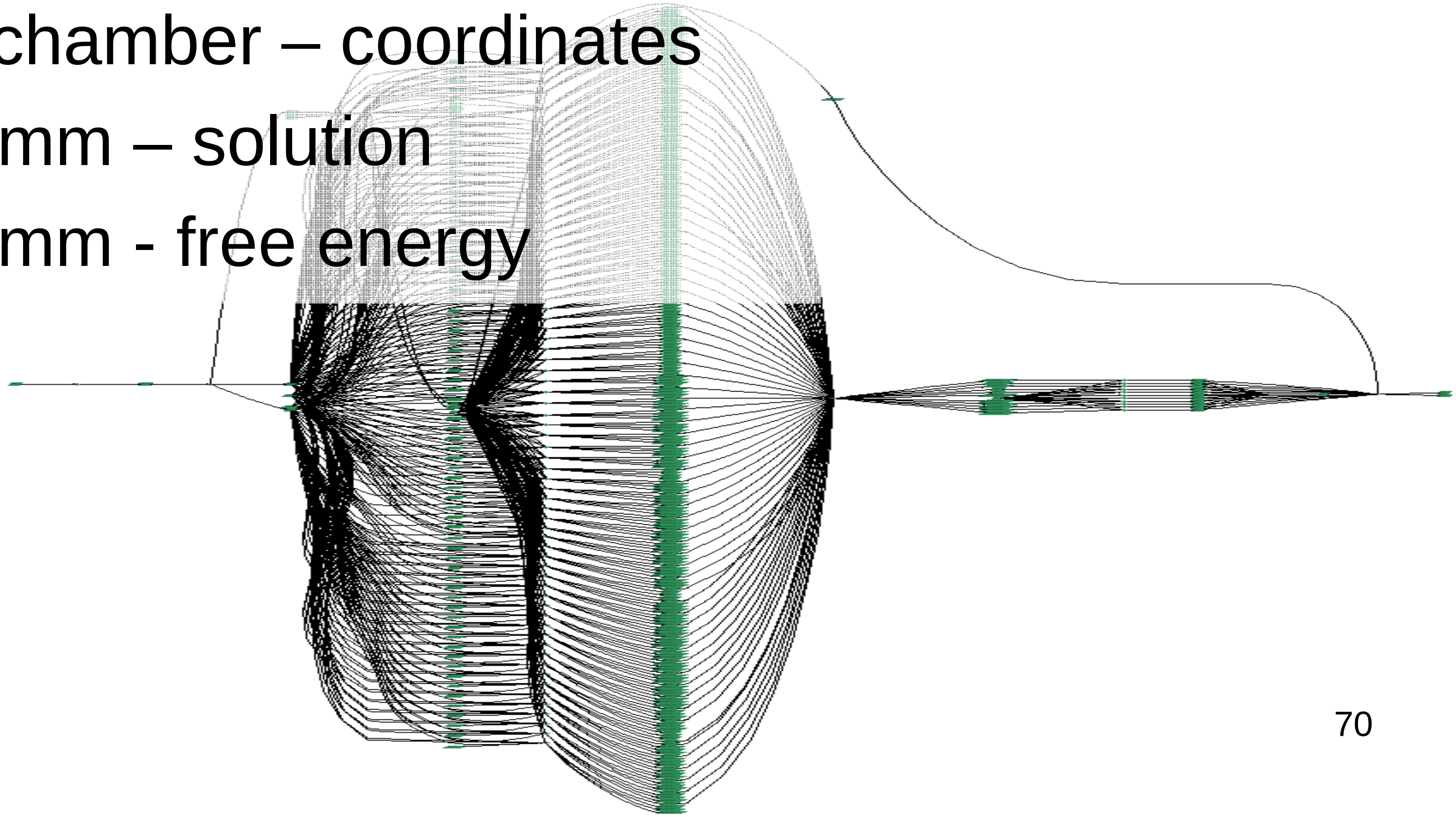
- Classic benchmarks for MapReduce
  - Word Count
  - Sort
- Swift performs similar or better than Hadoop (on 32 processors)





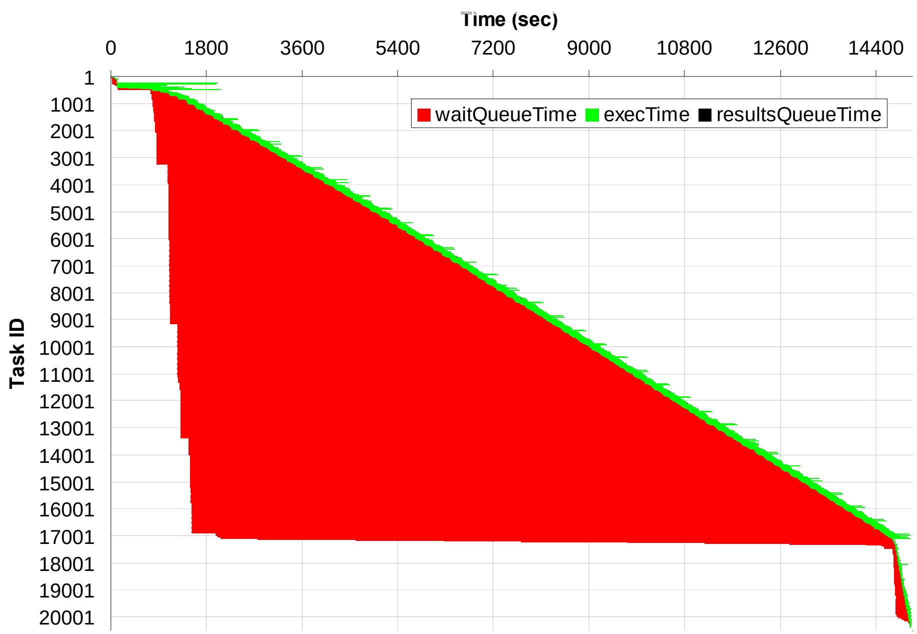
Molecular Dynamics

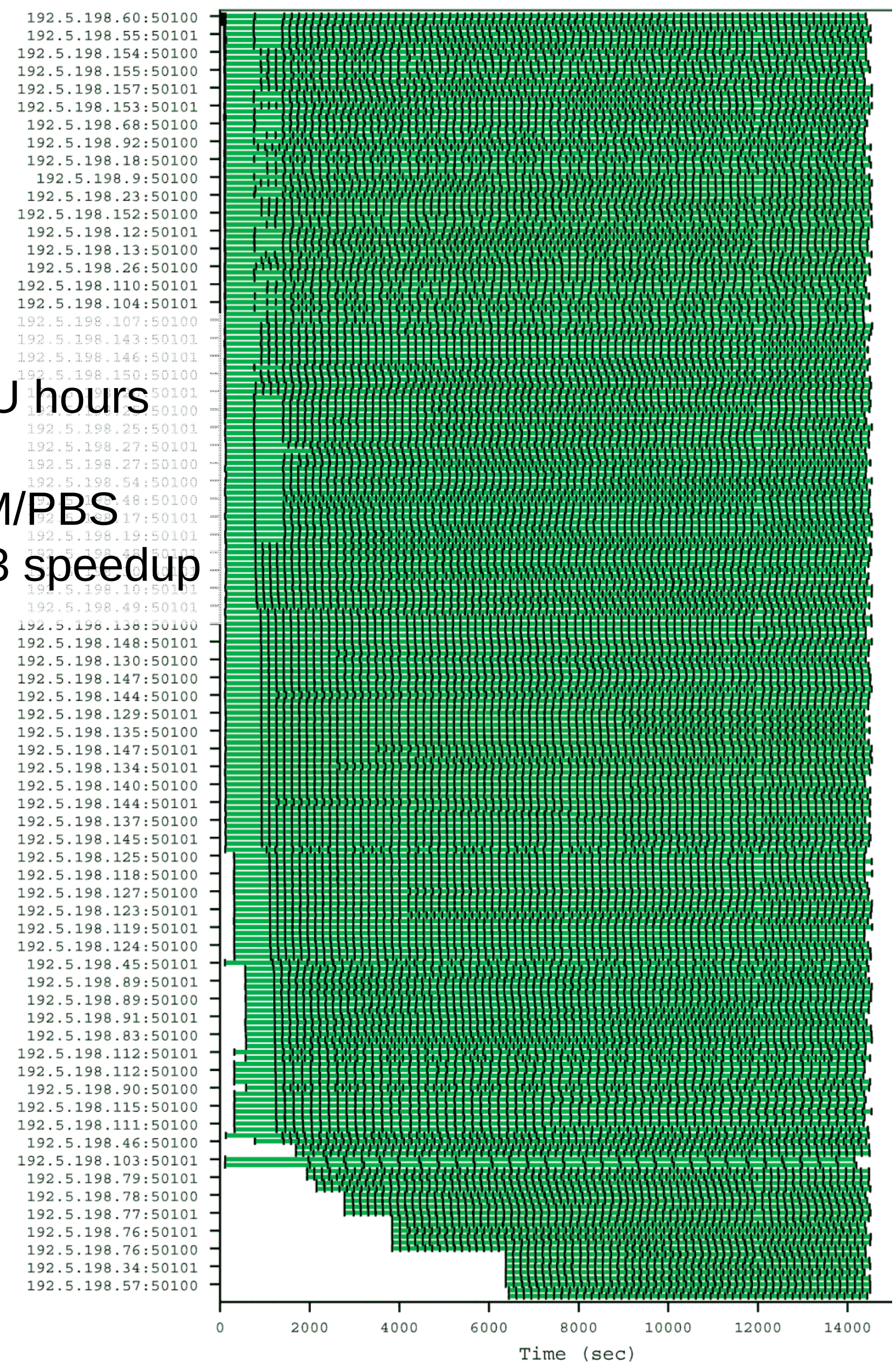
- Determination of free energies in aqueous solution
  - Antechamber – coordinates
  - Charmm – solution
  - Charmm - free energy



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





MTC: Many Task Computing

- Bridge the gap between HPC and HTC
- Loosely coupled applications with HPC orientations
- HPC comprising of multiple distinct activities, coupled via file system operations or message passing
- Emphasis on 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…

Growing Interest on enabling HTC/MTC on Supercomputers

- Project Kittyhawk
  - IBM Research
- HTC-mode in Cobalt/BG
  - IBM
- Condor on BG
  - University of Wisconsin at Madison, IBM
- Grid Enabling the BG
  - University of Colorado, National Center for Atmospheric Research
- Plan 9
  - Bell Labs, IBM Research, Sandia National Labs
- Falkon/Swift on BG/P and Sun Constellation
  - University of Chicago, Argonne National Laboratory

Many Large Systems available for Open Science Research

- Jaguar (#2)
  – DOE, ORNL
- Intrepid (#5)
  – DOE, ANL
- Ranger (#6)
  – NSF, TACC

Home ▸ Lists ▸ November 2008

TOP500 List - November 2008 (1-100)

R_{max} and R_{peak} values are in TFlops. For more details about other fields, check the TOP500 description.

Power data in KW for entire system

nex

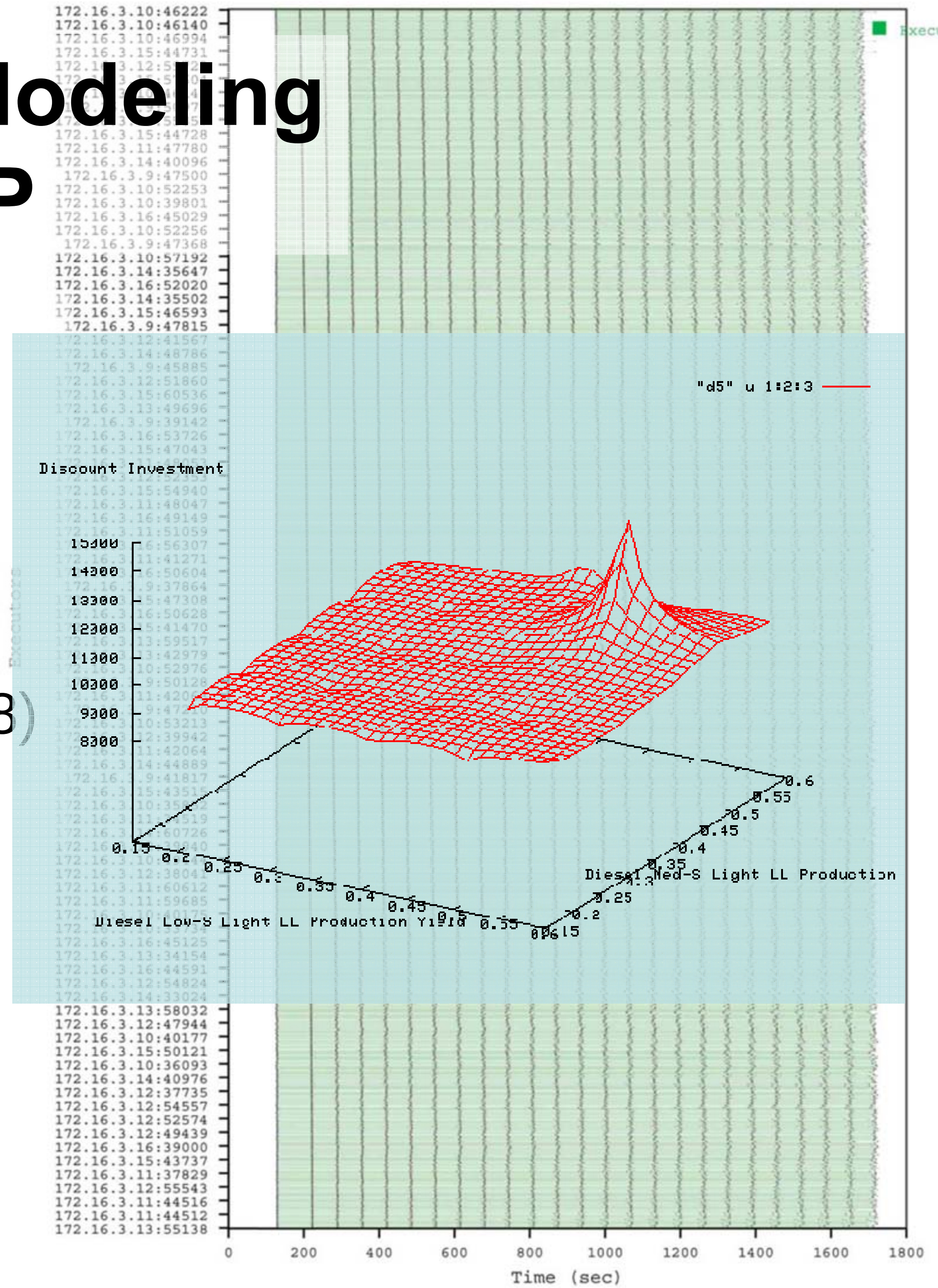
Rank	Site	Computer/Year Vendor	Cores	R_{max}	R_{peak}	Power
1	DOE/NNSA/LANL United States	Roadrunner - BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz , Voltaire Infiniband / 2008 IBM	129600	1105.00	1456.70	2483.47
2	Oak Ridge National Laboratory United States	Jaguar - Cray XT5 QC 2.3 GHz / 2008 Cray Inc.	150152	1059.00	1381.40	6950.60
3	NASA/Ames Research Center/NAS United States	Pleiades - SGI Altix ICE 8200EX, Xeon QC 3.0/2.66 GHz / 2008 SGI	51200	487.01	608.83	2090.00
4	DOE/NNSA/LLNL United States	BlueGene/L - eServer Blue Gene Solution / 2007 IBM	212992	478.20	596.38	2329.60
5	Argonne National Laboratory United States	Blue Gene/P Solution / 2007 IBM	163840	450.30	557.06	1260.00
6	Texas Advanced Computing Center/Univ. of Texas United States	Ranger - SunBlade x6420, Opteron QC 2.3 Ghz, Infiniband / 2008 Sun Microsystems	62976	433.20	579.38	2000.00
7	NERSC/LBNL United States	Franklin - Cray XT4 QuadCore 2.3 GHz / 2008 Cray Inc.	38642	266.30	355.51	1150.00
8	Oak Ridge National Laboratory United States	Jaguar - Cray XT4 QuadCore 2.1 GHz / 2008 Cray Inc.	30976	205.00	260.20	1580.71
9	NNSA/Sandia National Laboratories United States	Red Storm - Sandia/ Cray Red Storm, XT3/4, 2.4/2.2 GHz dual/quad core / 2008 Cray Inc.	38208	204.20	284.00	2506.00
10	Shanghai Supercomputer Center China	Dawning 5000A - Dawning 5000A, QC Opteron 1.9 Ghz, Infiniband, Windows HPC 2008 / 2008 Dawning	30720	180.60	233.47	

Why Petascale Systems for MTC Applications?

1. The I/O subsystem of petascale systems offers unique capabilities needed by MTC applications
2. The cost to manage and run on petascale systems is less than that of conventional clusters or Grids
3. Large-scale systems that favor large jobs have utilization issues
4. Some problems are intractable without petascale systems

MARS Economic Modeling on IBM BG/P

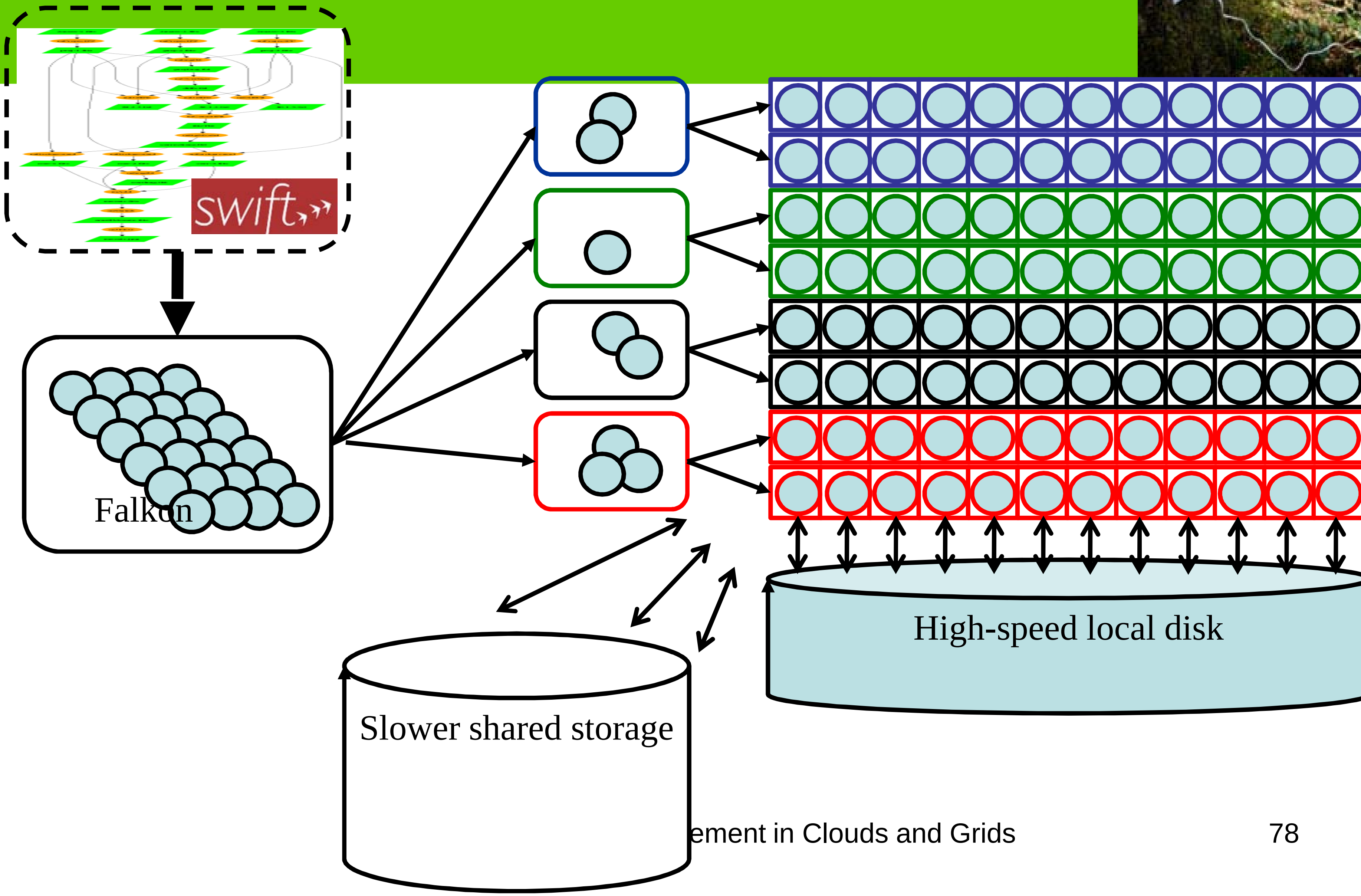
- CPU Cores: 2048
- Tasks: 49152
- Micro-tasks: 7077888
- Elapsed time: 1601 secs
- CPU Hours: 894
- Speedup: 1993X (ideal 2048)
- Efficiency: 97.3%



Scaling from 1K to 100K CPUs without Data Diffusion

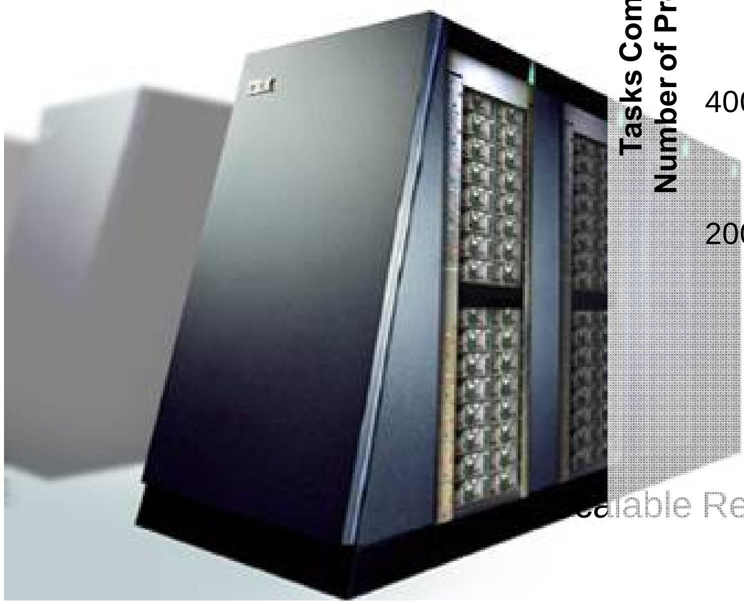
- At 1K CPUs:
  - 1 Server to manage all 1K CPUs
  - Use shared file system extensively
    - Invoke application from shared file system
    - Read/write data from/to shared file system
- At 100K CPUs:
  - N Servers to manage 100K CPUs (1:256 ratio)
  - Don’t trust the application I/O access patterns to behave optimally
    - Copy applications and input data to RAM
    - Read input data from RAM, compute, and write results to RAM
    - Archive all results in a single file in RAM
    - Copy 1 result file from RAM back to GPFS
- Great potential for improvements
  - Could leverage the Torus network for high aggregate bandwidth
  - Collective I/O (CIO) Primitives
  - Roadblocks: machine global IP connectivity, Java support, and time

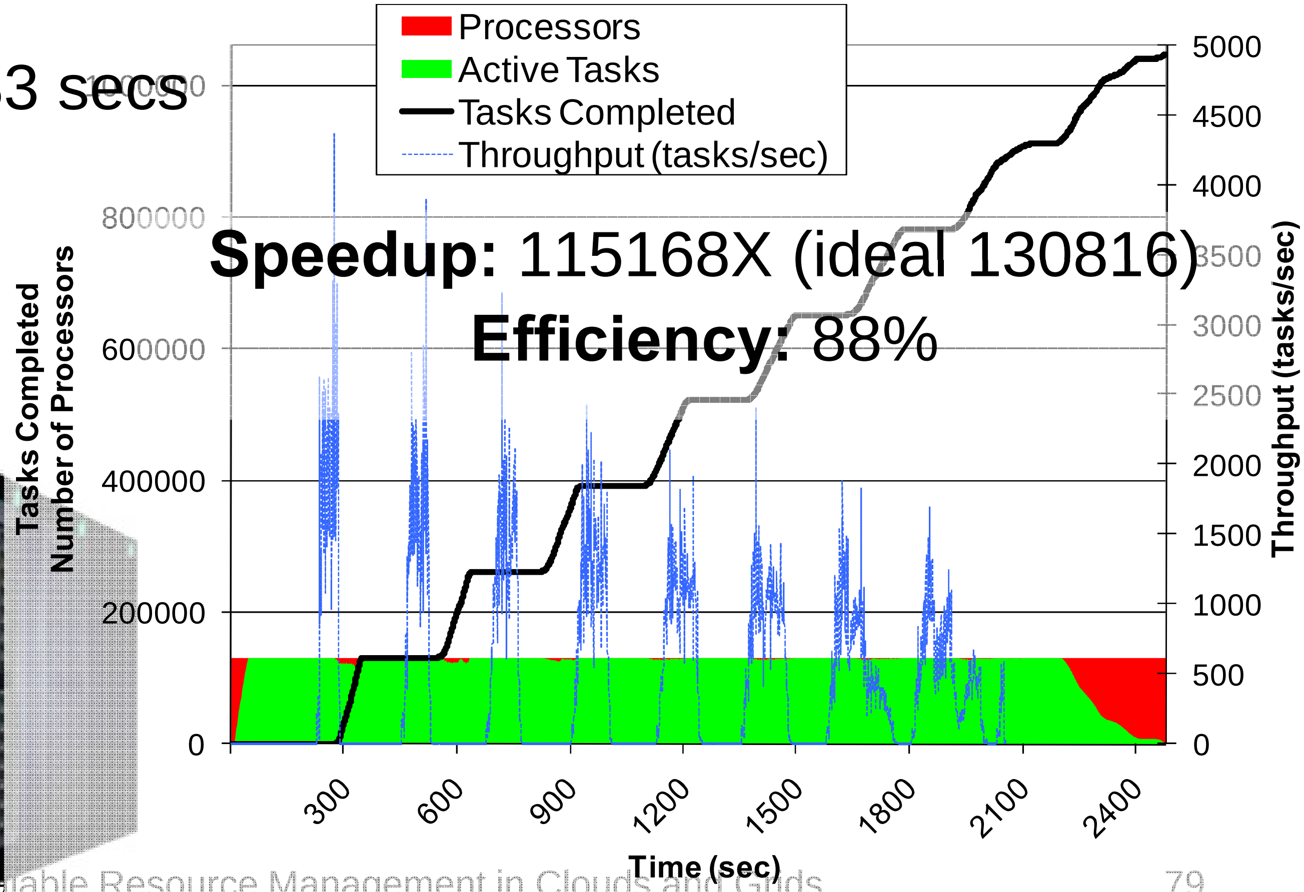
Managing 160K CPUs



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

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



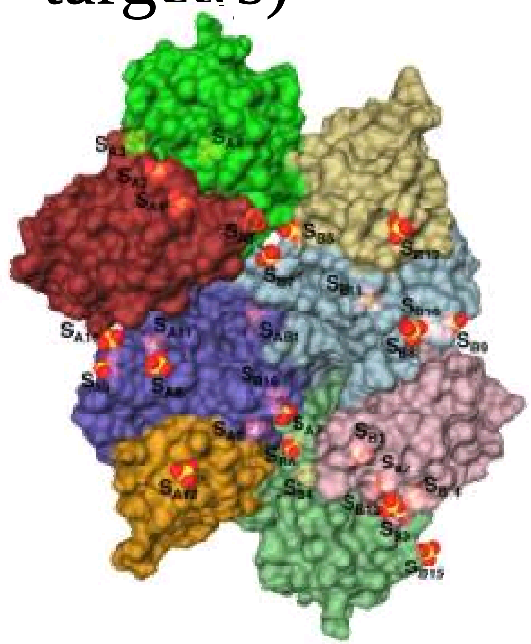


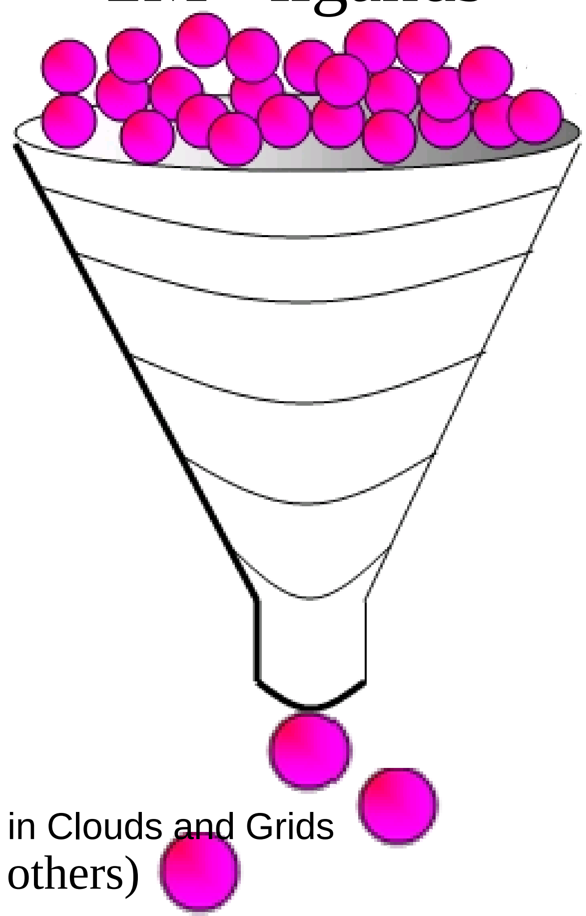
Speedup: 115168X (ideal 130816)

Efficiency: 88%

Many Many Tasks: Identifying Potential Drug Targets

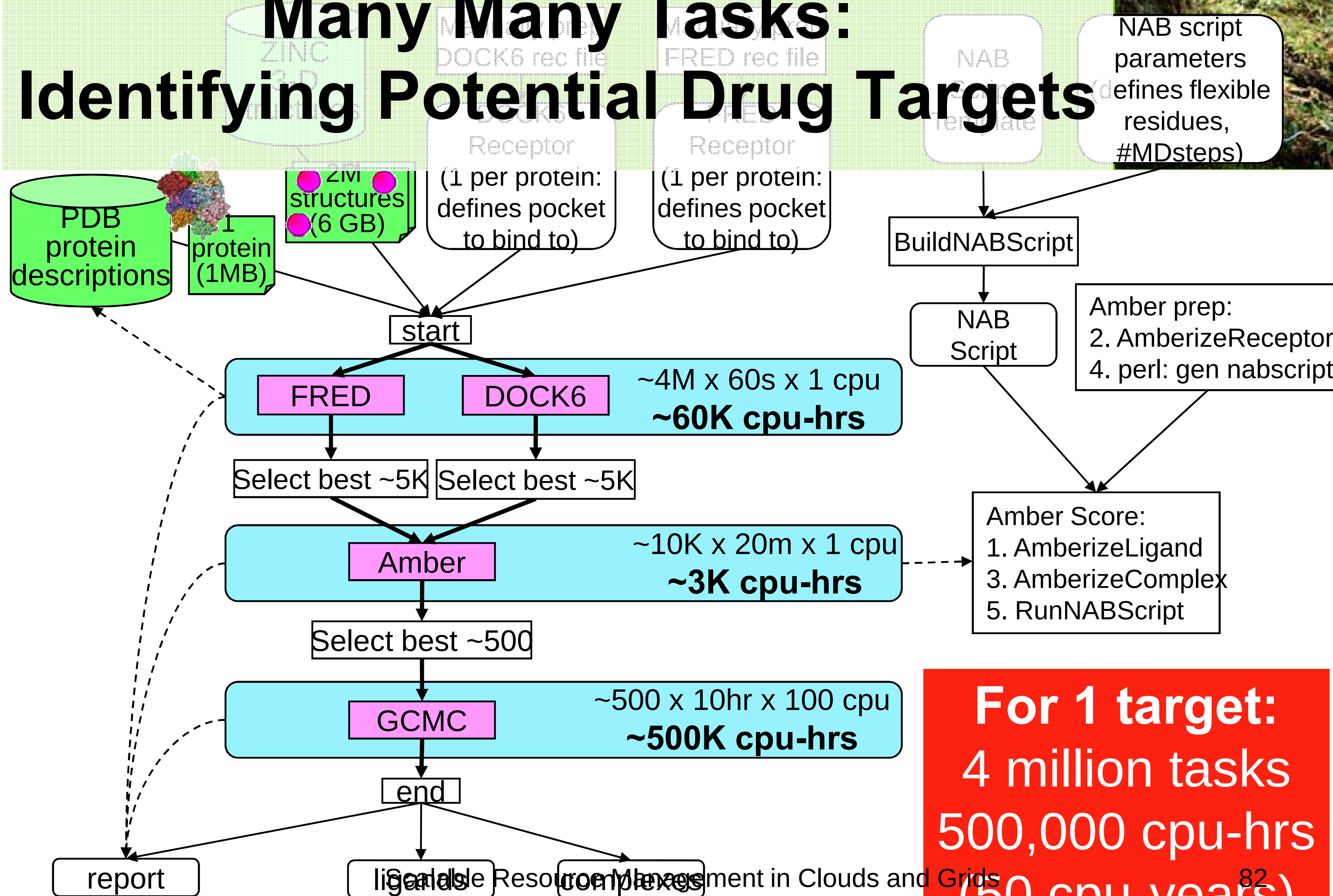
Protein target(s) x 2M+ ligands





(Mike Kubal, Benoit Roux, and others)

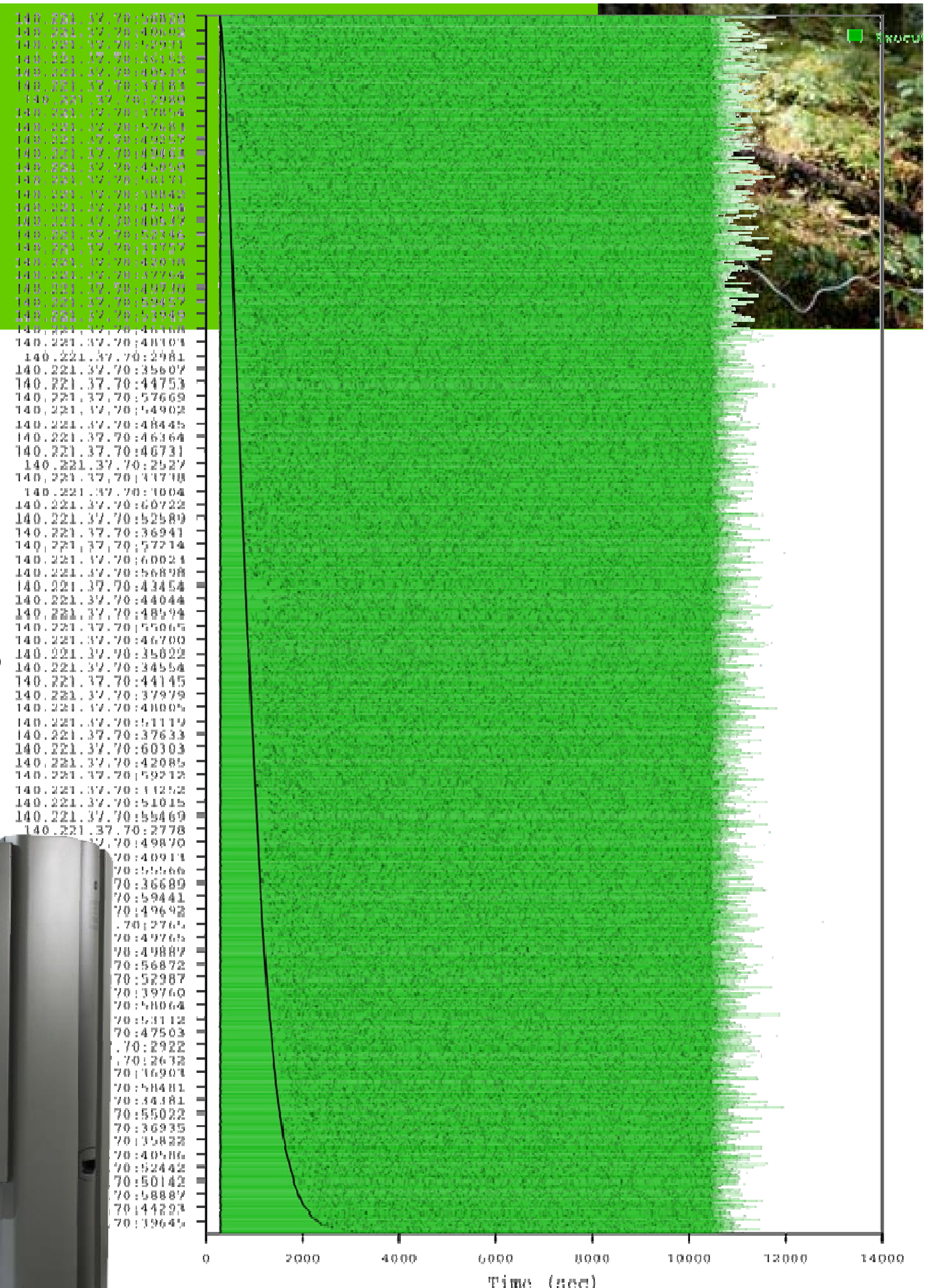
Many Many Tasks: Identifying Potential Drug Targets



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

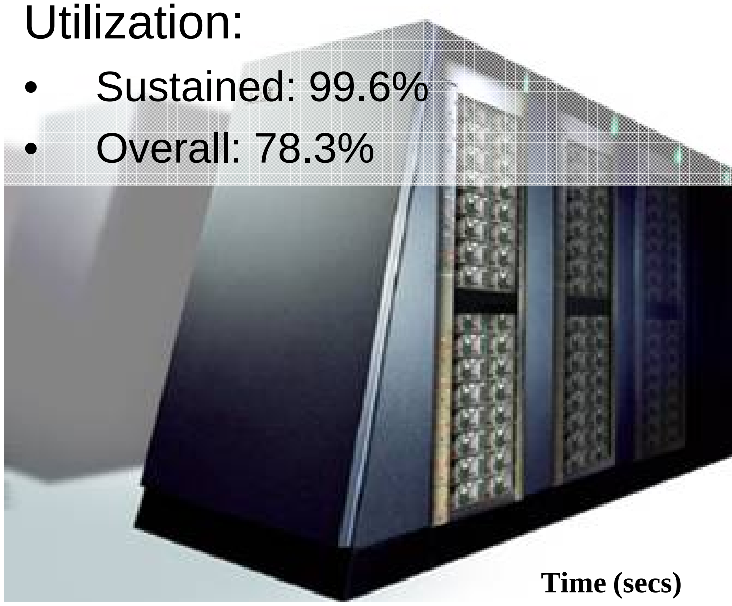
Average task time: 667 sec

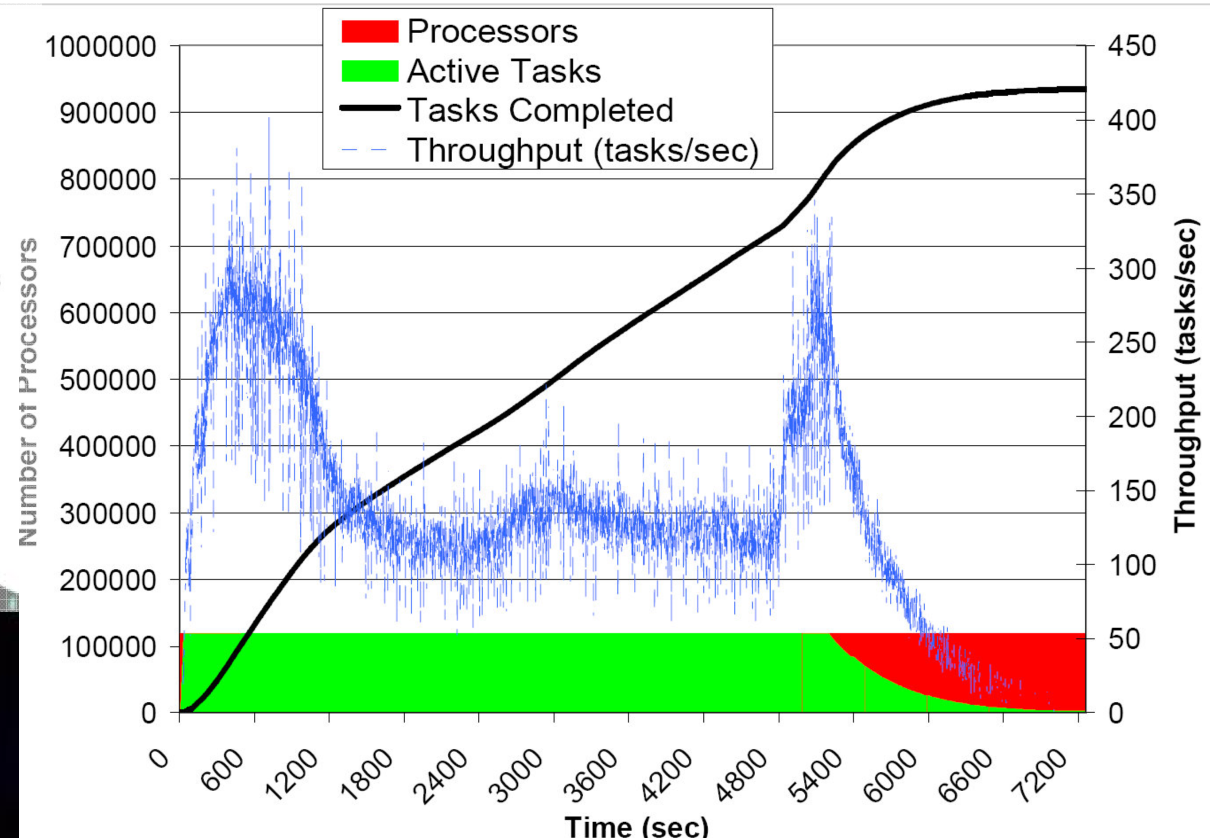
Relative Efficiency: 99.7%

(from 16 to 32 racks)

Utilization:

- Sustained: 99.6%
- Overall: 78.3%





Time (secs)

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

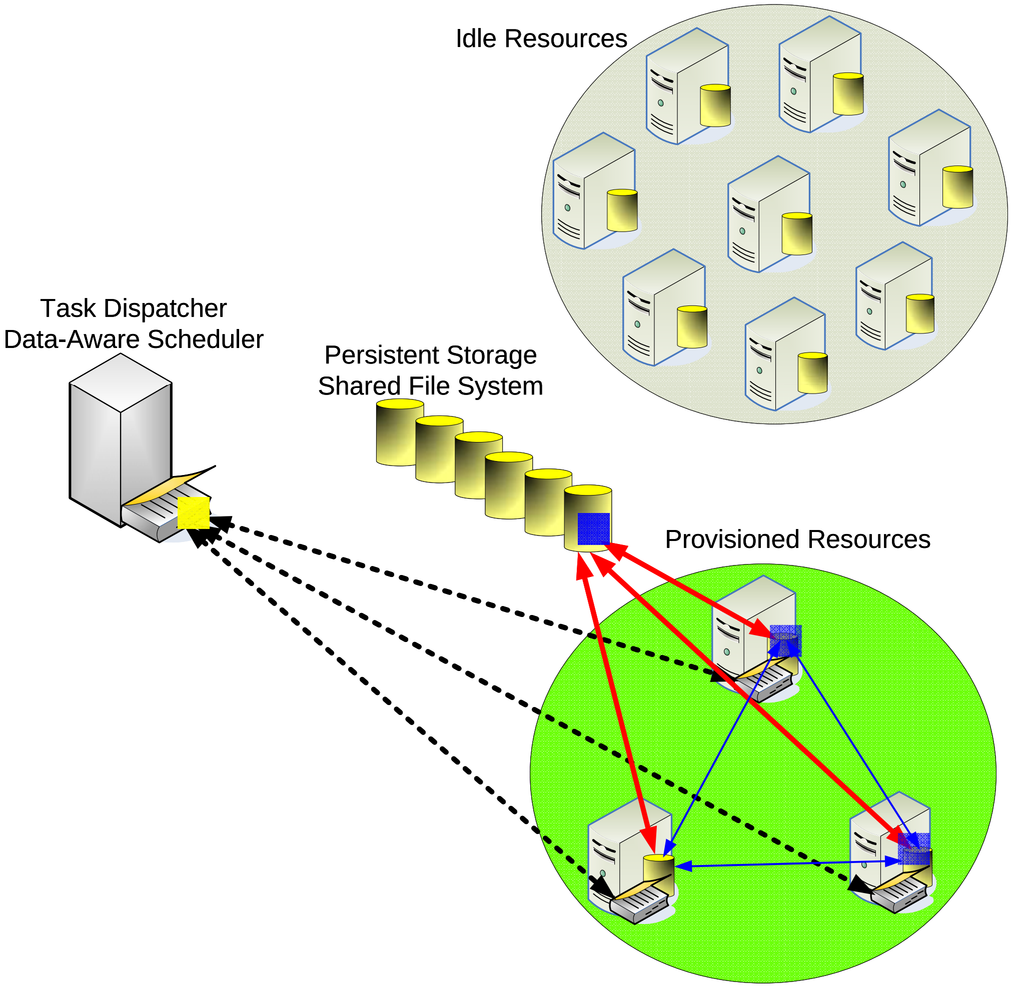
Conclusion: None allows for dynamic resizing of resource pool (independent of application logic) based on system load

Talk Overview

I. Introductions
   – University of Chicago, DSL
   – University of Chicago, CI
   – Argonne National Laboratory, MCS
II. Comparing Grids and Clouds
III. Scalable resource management challenges and solutions
   – Dispatch
   – Provisioning
   – Data Management

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



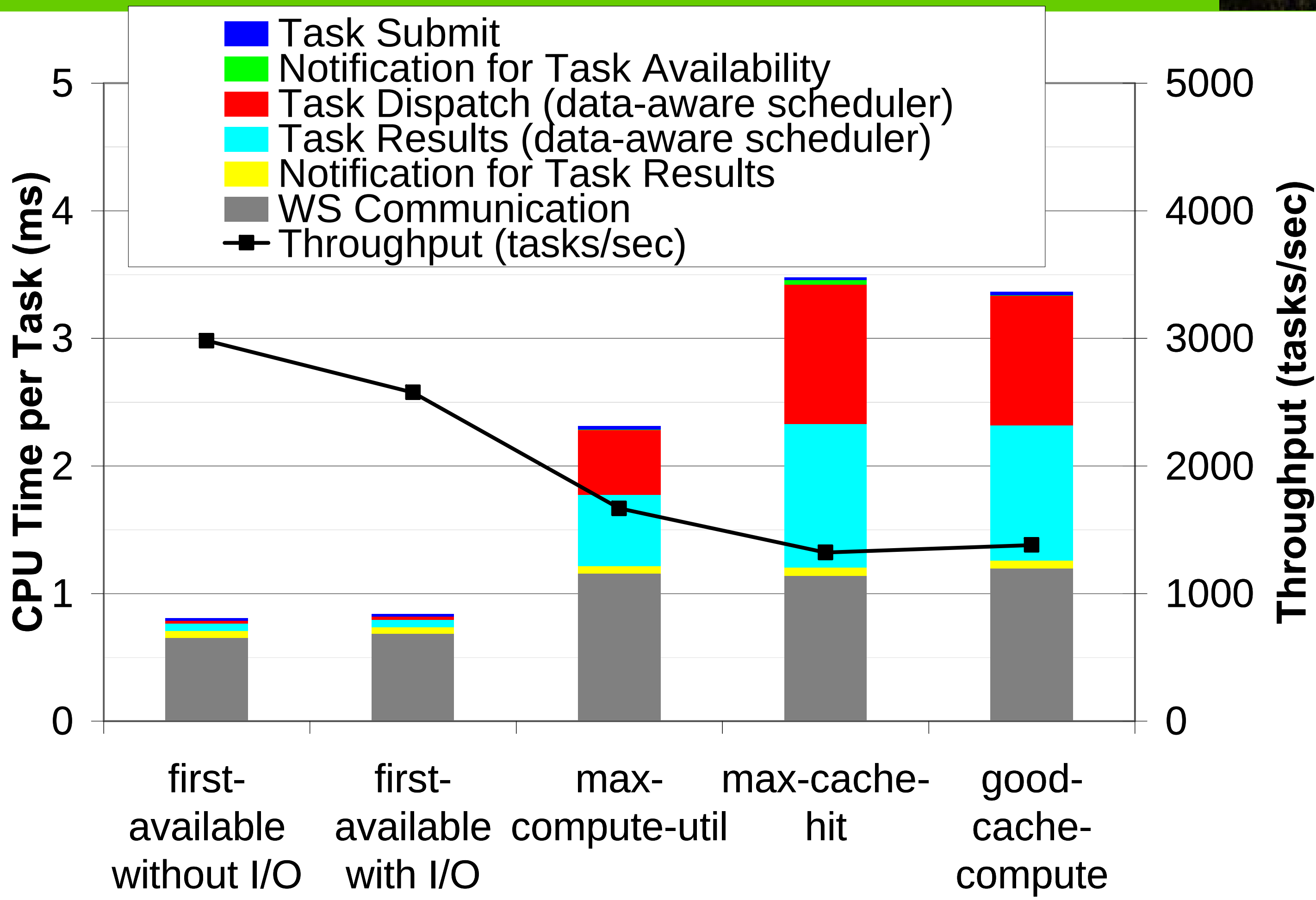
Data Diffusion

- 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

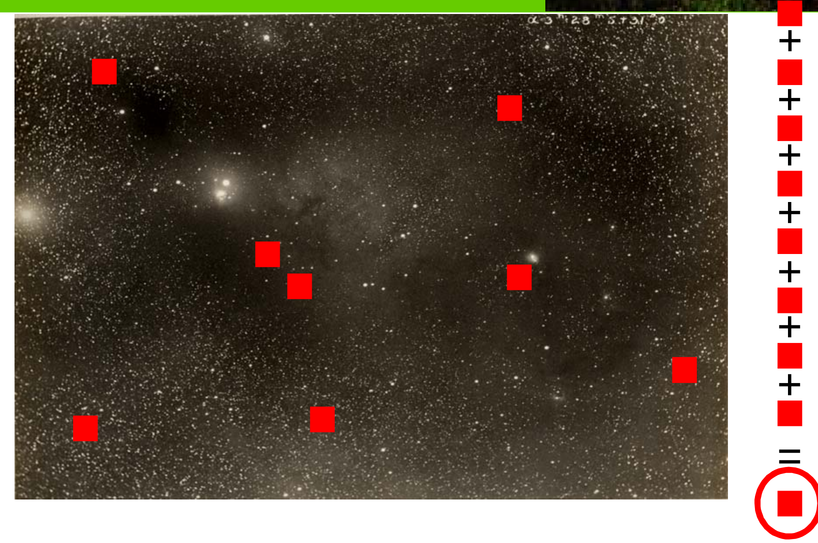
- first-available:
  - simple load balancing
- max-cache-hit
  - maximize cache hits
- max-compute-util
  - maximize processor utilization
- good-cache-compute
  - maximize both cache hit and processor utilization at the same time

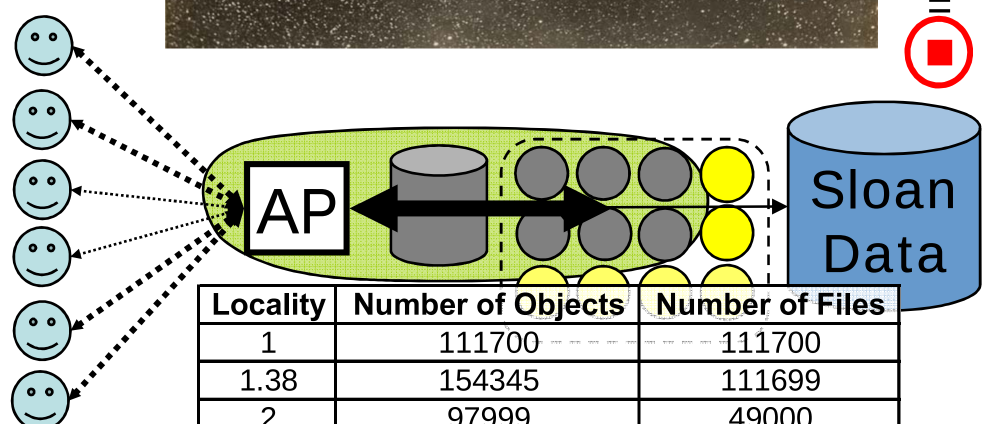
Data-Aware Scheduler Profiling



AstroPortal Stacking Service

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



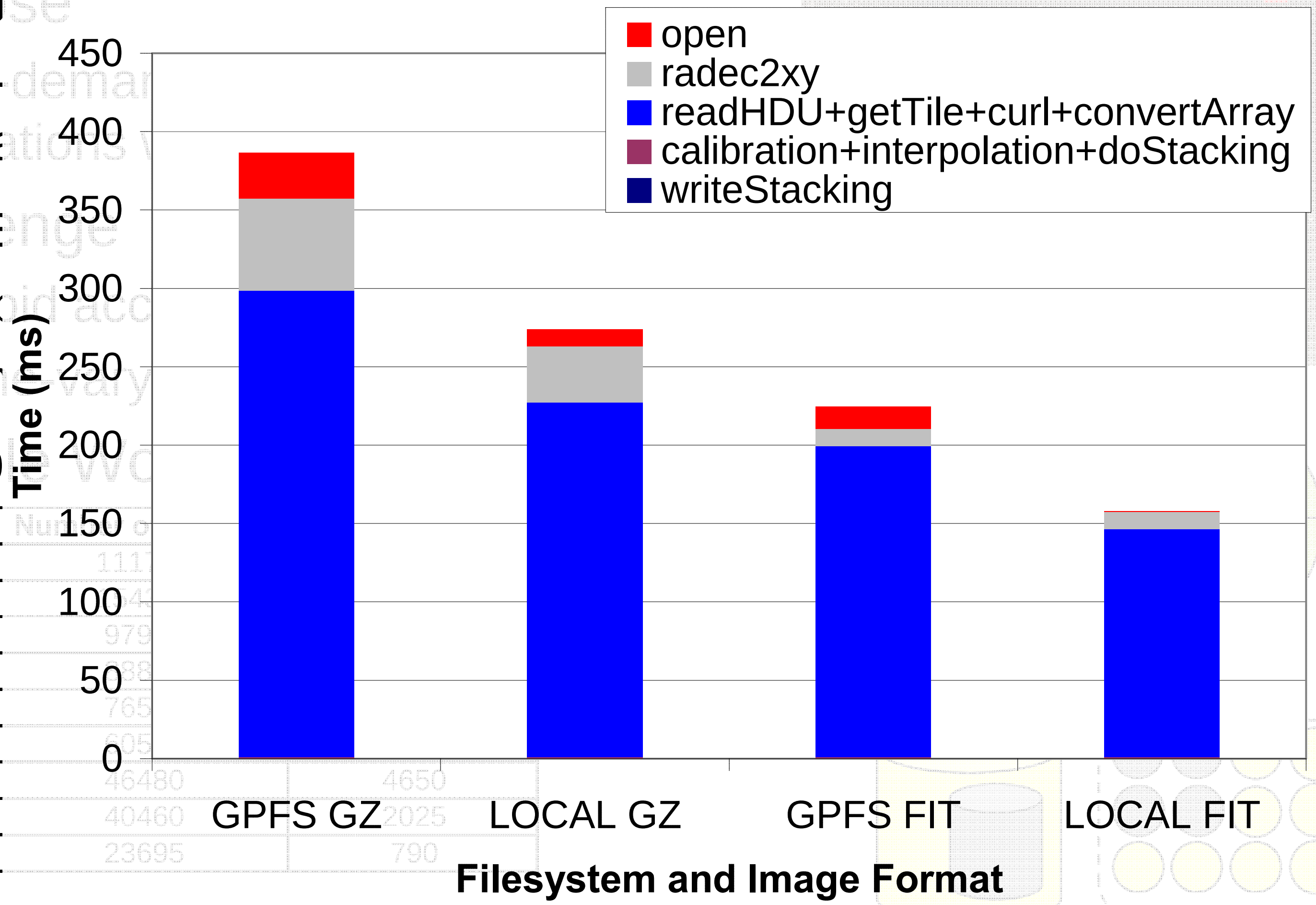


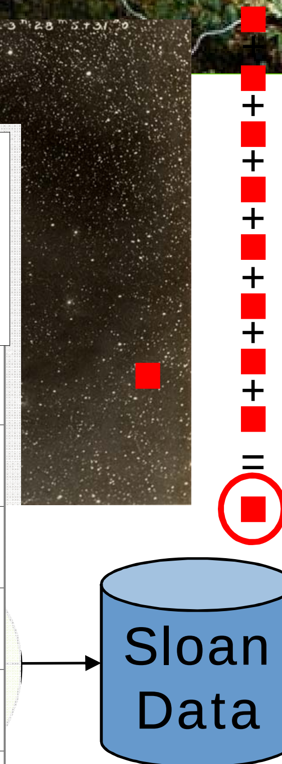
Locality	Number of Objects	Number of Files
1	111700	111700
1.38	154345	111699
2	97999	49000
3	88857	29620
4	76575	19145
5	60590	12120
10	46480	4650
20	40460	2025
30	23695	790

AstroPortal Stacking Service

- Purpose
  - On-
    loca
- Challe
  - Ra
  - Tim
- Samp

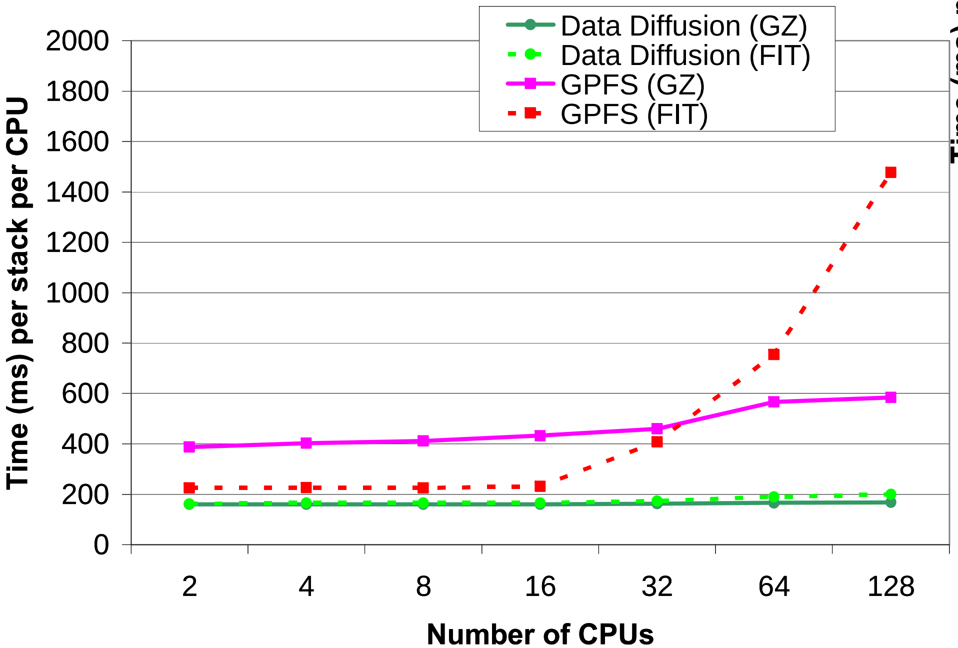
Locality		
1	[illegible]	
1.38	[illegible]	
2	979	
3	[illegible]	
4	765	
5	605	
10	46480	4650
20	40460	2025
30	23695	790

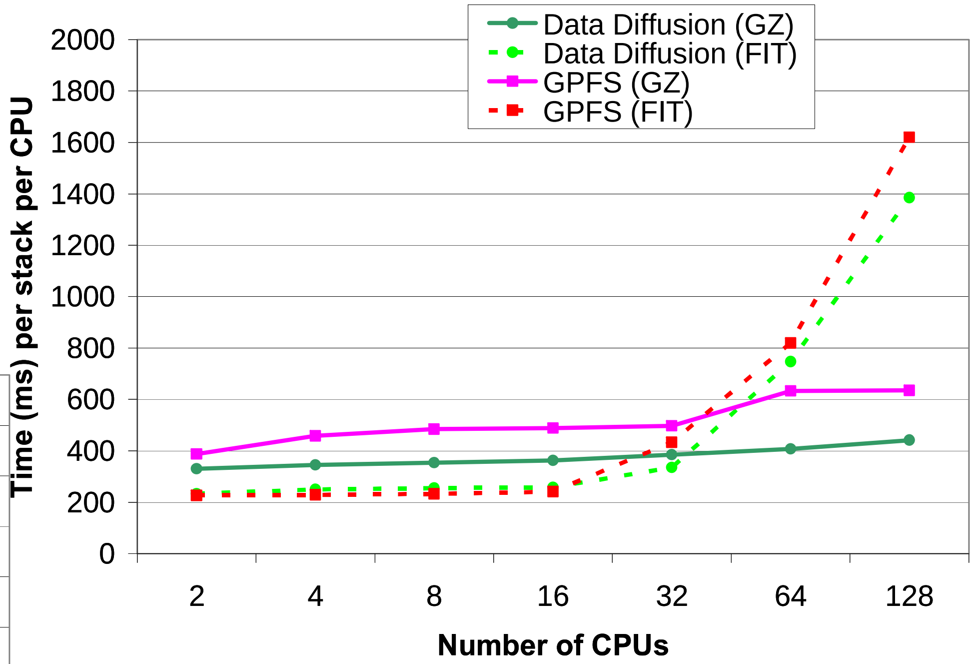




AstroPortal Stacking Service with Data Diffusion

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

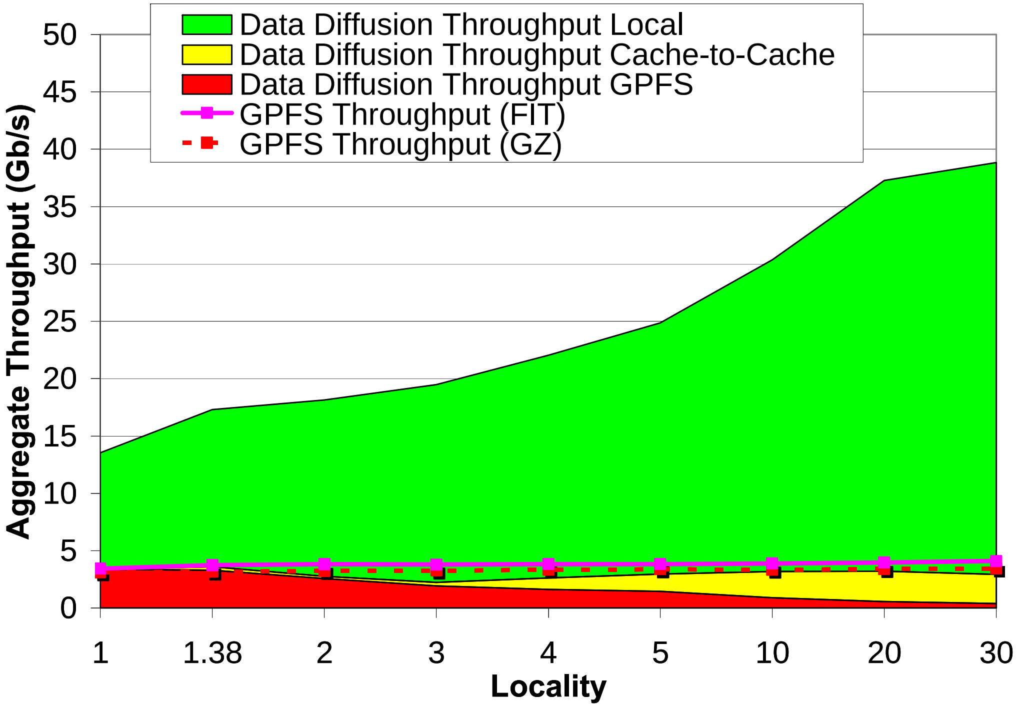


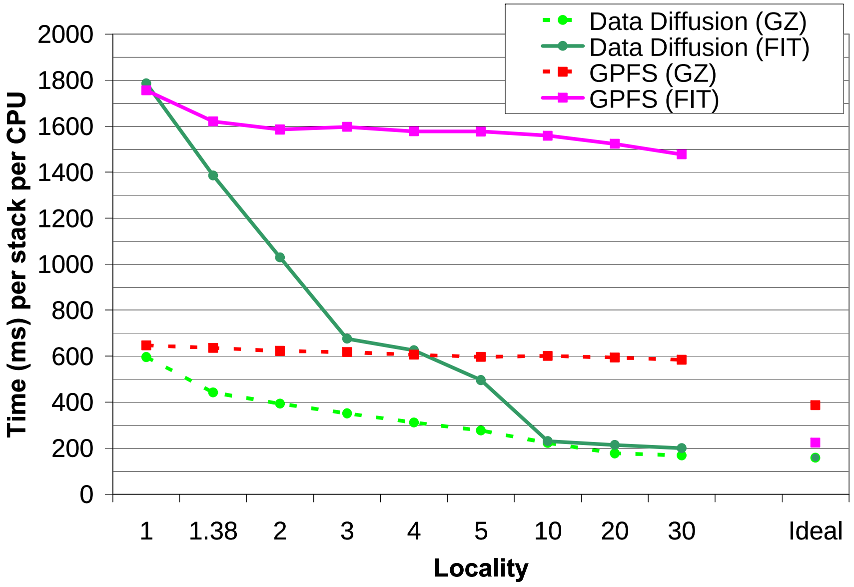


- 🡐High data locality
  - Near perfect scalability

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

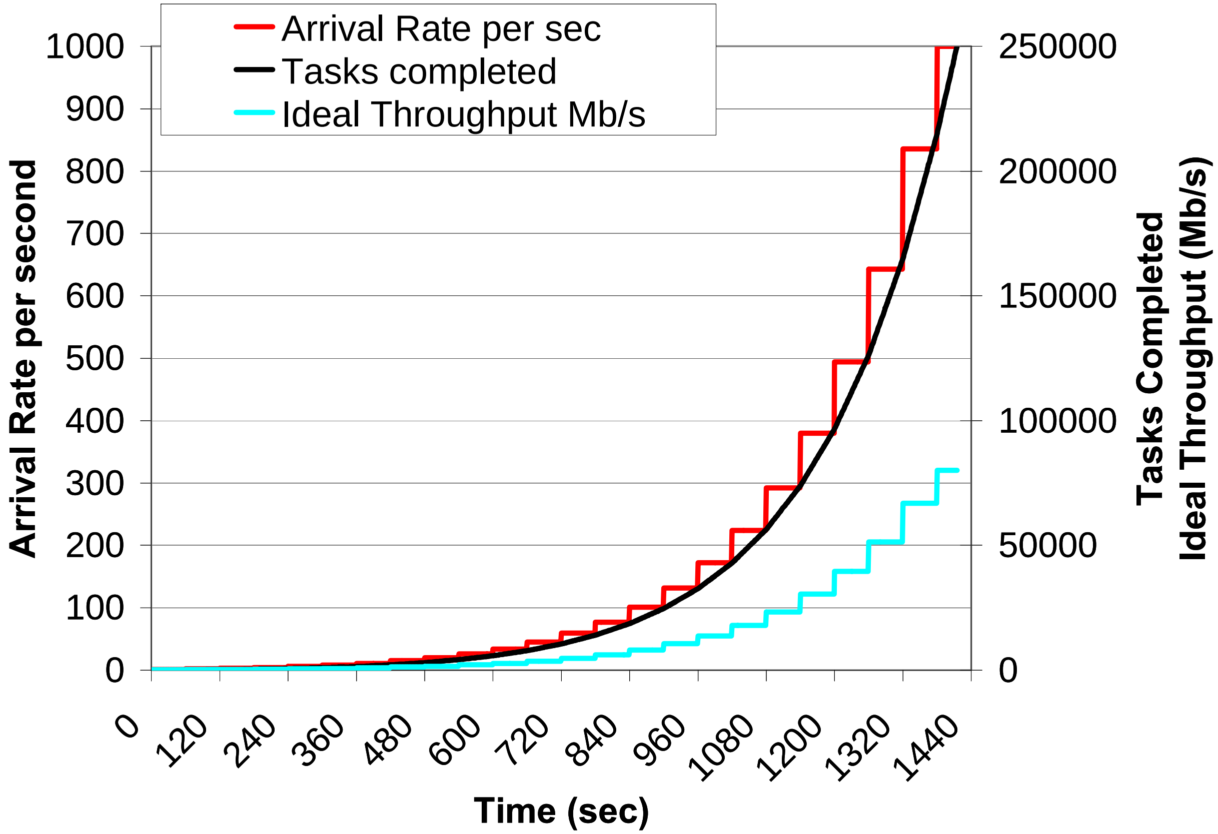




- Big performance gains as locality increases

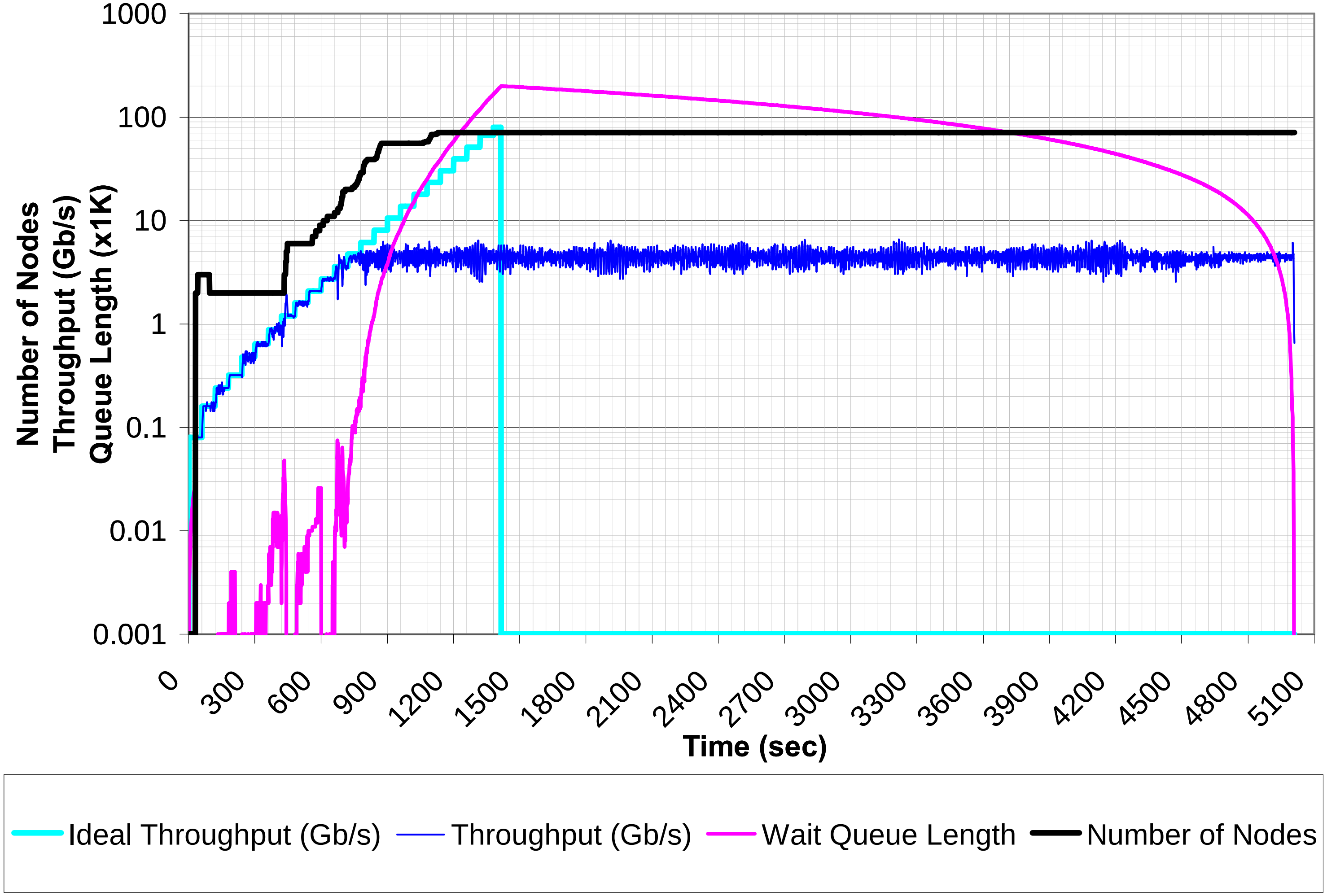
Monotonically Increasing Workload

- 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



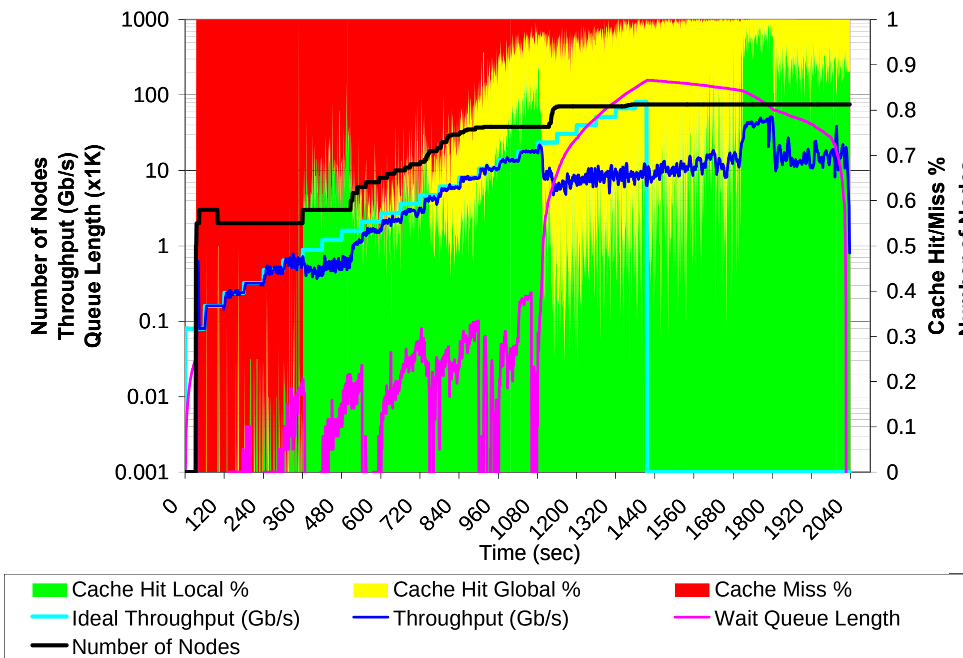
Data Diffusion: First-available (GPFS)

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

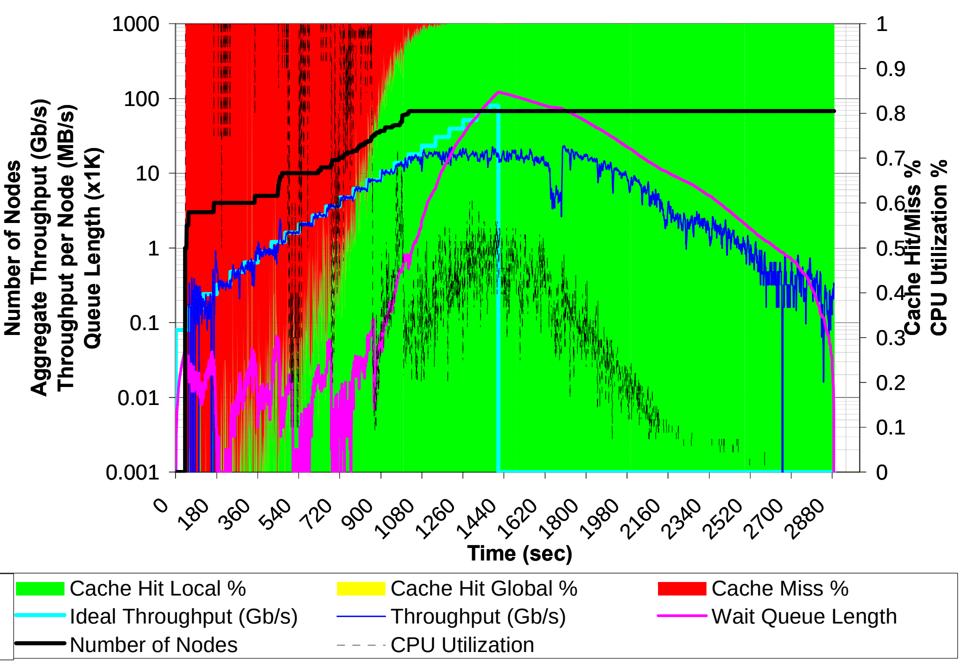


Data Diffusion:
Max-compute-util & max-cache-hit

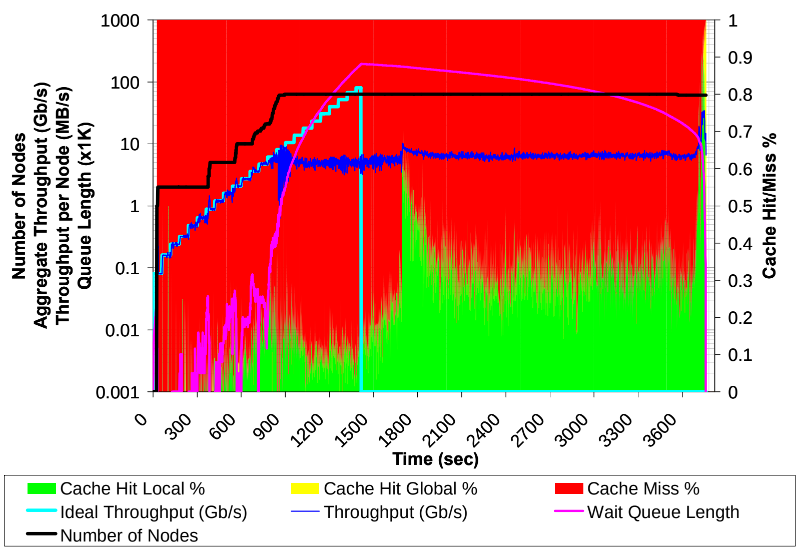
Max-compute-util



Max-cache-hit

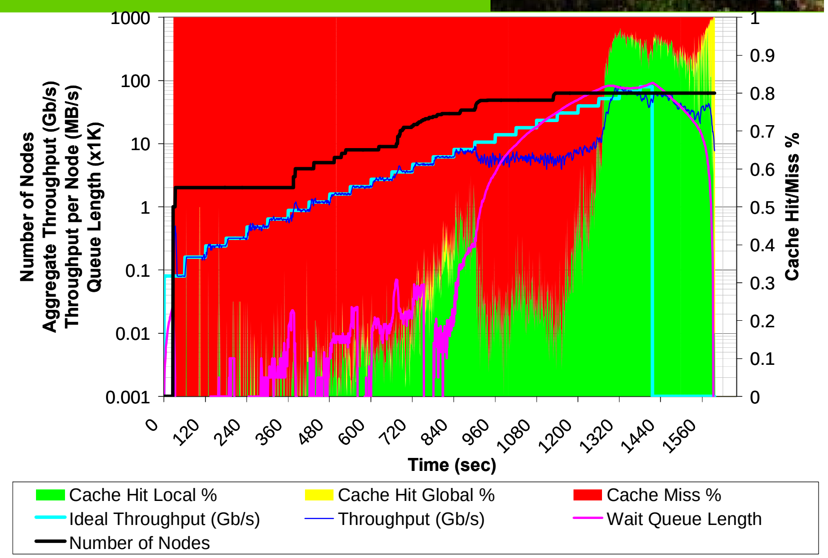


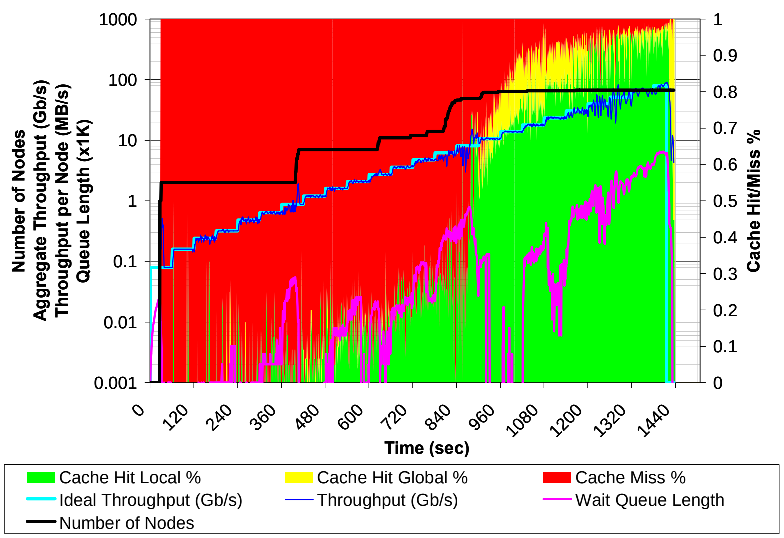
Data Diffusion: Good-cache-compute



← 1GB

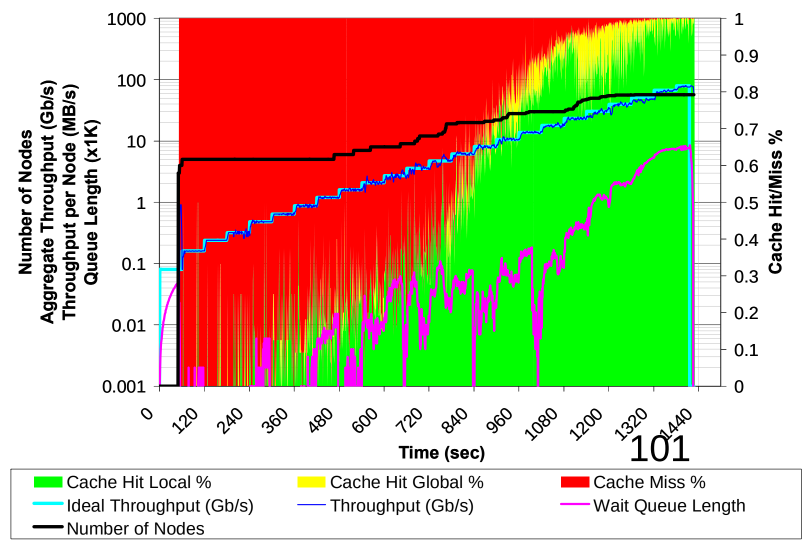
1.5GB →





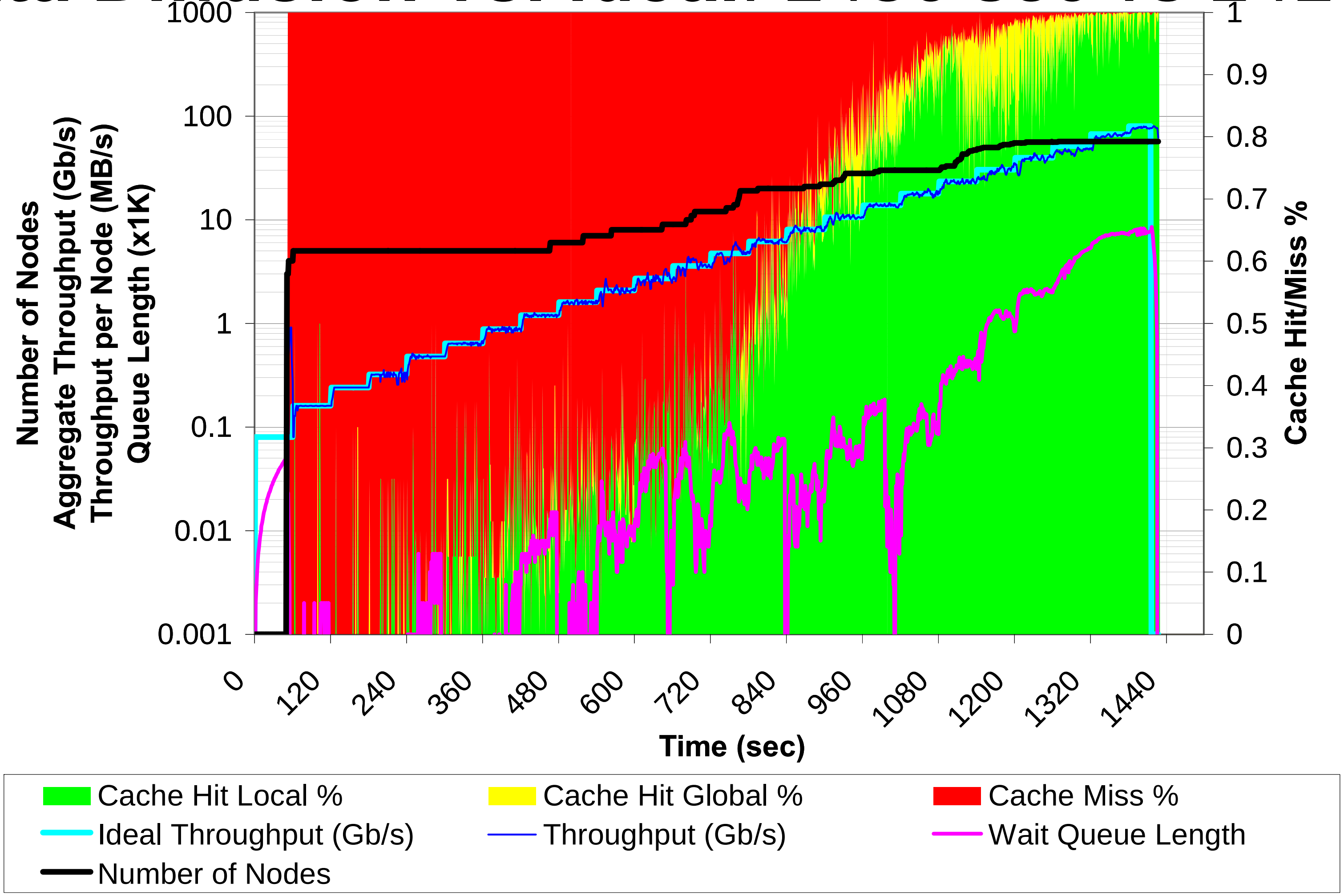
← 2GB

4GB →

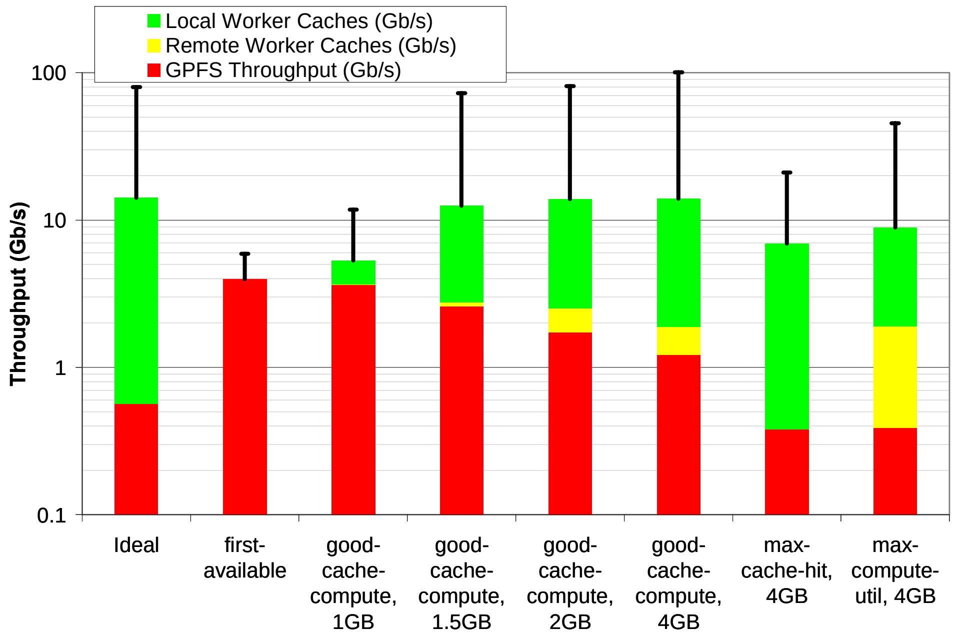


Data Diffusion: Good-cache-compute

- **Data Diffusion vs. ideal**: 1436 sec vs 1415 sec



Data Diffusion: Throughput and Response Time



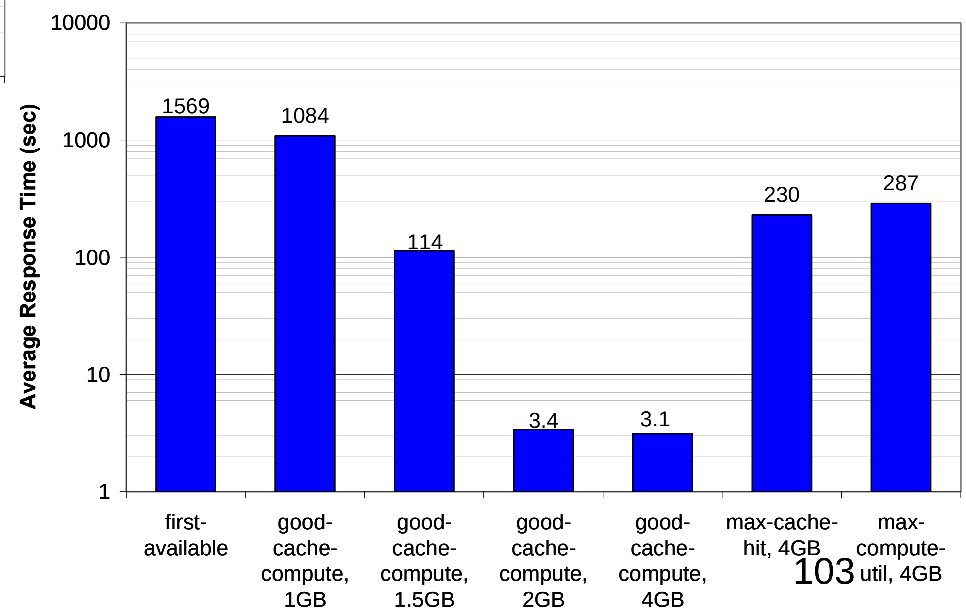
🡐 Throughput:

– Average: 14Gb/s vs 4Gb/s

– Peak: 100Gb/s vs. 6Gb/s

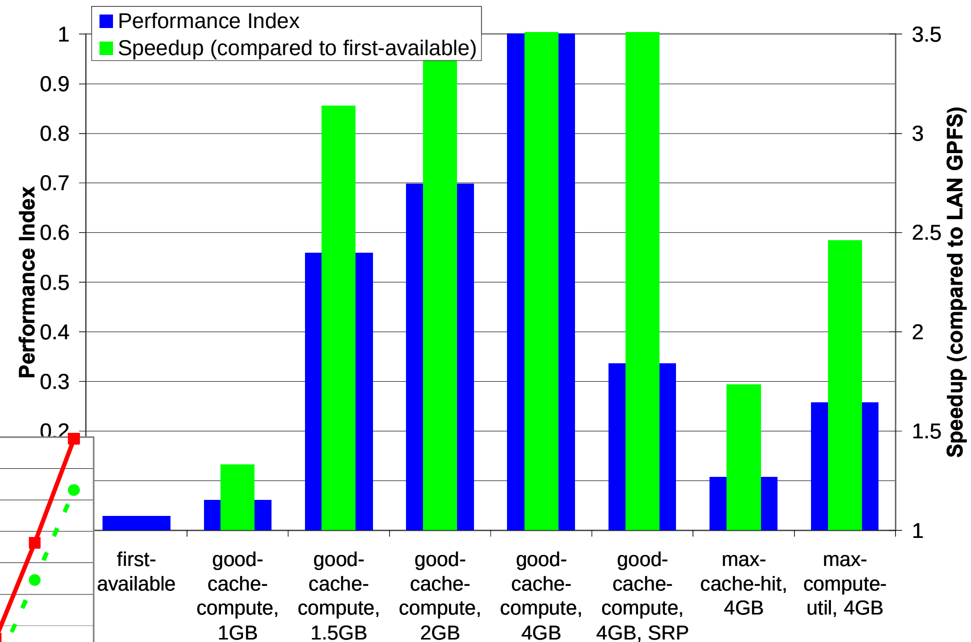
Response Time ➔

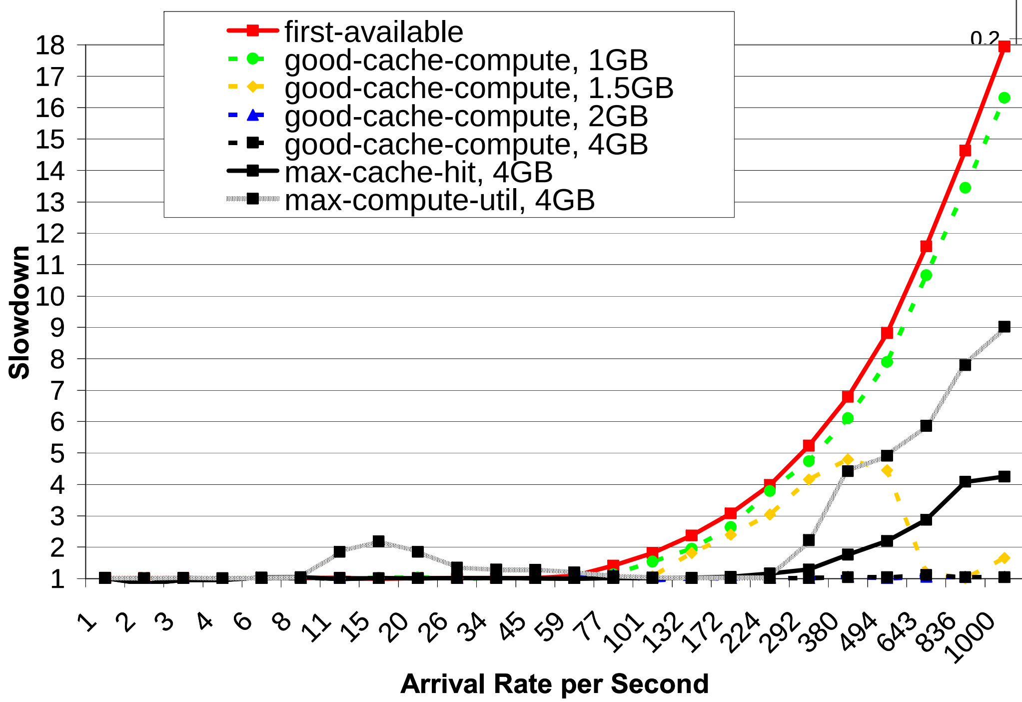
– 3 sec vs 1569 sec ➔ 506X



Data Diffusion: Performance Index, Slowdown, and Speedup

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



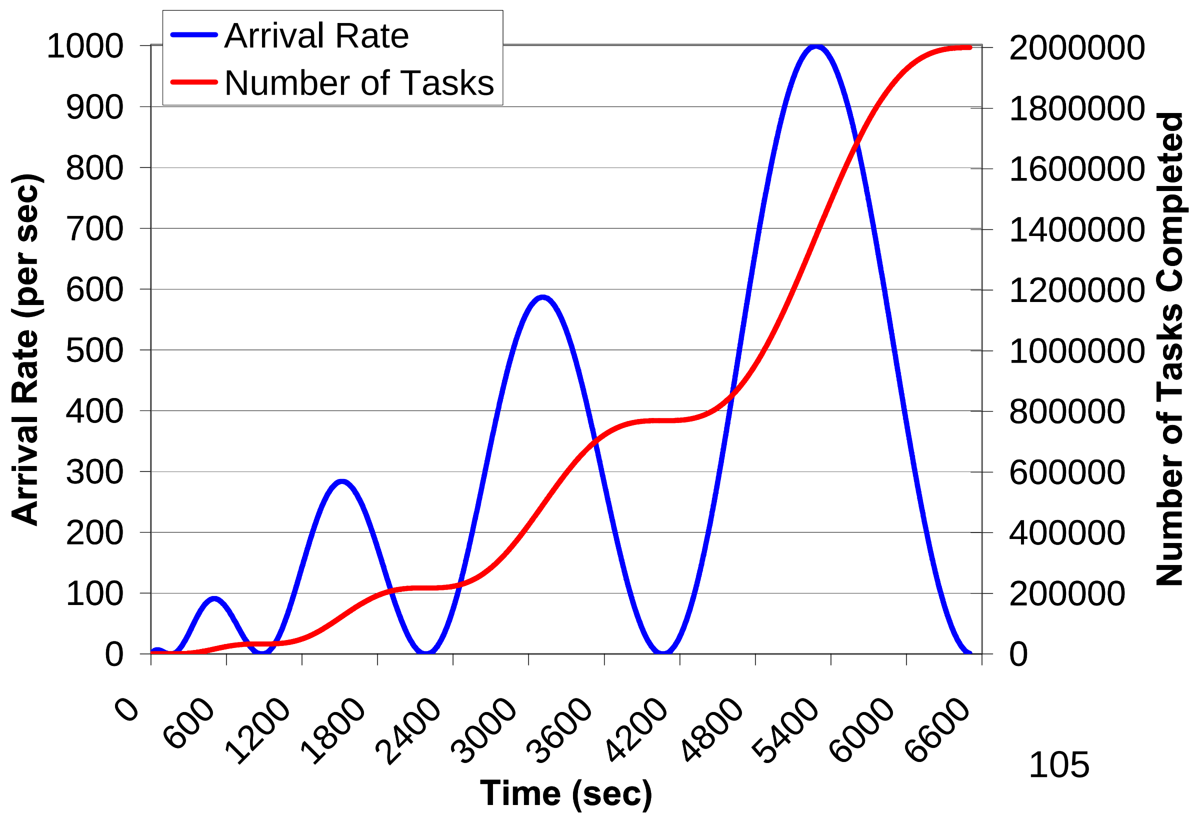


- Slowdown:
  - 18X slowdown for GPFS
  - Near ideal 1X slowdown for large enough caches

Sin-Wave Workload

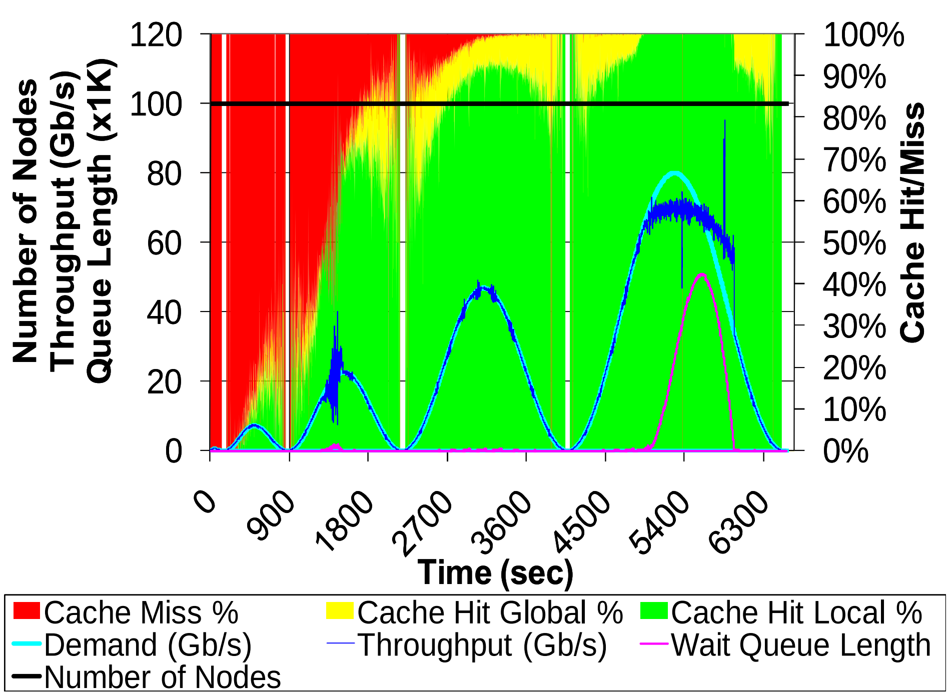
- 2M tasks
  - 10MB reads
  - 10ms compute
- Vary arrival rate:
  - Min: 1 task/sec
  - Arrival rate function:
  - Max: 1000 tasks/sec
- 200 processors
- Ideal case:
  - 6505 sec
  - 80Gb/s peak throughput

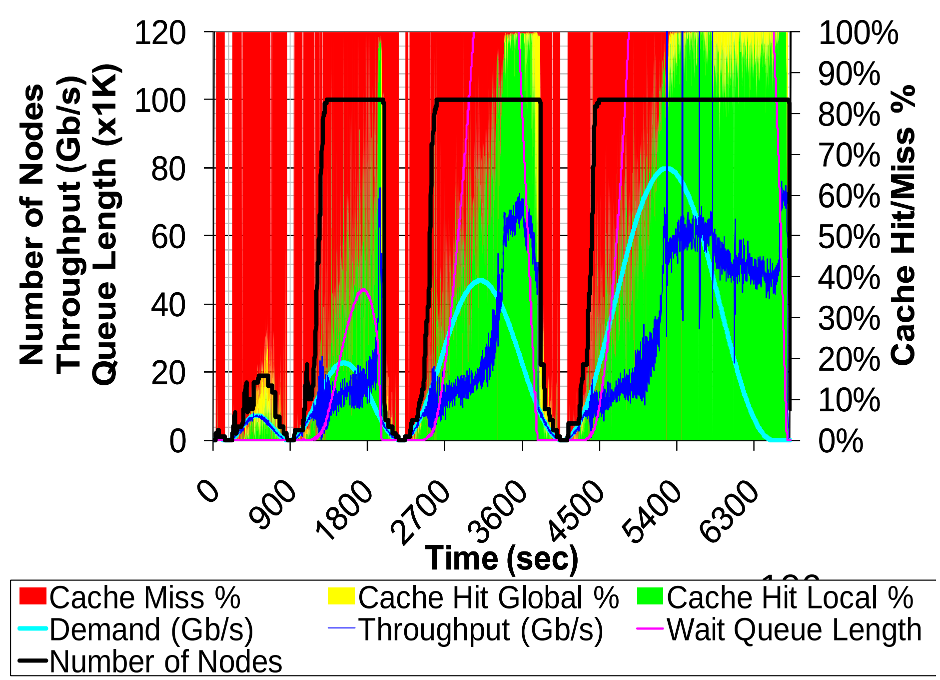
$$A = \left\lfloor (\sin(sqrt(time + 0.11) * 2.859678) + 1) * (time + 0.11) * 5.705 \right\rfloor$$



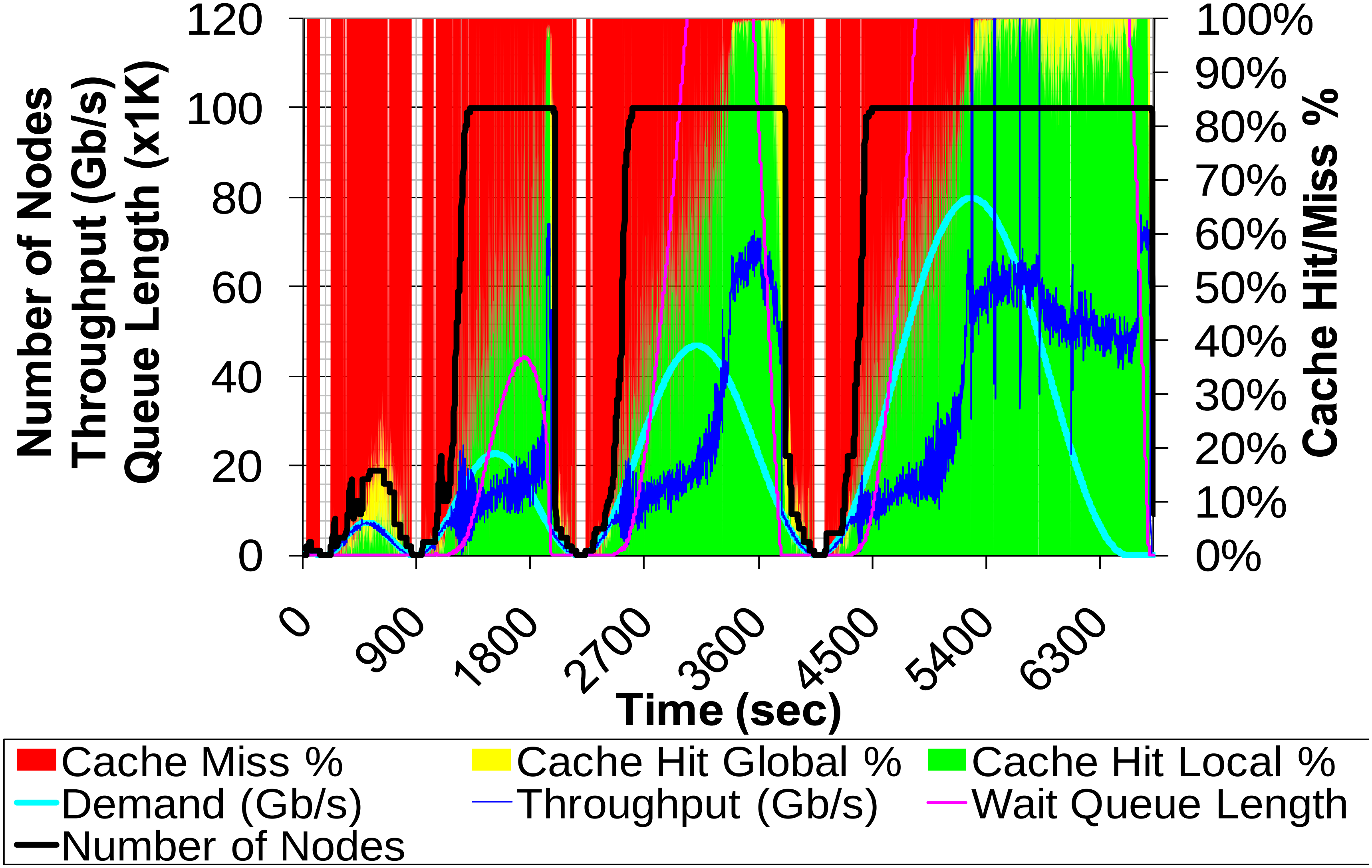
Sin-Wave Workload

- GPFS ➔ 5.7 hrs, ~8Gb/s, 1138 CPU hrs
- DF+SRP ➔ 1.8 hrs, ~25Gb/s, 361 CPU hrs
- DF+DRP ➔ 1.86 hrs, ~24Gb/s, 253 CPU hrs





Sin-Wave Workload



All-Pairs Workload

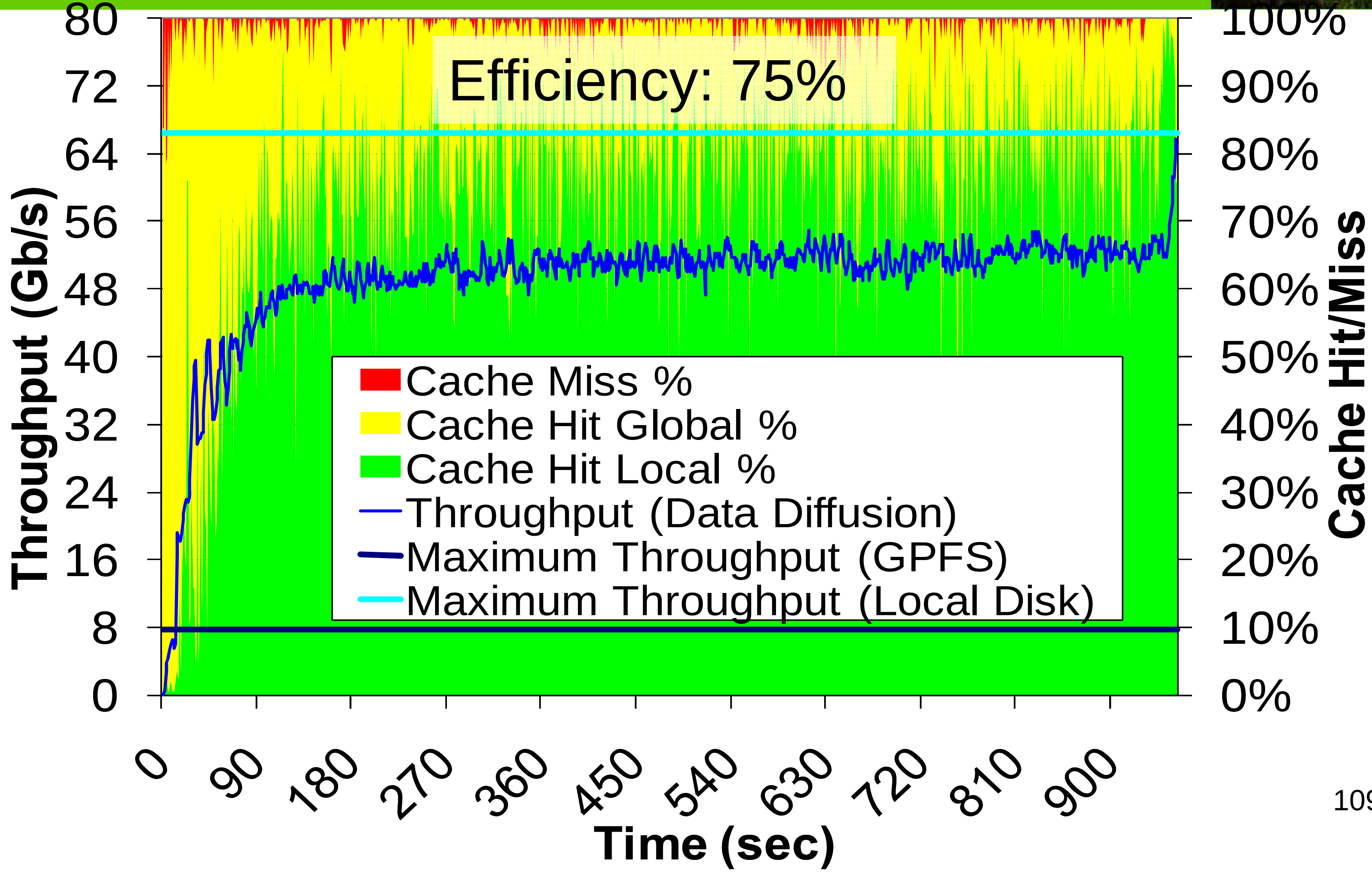
- 500x500
  - 250K tasks
  - 24MB reads
  - 100ms compute
  - 200 CPUs
- 1000x1000
  - 1M tasks
  - 24MB reads
  - 4sec compute
  - 4096 CPUs
- Ideal case:
  - 6505 sec
  - 80Gb/s peak throughput

- All-Pairs(set A, set B, function F) returns matrix M:
- Compare all elements of set A to all elements of set B via function F, yielding matrix M, such that

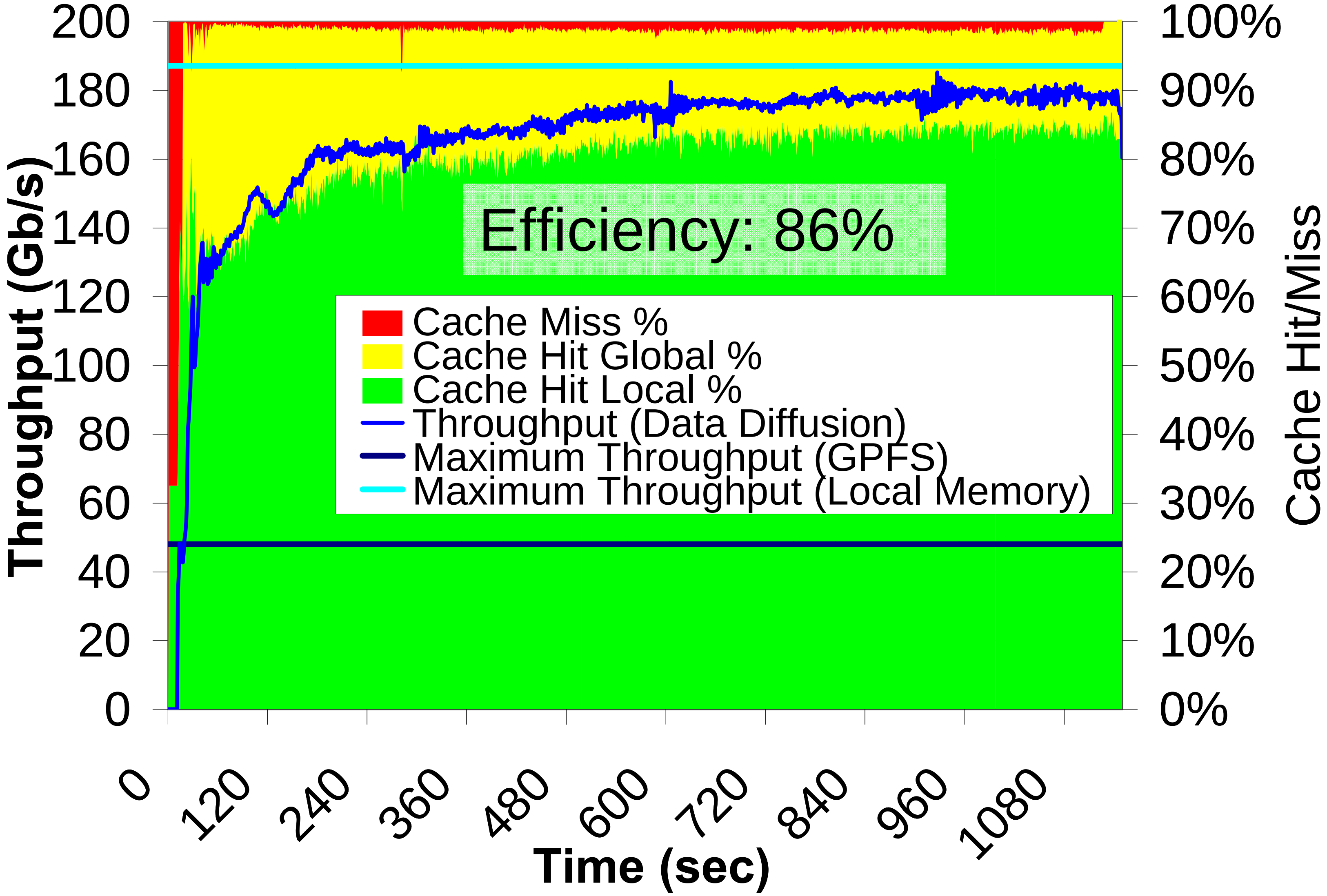
$$M[i,j] = F(A[i],B[j])$$

```
foreach $i in A
    foreach $j in B
        submit_job F $i $j
    end
end
```

All-Pairs Workload 500x500 on 200 CPUs

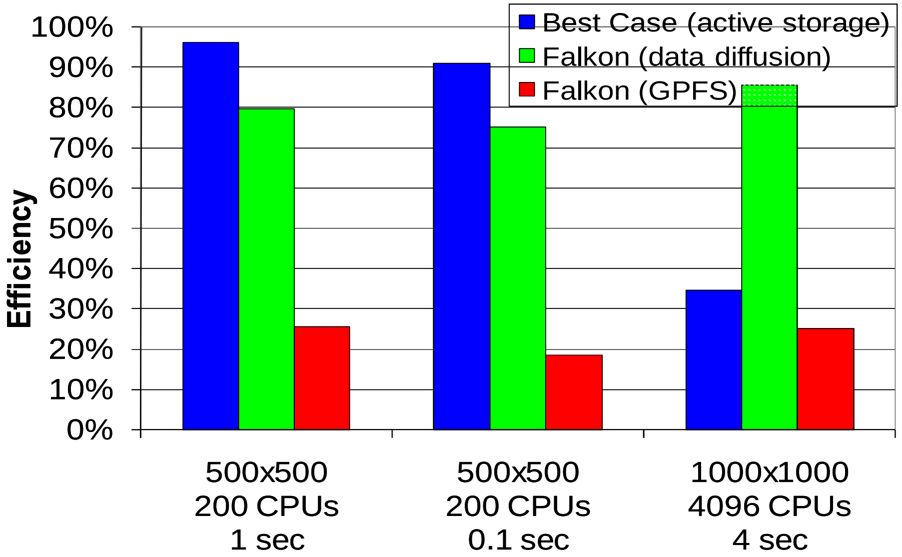


All-Pairs Workload
1000x1000 on 4K emulated CPUs



All-Pairs Workload
Data Diffusion vs. Active Storage

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



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

All-Pairs Workload
Data Diffusion vs. Active Storage

- Best to use active storage if
  - Slow data source
  - Workload working set fits on local node storage
  - Good aggregate network bandwidth
- Best to use data diffusion if
  - Medium to fast data source
  - Task working set << workload working set
  - Task working set fits on local node storage
  - Good aggregate network bandwidth
- If task working set does not fit on local node storage
  - Use parallel file system (i.e. GPFS, Lustre, PVFS, etc)

Limitations of Data Diffusion

- Needs Java 1.4+
- Needs IP connectivity between hosts
- Needs local storage (disk, memory, etc)
- Per task workings set must fit in local storage
- Task definition must include input/output files metadata
- Data access patterns: write once, read many

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 Many-Task Computing Paradigm
- Addressed real challenges in resource management in large scale distributed systems
  - Slow dispatch rates
  - Long wait queue times
  - Poor scaling of parallel file systems
- Show effectiveness of streamlined task dispatching and dynamic resource provisioning:
  - Astronomy, medicine, chemistry, molecular dynamics, economic modelling, and data mining
- Show effectiveness of data diffusion:
  - Real large-scale astronomy application and a variety of synthetic workloads

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
- **National Science Foundation**
  - 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)

- IEEE International Workshop on Scientific Workflows (SWF), 2009
- 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)