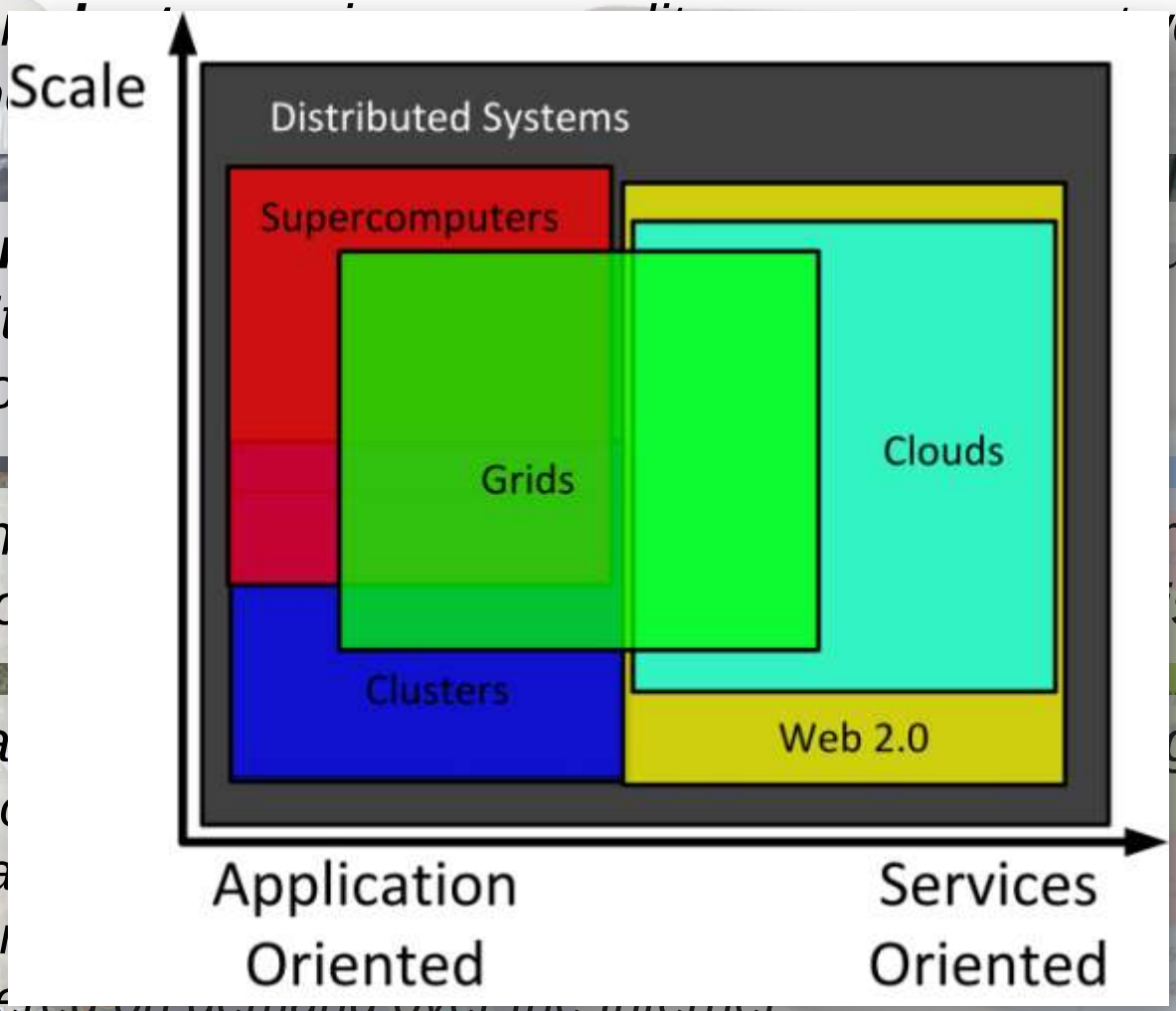
Local Resource Management

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

EECS 395 / EECS 495
Hot Topics in Distributed Systems: Data-Intensive Computing
January 21st, 2010

Clusters, Grids, Clouds, and Supercomputers

- *Computer* *work interconn*
- ***Supercom*** *using commodit* *interconnects and custo*
- ***Grids*** *ten* *e typically loosely co* *spersed*
- ***Clouds*** *a* *gm driven by:*
  1. *econ*
  2. *virtua*
  3. *dyna*
  4. *delive*

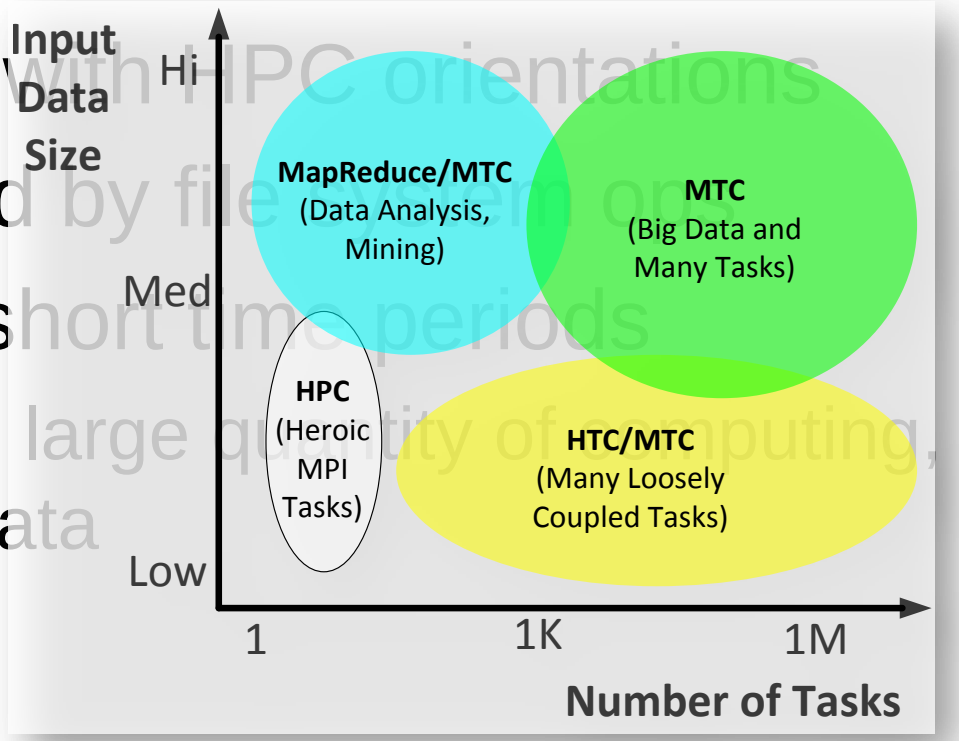


High-Throughput Computing & High-Performance Computing

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

MTC: Many-Task Computing

- Bridge the gap between HPC and HTC
- Applied in clusters, grids, and supercomputers
- Loosely coupled apps with HPC orientations
- Many activities coupled by file system ops
- Many resources over short time periods
  - Large number of tasks, large quantity of computing, and large volumes of data



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

Local Resource Manager (LRM) Features

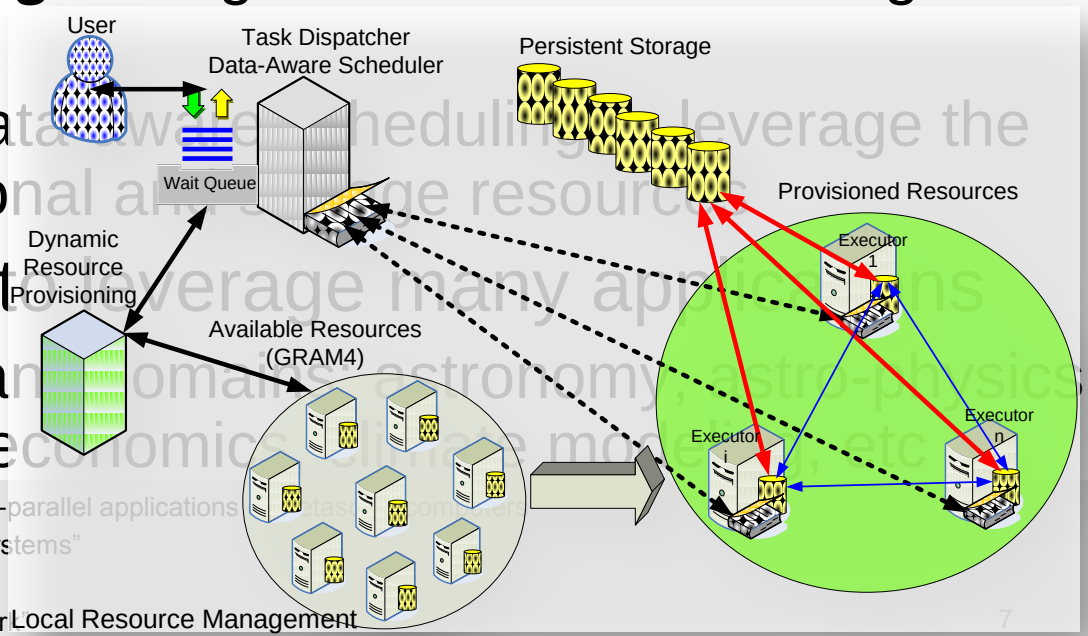
- Resource provisioning
- Job scheduling
  - FIFO
  - Priority support
  - Multiple queues
  - Back-filling
  - Advanced Reservations
  - Accounting

LRM: Local Resource Managers

- SGE
  - HTC, HPC, Sun Grid Engine
- PBS
  - HTC, HPC, originated from NASA
- LSF
  - HTC, HPC, IBM
- Cobalt
  - HPC, BlueGene
- Condor
  - HTC, HPC, open source, free
- Falkon
  - MTC, HTC, open source, free, part of my dissertation
- GRAM
  - An abstraction for other LRMs

Falkon

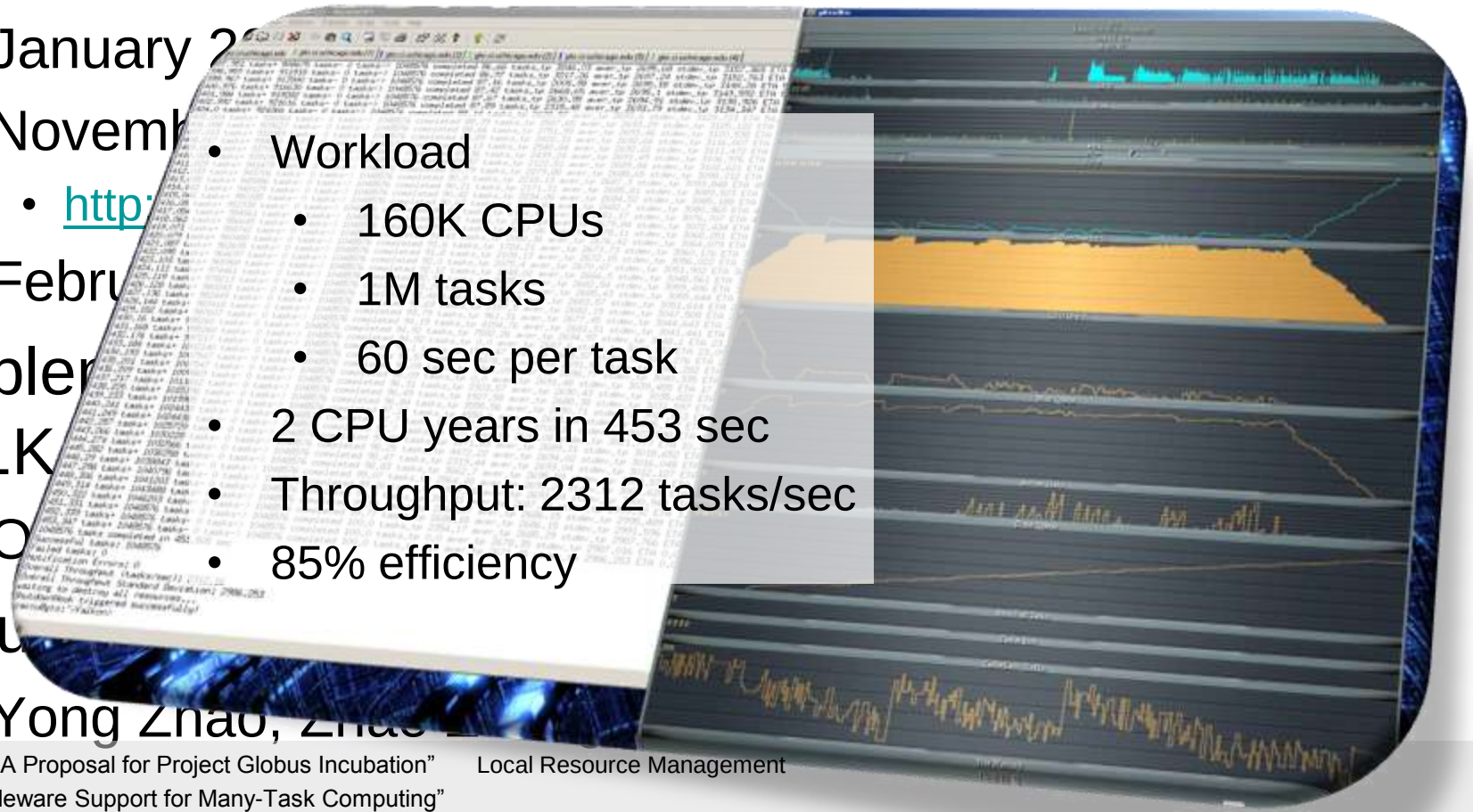
- ***Goal:*** enable the ***rapid and efficient*** execution of many independent jobs on large compute clusters
- Combines three components:
  - a ***streamlined task dispatcher***
  - ***resource provisioning*** through multi-level scheduling techniques
  - ***data diffusion*** and data aware scheduling to leverage the co-located computational and storage resources
- Integration into Swift to leverage many applications
  - Applications cover many domains: astronomy, astro-physics, medicine, chemistry, economic models, etc



[SciDAC09] "Extreme-scale scripting: Opportunities for large task-parallel applications on petascale computers"
[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"
[Globus07] "Falkon: A Proposal for Project Globus Incubation"
[SC07] "Falkon: a Fast and Light-weight tasK executiON framework"
[SWF07] "Swift: Fast, Reliable, Loosely Coupled Parallel Computation"

Falkon Project

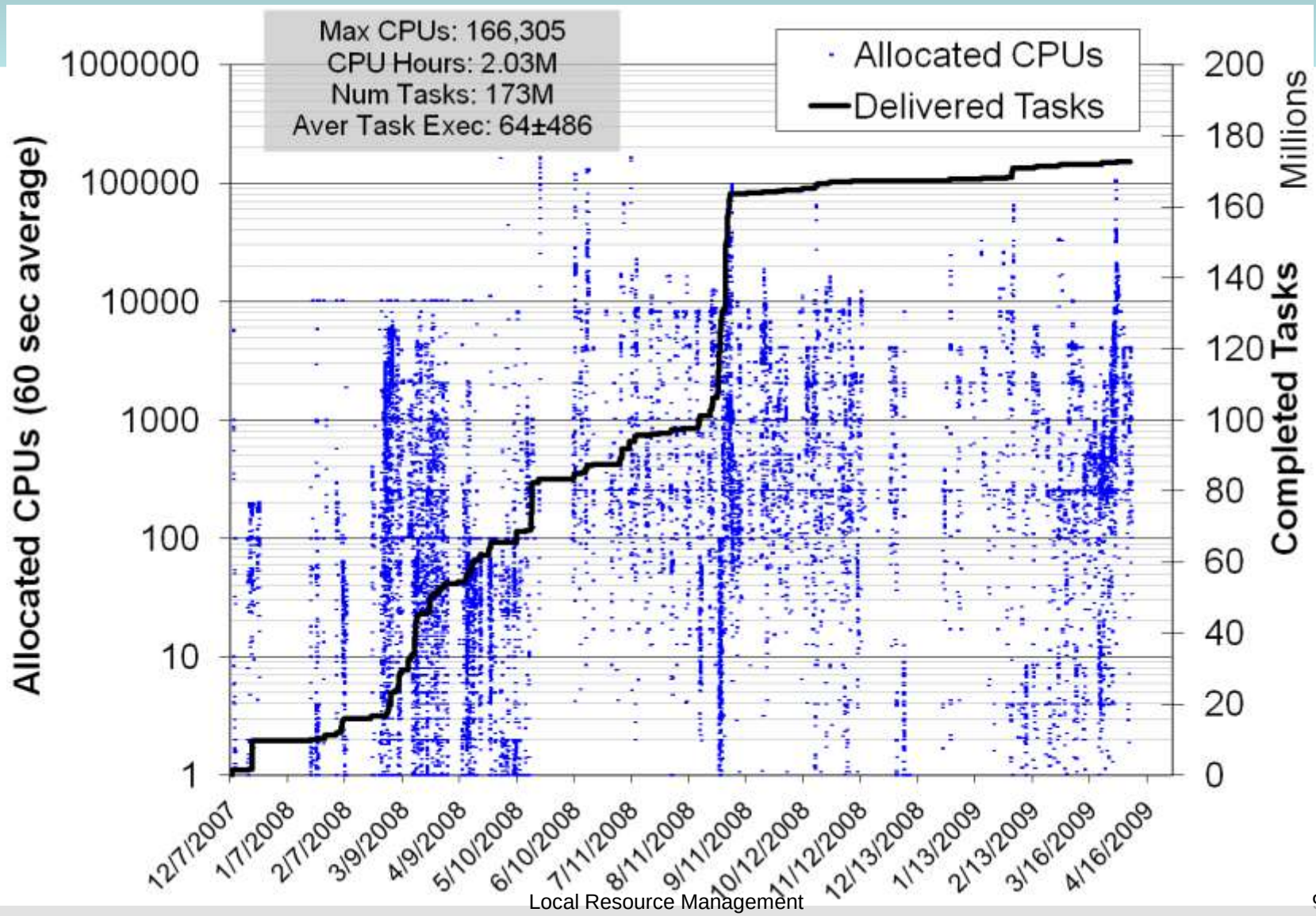
- Falkon is a real system
  - Late 2005: Initial prototype, AstroPortal
  - January 2
  - Novemb
    - http:
  - Febru
- Implem
(~1K
  - O
- Sou
  - Yong Zhao, Zhao



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

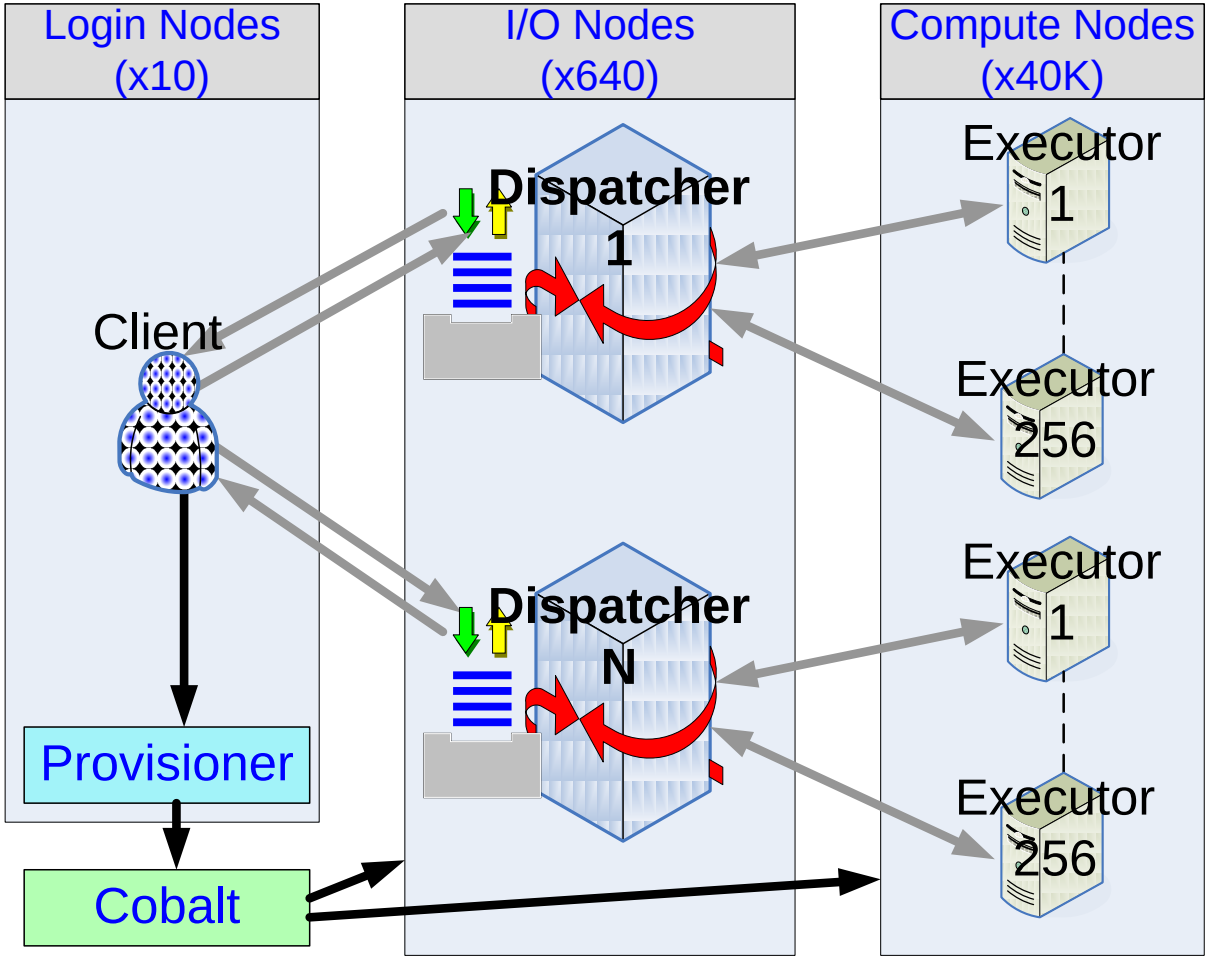
Local Resource Management

Falkon Activity History (16 months)



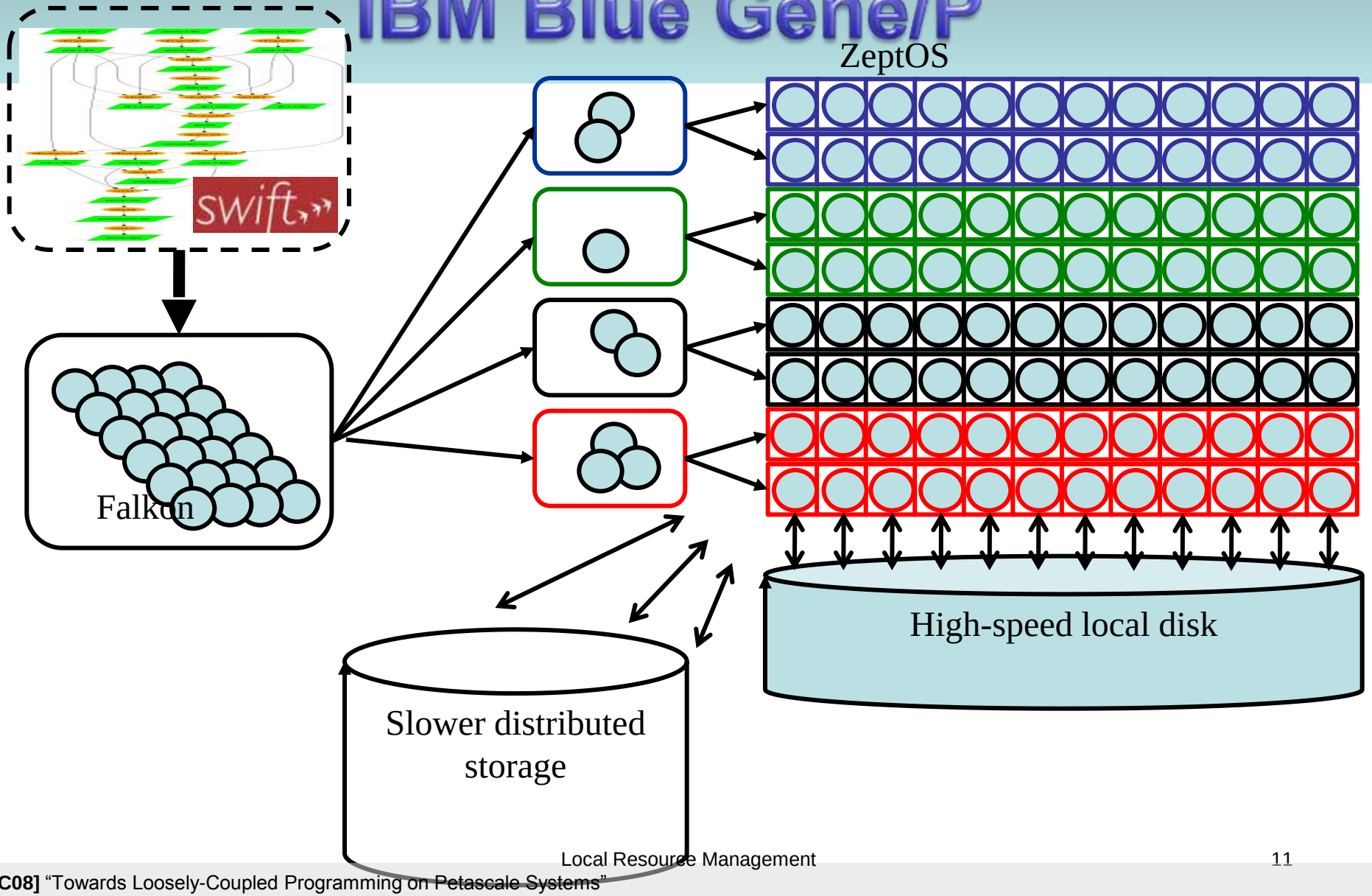
[TPDS09] "Middleware Support for Many-Task Computing", under preparation

Distributed Falkon Architecture



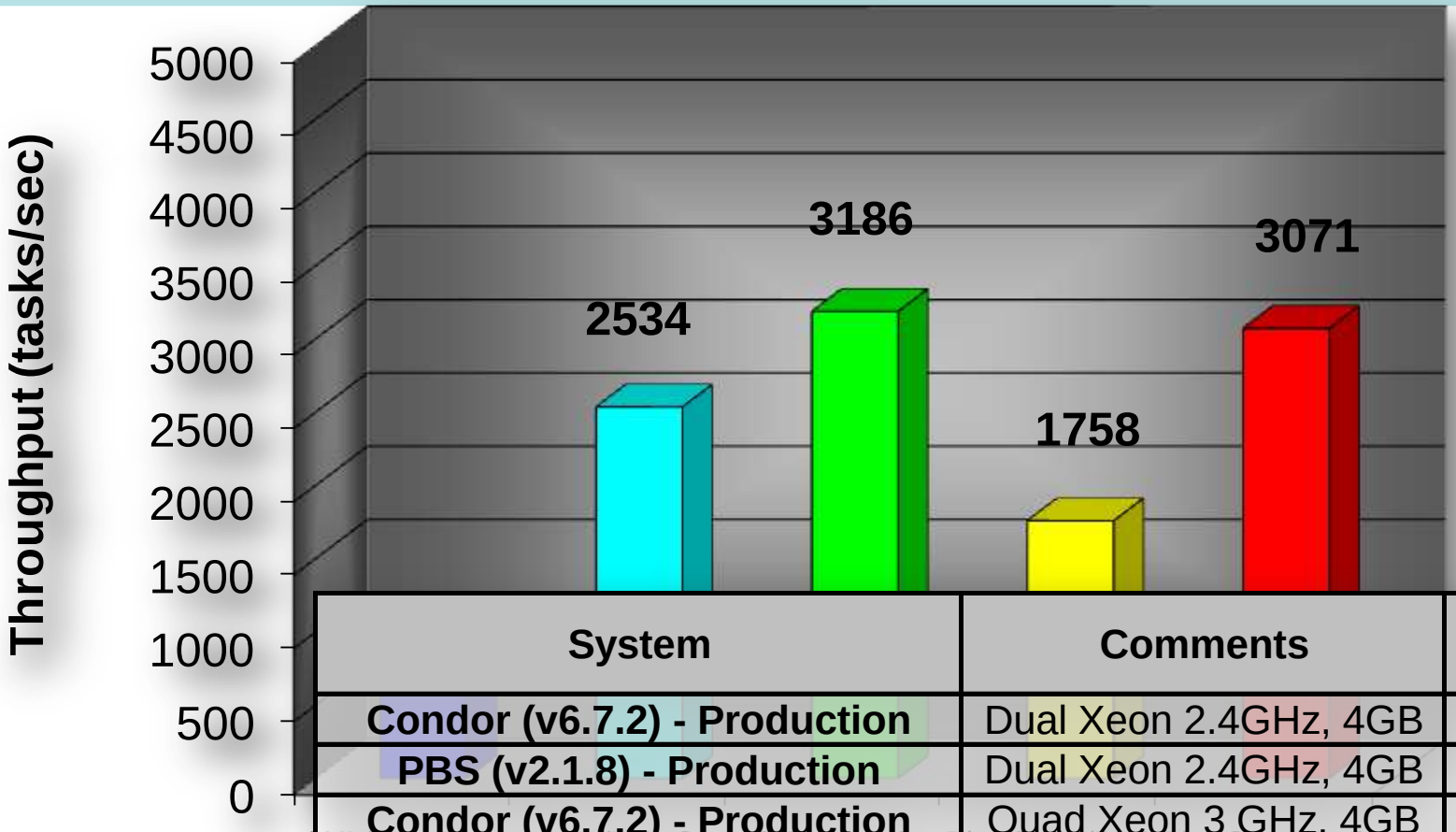
[SC08] "Towards Loosely-Coupled Programming on Petascale Systems"

Managing 160K CPUs
IBM Blue Gene/P



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

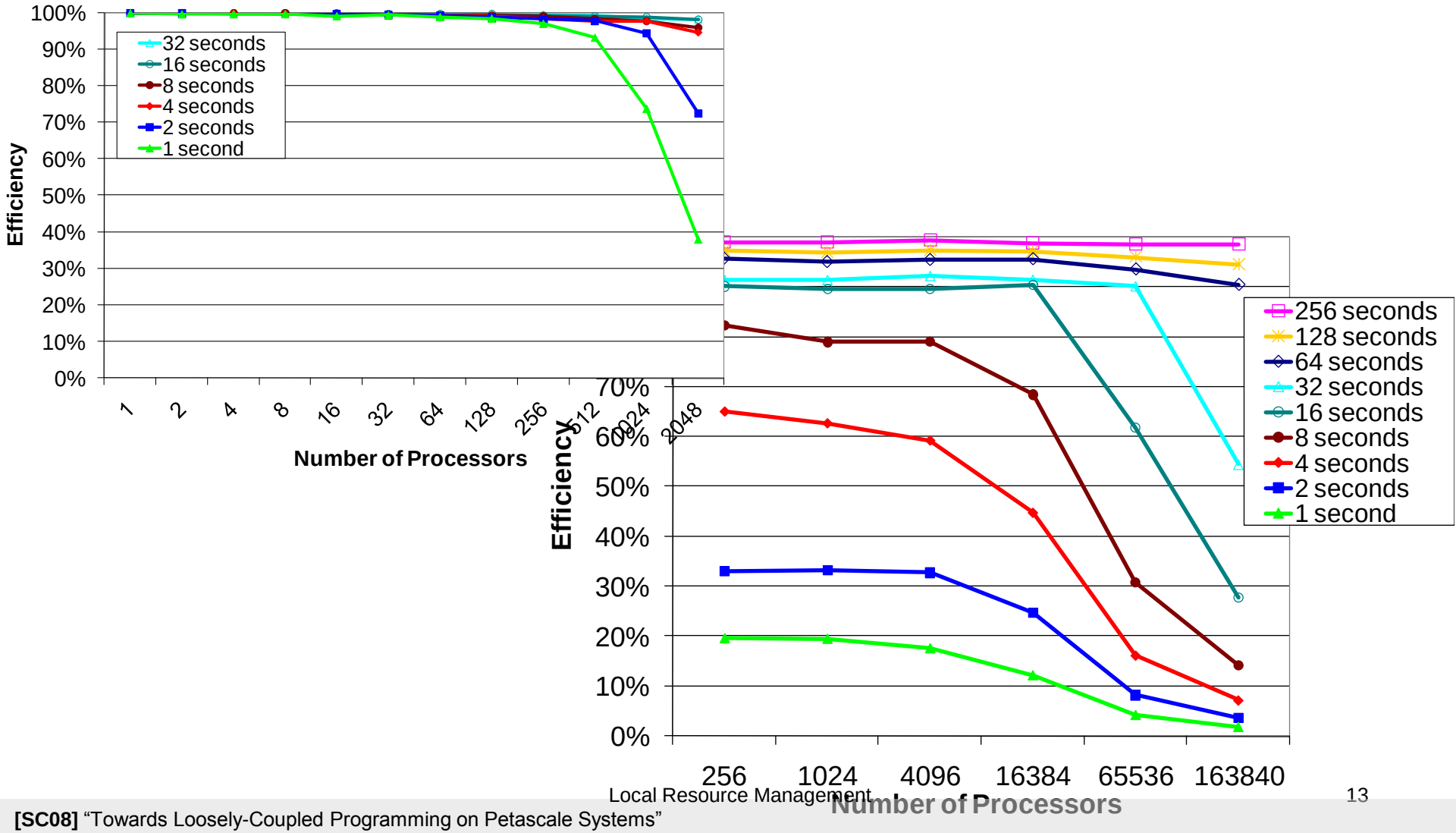
Dispatch Throughput



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

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

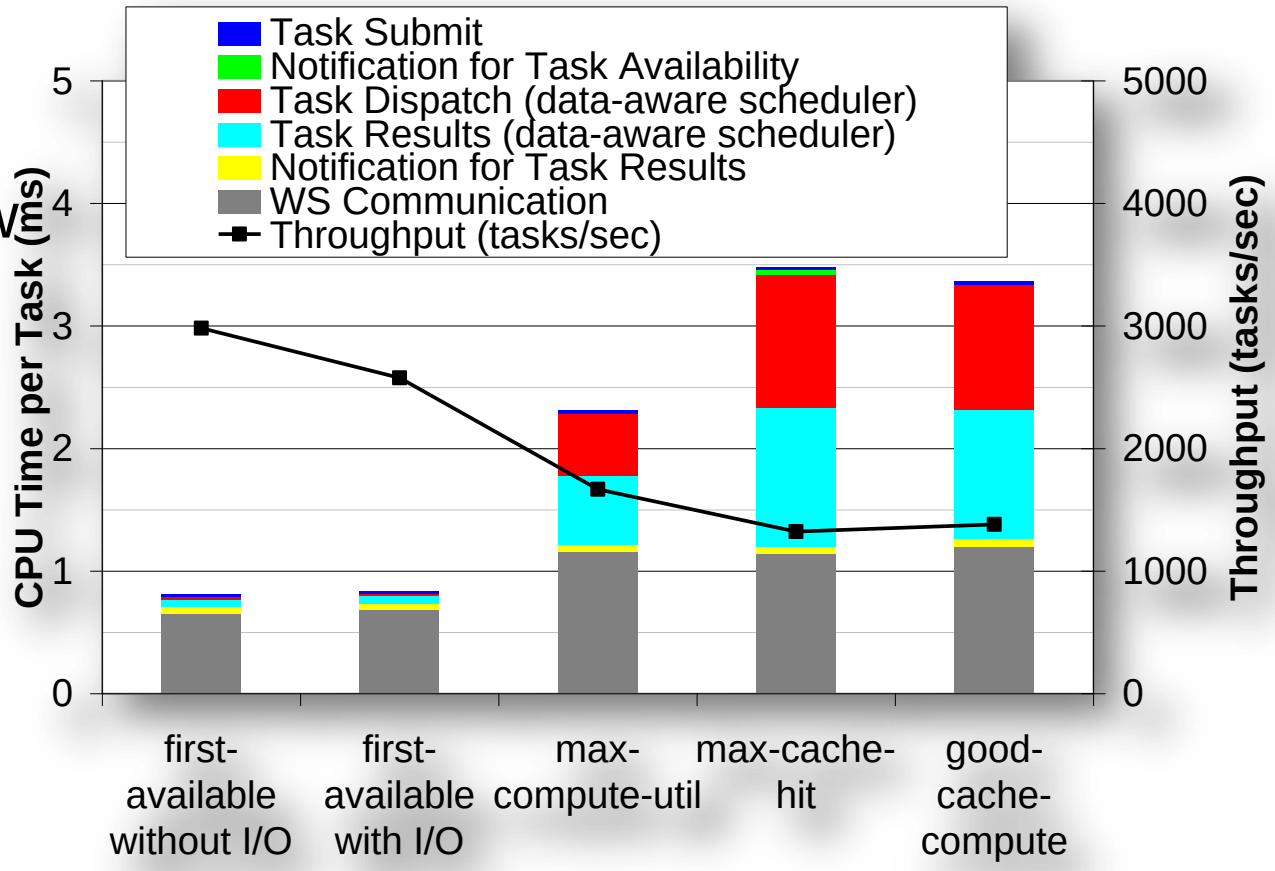
Execution Efficiency



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

Scheduler Profiling

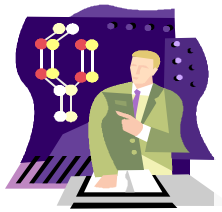
- 3GHz dual CPUs
- ANL/UC TG with 128 processors
- Scheduling window 2500 tasks
- Dataset
  - 100K files
  - 1 byte each
- Tasks
  - Read 1 file
  - Write 1 file

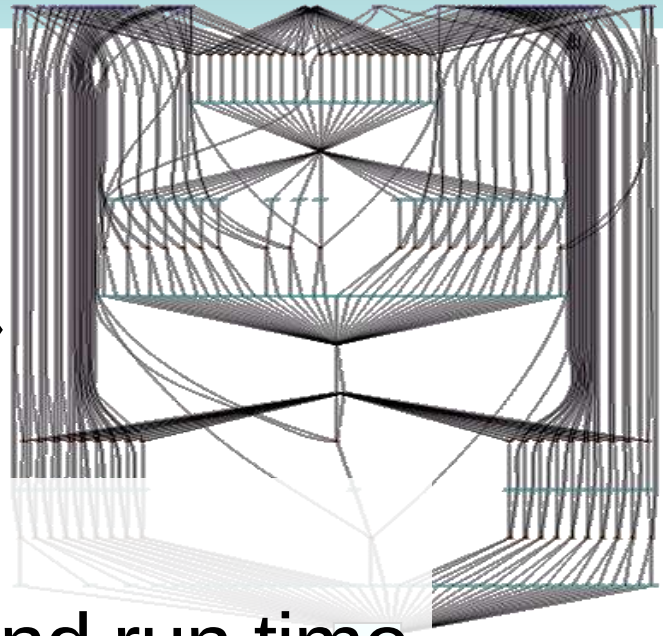


[DIDC09] "Towards Data Intensive Many-Task Computing", under review

Applications
Medical Imaging: fMRI



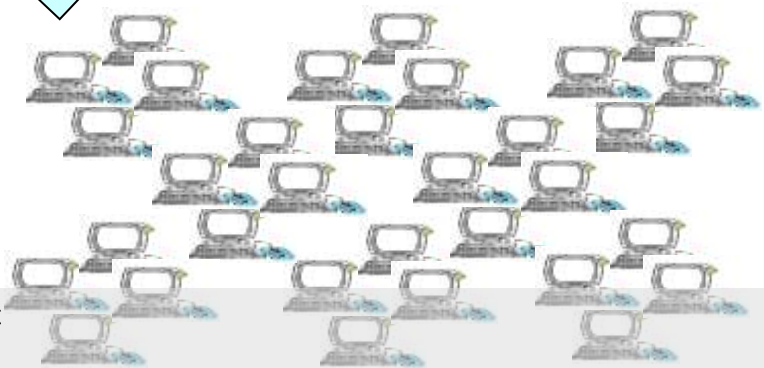




Improvement:

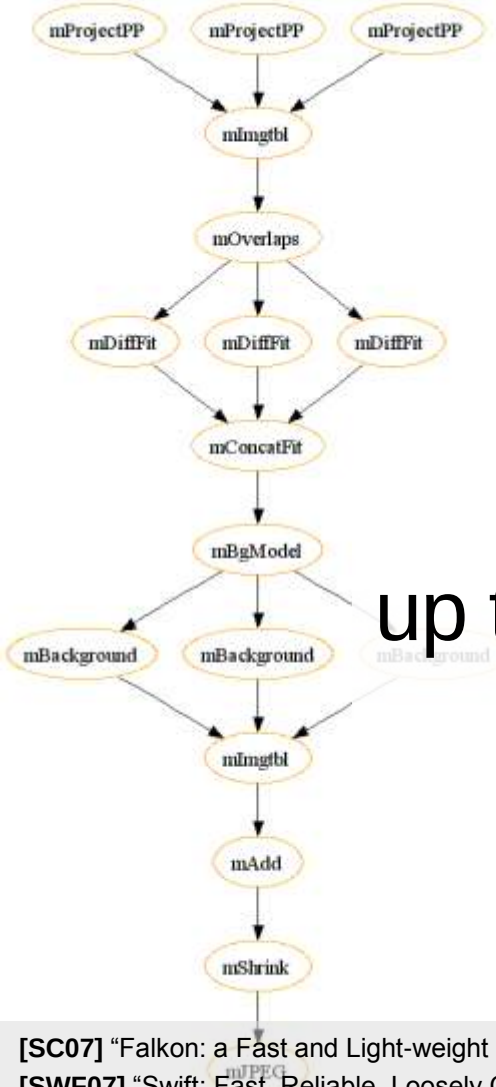
up to **90%** lower end-to-end run time

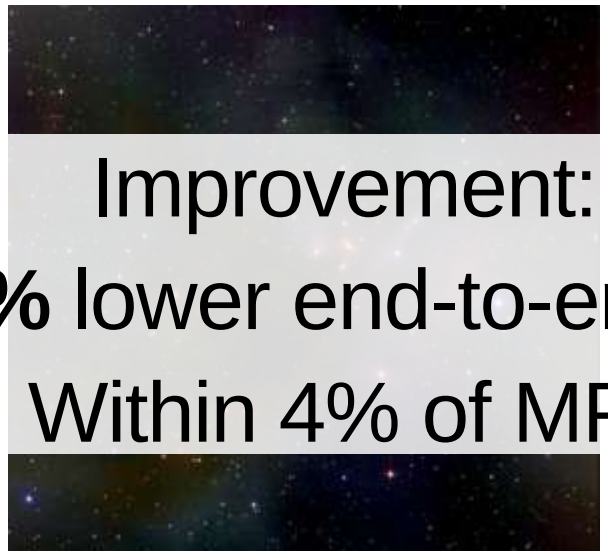
- Wide range of analyses
  - Testing, interactive analysis, production runs
  - Data mining
  - Parameter studies



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

Applications
Astronomy: Montage

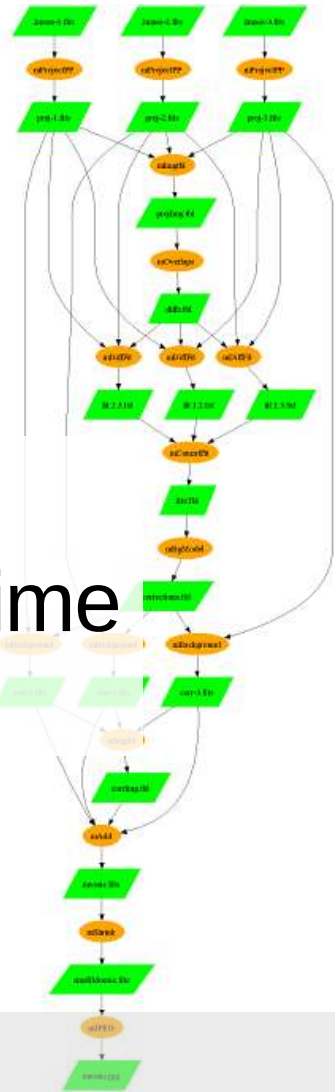




Improvement:

up to 57% lower end-to-end run time

Within 4% of MPI

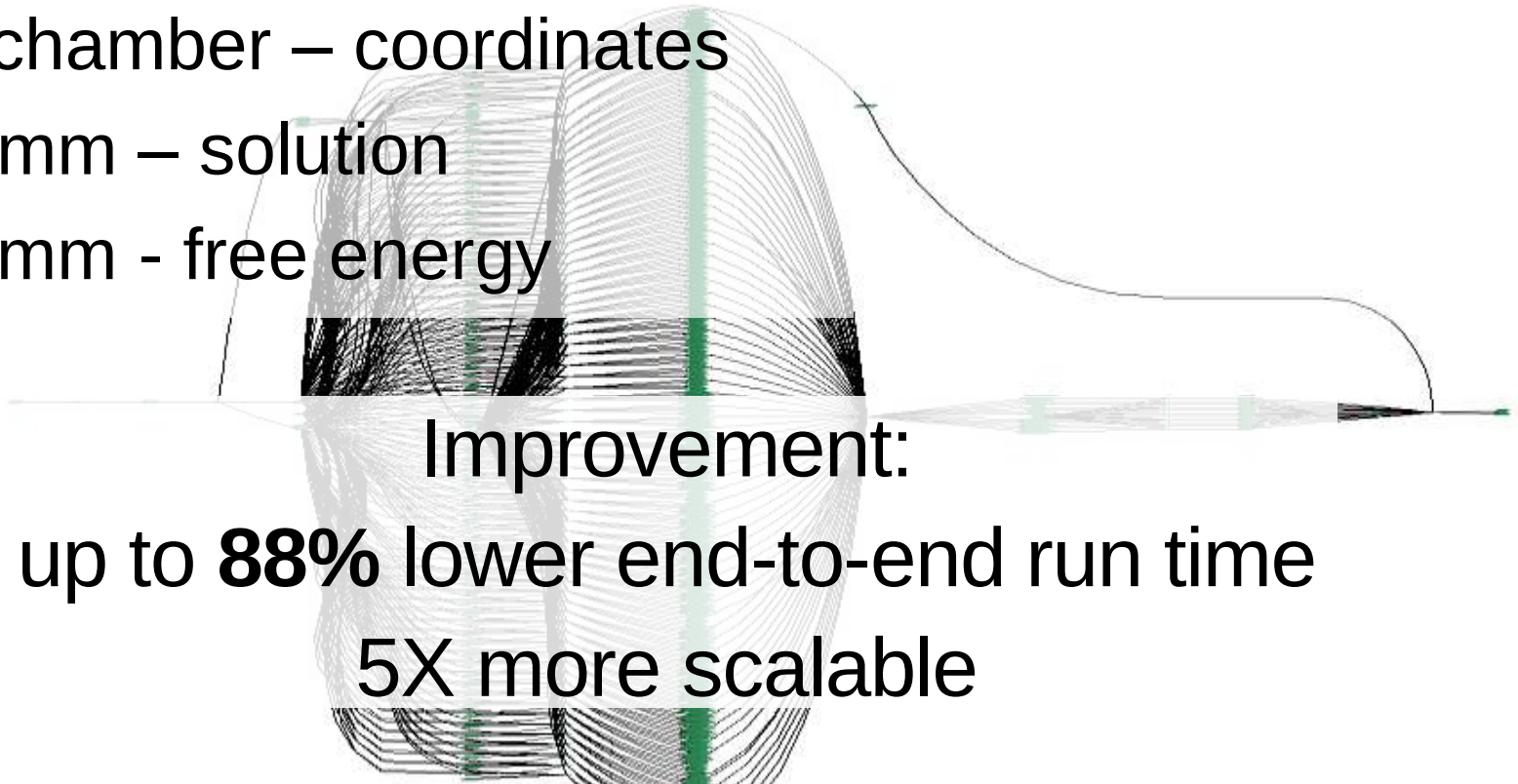


B. Berriman, J. Good (Caltech)
J. Jacob, D. Katz (JPL)

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

Applications
Molecular Dynamics: MolDyn

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



Improvement:

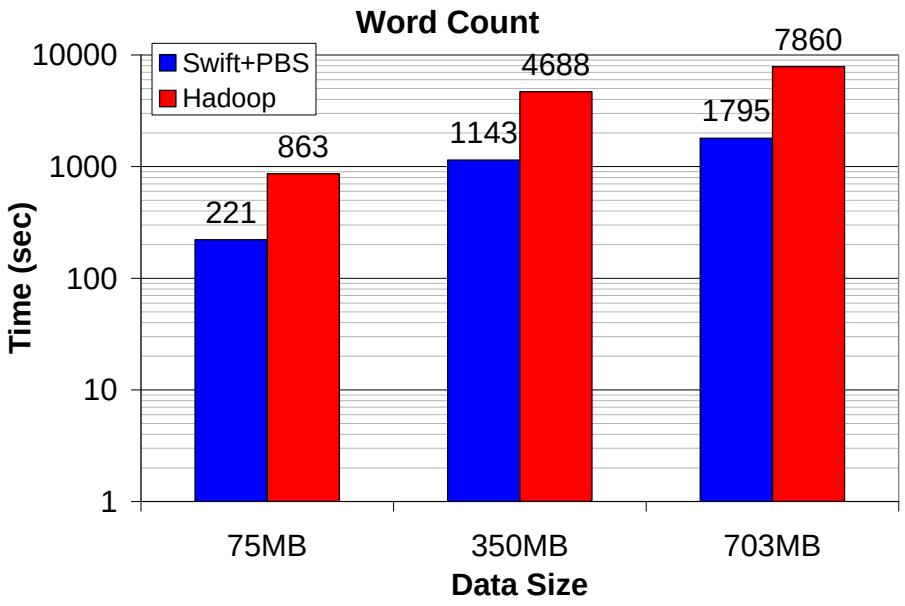
up to **88%** lower end-to-end run time

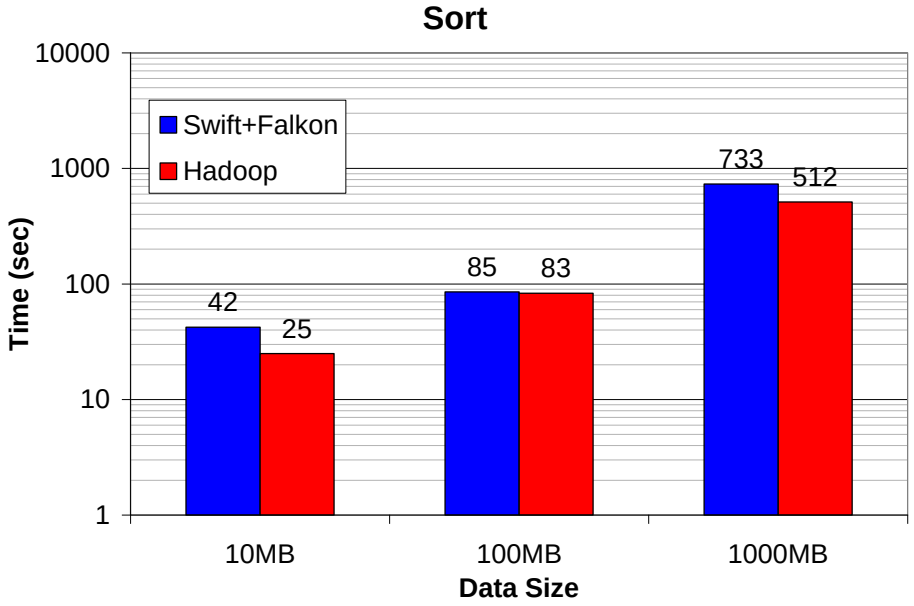
5X more scalable

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

Applications
Word Count and Sort

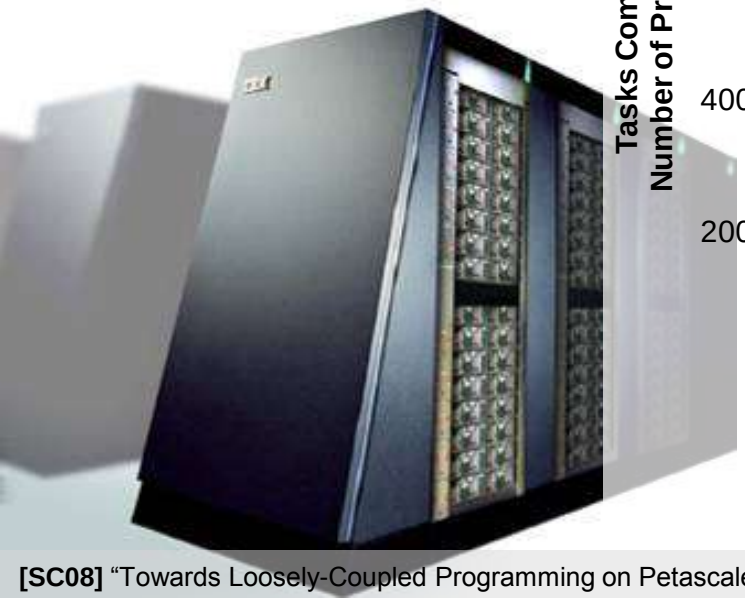
- Classic benchmarks for MapReduce
  - Word Count
  - Sort
- Swift and Falkon performs similar or better than Hadoop (on 32 processors)

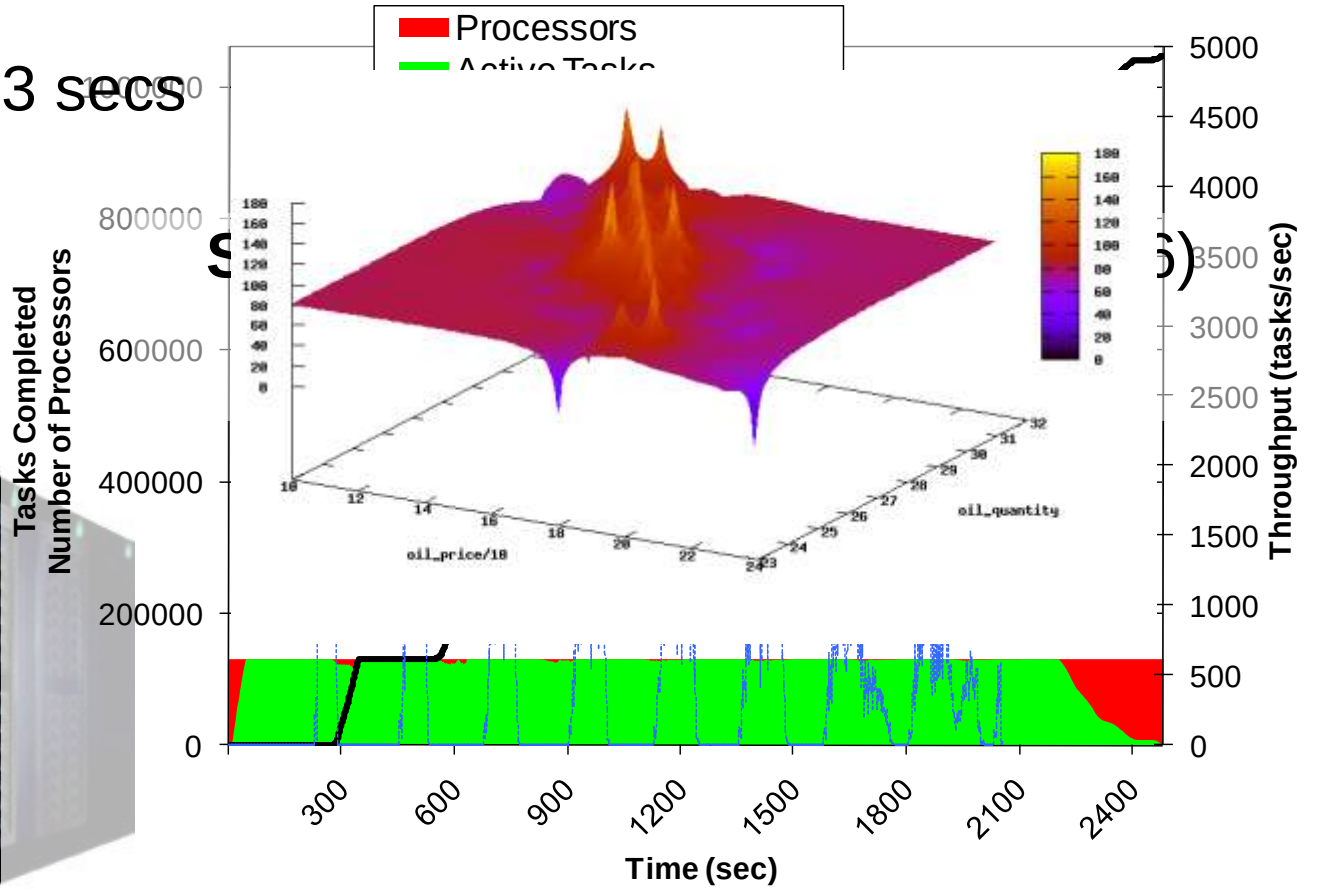




Applications
Economic Modeling: MARS

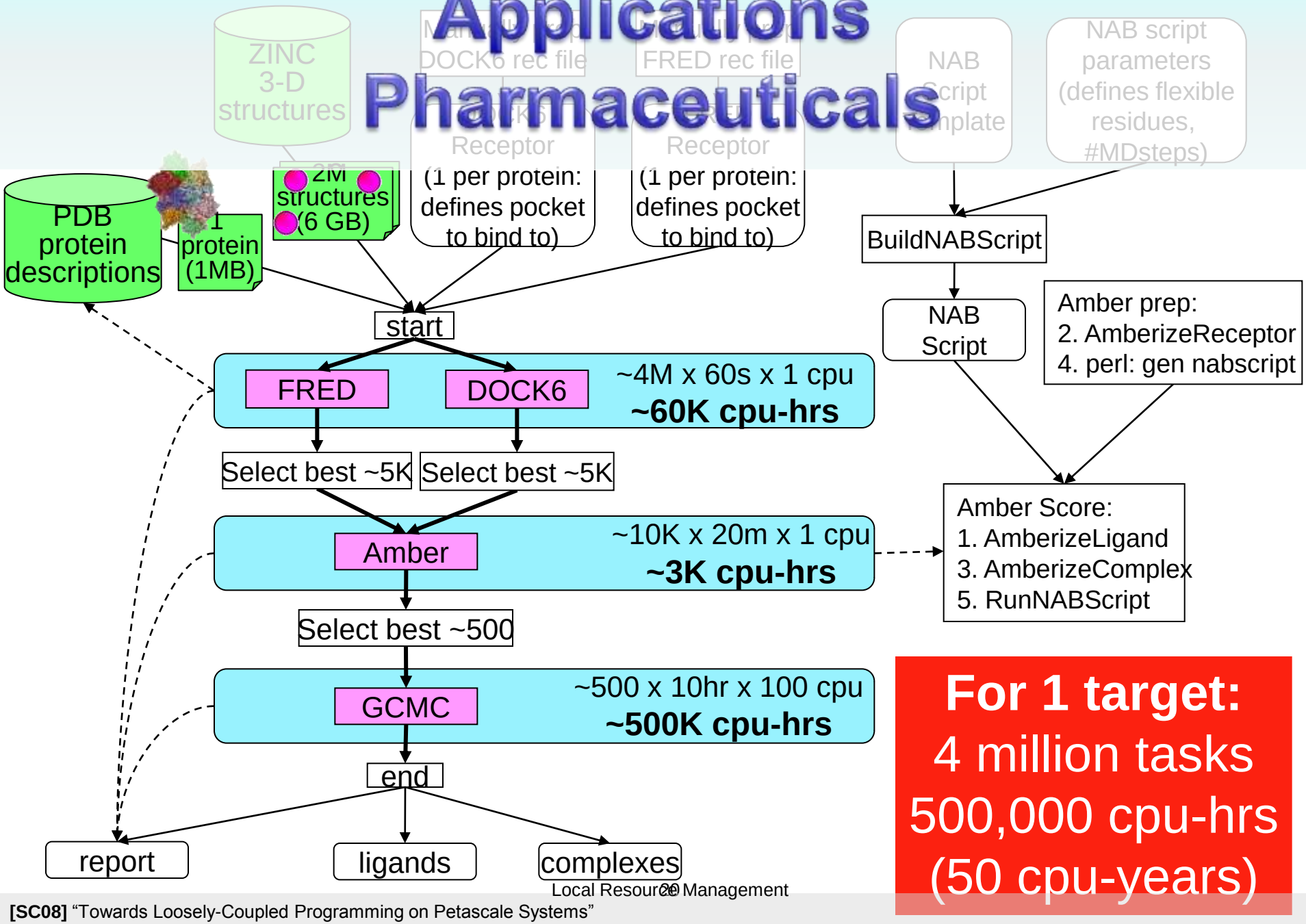
- CPU Cores: 130816
- Tasks: 1048576
- Elapsed time: 2483 secs
- CPU Years: 9.3





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

Applications Pharmaceuticals



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

Applications
Pharmaceuticals: DOCK

CPU cores: 118784

Tasks: 934803

Elapsed time: 2.01 hours

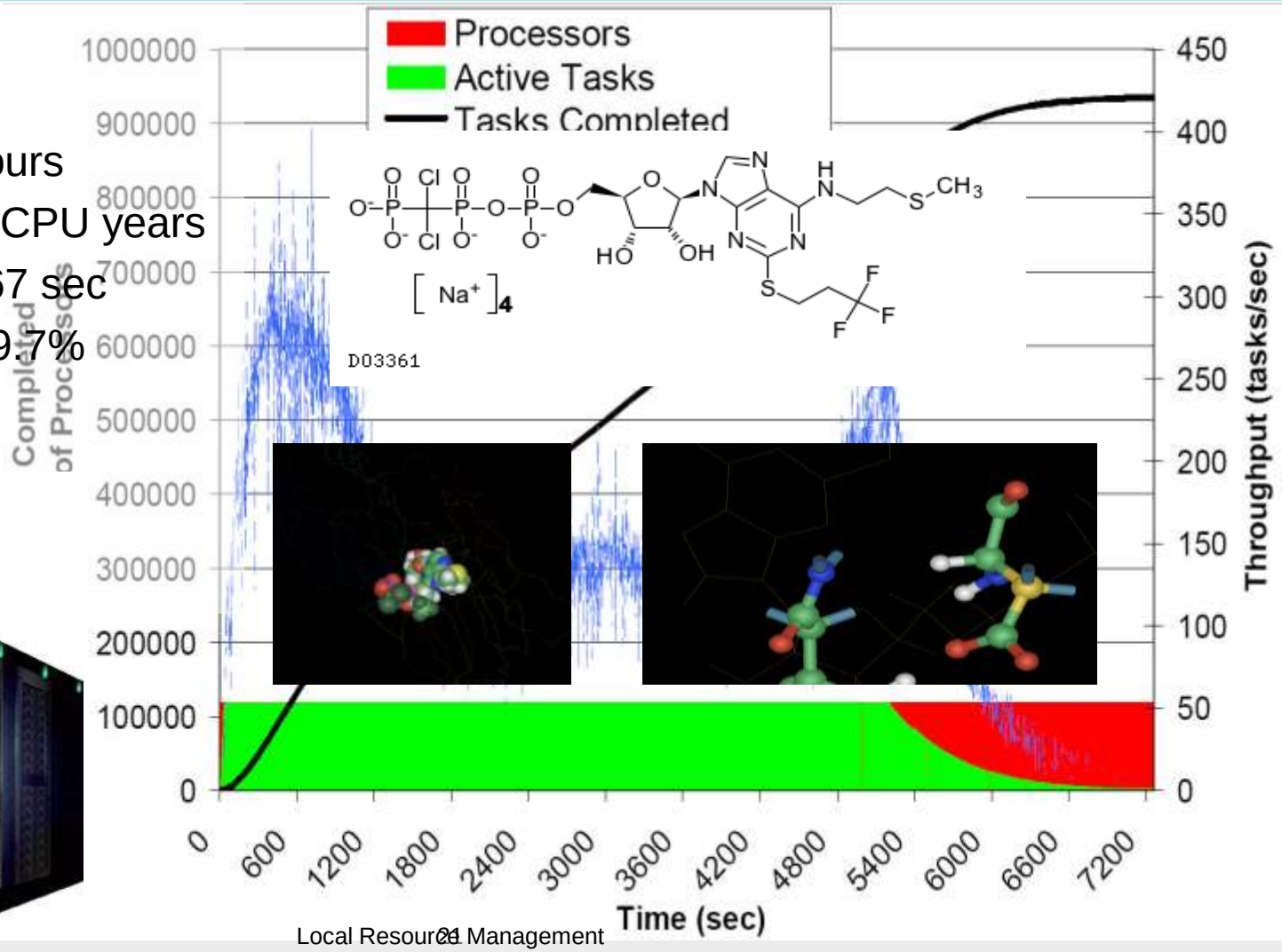
Compute time: 21.43 CPU years

Average task time: 667 sec

Relative Efficiency: 99.7%
(from 16 to 32 racks)

Utilization:

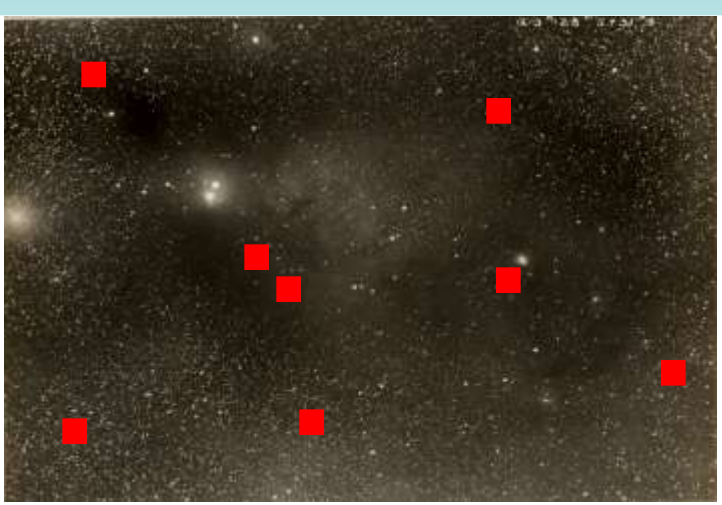
- Sustained: 99.6%
- Overall: 78.3%



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

Applications
Astronomy: AstroPortal

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

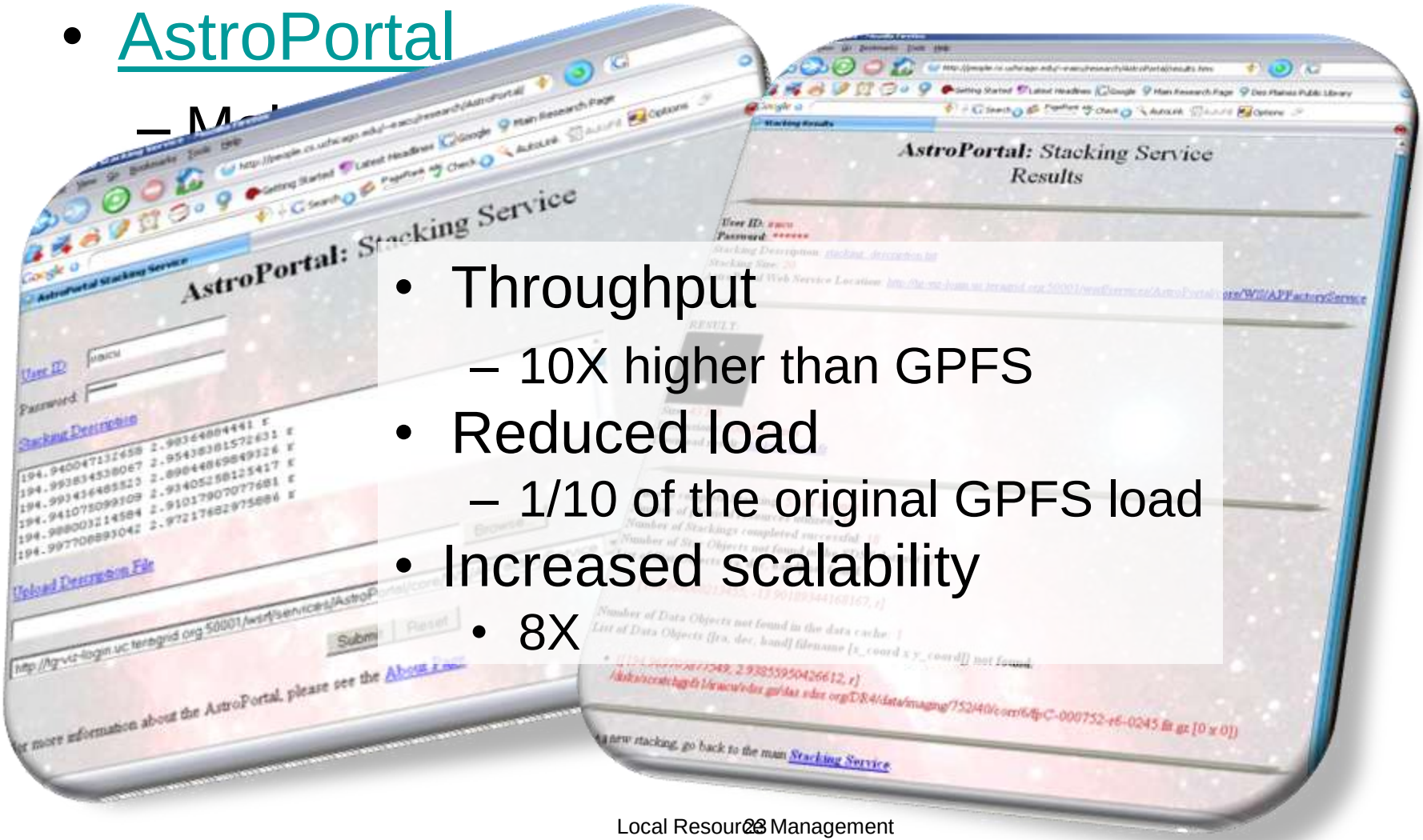


Locality	Number of Objects	Number of Files
1	111700	111700
1.38	154345	111699
2	97999	49000
3	88857	29620
4	76575	19145
5	60590	12120
10	46480	4650
20	40460	2025
30	23695	790

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

Applications
Astronomy: AstroPortal

- AstroPortal
  - Mo



- Throughput
  - 10X higher than GPFS
- Reduced load
  - 1/10 of the original GPFS load
- Increased scalability
  - 8X

Conclusions

- There is more to HPC than tightly coupled MPI, and more to HTC than embarrassingly parallel long jobs
  - MTC: Many-Task Computing
  - Addressed real challenges in resource management in large scale distributed systems to enable MTC
  - Covered many domains (via Swift and Falkon): astronomy, medicine, chemistry, molecular dynamics, economic modelling, and data analytics

- Identified that data locality is critical at large-scale ➔ data diffusion
  - Integrated streamlined task dispatching with data aware scheduling
  - Heuristics to maximize real world performance
  - Suitable for varying, data-intensive workloads
  - Proof of O(NM) Competitive Caching

Mythbusting

- ~~Embarrassingly~~ Happily parallel apps are trivial to run
  - Logistical problems can be tremendous
- Loosely coupled apps do not require “supercomputers”
  - Total computational requirements can be enormous
  - Individual tasks may be tightly coupled
  - Workloads frequently involve large amounts of I/O
  - Make use of idle resources from “supercomputers” via backfilling
  - Costs to run “supercomputers” per FLOP is among the best
- Loosely coupled apps do not require specialized system software
  - Their requirements on the job submission and storage systems can be extremely large
- Shared/parallel file systems are good for all applications
  - They don’t scale proportionally with the compute resources
  - Data intensive applications don’t perform and scale well
  - Growing compute/storage gap

It’s all about options.

Your very own Microsoft Recruiter (Alicia!) will be on campus this week!
Bring yourself (and pony up your resume, if you have it) to her anytime ***Fri 1/22***.
That’s it. You’re done. You’ve applied to Microsoft.
Nice.

FRIDAY, JAN. 22

In front of the Engineering Career Center…
9am – 11 am
Free Donuts, yum.

Career Fair (Norris Univ. Center)
2 pm – 6 pm
Free Micro-stuff, cool.

#1 Technology Internship in the Nation
Business Week’s “Best Places to Start”