



THE UNIVERSITY OF
CHICAGO



Toward Loosely Coupled Programming on Petascale Systems

Ioan Raicu

Distributed Systems Laboratory
Computer Science Department
University of Chicago

In Collaboration with:

Zhao Zhang & Ben Clifford, Univ. of Chicago

Mike Wilde & Ian Foster, Univ. of Chicago and Argonne National Lab.

Pete Beckman & Kamil Iskra, Argonne National Lab.

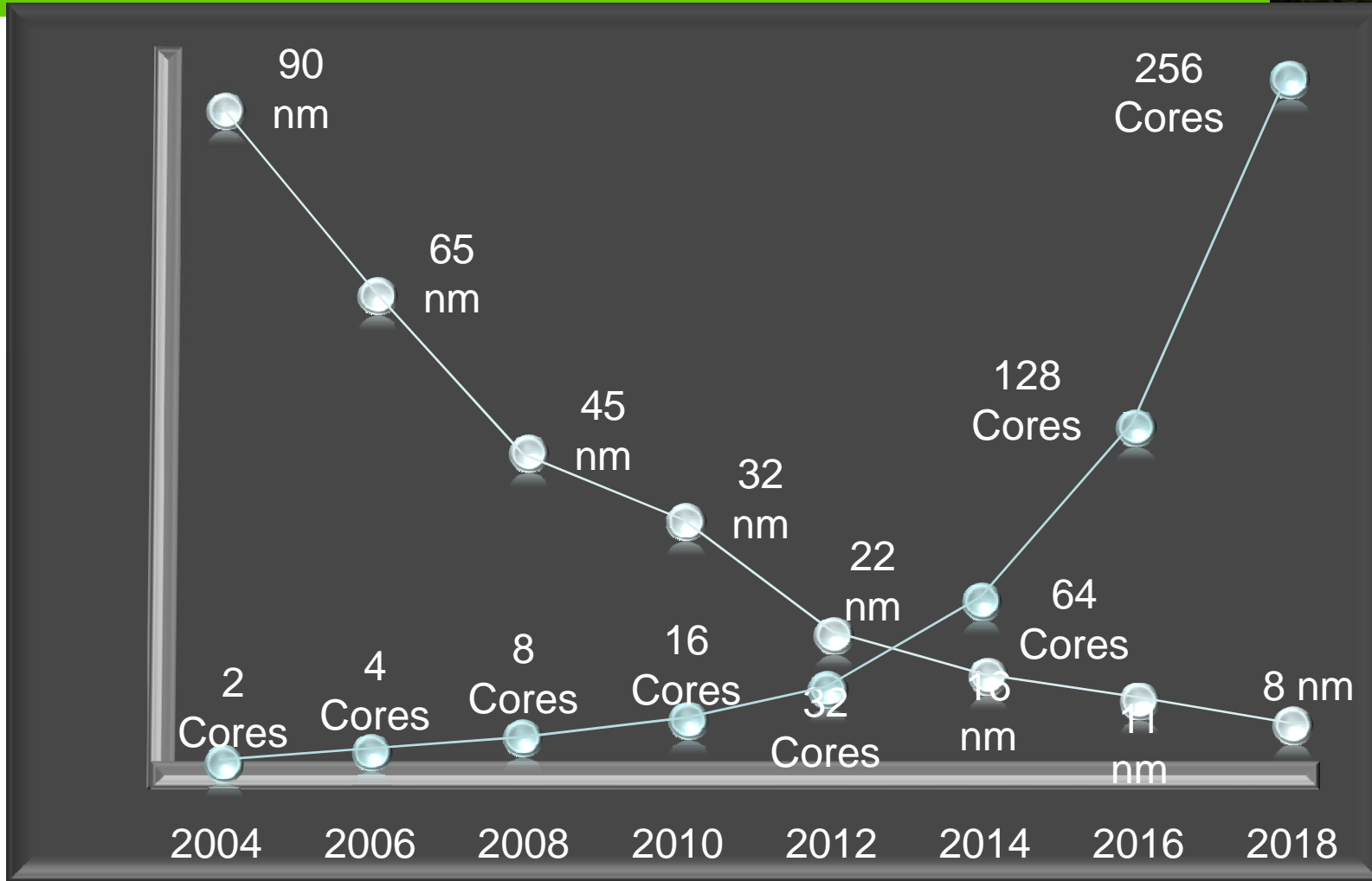
IEEE/ACM Supercomputing 2008
November 18th, 2008

PART I

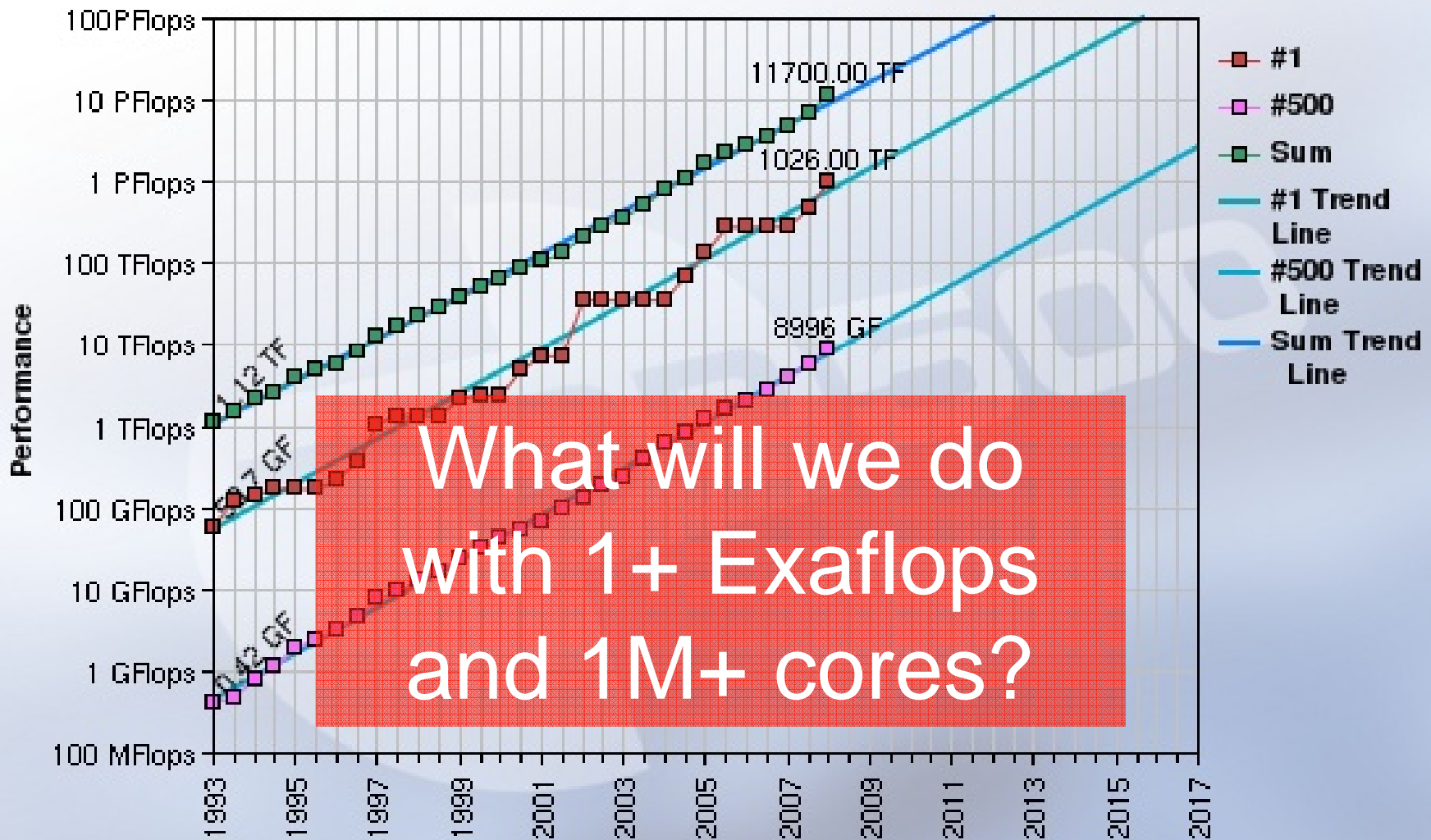


Motivation

Many-Core Growth Rates



Projected Performance Development

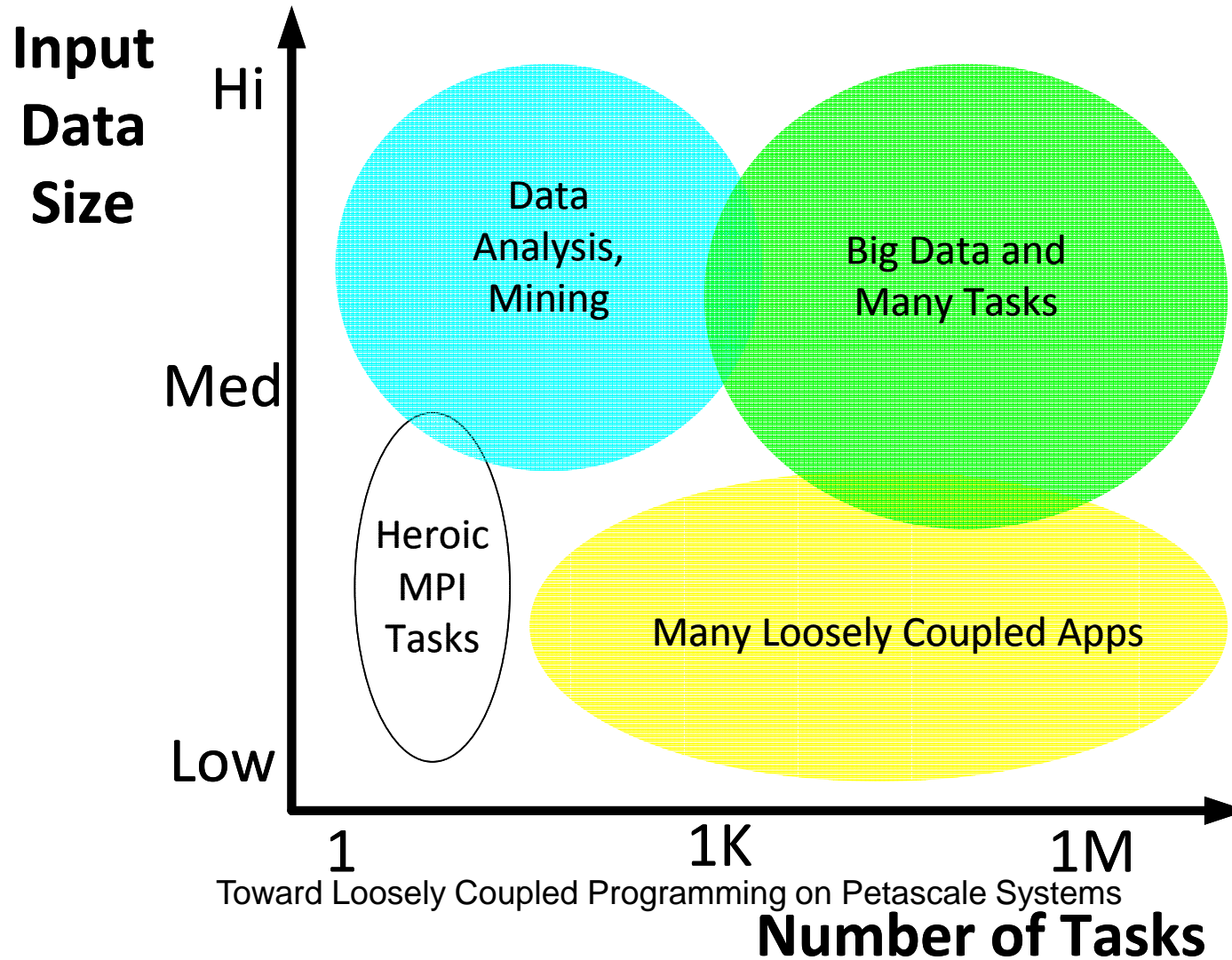


Programming Model Issues

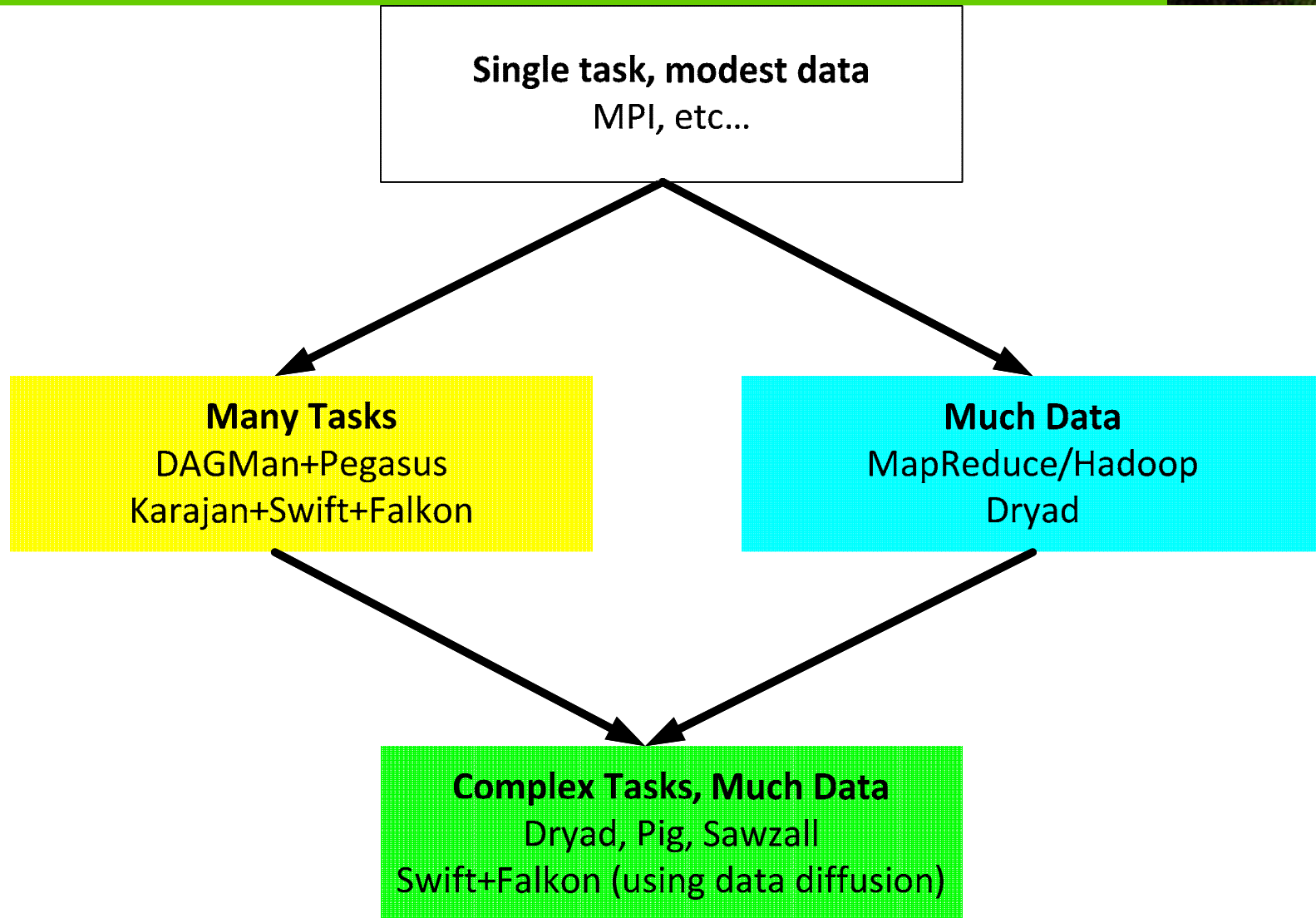


- **Multicore/Manycore** processors
- Massive **task parallelism**
- Massive **data parallelism**
- Integrating **black box applications**
- Complex **task dependencies** (task graphs)
- **Failure**, and other execution management issues
- **Dynamic task graphs**
- Documenting **provenance** of data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

Problem Types



An Incomplete and Simplistic View of Programming Models and Tools



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...

MTAGS08: Workshop on Many-Task Computing on Grids and Supercomputers - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://dsl.cs.uchicago.edu/MTAGS08/

Most Visited IGoogle Ioan Raicu's Web Site MTAGS08: Workshop o... Dashboard - Google An... Incubator/Falcon - Glo... Outreach/SC2008 - Glo... Linked in

Google top500 Search Bookmarks PageRank Settings

MTAGS

Workshop on Many-Task Computing on Grids and Supercomputers

co-located with ACM/IEEE SC08 (International Conference for High Performance, Networking, Storage and Analysis)
Austin, Texas -- November 17th, 2008

Home **Workshop Program**

Call for Papers
[\(TXT, PDF\)](#)

Program Committee

Important Dates

Paper Submission

Venue

Registration

Workshop Program

The MTAGS08 will be on Monday, November 17th, 2008, in room 11AB, from 8:30AM to 12PM. Please check in early to get your SC08 badge from the registration booths, as lines might be long. See below for the 7 talks that are part of the workshop, their abstracts, papers, and slides.

Time	Description	Authors	Links
8:30AM	Introduction		
8:35AM	<i>Keynote Talk: Many-task applications in use today with a look toward the future</i>	Alan Gara, IBM Fellow and Blue Gene chief architect, IBM Research	Abstract Slides
Session 1 (Reagan Moore, Session Chair)			
9:05AM	<i>Design and Evaluation of a Collective I/O Model for Loosely-coupled Peta-scale Programming</i>	Zhao Zhang, Univ. of Chicago Allan Espinosa, Univ. of Chicago Kanul Iskra, Argonne National Lab. Ioan Raicu, Univ. of Chicago	Abstract Paper Slides
9:30AM	<i>Exploring Data Parallelism in the Design of a Many-Task Computing System</i>	Alan Gara, IBM Fellow and Blue Gene chief architect, IBM Research	Abstract Paper Slides
9:55AM	<i>System Support for Many-Task Computing</i>	Ron Minnich, Sandia National Labs	Abstract Paper Slides
10:30AM	<i>A lightweight execution framework for massive independent tasks</i>	Li Hui, Peking University, China Yu Huashan, Peking University, China Li Xiaoming, Peking University, China	Abstract Paper Slides
10:55AM	Break		
Session 3 (Yong Zhao, Session Chair)			
11:10AM	<i>ViGs: A Grid Simulation and Monitoring Tool for Grid Workflows</i>	A.T. Thor, Univ. of Texas at Arlington G.V. Zaruba, Univ. of Texas at Arlington D. Levine, Univ. of Texas at Arlington K. De, Univ. of Texas at Arlington T.J. Wenaus, Univ. of Texas at Arlington	Abstract Paper Slides
11:35AM	<i>Embarrassingly Parallel Jobs Are Not Embarrassingly Easy to Schedule on the Grid</i>	Enis Afgan, Univ. of Alabama at Birmingham Purushotham Bangalore, Univ. of Alabama at Birmingham	Abstract Paper Slides
12:00PM	Closing Statements		



MTAGS08

Workshop on Many-Task Computing on Grids and Supercomputers

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) *[to be announced in 90 minutes]*
 - DOE, Oak Ridge National Laboratory
- Intrepid (#5)
 - DOE, Argonne National Laboratory
- Ranger (#6)
 - University of Texas / NFS TeraGrid

Why Petascale Systems for MTC Applications?



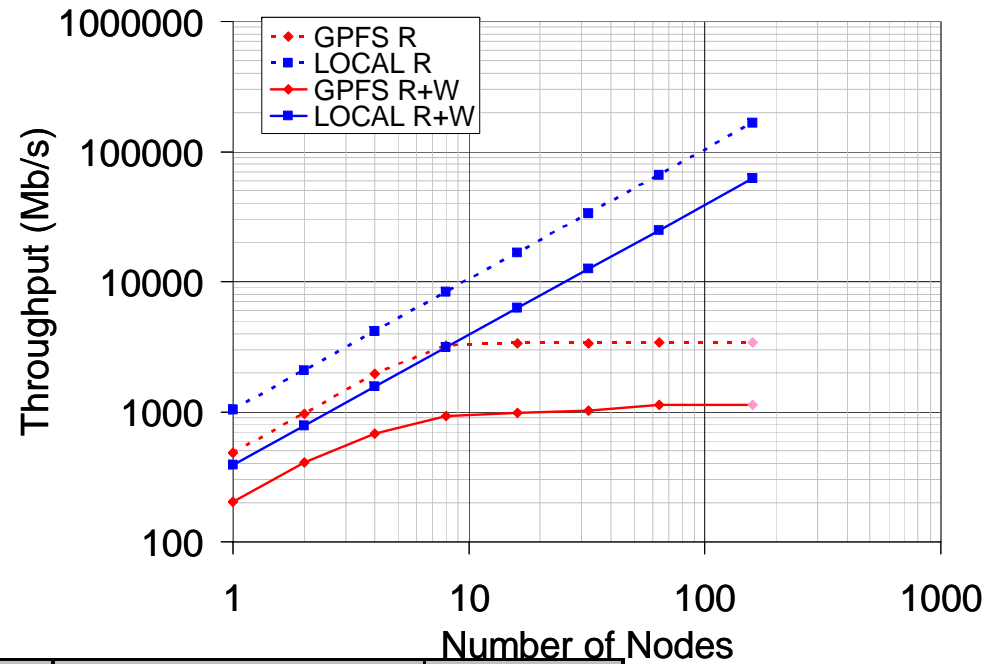
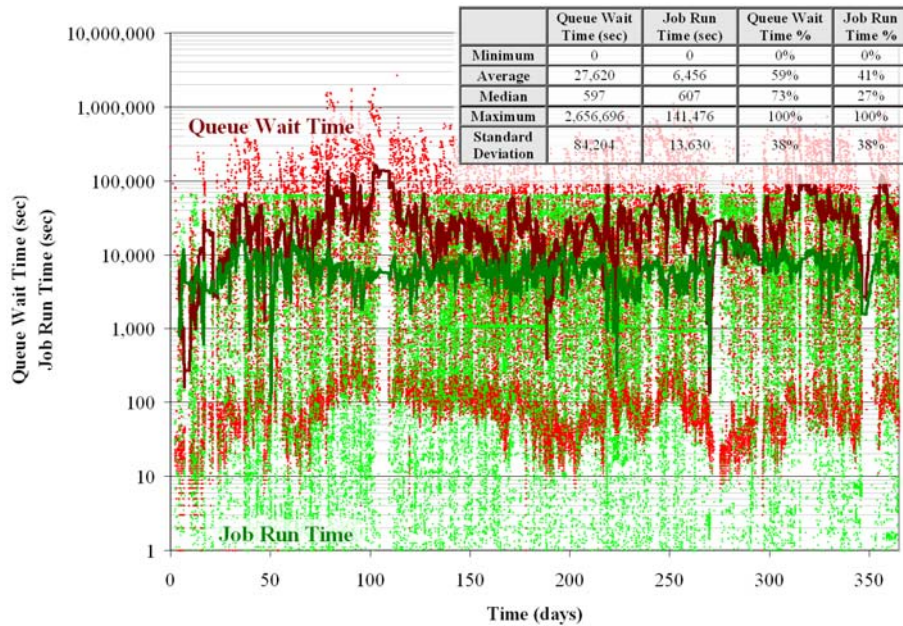
1. The I/O subsystem of petascale systems offers unique capabilities needed by MTC applications
2. The cost to manage and run on petascale systems is less than that of conventional clusters or Grids
3. Large-scale systems that favor large jobs have utilization issues
4. Some problems are intractable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

PART II



Some context on
systems we used as
building blocks

Obstacles running MTC apps in Clusters/Grids



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

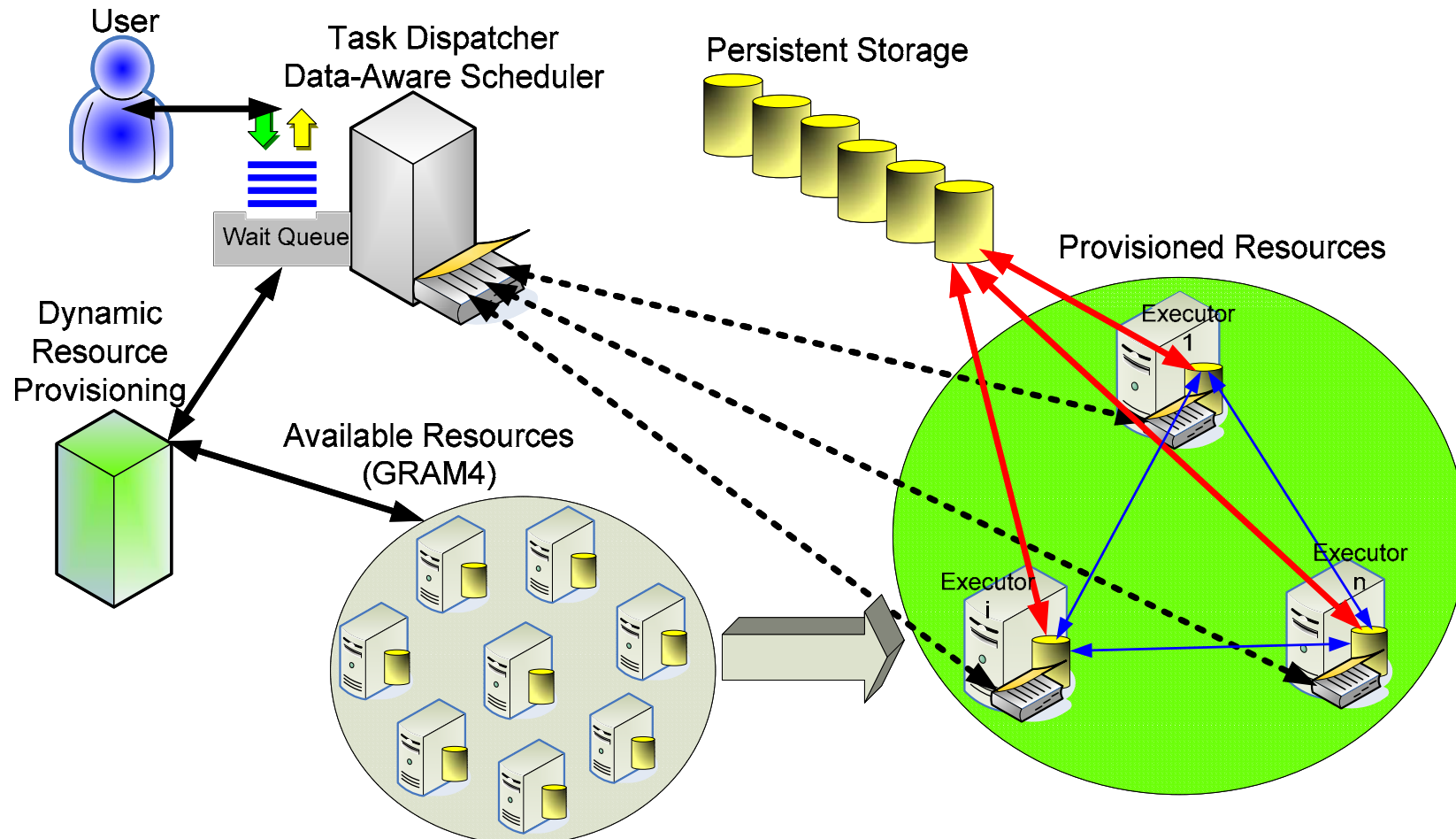
Toward Loosely Coupled Programming on Petascale Systems

Solutions



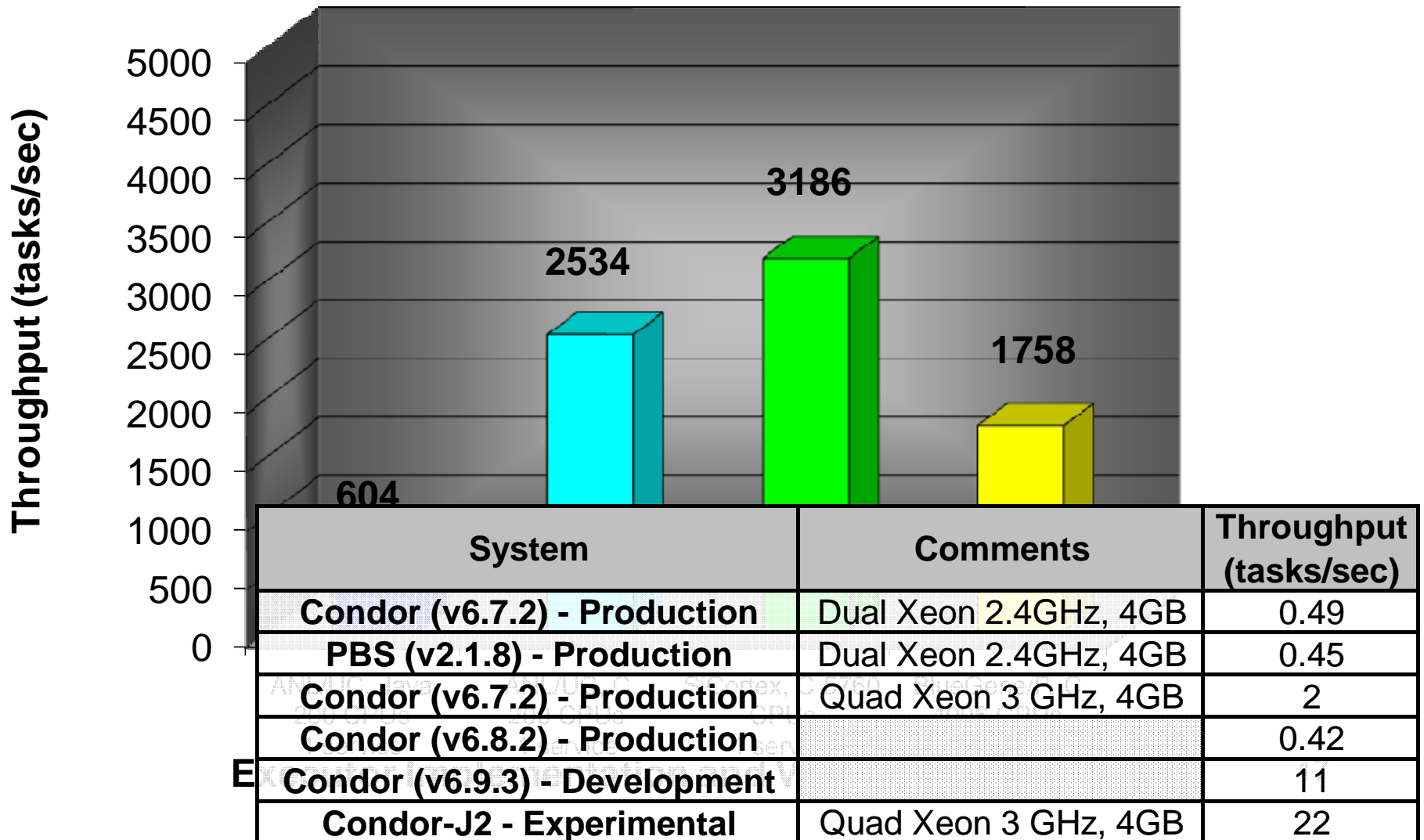
- Falkon: A Fast and Light-weight task executiON framework
 - **Goal:** enable the *rapid and efficient* execution of many independent jobs on large compute clusters
 - Combines three components:
 - A *streamlined task dispatcher*
 - *Resource provisioning* through multi-level scheduling techniques
 - *Data diffusion* and data-aware scheduling to leverage the co-located computational and storage resources
- Swift: A parallel programming system for loosely coupled applications
 - Applications cover many domains: Astronomy, astro-physics, medicine, chemistry, economics, climate modeling, data analytics

Falkon Overview

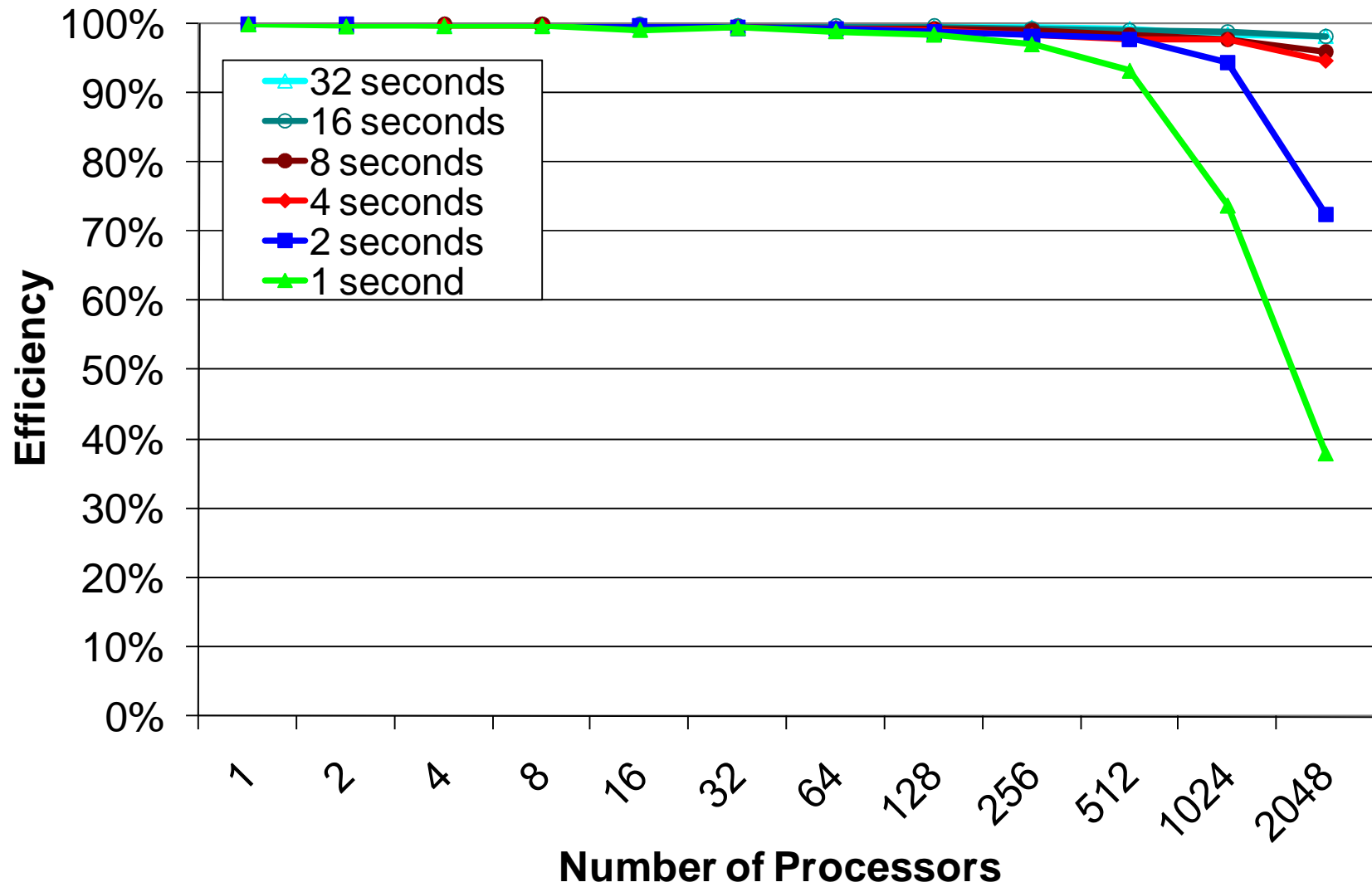


Toward Loosely Coupled Programming on Petascale Systems

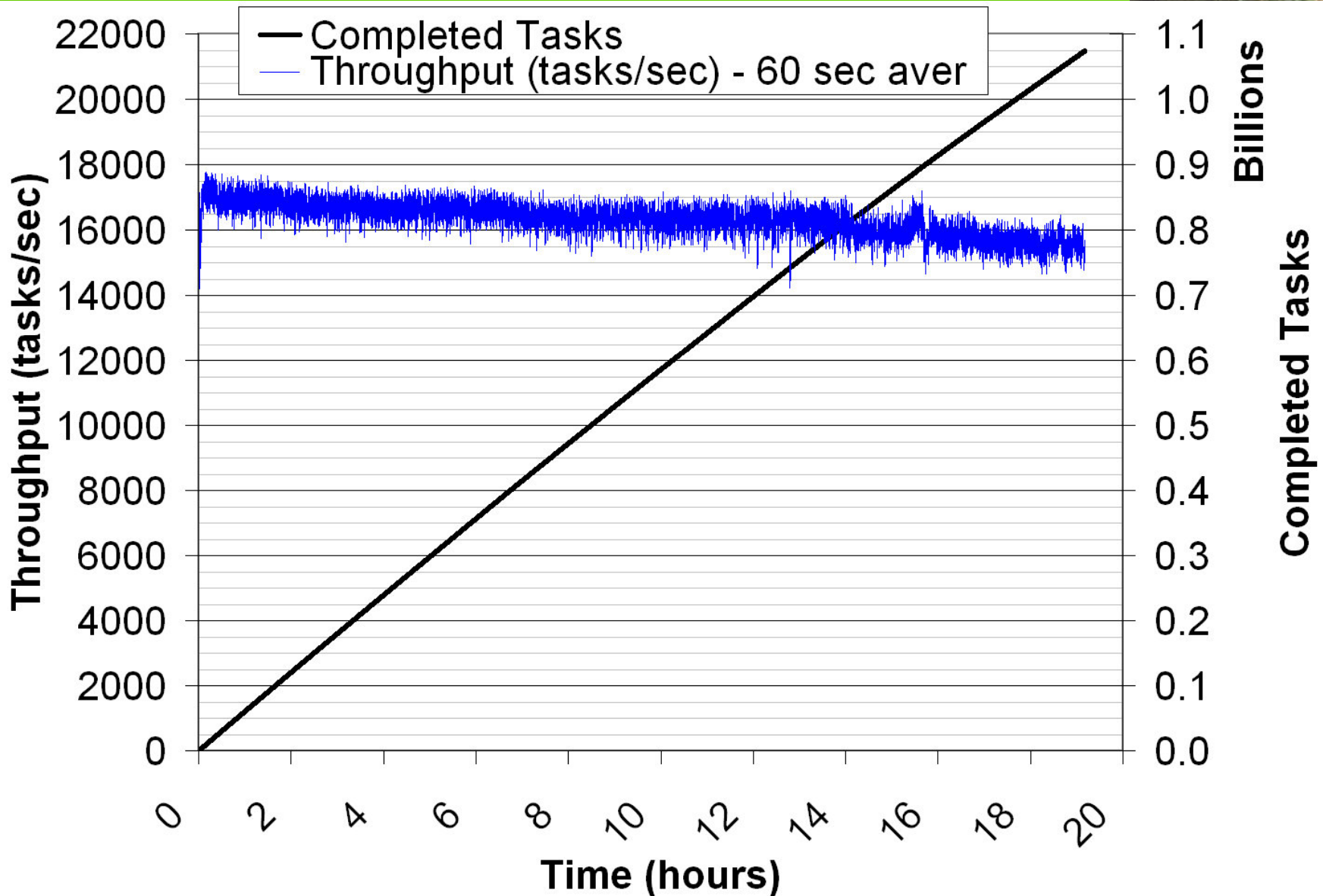
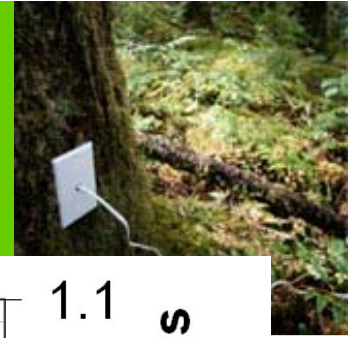
Dispatch Throughput



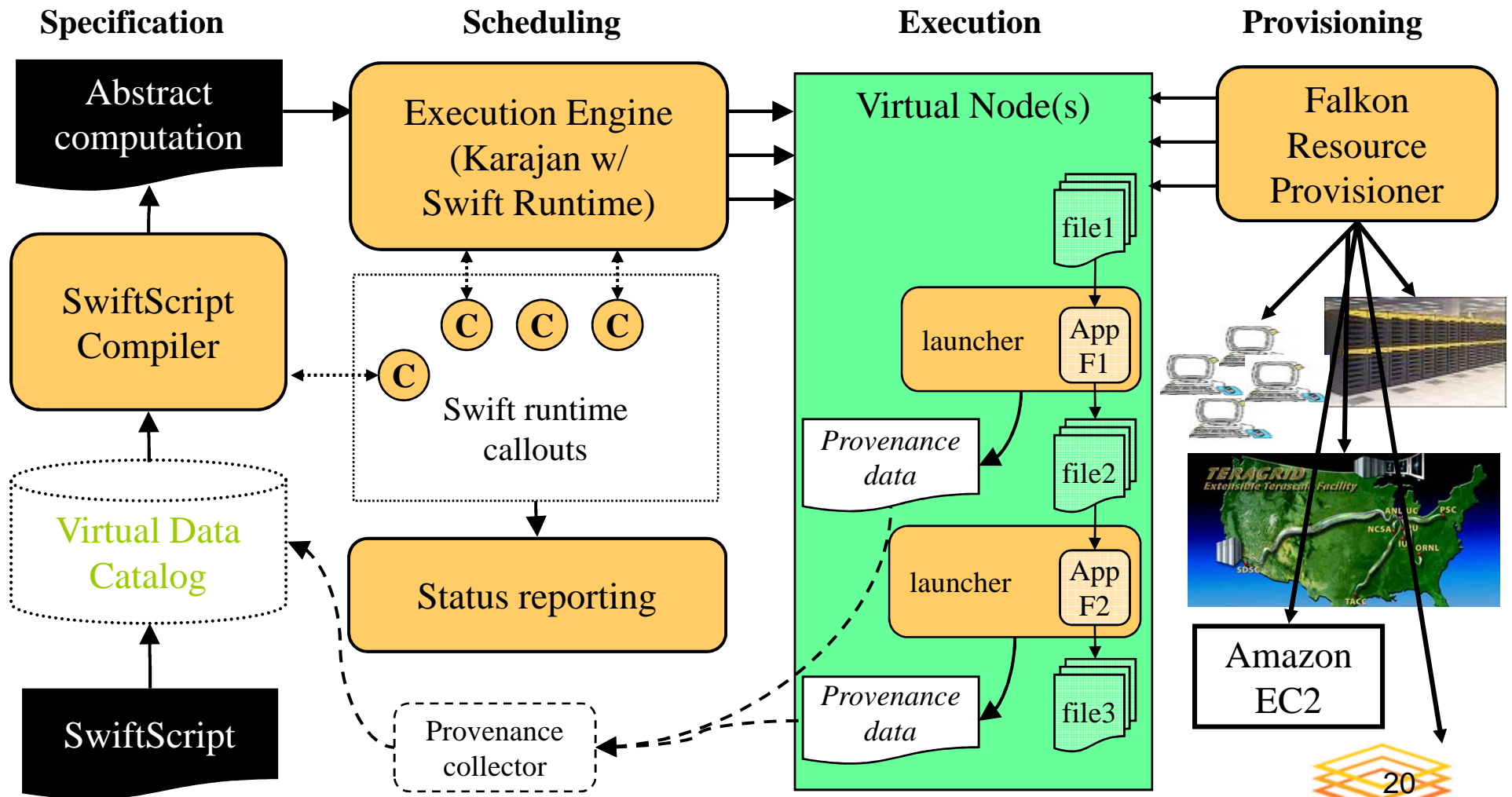
Efficiency



Falkon Endurance Test



Swift Architecture



Toward Loosely Coupled Programming on Petascale Systems

PART III



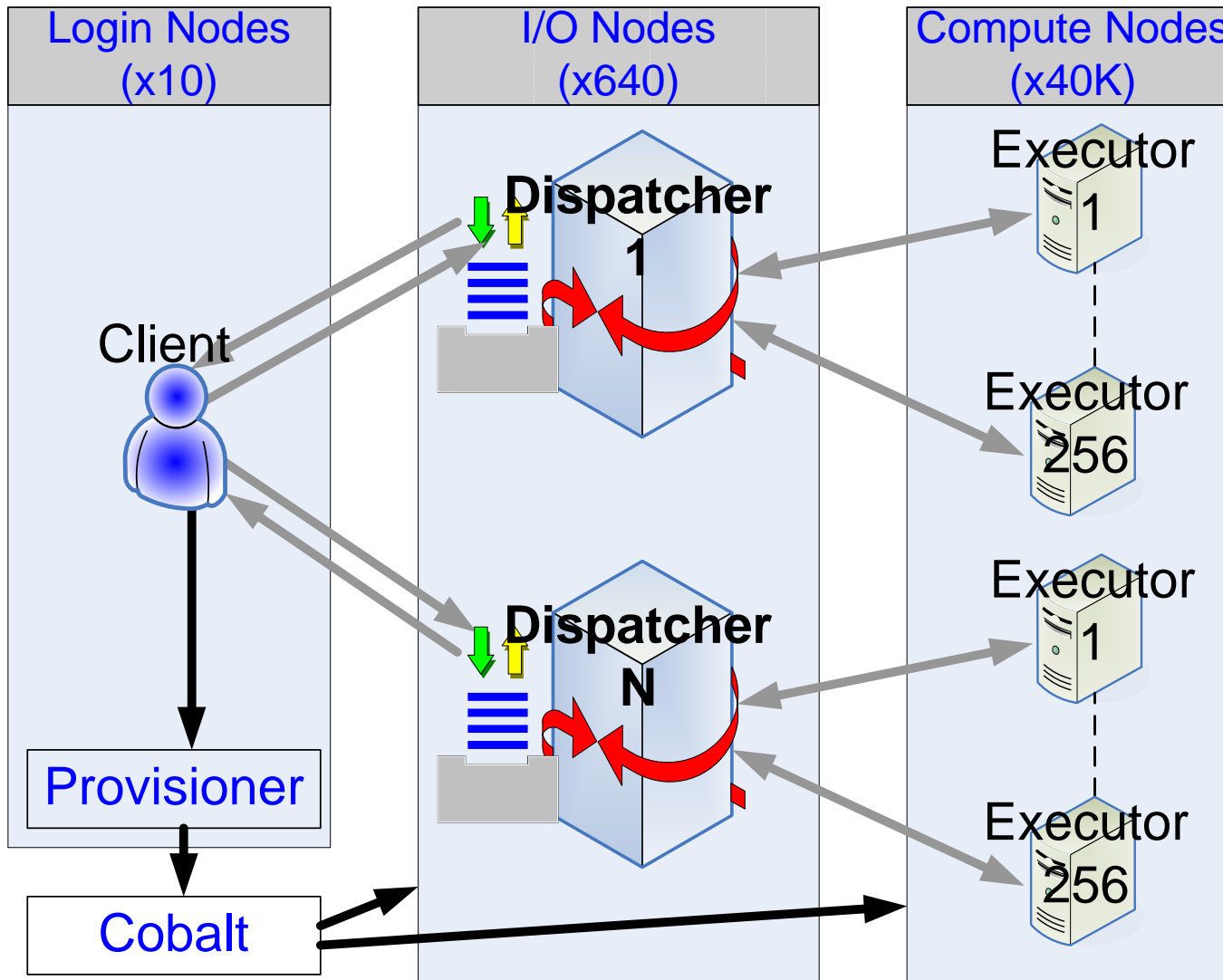
Contributions: Proposed Changes & Results

Scaling from 1K to 100K CPUs

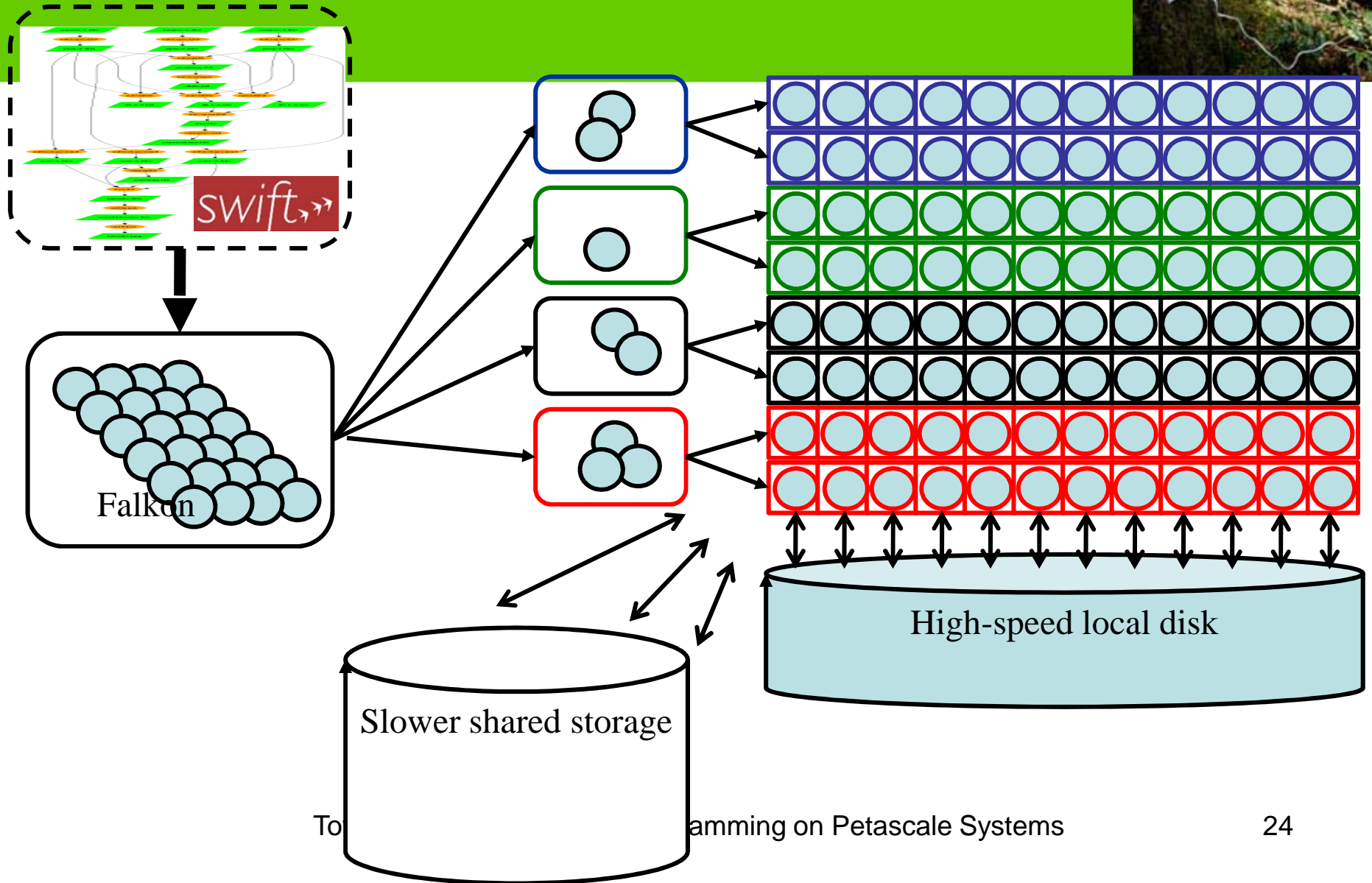


- At 1K CPUs:
 - 1 Server to manage all 1K CPUs
 - Use shared file system extensively
 - Invoke application from shared file system
 - Read/write data from/to shared file system
- At 100K CPUs:
 - N Servers to manage 100K CPUs (1:256 ratio)
 - Don't trust the application I/O access patterns to behave 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
 - Use collective I/O primitives to make app logic simpler
 - Leverage all networks (Ethernet, Tree, and Torus) for high aggregate bandwidth

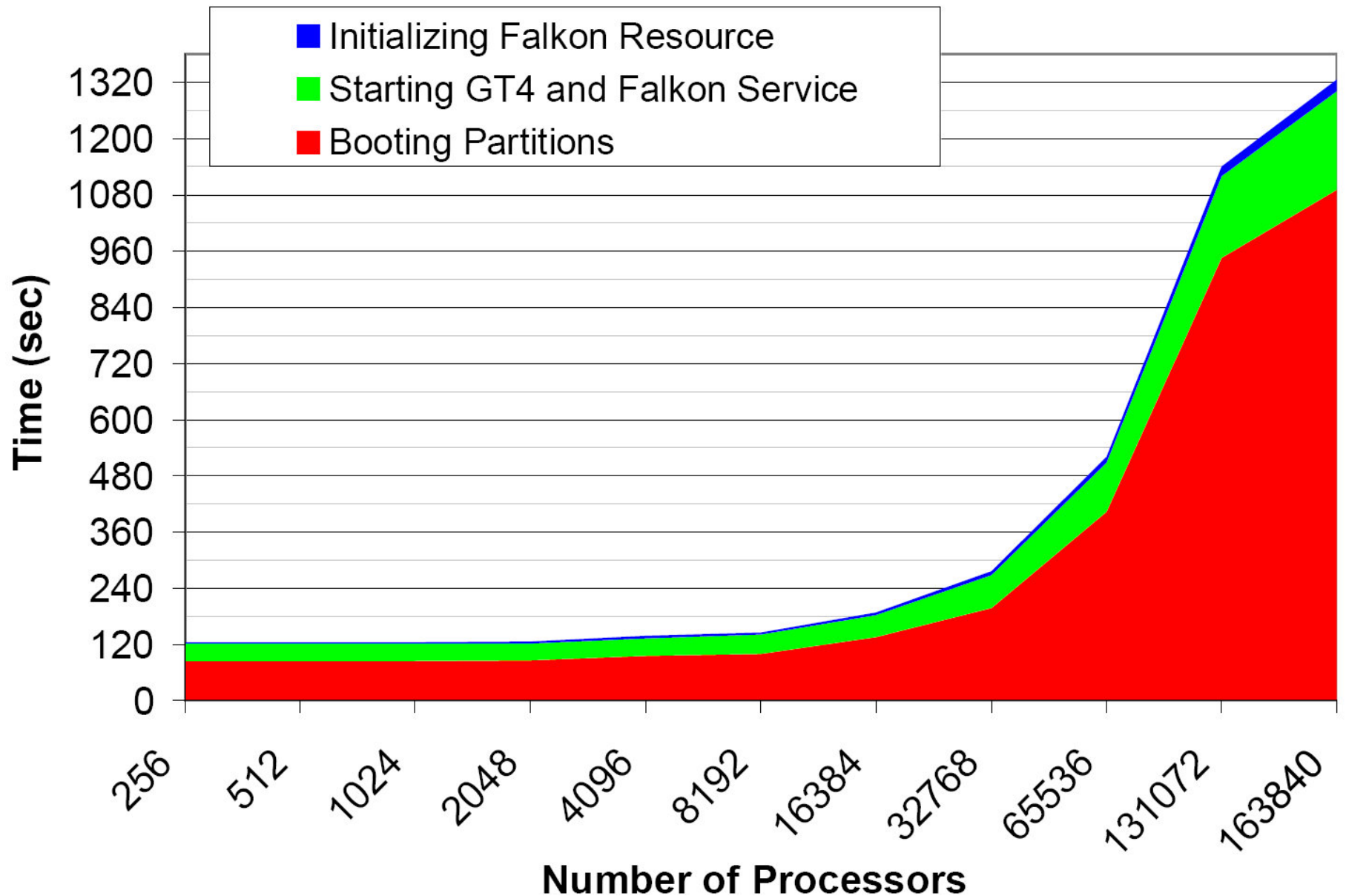
Distributed Falkon Architecture



Managing 160K CPUs



Falkon Bootstrapping



Falkon Monitoring

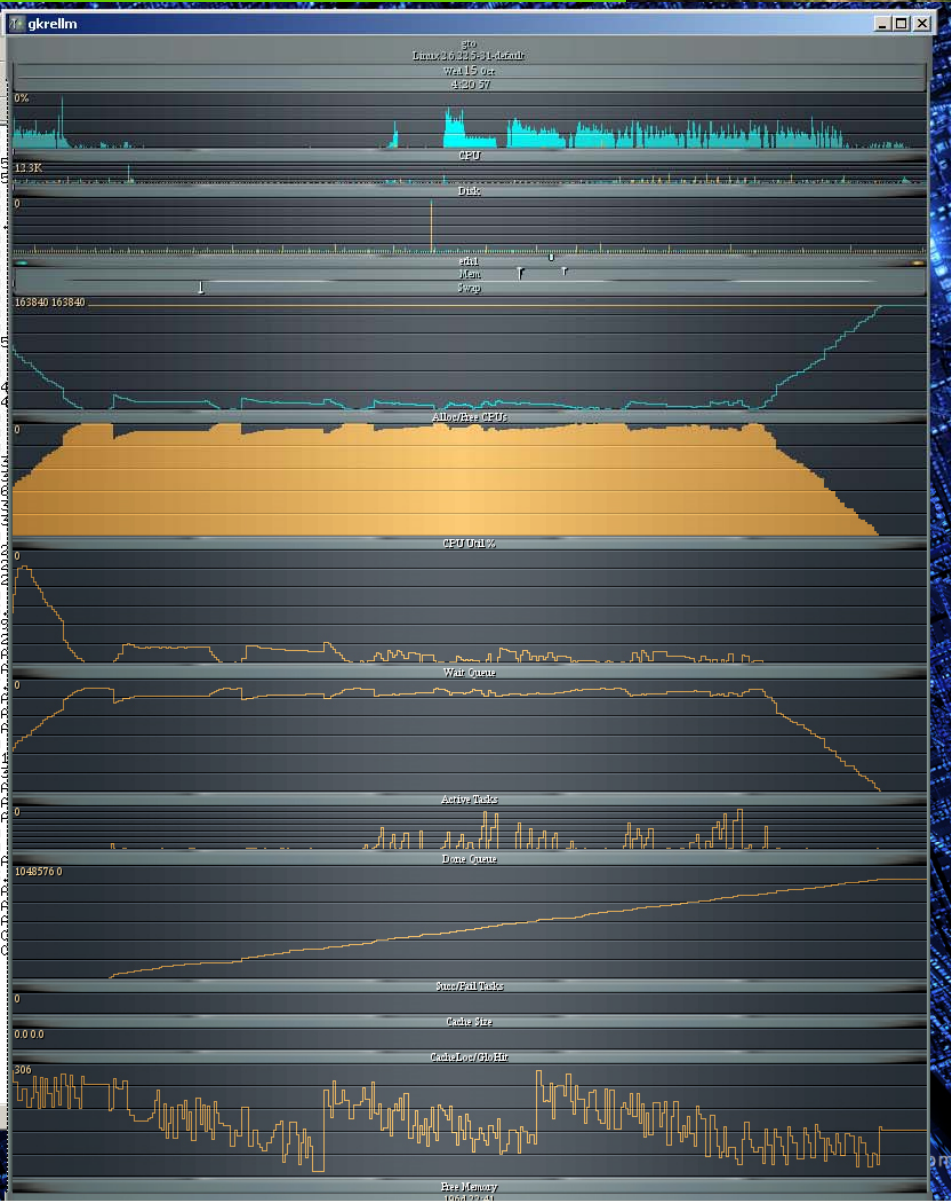
```
gto.ci.uchicago.edu (1) - SecureCRT
File Edit View Options Transfer Script Tools Help

gto.ci.uchicago.edu | gto.ci.uchicago.edu (1) | gto.ci.uchicago.edu (3) | gto.ci.uchicago.edu (2) | gto.ci.uchicago.edu (5) | gto.ci.uchicago.edu (4)

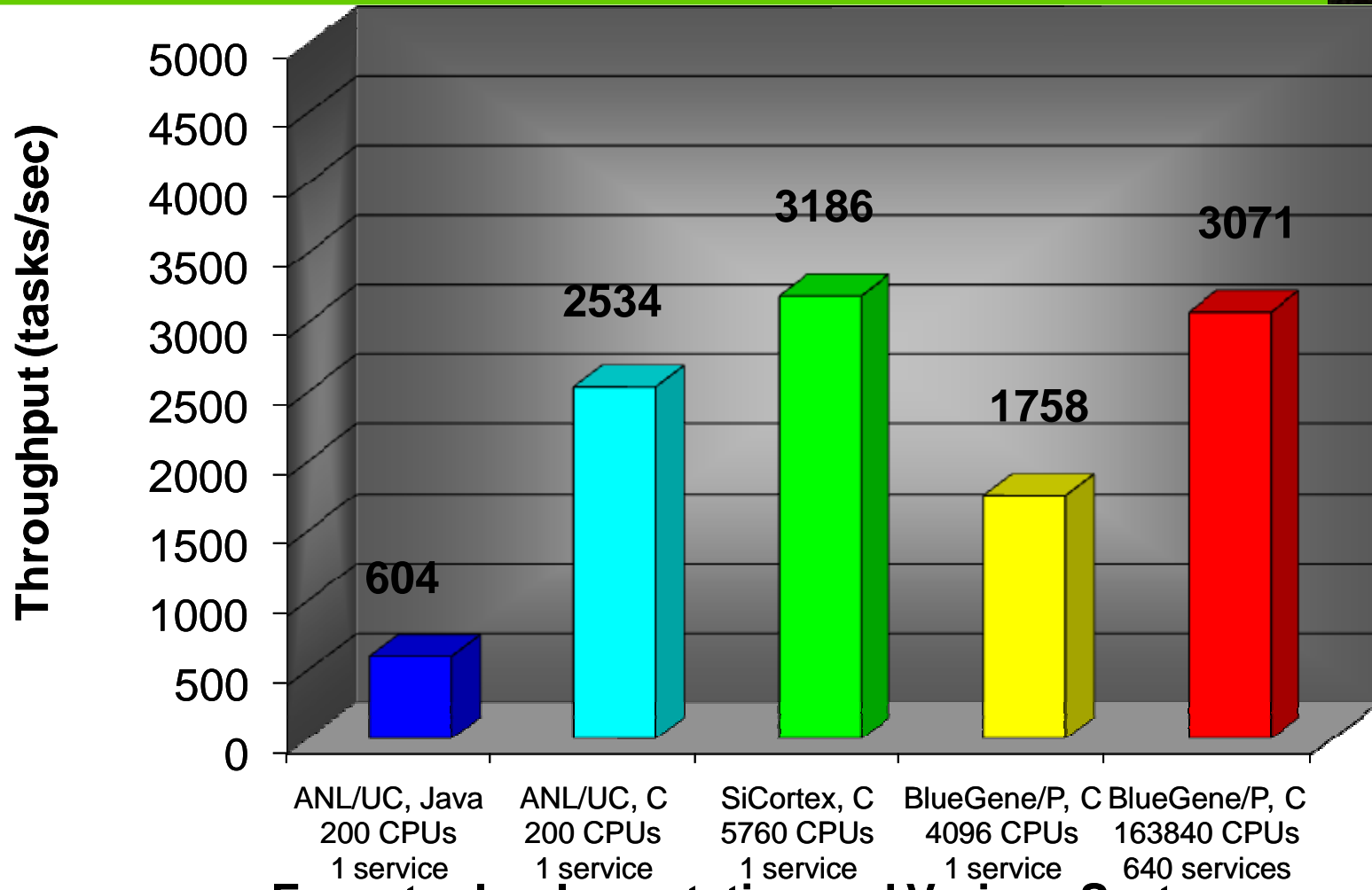
397,951 tasks+ 908675 tasks- 0 tasks-> 1048576 completed 86.66 tasks_tp 3246.03 aver_tp 2695.68 stdev_tp 3157.365 ETA
398,959 tasks+ 911918 tasks- 0 tasks-> 1048576 completed 86.97 tasks_tp 3217.26 aver_tp 2697.24 stdev_tp 3152.763 ETA
399,967 tasks+ 913940 tasks- 0 tasks-> 1048576 completed 87.16 tasks_tp 3205.95 aver_tp 2695.18 stdev_tp 3148.28 ETA
400,975 tasks+ 916630 tasks- 0 tasks-> 1048576 completed 87.42 tasks_tp 3268.65 aver_tp 2695.1 stdev_tp 3143.592 ETA
401,984 tasks+ 919282 tasks- 0 tasks-> 1048576 completed 87.67 tasks_tp 3230.95 aver_tp 2694.91 stdev_tp 3138.926 ETA
402,992 tasks+ 921616 tasks- 0 tasks-> 1048576 completed 87.89 tasks_tp 3215.48 aver_tp 2693.79 stdev_tp 3134.347 ETA
404,0 tasks+ 924266 tasks- 0 tasks-> 1048576 completed 88.14 tasks_tp 2628.97 aver_tp 2693.6 stdev_tp 3129.723 ETA
405,004 tasks+ 926864 tasks- 0 tasks-> 1048576 completed 88.39 tasks_tp 2587.65 aver_tp 2693.29 stdev_tp 3125.122 ETA
406,008 tasks+ 929627 tasks- 0 tasks-> 1048576 completed 88.66 tasks_tp 2751.99 aver_tp 2693.46 stdev_tp 3120.538 ETA
407,013 tasks+ 932059 tasks- 0 tasks-> 1048576 completed 88.69 tasks_tp 2422.31 aver_tp 2692.66 stdev_tp 3116.007 ETA
408,017 tasks+ 934610 tasks- 0 tasks-> 1048576 completed 89.13 tasks_tp 2540.84 aver_tp 2692.22 stdev_tp 3111.472 ETA
409,025 tasks+ 937191 tasks- 0 tasks-> 1048576 completed 89.36 tasks_tp 2439.04 aver_tp 2691.49 stdev_tp 3106.976 ETA
410,029 tasks+ 939803 tasks- 0 tasks-> 1048576 completed 89.57 tasks_tp 2122.51 aver_tp 2689.84 stdev_tp 3102.621 ETA
411,033 tasks+ 942446 tasks- 0 tasks-> 1048576 completed 89.79 tasks_tp 2279.89 aver_tp 2688.65 stdev_tp 3098.212 ETA
412,033 tasks+ 945106 tasks- 0 tasks-> 1048576 completed 90.0 tasks_tp 2218.13 aver_tp 2687.3 stdev_tp 3093.848 ETA
413,039 tasks+ 947825 tasks- 0 tasks-> 1048576 completed 90.21 tasks_tp 2171.31 aver_tp 2686.21 stdev_tp 3089.523 ETA
414,042 tasks+ 949129 tasks- 0 tasks-> 1048576 completed 90.42 tasks_tp 2234.06 aver_tp 2684.52 stdev_tp 3085.168 ETA
415,046 tasks+ 950429 tasks- 0 tasks-> 1048576 completed 90.63 tasks_tp 2047.81 aver_tp 2682.7 stdev_tp 3080.985 ETA
416,05 tasks+ 952338 tasks- 0 tasks-> 1048576 completed 90.84 tasks_tp 2144.42 aver_tp 2681.17 stdev_tp 3076.797 ETA
417,054 tasks+ 954861 tasks- 0 tasks-> 1048576 completed 91.05 tasks_tp 2214.14 aver_tp 2679.84 stdev_tp 3072.434 ETA
418,062 tasks+ 956045 tasks- 0 tasks-> 1048576 completed 91.23 tasks_tp 2067.46 aver_tp 2678.11 stdev_tp 3068.251 ETA
419,071 tasks+ 958742 tasks- 0 tasks-> 1048576 completed 91.43 tasks_tp 2090.06 aver_tp 2676.42 stdev_tp 3064.079 ETA
420,079 tasks+ 960480 tasks- 0 tasks-> 1048576 completed 91.6 tasks_tp 1724.21 aver_tp 2673.73 stdev_tp 3060.176 ETA
421,087 tasks+ 962005 tasks- 0 tasks-> 1048576 completed 91.8 tasks_tp 2106.13 aver_tp 2672.15 stdev_tp 3056.022 ETA
422,095 tasks+ 964687 tasks- 0 tasks-> 1048576 completed 92.0 tasks_tp 2075.4 aver_tp 2670.47 stdev_tp 3051.902 ETA
423,103 tasks+ 968960 tasks- 0 tasks-> 1048576 completed 92.12 tasks_tp 1248.02 aver_tp 2666.5 stdev_tp 3048.061 ETA
424,111 tasks+ 974461 tasks- 0 tasks-> 1048576 completed 92.93 tasks_tp 8425.6 aver_tp 2682.54 stdev_tp 3059.406 ETA
425,119 tasks+ 978213 tasks- 0 tasks-> 1048576 completed 93.29 tasks_tp 3722.22 aver_tp 2685.43 stdev_tp 3065.644 ETA
426,128 tasks+ 982243 tasks- 0 tasks-> 1048576 completed 93.48 tasks_tp 3011.9 aver_tp 2683.57 stdev_tp 3061.614 ETA
427,136 tasks+ 982449 tasks- 0 tasks-> 1048576 completed 93.73 tasks_tp 137.55 aver_tp 2682.19 stdev_tp 3047.938 ETA
428,144 tasks+ 983491 tasks- 0 tasks-> 1048576 completed 94.01 tasks_tp 144.76 aver_tp 2677.45 stdev_tp 3044.643 ETA
429,152 tasks+ 987623 tasks- 0 tasks-> 1048576 completed 94.23 tasks_tp 142.35 aver_tp 2681.51 stdev_tp 3041.441 ETA
430,16 tasks+ 992620 tasks- 0 tasks-> 1048576 completed 94.36 tasks_tp 142.35 aver_tp 2684.57 stdev_tp 3048.1 ETA
431,168 tasks+ 998290 tasks- 0 tasks-> 1048576 completed 94.92 tasks_tp 0.0 aver_tp 2687.6 stdev_tp 3047.182 ETA
432,176 tasks+ 997217 tasks- 0 tasks-> 1048576 completed 95.1 tasks_tp 1941.47 aver_tp 2685.57 stdev_tp 3043.076 ETA
433,184 tasks+ 994294 tasks- 0 tasks-> 1048576 completed 95.57 tasks_tp 1379.57 aver_tp 2681.53 stdev_tp 3041.282 ETA
434,193 tasks+ 997770 tasks- 0 tasks-> 1048576 completed 95.99 tasks_tp 136.77 aver_tp 2686.77 stdev_tp 3040.338 ETA
435,201 tasks+ 999076 tasks- 0 tasks-> 1048576 completed 96.33 tasks_tp 136.26 aver_tp 2685.26 stdev_tp 3039.495 ETA
436,209 tasks+ 1011812 tasks- 0 tasks-> 1048576 completed 96.49 tasks_tp 1327.58 aver_tp 2686.38 stdev_tp 3031.597 ETA
437,217 tasks+ 1011812 tasks- 0 tasks-> 1048576 completed 96.84 tasks_tp 3585.56 aver_tp 2690.71 stdev_tp 3027.863 ETA
438,225 tasks+ 1015295 tasks- 0 tasks-> 1048576 completed 97.27 tasks_tp 4580.6 aver_tp 2695.0 stdev_tp 3025.337 ETA
439,233 tasks+ 1019985 tasks- 0 tasks-> 1048576 completed 97.77 tasks_tp 2266.87 aver_tp 2686.15 stdev_tp 2995.489 ETA
440,241 tasks+ 1024983 tasks- 0 tasks-> 1048576 completed 98.11 tasks_tp 2266.18 aver_tp 2686.15 stdev_tp 2991.596 ETA
441,249 tasks+ 1030223 tasks- 0 tasks-> 1048576 completed 98.23 tasks_tp 4472.22 aver_tp 2694.02 stdev_tp 3018.682 ETA
442,257 tasks+ 1032955 tasks- 0 tasks-> 1048576 completed 98.47 tasks_tp 2319.44 aver_tp 2693.04 stdev_tp 3012.127 ETA
443,265 tasks+ 1036258 tasks- 0 tasks-> 1048576 completed 98.83 tasks_tp 3662.7 aver_tp 2695.53 stdev_tp 3009.572 ETA
444,273 tasks+ 1036258 tasks- 0 tasks-> 1048576 completed 99.07 tasks_tp 2864.48 aver_tp 2695.24 stdev_tp 3004.628 ETA
445,281 tasks+ 1041225 tasks- 0 tasks-> 1048576 completed 99.25 tasks_tp 1929.56 aver_tp 2693.24 stdev_tp 3000.948 ETA
446,289 tasks+ 1041225 tasks- 0 tasks-> 1048576 completed 99.33 tasks_tp 395.83 aver_tp 2687.24 stdev_tp 2999.32 ETA
447,297 tasks+ 1046203 tasks- 0 tasks-> 1048576 completed 99.51 tasks_tp 2266.87 aver_tp 2686.15 stdev_tp 2995.489 ETA
448,305 tasks+ 1046203 tasks- 0 tasks-> 1048576 completed 99.77 tasks_tp 2688.18 aver_tp 2686.15 stdev_tp 2991.596 ETA
449,314 tasks+ 1043989 tasks- 0 tasks-> 1048576 completed 100.0 tasks_tp 2354.17 aver_tp 2685.29 stdev_tp 2987.766 ETA
450,322 tasks+ 1048576 tasks- 0 tasks-> 1048576 completed 100.0 tasks_tp 0.0 aver_tp 2678.35 stdev_tp 2987.016 ETA
451,331 tasks+ 1048576 tasks- 0 tasks-> 1048576 completed 100.0 tasks_tp 0.0 aver_tp 2671.45 stdev_tp 2986.253 ETA
452,339 tasks+ 1048576 tasks- 0 tasks-> 1048576 completed 100.0 tasks_tp 0.0 aver_tp 2671.45 stdev_tp 2986.253 ETA
453,347 tasks+ 1048576 tasks- 0 tasks-> 1048576 completed 100.0 tasks_tp 0.0 aver_tp 2671.45 stdev_tp 2986.253 ETA

1048576 tasks completed in 453.505 sec
Successful tasks: 1048576
Failed tasks: 0
Notification Errors: 0
Overall Throughput (tasks/sec): 2312.16
Overall Throughput Standard Deviation: 2986.253
waiting to destroy all resources...
ShutdownHook triggered successfully!
iraicu@gto:~/falkon>
```

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

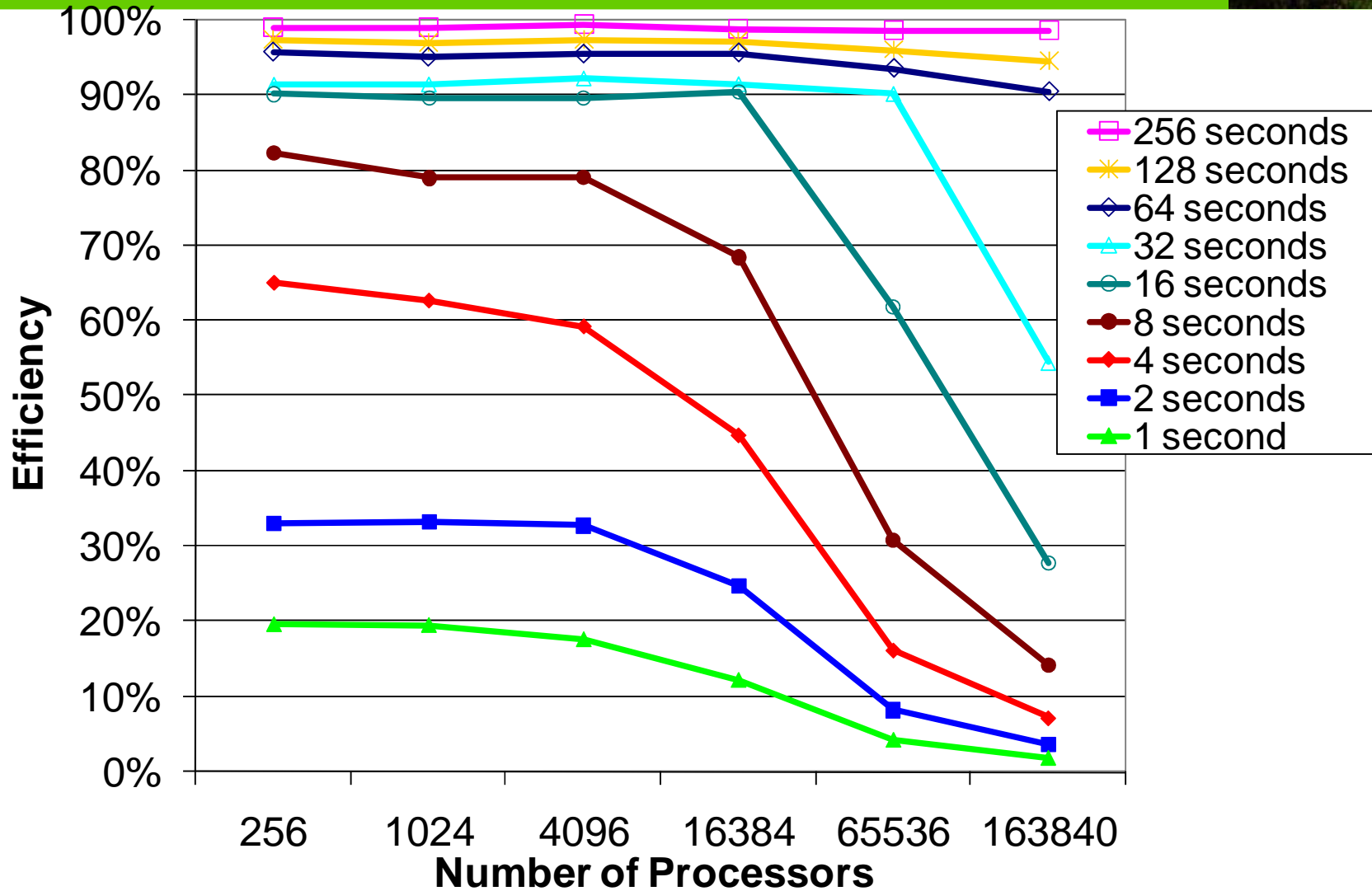


Dispatch Throughput



Executor Implementation and Various Systems

Efficiency

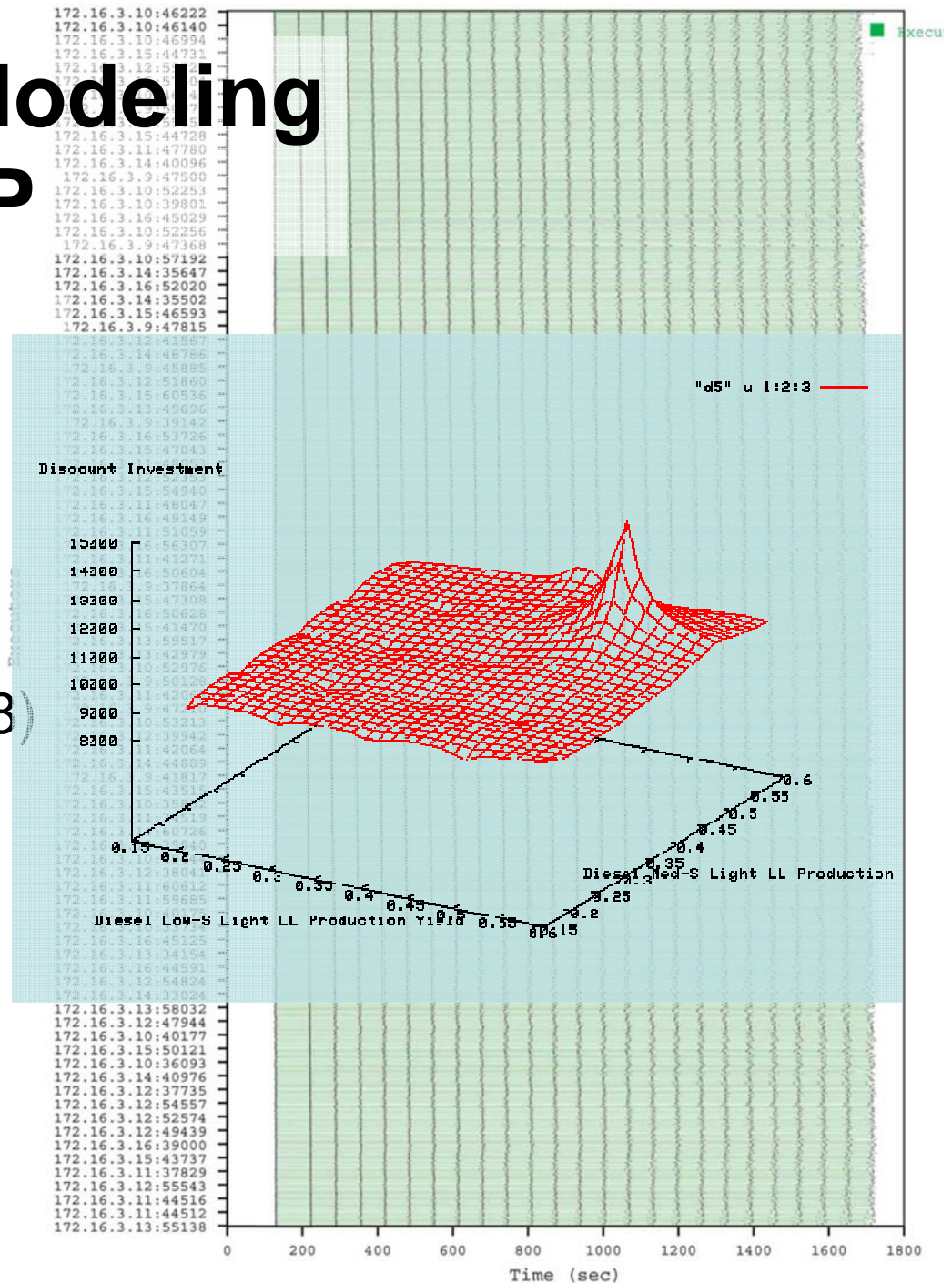


MARS Economic Modeling on IBM BG/P

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



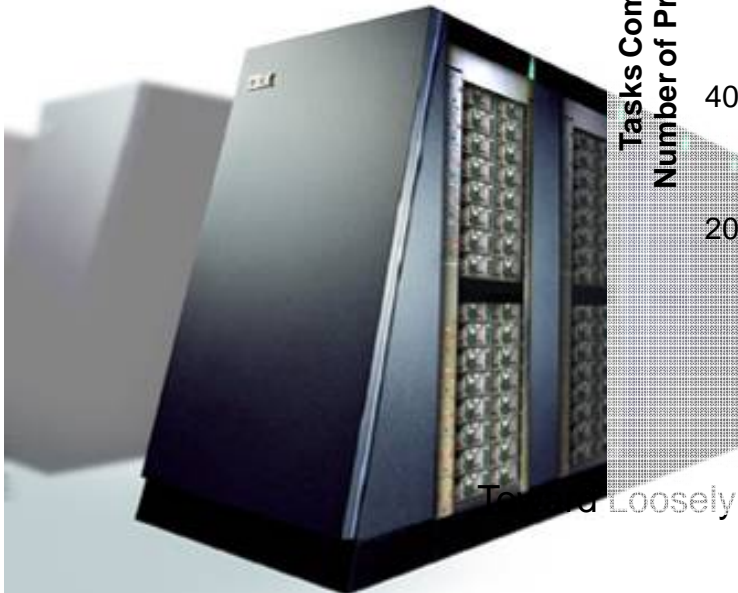
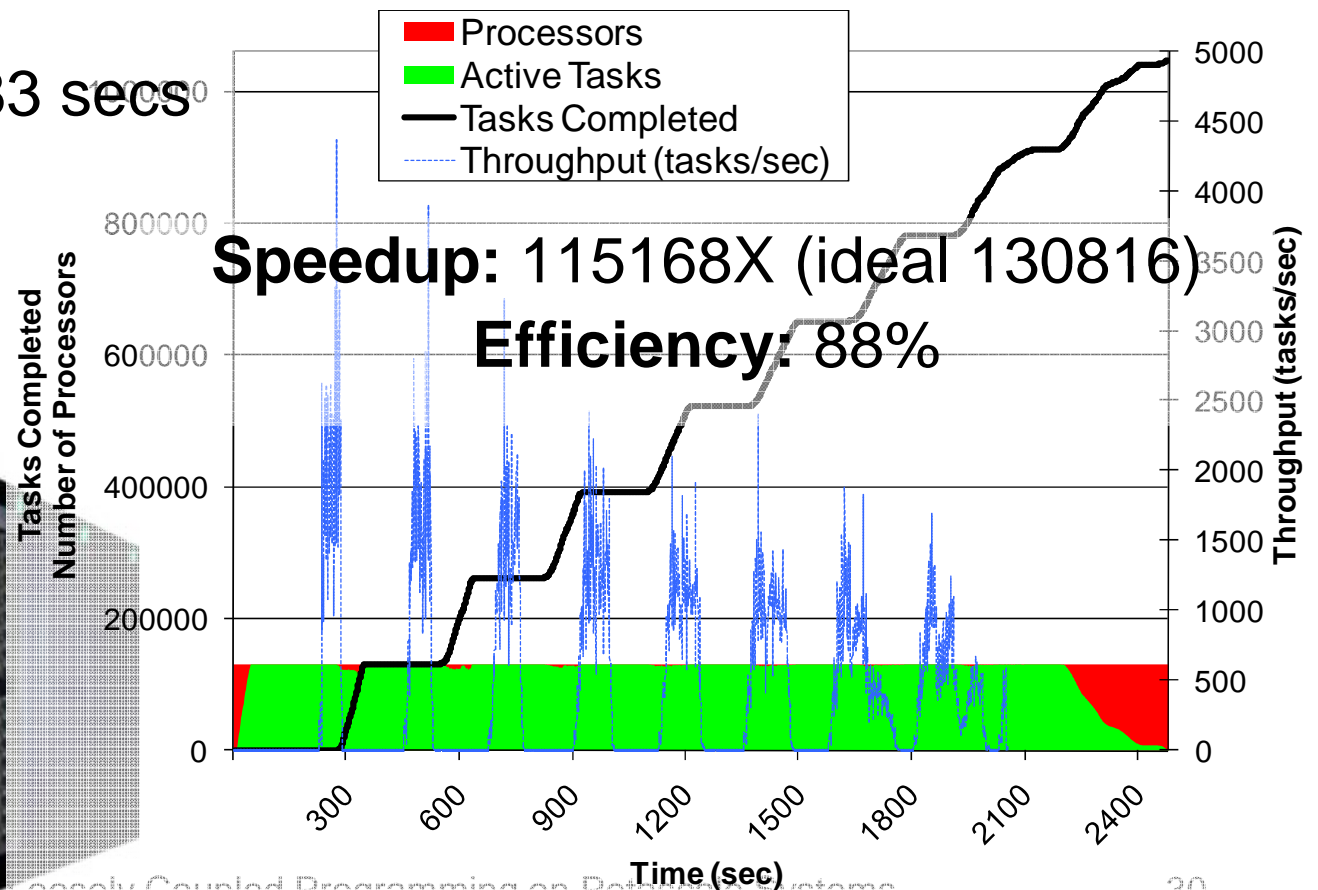
Couplec



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



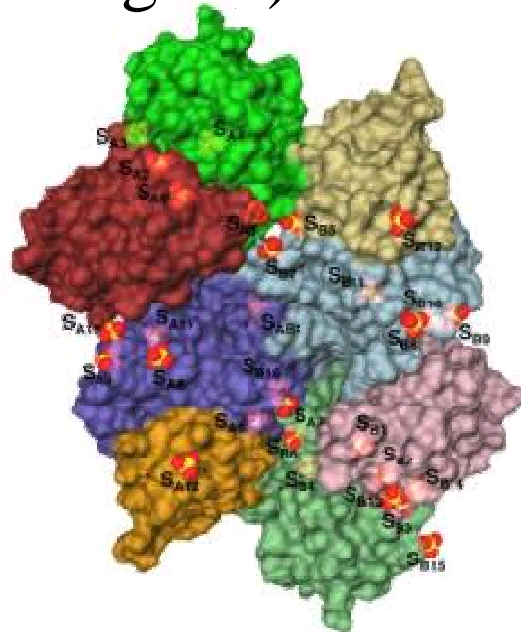
- CPU Cores: 130816
- Tasks: 1048576
- Elapsed time: 2483 secs
- CPU Years: 9.3



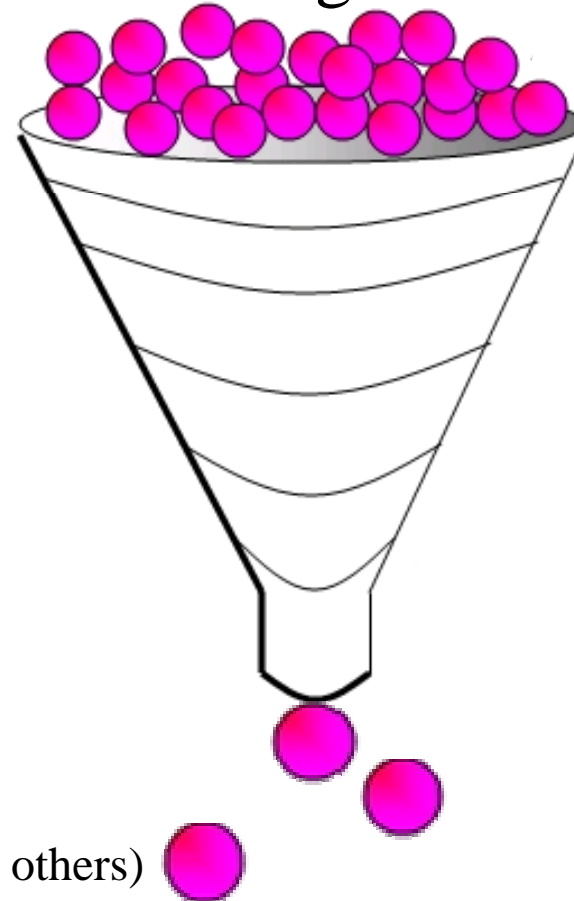
Many Many Tasks: Identifying Potential Drug Targets



Protein
target(s) x

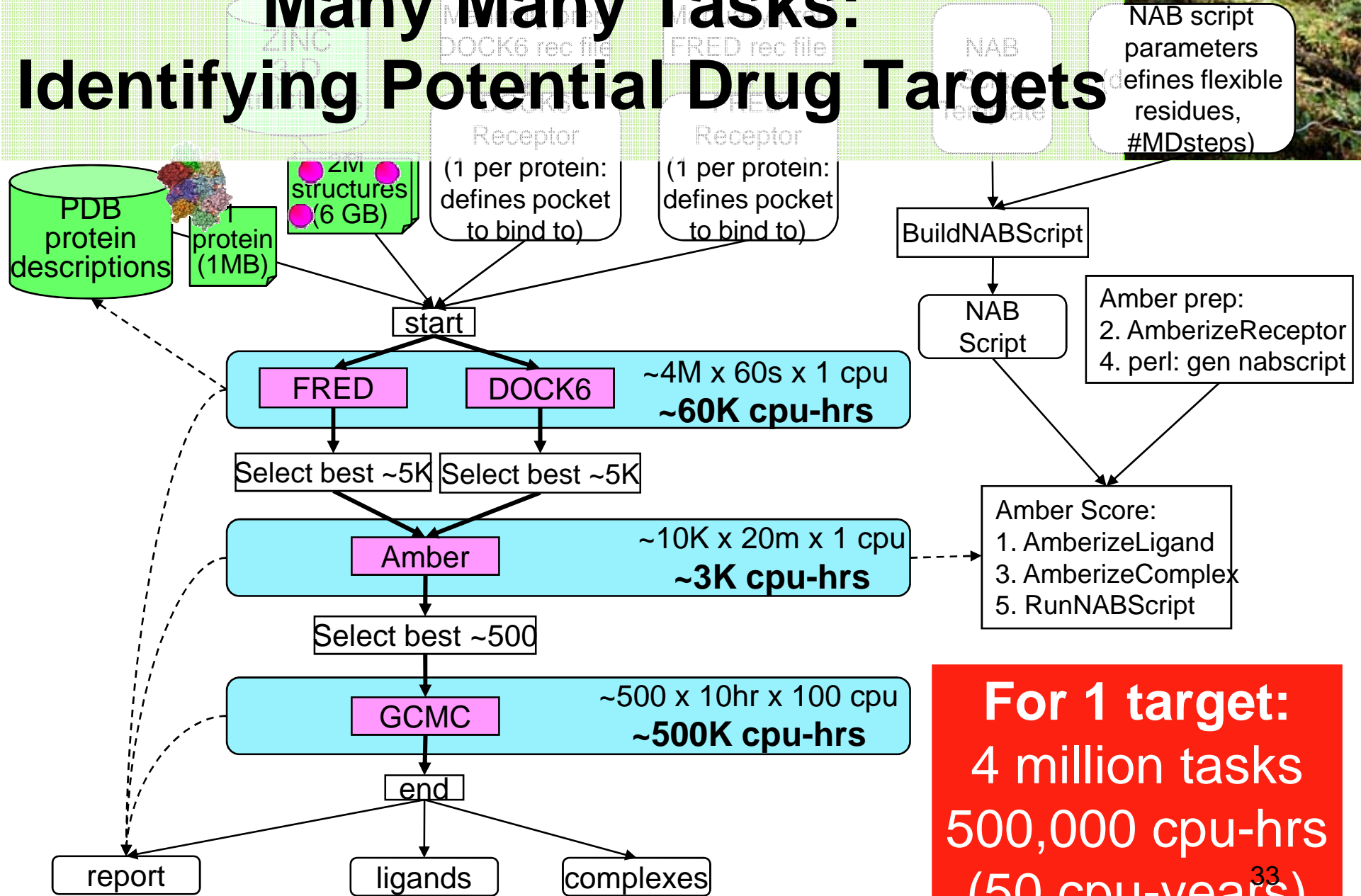


2M+ ligands



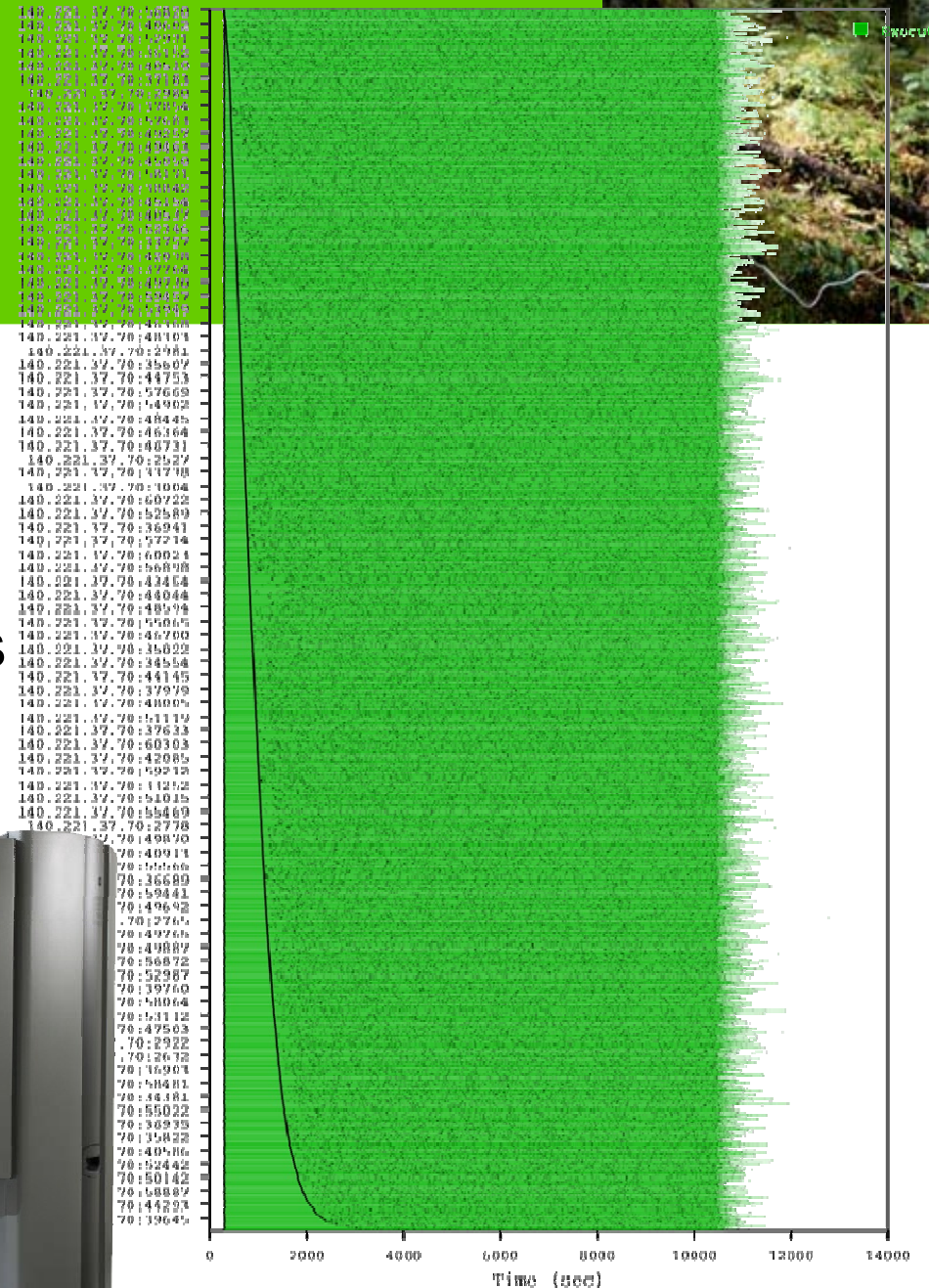
(Mike Kubal, Benoit Roux, and others)

Many Many Tasks: Identifying Potential Drug Targets



DOCK on SiCortex

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



DOCK on the BG/P



CPU cores: 118784

Tasks: 934803

Elapsed time: 2.01 hours

Compute time: 21.43 CPU years

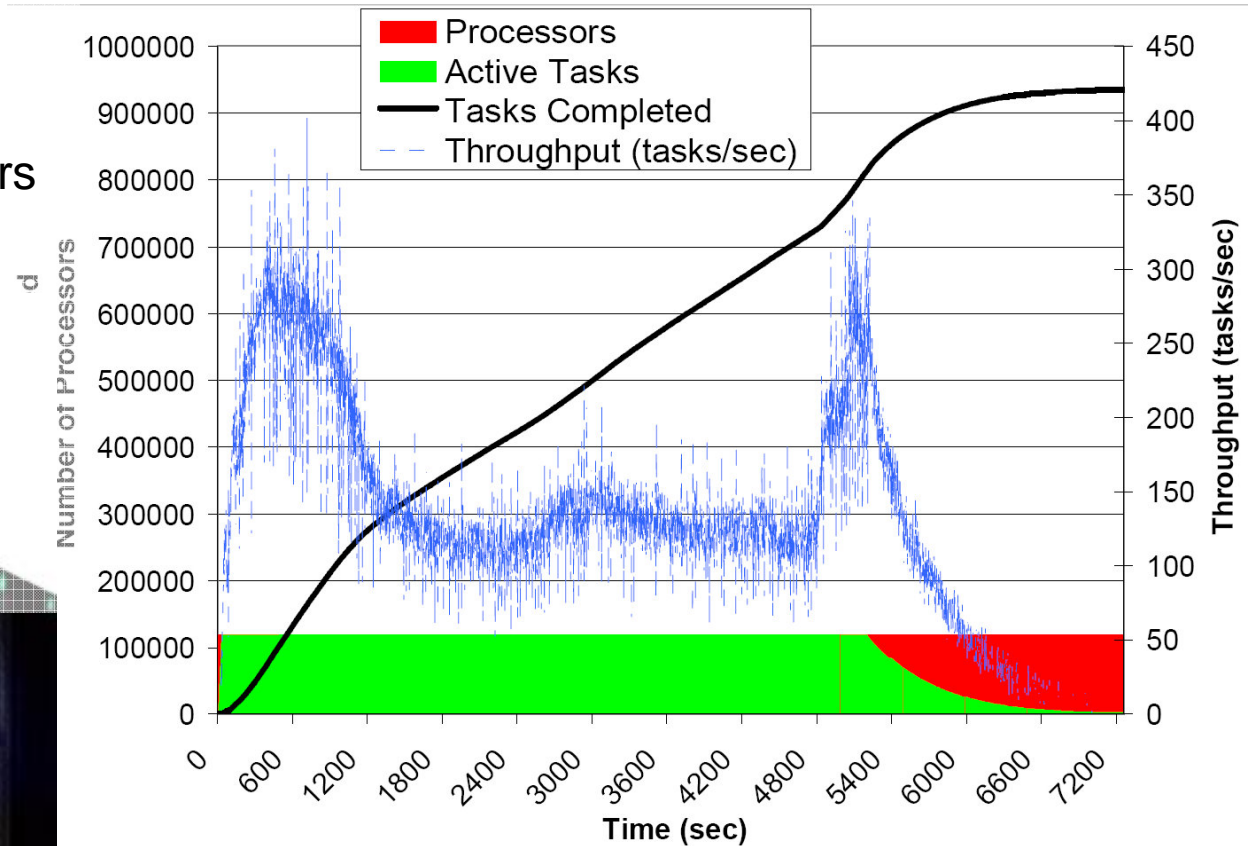
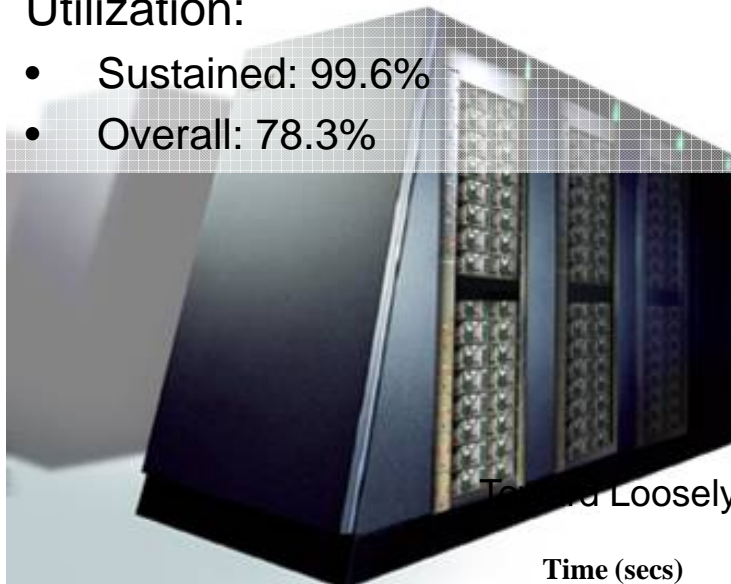
Average task time: 667 sec

Relative Efficiency: 99.7%

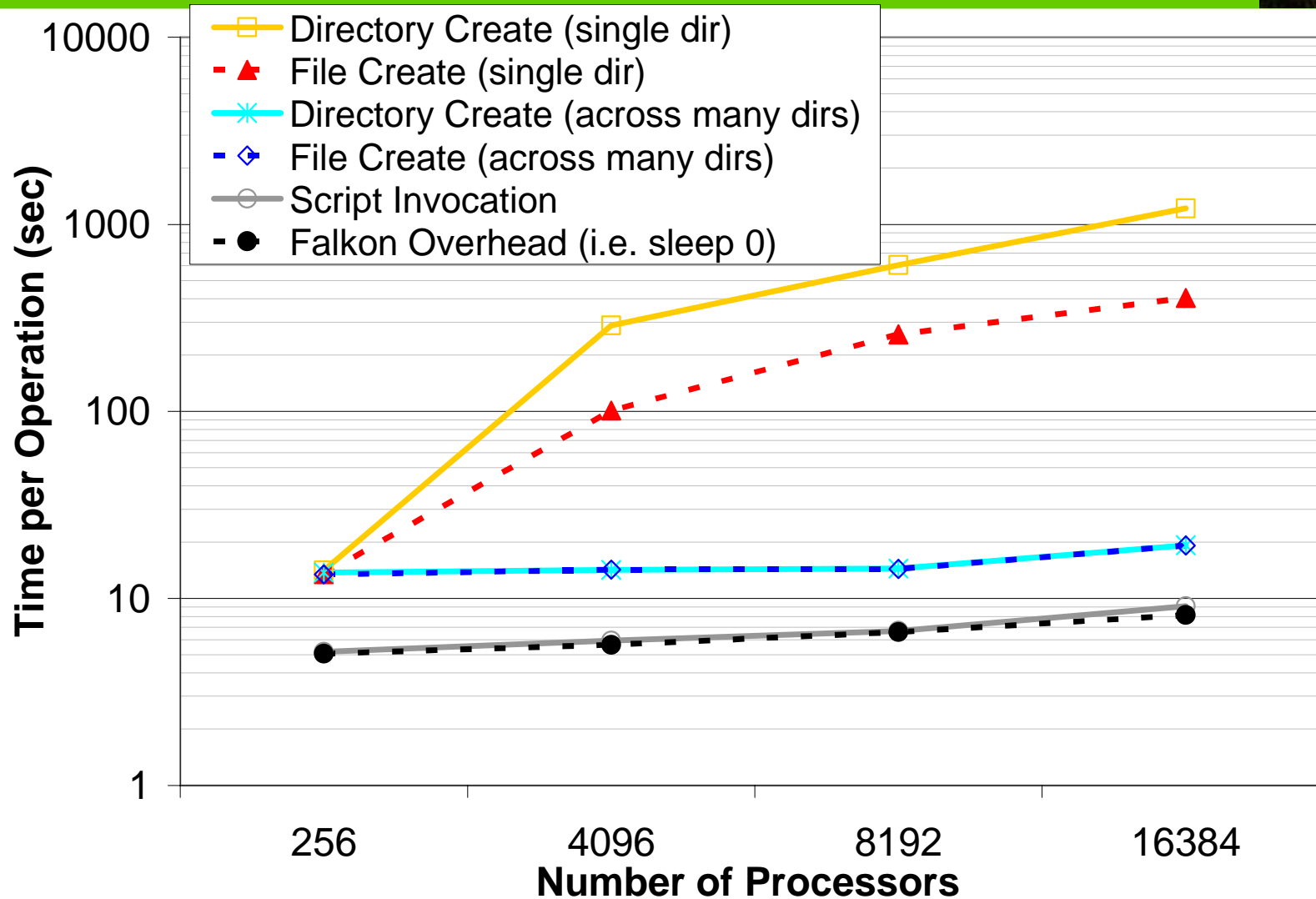
(from 16 to 32 racks)

Utilization:

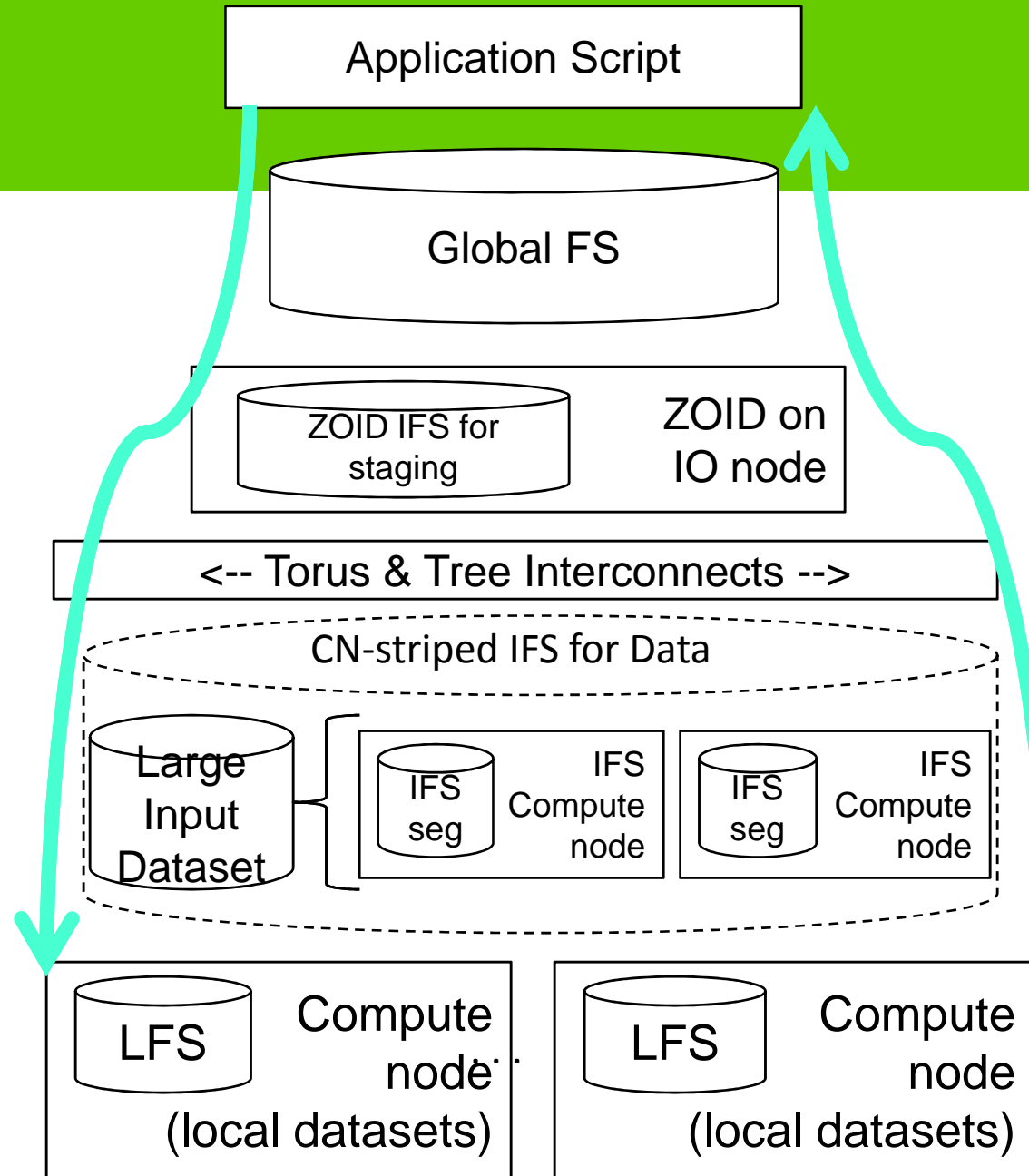
- Sustained: 99.6%
- Overall: 78.3%



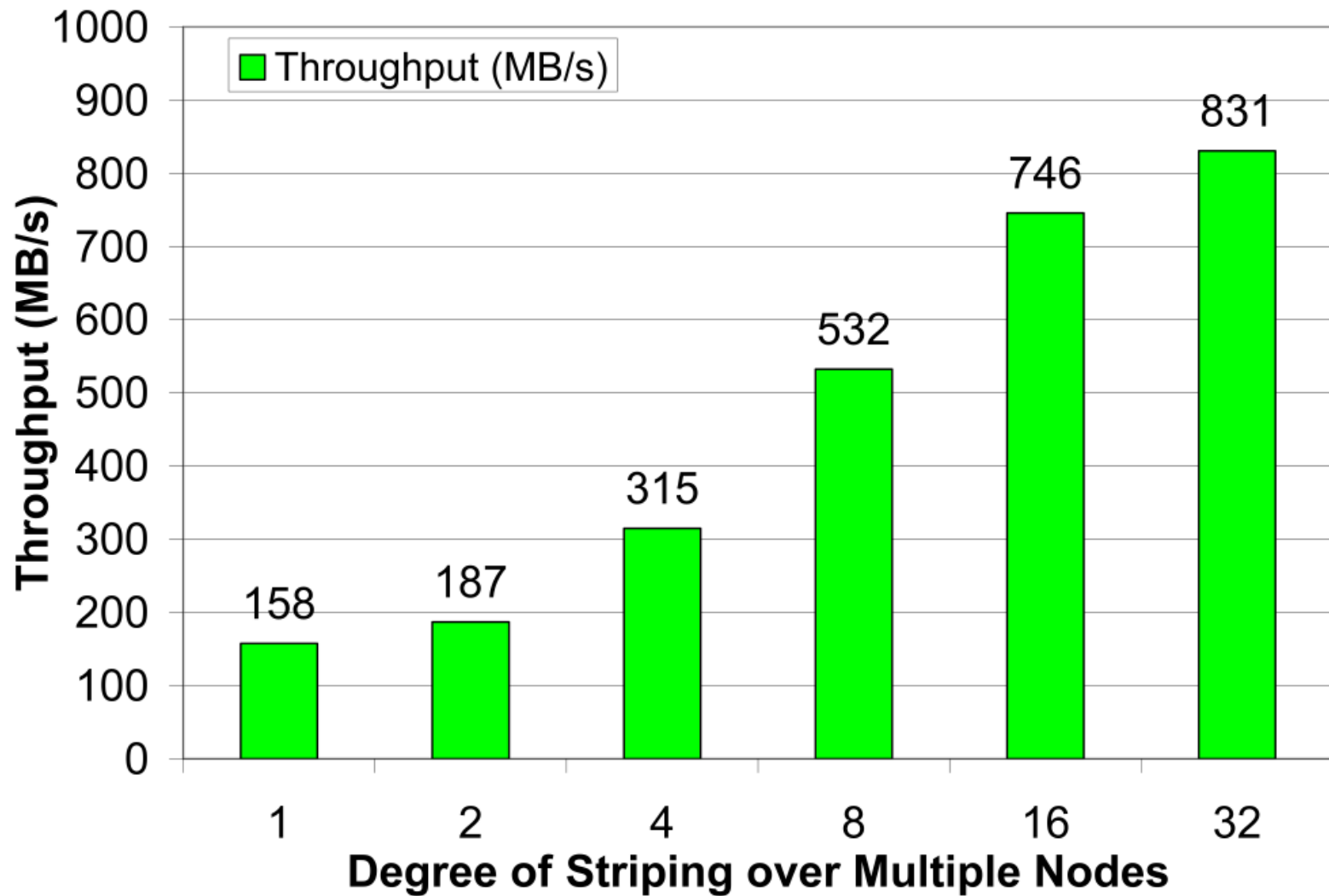
Costs to interact with GPFS



LCP Collective IO Model

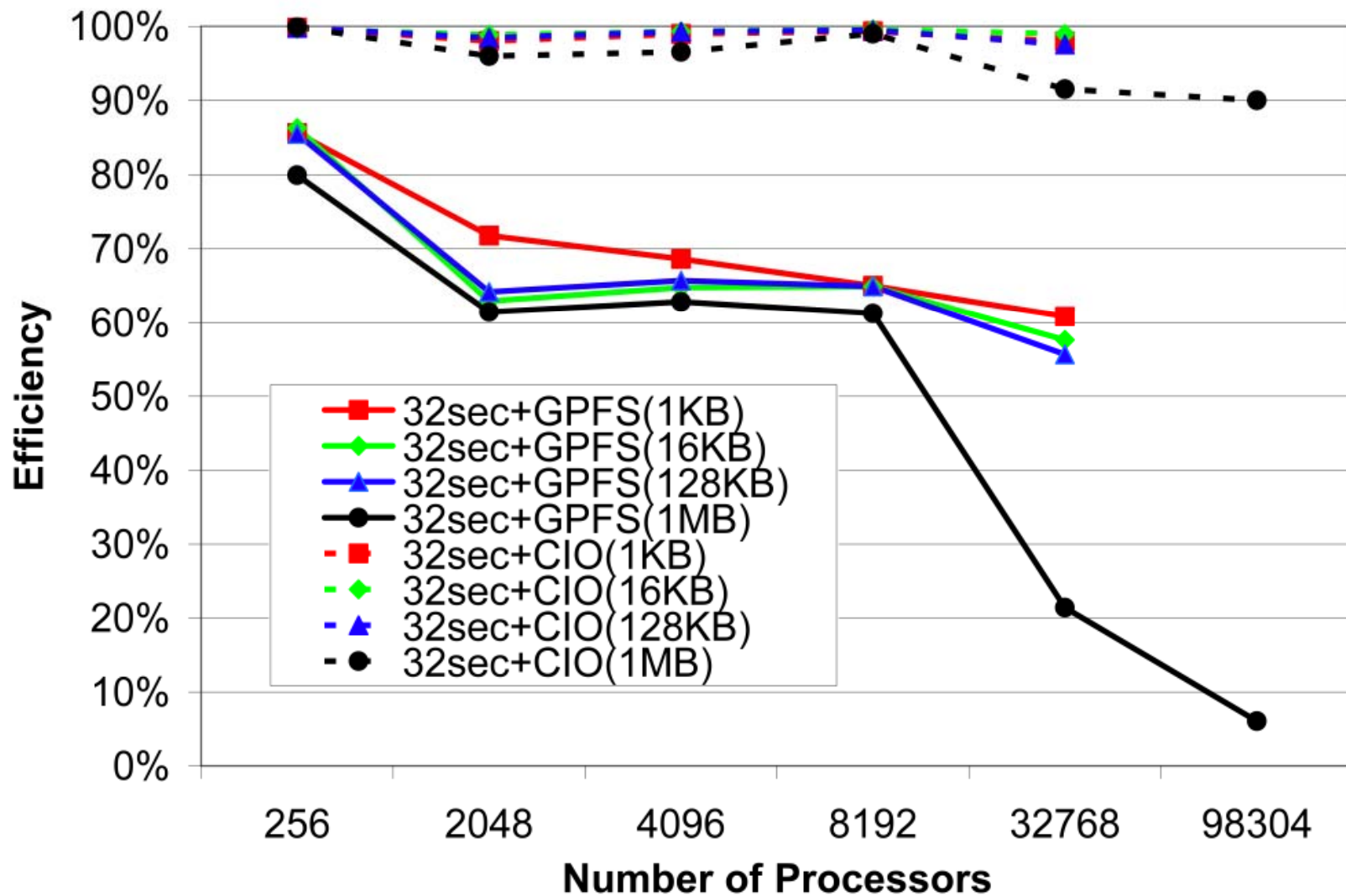


Read performance from IFS

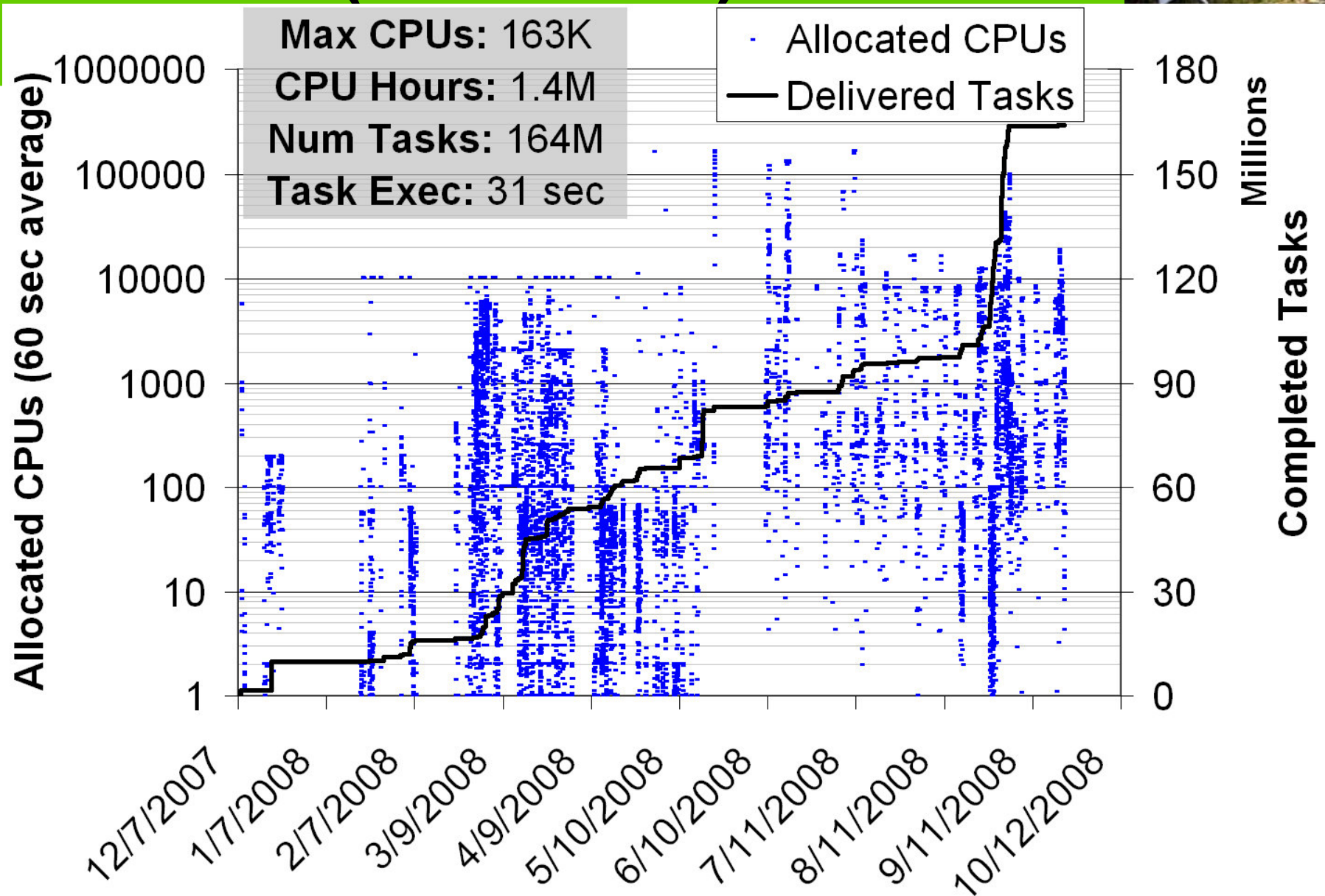
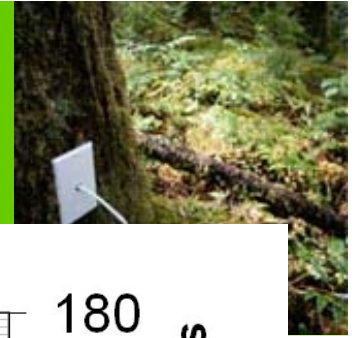


Write Performance

CIO vs. GPFS efficiency



Falkon Activity History (10 months)



PART IV



Conclusions and Future Work

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 a new class of applications: MTC
- Proved that MTC applications can be executed efficiently on supercomputers at full scale
- Extended Falkon by distributing the dispatcher/scheduler
- Falkon installed and configured on the BG/P for anyone to use

Future Work: Other Supercomputers

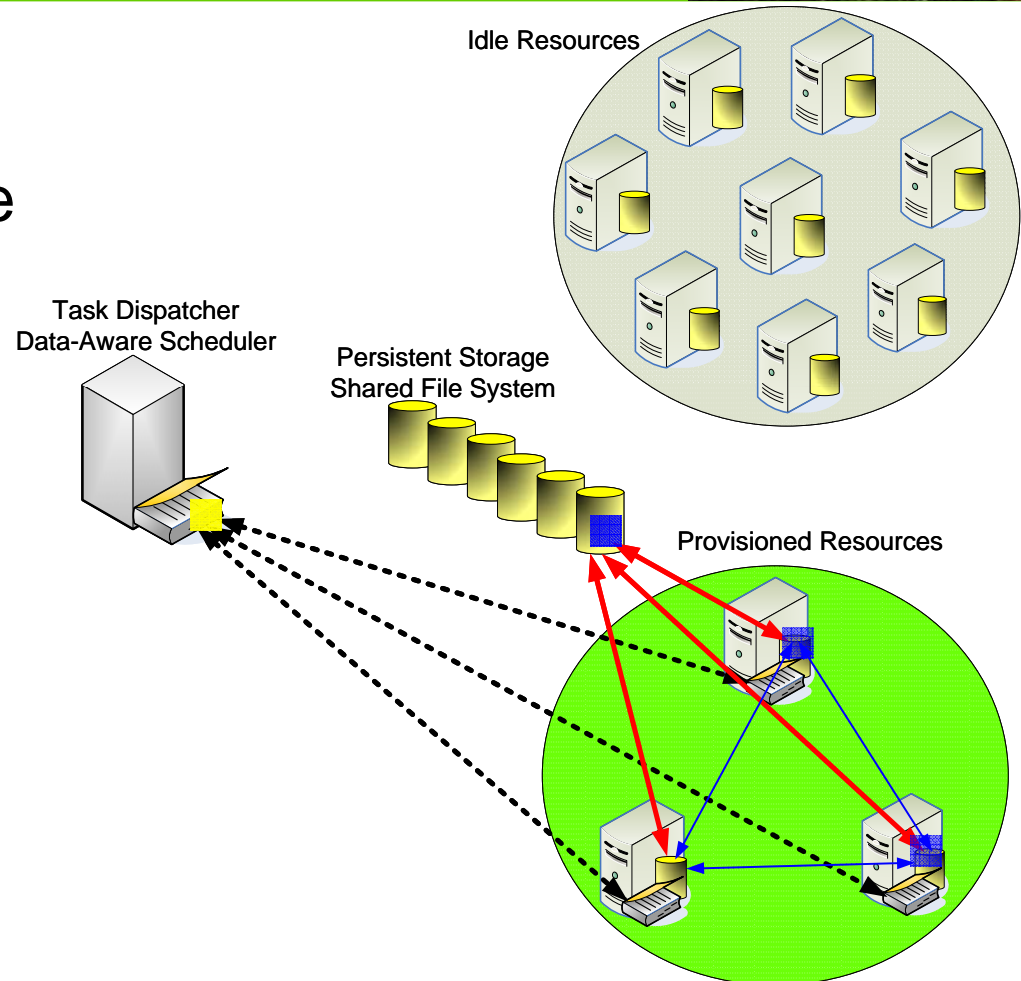


- Ranger: Sun Constellation
 - Basic mechanisms in place, and have started testing
- Jaguar: Cray
 - Plan to get accounts on machine as soon as its online
- Future Blue Gene machines (Q?)
 - Discussions underway between IBM, ANL and UChicago

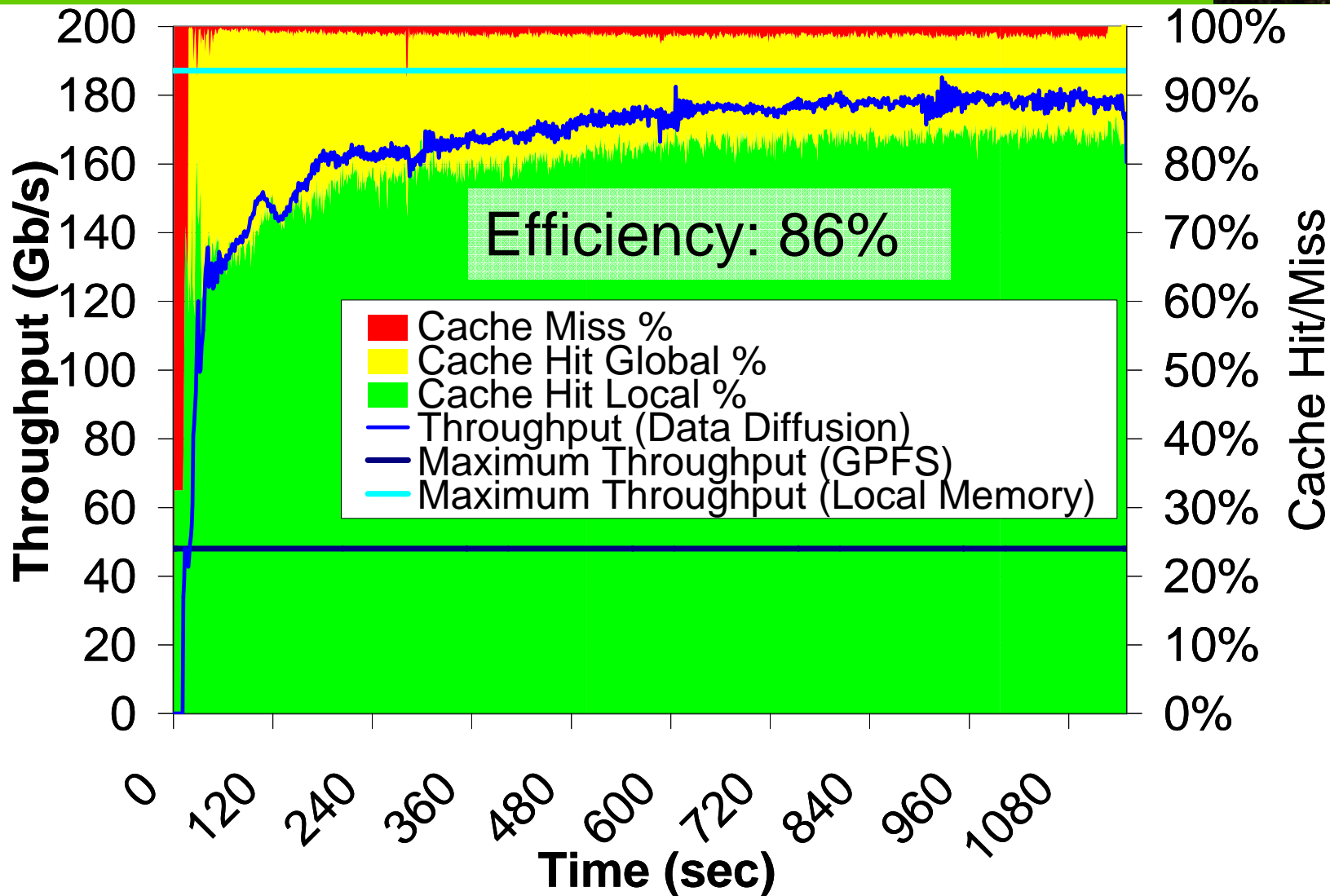
Future Work: Data Diffusion



- 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



All-Pairs Workload 1000x1000 on 4K emulated CPUs



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>
- 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

Main Page - MegajobBOF - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://gridfarm007.ucs.indiana.edu/megajobBOF/index.php/Main_Page

Most Visited iGoogle Ioan Raicu's Web Site MTAGS08: Workshop o... Dashboard - Google An... Incubator/Falkon - Glo... Outreach/SC2008 - Glo... CiteSeerX Google Scholar LinkedIn

Incubator/Falkon - Globus Ioan Raicu's Web Site Main Page - MegajobBOF

iraicu my talk preferences my watchlist my contributions log out

article discussion edit history move unwatch

bof information

- Main Page
- Presenters and Abstracts
- logistics
- presenters
- organizers

wiki tools

- Current events
- Recent changes
- Help

search

Go Search

toolbox

- What links here
- Related changes
- Upload file
- Special pages
- Printable version

Main Page

Megajobs: How to Run One Million Jobs [edit]

- What:** Birds-of-a-Feather Session at Supercomputing 2008, Austin Texas
- Date:** Tuesday, November 18th, 2008
- Time:** 05:30PM - 07:00PM
- Location:** Room 13A/13B
- Primary Session Leader:**
 - Marlon Pierce (Indiana University)
- Secondary Session Leader:**
 - Ioan Raicu (University of Chicago)
 - Ruth Pordes (Fermi National Laboratory)
 - John McGee (Renaissance Computing Institute)
 - Dick Repasky (Indiana University)

As large systems surpass 200K CPU cores and as applications increase in complexity, more scientists need to run thousands to millions of closely related jobs that are associated with individual projects. Scientists seek convenient means to specify and manage many jobs, arranging inputs, aggregating outputs, identifying successful and failed jobs and repairing failures. System administrators seek methods to process extraordinary numbers of jobs for multiple users without overwhelming queuing systems or disrupting fair-share usage policies. Under development are a new generation of queuing and scheduling systems and multi-level schedulers for use with existing queuing and scheduling systems, schedulers designed to handle millions of jobs. This Birds-of-feather session provides a venue for the exchange of information about processing large numbers of jobs. Short presentations of an invited sample of projects will be followed by discussion.

We are currently soliciting participation in the "Megajobs" BOF. We are looking for short, piquant presentations (5-10 minutes) from people who have worked on this problem or have a problem like this that needs to be worked on. If you are interested, please send a brief title and abstract (250 words) to [Marlon Pierce](#) by October 27th, 2008. Please feel free to contact us if you have questions.

For the latest information hosted by SC08, see <http://scyourway.nacse.org/conference/view/bof118>. The Megajobs BOF handout can also be found [here](#).

Related activities at SC08, that might be of interest to BOF attendees are:

- [Grid Computing Environments \(GCE\)](#)
- [Workshop on Many-Task Computing on Grids and Supercomputers \(MTAGS\)](#)

Done