

# **The Quest for Scalable Support of Data-Intensive Workloads in Distributed Systems**

**Ioan Raicu**

**Distributed Systems Laboratory  
Computational Institute  
University of Chicago**

**In Collaboration with:**

**Ian Foster, Yong Zhao, Douglas Thain, Amitabh Chaudhary, Philip Little, Christopher Moretti**

**ACM International Symposium on High Performance Distributed Computing (HPDC)  
June 13<sup>th</sup>, 2009**

digitalblasphemy.com

# State of the Art: Storage Systems

- Segregated storage and compute
  - NFS, GPFS, PVFS, Lustre
  - Batch-scheduled systems: Clusters, Grids, and Supercomputers
  - Programming paradigm: HPC, MTC, and HTC
- Co-located storage and compute
  - HDFS, GFS
  - Data centers at Google, Yahoo, and others
  - Programming paradigm: MapReduce
  - Others from academia: Sector, MosaStore, Chirp



# State of the Art: Storage Systems

- Segregated storage and compute

- NFS, GPFS, PVFS, Lustre

- Batch-scheduled

- Supercomputers

- Programming pa

- Located storage

- Network Link(s)

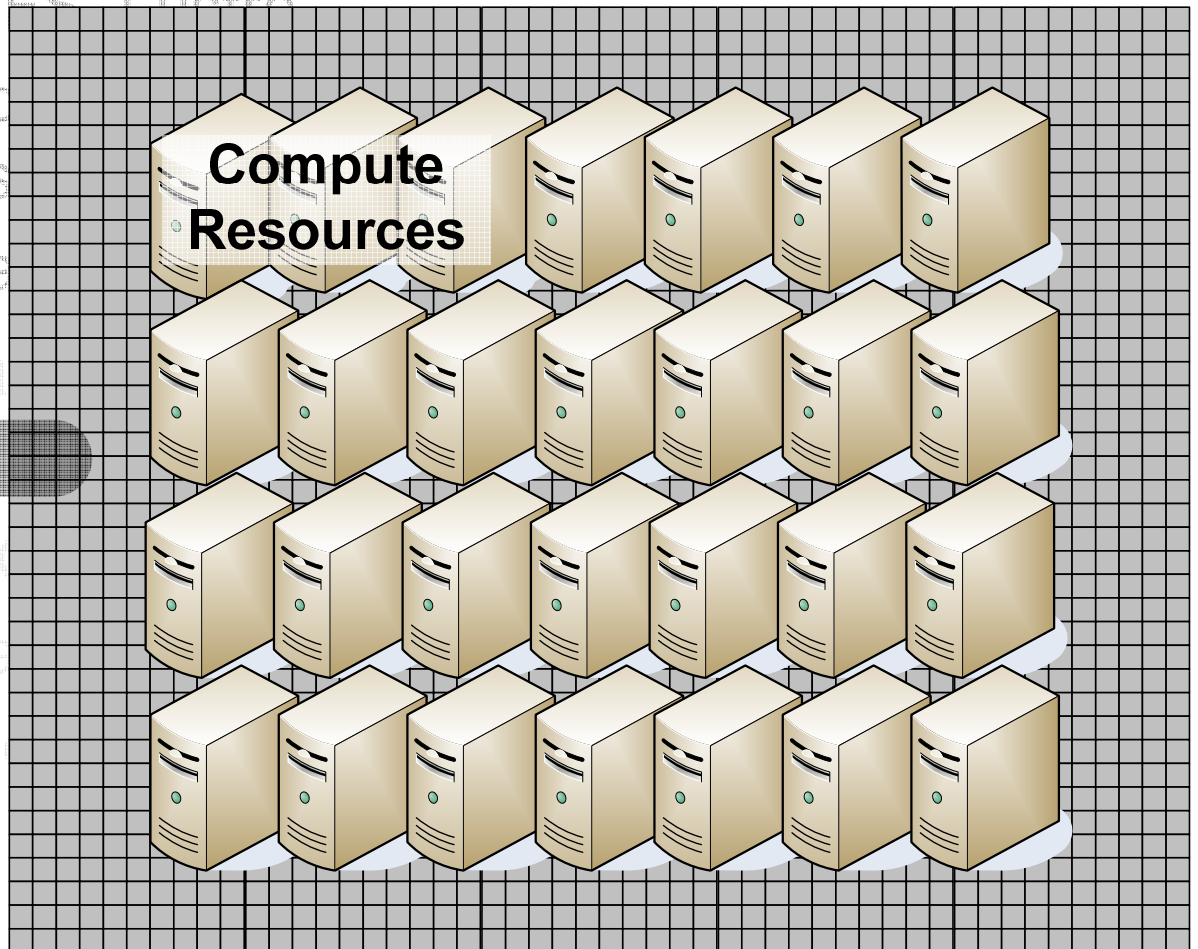
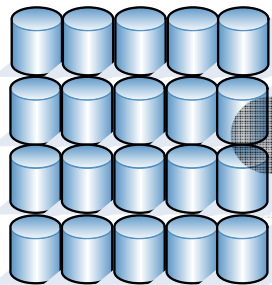
- Data centers at

- Programming pa

- Others from aca

**Network  
Fabric**

**Compute  
Resources**



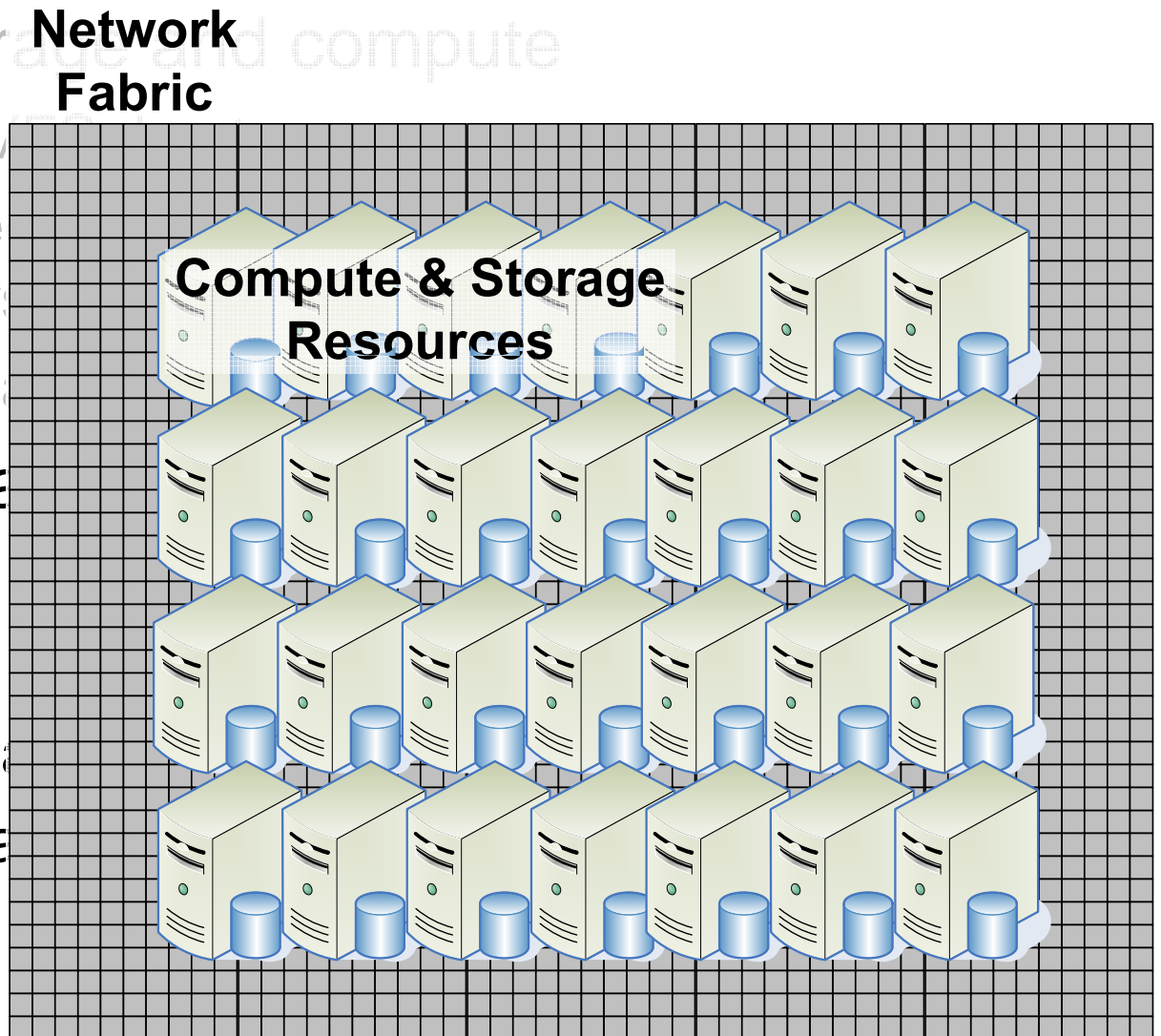
# State of the Art: Storage Systems

- Segregated storage and compute
  - NFS, GPFS, PVFS, Lustre
  - Batch-scheduled systems: Clusters, Grids, and Supercomputers
  - Programming paradigm: HPC, MTC, and HTC
- Co-located storage and compute
  - HDFS, GFS
  - Data centers at Google, Yahoo, and others
  - Programming paradigm: MapReduce
  - Others from academia: Sector, MosaStore, Chirp



# State of the Art: Storage Systems

- Segregated storage and compute
  - NFS, GPFS, PV
  - Batch-schedule Supercomputers
  - Programming p
- Co-located storage
  - HDFS, GFS
  - Data centers at
  - Programming p
  - Others from acc



# Growing Storage/Compute Gap

- Local Disk:

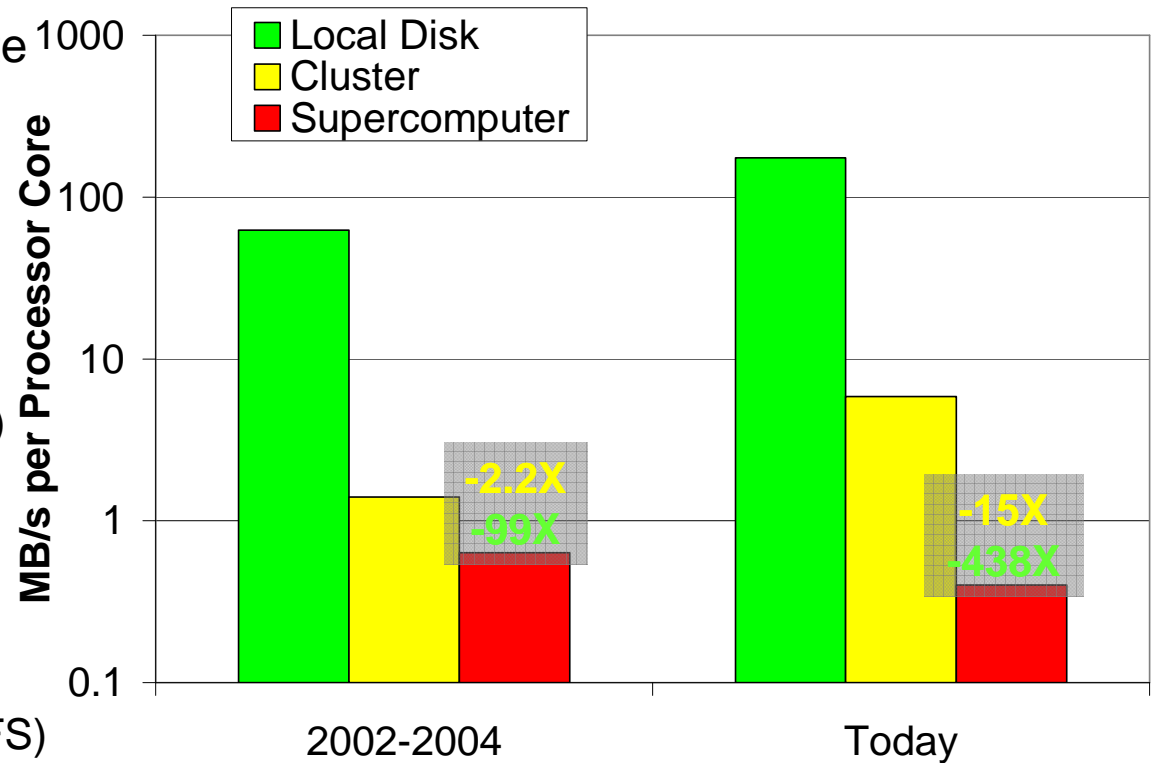
- 2002-2004: ANL/UC TG Site (70GB SCSI)
- Today: PADS (RAID-0, 6 drives 750GB SATA)

- Cluster:

- 2002-2004: ANL/UC TG Site (GPFS, 8 servers, 1Gb/s each)
- Today: PADS (GPFS, SAN)

- Supercomputer:

- 2002-2004: IBM Blue Gene/L (GPFS)
- Today: IBM Blue Gene/P (GPFS)



# Question

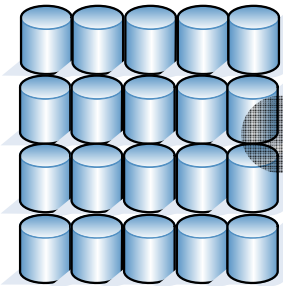
*What if we could combine the scientific community's existing programming paradigms, but yet still exploit the data locality that naturally occurs in scientific workloads?*



# Combine State of the Art Systems

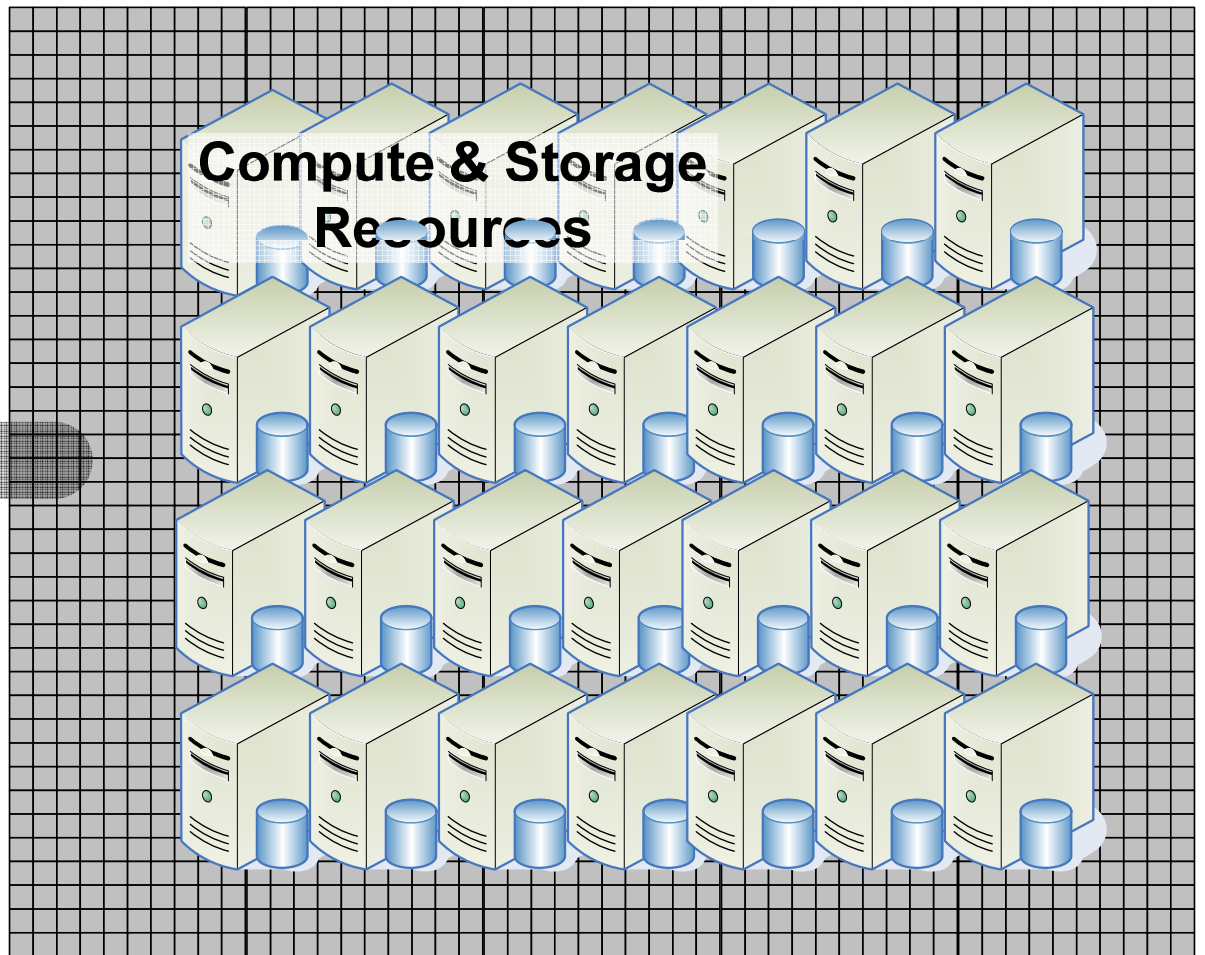
**Network  
Fabric**

**NAS**

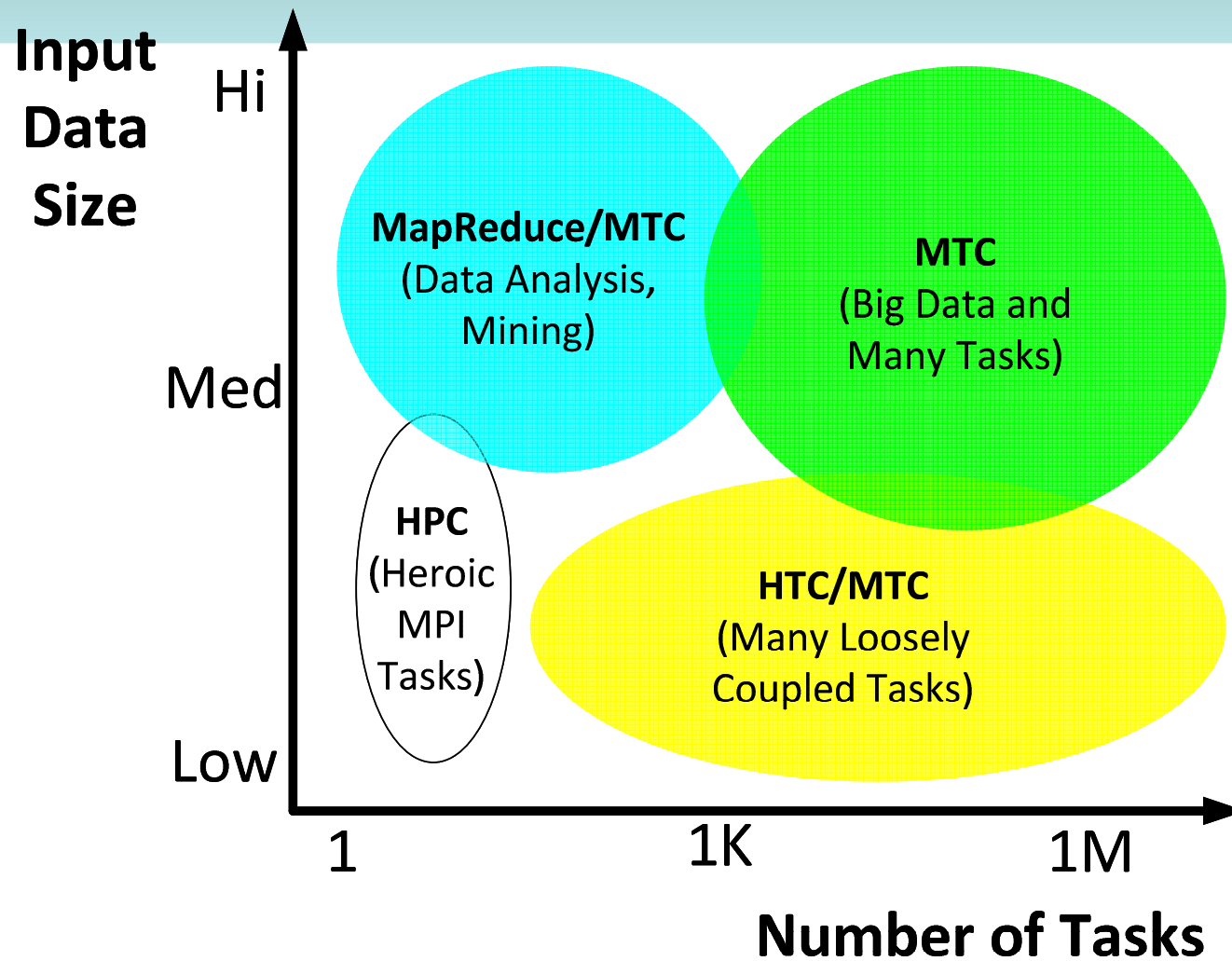


**Network Link(s)**

**Compute & Storage  
Resources**



# Problem Space



# Hypothesis

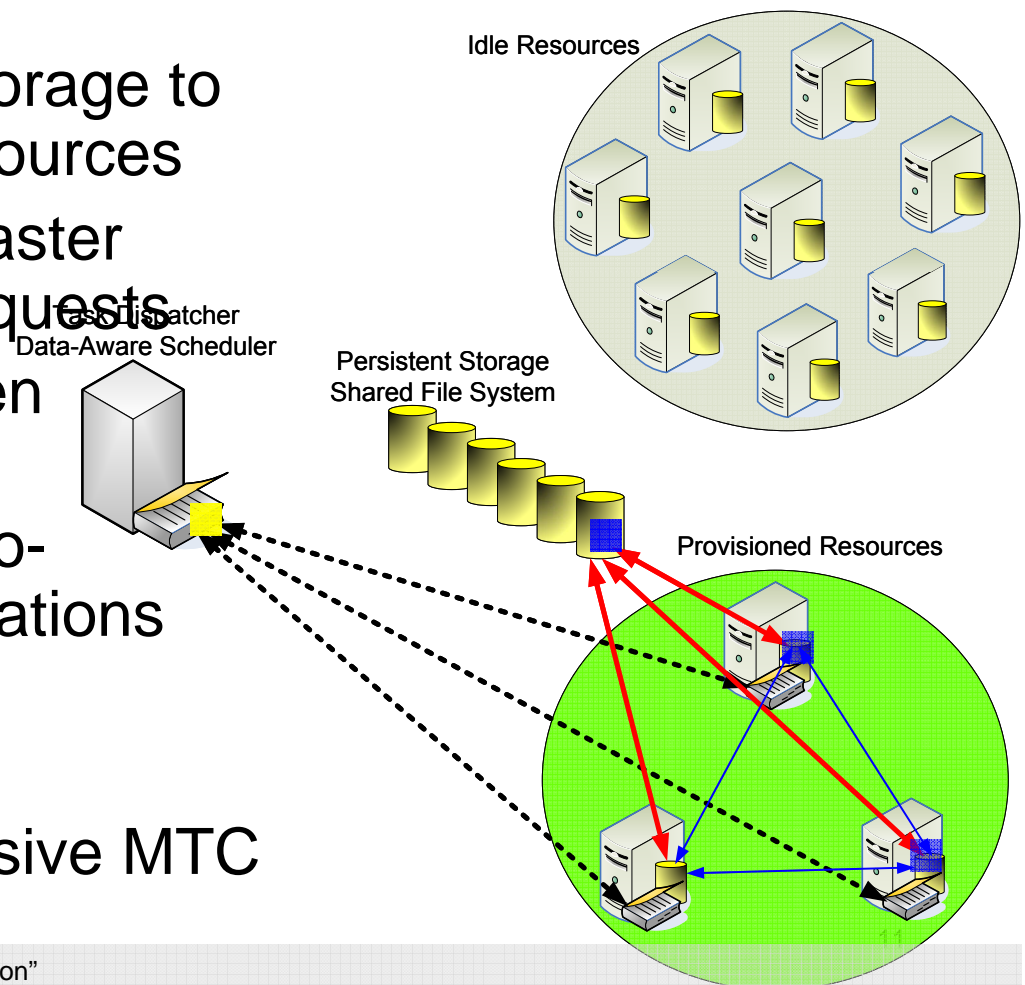
*“Significant performance improvements can be obtained in the analysis of large dataset by leveraging information about data analysis workloads rather than individual data analysis tasks.”*

- **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



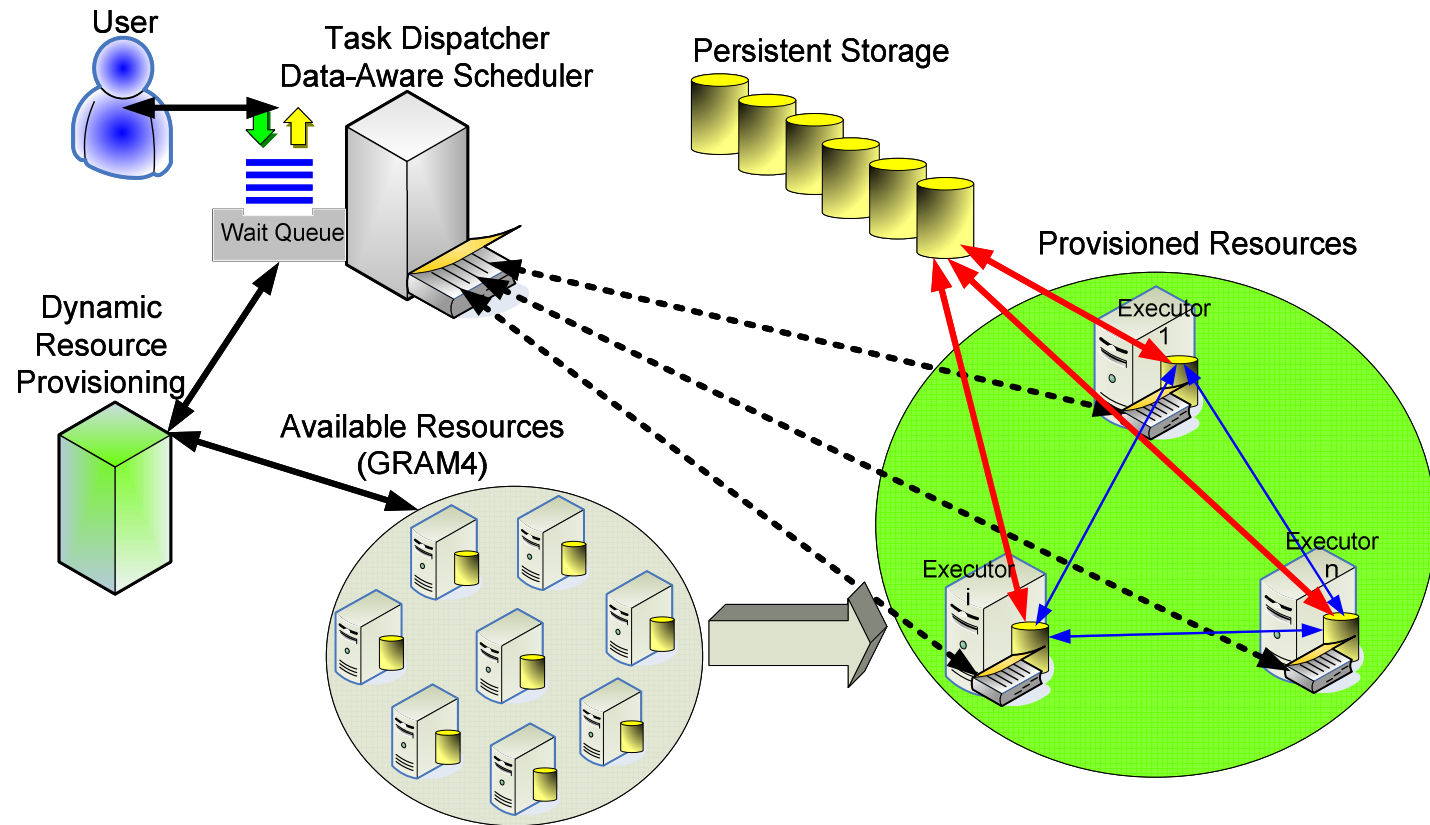
# Proposed Solution: Data Diffusion

- Resource acquired in response to demand
- Data diffuse from archival storage to newly acquired transient resources
- Resource “caching” allows faster responses to subsequent requests
- Resources are released when demand drops
- Optimizes performance by co-scheduling data and computations
- Decrease dependency of a shared/parallel file systems
- Critical to support data intensive MTC



# Data diffusion in Practice

- What would data diffusion look like in practice?
- Extend the Falcon framework



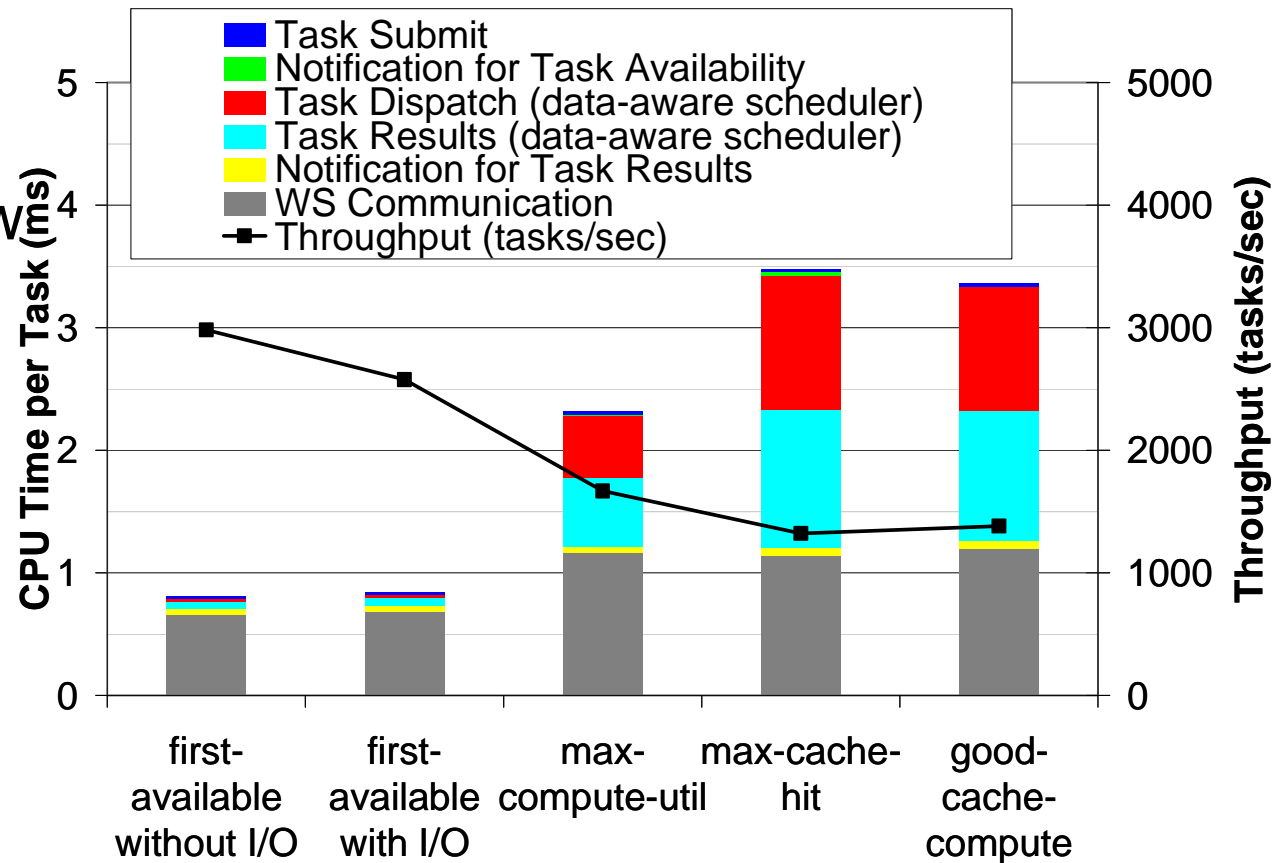
# Scheduling Policies

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



# Data-Aware 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

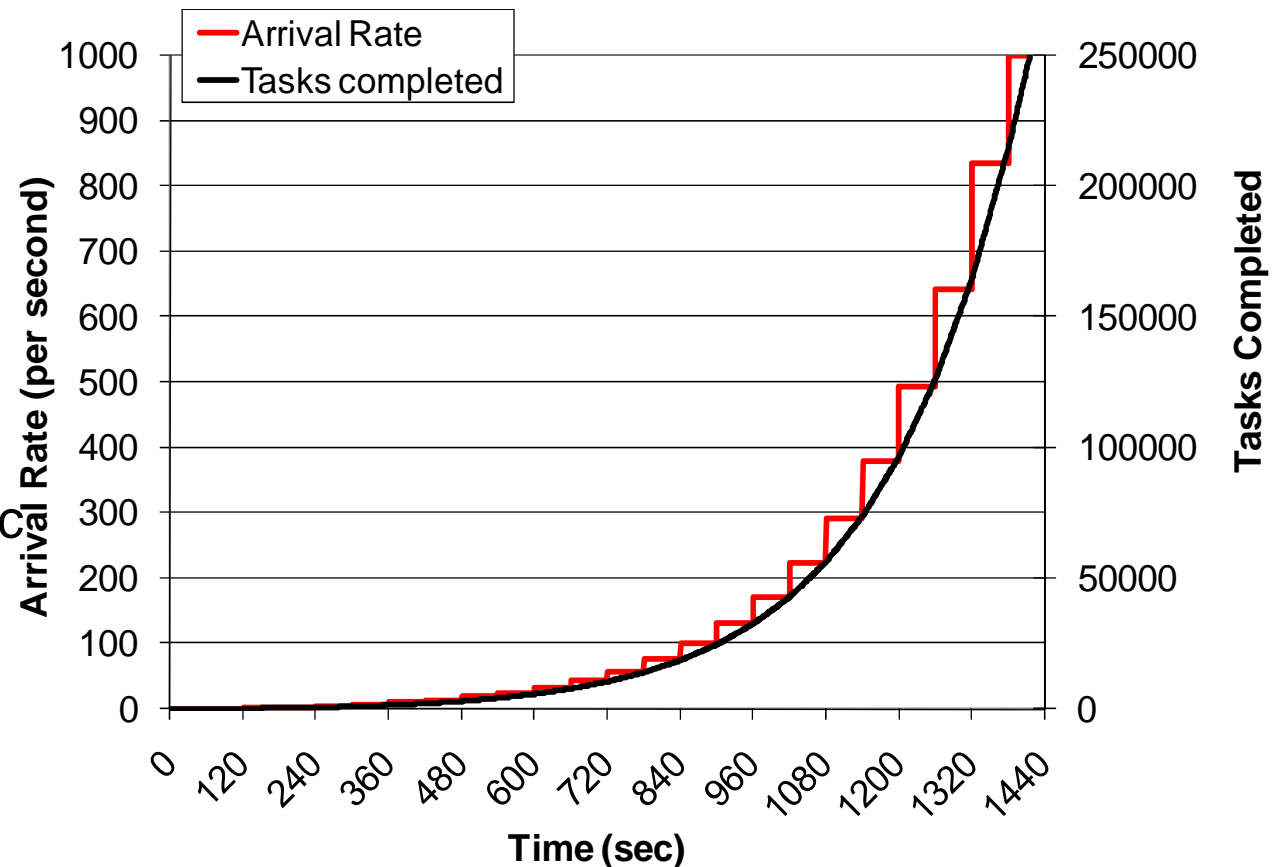


# Workloads

- Monotonically Increasing Workload
  - Emphasizes increasing loads
- Sine-Wave Workload
  - Emphasizes varying loads
- All-Pairs Workload
  - Compare to best case model of active storage
- Image Stacking Workload (Astronomy)
  - Evaluate data diffusion on a real large-scale data-intensive application from astronomy domain

# Monotonically Increasing Workload

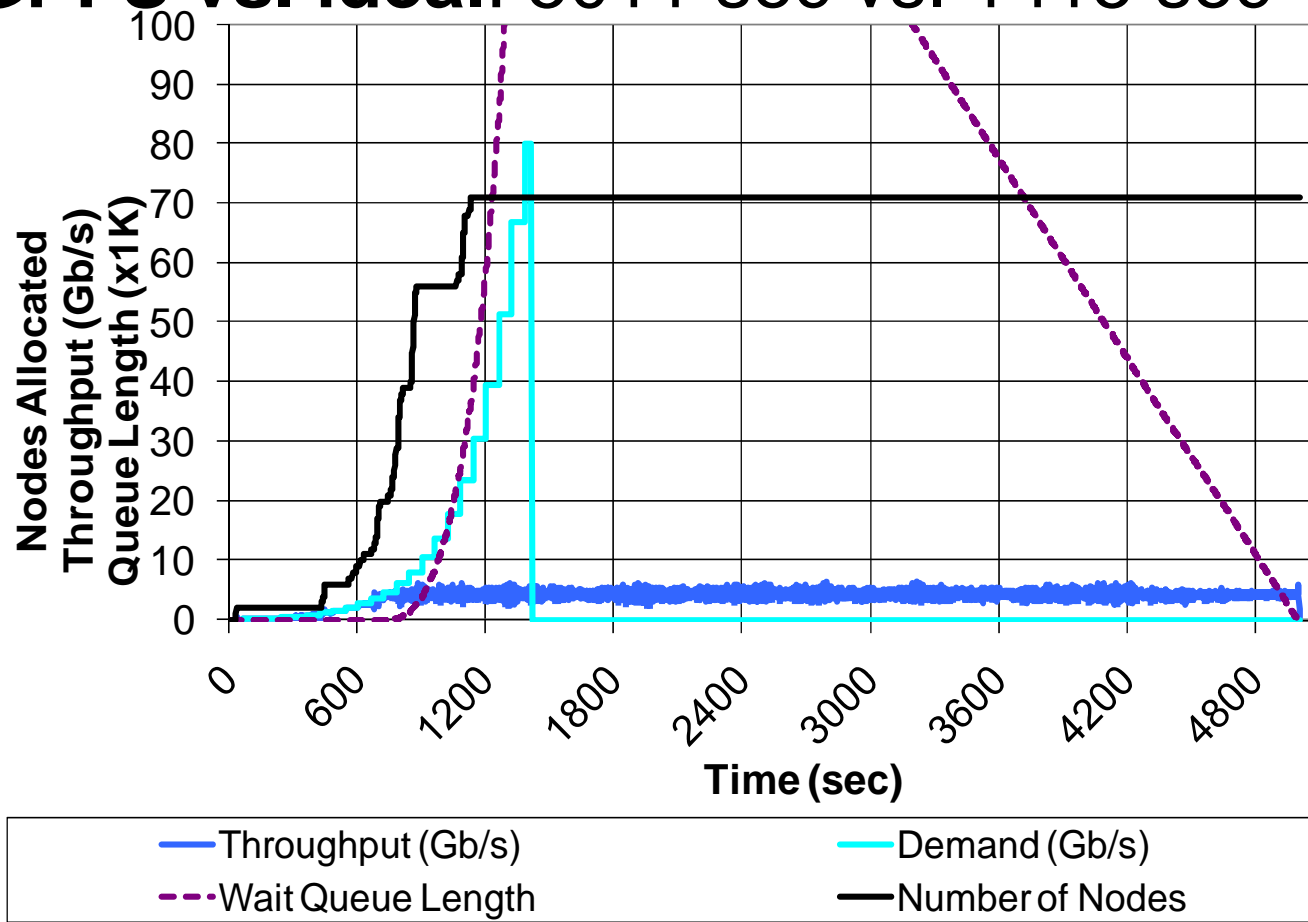
- 250K tasks
  - 10MB reads
  - 10ms compute
- Vary arrival rate:
  - Min: 1 task/sec
  - Increment function:  $\text{CEILING}(*1.3)$
  - Max: 1000 tasks/sec
- 128 processors
- Ideal case:
  - 1415 sec
  - 80Gb/s peak throughput





# Monotonically Increasing Workload First-available (GPFS)

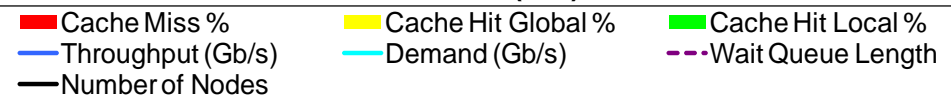
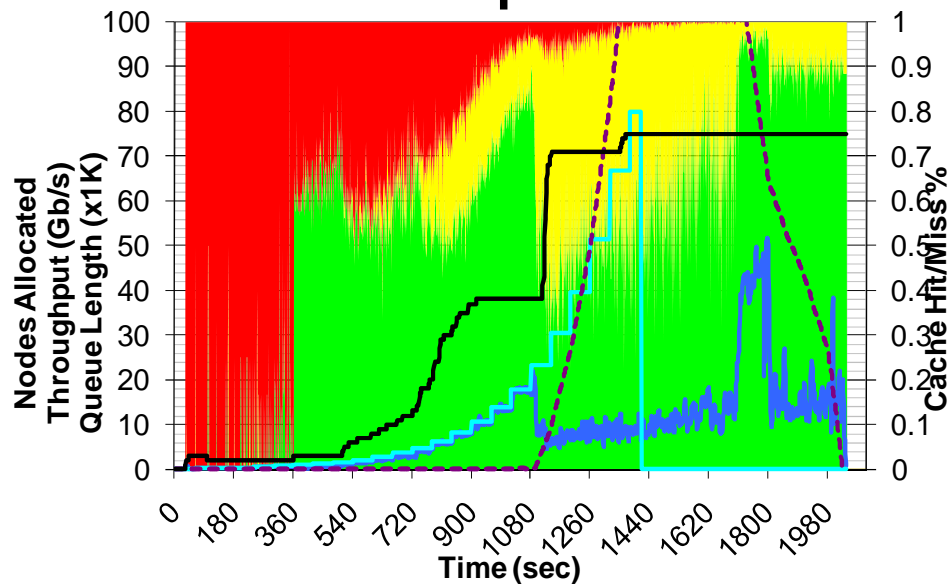
- **GPFS vs. ideal: 5011 sec vs. 1415 sec**



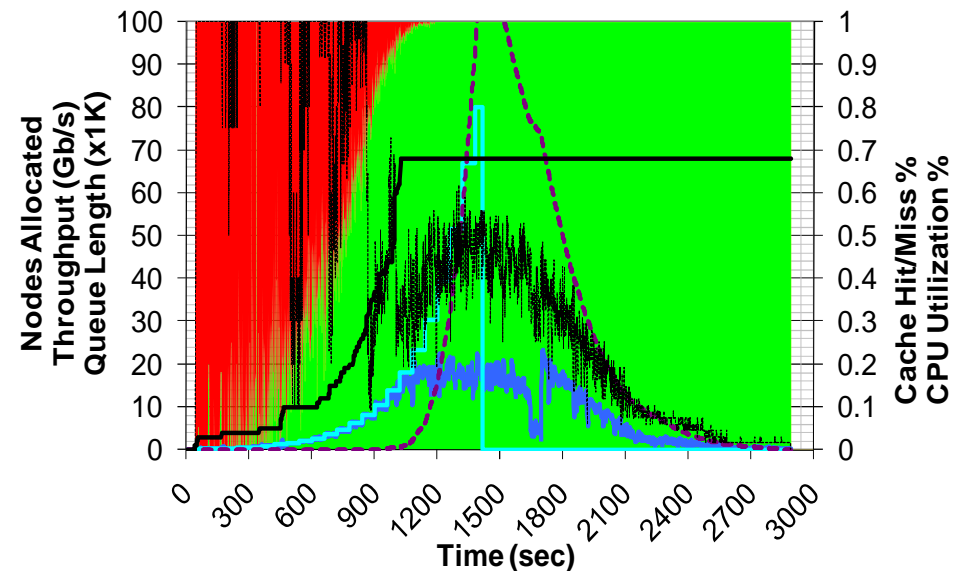
# Monotonically Increasing Workload

## Max-compute-util & Max-cache-hit

### Max-compute-util

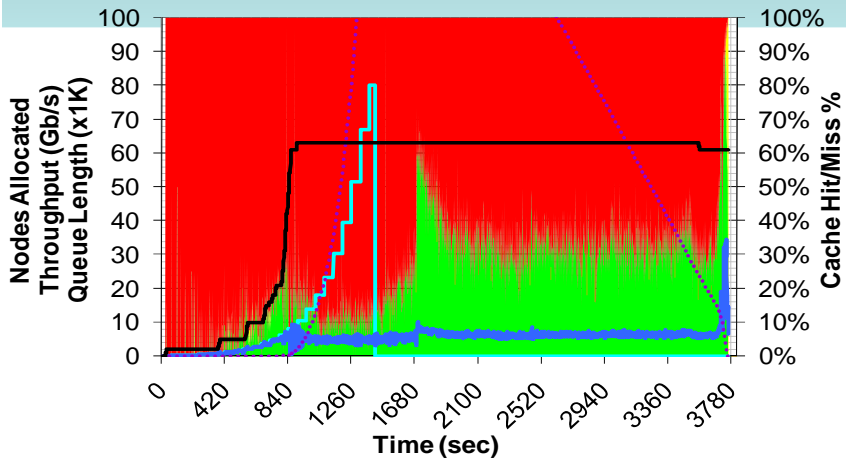


### Max-cache-hit

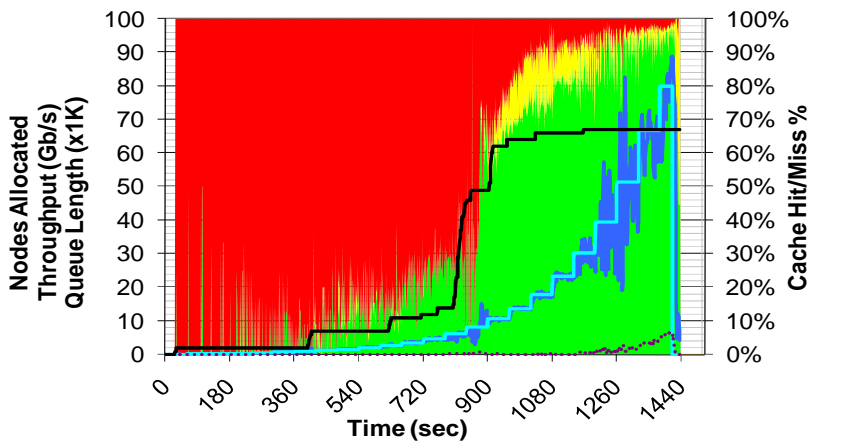
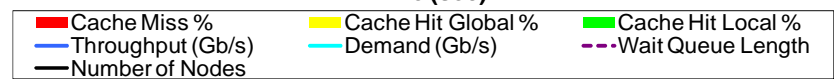
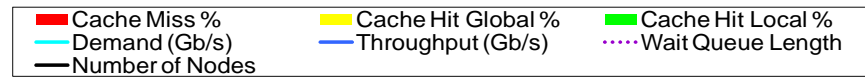
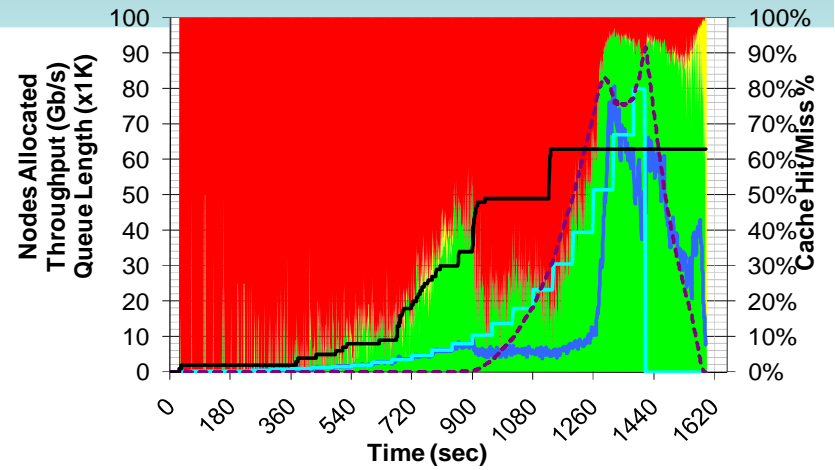


# Monotonically Increasing Workload

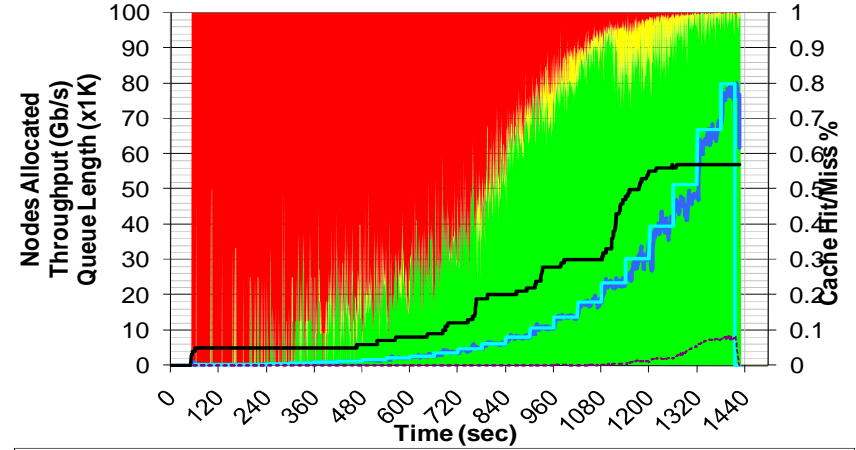
## Good-cache-compute



← 1GB  
1.5GB →



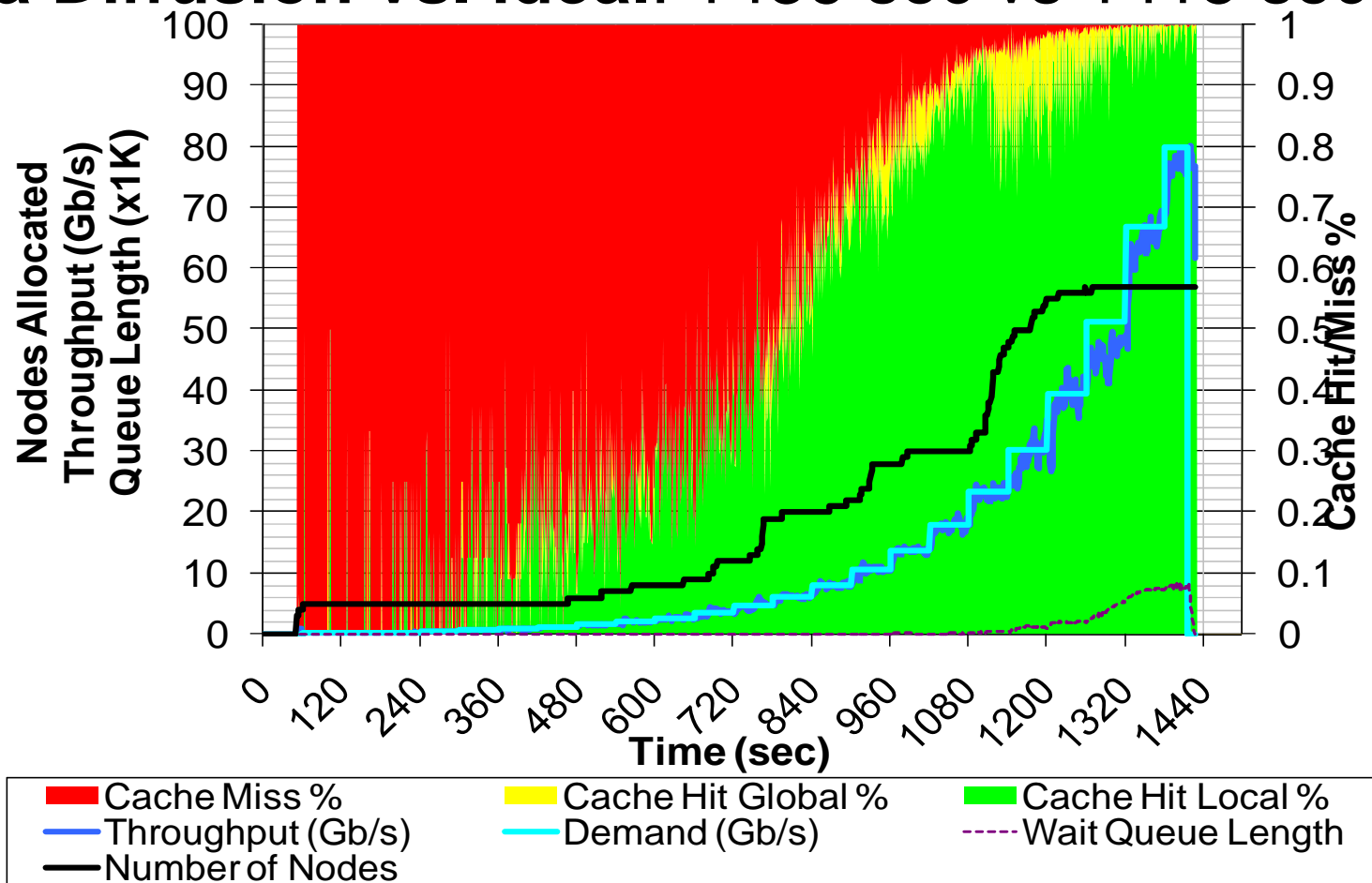
← 2GB  
4GB →



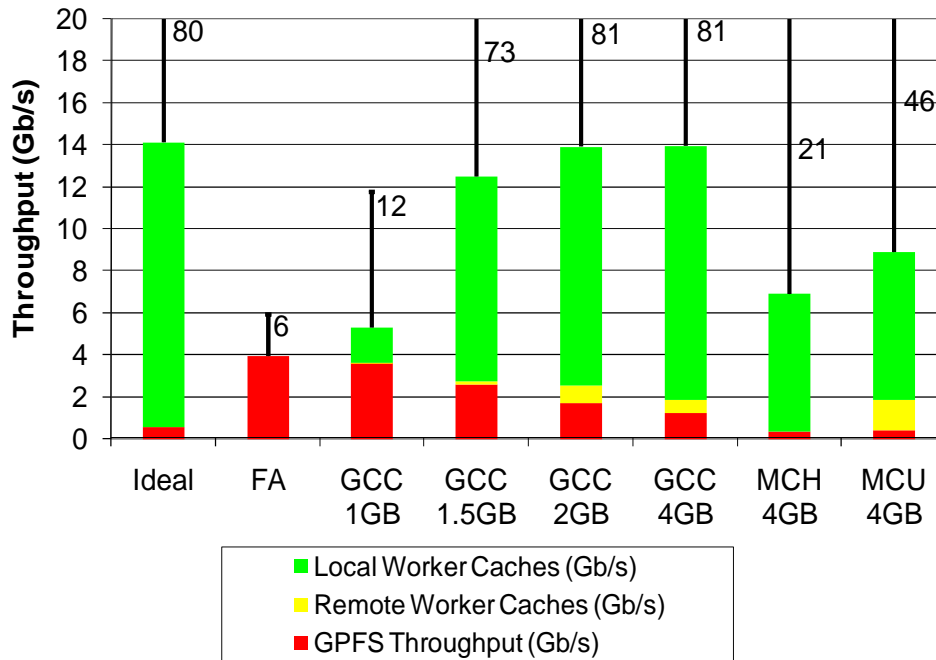
# Monotonically Increasing Workload

## Good-cache-compute

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

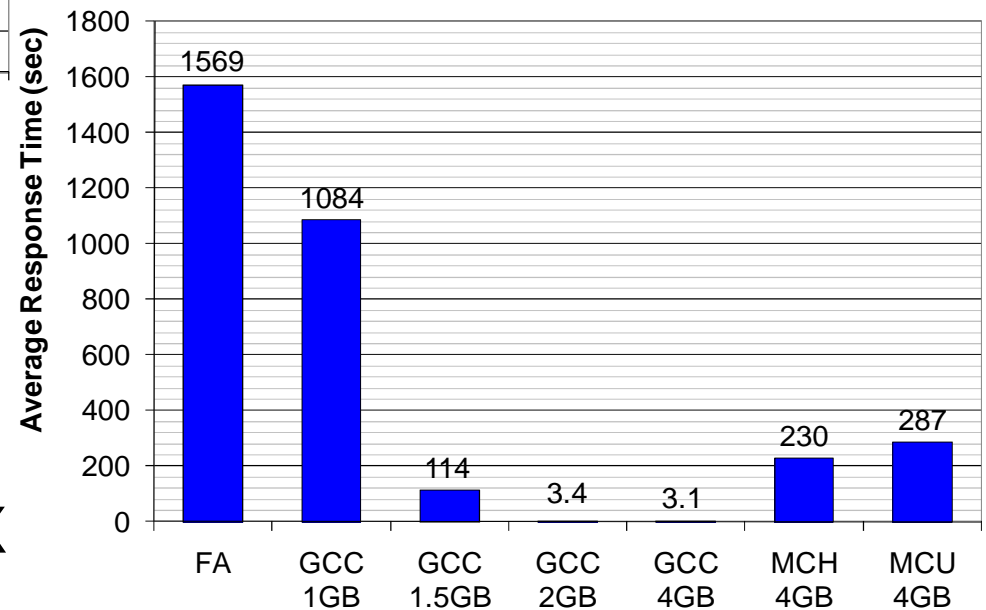


# Monotonically Increasing Workload Throughput and Response Time



← Throughput:

- Average: 14Gb/s vs 4Gb/s
- Peak: 81Gb/s vs. 6Gb/s



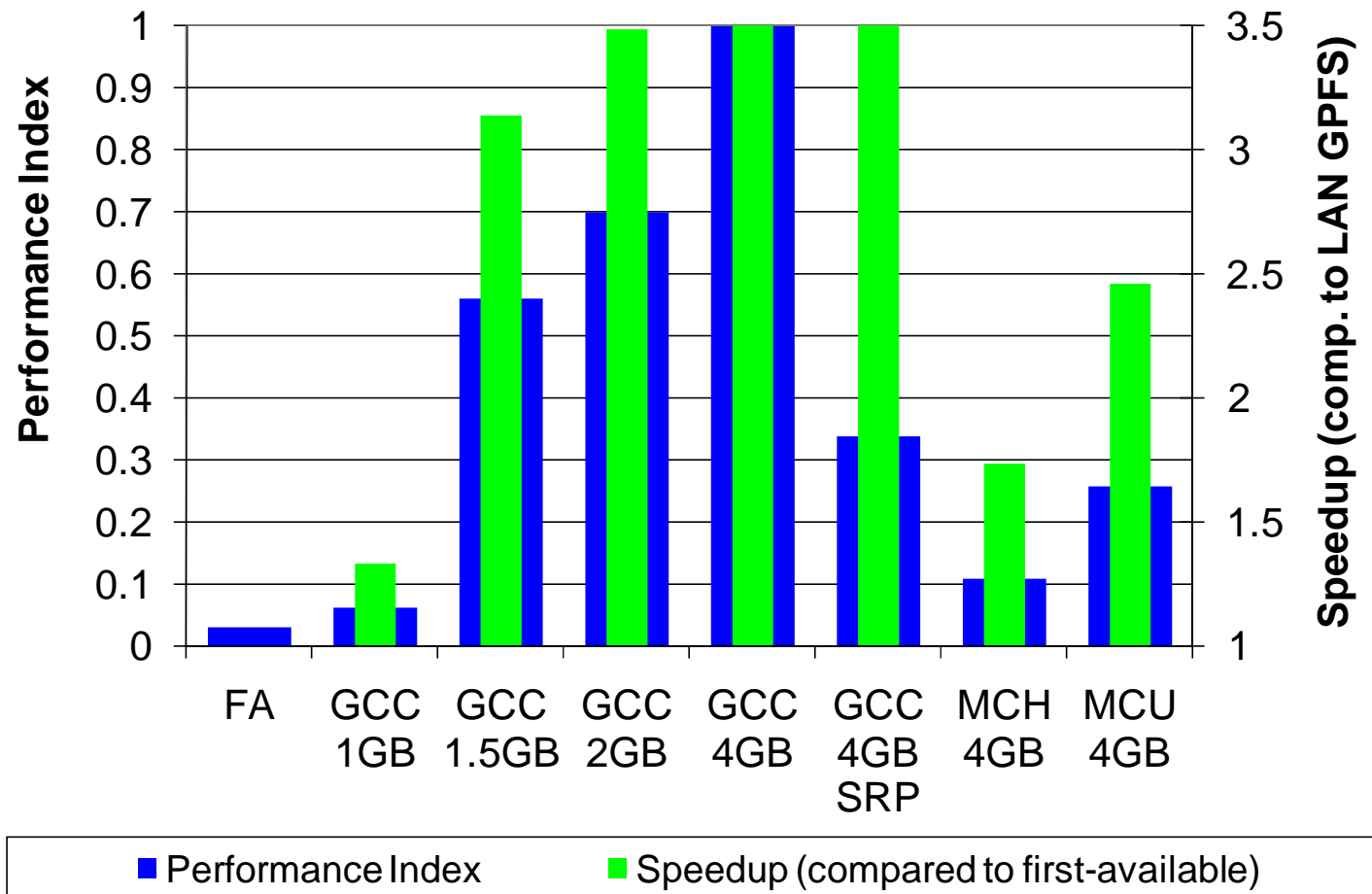
Response Time →

- 3 sec vs 1569 sec → 506X



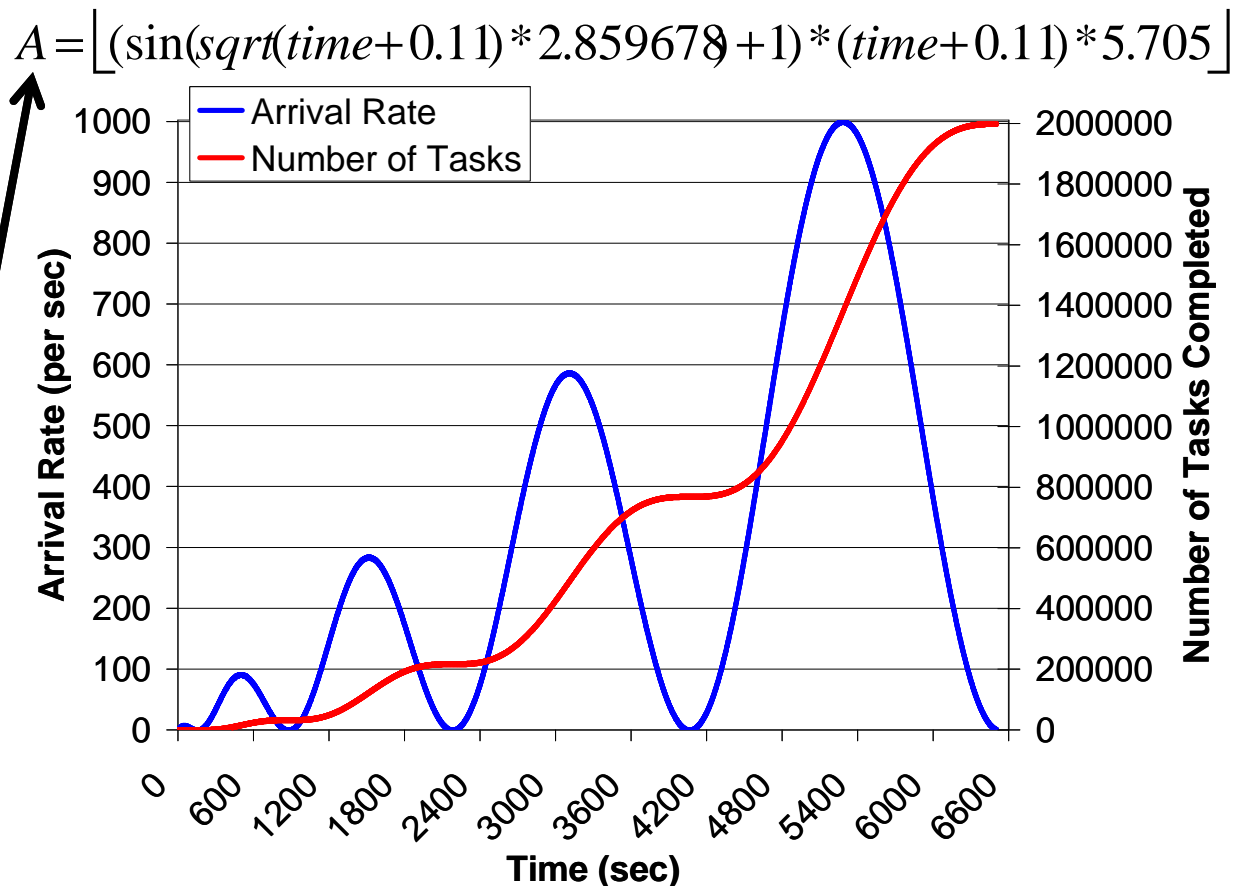
# Monotonically Increasing Workload Performance Index and Speedup

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



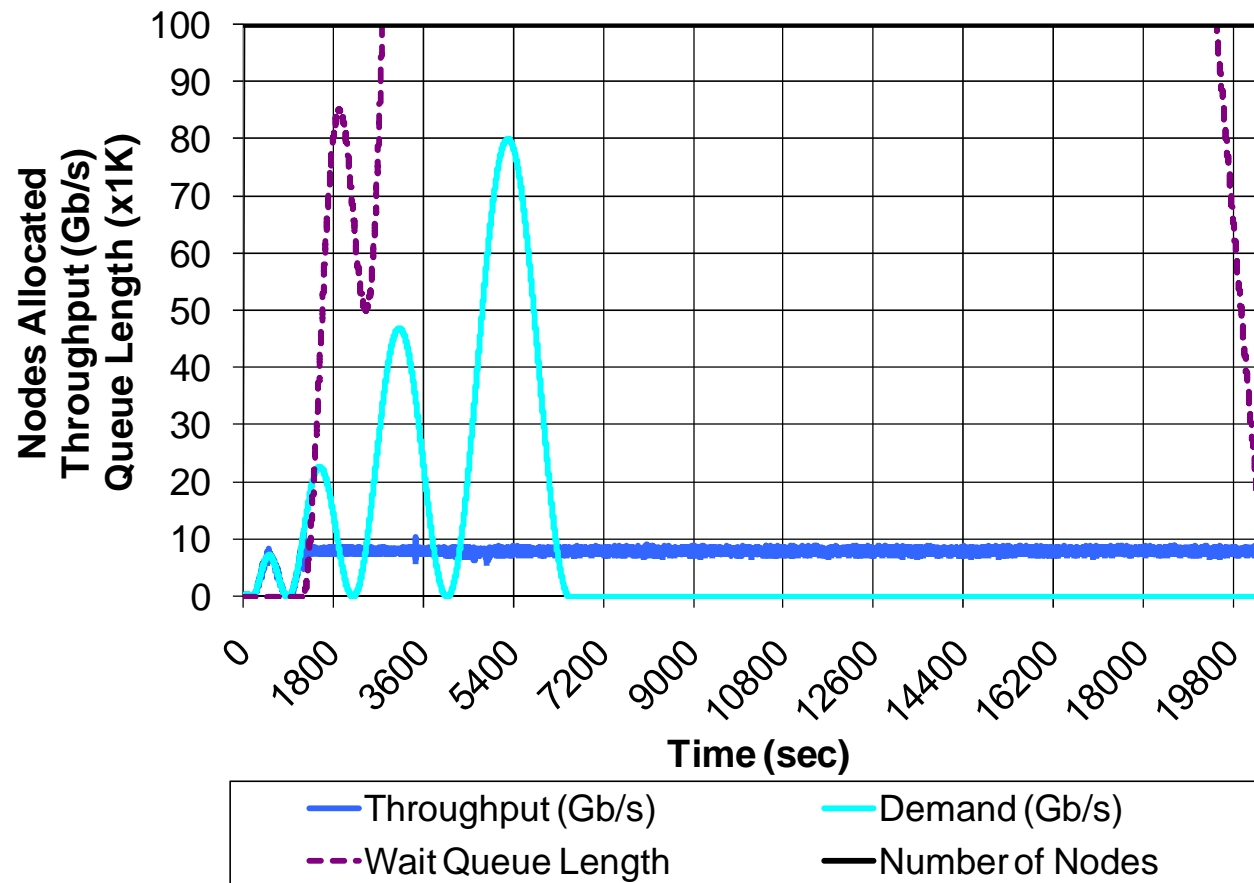
# Sine-Wave Workload

- 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



# Sine-Wave Workload First-available (GPFS)

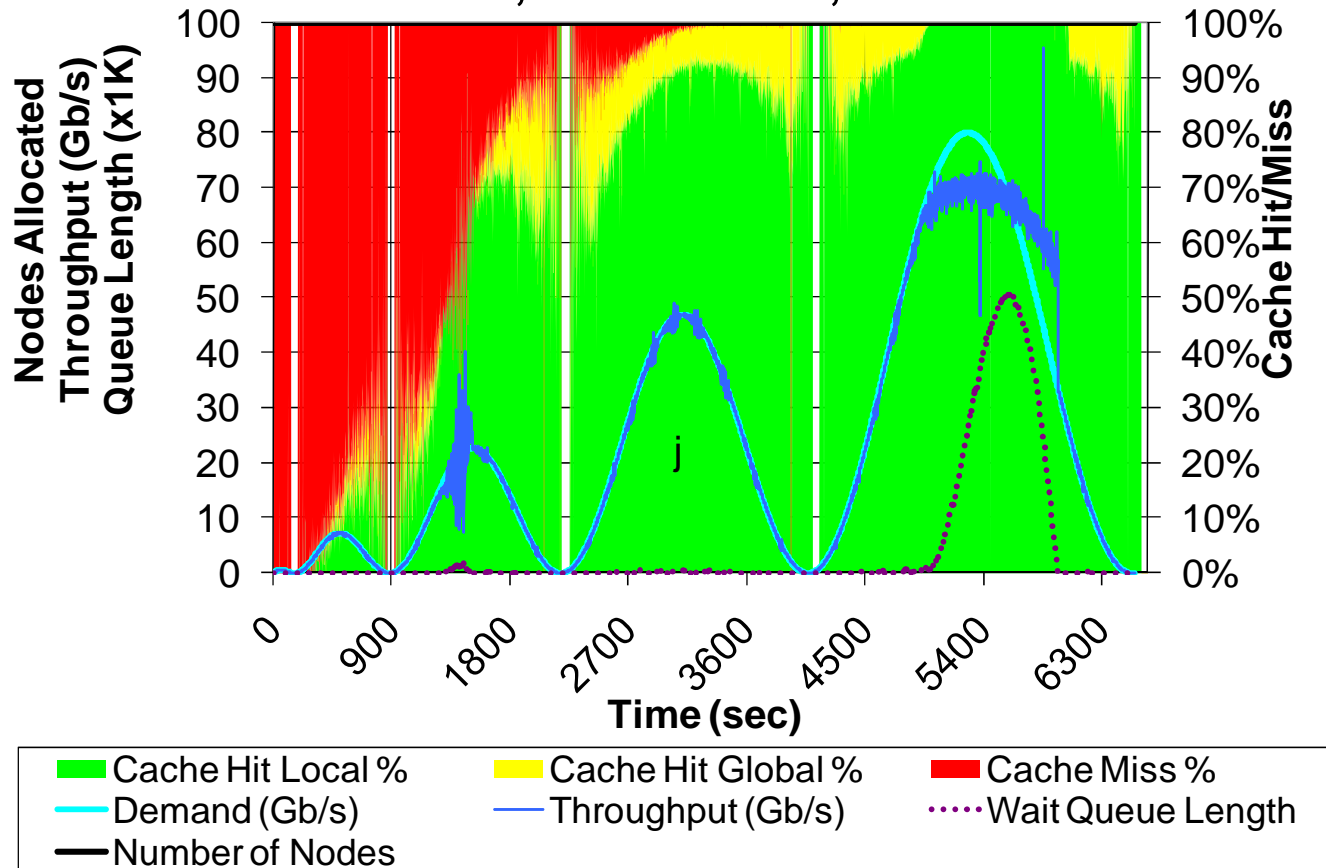
- GPFS → 5.7 hrs, ~8Gb/s, 1138 CPU hrs



# Sine-Wave Workload

## Good-cache-compute and SRP

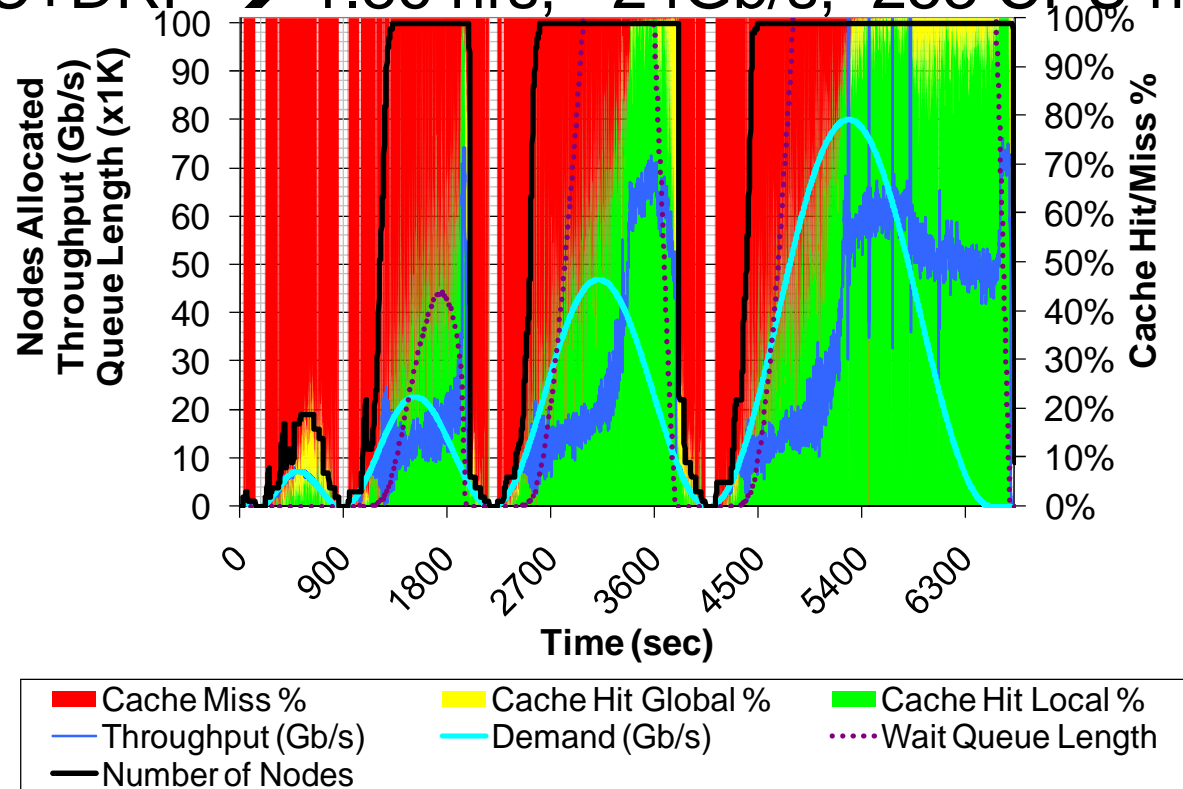
- GPFS → 5.7 hrs, ~8Gb/s, 1138 CPU hrs
- GCC+SRP → 1.8 hrs, ~25Gb/s, 361 CPU hrs



# Sine-Wave Workload

## Good-cache-compute and DRP

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



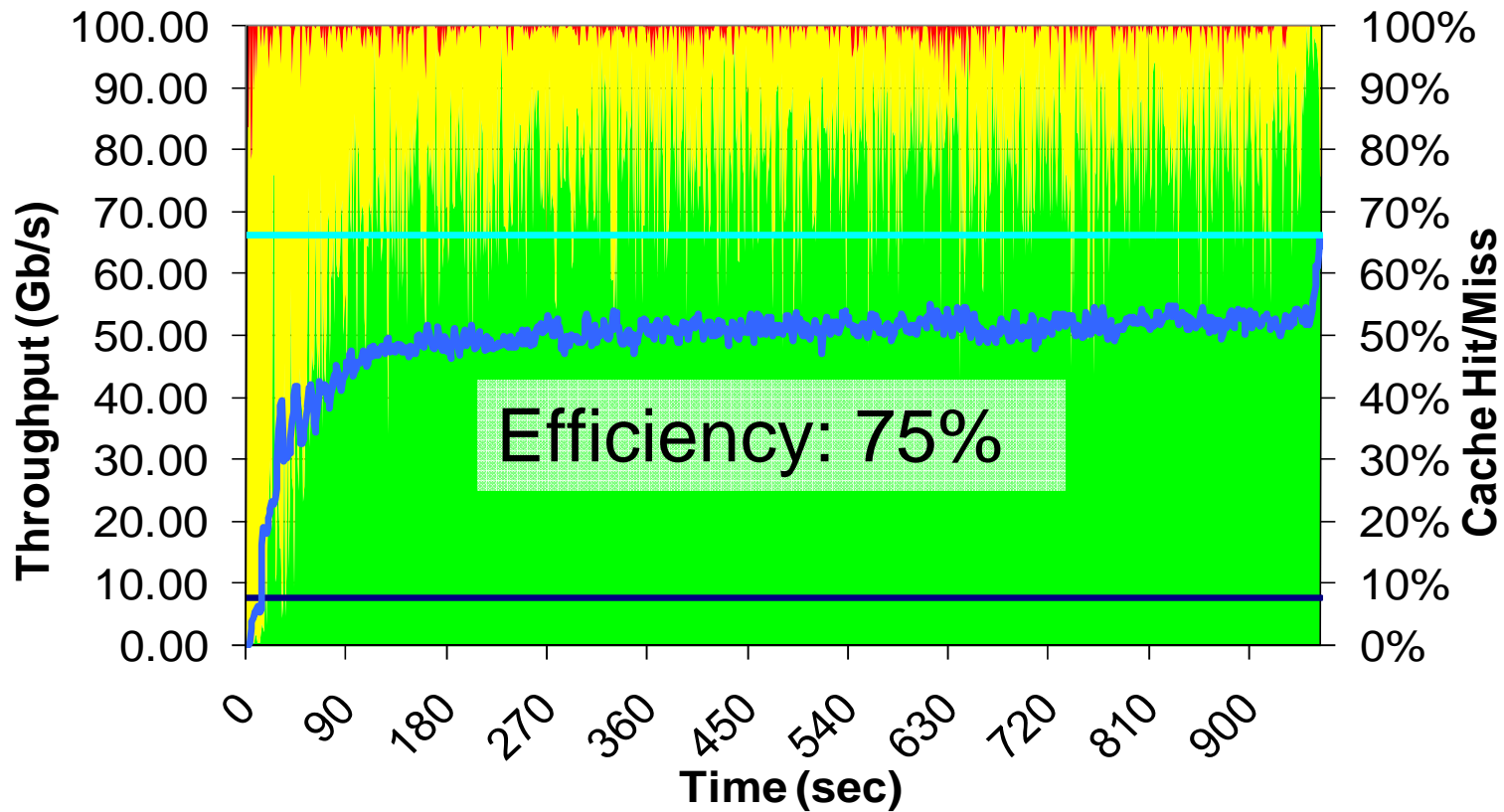
# All-Pairs Workload

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

```
1 foreach $i in A
2   foreach $j in B
3     submit_job F $i $j
4   end
5 end
```



# All-Pairs Workload 500x500 on 200 CPUs



Cache Hit Local %

Cache Miss %

Throughput (Data Diffusion)

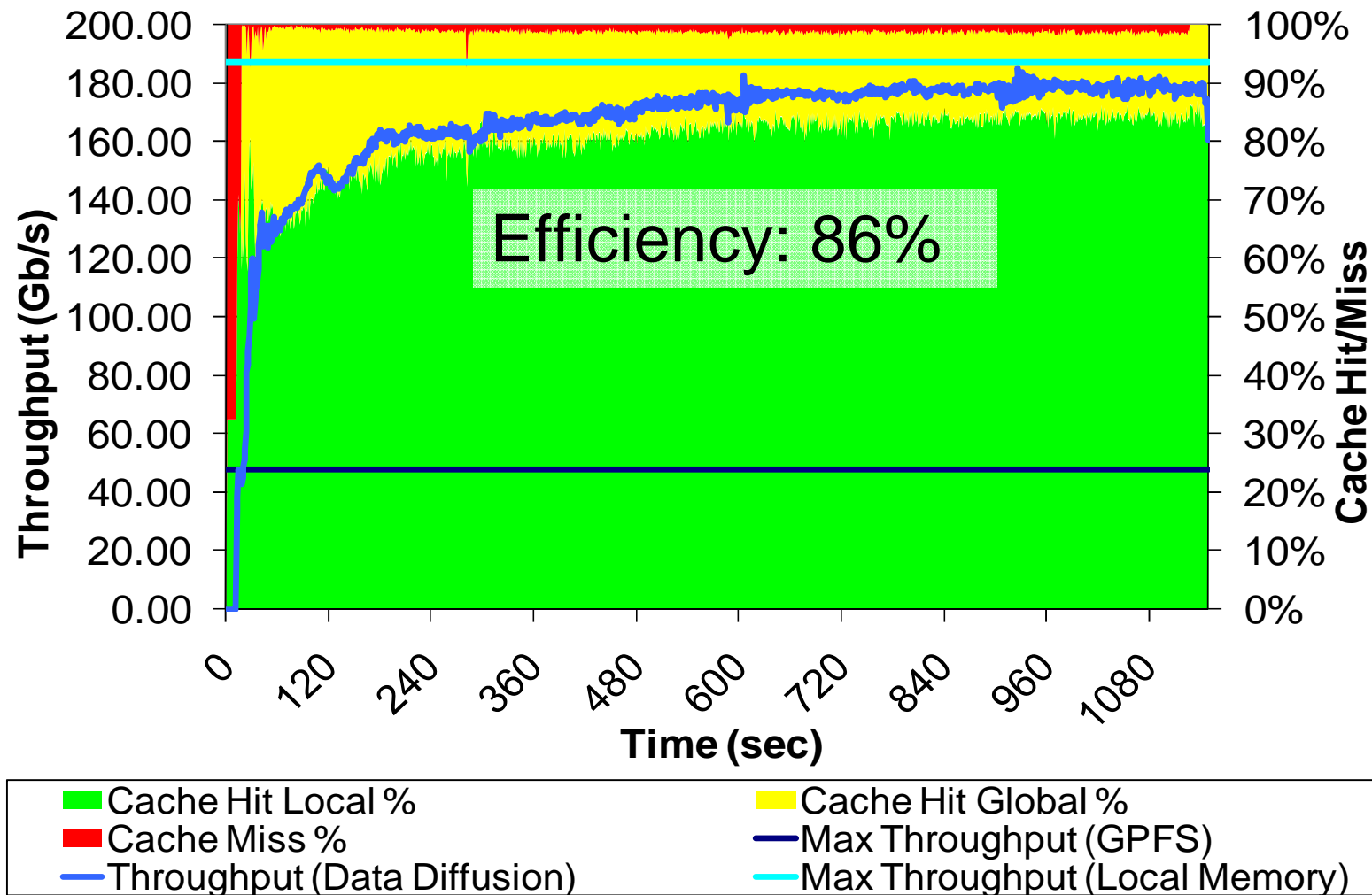
Cache Hit Global %

Max Throughput (GPFS)

Max Throughput (Local Disk)

# All-Pairs Workload

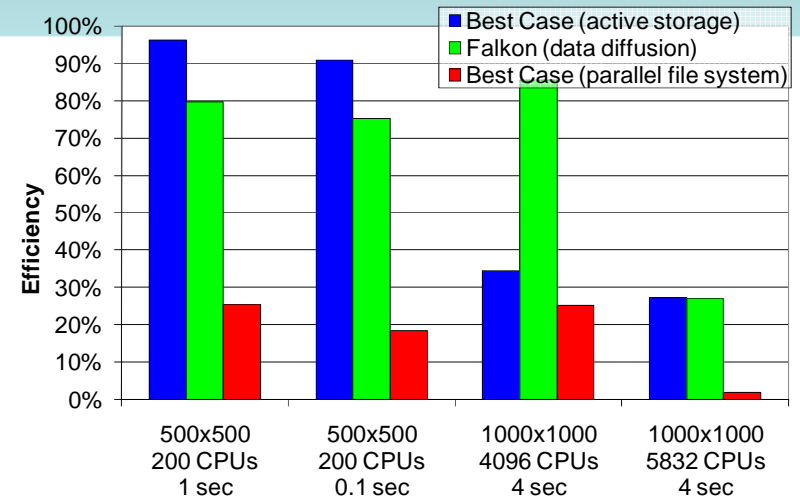
## 1000x1000 on 4K emulated CPUs



# All-Pairs Workload

## Data Diffusion vs. Active Storage

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



Experiment				
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
1000x1000 5832 CPUs 4 sec	Best Case (active storage)	24000	12288	24
	Falkon (data diffusion)	24000	3867	906

**Christopher Moretti, Douglas Thain,  
University of Notre Dame**

# 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
- Best to use data diffusion if
  - Medium to fast data source
  - Task working set  $\ll$  workload working set
  - Task working set fits on local node storage
- If task working set does not fit on local node storage
  - Use parallel file system (i.e. GPFS, Lustre, PVFS, etc)

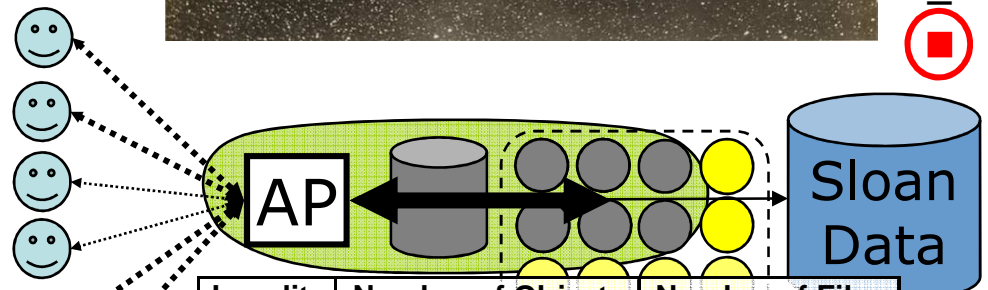
# Image Stacking Workload Astronomy Application

- 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

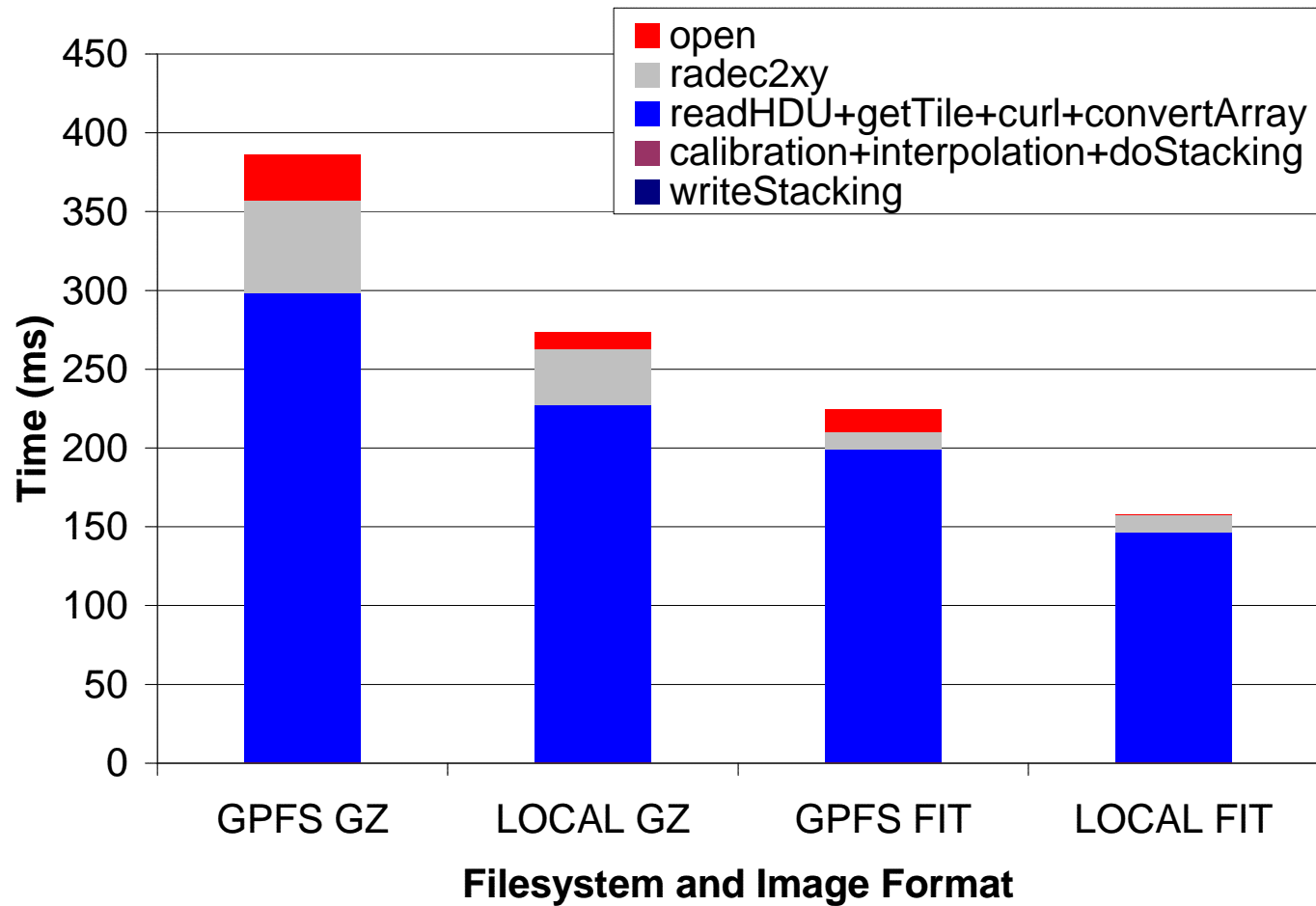


Locality	Number of Objects	Number of Files
1	111700	111700
1.38	154345	111699
2	97999	49000
3	88857	29620
4	76575	19145
5	60590	12120
10	46480	4650
20	40460	2025
30	23695	790

[DADC08] “Accelerating Large-scale Data Exploration through Data Diffusion”

[TG06] “AstroPortal: A Science Gateway for Large-scale Astronomy Data Analysis”

# Image Stacking Workload Profiling



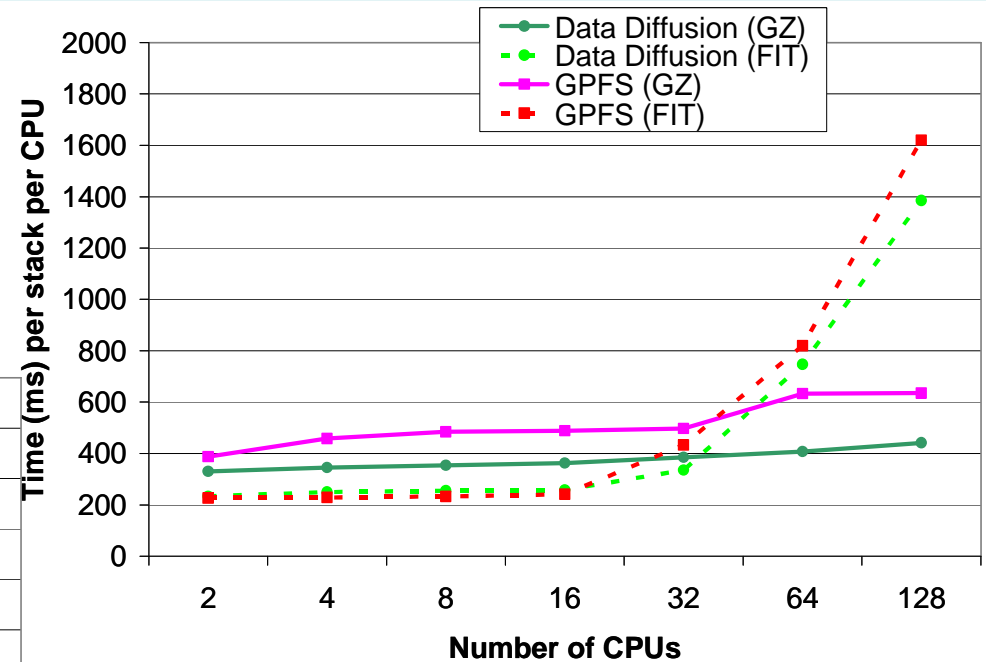
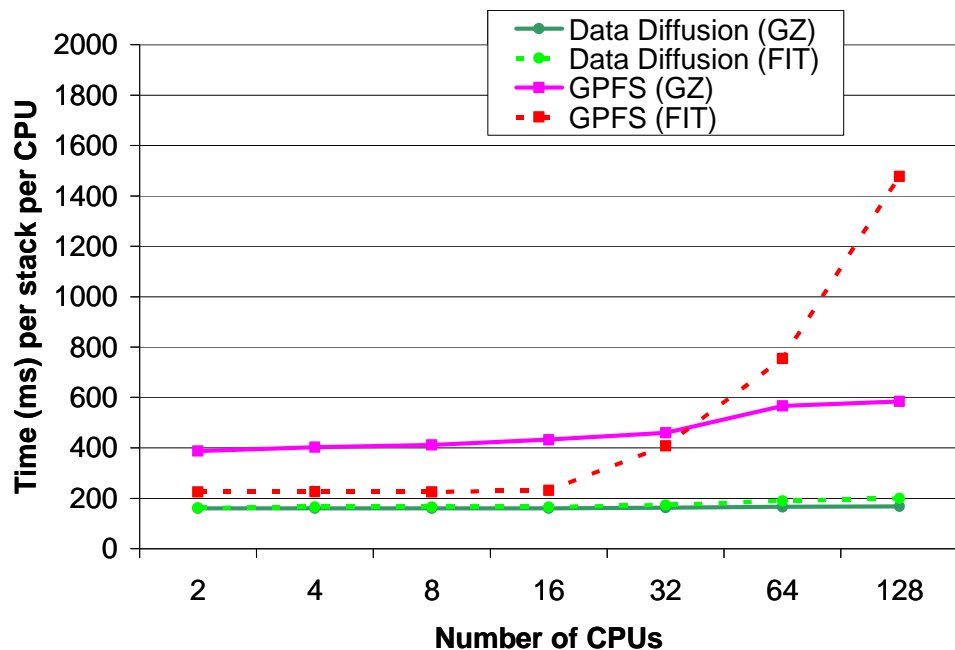


# Image Stacking Workload

## Varying Scale

Low data locality →

- Similar (but better) performance to GPFS

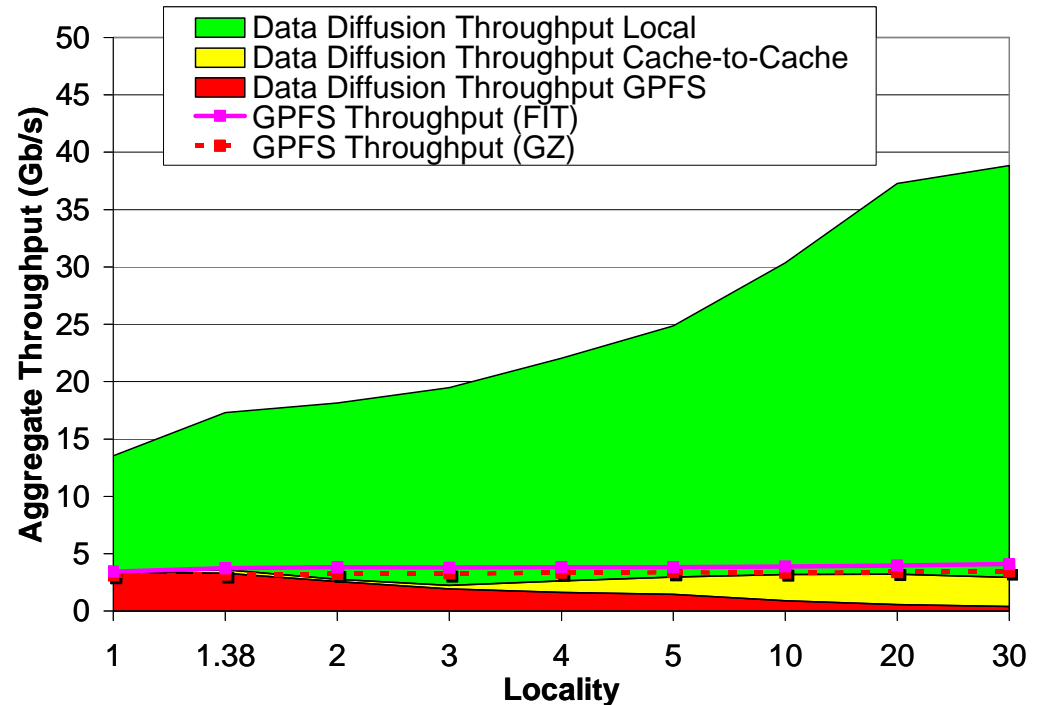
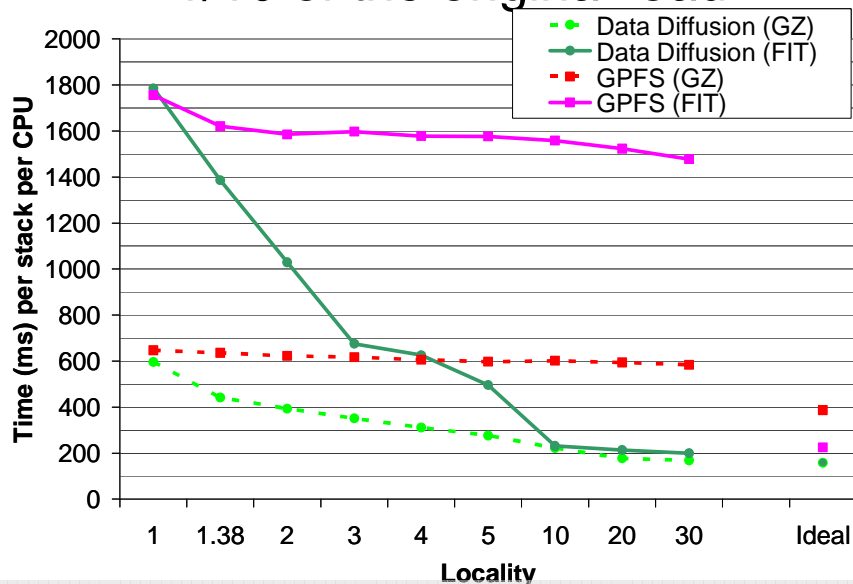


← High data locality

- Near perfect scalability

# Image Stacking Workload Varying Locality

- 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

# Limitations of Data Diffusion

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

# Data Diffusion vs. Others

- [Ghemawat03,Dean04]: MapReduce+GFS
- [Bialecki05]: Hadoop+HDFS
- [Gu06]: Sphere+Sector
- [Tatebe04]: Gfarm
- [Chervenak04]: RLS, DRS
- [Kosar06]: Stork
  
- **Conclusions**
  - *None focused on the co-location of storage and generic black box computations with data-aware scheduling while operating in a dynamic elastic environment*
  - *Swift + Falcon + Data Diffusion is arguably a more generic and powerful solution than MapReduce*

# Contributions

- Identified that data locality is crucial to the efficient use of large scale distributed systems for data-intensive applications → Data Diffusion
  - Integrated streamlined task dispatching with data aware scheduling policies
  - Heuristics to maximize real world performance
  - Suitable for varying, data-intensive workloads
  - Proof of  $O(NM)$  Competitive Caching

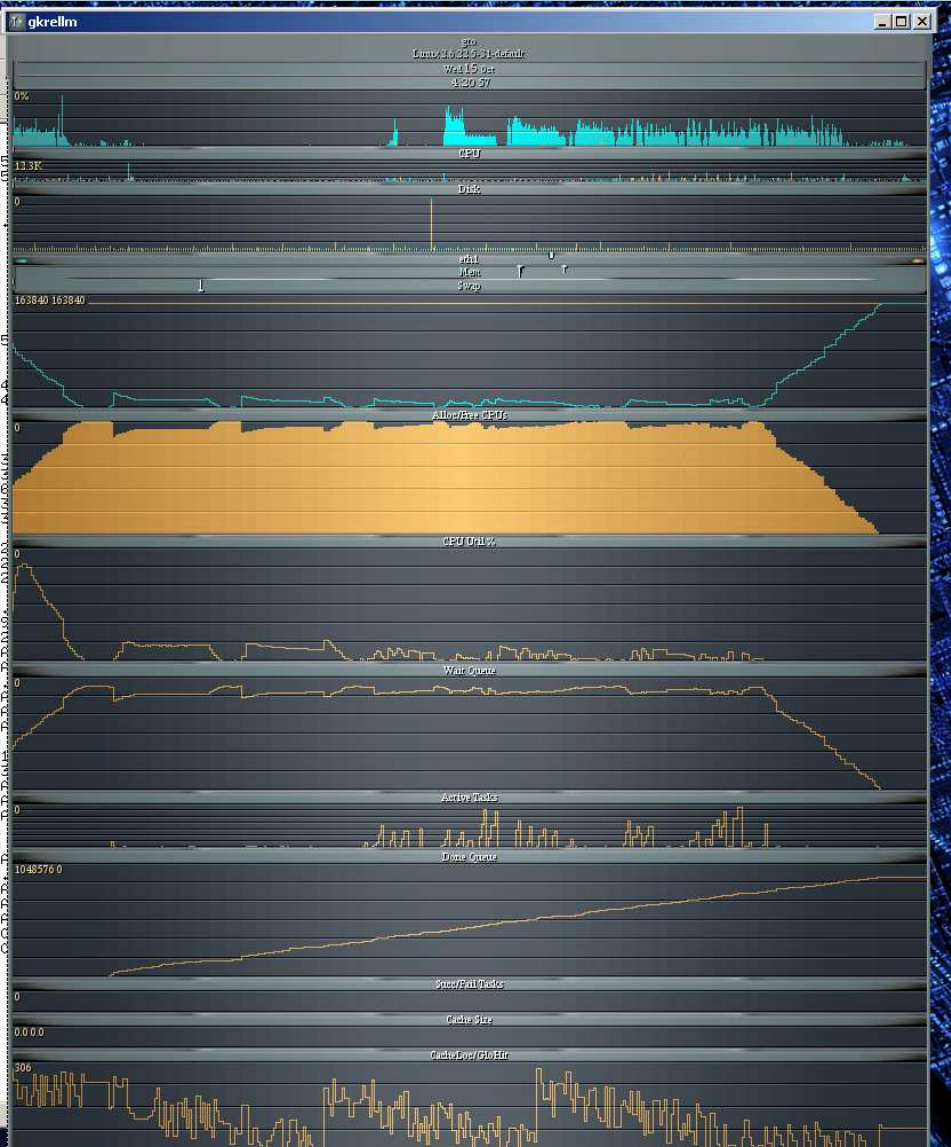
# Falkon Project

- Falkon is a real system
  - Late 2005: Initial prototype, AstroPortal
  - January 2007: Falkon v0
  - November 2007: Globus incubator project v0.1
    - <http://dev.globus.org/wiki/Incubator/Falkon>
  - February 2009: Globus incubator project v0.9
- Implemented in Java (~20K lines of code) and C (~1K lines of code)
  - Open source: svn co <https://svn.globus.org/repos/falkon>
- Source code contributors (beside myself)
  - Yong Zhao, Zhao Zhang, Ben Clifford, Mihael Hategan

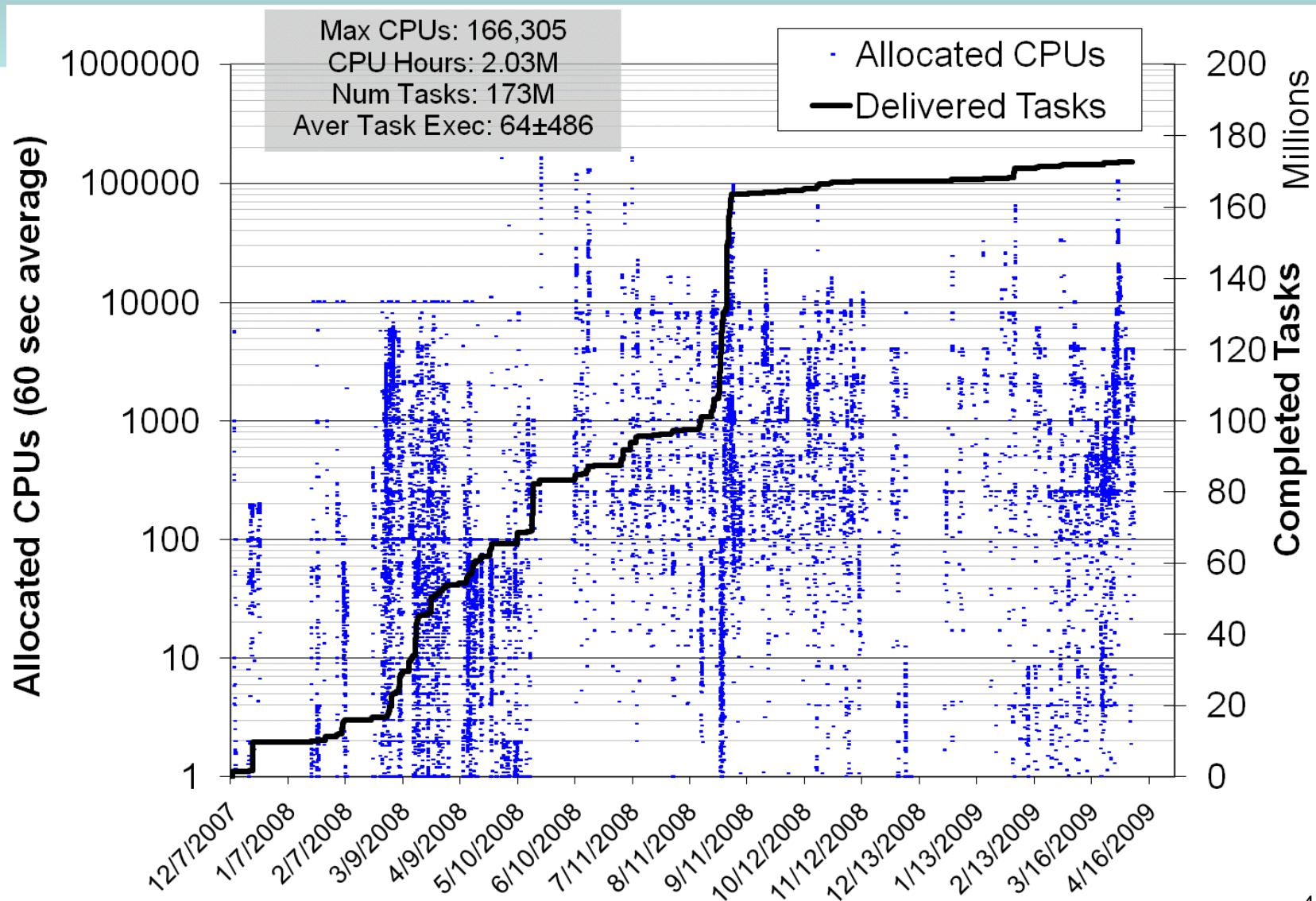


# Falkon Monitoring

```
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 88.89 tasks_tp 2422.31 aver_tp 2692.66 stdev_tp 3115.007 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
409,021 tasks+ 937299 tasks- 0 tasks-> 1048576 completed 89.36 tasks_tp 2439.24 aver_tp 2691.49 stdev_tp 3106.976 ETA
410,025 tasks+ 940027 tasks- 0 tasks-> 1048576 completed 89.57 tasks_tp 2122.51 aver_tp 2689.84 stdev_tp 3102.621 ETA
411,029 tasks+ 942795 tasks- 0 tasks-> 1048576 completed 89.79 tasks_tp 2279.89 aver_tp 2688.65 stdev_tp 3099.212 ETA
412,033 tasks+ 945706 tasks- 0 tasks-> 1048576 completed 90.0 tasks_tp 2216.13 aver_tp 2687.3 stdev_tp 3093.948 ETA
413,038 tasks+ 948866 tasks- 0 tasks-> 1048576 completed 90.21 tasks_tp 2171.31 aver_tp 2686.21 stdev_tp 3089.523 ETA
414,042 tasks+ 952129 tasks- 0 tasks-> 1048576 completed 90.42 tasks_tp 2234.06 aver_tp 2684.52 stdev_tp 3085.189 ETA
415,046 tasks+ 955499 tasks- 0 tasks-> 1048576 completed 90.62 tasks_tp 2047.81 aver_tp 2682.7 stdev_tp 3080.965 ETA
416,050 tasks+ 958976 tasks- 0 tasks-> 1048576 completed 90.82 tasks_tp 2144.42 aver_tp 2681.17 stdev_tp 3076.707 ETA
417,054 tasks+ 962561 tasks- 0 tasks-> 1048576 completed 91.02 tasks_tp 2214.14 aver_tp 2679.84 stdev_tp 3072.434 ETA
418,058 tasks+ 966245 tasks- 0 tasks-> 1048576 completed 91.23 tasks_tp 2067.46 aver_tp 2678.11 stdev_tp 3068.251 ETA
419,061 tasks+ 969942 tasks- 0 tasks-> 1048576 completed 91.43 tasks_tp 2090.36 aver_tp 2676.42 stdev_tp 3064.079 ETA
420,065 tasks+ 973749 tasks- 0 tasks-> 1048576 completed 91.6 tasks_tp 1794.21 aver_tp 2673.73 stdev_tp 3060.176 ETA
421,067 tasks+ 977565 tasks- 0 tasks-> 1048576 completed 91.8 tasks_tp 2106.13 aver_tp 2672.15 stdev_tp 3056.022 ETA
422,069 tasks+ 981392 tasks- 0 tasks-> 1048576 completed 92.0 tasks_tp 2075.4 aver_tp 2670.47 stdev_tp 3051.902 ETA
423,073 tasks+ 985230 tasks- 0 tasks-> 1048576 completed 92.12 tasks_tp 1248.02 aver_tp 2666.5 stdev_tp 3048.561 ETA
424,076 tasks+ 989086 tasks- 0 tasks-> 1048576 completed 92.33 tasks_tp 8425.6 aver_tp 2682.54 stdev_tp 3069.406 ETA
425,079 tasks+ 992951 tasks- 0 tasks-> 1048576 completed 92.54 tasks_tp 3722.22 aver_tp 2685.43 stdev_tp 3085.644 ETA
426,082 tasks+ 996824 tasks- 0 tasks-> 1048576 completed 92.75 tasks_tp 4187.5 aver_tp 2683.57 stdev_tp 3051.614 ETR
427,085 tasks+ 1000705 tasks- 0 tasks-> 1048576 completed 92.95 tasks_tp 1437.5 aver_tp 2682.19 stdev_tp 3047.508 ETA
428,088 tasks+ 1004594 tasks- 0 tasks-> 1048576 completed 93.16 tasks_tp 2044.76 aver_tp 2681.51 stdev_tp 3044.643 ETA
429,091 tasks+ 1008491 tasks- 0 tasks-> 1048576 completed 93.36 tasks_tp 132.25 aver_tp 2684.97 stdev_tp 3048.1 ETA
430,094 tasks+ 1012396 tasks- 0 tasks-> 1048576 completed 93.56 tasks_tp 94.92 aver_tp 2687.6 stdev_tp 3047.182 ETA
431,097 tasks+ 1016309 tasks- 0 tasks-> 1048576 completed 93.76 tasks_tp 1941.47 aver_tp 2685.57 stdev_tp 3043.276 ETA
432,100 tasks+ 1020229 tasks- 0 tasks-> 1048576 completed 93.95 tasks_tp 1437.57 aver_tp 2681.53 stdev_tp 3041.282 ETA
433,103 tasks+ 1024156 tasks- 0 tasks-> 1048576 completed 94.15 tasks_tp 1047.09 aver_tp 2681.77 stdev_tp 3040.335 ETA
434,106 tasks+ 1028089 tasks- 0 tasks-> 1048576 completed 94.34 tasks_tp 1047.56 aver_tp 2681.53 stdev_tp 3039.485 ETA
435,109 tasks+ 1032028 tasks- 0 tasks-> 1048576 completed 94.53 tasks_tp 1047.56 aver_tp 2681.53 stdev_tp 3038.635 ETA
436,112 tasks+ 1035973 tasks- 0 tasks-> 1048576 completed 94.72 tasks_tp 1047.56 aver_tp 2681.53 stdev_tp 3037.785 ETA
437,115 tasks+ 1040024 tasks- 0 tasks-> 1048576 completed 94.91 tasks_tp 1527.58 aver_tp 2685.38 stdev_tp 3031.597 ETA
438,118 tasks+ 1044081 tasks- 0 tasks-> 1048576 completed 95.1 tasks_tp 3585.56 aver_tp 2690.71 stdev_tp 3027.863 ETA
439,121 tasks+ 1048144 tasks- 0 tasks-> 1048576 completed 95.29 tasks_tp 4550.6 aver_tp 2695.8 stdev_tp 3025.337 ETA
440,124 tasks+ 1052213 tasks- 0 tasks-> 1048576 completed 95.48 tasks_tp 4550.6 aver_tp 2695.8 stdev_tp 3022.899 ETA
441,127 tasks+ 1056288 tasks- 0 tasks-> 1048576 completed 95.67 tasks_tp 4550.6 aver_tp 2695.8 stdev_tp 3020.977 ETA
442,130 tasks+ 1060369 tasks- 0 tasks-> 1048576 completed 95.86 tasks_tp 4472.22 aver_tp 2694.91 stdev_tp 3018.048 ETA
443,133 tasks+ 1064456 tasks- 0 tasks-> 1048576 completed 96.05 tasks_tp 2319.44 aver_tp 2693.04 stdev_tp 3012.127 ETA
444,136 tasks+ 1068549 tasks- 0 tasks-> 1048576 completed 96.24 tasks_tp 3662.7 aver_tp 2695.53 stdev_tp 3009.572 ETA
445,139 tasks+ 1072648 tasks- 0 tasks-> 1048576 completed 96.43 tasks_tp 2864.48 aver_tp 2695.24 stdev_tp 3004.628 ETA
446,142 tasks+ 1076753 tasks- 0 tasks-> 1048576 completed 96.62 tasks_tp 1929.56 aver_tp 2695.24 stdev_tp 3000.948 ETA
447,145 tasks+ 1080864 tasks- 0 tasks-> 1048576 completed 96.81 tasks_tp 395.83 aver_tp 2687.24 stdev_tp 2999.32 ETA
448,148 tasks+ 1084981 tasks- 0 tasks-> 1048576 completed 97.0 tasks_tp 2266.87 aver_tp 2686.15 stdev_tp 2996.489 ETA
449,151 tasks+ 1089104 tasks- 0 tasks-> 1048576 completed 97.19 tasks_tp 2683.16 aver_tp 2686.15 stdev_tp 2993.536 ETA
450,154 tasks+ 1093233 tasks- 0 tasks-> 1048576 completed 97.38 tasks_tp 2266.87 aver_tp 2686.15 stdev_tp 2993.536 ETA
451,157 tasks+ 1097368 tasks- 0 tasks-> 1048576 completed 97.57 tasks_tp 2354.17 aver_tp 2685.29 stdev_tp 2987.766 ETA
452,160 tasks+ 1101509 tasks- 0 tasks-> 1048576 completed 97.76 tasks_tp 0.0 aver_tp 2678.35 stdev_tp 2987.016 ETA
453,163 tasks+ 1105656 tasks- 0 tasks-> 1048576 completed 97.95 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>
```



# Falkon Activity History (16 months)





# MTAGS

## 2nd Workshop on Many-Task Computing on Grids and Supercomputers

co-located with [ACM/IEEE SC09 \(International Conference for High Performance, Networking, Storage and Analysis\)](#)  
Portland, Oregon -- November 16th, 2009

### [Home](#)

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

[Program](#)  
[Committee](#)

### [Important Dates](#)

[Paper](#)  
[Submission](#)

### [Venue](#)

### [Registration](#)

[Workshop](#)  
[Program](#)

### Important Dates

Abstract Due:	August 1st, 2009
Papers Due:	September 1st, 2009
Notification of Acceptance:	October 1st, 2009
Camera Ready Papers Due:	November 1st, 2009
Workshop Date:	November 16th, 2009

### Committee Members

#### Workshop Chairs

Ioan Raicu, University of Chicago  
Ian Foster, University of Chicago & Argonne National Laboratory  
Yong Zhao, Microsoft

#### Technical Committee

- \* David Abramson, Monash University, Australia
- \* Pete Beckman, Argonne National Laboratory, USA
- \* Ian Foster, University of Chicago & Argonne National Laboratory, USA
- \* Bob Grossman, University of Illinois at Chicago, USA
- \* Indranil Gupta, University of Illinois at Urbana Champaign, USA
- \* Alexandru Iosup, Delft University of Technology, Netherlands
- \* Zhou Lei, Shanghai University, China
- \* Shiyong Lu, Wayne State University, USA
- \* Reagan Moore, University of North Carolina at Chapel Hill, USA
- \* Marlon Pierce, Indiana University, USA
- \* Ioan Raicu, University of Chicago, USA
- \* Matei Ripeanu, University of British Columbia, Canada
- \* Greg Thain, Univeristy of Wisconsin, USA
- \* Matthew Woitaszek, The University Corporation for Atmospheric Research, USA
- \* Sherali Zeadally, University of the District of Columbia, USA
- \* Yong Zhao, Microsoft, USA

ACM MTAGS09 Workshop  
@ SC09  
Due Date: August 1<sup>st</sup>, 2009

## IEEE Transactions on Parallel and Distributed Systems

Special Issue on Many-Task Computing

[Home](#)

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

[Important Dates](#)

[Paper Submission](#)

Abstract Due:	December 1st, 2009
Papers Due:	December 21st, 2009
First Round Decisions:	February 22nd, 2010
Major Revisions if needed:	April 19th, 2010
Second Round Decisions:	May 24th, 2010
Minor Revisions if needed:	June 7th, 2010
Final Decision:	June 21st, 2010
Publication Date:	November, 2010

IEEE TPDS Journal  
Special Issue on MTC  
Due Date: December 1<sup>st</sup>, 2009

### Special Issue Guest Editors

Ian Foster, University of Chicago & Argonne National Laboratory  
Ioan Raicu, University of Chicago  
Yong Zhao, Microsoft



**Dr. Ian Foster** is the Associate Division Director and a Senior Scientist in the Mathematics and Computer Science Division at Argonne National Laboratory, and he is an Arthur Holly Compton Professor in the Department of Computer Science at the University of Chicago. He is also a member of the Grid Forum and with the Globus Alliance as an open source strategist. In 2006, he was appointed director of the Computation Institute, a joint effort of the University of Chicago and Argonne. An earlier project, Strand, received the British Computer Society Award for technical innovation. His research resulted in the development of algorithms for high-performance distributed computing and parallel computing. As a result he is denoted as "the father of the Grid". Foster was also the I-WAY wide-area distributed computing experiment, which connected supercomputers, databases and other high-end resources at 100 labs, the Distributed Systems Laboratory is the nexus of the multi-institute Globus Project, a research and development effort that encourages advances necessary for engineering, business and other fields. Furthermore the Computation Institute addresses many of the most challenging problems facing Grid implementations today. In 2004, he founded Univa Corporation, which was merged with United Devices in 2007 and his honors include the Lovelace Medal of the British Computer Society, the Gordon Bell Prize for high-performance computing (2001), as well as the American Association for the Advancement of Science in 2003. Dr. Foster also serves as PI or Co-PI on projects connected to the DOE Grid Computing Science Alliance, the NASA Information Power Grid project, the NSF Grid Physics Network, GRIDS Center, and International Grid Initiative, and other DOE and NSF programs. His research is supported by DOE, NSF, NASA, Microsoft, and IBM.



**Dr. Ioan Raicu** holds a Ph.D. in Computer Science from University of Chicago under the guidance of Dr. Ian Foster. He is a 3-year award winning Ph.D. student at the University of Chicago Research Center. His research work and interests are in the general area of distributed systems. His dissertation work focused on this relationship (MTC), which aims to bridge the gap between two predominant paradigms from distributed systems, High-Throughput Computing (HTC) and High-Performance Computing (HPC). For the last five years focused on defining and exploring both the theory and practical aspects of realizing MTC across a wide range of large-scale systems. He is interested in efficient task dispatch and execution systems, resource provisioning, data management, scheduling, and performance evaluation. His research is funded by the NASA Ames Research Center Graduate Student Research Program, as well as the DOE Office of Advanced Scientific Computing. His research interests include resource management in large scale distributed systems with a focus on many-task computing, data intensive computing, cloud computing

# More Information

- More information:
  - Other publications: <http://people.cs.uchicago.edu/~iraicu/>
  - Falkon: <http://dev.globus.org/wiki/Incubator/Falkon>
  - Swift: <http://www.ci.uchicago.edu/swift/index.php>
- Funding:
  - **NASA**: Ames Research Center, GSRP
  - **DOE**: Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy
  - **NSF**: TeraGrid
- Relevant activities:
  - ACM MTAGS09 Workshop at Supercomputing 2009
    - <http://dsl.cs.uchicago.edu/MTAGS09/>
  - Special Issue on MTC in IEEE TPDS Journal
    - [http://dsl.cs.uchicago.edu/TPDS\\_MTC/](http://dsl.cs.uchicago.edu/TPDS_MTC/)