Many-task Computing:

Bridging the Gap between High-Throughput Computing and High-Performance Computing

Ioan Raicu
Distributed Systems Laboratory
Computer Science Department
University of Chicago

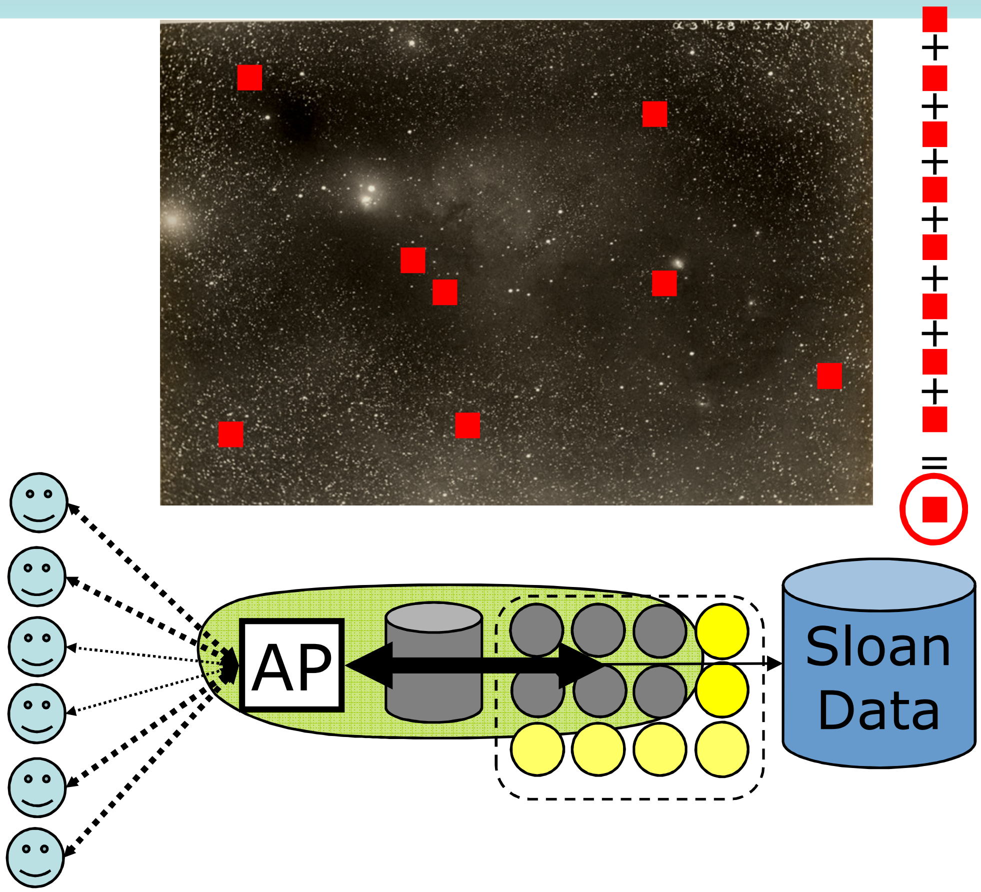
Dissertation Advisor:
Ian Foster, University of Chicago & Argonne National Laboratory

Committee Members :
Rick Stevens, University of Chicago & Argonne National Laboratory
Alex Szalay, John Hopkins University

Dissertation Defense
February 12th, 2009

Motivating Use Case: AstroPortal

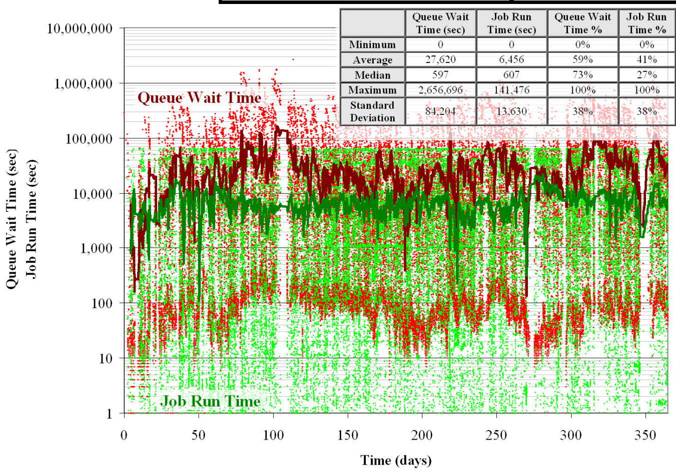
- Purpose
  - On-demand "stacks" of random locations within ~10TB dataset
- Challenge
  - Processing Costs:
    - O(100ms) per object
  - Data Intensive:
    - 40MB:1sec
  - Rapid access to 10-10K "random" files
  - Time-varying load



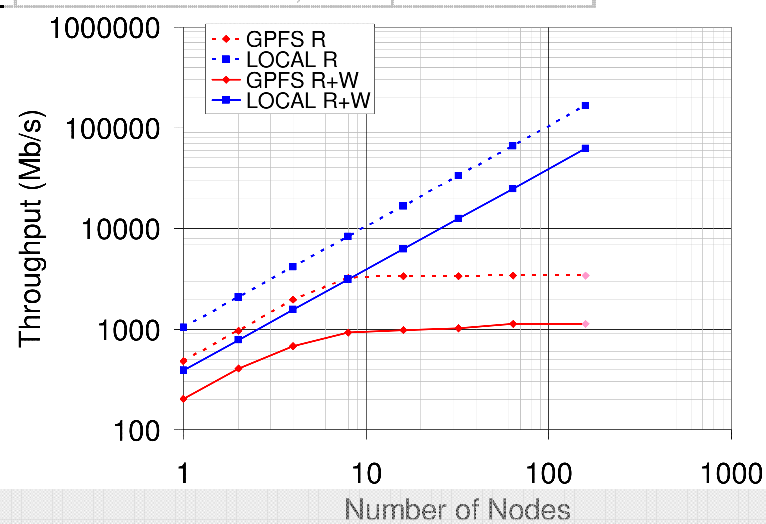
Challenges

1. Slow job dispatch rates
2. Long queue times
3. Poor shared/parallel file system scaling

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



	Queue Wait Time (sec)	Job Run Time (sec)	Queue Wait Time %	Job Run Time %
Minimum	0	0	0%	0%
Average	27,620	6,456	59%	41%
Median	597	607	73%	27%
Maximum	2,656,696	141,476	100%	100%
Standard Deviation	84,204	13,630	38%	38%



[UC07] "Harnessing Grid Resources with Data-Centric Task Farms"

High-Throughput Computing & High-Performance Computing

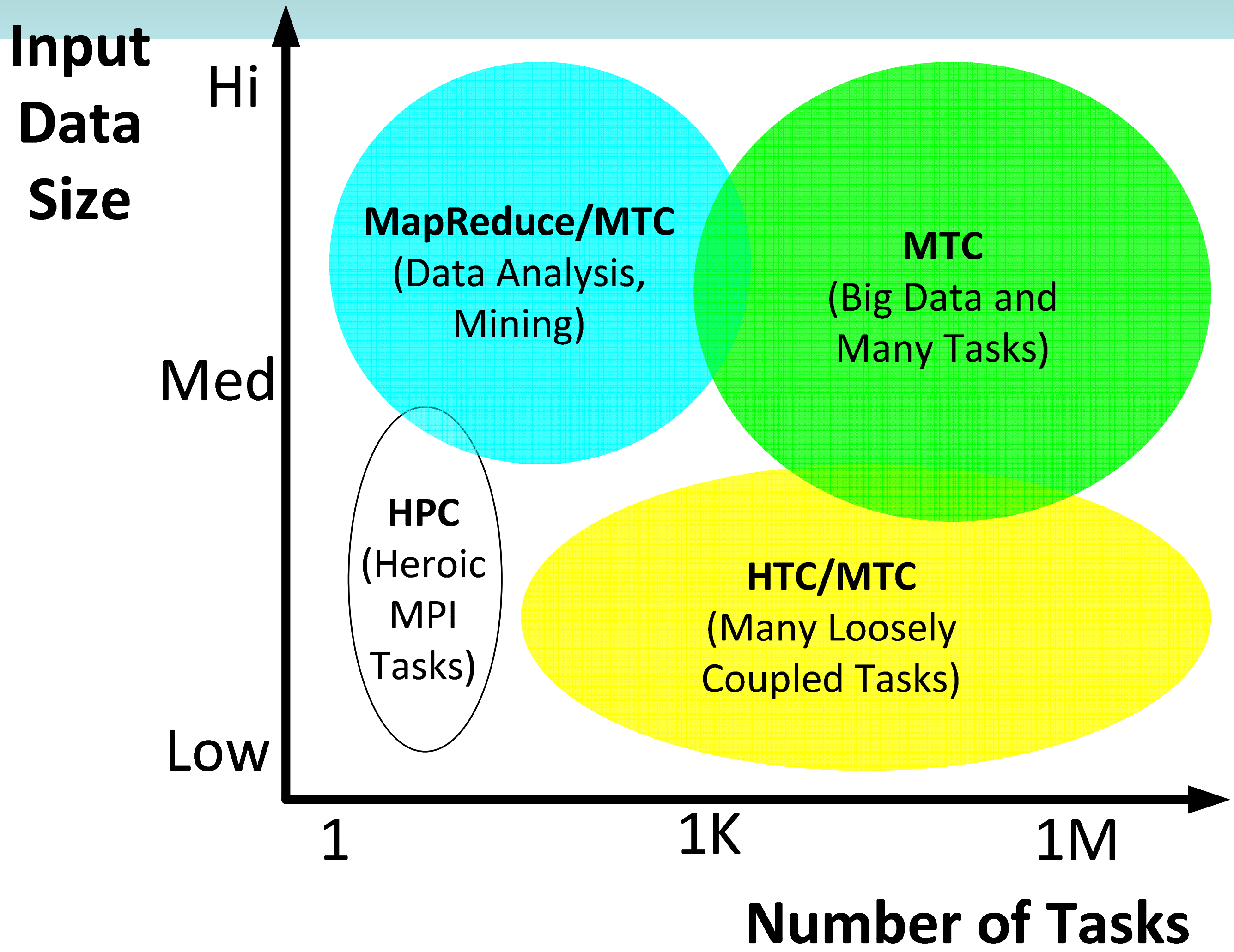
- HTC: High-Throughput Computing
  - Typically applied in clusters and grids
  - Loosely-coupled applications with sequential jobs
  - Large amounts of computing for long periods of times
  - Measured in operations per month or years
- HPC: High-Performance Computing
  - Synonymous with supercomputing
  - 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

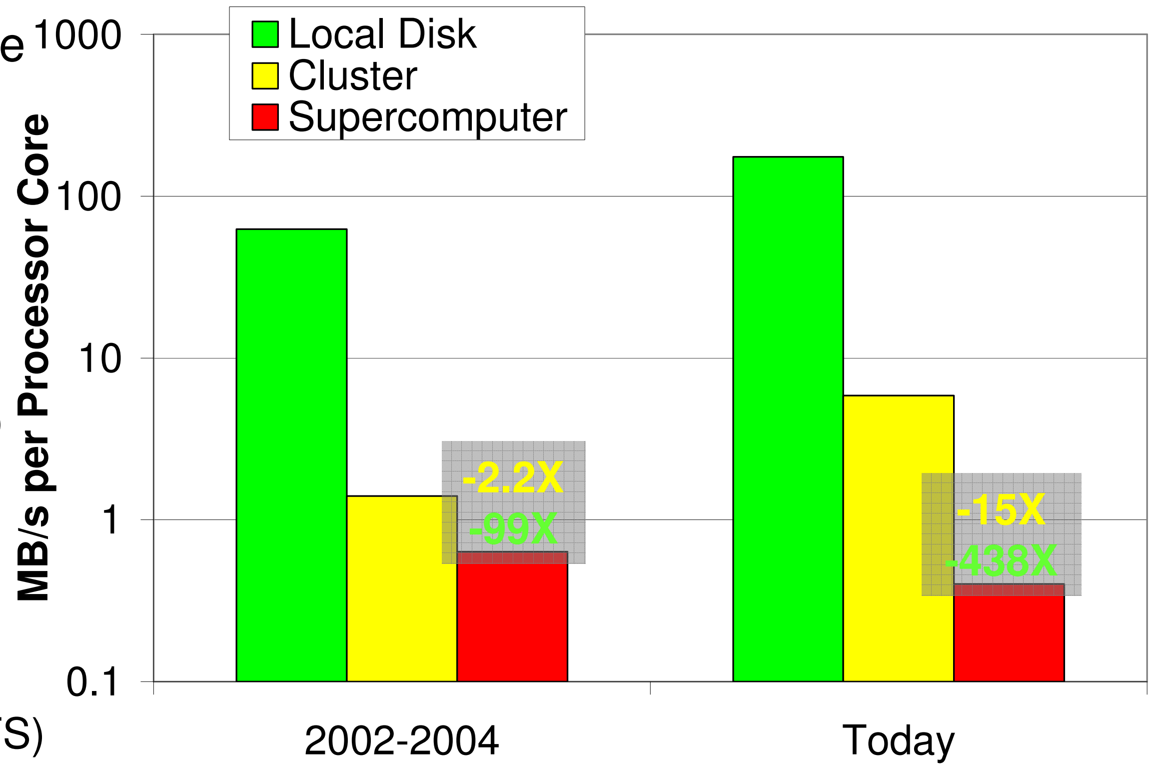
[MTAGS08 Workshop] Workshop on Many-Task Computing on Grids and Supercomputers 2008
[SC08] “Towards Loosely-Coupled Programming on Petascale Systems”
[MTAGS08] “Many-Task Computing for Grids and Supercomputers”

Problem Space



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)



Presentation Focus

- [JS09] "Middleware Support for Many-Task Computing", under preparation
- [HPDC09] "The Quest for Scalable Support of Data Intensive Workloads in Distributed Systems", under review
- [DIDC09] "Towards Data Intensive Many-Task Computing" , under review
- [SC08] "Towards Loosely-Coupled Programming on Petascale Systems"
- [MTAGS08 Workshop] Workshop on Many-Task Computing on Grids and Supercomputers
- [MTAGS08] "Many-Task Computing for Grids and Supercomputers"
- [MTAGS08] "Design and Evaluation of a Collective I/O Model for Loosely-coupled Petascale Programming"
- [GCE08] "Cloud Computing and Grid Computing 360-Degree Compared"
- [SWF08] "Scientific Workflow Systems for 21st Century e-Science, New Bottle or New Wine?"
- [DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"
- [TG08] "Data Intensive Scalable Computing on TeraGrid: A Comparison of MapReduce and Swift"
- [GlobusWorld08] "Managing and Executing Loosely Coupled Large Scale Applications on Clusters, Grids, and Supercomputers"
- [NOVA08] "Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments"
- [UC07] "Harnessing Grid Resources with Data-Centric Task Farms"
- [Globus07] "Falkon: A Proposal for Project Globus Incubation"
- [SC07] "Falkon: a Fast and Light-weight tasK executiON framework"
- [MSES07] "A Data Diffusion Approach to Large Scale Scientific Exploration"
- [SWF07] "Swift: Fast, Reliable, Loosely Coupled Parallel Computation"
- [TG07] "Dynamic Resource Provisioning in Grid Environments"
- [NASA06-08] "Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets"
- [SC06] "Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets"
- [TG06] "AstroPortal: A Science Gateway for Large-scale Astronomy Data Analysis"
- [NSF06] "The Importance of Data Locality in Distributed Computing Applications"

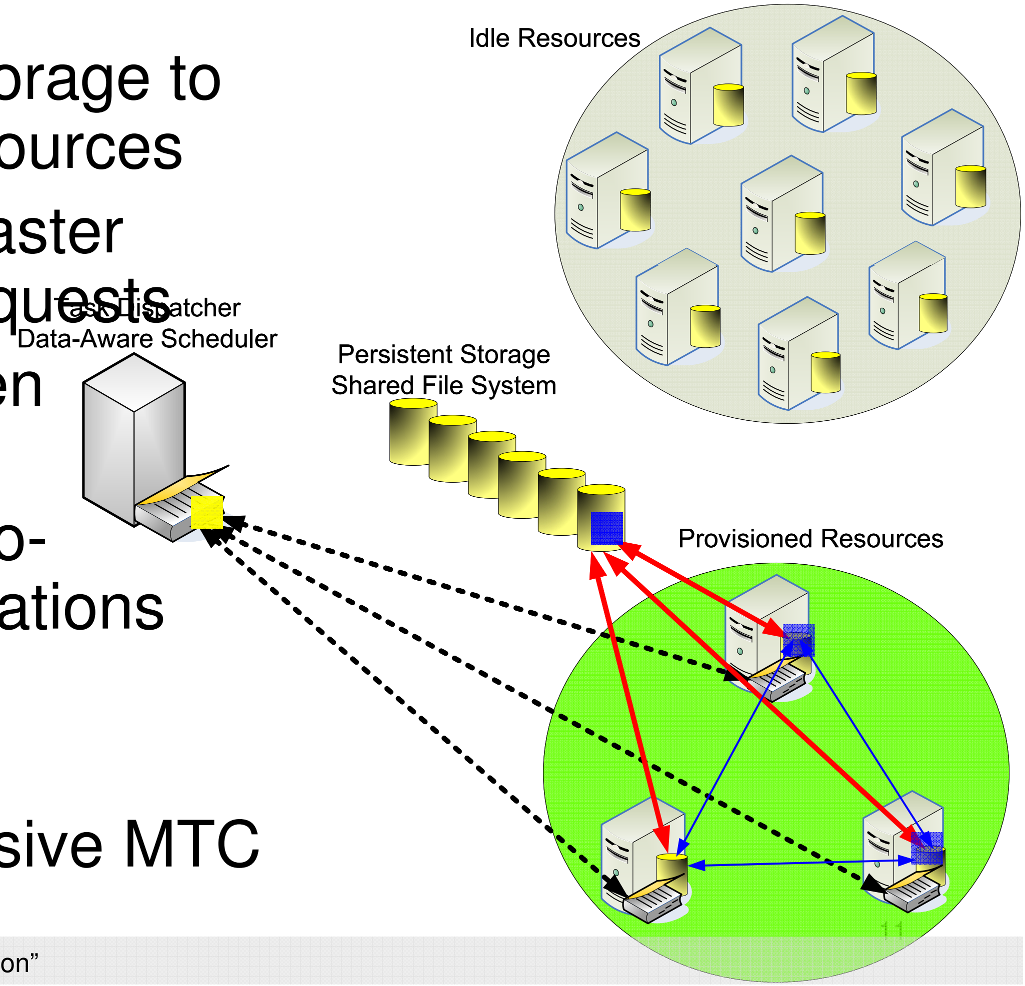
Hypothesis

"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



[DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"

Data Diffusion: Abstract Model

- Captures data diffusion properties
- Models the efficiency and speedup of entire workloads
- Base definitions
  - Data Stores (Persistent & Transient)
  - Compute resources (transient)
  - Data Objects
  - Tasks

[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review
[UC07] "Harnessing Grid Resources with Data-Centric Task Farms"

Data Diffusion: Execution Model

- Dispatch Policy
  - first-available (FA), max-compute-util (MCU), max-cache-hit (MCH), good-cache-compute (GCC)
- Caching Policy
  - random, FIFO, LRU, LFU, 2
- Replay Policy
- Data Fetch Policy
- Resource Acquisition Policy
  - one-at-a-time, additive, exponential, all-at-once
- Resource Release Policy

[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review
[UC07] "Harnessing Grid Resources with Data-Centric Task Farms"

Data Diffusion: Performance Model

Average time to complete task i	• $TK_i = C + R_l*HR_l + R_c*HR_c + R_s*HR_s$
Time to complete an entire workload D	• $T_N(D) = \sum_{i=1}^{K} TK_i$
Speedup	• $SP = T_1(D) / T_N(D)$
Efficiency	• $EF = SP / N$
Arrival Rate	• $A = (N*P/T)*K$
Utilization	• $U = A*T/(N*P)$

[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

Data Diffusion: O(NM)-Competitive Caching

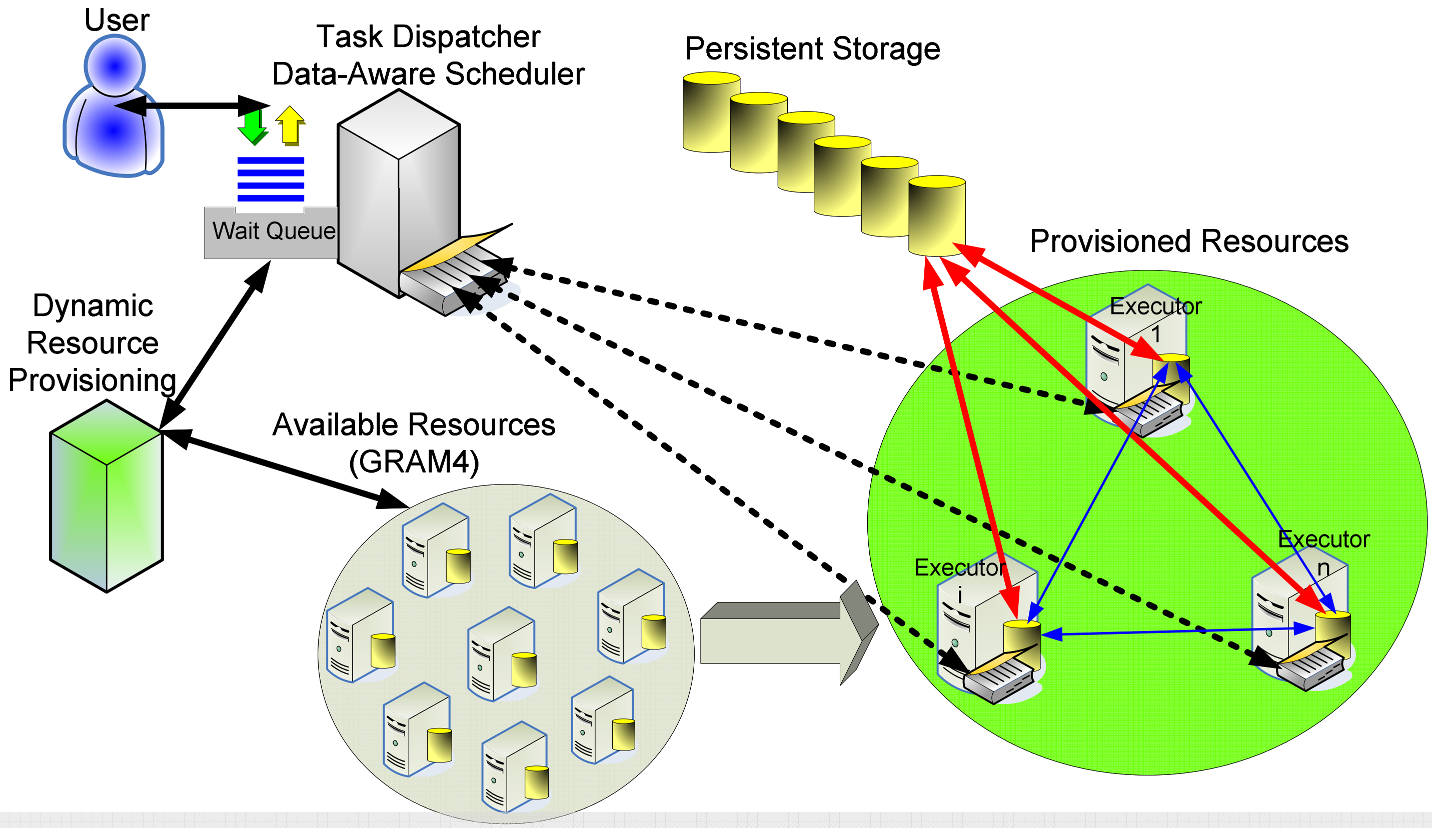
- Competitive ratio (worst case) between online algorithm and offline optimal
  - Measures the quality of the online algorithm, independent of data access patterns or workload characteristics
- The relation we prove to establish that 2Mark is O(NM)-competitive
  - 2Mark $(\sigma) \leq (NM + 2M / s + NM /(s + v)) \cdot$ OPT (σ) for all sequences σ

Philip Little, Amitabh Chaudhary, University of Notre Dame

[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

From Theory to Practice

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



[DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"
[SC07] "Falkon: a Fast and Light-weight tasK executiON framework"

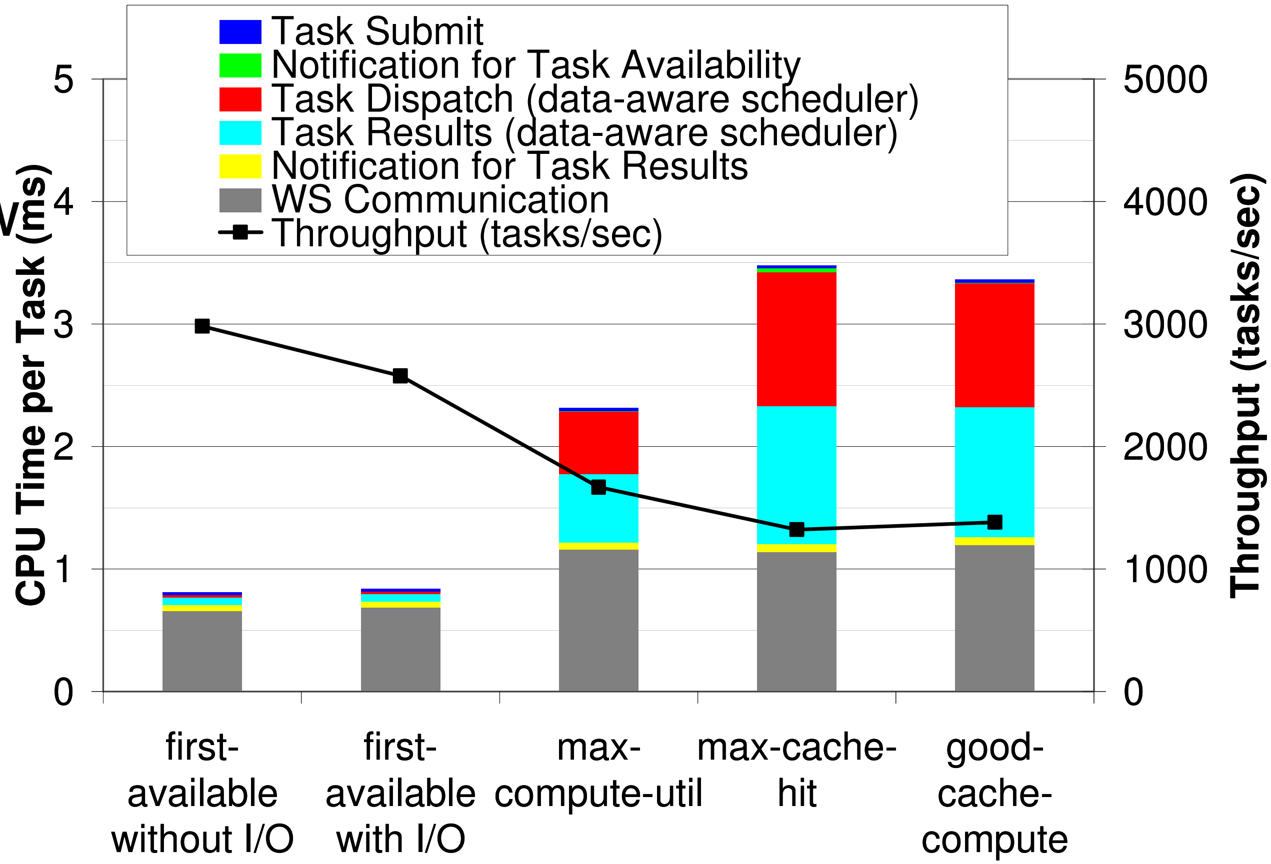
Scheduling Policies

- 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

[DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

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



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

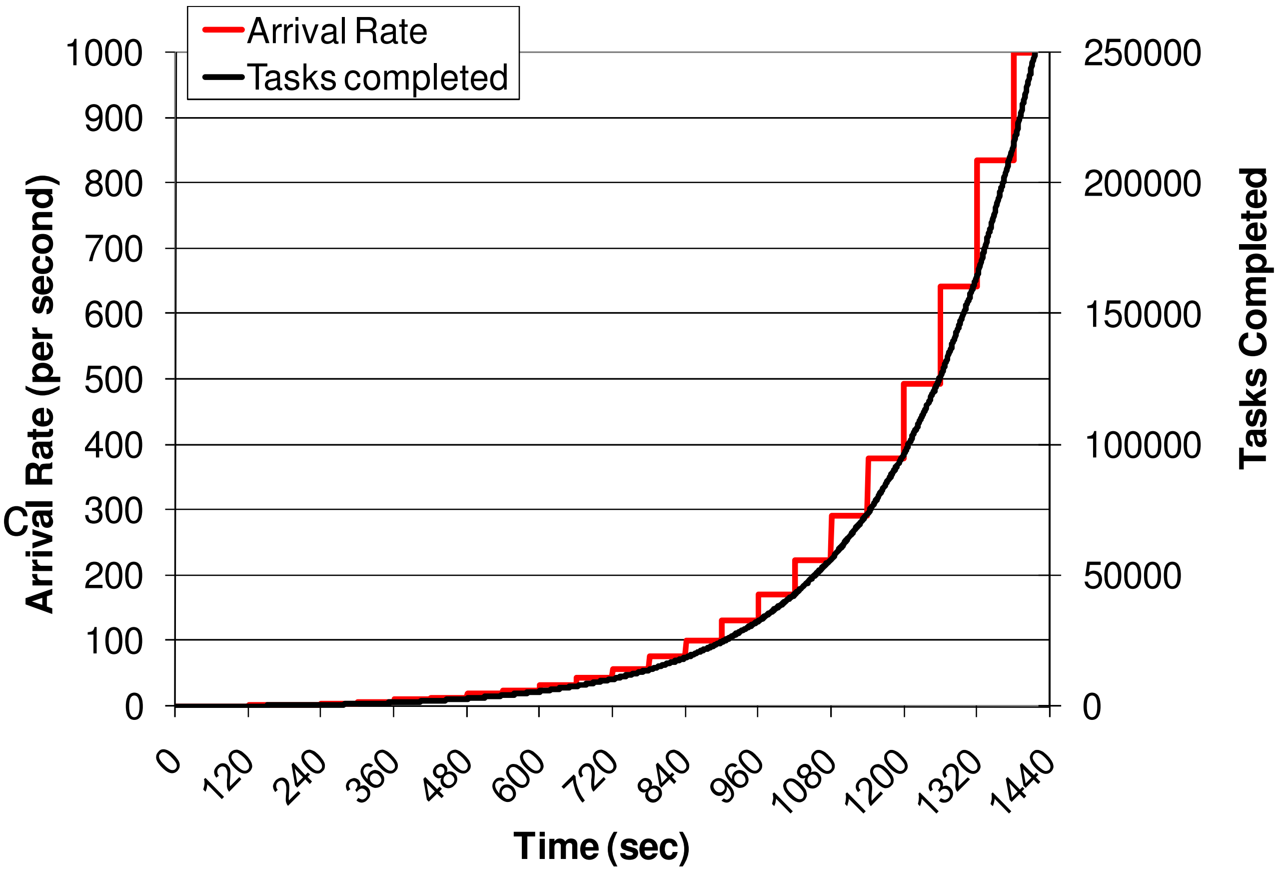
Workloads

- 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

[DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

Monotonically Increasing Workload

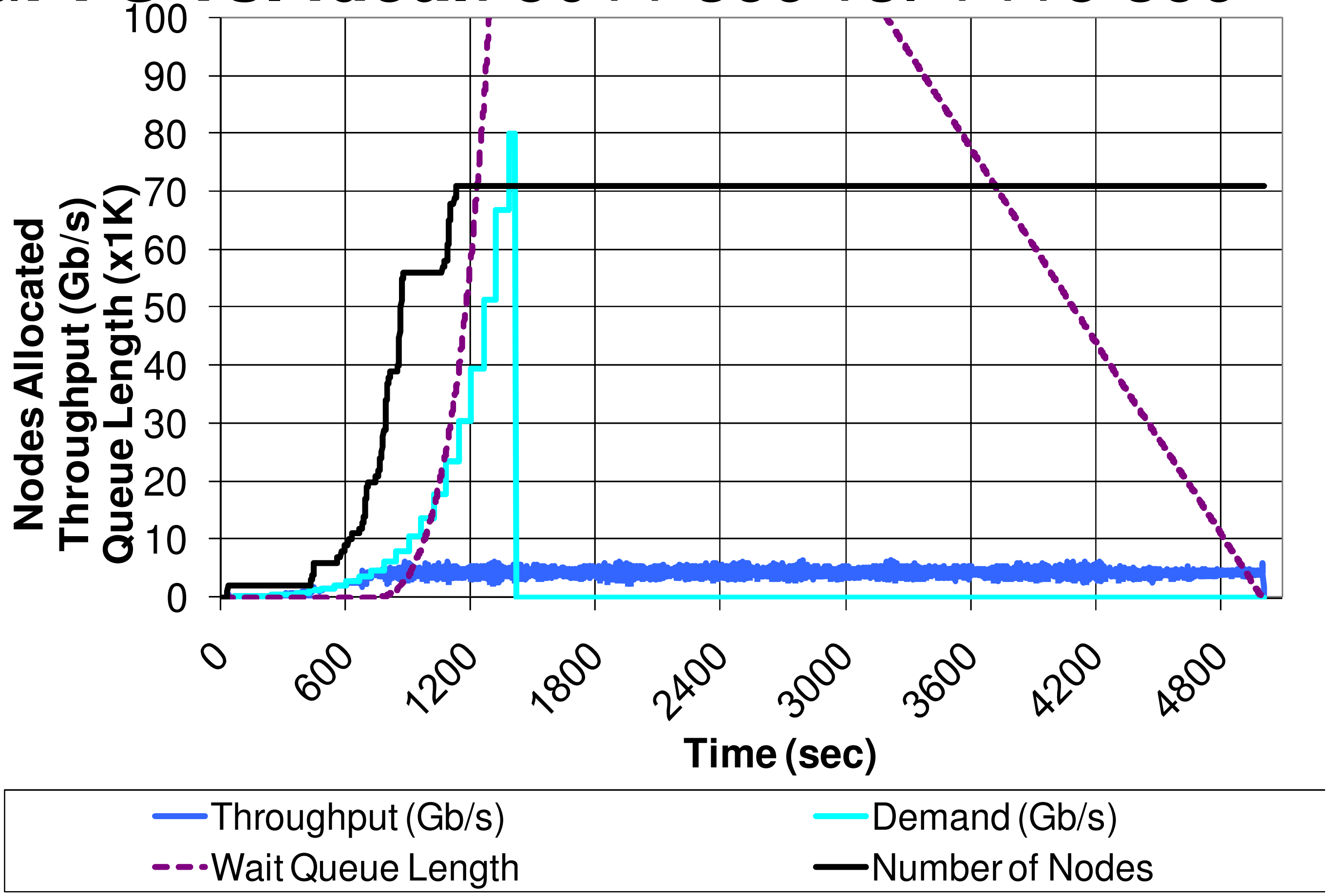
- 250K tasks
  - 10MB reads
  - 10ms compute
- Vary arrival rate:
  - Min: 1 task/sec
  - Increment function: CEILING(*1.3)
  - Max: 1000 tasks/sec
- 128 processors
- Ideal case:
  - 1415 sec
  - 80Gb/s peak throughput



[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

Monotonically Increasing Workload First-available (GPFS)

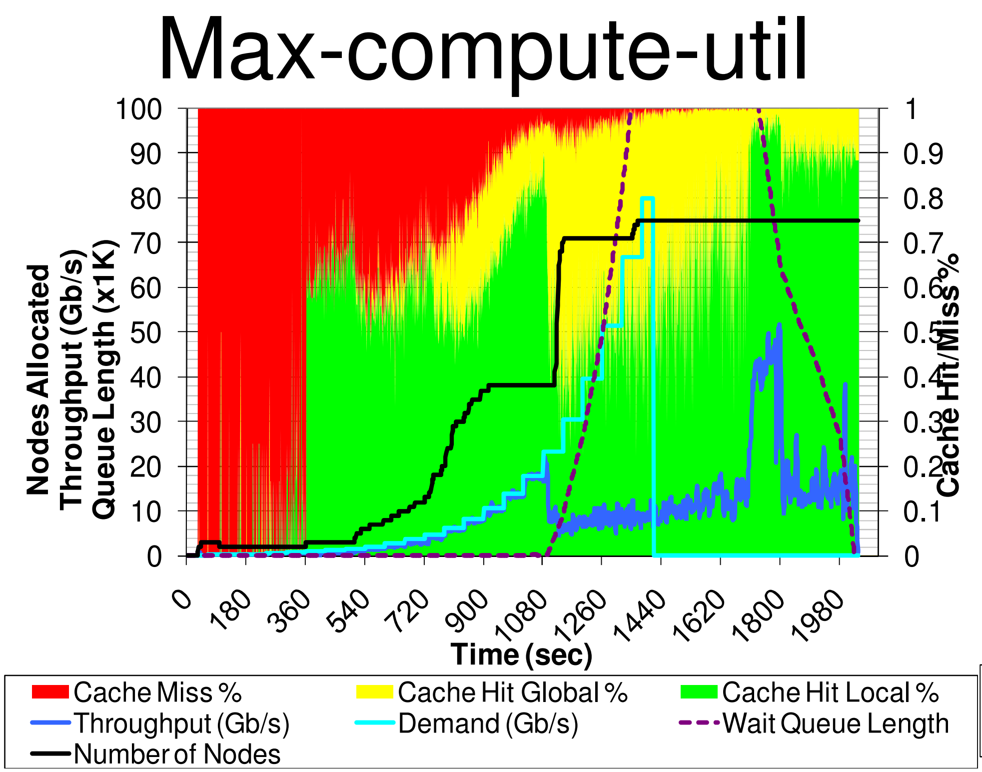
- **GPFS vs. ideal**: 5011 sec vs. 1415 sec

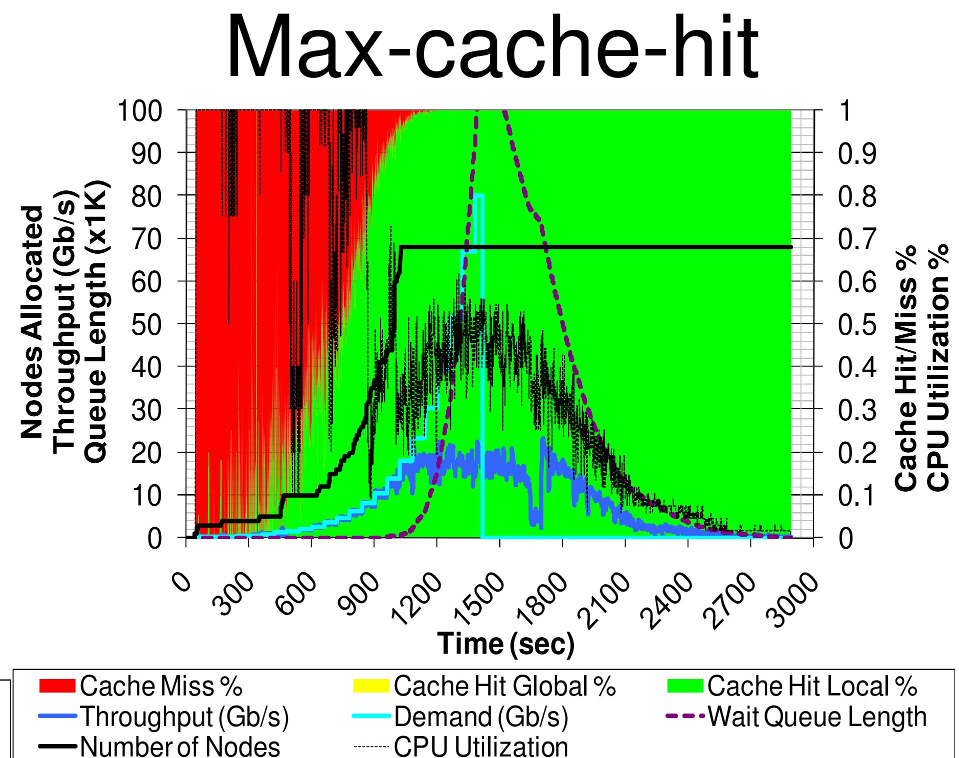


[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review

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

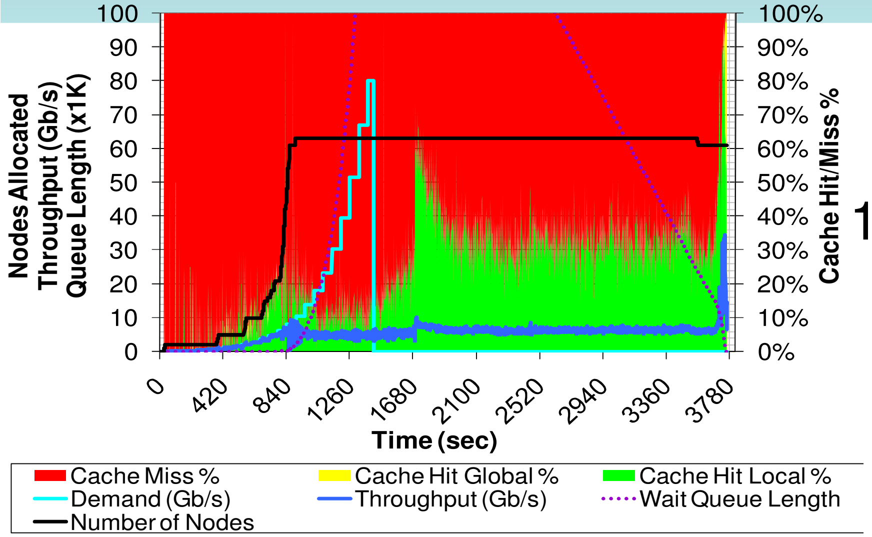
Monotonically Increasing Workload Max-compute-util & Max-cache-hit





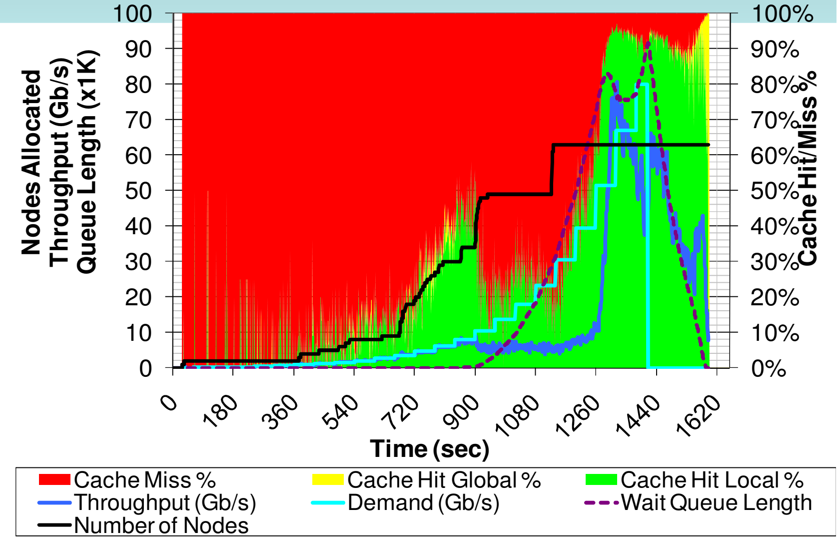
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

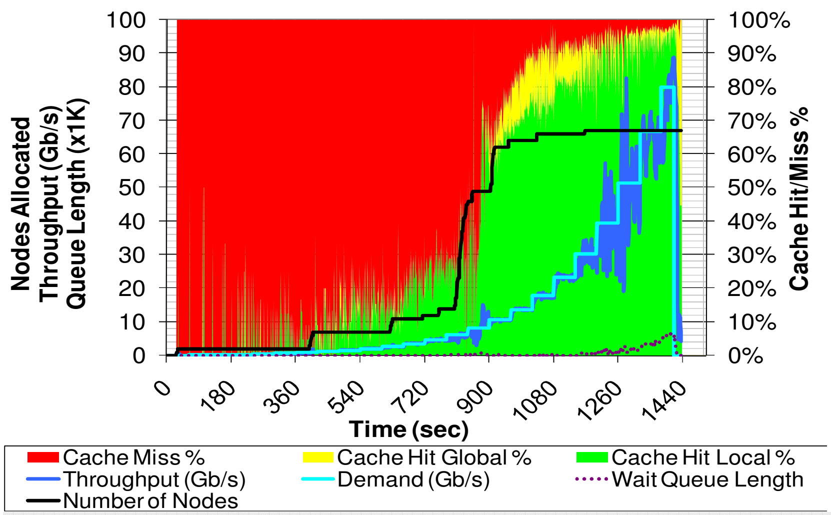
Monotonically Increasing Workload
Good-cache-compute



← 1GB

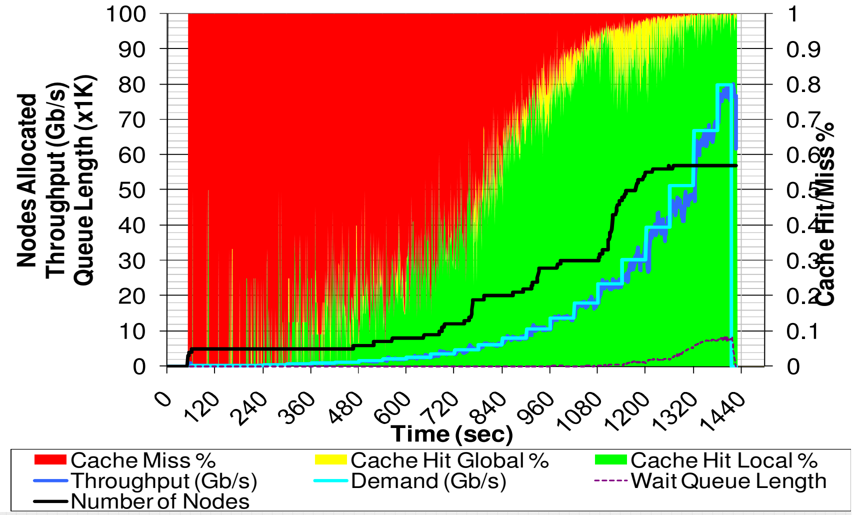
1.5GB →





← 2GB

4GB →

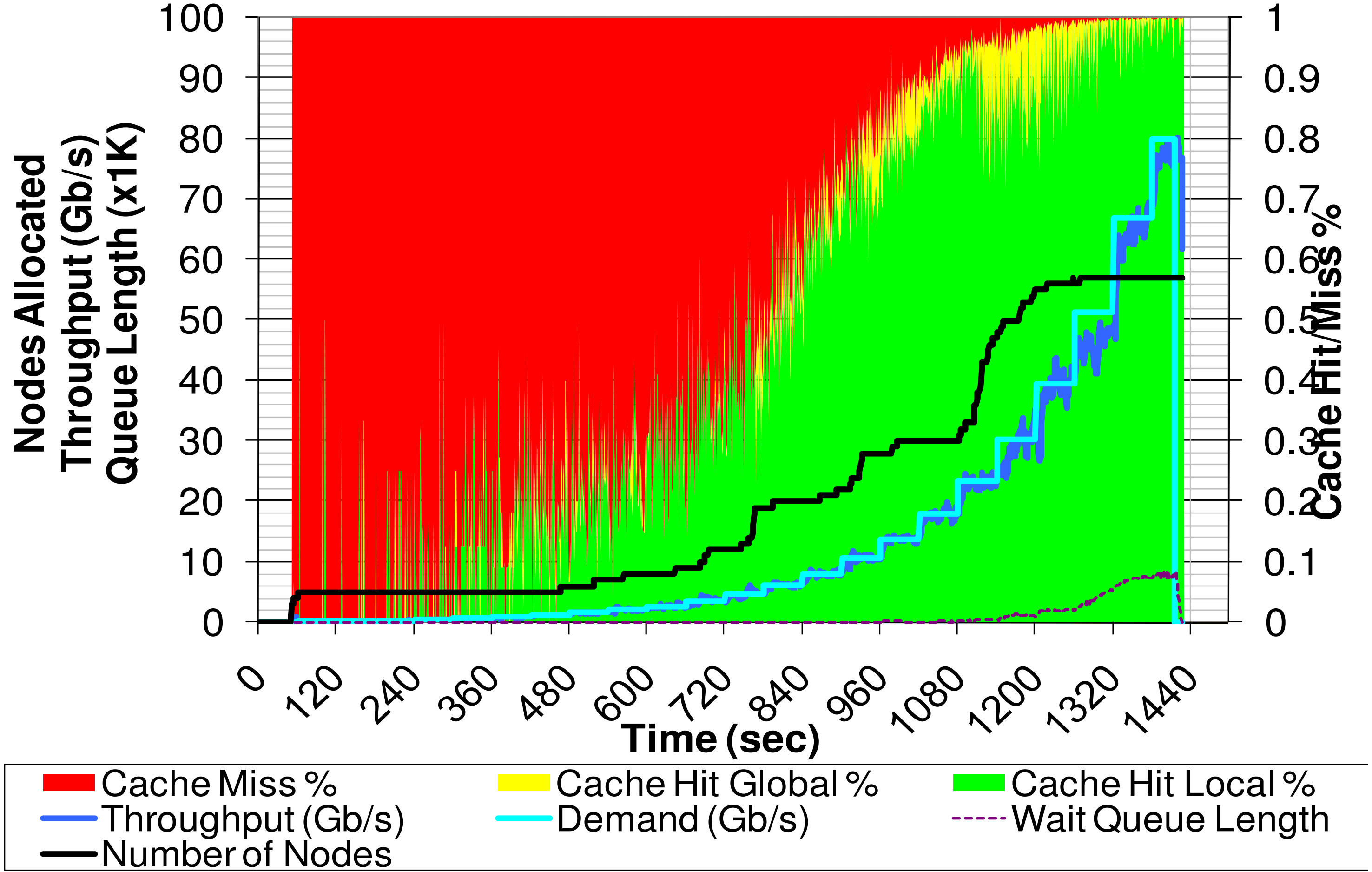


[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review

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

Monotonically Increasing Workload Good-cache-compute

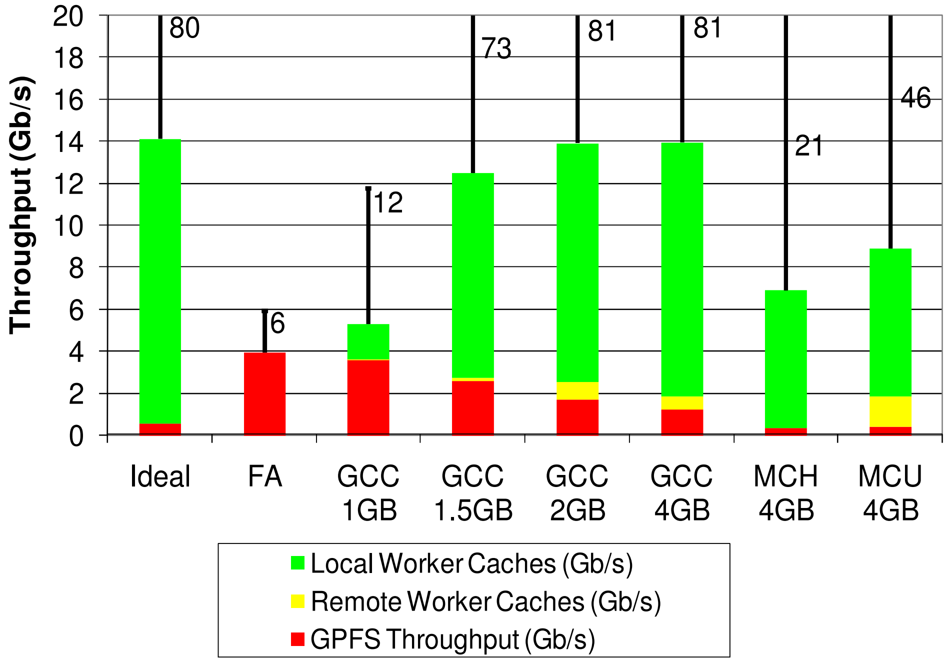
- **Data Diffusion vs. ideal**: 1436 sec vs 1415 sec



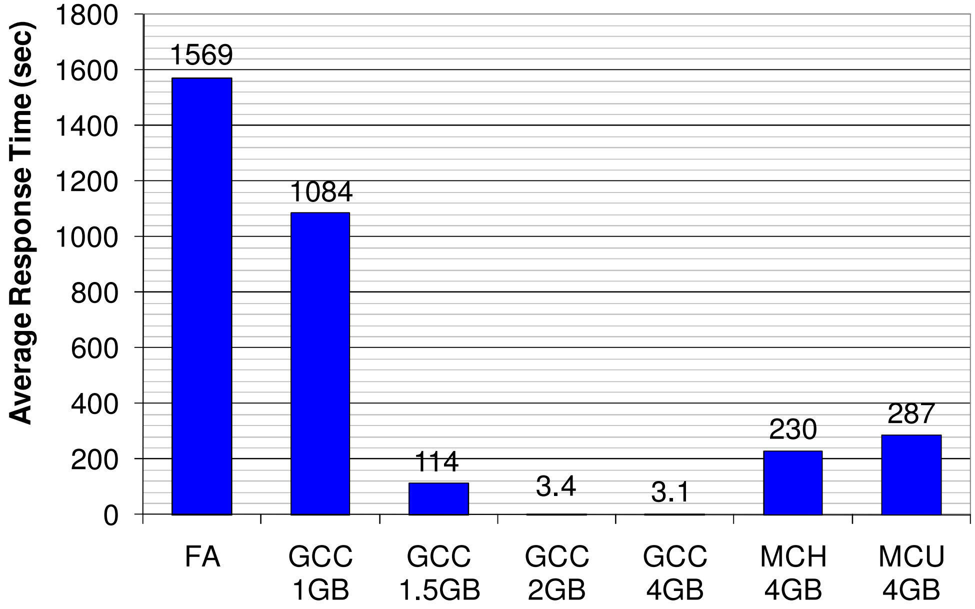
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review

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

Monotonically Increasing Workload Throughput and Response Time



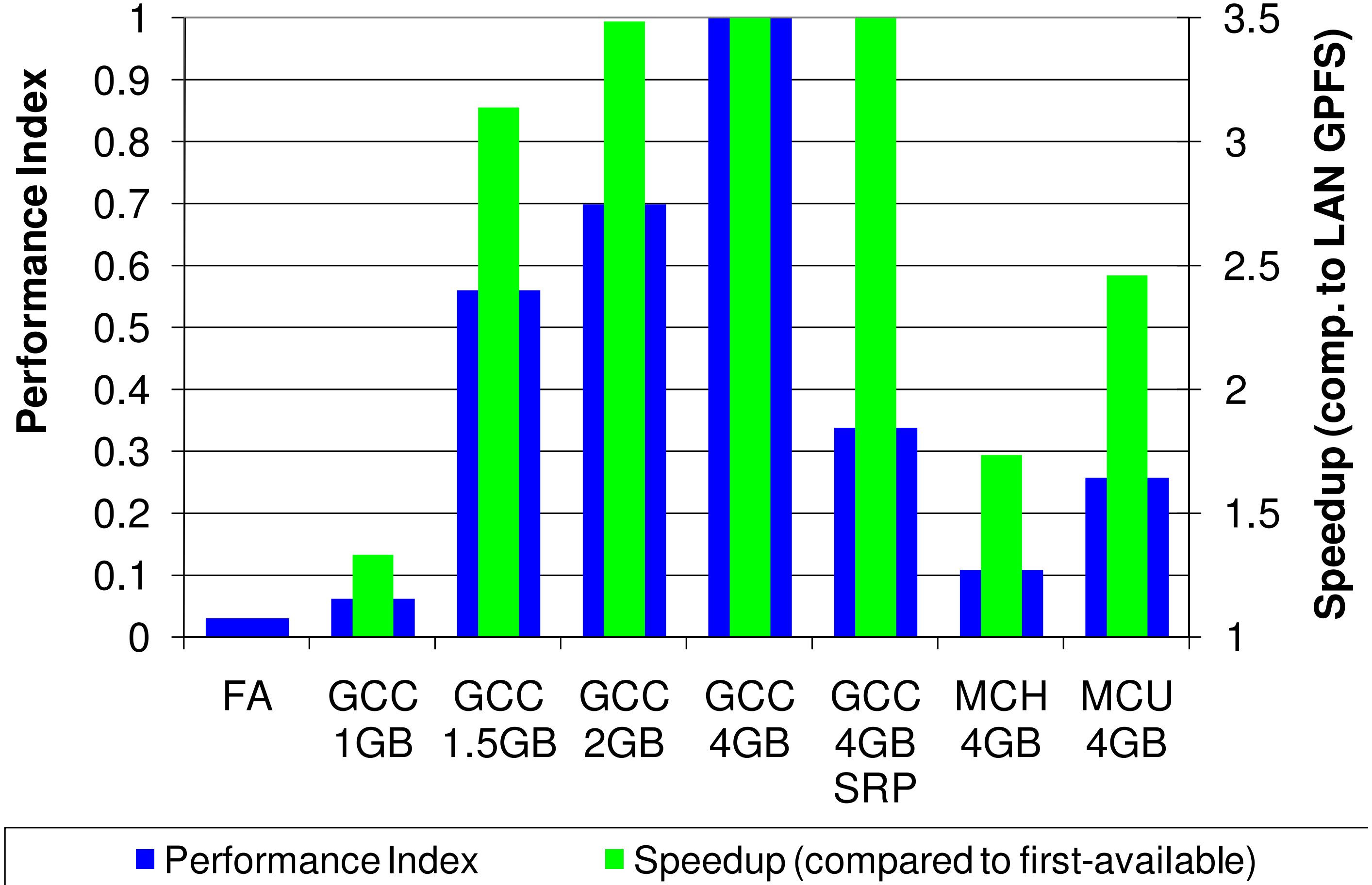
- ← Throughput:
  - Average: 14Gb/s vs 4Gb/s
  - Peak: 81Gb/s vs. 6Gb/s
- Response Time →
  - 3 sec vs 1569 sec → 506X



[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

Monotonically Increasing Workload Performance Index and Speedup

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



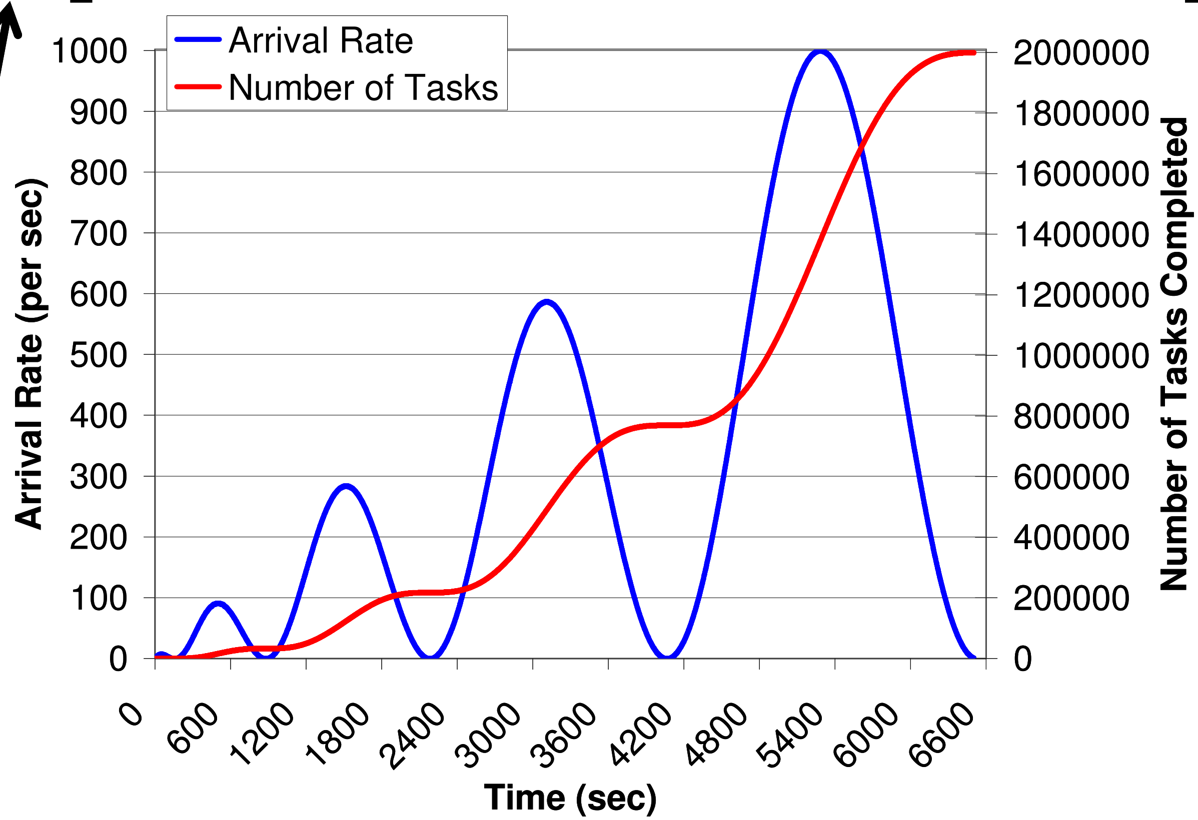
[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review

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

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

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

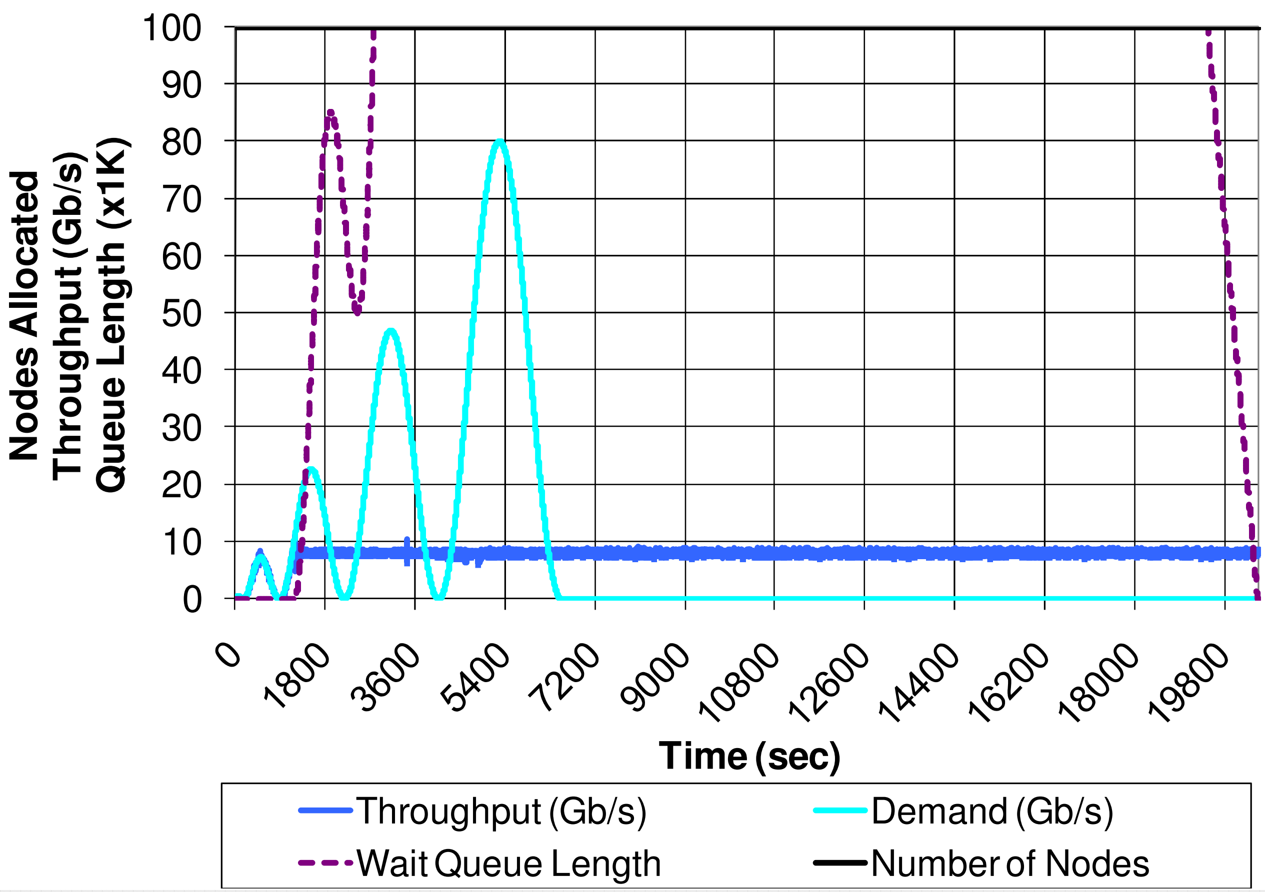


[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review

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

Sine-Wave Workload First-available (GPFS)

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

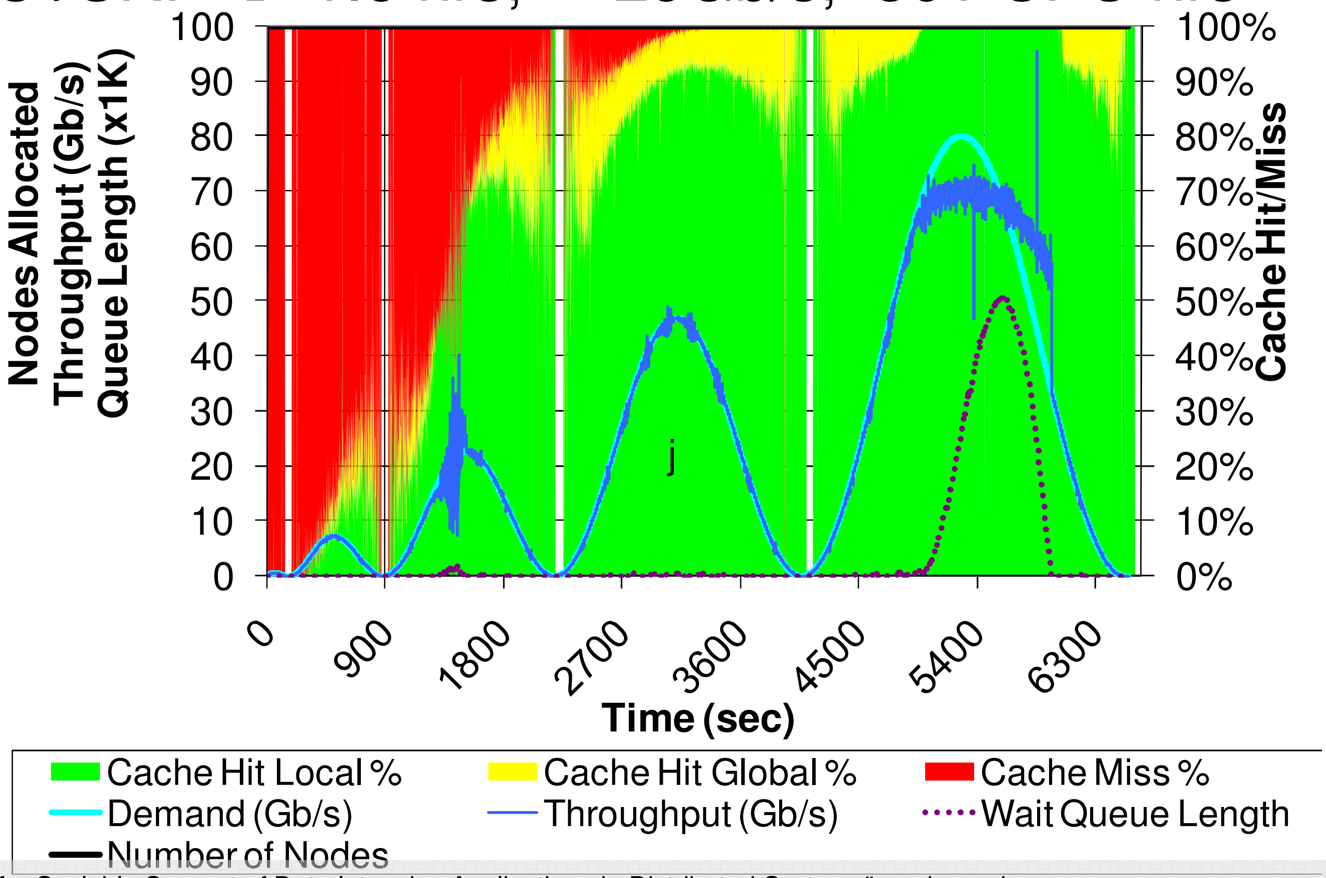


[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review

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

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

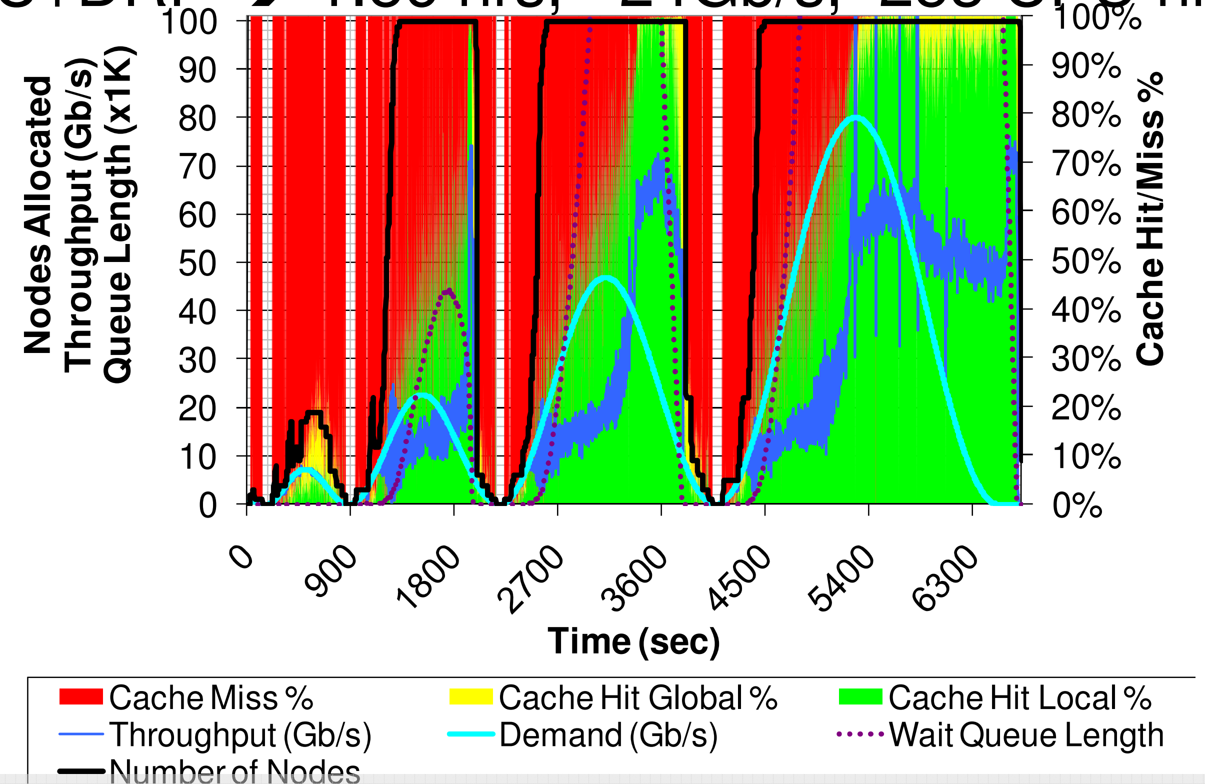


[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review

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

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



[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

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

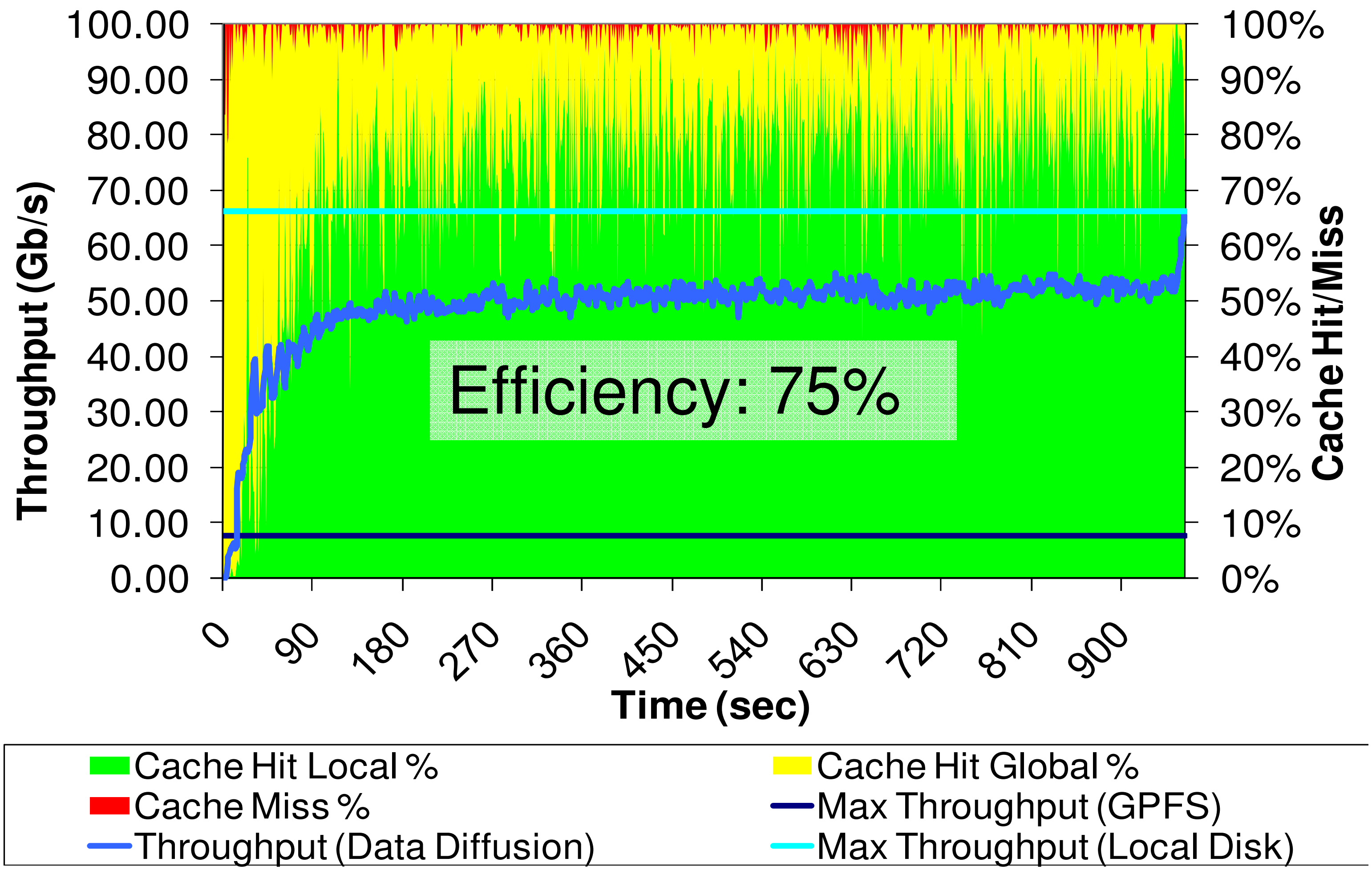
M[i,j] = F(A[i],B[j])

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

[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review

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

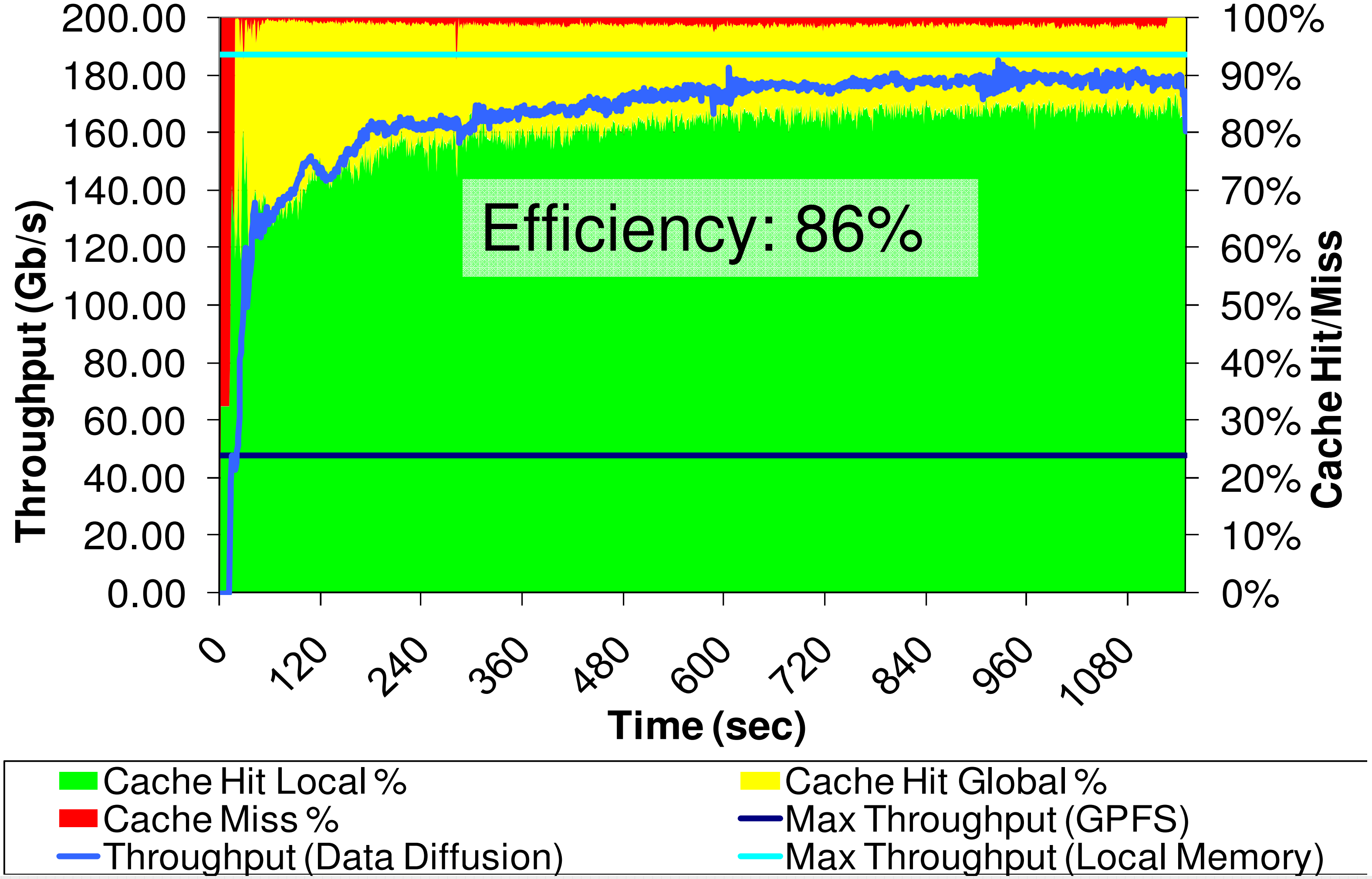
All-Pairs Workload
500x500 on 200 CPUs



[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review

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

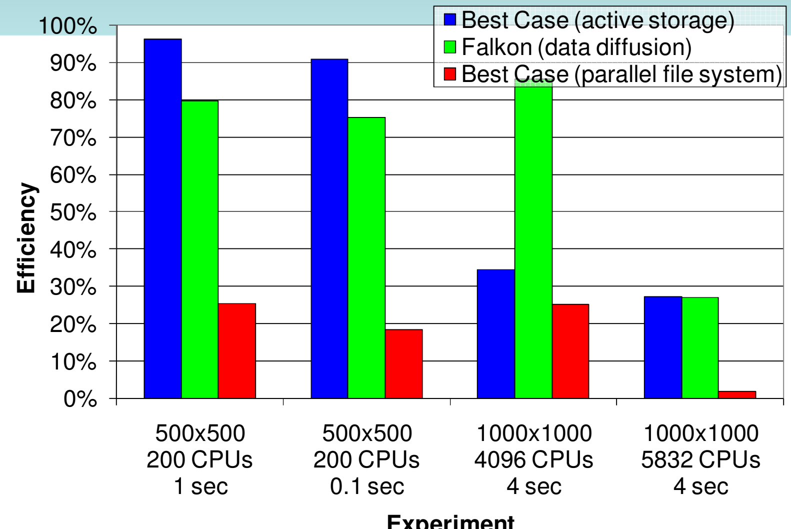
All-Pairs Workload 1000x1000 on 4K emulated CPUs



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

Christopher Moretti, Douglas Thain, University of Notre Dame



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

[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

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 << 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)

[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

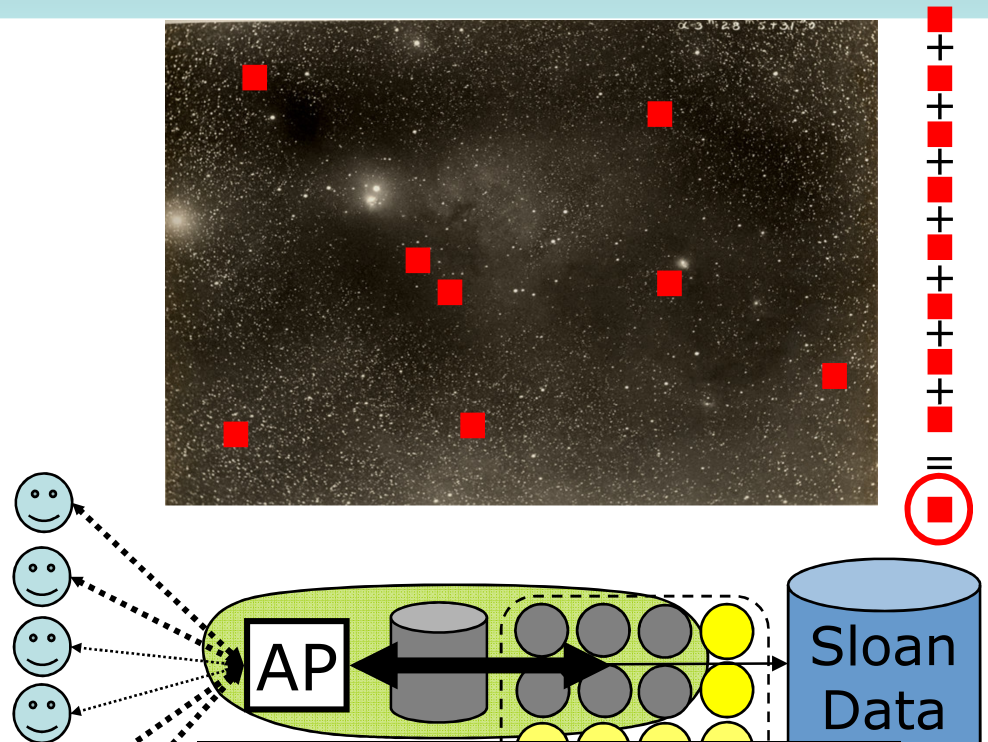
Data Diffusion vs. Others

- [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 + Falkon + Data Diffusion is arguably a more generic and powerful solution than MapReduce*

Image Stacking Workload Astronomy Application

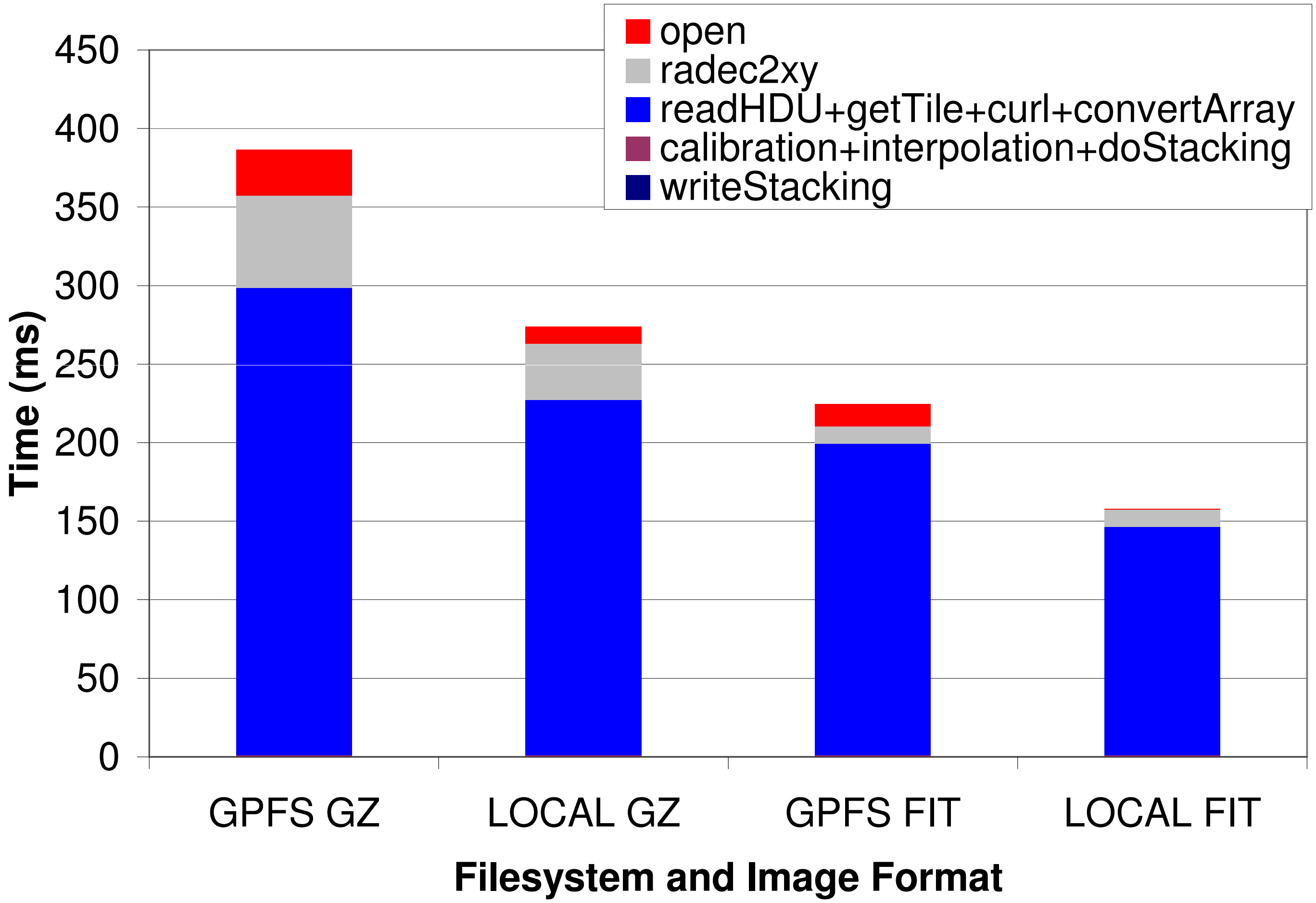
- Purpose
  - On-demand "stacks" of random locations within ~10TB dataset
- Challenge
  - Processing Costs:
    - O(100ms) per object
  - Data Intensive:
    - 40MB:1sec
  - Rapid access to 10-10K "random" files
  - Time-varying load



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

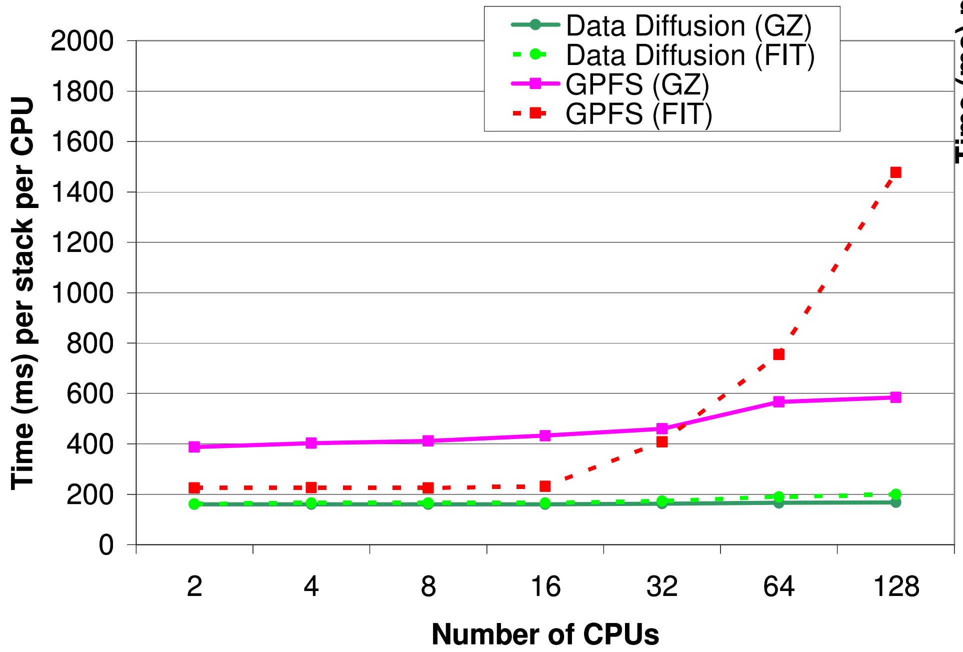


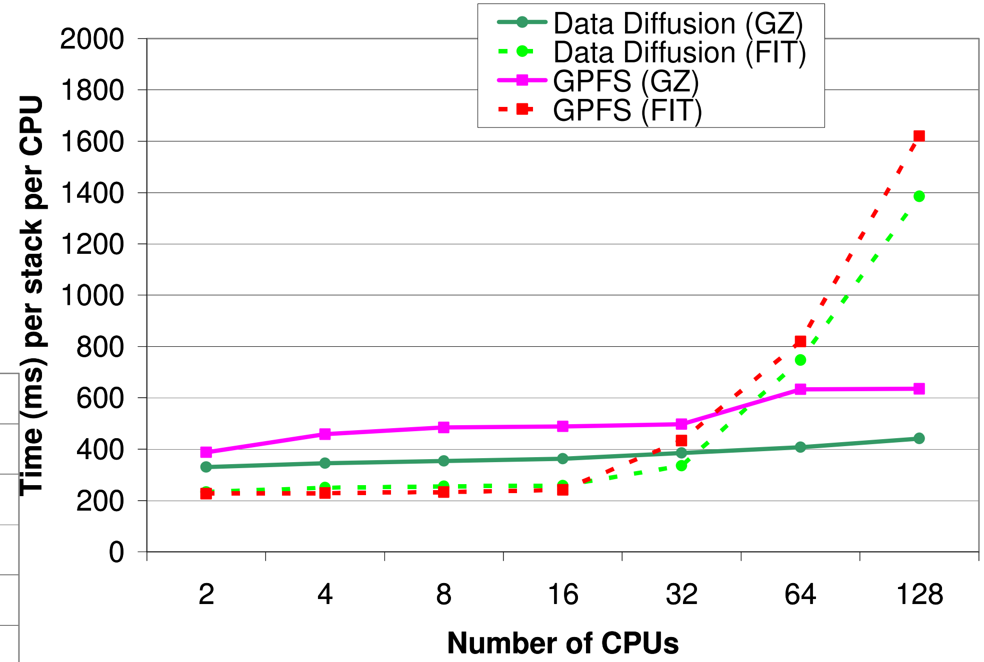
[DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"

Image Stacking Workload Varying Scale

Low data locality →

- Similar (but better) performance to GPFS





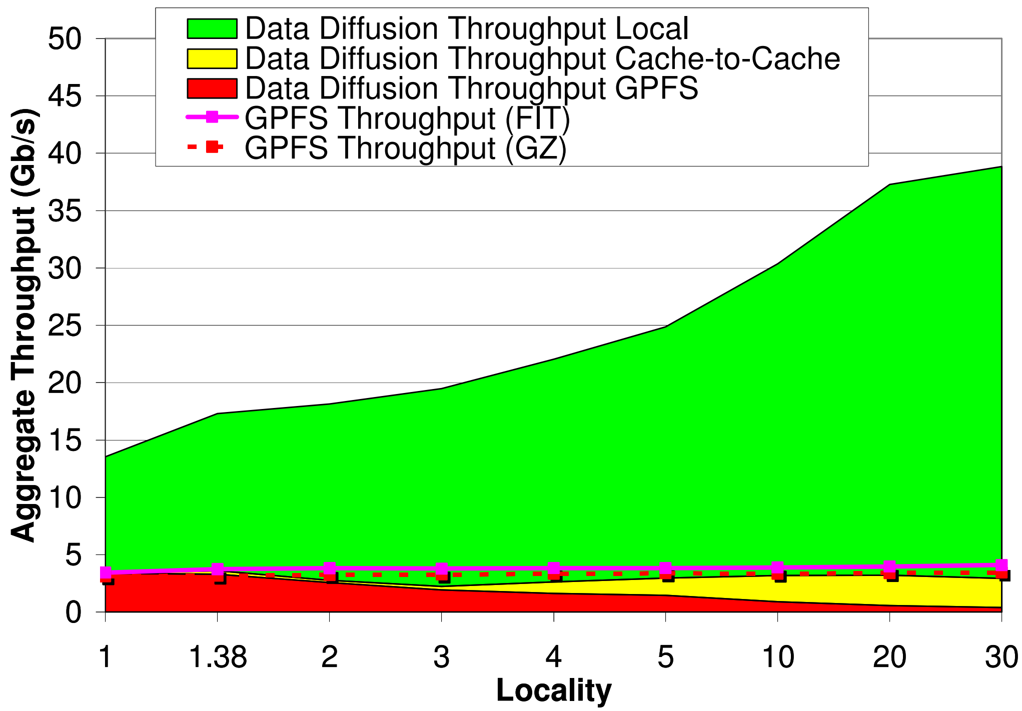
←High data locality

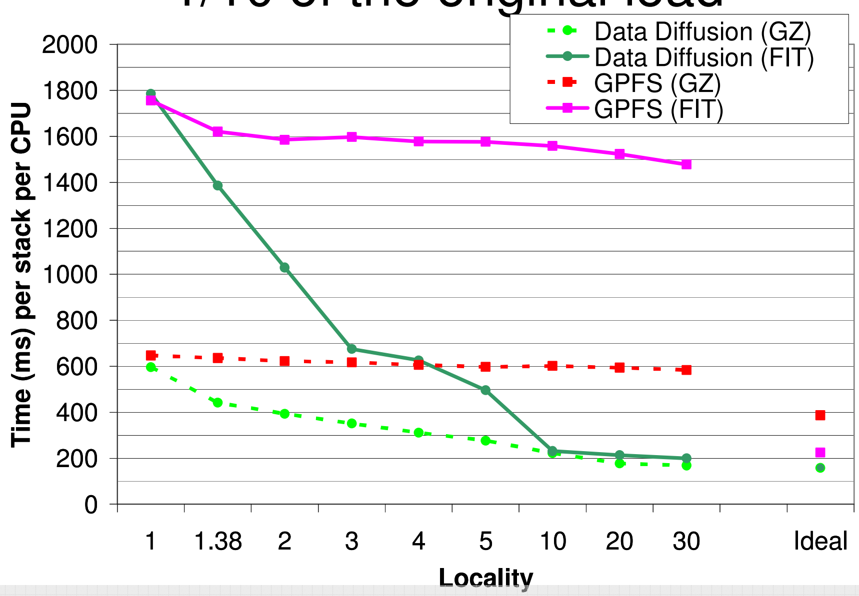
- Near perfect scalability

[DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"

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



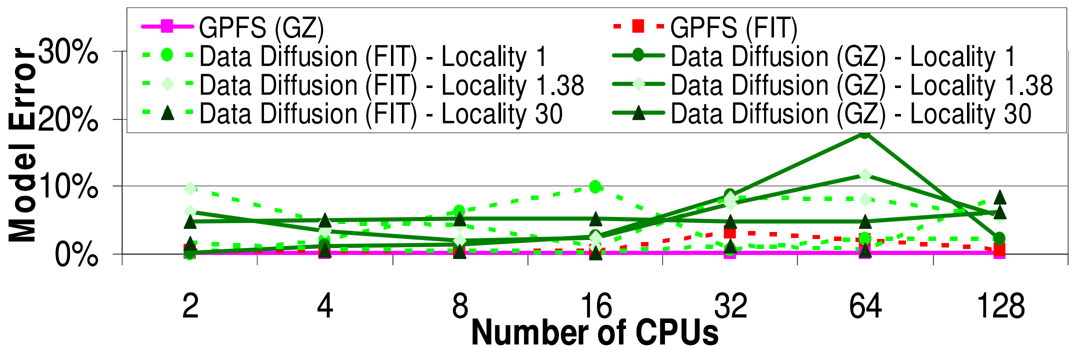


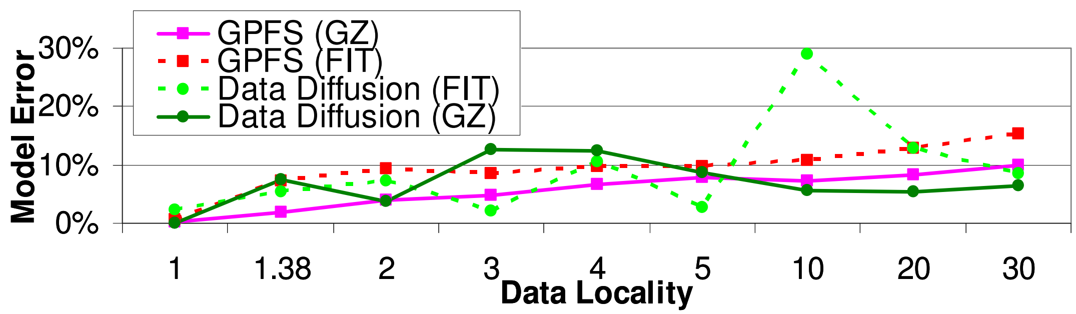
- Big performance gains as locality increases

[DADC08] "Accelerating Large-scale Data Exploration through Data Diffusion"

Image Stacking Workload Abstract Model Validation

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





[HPDC09] "The Quest for Scalable Support of Data Intensive Applications in Distributed Systems", under review
[DIDC09] "Towards Data Intensive Many-Task Computing", under review

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+

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

Contributions

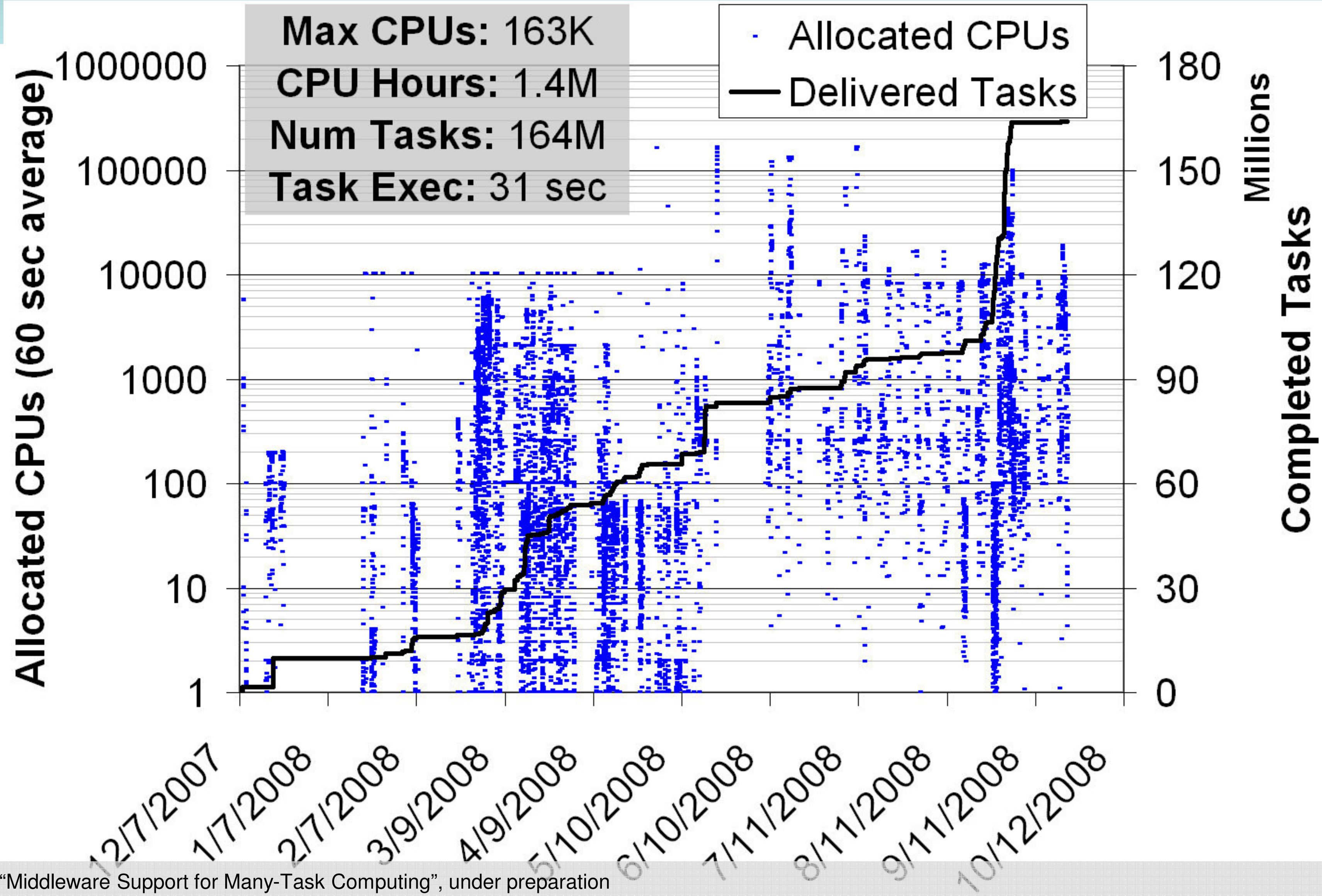
- There is more to HPC than tightly coupled MPI, and more to HTC than embarrassingly parallel long jobs
  - MTC: Many-Task Computing
  - Addressed real challenges in resource management in large scale distributed systems to enable MTC
  - Covered many domains (via Swift and Falkon): astronomy, medicine,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

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

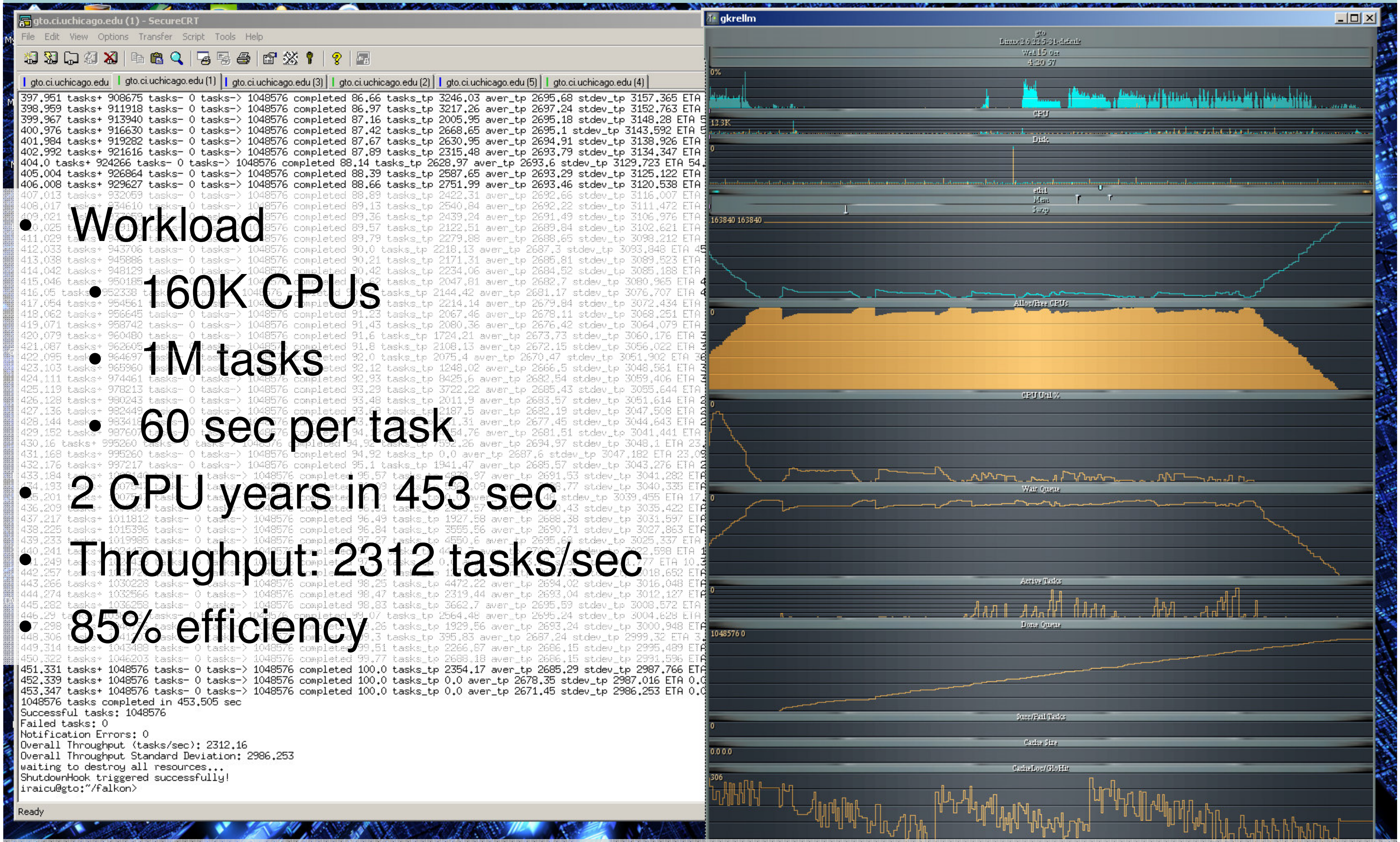
[Globus07] "Falkon: A Proposal for Project Globus Incubation"

Falkon Activity History (10 months)



[JS09] "Middleware Support for Many-Task Computing", under preparation

Falkon Monitoring



- Workload
  - 160K CPUs
  - 1M tasks
  - 60 sec per task
- 2 CPU years in 453 sec
- Throughput: 2312 tasks/sec
- 85% efficiency

[JS09] "Middleware Support for Many-Task Computing", under preparation

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

Publications Central to Dissertation

(2005 – 2009)

1. Ioan Raicu, Ian Foster, Yong Zhao, Alex Szalay, Philip Little, Christopher M. Moretti, Amitabh Chaudhary, Douglas Thain. “Towards Data Intensive Many-Task Computing”, under review as a book chapter in “Data Intensive Distributed Computing: Challenges and Solutions for Large-Scale Information Management”, IGI Global Publishers
2. Ioan Raicu, Ian Foster, Yong Zhao, Philip Little, Christopher Moretti, Amitabh Chaudhary, Douglas Thain. “The Quest for Scalable Support of Data Intensive Workloads in Distributed Systems”, under review at ACM HPDC09
3. Ioan Raicu, Ian Foster, Yong Zhao. “Many-Task Computing for Grids and Supercomputers”, Invited Paper, IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS08), 2008
4. Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu. “Cloud Computing and Grid Computing 360-Degree Compared”, IEEE Grid Computing Environments (GCE08) 2008.
5. 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”, IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS08) 2008.
6. Ioan Raicu, Zhao Zhang, Mike Wilde, Ian Foster, Pete Beckman, Kamil Iskra, Ben Clifford. “Towards Loosely-Coupled Programming on Petascale Systems”, IEEE/ACM Supercomputing 2008.
7. 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.
8. 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.
9. 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.
10. 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.
11. Ioan Raicu, Yong Zhao, Ian Foster, Alex Szalay. "Accelerating Large-scale Data Exploration through Data Diffusion", Int. Workshop on Data-Aware Distributed Computing 2008.
12. 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.
13. 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.
14. Yong Zhao, Ioan Raicu, Ian Foster, Mihael Hategan, Veronika Nefedova, Mike Wilde. “Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments”, Book chapter in Grid Computing Research Progress, ISBN: 978-1-60456-404-4, Nova Publisher 2008.
15. Ioan Raicu. “Harnessing Grid Resources with Data-Centric Task Farms”, University of Chicago, Computer Science Department, PhD Proposal, December 2007, Chicago, Illinois.
16. 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.
17. 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.
18. Ioan Raicu, Yong Zhao, Ian Foster, Alex Szalay. “A Data Diffusion Approach to Large Scale Scientific Exploration”, Microsoft Research eScience Workshop 2007.
19. Ioan Raicu, Yong Zhao, Catalin Dumitrescu, Ian Foster, Mike Wilde. “Falkon: a Fast and Light-weight tasK executiON framework”, IEEE/ACM SuperComputing 2007.
20. Ioan Raicu, Catalin Dumitrescu, Ian Foster. “Dynamic Resource Provisioning in Grid Environments”, TeraGrid Conference 2007.
21. 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.
22. 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.
23. Ioan Raicu, Ian Foster, Alex Szalay. “Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets”, poster, IEEE/ACM SuperComputing 2006.
24. Ioan Raicu, Ian Foster, Alex Szalay, Gabriela Turcu. “AstroPortal: A Science Gateway for Large-scale Astronomy Data Analysis”, TeraGrid Conference 2006, June 2006.
25. 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 (disjoint set)
(2002 – 2007)

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.
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
7. Catalin Dumitrescu, Ioan Raicu, Ian Foster. "DI-GRUBER: A Distributed Approach for Grid Resource Brokering", IEEE/ACM Super Computing 2005 (SC 2005)
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
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), 2004.
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
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, 2003.
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, 2003
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, 2003.
17. Ioan Raicu. "Efficient Even Distribution of Power Consumption in Wireless Sensor Networks", ISCA 18th International Conference on Computers and Their Applications, CATA 2003.
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.
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, 2002.