Energy Prediction for I/O Intensive Workflow Applications

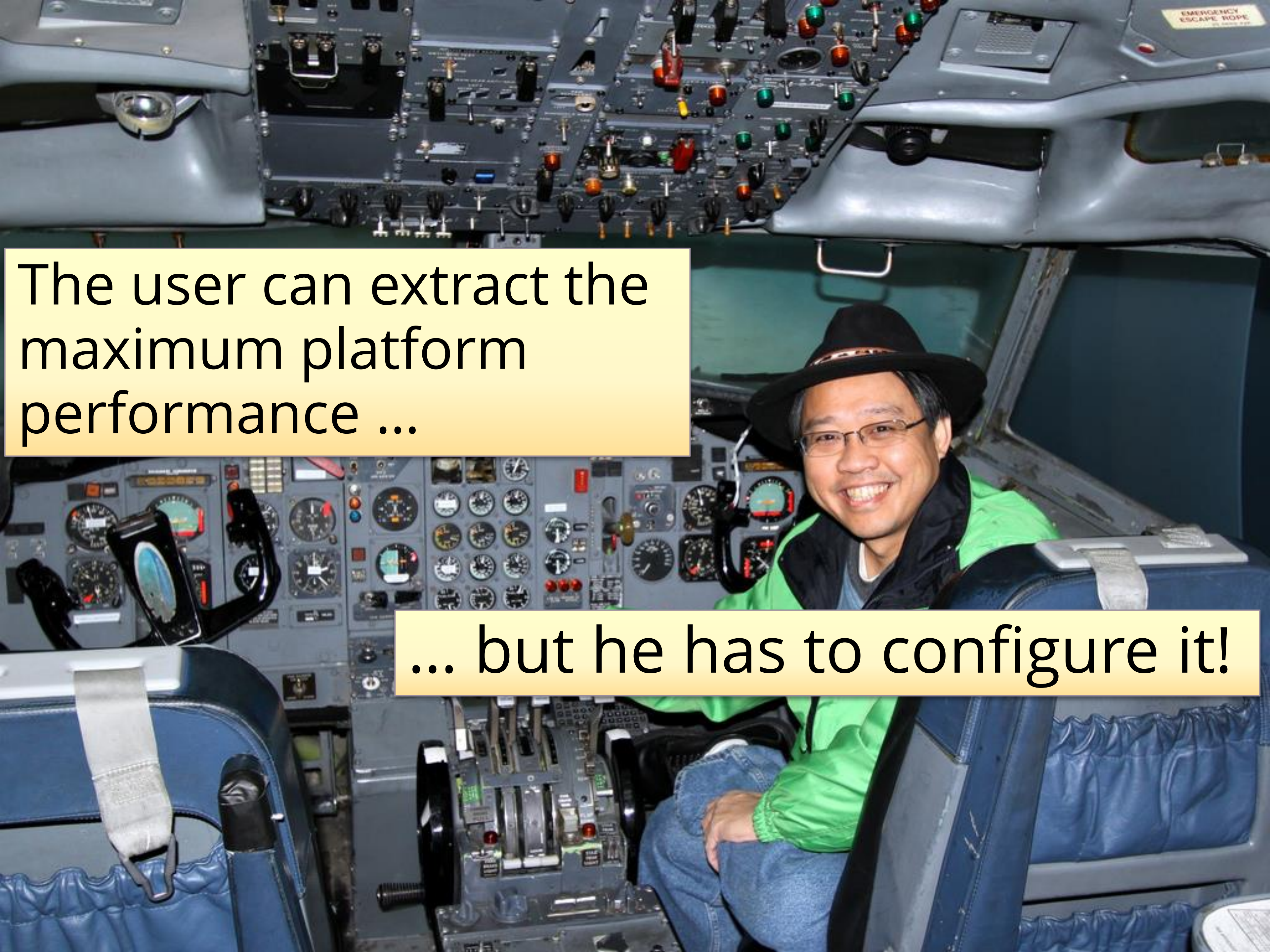
Hao Yang, Lauro Beltrão Costa, Matei Ripeanu

NetSysLab
Electrical and Computer Engineering Department
The University of British Columbia

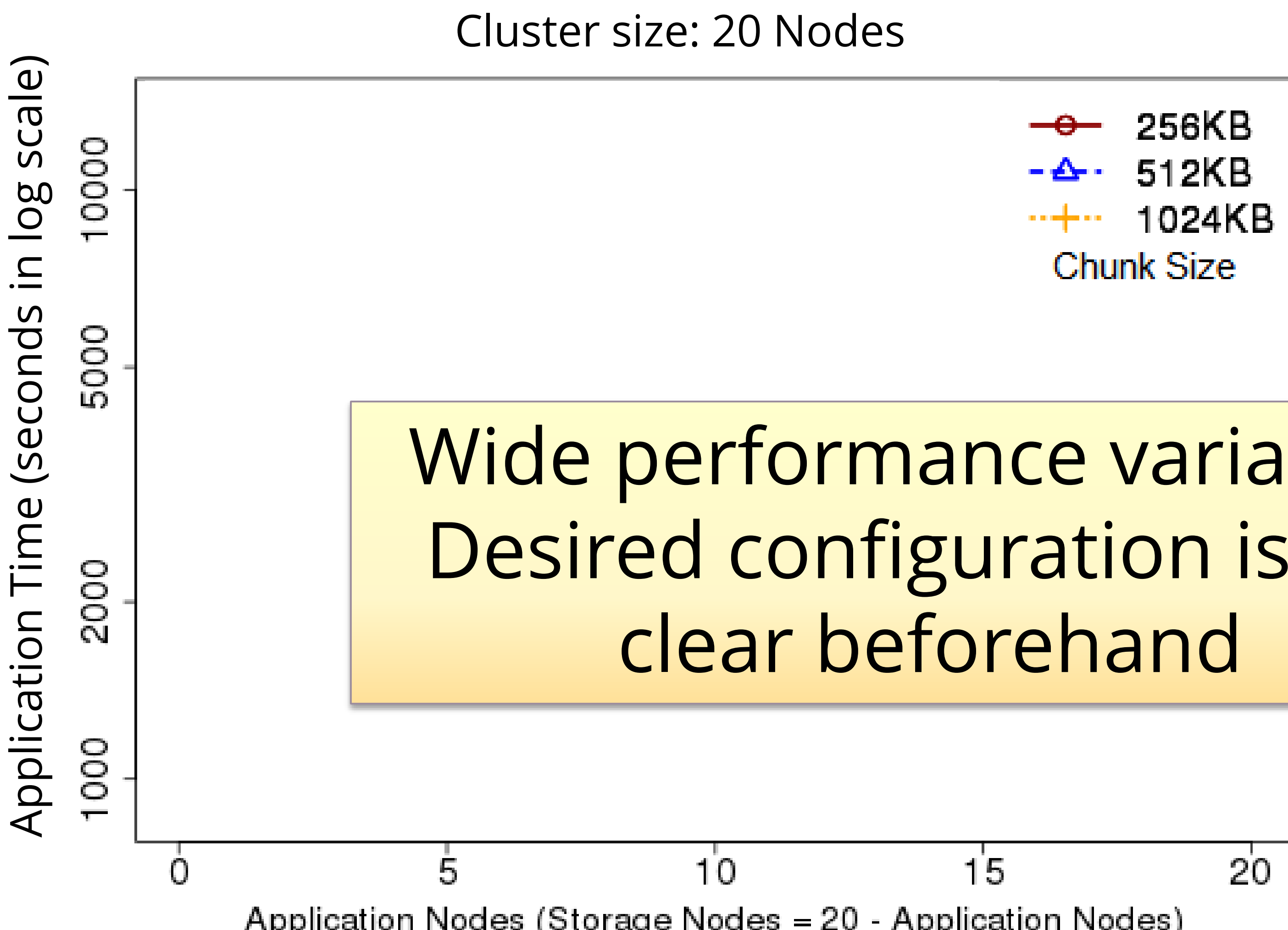
NetSysLab

Electrical and Computer Engineering

UBC


The user can extract the maximum platform performance …
… but he has to configure it!

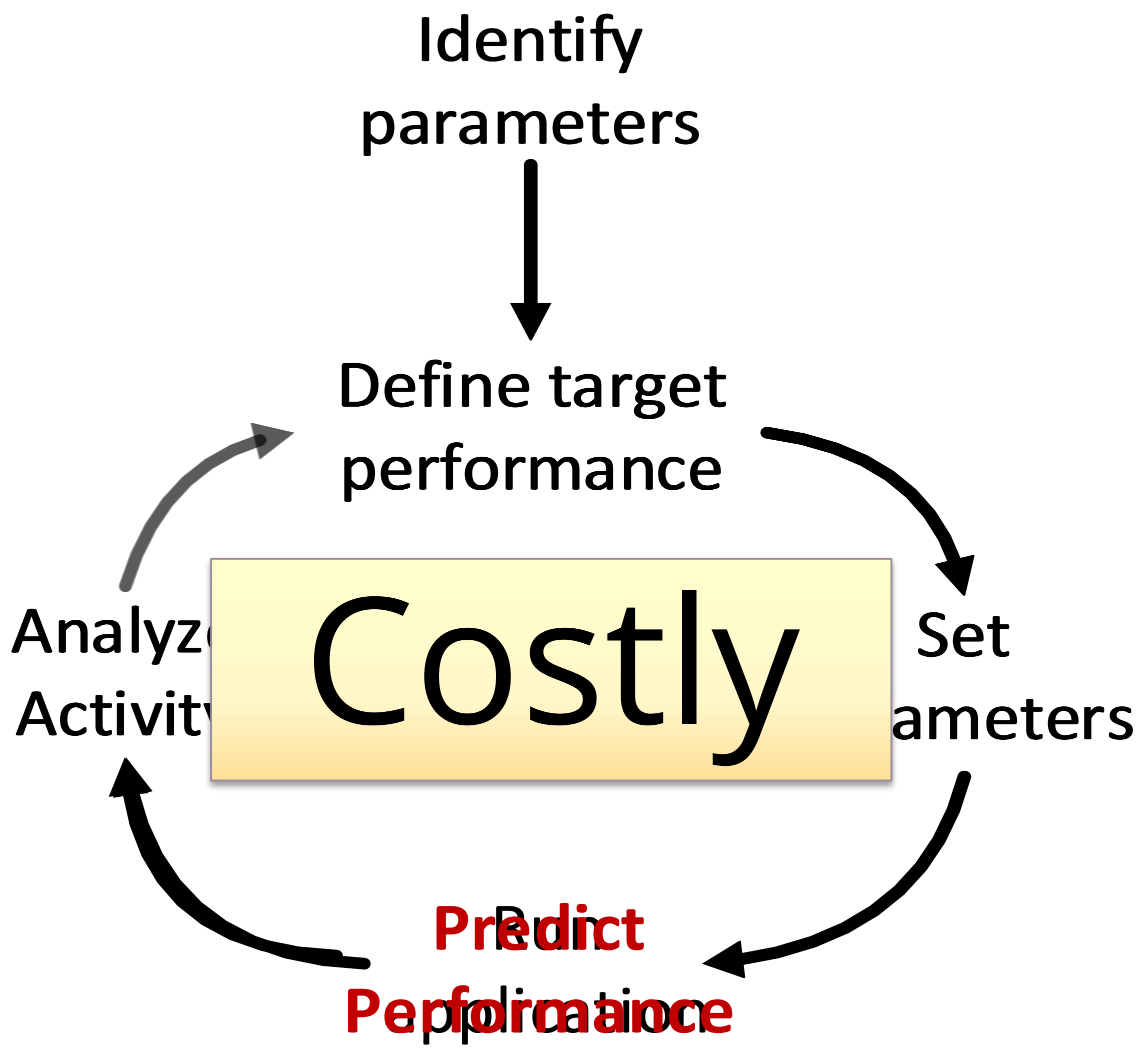
An Example



Wide performance variation. Desired configuration is not clear beforehand

← More Storage Nodes More Application Nodes →

Configuration Loop



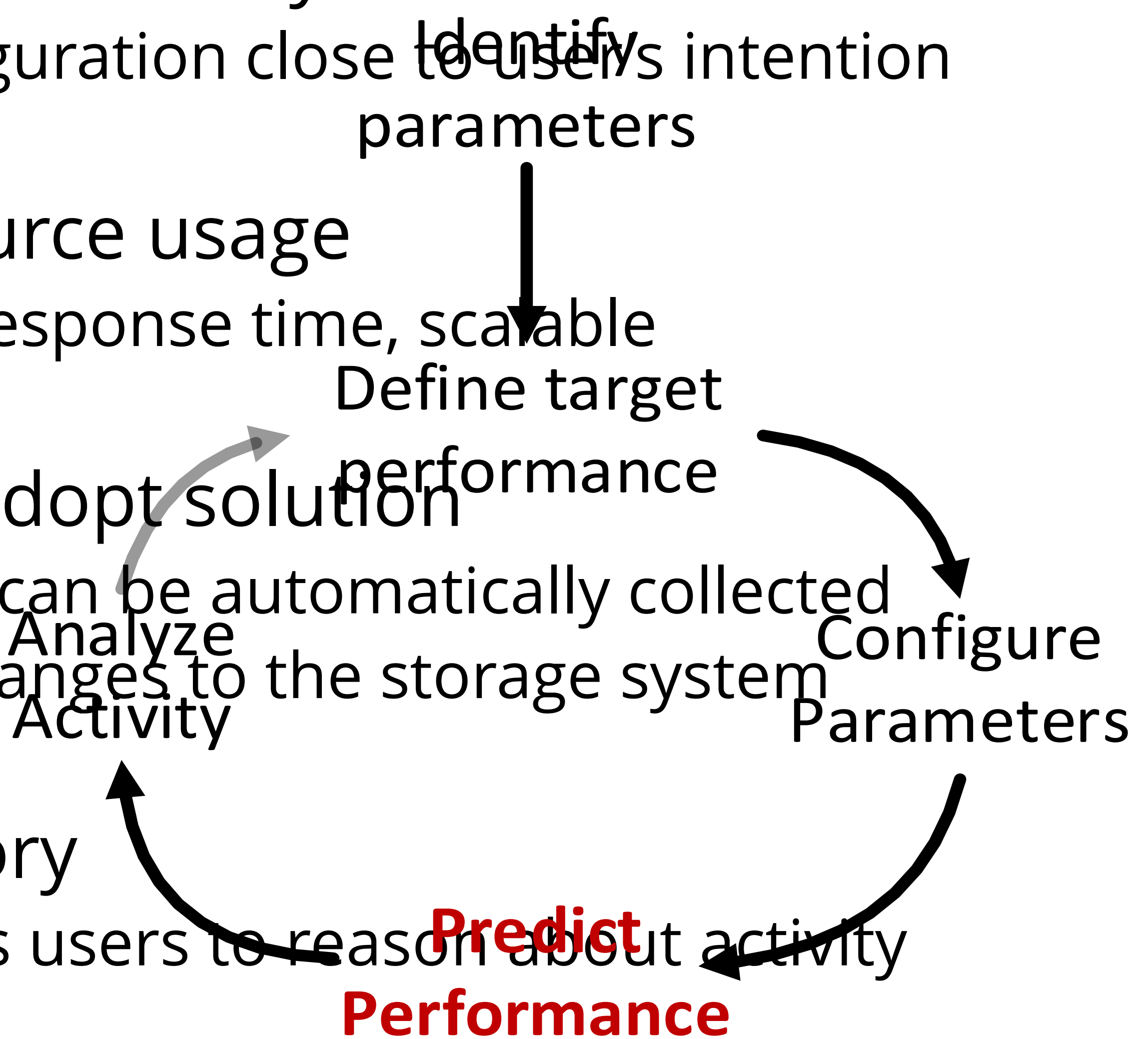
Requirements

Adequate Accuracy
- Configuration close to user's intention

Low resource usage
- Fast response time, scalable

'Easy' to adopt solution
- Input can be automatically collected
- No changes to the storage system

Explanatory
- Allows users to reason about activity



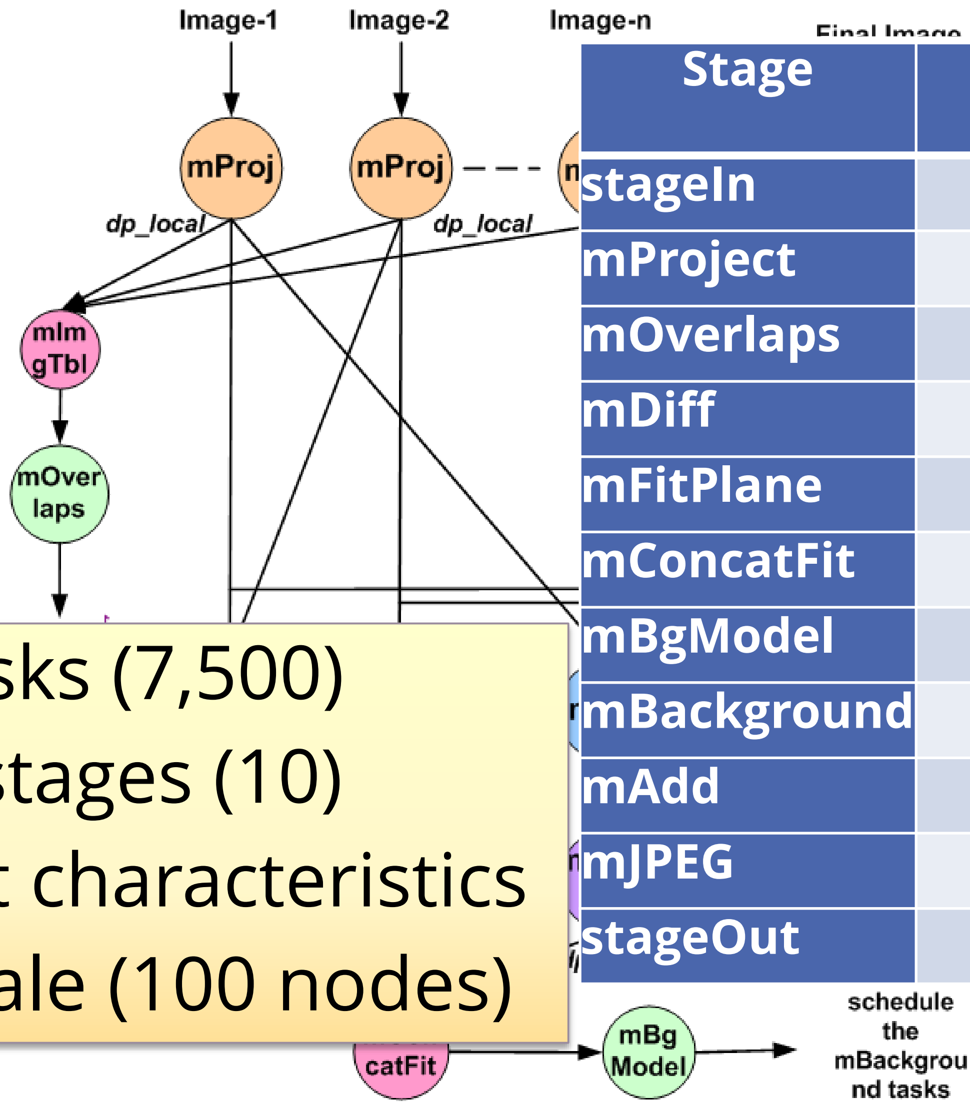
Our goal:

support for storage configuration/provisioning decisions

Success metrics:
[time] Application turnaround time, Total CPU time
[energy] Energy, Energy-delay product

Background: The Workload

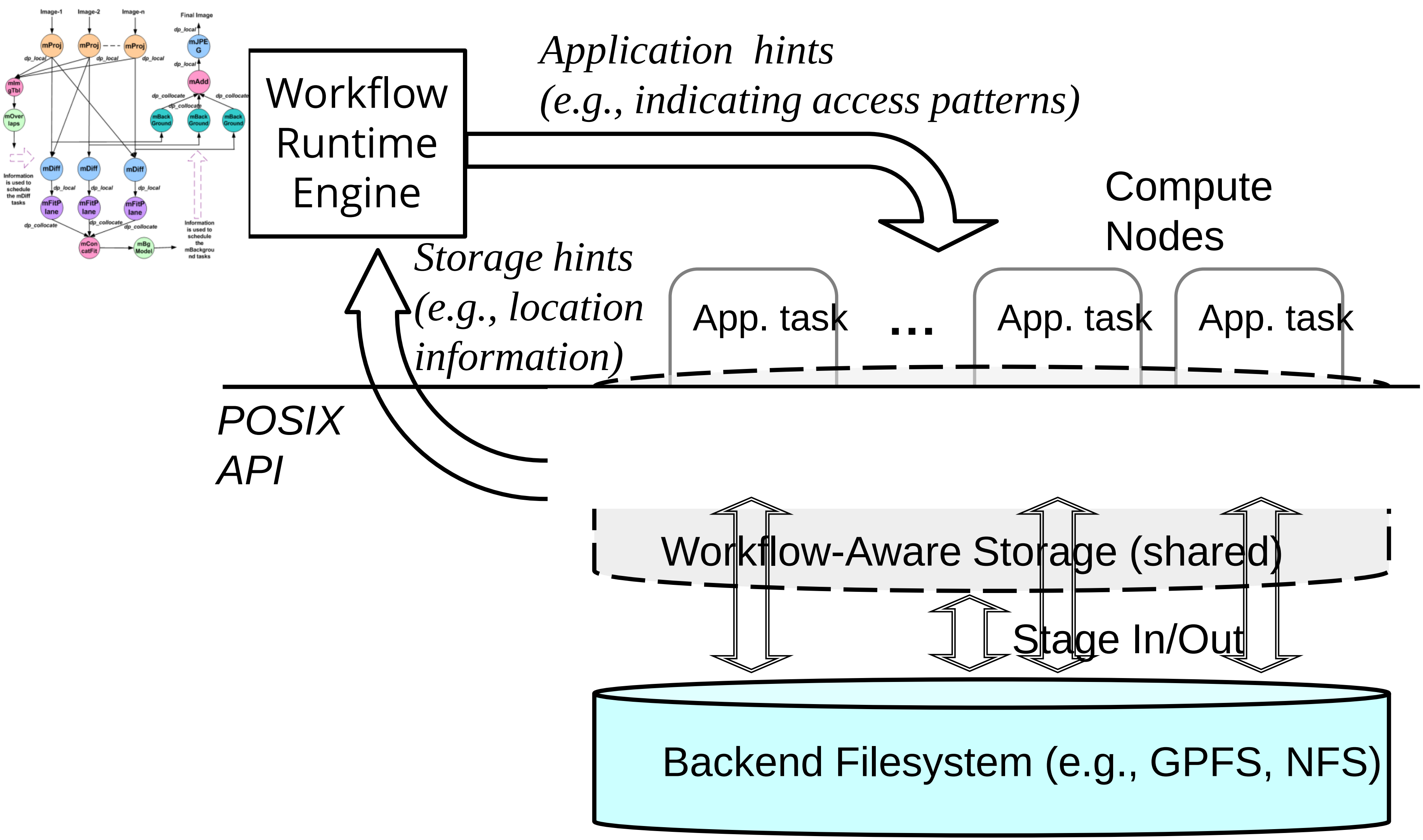
ManyTask Applications



Stage	Total Data	#files	File size
stageIn	1.9 GB	957	1.7 - 2.1 MB
mProject	8 GB	1910	3.3 - 4.2 MB
mOverlaps	336 KB	1	336 KB
mDiff	2.6 GB	564	0.1 - 3 MB
mFitPlane	5MB	1420	4 KB
mConcatFit	150 KB	1	150 KB
mBgModel	20 KB	1	20 KB
mBackground	8 GB	1913	3.3 - 4.2 MB
mAdd	5.9 GB	2	165MB-3GB
mJPEG	46 MB	1	46 MB
stageOut	3.1 GB	3	46MB-3GB

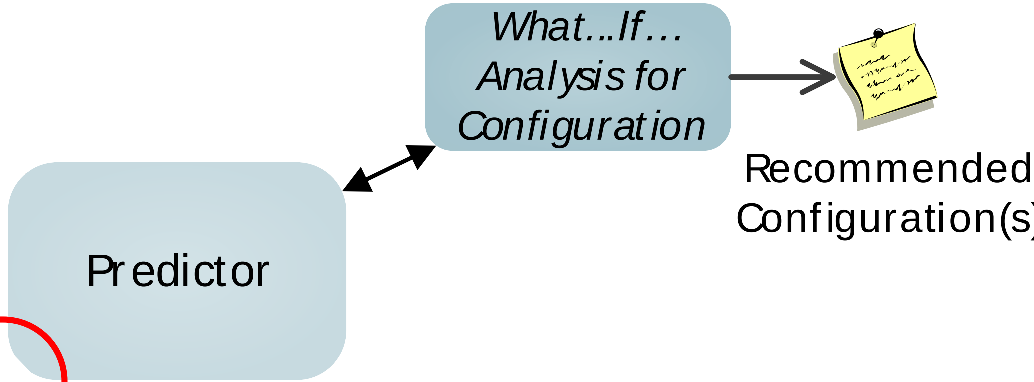
Many tasks (7,500)
Several stages (10)
Different characteristics
Large scale (100 nodes)

Background: The runtime platform



L Costa, H. Yang, E. Vairavanathan, A. Barros, K. Maheshwari, G. Fedak, D.S. Katz, M. Wilde, M. Ripeanu , S. Al-Kiswany, *The Case for Workflow-Aware Storage: An Opportunity Study using MosaStore*, Journal of Grid Computing 2014.

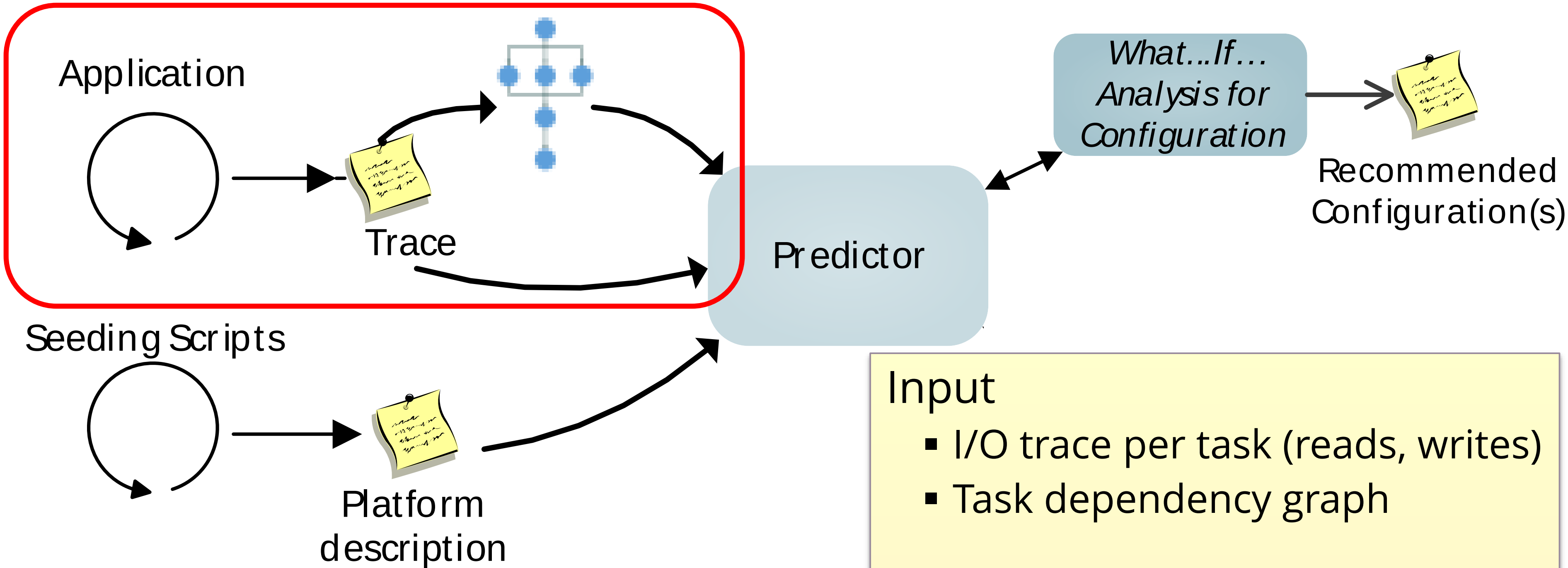
Solution Overview



Model Seeding

- Identify performance characteristics of the platform (a.k.a. system identification)

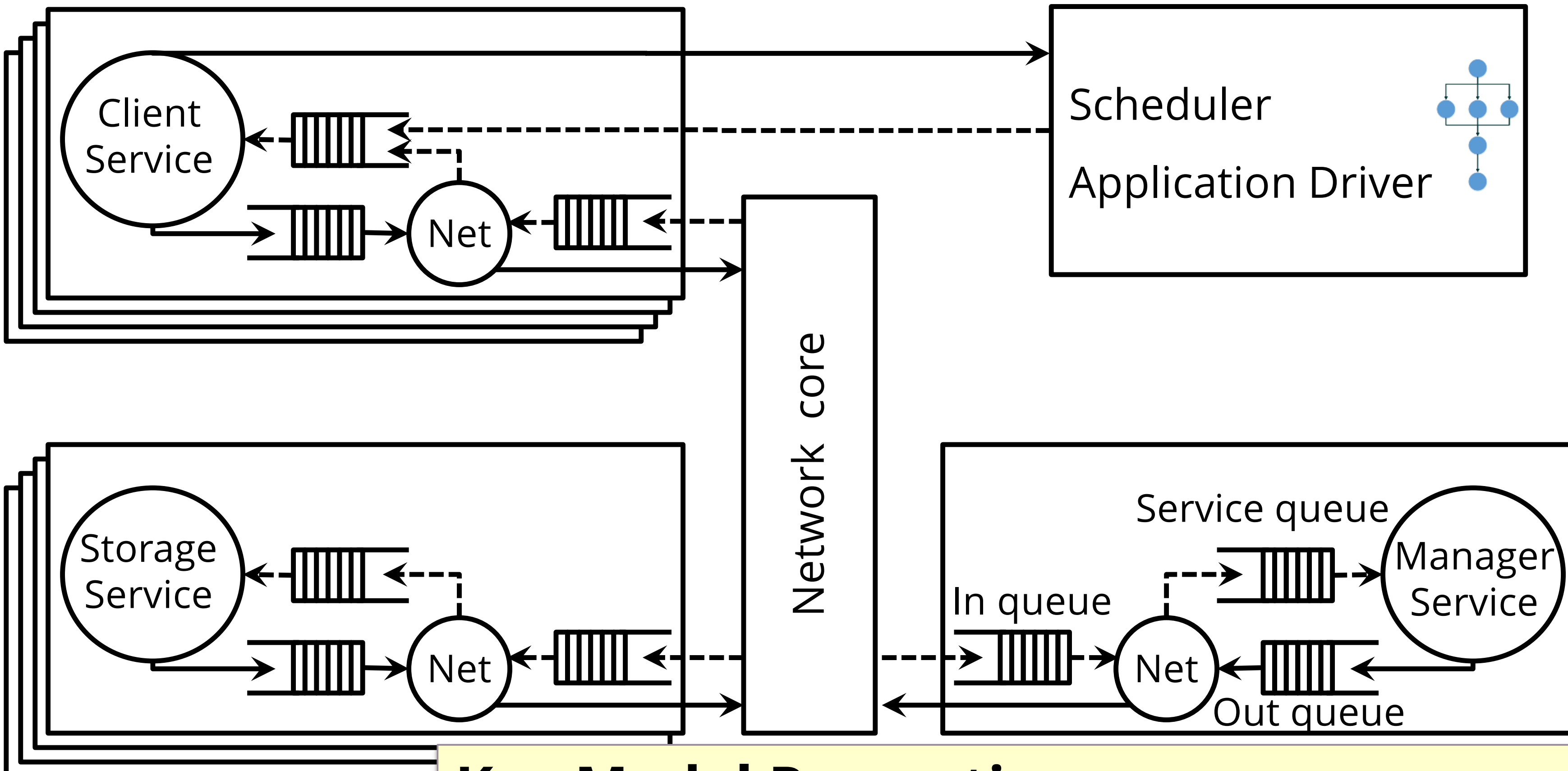
Workload Description



Input
- I/O trace per task (reads, writes)
- Task dependency graph

Preprocessing
- Aggregates I/O operations
- Infers computing time
- Infers scheduling overhead

Storage System Model



Key Model Properties:

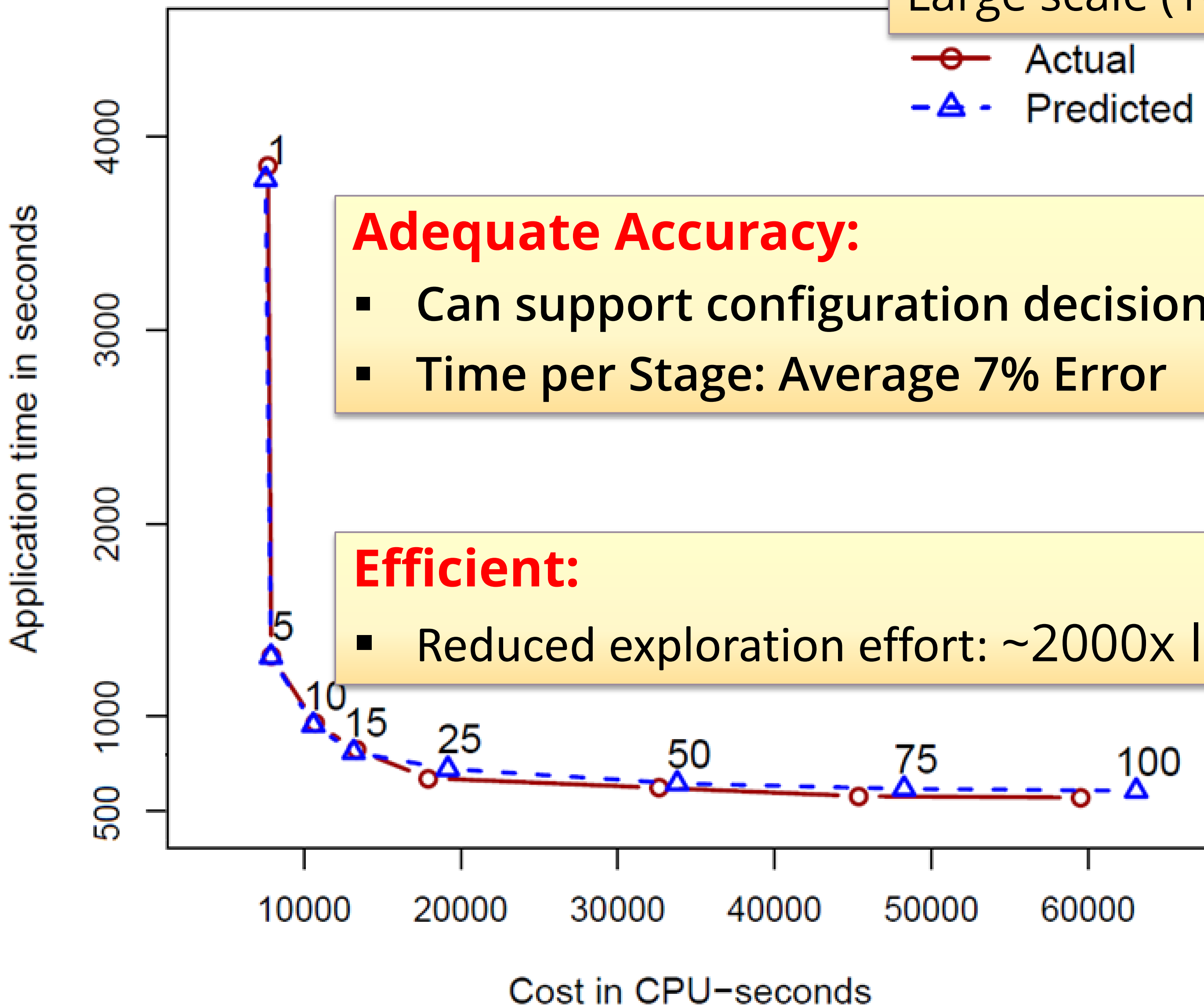
- Generic: all object-based storage architectures:
- Uniform: all system services modelled similarly
- Coarse: thus more scalable

How well does this work?

Predicting application turnaround time and total CPU cost for a **complex application** at **large scale**

Time vs. Allocation Cost

Montage Workload
Many tasks (7,500)
Several stages (10)
Different characteristics
Large scale (100 nodes)



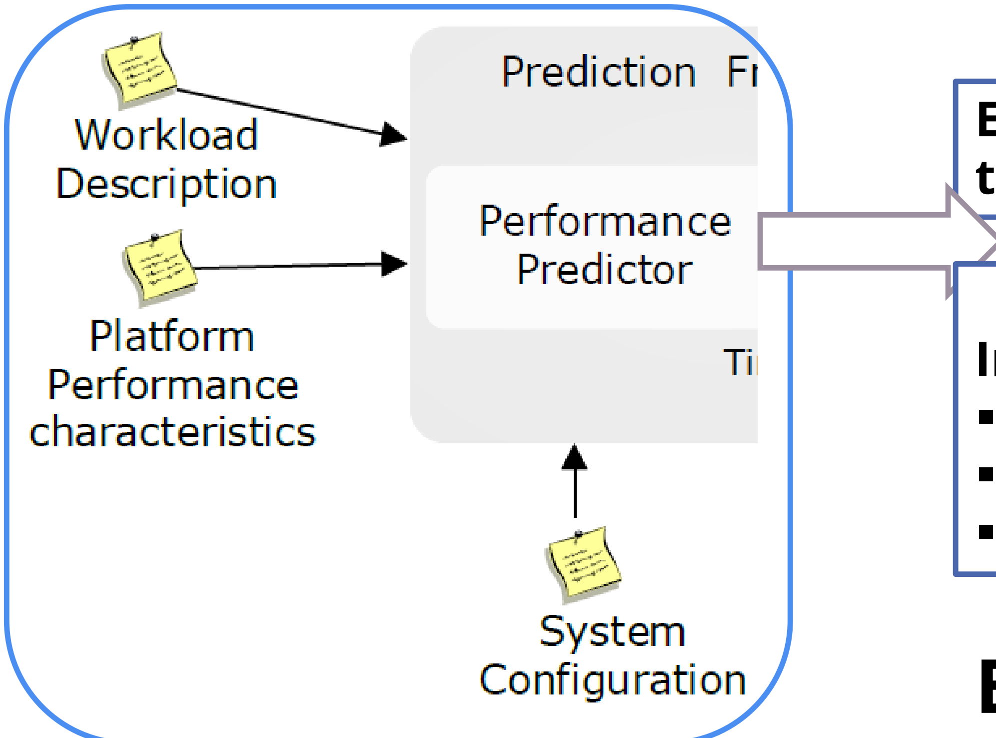
Adequate Accuracy:

- Can support configuration decisions
- Time per Stage: Average 7% Error

Efficient:

- Reduced exploration effort: ~2000x less

Taking advantage of detailed predictions

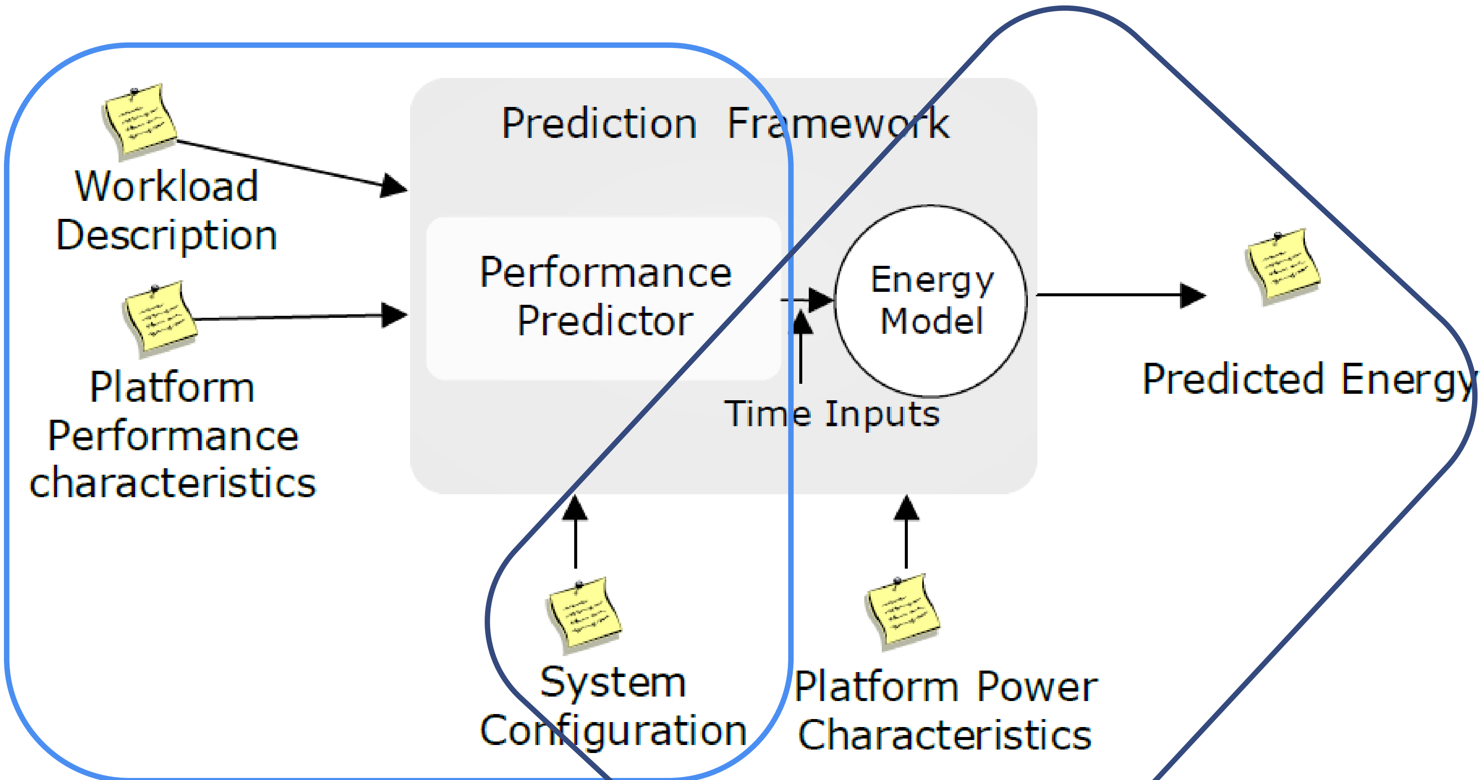


Estimates: Application turnaround runtime

Internally: time per-operation
- **Compute**
- **Network I/O**
- **Storage**

Estimating energy is possible with power information for these states

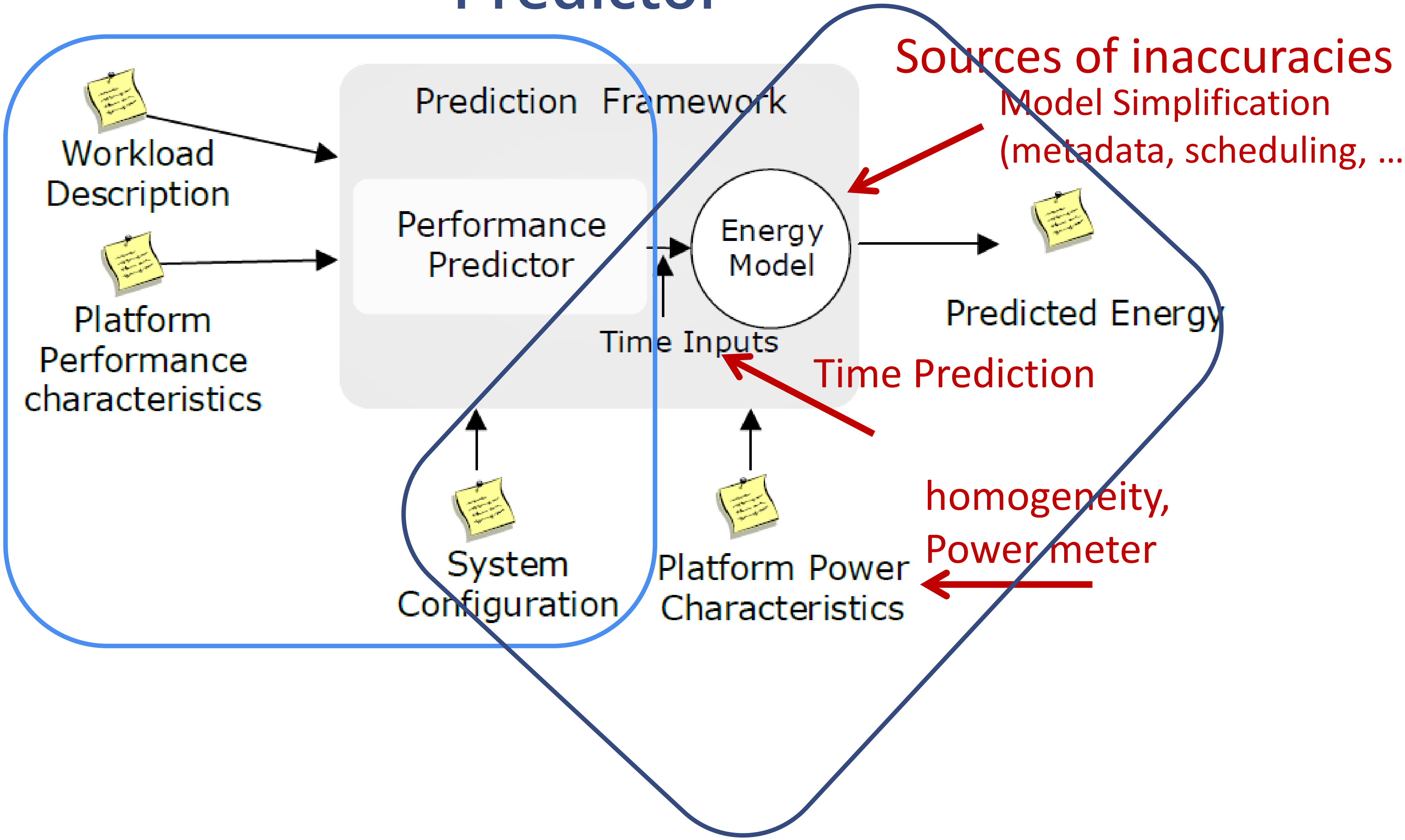
- *Supporting Storage Configuration for I/O Intensive Workflows*, L. Costa, S. Al-Kiswany, H. Yang, M. Ripeanu, ICS'14
- *Predicting Intermediate Storage Performance for Workflow Applications*, .L Costa, S. Al-Kiswany, A. Barros, H. Yang, M. Ripeanu, PDSW'13,



Energy Model

Execution States:	Energy	Power Profile * Predicted Times
Idle →	E^{idle} →	$P_i^{idle} * T^{total}$
Network Transfer →	ΔE^{net} →	$(P^{net} - P^{idle}) * T_i^{net}$
I/O ops (read, write) →	$\Delta E^{storage}$ →	$(P^{storage} - P^{idle}) * T_i^{storage}$
Task Processing →	ΔE^{app} →	$(P^{App} - P^{idle}) * T^{App}$

Methodology – Building Energy Consumption Predictor



Evaluation - Platform

Grid5000 Lyon site

Idle	P_i^{idle}	91.6W
App	$P_i^{App} - P_i^{idle}$	33.6W
Storage I/O	$P_i^{storage} - P_i^{idle}$	37.4W
Net transfer	$P_i^{net} - P_i^{idle}$	36.1W

- Taurus Cluster (11 nodes)

  two 2.3GHz Intel Xeon E5-2630 CPUs (each with 32GB memory, 10 Gbps NIC

- Sagittaire Cluster (16 nodes)

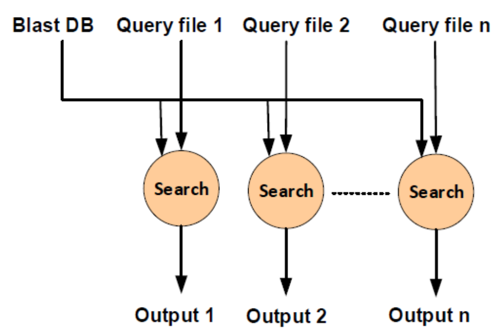
  two 2.4GHz AMD Opteron CPUs (each with one core), 2GB RAM and 1 Gbps NIC

- SME Omegawatt power-meter per Node

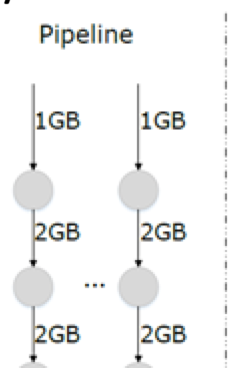
  0.01W power resolution at 1Hz sampling rate

Evaluation sample: What is the energy and performance impact of CPU throttling? Is it application-specific?

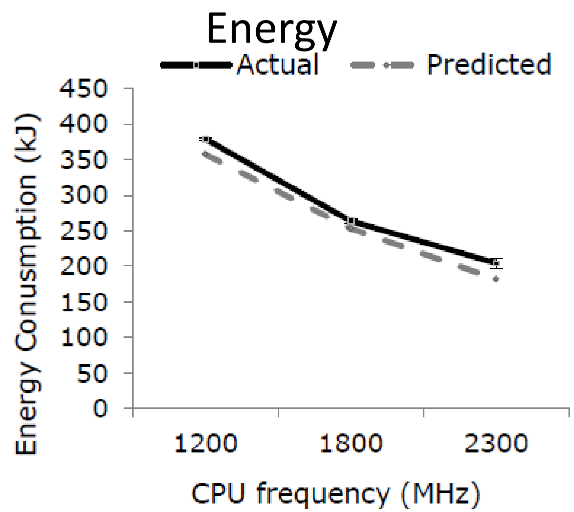
BLAST: CPU Intensive



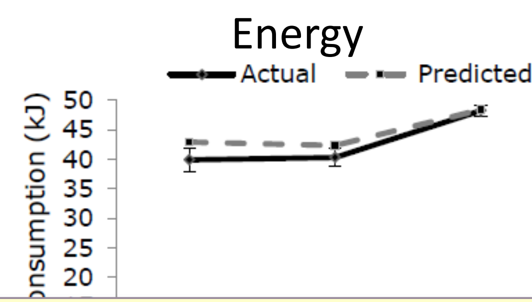
Pipeline: I/O Intensive



Frequency Levels: 1200MHz, 1800MHz, 2300MHz



Throtling a bad idea



Throtling a good idea

Energy predictions accurate enough to support configuration decisions

Summary

Intermediate Storage System

- Configuration and provisioning for one application
- Our prototype: MosaStore

Minimalist Model + Simple seeding

- Leverages applications' characteristics
- Easy to use, Low-runtime

Accuracy adequate to support correct configuration and provisioning decisions

Code & papers at: NetSysLab.ece.ubc.ca

Contributions

Predicting Energy Consumption
MTAGS '14; ERSS '11

Development Support Use-Case
SEHPC '14

Predictor Prototype
code

Performance Prediction Mechanisms: Models and Seeding Procedures
TPDS 'Sub; ICS '14; PDSW '13; Grid '10

Opportunity Study on Storage Techniques
JoGC '14; CCGrid '12

Storage System Design
FAST 'Sub

Storage System Prototype (MosaStore)
code

Collaborations and Publications

1. **Support for Provisioning and Configuration of Intermediate Storage Systems.** L. B. Costa, S. Al-Kiswany, H. Yang and M. Ripeanu. IEEE TPDS Submission. Oct. 2014.
2. **Energy Prediction for I/O Intensive Workflows.** H. Yang, L. B. Costa, and M. Ripeanu. MTAGS '14. SC Workshop. ACM. Sep. 2014.
3. **Experience with Using a Performance Predictor During Development: a Distributed Storage System Tale**. L. B. Costa, J. Brunet, and L. Hattori. SEHPC '14. SC Workshop. ACM. To Appear. Nov. 2014.
4. **Supporting Storage Configuration for I/O Intensive Workflows.** L. B. Costa, S. Al-Kiswany, H. Yang and M. Ripeanu, 28th ACM ICS. Jun. 2014
5. **Predicting Intermediate Storage Performance for Workflow Applications**. L. B. Costa, S. Al-Kiswany, A. Barros, H. Yang, M. Ripeanu, 8th PDSW '13 (SC Workshop). ACM. Nov. 2013
6. **Assessing Data Deduplication Trade-offs from an Energy Perspective.** L. B. Costa, S. Al-Kiswany, R. V. Lopes and M. Ripeanu. ERSS (Green Computing Workshop). IEEE. Jul. 2011
7. **Towards Automating the Configuration of a Distributed Storage System.** L. B. Costa and M. Ripeanu. 11th ACM/IEEE 2010. Oct. 2010

9. **The Case for Cross-Layer Optimizations in Storage: A Workflow-Optimized Storage System.** S. Al-Kiswany, L. B. Costa, H. Yang, E. Vairavanathan and M. Ripeanu. Journal Submission. In preparation.
10. **A Software Defined Storage for Scientific Workflow Applications.** S. Al-Kiswany, L. B. Costa, H. Yang, E. Vairavanathan and M. Ripeanu. FAST 'Submission. Submitted in October 2014.
11. **The Case for Workflow-Aware Storage: An Opportunity Study**. L. B. Costa, H. Yang, E. Vairavanathan, A. Barros, K. Maheshwari, G. Fedak, D. Katz, M. Wilde, M. Ripeanu and S. Al-Kiswany. Journal of Grid Computing. Accepted in Jun. 2014
12. **A Workflow-Aware Storage System: An Opportunity Study.** E. Vairavanathan, S. Al-Kiswany, L. B. Costa, Z. Zhang, D. Katz, M. Wilde and M. Ripeanu 12th IEEE/ACM CCGrid'12. May 2012

Prediction

13. **Efficient Large-Scale Graph Processing on Hybrid CPU and GPU Systems**. A. Gharaibeh, E. Santos-Neto, L. B. Costa and M. Ripeanu. ACM Transactions on Parallel Computing. Under Review. January 2014
14. **The Energy Case for Graph Processing on Hybrid CPU and GPU Systems.** A. Gharaibeh, E. Santos-Neto, L. B. Costa and M. Ripeanu. IA^3 (SC Workshop). ACM. Nov. 2013
15. **On Graphs, GPUs, and Blind Dating: A Workload to Processor Matchmaking Quest.** A. Gharaibeh, L. B. Costa, E. Santos-Neto and M. Ripeanu. 27th IEEE IPDPS. May 2013
16. **A Yoke of Oxen and a Thousand Chickens for Heavy Lifting Graph Processing.** A. Gharaibeh, L. B. Costa, E. Santos-Neto and M. Ripeanu. IEEE/ACM 21st PACT. Sep. 2012
17. **GPU Support for batch oriented workloads.** L. B. Costa, S. Al-Kiswany and M. Ripeanu. 28th IPCCC. IEEE. Dec. 2009
18. **Nodewiz: Fault-tolerant grid information service**. S. Basu, L. B. Costa, F. V. Brasileiro, S. Banerjee, P. Sharma, and S-J Lee. Journal of Peer-to-Peer Networking and Applications, 2(4):348--366. Springer, Dec. 2009.

Contributions

Predicting Energy Consumption
MTAGS '14; ERSS '11

Development Support Use-Case
SEHPC '14

Predictor Prototype
code

Performance Prediction Mechanisms: Models and Seeding Procedures
TPDS 'Sub; ICS '14; PDSW '13; Grid '10

Opportunity Study on Storage Techniques
JoGC '14; CCGrid '12

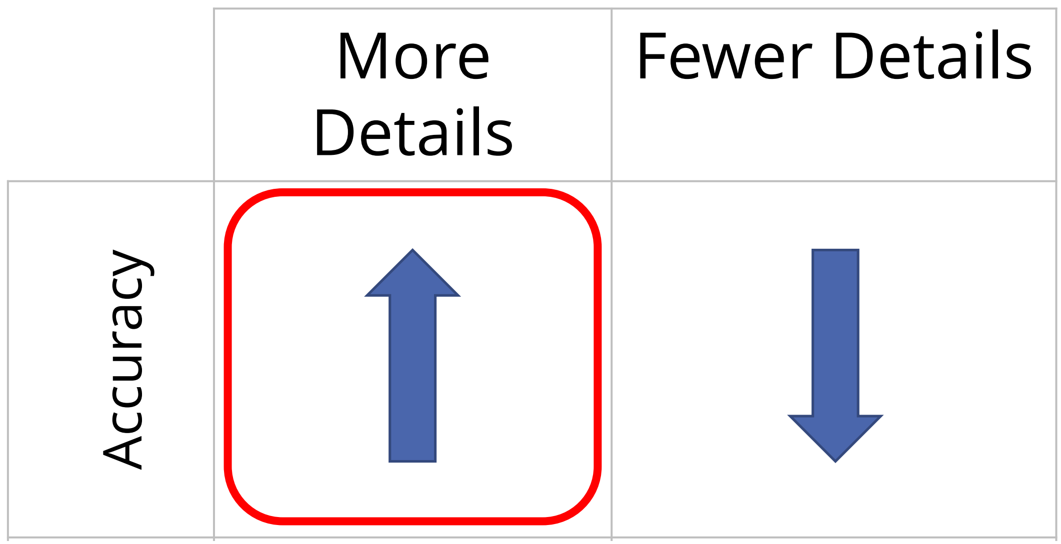
Storage System Design
FAST 'Sub

Storage System Prototype (MosaStore)
code

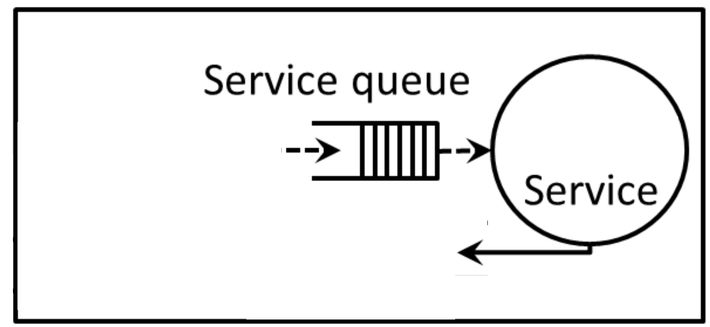
Backup Slides

- Synthetic Benchmarks
- Real Applications
- Other Scenarios
- Scalability
- Energy Prediction
- Limitations
- Related Work
- Future Work
- Supporting Development
- Data Deduplication
- Data Deduplication Energy
- Methodology: Development
- More on MosaStore

Modeling: Life is a trade-off



Storage System Model



Properties:

- General
- Uniform
- Coarse

Service times per chunk needed

- Read/Write for Client and Storage
- Open for Manager
- Local/Remote for Network

Model Parameters

System Deployment	
Number of Storage Nodes	N^{sm}
Number of Client Nodes	N^{cli}
Collocation of Storage and Client Modules	Colloc
Performance	
Manager Service Time	μ^{ma}
Storage Module Read Service Time	μ^{smRead}
Storage Module Write Service Time	$\mu^{smWrite}$
Client Service Time	μ^{cli}
Remote Network Service Time	μ^{remNet}
Local Network Service Time	μ^{locNet}

Workload Description

Input

- I/O trace per task (reads, writes)
- Task dependency graph

Preprocessing

- Aggregates I/O operations
- Infers computing time
- Infers scheduling overhead

Evaluation

Success Metrics

- Accuracy (time, cost)
- Time to predict

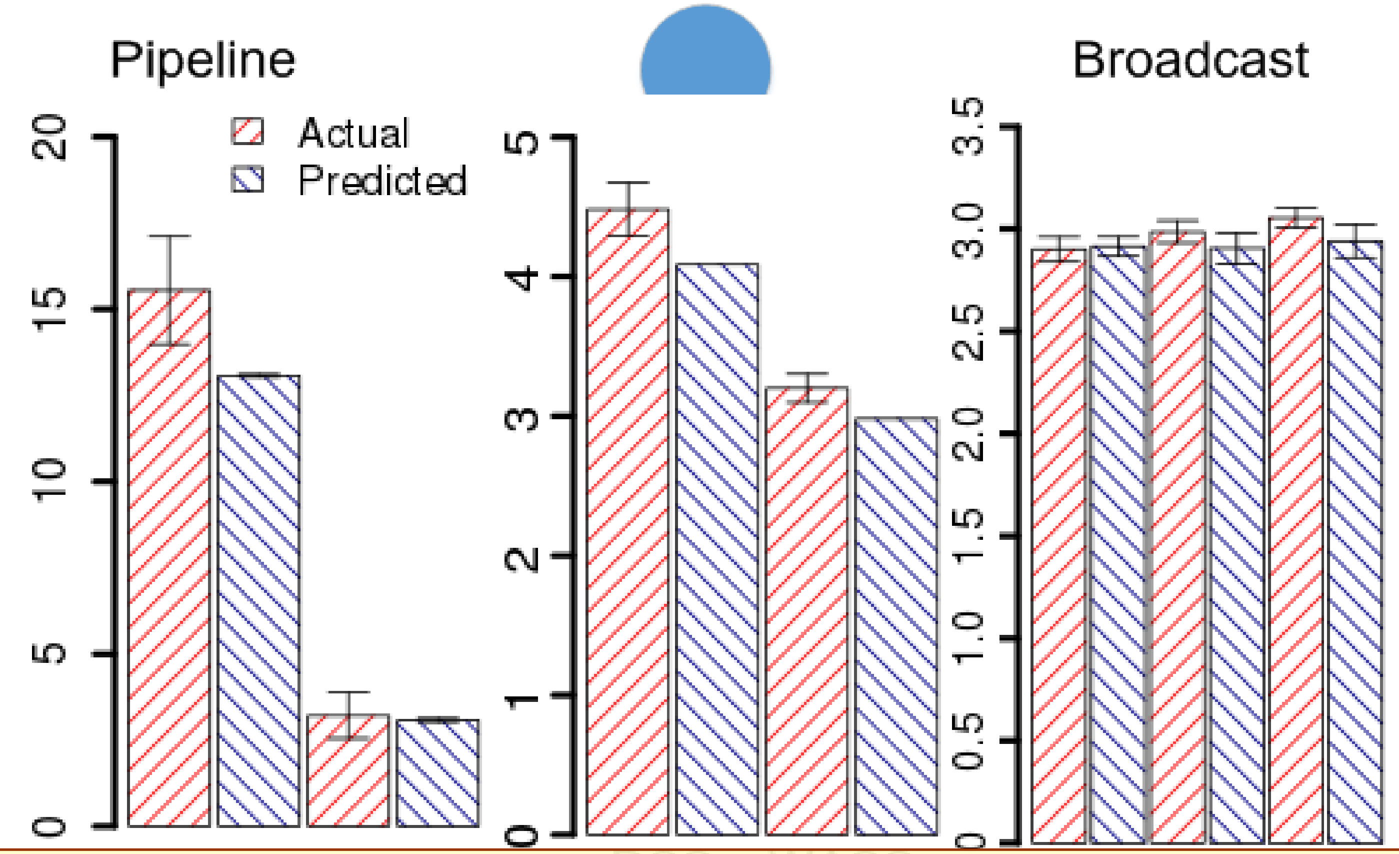
Workloads

- Synthetic benchmarks
- Real applications

Testbed

- NetSysLab - 20 nodes
- Grid 5K - 101 nodes

Synthetic Benchmarks



- Predicted time error: Underprediction, Average ~8%,

Common patterns in the structure of workflows

I/O only to stress the storage system

What about a real application?

Simple Application

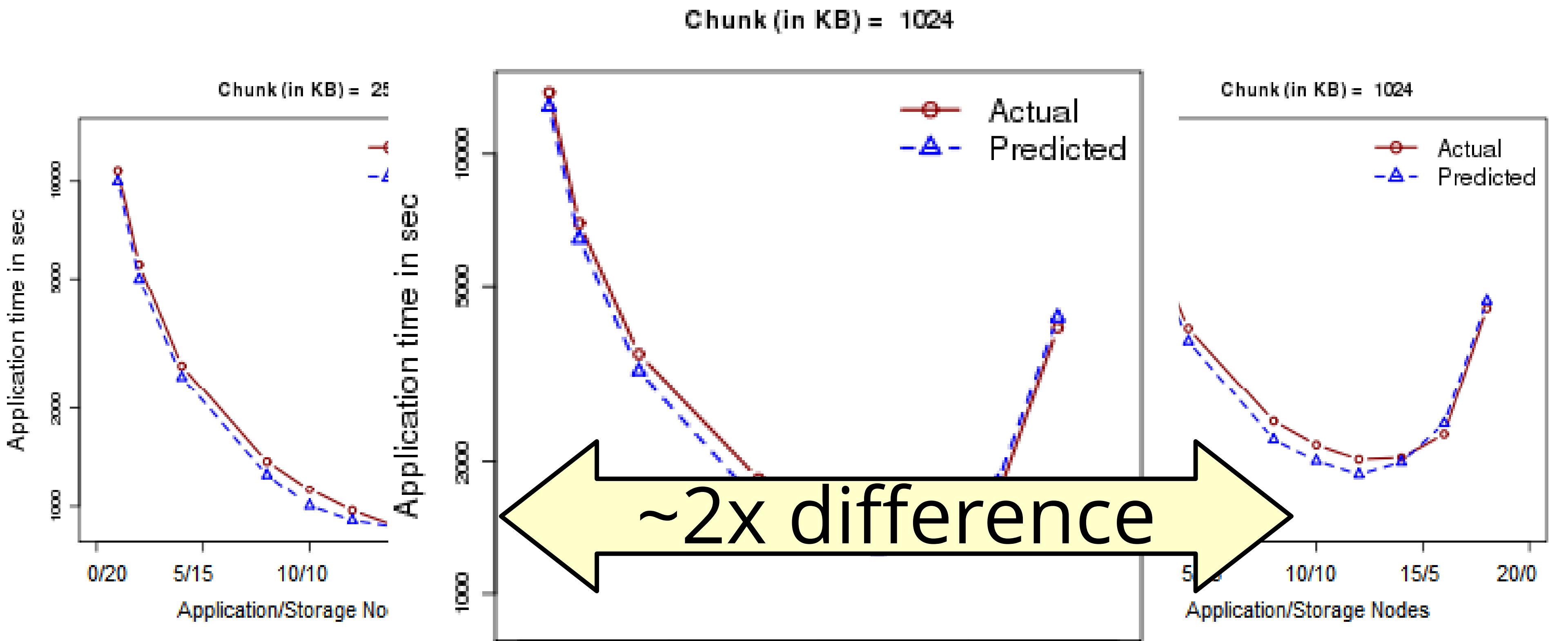
BLAST

- 200 queries (tasks) over a DNA database file, then reduce

Impact of different parameters

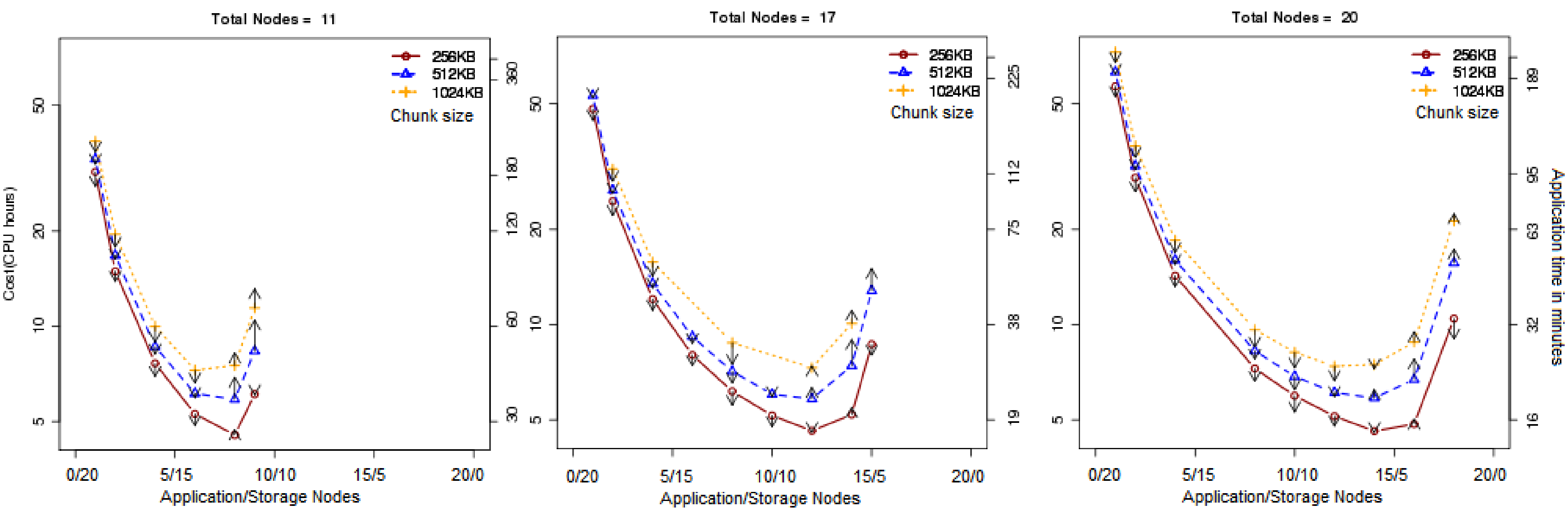
- # of storage nodes, # of clients
- chunk size

BLAST Performance



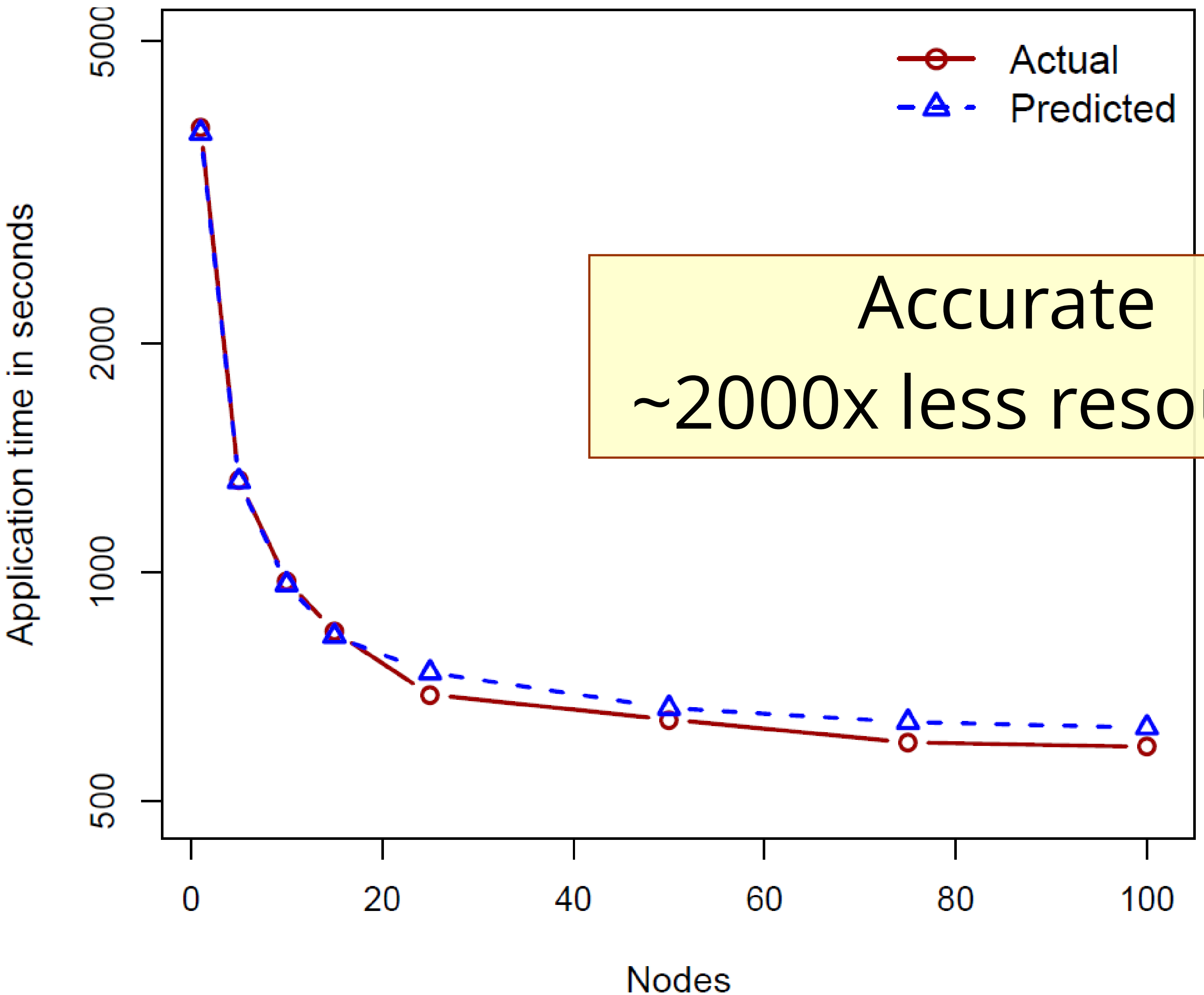
Prediction provides adequate accuracy
~3000x less resources
Similar accuracy with other configurations

BLAST Time vs. Cost



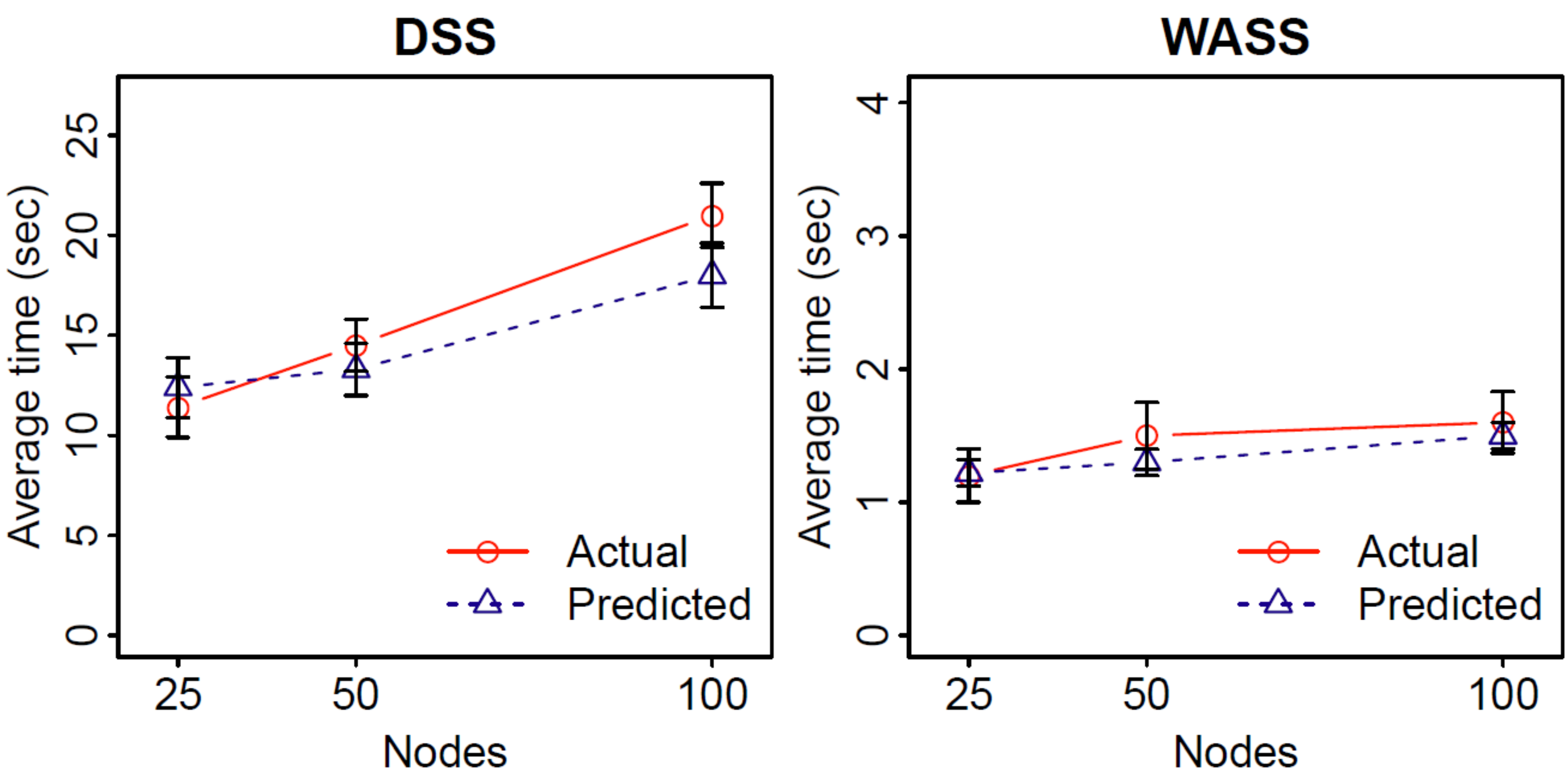
What is the impact of handling a complex application at large scale?

Montage Performance



Accurate
~2000x less resources

Pipeline on 100 nodes



Spinning Disks

[Add summary from thesis]

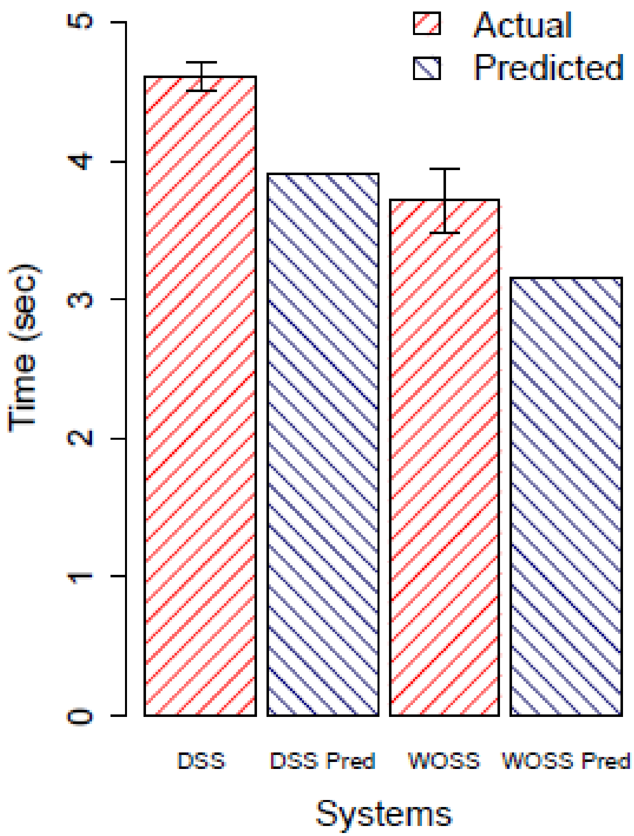
SDD trend

Supercomputers have no HDDs

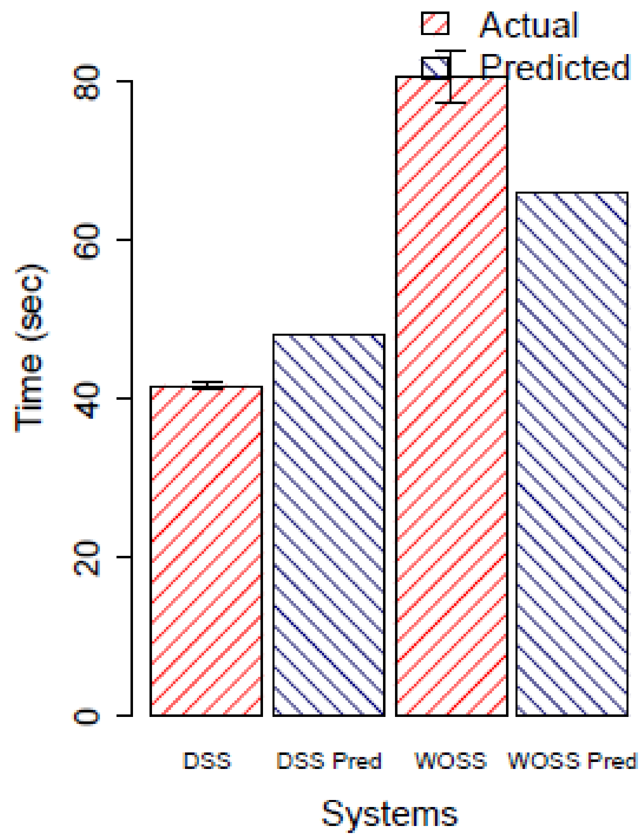
Other solutions for RAM-based

– E.g., Tachyon has grown

Spinning Disks: Worst Case

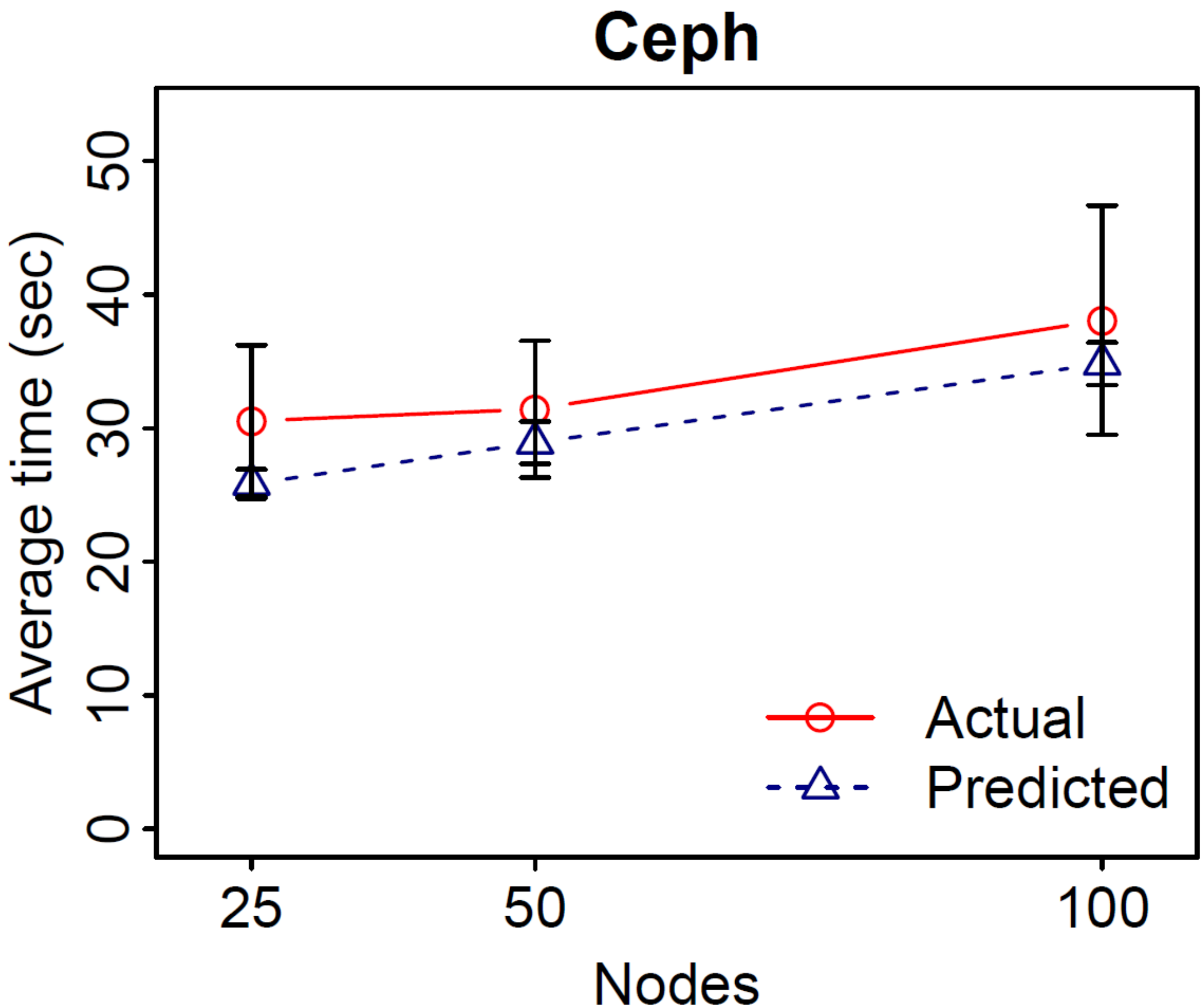


(a) Medium Workload



(b) Large Workload

Predicting Ceph



Other Scenarios

Various testbeds and benchmarks

- Similar results

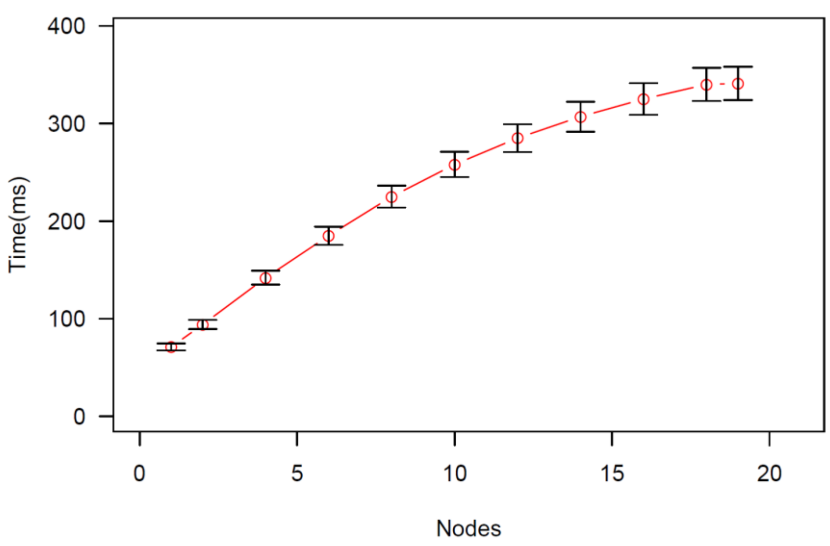
Online enabling data deduplication for checkpointing applications

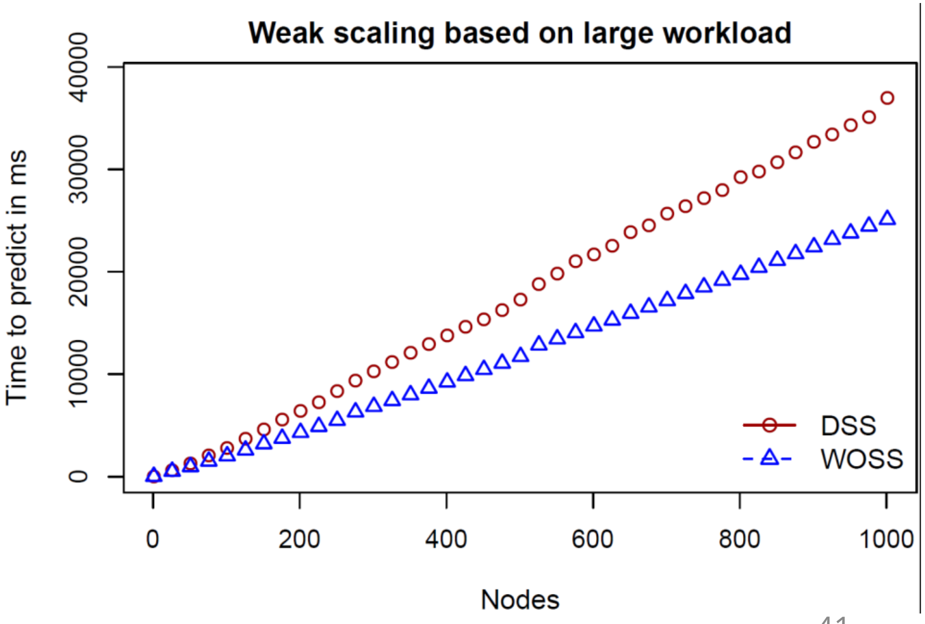
Energy Prediction

- Power consumption profile approach
- Workflow: Synthetic benchmarks have ~13% error; smaller Montage, ~26%
- Deduplication: Misprediction costs up to 10%

Predictor Scalability

- Summary of the text





Energy Model

Execution States:		Energy		Power Profile * Predicted Times
Idle	$\longrightarrow$	E^{idle}	$\longrightarrow$	$P_i^{idle} * T^{total}$
Network Transfer	$\longrightarrow$	ΔE^{net}	$\longrightarrow$	$(P^{net} - P^{idle}) * T_i^{net}$
I/O ops (read, write)	$\longrightarrow$	$\Delta E^{storage}$	$\longrightarrow$	$(P^{storage} - P^{idle}) * T_i^{storage}$
Task Processing	$\longrightarrow$	ΔE^{app}	$\longrightarrow$	$(P^{App} - P^{idle}) * T^{App}$

How to seed the energy model?

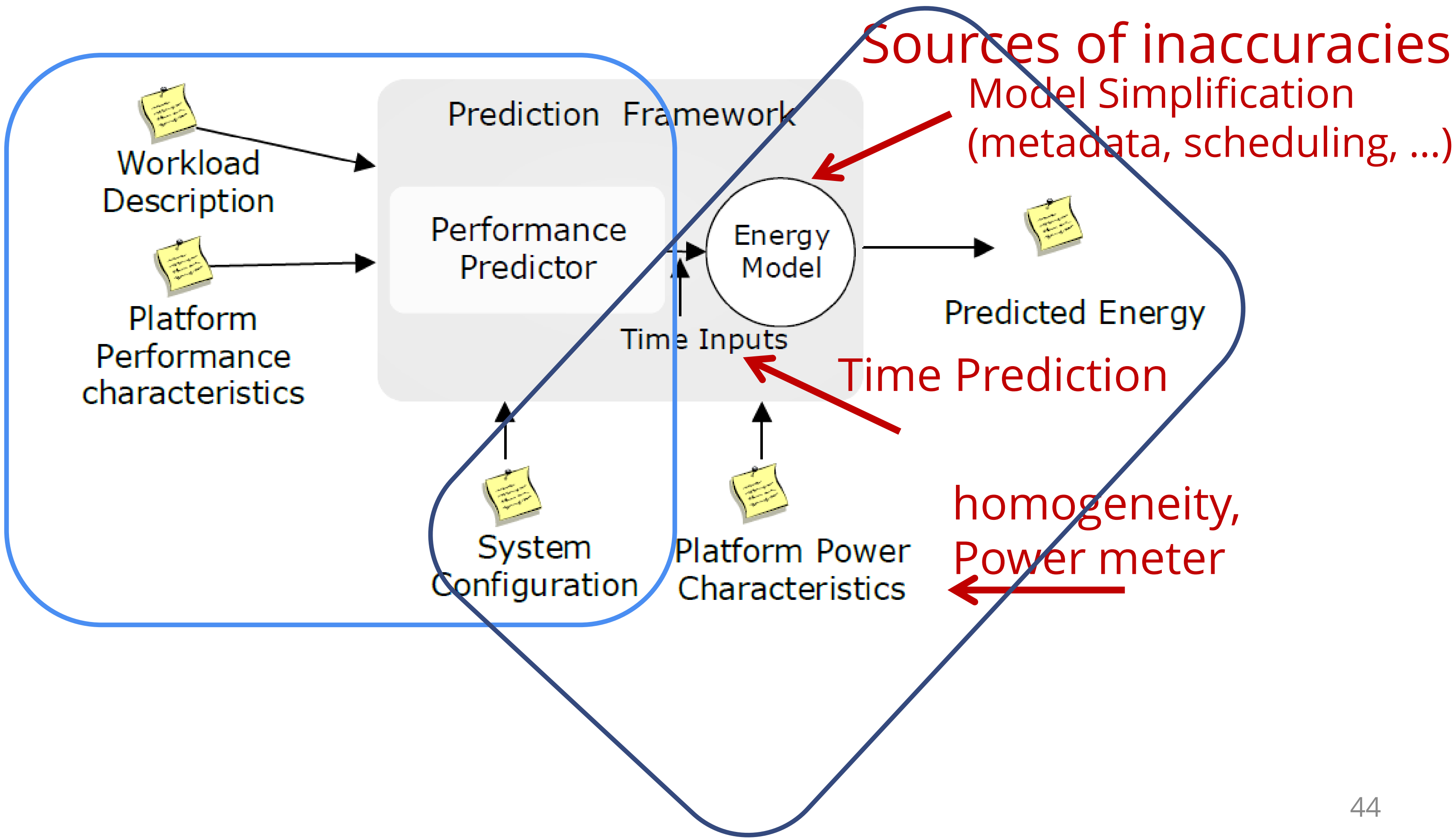
Power states

- uses synthetic benchmarks to get the power consumption in each state

Time estimates

- augments a performance predictor to track the time spent in each state.

Building Energy Predictor



Energy Evaluation: Testbed

Grid'5000

Idle	P_i^{idle}	91.6W
App	$P_i^{App} - P_i^{idle}$	33.6W
Storage I/O	$P_i^{storage} - P_i^{idle}$	37.4W
Net transfer	$P_i^{net} - P_i^{idle}$	36.1W

Taurus Cluster (11 nodes)

two 2.3GHz Intel Xeon E5-2630 CPUs (each with 6 cores), 32GB memory, 10 Gbps NIC

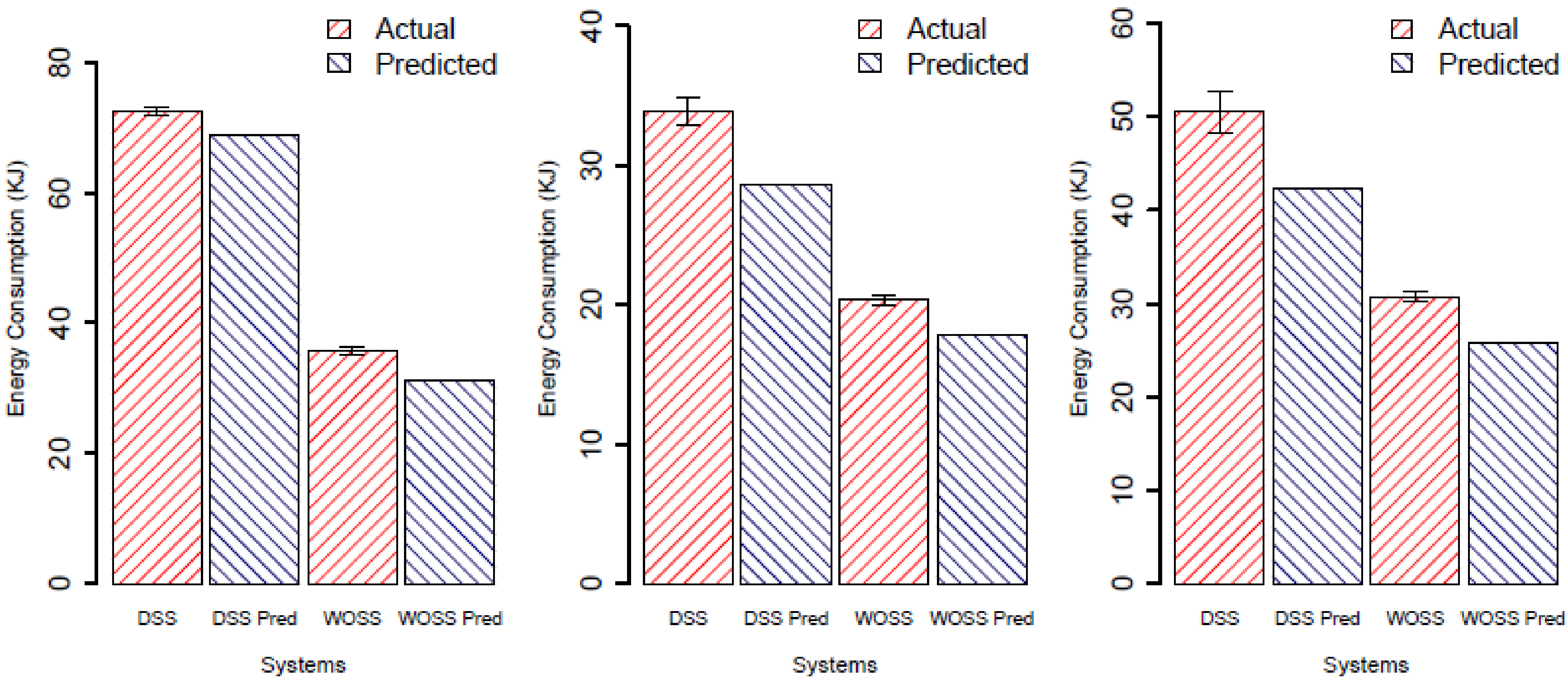
Sagittaire Cluster (16 nodes)

two 2.4GHz AMD Opteron CPUs (each with one core), 2GB RAM and 1 Gbps NIC

SME Omegawatt power-meter per Node

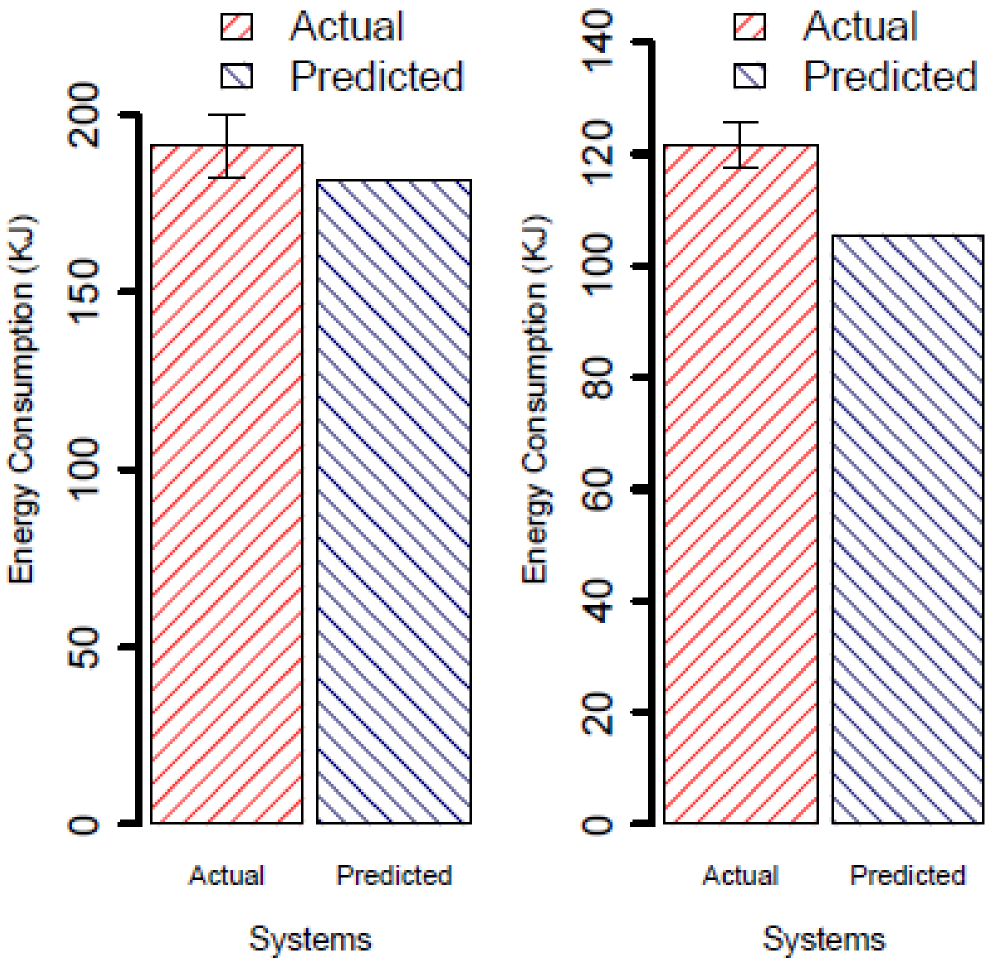
0.01W power resolution at 1Hz sampling rate

Energy Prediction Evaluation

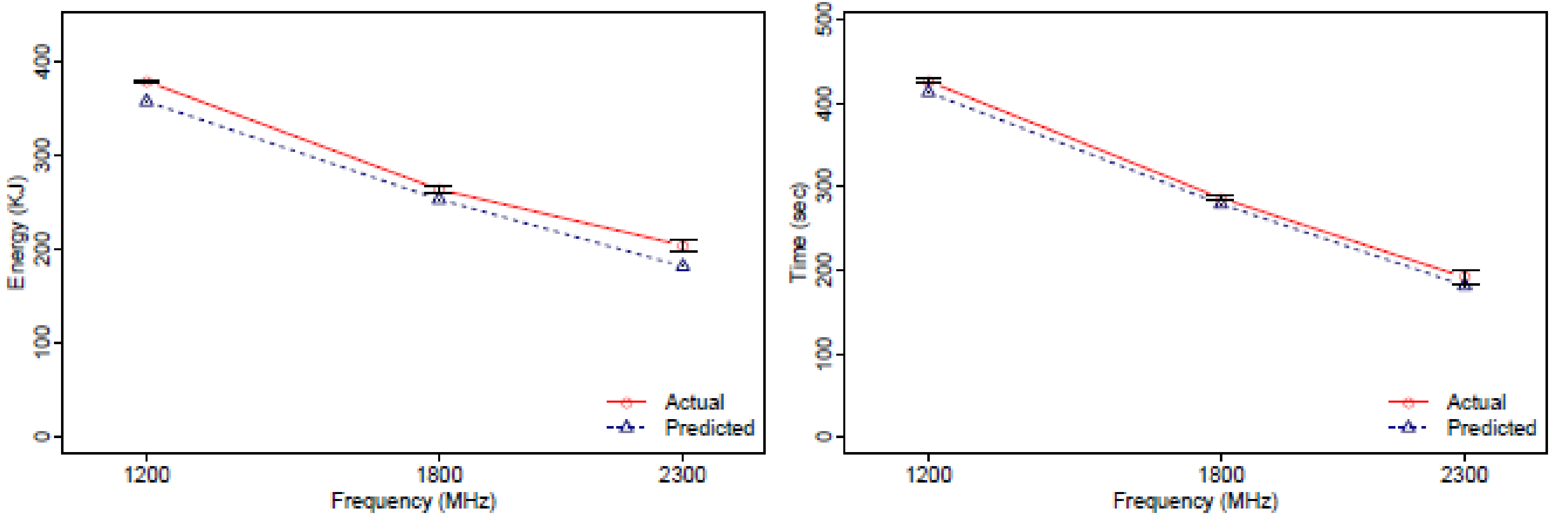


(a) Pipeline **(b)** Reduce **(c)** Broadcast

Energy Prediction Evaluation

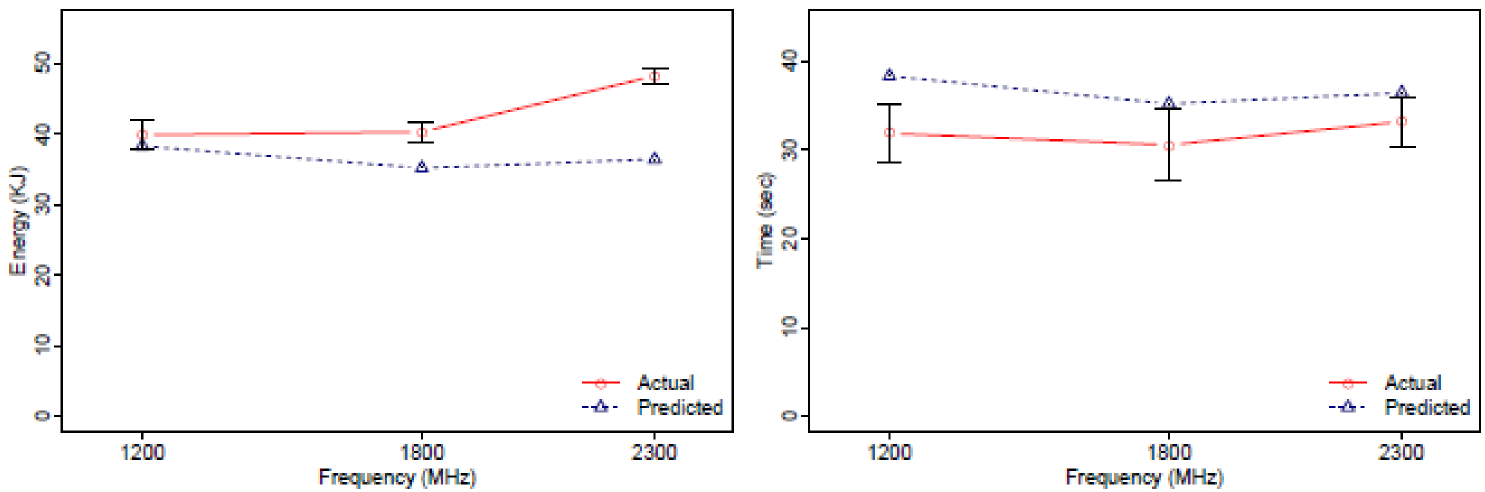


(a) BLAST (b) Montage



(a) Energy **(b)** Time

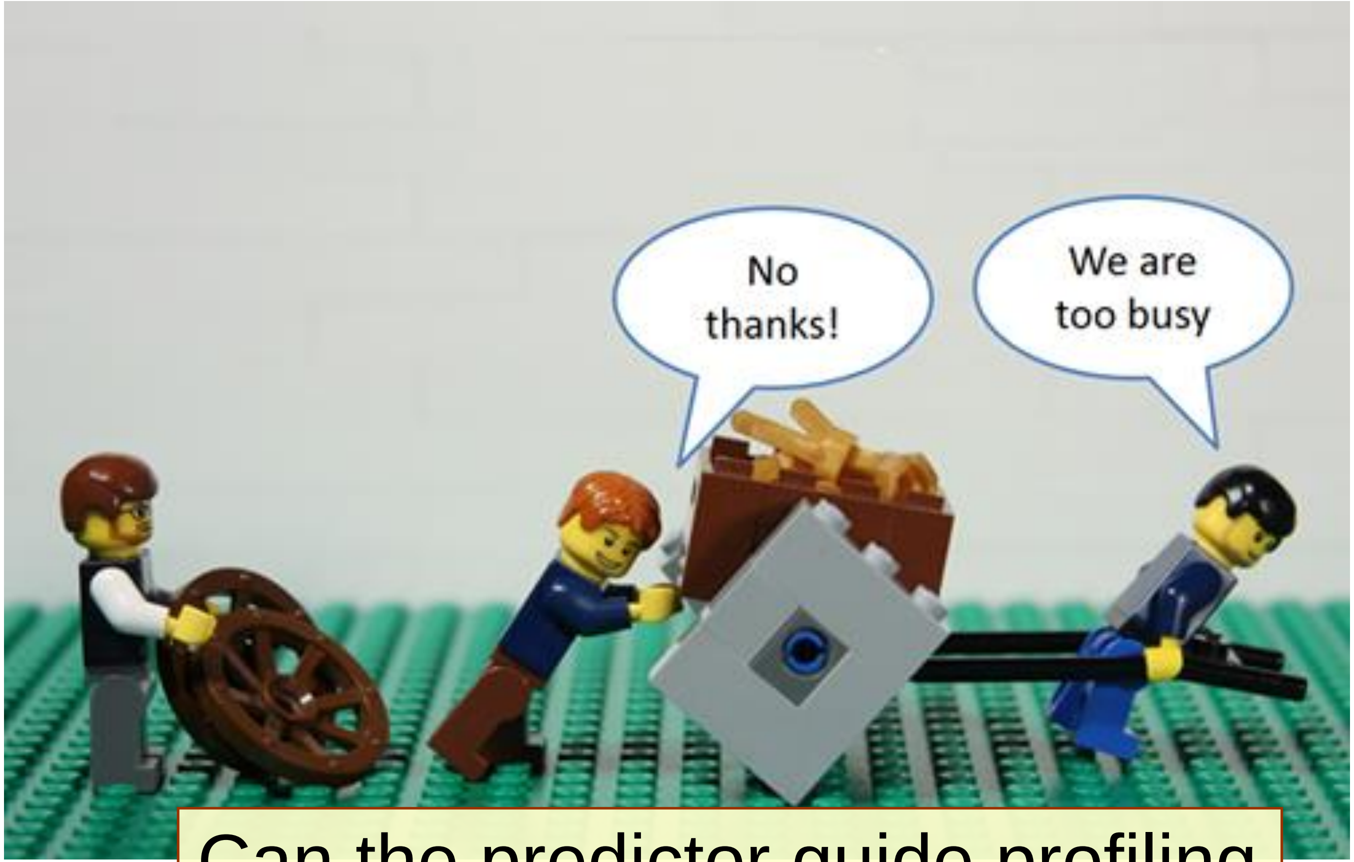
Figure 3.20: Actual and predicted average energy consumption and execution time for *BLAST* for various CPU frequencies.



(a) Energy **(b)** Time

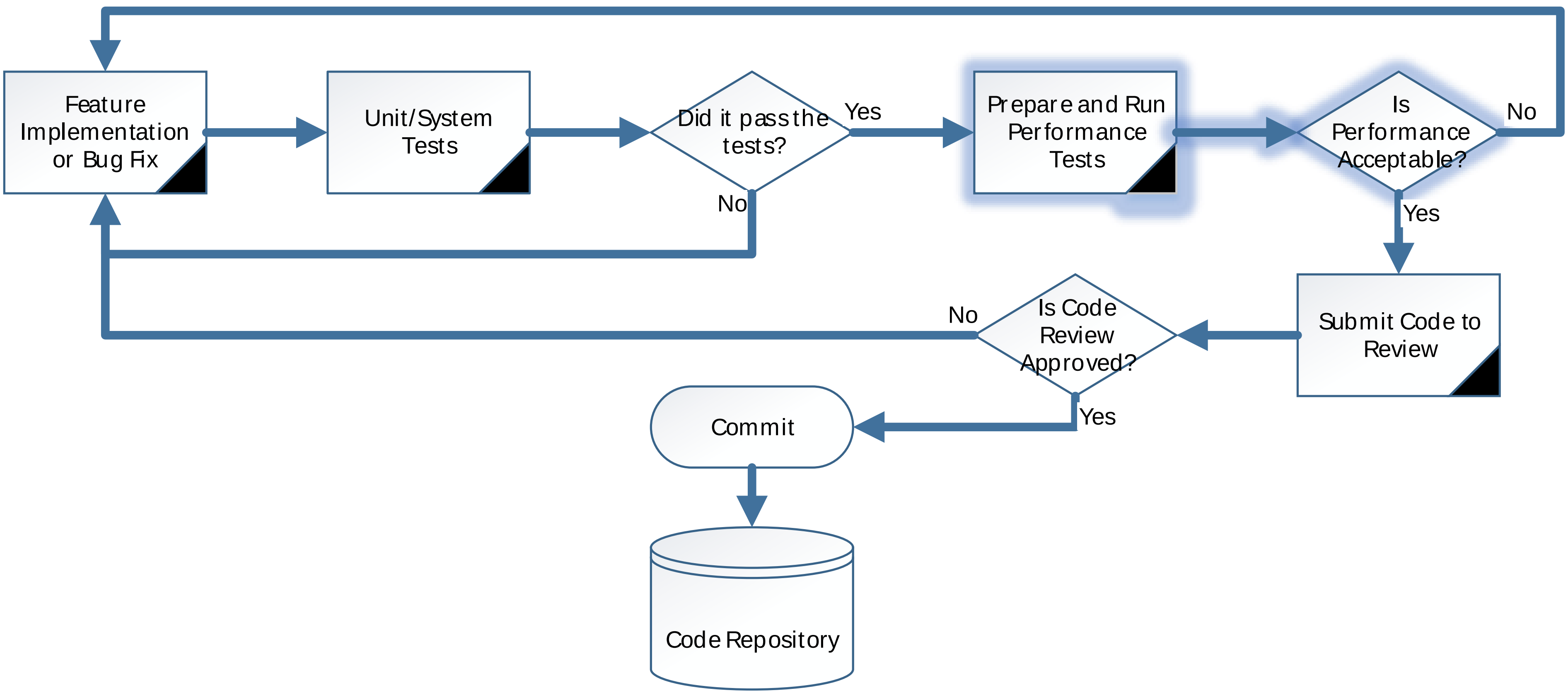
Figure 3.21: Actual and predicted average energy consumption and execution time for the *pipeline benchmark* for various CPU frequencies.

Supporting Development



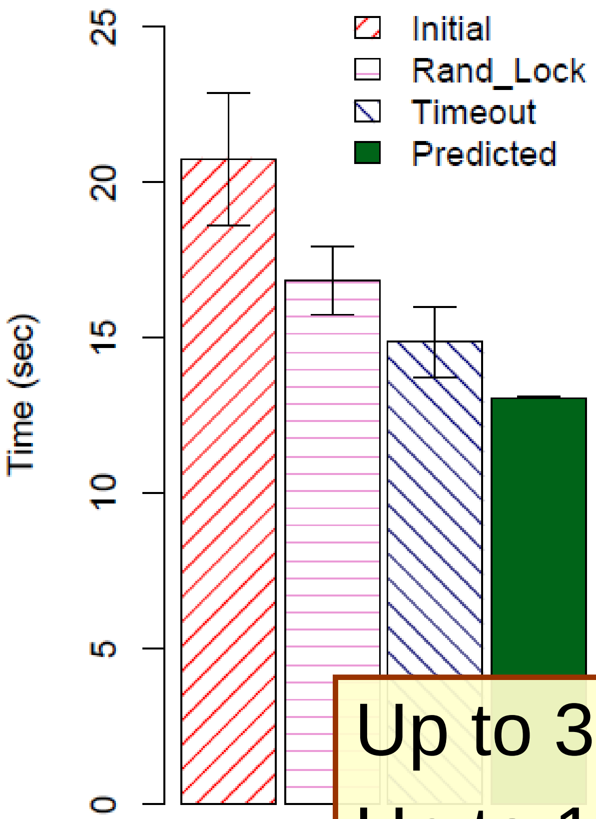
Can the predictor guide profiling and debugging efforts?

Development Flow

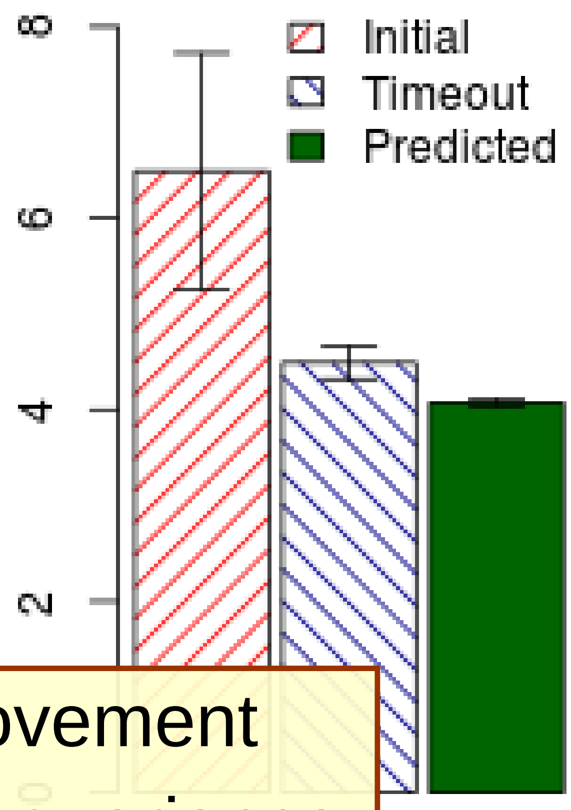


Development Evolution

Pipeline



Reduce



Up to 30% improvement
Up to 10x smaller variance

Future Work

Enhance Automation for Workflows

Heterogeneous Environment

Virtual Machines

Study on Support for Development

Applications out of Comfort Zone

GPU and Content-Based Chunking for Deduplication

Limitations

“Short” tasks

- Sensitive to any ‘noise’ or scheduling overhead
- e.g., up to 40% error in a Montage phase

At least one whole execution

- Limits heterogeneity exploration

Potentially, different network topologies

Old spinning disks

Sources of Inaccuracies

Source	Examples
Storage system	Fine granularity for the activity inside each component, detailed execution path, or maintenance services such as failure detection and garbage collection.
Infrastructure	Contention at the network fabric level, complex network topology, or detailed scheduling overhead.
Application	Tasks launched at the same time, absence of faults by crash, or machines with degraded performance.
System identification	Assumptions about client and storage service times.

Related Work: Different Target

Storage enclosure focused vs. distributed (e.g., HP Minerva)

Focus on per I/O request (e.g., average of many)

Lack prediction on the total execution time

Not on workflow applications, or data deduplication (e.g., Herodotou ’11)

Guide configuration using actual executions or ‘machine-learning’ models (e.g., Behzad ’13, ACIC ‘14)

Modeling

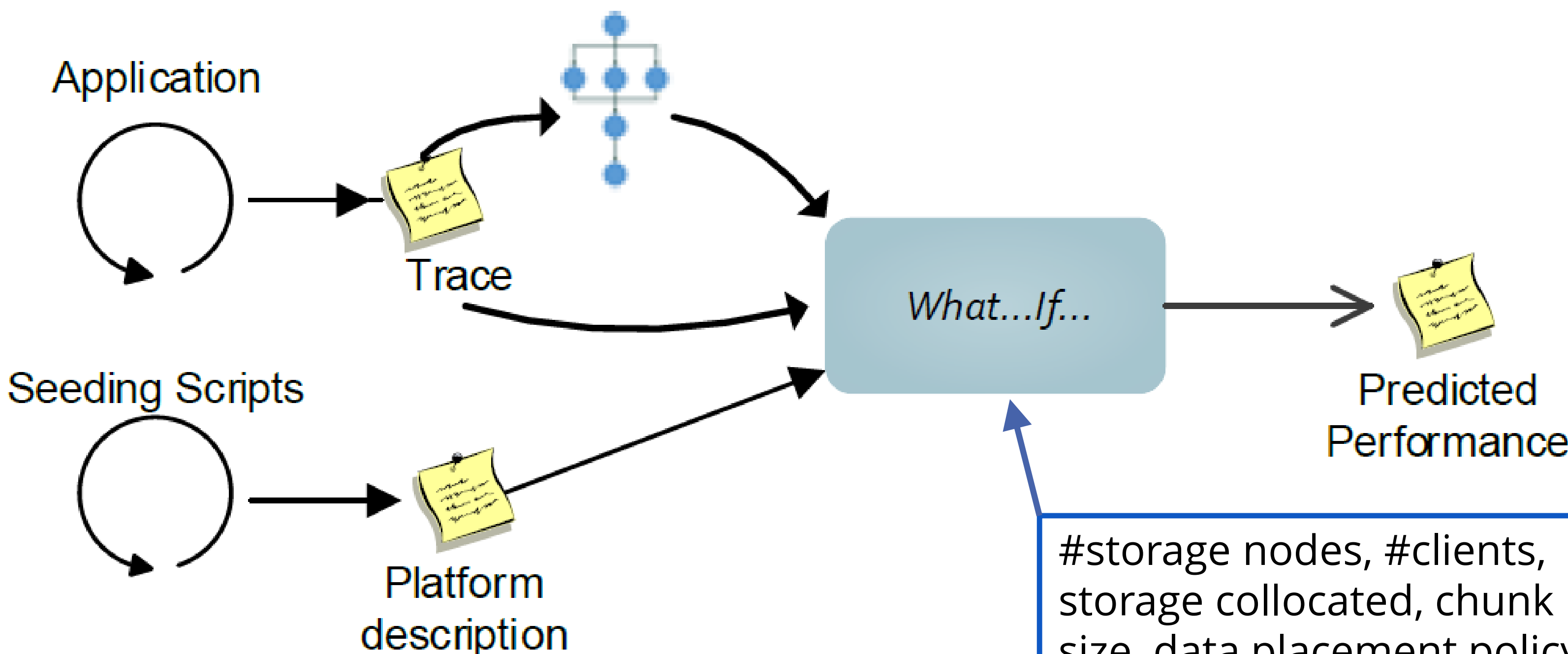
Approaches

White-Box	Black-Box
Simulations	Machine Learning
Analytical Models	

Properties

White-Box	Black-Box
Fine Granularity	Coarse Granularity
Less Data	More Data (to train)
More Exploratory	Close Already Deployed
Detailed Seeding	Application-Level Seeding

Architecture



#storage nodes, #clients, storage collocated, chunk size, data placement policy, cache size, stripe width, replication level

Scheduling: workqueue, workqueue + data aware.

Data Deduplication

- Storage technique to save storage space and improve performance

- Space savings can be as high as:
  - 60% for a generic archival workload[1]
  - 85% for application checkpointing[2]
  - 95% for a VM repository[3]

[1] S. Quinlan and S. Dorward, "*Venti: A new approach to archival data storage*," FAST '02.
[2] S. Al-Kiswany *et al.* "stdchk: A checkpoint storage system for desktop grid computing," ICDCS, 2008.
[3] A Liguori, E V Hensbergen. "Experiences with content addressable *storage and virtual disks*, *(WIOV)*, 2008.

Data Deduplication

- It performs hash computations over data to detect data similarity
  - Saving storage space

- It has computational overhead, but it can reduce I/O operations
  - Improving performance
  - Impact on energy?

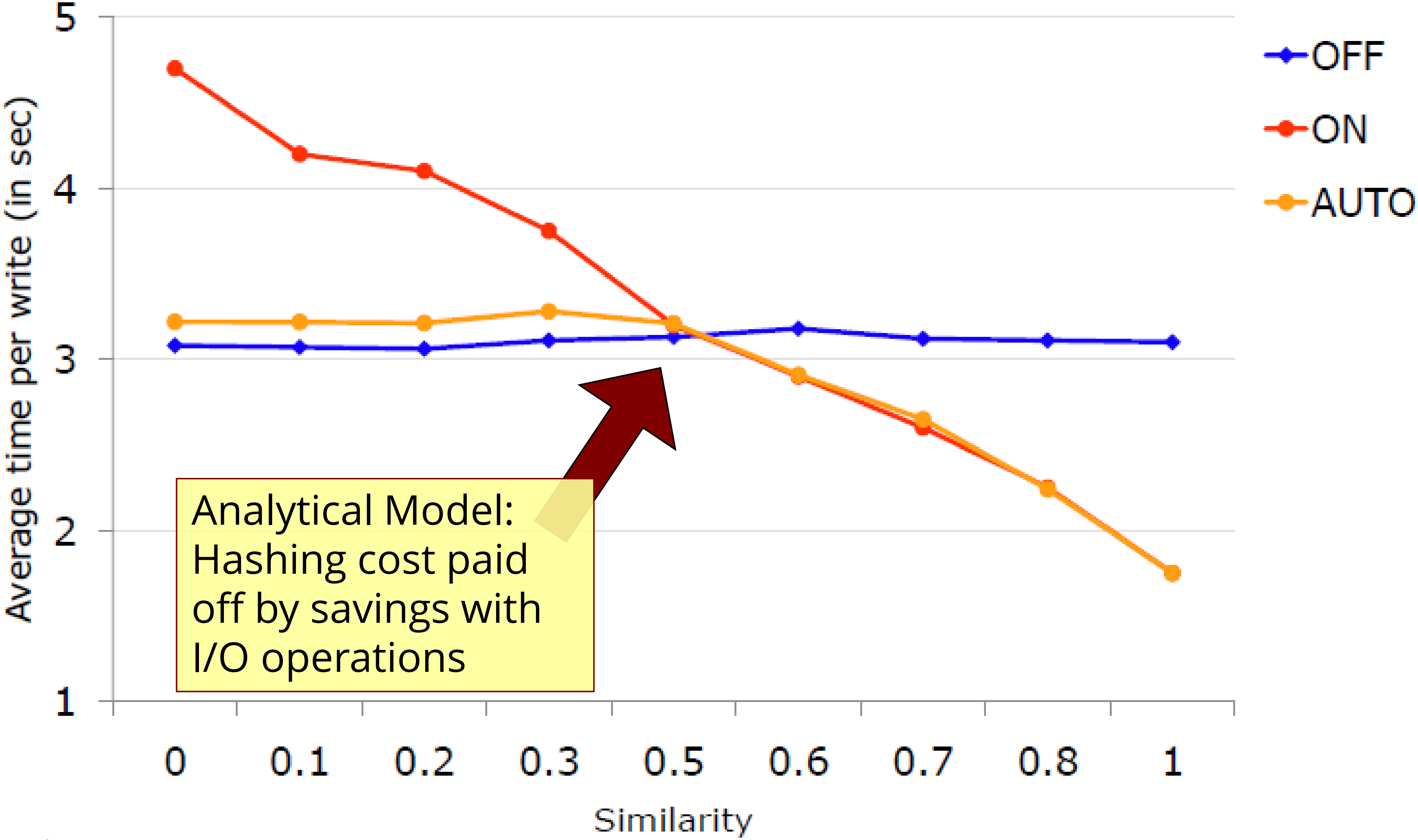
Deduplication for Checkpointing?

Checkpointing writes multiple snapshots

- Snapshots may have high data similarity

Deduplication detects similarity to save storage space and network bandwidth, but has high computational cost

Optimizing for Time



[1]L.B. Costa and M. Ripeanu, “Towards automating the configuration of a distributed storage system,” *2010 11th IEEE/ACM International Conference on Grid Computing*, IEEE, 2010, pp. 201-208.

What cases will lead to energy savings, if any?

What is the performance impact of energy-centric tuning?

What is the impact of more energy proportional hardware?

Energy Study - Methodology

- Empirical evaluation on a distributed storage system
- Identify break-even points for performance and energy
- Provide a simple analytical model

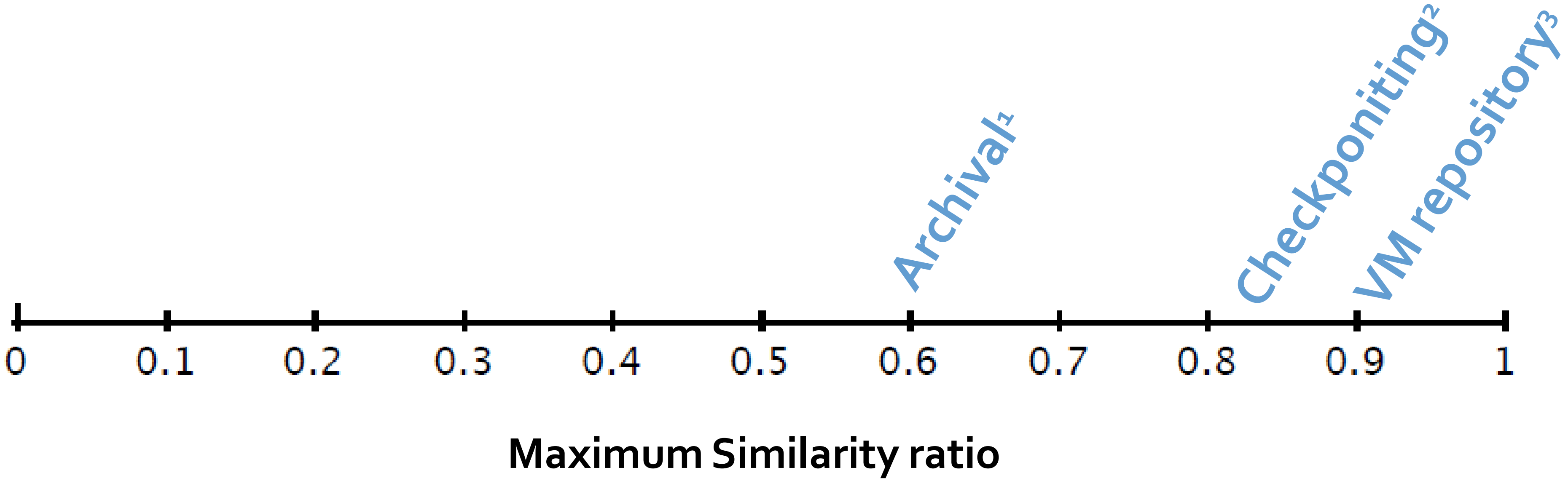
Test Bed

	Processor Launched	Processor	Memory	Power Idle	Power Peak
Old	Q4’06	Xeon E5395 (Clovertown) @ 2.66GHz	8GB	188W	262W
New	Q1’09	Xeon E5540 (Nehalem) @ 2.53GHz	48GB	86W	290W

Both: Similar NIC 1Gbps and 7200 rpm SATA disks

Synthetic Workload

- It varies similarity ratios
- It covers similarity ratios of several applications

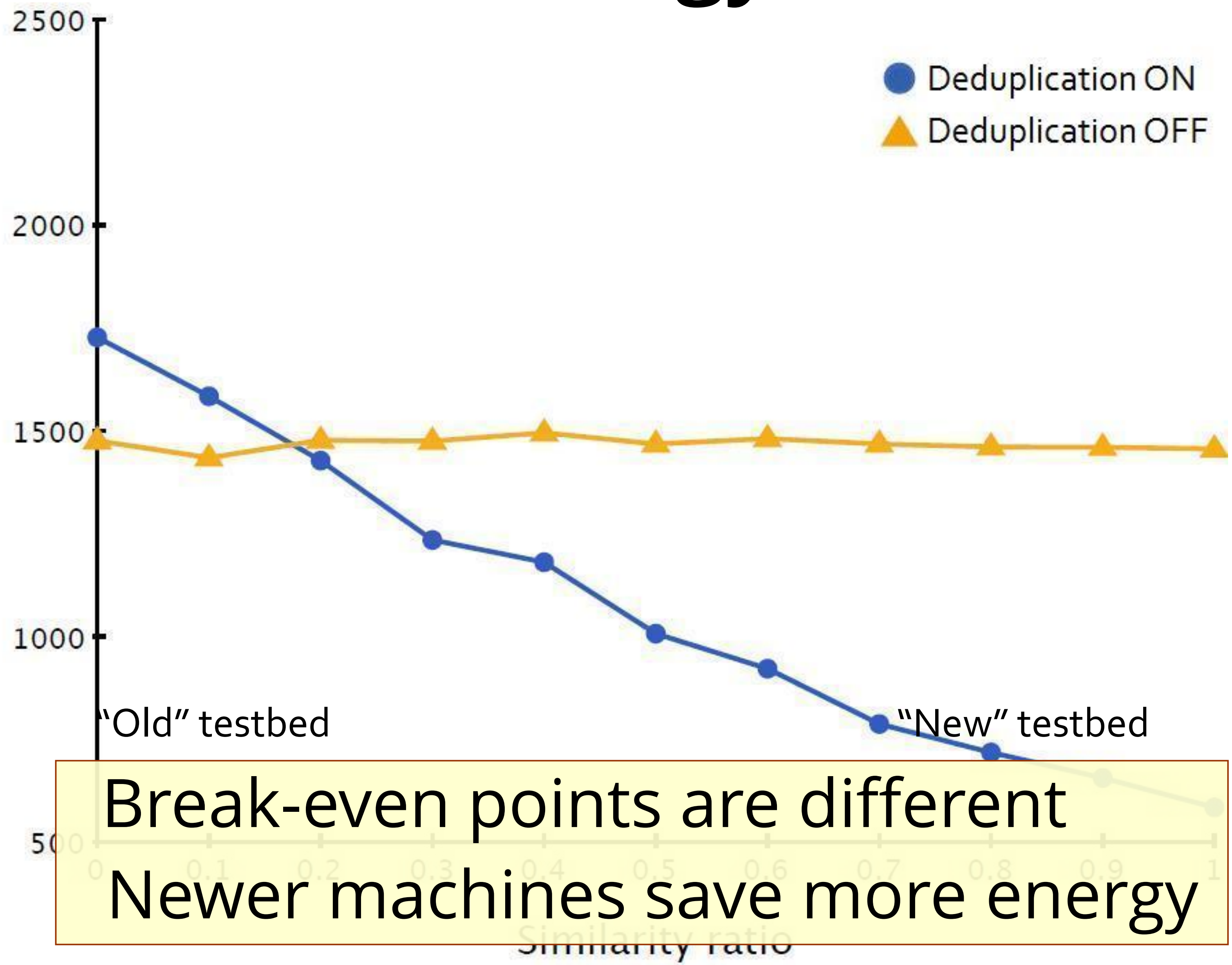


[1] S. Quinlan and S. Dorward, "*Venti: A new approach to archival data storage*," FAST '02.
[2] S. Al-Kiswany *et al.* "stdchk: A checkpoint storage system for desktop grid computing," ICDCS, 2008.
[3] A Liguori, E V Hensbergen. "Experiences with content addressable *storage and virtual disks*, *(WIOV)*, 2008.

What cases will lead to energy savings, if any?

What is the impact of more energy proportional hardware?

Energy

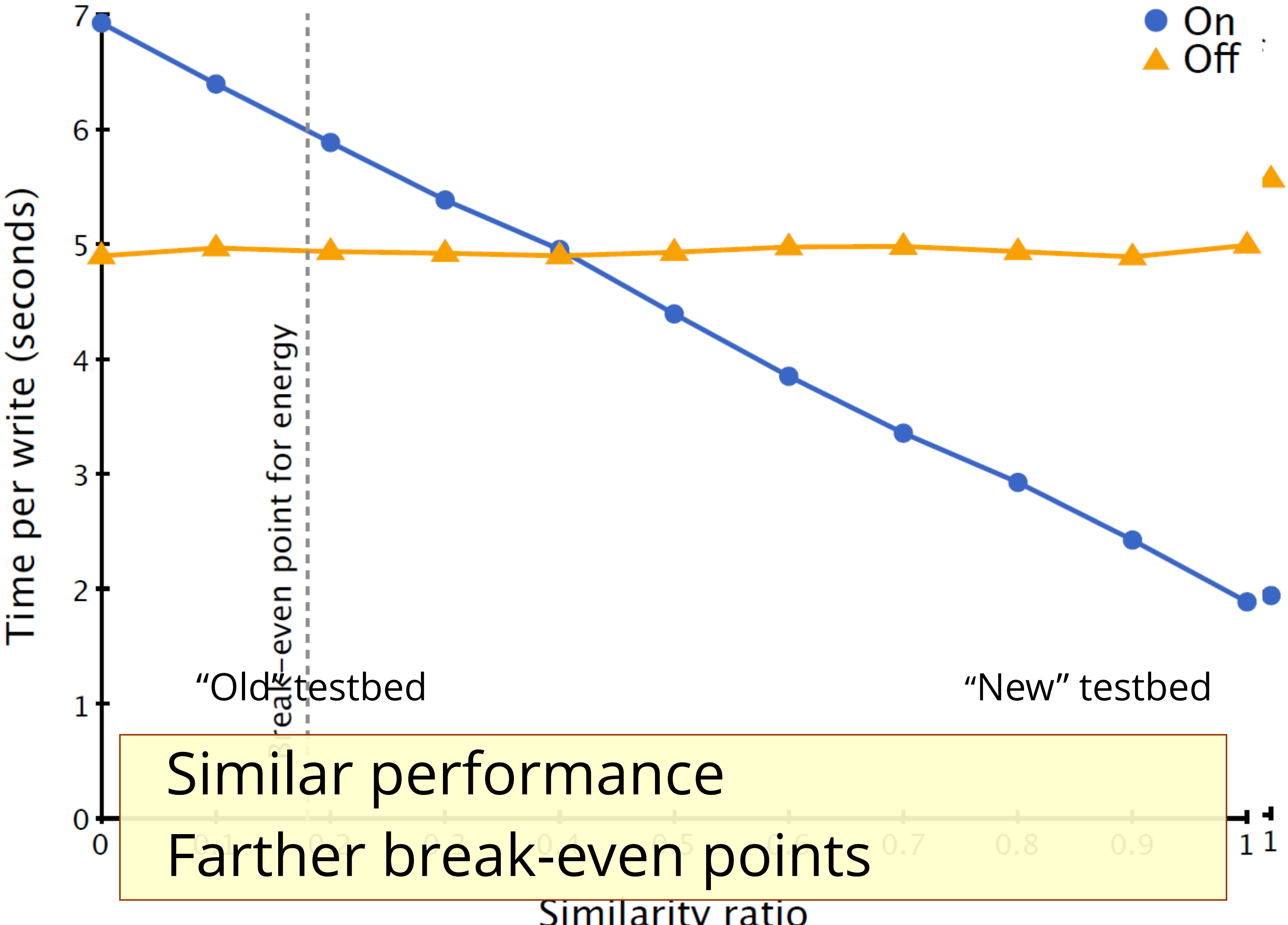


Break-even points are different

Newer machines save more energy

What is the performance impact of energy-centric tuning?

Write Time / Old Testbed / New Testbed



Similar performance

Farther break-even points

Summary of Evaluation

Deduplication can save energy

Newer machines showed little difference for performance, larger difference for energy

- Energy proportional hardware

Break-even points for performance and energy are different

- Trend to be farther

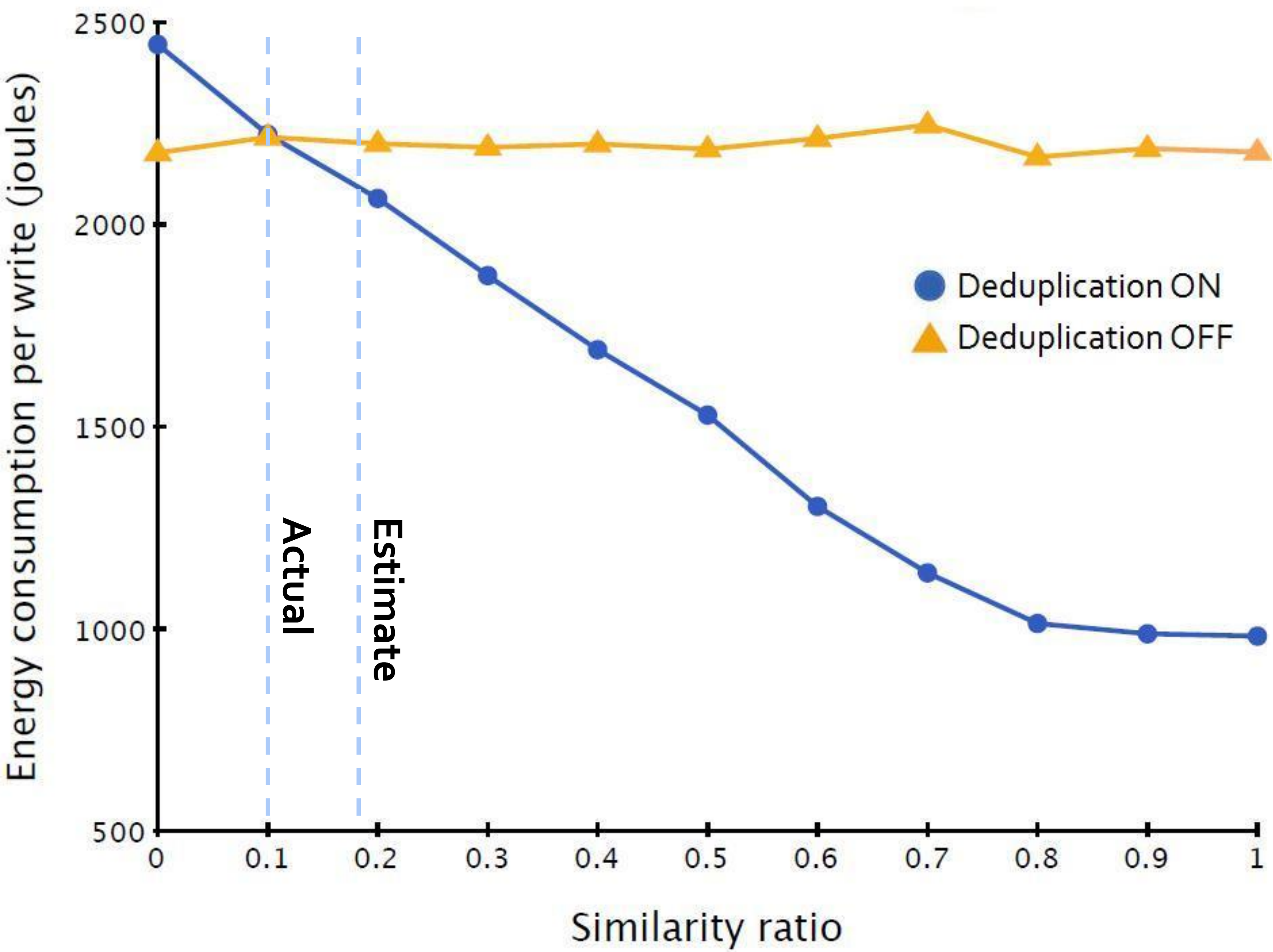
Model Input and Output

Simple benchmarks provide information on:

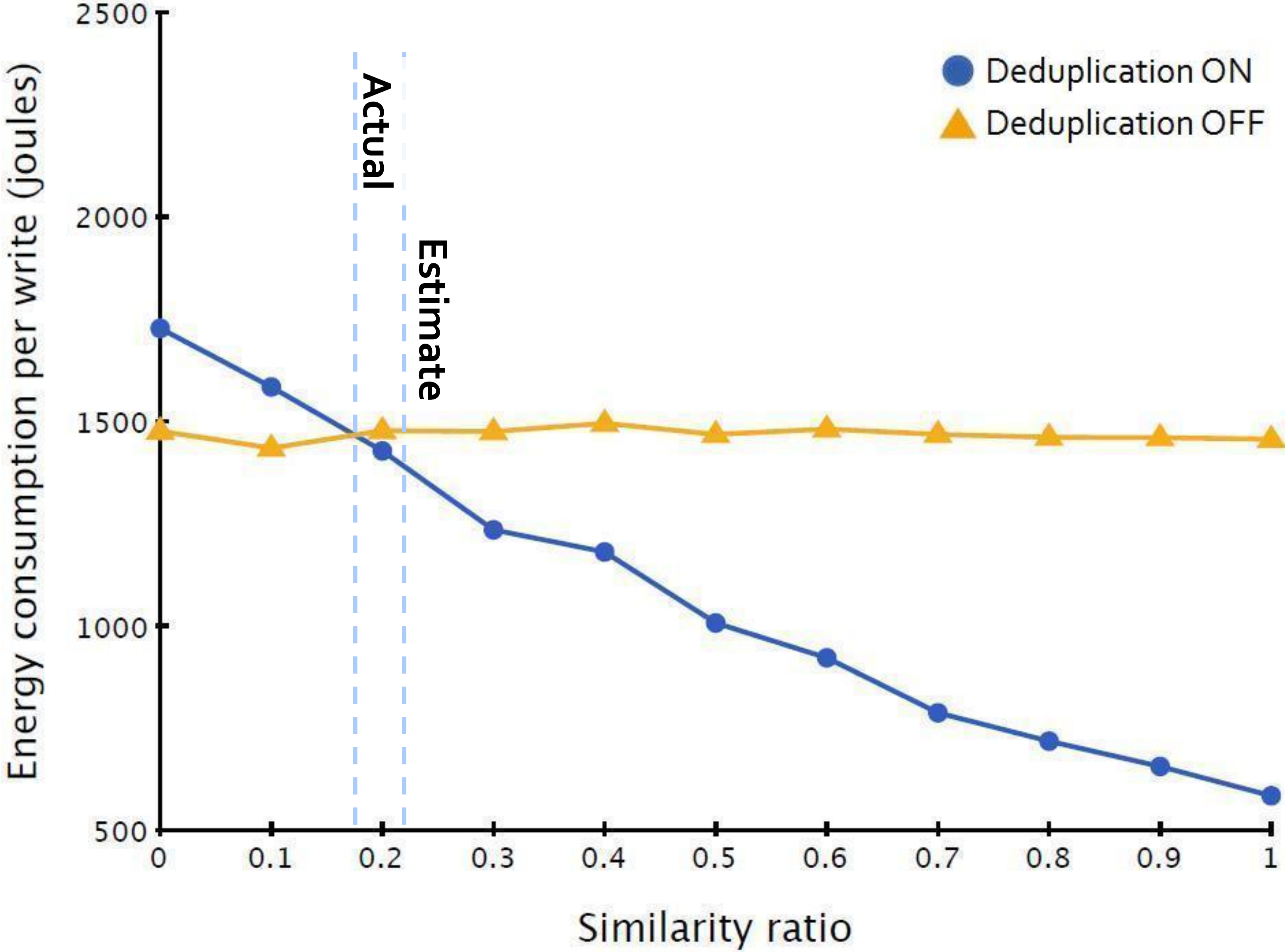
- Time to write and hash a block
- Power to write and hash a block

Model gives the similarity ratio of the break-even point

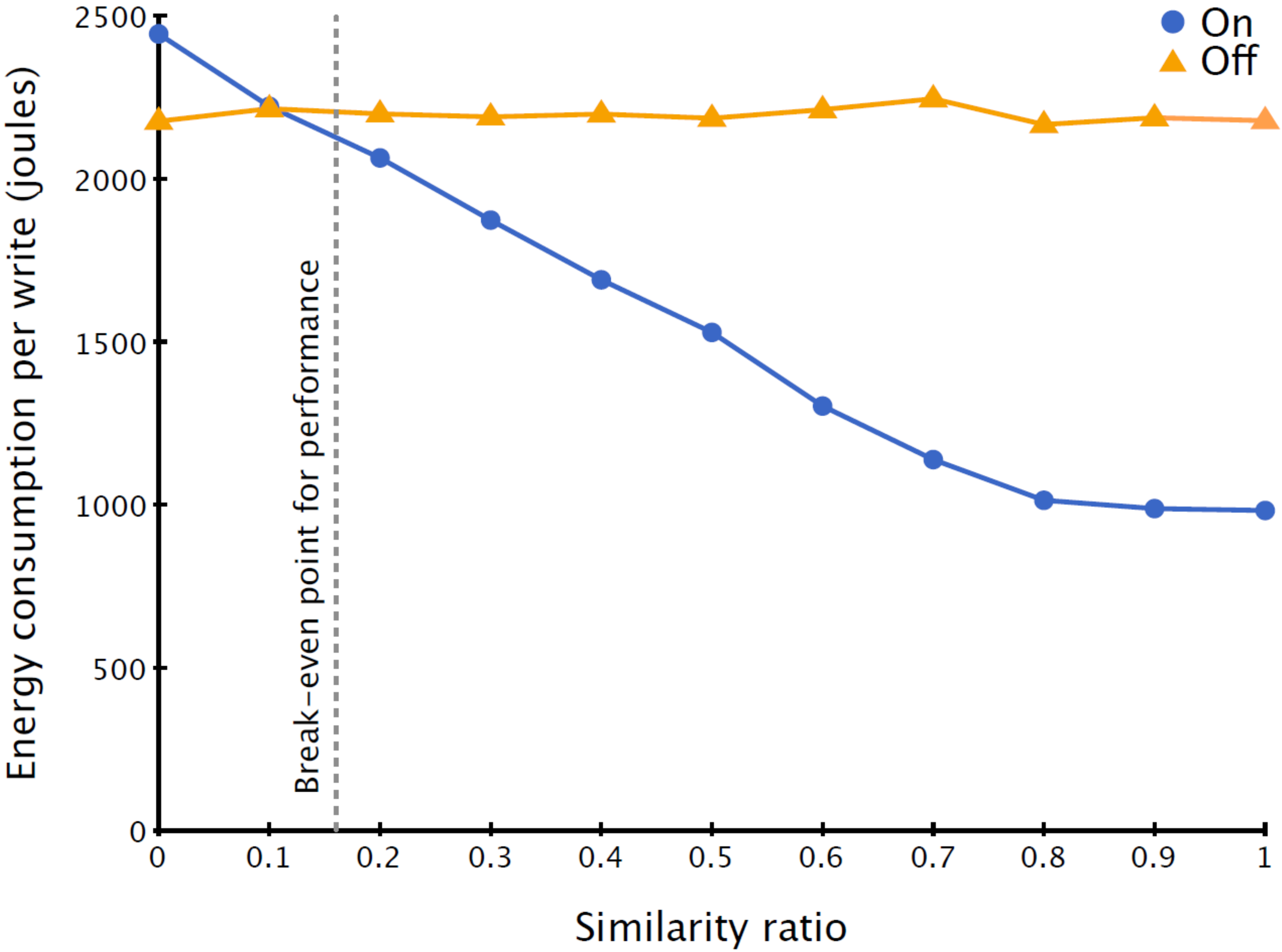
Actual vs. Model – Old test bed



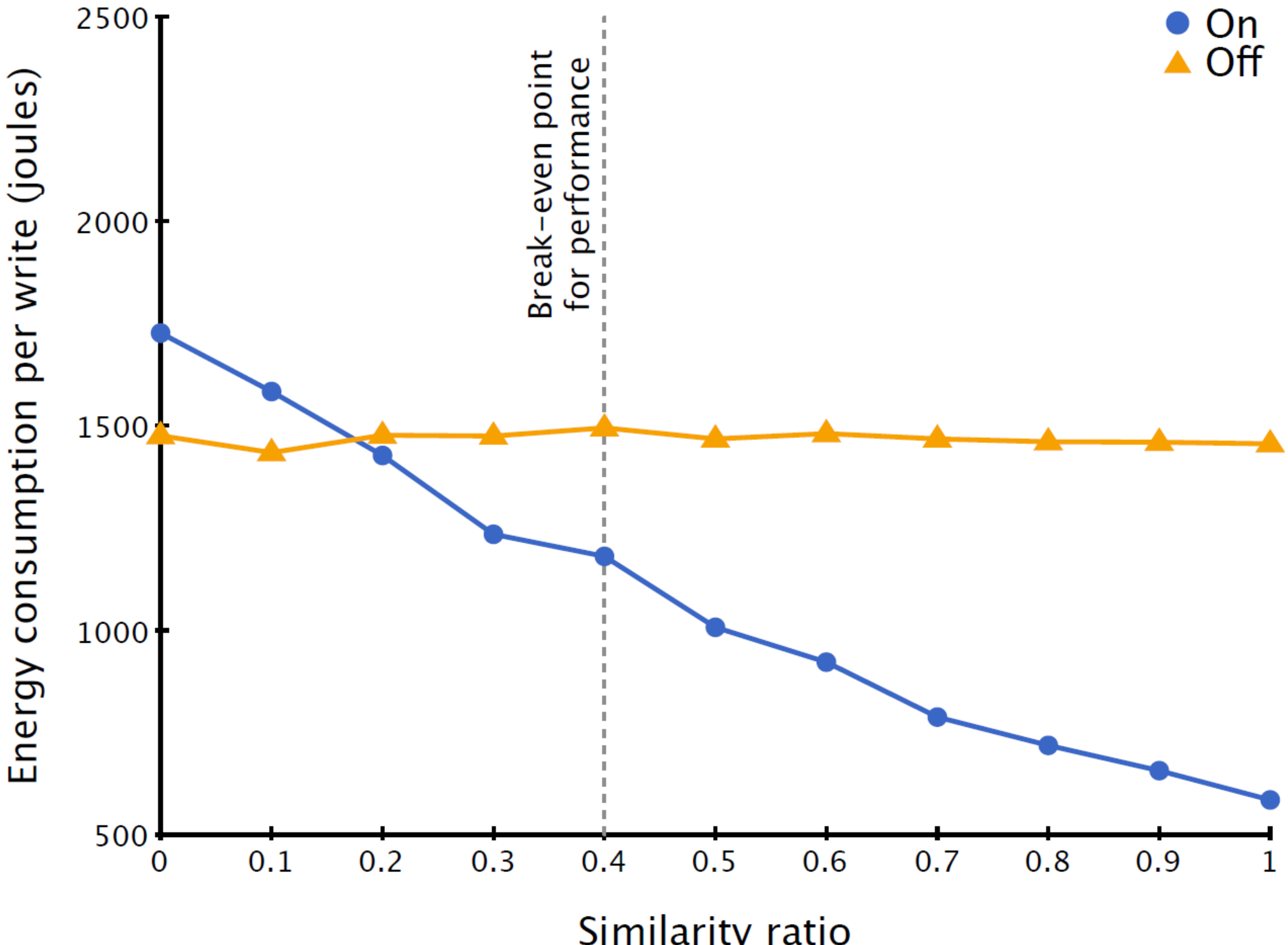
Actual vs. Model – New test bed



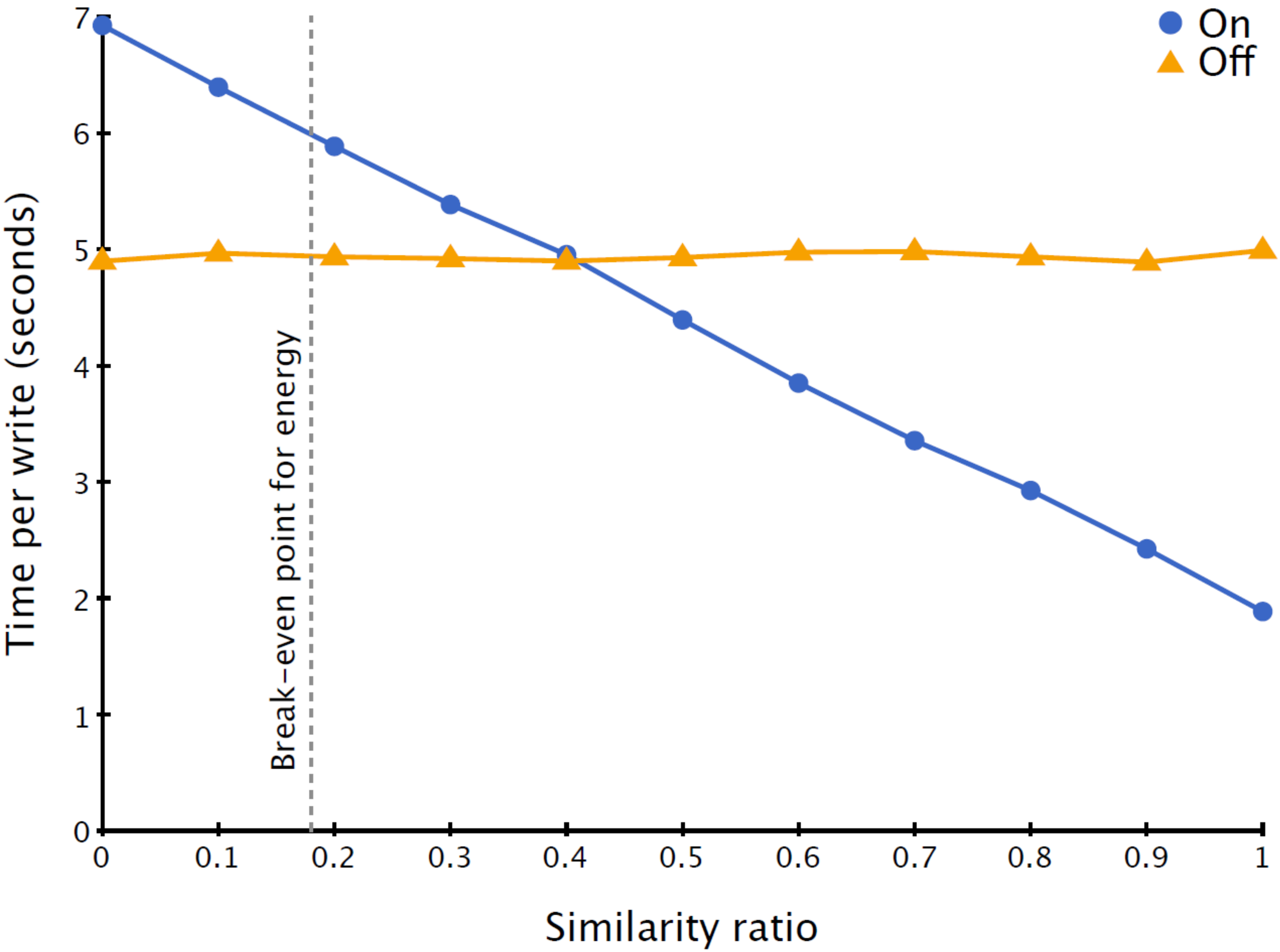
Energy - Old testbed



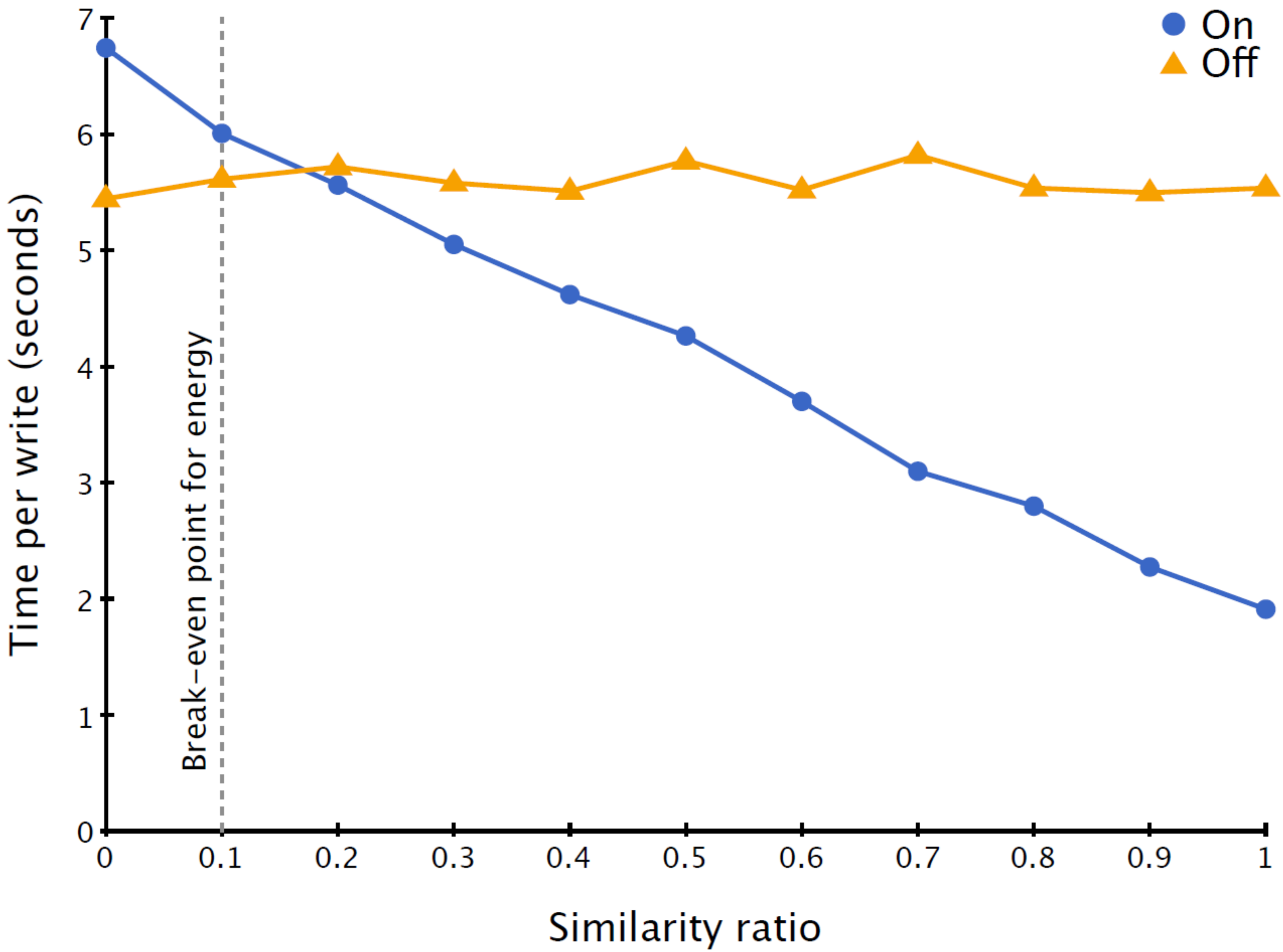
Energy - New testbed

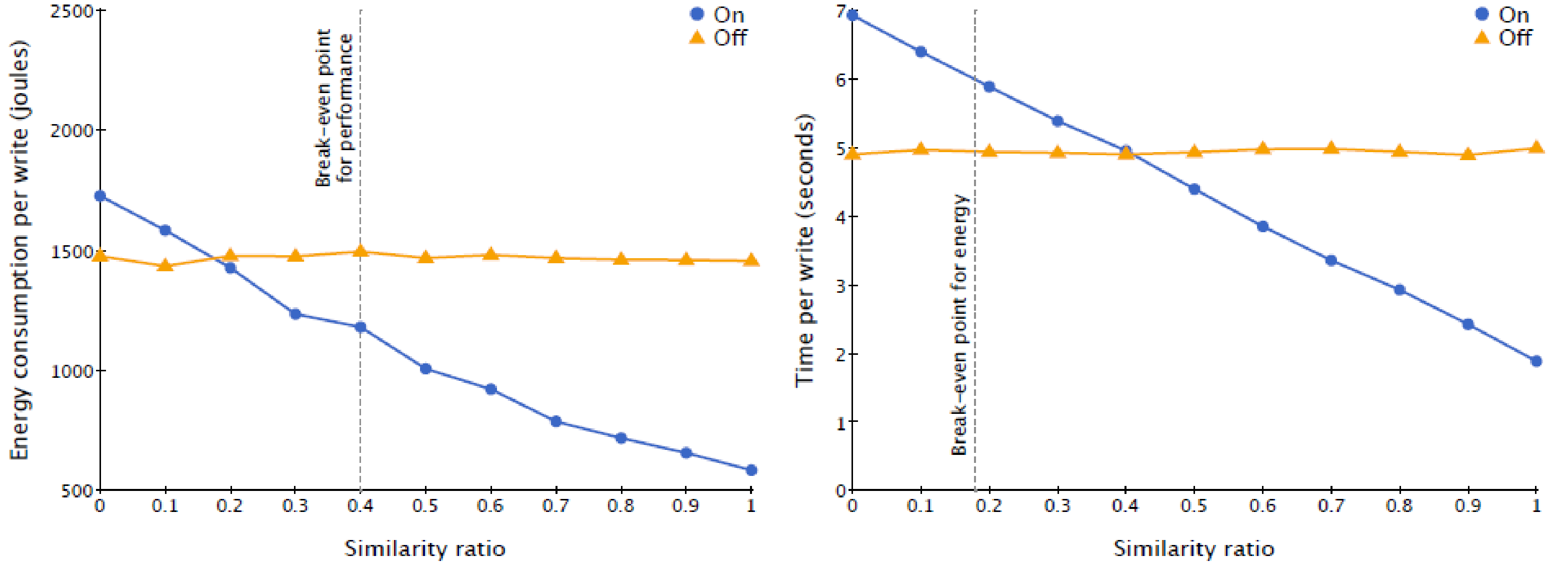


Write Time - New testbed



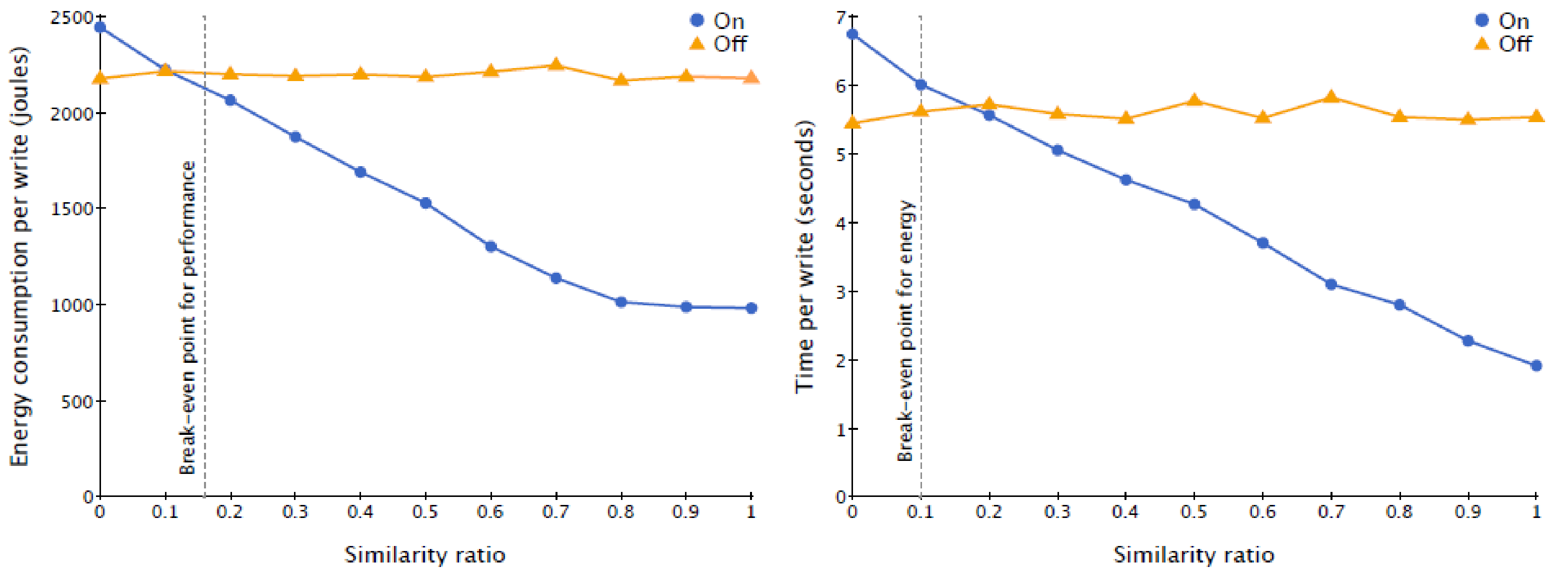
Write Time - Old testbed





(a) Average energy consumption (b) Average time to write

Figure 1. Average energy consumed and time to write a 256MB file for different similarity levels in the *'new'* testbed. Note: Y axes do not start at 0.



(a) Average energy consumption (b) Average time to write

Figure 2. Average energy consumed and time to write a 256MB file for different similarity levels in the *'old'* testbed. Note: Y axes do not start at 0

Methodology: Development

Unit tests

System tests

Code reviews

Some TDD

Workflow Applications on a Shared Storage

Simplicity for development, and debugging

- Application can be developed on a single workstation, and deployed on a cluster without changes

Support for legacy applications

- Stages or binaries can be easily integrated, since the communication via POSIX

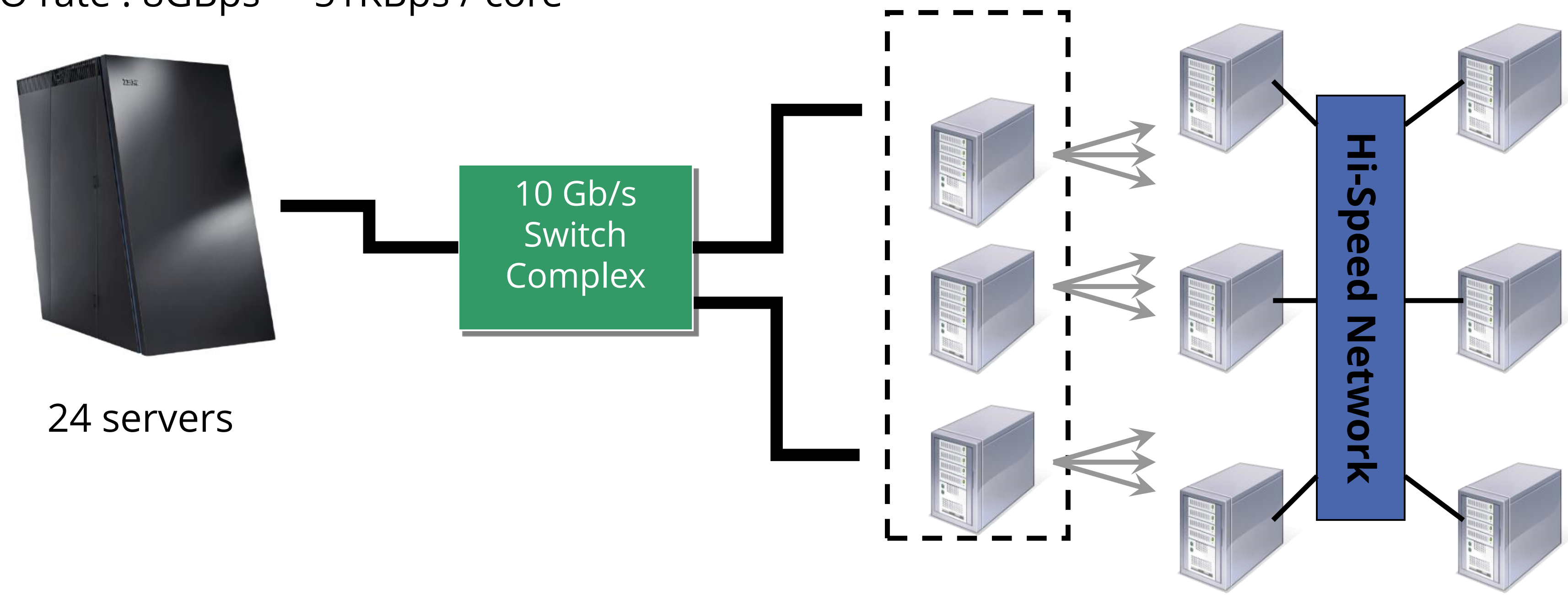
Support for fault-tolerance

- Keeping the task's input files and launching a new execution of the task, potentially on a different machine

Platform Example – Argonne BlueGene/P

GPFS **2.5K IO Nodes** **160K cores**

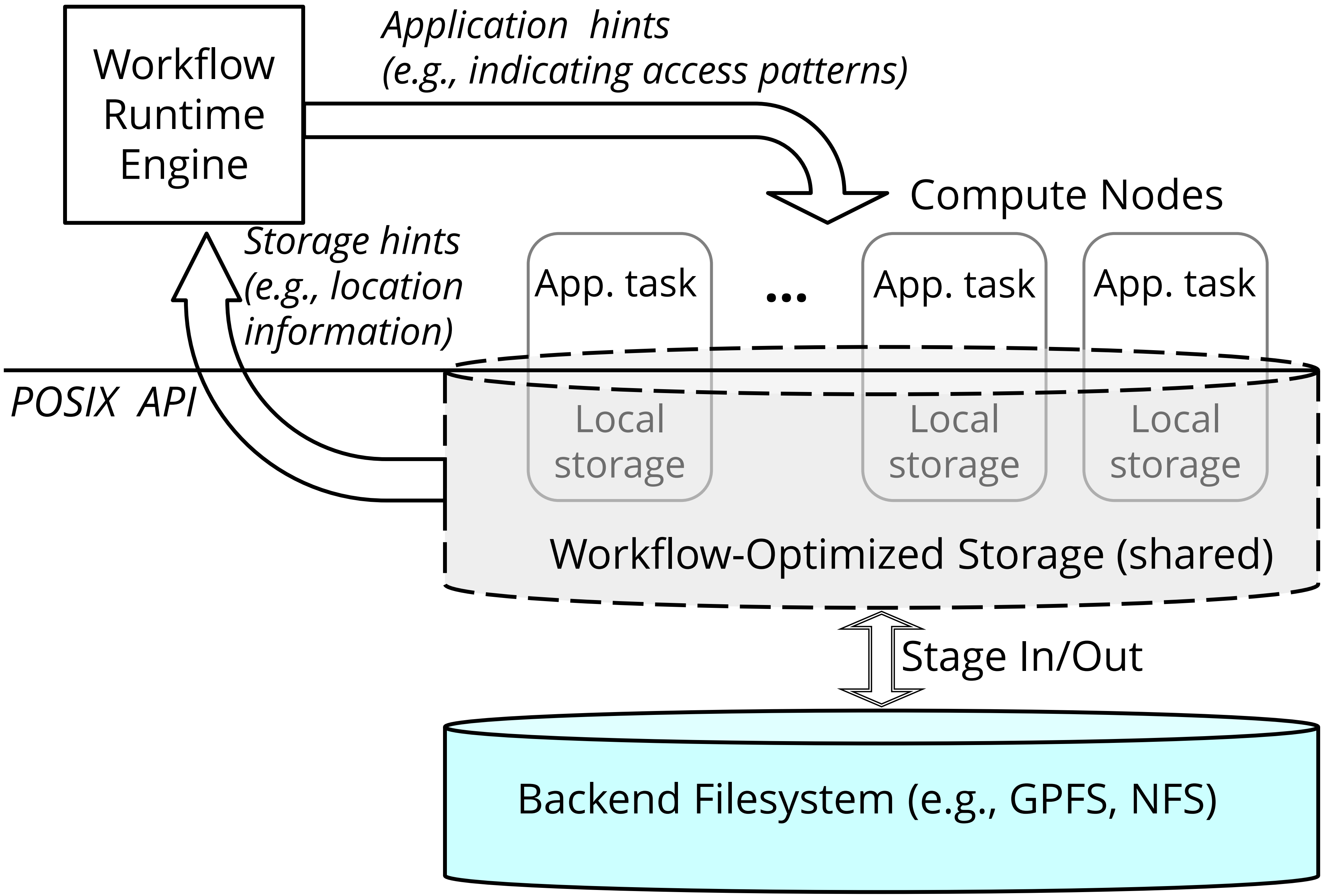
IO rate : 8GBps = 51KBps / core



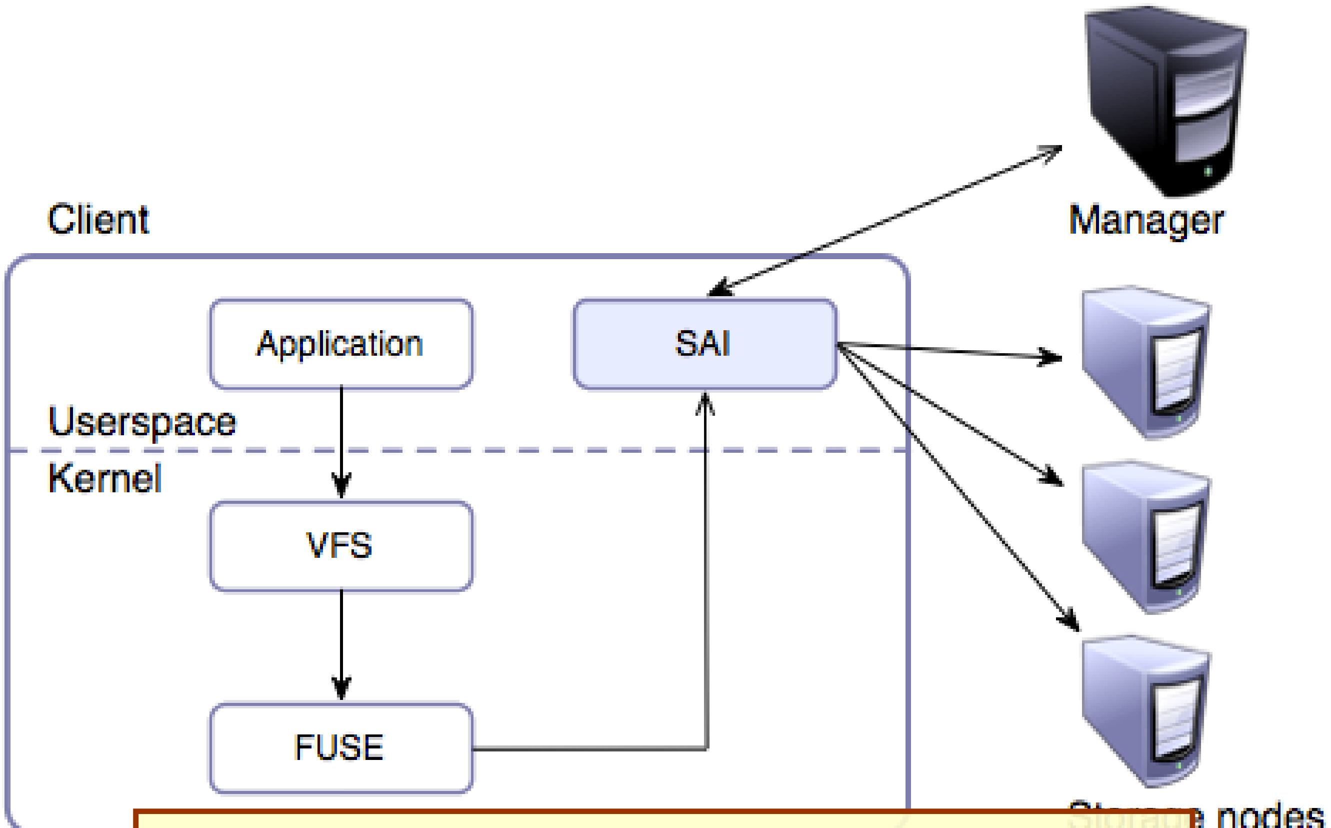
24 servers

Nodes dedicated to an application
Storage system coupled with the application's execution

WOSS Deployment



Execution Path: Client Example



Many components
Network stack gets more complex

Building a Predictor

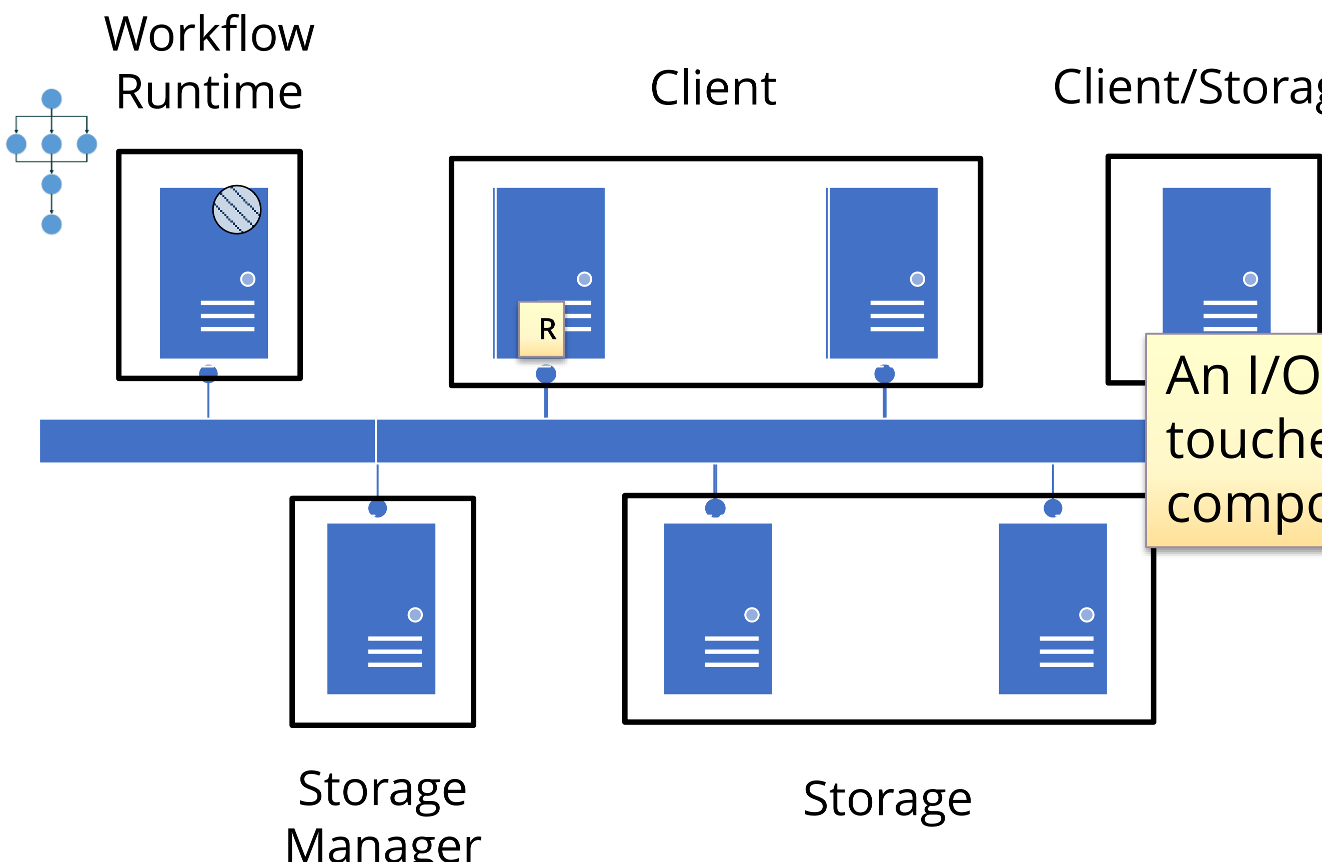
Properties

Fine	**Granularity**	Coarse
Component (Detailed)		Application-Level
White-Box / Less Data	**Seeding**	Black-Box / More Data (Training)
Freedom	**Exploratory**	Training-Coupled
More Information	**Explanatory**	Less Information

Approaches

Simulation	Analytical Models	Machine Learning

System Working



An I/O operation touches several components

There are several tasks in parallel

Modeling: Leveraging the Context

Focus is on application's overall performance

- Per I/O request accuracy is less important

Tasks have distinct phases (read, compute, write)

- Aggregate operations

Tasks' I/O operations have coarse granularity

Thanks for the Pictures

Flight deck - prayitno
http://www.flickr.com/photos/34128007@N04/5292213279/

http://www.flickr.com/photos/twmlabs/282089123/

http://commons.wikimedia.org/wiki/File%3ABalanced_scale_of_Justice.svg

By Perhelion [CC0], via Wikimedia Commons