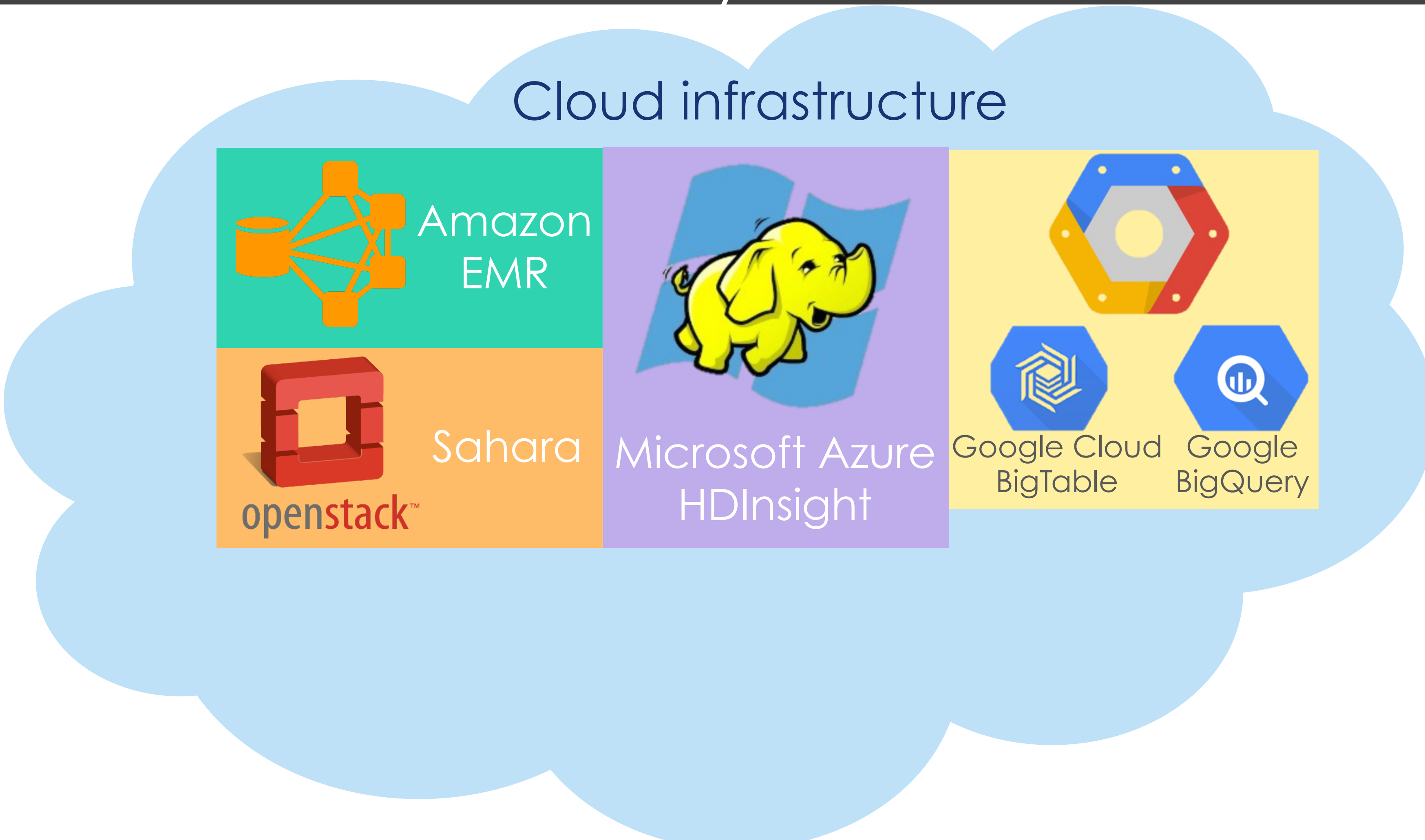
CAST: Tiering Storage for Data Analytics in the Cloud

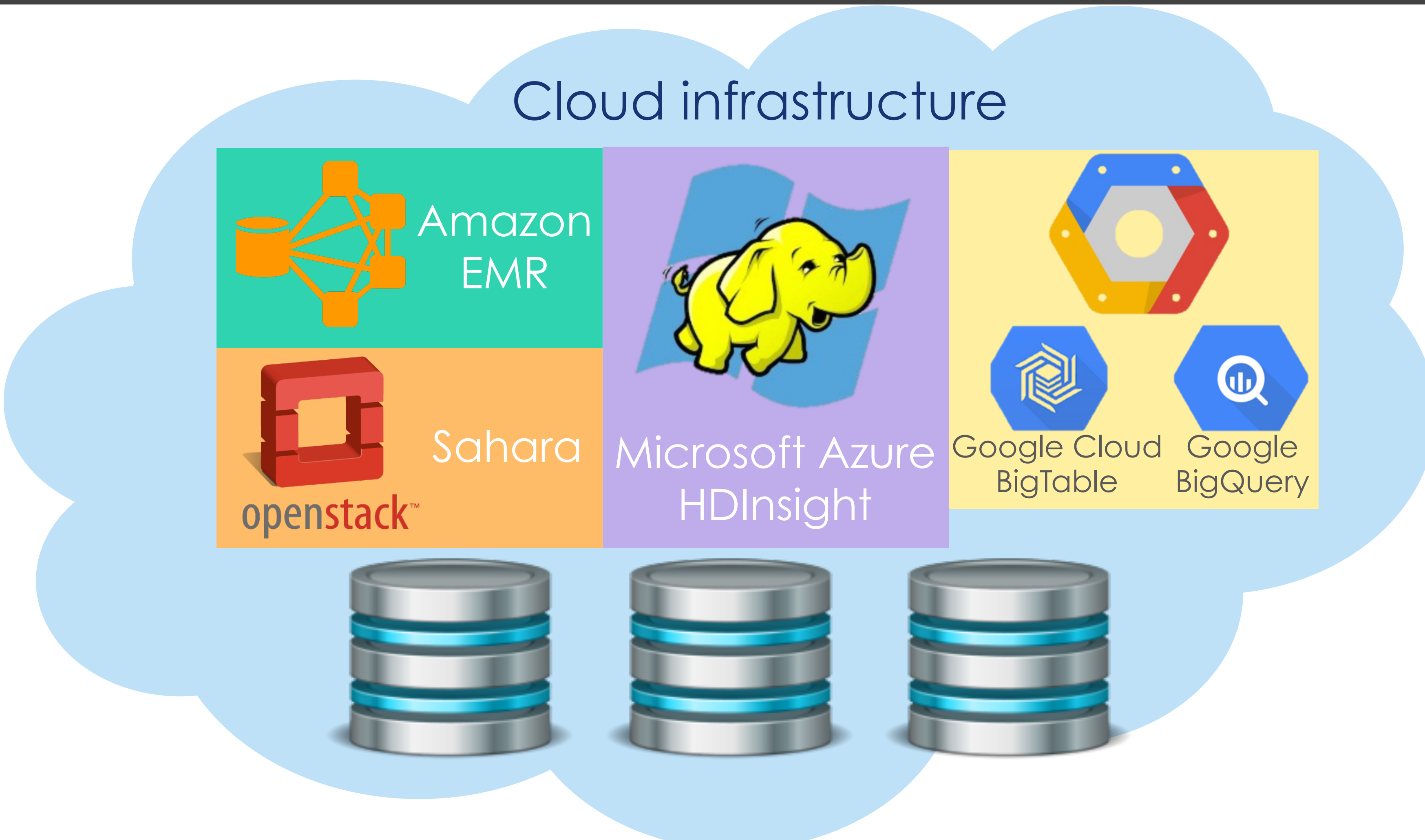
Yue Cheng★, M. Safdar Iqbal★, Aayush Gupta†, Ali R. Butt★

Virginia Tech★, IBM Research – Almaden†

Cloud enables cost-efficient data analytics



Cloud storage enables data analytics in the cloud



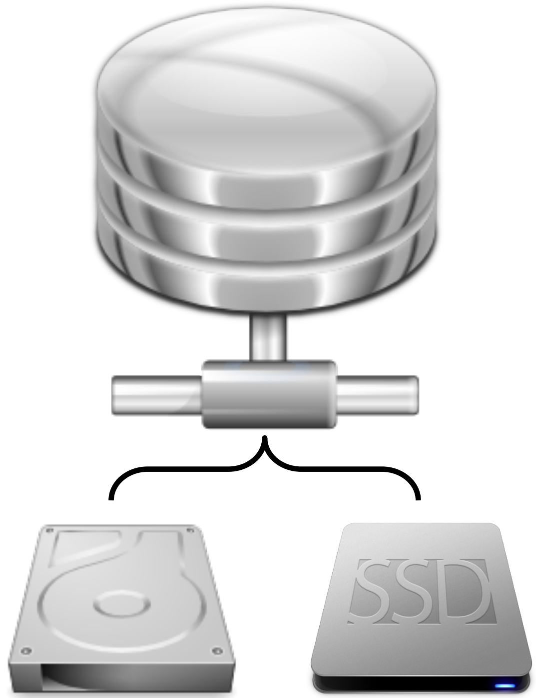
Vast variety of cloud storage services

Vast variety of cloud storage services

Object storage

Vast variety of cloud storage services

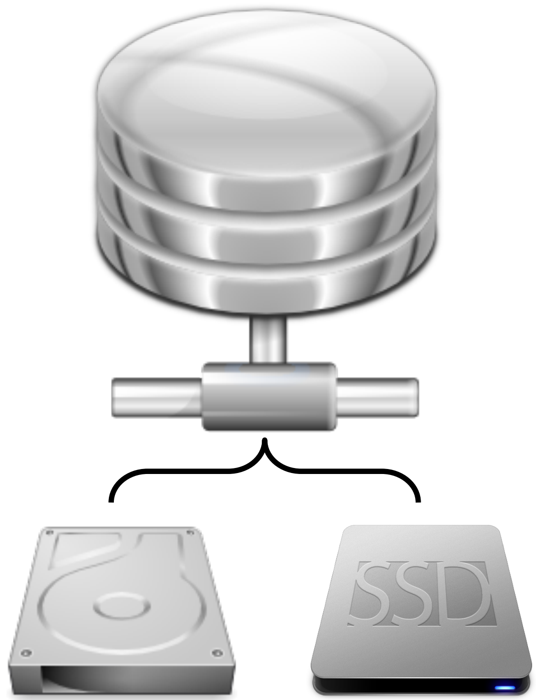
Object storage



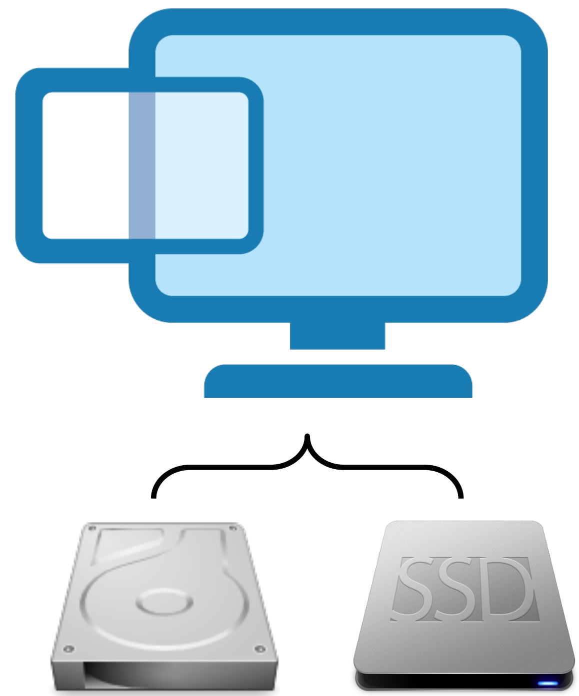
Network-attached
block storage

Vast variety of cloud storage services

Object storage



Network-attached
block storage



VM-local
ephemeral storage

Heterogeneity in cloud storage services

Storage type	Capacity (GB/volume)	Throughput (MB/sec)	IOPS (4KB)	Cost ($/month)
ephSSD	375	733	100000	0.218×375
persSSD	100	48	3000	0.17×100
	250	118	7500	0.17×250
	500	234	15000	0.17×500
persHDD	100	20	150	0.04×100
	250	45	375	0.04×250
	500	97	750	0.04×500
objStore	N/A	265	550	0.026/GB

ephSSD: VM-local ephemeral SSD, **persSSD**: Network-attached persistent SSD, **persHDD**: Network-attached persistent HDD, **objStore**: Google cloud object storage

Heterogeneity in cloud storage services

Storage type	Capacity (GB/volume)	Throughput (MB/sec)	IOPS (4KB)	Cost ($/month)
ephSSD	375	733	100000	0.218×375
persSSD	100	48	3000	0.17×100
	250	118	7500	0.17×250
	500	234	15000	0.17×500
persHDD	100	20	150	0.04×100
	250	45	375	0.04×250
	500	97	750	0.04×500
objStore	N/A	265	550	0.026/GB

ephSSD offers best performance w/o data persistence.

Heterogeneity in cloud storage services

Storage type	Capacity (GB/volume)	Throughput (MB/sec)	IOPS (4KB)	Cost ($/month)
ephSSD	375	733	100000	0.218×375
persSSD	**100**	**48**	3000	0.17×100
	250	**118**	7500	0.17×250
	500	**234**	15000	0.17×500
persHDD	100	20	150	0.04×100
	250	45	375	0.04×250
	500	97	750	0.04×500
objStore	N/A	265	550	0.026/GB

Performance of the network-attached block storage depends on the size of the volume.

Heterogeneity in cloud storage services

Storage type	Capacity (GB/volume)	Throughput (MB/sec)	IOPS (4KB)	Cost ($/month)
ephSSD	375	733	100000	0.218×375
persSSD	100	48	3000	0.17×100
	250	118	7500	0.17×250
	500	**234**	15000	**0.17**×500
persHDD	100	20	150	0.04×100
	250	45	375	0.04×250
	500	97	750	0.04×500
objStore	N/A	**265**	550	**0.026**/GB

objStore provides the cheapest service and offers comparable sequential throughput compared to that of a 500GB persSSD.

Heterogeneity in data analytics jobs

Application	I/O-intensive			CPU-intensive
	Map	Shuffle	Reduce	
Sort	✗	✔	✗	✗
Join	✗	✔	✔	✗
Grep	✔	✗	✗	✗
KMeans	✗	✗	✗	✔

Decision paralysis

Highly heterogeneous cloud storage services

Highly heterogeneous analytics workloads

Cloud tenant

How do I get the **MOST BANG-for-the-buck? $$**

Motivation

- A need for a comprehensive experimental analysis
  - To study the analytics-job to cloud-storage relationships
- How to exploit heterogeneity in cloud storage and analytics workloads
  - To reduce $ cost
  - To improve performance
  - To meet the deadline

Outline

~~Motivation~~

Quantitative analysis

CAST design

Evaluation

Outline

~~Motivation~~

Quantitative analysis

CAST design

Evaluation

Experimental study methodology

Application	I/O-intensive			CPU-intensive
	Map	Shuffle	Reduce	
Sort	✗	✔	✗	✗
Join	✗	✔	✔	✗
Grep	✔	✗	✗	✗
KMeans	✗	✗	✗	✔

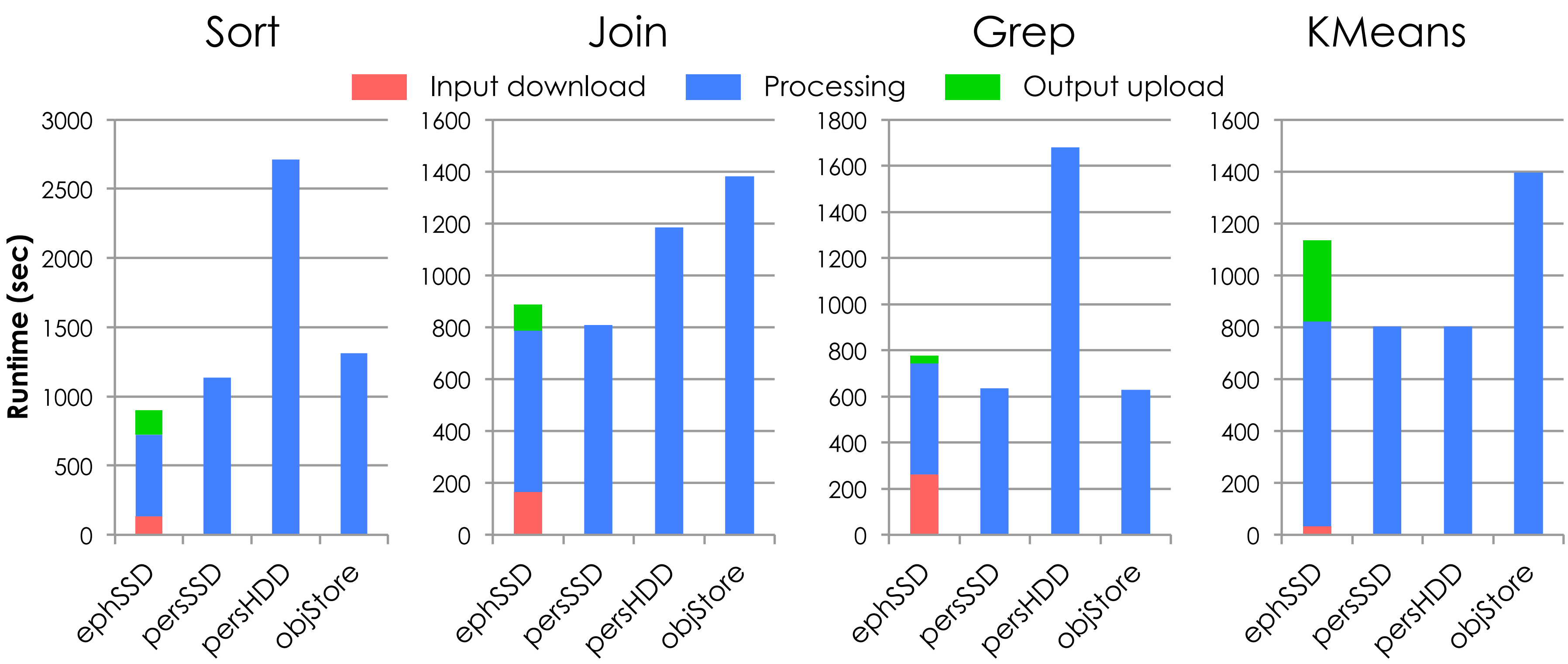
- Experiments on Google Cloud
  - One n1-standard-16 VM (16 vCPUs, 60GB RAM)

Experimental study methodology

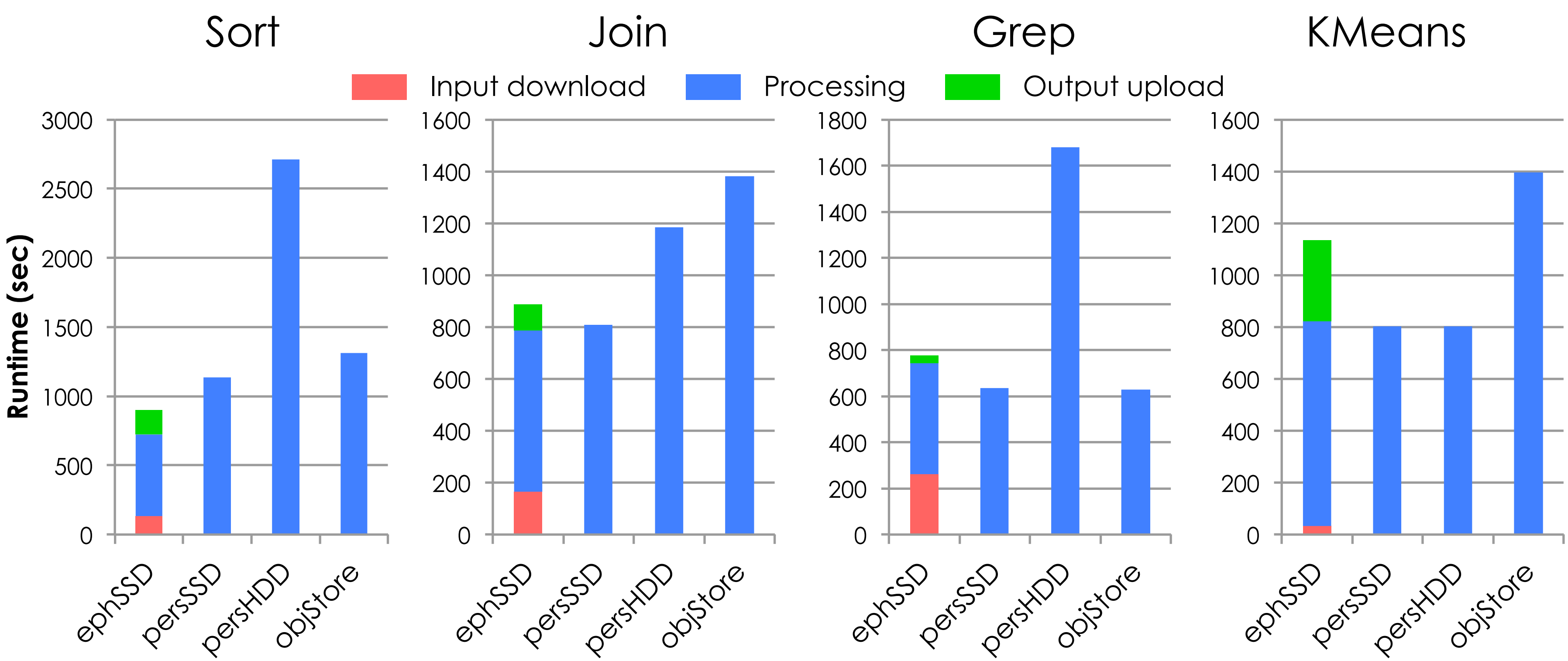
Application	I/O-intensive			CPU-intensive
	Map	Shuffle	Reduce	
Sort	✗	✔	✗	✗
Join	✗	✔	✔	✗
Grep	✔	✗	✗	✗
KMeans	✗	✗	✗	✔

- Experiments on Google Cloud
  - One n1-standard-16 VM (16 vCPUs, 60GB RAM)
- Application granularity
- Workload granularity

Application granularity: Performance

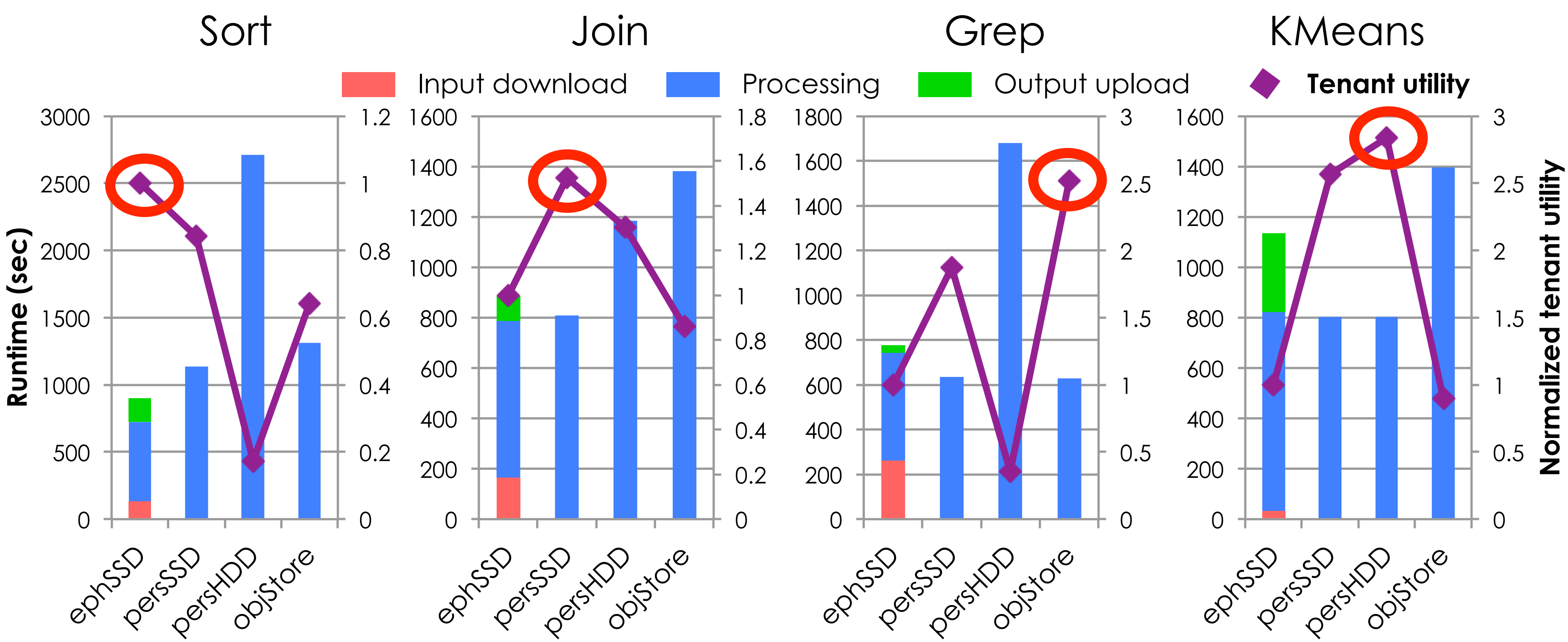


Application granularity: Performance



No storage service provides the best raw performance

Application granularity: Tenant utility

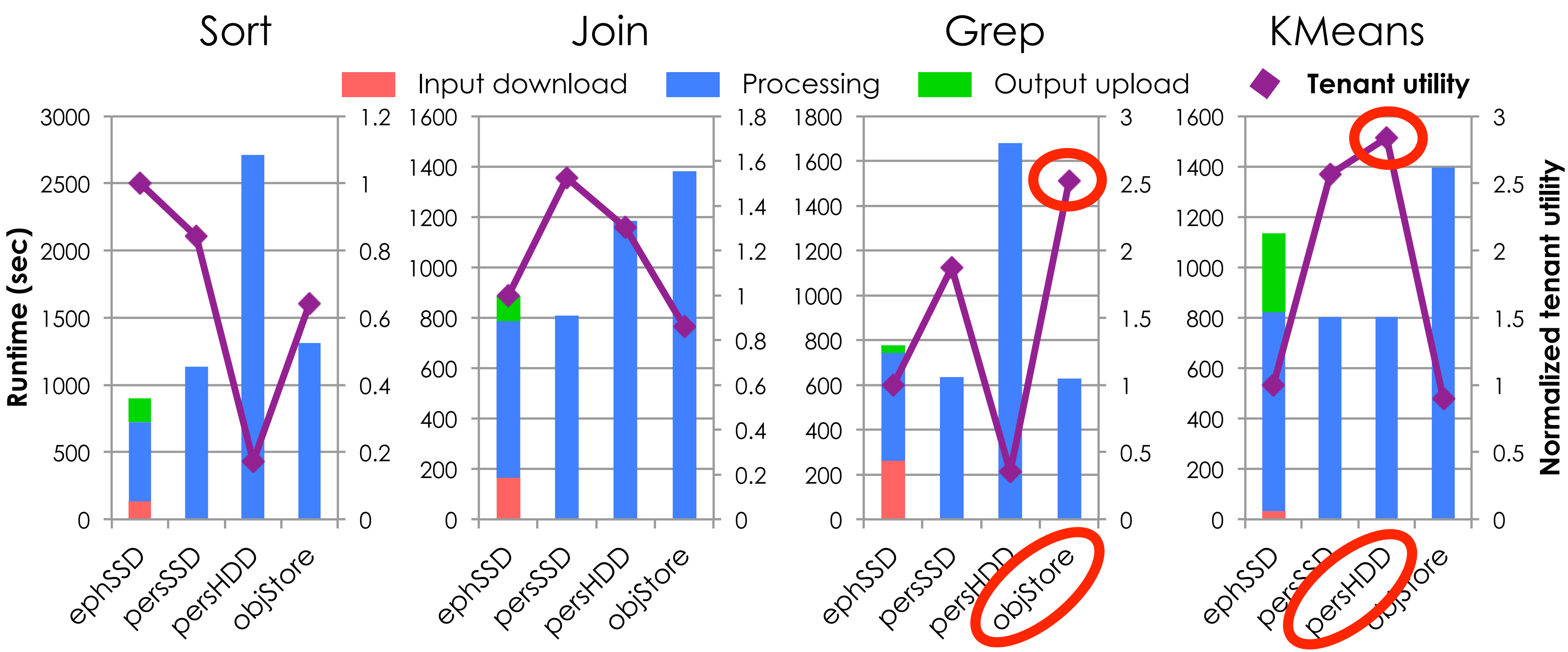


$$\text{Tenant utility} = \frac{1/T}{\$}$$

← Performance

← $ cost

Application granularity: Tenant utility



Slower storage, in some case, may provide higher utility & comparable performance

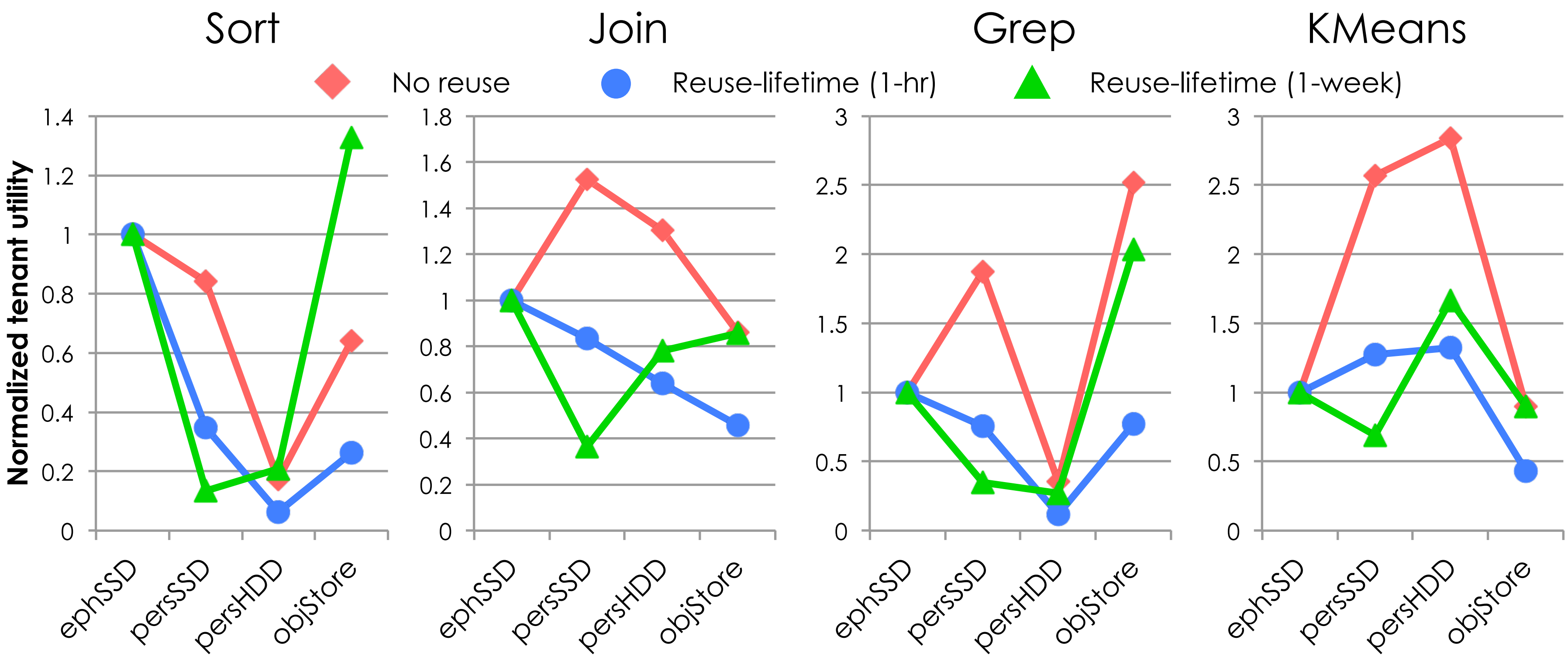
Workload granularity: Data reuse

$$\text{Tenant utility} = \frac{1/T}{\$}$$

← Performance

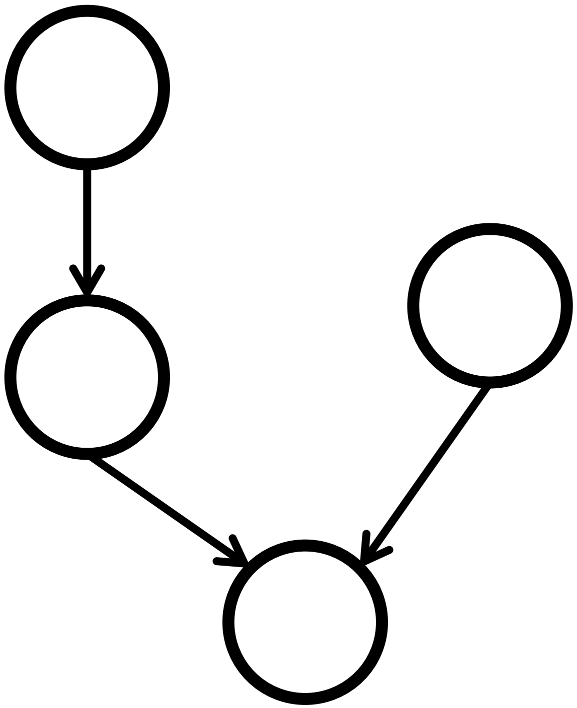
← $ cost

Workload granularity: Data reuse

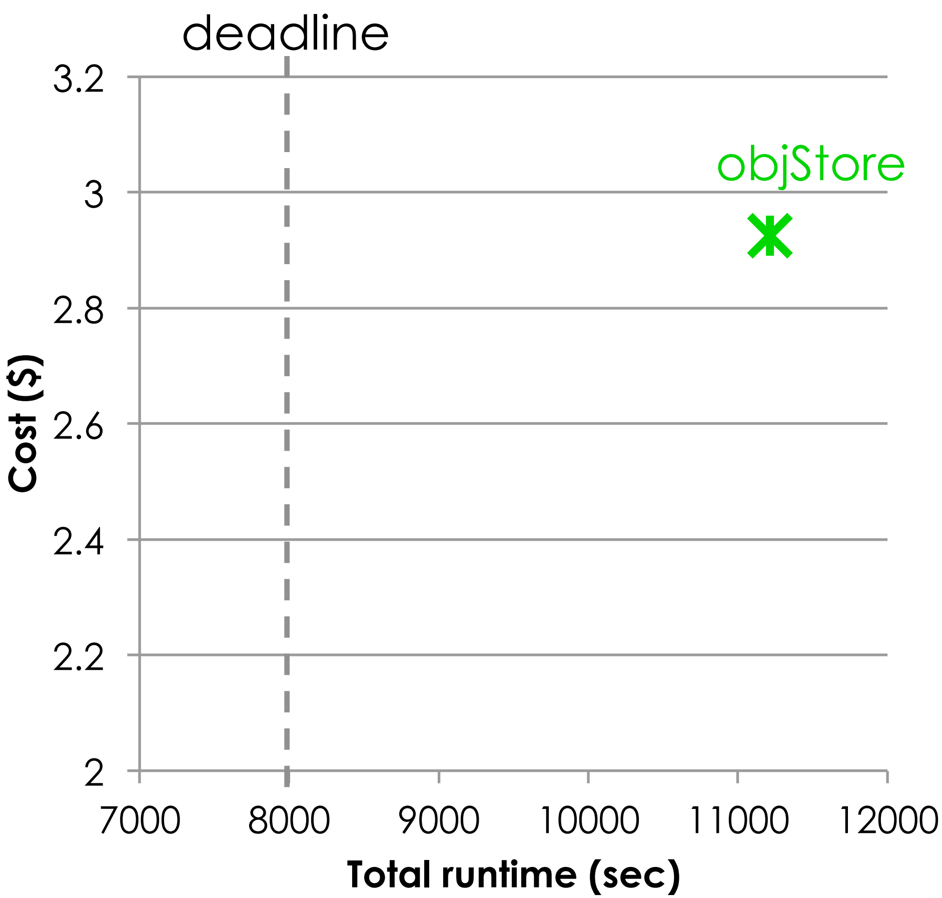


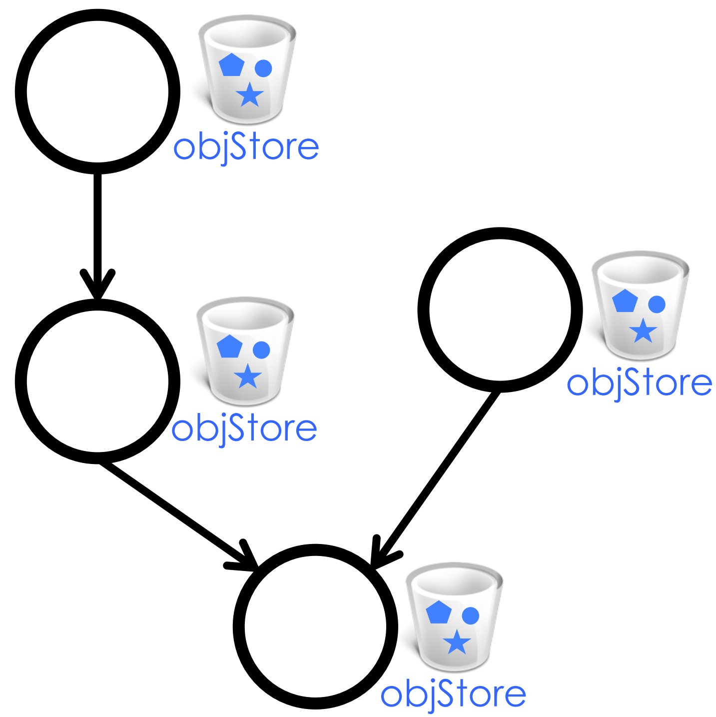
Data reuse patterns affect data placement choices

Workload granularity: Inter dependency

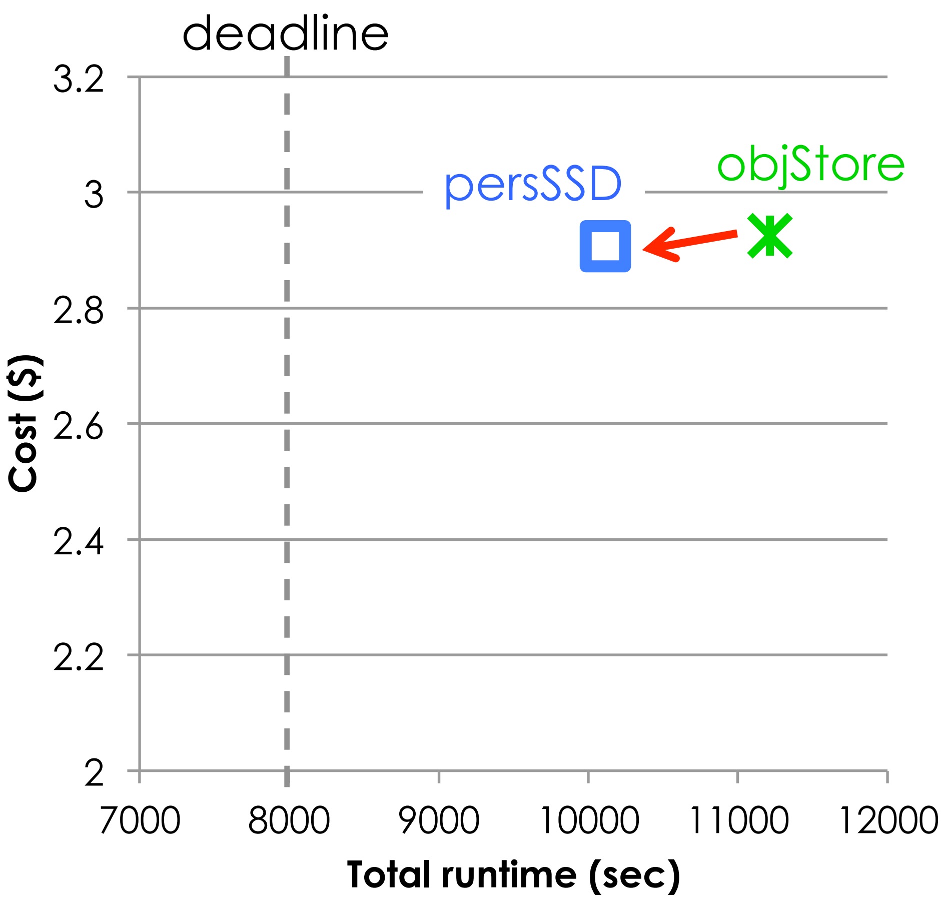


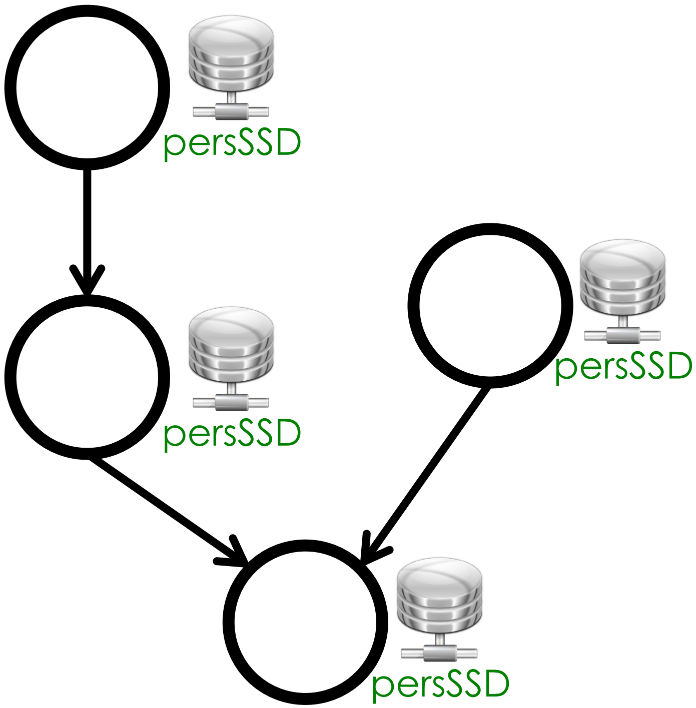
Workload granularity: Inter dependency



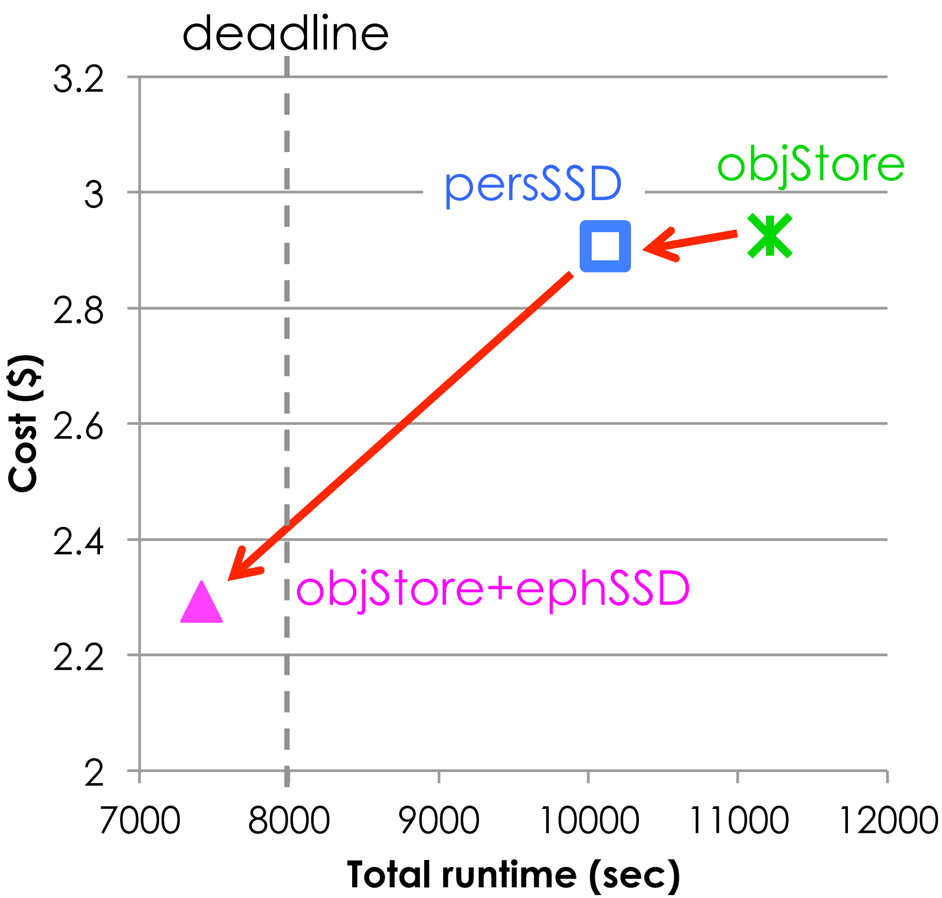


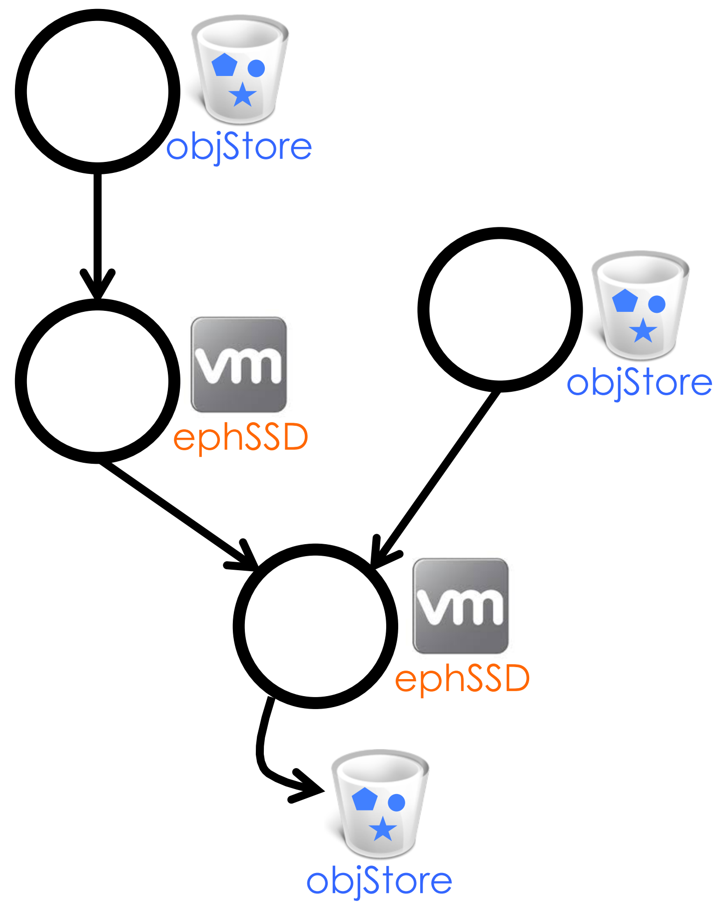
Workload granularity: Inter dependency



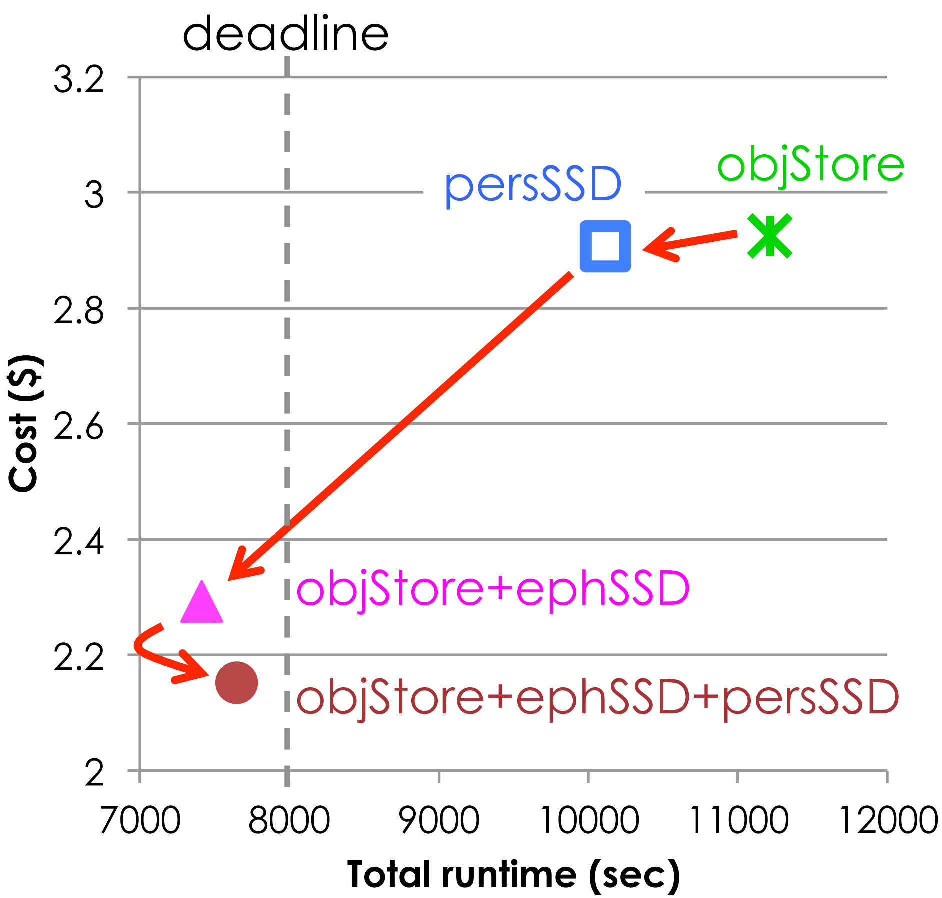


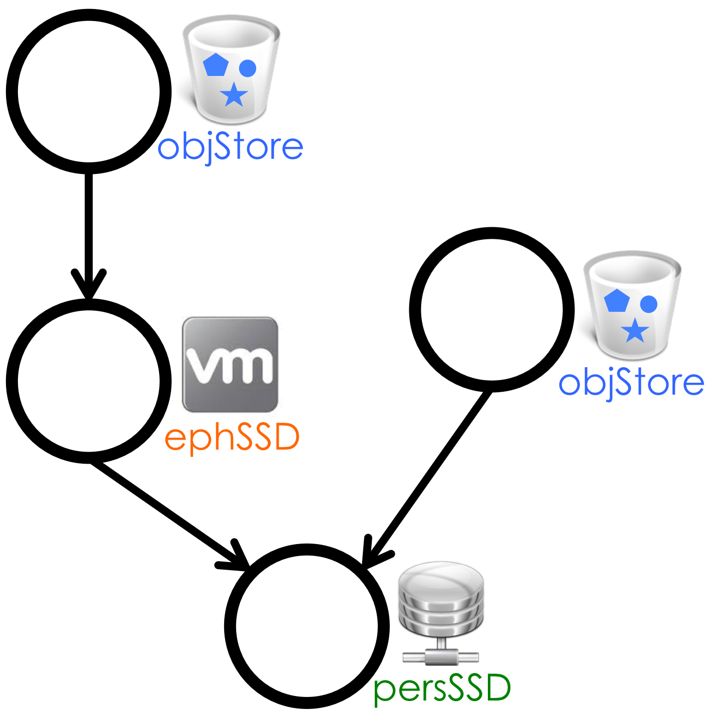
Workload granularity: Inter dependency



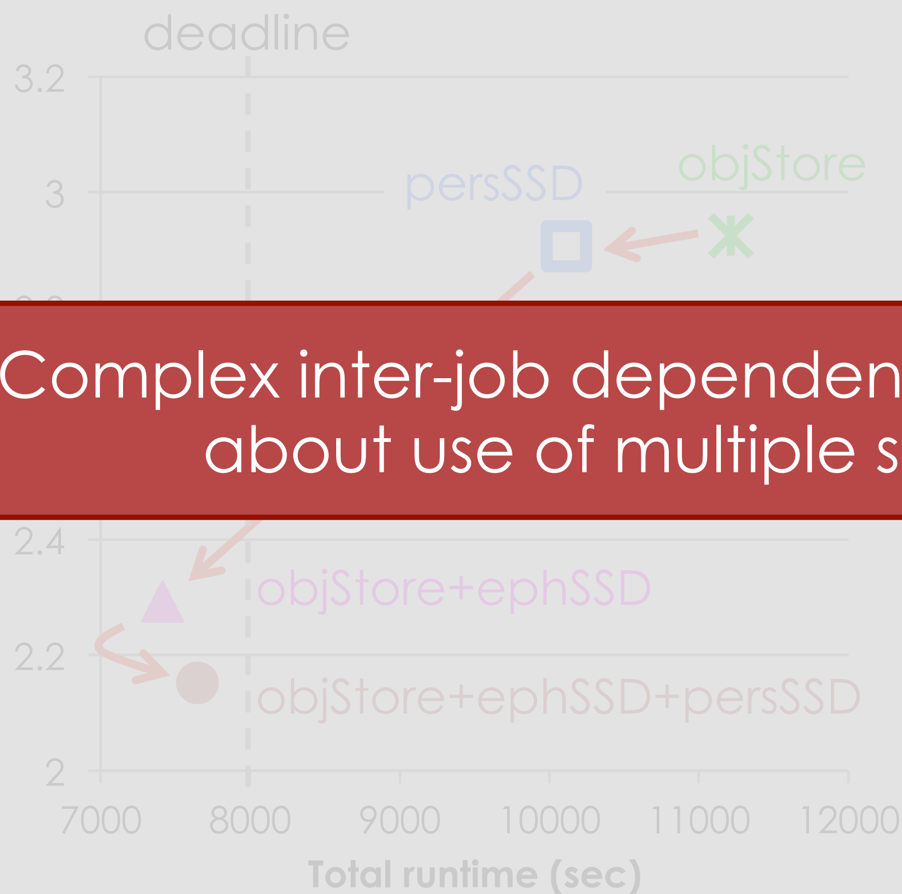


Workload granularity: Inter dependency





Workload granularity: Inter dependency



Complex inter-job dependencies require rethinking about use of multiple storage services

CAST: Cloud Analytics Storage Tiering

…

CAST —exploits→ **Different cloud storage services**

CAST —exploits→ **Heterogeneity in analytics workloads**

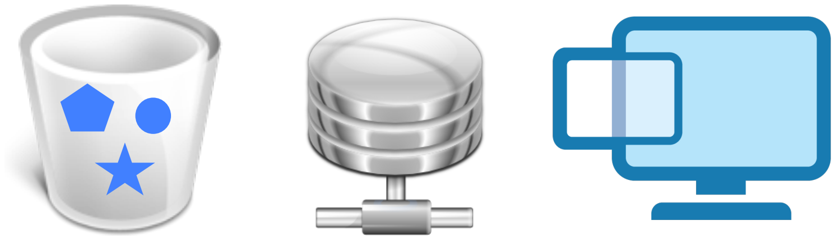
Different application characteristics

Inter-job dependency

Data reuse across jobs

…

CAST: Cloud Analytics Storage Tiering



…

CAST

exploits

Different cloud storage services

exploits

Different application characteristics

…

Inter-job dependency

Data reuse across jobs

Heterogeneity in analytics workloads

Outline

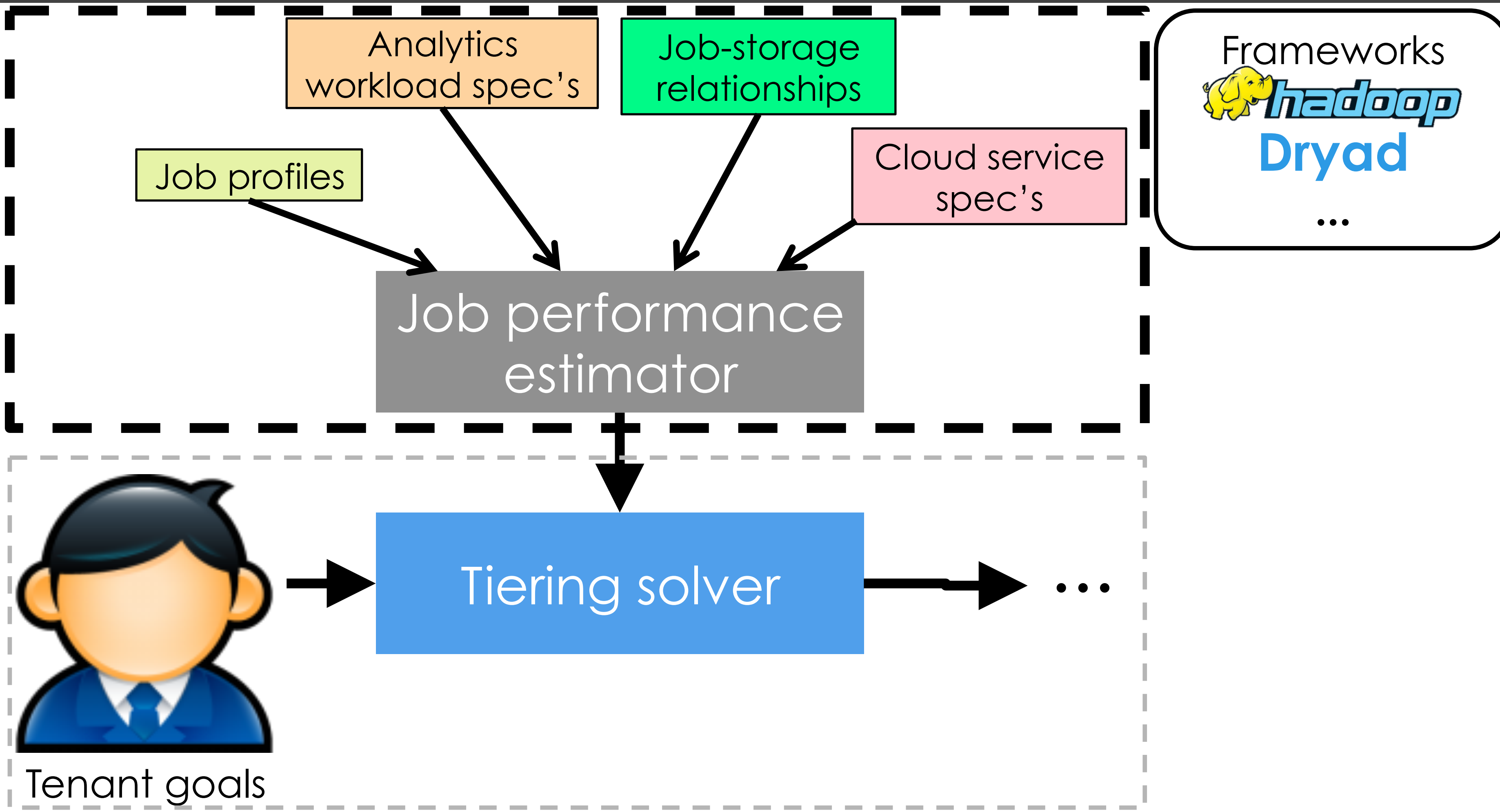
~~Motivation~~

~~Quantitative analysis~~

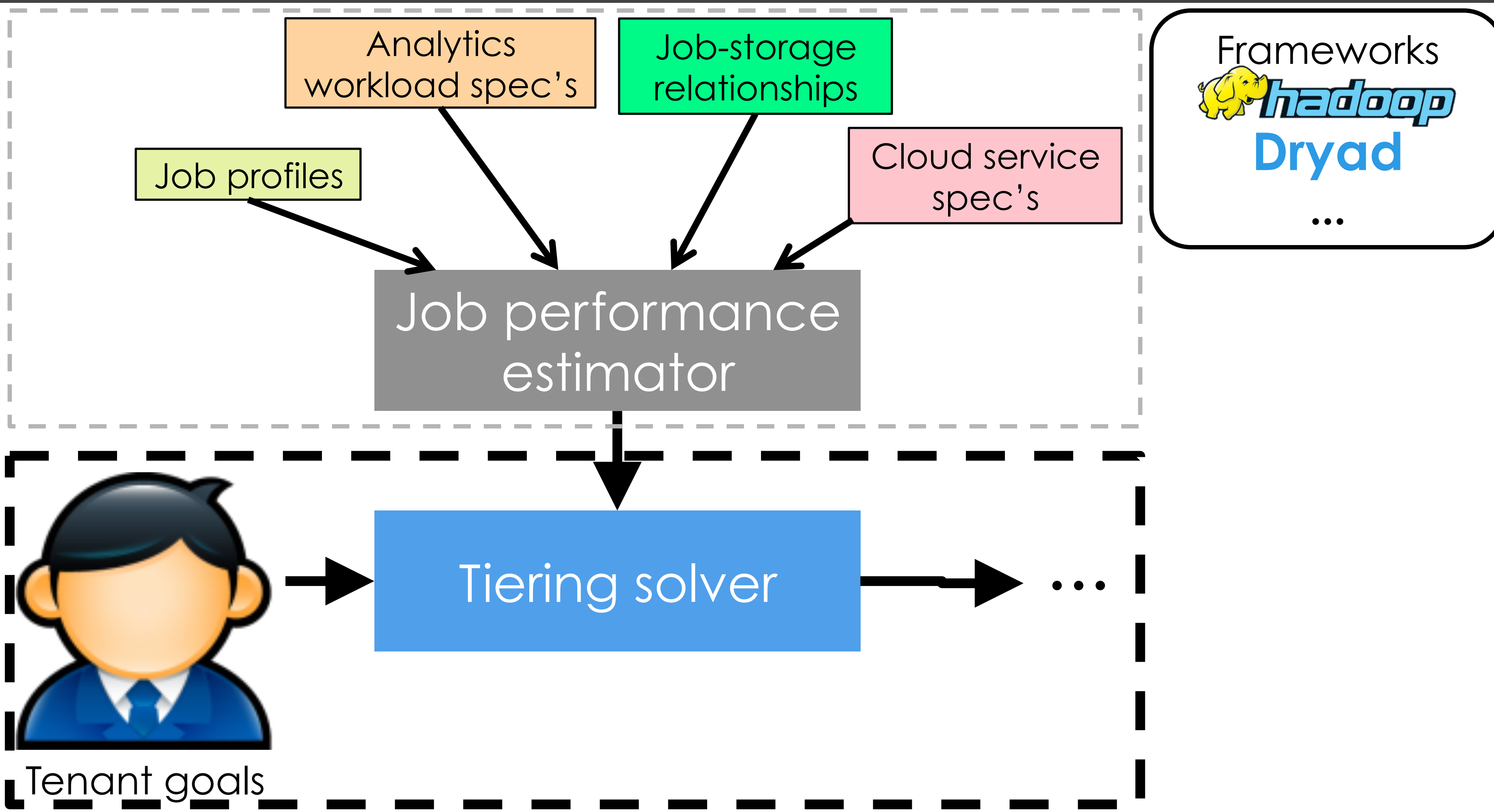
CAST design

Evaluation

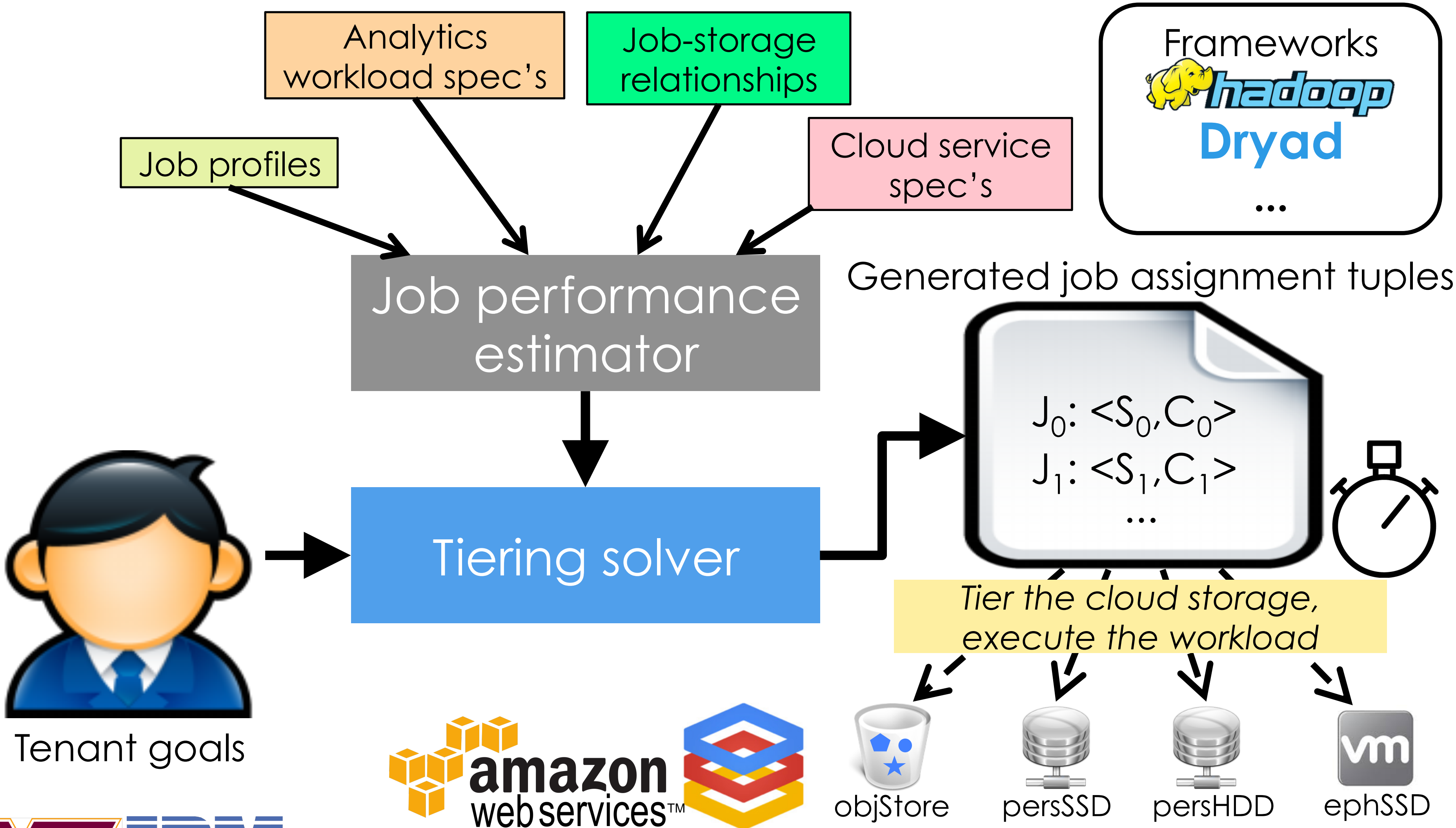
CAST framework



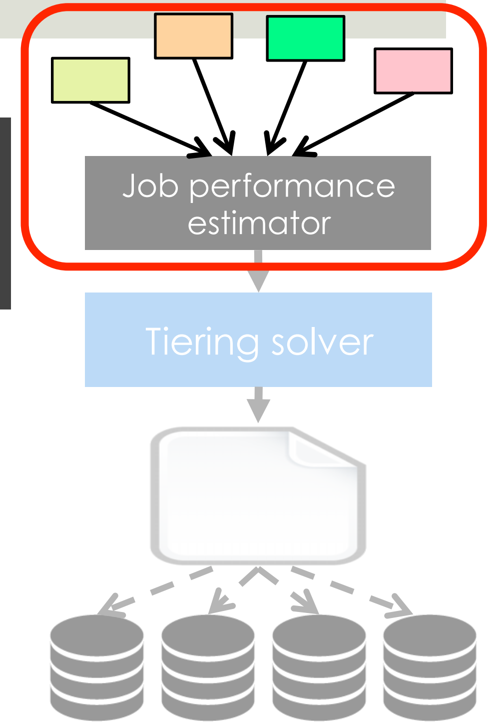
CAST framework



CAST framework



Job performance estimator



$$EST\left(\hat{R}, \hat{M}(s_i, L_i)\right) = \left\lceil \frac{m}{n_{vm} \cdot m_c} \right\rceil \cdot \left(\frac{input_i / m}{bw_{map}^{s_i}} \right) + \left\lceil \frac{r}{n_{vm} \cdot r_c} \right\rceil \cdot \left(\frac{inter_i / r}{bw_{shuffle}^{s_i}} \right)$$

$$+ \left\lceil \frac{r}{n_{vm} \cdot r_c} \right\rceil \cdot \left(\frac{ouput_i / r}{bw_{reduce}^{s_i}} \right)$$

Map phase: # waves · Runtime/wave

Shuffle phase

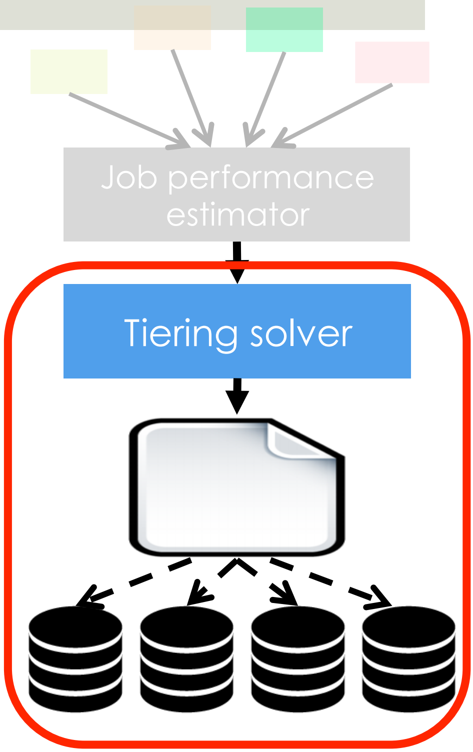
Reduce phase

* MRCute: Bazaar [SoCC'12]

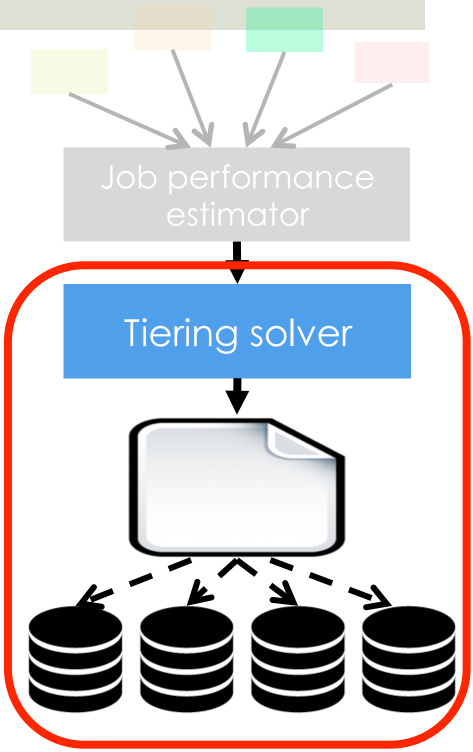
Tiering solver

- Optimization
  - Objective function

$$\textbf{max} \quad \text{Tenant utility} = \frac{1/T}{(\$_{vm} + \$_{store})}$$



Tiering solver



- Optimization
  - Objective function

$$\textbf{max} \quad \text{Tenant utility} = \frac{1/T}{(\$_{vm} + \$_{store})}$$

  - Constraints

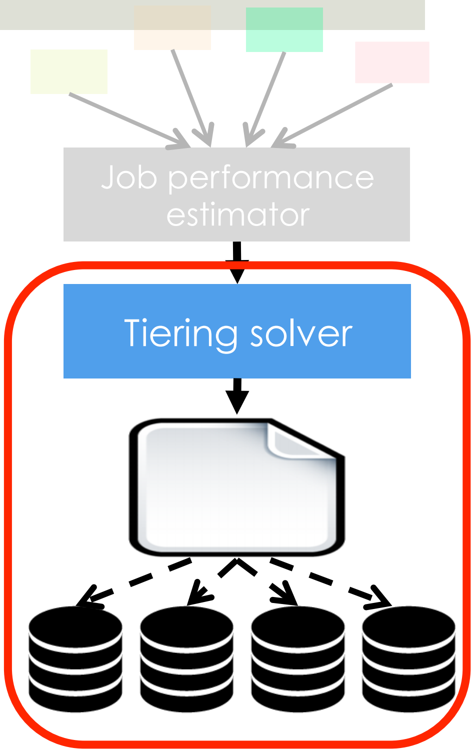
$$c_i \geq (I_i + M_i + O_i) \qquad (\forall i \in J)$$

$$T = \sum_{i=1}^{J} REG\left(s_i, C[s_i], \hat{R}, \hat{L}_i\right), \text{ where } s_i \in F$$

$$\$_{vm} = n_{vm} \cdot (P_{vm} \cdot T)$$

$$\$_{store} = \sum_{f}^{F} \left(C[f] \cdot (P_{store}[f] \cdot \left\lceil T/60 \right\rceil) \right)$$

Tiering solver



- Optimization
  - Objective function

$$\textbf{max} \quad \text{Tenant utility} = \frac{1/T}{(\$_{vm} + \$_{store})}$$

  - Constraints

$$c_i \geq (I_i + M_i + O_i) \qquad (\forall i \in J)$$

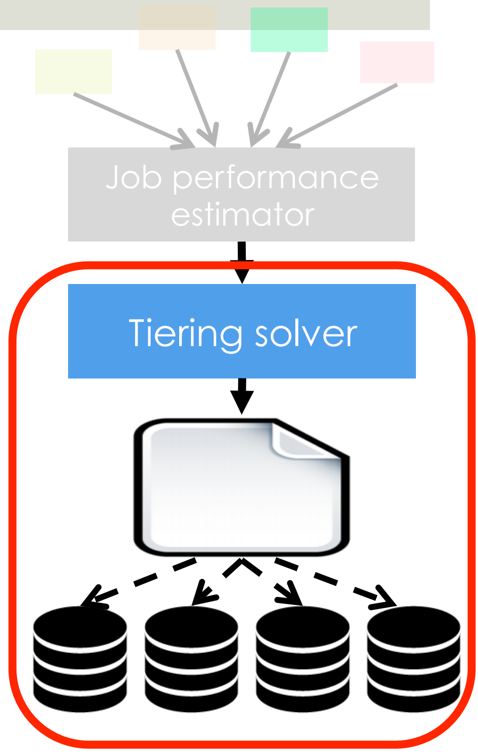
Space capacity constraint

$$T = \sum_{i=1}^{J} REG\left(s_i, C[s_i], \hat{R}, \hat{L}_i\right), \text{ where } s_i \in F$$

$$\$_{vm} = n_{vm} \cdot (P_{vm} \cdot T)$$

$$\$_{store} = \sum_{f}^{F} \left(C[f] \cdot (P_{store}[f] \cdot \left\lceil T/60 \right\rceil) \right)$$

Tiering solver



- Optimization
  - Objective function

$$\textbf{max} \quad \text{Tenant utility} = \frac{1/T}{(\$_{vm} + \$_{store})}$$

  - Constraints

$$c_i \geq (I_i + M_i + O_i) \qquad (\forall i \in J)$$

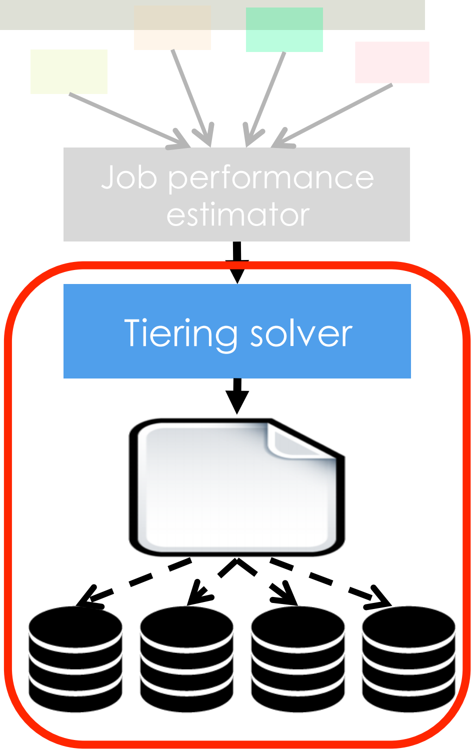
$$T = \sum_{i=1}^{J} REG\left(s_i, C[s_i], \hat{R}, \hat{L}_i\right), \text{ where } s_i \in F$$

Total workload runtime

$$\$_{vm} = n_{vm} \cdot (P_{vm} \cdot T)$$

$$\$_{store} = \sum_{f}^{F} \left(C[f] \cdot (P_{store}[f] \cdot \lceil T/60 \rceil) \right)$$

Tiering solver



- Optimization
  - Objective function

$$\textbf{max} \quad \text{Tenant utility} = \frac{1/T}{(\$_{vm} + \$_{store})}$$

  - Constraints

$$c_i \geq (I_i + M_i + O_i) \qquad (\forall i \in J)$$

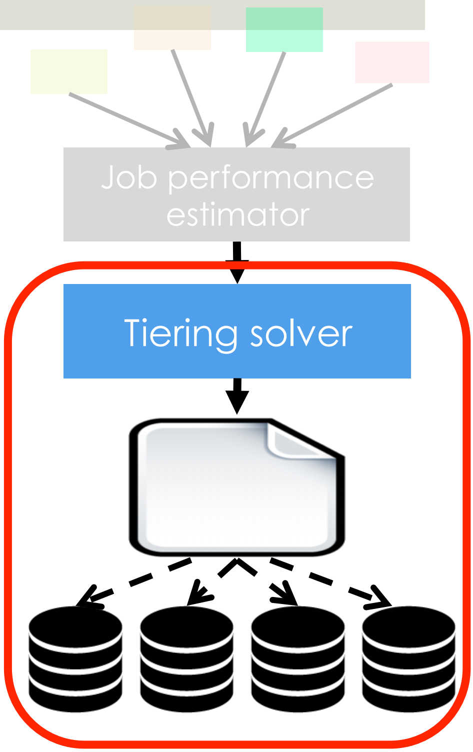
$$T = \sum_{i=1}^{J} REG\left(s_i, C[s_i], \hat{R}, \hat{L}_i\right), \text{ where } s_i \in F$$

$$\$_{vm} = n_{vm} \cdot (P_{vm} \cdot T)$$

VM \$ cost

$$\$_{store} = \sum_{f}^{F} \left(C[f] \cdot (P_{store}[f] \cdot \left\lceil T/60 \right\rceil) \right)$$

Tiering solver



- Optimization
  - Objective function

$$\textbf{max} \quad \text{Tenant utility} = \frac{1/T}{(\$_{vm} + \$_{store})}$$

  - Constraints

$$c_i \geq (I_i + M_i + O_i) \qquad (\forall i \in J)$$

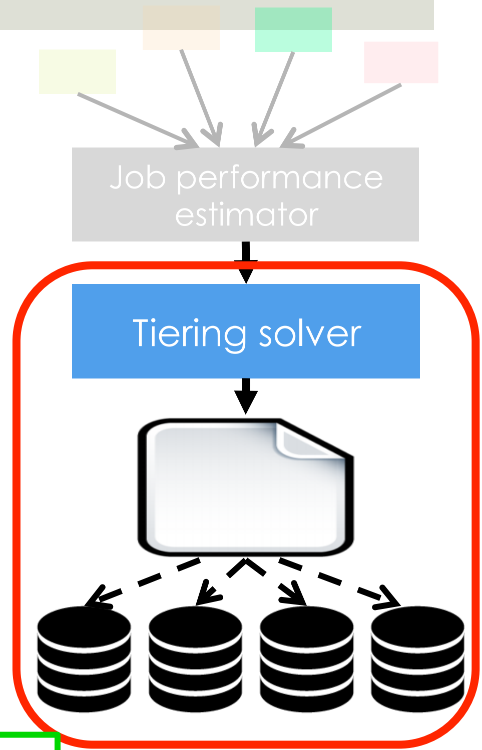
$$T = \sum_{i=1}^{J} REG\left(s_i, C[s_i], \hat{R}, \hat{L}_i\right), \text{ where } s_i \in F$$

$$\$_{vm} = n_{vm} \cdot (P_{vm} \cdot T)$$

$$\$_{store} = \sum_{f}^{F} \left(C[f] \cdot (P_{store}[f] \cdot \left\lceil T/60 \right\rceil) \right)$$

Storage $ cost

Tiering solver



- Optimization
  - Objective function

$$\textbf{max}\quad \text{Tenant utility} = \frac{1/T}{(\$_{vm} + \$_{store})}$$

  - Constraints

Tuning knob: Capacity of J_i

$$c_i \geq (I_i + M_i + O_i) \qquad (\forall i \in J)$$

Tuning knob: Storage service of J_i

$$T = \sum_{i=1}^{J} REG\left(s_i, C[s_i], \hat{R}, \hat{L}_i\right), \text{ where } s_i \in F$$

$$\$_{vm} = n_{vm} \cdot (P_{vm} \cdot T)$$

$$\$_{store} = \sum_{f}^{F} \left(C[f] \cdot (P_{store}[f] \cdot \left\lceil T/60 \right\rceil) \right)$$

Simulated annealing

J_0: <s_0,c_0>
J_1: <s_1,c_1>
J_2: <s_2,c_2>
…
Assigned job storage, adjusted storage capacity

Enhancements: CAST++

- **Enhancement 1:** Data reuse awareness
  - All jobs sharing the same dataset have the same storage service assigned to them
- **Enhancement 2:** Workflow awareness
  - Objective

    $$\textbf{min} \quad \$_{total} = \$_{vm} + \$_{store}$$
  - Constraints

    $$T \le deadline$$

    Depth-first traversal in workflow DAG for allocating storage capacities

Outline

~~Motivation~~

~~Quantitative analysis~~

~~CAST design~~

Evaluation

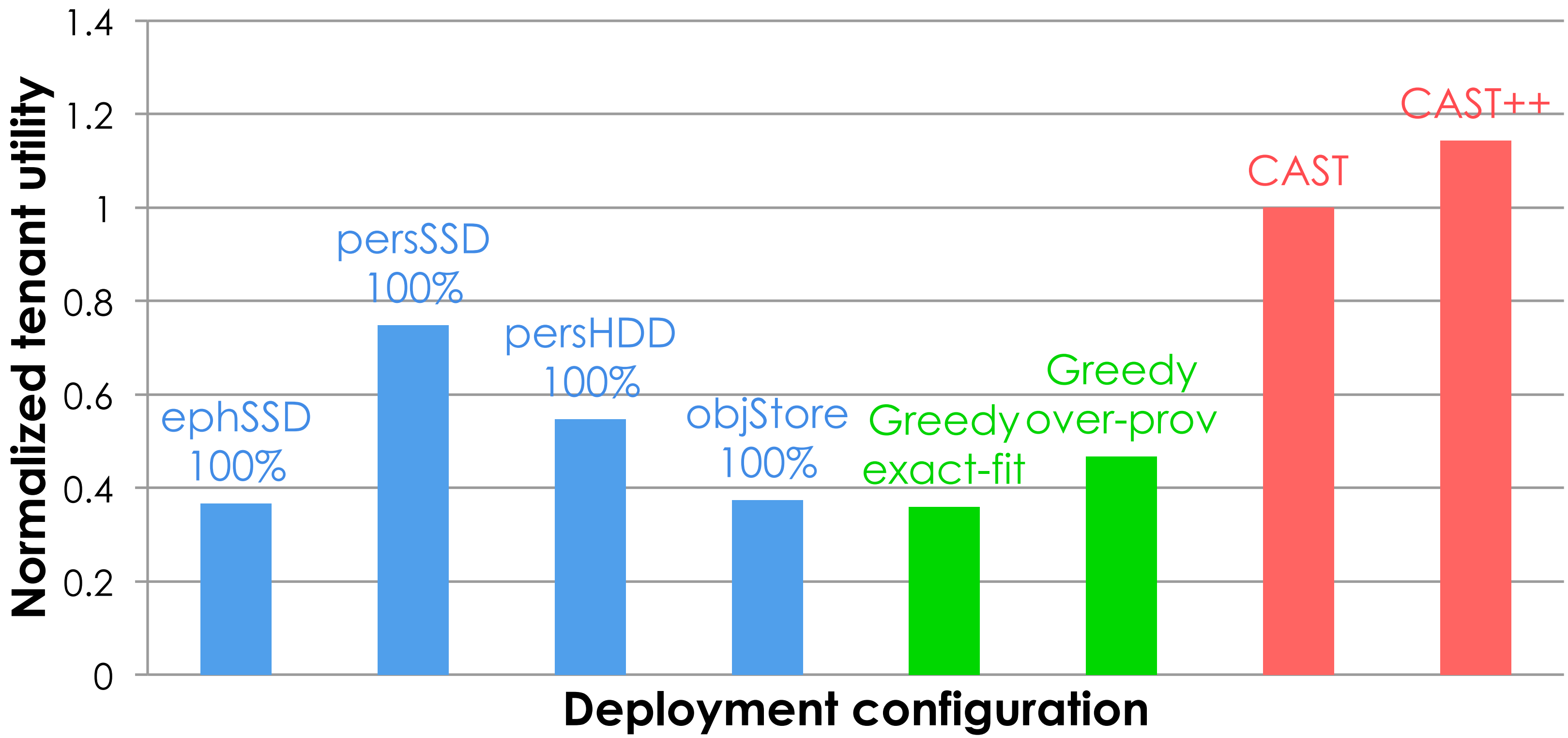
Methodology

- **400-core** Hadoop cluster in **Google Cloud**
  - 25 **n1-standard-16 VM** (16 vCPUs, 60GB RAM)

- Tenant utility measurement
  - CAST: Effectiveness for general workloads
  - CAST++: Effectiveness for data reuse

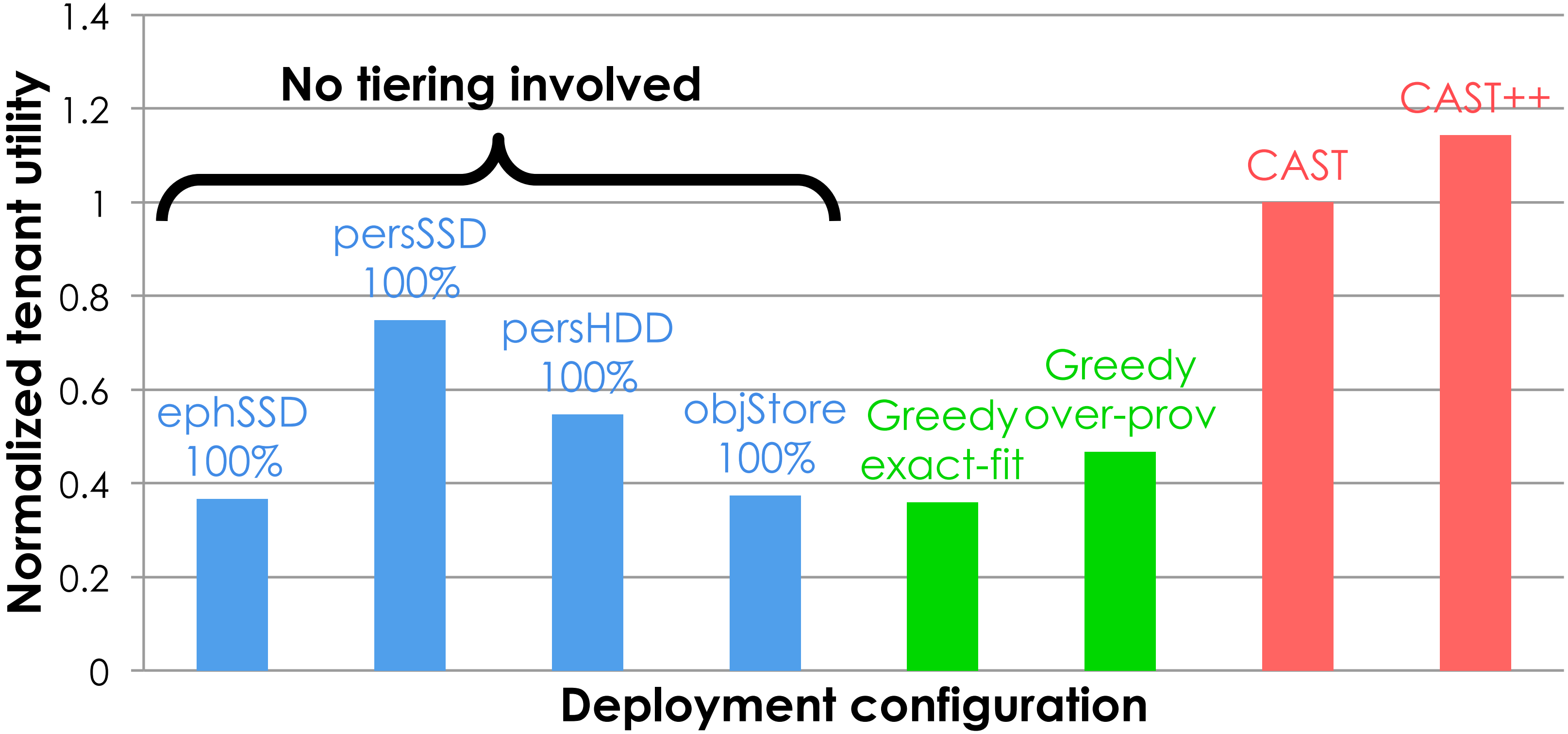
- Meeting workflow deadlines with CAST++

Tenant utility improvement



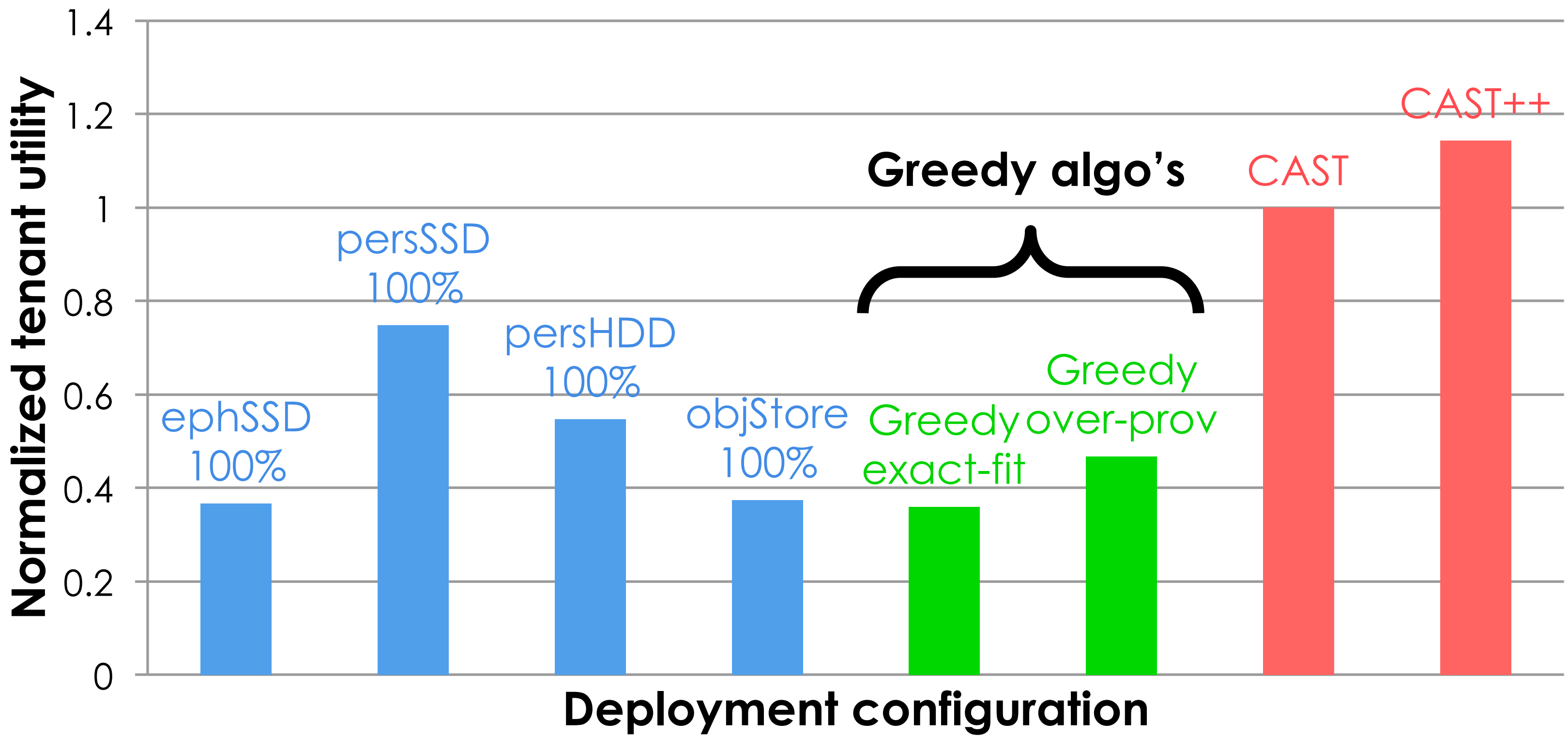
100-job Hadoop workload, simulating behaviors of Facebook's 3000-machine Hadoop cluster

Tenant utility improvement



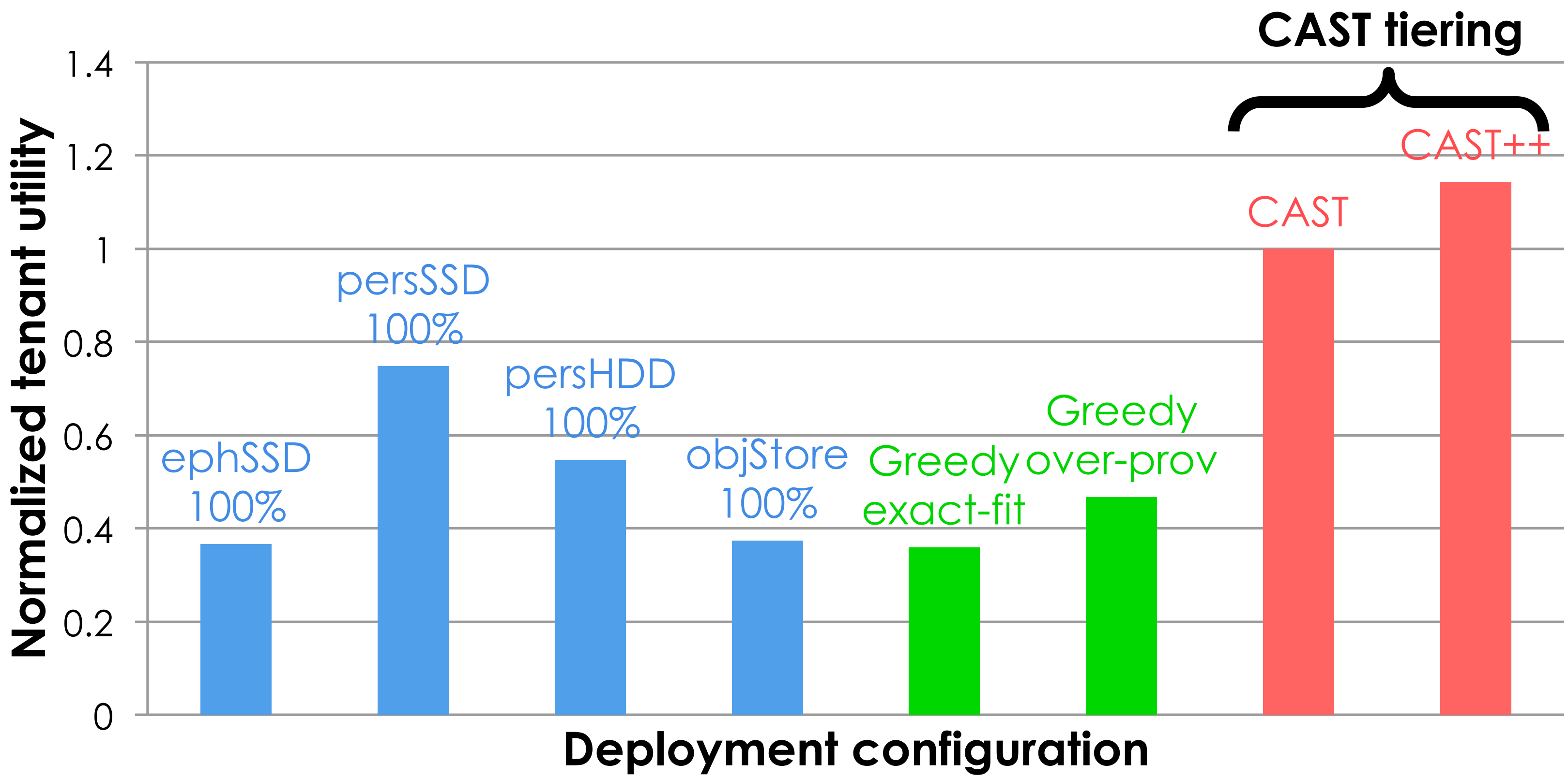
100-job Hadoop workload, simulating behaviors of Facebook's 3000-machine Hadoop cluster

Tenant utility improvement



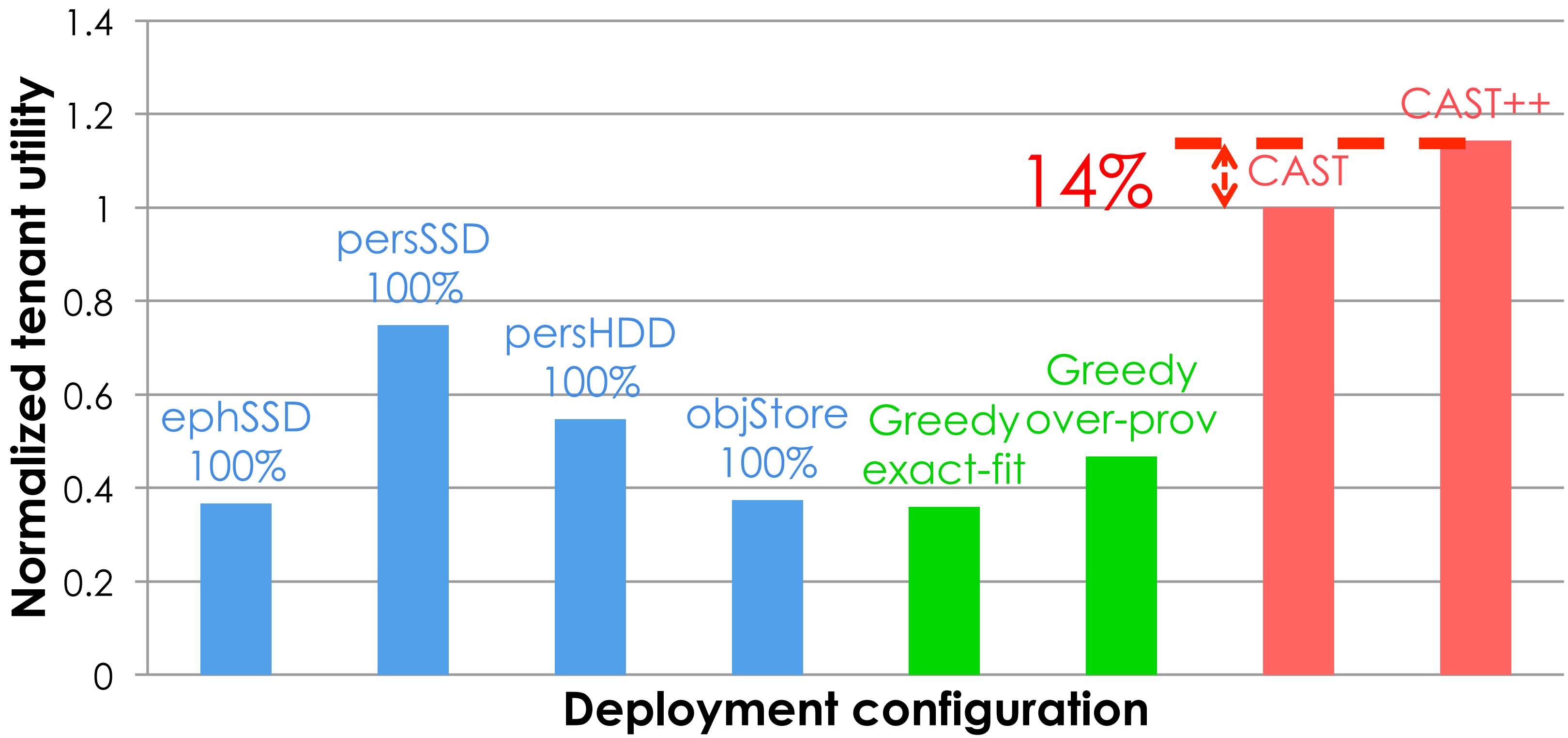
100-job Hadoop workload, simulating behaviors of Facebook's 3000-machine Hadoop cluster

Tenant utility improvement



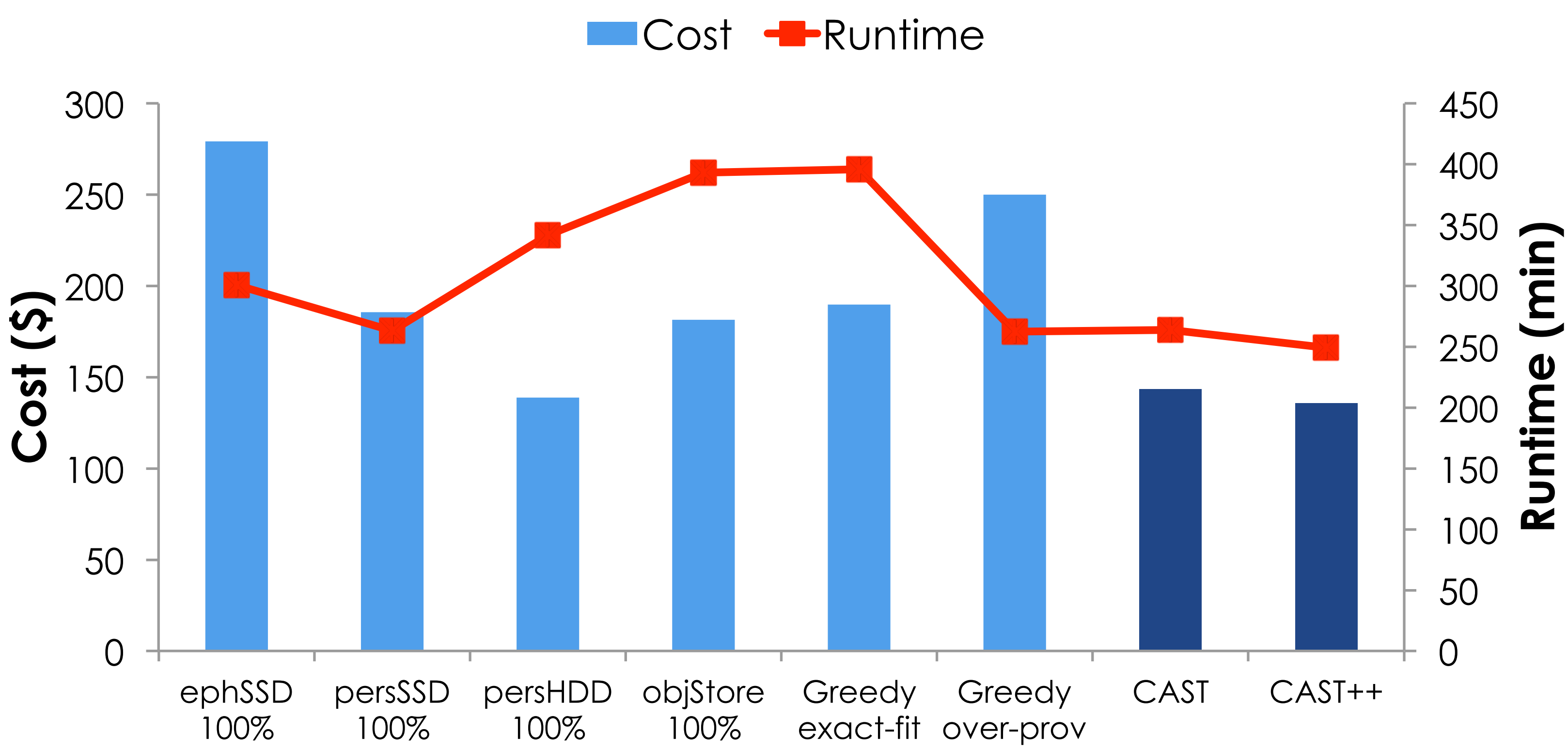
100-job Hadoop workload, simulating behaviors of Facebook's 3000-machine Hadoop cluster

Tenant utility improvement



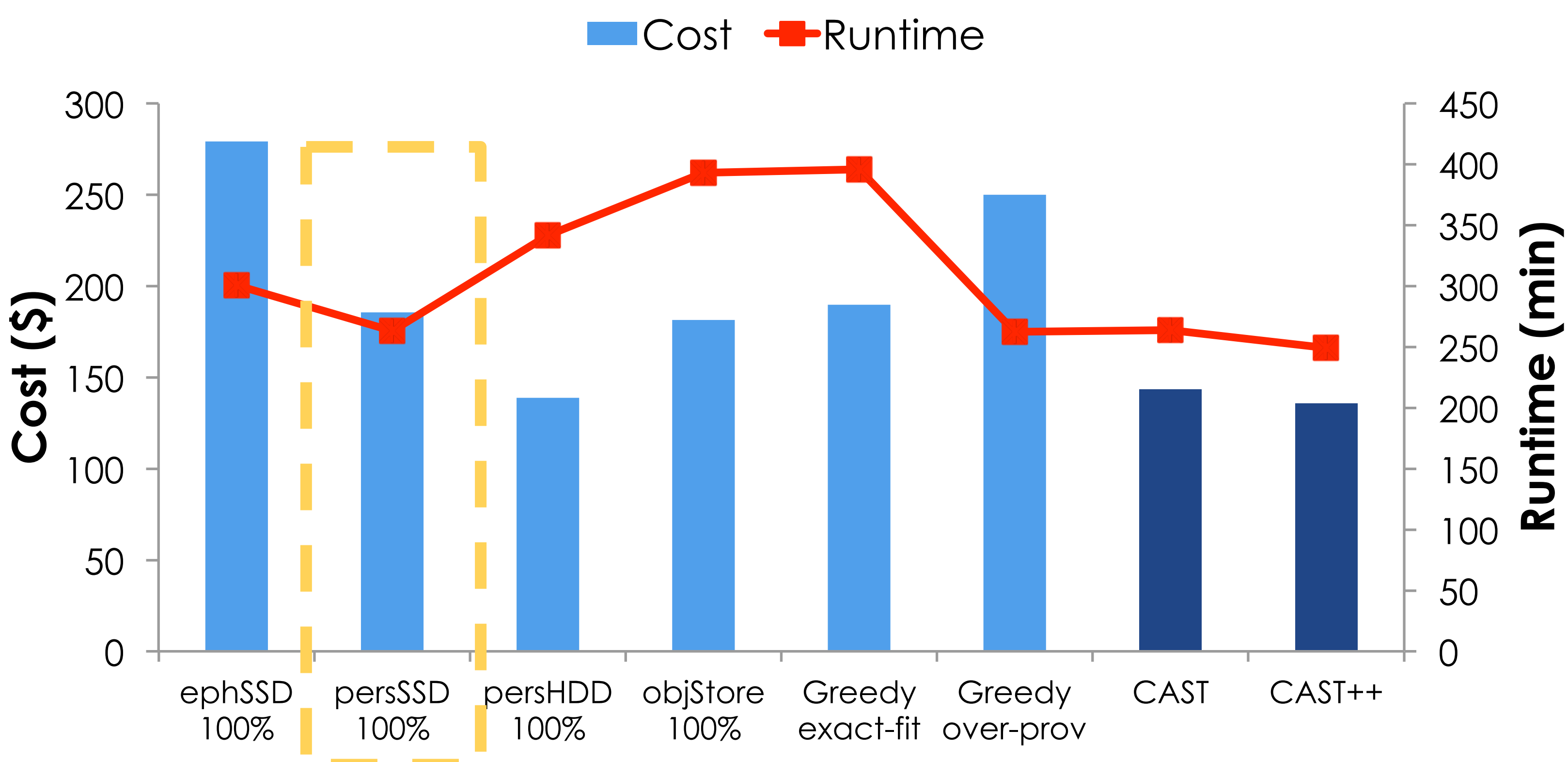
100-job Hadoop workload, simulating behaviors of Facebook's 3000-machine Hadoop cluster

$ cost vs. runtime



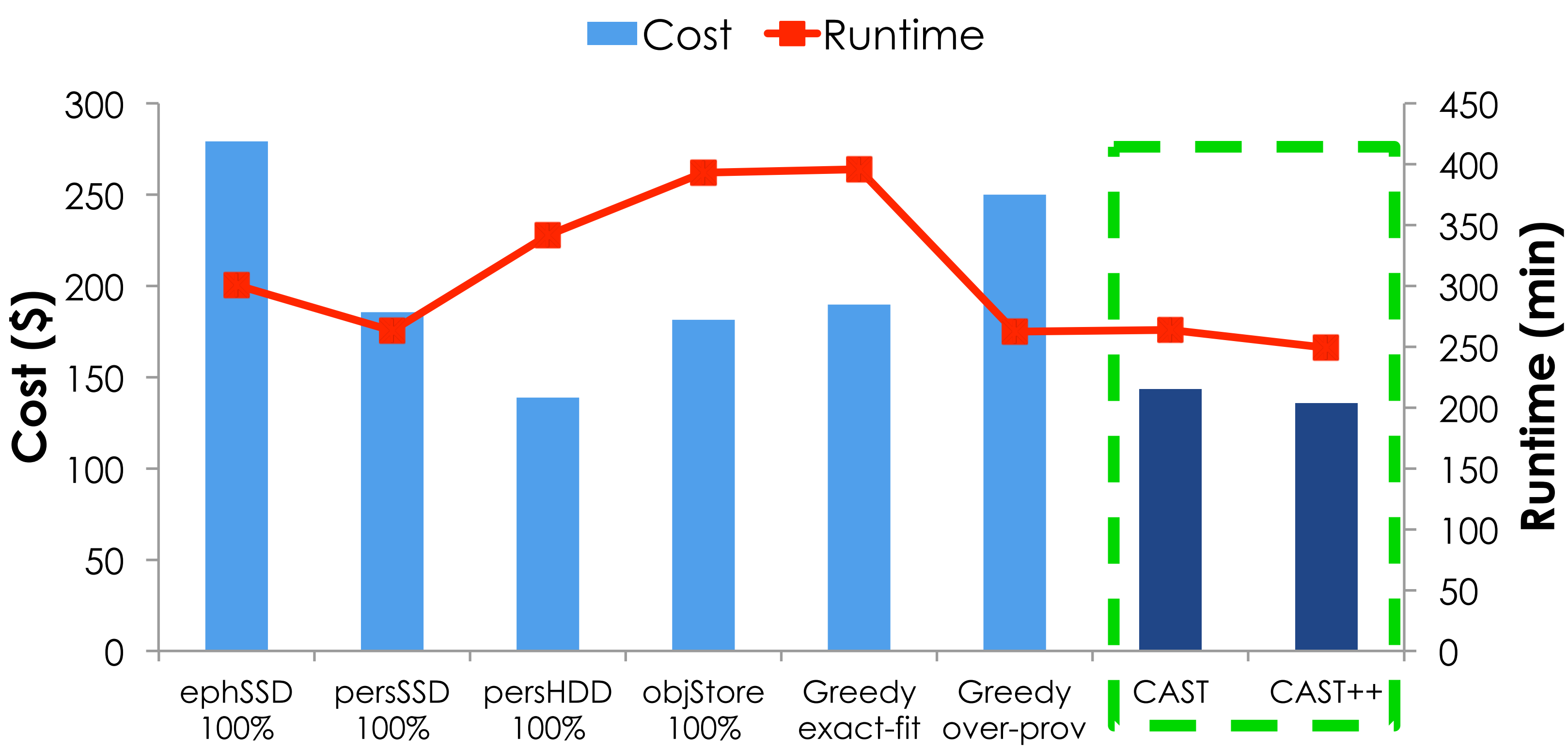
100-job Hadoop workload, simulating behaviors of Facebook's 3000-machine Hadoop cluster

$ cost vs. runtime



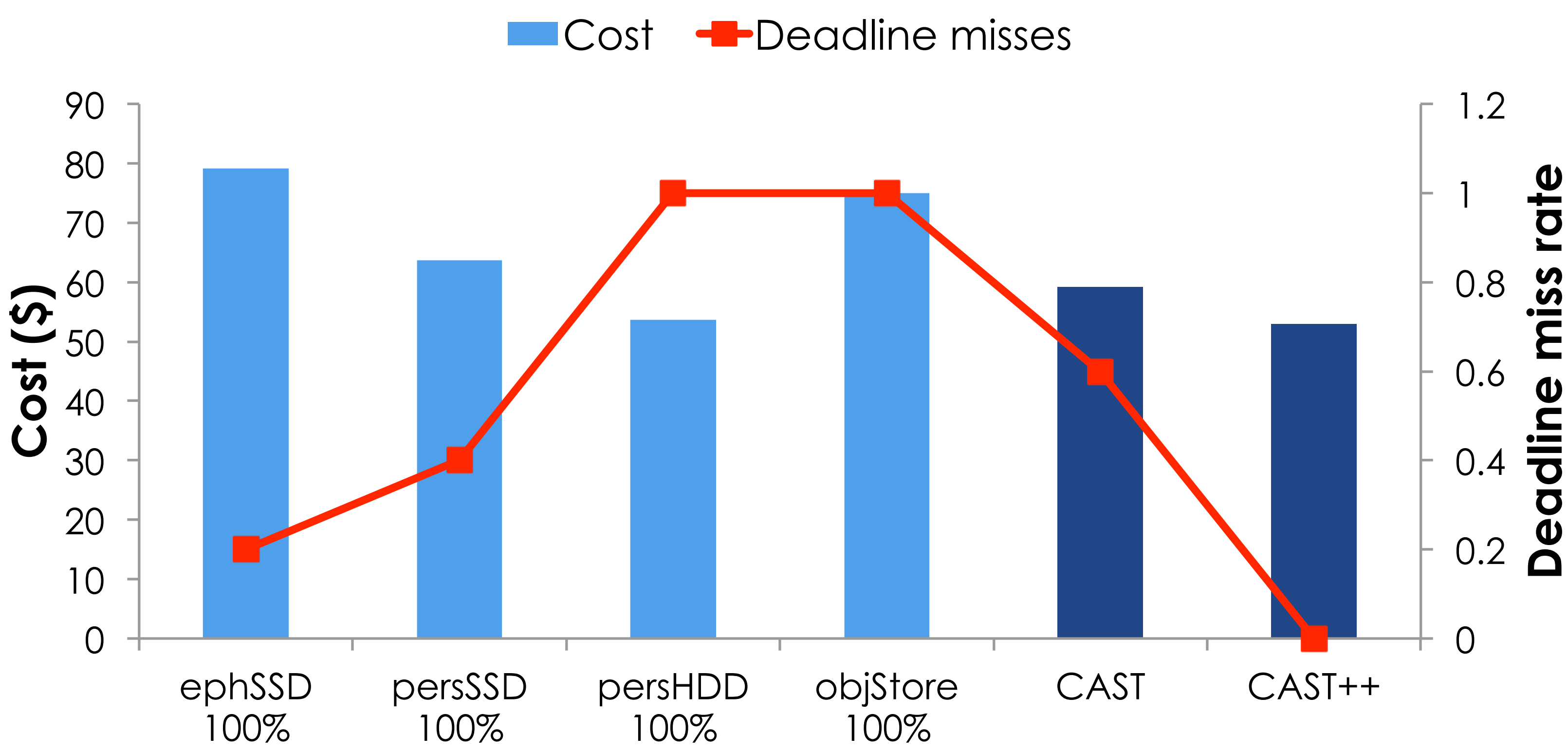
100-job Hadoop workload, simulating behaviors of Facebook's 3000-machine Hadoop cluster

$ cost vs. runtime



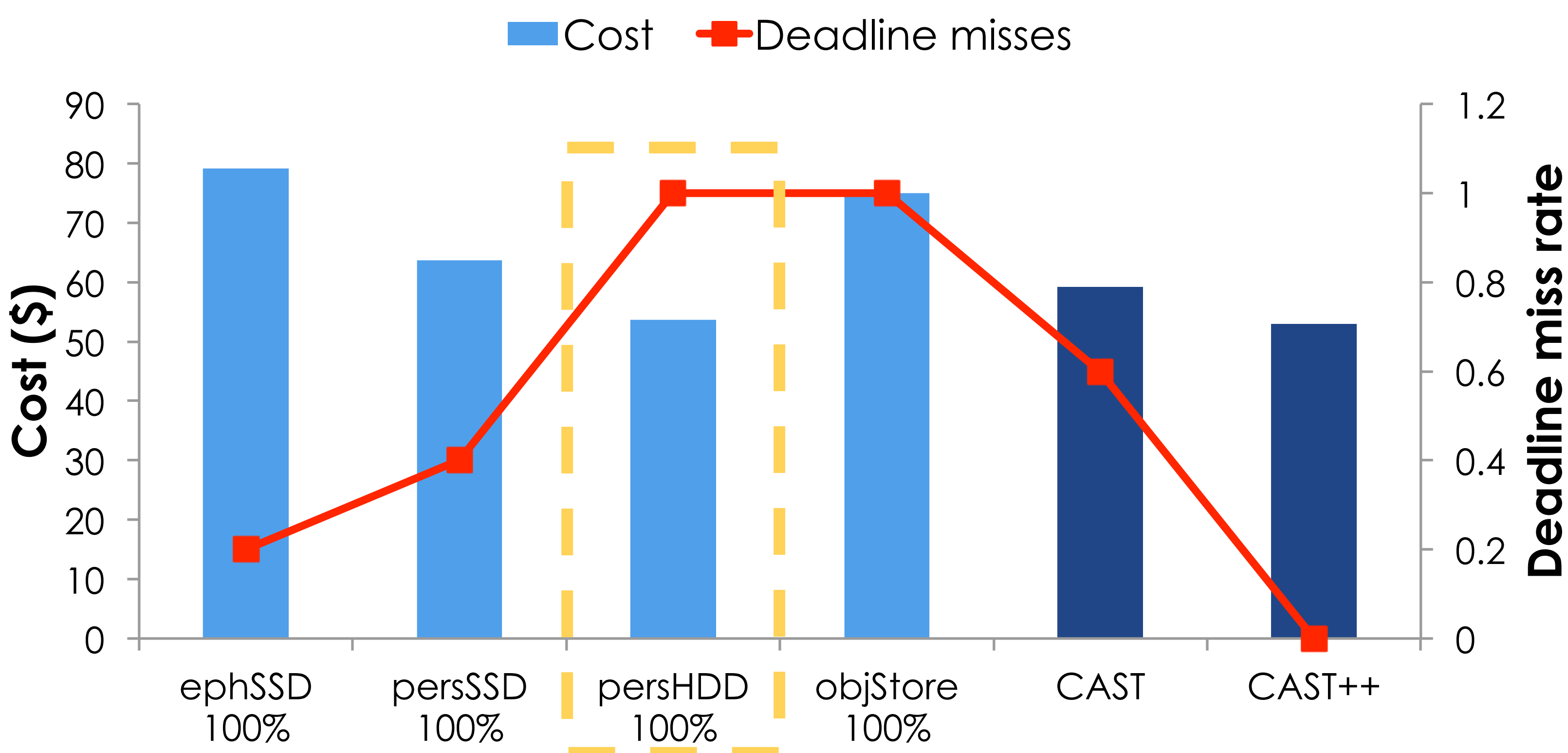
100-job Hadoop workload, simulating behaviors of Facebook's 3000-machine Hadoop cluster

Meeting workflow deadlines



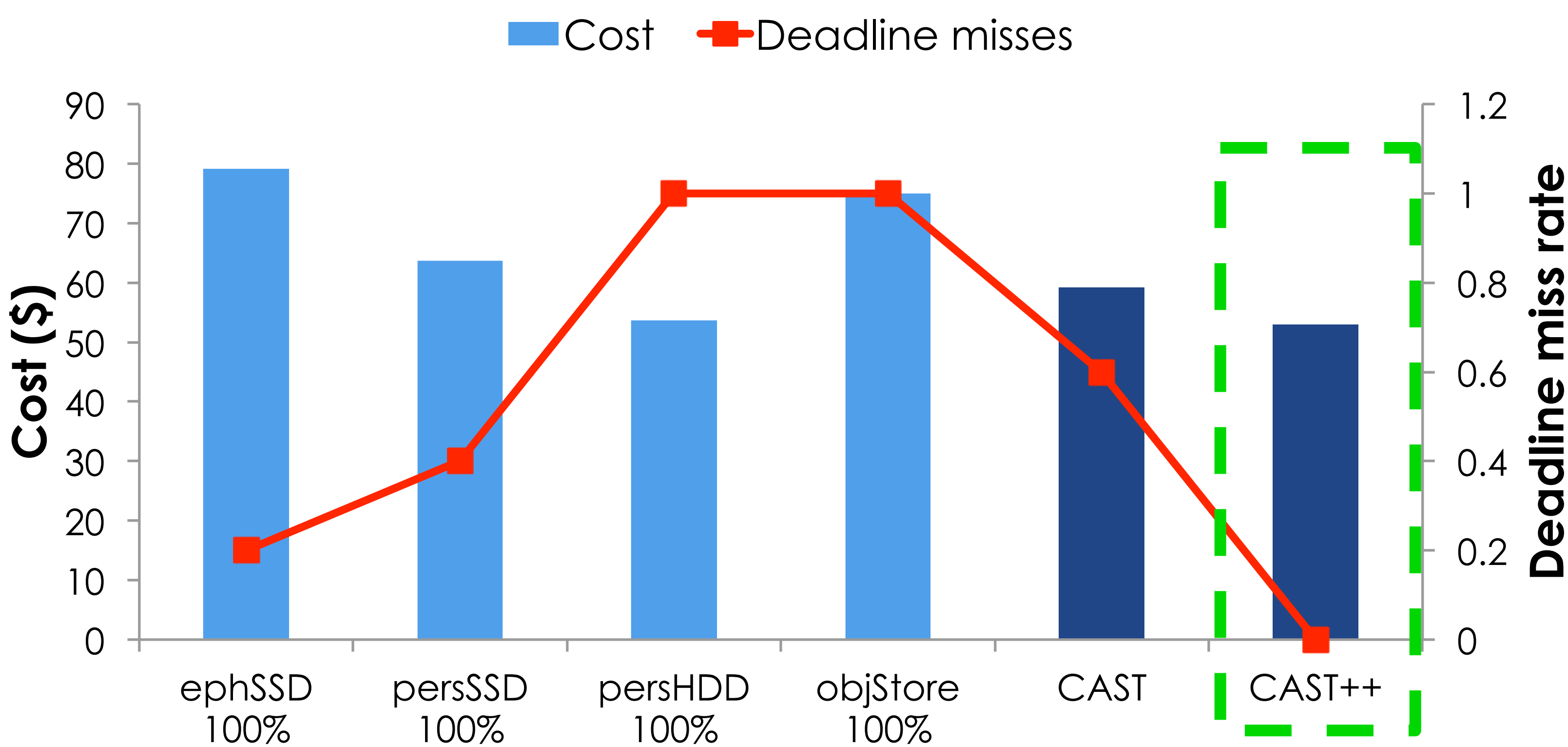
A workload consisting of 5 workflows, with a total of 31 analytics jobs

Meeting workflow deadlines



A workload consisting of 5 workflows, with a total of 31 analytics jobs

Meeting workflow deadlines



A workload consisting of 5 workflows, with a total of 31 analytics jobs

Conclusion

- **CAST** performs storage allocation and data placement for cloud analytics workloads
  - Leverages performance and pricing models of cloud storage services
  - Leverages analytics workload heterogeneity

- **CAST++** enhancements detect data reuse and inter-job dependencies
  - To further improve tenant utility
  - To effectively meet deadlines while minimizing $ cost

CAST

Thank you!

http://research.cs.vt.edu/dssl/

Yue Cheng M. Iqbal Safdar Aayush Gupta Ali R. Butt

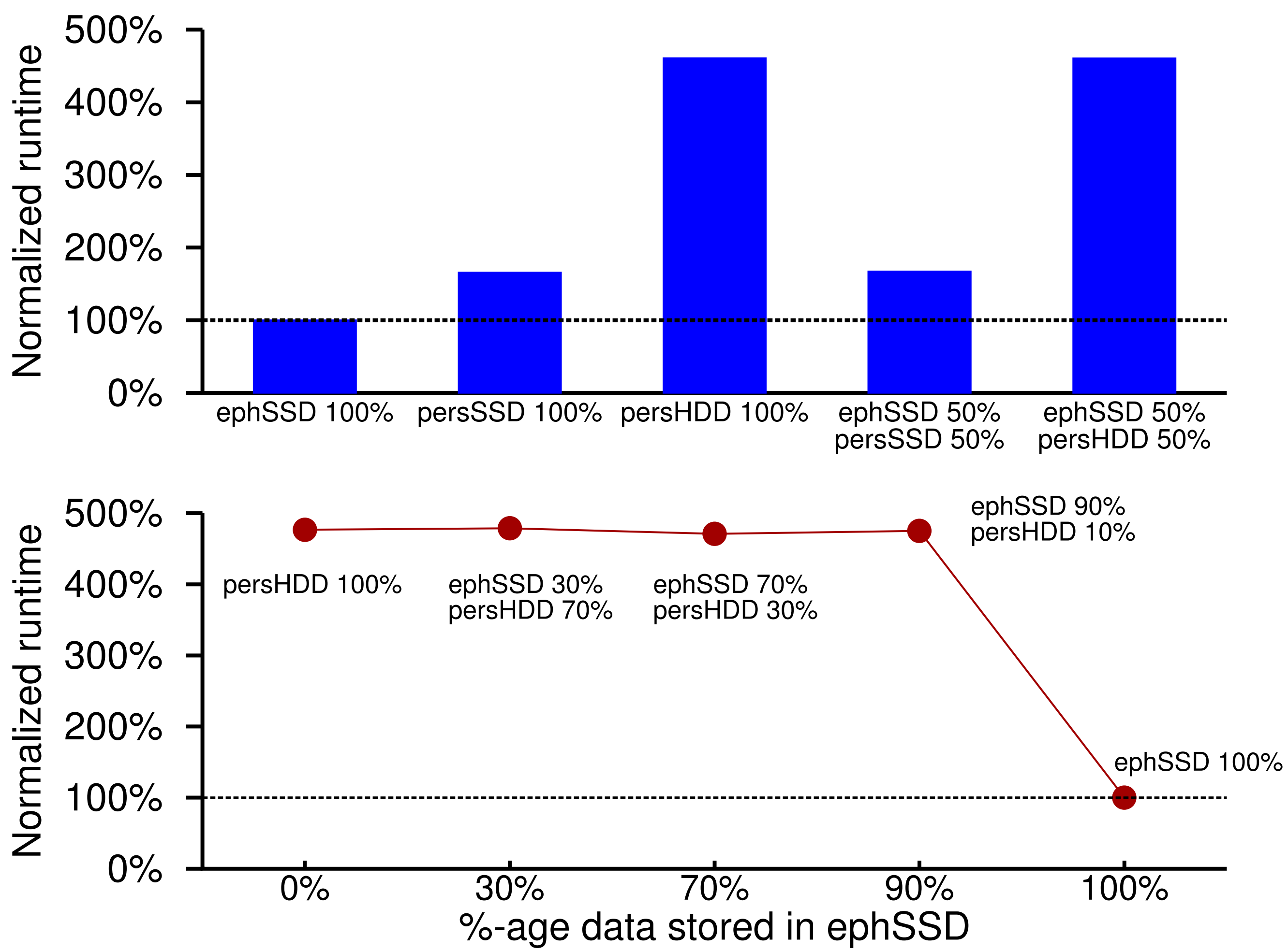
Backup Slides

Heterogeneity in cloud storage services

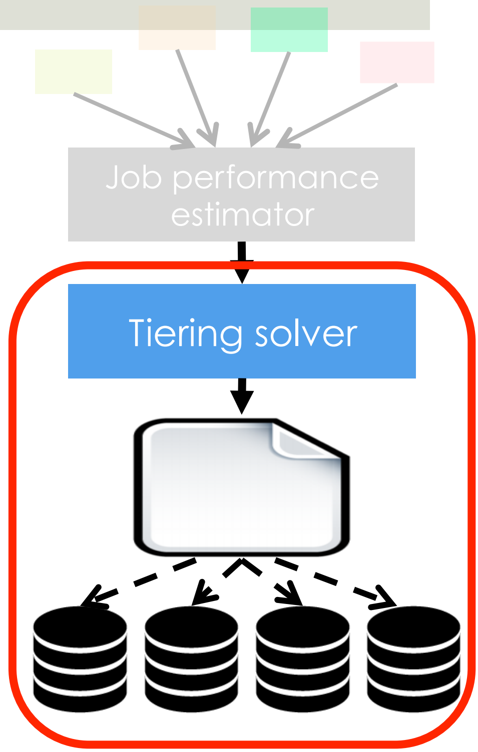
Storage type	Capacity (GB/volume)	Throughput (MB/sec)	IOPS (4KB)	Cost ($/month)
ephSSD	375	733	100000	0.218×375
persSSD	100	48	3000	0.17×100
	250	118	7500	0.17×250
	500	**234**	**15000**	0.17×500
persHDD	100	20	150	0.04×100
	250	45	375	0.04×250
	500	**97**	**750**	0.04×500
objStore	N/A	265	550	0.026/GB

A 500GB persSSD provides 1.4X higher throughput & 19X higher IOPS than a 500GB persHDD.

Straggler issue in fine-grained tiering



Tiering solver



- Optimization
  - Objective function

$$\textbf{max}\ \ \text{Tenant utility} = \frac{1/T}{(\$_{vm} + \$_{store})}$$

  - Constraints

Tuning knob: capacity of J_i

Input size of J_i

Output size of J_i

$$c_i \geq (I_i + M_i + O_i) \qquad (\forall i \in J)$$

Intermediate data size of J_i

Tuning knob: Storage service of J_i

$$T = \sum_{i=1}^{J} REG\left(s_i, C[s_i], \hat{R}, \hat{L}_i\right),\ \text{where}\ \ s_i \in F$$

$$\$_{vm} = n_{vm} \cdot (P_{vm} \cdot T)$$

$$\$_{store} = \sum_{f}^{F} \left(C[f] \cdot (P_{store}[f] \cdot \left\lceil T/60 \right\rceil) \right)$$

Hadoop traces from Facebook

- More than **99%** of data touched by large jobs that incur most of the storage cost
- The aggregated data size for small jobs is only **0.1%** of the total dataset size
- We focus on large jobs that have enough # mappers & reducers to fully utilize the cluster computing capacity

Storage capacity/service breakdown