SpotOn:

A BATCH COMPUTING SERVICE FOR THE SPOT MARKET

Supreeth Subramanya, Tian Guo, Prateek Sharma,
David Irwin, Prashant Shenoy

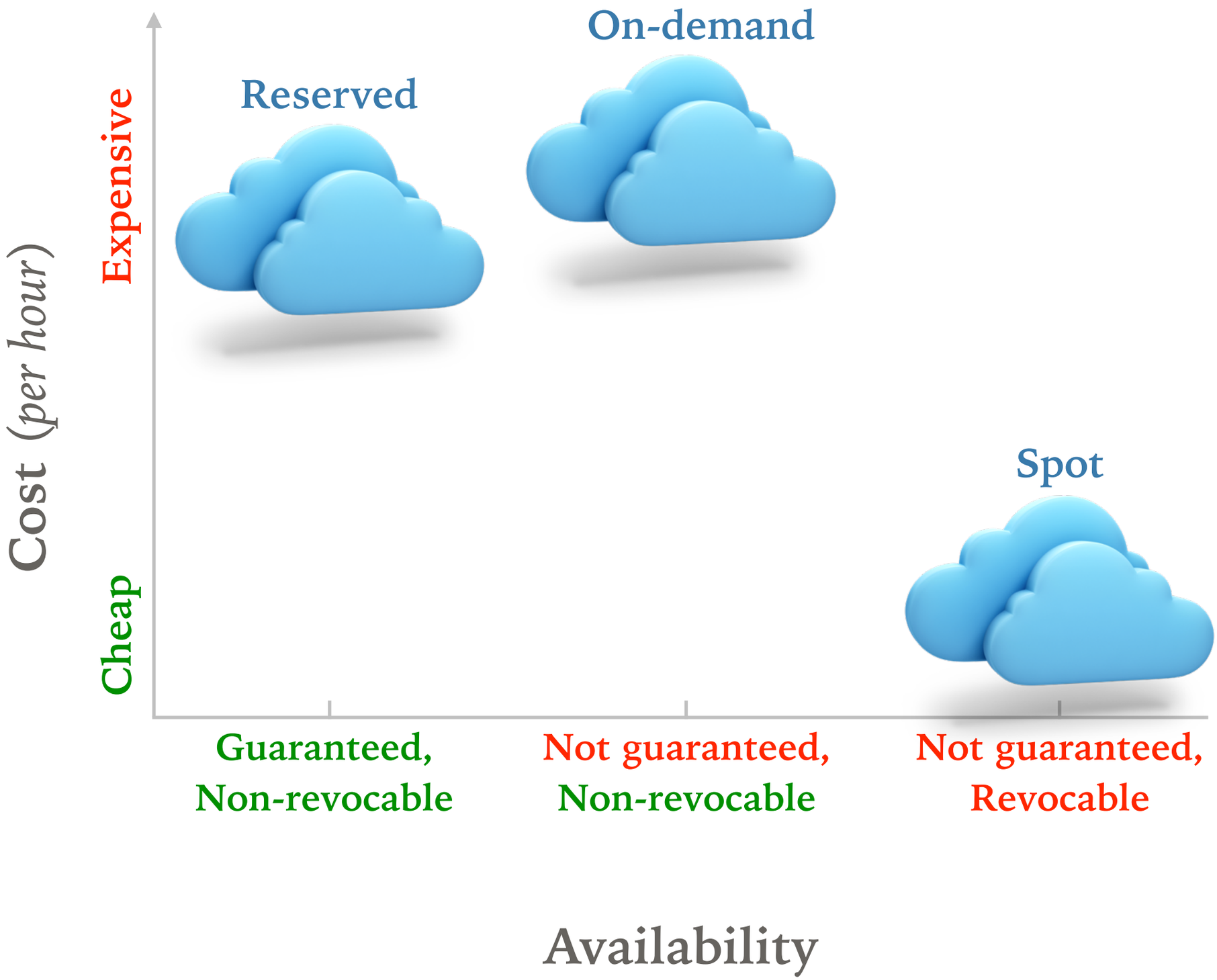
University of Massachusetts Amherst

INFRASTRUCTURE CLOUD

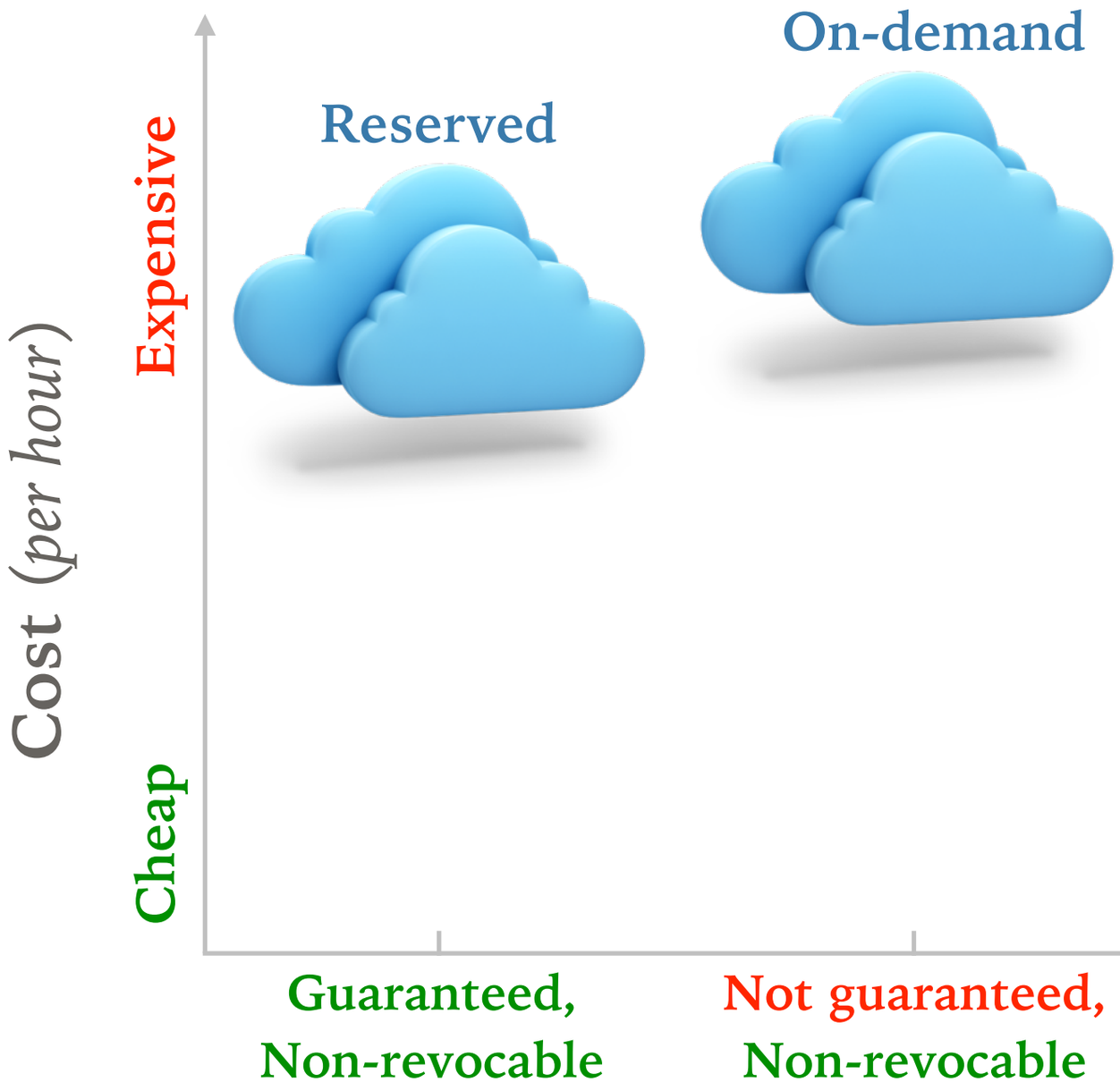
On-demand

INFRASTRUCTURE CLOUD

Cost vs. Availability Tradeoff in IaaS Clouds



Cost vs. Availability Tradeoff in IaaS Clouds



Spot Instances

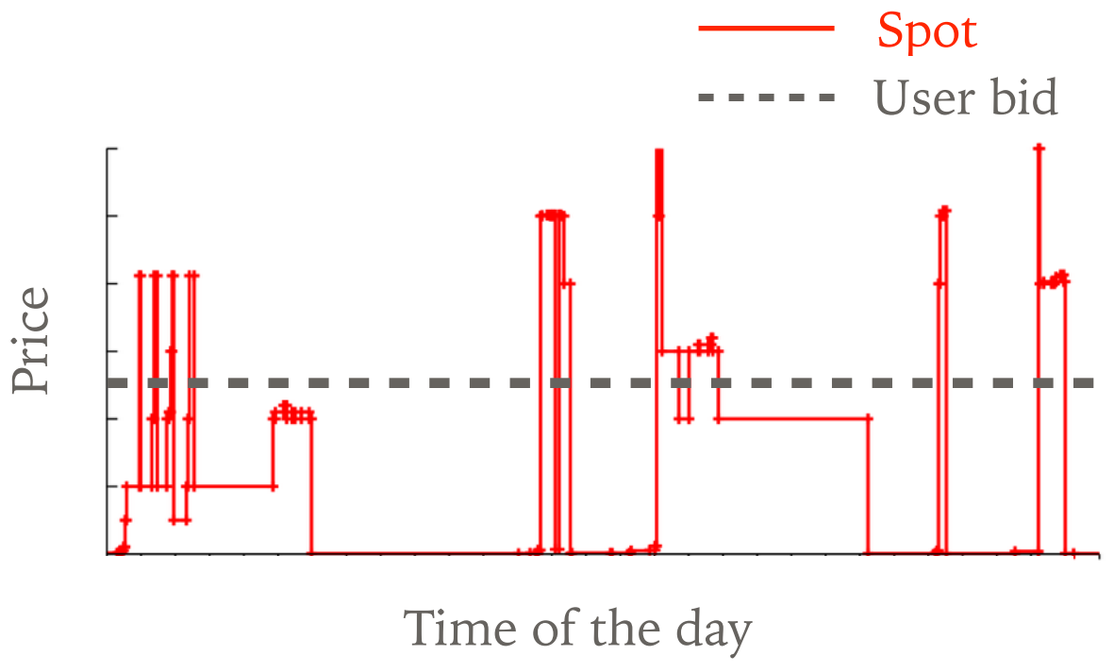
amazon web services™ | EC2

Google Cloud Platform
Preemptible VM

SPOT MARKETS

Amazon EC2

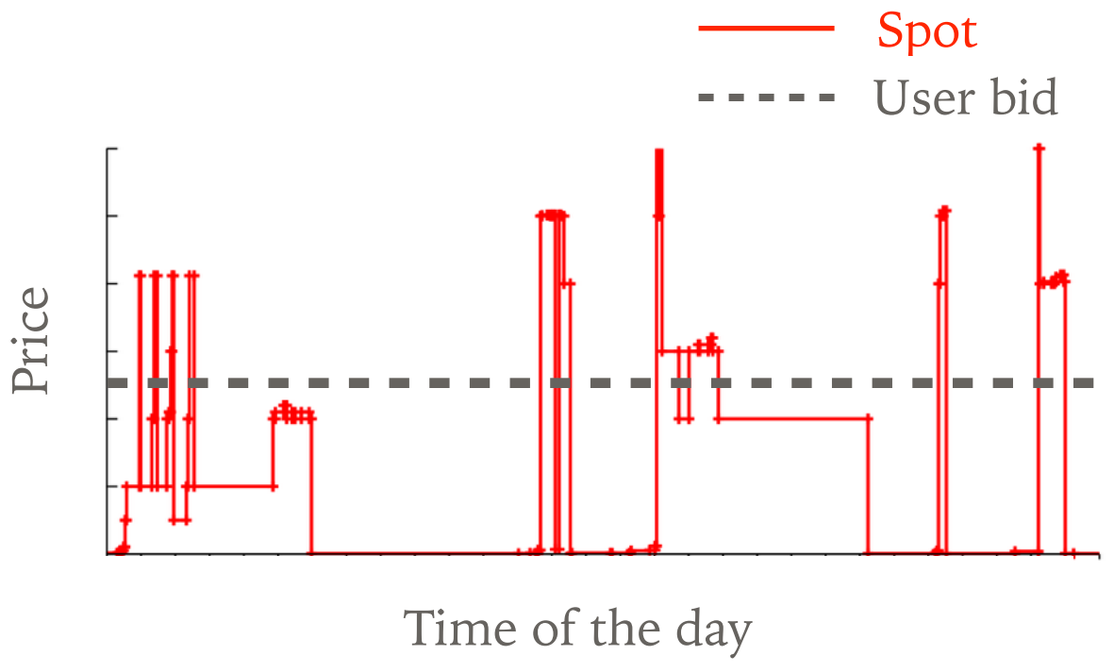
- ***Bid*** in a 2nd price auction
- ***Acquire*** when bid > spot price
- ***Terminate*** when spot price > bid



SPOT MARKETS

Amazon EC2

- ***Bid*** in a 2nd price auction
- ***Acquire*** when bid > spot price
- ***Terminate*** when spot price > bid



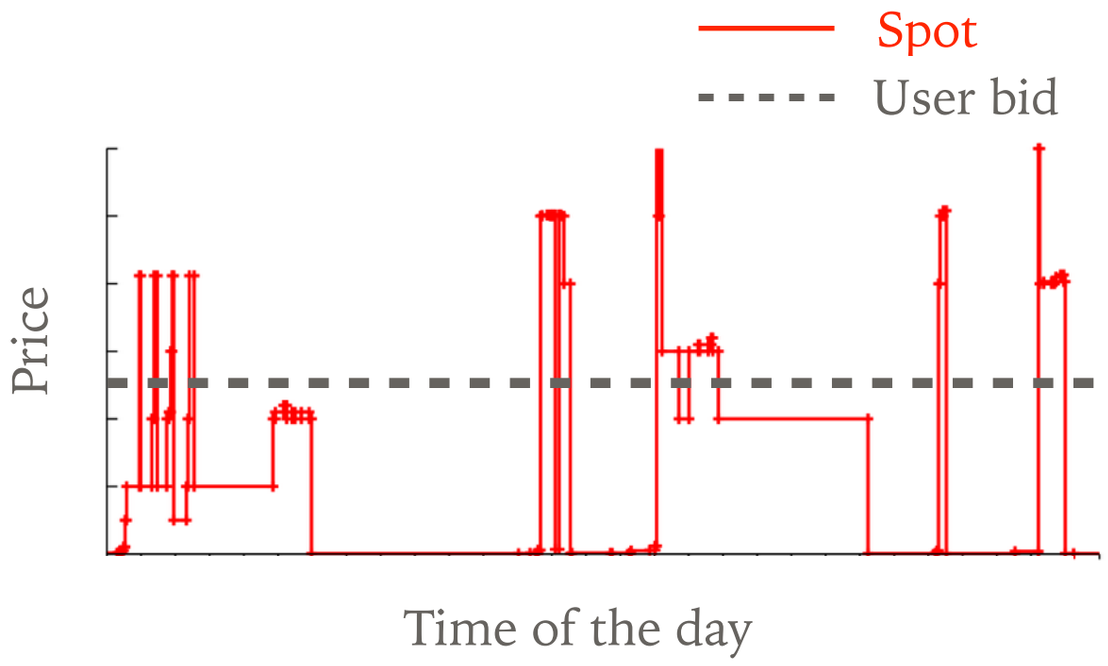
How do we mitigate the impact of revocation?

1. Raise the bid
2. Employ fault-tolerance mechanisms

SPOT MARKETS

Amazon EC2

- ***Bid*** in a 2nd price auction
- ***Acquire*** when bid > spot price
- ***Terminate*** when spot price > bid



How do we mitigate the impact of revocation?

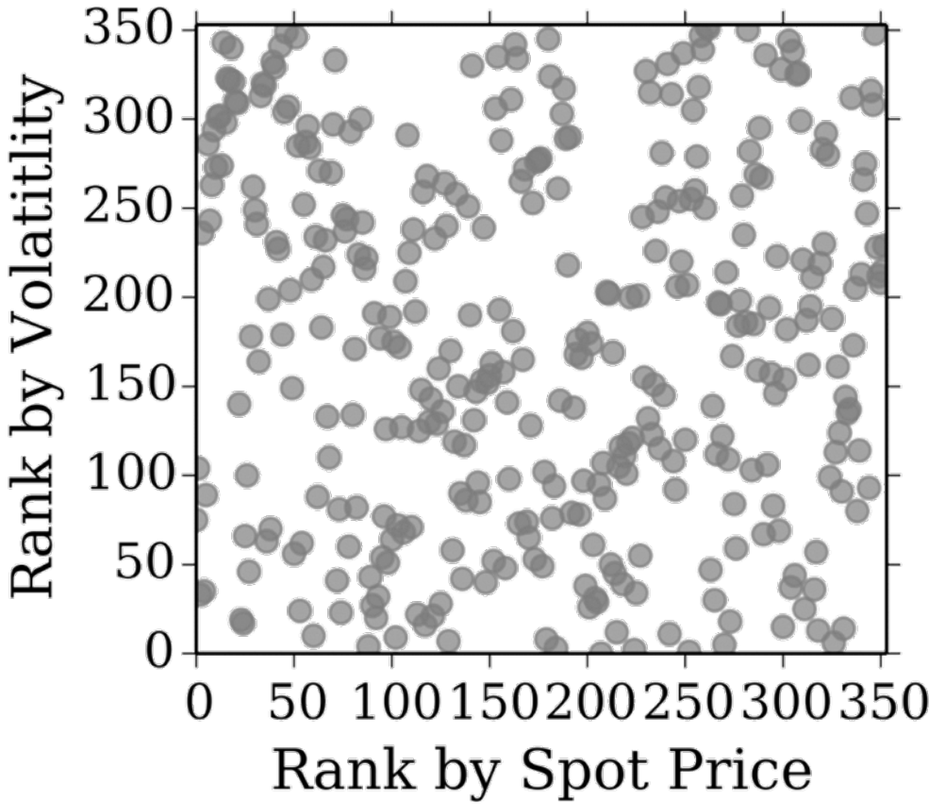
1. Raise the bid
2. **Employ fault-tolerance mechanisms**

CHALLENGES — SPOT MARKET COMPLEXITY

Amazon EC2 operates ***~4000*** spot markets

CHALLENGES — SPOT MARKET COMPLEXITY

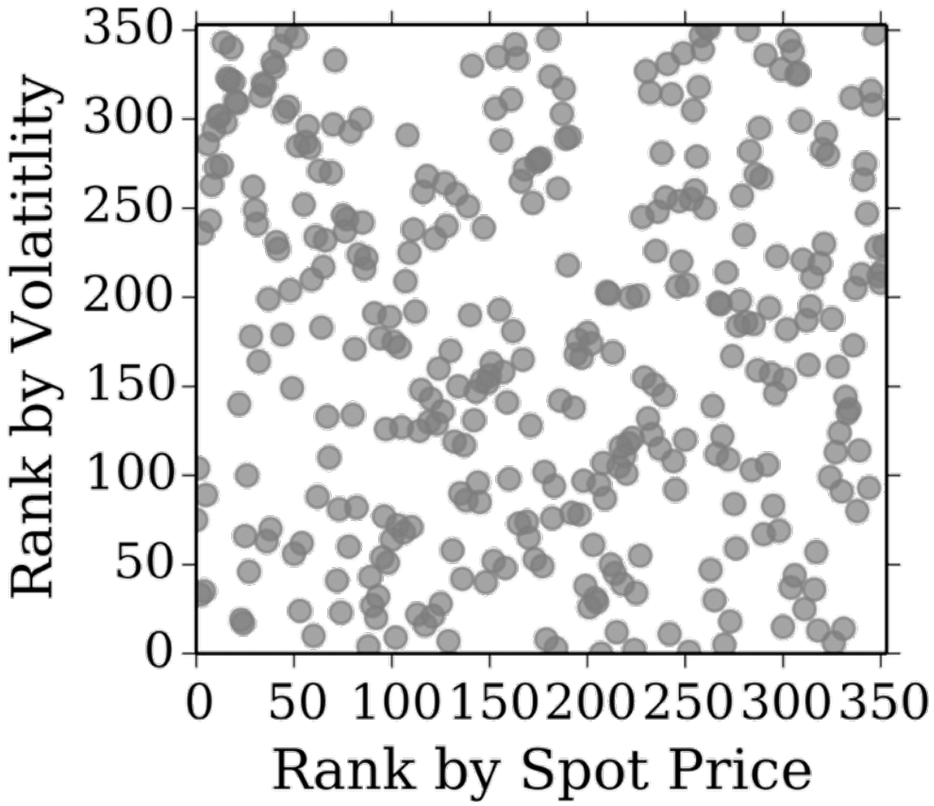
Amazon EC2 operates ***~4000*** spot markets



Scatterplot of ranks for EC2 spot markets

CHALLENGES — SPOT MARKET COMPLEXITY

Amazon EC2 operates ***~4000*** spot markets



Scatterplot of ranks for EC2 spot markets

Selecting an instance that yields

lowest cost per unit of computation

while also considering the

probability of revocation is complex

CHALLENGES — APPLICATION COMPLEXITY

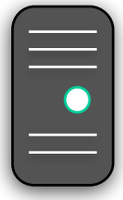
Resource Vector

CPU : IO	Working Set	Disk Type	Running Time
1:1	8GB	Remote	1 hour

CHALLENGES — APPLICATION COMPLEXITY

Resource Vector

CPU : IO	Working Set	Disk Type	Running Time
1:1	8GB	Remote	1 hour



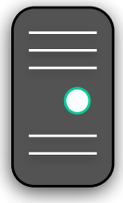
Spot VM

Cost	Revocation Rate	Fault-tolerance Mechanism
20% On-demand	2.4 per day	Checkpoint (every 900s)

CHALLENGES — APPLICATION COMPLEXITY

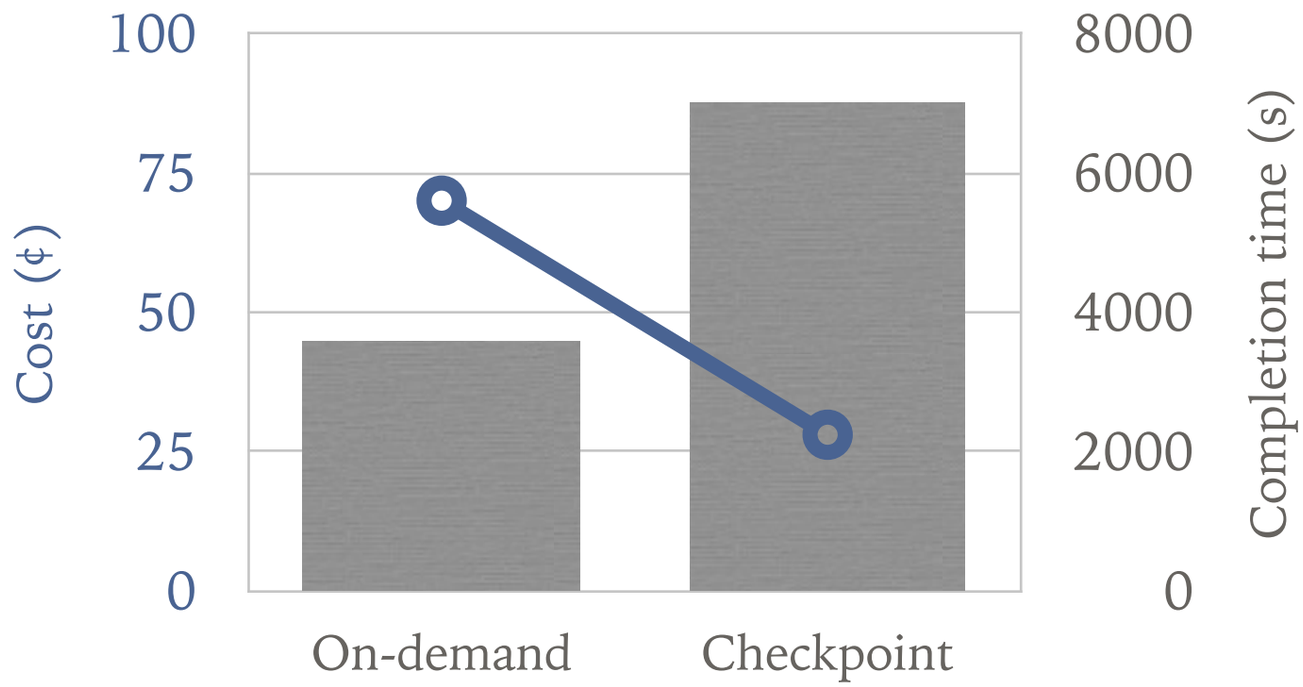
Resource Vector

CPU : IO	Working Set	Disk Type	Running Time
1:1	8GB	Remote	1 hour



Spot VM

Cost	Revocation Rate	Fault-tolerance Mechanism
20% On-demand	2.4 per day	Checkpoint (every 900s)

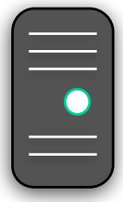


CHALLENGES — APPLICATION COMPLEXITY

Resource Vector

CPU : IO	Working Set	Disk Type	Running Time
1:1	8GB	R…	1 hour

Local



Spot VM

Cost	Revocation Rate	Fault-toler… …chanism
20% On-demand	2.4 per day	Ch… …00s)

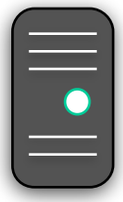
Replicate (deg=2)

CHALLENGES — APPLICATION COMPLEXITY

Resource Vector

CPU : IO	Working Set	Disk Type	Running Time
1:1	8GB	R…	1 hour

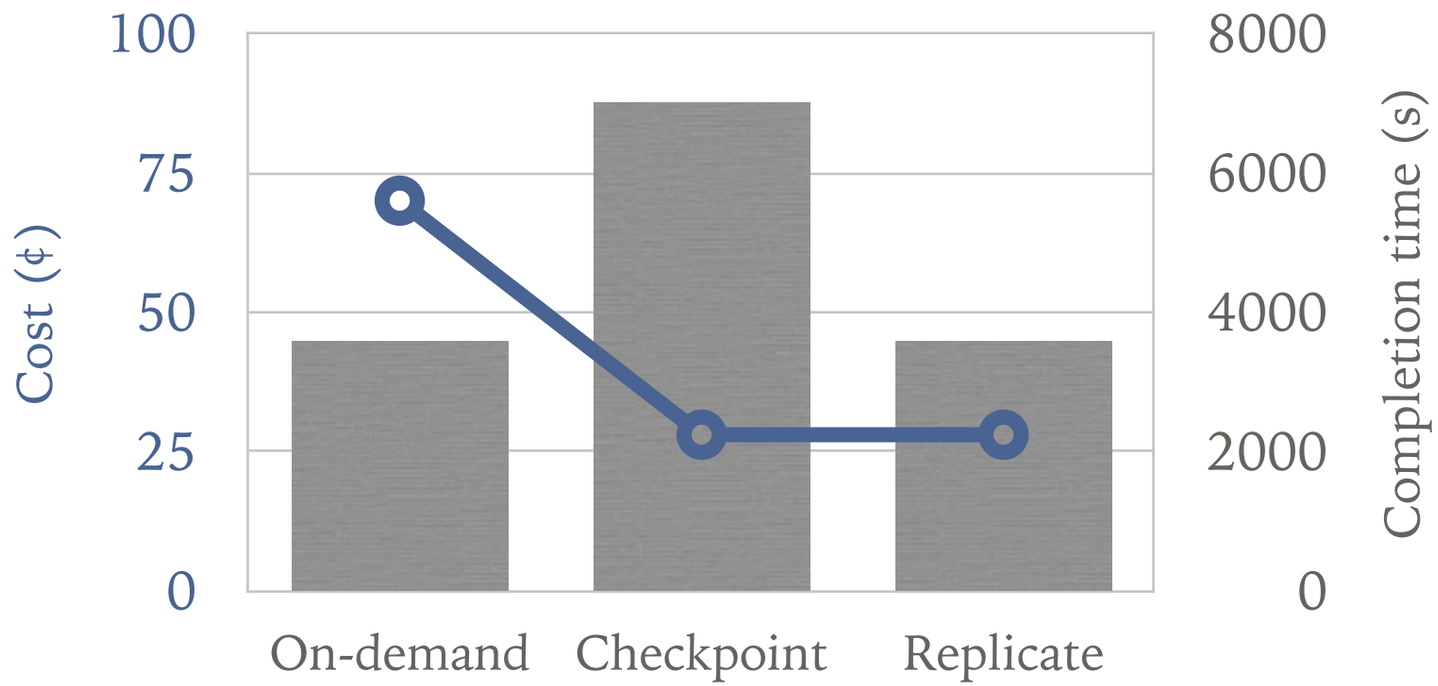
Local



Spot VM

Cost	Revocation Rate	Fault-tole…chanism
20% On-demand	2.4 per day	Che… …00s)

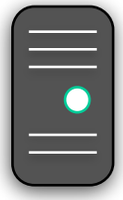
Replicate (deg=2)



CHALLENGES — APPLICATION COMPLEXITY

Resource Vector

CPU : IO	Working Set	Disk Type	Running Time
1:1	8GB	Local	1 hour



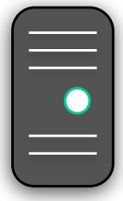
Spot VM

Cost	Revoca[illegible]ate	Fault-tolerance Mechanism
20% On-demand	>24 per day	Checkpoint (every 900s)

CHALLENGES — APPLICATION COMPLEXITY

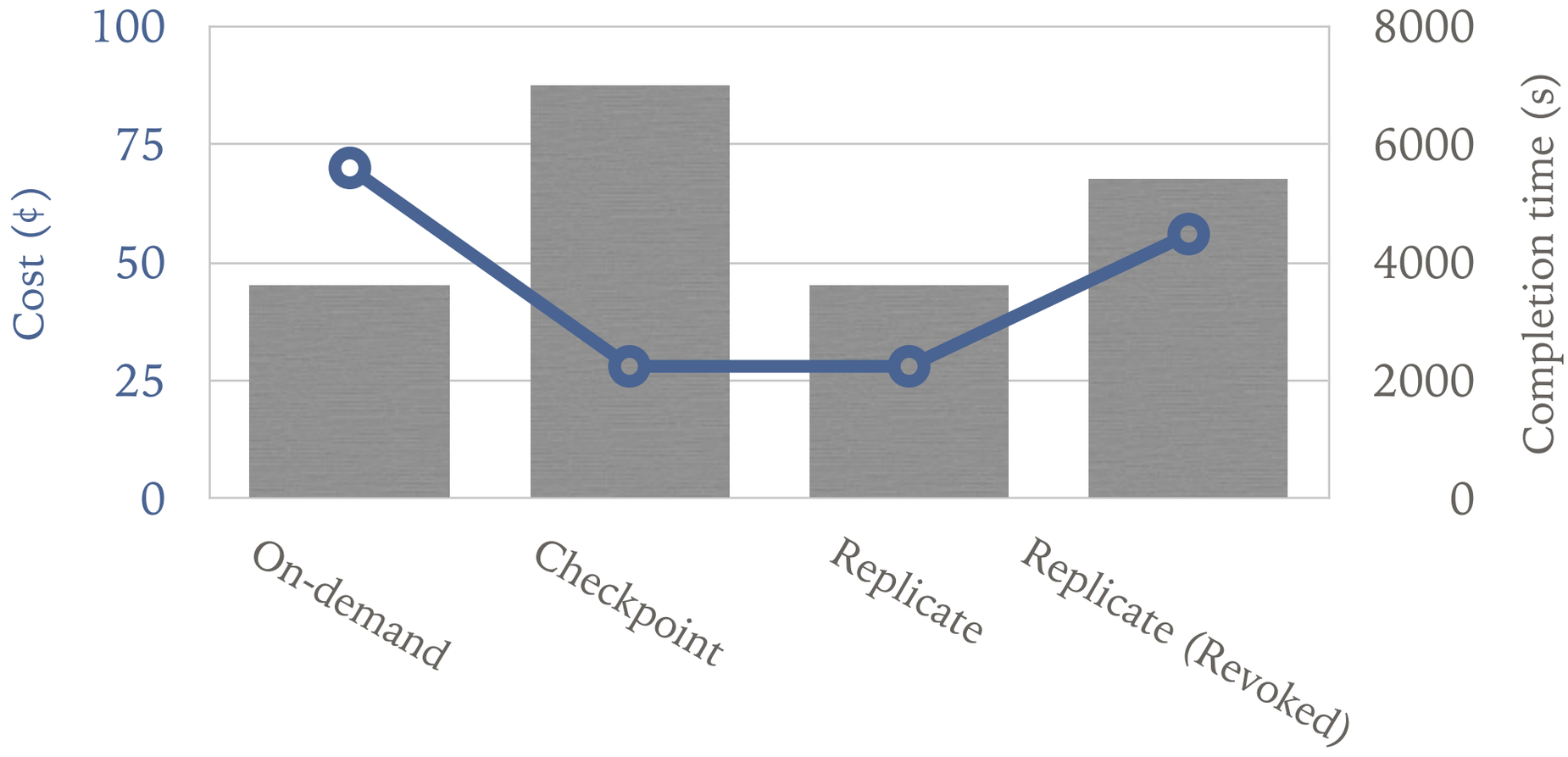
Resource Vector

CPU : IO	Working Set	Disk Type	Running Time
1:1	8GB	Local	1 hour



Spot VM

Cost	Revoca[illegible]te	Fault-tolerance Mechanism
20% On-demand	>24 per day	Checkpoint (every 900s)



SpotOn: a batch computing service

SpotOn: a batch computing service

Service that accepts batch jobs from users and runs them on spot instances

Manages application and spot market complexity transparently

SpotOn: a batch computing service

Service that accepts batch jobs from users and runs them on spot instances

Manages application and spot market complexity transparently

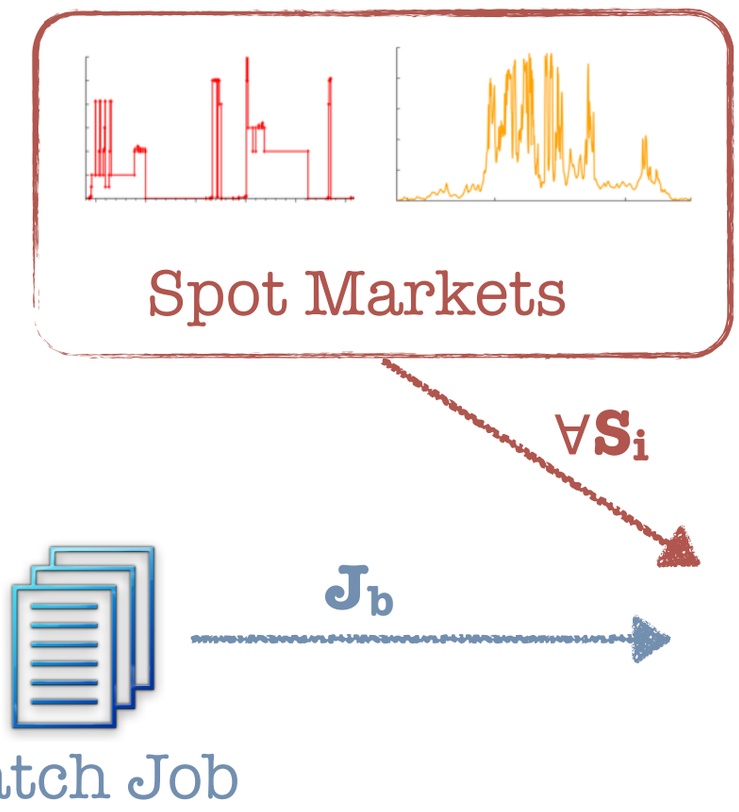
Run batch jobs at ***on-demand performance***
but paying ***spot market price***

GREEDY SELECTION ALGORITHM

Selecting the best spot market and fault-tolerance mechanism

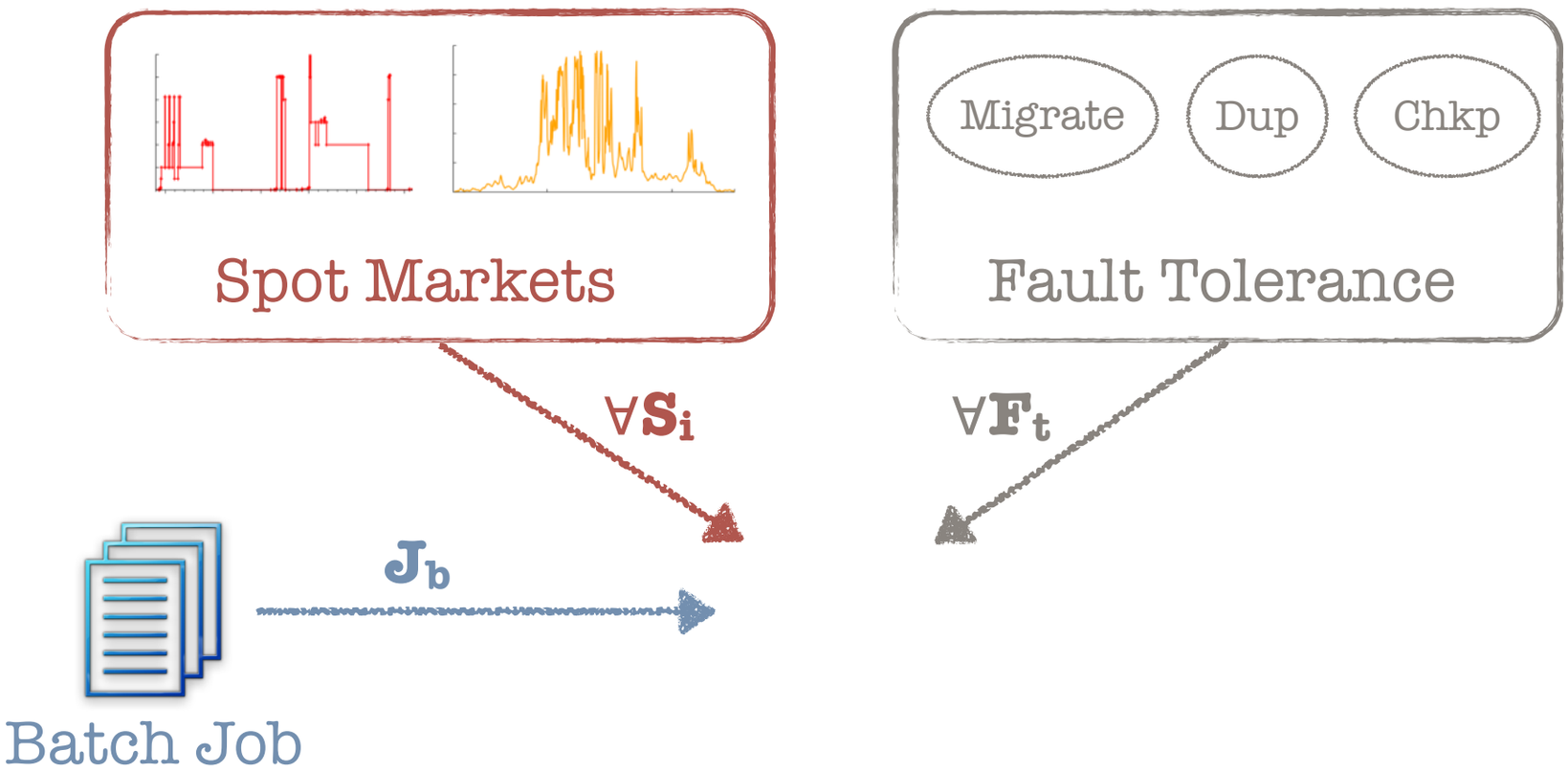
GREEDY SELECTION ALGORITHM

Selecting the best spot market and fault-tolerance mechanism



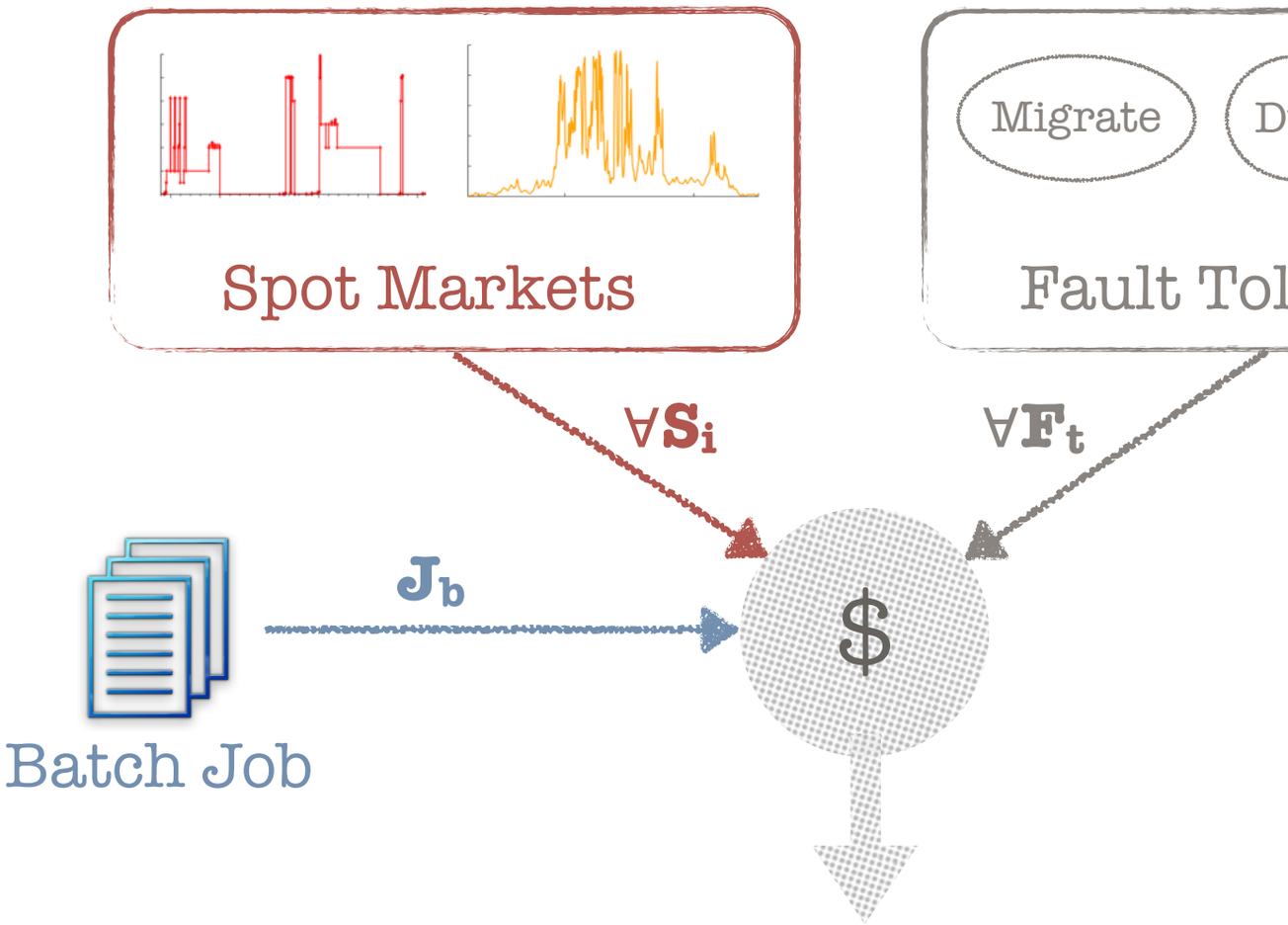
GREEDY SELECTION ALGORITHM

Selecting the best spot market and fault-tolerance mechanism



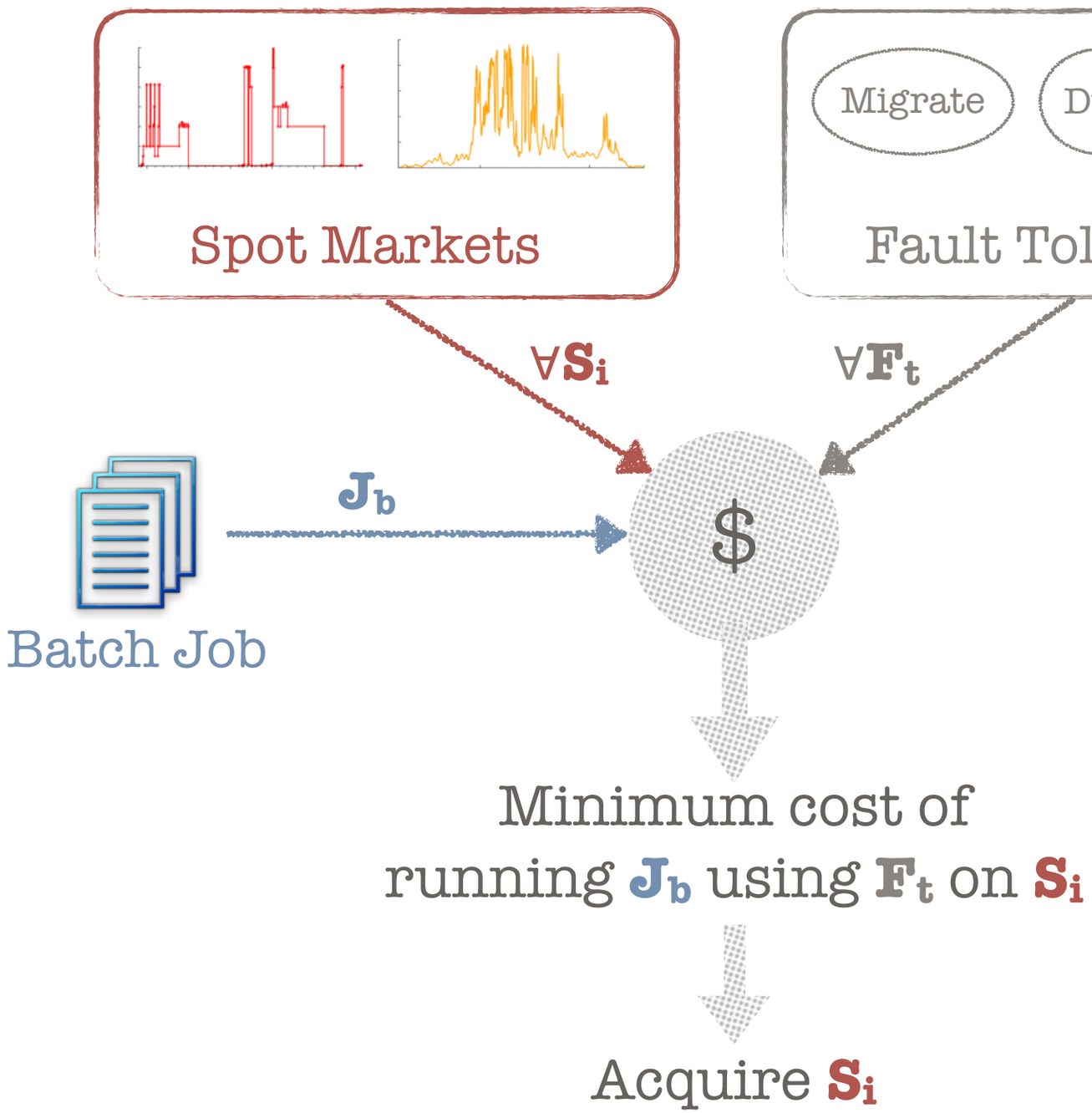
GREEDY SELECTION ALGORITHM

Selecting the best spot market and fault-tolerance mechanism



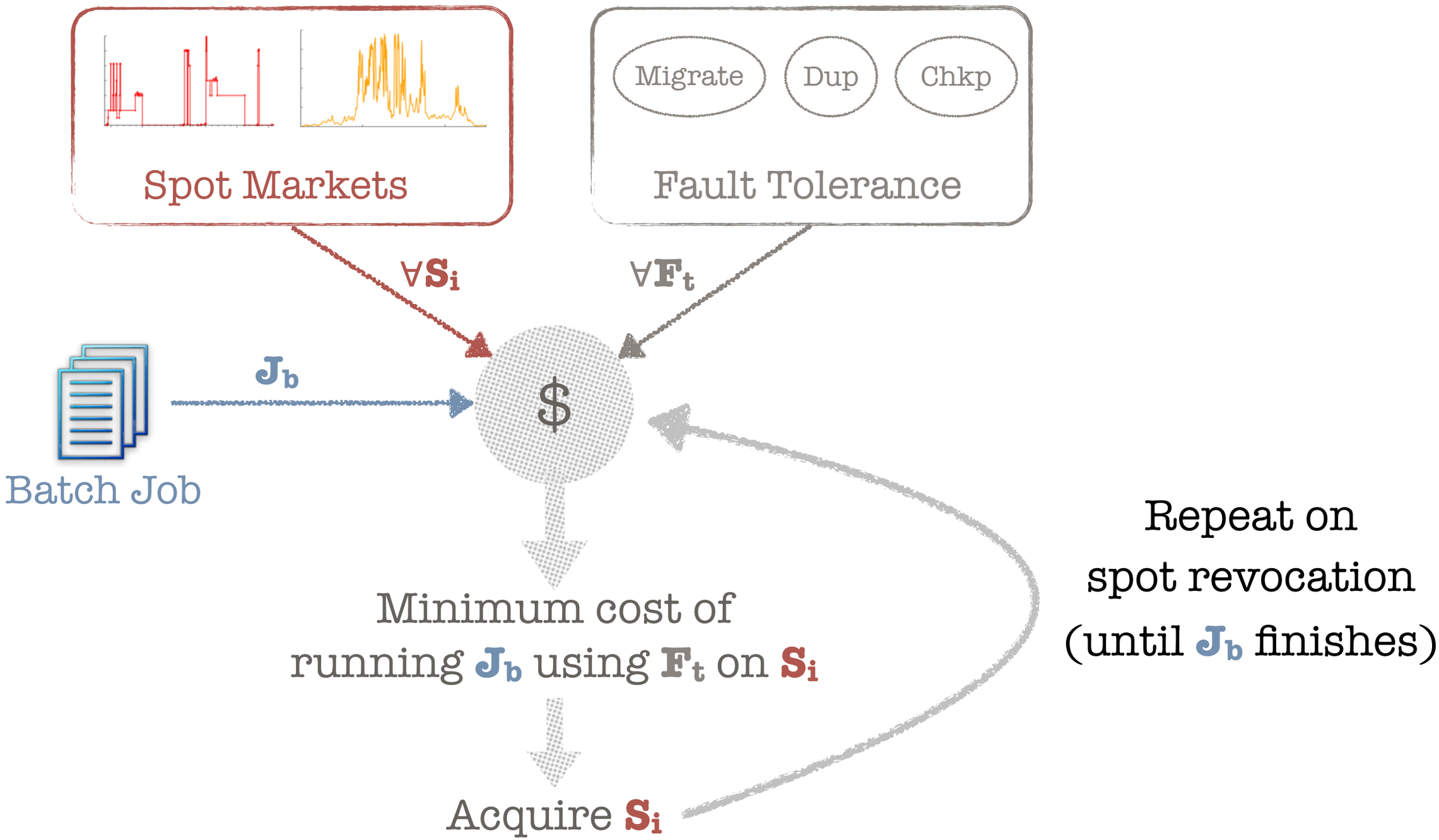
GREEDY SELECTION ALGORITHM

Selecting the best spot market and fault-tolerance mechanism



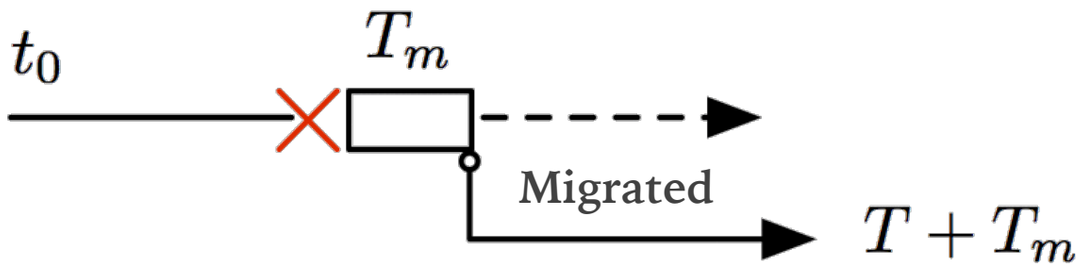
GREEDY SELECTION ALGORITHM

Selecting the best spot market and fault-tolerance mechanism



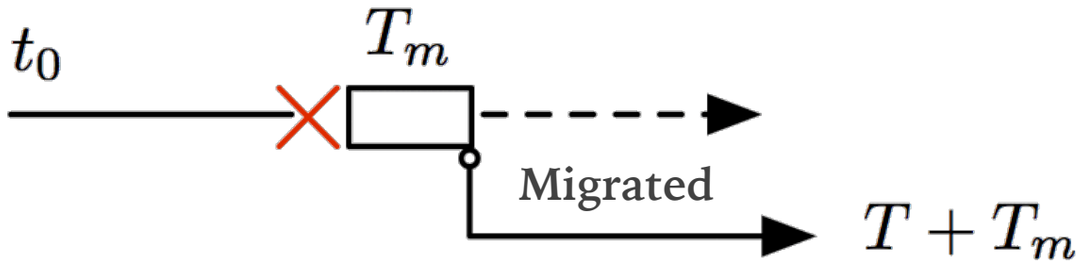
FAULT TOLERANCE (1/3)

Reactive Migration



FAULT TOLERANCE (1/3)

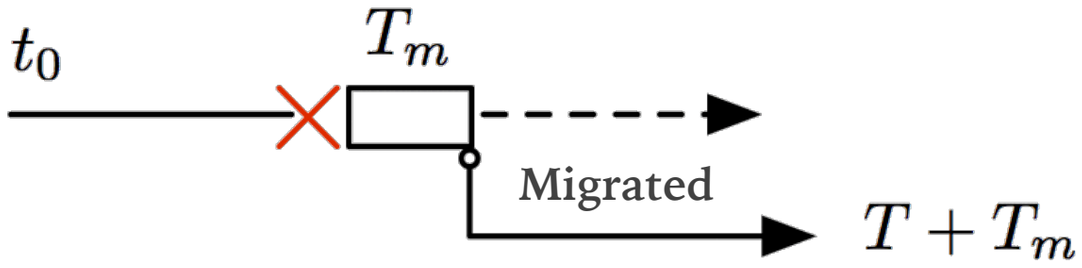
Reactive Migration



$$T_M \propto \frac{\text{size of memory + local disk}}{\text{remote disk bandwidth}}$$

FAULT TOLERANCE (1/3)

Reactive Migration



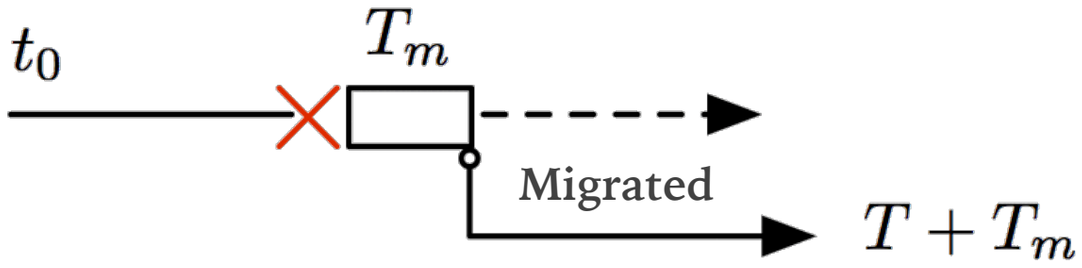
$$T_M \propto \frac{\text{size of memory + local disk}}{\text{remote disk bandwidth}}$$

$Z_k \rightarrow$ Random variable measuring time to revocation

$P_k \rightarrow$ Probability job gets revoked before completion

FAULT TOLERANCE (1/3)

Reactive Migration



$$T_M \propto \frac{\text{size of memory + local disk}}{\text{remote disk bandwidth}}$$

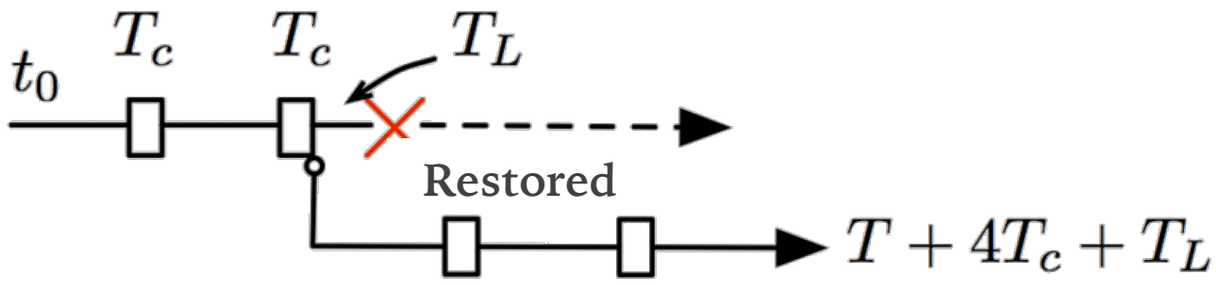
$Z_k \rightarrow$ Random variable measuring time to revocation

$P_k \rightarrow$ Probability job gets revoked before completion

$$\text{Cost} = \frac{E[\text{Price}_k]}{E[\text{Time}_k]} = \frac{[(1 - P_k) * T \quad + \quad P_k * (E(Z_k) + T_M)] * \text{spot-price}}{(1 - P_k) * T \quad + \quad P_k * E(Z_k)}$$

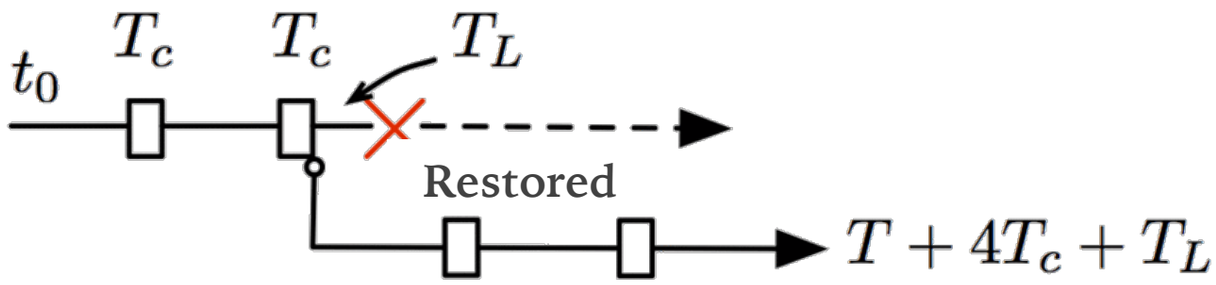
FAULT TOLERANCE (2/3)

Proactive Checkpoint



FAULT TOLERANCE (2/3)

Proactive Checkpoint



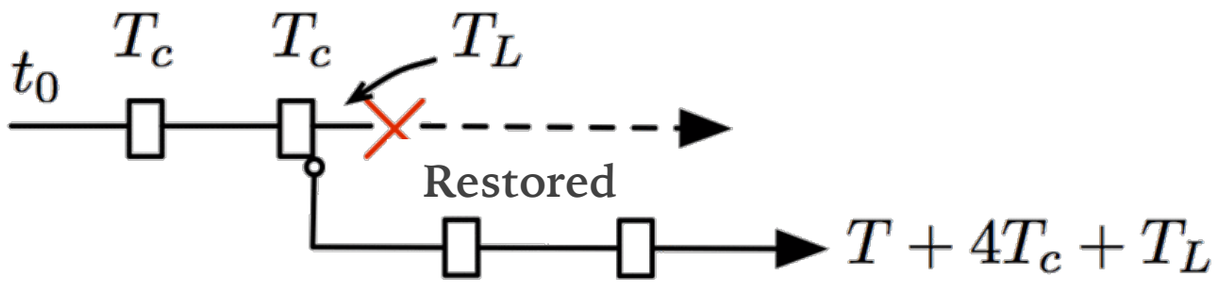
$$T_C \propto \frac{\text{size of memory + local disk}}{\text{remote disk bandwidth}}$$

$$T_L \propto (\tau/2)$$

τ: checkpoint frequency

FAULT TOLERANCE (2/3)

Proactive Checkpoint



$$T_C \propto \frac{\text{size of memory + local disk}}{\text{remote disk bandwidth}}$$

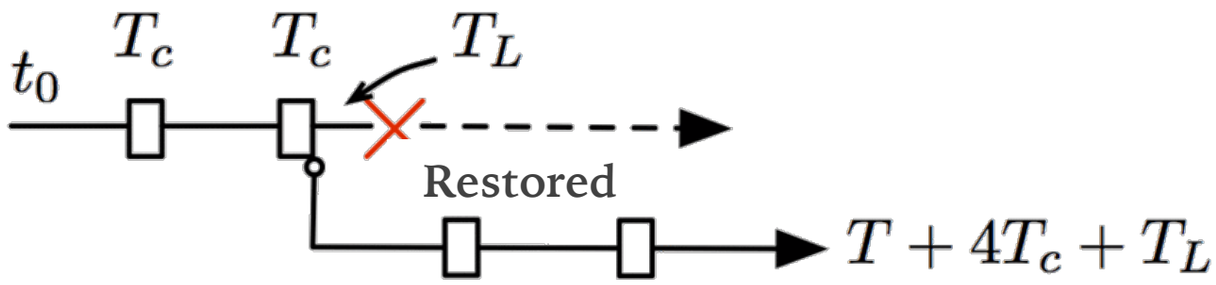
$$T_L \propto (\tau/2)$$

τ: checkpoint frequency

$$\text{Total Overhead} = (T/\tau) * T_c + T_L$$

FAULT TOLERANCE (2/3)

Proactive Checkpoint



$$T_C \propto \frac{\text{size of memory + local disk}}{\text{remote disk bandwidth}}$$

$$T_L \propto (\tau/2)$$

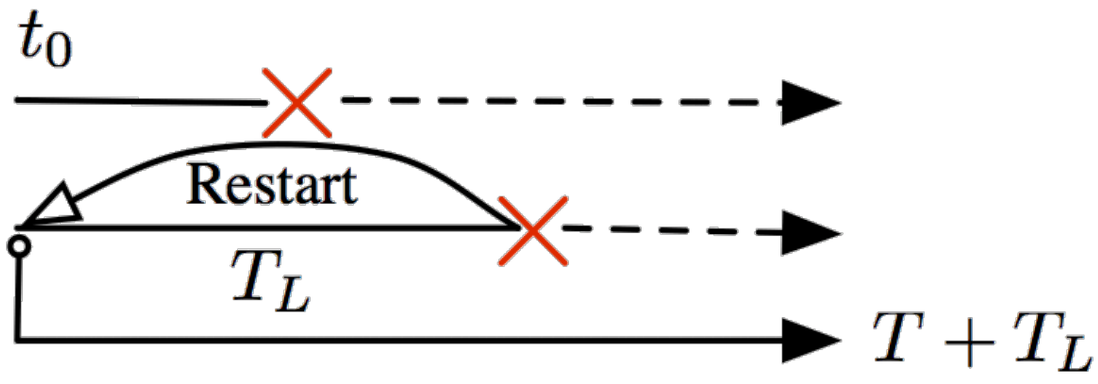
τ: checkpoint frequency

$$\text{Total Overhead} = (T/\tau) * T_c + T_L$$

Cost overhead is primarily a function of job's resource usage

FAULT TOLERANCE (3/3)

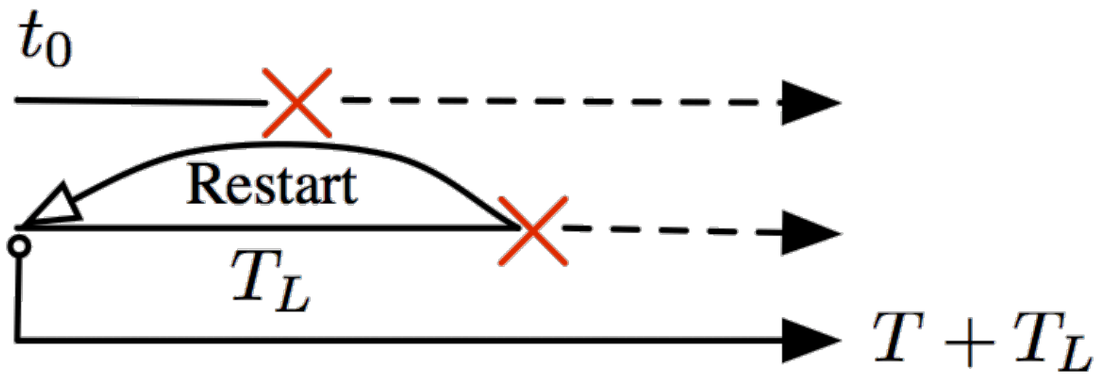
Spot Replication



$T_L \propto$ market volatility

FAULT TOLERANCE (3/3)

Spot Replication



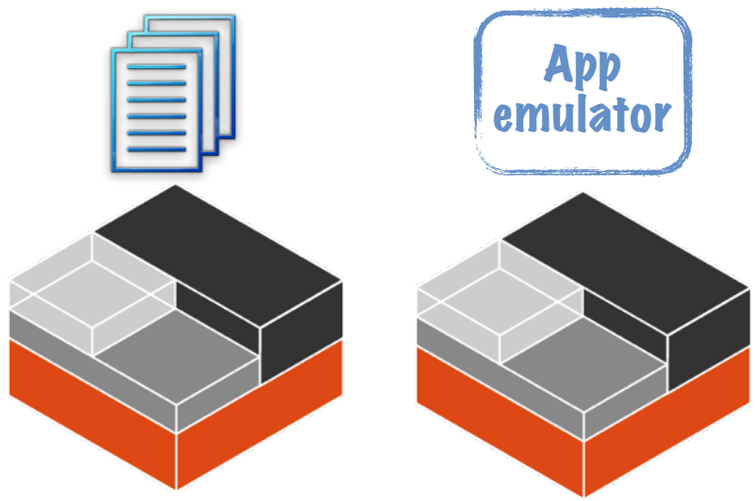
$T_L \propto$ market volatility

Total Overhead $\propto$ (Replication factor, T_L)

Cost overhead is primarily a function of market characteristics

EVALUATION (1/3)

SpotOn Prototype



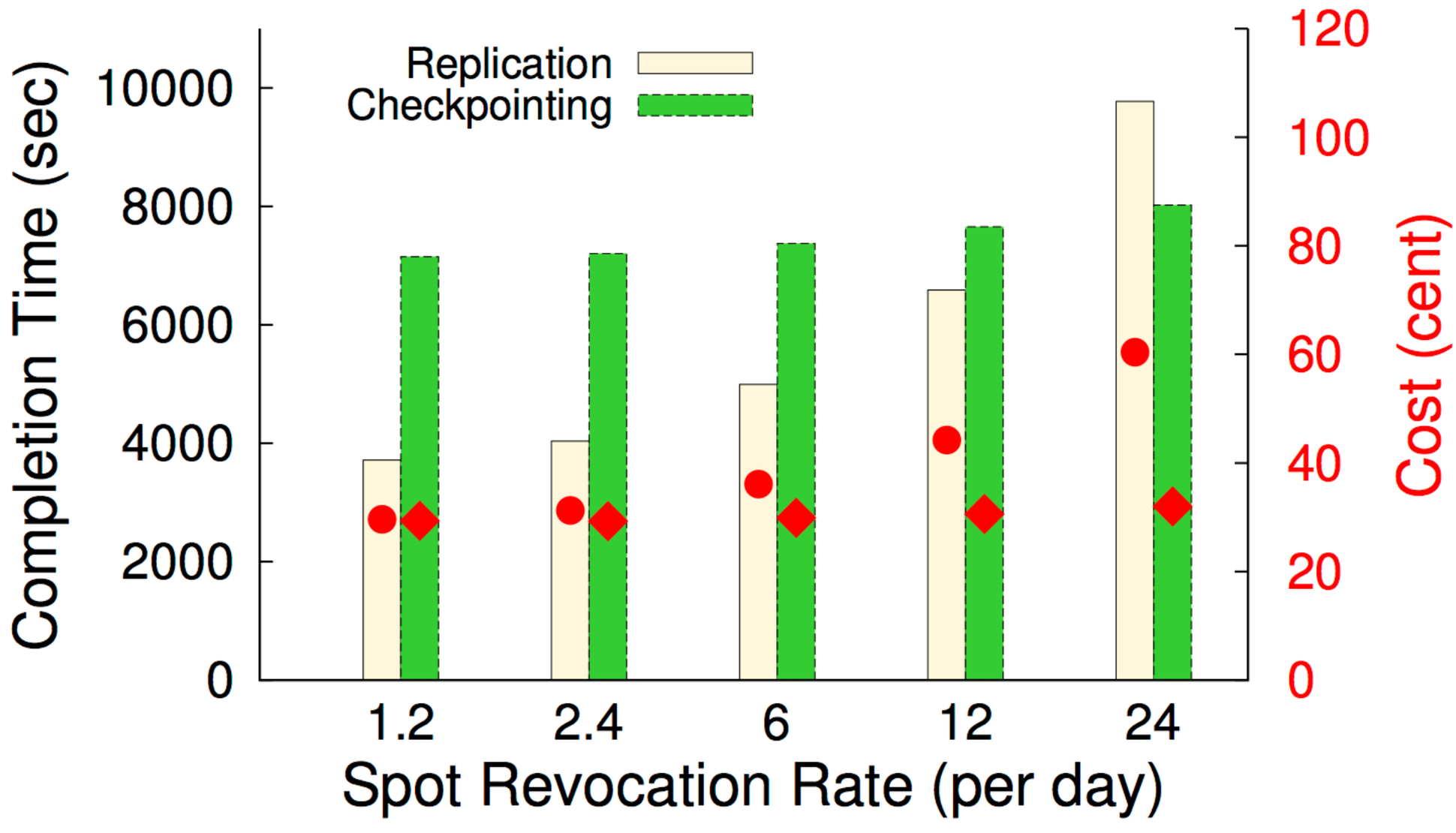
- Built on ***Linux Containers*** for efficient checkpointing / migration
- ***App Emulator*** to create synthetic jobs with varying resource usage

EVALUATION (2/3)

Effect of Application and Spot Market Characteristics

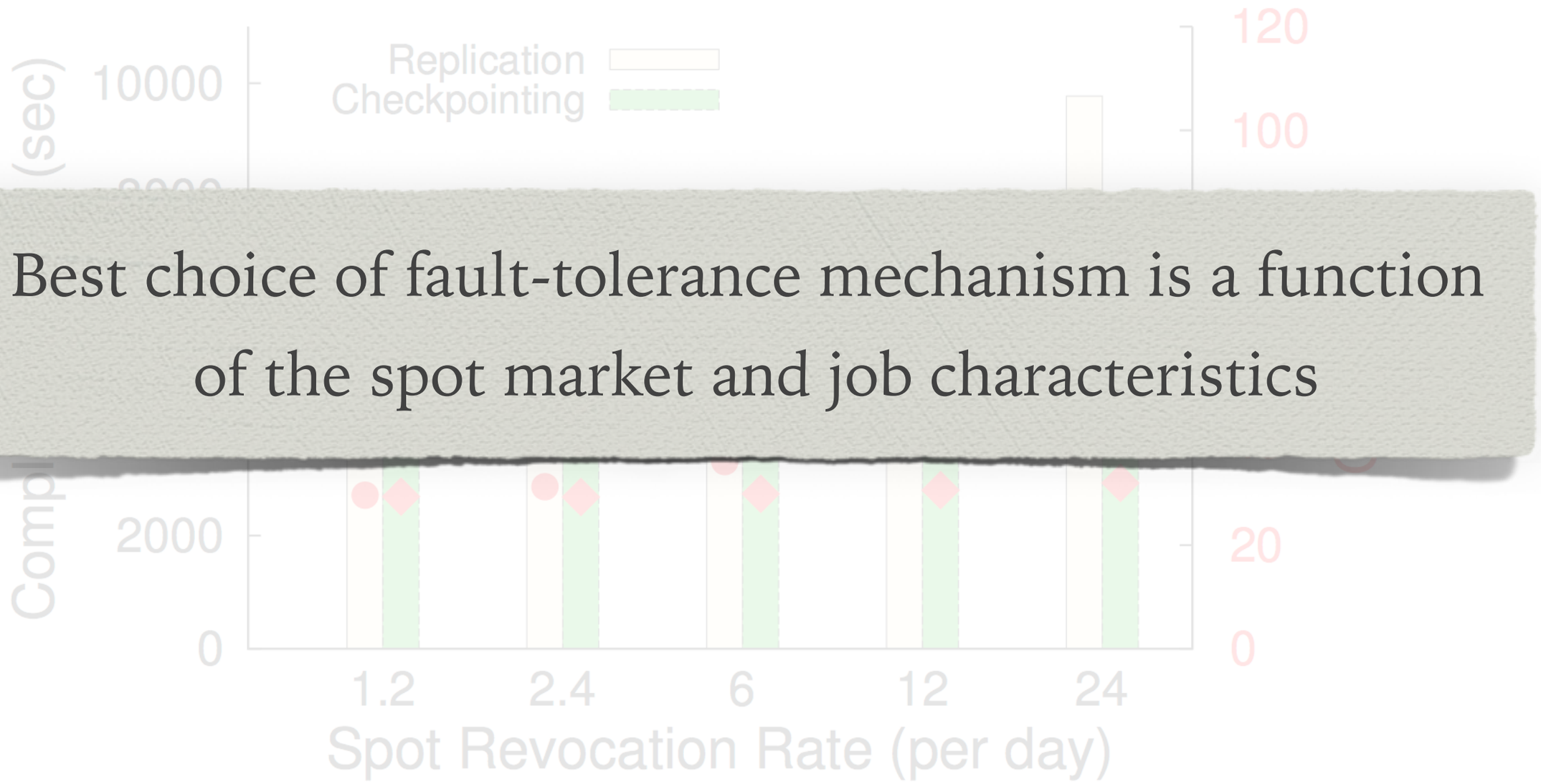
EVALUATION (2/3)

Effect of Application and Spot Market Characteristics



EVALUATION (2/3)

Effect of Application and Spot Market Characteristics



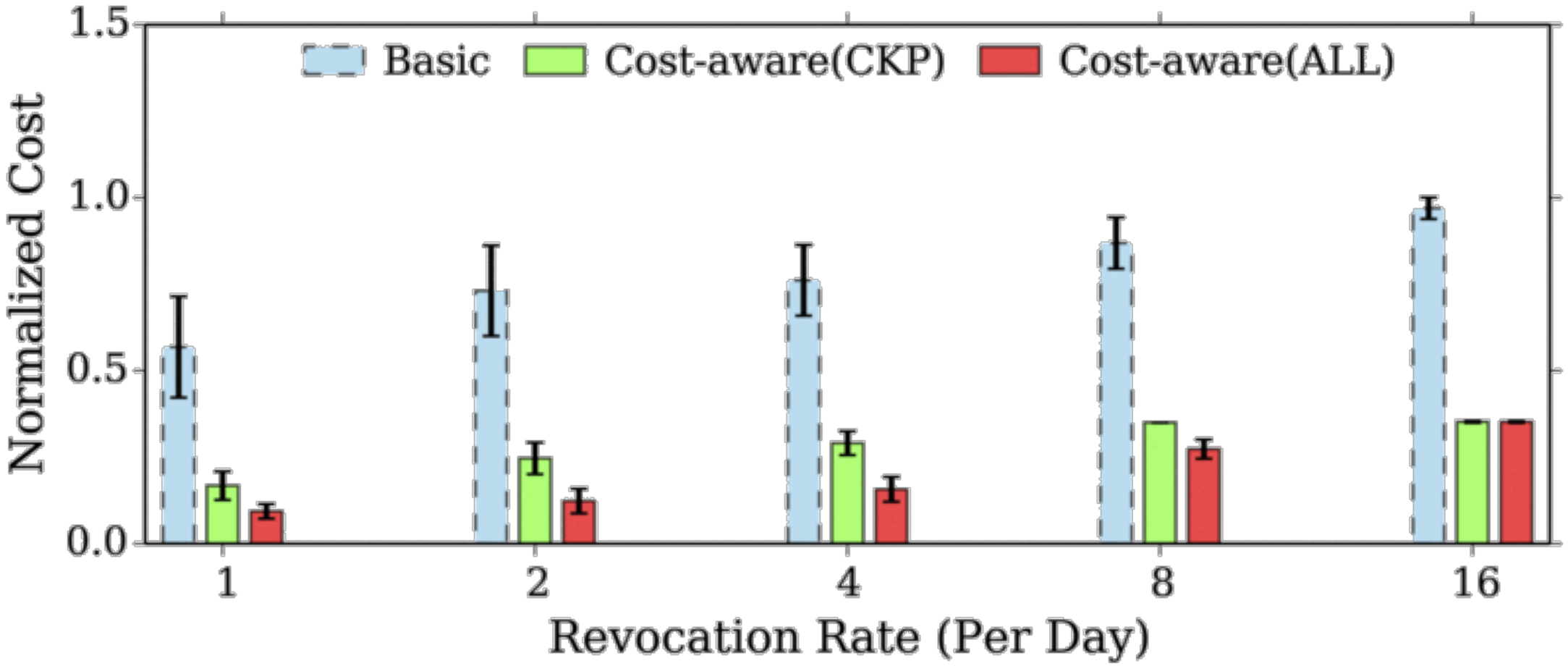
Best choice of fault-tolerance mechanism is a function of the spot market and job characteristics

EVALUATION (3/3)

Effect of Cost-aware Selection

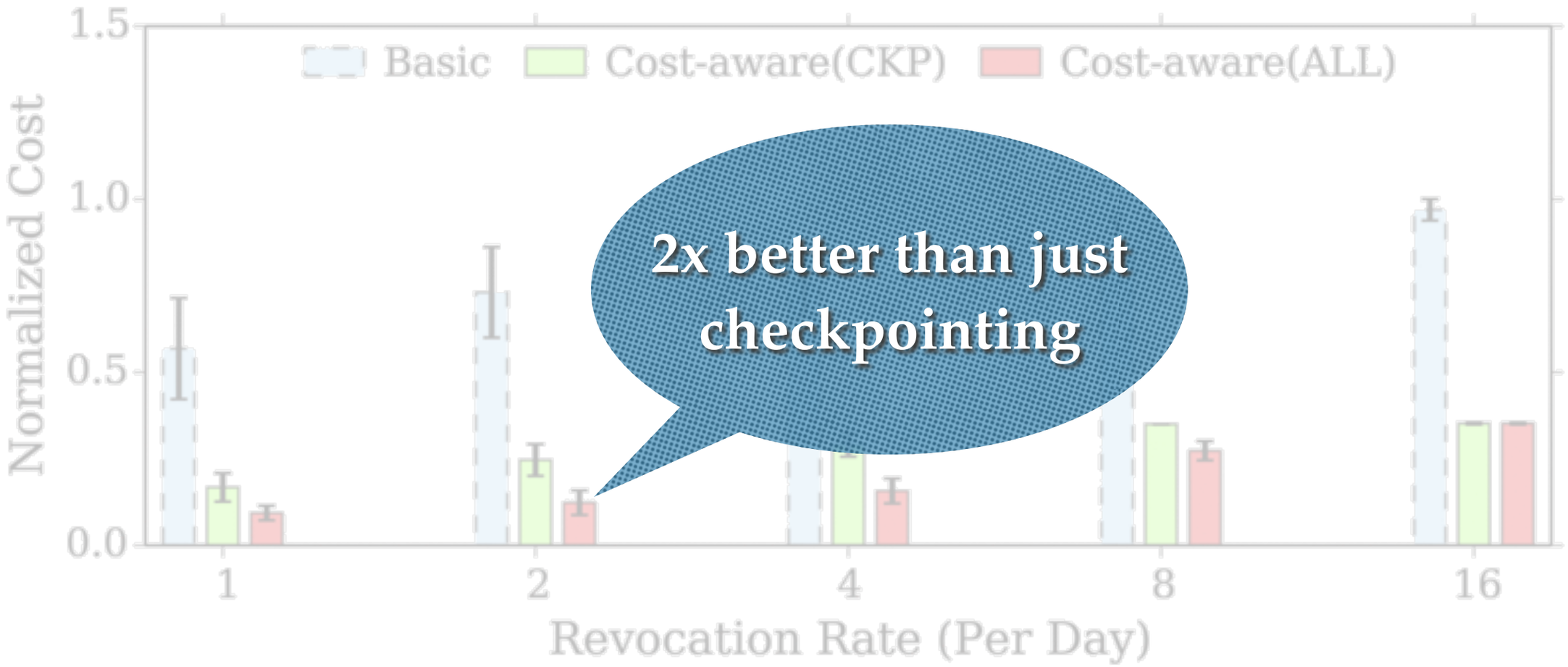
EVALUATION (3/3)

Effect of Cost-aware Selection



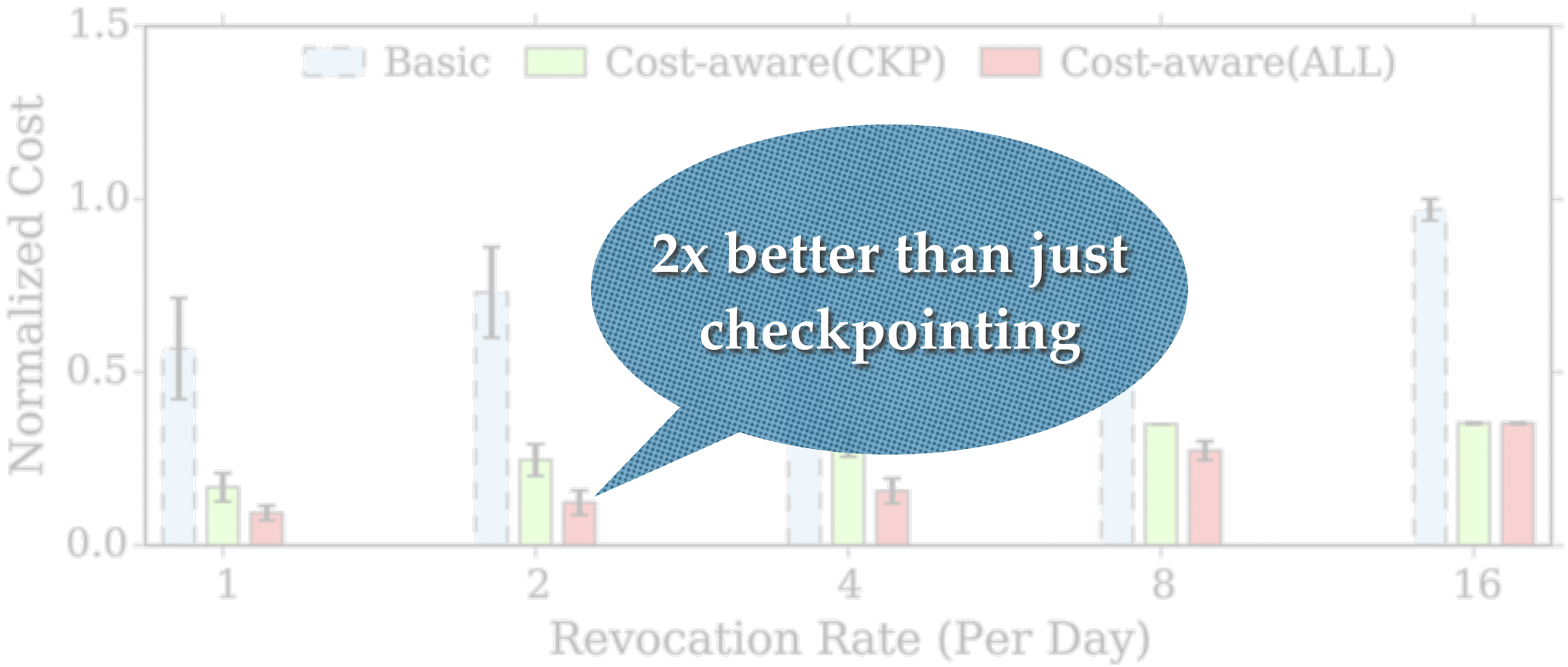
EVALUATION (3/3)

Effect of Cost-aware Selection



EVALUATION (3/3)

Effect of Cost-aware Selection



On Google cluster trace, Cost-aware selection achieved 91.9% savings with little impact on performance

CONCLUSION

- Spot markets offer **arbitrage** opportunities
- SpotOn manages application and market complexities
- We model **fault tolerance** and propose a **selection algorithm**
- **Prototype** on Amazon EC2
- Achieves **~90%** cost savings

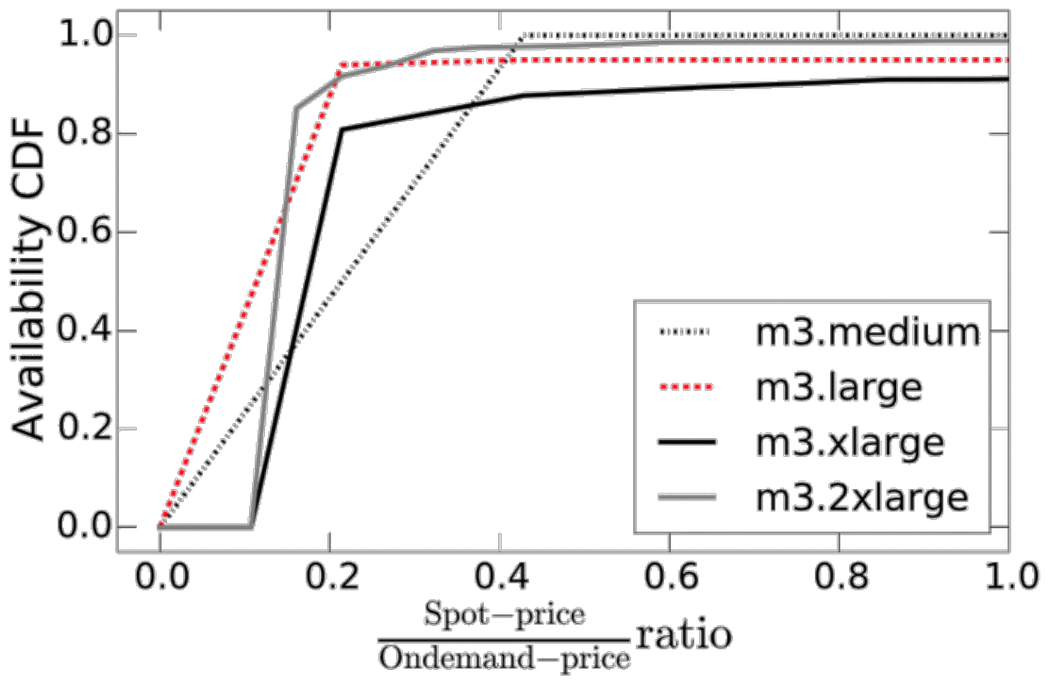
CONCLUSION

- Spot markets offer **arbitrage** opportunities
- SpotOn manages application and market complexities
- We model **fault tolerance** and propose a **selection algorithm**
- **Prototype** on Amazon EC2
- Achieves **~90%** cost savings

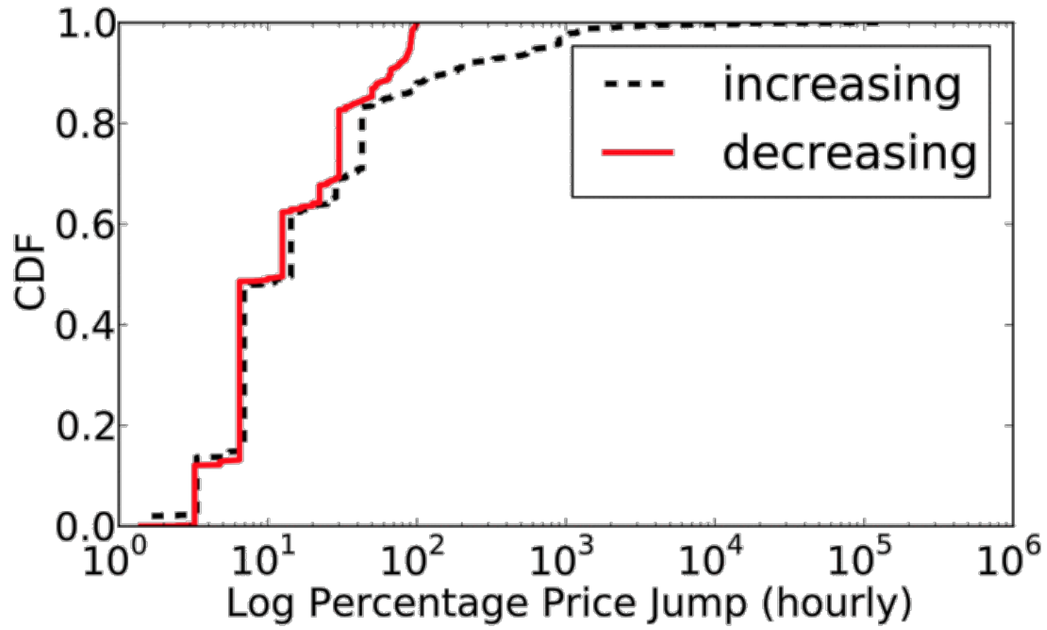
Thank you!

BACKUP SLIDES

BIDDING

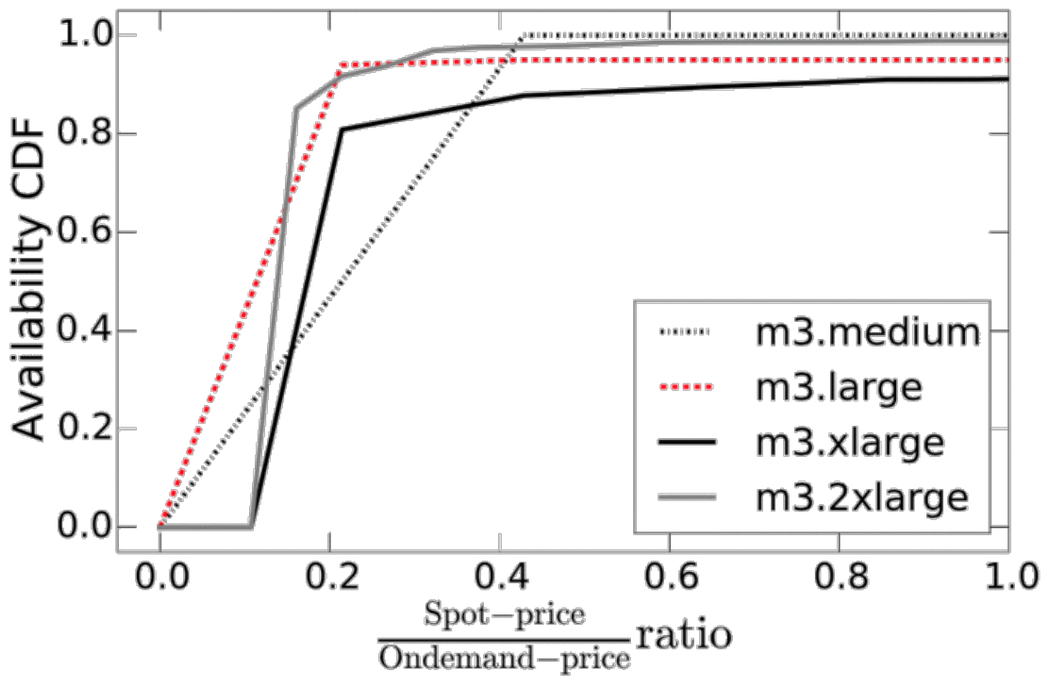


Spot price distribution has long tail

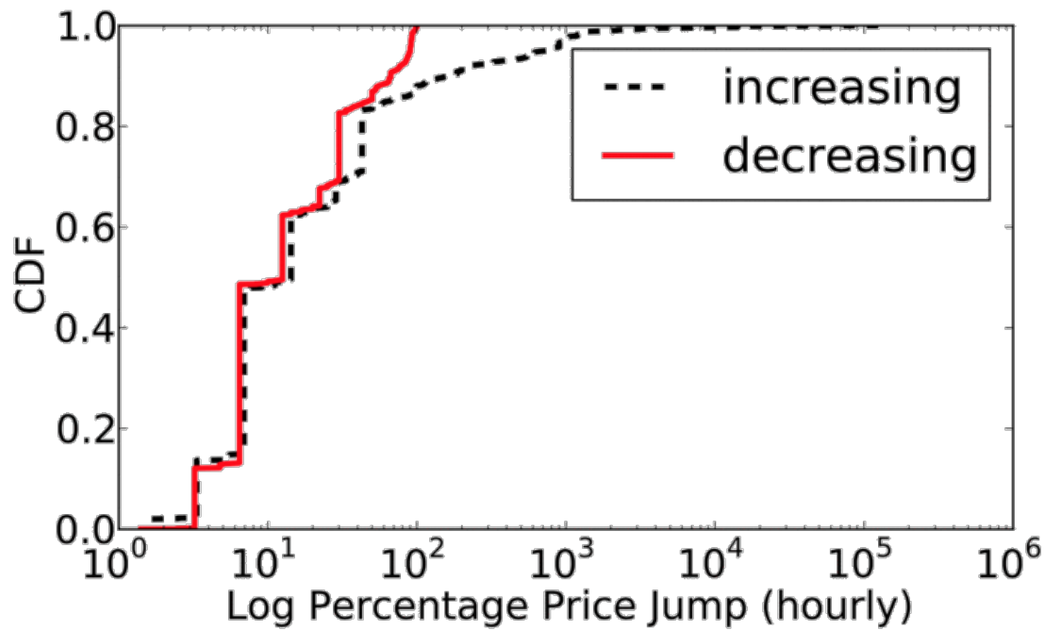


Spot price changes are peaky

BIDDING



Spot price distribution has long tail



Spot price changes are peaky

- Bidding does affect volatility but ***not drastically***

- SpotOn always bids at the on-demand price

  If spot price goes above on-demand, SpotOn would choose on-demand