University of Massachusetts Amherst

Is Sharing Caring? Analyzing the Incentives for Shared Cloud Clusters

Talha Mehboob, Noman Bashir, Michael Zink, and David Irwin

University of Massachusetts Amherst

Presenter: **John Murray**

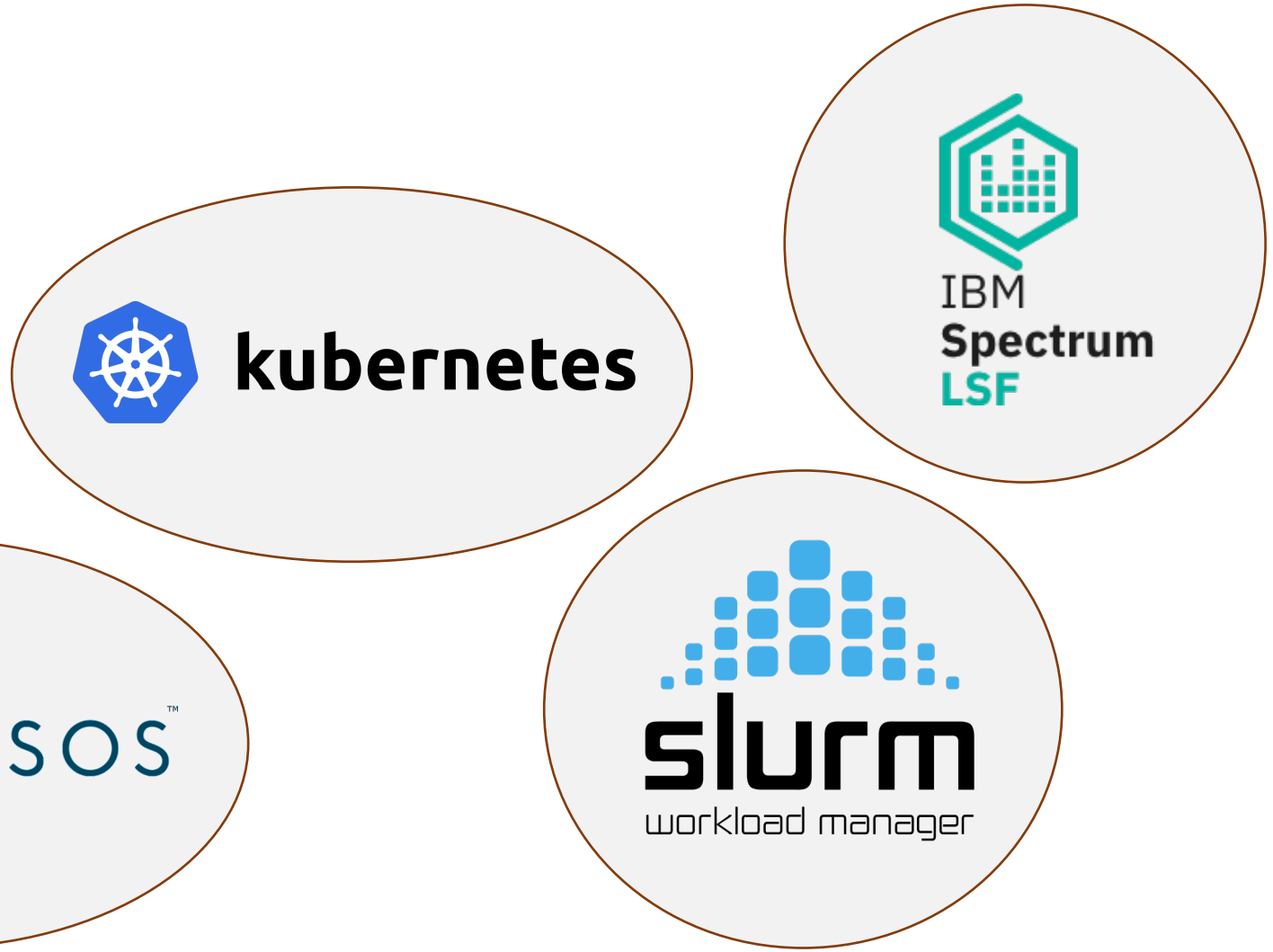
Advantages of Shared Clusters

- Reduced **cost**
- Complete **control**
- Optimized job **scheduling**

MGHPCC: Massachusetts Green High Performance Computing Center

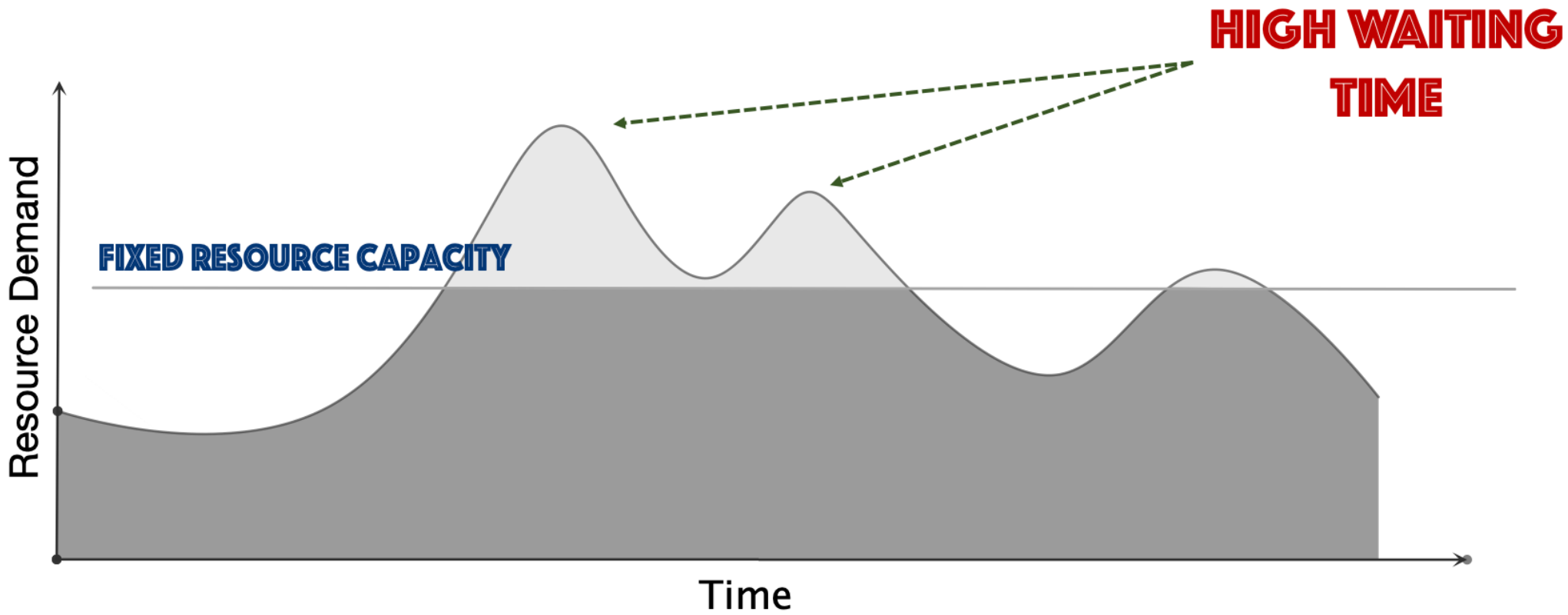
Advantages of Shared Clusters

- Reduced **cost**
- Complete **control**
- Optimized job **scheduling**



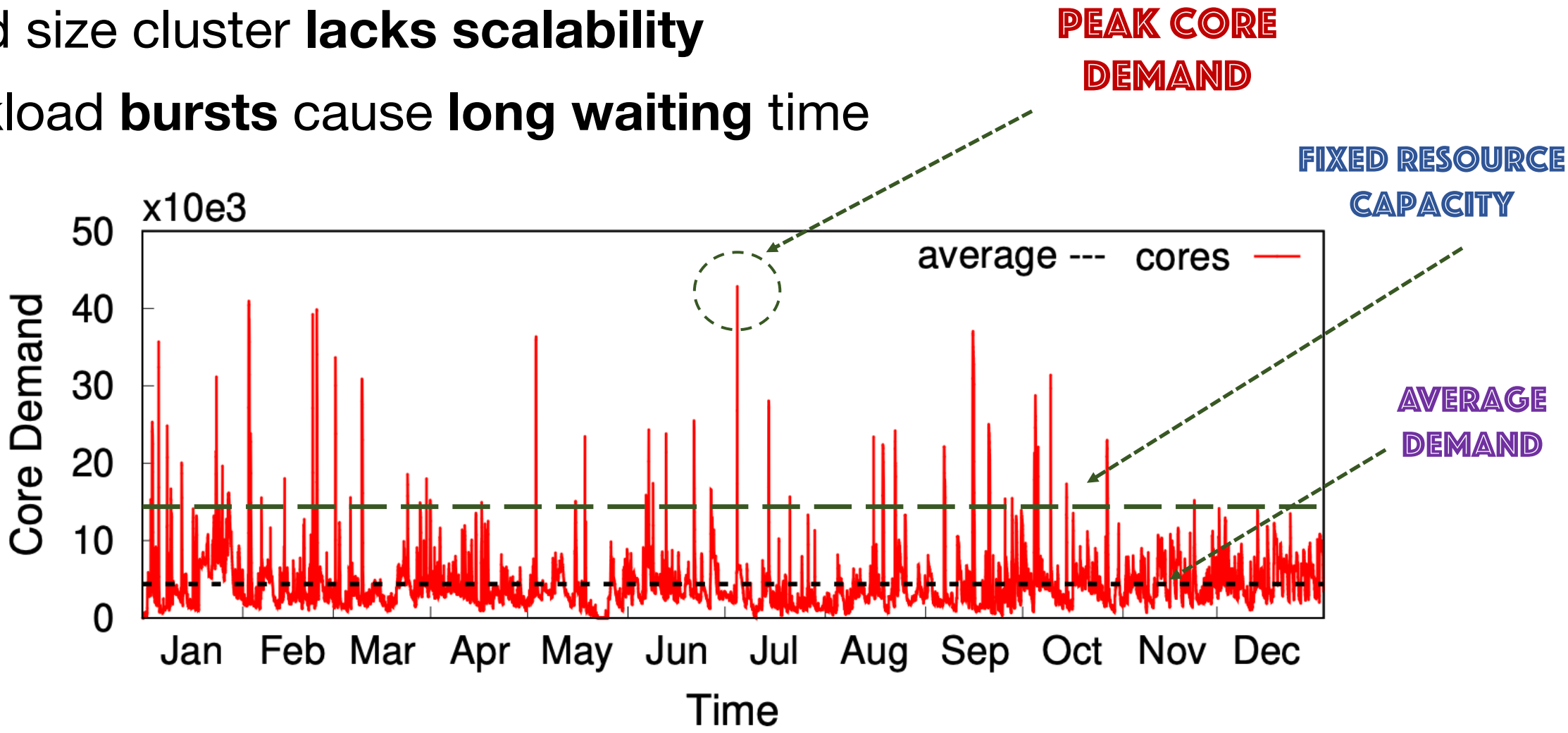
Challenges

- Fixed size cluster **lacks scalability**
- Workload **bursts** cause **long waiting** time



Challenges

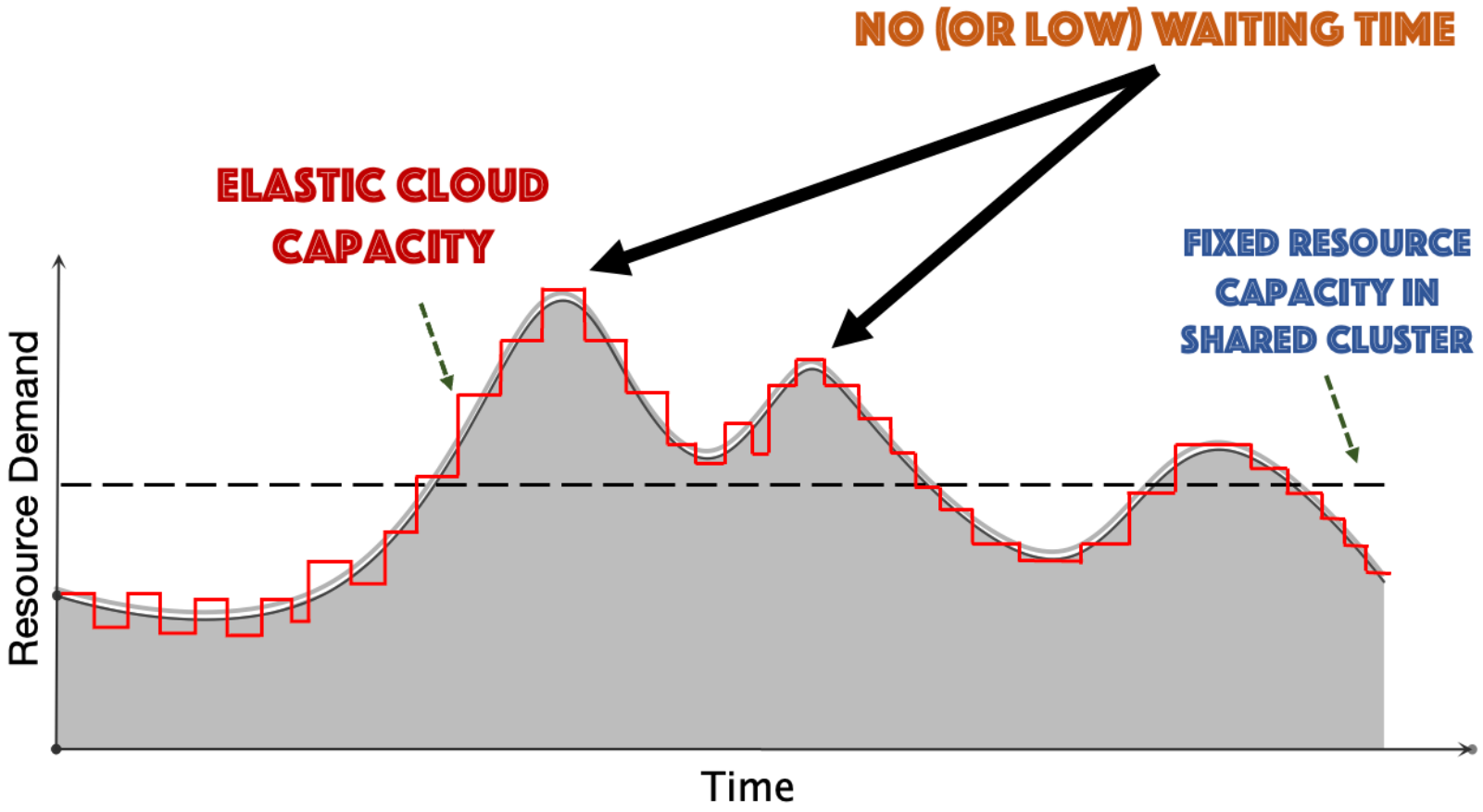
- Fixed size cluster **lacks scalability**
- Workload **bursts** cause **long waiting** time



Migrating to the Cloud

- **Addresses many challenges**

- **Cloud benefits**
  - Elasticity
  - Scalability
  - Low cost
  - Pay-as-you-go billing



Key Question – Cost Effective?

- **On-demand vs fixed (reserved)**
  - On-demand – **high cost**, **zero waiting time**
  - Fixed (reserved) – **lower cost**, **higher waiting time**
    - *Assumes high utilization*

Plan	Price / hour	Discounts
On-Demand	$ 2.4576	-
Reserved (1 Year)	$ 1.062	37%
Reserved (3 Years)	$ 1.548	57%

Key Question – Cost Effective?

University of Massachusetts Amherst

- **On-demand vs fixed (reserved)**
  - On-demand – **high cost**, **zero waiting time**
  - Fixed (reserved) – **lower cost**, **higher waiting time**
    - *Assumes high utilization*

*Should users **participate** in **shared cloud cluster**, or should they **defect**?*

Outline

1. **Trace analysis**
2. **Provisioning policies**
3. **Pricing policies**
4. **Evaluation**

Trace Analysis: Overview

- **Trace characteristics**
  - Large scale cluster (14k cores)
  - Longitudinal trace (8 Years)
  - 67 million jobs from ~1800 users

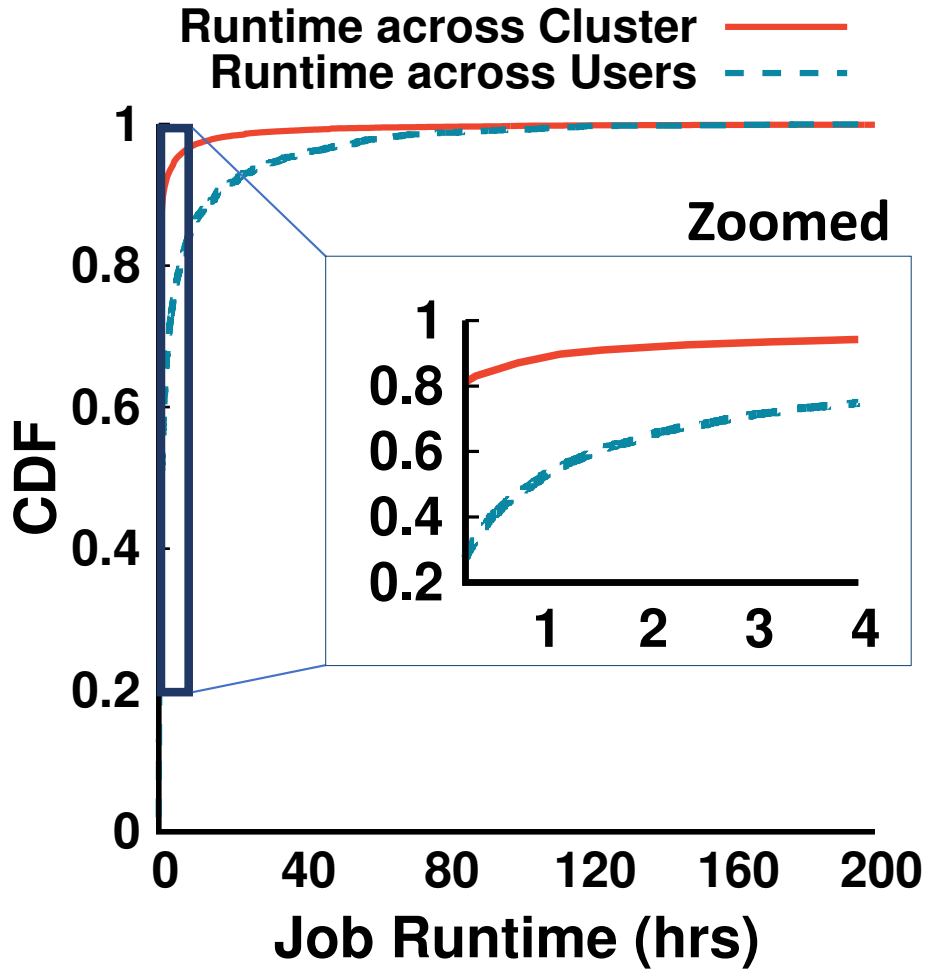
- **Key factors**
  - Job runtimes
  - Workload burstiness
  - Long-term patterns

MGHPCC: Massachusetts Green High Performance Computing Center

Trace Analysis: Job Runtime

- ***Most*** jobs have short (<15m) runtimes
- ***Many*** users have short (<15m) runtimes

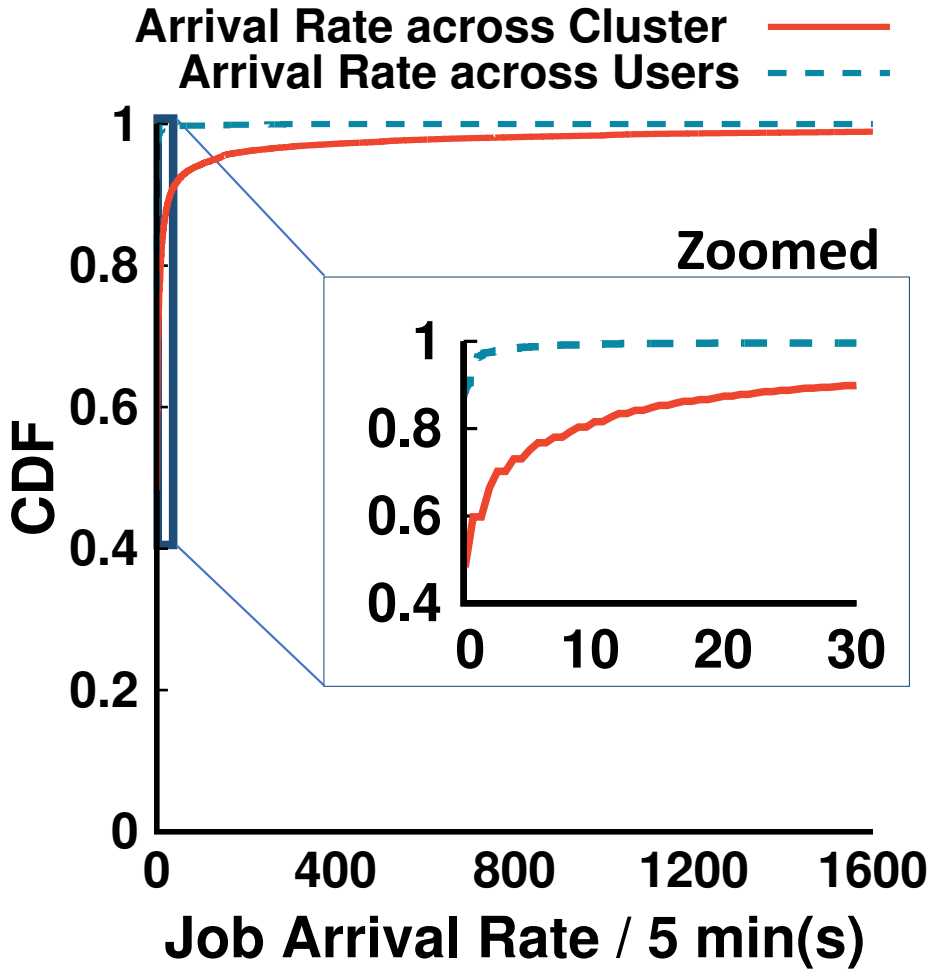
Shorter runtimes are **more sensitive** to **waiting** time



Trace Analysis: Workload Burstiness

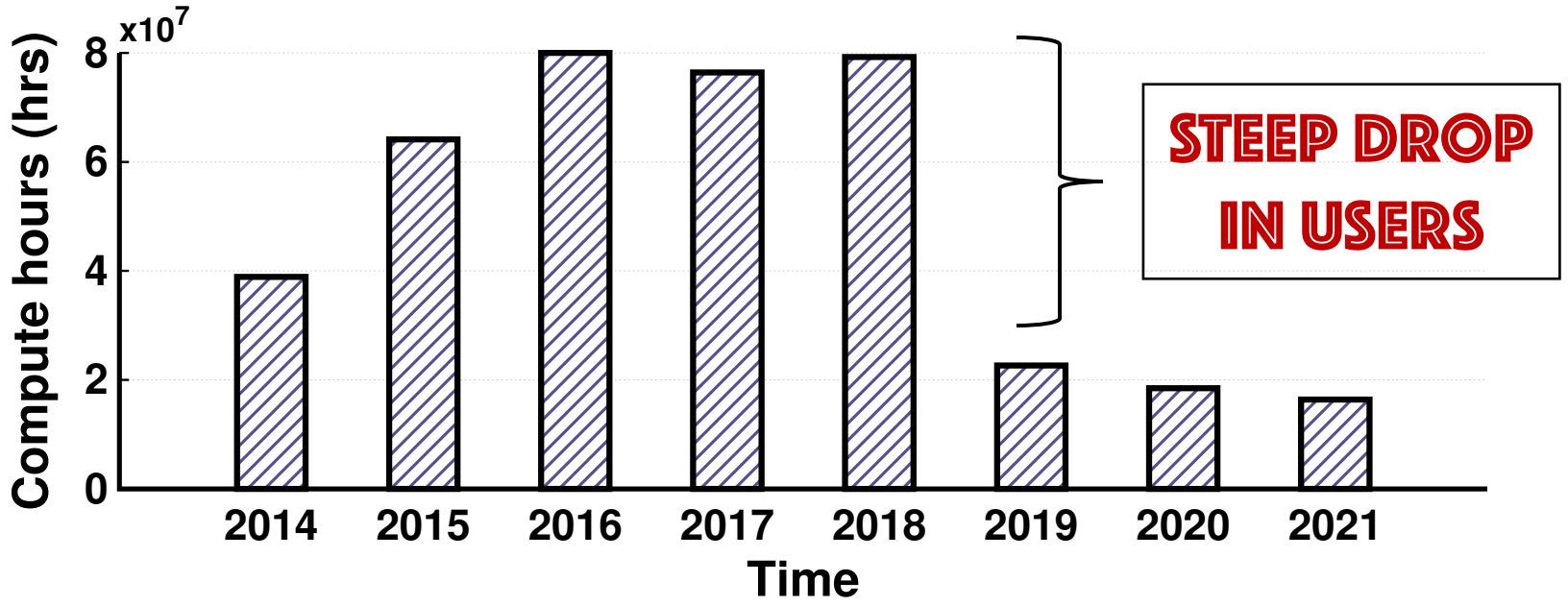
- ***Most*** user job bursts small
- ***Some*** cluster job bursts (very) large

Job bursts can cause very **high waiting** times



Trace Analysis: Usage Variations

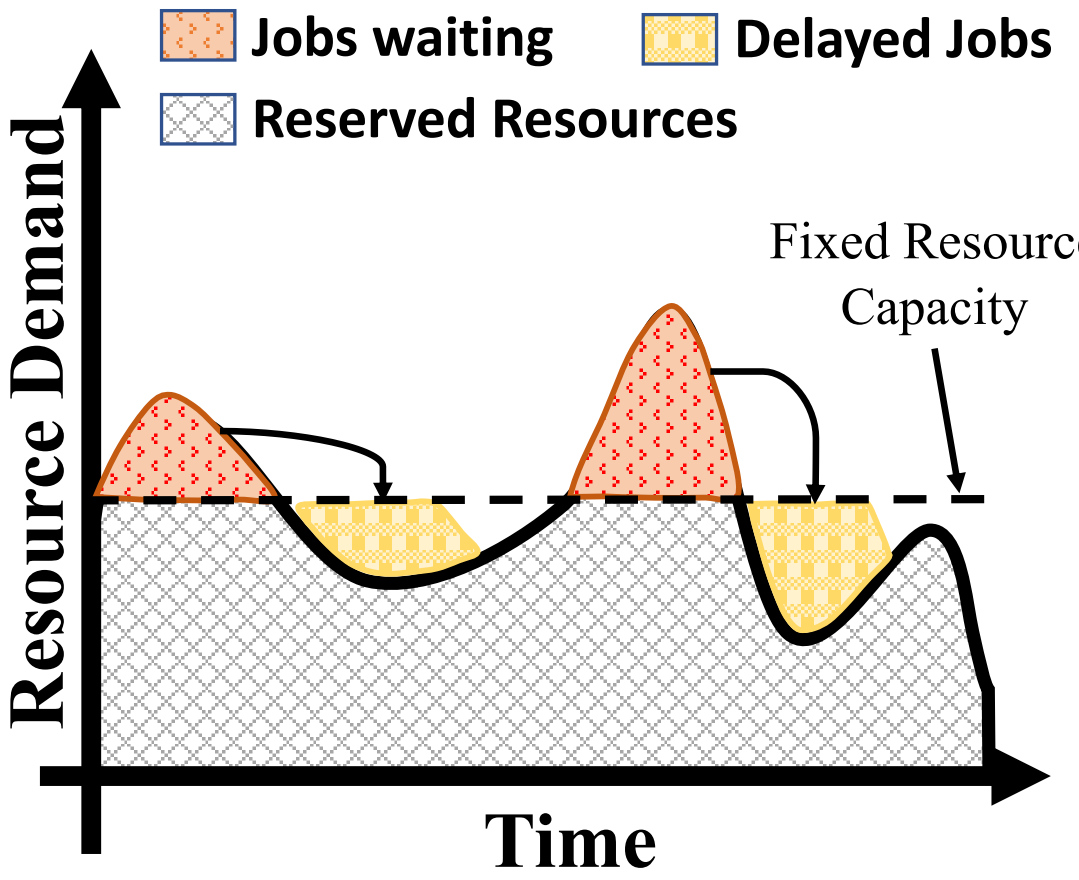
- **Exhibits large year-to-year usage variations**



Forecasting optimal fixed resource provisioning is hard

Provisioning Policies: Lift and Shift

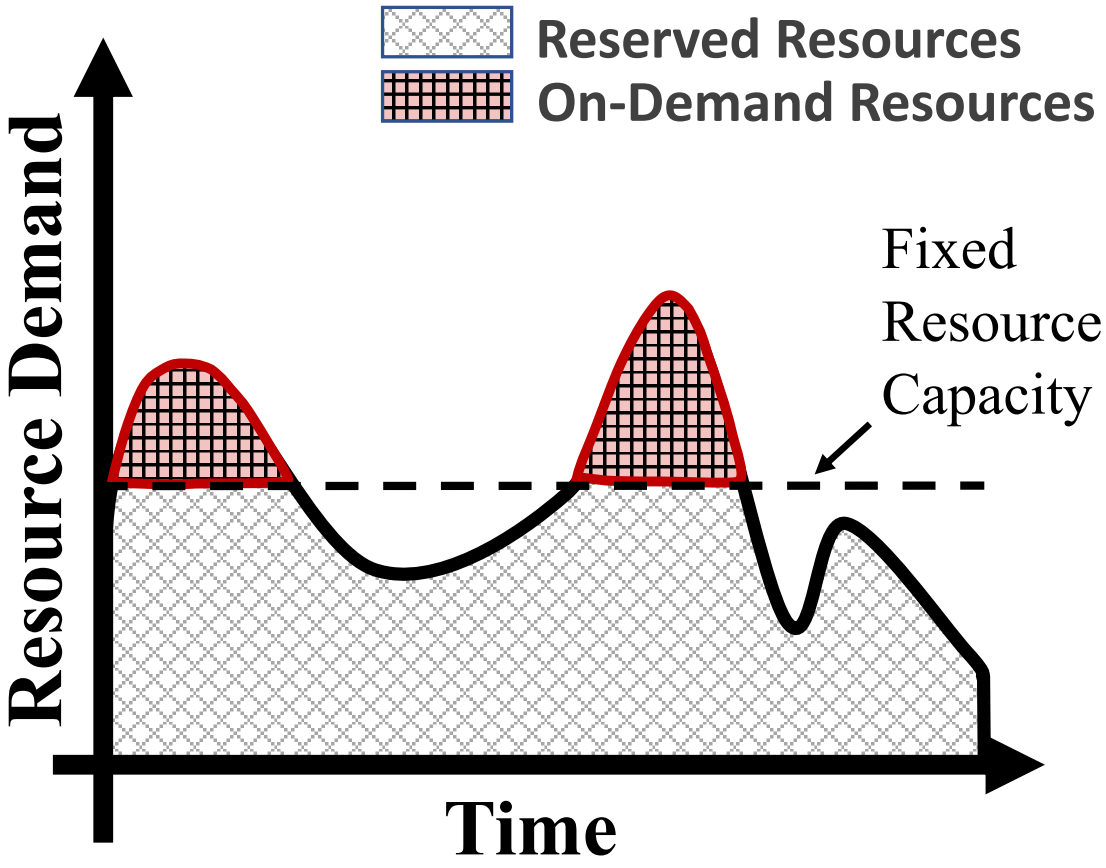
- **Basic approach**
  - Move fixed on-prem to cloud
  - No on-demand; only reserved
- **Benefits**
  - Reserving cheaper
- **Drawbacks**
  - *Recall*: optimizing fixed is hard
  - *Recall*: high waiting due to burstiness



Lift & Shift

Provisioning Policies: Cloud Bursting

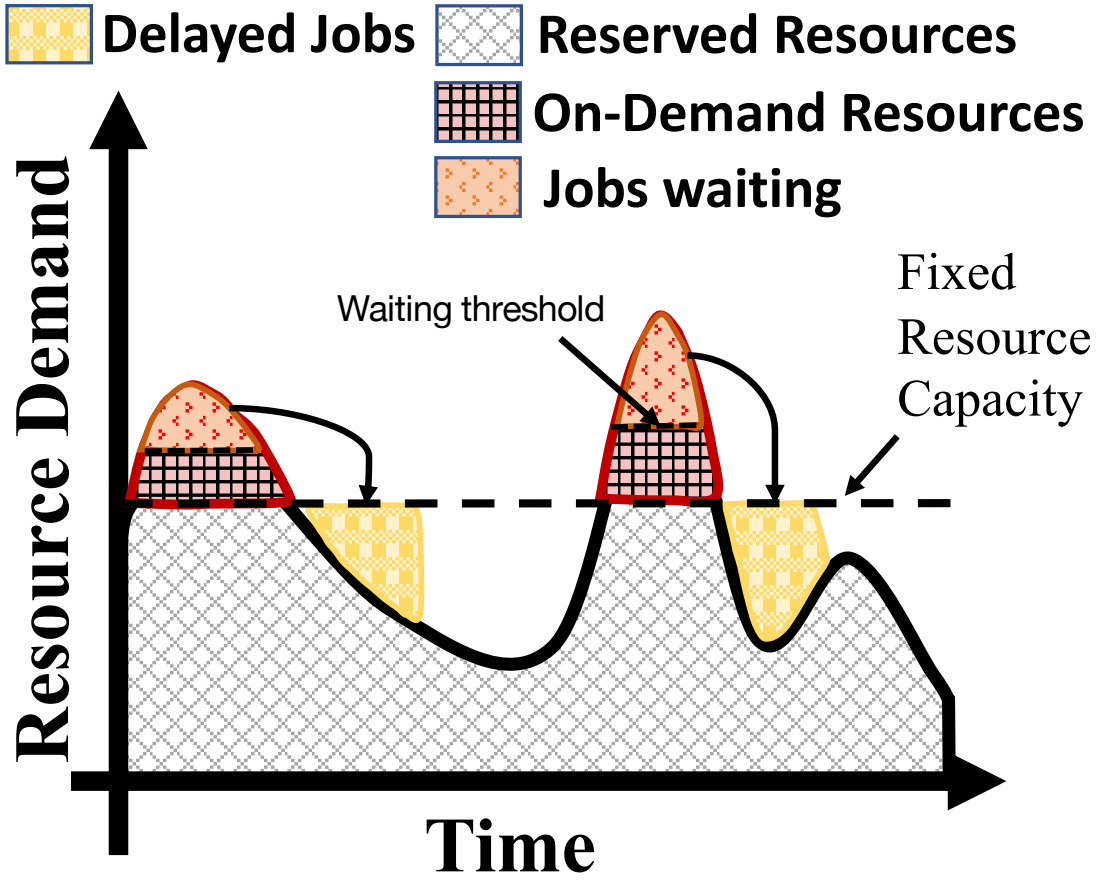
- **Hybrid approach w/o waiting**
  - Mix of reserved and on-demand
  - Use on-demand when fixed fully utilized
- **Benefits**
  - No waiting time
- **Drawbacks**
  - *Recall*: optimizing fixed is still hard
  - *Recall*: workloads bursty
    - Use many on-demand resources
    - Leads to high cost



Cloud bursting without waiting

Provisioning Policies: Cloud Bursting

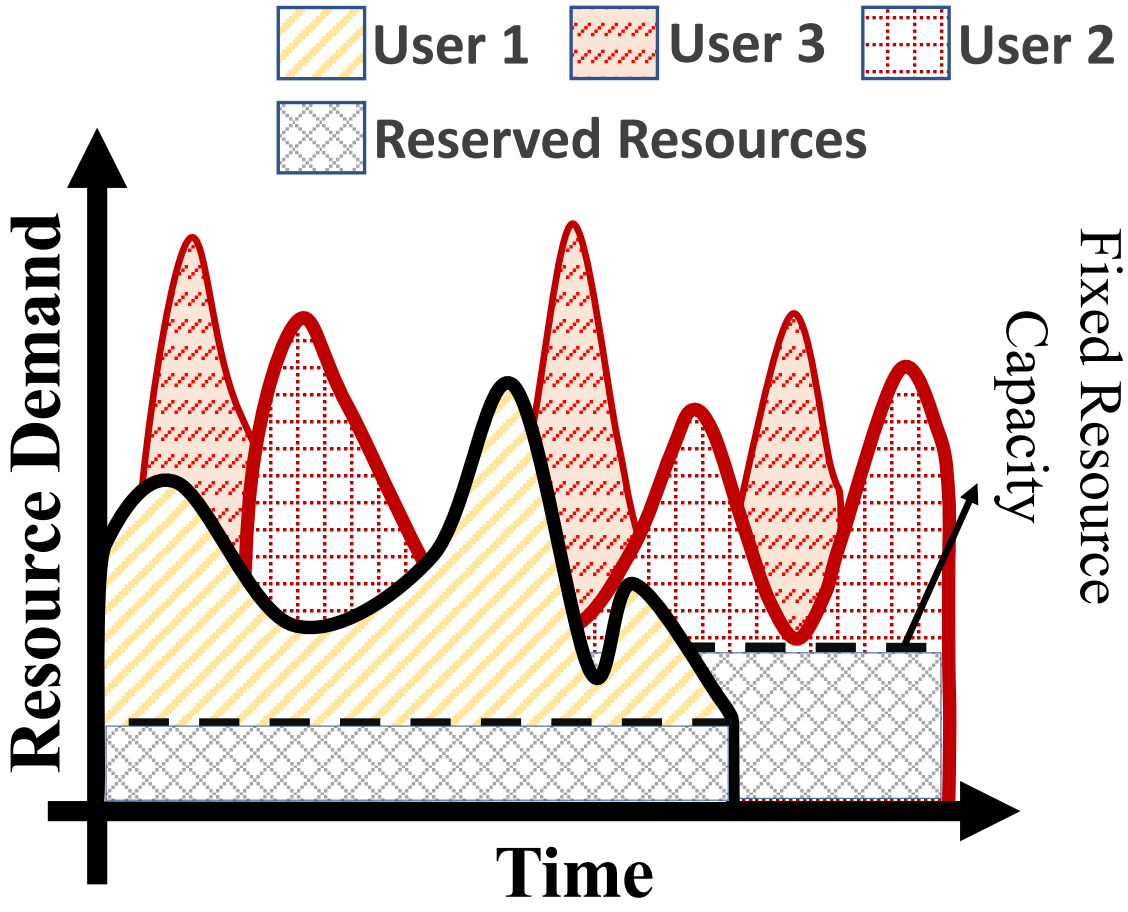
- **Hybrid approach w/ waiting**
  - Define waiting time threshold *t*
  - Use on-demand after waiting time *t*
  - Introduces cost-waiting time tradeoff
- **Benefits**
  - Configurable cost-waiting time
- **Drawbacks**
  - Tradeoffs not always attractive
  - Low cost == Very high waiting time



Cloud bursting with waiting

Provisioning Policies: Flying Solo

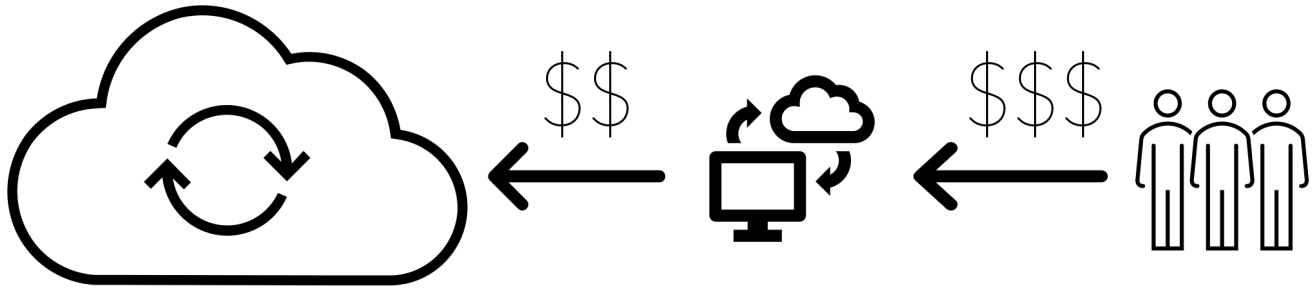
- **Basic approach**
  - Users *defect* from shared cluster
  - Rent cloud VMs individually
- **Benefits**
  - Configurable cost-wait time *per user*
  - Not affected by other users
- **Drawbacks**
  - No savings from statistical multiplexing



Flying solo

Pricing Policies: Shared Cluster

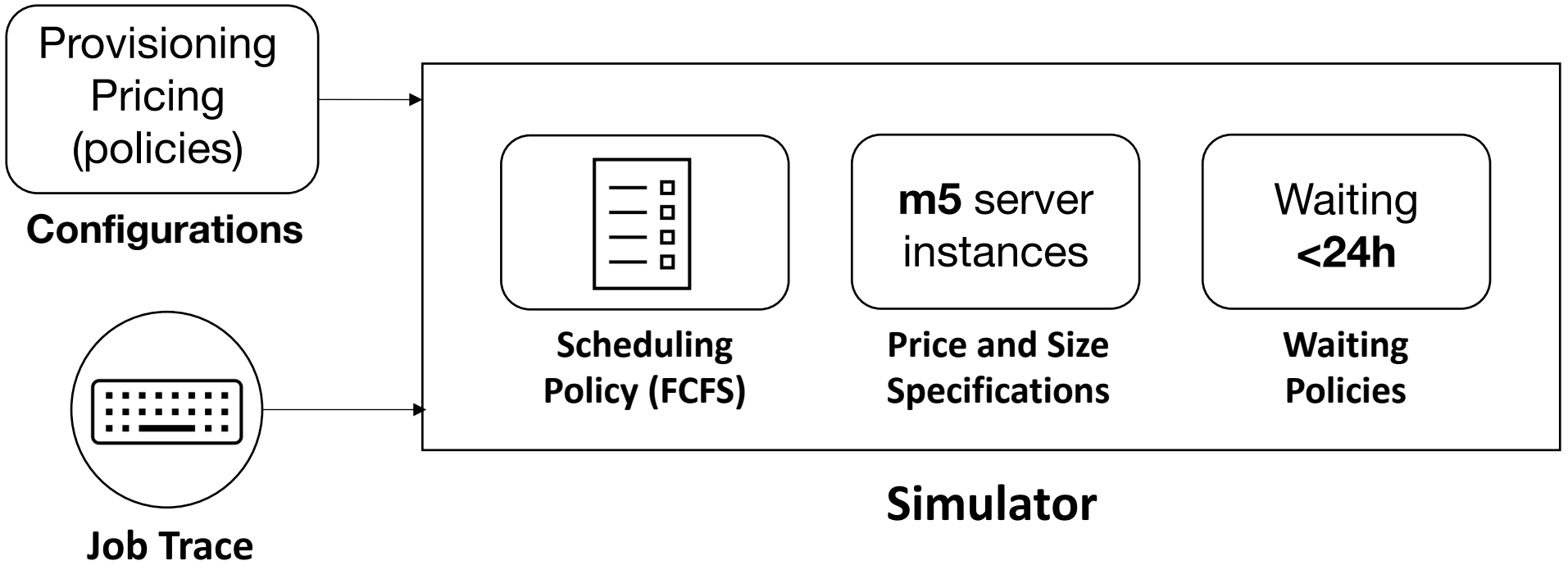
- **Socialist pricing model**
  - Charge single price for resources
  - Price = Amortized on-demand/reserved cost
- **Capitalist pricing model**
  - Charge different prices for on-demand/reserved



Cloud Charging Shared Clusters who charge People

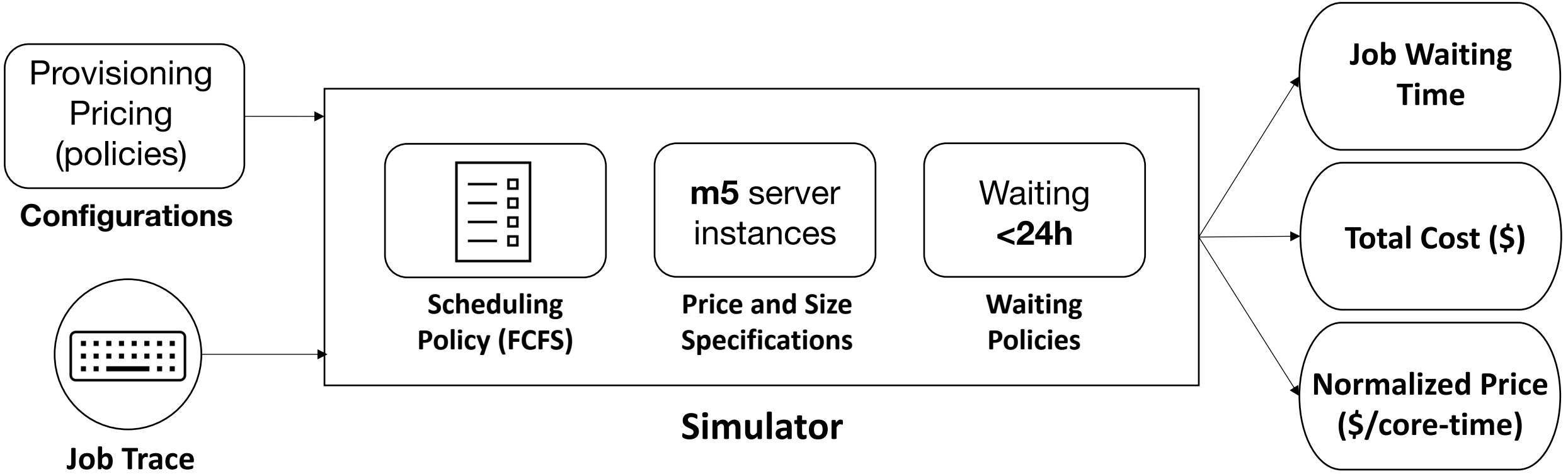
Evaluation: Implementation

- **Extended open-source job scheduling simulator**



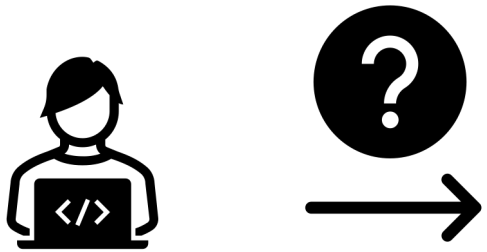
Evaluation: Implementation

- **Extended open-source job scheduling simulator**



Evaluation: Provisioning and Pricing

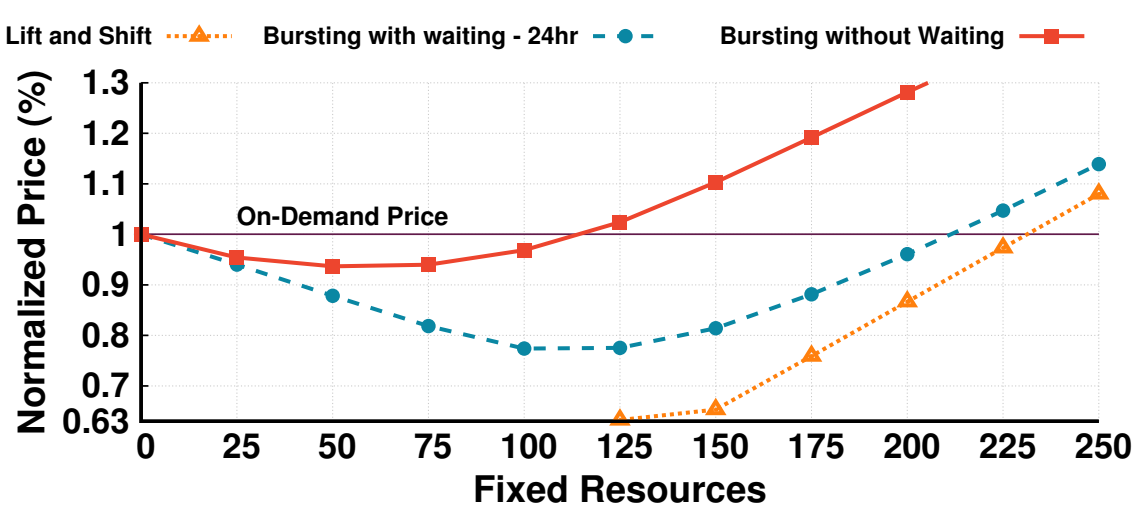
*Should users **participate** in **shared cloud cluster,** or should they **defect**?*



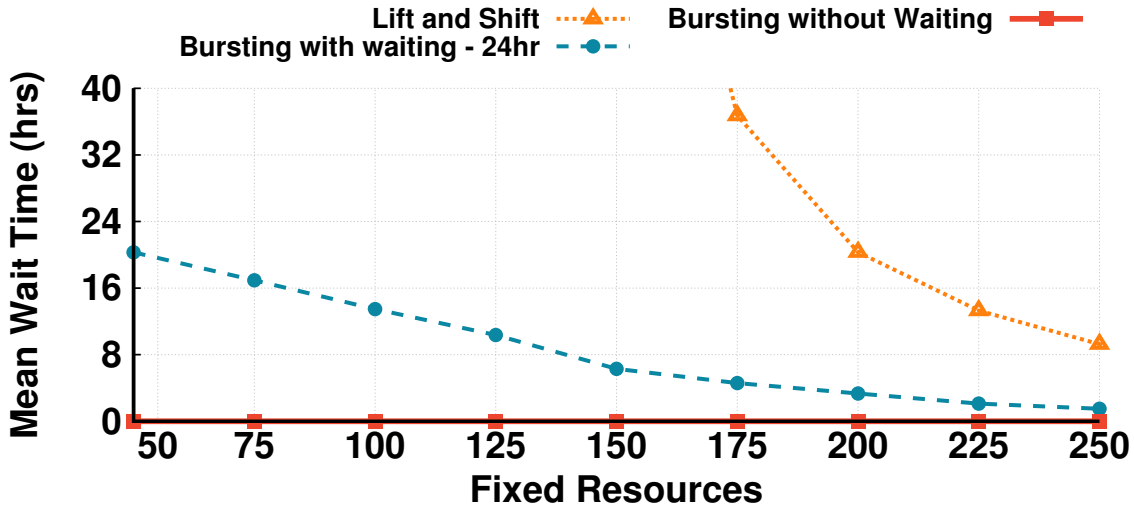
User

Shared Cloud Cluster

Evaluation: Overview



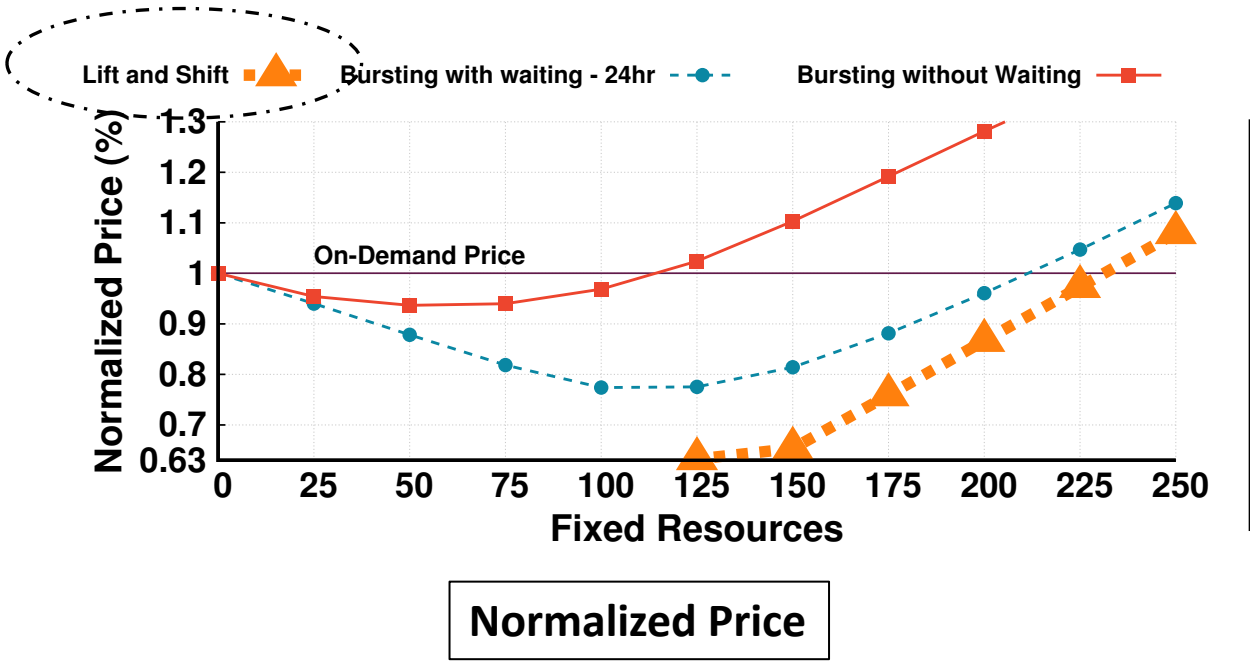
Normalized Price



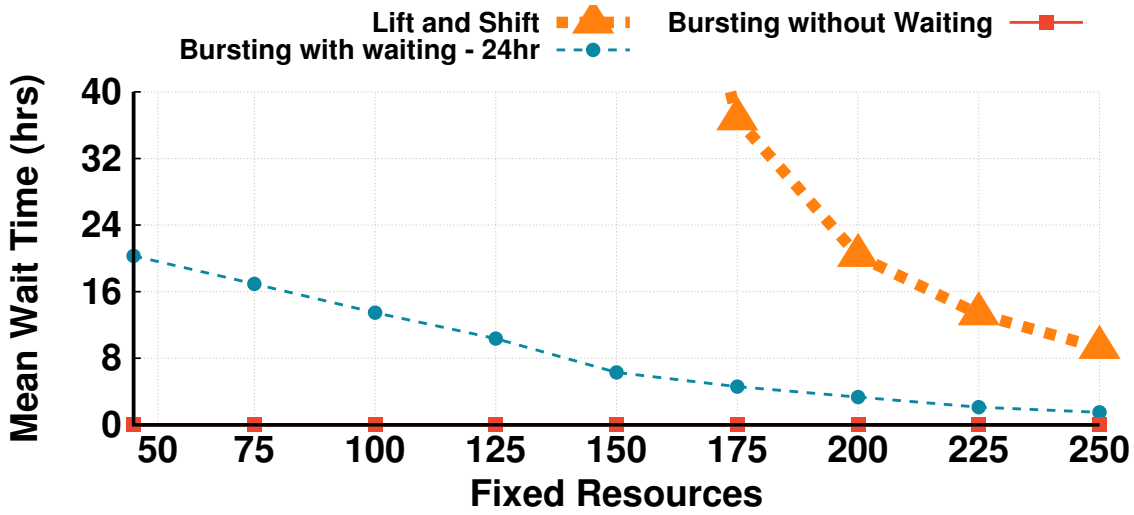
Waiting Time

- **Lower costs and waiting times are generally desirable**

Evaluation: Lift and Shift



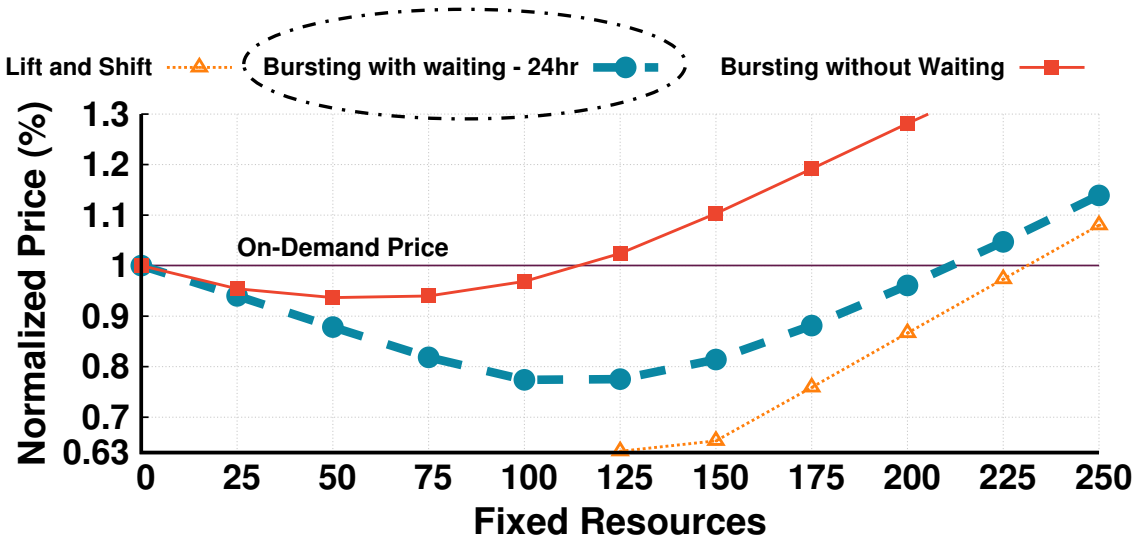
Normalized Price



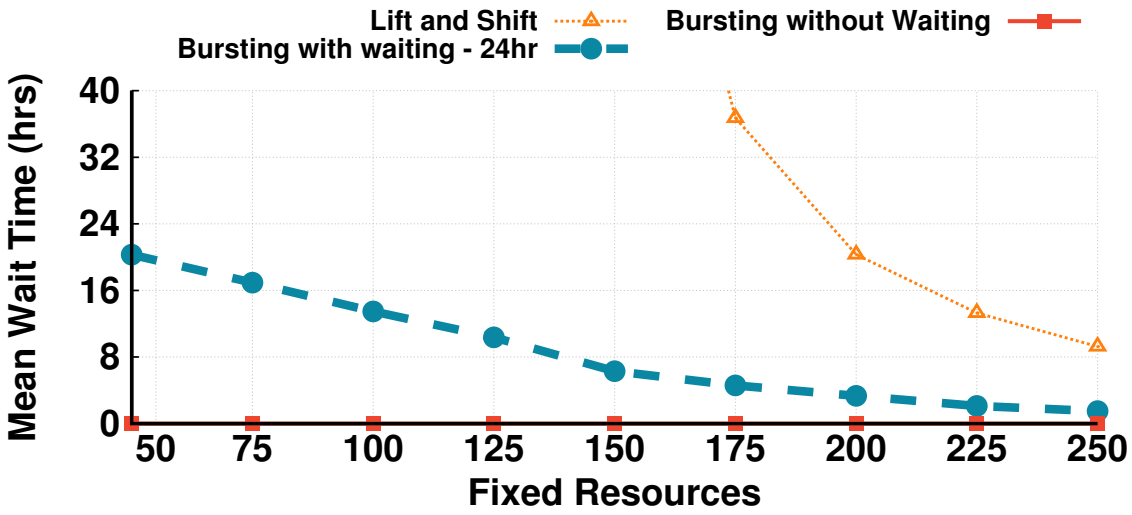
Waiting Time

- **Lift & shift – cheapest, but waiting times too high (many hours)**

Evaluation: Bursting with Waiting



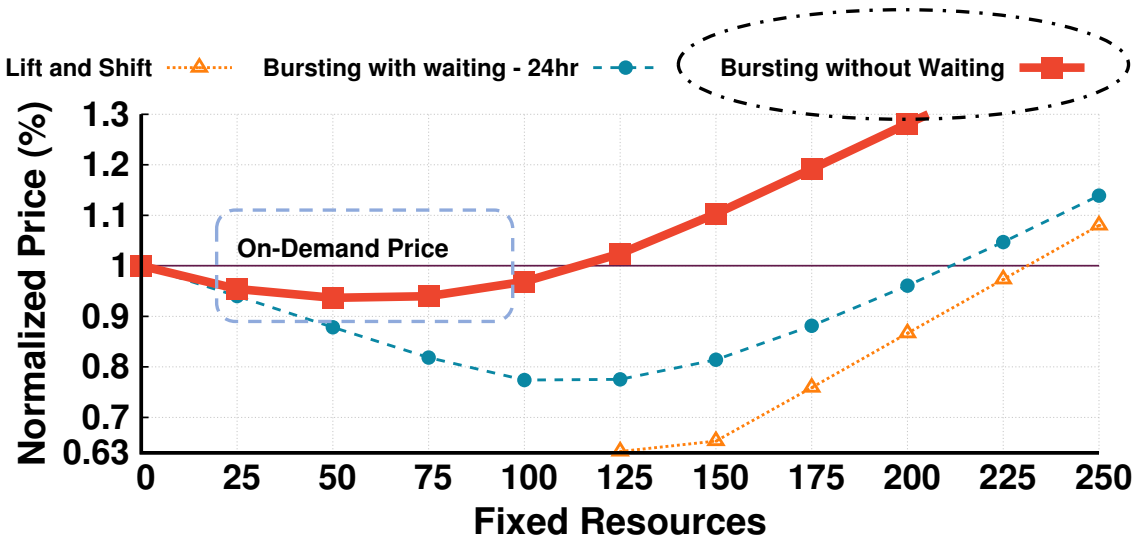
Normalized Price



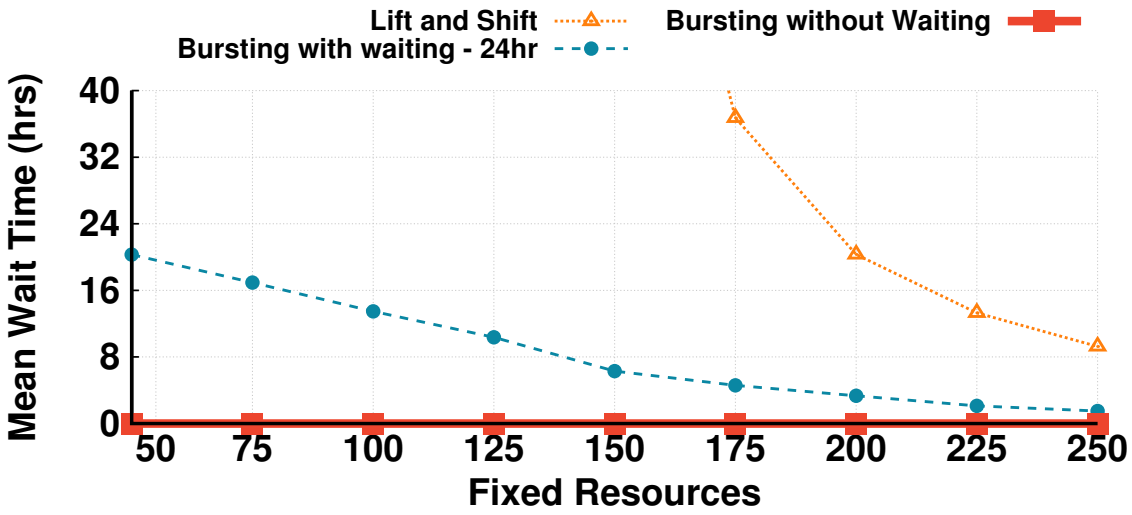
Waiting Time

- **Optimal provisioning - increases costs, decreases waiting time …**
  - … but waiting time still too high

Evaluation: Bursting w/o Waiting



Normalized Price

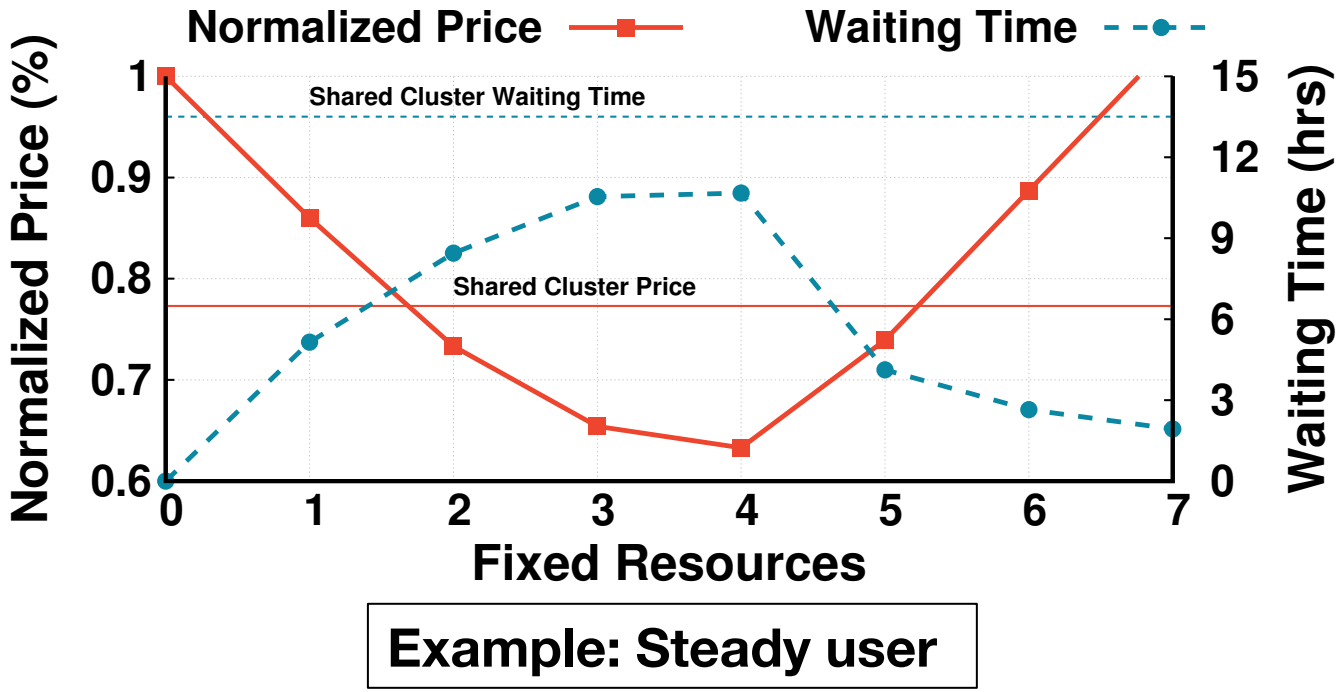


Waiting Time

- **Optimal provisioning - no waiting time, minimal cost savings**
  - Non-optimal provisioning decreases potential savings

Evaluation: Flying Solo

- **Two** broad **categories**
  - ***Small*** *users* **~46%**
  - ***Large*** *users* **~54%**
    - *Steadier users* **~ 2%**
    - *Bursty users* **~52%**



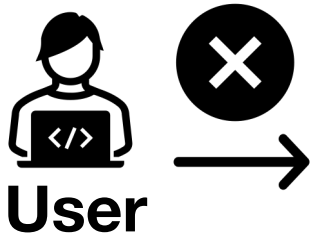
Example: Steady user

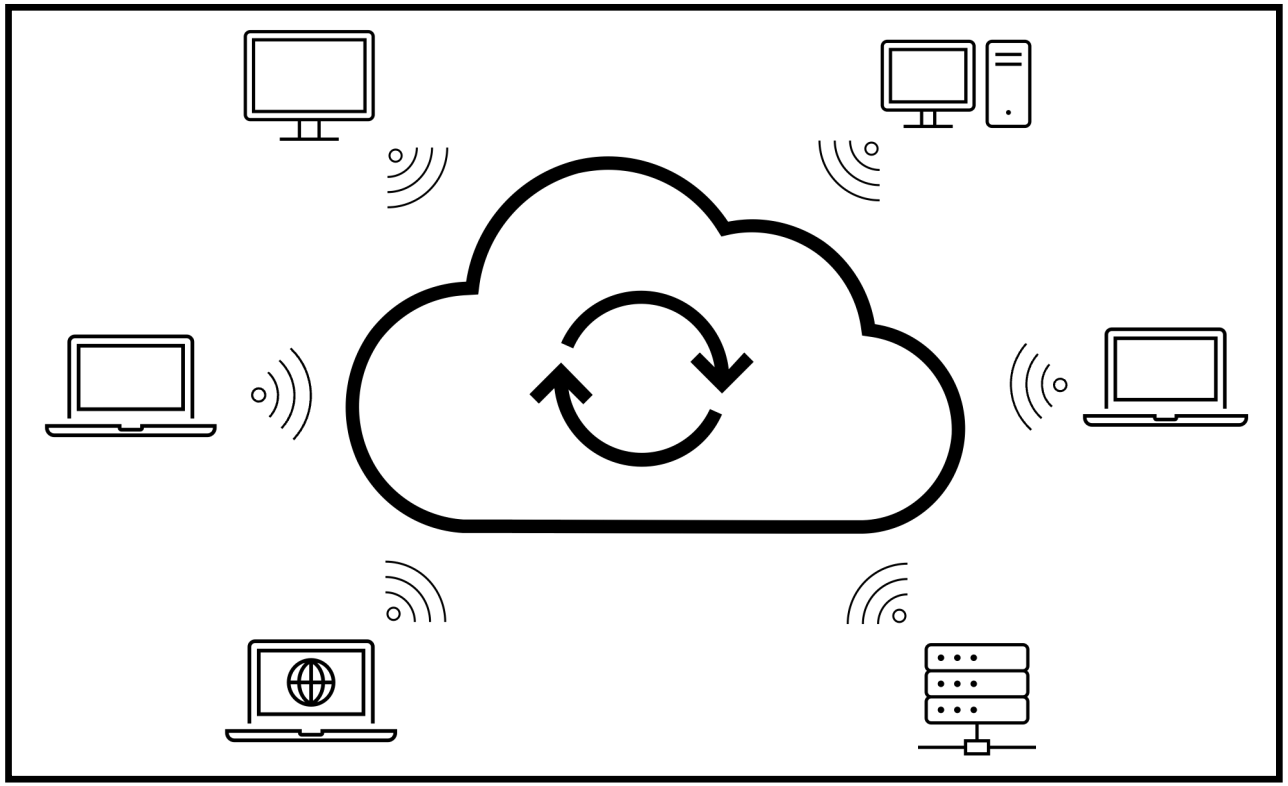
- **Few users can exploit the discounted reserved resources**
  - Most users should rent on-demand with no waiting time

Conclusion

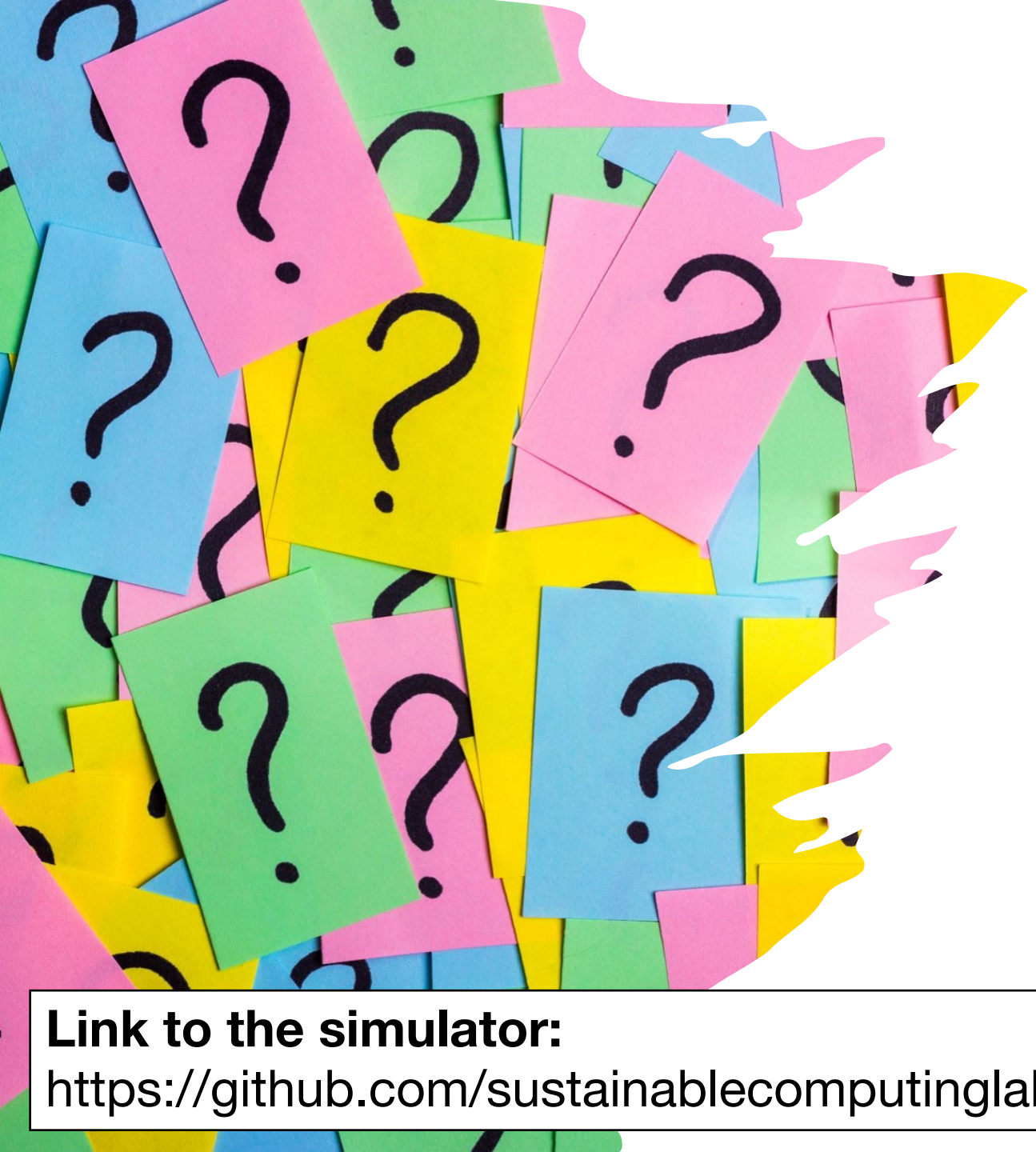
***Q:** Should users **participate** in **shared cloud cluster,** or should they **defect**?*

Defect - shared cloud clusters incur **costs near** the **on-demand** price but require **high wait times.**





Shared Cloud Cluster



Thank You!

Q&A

Talha Mehboob: tmehboob@umass.edu

Link to the simulator:
https://github.com/sustainablecomputinglab/waitinggame/tree/master/simulator