Dominant Resource Fairness in Cloud Computing Systems with Heterogeneous Servers

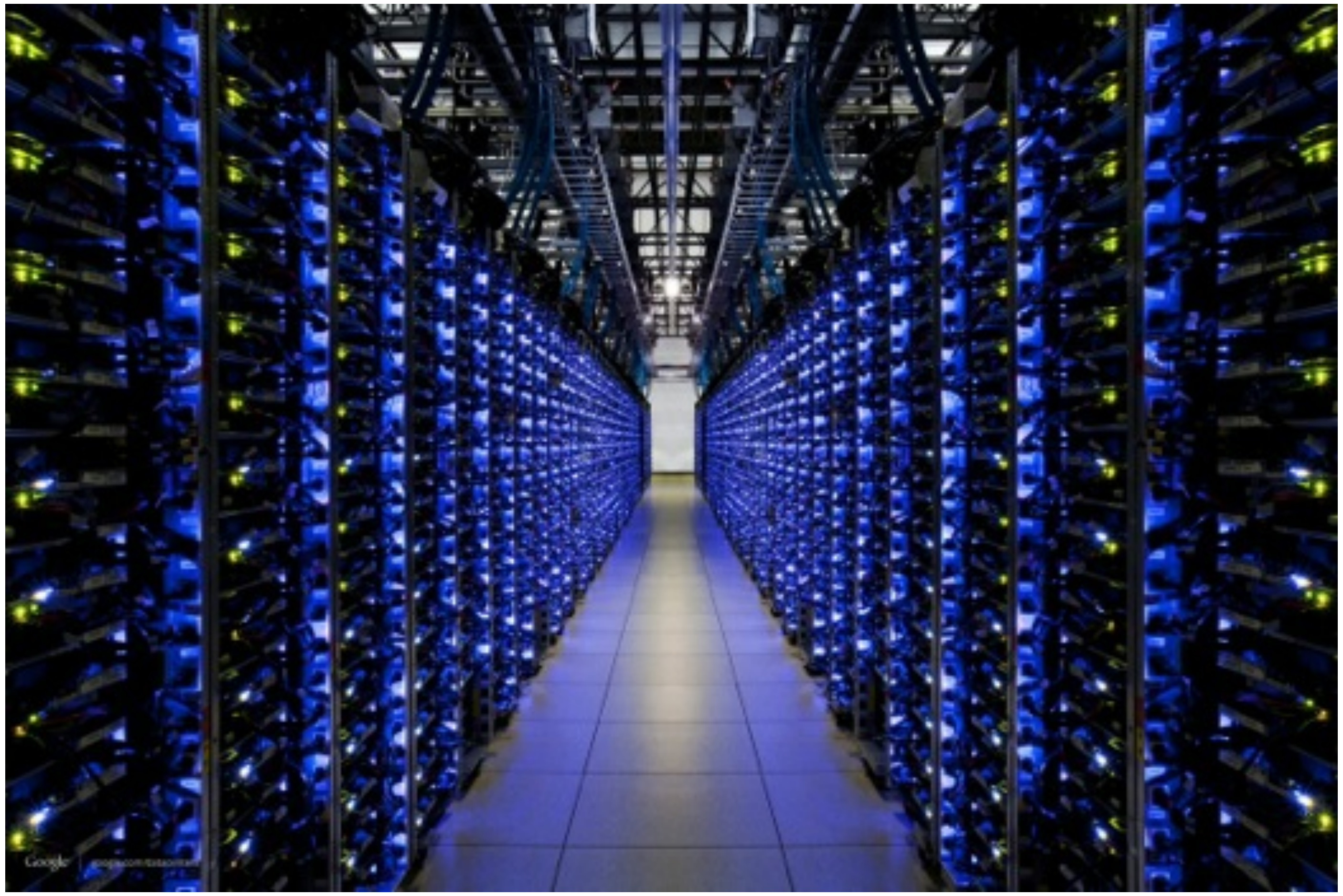
Wei Wang, Baochun Li, Ben Liang
Department of Electrical and Computer Engineering
University of Toronto

April 30, 2014

Introduction

Cloud computing system represents unprecedented heterogeneity

- Server specification
- Resource demand profiles of computing tasks



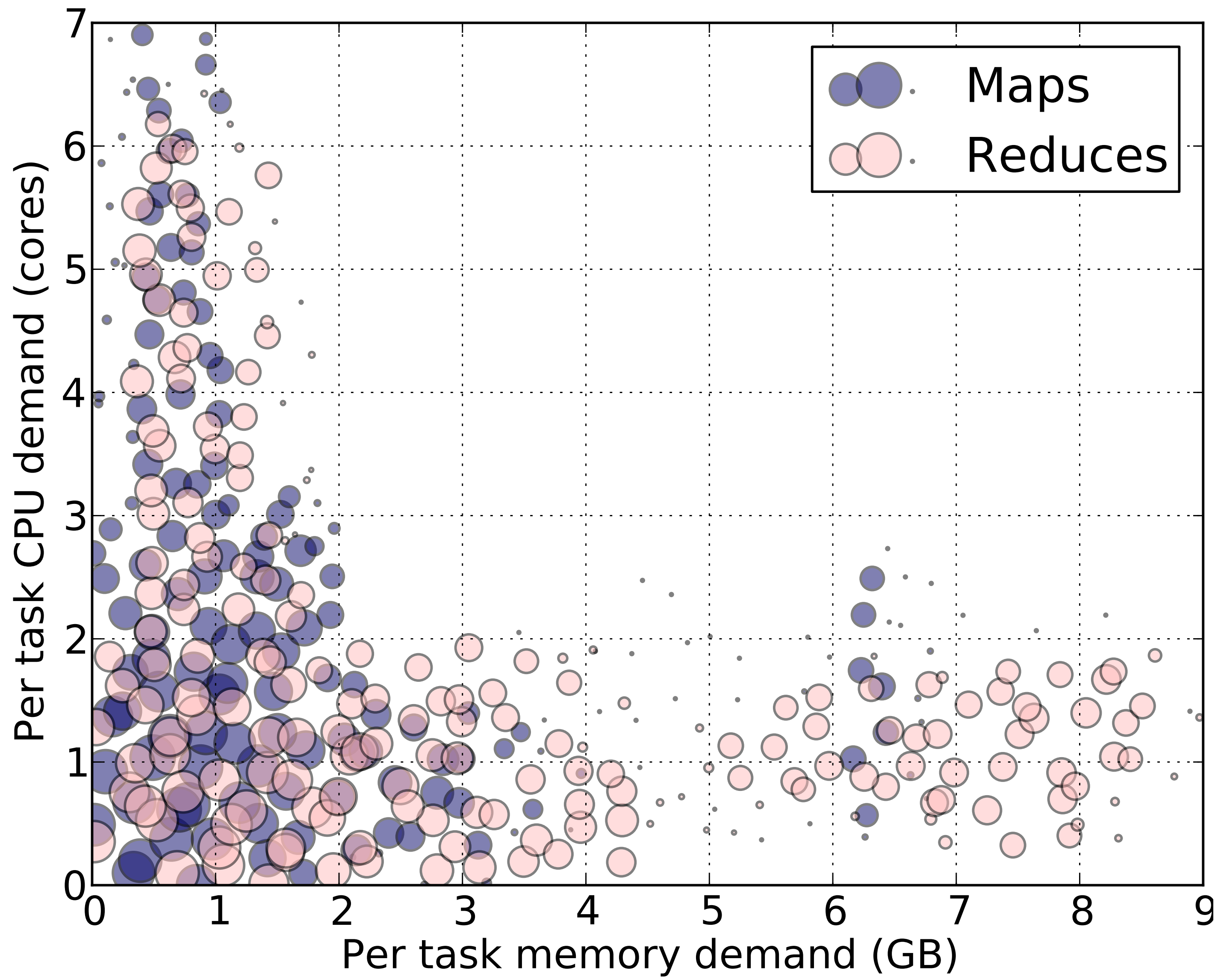
Heterogenous servers

Configurations of servers in one of Google's clusters

CPU and memory units are normalized to the maximum server

Number of servers	CPUs	Memory
6732	0.50	0.50
3863	0.50	0.25
1001	0.50	0.75
795	**1.00**	**1.00**
126	0.25	0.25
52	0.50	0.12
5	0.50	0.03
5	0.50	0.97
3	1.00	0.50
1	0.50	0.06

Heterogeneous resource demand



Ghodsi et al. NSDI11

How should resources be allocated *fairly* and *efficiently*?

State-of-the-Art Resource Allocation Mechanisms

Single-resource abstraction

Partition a server's resources into slots

E.g., a slot = (1 CPU core, 2 GB RAM)

Allocate resources to users at the granularity of slots

Hadoop Fair Scheduler & Capacity Scheduler

Dryad Quincy scheduler

Ignores the heterogeneity of both server specifications and demand profiles

Dominant Resource Fairness (DRF)

Dominant Resource Fairness (DRF)

Dominant resource

The one that requires the most allocation share

Dominant Resource Fairness (DRF)

Dominant resource

The one that requires the most allocation share

For example

A cluster: (9 CPUs, 18 GB RAM)

Job of user 1: (1 CPU, 4 GB RAM)

Job of user 2: (3 CPUs, 1 GB RAM)

Dominant Resource Fairness (DRF)

Dominant resource

The one that requires the most allocation share

For example

A cluster: (9 CPUs, 18 GB RAM)

Job of user 1: (1 CPU, 4 GB RAM)

Job of user 2: (3 CPUs, 1 GB RAM)

DRF allocation

Equalize the ***dominant share*** each user receives

3 jobs for User 1: (3 CPUs, 12 GB)

2 jobs for User 2: (6 CPUs, 2 GB)

Equalized dominant share = 2/3

Why DRF?

Why DRF?

Addresses the demand heterogeneity

Why DRF?

Addresses the demand heterogeneity

Highly attractive allocation properties [Ghodsi11]

- Pareto optimality
- Envy freeness
- Truthfulness
- Sharing incentive
- and more…

However…

DRF assumes an *all-in-one* resource model

The entire resource pool is modeled as one super computer

Ignores the heterogeneity of servers

Allocation depends only on the **total amount** of resources

May lead to an infeasible allocation

An infeasible DRF allocation

The same example

A cluster: (9 CPUs, 18 GB)

Job of user 1: (1 CPU, 4 GB)

Job of user 2: (3 CPUs, 1 GB)

DRF allocation

3 jobs for User 1: (3 CPUs, 12 GB)

2 jobs for User 2: (6 CPUs, 2 GB)

An infeasible DRF allocation

The same example

A cluster: (9 CPUs, 18 GB)

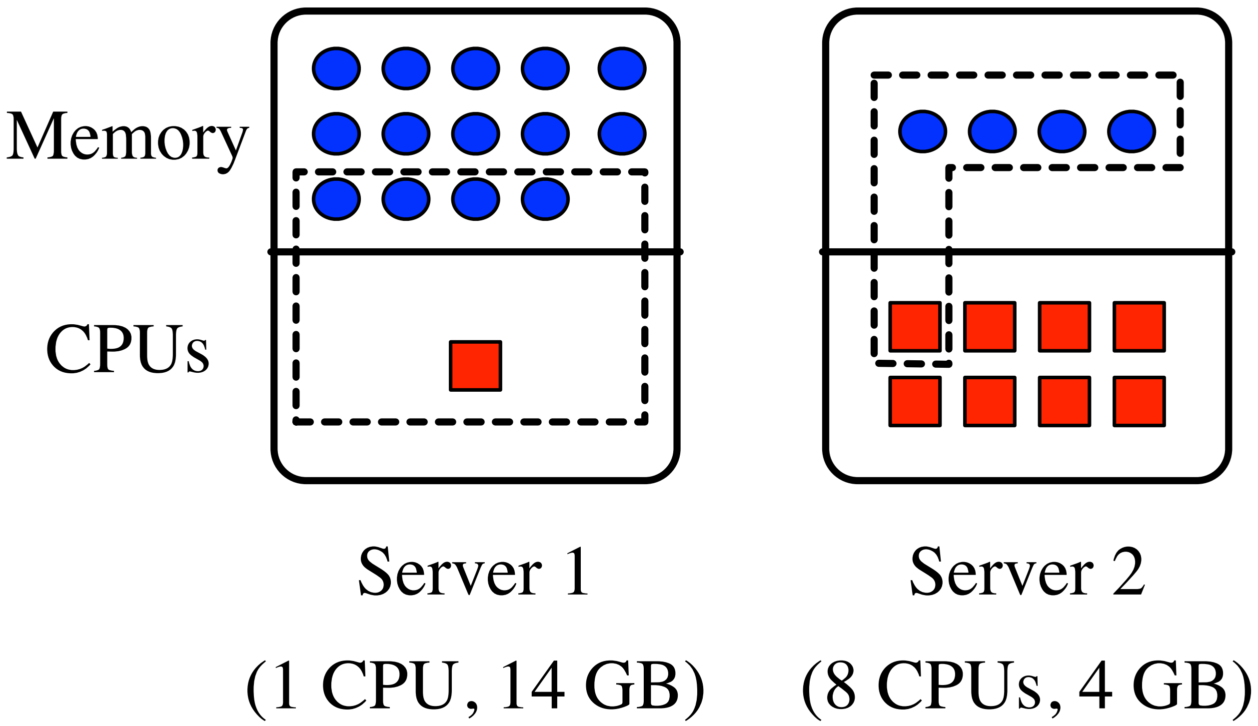
Job of user 1: (1 CPU, 4 GB)

Job of user 2: (3 CPUs, 1 GB)

DRF allocation

3 jobs for User 1: (3 CPUs, 12 GB)

2 jobs for User 2: (6 CPUs, 2 GB)



An infeasible DRF allocation

The same example

A cluster: (9 CPUs, 18 GB)

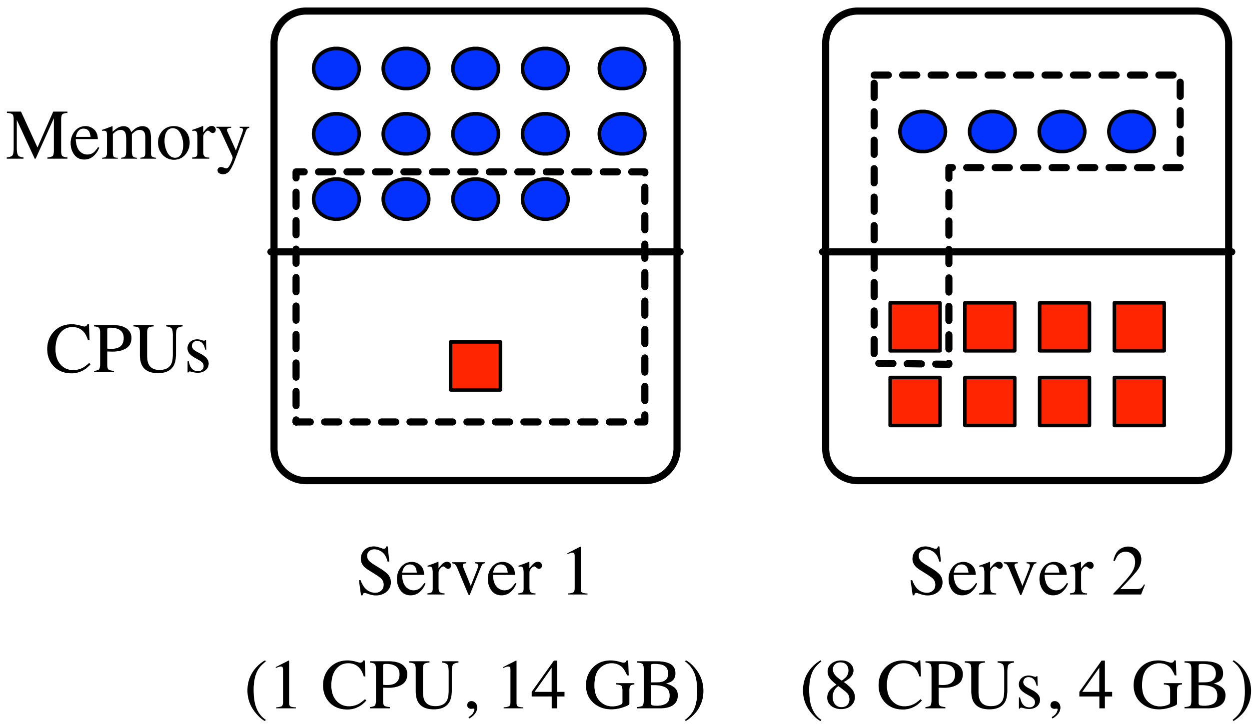
Job of user 1: (1 CPU, 4 GB)

Job of user 2: (3 CPUs, 1 GB)

DRF allocation

3 jobs for User 1: (3 CPUs, 12 GB)

2 jobs for User 2: (6 CPUs, 2 GB)



User 1 can schedule at most 2 jobs!

A quick fix of DRF

Per-Server DRF

For each server, allocate its resources to all users, using DRF

However…

Per-server DRF may lead to an arbitrarily inefficient allocation

See the paper for details

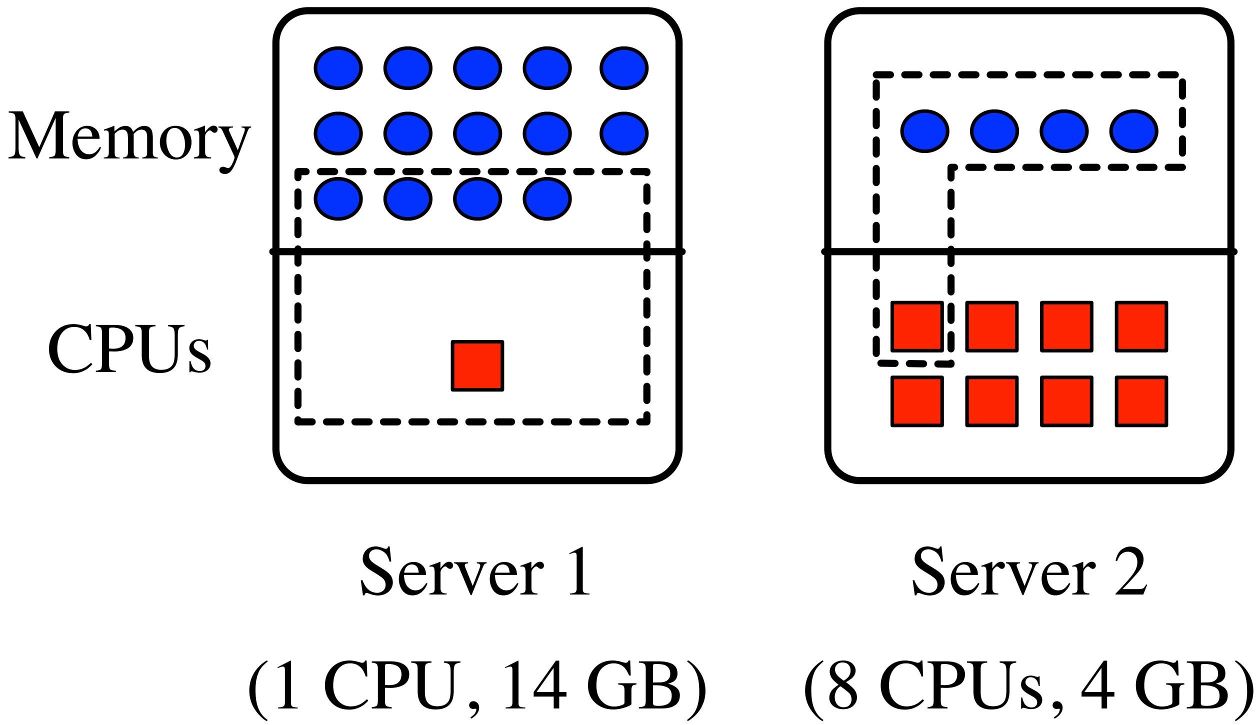
Can the attractiveness of DRF extend to a heterogeneous environment?

The ambiguity of dominant resource

The same example

A cluster: (9 CPUs, 18 GB)

Job of user 1: (1 CPU, 4 GB)

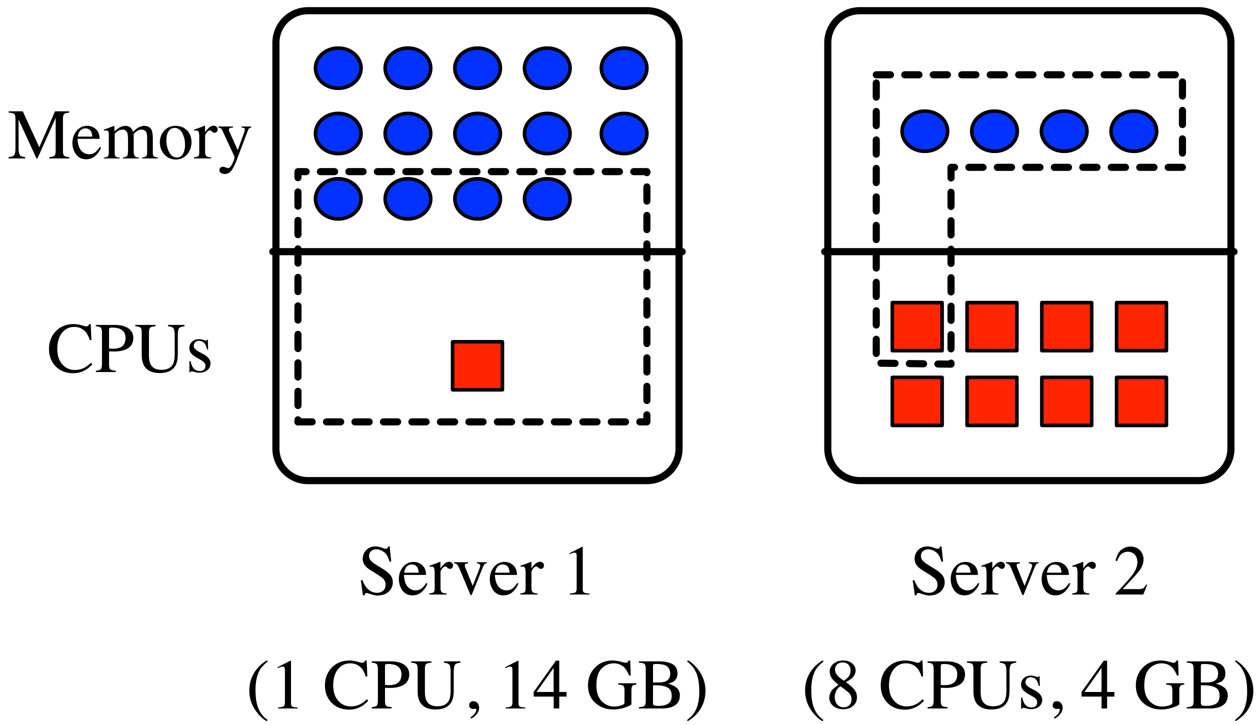


The ambiguity of dominant resource

The same example

A cluster: (9 CPUs, 18 GB)

Job of user 1: (1 CPU, 4 GB)



How to define dominant resource?

For server 1, the dominant resource is CPU

For server 2, the dominant resource is memory

For the entire resource pool, the dominant resource is memory

Our answer: DRFH

A generalization of *DRF* mechanism in *H*eterogeneous environments

- Equalizes every user's *global dominant share*

Retains almost all the attractive allocation properties of DRF

- Pareto optimality
- Envy-freeness
- Truthfulness
- Weak sharing incentive
- and more…

Easy to implement

DRFH Allocation

A global view of dominant resource

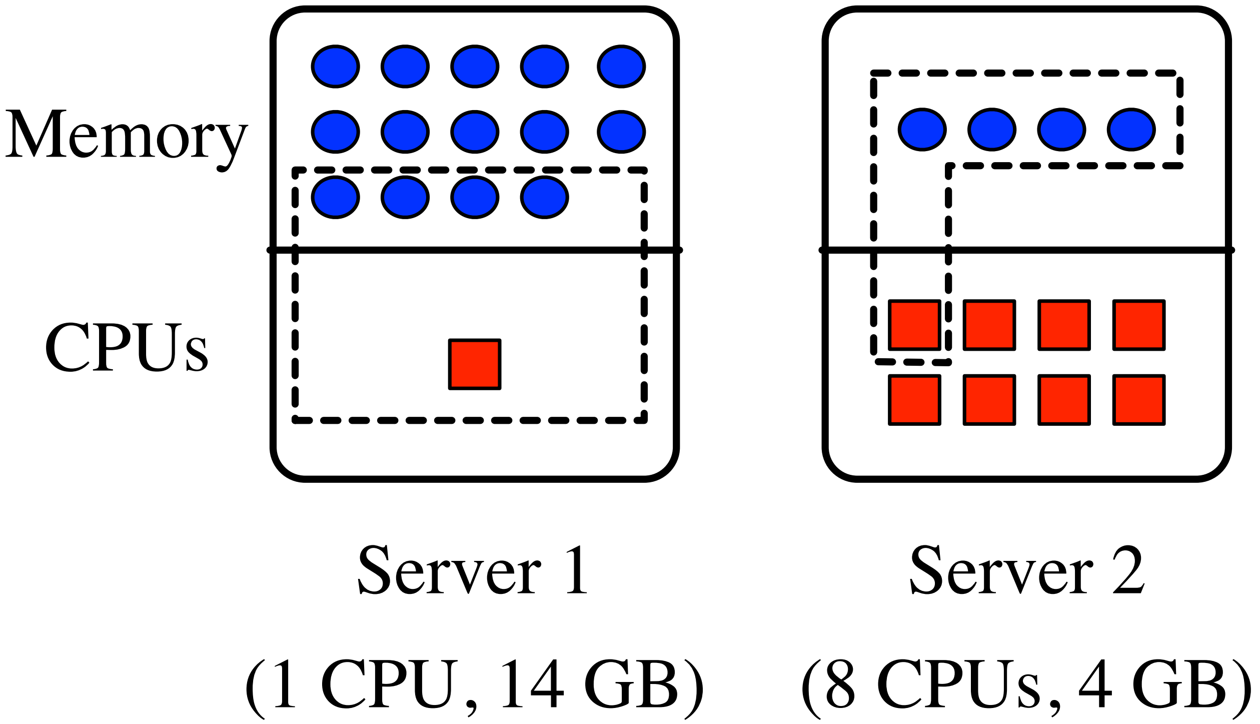
Global dominant resource

The one that requires the maximum allocation share of the entire resource pool

The same example

A cluster: (9 CPUs, 18 GB)

Job of user 1: (1 CPU, 4 GB)



Memory is the global dominant resource

Key intuition

Max-min fairness on the global dominant resources, subject to resource constraints per server

Global dominant share

$$\begin{aligned} \max_{\mathbf{A}} \quad & \min_{i \in U} G_i(\mathbf{A}_i) \\ \text{s.t.} \quad & \sum_{i \in U} A_{ilr} \le c_{lr}, \forall l \in S, r \in R \ . \end{aligned}$$

Allocation share of resource *r* user *i receives* on server *l*

Total availability of resource *r* on server *l*

DRFH Properties

Fairness property

Fairness property

DRFH is envy-free

- No user can schedule more computing tasks by taking the other's resource allocation
  - No one will envy the other's allocation

Fairness property

DRFH is envy-free

No user can schedule more computing tasks by taking the other's resource allocation

No one will envy the other's allocation

DRFH is truthful

No user can schedule more computing tasks by misreporting its resource demand

Strategic behaviours are commonly seen in real system [Ghodsi11]

Fairness property

DRFH is envy-free

- No user can schedule more computing tasks by taking the other's resource allocation
  - No one will envy the other's allocation

DRFH is truthful

- No user can schedule more computing tasks by misreporting its resource demand
  - Strategic behaviours are commonly seen in real system [Ghodsi11]

Resource utilization

DRFH is Pareto optimal

No user can schedule more tasks without decreasing the number of tasks scheduled for the others

- No resource that could be utilized to serve a user is left idle

Service isolation

Equal partition

Allocation **A** is an equal partition if it divides every resource evenly among all *n* users

$$\sum_{l \in S} A_{ilr} = 1/n, \quad \forall r \in R,\ i \in U$$

Allocation share of resource *r* user *i receives* on server *l*

Service isolation

Equal partition

Allocation **A** is an equal partition if it divides every resource evenly among all *n* users

$$\sum_{l \in S} A_{ilr} = 1/n, \quad \forall r \in R,\ i \in U$$

Allocation share of resource *r* user *i receives* on server *l*

Weak sharing incentive

There exists an equal allocation **A'** under which each user schedules fewer tasks than those under DRFH

DRFH is **unanimously preferred** to an equal allocation by all users

Comparison

DRFH

- Pareto optimality
- Envy freeness
- Truthfulness
- **Weak** sharing incentive

DRF (all-in-one model)

- Pareto optimality
- Envy freeness
- Truthfulness
- **Strong** sharing incentive

Comparison

DRFH

Pareto optimality

Envy freeness

Truthfulness

Weak sharing incentive

DRF (all-in-one model)

Pareto optimality

Envy freeness

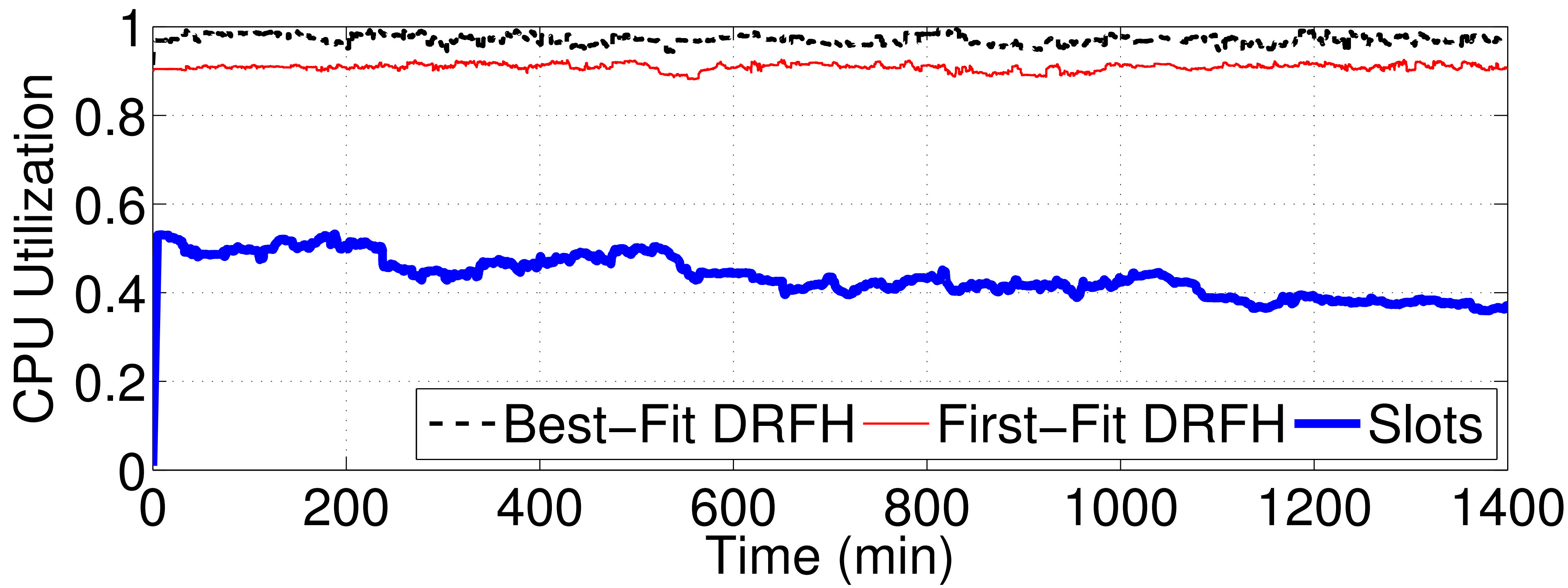
Truthfulness

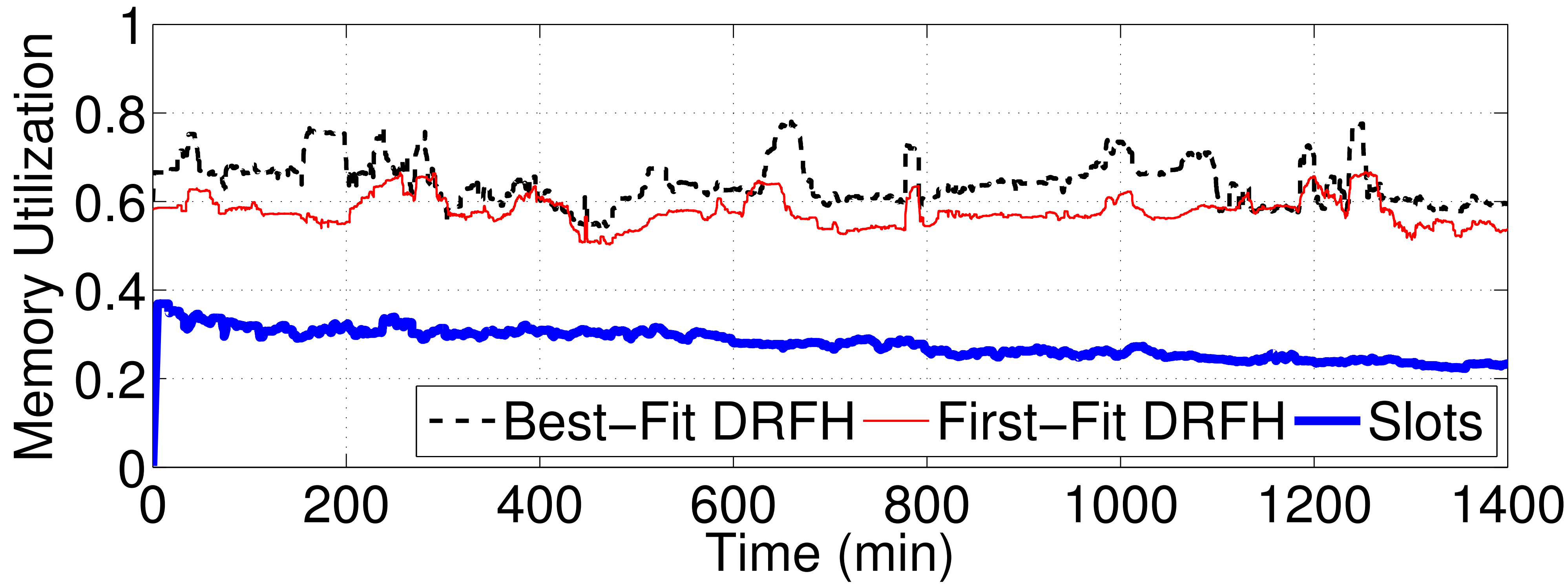
Strong sharing incentive

DRFH retains almost all the attractive properties of DRF

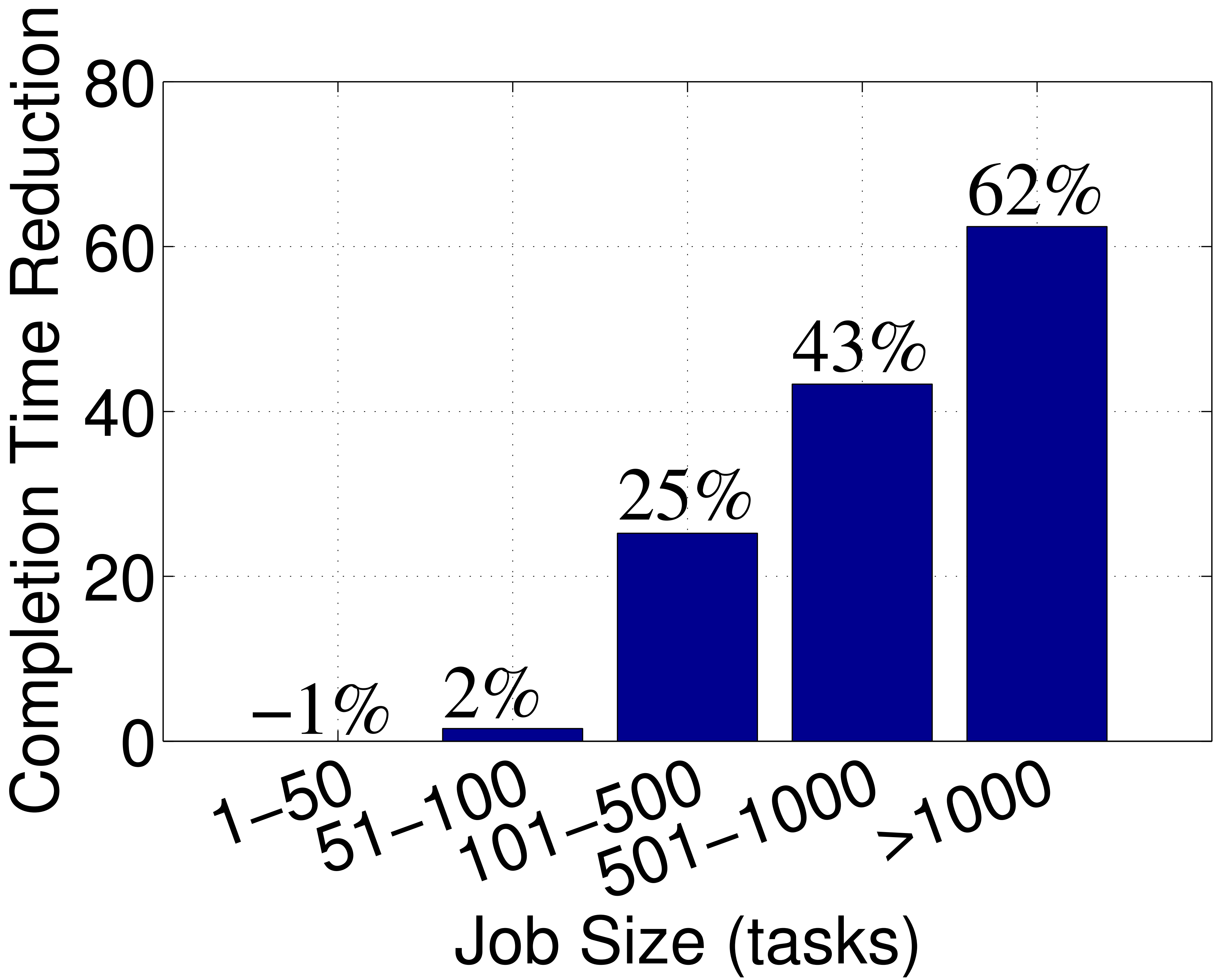
Trace-Driven Simulation

Resource utilization





Job completion times



Conclusions

We have studied a multi-resource fair allocation problem in a heterogeneous cloud computing system

We have generalized DRF to DRFH and shown that it possesses a set of highly attractive allocation properties

We have designed an effective heuristic algorithm that implements DRFH in a real-world system

http://iqua.ece.toronto.edu/~weiwang/