Preemptive Scheduling in Mesos Framework

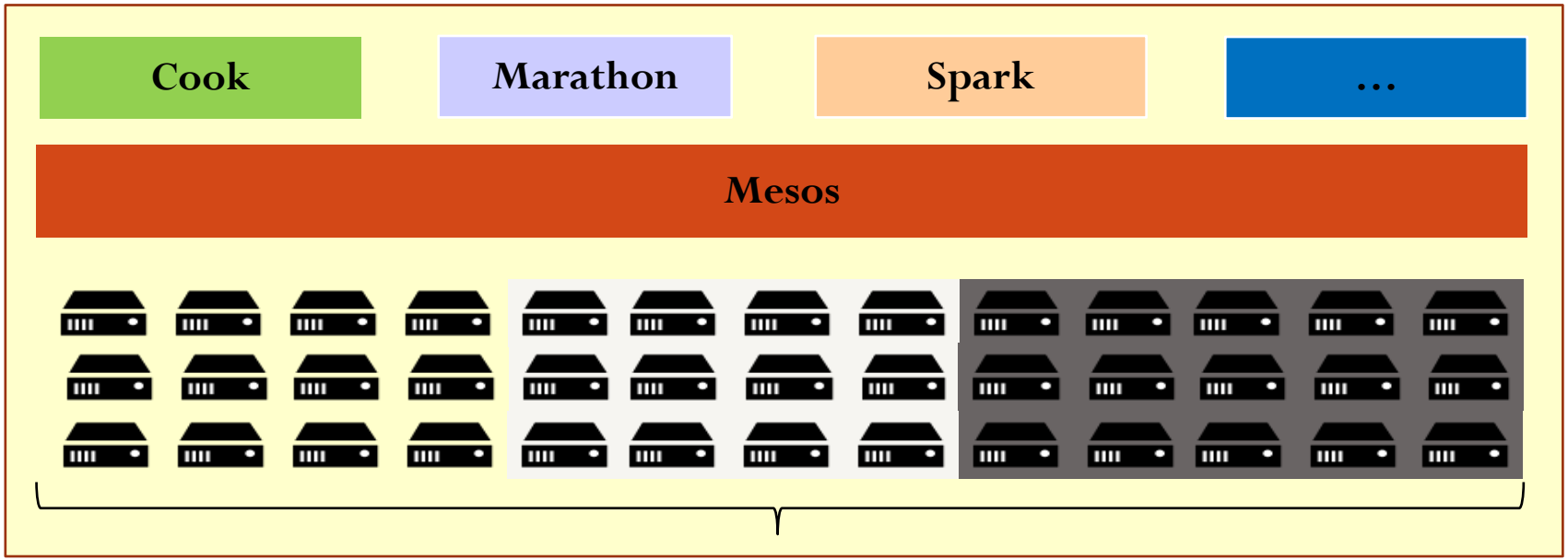
Li Jin

About Two Sigma

About Me

- Software Engineer @ Two Sigma

Mesos @ TS



Outline

- Cook: A Mesos Framework
- Problem: Utilization and Fairness
- Preemption: Intuition and Formalization
- Preemption in Cook: Implementation

What is Cook

- Two Sigma's ***Simulation*** Platform
  - Manage tens of thousands of simulations
  - Share compute resource among users

What is Simulation

- Idempotent, parallel, distributed, resource intensive computations
- One simulation = Multiple Mesos tasks

What is Simulation

- Simulation task footprint
  - 10 ~ 100 GB RAM
  - 1 ~ 20 CPUs
  - 15 minutes ~ a few hours
- Simulation use cases
  - Interactive Research
  - Batch computation

Outline

- Cook: A Mesos Framework
- Problem: Utilization and Fairness
- Preemption: Intuition and Formalization
- Preemption in Cook: Implementation

Problem

- High computation demand
  - 5 x capacity during peak hours
- Optimize
  - Utilization
    - Use as much compute resources as possible
  - Fairness
    - Allocate resources fairly for some definition of ‘fair’

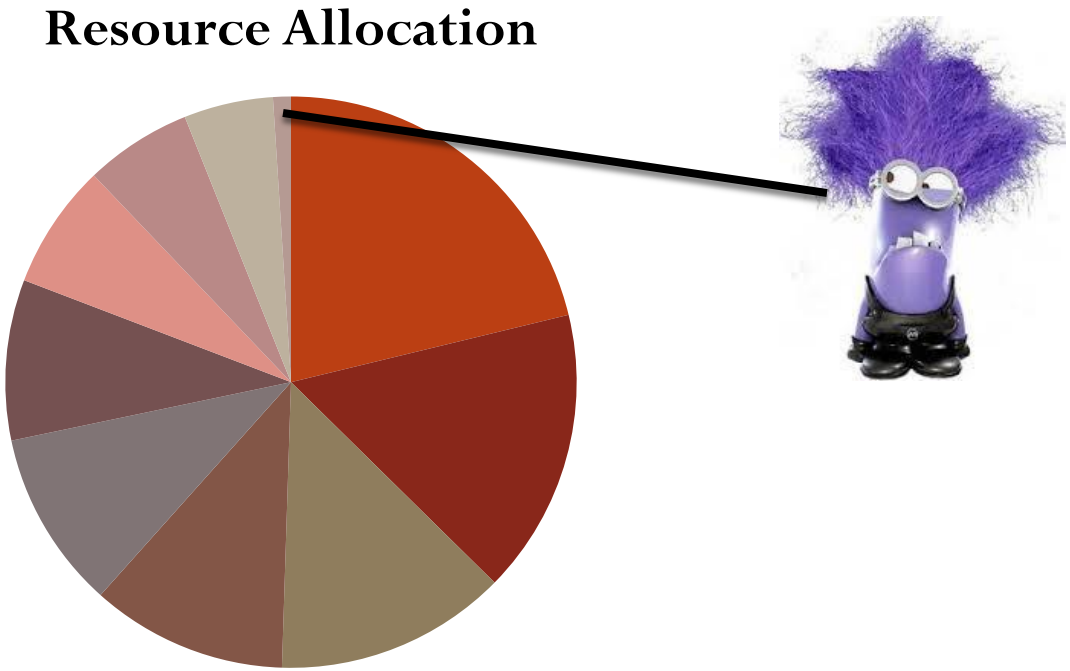
What is Fairness

- FIFO
- Time sharing
- Throw a dice
- …

What is Fairness

- A story…

What is Fairness



What is Fairness, Really

- Fairness is not about ‘fair’
- Fairness is about **user experience**
  - User should get their share of the cluster **whenever** they need it

Quota

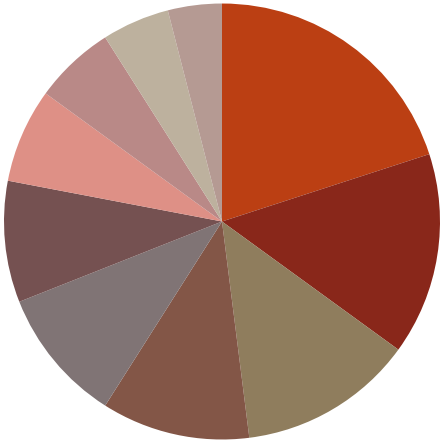
- Quota = Maximum percentage of the cluster allowed
- Static
  - 100 % / # Max concurrent users
- Pros:
  - Guarantee Fairness
- Cons:
  - Low Utilization

Dynamic Quota

- Dynamic
  - Quota * Utilization Adjustment
- Pros:
  - Higher Utilization
- Cons:
  - Slow reaction to change of demand

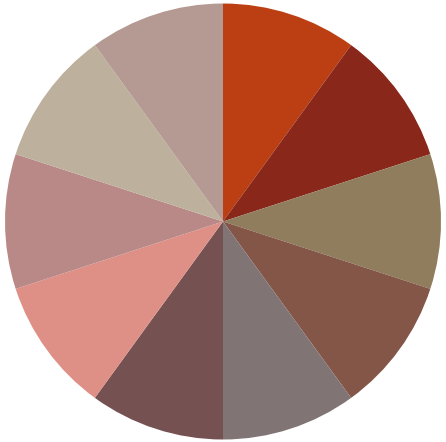
Dynamic Quota

Unfair Resource Allocation

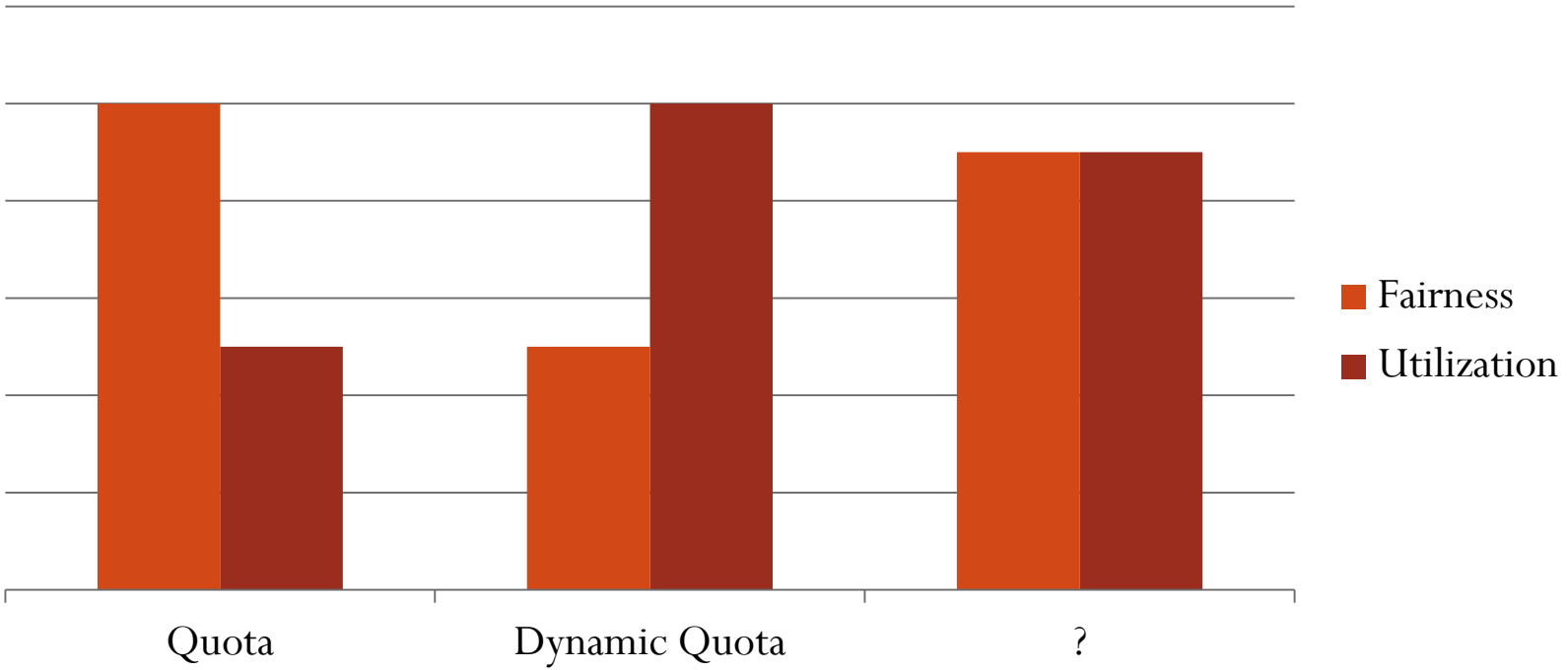


Hours…

Fair Resource Allocation



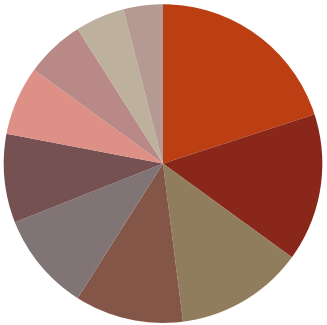
Can we do better?



Preemption

- Kill a Mesos task and reschedule later
- Reclaim resource faster!

Unfair Resource Allocation



Minutes!

Fair Resource Allocation

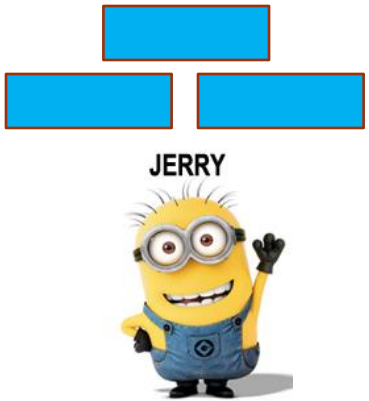
Outline

- Cook: A Mesos Framework
- Problem: Utilization and Fairness
- Preemption: Intuition and Formalization
- Preemption in Cook: Implementation

Preemption: Intuition

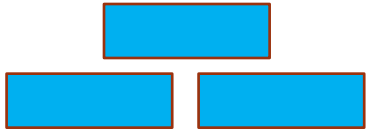
Waiting

Running



Preemption: Intuition

Waiting



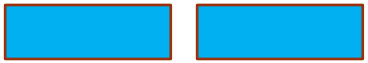
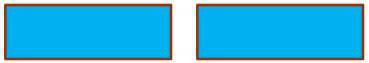
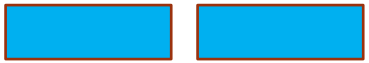
Running

Preemption: Intuition

Waiting



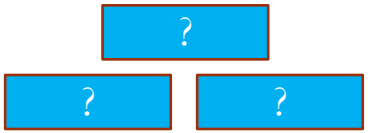
Running

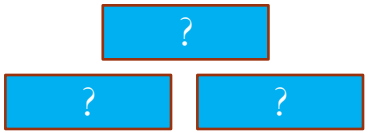




Preemption: Intuition

Waiting

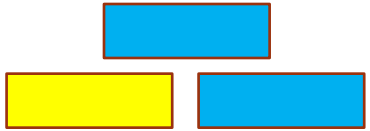
Running





Preemption: Intuition

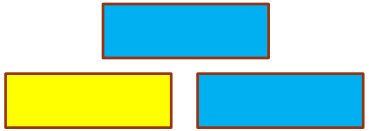
Waiting



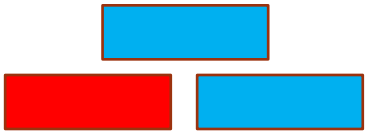
Running

Preemption: Intuition

Waiting

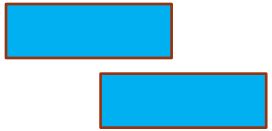


Running

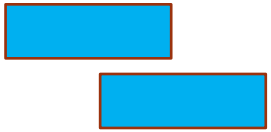


Preemption: Intuition

Waiting

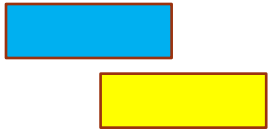


Running

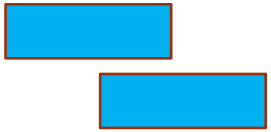


Preemption: Intuition

Waiting

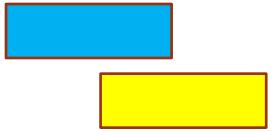


Running

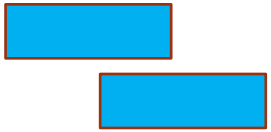


Preemption: Intuition

Waiting



Running



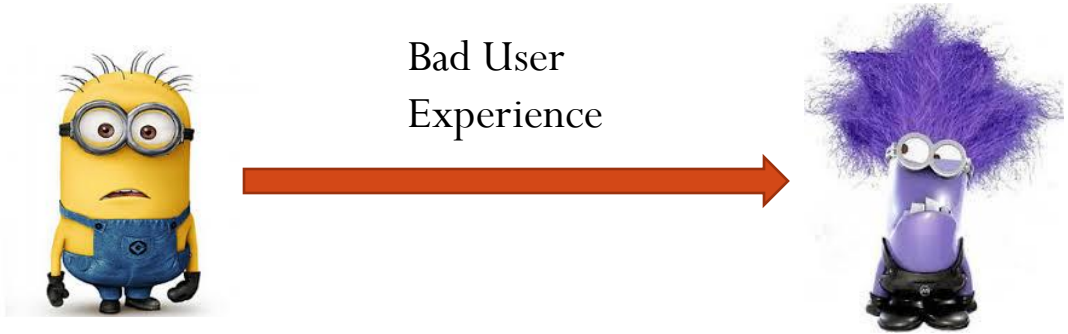
Preemption: Intuition

Waiting

Running

Problem

- Not all jobs are equal
  - We just preempted some important jobs!



Score Function

- Reflect task's value
  - Fairness
  - Importance
- Preempt low value task for high value task

Preemption: Intuition

Waiting

¥¥
¥¥¥
¥¥¥¥

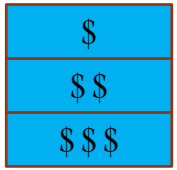
Running

€
€€
€€€

£££
££££
£££££

Preemption: Intuition

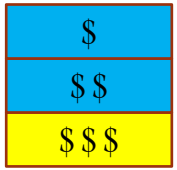
Waiting



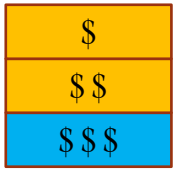
Running

Preemption: Intuition

Waiting

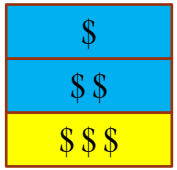


Running



Preemption: Intuition

Waiting

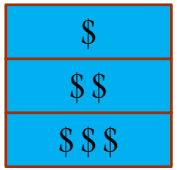


Running

Preemption: Intuition

Waiting

Running



Preemption: Intuition

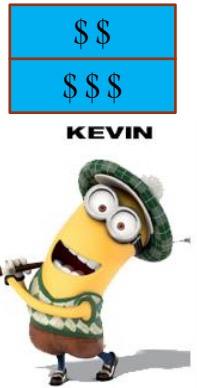
Waiting

Running

Preemption: Intuition

Waiting

Running



Outline

- Cook: A Mesos Framework
- Problem: Utilization and Fairness
- Preemption: Intuition and Formalization
- Preemption in Cook: Implementation

Cumulative Resource Share (CRS)

- Assuming there is an total order of jobs for ***each user***, where > means 'has higher value than'.
  - CRS of job j is sum of all jobs of the same user that are greater than or equal to j, divided by total resource.
- $CRS(j) = \frac{1}{R_{Total}} \sum_{j' \geq j} R_{j'}$

Preemption: Formalization

Waiting

3/6
2/6
1/6

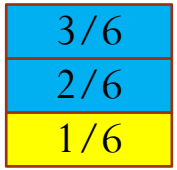
Running

3/6
2/6
1/6

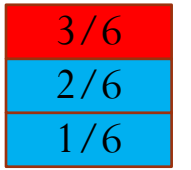
3/6
2/6
1/6

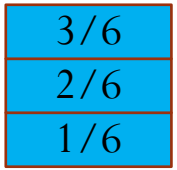
Preemption: Formalization

Waiting



Running

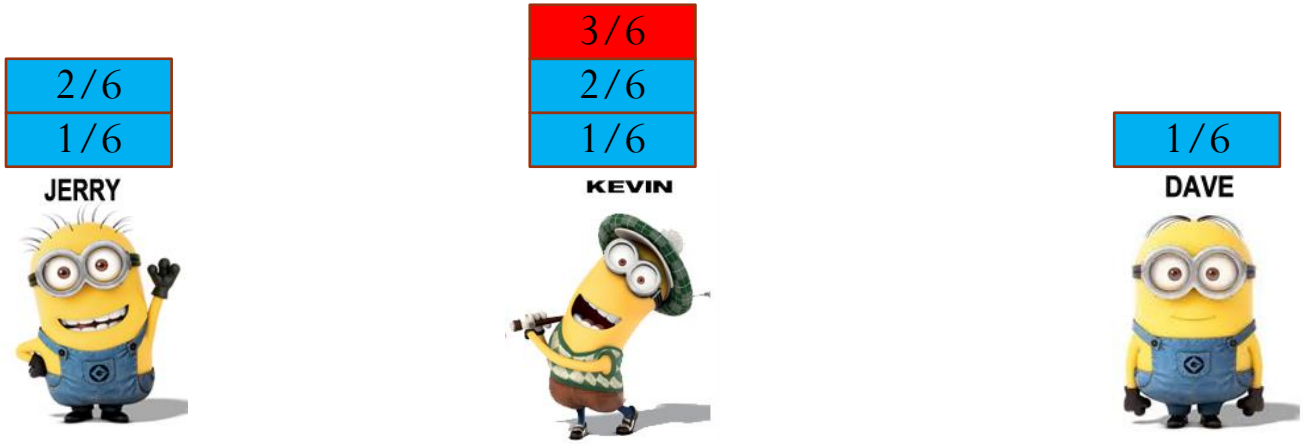




Preemption: Formalization

Waiting

Running



Preemption: Formalization

Waiting

3/6	3/6	3/6

Running

2/6	2/6	2/6
1/6	1/6	1/6

Cumulative Resource Share

Multiple Resources?

- Dominant Resource Fairness: Fair Allocation of Multiple Resource Types
  - UC Berkeley in 2011
  - Used in Mesos
- Dominant Resource Share:
  - DRS(u) = $\max_{R} \frac{R_u}{R_{Total}}$

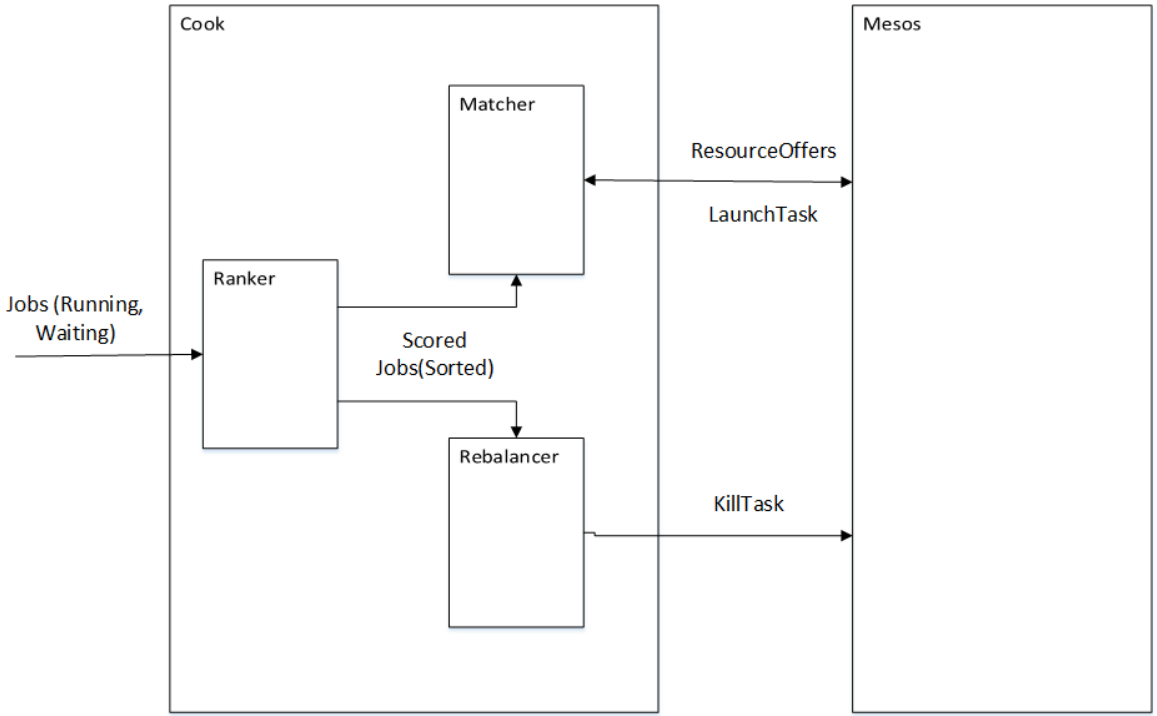
Dominant Cumulative Resource Share

- $CRS(j) = \frac{1}{R_{Total}} \sum_{j' \geq j} R_{j'}$
- $DCRS(j) = \max_{R} \frac{1}{R_{Total}} \sum_{j' \geq j} R_{j'}$
- $Score(j) = -DCRS(j)$

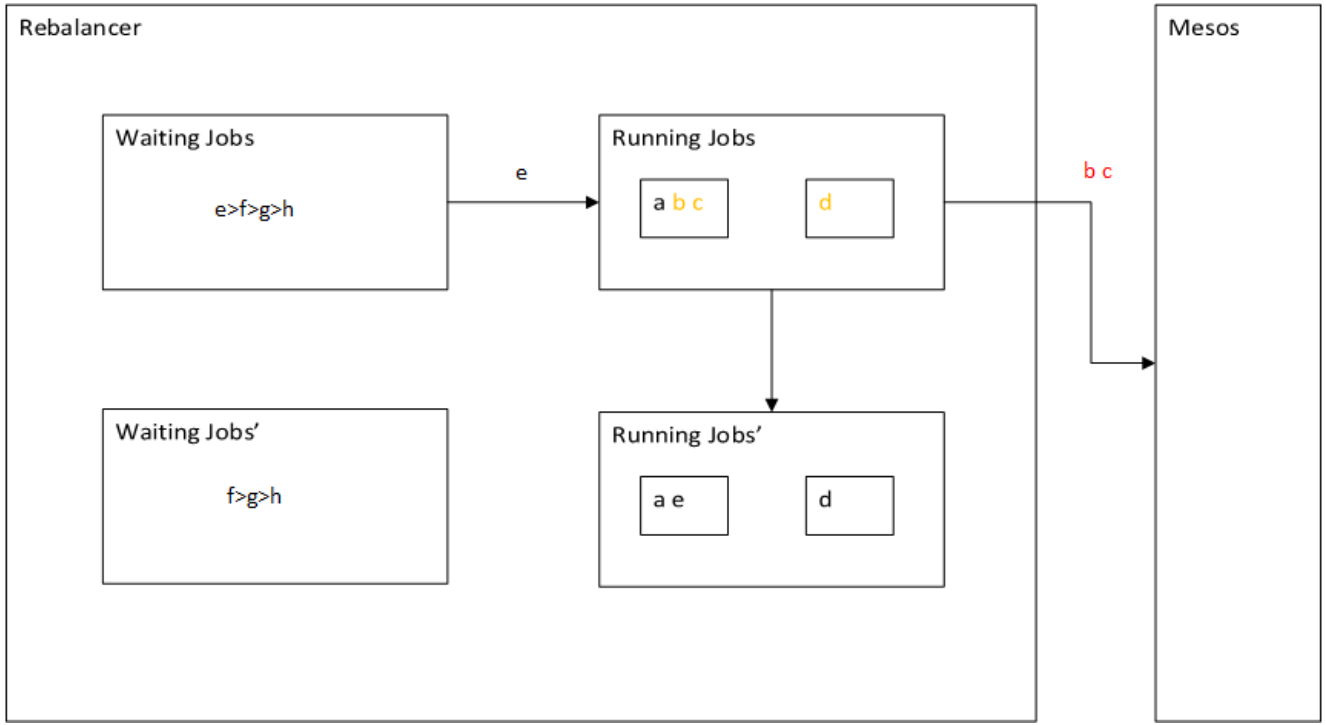
Outline

- Cook: A Mesos Framework
- Problem: Utilization and Fairness
- Preemption: Intuition and Formalization
- Preemption in Cook: Implementation

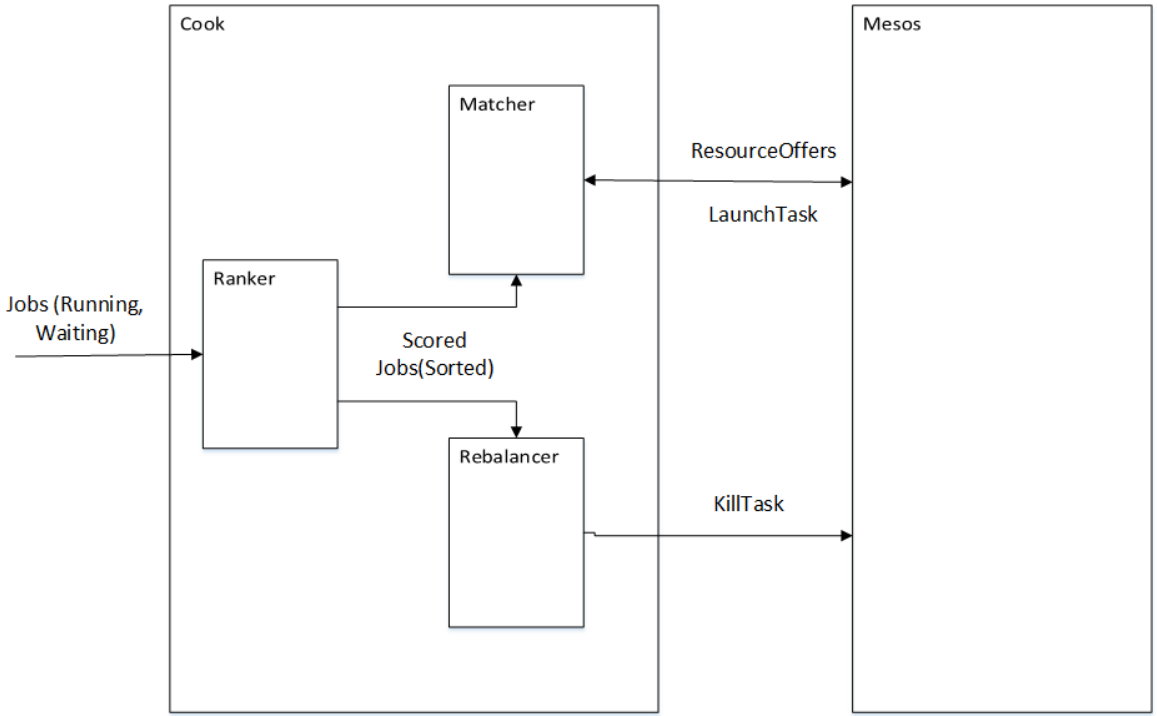
Cook: Architecture



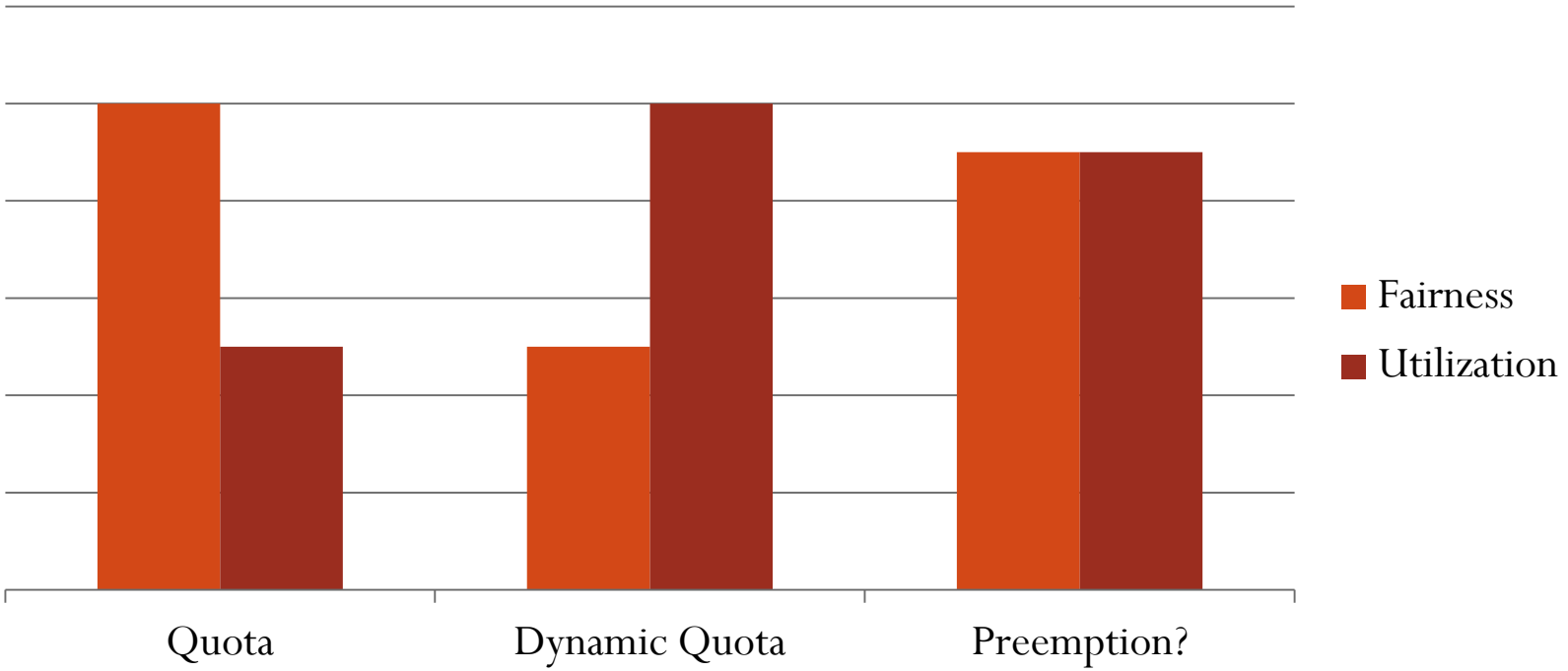
Rebalancer


Rebalancer
Waiting Jobs
e>f>g>h
e
Running Jobs
a b c
d
b c
Mesos
Waiting Jobs'
f>g>h
Running Jobs'
a e
d

Cook: Architecture

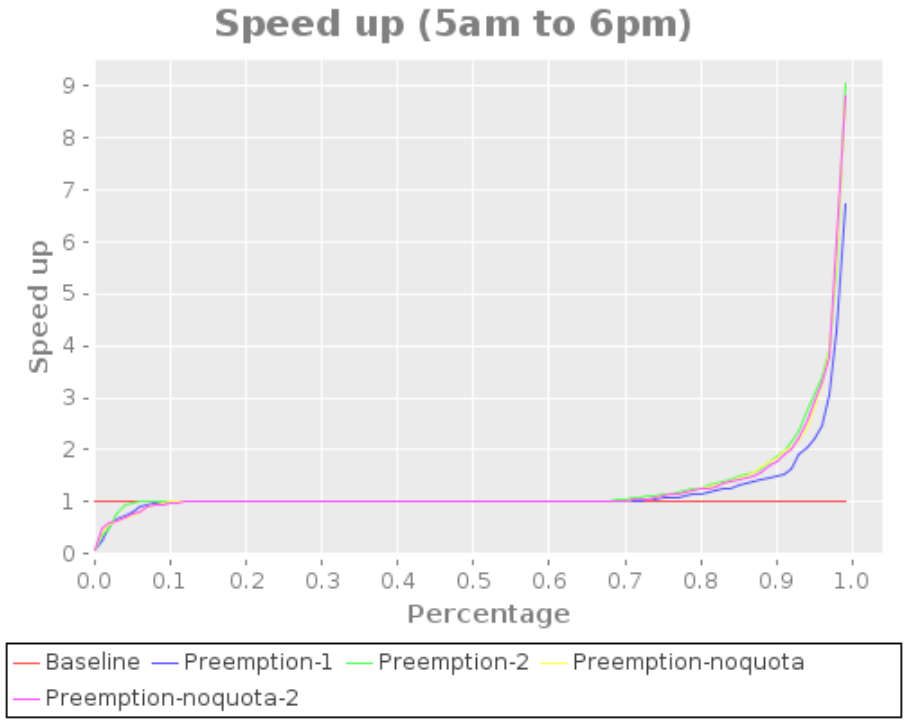


Are we doing better?

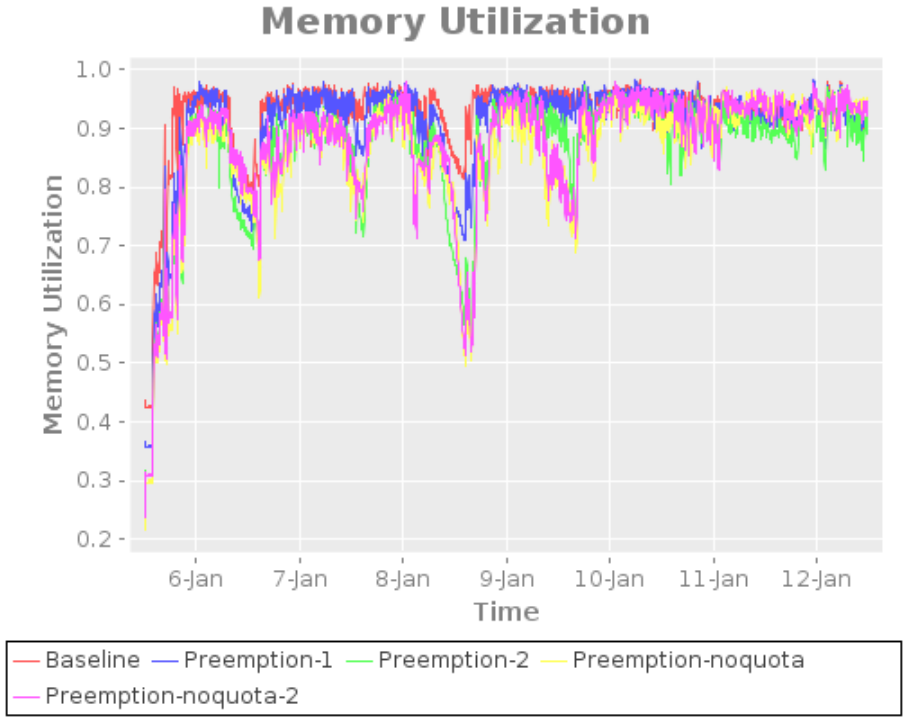


Yes!

It works!


Speed up (5am to 6pm)
Speed up
Percentage
0
1
2
3
4
5
6
7
8
9
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Baseline
Preemption-1
Preemption-2
Preemption-noquota
Preemption-noquota-2

It works!


Memory Utilization
Memory Utilization
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
6-Jan
7-Jan
8-Jan
9-Jan
10-Jan
11-Jan
12-Jan
Time
Baseline
Preemption-1
Preemption-2
Preemption-noquota
Preemption-noquota-2

Think Beyond Cook

- DCRS in Mesos?

Open Source

- Not quite yet… But we are getting close
- Github.com/twosigma