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

Resource Allocation
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

Fair Resource Allocation

Hours…
Can we do better?
Preemption

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

Unfair Resource Allocation

Fair Resource Allocation

Minutes!
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

JERRY
KEVIN
DAVE
Preemption: Intuition

Waiting

Running

JERRY

KEVIN

DAVE
Preemption: Intuition

Waiting

Running

JERRY

KEVIN

DAVE
Preemption: Intuition

Waiting

Running

JERRY

KEVIN

DAVE
Preemption: Intuition

Waiting

Running

JERRY

KEVIN

DAVE
Preemption: Intuition

Waiting

Running
Preemption: Intuition

Waiting

Running

JERRY

KEVIN

DAVE
Preemption: Intuition

Waiting

Running

JERRY

KEVIN

DAVE
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

Jerry

Kevin

Dave
Preemption: Intuition

Waiting

Running

$ 
$$ 
$$$

$ 
$$ 
$$$

$ 
$$ 
$$$

JERRY

KEVIN

DAVE
Preemption: Intuition

Running

Waiting

$  
$$  
$$$  

$  
$$  
$$$  

$  
$$  
$$$  

JERRY  

KEVIN  

DAVE
Preemption: Intuition

Waiting

Running

<table>
<thead>
<tr>
<th>Jerry</th>
<th>Kevin</th>
<th>Dave</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>$$</td>
<td>$$</td>
<td></td>
</tr>
<tr>
<td>$$$</td>
<td>$$$</td>
<td></td>
</tr>
</tbody>
</table>
Preemption: Intuition

Waiting

Running

Jerry

Kevin

Dave
Preemption: Intuition

Waiting

Running

JERRY

KEVIN

DAVE
Preemption: Intuition

Waiting

Running

JERRY

KEVIN

DAVE
Outline

• Cook: A Mesos Framework
• Problem: Utilization and Fairness
• Preemption: Intuition and Formalization
• Preemption in Cook: Implementation
Cumulative Resource Share (CRS)

- Assuming there is a total order of jobs for each user, where > means ‘has higher value than’.
  - CRS of job $j$ is the 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

Running

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JERRY</td>
<td></td>
</tr>
<tr>
<td>3/6</td>
<td>3/6</td>
<td>3/6</td>
</tr>
<tr>
<td>2/6</td>
<td>2/6</td>
<td>2/6</td>
</tr>
<tr>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KEVIN</td>
<td></td>
</tr>
<tr>
<td>3/6</td>
<td>3/6</td>
<td>3/6</td>
</tr>
<tr>
<td>2/6</td>
<td>2/6</td>
<td>2/6</td>
</tr>
<tr>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DAVE</td>
<td></td>
</tr>
<tr>
<td>3/6</td>
<td>3/6</td>
<td>3/6</td>
</tr>
<tr>
<td>2/6</td>
<td>2/6</td>
<td>2/6</td>
</tr>
<tr>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
</tbody>
</table>
Preemption: Formalization

Waiting

Running

<table>
<thead>
<tr>
<th>Jerry</th>
<th>Kevin</th>
<th>Dave</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/6</td>
<td>3/6</td>
<td></td>
</tr>
<tr>
<td>2/6</td>
<td>2/6</td>
<td></td>
</tr>
<tr>
<td>1/6</td>
<td>1/6</td>
<td></td>
</tr>
</tbody>
</table>
Preemption: Formalization

Waiting

Running

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>3/6</th>
<th>3/6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1/6</td>
<td>2/6</td>
</tr>
</tbody>
</table>

JERRY

KEVIN

DAVE

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>2/6</th>
<th>3/6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1/6</td>
<td>2/6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>1/6</th>
<th></th>
</tr>
</thead>
</table>
Preemption: Formalization

Waiting

Running

Jerry: 2/6
1/6

Kevin: 2/6
1/6

Dave: 2/6
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:
  • $\text{DRS}(u) = \max_R \frac{R_u}{R_{Total}}$
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

\[ e \rightarrow f \rightarrow g \rightarrow h \]

Running Jobs

\[ a \quad b \quad c \quad d \]

Waiting Jobs' \[ f \rightarrow g \rightarrow h \]

Running Jobs'

\[ a \quad e \quad d \]
Cook: Architecture

Cook

Matcher

ResourceOffers
LaunchTask

Mesos

Jobs (Running, Waiting)

Ranker

Scored Jobs (Sorted)

Rebalancer

KillTask
Are we doing better?

- Quota
- Dynamic Quota
- Preemption?

Chart showing fairness and utilization metrics for Quota, Dynamic Quota, and Preemption?
Yes!
It works!
It works!
Think Beyond Cook

- DCRS in Mesos?
Open Source

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