Good morning!

CS 744: DRF

Shivaram Venkataraman
Fall 2021
ADMINISTRIVIA

- Assignment 2 out!  → Python  ≈ 2 weeks
- Course Project
  - Project list by Oct 1
  - Form groups and submit project bids by Oct 8
  - Assigned project by Oct 14
  - Introductions due Oct 25
SETTING: FAIR SHARING

Equal Share

Max-Min Share

Maximize the allocation for most poorly treated users

Maximize the minimum
MOTIVATION: MULTI RESOURCES

- Very little mix of CPUs and memory.
- Lots of small tasks.
- Likely to reduce lot of memory, very little CPU.

Graph showing the relationship between per task CPU demand (cores) and per task memory demand (GB), with two distinct clusters labeled 'Maps' and 'Reduces'.
Users have a demand vector
\[<2, 3, 1>\] means user’s task needs 2 R1, 3 R2, 1 R3

Resources given in multiples of demand vector
i.e., users might get \(<4,6,2>\)

\(2\) CPUs, \(3\) GB, \(1\) disk space

\(\Rightarrow\) can run multiple tasks of \(<2, 3, 1>\) multi-resource
**Properties**

**Sharing Incentive**
Users should not be worse off when sharing the cluster.

**Strategy Proof**
Users cannot get more resources by lying.

**Pareto Efficiency**
Cannot give resources to one user without taking away from another.

**Envy free**
User does not wish for another user's allocation.
PROPERTIES

Sharing Incentive
User is no worse off than a cluster with $1/n$ resources

Strategy Proof
User should not benefit by lying about demands

Pareto Efficiency
Not possible to increase one user without decreasing another

Envy free
User should not desire the allocation of another user
DRF: APPROACH

Dominant Resource

Resource user has the biggest share of

Total: <10 CPU, 4 GB>
User 1: <1 CPU, 1 GB> = \frac{1}{10} < \frac{1}{4}
Dominant resource is memory

Dominant Share

Fraction of the dominant resource user is allocated

E.g., for User 1 this is 25% or 1/4

Allocate 2 tasks with

<1 CPU, 1 GB> = \frac{2}{4}
Equalize the dominant share of users

<table>
<thead>
<tr>
<th>User</th>
<th>Allocation</th>
<th>Dominant Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>User1</td>
<td>&lt;0 CPU, 0 GB&gt;</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>&lt;1 CPU, 4 GB&gt;</td>
<td>4/18 = 2/9</td>
</tr>
<tr>
<td></td>
<td>&lt;2 CPU, 8 GB&gt;</td>
<td>8/18 = 4/9</td>
</tr>
<tr>
<td></td>
<td>&lt;3 CPU, 12 GB&gt;</td>
<td>12/18 = 6/9</td>
</tr>
<tr>
<td>User2</td>
<td>&lt;0 CPU, 0 GB&gt;</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>&lt;3 CPU, 1 GB&gt;</td>
<td>3/9</td>
</tr>
<tr>
<td></td>
<td>&lt;6 CPU, 2 GB&gt;</td>
<td>6/9</td>
</tr>
</tbody>
</table>

Total Used: 9 CPU, 14 GB
Total:  <9 CPU, 18 GB>

User1: <1 CPU, 4 GB> per task
   <3 CPU, 12 GB> for 3 tasks
   dom res: mem
   dom share: 12/18 = 2/3

User2: <3 CPU, 1 GB>
   <6 CPU, 2 GB> for 2 tasks
   dom res: CPU
   dom share: 6/9 = 2/3
Whenever there are available resources:
Schedule a task to the user with smallest dominant share
Algorithm 1 DRF pseudo-code

\[ R = \langle r_1, \cdots, r_m \rangle \quad \triangleright \text{total resource capacities} \]
\[ C = \langle c_1, \cdots, c_m \rangle \quad \triangleright \text{consumed resources, initially 0} \]
\[ s_i \quad (i = 1..n) \quad \triangleright \text{user } i \text{'s dominant shares, initially 0} \]
\[ U_i = \langle u_{i,1}, \cdots, u_{i,m} \rangle \quad (i = 1..n) \quad \triangleright \text{resources given to user } i, \text{initially 0} \]

**pick** user \( i \) with lowest dominant share \( s_i \)

\( D_i \leftarrow \text{demand of user } i \text{'s next task} \)

**if** \( C + D_i \leq R \) **then**

\[ C = C + D_i \quad \triangleright \text{update consumed vector} \]
\[ U_i = U_i + D_i \quad \triangleright \text{update } i \text{'s allocation vector} \]
\[ s_i = \max_{j=1}^{m} \{ u_{i,j}/r_j \} \quad \triangleright \text{update their dominant share} \]

**else**

**return** 

\[ \text{资源不足} \quad \triangleright \text{the cluster is full} \]

**end if**
Asset Fairness: Equalize each user's sum of resource shares

Consider total of 70 CPUs, 70 GB RAM
U1 needs <2 CPU, 2 GB RAM> per task \( \leq 4 \)
U2 needs <1 CPU, 2 GB RAM> per task \( \leq 3 \)

Asset Fair Allocation:

U1: 15 tasks: 30 CPU, 30 GB (Sum = 60)
U2: 20 tasks: 20 CPU, 40 GB (Sum = 60)

\[ v_1 = 4, 8, \ldots, 60 \]
\[ v_2 = 3, 6, \ldots, 60 \]
Asset Fairness: Equalize each user’s sum of resource shares
Violates Sharing Incentive

Consider total of 70 CPUs, 70 GB RAM
U1 needs <2 CPU, 2 GB RAM> per task
U2 needs <1 CPU, 2 GB RAM> per task

Sharing incentive?
Half of the cluster is 35 CPU, 35 GB RAM
U1: 17 tasks
U2: 17 tasks
COMPARISON: CEEI

CEEI: Competitive Equilibrium from Equal Incomes

- Each user receives initially $1/n$ of every resource,
- Subsequently, each user can trade resources with other users in a perfectly competitive market
- Computed by maximizing product of utilities across users
Utility: Number of tasks user gets

COMPARISON: CEEI

Total: <9 CPU, 18 GB>  User1: <1 CPU, 4 GB>  User2: <3 CPU, 1 GB>

\[ \begin{align*}
\text{max } (x \cdot y) \\
\text{subject to } \\
x + 3y & \leq 9 \quad \text{CPU} \\
4x + y & \leq 18 \quad \text{Mem}
\end{align*} \]

User 1:
\[ \begin{align*}
x &= 4.05 \\
y &= 1.62
\end{align*} \]

y = 1.62

\[ \begin{align*}
\text{CPU} &: 66\% \\
\text{mem} &: 34\%
\end{align*} \]

\[ \begin{align*}
\text{CPU} &: 55\% \\
\text{mem} &: 45\%
\end{align*} \]
CEEI: STRATEGY PROOFNESS

Total: <9 CPU, 18 GB>

User2 Before:
CEEI: 55% CPU, 9% mem

Total: <9 CPU, 18 GB>
User1: <1 CPU, 4 GB>
User2: <3 CPU, 2 GB>

\[\begin{align*}
\max \quad x \cdot y \\
\text{s.t.} \\
x + 3y &\leq 9 \\
4x + 2y &\leq 18 \\
\end{align*}\]

\[x = 3.6, \quad y = 1.8\]
## COMPARISON

<table>
<thead>
<tr>
<th>Property</th>
<th>Allocation Policy</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Asset</td>
<td>CEEI</td>
<td>DRF</td>
</tr>
<tr>
<td>Sharing Incentive</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Strategy-proofness</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Envy-freeness</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Pareto efficiency</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Single Resource Fairness</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Bottleneck Fairness</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Population Monotonicity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Resource Monotonicity</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Properties of Asset Fairness, CEEI and DRF.
SUMMARY

DRF: Dominant Resource Fairness
Allocation policy for scheduling
Provides multi-resource fairness
Ensures sharing incentive, strategy proofness
DISCUSSION

https://forms.gle/wdN8bwqxEwjPEcAq7
Consider a system with 40 units of CPU, 20 units of memory and 160 units of disk. Consider three users with the following requirements:

- Alice (4 CPU, 1 memory, 1 disk)
- Bob (1 CPU, 4 memory and 4 disk)
- Carol (1 CPU, 2 memory and 16 disk)

List the dominant resource as defined in DRF for Alice, Bob and Carol:

Alice: \( \frac{4}{40}, \frac{1}{20}, \frac{1}{160} \)
Bob: \( \frac{1}{40}, \frac{4}{15}, \frac{4}{160} \)
Carol: \( \frac{1}{40}, \frac{2}{10}, \frac{16}{10} \)
What would be the final task allocation in the given cluster for Alice, Bob and Carol?

<table>
<thead>
<tr>
<th>Task Allocation</th>
<th>Alice</th>
<th>Dom. Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>4, 1, 1</td>
<td>1/10</td>
</tr>
<tr>
<td>Task 2</td>
<td>8, 2, 2</td>
<td>2/10 = 1/5</td>
</tr>
<tr>
<td>Task 3</td>
<td>12, 3, 3</td>
<td>3/10</td>
</tr>
<tr>
<td>Task 4</td>
<td>16, 4, 4</td>
<td>4/10</td>
</tr>
</tbody>
</table>

- **Alice** has 4 tasks, with 3 tasks in the static 1/3rd cluster.

<table>
<thead>
<tr>
<th>Task Allocation</th>
<th>Bob</th>
<th>Dom. Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>1, 4, 4</td>
<td>1/5</td>
</tr>
<tr>
<td>Task 2</td>
<td>2, 8, 8</td>
<td>2/5 = 4/10</td>
</tr>
</tbody>
</table>

- **Bob** has 3 tasks, but only 1 task in the static 1/3rd cluster.

<table>
<thead>
<tr>
<th>Task Allocation</th>
<th>Carol</th>
<th>Dom. Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>1, 2, 16</td>
<td>1/10</td>
</tr>
<tr>
<td>Task 2</td>
<td>2, 4, 32</td>
<td>1/5</td>
</tr>
<tr>
<td>Task 3</td>
<td>3, 6, 48</td>
<td>3/10</td>
</tr>
<tr>
<td>Task 4</td>
<td>4, 8, 64</td>
<td>4/10</td>
</tr>
</tbody>
</table>

- **Carol** has 4 tasks, with 3 tasks in the static 1/3rd cluster.

**Total:** 40, 20, 160

**1/3rd Cluster:**
- Alice: 3 tasks
- Bob: 1 task
- Carol: 3 tasks
What could be one workload / cluster scenario where DRF implemented on Mesos will NOT be optimal?

Placement preferences or locality
Next Week: Machine Learning
Assignment 2 out!