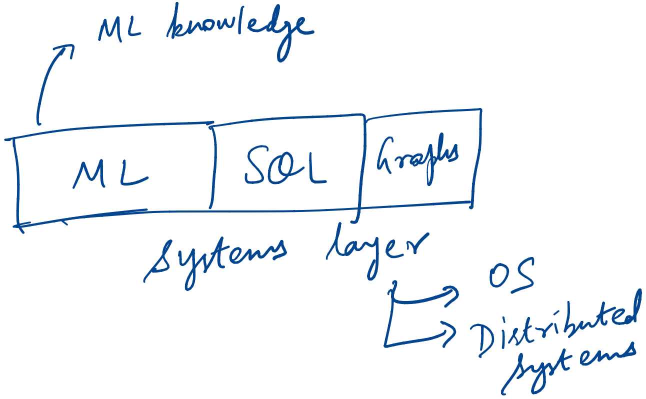
Good morning!

# CS 744: DRF

Shivaram Venkataraman

Fall 2020

# ADMINISTRIVIA




- Assignment 2 out!

→ Piazza

- Course Project
  - Form groups?
  - Project list by Monday (9/28)
  - Submit project bids by Thursday (10/1)
  - Assigned project by Friday (10/2)

~3 students

→ Google Form

Google Cloud

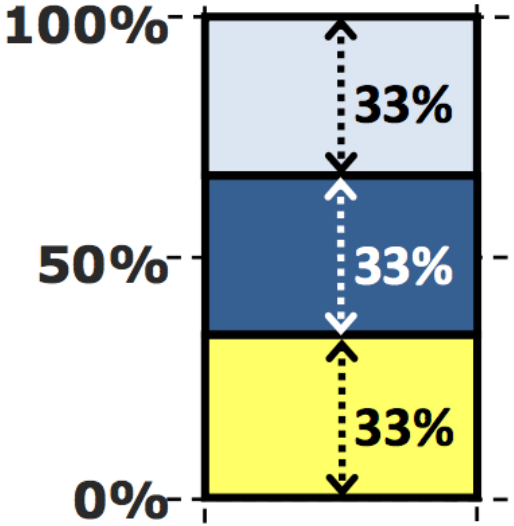
~ 2 months to work on the project

# SETTING: FAIR SHARING

3 users ⇒ each user = 1/3rd of resources

networking
OS → lottery scheduling

## Equal Share

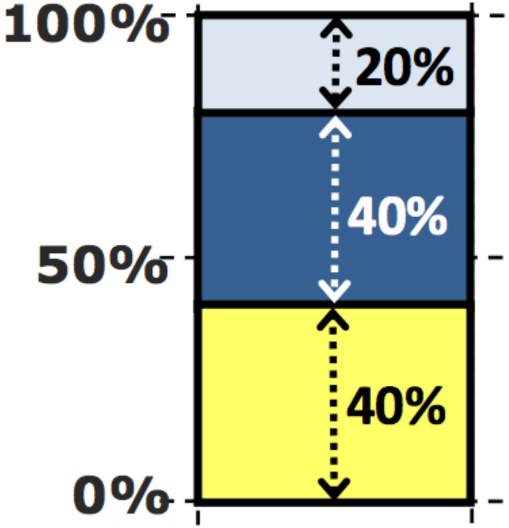



## Max-Min Share

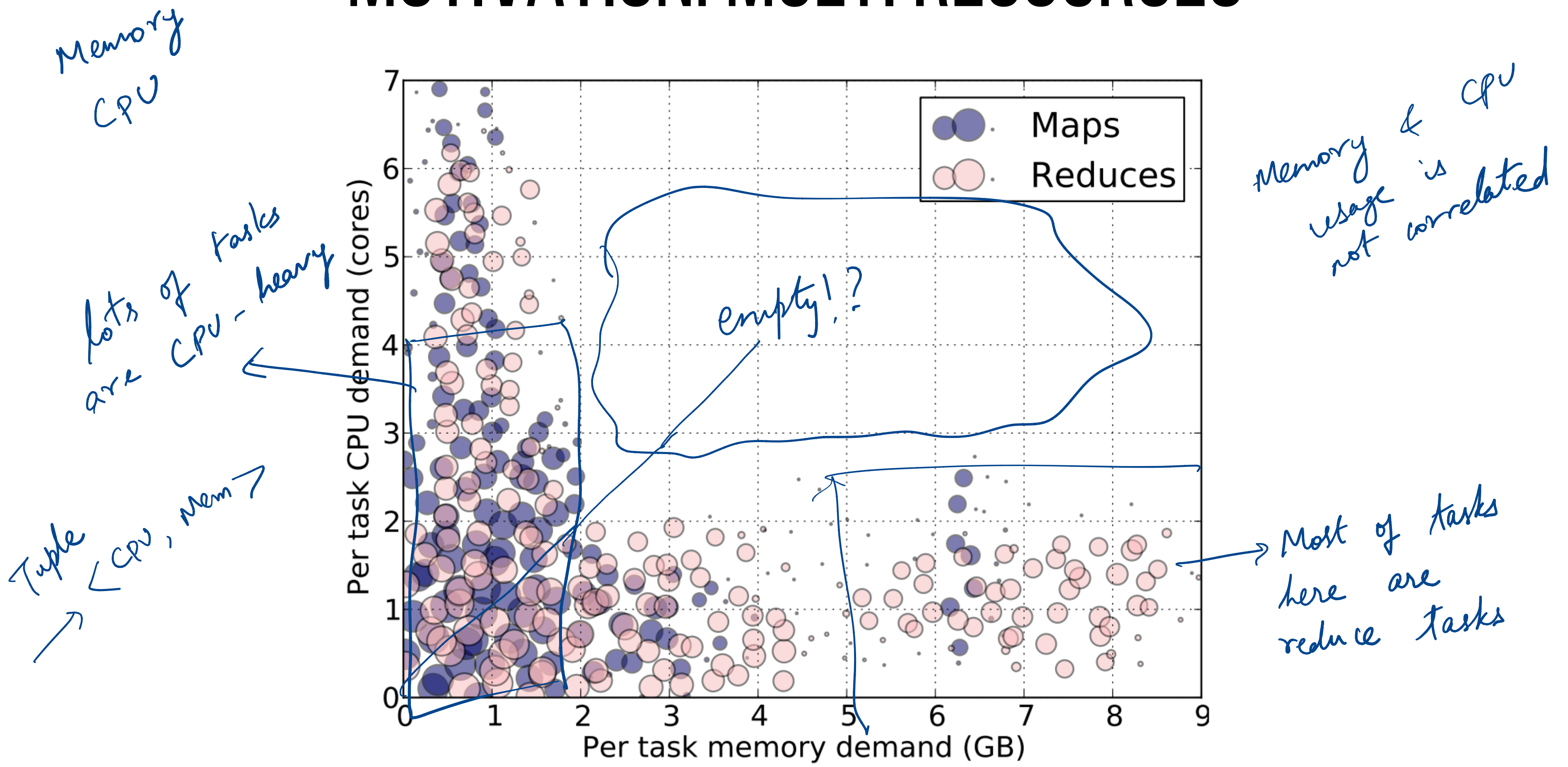
→ to handle different demands across users

Maximize the allocation for most poorly treated users

Maximize the minimum



# MOTIVATION: MULTI RESOURCES



# DRF: MODEL

Users have a demand vector

<2, 3, 1> means user's task needs 2 R1, 3 R2, 1 R3 → one task

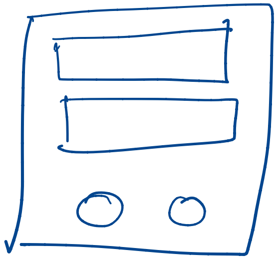
Resources given in multiples of demand vector

i.e., users might get <4,6,2> ← 2 tasks with their demand

slot based model

6 cores

10 GB



4 Map slots = 1 core 1.5 GB of mem

2 reduce slots

No containers

cgroups

linux

# PROPERTIES

## Sharing Incentive

User should get at least $1/n^{th}$ of resources

No worse off than their own cluster with $1/n^{th}$ resources

## Strategy Proof

You can't lie about what you need to get more

Incentivizing truth telling

## Pareto Efficiency

If you allocate more for one user, you need to take away from others

Utilization

## Envy free

Users should not envy allocation of another user

# 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>

Dominant resource is memory

cpu share $\frac{1}{10}$ mem share $\frac{1}{4}$

## Dominant Share

Fraction of the dominant resource user is allocated

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

# DRF: APPROACH

## Equalize the dominant share of users

Total: <9 CPU, 18 GB>

User1: <1 CPU, 4 GB>
dom res: mem

User2: <3 CPU, 1 GB>
dom res: CPU

Total used: 9 CPU, 14 GB

| User | Allocation | Dominant Share |
| --- | --- | --- |
| User1 | <0 CPU, 0 GB><br><1 CPU, 4GB><br><2 CPU, 8GB><br>[<3 CPU, 12GB>] | 0<br>2/9<br>4/9<br>6/9 |
| User2 | <0 CPU, 0 GB><br><3 CPU, 1GB><br>[<6 CPU, 2GB>] | 0<br>3/9<br>6/9 |

# DRF: APPROACH

Total: <9 CPU, 18 GB>

User1: <1 CPU, 4 GB> per task

<3 CPU, 12 GB> for 3 tasks

dom res: mem

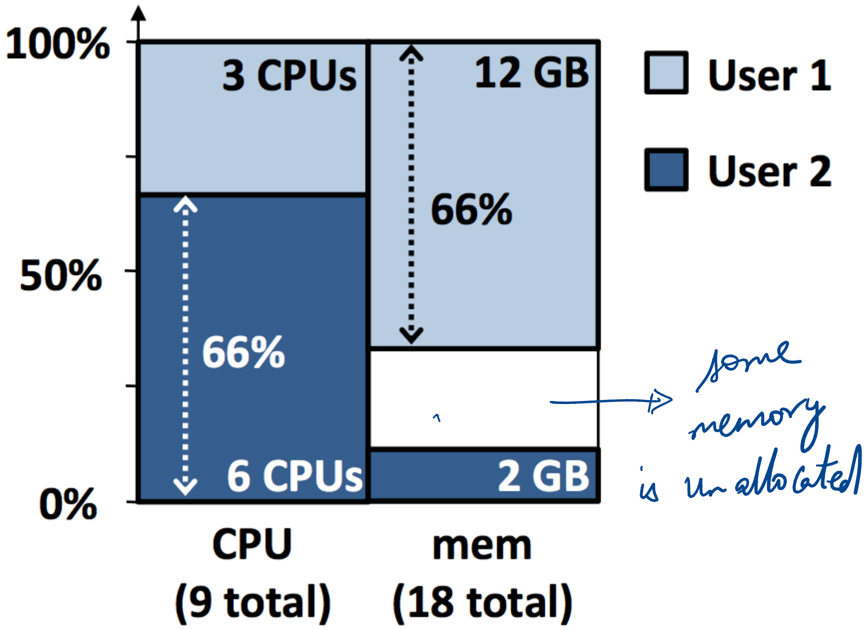
dom share: 12/18 = 2/3

User2: <3 CPU, 1 GB>

<6 GPU, 2 GB> for 2 tasks

dom res: CPU

dom share: 6/9 = 2/3



# DRF ALGORITHM

Whenever there are available resources:

Schedule a task to the user with smallest dominant share

# DRF ALGORITHM

**Algorithm 1** DRF pseudo-code

$R = \langle r_1, \cdots, r_m \rangle$ ▷ total resource capacities

$C = \langle c_1, \cdots, c_m \rangle$ ▷ consumed resources, initially 0

$s_i \ (i = 1..n)$ ▷ user $i$'s dominant shares, initially 0

$U_i = \langle u_{i,1}, \cdots, u_{i,m} \rangle \ (i = 1..n)$ ▷ resources given to user $i$, initially 0

**pick** user $i$ with lowest dominant share $s_i$

$D_i \leftarrow$ demand of user $i$'s next task

**if** $C + D_i \leq R$ **then**

  $C = C + D_i$ ▷ update consumed vector

  $U_i = U_i + D_i$ ▷ update $i$'s allocation vector

  $s_i = \max_{j=1}^{m} \{u_{i,j}/r_j\}$

**else**

  **return** ▷ the cluster is full

**end if**

cluster config

running sum

initialization

track resources given to each user

<1 CPU, 4GB>

if one resource($R_i$) is full you can still offer resources to tasks that don't need R.

# COMPARISON: ASSET FAIRNESS

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 = 4 units of resource

U2 needs <1 CPU, 2 GB RAM> per task = 3 units of resource

Asset Fair Allocation:

U1: 4, 8, 12 .... 60 ≈ 15 tasks for U1

U2: 3, 6, 9, 12 .... 60 ≈ 20 tasks for U2

# COMPARISON: ASSET FAIRNESS

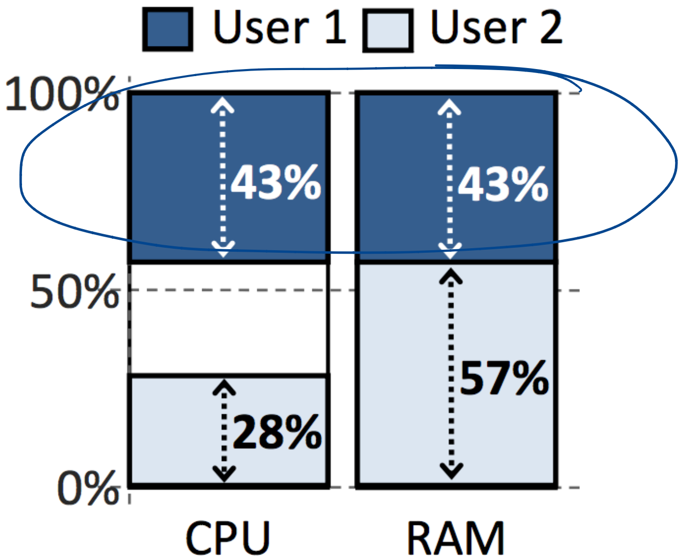
Asset Fairness: Equalize each user's sum of resource shares

Violates Sharing Incentive

U1 will be better off with a 50% cluster that is dedicated!

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

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



# 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

# COMPARISON: CEEI

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

(dominant resource)

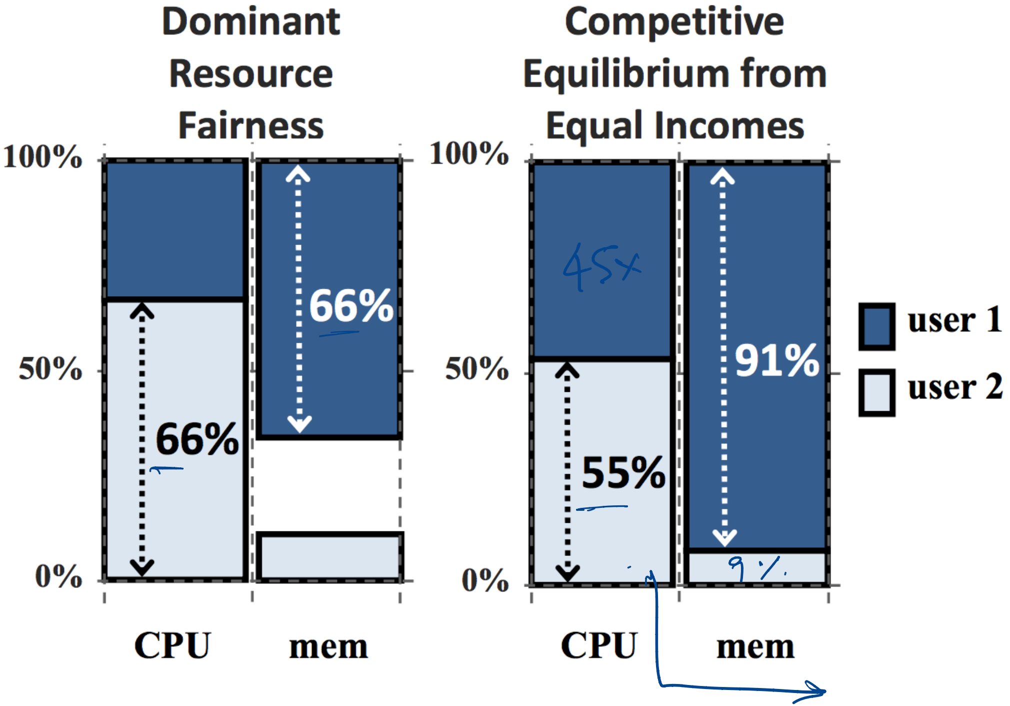
CEEI

$$\max\,(x \cdot y)$$

subject to

$$x + 3y \leq 9 \quad \text{(CPU)}$$
$$4x + y \leq 18 \quad \text{(mem)}$$

$x = 4.05$, $y = 1.62$



# CEEI: STRATEGY PROOFNESS

x·y used to be higher

Total: <9 CPU, 18 GB>

y = 1.62
x = 4.05

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

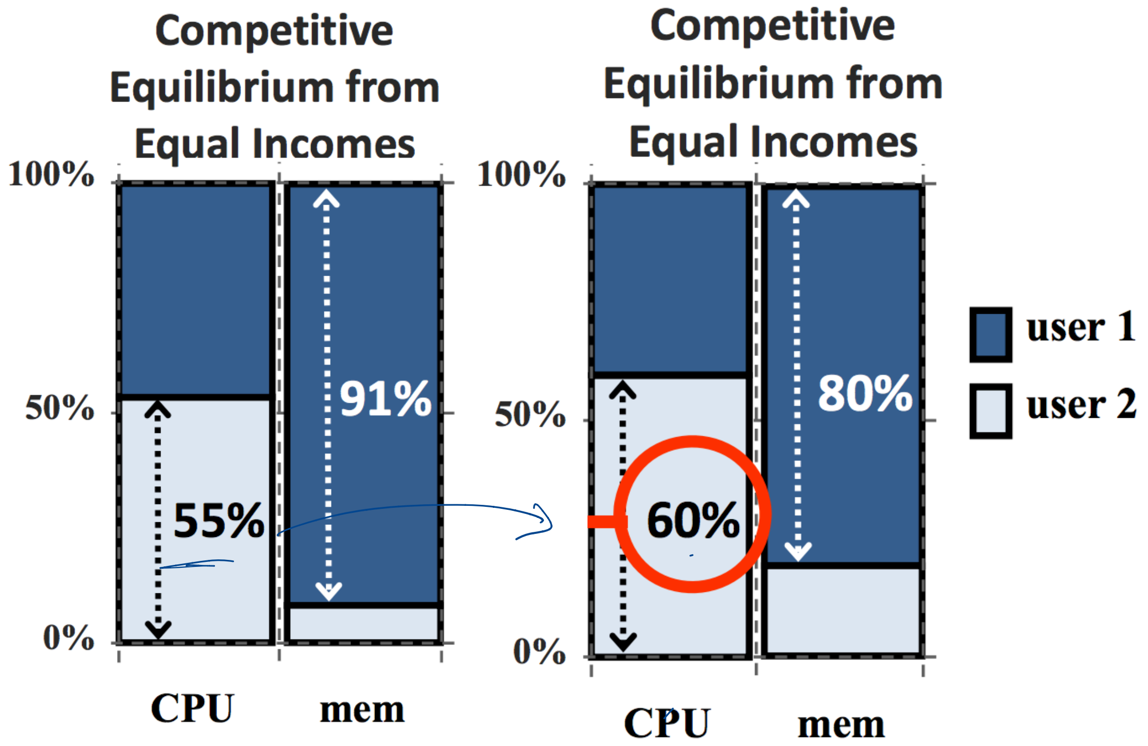
3.6 tasks ??
↳ discrete number of tasks?!

Total: <9 CPU, 18 GB>

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

$x + 3y \le 9$ max $x \cdot y$
$4x + 2y \le 18$

$x = 3.6$
$y = 1.8$



# COMPARISON

| Property | Allocation Policy Asset | CEEI | DRF |
|---|---|---|---|
| Sharing Incentive | | ✓ | ✓ |
| Strategy-proofness | ✓ | | ✓ |
| Envy-freeness | ✓ | ✓ | ✓ |
| Pareto efficiency | ✓ | ✓ | ✓ |
| Single Resource Fairness | ✓ | ✓ | ✓ |
| Bottleneck Fairness | | ✓ | ✓ |
| Population Monotonicity | ✓ | | ✓ |
| Resource Monotonicity | | | |

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

# SUMMARY

DRF: Dominant Resource Fairness

Allocation policy for scheduling

Provides multi-resource fairness → generalizes max-min fairness

Ensures sharing incentive, strategy proofness

# DISCUSSION

https://forms.gle/i7m7xXxKhtfvL9UD9

Consider a system with 100 units of CPU, 50 units of memory and 200 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 : CPU 2/50

Bob : Memory 4/50

Carol : Disk 4/50

# What would be the final task allocation in the given cluster for Alice, Bob and Carol ?

$x, y, z$ num tasks of Alice, Bob, Carol

$$\frac{4x}{100} = \frac{4y}{50} = \frac{16z}{200}$$

$$4x + y + z \le 100$$
$$x + 4y + 2z \le 50$$
$$x + 4y + 16z \le 200$$

replace $x$ in terms of $y$, $z$ & solve

$$x = 12.5$$
$$y = 6.25 = z$$

Every time Alice is allocated : 8/200

Bob = Carol = 16/200

Alice get two turns for every 1 turn of Bob, Carol

⋮

Alice = 12

Bob = 6, Carol = 6

either (14, 6, 6) or (12, 6, 7) ✓

# What could be one workload / cluster scenario where DRF implemented on Mesos will NOT be optimal?

→ If there aren't enough resources
if at least one task can run

→ Heterogeneous tasks
[over time ?]

→ Instantaneously fair ?

→ Locality pref ?

# NEXT STEPS

Mesos:
resource Offer (9 CPU, 18 GB)
→ task <1 CPU, 3 GB> at Ex1

Next Week: Machine Learning

Assignment 2 out!

Strategy proof: assuming rational actors

DRF starvation → one long running tasks & very highly contended cluster

optimization problem
≡ max min <dom share> subject to resource constraints

equalize $\frac{3x}{21} = \frac{4y}{21}$

R: <21, 21>

$D_1$: <3, 2> : $D_2$ <4, 1>

3 tasks 3 tasks

$\simeq \frac{9}{21} = \frac{3}{7}$ $\frac{12}{21} = \frac{4}{7}$

DRF allocation

D1: 3/21
D2: 4/21
D1: 6/21
D2: 8/21
D1: 9/21
D2: 12/21

$\frac{3x}{21} = \frac{4y}{21}$

$3x + 4y \le 21$

??