Hello

CS 744: DRF

Shivaram Venkataraman

Fall 2022

ADMINISTRIVIA

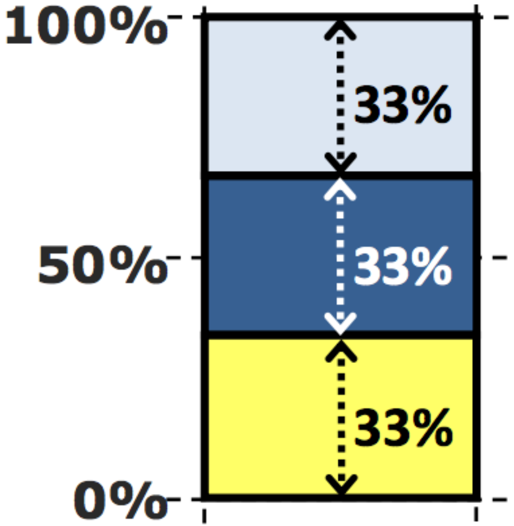
- Assignment 2 out! → ~ 2 weeks Pytorch / Machine learning
- Course Project
  - Project list by Oct 4 → seed ideas
  - Form groups and submit project bids by Oct 11
  - Assigned project by Oct 15
  - Introductions due Oct 25

3 clients in system
one resource

SETTING: FAIR SHARING

→ work conserving
= no resources are wasted

Equal Share

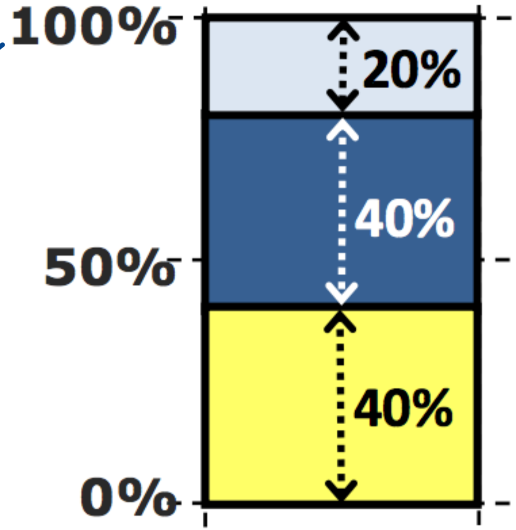


Max-Min Share

→ network
→ CPU lottery scheduling

Maximize the allocation for most poorly treated users

Maximize the minimum



SLOT-BASED MODEL

Slot: Fixed quantity of CPU and memory

Example: Hadoop MapReduce

Mapper: 2 CPU and 1 GB

Reducer: 1 CPU and 2 GB

Allocate in units of slots → 2 Map slots in machine 1 to job 1

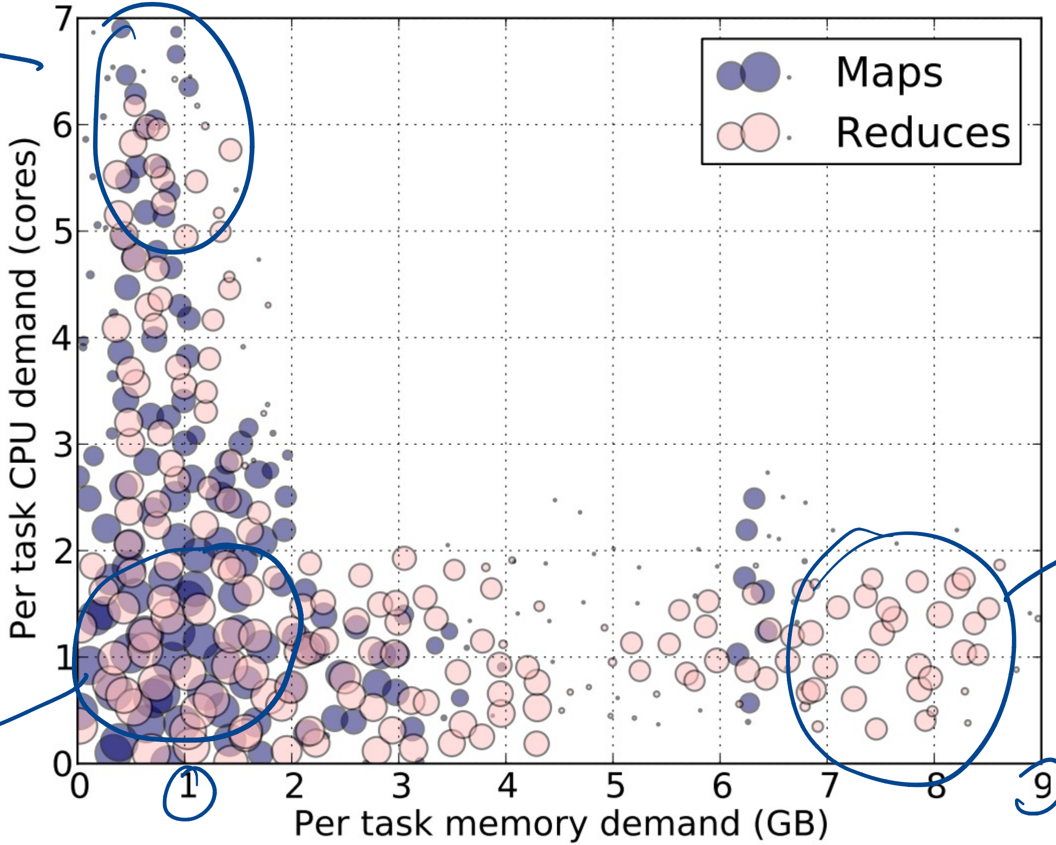
2 Reduce slots of job 2 in machine 1 ...

MOTIVATION: MULTI RESOURCES

slot based model

CPU intensive not memory intensive

large range of resource demands



majority of tasks

Reduce tasks memory intensive

DRF: MODEL

Users have a demand vector

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

<CPU, Mem, Disk> example

Resources given in multiples of demand vector

i.e., users might get <4,6,2>

PROPERTIES

Sharing Incentive

Users should not fare worse than having 1/n cluster

Strategy Proof

Users should not be able to lie to get a large share

Pareto "Efficiency"

We cannot increase allocation of one user without decreasing others

Envy free

Users should not desire 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

Dominant Share

Fraction of the dominant resource user is allocated

Total: <10 CPU, 4 GB>

User 1: <1 CPU, 1 GB>

Dominant resource is memory

$\frac{1}{10} < \frac{1}{4}$ → Dominant Share

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

DRF: APPROACH

x tasks for U1
y tasks for U2

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

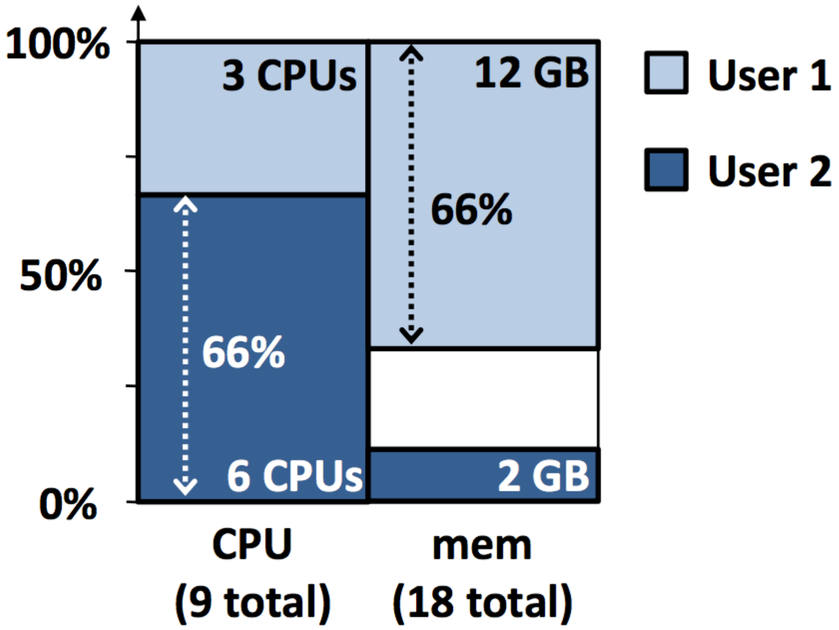
User	Allocation	Dominant Share
User1	<0 CPU, 0 GB> <1 CPU, 4GB> <2 CPU, 8 GB> <3 CPU, 12 GB> 3 tasks	~~0~~ 4/18 = 2/9 8/18 = 4/9 12/18 = 2/3
User2	<0 CPU, 0 GB> <3 CPU, 1GB> <6 CPU, 2 GB> 2 tasks	$\boxed{0}$ 3/9 6/9 = 2/3

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 CPU, 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

similar to max-min fairness but on dominant share

DRF ALGORITHM

n users
m resource types

Algorithm 1 DRF pseudo-code

$R = \langle r_1, \cdots, r_m \rangle$ ▷ total resource capacities → cluster limitation / capacity
$C = \langle c_1, \cdots, c_m \rangle$ ▷ consumed resources, initially 0
$s_i \ (i = 1..n)$ ▷ user i's dominant shares, initially 0 → dominant share
$U_i = \langle u_{i,1}, \cdots, u_{i,m} \rangle \ (i = 1..n)$ ▷ resources given to user i, initially 0 → Prior allocations made

pick user i with lowest dominant share s_i
$D_i \leftarrow$ demand of user i's next task → give me the resource vector
if $\boxed{C + D_i \le R}$ **then** → check if it fits
  $C = C + D_i$ ▷ update consumed vector
  $U_i = U_i + D_i$ ▷ update i's allocation vector
  $\underbrace{s_i = \max_{j=1}^{m}\{u_{i,j}/r_j\}}$ → dominant share update
else
  return ▷ the cluster is full
end if

Instantaneous fairness

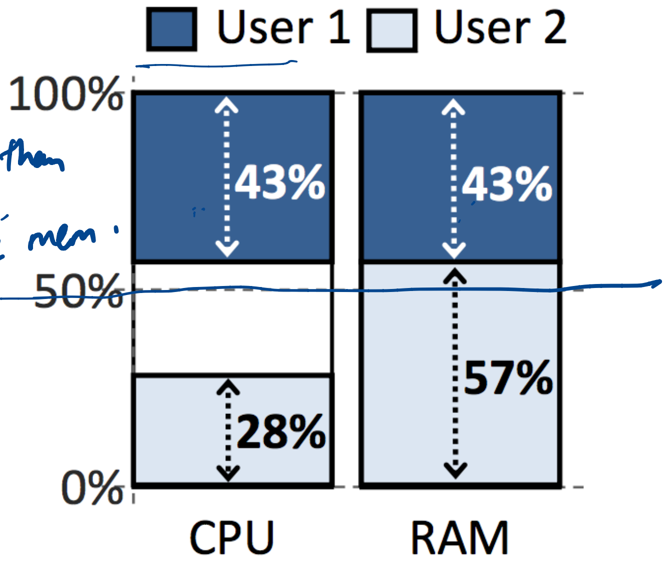
COMPARISON: ASSET FAIRNESS

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
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)

User 1
less than
50%
CPU & mem.



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
U2 needs <1 CPU, 2 GB RAM> per task

Sharing incentive?
Half of the cluster is 35 CPU, 35 GB RAM
U1: 17 tasks 34 CPU, 34 GB RAM → better than 15 tasks with Asset Fairness
U2:

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
- Nash solution: Maximize product of utilities across users

reaches market equilibrium

COMPARISON: CEEI

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

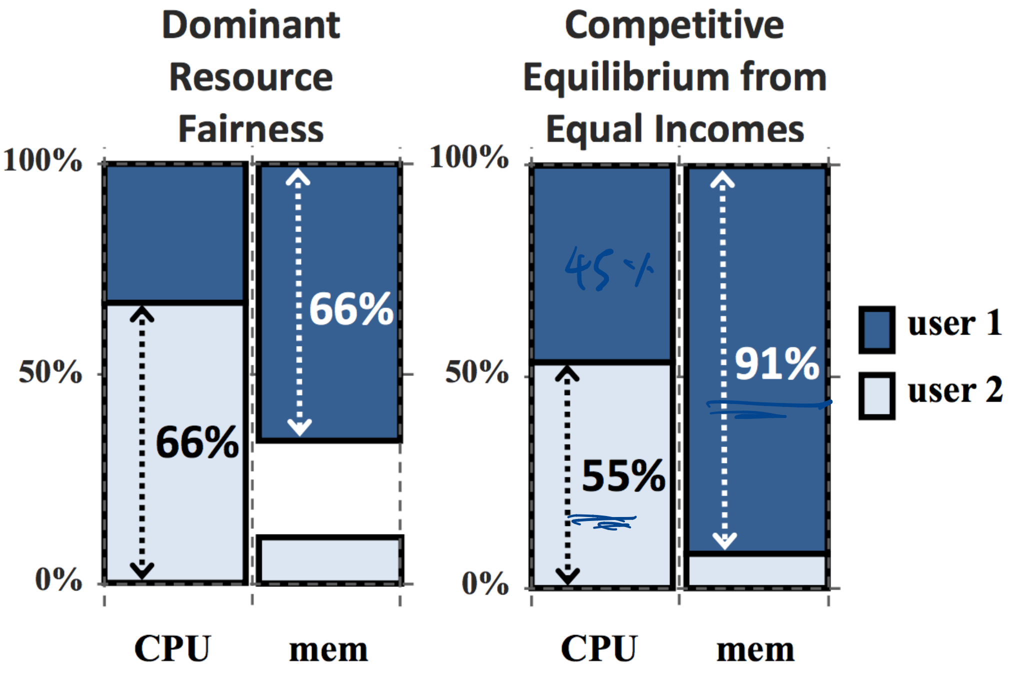
$\max (x \cdot y)$ product of utilities

subject to

$$
\begin{aligned}
x + 3y &\leq 9 \quad \text{CPU} \\
4x + y &\leq 18 \quad \text{Mem}
\end{aligned}
$$

$x = 4.05$

$y = 1.62$



CEEI: STRATEGY PROOFNESS

Total: <9 CPU, 18 GB>

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

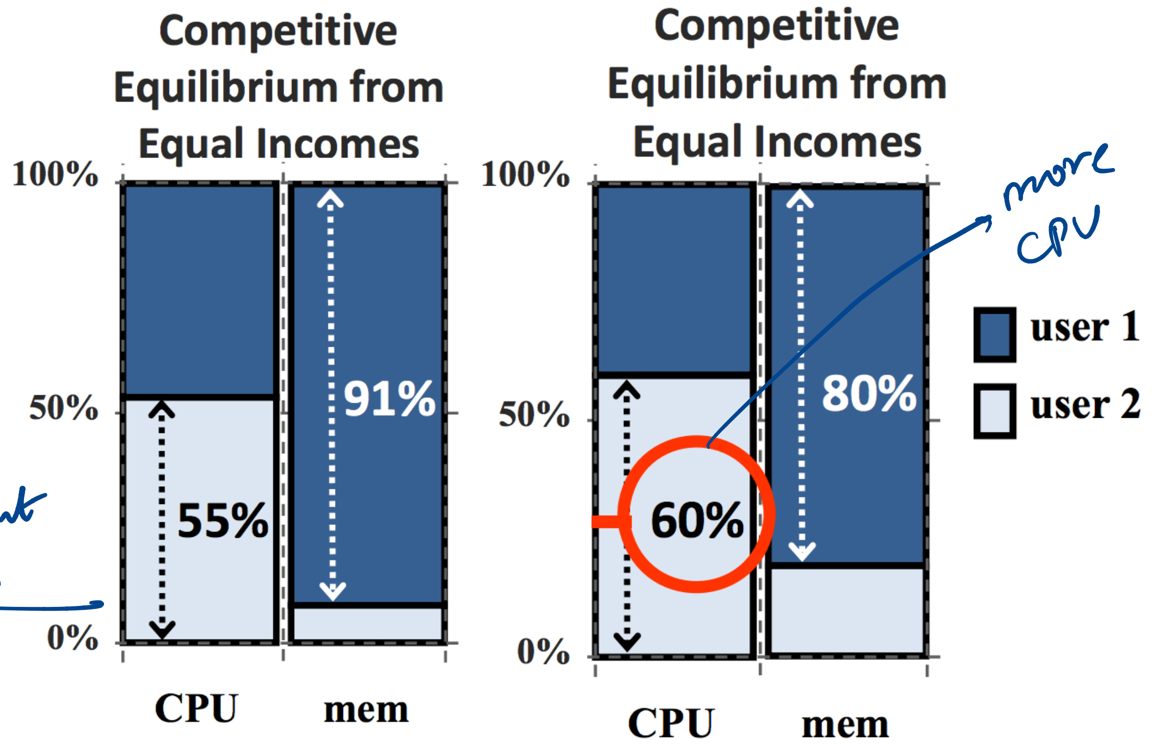
Total: <9 CPU, 18 GB>
User1: <1 CPU, 4 GB>
~~User2: <3 CPU, 1 GB>~~
User2: <3 CPU, 2 GB>

lied about mem req.

max $x \cdot y$

$x = 3.6$ $\quad$ $x + 3y \le 9$

$y = 1.8$ $\quad$ $4x + 2y \le 18$



more CPU

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

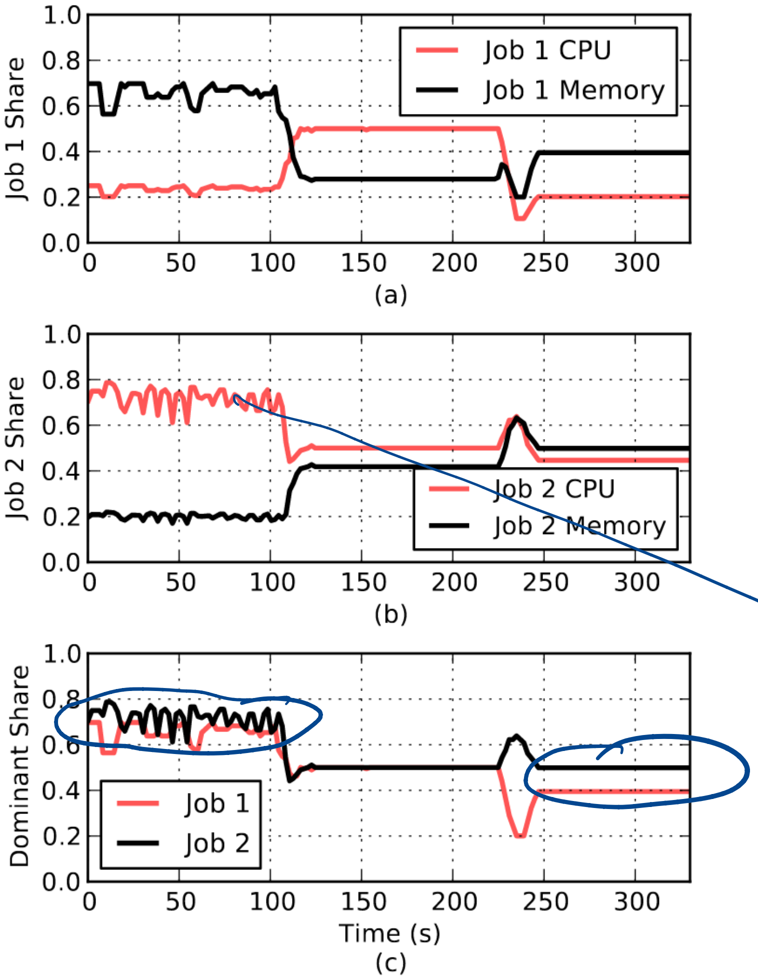
Allocation policy for scheduling

Provides multi-resource fairness

Ensures sharing incentive, strategy proofness

DISCUSSION

https://forms.gle/n97b12Qcs8Xv3C6L6



- Task requirements change and DRF adapts to the changes

- There is a gap between Job 1 / Job 2 (discretization)

- Both jobs have dom. share > 0.5

Task arrival / discretization?

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

- Users diff between dominant resource & next dom. resource is small → will this lead to strategy proof violations??
- If you cannot fit the user with smallest dominant share, resource waste?
- Handling locality / affinity preferences is challenging

NEXT STEPS

Next Week: Machine Learning

Assignment 2 out!