CS 744: DRF

Shivaram Venkataraman

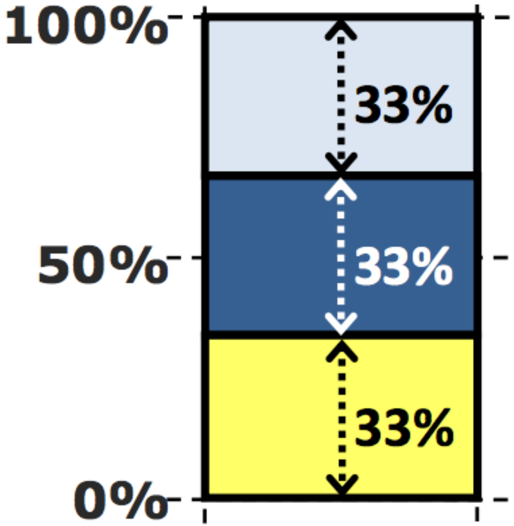
Spring 2024

ADMINISTRIVIA

- Assignment 2 done!
- Course Project
  - Form groups and submit project bids by tonight!
  - Assigned projects by March 1
  - Introductions due March 8

SETTING: FAIR SHARING

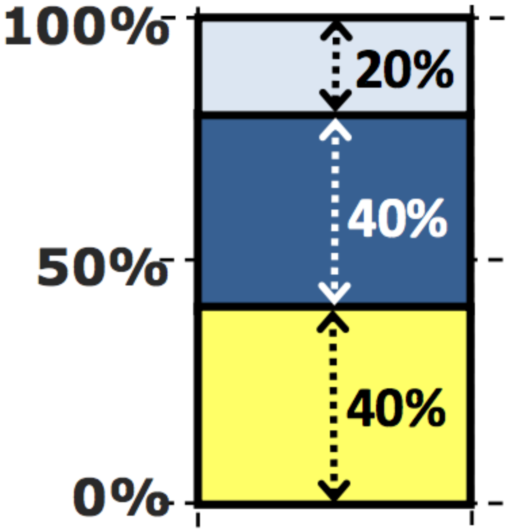
Equal Share



Max-Min Share

Maximize the allocation for most poorly treated users

Maximize the minimum



SLOT-BASED MODEL

Slot: Fixed quantity of CPU and memory

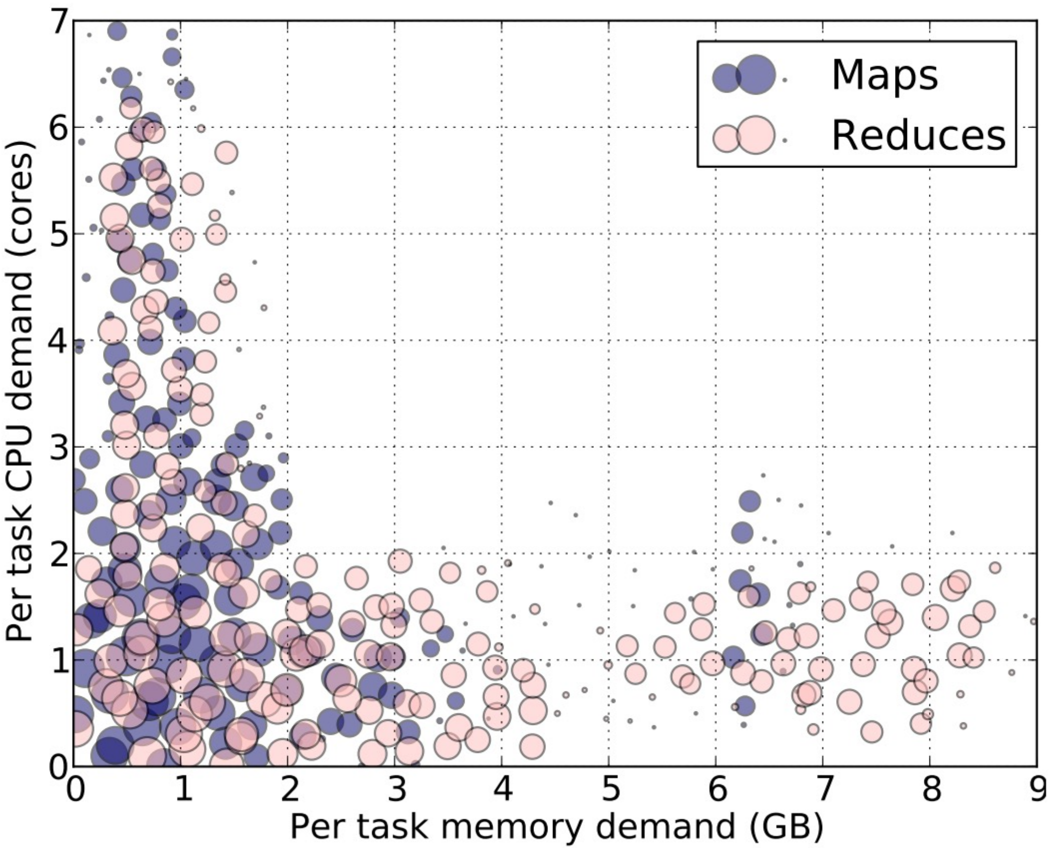
Example: Hadoop MapReduce

Mapper: 2 CPU and 1 GB

Reducer: 1 CPU and 2 GB

Allocate in units of slots

MOTIVATION: MULTI RESOURCES



DRF: MODEL

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>

PROPERTIES

Sharing Incentive

Strategy Proof

Pareto Efficiency

Envy free

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

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

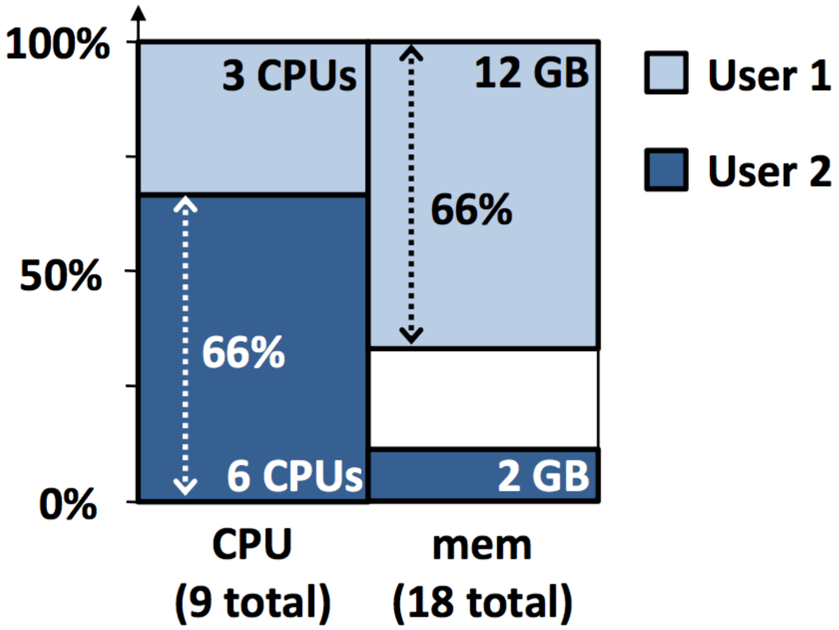
User	Allocation	Dominant Share
User1	<0 CPU, 0 GB>	0
User2	<0 CPU, 0 GB>	0

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

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

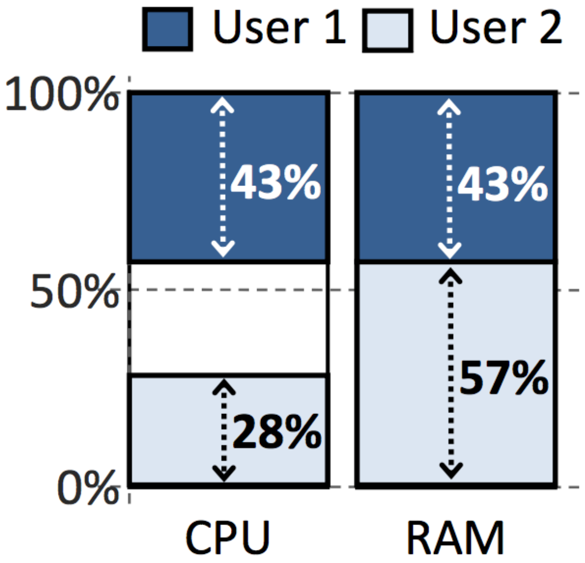
pick user i with lowest dominant share s_i
$D_i \leftarrow$ demand of user i's next task
if $C + D_i \leq R$ **then**
    $C = C + D_i$ ▷ update consumed vector
    $U_i = U_i + D_i$ ▷ update i's allocation vector
    $s_i = \max_{j=1}^{m}\{u_{i,j}/r_j\}$
else
    return ▷ the cluster is full
end if

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)



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:
U2:

COMPARISON: CEEI

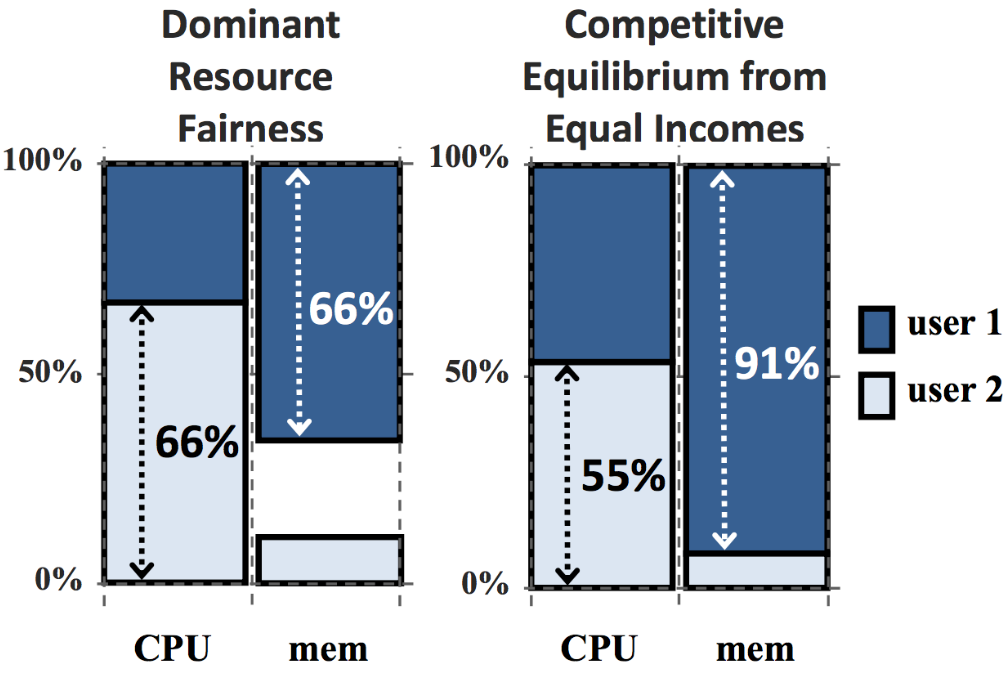
CEEI: Competitive Equilibrium from Equal Incomes

- 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

COMPARISON: CEEI

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

$$\begin{aligned} &\max\,(x \cdot y) \\ &\text{subject to} \\ &\quad x + 3y \leq 9 \\ &\quad 4x + y \leq 18 \end{aligned}$$

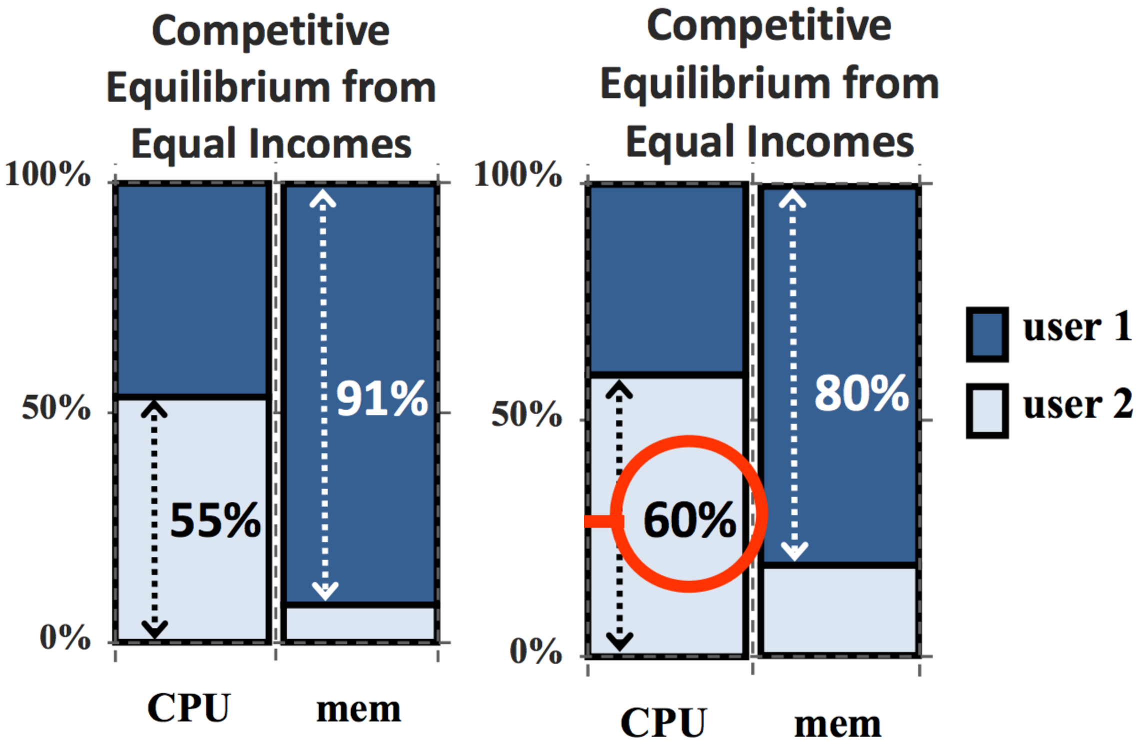


CEEI: STRATEGY PROOFNESS

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>



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

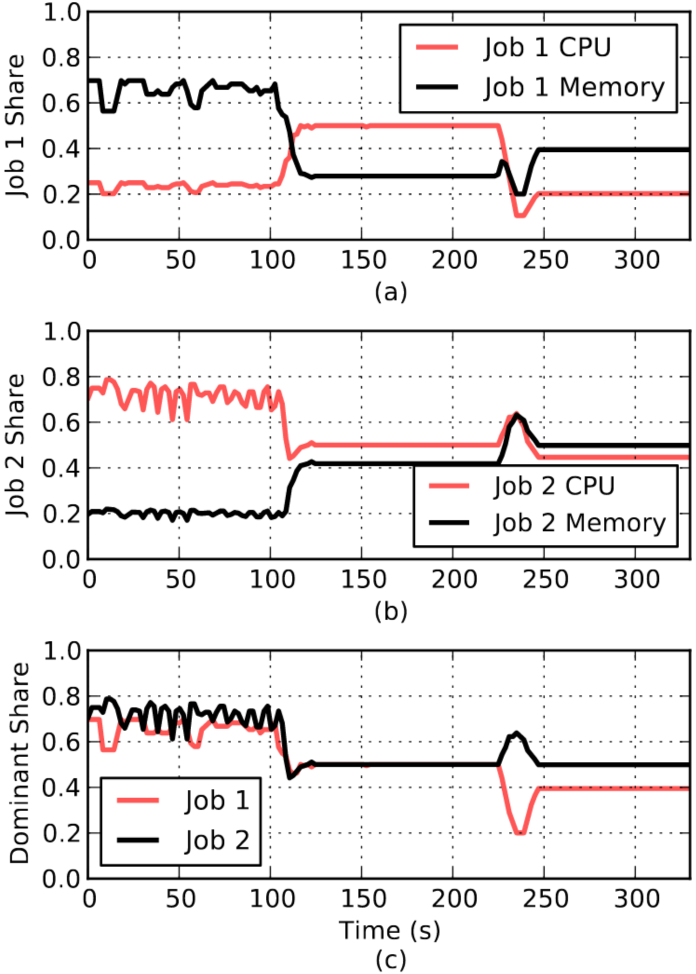
Allocation policy for scheduling

Provides multi-resource fairness

Ensures sharing incentive, strategy proofness

DISCUSSION

https://forms.gle/75faGZ4quQgWSYRQ8



Job 1 CPU
Job 1 Memory
Job 1 Share
(a)
Job 2 CPU
Job 2 Memory
Job 2 Share
(b)
Job 1
Job 2
Dominant Share
Time (s)
(c)

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

NEXT STEPS

Next class: Machine Learning Schedulers