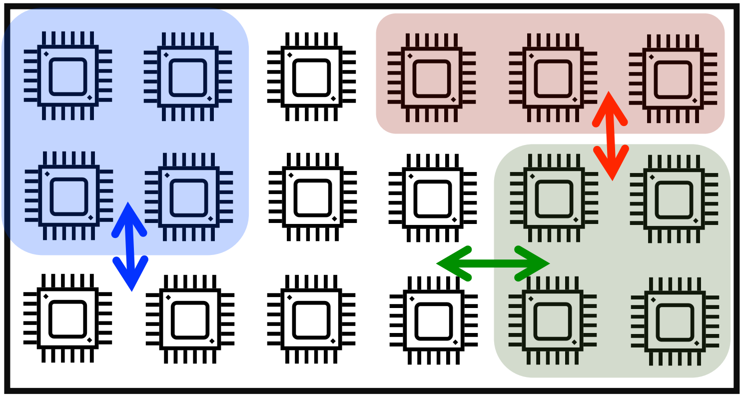
Algorithms for Cloud Computing

part II: Resource Allocation and Fairness

Georgios Paschos

amazon

Resource Allocation



Sharing resources among users

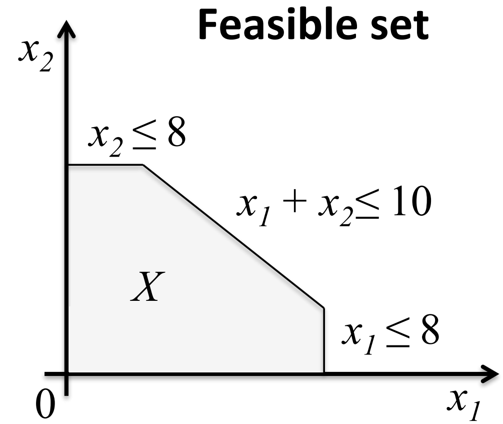
- Each user receives a "satisfaction" from resources
  - Maximize total satisfaction with available resources
  - Caveat: might be *unfair*…

Outline

- **Welfare maximization:** find resource allocations for the system's benefit
- **Fairness** and its relation to welfare maximization
- **Multi-resource fairness**

amazon

Welfare Maximization



- **Problem formulation:**
  - K: number of users
  - x_k: allocated resources to user k
  - X: set of feasible allocations (convex, bounded in positive orthant)
  - U_k: utility of user k (increasing, concave)

$$\max_{x \in \mathcal{X}} \sum_{k=1}^{K} U_k(x_k)$$

- **Convex solvers:** project gradient, Lagrangian relaxation, dual ascent, ADMM, …

amazon

Why care about welfare maximization?

- **Question:** why not just sell resources to the highest bidder?

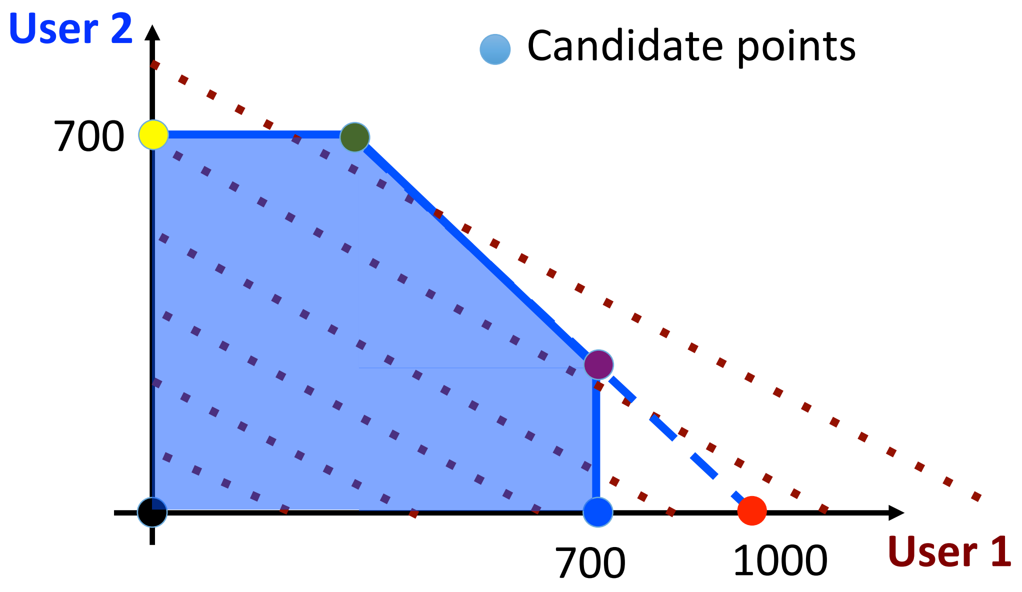
A1: In some systems, we sell SLAs and use available resources to meet them. Welfare helps to distribute surplus, and/or de-risk failing SLAs

A2: Sometimes SLAs can not be met, how do we decide which ones to violate, and by how much?

A3: Even if you do sell, to sell everything and not loose money, you still need to understand welfare maximization

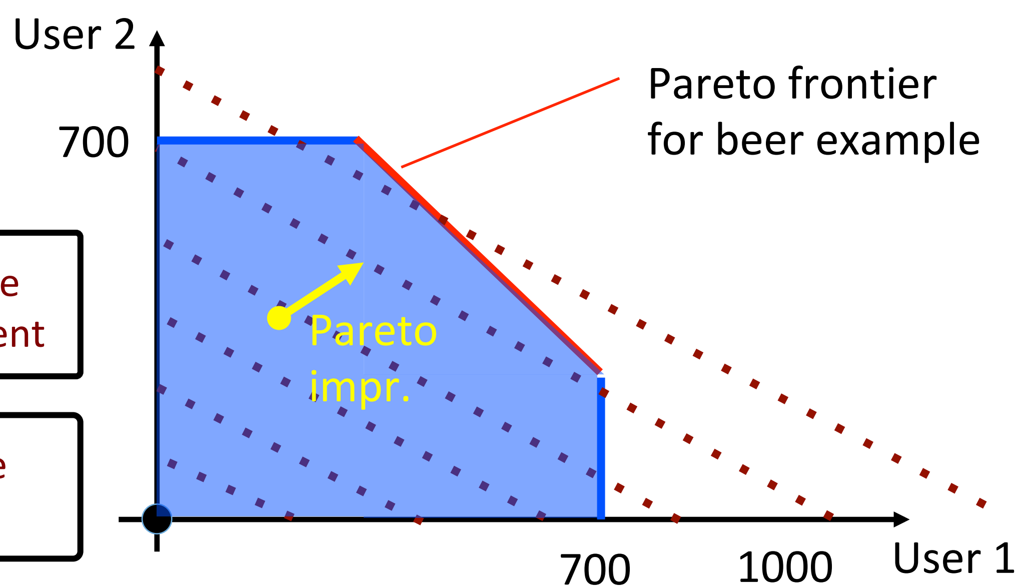
The beer example

- We have 1lt of beer, two glasses of 700ml, and user 2 is twice "thirsty" than user 1. $U(x) = x_1 + 2x_2$
- Questions: Describe feasible set. What point maximizes welfare?



Pareto Efficiency

- **Definition:** A feasible allocation y is a Pareto improvement for x if $U_k(y_k) \geq U_k(x_k), \quad k = 1, \ldots, K,$ and > for at least one
- **Definition:** A point is Pareto efficient if there is no Pareto improvement for it



Lem: The solution of welfare maximization is Pareto efficient

Q: what other points in the example can be Pareto?

amazon

Questions

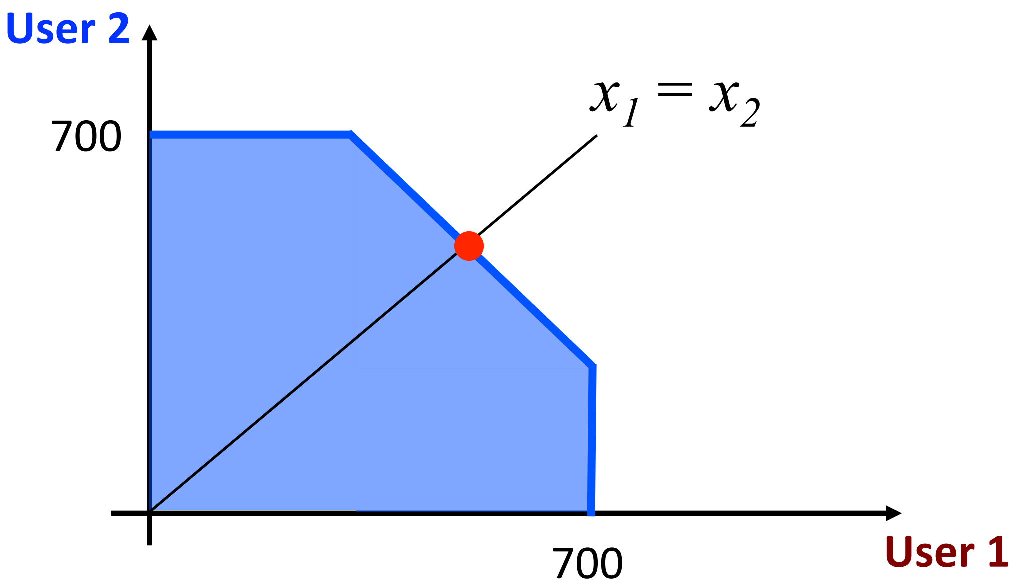
- Two persons, 100 bananas. Characterize the Pareto frontier
- Two persons, 2 bananas, 2 apples. P1 likes bananas and dislikes apples, P2 the opposite. Frontier?
- Two persons, 2 bananas, 2 apples. P1 likes bananas and is indifferent to apples, P2 the opposite. Frontier?
- P1 values 1 banana = 2 apples, and P2 the opposite. Frontier?

Max-min fairness

- **Definition:** A feasible allocation x is Max-Min Fair (MMF) in set X if for any other y it holds:

$$y_m > x_m \quad \Rightarrow \quad \exists\, n \neq m : y_n < x_n \leq x_m$$

To improve the utility of user m, we must worsen the utility of a “poorer” user



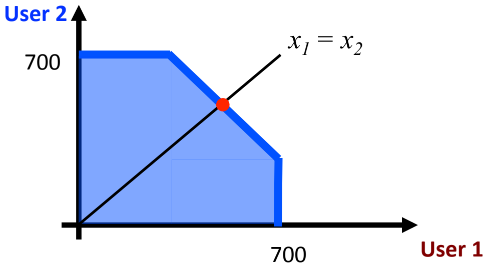
amazon

Max-min fairness: claims

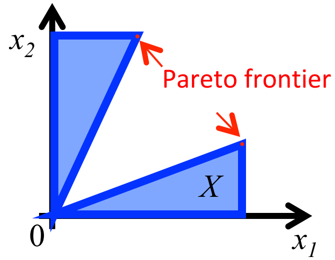
- **Definition:** A feasible allocation x is Max-Min Fair (MMF) in set X if for any other y it holds:

$$y_m > x_m \quad \Rightarrow \quad \exists\, n \neq m : y_n < x_n \leq x_m$$

To improve user m, we must worsen the allocation of a "poorer" user



- If X is convex, there exists a unique MMF
- Non-convex sets might have no MMF (see example)
- MMF is equivalent to max the minimum and subject to that, max the 2nd min, etc..
- If "all equal" is Pareto, it is MMF

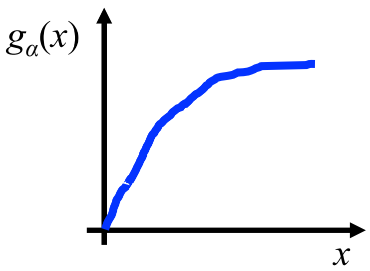


B. Radunovic and J.-Y. Le Boudec, "*A Unified Framework for Max-Min and Min-Max fairness with applications*", IEEE/ACM Trans. on Networking, 2007.

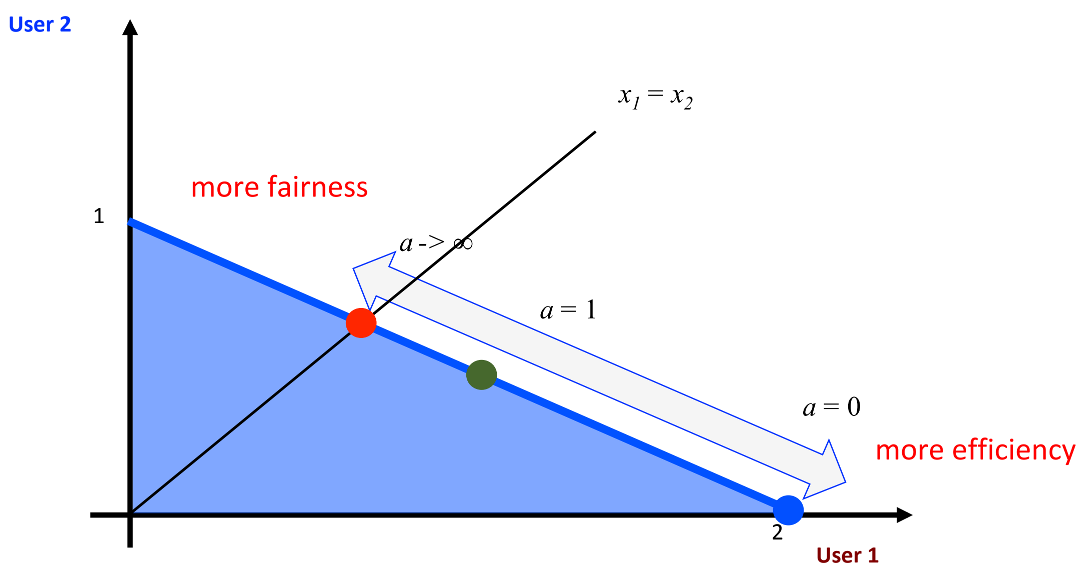
Alpha Fairness

- Family of concave utility functions

$$g_\alpha(x) = \begin{cases} \frac{x^{1-\alpha}}{1-\alpha} & \text{if } \alpha \in [0,1) \cup (1,\infty) \\ \log x & \text{if } \alpha = 1 \end{cases}$$



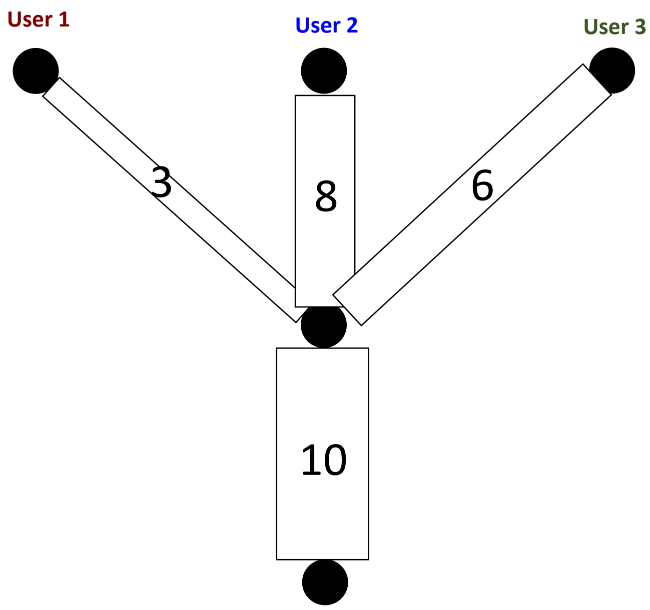
- Strictly convex Welfare $\sum_{k=1}^{K} \frac{x_k^{1-\alpha}}{1-\alpha}$ => unique solution for a>0



J. Mo and J. Walrand, "*Fair End-to-End Window-based Congestion Control*", IEEE/ACM Trans. on Networking, 2000.

Progressive Filling algorithm

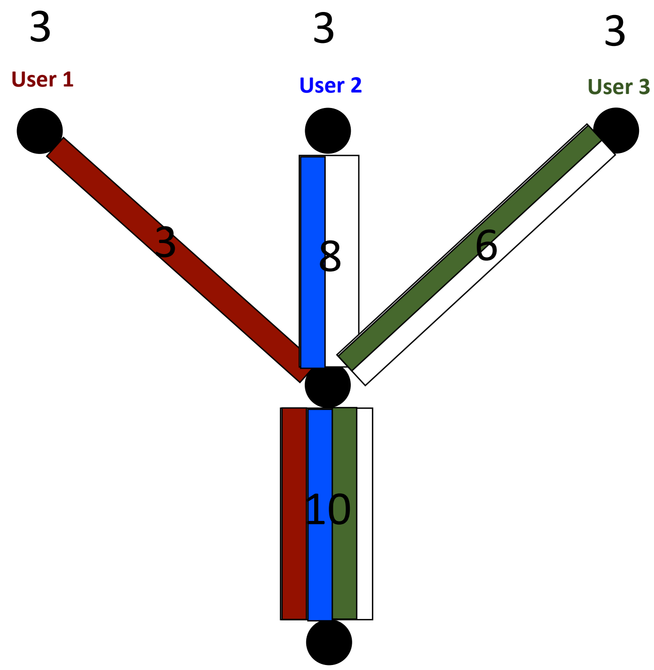
- Increase iteratively allocation until reaching a bottleneck



Progressive Filling algorithm

- Increase iteratively allocation until reaching a bottleneck

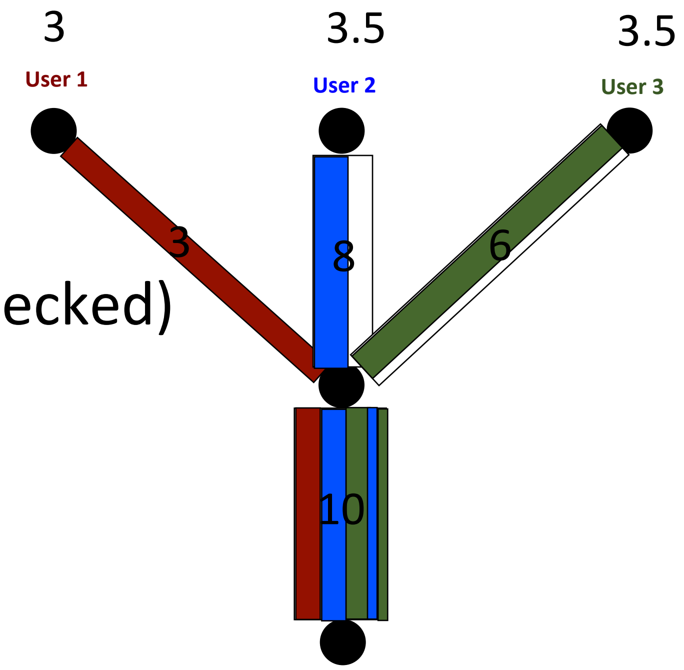
- Step 1: fill up to 3
    - User 1 = 3 (bottlenecked)



Progressive Filling algorithm

- Increase iteratively allocation until reaching a bottleneck

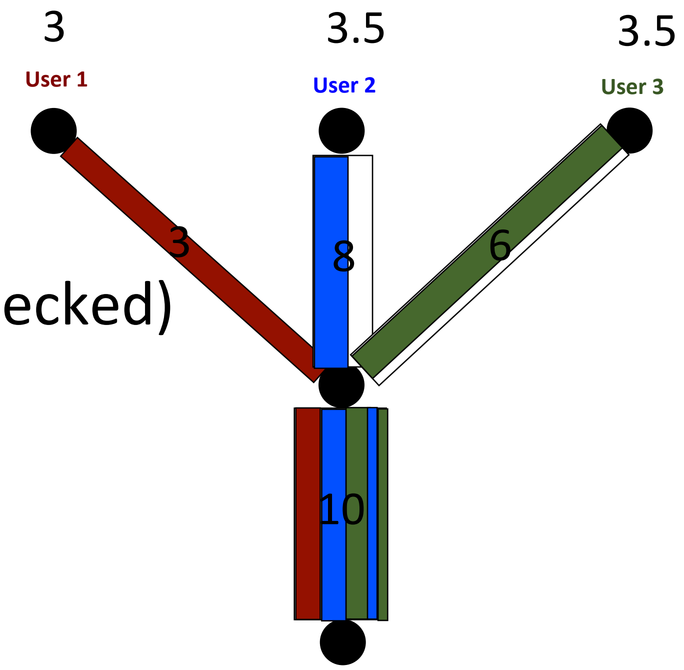
- Step 1: fill up to 3
  - User 1 = 3 (bottlenecked)
- Step 2: fill up to 3.5
  - User 2 and User 3 = 3.5 (bottlenecked)
- Algorithm stops



Progressive Filling algorithm

- Increase iteratively allocation until reaching a bottleneck

- Step 1: fill up to 3
  - User 1 = 3 (bottlenecked)
- Step 2: fill up to 3.5
  - User 2 and User 3 = 3.5 (bottlenecked)
- Algorithm stops

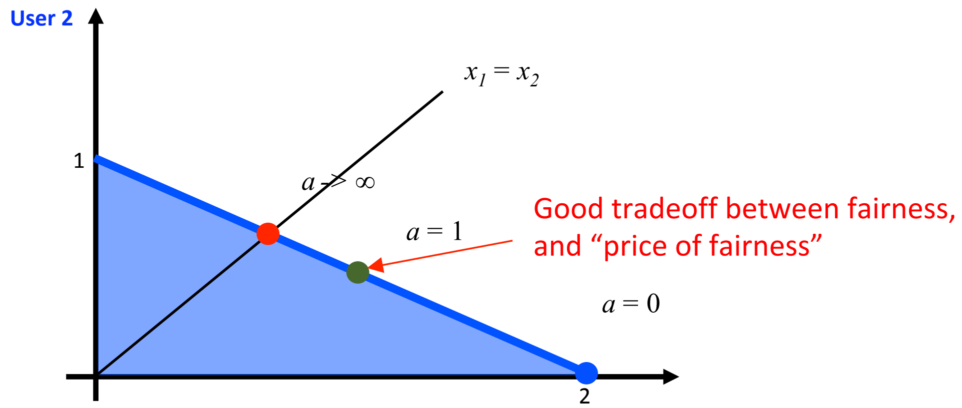


Thm: Progressive Filling converges to MMF.

Proportional fairness (α=1)

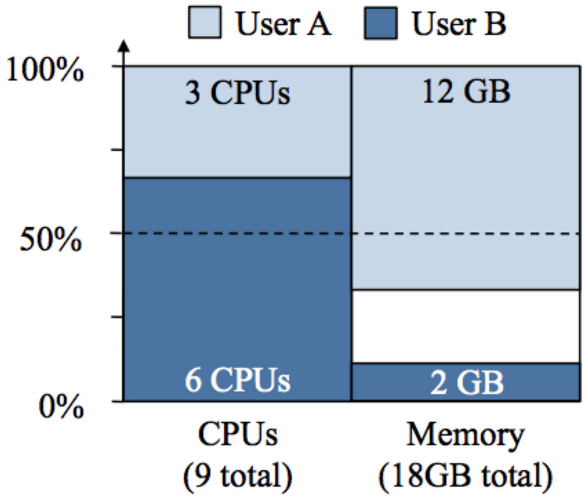
- **Definition:** A feasible allocation x is Proportional Fair (PF) in set X if any other y has a negative average change: $\sum_k \frac{y_k - x_k}{x_k} \leq 0$

Equivalent to solving the welfare maximization for logarithms: $\max_{x \in \mathcal{X}} \sum_{k=1}^{K} \log(x_k)$



Multi-resource Fairness

- Two resources (CPU and memory)
- How to generalize fairness?

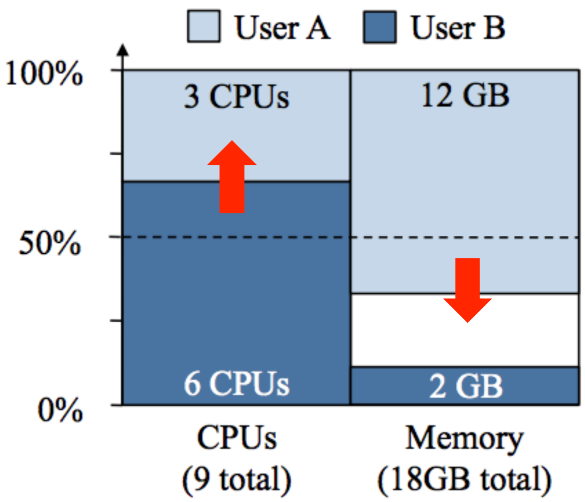


- Simple approach: "**Dominant Resource Fairness**"
  - Each user has a dominant resource share
  - Balance user shares with weighted MMF

A. Ghodsi et al., "*Dominant Resource Fairness: Fair Allocation of Multiple Resource Types*", NSDI, 2011.
T. Bonald and J. Roberts, "*Multi-Resource Fairness: Objectives, Algorithms and Performance*", ACM Sigmetrics, 2015.

Dominant Resource Fairness

- Two users:
  - User A $(w_{11}, w_{12}) = (1,4)$
  "4GBs for each CPU"
  - User B $(w_{21}, w_{22}) = (3,1)$
  "1GB for each 3CPUs"



- Total of 9CPUs and 18GBs

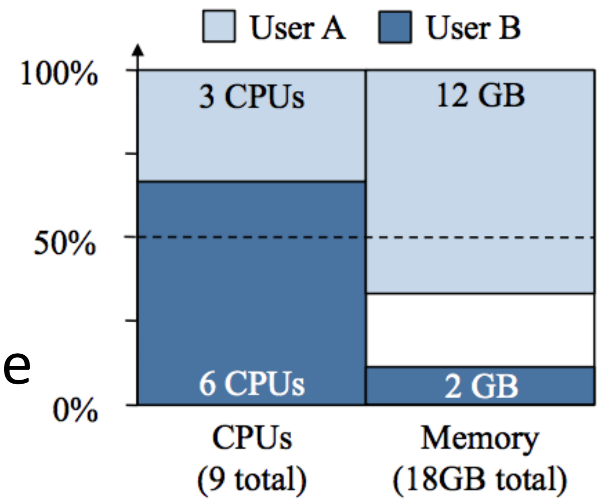
- Dominant resource of user 1: 1/9, 4/18 -> memory
- Dominant resource of user 2: 3/9, 1/18 -> CPU

No concept of allocation yet

amazon

Dominant Resource Fairness

- Two users:
  - User A $(w_{11}, w_{12}) = (1,4)$
  - User B $(w_{21}, w_{22}) = (3,1)$



- Change variables to dominant resource
  - $y1 = 4/18*x1$
  - $y2 = 3/9*x2$

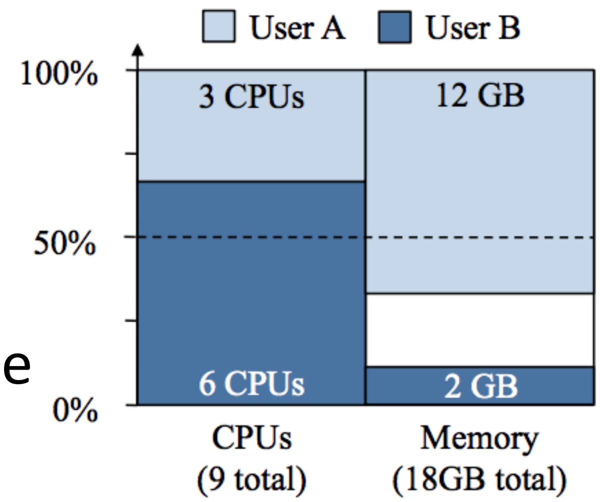
- **Quiz: solve for max-min fairness on Y**

amazon

Dominant Resource Fairness

- Two users:
  - User A $(w_{11}, w_{12}) = (1,4)$
  - User B $(w_{21}, w_{22}) = (3,1)$

- Change variables to dominant resource
  - y1 = 4/18*x1
  - y2 = 3/9*x2



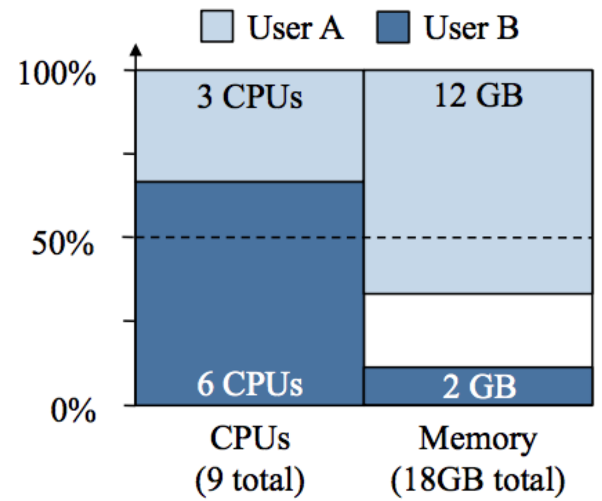
$$\text{for CPU}, \quad x_1 + 3x_2 \le 9$$
$$\text{for Memory}, \quad 4x_1 + x_2 \le 18$$

→

$$\text{for CPU}, \quad y_1 + 2y_2 \le 2$$
$$\text{for Memory}, \quad 6y_1 + y_2 \le 6$$

Dominant Resource Fairness

- Two users:
  - User A $(w_{11}, w_{12}) = (1,4)$
  - User B $(w_{21}, w_{22}) = (3,1)$



- Solve:

$$\max_{y \geq 0} \frac{y_1^{1-\alpha}}{1-\alpha} + \frac{y_2^{1-\alpha}}{1-\alpha}$$
$$\text{s.t. } y_1 + 2y_2 \leq 2$$
$$6y_1 + y_2 \leq 6$$

→ y1=y2 → y1=y2=2/3 → x1=18/4*2/3 = 3, x2=9/3*2/3 = 6

User A: 3CPU, 12GB
User B: 6CPU, 2GB

amazon

Questions?

paschosg@amazon.com

- For questions about the course
- For questions about internship opportunities

https://paschos.net/

- Course material & relevant papers

amazon