Amortized Graphics

Adrien Treuille
Carnegie Mellon University

Syllabus

Data Parallelism

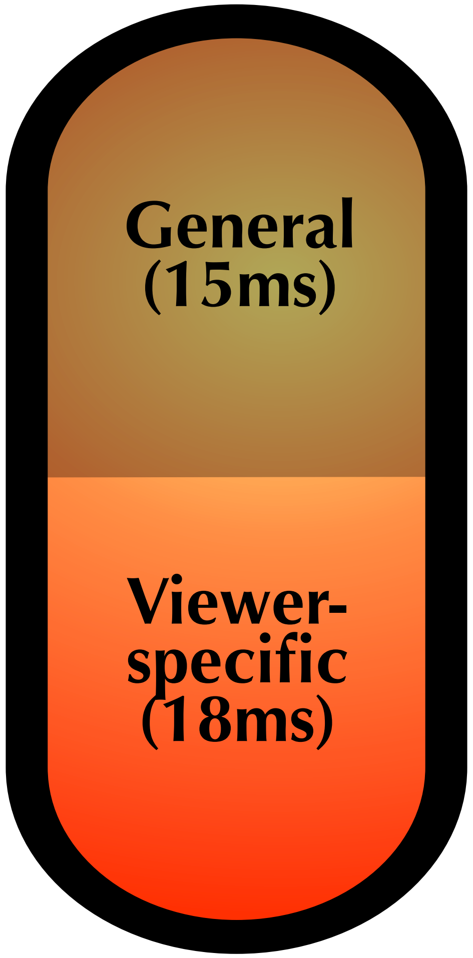
```
T Jan 11, 2011 - 01 - Overview
R Jan 13, 2011 - 02 - Data Parallelism
T Jan 18, 2011 -    - NO CLASS
R Jan 20, 2011 -    - NO CLASS
T Jan 25, 2011 - 03 - Graphics Architectures       - P1 Fractals
R Jan 27, 2011 - 04 - Linear Systems               - P1 Fractals
T Feb  1, 2011 - 05 - GPU Caching                  - P1 DUE
R Feb  3, 2011 - 06 - Poisson Blending             - P2 Blending
T Feb  8, 2011 - 07 - Graph Cuts                   - P2 Blending
R Feb 10, 2011 - 08 - Graph Cut Image Analysis     - P2 DUE (P3 Assigned)
T Feb 15, 2011 - 09 - Prefix Sums - Sorting / BFS  - P3 Graph Cuts
R Feb 17, 2011 - 10 - Fun GPU Tricks               - P3 Graph Cuts
T Feb 22, 2011 - 11 - Fun GPU Tricks               - P3 DUE
R Feb 24, 2011 - 12 - Fun GPU Tricks
T Mar  1, 2011 - 13 - P4 Presentations
R Mar  3, 2011 - 14 - P4 Presentations
```

Task Parallelism

```
T Mar  8, 2011 -    - SPRING BREAK
R Mar 10, 2011 -    - SPRING BREAK
T Mar 15, 2011 - 15 - Amortized Graphics
R Mar 17, 2011 - 16 - Light Fields                 - P5 Light Fields
T Mar 22, 2011 - 17 - Amortization Architecture    - P5 Light Fields
R Mar 24, 2011 - 18 - Queuing Theory               - P5 DUE
T Mar 29, 2011 - 19 - Precomputed Radiance Transfer - P6 Amortized Light Fields
R Mar 31, 2011 - 20 - Amortized Physics            - P6 Amortized Light Fields
T Apr  5, 2011 - 21 - Amortization Tricks          - P6 Due
R Apr  7, 2011 - 22 - Amortization Tricks          -
T Apr 12, 2011 - 23 - Amortization Tricks          -
R Apr 14, 2011 -    - NO CLASS                     -
T Apr 19, 2011 - 24 - P7 Presentations             -
R Apr 21, 2011 - 25 - P7 Presentations             -
T Apr 26, 2011 - 26 - P7 Presentations             -
R Apr 28, 2011 - 27 - P7 Presentations             -
```

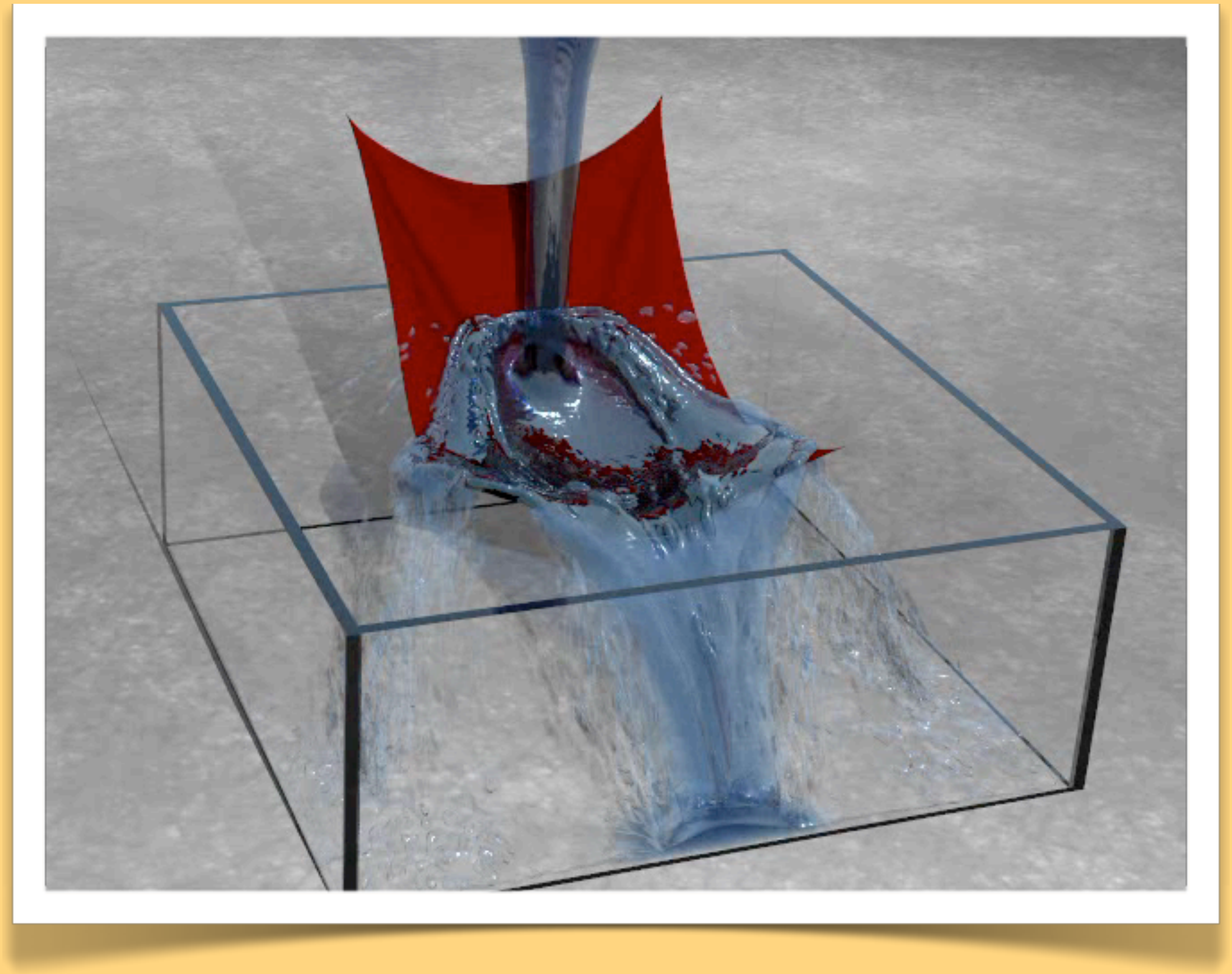
Single User Model

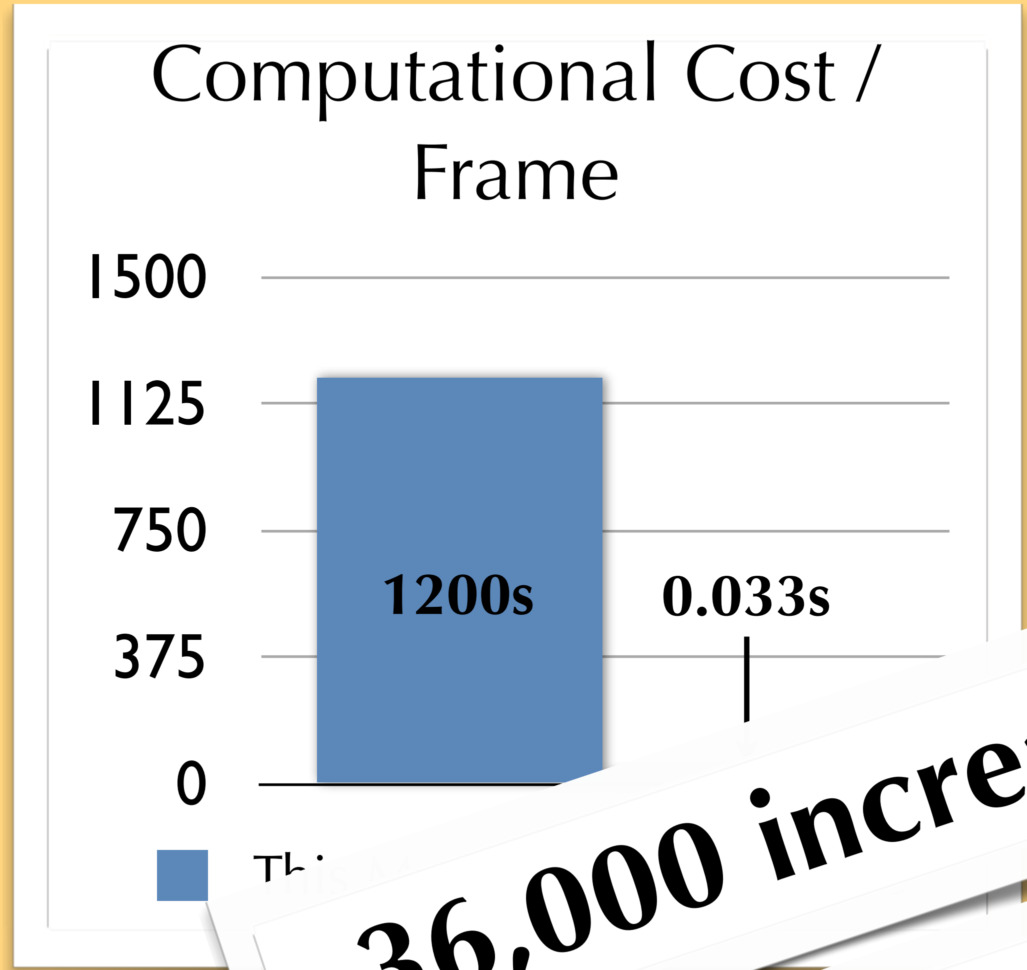
User ↔ Monitor ↔ Computer



1 unit of computation

Challenge: Computation



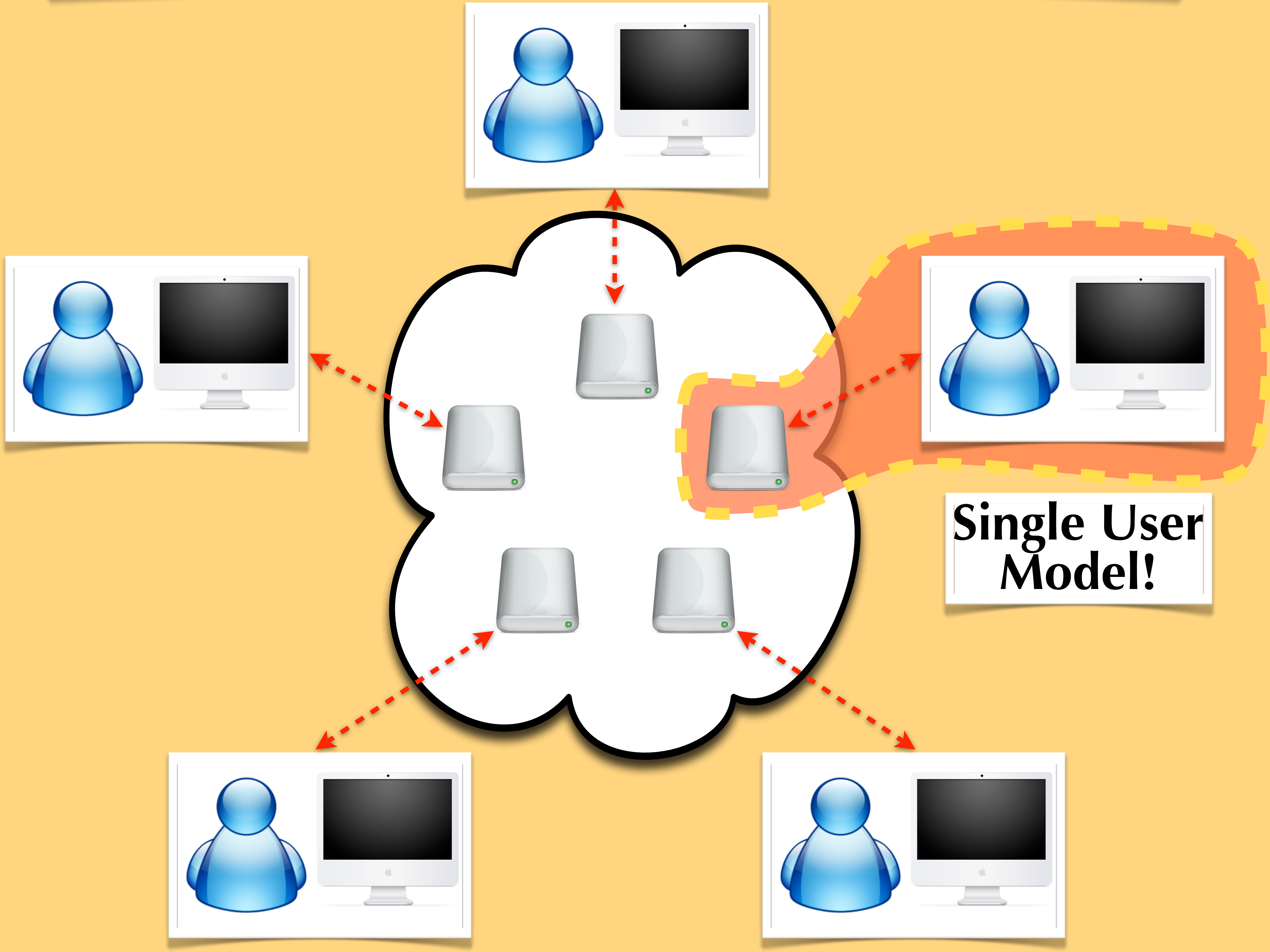


>36,000 increase

>15 doublings

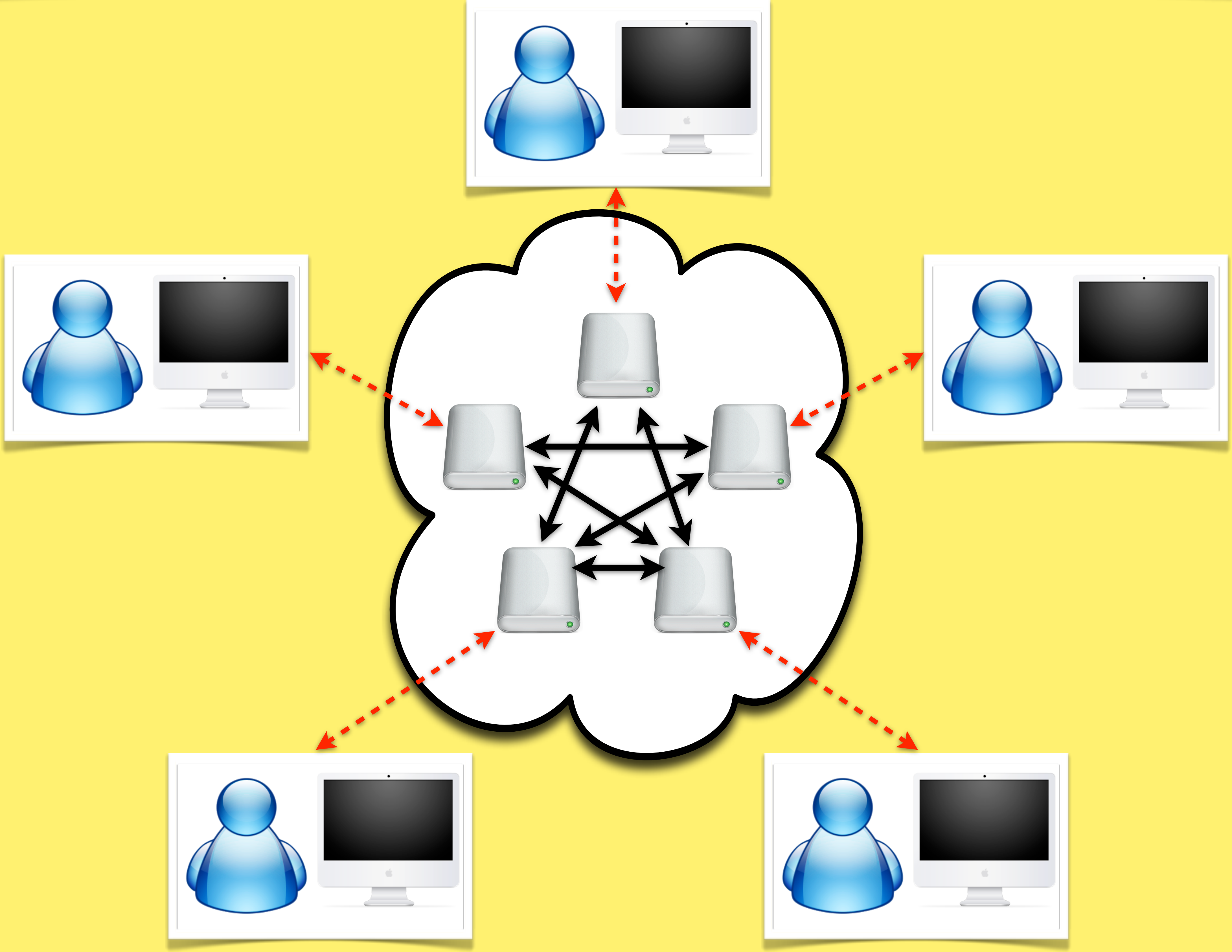
Parallelism in the Cloud

OnLive Architecture



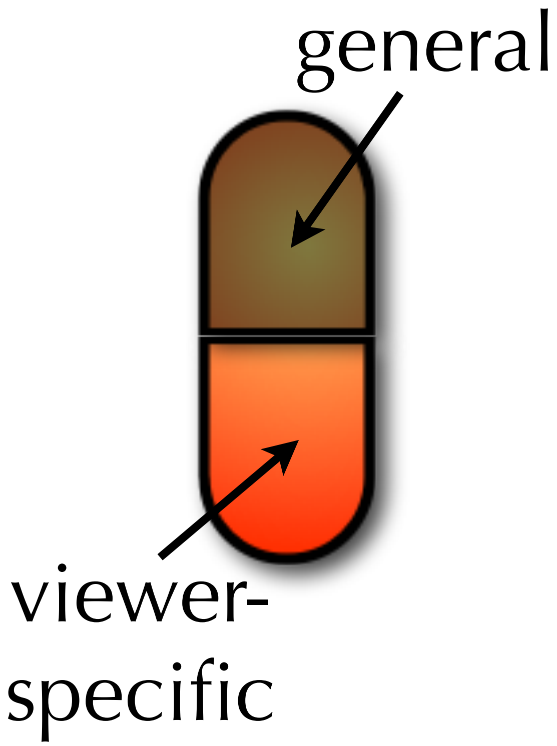
Single User Model!

Parallelism in the Cloud



Parallelism in the Cloud

Unit of Time

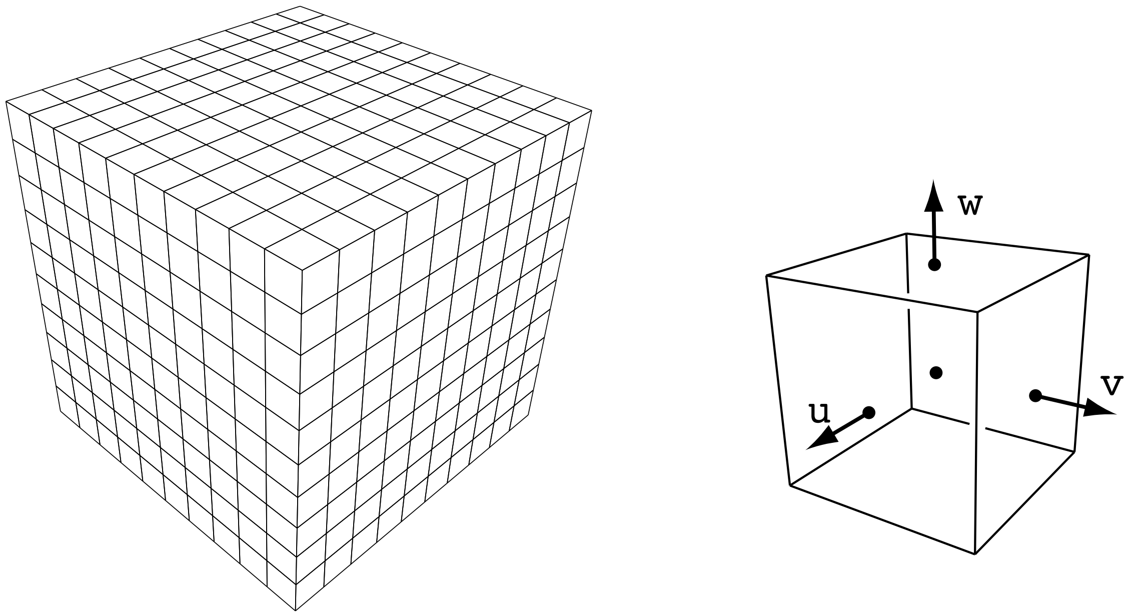


General Pool

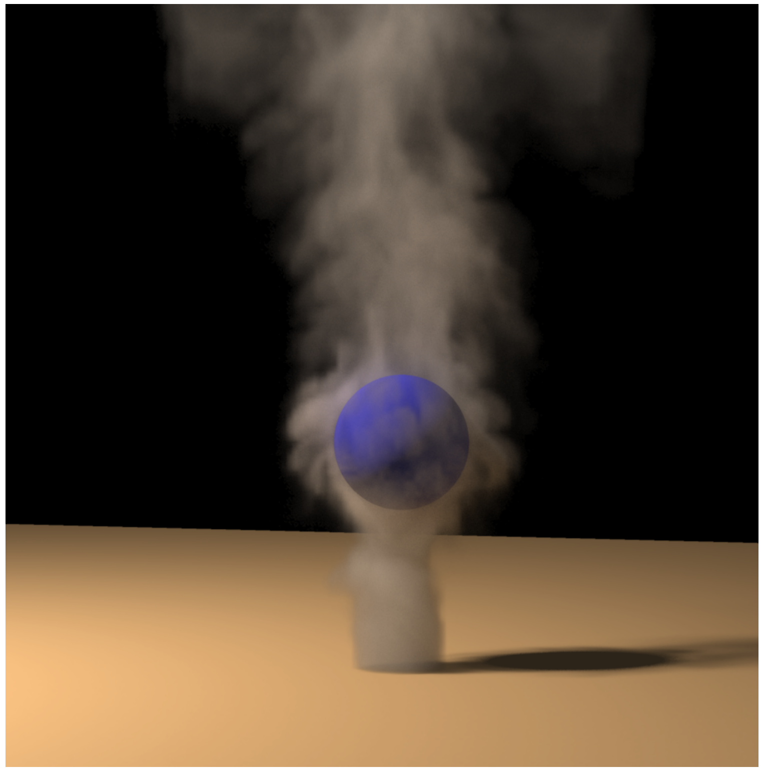
Some questions?

- How can we measure the "amortizability" of an algorithm?
- What kinds of amortizability are there?
- What algorithms could we well amortized?

Time Amortization



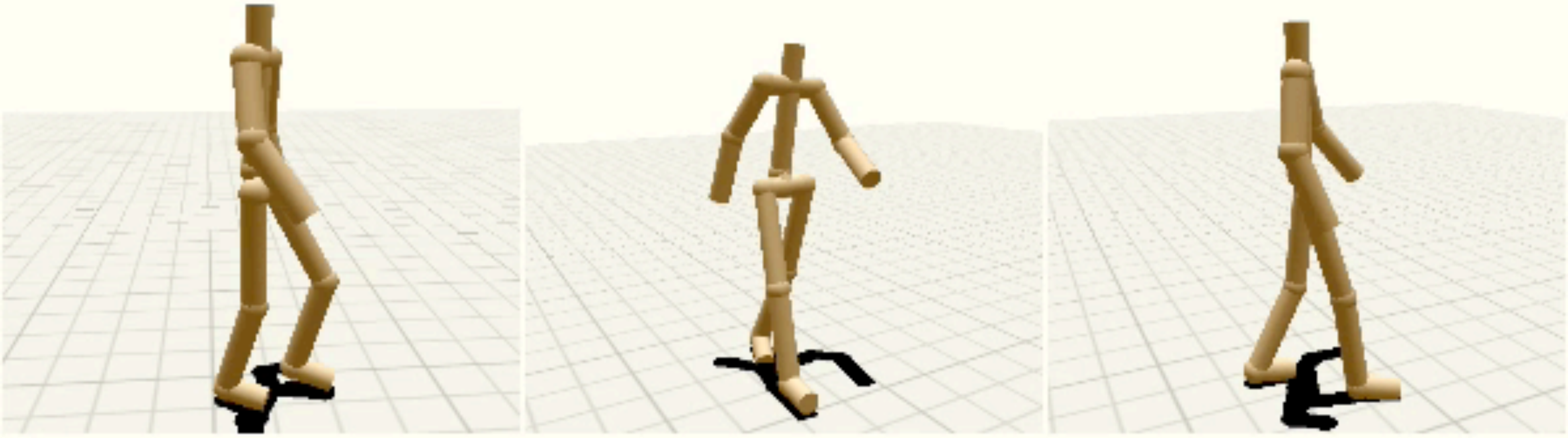
Simulation



Rendering

Space Amortization

Space Amortization



Types of Amortization

- Time amortization
- Space amortization
- Timestep constraints?
- Extent constraints?

Questions

- What algorithms are good to amortize?
- How could we actually set up computers to do this?
- What dangers / bottlenecks do you expect?