

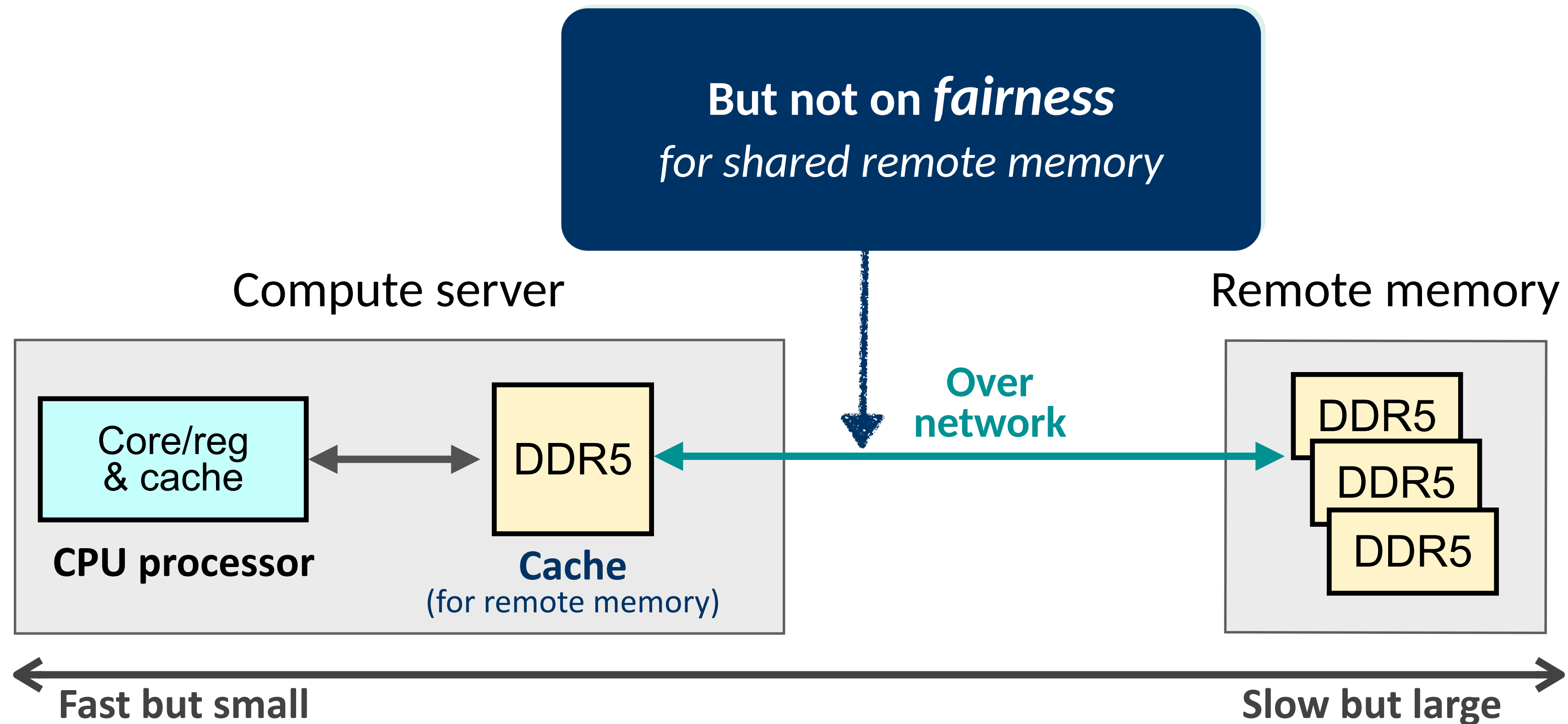
Spirit: Fair Allocation of Interdependent Resources in Remote Memory Systems

Seung-seob Lee, Jachym Putta, Ziming Mao*, Anurag Khandelwal

*Yale University, *UC Berkeley*

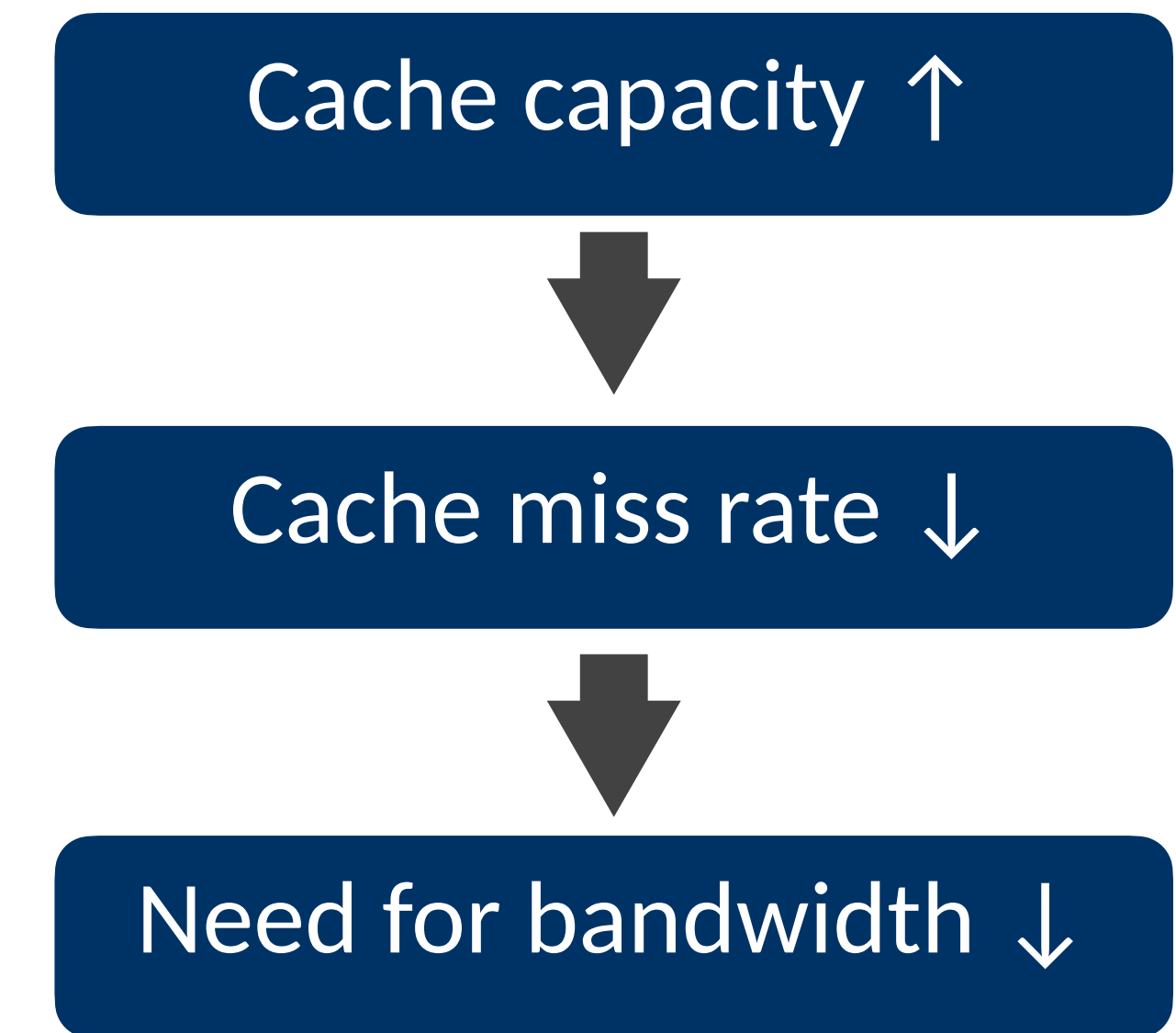
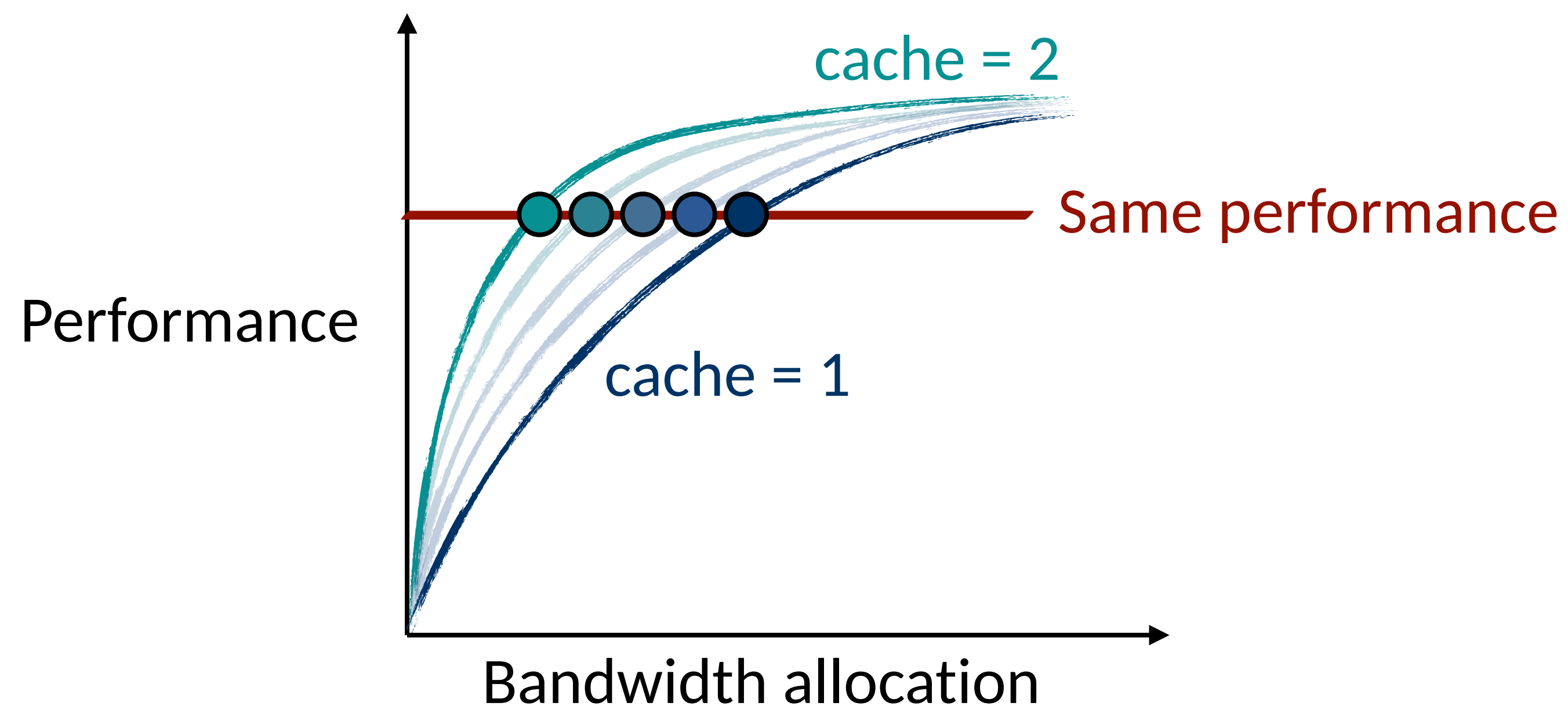
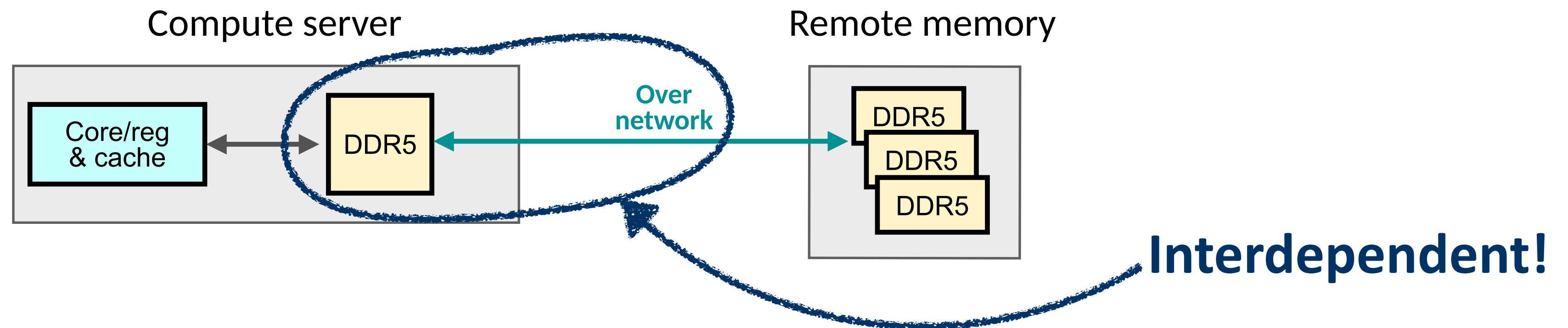


Remote Memory System

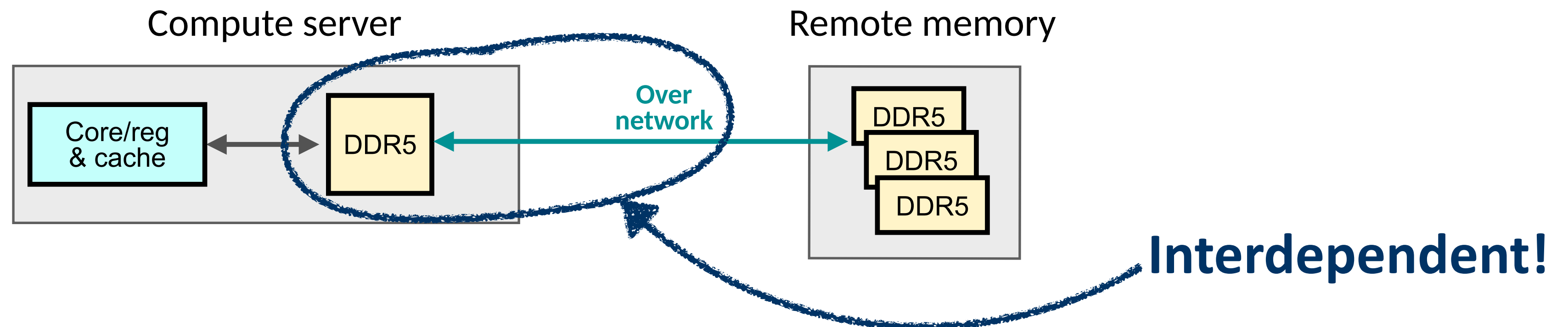


Remote memory enables higher resource utilization
by decoupling *memory* from *CPU allocation*

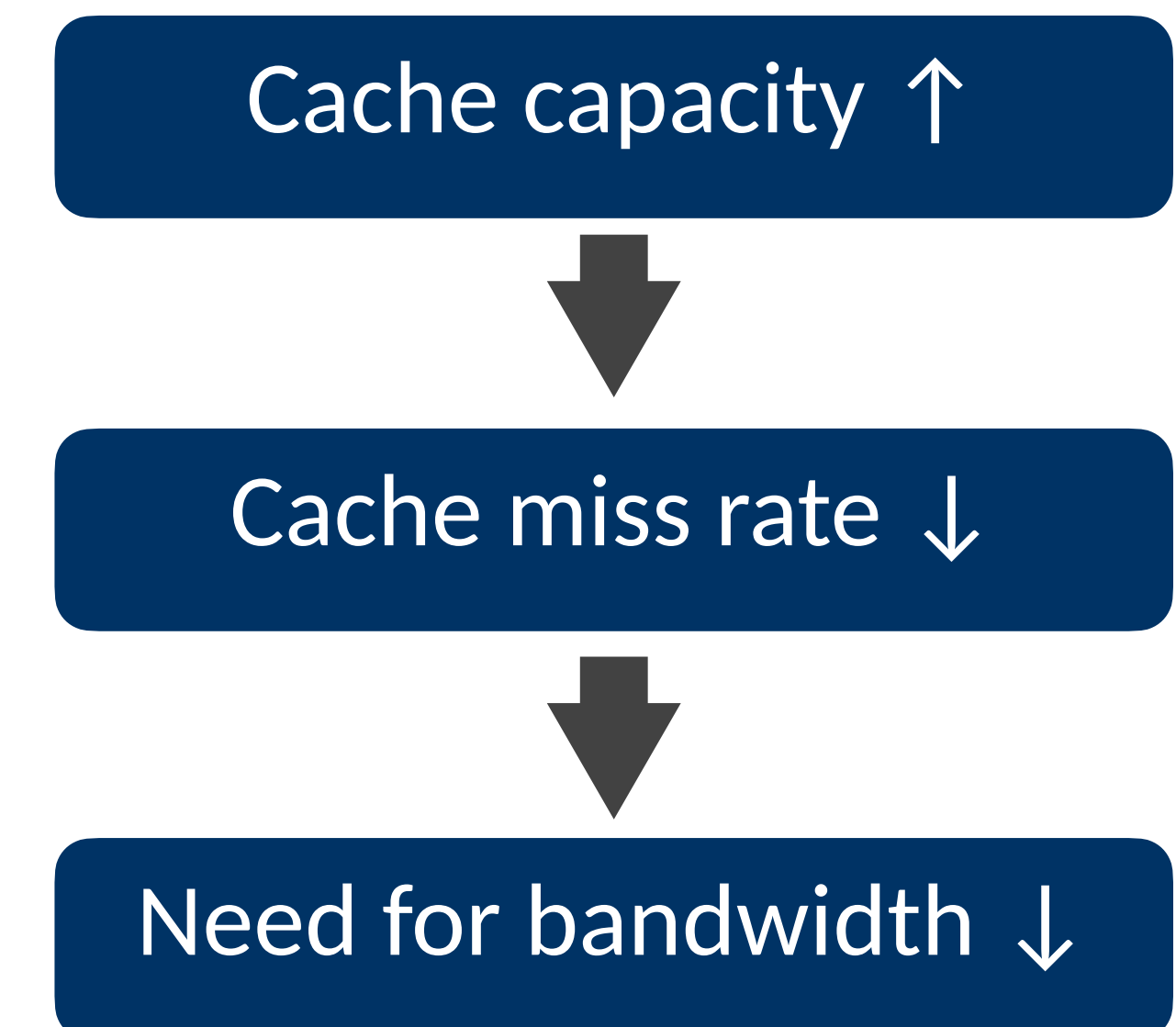
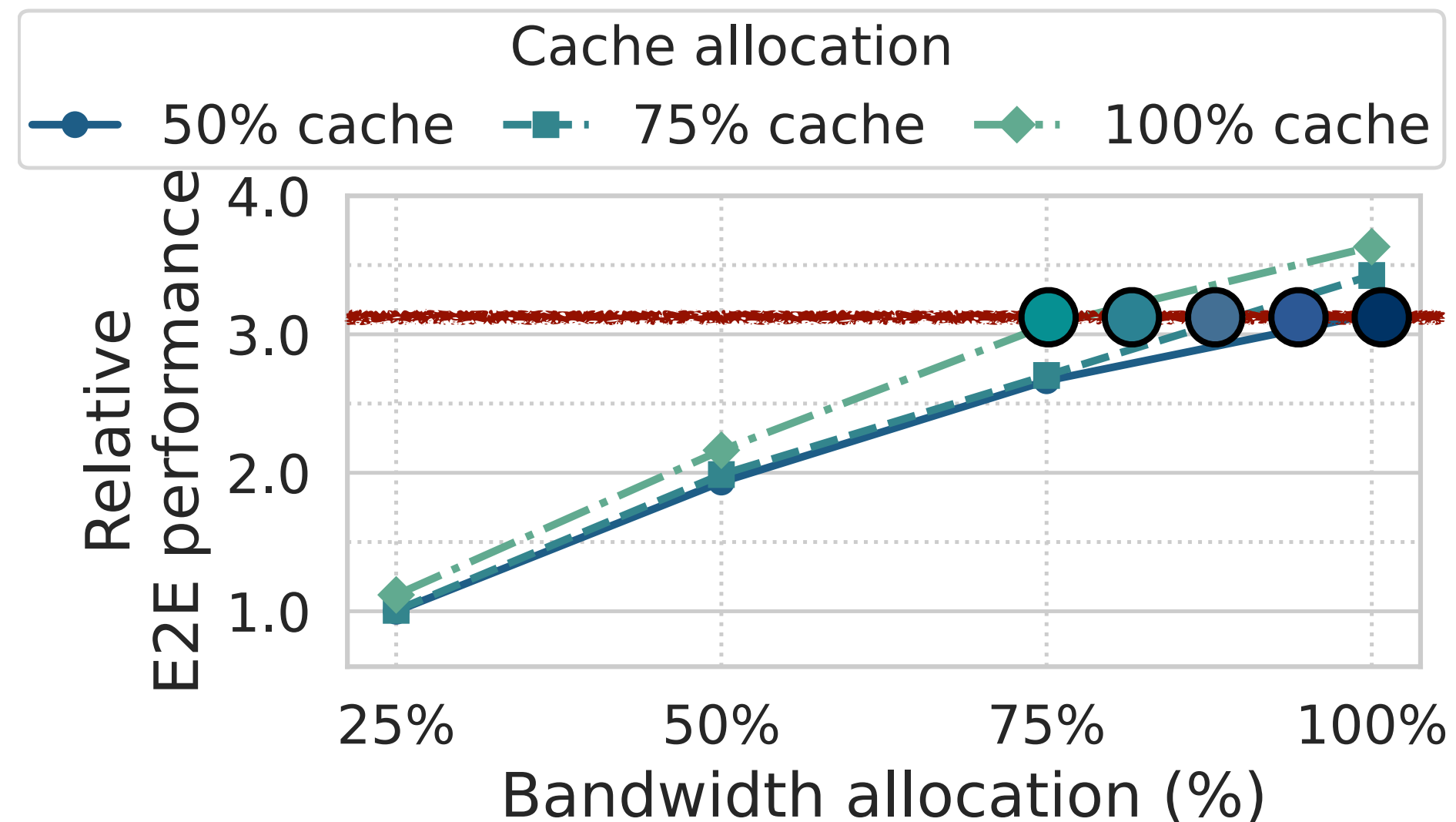
Cache and Bandwidth are Interdependent



Cache and Bandwidth are Interdependent



StreamBench



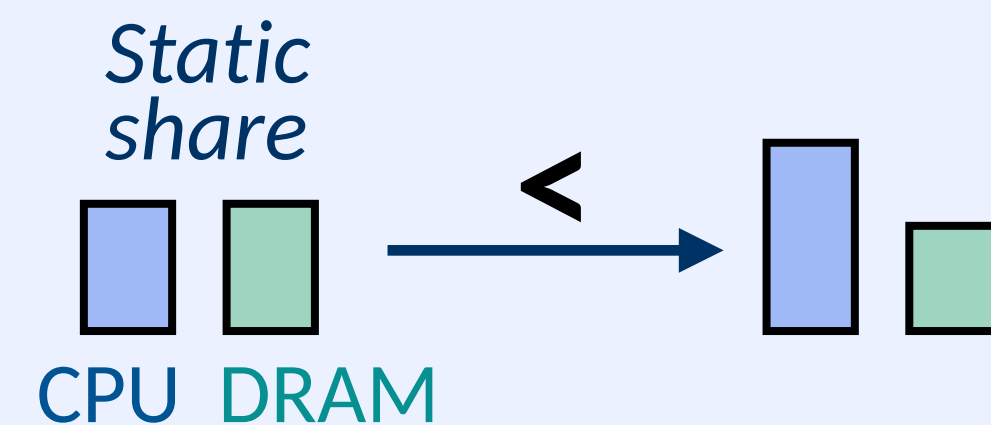
Prior Approaches for Fair Multi-resource Allocation

- Independent resources—E.g., DRF (NSDI'11)

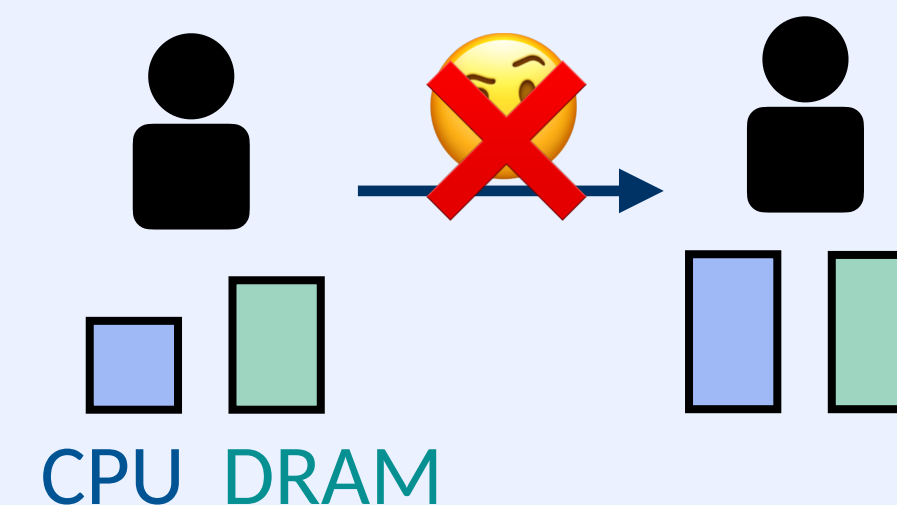


4 vCPU and 16 GB of DRAM
per Memcached container

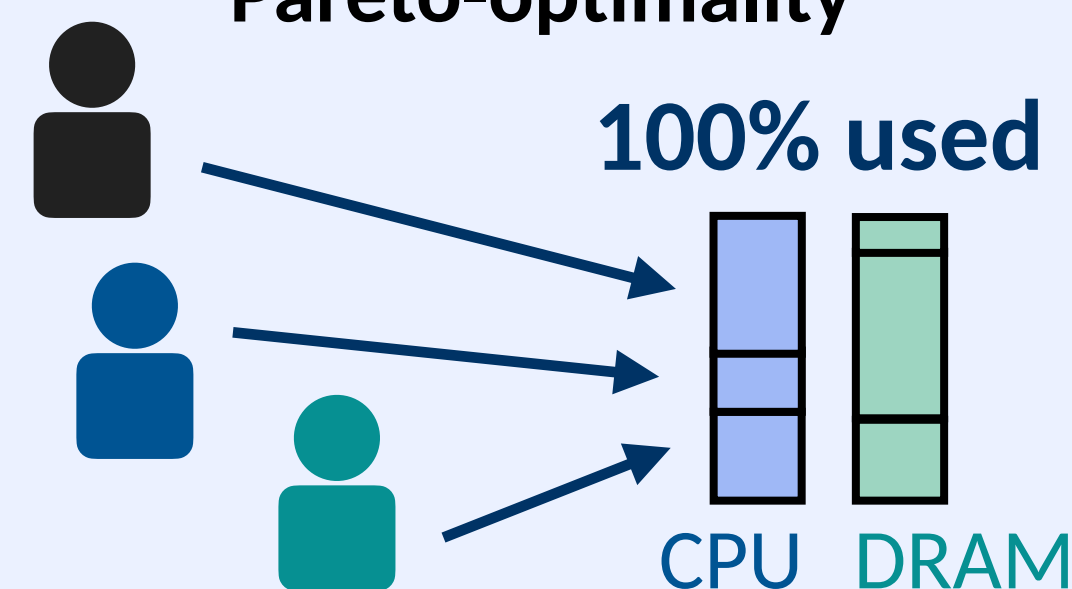
Sharing incentives



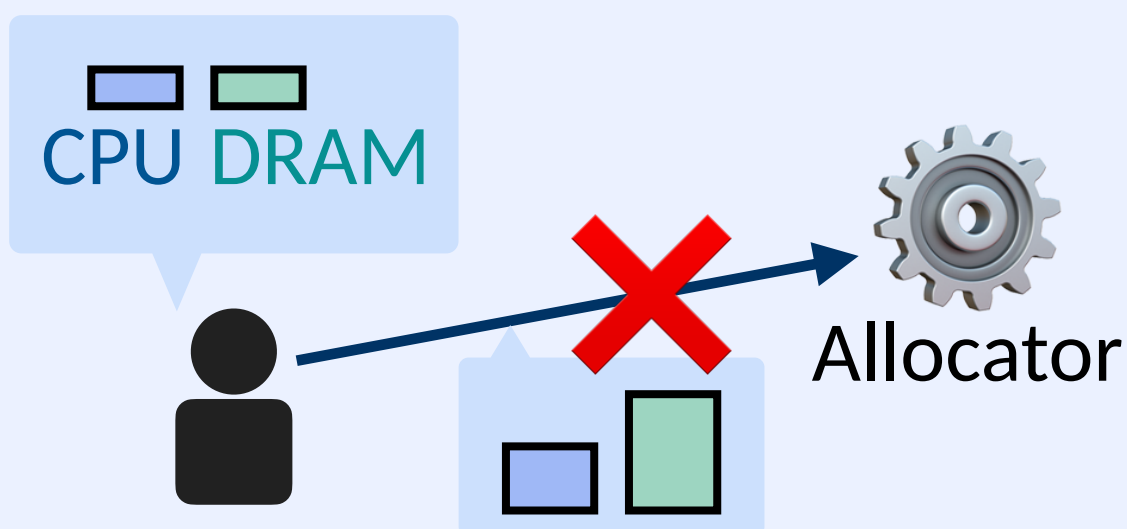
Envy-freeness



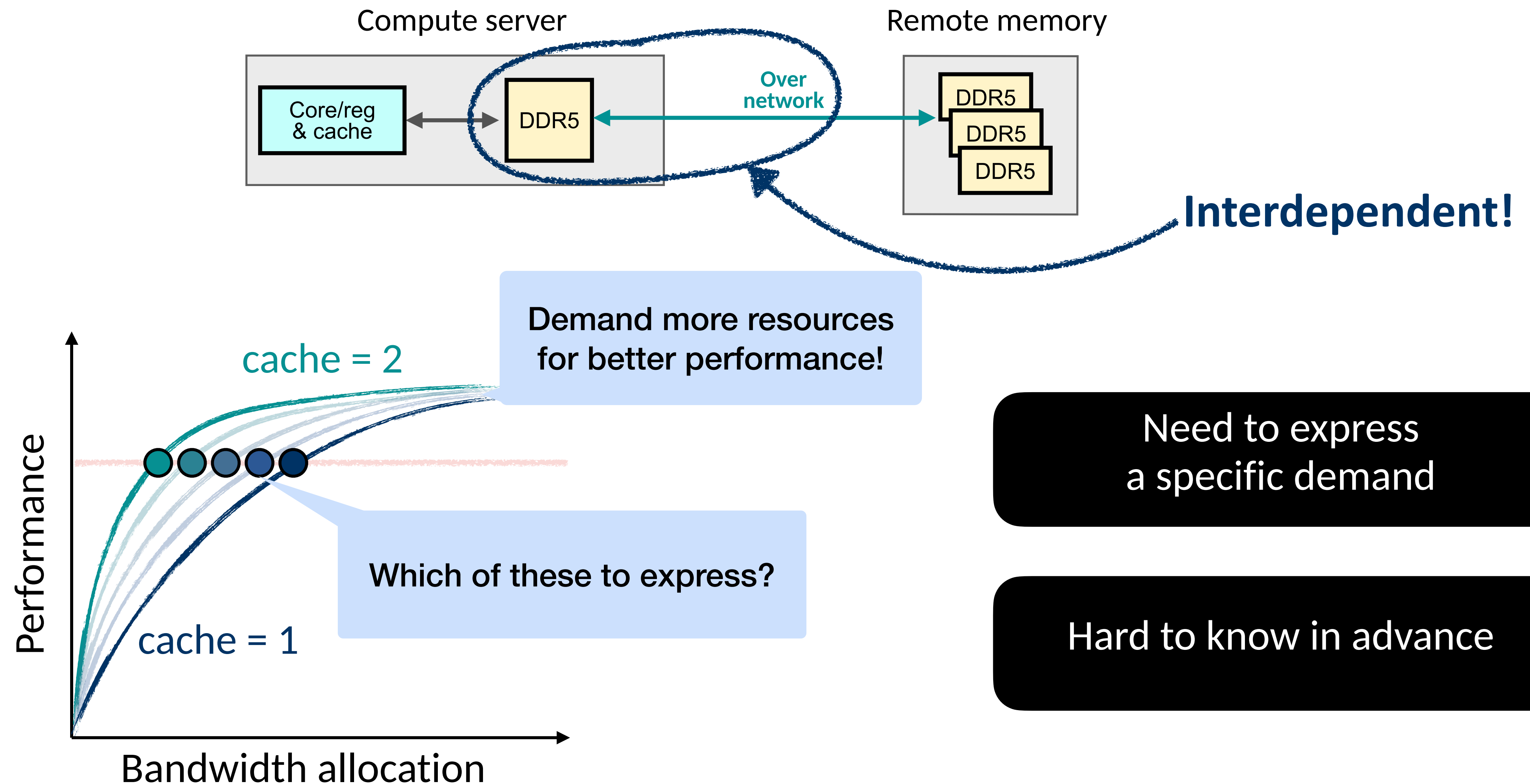
Pareto-optimality



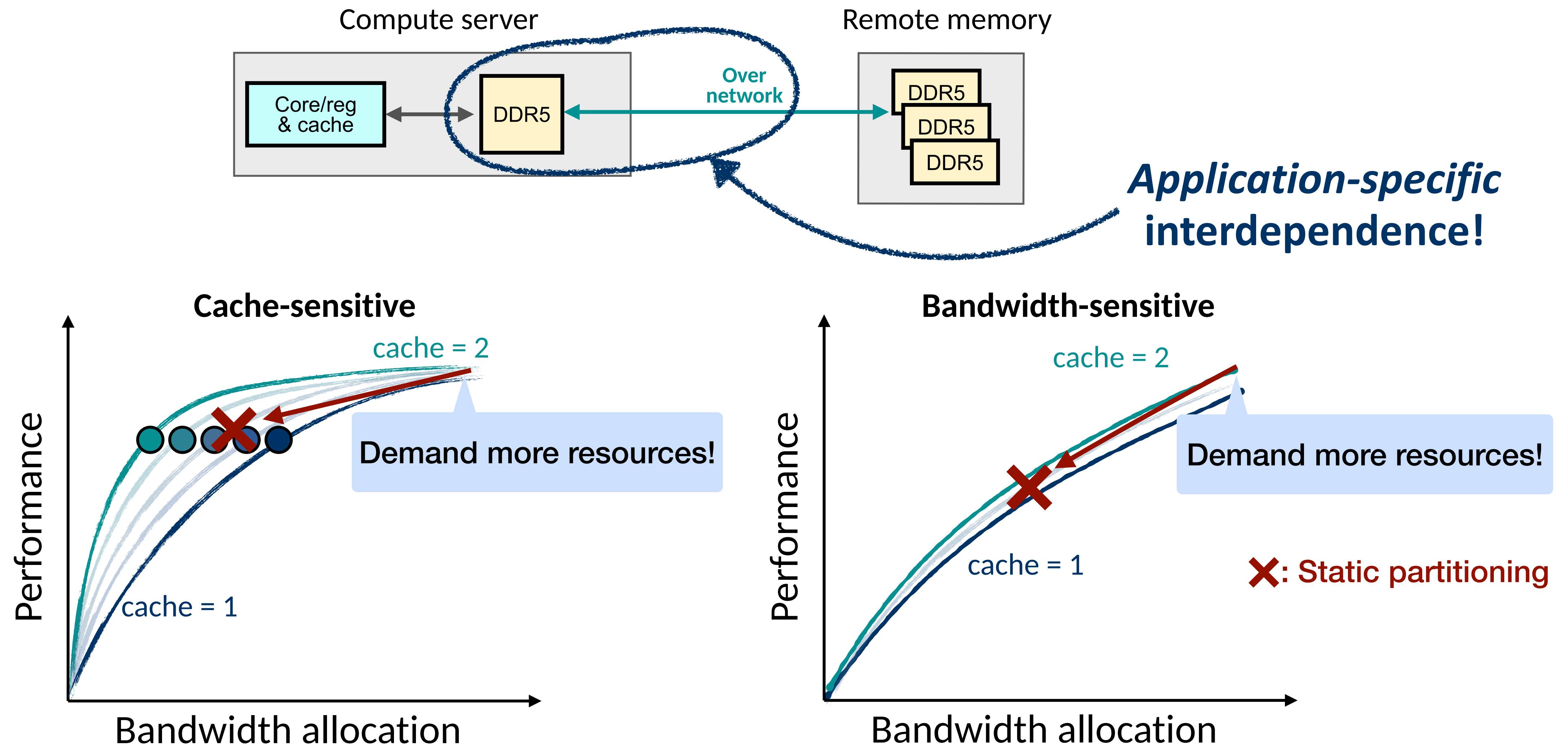
Strategy-proofness



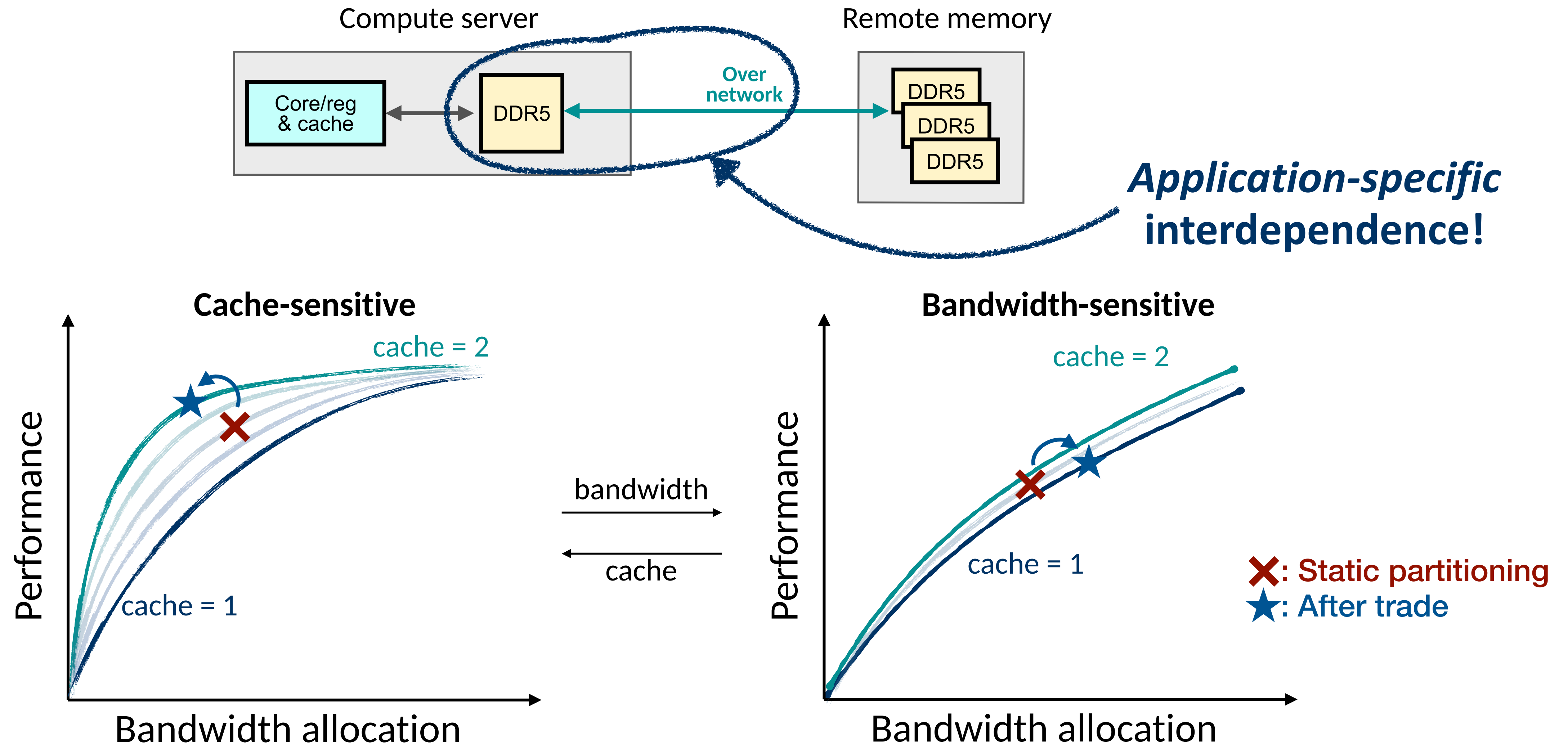
Prior Schemes are Insufficient



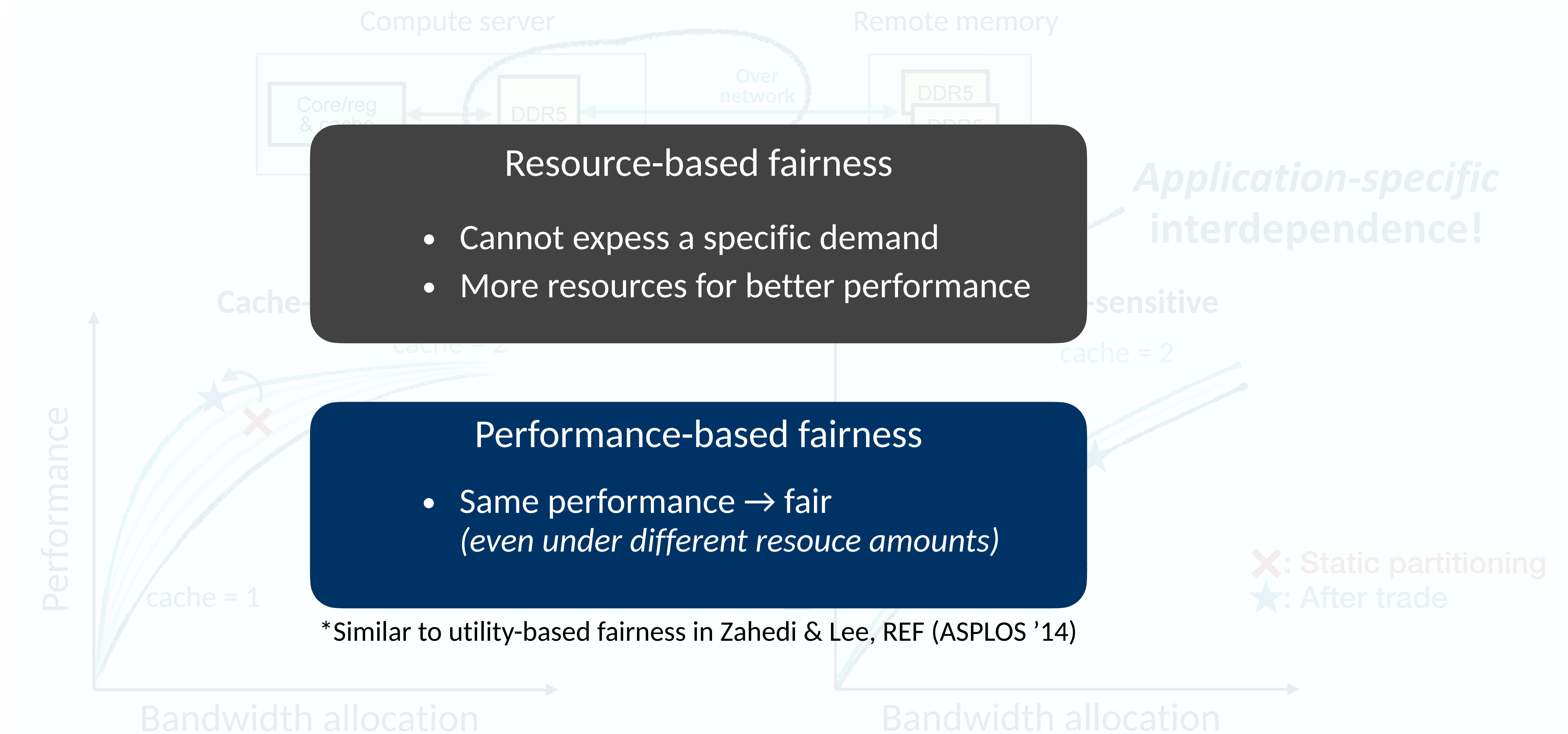
Prior Schemes are Insufficient



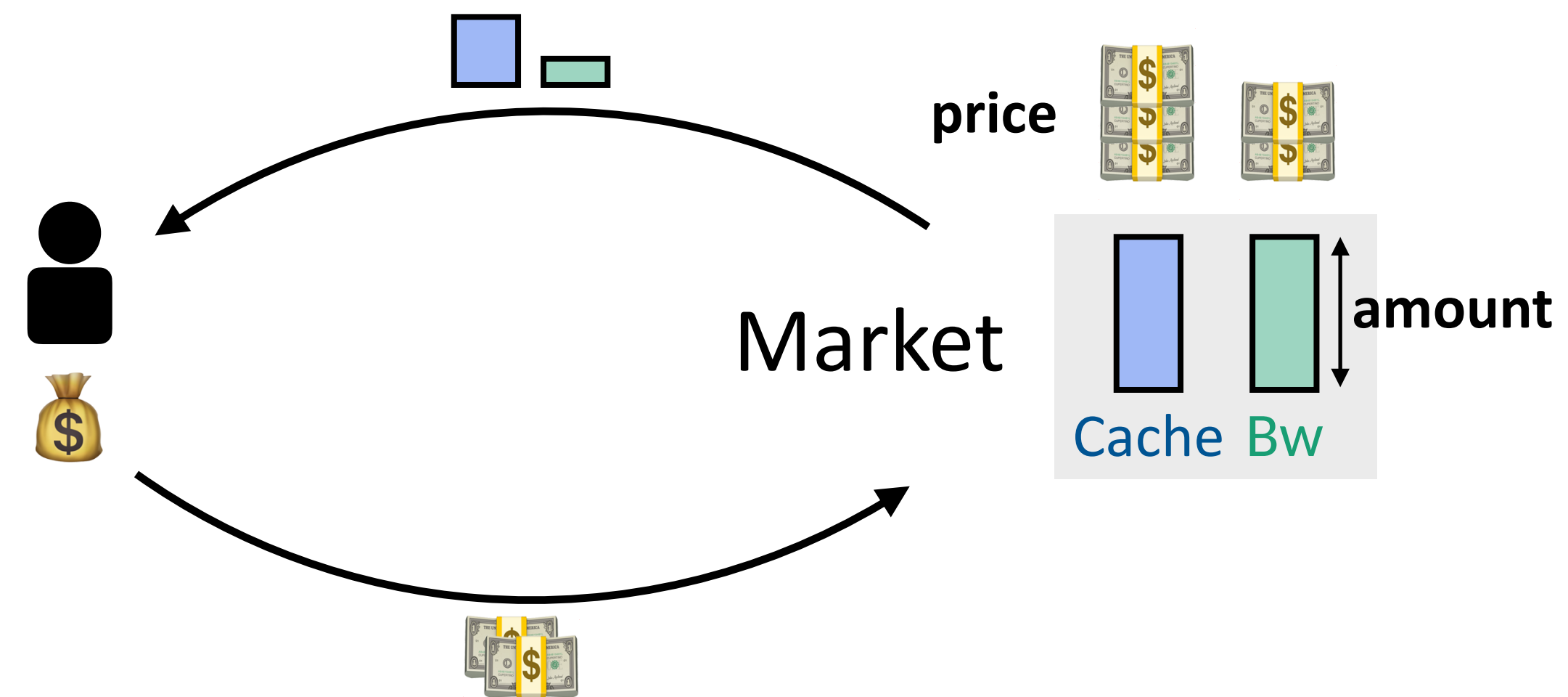
Key Ideas: Trading for Mutual Benefit



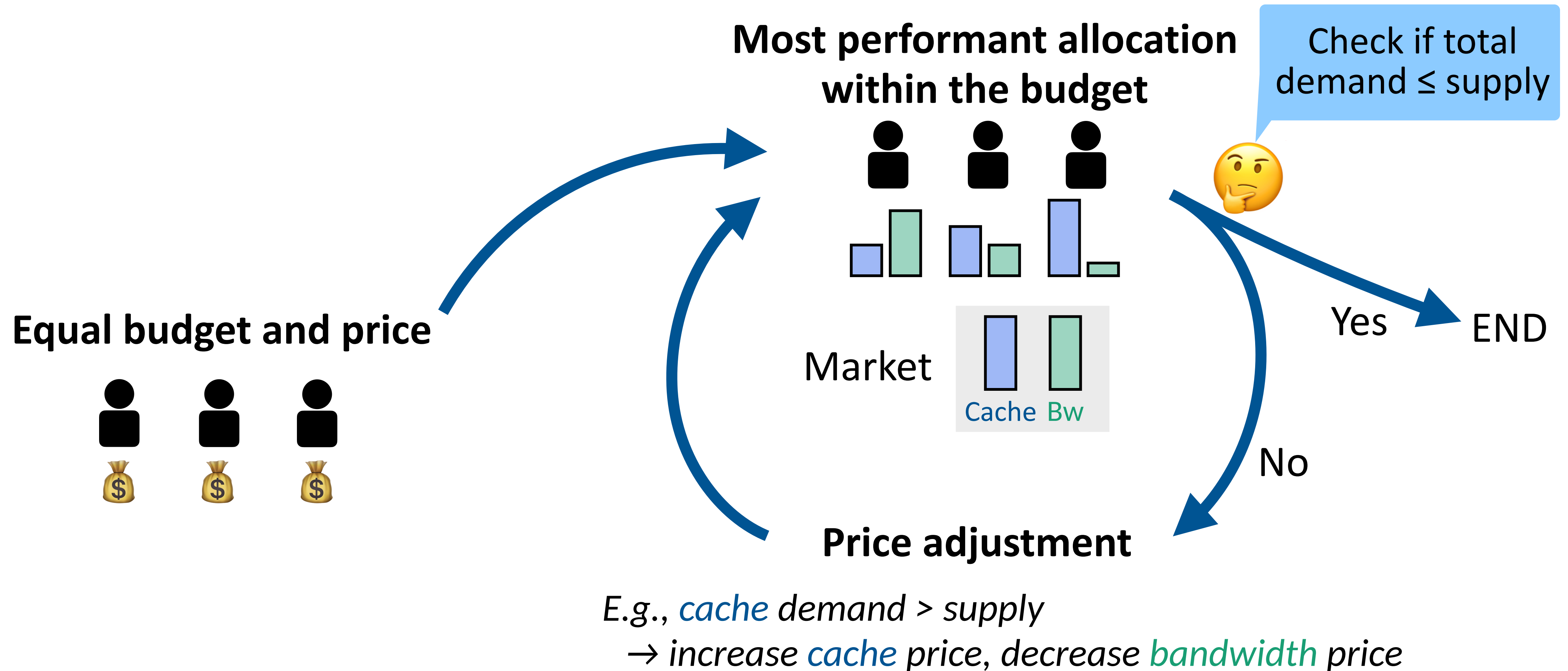
Key Ideas: Trading for Mutual Benefit



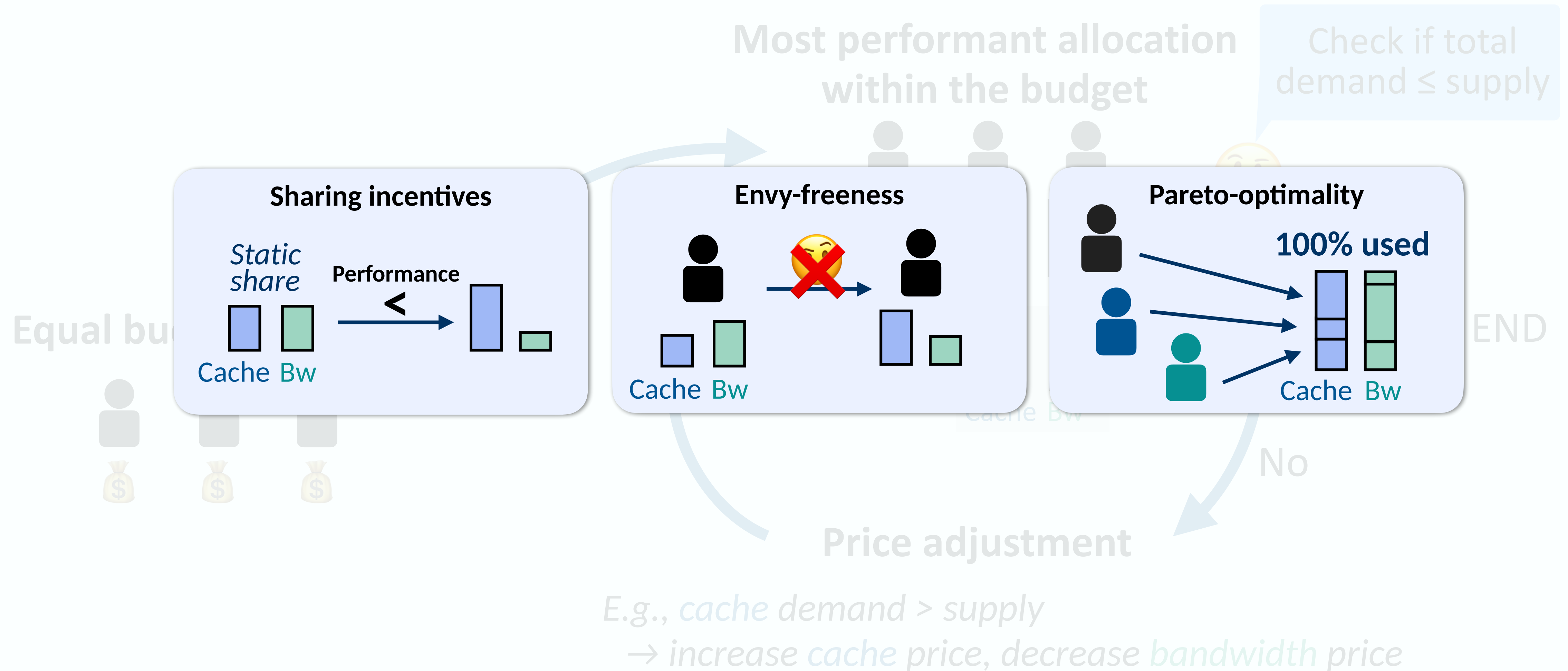
Symbiosis Algorithm



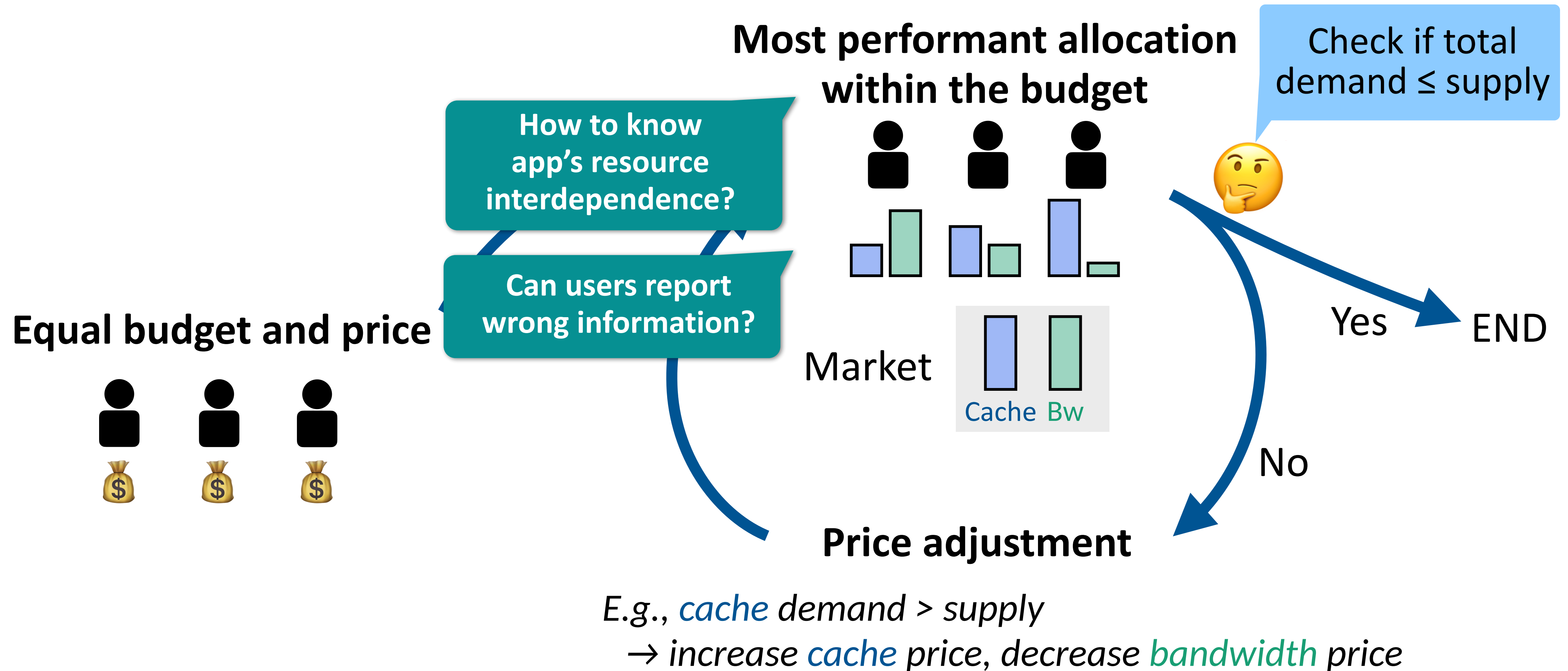
Symbiosis Algorithm



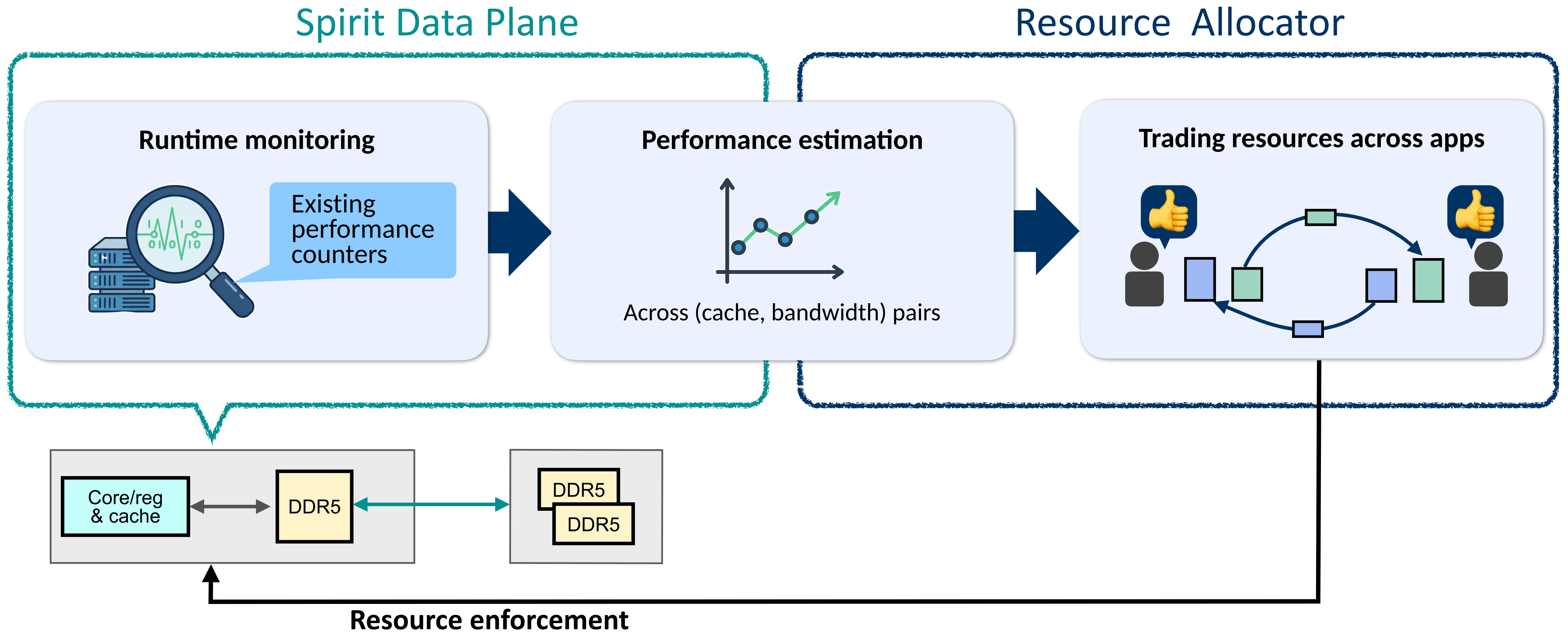
Symbiosis Algorithm



Symbiosis Algorithm



Spirit Design



Spirit Design

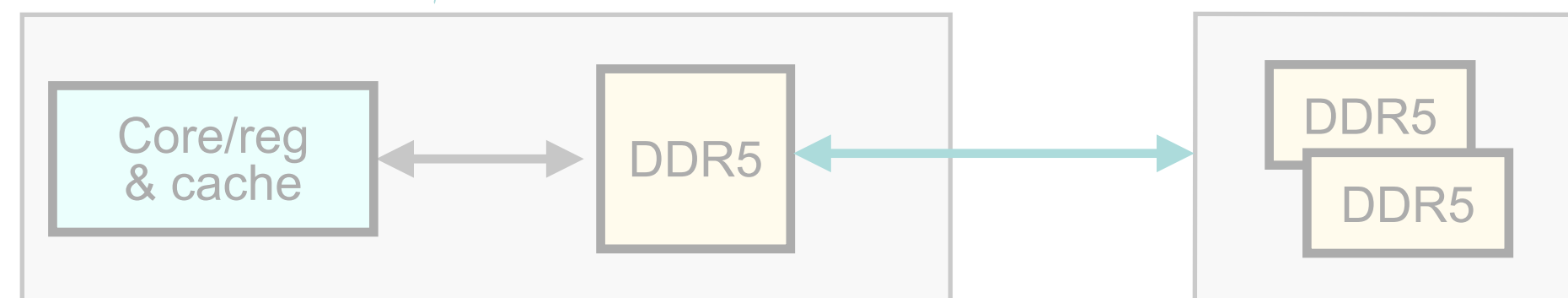
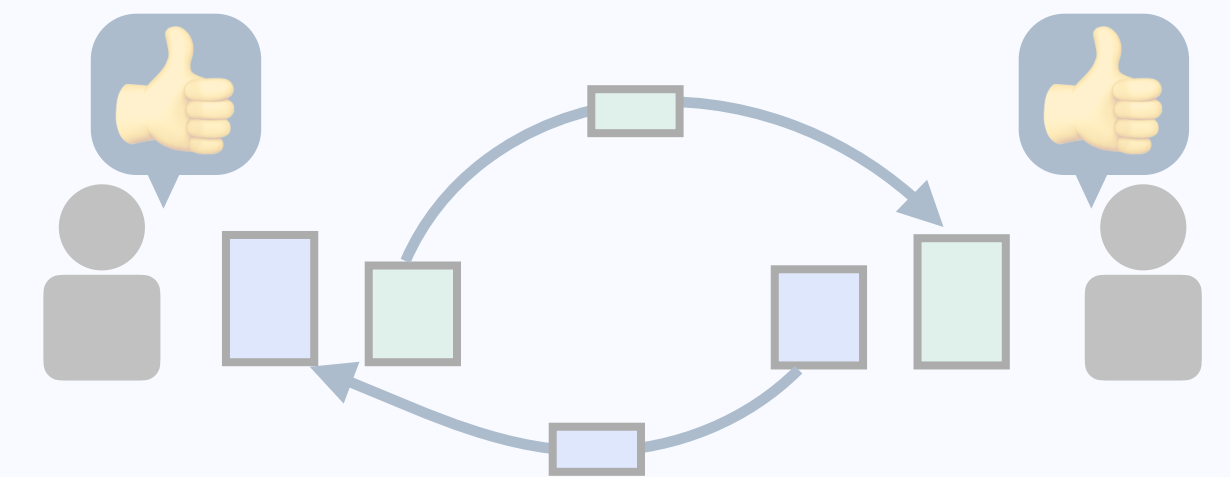
Spirit Data Plane

- Runtime noise & dynamics
- Instability of measured performance
- Resource enforcement w/o OS modifications
- Estimation overhead & more

→ see our paper for details!

Resource Allocator

Trading resources across apps



Resource enforcement

Performance Evaluation

Evaluation setup

Modeling AWS EC2 instance (m5a.8xlarge), Linux v6.13

- 32 vCPU, 128 GB memory, and 7.5 Gbps bandwidth
- 10 to 20 GB as local memory (among 128 GB)

Diverse applications & sensitivity

- **STREAM** : sensitive to cache & bw
- **Memcached** : sensitive to cache
- **SocialNetwork** : sensitive to cache
- **DLRM** (omitted) : compute-intensive

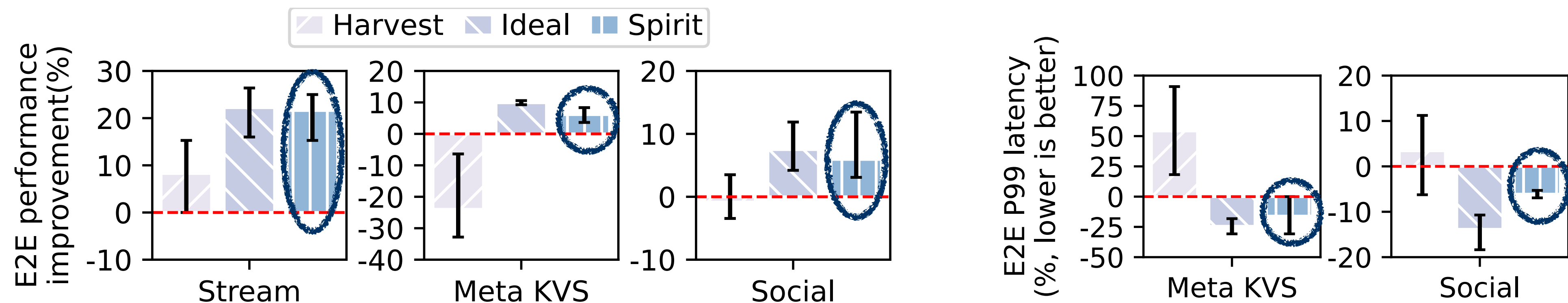
Compared schemes

- **Baseline**: static allocation
(~prior schemes under interdependence)
- **Harvest**: harvest from the most performant and reassign to the least* (No fairness)
- **Ideal**: hand-picked best solution

*Chen et al., PARTIES, ASPLOS '19

Performance Evaluation

Spirit improves throughput and latency, providing fairness (10 GB of \$)



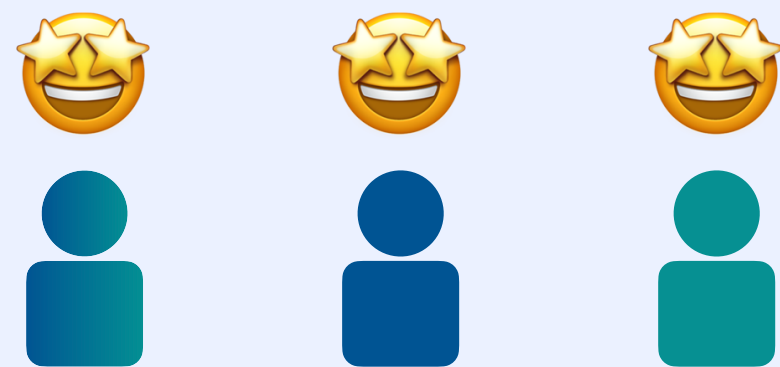
- End-to-end performance improves up to 21.6%, preserving fairness across applications
- Spirit reduces P99 latency by up to 16.8%

Spirit's trading enables mutual benefits for everyone

Conclusion

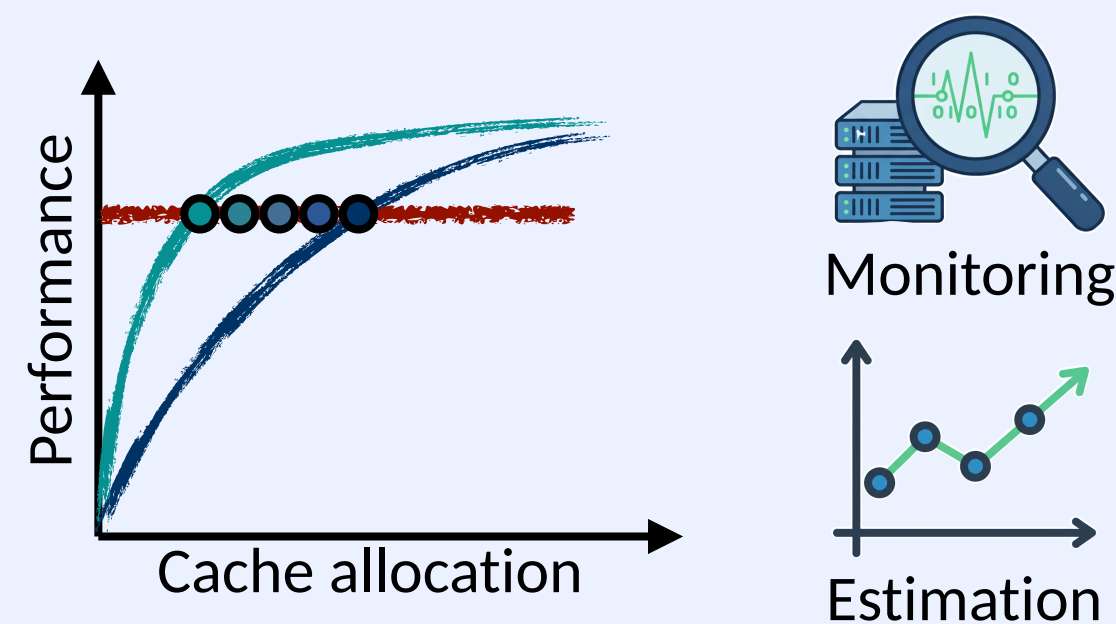
Summary and future directions

Fair resource allocation
for remote memory systems



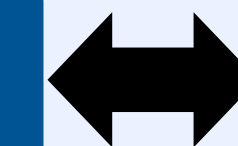
Mutual benefits for everyone

Spirit addresses resource
interdependence



Decoupled design

Allocation



Monitoring &
enforcement

Other resource-interdependency

CXL Compute
Express
Link™

M MEMCACHED

Available at <https://github.com/yale-nova/spirit>