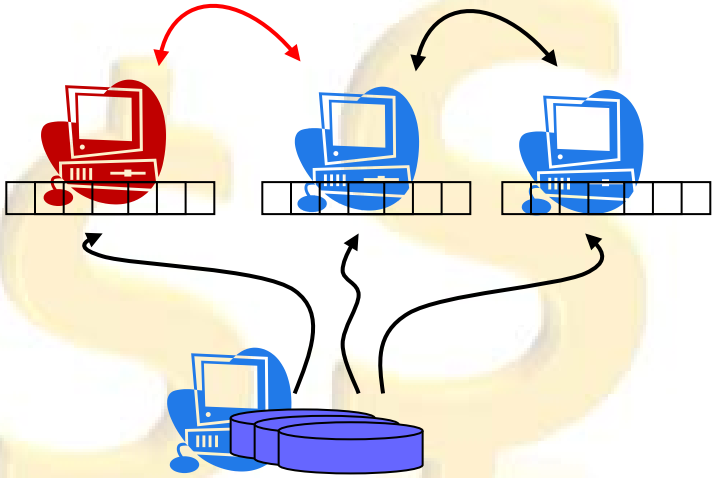
Technion – Israel Institute of Technology

Cooperative Caching with Return on Investment



Gala Yadgar
Technion

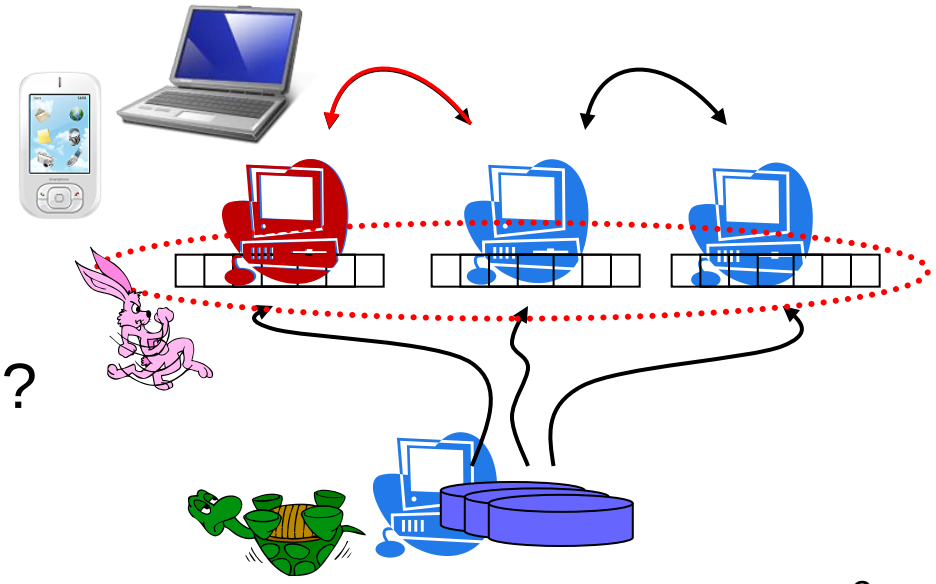
Michael Factor
IBM Research

Assaf Schuster
Technion

Rethinking Cooperative Caching

Traditionally:

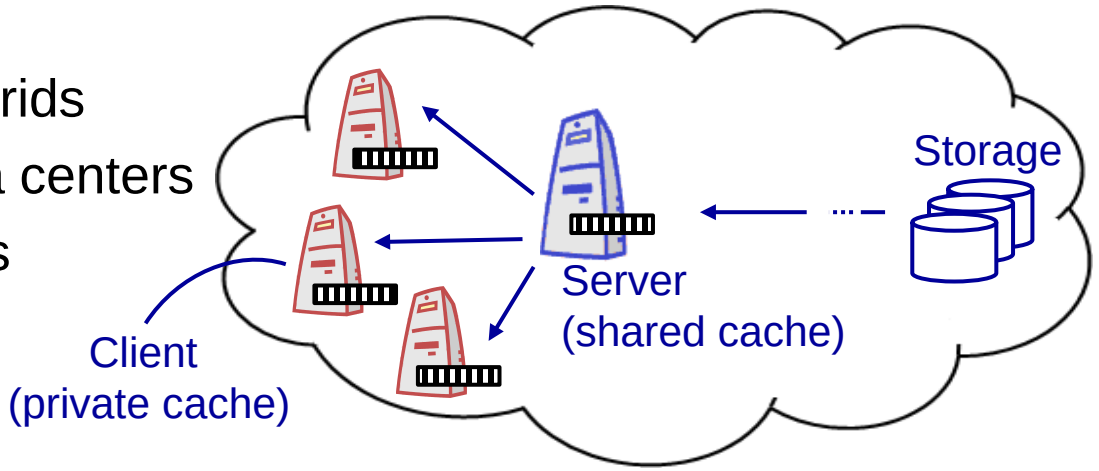
- “Global” LRU management
  - Global optimization
- Central ownership: cooperation is mandatory



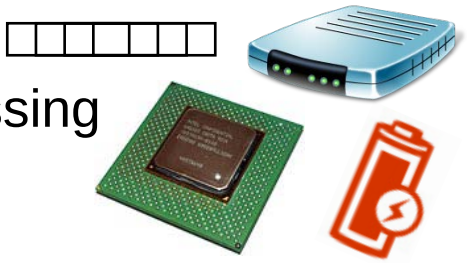
- What about **selfish** clients?

Selfish Clients: Why?

- Large scale resource consolidation
  - Storage clouds
  - Computational grids
  - Large scale data centers
  - Federated CDNs

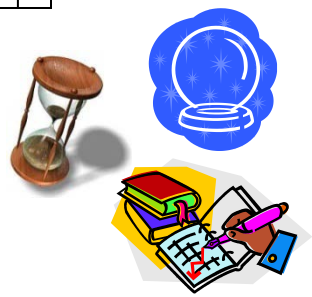


  - Resources owned/chartered by different entities
- Limited resources
  - Buffers, energy, bandwidth, processing



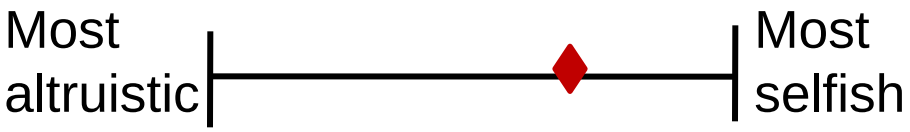
Selfish Clients: How?

- Refuse to cache unneeded blocks 
  - Manage cache according to their own policy
    - Traditionally assumed irrelevant



- Refuse to SERVE blocks
  - Even if they are cached
    - Traditionally assumed cost is negligible

- There is a scale of selfishness

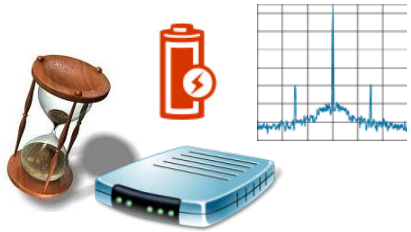


Approaches for Selfish Clients

- Storage caching
  - Limited “private” partition → Insufficient
  - Load balancing → Unfair
- Peer-to-Peer
  - “Tit for tat”, reputations
  - Virtual currency → Short term encounters
- Job scheduling
  - Market based models → Computation and message intensive

A New Model

- New operation: SERVE
  - Added to READ, WRITE etc.
  - Define **cost** in terms of client's objective function



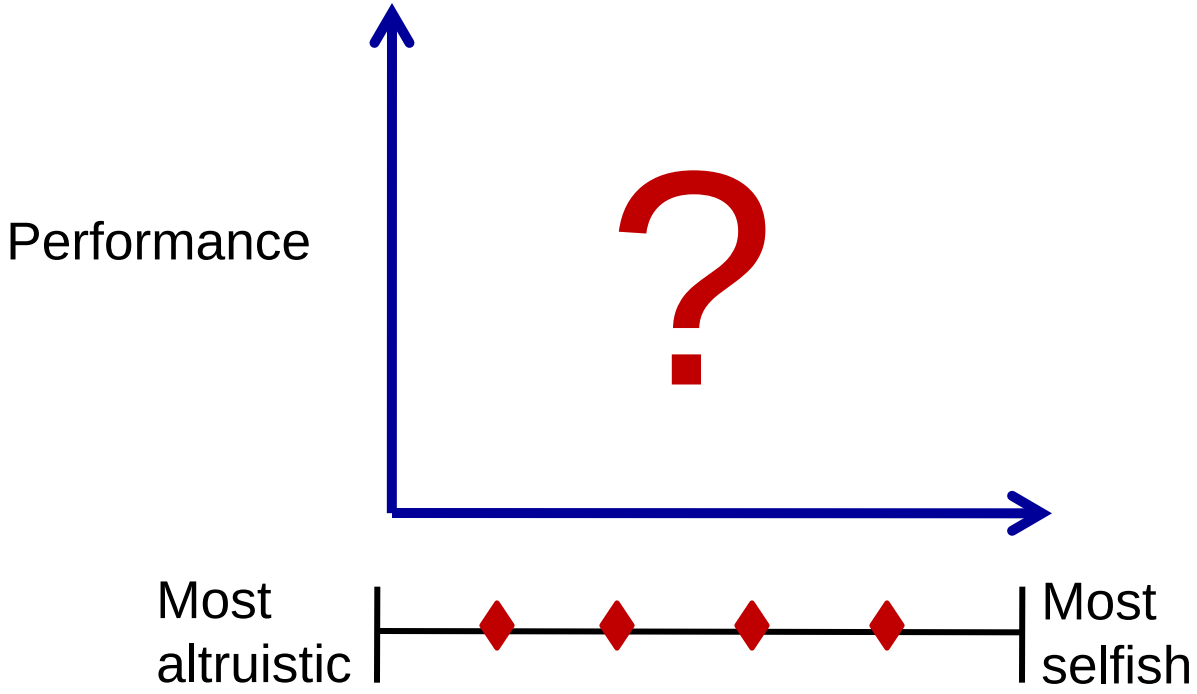
- Measure **utility** derived from cache content
  - Accurate or estimate

- Selfish clients cooperate iff

Utility (cache w/o cooperation) **< Utility** (cache with cooperation)
- Cost (total access to remote caches)
- Cost (total SERVES to peers)

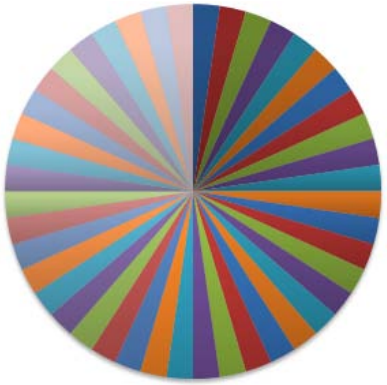
In The New Model

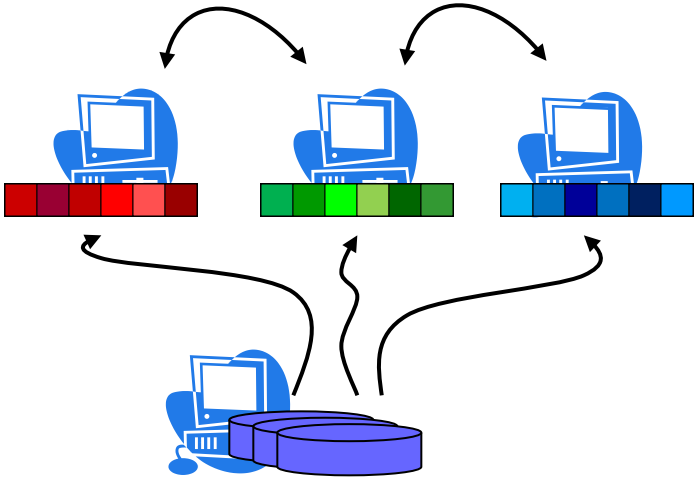
- When to cooperate?
- How to cooperate?



Cooperative DHT

- Distributed Hash Table [Stoica et al. 2003]
  - Distributed key assignment
- **Altruistic** clients
  - Cache : blocks they are assigned
  - SERVE: all requests

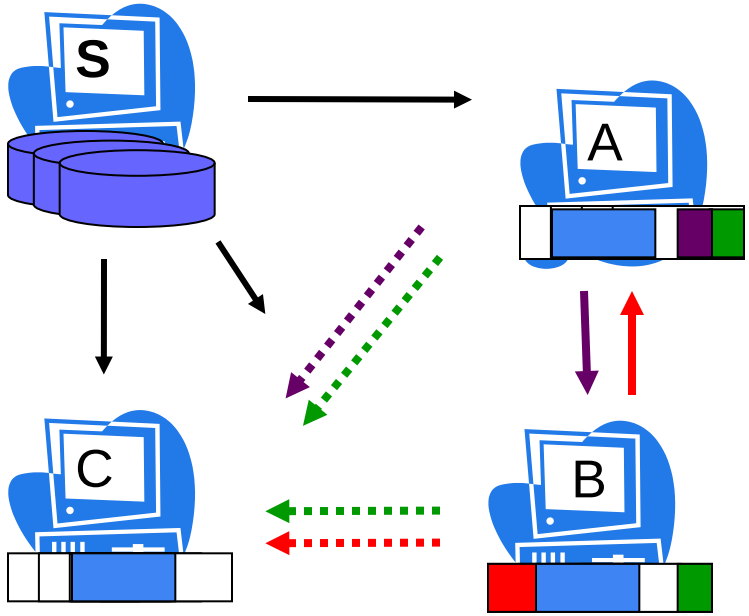




Altruistic — Selfish

Cooperative Peer to Peer

- BitTorrent based caching
  [Cohen 2003]
  - Server tracks accesses ("tracker")

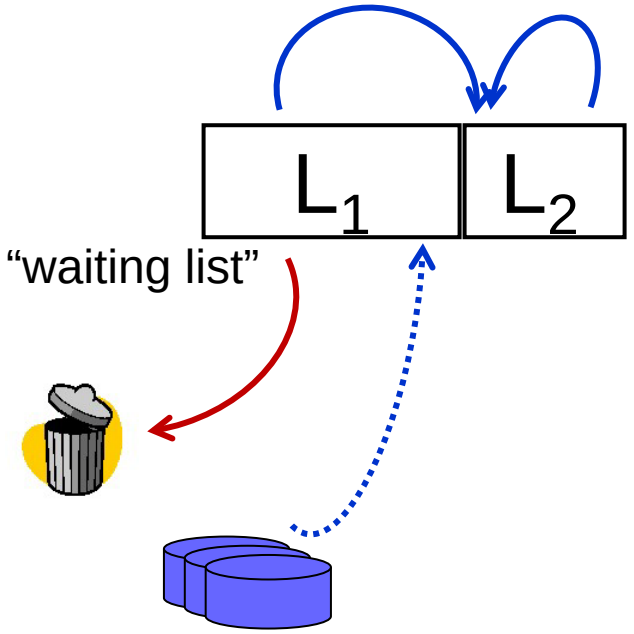


- **Selfish** clients
  - Cache : LRU without replication
  - SERVE: peers with positive balance ("tit for tat")

Altruistic — Selfish

Cooperative ARC

- Adaptive Replacement Cache
  [Megiddo and Modha 2003]
  - New blocks in L_1
  - Useful blocks in L_2



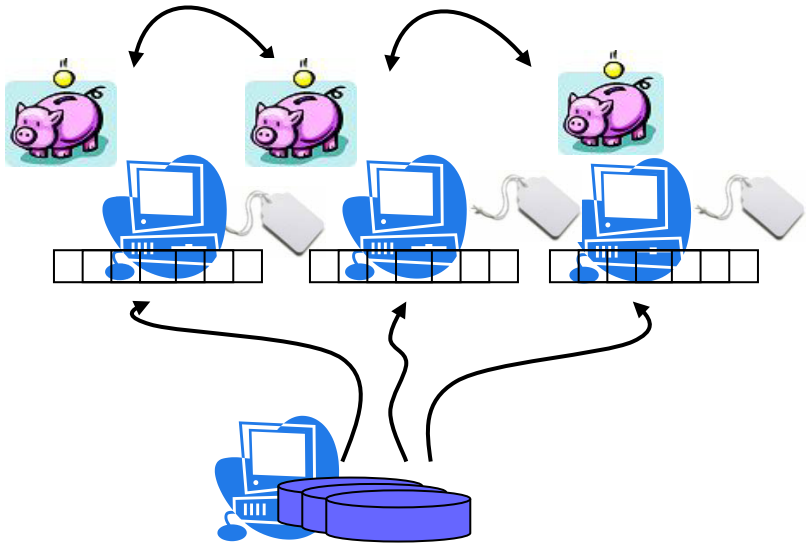
- **Very selfish** clients
  - Cache : ARC with replication in L_2
  - SERVE: peers with positive balance

Utility Based Cooperative Caching

- Use utility calculations [Yadgar et al. 2008]
  - Block accesses hinted or derived
  - Server constructs configuration

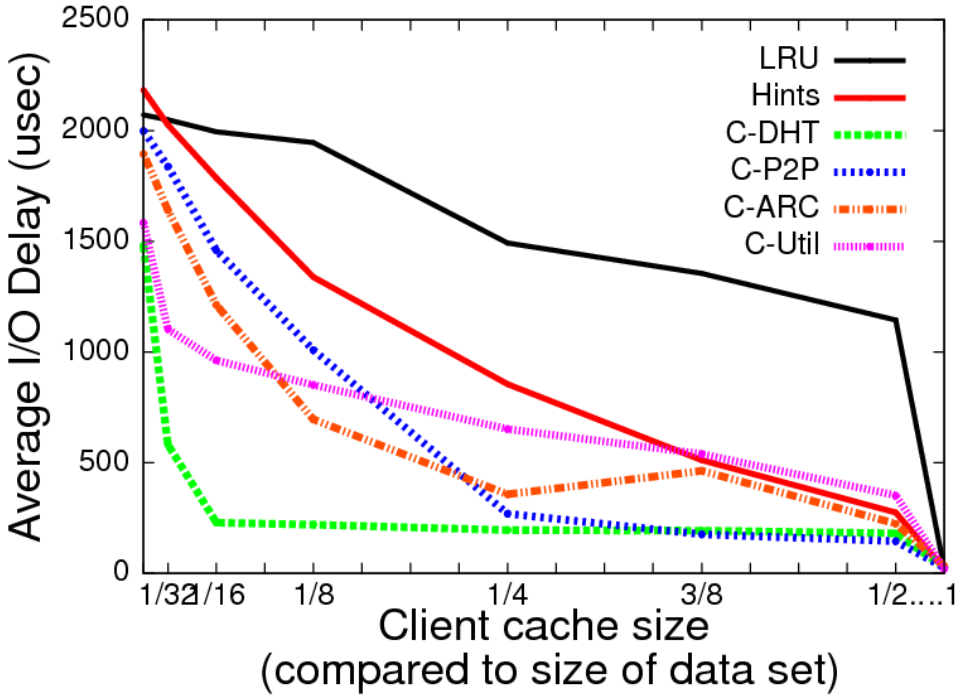
- **Very selfish** clients
  - Cache : by configuration
  - SERVE: by configuration

→ As long as utility increases



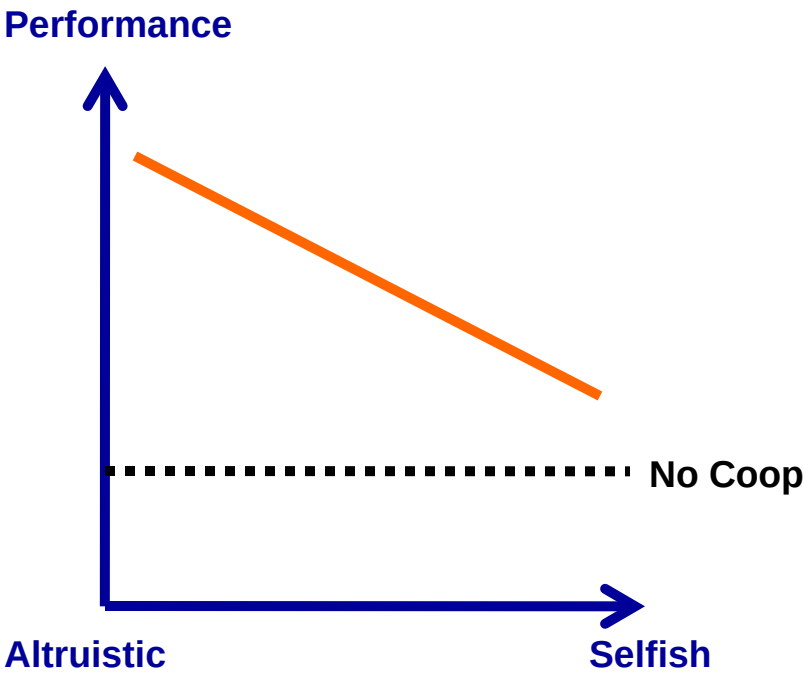
Altruistic — Selfish

Decision Support Workload (TPCH)

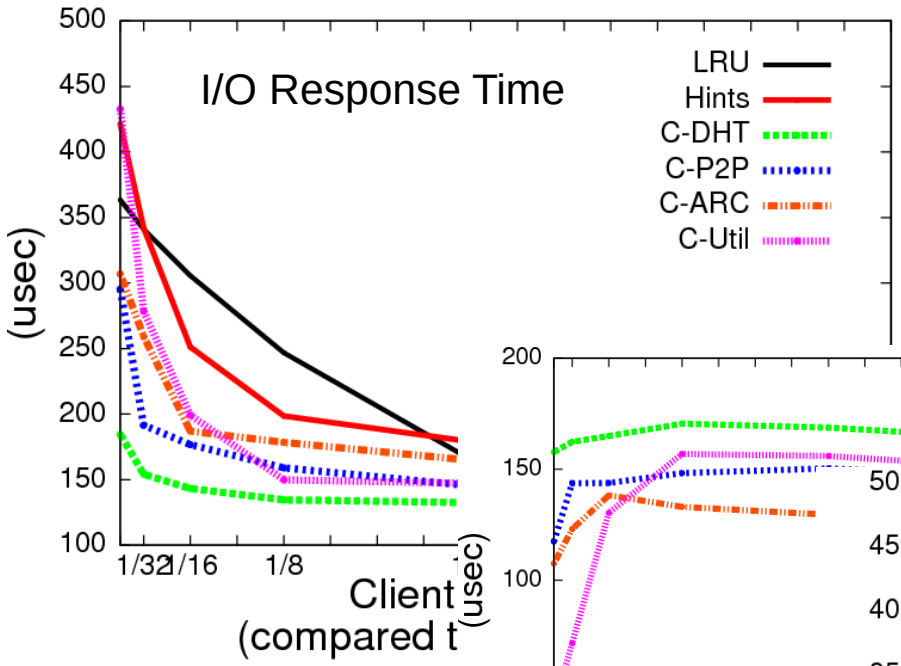


20 peers

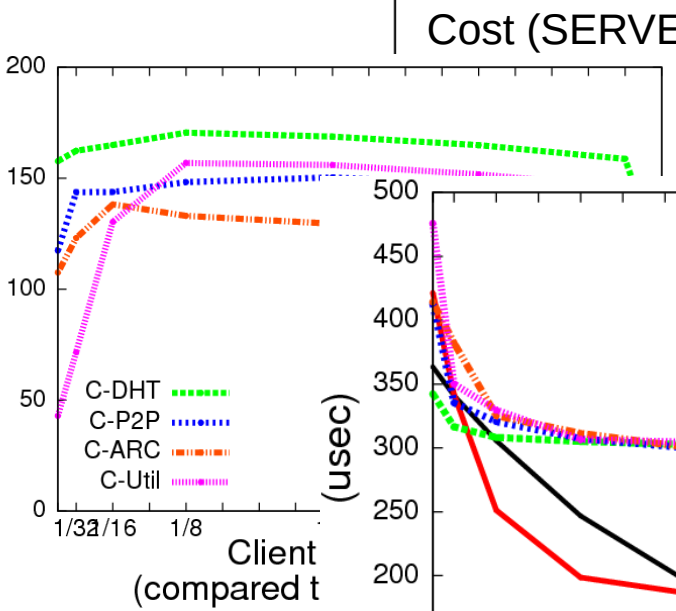
- Uniform distribution
- Short term information
- Partial correlation

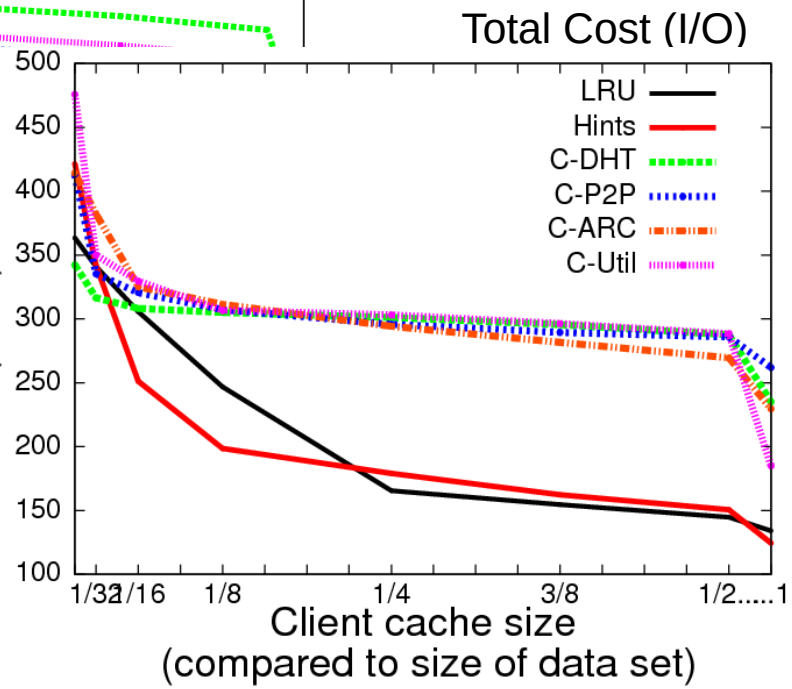


OLTP Workload (TPCC)



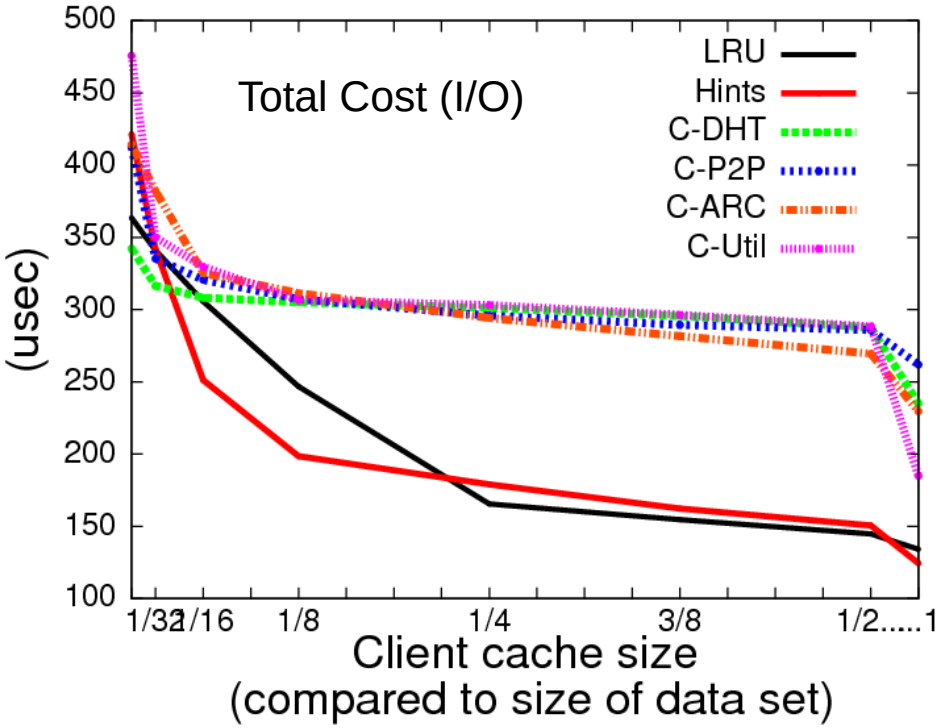
20 peers



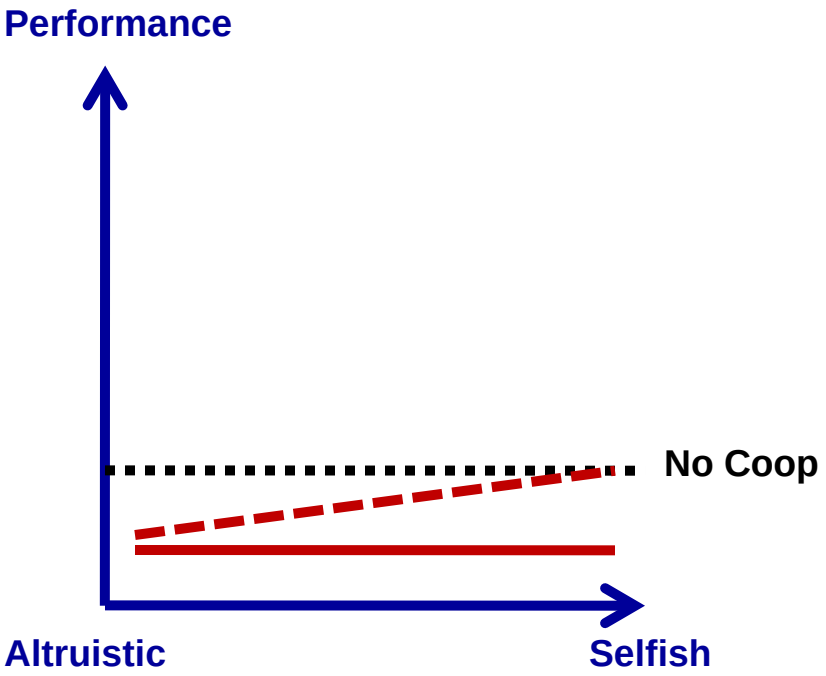


- Long tail distribution
- Partial information
- Strong correlation

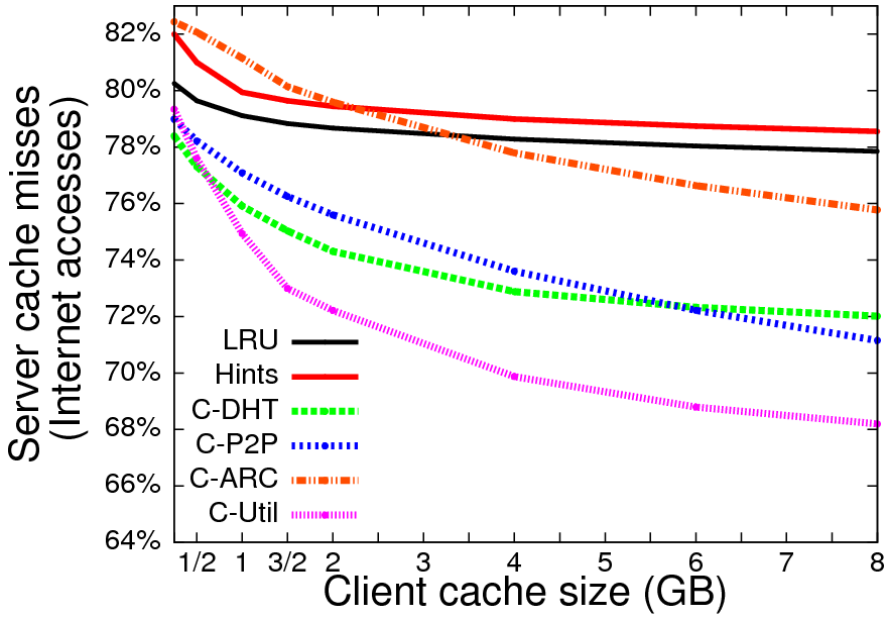
OLTP Workload



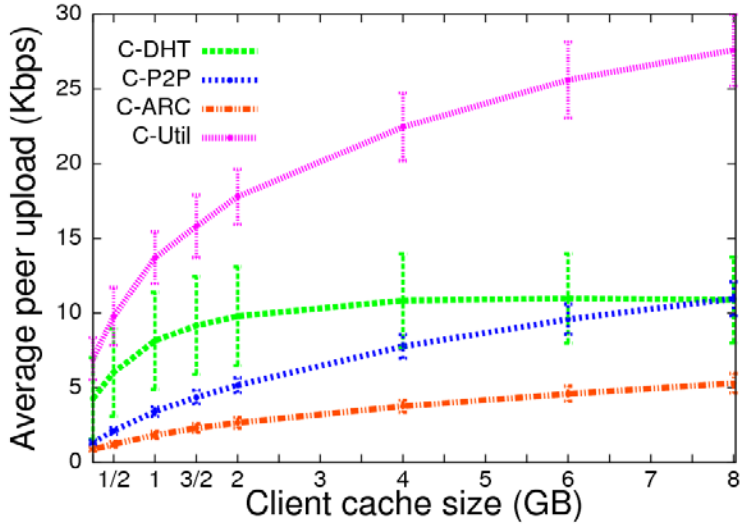
- Long tail distribution
- Partial information
- Strong correlation



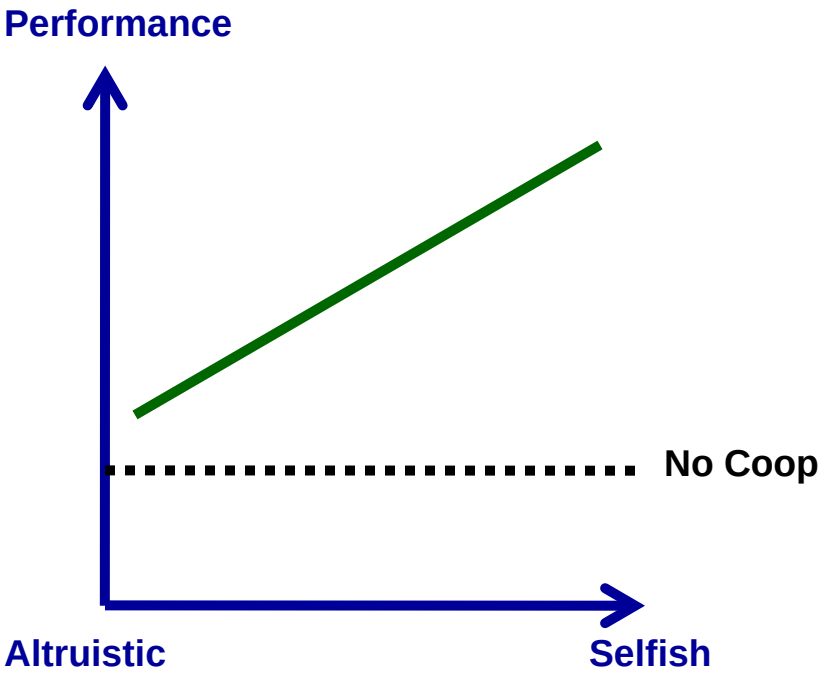
“YouTube” Workload



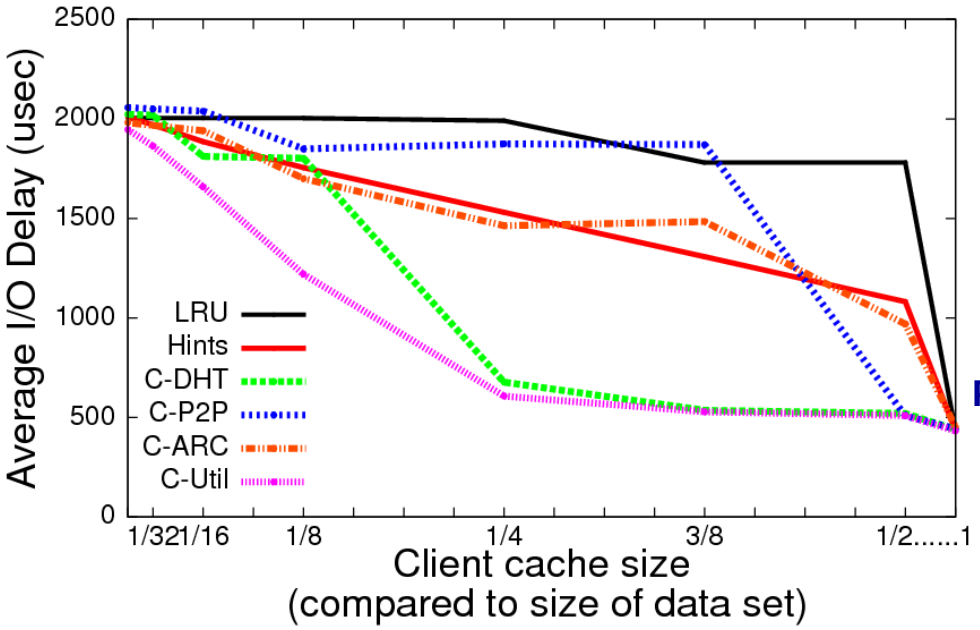
50 peers



- Heavy tail distribution
- General information
- Medium correlation
- Cheap SERVE

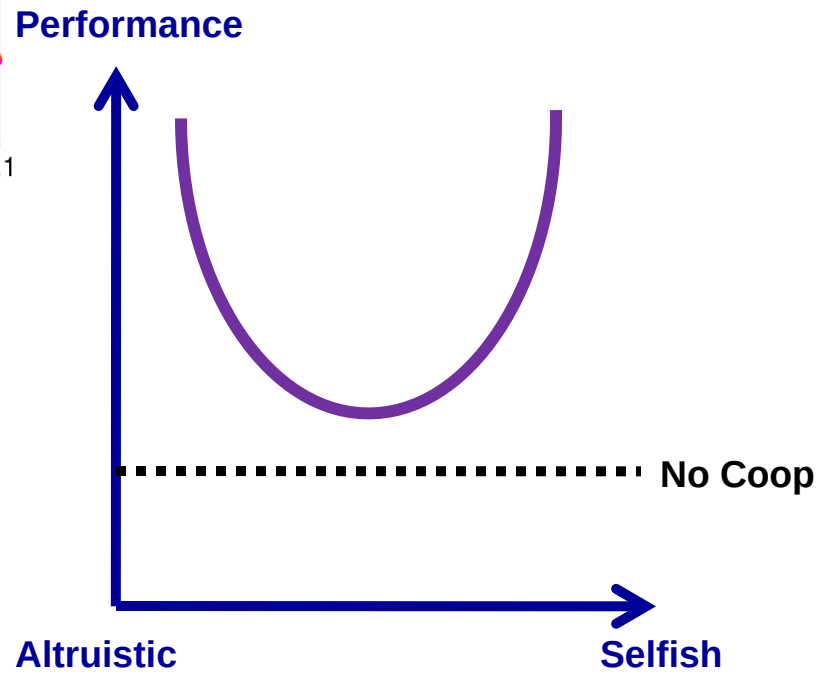


TPCH Queries



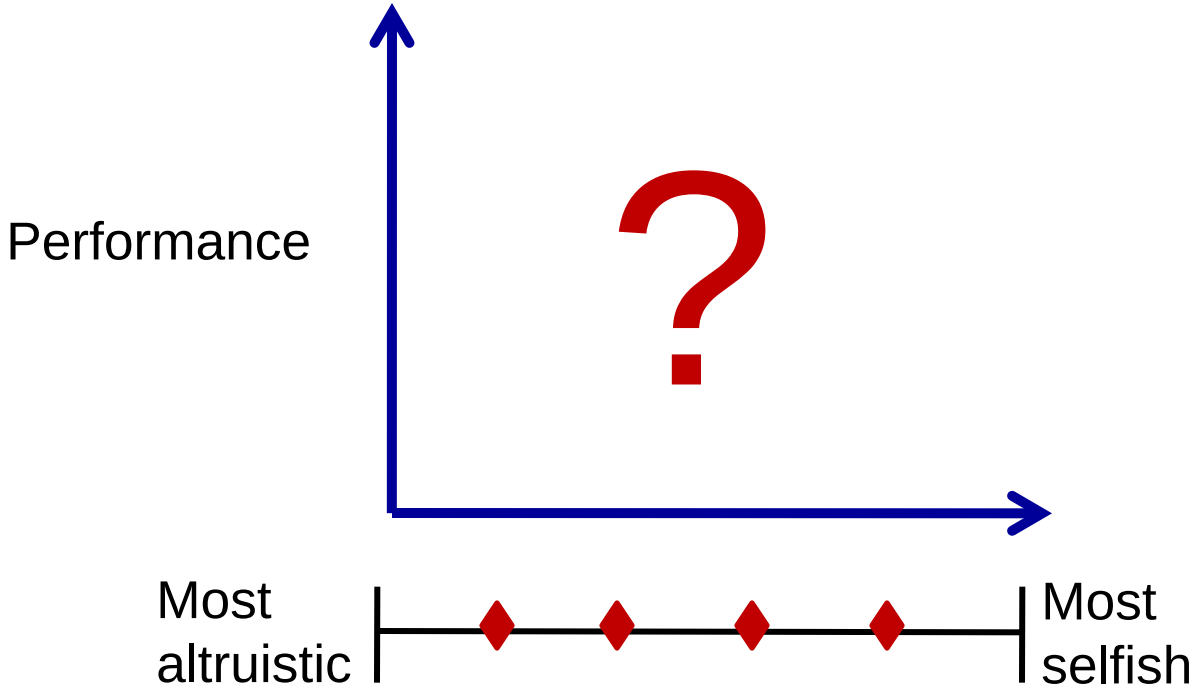
Queries 3,10, and18

- Uniform distribution
- Accurate information
- Medium correlation

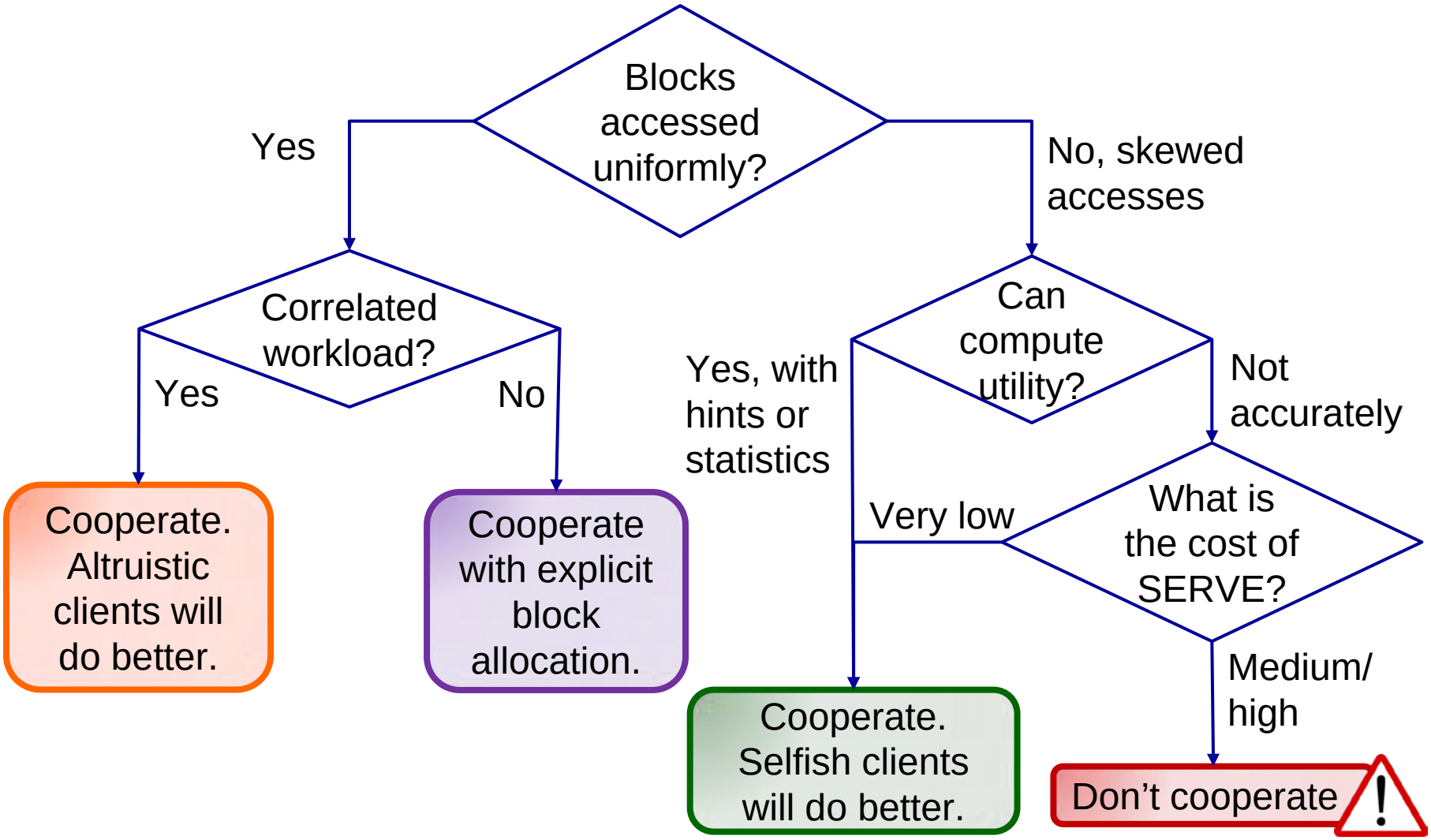


Conclusions

- When to cooperate?
- How to cooperate?



Conclusions



Coming Up

- Saving energy
- Federated CDN