Performance Models and Systems Optimization for Disk-Bound Applications

Mithuna Thottethodi[†], Rahul Shah[*], Vijay Pai[†], T.N. Vijaykumar[†], Jeffrey S. Vitter[‡]

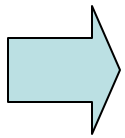
[†]School of Electrical and Computer Engineering
Purdue University

[‡]Department of Computer Science
Purdue University

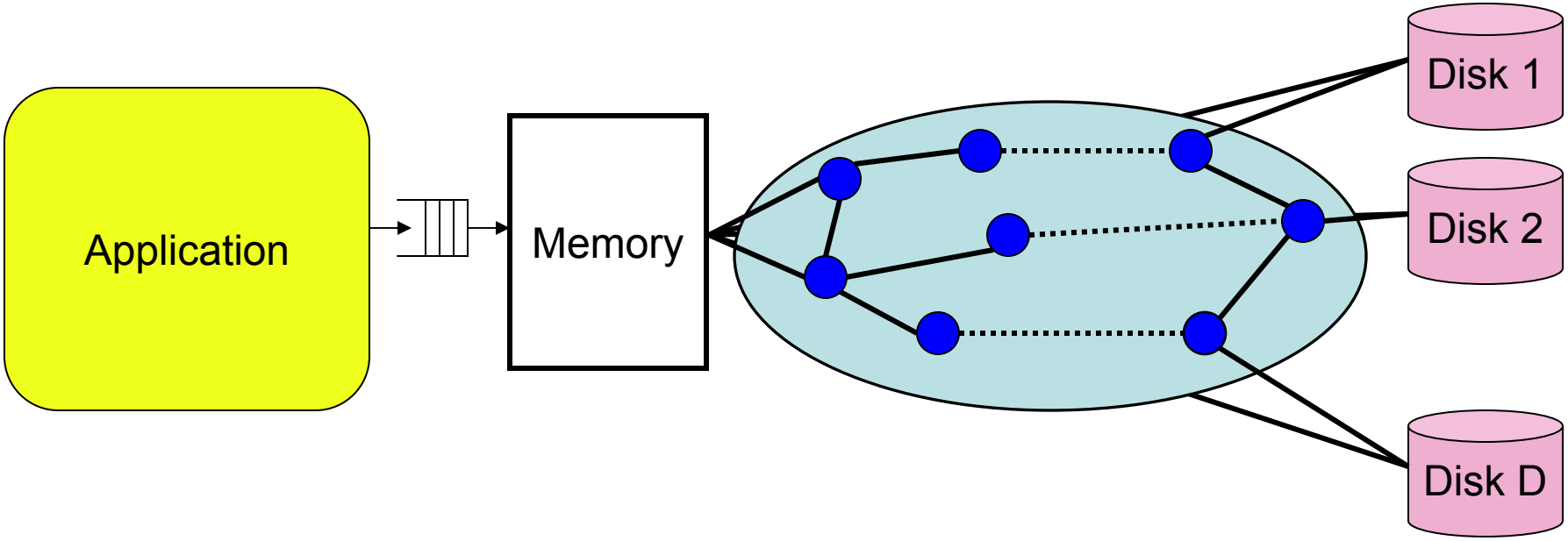
[*]Soon to be at Louisiana State University

Goal Recap

- Superior models
  - Workload characteristics
  - Caching
  - SAN contention
  - Disk access model



- More informed optimizations



Outline

- Theory
  - Key findings on caching/prefetching
- Systems
  - Key results
  - Details on one set of results
- Publications
  - Published, under review and pending submissions
- Education

Caching/Prefetching

- PDM
  - D parallel disks, M block cache, unit block access time
- Offline problem
  - All requests known
- Online problem
  - Lookahead L
  - How close to offline optimal?
    - Competitive analysis

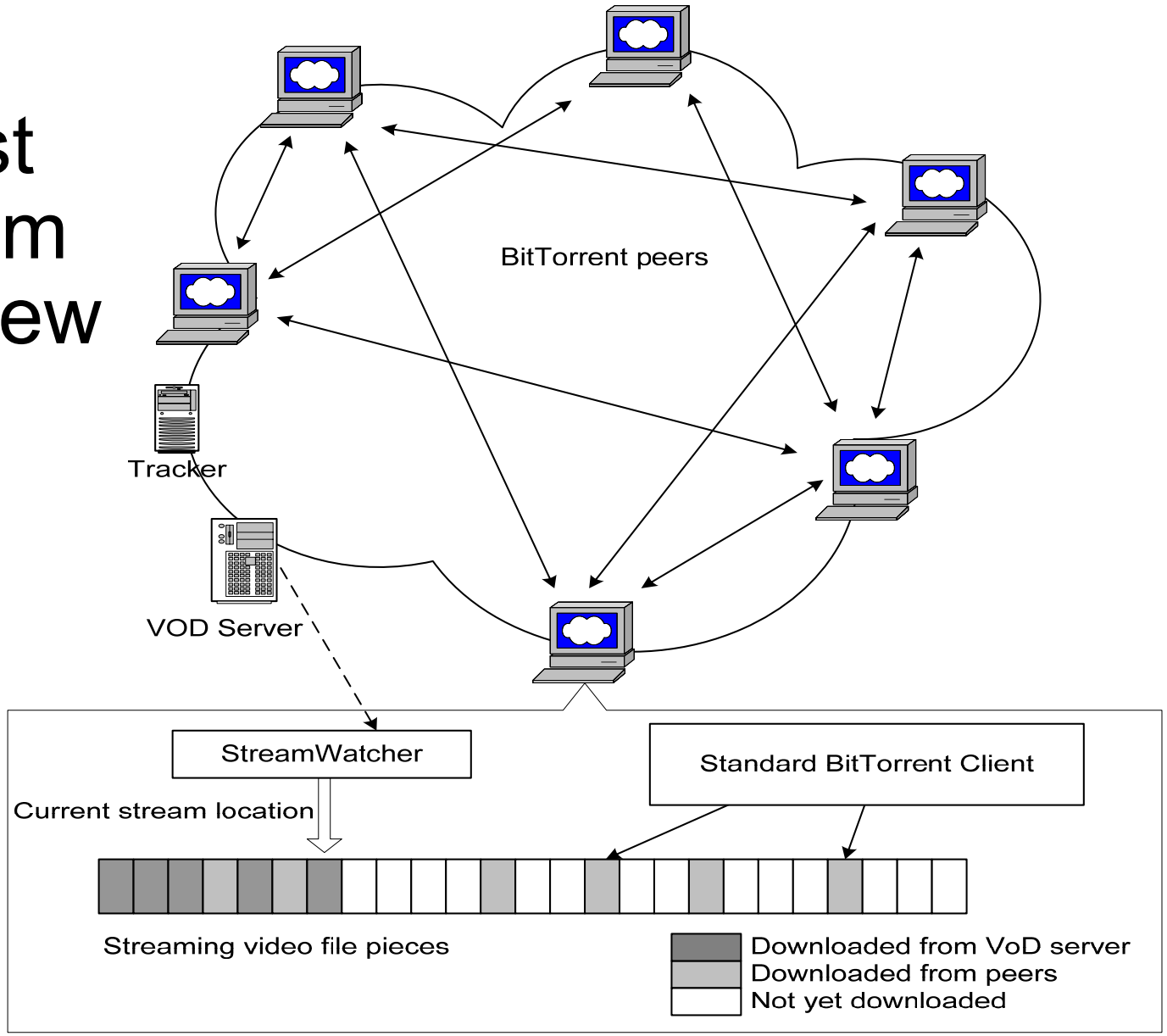
Key Findings

- New Lower bounds
- On Parallel Disk Model:
  - When lookahead $L = M$
  - Any online algorithm : $\Omega(D)$ competitive
  - When $L < M$, competitive ratio is at least $M-L$
- Randomization w/Oblivious adversary:
  - For lookaheads $L < M$,
  - Tight upper and lower bound of $\Theta(D \log ((M-L)/D))$

VoD Storage Server

- High bit-rate VoD server model
  - Disk-occupancy-based
  - Validation: Within 11% of prototype
- Load balance with performance imbalance
  - Fail-stutter faults
  - Based Random duplicate allocation (RDA) [Korst1997]
- Bit-Torrent Modification for VoD storage server
  - Toast saves VoD server 70-90% of data transfer load

Toast System Overview



Client Modifications

- StreamWatcher
  - Tracks client location in viewing stream based on configured bitrate
  - If a piece is reached that has not yet been received from BT peers, sends request to VoD server in a separate thread
  - When piece is received, it is added to the BT file store, (so it will not be downloaded again) and advertised to peers as if it had been received from BT

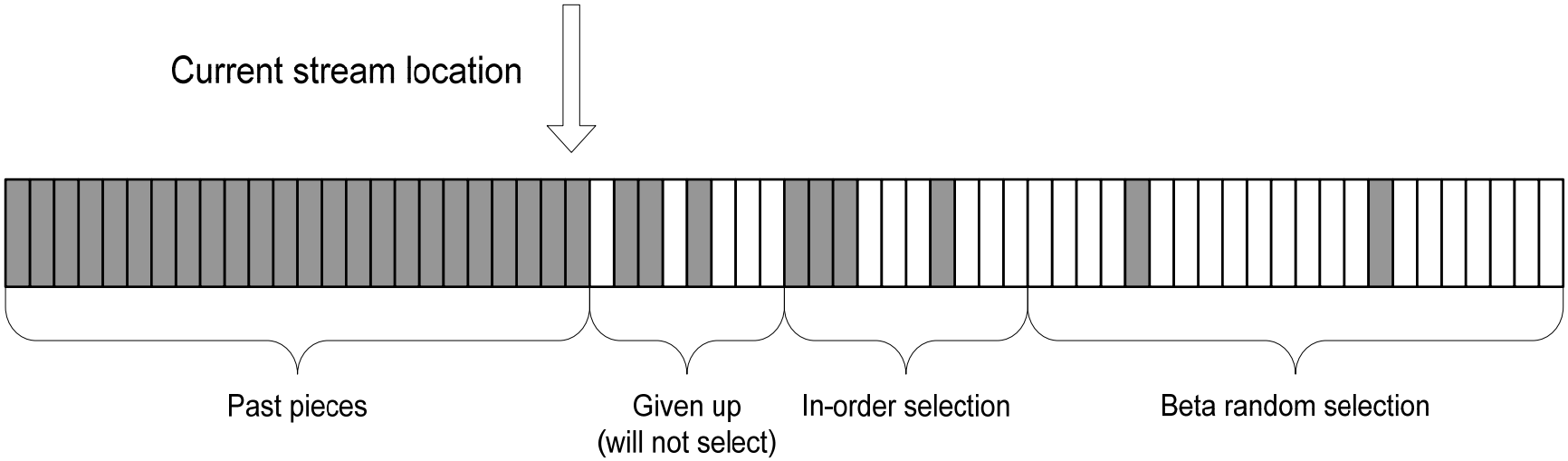
Client Modifications (cont)

- PiecePicker
  - Which piece to request from peers next?
  - BitTorrent uses local rarest-piece-first (with random to break ties)
  - Bias toward beginning (or upcoming pieces) to avoid making server requests
  - But keep good properties of existing protocol: randomness assures clients don't all have same pieces, allowing them to trade

Piece Picker Policies

- Rarest
  - BitTorrent default
- In-order
  - Select the piece that will be needed soonest
- Beta
  - Beta distribution with $\alpha = 1$, $\beta = 2$
  - Pdf decreases linearly
- Hybrid

Hybrid Picker Policy

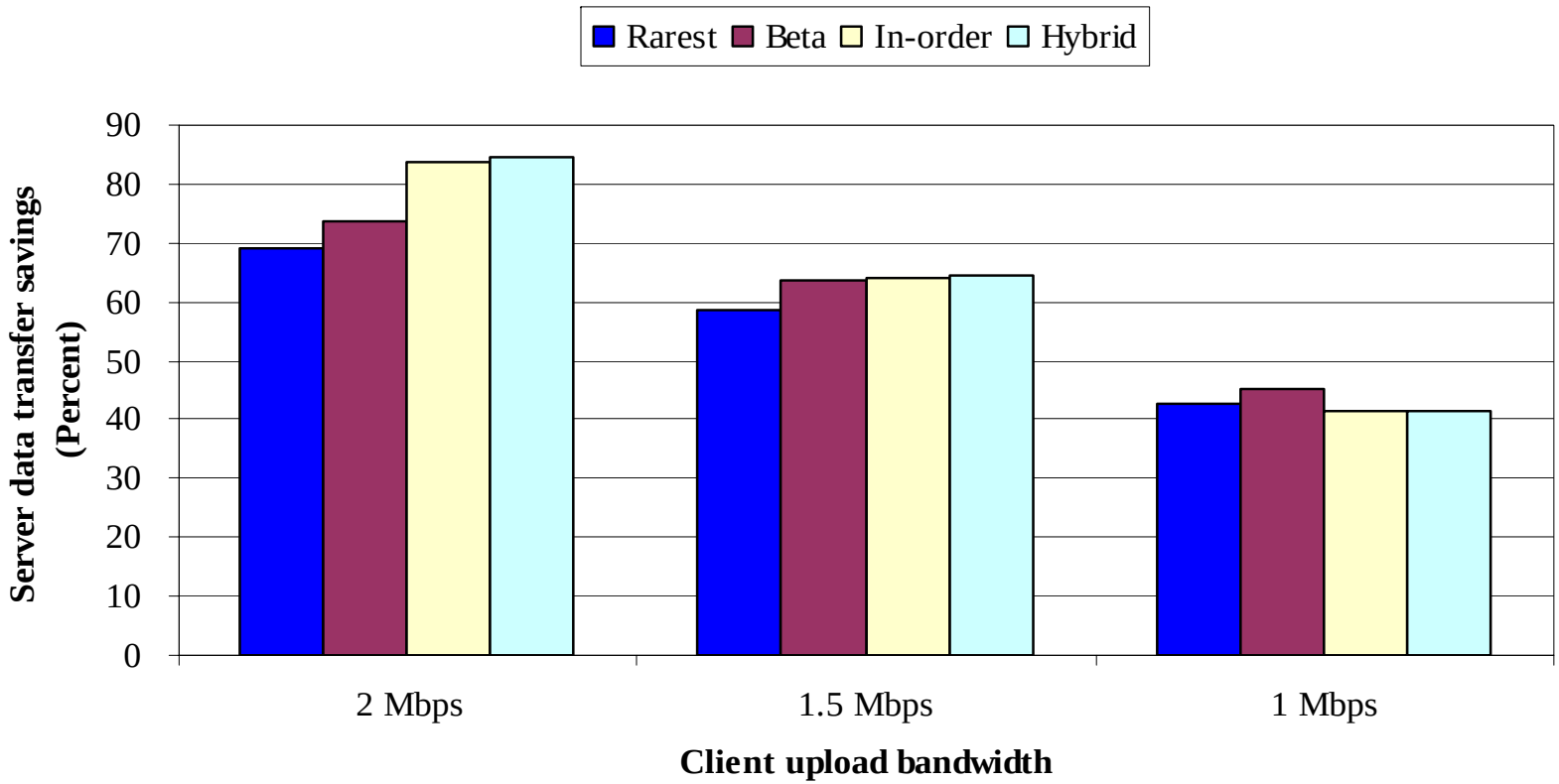


- In-order selection range
  - First selects pieces needed soon
- Beta random selection range
  - Select if In-order range filled or not available

Test Scenarios

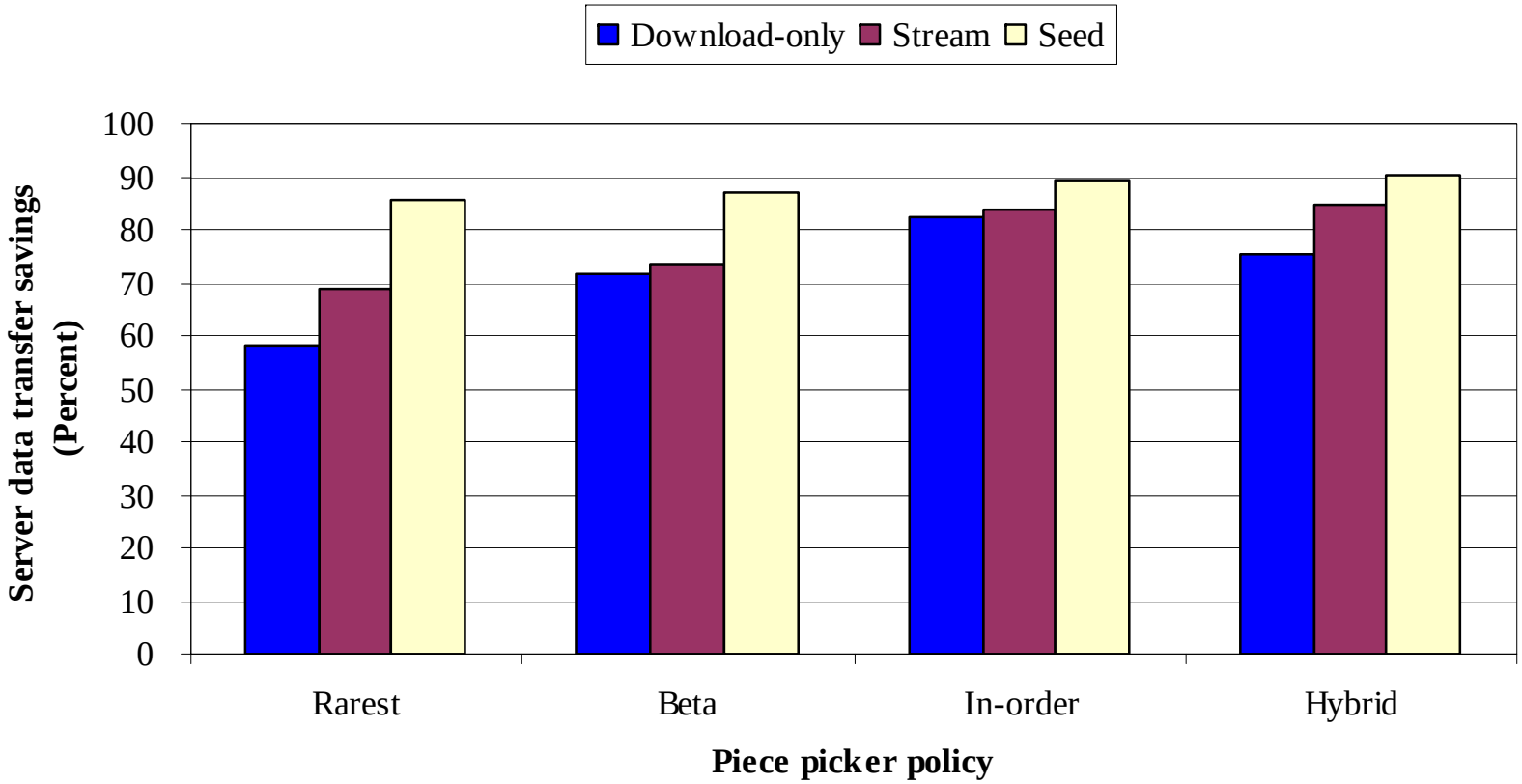
- Client Behavior
  - Download-only: Client leaves after download
  - Stream: Client stays until end of movie
  - Seed: Client stays until end of the test
- Local Buffer Size Limit
  - Client has only a limited amount of disk space
  - 100%, 75%, 50%, 25% of the file kept

Piece Picker policy comparison



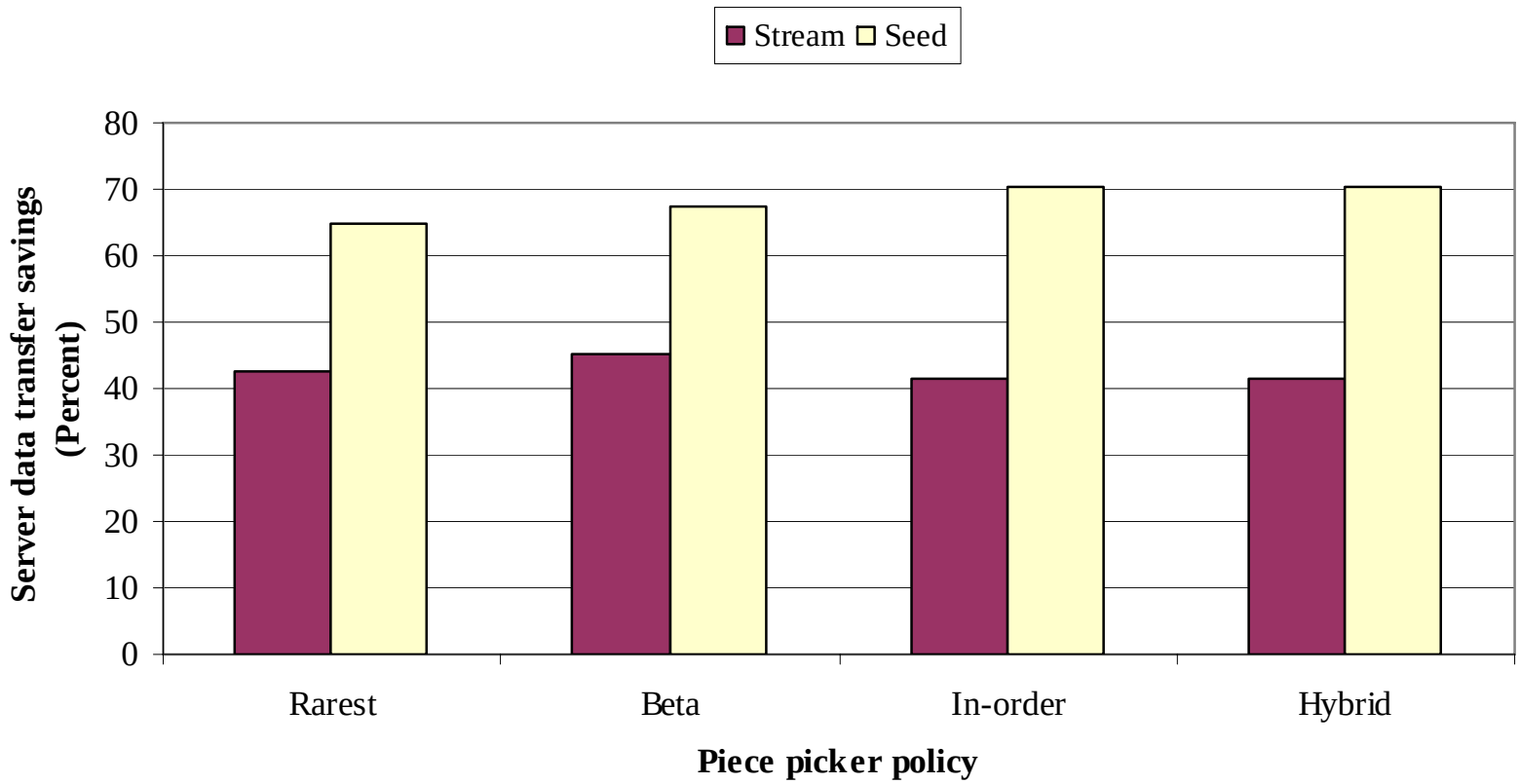
- Stream client sharing policy used

Effect of client sharing behavior



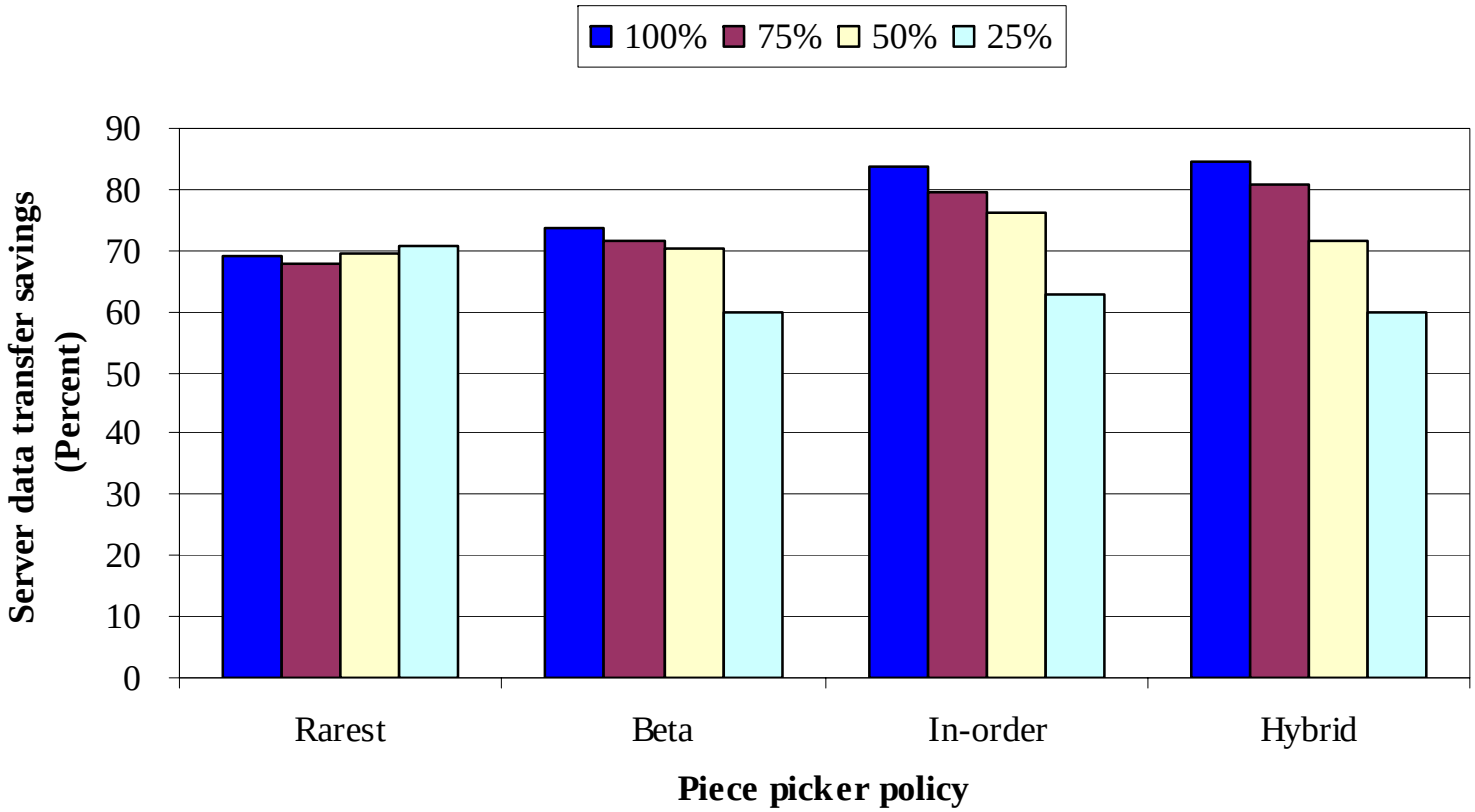
- 2 Mbps max client upload rate used

Effect of seeding in 1 Mbps upload rate



- Better than 1.5 Mbps stream policy

Effect of limiting local storage



- Stream client sharing policy, 2 Mbps upload rate

Publications

- Y. Choe, D. Schuff, J. Dyaberi, and V. Pai, Improving VoD Server Efficiency with BitTorrent, *ACM Multimedia*, Augsburg, Germany, (September, 2007).
- Y. Choe and V. Pai, Achieving Reliable Parallel Performance in a VoD Storage Server Using Randomization and Replication, *IEEE International Parallel & Distributed Processing Symposium*, Long Beach, CA, (March, 2007).
- Y. Choe, C. Douglas, V. Pai, A Model and Prototype of a Resource-Efficient Storage Server for High-Bitrate Video-on-Demand, *IPDPS Workshop on Performance Modeling, Evaluation, and Optimization of Parallel and Distributed Systems* , (March, 2007)

Pre-publication Results

- Under Review
  - W. Hon, R. Shah, P. J. Varman, J. S. Vitter. "Tight competitive ratios for caching / prefetching on parallel disks," Submitted to ***Symposium on Discrete Algorithms*** 2008
- Other interesting results
  - Pending submissions
    - Distributional Models
    - MapReduce Performance Optimization
  - Discuss offline

Education

- Two PhD students supported
  - One graduated with PhD
    - Thesis: “Design and Implementation of a Resource-Efficient Storage Server for VoD”
  - One interning at Google (Disk access Scheduling)
- Expanded storage coverage in graduate-level Parallel Architecture course
  - MapReduce projects
    - Hadoop and HDFS hacking

Questions