CS 525M – Mobile and Ubiquitous Computing Seminar

Ioanna Symeou

Broadcast Disks

Broadcast Disks: Data Management for Asymmetric Communication Environments

Swarup Acharya, Brown University
Rafael Alonso MITL
Michael Franklin, University of Maryland
Stanley Zdonik, Brown University

Appears in Proceedings of the ACM SIGMOD Conference, San Jose CA, May 1995

Broadcast Disks: Abstract & Introduction

- Asymmetric Communication Environments: downstream capacity differs from upstream capacity (ex. Wireless networks)

- Two reasons for asymmetry:
  - Bandwidth limitations
  - Patterns of information flow

- Improved performance in asymmetric communication environments

Broadcast Disks: Abstract & Introduction

- Pull based systems Vs push based systems:
  - Push based for asymmetric environments so channel becomes a disk
  - Items broadcasted more often appear to be on faster spinning disks closer to the clients.

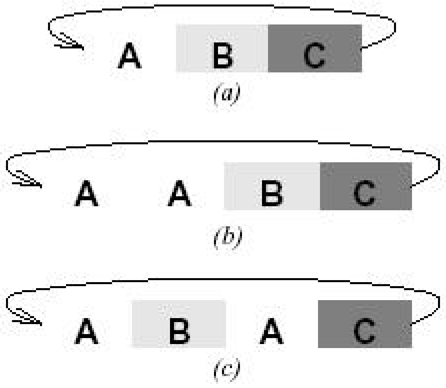
- Broadcast disks: Broadcast technique providing “illusion” of having data on multiple disks running in different speeds.

- Broadcast algorithms: Based on client population and data access probabilities
- New cache management policies: How each client manages its own cache

Broadcast Disks: Broadcast Algorithms

- Simple scenario: Flat broadcast
  - Server broadcasts data from all requests
  - Wait time: half broadcast period
- Broadcast items with different frequency
  - Bandwidth allocation problem
  - Match needs of different clients

-Additional advantages



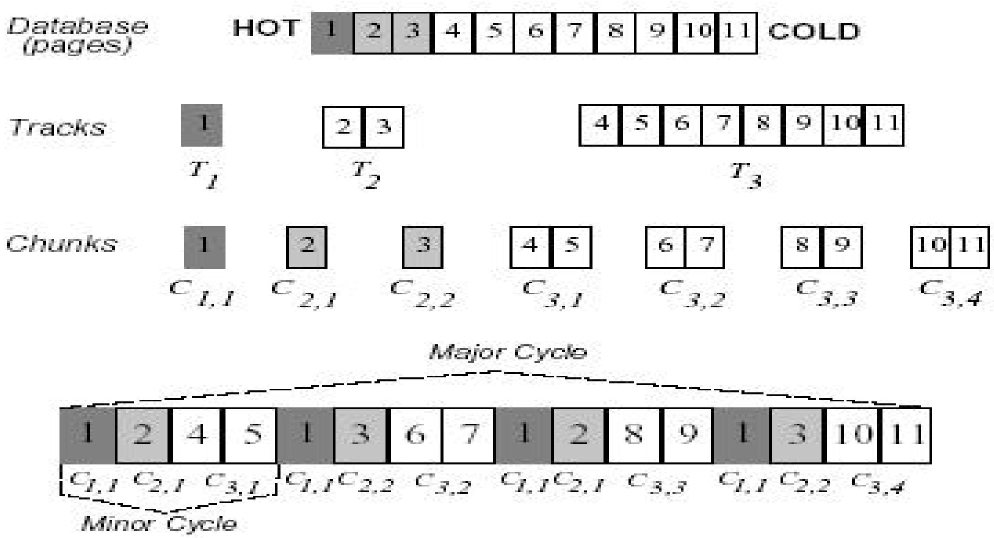
Access Probability			Expected Delay (in broadcast units)		
A	B	C	Flat (a)	Skewed (b)	Multi-disk (c)
0.333	0.333	0.333	1.50	1.75	1.67
0.50	0.25	0.25	1.50	1.63	1.50
0.75	0.125	0.125	1.50	1.44	1.25
0.90	0.05	0.05	1.50	1.33	1.10
1.0	0.0	0.0	1.50	1.25	1.00

Broadcast Disks: Broadcast Algorithms

- Broadcast program generator
  - Order pages to lists from hottest to coldest
  - Partition lists to multiple ranges (disks) based on access probabilities
  - Chose relative frequency of broadcast for each disk
  - Split disk into smaller units (chunks)
    - Max_chunks = LCM of frequencies
    - Num_chunks = Max_chunks / rel_frequency
  - Send chunks:

```
for i = 0 to max_chunks - 1
  for j = 1 to num_disks
    broadcast chunk j , i mod(num_chunks(j))
```

Broadcast Disks: Broadcast Algorithm



- Three parameters influence broadcast:
  - Number of disks determines number of frequencies
  - Number of pages per disk and relative frequencies determine size of broadcast
- Disadvantage: Some slots may be unused
  - Send extra info
  - Small fraction of slots
  - Adjust relative frequencies to reduce unused slots

Broadcast Disks: Cache management

- Traditionally: Clients cache *their* hottest data
- BUT in push based systems broadcast might not be optimal for a client
  - Inaccurate/time sensitive information from client about its access distribution
  - Higher priority to other clients/large client population
- Cache data for which local access probability is much greater than their broadcast frequency
- Cost-based page replacement: When replacing a page on a cache miss calculate cost of obtaining the page
  - Replace page with lowest P/X ratio
  - Requires though knowledge of access probabilities and comparison of values for all pages

Broadcast Disks: Environment model

- Client parameters

CacheSize	Client cache size (in pages)
ThinkTime	Time between client page accesses (in broadcast units)
AccessRange	# of pages in range accessed by client
θ	Zipf distribution parameter
RegionSize	# of pages per region for Zipf distribution

- Server parameters

ServerDBSize	Number of distinct pages to be broadcast
NumDisks	Number of disks
$DiskSize_i$	Size of disk i (in pages)
Δ	Broadcast shape parameter
Offset	Offset from default client access
Noise	% workload deviation

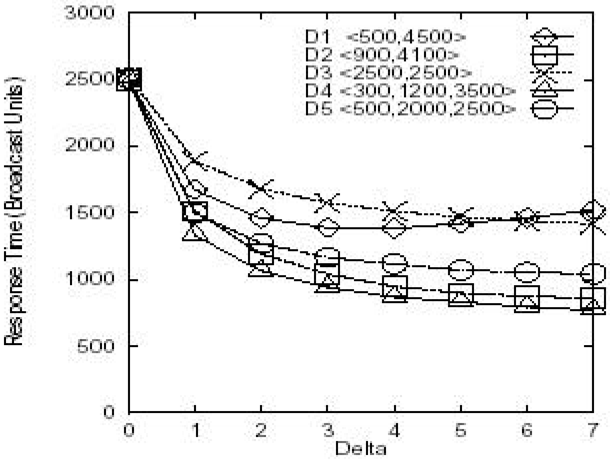
- rel_frequency(i) / rel_frequency(N) = (N – i)? + 1

Broadcast Disks: Experiments and results

- Parameters settings

ThinkTime	2.0
ServerDBSize	5000
AccessRange	1000
CacheSize	50(5%), 250(25%), 500(50%)
Δ	1,2,...,7
θ	0.95
Offset	0, *CacheSize*
Noise	0%, 15%, 30%, 45%, 60%, 75%
RegionSize	50

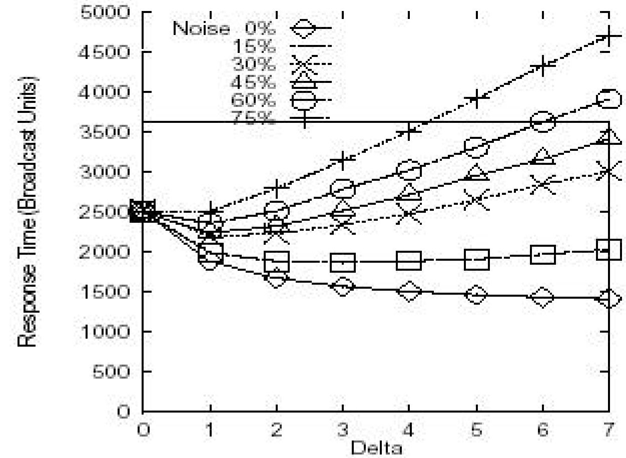
- Experiment 1: No caching, 0%Noise



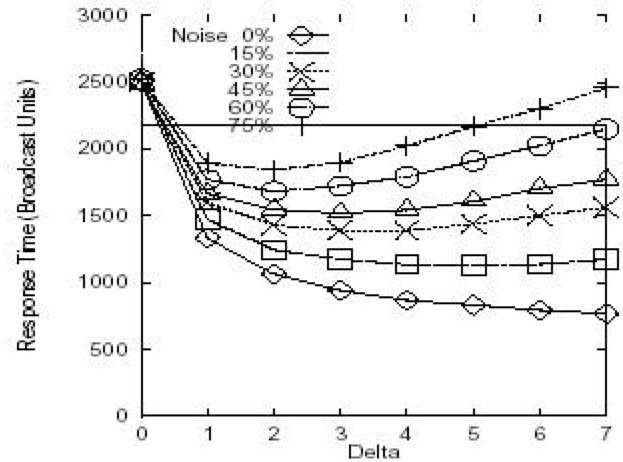
Broadcast Disks: Experiments and results

•Experiment 2: No caching and noise

• Two disk configuration
  <2500,2500>



•Three disk configuration
  <300,1200,3500>



Broadcast Disks: Experiments and results

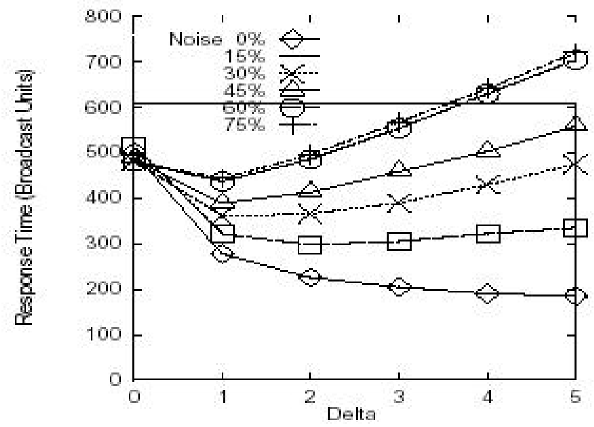
- Experiment 3: Caching and noise

  Three disk configuration

  <300,1200,3500>

  P replacement policy

  cache size = 500



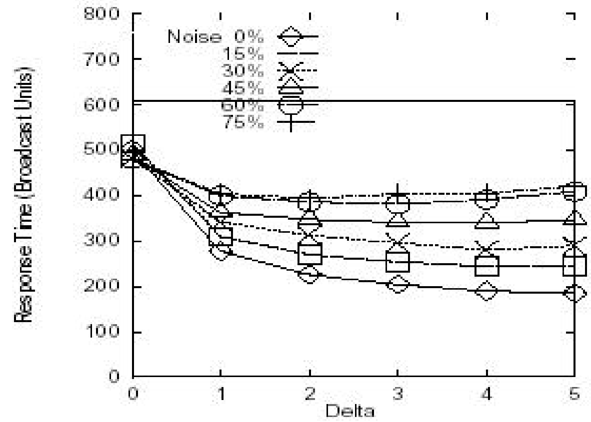
- Experiment 4: Caching and noise

  Three disk configuration

  <300,1200,3500>

  P/X replacement policy

  cache size = 500



Broadcast Disks: Experiments and results

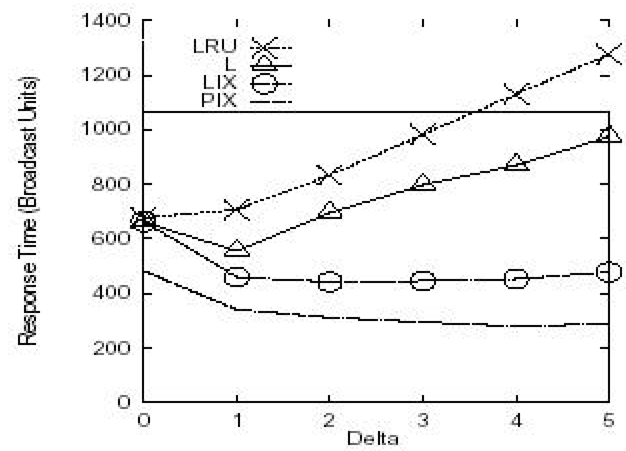
- Experiment 5: Caching and noise

  Three disk configuration

  <300,1200,3500>

  cache size = 500

  L/X replacement policy



- L/X replacement policy
  - One list for each disk
  - Always enter new page to a list according to its disk
  - Replace page with lower lix value
    - $lix = p_i / rel\_frequency$
    - $p_i = ? \ / \ (CurrentTime - t_i) + (1 - ?) p_i$

Broadcast Disks: Conclusions

- Applicable technique for asymmetric environments
- Two and three level disks can have better performance than flat broadcast
- Need for new cache replacement policy (cost based caching)
- What about:
  - Write and read case?
  - Data change in each cycle?
  - Intelligent clients?
  - Pre-fetching?
  - User initiated broadcast?