CSE6488:
Mobile Computing Systems

Sungwon Jung

Dept. of Computer Science and Engineering
Sogang University
Seoul, Korea
Email : jungsung@sogang.ac.kr

Your Host

- Name: Sungwon Jung
- Email: jungsung@sogang.ac.kr
- Course web page: http://mclab.sogang.ac.kr
- Class time: Wed. 15:00 – 17:45 AS912
- Office: AS811, phone: 705-8930
- Office hours: Tu Wed Th. 10:00 – 12:00, or by appointment/email

Course Logistics

- No textbook. Research papers from major database conferences and journals.
  - VLDB, SIGMOD, ICDE
  - CIKM, SSDBM, SSTD, GIS
  - Journals: ACM TODS, IEEE TKDE, VLDB Journal, etc.
- Grading Criteria
  - Mid Exam 35%
  - Final Exam 40%
  - Class participation 5%
  - Term Project 20%

Course Topics

- Data Dissemination in Wireless Broadcast Environment
- Indexing Techniques for Spatial Data
- LBS Query Processing Methods in Mobile Clients & Server Environment
- LBS Query Processing Methods in Road Network DB
- Range, kNN, Shortest Path Query Processing in Wireless Broad cast Environments
- Preference Aware Query Processing Methods
- Location Data Privacy

Data Dissemination in Mobile Computing Environments (1)

Sungwon Jung

Dept. of Computer Science and Engineering
Sogang University
Seoul, Korea
Email : jungsung@sogang.ac.kr

Properties of Mobile Computing Environment

- Mobility
- Portability
  - Limited Computing Resources
- Wireless Communications
  - Weak and Intermittent Connectivity
  - Frequent Disconnections

6

Data Dissemination in Mobile Computing Environments

- Communication asymmetry
  - Network asymmetry
    - High Bandwidth Downlink
    - Low Bandwidth Uplink
  - Client to server ratio
  - Data volume
- Data Dissemination options
  - Pull
  - Push (Wireless Data Broadcasting)
    - Allows simultaneous access

7

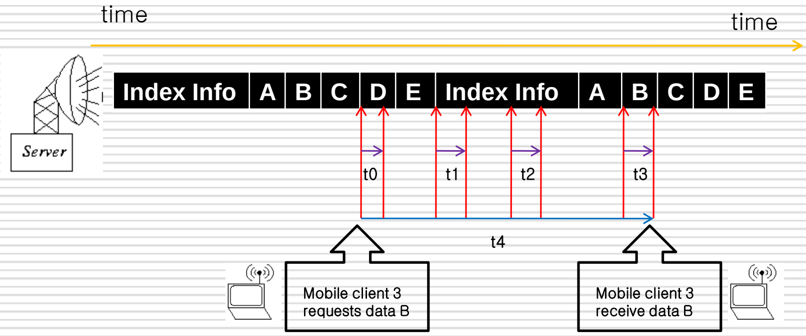
Commercial Wireless Broadcasting Services

- SPOT(Smart Personal Object Technology)
  - http://www.microsoft.com
    - .NET Micro Framework
    - Based on *DirectBand* Network using FM radio subcarrier frequencies
- StarBand
  - http://www.starband.com
- DIRECTWAY
  - http://www.directway.com

8

Parameters of Concern in Wireless Data Broadcast

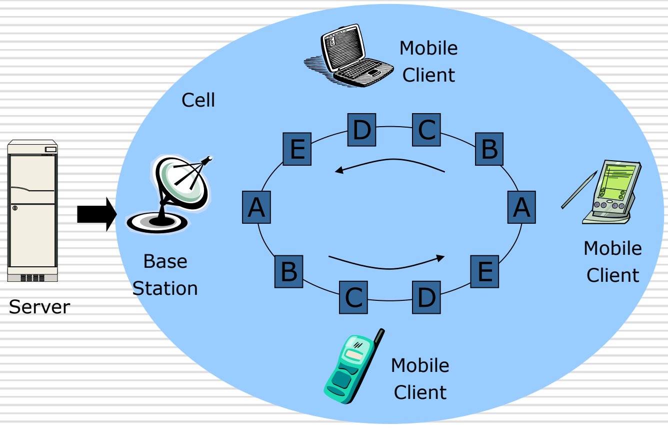
- Tuning Time: t0 + t1 + t2 + t3
- Access Time: t4



Research Issues on Wireless Data Broadcast

- Scheduling Data Broadcast Over a Single Wireless Channel
- Scheduling Data Broadcast Over Multiple Wireless Channels
- Indexing Techniques for Broadcast Data
- Cache Invalidation Methods for Mobile Clients
- Concurrency Control Methods for Mobile Transactions

Scheduling Data Broadcast Over a Single Wireless Channel



Scheduling Data Broadcast Over a Single Wireless Channel

- Restrictions of a broadcast environment
  - the client population and their access patterns do not change
    - the content and organization of the broadcast program remain static
  - data is read-only
    - there are no updates either by the clients or at the servers
  - clients retrieve data items from the broadcast on demand
    - there is no prefetching
  - clients make no use of their upstream communications capability
    - they provide no feedback to servers

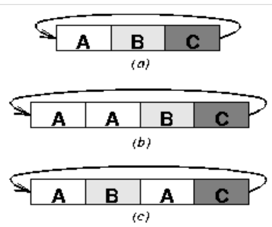
Scheduling Data Broadcast Over a Single Wireless Channel

- Two main interrelated issues
  - Given a client population and a specification of their access probabilities for the data items, how does the server construct a broadcast program?
  - Given that the server has chosen a particular broadcast program, how does each client manage its local data cache to maximize its own performance?

Scheduling Data Broadcast Over a Single Wireless Channel

- Flat Broadcast: (a)
- Skewed (random) Broadcast: (b)
- Multi-disk (regular) broadcast: (c)
  - Broadcast Disk

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



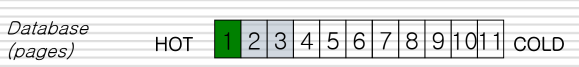
Scheduling Data Broadcast Over a Single Wireless Channel

- Broadcast Program Generation
  - Assume that data items are "pages", that is, they are of a uniform, fixed length

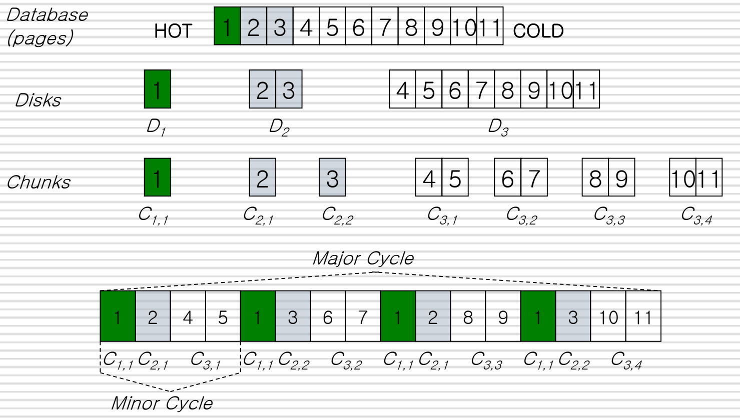
1. Order the page from hottest (most popular) to coldest
2. Partition the list of pages into multiple ranges, where each range contains pages with similar access probabilities. These ranges are referred to as disks
3. Choose the relative frequency of broadcast for each of the disks. The only restriction on the relative frequencies is that they must be integers.
4. Split each disk into a number of smaller units. These units are called chunks (Cij refers to the jth chunk in disk i). First, calculate max_chunks as the Least Common Multiple (LCM) of the relative frequencies. Then split each disk i into num_chunks(i) = max_chunks / rel_freq(i) chunks.
5. Create the broadcast program by interleaving the chunks of each disk in the following manner:
   1. for i := 0 to max_chunks – 1
   2. for j := 1 to num_disks
   3. Broadcast chunk Cj,(i mod num_chunks(j))

Example of Broadcast Disk Program Generation

- Assume a list of pages that has been partitioned into three disks
  - the pages in disk 1 are to be broadcast twice as frequently as those in disk 2, and four times as frequently as those in disk 3
    - rel_freq(1) = 4, rel_freq(2) = 2, and rel_freq(3) = 1
    - max_chunks = 4, num_chunks(1) = 1, num_chunks(2) = 2, and num_chunk(3) = 4



Example of Broadcast Disk Program



Parameters of Broadcast Disk Program

- the number of disks (*num_disks*)
  - Determine the number of different frequencies with which pages will be broadcast
- The number of pages per disk, and its relative frequency of broadcast (*rel_freq(i)*)
  - Determine the size of the broadcast, and hence the arrival rate for pages on each disk
    - Adding a page to a fast disk can significantly increase the delay for pages on the slower disks
    - Expect that fast disks will be configured to have many fewer pages than the slower disks
      - this model does not enforce this constraint
      - Possible to have arbitrarily fine distinctions in broadcasts such as disk that rotates 141 times for every 98 times a slower disk rotates
      - Result in a broadcast having a very long period (141*98 rotations of the fast disk)

Client Cache Management in Broadcast Disks Approach

- Tuning the performance of the broadcast is a zero-sum game:
  - Improving the broadcast for any one access probability distribution will hurt the performance of clients with different access distributions
  - Clients can't simply cache their hottest data as in traditional pull-based client-server systems (e.g., LRU).
    - In the push-based environment, this use of a cache can lead to a poor performance if the server's broadcast is poorly matched to the client's page access distribution
  - Broadcast pages are not all equidistant from the client !!!
- The server can tailor the broadcast program to the needs of a particular client
  - the client cache the hottest pages obtained from the broadcast disk.
  - Once the client has loaded the hottest pages into its cache, then server can place those pages on a slower spinning disk.
  - Frees up valuable space in the fastest spinning disks for additional pages

19

Client Cache Management in Broadcast Disks Approach

- Clients must use their cache to store those pages for which the local probability of access is significantly greater than the page's frequency of broadcast
  - For a page *P* accessed frequently only by client *C* and no other clients
    - a page P is likely to be broadcast on a slow disk
    - To avoid long waits for the page, client C caches P locally
  - For a page *Q* accessed frequently by most clients including *C*
    - Broadcast Q on a very fast disk, thus reducing the value of caching it

20

Client Cache Management in Broadcast Disks Approach

- Use a *cost-based* replacement algorithm that takes the frequency of broadcast into account
  - $\mathcal{PIX} = p/x$
    - where *p* is the probability of access and *x* is the frequency of broadcast
  - $\mathcal{PIX}$ ejects the cached page with the lowest value of *p/x*
    - e.g., For pages *a* and *b*, consider $p_a = 0.3$ and $x_a = 4$ vs. $p_b = 0.1$ and $x_b = 1$
- $\mathcal{PIX}$ is not a practical policy to implement because it requires:
  - perfect knowledge of access probabilities
  - comparison of $\mathcal{PIX}$ values for all cache-resident pages at page replacement time
- Use $\mathcal{LIX}$, an LRU-style policy, to approximate the performance of $\mathcal{PIX}$

21

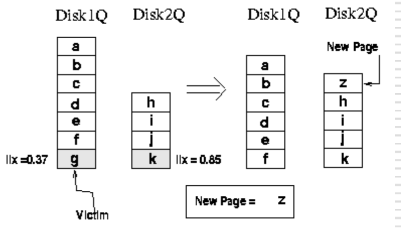
Client Cache Management in Broadcast Disks Approach

- LIX
  - LIX maintains a number of smaller chains: one corresponding to each disk of the broadcast
  - LIX reduces to LRU if the broadcast uses a single flat disk
- Algorithm
  - A page always enters the chain corresponding to the disk in which it is broadcast
  - Like LRU, when a page hit, it is moved to the top of its own chain
  - When a new page enters the cache, LIX evaluates *lix* value only for the page at the bottom of each chain
  - The page with the smallest lix value is ejected and the new page is inserted in the top of the appropriate chain
- The chains in LIX do not have fixed sizes

22

Client Cache Management in Broadcast Disks Approach

- An example of page replacement in *LIX*

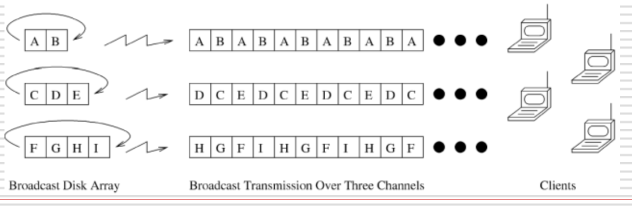


  - $Lix_i = p_i / x_i$ where $p_i = \lambda / (CurrentTime - t_i) + (1-\lambda)p_i$
    - x_i = *the frequency of the page i which is known exactly*
    - When the page *i* enters a chain, p_i is initially set to zero and t_i is set to the current time
    - If the page *i* is hit again, the new probability p_i is calculated and t_i is then subsequently updated to the current time
    - λ is a constant (e.g. 0.25)

23

Scheduling Data Broadcast Over Multiple Wireless Channels

- Why multiple wireless channels?
  - Application Scalability
  - Fault Tolerance
  - Reconfiguration of adjoining cells
- Multiple channel allocation problem
  - Allocate data to multiple channels to reduce the average expected delay of a request



24

Multiple Channel Allocation Problem

- K channels in a broadcast area, each denoted C_i, $1 \le i \le K$
- A database is made up of N unit-sized items, denoted by d_j, $1 \le j \le N$
  - Channel i broadcast N_i items, $1 \le i \le K$ where $\sum_{i=1}^{K} N_i = N$
- Each item d_i is assigned an access probability, p_i.
- Expected delay $w_i = \frac{N_j}{2}$
  - the expected number of ticks a client must wait for the broadcast of item i

Multiple Channel Allocation Problem

- Average Expected Delay (*AED*) for channel j

$$\sum_{i=1}^{N_j} w_i p_i \text{ where } w_i = \frac{N_j}{2}$$

  - the number of ticks a client must wait for an average request
- K Multichannel Average Expected Delay (*MCAED*)

$$\sum_{j=1}^{K} \left(\frac{N_j}{2} \sum_{d_i \in C_j} p_i \right) = \frac{1}{2} \sum_{j=1}^{K} \left(N_j \sum_{d_i \in C_j} p_i \right)$$

Multiple Channel Allocation Problem

- Goal: To allocate database items to K channels to minimize *MCAED*
  - *flat* design allocates an equal number of items to each channel
    - $N_i = \frac{N}{K}, \forall_i$
    - $MCAED = \frac{N}{2K}$
  - Focus on *skewed* design, where items are placed on varying sized channels, depending on their popularities

27

Example of MCAED

$$MCAED = \frac{1}{2}\sum_{j=1}^{K}\left(N_j \sum_{d_i \in C_j} p_i\right)$$

- a set of N = 6 items, {A, B, C, D, E, F} with the access probabilities such as:

TABLE 1
Access Probabilities

p_A	p_B	p_C	p_D	p_E	p_F
0.4	0.3	0.1	0.09	0.07	0.04

- Consider skewed design, allocating {A, B} to one channel and {C, D, E, F} to the other when K = 2 channels

TABLE 2
Example Comparison of Flat and Skewed Design $MCAED$s

flat design	$\frac{N_i}{2}\sum_{d_j \in C_i} p_j$
channel 1 ($i = 1$)	(1.5)(0.8)=1.2
channel 2 ($i = 2$)	(1.5)(0.2)=0.3
average expected delay	1.5

skewed design	$\frac{N_i}{2}\sum_{d_j \in C_i} p_j$
channel 1 ($i = 1$)	(1)(0.7)=0.7
channel 2 ($i = 2$)	(2)(0.3)=0.6
average expected delay	1.3

28

Allocation Algorithms for MCAED

- Optimal Allocation with DP Algorithm
  - *Single Channel AED* (*SCAED*) = C_{ij} = $\left(\frac{j-i+1}{2}\right)\sum_{q=i}^{j} p_q$ where $j \geq i,\ 1 \leq i, j \leq N$
  - $opt\_sol_{i,K}$ = the optimal solution (i.e., minimum *MCAED*) for allocating items from i through N on K channels
    - The optimal solution for items *i* to *N* given one channel is

      $opt\_sol_{i,1} = C_{i,N}$

$$opt\_sol_{i,K} = \min_{l \in \{i, i+1, \ldots, N-1\}} (C_{il} + opt\_sol_{l+1, K-1})$$

  - Although DP yields an optimal solution, its time and space complexity may preclude it from practical use.
    - Requires $O(KN^2)$ time and $O(KN)$ space

Allocation Algorithms for MCAED

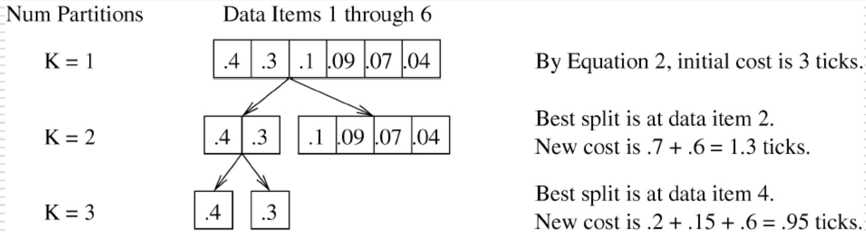
- *GREEDY approach*: $O((N+K) \log K)$

Algorithm 4.1
GREEDY
input: set of N unit sized items ordered by popularity, K partitions
begin
  $numPartitions := 1;$
  while $numPartitions < K$
    do Let $point_k$ be the split point that most reduces
        $SCAED$ for each partition k.
      Let $point'$ be the $point_k$ that most reduces $MCAED$.
      Create a split at $point'$.
      $numPartitions := numPartitions + 1;$
  od
end

Example of Greedy Algorithm

TABLE 1
Access Probabilities

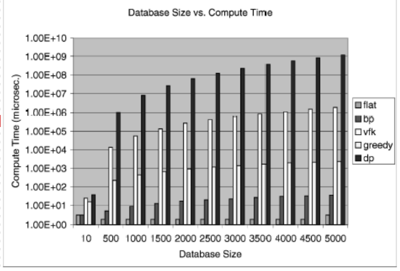
p_A	p_B	p_C	p_D	p_E	p_F
0.4	0.3	0.1	0.09	0.07	0.04



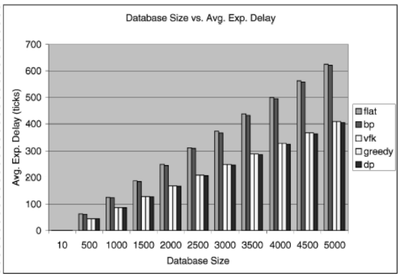
Performance Results

Algorithm	Denotation
FLAT [1]	flat
Bin-packing [14]	bp
VF^K [13]	vfk
GREEDY	greedy
Dynamic programming	dp

Parameter	Value Range	Control Value	Step
N - # of Items	1...5000	2500	500
Θ - Zipfian Skew, ($\Theta = 0$ for unif, 1 for high skew.)	0...1	(80/20 dist.)	.1
K - # Channels	1...2500	4	250

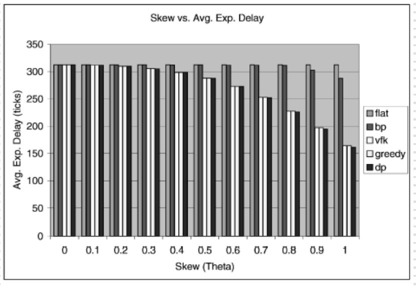


(a)

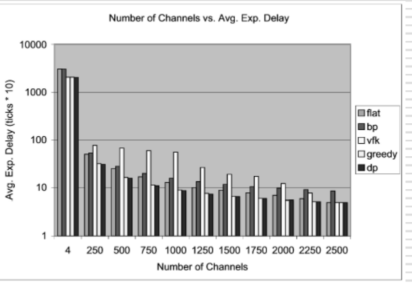


(b)

Performance Results



(a)



(b)

33