Broadcast Disks

Data Management for Asymmetric Communication Environments

Asymmetric Communication Environments (ACE)

- Downstream communication capacity is much greater than upstream.
- Reason:
  - Broadcast capability from server is much greater (mobile environment).
  - Information flow (remote classroom).
- Examples:
  - Wireless networks with stationary base stations.
  - Information dipersal systems for volatile, time sensitive information (stock prices).
  - Video-on-Demand service.
  - Information retrieval with large client population.

Broadcast Disks

- Pull based system is poor match for ACE.
  - Requires Substantial upstream communication capability.
- Broadcast disks are push based architechture.
  - Server continually and repeatedly broadcasts data to clients
  - Broadcast channel becomes a disk.
- New ideas:
  - Super impose multiple Disks (different capacity and speed) to create fined grained memory hierarchy.
  - Exploit client storage to extend memory hierarchy.

Scope

- Restricting ACE for study of broadcast disks.
  - Clients’ access patterns are static.
  - Data is read only.
  - No client prefetching.
  - No upstream communications.
- Paper’s Focus
  - Broadcast program generation.
  - Client cache management.

Structuring the Broadcast Disks

- Flat broadcast program
  - Union of all data items desired by clients.
  - Bounded latency for any data item retrieval from broadcast is constant.
- Problems:
  - Access probabilites are probably not uniform.
  - Desire to minimize Client population's Expected Delay.

Non-Flat Broadcast Programs

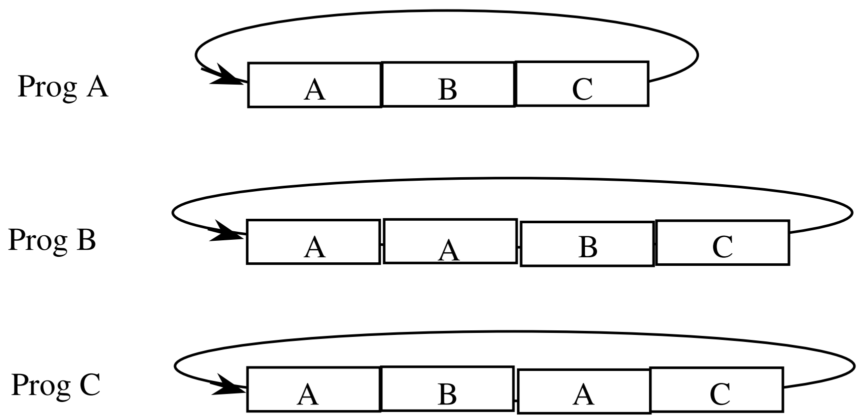
- Reduce problem to a bandwidth allocation problem to produce a non-flat broadcast program.
  - Calculate optimal percentage of broadcast bandwidth for each data item based on client population access probabilities.
  - May still not be optimal depending on variance of data item inter-arrival times.
    - Skewed broadcast programs will not perform as well as regular broadcast programs (see following example).

Expected Delay

- Client Population Expected Delay — CED
- Access Probability for Page "n" — APn
- Expected Delay for Page "n" — EDn

$$CED = \sum_{n=a,b,c} (APn * EDn)$$

Expected Delay Example



PAGE EXPECTED DELAY (In Broadcast Units)

	Page A	Page B	Page C
Prog A	1.5	1.5	1.5
Prog B	1.25	2.0	2.0
Prog C	1.0	2.0	2.0

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

Expected Delay Example (Continued)

- There are 3 major points demonstarated by this example:
  - For uniform page access probabilities, a flat disk has best performance.
    - Increasing broadcast rate of one data item decreases broadcast rate of others (continuous broadcast).
  - As page access probabilities become more skewed, the non-flat disks perform better.
  - Multi-disk program always performs better than the skewed program (Bus Stop Paradox).

Advantages Of Multi-Disk over Skewed

- Multi-Disk has better performance tha n Skewed.
- Other advantages of Multi-Disk over Skewed:
  - Fixed inter-arrival times for each data item benefit prefetching and power conservation techniques.
  - Periodicity may be required for providing correct semantics for updates.
- Broadcast program features base on findings
  - Inter-arrival times for each data item should be fixed
  - Broadcast programs should be periodic.
  - Broadcast progams should have the smallest dead time possible (continuous).

Broadcast Program Generation

- FIGURE 3
- Alogorithm:
  - (1) Order data items from highest access probability to lowest.
  - (2) Separate pages into disks by using a large gap in access probability as a division point.
  - (3) Choose a relative frequency(i) for every disk(i).
    - Try to make highest access probability disk the largest relative frequency. Disk with smallest relative frequency should be the lowest access probability disk. This might influence the segmentation of data into disks.
  - (4) Avg access prob for disk(i)/rel freq(i) =Const.
  - (5) Calculate LCM of all relative frequencies.
  - (6) Split each disk into (LCM/rel freq(i)) chunks.

Broadcast Program Generation (cont.)

- (7) Create broadcast program as follows:

```
for i:= 0 to LCM-1
    for j:=1 to num_disks
        Broadcast Chunk (i MOD (# of Chunks in Disk j) from
    Disk j
    endfor
endfor
```

Client Cache Management

- In push based systems cache plays a different role.
  - Caching hottest pages not optimal.
    - Broadcast program is not optimal for a single client.
  - Cache pages where local prob. of access is significantly greater than broadcast frequency.
- Cost based page replacement.
  - PIX = Prob. of access/Freq. of Broadcast
  - Page replacement when PIX of page causing cache miss is less than PIX of page in cache.
- PIX is not practical.
  - Requires access probability knowledge which is probably dynamic.

Client Cache Management (continued)

- LIX is a Least Recently Used (LRU)- style policy.
- Figure 12
- LIX Algorithm:
  - A queue is kept for each broad cast disk.
  - When a cache miss occurs:
    - The page with the smallest lix value is remove from its queue.
    - The cache space is reallocated for the new page on the appropriate queue.
- lix value:
  - Each queue entry contains
    - Running probability estimate - P_i.
    - Time of most recent access - T_i.
  - $P_i = W/(\text{CurrentTime} - T_i)+(1-W)P_i$
    - W is the appropriate wieght of the most recent access.

Client Execution Model

Server Execution Model

Experiments and Results

Previous Work

Summary and Future Work