Improving Data Circulation Performance in Mobile Databases

Wai Gen Yee
Department of Computer Science
Illinois Institute of Technology
yee@iit.edu
www.cs.iit.edu/~yee
Motorola, April 16, 2004
Vision

- Mobile database access that is robust to variations in network connectivity.
Mobile Databases in the Business Context

- Allows clients “anytime/anywhere” access to shared corporate data.
- Suitable for many business applications in CRM and ERP, e.g., salesforce automation, inventory management, etc.
- Heavily deployed in pharmaceuticals, financial planning, real estate...
ISDB State of Industry

- Supported by major vendors (Oracle OracleLite, Sybase SQLAnywhere, Microsoft, IBM).
- Many third parties, e.g., Synchrologic Imobile Suite, Puma Technology Intellisync.
Architectural Overview

- Wireless Base Station
- Fixed Network
- Docking Station
- Mobile Client
- Wireless Cells
- Fixed Host
- Mobile Client
- Mobile Client
- DB
Outstanding Issues with Mobile Databases

- Handling variable connectivity.
  - Managing client caches.
-Disconnected operation.
  - Logging updates made on cache.
- Synchronization during reconnection.
  - Scalable and consistent synchronization updates made while disconnected.
Today’s Topics

- Scalable synchronization of clients.
  - Efficient generation of update files by the server.
  - Scalably disseminating update files to clients.
Cycle of Data in Synchronization

1. The server generates an update file for each client.
2. Clients download update files when connected to the server.
Server Performance Issues

- Each of the $n$ client generates an average of $c$ updates.
- When generating update files, the server must decide which of the $cn$ updates to send to each of the $n$ client.
- Therefore, update server work is $O(n^2)$. 
Increasing Per-client Update Processing Time

Platform: Sybase SQLAnywhere v6, Synchrologic iMobile Suite v.2, Windows NT on Pentium PCs.
Proposed Data-centric Update Processing

- Solution – cluster updates into $k$ datagroups instead of $n$ subscriptions.
- Server generates update files for the $k$ datagroups.
- Each client downloads updates for a set of subscription-covering datagroups.

- Update server work is $O(n)$.
- Details in Yee01.
Design Example

Client-centric grouping

Data-centric grouping
Competing Canonical Datagroup Designs

- \texttt{cc} - client-centric, one datagroup per client.
- \texttt{dc} - data-centric, datagroups generated using greedy heuristic.
- \texttt{op} - one-per, one datagroup per publication.
- \texttt{og} - one-giant, one datagroup for all publications.
Examples of Canonical Designs

Client-centric (CC)

Data-centric (DC)

One-per (OP)

One-giant (OG)
Experiment: Vary Client Population

50 updates/client, client b/w is 57.6kbps, DB size is 100 pubs, sub rate is 1/client, update dist. is Zipfian.
Network Scalability Issues

- Using data-centric grouping, multiple clients may download the same data.
- Queuing theory suggests that single server unicast transmission is unscalable with client population.

Solution--broadcast (or push) scheduled updates to clients.
Canonical Broadcast Environment

- DB
- Server
- Data
- Clients
Broadcast Pros and Cons.

- Broadcasting is scalable – performance does not degrade with increasing client population.
- Eliminates need for clients to transmit requests for data.
- However, broadcast has high latency – clients must wait for desired data to arrive.
The schedule must:

- **be cyclic**, allowing data management optimizations, e.g., consistency control (Shanmugasundaram+99), client cache management (Acharya+95), indexing (Imielinski+94).


- **be simple to compute**, allowing large data sets to be scheduled.
Broadcast Scheduling Problem

- Given
  - database consists of N unit-sized items,
  - client requests over the N items are exponentially distributed,
  - the server knows the popularity $p_i$ of each data item $d_i$.
- How should data be scheduled for broadcast in order to satisfy the three criteria?
Solution: Schedule by Interleaving Data Partitions

- Schedule is clearly cyclic.
Bandwidth Efficiency

- Measured by average expected delay (aed).

- Let $N_i$ be the size of partition $C_i$, $K$ be the number of partitions, and $p_j$ be the popularity of item $d_j$.

\[
aed = \frac{K}{2} \sum_{i=1}^{K} N_i \sum_{d_j \in C_i} p_j
\]
Greedy - A Simple Partitioning Technique

- Given: A number of partitions, $K$, and a set of items defined by $p_i$.
- Generate the first partition by ordering the items by $p_i$.
- Split the partition that can most reduce cost (aed).
- Perform split $K-1$ times so there are $K$ partitions.
- Details in Yee02.
Experimental Candidate Algorithms/Measures

- Safok - proposed algorithm.
- Flat - all data items are given equal bandwidth.
- Opt - analytical lower cost bound, from Ammar85.
Experiment-Varying Skew

DB size is 500 unit sized items. Cost is in ticks.
Summary

- Model for ISDB performance.
- Algorithms for server side design related to update processing.
- Scheduling strategies for scalably disseminating updates - bridges gap existing between work in network and database communities.
- Investigating indexing techniques for pushed data over single and multiple channels.
Conclusions

- Current mobile database applications inefficiently use resources at the server and the network. This creates ample scope for investigating performance enhancement techniques.
- By reducing redundancy in these areas, performance improvements are possible. This leads to better scalability of shared resources and energy conservation for client devices.
- Addresses vital need in industry for upcoming applications.
Other Work

- Broadcast indexing.
- Peer to peer systems.
- Information retrieval.
Thanks!

- Contact info:
  - yee@iit.edu
  - www.cs.iit.edu/~yee
References

  - Overview of ISDB concept.
  - Datagroup design using operators.
  - Datagroup design using spanning trees.
“Fast data access on multiple broadcast streams,” Yee, Navathe, ICDE, 2003
  - Study of indexing broadcast schedules.
“Bridging the Gap Between Response Time and Energy Efficiency in Broadcast Schedule Design,” Yee, Navathe, Omiecinski, Jermaine, EDBT, 2002
  - Single channel schedule design.
“Efficient data allocation over multiple channels at broadcast servers,” Yee, Navathe, Omiecinski, Jermaine, IEEETC, 2002
  - Multiple channel schedule design.
“Scalability issues in mobile database applications,” Navathe, Yee, HPCAsia, 2002
  - Overview of the ISDB architecture with an emphasis on data dissemination.


“RxW: A Scheduling Approach for Large-Scale On-Demand Data Broadcast,” Aksoy, Franklin, ACM TON, 1999

“Scheduling Data Broadcast in Asymmetric Communication Environments”, Vaidya, Hameed, MONET, 1999

“Broadcast Scheduling for Information Distribution,” Su, Tassiulos, Tsotras, MONET, 1999

“Minimizing Service and Operation Costs of Periodic Scheduling,” Bar-Noy, Patt-Shamir, Ziper, SODA, 1998


“Broadcast disks: Data management for asymmetric communication environments,” Acharya, Alonso, Franklin, Zdonik, SIGMOD, 1997

“Efficient Concurrency Control for Broadcast Environments,” Shanmugasundaram, Nithrakashyap, Sivasankaran, Ramamritham, SIGMOD, 1999