Prioritized Sequencing for Efficient Query on Broadcast Geographical Information in Mobile-Computing

Jianting Zhang, Le Gruenwald

School of Computer Science
The University of Oklahoma
Norman, Oklahoma, 73019, U.S.A.
Table of Contents

- Introduction
- Problem and Motivation
- R-Tree Based Prioritized Sequencing
- Experiments and Results
- Future Work
Introduction (1)

Geographical Information in Mobile Computing

• Mobile Network
  • InfoStations (http://www.winlab.rutgers.edu)
  • Information Hoarding (Kubach, 2001)
  • Global Spatial Infrastructure (Hohl, 1999)
  • Context-Aware Applications (Chen, 2000)
  • Pervasive Computing (Castro, 2001)
  • Ubiquitous Computing (Hightower, 2001)

• Wireless Mobile Network
  • Location Management in cellular network
  • Geo-Routing in ad-hoc & Sensor networks
  • Location-Dependent Services
Introduction(1)

A Generic Mobile MAN Structure

- Mobile Host
- Base Station
- Cell Boundary
- Location Area Boundary
- Ad-hoc Network

Sensor Network  Block/Building Scale Cellular Network
Introduction (2)

Why Broadcasting

- Key problems in Mobile Computing
  - Bandwidth
    - Independent of number of users
    - Excellent scalability
  - Power Consumption:
    - Listen/Sleep mode consumes less power than in send mode
  - Mobility
    - No mobility management is required at neither server side nor client side
Introduction (4)

Why GI Broadcasting

- Public Information
  - Service locations: ATM machines, Restaurants…
  - Traffic & Road Conditions…
  - Weather…
- Large number of potential users of GI (metropolitan area for example)
- Relatively static/low update frequency
- Mostly Read-only
- Privacy is not a big concern
- Distributed in nature
Introduction (5)
Location-Based Services Over Broadcast GI

Assumptions:
- Clients know their current locations
- Tune in to index segment(s) at the beginning of a broadcast cycle and store the index in client memory
- Spatial query against index to retrieve pointers the positions of real data items in the sequence
- Tune in to the data items consecutively in active mode and switch to sleep mode in between.
Sequential Access to Broadcast Channel

Tune-in Time - TT:
- Time for downloading data from broadcast sequence
- In active (listen) mode
Introduction (6)
Parameters in Broadcast System

- **Access Time (Latency) – AT:**
  - The duration between the time starting accessing broadcast channel to the time of all data are retrieved.
  - User may switch to sleep mode in between active downloading.
Introduction (7)
Parameters in Broadcast System

Components in Access Time

- **Probe wait**: from start time to the time that an index segment is encountered
- **Bcast wait**: from the time an index segment is encountered to all the data items are downloaded
Problem (1)

- Previous Indexing Methods
  - Hashing (Imielinski, 1994a)
  - Signature Method (Lee, 1996)
  - Tree-based: (Imielinski, 1994b)
    - (1,m) indexing
    - Distributed indexing
  - Tree+Link List: (Imielinski, 1997)
    - Nonclustering
    - Multi-attribute Indexing
Problem (2)

- Geographic Information
  - Multidimensional
  - Geometric values are Continuous

- Using 1-D indexing consecutively for m-D indexing has poor performance (Gaede, 1998).

- Why not using spatial indexing methods?
Motivation

BD-INT/BD-PRI

\[ [1,2,3,4,5,6,7,8,9,10,11] \]
\[ \Rightarrow \text{bcast wait}=3 \]

\[ [3,5,6,8,1,2,4,7,9,10,11] \]
\[ \Rightarrow \text{bcast wait}=1 \]

\[ [1,2,3,4,5,6,7,8,9,10,11] \]
\[ \Rightarrow \text{bcast wait}=8 \]

\[ [3,5,6,8,1,2,4,7,9,10,11] \]
\[ \Rightarrow \text{bcast wait}=4 \]
Prioritized Sequencing (1)

Method

1. Generate a hot data item set
2. Build a R-Tree for the whole data item set and a hash table for the IDs of hot data items.
3. Travel the leaves in the R-Tree. For each leaf node:
   - Lookup the ID of the item in the hash table for the hot data item set.
   - If found then put the item ID in the vector of hot items.
   - Else put the item ID in the vector of cold items.
4. Output the contents of items in the hot data vector followed by the ones in the cold data vector.
Prioritized Sequencing (2)

Performance Analysis

- Compare to BD-INT
  - Tune-in Time: No
  - Probe Wait: No
  - Bcast Wait: YES!

- Average bcast wait is reduced from \((L_{\text{ind}} + L_w)/2\) to \((L_{\text{ind}} + L_h)/2\) if all retrieved items are hot
  - \(L_w\): the total length of the whole data items
  - \(L_h\): the total length of the hot data items

- Possible increase bcast waits of cold items
- Hopefully the average bcast waits are reduced
Prioritized Sequencing (3)

Performance Analysis

- If only one data item in each query result, prioritized sequencing reduces total bcast wait time

\[ T = F_c \cdot L_1 + F_h \cdot L_2 + T_{oth} \]
\[ T' = F_h \cdot L_1 + F_c \cdot L_2 + T_{oth} \]
\[ T' - T = (F_h - F_c) \cdot (L_1 - L_2) \]
\[ F_h > F_c, L_1 < L_2 \Rightarrow T' < T \]
Prioritized Sequencing (4)

Performance Analysis

- If Multiple data items in a range query result, which could include both hot and cold data items: no theoretical guarantee of bcast reduction

- Prioritized sequencing as a heuristic:
  - Hot data items are clustered and the neighbors of hot data items are also hot
  - Hot data items are accessed more frequently
  - A query result is more likely to contain higher percentage of hot data items
  - The overall bcast wait time could be reduced
Prioritized Sequencing (5)

Performance Analysis

- Alternative Measurement of Performance (Lee, 2000)
  - Data affinity index (DAI): $A_i = (1 - N_i/N)$
  - Aggregate data affinity (ADA): $\Sigma A_i$
  - Measure the “eagerness” for data experienced by a client

- ADA (A) has a linear relationship with broadcast wait time (T): $A = T/N$
Prioritized Sequencing (6)

Performance Analysis

\[
A = \sum_{i=0}^{N-1} A_i = \sum_{i=0}^{N-1} \left(1 - \frac{i}{N}\right)(t_{i+1} - t_i) \\
= \sum_{i=0}^{N-1} (t_{i+1} - t_i) - \sum_{i=0}^{N-1} \frac{i}{N}(t_{i+1} - t_i) \\
= (t_N - t_0) - \frac{1}{N} \left[1^* (t_2 - t_1) + 2^* (t_3 - t_2) + \ldots + (N - 1)^* (t_N - t_{N-1})\right] \\
= (t_N - t_0) - \frac{1}{N} \left(-t_1 - t_2 - t_3 \ldots + N^* t_N\right) \\
= \left(\frac{1}{N} \sum t_i\right) - t_0 \\
= \frac{T}{N}
\]
Experiment and Results (1)

Buffer Radius: 0.5 mile
Total Points: 586
# of Hot data items: 111
# of Cold data items: 475
Experiment and Results (3)

- Exact Query
  - For each test round, randomly select 10 hot items and 5 cold data items
  - Perform 10 rounds
  - Improvement:
    - Vary from 49% to 64%
    - Average 57.65%
Experiment and Results (3)

- Range Query
  - For each test round, randomly select 10 hot items and 5 cold data items
    - Use their locations as the centers
    - Use radius between (0.2,0.5) miles for hot data items
    - Use radius between (0.5,0.1) miles for hot cold items
  - Perform 10 rounds
  - Average reduction latency: 56.86%
Summary

- Propose to broadcast geographic information in mobile computing
- Use R-Tree spatial indexing method for GI broadcasting
- Present a simple and effective prioritized sequencing method that reduce bcast wait 50-60% on average
Future work (1)

- More heuristics in sequencing
  - Other index trees
  - Space-filling curves (Z-order, Hilbert)
  - Search orders (BFS/DFS/MST, etc.) of interconnection graphs (e.g. voronio diagram)

- Sequencing Optimizations
  - The # of possible orderings: n!
  - Low-cost approximation algorithms
Future work (2)

- A formal **Cost Model** for spatial query over broadcast geographical information
- **Extensions**
  - Moving objects
  - Network data
- More experiments using more data sets
Thanks!

Questions?