Proactive Caching for Spatial Queries in Mobile Environments & Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments

Jeppe R. Thomsen

Hong Kong Polytechnic University
Department of Computing

November 10, 2010

Overview

Introduction

Published in IEEE Transactions on Computers

by

Baihua Zheng, Jianliang Xu, and Kik L. Lee

October 2002

Motivation

Doing location based queries from a mobile phone, with a cache, the user could:

- Save Money
- Reduce Network Traffic
- Conserve battery

Existing cache replacement policies too simple. Does not consider valid scopes of cached items

Related Work

Temporal-dependent invalidation

- server sends *invalidation reports* to clients

Location-dependent invalidation

- Depending on users location or area after movement, data might be invalidated.

Semantic data caching

- cached items saved with locations associated with query

Problem

Mobile, cache enabled, clients communicate with fixed hosts, requesting location based service from fixed hosts.
A Geometric model is used, with identifying their location via e.g. GPS.

- Mobile clients
- Fixed hosts
- Geometric model

- Perceived Problem

Given a cache enabled client and a LBS server, which cache replacement technique performs best, and does data distribution have an effect.

Invalidation Strategies

Location-Dependent Invalidation Strategies

- PE Polygonal Endpoints
- AC Approximate Circle
- CEB Cache Efficientcy Based

Methods to support invalidation of cache items based on the valid scope of each items

CEB - Cache Efficientcy Based

$$E(v_i') = \frac{A(v_i')/A(v)}{(D+O(v_i'))/D} = \frac{A(v_i')D}{A(v)(D+O(v_i'))}$$

- v_i' - subregion of v
- $A(v_i')$ - area of v_i'
- $O(v_i')$ - overhead to record scope of v_i'
- D - Data size
- $A(v_i')/A(v)$ - Cache hit ratio (assuming uniform probability of queries from all locations)
- $(D + O(v_i'))/D$ - Cost ratio to archive hit ratio
- $E(v_i')$ - Caching efficiency of data item with respect to scope of v_i'

Cache Replacement Policies

Eject cache items furthest away from user

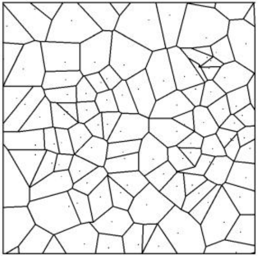
Favor cache items with larger valid scope

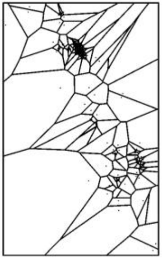
PA Probability Area $c_{i,j} = P_i * A(v'_{i,j})$

PAID Probability Area Inverse Distance $c_{i,j} = \frac{P_i * A(v'_{i,j})}{D(v'_{i,j})}$

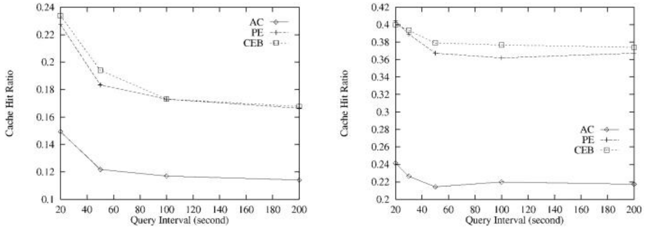
- $c_{i,j}$ - Cost function of data value j of cache item i
- P_i - Access probability of cache item
- $A(v'_{i,j})$ - Area of attached valid scope $v'_{i,j}$
- $D(v'_i)$ - Distance between user and valid scope $A(v'_{i,j})$

Setting


(a)

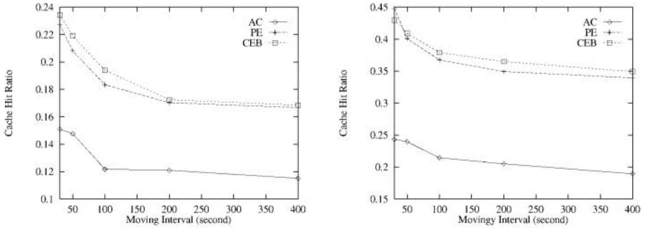

(b)

- 2 datasets, modeled as Voronoi Diagrams (110/185 Points)



(a) (b)

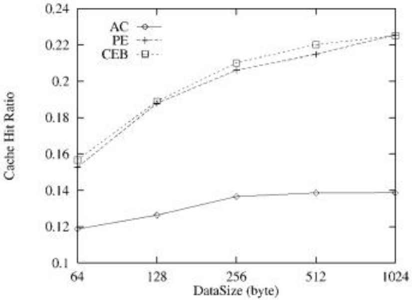
Cache hit ratio of invalidation schemes vs. query interval. (a) Scope distribution 1. (b) Scope distribution 2.

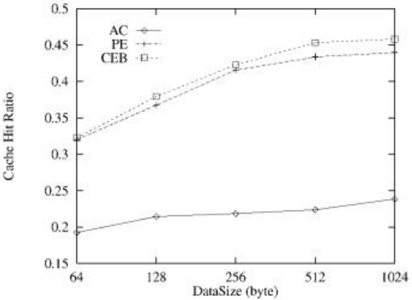


(a) (b)

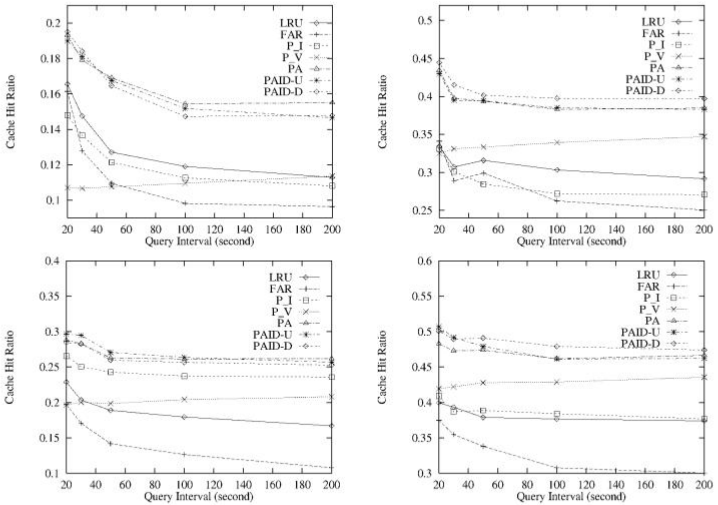
Cache hit ratio of invalidation schemes vs. moving interval. (a) Scope distribution 1. (b) Scope distribution 2.

Cache hit ratio / Data Size - scope 1 and 2

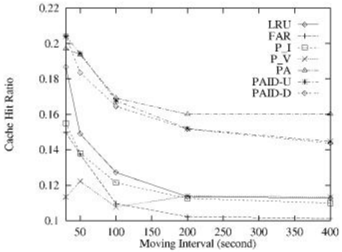


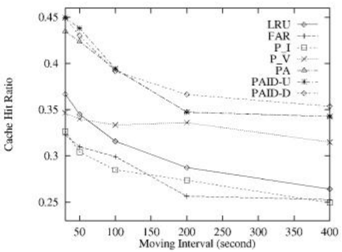


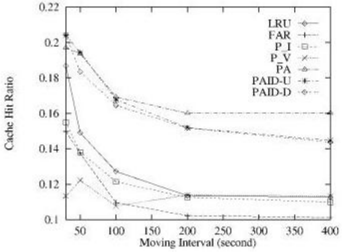
Cache hit ratio / Query interval - scope 1 and 2

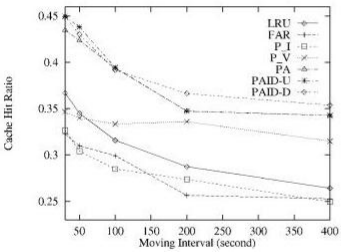


Cache hit ratio / Moving interval - scope 1 and 2









Conclusion

Introduce

- CEB
- PA
- PAID
- Data Distance (PAID)

Show experimentally that using PA and PAID will give some improvement

Impression

Paper too long, with very little content

Experimental section almost 2/3 of paper.

Paper does not state problem to be solved, is a rather just a description of methods

Contribution: experimentally showing that obvious improvements to existing ideas works okay.

Introduction

Published at: ICDE '05: Proceedings of the 21st International Conference on Data Engineering

by

Haibo Hu, Jianliang Xu, Wing Sing Wong, Baihua Zheng, Dik Lun Lee, and Wang-Chien Lee

April 2005

Motivation

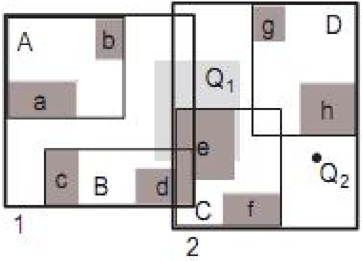
Related Work

Caching:

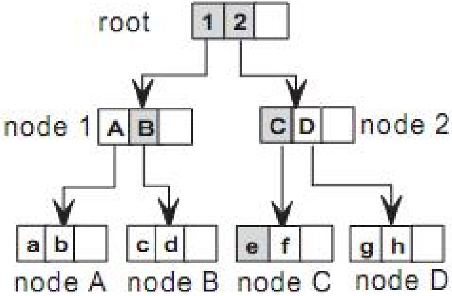
- Page Caching
- Semantic Caching

Problem

R-Tree



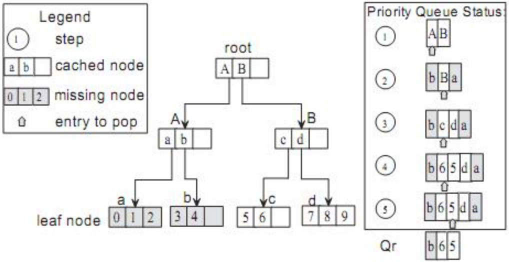
(a) Objects Placement



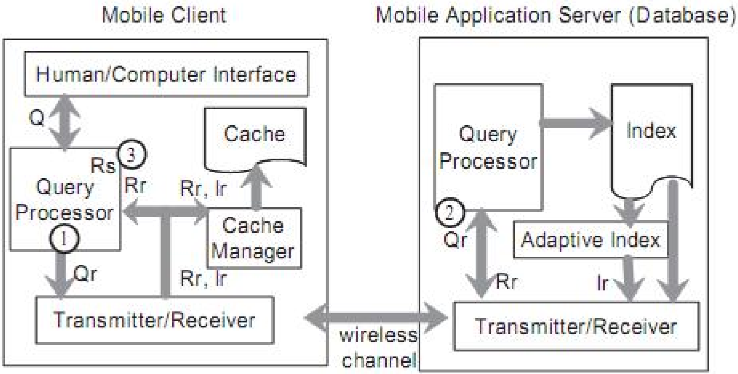
(b) Corresponding R-tree

Proactive Caching

1. Execute query Q on local partial R-tree index and cache
2. If any items are found while executing Q locally, then return immidiately
3. If Q is satisfied then terminate, else Construct $Q_r = Q + H$ and send to server



Proactive Caching



Response time and Hit rate

Query responsetime:
$resp(Q) = \frac{|R_r|(TQ_r + \frac{1}{2}|R_r| \cdot T_d}{|R|}$

Cache Hit rate:
$hit_c = \frac{|R_s|}{R}$

Algorithm is a 2-approximation algorithm

When cached node is removed, the children of that item should be considered into the benifit calculations.

$\sum_{j \in D(i)} prob(j) \times size(i) + prob(i) \times size(i)$

Conclusion

Proactive Caching outperforms page caching and semantic caching

Impression

Their results were not stellar

Graphs are "‘cut"’ just before compeditors might seem to become better

Concepts to be presented

- Addition of polynomials
- Multiplication of polynomials
- Coefficient Representation
- Point-value Representation