Fall 2010

Jeppe Rishede Thomsen
Department of Computing
Hong Kong Polytechnic University

Abstract

All the papers either solve the problem of a more efficient cache in a specific domain, or use the network domain, which are both relevant, but not really useful when looking at shortest path caching.

The papers show some interesting ways to use cache, but ultimately their approaches are very domain or query specific so their approaces to caching and cache replacement/invalidation can not be applied directly.

1. Introduction

1.1. Prepare for meeting

formulate problem why cache?

- less computational load
- faster response time
- better use of bandwidth 1
-

Describe strait forward solution

just store query results in cache and only cosider direct cache hit with a simpel cache policy like LRU or even FIFO

Come up with some simple solutions

prepare:

- The definition and problem setting of shortest path caching
- Some simple methods
- Running examples to shw how these methods work
- Identify advantages / disadvantages of these methods

1.2. Problem

1.2.1. Definitions and problem setting. We assume a setting where owners of mobile, positioning enabled, devices want route planning assistance. We assume users prefer online route planning services over offline solutions.

1.2.2. methods. We assume a setting where all users are equipped with a Mobile Device (MD) able to communicate and report the users position. All MDâĂŹs are online and are continuesly reporting the users location at predeïňĄned intervals. We use the terms user, mobile device, and client interchangeable and denote the set of MDâĂŹs by U âŁĆ N. We expect a MD to be cabable of visualizing its current location. We assume a 2D scenario, where the movements of users u âĹĹ U are restricted to a road network G(V, E). V is the set of vertices, where each vertice v âĹĹ V represents either a street intersection or an important landmark. E is the set of directed edges augmented by edge length and type. Edges are represented by a begin/end vertice pair and each edge represents the smallest unit of a road segment. e âĹĹ E, each e being a tuple specifying id, start-/end-vertices, length, and Road Type (RT) (eid , vs , ve , elength , eRT). RT is a hierarchy of the size/type of road i.e. highway, paved, or dirt road (Sec. 5.2). The simplest form of trajectory is a collection of tu- ples (time, longitude, latitude), ordered by the time attribute, but as we will work on a road network and in the spatio-temporal domain, such a basic notion of trajectories is not appropriate. We deïňĄne T as the set of trajectories, where each trajectory consist of an id (tid), and a sequence of tuples containing an edge and

2. Problem setting

We introduce relevant notations, define system requirements and specify the behavior of the proposed service.

First we briefly summarize the general scenario again. We assume users are willing to upload the positioning data from the routes (trajectories) they take when they move around on a road network. The system then anonymizes the trajectories according to a user specified "Privacy Profile" (Sec. **??**), specifying which places or routes are sensitive to the user (Fig. **??**A, 1-5). After anonymization the system makes the data

Symbol	Meaning
MD	Mobile Device
$\mathbf{U}$	Set of MDs
$\mathbf{V}$	Set of vertices
$\mathbf{E}$	Set of edges
$G(\mathbf{V}, \mathbf{E})$	Road network
$\mathbf{P}$	Collection of all PSRs
$\mathbf{T}$	Set of trajectories
Γ	Set of all trajectories which contains sensitive PSR p
RT	Road Type
AS	Protection scheme: Always Sensitive
ASTI	Protection scheme: Always Sensitive with time interval
RS	Protection scheme: Rarely sensitive

Table 1. Table of symbols and notation

available to potential service providers for whom it guaranties a minimum level of data quality.

2.1. Problem Definition

We assume a setting where all users are equipped with a Mobile Device (MD) able to communicate and report the users position. All MD's are online and are continuesly reporting the users location at predefined intervals. We use the terms *user, mobile device, and client* interchangeable and denote the set of MD's by $\mathbf{U} \subset \mathbb{N}$. We expect a MD to be cabable of visualizing its current location.

We assume a 2D scenario, where the movements of users $u \in \mathbf{U}$ are restricted to a road network $G(\mathbf{V}, \mathbf{E})$. $\mathbf{V}$ is the set of vertices, where each vertice $v \in \mathbf{V}$ represents either a street intersection or an important landmark. $\mathbf{E}$ is the set of directed edges augmented by edge length and type. Edges are represented by a begin/end vertice pair and each edge represents the smallest unit of a road segment. $e \in \mathbf{E}$, each e being a tuple specifying id, start-/end-vertices, length, and Road Type (RT) $(e_{id}, v_s, v_e, e_{length}, e_{RT})$. RT is a hierarchy of the size/type of road i.e. highway, paved, or dirt road (Sec. **??**).

The simplest form of trajectory is a collection of tuples $(time, longitude, latitude)$, ordered by the time attribute, but as we will work on a road network and in the spatio-temporal domain, such a basic notion of trajectories is not appropriate. We define $\mathbf{T}$ as the set of trajectories, where each trajectory consist of an id (t_{id}), and a sequence of tuples containing an edge and start-/end-time (τ_{s_i}/τ_{e_i}) of edge traversal. A trajectory is then $(t_{id}, \langle (e_j, \tau_{s_j}, \tau_{e_j}), \ldots, (e_k, \tau_{s_k}, \tau_{e_k}) \rangle)$, where $t_{id} \in \mathbb{N}$, $\tau_{s_i}, \tau_{e_i} | \tau_{s_i} < \tau_{e_i} \wedge \tau_{s_{i-1}} < \tau_{s_i} \wedge \tau_{e_{i-1}} < \tau_{e_i}, \tau_{s_i} \in \mathbb{N}, \tau_{e_i} \in \mathbb{N}$, and $e \in \mathbf{E}$, the set of edges. We restrict trajectories to the road network, and users are assumed to traverse the entirety of an edge.

A user $u \in \mathbf{U}$ is defined by a tuple $(u_{id}, \{s\}, \{t\})$, $u_{id} \in \mathbb{N}$, $s \in \mathbf{S}$ the set of Privacy Profiles (Sec. **??**), and $t \in \mathbf{T}$.

Traditional ways of obtaining user privacy often include spacial obfuscation by including the user in a bounding box, or cloaking region [**?**], [**?**]. Conventional classification methods working on trajectories [**?**], [**?**], [**?**] are not suited to work on cloaking regions, and applying such anonymization techniques will break these algorithms or require them to be modified to still function correctly.

Our approach has identical input and output format and keep the format simple by just having a list of trajectory ids, each one associated with a list of edge ids and their start-/end traversal time. We will thus not require any modifications of the traditional approaches for them to work on the anonymized dataset. The input format chosen only require a map to have ids on road edges, a reasonable assumption of most map data. After anonymization road edges may have been removed, and timestamps may have been modified, the output format is however still identical to the input format.

$\{\forall u \in \mathbf{U}\}$ specify at least one privacy profile. Each privacy profile specify the privacy settings for each section of road, such that for an edge $e \in \mathbf{E}$ traversed by the user, the system will anonymize e according to the privacy profile of u. Users need only specify one privacy profile, but may have more to cover different user contexts, e.g. work week, weekend, vacation.

We want to develop a solution for the described approach, accomplishing the following important goals:

- **Usability** To be useful, the privacy requirements has to be simple for the user to define.
- **Practical** Besides when user specify his privacy profile, it should not require user interaction while running correctly.
- **Flexible** Handle users specifying specific sensitive locations as well as specifying "everything but this" is sensitive.

Given a set of user $u \in \mathbf{U}$ with a set of privacy profiles $s \in \mathbf{S}$ and a set of trajectories $t \in \mathbf{T}$ on a road network $G(\mathbf{V}, \mathbf{E})$ the problem is then for each u to apply s on $\{t\}$.

3. Related work reference

reference support for related work section.

3.0.1. On effective presentation of graph patterns: a structural representative approach. They develop an approach that combine two focuses when mining patterns in graphs. 1. they introduce a method to relax the tightness of the pattern subgraph pattern matching, so they can have high support for subgraphs which are very similar, but not exact. 2. as many mining approaches return allot (often very similar) patterns, they propose a method to collapse similar patterns so the user is presented with something that is easier to get an overview of and gain an understanding of the data. [1]

3.1. Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments

They develop two cache replacement and invalidation techniques for mobile clients communicating with a LBS. They argue that in the setting of spatial data and LBS then it is important to consider more than just the access time when doing cache replacement. They look at the spatial area where an object in the cache is valid as well as the direction the user is moving. They do this besides calculating the probability that this object will be accessed again.

Assumes all POI objects are fixed size and no updates will be made. [2]

3.2. Nearest-Neighbor Caching for Content-Match Applications

[3]

3.3. Caching Content-based Queries for Robust and Efficient Image Retrieval

They study how to do caching with Content-based Image Retrieval, and they support range and kNN queries. They focus on how to do caching when many of the queries are similar, but not the same (e.g. picture cropped or color changes) without polluting the cache. Their approach works in metric space and they develop an approximate method to check if the result can be satisfied by the cache. They archive good results, getting few direct cache hits, but still satisfying many queries from similar queries in the cache.

[4]

3.4. Caching Complementary Space for Location-Based Services

They develop the notion of Complementary Space(CS) to help better use a cache on a mobile client. CS is different levels for representing the objects on a map within MBRs. At the lowest level they just show the object, and as the levels go up they include more and more objects within MBRs, looking at the trade of in communication up/down link from a mobile client. They always have the entire world represented within the clients cache, at different levels, and offer no solution to how they will handle server updates to the map.

This is very similar to [5], although the approach does not formally dependend on an R-tree, they still use one and offer no viable alternative, which lessens the difference even more. Their results are better than their competitors, including [5], though it seems that they stop their graphs just before [5] beats them.

3.5. Proactive Caching for Spatial Queries in Mobile Environments

They develop an approach which uses the index of an R-tree to add context to a cache of spatial object on a mobile client. They develop several communication and space saving techniques by representing less important parts of the R-tree in more compact ways, or just not storing the lower nodes/leaves of the tree. They also formally prove the assymtotic bounds of their algorithms.

[5]

3.6. Cache-Oblivious Data Structures and Algorithms for Undirected Breadth-First Search and Shortest Paths

[6]

3.7. Cached Shortest-Path Tree: An Approach to Reduce the Influence of Intra-Domain Routing Instability

They assume a network setting and try to reduce the time and computational load it takes when network topology changes, as well as prevent any links from being unreachable if the topology changes often. The propose a cache with shortest-path trees, arguing that even if the topology changes often, then it is mostly between the same configurations (e.g. a computer/router is turned off/on) meaning that a cache with the most common seen configurations will be able to drastically reduce the amount of computation needed to recalculate routing tables.

[7]

3.8. On Designing a Shortest-Path-Based Cache Replacement in a Transcoding Proxy

[8]

3.9. Optimizing Graph Algorithms for Improved Cache Performance

[9]

References

[1] C. Chen, C. X. Lin, X. Yan, and J. Han, "On effective presentation of graph patterns: a structural representative approach," in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*. New York, NY, USA: ACM, 2008, pp. 299–308.

[2] B. Zheng, J. Xu, and D. L. Lee, "Cache invalidation and replacement strategies for location-dependent data in mobile environments," *IEEE Trans. Comput.*, vol. 51, no. 10, pp. 1141–1153, 2002.

[3] S. Pandey, A. Broder, F. Chierichetti, V. Josifovski, R. Kumar, and S. Vassilvitskii, "Nearest-neighbor caching for content-match applications," in *WWW '09: Proceedings of the 18th international conference on World wide web*. New York, NY, USA: ACM, 2009, pp. 441–450.

[4] F. Falchi, C. Lucchese, S. Orlando, R. Perego, and F. Rabitti, "Caching content-based queries for robust and efficient image retrieval," in *EDBT '09: Proceedings of the 12th International Conference on Extending Database Technology*. New York, NY, USA: ACM, 2009, pp. 780–790.

[5] H. Hu, J. Xu, W. S. Wong, B. Zheng, D. L. Lee, and W.-C. Lee, "Proactive caching for spatial queries in mobile environments," in *ICDE '05: Proceedings of the 21st International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 403–414.

[6] G. S. Brodal, R. Fagerberg, U. Meyer, and N. Zeh, "Cache-oblivious data structures and algorithms for undirected breadth-first search and shortest paths," in *Algorithm Theory - SWAT 2004*, ser. Lecture Notes in Computer Science, T. Hagerup and J. Katajainen, Eds. Springer Berlin / Heidelberg, 2004, vol. 3111, pp. 480–492.

[7] S. ZHANG, K. IIDA, and S. YAMAGUCHI, "Cached shortest-path tree : An approach to reduce the influence of intra-domain routing instability," *IEICE transactions on communications*, vol. 86, no. 12, pp. 3590–3599, 2003-12-01. [Online]. Available: http://ci.nii.ac.jp/naid/110003221599/en/

[8] H.-P. Hung and M.-S. Chen, "On designing a shortest-path-based cache replacement in a transcoding proxy," *Multimedia Systems*, vol. 15, pp. 49–62, 2009.

[9] J.-S. Park, M. Penner, and V. K. Prasanna, "Optimizing graph algorithms for improved cache performance," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 9, pp. 769–782, 2004.