

Location-based Application based on Prefetching on Mobile Applications: Case Study of Finding Interesting Restaurants

K. Noipithuk
Department of Computing
Faculty of Science
Silpakorn University
Nakorn Pathom, Thailand

C. Kitkobchai
Department of Computing
Faculty of Science
Silpakorn University
Nakorn Pathom, Thailand

C. Chantrapornchai
Department of Computing
Faculty of Science
Silpakorn University
Nakorn Pathom, Thailand

ABSTRACT

In this work, we are interested in developing a mobile application which locates interesting points for a user. The interesting points are displayed in a map based on Google Map. We particularly consider the prediction of future points based on the user location, speed and direction. The data prefetching buffer is being used to store the future data. The prefetching engine is integrated in the mobile client.

General Terms

Data Prefetching, Mobile Application, Location-based Application,

Keywords

Data prefetching; mobile application, future location prediction.

1. INTRODUCTION

With the growth of mobile technology, location-based oriented applications become popular [1]. Users make use of location information very often, to search for interesting places, share information about visited places, etc. As the speed of Internet grows, these information sharing and searching are done using the map services. However, with the limitation of memory and display on mobile devices, the data buffering is a bottleneck. Limited information can be stored. Since the display size is small, the user may scroll through or enlarge the map to see more data. The information in the buffer should be available to be displayed along the scrolled/enlarge map.

In this paper, we are interested in developing a mobile application which presents the interesting places while integrating the prefetching data scheme. We present the use of data prediction based on user location, speed, and direction. The buffer is created to store the data prefetched. The approach is demonstrated with the restaurant finding application.

Many recent works present the prefetching in the GIS area. Some of the works are focused on mobile applications.

Weber (2008) studied the map browsers with a prefetching scheme on the mobile environment which has a limited memory and data transfer bandwidth. The approach divides the maps into tiles [2]. Each tile can be multi-layers for zooming. Details of locations may be requested on demands. Only portions of the maps are shown on the mobile screen called viewport. The map server communicates to the mobile

via a buffer. The buffer is treated like a cache where several replacement policies are investigated.

Yeslmurat (2010) presented the prefetching method for web application for GIS domains [3]. In the method, the map is divided into Hillbert Curve. Neighbourhood selection is done using a Markov Chain method. Predicted tiles are prefetched. Neighbourhood tiles are prefetched in levels.

Said , Omar and Robert (2009) proposed the prefetching algorithm for a mobile system called direct prefetching [4]. It considers the location of the user. The area is divided into squares and the probabilities are calculated for each square. The probability is created for each user profile and used as a guide to prefetch a square.

The work by Zhang, Cao, Tian and Chen (2008) focused on the mobile prefetching technique [5]. They considered the user usage pattern to predict the future data. The criteria also includes the penalty index which indicates the negative impact for bandwidth usage. The approach shows the effectiveness and the energy saving.

Wang, Pang and Luo (2010) presented the LBS-p platform for mobile map services [6]. The key of this architecture is the Byte-Map representation and modules to save time, power and space in transferring and searching data.

Fang, Jian and Fu (2008) proposed a prefetching approach which is based on Hierarchical Hilbert Grid [7]. Multiply layers are used to zoom in specific data area. Hilbert Curve is used to number the grid for prediction.

Im, Lee and Hwang (2006) used a semantic approach for data prefetching [8]. Semantic information are such as usage pattern, size of request, direction of the user. The semantic LRU cache replacement policy is proposed and investigated.

Lien and Wang also proposed a prefetching technique based on PPM. It captures the driver behavior for future prediction [9]. Markov prediction tree is used in consideration with the prefetching algorithm.

Chen (2005) proposed a web prefetching based on popularity prediction [10] [11]. The method is also based on partial matching (PBPPM). The approach works on proxy and web servers.

This paper is organized as following. Section 2 presents some backgrounds in about map concepts and location-based services. Section 3 presents the prefetching method used in the application. Section 4 is the application example. Finally, Section 5 is the conclusion.

2. BACKGROUNDS

Typically, a map is usually divided into grids. With the large map, each grid can have subgrid shown by layers as in Fig. 1. After the grids are defined, the grid is label in the lexicographical order. Given a particular location such as latitude and longitude, they are converted into the grid position. This conversion is quite subtle due to the mapping of the latitude and longitude to the screen grid. Once the grid number of known, current information about the tile can be retrieved from the database.

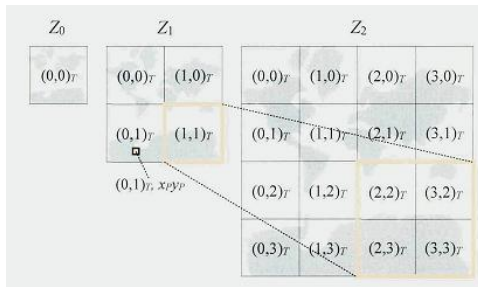


Fig. 1: Grid and subgrids of a map [2].

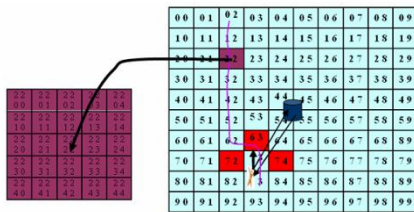


Fig. 2. Ordering of each grid [4].

To enable prediction of future information according to the current location, neighboring tiles can be considered as well as the direction. Figure 3 shows the current (dark color) and neighboring tiles (light color) in four directions. Along the traveling, the current tile is changed (from left to right). The tiles are aggregated in the last picture. Accordingly, the information in the current tiles and neighboring tiles are stored in the buffer. Figure 4 assumes the direction of the movement. Along the movement, some tiles are passed and new tiles are fetched. Thus, some information about passed tiles may not be important. They can be deleted from the prefetching. The replacement policy should be considered to manage the buffer. It is to the importance of each tile's information and to replace the one that is expected to be not used anymore.

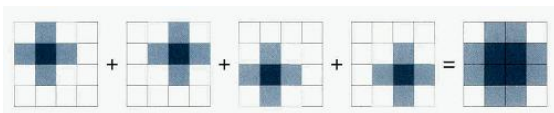


Fig. 3: Tile aggregation [4].

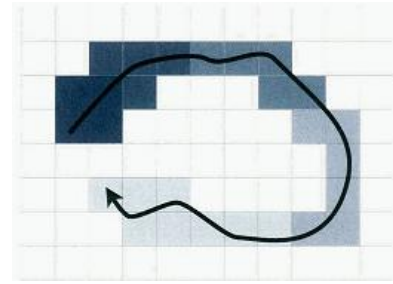


Fig. 4: Direction and significance of tiles [4].

On the mobile device, it is important to get the current location to obtain the information about the current location. Most mobile device currently integrates this feature. Typically, there are two kinds of location-based services: push service and pull service. In the push service, the information is pushed to the device irregardless of requesting it or not. In the pull service, the information request must be sent from the client side explicitly. Usually, the location-based service composes of 5 parts.

1. Mobile device
2. Communication network
3. Positioning component
4. Service and application provider
5. Data and content provider

To estimate the future position, the following formula is used.

The following is the explanation of the variables.

Table 1. Variables in the new location formula.

Variables	Description
<i>heading</i>	Direction of movement (in angles)
<i>speed</i>	Current velocity of the user (km/hr)
<i>lat0, lon0</i>	The current latitude and longitude of the user
<i>dtime</i>	The delay time used to send/receive data (seconds)
<i>PI</i>	3.1416
<i>Rt</i>	Earth radius =6378137.0 meters

The next location of the user from the current location is

$$y = speed \times \cos \left(heading \times \frac{PI}{180} \right) \times \frac{dtime}{3600}$$

$$x = speed \times \sin \left(heading \times \frac{PI}{180} \right) \times \frac{dtime}{3600}$$

Then from Equations (1)-(2), the latitude and longitude conversion are done as following.

$$newlon = lon0 + \frac{180}{PI} / \sin \left(lat0 \times \frac{PI}{180} \right) \times \frac{x}{Rt}$$

$$\frac{x}{Rt} newlat = lat0 + \frac{180}{PI} \times \frac{y}{Rt}$$

3. MOBILE PREFETCHING IMPLEMENTATION

To perform prefetching, we divide the our map area as into tiles in Figure 5.



Fig. 5: Tile map in Nakorn Pathom area.

For each current location (* in Figure 6), we are interested in 8 neighboring tiles.



Fig. 6: Current location and neighboring tiles.

The steps to get the information is as following.

1. The current location is retrieved as well as the direction. Equations (3)-(4) are used to calculate the future positions. Then these are mapped into the tile numbers.
2. The neighboring tiles of step 1 are calculated.
3. The current buffer is lookup to see whether the information about the tiles exist.
4. If they are not available, the client sends the request to the server to request the information about these tiles.

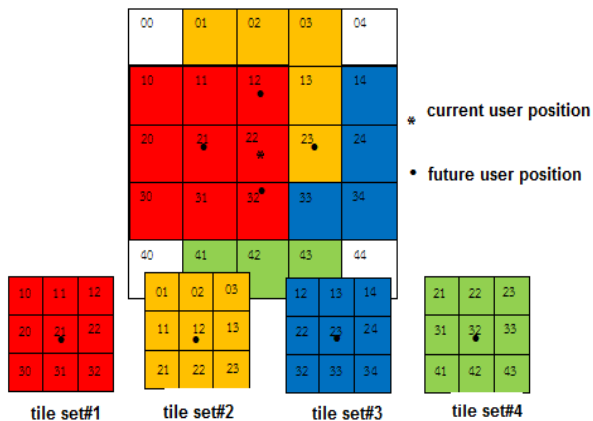


Fig. 7: Current user tiles and predicted tiles.

In Figure 7, assume that the current user position is marked by *. We aggregate the predicted positions in marked by • since we assume that it is equally likely that the user may change in any direction. Then, we collect the future four future positions as well as their neighboring tiles. However, this scheme may be changed to consider the preference of each predicted position or prioritize them. Then, these tiles are unioned and checked against the current buffer.

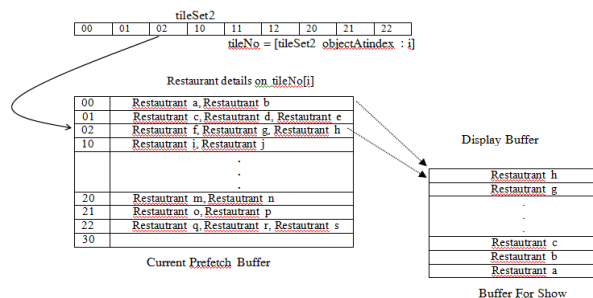


Fig. 8: Data structure to keep tiles and display.

Assume that we are interested to find the neighboring restaurant. In Figure 8, the data structure is shown. The aggregated tiles from Figure 7 are stored in an array. Then for each element, we use the tile number as an index to search in the current prefetch buffer. If it is found, the restaurant information are collected to show in the display buffer. If not, the request is sent to the server for such an information.

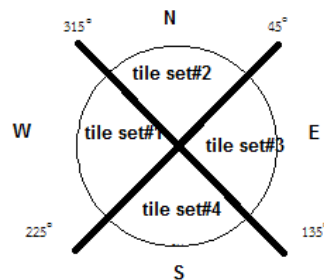


Fig. 9: Tile sets used to display.

In Figure 9, after the last step, the heading direction of the user is checked to obtain the tiles in the same user direction to display.

4. EXAMPLE APPLICATION

In the implementation, we are interested in finding restaurants nearby along the direction. The prefetching scheme is implemented in the client side to manage the buffer and request the future data. The prefetch buffer needs to be maintained with the replacement policy. It is also needed to make certain that only new future position data are requested and inserted to the buffer to minimize the network usage and maintain the buffer properly. The server side will only accept the queried locations sent by the client, look up the database and send the information back.

Figure 10 is the demonstration. The google map is used and the locations are marked in the map for the user to see the current information and the neighboring information. Also, some of the information may not be visible since the size of the mobile screen is small. These information is stored in the buffer. Once the user scrolls the screen, they are retrieved to be shown to the user. Place Tables/Figures/Images in text as

close to the reference as possible (see Figure 1). It may extend across both columns to a maximum width of 17.78 cm (7”).

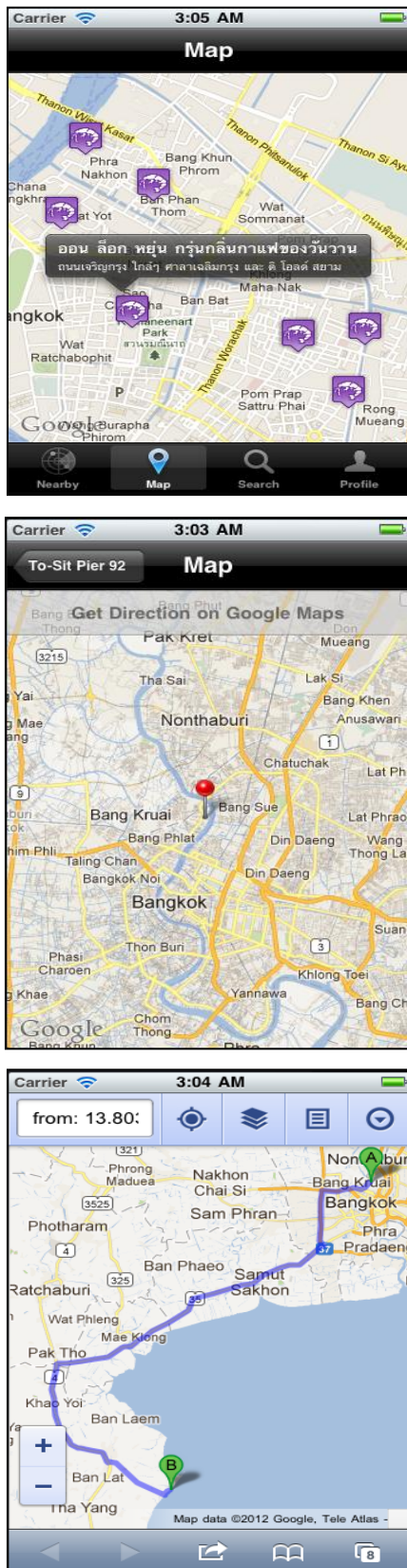


Fig. 10: Demo applications of restaurant finding with prefetching.

5. CONCLUSION

In this work, we present the location finding based on a prefetching scheme. We integrate the prefetching scheme into the location-based mobile application. The full methodology about the prefetching scheme starting from the map tiling is presented. The scheme considers the current user position, user direction and also predict the future user position. The information about future position is prefetched into the buffer. The mobile application about restaurant finding is used as a demonstration.

6. REFERENCES

- [1] U. Kubach and K. Rothermel, "Estimating the Benefit of Location-Awareness for Mobile Data Management Mechanisms," in Lecture Notes in Computer Science, 2002.
- [2] B. T. Weber, "Mobile map browser: Anticipated user interaction for data prefetching," 2008.
- [3] S. Yesilimurat T, "A Prefetching method for interactive web GIS," 2010.
- [4] E. G. Said, E. B. Omar and L. Robert, "Data Prefetching Algorithm in Mobile Environments," European Journal of Scientific Research, vol. 28, no. 3, pp. 478-491, 2009.
- [5] Y. Fang, C.-J. Jiang and Z.-H. Zhang, "A mobile navigation service platform based on traffic information grid," Journal International Journal of Services Operations and Informatics, vol. 1, pp. 23-37, 2006.
- [6] X. Wang, X. Pang and Y. Luo, "LBS-p: A LBS Platform Supporting Online Map Service," in Vehicular Technology Conference Fall, 2010.
- [7] Y. Fang, C. Jiang and Y. Fu, "Incremental Data Prefetching for Map Service in Mobile Navigation Application," in Wireless Communications, Networking and Mobile Computing, 2008.
- [8] S.-W. Kang, S. Im, J. Kim, S. Lee and C.-S. Hwang, "Considering a semantic prefetching scheme for cache management in location-based services," in The 10th international conference on Knowledge-Based Intelligent Information and Engineering Systems, 2006.
- [9] C.-C. Lien and C.-C. Wang, "An Effective Prefetching Technique for Location-Based Services with PPM," in 9th Joint Conference on Information Sciences (JCIS), 2006.
- [10] X. Chen, "Techniques of data prefetching, replication, and consistency in the internet," 2005.
- [11] X. Zhang, D. C. Cao, G. Tian and X. Chen, "Data Prefetching Driven by User Preference and Global Coordination for Mobile Environments," in The 3rd International Conference on Grid and Pervasive Computing - Workshops, 2008.