Volume 4, Issue 4, April 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Dispensation Location-Dependent Queries in Portable Environments

Rathika R
PG Scholar
Department of CSE
Chettinad College of Engineering & Technology, India

Padmaasree R
PG Scholar
Department of CSE
Chettinad College of Engineering & Technology, India

Deepti Raj G
Assistant Professor
Department of CSE
Chettinad College of Engineering & Technology, India

Abstract- Mobile devices are becoming a lot of pervasive, and it's turning into progressively necessary to integrate internet services into applications that run on these devices. They have a tendency to introduce a unique approach for dynamically invoking internet service ways from mobile devices with bottom user intervention that solely involves coming into an enquiry phrase and values for the strategy parameters. The design overcomes technical challenges that involve overwhelming discovered services dynamically by introducing a man-in-the middle (MIM) server that has an internet service whose responsibility is to get required services and build the client-side proxies at runtime. The design moves to the MIM server energy-consuming tasks that might otherwise run on the mobile device. Such tasks involve communication with servers over the net, XML-parsing files, and on-the-fly compilation of ASCII text file. It have a tendency to perform intensive evaluations of the system performance to live measurability because it relates to the capability of the MIM server in handling mobile consumer requests, and device battery power savings ensuing from scalability it relates to the capacity of empowerment by using mobility and the service has discovery tasks to the server.

Keywords —Nearest neighbor query, window query, spatial query processing, location-based service, mobile computing.

I. INTRODUCTION

Recent trends in mobile computing and the popularity of mobile devices has motivated interest in accessing web services from mobile devices in order to extend their functionality and gain access to remote data. However, the particularities and limitations of mobile devices and the mobile environment pose great challenges for consuming web services. To begin with, when using UDDI registries for service discovery, multiple costly network round trips are needed, and frequent unavailability of the wireless network may cause failures in the service discovery process, and will hinder the completion of the user request. Additionally, there are several issues and challenges that emerge from the fact that mobile devices have lower processing power, limited bandwidth, less memory, and finite battery power when compared to desktops. All the above suggest that architectures which target mobile devices should aim to minimize their interactions with the network and reduce their resource consuming processing whenever possible. Actually, the mentioned issues and challenges led to the architectural configuration proposed in this paper which introduces a server "in the middle" whose role is interfacing with web services in a way that is adapted to the capabilities and constraints of mobile devices. Most of the workload involved in the dynamic discovery of web services is passed to the server, thus relieving the mobile device of the time- and energy-consuming tasks, mainly related to communicating with Internet servers and parsing of Web Service Description Language (WSDL) files.

The quarantine area is such an area that as long as all result objects stay inside it and all non-result objects stay outside it the results of this query do not change. This area is used to identify the affected queries upon a source-initiated location update. For a range query, the quarantine area is simply the query rectangle; for a kNN query, the area can be any cir-cle centered at the query point[1]. The first scan is for deciding the *k*NN search range and the second scan is for retrieving *k* objects based on the search range .Therefore, we propose a sharing based nearest neighbor(SBNN) query approach to improve the current on air *k*NN query algorithm.[4]. Peer-to-peer cooperative caching can bring about several distinctive benefits to a mobile system improved access latency, reduced server workload, and alleviated point-to-point channel congestion.

II. PRELIMINARIES

Spatial queries are one of the most important LBSs. According to spatial constraints, spatial queries can be divided into several categories including nearest neighbor (NN) queries and window queries. In general, a mobile client continuously launches spatial queries until the client obtains a satisfactory answer. The naive method answering continuous spatial queries is to submit a new query whenever the query location changes. K.A., Cao *et al.* [10] a naive solution for the out-of-sync problem is once the client wakes up, it empties its previous result and sends a wakeup message to the server. The server replies by the query answer stored at the server side.

The naive method is able to provide correct results, but it poses the following problems: High power consumption. The power consumption of a mobile device is high since the mobile device keeps submitting queries to the LBS server. Heavy server load, continuous query usually consists of a number of queries to the LBS server, thereby increasing the load on the LBS server.

The architecture incorporates a man in- the-middle (MIM) server which will be used by mobile devices to discover needed web services and build their proxies. After getting the proxy, a mobile device can invoke a particular method of the web service and get the desired results. More specifically, the MIM server offers a web service which exposes a web method that the mobile device invokes and passes to it a search string. The MIM server's special service (or simply the MIM server) compares the submitted string to cached short-descriptions of Internet web services and generates a short list of services that best match the user's string. Next, the MIM server downloads the WSDL files of the short-listed services, and uses the included descriptions of the supported methods to identify the most appropriate service (i.e., the one whose method's description matched the user's query the most). After this step, the MIM server generates a source code file from the WSDL file of the chosen service and then compiles it using libraries that target the mobile device platform to generate the client-side proxy and ship it to the mobile device. The second approach, which we have adopted. Moreover, since mobile devices depend primarily on their battery power to communicate and run applications, power conservation becomes critical if the device needs to communicate over the Internet and serve user requests that involve discovering services and invoking their methods.

III. MAN IN THE MIDDLE

The architecture incorporates a man in- the-middle (MIM) server which will be used by mobile devices to discover needed web services and build their proxies. After getting the proxy, a mobile device can invoke a particular method of the web service and get the desired results. More specifically, the MIM server offers a web service which exposes a web method that the mobile device invokes and passes to it a search string. The MIM server's special service (or simply the MIM server) compares the submitted string to cached short-descriptions of Internet web services and generates a short list of services that best match the user's string. Next, the MIM server downloads the WSDL files of the short-listed services, and uses the included descriptions of the supported methods to identify the most appropriate service (i.e., the one whose method's description matched the user's query the most). After this step, the MIM server generates a source code file from the WSDL file of the chosen service and then compiles it using libraries that target the mobile device platform to generate the client-side proxy and ship it to the mobile device. The second approach, which we have adopted. Moreover, since mobile devices depend primarily on their battery power to communicate and run applications, power conservation becomes critical if the device needs to communicate over the Internet and serve user requests that involve discovering services and invoking their methods.

S. Saltenis *et al*. [5] introduce a location-based querying approach that is closely related to our work. They define *validity regions* for which the result of a query remains the same. Hence, clients do not need to resubmit queries that will not change the result set.In contrast to our work, they do not consider order sensitive ranking queries. They acknowledge the necessity of incremental algorithms that reduce redundant computations and communication, a requirement which we directly address in our approach.

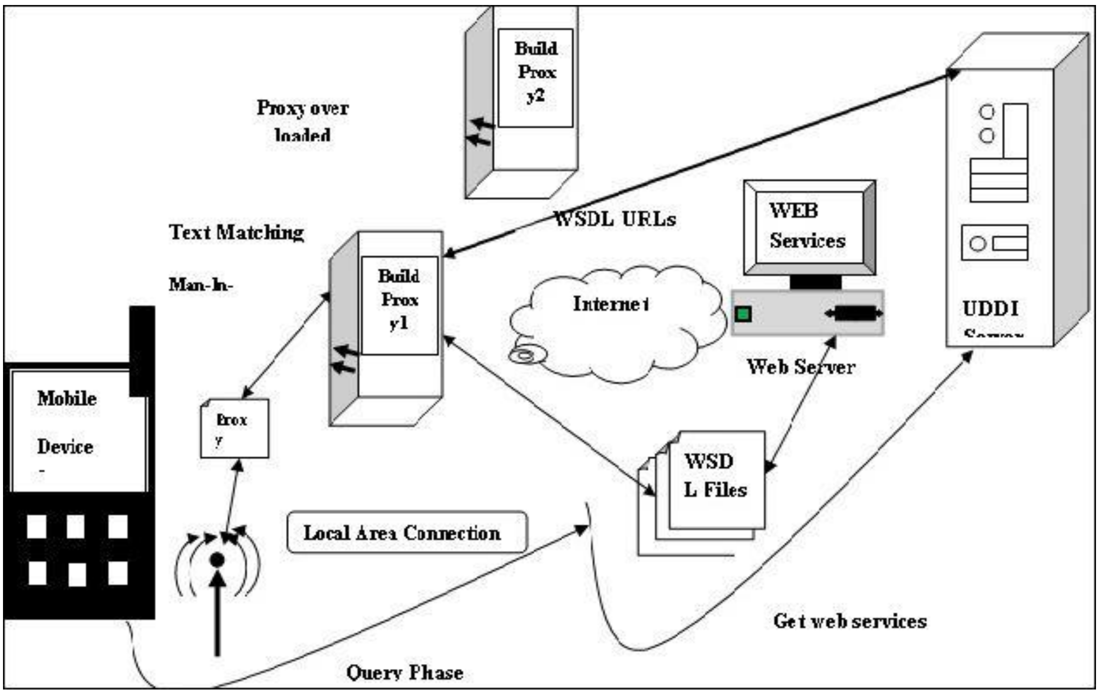

Fig. 1 Architecture Diagram

It depicts the suggested scheme architecture for NN and window query processing. The scheme architecture comprises of three parts: 1) an external LBS server, 2) Established proxies, and 3) the wireless purchasers. The LBS server is to blame for managing static data objects and responding the queries submitted by the proxies. The LBS server can use any catalogue structure (e.g., R-tree or grid catalogue) to method spatial queries. Each base station serves as an intermediate relay for queries between wireless purchasers and the affiliated proxy. Base stations ,proxies, and the LBS server are attached by a attached network. A mobile purchaser sustains a cache to store the query results and the corresponding EVRs. When a wireless purchaser has a spatial query, the wireless device first examines whether the current location is in the EVR of the retained result. As mobility introduces dynamic changes to the fundamental physical environment of end users, all these flows are expected to suffer deterioration caused by mobility [3].

IV. BOYER-MOORE ALGORITHM

This algorithm is suitable for this type of applications as it works the fastest when the alphabet is moderately sized and the pattern is relatively long. It scans the characters of the pattern from right to left beginning with the rightmost character. In our implementation, the measure of correspondence between the user-supplied search string.

```
void bmSearch ( )
{
   int i=0, j;
   while (i<=n-m)
   {
      j=m-1;
      while (j>=0 && p[j]==t[i+j]) j--;
      if (j<0)
      {
         report(i);
         i+=s[0];
      }
      Else
         i+=Math.max(s[j+1], j-occ[t[i+j]]);
   }
}
```

The Boyer-Moore algorithm searches for occurrences of P in T by theatre-in-the-round bright scent comparisons at different alignments. There are some distributed approaches such as booyer [6] and that rely on the processing capabilities of moving objects to detect changes in the answer presented to the user. In preference to of a brute-force cross-examination of enclosing alignments (of which in the air are m - n 1), Boyer-Moore uses information gained by preprocessing P to skip as many alignments as possible. In front of roam was down trodden the to be to come code of the comfortable would be compared to the reminder of the recur. If petty ponder occurred exhausted enough the topic would without exception be bad redolence by bouquet in an effort to find a steadiness.

V. EXTENSION OF MAM REGION

A. Communication-Level

At this level, proxies are in charge of handling all sorts of issues related to the communication protocols and abstractions. The main goal is to make device mobility and use of wireless links transparent to the higher software layers. Typical adaptations at this level are wired-wireless protocol translation or optimization, buffering, handover management. The probability of exiting the cell is critical in deciding how to incrementally maintain the index [2].

B. Middleware-Level

At the middleware level, proxies perform general tasks neither tailored to a specific type of application, nor related to a specific communication protocol. Examples are some forms of content adaptation, consistency management of cached data, service or resource discovery, security and authentication, and others.

The generated queries are served one by one. If the queried data is in the local cache, the client can serve the query locally otherwise, the client has to request the data from the server [7]. This level creates from the fact that middleware that exhibits temporal locality [9]. However, it views such approach is inadequate since their size can be large, resulting in many objects being evicted from the cache in order to free space.

C. Application-Level

Some proxy-based architecture is focused on a specific type of application such as Web-browsing, database access, P2P data sharing, and others. In this case, proxies execute tasks tailored to specific requirements and functions of an application class. For example, when comparing caching in Web and database applications, the former handles heterogeneous objects and essentially aims at reducing response time, while the latter usually handles homogeneous data but requires management of cache consistency.When a query intersects a semantic region in the cache,that region gets split into two smaller disjoint semantic regions, one of which is the intersection of the semantic region and the query, and the other is the difference of the semantic region with respect to the query [8].

D. Extensibility/Programability

Proxy extensibility, i.e. the possibility to adapt and customize its functions, is also an important criterion to differentiate architectures. In most systems, the proxy has pre-defined adaptive behavior, usually determined by the current state of the execution environment. As a first step towards extensibility, some approaches provide a generic framework in which proxies can be easily tailored to the specific needs of an application or middleware loading of filters or new modules implementing specific functionality.

VI. CONCLUSIONS

The primary goal of this report is suggested that the a proxy-based approach to continuous NN and window queries wireless Environments. The proxy takes advantage of spatial and temporal locality of spatial queries to conceive EVRs of NN and window queries distinct from former work, we devised new EVR creation and elongation algorithms for NN queries, enabling the proxy to construct productive EVRs productively. The developed algorithms make the proxy achieve high presentation even when the cache size is little. On the other hand, we suggest to comprise EVRs of window queries in the pattern of vectors called proximate window vectors, to achieve bigger approximated valid districts. After getting the proxy, a mobile device can invoke a particular method of the web service and get the desired results. More specifically, the MIM server offers a web service which exposes a web method that the mobile device invokes and passes to it a search string. The MIM server's special service (or simply the MIM server) compares the submitted string to cached short-descriptions of Internet web services and generates a short list of services that best match the user's string.

ACKNOWLEDGMENT

This work was supported in part by Industrial Technology by the formation of Research Institute of Taiwan, ROC, under contractA301AR2210, and by National Science Council of Taiwan, ROC, under contracts 99-2221-E-009-140-MY2 and 99-2219-E-002-029.

REFERENCES

[1] J. Chen, D. DeWitt, F. Tian, and Y. Wang. NiagaraCQ: A scalable continuous query system for internet databases. In Proc. SIGMOD, 2000.

[2] K. Raptopoulou, A. Papadopoulos, and Y. Manolopoulos. Fast nearest-neighbor query processing in moving-object databases. *GeoInformatica*, 7(2):113–137, 2003.

[3] N. Kara and V. Planat. Performance analysis of IP multimedia services over HSDPA mobile networks.pages 1–5, Dec. 2007.

[4] C.-Y. Chow, H. V. Leong, and A. T. S. Chan. Distributed Group-based Cooperative Caching in a Mobile Broadcast Environment. In *Mobile Data Management*, pages 97–106, 2005.

[5] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the positions of continuously moving objects. In *Proceedings of ACM SIGMOD*, pages 331–342, Dallas, TX, May 2000.

[6] Y. Tao, N. Mamoulis, and D. Papadias, "Validity Information Retrieval for Spatio-Temporal Queries: Theoretical Performance Bounds," Proc. Symp. Spatial and Temporal Databases (SSTD),pp. 159-178, 2003.

[7] Q. Hu and D. Lee, "Cache Algorithms Based on Adaptive Invalidation Reports for Mobile Environments," Cluster Computing, pp. 39-48, Feb. 1998.

[8] A. Keller, J. Basu. A predicate-basedcaching scheme for client-server database architectures. *VLDB J*, 5(1), 1996.

[9] N.E. Young, "On-line caching as cache size varies," *in Proc. of Symposium on Discrete Algorithms (SODA'91),*pp. 241-250, Jan., 1997.

[10] Cai, Y., Hua, K.A., Cao, G.: Processing Range-Monitoring Queries on Heterogeneous Mobile Objects. In: Mobile Data Management, MDM. (2004)

