A Semantic-Driven Cache Management Approach for Mobile Applications

Guiyi Wei, Jun Yu, Hanxiao Shi, and Yun Ling

Zhejiang Gongshang University, Hangzhou, 310035, P.R. China
weiguiyi@tom.com, {yj, hxshi, yling}@mail.zjgsu.edu.cn

Abstract. With the development of wireless communication technology, mobile business become more and more popular. Using GPRS or WAP protocals, the wireless devices can connect to the Web servers, retrieve information from the online databases and run special application programs. Because of the limitation of the wireless communication and mobile computing enviroment, it is difficult to improve the execution efficiency for the program that located in mobile devices. To solve the problem, introducing the cache mechanism is the major and effective method. But the traditional cache model can not achieve an acceptable cache hit ratio. The semantic caching is particularlly attractive in a mobile business environment, due to its content-based reasoning ability and semantic locality. In semantic-driven cache model, only the required data is transmitted to wireless device. In this paper we propose an application-oriented semantic cache model. It establishes an semantic associated rule-base according to the knowledge of application domains, makes use of the semantic locality for data prefetching, and adopts a Two-level LRU algorithm for cache replacement. Several experiments demonstrate that the semantic-driven cache model can achieve higher hit ratio than traditional models.

1 Introduction

E-business is being used to overcome the geographic limitation and improve the efficiency of commercial activities[1]. It plays a more and more important role in world commerce for its high quality of services. At the same time, the popularization of the wireless devices, such as cell-phone, PAD, laptop and etc., has enabled the mobile business becoming realized. Some mobile applications have come into our lifes. People can play nework games with connecting to an internet host through wireless network. Customers can perform online payment through remote banking system. Individual investors can browser the instant quotations of the stock market and make stockbroking. The travelers can book hotel with their wireless communication devices. Therefore, the development of mobile business brings the revolutionary changes to our future lifes.

Using GPRS or WAP protocals, the wireless devices can connect to the Web servers, retrieve information from the online databases and run special application programs. In some case, the mobile device need to run some special programs with

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part III, LNCS 3993, pp. 184 – 191, 2006.

amount of data, such as reporting, analysis and etc., rather than only run a Web browser. Because of the limitation of the wireless communication and mobile computing enviroment, it is difficult to improve the execution efficiency for the program that located in mobile devices. To solve the problem, introducing the cache mechanism is the major and effective method. But the traditional cache model can not achieve an acceptable cache hit ratio. The major method to solve this problem is to adopt the cache management with prefetched data mechanism[4,5]. The related research attemped to apply command-driven and data-driven methods for data prefetching. These methods partly alleviate the problem of the low bandwidth and high delays in the mobile computing environment[6,7]. Lacking the semantic understanding for application domains, the production of associated rules, the method of data prefetching and cache replacement are blindness and maybe invalid. It would waste limited bandwidth. And the hit-ratio and validity of cache are not observably improved. The semantic caching is particularlly attractive in a mobile business environment, due to its content-based reasoning ability and semantic locality. In semantic-driven cache model, only the required data is transmitted to wireless devices. In this paper we propose an application-oriented semantic cache model. It establishes an semantic rule-base according to the knowledge of application domains, makes use of the semantic locality for data prefetching, and adopts a Two-level LRU algorithm for cache replacement.

2 Mobile Processes Model

In the mobile process model (depicted in Fig.1), the mobile application consists of a client component and a server component. The client program resides in mobile terminals, and the server program and database are reside in the online web hosts. They are connected with wireless comminication network [2,3].

Before the client program execute, it is divided into a key process and several relative independent sub processes or threads, which are responsible for query generating and data processing. A supervisor process of local operating system take charge of coordination of these sub processes and the key process. It schedules processes with consideration of the processor's workload, state of the network and data ready state in the cache.

The cache of the mobile terminal is divided into a basic cache and a prefetched data set. The basic cache accommodates the data produced with the temporal locality principle. The prefetched data set stores the data prefetched with the semantic locality principle, which will be used when the basic cache mismatch (not hit). The prefetched data set is replaced using assosiated rules which derived from the knowledge of special application domain.

The server side provides source data management services. The data service is divided into several lightweight transactions. Each lightweight transactions is response to one sub-process of mobile client. This will alleviate the impact of the conceivable network linkage failure.

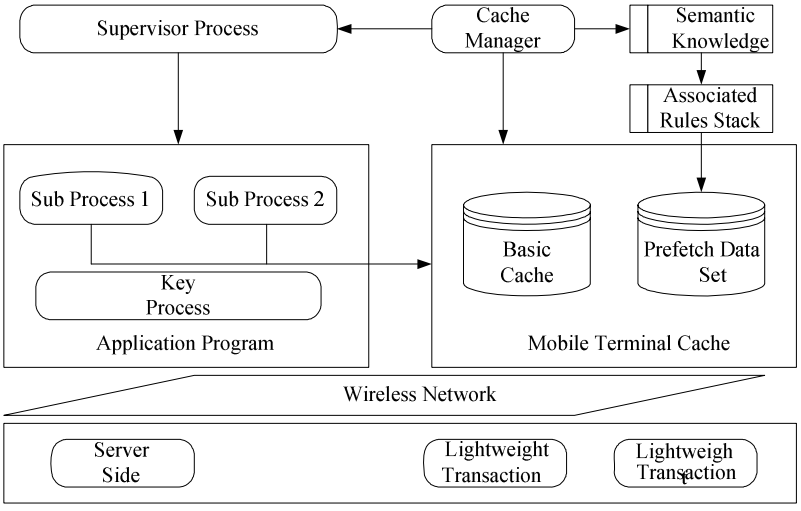

Fig. 1. The process model in the mobile application environment

3 Cache Management Language (CML)

3.1 Definition of CML

The CML language should express three aspects of mobile cache management, the data correlativity, the data dependency and the critical point of replacement. It is similar to the profile language in [10]. The CML instruction is executed by a grammar interpreter without compiling. In CML, there are three basic elements, *domain*, *utility* and *operation*. The *domain* defines a data set, the *utility* defines relations of data in *domains*, and *operation* is action performed when certain conditions are satisfied.

3.2 Prefetch Algorithms

At present, the mobile network communication is characterized by low bandwidth and high latency. It is a challenge to mobile applications in dada communication. The mobile terminal usually uses the method of demanded data prefetching to handle these problems. The prefetch algorithm for mobile cache can be classifed two types, preemptive and non-preemptive. Our cache model mainly aimed the mobile business application that is interactive program and performs complex dada processing workflow, so the preemptive prefetch algorithm is adopted.

The preconditions of prefetch algorithm include: O is a finite set of data object; o is any element of O; $s(o)$ is the space of data object o; C is the capacity of semantic cache. pr is the possibility of preemption; $S = <o_1, o_2, \ldots, o_n>$ is a subset of O that satify the conditions of formulas (1)-(5); p is an operation instance described with CML language, g is a utility function; fp is weights calculation function.

$$\sum_{i=1}^{n} s(o_i) \le C \tag{1}$$

$$\text{Max} \quad g_{p,\,pr}(O) \tag{2}$$

$$g_{p,\,pr}(S) = e_1 + e_2 \tag{3}$$

$$e_1 = (1 - pr)(f_p(S)) \tag{4}$$

$$e_2 = pr\left(\frac{(\sum_{i=2}^{n} s(o_i) \cdot f_p(\{o_1, \ldots, o_{n-1}\}))}{s(o_1) + \ldots + s(o_n)}\right) \tag{5}$$

The prefetch algorithm is an ameliorated greedy algorithm according to formulas (1)-(5). It improved the the performance of common greedy algorithm in cache management.

3.3 Semantic Cache Model

The semantic cache usually consists of two parts, the index part and the content part [8, 9]. The content part consists of pages that store the semantic segments, each page has a unique page number for identifcation. A page number can be mapped to a physical page in the cache. The semantic segment can be represented as formula (6): S_R is the source data objects participated prediction and the relationships among them; S_A is the common attributes of a data object set; S_P is the data objects prediction using associated rules; S_C is the data content of semantic segment; S_{PG} is the page that contains the semantic segment; S_T is the latest accessed time of the semantic segment.

$$S = < S_R, S_A, S_P, S_C, S_{PG}, S_T > \tag{6}$$

The emphasis of semantic cache organization is its replacement policy. We use an improved two-level LRU algorithm for semantic cache replacement (detailed described in section 4 of this paper) .The operation of semantic cache is depicted as Fig.2.

3.4 Assosiated Rules Generation

We define semantic knowledge set $P = \{p_1, p_2, \ldots, p_i, \ldots, p_n\}$, data item $F = \{i_1, i_2, i_m\}$. S_i is a set of semantic data item F which adapted to semantic knowledge p_i. If X and Y are two data item sets, $X \subseteq S_i$, $X \subset F$, $Y \subset F$, and $X \cap Y = \Phi$, then $X \Rightarrow Y$ is an assosiated rule.

$Sup(X)$ is the supported degree of $X \subseteq S_i$, $Sup(R) = Sup(\{X, Y\})$ is the supported degree of formula $R: X \Rightarrow Y$. $Conf(R) = \frac{Sup(\{X, Y\})}{Sup(X)} * 100\%$ is the confidence degree.

Before an associated rule generating, a minimum supported degree *min-sup* and a minimum confidence degree *min-conf* are set. The generation algorithm automatically generate an X with $Sup(X)$ is not smaller than *min-sup*. Then it produce a Y according to X , and compute R with *Conf (R)* not smaller than *min-conf*.

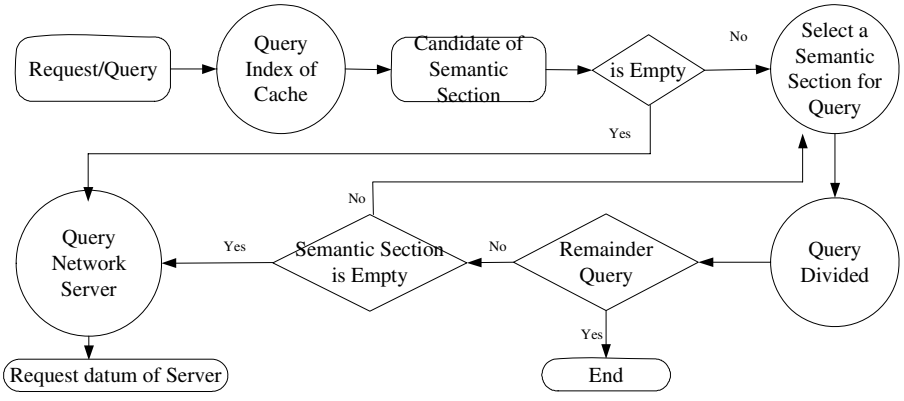

Fig. 2. Operation of semantic Cache

4 Replacement Algorithm

Traditional cache replacement policies are based on temporal locality or spatial locality. In order to preserve data items that are most likely to be accessed in the future and improve cache hit-ratio, our cache model adopted semantic locality for cache replacement. It use a two-level LRU(least recently used) replacement policy to predict the future access probablity by analyzing the semantic locality with associated rules.

The first level LRU generates candidate page set using temporal locality policy while cache mismatch. The second level LRU uses semantic locality policy to choose page from candidate page set. The pages selected by second level LRU will be swapped out.

Algorithm. Two-Level LRU Cache Replacement

```
while ( page fault occur)
{
  cs <- generate candidate page set using temporal
        locality based LRU algorithm (First Level LRU);
  while ( cs is not NULL)
  {
    sd <- max value;
    sp <- NULL; selected page;
    p  <- fetch next page from cs;
    psd <- compute the support degree with p and semantic
           assosiated rules;
    if psd < sd then
      {sd <- psd; sp <- p;}
     move p from cs;
   }
   discard p from semantic cache;
   add free page;
 }
```

5 Experiments

A web based mini ERP system and a mobile phone are used to simulate a mobile business application environment. The ERP system includes a sale management module and a data analysis module. These modules are interactive programs, and run with volumes of data. The mobile phone can adapt to embedded memory with variable volume sizes. The mobile device communicates with online application server using GPRS protocal. Before the client of the application executes, its kernal code and necessary data have been reside in the mobile terminal. The kernal code will run as kernal process as depicted in the figure 1. It surely can be downloaded and updated online. During the execution of the program, additional function codes of the application may be downloaded as demanded, and volumes of data should be transferred from and to between client and server.

Based on the implemented prototype of our proposed model and its algorithms, some performance evaluations are carried out.

In the first experiment, we split whole client memory up into two parts equally (50% basic cache and 50% prefetched data set). The replacement of basic cache uses the first level LRU algorithm with temporal locality. The replacement of prefetched data set uses the second level LRU algorithm with semantic locality. The changes of cache hit ratio according to different cache size are illustrated in Fig.3.

In the second experiment, whole cache size is fixed to 1 megabytes, prefetch data set size is changed dynamically from 40% to 90% . Additionally, with the same cache size, a data-driven method and a command-driven method are evaluated to compare with the semantic-driven method. Tests of data-driven method and command-driven method use same replacement algorithm for two parts of caches. The changes of cache hit ratio of different methods are illustrated in Fig.4.

As depicted in figure 4, we ca can see: (1) the semantic-driven method can achieve higher hit ratio with more prefetch set proportion given initially, the highest hit ratio appears with the proportion of 75% approximately, and then hit ratio declines gradually; (2) the semantic-driven method can achieve higher hit ratio than command-driven method and data-driven method. Further experiments will be done to refine semantic-driven model and its algorithms.

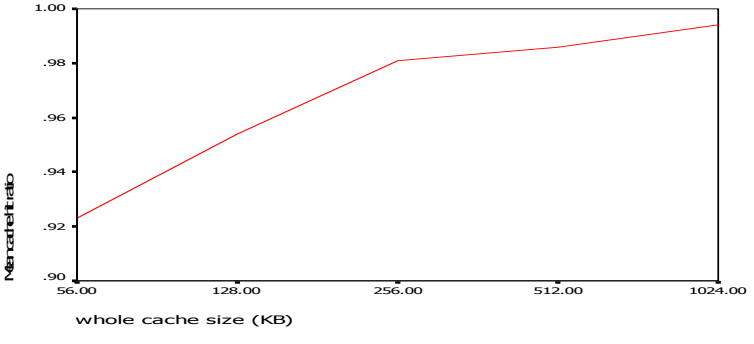

Fig. 3. Cache hit ratio according to different cache size with fixed division

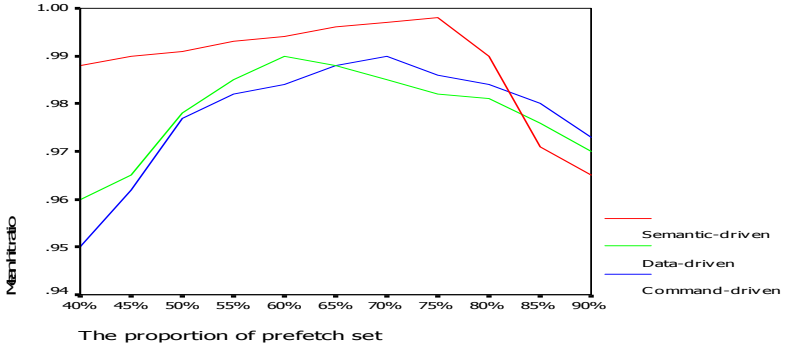

Fig. 4. Cache hit ratio according to different methods with dynamic division

6 Conclusions

In this paper, we proposed an mobile process model suitable for mobile client program, and discussed a semantic-driven model for cache management of mobile devices in mobile business environment. The design and implementation of a prefetch algorithm and a two-level LRU cache replacement algorithm are detailedly illuminated. The results of corresponding experiments demonstrated that the semantic-driven model (1) is effective and efficient, (2) can achieve higher hit ratio than data-driven and command-driven methods.

Acknowledgments

The authors wish to thank the Natural Science Foundation of Zhejiang Province, P. R. China for the National Science Fund under grant Number Y105356. We would like to thank CRMB (the Center for Research in Modern Business, Zhejiang Gongshang University) for they partially supported this research. We would also like to thank our referees for their helpful comments and suggestions.

References

1. André Köhler, Volker Gruhn: Analysis of Mobile Business Processes for the Design of Mobile Information Systems. Proceedings of E-Commerce and Web Technologies, EC-Web 2004, LNCS3182, Springer, (2004)
2. Noor, N. M. M., Papamichail, K. N., Warboys, B.: Process Modeling for Online Communications in Tendering Processes. Proceedings of the 29th EUROMICRO Conference. IEEE Computer Society, (2003)17-24
3. Ritz, T., Stender, M.: Modeling of B2B Mobile Commerce Processes. 17th International Conference on Production Research ICPR-17. Virginia Tech, Blacksburg, (2003)
4. D. Barbara and T. Imielinski: Sleepers and Workaholics: Caching Strategies for Mobile Environments. ACM SIGMOD, (1994)1–12

5. G. Cao: A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments. IEEE Transactions on Knowledge and Data Engineering. ACM MobiCom'00, (2000)
6. H. Song and G. Cao: Cache-Miss-Initiated Prefetch in Mobile Environments. IEEE International Conference on Mobile Data Management (MDM), (2004)
7. V. Grassi. Prefetching Policies for Energy Saving and Latency Reduction in a Wireless Broadcast Data Delivery System. In ACM MSWIM 2000, Boston MA, (2000)
8. R. Agrawal and R. Srikant: Fast Algorithms for Mining Association Rules. Proc. 20th Int. Conf. Very Large Data Bases. Morgan Kaufmann (1994)487–499
9. R. Agrawal, Tomasz Imielinski, and Arun Swami: Mining Association Rules Between Sets of Items in Large Databases. In Proc. of the ACM SIGMOD Conference on Management of Data. (1993) 207–216
10. Mitch Cherniack,Eduardo F. Galvez,Michael J. Franklin, Stan Zdonik: Profile-Driven Cache Management. Proceedings of International Conference on Data Engineering (ICDE), Bangalore, India, (2003)