



An Approach to Improve the E-Commerce Performance using Hybrid Caching Architecture: A Review

Arti Brar*
CSE, MITM
India

Mrs. Shagun Girdhar
CSE, MITM
India

Abstract— the cost and time issues commonly arises for E-Commerce websites. As the number of users increase, the performance related problems need to be solving by caching concepts. E-business sites are using more dynamic pages because dynamic pages provide much wider range of interaction than static HTML pages. Based on various parameters, Dynamic page generation technologies allow a Web site to generate pages at run-time. Postponing content choices until run-time affords a Web site significant customizing page content, thereby enriching users' Web experiences. On the other hand, dynamic page generation technologies have resulted in serious performance problems due to the greater than before load placed on the server-side infrastructure. For many software applications, system support for caching us insufficient for improving performance; application-level caching is needed as well. This paper provides an overview of caching technique for building application level caches. We also describe how we deployed application level caching for improving performance. The hybrid caching concept has been used in this paper.

Keywords— Caching, Application, Hybrid, Dynamic Web-Pages, E-Commerce.

I. INTRODUCTION

The performance of the web services is the major factor of consideration to grow the business at higher level and response of such kind of services should be efficient in term of time and memory which is based on efficient algorithm and techniques. There are existing techniques on which the complete E-Services are based. Caching in web application can be done mainly in these ways, on the server side (data caching/page output caching) or in between the client and the server (proxy and reverse proxy caching) or on the client side (client browser)

There should be efficient e-Business infrastructure, in Multi-user environment; Caching technique for database seems good option because of enhancements to the following attributes:

- (A) It provides flexibility with Quality Of Service control where each cache hosts different parts of the backend data, e.g., the data of Platinum customers is cached While that of ordinary customers is not
- (B) It increases performance by potentially responding to locality patterns in the workload and smoothing out load peaks.
- (C) It also increases availability by continued service for applications that depend only on cached tables even if the backend server is unavailable.
- (D) It also increases scalability by distributing query workload from backend to multiple cheap front-end systems.

The Application level cache means maintains the cache at application level. The business related web services are often critical part of infrastructure which is needed for the success of organization in terms of processing time. The application performance benefits can be achieved by the query calculation. The cached objects are often stored in hash table and balanced tree and indexing which can be byte stream, numerical value or by the Strings. The application level cache has been designed with API which allows the developer to manage the cache contents explicitly. There are all features likely to the database which can be update or Modify, Add and Delete. The developers can easily work with the cache by taking the knowledge of application specific Structure when caching of data. There are different approaches for caching which can be described as hybrid Caching, shareable and un-shareable caching.

In un-shareable approach for Caching of Data, the cache is run independently. This feature enable feature of avoid the overhead and complexity of inter-process Communication. There is method which can be used for deploying a cache is to publish as a library which will be as a part of application.

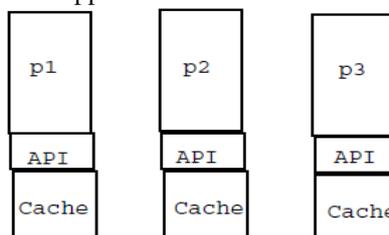


Fig1 Graphical Representation of un- shareable approach

If an application is distributed across multiple processes which need to access a cache, the unshared architecture may be unsatisfactory because it requires each process to manage a different cache. Extra space is required for storing same data in multiple copies if multiple processes need to access it.

In shareable approach, cache operates as a long running process which communicates with multiple processes is known as shared architecture. Using this architecture, it's only necessary to cache one copy of an object. There are two drawbacks to the shared architecture; one is latency for accessing a cached object may be higher because inter-process communication is required. Second drawback is that if the request rate to a shared cache is high, the cache can become a bottleneck.

II. LITERATURE REVIEW

The author has been explained the concept of caching to improve the performance of websites. Today a major drift is observed from static website to dynamic websites. The dynamic websites deliver the customized contents to their users. These websites cover a wide spectrum of applications which largely vary in its content, configuration and volume of traffic. The popular metrics designed to evaluate a dynamic website are response time, user satisfaction, performance, scalability, usability personalization, reliability, reusability and security. But for dynamic website delivery, update, consistency maintenance are main hurdles. To overcome them and to serve and deliver the contents efficiently for dynamic website researchers have proposed several caching mechanisms with an aim to reduce the construction overhead to improve the response time as well as reusability of its contents. A great research has been performed on various aspects of caching [1].

Author explained that the increasing popularity of the World Wide Web is placing great demands on the Internet. The world wide can be well-thought-out as a large distributed information system providing access to shared data objects. World Wide Web is one of the most popular applications currently running on the Internet, it is of an exponential growth in size, which results in network congestion and server overloading. To meet these increasing demands a key strategy for scaling the Internet is to cache data near clients and thus improves access latency and reduce network and server load. Though web cache can decrease the load on the server considerably, network congestion, and the latency of access, it arises a question of cache consistency. That is, a stale cache might return to a client. To avoid returning stale pages to users, all web caches must try to keep cached objects up to date with the original copies of that object. They initially defined an existing web cache consistency technique which has been used in web caching system and proposed a strong cache consistency that is be appropriate for web appellations. Our proposed system is more efficient than previous strong consistency.

This paper proposes an optimistic concurrency control scheme for use in transactional client-server database systems. To improve performance, objects are cached at the clients. The proposed scheme provides serializability for committed transactions. This is in contrast to many modern systems that only provide the snapshot isolation property which is weaker than serializability. A novel feature is that the processing load for validating transactions at commit time is shared between clients and the database server, thereby reducing the load at the server. The performance of the scheme is evaluated using simulation experiments. The experiments demonstrate that for mostly read only transaction load - for which caching is most effective - the scheme outperforms the existing concurrency control scheme with client-side caching considered to be the best, and matches the performance of the widely used scheme that only provides snapshot isolation.

The Distributed Web caching system provides fast retrievals of web pages but still limited by the delays incurred by servers. As these systems are affected and by low robustness due to highly congested servers. As in real time, servers get disconnected frequently providing tradeoffs to service. Robust Distributed Web Caching (RDWC) handles robustness but fails to deal with the frequent disconnections. So in this paper, we discussed effective Distributed Web Caching (DWC), Distributed Web Caching with Clustering (DWCC) and Robust Distributed Web Caching techniques. Then a scheme is proposed and designed which can overcome frequent disconnections of proxy servers & more delays incurred in these schemes, it can send the replies back to clients in lesser time (by factor n- Best Case). Clustering of Proxy Servers is used with dynamic allocation of requests to less congested servers to achieve load balancing and robustness. Then a comparison of our Scheme is done with these and results are given in comparative table.

To increase the speed of Web loading processes web caching and pre-fetching are vital technologies. Since speed and memory are critical aspects in improving the performance of mobile applications and websites, a superior technique for Web loading process should be investigated. The flaws of the conventional Web caching policy include meaningless information and vagueness of knowledge representation in Web logs data from the proxy cache to mobile-client. To deal with the uncertainties explicit representation is required by the organization and learning task of the knowledge-processing for Web logs data. This is because there is exponential growth in rules for finding an appropriate knowledge representation from the proxy cache to the mobile client. Consequently, Rough Set is selected in this research to generate Web pre-caching decision rules to guarantee the meaningless Web log data can be changed to meaningful information.

III. PROBLEM FORMULATION

The Response time is high in E-Commerce websites due to High number of request/ Load.

1. Fetch same data from database for every User.
2. Repeat processing for same kind of search.
3. Multiple query execution at same time decrease site performance.

4. Shared caching architecture having problem of higher latency for accessing a cached object because of inter-process communication.
5. Unshared caching is required extra space for storing all n copies. Maintain Consistency is also required.

IV. OBJECTIVES

1. Remove disadvantages of Shared and Unshared architecture of caching.
2. Design a new Architecture, Hybrid architecture by combining both caching architectures.
3. Design an Algorithm which will handle problem of same data fetching process by maintaining search keywords and counts.
4. Remove caching Memory to store new Data

V. PROPOSED METHODOLOGY

The flow chart has been proposed for enhance the performance of E-commerce websites and proposed work will use the concept of shareable and un-shareable known as hybrid caching.

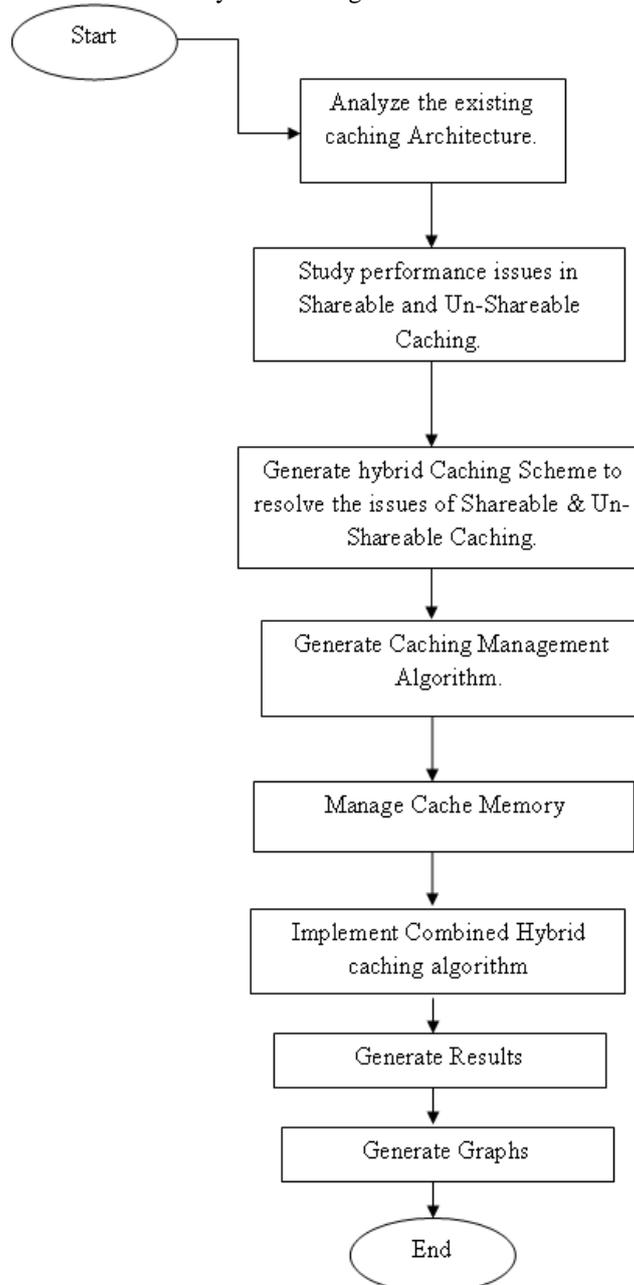


Fig2 Proposed Flow Chart

VI. CONCLUSION AND FUTURE WORK

In this paper, we have been proposed the Caching algorithm and Concept which can be used for enhance performance of E-Commerce website. The proposed is not explained in this paper.

The implementation part will be covered in the next paper, which will demonstrate the real working of proposed algorithm.

REFERENCES

- [1] Mehrotra, D. ;Nagpal, R. ; Bhatia, P., “*Designing metrics for caching techniques for dynamic website*”, Publication Year: 2010 , Page(s): 589 – 595
- [2] YanjunLiu; Yueping Wang ; Huilin Du, “*Strong cache consistency on World Wide Web*”, Advanced Computer Theory and Engineering (ICACTE), Publication Year: 2010 , Page(s): V5-62 - V5-65
- [3] Bukhari, F. ;Shrivastava, S., “*An Efficient Distributed Concurrency Control Scheme for Transactional Systems with Client-Side Caching*”, High Performance Computing and Communication & 2012 IEEE, Publication Year: 2012 , Page(s): 1074 - 1081
- [4] Tiwari, R. ; Kumar, N. , “*Dynamic Web caching: For robustness, low latency & disconnection handling*”, Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference, Publication Year: 2012 , Page(s): 909 – 914
- [5] Sulaiman, S. ; Shamsuddin, S.M. ; Ahmad, N.B. ; Abraham, A., “*Meaningless to meaningful Web log data for generation of Web pre-caching decision rules using Rough Set*”, Data Mining and Optimization (DMO), 2012 4th Conference, Publication Year: 2012 , Page(s): 91 – 98.
- [6] Sathiyamoorthi, V. ;MuraliBhaskaran, V., “*Web caching through modified cache replacement algorithm*”, Recent Trends In Information Technology (ICRTIT), Publication Year: 2012 , Page(s): 483 - 487
- [7] Tiwari, R. ; Kumar, N., “*A Novel Hybrid Approach for Web Caching*”, Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS),Publication Year: 2012 , Page(s): 512 – 517.