CachePortal II: Acceleration of Very Large Scale Data Center-Hosted Database-driven Web Applications

Wen-Syan Li Oliver Po Wang-Pin Hsiung K. Selçuk Candan
Divyakant Agrawal Yusuf Akca Kunihiro Taniguchi

NEC Laboratories America, Inc.
10080 North Wolfe Road, Suite SW3-350
Cupertino, California 95014, USA
Email:wen@ccrl.sj.nec.com

1 Introduction

Wide-area database replication technologies and the availability of data centers allow database copies to be distributed across the network. This requires a complete e-commerce web site suite (i.e. edge caches, Web servers, application servers, and DBMS) to be distributed along with the database replicas. A major advantage of this approach is, like the caches, the possibility of serving dynamic content from a location close to the users, reducing network latency. However, this is achieved at the expense of additional overhead, caused by the need of invalidating dynamic content cached in the edge caches and synchronization of the database replicas in the data center.

A typical data center architecture for hosting Web applications requires a complete e-commerce Web site suite (i.e. Web server, application server, and DBMS) to be distributed along with the database replicas. Typically, the WS/AS/DBMS suite is installed in the network to serve non-transaction requests which require accesses to read-only database replicas of the master database at the origin site. In order to distinguish between the asymmetric functionality of master and slave DBMSs, we refer the mirror database in the data center as data cache or DB Cache. DBCache can be a lightweight DBMS without the transaction management system and it may cache only a subset of the tables in the master database. Updates to the database are handled using a master/slave database configuration: all updates and transactions are processed at the master database at the origin site.

This architecture has two drawbacks: (1) all requests must be processed on demand at the data center or the origin Web site; and (2) the freshness of pages generated at the data center is dependent on the synchronization frequency between the DBCache and the master database at the origin site. Since database synchronization across network is very expensive, the synchronization frequency in most commercial products is usually recommended to be on the order of hours and days. As a result, the dynamic content served from the typical data center-based architecture is not guaranteed to be fresh.

2 CachePortal II System Architecture

In [1], we developed the CachePortal technology for enabling caching of dynamic contents generated by database-driven Web applications. In [2], we demonstrated deployment of such technology on stand-alone E-commerce Web sites. In [3], we develop the invalidation theory to enable dynamic content caching on data centers. In [4], we design a new CachePortal architecture that can be deployed for data center hosted Web applications. The proposed system architecture has been implemented as CachePortal II.

One unique feature of CachePortal is that the construction of mapping between the database content and the corresponding Web pages is automated without modification to the existing applications. We build on these results by developing a novel system architecture that accelerates data center hosted Web applications through deployment of dynamic content caching solutions. The proposed system architecture is shown in Figure 1. A data center system architecture deploying CachePortal II has the following components:

- Master DB Log Manager: It is deployed at the master database. It is responsible for tracking the database content changes and the list of tables mirrored at each data center.
- DB Log Processor: It is installed at each data center and it is responsible for retrieving the database content

**Proceedings of the 29th VLDB Conference,
Berlin, Germany, 2003**

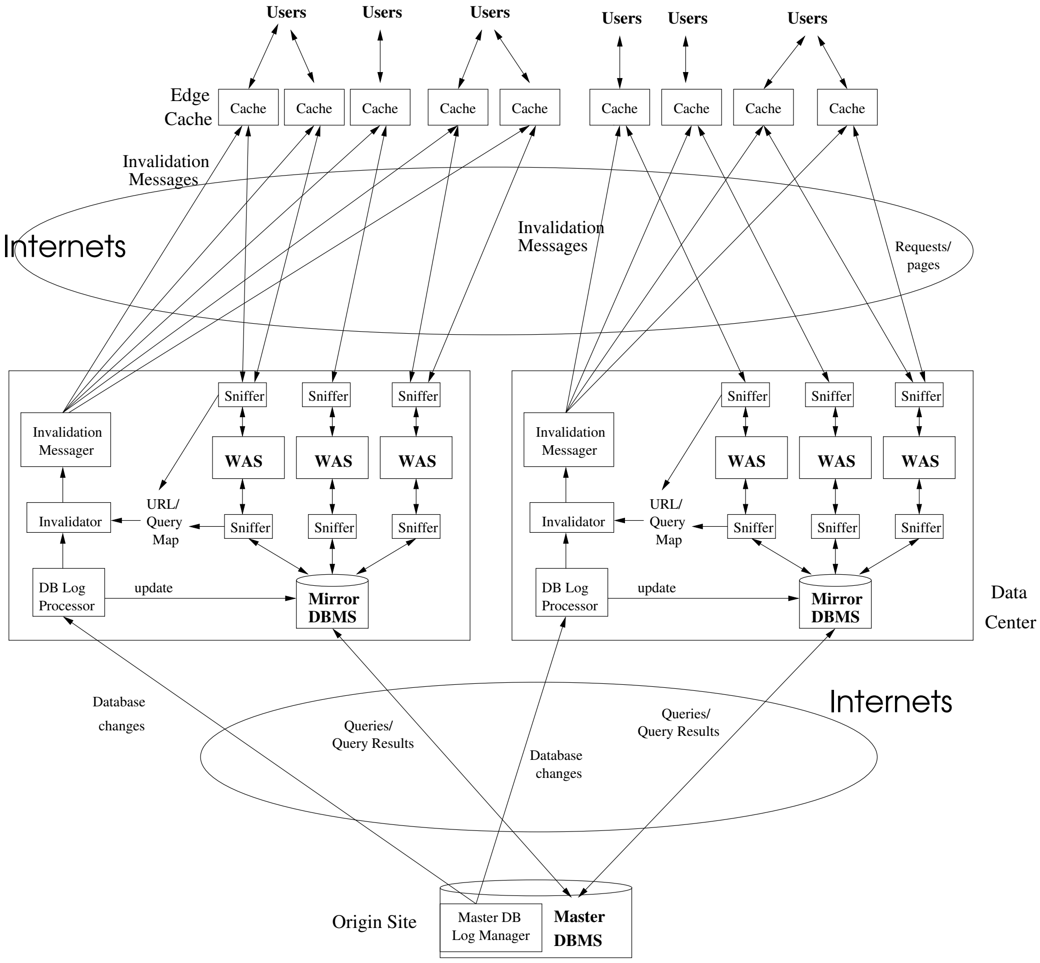


Figure 1: Data Center hosted Database-driven Web Site with Deployment of CachePortal Technology

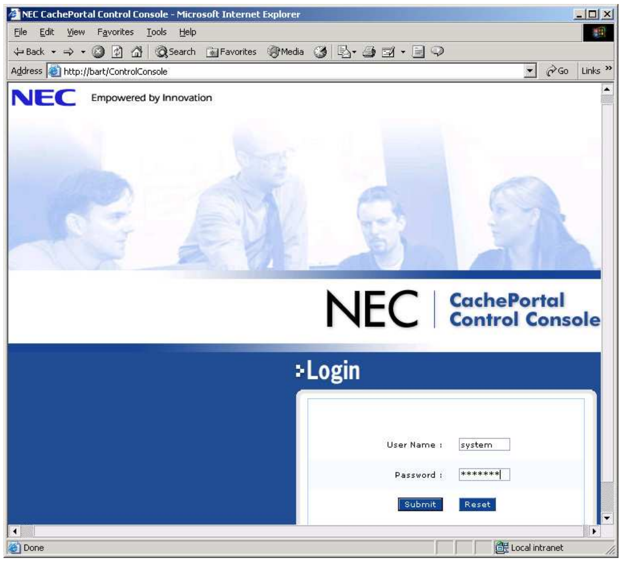
change log from the master database and propagating the changes to the mirror database.

- Sniffer: It is installed at the data center to identify the mappings between the URLs (identifications of pages requested) and the query statements issued for the pages requested.
- Invalidator: It is responsible for performing invalidation checking for the cached pages at the edge caches based on the database content change log, URL and database query mapping, and the content in the mirror database.
- Invalidation Messager: It is responsible for broadcasting invalidation messages to corresponding edge cache servers.

2.1 Operational Flow

The interactions between the components are as follows:

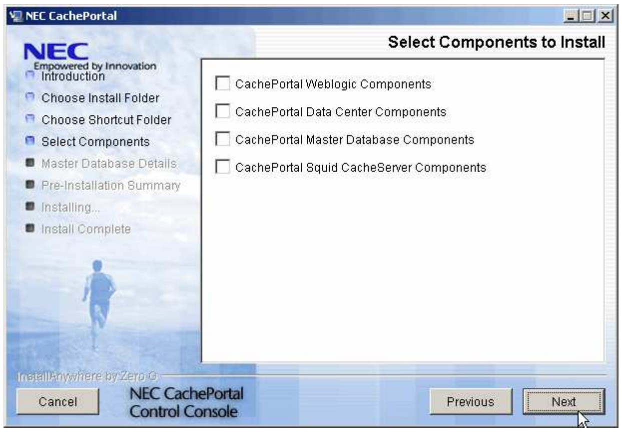
1. A request is directed to the edge cache closest to the user based on the network proximity. If there is a cache hit, the requested page is returned to the user. Otherwise, the request is forwarded to the WAS in the closest data center.
2. If the request is non-transactional and the DB Cache has all the required data, the request is processed and the page is generated dynamically and returned to the user.
3. Otherwise, the request is forwarded to the master database at the original site.
4. The dynamic content pages generated are cached in the edge caches if they are cacheable.
5. The changes to database contents are periodically reflected from the master database at the origin Web site to the mirror database at the data center for synchronization and invalidation. In our implementation, database update log is scanned every second and the new log is copied to the data center.
6. The invalidator runs as a daemon in the data center. It reads the unprocessed database update log and performs the invalidation checking and synchronization tasks. These three tasks are done as an invalidation/synchronization cycle. After one cycle is completed, the invalidator starts the next cycle immediately. Since the log scanning and invalidation/synchronization are performed in parallel, the in-


NEC CachePortal Control Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://bart/ControlConsole
Go Links
NEC Empowered by Innovation
NEC CachePortal Control Console
Login
Submit Reset
Done
Local intranet

User Name : system

Password : *******

Figure 2: CachePortal II System Console



Select Components to Install

☐ CachePortal Weblogic Components
☐ CachePortal Data Center Components
☐ CachePortal Master Database Components
☐ CachePortal Squid CacheServer Components

Figure 3: Installation Selection Console

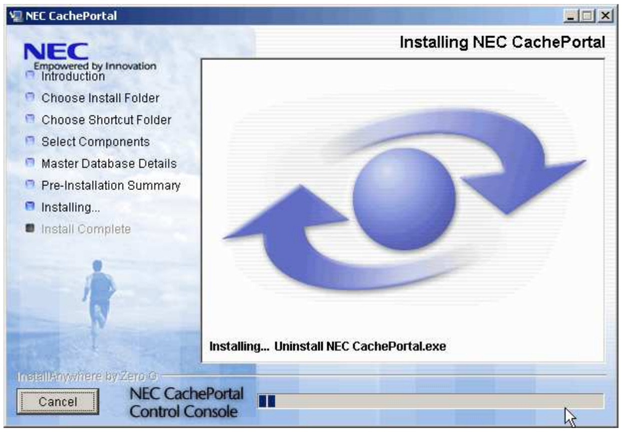
validator does not need to wait for the completion of log scanning.

7. The invalidator passes the list of Web page URLs and the locations of these pages (i.e. IP addresses of edge cache servers) to the invalidation messager. The invalidation messager then sends out messages to invalidate or refresh the Web pages impacted by the database content changes.

As shown in Figure 1, a data center may deploy multiple edge caches and multiple application servers; depending on the user request rates. CachePortal can also be applied to the architecture with a single master database and multiple data centers (i.e. multiple mirror databases). Dynamic content pages in the edge caches may be generated by the database content in the mirror database or the master database.

2.2 Benefit of Deploying CachePortal

The proposed data center architecture with deployment of CachePortal II has the following advantages:



Installing NEC CachePortal

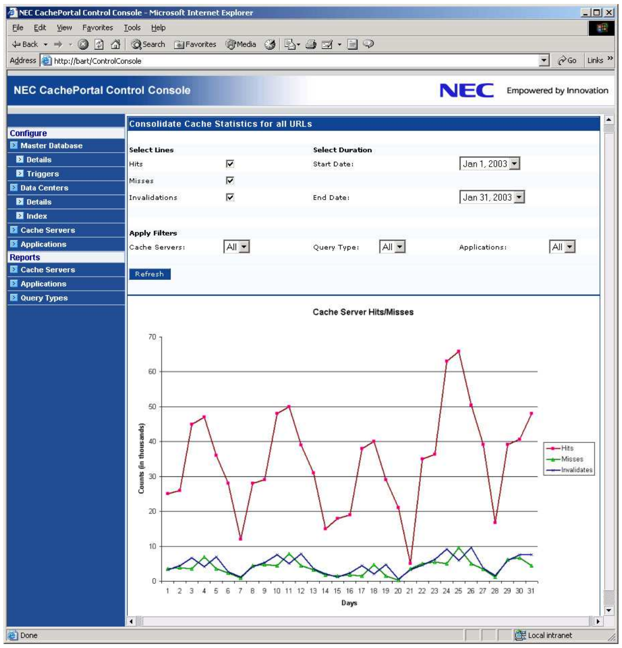
Installing... Uninstall NEC CachePortal.exe

Figure 4: Installation in Progress

- *Fast response time:* Serving cached dynamic content pages is much faster than generating pages on demand; edge caches are deployed close to the end users; consequently, the network latency between end users and data centers is eliminated.
- *Scalability:* Based on the study in [5], the system architecture with deployment of dynamic content caching solution has much higher scalability Since a bulk of the load is distributed to the edge caches, the WAS and the DB cache at the data center have lighter loads. As a result, the Web applications hosted at the data center can generate requested pages faster or can be further extended to serve more customized dynamic content with the same infrastructure. Furthermore, CachePortal II allows faster user response time with less hardware and bandwidth investment and can handle burst traffic.
- *Content Freshness:* Freshness of pages cached in the edge caches and those generated at the data center on demand is assured to be not older than the invalidation/synchronization cycle. Since the freshness that can be provided depends on the length of the invalidation/synchronization cycle, this parameter has to be carefully tuned as described in [4].

3 System Evaluation

CachePortal II is designed to accelerate very large scale data center hosted database-driven Web applications. We have evaluated the system using an e-commerce application and it is capable of tracking 50 million dynamic content pages in 10 cache servers and it assures content freshness of these 50 million pages by invalidating impacted pages within 12 seconds once database content is changed. Furthermore, even a Web site that enables dynamic content caching can further benefit from our solution and improves content freshness up to 7 times, especially under heavy user request traffic and long network latency conditions. Our approach also provides better scalability and significantly reduced response times up to 70% in the experiments. The detailed experimental results are described in [4].



Consolidate Cache Statistics for all URLs

Select Lines

Hits ☑

Misses ☑

Invalidations ☑

Select Duration

Start Date: Jan 1, 2003

End Date: Jan 31, 2003

Apply Filters

Cache Servers: All

Query Type: All

Applications: All

Figure 5: Cache Manager

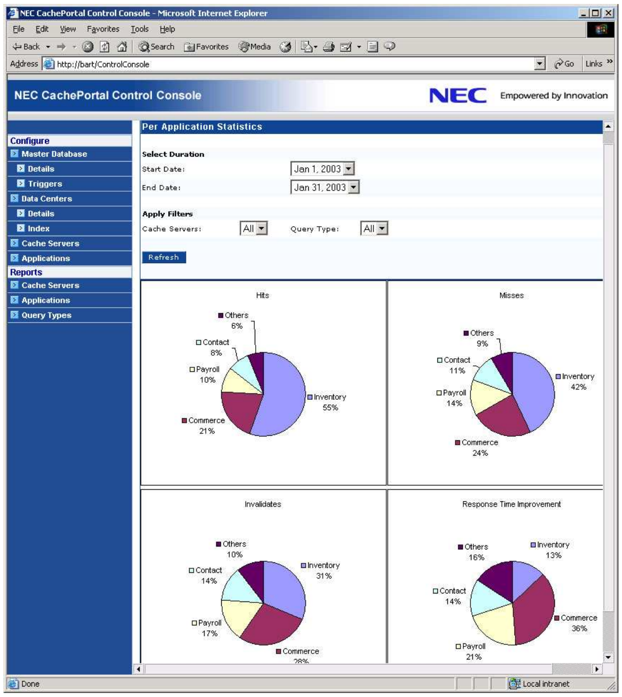
4 Deployment and Management

To demonstrate the functionality and benefits of deploying CachePortal II, a Web site based on data center architecture is implemented using Squid, BEA WebLogic Application Server, and Oracle DBMS and CachePortal II. The database used in this experiment contains 7 tables each holding more than 1 million records. In the demonstration system, the benefits of deploying CachePortal II is clearly visualized in two side-by-side windows, where a large number of parallel requests are served from two Web sites of different configurations: one deploys CachePortal II and the other does not.

Installation and configuration of CachePortal II is user-friendly and it can be performed through a simple Web-based GUI. CachePortal II can be downloaded and remotely installed on an existing Web site and the data center and edge caches can be configured into an integrated content delivery system. Some window dumps are shown in Figure 2, 3, and 4. CachePortal II also supports a suite of operation management tools, including (1) cache performance and behavior monitoring, shown in Figure 5; and (2) application performance monitoring and pattern analysis, shown in Figure 6.

5 Concluding Remarks

This technical description has highlighted many features of CachePortal II using an e-commerce site built using some of most popular components on the market, such as Oracle DBMS and BEA WebLogic. It shows a real and useful application of integrating database and Web technologies. CachePortal II Beta version is currently available for J2EE complient application servers and most commercial DBMSs running on Linux, Unix, and Window 2000/XP.



Per Application Statistics

Select Duration

Start Date: Jan 1, 2003

End Date: Jan 31, 2003

Apply Filters

Cache Servers: All

Query Type: All

Figure 6: Application Manager

References

[1] K. Selçuk Candan, Wen-Syan Li, Qiong Luo, Wang-Pin Hsiung, and Divyakant Agrawal. Enabling Dynamic Content Caching for Database-Driven Web Sites. In *Proceedings of the 2001 ACM SIGMOD Conference*, Santa Barbara, CA, USA, May 2001. ACM.

[2] Wen-Syan Li, K. Selçuk Candan, Wang-Pin Hsiung, Oliver Po, Divyakant Agrawal, Qiong Luo Wei-Kuang Wayne Huang, Yusuf Akca, and Cemal Yilmaz. CachePortal: Technology for Accelerating Database-driven E-commerce Web Sites. In *Proceedings of the 2001 VLDB Conference*, Roma, Italy, September 2001.

[3] K. Selçuk Candan, Divyakant Agrawal, Wen-Syan Li, Oliver Po, and Wang-Pin Hsiung. View Invalidation for Dynamic Content Caching in Multitiered Architectures . In *Proceedings of the 28th Very Large Data Bases Conference*, Hongkong, China, August 2002.

[4] Wen-Syan Li, Oliver Po, Wang-Pin Hsiung, K. Selçuk Candan, and Divyakant Agrawal. Engineering and Hosting Adaptive Freshness-sensitive Web Applications on Data Centers. In *Proceedings of the 12th World-Wide Web Conference*, Budapest, Hungary, May 2003.

[5] Wen-Syan Li, Wang-Pin Hsiung, Dmitri V. Kalashnikov, Radu Sion, Oliver Po, Divyakant Agrawal, and K. Selçuk Candan. Issues and Evaluations of Caching Solutions for Web Application Acceleration. In *Proceedings of the 28th Very Large Data Bases Conference*, Hongkong, China, August 2002.