The Initial Design of a Website Parallel Archiving System Using Virtualization

David Chao  
*San Francisco State University, dchao@sfsu.edu*

Sam Gill  
*San Francisco State University, sgill@sfsu.edu*

Follow this and additional works at: http://aisel.aisnet.org/amcis2009

**Recommended Citation**  
http://aisel.aisnet.org/amcis2009/551

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
The Initial Design of a Website Parallel Archiving System Using Virtualization

David Chao
San Francisco State University
dchao@sfsu.edu

Sam Gill
San Francisco State University
sgill@sfsu.edu

ABSTRACT

Many business applications are designed and organized to support business activities for a period of time and to be renewed at the turn of the period akin to the perpetual seasonal change and renewal of nature. Architecture, design, and policy changes are typically implemented in a revision of the application that supports future periods to assure smooth operation. Very often the applications supporting the previous periods need to be operational concurrently and continuously even after the application for the new period started. Parallel operation of current and previous periods’ applications may be problematic for web-based applications due to the rapid change in Internet technologies. This paper proposes a parallel archiving scheme that uses virtualization to run each period’s application in a virtual machine so that previous periods’ applications will run in parallel with the current period system and forms an easy-to-access archive for historical data.

Keywords

Website, archiving, virtualization

INTRODUCTION

Server virtualization offers the ability to run applications under different infrastructures by using one physical server to emulate many servers with virtual machines where each may run a different operating system (Chappell, 2007). The literature lists four major benefits that server virtualization offers (Microsoft, 2008): 1. Server Consolidation: Virtualization allows organizations to increase server utilization or reduce the cost of buying and managing multiple heterogeneous servers. Server consolidation lowers the total cost of ownership by decreasing hardware requirements and also lowering power, cooling, and management costs. 2. Business Continuity and Disaster Recovery: Virtualization allows organization to minimize both scheduled and unscheduled downtime. The time saved includes time lost to routine functions, such as maintenance and backup, as well as unanticipated outages. 3. Testing and Development: Virtualization allows organization to conduct testing and development in a wide variety of scenarios in a safe, self-contained environment that accurately approximates the operation of physical servers and clients with minimal disruption to the production systems. 4. Dynamic Data Center: Virtualization also allows organizations to create IT environments that can respond to dynamically changing demands for computing resources.

This paper presents a work-in-progress that focuses on a key new capability enabled by virtualization: Archiving. Traditional digital archiving emphasizes preserving the binary code of digital documents or databases. However, with the rapid change of technologies, the technologies used to create the archived data may become obsolete and preserving the data alone may be useless unless the technologies are also preserved. Virtualization can provide an environment where both the obsolete technologies and data are preserved. Preserving digital artworks is one such example where both the binary code of the artwork files and the technologies used to create the artworks need to be preserved in order to assure the rendering of the artworks (Lorie, 2002). A similar application is for a website snapshot management system (Chao and Gill, 2008) to render the snapshot of a webpage which may be created by obsolete technologies.

Extending the application of virtualization to archiving, we propose a parallel archiving scheme that uses virtualization to archive web-based applications. The scheme is based on the observation that many business applications are designed and organized to support business activities for a period of time and to be renewed at the turn of the period akin to the perpetual
seasonal change and renewal of nature. This period of time may be a year, such as applications supporting the operation of a fiscal year, or a quarter or a season such as applications supporting a university’s semester or quarter. During the operational period, the design of applications such as user interface and supporting technologies are rarely changed to assure smooth operation. Design changes are typically implemented in a revision of the application that supports future periods. When a change of period occurs, the application is reinitialized to support the new period and the application and the data of the previous period become archival.

Very often the applications supporting the previous periods need to be operational continuously even after the application for the new period started. First, they are needed to process incomplete transactions from the previous periods. Two types of updates may occur to data of previous periods: 1. Retroactive corrections: These are corrections to data of previous periods after periods ended. Retroactive corrections may include modifications and deletions. 2. Anticipatory insertions: These are data of the previous periods that have not been entered during those periods. Second, they are needed for informational purposes. Historical data are useful for decision making and their value tends to decrease as it becomes older. The data of the recent periods have higher value and are accessed more often. Allowing users to access historical data through a familiar interface is better than redirecting users to other unfamiliar archive locations. Therefore, it is beneficial for applications supporting previous periods to operate in parallel with applications supporting the current period.

Parallel operation of current and previous periods’ applications may be problematic for web-based applications. Due to the rapid change in Internet technologies, websites must keep themselves up-to-date by adopting new technologies. The infrastructure of a dynamic website typically includes the operating system, the web server, the database management system, and the server-side computer language used to create the dynamic pages, collectively known as the “stack”. Two examples of such stack are the Microsoft stack with Windows, Internet Information Service (IIS), SQL Server, and a .NET language; and the LAMP stack with Linux, Apache, MySQL, and PHP. It is possible that the infrastructure of the current period application may not be compatible with that of the previous periods.

The proposed parallel archiving scheme uses virtualization to run each period’s application and its stack in a virtual machine so that previous periods’ applications will run in parallel with the current period system and forms an easy-to-access archive for historical data. This scheme is good for applications with the following properties: 1. the applications are periodically renewed, 2. previous periods’ applications are required to be operational after new period starts, 3. the applications may change in terms of design and supporting technologies from period to period, but remain unchanged during the period. We present the initial system design in the next section.

PARALLEL ARCHIVING SYSTEM DESIGN

Figure 1 presents an overview of the parallel archiving system. The core of the system is a Virtual Host System consisting of a collection of hosts running on a host farm. Each host operates a Host Virtual Server System consisting of a collection of virtual web servers. We consider a web server as a system defined by the four components of the stack: the host operating system, O, the web service, S, the database management system, D, and the server-side computer language, L; the stack remains unchanged in a period. Each web server hosts one website that runs the version of the application, A, of a specific period. So there exists a one-web server/one-website relationship. We assume a website, WS, has a life of N periods and will be retired at the end of the Nth period. Let i denote the ith period since a website starts, then a website may have a remaining life, R, of N - i + 1 periods. The parallel archiving system eventually will have N websites operating in parallel, each with N - i + 1 periods remaining life where i ranges from 1 to N. In addition to the stack and the application it supports, each website implements a management policy, P, that governs its activities such as user management and database updates associated with business transactions. This policy may change over time. For instance, the current period website may initially accept all types of database updates including insertions, modifications and deletions that occur in the business environment. As the website aging, certain updates may be restricted and eventually the website may become a read-only archive. Another example for a policy change may be a business rule associated with the application A, such as the Grade Point Average (GPA) for a student that would allow listing in an honor role. Therefore, websites can be distinguished with these attributes, WS(Stack(O, S, D, L), R, A, P).

Users of the system initiate business transactions and submit queries related to a specific period. The Seamless Login Module enables transparent login to each website regardless of underlying server structure. It presents users with an easy-access interface where websites are accessible by push buttons. Once in the website, data of business transactions and queries are handled by the Website Data Manager. Since there are N websites operating in parallel, these Website Data Managers are in fact managing the archived data of these N periods.
The function of the DS/DW Integrator is to gather and prepare data for an enterprise’s decision support system and data warehouse. Recognizing the value of business intelligence to an enterprise, today’s e-Commerce systems typically integrate decision support data acquisition module in the system design (Nickerson, 2002). Because of the heterogeneous nature of the Virtual Host System, the DS/DW Integrator must have the ability to work with heterogeneous data sources.

A website will eventually retire when it completes its N life periods and will be removed from the Parallel Archiving System. The website with its supporting technologies and database is a valuable historical resource of an enterprise. Historical data may be useful in supporting applications that require historical data, such as applications that perform analyses to study certain trends in the study subject, or answering questions about website content in the past for audit and compliance purposes. Websites may also be required to preserve historical data due to government or organizational policies. The Website Archiving Integrator implements the enterprise’s policy in archiving websites.

There are many ways to archive websites. One popular practice is periodically creating date-time stamped read-only copies of the website. This type of archive is unable to satisfy a user’s request for historical data between archives. Website snapshot management system is one solution to manage website’s historical data. A framework of this system is proposed in (Chao and Gill, 2008). The framework incorporates a Web Server Virtualization system that is able to support websites using different stacks and contains a Website Changes Tracking System for tracking and preserving website changes. Such a system is capable of recreating webpage snapshots of every published webpage including their code and rendition upon users’ requests. Adopting the framework of website snapshot management system will enable the transition of a retired website from the Parallel Archiving System to the enterprise’s archiving system.

An example of the Virtual Host System

![Diagram of Virtual Host System](image)
Figure 2 gives an example of the Virtual Host System assuming the website is renewed annually with a new stack. The 2006 and 2007 websites are resided on Physical Host 1 and each website is supported by a virtual machine, VM1 and VM 2 respectively. Starting from the year 2008 the website is moved to Physical Host 2 and currently the Virtual Host System consists of two physical hosts.

**Alternative one-web server/many-website relationship**

The Parallel Archiving System assumes a one-web server/one-website relationship. If, however, the stack used to support a website does not change from one period to the next, then a web server can actually support more than one website. In the above example, if the stack supporting 2008 and 2009 websites has not changed, then the virtual machine VM 3 can support both the 2008 and 2009 websites as illustrated in Figure 3.

![Figure 3: Use one virtual machine to support many websites using the same stack.](image)

**Determining the value of N**

The Parallel Archiving System assumes a website has a life of N periods. The actual value of N may be influenced by many factors: 1. An organization’s policy in accepting delayed changes. For instance, an organization may have a policy of closing a database to updates after n periods after the end of its designated period. 2. The number of delayed updates. The actual number of delayed updates typically decreases as the website ages. A cut-off point may be determined where updates may be more efficiently maintained offline. 3. The value of information. As discussed earlier, the data of the recent periods have higher value and are accessed more often. A cut-off point may be determined where the costs of maintaining the website online are higher than the value of the information.

**CONTINUE RESEARCH AND CONCLUSION**

As discussed earlier, the proposed scheme is good for applications that are periodically renewed and upgraded in design and supporting technologies and are required keeping the previous periods’ applications operational in parallel with the current period application. An example of an application with these characteristics is a university’s learning management system supporting faculty and students that may be renewed every academic period while allowing users to access previous periods’ data. Many accounting systems also have similar requirements. We continue research on identifying applications that may benefit from the proposed scheme.

**References**