https://doi.org/10.48009/1_iis_2015_226-231

THE DESIGN OF A CLOUD-BASED WEBSITE PARALLEL ARCHIVING SYSTEM

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. Design 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 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. Cloud computing provides a solution to this problem with the capability of offering virtual servers with user-specified configurations. This paper proposes a parallel archiving scheme that uses virtual server to run each period's application in a cloud platform 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: Cloud Computing, Website Archiving, Virtualization

INTRODUCTION

Cloud computing delivers computing resources "as a service" to clients via Internet [11]. Examples of such services include software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). With the SaaS, users subscribe to business applications running on vender's server and pay a subscription fee. The PaaS delivers a computing platform including programming language and execution environment where developers can develop their software solutions on a cloud platform. And the IaaS offers virtual machines that meet users' requirements to run applications. Using PaaS and IaaS users are capable of developing and deploying their own business applications. Cloud computing is a growing industry [9]. A recent Goldman Sachs study projects that spending on cloud computing infrastructure and platforms will grow at a 30% growth rate from 2013 through 2018, and by 2017, 35% of new applications will use cloud computing to deliver services [4].

Cloud computing does not require businesses to invest heavily on IT infrastructure out-front. The major benefits of cloud computing includes [3]: 1. Efficiency: Cloud computing allows businesses to rapidly deploy applications due to lower requirements for initial investment on technologies and maintenance of the infrastructure. 2. Agility and innovation: Businesses can react to the business environment faster and test innovative services before full-scale implementation. 3. Cost Savings: Businesses only pay for the computing capabilities they use without purchasing the infrastructure, and can determine the computing capabilities dynamically based on the demand of the applications. 4. Increased scalability: Businesses can rapidly "scale up" their computing capabilities, and rapidly release those services to quickly "scale in." One such example is the Cloud Service offered by Microsoft Azure [7]. A cloud service is a multi-tier web application in Azure, consisting of Web Roles which are dedicated virtual machines for hosting front-end web applications, and Worker Roles which are dedicated virtual machines that run time-consuming tasks sent from a web role asynchronously in the back-end. And the number of Web Roles and Worker Roles can be configured dynamically based on the business needs. Other benefits without elaborating include easy access from anywhere, disaster recovery, provision of mobile applications, etc.

Cloud computing platform vendors such as Amazon Web Services [1], Google Cloud Platform [6] and Microsoft Azure [7] offer virtual machines as IaaS and typically offer these services regarding virtual machines: 1.They offer an extensive list of pre-configured virtual machine images encompassing a wide range choice of operating systems, database management systems and development technologies. 2. Users are able to create their own virtual machine images with user-specified configurations. 3. Users can upload and deploy their own virtual machine images. 4. Users can activate and deactivate the virtual machines as needed. 5. Users can take snapshots of the virtual machines which is a file-based representation of the state of a virtual machine along with the database at a given time. Virtual machine snapshots help to back up or archive virtual machines. 6. Users can capture an image of a running virtual machine as a template to create other virtual machines. These services let users to customize virtual machines to their requirements, control the costs of deploying the virtual machines and create backups of the virtual machines.

This paper presents a scheme that focuses on a key new cloud computing 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 [7]. A similar application is for a website snapshot management system [2] to render the snapshot of a webpage which may be created by obsolete technologies.

Extending the cloud computing 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. 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 and updated data will be available for analysis and decision making. 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 virtual servers of a cloud platform 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.

PARALLELL 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 virtual host servers running on a cloud platform. Each virtual host server operates 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 web language, L; the stack remains unchanged in a period. Each web server hosts one website that runs the application of a specific period. So there exists a one-host server/many-web server relationship, and 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 *i*th 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. Websites can be distinguished with these attributes, WS(Stack(O, S, D, L), R).

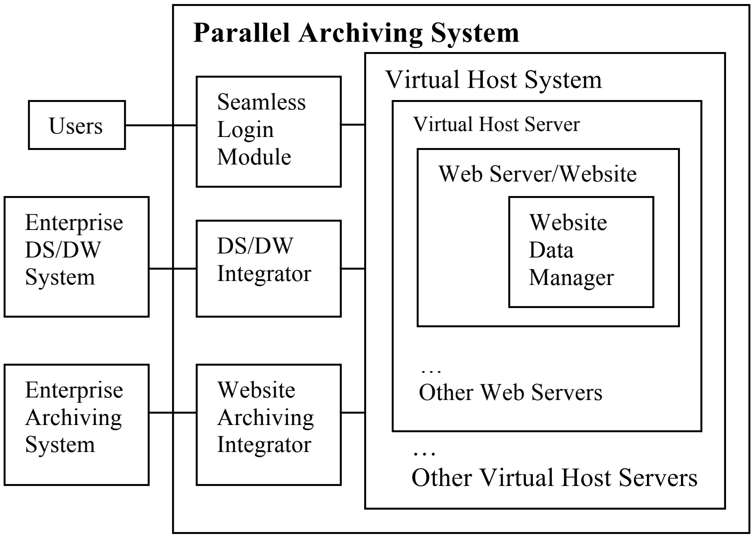

Figure 1. An Overview of the Parallel Archiving System

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 [10]. 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. One popular practice is periodically creating date-time stamped read-only copies of the website. In cloud computing this can be done by creating the virtual machine snapshots.

An Example of the Virtual Host System

Figure 2 gives an example of the Virtual Host System assuming the website has been running for four years and renewed every two years with a new stack. The Parallel Archiving System assumes a one-virtual host server/one-website relationship. Each year the website is running on a separate virtual host server. If, however, the stack used to support a website does not change from one period to the next, then a virtual host server can actually support more than one website. In the above example, if the stack supporting 2014 and 2015 websites has not changed, then the virtual machine VM 3 can support both the 2014 and 2015 websites as illustrated in Figure 3.

The arguments supporting the one-virtual host server/one-website relationship are: 1. assuring the current operational system running smoothly without interruptions from other activities is the primary objective of an information system. 2. It will ease the management of the parallel archiving system. As discussed earlier, a typical cloud computing platform offers services to take snapshot and create image of a virtual machine. With each period's system running on a separate virtual machine, it will be easier to create snapshot and the image of the virtual machine for that period. 3. There exists unbalanced demand for each period. The demand for the previous periods will decrease as time elapsed. Since a virtual machine can be activated/deactivated by the client of a cloud computing platform, to save the costs of running the parallel archiving system, an organization may activate the previous period systems on an on-demand basis.

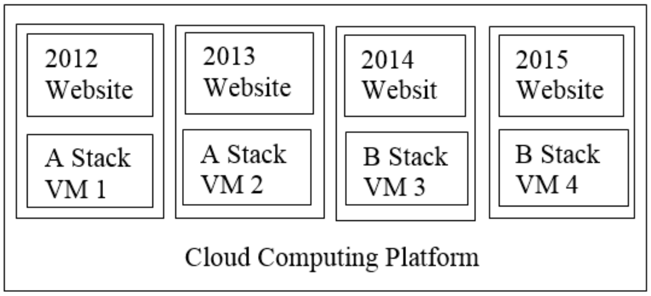

Figure 2: An example of the Virtual Host System

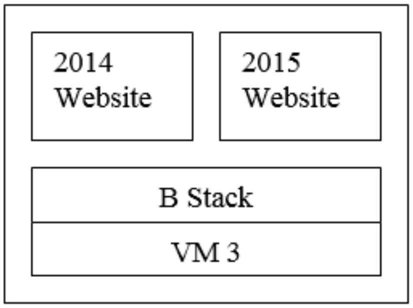

Figure 3: Use One Virtual Host Server to Support Many Websites Using the Same Stack

PARALLEL ARCHIVING MANAGEMENT

In this section we discuss two issues in the management of parallel archiving: 1. determining the length of a website's life in the parallel archiving system from its initiation to retirement. 2. Managing the activating/deactivating of a website.

Determining the Length of a Website's Life

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.

Managing the Activating/Deactivating of a Website

The cost of running N virtual machine images concurrently in the parallel archiving can be high. Cloud computing platform vendors typically let users to activate or deactivate a virtual machine as needed to save cost. One way of achieving this is using the management tools provided by the platform vendors. For instance, Microsoft Azure offers management portal to manage virtual machines using easy-to-use graphic interface [8]. This scenario is best for virtual machines with a predefined operating schedule. Another approach is using virtual machine "dehydrating" and "rehydrating" techniques. Dehydration is the process of serializing the state of a long running business process into a database. Rehydration is the reverse of this process: deserializing the last running state of a long running business process from the database [5]. Dehydration is used to minimize the cost of a virtual machine by reducing its utilization of cloud resources. With the parallel archiving, the Virtual Host System might determine that a virtual machine has been idle for a relatively long period of time. It calculates thresholds to determine how long it will wait for various actions to take place, and if those thresholds are exceeded, it dehydrates the virtual machine. This can occur under several circumstances, for instance, when the virtual machine is waiting to receive a message, and the wait is longer than a threshold determined by the host. The host can be triggered to rehydrate a virtual machine, restores its state, and runs it from the point where it left off.

CONCLUSIONS

This paper presents a scheme for web applications that are periodically renewed, frequently changed in design and supporting technologies from period to period, and are required to keep 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. The scheme runs each period's application in a virtual machine so that the technologies supporting the period's application are preserved with the application. At the turn of a new period, the new application with its design changes which possibly supported by new technologies is run with a new virtual machine, and the previous period's application and its virtual machine becomes an operational and active archive. The proposed scheme is a cloud-based system to take advantage of the cost-effectiveness of the cloud computing. To further reducing the cost of running multiple virtual machines, the system may activate/deactivate virtual machines based on demand. The adoption rate of cloud computing continues to grow and more information system projects are expected to run on a cloud platform. The proposed scheme helps managing the specific type of information systems we identified on the cloud.

REFERENCES

1. Amazon Web Services. Available: http://aws.amazon.com
2. Chao, D., & Gill, S. (2008) A framework for a website snapshot management system. *Issues in Information Systems, IX*(2), 279-285.
3. Cloud.CIO.Gov. Benefits of cloud computing. Available: http://cloud.cio.gov/topics/benefits-cloud-computing
4. Columbus, L. (2015). Roundup of cloud computing forecasts and market estimates. Available: http://www.forbes.com/sites/louiscolumbus/2015/01/24/roundup-of-cloud-computing-forecasts-and-market-estimates-2015/
5. DOTNETROBERT.COM. (2013). Hydrating and dehydrating workflows. Available: http://www.dotnetrobert.com/node/159
6. Google Cloud Platform. Available: https://cloud.google.com/
7. Lorie, R. A. (2002). The UVC: A method for preserving digital documents - Proof of Concept. IBM/KB Long-term Preservation Study Report Series, IBM Global Services Netherlands.

8. Microsoft Azure. Available: http://azure.microsoft.com/en-us
9. Miller, P. (2015) Cloud Computing market trends in 2015. Available: http://research.gigaom.com/report/cloud-computing-market-trends-in-2015
10. Nickerson, R. (2002) An E-Commerce System Model. *Proceedings of the 8th America's conference on Information Systems*, 310-316
11. Wikipedia. Cloud computing. Available: http://en.wikipedia.org/wiki/Cloud_computing