AN ADAPTIVE REPLICATION FRAMEWORK FOR IMPROVING THE QOS OF WEB SERVICES

Marwa F. Mohamed[1], Hany F. ElYamany[1] Mohamed K. Hussien[1],
Nashwa M. Yhiea[2] and Hamed M. Nassar[1]
[1]*Computer Sciences Department, Faculty of Computers and Informatics, Suez Canal University, Ismailia, Egypt*
[2]*Mathematics Department, Faculty of Science, Suez Canal University, Ismailia, Egypt*

Keywords: Adaptive Replication, Quality of Service (QoS), Dynamic Load Balancing and Web Services.

Abstract: This paper presents an adaptive framework for managing dynamic replication of Web services in a distributed environment including the Service-Oriented Architecture (SOA) environment. The framework aims to improve the web services availability and to reduce the response time by supporting an automatic replication of the consumed web services according to environment changing conditions that might occur at the services provider side such as failure or increasing loading. For example, if one service or server fails, the framework replicates automatically the consumed service on another particular selected server based on some Service-Level Agreements (SLAs) including their performance and availability. Further, the framework balances the incoming requests using Round Robin a load-balancing algorithm. Moreover, the proposed framework is designed to predict the load of the involved candidate servers within the replication process through utilizing a statistical regression technique.

1 INTRODUCTION

The Web service technology has gained a great momentum in both academic and industry over the last decades. A web service is a well-defined software component which provides a specific functionality over the Internet. Web services are published, described, discovered and executed using several standard protocols including WSDL for service description, SOAP for intercommunication, and UDDI for publishing and discovering the abstracted and loose coupled web services (Papazoglou, 2007). The quality of service (QoS) is an essential issue for managing the access agreement among the services providers and consumers when selecting and composing the distributed service-based applications. In particular, the Service Level Agreement (SLA) is the predefined expected QoS attributes standard including response time and availability for controlling and monitoring the interoperability between the services provider and consumer. Specifically, response time is the time required to complete a web service request. Availability is the probability that the system will be able to respond to the client request at any times (May, Schmidt and Thomas, 2009).

Replication techniques can generally support web services availability through providing multiple replicas of web services; for example, if one service fails, other copied services will be possibly initiated in order to answer the client requests. Moreover, it can basically reduce the response time through broadcasting client requests among the available replicas, particularly during runtime (Silva and Mendonca, 2004).

In this paper, we will present a framework for managing an adaptive replication of Web services in a distributed environment such as the Service-Oriented Architecture (SOA) environment. The suggested adaptive replication framework supports an automatic replication of different web services according to the changing that might occur at the services provider side or within the SLA elements of the hosting servers. The proposed framework is focusing on only replicating the web services, not replicating the client requests. Basically, the suggested framework issues the necessary decisions when a service fails or overloads, and selects the suitable servers that will be needed to replicate web services on it automatically.

The rest of the paper is organized as follows. Section 2 shows the related work. Section 3

F. Mohamed M., F. ElYamany H., K. Hussien M., M. Yhiea N. and M. Nassar H..
AN ADAPTIVE REPLICATION FRAMEWORK FOR IMPROVING THE QOS OF WEB SERVICES.
DOI: 10.5220/0003926102830286
In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science* (CLOSER-2012), pages 283-286
ISBN: 978-989-8565-05-1

introduces a description of the suggested adaptive replication framework. Section 4 describes the case study and experimental results. Finally, conclusions and future work are presented in section 5.

2 RELATED WORK

Several researchers have classified the process of web service replication in different ways, for example by determining which replica is handling the incoming requests and selecting the optimal server for hosting a replica. In (Salas, Perez-sorrosal, Patiño-martínez and Jiménez-peris, 2006) the authors classified the web service replication process according to only determining which replica could process the incoming requests. Basically, they defined three different techniques called active, semi-active and passive replication. In, (May et al., 2009) the replication process has been classified based on three techniques named as parallel, serial and composite strategies. In (Zheng and Lyu, 2008), a number of replication strategies have been proposed based on adding time redundancy to a combination of active and passive replication in order to facilitate the process of selecting an appropriate strategy for the different business case.

In (Ge and Zhang, 2010), a stochastic Perti net is used to model the performance of composite services as well as the environment in order to achieve adaptive reselection during runtime. The suggested framework accomplishes an adaptive reselection of a higher priority service when the failure occurs on the other services or an environment. The service priority order is based on availability of the network and the host machine; and also the execution time and reliability of the services. The main drawback of this framework is that it does not consider some other crucial metrics within the replica reselection process such as the service or host load.

In (Yau, Goyal and Yao, 2005), an adaptive replica selection framework has been proposed. The framework considers both the characteristics of the replica hosting environment and the consumer requirements. The reselection process is based on calculating a set of hybrid metrics including service failure, host failure and dynamic load as well as the customer business-value degree. However, the framework has initially assumed that a fixed number of hosts are available during runtime for the replication process.

To this end, the main ideas of the proposed adaptive service replication frameworks based on selecting a web server for hosting the needed replica from a predetermined set of available hosts (Ge and Zhang, 2010; Keidl, Seltzsam and Kemper, 2003). However, they didn't take into account to predict the changing load conditions of the service and the web server during runtime.

3 THE ADAPTIVE REPLICATION FRAMEWORK

This section introduces an adaptive framework for adaptive replication management of Web services in a SOA environment. The suggested adaptive replication framework supports an automatic replication of different web services according to the changing that might occur at the services provider side or the SLA elements of the hosting servers such as capacity and availability. Generally, the framework adaptively improves the availability of the published services, and improves the Quality of services (QoS) of the web services such as the performance during runtime.

3.1 The Proposed Adaptive Replication Framework

Figure 1 shows the suggested adaptive replication framework. The framework involves three basic layers: the clients, the replication middleware and servers layer. The clients layer represents the published services which consumers can access. The servers layer contains the candidate hosts that might be selected for holding possible replicas. Finally, the replication middleware layer is located at the services provider side.

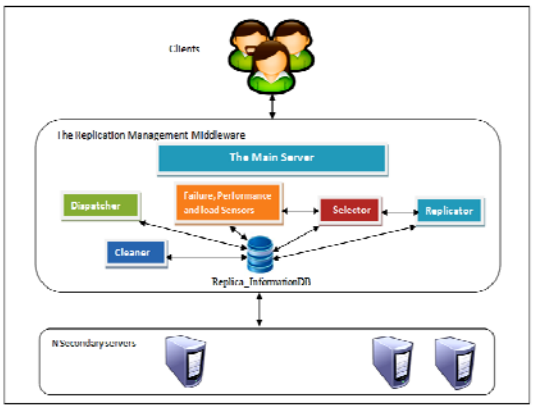

Figure 1: The adaptive replication framework.

3.2 The Main Server Components

Sensors: This service is for monitoring all possible

changes that can occur on the secondary servers, such as: *Failure sensor* checks the secondary servers status (available or failure). *Performance sensor* collects information about the secondary servers as the memory capacity, hard disk capacity and processor type. Load sensor gets the CPU usage of the secondary servers.

Selector: This service analyzes the information received from the sensors. Then, it sorts the list of secondary servers from the best to the worst according to the following metrics as: *server Availability, server load and server capabilities.*

Dispatcher: This service is the bridge between the client and the secondary servers as it does the following: *Responder*, it forwards requests from the clients to the selected secondary server. Then, it forwards response from the secondary server to the clients. *Load balancer*, it executes the dynamic load balancing process that demonstrated in (Alakeel, 2010).

Replicator: it replicates web services on the selected secondary server. In case of the master replica that process the client request fails or becomes overloaded.

Cleaner: is acting as the garbage collector which deletes unused web services from the secondary servers.

4 CASE STUDY AND EXPERIMENTAL RESULTS

One of the framework objectives is how to take advantage of all available possible advantages in the execution environment of any institution as much as possible. For example, in a university environment; the faculties differ in the number of student and the date of grade announcement. We can see that one faculty server is overloaded while others are less loaded at the same time. Therefore, we thought about how to take an advantage of the less loaded servers when other servers become overloaded or failure by proposing an adaptive replication framework. In order to evaluate the proposed framework, we present three different scenarios:

Static Scenario: The server of each faculty works separately (or independently) and there is no relation in between. For a student to check his mark online; he can just type his ID and presses submit. If the faculty server fails, the student will not be able to check his mark. If the faculty server is overloaded, the student will have to wait a bit longer to get his mark.

Adaptive Replication Scenario: The university server manages the web services replication process. If the university server detects that one service or faculty server fails, the framework automatically replicates the consumed service on another particular selected faculty server based on certain Service-Level Agreements (SLAs) including their performance and availability. Moreover, if the university server detects that a particular service or server is overload, the framework automatically replicates the consumed service on another particular selected faculty server then used the Round Robin load balancing algorithm (WebsiteGear, 2004) in order to balance the requests between the replicas.

Adaptive with Load Prediction Scenario: adding to the previous case, a prediction ability using the following factors: *The linear regression model* (Montgomery and Runger, 2007, chap 11), and *Scheduling* which is the regular server schedule that determines when it might be loaded.

The following experiments were conducted to evaluate the performance of the framework using the previous scenarios (static, adaptive and adaptive with load prediction). The test environment is VMware simulation consists of 6 servers: 1 MS, 4 SSs and 1 DNS. The capabilities of these servers are established in Table 1. The experiments were conducted using ApacheBench Version 2.0.40-dev <$Revision: 1.146 $>apache-2.0

Table 1: The capabilities of simulation servers.

Capabilities	MS	SSs	DNS
HD	40 GB	40 GB	40 GB
CPU	Intel® core ™ Num of core using by VMware 1		
Memory	500 MB	384 MB	384 MB
OS	Microsoft windows server 2003		
Language	PHP scripting language		-

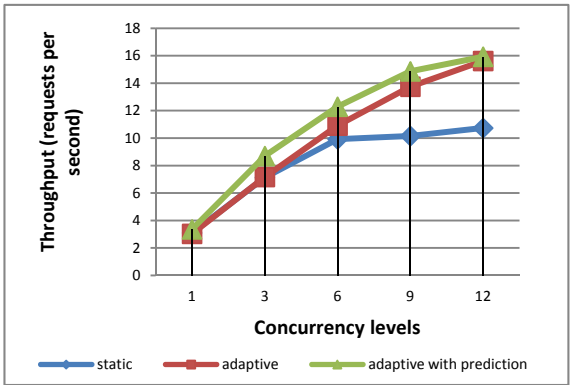

Figure 2: The average Throughput of running 7500 requests.

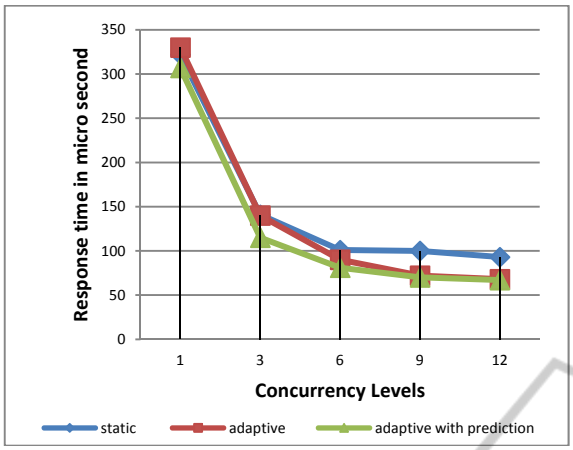

Figure 3: The average Response time of running 7500 requests.

In Figures 2 and 3, the performance order of the previous approaches from the best to the worst is adaptive with load prediction property, adaptive replication and static approaches. Further, when the concurrency level is increased, the gap between static and adaptive is increased; and also the gap between adaptive and adaptive with prediction calculation is decreased except when concurrency level is equal to 1.

5 CONCLUSIONS AND FUTURE WORK

In this paper, a framework for an adaptive replication process is introduced. The framework includes several components for managing the replication process in addition to a particular component which is structured for predicting the future load on the servers that host the original copies of the web services. Further, a case study is demonstrated as well as the experimental results are provided. The experiment examined the suggested framework within three different scenarios: static, adaptive and adaptive with load prediction property and measured the replication process performance and throughput. The shown outcomes established that the framework is working more efficiently when it runs within the adaptive with prediction property scenario.

For the future work, we plan to apply the adaptive replication framework on composite web services. Furthermore, we aim to apply partial adaptive replication; the web services may encapsulate more than one business operation and the load on each operation is not equal. Hence, we plan to apply an adaptive replication only on the overload business operation. Finally, we also plan to enhance the prediction load algorithm by using another robust statistical prediction technique for handling and broadcasting properly all data sets types including the linear and nonlinear sets.

REFERENCES

Ali M. Alakeel, 2010. A Guide to Dynamic Load Balancing in Distributed Computer Systems. In *International Journal of Computer Science and Network Security*.

Douglas C. Montgomery and George C. Runger, 2007. *Applied Statistics and Probability for Engineers*, John Wiley & Sons, Inc., USA, 4th edition.

Jorge Salas, Francisco Perez-sorrosal, Marta Patiño-martínez and Ricardo Jiménez-peris, 2006. WS-Replication: a Framework for Highly Available Web Services. In *15th International Conference on the World Wide Web*. Edinburgh, Scotland.

Jose A. Silva and Nabor d. Mendonca, 2004. Dynamic Invocation of Replicated Web Services. In *the Proceedings of the WebMedia & LA-Web 2004 Joint Conference 10th Brazilian Symposium on Multimedia and the Web 2nd Latin American Web Congress (LA-Webmedia'04)*. Ribeirao, Preto, Brazil.

Liang Ge and Bin Zhang, 2010. A Modeling Approach on Self-Adaptive Composite Services. In *International Conference on Multimedia Information Networking and Security*. Nanjing, Jiangsu China.

Markus Keidl, Stefan Seltzsam and Alfons Kemper, 2003. Reliable Web Service Execution and Deployment in Dynamic Environments. In *4thInternational Workshop on Technologies for E-Services (TES)*. Berlin, Germany.

Michael Papazoglou, 2007. *Web Services: Principles and Technology*, Pearson Education Limited, England.

Nicholas R. May, Heinz W. Schmidt and Ian E. Thomas, 2009. Service Redundancy Strategies in Service-Oriented Architectures. In *Software Engineering and Advanced Application*. Patras, Greece. IEEE Comp. Soc.

Stephen S. Yau, Gaurav Goyal and Yisheng Yao, 2005. Replication for Adaptive Responsiveness in Service-Oriented Systems. In *International Conference on Quality Software (QSIC'05)*. IEEE Comp. IEEE Computer Society Washington, DC, USA ©2005.

Zibin Zheng and Michael R. Lyu, 2008. A distributed replication strategy evaluation and selection framework for fault tolerant web services. In *IEEE International Conference on Web Services*. Beijing, China.

WebsiteGear, Server Load Balancing: Algorithms, 2004. Retrieved December 10, 2011, from http://content.websitegear.com/article/load_balance_types.htm.