ISSN 2278-3083

International Journal of Science and Applied Information Technology (IJSAIT), Vol.2 , No.2, Pages : 15-19 (2013)
Special Issue of ICA4E 2013 - Held during 15-16 April, 2013 in Singapore

Timely Consistency of Replicated Unified Course Contents

Omar Abdullah Batarfi
King Abdulaziz University, Saudi Arabia, obatarfi@kau.edu.sa

ABSTRACT

In academia, it may be the case that group of instructors teaching a unified course and they share course content in limited period of time. In academia, consistency not only influences the performance and availability of the systems, but also influences overall teaching quality because students should get the same information regardless of teacher. The main contradiction of consistency is modification shared course contents by instructors in separate. This paper proposes a collaborative framework of interacting agents, capable of concurrent execution of consistency checks. This framework provides a continuous consistency for replicas of the course content. In next section, the related work for consistency detection is presented. Then, it discusses the detailed design of agent inconsistency detection module, the core components of this framework, which can detect inconsistency among shared course content in a timely manner. The proposed agent inconsistency detection module is evaluated via prototype of consistency management system, which shows that this module can significantly reduce the time to detect inconsistency among shared course content without adding much maintenance cost.

Keywords: Consistency, Agent, Maintain, Course content

INTRODUCTION

In academia, a same course is taught by more than one instructor for different reasons. First, number of students who would like to register same course exceeds the capacity of a class, so creating more than one section is the appropriate action. Second teaching the course by different instructors enriches teaching and meets the different desires of students on the teaching style they like. In such situation, conventional module require replicating course contents equal to the number of classes of the same course to increase availability and performance for instructor and student who accessing course contents [1]. The conventional module doesn't offer flexibility to instructor to alter course material that is assigned to his class; moreover if there a necessary modification to the course content, this information must be passed over to all instructors to adapt this modification to their course content. Even when the modification information was passed over to all instructors (here after is called nodes), there is a chance that a connection for one of the node is down which will block the modification process to cover all the nodes. Consistency management in such environment has become critical because poor consistency for replicated services results in poor QoS and even monetary losses for e-business applications [1]. In this type of module the overhead of consistency maintenance will equal to the number of replicas. None conventional module is based on cooperation between nodes where they are distributed becomes usual [2]. But most importantly, focusing on the ability to check and maintain consistency between all distributed nodes that have pre-shared course content throughout their modification or addition. Inconsistency and conflict may occur between distributed content because of the multiplicity of participants and lecturers [3]. Agent can facilitate consistency in offering an infrastructure to manage the consistency checks in a distributed environment. Agents exchange information based on cooperation and interaction at the base architectural level.

The proposed system contains a base interaction layer that facilitates consistency management service between pre-shared distributed course contents. Moreover, it allows instructor a freedom to alter and update the course content without any constrain. Proposed system must achieve a high performance to monitor the consistency and based on that can discover inconsistencies in a timely manner, and can solved when detection of inconsistencies. If the proposed system cannot achieve this, it will affect the quality of service

RELATED WORK

Consistency becomes even more important in cloud computing, in such environment conventional databases will cause bottleneck because of the high of data that is transferred between clients and data storages. [4] Tackles consistency in cloud and emphasize that maintain high consistency implies high cost per transaction and in some situations

ISSN 2278-3083

International Journal of Science and Applied Information Technology (IJSAIT), Vol.2 , No.2, Pages : 15-19 (2013)
Special Issue of ICA4E 2013 - Held during 15-16 April, 2013 in Singapore

reduced availability. [4] is proposed an analytical model using MM1 queuing for data consistency on private cloud storage system by applying discarding probability of update request to eliminate any conflict. In the proposed system, readability is also improved after updating besides improving consistency of storage on the cloud environment.

In [5] a software agent is used to provide a solution to maintain consistency in a distributed setting. The proposed system is an architecture that has two interacting levels, software agent co-operation level and consistency management level. The architecture facilitates collaboration and interaction between distributed architecture components through agents to support concurrent execution of consistency checks of heterogeneous distributed documents.

ARCHITECTURE DESCRIPTION

Course content firstly is downloaded to an instructor computer equal to the number of instructors who participate in the teaching. All course contents are compatible versions. During the process of teaching over the semester, which takes nearly three months, could happen one of instructor adds new information or deleting from the course content on his computer. Moreover, changing can occur with more than one instructors involved in teaching the same material. The proposed architecture described in the Fig. 1 is to handle such problem of different course contents that were originally unified.

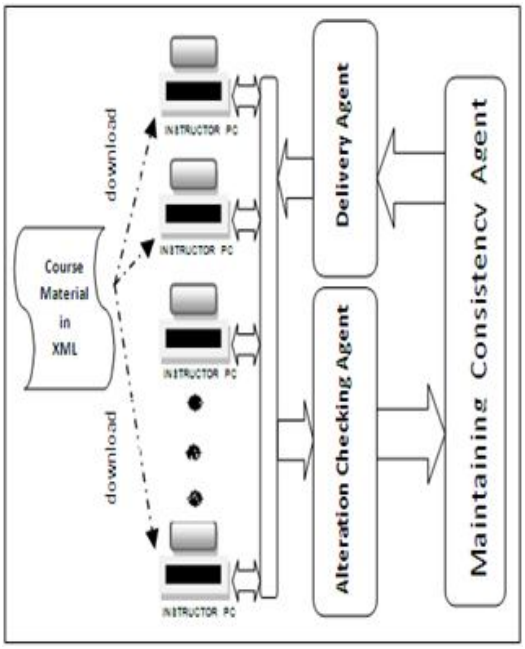

Fig 1: Consistency management architecture

Agents

An agent is computer program that can act independently to perform the user's operations. The main operation performed by agent is the perceiving surrounding environment to accomplish pre-assigned tasks. A gent perceives its environment through sensors and behaves through effectors [6]. Operations performed by the agent ranging from basic analytical tools to the complex processes of knowledge-based. The agent can analyze the situation, make decision, move to the appropriate node, interact and negotiate with other agents as tasked. Finally, the agent then transfers the results to the system and its users [7].

There are three kinds of software agents are [8]:

- Personal agents behave on behalf of a user; therefore, they react directly with the user to present his/her personality or character. Moreover, personal agents monitor the user's activities; also they learn the user's style and preferences. Personal agents are used to simplify and automate tasks.
- Mobile agents have mobility nature because they travel to remote nodes and collect the required information or to accomplish pre-assigned tasks and return with the results.

Collaborative agents work in group; therefore, they communicate and negotiate between them. They represent users, organizations and services.

A. Consistency Checking Agent

This agent provides a framework to check for any modification event that violates consistency relations between multiple replicas [9] of the same course contents. The consistency framework supports a cross-check homogeneous structured of each course contents.

B. Maintain Consistency Agent

This agent is located in the lower layer and its role mainly to satisfy the consistency between the replicas of the course content. After Consistency check agent discovered modification in any replica of the course content, it forwards this copy to the maintain consistency agent. Maintain consistency agent will compare the last delivered course content with the identified altered replica of the course content to find out the modified part. After maintain consistency agent accomplish the process of checking and allocate the modification, it forwards the altered part delimited by the section tags to delivery agent.

ISSN 2278-3083

International Journal of Science and Applied Information Technology (IJSAIT), Vol.2 , No.2, Pages : 15-19 (2013)
Special Issue of ICA4E 2013 - Held during 15-16 April, 2013 in Singapore

C. Delivery Agent

Delivery agent has to finalize the consistency process based on the following protocol (hereafter called the "consistency protocol"):

- Must ensure that other instructors do not already cover the topic that has been altered. If this condition is not satisfied the modified part will be rejected.
- If the first condition is satisfied, the delivery agent must confirm the acceptance of all instructors for modifying their copies of the course contents. If one instructor do not accept, the modification will be rejected
- If the modified part rejected, the deliver agent will inform the maintain consistency agent about the rejection. The maintaining consistency agent will resend the last saved copy to the instructor that issued the rejected modification.
- If the maintaining consistency agent does not receive a rejection notification from delivery agent, it saves the new copy as the last saved copy of the course content.

Course content representation

To simplify design, course content will be represented by using the eXtensible Markup Language (XML) language to ease the process of comparison when checking for modification. XML is a simple text-based format for representing structured information. It was derived from a Standard Generalized Markup Language (SGML). XML is a metalanguage which supports you to define a document markup language and its structure. You can use XML to create your own markup language, which includes a set of rules and tags that describe the information that fits your needs, for example, name, title, address, and zip code. You can define this markup language in a document type definition (DTD) which works as a standard way to describe your information.

XML depicts a class of data objects called XML documents and describes part of the behavior of computer programs that address these data objects. XML documents form a tree structure which provides an effective method of access to the data by parsing XML and traversing the XML tree [10].

SEQUENCE DIAGRAM

The sequence diagram in the figure 2 depicts interactions between different agents. When a modification has done by an instructor, alteration checking consistency agent first throws event getsignsave() to retrieve the signature for saved course content. Alteration checking consistency agent before starts the checking process will also retrieve the signature value for each copy on instructor PCs by executing event getsigncopy(). Thereafter, alteration checking consistency agent throws event comparesign(). If the two signature values are differ, alteration checking consistency agent will forward a copy of the course content of the instructor PC to maintain consistency agent. Maintain consistency agent throws event findmodification() to find out the difference between the retrieved copy and the saved one. Consequently, maintain consistency agent sends modification to delivery agent. Finally delivery agent will apply the consistency protocol. In the case that the consistency protocol allows for applying the modification, delivery agent will replace the altered part on all instructor PCs.

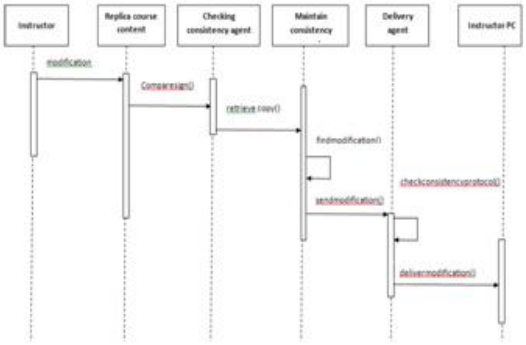

Fig 2: Sequence diagrams model

SYSTEM IMPLEMENTATION AND TESTING

To test our stem, we applied an experiment on four computers and these computers are the same. They are Dell computers with processor speed is 1 GHZ and the memory size is 1 GB. All four computers were connected by LAN 10-100 Mbps and we named them as Oma1 to Oma4 with addresses 192.168.0.10 to 192.168.0.13.

We use in the experiment the Wireshark program. It is used to monitor the traffic running on the four computers [11]. Moreover, we use Aglet to execute agent mobility; Aglet was developed at the IBM Tokyo Research Laboratory. The Aglet Software Development Kit (ASDK) is a framework and environment for developing and executing mobile agents [12]. The Aglet platform has different operations such as create, dispatch, retract, clone and dispose the agent. Our system uses three agents, the Alteration Checking Agent, Delivery Agent and Maintaining Consistency Agent.

We dispatched the Alteration Checking Agent after any modification on any replica of the course content. We use Secure Hash Algorithm (SHA-1) to create the signature of the course content; moreover, SHA-1 is used to ease the detection process in the case of any modification took place on any replica of the course content.

In the experiment, all the four computers have the XML representation of the following course content:

> Web applications suffer more than their share of security attacks. Here's why. Websites and the applications that exist on them are in some sense the virtual front door of all corporations and organizations. Growth of the Web since 1993 has been astounding, outpacing even the adoption of the television and electricity in terms of speed of widespread adoption.

The hash signature for the above course content is: 2a40af4f8ee48c8a994dc714f995f6aa885563f1

After all instructors downloaded a copy of the course content, this copy is saved as last delivered course content.

We did modify the course content on the Oma3 to be as the following:

> Web applications suffer more than their share of security attacks. Websites and the applications that exist on them are in some sense the virtual front door of all corporations and organizations. Growth of the Web since 1993.

The hash signature for the modified course content is: ff9f05b4bcc2c9ece86d48d4845387fc3e6df4c9

After dispatched the Alteration Checking Agent immediately detect that the course content has modified on Oma3. After the Maintaining Consistency Agent find the modification after compared the two signatures (signature of the last delivered course content and the signature of the retrieved and modified course content).

This experiment demonstrates how our proposed architecture supports maintaining timely consistency of replicas of course content.

There are number of issues that have not applied in the experiment to make the experiment simple, these issues are:

- We manually dispatch the agents to accomplish the required task.
- After Maintaining Consistency Agent found the modified part, we made the Maintaining Consistency Agent to deliver the whole course content to the Delivery Agent instead deliver the modified part(s).
- We disable the consistency protocol; therefore, the Delivery Agent role to be restricted to deliver the modified course content to other computers (Oma1, Oma2 and Oma4).

CONCLUSION

Satisfying consistency in conventional manner will affect efficiency and if it is applied in academia will affect the learning process. Consistency is required in academia especially in the situation where more than one instructor teaches a unified course and they sharing course content in limited period of time. This paper is proposed a consistency management system that able to maintaining consistency between replicas of the same course content. The proposed system based on a collaborative framework of interacting agents, capable of concurrent execution of consistency checks. This framework provides a continuous consistency for replicas of the course content in timely manner. The detailed design of the detection system was evaluated by experiment. Result shows that the proposed consistency management system able to detect inconsistency among replicas of the course content and maintain the existence of consistency.

REFERENCES

[1] Y. L. a. H. J. Yijun Lu, "Adaptive Consistency Guarantees for Large-Scale Replicated Services," in *Networking, Architecture, and Storage, 2008. NAS '08. International Conference on*, 2008.

[2] H. J. a. D. Feng, "An Efficient, Low-Cost Inconsistency Detection Framework for Data and Service Sharing in an Internet-Scale System," in *e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on*, 2005.

[3] D. J. P. G. Parker, G. Rudisin, A. Stoughton, B. Walker, E. Walton, J. Chow, D. Edwards, S. Kiser and C. Kline, "Detection of Mutual

Inconsistency in Distributed Systems," in *Software Engineering, IEEE Transactions on*, 1983.

[4] Y. N. Aye, "Data Consistency on Private Cloud Storage," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS),* pp. 101-105, 2012.

[5] D. S. Anthony Finkelstein, "Software Agent Architecture for Consistency Management in Distributed Documents," in *6th International Conference on Information Systems Analysis and Synthesis*, Orlando, 2000.

[6] G. T. N. A. Surjeet Dalal, "Designing CBR-BDI Agent for implementing Supply Chain system," *International Journal of Engineering ,* vol. 3, no. 1, pp. 1288-1292, 2013.

[7] D. K. S. N. D. R. G. C. Robert Gray, Mobile agents for mobile computing, Hyderabad: University Press, 2004, p. 17.

[8] M. L. Griss, "Software Agents as Next Generation Software Components," in *Component-Based Software Engineering: Putting the Pieces Together*, Addison-Wesley, 2001.

[9] J. W. G. a. C. S. C. W. I. M. S. E. Clarence, "A Modern Approach to Distributed Artificial Intelligence.," *G. Weiss, MIT PRess,* 1999.

[10] W3C, "W3C Recommendation," [Online]. Available: http://www.w3.org/TR/REC-xml/.

[11] Wireshark. [Online]. Available: http://www.wireshark.org/. [Accessed 1 2 2013].

[12] A. D. Group, "Aglet Software Development Kit," 12 March 2009. [Online]. [Accessed 12 1 2013].