ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Load Balancing and Coordination of Web Services in Cloud

[I]Lingajothi V, [II]Umarani Srikanth G
[I, II]S.A Engineering College, Chennai, India

Abstract

Web services requests are larger than requests encoded with binary protocol. The extra size is really an issue in either low speed connections or extremely busy connections. Traditional Byzantine Fault Tolerance (BFT) replication mechanism can be used to mitigate the threats to the Web Service Business Activities (WS-BA) coordination service. The present lightweight BFT algorithms mitigate the threats and avoid the runtime overhead associated with traditional BFT algorithm. Some of the problem found in present system is less security, low performance and less data availability. The cloud coordination service should be replicated the service in order to achieve high availability and data integrity of the service and also provide the cloud storage this cloud storage provides as a service to storage consumers. Also, the load balancing algorithm is used for cloud computing. It benefits to achieve a high user satisfaction and resource consumption ratio, therefore improving the overall performance and resource utility of the system.

Keyword

Cloud computing, Load balancing, web services, Distributed computing, Byzantine Fault Tolerance (BFT)

I. Introduction

The paradigm of distributed computing was born when the emergence of computer networks. The web applications are divided into two parts. The first part is client initiating a distributed activity and the second part is server, it carry out the distributed activity from the client. The bottlenecks should be minimized in this decentralized host by using distributed computing that is the workload should be distributed across multiple systems. This is used to provide flexibility to the unknown centralized hosts. Some of the issue should be arises in this two-tier architecture so overcome this issues, newly introducing the three-tier architecture.

This new introducing architecture is called as Middle tier. This tier contains the business logic and dealing with the data and so separate the application part into presentation part. It also provides the system Remote Procedure Call (RPC) is used to communicate between the distributed applications. Now present, the web service consists of three type of middleware. These types are Distributed Component Object Model (DCOM), COBRA and RMI.

There is a rising expectation to the potential capability of cloud systems as an IT infrastructure that will help to create new values [1]. If cloud systems are to beapplied towider areas, such as the core business of enterprises, e-administration and other social infrastructure services, it will be required to reflect certain end-to-end service value cover not only the computing resources of cloud systems but also the networks involved, and to chance requests for consistency as well as compliance, governance etc. and needs for power saving.

When services are providing by a single cloud system (some time called as single cloud), an unanticipated level of overload (traffic flow from the Internet), or a normal tragedy may demand to complement reserved resources [2]. However, reserved resources believed by a single cloud are commonly restricted, and capability of a single cloud to stay services may reach limits. Appropriate to create the cloud system to be capable to stay to fulfill difficulties for guaranteed quality of service convenience and performance, even in such cases, it is necessary to establish that cloud systems supplement each other for example to obtain resources from other cloud systems by collaborating with other cloud system (called as inter-cloud) linked via broadband networks.

So far, single cloud provider has been organizing cloud system based on each provider's exclusive conditions. Regularization the interfaces among systems will support inter-clouds computing, which can be responsible for more trustworthy and higher-quality cloud services than a single cloud system. As a result, it is estimated to stimulate the happenings of culture as a whole, including business and administrative activities.

Load balancing is one of the methods of computer networking for allocating the workload across multiple system or computers such as computer cluster. The aim of Load balancing is to increase resource use, increase throughput, reduce response time, and reduce overload of any one of the resources in computer cluster [3]. Using single components alternatively to use multiple components with load balancing may be optimizing reliability. Server Farm is one of the commonly used applications in load balancing that is to provide a service from multiple servers in single internet services.

The collection of computer server is known as server farm or server clusters. Server farm have backup servers so it can easily recover the data or service even after failure of primary server. Cluster computing is one of the applications of server farm. The cluster computing is established to increase the performance and improve the availability over a single computer. It is a set of computer that is loosely or tightly connected together so that the information should be viewed in single system.

II. The Web Service Business Activity

A Web Service is a method that communicates between client and server application over the Internet using open protocol. The web services are creating more and more business activities and then transforming into distributed computing environment. The Web Service Business Activity (WS-BA) specification was used to create and to enable the web service to participate site in long running application. This WS-BA specification is loosely coupled transaction to the web service. This type of transaction is mostly used for modeling business application. For example, work flow system, the execution engine is responsible only for the coordinating the participants in the business logic as per rule defined in business process.

The business activity is a set of initiator, coordinator and one or more participants. The initiator is used to start the business activities and also terminate the business activities. The coordinator components consist of registration service, activation service, coordinator service, completion service [4]. When participants invoke to the web service, initiator create a web service business activity for the particular request from the participants. Then the coordinator component check whether the created business activity is propagated to the web service and also check whether it's terminated predefined policy or not.

The coordination service of the business activities should be replicated because to achieve high availability of services and high integrity of the service. The coordination services are tightly coupled to the web services. So the lack of well-defined interface between the application and coordinator requires coordination service that is built into the application. The participant have trust the coordination services. The coordination service contains all the state notifications and records all the state changes. Using coordination context, the initiator increases the business activity to the participants according their request. In business application the workflow engine provide the WS-BA coordination services. This coordination services will enable the web service environment without workflow engines to create a web service, to coordinate between initiator and participate in long running application.

An initiator of a web service business activity is to communicate with the coordinator of the web service business activity by using the WS-BA specification. This specification does not specific any standard way for communication. So it has the coordinator replication for Byzantine Fault Tolerance (BFT).The BFT framework support byzantine fault model in client server applications. Byzantine Fault Tolerance (BFT) replication is used to reduce danger of attack from the web application with high security.

The Byzantine fault is an arbitrary fault that caused by both omission failures and commission failures during the execution of the program or algorithm by a distributed system. The omission failure means break down failures, failing to receive a request or failing to send a response [5]. The commission failure occurs when processing a request inaccurately, mortifying local state and/or sending an inconsistent response to request. The system may respond in an unpredictable way when a Byzantine failure has occurred, unless it is designed to have Byzantine Fault Tolerance (BFT).

Byzantine Fault Tolerance is a sub field of failures tolerance. The main objective of BFT is that protect against Byzantine failures in which components of a system fails does not dependence on any procedure or algorithm. BFT requires 3f+1 based on replication. f refers to faulty. All server replicas reach agreement on the Byzantine faulty replicas and clients such as achieved by ensuring BFT. This agreement referred as Byzantine agreement.

A. WS-Business Activity Protocol

The coordination services are specified by the WS-BA standard. This coordination consists of several atomic transaction defines in long running transaction. The processing of request should be taken some more time that is this coordination type processing requires human interaction. This means that roll-back function cannot be realized by a un-do. Alternatively a un-do need an advanced recompense mechanism involved in business logic. For example, in booking application the cancelling of booking need this advanced recompense mechanism that include the billing of cancellation fees.

The business activity coordination type has the following attribute characterize transactions:

1. During transaction time, all the state changes should be recorded.
2. The coordinator and the participant should have the recognition state because the message has the acknowledged.
3. It does not allow the combine message because each message should be send individually.

The WS-BA permit the nesting transaction that is nesting business activity may be including another business activity or transaction. This nesting transaction is used to preventing and handling the error that precede a failure from the parent transaction. The best example is supply of product components. If one of the nested transaction should get fails means immediately another transaction may be able to serve the service.

B. The Byzantine Agreement with Authentication

Already known that assumption required for authentication mechanism that use "signed messages" in Byzantine agreement protocols sometimes get fails if these assumptions may be violated.

A State Machine approach is a basic requirement for replicated processors in fault tolerant systems to achieve agreement to values of single-source data. The Oral message is known as new introducing protocols. This oral message protocol is used to add the authentication with additional resilience is obtained and also does not require the assumptions about the security of authentication.

The oral messages assumptions are as following:

A1: Every message should be sent between the Non faulty processors and is correctly delivery.

A2: The receiver knows that who send the messages.

A3: To detect the absence of a message.

Here, Number of participants called as "Processor". The communication media that is communication media between one processor to another processor called as "Transmitter". Receiving message is called as "Receiver".

All the Byzantine agreement protocols consider of two rounds. In first round, the value should be sends to all other processors from the transmitter. In second round, the processor exchanges the received values among themselves. This exchange is used to detect inconsistencies of receiving values.

III. Related Work

A. Existing System

In the present of WS-BA, the web application and internet should become important things in now days [16], [17], the WS-BA is used to interact between multiple enterprises. The WS-Trust specification attentions on the expenditure of security tokens and permits within the framework of web services, whereas, in this paper, to motivate on the high-availability and high integrity features of load balancing of web service business activities.

BFT has been of great investigation importance for the last few years. The inspiring study of this field is that of Lamport et al. [21], who established that four replicas are necessary to accept one Byzantine faulty replica (three replicas do not suffice) and, more generally, that 3f þ 1 replicas are necessary to accept f Byzantine faulty replicas. Other important influences to Byzantine agreement/ consensus around the same time were prepared by Dolev et al. Consequently, several sets of investigators established Byzantine fault-tolerant group communication systems [6], [19], [23], [26], [27], for standard distributed applications.

All of above systems should send a message in total order at the destinations. In compare, lightweight BFT algorithm for trustworthy coordination of WS-BA sends message in source order, i.e., it sends the sender sent order, not send total order at the destinations. It organizes so by take advantage of the state model

of the WS-BA specification. The PBFT agreement algorithm [5] of Castro and Liskov describes that among one of the replicas as the primary and the other replicas taking as the backup.
The PBFT algorithm includes three stages to order a request. This stages are pre-prepare phase, prepare phase, and commit phase. It also contains an opinion change algorithm that concludes a new primary, if the current primary is judged to be faulty. Kihlstrom et al. [20] designated a BFT consensus algorithm. Their BFT algorithm for making compromise activates in three rounds, and uses a coordinator to select the compromise value and Byzantine fault indicators to discover faults in the coordinator and other processes. The application of BFT techniques to transactional systems was first informed by Mohan et al. [20]. They develop the two-phase commit protocol by accomplishment Byzantine agreement on the consequence of an atomic transaction among all of the nodes in a root cluster.
In recent times, we reentered this problem and suggested a more well-organized solution for atomic transactions by controlling the Byzantine agreement to only the coordinator replicas. Garcia-Molina et al. [15] have realistic Byzantine agreement to distributed database systems, in particular, for distributing transactions to processing nodes. However coordination of atomic transactions tolerates some resemblance to coordination of business activities, there are more number of substantial changes.
In atomic transactions, the coordinator and the participants are closely coupled. Any participant can individually terminate a transaction. Furthermore, the coordinator can select on the consequence of the transaction, established on the votes together in the two-phase commit protocol. Even Though, in business activities, the outcome is complete especially by the initiator permitting to the business logic to faulty participant potency be able to use some impact on, but not resolve on, the outcome of the business activity.
These features require the practice of a lightweight BFT solution for the load balancing of WS-BA, as described in this paper. The application of BFT techniques to web services has been described in [24]. Although the results suggested could be used to keep the coordination services of WS-BA beside Byzantine faults, they are without cause costly. By allowing for the state model of the WS-BA coordination services, lightweight BFT algorithm added only one additional round of message exchange for each request on the coordinator abort from that of two or more additional rounds of message exchange required in [24].

B. Proposed System

The initiator searches the web service for an offer, from which it receives a reply, and then it searches the data integrity web service from which it receives a reply. Next, it invokes the coordinator service to create token for the participants. Using the invitation token, it uploads a data and downloads data. On receiving a token request from the initiator, the web service registers with the registration service; similarly the data integrity web service registers with the registration service. The initiator sends a Complete Participants message to the coordinator service. The coordinator sends an acknowledgment to the initiator as soon as it has sent a complete message to the participant services of the web service and data integrity web services without waiting to receive their completed messages.

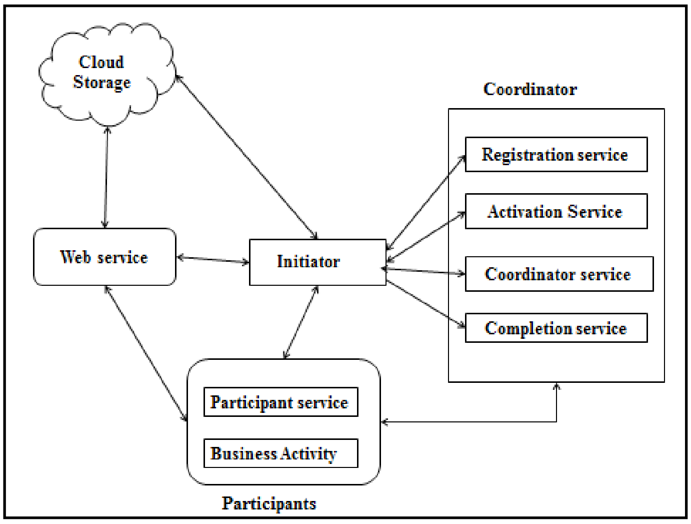

Fig. 1.1: Architecture Diagram

The Activation service creates a Coordinator object and a web service context for all web services. Basically, the Activation service haves like a factory object that creates Coordinator objects. The Activation service is used for all data process in business activity. It is provided by a single object, which is replicated in BFT framework. When web service is activated, a Coordinator object is created. The Coordinator object provides the Registration service, the Completion service and the Coordinator service. The web service context contains a unique storage id and an endpoint reference for the Registration service, and is included in all request Messages sent during the cloud storage. The Coordinator object is replicated in our BFT framework shown in fig 1.1
On the Coordinator-side, the services comprise:

- Registration service: The Registration service allows the Participants and the Initiator to register their endpoint references.
- Completion service: The Completion service allows the Initiator to signal the start of the distributed commit.
- Coordinator service: The Coordinator service runs the 2PC protocol, which ensures atomic commitment of the distributed cloud storage.

IV. Conclusion

Several companies, for example Amazon, Google, Microsoft, and Apple, are now proposing cloud services. Some of them might offer coordination services used for web services in the cloud. Such coordination services could permit smaller companies to propose combined web services to offer value-added services to their customers without having to invest heavily in computing infrastructures.
The Coordinator object is replicated in our BFT framework. The cloud coordination service should be replicated the service in order to achieve high availability and data integrity of the service and also provide the cloud storage this cloud storage provides as a service to storage consumers. Also, the load balancing algorithm is used for cloud computing. It benefits to achieve a high user satisfaction and resource consumption ratio, therefore improving the overall performance and resource utility of the system.

References

[1] *www.gictf.jp/doc/GICTF_Whitepaper_20100809.pdf*
[2] *Apache WSS4J, http://ws.apache.org/wss4j/, 2013.*
[3] *ZhongXu, Rong Huang,(2009)"Performance Study of*

Load Balanacing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report.

[4] C. Attiya, D. Dolev, and J. Gil, "Asynchronous Byzantine Consensus," Proc. ACM Symp.Principles of distributed Computing, pp. 119-133, 1984.

[5] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," ACM Trans. Computer Systems, vol. 20, no. 4, pp. 398-461, Nov. 2002.

[6] D. Davis, A. Karmarkar, G. Pilz, S. Winkler, and U. Yalcinalp, Web Services Reliable Messaging (WS-ReliableMessaging) Version 1.1, OASIS Standard, Jan. 2008.

[7] D. Dolev, "An Efficient Algorithm for Byzantine Agreement without Authentication," Information and Control, vol. 52, no. 3, pp. 257-274, Mar. 1982.

[8] D. Dolev and H.R. Strong, "Authenticated Algorithms for Byzantine Agreement," SIAM J. Computing, vol. 12, pp. 656-666, 1983.

[9] H. Erven, H. Hicker, C. Huemer, and M. Zapletal, "The Web Services-BusinessActivity-Initiator (WS-BA-I) Protocol: An Extension to the Web Services-BusinessActivity Specification," Proc. IEEE Int'l Conf. Web Services, pp. 216-224, July 2007.

[10] T. Freund and M. Little, Web Services Business Activity, Version 1.1, OASIS Standard, Apr. 2007.

[11] H.Garcia-Molina, F. Pittelli, and S. Davidson, "Applications of Byzantine Agreement in Database Systems," ACM Trans. Database Systems, vol. 11, no. 1, pp. 27-47, 1986.

[12] Y. Gil and D. Artz, "A Survey of Trust in Computer Science and the Semantic Web," J. Web Semantics, vol. 5, pp. 58-71, 2007.

[13] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," ACM Trans. Programming Languages and Systems, vol. 4, no. 3, pp. 382-401, July 1982.

[14] D. Malkhi and M. Reiter, "Secure and Scalable Replication in Phalanx," Proc. IEEE 17th Symp. Reliable Distributed Systems, pp. 51-58, 1998.

[15] Y. Fang, F. Wang, and J. Ge, "A Task cheduling Algorithm Based on Load Balancing in Cloud Computing", WebInformation Systems and Mining, Lecture Notes in Computer Science, Vol. 6318, 2010, pages 271-277.

[16] K. Iwasa, J. Durand, T. Rutt, M. Peel, S. Kunisetty, and D. Bunting, WS-Reliability 1.1, OASIS Standard, Nov. 2004.

[17] K.P. Kihlstrom, L.E. Moser, and P.M. Melliar-Smith, "The SecureRing Group Communication System," ACM Trans. Information and System Security, vol. 4, no. 4, pp. 371-406, Nov. 2001.

[18] K.P. Kihlstrom, L.E. Moser, and P.M. Melliar-Smith, "Byzantine Fault Detectors for Solving Consensus," Computer J., vol. 46, no. 1,pp. 16-35, 2003.

[19] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," ACM Trans. Programming Languages and Systems, vol. 4, no. 3, pp. 382-401, July 1982.

[20] D. Malkhi and M. Reiter, "Secure and Scalable Replication in Phalanx," Proc. IEEE 17th Symp. Reliable Distributed Systems, pp. 51-58, 1998.

[21] C. Mohan, R. Strong, and S. Finkelstein, "Method for Distributed Transaction Commit and Recovery Using Byzantine Agreement within Clusters of Processors," Proc. ACM Symp.Principles of Distributed Computing, pp. 89-103, 1983.

[22] L.E. Moser and P.M. Melliar-Smith, "Byzantine-Resistant Total Ordering Algorithms," J. Information and Computation, vol. 150, pp. 75-111, 1999.

[23] L.E. Moser, P.M. Melliar-Smith, and N. Narasimhan, "The SecureGroup Group Communication System," Proc. IEEE Information Survivability Conf., vol. 2, pp. 256-279, Jan. 2000.

[24] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker, Web Services Security: SOAP Message Security 1.1, OASIS Standard, Nov. 2006.

Ms. V. Lingajothi is currently pursuing M.E CSE in S.A. Engineering College, Chennai. She has completed UG degree from Pavendar Bharathidasan College, Trichy. Her areas of interests are Image processing, Network security and Cloud computing.

Dr.G.Umarani Srikanth is currently working as a Professor and Head, Department of PG Studies, S.A.Engineering College, Chennai, India. She has 17 years of teaching experience. Her areas of interests include Compilers, Theory of Computation, Data Structures and Soft Computing.