On Demand Logical Resource Replication Scheme as a Service

Vardhan Manu[1], Goel Akhil[1], Verma Abhinav[1],
Mishra Shakti[2], and Kushwaha D.S.[1]

[1] Computer Science and Engineering Department, MNNIT Allahabad
Allahabad, India
{rcs1002,cs084084,cs084076,dsk}@mnnit.ac.in
http://www.mnnit.ac.in
[2] Institute for Development & Research in Banking Technology, Hyderabad, India
smishra@idrbt.ac.in

Abstract. Logical resource replication in cloud environment along with the issue of transparency and its impact on overall system performance has been rarely focused. This work proposes an On-demand Logical Resource Replication Scheme (OLRRS) for file replication. OLRRS approach provides migration, access and performance transparency to the system, thereby ensuring the migration decisions about the files. It is also responsible for replicating the file, from one peer server to the other peer server, when the total number of request, on a peer server, for transferring a file reaches the threshold value. The scheme is simulated on JAVA platform and tested on a private cloud. A comparative study of the pro- posed approach with the Request Reply Acknowledgement (RRA) and Request Reply (RR) protocol is presented, showing the significant reduction by 37.5% to 58%, in terms of total number of messages exchanged for replication.

Keywords: Replication, Logical resource (LR), Service, Private cloud, IaaS.

1 Introduction

Cloud computing is the future of technology, with its application distributed in every field. Cloud computing eliminates the need of having efficient hardware resources and infrastructure requirements by providing the resources on Pay-as-you-go basis. All that is required is a machine with an enabled web browser. Cloud can be deployed in three ways viz., Public, Private [8] and Hybrid cloud. When it comes to cloud delivery architecture, there exist three delivery models, viz., SaaS [7], PaaS [7] and IaaS [7]. IaaS is the most widely used delivery architecture, as it provides the provision for processing, storage, networks and other fundamental computing resources. To increase the system reliability, some fault tolerant mechanism should be used, so that the system keeps functioning in case of failure. One such method is replication, which replicates the critical software components, so that if one of them fails, the others can be used to continue. On demand logical resource replication scheme (OLRRS), provides an on-demand replication of logical resources (files, service), with a view to

V.V. Das and J. Stephen (Eds.): CNC 2012, LNICST 108, pp. 31–36, 2012.
© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2012

minimize the network resource utilization by minimizing the message exchange overhead, to speed up the overall system performance.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 introduces the proposed architecture in a private cloud environment. Section 4 discusses the simulation and results with a case study for OLRRS approach and then the result is concluded in Section 5 followed by the future work.

2 Related Work

Replication in cloud environment is done to achieve high availability of resources. Resources can be replicated dynamically or on-demand to minimize the overhead of maintaining the consistency of the replicated files, to some extent. Similar kind of work is carried out by the authors in distributed environment considering the various performance issues that can arise and affect the overall system performance. Richard T. Hurley and Soon Aun Yeap [1] have proposed file replication and migration policy, by which the total mean response time for a requested file at a particular site can be reduced. Sometimes instead of using file replication mechanism, it is preferred to use process migration to achieve better system performance and minimum utilization of network resources. A similar approach has been taken by Anna Hac [3]. Author has proposed the file replication, migration and process migration techniques based on the workload of local and remote host. Concern with cloud computing is, how to deploy the application on cloud and in what manner should the deployed application, be delivered as a service. Pengzhi Xu, et. al. [5] presented a prototype named PosixCloud, which is designed to deliver general purpose cloud storage via standard POSIX interface and provides support for the traditional applications which are based on standard file system interface. To facilitate logical resource (file, service) replication in cloud environment, a replication mechanism is required which aims at facilitating replication considering the issues related with replication in cloud and distributed computing. Wei-Tek Tsai, et. al. [6] has proposed a replication scheme for services. Whenever there is an increase in number of request a service can handle, additional resources are acquired by replicating the service.

3 Proposed Approach

On-demand replication of logical resources (file or service), is achieved by replicating resources, when the number of requests for a specific resource reaches the threshold value. The proposed architecture is discussed below:

3.1 Architecture

The architecture proposed, implements, on demand logical resource replication scheme (OLRRS). Figure 1 shows the set of peer servers called the File Replication Servers (FRS), responsible for providing the replication service in the cloud environment. Based on the number of request received for a particular file by the FRS, the resource (File) is replicated when the total number of request reaches a threshold value.

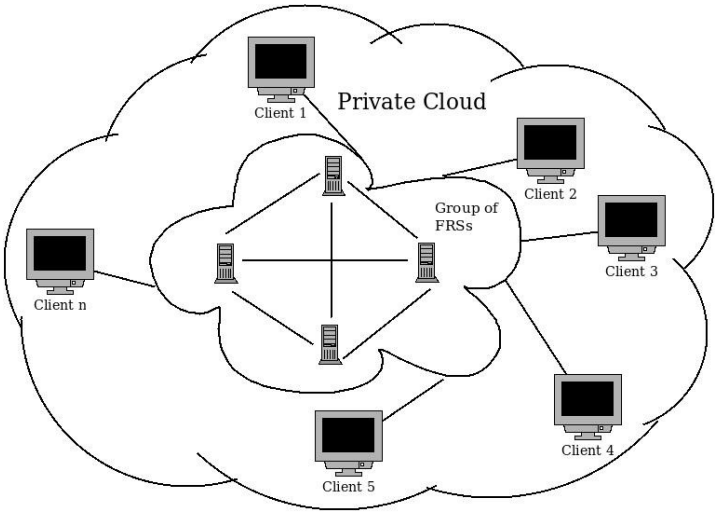

Fig. 1. Architecture of private cloud

3.2 Data Structures

Fig. 1 shows the group of File Replicating Servers (FRSs), which represents the peer severs for the proposed architecture. Two data structures are maintained by every peer server:

Request count
Peer server information

Request count: Every peer server will maintain the information about the files requested by the clients. This helps in deciding, when to create the replica of the requested file. It is created when the total number of requests on a server reaches a threshold value. The information will be stored under the following heads: FileID, Filename, Request Count and Metadata.

Peer server information: This maintains the IP and PORT of the peer server having the replica of the requested logical resource as follow: SERVER IP and SERVER PORT.

3.3 Message Definitions for OLRRS Approach

OLRRS approach has four types of messages viz., M_1, M_2, M_3, and M_4. Functionality implemented by each message is described below:

M_1: This message will contain the request for GET/PUT operation.
M_2: This message responds to the request based on the servers' current state.
M_3: Copy the file from source to destination.
M_4: Provides the details to the client, about the peer server on which the logical resource is being replicated.

3.4 RRA and RR Protocol [9]

RRA [9] protocol use three messages for completing a request viz., request message, reply message and acknowledgement message. RR [9] protocol use two messages for completing a request viz., request message and the reply message which also serves as the acknowledgement for the request message.

3.5 Strengths and Limitations of the Proposed Methodology

Strengths: OLRRS approach provides access, migration and performance transparency to the system. Also this approach controls file replication redundancy to certain extent.

Limitations: False sharing can occur with coarse grained granularity, thus reducing the parallelism. But it will not affect the overall performance of the system.

4 Simulation and Results

The proposed model is simulated on JAVA platform. OLRRS approach is com- pared with Request Reply Acknowledgement (RRA) [9] and Request Reply (RR) protocol [9]. It outperforms RRA and RR protocol in terms of on-demand replication. Given below are the details and possible cases for better understanding of OLRRS approach:

Case 1: The first case is, when the client C_j sends a GET request to the server S_o and receive the LR_i from it. Numbers of messages exchanged are as follow: M_1: $C_j \rightarrow S_o$; M_2: $S_o \rightarrow C_j$; M_3: $S_o \rightarrow C_j$; Total number of messages exchanged: 3

Case 2: S_o reports the OVERLOAD condition to the client C_j. Number of messages exchanged are: M_1: $C_j \rightarrow S_o$; M_2: $S_o \rightarrow C_j$; M_4: $S_o \rightarrow C_j$; Total number of messages exchanged: 3

Case 3: Replication of LR_i from S_o to S_r and client C_j is informed about the replicated LRi. Number of messages exchanged are as follow: M_1: $C_j \rightarrow S_o$; M_1: $S_o \rightarrow S_r$; M_4: $S_o \rightarrow C_j$; M_2: $S_o \rightarrow C_j$; M_3: $S_o \rightarrow S_r$; Total number of messages exchanged: 5

Table 1 shows the comparison of the OLRRS approach in terms of messages exchanged per request with the existing RR protocol and RRA protocol. In RR and RRA protocol, there is no routine mechanism for getting the IP address of the peer server on which the file is replicated, as compared to OLRRS approach. Here STATUS message, in case of RR and RRA protocol, will provide the IP address of peer server containing the copy of replicated resource.

Fig. 2 shows the comparison between OLRRS, RR and RRA protocol. It shows the total number of messages exchanged per resource per client for successfully completing an operation (GET, PUT, OVERLOAD and REPLICATE). It clearly shows that OLRRS approach runs well for REPLICATE and OVERLOAD operation. It outperforms the other two protocols, when used for these (REPLICATE and OVERLOAD) operations. In terms of total number of messages exchanged, OLRRS approach shows significant performance improvement, for OVERLOAD and REPLICATE operation, as compared to GET/PUT.

Table 1. Number of messages exchanged per request

OPERATION	OLRRS	RR	RRA
GET	3	2	3
PUT	2	2	3
OVERLOAD	3	$M_1(2)$ + STATUS(2) = 4	$M_1(3)$ + STATUS(3) = 6
REPLICATE	5	$M_1(2)$ + STATUS(2) + $M_1(2)$ + $M_3(2)$ = 8	$M_1(3)$ + STATUS(3) + $M_1(3)$ + $M_3(3)$ = 12

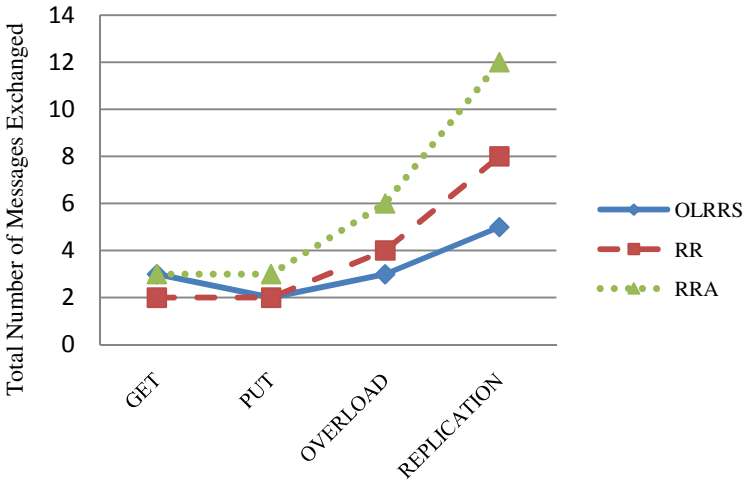

Fig. 2. Graph showing comparison between OLRRS, RR and RRA protocol

5 Conclusion

On demand logical resource replication scheme (OLRRS), proposed in this paper, aims at implementing the replication mechanism, with minimum number of messages required to complete the replication operation, thus minimizing the network resource utilization. OLRRS approach ensures various types of transparencies viz., migration, access and performance. Migration decisions about the file movement are made by OLRRS approach, without any user intervention. It is responsible for replicating the file, from one peer server to the other peer server, when the total number of request, on a peer server, for transferring a file reaches the threshold value. Thus, enhances the performance of the peer servers, which constitutes to the system performance.

In the future, we will, enhance OLRRS approach. In particular, proposing a consistency mechanism, increasing reliability and security setup so that the purpose of building OLRRS approach remains intact.

References

1. Hurley, R.T., Yeap, S.A.: File migration and file replication: a symbiotic relationship. IEEE Trans. on Parallel and Distributed Systems 7(6), 578–586 (1996)
2. Mei, A., Mancini, L.V., Jajodia, S.: Secure Dynamic Fragment and Replica Allocation in Large-Scale Distributed File System. IEEE Trans. on Parallel and Distributed Systems 14(9), 885–896 (2003)
3. Hac, A.: A Distributed Algorithm for Performance Improvement Through File Replication, File Migration, and Process Migration. IEEE Trans. on Software Engg. 15(2), 1459–1470 (1989)
4. Xiong, K., Perros, H.: Service Performance and Analysis in Cloud Computing. In: World Conference on Services-I, pp. 693–700 (2009)
5. Xu, P., Zheng, W., Wu, Y., Huang, X., Xu, C.: Enabling cloud storage to support traditional applications. In: 5th Annual ChinaGrid Conference, pp. 167–172 (2010)
6. Tsai, W.-T., Zhong, P., Elston, J., Bai, X., Chen, Y.: Service Replication with MapReduce in Clouds. In: 10th International Symp. on Autonomous Decentralized Systems (ISADS), pp. 381–388 (2011)
7. Cloud computing - A premier The Internet protocol Journal 12, `http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_12-3/123_cloud1.html` (accessed on October 22, 2011)
8. Zheng, L., Hu, Y., Yang, C.: Design and Research on Private Cloud Computing Architecture to Support Smart Grid. In: International Conf. on Intelligent Human-Machine Systems and Cybernetics (IHMSC), August 26-27, pp. 159–161 (2011)
9. Spector, A.Z.: Performing remote operation efficiently on a local computer Network. Communications of the ACM 25(4), 246–259 (1982)