IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 3 (Mar. - Apr. 2013), PP 50 -53
www.iosrjournals.org

An Database Query Service In Large Scale Reliable Storage System For Automatic Reconfiguration

K.Vinothini [1], Ms. B. Amutha[2], B.Renganathan[3]
[1]*M.E. (Cse), SrinivasanEngg College, Perambalur.*
[2]*Ap/ Cse, SrinivasanEngg College, Perambalur.*
[3]*M.E. (Cse), SrinivasanEngg College, Perambalur*

Abstract: *Large scale reliable storage system with dynamic membership access is currently facing lot of problems like human configuration errors and it is strong enough only for the static set of replica...An membership service and dBQS carried out the dynamic membership access and dBQS is a novel distributed hash table differ from the usual DHT by providing byzantine fault tolerant and strong semantics. We develop two heuristic algorithm and automatic reconfiguration technique to maintain and carried out those membership services. Experimental result shows that the membership service works well and the membership service is able to manage large system and the cost to maintain the service is low.*

Index terms- *DHT, dBQS,Byzantine fault tolerant*

I. Introduction

Now a days, Byzantine fault tolerant system assumes only static set of replicas and it is having limitations in handling the reconfiguration. Generally network security is to prevent and monitor unauthorized access,misuse and modification of data. Here, we use membership service under network security concept to maintain and carried out the storage system service. The system is classified into two parts as membership service (MS) and dBQS. One is to monitor the membership changes and the other is to automatically reconfigure the system respectively.

Automatic reconfiguration is mainly done in MS to avoid human configuration errors and dBQS(data base query services) is differ from usual DHT by providing strong semantics and it is an byzantine fault tolerant by extending the existing byzantine quorum protocol.MS is itself an byzantine fault tolerant and it produces configuration periodically rather than changes an membership.

The MS may need to move to new group of servers in case of reconfiguration for this we allow the system to operate correctly, even there is an failure bound in the original group of Ms relicas.an Byzantine-fault-tolerant group is designed for this reconfiguration and our results shows that the MS is able to manage the storage system of replicas with low cost.

II.Related Work

A group membership protocol enables processes in a distributed system to agree on a group of processes that are currently operational. Membership protocols are a core component of many distributed system and have proved to be fundamental for maintaining availability and consistency in distributed applications. We present am membership protocol for asynchronous distributed system that tolerates the malicious corruption of group members. Our protocols ensure that correct members control and consistency observe changes to the group membership, provided that in each instance of the group membership, fewer than one-third of the members are corrupted or fail benignly.

The protocol semantics shows that Uniqueness is common to much membership protocol and also stronger than ordering semantics of various protocols.

Uniqueness is that if p_i and p_j are correct and v_i^xandv_j^x are defined, **then $v_i^x=v_j^x$**.

The protocol has many potential applications in secure systems and, in particular, is a central component of a toolkit for constructing secure and fault-tolerant distributed services that we have implemented. The Membership protocol implementation for an atomic broadcast a set of techniques that make such a tool kit practical.

Membership protocol is suitable for used in distributed system in which some process may be corrupted by a malicious intruder. This protocol achieves is an asynchronous system, provided that in each instance of the group membership, fewer than one-third of the group members are corrupted or fail. Although the protocol can be used to remove them from the group once detected.

The main drawbacks of the protocols are their relatively large round complexity for group merge operations. This approach is that the resulting protocols are not optimal in their performance, i.e., the compiler adds a certain overhead of additional messages which do not seem necessary.

The service Rambo, which stands for "Reconfigurable Atomic Memory" for Basic Objects The rest of the paper, presents our algorithm and its analysis. The algorithm carries out three major activities, all concurrently: reading and writing objects, introducing new configurations, and removing ("garbage-collecting") obsolete configurations. The algorithm is composed of a main algorithm, which handles reading, writing, and garbage-collection, and a global reconfiguration service, Recon, which provides the main algorithm with a consistent sequence of configurations. Reconfiguration is loosely coupled to the main algorithm, in particular, several configurations may be known to the algorithm at one time and read and write operations can use them all.

III. Techniques

3.1 STORAGE

Byzantine Quorum Protocol, It's used to handle changes in replica set, and it provides strong semantics. That is, to enable them to be reconfigurable while continuing to provide atomic semantics across changes in the replica set. It includes protocols for read and writes operations and processing of messages during replica changes. Each object is stored at n=3f+1 nodes and quorums consist of any subset containing 2f+1 node.

A description of the client-side read and writes protocols for 2 functions are shown below,

Write (data)

Send messages to the replicas in the group that stores and wait for valid responses, all for server. Then send messages to all replicas and wait for valid response.

The write operation for a public-key object will normally has two phases. In the read phase, a quorum of 2f+1replicas iscontacted to obtain a set of version numbers for the object.

Read (data)

Send messages to the replicas in the group that stores and wait for valid responses, all for server, send messages to all replicas and wait for valid response. Then return data to the user.

To perform a read operation, the client requests the object from all replicas in the read phase. Normally, there will be 2f+1 valid reply that provide the same version number; in this case the result is the correct response and the operation completes. However, if the read occurs concurrently with a write, the version numbers may not agree.

As shown in fig 1.1 the user get back the data from server for that write-back operation is used In this case, there is a write-back phase in which the client picks the response with the highest version number, writes it to all replicas, and waits for a reply from a quorum.

3.2 PBFT

Practical Byzantine Fault Tolerance (PBFT), it describes a new replication algorithm that tolerates Byzantine faults and practical (asynchronous environment, better performance). The algorithm provides safety if all non-faulty replicas agree on the sequence numbers of requests that commit locally. It provides replicas must change view if they are unable to execute a request.

Replicas probe independently, and a replica proposes an eviction for a server node that has missed p_{ropose} probe responses.

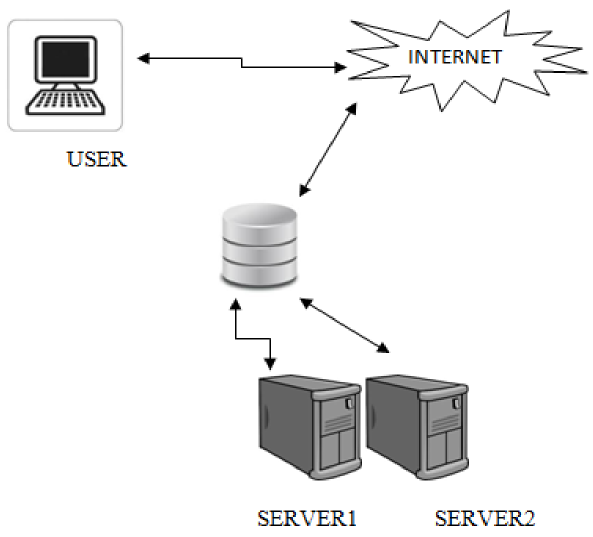

Fig 1.1 System Architecture

It does this by sending eviction messages toothier MS replicas, and then, waiting for signed statements from at least f_{MS}+1 MS replicas (including itself) that agree to evict that node. Other MS replicas accept the eviction (and sign a statement saying so).

If their last n_{evict}pings for that node have failed, where $n_{evict}<n_{propose}$.

Because the initiation of the eviction waited a bit longer than necessary, most eviction proposals will succeed if the node is really down.

IV.Experimental Result

We implemented the membership service and dBQS. Our experiments show that our approach is practical and could be used in a real deployment: the MS can manage a very large number of servers and reconfigurations have little impact on the performance of the replicated service.

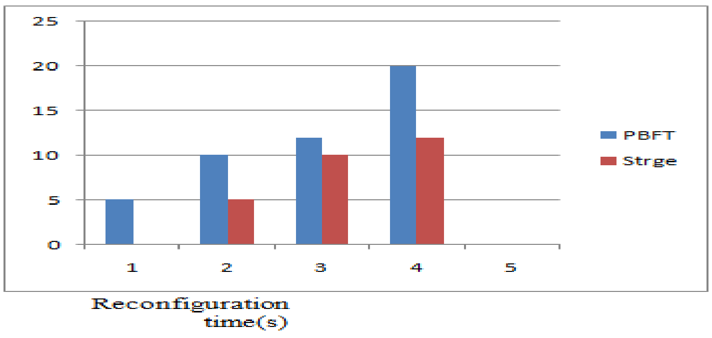

Fig 1.2 Performance Result

Throughout our experiments, we tried to determine how system components interfere with each other. For this experiment, the replica doing the pings was associated with an instance of dBQS (since we intend to run committees on system nodes). We repeated the experiment under three different degrees of activity of the dBQS server: when it is not serving any data (which will be the case when the MS does the pings and runs on special nodes that don't also handle the application), when it is handling 30 queries/second, and when clients saturate the server with constant requests, which leads to the maximal number of about 300 queries/second. Each query requested a download of a 512 byte block. little impact on the performance of the replicated service.

V. Conclusion And Future Work

A dynamically changing system membership in a large scale reliable storage system is maintained and carried by a special service. For that we provide a membership service that track the system periodically and notify the changes by using storage and PBFT algorithm. By which the system will reconfigured automatically. In future research, the more committees are needed for the data will be needed in the system size increases. The Membership service can accept the committee dynamically is based on system size and to add extension of our system. The design of a mechanism to determine which machines to place file replicas on the other file to use membership service.

References

[1] Birman. K and T. Joseph, "Exploiting Virtual Synchrony in Distributed Systems," Proc. 11th ACM Symp. Operating Systems Principles, pp. 123-138, Nov. 2007.

[2] Clement. A, M. Marchetti, E. Wong, L. Alvisi, and M. Dahlin,"Making Byzantine Fault Tolerant Systems Tolerate ByzantineFaults," Proc. Sixth USENIX Symp. Networked Systems Design and Implementation (NSDI '09), Apr. 2009.

[3] Cowling. J, D.R.K. Ports, B. Liskov, R.A. Popa, and A. Gaikwad, "Census: Location-Aware Membership Management for Large-Scale Distributed Systems," Proc. Ann. Technical Conf. (USENIX '09), June 2009

[4] Chen.K., "Authentication in a Reconfigurable Byzantine Fault Tolerant System," master's thesis, Massachusetts Inst. of Technology, July 2004.

[5] Castro.M and B. Liskov, "Practical Byzantine Fault Tolerance," Proc. Third Symp.Operating Systems Design and Implementation (OSDI '99), Feb. 1999.

[6] Dabek.F., M.F. Kaashoek, D. Karger, R. Morris, and I. Stoica,"Wide-Area Cooperative Storage with CFS," Proc. 18th ACM Symp. Operating Systems Principles (SOSP '01), Oct. 2001.

[7] Douceur.J., "The Sybil Attack," Proc. First Int'l Workshop Peer-to-Peer Systems (IPTPS '02), 2002.

[8] DeCandia.G, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's Highly Available Key-Value Store," Proc. 21st ACM Symp. Operating Systems Principles, pp. 205-220,2007.

[9] Lynch. N. and A.A. Shvartsman, "Rambo: A Reconfigurable Atomic Memory Service," Proc. 16th Int'l Symp. Distributed Computing (DISC '02), 2002.

[10] Rodrigo Rodrigues, Barbara Liskov, Member, IEEE, Kathryn Chen, Moses Liskov, and David Schultz "Automatic Reconfiguration for Large-Scale Reliable Storage Systems"-IEEE transaction on dependable and secure computing.

[11] Reiter.M., "A Secure Group Membership Protocol," IEEE Trans. Software Eng., vol. 22, no. 1, pp. 31-42, Jan. 1996.

AUTHORS PROFILE

Ms.K.vinothinigot B.Tech degree in information technology and now she is doing her M.E (computer science and engineering) in srinivasan engineering college,perambalur,TN,India

Ms.B.Amuthais working as Assistant Proffessor/CSE insrinivasan engineering college,perambalur,TN,India

Mr.B.Renganathangot B.Tech degree in information technology and now he is doing his M.E (computer science and engineering) in srinivasan engineering college,perambalur,TN,India