“AUTOMATIC RECONFIGURATION FOR LARGE SCALE RELIABLE STORAGE SYSTEM”

PROJECT REFERENCE NO. : 37S0696

COLLEGE : K.L.E COLLEGE OF ENGINEERING & TECHNOLOGY, CHIKODI
BRANCH : COMPUTER SCIENCE ENGINEERING.
GUIDE : BAHUBALI M AKIWATE
STUDENTS : ASHWINI KHANDEKAR
            SRILAXMI S DODAMANI
            NIMESH KUMAR
            PRATIMA BANSODE

Keywords: Byzantine Fault Tolerance, Membership Service (MS), Dynamic System Membership, Distributed Hash Tables.

Introduction:

Today we are more and more dependent on Internet services, which provide important functionality and store critical state. These services are often implemented on collections of machines residing at multiple geographic locations such as a set of corporate data centers. In Existing System, replication enhanced the reliability of internet services to store the data. The preserved data should be secured from software errors. But existing Byzantine-fault tolerant systems is a static set of replicas. It has no limitations. So, scalability is inconsistency. So, these data are not came for long-lived systems.

Our solution has two parts. The first is a MS that tracks and responds to membership changes. The MS works mostly automatically, and requires only minimal human intervention; this way we can reduce manual configuration errors, which are a major cause of disruption in computer systems. The second is storage system dBQS that provides Byzantine-fault-tolerant replicated storage with strong consistency.
Objectives:

- We introduced the MS that can reduce manual configuration errors and also a storage system, dBQS that provides Byzantine-fault-tolerant replicated storage with strong consistency.
- Online tracking using membership service which runs automatically to avoid the human errors which is called as Byzantine fault- tolerant and reconfigurable.
- Wireless Mesh Network is a promising technology and is expected to be widespread due to its low investment feature and the wireless broadband services it supports, attractive to both service providers and users.
- Anonymity & privacy issues focus on investigating anonymity in different context.

Methodology:

In this approach, we presented a complete solution for dynamically changing system membership in a large-scale Byzantine-fault-tolerant system. Byzantine-fault-tolerant replication enhances the availability and reliability of Internet services that store critical state and preserve it despite attacks or software errors. This can be problematic in long-lived, large-scale systems where system membership is likely to change during the system lifetime. We are developed our software using following requirements:

**Minimum Software required:**

- Operating System : Windows Family
- Language : Java
- Front End : JAVA, Swing (JFC), J2ME
- Tool : My Eclipse

**Minimum Hardware required:**

- Processor : Pentium 4 and above
- RAM : 512Mb and above
- Hard Disk : 40 GB
- Input device : Standard Keyboard and Mouse
- Output device : VGA and High Resolution Monitor
- Network device : Ethernet LAN adapter
System Architecture

Modules:

- Admin Module (Cloud Manager)
- User Module (Remote Client)
- Reliable Automatic Reconfiguration
- Tracking Membership Service
- Byzantine Fault Tolerance
- Dynamic Replication

Results and Conclusion:

Results: In this approach the membership service is able to manage a large system and the cost to change the system membership is low. Reconfiguration allows faulty servers to be removed from service and replaced with newly introduced correct servers. Reconfiguration is also desirable because the servers can become targets for malicious attacks, and moving the service thwarts such attacks.

Conclusion: From a proper analysis of positive points and constraints this provides a complete solution for building large scale, long-lived systems that must preserve critical state inspite of malicious attacks and Byzantine failures. We presented a storage service with these characteristics called dBQS, and a membership service that is part of the overall system.
design, but can be reused by any Byzantine-fault tolerant large-scale system. The membership service tracks the current system membership in a way that is mostly automatic, to avoid human configuration errors. It is resilient to arbitrary faults of the nodes that implement it, and is reconfigurable, allowing us to change the set of nodes that implement the MS when old nodes fail, or periodically to avoid a targeted attack. 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.

**Scope and Future Work:**

**Scope:** It is designed to work at large scale, e.g. tens or hundreds of thousands of servers. Support for large scale is essential since systems today are already large and we can expect them to scale further.

**Future Work:** “Tracing Fraud Users in Mobile Device” We make use of technique called cubix technique in which some notorious problem common in P2P communication systems where, some peers take advantage of the system by providing little or no service to other peers or by leaving the system immediately after the service needs are satisfied. Peer cooperation is thus the fundamental requirement for P2P systems to operate properly. Since peers are assumed to be selfish, incentive mechanisms become essential to promote peer cooperation in terms of both cooperativeness and availability. Typical incentive mechanisms for promoting cooperativeness include reputation and payment-based approaches. In the reputation-based systems, peers are punished or rewarded based on the observed behavior. However, low availability remains an unobservable behavior in such systems, which hinders the feasibility of the reputation-based mechanism in improving peer availability. By contrast, the payment-based approach provides sufficient incentives for enhancing both cooperativeness and availability, and thus, is ideal to be employed in multi-hop uplink communications among peer clients in our Wireless Mesh Network system.