

Continuous Availability in Replicated Program

Tarikesh Hatkar^{1*}, Bhushan Dhikale², Arshad Shaikh³ and Santosh Durugkar⁴

^{1*,2,3,4}Deptt of Computer, Late G.N.Sapkal College of Engineering, Nashik(India)

www.ijcseonline.org

Received: 09/03/2014

Revised: 22/03/2014

Accepted: 23/04/2014

Published: 30/04/2014

Abstract— One of the main reasons for income loss in enterprises that is usually addressed by providing redundant software and hardware is Service downtime. A way to provide software redundancy is to create a replica of the state of a program in execution to a set of replicas such as, when the primary fails. The place of the formerly running application occurrence is presumed by one of the replicas. A prototype of continuously available replica programs is provided, in case of failure of primary a replica takes over the functionality without any service downtime. Thus, the costly task of starting a new application occurrence on the other machine is avoided. When the mail servers get disconnected, design offline email gears service. Or email gears are being developed to encounter the network failure. All email operations can be performed even if the network fails.

Index Term— IP Switching, IP Pooling, Replication, POP3, SMTP, ICMP, ARP

I. INTRODUCTION

Email application brings your email data online and makes it available anywhere there's an Internet connection. Consider yourself being in a flight or areas where Wi-Fi is unavailable?



Fig 1. Stress during Service Downtime

The project called Continuous Availability in Email System that enables your client applications to be available without internet. Continuous Availability in Email System are an incremental improvement to the email service as it is today. It acts in such way so as to make the existing email application work without internet connection.

Continuous Availability in Email System today covers what we think is the minimal set of primitives required for offline apps. It is still a bit rough and in need of polish, but we are releasing it early because we think the best way to make Continuous Availability in Email System really useful is to evolve it into an open standard. Once you install the extension, the email client Gears-enabled application will ask your permission before storing data offline.



The way in which a function is implemented in an application the corresponding data is cached and stored thus making them accessible locally, stores data in a SQL database which contains powerful search functionalities and simultaneously syncs data in the background. All the information being available continuously in Email System will ensure that you will read the most recent messages from the email server, locally or to compose new emails, delete, view your inbox even if you are not connected to the internet.

II. LITERATURE SURVEY

One of the main reasons for income loss in enterprises that is usually addressed by providing redundant software and hardware is Service downtime. A way to provide software redundancy is to create a replica of the state of a program in execution to a set of replicas such as, when the primary fails. The place of the formerly running application occurrence is presumed by one of the replicas. A prototype of continuously available replica programs is provided, in case of failure of primary a replica takes over the functionality without any service downtime. A prototype of continuously available replica programs is provided, in case

of failure of primary a replica takes over the functionality without any service downtime.[1][3]

So to avoid the problem of service downtime we are going to develop system which allow you to perform offline operation while network failure.

1. PROGRAM REPLICATION

The program replication technique Continuous Available Replicated Program is based on Record and Replay (RR) principle. In principle, any program replication system that provides similar capabilities, such as RR designed to support the recording and successive replay of the execution of unmodified applications running on multiprocessor systems can be used to develop Continuous Available Replicated Program.

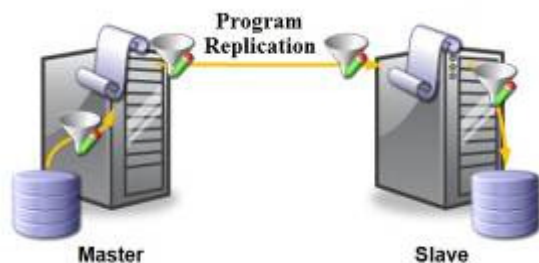


Fig 2. Program Replication

The storage related issues and support for recovery from silent data errors in a replicated program environment are considered by Continuous Available Replicated Program.

2. INTEGRATED REPLICATION

As stated above, in order to recover cleanly from a site failure combined storage and program replication mechanism is required. For example, if an application and its linked storage both are down at the primary site, a synchronized switch-over to a replica site must be ensured by the mechanism[3]. Else, if the program replication mechanism does a failover first, the services may be provided by replica site with varying storage data[6]. We studied three possible mechanisms that might provide continuous availability and protection to site failures.

1. The first mechanism is based on LVM mirroring, which keeps the contents of the logical volumes synchronized by replicating the data writes between the LVM layers in primary and replica.
2. The second mechanism makes use of general block level synchronous replication between replica storage systems and the primary to achieve the same goal.
3. In the third mechanism, storage replication is ensured by the program and its replicas. local storage is accessed independently by the program replicas.

Considered above three Mechanism, continuous application availability it supported only by the third mechanism upon site failure. With the first two mechanisms, since the replica volume is read only and being constantly updated to reflect

changes in the primary node, the application on the secondary node cannot mount a file system. Hence, the secondary program's read/write operations need to be monitored and its input data is provided from the primary node using the program replica of program. If primary node fails, the only recovery option available is to mount the secondary's file system on the replica volume. But state of the replica volume needs to be changed from write to read-write state[4][5]. This problem can be overcome by restarting the application, but that would interrupt the continuous availability constraint. The third mechanism works with continuous availability constraint since it is ensured that all replica file systems are having the similar data, there is no loss of data at replica sites due to uncommitted writes at the primary site, allowing recovery of the secondary application without need of a restart.

III. PROPOSED SYSTEM

1. Replication

In computing, ensure consistency between redundant resources required sharing of the information such as software or hardware components, Replication is to improve reliability, accessibility, fault-tolerance and helps in concurrency controls[2].

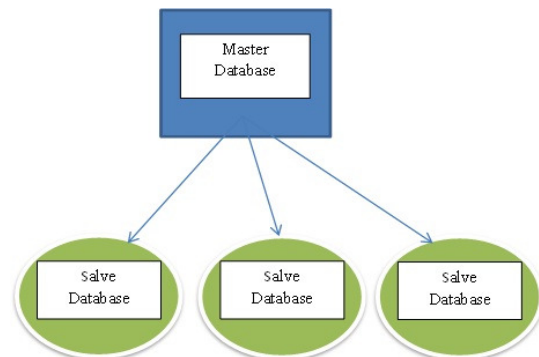


Fig 3.1. Master Slave Architecture

1. Data replication means multiple storage devices can stored same data.
2. Computation replication means Multiple times executed same computing task.

The access to a replicated entity is typically same like access to a single, non-replicated entity. The replication of program itself should be clear to an outside user.

2. NEED BEHIND REPLICATION

In a distributed system replication of data deals with following potential benefits:

1. Increases availability
2. Increased reliability
3. Improved response time
4. Reduced network traffic

5. Improved system throughput
6. Better scalability

The main goal of this paper is to provide a accurate and complete list of the necessities of *Continuous Availability in Replicated Program*. Gears are an incremental enhancement to the email service as it is. It improves just sufficient to make current email client applications work without internet (mail server downtime).

This Email Gear program allows storing, downloading entire Web sites at Record server which locally situated. Client can storing and viewing them in future, at user's suitability. Email gear affords right to use to the downloaded information at any time, no problem where its users are. It really goes away from traditional web browsing by participating powerful and original features to analyse, find, track and manage information on the Internet.

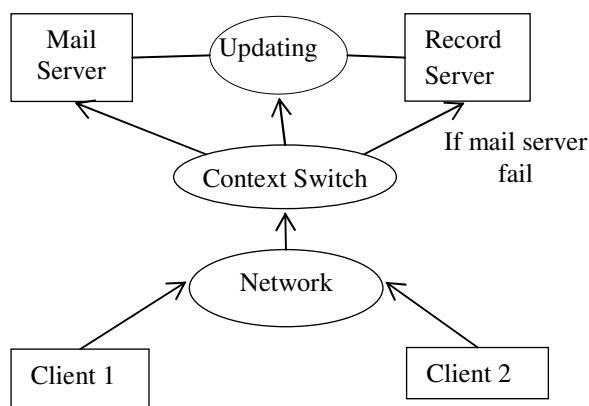


Fig 3.2. Proposed System

Email Gears will allow you to read the most recent messages from the email server while internet is not available or to compose your emails, delete, view your inbox. Proposed system which consist of Program and storage replication system may provide high data replica consistency, availability and implements a mechanism for program to detect and recover from various data errors.

So we are going to develop system to avoid the problem of service interruption which allow you to perform operation without internet while network failure.

3. EMAIL GEARS SERVICES

Gears are an incremental improvement to the email service as it is today. It improves just sufficient to make current email client applications work without internet. Gears today covers what we think is the minimal set of primitives required for offline apps. Email Gears will enable you to read the most recent messages from the email server while offline or to compose your emails, delete, view your inbox in even without a network connection. Email gear monitor and record the activity of the mail server.

Modules:-

1. Monitoring
2. Record events
3. IP pooling.
4. IP releasing
5. Rollback records of database.

1. MONITORING

Gear servers monitor all the activities of mail server and update database by synchronizing mail. It maintain all information of mail server what operation it perform & how it perform, So while no connection the information of operation performing given by this module.

2. RECORD EVENTS

Gear server record all the events of email and update in the database. It keep record of Email operation all type of events so when network problem is solved then according to event updating to mail server.

3. IP POOLING

During the connection failure of mail server, gear server enable IP POOLING and assign IP address of mail server to the gear server so, that gear server act as a mail server and continue with the services to be fulfilled for client. IP POOLING is process in which IP address of mail server so while connection failure IP address from IP pool is assign to passive server.

4. IP RELEASING

When mail server connection comes into his original state then IP RELEASING activity take place and IP address is again assign to the mail server and all process are continue normally. While connection failure IP address of mail server is assign to passive server which is gear server. So when connection to mail server is establish then by releasing IP from passive server and assign it to the mail server.

5. ROLLBACK RECORD

Gear server rollback the records every update done during the offline connection to the mail server. When network fail then all operation are perform on passive server but when connection to mail server comes into its original position then make all changes to mail server. In rollback process record are updated to mail server.

IV.IMPLEMENTATION DETAILS

1. Load balancing

The development of distributing systems with the opportunity of sharing the existing resources has managed to the capability of executing some jobs arrived at a definite node remotely. Therefore, load balancing is pointing at decreasing the total response time of jobs execution in any distributed system. This increases the possibility of maximizing the overall utilization of any given distributed

system by sending some of the tasks at highly loaded nodes to be executed remotely.

Almost all scheduling strategies in distributed systems depend on load status and system nodes locations. The used strategies are normally categorised under centralized, distributed and internally source or server initiative and other. Such classification entails unavoidable overheads which stem from the fact that heavy computations are necessary to balance the load. To take a reasonable decision of sending a job to be executed remotely or not depends, to some degree, on the amount of information available at the load balancer node. At such node a formal assessment is needed to be done to take a right decision of where to perform the newly arriving job. The information needed to take a decision by a load balancer is composed of the status of each other node in the system, the location of the nodes that are willing to accept the transferred jobs, and some other information depending on the load balancing strategy applied, such as the criteria of transferring a job.

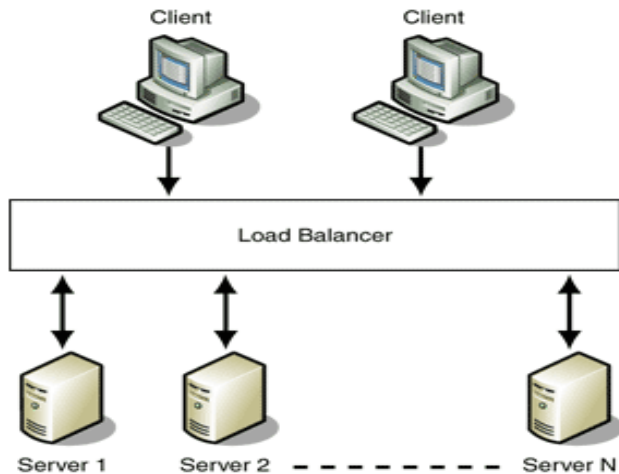


Fig 4.1. Load balancing

Email Gears will enable you to read the most recent messages from the email server while offline or to compose your emails, delete, view your inbox in even without a internet connection.

A simple overview of the Continuous Available Replicated Program architecture is shown in although the principles underlying Continuous Available Replicated Program can be extended to multiple replicas, for simplicity and to reflect our prototype implementation, we consider the case of a single secondary replica supporting a primary application. In the description of the design, we assume without the loss of generality that program A and its clone B run on different hosts, each in an RR Container. Each program is backed by a file system and back-end storage environment, and the two environments are equivalent in that the file system name-space relevant to programs A and B are identical.

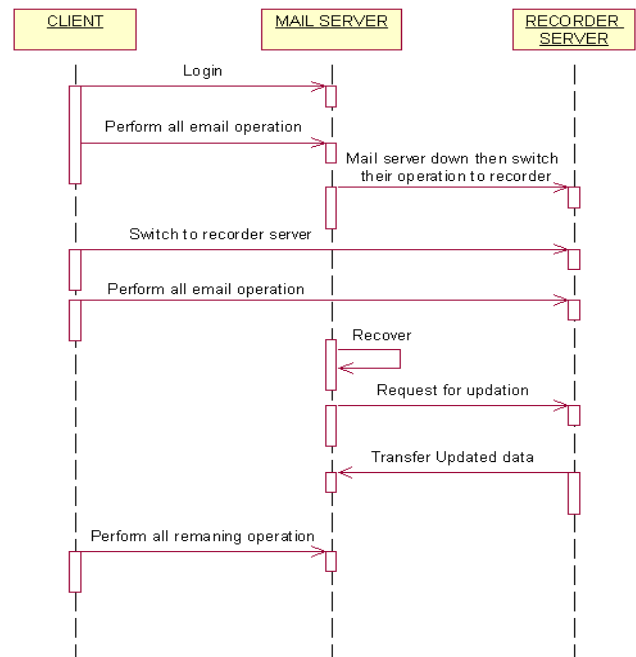


Fig 4.2. Sequence Diagram for System

A and B each issue system calls (e.g., open, close, read, write) on separate files of the same name on each host, and each host's file system stores the underlying data on a separate storage system. A replication link between the two containers is used to exchange program state information (needed for program replication) as well as I/O-related information (needed to handle silent data errors) to ensure the deterministic execution of program B with respect to A. Continuous Available Replicated Program is transparent by design and works with unmodified applications since Continuous Available Replicated Program operates only by intercepting file system calls made by these applications.

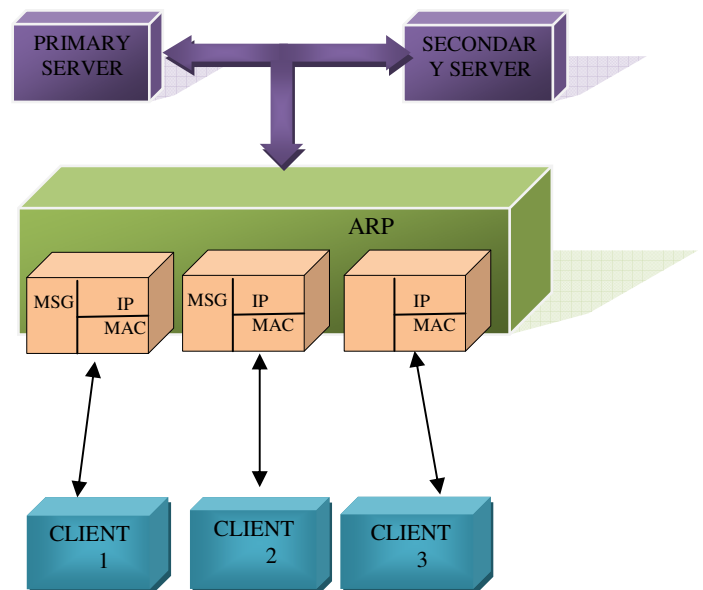


Fig 4.3. System Architecture

2. SMTP

SMTP Means Simple Mail Transfer Protocol and is how your email service delivers your message to the recipient. If you are on a browser email service then you will not have to worry about setting anything up but if you are using an email client, you may have to tell your program what server your email service is located on. For example, Google's SMTP is located at mail.google.com. If the recipient is using the same server as you then the SMTP simply hands your message over to the recipient's account. If the recipient is using a different email service, however, then the SMTP must first request the location of the recipient account from the domain that it is using. Once the SMTP locates the recipient's account, it passes your message over to the other domain which then hands it over to the recipient's account.

3. POP3

Full form of POP3 is Post Office Protocol. Post Office Protocol is what allows your email client to receive messages from your browser email service. POP3 requires that you setup a username and password and generally must be setup from both your browser email service and your email client. When the POP3 server contacts your account and you sign in, it copies the emails from your online text file and downloads them to your computer. Usually POP3 deletes your messages from the server once it downloads them to your computer but some email services, such as Gmail, allow you to keep the messages on the server so that you can still view them online at any time.

4. IMAP

Full form of IMAP is Internet Mail Access Protocol. Internet Mail Access Protocol is one other method of receiving messages from browser email service. Internet Mail Access Protocol permits to create online folders in which you place your emails and your online email account simply connect from your computer rather than truly downloading the messages. Internet Mail Access Protocol allows you to view messages on multiple computers rather than having your messages deleted from the server once they have been downloaded. Upgrades in POP3, however, make Internet Mail Access Protocol obsolete as you can keep your emails on the email server anyway and have them forwarded to email clients on multiple computers.

PASSIVE SERVER CHARACTERSTICS

1. Monitoring
2. Record events
3. Ip pooling.
4. Ip releasing
5. Rollback records of database.

General Overview of System

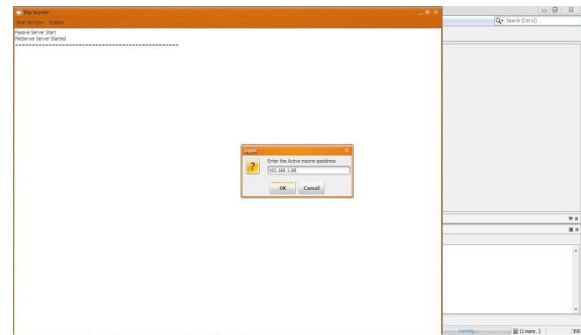


Fig. 4.4 Activating the "active" Server

Advantages of System:

1. Increment Server Efficiencies:

The main benefit of the offline web server is to increment the server efficiency .

2. Consistent accommodations during offline:

The accommodation will be remaining consistent during offline, when the server connection is down or if there is slow internet connection.

3. Offline working:

This is the most paramount feature of the system. Client can send or receive all his mail and can perform operations on them when he is not online.

Our consequentiality is on developing a system which works on following things:

- Provide a system this will give access to client's Inbox which is a genuine task intended.
- Provide a system which will keep track of multiple users for multi utilizer task.
- Provide a system which will sanction to client send, compose, delete, forward mails.
- Provide a system which will interact with mail server, It gives utilizer cordial approach.

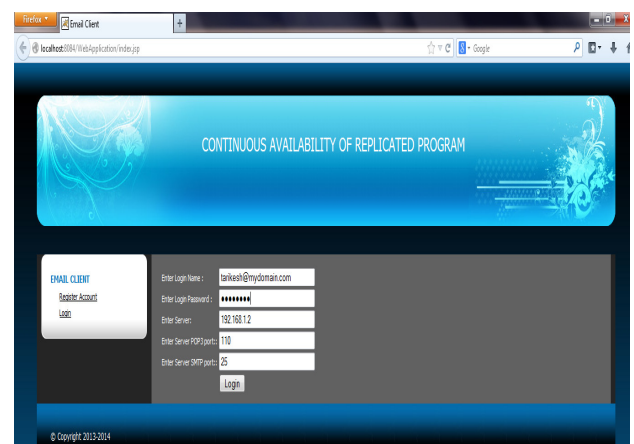


Fig 4.5. Authentication Process of Mail Application

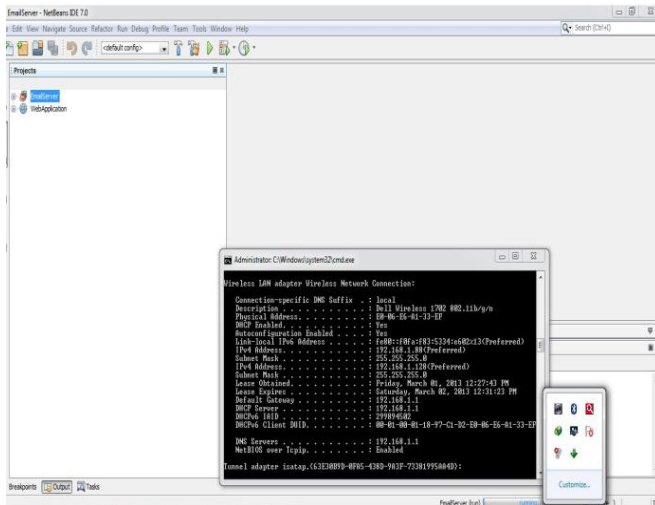


Fig 4.6. IP switching

V. CONCLUSION

Recent storage-recovery requirements present a problem for predictable business-continuity arrangement. With the appropriate logistics, requirements can be made to substitute system platforms, networks and also user work areas quickly. Time to data is real key to recovery. How rapidly data can be restored for use by business application and customers are basis of successful recovery.

The Continuous Availability in Replicated Program allows your all client applications to be accessible without internet. Client can do all action like compose, send, delete forward, reply to the mails without internet which accessible from local or record server. when primary mail server start working the data will get automatically updated. Email Gears is an incremental growth to the email service as it is . It adds just enough to make current email client applications work offline.

Given this fact, the proliferation both in volume of data and in the type and topology of storage platforms within a single company can create requirements that will make or break the efficacy of all other recovery plans.

VI. SCOPE FOR FUTURE RESEARCH

This system further extended to be used for individual Computer without using passive server machine at different location.

REFERENCES

- [1] Lanyue Luy, Prasenjit Sarkar, Dinesh Subhraveti and Ahmed Bashir Rice "CARP: Handling silent data errors and site failures in an integrated program and storage replication mechanism", IEEE International Conference on Distributed Computing Systems, 2011

- [2] Arvind Kumar, Rama Shankar Yadav, Ranvijay, Anjali Jain, "Fault Tolerance in Real Time Distributed System", International Journal on Computer Science and Engineering (IJCSSE), ISSN : 0975-3397 Volume- 03, Issue -02, Page No- (933-939), Feb 2011.
- [3] J. L. Hafner, V. Deenadhayalan, W. Belluomini, and K. Rao, "Undetected Disk Errors in RAID Arrays," in IBM Journal of Research and Development. VOL.52, 2010.
- [4] L. N. Bairavasundaram, G. R. Goodson, B. Schroeder, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "An Analysis of Data Corruption in the Storage Stack," in FAST, 2009.
- [5] B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, and A. Warfield, "Remus: High availability via Asynchronous Virtual Machine Replication," in NSDI, 2008.
- [6] Andrew S. Tanenbaum "Computers Networks" (4th edition), Prentice Hall PTR 2003.
- [7] Roger S "Software Engineering- A Practitioner's Approach" 5th Edition McGraw-Hill, 2005.

Authors Profile



Mr. Tarikesh Hatkar
BE (Computer Engineering)
Late G N Sapkal COE
Nashik



Prof. Santosh Durugkar
M.E.I.T., PGDFM LMISTE,
Late G N Sapkal COE
Nashik



Mr. Bhusan Dhikale
BE (Computer Engineering)
Late G N Sapkal COE
Nashik



Mr. Arshad Shaikh
BE (Computer Engineering)
Late G N Sapkal COE
Nashik