www.ijsrnsc.org

Available online at www.ijsrnsc.org

IJSRNSC

Volume-1, Issue-4, Oct- 2013
Research Paper

Int. J. Sci. Res. in Network Security and Communication

ISSN: 2321-3256

A Shared Memory Technique for Windows Environment through Virtualization

Ganesan.T[1*], Tamizharasan.P[2], Sabari Giri Murugan.S[3]

[1*,2,3] *Department of Information Technology, V.S.B.Engineering College, Karur, India.*

Received: 11 September 2013 Revised: 15 September 2013 Accepted: 12 October 2013 Published: 30 October 2013

Abstract— In a network file sharing system, many practical algorithms to support Byzantine Fault-Tolerant distributed applications has been made in topical years. These solutions are designed to make the applications resistant to successful attacks alongside the system, thereby making services tolerant to intrusions. In recent times, some of these studies have considered the use of virtual machines for building a trusted computing environment. This project presents SMIT (Shared Memory based Intrusion Tolerance), an design for Intrusion Tolerance using virtual machines that benefits from a shared memory to simplify the consensus protocol. P2P overlay topologies and their dynamics, focusing on the modern network. We present Cruiser, a fast and exact P2P crawler, which can capture a complete snapshot of the network of more than one million peers in just a few minutes. Leveraging recent overlay snapshots captured with Cruiser, we distinguish the graph-related properties of individual overlay snapshots and overlay dynamics across slices of back-to-back snapshots. Our results reveal that while the network has dramatically grown and changed in many ways, it still exhibits the cluster and short path lengths of a small world network.

Keywords- Virtual Machines, Memory, Intrusion, Network

I. INTRODUCTION

The role of the computing systems play in our society is growing in importance. The trust and dependency over such systems have increased considerably, day after day. It is necessary that they behave properly even under the presence of faults, which can lead to large losses, from financial to human. Recently, operating systems faults have been frequently appeared as intrusions (e.g. viruses, Trojans, etc), which are the result of a malicious attack that achieves success by exploiting one or more vulnerabilities of the system (e.g. bugs) [1].

To ensure that these systems remain available, correct and safe even in the presence of faults and vulnerabilities, is necessary the development of mechanisms to provide intrusion tolerance in these environments [3]. To ensure security to these systems, solutions for the development of fault-tolerant applications have been researching so for and in the last decade several studies with solutions for Byzantine Fault-Tolerant (BFT) State Machine replication (active replication) with practical viability were proposed [1]-[4].

The BFT protocols are, in general, replicated services in a set of machines that communicate between themselves to provide a safe and reliable service, even under the presence of a limited number of malicious member's intruders that behave out of the protocol specification [7]. Such protocols are designed to allow the implementation of replicated services that are able to meet the requirements for reliability, integrity and availability, which are fundamental to achieving dependability [5].

In our project, each replica is executed in a virtual environment, with the whole set of replicas running in the same computer. These proposals make feasible to implement the concept of services and operating systems diversity at virtual machines level. An obvious advantage of such approach is the reduction of the replication cost to implement an intrusion tolerant service by the use of a single physical machine.

These proposals are based on the observation that Byzantine failures with malicious intent (intrusion) occur by the attempt to use the vulnerabilities of the part of computer system implemented in software (i.e operating system, device drivers, services, applications, etc.).

In our project, we proposes SMIT (Shared Memory based Intrusion Tolerance), a new architecture for development of intrusion-tolerant services using virtualization technology, and an algorithm to perform services over that replicated architecture. The most important point about our project is the application of a shared memory abstraction between the virtual machines to simplify the consensus protocol.

- Thus, we are able to avoid point to point message passing communication between the replicated services, what could increase the algorithm complexity to reach consensus. Our project demonstrates that this safe

Corresponding Author: *Ganesan.T*

component only needs to provide a simple shared memory abstraction to reach this reduction. Furthermore, the host machine does not play an active role in our proposal, because it only needs to provide a communication abstraction.

II. LITERATURE SURVEY

P2P system can cause troubles like excessive network traffic and/or overloads on the server. In addition to these problems, due to the irregular response for searching data, the result is not always guaranteed and it is often inefficient. Thus, the solution is the Structured Network P2P system with DHTs studied very actively [2]-[3]. It is the table search algorithm which enhances search speed by using array of combinational keys made by hashing. The algorithm that searches and recognizes the nodes in the network and enables to share the actual files by using DHTs method in P2P system is implemented to enable rapid and systematic searching and routing. The file sharing P2P system with DHTs method, for example, users in shared network are provided with unique keys which help to search files which are registered in DHTs [6]. Each node and connected values are associated with hashed result. These work lead to find nodes wanted other users by hashing the file name [10].

System Study-Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimate. During system analysis the feasibility study of the proposed system is to be carried out. This is to make sure that the proposed system is not a burden to the company. For feasibility analysis, some kind of the major requirements for the system is essential.

Three key considerations concerned in the feasibility analysis are

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

Economical Feasibility-This study is carried out to check the economic impact that the system will have on the organization. The entire of fund that the company can pour into the research and development of the system is limited. The expenditures must be acceptable. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchase.

Technical Feasibility-This study is carried out to check the technical feasibility, that is, the technical necessities of the system. Any system industrial must not have a high require on the available technical property. This will lead to high load on the available technical property. This will lead to high difficulty being placed on the client. The developed system must have a diffident requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility-The aspect of study is to check the level of acceptance of the system by the user. These include the process of instruction the user to use the system resourcefully. The user must not feel threatened by the system, instead must accept it as a requirement.

The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him common with it. His level of confidence must be raise so that he is also able to make some constructive criticism, which is welcome, as he is the final user of the system.

Existing System

Previous studies which mainly based on the Distributed Hash Tables (DHTs).In which location of the data in the system connected to the node can be stored and it may provide fast access to sharing of data [8]-[9].

But it may has the following drawbacks such as,

- The system connected to the node only can share the data.
- Hackers know the node address can easily change the data.
- Malicious programs such as worms and threads can easily corrupt the data.

III. PROPOSED SYSTEM

In our proposed system, we maintain a virtual tools to maintain guest operating system through VMware, VirtualPC and Virtual Box etc.,.The guest operating system which holds the network IP address of the host system and data can be shared to the other system through Guest Operating system. We maintain a shared folder between host and guest operating system and we have taken a primary backup in the host system. The data shared to the other system through the network.

The main advantages of using this come within reach of are,

- The same data can be used even though the system is affected by malicious program.
- The unauthenticated person not able to hack the data in the system.

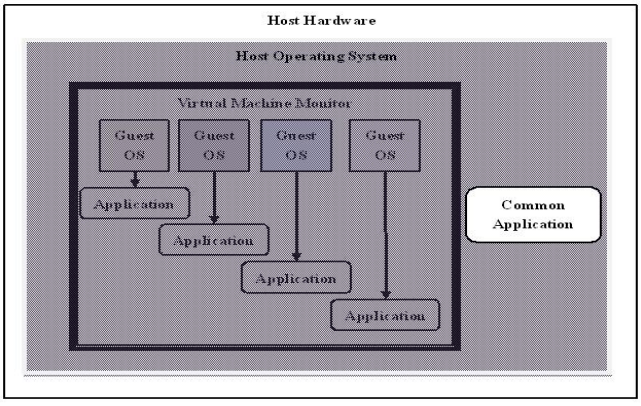

Figure 1: Virtualization Architecture

Module Description

- Information sharing
- Authentication Module
- Peer List
- File Searching
- File Uploading
- File Downloading

Information Sharing- Information sharing module mainly focuses on keeping the primary backup of the data .In this module data can be shared between both guest and host operating system.

Authentication Module- The Authentication Module is a main purpose of providing secure entry to project Phase. After the registration processes, once registered by the User, the Authenticated user can enter into project.

=*Peer List*- The Main Purpose of searching a peer is to improve the communication between the two peers. In this development, we are searching a peer list. According to our project we get more number of peers to improve file sharing process. During this method we are getting peer list according to their company workgroup name.

File Searching- In this module file Searching Process has been carried out by communicating with selected peers. It is possible to search the files with active peers.

File Uploading- In this file uploading Process, active peers have been communicated with selected peers; can upload file to other peers. Through this method, it is achievable to upload all kind of files such as audio files, video, images, documents, etc.

File Downloading- In this downloading Process, active peers have been communicated with selected peers; can download files from other peers. Through this method, it is possible to download and view all kind of files such as audio files, video files, images files, documents, etc.

Data Flow Diagrams

Level 0

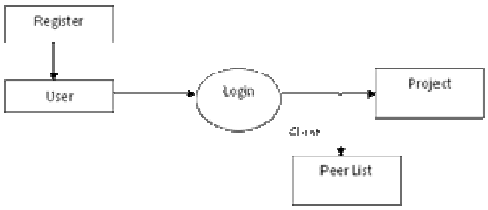

Figure 2: Data Flow Diagram Level 0

As shown in the figure the registration has been carried out. Only the registered user can login to the project phase and the peer list will be displayed.

Level 1

Figure 3: Data Flow Diagram Level 1

After the successful login into the project phase it shows the list of peer lists in the form of active and inactive, it is possible by the network application.

Level2

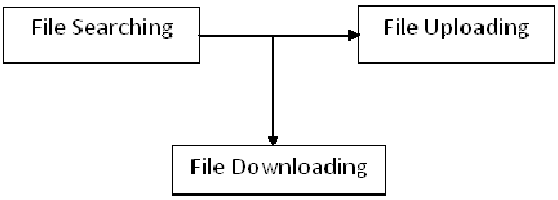

Figure 4: Data Flow Diagram Level 2

Alone the peer lists are made into topology, which means it breaks from the current structure and randomize through the network application. So it makes the user easier to share the files.

Level 3

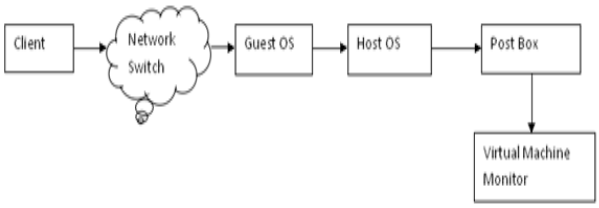

Figure 5: File Sharing Between Host And Guest

The client in the network connection can share the data to server through the Guest

System Development

System development is the important phase in which each and every module of the project has been developed. Over all in this paper there are seven module, so here some of the modules have been developed as follows

Development of Modules

- Authentication
- Peer list
- File Sharing

Authentication

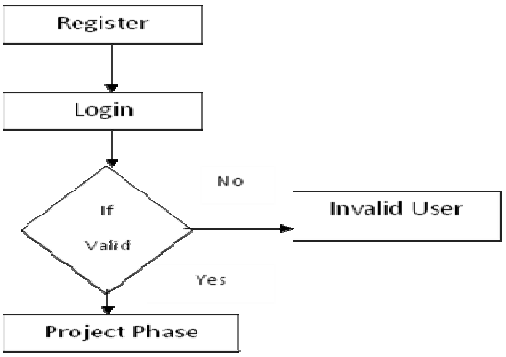

Figure 6: Authentication Phase

In this module there are two tabs one for registration and another one for login. Only the Authenticated user can enter into the project phase

Peer List

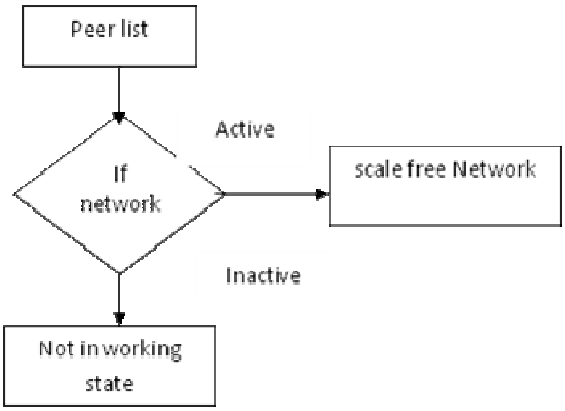

Figure 7: Peer List Phase

In the next module the peer list are get by the user. Active and Inactive peers are generated through the Gnutella application where it performs the scale free network in which the node breaks from its original network scale, and forms scale free network as shown below.

Most modern file-sharing networks use a two-tier topology where a subset of peers Called ultra peers, form an sparse graph while Other participating peers, called leaf peers, are connected to The top-level overlay through one or multiple ultra peers.

File Sharing- In this phase there are two tabs namely as File uploading and File downloading, with the help of the random scale network the sharing will be faster. In File downloading tab the user can able to get the files from the active peers such as text, images and documents and save it in their system memory, same as in the File uploading tab user can able to upload files from the parent node to the active peer. Thus the way the file sharing phase has been designed and developed.

IV. SAMPLE INPUT AND OUTPUT

Guest And Host Os

Figure 8: Guest And Host Os

Description

- In this data can be kept as backup in host os.
- Sharing of data in a network can be taken been place using guest os.

Main Form

Figure 9: Main Form

Description

- It is the initial operation of our project.
- After this only we can go for login and registration operation.

Login Form

Figure 10: Login

Description

- Only authenticated user can able to perform its sequence of operation.

- Once user enters its value, it can be checked with data base for verification.

Registration FORM

Figure 11: Registration

Description

- User has to register their username and password for further process.
- After registration only user able to perform login operation.

Peer List Form

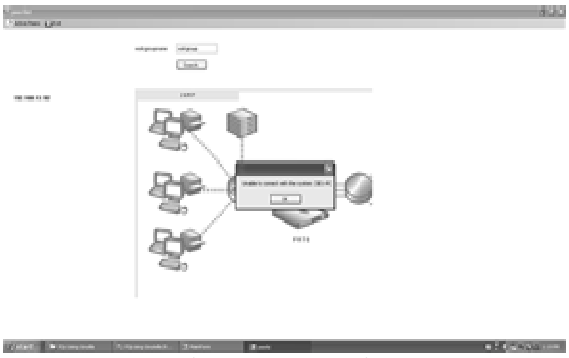

Figure 12: Peer List

Description

- In this form user have to mention its workgroup for sharing of data.
- It will display the IP address of the system belonging to the workgroup.

Work Group Form

Figure13: Work Group

Description

- In this form, once we entered workgroup name it will display the active and inactive system in a group.

Crawler

Figure 14: Crawler

Description

- It is mainly used for searching for the data to be shared.
- Once we select the data it can perform sharing operation.

File Sharing

Figure 15: File Sharing

Description

- It is used for select the files to perform sharing operation.
- We can perform sharing of files only between active peer systems.

File Uploading

Figure 16: File Uploading

Description

- In this file uploading process, active peers have been communicated with selected peers.
- It is possible to upload all kind of files such as audio, video, images, documents.

File Downloading

Figure 17: File Downloading

Description

- In this file downloading process, active peers have been communicated with selected peers.
- It is possible to download all kind of files such as audio, video, images, documents.

V. CONCLUSION

To conclude , an algorithm for BFT replication using virtualization to deploy replication and service diversity to provide as a result an Intrusion Tolerance System. To evaluate the practical viability of the model, we have developed a prototype, running micro-benchmarks over it to measure the response time and the throughput obtained. The future works are focused on architectural and algorithmic improvements to support malicious clients, on the architecture implementation using another VMM on the possibility of create a postbox in main memory and on a distributed version, in order to tolerate crash faults in the host system. The captured topology is a snapshot of the system as a graph, through the peers represented as vertices and the connections as edges. Though, capturing accurate snapshots is inherently difficult for two reasons:

- Overlay topologies modify as the crawler operates and
- Non-negligible fraction of peers in each snapshot is not directly reachable by the crawler. When a crawler is slow virtual to the rate of overlay change, the resulting snapshot will be considerably distorted.

VI. FUTURE ENHANCEMENT

Every application has its have merits and demerits. The project has enclosed almost all the requirements. Additional requirements and improvements can easily be done since the coding is mainly structured or modular in nature. By Changing the existing modules or adding new modules can append improvements. More enhancements can be made to the application, so that the web site functions very attractively and in a useful manner than the present one. As far now we are doing this p2p file sharing in the LAN (Local Area Network) network only (i.e.), inside a room or a building, but in future it can be extended to share the files from one peer to another and keep the data secure from intruders and malicious programs which is located in far away distance using WAN (Wide Area Network).

VII. REFERENCES

[1.] B. G. Chun, P. Maniatis, and S. Shenker, "Diverse replication for single-machine byzantine-fault tolerance," in Proceedings of the 2010 USENIX Annual Technical Conference, 2010.

[2.] H. P. Reiser and R. Kapitza, "VM-FIT: Supporting intrusion tolerance with virtualization technology," in Proceedings of the 1st Workshop on Recent Advances on Intrusion-Tolerant Systems, 2009, pp. 18–22.

[3.] M. Correia, N. Neves, L. Lung, and P. Verissimo,"Worm-IT– a wormhole-based intrusion-tolerant group communication system," The Journal of Systems & Software, vol. 80, no. 2, pp. 178–197, 2009.

[4.] B. G. Chun, P. Maniatis, and S. Shenker, "Diverse replication for single-machine byzantine-fault tolerance," in Proceedings of the 2010 USENIX Annual Technical Conference, 2010.

[5.] H. P. Reiser and R. Kapitza, "VM-FIT: Supporting intrusion tolerance with virtualization technology," in Proceedings of the 1st Workshop on Recent Advances on Intrusion-Tolerant Systems, 2009, pp. 18–22.

[6.] M. Correia, N. Neves, L. Lung, and P. Verissimo,"Worm-IT– a wormhole-based intrusion-tolerant group communication system," The Journal of Systems & Software, vol. 80, no. 2, pp. 178–197, 2009.

[7.] Wine Project, "Wine user guide," http://www.winehq.com/site/docs/wine-user/index.

[8.] Kevin Lawton, Bryce Denney, N. David Guarneri, Volker Ruppert, Christophe Bothamy, and Michael Calabrese, "Bochs x86 pc emulator users manual," http://bochs.sourceforge.net/, 2003.

[9.] "Transmeta corp crusoe processor," http://www.erc.msstate.edu/ reese/EE8063/html/transmeta/transmeta.pdf.

[10.] "Qemu cpu emulator," http://fabrice.bellard.free.fr/qemu/qemu-tech.html, 2004.

