ANALYSIS OF SOFTWARE REJUVENATION ON ADHOC CONFERENCING AND MEDIA DISTRIBUTION SYSTEM

Karthika Murthy
M.Tech [ISE IV\textsuperscript{rd} Sem]
karthikamurthy@gmail.com

Rashmi R
Assistant Professor
rashmi@gmail.com

Information Science Department
R.V. College of Engineering
Bangalore, India- 560059

Abstract: The project facilitates the concept of Software Rejuvenation and its importance in day to day software applications. Software Rejuvenation prevents ageing in software applications and helps in achieving a non-compromising performance. “Adhoc conferencing system” is chosen as a candidate as this has a variety of functionalities that involve conferencing, screen sharing and file sharing with more possibilities of anomalies like memory leaks, data corruption, etc., over a period of time.

Keywords: Memory Leaks, Software Rejuvenation, Video Conferencing, Media Distributed System,

1. Introduction

Video Conferencing involves coordinating a small or large number of people together in the same place at the same time in order to discuss real time issues such as introducing new products, staff members, ideas, or for working through issues, problems, customer requirements or other mission critical issues. In our modern world, face-to-face conferencing is costly and difficult to achieve. As most of the companies are multinational, it becomes a herculean task when one has to travel from one part of the world to another for a meeting [1]. Hence video conferencing plays an important role here. This has become an extremely prominent business approach to video conferencing as it reduces the stress and strain of journey as well as cost involved with it. Based on this approach, lot of new products are being designed which gains the market with respect to the available feature which are different from others.

Linux is widely used as enterprise operating system. But there is a lack of solutions available for collaboration and conferencing using Linux. Most of the solutions currently available either is proprietary licensed or support very limited features. Linux has good proficiencies available for media processing and networking which can be utilized to support conferencing. In this project, open source libraries are used to enable audio and video conferencing in Linux. It will include the following features like Audio conferencing, Video conferencing; Screen sharing, File sharing. There are already many existing solutions available in Linux platform. But these solutions are proprietary licensed and have many limitations like the number of persons joining the conference, platform independency, audio support, video support, multiple OS support. Apart from these issues any available software nowadays due to the prominent usage of the tools suffers from software aging. Hence the main challenge is to develop a software rejuvenation methodology for multimedia application and also to develop an open source application that integrates an audio, video conferencing, screen sharing and file sharing.
2. Related Work

During 1997, Heather Molyneaux, Susan O’Donnell, Helen Fournier, and Kerri Gibson discussed about the current state-of-the-art of immersive communication that provides a vision for the future and the associated benefits, and considers the technical challenges in achieving the communication [2]. The idea is to make people actively participate in conferencing, especially rural places.

Rejuvenation on other hand is also being tried to achieve on TV presentations. Marcio Ferreira Moreno, Luiz Fernando Gomes Soares aims at providing resilience to digital TV presentations. As proof of concept, the recovery plan has been incorporated to the Ginga-NCL reference implementation [4].

Kishor S. Trivedi, Kalyanaraman Vaidyanathan and Katerina Goˇseva-Popstojanova presented measurement based approach which turned out to be an important step towards predicting software aging related failures based on actual measurements. Intend of the presentation is to help development of policies that automate the proactive handling of software aging related problems [5].

Being a media distribution system with video conferencing, Screen sharing, and file sharing functionalities, it leads to memory leaks and memory corruption while running for a long time. Dynamic monitoring tools, such as the state-of-the-art are commonly used to detect memory leaks and memory corruption [7]. Even some of the techniques like multivariate state estimation technique (MSET) or empirical evaluation are also used to perform proactive detections [8 ][9].

3. Proposed Work

The software is being developed to encourage participation among a group of people at multiple locations using Adhoc conferencing. The focused areas are 1. Understating the interactions between users. 2. Technology 3. Group Dynamics.

Initially the clients creates a new Universal Plug and Play (here onwards referred as UPnP) context and then create UPnP control point (which would use UPnP protocol to control UPnP devices) targeting the available servers. Following, the client will try to connect to the server, only if any of the servers matches to the service requirement. Identifying the service given by the server and the action provide by each service, the client invokes the required action, ensuing successful connection between the client and server. Subsequently the client sends and receives the data stream form the server, decodes the received data stream and plays the specific multimedia elements. Finally the client closes connection with the server. In order to set up a server, a device description file and a service description file (defining the service specifications given by the server) are created. A server creates an UPnP context with a host IP address, then creates a root device and activates itself in order to announce it. Consequently it would wait for the client request and connect to the client when any request is confronted. Last the server sends and receives data with the client, add or remove them dynamically. Finally the server stops the streaming data to client.
4. Software Rejuvenation

4.1 Software Rejuvenation to avoid Memory Leaks

This paper discuss about the rejuvenating the multimedia system. When a multimedia application is being played on the desktop or laptop the resource (memory) allocation is limited hence effective usage of the available memory plays an important role. Multimedia is involved with lot of data transfer with respect to bit stream. Each bit stream is stored in queue element and then set to play status. Though the queue memory is cleared ideally but practically the memory contains some residue which has to be cleared to provide a delay less video display. The problems that can lead to software aging over time resulting in software failure and server downtime include Memory leaks—memory leaks from software can cause virtual memory, non-paged pool and paging space to increase, and over time this degrades performance to the point at which the system is unresponsive. As a result of memory leak, Data corruption and Non terminated threads creates performance problems [6].

4.2 Scope of Innovation in Software Rejuvenation

Software Rejuvenation can be implemented at three different layers. They are OS layer; Kernel layer; and Application layer.

Based on the industrial experiences the figure below is designed. The below diagram depicts the probability of innovating new techniques to implement software rejuvenation. In the diagram the area is directly proportional to the scope of innovation in the software rejuvenation.

Fig 4.1 Scope of Innovation in Software Rejuvenation

5. Experimental Study

Experimental setup includes the technologies used for formulating the proposed solution is as follows. Linux kernel being one of the powerful platforms for media processing, is used as operating system. For a simple integrated development environment Qt framework is used. C++ language is used for programming. In order to perform media processing Gstreamer framework is used as well as to perform networking operations GUPnP framework is used.

The technologies here that are used are chosen such that the end product becomes completely open source and does not have any proprietary issues associated with it. The solution has been neatly packed with tools and technologies with this criterion in mind. These technologies also have the capabilities to include the above mentioned features. The features
are to be implanted in such a way that it eradicates the difficulties faced in the existing tools.

GStreamer is used as the backend for streaming the data from one end to another. Qt is used as the front end to create the user interface of the application. UPnP is used at the backend to provide network connectivity of the system. Hence as mentioned earlier, the above specified tools are completely open source, thus the end product is expected to be non-proprietary. For solving software aging problem, software rejuvenation is implemented.

The idea is to create a server and client which would take care of the communication between users. The server has the priority to add and remove clients dynamically in the system. Once the client is added to the network, the client puts in a request for the service to the server. The server analyzes the request and displays the services available for that client. The client chooses the services required from the list of available services. The server now provides the chosen service to the client.

The role of the server in the system is clearly depicted with the help of the Use case diagram below. The main role of the server is to administer the clients over the network and cater the service that they request for. The server should have the capacity to handle the load. The server generates the port number using the Random Number Generator system. The initial task of the server is to multicast itself. It has to advertise itself to all the clients over the network. It should also list out all the services it caters. Then once it receives a request from the client, it adds the client to the network. It processes the request and responds to it. The parameter that comes along with the request of the client is checked by the server. If it encounters any incorrect parameters, the client will be terminated. Otherwise the output is streamed to the client. And finally the client is removed from the network.

The client forms the important part of the system. It searches for the available servers over the network. As a result of the hunt, it enlists all the available servers. Then a particular media server is identified. The client now sends the request to the server along with the necessary parameters required for the streaming. Now the request is processed if the parameters encountered are valid. Finally client is removed from the network as the process is over. There can be multiple clients over the network. The clients can be added and removed dynamically. But server is the decision authority. In this system, any machine can start the conference; i.e. any machine can be the server and any machine can be the client. The only constraint is that the server system should run on Linux

The integrated model of server and client of the system is depicted below. This diagram describes the flow of data between client and the server. Once the client decides on the server, it sends a request to the server to stream the media. Along with this request, the client sends the required parameters for the server to add the client over the network. The server verifies the parameters and then adds the client to the network. Hence the client is approved if the parameters are correct.

Rejuvenation was implemented at the server side programmatically at the application level.

Once the client is connected, server starts streaming the data at the client. At the server, the client connected port and ip address is stored and the networking part is refreshed and also the streaming
Elements are refreshed by gracefully terminating and recreating them. The client side since the bit streams are saved and played; the client player is also gracefully terminated and recreated periodically.

The above is achieved using a timer. The timer will be started when the application starts and it will timeout for every configured rejuvenation time. After the first timeout, a rejuvenating process is created to check if any other process is running which is responsible for networking, if yes then it will check to check if any other process is running which is responsible for streaming, if yes it will snooze itself or go to sleep mode for the configured amount of time.

After the configured time, rejuvenating process gets active and it will check again whether any process is active or not for every configured (snooze) time.

If any time it finds there is no active process, it will restart networking or streaming part of the server and reset the rejuvenation timer to initial state.

Some of the dynamically linked shared object libraries (.so) which are placed in memory initially when an application makes connection with a client are later in an ideal state. Rejuvenation is applied here by clearing those object libraries from the memory when it’s in ideal state. The software elements created are responsible for performing media streaming. As a rejuvenating process such elements are deleted when a client is not connected and created once the client get connected.

In this application, rejuvenation is implemented in the application level. Some of the good things achieved when this kind of rejuvenation implemented is Rejuvenation becomes system independent as it is independent of kernel and OS, Extendibility increases that is rejuvenation can be extended to required level. Because its application specific is any other software which performs rejuvenation kernel level may be used without affecting the existing one. Some of the undesired effects will be Code Reusability that is this rejuvenating technique used here might apply only to the multimedia application.

Testing tools used are Valgrind. Valgrind is a tool which is used to give insights about the amount of memory used and, more about possible memory leaks in the program. Hence it is used to plot a graph against the time.

6. Results

Based on memory leakage, below graphs gives a comparative analysis between rejuvenated and non-rejuvenated software. Memory leakage was measured in terms of Megabyte using Valgrind tool. Time was measured in terms of minute so as to calculate the data corruption at equal intervals as before and after rejuvenation.
Once rejuvenation was implemented on Adhoc conferencing and Media distribution system it is observed that Corruptions were noticed initially but once rejuvenation was implemented the corruption and delay in the video displayed reduced. Memory usage was limited throughout the application running period. Memory leakage was controlled Frequency of Application failure or system hang was reduced. Observation reveals that for a larger multimedia system when rejuvenation is implemented it acts like a self-healing module.

7. References


Date of Conference: 2000 Page(s): 270- 279

[6] IBM Director Software Rejuvenation, IBM Corporation 2000, USA

