

Comparative Analysis of Media Pipeline Plug-in Architectures

Veronica Sentongo¹, H. Anthony Chan¹, Mqhele E. Dlodlo¹, and Keith Ferguson²

¹Department of Electrical Engineering, University of Cape Town, Rondebosch, South Africa

²CSIR, Meraka Institute, South Africa

Emails: sntver001@uct.ac.za; h.a.chan@ieee.org; Mqhele.Dlodlo@uct.ac.za; kferguson@csir.co.za

Abstract- The future of Next Generation Networks relies heavily on the effective operation of real-time applications. Media pipeline plug-in architectures have been introduced to provide the efficiency and improvements that these wireless applications require. The pipeline architectures exist for the efficient capture and processing of multimedia streams into a format suitable for transmission in the context of real-time processing. Numerous media pipeline plug-in architectures exist for processing audio and video applications. However, this research paper is a comparative analysis of only three of these architectures: *GStreamer*, *Microsoft DirectShow* and *RealNetworks Helix*. The project investigates the feasibility of implementing these architectures in South Africa.

Index terms- Media pipeline plug-in architectures, Real-time applications, Streaming audio/video

I. INTRODUCTION

Cellular technology has undergone a tremendous evolution from the original 1G analog technology to the current Next Generation (NGN) heterogeneous or 4G Networks. The future, however, of these 4G wireless technologies greatly depends on the success and improvement of real-time applications.

To improve on the efficiency and speed of real-time applications, media pipeline plug-in architectures have been introduced. The media pipeline architectures discussed in this project are of a “plug-in” nature. This means that the plug-ins can be designed remotely and seamlessly plugged in to the existing architecture to promote its efficiency and operation. In doing this, the modularity of the application’s architecture is eased [1].

In order to keep up with the fast evolving technologies of the world, South Africa has acknowledged the need to plunge into intensive research on how to improve the existing multimedia technologies. The purpose of this particular research project is to investigate three different media pipeline plug-in architectures that have successfully been implemented elsewhere in the world and determine which of these architectures would be most feasible in South Africa.

In particular, this project focuses on the possibility of implementing these architectures in highly congested and low infrastructure networks. This thesis is a comparative analysis of three media pipeline plug-in architectures. These include:

- GStreamer
- Microsoft DirectShow
- RealNetworks Helix.

II. PLUG-IN ARCHITECTURES

A media pipeline refers to a chain of processing events such that the output of one element (program) becomes the input of the next element (program) [2].

Plug-in architectures refer to parts of the application’s software (codec) that can be separately developed, improved or modified to extend the application. This codec is designed to be seamlessly plugged back into the host architecture and actuate the required improvements [1]. They can be developed separately and designed to be compatible with the existing application.

III. ANALYSIS OF EACH ARCHITECTURE

• *GStreamer*

GStreamer is an object-oriented library that has been developed mainly for Linux operating systems. Unlike its contending operating systems, Windows and Mac, Linux does not have an advanced support for playback, capture and processing of multimedia content. Therefore, in order to compete on a professional level with these other operating systems, GStreamer has been developed [3].

According to [4], GStreamer is a highly modular framework that can seamlessly incorporate new plug-in modules. It eases development of applications that handle audio, video and data content. The GStreamer framework is built entirely on a library of plug-ins which provides the necessary functionality to the architecture. The plug-in libraries are dynamically loaded to support a wide range of input/output drivers and codecs.

Figure 2 below shows the simple operation of GStreamer being used to download an MP3 file [5] [6]. The three step process starts at the file plug-in where the file is read from the computer’s hard drive. The file is then transferred through the source and sink pads to the decoder where the MP3 file is broken down into PCM samples. The PCM samples are then transferred to the ALSA (Advanced Linux Sound Architecture) which sends the PCM samples to the computer’s speakers through which the song is heard.

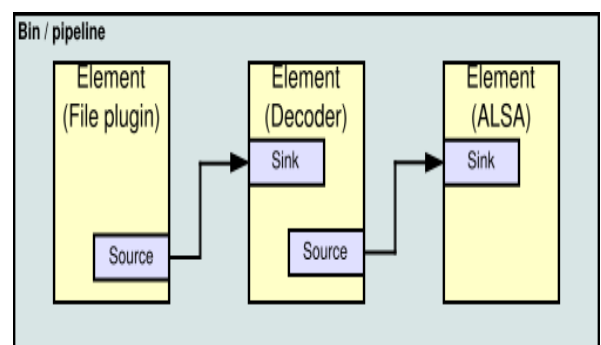


Figure 2: Downloading an MP3 file using GStreamer [5] [6]

IV. PROPOSED SCHEME

• Microsoft DirectShow

Microsoft DirectShow is a multimedia architecture that has been designed for streaming media by Microsoft for the Windows operating system. This media pipeline plug-in architecture is an extensible filter-based framework. Filters have a number of input and output pins which connect them together. This design ensures that filters can be connected together in many different ways to achieve different tasks. [7].

Figure 3 below shows a typical playback graph for playing an AVI file along with the filters that perform each step. The basic steps are: Read raw data from a file in a stream of bytes using the *source filter*. Then using an *AVI splitter filter*, examine the AVI headers, and parse the byte stream into separate video frames and audio samples. Decode the audio and video frames using various *decoder filters*, depending on the compression format, and decompress the data. The *video renderer filter* draws the video frames. Finally, send the audio samples to the sound card using the *default DirectSound device filter*.

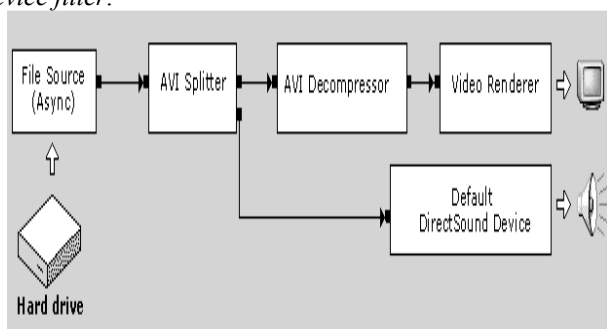


Figure 3: Typical playback graph for an AVI file showing the filters involved [8].

• RealNetworks Helix

RealNetworks Helix is an open standard for multimedia development and deployment. It is a universal media delivery platform that contains standardized API's that allow playback of most multimedia formats [10]. RealNetworks Helix runs on a powerful server called the **Helix Server**. According to [11], *RealNetworks Helix Server Version 11* is the most powerful server software available for streaming media files across an intranet or the Internet. It can stream on-demand clips and broadcast live events in more media formats than any other media server and can function on various Operating Systems.

The Helix Server delivers clips and streams but does not create them. As a result, the structure of the RealNetworks Helix server is very simplistic in design. For delivering on-demand clips, the three major steps involved are shown in figure 4 below [9]. Encode a clip with an encoding tool, stream the clip through a streaming server and finally, play the clip with a media player.

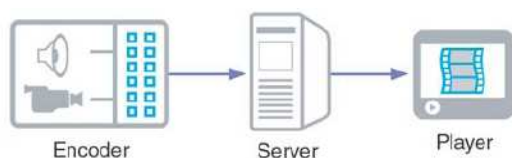


Figure 4: Delivering an on-demand video clip [9]

"This work was funded by the DST-Innovation Fund on project no. T70026: Bandwidth Adaptive Real-Time Video Broadcasting over Internet."

In order to effectively analyse these architectures, numerous simulation experiments are required. The relevant SDK for each architecture will be downloaded from the Internet. A test real-time video processing pipeline will be written with timing test points to evaluate the performance using C++ programming language. Particular interest will be taken to the multithreading approaches taken by each architecture and the speed at which the test media pipeline takes to execute. A comparative analysis of the results will be made thereof. On completion of the research, it should be evident whether these pipeline architectures will be easily and effectively implemented in South Africa.

V. CONCLUSION

Media pipeline plug-in architectures have been designed as an innovative means of improving the capture and processing of streams into a format suitable for transmission of audio and video content. This has greatly improved the effective operation of real-time applications. Considering the network constraints and bandwidth limitations, this research project investigates the possibility of implementing the most feasible pipeline architecture in South Africa. The project is also meant to identify the conditions under which this preferred architecture will operate best.

VI. REFERENCES

- [1] Jeffrey Stylos *User Interface Software, Fall 2004. Plug-in Architectures*
- [2] Pipes: A brief Introduction. [ONLINE] Available: <http://www.linfo.org/pipe.html>
- [3] GStreamer framework [ONLINE] Available: <http://www.linux.com/articles/51005>
- [4] *Taymans et al.*, GStreamer Application Development Manual (0.10.17.1)
- [5] GStreamer: Open Source Media Framework [ONLINE] Available: <http://gstreamer.freedesktop.org/>
- [6] [ONLINE] Available: <http://en.wikipedia.org>
- [7] Direct Show [ONLINE] Available: <http://www.gdcl.co.uk/dshow.htm>
- [8] DirectShow system overview [ONLINE] Available: [http://msdn.microsoft.com/en-us/library/ms783354\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms783354(VS.85).aspx)
- [9] HELIX SERVER ADMINISTRATION GUIDE Helix™ Server Version 11, Revision Date: 2 September 2005
- [10] Linux.Com: Open Source Helix [ONLINE] Available: <http://www.linux.com/articles/24155>
- [11] Media Servers. Helix Server v12 [ONLINE] Available: <http://www.realnetworks.com/info/helixserverv12.html>

BIOGRAPHY

Veronica Sentongo obtained her BSc (Elec Eng) degree from the University of Cape Town in 2007. She is currently in the first year of her MSc (Elec Eng) degree majoring in Telecommunications. Her research is sponsored by the Council for Scientific and Industrial Research (CSIR) through the DST-Innovation Fund.