Modeling and Implementation of a Multicore Architecture for Real-Time Videoconference in Embedded Systems

Ronaldo Husemann, Carlos Haas Costa, Alfredo Capella, Valter Roesler

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{rhusemann, chcosta, agcjuniour, roesler}@inf.ufrgs.br

Abstract. Emergent videoconference applications can be found in a large set of products like videophones terminals, 3G cell phones and multimedia live chat softwares. This paper presents the modeling of an architecture for real-time videoconference applications which can be adopted in Personal Computers and also in embedded systems. Some practical experiments have been performed over a real DSP hardware platform in order to validate the proposed architecture.

1. General Information

Modern real time multimedia systems supporting interactive transmission of audio and video can be used to reduce real distances through the use of virtual meeting environments (Synnes, 2001). Particularly videoconference systems must respect some important constraints, like guarantee of a constant data flow in the transmission channel (fidelity), automatic audio and video synchronization (fidelity) and small end-to-end delay (Schall, 2006). All these dependences can be considered as temporal constraints.

Eventually, due to limitations in computing power or network bandwidth, some parts of the audio and video information can be discarded to guarantee small end-to-end delay in detriment of the video quality (Steinmetz, 1996). Considering that most of videoconference applications can be listed as soft real-time, generally the video quality is reduced in order to keep the real-time compromise.

During the last years, some important works have been proposed to provide real-time support for multimedia applications. Some examples are (Tanguay, 2004), (Rizvanovic, 2006) and (Kao, 2005)

Continuous evolution in high performance processors aims to add complexity in the same chip area and also increase the maximum chip speed, resulting in development of multiprocessing chips (MPCs) (Goren, 2003). An efficient algorithm for the multicore technology must consider this parallelism condition during application modeling. Embedded chips developed for multimedia market follow the same direction of using dual or quad core architecture (O’Driscoll, 2000). Based on that, we propose a parallel architecture aiming real-time videoconference applications.

To validate the proposed architecture, we performed some practical experiments to identify the execution time information in different routines of a videoconference application which was implemented over a real DSP hardware platform.
The selected platform to validate our architecture was based on the Davinci family technology (multimedia DSP solution from Texas Instruments). Hardware executes a real-time Linux application which manages different internal threads and user interface.

2. Proposed architecture

To model the proposed architecture we used the Unified Modeling Language Real Time (UML-RT) (Douglas, 1998). It was developed to support multiple users, who can send and receive simultaneously multimedia contents in a multipoint environment. In our model, any user’s data is transmitted via unicast to a multicast server, which combines the participant’s flows in one single flow and sends it through multicast to all users. Two separate entities has been created. One of them is the “Presenter”, responsible for raw data capture, encoding and unicast transmission. The other is the “Viewer”, responsible for receiving the multicast signal, decoding and presenting. The following Class Diagrams (Figure 1 and 2) represent respectively Presenter and Viewer models.

![Figure 2. Presenter’s internal modules.](image1)

The central elements of this application are the Stream and Media classes. The Capture class has the attributes related to the camera or microphone drivers, decoded data and methods for conversion are necessary. It interacts with the system through the Queue class. The method that access a queue “Read()”, and consumes the data produced on capture, is the “Encode()”, from Encoder class. Before transmission, encoded frames are stored in a second queue (transport stream). The concept and relations of classes described here are presented below, but the messages communication between internal modules and sequence of operation are shown in Sequence Diagram.

![Figure 2. Modeling of Viewer’s internal modules.](image2)

Finally, the classes interactions including temporal constraints, are described through the following UML-RT diagram (RT Sequence Diagram). The time constraints, when necessary, are positioned over correspondent messages that start each operation. In the Viewer software, when a user starts the application, the first methods are the Queue initialization methods. After that, the network is initialized.
A complete cycle limit time is determined by the frames per second (fps) rate chosen for the application. In the example depicted in Figure 3 and 4, the rate of 25 fps led to the system a soft real time compromise to capture a frame and send it each 40 milliseconds, while receive and show in the other side with the same constraint.

The sequence diagram from the Presenter software is analogous to the Viewer. Basically the same constraints are involved, and the same pipelined parallelism can be explored.
3. Practical Experiment

In order to validate the architecture we implement our proposal in a Davinci DSP evaluation board. One of the main reasons to select that is because Texas Davinci CPU adopts a dual core architecture: one general purpose core ARM9 and one dedicated DSP core. Additionally Davinci chip incorporates an OSD module, responsible by video exhibition. In our study case we develop a single videoconference application that links two equipments by a unicast network. The multimedia specifications are:

- MPEG-4 video encoding and decoding using resolution of 640x480 pixels with 25 fps. Video network bitrate is 1500kbps;
- G.711 video encoding and decoding. Audio network bitrate is 64kbps.

The main observed temporal constraint of our real-time videoconference application is the video cyclic time (40ms to support actualization of 25 frames per second). We can observe that in that case study the desired temporal constraint has been fulfilled. The bigger registered delay was 38.724ms for the Davinci DSP core (lower then 40ms).

4. Conclusions

In this paper it was presented a UML-RT modeling of a software architecture, which can help development of real-time multimedia applications exploring parallelism in a multicore platform. Developers can use this strategy to eliminate application’s bottlenecks by distributing CPU load on the critical path. Performed experiments in an embedded platform validate the proposal in the adequate fulfillment of temporal requirements of a practical real-time videoconference application.

References

Steinmetz, R., Human perception of jitter and media synchronization. IEEE Journal on Selected Areas in Communications. v14 i1. 61-72.