

UMA IMPLEMENTAÇÃO DO COMPONENTE MEDIA PROCESSING USANDO A BIBLIOTECA LIBVLC PARA O MIDDLEWARE GINGA

TROJAHN, Tiago Henrique
Universidade Federal de Pelotas

MATTOS, Júlio Carlos Balzano
Universidade Federal de Pelotas

AGOSTINI, Luciano Volcan
Universidade Federal de Pelotas

GONÇALVES, Juliano Lucas
Universidade Federal de Pelotas

DA ROSA JUNIOR, Leomar Soares
Universidade Federal de Pelotas

1 INTRODUÇÃO

O *middleware* é uma camada de *software* intermediária presente no receptor da Televisão Digital e Interativa - TVDI, o terminal de acesso (MORENO 2006). Sua função é abstrair as particularidades do *hardware* do terminal receptor, definindo uma máquina virtual com *interfaces* de programação padronizadas que são utilizadas para construir as aplicações de TVDI. O *middleware* viabiliza assim a execução de uma mesma aplicação em aparelhos receptores com *hardware* e sistema operacionais distintos. Isso é de extrema importância em um ambiente de transmissão de TVDI, visto que uma mesma aplicação é transmitida para milhares de receptores, com diferentes características e fabricantes.

Atualmente o Ginga, *middleware* do Sistema Brasileiro de TV Digital - SBTVD, possui dois sistemas separados, o GingaNCL (SOARES, RODRIGUES E MORENO 2007) e o GingaJ (FILHO, LEITE E BATISTA 2007). O GingaNCL suporta a execução de aplicações declarativas baseadas na linguagem *Nested Context Language* - NCL, enquanto o GingaJ suporta o desenvolvimento de aplicações procedurais através da linguagem Java. A separação dos ambientes vem sendo um limitador para os desenvolvedores, porque além de se preocuparem com o tipo de aplicação a ser desenvolvida (declarativa ou procedural), também se torna necessário instalar o ambiente de execução que dê suporte à mesma, acarretando um esforço adicional, o qual poderia ser evitado com a existência de um único sistema com suporte a ambos os tipos de aplicação.

Em virtude dessas dificuldades, em fevereiro de 2009, foi criado o projeto Ginga *Code Development Network* – GingaCDN (LAVI 2009). O objetivo do GingaCDN é produzir um núcleo comum chamado de Ginga *Common Core* - GingaCC, que dê suporte aos sistemas GingaNCL e GingaJ. O *middleware* Ginga, ao qual o projeto GingaCDN se propõe a desenvolver, é ilustrado na Figura 1.

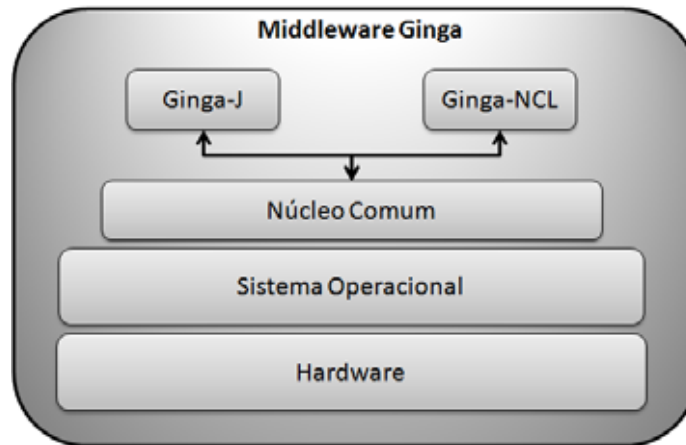


Figura 1 – Representação do *middleware* Ginga em implementação pelo projeto GingaCDN.

Um dos componentes formadores do GingaCC é o Media Processing, o qual é responsável por decodificar os fluxos de vídeo recebidos do componente demux. O objetivo deste trabalho é o desenvolvimento desse componente utilizando a biblioteca LibVLC (LIBVLC 2010).

Para tanto, esse trabalho está organizado como segue: No capítulo 2, é apresentada a metodologia utilizada no desenvolvimento do trabalho. O capítulo 3 mostra os resultados obtidos. Por fim, o capítulo 4 apresenta as conclusões e alguns possíveis trabalhos futuros.

2 METODOLOGIA

Essa seção apresenta uma visão geral do componente Media Processing e também da biblioteca gráfica LibVLC utilizada em seu desenvolvimento.

2.1 O Componente Media Processing

O componente Media Processing está diretamente conectado com outros dois componentes do GingaCC, o Demux e o Graphics, descritos abaixo:

- **Demux:** Componente responsável por separar os vários componentes do fluxo de transporte (enviado pela emissora) em fluxos básicos, como vídeo, áudio e legendas, e enviá-los aos componentes encarregados de processar o fluxo. O Media Processing recebe fluxos de vídeo e de legendas do Demux.
- **Graphics:** Componente responsável por exibir na tela o resultado da decodificação realizada pelo Media Processing. É o último componente envolvido na exibição de mídias.

Assim como os demais componentes do projeto GingaCDN, o Media Processing está sendo desenvolvido na linguagem C++ e utilizando um modelo de componentes chamado de FlexCM. O Media Processing utiliza-se, ainda, da biblioteca gráfica LibVLC para a realização da decodificação do fluxo de vídeo.

A implementação atual do Media Processing, além dos métodos básicos de carga, reprodução e pausa na decodificação de fluxos de vídeo, também realiza uma série de outras funcionalidades, como descritas abaixo:

- Decodificação de diversos padrões de codificação de vídeo, além do padrão H.264/AVC definido para o SBTVD.

- Carga, seleção e exibição de legendas nos formatos SubRip (SRT) e *Advanced Subtation Alpha* (ASS).
- Retorno de diversas informações sobre o fluxo de vídeo, como razão de aspecto, taxa de quadros por segundo, resolução, duração (se disponível) entre outros.

A seguir é apresentada a biblioteca gráfica LibVLC e suas principais características.

2.2 A Biblioteca Gráfica LibVLC

A LibVLC é uma biblioteca gráfica desenvolvida pela VideoLAN sobre a licença GNU *General Public License* (GPL) versão 2. No desenvolvimento desse trabalho foi utilizada a versão 1.1.0 da biblioteca LibVLC. Optou-se pela utilização dessa biblioteca, pois além de possuir código livre, apresenta características que auxiliam no desenvolvimento do componente, tais como:

- Compatibilidade com vários tipos de mídia, incluindo o padrão H.264/AVC do SBTVD.
- Portabilidade com um grande número de sistemas operacionais, tais como o Microsoft Windows, GNU Linux, MacOS, BeOS e o FreeBSD.
- Suporte a vários tipos de saída de vídeo, como DirectX, X11, entre outros.
- A biblioteca em si foi escrita na linguagem C, oferecendo um alto desempenho necessário para processar mídias.
- Possui suporte nativo a operações *multithread* como, por exemplo, a separação entre o fluxo principal de controle e a exibição de vídeo.

Na seção 3 são apresentados os testes, aos quais, o componente foi submetido com o intuito de avaliar o seu funcionamento e eficiência.

3 RESULTADOS E DISCUSSÕES

Os testes de desempenho, ao qual o componente Media Processing implementado foi submetido, consiste na avaliação da taxa de uso de processador e do custo de memória necessários para a operação do componente.

Os testes foram realizados em um micro-computador equipado com um processador Intel Core 2 Quad Q6600 com 2GB de memória RAM e sistema operacional Ubuntu 10.04. Foram utilizados quatro vídeos progressivos disponíveis em três resoluções diferentes: 848x480 (480p), 1280x720 (720p) e 1920x1080 (1080p). Todos os vídeos apresentavam uma taxa de 30 quadros/segundo, padrão definido pelo SBTVD.

Cada amostra do teste foi coletada com o aplicativo Procps, que verifica dados referentes aos processos em execução no sistema operacional. O teste consiste em três execuções dos vídeos com duração de três minutos cada, perfazendo 540 amostras para cada vídeo. No total, as 6480 amostras foram analisadas, excluindo-se os 5% dos maiores e dos menores valores, para fins de normalização. A Figura 2 apresenta os resultados obtidos para uso de processador e memória ao se executar a bateria de testes do componente Media Processing no micro-computador supracitado.

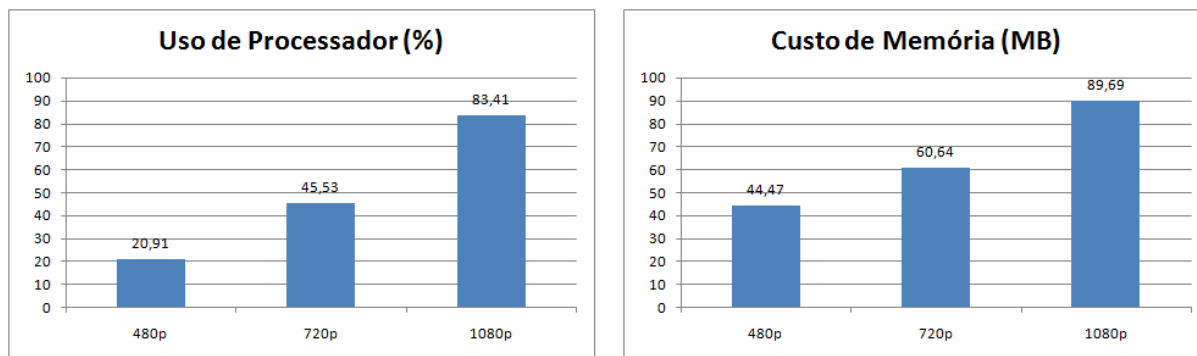


Figura 2 – Média aritmética dos resultados obtidos nos testes de desempenho.

Os testes demonstraram que o componente Media Processing teve uma taxa média de uso de processador de 49,95%, sendo capaz, inclusive, de reproduzir vídeos em alta resolução (720p e 1080p). Em termos de custo de memória, o Media Processing foi capaz de decodificar todos os fluxos entrantes com menos de 100MB de memória principal, sendo o consumo médio em todas as resoluções de apenas 64,93MB. Isso é um fator importante, pois permite a utilização do componente em arquiteturas que disponham de pouca memória.

4 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou a implementação do componente Media Processing do middleware Ginga utilizando a biblioteca gráfica LibVLC. Para avaliar o desempenho da implementação, foi realizada uma bateria de testes consistindo na reprodução de fluxos de vídeos em diferentes resoluções. Os resultados obtidos mostram que o componente Media Processing apresentou um consumo de memória inferior a 100MB para reprodução de vídeos em alta resolução. Esses resultados, mesmo que ainda sejam iniciais, já possibilitariam sua adaptação em *hardware* para uso no terminal de acesso brasileiro e em aparelhos celulares.

Futuramente, pretende-se adicionar o suporte a fluxos de áudio ao componente. Também está prevista a sua integração com outros componentes desenvolvidos pelo projeto GingaCDN, possibilitando a criação do *middleware* Ginga único.

REFERÊNCIAS

- FILHO, G. L. S., LEITE, L. E. C. e BATISTA, C. E. C. F. Ginga-J: The Procedural Middleware for the Brazilian Digital TV System. **Journal of the Brazilian Computer Society**, p.47-56, 2007.
- LAVID. GingaCDN. **GingaCDN, Ginga Code Development Network**, 2009. Disponível em: <<http://ginga.lavid.ufpb.br/>>. Acesso em: 28 Julho 2010.
- LIBVLC. “libVLC – VideoLAN Wiki”, 2010. Disponível em: <<http://wiki.videolan.org/Libvlc>>. Acesso em: 28 Julho 2010.
- MORENO, M. F. **Um middleware declarativo para Sistemas de TV Digital Interativa**. Dissertação (Mestrado em Informática), PUC-Rio, Rio de Janeiro, 2006.
- SOARES, L. F. G., RODRIGUES, R. F. e MORENO, M. F. Ginga-NCL: the declarative environment of the Brazilian Digital TV System. **Journal of the Brazilian Computer Society**, p.37-46, 2007.