

UMA IMPLEMENTAÇÃO DO COMPONENTE MEDIA PROCESSING COM SUPORTE A ACELERAÇÃO EM HARDWARE PARA O MIDDLEWARE GINGA

**GONÇALVES, Stèphano Machado Moreira¹; TROJAHN, Tiago Henrique¹;
GONÇALVES, Juliano Lucas¹; AGOSTINI, Luciano Volcan¹;
ROSA JUNIOR, Leomar Soares²**

¹Universidade Federal de Pelotas, Bacharelado em Ciência da Computação; ²Universidade Federal de Pelotas, CDTec - Computação.
smmgoncalves@inf.ufpel.edu.br, leomarjr@inf.ufpel.edu.br

1 INTRODUÇÃO

A Televisão Digital e Interativa agrega elementos de hardware típicos da televisão tradicional com outros antes só encontrados em computadores. O telespectador passa agora a ter papel ativo na programação que anteriormente apenas assistia. Usando o controle remoto, o telespectador pode interagir diretamente no que assiste, quer seja escolhendo o ângulo para assistir um jogo de futebol ou para fazer compras. O Ginga, middleware do Sistema Brasileiro de TV Digital - SBTVD é responsável por essa interação. Ele é composto por três subsistemas, o GingaNCL, GingaJ e Ginga *Common Core* (GingaCC). O GingaNCL (SOARES; RODRIGUES; MORENO, 2007) desenvolvido pela PUC-Rio (PUC-RIO, 2011) é o subsistema responsável por suportar aplicações escritas na linguagem NCL/LUA. Por outro lado, o GingaJ (GUIDO; LEITE; BATISTA, 2007) desenvolvido pela UFPB (UFPB, 2011) é responsável por suportar aplicações escritas na linguagem Java. O GingaCC é um módulo que funciona como uma ponte entre os subsistemas GingaNCL e GingaJ, ou seja, não se comunica diretamente com as aplicações. O GingaCC foi um esforço oriundo do projeto GingaCDN (GINGACDN, 2010) e foi dividido em treze módulos ou componentes e cada um deles é responsável por uma tarefa específica. O componente responsável pela decodificação de fluxos de mídia, tais como vídeo, áudio e legendas, é chamado de Media Processing.

Este artigo tem por objetivo apresentar uma implementação do Media Processing para o *middleware* Ginga. Esta implementação provê suporte a decodificação com aceleração em *hardware* utilizando unidades gráficas de processamento (GPU). Com o intuito de avaliar esta nova versão do Media Processing, foram realizadas uma série de testes de desempenho em três computadores, utilizando vídeos de diferentes resoluções. Os principais resultados obtidos são apresentados e discutidos nesse artigo.

A seção 2 apresenta a metodologia utilizada no desenvolvimento desse trabalho. Na seção 3 os resultados obtidos são apresentados e discutidos. Por fim, a seção 4 apresenta as conclusões e possibilidades de trabalhos futuros.

2 METODOLOGIA (MATERIAL E MÉTODOS)

A biblioteca libVLC (VIDEOLAN, 2010) na versão 1.1.5 foi utilizada para implementar o módulo Media Processing. O Media Processing foi desenvolvido para ser capaz de fornecer todas as características básicas esperadas de um decodificador de vídeo, incluindo os métodos:

- Play - A funcionalidade mais utilizada e fundamental de todos os decodificadores de mídia. É responsável por começar toda a operação de

decodificação e fornecer o fluxo decodificado para o telespectador ou para outro aplicativo.

- Pause - Responsável por pausar a operação sem desalocar os recursos para continuar a decodificação.
- Stop - Método de finalização, usado para desalocar todos os recursos e desligar o processo de decodificação.

No entanto, o SBTVD exige mais recursos do Media Processing que estes métodos básicos. Por exemplo, o Media Processing deve fornecer dados para outros módulos e, às vezes, diretamente para os aplicativos instalados pelos telespectadores. Assim, o Media Processing implementado neste trabalho apresenta um controle de exceção em todos os métodos fornecidos para os outros módulos ou aplicações. Alguns tratamentos de exceções mais complexos, como problemas de alocação, são realizados diretamente pela libVLC.

Os testes de desempenho levaram em consideração a análise do uso de processador e de memória necessário para a decodificação dos vídeos. Foram usadas as animações Big Buck Bunny (BLENDER INSTITUTE, 2008), Elephants Dream (ORANGE OPEN MOVIE PROJECT, 2006) e Sintel (BLENDER INSTITUTE, 2010) para fluxos de entrada em três diferentes resoluções: 848x480 (480p), 1280x720 (720p) e 1920x1080 (1080p). Cada fluxo de entrada de áudio e vídeo foi reproduzido durante nove minutos. Este experimento foi repetido três vezes, a fim de obter resultados mais precisos. As amostras foram geradas com a aplicação procps em um sistema operacional Ubuntu 10.04 com as configurações de instalação recomendadas.

Um processador Intel Core i5 760 com 2,8 GHz e 4 GB de memória RAM foi a plataforma básica para o experimento. Foram utilizadas três configurações de placas de vídeo. Para facilitar a identificação adotou-se a seguinte nomenclatura:

- Computador A - Sem qualquer placa de vídeo dedicada e utilizando apenas a decodificação baseada em software do Media Processing.
- Computador B - Foi usada uma placa gráfica XFX Geforce 9500GT (NVIDIA, 2011) com 512 MB de memória DDR2 RAM e decodificação acelerada de hardware do Media Processing ativada.
- Computador C - Foi usada uma placa gráfica Zotac Geforce GTX 470 (ZOTAC, 2011) com 1280 MB de memória RAM DDR5 e a decodificação acelerada de hardware do Media Processing ativada.

3 RESULTADOS E DISCUSSÃO

Os resultados obtidos apresentaram um incremento de utilização do processador quando se aumenta a resolução de vídeo, o que pode ser explicado pelo aumento dos pixels que precisam ser processados.

O computador A e o computador B apresentaram um desempenho equivalente quanto ao uso do processador. Já o computador C apresentou um uso de processador bem inferior aos outros dois. O poder de processamento da Geforce GTX 470 quando comparado com a Geforce 9500GT sugere que a redução do uso do processador principal está diretamente relacionada ao aumento do poder de processamento da placa de vídeo. Além disso, a equivalência dos testes entre os computadores A e B sugerem que a placa de vídeo do computador B não é capaz de reduzir a carga do processador principal de uma maneira satisfatória ao decodificar os vídeos.

Por outro lado, no que diz respeito à memória, as implementações utilizando a aceleração de hardware em placa de vídeo apresentaram um consumo maior de memória. Este comportamento pode ser explicado pela metodologia de decodificação que a libVLC adota. Basicamente, o quadro codificado é enviado para a placa de vídeo e o quadro decodificado é mantido em memória enquanto não é exibido. O custo de memória aumenta ainda mais porque vários quadros são mantidos na memória para prevenir problemas de sincronização. Tal como acontece com os testes de uso do processador, o consumo de memória se elevou como o aumento da resolução de vídeo.

As figuras 1 e 2 apresentam, respectivamente, os resultados obtidos de uso do processador, em porcentagem, e o uso de memória, em megabytes, para os testes de desempenho

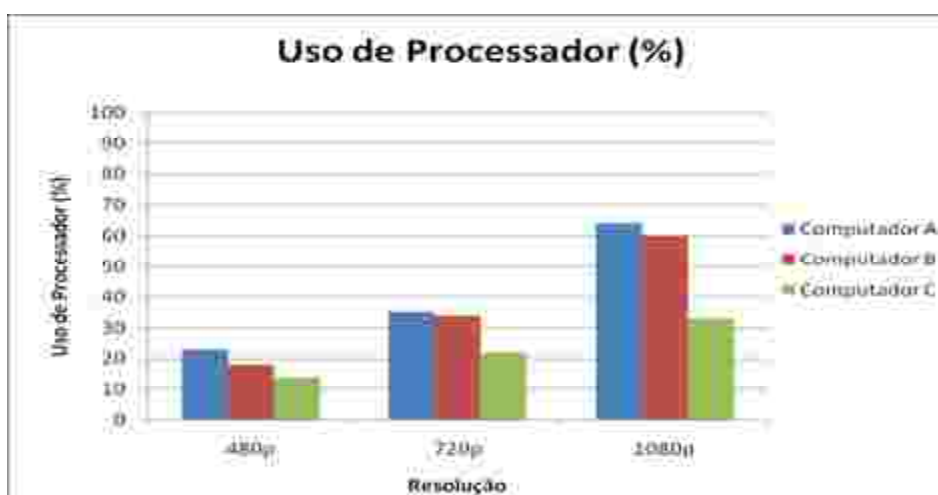


Figura 1 - Uso do processador (%) do Media Processing

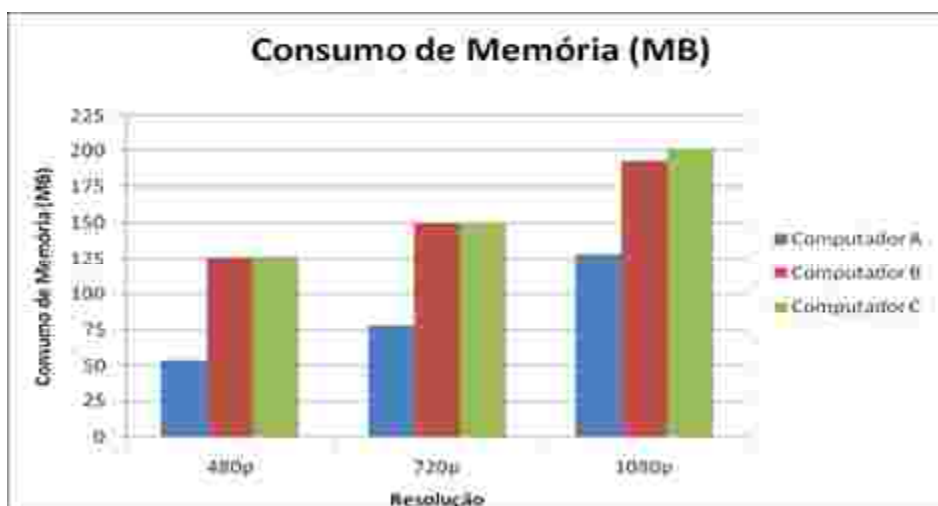


Figura 2 - Consumo de memória (MB) do Media Processing.

4 CONCLUSÃO

Este trabalho apresentou um componente de decodificação, chamado Media Processing, para o middleware Ginga. Foram realizados testes de desempenho em três computadores com configurações diferentes para avaliar o comportamento do componente quando o suporte a aceleração em hardware está ativado. Os testes de desempenho do Media Processing demonstram que melhores resultados podem ser

obtidos em termos de uso de processador quando se dispõe de uma placa de vídeo com grande poder de processamento e suporte a aceleração de hardware ativada.

Ao usar uma placa de vídeo com menor poder de processamento, a vantagem torna-se insignificante. Quanto à memória, o uso da aceleração em hardware para as resoluções 480p e 720p, aumentou duas vezes o custo de memória quando comparada com a solução em software. Portanto, se o dispositivo apresentar uma placa de vídeo compatível e memória suficiente, a melhor solução consiste em ativar a aceleração em hardware de placa de vídeo. Por outro lado, se a memória é limitada, a decodificação via software é a melhor alternativa. Contudo, esta estratégia acarretará em um aumento significativo no uso do processador.

Para trabalhos futuros pretende-se fazer um teste mais amplo de desempenho usando outras otimizações baseadas na arquitetura do processador. Pretende-se também usar o Media Processing em um dispositivo portátil para analisar o seu comportamento durante a execução de fluxos de vídeo com resoluções diferentes.

5 REFERÊNCIAS

BLENDER INSTITUTE. “Sintel”, 2010. Disponível em: www.sintel.org. Acesso em: 20 de fevereiro de 2011.

GINGACDN. Ginga Code Development Network. Disponível em: <http://ginga.lavid.ufpb.br/>. Acesso em 14 de abril de 2011.

GUIDO, L. S.; LEITE C.; BATISTA, C. E. C. F.. “Ginga-J: The Procedural Middleware for the Brazilian Digital TV System”. **Journal of the Brazilian Computer Society**, 2007, vol.12, pp.47-56.

ORANGE OPEN MOVIE PROJECT. “Elephants Dream”, 2006. Disponível em: www.elephantsdream.org. Acesso em: 2 de março de 2011.

PUC-RIO. Pontifícia Universidade Católica do Rio de Janeiro, 2011. Disponível em: <http://www.puc-rio.br>.

SOARES, L. F. G., RODRIGUES, R. F., MORENO, M. F.. GingaNCL: the declarative environment of the Brazilian Digital TV System. **Journal of the Brazilian Computer Society**, Vol.1, 37-46, 2007.

UFPB. Universidade Federal da Paraíba, 2011. Disponível em: <http://www.ufpb.br>. Acesso em: 15 de julho de 2011.

VIDEOLAN. “libVLC – VideoLAN Wiki”, 2010. Disponível em: wiki.videolan.org/LibVLC. Acesso em: 22 de junho de 2011.

NVIDIA. Geforce XFX 9500GT. Disponível em: <http://xfxforce.com/pt-br/products/graphiccards/9series/9500gt.aspx?lang=pt-br> Acesso em: 11 de julho de 2011.

ZOTAC. GeForce Zotac GTX 470. Disponível em: <http://www.Zotacusa.com/amp-geforce-gtx-470-1280mb320-bit-656mhz-3402mhz-zt-40201-10p.htm>. Acesso em: 10 março de 2011.