Proposta e Avaliação de Recriptografia Assistida por GPU Aplicada a Servidores Escaláveis de Distribuição de VoD

Leandro A. S. Gomes, Bruno S. Neves e Leonardo B. Pinho

Engenharia de Computação – Universidade Federal do Pampa (UNIPAMPA) - Bagé
Caixa Postal 07 – 96.413-170 – Bagé – RS – Brasil

lasg1309@gmail.com, {bruno.neves,leonardo.pinho}@unipampa.edu.br

Resumo. *A construção de sistemas de distribuição de vídeo sob demanda baseados em servidores proxy envolve, entre outras, estratégias de transmissão segura do conteúdo, implicando em maior necessidade de processamento. Neste trabalho é proposta e avaliada a inclusão de um módulo de recriptografia em GPU dos trechos de vídeo a serem enviados de forma individualizada aos clientes, em contraste com sistemas baseados apenas em processadores de propósito geral padrões de mercado. Os experimentos preliminares realizados indicam que a vazão obtida com a GPU é significativamente superior à obtida com um processador multicore, sugerindo que as GPUs tendem a ser uma alternativa de melhor custo-benefício para esta aplicação.*

1. Introdução

Nos últimos anos, a inter-relação entre três termos iniciados com a letra P tem sido foco crescente de estudos da comunidade envolvida em pesquisa na área de arquiteturas de sistemas computacionais: *Performance* (desempenho), *Power* (consumo energético) e *Price* (preço). Esta preocupação pode ser sintetizada com o termo: *high efficiency computing* (computação de alta eficiência). Atualmente, o uso, para processamento de propósito geral, das chamadas Unidades Gráficas de Processamento (*graphics processing units* - GPUs), presentes nas placas de vídeo modernas, vem chamando a atenção de vários pesquisadores como alternativa para aumentar o desempenho de aplicações e possibilitar alto desempenho com poucos recursos de hardware [Accelerators 2009]. Resumidamente, a ideia básica é a de aproveitar a significativa e crescente capacidade de processamento [NVIDIA], muitas vezes ociosa, oferecida por estes dispositivos de hardware originalmente introduzidos nos sistemas computacionais para o processamento gráfico, como elemento de co-processamento para execução eficiente de aplicações de propósito geral. Neste contexto se enquadra o problema da distribuição de vídeos sob demanda (VoD), uma vez que essa aplicação demanda processamento paralelo escalável à medida que o número de clientes cresce.

Na distribuição de VoD, as soluções com *proxies* são as de melhor relação custo-benefício por aumentarem a capacidade do sistema de distribuição, reduzindo a latência de início de exibição (em sistemas de distribuição de vídeo em geral, mas especialmente no caso de vídeo sob demanda) e a largura de banda necessária no enlace de saída do servidor principal. Os *proxies* são responsáveis pelo gerenciamento do conteúdo que deve ser mantido em cache, prevendo um novo acesso aos trechos de vídeo em função de novas requisições de vídeo. Além desta tarefa principal, que demanda processamento

significativo em função da necessidade de calcular em tempo real a prioridade de descarte dos trechos de vídeo, existe a necessidade de resolver paralelamente o problema do DRM (*Digital Rights Management*).

Basicamente, a questão do DRM consiste em proteger os direitos autorais correspondentes aos vídeos transmitidos pelo sistema, dificultando significativamente o acesso dos clientes não autorizados pelos detentores dos direitos autorais. Uma das maneiras de realizar a proteção dos direitos autorais é utilizar a criptografia, o que, para filmes muito populares que exigem maior escalabilidade do sistema, implica em múltiplas recriptografias dos trechos de vídeo a serem enviados de forma individualizada aos clientes.

Neste trabalho é proposto e avaliado empiricamente o emprego de GPUs em *proxies* de VoD como um módulo de recriptografia, usando o algoritmo de criptografia AES (*Advanced Encription Standart*), na sua variante ECB, que possui uma implementação sequencial de referência, na biblioteca OpenSSL, o qual possui um alto potencial para exploração do paralelismo, pois não possui dependência entre os blocos de entrada. Para tanto, a eficiência da proposta é contrastada com uma implementação paralela baseada em Pthreads, capaz de explorar múltiplos núcleos de processadores de propósito geral padrões de mercado.

Ao focar a aplicação de VoD, o presente trabalho se diferencia de trabalhos relacionados onde é avaliada a eficiência da GPU em termos de desempenho (tempo de execução). Como exemplo representativo, em [Seshadrinathan e Dempski 2008] os autores utilizaram a API HLSL (*High Level Shader Language*), que hoje em dia não é a mais adequada para GPGPU (*General Purpose Computing on Graphics Processing Units*), como foi demonstrado pelos autores de [Manavski 2007]. Já em [Di Biagio et al.2009], os autores apresentaram duas estratégias de paralelização do AES, a *fine-grained*, que atua entre os *rounds* do algoritmo e a *coarse-grained*, que possui o foco em um paralelismo de mais alto nível, exposto pelos modos de operação ECB e CTR.

2. Recriptografia assistida por GPU aplicada a Sistemas VoD

Conforme mencionado anteriormente, a proposta passa pela paralelização do AES-ECB com base no núcleo de criptografia presente na OpenSSL. Essa paralelização foi feita em três níveis (*coarse, fine e mix-grained*), gerando três modelos diferentes: No *coarse-grained*, cada *thread* é responsável por cifrar um frame inteiro para um cliente, cada um deles com uma chave de criptografia diferente; no *fine-grained*, todas as *threads* responsáveis pela criptografia cifram, em cooperação, o mesmo frame para um cliente, esse procedimento é repetido até que todos os clientes tenham o *frame* cifrado; o *mix-grained* une a concorrência da versão *coarse-grained* e a cooperação da versão *fine-grained*, de modo que um grupo de *threads* cifra um *frame* para um cliente em concorrência com outros grupos de *threads*.

Supondo que os clientes requisitem o mesmo vídeo ao mesmo tempo, basta que um *frame* por vez seja enviado para a memória da GPU a fim de codificar o vídeo inteiro para cada um dos clientes, de maneira exclusiva, pois as chaves de criptografia são diferentes. Foram implementadas diferentes versões baseadas nestes três modelos, as quais foram codificadas em C/C++, de modo que todas elas possuem uma *thread* de leitura que produz *frames* e coloca em um *buffer* de entrada, enquanto uma *thread* de

controle consome os *frames* do *buffer* de entrada e passa esse *frame* para as *threads* responsáveis pela criptografia. As versões que fazem uso da GPU utilizam a API CUDA (*Compute Unified Device Architecture*). Por outro lado, as versões que são executadas somente na CPU *multicore* também têm suas *threads* de criptografia criadas através da API Pthreads, competindo pelo reduzido número de núcleos de processamento quando comparado com as GPUs mais recentes.

3. Análise de desempenho

Os experimentos foram realizados em um ambiente de testes composto por uma GPU nVidia GTS250 utilizando o driver (195.36.15), SDK e Toolkit (nas versões 3.0) do CUDA, um processador Intel Core i5 750 (que possui quatro núcleos), memória RAM de 4GB, sistema operacional Ubuntu Linux 9.10 (kernel 2.6.31-14-generic) e compilador GCC 4.3.4.

Com auxílio de uma função do CUDA que extrai o valor de um contador de tempo, foi possível estimar o desempenho em termos de tempo de execução, tanto para versões com GPU quanto para versões "CPU *only"*. Também foi criado um *stream generator* que emula o envio de frames de vídeo de determinado tamanho a uma determinada taxa, para melhor simular um ambiente real de um *proxy* para avaliar e as versões criadas atendiam à demanda da aplicação, ou seja, se conseguiriam cifrar todos os *frames* do vídeo em questão para todos os clientes em paralelo dentro das restrições de tempo de envio características de uma aplicação *soft-realtime*. Por outro lado, o *stream generator* foi um limitante do desempenho, pois quando a leitura do vídeo foi feita diretamente pelo arquivo de entrada o desempenho das melhores versões cresceu, principalmente para a versão *mix-grained* que roda na GPU.

As primeiras experiências realizadas com as versões portadas para CUDA (GPU) seguindo o modelo coarse-grained não atingiram o desempenho demandado pela aplicação, motivando o desenvolvimento das versões baseadas no modelo *fine-grained* e posteriormente no *mix-grained*, sendo este último o que obteve o melhor desempenho em relação a todas as outras versões simuladas, tanto na CPU quanto na GPU, como pode ser verificado nas Figuras 1 e 2. A melhora de desempenho incremental para as versões CUDA se deu, principalmente, pelo fato das *threads* terem ficado mais leves (com carga de trabalho reduzida) e em quantidades maiores. Essa estratégia é a que melhor tira proveito do hardware massivamente paralelo da GPU.

Em particular, os resultados preliminares da avaliação de desempenho com as diferentes versões sugerem que um processador de um único núcleo não daria conta da demanda desta aplicação. Essa afirmação decorre dos resultados da Figura 2 onde, tanto no caso em que o número de clientes supera os 512 quanto onde a taxa de exibição do vídeo é superior a 1Mbps, a criptografia demanda um período de tempo superior ao que seria suportado pela aplicação e, por conseguinte, atrasaria a distribuição de vídeo para todos os clientes sendo atendidos naquele instante.

A partir dos resultados do desempenho em termos do tempo de execução é possível gerarmos os resultados referentes à vazão do sistema.

Nas Figuras 1 e 2 o número após o rótulo representa a quantidade de *threads* de criptografia que são lançadas na versão Pthreads (casos de desempenho mais

representativo), e as linhas tracejadas são uma estimativa da demanda de processamento necessária para atender a um determinado número de clientes.

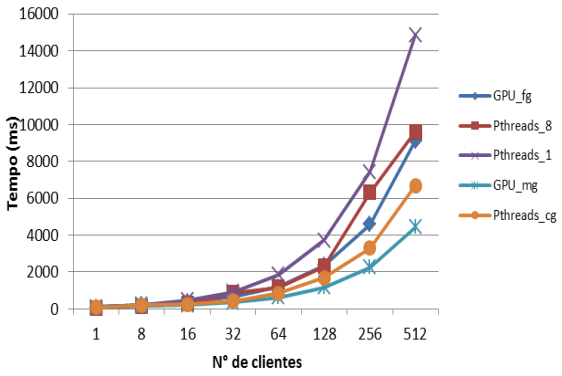

Figura 1. Melhores versões, input 1,9MB

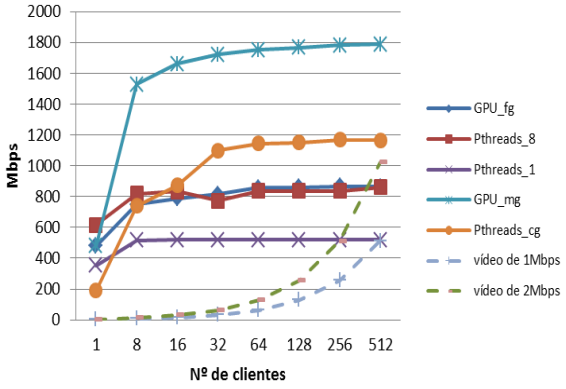

Figura 2. Throughput, input 19MB

6. Conclusões e trabalhos em andamento

Este trabalho realiza a interface entre sistemas de distribuição de vídeo sob demanda e criptografia assistida por GPU, de tal forma que foi feita uma análise qualitativa e quantitativa sobre a viabilidade do uso das GPUs para esta aplicação ao invés de um ou mais processadores *multicore*. Os resultados preliminares obtidos demonstraram que somente um processador não atende a demanda dessa aplicação e que a proposta baseada em GPU tende a ter melhor custo-benefício por fornecer uma vazão significativamente superior às versões que fazem uso somente do processador *multicore*, com a ressalva de que a diferença quantitativa expressa o comportamento para o ambiente experimental adotado, tal que, considerando-se a evolução das GPUs e dos processadores *multicore*, especula-se que a diferença de desempenho se torne ainda maior. Dentro do conceito da computação de alta eficiência, atualmente estão sendo realizados estudos para estimar a eficiência energética desta proposta.

Referências

Accelerators and GPUs track. IEEE SBAC-PAD´09, São Paulo, Brazil, IEEE, Computer Society, 2009.

NVIDIA, http://www.nvidia.com.br/object/what_is_cuda_new_br.html

Seshadrinathan, M., and Dempski, K. L. (2008) "Implementation of Advanced Encryption Standard for Encryption and Decryption of Images and Text on a GPU", IEEE CVPRW´08, Anchorage, AK, USA, 2008.

Manavski, S. A. (2007) "CUDA Compatible GPU as an Efficient Hardware Accelerator for AES Cryptography", IEEE ICSPC´07, Dubai, United Arab Emirates.

Di Biagio, A., Barenghi, A., Agosta G., and Pelosi, G.,(2009) "Design of a Parallel AES for Graphics Hardware using the CUDA framework", IEEE IPDPS´09, Rome, Italy.