Paralelização de Tarefas de Codificação de Vídeo VVC utilizando GPGPUs

Iago Storch
icstorch@inf.ufrgs.br
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brasil

Daniel Palomino
dpalomino@inf.ufpel.edu.br
Universidade Federal de Pelotas (UFPel)
Pelotas, Brasil

Sergio Bampi
bampi@inf.ufrgs.br
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brasil

ABSTRACT

The compression required in video-based applications imposes a significant computational workload. Moreover, computing systems are becoming increasingly heterogeneous, and GPUs have gained popularity. This Ph.D. Thesis proposes a GPU acceleration methodology that provides guidelines for efficiently leveraging heterogeneous CPU+GPU systems to accelerate video coding applications. Experimental results demonstrate that the solutions developed following the proposed methodology accelerate the encoding and improve its energy efficiency while posing minor coding efficiency losses.

KEYWORDS

sistemas heterogêneos, computação em GPU, codificação de vídeo

1 INTRODUÇÃO

Aplicações multimídia baseadas em vídeos são muito populares, representando cerca de 55% do tráfego total de dados da internet [3]. No entanto, estas aplicações requerem que os dados sejam comprimidos seguindo algum padrão de codificação de vídeo, impondo uma carga computacional substancial – a indústria de vídeos digitais é responsável por 1% das emissões globais de gases de efeito estufa [1]. Além disso, ferramentas de codificação de vídeo têm sido usadas como base para muitas aplicações emergentes baseadas em vídeos esféricos, imagens *light field* e nuvens de pontos. Sendo assim, é evidente que o desenvolvimento de algoritmos eficientes e o uso apropriado dos recursos computacionais são de interesse primordial para a evolução e popularização de aplicações multimídia.

Simultaneamente, os sistemas computacionais modernos são cada vez mais heterogêneos, onde a CPU trabalha em conjunto com aceleradores de hardware (geralmente uma GPU) para alcançar um objetivo comum. A evolução das aplicações de aprendizado profundo só foi possível devido à evolução nas capacidades das GPUs e ao desenvolvimento de algoritmos eficientes explorá-las. Isso demonstra que a popularização de algumas aplicações pode exigir a exploração do paradigma de computação em GPUs.

Esta tese de doutorado propõe a primeira metodologia de aceleração por GPU para aplicações de codificação de vídeo, projetada para explorar de forma eficiente os recursos disponíveis em sistemas compostos por CPU+GPU. Além disso, essa metodologia é aplicada a duas ferramentas modernas de codificação de vídeo.

In: VI Concurso de Teses e Dissertações (CTD 2024). Anais Estendidos do XXX Simpósio Brasileiro de Sistemas Multimídia e Web (CTD'2024). Juiz de Fora/MG, Brazil. Porto Alegre: Brazilian Computer Society, 2024.

ISSN 2596-1683

2 METODOLOGIA INTEGRADA DE ACELERAÇÃO COM GPUS PARA APLICAÇÕES DE CODIFICAÇÃO DE VÍDEO (MIAG-CV)

Diversos trabalhos foram propostos para acelerar sistemas de codificação de vídeo utilizando GPUs. No entanto, esses trabalhos discutem suas contribuições em múltiplos níveis simultaneamente, o que cria desafios na compreensão dos algoritmos e de como algumas contribuições poderiam ser reutilizadas entre diferentes ferramentas. Se essas contribuições pudessem ser classificadas de acordo com seus níveis de abstração e como interferem umas nas outras, certamente seria mais fácil entender, comparar e adaptar soluções existentes para diferentes ferramentas.

Esta tese de doutorado propõe preencher essa lacuna definindo a Metodologia Integrada de Aceleração com GPUs para Aplicações de Codificação de Vídeo (MIAG-CV). A MIAG-CV divide as estratégias de aceleração em três níveis de abstração para distinguir entre contribuições para o sistema geral de codificação, contribuições adaptadas para algumas ferramentas de codificação e contribuições relacionadas à implementação. Uma visão geral da MIAG-CV é apresentada na Figura 1. Partindo do nível de abstração mais alto em direção ao mais baixo, as contribuições são divididas em **Gerenciamento de Carga de Trabalho**, **Paralelização de Algoritmo** e **Co-otimização de Algoritmo e Recursos da GPU**.

As contribuições de **Gerenciamento de Carga de Trabalho** se concentram em decisões de alto nível que não interferem diretamente na implementação. Isso inclui gerenciar a codificação de

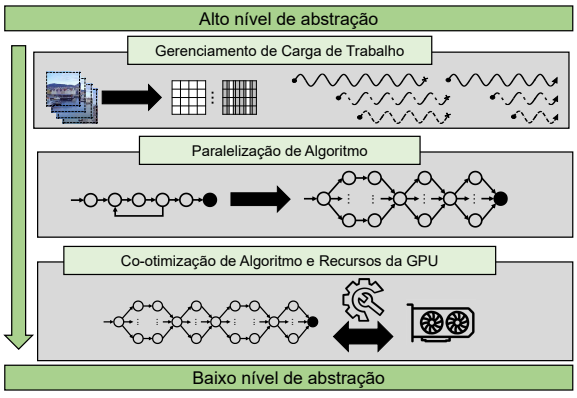

Figura 1: Metodologia Integrada de Aceleração com GPUs para Aplicações de Codificação de Vídeo (MIAG-CV).

blocos com múltiplas dimensões e posições, mapear as tarefas de codificação em kernels para fornecer encapsulamento, e gerenciar a execução e comunicação de kernels para maximizar a utilização dos recursos. Esta tese de doutorado também propõe uma estratégia chamada *unidades de carga de trabalho* para suportar as estruturas de particionamento flexíveis dos padrões de codificação de vídeo modernos. Essa estratégia consiste em enumerar todas as possibilidades de partição permitidas para uma ferramenta de codificação e, em seguida, para cada bloco de 128×128 pixels da imagem, organizar essas partições em uma estrutura semelhante a um mosaico (uma *unidade de carga de trabalho*) que mantém blocos com a mesma dimensão na mesma unidade para manter a regularidade. Cada região 128×128 gera várias unidades de carga de trabalho para cobrir todas as possibilidades de particionamento, e todas as unidades de carga de trabalho são processadas simultaneamente. Essa estratégia maximiza a ocupação enquanto mantém uma estrutura de processamento regular para se adequar à arquitetura das GPUs.

O nível de **Paralelização de Algoritmo** está preocupado em identificar e criar oportunidades de paralelização nos algoritmos de codificação de vídeo. Isso envolve identificar estágios que podem ser paralelizados de maneira direta devido à ausência de dependências de dados e também desenvolver métodos para quebrar dependências de dados não críticas e permitir a execução paralela de algoritmos que, de outra forma, seriam sequenciais — nesse caso, assume-se perdas de eficiência de codificação. Estas contribuições estão focadas na concepção geral do algoritmo, e não em sua implementação.

As contribuições no nível de **Co-otimização de Algoritmo e Recursos da GPU** consistem em desenvolver implementações eficientes dos algoritmos paralelos considerando o hardware das GPUs. Isso inclui gerenciar o layout de dados e a hierarquia de memória explicitamente para coalescer os acessos à memória e maximizar o reuso de dados, usar instruções especializadas da GPU e escalonar as tarefas na GPU para maximizar o throughput. Esta tese de doutorado propõe uma abstração lógica chamada de *conjuntos de trabalho* para auxiliar nesse escalonamento. Cada grupo de threads do lado do software é responsável por uma unidade de carga de trabalho com múltiplos blocos. Os conjuntos de trabalho são usados para subdividir as threads em grupos menores de acordo com a estrutura de particionamento de cada unidade de carga de trabalho, facilitando o mapeamento entre threads e amostras em um bloco.

3 VALIDAÇÃO EXPERIMENTAL

O padrão Versatile Video Coding (VVC) [2] alcança eficiência de codificação estado-da-arte, mas também impõe uma grande carga computacional. Portanto, a MIAG-CV é aplicado a duas ferramentas de codificação introduzidas pelo VVC. Além de demonstrar a aplicação da metodologia, isso também auxilia na redução da carga computacional dos codificadores VVC. As ferramentas abordadas são a Estimação de Movimento Afim (Affine ME) e a Predição Intra Baseada em Matrizes (MIP) [2]. A Affine ME é projetado para explorar padrões de movimento não translacionais. Em contrapartida, a MIP explora métodos baseados em aprendizado de máquina para representar um bloco com base nas amostras vizinhas já codificadas.

Para a **Affine ME**, as contribuições no gerenciamento de carga de trabalho incluem a definição das unidades de carga de trabalho e a exclusão de blocos improváveis. As contribuições na paralelização do algoritmo incluem o cálculo dos vetores de movimento ignorando blocos adjacentes, o cálculo de erro de predição e gradiente das amostras em paralelo, e a construção de sistemas de equações em paralelo. No nível mais baixo, quatro granularidades de dados são definidas para processar diferentes estágios de forma eficiente. A ferramenta **MIP** também utiliza unidades de carga de trabalho no nível de gerenciamento de carga, além de um método baseado em filtros passa-baixa para reproduzir o efeito da quantização e quebrar as dependências entre blocos adjacentes. As contribuições na paralelização do algoritmo incluem o cálculo do erro de predição em paralelo. No nível de implementação, operações especializadas de produto escalar são usadas para gerar a predição. A abstração definida pela MIAG-CV permite que algumas contribuições sejam reutilizadas entre duas ferramentas completamente diferentes.

A validação experimental comparou o tempo de processamento, o consumo energético e a eficiência de codificação dos kernels de codificação implementados em GPU com o codificador de referência rodando em CPU. O codificador de referência do VVC, chamado VTM, foi executado em um processador Intel Core i9 7900X com 64GB de RAM. Os kernels de GPU foram implementados em OpenCL e executados em três dispositivos: NVIDIA GTX 1080, NVIDIA Titan V e Radeon RX 6900XT. Os utilitários RAPL, `nvidia-smi` e `rocm-smi` foram utilizados para monitorar o consumo de energia.

Os experimentos com Affine ME mostraram que a codificação baseada em GPU foi de 19 a 267 vezes mais rápida que o codificador de referência, enquanto consumiu entre 2,10% e 22,42% da energia consumida pela CPU. Os resultados variam com base no conteúdo do vídeo e no dispositivo GPU, mas a solução proposta é tanto mais rápida quanto mais eficiente em termos de energia em todos os casos. As modificações do algoritmo para expor o paralelismo requerem um bitrate adicional de 0,017% $\sim$ 1,784% para alcançar a mesma qualidade visual, de acordo com a métrica BD-BR.

Para a MIP, os kernels de GPU aceleram a codificação entre 10 e 136 vezes, enquanto consomem entre 0,75% e 21,45% da energia consumida pela CPU. Novamente, a solução baseada em GPU é mais rápida e mais eficiente em termos de energia. Por fim, a solução com GPUs exige um bitrate adicional entre 0,105% e 0,736% para alcançar a qualidade visual do codificador VTM.

4 CONCLUSÃO

Esta tese de doutorado propôs a MIAG-CV, uma metodologia hierárquica para desenvolver soluções de aceleração por GPU em aplicações de codificação de vídeo. A abstração proporcionada pela MIAG-CV permitiu que procedimentos de aceleração semelhantes fossem aplicados em ferramentas de codificação distintas. Resultados experimentais mostraram que os kernels de computação em GPU, desenvolvidos com a MIAG-CV, são centenas de vezes mais rápidos que o codificador em CPU, consumindo menos de 22% da energia utilizada pela CPU. Os impactos na eficiência de codificação foram praticamente desprezíveis.

REFERÊNCIAS

[1] Bitmovin. 2023. *Bitmovin and GAIA Project: Streaming Sustainability Progress Report.* Technical Report. Bitmovin.
[2] Benjamin Bross et. al. 2021. Overview of the Versatile Video Coding (VVC) Standard and its Applications. *IEEE Trans. on Circ. and Syst. for Video Technol.* 31, 10 (2021), 3736–3764.
[3] Sandvine. 2024. *The Global Internet Phenomena Report.* Technical Report. Sandvine.