

Análise Comparativa do Desempenho de Unidades de Processamento Gráfico Nvidia e ATI através de filtros digitais de imagens

Darlisson Marinho de Jesus¹
Raimundo Corrêa de Oliveira¹

¹Engenharia de Computação
Universidade do Estado do Amazonas - UEA

Novembro - 2013

Sumário

- Introdução
 - Descrição do Problema
 - Justificativa
 - Objetivo Geral
 - Objetivos Específicos
- Metodologia
- Desenvolvimento
 - Arquitetura da GPU Moderna
 - A Linguagem OpenCL
 - Filtro Sobel
 - Filtro Passa-baixa
- Resultados
- Referências

Descrição do Problema

- As áreas da computação de alto desempenho estão adotando modernas unidades de processamento gráfico para resolver problemas de cálculo em grande escala, como comparação de imagens e modelagem climática [4].
- Diante disso, como auxiliar engenheiros e cientistas a escolherem as GPUs que forneçam o melhor desempenho aliado ao menor custo das tecnologias com o passar do tempo?

Descrição do Problema

- Aceleradores/Co-processadores dos Supercomputadores - Top 500

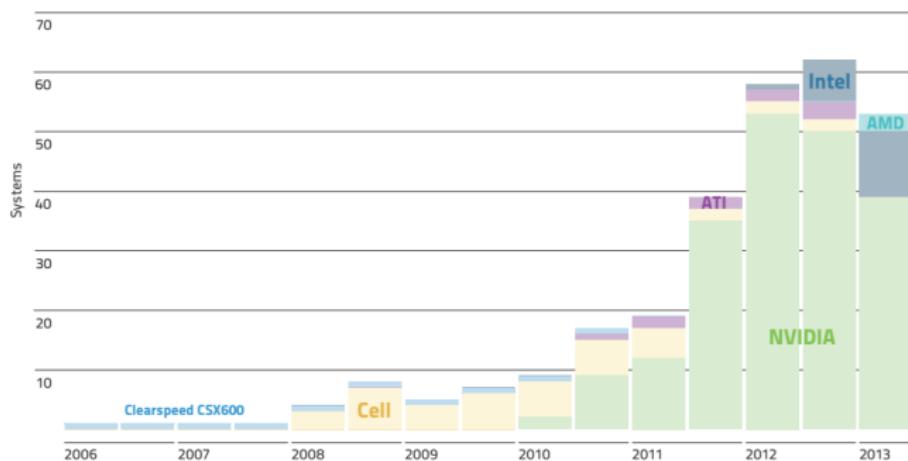


Figura: Fabricante dos co-processadores de Supercomputadores (Fonte: [3])

Descrição do Problema

- É necessário avaliar quais GPUs fornecem o melhor desempenho para soluções de computação paralela, e assim, auxiliar no projeto das soluções para a computação de alta performance.

Justificativa

- Não foi encontrado registros de trabalhos que fazem uma análise comparativa do desempenho de GPU de diferentes fabricantes usando filtros digitais escritos com a linguagem OpenCL.
- Este trabalho apresenta o desempenho de GPUs com preços acessíveis.

Justificativa

- GPU Nvidia Tesla k20 a mais utilizada na computação de alto desempenho.

Placa de Vídeo NVIDIA TESLA K20 5GB GDDR5 320Bits 2496 Cuda Cores



Figura: Preço em reais da GPU Nvidia Tesla k20

Objetivo Geral

- Comparar o desempenho das Unidades de Processamento Gráfico das fabricantes NVIDIA e ATI, através do Processamento Digital de Imagens com os filtros Passa-baixa e o filtro Sobel para detecção de borda implementados na linguagem OpenCL. E também, avaliar o quanto o desempenho das GPUs desses fabricantes é superior ao desempenho das CPUs da fabricante Intel.

Objetivos Específicos

- Determinar os indicadores de desempenho que permitam avaliar as rotinas destes filtros no contexto das GPUs e CPUs;
- Avaliar a arquitetura das GPUs da Nvidia e ATI, buscando identificar as diferenças que podem afetar no desempenho das implementações;
- Implementar o algoritmo do filtro Passa-Baixa na linguagem OpenCL e obter os dados de desempenho das GPUs Nvidia e ATI;

Objetivos Específicos

- Determinar os indicadores de desempenho que permitam avaliar as rotinas destes filtros no contexto das GPUs e CPUs;
- Avaliar a arquitetura das GPUs da Nvidia e ATI, buscando identificar as diferenças que podem afetar no desempenho das implementações;
- Implementar o algoritmo do filtro Passa-Baixa na linguagem OpenCL e obter os dados de desempenho das GPUs Nvidia e ATI;

Objetivos Específicos

- Determinar os indicadores de desempenho que permitam avaliar as rotinas destes filtros no contexto das GPUs e CPUs;
- Avaliar a arquitetura das GPUs da Nvidia e ATI, buscando identificar as diferenças que podem afetar no desempenho das implementações;
- Implementar o algoritmo do filtro Passa-Baixa na linguagem OpenCL e obter os dados de desempenho das GPUs Nvidia e ATI;

Objetivos Específicos

- Determinar os indicadores de desempenho que permitam avaliar as rotinas destes filtros no contexto das GPUs e CPUs;
- Avaliar a arquitetura das GPUs da Nvidia e ATI, buscando identificar as diferenças que podem afetar no desempenho das implementações;
- Implementar o algoritmo do filtro Passa-Baixa na linguagem OpenCL e obter os dados de desempenho das GPUs Nvidia e ATI;

Objetivos Específicos

- Implementar o algoritmo do filtro Passa-Baixa na linguagem C e obter os dados de desempenho das CPUs Intel;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem OpenCL e obter dados de desempenho das GPU Nvidia e ATI;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem C e obter dados de desempenho das CPUs Intel;
- Comparar o desempenho das GPUs dos fabricantes Nvidia e ATI
- Comparar o desempenho das GPUs Nvidia e ATI com a CPU Intel

Objetivos Específicos

- Implementar o algoritmo do filtro Passa-Baixa na linguagem C e obter os dados de desempenho das CPUs Intel;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem OpenCL e obter dados de desempenho das GPU Nvidia e ATI;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem C e obter dados de desempenho das CPUs Intel;
- Comparar o desempenho das GPUs dos fabricantes Nvidia e ATI
- Comparar o desempenho das GPUs Nvidia e ATI com a CPU Intel

Objetivos Específicos

- Implementar o algoritmo do filtro Passa-Baixa na linguagem C e obter os dados de desempenho das CPUs Intel;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem OpenCL e obter dados de desempenho das GPU Nvidia e ATI;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem C e obter dados de desempenho das CPUs Intel;
- Comparar o desempenho das GPUs dos fabricantes Nvidia e ATI
- Comparar o desempenho das GPUs Nvidia e ATI com a CPU Intel

Objetivos Específicos

- Implementar o algoritmo do filtro Passa-Baixa na linguagem C e obter os dados de desempenho das CPUs Intel;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem OpenCL e obter dados de desempenho das GPU Nvidia e ATI;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem C e obter dados de desempenho das CPUs Intel;
- Comparar o desempenho das GPUs dos fabricantes Nvidia e ATI
- Comparar o desempenho das GPUs Nvidia e ATI com a CPU Intel

Objetivos Específicos

- Implementar o algoritmo do filtro Passa-Baixa na linguagem C e obter os dados de desempenho das CPUs Intel;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem OpenCL e obter dados de desempenho das GPU Nvidia e ATI;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem C e obter dados de desempenho das CPUs Intel;
- Comparar o desempenho das GPUs dos fabricantes Nvidia e ATI
- Comparar o desempenho das GPUs Nvidia e ATI com a CPU Intel

Objetivos Específicos

- Implementar o algoritmo do filtro Passa-Baixa na linguagem C e obter os dados de desempenho das CPUs Intel;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem OpenCL e obter dados de desempenho das GPU Nvidia e ATI;
- Implementar o algoritmo do filtro para detecção de borda Sobel na linguagem C e obter dados de desempenho das CPUs Intel;
- Comparar o desempenho das GPUs dos fabricantes Nvidia e ATI
- Comparar o desempenho das GPUs Nvidia e ATI com a CPU Intel

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos

- Revisão bibliográfica;
- Análise e experimentação da linguagem OpenCL;
- Desenvolvimento dos Filtros Digitais (Sobel e Passa-baixa);
- Coleta das imagens para serem processadas;
- Coleta e análise dos dados;

Métodos - medidas de desempenho

- **Tempo Médio de Execução Total** - corresponde ao tempo total da execução do programa, medida usada para GPU e CPU.
- **Tempo Médio de Execução do Kernel** - corresponde ao tempo de execução da rotina específica do filtro digital, medida usada para GPU e CPU.
- **Taxa Média de Transferência Dados da Memória** - corresponde ao número de bytes por unidade de tempo transmitidos entre a memória do computador e a memória da placa gráfica, escrita e leitura, mediada usada somente para GPU.

Materiais

Equipamentos:

- Três PC/x64 com Sistema Operacional Microsoft Windows 8 Professional 64 Bits, Processador *Intel(R) Core™ 2 Duo E7400* 2.80GHz e 4 GB de memória RAM.
- Três Unidades de processamento gráfico. A Tabela 1 apresenta as especificações de hardware detalhados dessas GPUs.

Materiais - Ambiente de teste

- Diagrama do ambiente de teste.



Figura: Diagrama do ambiente de teste

Materiais Cont.

Modelo	Geforce GT 520	Geforce 210	Radeon HD 6450
Processadores de Stream	48	16	160
Clock do processador	810 MHz	589 MHz	750 MHz
Arquitetura da GPU	Tesla	Fermi	Caicos
Memória	—	—	—
Clock da memória	900 MHz	533 MHz	1066 MHz
Tamanho da memória	1024 MB	512 MB	1024 MB
Interface da memória	64-bit	64-bit	64-bit
Largura de Banda (GB/sec)	14.4	8.0	8.5
Tipo de memória	DDR3	DDR3	DDR3

Tabela: Resumo das especificações das GPUs Nvidia e ATI

Materiais - Desenvolvimento.

As seguintes ferramentas foram utilizadas para compilação dos programas:

- Microsoft Visual Studio 2010 versão 10.0.30319.1 RTMRel
- Microsoft .NET Framework versão 4.5.50709 RTMRel
- OpenCL-GPU: Pacote AMD APP SDK versão 2.8.1 e driver de vídeo ATI Catalyst versão 12.104 no Windows 8 64 bits.
- OpenCL-GPU: Pacote Nvidia Cuda Toolkit versão 5.0 e driver de vídeo versão 320.18 no windows 8 64 bits.

Materiais - Amostras.

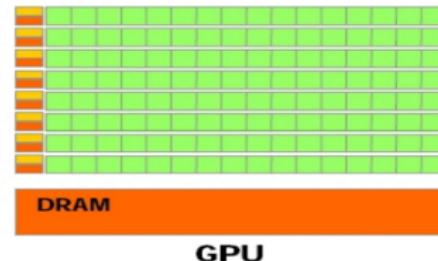
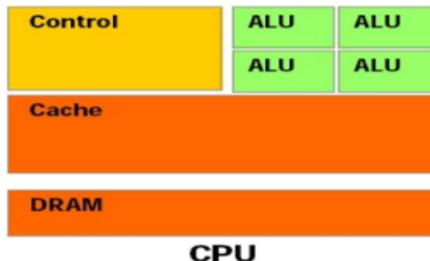
- Usamos 48 imagens no formato *PGM* (do inglês, *Portable Gray Map*), divididas em 8 amostras para cada uma das seguintes dimensões (em pixels): 256x256, 512x512, 1024x1024, 2048x2048, 4096x4096 e 8192x8192.
- A resolução da maior imagem utilizada é de aproximadamente 67 MegaPixels.

Materiais - Análise e Coleta dos Dados

A seguinte ferramenta foi utilizada para geração dos gráficos:

- Software R versão 3.0.1

CPU X GPU



- Própria para tarefas sequenciais
- Cache eficiente
- Maior quantidade de memória principal
- Número de cores de 1 ordem de grandeza
- 1, 2 threads por core

- Própria para tarefas com paralelismo de dados
- Maior (capacidade) operações de ponto flutuante por segundo
- Alto throughput de memória
- Dezenas de multiprocessors
- Múltiplas threads por core

Visão geral da arquitetura Nvidia Tesla

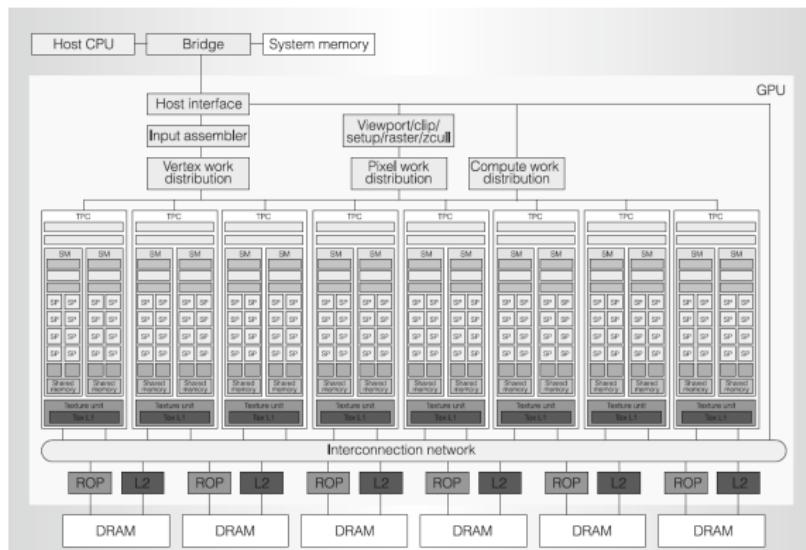


Figura: Visão Geral da arquitetura Nvidia Tesla. Fonte:([1])

Detalhes da arquitetura Nvidia Tesla

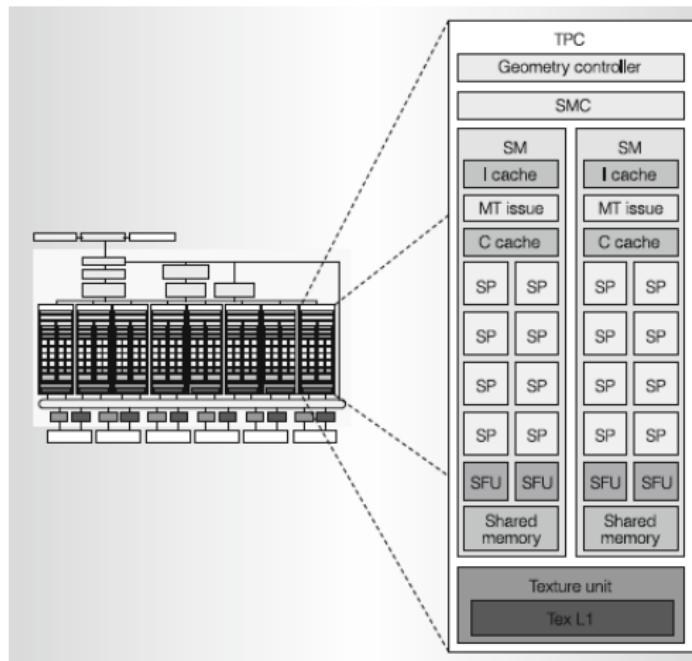


Figura: Detalhes da arquitetura Nvidia Tesla. Fonte: ([1])

Visão geral da arquitetura Nvidia Fermi

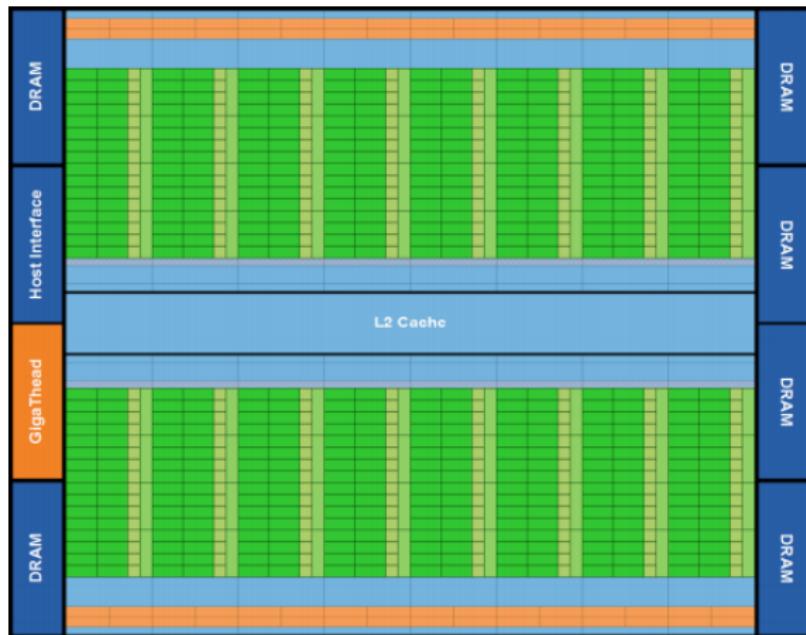


Figura: Visão geral da arquitetura Nvidia Fermi

Detalhes da arquitetura Nvidia Fermi

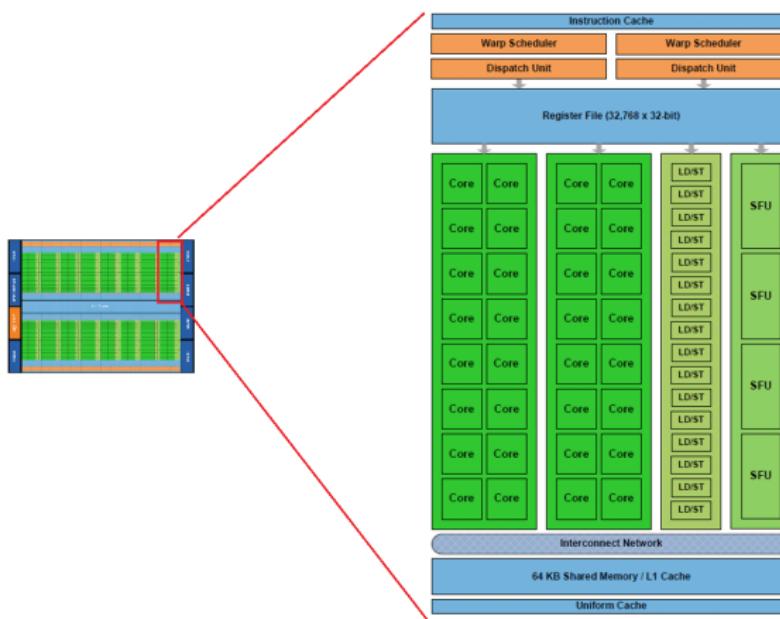


Figura: Detalhes da arquitetura Nvidia Fermi

Diferenças Fermi x Tesla

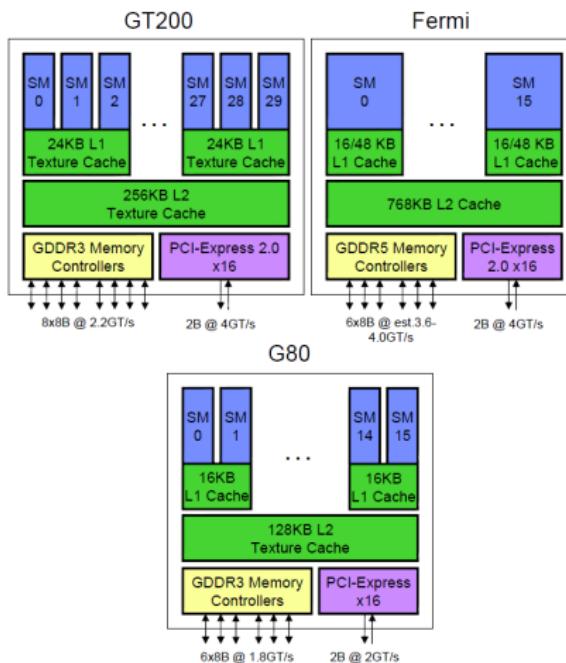


Figura: Diferenças entre a Fermi e a Tesla

Visão geral da arquitetura ATI Caicos

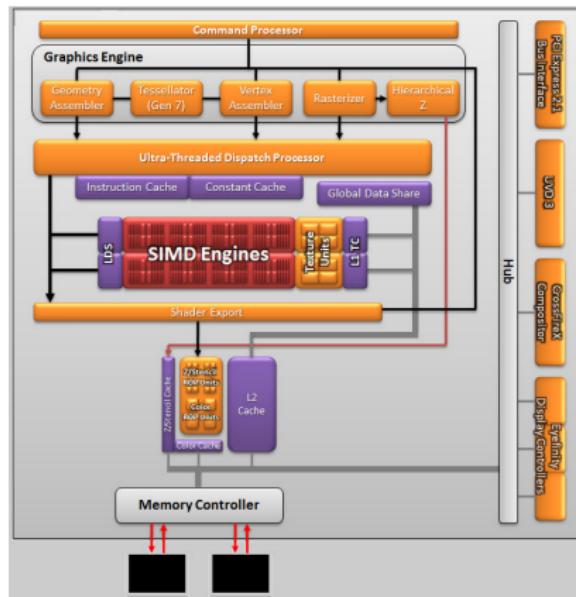
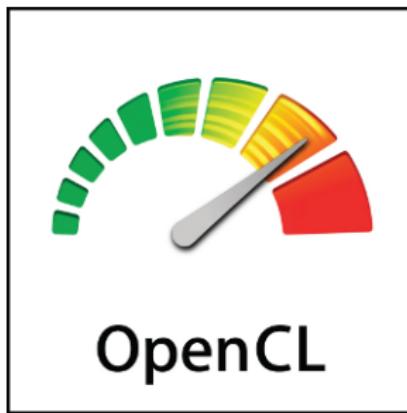


Figura: Visão geral da arquitetura ATI Caicos

GPGPU

- **General-purpose computing on Graphics Processing Units**
 - Técnica de uso de GPU para computação de propósito geral
- Linguagens/API's
 - Brook
 - Brook+
 - OpenCL
 - CUDA (Linguagem proprietária da Nvidia, a mais utilizada atualmente em programação de GPUs)

OpenCL - Open Computing Language



**"Padrão aberto para a
programação paralela de sistemas
heterogêneos"**

OpenCL - Open Computing Language

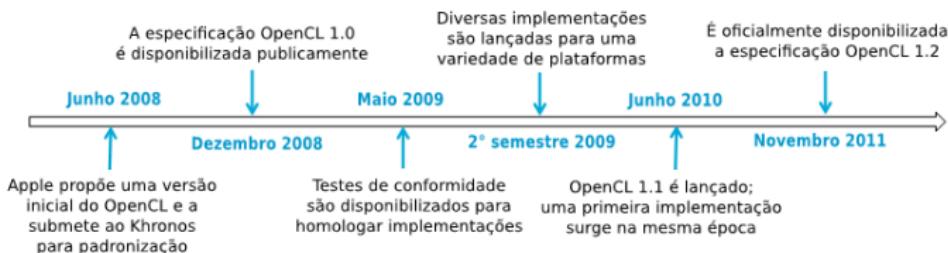
Principais características:

- Provê interface homogênea para a exploração da computação paralela heterogênea
 - Abstração do hardware
 - CPU's (AMD, ARM, IBM, Intel), GPU's (AMD, ARM, Intel, Nvidia), APU's, CBE, DSP's, FPGA's
- Padrão aberto
 - Especificação mantida por vários membros gerenciada pelo grupo Khronos
- Multi-plataforma
 - Disponível em várias classes de hardware e sistemas operacionais
 - Código portável entre arquiteturas e gerações
- Especificação baseada nas linguagens C e C++

OpenCL - História

História

- **2003:** GPUs começam a adquirir características de propósito geral: a era da programabilidade
- **2003-2008:** Cenário GP-GPU fragmentado, com várias soluções proprietárias
- **2008:** Apple elabora o rascunho inicial da especificação do OpenCL



- **2013:** OpenCL 2.0 é lançado

OpenCL - Contribuidores



Figura: Contribuidores para o OpenCL em 2013

Filtro Sobel

O filtro Sobel calcula o gradiente da intensidade da imagem em cada ponto, dando a direcção da maior variação de claro para escuro e a quantidade de variação nessa direcção, através de duas matrizes 3x3, que são convoluídas com a imagem original para calcular aproximações das derivadas - uma para as variações horizontais G_x e uma para as verticais G_y .

Máscara de Sobel 3x3

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

A magnitude do gradiente é dado por:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Filtro Passa Baixa no domínio da Frequência

Filtro passa-baixas ideal - Um filtro passa-baixas 2-D ideal é aquele cuja função de transferência satisfaz a relação:

$$H(u, v) = \begin{cases} 1, & \text{se } D(u, v) \leq Do, \\ 0, & \text{se } D(u, v) > Do. \end{cases}$$

onde Do é um valor não-negativo (corresponde à frequência de corte de um filtro 1-D), e $D(u, v)$ é a distância do ponto (u, v) à origem do plano de frequência; isto é,

$$D(u, v) = \sqrt{(u - P/2)^2 + (v - Q/2)^2}$$

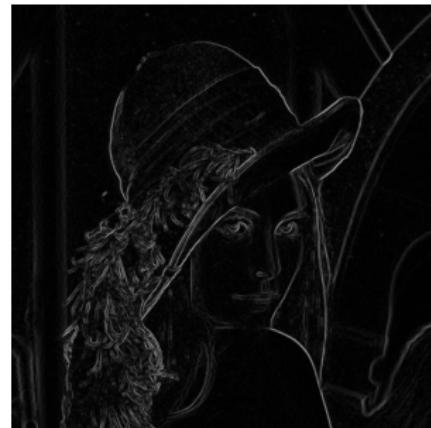
onde, P e Q são, respectivamente, a largura e a altura da imagem.

Quanto menor o raio Do , menor a frequência de corte e, portanto, maior o grau de borramento da imagem resultante.

Filtro Sobel



(a) Imagem original

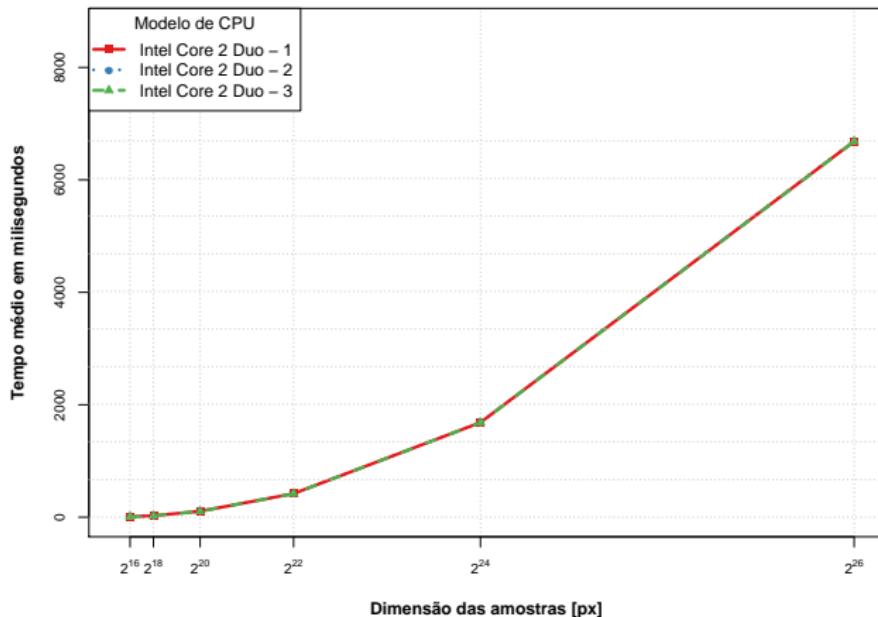


(b) Imagem após a aplicação do filtro Sobel

Figura: Resultado da aplicação filtro Sobel para detecção de borda

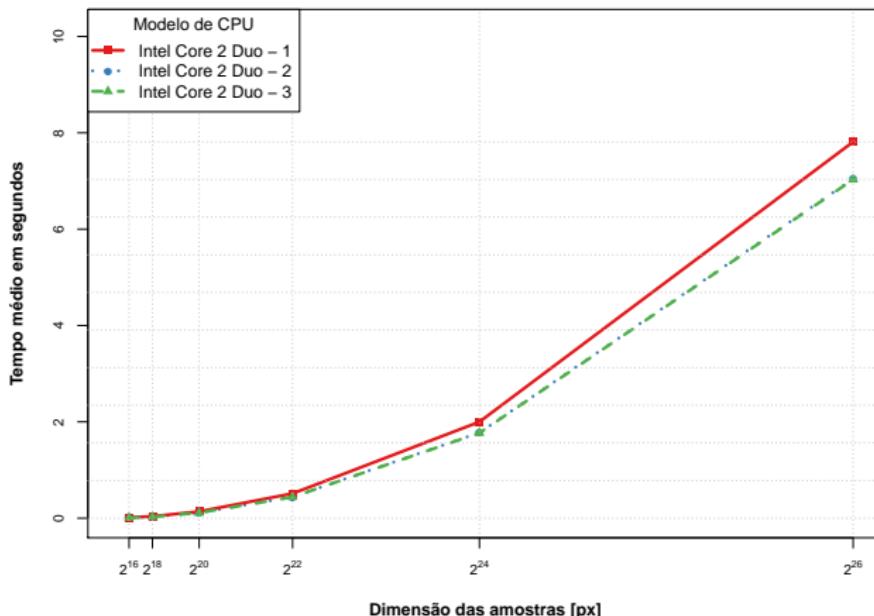
Filtro Sobel CPU - Tempo médio de execução do filtro

CPU – Sobel – Tempo Médio de execução do kernel



Filtro Sobel CPU - Tempo médio de execução total

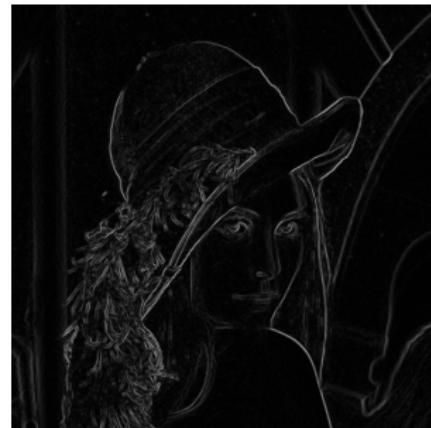
CPU – Sobel – Tempo Médio de execução total



Filtro Sobel



(a) Imagem original

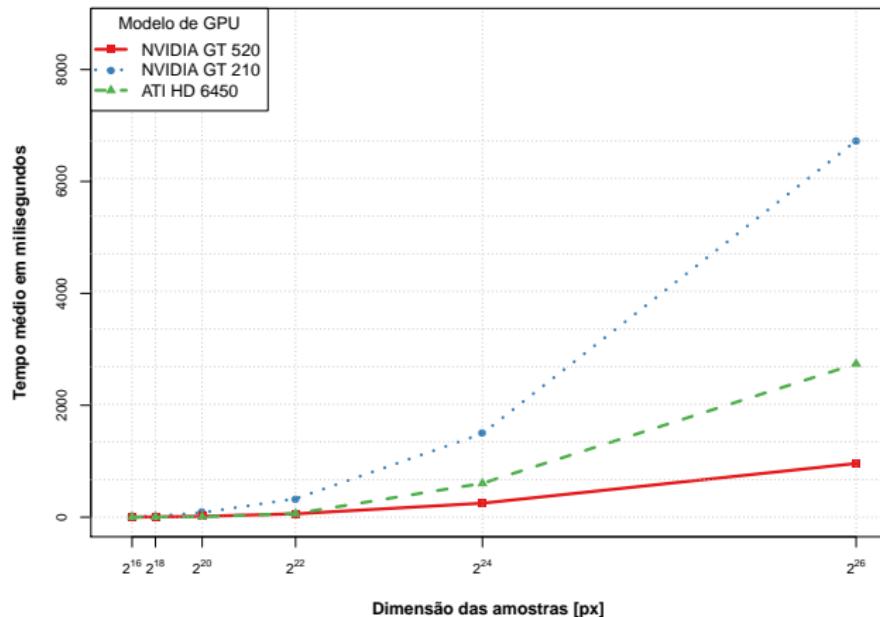


(b) Imagem após a aplicação do filtro Sobel

Figura: Resultado da aplicação filtro Sobel para detecção de borda

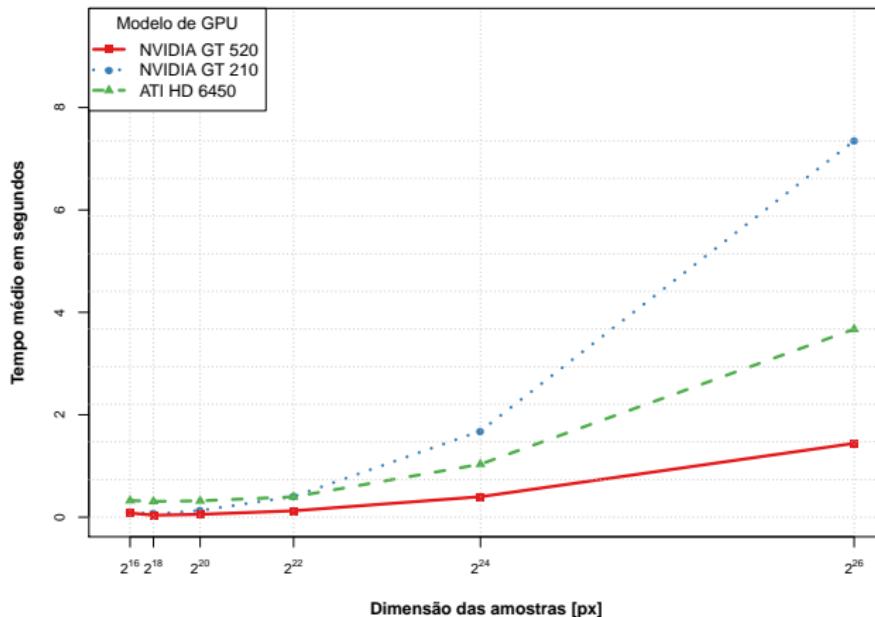
Filtro Sobel GPU - Tempo médio de execução do filtro

GPU – Sobel – Tempo Médio de execução do kernel



Filtro Sobel GPU - Tempo médio de execução total

GPU – Sobel – Tempo Médio de execução total



Filtro Sobel - Taxa média de transferência

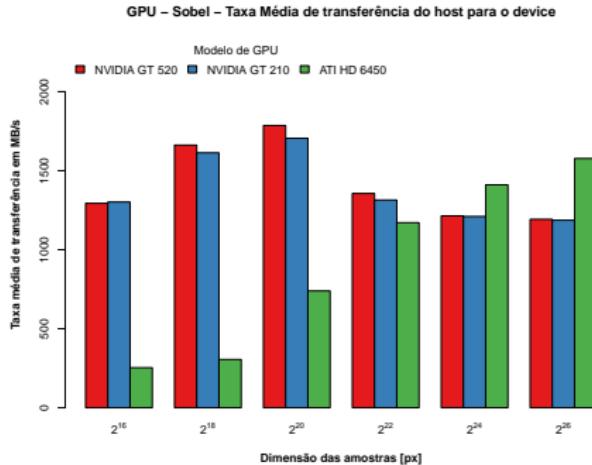


Figura: Taxa média de transferência do Host para o device

Filtro Sobel - Taxa média de transferência

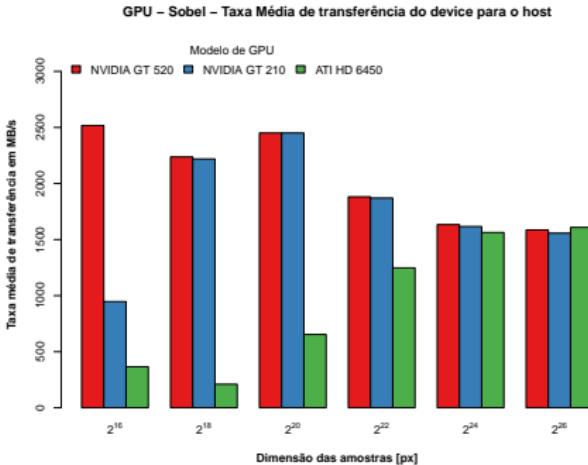


Figura: Taxa média de transferência do Device para o host

Filtro Passa-baixa



(a) Imagem original

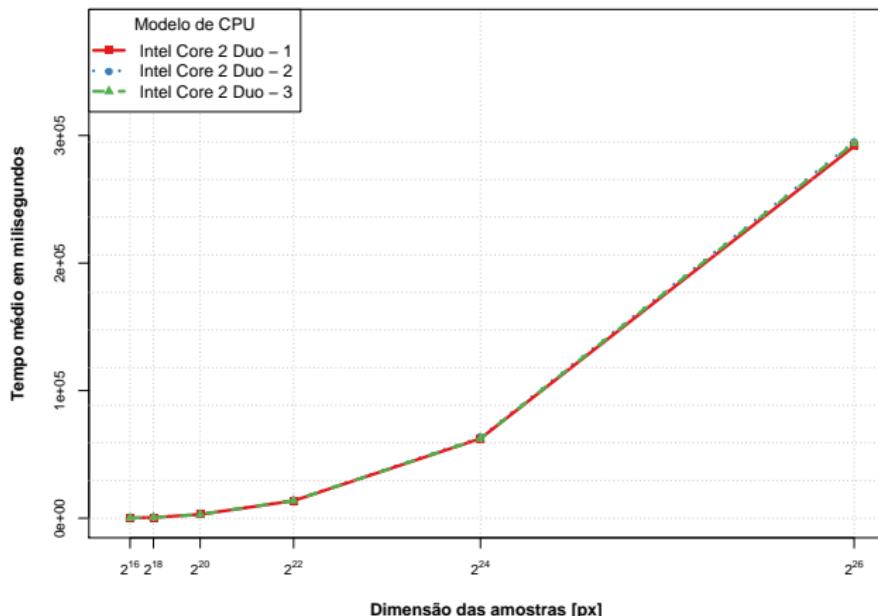


(b) Imagem após a aplicação do filtro Passa-baixa

Figura: Resultado da aplicação do filtro Passa-baixa

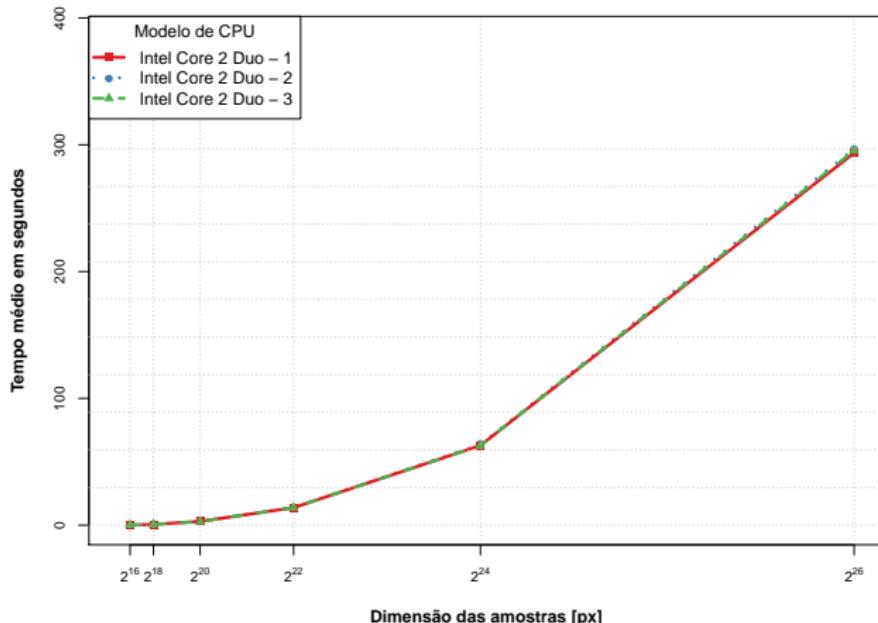
Filtro Passa-baixa- Tempo médio de execução do filtro

CPU – FFT – Tempo Médio de execução do kernel



Filtro Passa-baixa- Tempo médio de execução total

CPU – FFT – Tempo Médio de execução total



Filtro Passa-baixa



(a) Imagem original

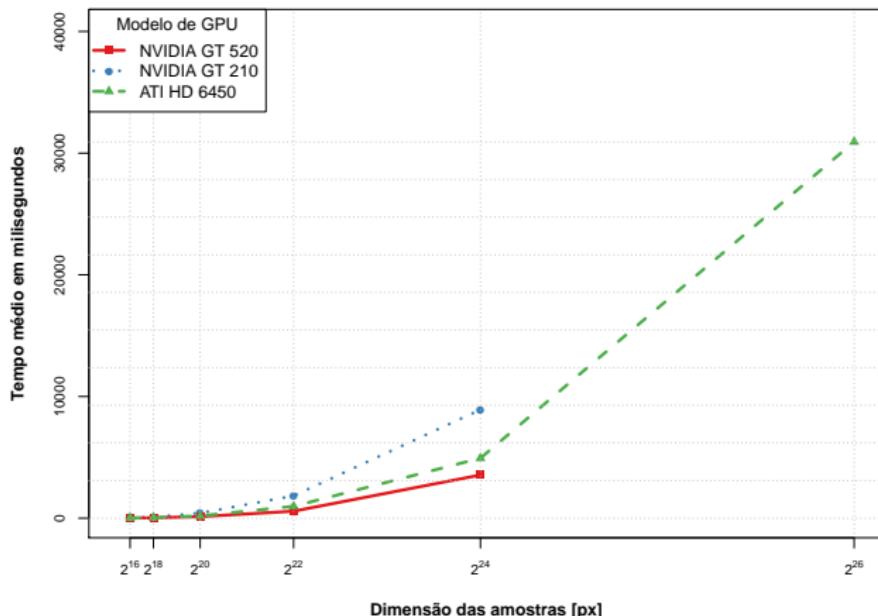


(b) Imagem após a aplicação do filtro Passa-baixa

Figura: Resultado da aplicação do filtro Passa-baixa

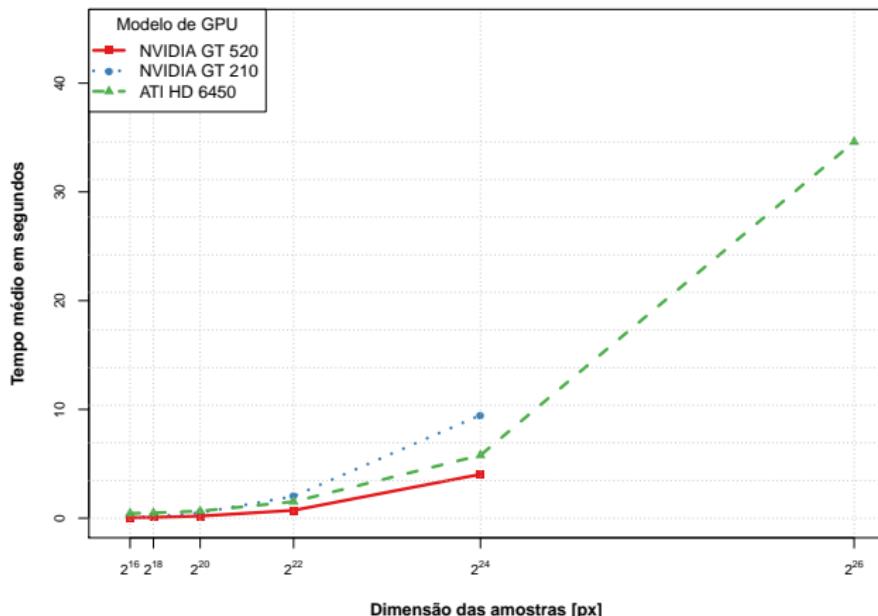
Filtro Passa-baixa- Tempo médio de execução do filtro

GPU – FFT – Tempo Médio de execução do kernel



Filtro Passa-baixa- Tempo médio de execução total

GPU – FFT – Tempo Médio de execução total



Filtro Passa-baixa - Taxa média de transferência

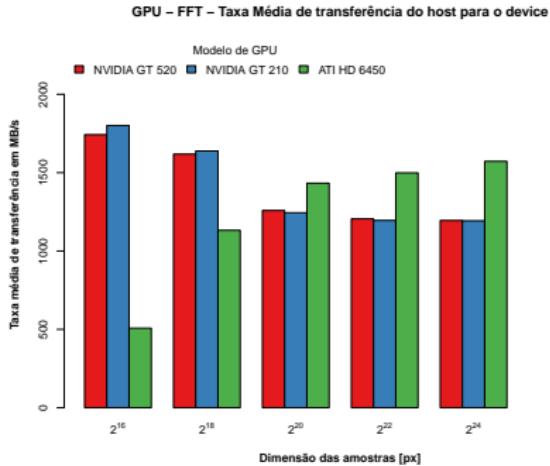


Figura: Taxa média de transferência do Host para o device

Filtro Passa-baixa - Taxa média de transferência

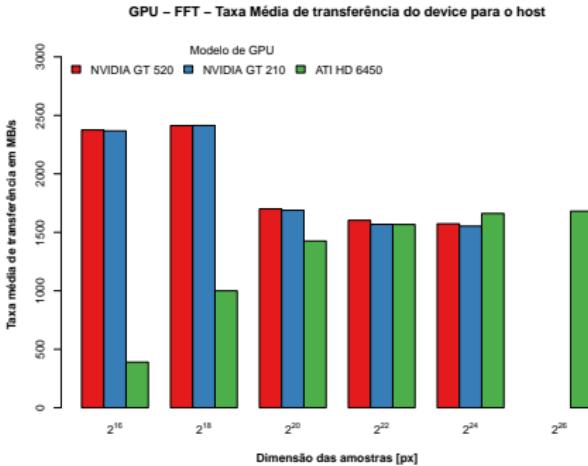


Figura: Taxa média de transferência do Device para o host

Conclusões

- As GPUs tiveram um desempenho superior ao desempenho da CPU.
- A GPU Nvidia Geforce GT 520 possui uma arquitetura que favorece as aplicações que exigem altas taxas de transferências.
- A GPU ATI Radeon HD 6450 possui uma arquitetura que favorece as aplicações em que o processamento dos dados é mais importante.

Referências

-  E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym.
Nvidia tesla: A unified graphics and computing architecture.
Micro, IEEE, 28(2):39–55, 2008.
-  D. Martin, C. Fowlkes, D. Tal, and J. Malik.
A database of human segmented natural images and its application
to evaluating segmentation algorithms and measuring ecological
statistics.
In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423,
July 2001.
-  Top500.
Top 500 supercomputer sites.
<http://www.top500.org/>, June 2013.
-  Ying Zhang, Lu Peng, Bin Li, Jih-Kwon Peir, and Jianmin Chen.
Architecture comparisons between nvidia and ati gpus: Computation
parallelism and data communications