

Tema: Processamento de Alto Desempenho com PDI

PROCESSAMENTO DE ALTO DESEMPENHO APLICADO AO PROCESSAMENTO DE IMAGENS DE SATÉLITES COM AS TECNOLOGIAS OPENMP, CUDA E OPENCL

CAP-378 Tópicos em Observação da Terra

Professores:

**Antônio Miguel Vieira Monteiro
Sérgio Rosim**

Grupo 3:

**Marcos Lima Rodrigues
Marilyn Menecucci Ibañez
Rodrigo Augusto Rebouças
Viny Cesar Pereira**



PÓS-GRADUAÇÃO

COMPUTAÇÃO APLICADA

Roteiro

- **Objetivo**
- **Introdução**
 - **Processamento de Alto Desempenho**
 - **Processamento Digital de Imagens**
- **Resultados**
 - **Sequencial, OpenMP**
 - **OpenCL, CUDA**
- **Trabalhos Relacionados**
 - **FPGA**
- **Conclusão**
- **Referências**

Objetivo

Apresentar as técnicas de otimização com hardware e software para a programação em Processamento de Alto Desempenho utilizando metodologias de Processamento Digital de Imagens.

Introdução

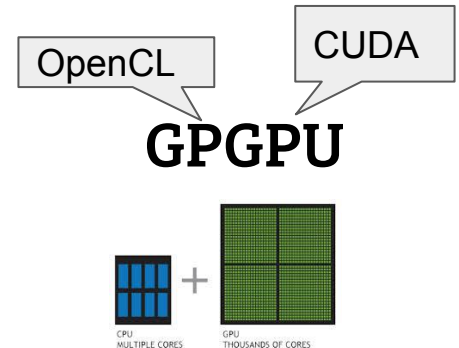
C Otimizado



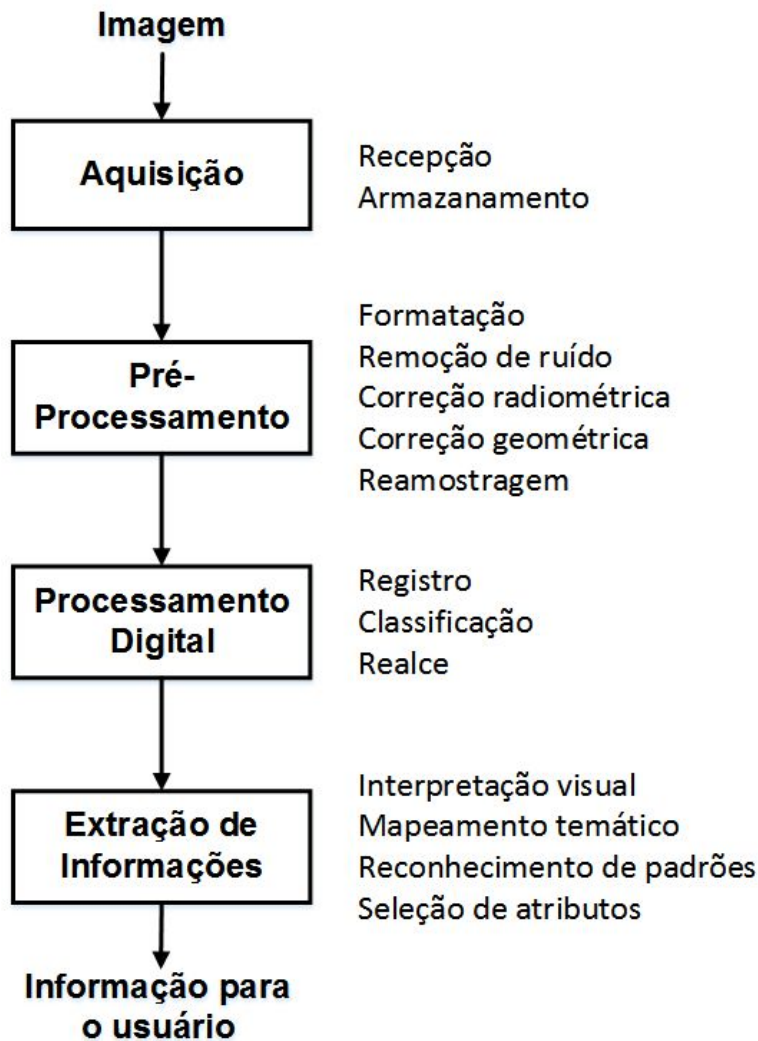
GCC O1, O2, O3

Processamento de Alto Desempenho - PAD

OpenMP



Introdução



Processamento Digital de Imagens - PDI

Realce de Imagem

Operador Sobel

Operador Canny

OpenCV



Dados Utilizados

- **Landsat 8 - Vale do Paraíba**
 - Resolução: 7621x7731 px
- **Composição das Bandas**
 - 4 (Vermelho)
 - 3 (Verde)
 - 2 (Azul)
- **Recorte**
 - SJC
 - 1177x1117 px



Fonte: <http://www.dgi.inpe.br/catalogo/>



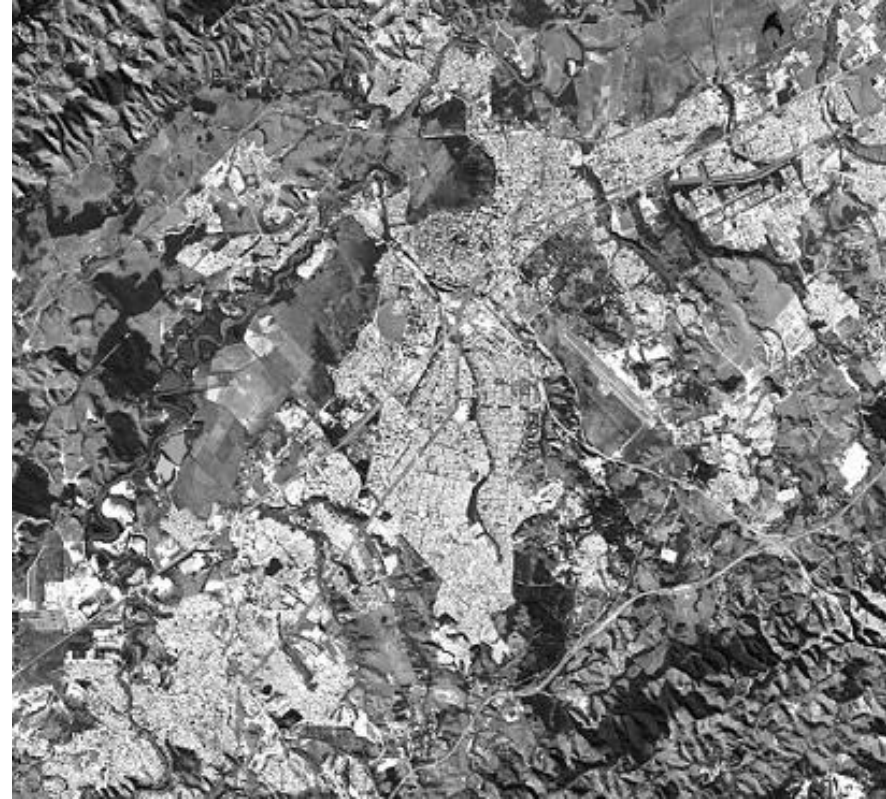
Dados Utilizados

Fonte: <http://www.dgi.inpe.br/catalogo/>

- **Landsat 8 - SJC**
 - **Resolução: 1449x1328 px**
- **Banda 8 Pancromática**
 - **Recorte**



Aeroporto
400x400 px



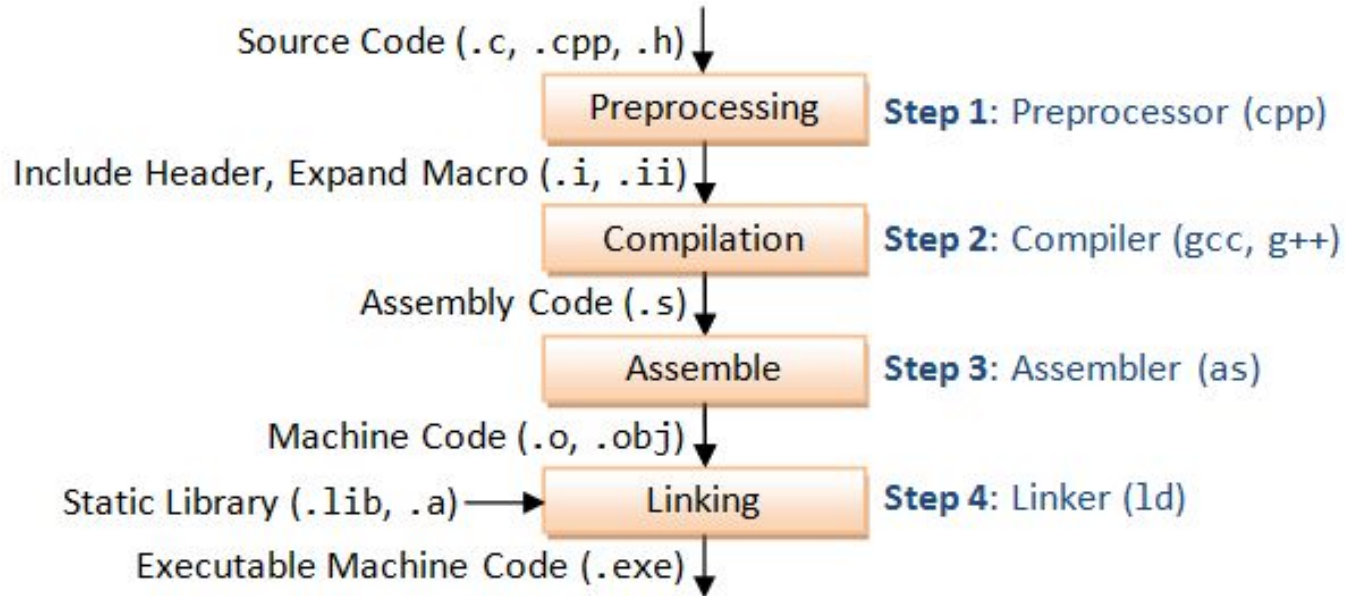


GNU Compiler Collection

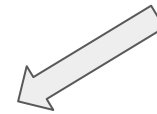
- **Front end para várias linguagens**
- **Originalmente desenvolvido para aplicação no sistema GNU**
- **Níveis de Otimização:
00, 01, 02, 03**



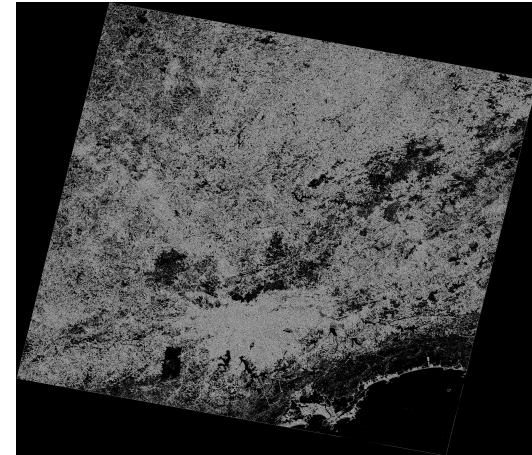
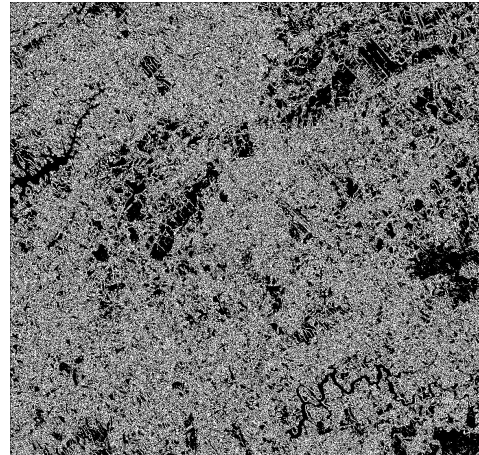
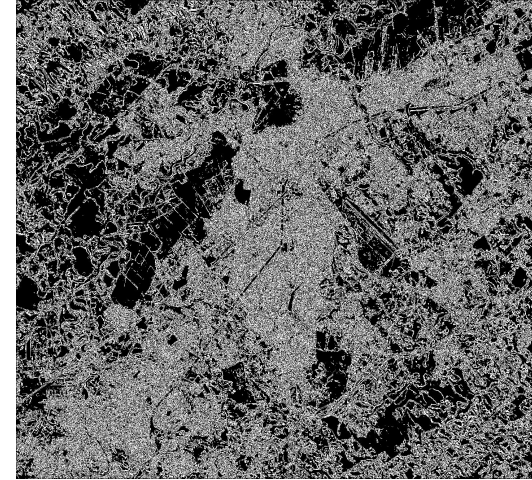
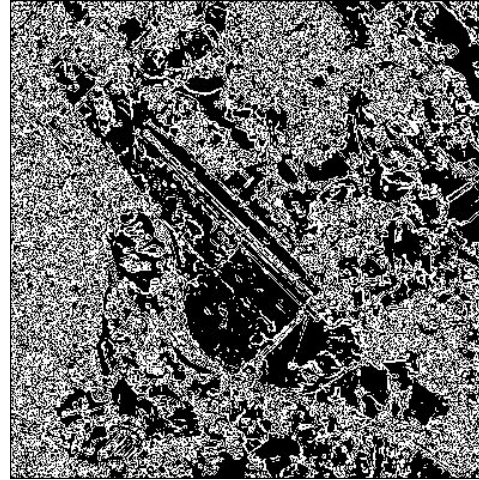
GNU Compiler Collection



Otimização

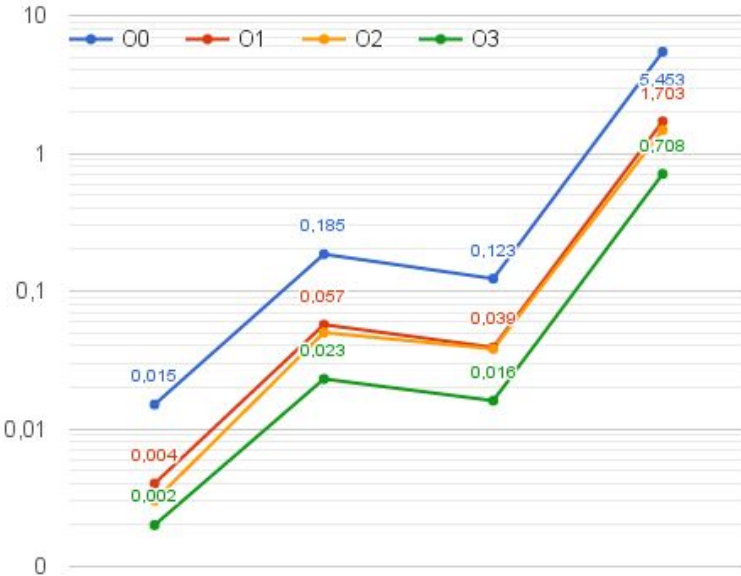


- Intel Core i7 3.60GHz
- Memória RAM: 8GB
- GCC 5.4.0



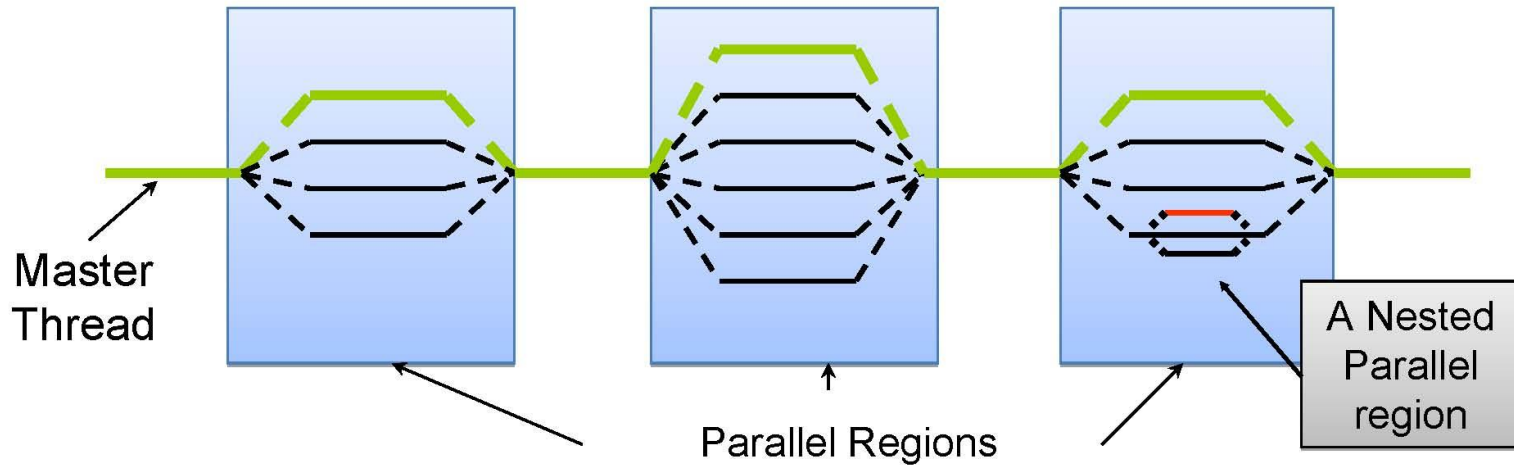
Filtro Sobel - GCC

Imagens / Tempo (s)



OpenMP

- API para programação paralela em C, C++ e Fortran em memória compartilhada



OpenMP

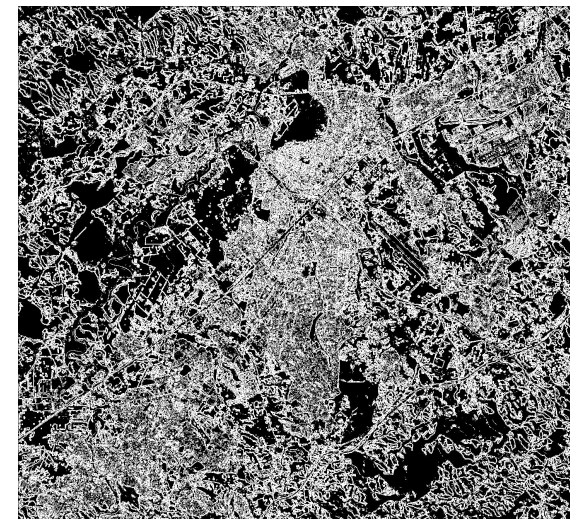
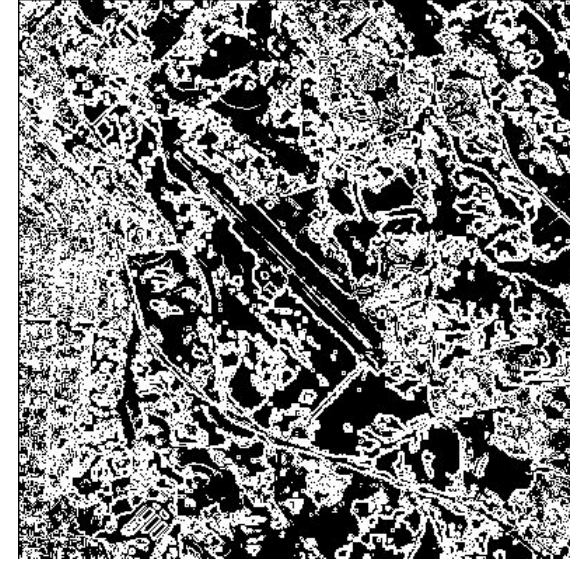
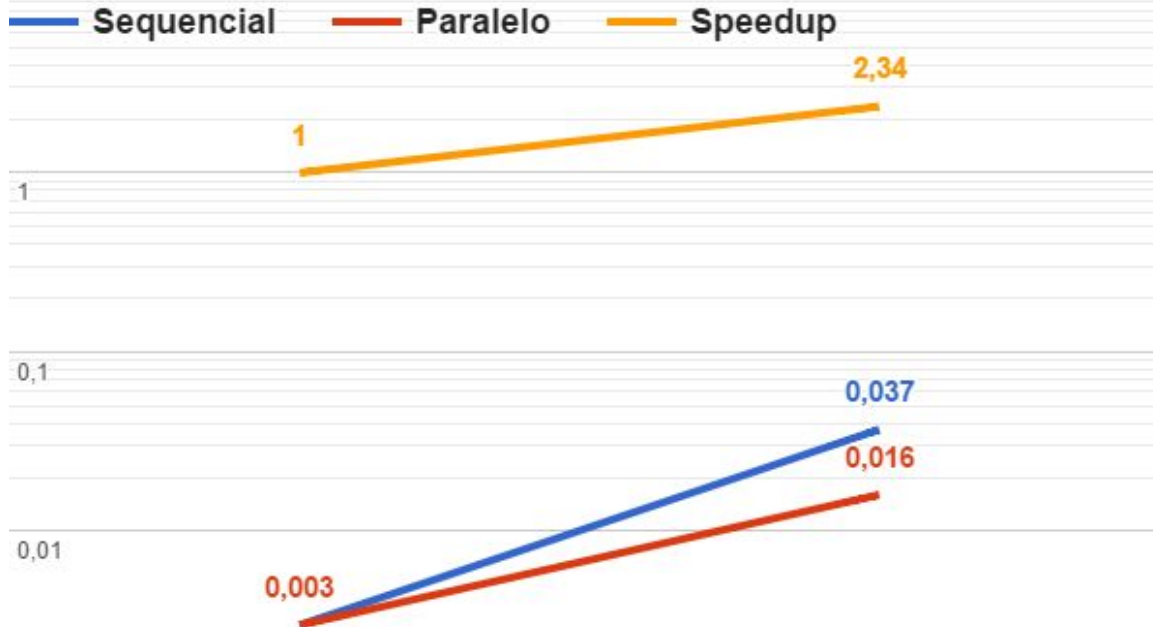
- **Diretiva de compilador**
 - **C/C++: #pragma omp parallel**
 - **Fortran: !\$OMP PARALLEL**
- **Rotinas de biblioteca**
 - **omp_set_num_threads()**
- **Variáveis de ambiente**
 - **OMP_NUM_THREADS**

```
...  
omp_set_num_threads();  
#pragma omp parallel for  
for(int i = 0; i <10000; i++){  
    <<instruções>>  
}  
...
```

- Intel Core i3 2.4GHz
- Memória RAM: 3GB
- OpenMP 2.5

Sobel - OpenMP

Imagens / Tempo (s)





Framework para Programação Paralela

Arquitetura

Linguagem

API

Desenvolvido pelo Kronos Groups

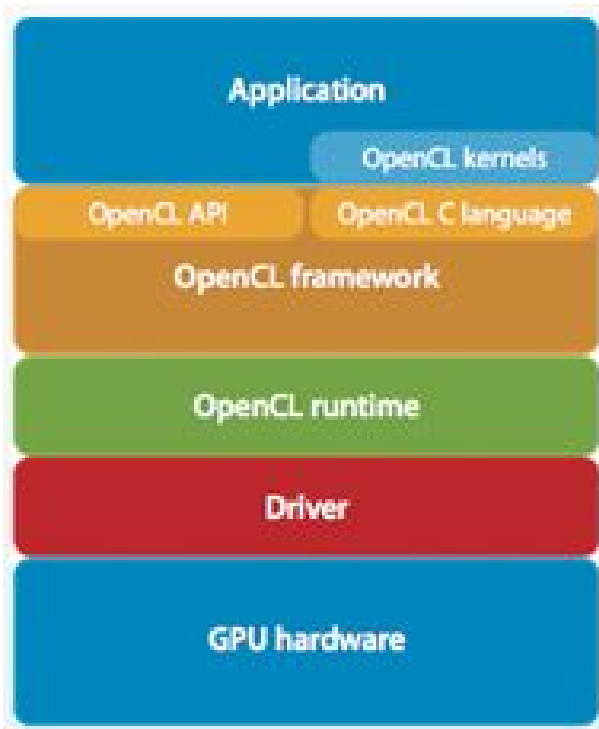
Nvidia

ATI
Technologies

Sun
Microsystem

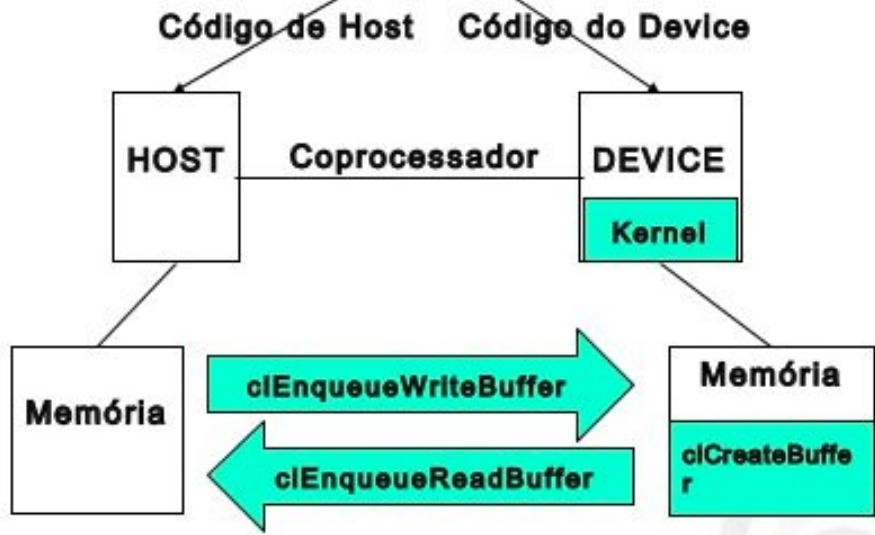
+ 120

AMD  **OpenCL™ APP SDK 2.6**



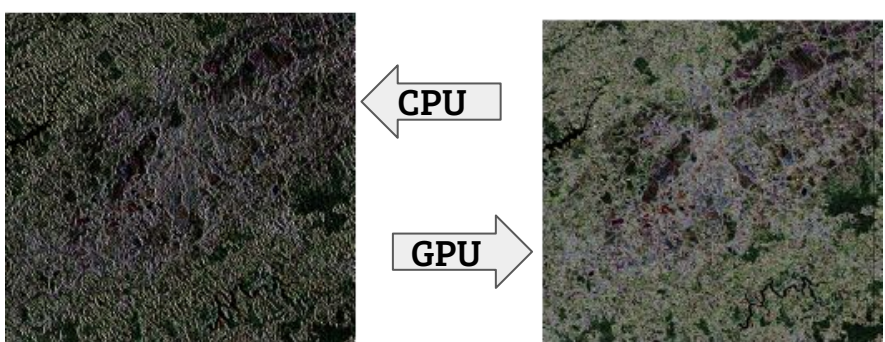
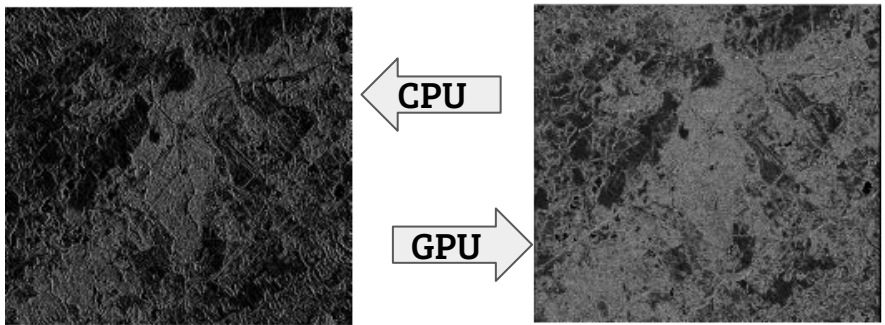
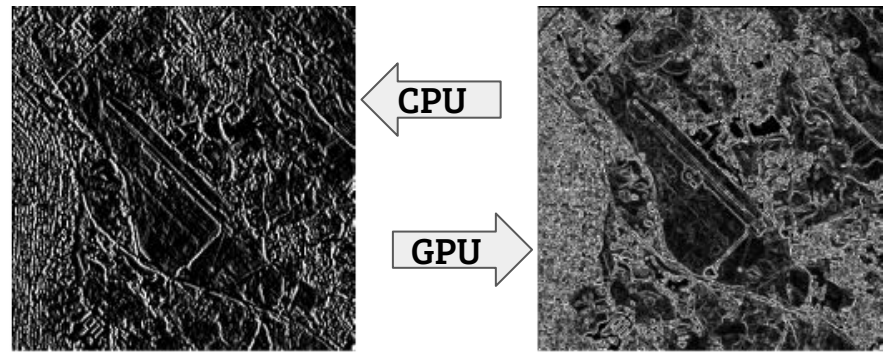
Modelo da Estrutura de Programação

Programa em OpenCL
(Escrito em C com extensões específicas)

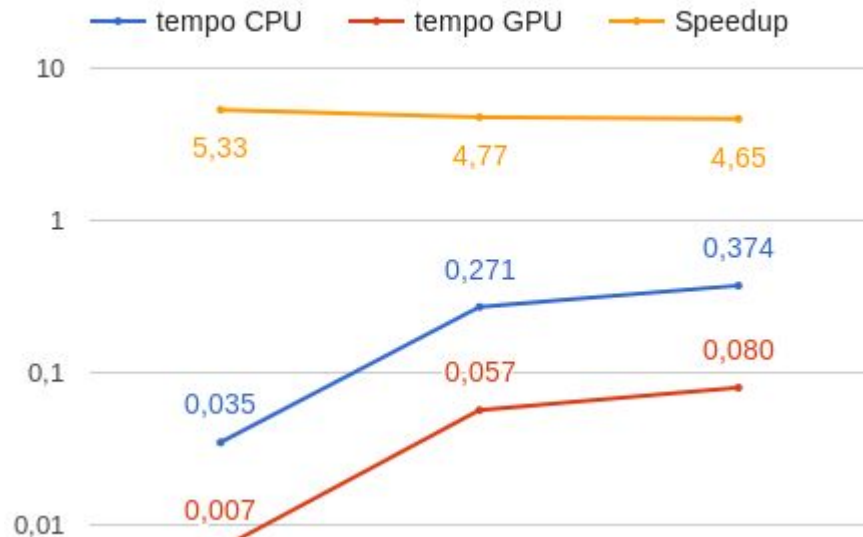


Modelo

- AMD A6-3420M 1.5GHz
- Memória 6GB
- Placa de Vídeo: Radeon Graphics HD 6720G2 2GB

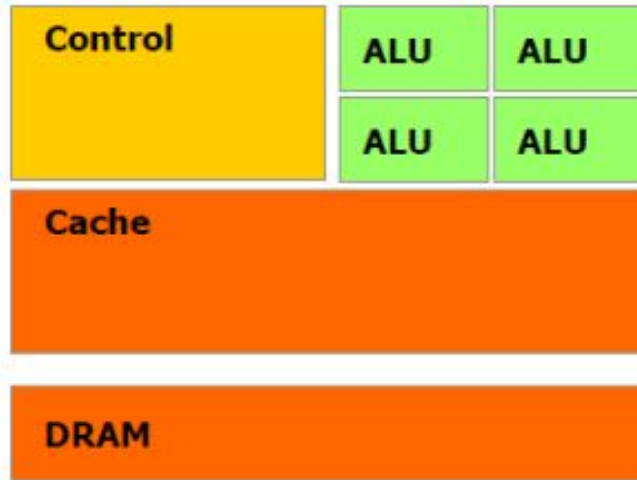


Sobel - OpenCL

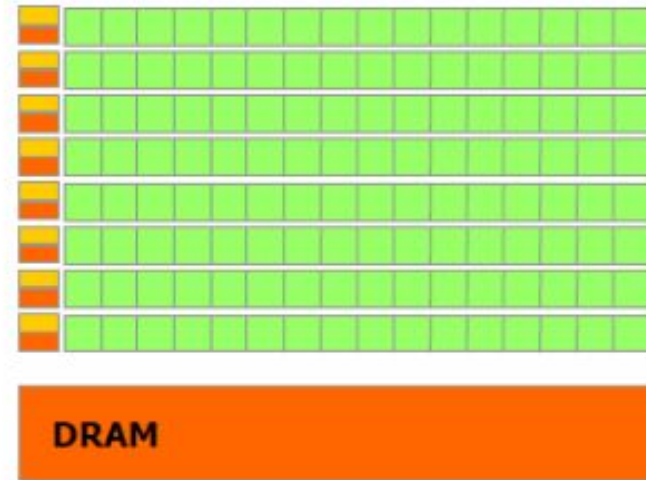




Arquitetura Massivamente Paralela

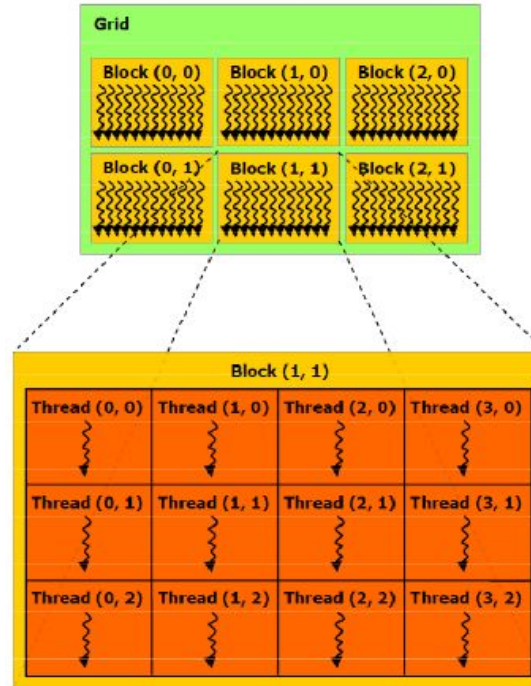


CPU

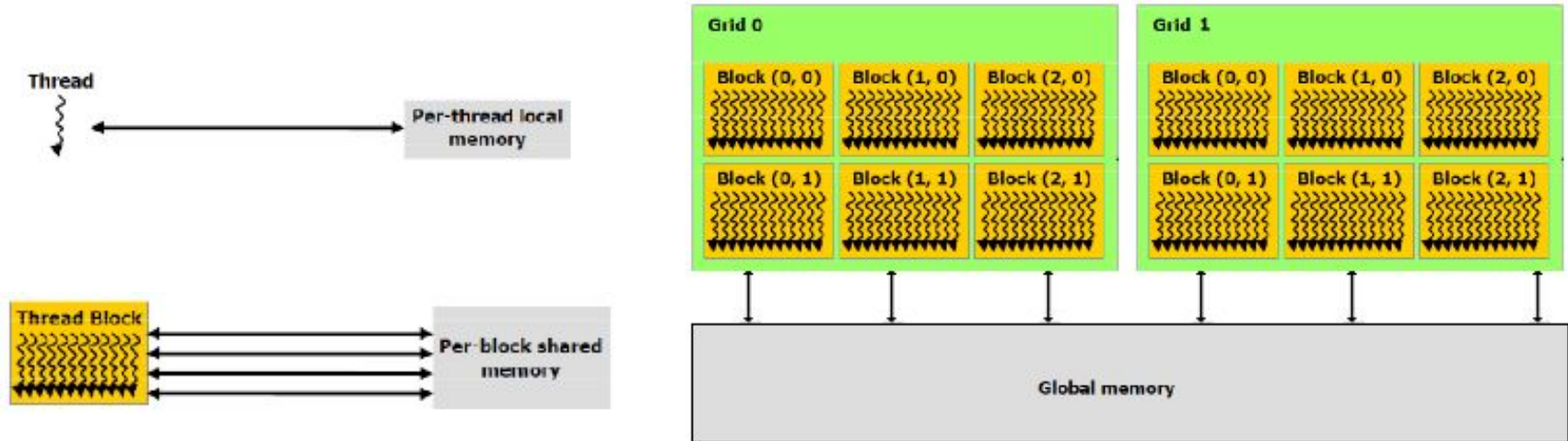


GPU

Arquitetura Multithreads



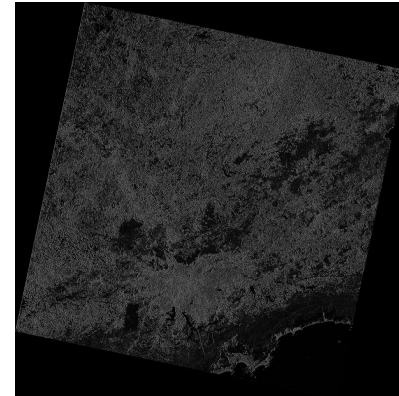
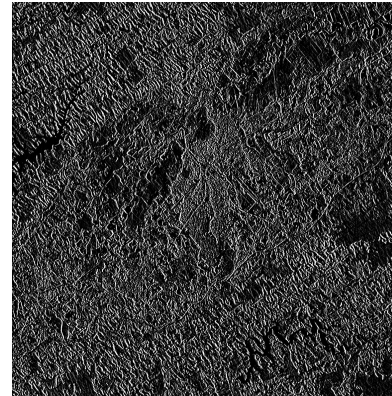
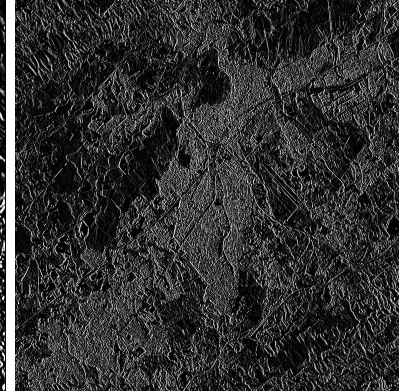
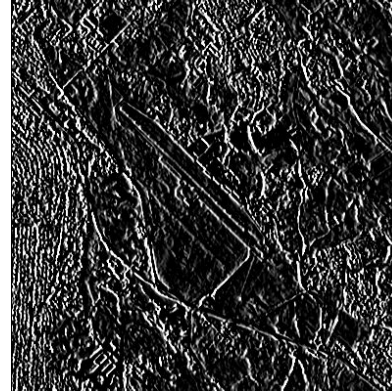
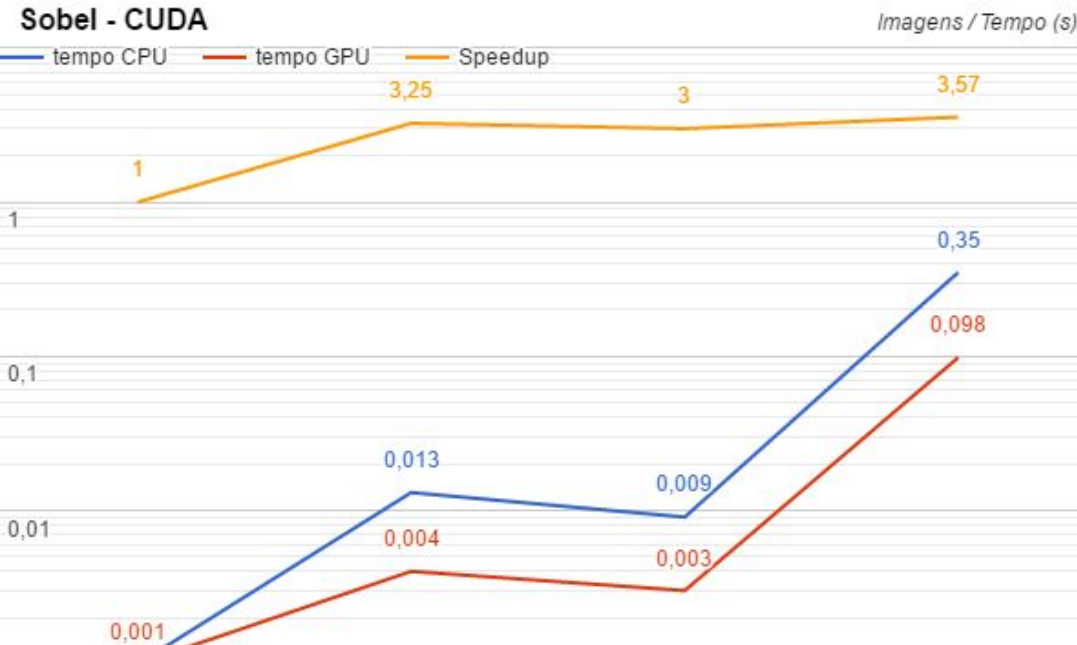
Hierarquia de Memória



- Intel Core i5 4690K 3.5GHz
- Memória 8GB
- Placa de Vídeo: NVIDIA Geforce GTX 950 2GB (768 CUDA Cores)



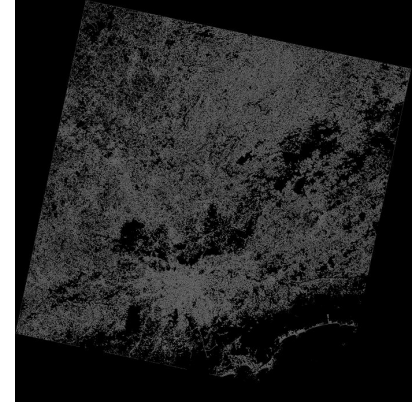
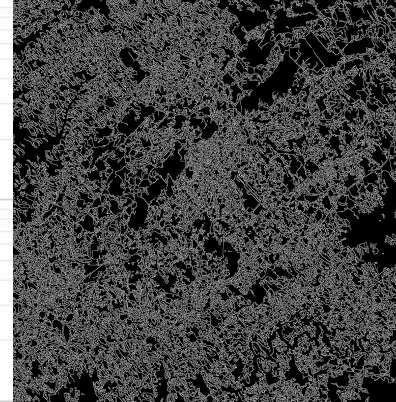
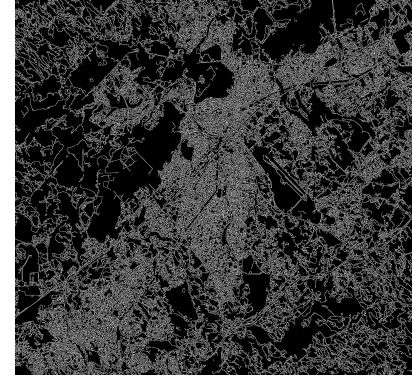
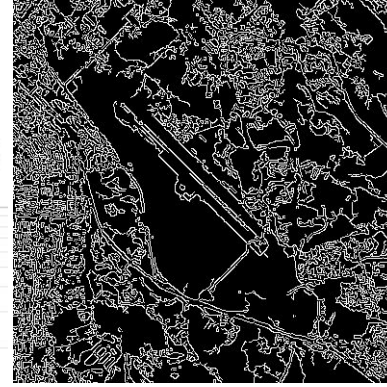
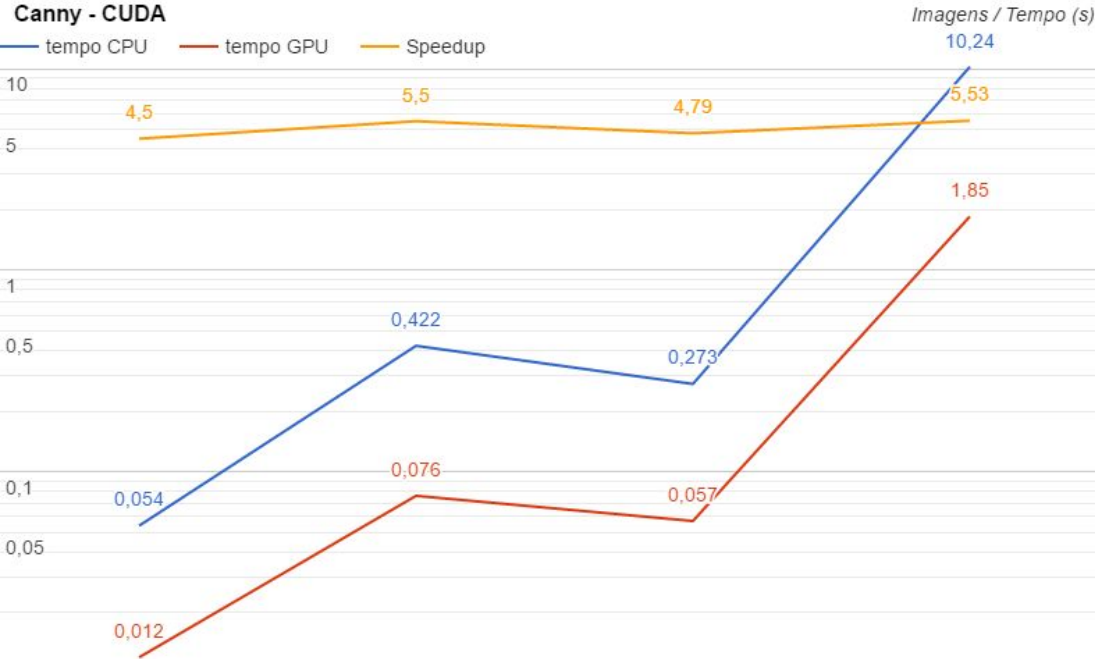
GPU Computing
CUDA



- Intel Core i5 4690K 3.5GHz
- Memória 8GB
- Placa de Vídeo: NVIDIA Geforce GTX 950 2GB (768 CUDA Cores)



GPU Computing
CUDA



Trabalhos Relacionados

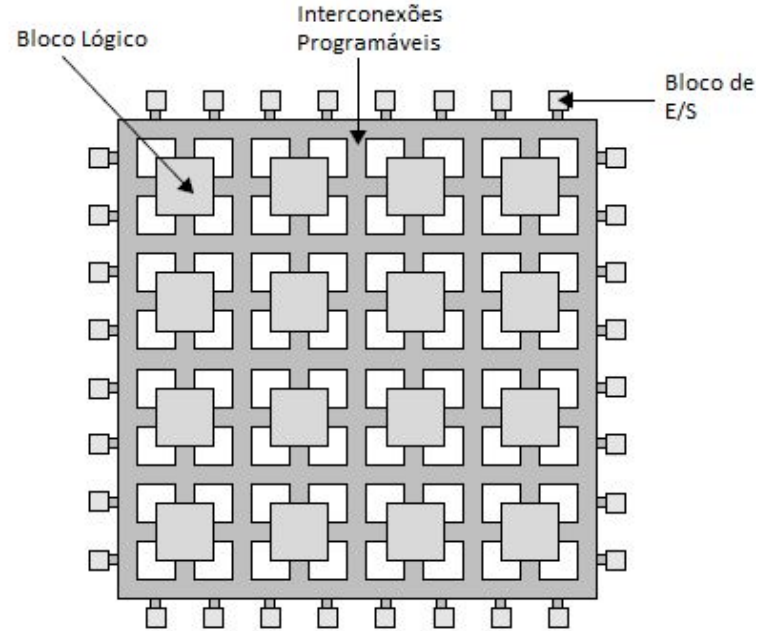
Field-Programmable Gate Array (FPGA)

- Aplicações críticas
- Altamente paralelizável
- Baixo custo de desenvolvimento



Fonte:

<http://cdn.electronics-eetimes.com>

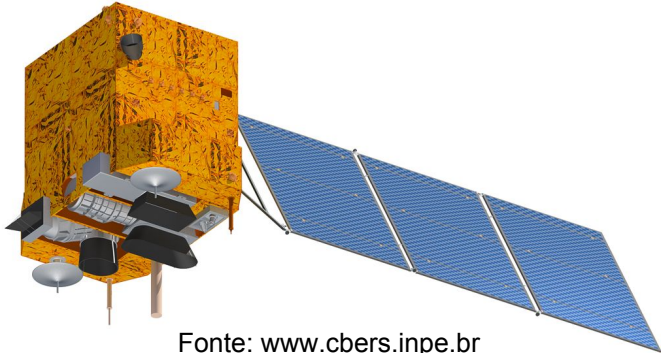


Fonte: (ZEIDMAN, 2002)

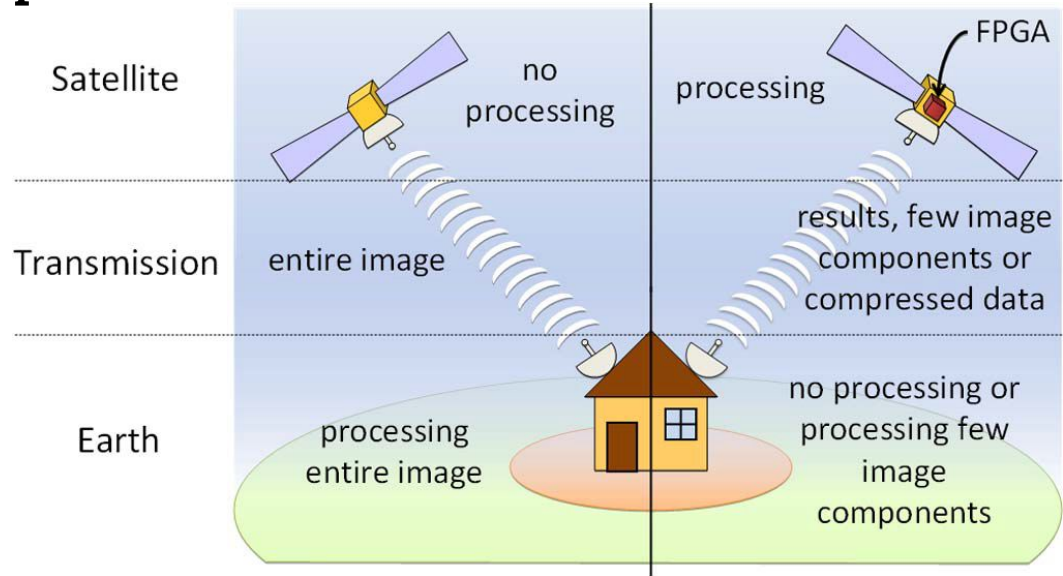
Trabalhos Relacionados

FPGA em Processamento de Imagens Espaciais

- Pré-processamento em tempo real
- Detecção de Nuvens
- Análise de Imagens
- Compressão



Fonte: www.cbbers.inpe.br



Fonte: (GONZÁLEZ et al, 2012)

Conclusão

- **Estudo das tecnologias de PAD**
 - **Sequencial, OpenMP**
 - **OpenCL, CUDA**
- **Aplicação das tecnologias em PDI**
 - **Realce**
- **Ganho de desempenho**
 - **Sequencial x Paralelo**
 - **Maior processamento \equiv Maior desempenho**

Referências

- YANG, L.; GUO, M. High-Performance Computing: Paradigm and Infrastructure. [S.l.]: Wiley, 2005.
- CHAPMAN, B.; JOST, G.; PAS, R. V. D. Using OpenMP: portable shared memory parallel programming. [S.l.]: MIT press, 2008.
- GONG, H. X.; HAO, L. Roberts edge detection algorithm based on GPU. Journal of Chemical and Pharmaceutical Research, v. 6, n. 7, p. 1308–1314, 2014.
- KRONOS. Getting Familiar with GCC Parameters. 2016. Acessado em 30 de julho de 2016.
- GAO, W.; HUYEN, N. T. T.; LOI, H. S.; KEMAO, Q. Real-time 2D parallel windowed Fourier transform for fringe pattern analysis using Graphics Processing Unit. Optics express, 2009.
- GONZALEZ, R.; WOODS, R. Processamento Digital de Imagens. [S.l.]: Pearson, 2010.
- BRADSKI, G.; KAEHLER, A. Learning OpenCV: Computer vision with the OpenCV library. [S.l.]: "O'Reilly Media, Inc.", 2008.
- ZEIDMAN, B. Designing with FPGAs and CPLDs. [S.l.]: Taylor & Francis, 2002.
- GONZÁLEZ, C.; MOZOS, D.; RESANO, J.; PLAZA, A. Fpga implementation of the n-findr algorithm for remotely sensed hyperspectral image analysis. 2012.
- SHAN, N.; ZHENG, T. y.; WANG, Z. s. Onboard real-time cloud detection using reconfigurable fpgas for remote sensing. In: 2009 17th International Conference on Geoinformatics. [S.l.: s.n.], 2009.

Tema: Processamento de Alto Desempenho com PDI

PROCESSAMENTO DE ALTO DESEMPENHO APLICADO AO PROCESSAMENTO DE IMAGENS DE SATÉLITES COM AS TECNOLOGIAS OPENMP, CUDA E OPENCL

CAP-378 Tópicos em Observação da Terra

Professores:

Antônio Miguel Vieira Monteiro

Sérgio Rosim

Grupo 3:

Marcos Lima Rodrigues

Marilyn Menecucci Ibañez

Rodrigo Augusto Rebouças

Viny Cesar Pereira



PÓS-GRADUAÇÃO

COMPUTAÇÃO APLICADA