Comparative study of methods of Principal Component Analysis of automatic segmentation of functional magnetic resonance imaging (fMRI).

CPU (Central Processing Unit) -wikipedia

GPU (Graphics Processing Unit) – wikipedia

GPU (Graphic Processor Unit) – CUDA documentation

1. Motivation - how to proces huge volume of data to obtain results in a real-time application
2. In the beginning of the last decade an increase of CPU (Central Processing Unit) clock speed was generally stopped. There are some reasons for that, the essential one is because of the thermal losses.
3. Nowadays in order to maintain continuous increase of the performance processors comprise many cores (multicores processor).
4. This implies that a paradigm of sequentially written programs has become unable to fully utilize this architecture. To achieve that it is necesarry to develop parallel applications i.e. applications which exploit all available cores so that the total execution time of the program is lower than the one implemented sequentially using one CPU.
5. In practice there are two main approaches to develop parallel applications. The first one is about processors containing several cores (2,4,6,8,…), each one (processor) processing several „heavy” threads. This approach is common for clusters of CPU processors.

Another type of processors are those which contain many cores (hundreds, thousands) being able to proces many „light” threads. This is how GPU (Graphic Processor Unit) works.

1. Nowadays numerical applications with big computational complexity are implemented mainly on GPU. This is because of some of the features they have. Graphic procesor unit (GPU) is specialized for compute-intensive, highly parallel computation - exactly what graphics rendering is about - and therefore designed such that more transistors are devoted to data processing rather than data caching and flow control [1 cuda programming guide].
2. Tutaj może o tym co na slajdzie 11 pierwszego wykładu RIM – dlaczego cuda – cena rozpowszechnienie, itd.

Other reasons for that is they are cheap and very common (many GPUs are able to do computational work).

1. Objectives

The aim of this work was to implement PCA algorithm on CUDA (Compute Unified Device Architecture) platform in order to obtain the better performance (speedup) regarding to matlab version of the algorithm. The study includes the comparison of the execution time of the methods implemented in Matlab and CUDA. These methods were tested for real problems (on data sets obtaining from fMRI scanning).

1. Explanation of CUDA and Matlab computational platform

NO DO TEGO DO PRZEDE WSZYSTKIM TEŻ WYKŁAD 2 RIM

Cuda programming guide – początek, tam jest dużo o platformie cuda

Hardware implementation

Jak zacząć? – może slajd 6 pierwszego wykładu RIM

Może też coś z 12 slajdu

1. Guide / rim
2. Guide / rim
3. Slajd 12

Osiągnięcia technologii cuda:

1. Duża i buforowana pamięć programu
2. Model programowania wielką liczbą hierarchicznie zorganizowanych wątków, które

Mogą się synchronizować na barierze i komunikować przez pamięć współdzieloną

1. Przezroczysta skalowalność i zgodność w przód

CUDA language – as its core are three key abstractions – a hierarchy of thread groups, shared memories and barrier synchronization

Model SIMD (Single Instruction, Multiple Data)

Cuda operates on SIMT (Single Instricution, Multiple Threads)

Wydajność

Przepustowość – very high memory bandwidth

Wykorzystanie krzemu

For example about flops and bandwidth (wydajność , przepustowość – memory bandwidth) - o tym pod koniec, bo wydajność przepustowość wynika z pewnych cech CUDA, o których mamy napisać

JEST RÓŻNICA POMIĘDZY ARCHITEKTURĄ GPU A PLATFORMĄ CUDA

PLATFORMA CUDA TO HOST Z PODLEGAJĄCYMI MU URZĄDZENIAMI GPU

Plan na podstawie wykładów RIM:

1. Różnice między CPU a GPU – slajd 6, wykład 1
2. Platforma CUDA – wykład 2 RIM
3. Dlaczego GPU – wydajność, przyspieszenie, przepustowość, wykorzystanie krzemu, osiągnięcia technologii CUDA – wykład 1 RIM

I w pomiędzy te punkty wcisnąć coś z programming guide cuda / hardware implementation / simt architecture

1. CPU cores are designed to execute instructions sequentially, so they are specialized in flow control. They have much bigger cache than GPUs to minimize memory access latency, because memory bandwitdh in CPUs are generally low.

On the other hand GPU’s architecture was optimized for games, so they contain many simple floating-point ALU executing in groups millions of instructions. The control flow is simplified. Many „light” threads executing simultaneously, hiding memory access.

1. CUDA platform consists of a host (CPU) and one or more devices (GPU) under host’s control.

GPU processor