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

1. Motivation - how to proces huge volume of data to obtain results in a real-time application
2. Zatrzymanie się rozwoju szybkości zegara procesora.
3. Zwiększanie mocy obliczeniowej komputerów poprzez dodawanie kolejnych rdzeni
4. Konieczna zmiana paradygmatu z programów sekwencyjnych na równoległe.
5. Dwie ścieżki rozwoju – kilku rdzeniowe – każdy rdzeń wspiera jeden cieżki wątek lub procesory wielordzeniowe (tysiące) wspierające lekkie wątki
6. Aplikacje numeryczne o dużej złożoności obliczeniowej implementowane są obecnie głównie na gpu – dlaczego – tutaj krótko a więcej w explanation.
7. In the beggining of the last decade an increase of cpu clock speedup was generally stopped. There are some reasons for this, the essential one is because of the thermal losses.
8. In order to maintain continuous growth of performance/ increase performance nowadays processors comprise many cores.
9. A paradigm of sequentially written programs has become unable to fully utilize this architecture. To do that it is necesarry develop parallel applications i.e. app will exploit all available cores as much as possible
10. 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 processing several „heavy” threads. In the second approach processors contain many cores (hundreds, thousands). These processors are able to process the same numer of „light” threads.
11. Nowadays numerical applications with big computational complexity are implemented mainly on gpu processors . 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].
12. 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

Wykład 1 RIM

Model SIMD (Single Instruction, Multiple Data)

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