MATERIAL for Lab 3 of Advanced Computing Systems

Groups 1 to 13: Brain simulation algorithm by Hoang-Anh Du-Nguyen

- Thesis and paper (see BRAIN folder in zipfile)
- Source code (C code and GPU code) of brain simulation

Groups 14 to 26: DOPA GPU implementation by Marijn Kentie

- Thesis and paper (see SW folder in zipfile)
- Source code (C code and GPU code) of Smith-Waterman
- For realistic protein alignments you can use the latest version of Swiss-Prot ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.fasta.gz

LAB ASSIGNMENTS

Groups 1 to 13: Brain simulation algorithm

1. Read the reference (thesis) and understand how the algorithm functions to simulate a brain neuron model. Run the provided C code implementation (for a small number of cells up to 16384 cells) on CPU to benchmark the performance.
2. Analyze the GPU implementation in the thesis and write a short summary of the techniques used to accelerate the implementation and discuss the results.
3. Run the GPU implementation on the GeForce GTX750ti on similde.et.tudelft.nl. Compare the speedup you get on the system with the performance reported by the thesis (see Table 5.3). Explain the increase/decrease in measured performance. Remember that CUDA code is portable, but performance is not portable.
4. Create your own accelerated GPU implementation of the C code of the brain model on the GTX750ti. Compare the speedup you get on the system with the performance you get from Question 3 and with that reported in the thesis. Discuss the differences. What is the fundamental limiting factor for performance?

Groups 14 to 26: Protein alignment algorithm

1. Read the references (thesis and papers) on the Smith-Waterman algorithm and understand how the algorithm functions to align two different protein sequences. Run the provided C code implementation on CPU to compare performance using the same sequences mentioned in Table 1 in the DOPA paper (database and sequences are on http://kentie.net/article/thesis/index.htm)
2. Analyze the DOPA GPU implementation and write a short summary of the techniques used to accelerate the implementation and discuss the results.
3. Run the GPU implementation on the Tesla C2075 on ce-tesla.et.tudelft.nl. Compare the speedup you get on the system with the performance reported by DOPA. Explain the increase/decrease in measured performance.
4. Create your own accelerated GPU implementation of the C code implementation of Smith-Waterman on the Tesla C2075. Compare the speedup you get on the system with the performance you get from Question 3 and with that reported by DOPA. Discuss the differences. What is the fundamental limiting factor for performance?

Note: do not include code in your report, rather include it as part of the zip package you submit.