KARLSRUHE INSTITUTE OF TECHNOLOGY

Mehrdimensionale Signalverarbeitung und Bildauswertung mit Graphikkarten und anderen Mehrkernprozessoren

Manuel Lang

March 29, 2018

Contents

1 Introduction

- Parallel computing: Use of multiple (interacting) computational units(CU) to execute a (divisible) task.
- Amdahls law: We want to know how fast we can complete a task on a particular data set by increasing the CU count. $\eta_n = \frac{TW}{T(n)W} = \frac{T}{t_s + \frac{t_p}{n} + t_{comm}}$
  - η_n: Speeup
  - W: Work load
  - T: Total runtime $t_s + t_p$
  - t_s: Runtime serial part
  - t_p: Runtime parallel part
  - t_{comm}: Communication time (normally not included)
  - n: Number of computational units
- Gustafson's law: We want to know if we can analyze more data in approximately the same amount of time by increasing the CU count. $\eta_n = \frac{TW(n)}{TW} = (1-p)W + npW$
  - η_n: Speeup
  - W: Work load
  - T: Total runtime
  - p: Workload fraction benefiting from additional CUs
  - $W(n) : (1-p)W + npW \equiv aW + n(1-a)W = nW - a(n-1)W$
  - n: Number of computational units
- Data parallelism: Each CU performs the same task on dierent data
  - The CPU cores and GPU streaming-cores are OpenCL compute devices
  - Concurrent processing on all herterogeneous cores.
- Task parallelism: Each CU performs a different task on the same data
- Instruction level parallelism: Automatic parallel execution of instructions by processor
- Spatial parallelism: More units work in parallel
- Temporal parallelism → Pipelining
- Latency
  - The latency of an instruction is the **delay** that the instruction generates in a dependy chain. The measurement unit is clock cycles.
  - CPUs try to minimize latency. Low efficiency on parallel portions.

- Throughput
    - The throughput is the maximum number of instructions of the same kind that can be executed per clock cycle when the operands of each instruction are independant of the preceding instrucions.
    - GPUs try to maximize throughput. Low performance on sequentiel portions.
- Graphic card slang
    - A GPU executes a program, the kernel.
    - A thread executes an instance of the kernel.
    - Threads are combined into warps/wavefronts running in lockstep. Individual threads composing a warp start together at the same program address, but they have their own instruction address counter and register state and are therefore free to branch and execute independently.
    - Warps/wavefronts are part of threaded blocks/work groups. These are defined by the user.
    - A thread runs on a core. A number of cores form a *Streaming Multiprocessor (SM) / Compute Unit (CU)*. Thread blocks/work groups are scheduled over SMs/CUs.

2 Parallelism / Programming models

3 OpenMP

4 OpenACC

5 OpenCL

5.1 OpenCL-API

5.2 OpenCL-C

5.3 OpenCL-Memory

6 Optimization for CPUs

7 Optimization for GPUs

8 SIFT Optimization