INF5063: Programming heterogeneous multi-core processors

… because the OS-course is just to easy!

Home Exam 3: Distributed Video Encoding using Dolphin PCI Express Networks

November 3rd 2016

Håkon Kvale Stensland

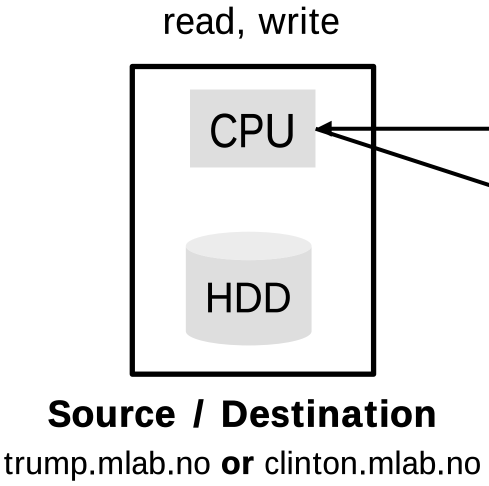
Lab Hardware for Home Exam 3

- **nVIDIA Quadro K2200**
  - **All machines have the same GPU**
  - **1st Gen Maxwell Architecture**
- Based on the full GM107 chip
  - 1870 million transistors
  - *640 Processing cores (SP) at 1000 MHz (5 SMM)*
  - *4096 MB GDDR5 Memory with 80 GB/sec bandwidth*
  - 1280 GFLOPS theoretical performance
  - Compute version 5.0
- *Supports GPUDirect RDMA*

Precode

- Same precode as Home Exam 1 and 2!
- Use makefile and program structure from non-mandatory assignment 3.
- You are not allowed to change out the Motion Estimation, Motion Compensation or DCT algorithms.
- You are **not** allowed to paste code from other projects / encoders.
- You only need to optimize the Codec63 encoder!
- Your implementation must use Dolphin PCI Express networks for communication between the machines.

Architecture for Home Exam 3



- 1 source and destination machine (*Intel Sandy Bridge*).
  - clinton.mlab.no *or* trump.mlab.no
- 1 processing machine (*Intel Haswell*) and a Nvidia GPU based on the Maxwell architecture (Quadro K2200).
  - mpg-2014-xx.mlab.no
- All machines are connected with a PCI Express switch.

Your task: I/O Machines

- You can have up to three frames “in flight” (not counting any double buffering).

- DMA is recommended for data transfers, use PIO for synchronization. Remember, you might have to try multiple approaches. There are also different techniques for using SISCI to communicate between machines.

- **Source & Destination (clinton.mlab.no *or* trump.mlab.no):**
  - Four cores are available on this machine, no GPU.
  - All machines use x86, but the I/O machines have a slightly older microarchitecture (Sandy Bridge)

- The use of two machines and PCIe interconnect is an absolute requirement for passing this exam.

Your task: Processing Machine

- No extra points will be given for using MMX, SSE or AVX on the processing machine.
- Use the GPU on the processing machines for at least Motion Estimation.
- The efficiently of your ME, MC and DCT/iDCT will not be evaluated on this assignment, this has already been evaluated on H2.
- You can use asynchronous transfers between CPU and GPU (CUDA streams) on the processing machines.
- Evaluate features like GPUDirect RDMA to copy data directly into GPU memory from the source machine.
- In the report you should describe all the optimizations that have been done, also the ones that did not work.

How are you evaluated?

- Make sure that your implementation **compiles** and **run**, and that it can **produce correct video output** (we also check the motion prediction). Our main focus will be on tractor video!
- Is different strategies for using SISCI evaluated? Is optimizations like GPUDirect evaluated?
- Communication and synchronization protocol between the IO machine and the processing node
- Is both the I/O and processing node optimized?
- Quality of the report. Is profiling of the code done between the different steps and how are the different optimization attempts documented and discussed in the report.
- Use of GPU for Motion Estimation, Motion Compensation and DCT/iDCT. ***Not the local performance tuning that you achieve, but the gains achieved by parallelization and distribution (already evaluated on H1 and H2).***
- Presentation of your solution in the “poster session” is required to pass the exam!

Formal Information

- Deadline: **Friday November 25th – 15:00**

- The assignment will be graded, and count 33% of the final grade.

- Deliver your code, report and poster to: https://devilry.ifi.uio.no/

- Prepare a poster (two A3 pages) and a short talk (2 minutes without slides) to pitch your poster for the class (November 30th). Best poster & presentation will be awarded!

Competition!

- Will be announced on Wednesday November 21st
- Winners will be announced during the last session on November 30th
- Prizes will be awarded to the best groups!
- Prizes for best poster and presentations will also be awarded on this final session!

Last but not least!

- Codec63 precode available for download in git. Clone the repository and work on you own local version.

```
git clone https://bitbucket.org/mpg_code/inf5063-codec63.git

git clone https://bitbucket.org/mpg_code/sisci-assignment.git
```

- Bugs in the code can be reported in Bitbucket’s issue tracking system, or Slack.

Good Luck!

PS! Start early!