Many-cores Accelerators Evaluation at CERN/Openlab

S. Jarp, A. Lazzaro, J. Leduc, A. Nowak
CERN openlab

International Conference on Computing in High Energy and Nuclear Physics 2010 (CHEP2010)

October 21st, 2010

Academia Sinica, Taipei

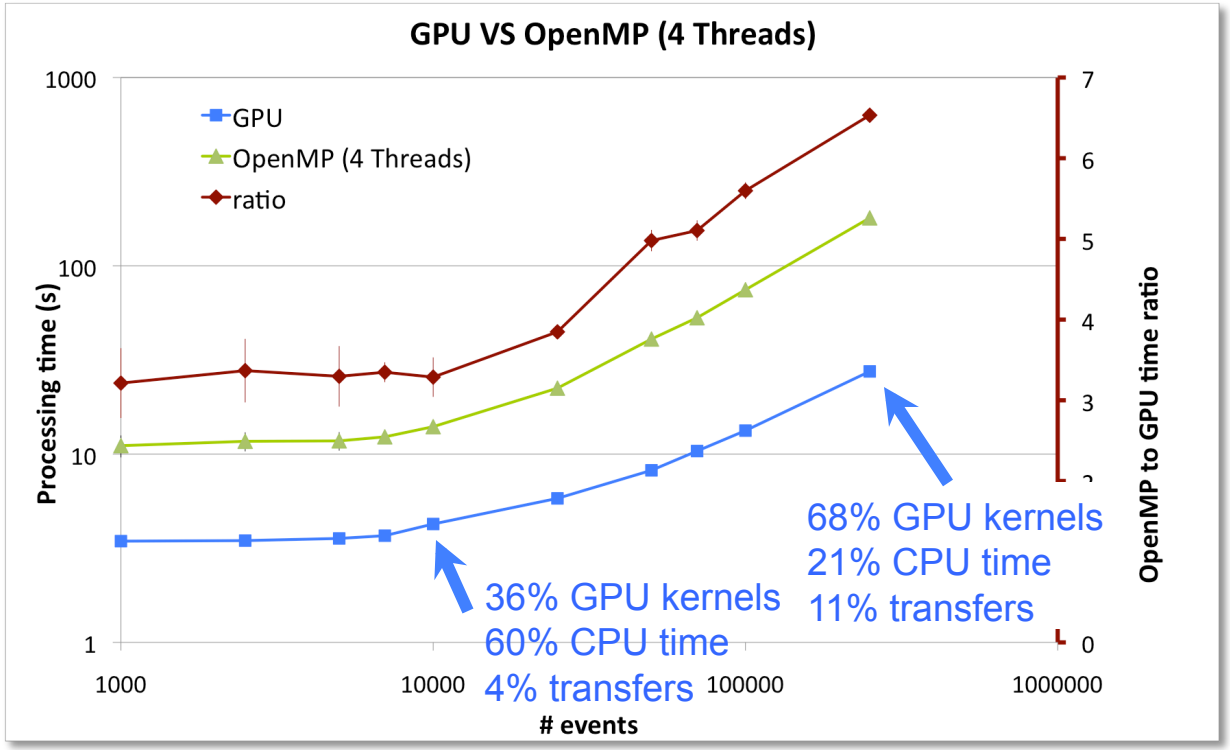
Presentation by Alfio Lazzaro

Recent activity

- Evaluated on a commodity card (GTX470 Nvidia) the possibility of porting the code for maximum likelihood fitting (data analysis) based on ROOT/ RooFit
  - See my presentation this afternoon at **Parallel Session 49: Event Processing**
    - http://117.103.105.177/MaKaC/sessionDisplay.py?sessionId=79&slotId=0&confId=3#2010-10-21
- The code is based on CUDA implementation and it was mainly carried out by an openlab summer student, Felice Pantaleo (2.5 months work)
  - About 1 month spent to take familiarity with the card and the code

GPU VS OpenMP

- Fair comparison
  - Same algorithm for GPU and CPU
  - Algorithm on CPU optimized and parallelized (4 threads)
  - Some calculation done by only CPU
- Check that the results are compatible: asymmetry less than 10^{-12}



- Speed-up increases with the dimension of the sample, taking benefit from the data streaming on GPU and the integral calculation only on the CPU
  - ~3x for small samples, up to ~7x for large samples

Results

- Implementation of the algorithm in CUDA
  - Require not so drastic changes in the existing RooFit code
  - New design of the algorithm
- The CUDA implementation “forces” us to develop an OpenMP implementation on the CPU of the same algorithm
  - With 1 thread +34% better performance with respect to RooFit implementation
- In our test GPU implementation gives >3x speed-up (~7x for large samples) with respect to OpenMP with 4 threads
  - Note that our target is running fits at the user-level on the GPU of small systems (laptops), i.e. with small number of CPU cores
- Now working on generalization of the code to be included in ROOT for testing by users
  - At the moment there is not clear deadline

Future work

- Try to use OpenCL
  - The great benefit is the possibility to have hardware-independent code, i.e. GPUs (NVidia, AMD, Intel) and CPUs
  - There are issue related to the implementation that are to be investigated.
  - In contact with a guru, Tim Mattson (Intel)
- **It turns out that the new implementation of the algorithm (which is required to run on the GPU) gives better performance on the CPU and it is easy to parallelize (using OpenMP)**
  - We will continue to improve this version. This is our first priority
- We are working on the evaluation of the Knights Ferry (32 cores) and soon of the Single-Chip Cloud Computer (48 cores, no cache coherency), as part of the collaboration with Intel
  - Very promising architectures for massive parallelization with intensive calculations
  - It can be put in the general context of accelerators