GPGPUs and LHCb

LHCb is evaluating GPGPU technologies and related issues in an effort to make hardware decisions for Run 3 circa March 2017. A key element is developing a number of demonstrators as "proof-of-principle" projects. Success is deemed necessary, but not sufficient, to move in this direction. Some important questions related to using GPUs are also important for other architectures: how do we convert our algorithms to be "stateless"? how can the framework manage GPUs and other accelerators? how do we write efficient parallel algorithms to take advantage of SIMD and vector processors? how do we determine the functional equivalency of algorithms which produce architecture-specific results? how do we manage memory usage? what level of expertise is required to write and maintain good code? how should we evaluate life-cycle hardware and software costs? In this presentation, I will discuss elements of the Roadmap for an Upgrade Software and Computing TDR produced earlier this year.

LHCb's Roadmap for an Upgrade Software and Computing TDR & GPU Considerations

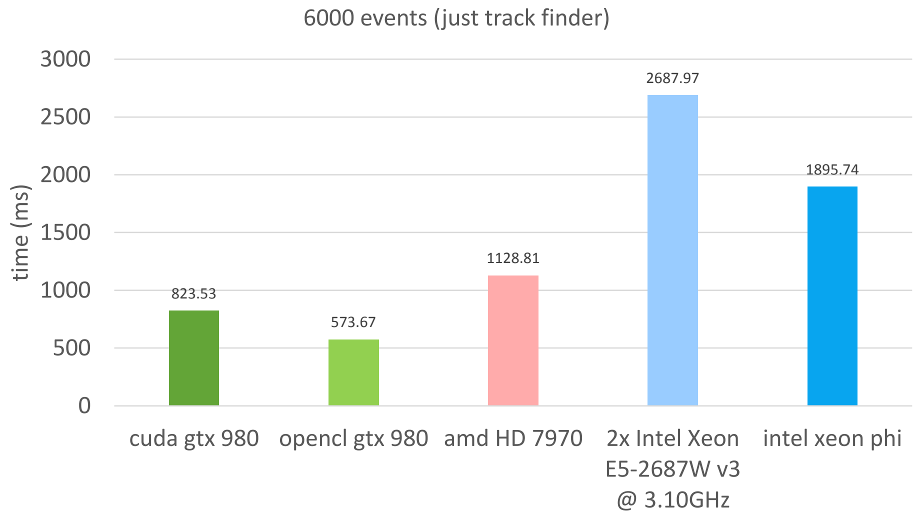
- LHCb plans to prepare a Run 3 Technical Design Report for Software and Computing by the end of calendar 2017.
- A decision whether to use GPU and/or other accelerators needs to be made circa March 2017.
- LHCb plans to eliminate its level 0 hardware trigger and process 40 MHz of events using a pure software trigger.
- The current framework does not support multi-threading, much less efficient use of vector and SIMD processors.
- Trying to exploit heterogeneous resources efficiently will require dynamically scheduling algorithms balance loads.
- Current algorithms are designed for serial execution. Functionally equivalent parallel algorithms need to be developed and their relative performances and quantified.

Demonstrator Projects

- The primary goal of these projects is to determine whether GPU codes can match the physics performance of bench serial codes and provide significantly better performance.
- The targets for the next year are primarily reconstruction algorithms required for Hlt1. The highest priorities are:
  - VeloPix – find straight line segments in a pixel detector
  - Vertex finding – build primary and secondary vertex candidates from VeloPix straight tracks
  - VeloUT tracking – project VeloPix tracks into a strip detector sitting in a modest magnetic field.
  - Track fitting – determine track parameters and error matrices; anticipated to use as much as 60% of Hlt1 processor time in Run 3.

VeloPIX using OpenCL

Cross-architecture timing



- All are running the same algorithm, except for "cuda gtx 980", which is running a CUDA version.
- In the case of the Intel Xeon, all logical cores (40) were used in the test.
- All runs produced the same results, including the CUDA implementation.

VeloPIX Cost Comparison

Cost comparison for same throughput

Processor	Cost for one Intel Xeon throughput ($)
Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz	684.99
NVIDIA GTX 980	277.37 (= 508.00 * 0.546)
AMD HD 7970	204.24 (= 189.99 * 1.075)
Intel Xeon Phi	- (preproduction)

Note that the nVidia GPU is a gamer board, not an HPC board .

- Prices obtained from www.newegg.com
- NVIDIA GTX 980 comes bundled with *Assasin's Creed Syndicate*.

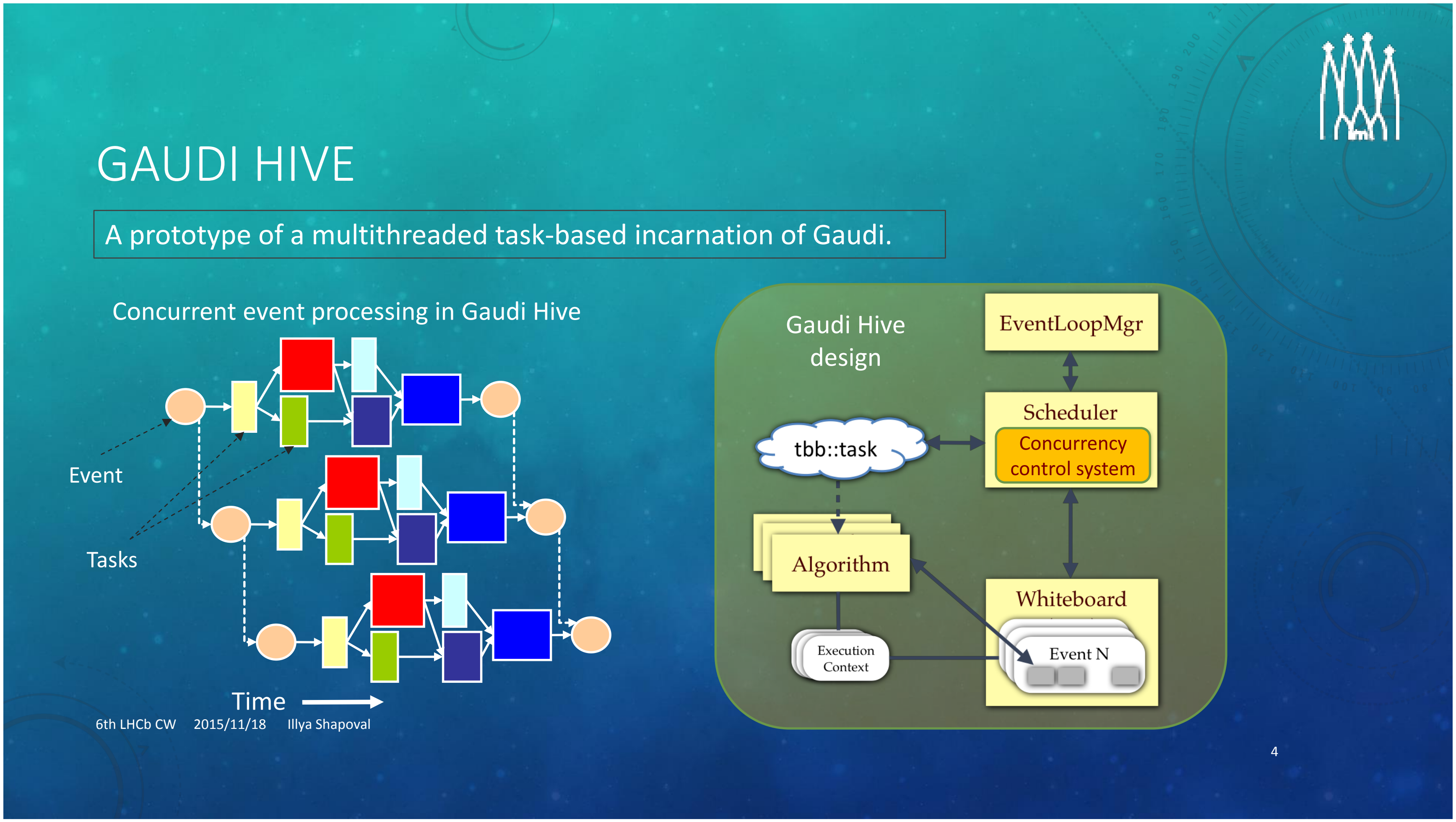
ONLINE

19/11/2015 DANIEL HUGO CÁMPORA PÉREZ - OPENCL VELOPIX CROSS-PLATFORM STUDIES 20

"Successful" GPU Demonstrator Projects are Necessary but Not Sufficient

- Algorithms which execute well on a GPU are necessary, but they will live inside a complex intra- and inter-event world.
- We will need a framework which supports multi-threaded executions and allocates resources to maximize occupancies of processors and minimizes latencies.
  - Gaudi Hive is a proto-type
- We will need an event model which organizes the data for effective use by vector and GPU processors.
  - No significant work done yet.
- We will need a model for developing and maintaining software.
  - No significant work done yet.

Gaudi Hive



GAUDI HIVE
A prototype of a multithreaded task-based incarnation of Gaudi.
Concurrent event processing in Gaudi Hive
Event
Tasks
Time
6th LHCb CW
2015/11/18
Illya Shapoval
Gaudi Hive design
EventLoopMgr
Scheduler
Concurrency control system
tbb::task
Algorithm
Whiteboard
Execution Context
Event N
4

Parallel Algorithms

- Some problems are "embarrassingly" parallel, and they are "easily" ported to GPUs.
  - Video processing is the classic use case;
  - Lattice QCD is a classic use case in the physics world;
  - Maximum likelihood fits are another example.
- Other problems can be re-formatted to be parallel by aligning data to take advantage of vector-like architectures, with GPUs being the extreme case of SIMD.
- LHCb algorithms are almost all explicitly serial. They will need to be re-written, perhaps from scratch, to take advantage of vector architectures of any type.
- Developing parallel algorithms should be useful for the next generation of hardware, be it "conventional CPUs", GPGPUs, Knight's Landing, or something else.

Evaluating Life Cycle Costs

- How performant are GPUs for individual reconstruction steps?
- In a realistic environment, how "commensurate" are GPU and CPU resources?
- Will using GPUs reduce capital and operating costs?
  - Will it be possible to use "gamer" processors rather than HPC boards?
- Will "ordinary" physicists be able to contribute to GPU software? What is the cost (broadly interpreted) of training enough personnel to write production quality code?
  - Is it necessary for the same people to develop the underlying algorithms and their implementations?
  - How will the software be maintained?

Questions To Be Addressed

- Need parallel algorithms to process data from individual events.
- Need an event model to support parallel algorithms.
- Need a multi-threaded framework to process data from independent events concurrently. It should also minimize latency and maximize occupancy within single events.
- Need realistic metrics for evaluating performance and life cycle costs.
- Is it necessary to use the same hardware/software online and offline. What does "functionally equivalent" mean?
- How much progress would be required in the next year to identify GPGPUs as part of our computing hardware model for Run 3?