

Intel MIC and the openlab experience

Andrzej Nowak
CERN openlab
November 28th 2012

**4th International Workshop for Future
Challenges in Tracking and Trigger Concepts**

Outline

- > **openlab's involvement with the MIC technology**
- > **Hardware**
 - Motivation
 - What the architecture means for physics
- > **Software**
 - Programmability
- > **Experiments**
- > **Summary**

All MIC results obtained on and quoted for pre-production hardware and software

Where is mainstream HEP now?

- > **Very limited or no vectorization**
 - Online has somewhat better conditions to vectorize
- > **Sub-optimal instruction level parallelism (CPI at >1)**
- > **Hardware threading unused, but often beneficial**
- > **Cores used well through multiprocessing – bar the stiff memory requirements**
 - However, systems put in production with delays
- > **Sockets used well**
- > **Multiple systems used very well**
- > **Relying on in-core improvements and # cores for scaling**

From “Larrabee” to Xeon Phi

- > **Project “Larrabee” was an x86 processor with wide vectors destined for graphics**
- > **Adapted into a “throughput computing” solution– the “Knights” family**
 - select Intel collaborators working in the program
 - CERN openlab amongst them
 - software opened up and supported for general purpose compute
- > **Successor(s) foreseen**

Intel MIC and openlab

Early access

- Work since MIC alpha (under RS-NDA)
- ISA reviews in 2008

Results

- 3 benchmarks ported from Xeon and delivering results: ROOT, Geant4, ALICE HLT trackfitter

Expertise

- Understood and compared with Xeon

Specific interests

- > What are the opportunities and challenges?**
- > Is the architecture adapted to what we need and compatible with what we have?**
- > Can we run our code without major changes?**
- > Will our code perform without major changes?**
- > Review: what are the opportunities and challenges?**

Intel MIC at openlab: specifics

- > **Now a major point on our agenda**
- > **openlab benchmarks stabilized, characterized and optimized where possible**
 - Ports finished and stabilized
 - Performance analysis and tuning
 - Extensive review of math function performance
- > **Feedback provided regularly to the Intel team**
- > **Pre-production hardware: multiple KNC cards installed**
- > **Increased interest as we got closer to official launch: 6 seminars given at CERN in 6 months**

MIC software - scenarios



Native mode

workload runs entirely on a MIC system (networked via PCIe)



Offload

MIC as an accelerator where host gets weak



Balanced

MIC and host work together



Cluster

application distributed across multiple MIC cards (possibly including host)



MIC – porting and writing software

- > **Ideal situation: just add a compiler switch and recompile**
- > **Less-than-ideal: minor adaptations, including GCC/ICC differences if any + above step**
- > **More likely: write parallel code or parallelize existing code + above steps**
- > **Numerous libraries available: OpenMP, MPI, TBB, Cilk, MKL etc**
- > **Vectorization (data parallelism) is key to achieve full performance**
- > **Target: OSS support, actively sought by the HEP community**

Ported benchmarks

- > **Experimental benchmarks**
- > **ALICE Trackfitter prototype - DAQ**
 - Data intensive
 - Vectorized, threaded
- > **Multi-threaded Geant4 prototype - simulation**
 - Test40 and ParFullCMS
 - Heavy threads with pthreads, no vectorization
- > **MLFit - analysis**
 - Vectorized, threaded
 - Sensitive to floating point results
- > **HEPSPEC06 (test only)**

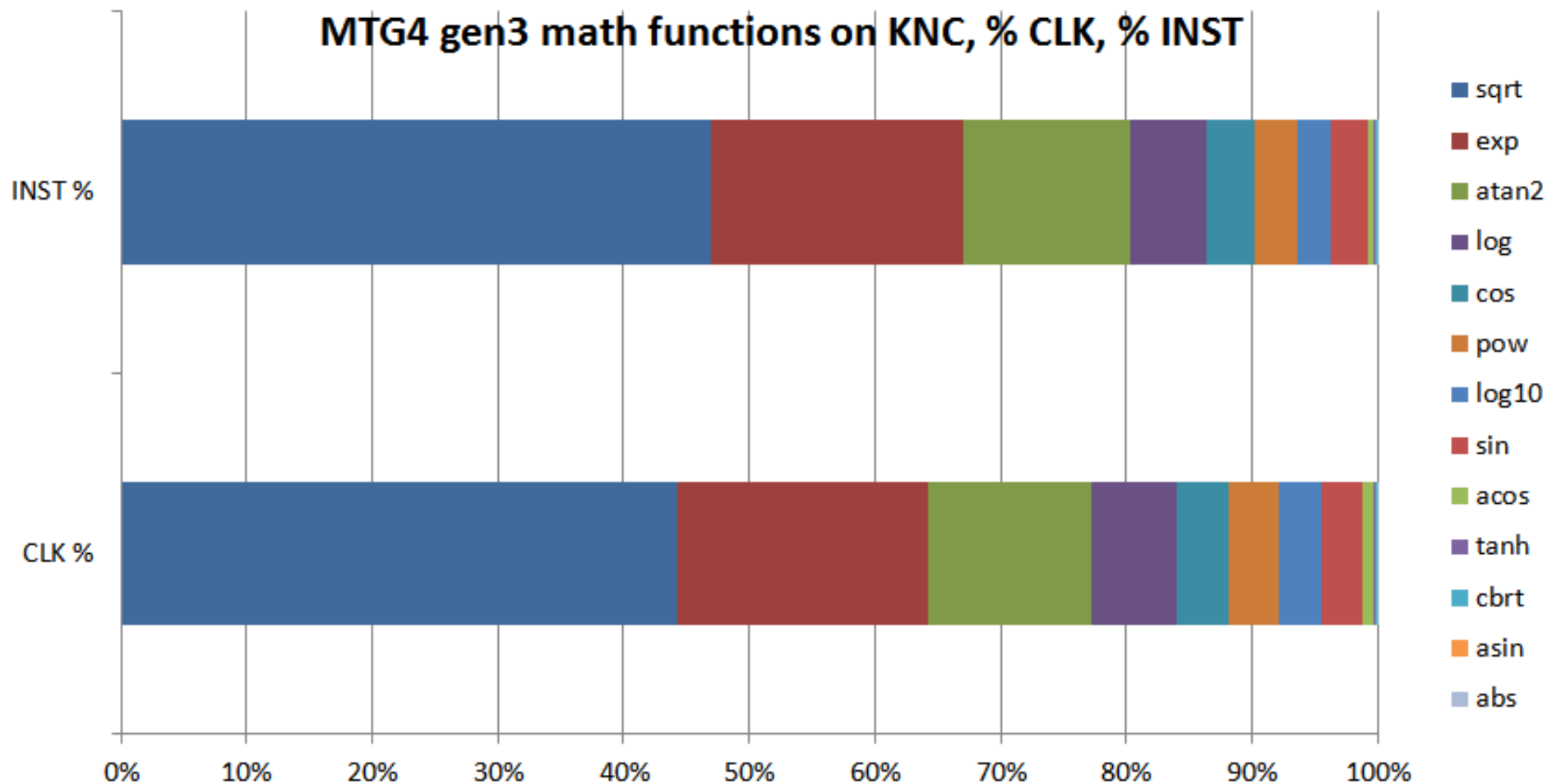
Trackfitter conclusions

- > Parallelism on multiple levels can be easily expressed in several ways**
- > Relatively straightforward port of a vectorized benchmark**

Geant4 conclusions

- > **Initial port difficult**
 - Unstable benchmark
 - Unstable (pre-alpha) SW environment
 - Unstable (pre-alpha) HW environment
- > **Benchmark sub-packages not fully ready to be ported to a difference architecture/OS**
 - In particular, cross-compilation was causing problems
- > **Subsequent ports performed in a matter of only 4-6 hours each**
- > **Vectorization is imperative to achieve performance**
- > **SMT is important for good performance**
 - up to 2x, progressive improvements
 - Full thread count run
- > **DP math functions need more attention**
- > **The compiler plays a large role and is not yet equivalent to the Xeon one**

MTG4 prototype “math only” profile



MLFit conclusions

- > Various popular parallelization environments work**
 - MPI, OpenMP, TBB, Cilk, etc.
- > The compiler plays a large role**
 - Multiple switches tried
- > Floating-point compatibility with Xeon not understood in sufficient detail to enable symmetric mode**

Expanding collaboration with the PH department and experiments

- > Consulting on new and existing technologies
- > Assistance with computing systems
- > Active engagement in the Concurrent Frameworks forum
- > Made MIC cluster and tools available to research groups
- > Multiple seminars given

Where will we be tomorrow?

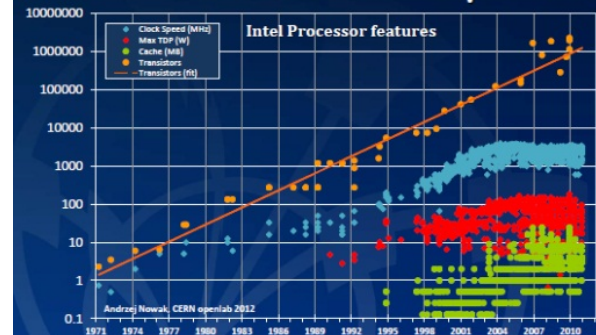
	SIMD	ILP	HW THREADS	CORES	SOCKETS
MAX	8	4	1.35	12	4
TYPICAL	6	1.57	1.25	10	2
HEP	1	0.80	1.25	8	2

	SIMD	ILP	HW THREADS	CORES	SOCKETS
MAX	8	32	43.2	518.4	2073.6
TYPICAL	6	9.43	11.79	117.86	235.71
HEP	1	0.8	1	8	16

Andrzej Nowak - The growth of commodity computing and HEP software - do they mix?

29

Hardware landscape



Andrzej Nowak - The growth of commodity computing and HEP software - do they mix?

4

Points for the immediate future

- > Can HEP re-investigate vectorization and thus leverage the current batch of accelerators?**
- > What will future hardware look like?**
- > Will accelerator and CPU languages and features converge?**
- > The online community is leading the efforts**

Q & A



Other questions? Andrzej.Nowak@cern.ch