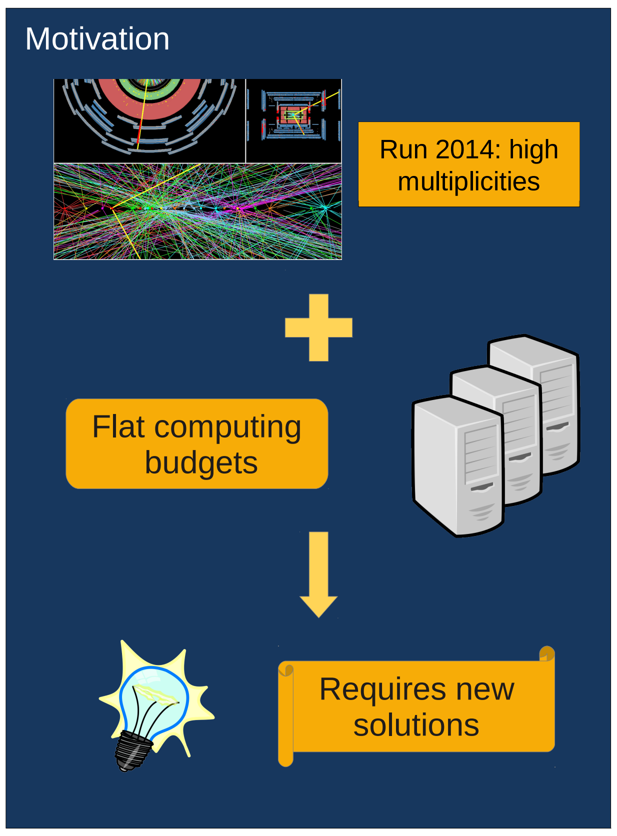
Computing in High Energy Physics 2013

Experience with Intel's Many Integrated Core (MIC / Xeon Phi) in ATLAS Software

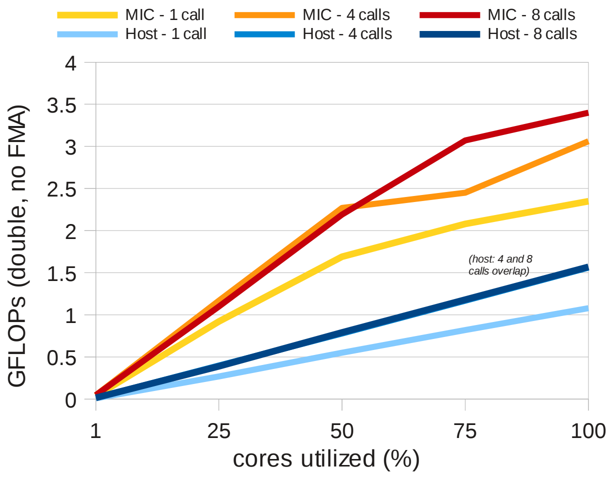
Motivation


Run 2014: high multiplicities

Flat computing budgets

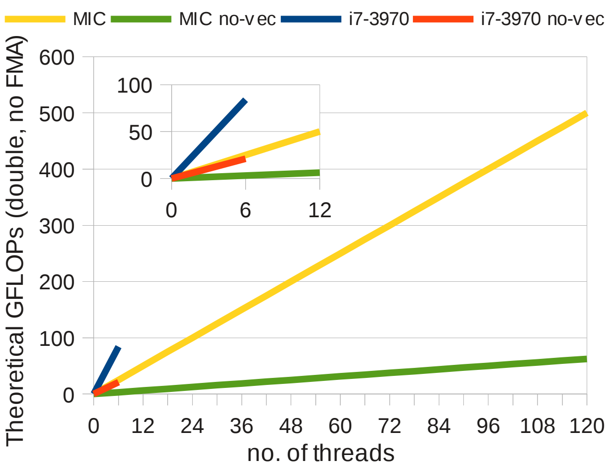
Requires new solutions

Object Orientation considered harmful



Virtual method calls into shared libraries prevent hardware optimizations: MIC wins given enough parallelism, because of bandwidth advantage. It scales super linearly with # calls. However, absolute performance for both is very low.

The Promise: $O(10^{12})$ ops/s in-a-box



Threads and vectorization

- 240+ threads per card
- 8-way vectorization for doubles
- *But*: bundled issue, no back-to-back
  - 2x #cores threads to saturate

Technology: compilers and tools

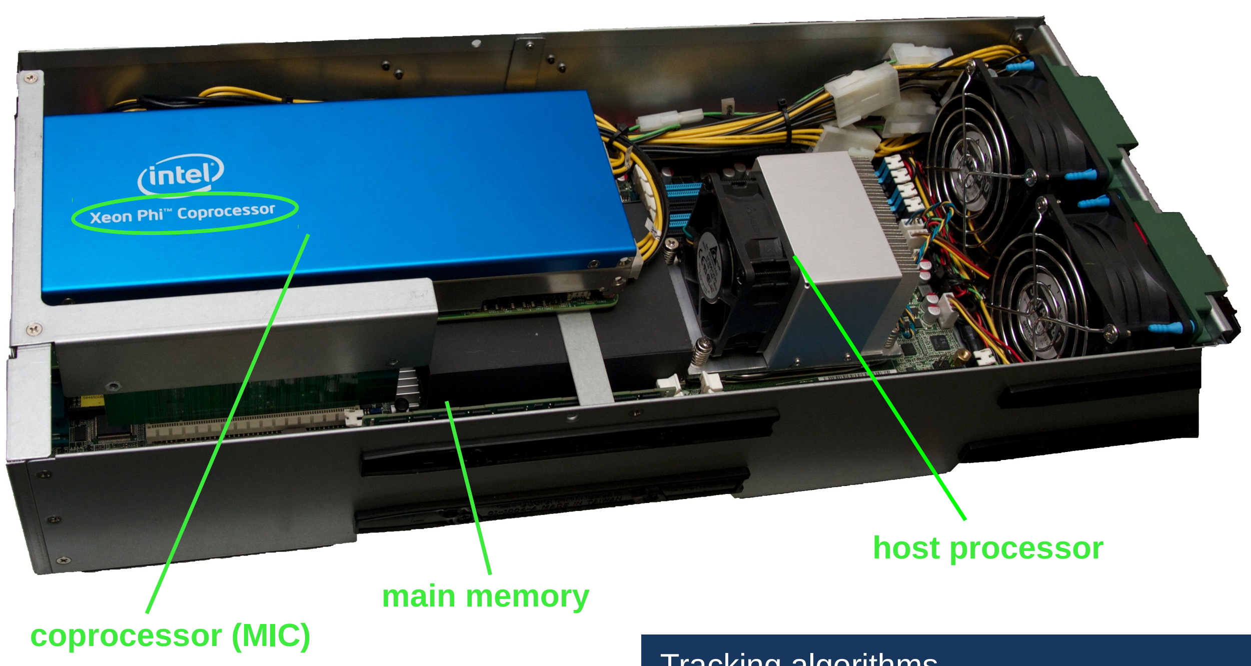
Technology	*Effective?*	*Xeon Phi support?*
Intel C/C++ Compiler (ICC)	★★	★★★★
GNU Compiler Collection (GCC)	★★	★★★
SIMD/Cilk Plus	★★★	★★★★
Threading Building Blocks	★★★★	★★★
Intel SPMD Program Compiler (ISPC)	★★★★	★
OpenMP (with ICC)	★★★	★★★★
OpenCL	★★	★★★

Currently available tools are *disappointing ...*

- Auto-vectorization is unpredictable and therefore unmaintainable
- Forced vectorization of aligned memory works, but is also hard to maintain
- Overall performance relatively weak outside specific use cases
- No support for automatic work/task sharing

Lots of choices, most tools are from Intel … which will be around long-term?

The Xeon Phi (*Many Integrated Core* or *MIC*) coprocessor



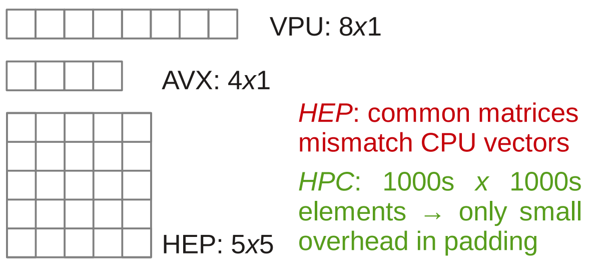
Coprocessor features:

- Intel Xeon Phi 5110P (B1 stepping)
- 60 x86 cores @ 1.05GHz
  - 4 hardware threads / core
- 8GB of RAM
- 512bit vector and masking registers
- 210GB/s effective memory bandwidth

Host features:

- Intel Xeon, E5-2603
  - 2 hyper threads / core
- 4 x86 cores @ 1.80GHz
- 32GB of RAM
- 256bit vector registers
- 34GB/s max memory bandwidth

Our codes are different



HEP: common matrices mismatch CPU vectors

HPC: 1000s x 1000s elements → only small overhead in padding

Tracking algorithms

Candidates for offloading:

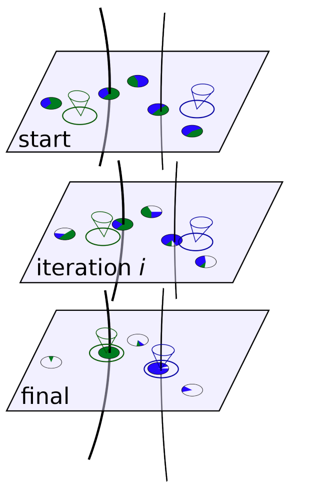
- Track finding in Pixel and SCT detectors: combinatorics allow parallelization
- Ambiguity processing after seed finding, new algorithm: *Multi Track Fitter* (MTF)
  - Hits assigned to tracks with probability weights, rather than exclusively
  - All tracks updated in parallel for each iteration

Engineering constraints:

- Access to geometry, material properties, and magnetic field
- Differences in lengths of individual tracks and dimensions of measurements

Graphics Processor Unit	Many Integrated Core
Effective contention control	Needs independent parallelization
Load a (simplified, due to limited memory) geometry and material description onto the card. Or, select a slice with a reference track on the host and only load the selection.	
Add "zero-measurements" to equalize track sizes.	
Measurements zero-padded for vectorization, but larger matrices result in more compute-intensive inversions.	

Multi Track Fitter

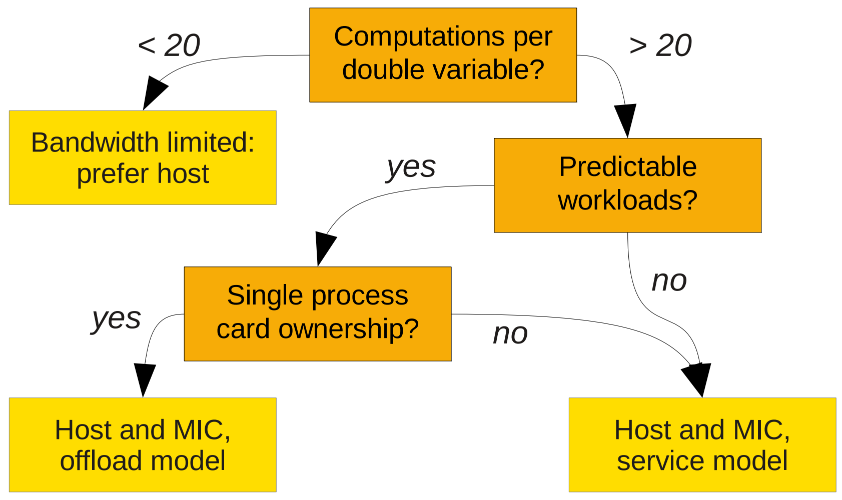


Framework Integration

The offload model is easiest to program

Task scheduling and coprocessor sharing requires a service programming model

The service model makes available all compute resources even when inefficient


Computations per double variable?

< 20

> 20

Bandwidth limited: prefer host

Predictable workloads?

yes

no

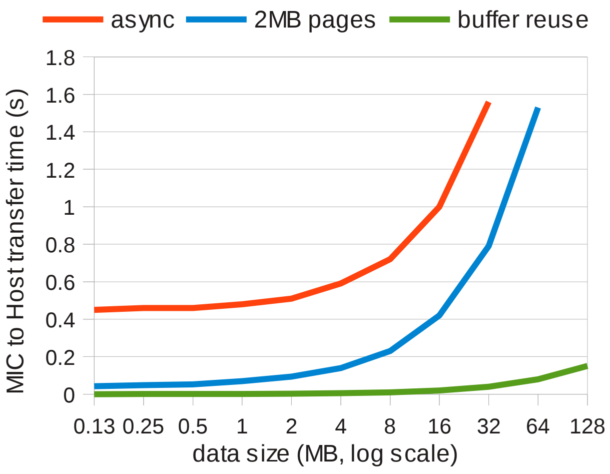
Single process card ownership?

yes

no

Host and MIC, offload model

Host and MIC, service model



Design for GaudiHive:

- Simple offload models block on host
- TBB tasks allow effective work balancing in "whole node" operation
- Control task scheduling through thread pool and task sizes

Engineering constraints:

- Offload data needs to be streamable
  - Allows conversion on-the-fly
- *Control process* for MIC access
  - Schedule resource allocation
  - Manage buffers for data transfers

Future

Next generation Many Integrated Core cards: Knights Landing (*)

- Both *standalone* and as coprocessor
- Support for AVX-512 (efficiently compatible with AVX and SSE)
- 14nm process, 2nd generation 3-D tri-gate
- Increased memory and memory bandwidth

** Source: http://newsroom.intel.com/community/intel_newsroom/blog/2013/06/17 /intel-powers-the-worlds-fastest-supercomputer-reveals-new-and-future-high-performance-computing-technologies*

MIC as a standalone CPU, i.e. directly on motherboard, would:

- Allow extension of memory banks?
- Remove programming complexities of data transfers

To be continued …

Fleischmann S., Kama S., Lavrijsen W., Neumann M., and Vitillo R. for the ATLAS Collaboration