Roofline plot for RICH pattern detection algorithm on Intels Knights Landing Platform

Christina Quast

tCSC 2017

June 9, 2017

Intel Xeon Phi Knights Landing

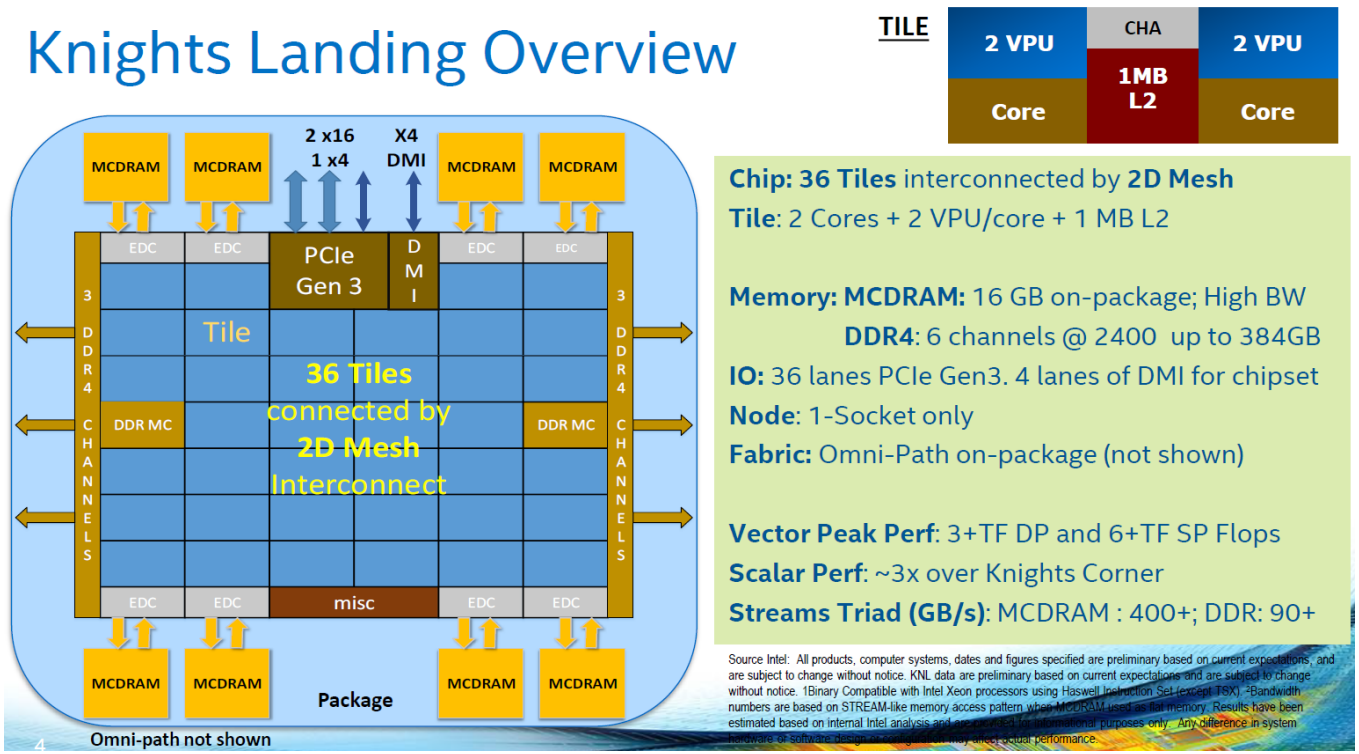

Figure: KNL [1]

[1] https://www.extremetech.com/wp-content/uploads/2016/04/KnightsLanding.png

Theoretical performance

- 40B input data + 12B output data = 52 B
- DRAM: $\approx 80 GBps$
- MCDRAM (High Bandwidth Memory on KNL): $\approx 340 GBps$

Theoreticaly best performance in time per photon:
$52.0 B / 340 GBps = 0.153$ ns

Memory layout: AOS to SOA

Arrange memory for better unit strides

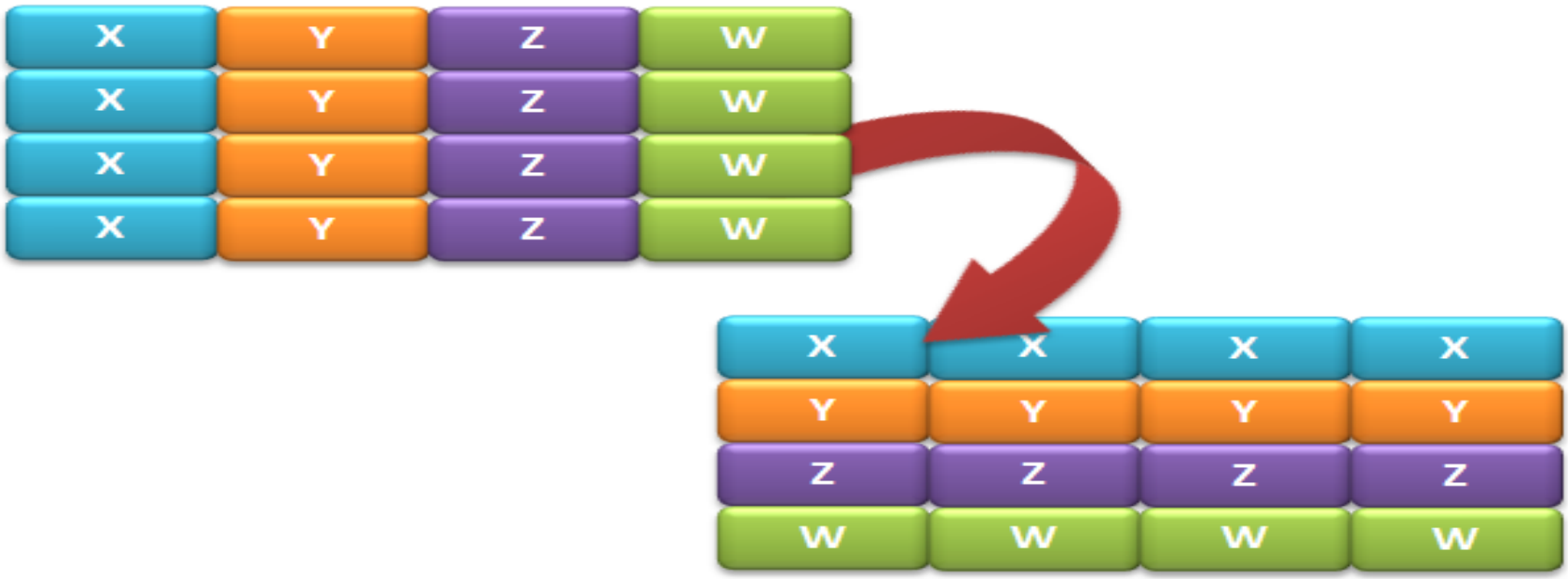

Figure: AOS to SOA [2]

[2] http://www.spuify.co.uk/?p=645

Memory and Cacheline optimizations

- Alignment of variables to 64 Byte boundaries (Cacheline size)
- Vectorization through vectorclass library (basically intrinsics abstraction)
- Const keyword helps compiler to optimize

Approximate functions

- Inverse approximate functions up to 10 times faster
- sqrt() replaced with approx_recipr (approx_rsqrt ())
- division replaced by approx_recip ()

Instruction	uops	reciprocal througput
VSQRT14PS	18	16
VRSQRT14PS	1	3
VDIVPS	18	32
VRCP28PS	1	3

Mathematical improvements

- Removed some divisions
- Extracted multiplication factors
- Term $\cos(\arcsin(\sin(\beta)))$ replaced by $\sqrt{(1-\sin^2(\beta))}$
- Removed cubic root (very expensive) and quartic solver, replaced with Newton

MCDRAM (340 GBps)

- numactl to bind execution to CPUs and MCDRAM Memory

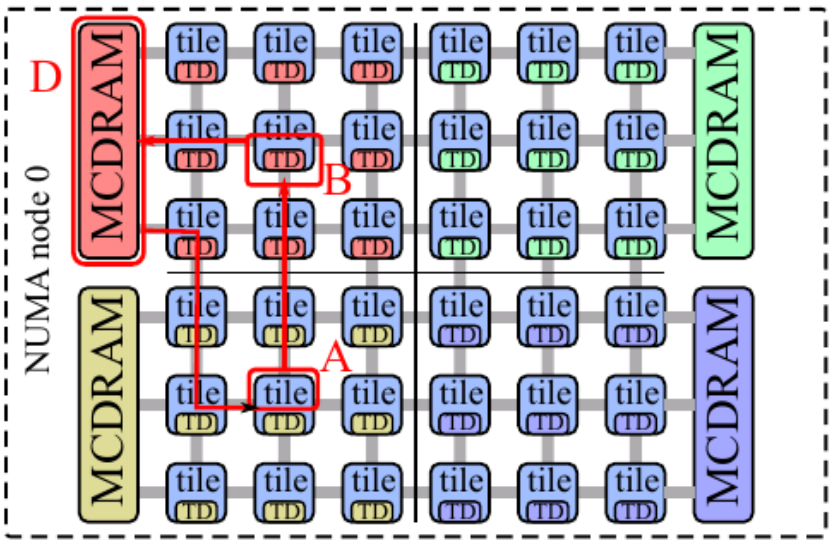

Figure: Quadrant Clustering mode [2]

Nanoseconds per photon

Theoretical limit: $52.0B/340GBps = 0.153$ ns per photon

Improvement	Execution time per photon	Speedup over baseline code with OMP
Baseline code without OMP	1000.26 ns	-
From here: always OpenMP 256 thread		
Baseline code	7.13 ns	-
Pinned on MCDRAM (with numactl)	6.63 ns	1.07x
Mathematical improvement	4.67 ns	1.53x
Vectorization and Memory alignment	0.933 ns	7.64x
All three	0.195 ns	36.47x

Roofline plot

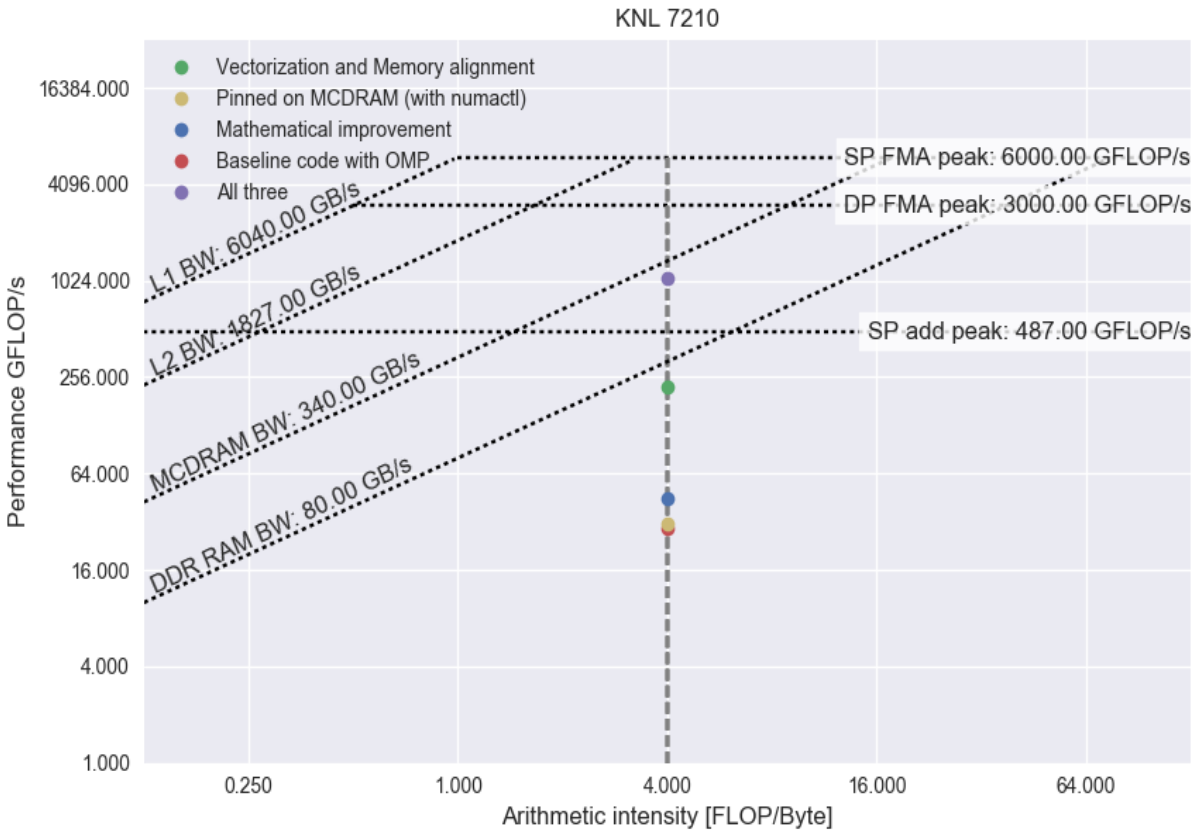

Figure: Roofline plot with mathematic improvements

Speedup and Efficiency

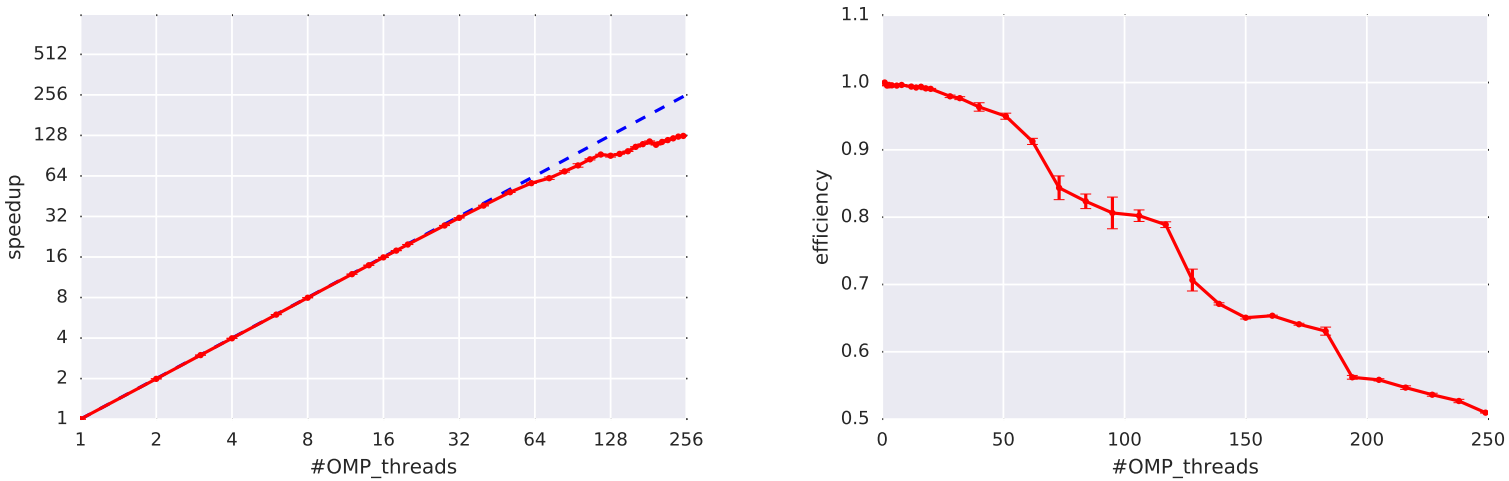

Figure: Strong scaling speedup (left) and efficiency (right) for 10485760 photons and OMP workgroup size of 128


R. Forty and O. Schneider.
Rich pattern recognition.
LHCB/98-040, 30 April 1998.


A. Vladimirov and R. Asai.
Clustering modes in knights landing processors: Developer's guide.
Colfax International, May 11, 2016.

Cherenkov angle

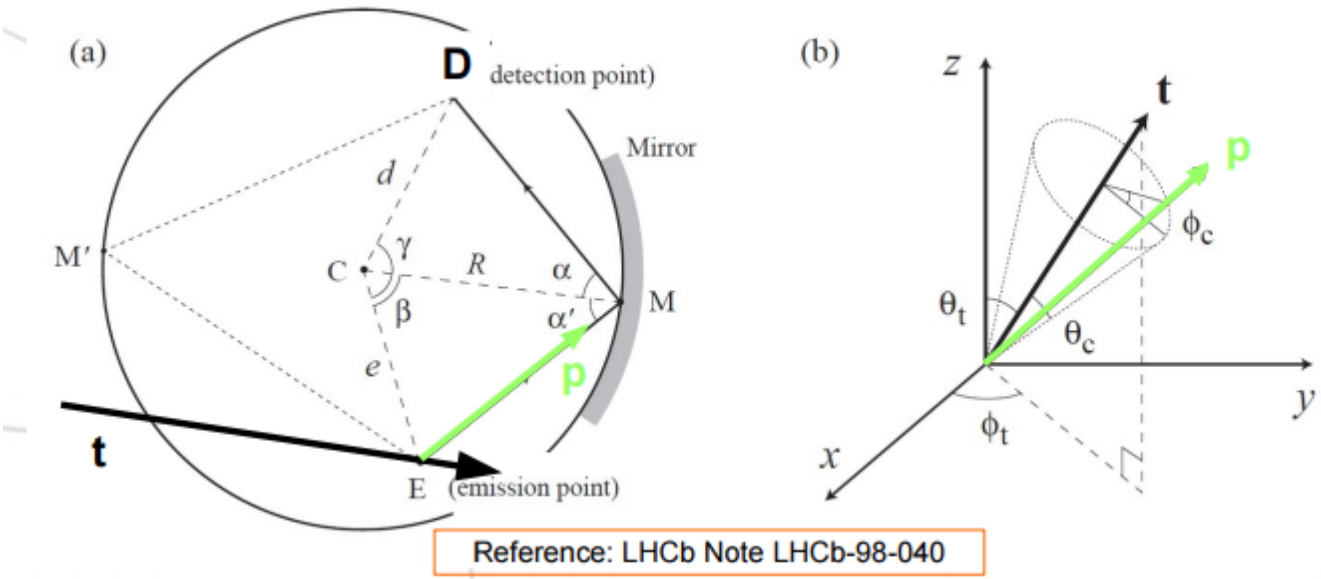

Figure: Cherenkov angle calculation[1]

Struct

```
template <typename T, std::size_t DIM = 16>
class PhotonReflection
{
    public:
        typedef typename XYZPoints<T, DIM>::vec_type vector;
    public:
        XYZPoints<T, DIM> emissPnt;
        XYZPoints<T, DIM> centOfCurv;
        XYZPoints<T, DIM> virtDetPoint;
        XYZPoints<T, DIM> sphReflPoint;
        std::array<T,DIM> radius;
};
```