Experiences Optimizing NWP for Intel's Xeon Phi Processor

John Michalakes

Naval Research Laboratory Marine Meteorology Division

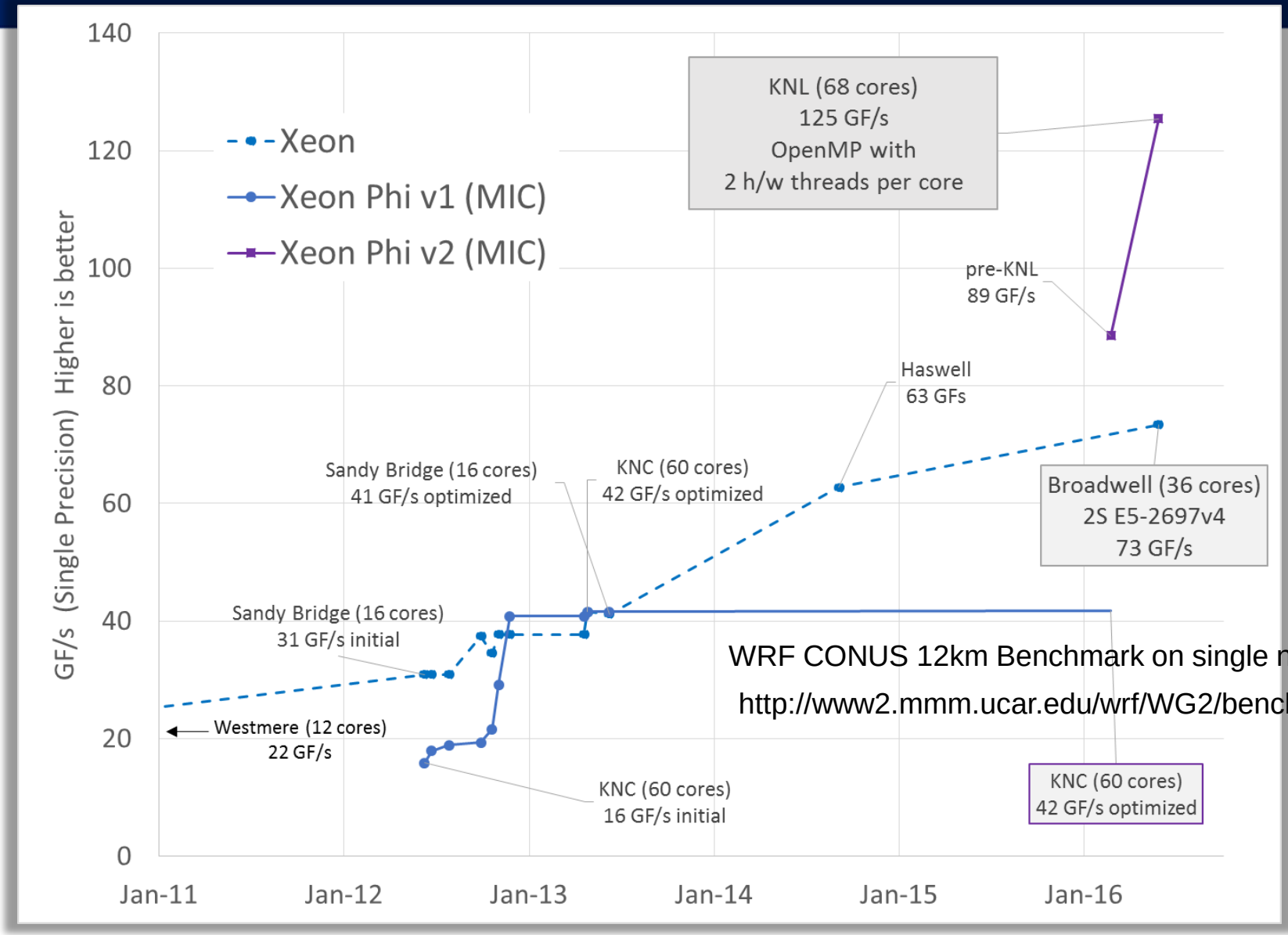
University Corporation for Atmospheric Research

Boulder, Colorado USA

17th Workshop on High Performance Computing in Meteorology

24 October 2016

WRF Performance Timeline

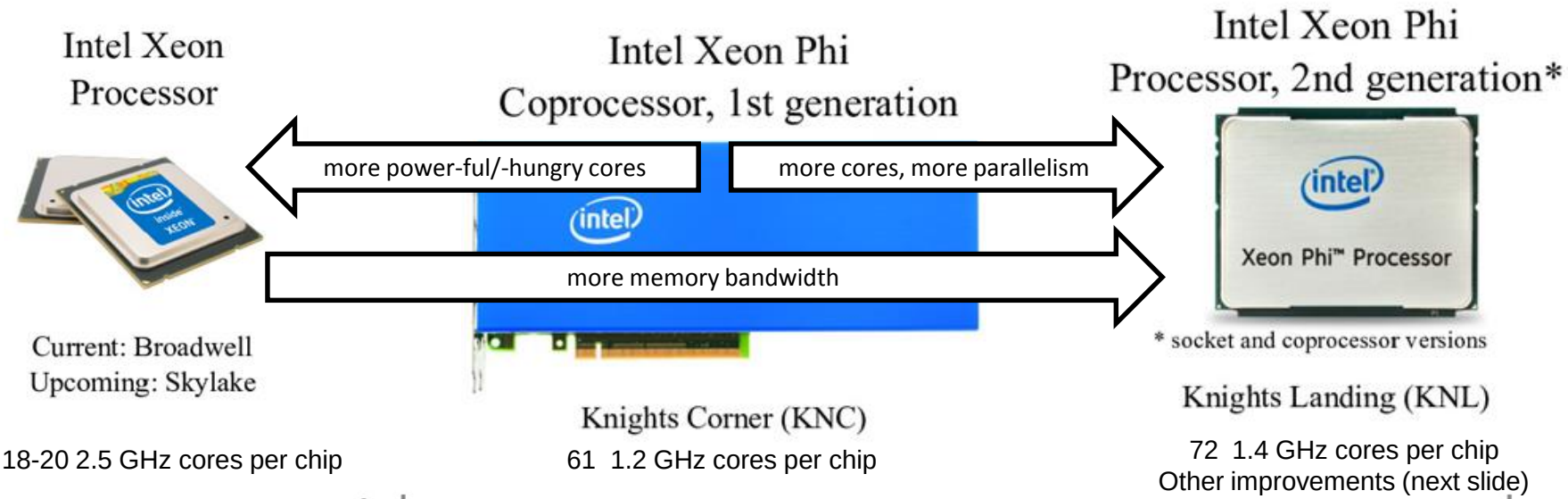


WRF results on KNL from Gokhale & Michalakes in:
Reinders, J. and J. Jeffers. Intel Xeon Phi Processor High Performance Programming, 2nd Edition Knights Landing Edition. Morgan Kaufman. 2016. ISBN: 9780128091944

Outline

- *A Brief History of Time*(-ings)
- Hardware and software
  - Knights Landing overview
  - Applications: NEPTUNE, WRF and kernels
- Optimizing for KNL
  - Roofline
  - Baselines and optimizations
- Other observations
- Summary

Hardware: Xeon Multi/Many-core Computing Platforms



Multi-Core Architecture

- Up to ~ 1 TFLOP/s in DP*
- Up to ~ 2 TFLOP/s in SP*
- Up to 3072 GiB DDR4 RAM*
- ~ 154 GB/s bandwidth*
- Hardware-rich: forgiving of sub-optimal code

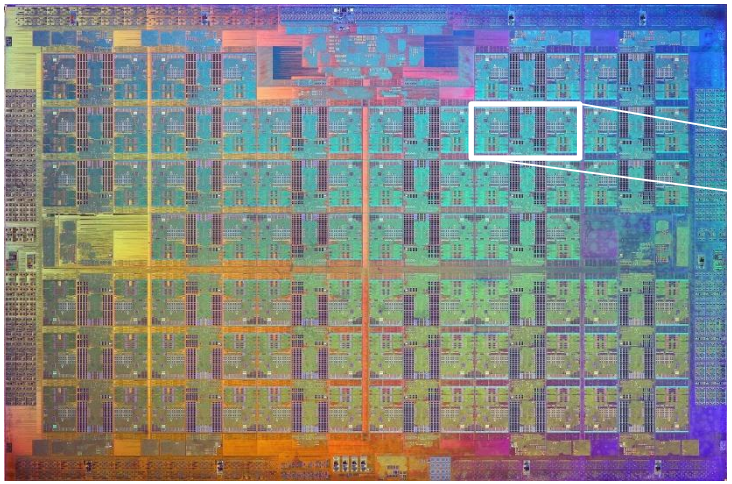
Intel Many Integrated Core (MIC) Architecture

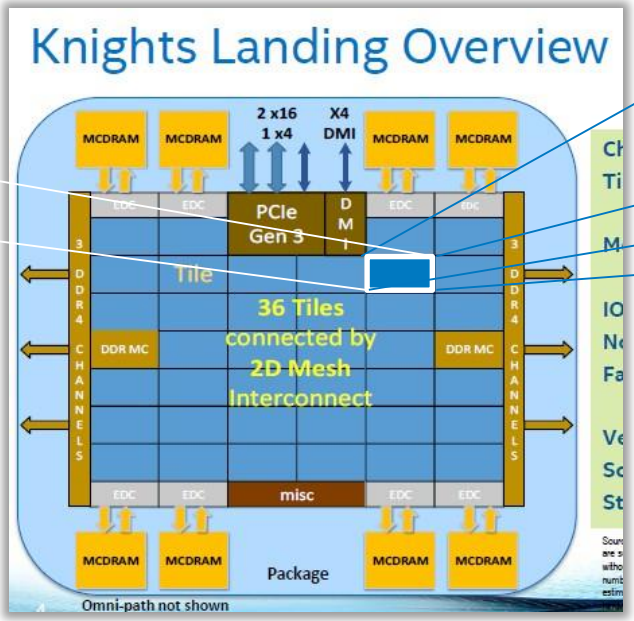
- PCIe add-in card
- ~ 1.2 TFLOP/s in DP
- ~ 2.4 TFLOP/s in SP
- Up to 16 GiB GDDR5 RAM
- ~ 176 GB/s bandwidth

- Socket version or coprocessor
- 3+ TFLOP/s in DP
- 6+ TFLOP/s in SP
- Up to 16 GiB MCDRAM
- ~ 400 GB/s MCDRAM bandwidth
- Up to 384 GiB DDR4 RAM
- ~ 90 GB/s DDR4 bandwidth
- Supports common OS

http://colfaxresearch.com/how-series-archive/

Hardware: Xeon Multi/Many-core Computing Platforms





- Intel Xeon Phi 7250 (Knights Landing) announced at ISC'16 in June
  - 14 nanometer feature size, > 8 billion transistors
  - 68 cores, 1.4 GHz modified "Silvermont" with out-of-order instruction execution
  - Two 512-bit wide Vector Processing Units per core
  - Peak ~3 TF/s double precision, ~6 TF/s single precision
  - 16 GB MCDRAM (on-chip) memory, > 400 GB/s bandwidth
  - "Hostless" – no separate host processor and no "offload" programming
  - Binary compatible ISA (instruction set architecture)

Models: NEPTUNE/NUMA

numa

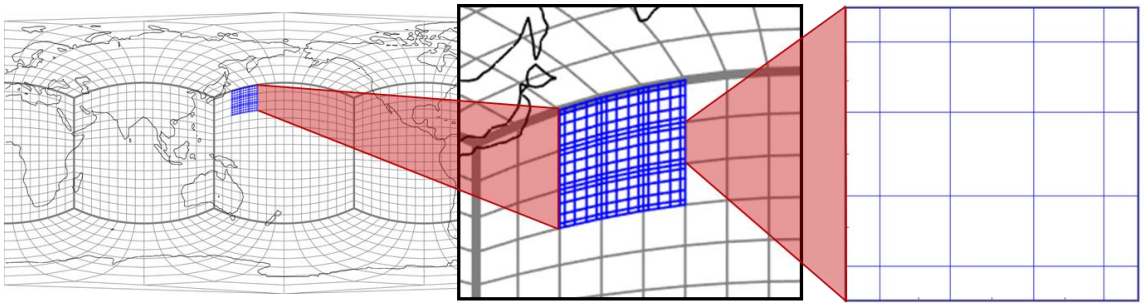
- **NUMA** = **N**on-hydrostatic **U**nified **M**odel of the **A**tmosphere
- engine inside the Navy's next generation weather prediction system NEPTUNE (**N**avy's **E**nvironment **P**rediction sys**T**em **U**sing the **N**uma **E**ngine)
- developed by **Prof. Francis X. Giraldo** and generations of postdocs

numa

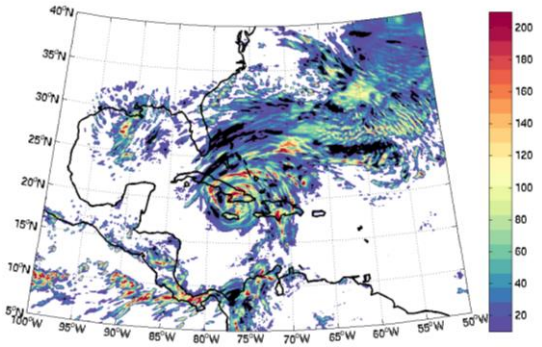
Andreas Mueller, NPS, Monterey, CA; and M. Kopera, S. Marras, and F. X. Giraldo. "Towards Operational Weather Prediction at 3.0km Global Resolution With the Dynamical Core NUMA". 96th Amer. Met. Society Annual Mtg. January, 2016. https://ams.confex.com/ams/96Annual/webprogram/Paper288883.html

Models: NEPTUNE/NUMA

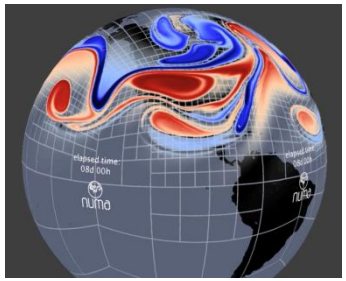
- Spectral element
  - 4th, 5th and higher-order* continuous Galerkin (discontinuous planned)
  - Cubed Sphere (also icosahedral)
  - Adaptive Mesh Refinement (AMR) in development



- Computationally dense but highly scalable
  - Constant width-one halo communication
  - Good locality for next generation HPC
  - Supports hybrid OpenMP/MPI (new!)



NEPTUNE 72-h forecast (5 km resolution) of accumulated precipitation for Hurr. Sandy



Example of Adaptive Grid tracking a severe event courtesy: Frank Giraldo, NPS

*"This is not the same 'order' as is used to identify the leading term of the error in finite-difference schemes, which in fact describes accuracy. Evaluation of Gaussian quadrature over N+1 LGL quadrature points will be exact to machine precision as long as the polynomial integrand is of the order 2x(N-1) -3, or less." Gabersek et al. MWR Apr 2012.DOI: 10.1175/MWR-D-11-00144.1

Optimizing for Intel Xeon Phi

- Most work in MIC programming involves *optimization* to achieve best share of peak performance
  - Parallelism:
    - KNL has up to 288 hardware threads (4 per core) and a total of more than 2000 floating point units on the chip
    - Exposing coarse, medium and especially fine-grain (vector) parallelism in application to efficiently use Xeon Phi
  - Locality:
    - Reorganizing and restructuring data structures and computation to improve data reuse in cache and reduce floating point units idle-time waiting for data: ***memory latency***
    - Kernels that do not have high data reuse or that do not fit in cache require high ***memory bandwidth***
- The combination of these factors affecting performance on the Xeon Phi (or any contemporary processor) can be characterized in terms of ***computational intensity***, and conceptualized using **The Roofline Model**

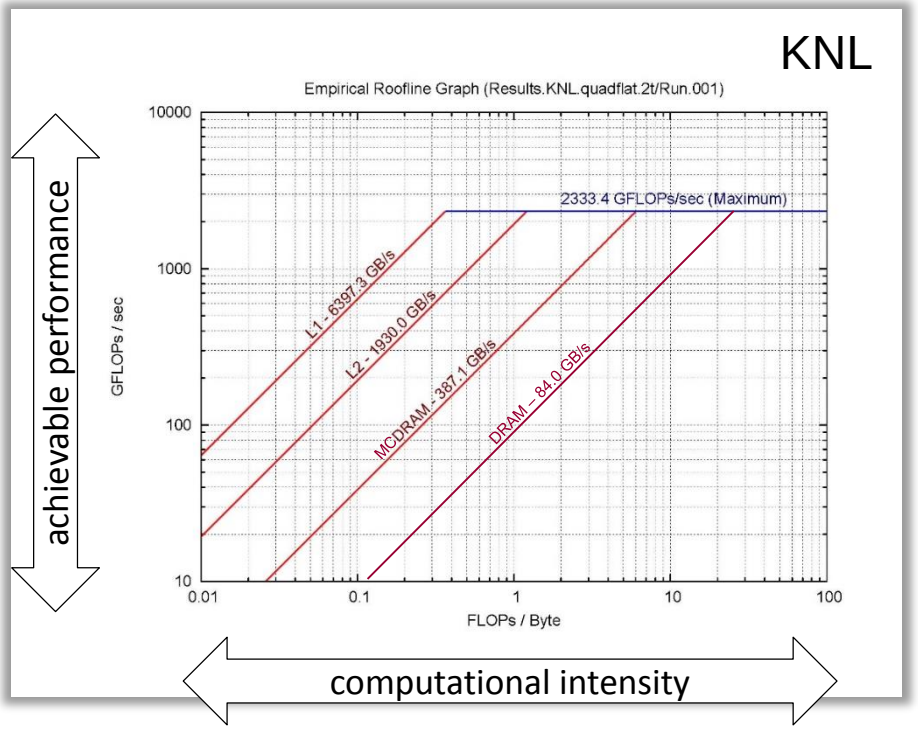
Optimizing for Intel Xeon Phi

- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity** – the number of floating point operations per byte moved between the processor and a level of the memory hierarchy.

Empirical Roofline Toolkit

https://crd.lbl.gov/departments/computer-science/PAR/research/roofline/

Thanks: Doug Doerfler, LBNL



*Sometimes referred to as:

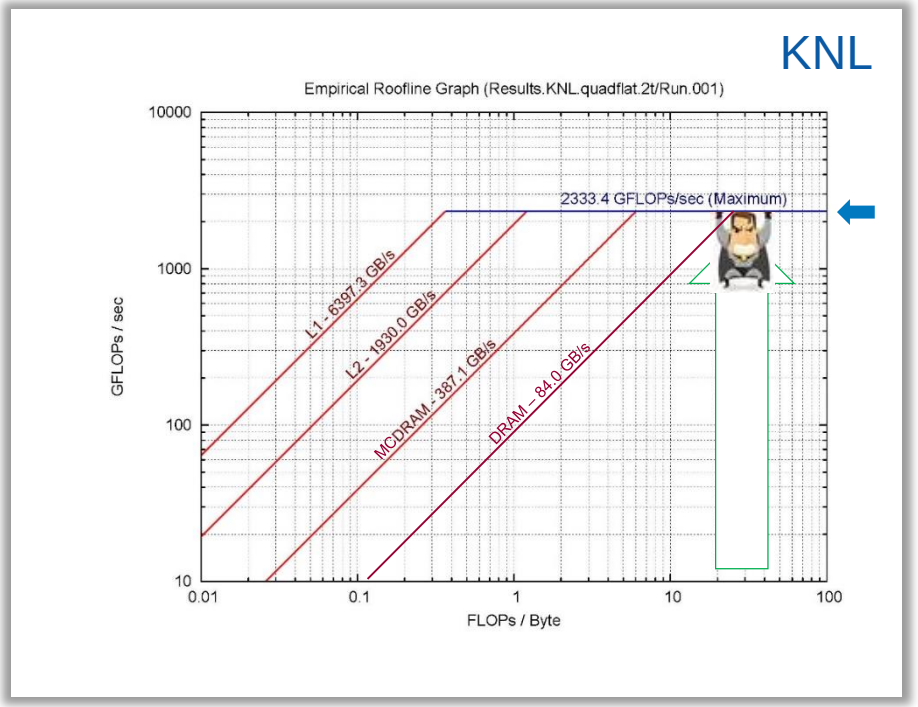
Arithmetic intensity (registers→L1): *largely algorithmic*

Operational intensity (LLC→DRAM): *improvable*

Williams, S. W., A. Waterman, D. Patterson.
Electrical Engineering and Computer Sciences, University of California at Berkeley
Technical Report No. UCB/EECS-2008-134. October 17, 2008

Optimizing for Intel Xeon Phi

- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity**
  - If intensity is high enough, application is "compute bound" by floating point capability
    - 2333 GFLOP/s double precision vector & fma

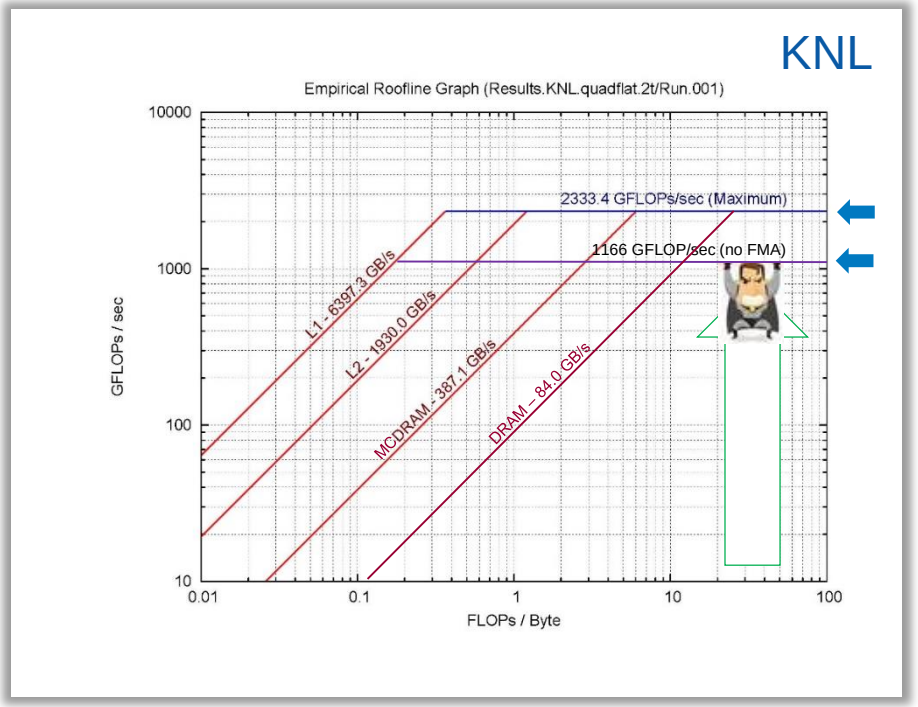


(Vectorization means the processor can perform 8 double precision operations at once as long as the operations are independent)

Williams, S. W., A. Waterman, D. Patterson.
Electrical Engineering and Computer Sciences, University of California at Berkeley
Technical Report No. UCB/EECS-2008-134. October 17, 2008

Optimizing for Intel Xeon Phi

- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity**
  - If intensity is high enough, application is "compute bound" by floating point capability
    - 2333 GFLOP/s double precision vector & FMA
    - 1166 GFLOP/s double precision vector, no FMA

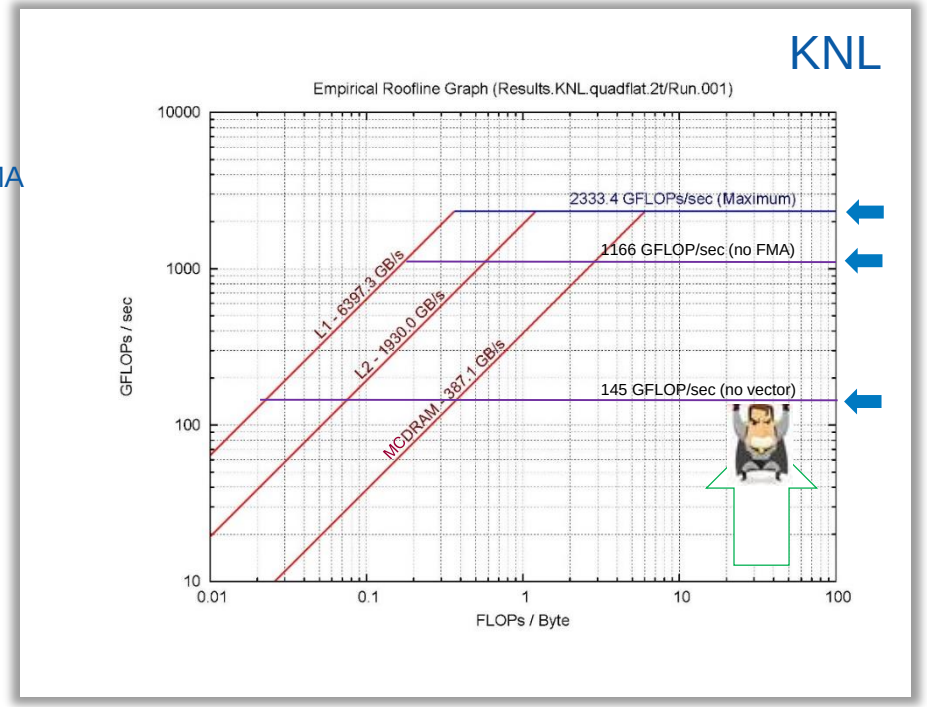


(FMA instructions perform a floating point multiply and a floating point addition as the same operation when the compiler can detect such expressions of the form: *result = A*B+C*)

Williams, S. W., A. Waterman, D. Patterson.
Electrical Engineering and Computer Sciences, University of California at Berkeley
Technical Report No. UCB/EECS-2008-134. October 17, 2008

Optimizing for Intel Xeon Phi

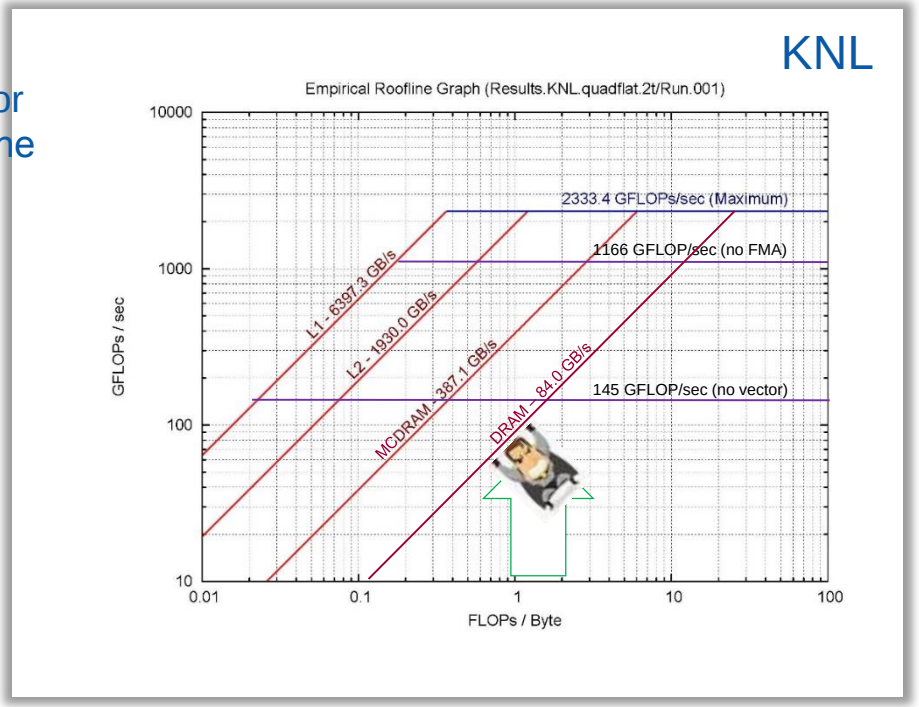
- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity**
  - If intensity is high enough, application is "compute bound" by floating point capability
    - 2333 GFLOP/s double precision vector & FMA
    - 1166 GFLOP/s double precision vector, no FMA
    - 145 GFLOP/s double precision no vector nor FMA



Williams, S. W., A. Waterman, D. Patterson.
Electrical Engineering and Computer Sciences, University of California at Berkeley
Technical Report No. UCB/EECS-2008-134. October 17, 2008

Optimizing for Intel Xeon Phi

- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity**
  - If intensity is high enough, application is “compute bound” by floating point capability
  - If intensity is not enough to satisfy demand for data by the processor’s floating point units, the application is“memory bound”
    - 128 GB main memory (DRAM)

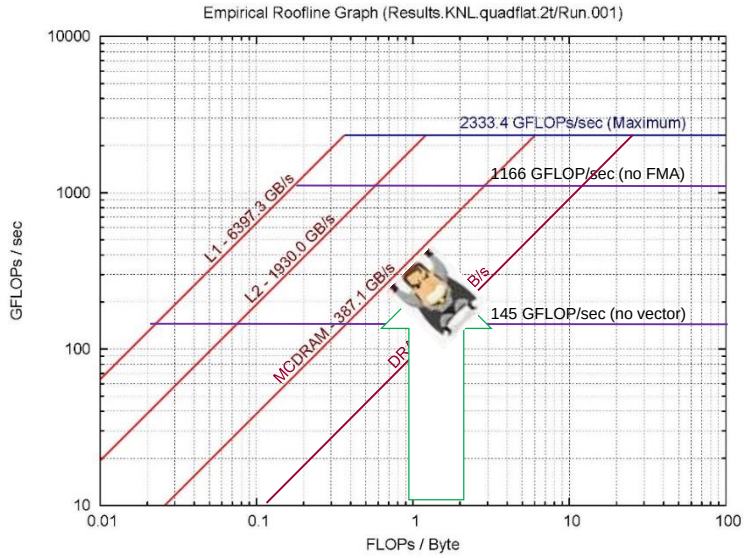


Williams, S. W., A. Waterman, D. Patterson.
Electrical Engineering and Computer Sciences, University of California at Berkeley
Technical Report No. UCB/EECS-2008-134. October 17, 2008

Optimizing for Intel Xeon Phi

- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity**
  - If intensity is high enough, application is “compute bound” by floating point capability
  - If intensity is not enough to satisfy demand for data by the processor’s floating point units, the application is“memory bound”
    - 128 GB main memory (DRAM)
    - 16 GB High Bandwidth memory (MCDRAM)

KNL



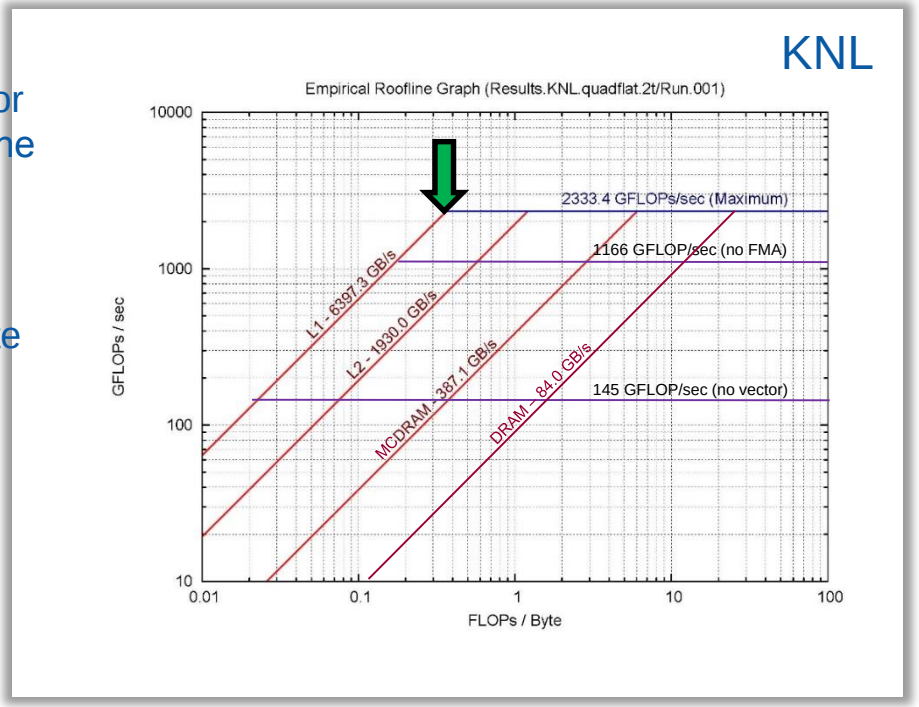
Among the ways to use MCDRAM:

- Configure KNL to use MCDRAM as direct mapped Cache
- Configure KNL to use MCDRAM as a NUMA region
  numactl –membind=0 ./executable (main memory)
  numactl –membind=1 ./executable (MCDRAM)
- More info: http://colfaxresearch.com/knl-mcdram

Williams, S. W., A. Waterman, D. Patterson.
Electrical Engineering and Computer Sciences, University of California at Berkeley
Technical Report No. UCB/EECS-2008-134. October 17, 2008

Optimizing for Intel Xeon Phi

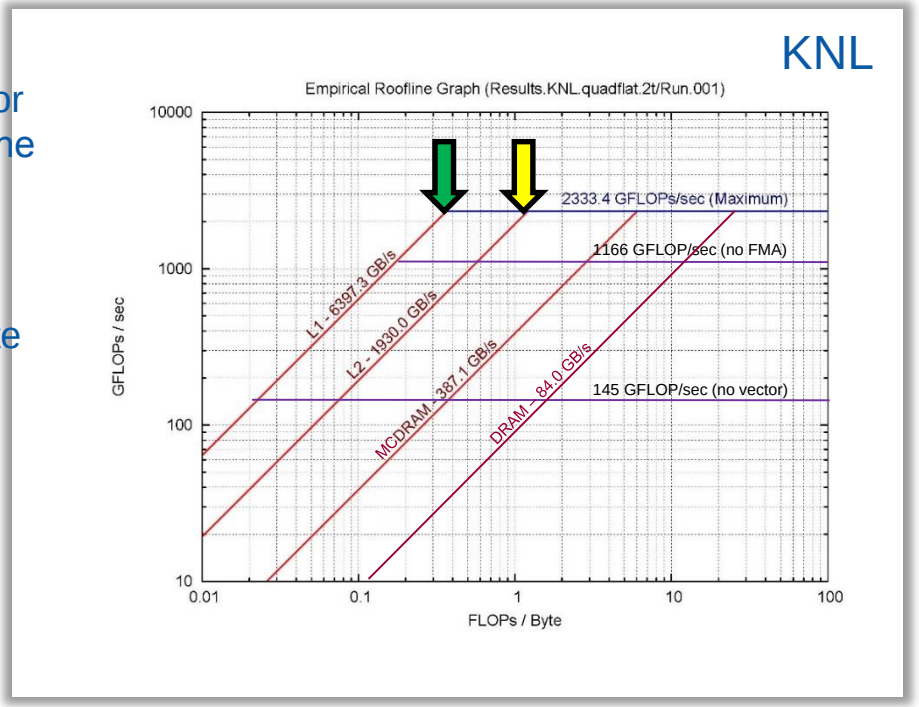
- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity**
  - If intensity is high enough, application is “compute bound” by floating point capability
  - If intensity is not enough to satisfy demand for data by the processor’s floating point units, the application is“memory bound”
    - 128 GB main memory (DRAM)
    - 16 GB High Bandwidth memory (MCDRAM)
  - KNL is nominally 3 TFLOP/sec but to saturate full floating point capability, need:
    - 0.35 flops per byte from L1 cache



Williams, S. W., A. Waterman, D. Patterson.
Electrical Engineering and Computer Sciences, University of California at Berkeley
Technical Report No. UCB/EECS-2008-134. October 17, 2008

Optimizing for Intel Xeon Phi

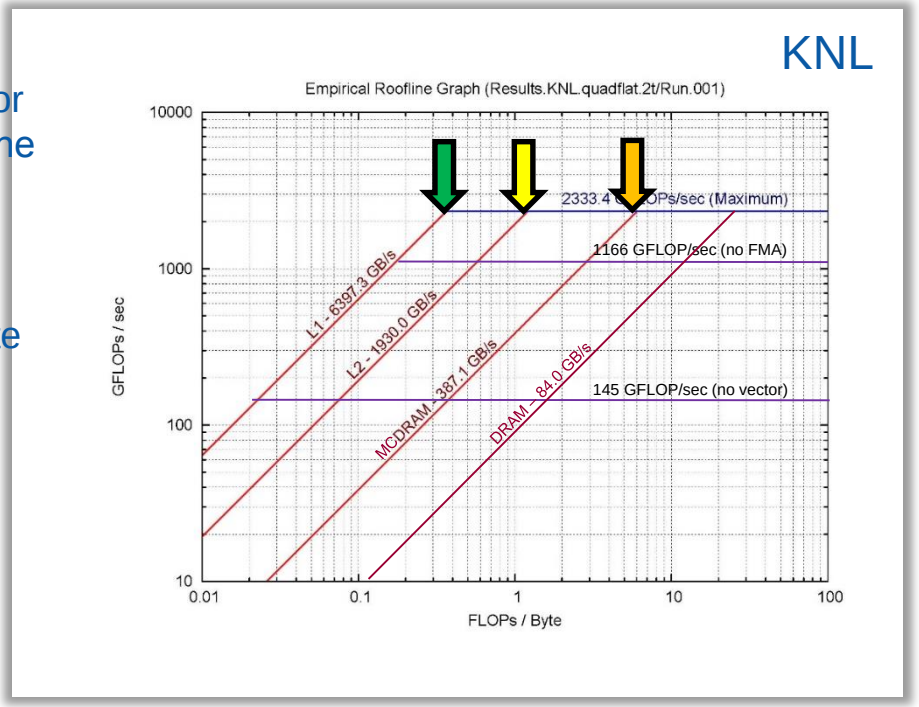
- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity**
  - If intensity is high enough, application is “compute bound” by floating point capability
  - If intensity is not enough to satisfy demand for data by the processor’s floating point units, the application is“memory bound”
    - 128 GB main memory (DRAM)
    - 16 GB High Bandwidth memory (MCDRAM)
  - KNL is nominally 3 TFLOP/sec but to saturate full floating point capability, need:
    - 0.35 flops per byte from L1 cache
    - 1 flop per byte from L2 cache



Williams, S. W., A. Waterman, D. Patterson.
Electrical Engineering and Computer Sciences, University of California at Berkeley
Technical Report No. UCB/EECS-2008-134. October 17, 2008

Optimizing for Intel Xeon Phi

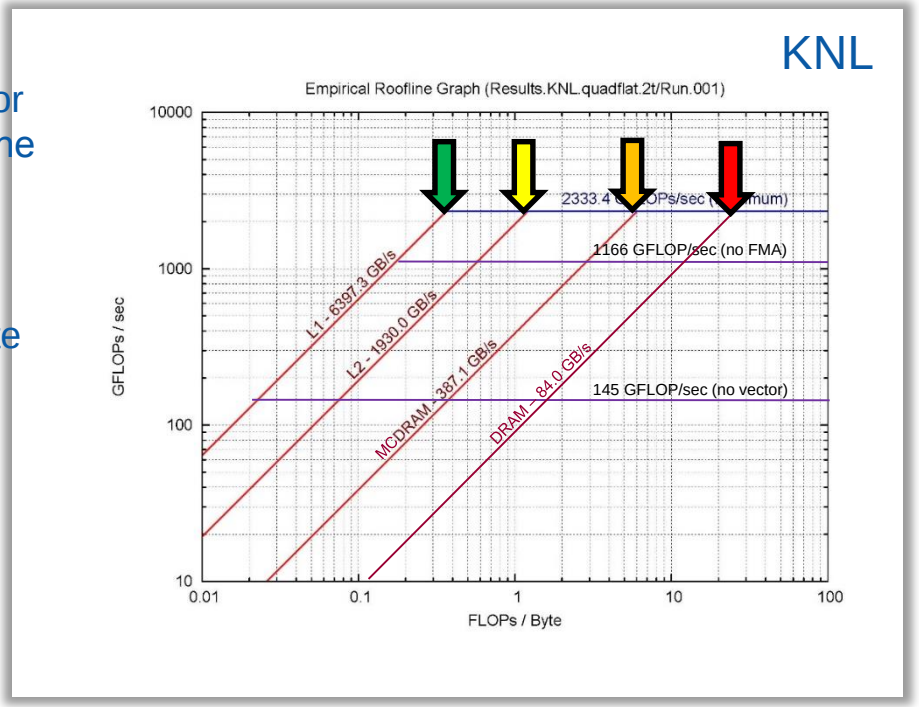
- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity**
  - If intensity is high enough, application is “compute bound” by floating point capability
  - If intensity is not enough to satisfy demand for data by the processor’s floating point units, the application is“memory bound”
    - 128 GB main memory (DRAM)
    - 16 GB High Bandwidth memory (MCDRAM)
  - KNL is nominally 3 TFLOP/sec but to saturate full floating point capability, need:
    - 0.35 flops per byte from L1 cache
    - 1 flop per byte from L2 cache
    - 6 flops per byte from high bandwidth memory



Williams, S. W., A. Waterman, D. Patterson.
Electrical Engineering and Computer Sciences, University of California at Berkeley
Technical Report No. UCB/EECS-2008-134. October 17, 2008

Optimizing for Intel Xeon Phi

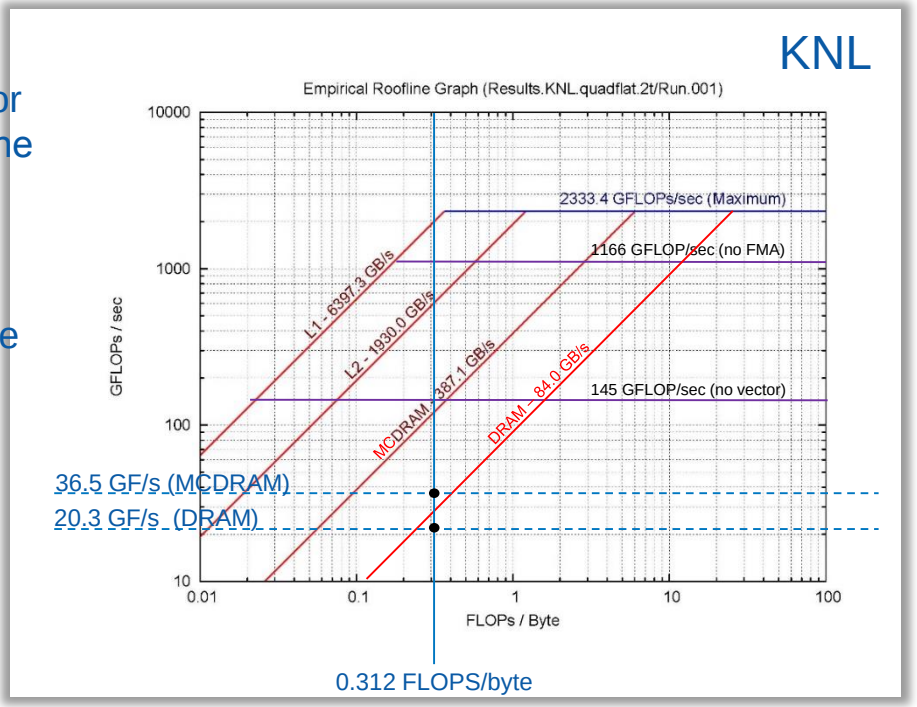
- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity**
  - If intensity is high enough, application is “compute bound” by floating point capability
  - If intensity is not enough to satisfy demand for data by the processor’s floating point units, the application is“memory bound”
    - 128 GB main memory (DRAM)
    - 16 GB High Bandwidth memory (MCDRAM)
  - KNL is nominally 3 TFLOP/sec but to saturate full floating point capability, need:
    - 0.35 flops per byte from L1 cache
    - 1 flop per byte from L2 cache
    - 6 flops per byte from high bandwidth memory
    - 25 flops per byte from main memory



Williams, S. W., A. Waterman, D. Patterson.
Electrical Engineering and Computer Sciences, University of California at Berkeley
Technical Report No. UCB/EECS-2008-134. October 17, 2008

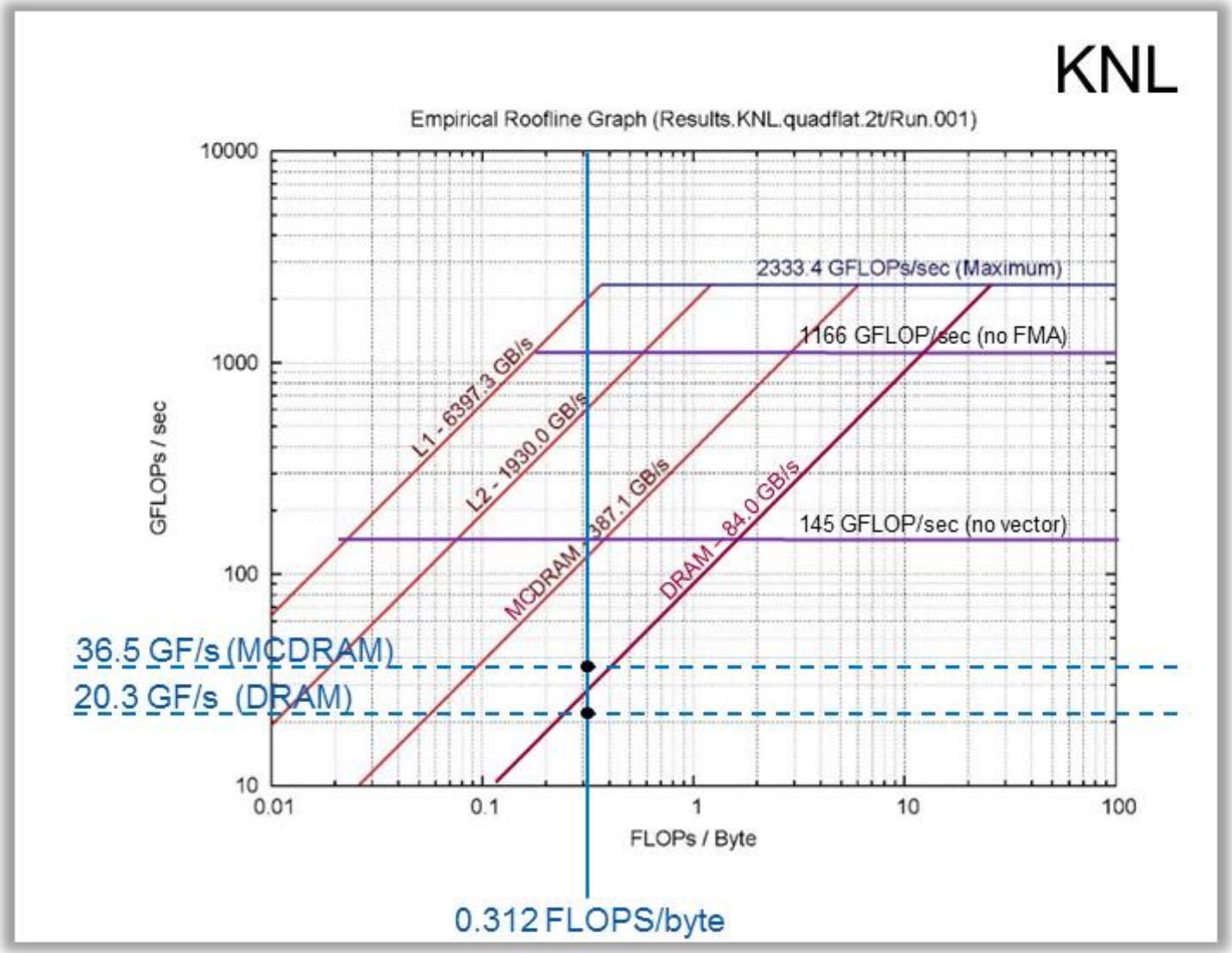
Optimizing for Intel Xeon Phi

- Roofline Model of Processor Performance
  - Bounds application performance as a function of **computational intensity**
  - If intensity is high enough, application is “compute bound” by floating point capability
  - If intensity is not enough to satisfy demand for data by the processor’s floating point units, the application is“memory bound”
    - 128 GB main memory (DRAM)
    - 16 GB High Bandwidth memory (MCDRAM)
  - KNL is nominally 3 TFLOP/sec but to saturate full floating point capability, need:
    - 0.35 flops per byte from L1 cache
    - 1 flop per byte from L2 cache
    - 6 flops per byte from high bandwidth memory
    - 25 flops per byte from main memory
  - Hard to come by in real applications!
    - NEPTUNE benefits from MCDRAM (breaks through the DRAM ceiling) but realizing only a fraction of the MCDRAM ceiling



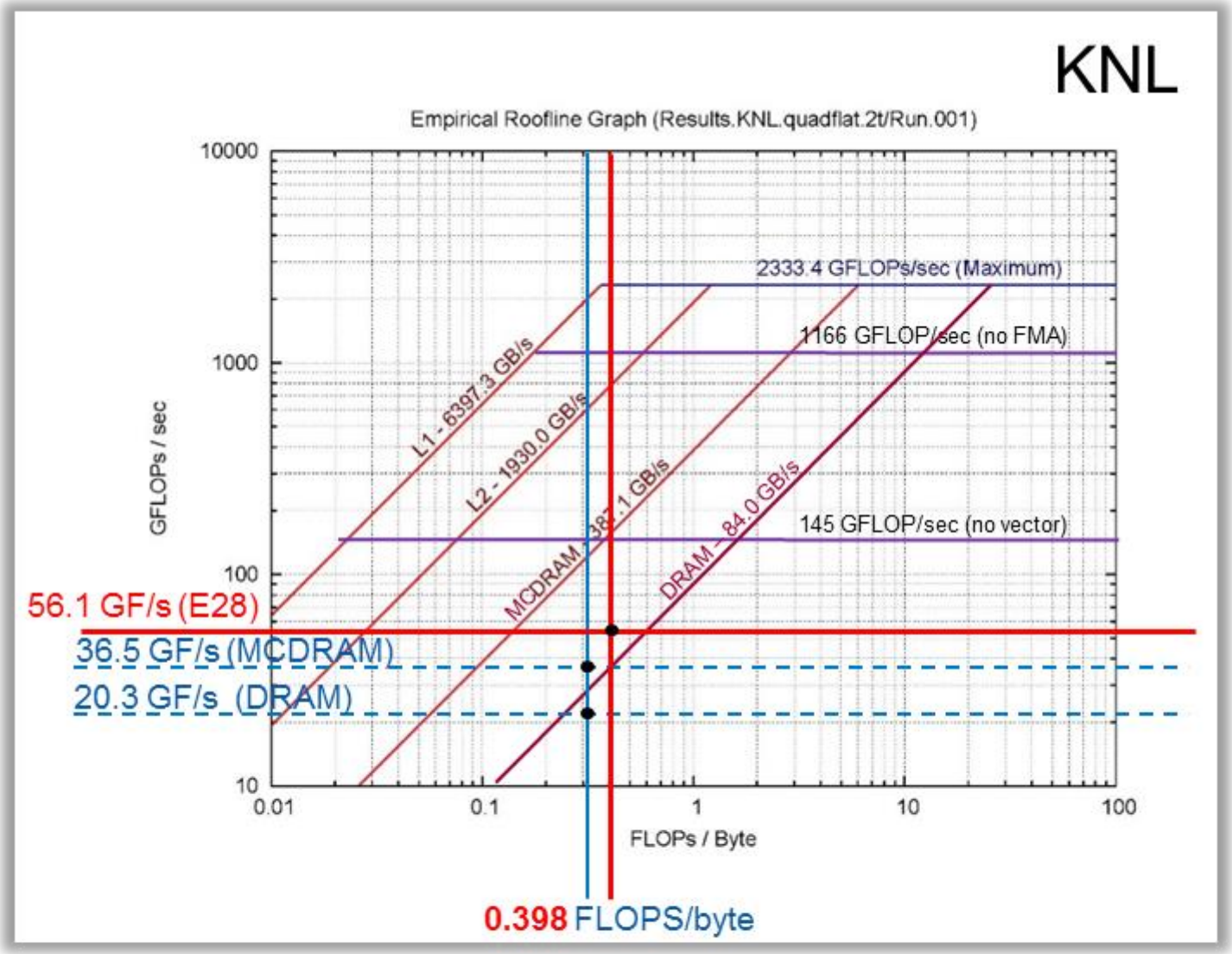
NEPTUNE E14P3L40
(U.S. Navy Global Model Prototype)

Optimizing for Intel Xeon Phi



NEPTUNE E14P3L40
(U.S. Navy Global Model Prototype)

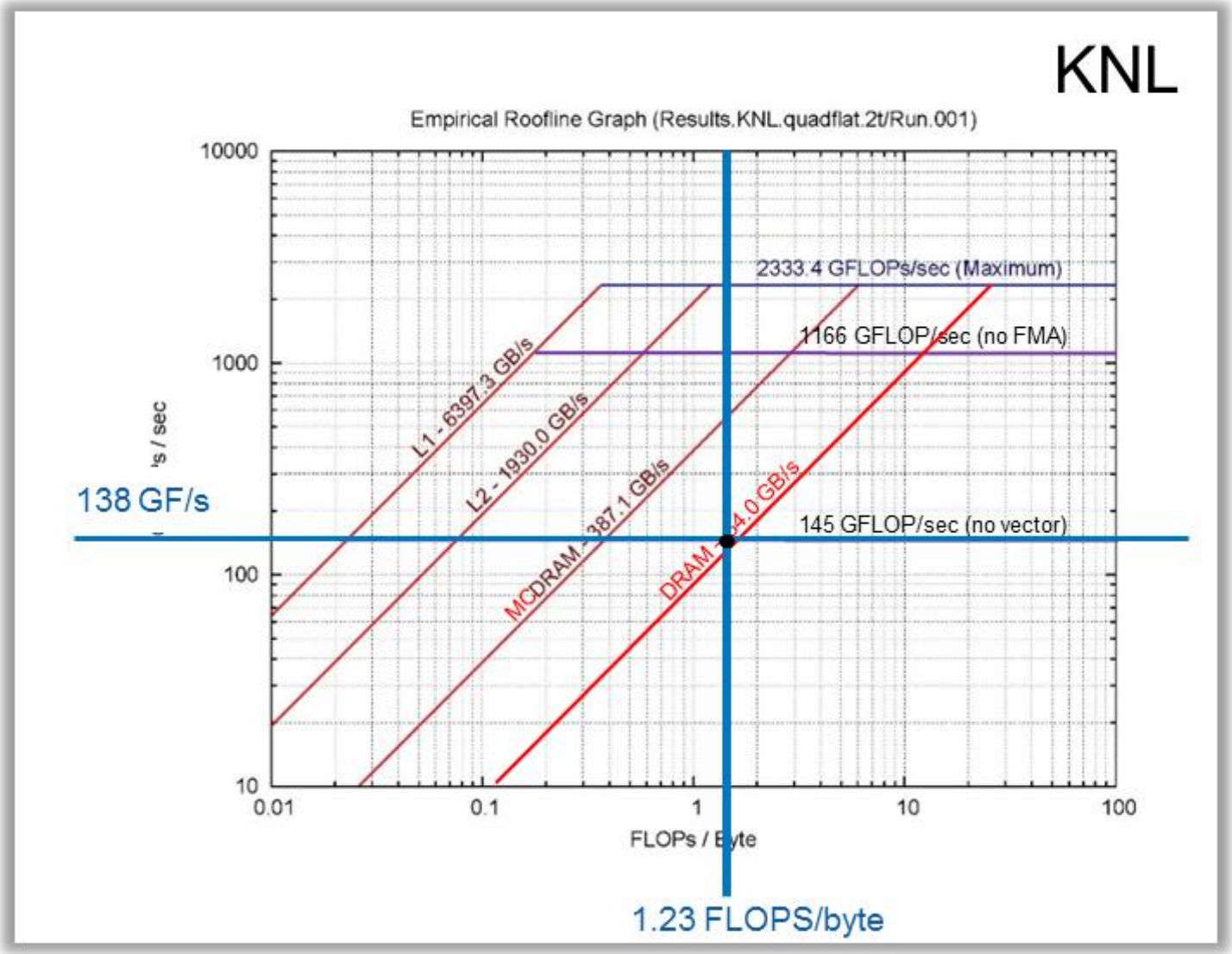
Optimizing for Intel Xeon Phi



NEPTUNE **E28**P3L40

(U.S. Navy Global Model Prototype)

Optimizing for Intel Xeon Phi

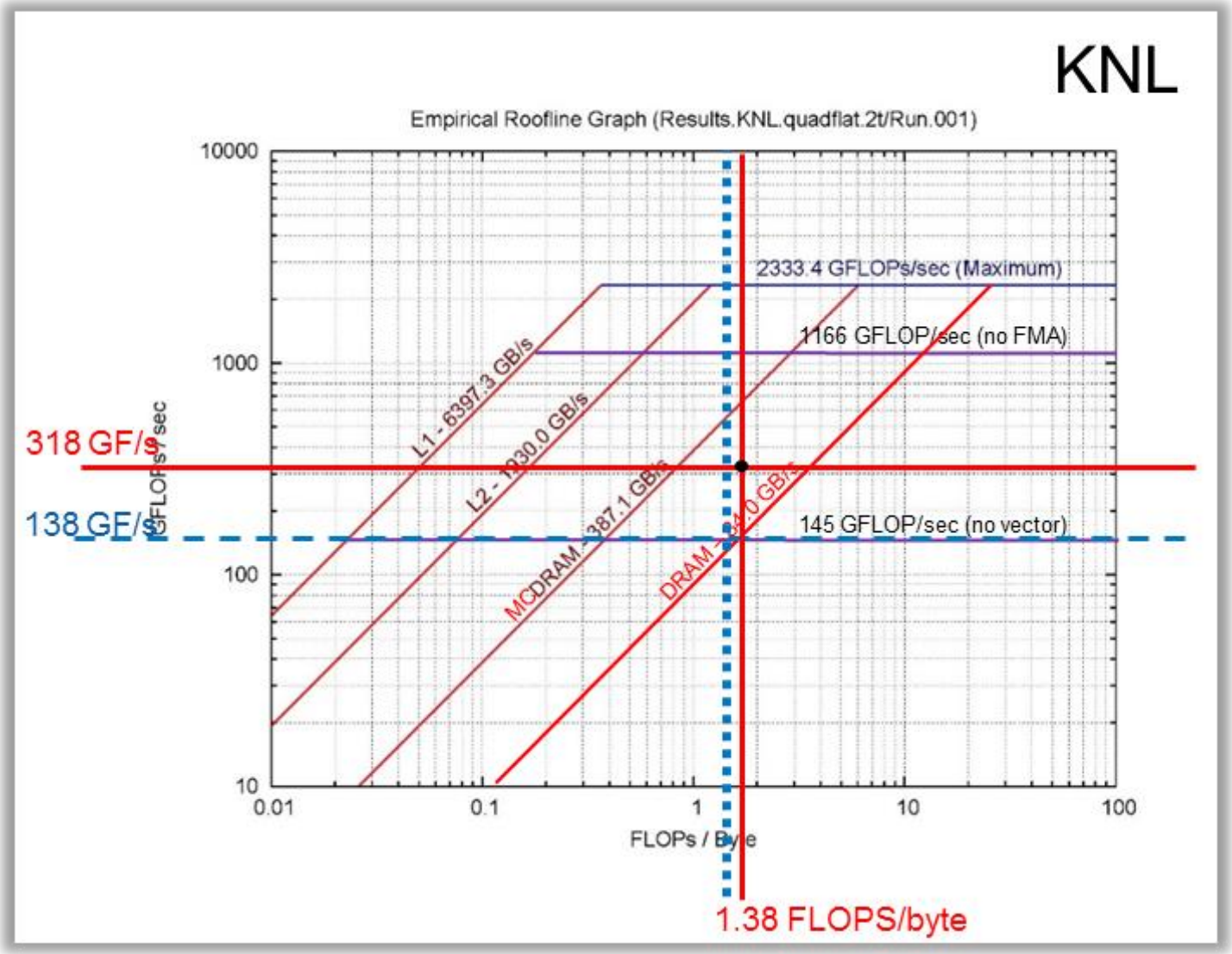


NEPTUNE diffusion
(create_laplacian)

- Optimizations
  - Compile time specification of important loop ranges
    - Number of variables per grid point
    - Dimensions of an element
    - Flattening nested loops to collapse for vectorization
  - Facilitating inlining
    - Subroutines called from element loops in CREATE_LAPLACIAN and CREATE_RHS
    - Matrix multiplies (need to try MKL DGEMM)
  - Splitting apart loops where part vectorizes and part doesn't
  - Fixing unaligned data (still a benefit on KNL)

Create_ laplacian	CPU Time	L2 Hit Rate	L2 Hit Bound	L2 Miss Bound	L2 Miss Count	CPI Rate	Instructions
Orig.	963.645	0.80597	0.0286706	0.0933823	585,017,550	1.38354	9.9862E+11
New	422.305	0.81075	0.0584720	0.184662	507,015,210	2.13113	2.8283E+11

Optimizing for Intel Xeon Phi



NEPTUNE diffusion
(create_laplacian)

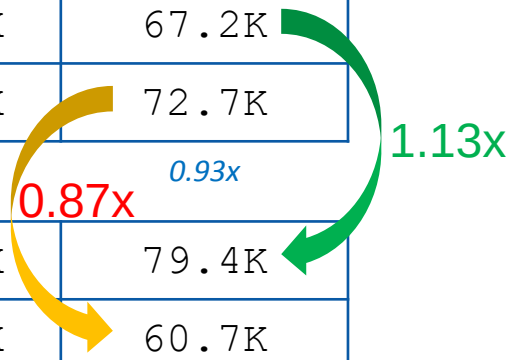
NEPTUNE Results

Time in seconds for 30 time steps Lower is better		create_laplacian			Whole code		
		Original	Optimized	*Benefit:*	Original	With optimized create_laplacian	*Benefit:*
E14 (15,288 elements)	KNL (68c)	1.53	1.20	*1.28x*	7.45	6.66	*1.09x*
	BDW (72c 2S)	1.43	0.72	*1.99x*	6.98	6.31	*1.11x*
	Advantage KNL→	*0.93x*	*0.60x*		*0.94x*	*0.93x*	
E28 (61,152 elements)	E28 (KNL)	5.92	2.65	*2.23x*	26.4	23.5	*1.15x*
	E28 (2S BDW)	6.07	3.61	*1.68x*	32.2	30.2	*1.07x*
	Advantage KNL→	*1.03x*	*1.36x*		*1.22x*	*1.31x*	

- Benchmark comparisons for two “supercell” workloads: E28 workload is 4x E14
- Original NEPTUNE code and with optimizations to most expensive diffusion routine
- Hardware:
  - KNL: Knights Landing 7250 (B0), 68c, 1.4 GHz
  - BDW: E5-2697v4, 2.3 GHz, 18c x 2 sockets

NEPTUNE Results

Units are element-steps per second. Higher is better.		Simulation Rate	
		Original	With optimized create_laplacian
E14 (15,288 elements)	KNL (68c)	61.5K	67.2K
	BDW (72c 2S)	65.7K	72.7K
	Advantage KNL→	*0.94x*	*0.93x*
E28 (61,152 elements)	E28 (KNL)	69.2K	79.4K
	E28 (2S BDW)	56.9K	60.7K
	Advantage KNL→	*1.22x*	*1.31x*



- KNL efficiency increases with larger workload; Broadwell declines.

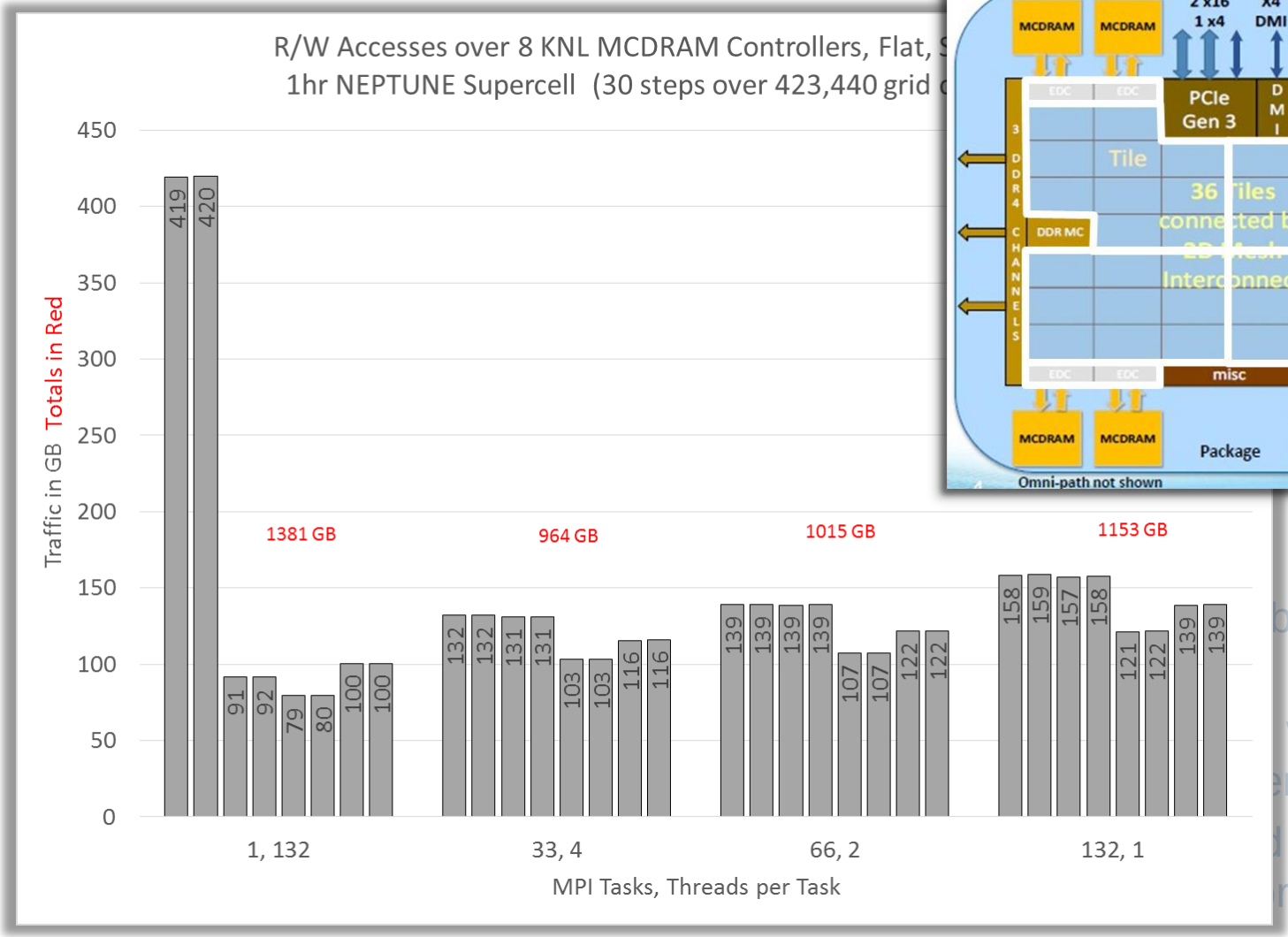
Realizing performance potential?

- Whole code
  - NEPTUNE: 56 GF/s (< 2 percent D.P. peak fp)
  - WRF: 125 GF/s (~2 percent S.P. peak fp)
- Discouraged? Remember:
  - NWP is multiphasic with mix of different roofline characteristics
    - Optimized create_laplacian (NEPTUNE diffusion) is reaching 318 GF/s (> 10 percent D.P. peak fp)
    - Optimized WSM5 (WRF microphysics) is reaching 447 GF/s (7.5 percent of S.P. peak fp)
  - Relatively flat profile – need to pay specific attention to large swath of code

Other observations: hybrid parallelism

- NEPTUNE on KNL
  - Best performance with **straight MPI**, 2 ranks per core
- WRF on KNL (CONUS 12km benchmark)
  - Best performance with **all OpenMP** threads, 2 per core
- Why the difference?
  - Both WRF and KNL use high-level SPMD threads
  - But NEPTUNE does many hundreds of OMP BARRIER synchronizations per time step
    - Needed to prevent race conditions on node points shared between neighboring elements computed on different threads
    - Andreas Mueller's work on Blue Gene/Q showed OpenMP was a win in spite of OMP BARRIER -- BG/P barriers more efficient?
    - "CGd" organization (used in HOMME) will eliminate shared nodes between neighboring elements, at cost of additional memory
- Be careful of "first touch" in Sub-NUMA clustering mode

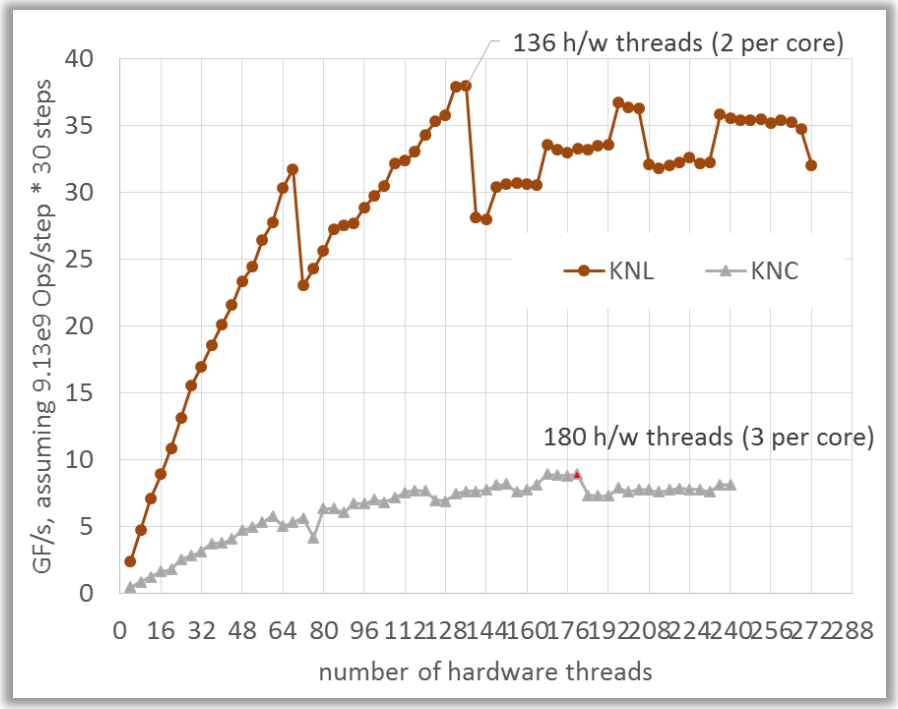
Other observations: NUMA modes



- Be careful of “first touch” in Sub-NUMA clustering mode

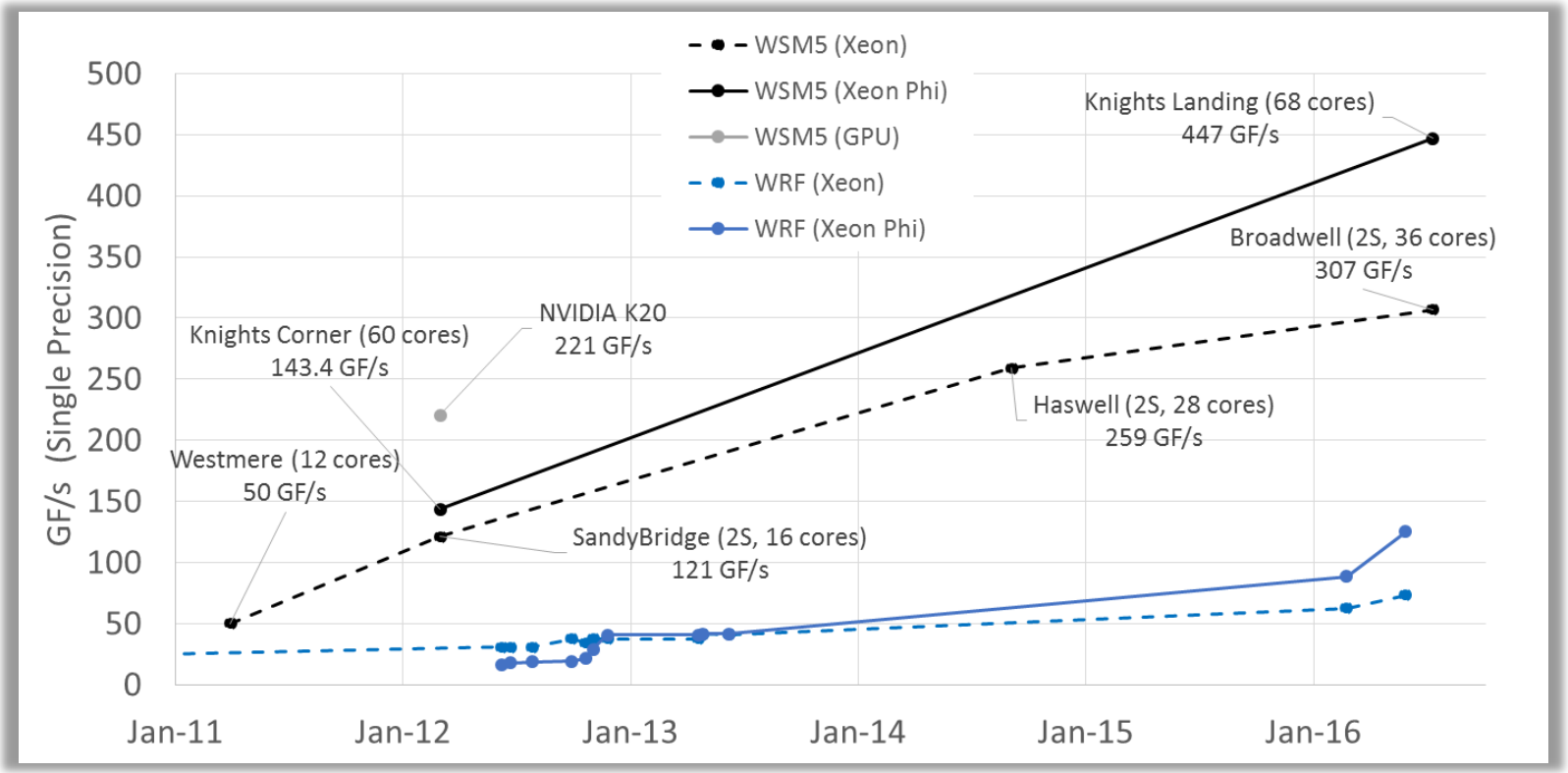
Other observations: SMT (hyperthreading)

- Both WRF and NEPTUNE give best results using half the available (2 of 4) hardware threads per KNL core
- Earlier KNC gave best results using 3 of 4 available hardware threads per core.
- Why?
  - KNL cores have out-of order instruction execution
  - ILP is hiding some of the latency that in-order KNC needed 3 threads to hide



Physics Kernels (Microphysics)

WSM5 Microphysics (single precision)	original	optimized	original→optim.
KNL	102 GF/s	447 GF/s	4.4x
BDW	147 GF/s	307 GF/s	2.1x
KNL / BDW	0.69 x	**1.46 x**	



http://www2.mmm.ucar.edu/wrf/WG2/GPU/WSM5.htm

Physics Kernels (radiation and chemistry)

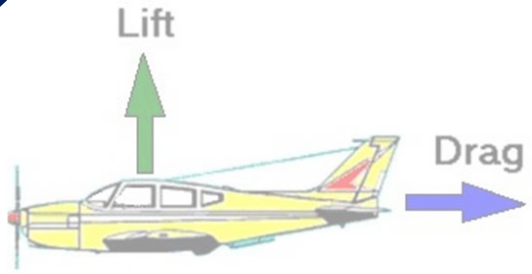
RRTMG Radiation (double precision)	original	optimized	original→optim.
KNC	10 GF/s	36 GF/s	3.6x
KNL (7250 68c, 1.4 GHz)	64 GF/s	116 GF/s	1.8x
BDW (2S E5-2697v4 2.3 GHz)	95 GF/s	**128 GF/s**	1.35x
KNL / KNC	3.6 x	**2.2 x**	
KNL / BDW	0.67 x	**0.91 x**	

Michalakes, Iacono and Jessup. Optimizing Weather Model Radiative Transfer Physics for Intel's Many Integrated Core (MIC) Architecture. *Parallel Processing Letters*. World Scientific. Accepted.

Reactive Chemistry	original	optimized	original→optim.
KNL	32 GF/s	70 GF/s	2.18 x
BDW	59 GF/s	34 GF/s	0.57 x
KNL / BDW (unopt.)	0.54 x	**1.18 x**	

http://www2.mmm.ucar.edu/wrf/WG2/GPU/Chem.htm

Overall experiences



- Lift factors
  - Improved memory bandwidth in KNL's MCDRAM
  - Standard languages, programming models, and tools (next slide)
  - Improvements for Xeon usually benefit Phi and vice versa
- Drag factors
  - Low computational intensity, flat profiles in NWP models
  - Unexamined use of double precision, -fp-model precise, full-precision intrinsics
  - No hardware floating point counters
    - Today's performance based analysis depends on computational intensity, which requires data traffic and floating point measurements of code and representative (ie. *large*) workloads
    - Only an emulator (SDE) is available that runs orders of magnitude slower than real-time
  - Leap-frogging Xeon and Xeon Phi

Tools overview

	Profiling	Opcount and FP rate	Memory traffic and BW	Thread utilization	Vector utilization	Roofline analysis	Execution traces
Intel Vtune Amplifier	By routine, line, and instruction		Timeseries B/W output but and traffic counts	Histogram	General exploration output		B/W, CPU time, spin-time, MPI comms, as fcn of time
Intel Software Development Emulator		Only way to count FP ops on current Xeon	Count load/store between registers and first level cache				Not tried
Intel Advisor	Not tried	Not tried	Not tried		Routine by routine and line by line	Routine by routine (In beta)	
MPE/Jumpshot (ANL/LANS)	By routine, Instrument-ed region						Detailed Gantt plots
Empirical Roofline Toolkit (NERSC/LBL)		Max FP rate	Multiple levels of memory hierarchy			Yes	
Linux Perf PAPI or other	Possible	Possible	Count load/store between LLC and Mem.		Possible		Possible

Acknowledgements

- Intel: Mike Greenfield, Ruchira Sasanka, Indraneil Gokhale, Larry Meadows, Zakhar Matveev, Dmitry Petunin, Dmitry Dontsov, Naik Sumedh, Karthik Raman
- NRL and NPS: Alex Reinecke, Kevin Viner, Jim Doyle, Sasa Gabersek, Andreas Mueller (now at ECMWF), Frank Giraldo
- NCAR: Bill Skamarock, Michael Duda, John Dennis, Chris Kerr
- NOAA: Mark Govett, Jim Rosinski, Tom Henderson (now at Spire)
- Google search, which usually leads to something John McAlpin has already figured out...