Lattice2017

35TH INTERNATIONAL SYMPOSIUM ON LATTICE FIELD THEORY

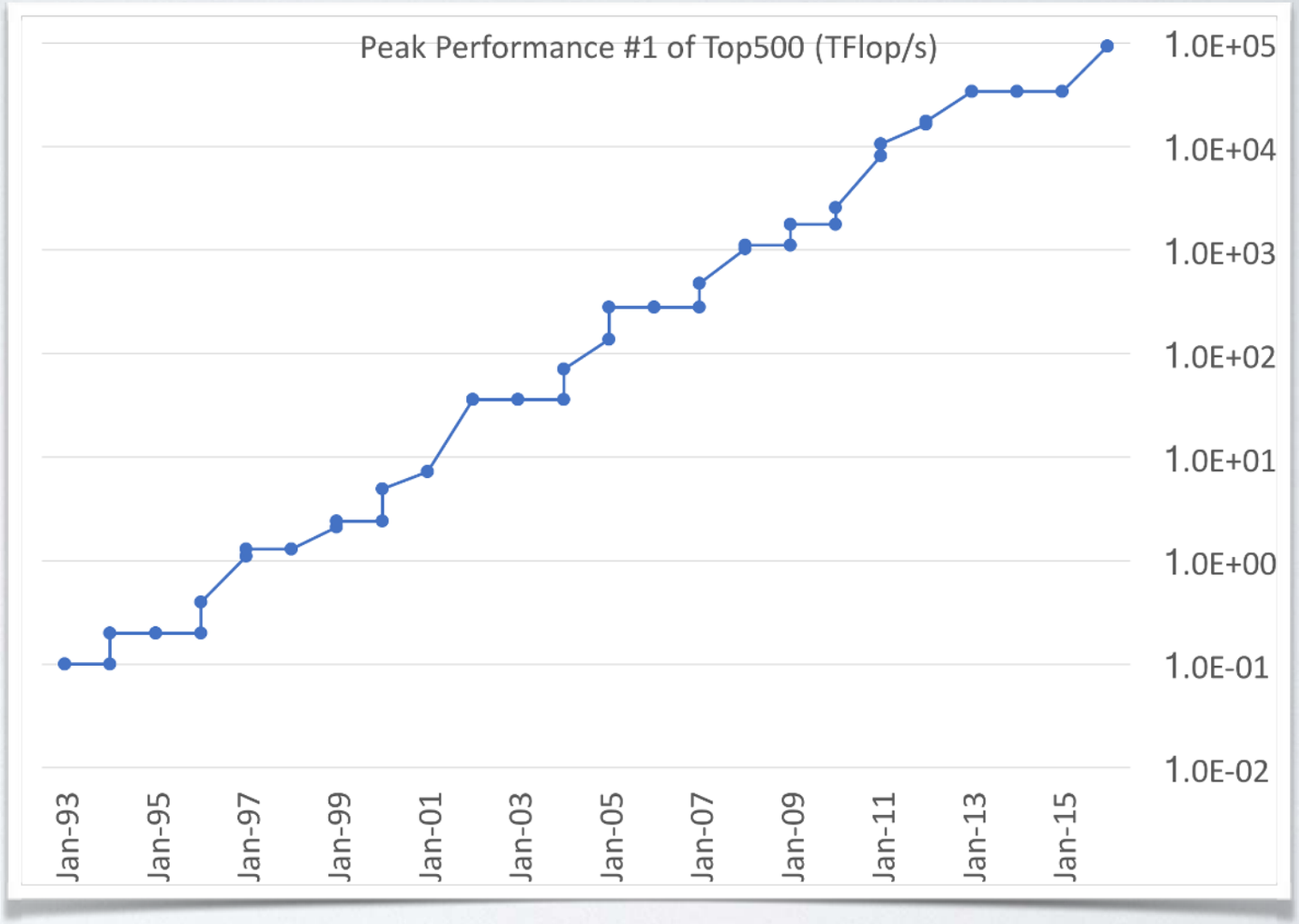
LATTICE QCD ON NEW CHIPS:

A COMMUNITY SUMMARY

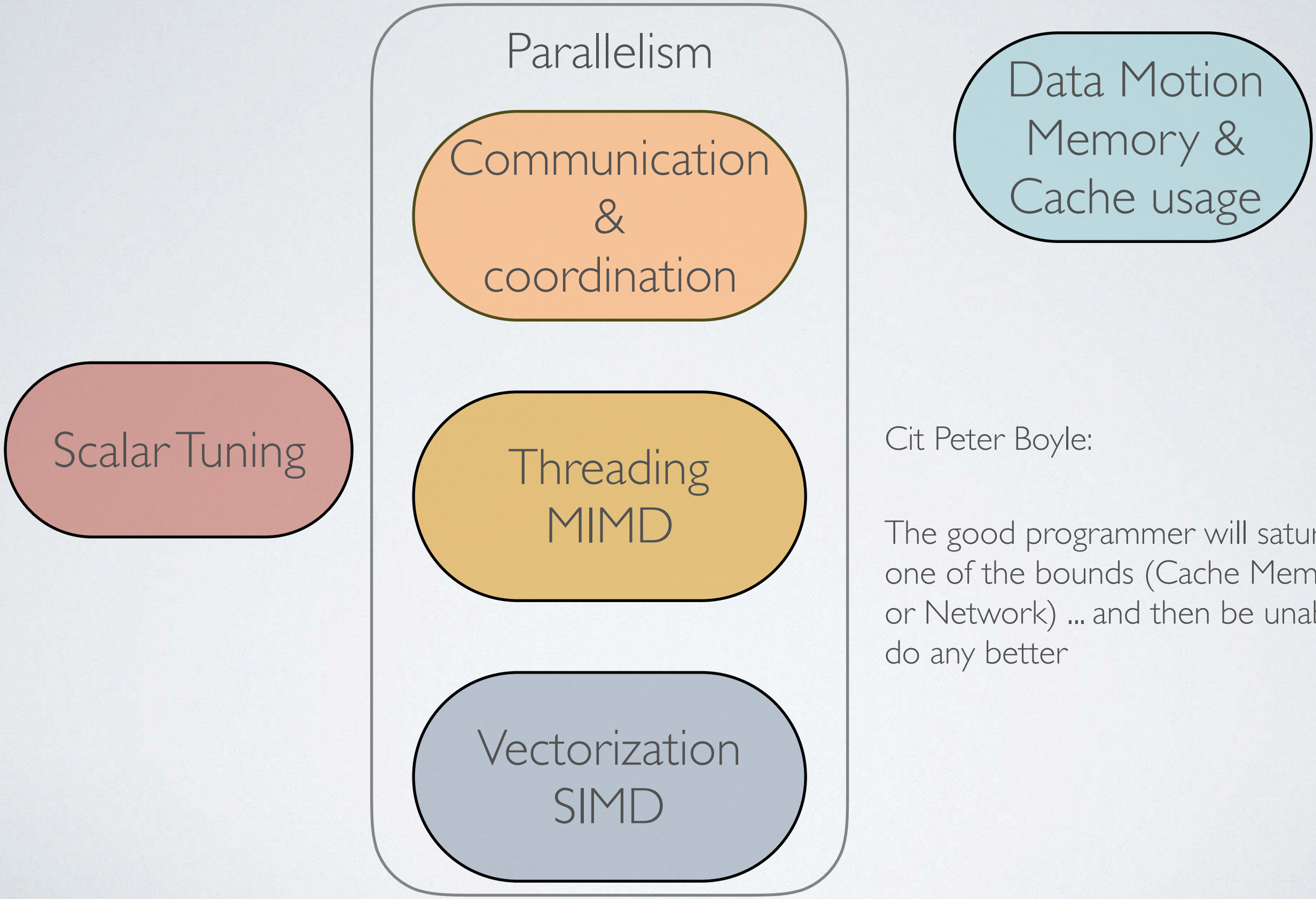
Antonio Rago
Plymouth University

MOORE'S LAW

- 1965 - Moore proposed that companies should double the number of transistors on a given area of a chip every two years
- This growth over the years have been declined as:
  - Increase in number of cores
  - Increase in vectorisation
  - Memory's structure/hierarchy
  - Interconnection speed (not really)
- Silicon era might be over: what next? (Tunnelling Transistor, Spintronic)
- Companies might be more interested in energy efficiency and not raw computational power.



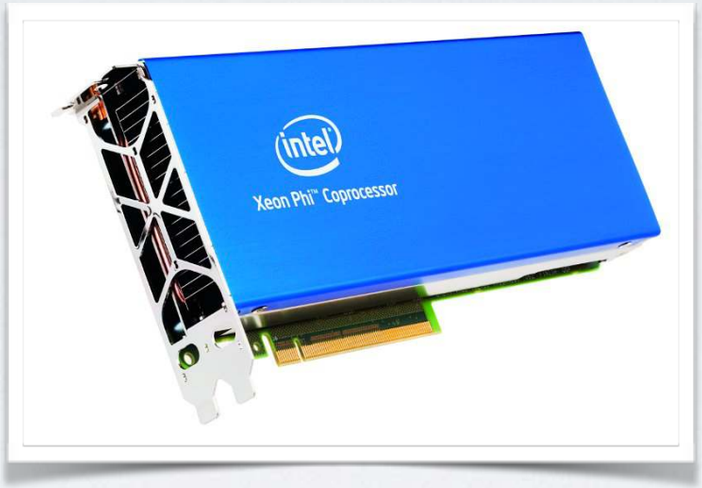
PROGRAMMER'S ROADMAP

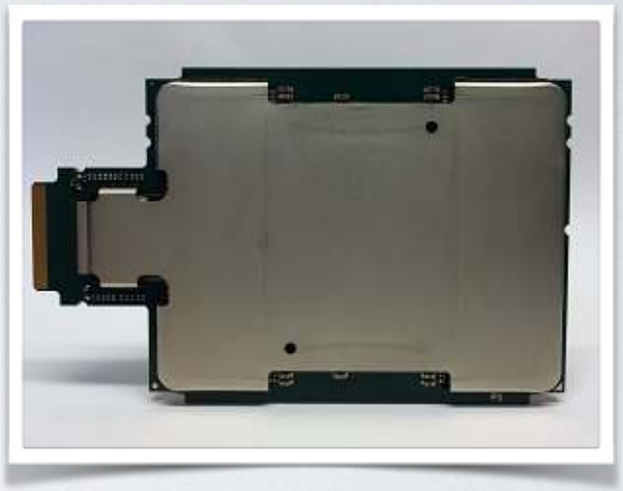


Cit Peter Boyle:

The good programmer will saturate one of the bounds (Cache Memory or Network) ... and then be unable to do any better

Novel(?) Hardware: Xeon Phi





Bootable Host Processor

- RHEL/CentOS/SUSE/Win
- 64cores×4HT,1.3GHz
- ≤384GiBDDR4,>90GB/s
- 16GiBHBM,>400GB/s
- PCIe bus for networking

KNL Coprocessor

- PCIe add-in card
  - Requires host
- Multiple KNLs in a system

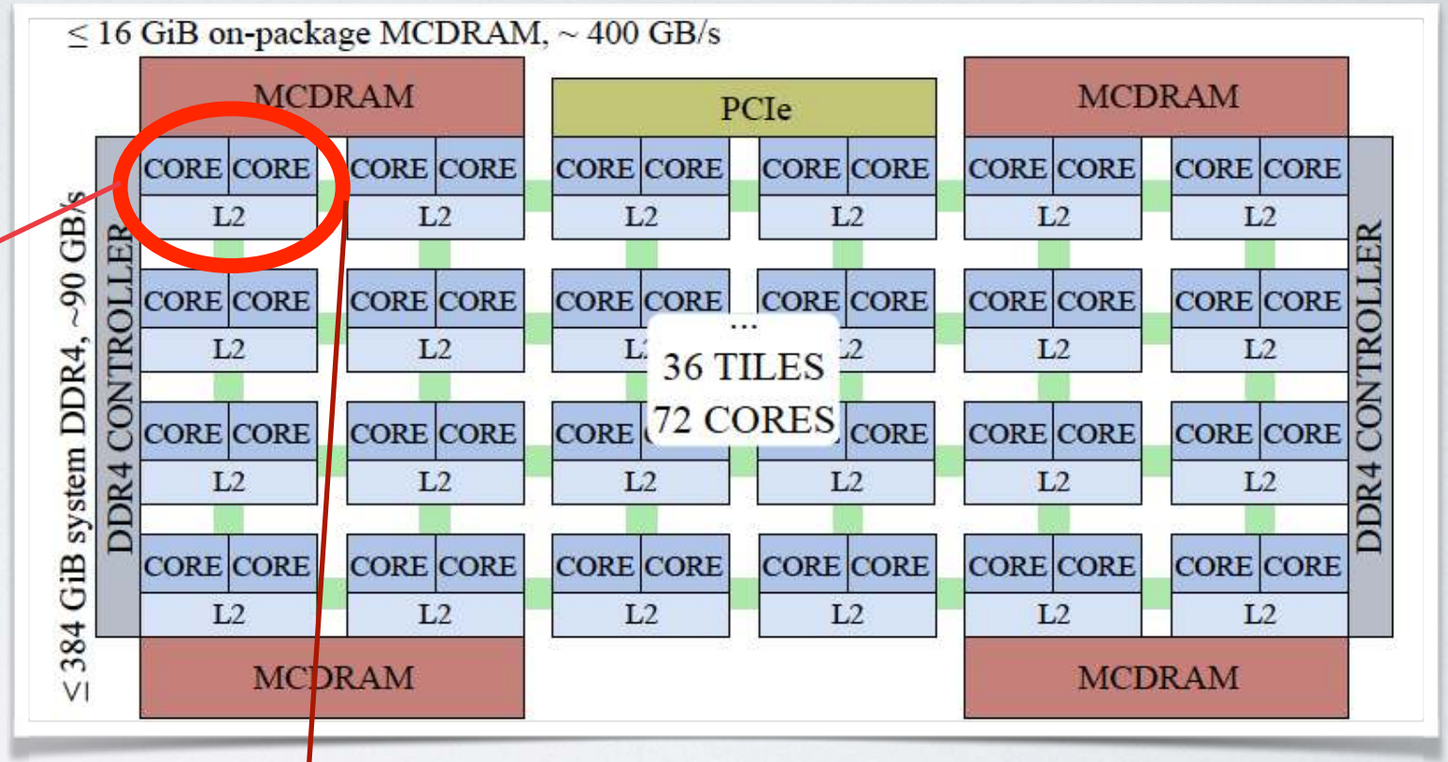
KNLF: KNL with Fabric

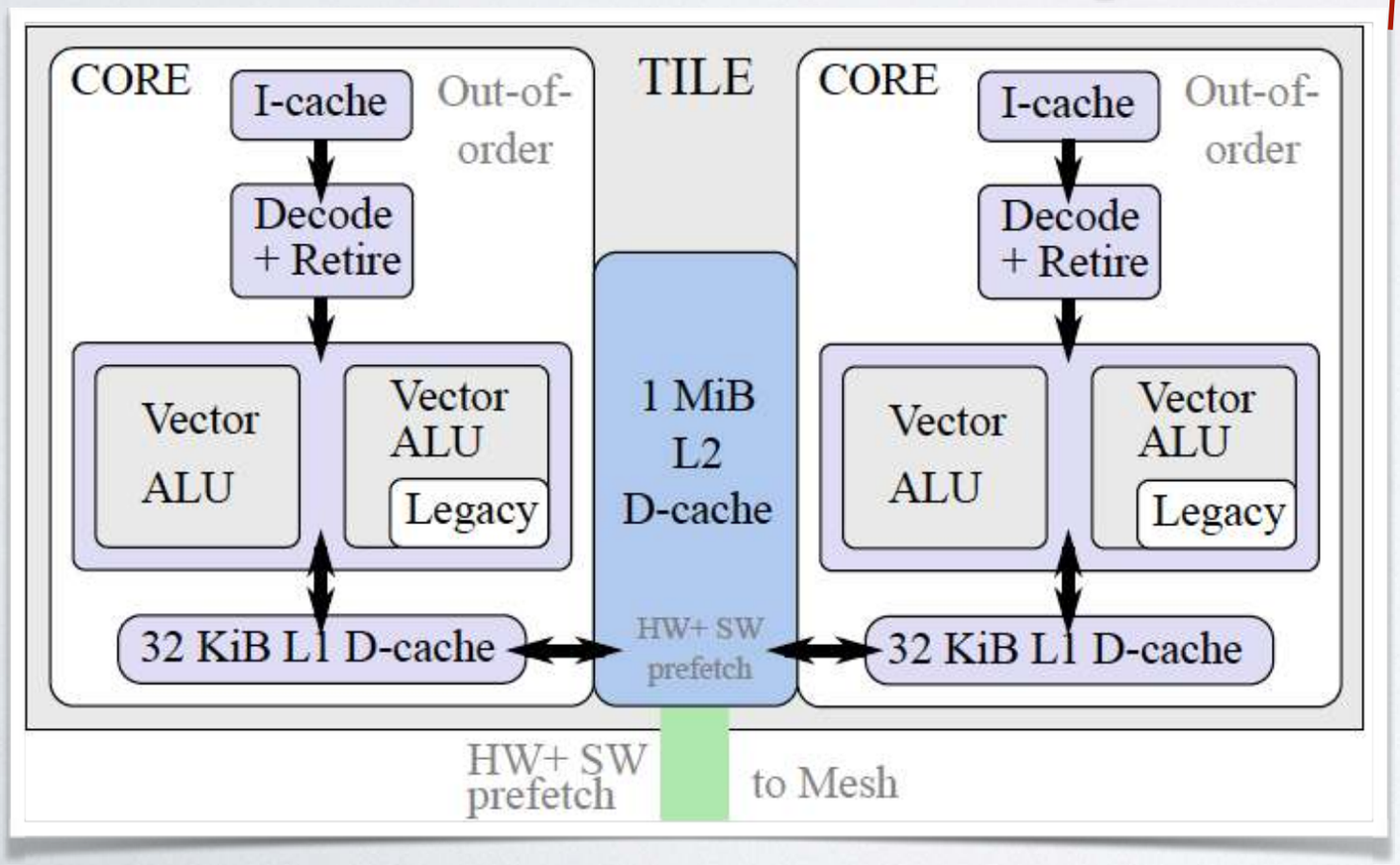
- Fabric integrated on CPU
  - Intel® Omni-Path Architecture
- Socket mount processor

Credit: Intel/Colfax

KNL ORGANIZATION

- Up to 36 tiles, each with 2 physical cores (72total).
- Distributed L2 cache across a mesh interconnect.



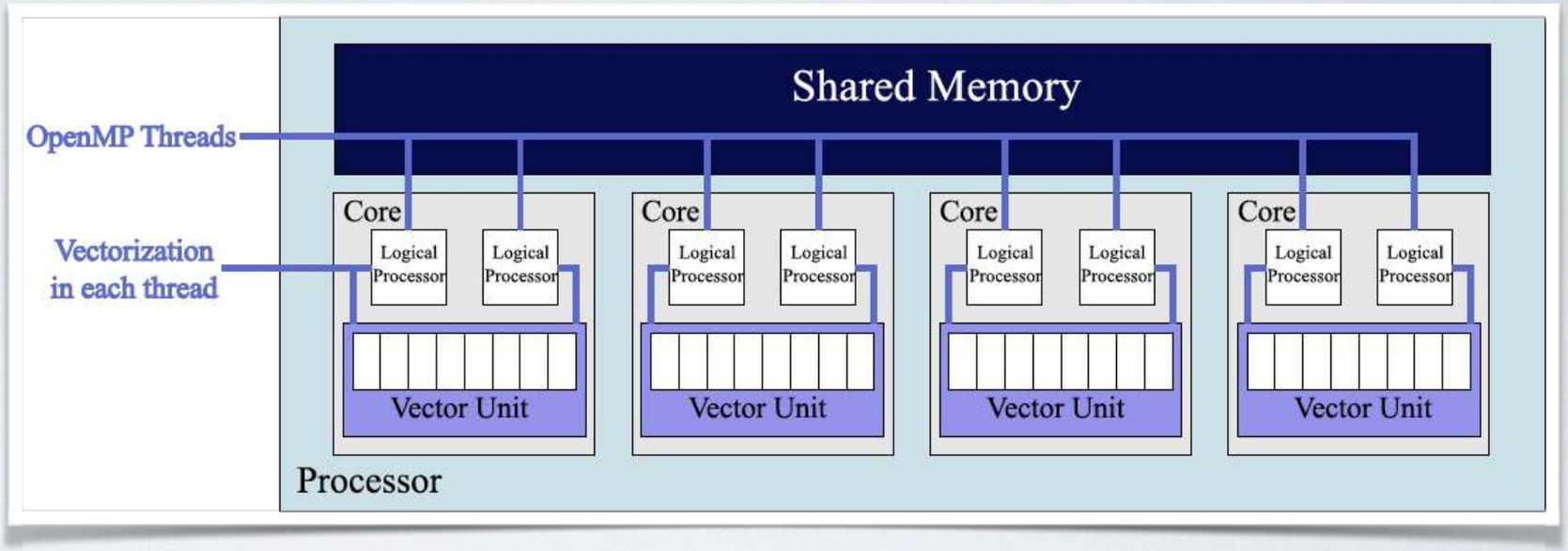


- 4-way hyper-threading (up to 4×72=288 logical processors)
- L2 cache shared by 2cores on a tile.

Credit: Intel/Colfax

KNL CPU & VPU

Cores: run multiple threads/ processes (MIMD)
Vectors: each thread (process) issues vector instructions (SIMD)

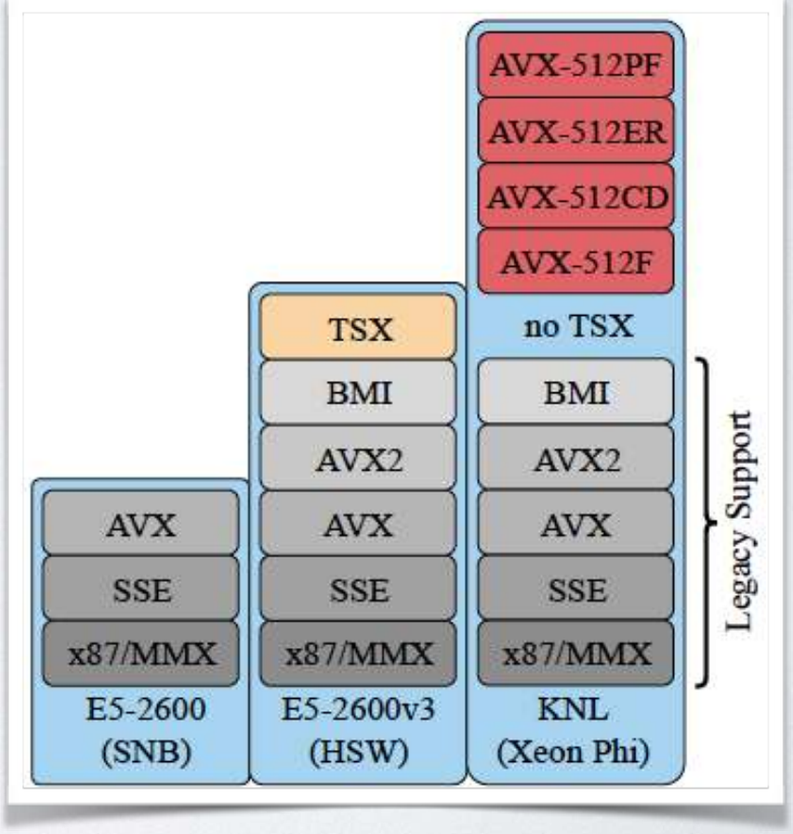


Each core has two Vector Processing Units (VPUs).

Gain for vectorized code

SP → 512 bit registers /32 bits × 2 VPUs = **32** SIMD lanes

DP → 512 bit registers /64 bits × 2 VPUs = **16** SIMD lanes



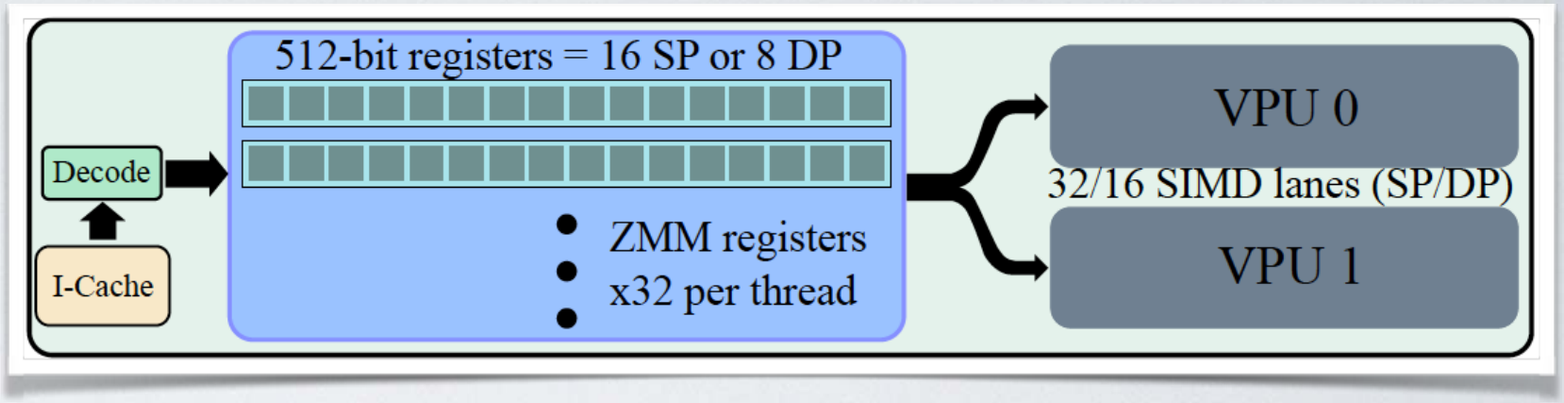
Credit: Intel/Colfax

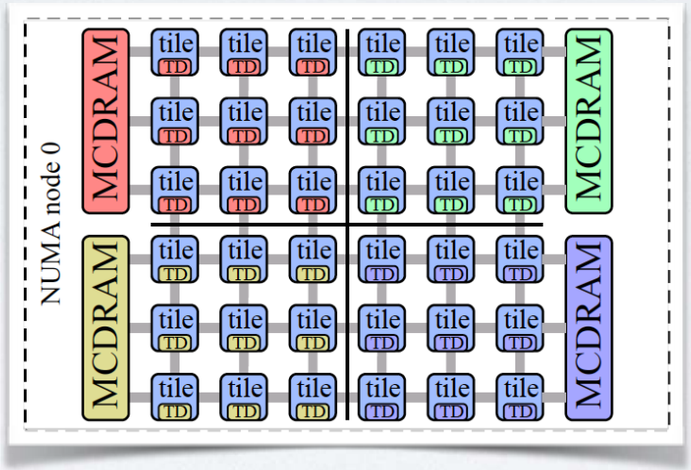
KEY POINTS IN KNL OPTIMIZATION

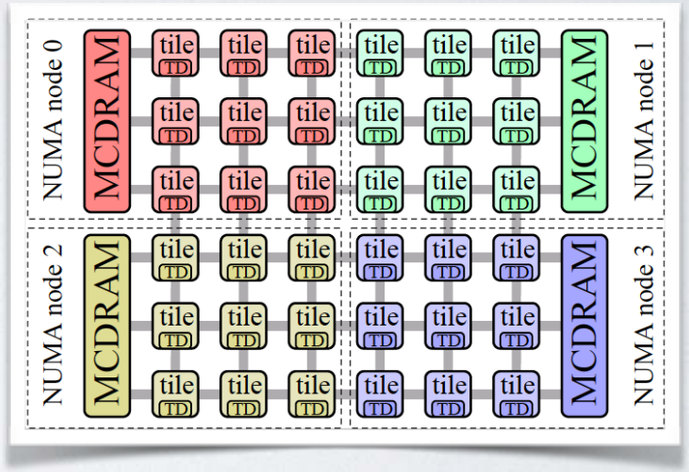
- Vectorization: Take full advantage from the 512 bit registers of the 2 VPU (compiler automatic vectorization?)

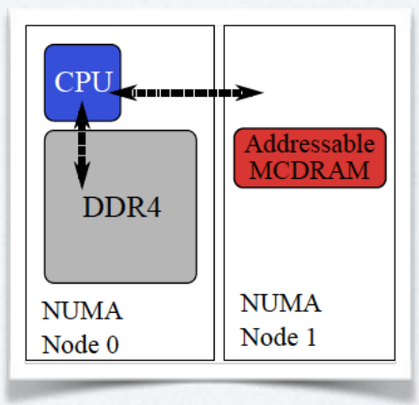
- Distributed L2 Cache with a mesh interconnect: data locality is of paramount importance

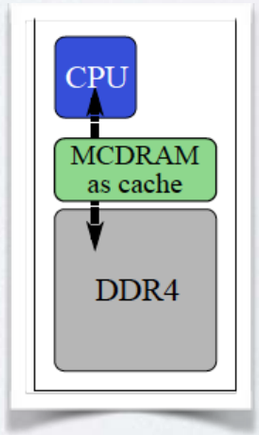
- High Bandwidth 400 GB/s (4 times the DDR) MCDRAM: 16GB (Flat, Cache, Hybrid)

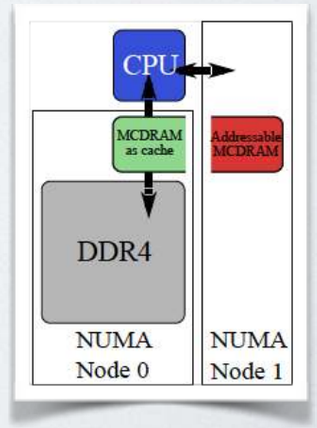












Credit: Intel/Colfax

KNL CONTRIBUTION LIST:

- Durr (Monday, 19 June 2017 16:40)
  Optimization of the Brillouin operator on the KNL architecture

- Boyle, Cossu, Portelli, Yamaguchi (cancelled)
  Grid software status and performance

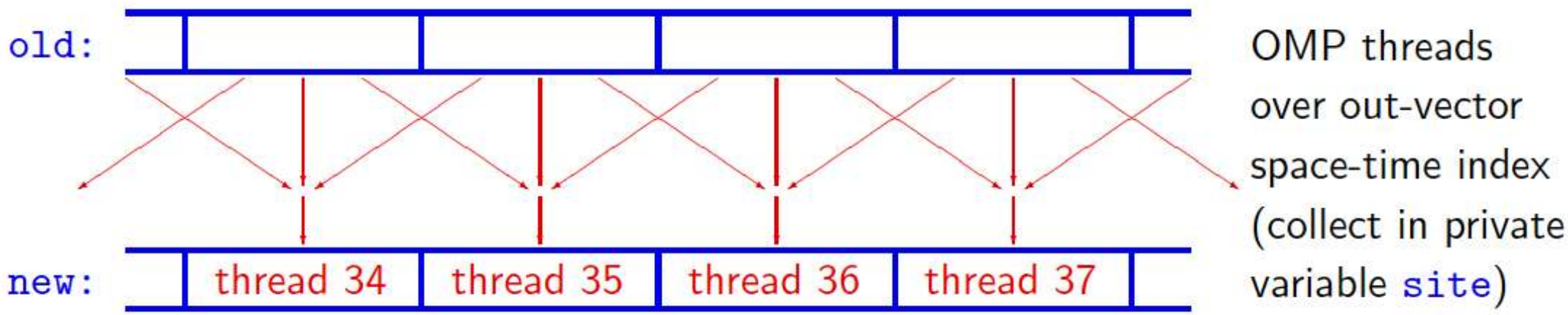
- Kanamori, Matsufuru (Friday, 23 June 2017 17:10)
  Wilson and Domainwall Kernels on Oakforest-PACS

- Li, DeTar, Gottlieb, Toussaint (Tuesday, 20 June 2017 17:10)
  MILC code performance on high end CPU and GPU supercomputer clusters

Optimization of the Brillouin operator on the KNL architecture

Durr

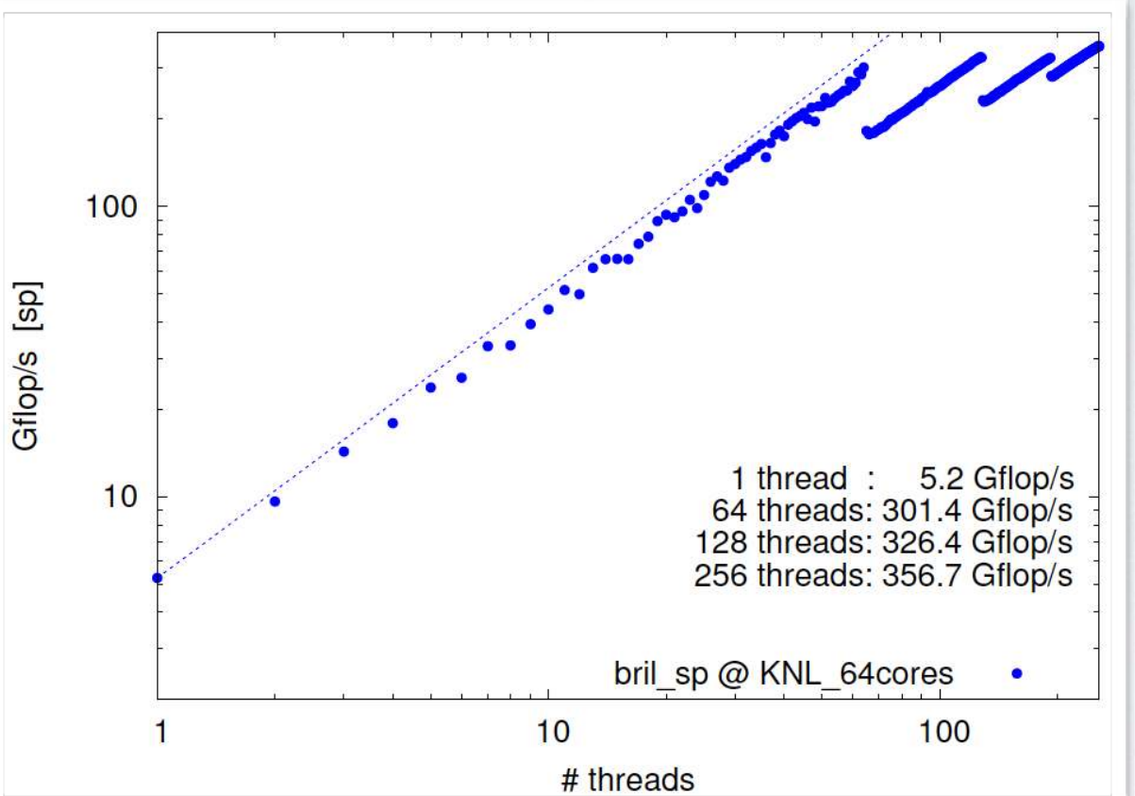
- Do NOT alter the data layout of the original code
- SIMDize only on number of right-hand sides, not over spacetime
- Distribute the space-time geometry over the OMP thread



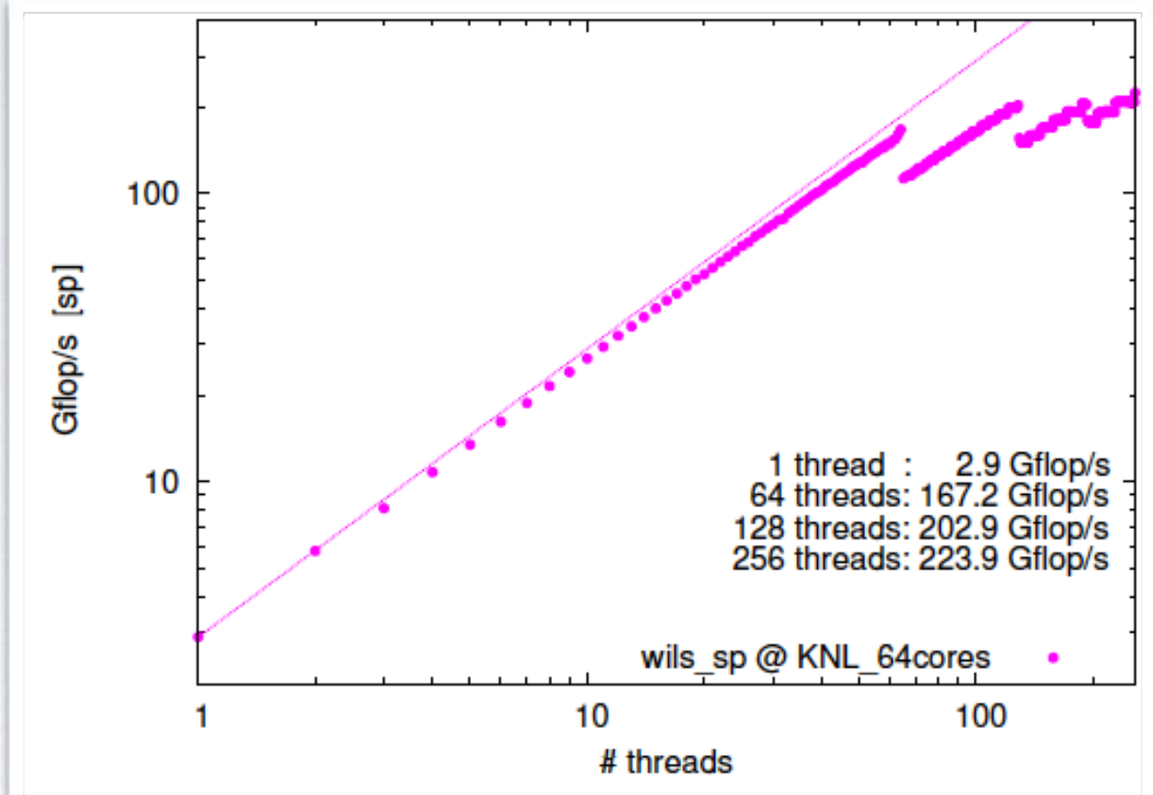
Declare `old/new` vector as complex array of size `(1:Nc,1:4,1:Nv,1:Nx*Ny*Nz*Nt)`:

* color (`1:Nc`) innermost/first ⟵ unroll
* spinor (`1:4`) next ⟵ unroll
* rhs-idx (`1:Nv`) next ⟵ SIMD via OMP pragma
* site (`1:Nx*Ny*Nz*Nt`) outermost/last ⟵ distribute among OMP threads

Single Precision performances



Brillouin



Wilson

Brillouin on KNL: 360 Gflop/s for $N_c = 3, N_V = 4N_c$ and $N_{thr} = 256$ is ~6.3% of peak

Wilson on KNL: 224 Gflop/s for $N_c = 3, N_V = 4N_c$ and $N_{thr} = 256$ is ~3.8% of peak

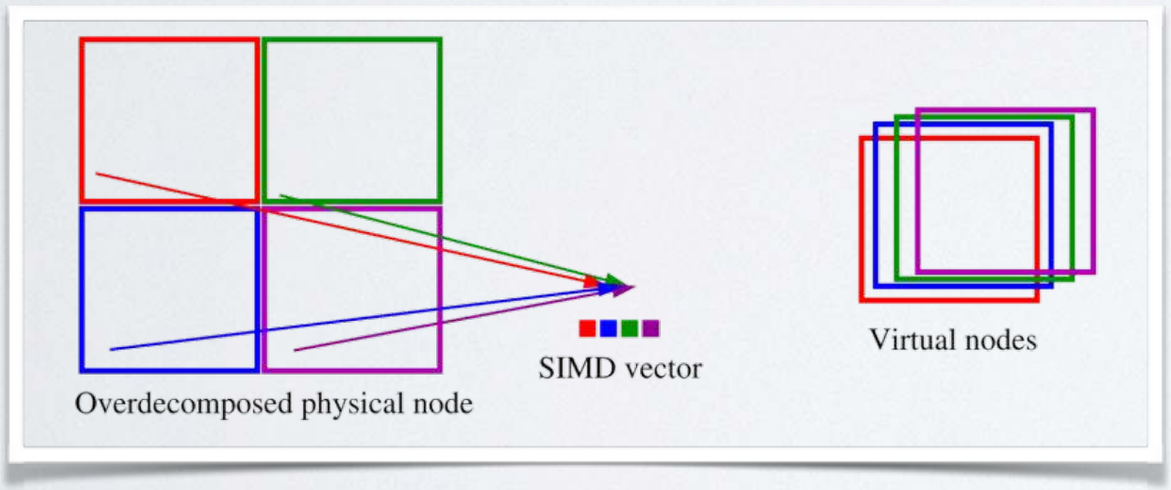
The ratio of computational intensity of Brillouin to Wilson is 2.5

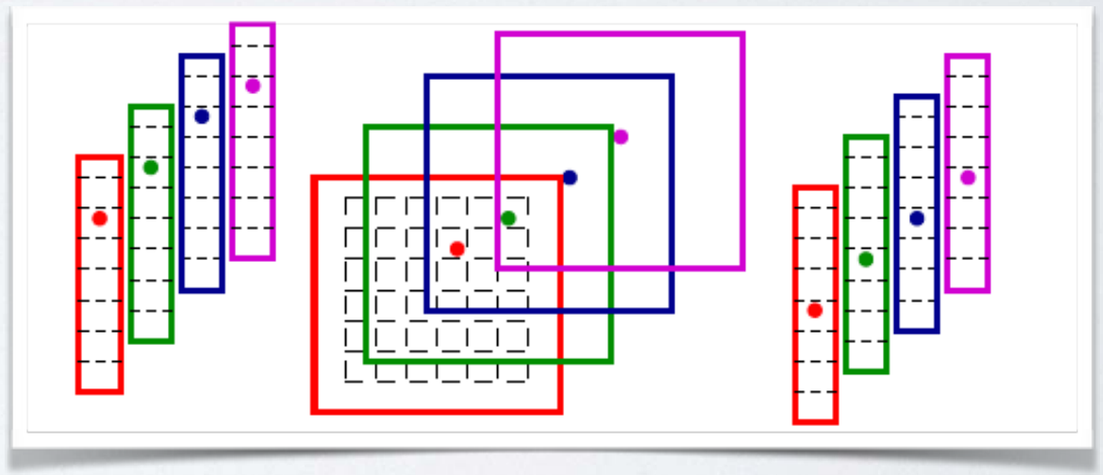
Brillouin I7 Broadwell with 6 cores: 160 Gflop/s for $N_c = 3, N_V = 4N_c$ and $N_{thr} = 012$ is ~23% of peak

Grid software status and performance

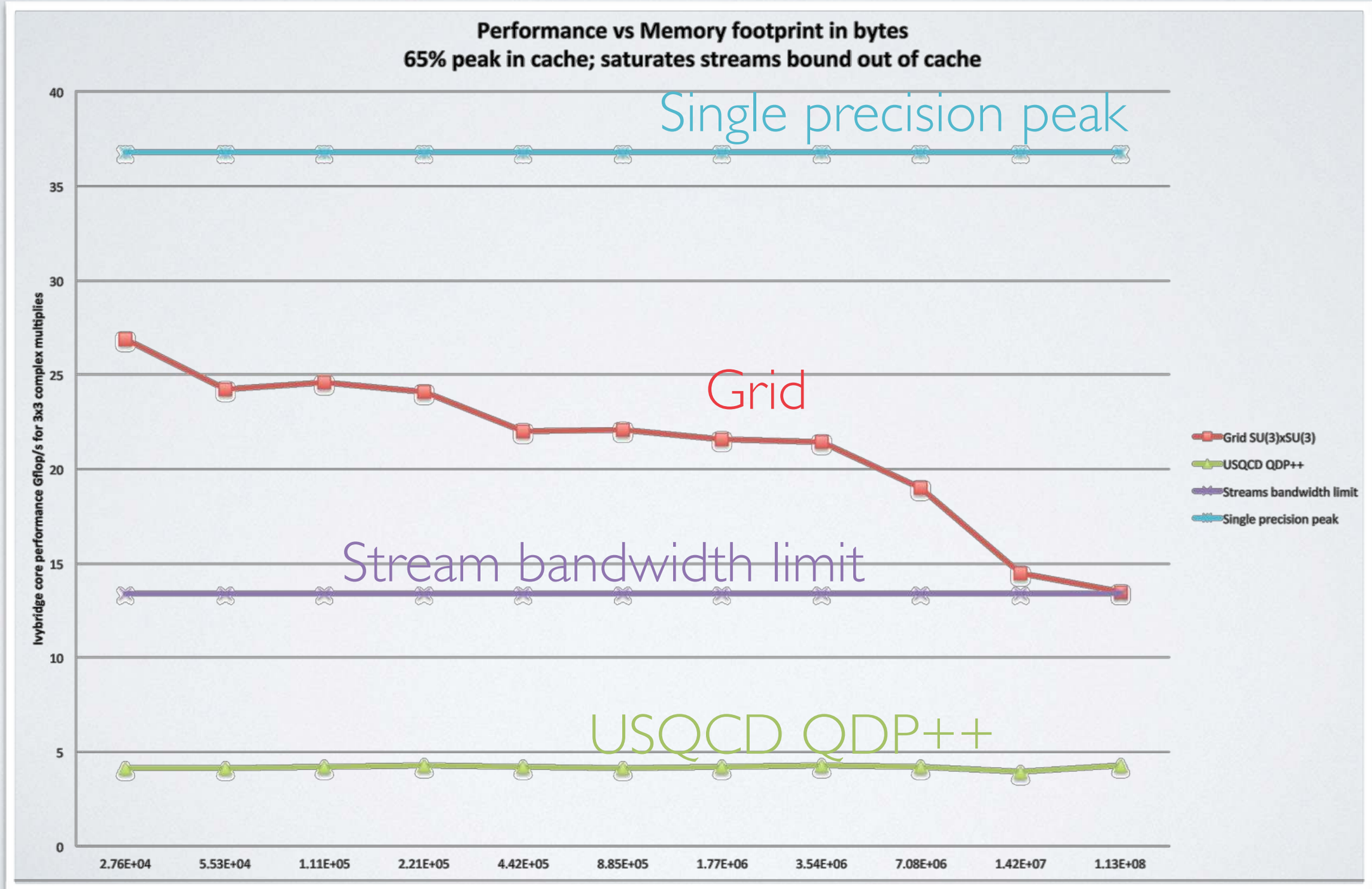
Boyle, Cossu, Portelli, Yamaguchi

- Geometrically decompose cartesian arrays across nodes (MPI)
- Subdivide node volume into smaller virtual nodes
- Spread virtual nodes across SIMD lanes
- Use OpenMP+MPI+SIMD to process conformable array operations
- Same instructions executed on many nodes, each node operates on Nsimd virtual nodes
- Modify data layout to align data parallel operations to SIMD hardware
- Conformable array operations are simple and vectorise perfectly



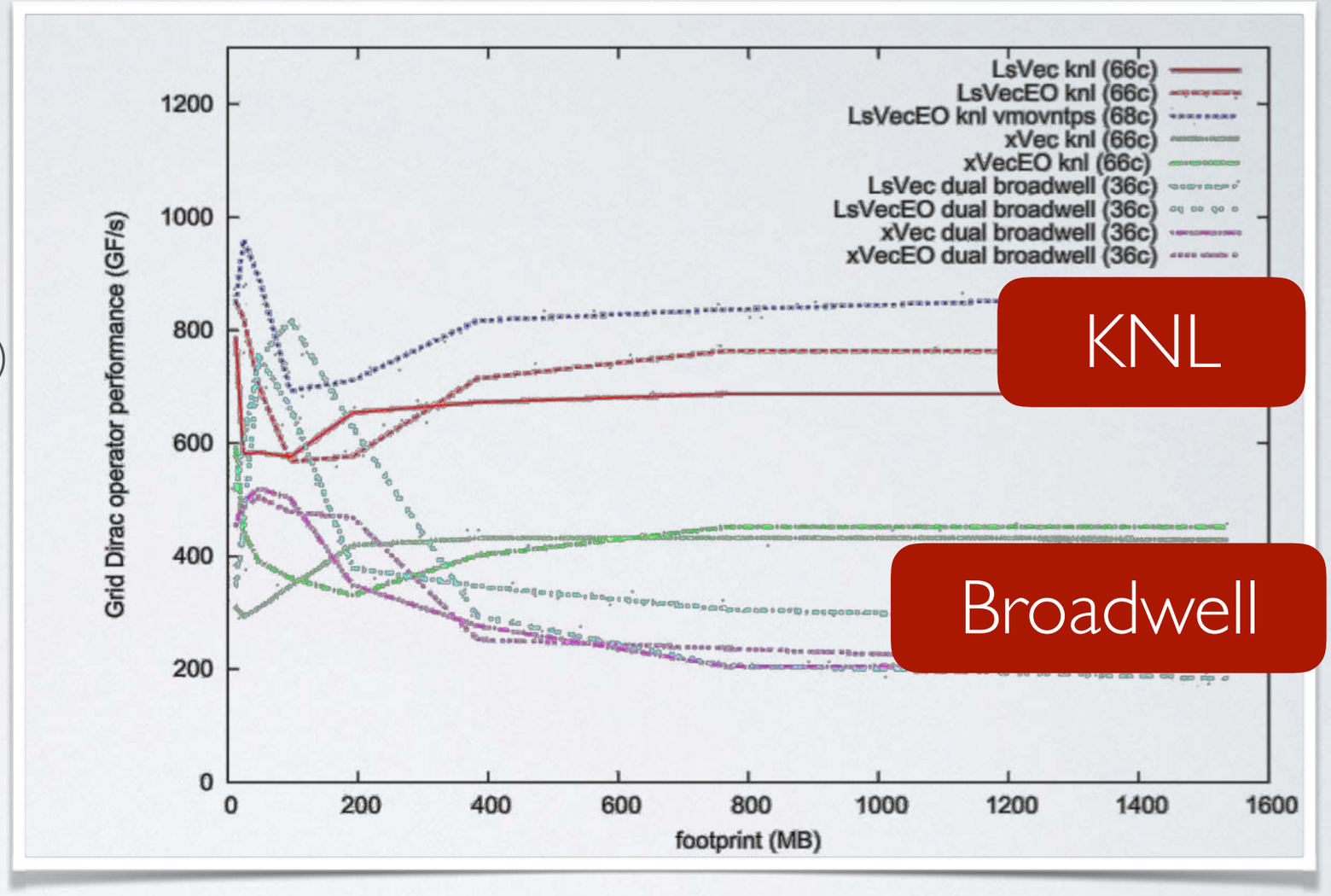


This strategy allows in the case of **SU(3) matrix multiply** on Intel-i7-3615QM (AVX)
Single precision, 65% of peak when in cache



Grid single node, single precision performance for multiRHS Wilson term

- One KNL substantially faster than two Broadwell's (18+18) out of cache
- 1 thread per core fastest after writing in assembler (*not* intrinsics)
- Single core instructions-per-cycle (IPC) is 1.7 (85% of theoretical, 2 IPC)
- Multi-core L1 hit rate is 99% (perfect SFW prefetching)
- Multi-core MCDRAM bandwidth 97% (370GB/s)



Architecture	Cores	Gflops/s	Percentage of Peak
Intel Knight's Landing 7250	68	960	16%
Intel Broadwell x2	36	800	30%

Grid single node, single precision performance for multiRHS Wilson term

L1 read	L1 write	L2 read	MCDRAM read	MCDRAM write
550	12 (12 theo.)	98 (96 theo.)	68 (12 theo.)	12 (12 theo.)

- Infinite L2 capacity has *in principle* 8x reuse,
  Measured **1.44** = 98/68

- Making all tiles work on same data at same time (interleaving work) proved ineffective. L2 - L2 transfers not scaling well ?

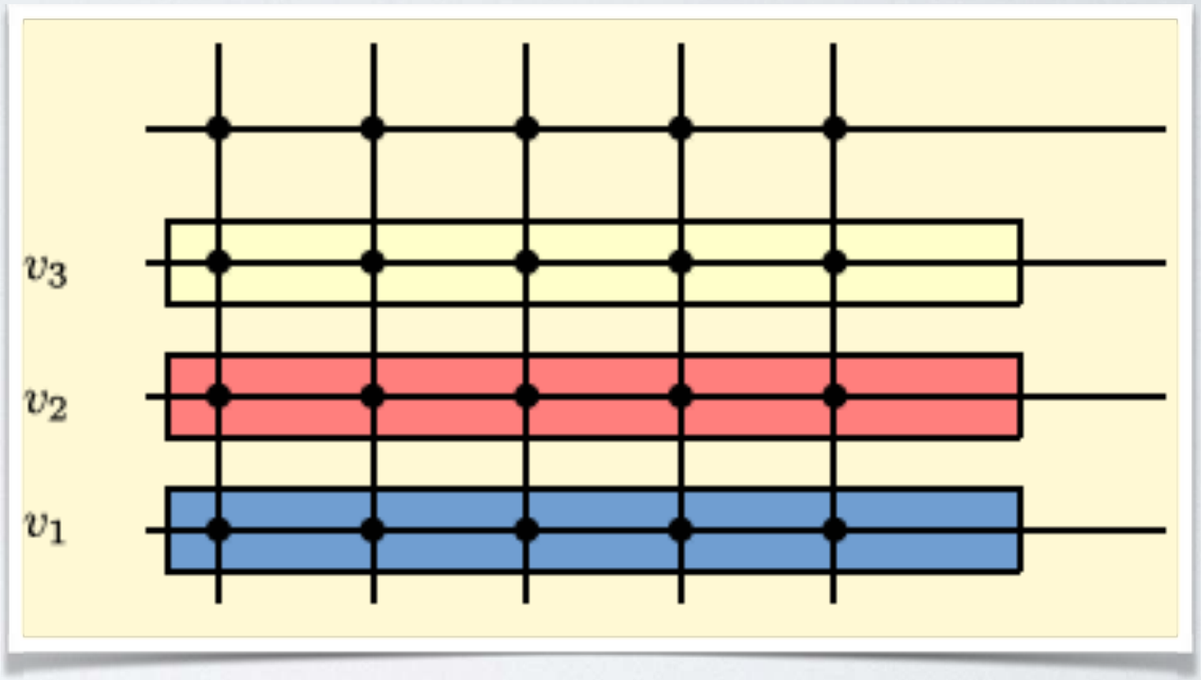
- Was effective on KNC, but perhaps 30% loss here

Wilson and Domainwall Kernels on Oakforest-PACS

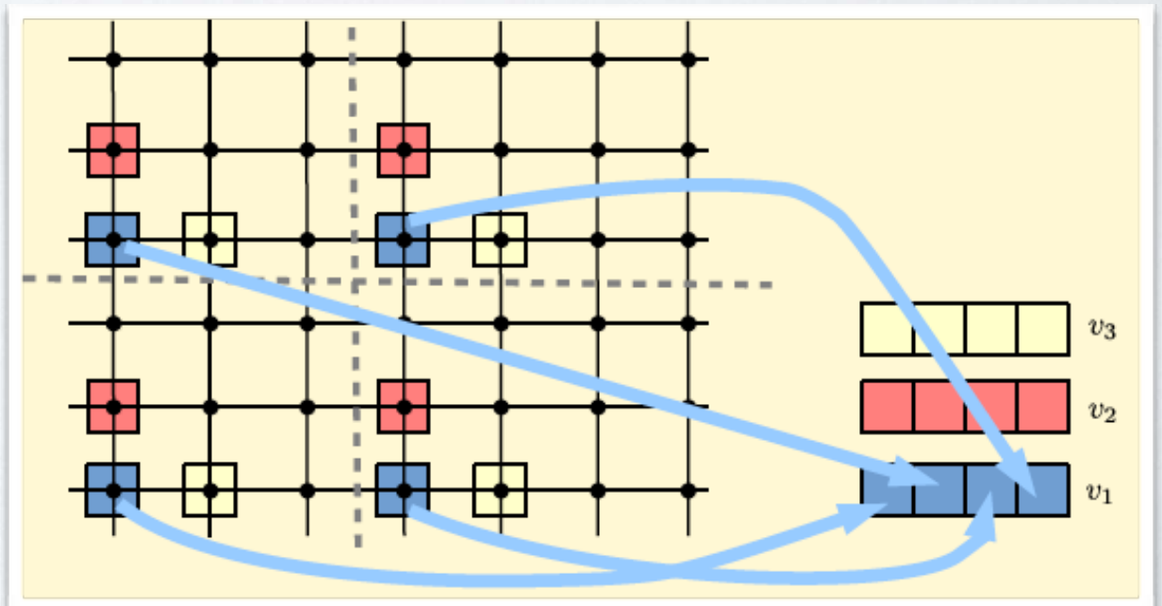
Kanamori, Matsufuru

Developed for Bridge++ two implementations of a Wilson/ Domainwall kernel

- Simple (impl 1):
  - simd vector is continuously packed in x-direction
  - no MPI parallelization in x- direction
  - blocking comm.
  - no manual prefetch
  - AVX512 intrinsics for arithmetics

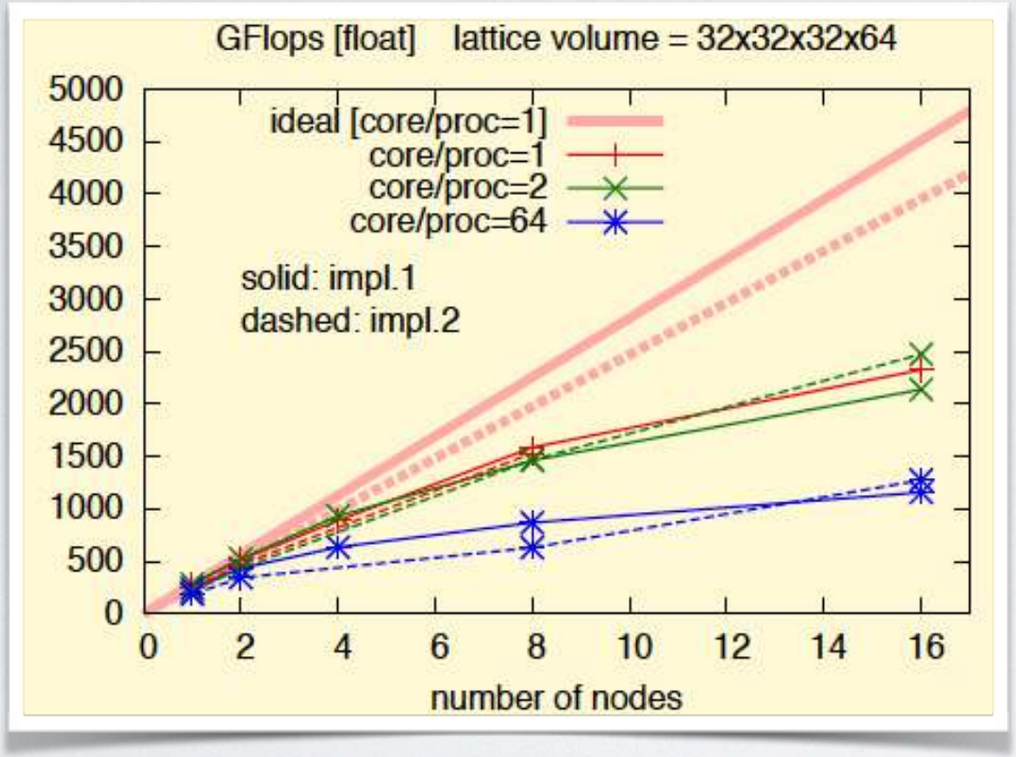


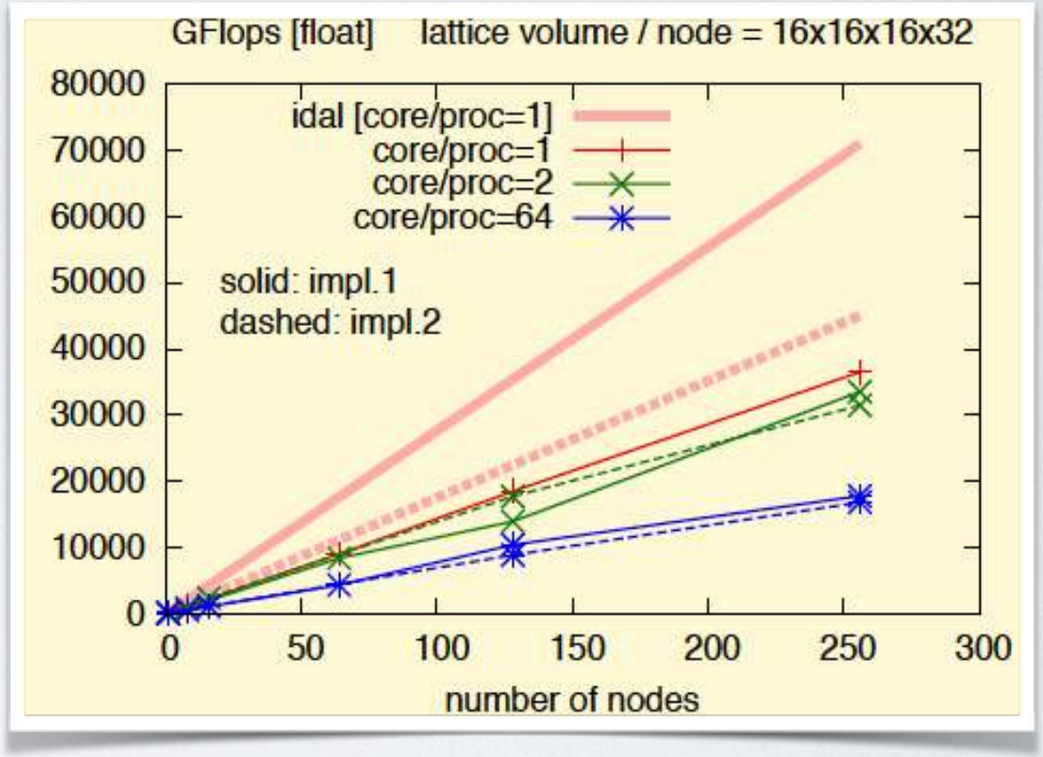
- Aggressive (impl 2) Grid-like
  - simd vector is distributed to subdomains
  - non-blocking comm.
  - (partial) loop tiling
  - manual prefetch



OakForest-PACS PRIMERGY CX1640 M1 by Fujitsu:
Intel Xeon Phi 7250 68C 1.4GHz (KNL) + Intel Omni-Path network topology: Full-bisection Fat Tree

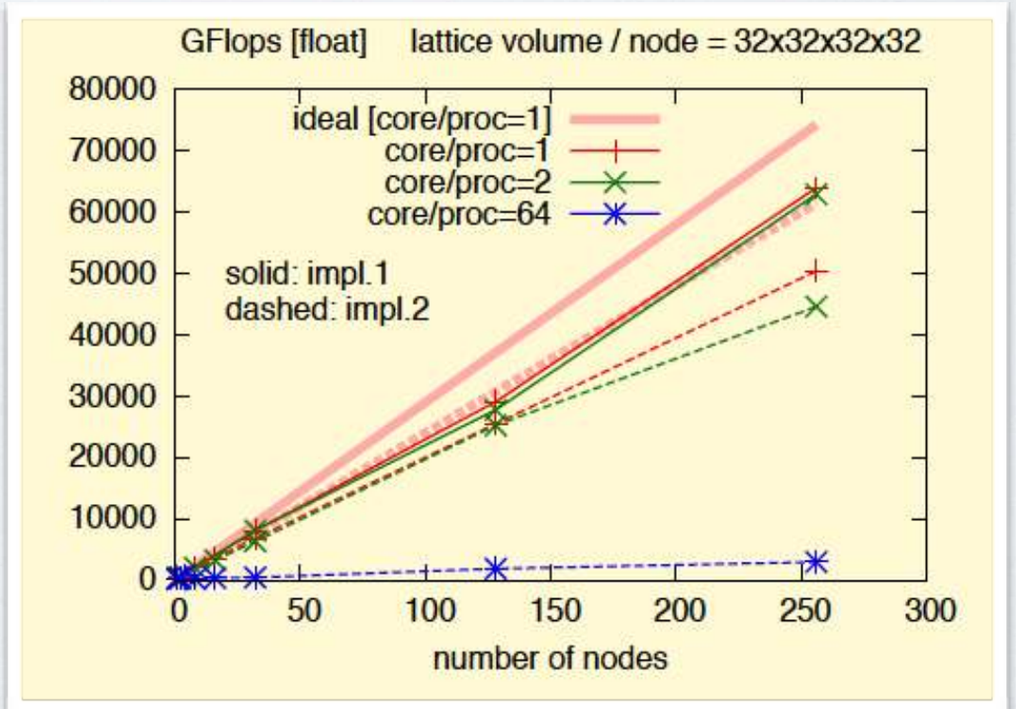
Strong Scaling





Weak Scaling

- Up to 290 GFlops (impl. 1) and 250 GFlops (impl. 2) for single node
- Roughly 1/2 of the ideal scaling
- Fewer OMP threading gives better scaling
- Omp barrier costs a lot



MILC code performance on high end CPU and GPU supercomputer clusters

Li, DeTar, Gottlieb, Toussaint

Libraries:

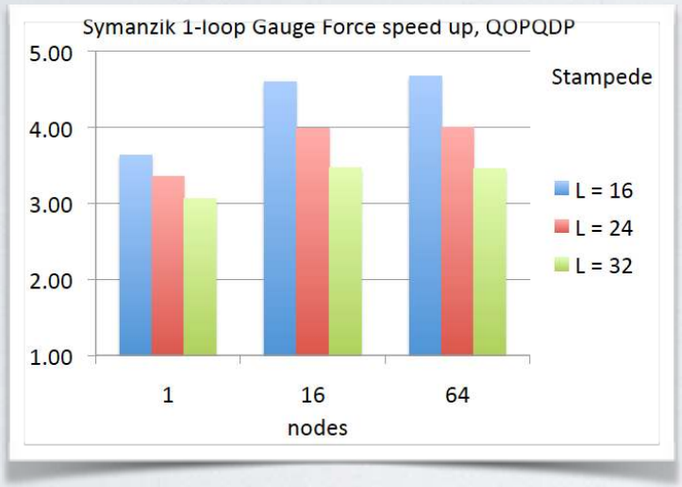
- **QOPQDP**: SSE2 intrinsic (currently no direct AVX support)
- **Staggered QPhiX**: explicit AVX512, optimised for KNL for cache reuse and OMP parallelisation

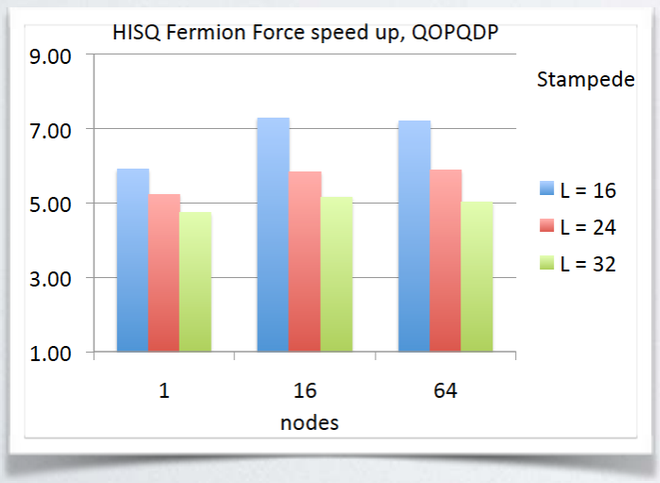
- Intel Xeon: Haswell
  - NERSC Cori cluster, Cray Aries interconnect
- Intel Xeon Phi: Knights Landing
  - NERSC Cori2, KNL 7250, Cray Aries interconnect
  - TACC Stampede, KNL 7250, Intel Omnipath
  - ALCF Theta, KNL 7210, Cray Aries interconnect

MILC code performance on high end CPU and GPU supercomputer clusters

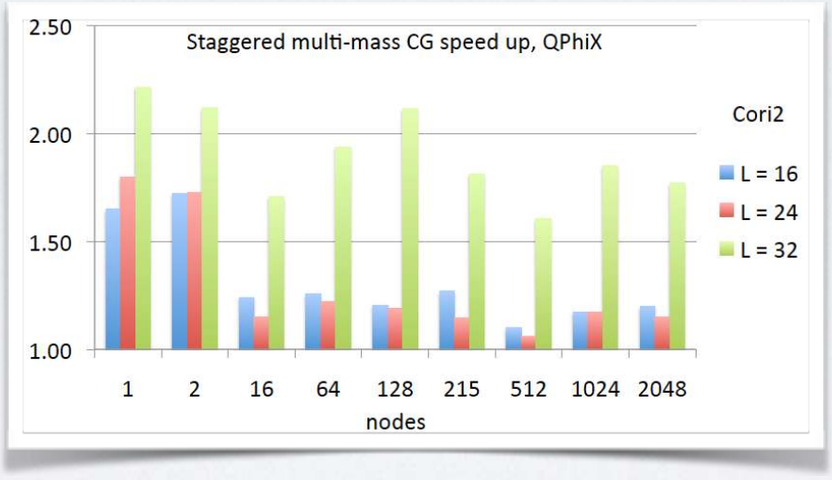
Li, DeTar, Gottlieb, Toussaint

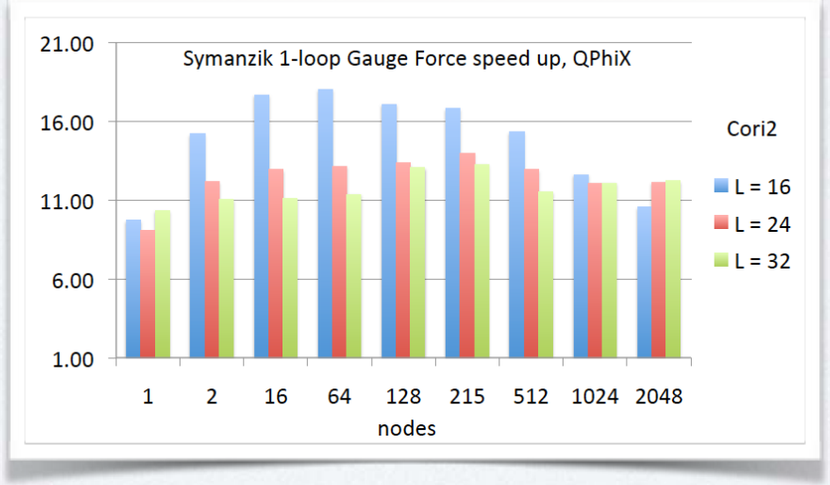
Stampede





Cori2





Cori2

	MILC b.	QPhiX
Gflops, 1 node	70	120
Scaling factor ($16 \le L \le 32$)	0.3 ~ 0.55	0.25 ~ 0.45

	MILC b.	QPhiX
Gflops, 1 node	60	380 (kernel)
Scaling factor ($16 \le L \le 32$)	0.4 ~ 0.85	0.65 ~ 0.9

Novel Hardware: TESLA VOLTA

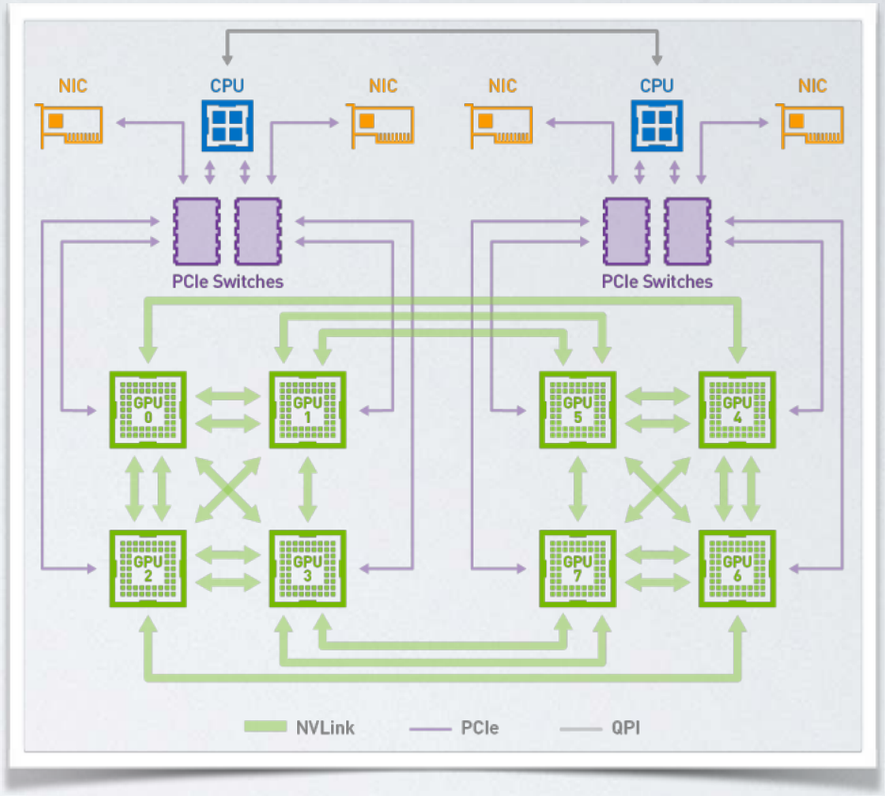
	P100	V100	Ratio
Training acceleration	10 TFLOPS	120 TFLOPS	12x
Inference acceleration	21 TFLOPS	120 TFLOPS	6x
FP64/FP32	5/10 TFLOPS	7.5/15 TFLOPS	1.5x
HBM2 Bandwidth	720 GB/s	900 GB/s	1.2x
NVLink Bandwidth	160 GB/s	300 GB/s	1.9x
L2 Cache	4 MB	6 MB	1.5x
L1 Caches	1.3 MB	10 MB	7.7x

- A GV100 Chip contains 84 SMs
- GV100 SM incorporates 64 FP32 cores and 32 FP64 cores per SM
- Is partitioned into four processing blocks, each with
  - 16 FP32 Cores
  - 8 FP64 Cores
  - 16 INT32 Cores
  - 2 mixed-precision Tensor Cores
  - L0 instruction cache
  - one warp scheduler
  - one dispatch unit
  - 64 KB Register File

INTRANODE COMMUNICATION NVLINK VOLTA

NVlink allows interconnection among GPU or GPU-CPU

- Aggregate BW 300 GB/s
- 50% more links
- 28% faster signalling



Main improvements for LQCD comes from

- 1.5x faster sustained memory bandwidth
  Volta sustains 95% of peak memory bandwidth in STREAM
- Faster NVLink for better multi-GPU performance (300GB/S)
- Improved L1 cache

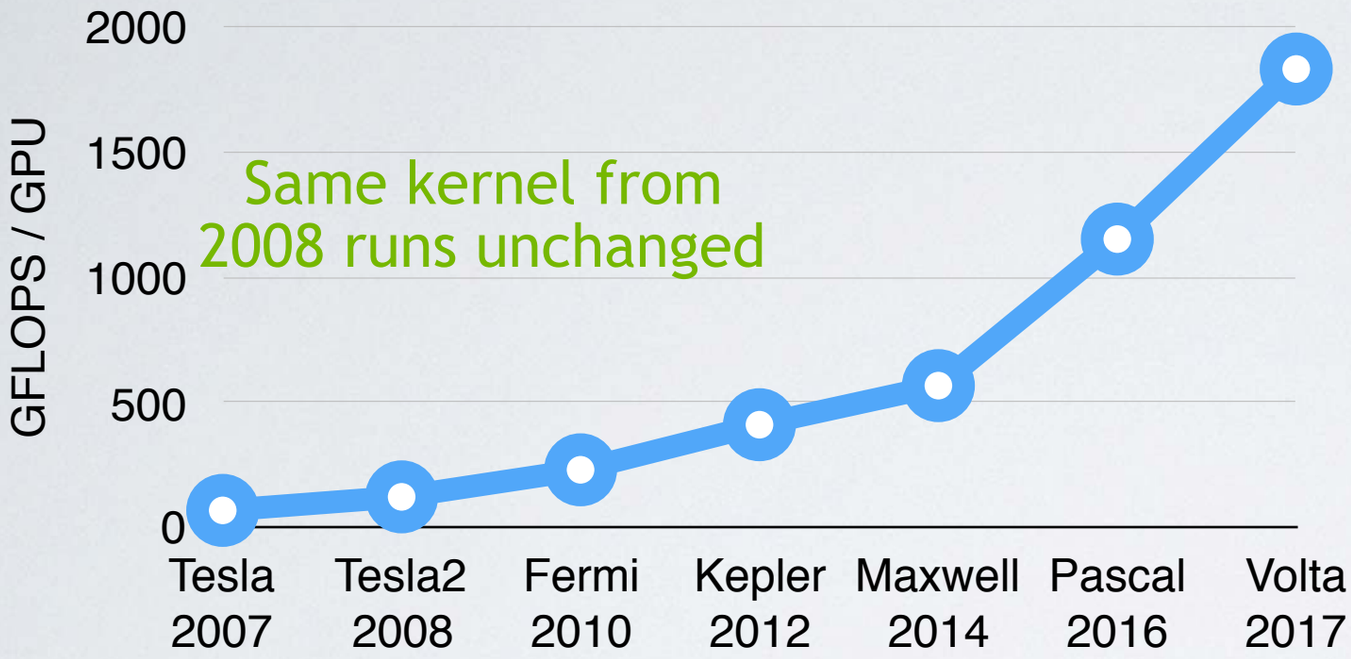
GPU/CUDA CONTRIBUTION LIST:

- Clark, Wagner (Friday, 23 June 2017 15:00)
  Developing QCD Algorithms For NVIDIA GPUs Using the QUDA Framework

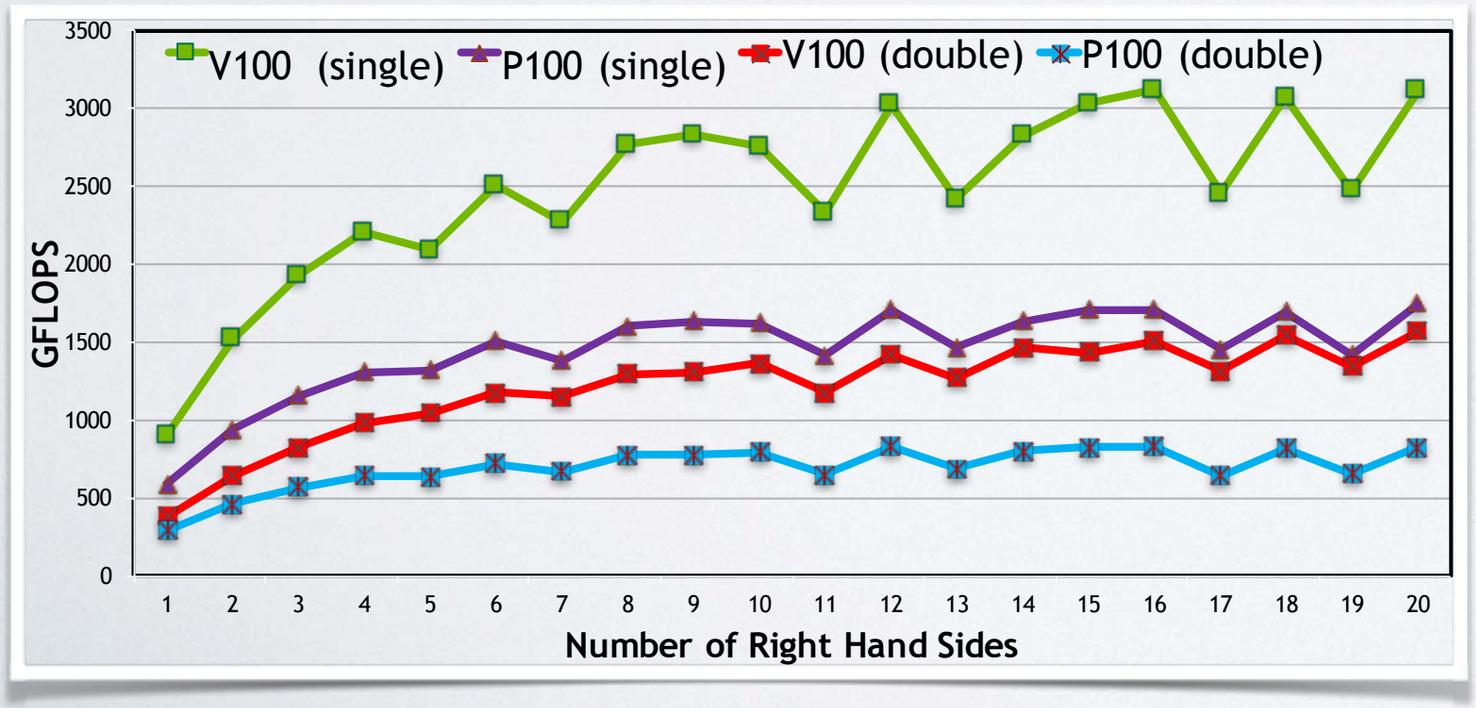
- Lin, Boyle, Clark, DeTar, Rana, Vaquero Avilés-Casco (Friday, 23 June 2017 15:40)
  Performance Portability Strategies for Grid C++ Expression Template

Developing QCD Algorithms For NVIDIA GPUs Using the QUDA Framework

Clark, Wagner



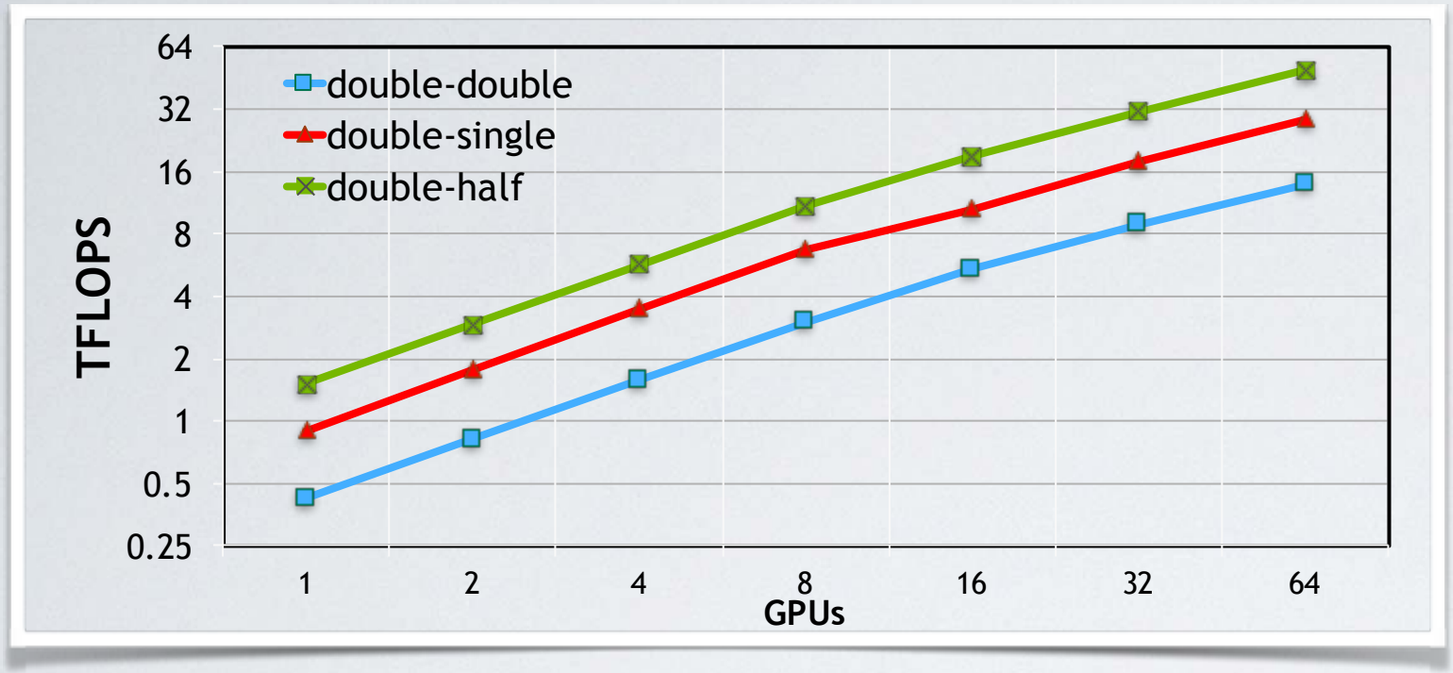
- Single precision Wilson-dslash performance



- Hisq Dslash for multiple rhs

MULTI GPU SCALING

Quda SOLVER WEAK scaling

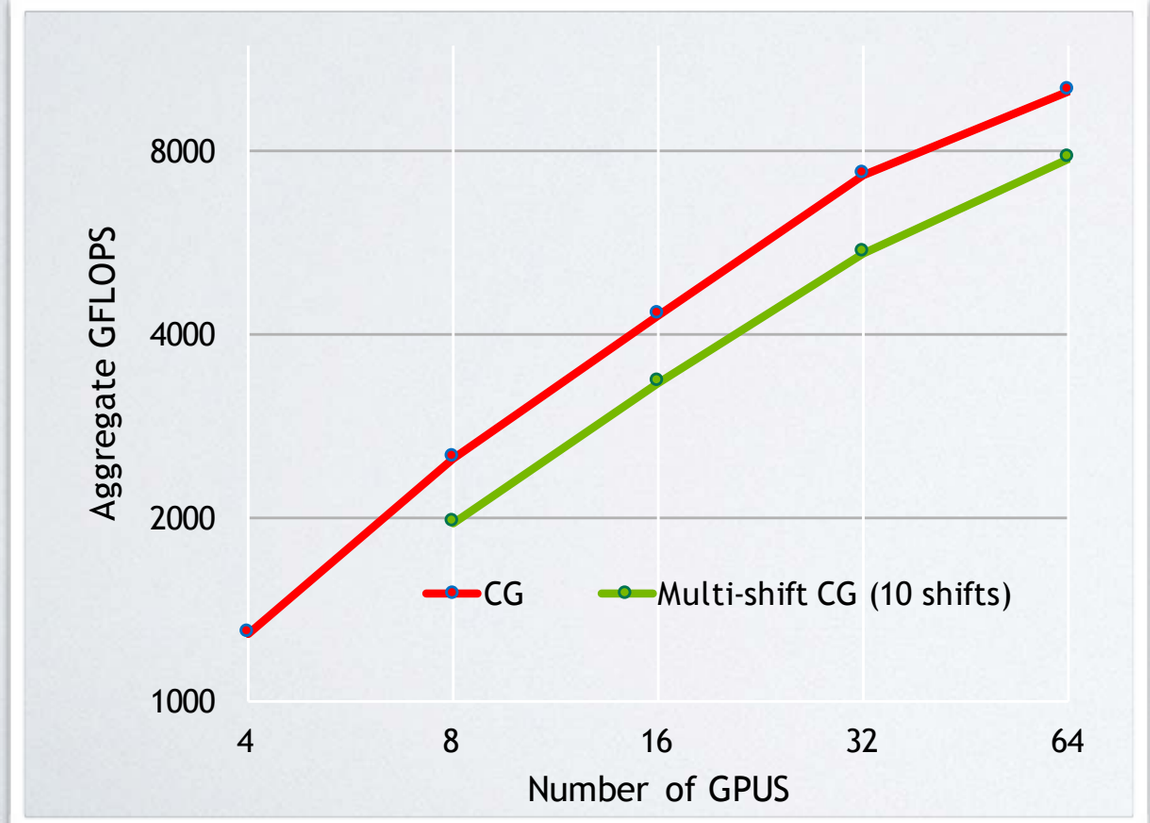


Volume per GPU 24^4x16, Mixed-precision Shamir, Saturn V

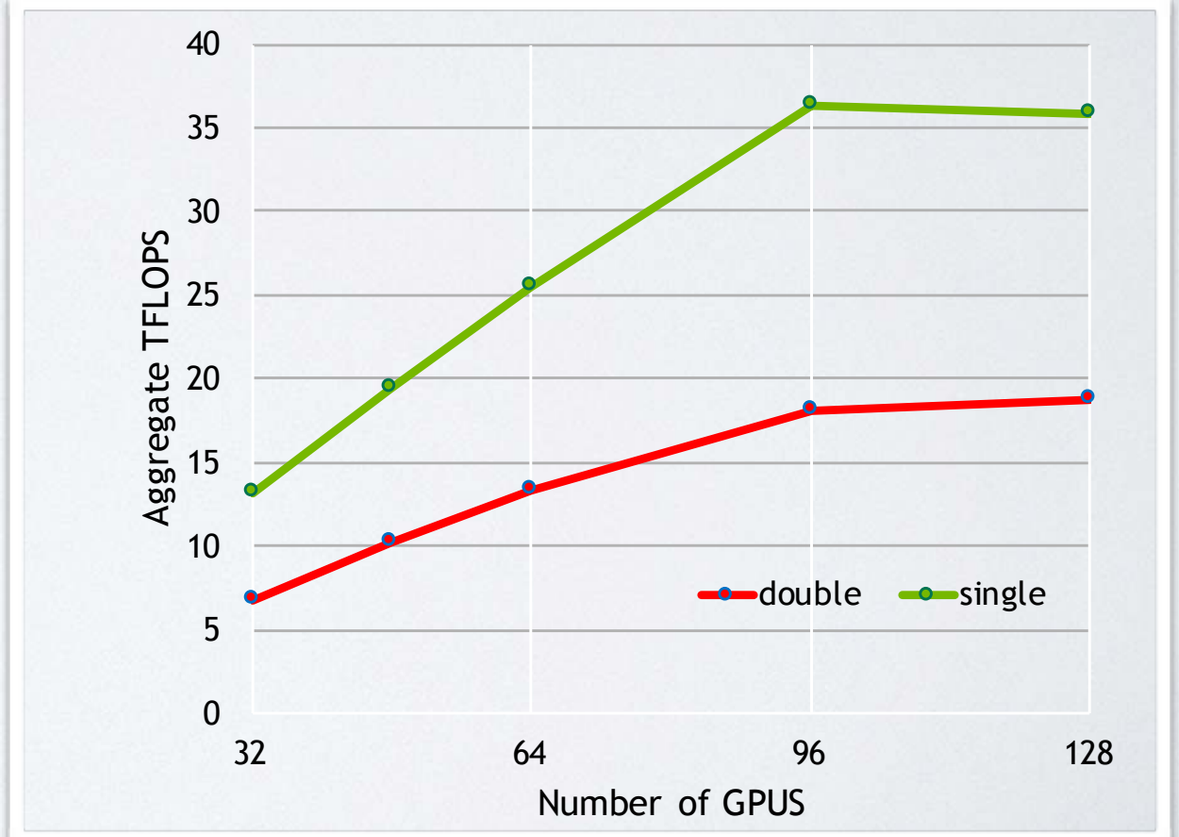
8x P100 GPUs connected through NVLink

4x EDR for inter-node communication

Optimal placement of GPUs and NIC

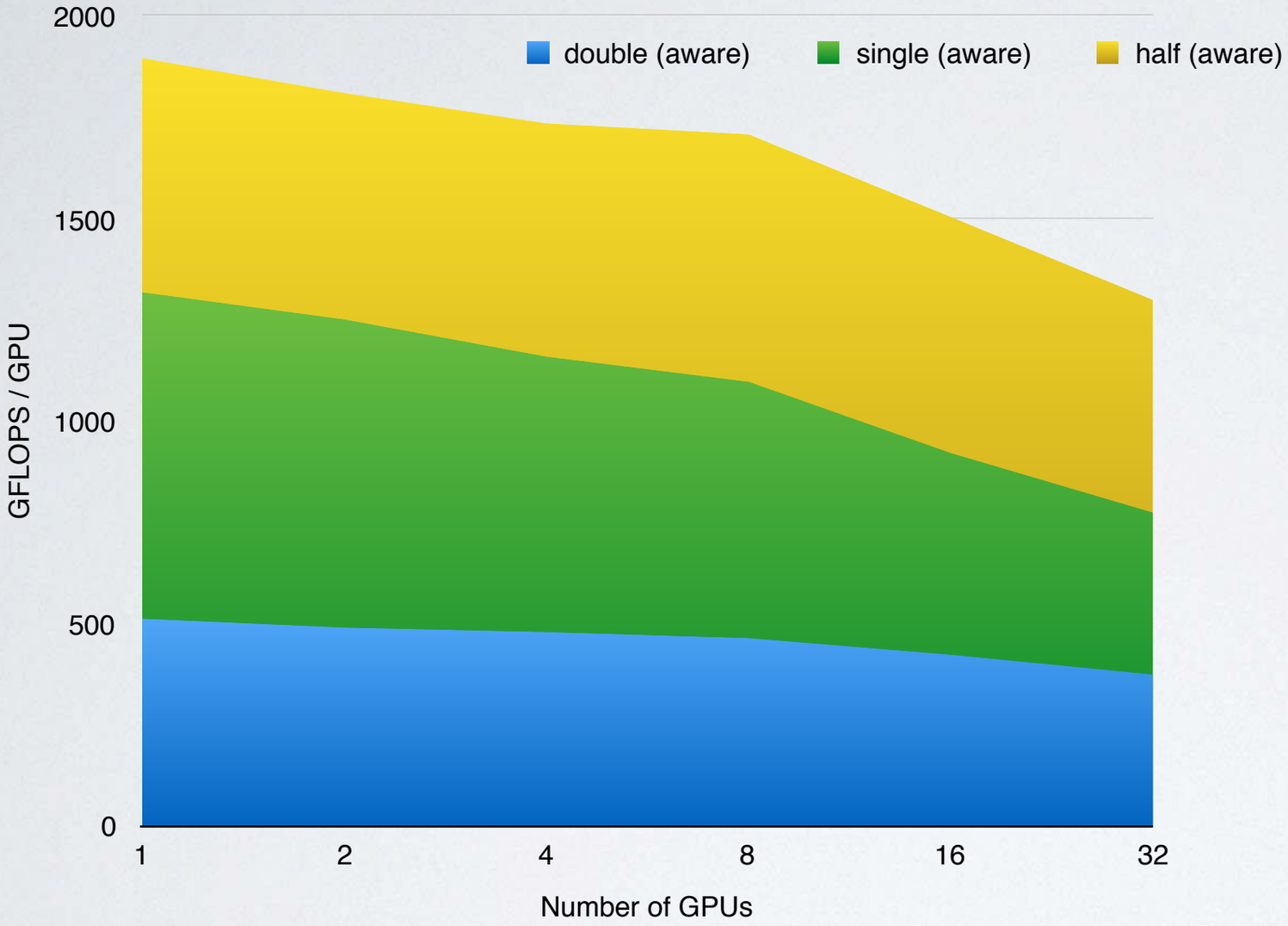


Quda Clover Strong scaling

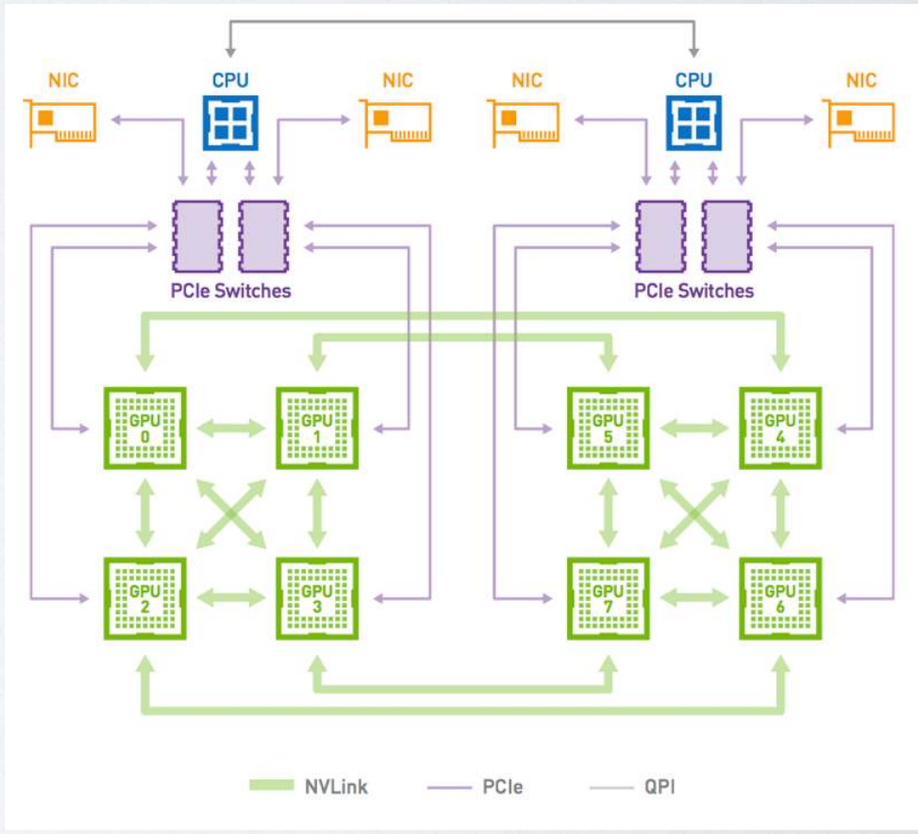


Quda HISQ Strong scaling

STRATEGY FOR MULTI GPU SCALING



- Direct peer-to-peer communication through NVLink
- GPU Direct for inter-node communication
- Topology aware communication



24^4 per GPU weak scaling benchmark
(Ls=16 domain wall Shamir operator)

Data from Pascal

Performance Portability Strategies for Grid C++ Expression Template

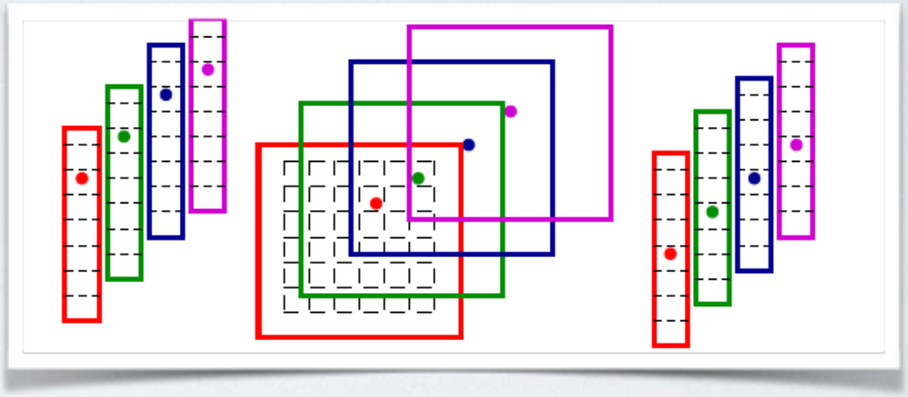
Lin, Boyle, Clark, DeTar, Rana, Vaquero Avilés-Casco

Port the Grid library to GPUs in an "effortless" way.

Strategy is to map:

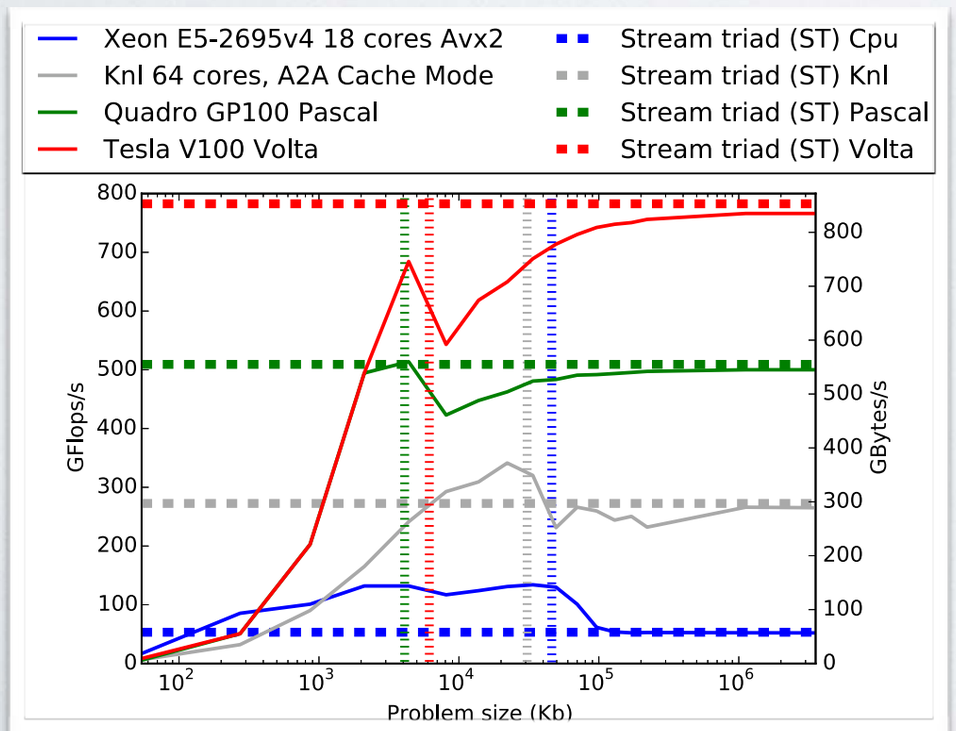
block ➡ vector

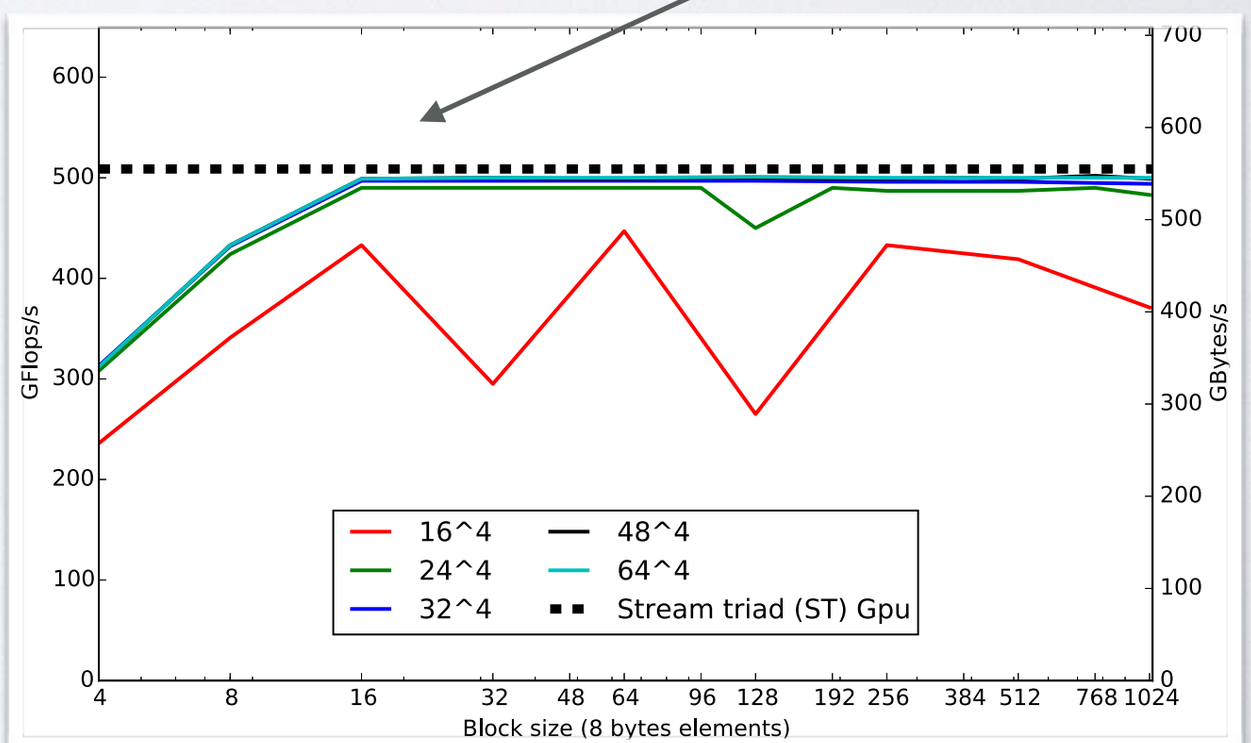
threads ➡ elements in a vector



SU(3) multiplication test: A=BxC

1024 bit





Novel Hardware: Intel Omni-Path

Omni-Path is a high-performance communication architecture.

- Low communication latency, low power consumption and a high throughput.
- Designed for MPI
- Scalable All-To-All communication
- Network functions executed by CPU
- It will serve as the on-ramp to exascale computing

In our community:

Single rail OPA: OakForest-PACS, QPACE 3, Jlab, Marconi

Dual rail OPA: BNL

OMNI-PATH CONTRIBUTION LIST:

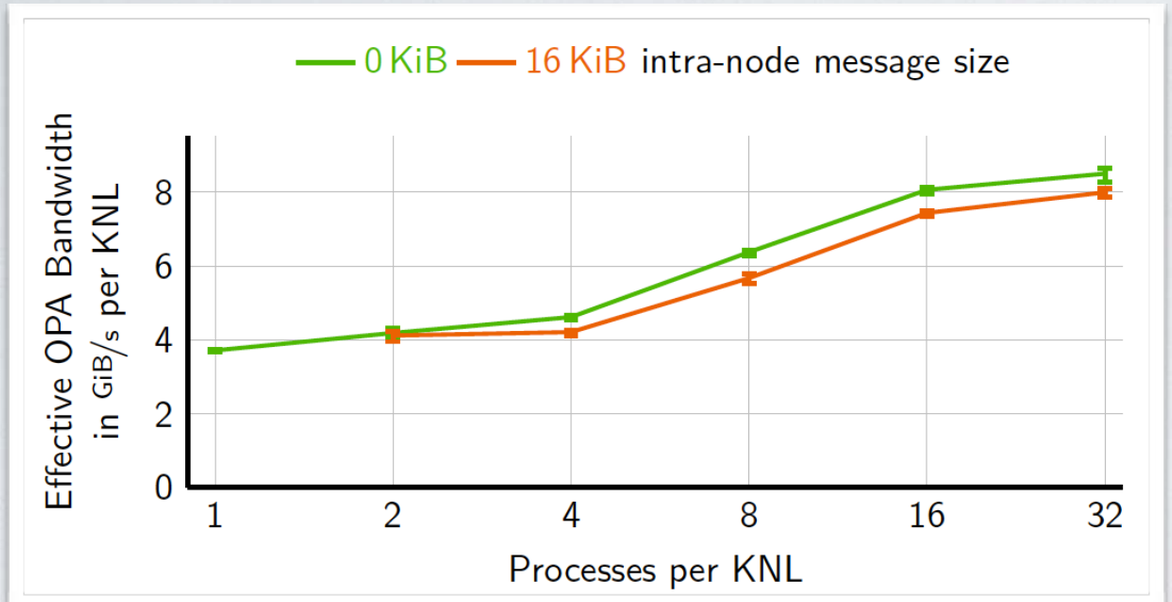
- Georg, Richtmann, Wettig (Monday, 19 June 2017 17:00 & Poster)
  An in-depth evaluation of the Intel Omni-Path network for LQCD applications
  An implementation of the DD-α AMG multigrid solver on Intel Knights Landing

- Boyle, Cossu, Portelli, Yamaguchi (Cancelled)
  Grid software status and performance

- Boyle, Kelly
  Half Precision Communications

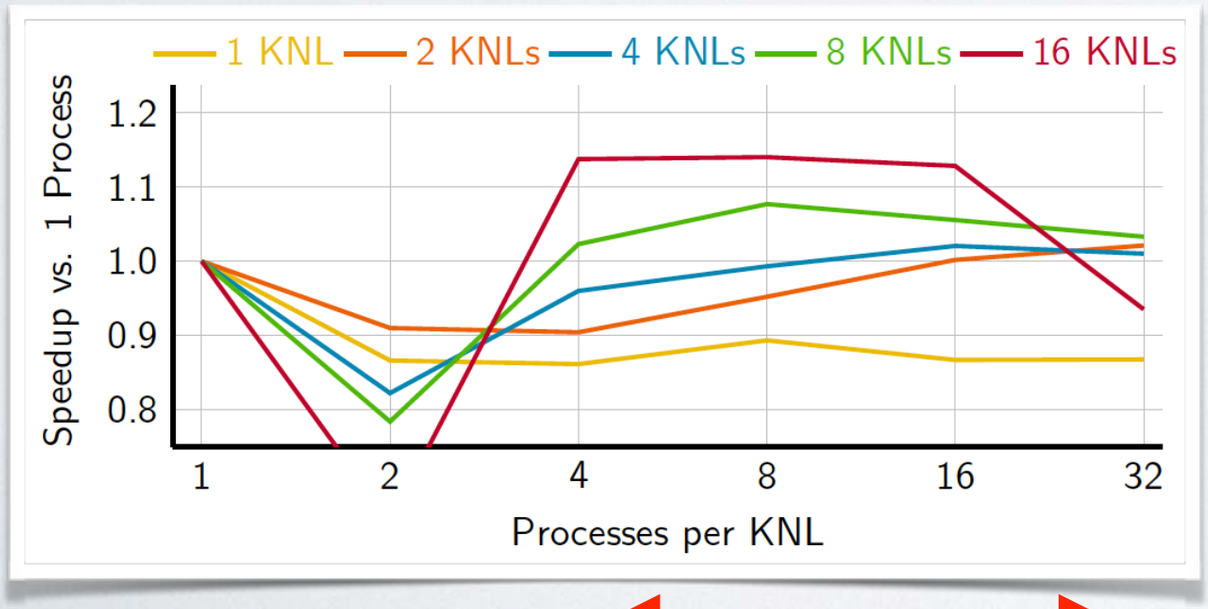
An in-depth evaluation of the Intel Omni-Path network for LQCD applications
An implementation of the DD-α AMG multigrid solver on Intel Knights Landing

Georg, Richtmann, Wettig



A synthetic halo exchange example:
Message size 512KiB/KNL
Peak BW 12.5 GiB/s

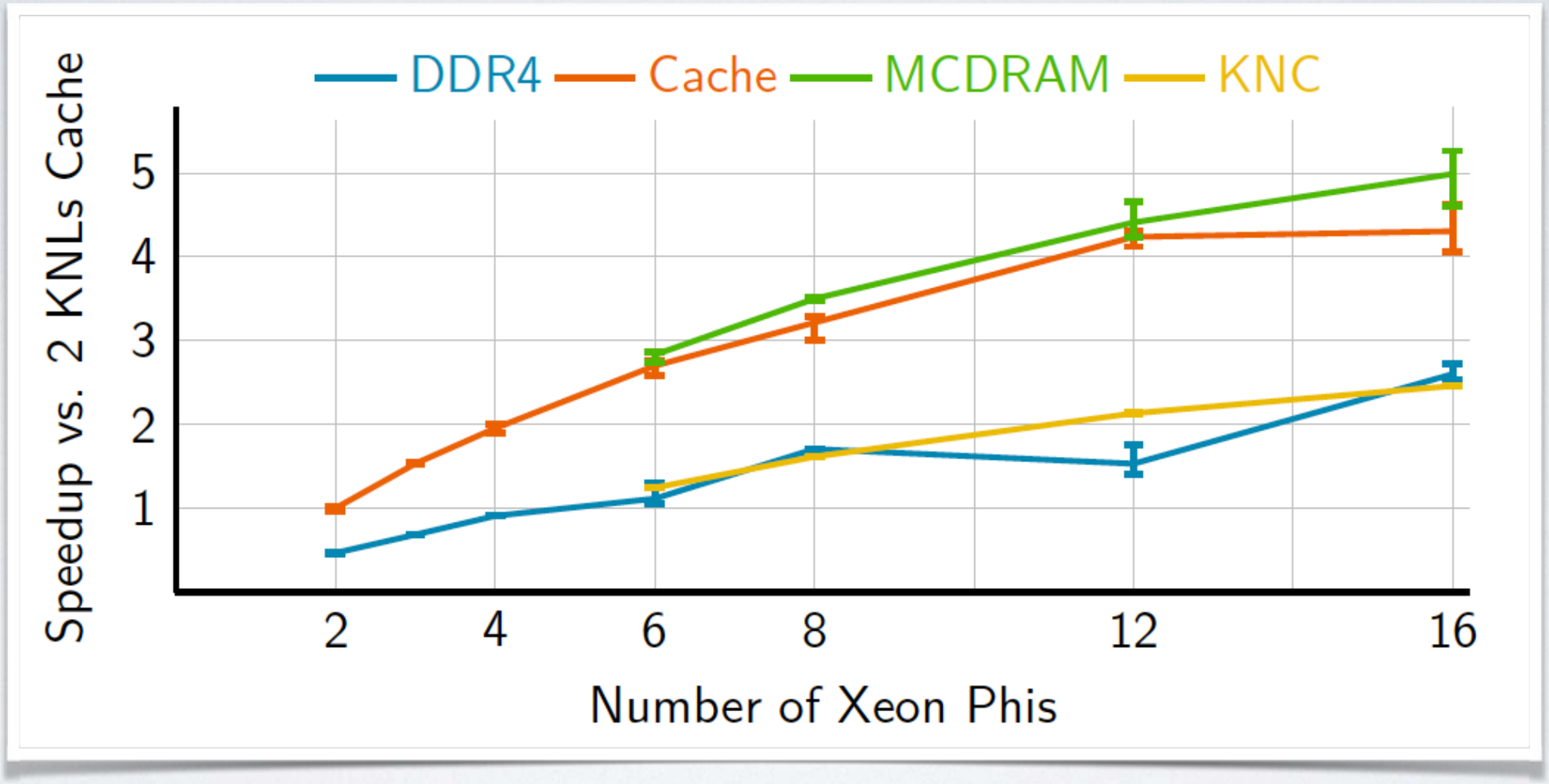
Solver DD-αAMG (two level multigrid) 32^3x96



SNC4

- Single node Performance decreases with number of process
- Multi node Performance increases with number of process

Solver DD-αAMG (two level multigrid) 32^3x96



- DDR4 with 100 Gbit/s OPA against KNC with 28 Gbit/s IB FDR
- Cache doesn't seem to scale beyond 12 KNL
- Using MCDRAM there is a 2.2 speedup over KNC

Grid software status and performance

Boyle, Cossu, Portelli, Yamaguchi

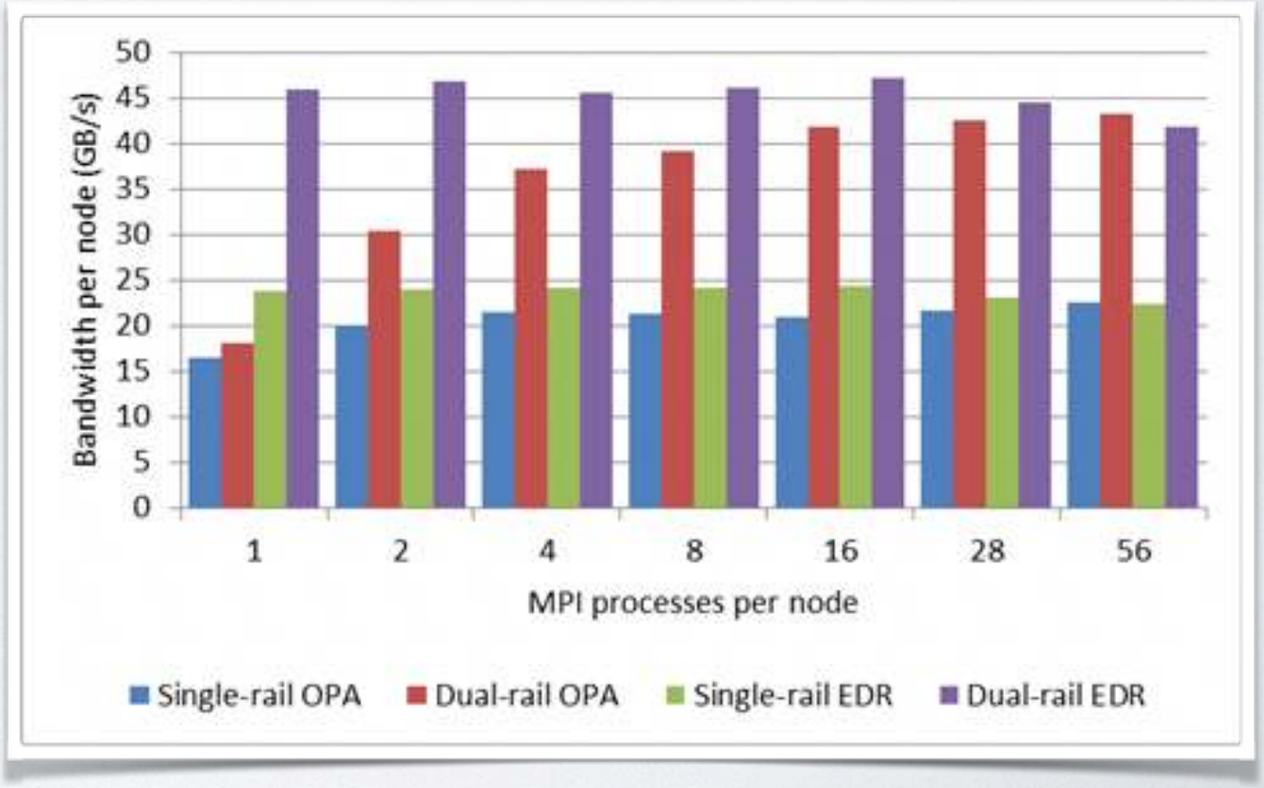
Grid HaloExchange (multiple sizes) on Intel-OPA, Mellanox-EDR and Cray-Aries

Machine	Node type	Network	Bidi peak GB/s	1 rank per node GB/s	4 ranks per node GB/s
Cori2	KNL	Aries	~16	11	11
BNL	KNL	Dual OPA	50	14	44
SGI ICE-X	Xeon	Dual EPR	50	44	44

OPA vs EDR

Comparison between Intel OPA and Mellanox EDR dual rail on SGI ICE-X Xeon E5-2690 2.4Ghz nodes.
Only Best Performances reported

courtesy: SGI HPE

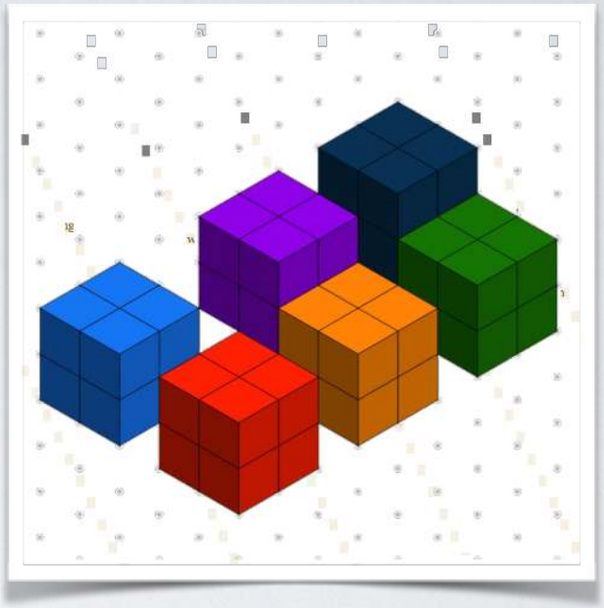


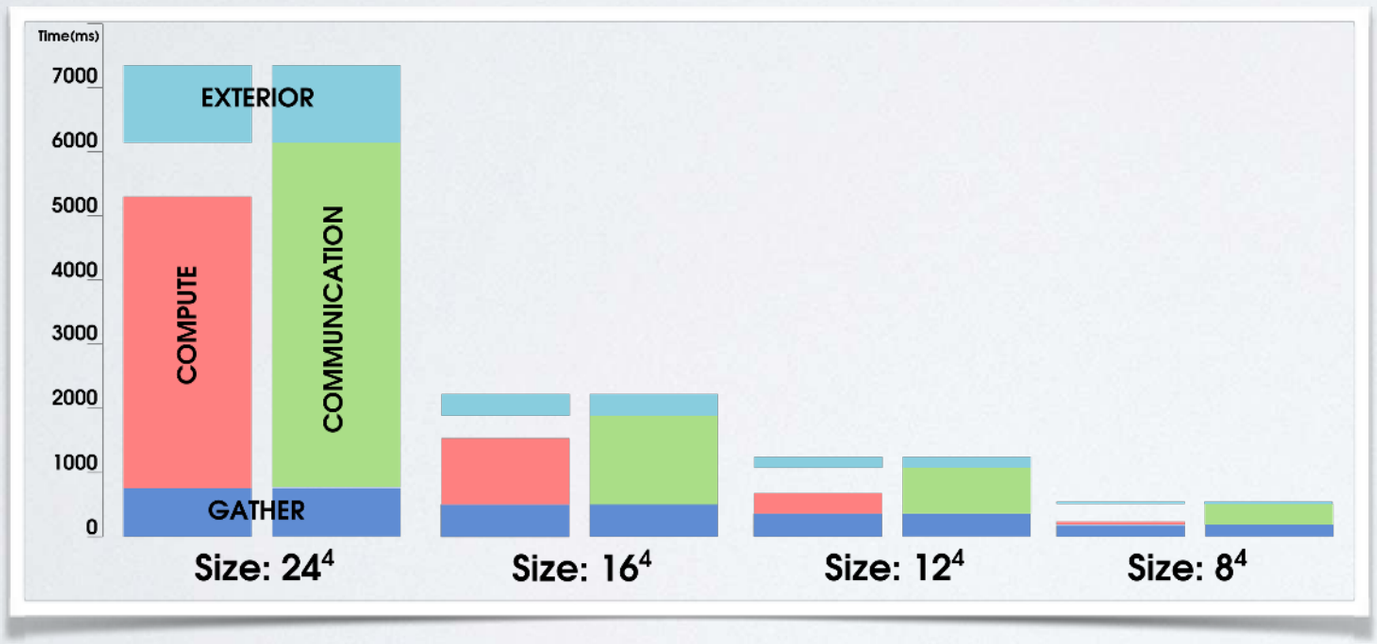
Cartesian mapping: a possible workaround

Specific for Omni-Parth:

MPI-3 to accelerate intranode comms

- Carefully ensure ranks on same node (e.g. consecutive) assume cartesian coordinates in cubes
- Maximises interior fast communications with multiple ranks per node
- Perform comms by direct copy into SHM and no interior MPI calls.





Local vol	GF/s compute	GF/s multinode
24^4	770	450
16^4	880	330
12^4	880	225
8^4	880	110

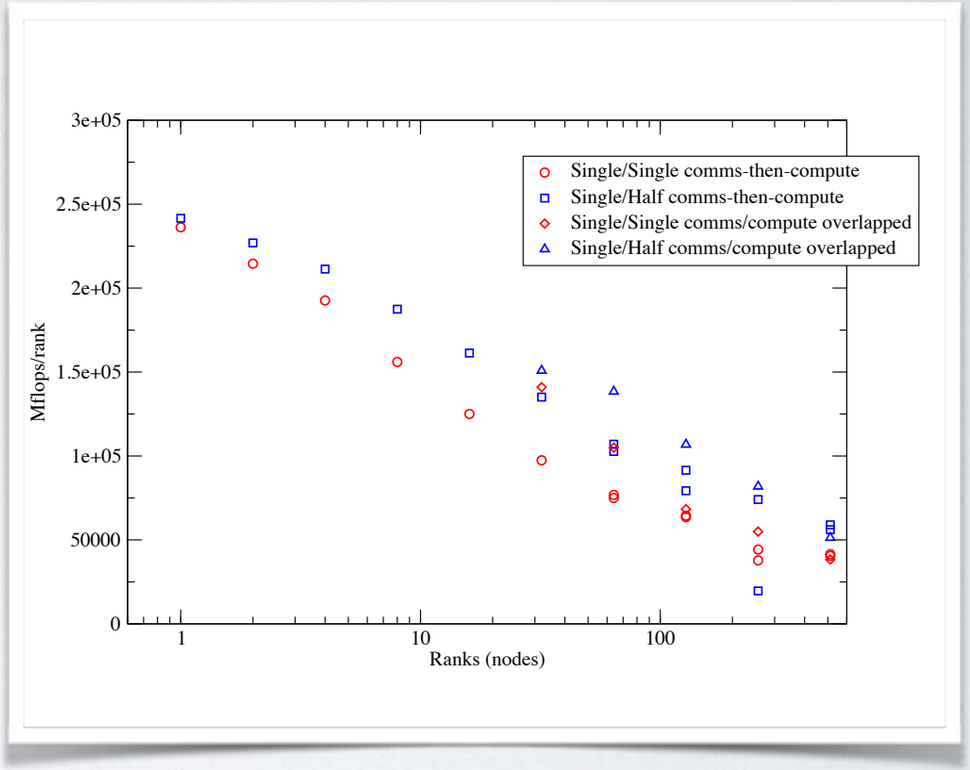
Communication dominates on local volumes.
Various improvement are in progress (one on next slide)
10-20% further gain and over 500GF/s per node on 24^4 and 16^4 local volumes.

Half Precision Communications

Kelly, Boyle

...a more generic workaround for Interconnection

- Performing comms of inner single-compute CG in half-precision comms
- Numerical test (quick and dirty) compress/ decompress buffers (adds cost)
- Demonstrated that for right inner CG tolerance, matrix applications can be kept roughly constant
- algorithm surprisingly tolerant to damaging neighbour information



Strong-scaling performance of Grid on Cori II Volume $32^3 \times 64 \times 12$ for Dslash of DW action

Compute type	Comms type	Comms (µs)	Compress (µs)	Decompress (µs)
Double	Double	2200	-	-
Single	Single	1200	-	-
Single	Half	750	230	190

CONCLUSIONS

- Data Motion, Memory and Cache bounds are defining the performances of our codes.

- CPU evolution (like Intel Skylake) will include larger vector units (AVX512). Automatic vectorisation will take advantage of it (EX. Durr's implementation).

- Intel OPA problem seems to be software related:
  We wait (and hope) for a new version of the PSM2 driver.

- Very good performances of NVIDIA Volta and NVLINk.