The experience of the HLST on Europes biggest KNL cluster

Tamás Fehér, Serhiy Mochalskyy, Nils Moschüring
Roman Hatzky
Intel MIC Programming Workshop
LRZ

Marconi – KNL at CINECA, Bologna

Total number of KNL nodes: 3600

Partition dedicated to the EUROfusion community: 392 (144 flat / 248 cache mode)

-> about 1 Pflop/s

Photo: F.Pierantoni

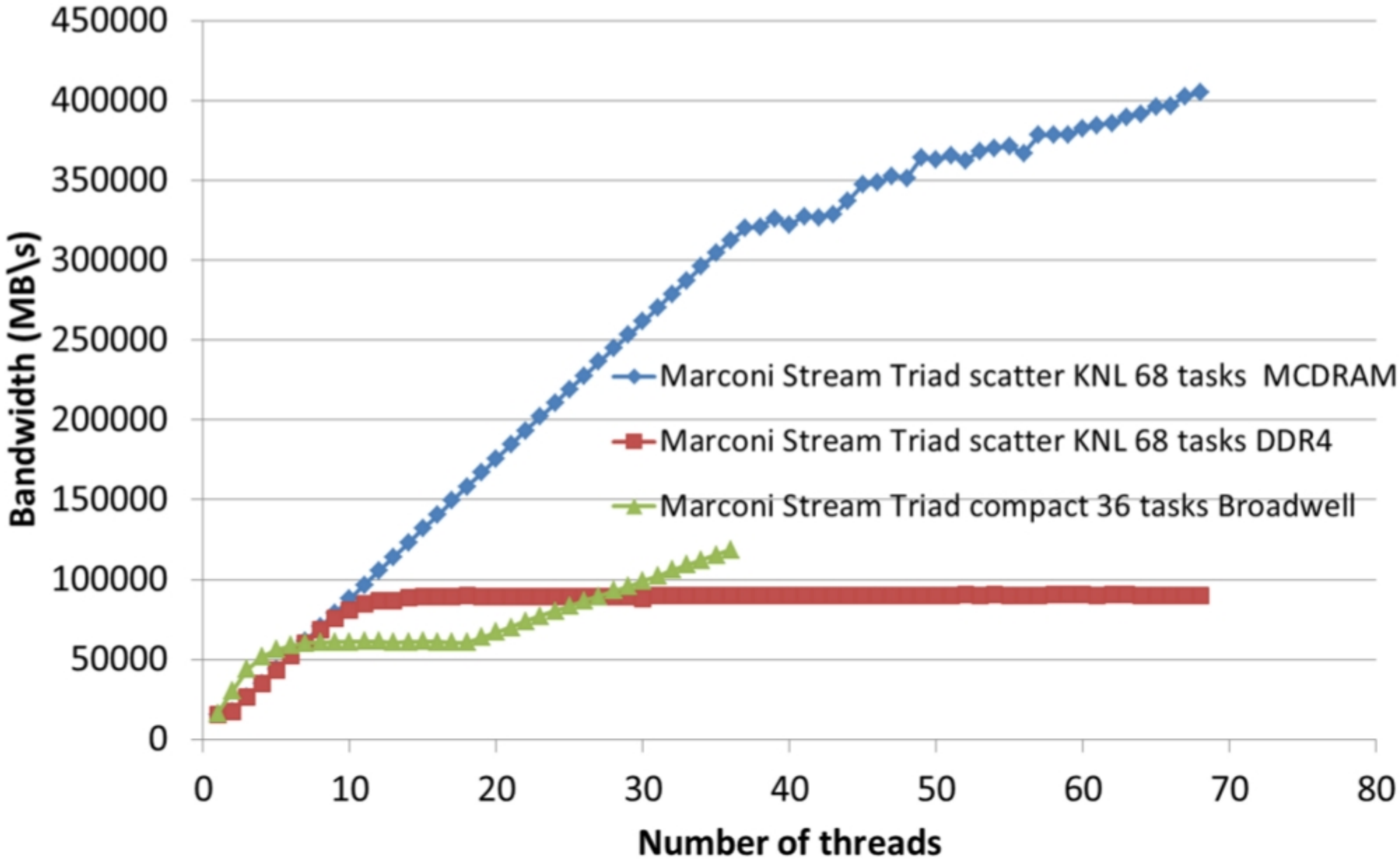
Overview

- Memory Bandwidth benchmarks
- Latency benchmarks
- OpenMP Benchmarks
- Code Performance
- Summary

STREAM and IMB

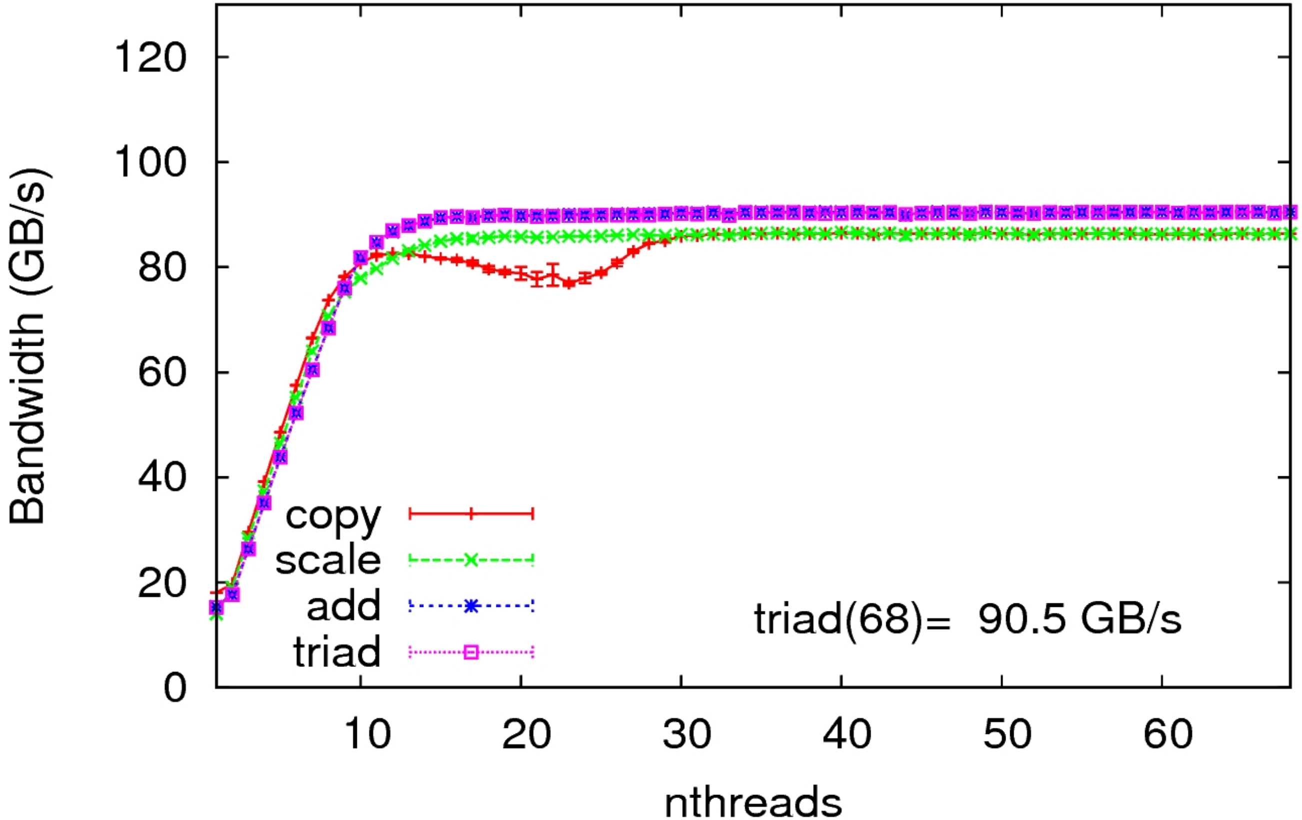
MEMORY BANDWIDTH BENCHMARKS

STREAM Memory Bandwidth different architectures

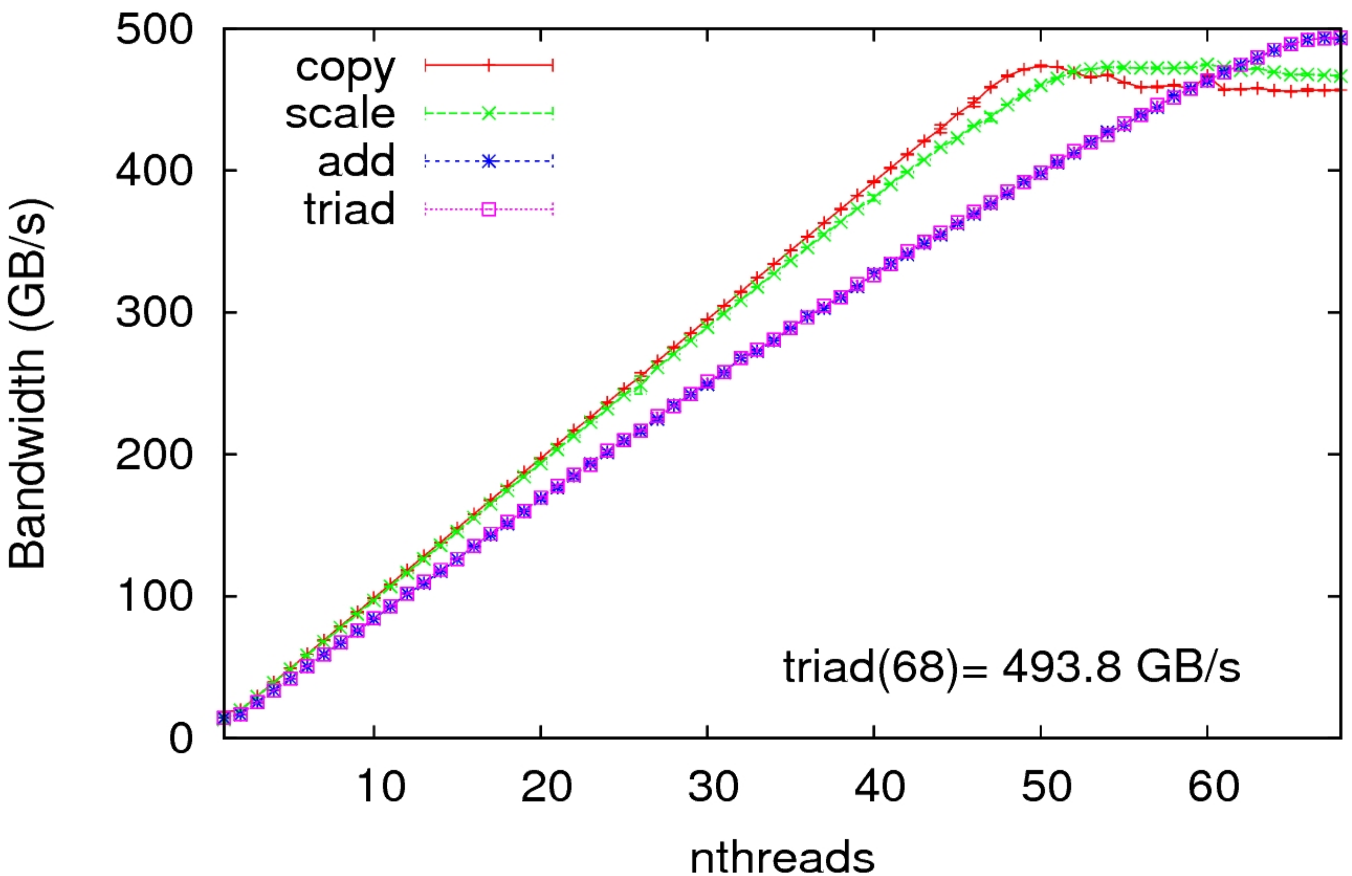


Using the STREAM benchmark by John D. McCalpin, https://www.cs.virginia.edu/stream

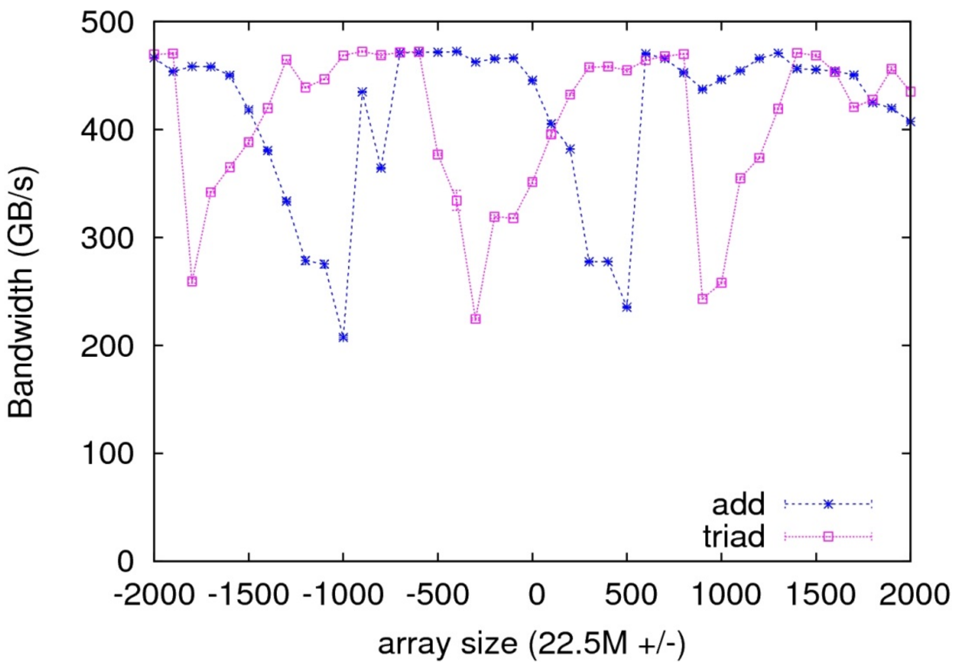
STREAM Memory Bandwidth flat mode DDR4

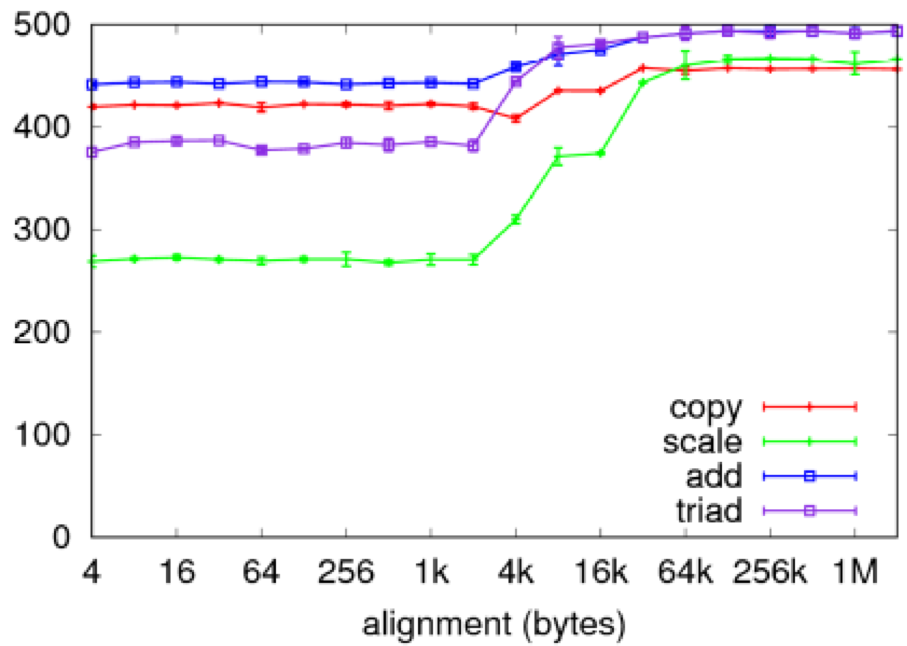


STREAM Memory Bandwidth flat mode MCDRAM



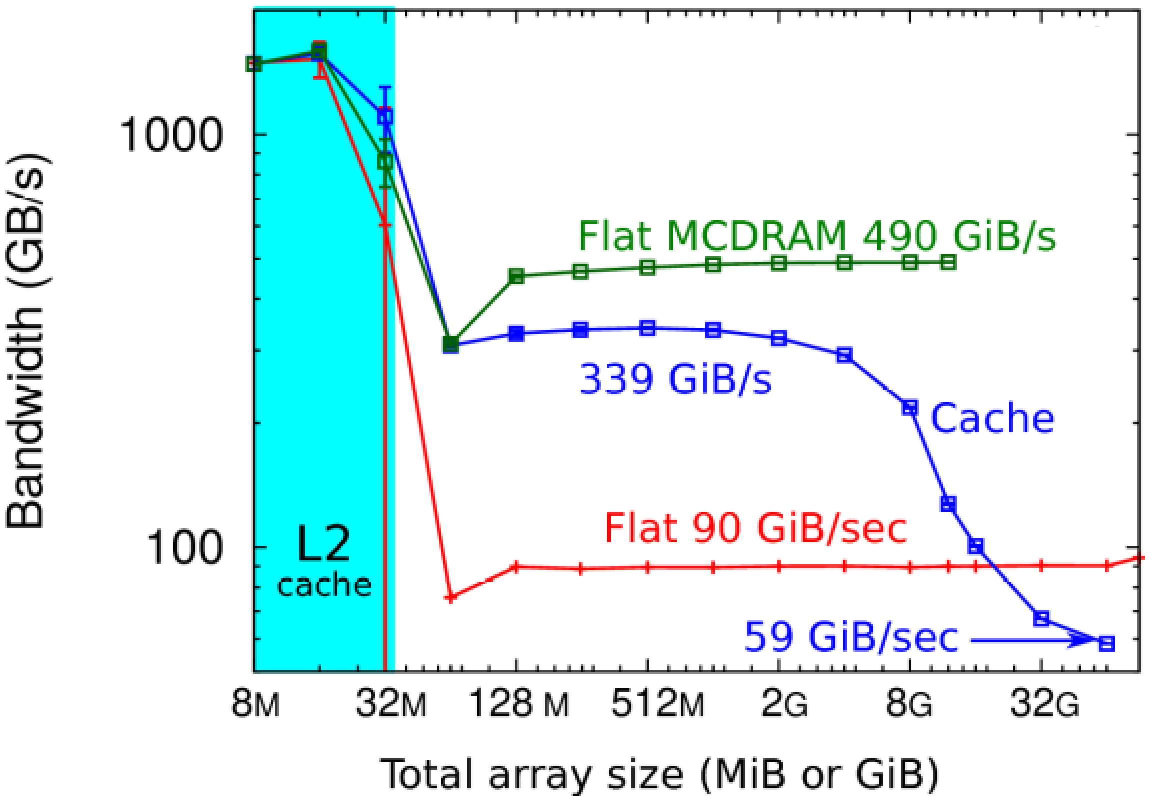
STREAM Memory Bandwidth flat mode MCDRAM - alignment



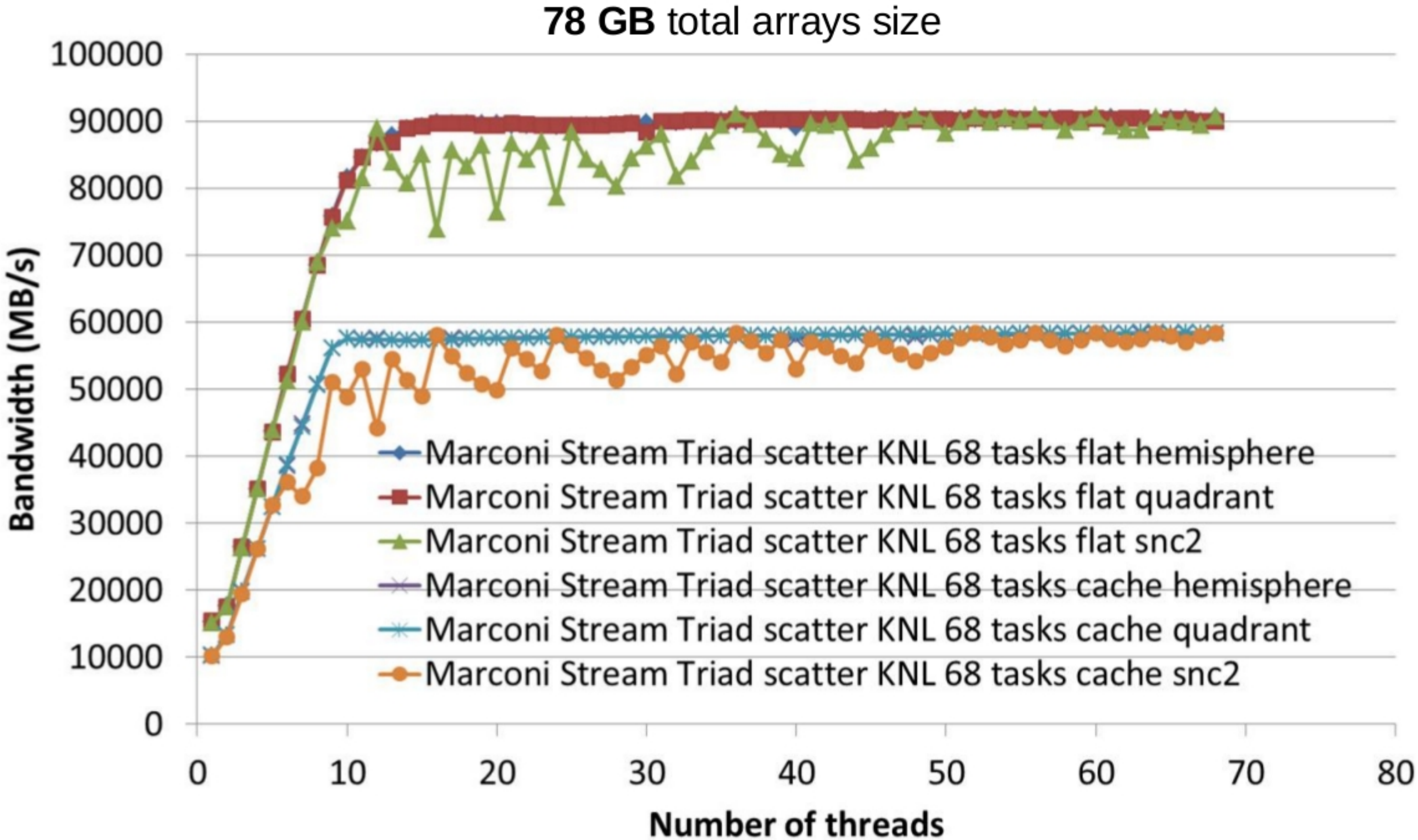


Alignment also important for cache mode

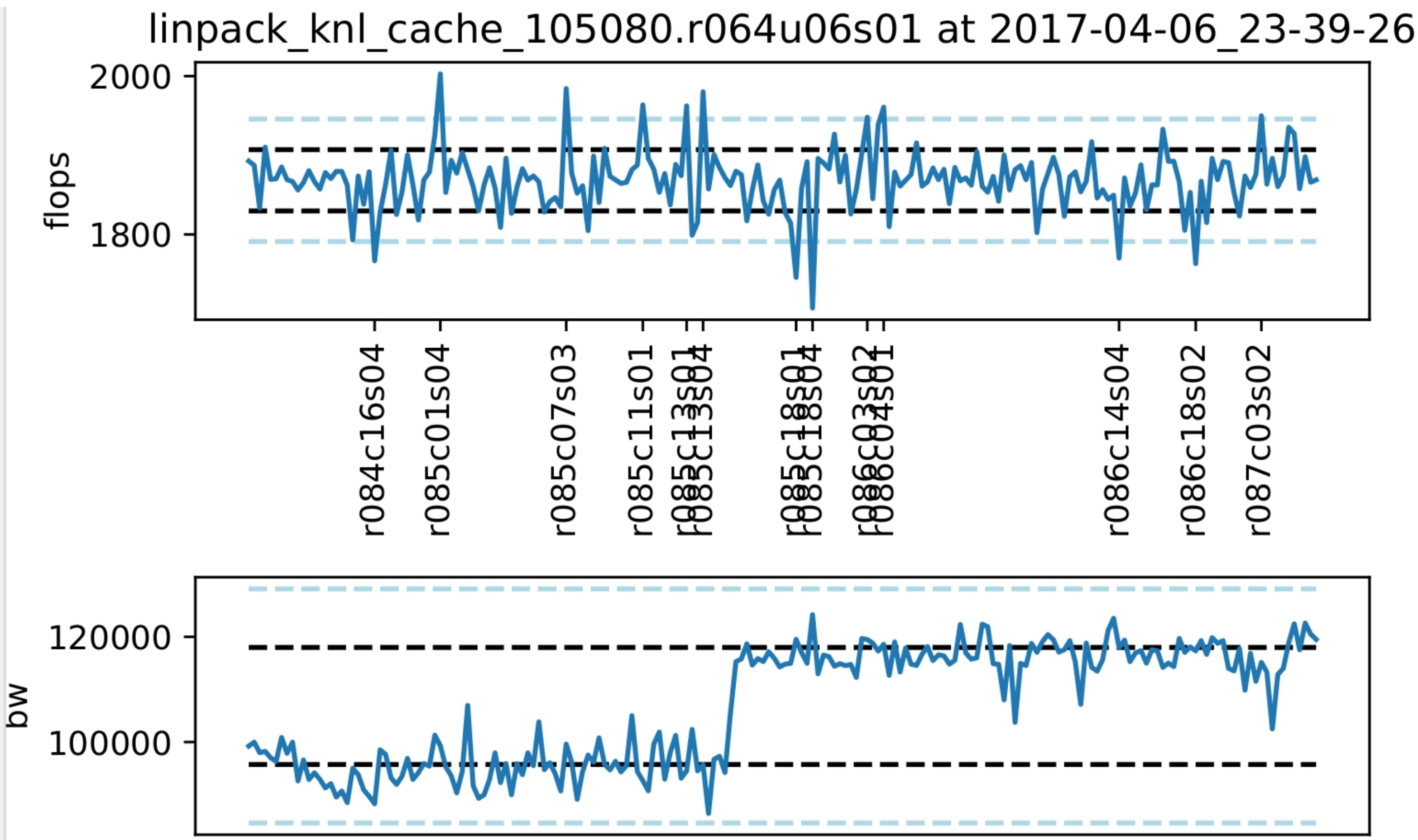
STREAM Memory Bandwidth
Array size



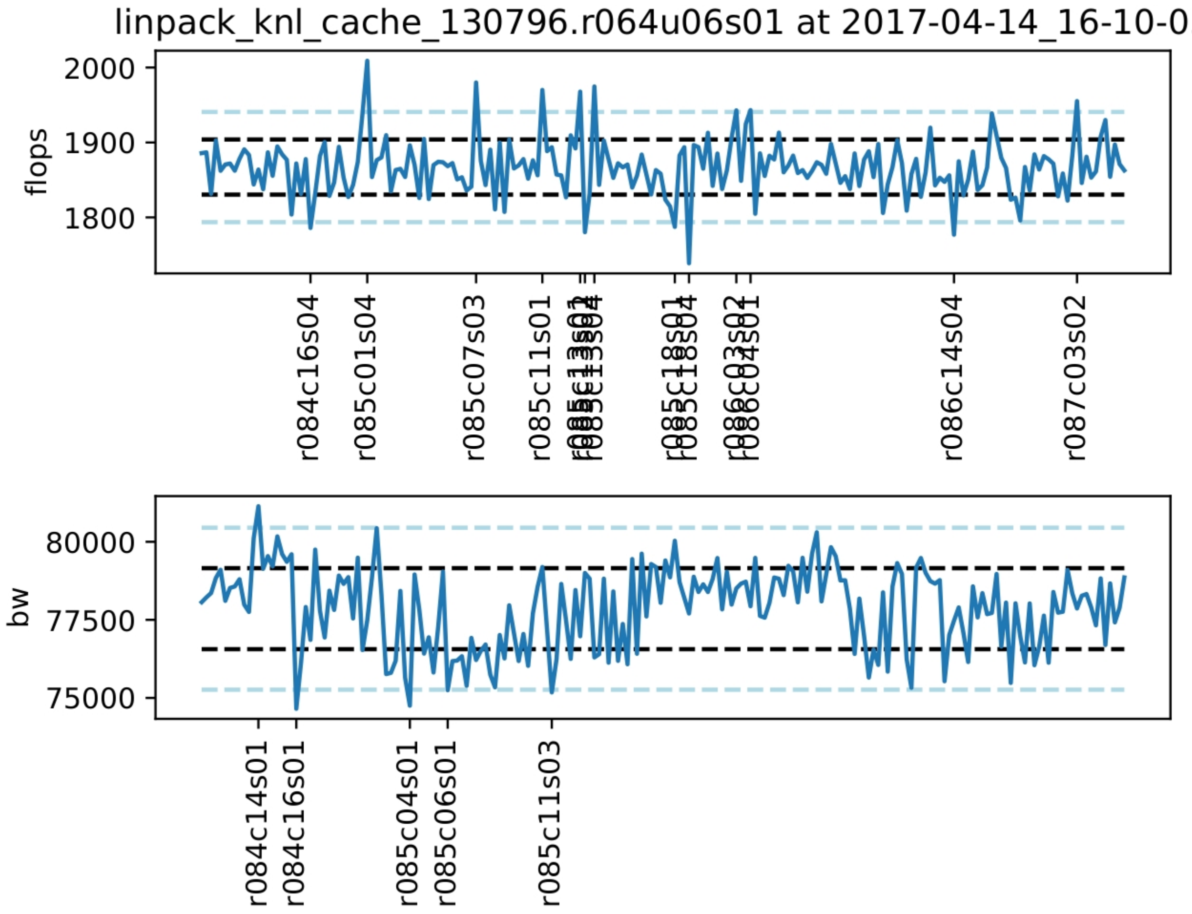
STREAM Memory Bandwidth cache versus flat



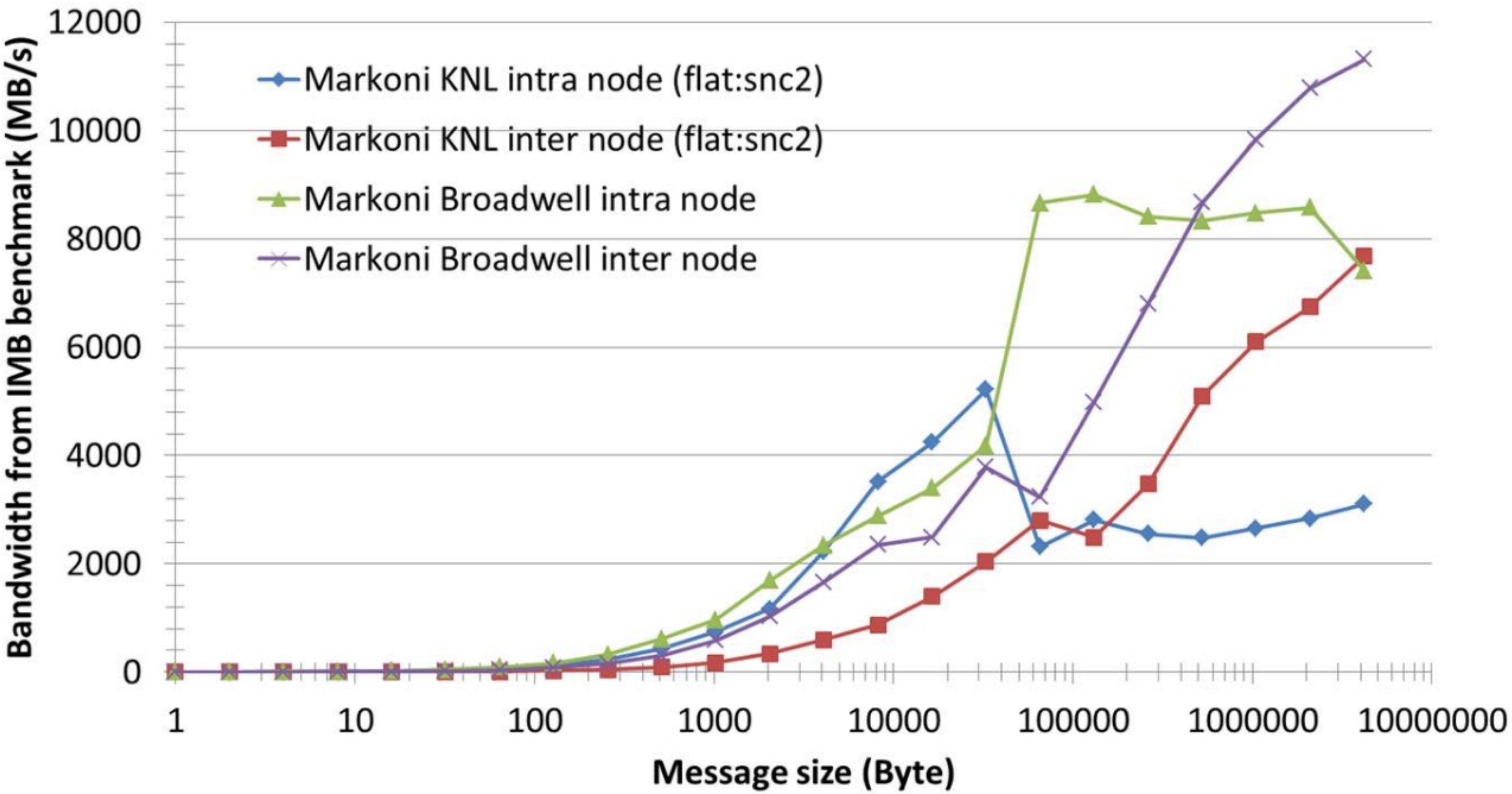
STREAM Memory Bandwidth over time



STREAM Memory Bandwidth over time

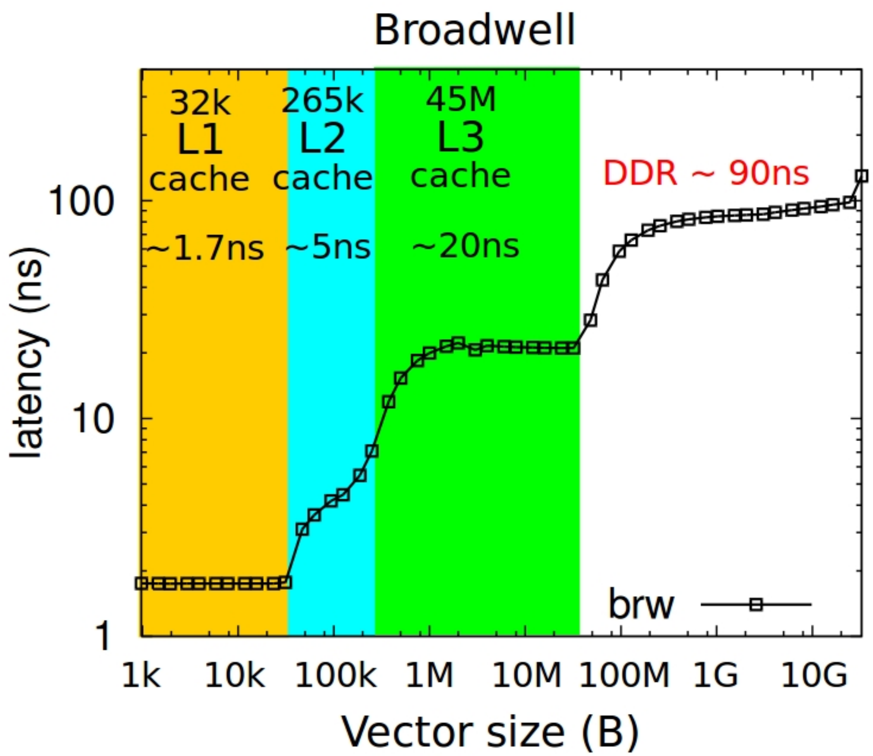


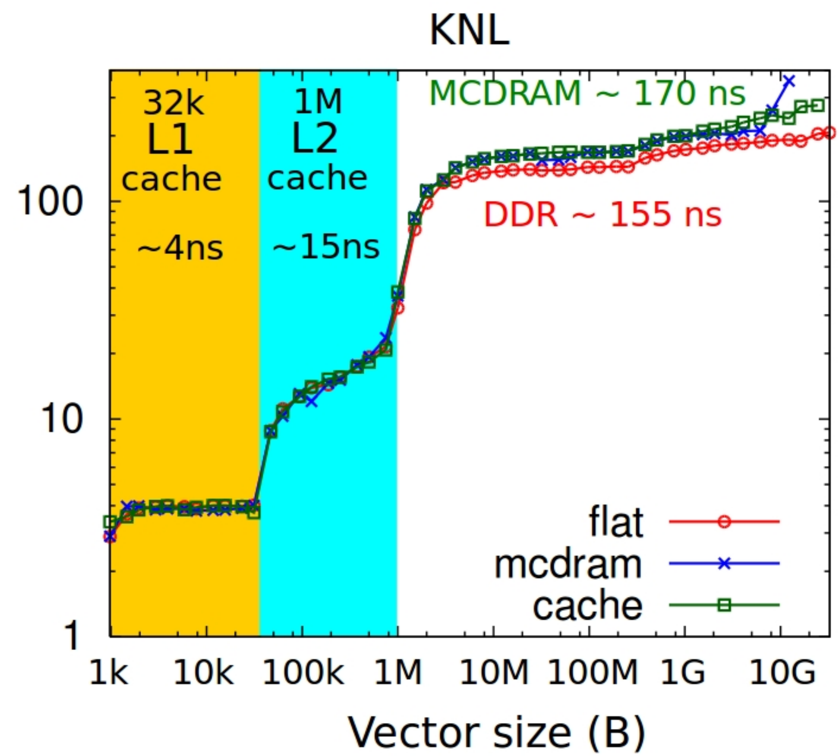
Intel MPI benchmark - bandwidth



LATENCY BENCHMARKS

Latency


Broadwell
32k L1 cache ~1.7ns
265k L2 cache ~5ns
45M L3 cache ~20ns
DDR ~ 90ns
brw
latency (ns)
Vector size (B)
1
10
100
1k 10k 100k 1M 10M 100M 1G 10G


KNL
32k L1 cache ~4ns
1M L2 cache ~15ns
MCDRAM ~ 170 ns
DDR ~ 155 ns
flat
mcdram
cache
Vector size (B)
1
10
100
1k 10k 100k 1M 10M 100M 1G 10G

IMB – Ping Pong Test - Latency

Intra node Marconi

Broadwell

node0	CPU0	0.61
	CPU1	1.09
Latency (μs)		CPU0
		node0

Knights Landing

node0	KNL0	0.85
Latency (μs)		KNL0
		node0

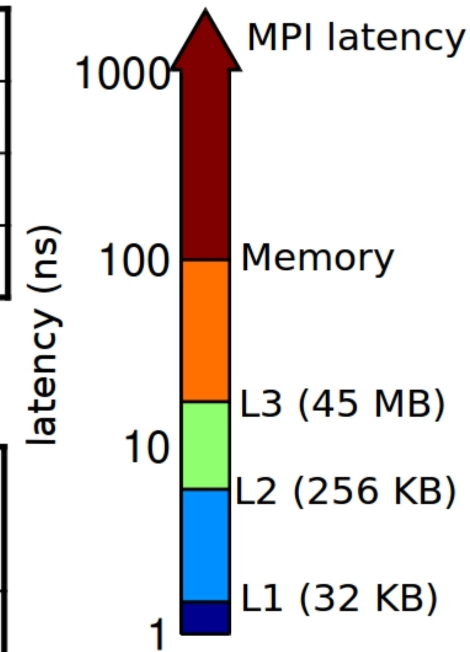
Intra node HELIOS

Sandy Bridge

node0	CPU0	0.25
	CPU1	0.64
Latency (μs)		CPU0
		node0

Knights Corner

node0	KNC0	2.7
Latency (μs)		KNC0
		node0



IMB – Ping Pong Test - Latency

Inter node Marconi

Broadwell

node0	CPU0	1.49
Latency (µs)		CPU0
		node1

Knights Landing

node0	KNL0	3.99
Latency (µs)		KNL0
		node1

Inter node HELIOS

Sandy Bridge

node0	CPU0	1.13
Latency (µs)		CPU0
		node1

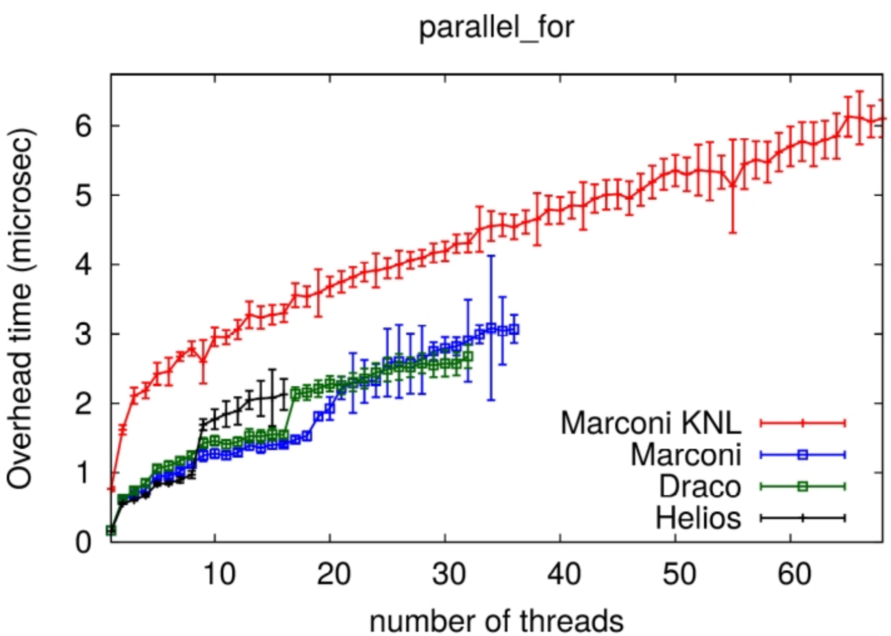
Knights Corner

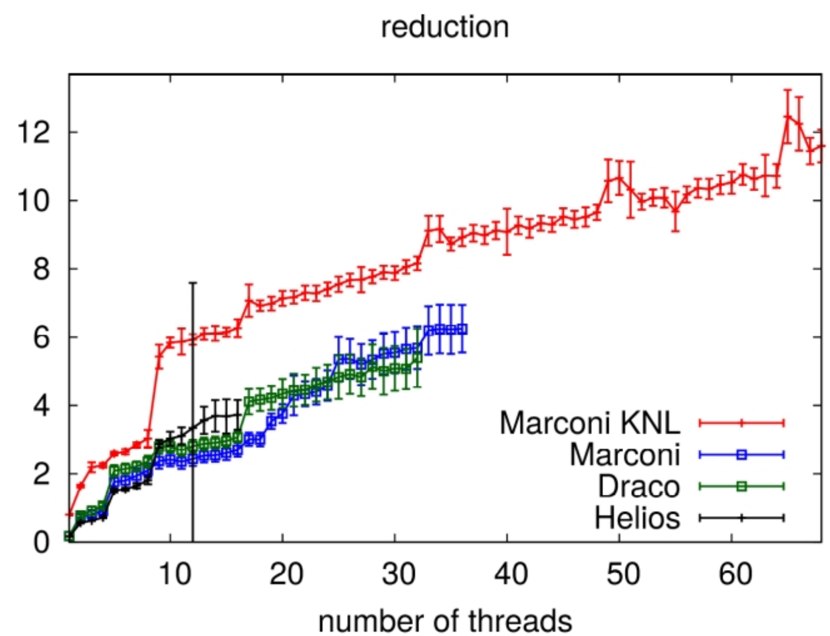
node0	KNC0	6.00
Latency (µs)		KNC0
		node1

OPENMP BENCHMARKS

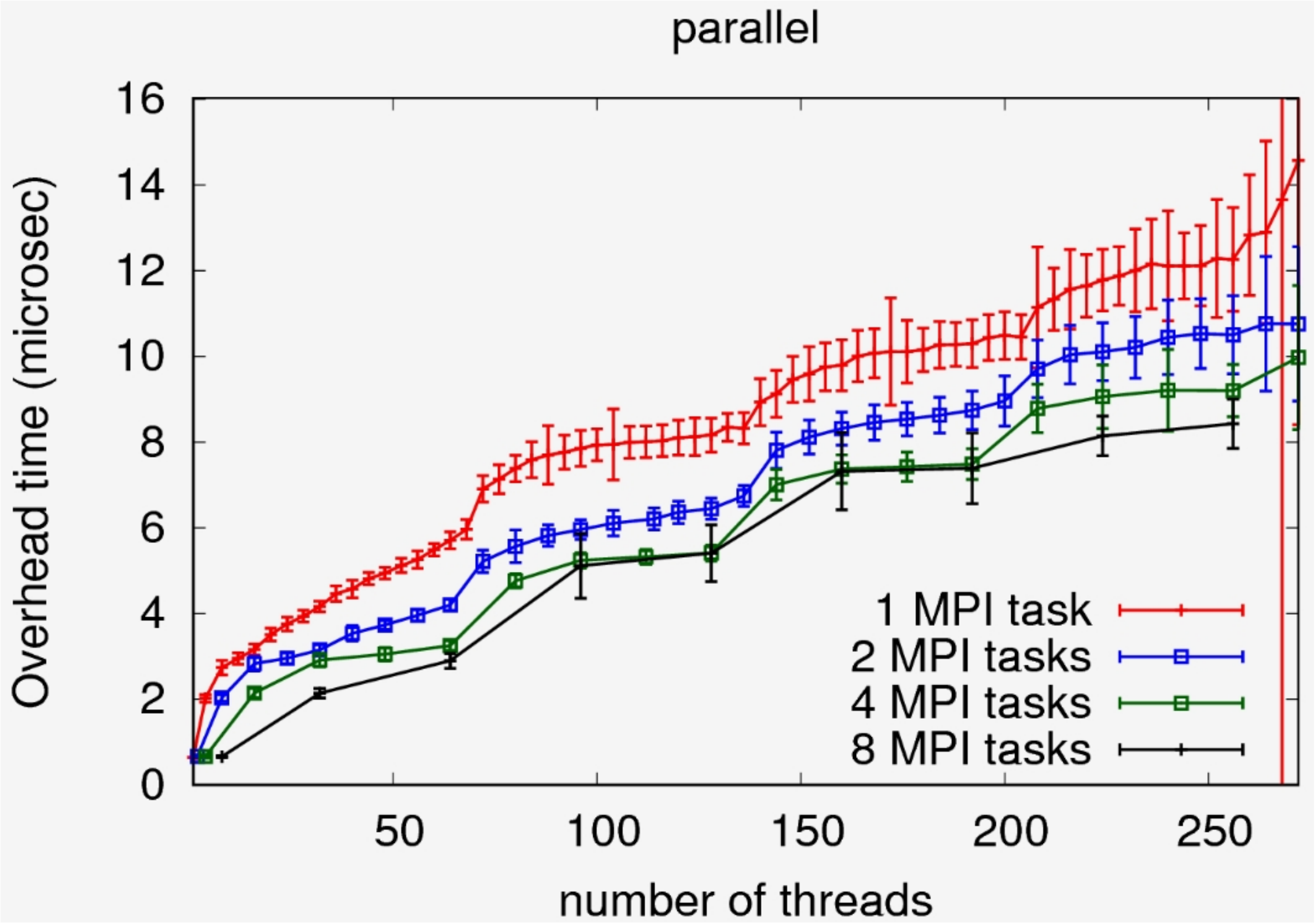
OpenMP overhead

- KNL overhead ≈ 2x larger:
  - more threads
  - lower CPU frequency
- Exception: ATOMIC 5x longer, use CRITICAL instead
- Using EPCC OpenMP Microbenchmarks J.M. Bull et. al



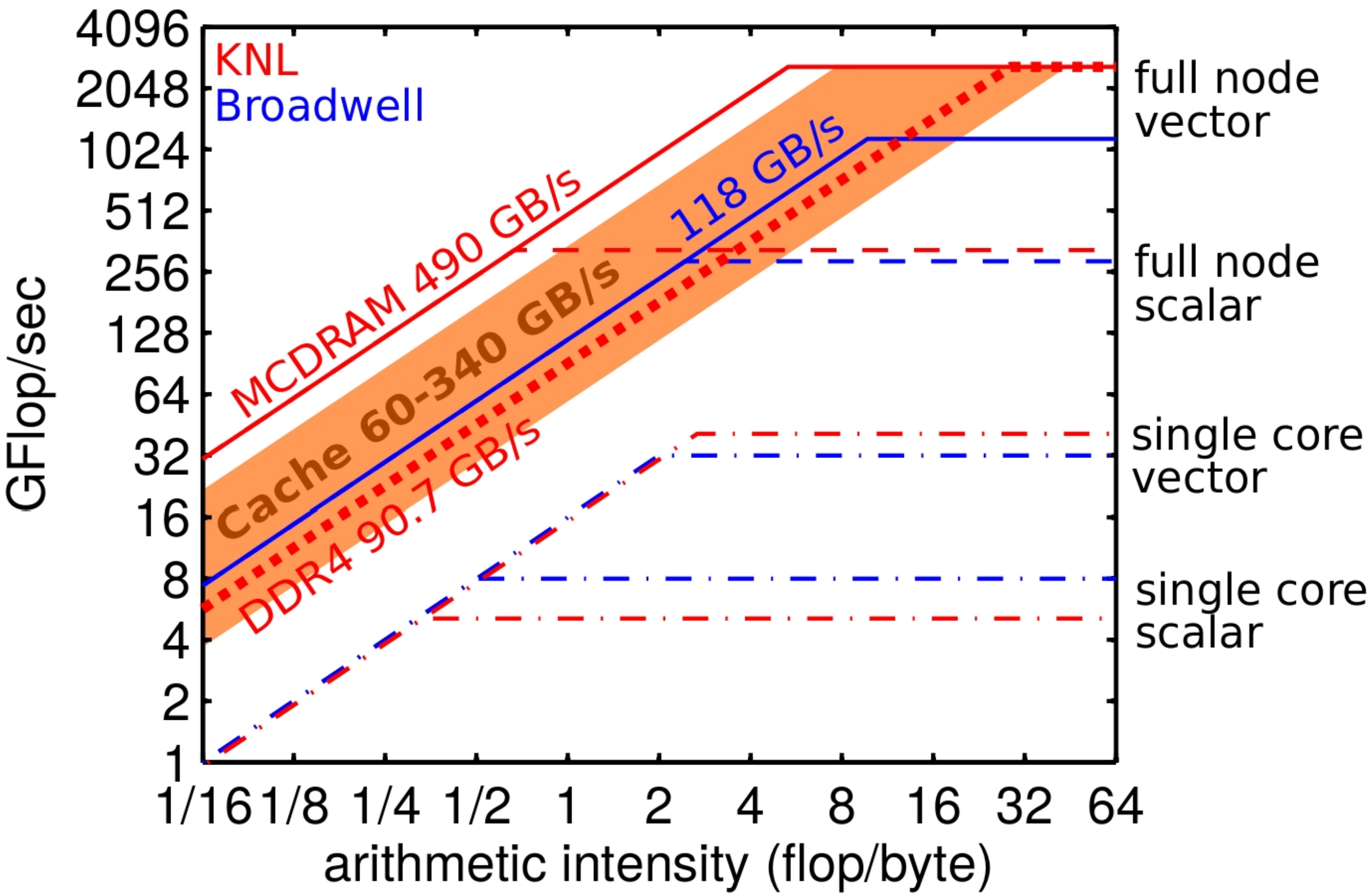


OpenMP overhead + hyper-threading



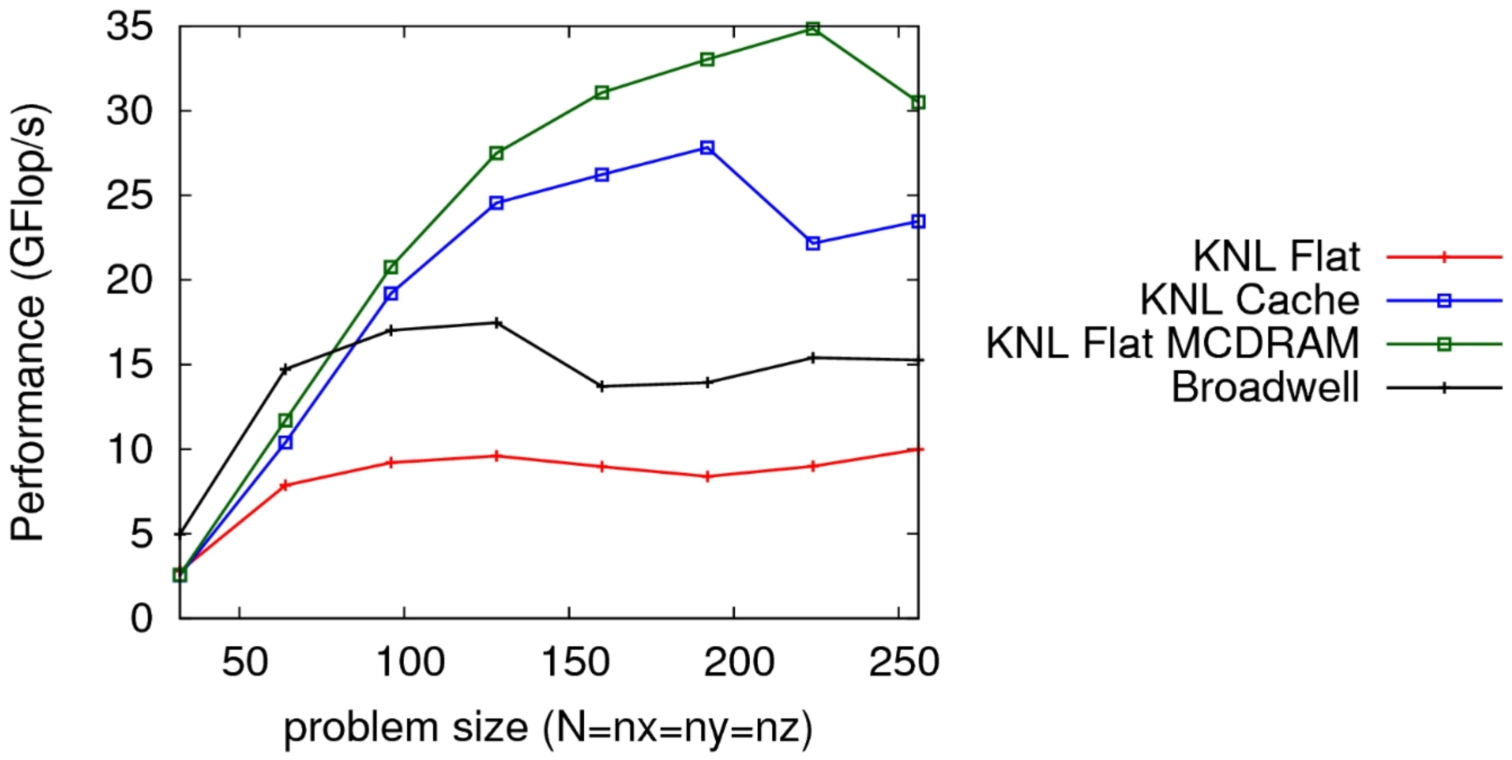
CODE PERFORMANCE

Roofline Model



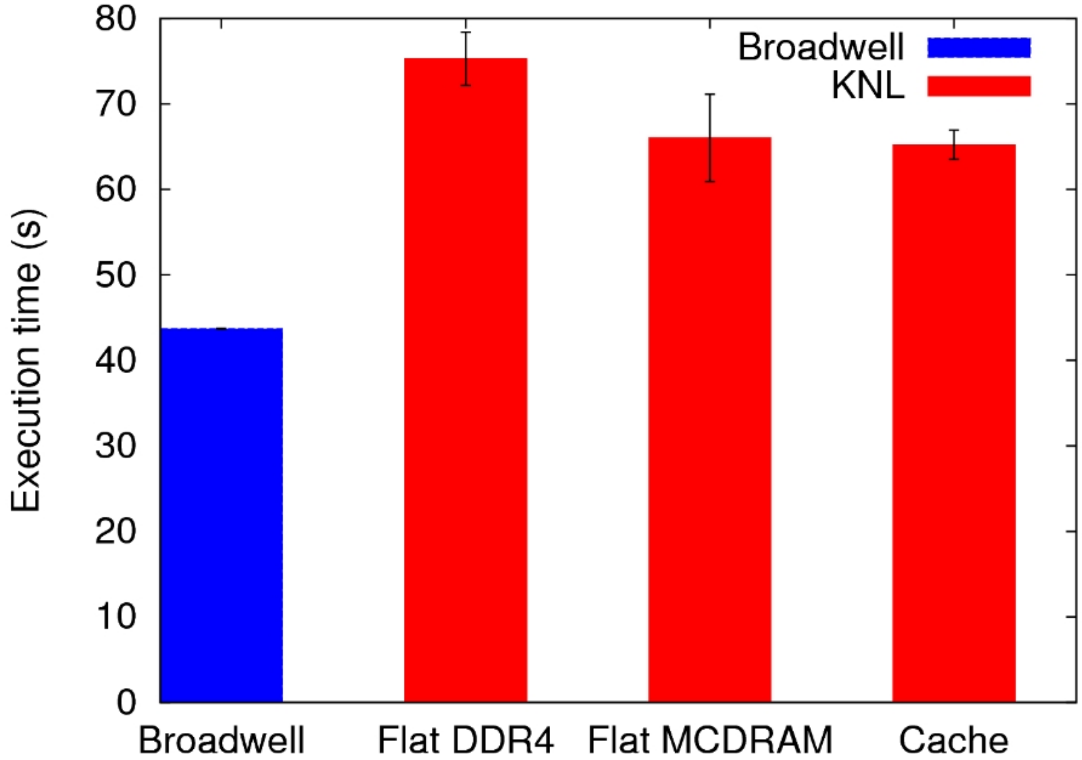
HPCG benchmark

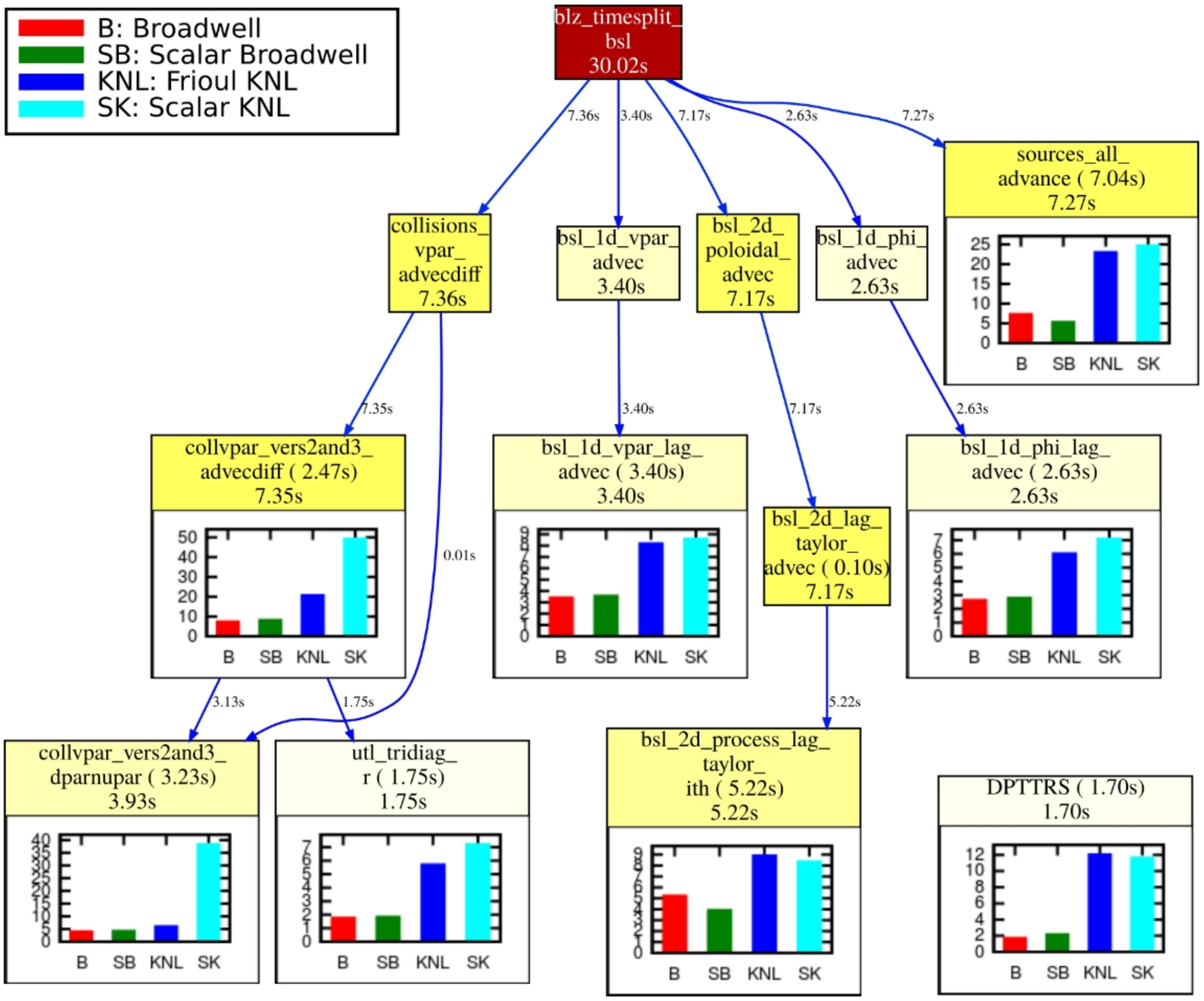
- HPCG: sparse 3D problem with multigrid preconditioned conjugate gradient solver.
- The Intel optimized version of the HPCG benchmark was executed in one node.



Gysela execution time

- Test case: 127 x 256 x 64 x 63 (Nr x Ntheta x Nphi x Nvpar, Nmu=0)
- 1 node, 4 MPI tasks, 8 threads (Broadwell) / 16 threads (KNL)



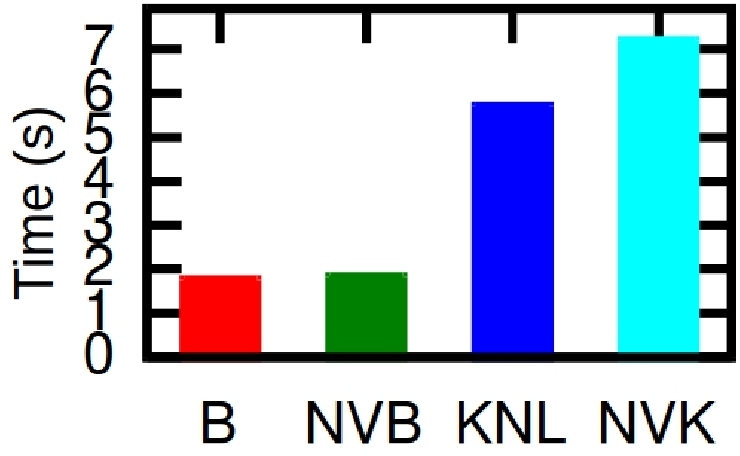


B: Broadwell
SB: Scalar Broadwell
KNL: Frioul KNL
SK: Scalar KNL
blz_timesplit_ bsl 30.02s
7.36s
3.40s
7.17s
2.63s
7.27s
collisions_ vpar_ advecdiff 7.36s
bsl_1d_vpar_ advec 3.40s
bsl_2d_ poloidal_ advec 7.17s
bsl_1d_phi_ advec 2.63s
sources_all_ advance (7.04s) 7.27s
7.35s
3.40s
7.17s
2.63s
0.01s
collvpar_vers2and3_ advecdiff (2.47s) 7.35s
bsl_1d_vpar_lag_ advec (3.40s) 3.40s
bsl_2d_lag_ taylor_ advec (0.10s) 7.17s
bsl_1d_phi_lag_ advec (2.63s) 2.63s
3.13s
1.75s
5.22s
collvpar_vers2and3_ dparnupar (3.23s) 3.93s
utl_tridiag_ r (1.75s) 1.75s
bsl_2d_process_lag_ taylor_ ith (5.22s) 5.22s
DPTTRS (1.70s) 1.70s
B SB KNL SK

UTL_TRIDIAG_R

- Solve a tridiagonal system
  - forward elimination
  - back substitution
- Not vectorizable

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}$$



instruction	instruction	Broadwell latency	Broadwell throughput	KNL latency	KNL throughput
3N	FMA	5	2	6	2
2N-1	DIVSD	10-14	1/5-1/4	42	1/42
2N-1	VDIVPD	19-23	1/16	32	1/32

Source: A. Fog Instruction tables, TU Denmark 2016, http://www.agner.org

SUMMARY

Summary

- Good stuff:
  - MPI latency and OpenMP overhead comparable
  - KNL can match Broadwell performance without extensive tuning for most codes
  - Optimization on KNL helps on Broadwell and vice versa
- Bad Stuff:
  - Cache mode operation can be dubious
  - Peak performance hard to reach
  - Hyperthreading rarely useful

Summary

- KNL is equal to Broadwell if your code either
  - Has very good scalability (to make use of increased core count)
  - Has very good vectorization (to make use of more vector units)
  - Effectively uses only 16 GB (to make use of higher bandwidth)
- If more than one holds, you will probably get more performance than on Broadwell
- Memory mode Quadrant seems to be the best