Design Considerations for Latency and Throughput on KNL

Jed Brown jed@jedbrown.org (CU Boulder)
Collaborators: Karl Rupp, Satish Balay, Matthew Knepley, Richard Mills, Barry Smith

MultiCore6, 2016-09-14

Scaling goals

Aurora ESP: Evaluation of Proposals

- An existing or reasonably well-planned implementation to make use of thread concurrency on Aurora. ALCF expects to see strong-scaling up to at least 8 threads per MPI rank, with greater than 75% efficiency. – `https://www.alcf.anl.gov/programs/aurora-esp`

- How much memory bandwidth is achievable with 8 threads on KNL/7210?
- MCDRAM: 110 GB/s (of 420 GB/s)
- DRAM: 80 GB/s (of 88 GB/s)

Scaling goals

Aurora ESP: Evaluation of Proposals

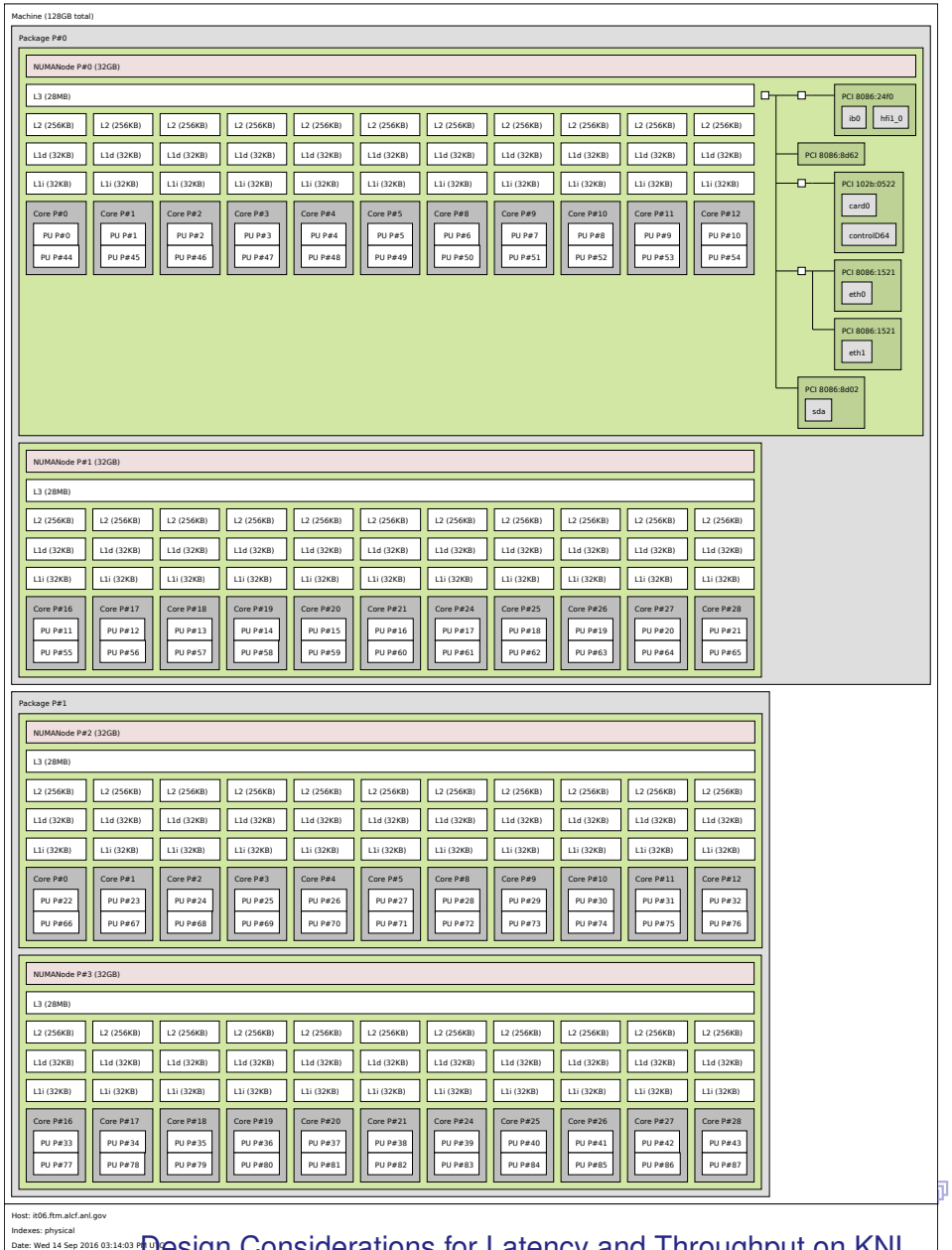
- An existing or reasonably well-planned implementation to make use of thread concurrency on Aurora. ALCF expects to see strong-scaling up to at least 8 threads per MPI rank, with greater than 75% efficiency. – `https://www.alcf.anl.gov/programs/aurora-esp`

- How much memory bandwidth is achievable with 8 threads on KNL/7210?
- MCDRAM: 110 GB/s (of 420 GB/s)
- DRAM: 80 GB/s (of 88 GB/s)

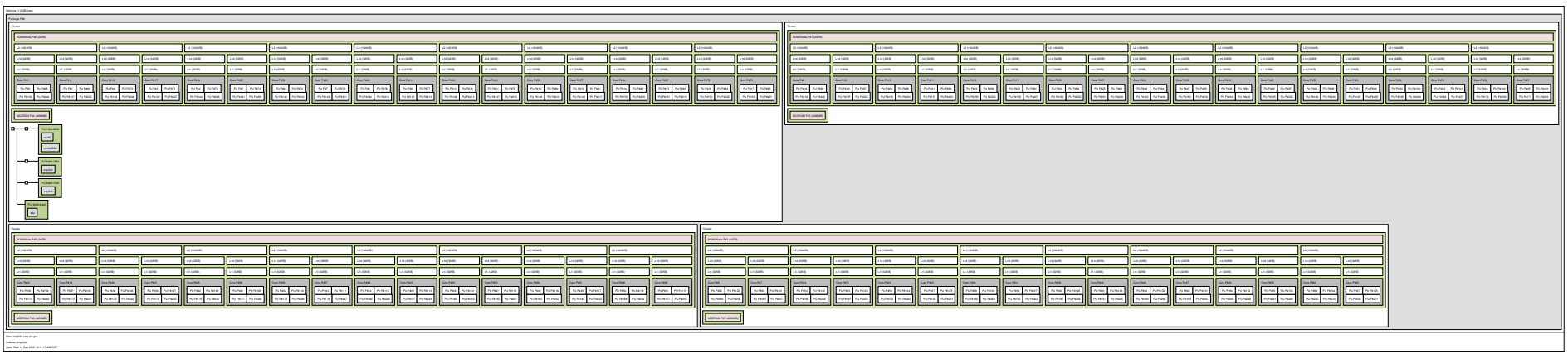
But affinity is hard

- MPI in Flat:Quadrant worked with default settings – 110 GB/s for 8x1
- MPI in Flat:SNC-4 seems to require manual enumeration
  `mpiexec -n 8 -env I_MPI_PIN_PROCESSOR_LIST 0,2,18,20,36,38,52,54 numactl -m 4,5,6,7`
- OpenMP with all tested variants of affinity flags – 63 GB/s for 1x8
  - My Intel colleague was adamant this was the best possible and I must have a bug in my MPI test.
- `KMP_AFFINITY='explicit,proclist=[0,8,16,24,32,40,48,56],gra`
  `OMP_NUM_THREADS=8 numactl -m 1 ./stream` – 110 GB/s

NUMA architecture: E5-2699v4



MCDRAM placement: Xeon Phi 7250



- Cores are not in a NUMA domain.

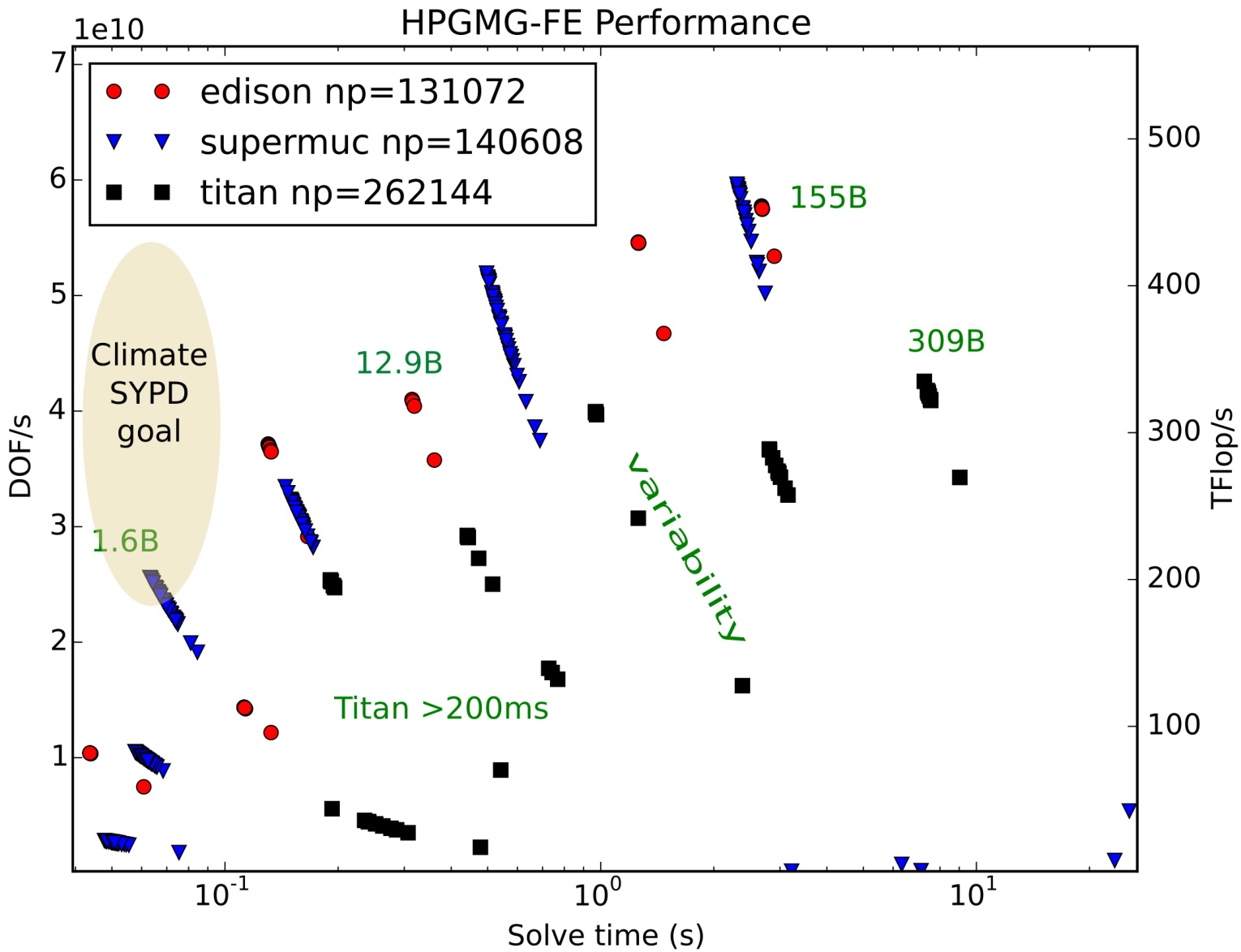
Automatic NUMA migration

- Linux feature for a few years now (Rik van Riel, Red Hat)

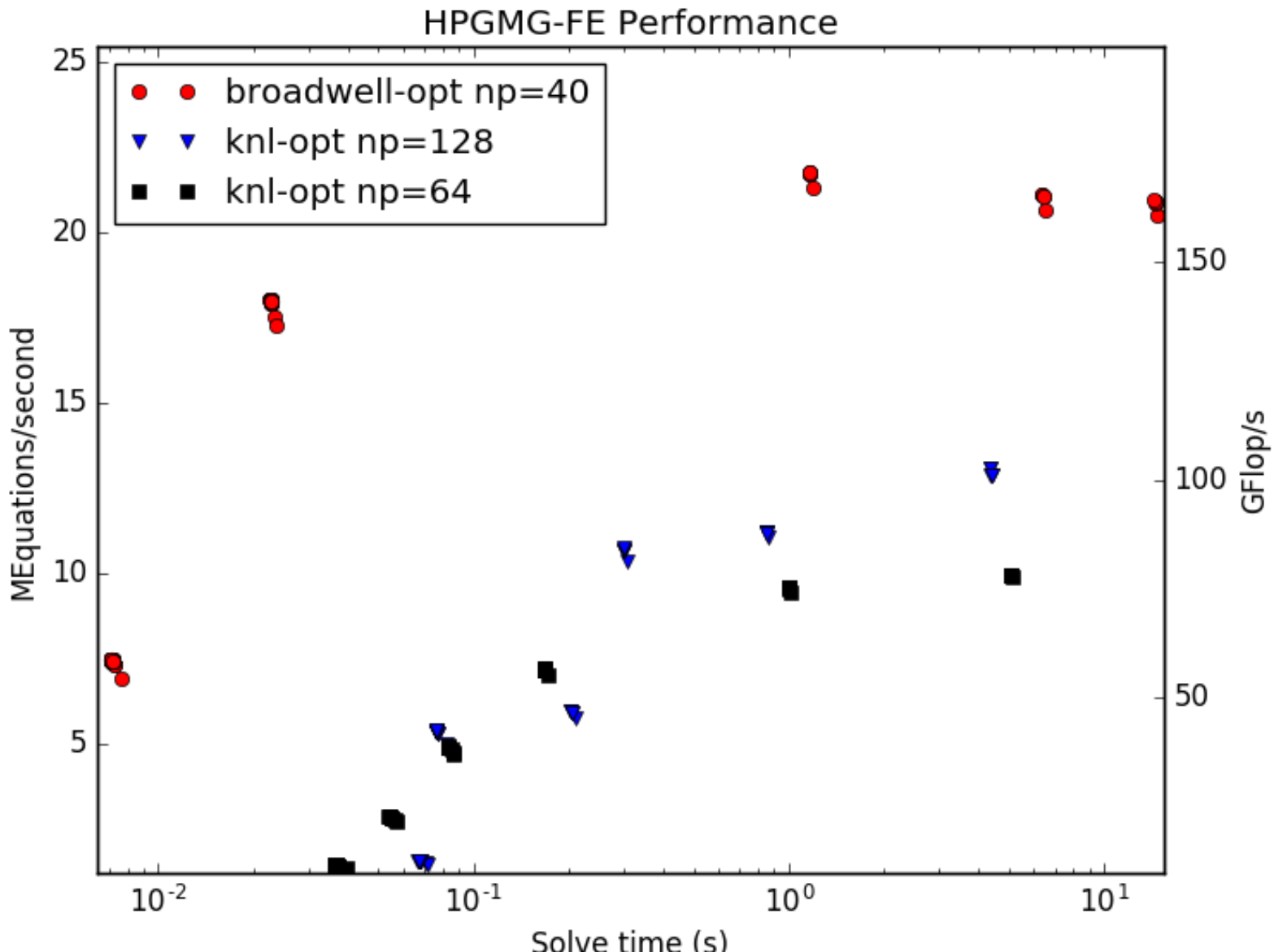
NUMA page migration

- NUMA page faults are relatively cheap
- Page migration is much more expensive
  - ... but so is having task memory on the “wrong node”
- Quadratic filter: only migrate if page is accessed twice
  - From same NUMA node, or
  - By the same task
  - CPU number & low bits of pid in page struct
- Page is migrated to where the task is running

Scaling regime: HPGMG-FE on Edison, SuperMUC, Titan



HPGMG-FE: Broadwell and KNL



HPGMG-FE: Broadwell profile

% time	cumulative seconds	self seconds	calls	self s/call	total s/call	name
19.87	3.71	3.71	1979816	0.00	0.00	DMFEExtractElements
18.80	7.22	3.51	2330024	0.00	0.00	OpPointwiseElement_Poiss
14.78	9.98	2.76	14040600	0.00	0.00	TensorContract_FMA_8_1_3
13.82	12.56	2.58	676	0.00	0.02	OpApply_Poisson
8.89	14.22	1.66	3176520	0.00	0.00	TensorContract_FMA_8_3_3
7.82	15.68	1.46	1023800	0.00	0.00	DMFESetElements
3.32	16.30	0.62				DMGlobalToLocalEnd_FE
3.27	16.91	0.61				DMLocalToGlobalEnd_FE
2.62	17.40	0.49	168	0.00	0.00	DMFERestrict
1.71	17.72	0.32	168	0.00	0.00	DMFEInterpolate

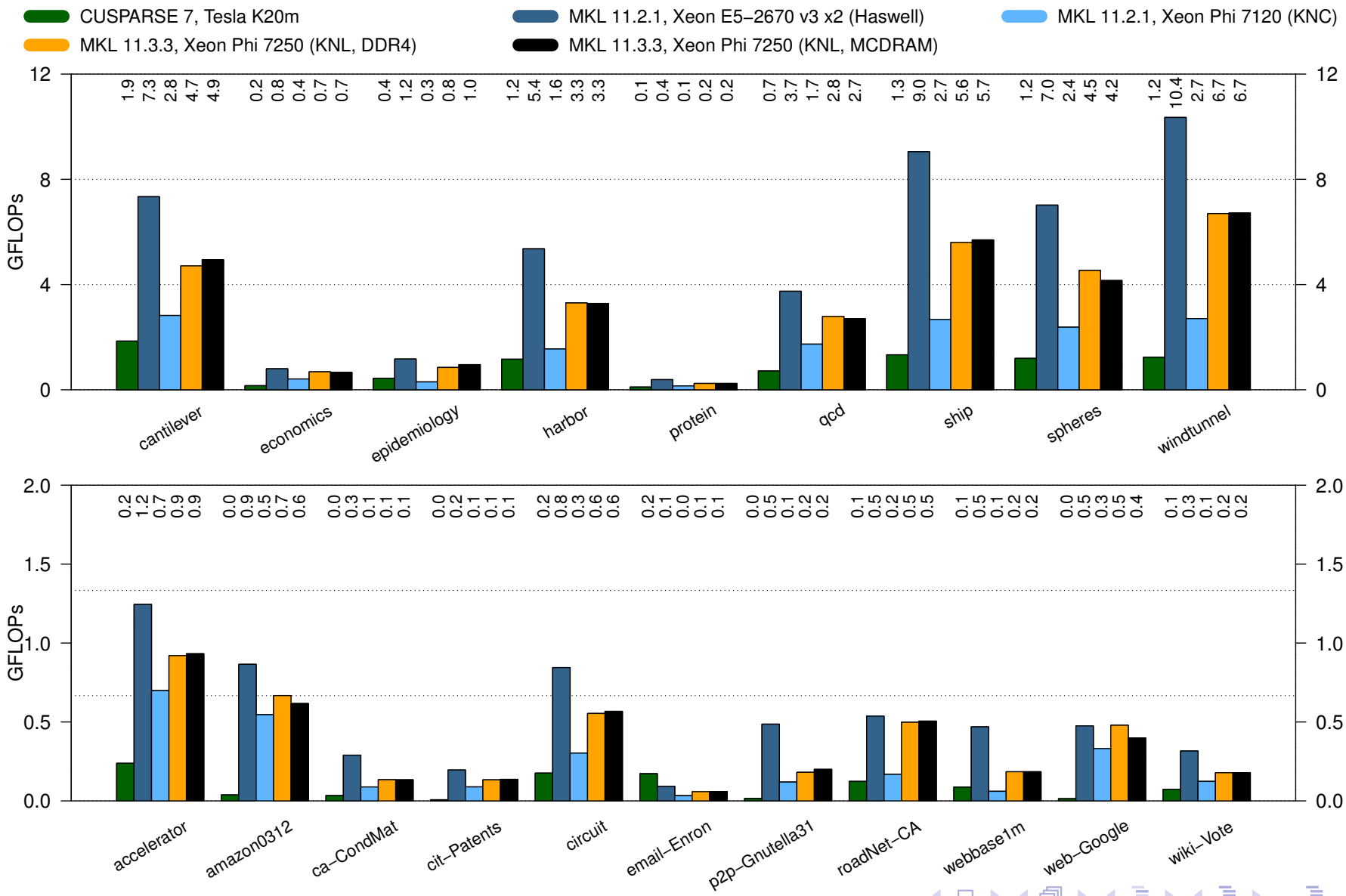
HPGMG-FE: KNL profile

% time	cumulative seconds	self seconds	calls	self s/call	total s/call	name
21.01	5.06	5.06	647544	0.00	0.00	DMFEExtractElements
15.95	8.90	3.84	396	0.01	0.04	OpApply_Poisson
13.64	12.19	3.29	5220144	0.00	0.00	TensorContract_AVX512_8_
12.96	15.31	3.12	865620	0.00	0.00	OpPointwiseElement_Poiss
12.15	18.23	2.93	1049004	0.00	0.00	TensorContract_AVX512_8_
8.10	20.18	1.95	336528	0.00	0.00	DMFESetElements
2.45	20.77	0.59	5497632	0.00	0.00	OpPointwiseForcing_Poiss
2.37	21.34	0.57	140	0.00	0.00	DMFERestrict
1.95	21.81	0.47	140	0.00	0.00	DMFEInterpolate
1.95	22.28	0.47				DMGlobalToLocalEnd_FE
1.87	22.73	0.45				DMLocalToGlobalEnd_FE

HPGMG-FE: KNL assembly

```
vmulpd zmm1,zmm3,ZMMWORD PTR [r12+r14*1+0x40]
vmulpd zmm2,zmm3,ZMMWORD PTR [r12+r14*1+0x80]
vmulpd zmm4,zmm3,ZMMWORD PTR [r12+r14*1+0xc0]
vbroadcastsd zmm5,QWORD PTR [r15+rsi*1+0x8]
vmovups ZMMWORD PTR [rax+0x40],zmm5
vfmadd231pd zmm0,zmm5,ZMMWORD PTR [r12+r14*1+0x100]
vmovups ZMMWORD PTR [rdi+r9*1],zmm0
vfmadd231pd zmm1,zmm5,ZMMWORD PTR [r12+r14*1+0x140]
vmovups ZMMWORD PTR [rdi+r9*1+0x40],zmm1
vfmadd231pd zmm2,zmm5,ZMMWORD PTR [r12+r14*1+0x180]
vmovups ZMMWORD PTR [rdi+r9*1+0x80],zmm2
vfmadd132pd zmm5,zmm4,ZMMWORD PTR [r12+r14*1+0x1c0]
vmovups ZMMWORD PTR [rdi+r9*1+0xc0],zmm5
```

Sparse matrix-matrix products (AMG setup)



Outlook

- MCDRAM is likely all we need for strong scaling
- SNC-4 requires manual affinity for MCDRAM (due to NUMA mapping)
- If the kernel understood the actual memory architecture, perhaps it could automate placement
- TLB effect? 400 GB/s with MPI versus 420 GB/s with threads
- Irregular access/packing is becoming more expensive