Trinity Benchmarks On Xeon Phi (Knights Corner)

https://www.nersc.gov/users/computational-systems/nersc-8-system-cori/nersc-8-procurement/trinity-nersc-8-rfp/nersc-8-trinity-benchmarks.

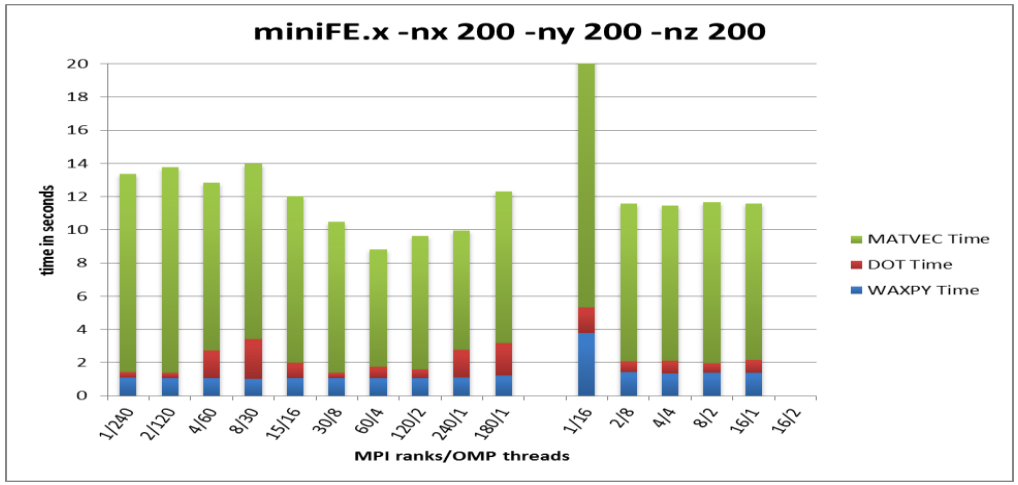
Mahesh Rajan, Doug Doerfler, Si Hammond, Christian Trott, Richard Barrett

Sandia National Laboratories

Albuquerque, NM

NERSC-8/Trinity benchmarks; Optimal MPI tasks & OMP threads on Knights Corner and host dual Sandy Bridge(E5-2670) investigated

Benchmark	Domain/Physics	Algorithm/Kernel	Phi (Knights Corner)to Dual Sandy Bridge ; wall time Ratio
miniFE	Finite Element	Sparse Linear Algebra/ MatVec	0.81
AMG	Algebraic Multi Grid	Hypre / csr_solve / MatVec	2.01
UMT	Det. Photon Transport	Boltzmann Transport solve/ OMP 'ordinate' computations	1.38
SNAP	Proxy Particle Transport	Boltzmann Transport Solve/OMP Loop over outer energy domain	2.95
GTC	3-D PIC plasma	Vlasov Eqn. Solve / OMP over particle flux computations	1.95
MILC	QCD	Ks_dynamical Simul. / OMP over Conjug. Grad. Solve only	2.32
miniDFT	Plane-wave DFT based on Espresso	Folk Matrix Diagonalization and FFT	3.30

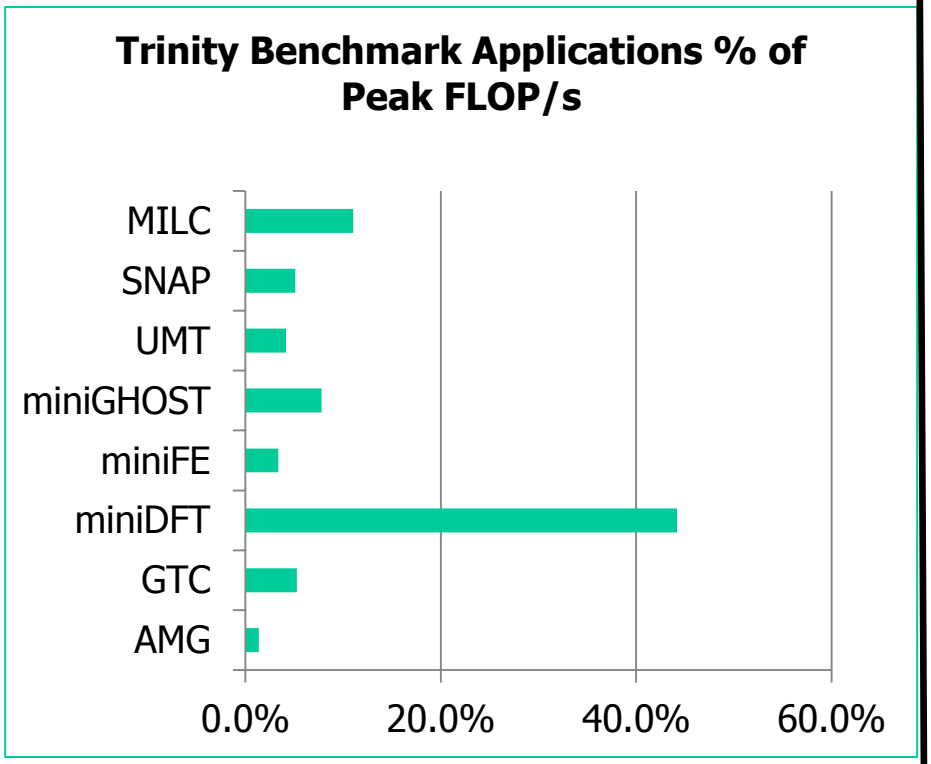


miniFE performance on Knights Corner (left) and Dual Sandy Bridge host right)

miniFE Vectorization Intensity for Sparse MatVec Kernel = VPU_ELEMENTS_ACTIVE/VPU_INSTRUCTIONS_EXCECUTED = 7.634e08/4.363e08 = 1.75
Optimal=8; Plan to benchmark using Intel MKL with tuned Sparse Matrix vector kernel.

https://software.intel.com/en-us/articles/the-intel-math-kernel-library-sparse-matrix-vector-multiply-format-prototype-package

Vectorization Investigation & Sustained FLOPS



Measurement with CrayPat; Percentage of peak on Cielo (Cray XK6) with Trinity "small" input benchmarks; Points to even greater challenge with TFLOPS Phi; we need to improve vectorization and minimize threading & MPI overhead

Application	miniFE	AMG	UMT	SNAP
Vectorization %age Gain	4.68%	-6.52%	17.95%	19.52%

compiler vectorization; compare with vectorization (-O3) and without vectorization(-novec –O3)

PMU counter usage and its limitations; Example DGEMM: Vectorization intensity defined as:

Vectorization Intensity = VPU_ELEMENTS_ACTIVE / VPU_INSTRUCTIONS_EXECUTED

Measurements of this metric ratio on the MIC with 4 and 8 threads for DGEMM :

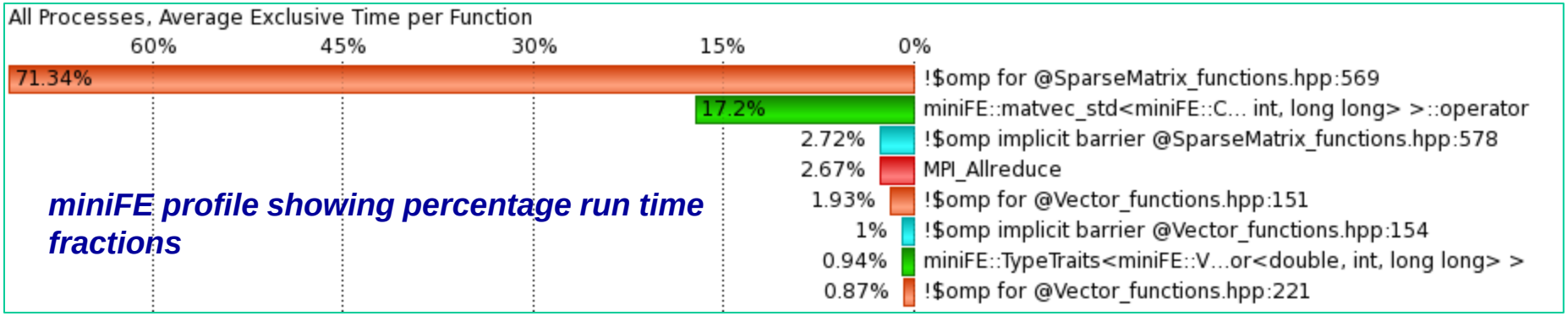
vectorization intensity DGEMM = 7.84

This metric has an upper bound of 8. Values close 8 suggest efficient use of MIC's SIMD units. However since the VPU_ELEMENTS_ACTIVE counter measures vector instructions like vector load/stores from memory, and instructions to manipulate vector mask registers, in addition to the double precision floating point instructions, caution is needed in use of this metric for performance tuning. The fact that our measurements of this metric achieves close to the peak showing high vectorization intensity is misleading if our goal is to achieve high floating point operations throughput. The percentage of peak double precision floating point operations achieved with MKL DGEMM in this test is about 30%

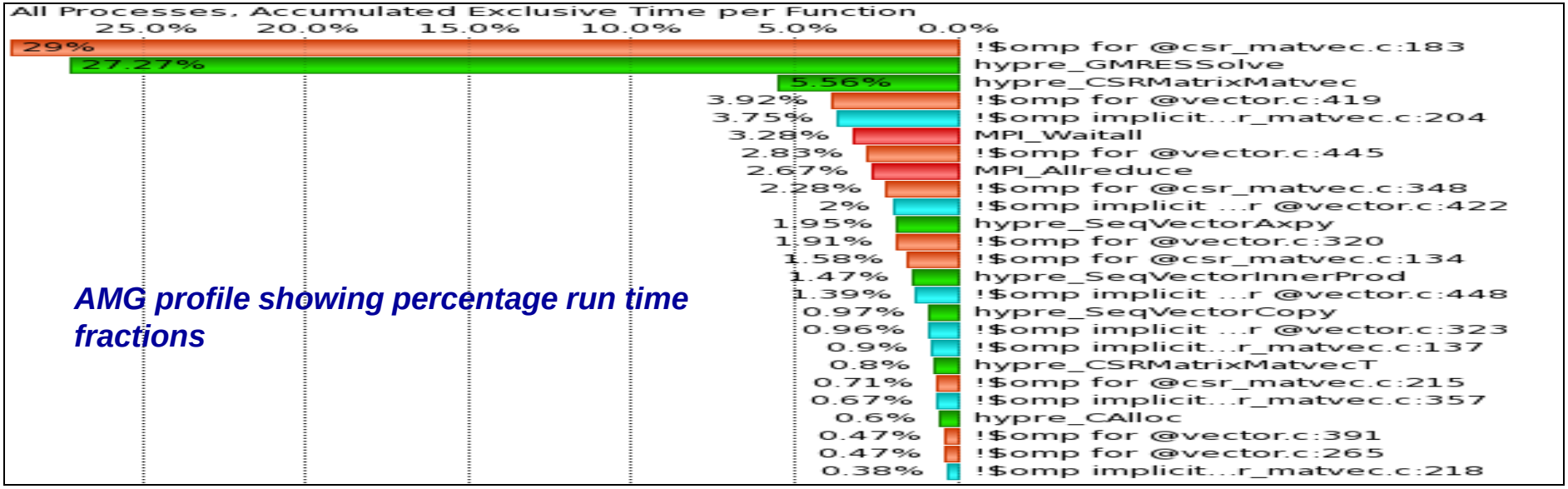
Need in KNC & KNL: DP_OPS counter

Tools to understand performance; Vampir Profile; Needed information to tune: %age fraction of run time in OMP loops, OMP overhead, MPI, MPI overhead, and application functions

Function Legend



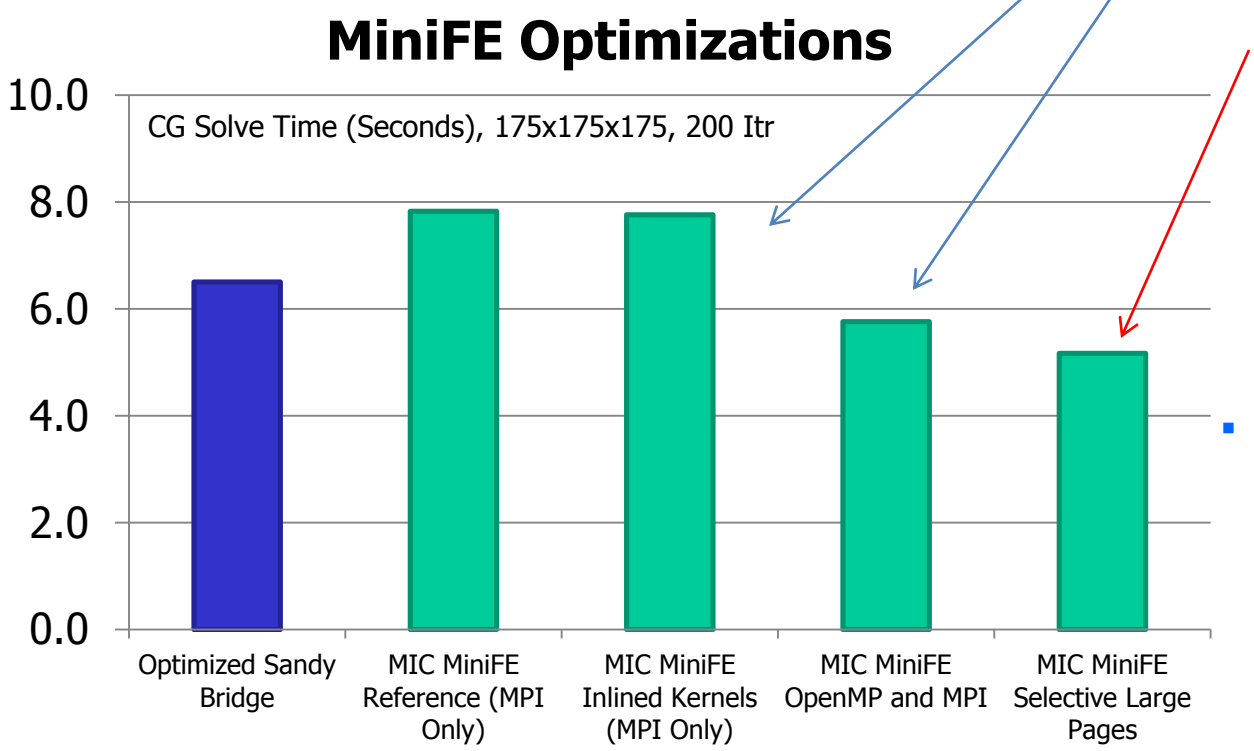
miniFE profile showing percentage run time fractions



AMG profile showing percentage run time fractions

Tuned Performance of MiniFE

- **Baseline all MPI performance 23% slower than optimized Sandy Bridge implementation**
- **Goal to provide strong performance using 1 MPI rank up to 224 threads**



- Add affinitized OpenMP instead of MPI and inlining small kernels gets 26% improvement
- Disable transparent huge pages and selectively use large page allocations for vector data structures (to lower TLB miss rate)
- End performance is approximately 33% higher on 57 core Xeon Phi, ~20% over SNB