Lessons Learned From Optimizing Applications on Xeon Phi

Jack Deslippe

HPC Consultant

NERSC

What’s unique about my tuning work

- BerkeleyGW
- Materials Science
- Native, MPI+OpenMP
- Vtune, SDE, IPM, CrayPat

Performance

- Slight (~10%) advantage on MIC vs Dual Socket Xeon
- Speed ups vary by kernel. Some as large 4x on Xeon. Essentially infinite speed-up gained on Xeon-Phi as original code could only use fraction of cores on card due to memory limitations.
- Refactoring loops, improve memory locality (up to 2x)
- Adding OpenMP (large improvements on MIC, minimal on host)
- Ensuring inner loop vectorization (>2x on MIC)

Insights

- Locality and on-node parallelism extremely important. Changes targeting MIC greatly improve code on Xeon as well.
- Need large inner loops to vectorize or alignment issues become import.
- OpenMP simd flags are cool.
- Vtune should be used early can easily tell you if your code is compute or memory bandwidth bound
- Divide instructions remain a challenge