MAGMA MIC: HPC Linear Algebra for Intel Xeon Phi

Piotr Luszczek

Research Director

University of Tennessee

Knoxville, Tennessee, USA

What's unique about my tuning work

- Application:
  - MAGMA MIC – numerical linear algebra library for dense matrices
- Numerical methods
- Execution mode: offloaded
- Software tools:
  - SCIF, COI, MIC offload pragmas, MKL routines
  - SCIF small asynchronous transfers
  - We prefer COI for portability and simplicity but cannot afford to loose performance
  - MIC offload pragmas are the most convenient and used for testing and debugging
  - MKL alleviates the burden of assembly coding and getting low-level bits just right

Performance

- Compelling (and competitive) performance with MIC:
  - ❑ MIC results match or exceed performance of NVIDIA GPUs
- Speedup on Xeon only = 2x
- Speedup on MIC only = 8x (vs. autotuned open source on x86)
- Optimizations
  - ❑ Use of MKL (2x, 3x, …)
    - With the right library call, performance matches expectations
  - ❑ Selective multi-threading (50%+)
    - More threads than x86 for memory-bound operations
  - ❑ Nested parallelism (40%+)
    - OpenMP API calls, affinity setting
  - ❑ Use of intrinsics (2x+)
    - Last resort for custom kernels

Insights

- I learned that with right expectations Phi coding is easy
- Start with reference implementation
- Test native libraries for "edge cases" exposed by your algorithm
  - ❑ Oddly shaped and misaligned matrices/vectors
- We have our internal tracing, debugging, performance tools: let us know if you're interested
- Biggest surprises:
  - ❑ Importance of memory alignment
  - ❑ Effects of in-order scheduling
  - ❑ Versatility of the software that "just works"
- A lot of work to do to cover all of LAPACK functionality
- How will porting from KNC to KNL work?