NASA Overflow performance testing on Intel® Xeon Phi™

Tim Prince

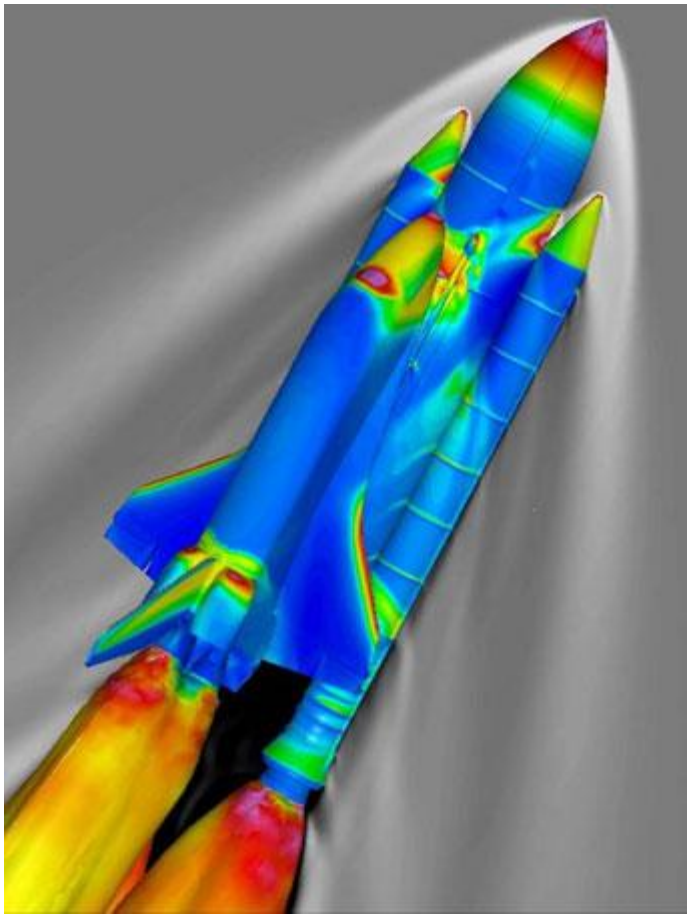
Intel SSG/DRD

Overset conference Oct. 17 2012

Overflow Visualization

More applications:

- Boeing aircraft
- Environmental
- ….



Agenda

Intel® Xeon Phi™ development environment

Intel® Xeon Phi™ build/run overview

MPI vs. offload model

Performance issues

Recommendations

Intel® Xeon Phi™ development environment

- Familiar HPC programming models
  - Fortran, C, C++
  - MPI, OpenMP
- Build separate Xeon host and Xeon Phi™ executables
  - Sharing same source code
- MPI mode: mpiexec designates ranks for host and Xeon Phi™

Build/run models for Overflow

Large cluster runs are expected

- MPI (no OpenMP) in competitive acquisitions

First effort: offload within node

- About 100 offload directives
- Work around limitation of MIC OpenMP

Current effort: MPI/OpenMP hybrid

- Xeon Phi™/KNC runs same source code as host
- Loop count and vector no peeling directives

MPI hybrid vs. offload

Offload model: 1 large OpenMP "rank" per Phi card

- Multiple data transfers in "out of the box" model
  - restructuring needed for better locality
- Threaded scaling is weak beyond 9 threads

MPI/OpenMP hybrid opportunity

- Avoid excessive memory requirement of MPI alone
  - num_threads per MPI process within peak scaling
- KNC native scales to 16 processes, 9 threads per process
  - Overflow has no serious serial non-vector obstacles
- KNC + Xeon host mode current best result:
  - 5 ranks on 61-core KNC x 36 threads/12 cores per process
    - Must execute via script on MIC side to set ulimit etc.
    - + 6 ranks x 2 threads/2cores per process on host side

Technical performance issues

"AOS"-like local data

- 24 physical properties per storage element
  - Stride 24 access in xmet,ymet,zmet
  - KNC is first Intel hardware for strided vectorization
  - Experimentally, little performance effect with SOA change

Limited vector loop length

- 200x200x200 grid fits
  - split 50/50 between host and card
- Default build options result in little effective prefetch
  - Loop count directives recommended throughout
    - Customer resistance

Remaining performance issues

30% rate of L2 miss not covered by prefetch

- Platform favors longer vectors

2% performance tied up in DTLB miss rate?

- Practical method to try 64K page?

Summary

MIC is promising platform for HPC

- Competitive advantage with standard programming
- Mainstream HPC development model support
  - Fortran/C interoperable
  - OpenMP
  - MPI

Increased computation per cluster node

- Optimize around 4M grid points per 8GB Phi card
  - Competitive only with both host and MIC working
- > 90% gain on WSM node for Overflow
  - “symmetric” (host 6 processes, KNC 5 processes)
  - SNB TBD

Recommendations

Optimize for host + KNC MPI/OpenMP hybrid mode

- "many" threads/rank on KNC (few ranks)
- "few" threads/rank on host

Maximize loop lengths for KNC

- use compile options to optimize for loop count

SNB-EP may need 2 KNC cards for optimum config

- should be optimum for 12-16M grid per node