Supercomputing Notes

Focusing on Science and GPUs

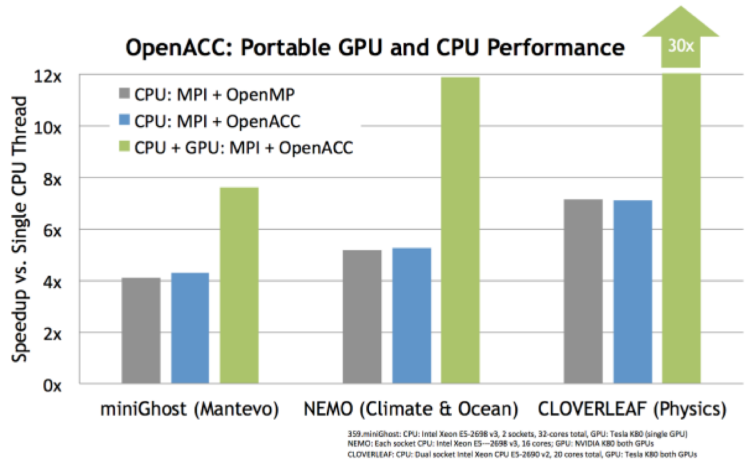
A. Norman

Fermilab

GPU Impressions

- Common theme from all major GPU players booths (Nvidia, AMD, Intel)
  - “Our specialized <language, libs, API> is what you should use”
  - “But if you don’t you should use OpenMP, you’ll take a 10-20% performance hit on most standard code relative to hand optimized algorithms”
  - Booths were all showing the same benchmarks
- Compiler booths are similar
  - Emphasize their support for OpenMP 4.x
  - All (but PGI) claim to have the best implementation*
  - Nvidia emphasizing pre-optimized libraries of standard algorithms for STL containers

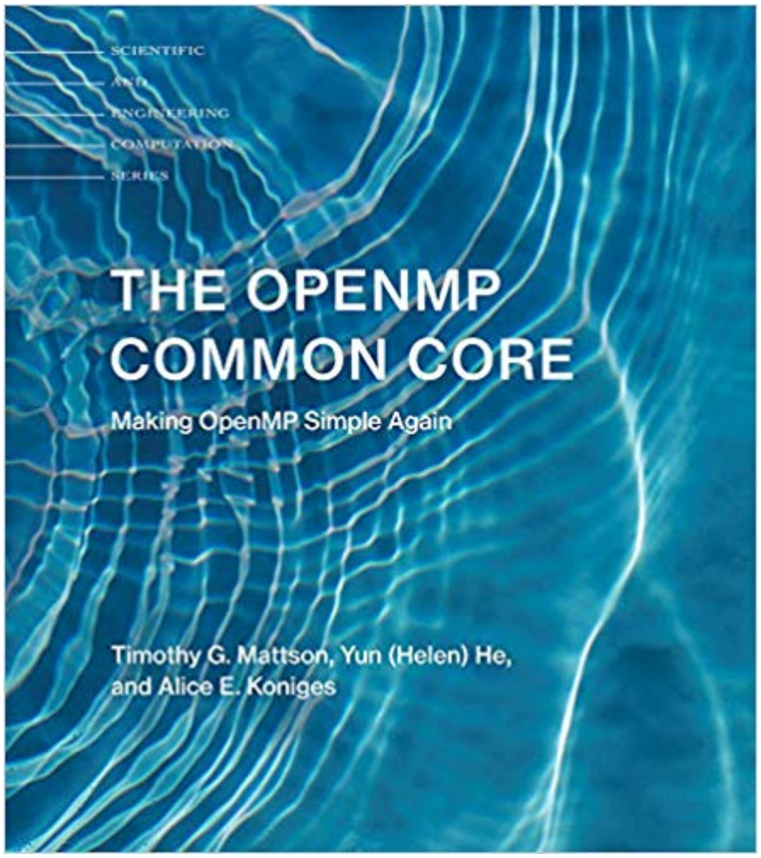
OpenMP
Enabling HPC since 1997



*on whichever flavor of GPU they specifically support

OpenMP Training

- New spec 5.0 is out but…
  - Real progress is on distilling down to the "common core" and compiler support for 4.5
  - Essential directives and patterns that cover most scientific use cases
    - OpenMP was touting this (passing out cheat sheets), talking up new book.
    - Major initiative towards onboarding applications quickly
  - Compilers are better optimization for common core directives (i.e. sensible default behaviors less tuning)
    - https://www.openmp.org/resources/openmp-compilers-tools/
  - Tutorial was actually VERY good (joint with NERSC)
    - Easy to replicate
  - Low hanging fruit for some experiment code



SCIENTIFIC
AND
ENGINEERING
COMPUTATION
SERIES

THE OPENMP
COMMON CORE

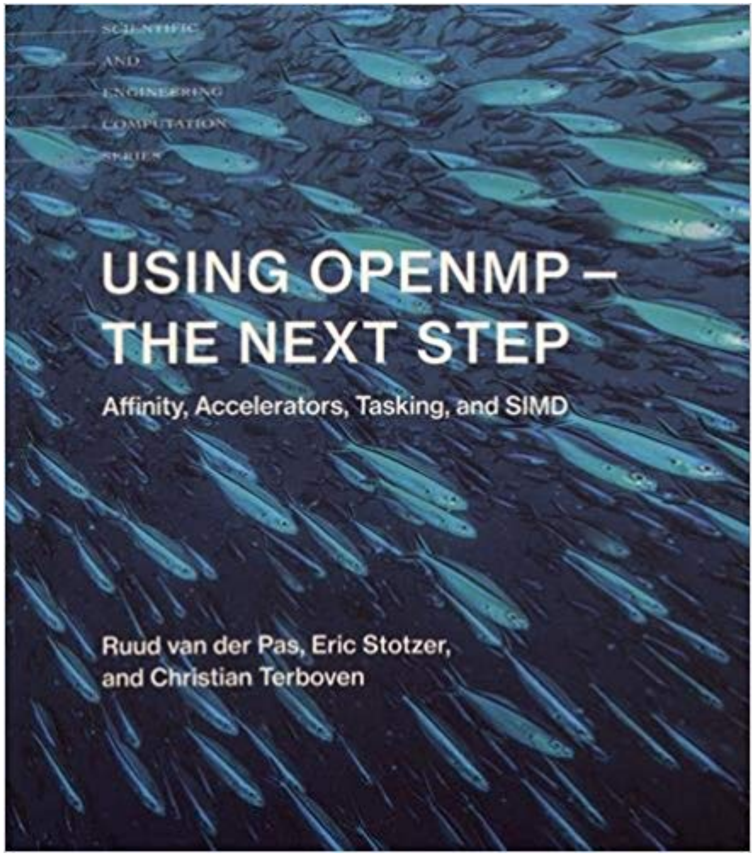
Making OpenMP Simple Again

Timothy G. Mattson, Yun (Helen) He,
and Alice E. Koniges

- **GPU offloading a minimal extension to common core**

OpenMP GPU Training

- Simplified offloading to target devices in the base part of the spec
  - *Builds directly off common core directives*
  - Can effectively swap out a single directive in most cases to go from OpenMP parallel to OpenMP GPU accelerated
  - Performance is “meh…” without tuning and memory model considerations
  - Example codes were getting get 4-8x ish boosts
  - Tune examples get 20x
- Value is in portability and ease of migration
  - Very real possibility for our science codes that don’t lend themselves to hand optimization
  - **Documentation and training materials are good**


SCIENTIFIC AND ENGINEERING COMPUTATION SERIES
USING OPENMP – THE NEXT STEP
Affinity, Accelerators, Tasking, and SIMD
Ruud van der Pas, Eric Stotzer, and Christian Terboven

GPU Hackathon

- Connected with GPU Hackathon team
  - Learned more about what to expect and how to schedule a hackathon (this is in the NESAP context of our NESAP project)
  - For application porting they want:
    - 1-3 people to participate (coder, algorithm person, person for testing)
    - Start 4-6 week before actual hackathon
    - Need code to compile using Cray compiler
    - They want a kernel identified if possible, but are willing to work with more generalized code
-

Rescale

- Single API (and accounting!) for AWS, Google, Microsoft
- Can buy time through them or…
  - Bring your own allocations
    (specifically asked about Heidi usecase of a Microsoft Educational allocation)
- Claim to have HARD CAPS and cut offs on per group basis and linked to funding and administrative limits.
  - Want to see accounting interface
- This actually may be a viable path to avoid separate integration for each cloud system. Would want to see more.

IBM

- Was given the briefing (hard sell) on LSF batch

- Claim is that it can scale now.
- Lacks various accounting controls and monitoring

- Want us to use it with HEPCloud
- Want to do a more complete briefing for us