Profiling and Optimizing for Xeon Phi with Allinea MAP

Discovering bottlenecks without pain

Beau Paisley
Allinea Software
bpaisley@allinea.com
970.389.0868
Three Challenges for Tools

- **Scalability**
  - Speed and Simplification

- **Heterogeneity**
  - Accelerators and Coprocessors

- **Adoption**
  - Ease of Use and Education
Allinea Unified Environment

- A modern integrated environment for HPC developers

- Supporting the lifecycle of application development and improvement
  - Allinea DDT: Scalable, productive debugging
  - Allinea MAP: Easy-to-use application profiling

- Designed for productivity
  - Consistent easy to use tools
  - Enables effective HPC development

- Improve system usage
  - Fewer failed jobs
  - Higher application performance
Allinea MAP

Increase application performance

• Parallel profiler designed for:
  – C/C++, Fortran
  – Multi-process code
    ▪ Interdependent or independent processes
  – Multithreaded code
    ▪ Monitor the main threads for each process
  – Accelerated codes
    ▪ GPUs, Intel Xeon Phi

• Improve productivity:
  – Helps you detect performance issues quickly and easily
  – Tells you immediately where your time is spent in your source code
  – Helps you to optimize your application efficiently
• Look at the entire application on real data sets
  – Visualize the entire run at full scale, not just reduced sets
  – Zoom in to explore iterations, functions and loops

• Understand the nature of bottlenecks
  – Source code viewer pinpoints bottleneck locations
  – CPU, MPI, I/Os and memory metrics identify the cause
• **Unique profiling methodology (even on Intel Xeon Phi)**
  - No need to instrument your code
  - Just 5% wall-clock overhead
  - Small output files (10-20Mb is typical)
  - Integrates with debugging as an application and a process

• **Benefit:**
  - Most profiling activity can now be done by end-users
  - Helps to focus on the right bit of the application
  - Brings more value to the expertise of application support
  - Enables and encourages optimization sooner rather than later
Optimizing for the Xeon Phi
But what matters?

- Vectorization
- Performance
- Other stuff
Optimizing for the Xeon Phi
Is my code well-vectorized?
Optimizing for the Xeon Phi
Is my code well vectorized?
Optimizing for the Xeon Phi
Is my code well vectorized?

Not in this loop
(16.5% of total time)
Optimizing for the Xeon Phi
Non-obvious tradeoffs
Optimizing for the Xeon Phi
Non-obvious tradeoffs

Here a loop taking 55% of total runtime isn’t vectorized at all.

Taking the unvectorizable `rand()` out of the loop allows the `sqrt` workload to be fully-vectorized – reverse loop fusion!
Now the floating-point workload is fully-vectorized

But all the time is being spent in the random number generation, so that’s what really needs to be optimized
Optimizing for the Xeon Phi
Running on the card!

Allinea MAP runs with full metrics on Xeon Phi cards!
What Our Users are Saying

“My group routinely debugs code at over 100,000 processes using Allinea DDT. No other debugger comes close – obviously it’s a hit with users,” Oak Ridge National Laboratory

“Allinea’s experience and tools will make a big impact in the speed at which scientists can complete their research,” NCSA Blue Waters

“Previous experiences with other profilers had left us more confused than informed. Allinea MAP is the opposite.”
Thank You!

Try it out at:
http://www.allinea.com/products/trials/

Beau Paisley
Allinea Software
bpaisley@allinea.com
970.389.0868