Debugging and Optimizing Programs Accelerated with Intel® Xeon® Phi™ Coprocessors

Chris Gottbrath, Principal Product Manager
May 7th, 2013

CUG 2013, Napa, CA
Rogue Wave Today

The largest independent provider of cross-platform software development tools and embedded components for the next generation of HPC applications.

Highlights
- Pioneers in C++/object-oriented development
- Leading the way in cross-platform, parallel development

History
- Founded: 1989
- Acquired by Audax Group: 2012
- Acquired:
  - Visual Numerics: 2009
  - TotalView Technologies: 2009
  - Acumem: 2010
  - IBM ILOG Views C++: 2012
- 40 years of experience in HPC

Customers
- 3,000+ customers in 36 countries
- Multiple sectors:
  - Financial services
  - Telecom
  - Oil and gas
  - Government and aerospace
  - Research and academic
Rogue Wave Solution Portfolio

Developing parallel, data-intensive applications is hard.

We make it easier.
What is TotalView?

- Application Analysis and Debugging Tool: Code Confidently
  - Debug and Analyze C/C++ and Fortran on Linux, Unix or Mac OS X
  - Laptops to supercomputers (Cray, BG, BullX, etc..)
  - Makes developing, maintaining and supporting critical apps easier and less risky

- Major Features
  - Easy to learn graphical user interface with data visualization
  - Parallel Debugging
    - MPI, Pthreads, OpenMP, GA, UPC
    - CUDA and OpenACC, Xeon Phi (early access)
  - Includes a Remote Display Client freeing you to work from anywhere
  - Memory Debugging with MemoryScape
  - Deterministic Replay Capability Included on Linux/x86-64
  - Non-interactive Batch Debugging with TVScript and the CLI
  - TTF & C++View to transform user defined objects
TotalView for Xeon Phi

- Support Multiple Intel Xeon Phi configurations
  - Native Mode
    - With MPI
  - Offload Directives
    - Similar to GPU
  - Multi-device
  - Multi-node
    - Certain configurations
    - CS300-AC, Future XC30

- User Interface
  - MPI Debugging Features
    - Process Control
  - View Across
  - Shared Breakpoints
  - Heterogeneous Debugging
    - Debug Both Xeon and Xeon-Phi Processes
Spectrum of Execution Models

CPU-Centric
- Multi-core Hosted
  - General purpose serial and parallel computing
    - Main()
    - Foo()
    - MPI_*()

Intel® MIC-Centric
- Many-Core Hosted
  - Codes with highly-parallel phases
    - Main()
    - Foo()
    - MPI_*()
  - Codes with balanced needs
    - Main()
    - Foo()
    - MPI_*()
  - Highly-parallel codes
    - Main()
    - Foo()
    - MPI_*()

Productive Programming Models Across the Spectrum
Remote Debugging of Applications on Xeon Phi

- Just run as `totalview -r mic0 <program>`
- Attach to running application
- See thread private data
- Investigate individual threads
- Kill stuck processes on MIC-coprocessor
Debugging MPI Applications

- Attach to subset of processes on MIC coprocessor
- Set breakpoints
- Debug “as usual” MPI
Debugging Applications with Offloaded Code

Xeon side

Xeon Phi side

One debugging session for MIC-accelerated code
What’s New in TotalView 8.12

- Xeon Phi Support
- Formal support for Cray XC
- AVX Instruction Support (phase 1)
- Cray ATP Support

- Mac OS X Lion and Mountain Lion support
- Sessions Manager
- STL support for set, multi-set, multi-map
- Improvements for specifying addresses in C++ template breakpoints
- Updated OS and Compiler Support
Multi-phase R&D Projects Underway

• Massive Scalability
  – Collaboration with LLNL and Tri-lab partners
  – Targeting Cray, Blue Gene and Linux Clusters

• Shiny new GUI
  – Sleek, Modern and Fast
  – Configurable
  – Improved Usability
  – Provides aggregation capabilities for big data and scale
  – Leveraging math and stat expertise from IMSL

• Working with customers through early access programs
  – Customer input is key to the success of both programs
TotalView debugs 786,432 cores. Climb with Rogue Wave towards exacale.
Some more details on the 786,432 core test

- The test was performed on 48 racks of Sequoia
- The test code
  - Implements a Jacobi Linear Equation Solver
  - The test code is a hybrid MPI + OpenMP code
  - 16 threads per process, one process per node
- The test operations
  - Start up
  - Setting breakpoints / removing breakpoints
  - Single stepping all threads
- Tests performed at a variety of scales to understand scalability
Second test - Oversubscription

• Same framework
  – same code
  – same machine

• Oversubscription
  – Scheduled more than one thread per physical core
  – This is a reasonable use case since the BG/Q supports 4 logical threads per core

• TotalView Debugged 1,048,576 threads
What is ThreadSpotter?

• Runtime Cache Performance Optimization Tool: Tune into the Multi-Core Era
  – Realize More of the Performance Offered by Multi/Many-Core Chips
  – Quickly Detects and Prioritizes Issues -- and then Provides Usable Advice!
    • Brings Cache Performance Into Reach for Every Developer
    • Makes Experienced Cache Optimizers Hyper-Efficient

• Features
  – Supports Linux x86/x86-64 & Windows
  – Any compiled code
  – Runtime Analysis
    • Low overhead
  – Cache Modeling
    • Prioritizes Issues
    • Identifies Problem Lines of Code
  – Provides Advice
    • Explanations
    • Examples
    • Detailed statistics (if desired)
**Simple modifications can make a big difference**

<table>
<thead>
<tr>
<th><strong>Program A</strong></th>
<th><strong>Program B</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>struct DATA</td>
<td>struct DATA</td>
</tr>
<tr>
<td>{</td>
<td>{</td>
</tr>
<tr>
<td>int a;</td>
<td>int a;</td>
</tr>
<tr>
<td>int b;</td>
<td>int b;</td>
</tr>
<tr>
<td>int c;</td>
<td>int c;</td>
</tr>
<tr>
<td>int d;</td>
<td>int d;</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
</tr>
<tr>
<td>DATA * pMyData;</td>
<td>DATA * pMyData;</td>
</tr>
<tr>
<td>for (long i=0; i&lt;10<em>1024</em>1024; i++)</td>
<td>for (long i=0; i&lt;10<em>1024</em>1024; i++)</td>
</tr>
<tr>
<td>{</td>
<td>{</td>
</tr>
<tr>
<td>pMyData[i].a = pMyData[i].b;</td>
<td>pMyData[i].a = pMyData[i].b;</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
</tr>
</tbody>
</table>

**Partially Used Structures**
Partially Used Structures

Defined data structure includes a, b, c, d... but only uses a & b

Redefined data structure includes a, b, a, b, a, b... c, d are elsewhere.
Other opportunities for optimization include

- Alignment Problems
- False Sharing
- Excessive communication (cache coherence) traffic
- Temporal locality issues
- Spatial locality issues
- Loop fusion
Recent improvements to ThreadSpotter

• Improved parallel support
  – Support for sampling all MPI processes in an MPI job
  – Cray XT, XE, XK Support
    • ALPS, SLURM and Torque
  – Continued additions to the processor library
    • Including cross-processor analysis
Next release: Improving ThreadSpotter MPI support

• Launchmon
  – Provides scalable mechanism for launching the tool in HPC clusters
  – Allows for coordination and synchronization of sampler activity
    • Will reduce “load balancing” bias that might otherwise be introduced by uncoordinated burst sampling with ThreadSpotter
  – Parallel framework can also be used for post-sampling processing

• Clustering Analysis
  – Some level of variability in sample results across the run
    • However the bulk of the results will be similar
  – Identify clusters of similar performance data
  – Present a small number (2-5) of reports that represent those clusters
  – Cluster analysis is done in parallel right after the sampling is completed
ThreadSpotter work towards supporting the Xeon Phi

- Xeon Phi has an interesting cache architecture
  - L1 & L2 caches for each core
  - The set of all the L2 sometimes described as “shared”
  - L2 caches organized around a ring-shaped bus
    - Duplication of data referenced by more than one
  - Successful cache utilization is important to achieving performance
- Modeling and analysis of this cache architecture
- Sampler
  - Updated for Xeon Phi vector instructions
  - Scaling up the sampler for many-core thread parallelism
- Project is still ongoing
Thanks!

• Talk to us here at CUG

• Contact me at: chris.gottbrath@roguewave.com
  – Sign up for the TotalView 8.12 beta (Xeon Phi)
  – Learn more about ThreadSpotter
  – Feedback, suggestions, use cases

• Learn more at: www.roguewave.com
  – White papers
  – Product Documentation
  – Videos
  – Product evaluation