Debugging in Heterogeneous Environments with TotalView

ECMWF HPC Workshop
30th October 2014
Agenda

- Introduction
- Challenges
- TotalView overview
- Advanced features
- Current work and future plans
Introduction
Rogue Wave

What we do

Rogue Wave helps organizations simplify complex software development, improve code quality, and shorten cycle times

- History
  - Founded: 1989
  - Portfolio company of Audax Group
  - Acquisitions:
    - Visual Numerics: 2009
    - TotalView Technologies: 2010
    - ILOG Visualization for C++ : 2012
    - OpenLogic : 2013
    - Klocwork : 2013
    - ILOG Visualization for Java: 2014

- Customers
  - 3,000+ in 57 countries
  - Financial services, telecoms, oil and gas, government and aerospace, research and academic

- Global Locations
  - HQ: Boulder, CO
  - NA: Houston, TX; Corvallis, OR; Natick MA
  - EMEA: France, Germany, UK
  - APAC: Japan
Challenges of developing for heterogeneous environments
Challenges

- Number of CPU cores increasing but clock speed is static or decreasing
- How to program accelerators
  - Lower level languages (OpenCL, CUDA)
  - Directives based (OpenACC, OpenMP)
- New algorithms or programming models may be needed
- Data sizes increasing exponentially
- Memory is increasingly important
- Power consumption and constraints
How does Rogue Wave help?

TotalView debugger

• Troubleshooting and analysis tool
  – Visibility into applications
  – Control over applications

• Scalability

• Usability

• Support for HPC platforms and languages
TotalView Overview
What is TotalView®?

Application Analysis and Debugging Tool: Code Confidently

- Debug and Analyse C/C++ and Fortran on Linux™, Unix or Mac OS X
- Laptops to supercomputers
- 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™
  - CUDA™, OpenACC®, and Intel® Xeon Phi™ coprocessor
- Low tool overhead resource usage
- Includes a Remote Display Client which frees 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

© 2014 Rogue Wave Software, Inc. All Rights Reserved
• **Runtime Memory Analysis : Eliminate Memory Errors**
  – Detects memory leaks *before* they are a problem
  – Explore heap memory usage with powerful analytical tools
  – Use for validation as part of a quality software development process

• **Major Features**
  – Included in TotalView, or Standalone
  – Detects
    • Malloc API misuse
    • Memory leaks
    • Buffer overflows
  – Supports
    • C, C++, Fortran
    • Linux, Unix, and Mac OS X
    • Intel® Xeon Phi™
    • MPI, pthreads, OMP, and remote apps
  – Low runtime overhead
  – Easy to use
    • Works with vendor libraries
    • No recompilation or instrumentation
Deterministic Replay Debugging

- Reverse Debugging: Radically simplify your debugging
  - Captures and Deterministically Replays Execution
    - Not just “checkpoint and restart”
  - Eliminate the Restart Cycle and Hard-to-Reproduce Bugs
  - Step Back and Forward by Function, Line, or Instruction

- Specifications
  - A feature included in TotalView on Linux x86 and x86-64
    - No recompilation or instrumentation
    - Explore data and state in the past just like in a live process, including C++View transformations
  - Replay on Demand: enable it when you want it
  - Supports MPI on Ethernet, Infiniband, Cray XE Gemini
  - Supports Pthreads, and OpenMP
  - New: Save / Load Replay Information (CLI only)
• Survey conducted by the Judge Business School at Cambridge University concluded that Reverse Debuggers allow users, on average, to spend 13% less of their programming time debugging.
  – Programming was 50% of total work week on average
  – Debugging was 50% of programming time without reverse debugging
  – Debugging was 37% of programming time with reverse debugging
  – That frees up 130 hours (>3 work weeks, 6.5% total time) per developer per year for design and new feature development

• The survey looked at total value (salaries & overhead) of debugging as a task and they determined that this savings could, across the whole world economy, be work $41 billion in increased productivity.
  – The productivity improvement should be worth $2,500 per developer per year (salary only) or $5,000 per year with overhead.

TotalView for the NVIDIA® GPU Accelerator

- NVIDIA CUDA 6.5
  - With support for Unified Memory
- Features and capabilities include
  - Support for dynamic parallelism
  - Support for MPI based clusters and multi-card configurations
  - Flexible Display and Navigation on the CUDA device
    - Physical (device, SM, Warp, Lane)
    - Logical (Grid, Block) tuples
  - CUDA device window reveals what is running where
  - Support for types and separate memory address spaces
  - Leverages CUDA memcheck
Displaying NVIDIA GPU Device Information
TotalView for OpenACC

- Step host & device
- View variables
- Set breakpoints

Compatibility with Cray CCE 8 OpenACC
Supports All Major Intel Xeon Phi Coprocessor Configurations
- Native Mode
  - With or without MPI
- Offload Directives
  - Incremental adoption, similar to GPU
- Symmetric Mode
  - Host and Coprocessor
- Multi-device, Multi-node
- Clusters

User Interface
- MPI Debugging Features
  - Process Control, View Across, Shared Breakpoints
- Heterogeneous Debugging
  - Debug Both Xeon and Intel Xeon Phi Processes

Memory Debugging
- Both native and symmetric mode
Spectrum of Intel Xeon Phi Execution Models

CPU-Centric

- General purpose serial and parallel computing
  - Main()
  - Foo()
  - MPI_*()

Intel® Xeon Phi-Centric

- 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
Debugging Intel Xeon Phi Applications with Offloaded Code

One debugging session for MIC-accelerated code
Debugging Intel Xeon Phi MPI Applications

- Start multi-host multi-card MPI job
- Attach to subset of processes on MIC coprocessor
- Set breakpoints
- Debug “as usual” MPI
Coarray Fortran

Diving on CAF array $y$

Supported on Cray platforms with CCE
Advanced Features
Remote Display Client

• Offers users the ability to easily set up and operate a TotalView debug session that is running on another system

• Consists of two components
  • Client – runs on local machine
  • Server – runs on any system supported by TotalView and “invisibly” manages the secure connection between host and client

• Remote Display Client is available for:
  • Linux x86, x86-64
  • Windows XP, Vista, 7
  • Mac OS X
Multi-dimensional array viewer

- See your arrays on a Grid display
- 2-D, 3-D, … N-D
- Arbitrary slices
- Specify data representation
- Windowed data access – Fast
Visualizing Arrays

- Visualize array data using Tools > Visualize from the Variable Window
- Large arrays can be sliced down to a reasonable size first
- Visualize is a standalone program
- Data can be piped out to other visualization tools

- Visualize allows to spin, zoom, etc.
- Data is not updated with Variable Window; You must revisualize
- $\text{visualize}()$ is a directive in the expression system, and can be used in evaluation point expressions.
Debugging MPMD applications

```
totalview -args aprun -n 9 worker : -n 1 master
```

![Debugging MPMD applications](image)
Message Queue Debugging

- Filtering
  - Tags
  - MPI Communicators
- Cycle detection
  - Find deadlocks
• Gives you non-interactive access to TotalView’s capabilities
• Useful for
  – Debugging in batch environments
  – Watching for intermittent faults
  – Parametric studies
  – Automated testing and validation
• TVScript is a script (not a scripting language)
  – It runs your program to completion and performs debugger actions on it as you request
  – Results are written to an output file
  – No GUI
  – No interactive command line prompt
• Used at sites such as DMI and STFC Daresbury for automated comparative debugging
C++View

- C++View is a simple way for you to define type transformations
  - Simplify complex data
  - Aggregate and summarize
  - Check validity

- Transforms
  - Type-based
  - Compose-able
  - Automatically visible

- Code
  - C++
  - Easy to write
  - Resides in target
  - Only called by TotalView
VIEWING FORTRAN USER-DEFINED TYPES

TYPE WHOPPER

  LOGICAL, DIMENSION(ISIZE) :: FLAGS
  DOUBLE PRECISION, DIMENSION(ISIZE) :: DPSA
  DOUBLE PRECISION, DIMENSION(:), POINTER :: DPPA

END TYPE WHOPPER

TYPE(WHOPPER), DIMENSION(:,), ALLOCATABLE :: STUFTYP1
Current Work and Future Plans
What is new in TotalView 8.14.1

- CUDA 6.5 support
- Coarray Fortran support for the Cray CCE compiler
- Extended support for type transformations with the Intel compiler (unordered STL collection classes)
- Improved delayed symbol processing (better performance for larger executables)
Multi-phase R&D Projects Underway

• Massive Scalability
  – Collaboration with LLNL and Tri-lab partners
  – Targeting Cray, Blue Gene and Linux Clusters
  – MRNet software overlay network for multicast and reduction

• 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
Thanks!

- Visit the website
  - Videos (3 new videos on Xeon Phi)
  - Documentation
  - Sign up for an evaluation

- Visit us at SC14 (booth 2338)