TotalView: Debugging from Desktop to Supercomputer

Peter Thompson
Principal Software Support Engineer
August 12, 2015
Rogue Wave helps organizations *simplify* complex software development, *improve* code quality, and *shorten* cycle times.
Capabilities

- **Application Security**: Klocwork, OpenLogic, TotalView, IMSL, SourcePro
- **Code Refactoring**: Klocwork
- **Code Review**: Klocwork, OpenLogic
- **Debugging Complex Code**: Klocwork, TotalView
- **Reusable Math Algorithms**: IMSL, SourcePro
- **Open Source Auditing**: OpenLogic
- **Open Source Management**: OpenLogic
- **Open Source Support**: OpenLogic
- **Certified Open Source**: OpenLogic
- **Static Code Analysis**: Klocwork
- **Developing User Interfaces**: Visualization, Stingray, PV-WAVE
- **Code Migration**: SourcePro, IMSL, HydraExpress
- **Code Building Blocks**: SourcePro, IMSL, Stingray, Visualization
Used by 3,000 customers in over 57 countries across diverse industries to develop mission-critical applications and software.

Financial Services
- Bank of America
- Citibank
- Credit Suisse
- Liberty Mutual
- Goldman Sachs
- AT&T
- Huawei
- Alcatel-Lucent
- GenCorp
- Sprint
- BlackBerry
- Deutsche Bank
- JPMorgan Chase

Telecom
- BT
- Vodafone
- MIK
- Motorola
- Sprint
- Polycom
- Qualcomm

Gov't / Defense
- Boeing
- NASA
- Northrop Grumman
- BAE Systems
- Raytheon
- Lawrence Livermore National Laboratory
- Lockheed Martin
- Sandia National Laboratories

Technology
- Cisco
- HP
- Oracle
- IBM
- CA
- BMC Software
- EMC
- Dell
- RSA Security
- Blackberry
- Sandia National Laboratories

Other Verticals
- Pixar
- Exodus
- Schlumberger
- Delphi
- Disney
- Industrial Light & Magic
- Shell
- Cerner
- Olympus
- Siemens
- Schneider Electric
- Total
- John Deere
Debugging occurs in many industries

They use software to deliver value or inform decisions

Financial services
Oil and Gas
Aerospace and Defense
Engineering
Digital Content Creation
ISVs
Biological sciences
Scientific and Technical Computing
High Performance Computing
HPC Trends

• What do we see
  – NVIDIA Tesla GP-GPU computational accelerators
  – Intel Xeon Phi Coprocessors
  – Complex memory hierarchies (numa, device vs host, etc)
  – Custom languages such as CUDA
  – Directive based programming such as OpenACC and OpenMP
  – Core and thread counts going up

• A lot of complexity to deal with if you want performance
  – C or Fortran with MPI starts to look “simple”
  – Everything is Multiple Languages / Parallel Paradigms
  – Up to 4 “kinds” of parallelism (cluster, thread, heterogeneous, vector)
  – Data movement and load balancing
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
- Student Express license available for both undergraduate and grad students
TotalView Overview
TotalView Origins

Mid-1980’s Bolt, Berenak, and Newman (BBN) Butterfly Machine
An early ‘Massively Parallel’ computer
How do you debug a Butterfly?

- TotalView project was developed as a solution for this environment
- Able to debug multiple processes and threads
- Point and click interface
- C, C++, and Fortran (and assembler)
A solution in search of a problem…

• From the Butterfly, TotalView was ported to other machines
  – IBM RS6000, Cray, Solaris Sparc, DEC Alpha, Irix…

• As various MPI’s were being developed, Bill Gropp, Rusty Lusk and Jim Cownie worked on an interface for automatic process acquisition

• Some years later, Bill and Jim developed another interface for showing Message Queue information
Other capabilities added

- Support for most types of MPI
- Linux
- Lightweight Memory Debugging
- Type transformations
- Memscript and tvscript
- Reverse Debugging
- Remote Display Client
- GPU debugging
- Intel Xeon Phi
Key features of TotalView

- Interactive Debugging
- Interactive Memory Debugging
- Reverse Debugging
- Unattended Debugging

Serial, Parallel and Accelerated applications
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™, Fortran Coarrays
  - 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
What Is MemoryScape®?

- 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
Memscript and Tvscript

• Command line invocation to run TotalView and Memoryscape unattended

• Tvscript can be used to set breakpoints, take actions at those breakpoints and have the results logged to a file. It can also do memory debugging
  – tvscript –create_actionpoint “method1=>display_backtrace show_arguments” \\ -create_actionpoint “method.c#342=>print x” myprog –a dataset 1

• Memscript can be used to run memory debugging on processes and display data when a memory event takes place. Exit is ALWAYS an event

  Memscript -event_action \
  "alloc_null=list_allocations,any_event=check_guard_blocks” \
  -guard_blocks -maxruntime "00:30:00” -display_specifiers \
  "noshow_pc,noshow_block_address,show_image”\n  myProgram -a myProgramArg1

• Memscript data can be saved in html, memory debug file, text heap status file
Dassault Systems

- “MemoryScape enabled us to identify memory issues, and by using its scripting interface, we were able to automate the evaluation process. Now, the system automatically uncovers any hidden latent errors in our code with every build, allowing our developers to proactively fix potential errors prior to release.”
  – Nick Monyatovsky, Software Engineer at SIMULIA

- Computer Aided Engineering ISV for Aero/Auto/Industry

- Struggling with intermittent errors

- Continuous Integration – Better Product Quality
Remote Display Client (RDC)

- Push X11 bits and events across wide networks can be painful. The RDC can help.

Figure 17 – Remote Display Components
The RDC setup

1. Enter the Remote Host to run your debug session:
   - Remote Host: vesta.alcf.anl.gov
   - User Name: thompson
   - Advanced Options

2. As needed, enter hosts in access order to reach the Remote Host:
   - Host Access By Access Value Commands
     - 1 User Name
     - 2 User Name

3. Enter settings for the debug session on the Remote Host:
   - Path to TotalView on Remote Host: /soft/debuggers/totalview/bin/totalview
   - Arguments for TotalView:
   - Your Executable (path & name): runjob
   - Arguments for Your Executable: $ -p 1 --np 512 --block ${COBALT_PARTNAME} : ALLc2
   - Submit Job to Batch Queueing System: Custom
   - TotalView
   - MemoryScape

4. Enter batch submission settings for the Remote Host:
   - Submit Command: qsub
   - Script to execute via Submit Command: ./tv_PBS.csh
   - Additional Submit Command Options: -q ATPESC2015 -t 60 -n 512 --mode script -O LOG

Launch Debug Session
Support for New Platforms
TotalView for the NVIDIA® GPU Accelerator

- **NVIDIA CUDA 6.0, 6.5 and 7.0**
- **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 CUDA Core debugging
  - Leverages CUDA memcheck
  - Support for OpenACC
TotalView for the Intel® Xeon Phi™ coprocessor

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

Anticipated support for KNL in late 2015 – early 2016
Knights Landing Memory

• KNL will have on-board High Bandwidth Memory (HBM) which can be accessed much faster than going out to main memory.
  – Cache
  – Explicitly managed for placement of frequently accessed data

• MemoryScape will be able to track allocations made both the standard heap and the on-chip HBM

• Optimization may include making sure that the right data structures are available to the processor in HBM
  – MemoryScape can show you data structure usage and placement
Linux OpenPower (LE) support

- Experimental support for OpenPower (Linux power LE)
  - All major functionality
  - Support for CUDA Debugging on GPU Accelerators
  - Contact Nikolay.Piskun@roguewave.com
Current Work and Future Plans
New Features

- Scalable Infrastructure
  - Faster start up on Linux
  - Scales to $O(100,000)$ processes & $O(1,000,000)$ threads
- Updated CUDA support
  - CUDA 7.0
- Support updates including:
  - Clang 3.5
  - Intel 15.0
  - MPT 2.12
  - SLES 12, Fedora 21
TotalView’s Scalability Strategy

Push debugger “smarts” to the backend, not the whole debugger!

TotalView uses an “MRNet tree” of servers

Use classic optimization techniques too: caching, hoisting invariants, etc.

Remain lightweight in the backend!

© 2015 Rogue Wave Software, Inc. All Rights Reserved
TotalView’s Memory Efficiency

- TotalView is lightweight in the back-end (server)
- Servers don’t “steal” memory from the application
- Each server is a multi-process debugger agent
  - One server can debug thousands of processes
  - Not a conglomeration of single process debuggers
  - TotalView’s architecture provides flexibility (e.g., P/SVR)
  - No artificial limits to accommodate the debugger (e.g., BG/Q 1 P/CN)
- Symbols are read, stored, and shared in the front-end (client)
- Example: LLNL APP ADB, 920 shlibs, Linux, 64 P, 4 CN, 16 P/CN, 1 SVR/CN

<table>
<thead>
<tr>
<th>Process</th>
<th>VSZ (largest, MB)</th>
<th>RSS (largest, MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV Client</td>
<td>4,469</td>
<td>3,998</td>
</tr>
<tr>
<td>MRNet CP</td>
<td>497</td>
<td>4</td>
</tr>
<tr>
<td>TV Server</td>
<td>304</td>
<td>53</td>
</tr>
</tbody>
</table>

© 2014 ROGUE WAVE SOFTWARE, INC. ALL RIGHTS RESERVED
5x faster at 16k between 8.14.1 and 8.15.4. Note that we switched to MRNET by default in 8.15.0.
BG Start up Performance in TV 8.15.4

6.4x faster at 16k between 8.14.1 and 8.15.4.
Note that we switched to MRNET by default in 8.15.0
TotalView debugs 786,432 cores. Climb with Rogue Wave towards exascale.
New Revolutionary UI Framework

- **Design** your own Debugger/Tool
- Personalize to your likeness.
- In **Evergreen Beta** soon
More Information

- Product documentation on website:
  http://www.roguewave.com/help-support/documentation/totalview
- Contact sales@roguewave.com with any inquiries about our future plans with regard to TotalView product.
Questions
Thanks!

- Visit the website
  - Documentation
  - Sign up for an evaluation
  - Contact customer support & post on the user forum