Leap to Petascale

Developing Developer Tools
TotalView and Blue Gene/Q

May 23 2012
Agenda

- Who is Rogue Wave?
- Early Blue Gene Days with TotalView
- Blue Gene/Q Advancements
- Techniques for Debugging Challenges
- What's New with TotalView
Who is Rogue Wave Software?
Solution Portfolio

Enterprise C++
Development

Developing for Parallel Architectures
Parallel, Data-intensive Applications

Leveraging High Performance
Data Analytics

SourcePro C++
Visualization

TotalView
IMSL
PV-WAVE
MemoryScape
ThreadSpotter
Early Blue Gene Days with TotalView
TotalView Blue Gene Support

- TotalView involvement started in 2003 on BG/L

Gotta love that yellow duct tape!
TotalView Blue Gene/L Support

- Support for Blue Gene/L since 2005
- Debugging interfaces developed via close collaboration with IBM (CIOD)
- Used on DOE/NNSA/LLNL's Blue Gene/L system containing 212 K cores
  - Heap memory debugging support added
  - Blue Gene/L scaling and performance tuning project
  - TotalView has debugged jobs as large as 32,768 processes

Blue Gene/L work facilitated Blue Gene/P support
TotalView Blue Gene/P Support

- Continued close collaboration with IBM
- Currently running on several BG/P installations in Germany, France, the UK, and the US.
- Support for shared libraries, threads, and OpenMP
- TotalView has debugged jobs as large as 32,768
- Active workshop participation through the development
  - ANL’s ALCF INCITE Performance Workshop
  - Jülich’s Blue Gene/P Porting, Tuning, and Scaling Workshops
Porting TotalView began in June 2011
Access to Q32 at IBM began in August
Basic debugging operations in October
Used in Synthetic Workload Testing in December
Fully functional in March 2012
TotalView Blue Gene/Q Support (cont)

- Thanks to the ongoing collaboration with IBM and the BG Kernel Team, early access versions of TotalView for BG/Q is available.
- Argonne National Laboratory
- Lawrence Livermore National Laboratory

Blue Gene/Q packaging hierarchy
Blue Gene/Q Advancements with TotalView
TotalView on BG/Q Support

• BG/Q TotalView is as functional as BG/P TotalView
  – MPI, OpenMP, pthreads, hybrid MPI+threads
  – C, C++, Fortran, assembler; IBM and GNU compilers
  – Basics: source code, variables, breakpoints, watchpoints, stacks, single stepping, read/write memory/registers, conditional breakpoints, etc.
  – Memory debugging, message queues, binary core files, etc.

• PLUS, features unique to BG/Q TotalView
  – QPX (floating point) instruction set and register model
  – Fast compiled conditional breakpoints and watchpoints
  – Asynchronous thread control

• Working with IBM on debugging interfaces for TM/SE regions
  – TM == transactional memory; SE == speculative execution
Advanced TotalView Features on BG/Q

• Asynchronous thread control
  – A feature on Linux, and other TotalView platforms, ported to BG/Q
  – Allows you to individually control the execution of threads
  – Run and halt individual threads
  – Single-step a group of threads in lockstep
  – Hold and release the execution of individual threads
  – Create stop-thread and thread barrier breakpoints

• Fast compiled conditional breakpoints and watchpoints
  – A feature on AIX and other TotalView platforms, ported to BG/Q
  – Conditional breakpoints and watchpoints execute in as little as 7 μsecs
  – Conditional expressions are compiled and dynamically patched into the process
  – Evaluation is performed by the triggering thread, in parallel
Hierarchical infrastructure components are distributed throughout the system.
Tool Challenges

One rack of BG/Q:
1K CNs, 16K Cores, 64K HW Threads

A “generous”
128:1 CN:ION ratio: 8 IONs

A “beefy” FEN
P7, 3 GHz+, 32 GB+

Compared to the size of target application, there’s not much “room” for the tool!
Overcoming High CN:ION Ratios

• On BG/Q, at a given ratio, on each IO node, tool daemons may be responsible for up to

<table>
<thead>
<tr>
<th>CN:ION</th>
<th>Processes</th>
<th>Threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>64:1</td>
<td>4,096</td>
<td>20,480</td>
</tr>
<tr>
<td>128:1</td>
<td>8,192</td>
<td>40,960</td>
</tr>
<tr>
<td>256:1</td>
<td>16,384</td>
<td>81,920</td>
</tr>
<tr>
<td>512:1</td>
<td>32,768</td>
<td>163,840</td>
</tr>
</tbody>
</table>

• But each IO node has
  – 1.6 GHz A2 17 core processor (not too swift)
  – 16 GB (limited physical memory)
Where to put the “weight” of the debugger?

- Most of the “weight” of the debugger is in the symbol table
- Real-world applications are huge and complex
- A recently analyzed mission critical application revealed
  - 1.5 million function definitions
  - 16 million line number definitions
  - DWARF symbol information in excess of 2 GB
  - 100s or 1000 of shared libraries

- You don’t want to be big in the back end!
- And nothing too compute intensive either
TotalView’s Architecture

- Extremely lightweight back-end daemon processes
  - Small footprint plus a few hundred bytes per CN process or thread
  - Each daemon can handle thousands of processes and threads
  - The daemons do not store the symbol table!
- The “weight” of the debugger is on the front-end node
- Symbol tables are indexed and stored on the FEN
  - Debugger has exactly one copy of the symbol table for each image file
  - Symbol tables are shared across all processes and thread
  - Aggregate memory consumption is minimal
- Well suited to Blue Gene!
There’s still the high P/T count per IO node problem

- Process and threads counts per IO node are still high!

- What to do about that?

- “Divide and conquer”
  - Place a small number of daemons on the ION
  - We do have 17 cores we can use

- Unlike CIOD on BG/L&P, CDTI on BG/Q can operate in parallel
  - There’s one CDTI debug channel per compute node
Solution: TotalView/MRNet Trees on the IO Nodes

The MRNet Commnode Process connects the daemons to the rest of the tree.

Instead of one daemon managing all 128 CNs.
TotalView on BG/Q Support

• **BG/Q TotalView is as functional as BG/P TotalView**
  – MPI, OpenMP, pthreads, hybrid MPI+threads
  – C, C++, Fortran, assembler; IBM and GNU compilers
  – Basics: source code, variables, breakpoints, watchpoints, stacks, single stepping, read/write memory/registers, conditional breakpoints, static/dynamic executables, etc.
  – Memory debugging, message queues, binary core files, etc.

• **PLUS, advanced BG/Q TotalView features**
  – QPX (floating point) instruction set and register model
  – Fast compiled conditional breakpoints and watchpoints
  – Asynchronous thread control

• **Working with IBM on debug interfaces for TM/SE regions**
  – TM == transactional memory; SE == speculative execution
Advanced BG/Q TotalView Features

• **Asynchronous thread control**
  - A TotalView feature on Linux and other platforms, ported to BG/Q
  - Allows you to individually control the execution of threads
  - Run and halt individual threads
  - Single-step a group of threads in lockstep
  - Hold and release the execution of individual threads
  - Create stop-thread and thread barrier breakpoints

• **Fast compiled conditional breakpoints and watchpoints**
  - A TotalView feature on AIX and other platforms, ported to BG/Q
  - Conditional breakpoints and watchpoints execute in as little as 7 µsecs
  - Conditional expressions are compiled and dynamically patched into the process
  - Evaluation is performed in parallel by the triggering thread
TotalView Availability

- **TotalView on Blue Gene/Q Today**
  - LLNL has it up and running on rzuseq, and is using it to debug applications.
  - IBM is using it internally for debugging and testing.
  - It's installed on IBM's Blue Gene On Demand Center Q32 (if anyone has access to that system).

- **TotalView On Blue Gene at Argonne**
  - 1024 Tokens (BG/P)
  - Research license is available with 65,536 tokens
TotalView on VEAS!
TotalView on VEAS!
Techniques for Debugging Challenges
What is TotalView?

A comprehensive debugging solution for demanding parallel and multi-core applications

- **Wide compiler & platform support**
  - C, C++, Fortran 77 & 90, UPC
  - Unix, Linux, OS X

- **Handles Concurrency**
  - Multi-threaded Debugging
  - Multi-process Debugging

- **Integrated Memory Debugging**

- **Reverse Debugging available**

- **Supports Multiple Usage Models**
  - Powerful and Easy GUI – Highly Graphical
  - CLI for Scripting
  - Long Distance Remote Debugging
  - Unattended Batch Debugging
Debugging Complex Codes

• Mechanize
• Minimize
• Visualize
• ... and Don’t Forget the Memory
Automated Debugging

TVscript

• Non-Interactive Batch Debugging –
  – Work in the “main” batch queue
  – Don’t have to baby-sit job waiting on it to run
  – Use scripting to perform checks that would be tedious to do by hand
  – Verification through automated processes (nightly build and test)

TTF and C++View

• Automatic Transformation of Data –
  – Simplify interactive (and scripted) debugging
  – Perform validation/sanity checking of large datasets
  – Comparative debugging
  – Allows you to focus on troubleshooting your program
Non-Interactive Batch Debugging with TVscript

- Run multiple debugging sessions without the need for recompiling, unlike with printf
- TVscript syntax:
  
  \texttt{tvscript [ options ] [ filename ] [ -a program\_args ]}

- More complex actions-to-events are possible, utilizing TCL within a CLI file
- TVscript lets you define what events to act on, and what actions to take

<table>
<thead>
<tr>
<th>TVscript uses a simple, Event/Action interface</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Typical Events</strong></td>
</tr>
<tr>
<td>• Action_point</td>
</tr>
<tr>
<td>• Any_memory_event</td>
</tr>
<tr>
<td>• Guard_corruption_error</td>
</tr>
<tr>
<td>• Print [-slice \textit{slice_exp} \textit{variable}</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Unattended Debugging with Tvscript

Example
The following tells tvscript to report the contents of the foreign_addr structure each time the program gets to line 85:
-create_actionpoint "#85=>print foreign_addr"

Typical output sample with tvscript:

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Customize your own Transformations

In `$HOME/.tvdrc`:

```cpp
::TV::TTF::RTF::build_struct_transform {
  name {^class x1$}
  members {
    { pmonth { month } }
    { pName { xbase upcast { * pName } } }
    { pStreet { xbase upcast { * pStreet } } }
    { pVoid1 { "$string *" cast v } }
    { pVoid2 { * { "class x2 *" cast q } } }
  }
}
```

**Meta Language:**

- `{member}
- `{expr
- `{expr . Expr
- `{expr -> expr
- `{datatype case expr
- `{baseclass upcast expr

---

Creating Type Transformations
C++View

- C++View is an easy way to customize TotalView’s display of object data.
- How does it work?
  - User writes short display functions within their program
  - TotalView uses these functions to simplify the display of data when the user explores their data within that program
  - C++View transforms are easy to define
  - Great for collaborative codes (transforms can be distributed with the program)
- Benefit: Easier for scientists and developers to work with complex applications

Developers can now write display and analysis functions for their C++ classes that are invoked whenever an object is inspected interactively in the debugger.
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
Minimize
Reduce the Scope of Effort
Subset Debugging
With TotalView
subset attach

you need not be attached to the entire job

- you can be attached to different subsets at different times through the run
- you can attach to a subset, run till you see trouble and then 'fan out' to look at more processes if necessary.
- this greatly reduces overhead
- it also reduces license size requirements
Message Queue Graph

- Hangs & Deadlocks
- Pending Messages
  - Receives
  - Sends
  - Unexpected
- Inspect
  - Individual entries
- Patterns
Message Queue Debugging

- Filtering
  - Tags
  - MPI
  - Communicators
- Cycle detection
  - Find deadlocks
Get the big picture – Observe anomalies – Utilize Pattern recognition – Save time!
... And Don’t Forget the Memory!
Memory bugs often go undetected until the worst possible time
• Symptoms often surface long after the actual damage is done
• Some only surface after hours or even days of operation
• In many cases, the programs affected are “innocent bystanders”

MemoryScape: Fully Integrated in TotalView
No Source Code or Binary Instrumentation
• Use it with your existing builds
• Programs run nearly full speed
• Low performance overhead
• Low memory overhead • Efficient memory usage

MemoryScape Feature Highlights
• Automatic allocation problem detection
• Heap Graphical View
• Leak detection
• Block painting
• Dangling pointer detection
• Deallocation/reallocation notification
• Memory Corruption Detection - Guard Blocks
• Memory Hoarding
• Memory Comparisons between processes
• Collaboration features
What’s Coming

• **Increased Scalability**
  – Leveraging TotalView’s Architecture
  – Efficient Use of Cluster Resources
    • Extremely light weight debug agents; Minimal memory footprint
    • More space on the compute nodes for user application code
  – Tree-Based Overlay Network
    • Broadcast of Operations; Aggregation of Events and Data

• **Advanced User Interface**
  • New GUI Framework
    • Changes focused on extreme scale debugging

• **CUDA 4.1 now; 4.2 and 5.0 this year**

• **Replay Enhancements**
  – Record on Demand (in Beta)
  – Replay Debug from Core File

• **OpenACC Support**

• **Intel MIC Support**
  – Come see a demo at ISC ‘12
Developing for Parallel Architectures

• Code debugging
  • Highly scalable interactive GUI debugger
    – Easy to use -- without sacrificing detail that users need to debug
    – Used from workstations to the largest supercomputers
  • Powerful features for debugging multi-threaded, multi-process, and MPI parallel programs
  • Compatible with wide variety of compilers across several platforms and operating systems

• Memory Debugging
  • Parallel memory analysis and error detection
    – Intuitive for both intensive and infrequent users
  • Easily integrated into the validation process

• Reverse Debugging
  • Parallel record and deterministic replay within TotalView
    • Users can run their program “backwards” to find bugs
  • Allows straightforward resolution of otherwise stochastic bugs

• GPU CUDA Debugging
  • Full Hybrid Architecture Support
  • Asynchronous Warp Control
  • Multi-Device and MPI Support
Developing parallel, data-intensive applications is hard. We make it easier.

www.roguewave.com