Large Scale Debugging on Titan and Mira with Allinea DDT

David Lecomber
Allinea Software
david@allinea.com
Allinea’s Tools

- Reduce Development Time
- Increase Application Performance
- Understand Application Behaviour
HPC could be brain surgery

• Brain aneurysms
  – 2-5% of population – most are undiagnosed
  – 30,000 rupture in US each year – 40% fatal
  – Early discovery and treatment increases survival rates

• Neurosurgery as HPC
  – MRI provides the blood vessel structure
  – Intra-cranial blood flow and pressures is just complex CFD
  – Full brain 3D model is 2-10GB geometry
Impact of Petascale and Beyond

• Individualized HPC
  – Patient’s MRI scan enables surgical decision: whether to operate, how to operate, …
  – Circle of Willis requires super-Petascale machine software
  – Need answer in minutes or hours

• Machines can do 20 PetaFLOPs
  – Super-Petascale will be affordable soon
  – Software has to scale
Real scaling challenge

• Crashes at 49,152 cores on Cray XC30
  – Error message “Terminated”. Thanks.

• Now what?
  – Try other (inferior?) partitioner?
  – Invest weeks in bug fix by trial and error?
  – Write own partitioning library?

• Why use a debugger?
  – It’s about time
Run at problem size (49,152 processes)

... a debugger!

Ah... Integer overflow!
Debugging in practice…

1. Compile
2. Run
3. Crash
4. Hypothesis
5. Insert print statements

Flow: Compile → Run → Crash → Hypothesis → Insert print statements → Compile
## Some types of bug

<table>
<thead>
<tr>
<th>Bug Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bohrbug</td>
<td>Steady, dependable bug</td>
</tr>
<tr>
<td>Heisenbug</td>
<td>Vanishes when you try to debug (observe)</td>
</tr>
<tr>
<td>Mandelbug</td>
<td>Complexity and obscurity of the cause is so great that it appears chaotic</td>
</tr>
<tr>
<td>Schroedinbug</td>
<td>First occurs after someone reads the source file and deduces that it never worked, after which the program ceases to work</td>
</tr>
</tbody>
</table>
Print statement debugging

• The first debugger: print statements
  – Each process prints a message or value at defined locations
  – Diagnose the problem from evidence and intuition

• A long slow process
  – Analogous to bisection root finding

• Broken at modest scale
  – Too much output – too many log files
Bug fixing as scale increases

Reproduce at a smaller scale?

- Reduced data set - may not trigger the problem?
- Didn't you already try the code at small scale?
- Is it a system issue?
- Is probability stacking up against you?

Debugging at extreme scale is a necessity
Extreme machines are everywhere

Software scale grows as machines grow

Machine sizes are exploding

No. 1
No. 100
No. 500

CPU Cores

2010 2011 2012 2013 2014
Titan and Mira

Titan

- 18,688 nodes
- 18,688 NVIDIA Kepler K20 GPUs
- 299,008 CPU cores
- 50,233,344 CUDA cores

Mira

- 49,152 nodes
- 786,432 cores
- 3,145,728 hardware threads

Does the printf workflow “work”?
ALCF, OLCF and Allinea deliver

2009 - Allinea and Oak Ridge begin collaboration to provide super-Petascale debugging

2010 - Allinea and Argonne collaboration to extend scaling to BlueGene systems

2013 - Mira and Titan full size debugging in place
What you should expect (demand!) for debugging at scale

Scalability
- A debugger that works to at least as high a scale as you need

Hardware and software support
- Whatever software you use and wherever you use it – the debugger supports it

Assistance
- Debugger is installed, configured, and documented – with site experts and training
Allinea DDT
Fix software problems, fast

- Powerful graphical debugger designed for:
  - C/C++, Fortran, UPC, …
  - MPI, OpenMP and mixed-mode code
  - Accelerators and coprocessors: CUDA and Intel Xeon Phi

- Unified interface with Allinea MAP:
  - One interface eliminates learning curve
  - Spend more time on your results

- Slash your time to debug
  - Reproduces and triggers your bugs instantly
  - Helps you easily understand where issues come from quickly
  - Helps you to fix them as swiftly as possible
Allinea DDT: Scalable debugging by design

**Where did it happen?**
- Allinea DDT leaps to source automatically
- Merges stacks from processes and threads

**How did it happen?**
- Some faults evident instantly from source

**Why did it happen?**
- Real-time data comparison and consolidation
- Unique “Smart Highlighting” – colouring differences and changes
- Sparklines comparing data across processes

**Force crashes to happen?**
- Memory debugging makes many random bugs appear every time
Five great things to try with Allinea DDT

- The scalable print alternative
- Stop on variable change
- Static analysis warnings on code errors
- Detect read/write beyond array bounds
- Detect stale memory allocations
Beneath the Petascale Allinea DDT

- **Scalable tree network**
  - Sends bulk commands and merge responses
  - Aggregations maintain the essence of the information
  - Step 100,000 processes? 100-150ms

- **Usability matters**
  - The interface is as important as the speed
  - Focus on scalable components
Example – ORNL’s Titan

- HPC code fails on 98,304 cores
- Random processes crashing
- Printf? Which processes and where?
- Too costly to repeat
- Allinea DDT finds cause first time
Offline debugging

- Interactive access difficult
- Use offline mode
  - Submit and forget
- Post-mortem analysis
Example – ANL Mira

1. HPC code fails on 16,384 cores
2. Code abandoned – bug couldn’t be fixed
3. Machine too busy for interactive debugging
4. Allinea DDT offline mode runs bug case overnight
5. Found error in initialization
Interlude: Local Demonstration

- Simple persistent hanging
  - Stepping through a code

- Process count dependent hanging:
  - Attaching to the running job
Getting started on Titan

• How?
  module load ddt
ddt
• Congratulations, you are now ready to debug.
Getting started on Mira/Tukey

- Install local client on your laptop
  - www.allinea.com/downloads
    - Linux – installs full set of tools
    - Windows, Mac – just a remote client to the remote system
  - Run the installation and software
  - “Connect to remote host”
  - Hostname:
    - username@cetus.alcf.anl.gov
    - username@tukey.alcf.anl.gov
  - Remote installation directory: /soft/debuggers/ddt
  - Click Test
- Congratulations you are now ready to debug.
Favorite Allinea DDT Features for Scale

- Parallel stack view
- Automated data comparison: sparklines
- Parallel array searching
- Step, play, and breakpoints
- Offline debugging
Summary

Debugging at scale is not difficult
- 300,000 cores is as easy as 30 cores
- The user interface is vital to success

Debugging at scale is not slow
- High performance debugging – at Mira and Titan scale
- Logarithmic performance

Stable, in production and well supported
- Routinely used over 100,000 cores

www.allinea.com