Data centric debugging: Scaling to infinity and beyond

David Abramson

Research Computing Centre,
University of Queensland,
Brisbane
Australia

Minh Dinh (UQ)
Chao Jin (UQ)
Luiz DeRose (Cray)
Bob Moench (Cray)
Andrew Gontarek (Cray)
Are you serious???

AN AFTER DINNER TALK ABOUT DEBUGGING?
“Windows on the Universe?”
Purely Academic

Cast

Prof John Holywell  Southern University academic in his early 50's
Prof Martin Godson  Middleton University academic in his mid 50's
Prof Mary Long  Southern University academic in her early 40's
Charles Mittleman  Initially a 30 year old PhD student at Wooton College and Southern University, but then moves to Middleton University as a young academic.
Joanne  Southern University software developer in her mid 30's. Works in Prof Holywell's lab and is pregnant.
Prof Max Williams  St George College academic in his late 50's. Serves as the chair of the Shaw Trust, a not for profit society that supports research projects with grants.
Anna  Middleton University administrator in her 30's. She also serves as an administrator on the Shaw Trust, taking notes and helping with the grant assessment exercises.
Mark  Early career academic
Robin, Cheryl, Newsreader (voices only. Can be played by other actors)

This play starts in mid to late 1990's.

david.abramson@uq.edu.au
BUGS AND DINNER DON’T REALLY MIX
Thenus orientalis

- The United Nations' Food and Agriculture Organization prefers the name flathead lobster, while the official Australian name is Bay lobster.
  - In Australia, it is more widely known as the Moreton Bay bug after Moreton Bay, near Brisbane, Queensland.
How do you catch these bugs?
State of the art in debugging?

printf("%f %f %fn", a[i], b[i], c[i])
a.out

da.out > dumpfile; b.out > dumpfile1
diff dumpfile1 dumpfile2

If debugging is the process of removing software bugs, then programming must be the process of putting them in.

—Edsger Dijkstra—
Debugging large codes

• **Cognitive challenge**
  – Large number of processes
    • Particular problems for UI
  – Large data structures
    • Infeasible to examine individual cells of multi-dimensional, floating point, structures.
  – Heterogeneity
    • A great source of errors
    • Hard to debug when do fail

• **Performance Challenge**
  – High level debugging is expensive
  – Debuggers generally don't use underlying parallel platform

• In the Exascale this just gets worse!
COMPARATIVE DEBUGGING
Debugging Evolved Applications

• Large codes are constantly evolving
  – User requirements
  – Underlying algorithms
  – New architectures

• Subtle errors occur often
  – Programmers spend lots of time debugging
  – Identify the source of a discrepancy
  – Follow it back to original source of deviation
Comparative Debugging

• What is comparative debugging?
  – Data centric approach
  – Two applications, same data
  – Key idea: The data should match
  – Quickly isolate deviating variables
  – Focus is on where deviations occur

• How does this help me?
  – Algorithm re-writes
  – Language ports
  – Different libraries/compilers
  – New architectures
Comparative Debugging

- Specify conditions for correct behavior prior to execution
- Debugger:
  - keeps track of breakpoints
  - performs comparison automatically
- Control returned to user:
  - examination of state
  - continuation of execution

assert P1::big[100..199]"file.c":240 = P2::small"prog.f":300
Why this works?

- Iterative refinement of problem area
Why this works?

- Iterative refinement of problem area
Why this works?

• Iterative refinement of problem area
Why this works?

• Iterative refinement of problem area
VISUALIZATION
Reporting Differences

Values of scalars, small arrays

Starting execution of processes
Comparing c and c.
Maximum difference between values: 1.15442e-23
Total difference between values: 4.37116e-23
Number of differences detected = 823
First 10 errors are:
At Index: (30) = (Diff, Value 1, Value2) 0.000488
At Index: (32) (32) = (Diff, Value 1, Value2) 0.
The power of visualization
The power of visualization

- Difference in physics of planetary boundary layer
  - Computation of steps suited to parallel execution
  - Evident in 3 dimensional visualisation
- Error in radiation time step computation
- More complete physics in long wave radiation
The power of visualization
SCALABILITY
Network Infrastructure

Ref: [Image]

- Reference Signature Array
- Suspect Signature Array

- Debug client

- Multi-signatures

- Debug Server
- Multi-signatures

- Sub-structure
  - Multi-block Segmentation
  - Hash
  - Multi-signatures

- Network Infrastructure

- Multi-signatures

- Sub-structure
  - Multi-block Segmentation
  - Hash
  - Multi-signatures
Point to Point protocol

Head node

Interconnection Network

Compute Node
Compute Node
Compute Node
Compute Node
Compute Node
Compute Node
Compute Node
Compute Node

Debug Server
Debug Server
Debug Server
Debug Server

Application
Application
Application
Application

Data
Data
Data
Data

?
HETEROGENEITY
Architecture Independent Format

- Ability to represent data from different architectures in an architecture neutral way
- Need to perform numerical operations on data in this format
- Need to be able to convert to/from native formats

```c
struct {
    int a;
    float b[3];
};
```

```plaintext
{a:is4,b:[r0..2is4]f4}

<table>
<thead>
<tr>
<th>byte 1</th>
<th>byte 2</th>
<th>byte 3</th>
<th>byte 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>exponent</td>
<td>mantissa</td>
<td></td>
</tr>
<tr>
<td>s</td>
<td>exponent</td>
<td>mantissa</td>
<td></td>
</tr>
<tr>
<td>s</td>
<td>exponent</td>
<td>mantissa</td>
<td></td>
</tr>
</tbody>
</table>
```
Flexibility in Comparisons

• Tolerances used for inexact equality
• Data structures should be:
  – type conformant (with conversion)
  – same size, but can be differing shapes
• Arrays
  – Differences allowed are:
    • offset ranges in arrays
    • ordering of indexes
    • Number of indexes
    • Language
• Dynamic data
  – Linked lists
  – Objects
Programming Languages other than C/F

• OpenACC/OpenMP
  – Sequential regions executed on CPU
  – Parallel regions offloaded to GPU
  – Data dynamically moves between CPU and GPU
  – Separated address spaces for CPU and GPU codes
  – Inconsistent precision of floating numbers across CPU and GPU

• UPC: a virtual global memory space
  – Automatically decomposing the global data across a number of SPMD threads
  – Exchanging data between threads is managed by the UPC runtime system
WITHOUT A REFERENCE CODE?
Statistical Assertions

• Asserting descriptive statistics of a given dataset
  – Mean, standard deviation …
• Asserting statistical hypotheses
  – Distribution functions
  – Statistical tests
• Adjacent time steps show high data correlation
  – Can help identifying potential errors and outliers
• Asserting program states across time steps

```plaintext
history etot $a::dvalue@“thermo.cpp”:1521 10 100
set reduce stdev; compare etot < 0.1
```
Statistical Assertions

- **Statistical parameters** (mean, SD, etc)
- **Statistical tests** ($T$, $\chi^2$, etc)
- **Distributions**

Many high speed particles

Speed histogram for incorrect code  Speed histogram for correct code
STATUS AND IMPLEMENTATION
CCDB on Cray supercomputers

• Supporting Cray XE, XK, and XC supercomputers
• CCDB client: a comparative debugging interface
  – Launching parallel applications onto the back-end
  – Controlling the execution of the programs remotely
  – Compare key data structures between different applications
• CCDB server: a pluggable architecture
  – GDB: C, Fortran, and UPC programs
  – CUDA-GDB: OpenACC, OpenMP
  – MRNet
  – Scalable communication between the CCDB client and servers
  – AIF (Architecture Independent Format)
  – ‘Normalizing’ the data across platforms and languages
TO INFINITY AND BEYOND?
Exascale

- Probably big!
- Heterogeneous
- Mixed precision
- Hierarchical memories
- Algorithms
  - Loose synchronization
  - Fault tolerant
Debugging and Correctness

Scaling Debugging Techniques
Debugging Hybrid and Heterogeneous Architectures
Specialized Memory Systems
Domain Specific Languages
Mixed Precision Arithmetic
Adaptive Systems
Correctness Tools
Debugging and Correctness

Scaling Debugging Techniques ✔
Debugging Hybrid and Heterogeneous Architectures ✔
Specialized Memory Systems ✔
Domain Specific Languages ✔
Mixed Precision Arithmetic ✔
Adaptive Systems ?
Correctness Tools ?

✔ means some progress
Onto Dessert ...