allinea

Leaders in parallel software development tools

Debugging for the hybrid-multicore age (A HPC Perspective)

David Lecomber

CTO, Allinea Software

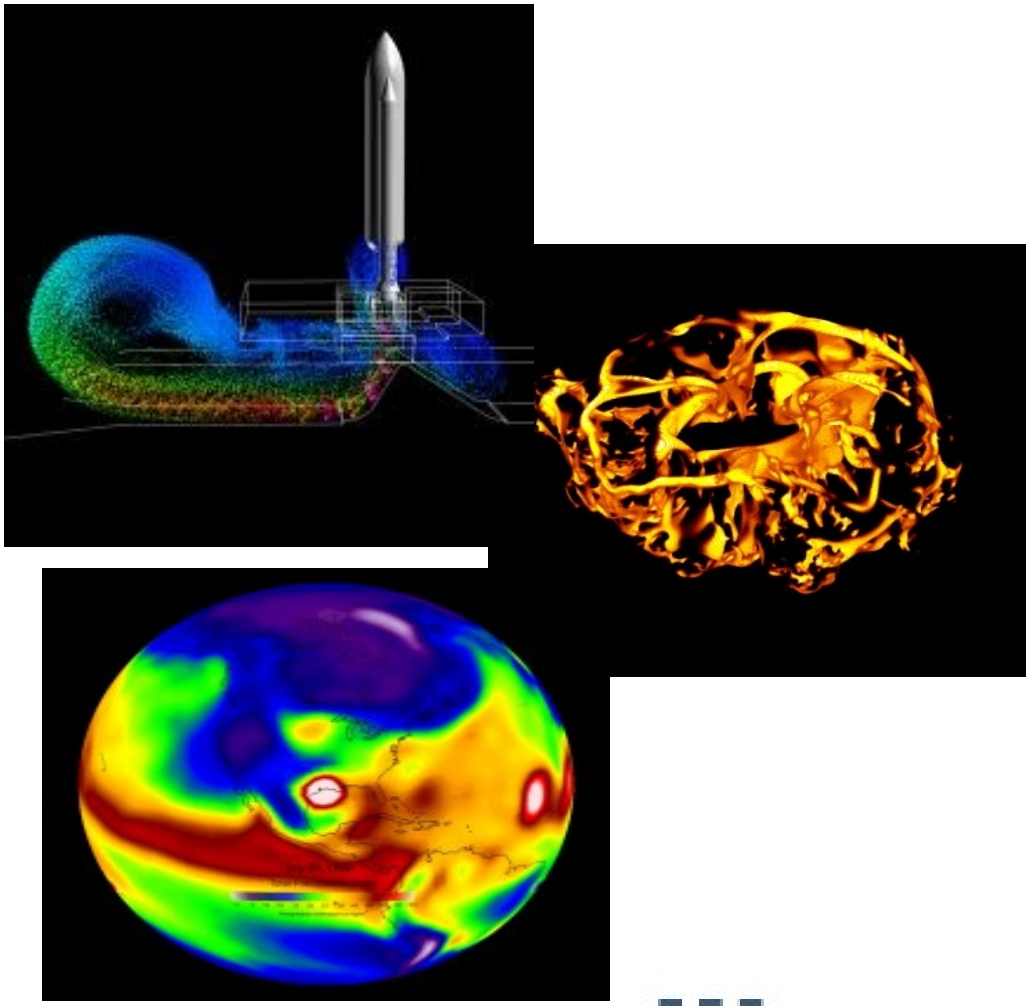
david@allinea.com

Agenda

- What is HPC?
- How is scale affecting HPC?
- Achieving tool scalability
- Scale in practice
- The future

What is HPC?

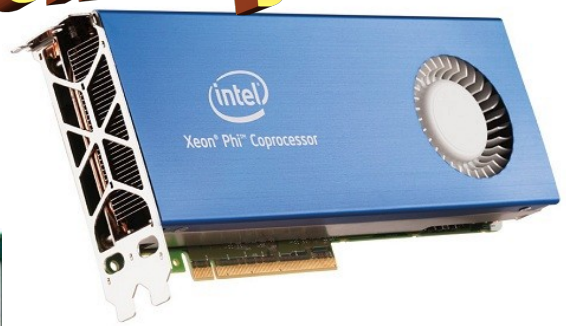
- High Performance Computing
  - Simulation of some natural process/thing
  - Intense number crunching – CPUs worked flat out
  - Very large number of usually interrelated calculations – too big for single machine (data or time)
  - Rarely real time
  - Distinct from data crunching
- Who uses HPC?
  - Engineering
  - Weather and climate
  - Physical sciences
  - …
  - ...



Challenges for HPC developers

Scale

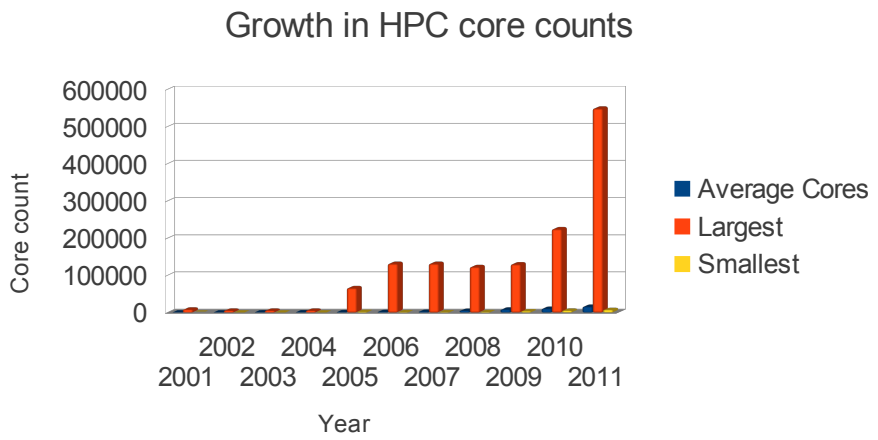
Heterogeneity

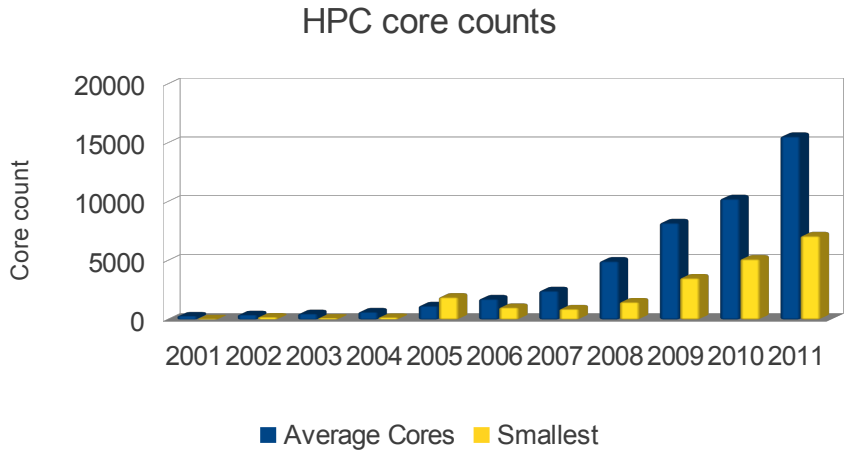


Diversity

allinea
www.allinea.com

Vast concurrency is already here





- Machine sizes are exploding
  - Skewed by largest machines

    … but a common trend
  - Largest (Jun 2012) 1.5M cores
- Petaflop club is growing
  - European systems:

    SuperMUC, Fermi, JuQUEEN and Bull-X Curie
- Software is critical to success
  - 10,000, 100,000, 1,000,000 cores..
  - GPU and Intel Xeon Phi
- Big **and** heterogeneous

Bugs exist at scale...

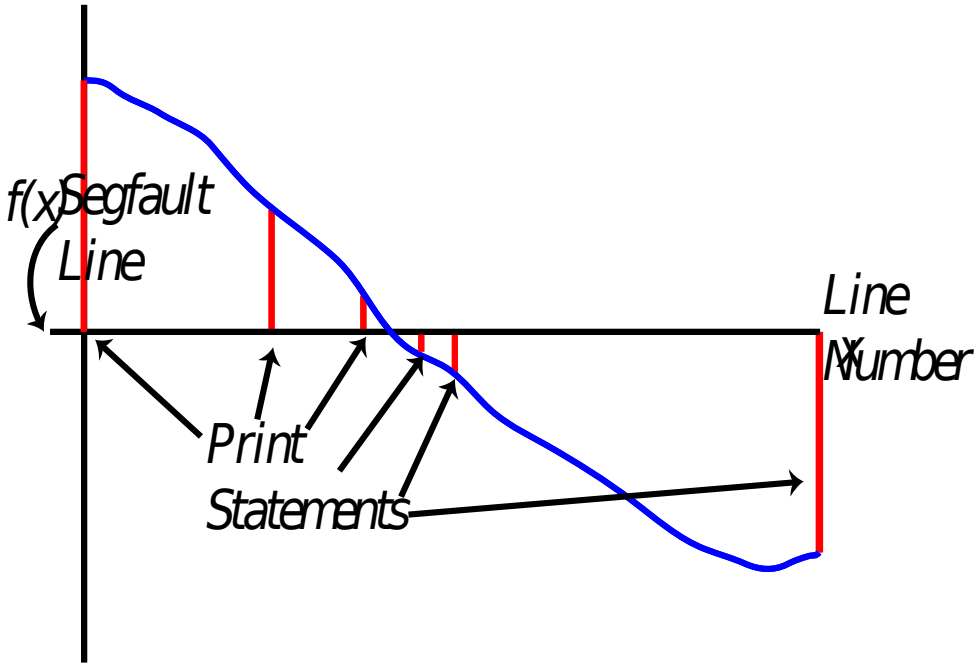
- Long term study at LANL
  - Up to 24% of job failures due to software errors
- Anecdotal examples
  - MPI library – overflow in MPI_Alltoallv due to 32 bit integer overflow in length **- [deterministic]**
  - MPI library – incorrect behavior for process ID wrap in startup **[random]**
  - A debugger that fails to kill child process if PID < parent PID **[random]**
  - User code – segmentation fault in a random process at 98,304 cores **[random]**
  - Recent industrial example – a 10,000 core segfault **[deterministic]**

Bug fixing as scale increases

- Can we reproduce at a smaller scale?
  - Attempt to make problem happen on fewer nodes
    - Often requires reduced data set – the large one may not fit
    - Smaller data set may not trigger the problem
  - Run at scale and debug a subset?
    - Bugs that trigger on a random process defeat this strategy!
  - Does the bug even exist on smaller problems?
    - Didn't you already try the code at small scale?
    - Is it a system issue – eg. an MPI problem?
  - Is probability stacking up against you?
    - Unlikely to spot on smaller runs – without many many runs
    - But near guaranteed to see it on a many-thousand core run
- Debugging at extreme scale is a necessity

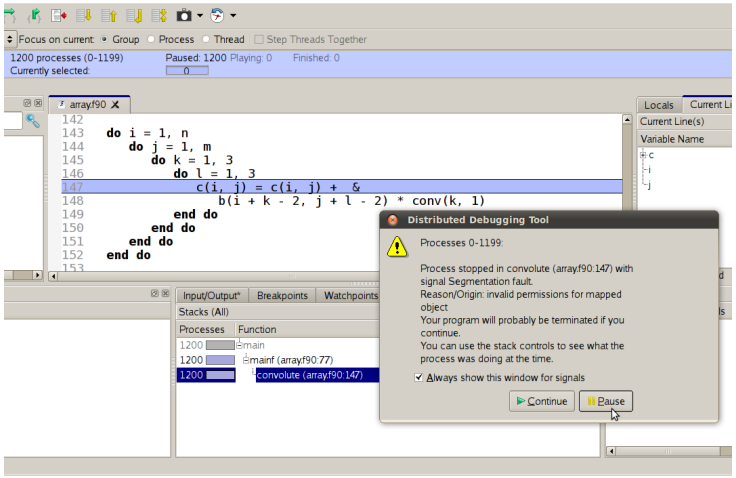
Why debuggers were invented

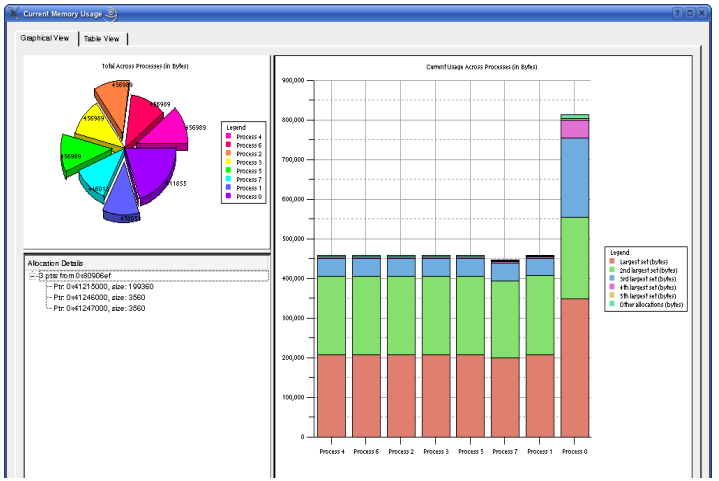
- The first debugger: print statements
  - Each process prints a message or value at defined locations
  - Diagnose the problem from evidence and intuition
- A long slow process
  - Analogous to bisection root finding
- Broken at modest scale
  - Too much output – too many log files



Allinea DDT in a nutshell

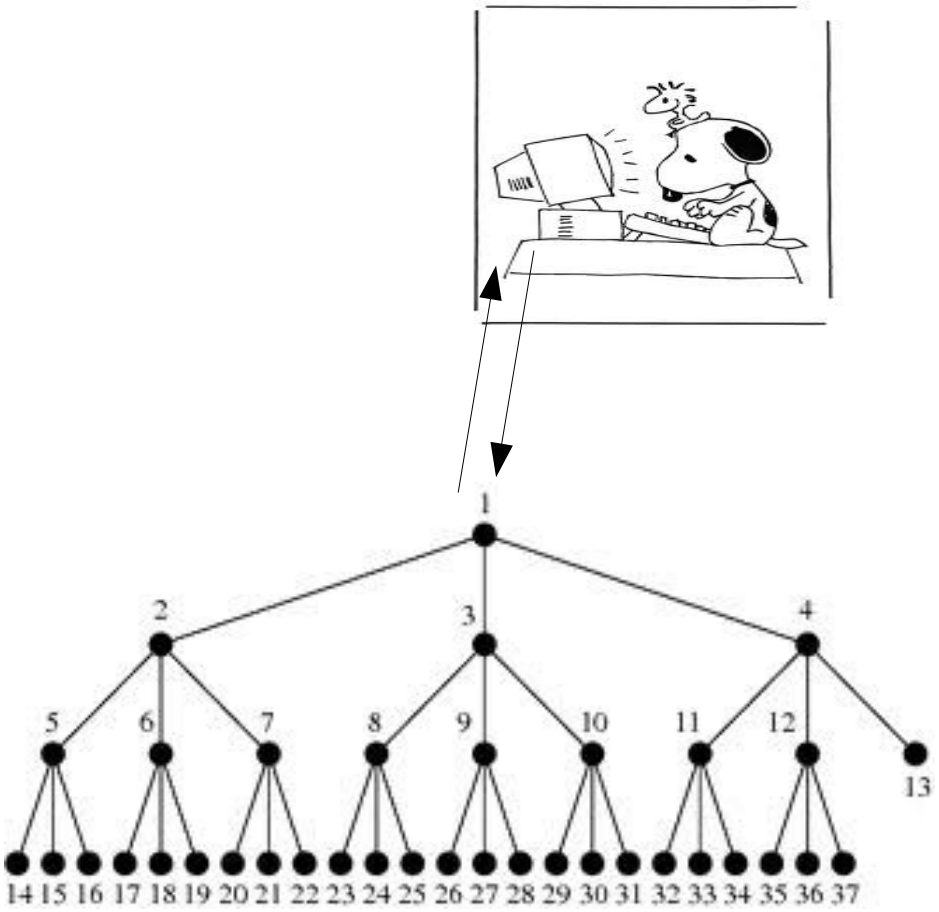
- Graphical source level debugger for
  - Parallel, multi-threaded, scalar or hybrid code
  - C, C++, F90, Co-Array Fortran, UPC
- Strong feature set
  - Memory debugging
  - Data analysis
- Managing concurrency
  - Emphasizing differences
  - Collective control





How to make a Petascale debugger

- A control tree is the solution
  - Ability to send bulk commands and merge responses
  - Compact data type to represent sets of processes
  - Develop aggregations
    - Not everything can merge losslessly
    - Maintain the essence of the information
      - eg. min, max, distribution



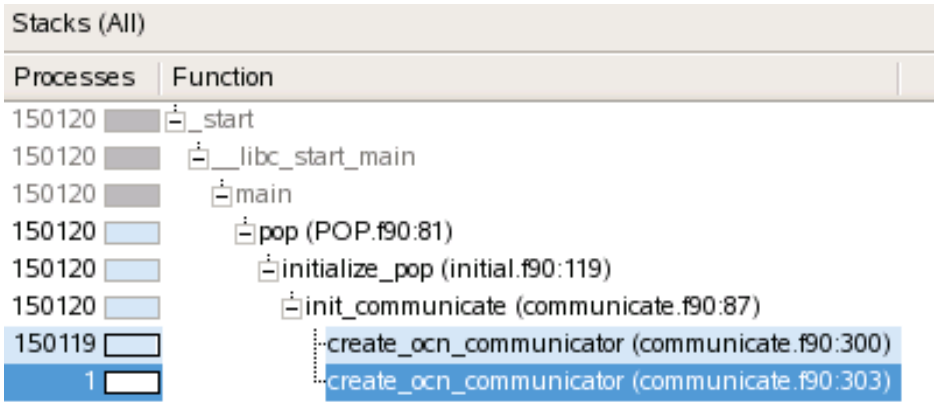
Fixing the everyday crash

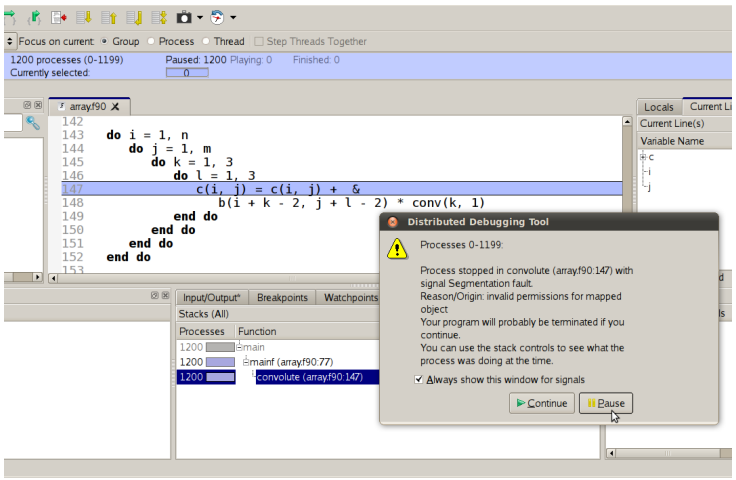
- The typical application crash or early exit:
  - Run your program in the debugger

    ddt {application} {parameters}
  - Application crashes or starts to exit

- **Where** did it happen?
  - Allinea DDT merges stacks from processes and threads into a tree
  - Leaps to source automatically

- **Why** did it happen?
  - Some faults evident instantly from location(s)
  - But for others we need to look further – at variables

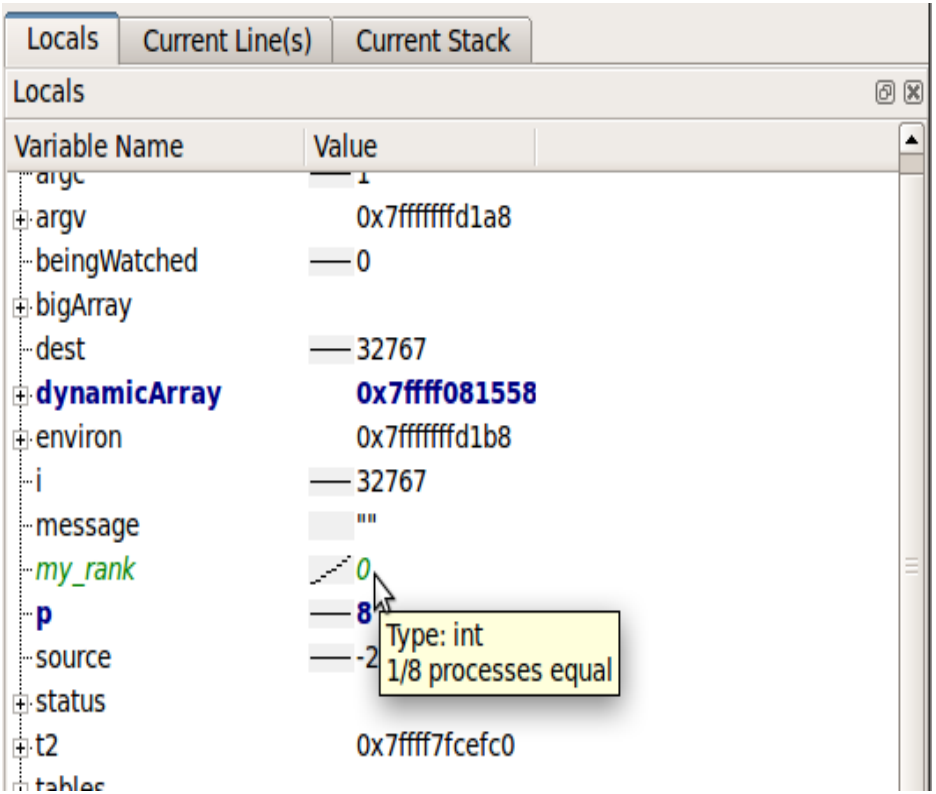




allinea
www.allinea.com

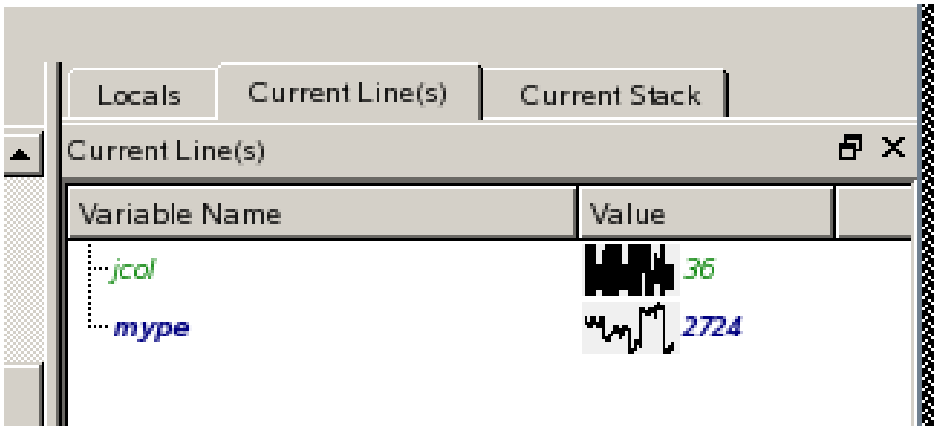
Simplifying data divergence

- Need to understand the data
  - Too many variables to trawl manually
  - Allinea DDT compares data automatically
- Smart highlighting
  - Subtle hints for differences and changes
  - With sparklines!
- More detailed analysis
  - Full cross process comparison
  - Historical values via tracepoints

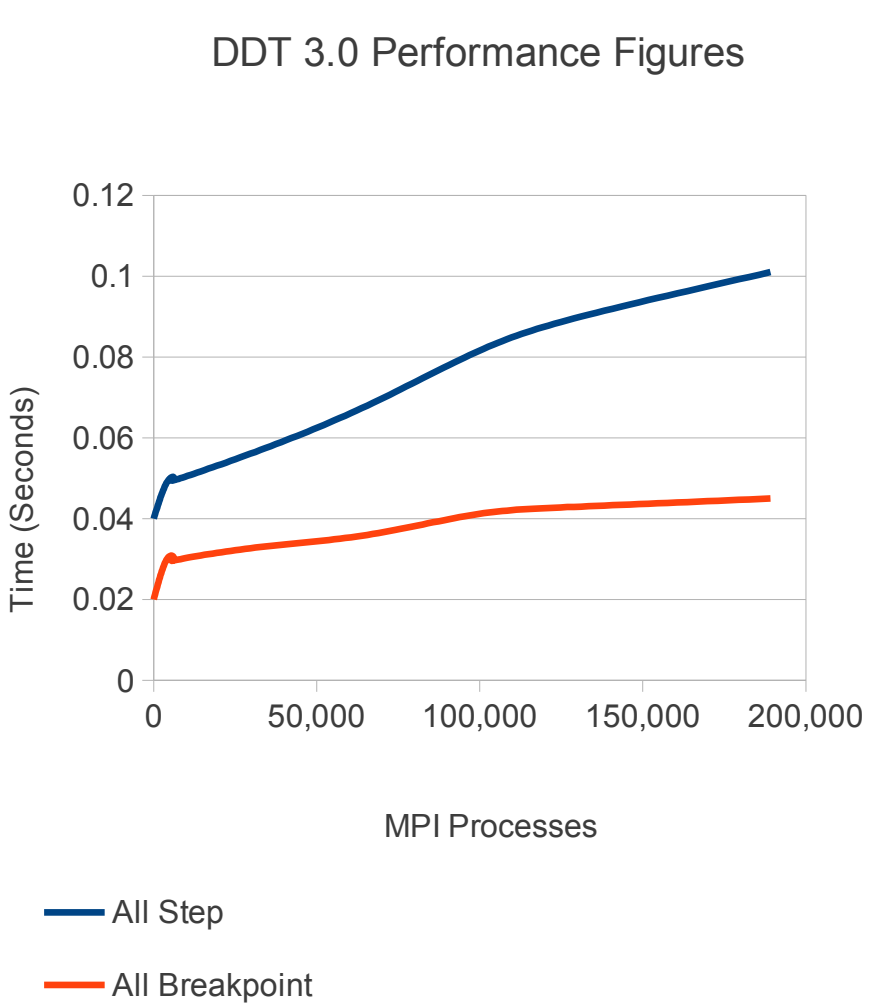


Key features at scale

- Top 5 features at scale
  - Parallel stack view
    - Ideal for divergence or deadlock
  - Automated data comparison: sparklines
    - Rogue data is easily seen
  - Parallel Array searching and visualization
    - Data is too large to examine manually
  - Process control with step, play, and breakpoints
    - Still essential
  - Offline debugging
    - Access to machine may be hard – try offline debugging instead



Allinea DDT - Petascale and beyond

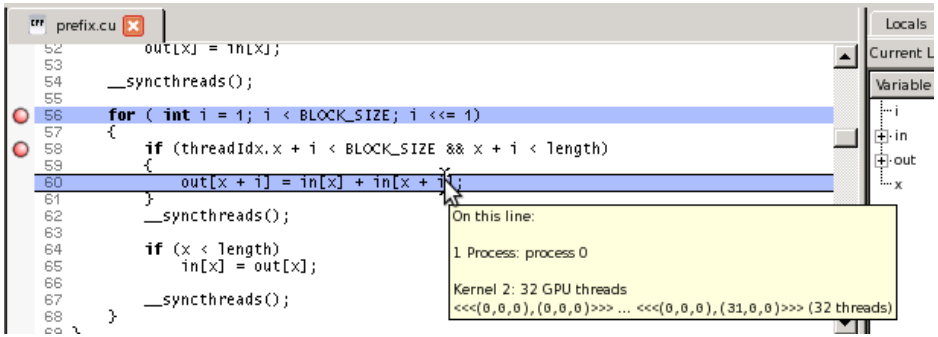


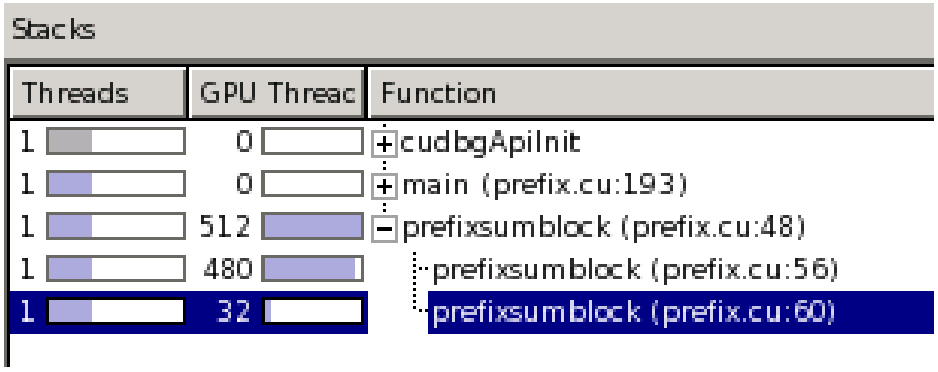
- Scale doesn't have to be **slow**
  - High performance debugging - even at 200,000 cores
  - Step all and display stacks: 0.1 seconds
  - Logarithmic performance
- Stable and in production use
  - Routinely used by end users at over 100,000 cores
- Scale doesn't have to be **hard**
  - 100,000 cores should be as easy as 100 cores
  - The user interface is vital to success

allinea
www.allinea.com

Heterogeneous debugging - GPUs

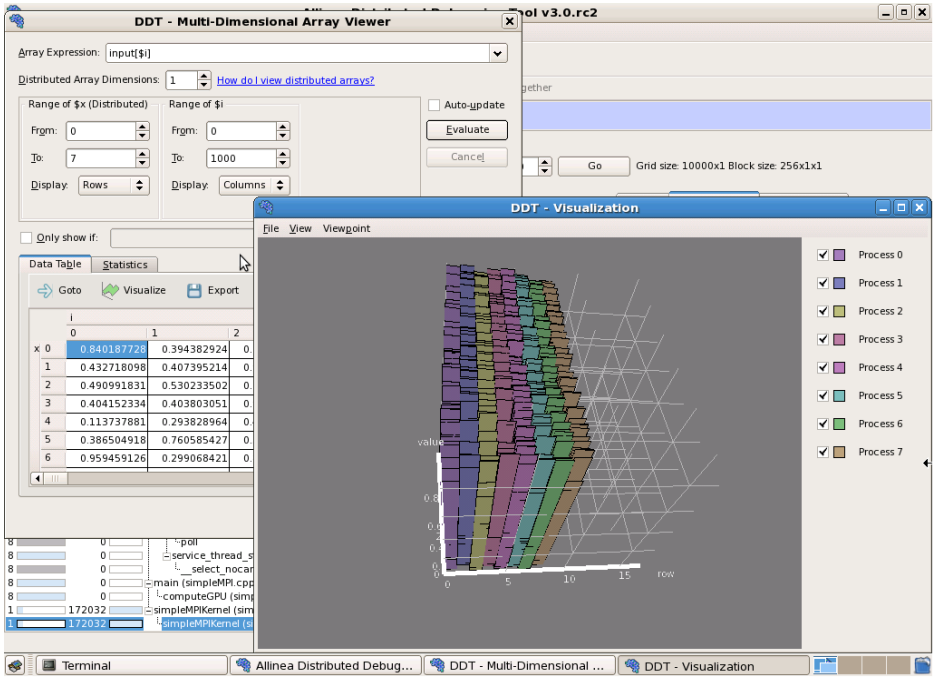
- As easy as debugging a CPU
  - Run to a crash
  - Step and observe
  - Simultaneously debugs CPU code
- CPU-style debugging features
  - Double click to set breakpoints
  - Hover the mouse for information
  - Step a warp, block or kernel
- Simplifies complexity of GPU threads





Examining GPU data

- Read variables and data easily
  - All memory classes
    - Shared, constant, local, global, register..
    - CPU or GPU
  - Plot larger arrays directly
    - Device or host memory
    - Filtering and export to file
- Support for memory error detection
  - CUDA Memcheck



Challenges for developers

Scale

Heterogeneity

Diversity

allinea
www.allinea.com