allinea

SCALE TO NEW HEIGHTS

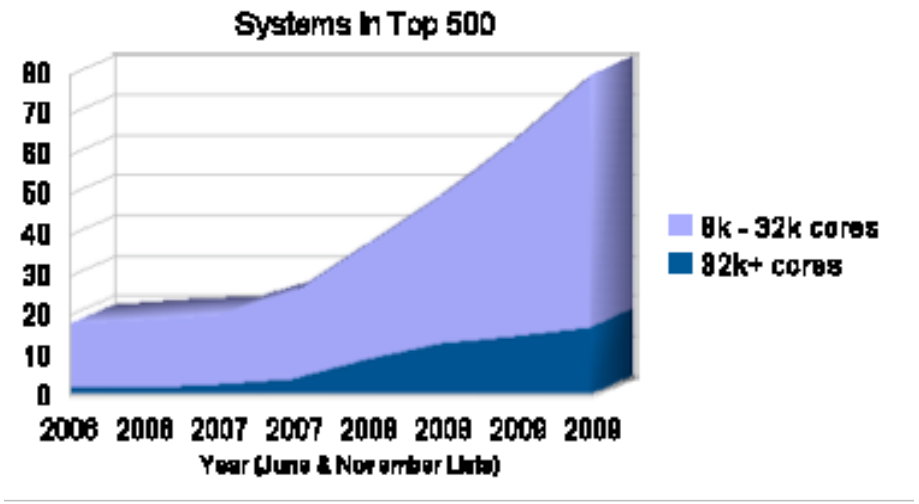
Debugging Applications in Jaguar XT5

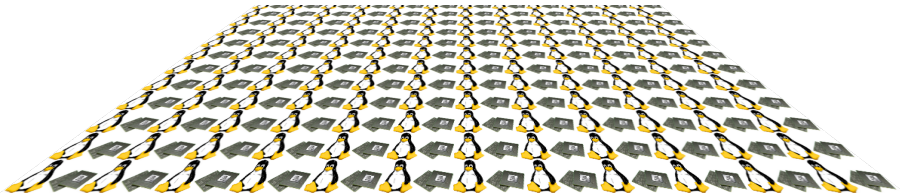
Presented by:
Oscar Hernandez, Rich Graham
ORNL

David Lecomber, David Maples
Allinea

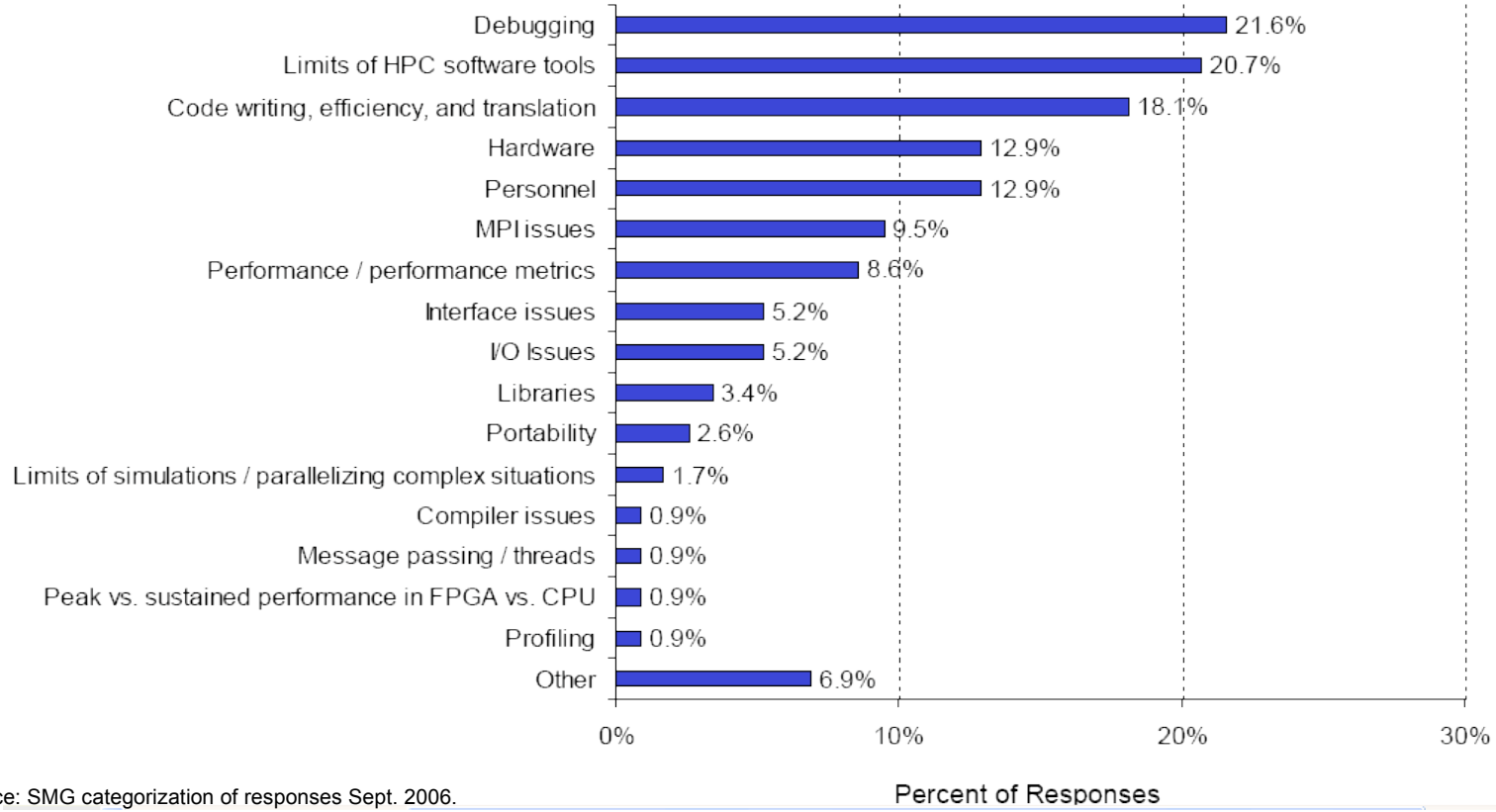
Interesting Times ...

- Processor counts growing rapidly
- GPUs entering HPC
- Large hybrid systems imminent
- But what happens when software doesn't work?





HPC Bottlenecks



Source: SMG categorization of responses Sept. 2006.

- Shortening application development time is vital
- Debugging is now the major issue !

Allinea Software

- HPC tools company since 2001
  - DDT - Debugger for MPI, threaded/OpenMP and scalar
  - OPT - Optimizing and profiling tool for MPI and non-MPI
  - DDTLite – Parallel Debugging Plugin for Microsoft Visual Studio 2008 SP1 and above
- Large European and US customer base
  - Ease of use – means tools get used
  - Users debugging regularly at all scales
  - Scalable interface – easy to use at 1 or 100,000s of cores
- Looking to the future
  - In use at Petascale, collaboration with ORNL
  - Building debuggers that scale on Jaguar XT5
  - GPU product in Beta

Why Debug with Tools?

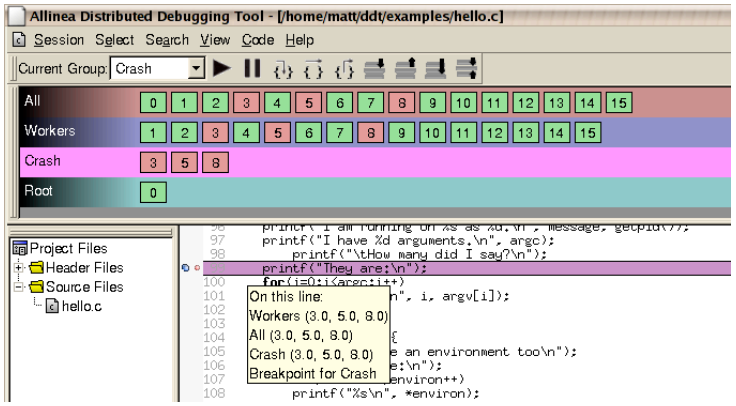
- We need good debuggers
  - Instrument the code: printf - is not a time-effective solution
  - Command line debugging: doesn't scale
  - Complexity scales..
    - User overload: beware the deluge of information
    - Good tools can present information better
    - Aggregation of control across cores
  - Good software tools lead to
    - Faster development
    - More effective maintenance/deployment

DDT

- A powerful and highly intuitive tool
  - Traditional focus has been HPC
- Cross-platform support
  - Linux, Solaris, AIX, Super UX, Blue Gene O/S
  - Blue Gene, Cell, x86-64, ia64, PowerPC, Sparc, NEC, NVIDIA
  - GNU, Absoft, IBM, Intel, PGI, Pathscale, Sun compilers
- Across all MPI and OpenMP implementations
  - From low end to high end
- Support for all scheduling systems
  - SGE, PBS, LSF, MOAB, ...
  - Flexible, powerful, easy to use queue submission

DDT Basic Principles

- A sophisticated GUI aids development
  - Helps the user to control parallel execution
  - Helps to find and focus on potential problems
  - User controls actions.....
    - Set breakpoints, watchpoints, lock step, align stacks etc
    - Can focus in on individual threads/processes
    - Can quickly and easily create groups of processes
      - Controlled in the same way as individual
      - See stack across all processes
    - All based on code logic !



How to use it?

Loading DDT in Jaguar XT5

%module load ddt

Compiling your application with debugging option

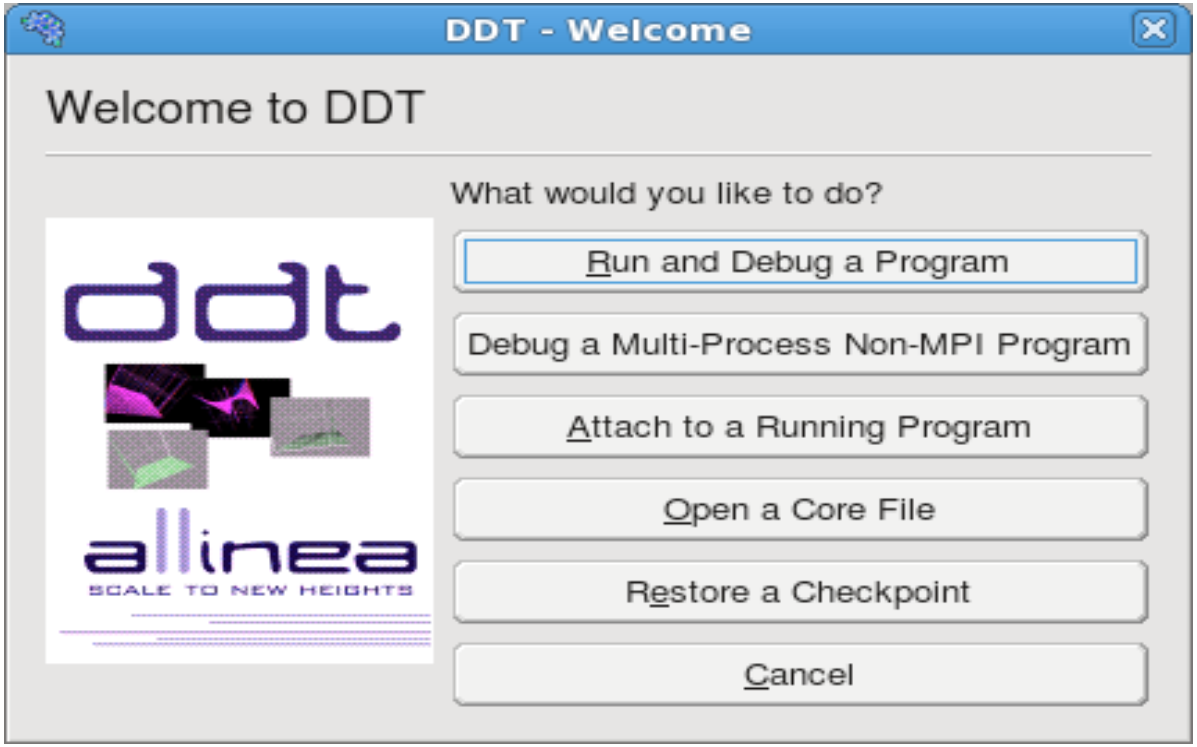
%cc –g myapplication.c –o myapplication

Invoking DDT

%ddt myapplication

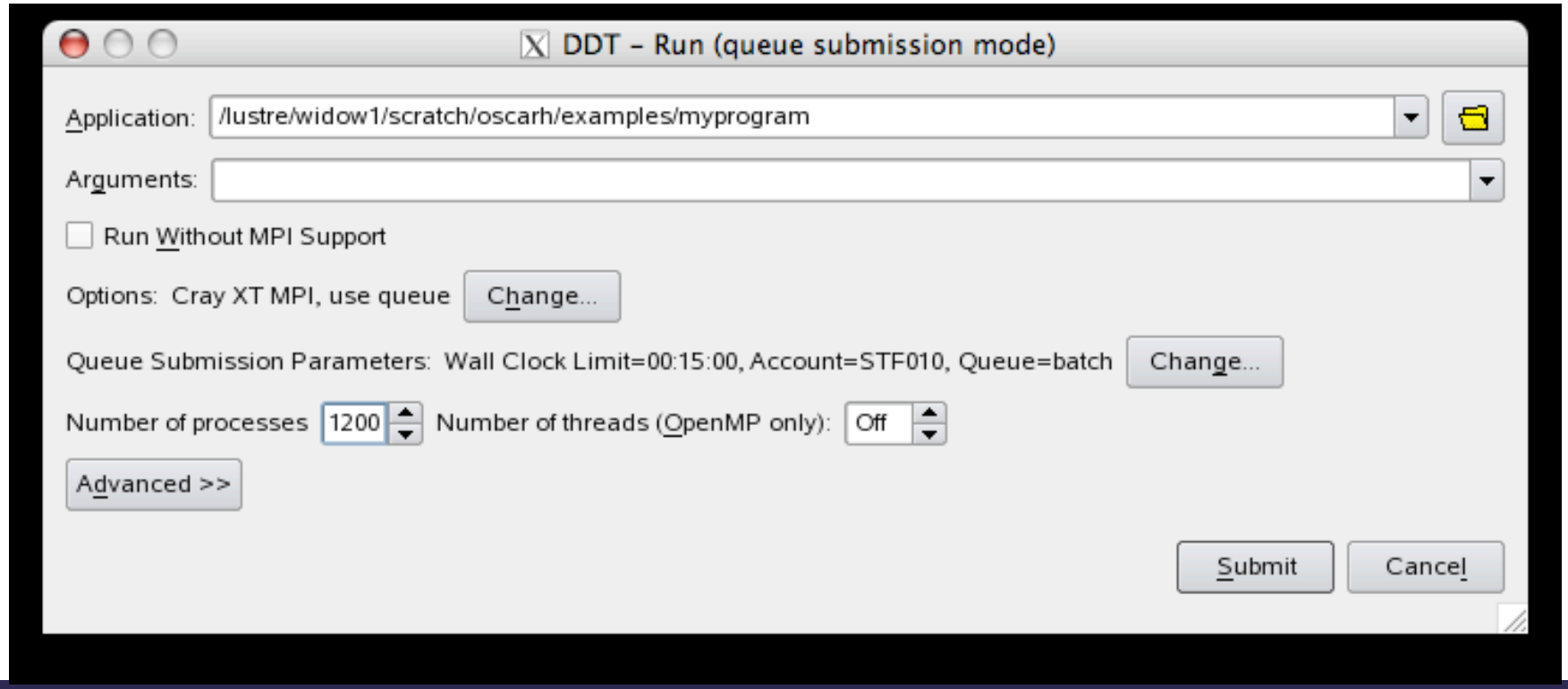
DDT - Start Debugging

- Ability to run in numerous modes



Queue Submission

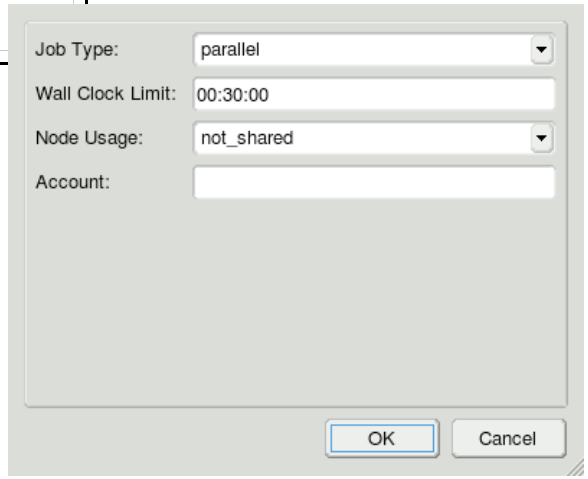
- DDT has a GUI to submit jobs
- Can customize submission scripts with GUI
- You can specify MPI versions, input data information, advanced submission options.



DDT – Scheduler Launch

- Enhanced Queuing Submission
  - Simple mechanism to customise options
  - Submit non-mpi jobs to a queue
  - Set pre-defined variables in submission script
    - System administrator configures

```
# Name: LoadLeveler
# JOB_TYPE_TAG: {type=select,options=parallel|serial,label="Job Type",default=parallel}
# WALL_CLOCK_LIMIT_TAG: {type=text,label="Wall Clock Limit",default="00:30:00",mask="09:09:09"}
# NODE_USAGE_TAG: {type=select,options=not_shared|shared,label="Node Usage",default=not_shared}
# ACCOUNT_TAG: {type=text,label="Account",global}
```

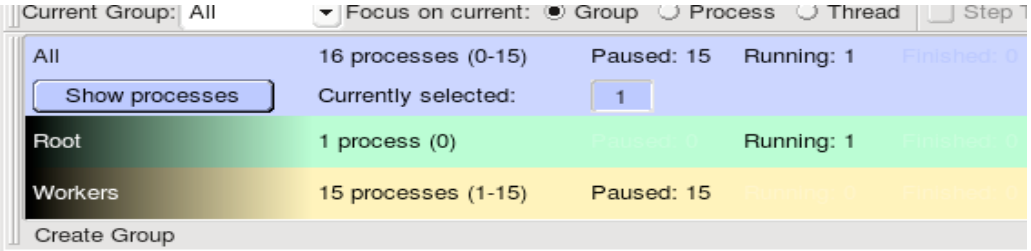


  - Alter the pre-defined variables at run-time
    - User selects own specific values

DDT - Features

- Scalar Features
  - Advanced C++ support including STL, namespaces, virtual functions and templates
  - Advanced Fortran 90, 95 and 2003 support including modules, allocatable data, pointers and derived types

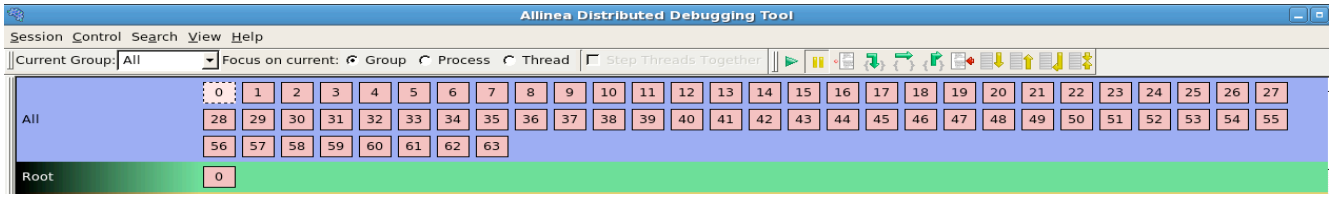
- Multithreading & OpenMP features
  - Perform actions individually or collectively



- MPI features
  - Control processes
  - Individually or by groups

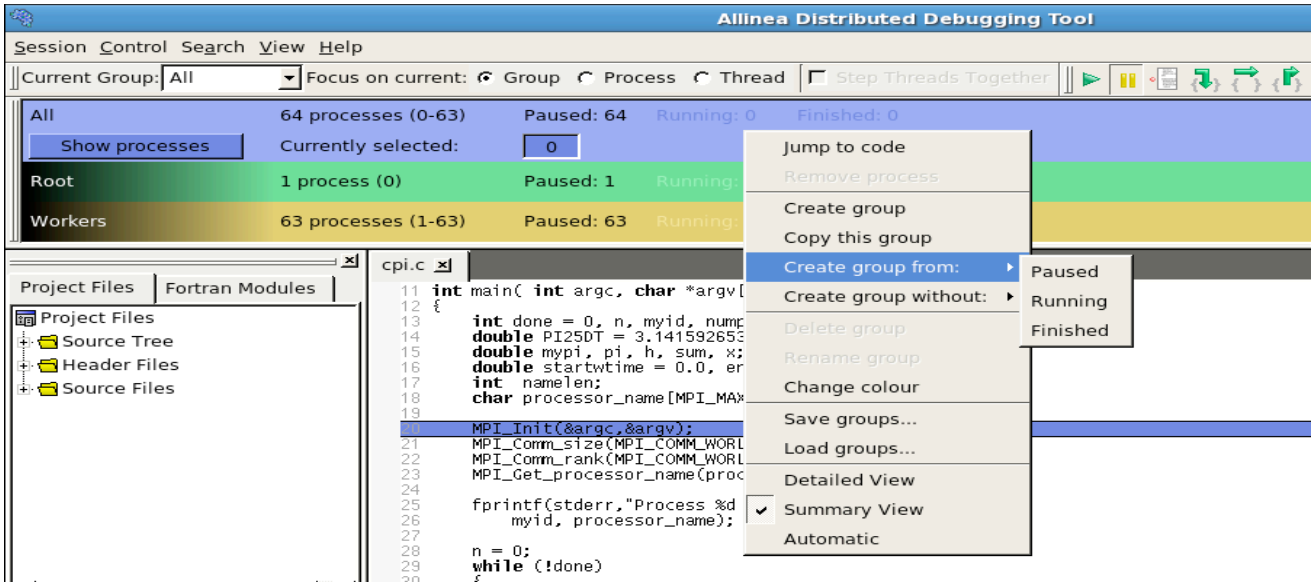
DDT - Scalability

- Process View (Summary or Detail)
  - Chosen automatically or user driven
  - Create new groups based on process “status”



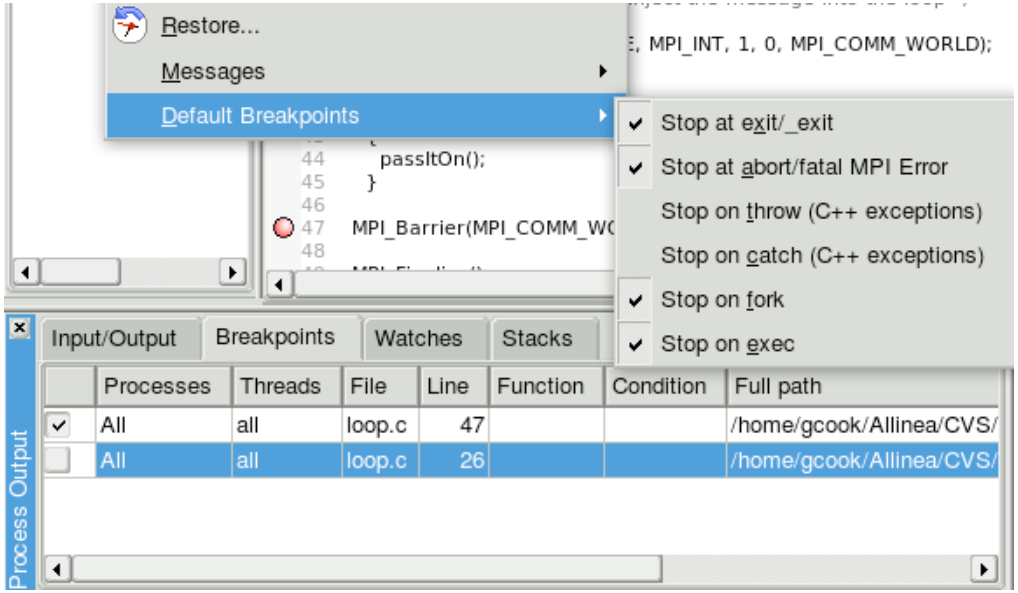
DDT - Scalability

- Process View (Summary or Detail)
  - Chosen automatically or user driven
  - Create new groups based on process “status”
  - Control 1000s of processes in the same way
  - All based on code logic



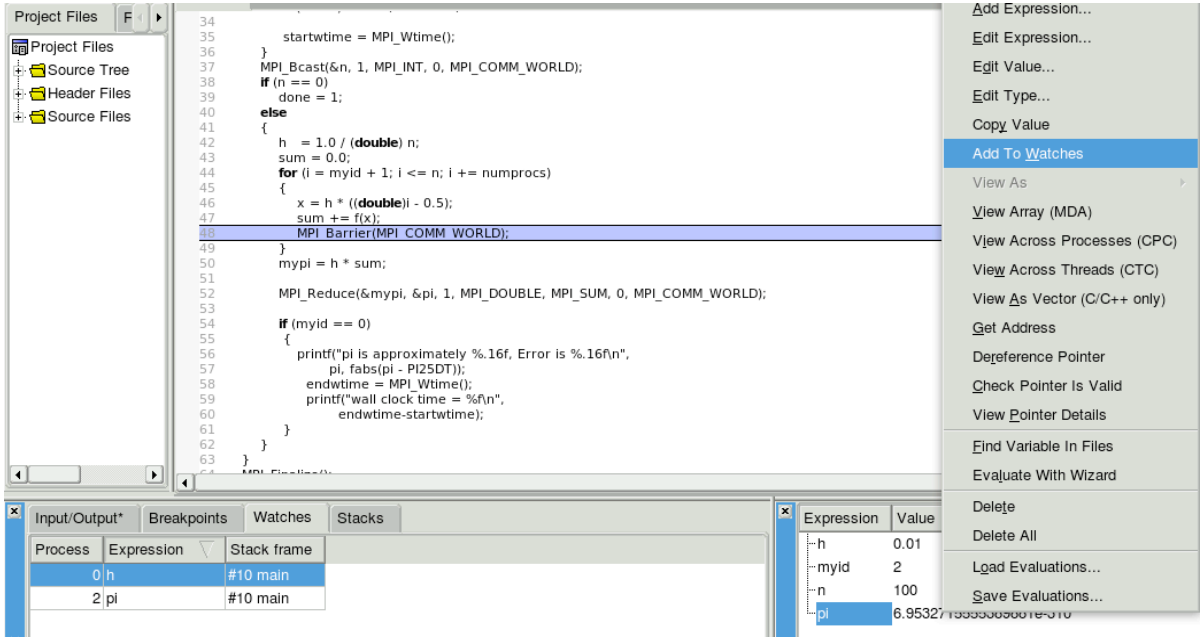
DDT - Breakpoints/Watchpoints

- Quickly and easily control you debug session
  - Breakpoints
    - Default settings (enable/disable)
    - Double click to add them
    - For all processes, groups or a single process
    - Enable/disable from window panel
    - Add a condition



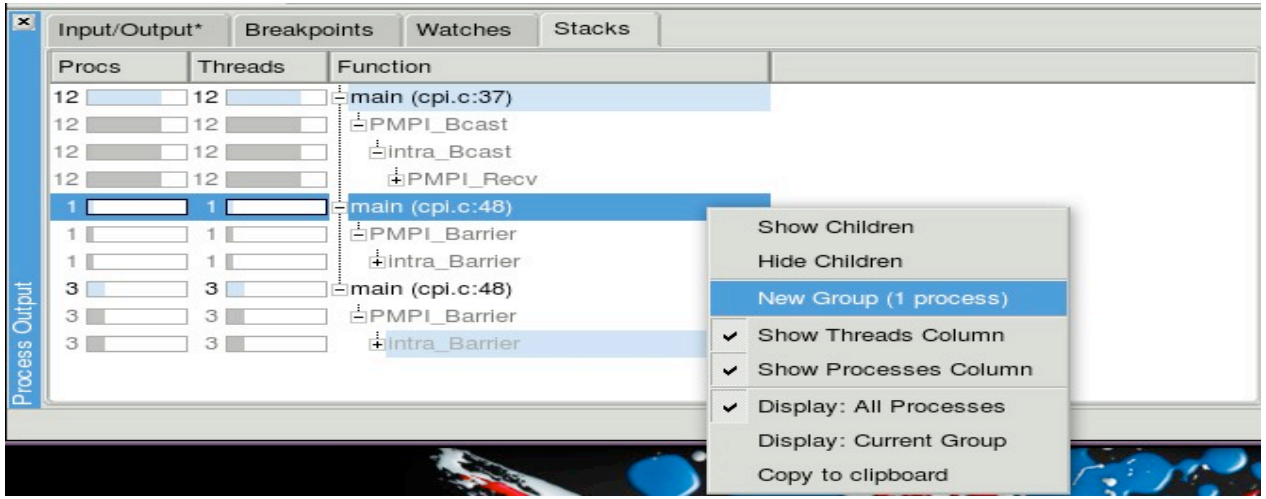
DDT - Breakpoints/Watchpoints

- Quickly and easily control you debug session



  - Watchpoints
    - Select from evaluate, locals or current line
    - Set if variable changes
    - Program halts if change occurs
    - Single process only

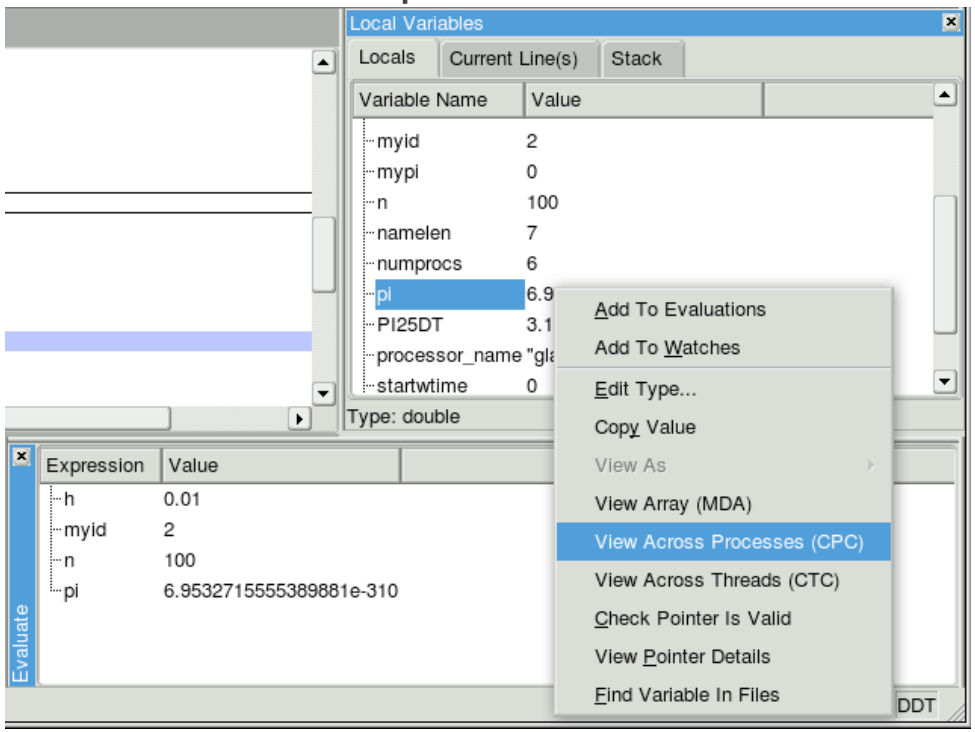
DDT - Scalability



- Parallel Stack View
  - Finds rogue processes faster
  - Identifies classes of process behaviour easily
  - Allows rapid grouping of processes
  - All based on code logic

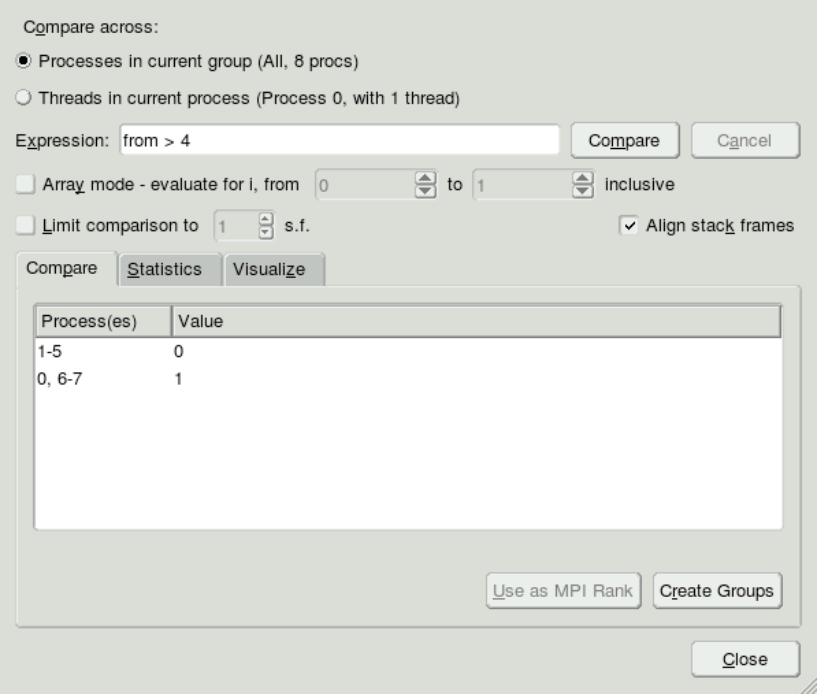
DDT – Data Comparison

- Cross Process/Thread Comparison (CPC)
  - Run from evaluate, locals or current line
  - Variable or expression



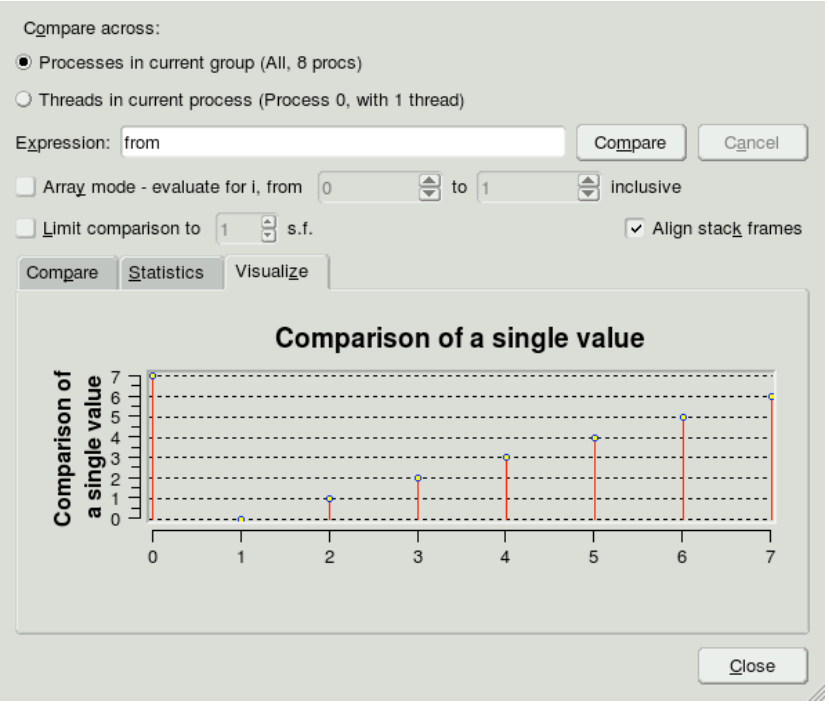
DDT – Data Comparison

- Cross Process/Thread Comparison (CPC)
  - Run from evaluate, locals or current line
  - Variable or expression
  - Create groups from results



DDT – Data Comparison

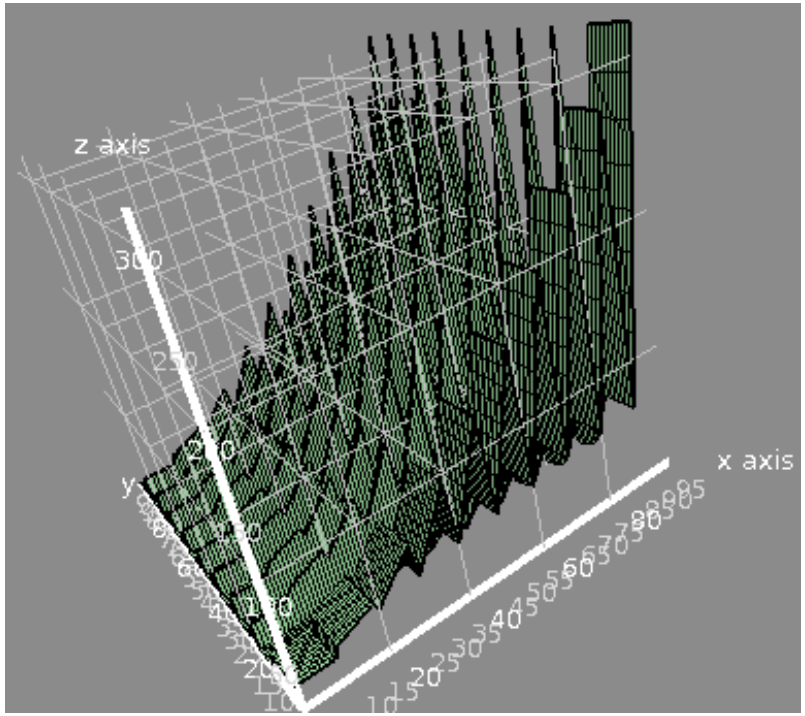
- Cross Process/Thread Comparison (CPC)
  - Run from evaluate, locals or current line
  - Variable or expression
  - Create groups from results
  - Visualize



DDT – Data Comparison

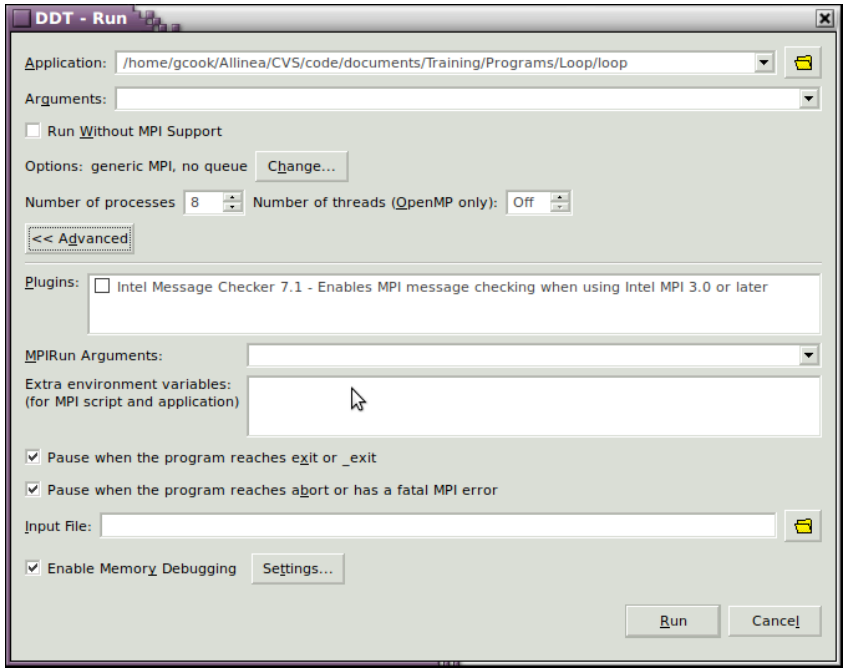
- Cross Process/Thread Comparison (CPC)
  - Run from evaluate, locals or current line
  - Variable or expression
  - Create groups from results
  - Visualize

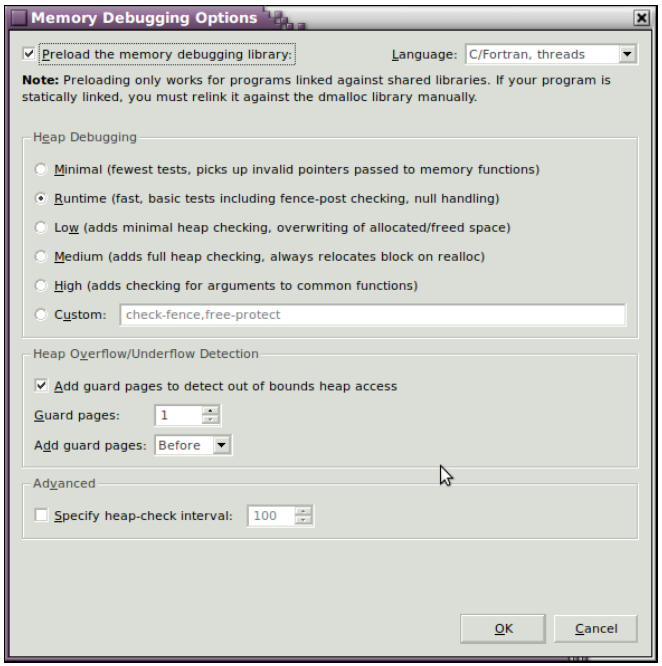
- Multi Dimensional Array viewer
  - 3D OpenGL array viewer
  - Export csv file for post analysis



DDT – Memory Debugging

- **Simple to use**
  - Tick check box and use defaults.....
  - .....or select “Settings”
  - Enhanced heap debugging settings
  - Enable/disable guard pages (before or after)





DDT – Memory Debugging

Check your overall memory usage

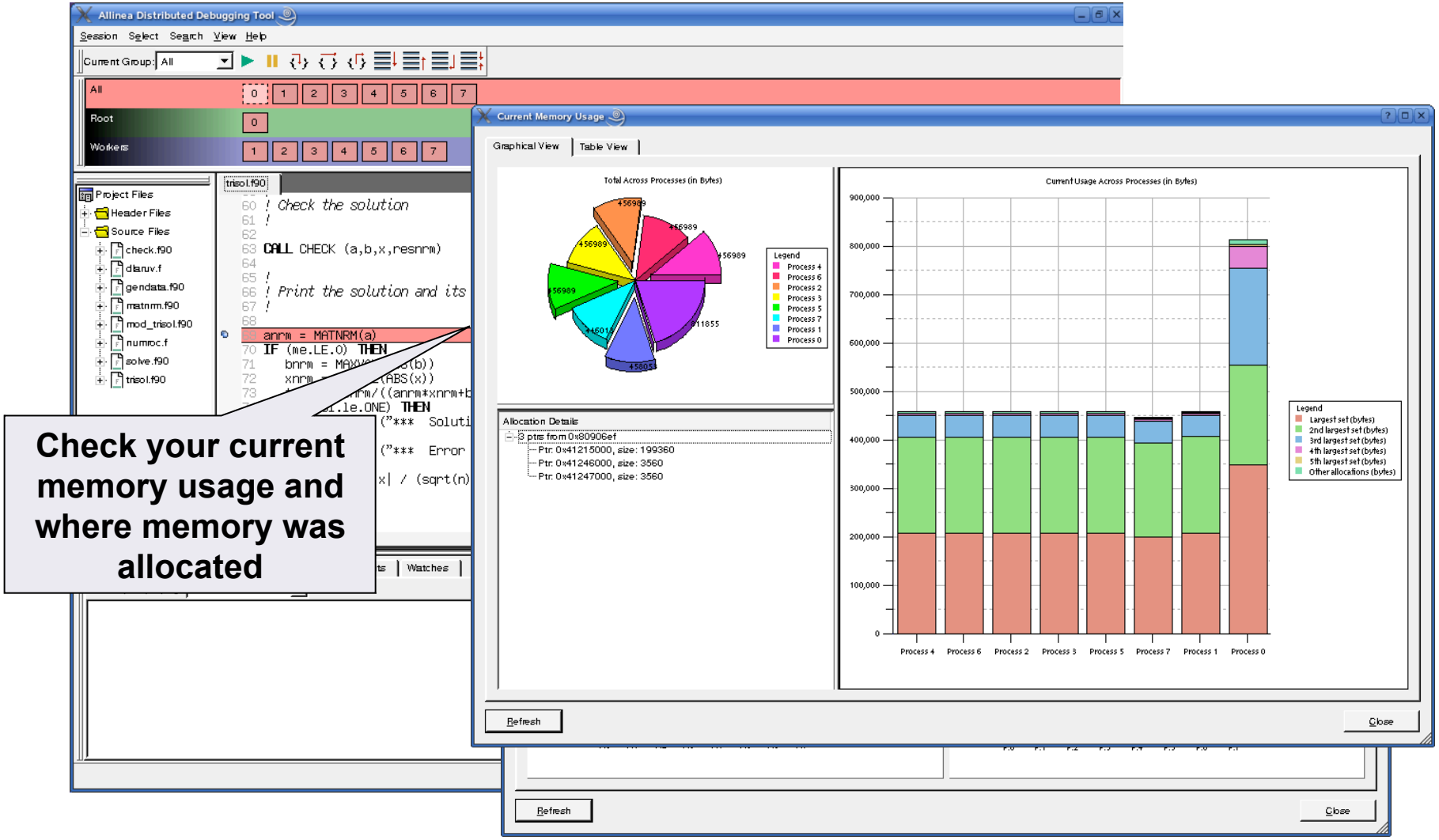
Allinea Distributed Debugging Tool

Overall Memory Stats

Total bytes allocated/freed

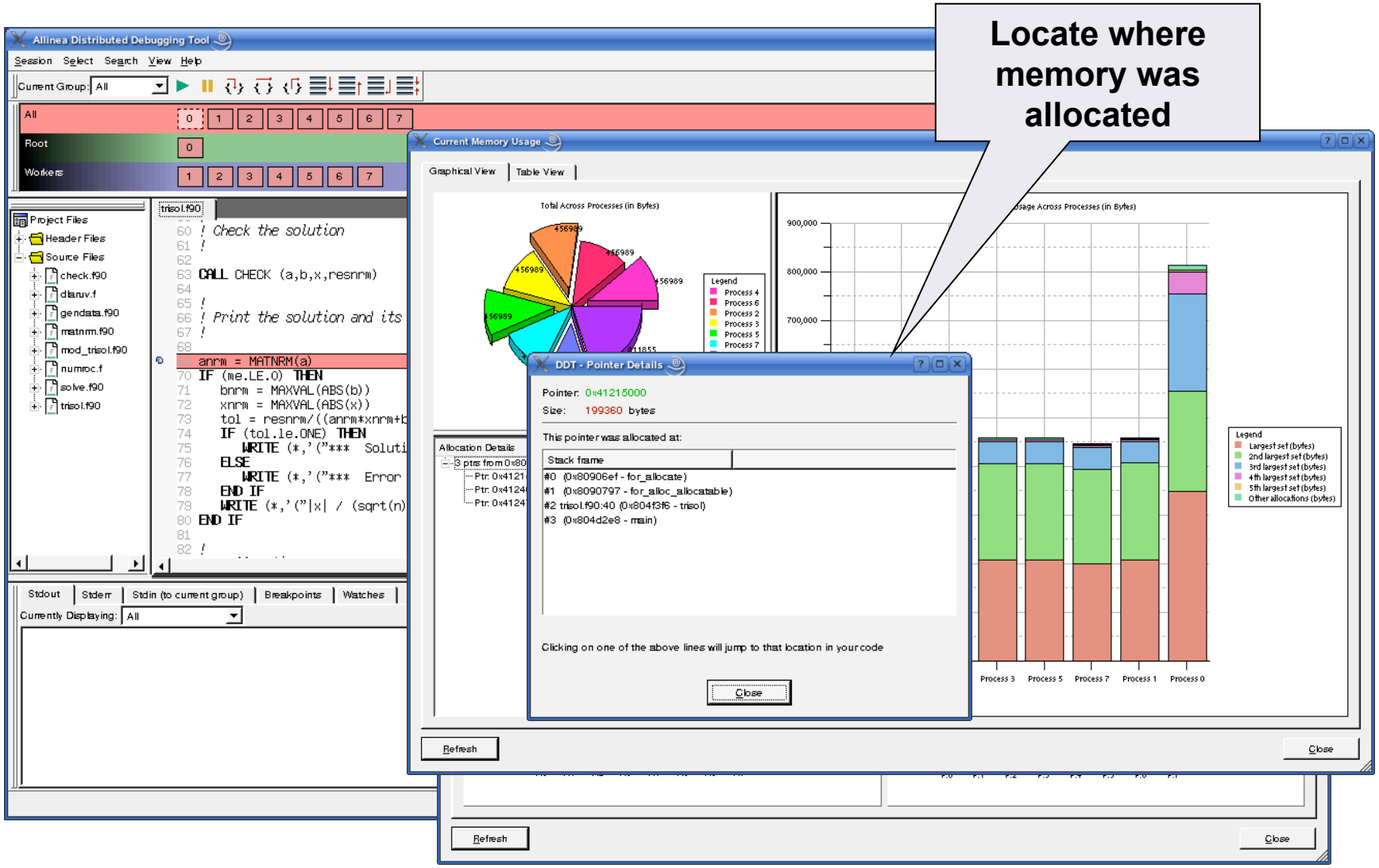
Total number of memory allocation/deallocation calls

DDT – Memory Debugging



Check your current memory usage and where memory was allocated

DDT – Memory Debugging



DDT – Memory Debugging

Allinea Distributed Debugging Tool

Session Select Search View Help

Current Group: All

All 0 1 2 3 4 5 6 7

Root 0

Workers 1 2 3 4 5 6 7

Project Files
- Header Files
- Source Files
  - check.f90
  - dlaruv.f
  - gendata.f90
  - matnrm.f90
  - mod_trisol.f90
  - numroc.f
  - solve.f90
  - trisol.f90

trisol.f90

```
60  ! Check the solution
61  !
62
63  CALL CHECK (a,b,x,resnrm)
64
65  !
66  ! Print the solution and its
67  !
68
69  anrm = MATNRM(a)
70  IF (me.LE.0) THEN
71     bnrm = MAXVAL(ABS(b))
72     xnrm = MAXVAL(ABS(x))
73     tol = resnrm/((anrm*xnrm+b
74     IF (tol.le.ONE) THEN
75        WRITE (*,'("***  Soluti
76     ELSE
77        WRITE (*,'("***  Error
78     END IF
79     WRITE (*,'("|x| / (sqrt(n)
80  END IF
81
82  !
```

Stdout | Stderr | Stdin (to current group) | Watches

Current Memory Usage

Graphical View | Table View

Total Across Processes (in Bytes)

Legend: Process 4, Process 6, Process 2, Process 3, Process 5, Process 7

Current Usage Across Processes (in Bytes)

900,000 800,000 700,000

Process 3 Process 5 Process 7 Process 1 Process 0

Largest set (bytes)
2nd largest set (bytes)
3rd largest set (bytes)
4th largest set (bytes)
5th largest set (bytes)
Other allocations (bytes)

Clicking on one of the above lines will jump to that location in your code

Close

Refresh | Close

Distributed Debugging Tool

Processes 0-7:
Program stopped in vfprintf from /lib/libc.so.6 with signal SIGSEGV.
Reason/Origin: address not mapped to object
Your program will probably be terminated if you continue.
You can use the stack controls to see what the process was doing at the time.

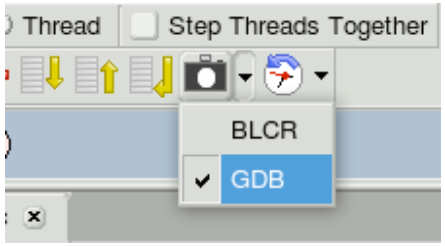
☑ Always show this window for signals

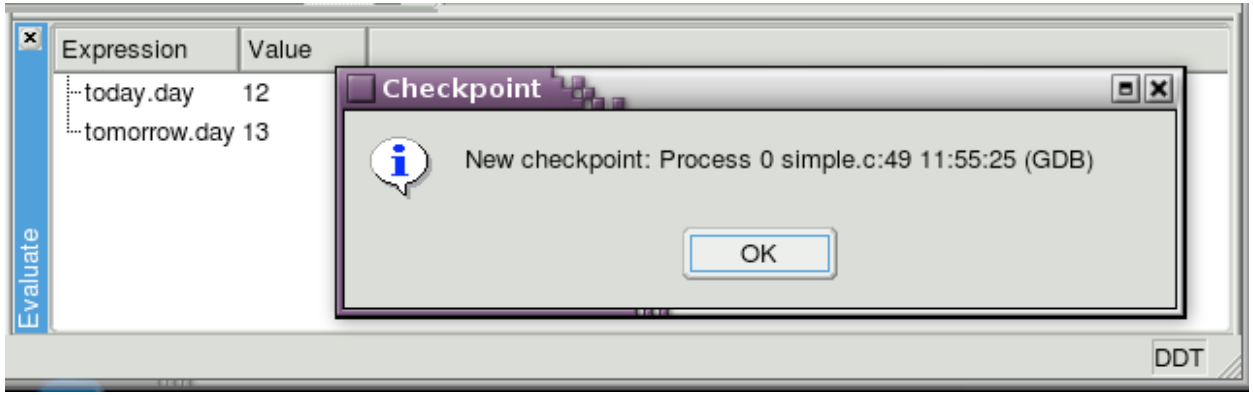
Continue | Pause

Guard Pages
Stop immediately on read/write beyond array end and common errors

DDT - Checkpointing

- GDB
  - Single session, no option to save/restore
  - MPI support
  - No thread support
  - Still in the experimental stage



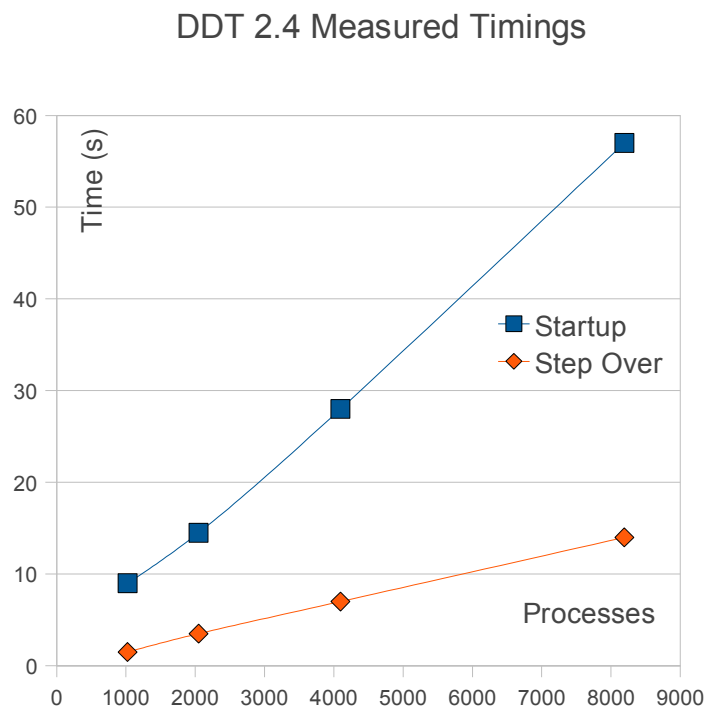


Current Scalability

- What has been achieved in last 12 months?
- Scalable GUI
  - For the first time debug 5,000 with same ease as 100
  - At a glance full stack and status of all processes
- 10x improvement in scale limits
  - Iterative improvement has brought benefit
  - Debugging 8k processes is comfortable
  - Regular users at 4k cores
  - Test rig emulating 32k cores at native speed
- High end platforms
  - BlueGene/P support added to list Q2/09
  - Cray XT4, XT5 users at scale - 8k
  - Ranger at TACC – Infiniband Sun Constellation cluster - 8k

Massive Scalability

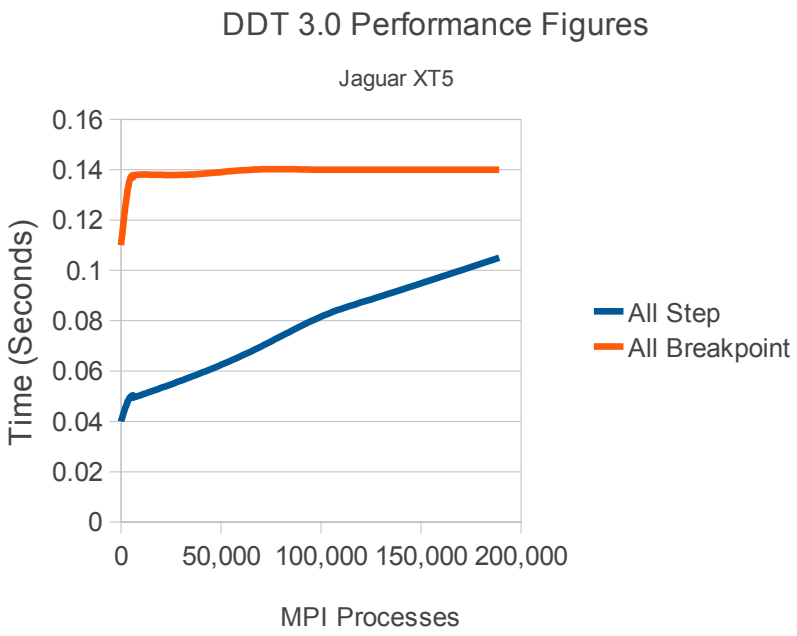
- Changing DDT communication network to be a shallow tree
  - Current limits ~4-16k processes
    - Linear performance
  - Progress towards >> 32k processes
    - Target of 232k



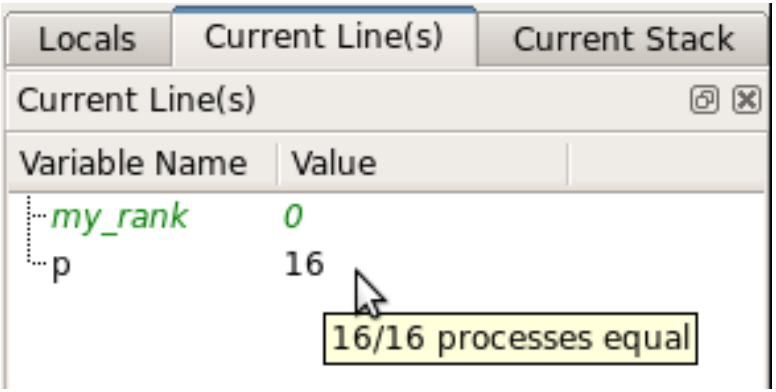
Dual Core 2.0 GHz 1GB RAM frontend

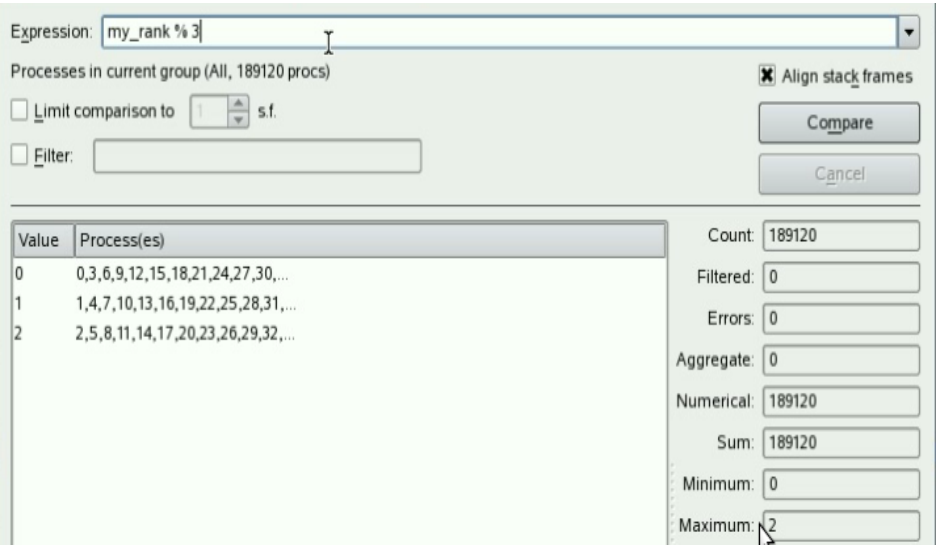
Massive Scalability

- Changing DDT communication network to be a shallow tree
  - Current limits ~4-16k processes
    - Linear performance
  - Progress towards >> 32k processes
    - Target of 232k
  - Massive increase in scalability
    - Aggregation of control messages..
    - ....and responses
  - Logarithmic performance



Presenting Data, Usefully





- Gather from every node
  - Potentially costly – if all data different
  - Easy if data mostly same
  - New ideas
    - Aggregated statistics
    - Probabilistic algorithms optimize performance – even in pathological case
  - ~130ms for 130,000 cores
- Watch this space!
  - With a fast and scalable architecture, new things become possible

Where Next?

- DDT is the first Petascale debugger..
  - A debugging tool has finally caught up with the hardware!
    - Work is in progress to port every feature for scale
    - Memory debugging, data visualization, ...
  - How can the infrastructure be built upon?
    - Does DDT offer the right framework for collaboration?
    - Can we encourage a codebase of user-generated MPI tools/utilities?

- ... but large clusters are a fraction of HPC
  - Most parallel development starts smaller
  - Is now starting even smaller: GPUs

Debugging Roadmap

April 2009	Q4 2009	Q2 2010	Q4 2010
DDT 2.4.1 General Release	DDT 2.5, 2.5.1	DDT 2.6	DDT 3.0
DDT for Blue Gene/P	Scalability Performance QT4	GPGPU Debugging for CUDA	Massive Scalability

Questions & Answers