allinea

Leaders in parallel software development tools

Allinea DDT: Your Partner in Finding Debugged Paths on Mira

Ian Lumb <ilumb@allinea.com>
Senior Systems Engineer, Allinea Software Inc.

Mira Community Conference 2013

[L2P] Summary

Leap to Petascale
Making the Move Towards Mira
May 22-25 -Argonne National Laboratory

- Petascaling for > 1 year
  - Petascaled infrastructure and UI
- Scaling for IBM Blue Gene /P
  - Acceptance testing at ALCF
- Scaling for IBM Blue Gene /Q
  - Addressing ALCF requirements
    - Early access for IBM Blue Gene /Q expected July 2012
- Architecture applicable elsewhere
  - Multicore/GPU??? architectures
- Exascaling ...

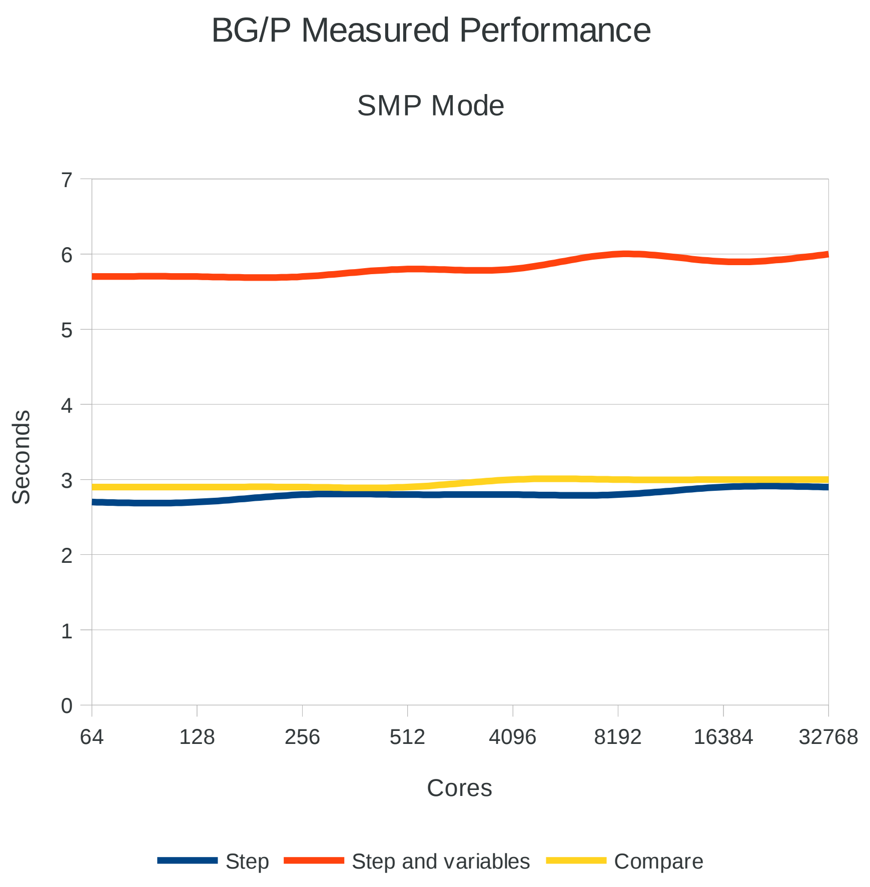
http://www.alcf.anl.gov/sites/www.alcf.anl.gov/files/L2PAllinea_0.pdf

A Path to Petascale on IBM BG /P

- Phase 1 [2010]
  - Cut memory usage per compute process at I/O node
  - Debuggers share common internal tables
    - Memory mapping of symbol tables
    - Raises limit to ~128 processes
- Delivered!

The memory mapped result

- Simplest to achieve – with benefits to multicore systems
  - Boosted max cores per I/O node to 256
- Reached 32K cores
  - 32,000 cores as quick as 64 cores
  - … flat – but not instantaneous
  - Most operations ~ 3 seconds
  - Close work with ANL – ran at scale on Intrepid

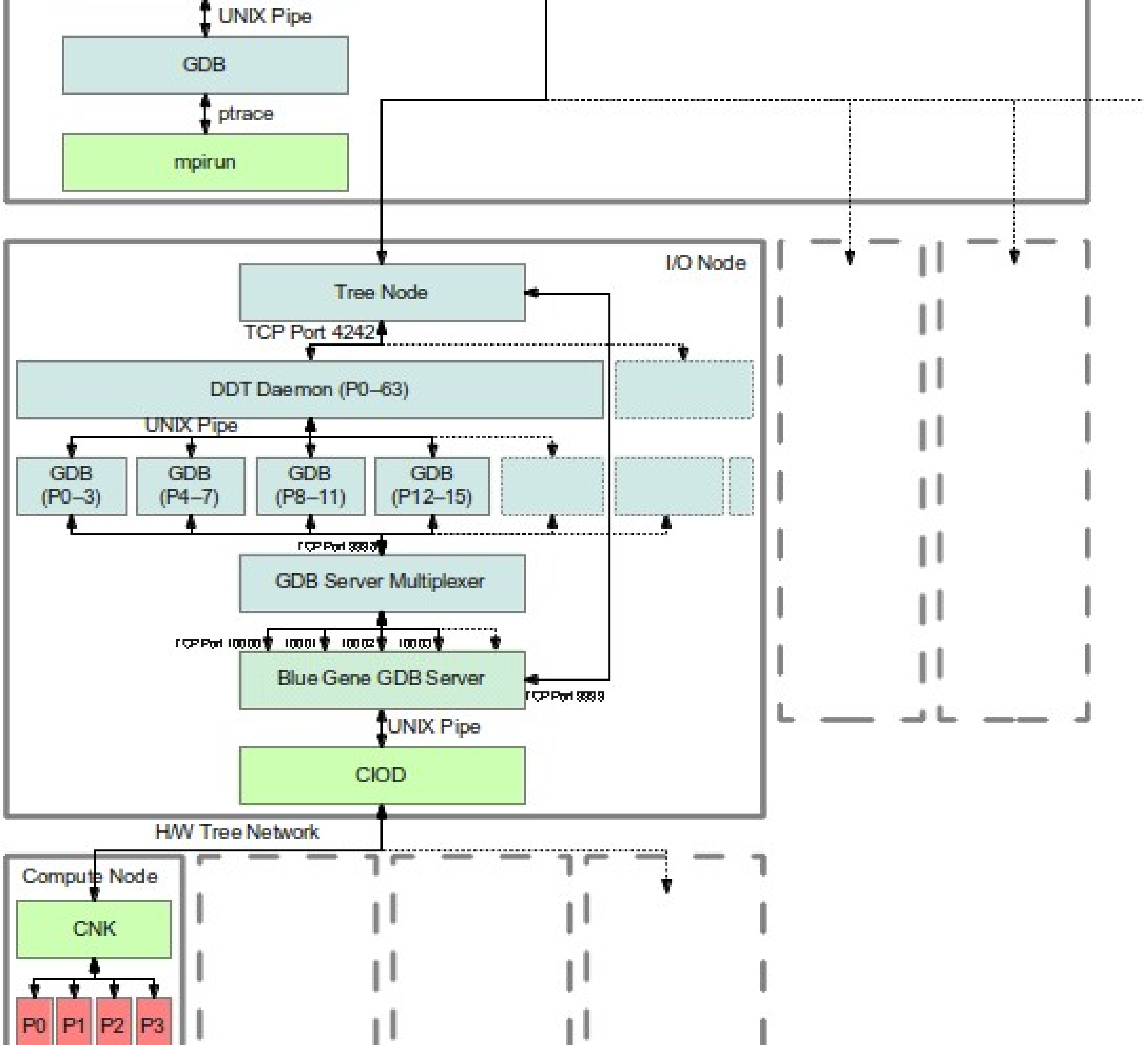


www.allinea.com

Petascale IBM Blue Gene /P Debugging

- Phase 2 [2011]
  - Reduce per-I/O-node daemon count
  - Reduces context thrashing: faster!
  - Each daemon handles multiple compute processes
    - Multiplexing commands and responses via CIOD
    - Multiplexing within the debugger
    - Cuts memory usage and improves speed
  - Limit 256-512 processes per I/O node
- Delivery: July 2012

Multiplexed Architecture



BG /P Case Study: Background

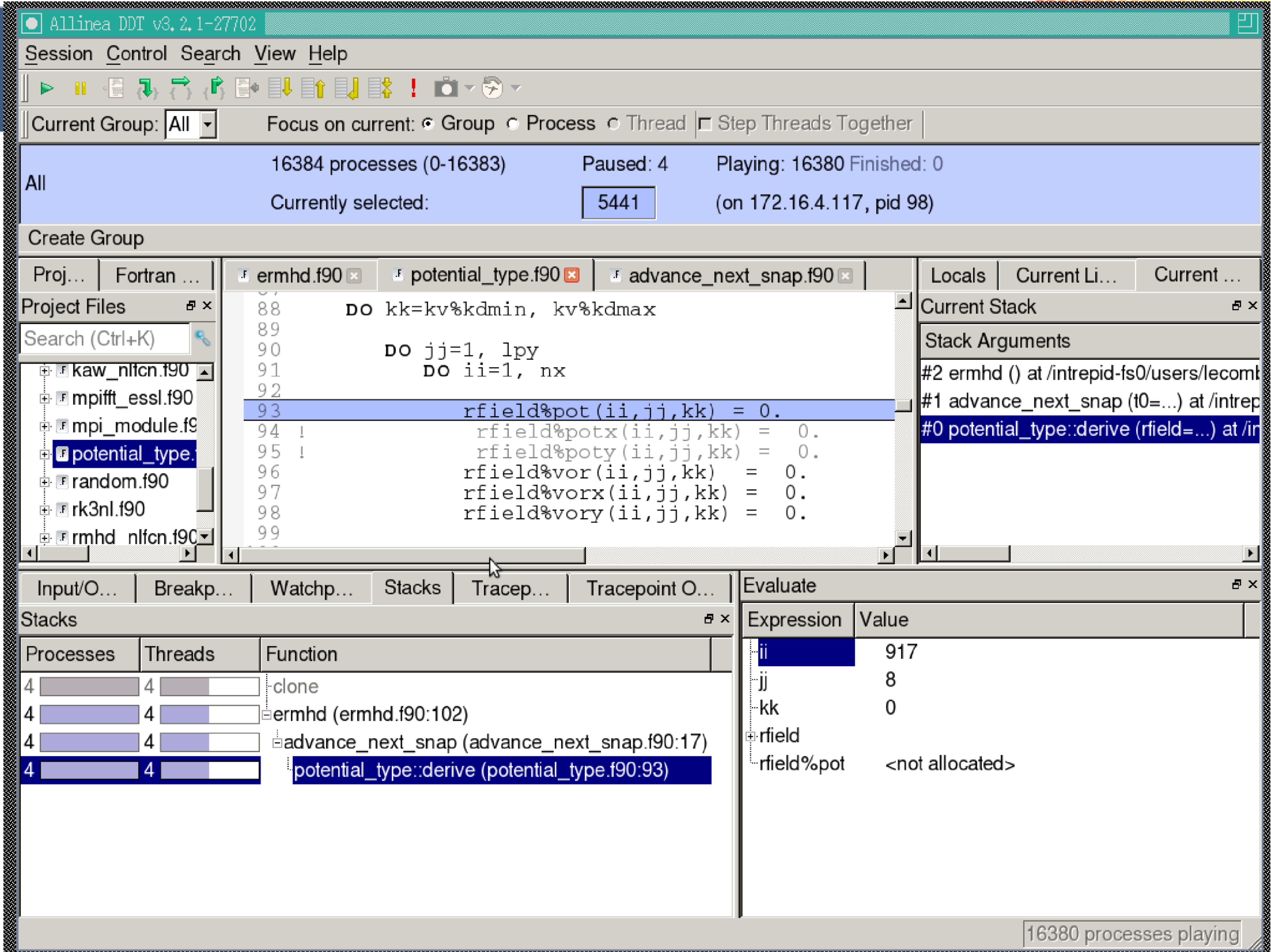
- Outstanding problems in heliospheric physics
  - Origin of the solar wind
  - Heating of the solar corona
- Large-scale numerical simulations
  - ***Simulation crashes at 16,386 MPI processes***

Why debug at scale?

- Increasing job sizes leads to unanticipated errors
  - Regular bugs
    - Logic issues and control flow
    - Data issues from larger data sets – eg. garbage in..., overflow
  - Increasing probability of independent random error
    - Memory errors/exhaustion – “random” bugs!
    - System problems – MPI and operating system
  - Coded boundaries
    - Algorithmic (performance) or hard-wired limits (“magic numbers”)
  - Unknown unknowns
- Machine time is too expensive to ignore failures!

BG /P Case Study: Debugging Process

- Reproduced the crash
  - Ran Allinea DDT in offline mode
    - Viewed HTML results via Web browser
      - Crash inside an MPI function call on about 128 of the 16384 cores
        - MPI implementation bug?
        - Memory bug?
- Ran Allinea DDT in offline mode again
  - Memory debugging enabled
    - Crash inside a harmless looking loop
      - Issue with loop index
- Ran Allinea DDT in GUI mode
  - Early calculation of the X-Y-Z grid is incorrect


Allinea DDT v3.2.1-27702
Session Control Search View Help
Current Group: All
Focus on current: ● Group ○ Process ○ Thread ☐ Step Threads Together
Create Group
Proj… Fortran …
Project Files
Search (Ctrl+K)
ermhd.f90 potential_type.f90 advance_next_snap.f90
Locals Current Li… Current …
Current Stack
Input/O… Breakp… Watchp… Stacks Tracep… Tracepoint O…
Evaluate

All 16384 processes (0-16383) Paused: 4 Playing: 16380 Finished: 0
Currently selected: 5441 (on 172.16.4.117, pid 98)

Project Files:
- kaw_nlfcn.f90
- mpifft_essl.f90
- mpi_module.f9
- potential_type.
- random.f90
- rk3nl.f90
- rmhd_nlfcn.f90

```
88       DO kk=kv%kdmin, kv%kdmax
89
90          DO jj=1, lpy
91             DO ii=1, nx
92
93                rfield%pot(ii,jj,kk) = 0.
94 !               rfield%potx(ii,jj,kk) =  0.
95 !               rfield%poty(ii,jj,kk) =  0.
96                rfield%vor(ii,jj,kk)  =  0.
97                rfield%vorx(ii,jj,kk) =  0.
98                rfield%vory(ii,jj,kk) =  0.
99
```

Stack Arguments
#2 ermhd () at /intrepid-fs0/users/lecomb
#1 advance_next_snap (t0=...) at /intrep
#0 potential_type::derive (rfield=...) at /in

Stacks

Processes	Threads	Function
4	4	clone
4	4	ermhd (ermhd.f90:102)
4	4	advance_next_snap (advance_next_snap.f90:17)
4	4	potential_type::derive (potential_type.f90:93)

Expression	Value
ii	917
jj	8
kk	0
rfield	
rfield%pot	<not allocated>

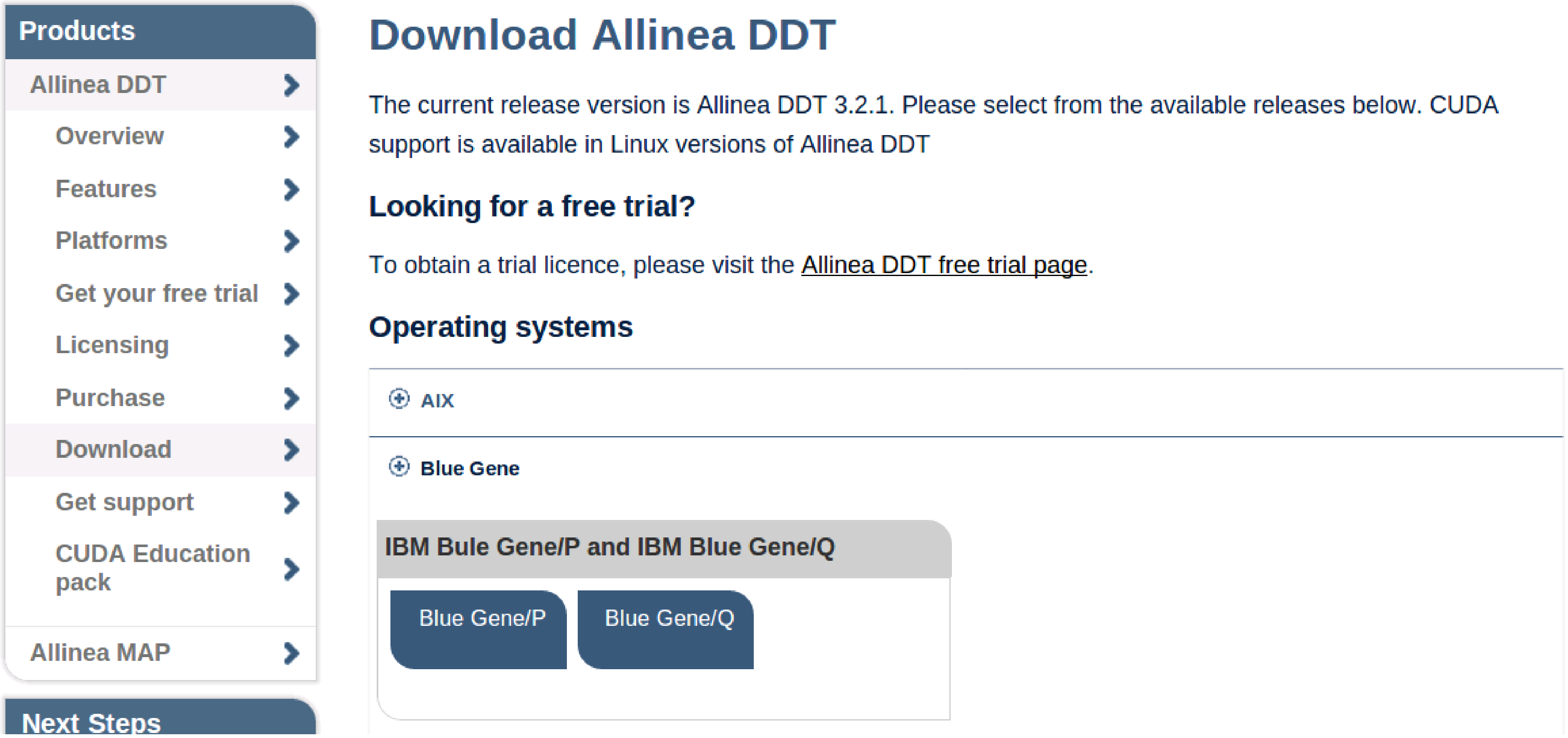
16380 processes playing

TRAFFIC

- Debugging
  - Transforming a broken program into a working one
- **How?**
  - **T**rack the problem
  - **R**eproduce
  - **A**utomate - (and simplify) the test case
  - **F**ind origins – where could the “infection” be from?
  - ***F**ocus – examine the origins*
  - ***I**solate – narrow down the origins*
  - ***C**orrect – fix and verify the testcase is successful*

Zeller A., “Why Programs Fail”, 2nd Edition, 2009

Allinea DDT 3.2.1 – October 2012



Download Allinea DDT

The current release version is Allinea DDT 3.2.1. Please select from the available releases below. CUDA support is available in Linux versions of Allinea DDT

Looking for a free trial?

To obtain a trial licence, please visit the Allinea DDT free trial page.

Operating systems

AIX

Blue Gene

IBM Bule Gene/P and IBM Blue Gene/Q

http://www.allinea.com/ddt-download/

allinea
www.allinea.com

Learn. Be Inspired. Find a Path.

Allinea DDT and Mira

“This tool has already proven its value in the migration of our early science applications onto Mira,” said Kalyan Kumaran, who manages ALCF’s applications performance engineering team. “These projects cover the range of scientific fields, numerical methods, programming models and computational approaches expected to run on Mira, so **accurate debugging is critical**.”

Allinea DDT

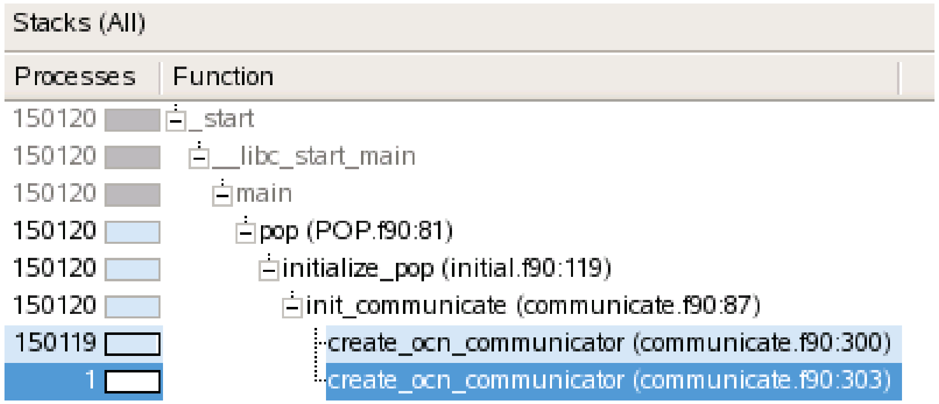
- Licensing
  - 32K-process permanent license
  - Full machine development license available (contact support)
- Startup overview
  - Compile –g –O0
    - OMP code compile -qsmp=omp:noauto:noopt
  - Softenv key "+ddt"
  - Need X11 server and ssh –X forwarding
  - [BG/P only] Start interactive job with *isub*
  - [BG/P or BG/Q] Run ddt and submit job through GUI
- More details:
  - [BG/P] http://www.alcf.anl.gov/resource-guides/allinea-ddt

http://www.alcf.anl.gov/sites/www.alcf.anl.gov/files/2013MiraCon_debugging_0.pdf

Fixing the everyday crash

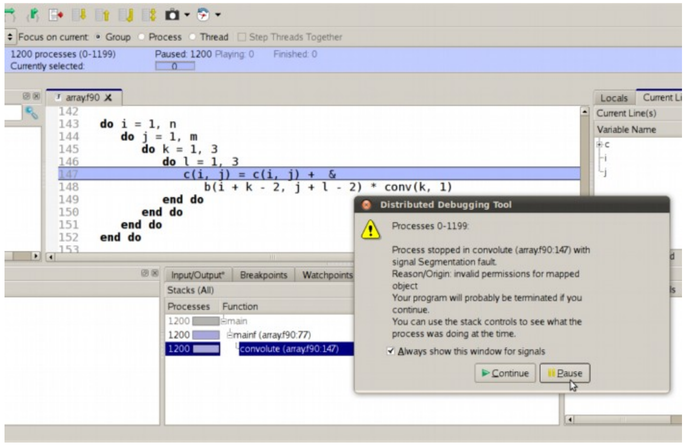
- The typical application crash or early exit:
  - Run your program in the debugger

    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
  - For others look deeper – at variables



Stacks (All)

Processes	Function
150120	_start
150120	__libc_start_main
150120	main
150120	pop (POP.f90:81)
150120	initialize_pop (initial.f90:119)
150120	init_communicate (communicate.f90:87)
150119	create_ocn_communicator (communicate.f90:300)
1	create_ocn_communicator (communicate.f90:303)



Focus on current: Group Process Thread Step Threads Together

1200 processes (0-1199) Paused: 1200 Playing: 0 Finished: 0
Currently selected: 0

array.f90

```
do i = 1, n
   do j = 1, m
      do k = 1, 3
         do l = 1, 3
            c(i, j) = c(i, j) +  &
               b(i + k - 2, j + l - 2) * conv(k, 1)
         end do
      end do
   end do
end do
```

Stacks (All)

Processes	Function
1200	main
1200	mainf (array.f90:77)
1200	convolute (array.f90:147)

Distributed Debugging Tool

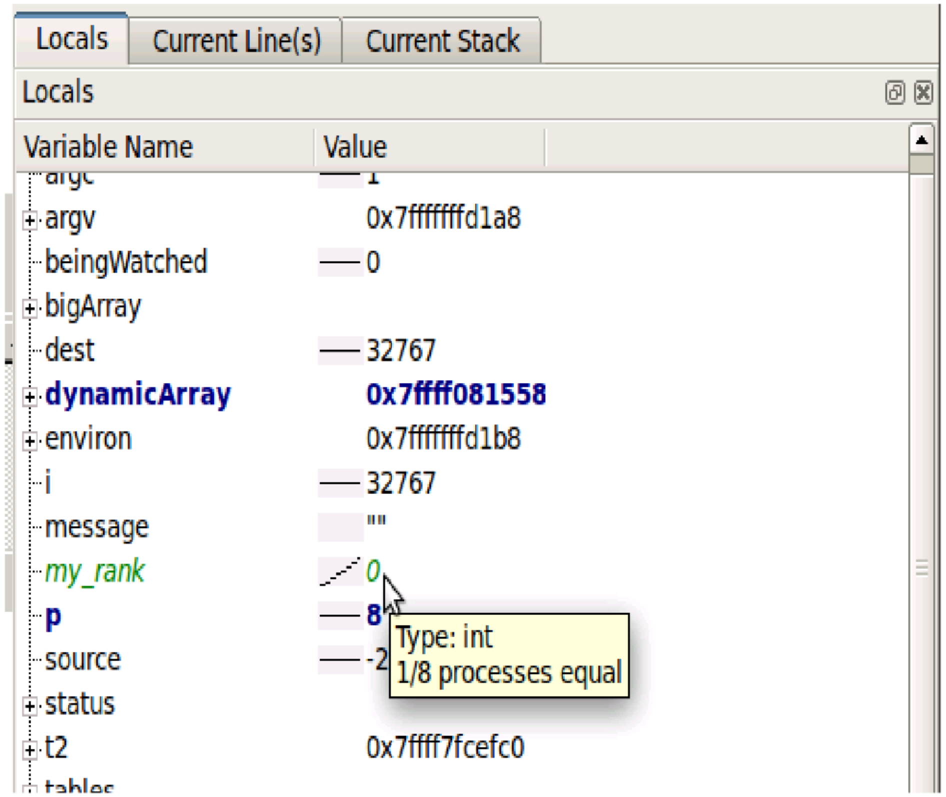
Processes 0-1199:

Process stopped in convolute (array.f90:147) with signal Segmentation fault.
Reason/Origin: invalid permissions for mapped 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

allinea
www.allinea.com

Simplifying the data deluge



Variable Name	Value
argc	1
argv	0x7fffffffd1a8
beingWatched	0
bigArray	
dest	32767
dynamicArray	**0x7ffff081558**
environ	0x7fffffffd1b8
i	32767
message	""
my_rank	*0*
p	**8**
source	-2
status	
t2	0x7ffff7fcefc0
tables	

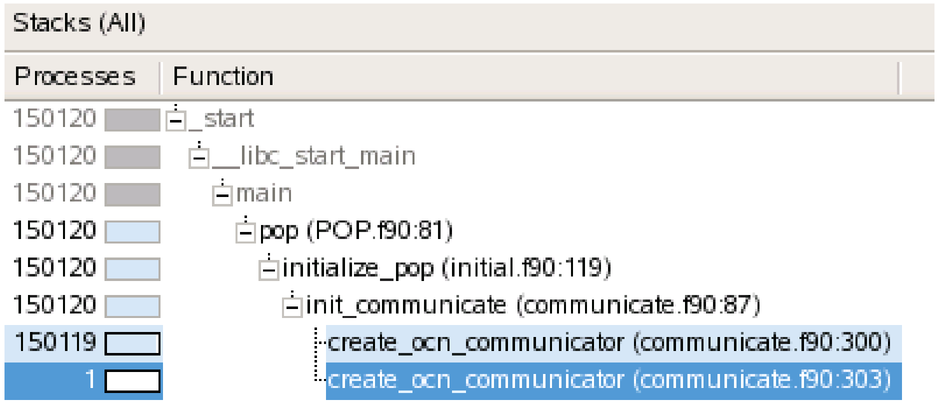
Type: int
1/8 processes equal

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

allinea
www.allinea.com

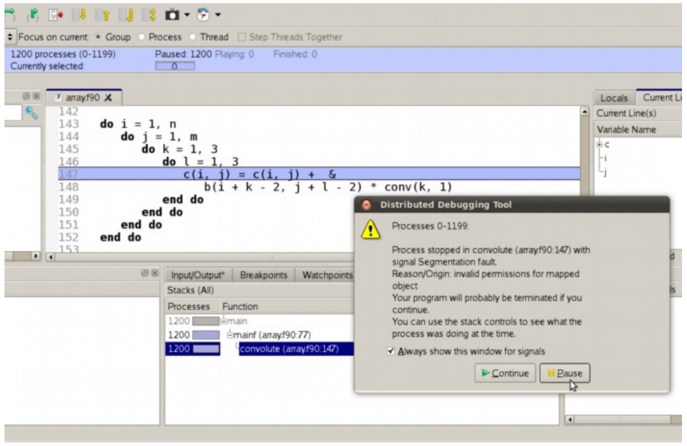
Allinea DDT: Proved to the extreme

- Scalability by design
  - User interface that scales
  - High performance tree architecture
- Proven performance at Petascale
  - Measured in milliseconds
  - **Routine use** at 100,000+ cores
- 300,000+ cores
  - Easy to use
  - Scalable GUI



Stacks (All)

Processes	Function
150120	_start
150120	__libc_start_main
150120	main
150120	pop (POP.f90:81)
150120	initialize_pop (initial.f90:119)
150120	init_communicate (communicate.f90:87)
150119	create_ocn_communicator (communicate.f90:300)
1	create_ocn_communicator (communicate.f90:303)



1200 processes (0-1199)   Paused: 1200 Playing: 0   Finished: 0
Currently selected: 0

array.f90

```
    do i = 1, n
       do j = 1, m
          do k = 1, 3
             do l = 1, 3
                c(i, j) = c(i, j) +  &
                   b(i + k - 2, j + l - 2) * conv(k, 1)
             end do
          end do
       end do
    end do
```

Variable Name: c, i, j

Stacks (All)

Processes	Function
1200	main
1200	mainf (array.f90:77)
1200	convolute (array.f90:147)

Distributed Debugging Tool

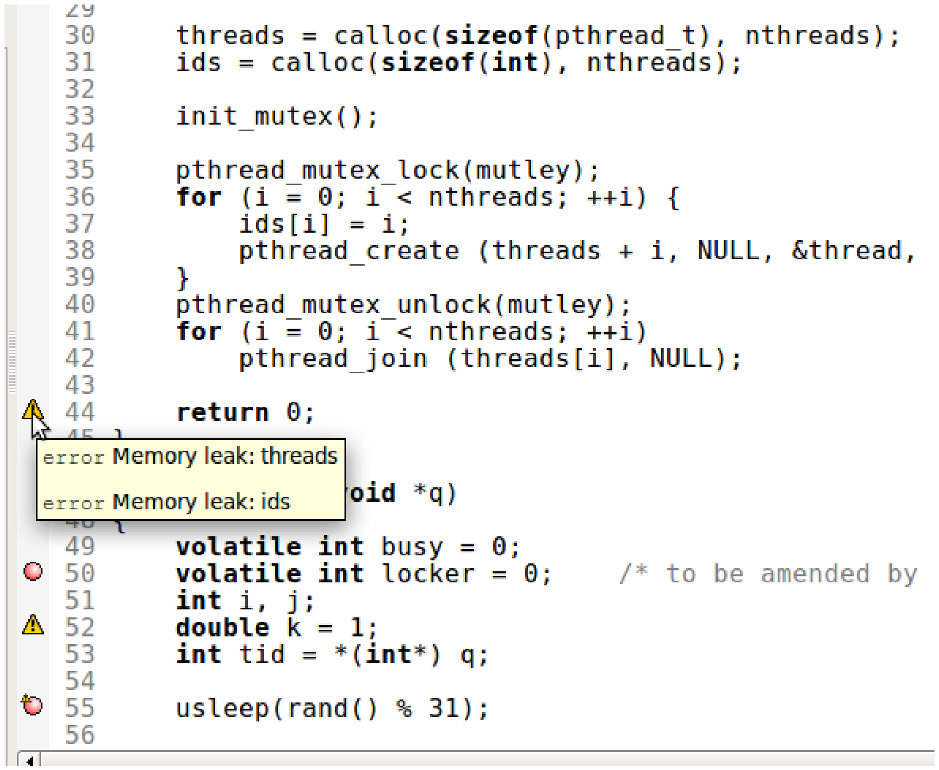
Processes 0-1199:

Process stopped in convolute (array.f90:147) with signal Segmentation fault.
Reason/Origin: invalid permissions for mapped 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

Allinea DDT: More than debugger

- Integrated automated detection of bugs
  - Static analysis
  - Memory leaks and errors
- Open plugin architecture
  - MPI checking tools
- Offline mode - debug in batch mode



```
    threads = calloc(sizeof(pthread_t), nthreads);
    ids = calloc(sizeof(int), nthreads);

    init_mutex();

    pthread_mutex_lock(mutley);
    for (i = 0; i < nthreads; ++i) {
        ids[i] = i;
        pthread_create (threads + i, NULL, &thread,
    }
    pthread_mutex_unlock(mutley);
    for (i = 0; i < nthreads; ++i)
        pthread_join (threads[i], NULL);

    return 0;
}

oid *q)
{
    volatile int busy = 0;
    volatile int locker = 0;     /* to be amended by
    int i, j;
    double k = 1;
    int tid = *(int*) q;

    usleep(rand() % 31);
```

error Memory leak: threads

error Memory leak: ids

allinea
www.allinea.com

Allinea DDT - Debugging++

- Productively **debug** your parallel code
- Completely **understand** your parallel code
  - Interact with data, algorithms, codes, programs and applications in real time
- **Develop** parallel your code from scratch
- **Port** parallel algorithms, codes, programs and applications to X
- **Scale** your algorithms, codes, programs and applications

www.allinea.com

The Allinea Environment: Benefits

- **At last:** a modern **integrated** environment for the HPC developer
- Supporting the lifecycle of application development and improvement
  - Productively debug code
  - Enhance application performance
- Designed for productivity
  - Consistent integrated easy to use tools
  - Enables effective HPC development
- Improve system usage
  - Fewer failed jobs
  - Higher application performance

PERFORM
the allinea environment
FIX

www.allinea.com

What's really new?