allinea

High performance tools to debug, profile, and analyze your applications

High-productivity development tools for science

Beau Paisley

bpaisley@allinea.com

allinea FORGE

allinea DDT

allinea MAP

allinea PERFORMANCE REPORTS

Industry Standard Tools

OAK RIDGE National Laboratory

cea

BLUE WATERS SUSTAINED PETASCALE COMPUTING

NERSC NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

H L R I S

Argonne NATIONAL LABORATORY

University of California Lawrence Livermore National Laboratory

SNIC

TACC

BERKELEY LAB

SHARCNET™

NSF

Los Alamos NATIONAL LABORATORY EST. 1943

JÜLICH FORSCHUNGSZENTRUM

GENCI

ONERA THE FRENCH AEROSPACE LAB

BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación

TECHNISCHE UNIVERSITÄT DRESDEN

CINECA

lrz

Science & Technology Facilities Council

epcc

hpc CYMRU WALES

ICHEC Irish Centre for High-End Computing

جامعة الملك عبدالعزيز KING ABDULAZIZ UNIVERSITY

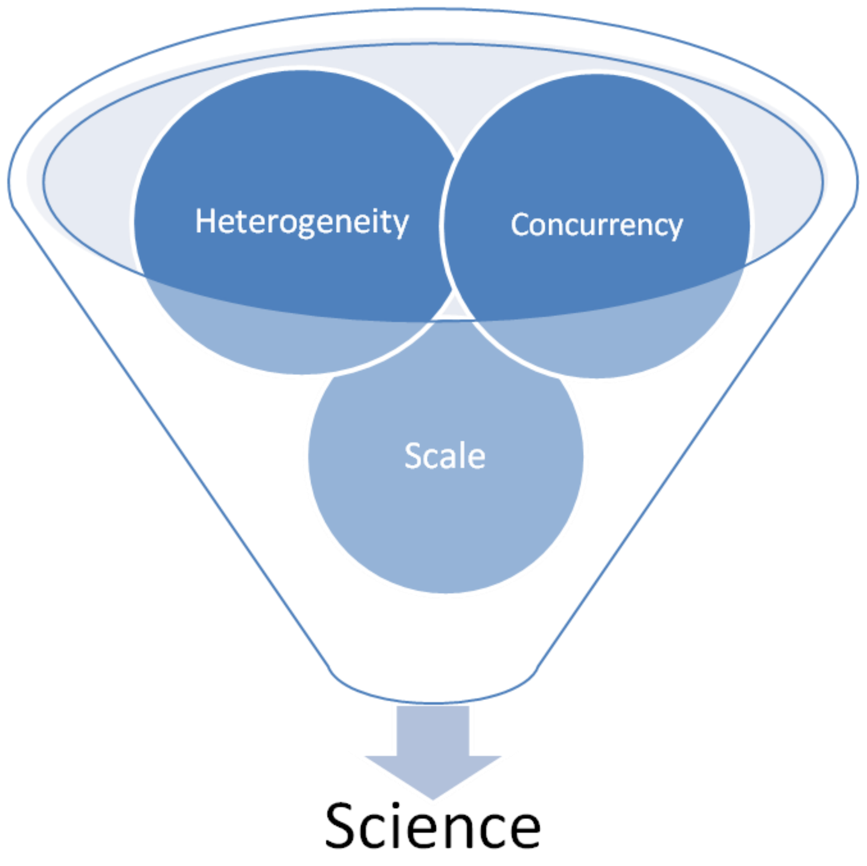
IFERC

أرامكو السعودية Saudi Aramco

NCHC 財團法人國家實驗研究院 國家高速網路與計算中心 National Center for High-performance Computing

allinea FORGE

Today's Challenge



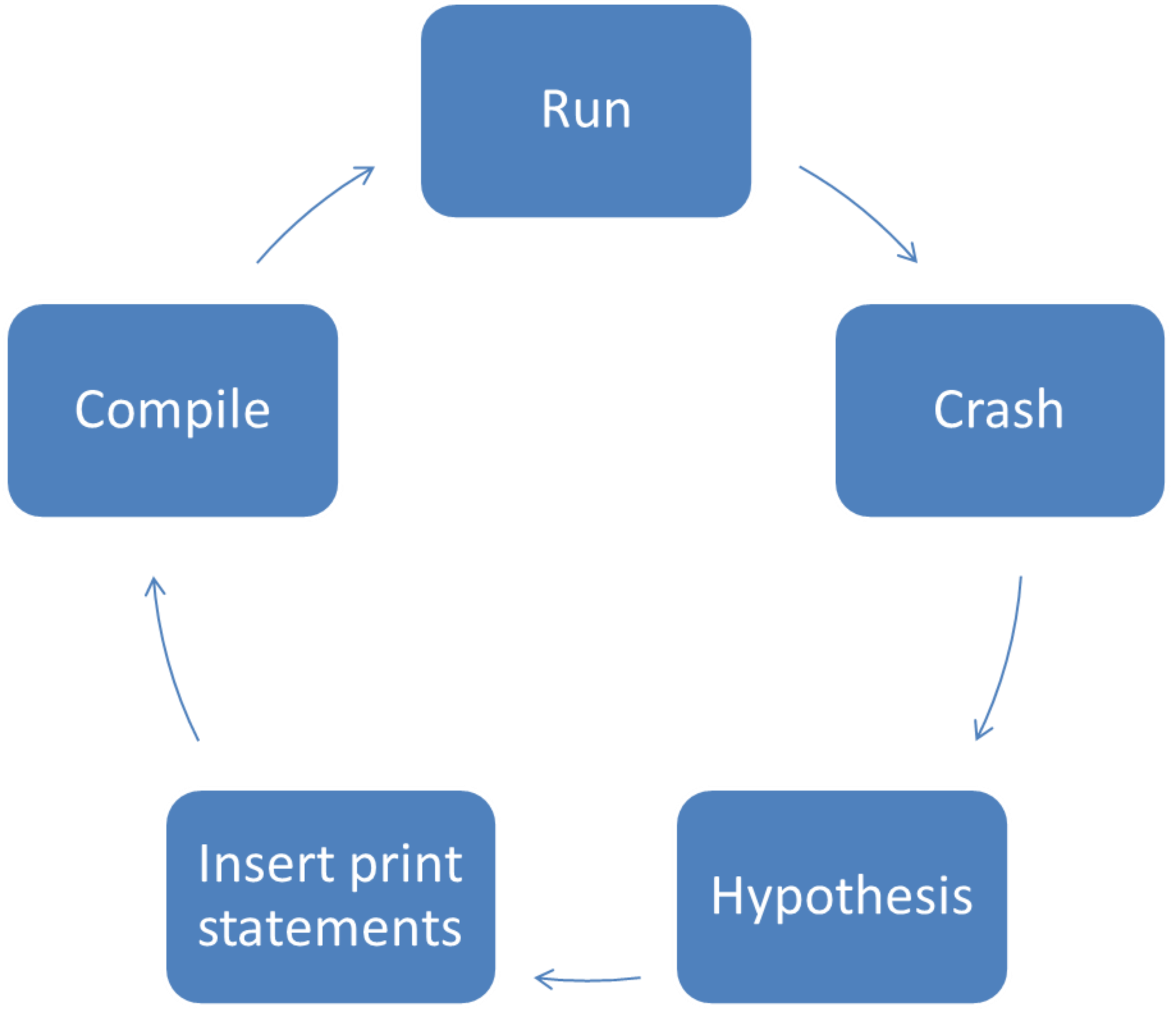
Q: What is the impact of current trends in HPC on your application?

Q: How can you make your science run well on the available system?

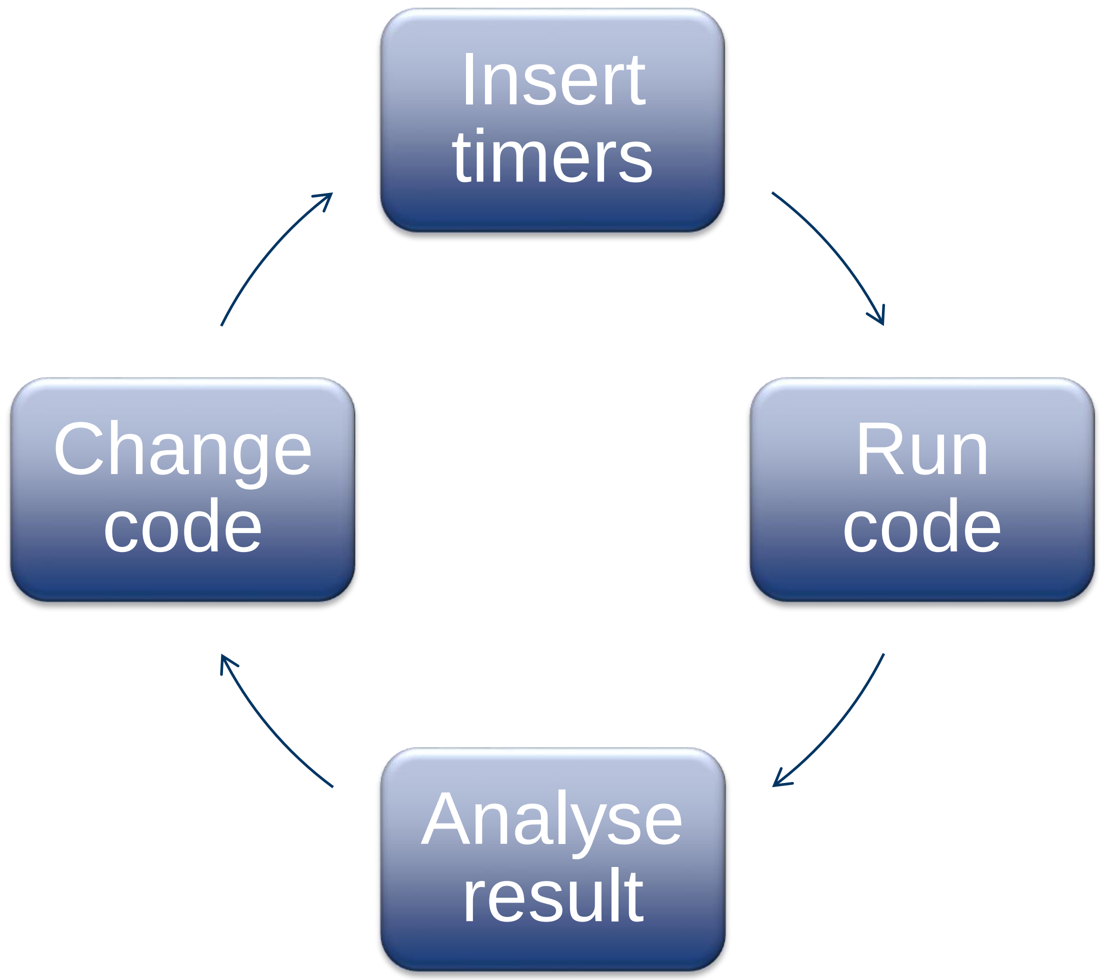
A: Development.

Development implies both fixing problems and optimizing the computation.

Debugging in Practice, …

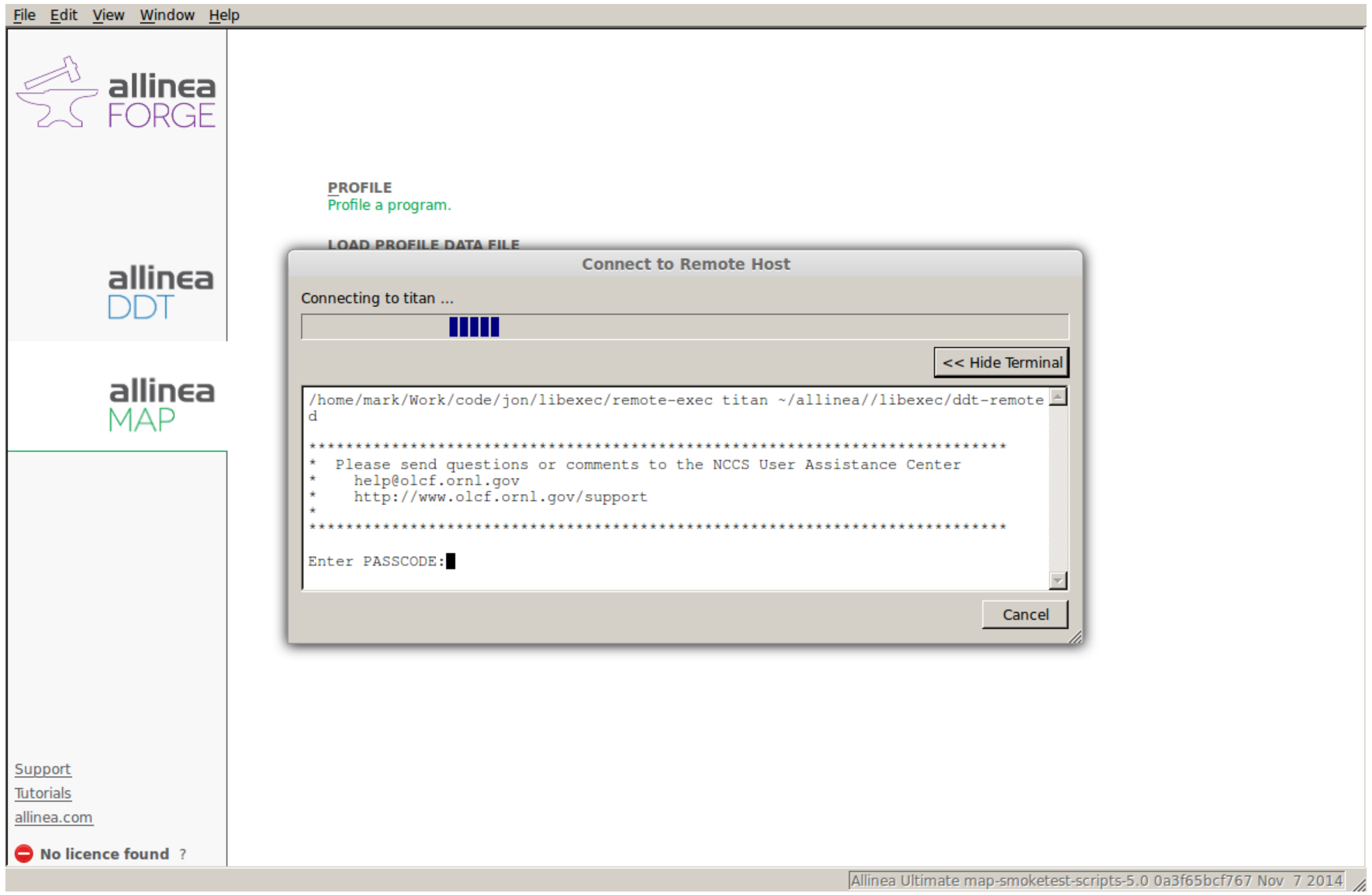


Optimization in Practice, …



HPC means being able to work productively on remote machines

- ✓ Linux
- ✓ OS/X
- ✓ Windows
- ✓ Multiple hop SSH
- ✓ RSA + Cryptocard
- ✓ Uses server license

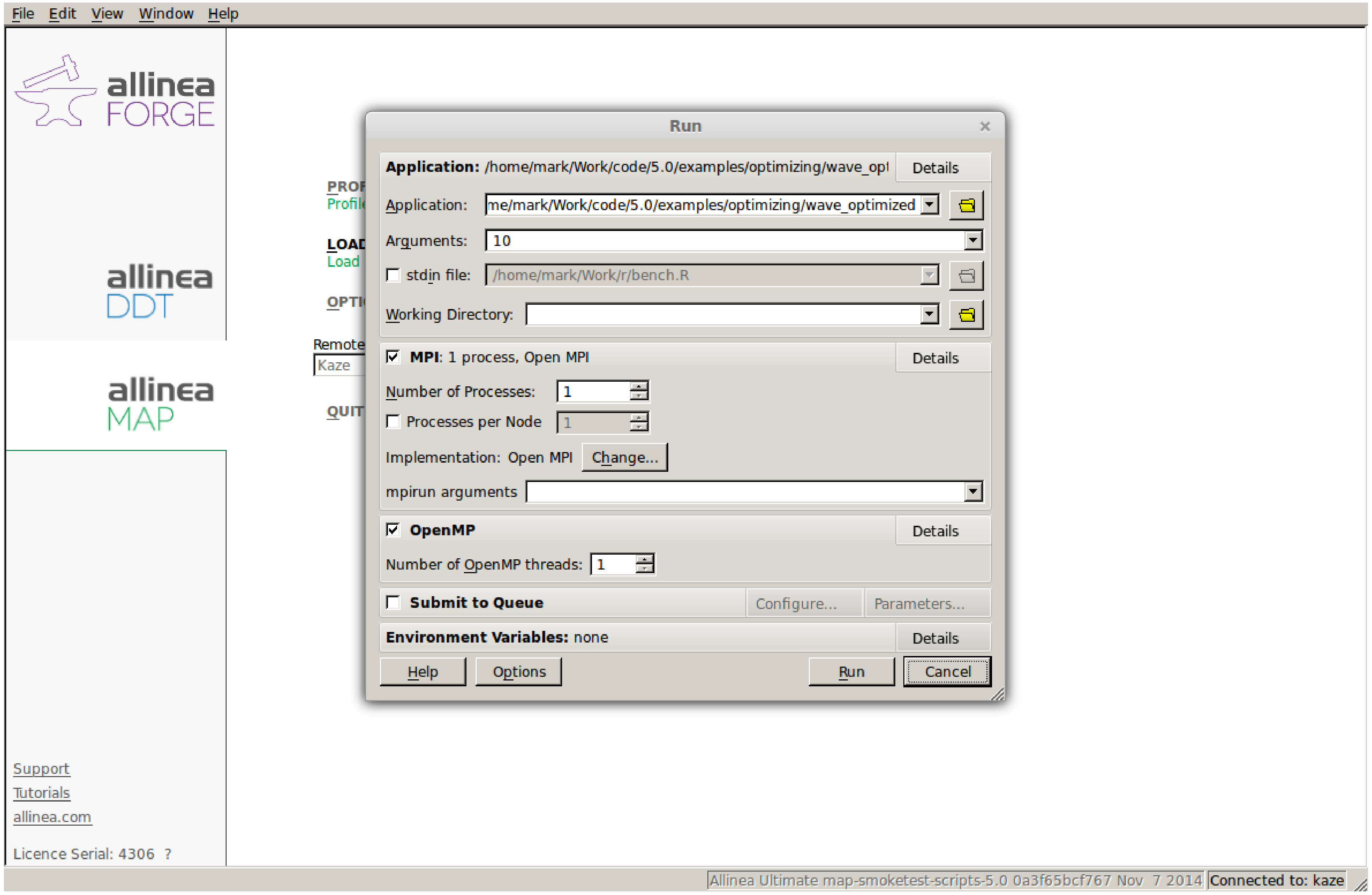


```
/home/mark/Work/code/jon/libexec/remote-exec titan ~/allinea//libexec/ddt-remote
d

****************************************************************************
*  Please send questions or comments to the NCCS User Assistance Center
*    help@olcf.ornl.gov
*    http://www.olcf.ornl.gov/support
*
****************************************************************************

Enter PASSCODE:
```

Submit to job queues or run interactively on any system



File Edit View Window Help

allinea FORGE

PROF
Profile

LOAD
Load

allinea DDT

OPTI

Remote
Kaze

allinea MAP

QUIT

Support
Tutorials
allinea.com

Licence Serial: 4306 ?

Run

Application: /home/mark/Work/code/5.0/examples/optimizing/wave_opt — Details

Application: me/mark/Work/code/5.0/examples/optimizing/wave_optimized

Arguments: 10

☐ stdin file: /home/mark/Work/r/bench.R

Working Directory:

☑ **MPI:** 1 process, Open MPI — Details

Number of Processes: 1

☐ Processes per Node 1

Implementation: Open MPI — Change...

mpirun arguments

☑ **OpenMP** — Details

Number of OpenMP threads: 1

☐ **Submit to Queue** — Configure... Parameters...

Environment Variables: none — Details

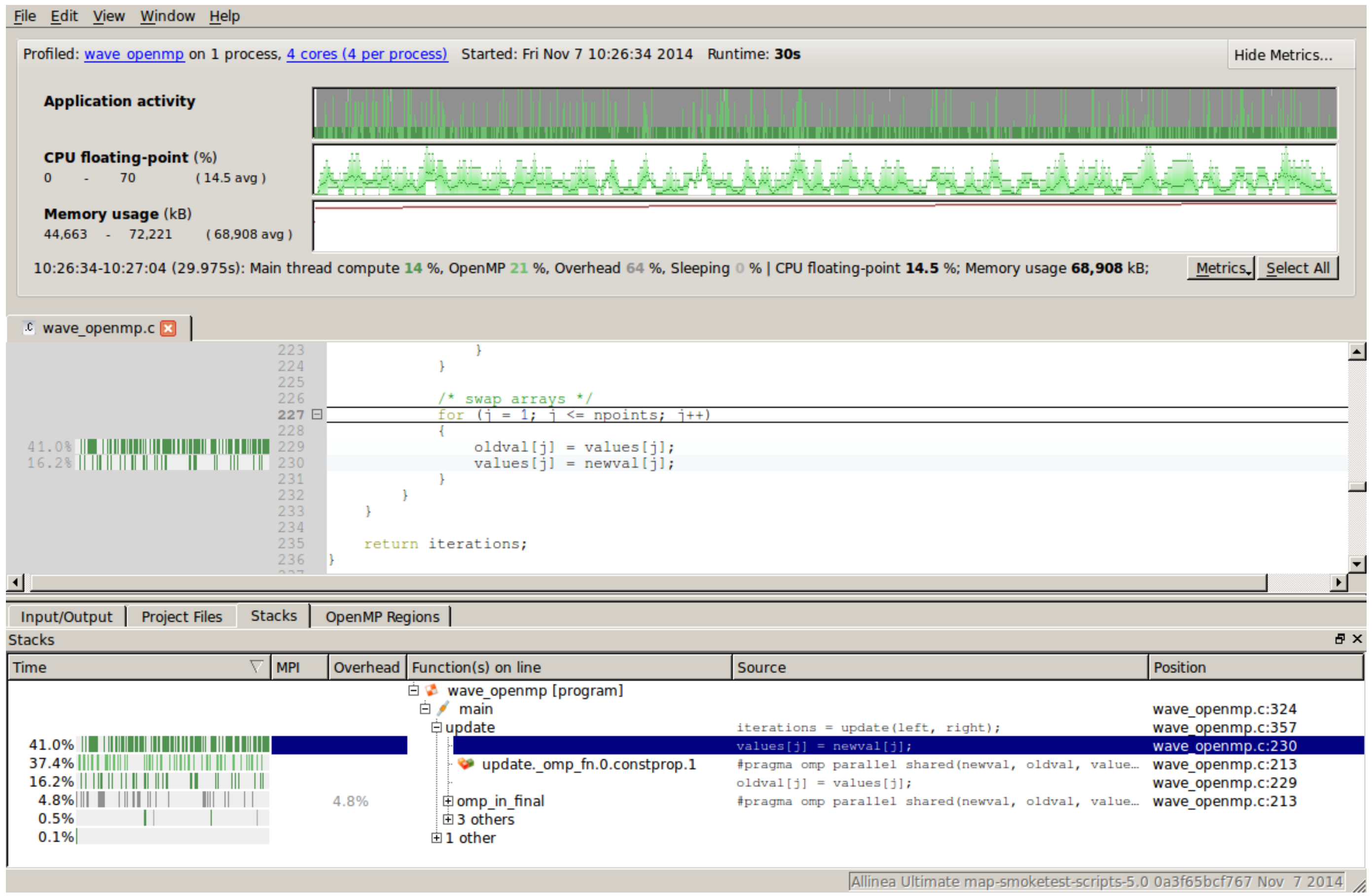
Help Options Run Cancel

Allinea Ultimate map-smoketest-scripts-5.0 0a3f65bcf767 Nov 7 2014 Connected to: kaze

allinea FORGE

Simplified Code Optimization

- Small data files
- <5% slowdown
- No instrumentation
- No recompilation



Profiled: wave_openmp on 1 process, 4 cores (4 per process) Started: Fri Nov 7 10:26:34 2014 Runtime: **30s**

Application activity

CPU floating-point (%)
0 - 70 (14.5 avg)

Memory usage (kB)
44,663 - 72,221 (68,908 avg)

10:26:34-10:27:04 (29.975s): Main thread compute 14 %, OpenMP 21 %, Overhead 64 %, Sleeping 0 % | CPU floating-point **14.5** %; Memory usage **68,908** kB;

```
                    }
                }

                /* swap arrays */
                for (j = 1; j <= npoints; j++)
                {
41.0%               oldval[j] = values[j];
16.2%               values[j] = newval[j];
                }
            }
        }

    return iterations;
}
```

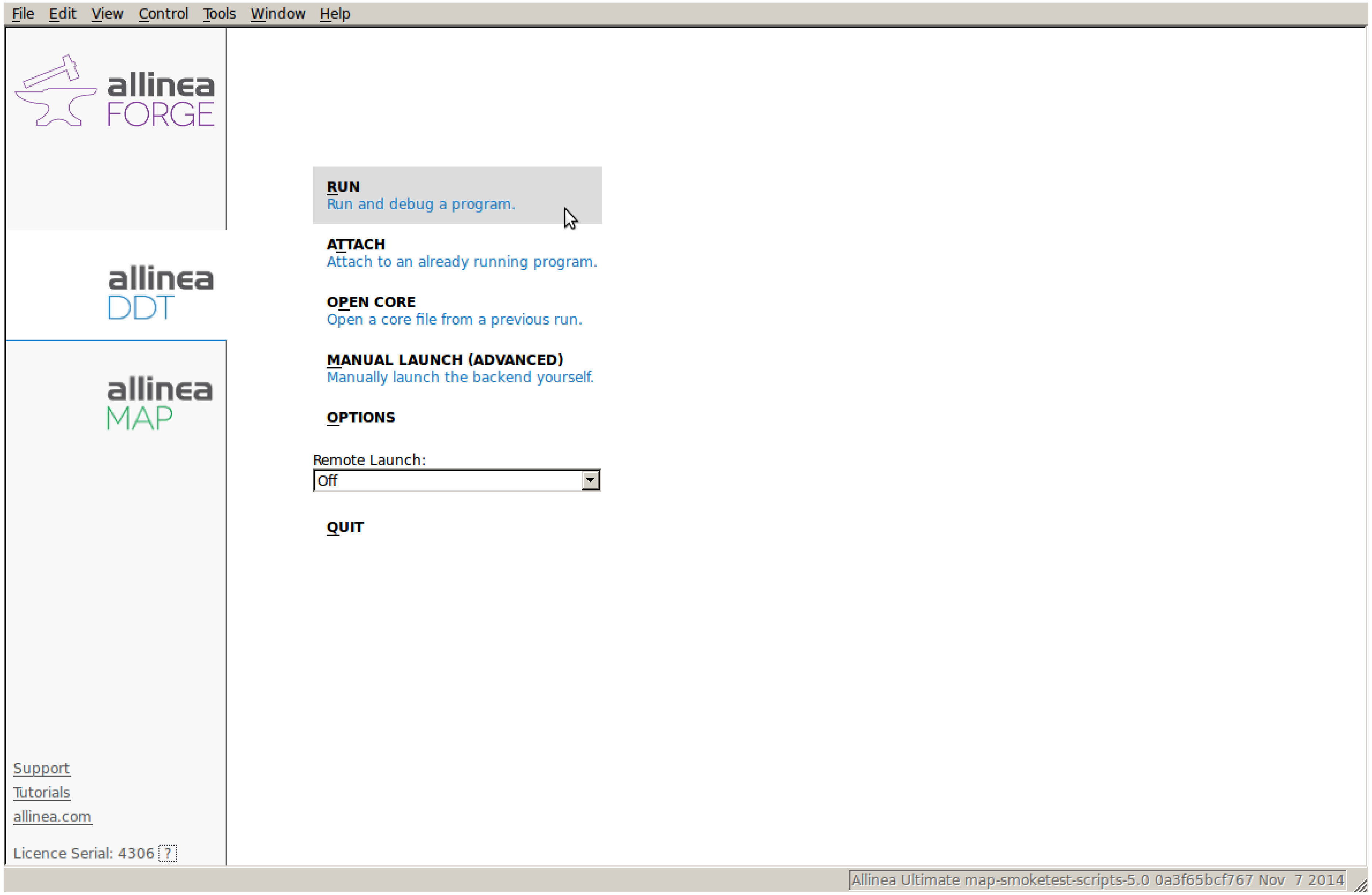
Stacks

Time	MPI	Overhead	Function(s) on line	Source	Position
			wave_openmp [program]		
			main		wave_openmp.c:324
			update	iterations = update(left, right);	wave_openmp.c:357
41.0%				values[j] = newval[j];	wave_openmp.c:230
37.4%			update._omp_fn.0.constprop.1	#pragma omp parallel shared(newval, oldval, value…	wave_openmp.c:213
16.2%				oldval[j] = values[j];	wave_openmp.c:229
4.8%		4.8%	omp_in_final	#pragma omp parallel shared(newval, oldval, value…	wave_openmp.c:213
0.5%			3 others		
0.1%			1 other		

Allinea Ultimate map-smoketest-scripts-5.0 0a3f65bcf767 Nov 7 2014

allinea MAP

While still connected to the server we switch to the debugger



RUN
Run and debug a program.

ATTACH
Attach to an already running program.

OPEN CORE
Open a core file from a previous run.

MANUAL LAUNCH (ADVANCED)
Manually launch the backend yourself.

OPTIONS

Remote Launch: Off

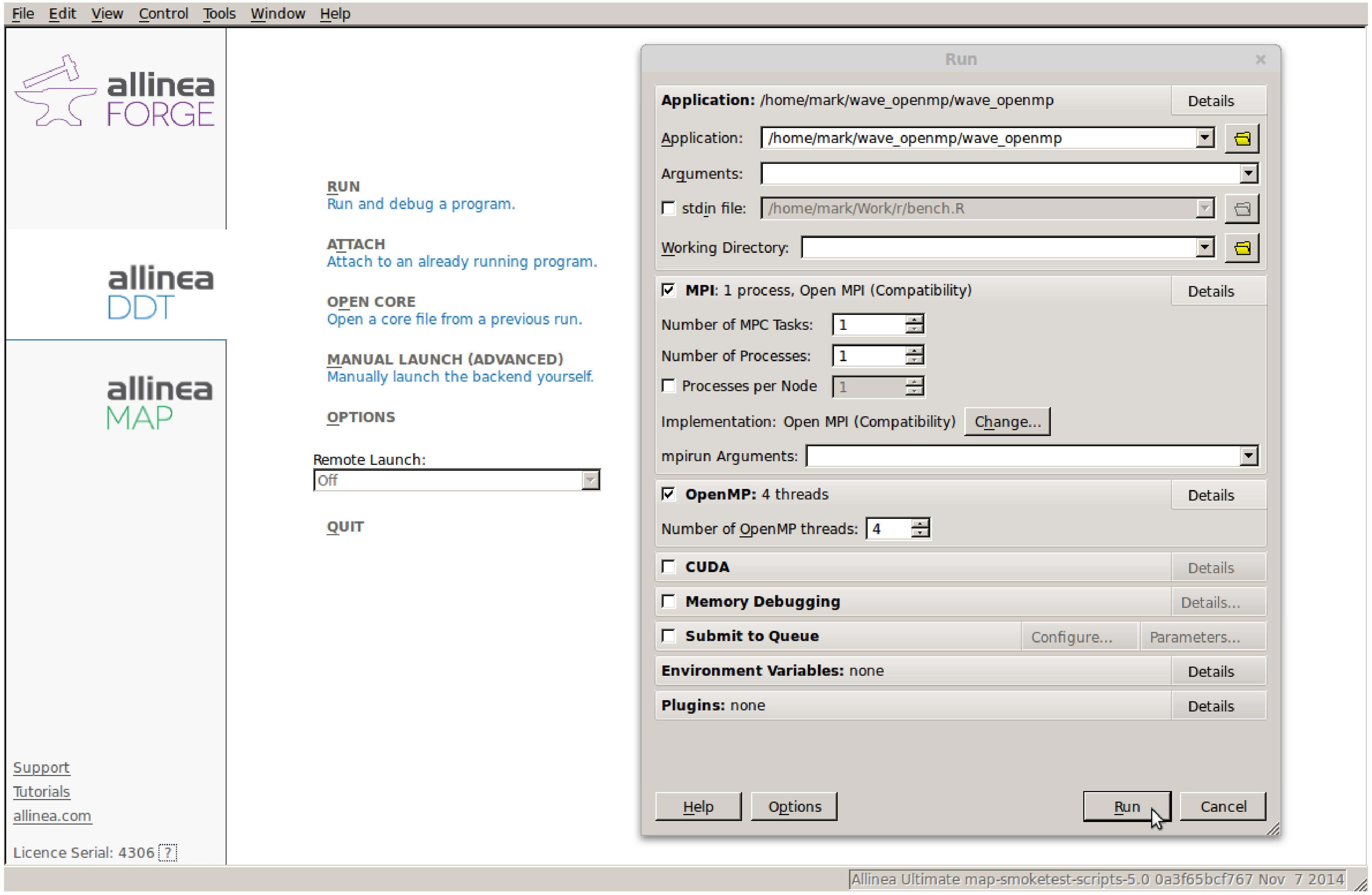
QUIT

Support
Tutorials
allinea.com

Licence Serial: 4306 ?

Allinea Ultimate map-smoketest-scripts-5.0 0a3f65bcf767 Nov 7 2014

It's already configured to reproduce the profiling run



Remote Launch: Off

Licence Serial: 4306

Application: /home/mark/wave_openmp/wave_openmp

Application: /home/mark/wave_openmp/wave_openmp

Arguments:

☐ stdin file: /home/mark/Work/r/bench.R

Working Directory:

☑ **MPI:** 1 process, Open MPI (Compatibility)

Number of MPC Tasks: 1

Number of Processes: 1

☐ Processes per Node 1

Implementation: Open MPI (Compatibility)

mpirun Arguments:

☑ **OpenMP:** 4 threads

Number of OpenMP threads: 4

☐ **CUDA**

☐ **Memory Debugging**

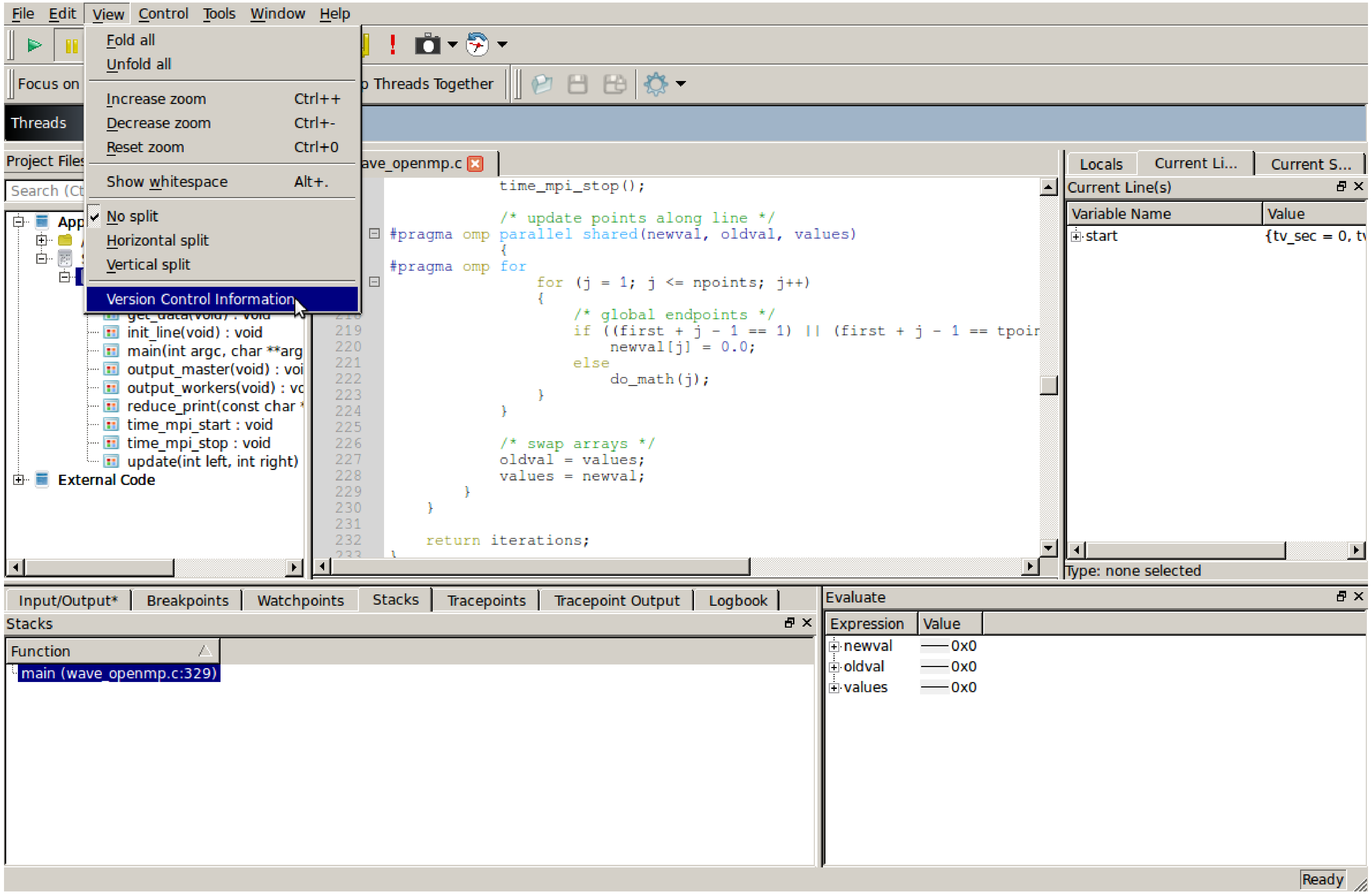
☐ **Submit to Queue**

Environment Variables: none

Plugins: none

Allinea Ultimate map-smoketest-scripts-5.0 0a3f65bcf767 Nov 7 2014

Our tools understand your version control system



```
get_data(void) : void
init_line(void) : void
main(int argc, char **arg
output_master(void) : voi
output_workers(void) : vc
reduce_print(const char *
time_mpi_start : void
time_mpi_stop : void
update(int left, int right)
External Code
```

```
                time_mpi_stop();

                /* update points along line */
#pragma omp parallel shared(newval, oldval, values)
                {
#pragma omp for
                        for (j = 1; j <= npoints; j++)
                        {
                                /* global endpoints */
                                if ((first + j - 1 == 1) || (first + j - 1 == tpoir
                                        newval[j] = 0.0;
                                else
                                        do_math(j);
                        }
                }

                /* swap arrays */
                oldval = values;
                values = newval;
        }
    }

    return iterations;
```

Current Line(s)

Variable Name	Value
start	{tv_sec = 0, tv

Type: none selected

Stacks

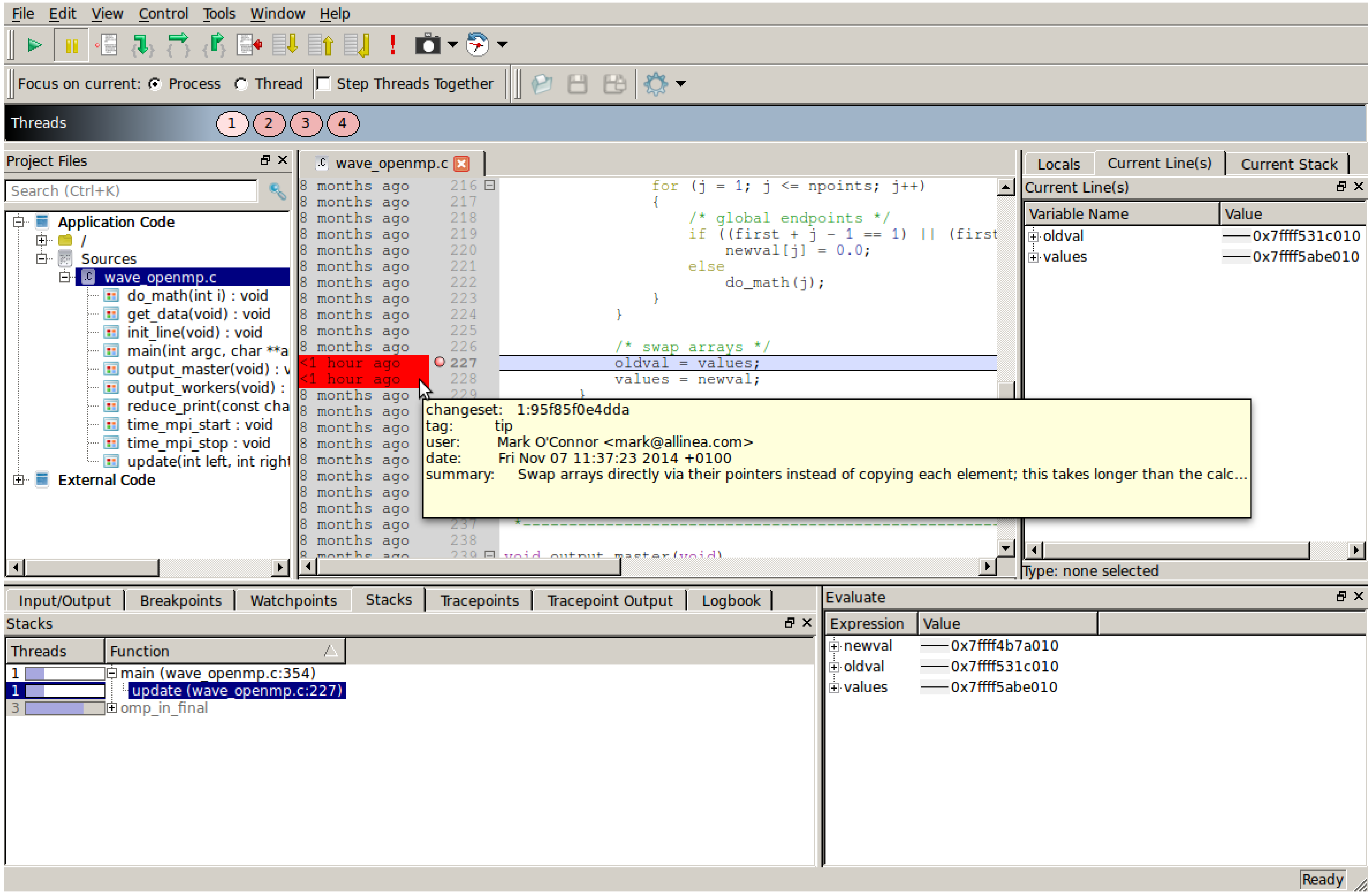
Function
main (wave_openmp.c:329)

Evaluate

Expression	Value
newval	0x0
oldval	0x0
values	0x0

allinea DDT

Most new bugs are in or around recently changed code



```
8 months ago   216          for (j = 1; j <= npoints; j++)
8 months ago   217          {
8 months ago   218              /* global endpoints */
8 months ago   219              if ((first + j - 1 == 1) || (first
8 months ago   220                  newval[j] = 0.0;
8 months ago   221              else
8 months ago   222                  do_math(j);
8 months ago   223          }
8 months ago   224      }
8 months ago   225
8 months ago   226      /* swap arrays */
<1 hour ago    227      oldval = values;
<1 hour ago    228      values = newval;
8 months ago   229  }
8 months ago
8 months ago
8 months ago
8 months ago
8 months ago
8 months ago
8 months ago
8 months ago
8 months ago   237  *-----------------------------------------
8 months ago   238
8 months ago   239  void output_master(void)
```

```
changeset:  1:95f85f0e4dda
tag:        tip
user:       Mark O'Connor <mark@allinea.com>
date:       Fri Nov 07 11:37:23 2014 +0100
summary:    Swap arrays directly via their pointers instead of copying each element; this takes longer than the calc...
```

Current Line(s)

Variable Name	Value
oldval	0x7ffff531c010
values	0x7ffff5abe010

Type: none selected

Stacks

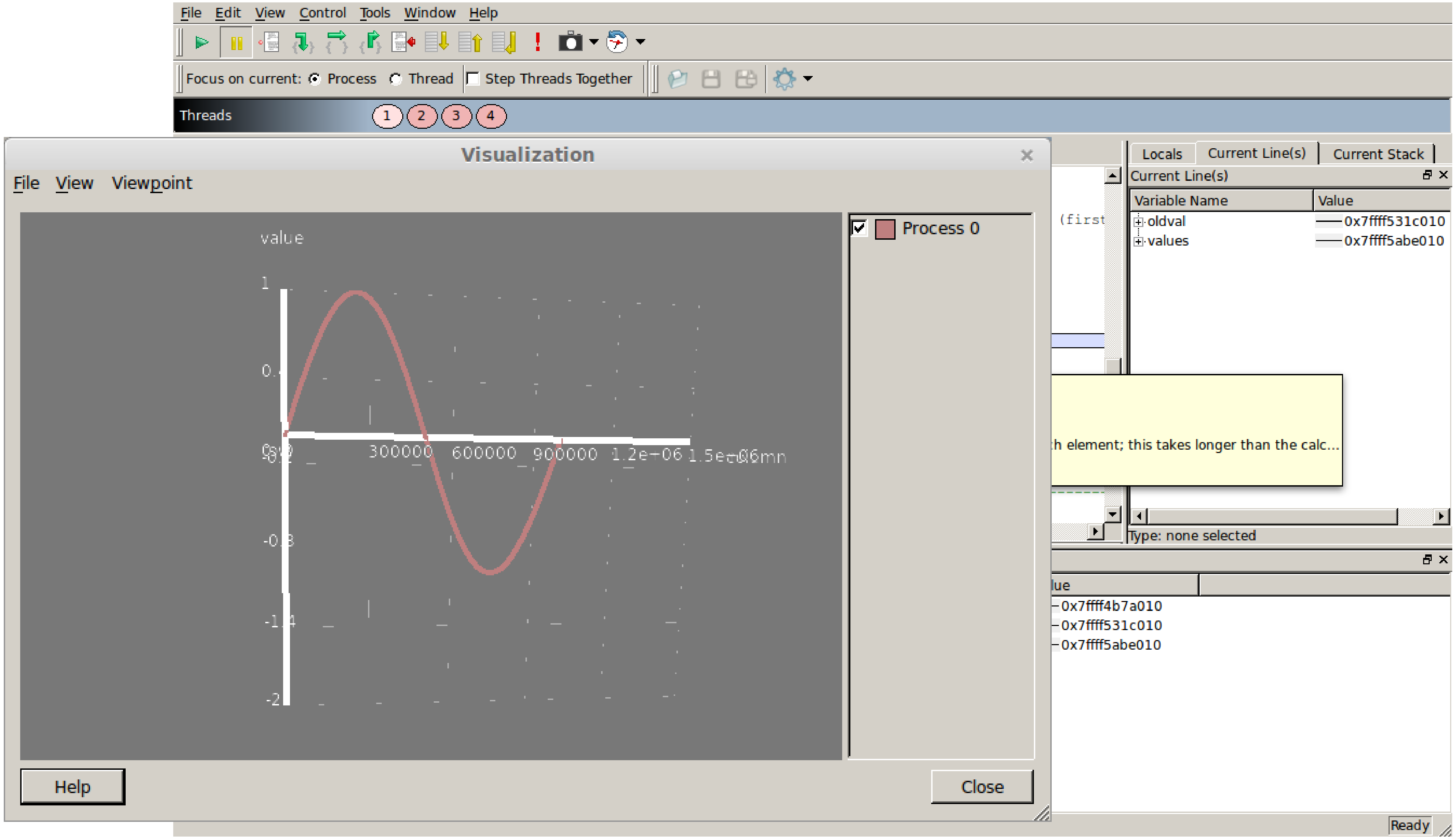
Threads	Function
1	main (wave_openmp.c:354)
1	update (wave_openmp.c:227)
3	omp_in_final

Evaluate

Expression	Value
newval	0x7ffff4b7a010
oldval	0x7ffff531c010
values	0x7ffff5abe010

allinea DDT

We can visualize multidimensional data across all processes



File Edit View Control Tools Window Help
Focus on current: Process Thread Step Threads Together
Threads 1 2 3 4
Visualization
File View Viewpoint
value
1
0.
-0.3
-1.4
-2
300000 600000 900000 1.2e+06 1.5e+06mn
Process 0
Help
Close
Locals Current Line(s) Current Stack
Ready

Current Line(s)

Variable Name	Value
oldval	0x7ffff531c010
values	0x7ffff5abe010

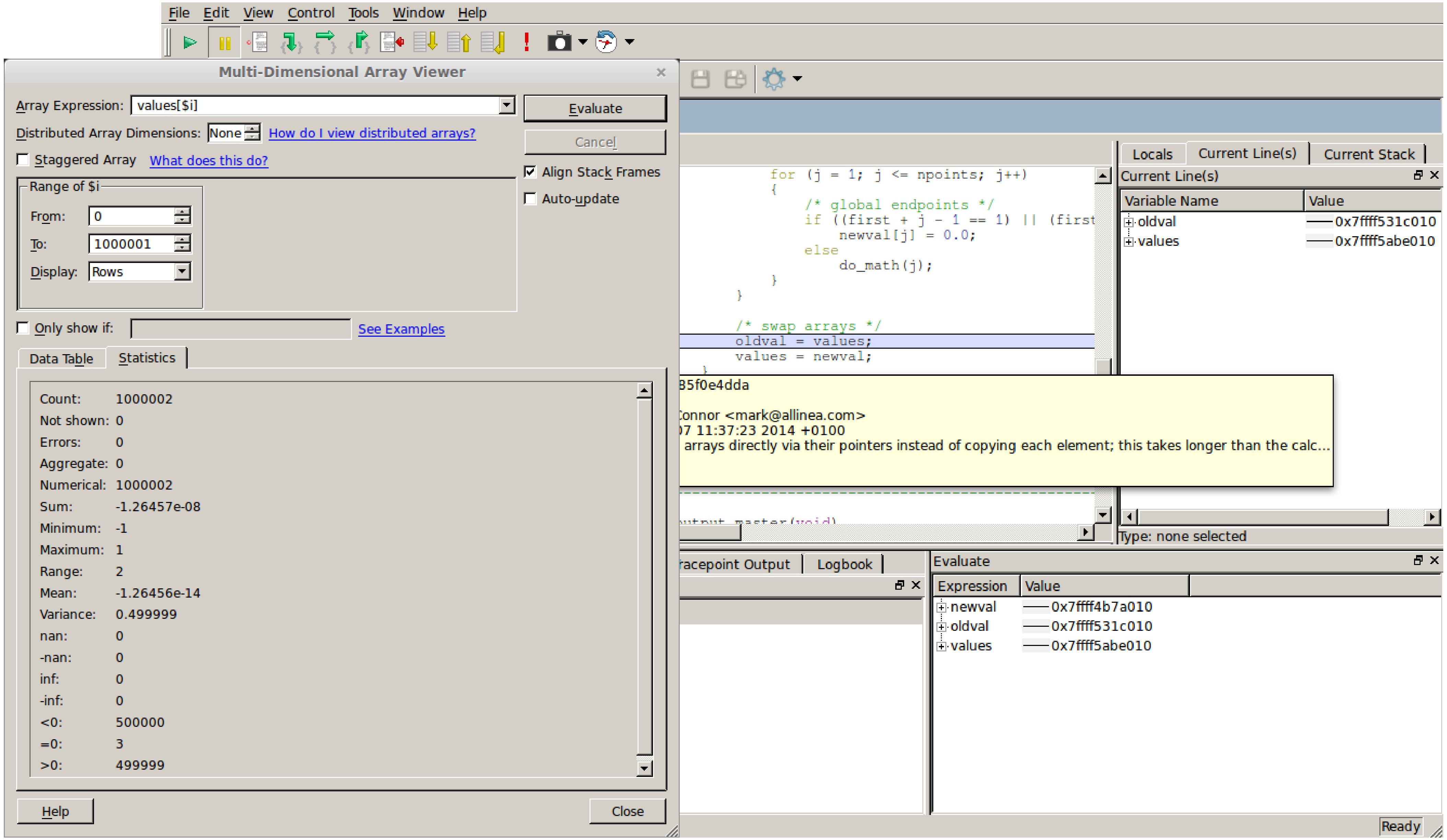
Type: none selected

…h element; this takes longer than the calc…

lue
0x7ffff4b7a010
0x7ffff531c010
0x7ffff5abe010

allinea DDT

And generate statistical summaries of their contents



Array Expression: values[$i]
Distributed Array Dimensions: None
Staggered Array: ☐
Align Stack Frames: ☑
Auto-update: ☐

Range of $i
From: 0
To: 1000001
Display: Rows

Only show if: ☐

```
Count:      1000002
Not shown:  0
Errors:     0
Aggregate:  0
Numerical:  1000002
Sum:        -1.26457e-08
Minimum:    -1
Maximum:    1
Range:      2
Mean:       -1.26456e-14
Variance:   0.499999
nan:        0
-nan:       0
inf:        0
-inf:       0
<0:         500000
=0:         3
>0:         499999
```

```
            for (j = 1; j <= npoints; j++)
            {
                /* global endpoints */
                if ((first + j - 1 == 1) || (first
                    newval[j] = 0.0;
                else
                    do_math(j);
            }
        }

        /* swap arrays */
        oldval = values;
        values = newval;
```

```
35f0e4dda

Connor <mark@allinea.com>
07 11:37:23 2014 +0100
 arrays directly via their pointers instead of copying each element; this takes longer than the calc...
```

Current Line(s)

Variable Name	Value
oldval	0x7ffff531c010
values	0x7ffff5abe010

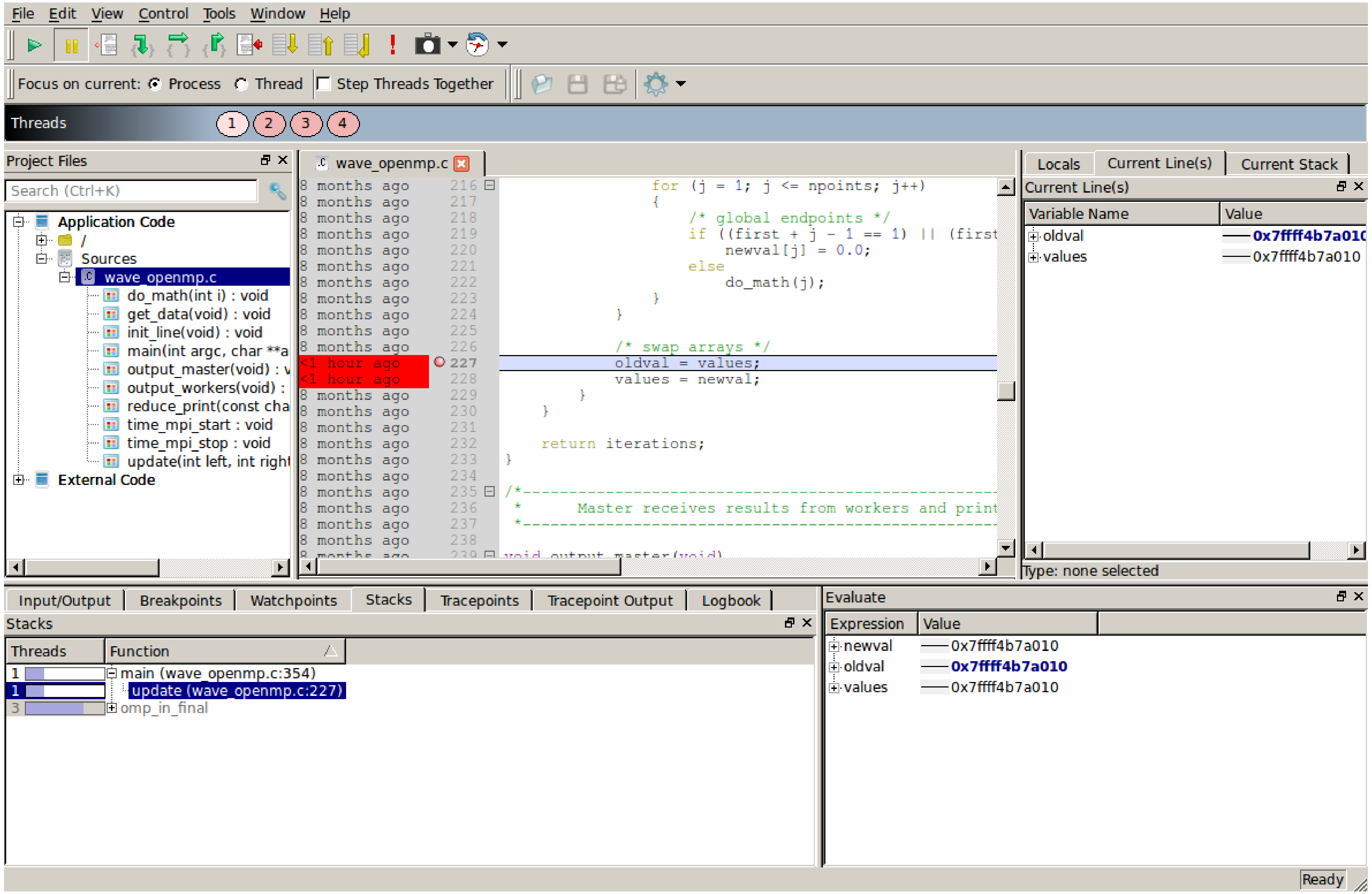
Type: none selected

Evaluate

Expression	Value
newval	0x7ffff4b7a010
oldval	0x7ffff531c010
values	0x7ffff5abe010

allinea DDT

Variables are compared across all threads and processes automatically



File Edit View Control Tools Window Help
Focus on current: (•) Process ( ) Thread [ ] Step Threads Together
Threads 1 2 3 4
Project Files
Search (Ctrl+K)
Locals | Current Line(s) | Current Stack
Input/Output | Breakpoints | Watchpoints | Stacks | Tracepoints | Tracepoint Output | Logbook
Ready

Project Files:

- Application Code
  - /
  - Sources
    - wave_openmp.c
      - do_math(int i) : void
      - get_data(void) : void
      - init_line(void) : void
      - main(int argc, char **a
      - output_master(void) : v
      - output_workers(void) :
      - reduce_print(const cha
      - time_mpi_start : void
      - time_mpi_stop : void
      - update(int left, int righ
- External Code

wave_openmp.c

```
8 months ago  216                  for (j = 1; j <= npoints; j++)
8 months ago  217                  {
8 months ago  218                      /* global endpoints */
8 months ago  219                      if ((first + j - 1 == 1) || (first
8 months ago  220                          newval[j] = 0.0;
8 months ago  221                      else
8 months ago  222                          do_math(j);
8 months ago  223                  }
8 months ago  224              }
8 months ago  225
8 months ago  226              /* swap arrays */
<1 hour ago   227              oldval = values;
<1 hour ago   228              values = newval;
8 months ago  229          }
8 months ago  230      }
8 months ago  231
8 months ago  232      return iterations;
8 months ago  233  }
8 months ago  234
8 months ago  235  /*------------------------------------------------------------
8 months ago  236   *      Master receives results from workers and print
8 months ago  237   *------------------------------------------------------------
8 months ago  238
8 months ago  239  void output_master(void)
```

Current Line(s)

Variable Name	Value
oldval	**0x7ffff4b7a010**
values	0x7ffff4b7a010

Type: none selected

Stacks

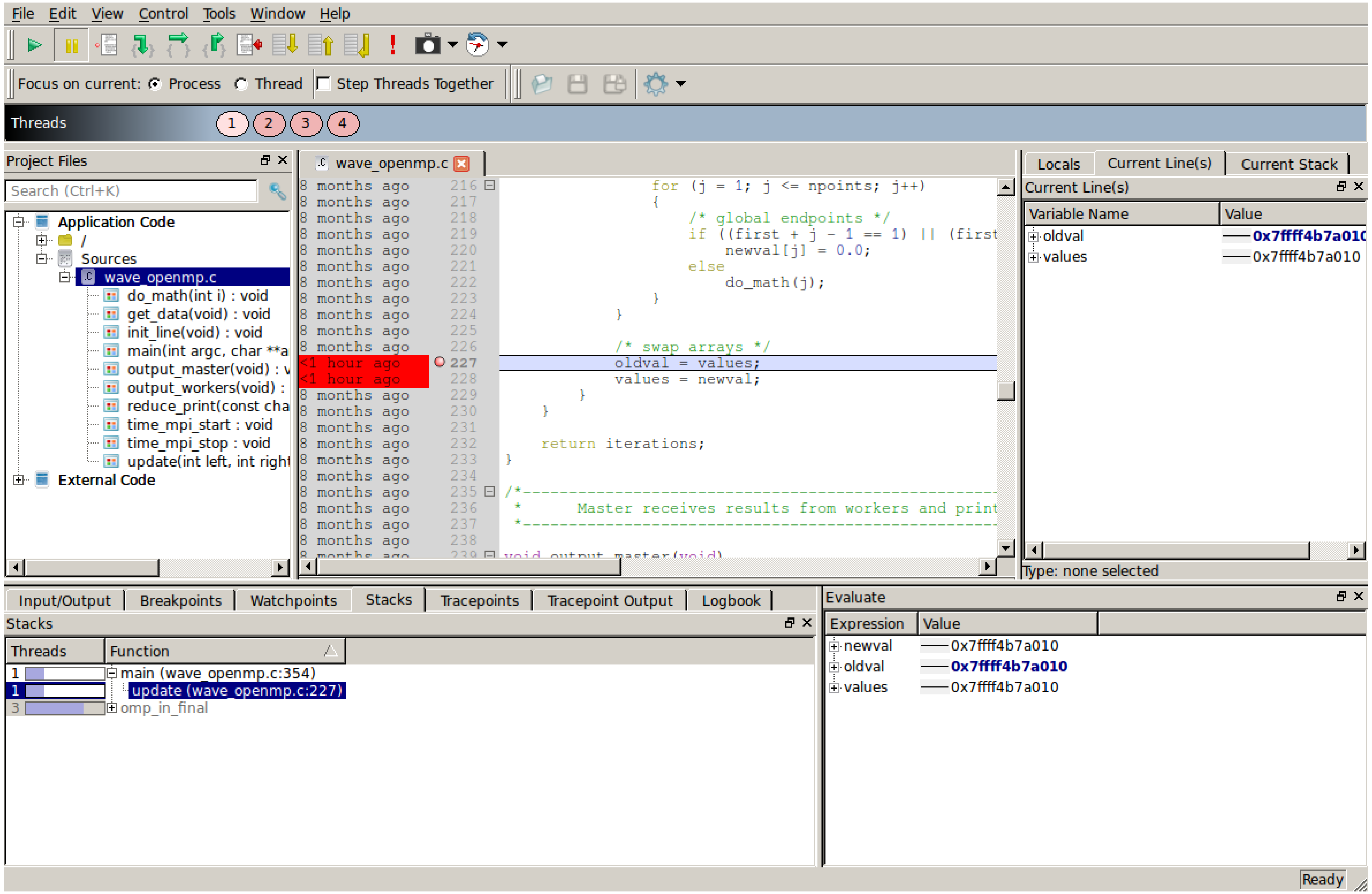
Threads	Function
1	main (wave_openmp.c:354)
1	update (wave_openmp.c:227)
3	omp_in_final

Evaluate

Expression	Value
newval	0x7ffff4b7a010
oldval	**0x7ffff4b7a010**
values	0x7ffff4b7a010

allinea DDT

These arrays are all pointing to the same area of memory!



```
            for (j = 1; j <= npoints; j++)
            {
                /* global endpoints */
                if ((first + j - 1 == 1) || (first
                    newval[j] = 0.0;
                else
                    do_math(j);
            }
        }

        /* swap arrays */
        oldval = values;
        values = newval;
    }
}

return iterations;
}

/*------------------------------------------------------
 *      Master receives results from workers and print
 *------------------------------------------------------

void output_master(void)
```

Current Line(s)

Variable Name	Value
oldval	0x7ffff4b7a010
values	0x7ffff4b7a010

Type: none selected

Stacks

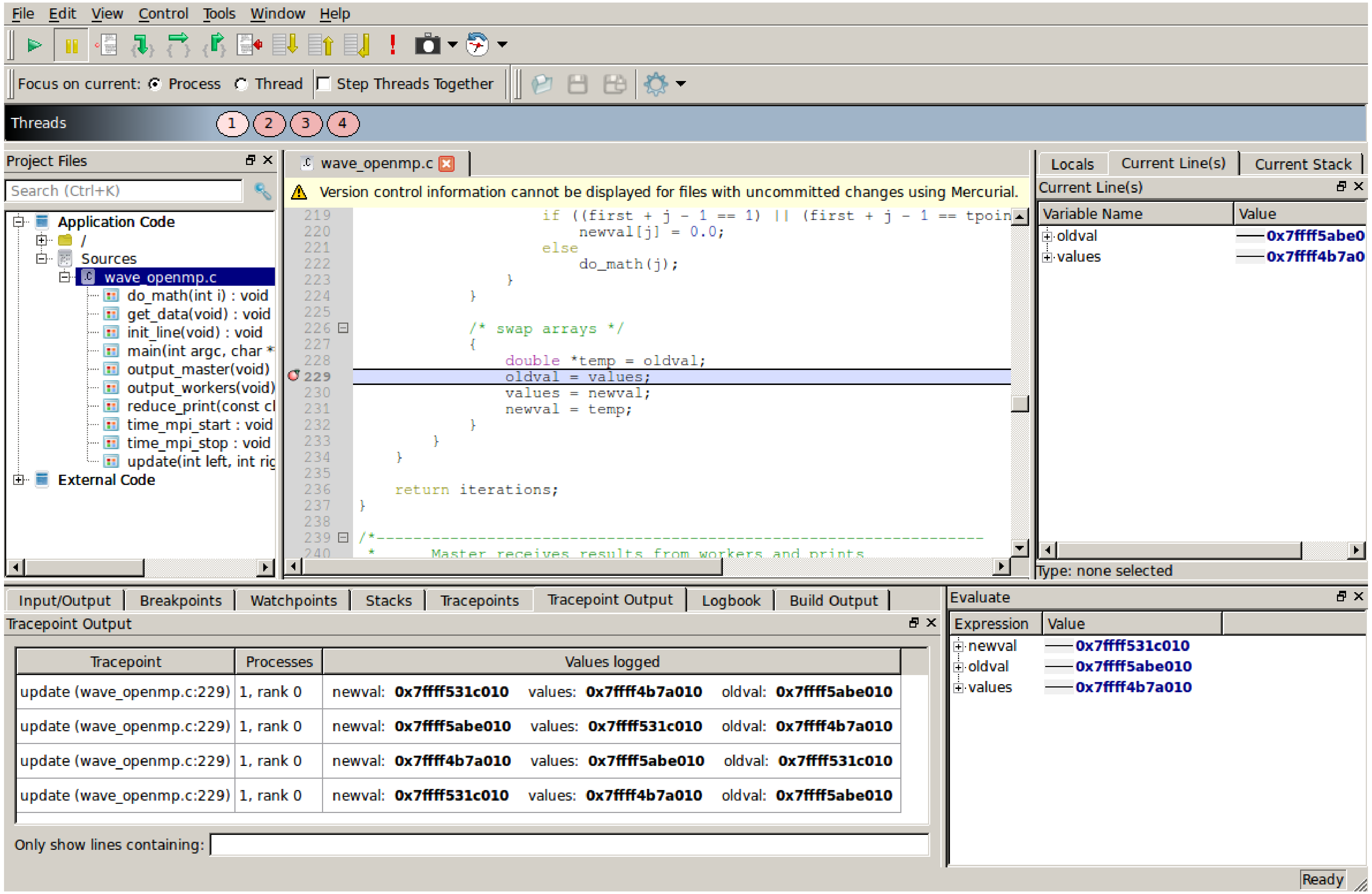
Threads	Function
1	main (wave_openmp.c:354)
1	update (wave_openmp.c:227)
3	omp_in_final

Evaluate

Expression	Value
newval	0x7ffff4b7a010
oldval	0x7ffff4b7a010
values	0x7ffff4b7a010

allinea DDT

Verify our fix before committing it



Version control information cannot be displayed for files with uncommitted changes using Mercurial.

```
219                     if ((first + j - 1 == 1) || (first + j - 1 == tpoin
220                         newval[j] = 0.0;
221                     else
222                         do_math(j);
223                 }
224             }
225 
226             /* swap arrays */
227             {
228                 double *temp = oldval;
229                 oldval = values;
230                 values = newval;
231                 newval = temp;
232             }
233         }
234     }
235 
236     return iterations;
237 }
238 
239 /*---------------------------------------------------------------------
240  *       Master receives results from workers and prints
```

Current Line(s)

Variable Name	Value
oldval	0x7ffff5abe0
values	0x7ffff4b7a0

Tracepoint Output

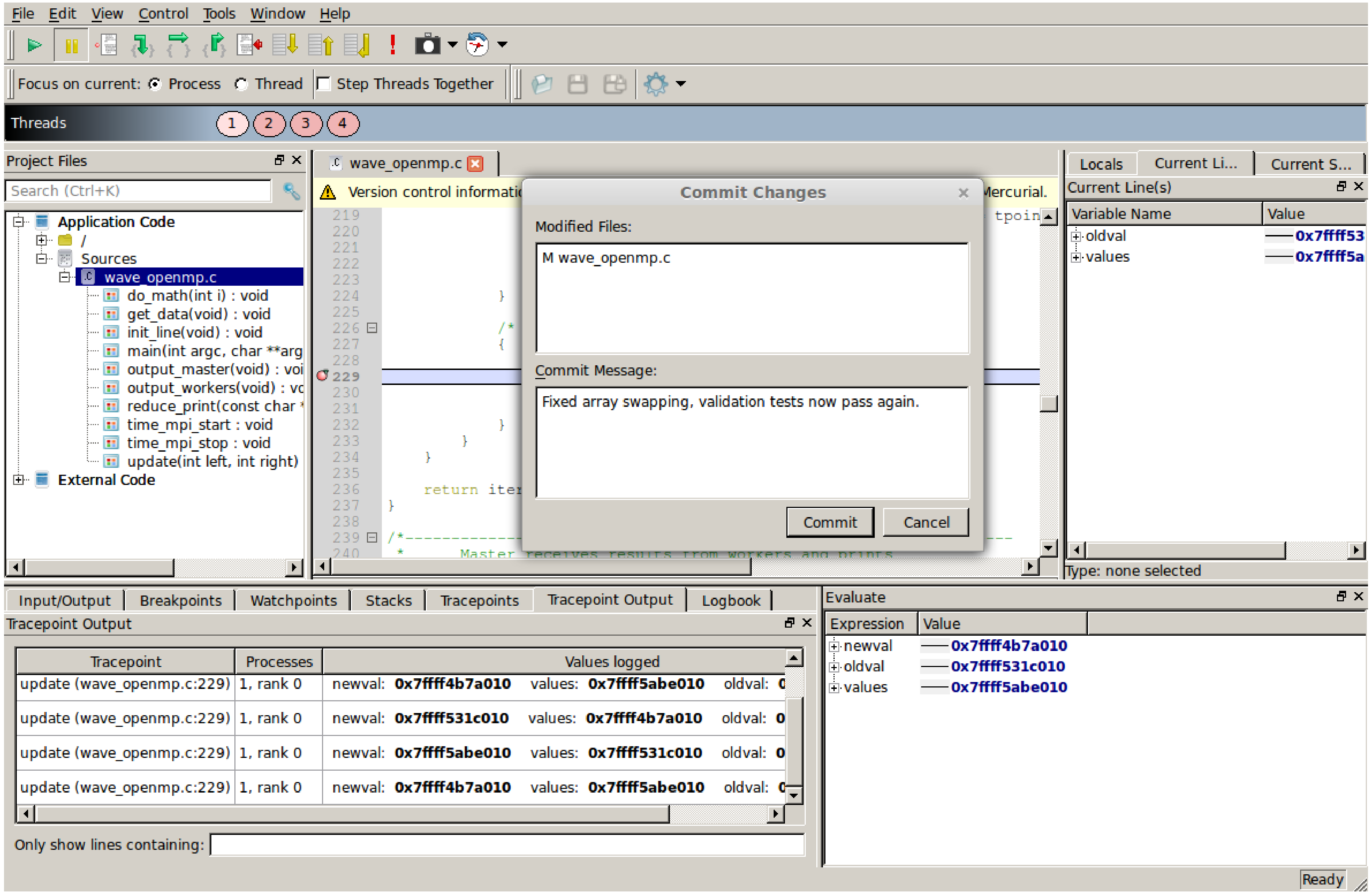
Tracepoint	Processes	Values logged
update (wave_openmp.c:229)	1, rank 0	newval: **0x7ffff531c010** values: **0x7ffff4b7a010** oldval: **0x7ffff5abe010**
update (wave_openmp.c:229)	1, rank 0	newval: **0x7ffff5abe010** values: **0x7ffff531c010** oldval: **0x7ffff4b7a010**
update (wave_openmp.c:229)	1, rank 0	newval: **0x7ffff4b7a010** values: **0x7ffff5abe010** oldval: **0x7ffff531c010**
update (wave_openmp.c:229)	1, rank 0	newval: **0x7ffff531c010** values: **0x7ffff4b7a010** oldval: **0x7ffff5abe010**

Evaluate

Expression	Value
newval	0x7ffff531c010
oldval	0x7ffff5abe010
values	0x7ffff4b7a010

allinea DDT

A tracepoint shows the arrays pointers are swapping correctly now



Focus on current: (●) Process ( ) Thread [ ] Step Threads Together

Threads 1 2 3 4

Project Files

- Application Code
  - /
  - Sources
    - wave_openmp.c
      - do_math(int i) : void
      - get_data(void) : void
      - init_line(void) : void
      - main(int argc, char **arg
      - output_master(void) : voi
      - output_workers(void) : vc
      - reduce_print(const char *
      - time_mpi_start : void
      - time_mpi_stop : void
      - update(int left, int right)
- External Code

wave_openmp.c

Version control informati ... Mercurial.

Commit Changes

Modified Files:

```
M wave_openmp.c
```

Commit Message:

```
Fixed array swapping, validation tests now pass again.
```

Commit | Cancel

Current Line(s)

Variable Name	Value
oldval	0x7ffff53
values	0x7ffff5a

Type: none selected

Tracepoint Output

Tracepoint	Processes	Values logged
update (wave_openmp.c:229)	1, rank 0	newval: **0x7ffff4b7a010** values: **0x7ffff5abe010** oldval: 0
update (wave_openmp.c:229)	1, rank 0	newval: **0x7ffff531c010** values: **0x7ffff4b7a010** oldval: 0
update (wave_openmp.c:229)	1, rank 0	newval: **0x7ffff5abe010** values: **0x7ffff531c010** oldval: 0
update (wave_openmp.c:229)	1, rank 0	newval: **0x7ffff4b7a010** values: **0x7ffff5abe010** oldval: 0

Only show lines containing:

Evaluate

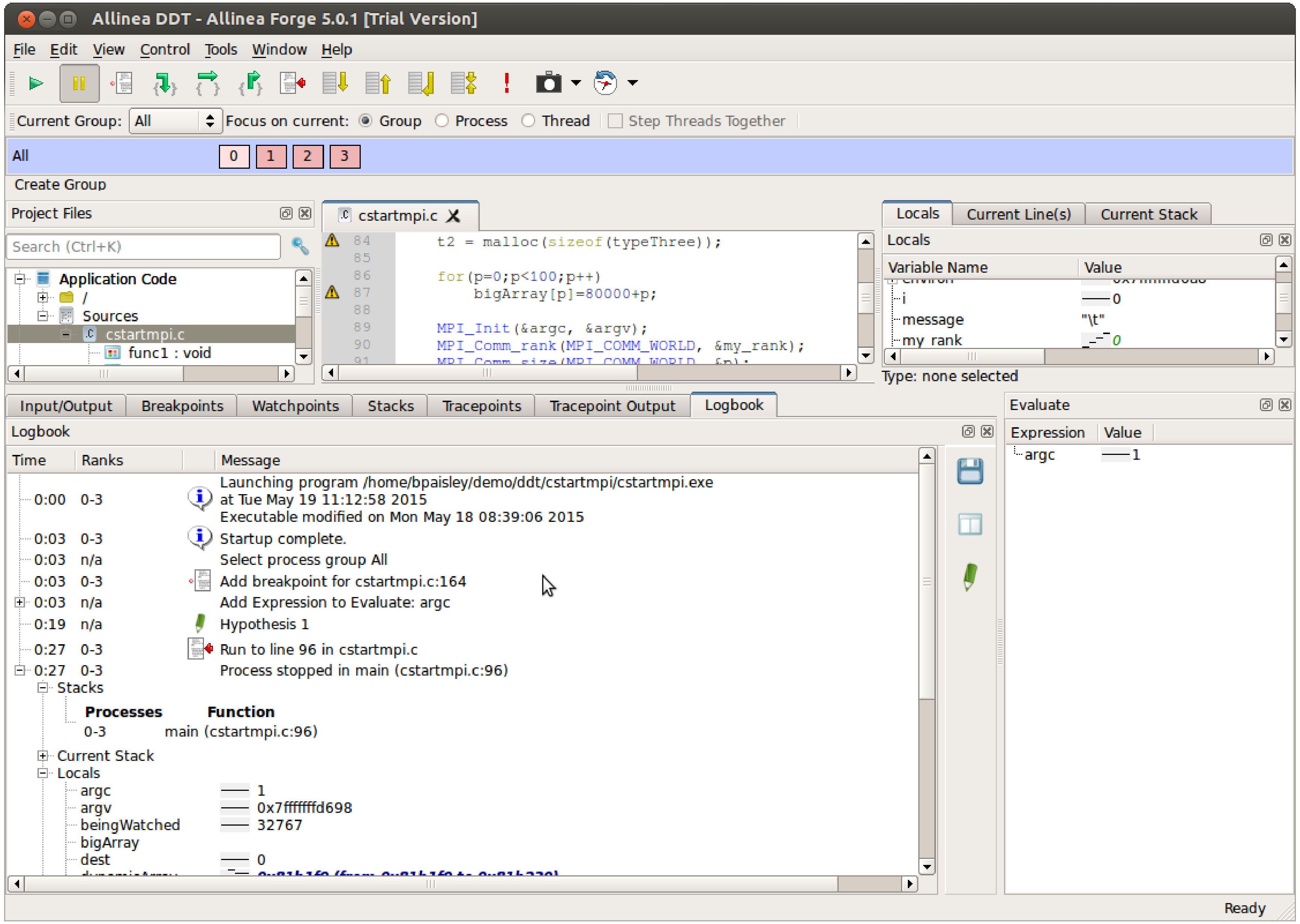
Expression	Value
newval	0x7ffff4b7a010
oldval	0x7ffff531c010
values	0x7ffff5abe010

Ready

File Edit View Control Tools Window Help
Input/Output Breakpoints Watchpoints Stacks Tracepoints Tracepoint Output Logbook
Locals Current Li... Current S...

allinea DDT

Debug with the Scientific Method


Allinea DDT - Allinea Forge 5.0.1 [Trial Version]
File Edit View Control Tools Window Help
Current Group: All
Focus on current: Group Process Thread
Step Threads Together
All 0 1 2 3
Create Group
Project Files
Search (Ctrl+K)
Application Code
/
Sources
cstartmpi.c
func1 : void
cstartmpi.c
Locals Current Line(s) Current Stack
Input/Output Breakpoints Watchpoints Stacks Tracepoints Tracepoint Output Logbook
Evaluate
Ready

```
    t2 = malloc(sizeof(typeThree));

    for(p=0;p<100;p++)
        bigArray[p]=80000+p;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
```

Locals

Variable Name	Value
i	0
message	"\t"
my_rank	0

Type: none selected

Evaluate

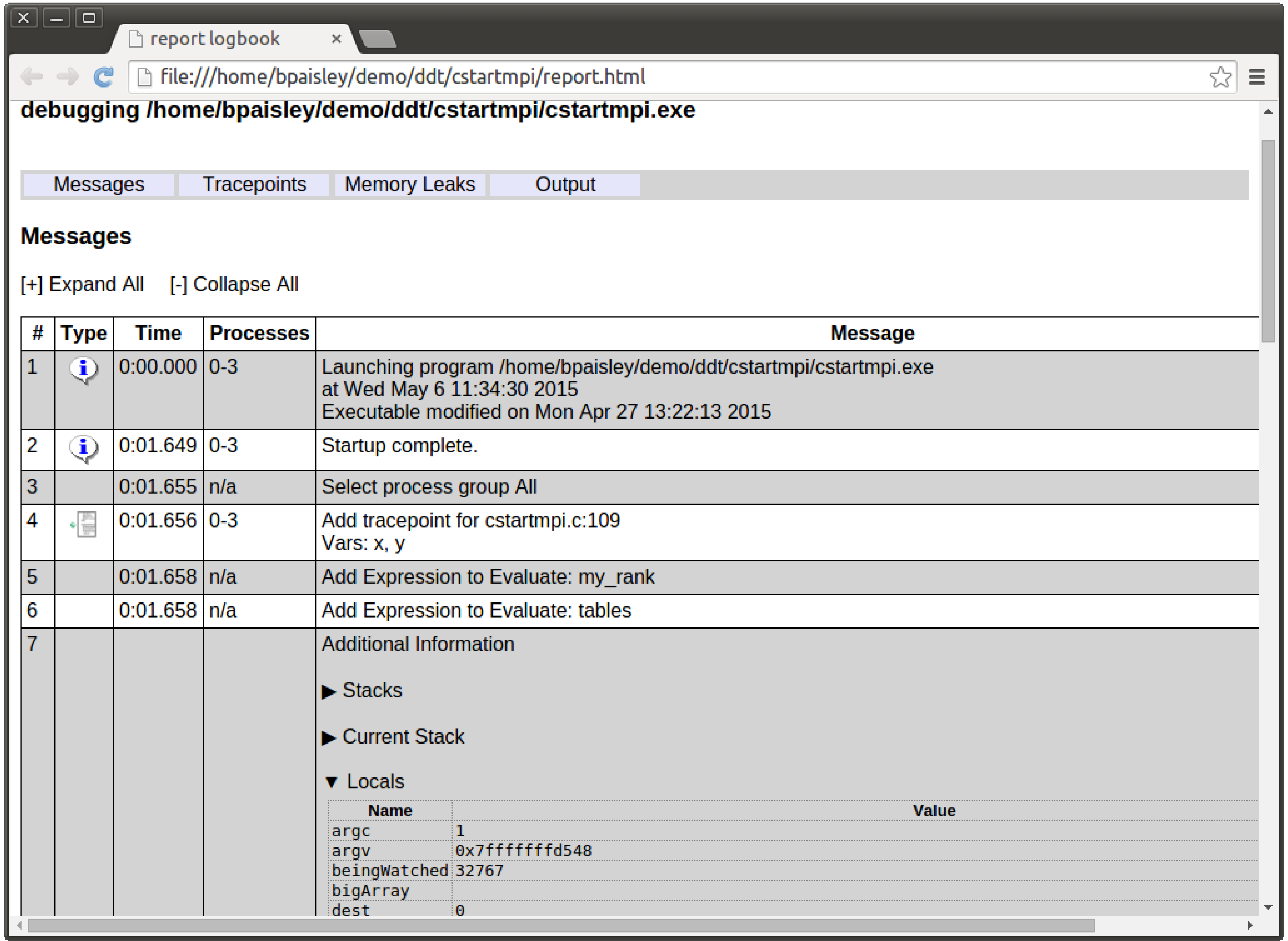
Expression	Value
argc	1

Logbook

```
Time   Ranks   Message
0:00   0-3     Launching program /home/bpaisley/demo/ddt/cstartmpi/cstartmpi.exe
               at Tue May 19 11:12:58 2015
               Executable modified on Mon May 18 08:39:06 2015
0:03   0-3     Startup complete.
0:03   n/a     Select process group All
0:03   0-3     Add breakpoint for cstartmpi.c:164
0:03   n/a     Add Expression to Evaluate: argc
0:19   n/a     Hypothesis 1
0:27   0-3     Run to line 96 in cstartmpi.c
0:27   0-3     Process stopped in main (cstartmpi.c:96)
  Stacks
     Processes     Function
     0-3        main (cstartmpi.c:96)
  Current Stack
  Locals
     argc           1
     argv           0x7fffffffd698
     beingWatched   32767
     bigArray
     dest           0
```

allinea DDT

Debugging While you Sleep



report logbook

file:///home/bpaisley/demo/ddt/cstartmpi/report.html

debugging /home/bpaisley/demo/ddt/cstartmpi/cstartmpi.exe

Messages

[+] Expand All [-] Collapse All

#	Type	Time	Processes	Message
1	ⓘ	0:00.000	0-3	Launching program /home/bpaisley/demo/ddt/cstartmpi/cstartmpi.exe at Wed May 6 11:34:30 2015 Executable modified on Mon Apr 27 13:22:13 2015
2	ⓘ	0:01.649	0-3	Startup complete.
3		0:01.655	n/a	Select process group All
4		0:01.656	0-3	Add tracepoint for cstartmpi.c:109 Vars: x, y
5		0:01.658	n/a	Add Expression to Evaluate: my_rank
6		0:01.658	n/a	Add Expression to Evaluate: tables
7				Additional Information ▶ Stacks ▶ Current Stack ▼ Locals

Name	Value
argc	1
argv	0x7fffffffd548
beingWatched	32767
bigArray	
dest	0

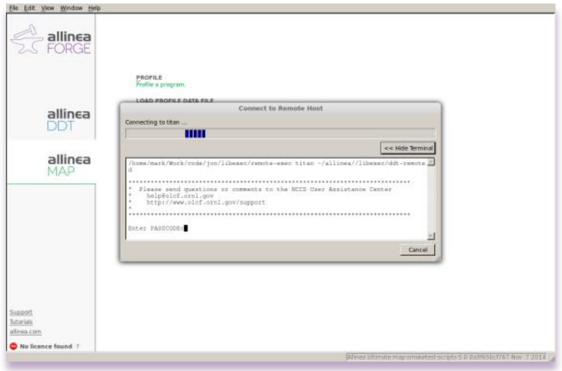
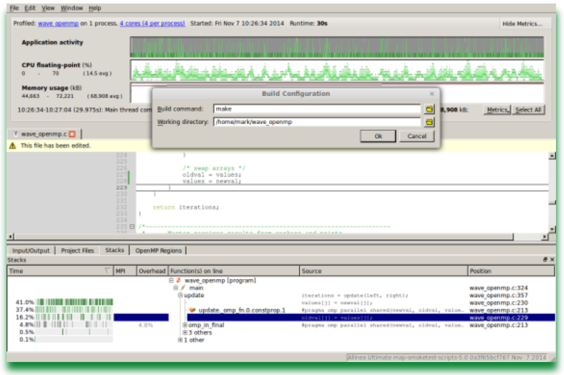
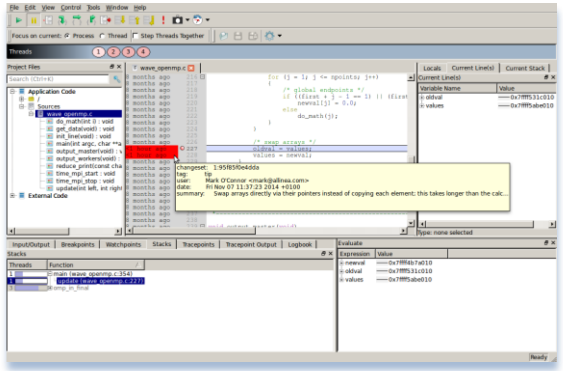
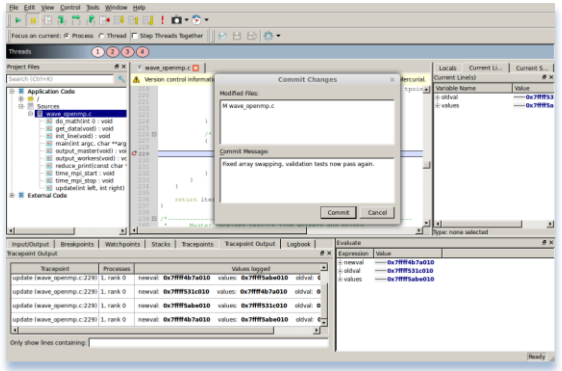
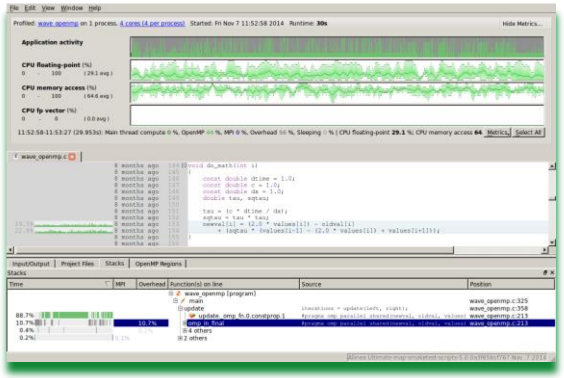
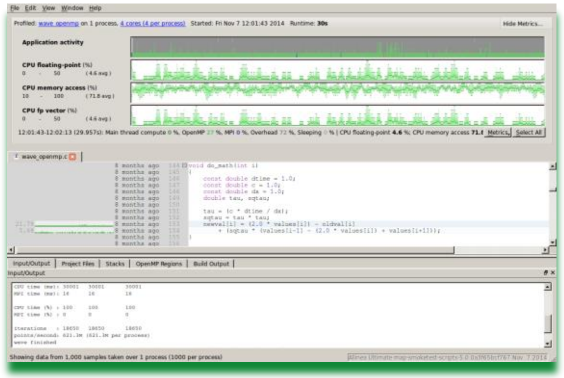
allinea DDT

A Productive HPC Development Workflow

allinea FORGE

allinea DDT

allinea MAP

allinea

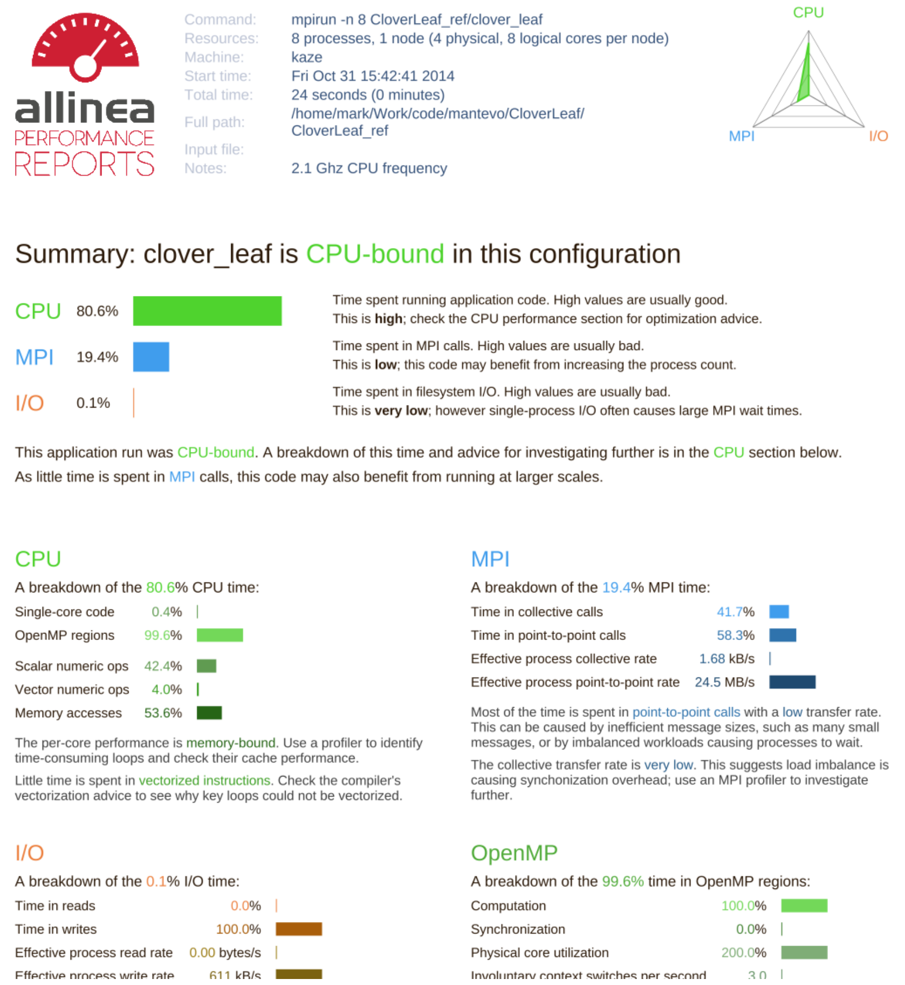
Analyze and tune application performance

A single-page report on application performance for users and administrators

Identify configuration problems and resource bottlenecks immediately

Track mission-critical performance over time and after system upgrades

Ensure key applications run at full speed on a new cluster or architecture



Command:	mpirun -n 8 CloverLeaf_ref/clover_leaf
Resources:	8 processes, 1 node (4 physical, 8 logical cores per node)
Machine:	kaze
Start time:	Fri Oct 31 15:42:41 2014
Total time:	24 seconds (0 minutes)
Full path:	/home/mark/Work/code/mantevo/CloverLeaf/ CloverLeaf_ref
Input file:	
Notes:	2.1 Ghz CPU frequency

Summary: clover_leaf is CPU-bound in this configuration

CPU	80.6%	Time spent running application code. High values are usually good. This is **high**; check the CPU performance section for optimization advice.
MPI	19.4%	Time spent in MPI calls. High values are usually bad. This is **low**; this code may benefit from increasing the process count.
I/O	0.1%	Time spent in filesystem I/O. High values are usually bad. This is **very low**; however single-process I/O often causes large MPI wait times.

This application run was CPU-bound. A breakdown of this time and advice for investigating further is in the CPU section below.
As little time is spent in MPI calls, this code may also benefit from running at larger scales.

CPU

A breakdown of the 80.6% CPU time:

Single-core code	0.4%
OpenMP regions	99.6%
Scalar numeric ops	42.4%
Vector numeric ops	4.0%
Memory accesses	53.6%

The per-core performance is memory-bound. Use a profiler to identify time-consuming loops and check their cache performance.

Little time is spent in vectorized instructions. Check the compiler's vectorization advice to see why key loops could not be vectorized.

MPI

A breakdown of the 19.4% MPI time:

Time in collective calls	41.7%
Time in point-to-point calls	58.3%
Effective process collective rate	1.68 kB/s
Effective process point-to-point rate	24.5 MB/s

Most of the time is spent in point-to-point calls with a low transfer rate. This can be caused by inefficient message sizes, such as many small messages, or by imbalanced workloads causing processes to wait.

The collective transfer rate is very low. This suggests load imbalance is causing synchonization overhead; use an MPI profiler to investigate further.

I/O

A breakdown of the 0.1% I/O time:

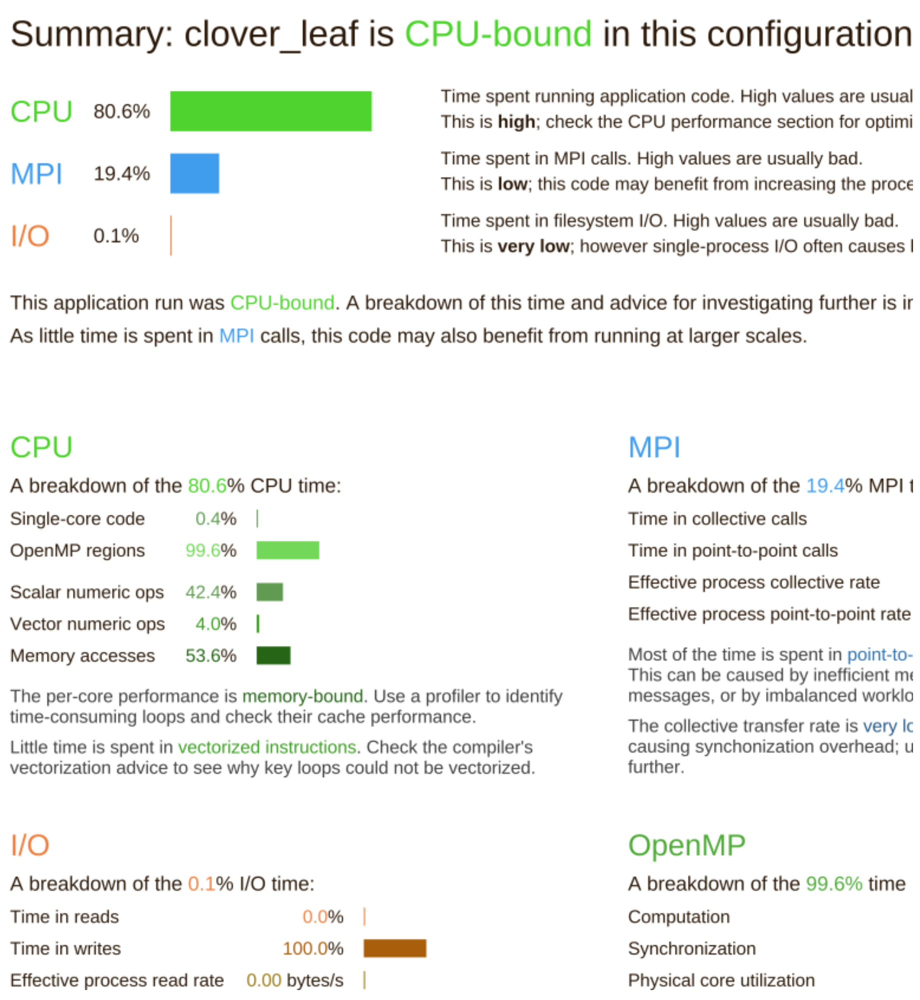
Time in reads	0.0%
Time in writes	100.0%
Effective process read rate	0.00 bytes/s
Effective process write rate	611 kB/s

OpenMP

A breakdown of the 99.6% time in OpenMP regions:

Computation	100.0%
Synchronization	0.0%
Physical core utilization	200.0%
Involuntary context switches per second	3.0

A single-page report for users and administrators



Summary: clover_leaf is CPU-bound in this configuration

CPU	80.6%	Time spent running application code. High values are usual This is **high**; check the CPU performance section for optimi
MPI	19.4%	Time spent in MPI calls. High values are usually bad. This is **low**; this code may benefit from increasing the proce
I/O	0.1%	Time spent in filesystem I/O. High values are usually bad. This is **very low**; however single-process I/O often causes l

This application run was CPU-bound. A breakdown of this time and advice for investigating further is ir

As little time is spent in MPI calls, this code may also benefit from running at larger scales.

CPU

A breakdown of the 80.6% CPU time:

Single-core code	0.4%
OpenMP regions	99.6%
Scalar numeric ops	42.4%
Vector numeric ops	4.0%
Memory accesses	53.6%

The per-core performance is memory-bound. Use a profiler to identify time-consuming loops and check their cache performance.

Little time is spent in vectorized instructions. Check the compiler's vectorization advice to see why key loops could not be vectorized.

MPI

A breakdown of the 19.4% MPI t

Time in collective calls

Time in point-to-point calls

Effective process collective rate

Effective process point-to-point rate

Most of the time is spent in point-to-
This can be caused by inefficient me
messages, or by imbalanced worklo

The collective transfer rate is very lo
causing synchonization overhead; u
further.

I/O

A breakdown of the 0.1% I/O time:

Time in reads	0.0%
Time in writes	100.0%
Effective process read rate	0.00 bytes/s

OpenMP

A breakdown of the 99.6% time

Computation

Synchronization

Physical core utilization

No recompilation or instrumentation necessary

Less than 5% application slowdown on most systems

Summarizes performance of individual application runs

Save data in HTML, TXT or CSV formats for analysis

allinea

High performance tools to debug, profile, and analyze your applications

https://www.allinea.com/products/downloads/free-trial

allinea FORGE

allinea DDT

allinea MAP

allinea PERFORMANCE REPORTS