allinea

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

High-productivity development tools for science

Beau Paisley

bpaisley@allinea.com

allinea FORGE

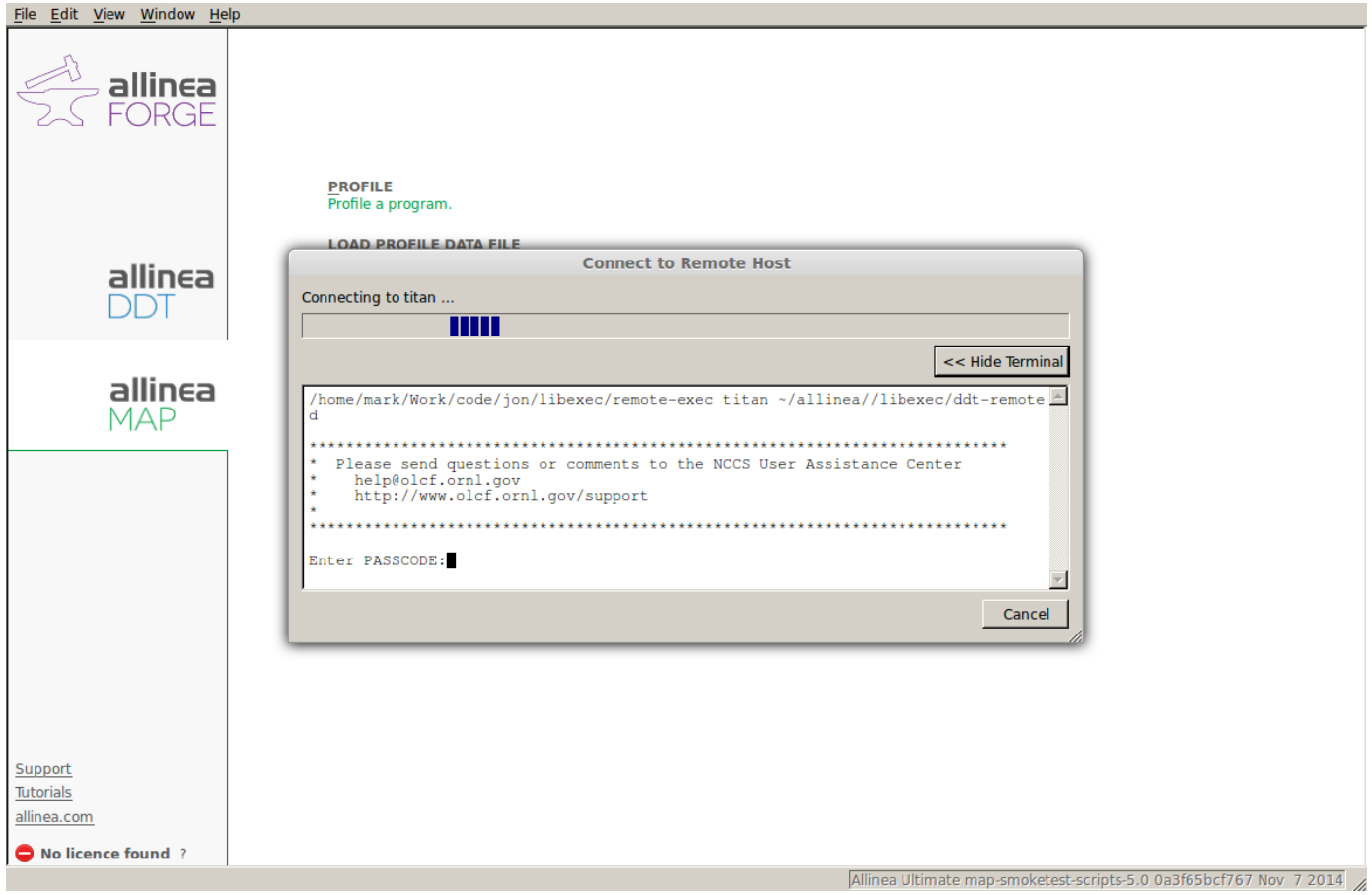
allinea DDT

allinea MAP

allinea PERFORMANCE REPORTS

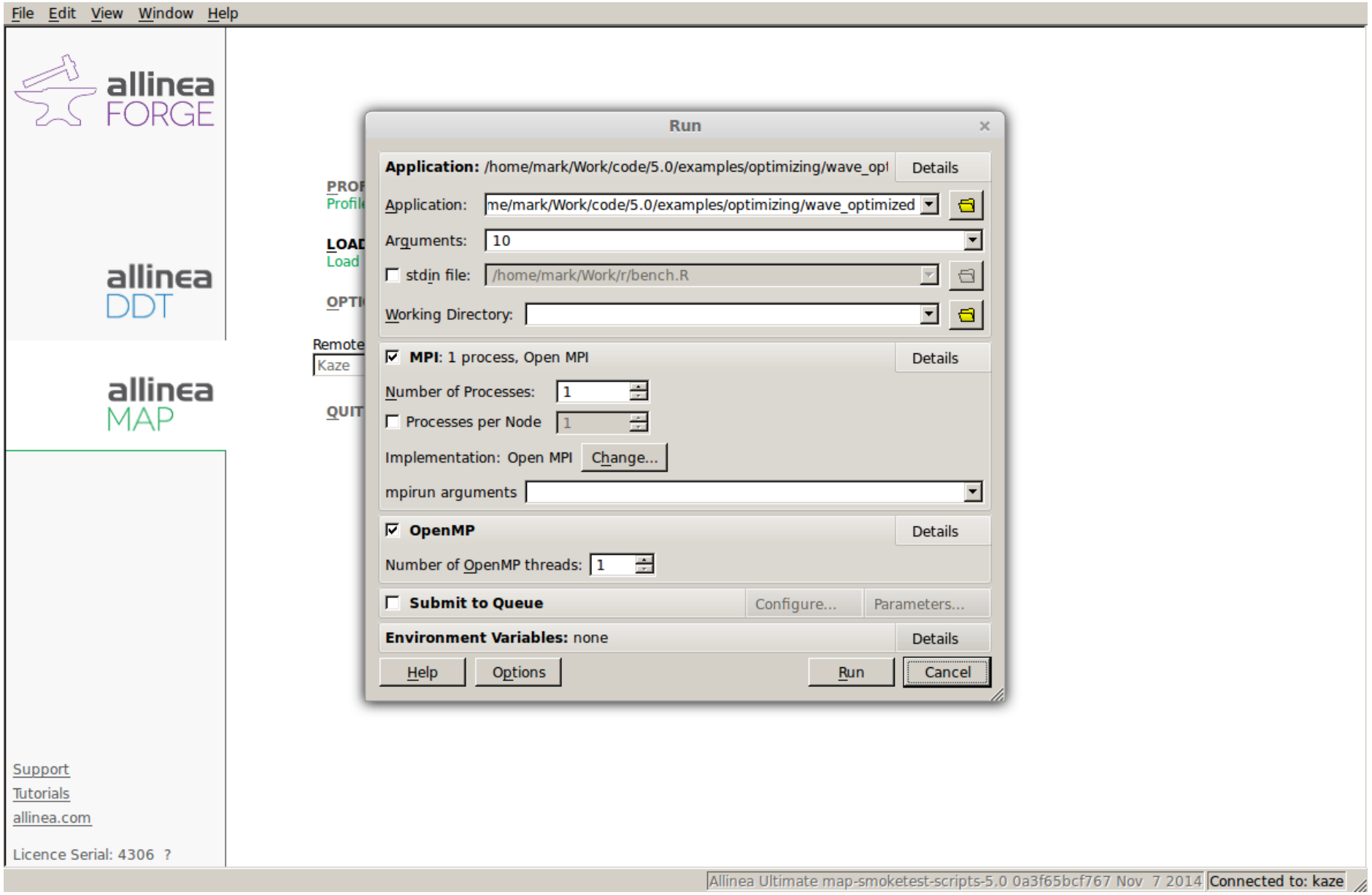
HPC means being able to work productively on remote machines

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



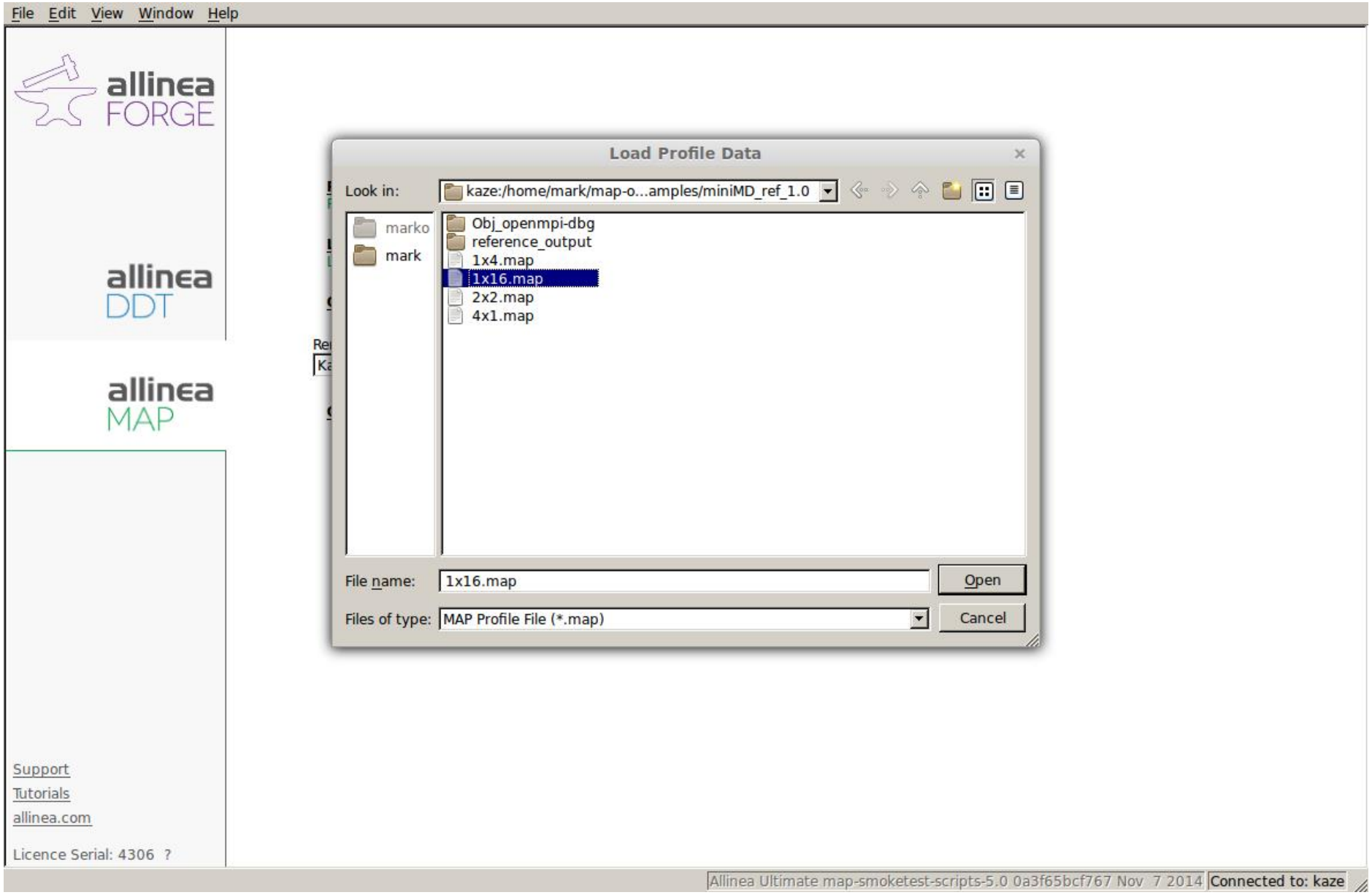
allinea FORGE

Submit to job queues or run interactively on any system



allinea FORGE

Open and view files directly on the remote system



We've been asked to look at a profile recorded for a wave equation solver

File Edit View Window Help

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

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; Metrics Select All

wave_openmp.c

```
223                 }
224             }
225
226             /* swap arrays */
227             for (j = 1; j <= npoints; j++)
228             {
229                 oldval[j] = values[j];
230                 values[j] = newval[j];
231             }
232         }
233     }
234
235     return iterations;
236 }
```

41.0%
16.2%

Input/Output | Project Files | Stacks | OpenMP Regions

Stacks

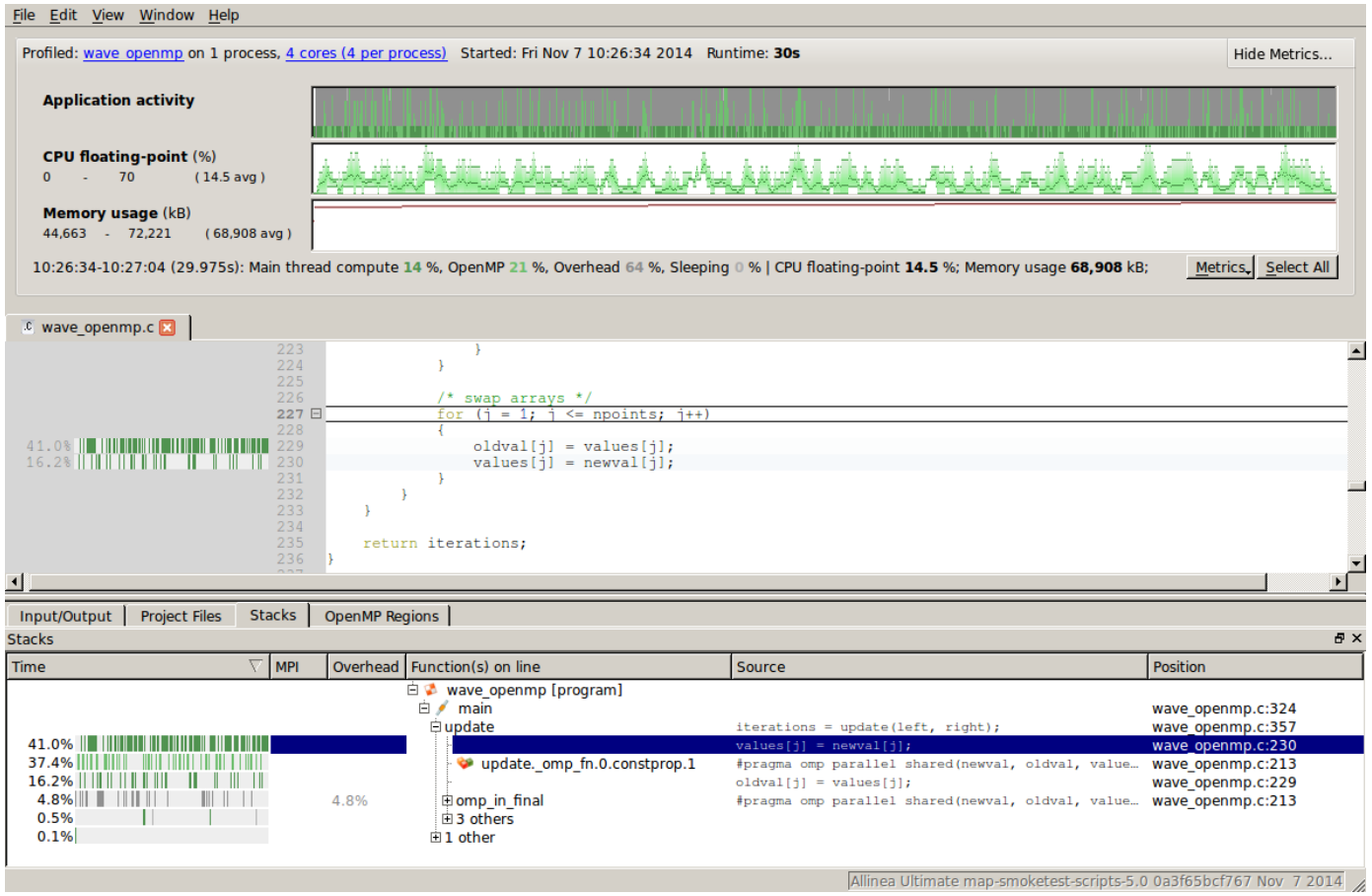
Time	MPI	Overhead	Function(s) on line	Source	Position
			wave_openmp [program]		
			main		wave_openmp.c:324
			update	iterations = update(left, right);	wave_openmp.c:357
41.0%				values[j] = newval[j];	wave_openmp.c:230
37.4%			update._omp_fn.0.constprop.1	#pragma omp parallel shared(newval, oldval, 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

We go the extra mile to make products that just work

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



allinea MAP

57.2% of the time is spent swapping arrays – this isn't even in an OpenMP loop

File Edit View Window Help

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

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; Metrics Select All

wave_openmp.c

```
223                 }
224             }
225
226             /* swap arrays */
227             for (j = 1; j <= npoints; j++)
228             {
229                 oldval[j] = values[j];
230                 values[j] = newval[j];
231             }
232         }
233     }
234
235     return iterations;
236 }
```

41.0%
16.2%

Input/Output | Project Files | Stacks | OpenMP Regions

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

There’s a much quicker way to swap arrays than by copying their data

File Edit View Window Help

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

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; Metrics Select All

wave_openmp.c*

This file has been edited.

```
223                 }
224             }
225
226             /* swap arrays */
227             oldval = values;
228             values = newva
229
230             /*for (j = 1; j <= npoints; j++)
231             {
232                 oldval[j] = values[j];
233                 values[j] = newval[j];
234             }*/
```

41.0%
16.2%

Input/Output | Project Files | Stacks | OpenMP Regions

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

We can build this change on the remote system directly

File Edit View Window Help

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

Hide Metrics...

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 com

Build Configuration

Build command: make

Working directory: /home/mark/wave_openmp

Ok Cancel

8,908 kB; Metrics Select All

wave_openmp.c

This file has been edited.

```
224             }
225
226             /* swap arrays */
227             oldval = values;
228             values = newval;
229         }
230     }
231
232     return iterations;
233 }
234
235 /*-----------------------------------------------------------------
```

Input/Output | Project Files | Stacks | OpenMP Regions

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
				values[j] = newval[j];	wave_openmp.c:230
41.0%			update._omp_fn.0.constprop.1	#pragma omp parallel shared(newval, oldval, value…	wave_openmp.c:213
37.4%					
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

We can build this change on the remote system directly

File Edit View Window Help

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

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; Metrics Select All

wave_openmp.c

This file has been edited.

```
224                 }
225
226                 /* swap arrays */
227                 oldval = values;
228                 values = newval;
229             }
230         }
231
232         return iterations;
233     }
234
235     /*-----------------------------------------------------------------
```

Input/Output | Project Files | Stacks | OpenMP Regions | Build Output

Build Output

```
10:41:38 - Running 'make' in '/home/mark/wave_openmp'
mpicc -g -O3 -fopenmp wave_openmp.c -o wave_openmp -lm

10:41:38 - Finished 'make' in '/home/mark/wave_openmp' - build time: 0 seconds
```

10:41:38 - Finished 'make' in '/home/mark/wave_openmp' - build time: 0 seconds

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

allinea MAP

And commit it to source control

File Edit View Window Help

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

Hide Metrics...

Application activity

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

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

10:26:34-10:27:03 (29.975s): Main thread compute **14** %, OpenMP **21** %, Overhead **64** %, Sleeping **0** % | CPU floating-point **14.5** %; Memory usage **68,908** kB;

Metrics Select All

wave_openmp.c

This file has been edited.

```
225
226             /* swap arrays */
227             oldval = values;
228             values = newval;
229         }
230     }
231
232     return iterations;
233 }
234
```

Commit Changes

Modified Files:

M wave_openmp.c

Commit Message:

Swap arrays directly via their pointers instead of copying each element; this takes longer than the calculation does with 4 cores...

Commit Cancel

Input/Output | Project Files | Stacks | OpenMP Regions | Build Output

Build Output

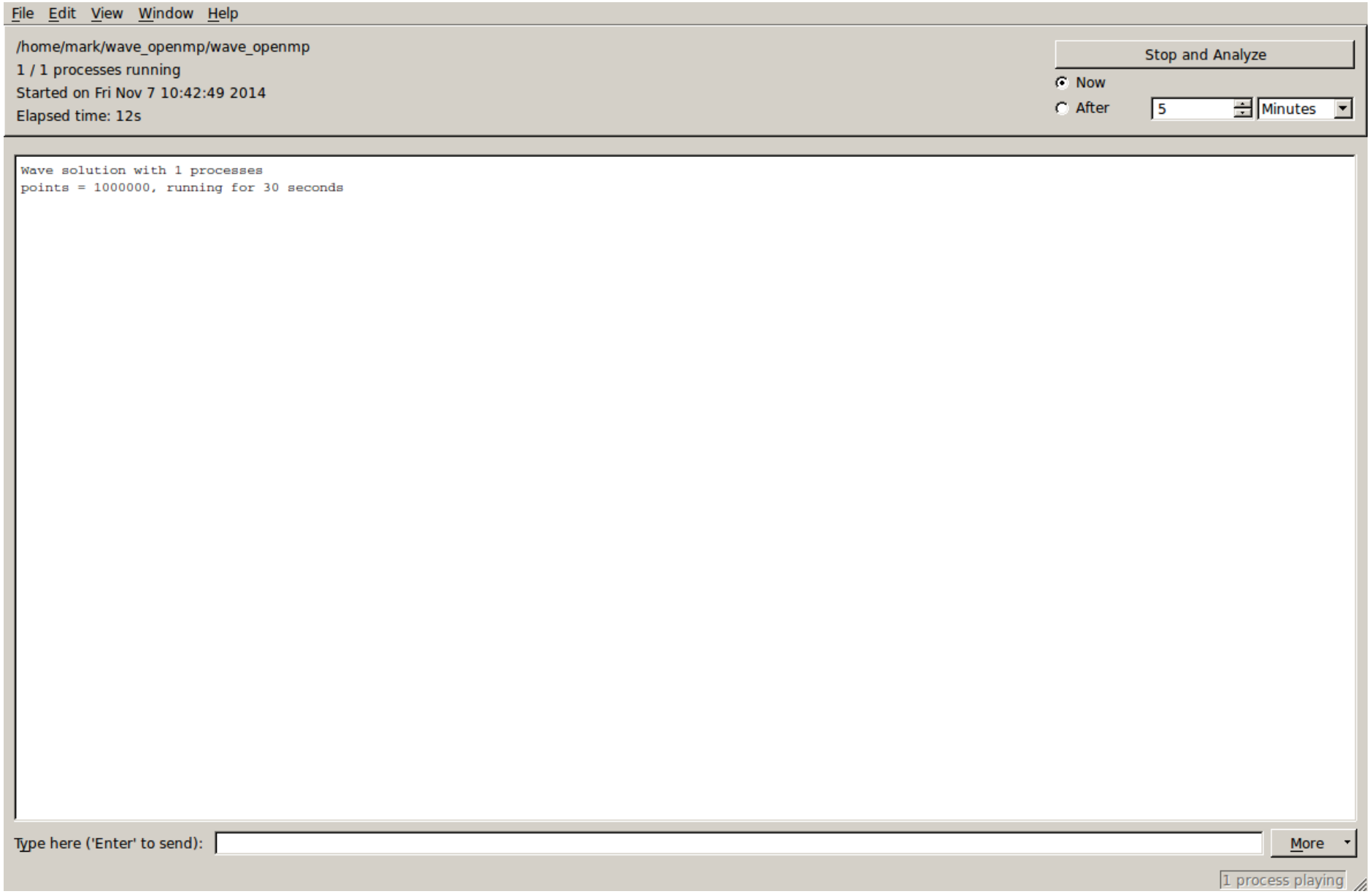
```
11:28:46 - Running 'make' in '/home/mark/wave_openmp'
mpicc -g -O3 -fopenmp wave_openmp.c -o wave_openmp -lm

11:28:46 - Finished 'make' in '/home/mark/wave_openmp' - build time: 0 seconds
```

11:28:46 - Finished 'make' in '/home/mark/wave_openmp' - build time: 0 seconds

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

allinea MAP

How much faster will the application be now?



allinea MAP

Perhaps we should have tested it before committing that change…

File Edit View Window Help

Profiled: wave_openmp on 1 process, 4 cores (4 per process) Started: Fri Nov 7 10:42:49 2014 Runtime: **30s** Hide Metrics...

Application activity

CPU floating-point (%)
0 - 100 (45.0 avg)

Memory usage (kB)
49,656 - 72,217 (68,899 avg)

10:42:49-10:43:19 (29.943s): Main thread compute **0** %, OpenMP **68** %, MPI **0** %, Overhead **32** %, Sleeping **0** % | CPU floating-point **45.0** %; Memory usage **68,899** | Metrics Select All

wave_openmp.c

```
<0.1%  348
       349    /* get program parameters and initialize wave values */
       350    get_data();
       351    init_line();
       352
       353    /* update values along the line for nstep time steps */
99.9%  354    iterations = update(left, right);
       355
       356    /*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*/
       357    /* collect results and print */
       358
       359    if (rank == 0) output_master();
       360    else output_workers();
       361
```

Input/Output | Project Files | Stacks | OpenMP Regions

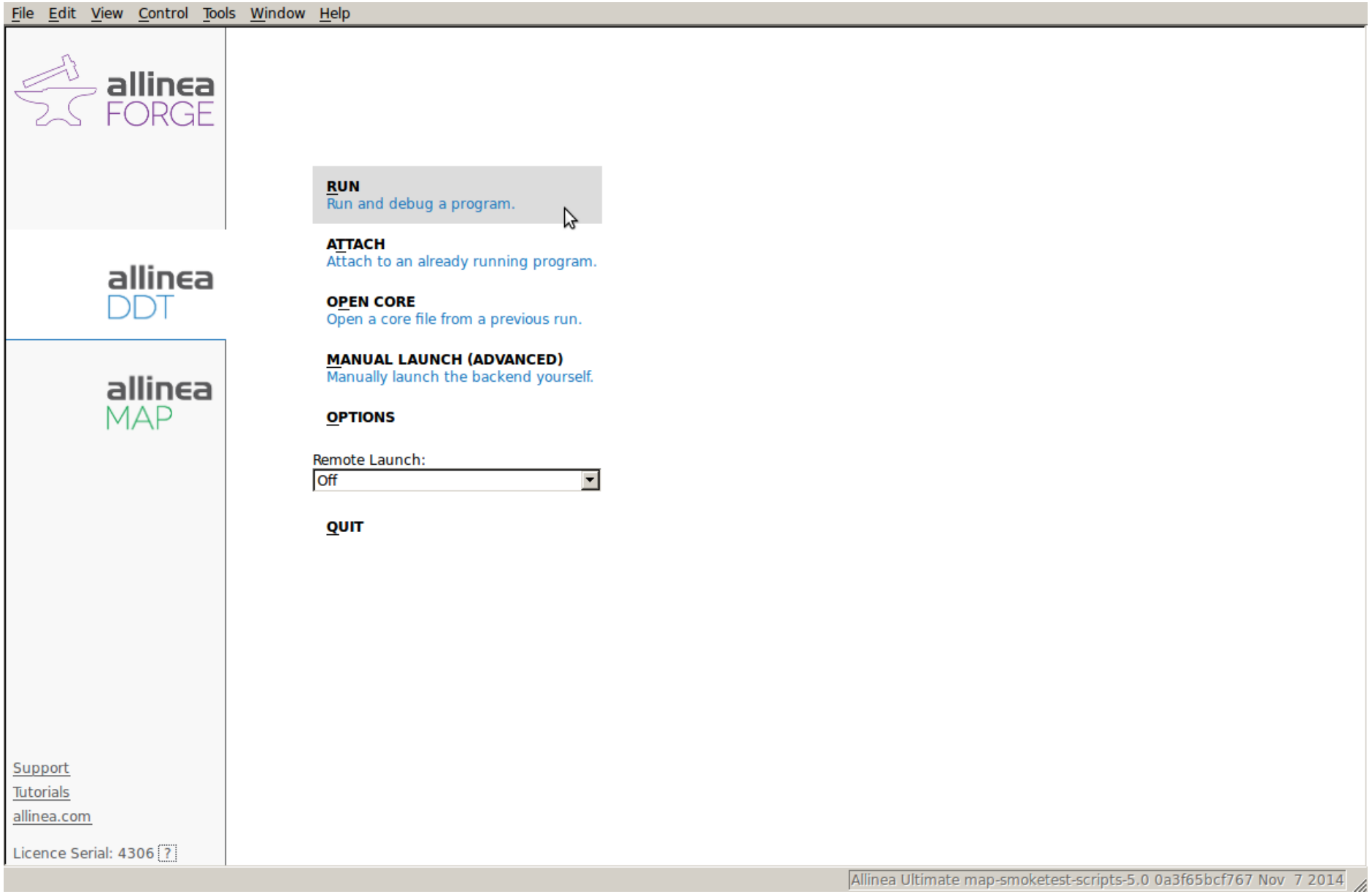
Input/Output

```
Wave solution with 1 processes
points = 1000000, running for 30 seconds

Points for validation:
0:0.00  200000:0.85  400000:0.35  600000:-0.82  800000:-1.03  999999:0.00
0:0.00  200000:0.95  400000:0.59  600000:-0.59  800000:-0.95  999999:0.00
Error, validation check failed!
```

Showing data from 1,000 samples taken over 1 process (1000 per process) Allinea Ultimate map-smoketest-scripts-5.0 0a3f65bcf767 Nov 7 2014

allinea MAP

While still connected to the server we switch to the debugger



It’s already configured to reproduce the profiling run

File Edit View Control Tools Window Help

allinea FORGE

allinea DDT

allinea MAP

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 ?

Run

Application: /home/mark/wave_openmp/wave_openmp — Details

Application: /home/mark/wave_openmp/wave_openmp

Arguments:

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

Working Directory:

MPI: 1 process, Open MPI (Compatibility) — Details

Number of MPC Tasks: 1

Number of Processes: 1

Processes per Node 1

Implementation: Open MPI (Compatibility) Change...

mpirun Arguments:

OpenMP: 4 threads — Details

Number of OpenMP threads: 4

CUDA — Details

Memory Debugging — Details...

Submit to Queue — Configure... Parameters...

Environment Variables: none — Details

Plugins: none — Details

Help Options Run Cancel

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

allinea FORGE

Our tools understand your version control system

File Edit View Control Tools Window Help

Fold all
Unfold all
Increase zoom Ctrl++
Decrease zoom Ctrl+-
Reset zoom Ctrl+0
Show whitespace Alt+.
No split
Horizontal split
Vertical split
Version Control Information

```
            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;
```

Input/Output* | Breakpoints | Watchpoints | Stacks | Tracepoints | Tracepoint Output | Logbook

Stacks

Function
main (wave_openmp.c:329)

Evaluate

Expression	Value
newval	0x0
oldval	0x0
values	0x0

Locals | Current Li... | Current S...

Current Line(s)

Variable Name	Value
start	{tv_sec = 0, tv

Type: none selected

Ready

allinea DDT

Most new bugs are in or around recently changed code

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)

- 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 right
- 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  }
```

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...

Locals | Current Line(s) | Current Stack

Current Line(s)

Variable Name	Value
oldval	0x7ffff531c010
values	0x7ffff5abe010

Type: none selected

Input/Output | Breakpoints | Watchpoints | Stacks | Tracepoints | Tracepoint Output | Logbook

Stacks

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

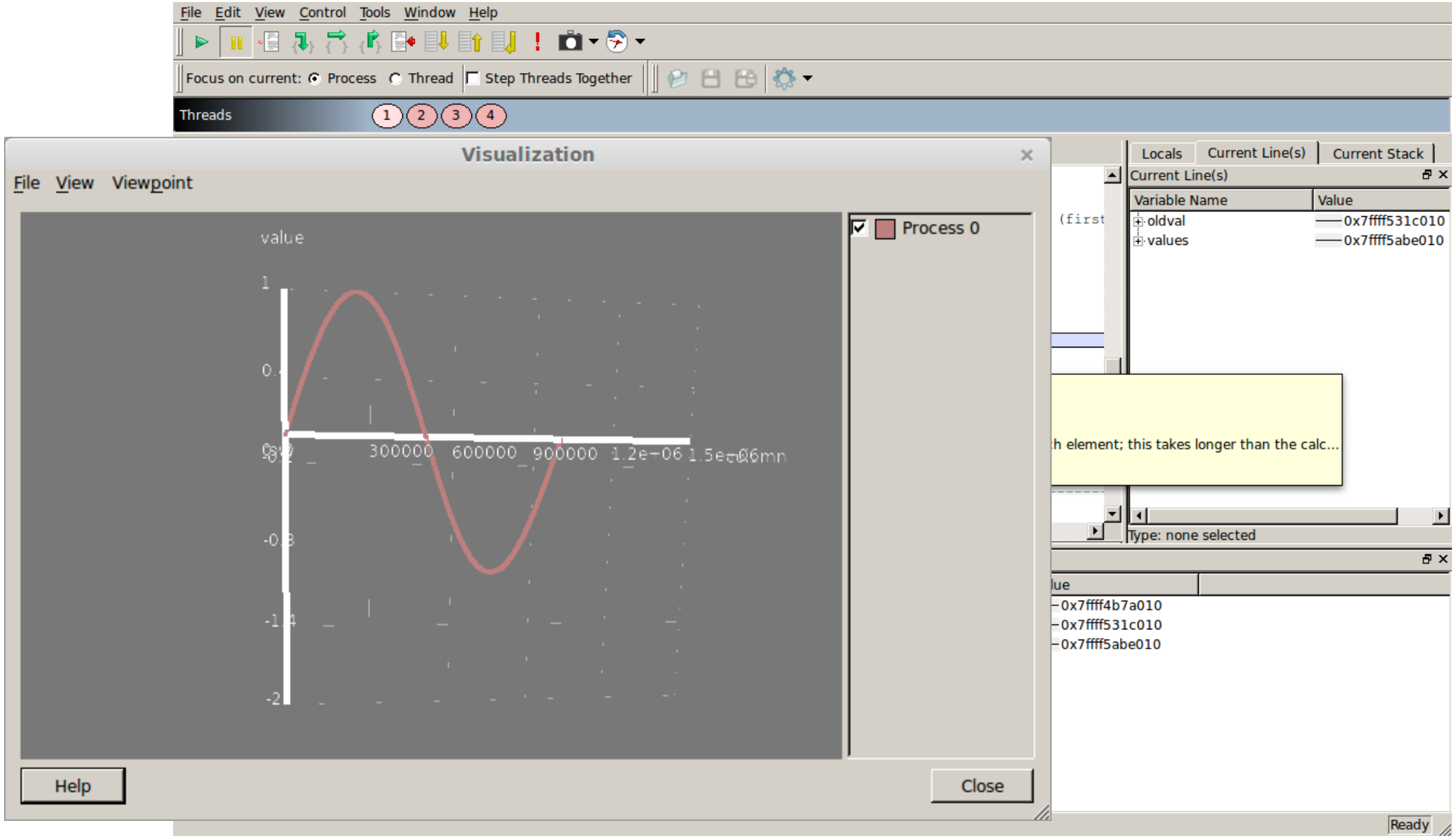
Evaluate

Expression	Value
newval	0x7ffff4b7a010
oldval	0x7ffff531c010
values	0x7ffff5abe010

Ready

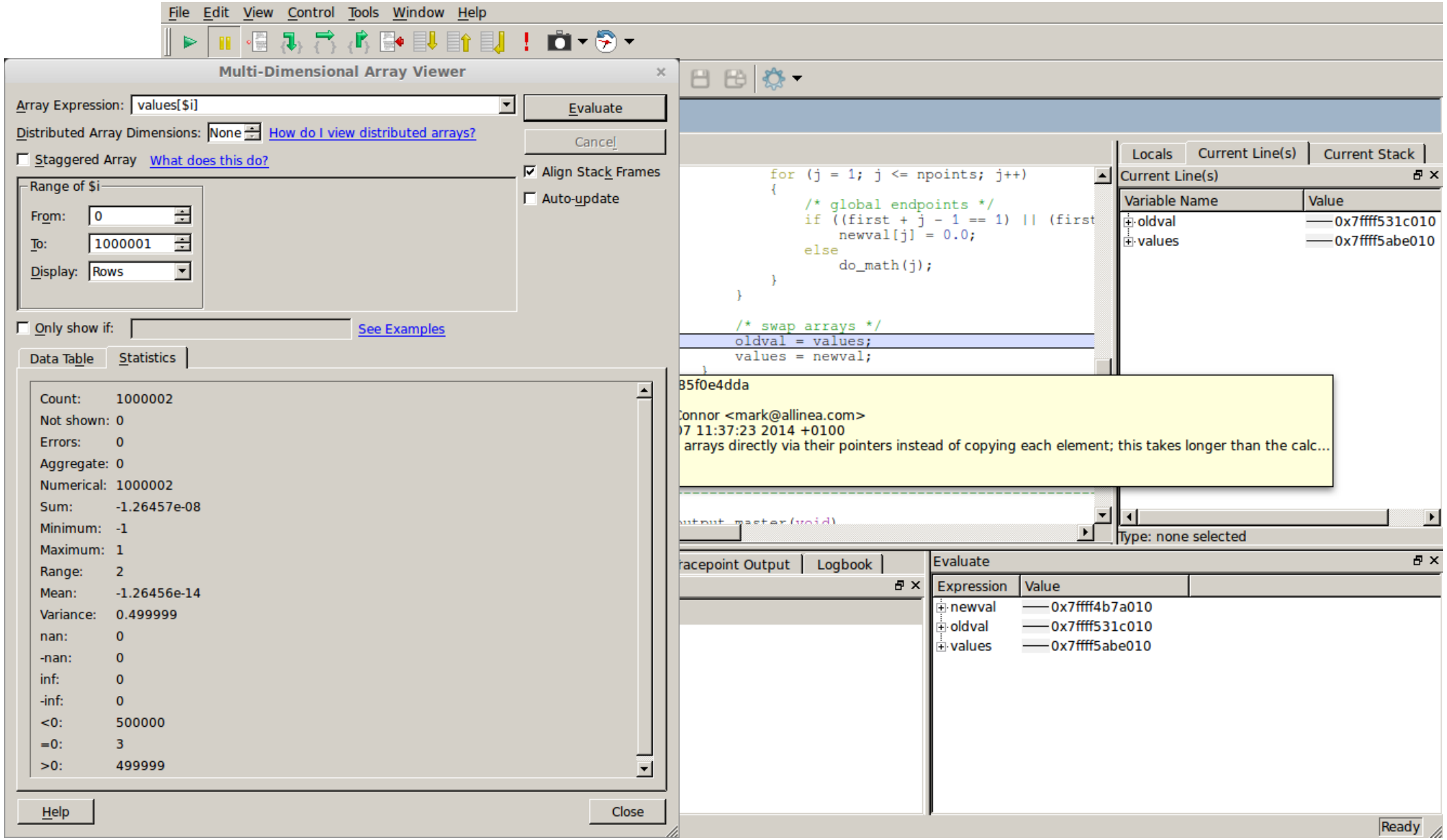
allinea DDT

We can visualize multidimensional data across all processes



allinea DDT

And generate statistical summaries of their contents



allinea DDT

Let’s run one iteration. Changes are automatically highlighted:

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)

- 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 right
- 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
```

Locals | Current Line(s) | Current Stack

Current Line(s)

Variable Name	Value
oldval	0x7ffff5abe01(
values	0x7ffff4b7a01(

Type: none selected

Input/Output | Breakpoints | Watchpoints | Stacks | Tracepoints | Tracepoint Output | Logbook

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	0x7ffff5abe010
values	0x7ffff4b7a010

Ready

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)

- 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 right
- 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)
```

Locals | Current Line(s) | Current Stack

Current Line(s)

Variable Name	Value
oldval	0x7ffff4b7a010
values	0x7ffff4b7a010

Type: none selected

Input/Output | Breakpoints | Watchpoints | Stacks | Tracepoints | Tracepoint Output | Logbook

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

Ready

allinea DDT

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

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)

- 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 right
- 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)
```

Locals | Current Line(s) | Current Stack

Current Line(s)

Variable Name	Value
oldval	0x7ffff4b7a010
values	0x7ffff4b7a010

Type: none selected

Input/Output | Breakpoints | Watchpoints | Stacks | Tracepoints | Tracepoint Output | Logbook

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

Ready

allinea DDT

Let’s fix that right now

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)

- 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 right
- External Code

wave_openmp.c*

This file has been edited.

```
216                 for (j = 1; j <= npoints; j++)
217                 {
218                     /* global endpoints */
219                     if ((first + j - 1 == 1) || (first + j - 1 == tp
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 = te
232             }
233         }
234
235         return iterations;
236     }
```

Locals | Current Line(s) | Current Stack

Current Line(s)

Variable Name	Value

Type: none selected

Input/Output | Breakpoints | Watchpoints | Stacks | Tracepoints | Tracepoint Output | Logbook

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

Ready

allinea DDT

This time we'll verify our fix before committing it

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)

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

wave_openmp.c

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
```

Locals | Current Line(s) | Current Stack

Current Line(s)

Variable Name	Value
oldval	0x7ffff5abe0
values	0x7ffff4b7a0

Type: none selected

Input/Output | Breakpoints | Watchpoints | Stacks | Tracepoints | Tracepoint Output | Logbook | Build Output

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**

Only show lines containing:

Evaluate

Expression	Value
newval	0x7ffff531c010
oldval	0x7ffff5abe010
values	0x7ffff4b7a010

Ready

allinea DDT

A tracepoint shows the arrays pointers are swapping correctly now

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)

- 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 informatic ... Mercurial.

```
219
220
221
222
223
224                 }
225
226                 /*
227                 {
228
229
230
231
232                 }
233             }
234         }
235
236         return iter
237     }
238
239     /*----------
240      *      Master receives results from workers and prints
```

Commit Changes

Modified Files:

M wave_openmp.c

Commit Message:

Fixed array swapping, validation tests now pass again.

Commit Cancel

Locals | Current Li... | Current S...

Current Line(s)

Variable Name	Value
oldval	0x7ffff53
values	0x7ffff5a

Type: none selected

Input/Output | Breakpoints | Watchpoints | Stacks | Tracepoints | Tracepoint Output | Logbook

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

allinea DDT

We have achieved just under a 2x speedup already

File Edit View Window Help

/home/mark/wave_openmp/wave_openmp
0 / 1 processes running
Started on Fri Nov 7 11:52:58 2014
Stopped on Fri Nov 7 11:53:31 2014

1501 samples
Runtime: 33s
Collection time: 2s

30%
Collecting and merging samples...

```
Wave solution with 1 processes
points = 1000000, running for 30 seconds

Points for validation:
0:0.00  200000:0.94  400000:0.58  600000:-0.58  800000:-0.94  999999:0.00
0:0.00  200000:0.95  400000:0.59  600000:-0.59  800000:-0.95  999999:0.00
Validation check passed

                min     mean        max
All time (ms): 30013   30013       30013
CPU time (ms): 29998   29998       29998
MPI time (ms): 15      15          15

CPU time (%) : 100     100         100
MPI time (%) : 0       0           0

Iterations   : 18440   18440       18440
points/second: 614.4M (614.4M per process)
wave finished
```

Type here ('Enter' to send):

More

1 process busy

allinea MAP

Now the majority of the time is in the do_math function where it belongs

File Edit View Window Help

Profiled: wave_openmp on 1 process, 4 cores (4 per process) Started: Fri Nov 7 11:52:58 2014 Runtime: **30s** Hide Metrics...

Application activity

CPU floating-point (%)
0 - 100 (29.1 avg)

Memory usage (kB)
49,115 - 80,462 (68,923 avg)

11:52:58-11:53:27 (29.953s): Main thread compute **0** %, OpenMP **44** %, MPI **0** %, Overhead **56** %, Sleeping **0** % | CPU floating-point **29.1** %; Memory usage **68,923** | Metrics Select All

wave_openmp.c

```
8 months ago  214          {
8 months ago  215  #pragma omp for
8 months ago  216              for (j = 1; j <= npoints; j++)
8 months ago  217              {
8 months ago  218                  /* global endpoints */
1.5%  8 months ago  219            if ((first + j - 1 == 1) || (first + j - 1 == tpoints))
0.2%  8 months ago  220                newval[j] = 0.0;
8 months ago  221                  else
42.3% 8 months ago  222                do_math(j);
8 months ago  223              }
8 months ago  224          }
8 months ago  225
<0.1% <1 hour ago  226         /* swap arrays */ {...}
8 months ago  233      }
```

Input/Output | Project Files | Stacks | OpenMP Regions

Stacks

Time	MPI	Overhead	Function(s) on line	Source	Position
			wave_openmp [program]		
			main		wave_openmp.c:325
			update	iterations = update(left, right);	wave_openmp.c:358
88.7%			update._omp_fn.0.constprop.1	#pragma omp parallel shared(newval, oldval, values)	wave_openmp.c:213
10.7%		10.7%	omp_in_final	#pragma omp parallel shared(newval, oldval, values)	wave_openmp.c:213
0.4%		0.1%	4 others		
0.2%	0.1%		2 others		

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

allinea MAP

But this function is not using vectorized operations…

File Edit View Window Help

Profiled: wave_openmp on 1 process, 4 cores (4 per process) Started: Fri Nov 7 11:52:58 2014 Runtime: **30s** Hide Metrics...

Application activity

CPU floating-point (%)
0 - 100 (29.1 avg)

CPU memory access (%)
0 - 100 (64.6 avg)

CPU fp vector (%)
0 - 0 (0.0 avg)

11:52:58-11:53:27 (29.953s): Main thread compute **0** %, OpenMP **44** %, MPI **0** %, Overhead **56** %, Sleeping **0** % | CPU floating-point **29.1** %; CPU memory access **64**. Metrics Select All

wave_openmp.c

```
8 months ago  144  void do_math(int i)
8 months ago  145  {
8 months ago  146      const double dtime = 1.0;
8 months ago  147      const double c = 1.0;
8 months ago  148      const double dx = 1.0;
8 months ago  149      double tau, sqtau;
8 months ago  150
8 months ago  151      tau = (c * dtime / dx);
8 months ago  152      sqtau = tau * tau;
19.5%  8 months ago  153      newval[i] = (2.0 * values[i]) - oldval[i]
22.8%  8 months ago  154          + (sqtau * (values[i-1] - (2.0 * values[i]) + values[i+1]));
8 months ago  155  }
8 months ago  156
```

Input/Output | Project Files | Stacks | OpenMP Regions

Stacks

Time	MPI	Overhead	Function(s) on line	Source	Position
			wave_openmp [program]		
			main		wave_openmp.c:325
			update	iterations = update(left, right);	wave_openmp.c:358
88.7%			update._omp_fn.0.constprop.1	#pragma omp parallel shared(newval, oldval, values)	wave_openmp.c:213
10.7%	10.7%		omp_in_final	#pragma omp parallel shared(newval, oldval, values)	wave_openmp.c:213
0.4%		0.1%	4 others		
0.2%	0.1%		2 others		

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

allinea MAP

Can we remove this conditional from the inner loop?

File Edit View Window Help

Profiled: wave_openmp on 1 process, 4 cores (4 per process) Started: Fri Nov 7 11:52:58 2014 Runtime: **30s** Hide Metrics...

Application activity

CPU floating-point (%)
0 - 100 (29.1 avg)

CPU memory access (%)
0 - 100 (64.6 avg)

CPU fp vector (%)
0 - 0 (0.0 avg)

11:52:58-11:53:27 (29.953s): Main thread compute **0** %, OpenMP **44** %, MPI **0** %, Overhead **56** %, Sleeping **0** % | CPU floating-point **29.1** %; CPU memory access **64**. Metrics Select All

wave_openmp.c

```
8 months ago  214            {
8 months ago  215  #pragma omp for
8 months ago  216                for (j = 1; j <= npoints; j++)
8 months ago  217                {
8 months ago  218                    /* global endpoints */
1.5%  8 months ago  219              if ((first + j - 1 == 1) || (first + j - 1 == tpoints))
0.2%  8 months ago  220                  newval[j] = 0.0;
8 months ago  221                    else
42.3% 8 months ago  222                  do_math(j);
8 months ago  223                }
8 months ago  224            }
8 months ago  225
<0.1% <1 hour ago  226       /* swap arrays */ {...}
```

Input/Output | Project Files | Stacks | OpenMP Regions

Stacks

Time	MPI	Overhead	Function(s) on line	Source	Position
			wave_openmp [program]		
			main		wave_openmp.c:325
			update	iterations = update(left, right);	wave_openmp.c:358
88.7%			update._omp_fn.0.constprop.1	#pragma omp parallel shared(newval, oldval, values)	wave_openmp.c:213
10.7%	10.7%		omp_in_final	#pragma omp parallel shared(newval, oldval, values)	wave_openmp.c:213
0.4%		0.1%	4 others		
0.2%	0.1%		2 others		

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

allinea MAP

We can zero the endpoints after the computation

File Edit View Window Help

Profiled: wave_openmp on 1 process, 4 cores (4 per process) Started: Fri Nov 7 11:52:58 2014 Runtime: **30s** Show Metrics...

wave_openmp.c*

This file has been edited.

```
<0.1%   199          if (first != 1) { ...}
        204          /* Exchange data with "right-hand" neighbor */
        205          if (first + npoints -1 != tpoints) { ...}
<0.1%   210          time_mpi_stop();
        211
        212          /* update points along line */
2.7%    213 #pragma omp parallel shared(newval, oldval, values)
        214          {
        215 #pragma omp for
        216              for (j = 1; j <= npoints; j++)
        217              {
42.3%   218                  do_math(j);
        219              }
        220          }
        221
        222          /* zero global endpoints */
        223          if (first == 1)
        224              newval[1] = 0.0;
        225          if (first + npoints - 1 == tpoints)
        226              newval[npoints] = 0.0;
        227
        228
<0.1%   229          /* swap arrays */ {...}
        236      }
        237  }
        238
        239  return iterations;
        240 }
```

Input/Output | Project Files | Stacks | OpenMP Regions

Stacks

Time	MPI	Overhead	Function(s) on line	Source	Position
			wave_openmp [program]		
			main		wave_openmp.c:325
			update	iterations = update(left, right);	wave_openmp.c:358
88.7%			update._omp_fn.0.constprop.1	#pragma omp parallel shared(newval, oldval, values)	wave_openmp.c:213
10.7%		10.7%	omp_in_final	#pragma omp parallel shared(newval, oldval, values)	wave_openmp.c:213
0.4%		0.1%	4 others		
0.2%	0.1%		2 others		

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

allinea MAP

The new code compiles so let’s commit our changes and see if it’s faster

File Edit View Window Help

Profiled: wave_openmp on 1 process, 4 cores (4 per process) Started: Fri Nov 7 11:52:58 2014 Runtime: **30s** Show Metrics...

wave_openmp.c

This file has been edited.

```
199            if (first != 1) { ...}
204            /* Exchange data with "right-hand" neighbor */
205            if (first + npoints -1 != tpoints) { ...}
210            time_mpi_stop();
211
212            /* update points along line */
213 #pragma omp parallel shared(newval, oldval, values)
214            {
215 #pragma omp for
216                for (j = 1; j <= npoints; j++)
217                {
218                    do_math(j);
219                }
220            }
221
222            /* zero global endpoints */
223            if (first == 1)
224                newval[1] = 0.0;
225            if (first + npoints - 1 == tpoints)
226                newval[npoints] = 0.0;
227
228
229            /* swap arrays */ {...}
236        }
237    }
238
239    return iterations;
240 }
```

<0.1% 2.7% 42.3% <0.1%

Commit Changes

Modified Files:

M wave_openmp.c

Commit Message:

Zero the endpoints after the calculation rather than checking each element; this should allow the compiler to auto-vectorize the do_math function when called in this loop.

Commit Cancel

Input/Output | Project Files | Stacks | OpenMP Regions | Build Output

Build Output

```
12:00:22 - Running 'make' in '/home/mark/wave_openmp'
mpicc -g -O3 -fopenmp wave_openmp.c -o wave_openmp -lm

12:00:22 - Finished 'make' in '/home/mark/wave_openmp' - build time: 0 seconds
```

12:00:22 - Finished 'make' in '/home/mark/wave_openmp' - build time: 0 seconds

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

allinea MAP

Apparently it’s only 1% faster… what’s going on?

File Edit View Window Help

Profiled: wave_openmp on 1 process, 4 cores (4 per process) Started: Fri Nov 7 12:01:43 2014 Runtime: **30s** Hide Metrics...

Application activity

CPU floating-point (%)
0 - 50 (4.6 avg)

Memory usage (kB)
57,561 - 78,311 (68,942 avg)

12:01:43-12:02:13 (29.957s): Main thread compute **0** %, OpenMP **27** %, MPI **0** %, Overhead **72** %, Sleeping **0** % | CPU floating-point **4.6** %; Memory usage **68,942** kB Metrics Select All

wave_openmp.c

```
0.8%   8 months ago  213  #pragma omp parallel shared(newval, oldval, values)
       8 months ago  214          {
       8 months ago  215  #pragma omp for
       8 months ago  216              for (j = 1; j <= npoints; j++)
       8 months ago  217              {
27.3%  <1 hour ago   218                  do_math(j);
       8 months ago  219              }
       8 months ago  220          }
       <1 hour ago   221
       <1 hour ago   222          /* zero global endpoints */
       <1 hour ago   223          if (first == 1)
       <1 hour ago   224              newval[1] = 0.0;
       <1 hour ago   225          if (first + npoints - 1 == tpoints)
       <1 hour ago   226              newval[npoints] = 0.0;
       <1 hour ago   227
       8 months ago  228
```

Input/Output | Project Files | Stacks | OpenMP Regions | Build Output

Input/Output

```
CPU time (ms): 30001    30001       30001
MPI time (ms): 16       16          16

CPU time (%) : 100      100         100
MPI time (%) : 0        0           0

Iterations   : 18650    18650       18650
points/second: 621.3M (621.3M per process)
wave finished
```

Showing data from 1,000 samples taken over 1 process (1000 per process) Allinea Ultimate map-smoketest-scripts-5.0 0a3f65bcf767 Nov 7 2014

allinea MAP

The floating point operations are more efficient but the code is memory-bound

File Edit View Window Help

Profiled: wave_openmp on 1 process, 4 cores (4 per process) Started: Fri Nov 7 12:01:43 2014 Runtime: **30s** Hide Metrics...

Application activity

CPU floating-point (%)
0 - 50 (4.6 avg)

CPU memory access (%)
10 - 100 (71.8 avg)

CPU fp vector (%)
0 - 50 (4.6 avg)

12:01:43-12:02:13 (29.957s): Main thread compute **0** %, OpenMP **27** %, MPI **0** %, Overhead **72** %, Sleeping **0** % | CPU floating-point **4.6** %; CPU memory access **71.8** Metrics Select All

wave_openmp.c

```
8 months ago  144  void do_math(int i)
8 months ago  145  {
8 months ago  146      const double dtime = 1.0;
8 months ago  147      const double c = 1.0;
8 months ago  148      const double dx = 1.0;
8 months ago  149      double tau, sqtau;
8 months ago  150
8 months ago  151      tau = (c * dtime / dx);
8 months ago  152      sqtau = tau * tau;
21.7%  8 months ago  153      newval[i] = (2.0 * values[i]) - oldval[i]
5.6%   8 months ago  154          + (sqtau * (values[i-1] - (2.0 * values[i]) + values[i+1]));
8 months ago  155  }
8 months ago  156
```

Input/Output | Project Files | Stacks | OpenMP Regions | Build Output

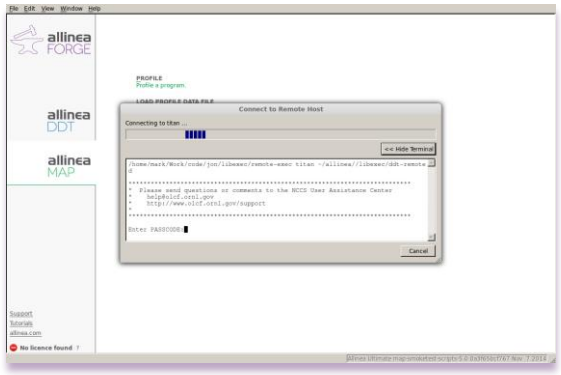
Input/Output

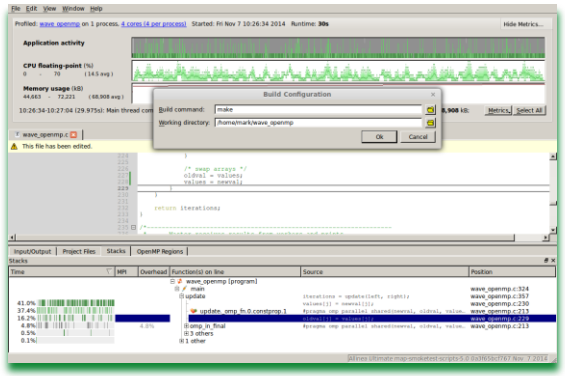
```
CPU time (ms): 30001    30001       30001
MPI time (ms): 16       16          16

CPU time (%) : 100      100         100
MPI time (%) : 0        0           0

Iterations   : 18650    18650       18650
points/second: 621.3M (621.3M per process)
wave finished
```

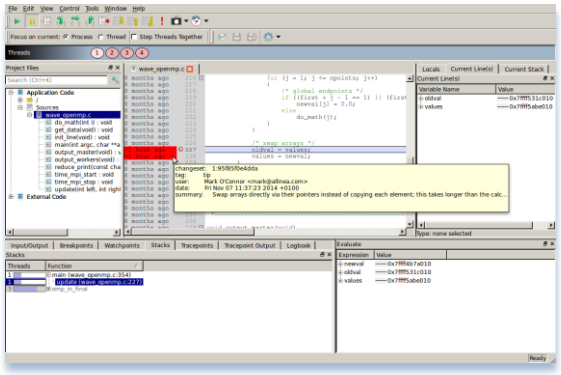
Showing data from 1,000 samples taken over 1 process (1000 per process) Allinea Ultimate map-smoketest-scripts-5.0 0a3f65bcf767 Nov 7 2014

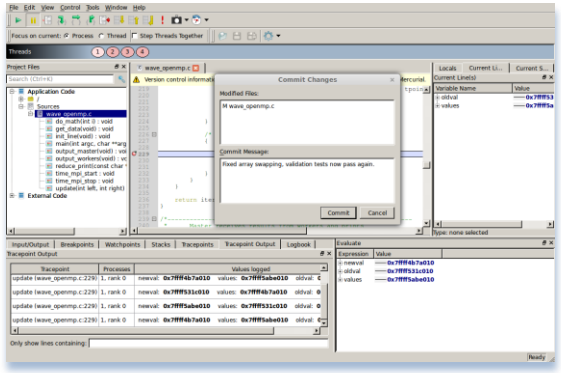
allinea MAP

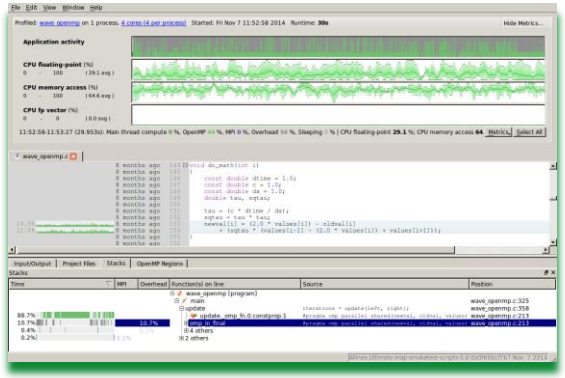
Optimizing a 1-D OpenMP/MPI Wave Equation Solver

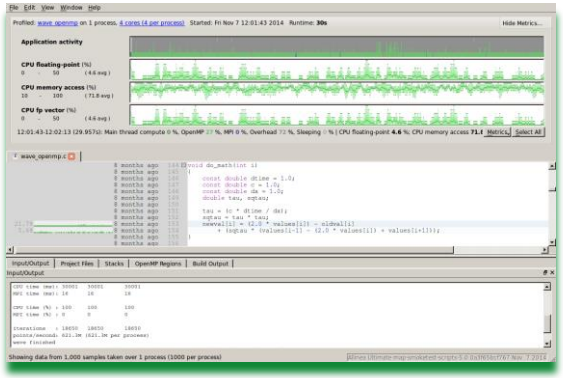




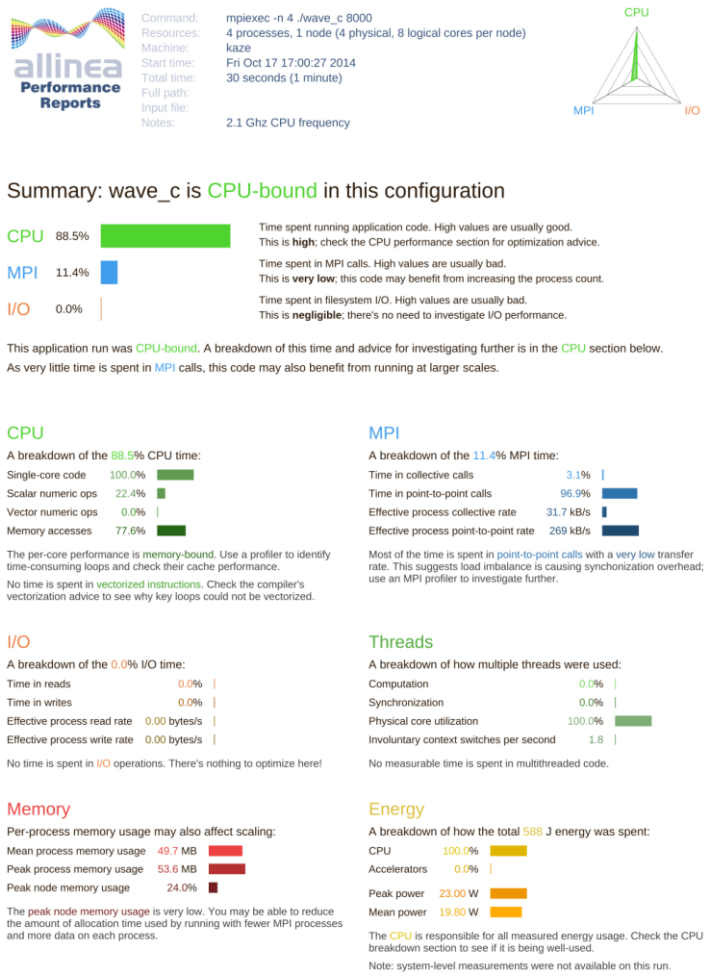








Our wave equation solver is put into production…



allinea Performance Reports

Command: mpiexec -n 4 ./wave_c 8000
Resources: 4 processes, 1 node (4 physical, 8 logical cores per node)
Machine: kaze
Start time: Fri Oct 17 17:00:27 2014
Total time: 30 seconds (1 minute)
Full path:
Input file:
Notes: 2.1 Ghz CPU frequency

CPU MPI I/O

Summary: wave_c is CPU-bound in this configuration

CPU 88.5% — Time spent running application code. High values are usually good. This is **high**; check the CPU performance section for optimization advice.

MPI 11.4% — Time spent in MPI calls. High values are usually bad. This is **very low**; this code may benefit from increasing the process count.

I/O 0.0% — Time spent in filesystem I/O. High values are usually bad. This is **negligible**; there's no need to investigate I/O performance.

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

CPU

A breakdown of the 88.5% CPU time:

Single-core code	100.0%
Scalar numeric ops	22.4%
Vector numeric ops	0.0%
Memory accesses	77.6%

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

No 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 11.4% MPI time:

Time in collective calls	3.1%
Time in point-to-point calls	96.9%
Effective process collective rate	31.7 kB/s
Effective process point-to-point rate	269 kB/s

Most of the time is spent in point-to-point calls with a very low transfer rate. This suggests load imbalance is causing synchonization overhead; use an MPI profiler to investigate further.

I/O

A breakdown of the 0.0% I/O time:

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

No time is spent in I/O operations. There's nothing to optimize here!

Threads

A breakdown of how multiple threads were used:

Computation	0.0%
Synchronization	0.0%
Physical core utilization	100.0%
Involuntary context switches per second	1.8

No measurable time is spent in multithreaded code.

Memory

Per-process memory usage may also affect scaling:

Mean process memory usage	49.7 MB
Peak process memory usage	53.6 MB
Peak node memory usage	24.0%

The peak node memory usage is very low. You may be able to reduce the amount of allocation time used by running with fewer MPI processes and more data on each process.

Energy

A breakdown of how the total 588 J energy was spent:

CPU	100.0%
Accelerators	0.0%
Peak power	23.00 W
Mean power	19.80 W

The CPU is responsible for all measured energy usage. Check the CPU breakdown section to see if it is being well-used.

Note: system-level measurements were not available on this run.

CPU

A breakdown of the 88.5% CPU time:

Single-core code	100.0%
Scalar numeric ops	22.4%
Vector numeric ops	0.0%
Memory accesses	77.6%

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

No time is spent in vectorized instructions. Check the compiler's vectorization advice to see why key loops could not be vectorized.

Memory

Per-process memory usage may also affect scaling:

Mean process memory usage	49.7 MB
Peak process memory usage	53.6 MB
Peak node memory usage	24.0%

The peak node memory usage is very low. You may be able to reduce the amount of allocation time used by running with fewer MPI processes and more data on each process.