Using DDT

Debugging programs with DDT

Peter Towers

HPC Systems Section

Peter.Towers@ecmwf.int

ECMWF

Allinea DDT

- **DDT is a very popular interactive debugger**
  - **Developed by Allinea Software in the UK**
  - **Modern graphical user interface**
  - **Highly scalable to large numbers of tasks/threads**
- **Cray provide a 2048 task licence**
  - **No limit to threads per task**
  - **Support from Cray backed up by Allinea**
- **Version 4.2.1 (default) and 5.0.1 installed**
  - **Version 5 also called Allinea Forge**
  - **Version 6 will be available from Cray later this year**
- **Version 4.2.2 also available on the latest desktops**
  - **Uses the same licence**

DDT Features

- **Examine sources**
- **Set breakpoints**
  - **Pause at a line of source**
  - **Conditional breakpoints supported**
- **Examine variables, Fortran modules, stack**
- **Set tracepoints**
  - **Output values of variables at a line without pausing**
- **Set watchpoints**
  - **Pause when a variable changes value**

DDT Features

- **Step through execution line by line**
  - Step to the next line or into a function
  - Step over a line
  - Step out of a function
- **Catch signals**
  - Segmentation violations
  - Floating point exceptions
- **Track and debug memory usage**
  - Record all memory allocations
  - Find memory overwrites
- **Examine message queues**
- **……**

Compilation

- **DDT uses the GNU Debugger (GDB) under the covers**
- **Compile with symbolic debug information turned on**
- **With CCE use**
  - **-g for debug info and no optimisation**
  - **-G1 or -G2 for debug info with optimisation**
- **With Intel use**
  - **-g for debug info**

Launching DDT in batch

export DISPLAY=<*your workstation*>:0.0

module load ddt/5.0.1.3_42607

ddt –n ? –mpiargs ‘more aprun args’ a.out

For example

ddt –n 4 –mpiargs ‘–N 4 –ss –cc cpu –d 6’ a.out

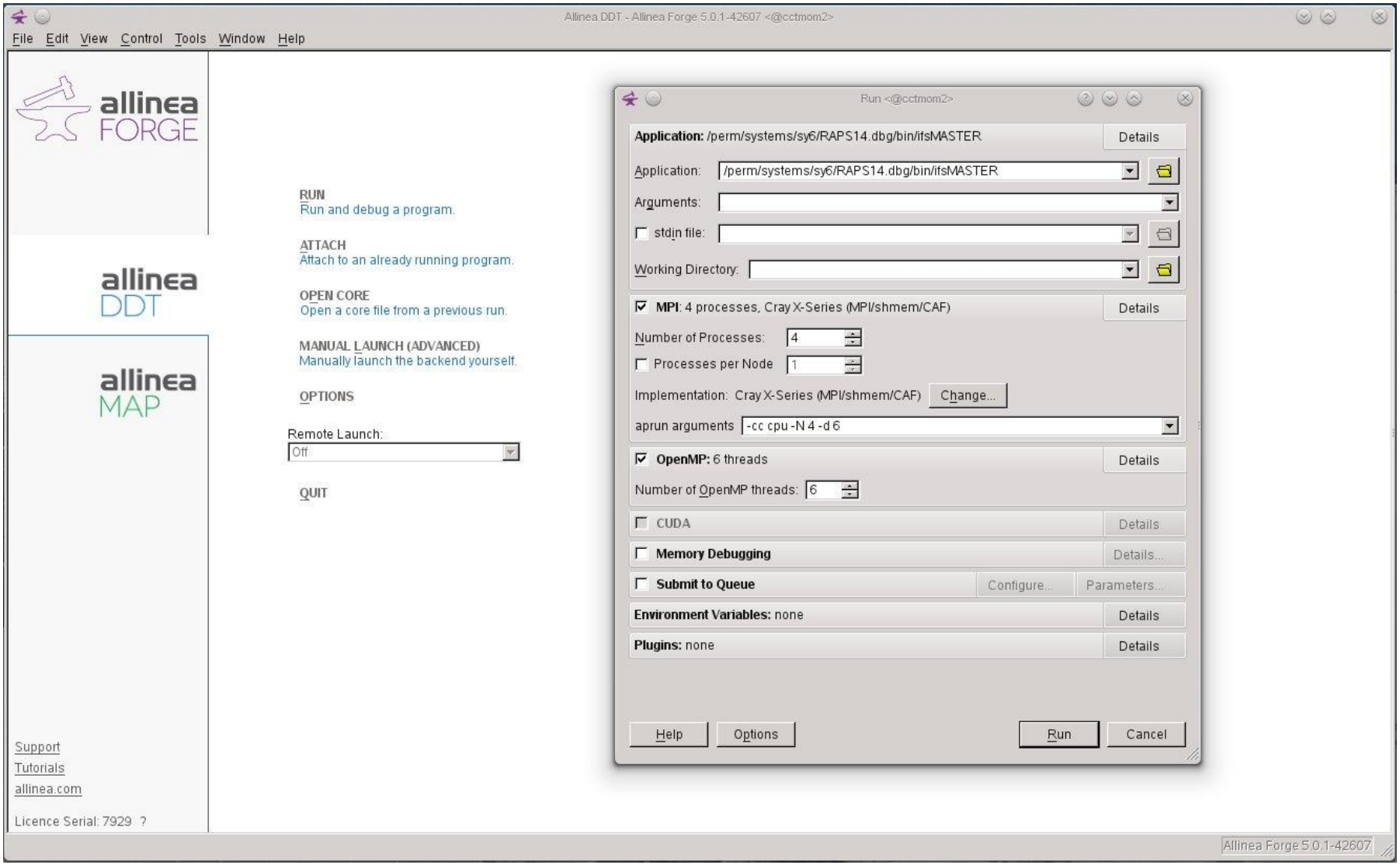
Simple Job Script

```
#!/bin/ksh
#PBS -q np
#PBS -N ddtdemo
#PBS -j oe
#PBS -o job.out
#PBS -l EC_total_tasks=4
#PBS -l EC_threads_per_task=6
#PBS -l EC_hyperthreads=1
#PBS -l walltime=00:60:00

cd $HOME/Debug

module load ddt

export DISPLAY=juliet:0.0

export OMP_NUM_THREADS=6

ddt -n 4 -mpiargs '-ss -cc cpu -N 4 -d 6' ./hello_mpi
```

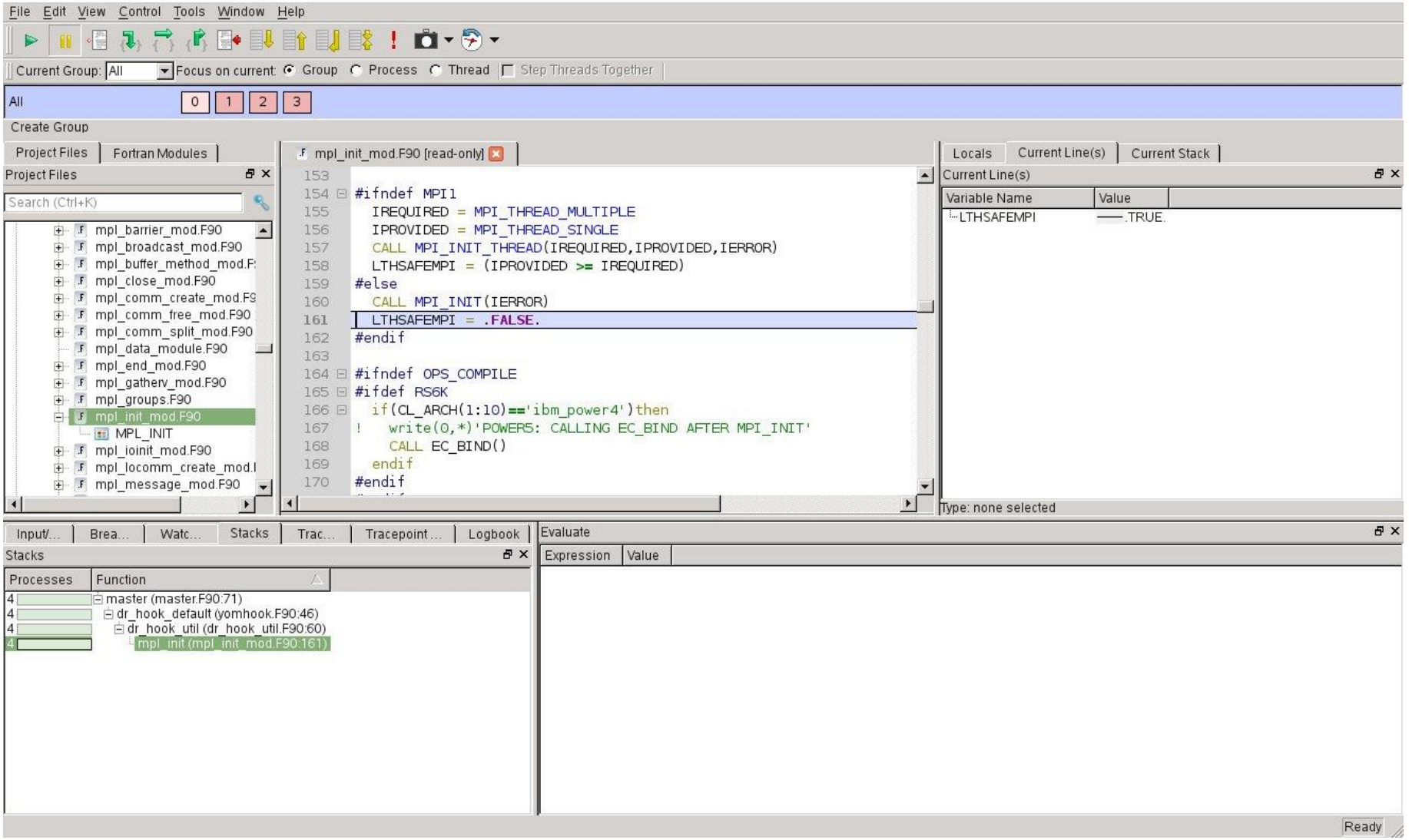
Live Demo Using IFS

- **TL159 running on 1 node**
  - **Initially with 4 tasks x 6 threads**
  - **Then with 24 tasks x 1 thread**
- **Latest RAPS14 benchmark code (CY41R2)**
- **Run in batch**
- **Selected screen shots from DDT follow in the presentation material**
- **Now for the live demo…….**

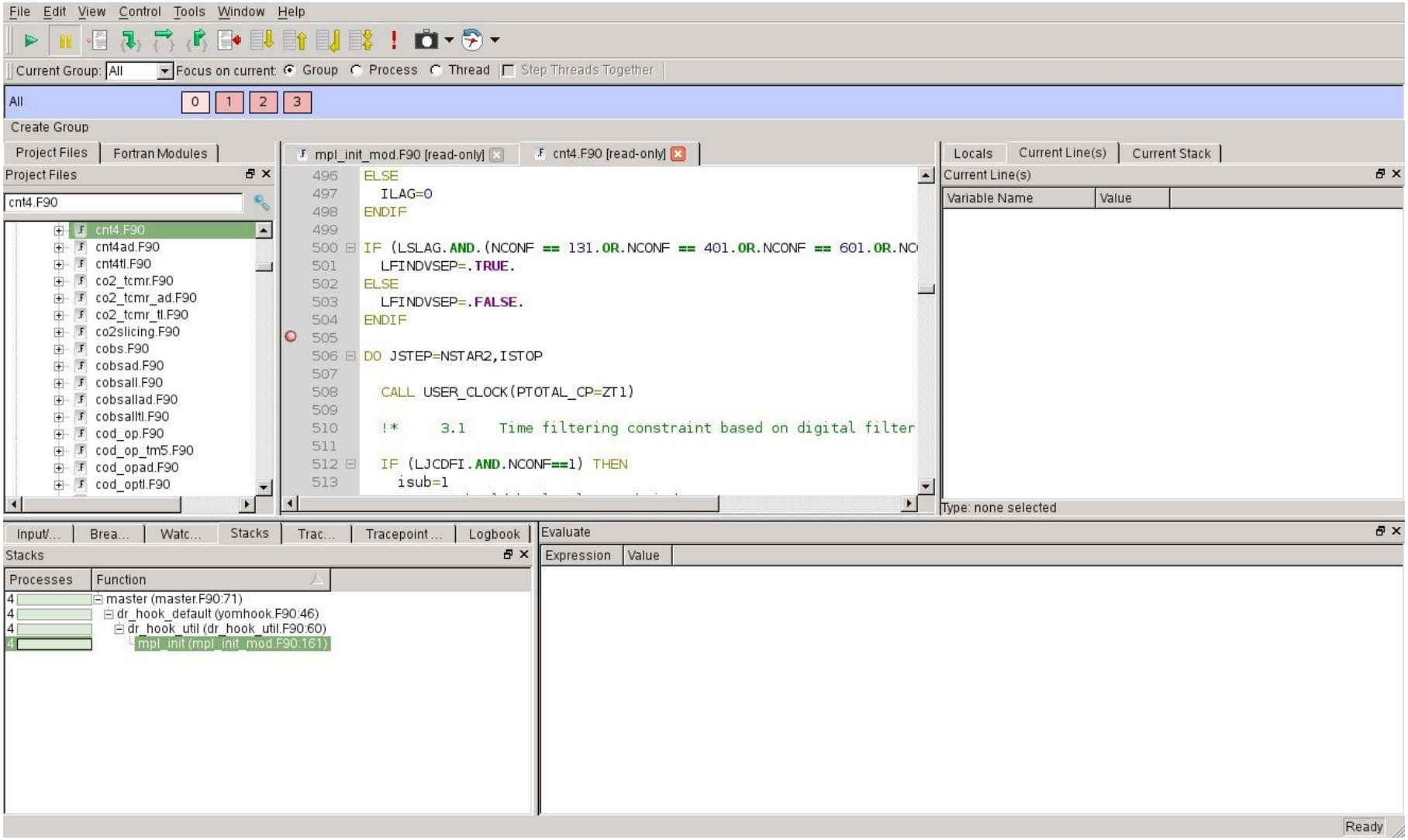
Before launching executable



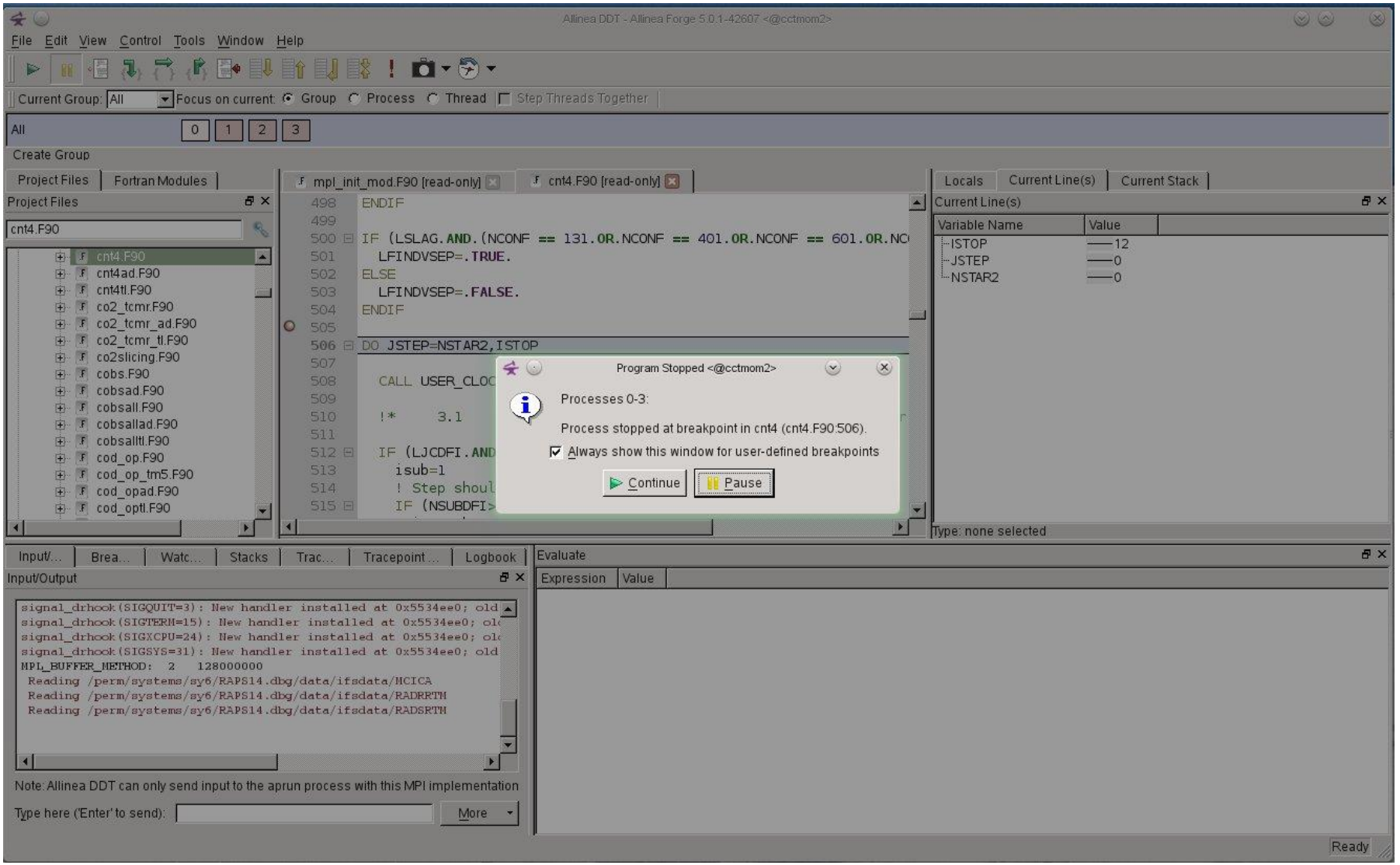
Paused after MPI_INIT



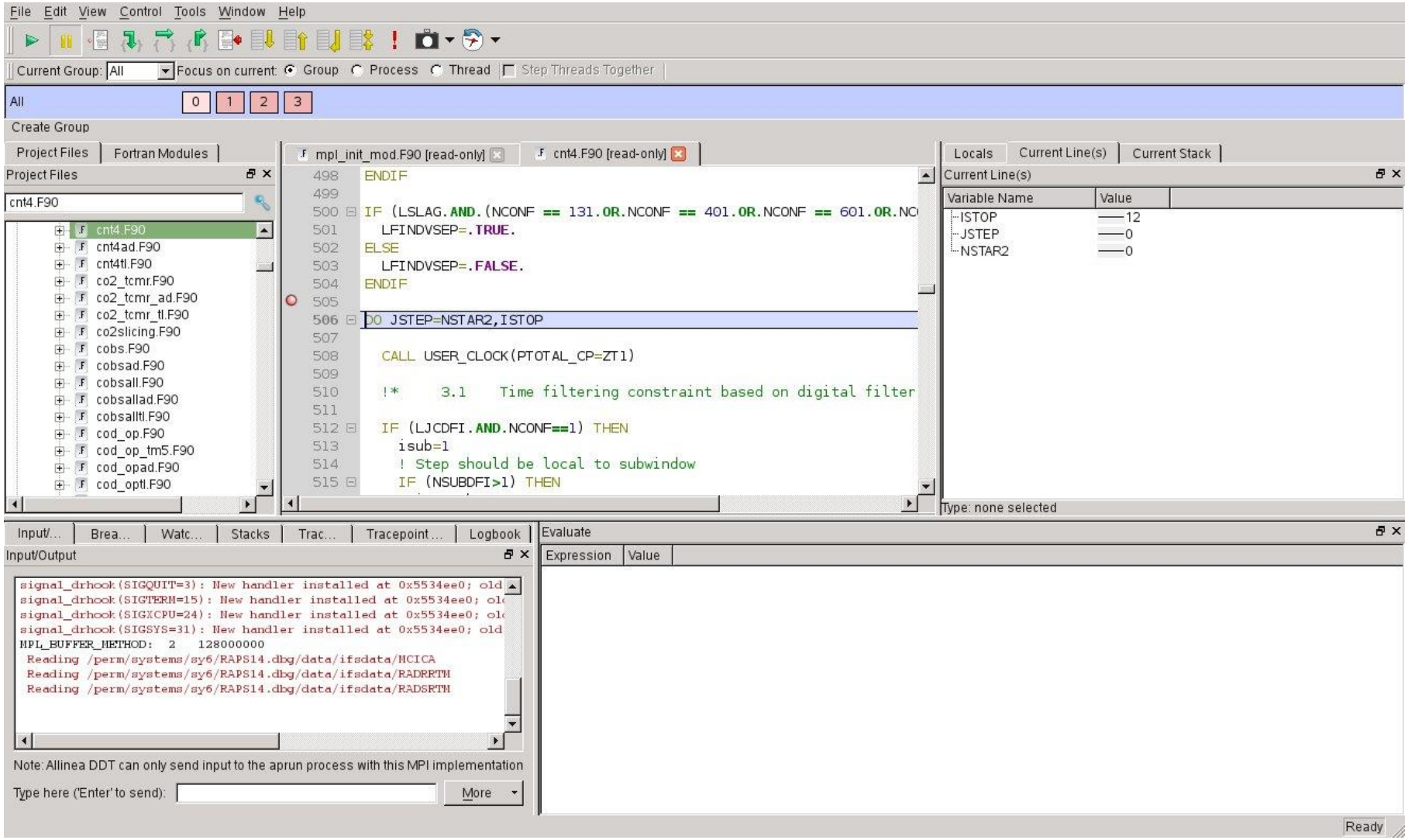
After selecting cnt4.F90 and setting a breakpoint at line 505



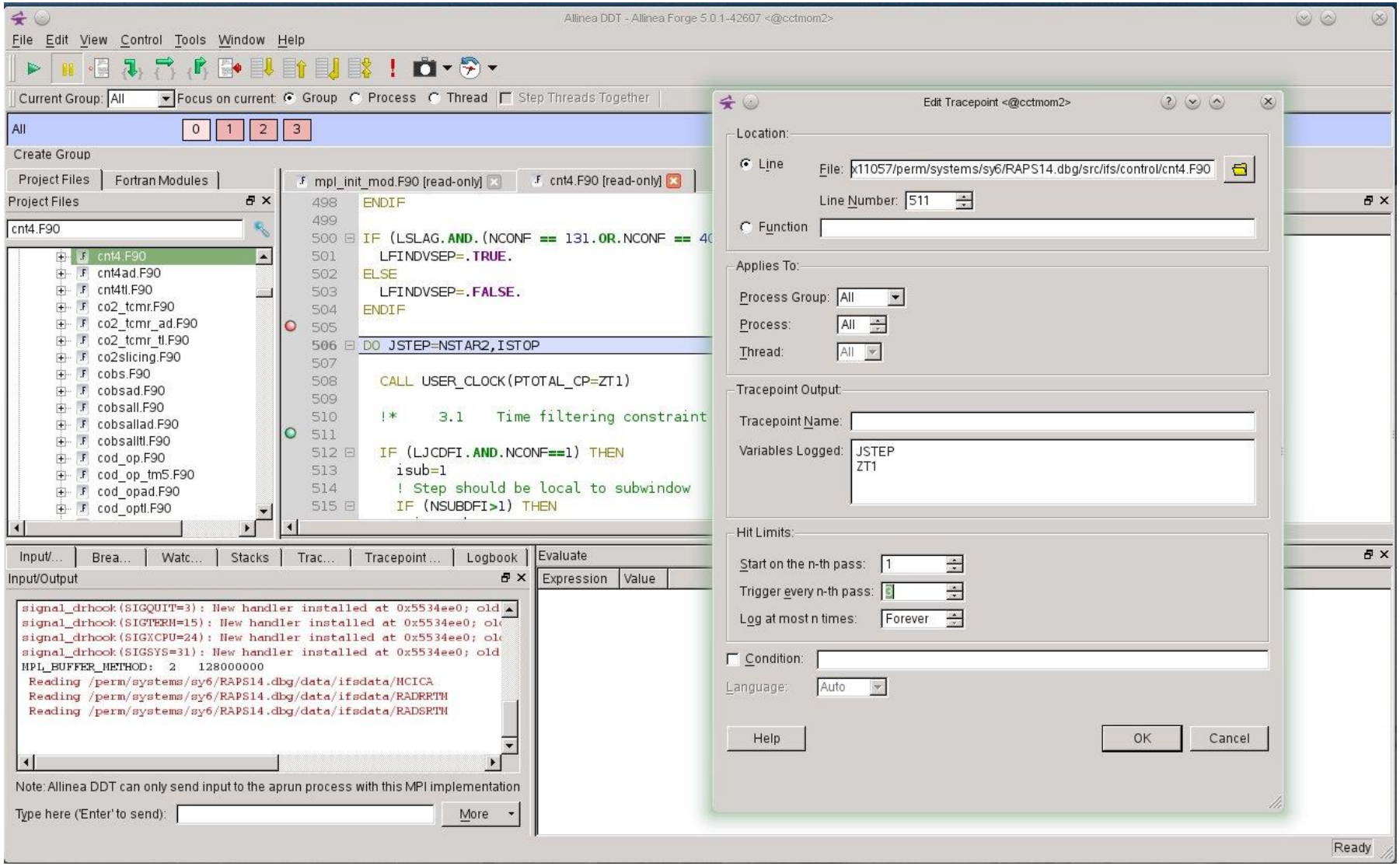
Hitting the breakpoint



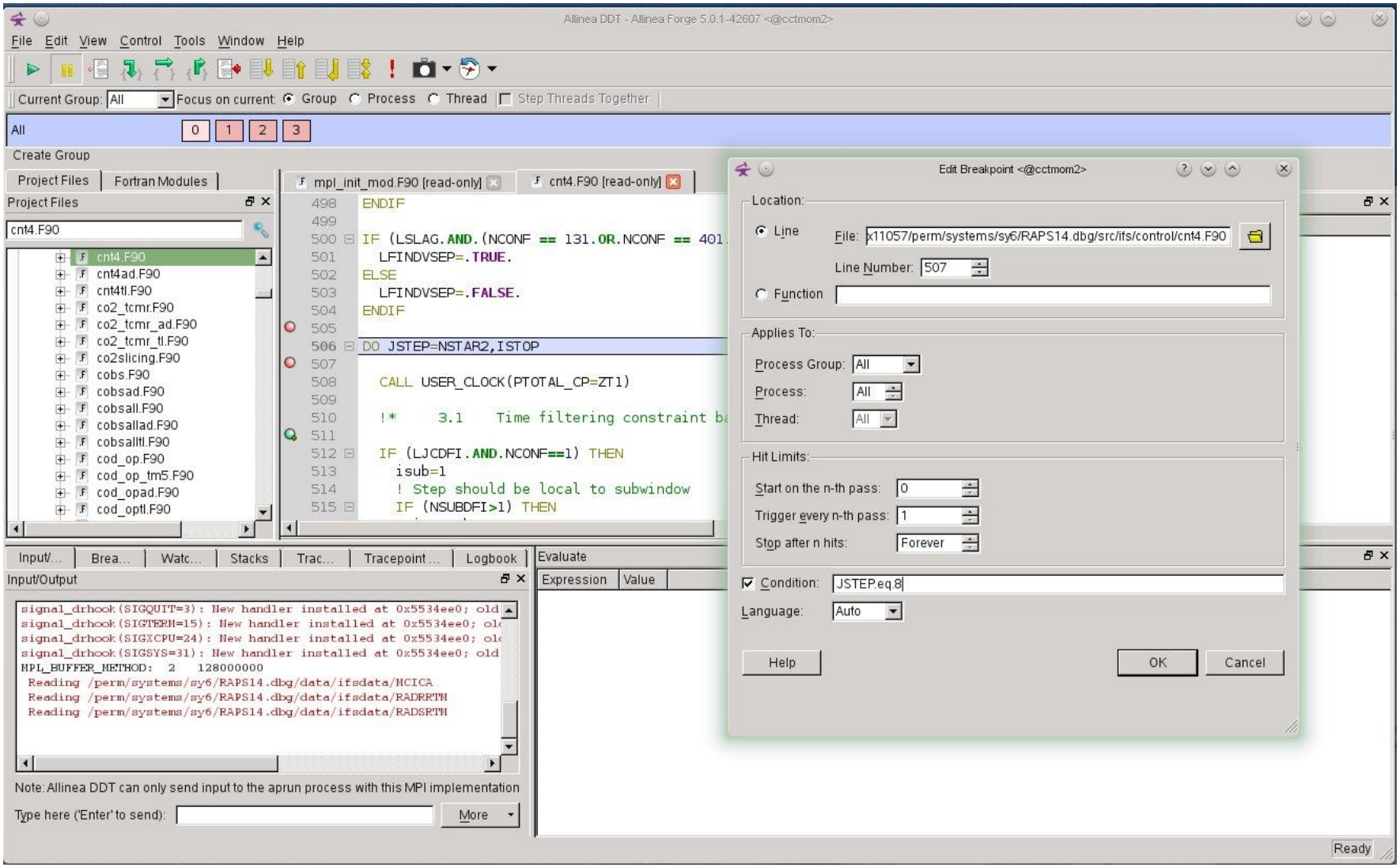
At the breakpoint



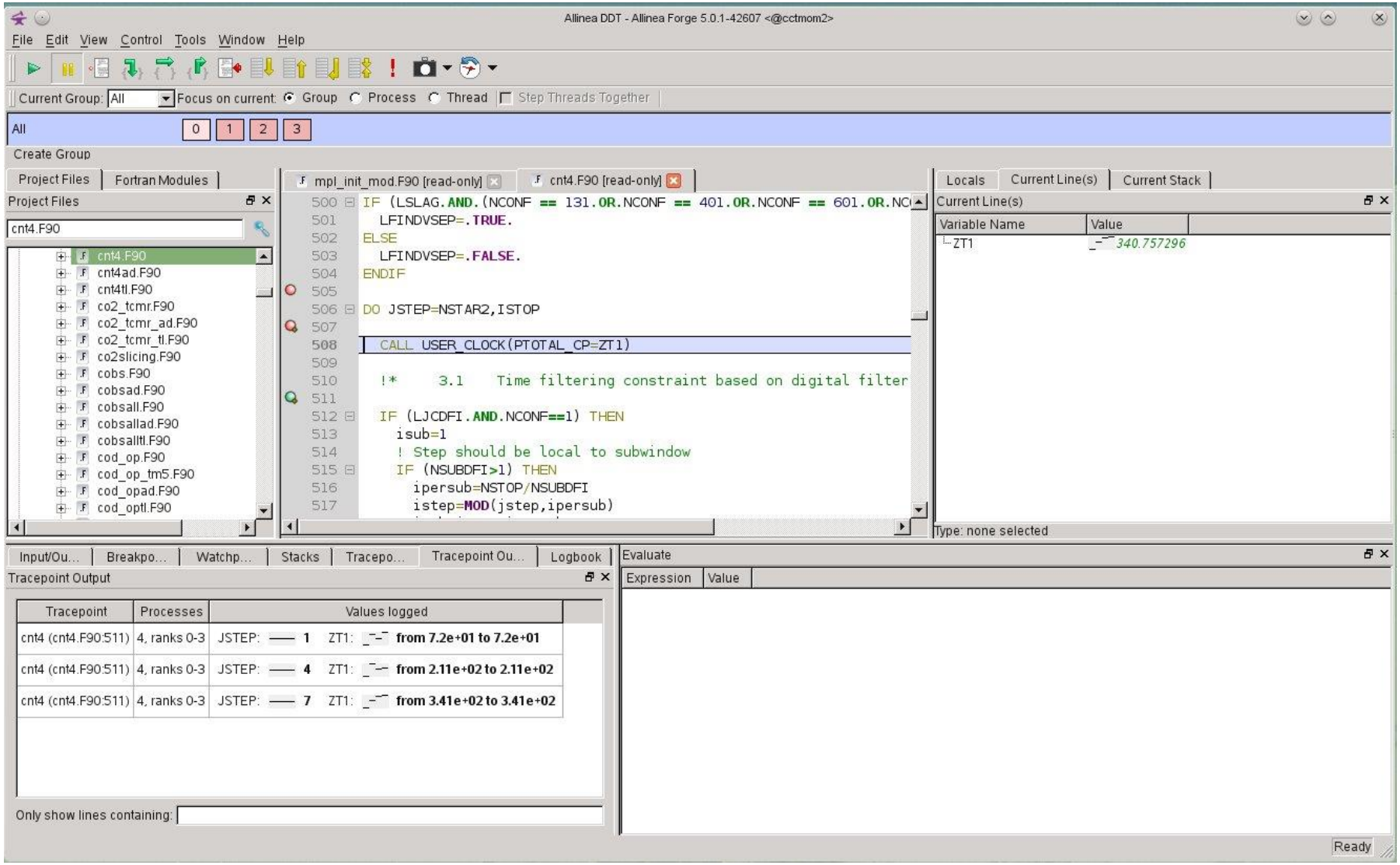
Setting a trace point at line 511 for JSTEP and ZT1 every 3rd step



Setting a conditional breakpoint at line 507 for step 8



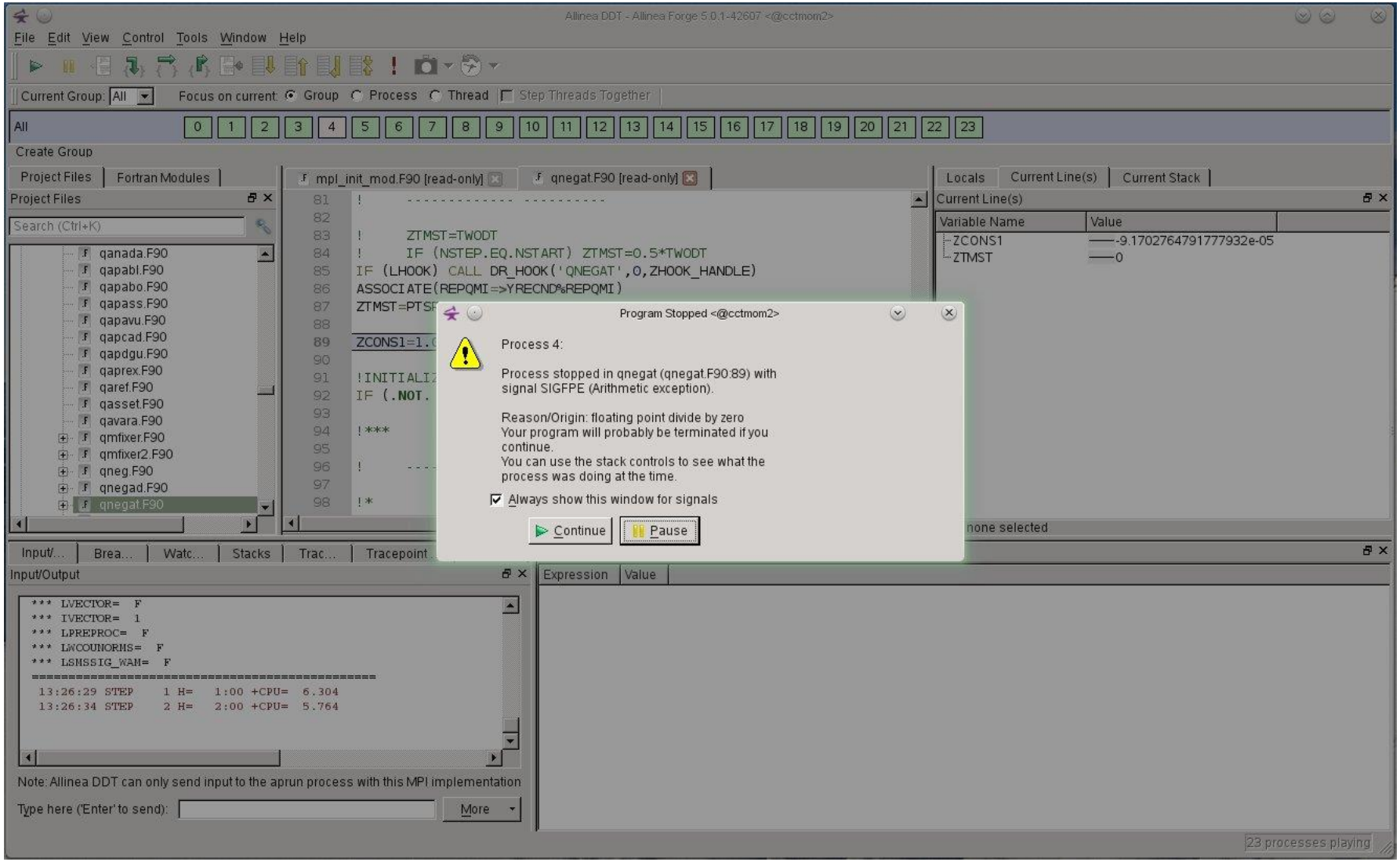
Trace point output after 8 steps



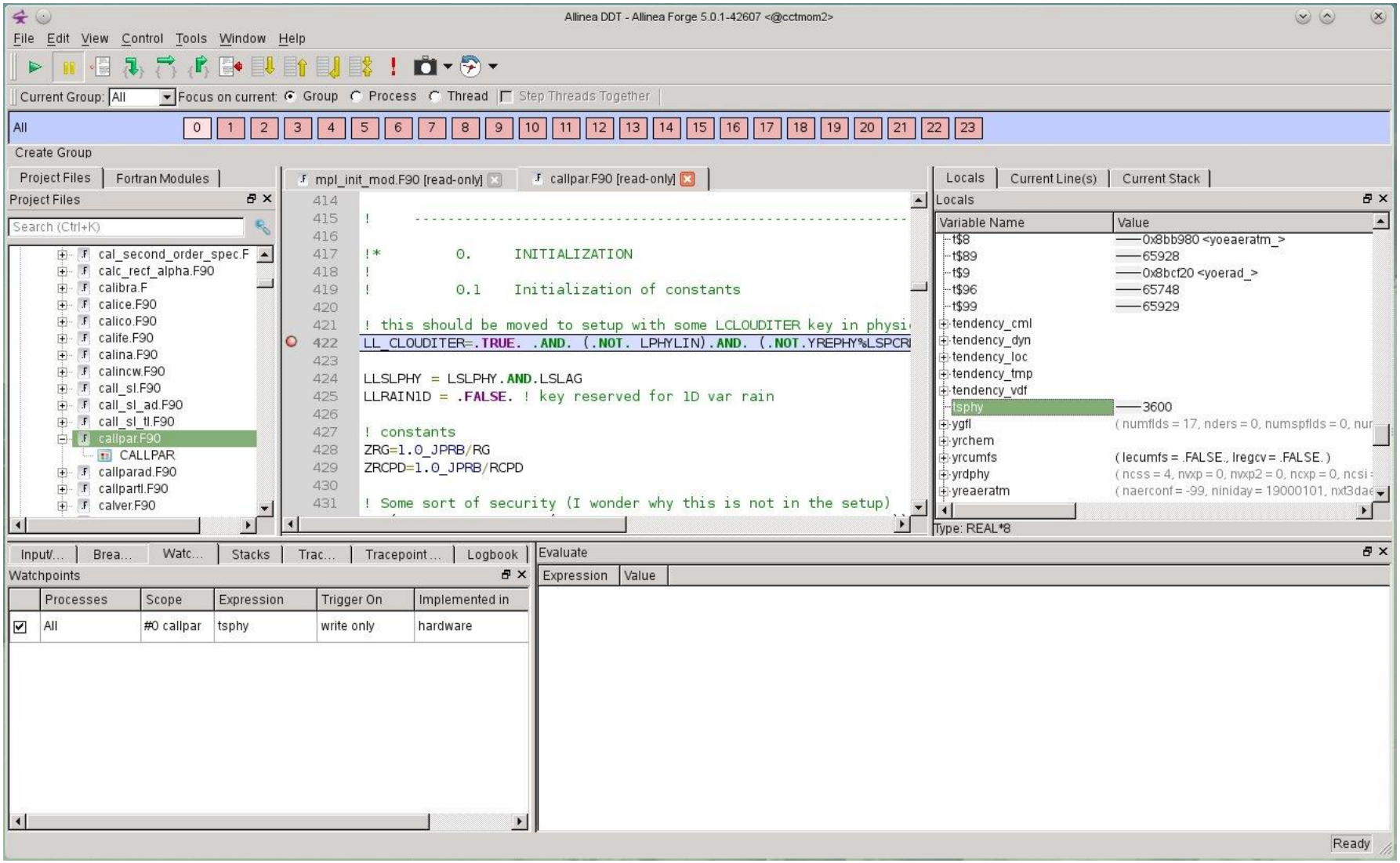
Running a 24 x 1 configuration

- **IFS runs until it gets a Floating Point Exception**
- **Caused by a divide by zero**
- **The physics time slice has gone to zero**
- **Will use a watch point to find out where**

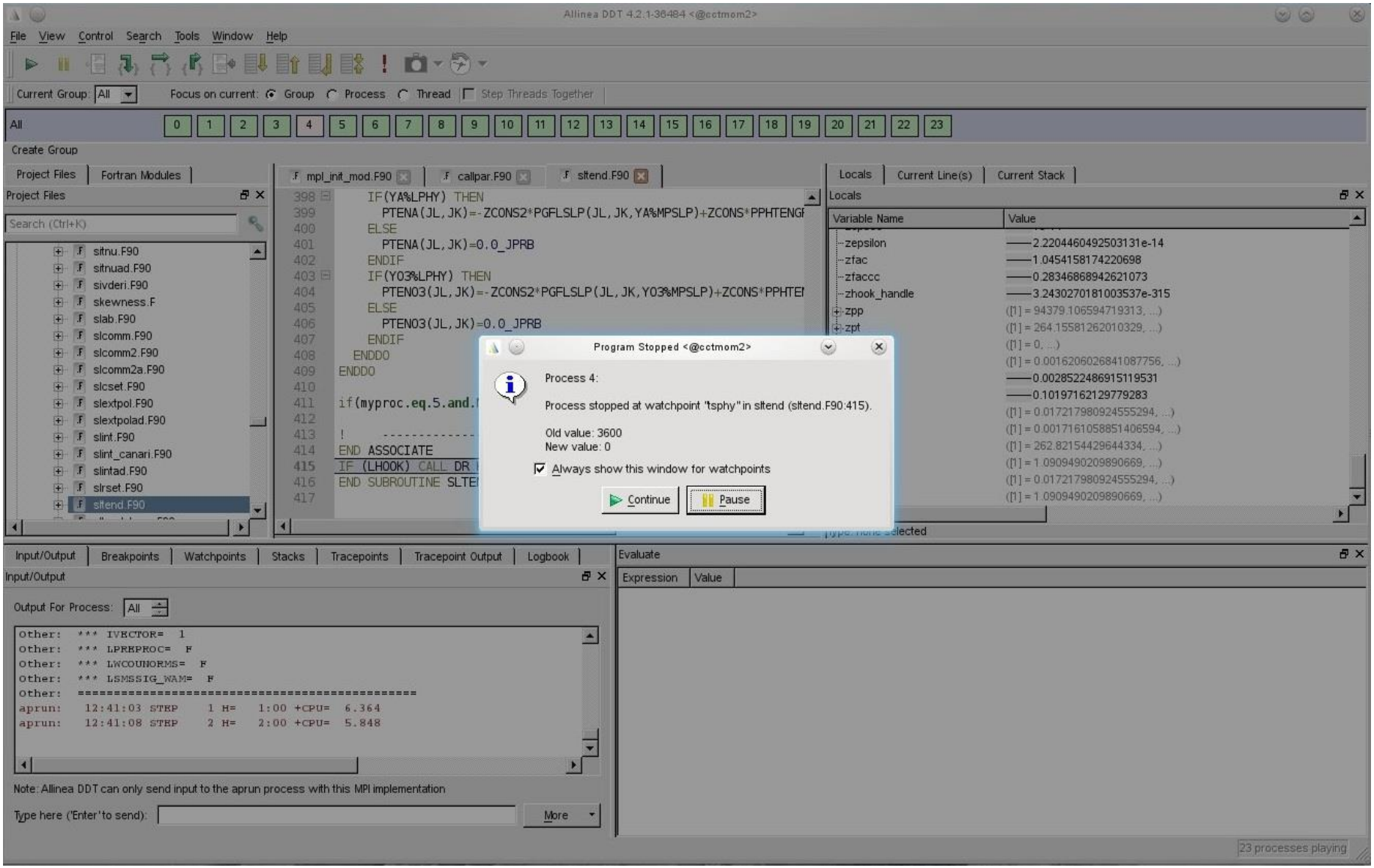
Hitting the divide by zero



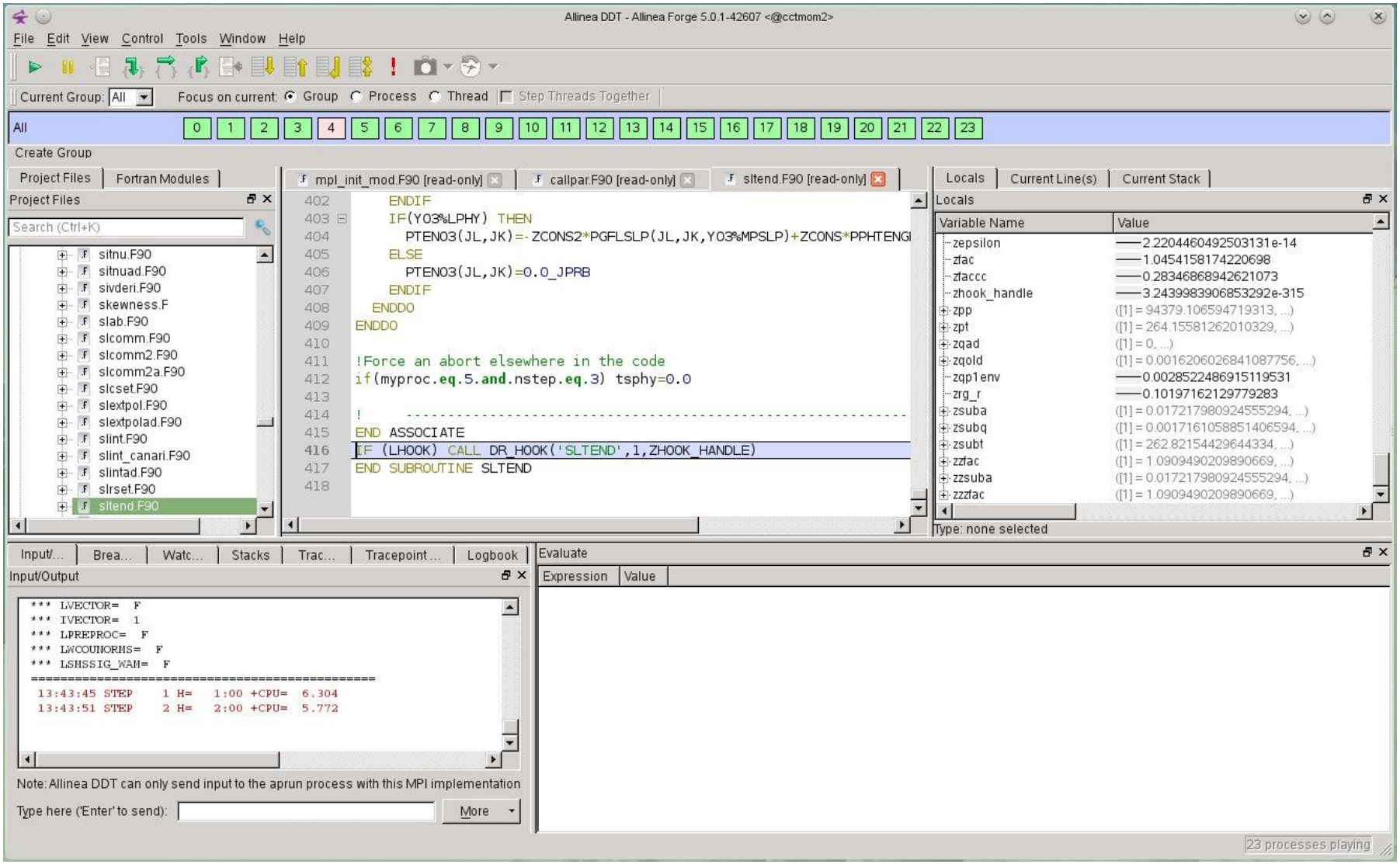
After reaching a breakpoint in callpar.F90 and setting a watch point on tsphy



After the watch point triggers in sltend.F90



Line 412 did it!



Allinea Map

- **A product that complements DDT**
- **A simple code profiler for MPI application**
- **Version 5 supports OpenMP as well**
- **Designed to be lightweight at scale**
  - **1000 samples per thread**
  - **Sampling rate adjust automatically**
- **Same source code browser as DDT**
- **Find hot spots in the application**

Using Allinea Map

- **Compile with debug info**
  - **-G2 for Cray compiler, -g for Intel/Gnu**
- **Load the DDT module**
  - **module load ddt/5.0.1.3_42607**
- **Execute make_profiler_libraries script**
  - **Creates libmap-sampler.so libmap-sampler-pmpi.so**
- **Link libraries into application**
  - **-lmap-sampler –lmap-sampler-pmpi**
- **Run in batch to collect data via map --profile**
  - **Creates a “profiling file”**
- **Analyse interactively via map “profiling file”**

Using Allinea Map in batch

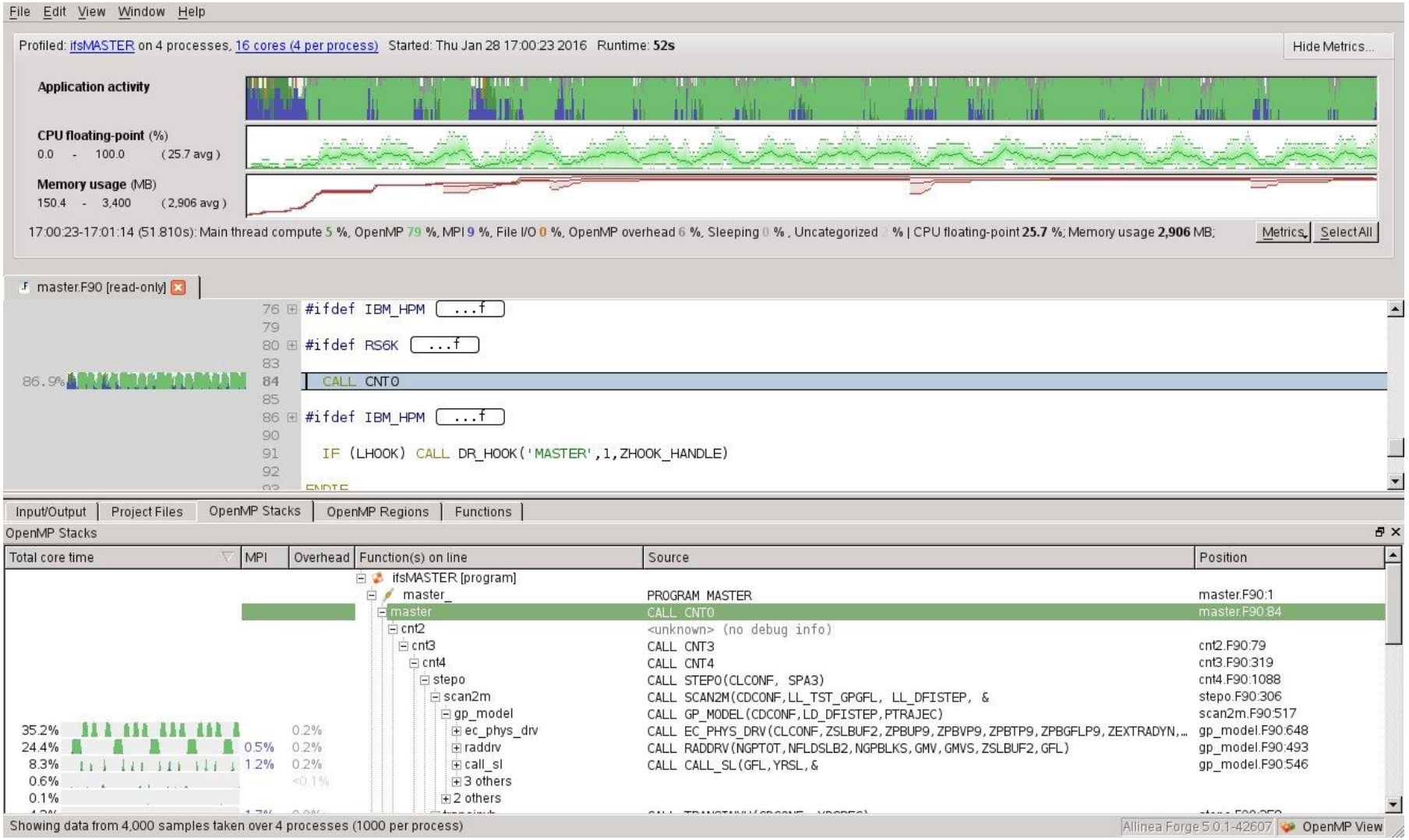
module load ddt/5.0.1.3_42607

map --profile –n ? –mpiargs ‘more aprun args’ a.out

For example

map --profile –n 4 –mpiargs ‘–N 4 –ss –cc cpu –d 6’ a.out

Allinea Map - Screenshot



Using Allinea Map

- **See the DDT/MAP userguide for more information**
  - **/opt/cray/ddt/4.2.1.2_36484/doc/userguide.pdf**
  - **/opt/cray/ddt/5.0.1.3_42607/doc/userguide-forge.pdf**

Very Simple DDT Example

tar –xzf ~trx/exercises/debug.tgz
cd Debug
compit
edit job.ddt and set DISPLAY
qsub job.ddt
Try setting

- a breakpoint at line 26
- a watch point on nthreads
- a conditional trace point on line 50 for nstep 50

Find the error output and trace output when it fails