

Debugging and Profiling Lab

Carlos Rosales, Kent Milfeld and Yaakoub Y. El Kharma
carlos@tacc.utexas.edu

Setup

- Login to Ranger:
 - `ssh -X username@ranger.tacc.utexas.edu`
- Make sure you can export graphics to your laptop screen:
 - `xclock`

If you do not see a clock, contact an instructor



- Untar the lab files:
 - `cd`
 - `tar xvf ~train00/dbg_prof_2010.tar`
- Change directories and ls to see the files:
 - `cd dbg_prof_2010`
 - `ls`

Overview

labs you should REALLY do

- DDT Lab
- IPM Lab
- PerfExpert Lab

optional labs

- mpiP Lab
- Tau Lab

DEBUGGING LAB

Finding a deadlock with DDT

- In this example we will use **DDT** to debug a code that deadlocks.

- Compile the deadlock example:

```
% cd debug
```

```
% mpicc -g -O0 ./deadlock.c
```

- Load the DDT module:

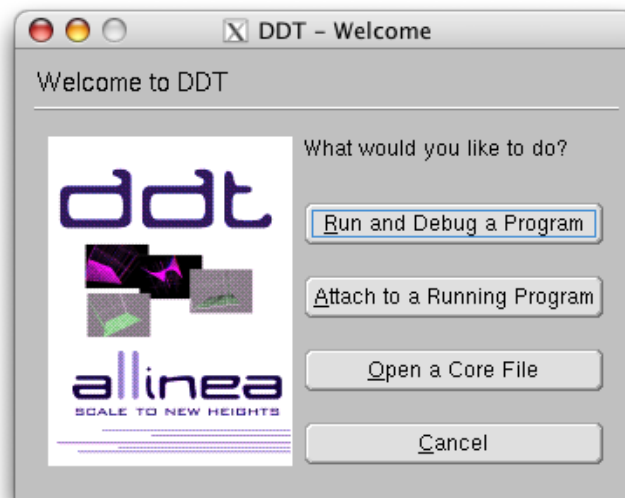
```
% module load ddt
```

- Start up DDT:

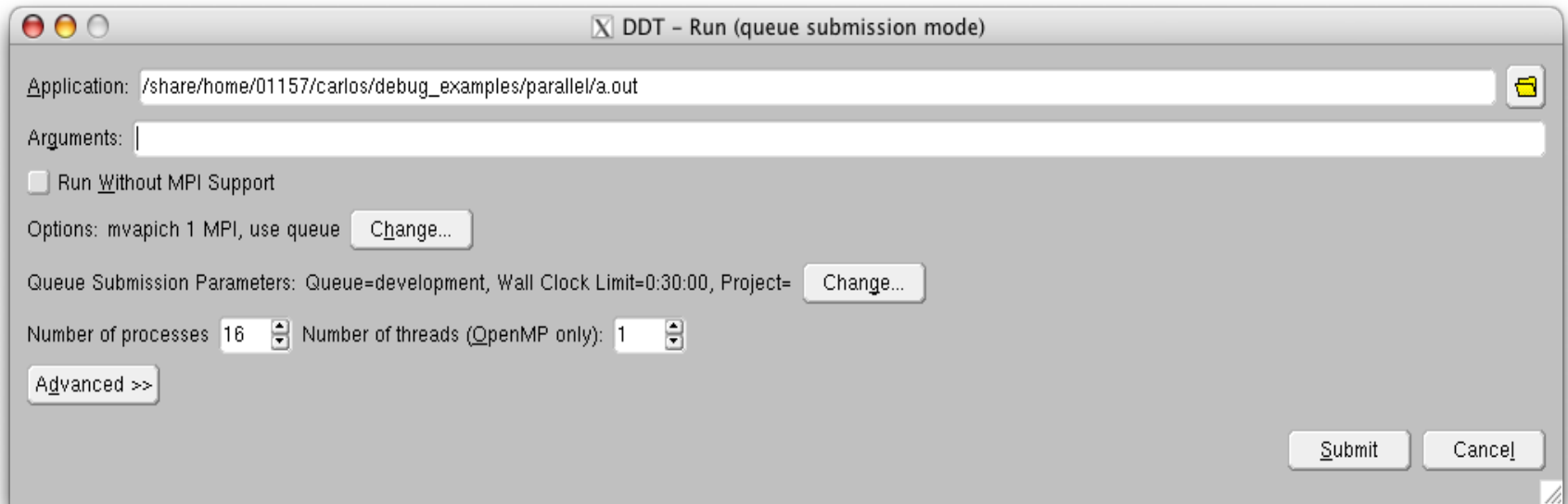
```
% ddt ./a.out
```

Configure DDT: Welcome

When you see the welcome screen below click the button that says “Run and Debug a Program”.



Configure DDT: Job Submission

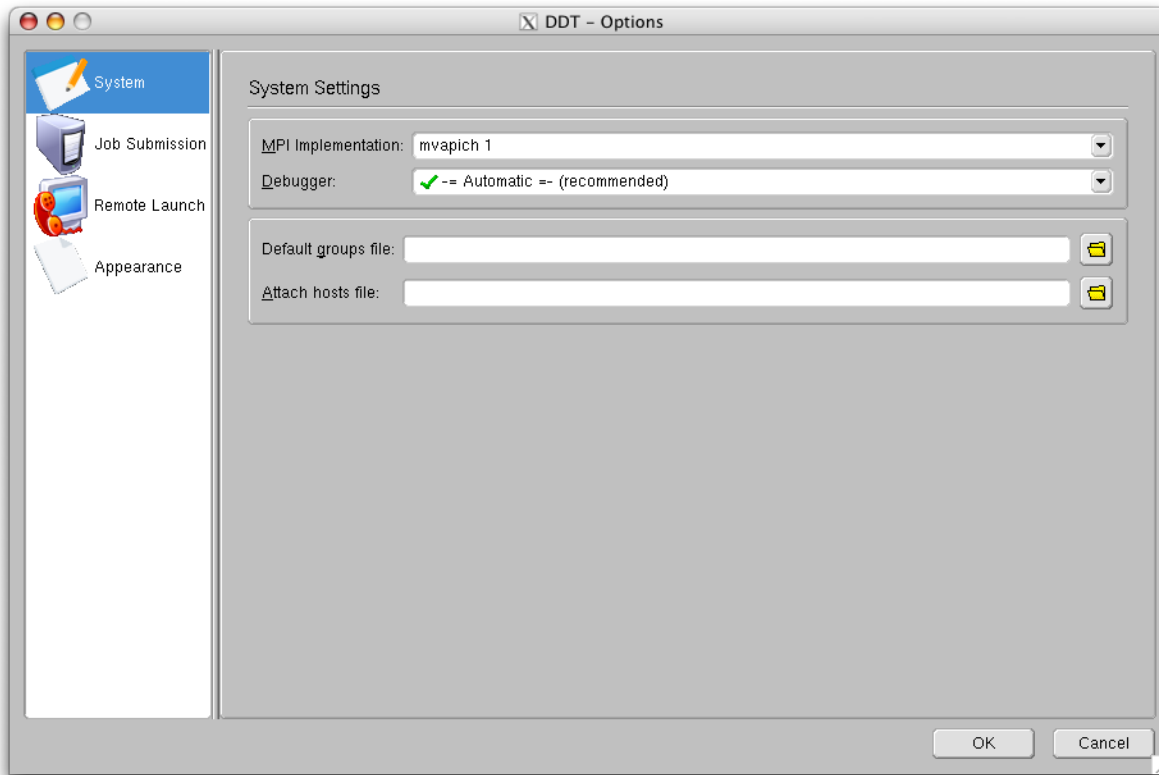


Don't click submit yet! We need to configure:

- General Options
- Queue Submission Parameters
- Processor and thread number
- Advanced Options

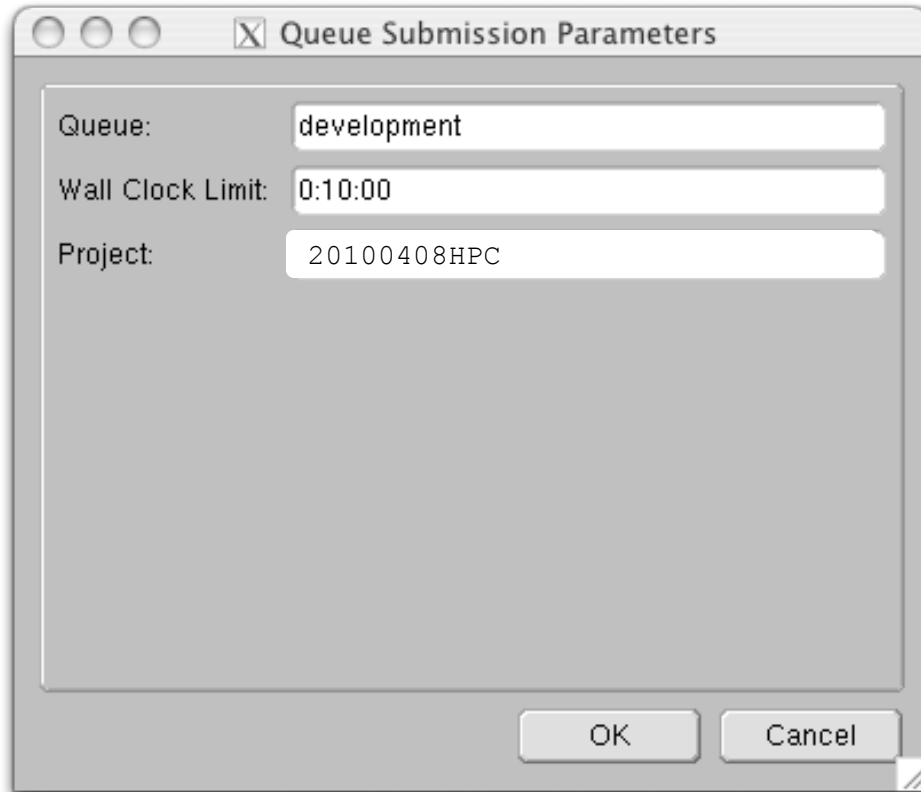
Click on Options -> Change

Configure DDT: Options



- Choose the correct version of MPI
 - mvapich 1
 - mvapich 2
 - openMPI
- Leave the default MPI (mvapich 1)
- Leave Debugger on the Automatic setting

Configure DDT: Queue Parameters



Queue Submission Parameters

Queue: development

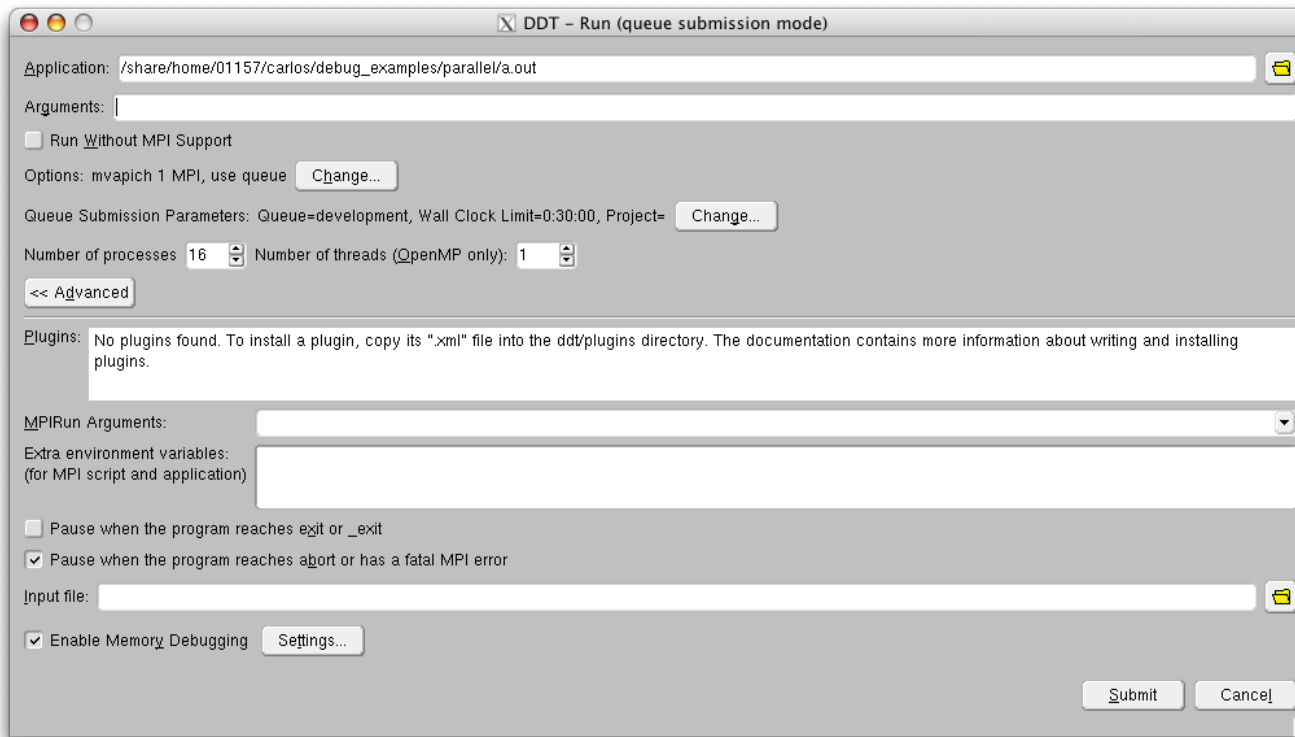
Wall Clock Limit: 0:10:00

Project: 20100408HPC

OK Cancel

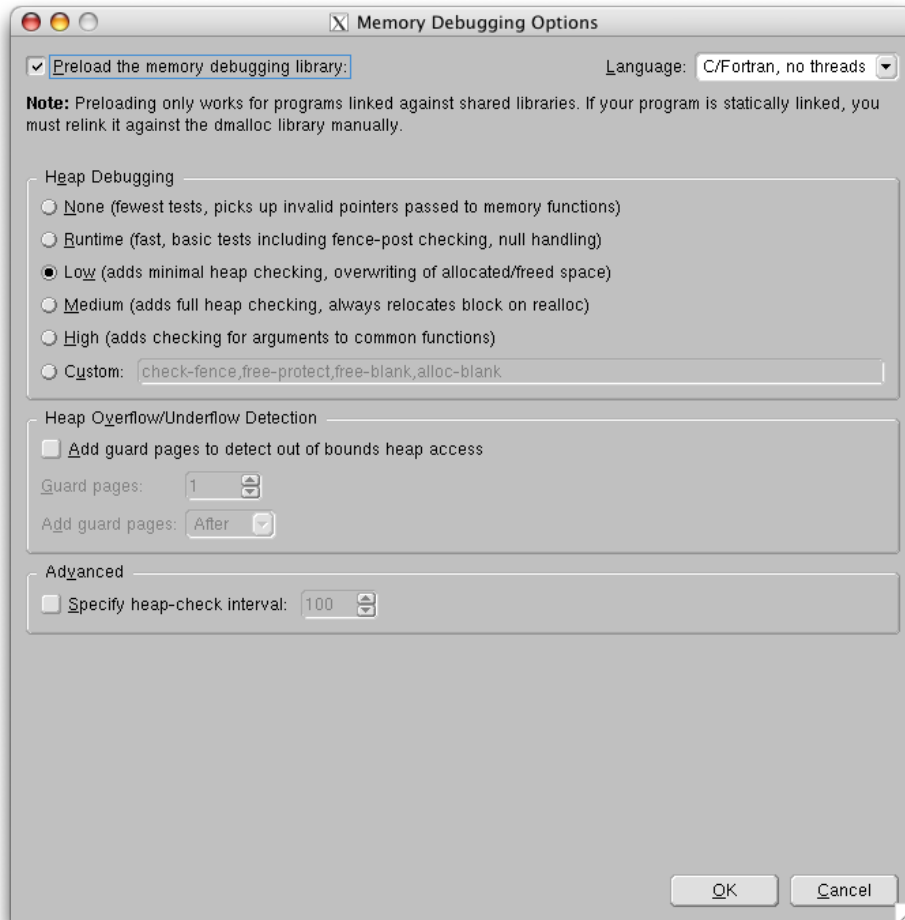
- Choose the “development” queue
- Set the Wall Clock Limit to 10 minutes (H:MM:SS)
- Set your project code - for this training class use 20100408HPC
- Click OK and double check that you have selected 16 CPUs / 1 thread in the main Job Submission window.

Configure DDT: Memory Checks



- Open the Advanced tab.
- Enable Memory Debugging (bottom left check box)
- Open the Memory Debug Settings

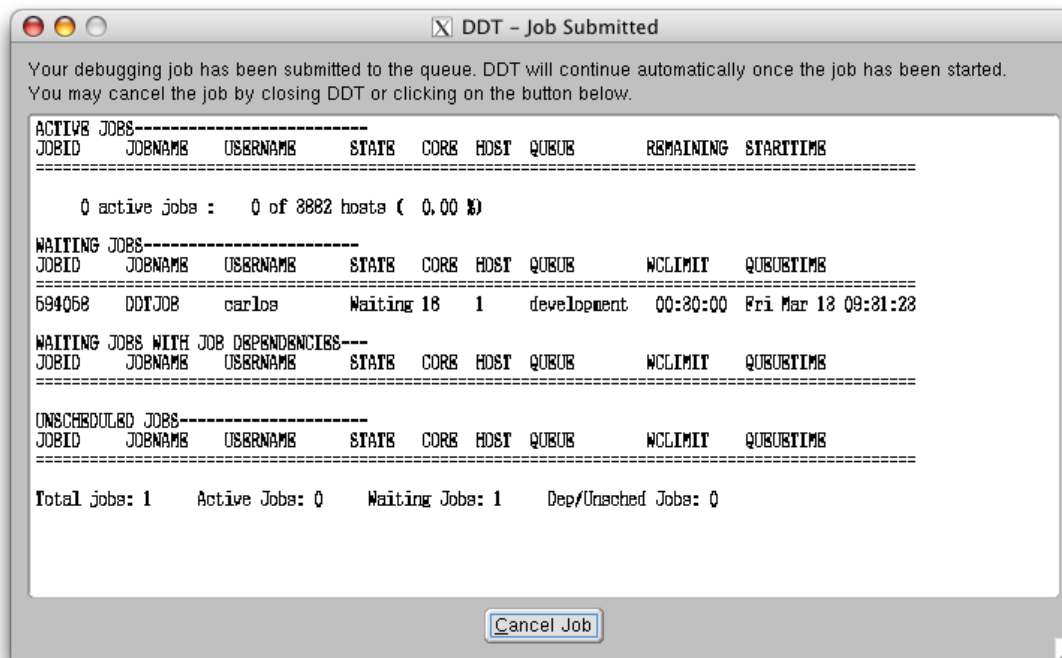
Configure DDT: Memory Options



- Change the Heap Debugging option from the default **Runtime** to **Low**
- Even the option None provides some memory checking
- Leave Heap and Advanced unchecked

DDT: Job Queuing

Add any necessary arguments to the program (none for the example)
Click the Submit button. A new window will open:

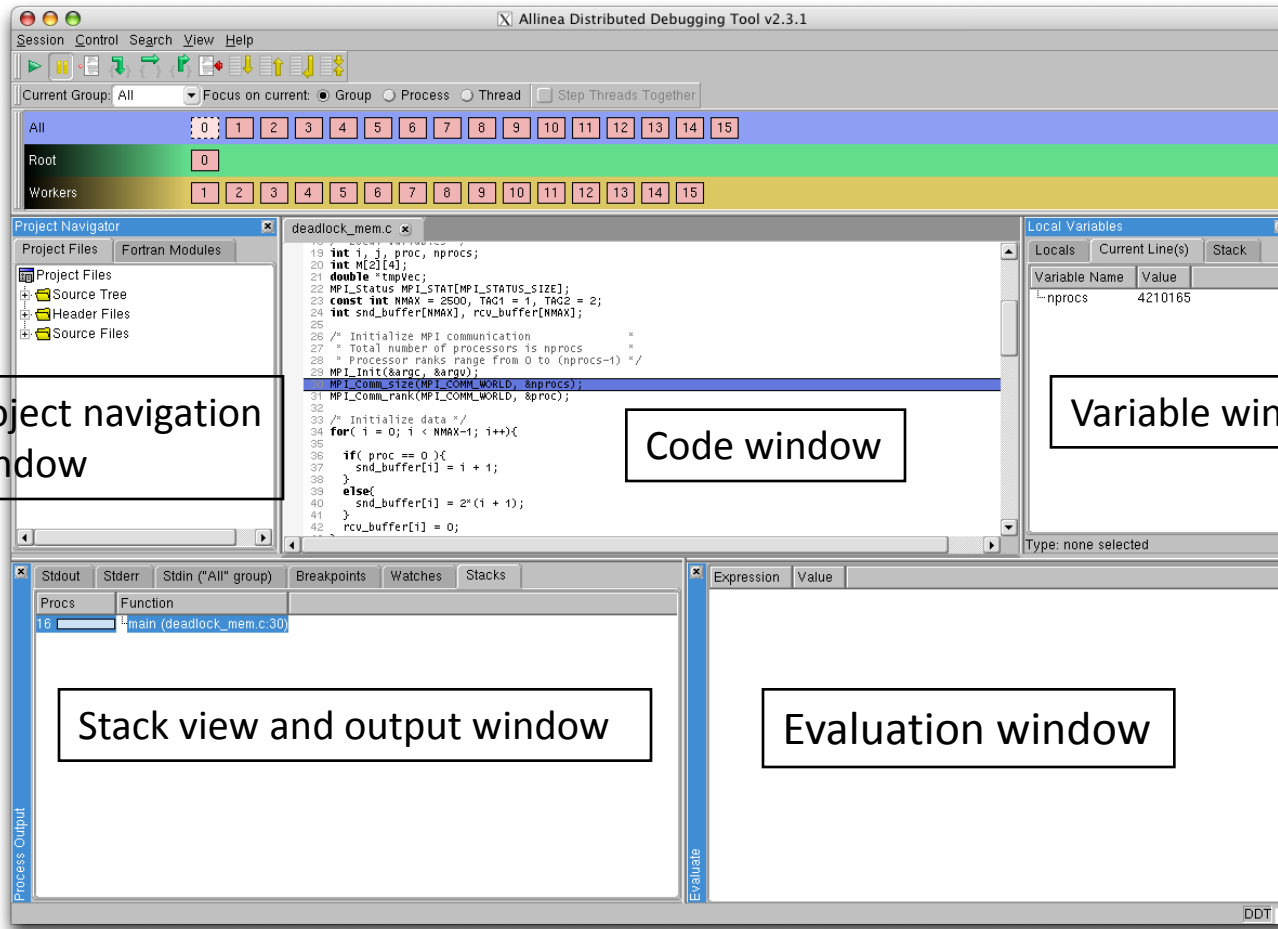


The job is submitted to the specified queue.

An automatically refreshing job status window appears.

The debug session will begin when the job starts.

DDT: The debug session



← Process controls

← Process groups window

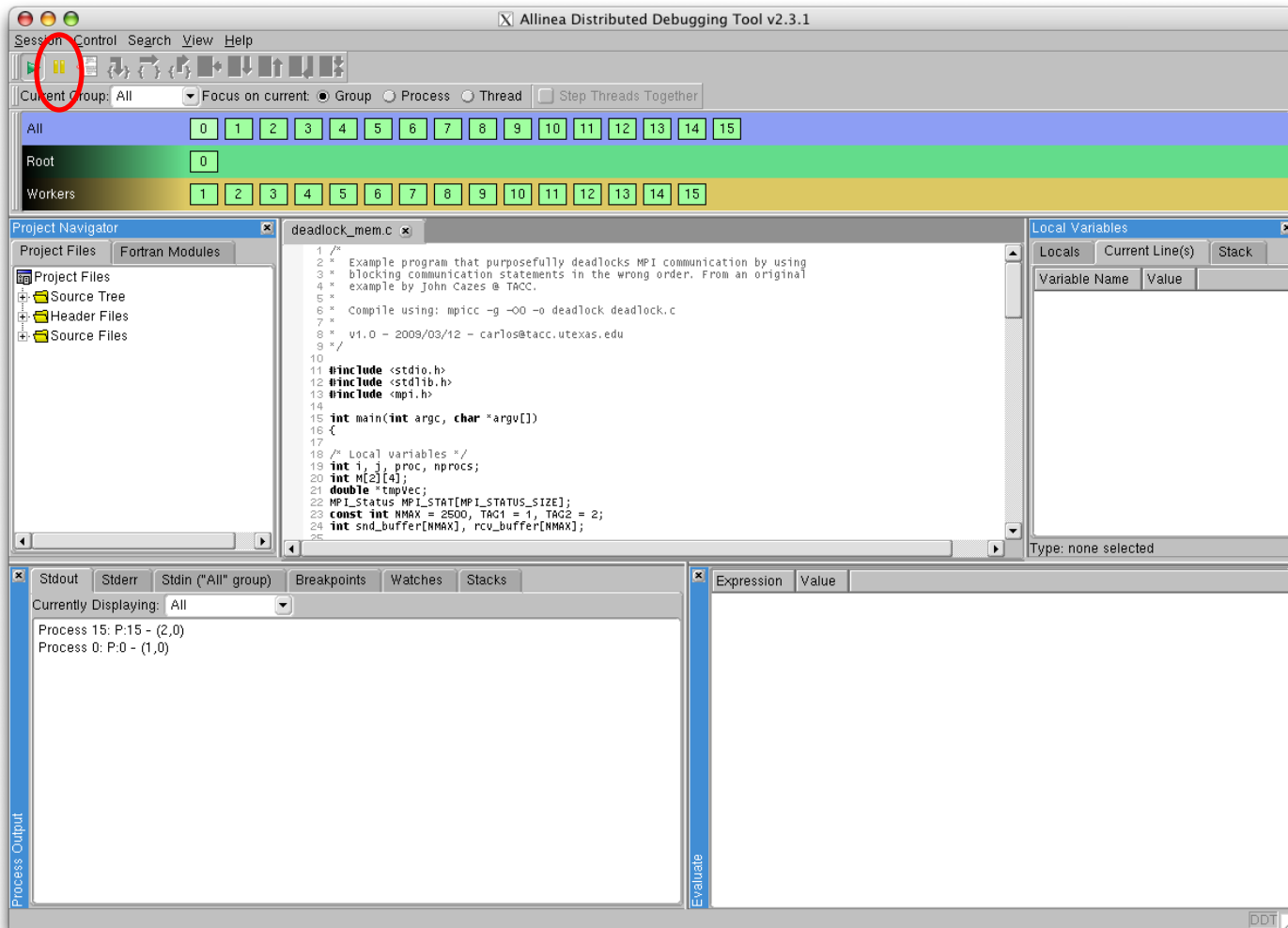
Project navigation window

Variable window

Stack view and output window

Evaluation window

DDT: Program Hangs

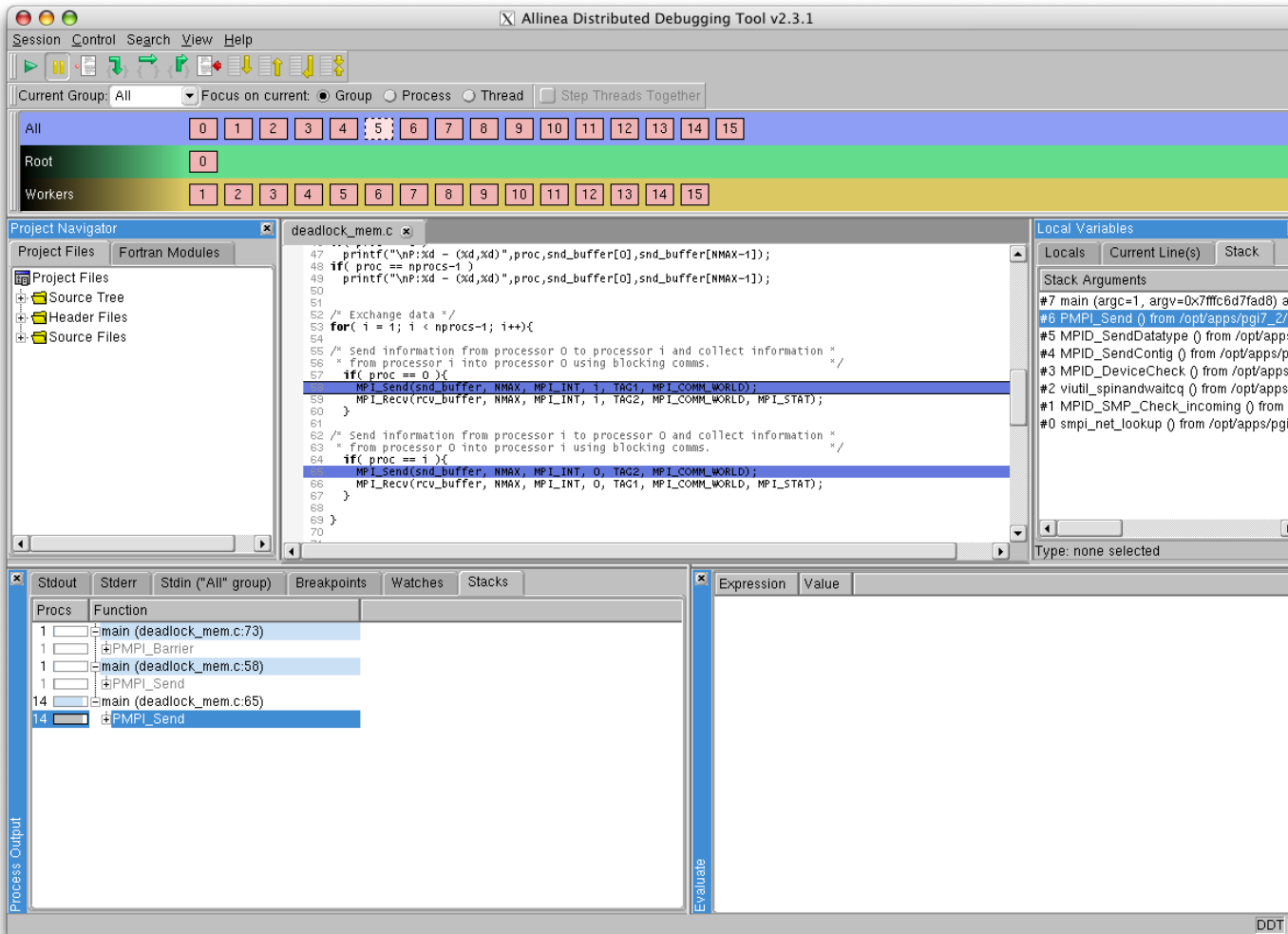


The output we expect does not appear in the Stdout window.

No active communication between procs.

Stop execution to analyze the program status (top left).

DDT: Stacks

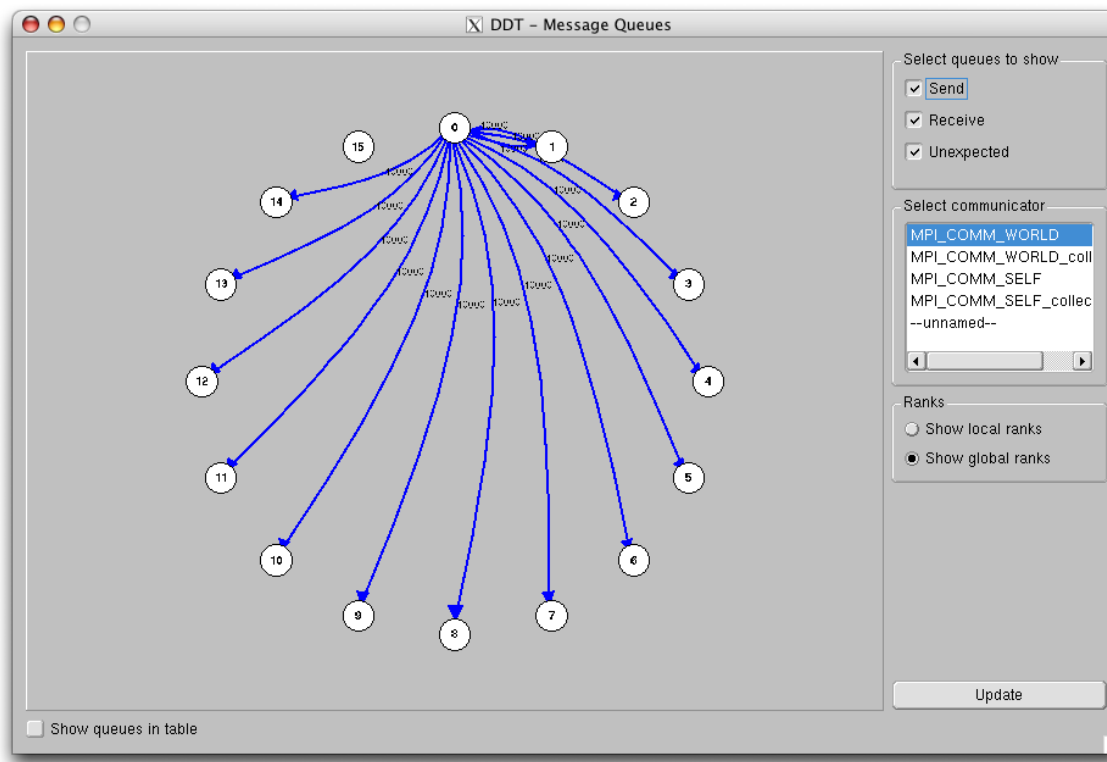


On the bottom left window select the Stacks view.

All processors seem to be stuck on a `MPI_Send()`.

DDT: Message Queues

Go to View -> Message Queues



There are uncompleted Send messages everywhere!

You can double-check that all communications are in the “Unexpected queue” (select on top right)

This is characteristic of a deadlock.

Find the source of the deadlock in the code.