

# Debugging Parallel Applications

Lars Koesterke

University of Porto, Portugal

May 28-29, 2009



THE UNIVERSITY OF TEXAS AT AUSTIN  
**Texas Advanced Computing Center**



# Setting Up DDT on Ranger(1)

- Logon to ranger  
your-desktop\$ ssh -X login3.tacc.utexas.edu
- Copy the test case to your directory  
login3\$ cp ~train00/ddt\_debug/debug\_code.f ./
- Compile the code with -g -O0  
login3\$ mpif90 -g -O0 debug\_code.f

# Setting Up DDT on Ranger(2)

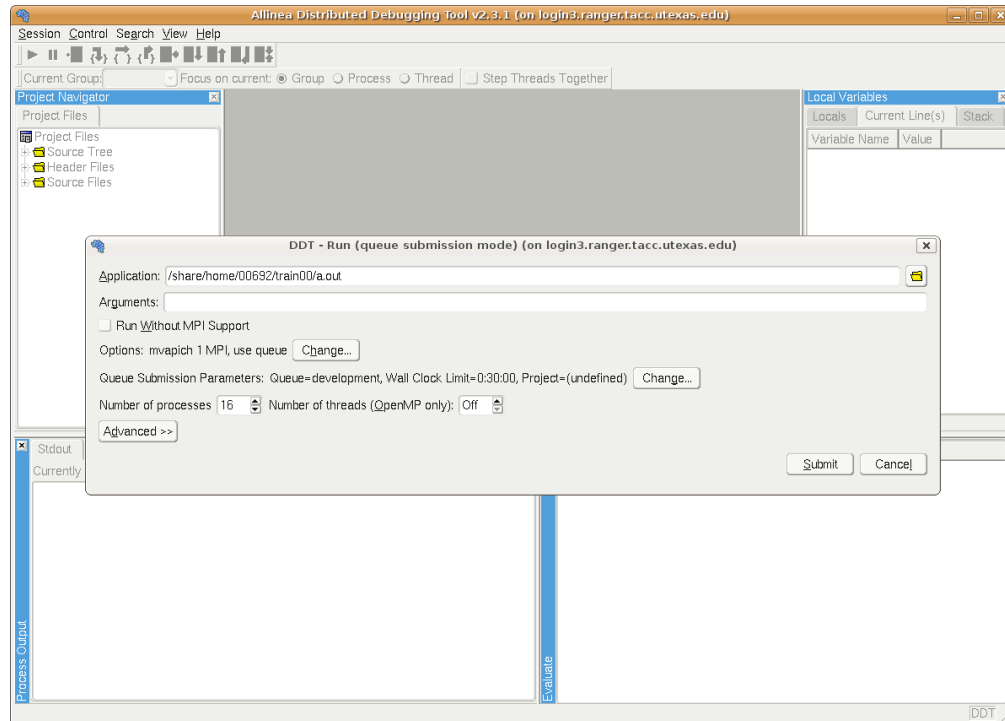
- Load the ddt module file:  
login3\$ module load ddt
- Create a .ddt directory in your \$HOME directory  
login3\$ cd  
login3\$ mkdir .ddt
- Copy \$DDTROOT/templates/config.ddt  
login3\$ cd .ddt  
login3\$ cp \$DDTROOT/templates/config.ddt ./

# Setting Up DDT on Ranger(3)

- Start the DDT debugger

```
login3$ cd ../
```

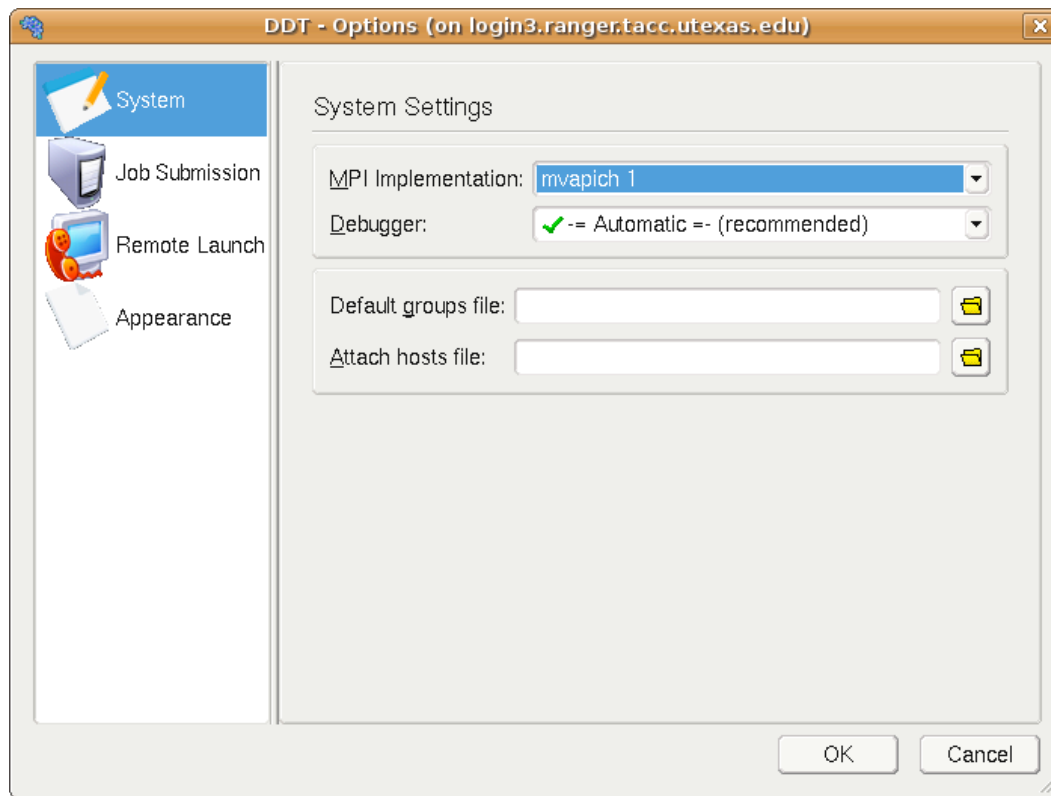
```
login3$ ddt ./a.out
```



# Configuring DDT(1)

Set up your job for submission.

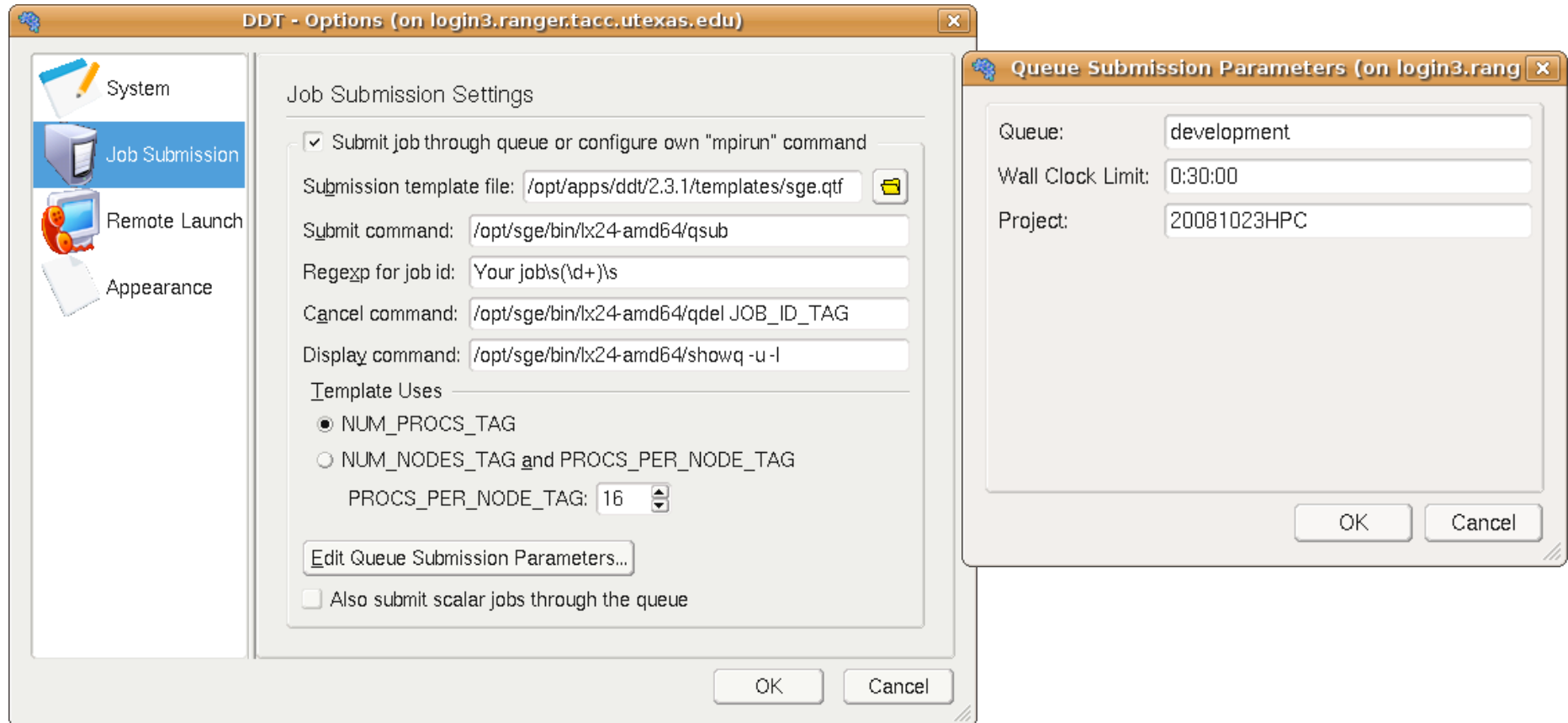
- Choose the correct version of MPI



mvapich 1  
mvapich 2  
OpenMPI

# Configuring DDT(2)

Set the correct queue submission parameters.



# Configuring DDT(3)

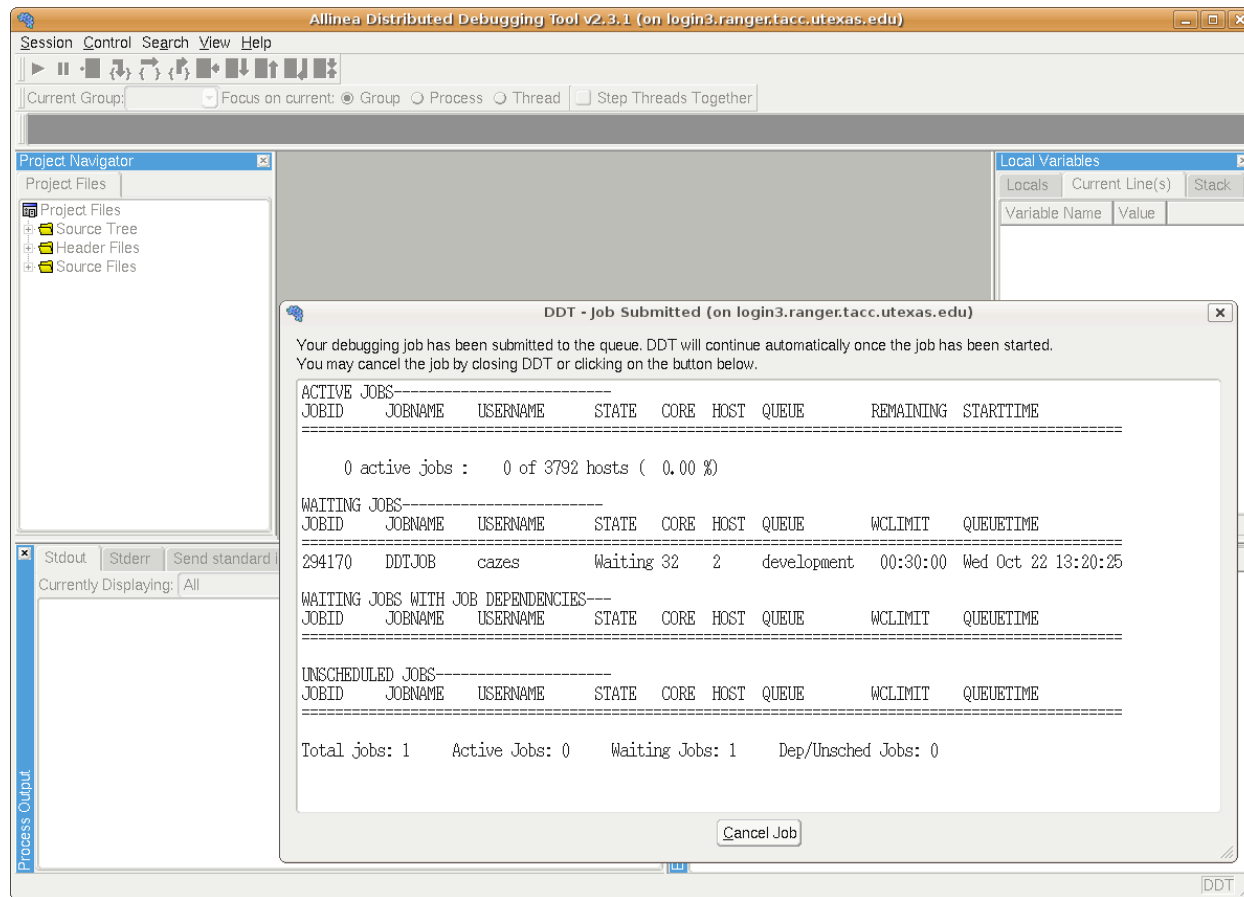
Submit your ddt run.

- Add any command line arguments needed or input files to the appropriate boxes.
- Set the number of tasks.
  - For Ranger this must be a multiple of 16.
- Press the submit button.

DDT will submit your job to the normal queue and the session will begin once your job has started.

# Configuring DDT(3)

Submit your ddt run.





# Debug code

## Set breakpoints

The screenshot displays the Allinea Distributed Debugging Tool (DDT) interface. At the top, the title bar reads "Allinea Distributed Debugging Tool (on login3.ranger.tacc.utexas.edu)". The main window is divided into several panes:

- Top Panel:** Contains session control buttons (run, stop, step, etc.) and a "Current Group" dropdown set to "All". Below this is a progress bar with 16 numbered steps (0-15) for "All", "Root", and "Workers".
- Left Panel:** A "Project Files" tree view showing a hierarchy of "Source Tree", "Header Files", and "Source Files".
- Center Panel:** A code editor showing Fortran code from lines 54 to 83. A red circle indicates a breakpoint is set at line 73: `call mpi_barrier(mpi_comm_world, mpierr)`. The code includes MPI-related operations like `mpi_send`, `mpi_recv`, and `mpi_barrier`.
- Right Panel:** A "Locals" window showing the current state of variables: `npes` with value 0 and `tmppe` with value -1729611376.
- Bottom Panel:** A "Breakpoints" window showing a list of active breakpoints. One breakpoint is listed at line 14 of `debug_code.f14`, with the function `deadlock`.

# Debug code

Run the code

The screenshot displays the Allinea Distributed Debugging Tool (DDT) interface. At the top, the title bar reads "Allinea Distributed Debugging Tool (on login3.ranger.tacc.utexas.edu)". Below the title bar is a menu bar with "Session", "Control", "Search", "View", and "Help". A toolbar contains various icons for running and debugging. The main window is divided into several panes:

- Process Monitor:** Located at the top, it shows a tree view with "All", "Root", and "Workers". Each node has a row of 16 colored boxes (0-15) representing individual processes or threads.
- Code Editor:** The central pane shows Fortran code from a file named "debug\_code.f". The code includes MPI-related operations such as `call mpi_send`, `call mpi_recv`, `call mpi_barrier`, and `call mpi_finalize`. Line 73 is highlighted in blue, indicating the current execution point.
- Project Files:** A sidebar on the left shows a tree view of the project structure, including "Project Files", "Source Tree", "Header Files", and "Source Files".
- Locals/Stacks:** A pane on the right is currently empty, with tabs for "Locals", "Current Line(s)", and "Stack".
- Process Monitor (Bottom):** A pane at the bottom left shows a list of processes with columns for "Procs" and "Function". The "Function" column lists various system and user-level functions like `_libc_start_main` and `main`.

# Debug code

If the code hangs, halt the execution.

The screenshot displays the Allinea Distributed Debugging Tool (DDT) interface. At the top, the title bar reads "Allinea Distributed Debugging Tool (on login3.ranger.tacc.utexas.edu)". The main window is divided into several sections:

- Control Panel:** Located at the top, it includes a menu bar (Session, Control, Search, View, Help) and a toolbar with various execution and debugging icons. Below the toolbar, there are dropdown menus for "Current Group: All", "Focus on current: Group", "Process", and "Thread", along with a "Step Threads Together" button.
- Thread Monitor:** A horizontal bar below the control panel shows the execution state of threads. It is divided into "All" (blue), "Root" (green), and "Workers" (yellow) sections. Each section contains a row of numbered buttons (0-15) representing individual threads.
- Code Editor:** The central pane shows Fortran code from a file named "debug\_code.f". The code includes a loop over tasks, an MPI send operation on line 43, and an MPI receive operation on line 49. Line 43 is currently selected and highlighted in blue.
- Project Files:** A sidebar on the left shows a tree view of project files, including "Source Tree", "Header Files", and "Source Files".
- Locals and Stack:** On the right side, there are panels for "Locals", "Current Line(s)", and "Stack". The "Stack" panel shows a table with columns for "Variable Name" and "Value".
- Process Stack:** At the bottom left, a "Procs" panel shows a list of active processes and their current functions. The stack includes "start\_thread", "async\_thread", "ibv\_get\_async\_event", "start", "jlibc\_start\_main", "main", and several "deadlock" and "pmpi\_send\_" entries.
- Expression and Value:** At the bottom right, there is a panel for "Expression" and "Value".

# Debug code

If the code hangs, halt the execution.

The screenshot displays the Allinea Distributed Debugging Tool (DDT) interface. At the top, the title bar reads "Allinea Distributed Debugging Tool (on login3.ranger.tacc.utexas.edu)". The main window is divided into several sections:

- Control Panel:** Located at the top, it includes a menu bar (Session, Control, Search, View, Help) and a toolbar with various debugging icons. Below the toolbar, there are dropdown menus for "Current Group: All", "Focus on current:", "Group", "Process", "Thread", and a checkbox for "Step Threads Together".
- Process Tree:** A horizontal bar below the control panel shows the execution state of different components. "All" is highlighted in blue with a progress bar from 0 to 15. "Root" is highlighted in green with a progress bar from 0 to 0. "Workers" is highlighted in yellow with a progress bar from 1 to 15.
- Code Editor:** The central pane shows Fortran code from a file named "debug\_code.f". The code includes MPI-related operations such as `call mpi_recv`, `call mpi_send`, and error checking routines. Line 60 is currently selected and highlighted in blue.
- Locals and Stack:** On the right side, there are panels for "Locals" and "Stack", each with a table for "Variable Name" and "Value". The "Locals" panel is currently empty.
- Process List:** At the bottom left, a "Procs" table lists the execution stack for the current process. The stack includes `start_thread`, `async_thread`, `ibv_get_async_event`, `start`, `libc_start_main`, `main`, and several `deadlock` calls at lines 73, 59, and 42 of `debug_code.f`.
- Expression and Value:** At the bottom right, there is a table for "Expression" and "Value", which is currently empty.

# Fix the Code

Reorder the send and receive for task 0.

```
call mpi_recv(
```

```
+   b(1), 1000000, mpi_real, tmppe, 9902,
```

```
+   mpi_comm_world, mpierr
```

```
call mpi_send(
```

```
+   a(1), 1000000, mpi_real, tmppe, 9901,
```

```
+   mpi_comm_world, mpierr)
```

# Questions?