

Debugging on the ALCF XC40 System

Computational Performance Workshop

April 30, 2019

Ray Loy

ALCF

OUTLINE

- Interactive jobs
- Core dumps – ATP
- Snapshots - STAT
- Basic parallel debugging – lgdb
- DDT / MAP

Interactive runs for tests (Theta)

Submit an interactive job to the queue, e.g.

– `qsub -l -t 30 -n 512`

When job "runs", the nodes are allocated, and you get a (new) shell prompt on a "mom" node.

– This shell behaves like the one in a Cobalt script job

– Start your compute node run just like you would in a Cobalt script job.

- Theta: `aprun -N 64 -d 1 -j 1 -cc depth myprogram.exe`
 - Or just run your Cobalt job script: `./myjobscript.sh`

When you exit the shell, the Cobalt job will end

Note: When the Cobalt job runs out of time, there is no message.

– Telltale sign: *aprun will fail*

– Check your job status with "`qstat $COBALT_JOBID`"

Theta: ATP

ATP = Abnormal Termination Processing

- generates a STAT format merged stack backtrace (file `atpMergedBT.dot`)
- view the backtrace file with **stat-view** (*module load stat*)

Link your app with ATP

- Before linking, check that the "atp" module is loaded (*module list*)
- Cray and Intel compilers will link in ATP automatically

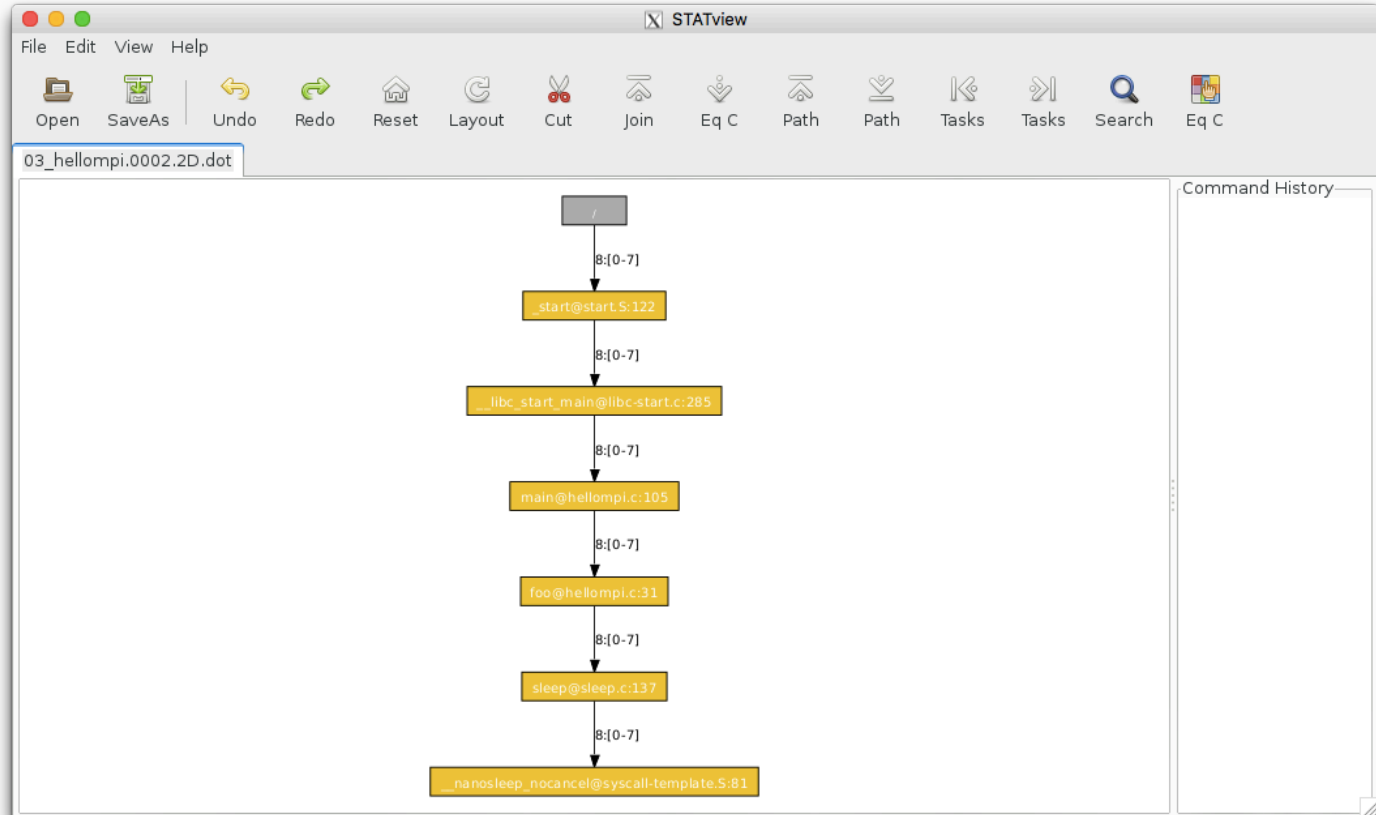
In your job script, set environment before running your app

- `export ATP_ENABLED=1`
- `aprun ...`

- If your program crashes, ATP will invoke STAT to dump a backtrace file

STAT-VIEW

module load stat



THETA: STAT snapshot

While program is running (e.g. deadlocked), you can generate a merged backtrace snapshot showing where your program is.

On the MOM node, invoke "stat-cl *pid*" where *pid* is the aprun pid

In job script (or interactive job shell)

- hostname # identify the MOM node you are on
- module unload xalt # xalt wraps aprun resulting in 2 processes named "aprun"
- aprun ...

During the run, ssh to the same MOM node

- ps -u *username* # Determine pid of aprun
- module load stat
- DISPLAY="" stat-cl *pid*

Optional

- aprun ... &
- echo "aprun pid is \$!"
- wait

Igdb

Igdb connects a gdb to each rank and provides a text interface

module load cray-igdb

Modify your script job.sh to mark your aprun:

```
#cray_debug_start  
aprun -n 8 -N 1 -d 1 -j 1 a.out  
#cray_debug_end
```

Igdb

– launch \$a(8) --qsub=job.sh a.out

- Submits job.sh to run 8 ranks, your executable is a.out

Useful commands

– backtrace (bt), continue (cont), break, print

– See "man Igdb"

Allinea DDT and MAP

- Environment
 - Theta: module load forge/19.0.2 (*/soft/environment/modules/modulefiles*)
 - **Do not use** *module load forge, module load ddt*
- Compiling your code
 - Compile `-g -O0`
- More details:
 - <http://www.alcf.anl.gov/user-guides/allinea-ddt>

Allinea DDT startup

Option 1: Run using remote client (RECOMMENDED)

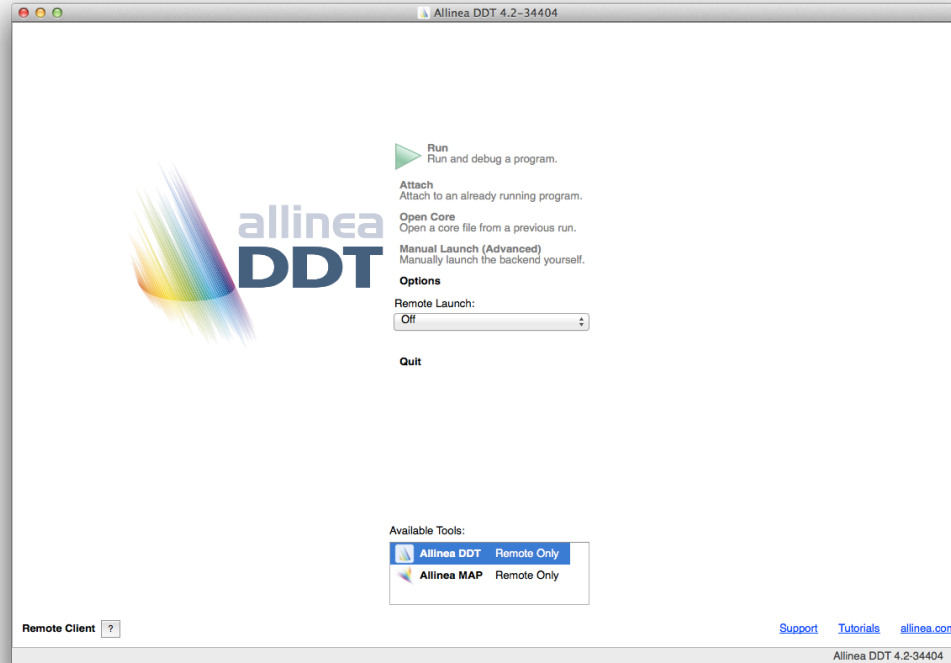
- Download and install Mac or Windows "Remote client" from
 - <https://developer.arm.com/tools-and-software/server-and-hpc/arm-architecture-tools/downloads/download-arm-forge>
- Optional: use ssh master mode so you only need log in once per session
 - Note: supported on Mac OS/X; not supported in Windows <= XP (? for >XP)
 - `~/.ssh/config`
 - `ControlMaster auto`
 - `ControlPath ~/.ssh/master-%r@%h:%p`

Option 2: Run from login node

- Need X11 server on your laptop and ssh `-X` forwarding
- Run ddt and let it submit job through GUI

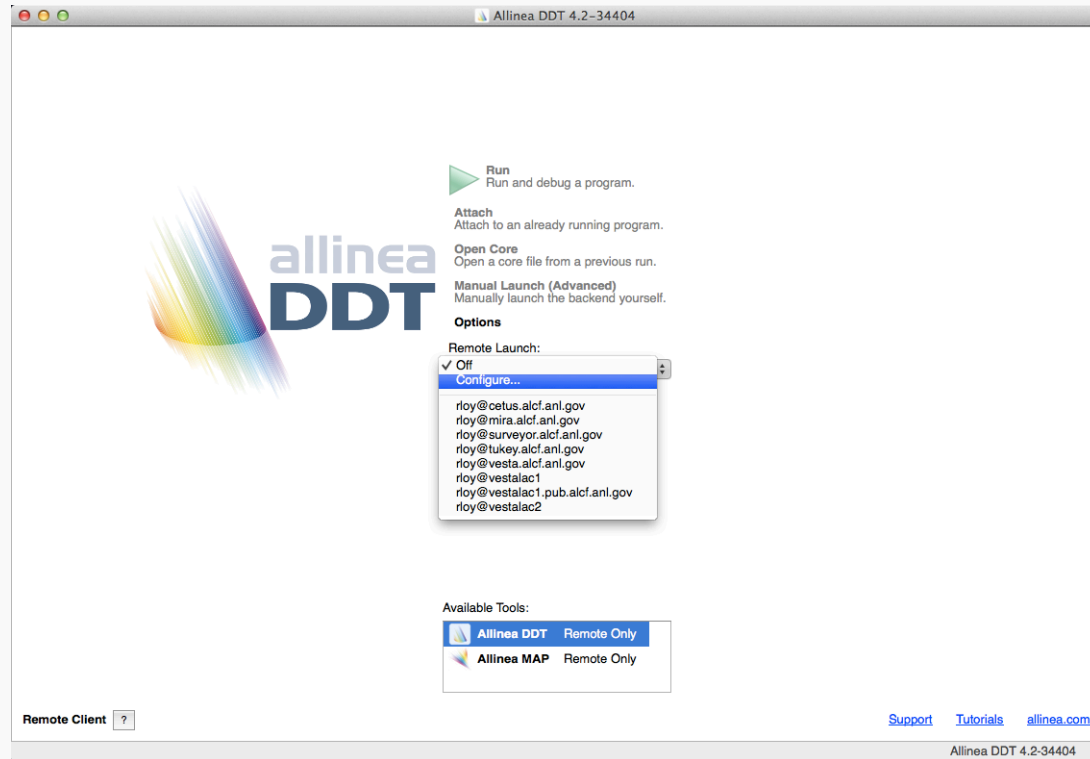
DDT Remote Client (0)

GUI looks just like the X11 Client



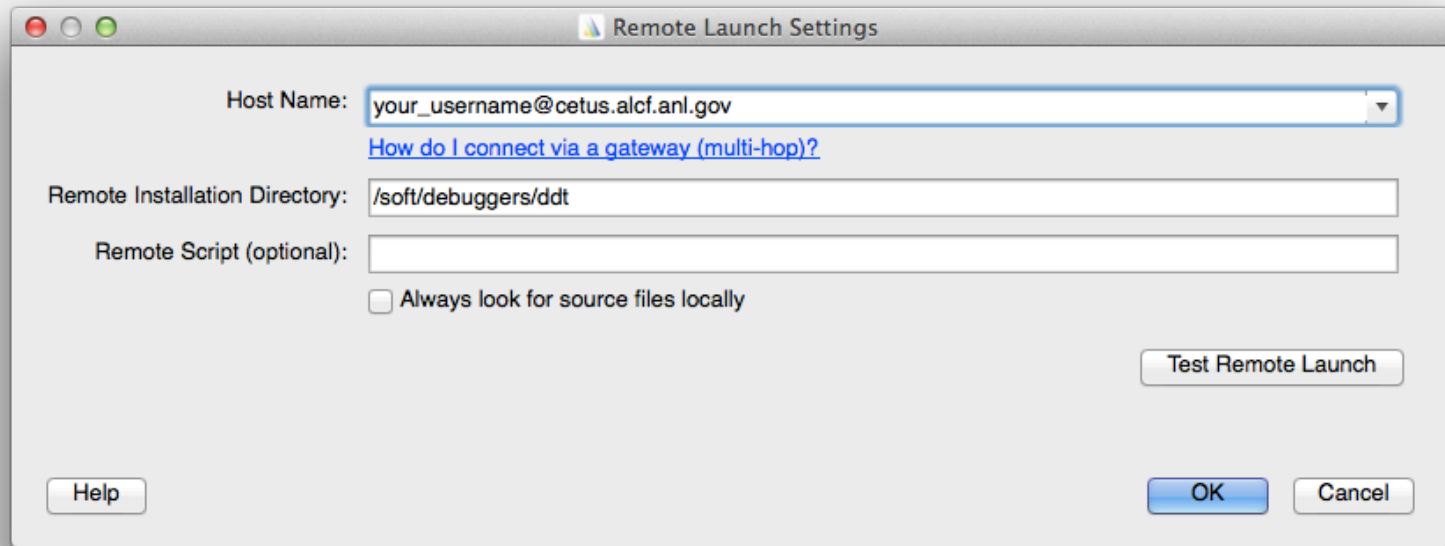
DDT Remote Client (1)

Select "configure" to add a new remote host



DDT Remote Client (2)

Note: this remote installation directory is the default version of DDT, corresponding to +ddt or module
Click "Test Remote Launch" to verify

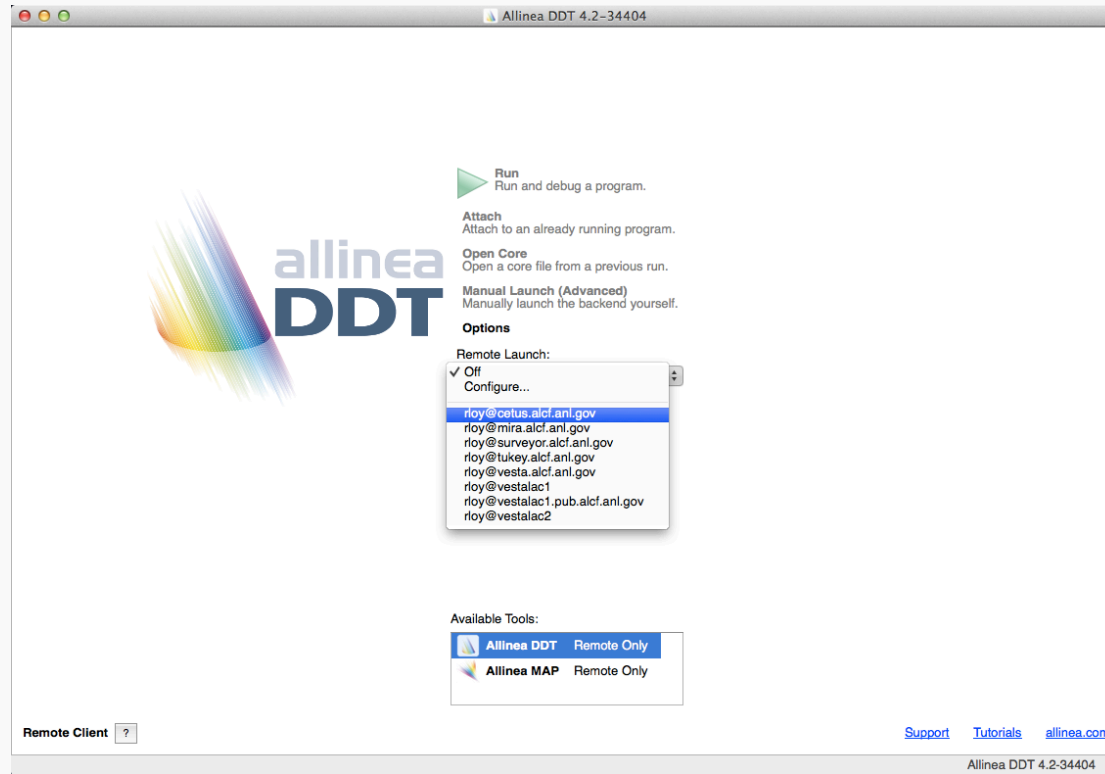


The screenshot shows a macOS-style dialog box titled "Remote Launch Settings". It contains the following fields and controls:

- Host Name:** A text field containing "your_username@cetus.alcf.anl.gov". Below it is a blue hyperlink: [How do I connect via a gateway \(multi-hop\)?](#)
- Remote Installation Directory:** A text field containing "/soft/debuggers/ddt".
- Remote Script (optional):** An empty text field.
- Always look for source files locally
- Buttons:** "Help" (bottom left), "Test Remote Launch" (right side), "OK" (bottom right), and "Cancel" (bottom right).

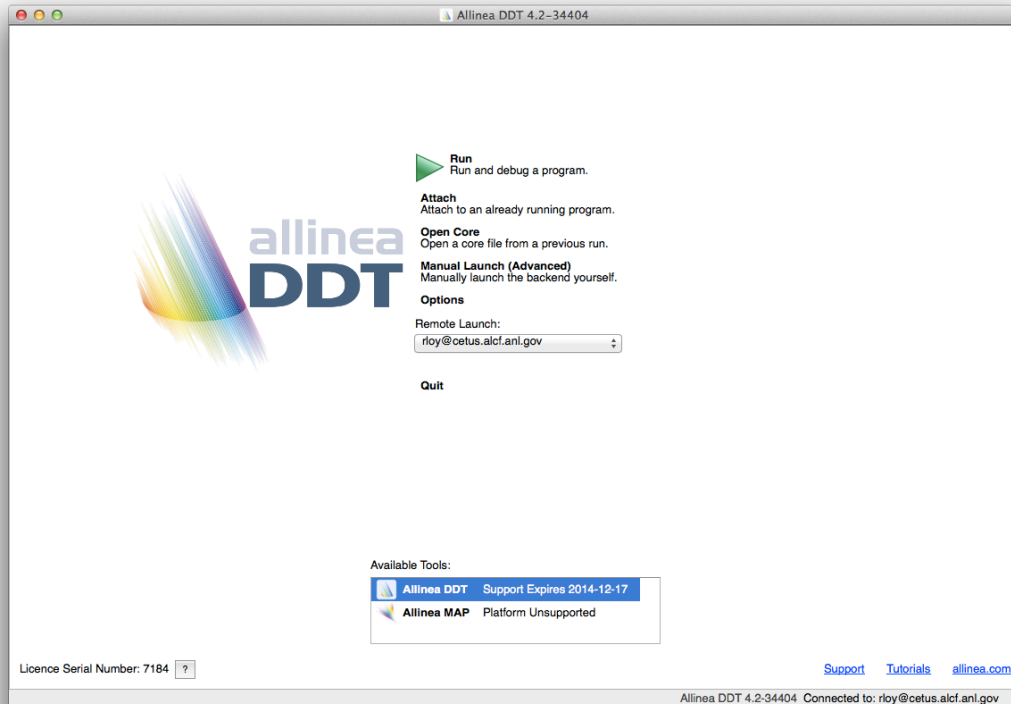
DDT Remote Client (3)

Now that it is defined, select remote machine



DDT (4)

Connected (note License info in lower left corner)
From this point, remote GUI works same as local



DDT Startup – Reverse Connect

Start remote client and connect to login node (or start X11 client on login node)

In an ssh session to the login node

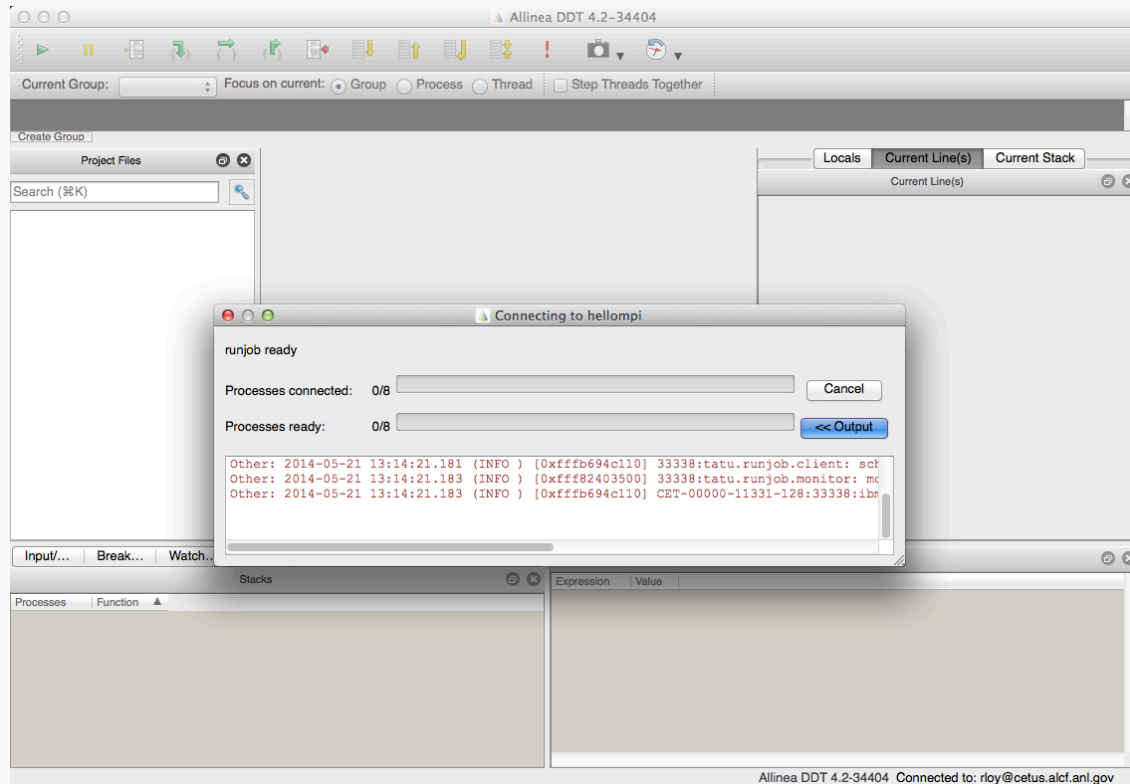
- Run an interactive job (qsub -l)
 - Theta: Instead of aprun ... myprog.exe
 - /soft/debuggers/forge/bin/ddt --connect aprun ... myprog.exe
 - Handy tip: run your job script from the interactive job command line

Likewise with Alinea MAP

- Theta: /soft/debuggers/forge/bin/map --connect aprun ... myprog.exe

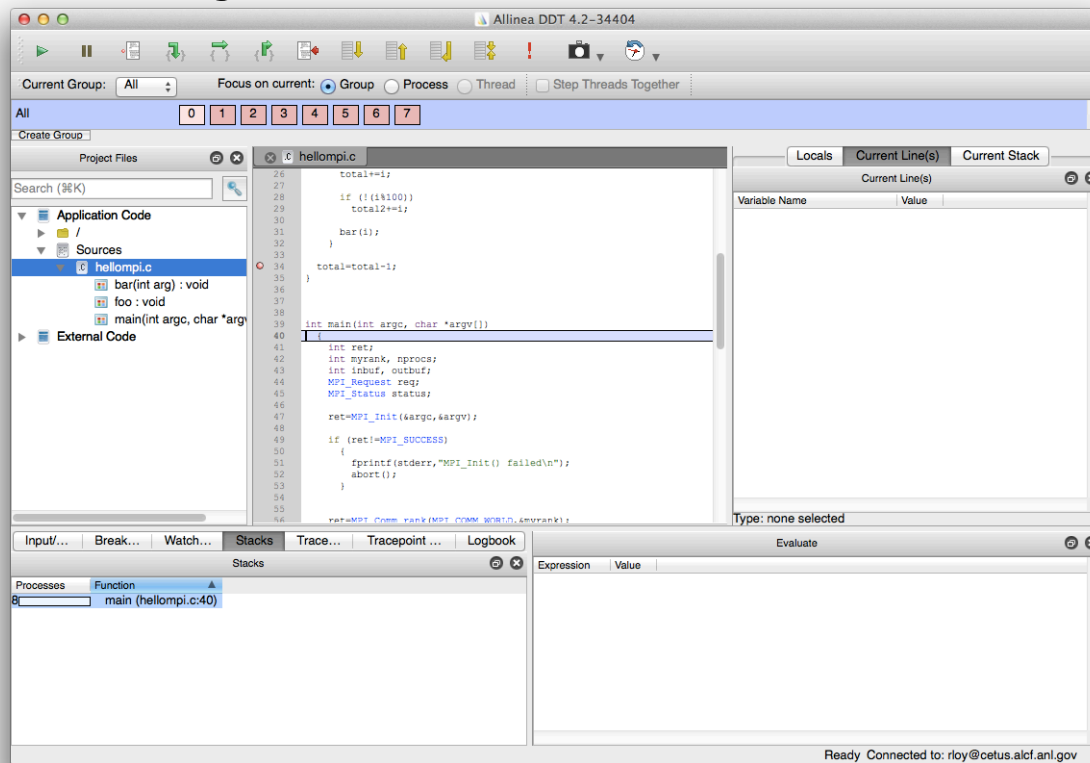
DDT

When job starts running, connection status will show



DDT

Ready to debug!



Questions

See also

– <http://www.alcf.anl.gov/user-guides>

– support@alcf.anl.gov

