

Debugging on the ALCF Theta and Cooley Systems

Computational Performance Workshop
May 6, 2020

Ray Loy
ALCF

OUTLINE

- Interactive jobs
- Core dumps - ATP
- Snapshots - STAT
- Starting up DDT / MAP

Interactive jobs for test runs (Theta, Cooley)

- Submit an interactive job to the queue, e.g.
 - `qsub -I -t 30 -n 4 -A Comp_Perf_Workshop -q queue_name`
- When the job "runs", the nodes are allocated, and you get a prompt from a new shell started on a "MOM" node (Theta) or job head node (Cooley)
 - This shell behaves like the one that executes a Cobalt script job
- Start your compute node run just like you would inside a Cobalt script job.
 - Theta: `aprun -N 64 -d 1 -j 1 -cc depth a.out`
 - Cooley: `mpirun -f $COBALT_NODE_FILE -n 12 a.out`
 - Or just run your Cobalt job script: `./myjobscript.sh`
- When you exit the shell, the Cobalt job will exit

Interactive jobs for test runs (Theta, Cooley)

- Note: When the Cobalt job runs out of time, the shell continues running.
 - There is no warning message
 - Telltale sign: *aprun will fail*
 - Check job status with "qstat \$COBALT_JOBID"

- Note: If you need X11 forwarding from your interactive job to display back to your laptop
 - Connect from your laptop to the login node using `ssh -Y`
 - Add the following to your `~/.ssh/config` on Theta/Cooley (*not on your laptop*):
 - `ForwardX11 yes`
 - `FowardX11Trusted yes`
 - Submit your interactive job

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

Example code that fails

```
void branch3(int myrank, int nprocs) {
    if (myrank>0)
        sleep(600);
    else {
        sleep(5);
        abort();
    }
}
```

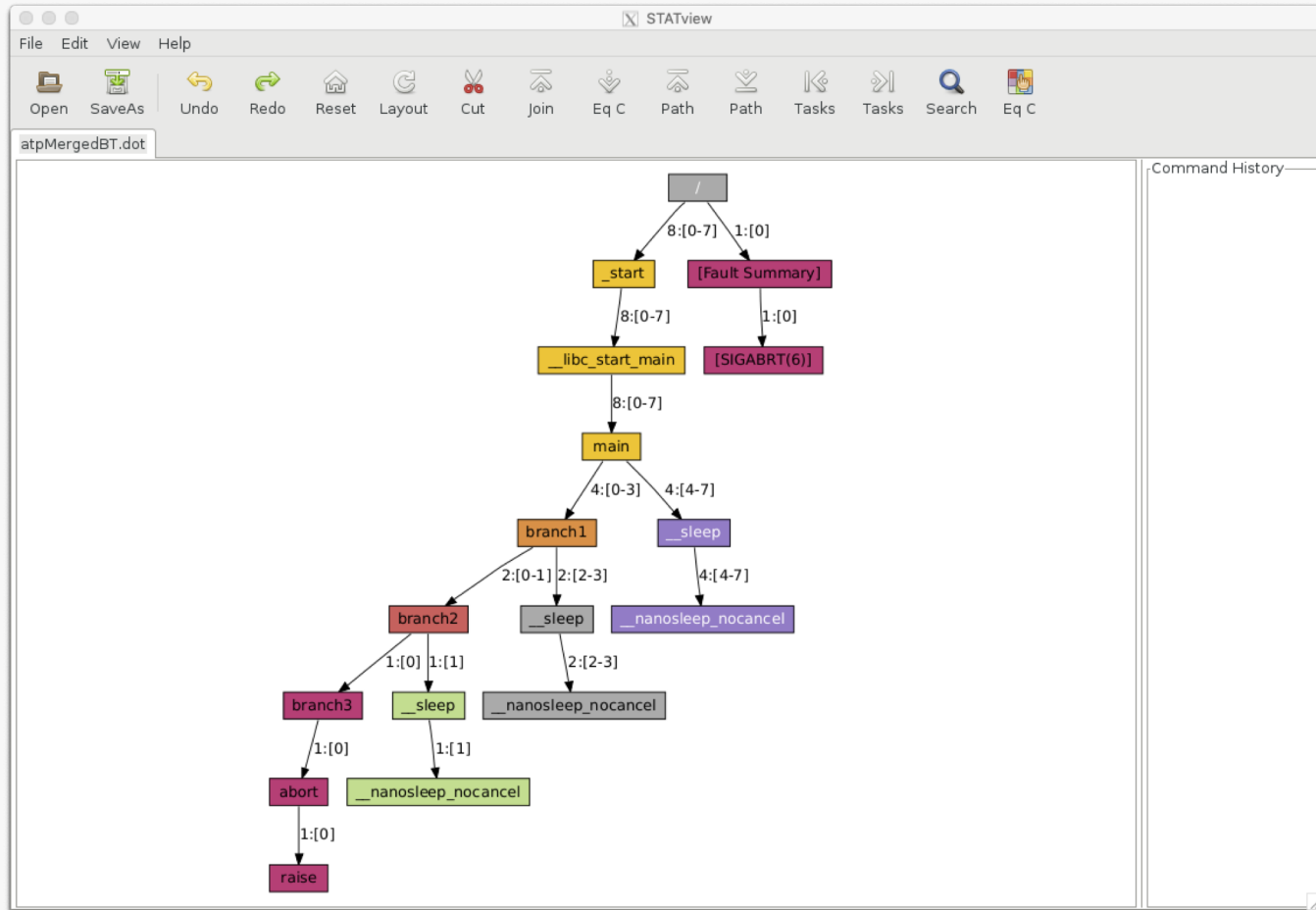
```
void branch2(int myrank,int nprocs) {
    if (myrank < nprocs/8)
        branch3(myrank, nprocs);
    else
        sleep(600);
}
```

```
void branch1(int myrank, int nprocs) {
    if (myrank < nprocs/4)
        branch2(myrank,nprocs);
    else
        sleep(600);
}
```

```
int main(int argc, char *argv[]) {
    ...
    if (myrank < nprocs/2)
        branch1(myrank,nprocs);
    else
        sleep(600);
    ...
}
```

STAT-VIEW

module load stat
stat-view atpMergedBT.dot



THETA: STAT snapshot

- While program is running (e.g. deadlocked), you can generate a merged backtrace snapshot showing where your program is.
- Summary: on the MOM node, invoke "stat-cl *pid*" where *pid* is the aprun pid
- In an interactive job:

```
hostname                # identify the MOM node you are on
module unload xalt      # xalt wraps aprun resulting in 2 processes named "aprun"
aprun ... &
echo "aprun pid is $!"

# wait until you think the program is deadlocked
module load stat
DISPLAY="" stat-cl pid
```
- Snapshot files are placed in the subdirectory stat_results e.g.
stat_results/stat-test.0000/

THETA: STAT snapshot (con't)

- Or in your job script:

```
hostname          # identify the MOM node you are on
module unload xalt # xalt wraps aprun resulting in 2 processes named "aprun"
aprun ... &
echo "aprun pid is $!"
wait
```

- During the run, ssh to the same MOM node

```
ps -u username # Alternate way to determine pid of aprun
module load stat
DISPLAY="" stat-cl pid
```

- Snapshot files are placed in the subdirectory `stat_results`, e.g.
`stat_results/stat-test.0000/`

Allinea DDT and MAP

- Environment
 - Theta: module load forge (/soft/environment/modules/modulefiles)
 - Cooley: Add key +ddt to ~/.soft.cooley (or **soft add +ddt**)
 - Other modules/keys available for specific versions
- Check version by typing **ddt --version**
 - ALCF is currently at version 19.1.2 (May 1, 2020)
- Compile with `-g -O0`
- See also:
<https://www.alcf.anl.gov/support-center/theta/arm-ddt-theta>

Aside: ssh ControlMaster

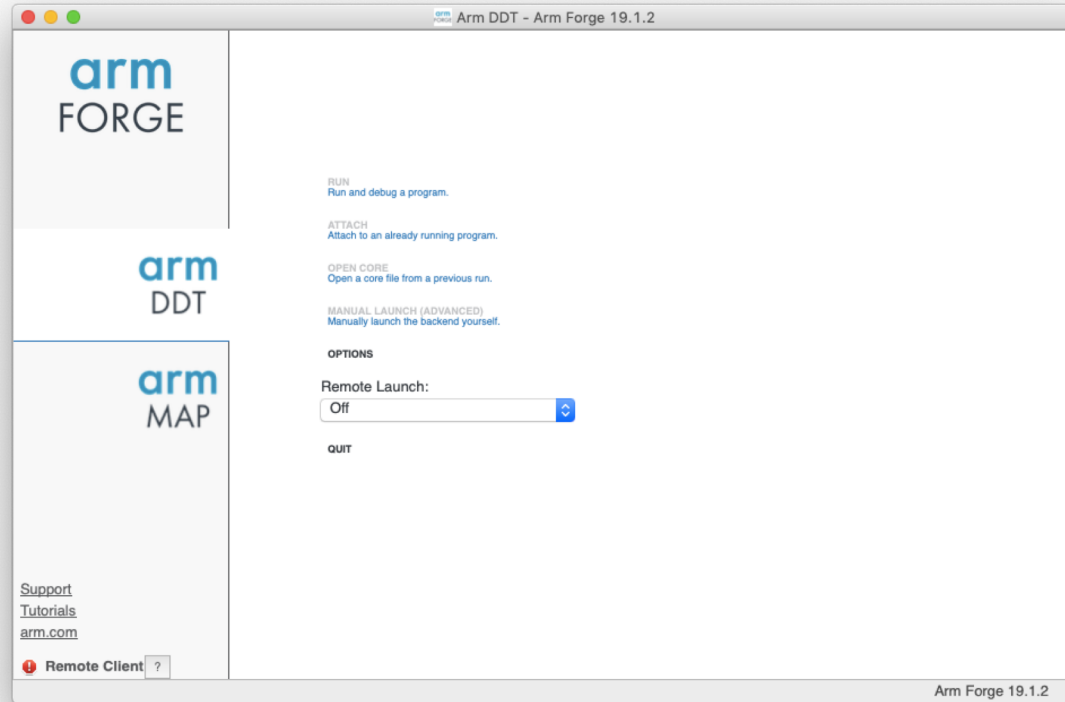
- If you enable ssh ControlMaster, you will only need to authenticate your first connection to a remote host.
 - Additional ssh connections to the same host will piggyback
 - Be mindful of what else on your laptop might invoke ssh
- Supported on Linux, Mac. For Windows only \geq Windows 10 if enabled (?)
- In your laptop `~/.ssh/config` add these lines:
 - `ControlMaster auto`
 - `ControlPath ~/.ssh/master-%r@%h:%p`
- Note: If you need X11 forwarding (i.e. `ssh -Y`) then you must start the first ssh connection using `-Y` and also include `-Y` on any secondary ssh connections you make that need the forwarding.

Allinea DDT startup (likewise for MAP)

- Option 1: Run using remote client (RECOMMENDED)
 - For Mac or Windows, download and install “Remote client”
 - <https://developer.arm.com/tools-and-software/server-and-hpc/downloads/arm-forge>
 - For Linux – download full version of Forge (no license required for use as remote client)
 - Note version must match between remote client and target machine
 - N.B. *Remote Client downloads for older versions of Arm Forge*
- Option 2: Run client on login node (Not recommended)
 - Start X11 server on your laptop and ssh -Y to login node
 - module load forge
 - ddt &

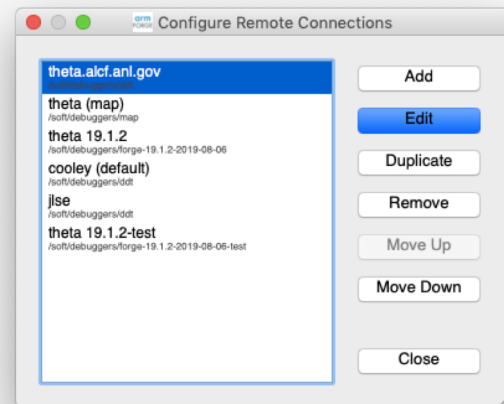
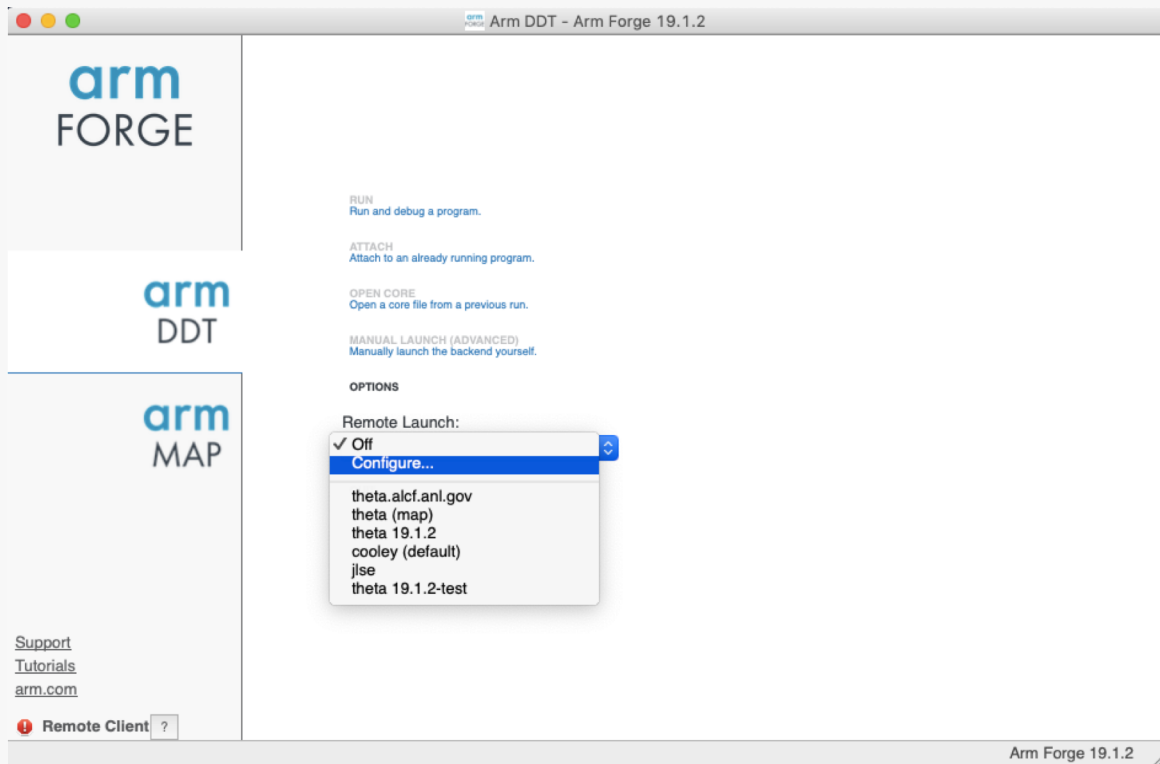
DDT Remote Client (1)

GUI looks just like the X11 Client



DDT Remote Client (2,3)

Select "configure", then in dialog box "Add" to set up a new connection



DDT Remote Client (4)

Note: this remote installation directory is the system default version of DDT, corresponding to *module load forge*. Click "Test Remote Launch" to verify, then OK.

Remote Launch Settings

Connection Name: user_name@theta.alcf.anl.gov

Host Name: user_name@theta.alcf.anl.gov

[How do I connect via a gateway \(multi-hop\)?](#)

Remote Installation Directory: /soft/debuggers/ddt

Remote Script Optional

Always look for source files locally

KeepAlive Packets: Enable

Interval: 30 seconds

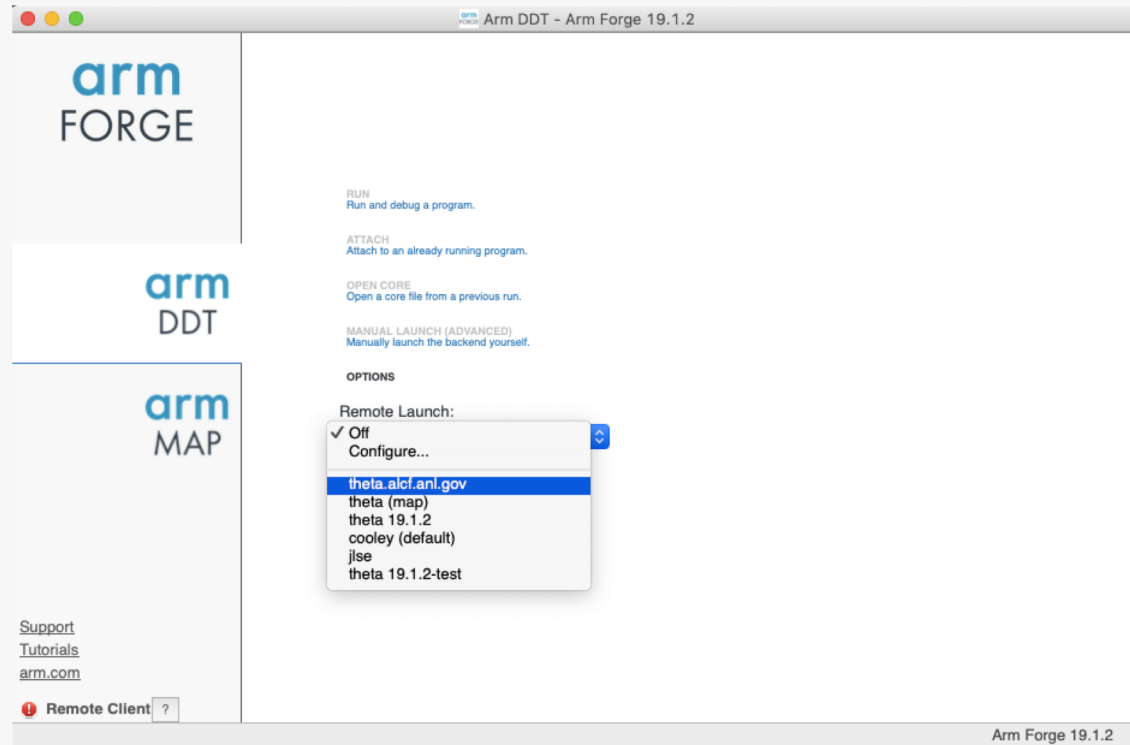
Proxy through login node

Test Remote Launch

Help OK Cancel

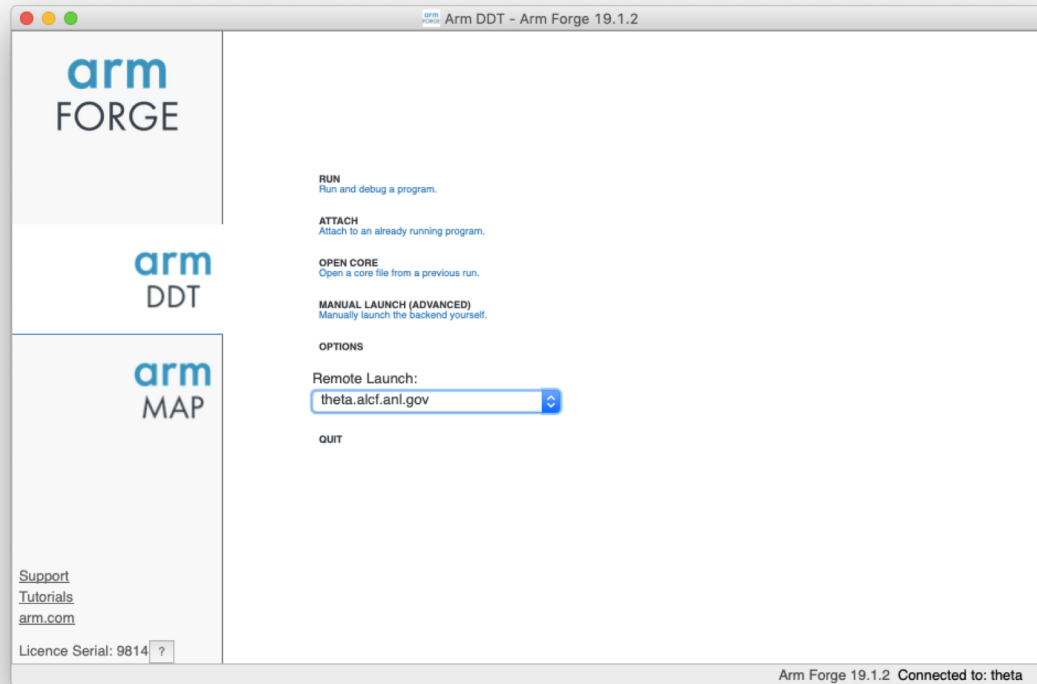
DDT Remote Client (5)

Now that it is defined, select remote machine



DDT Remote Client (6)

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

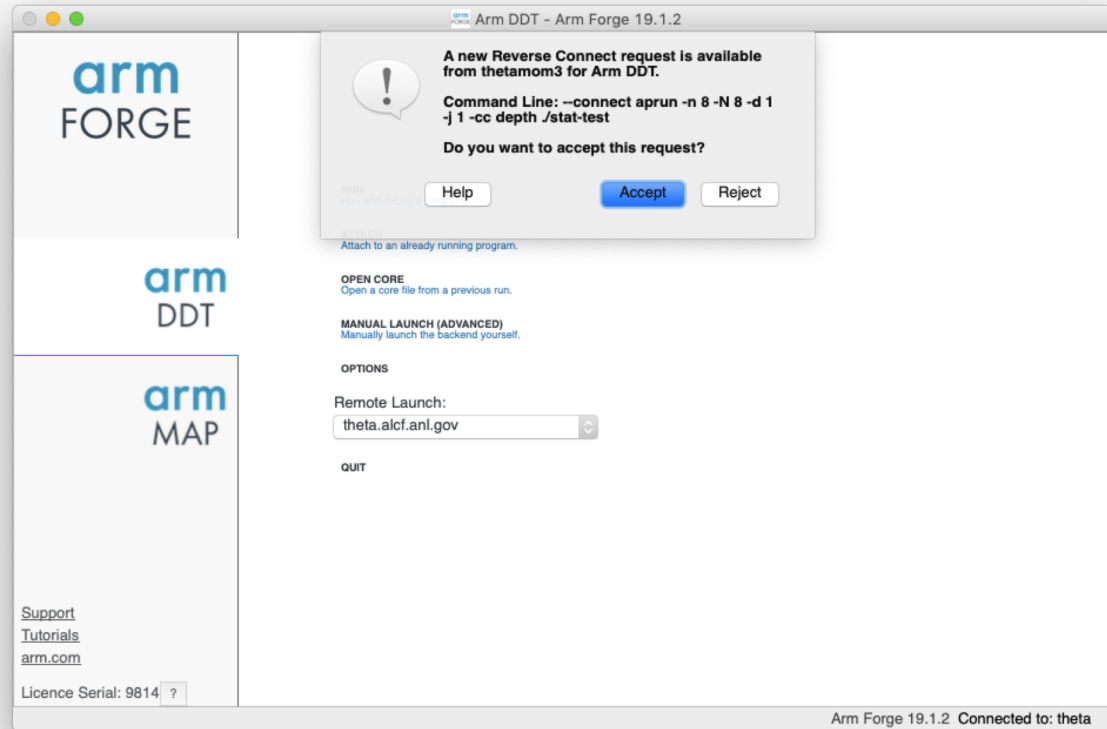


DDT Startup - Reverse Connect

- Start remote client and connect to login node (or start client on login node displaying back via X11). Let this window sit for now.
- In an ssh session to the login node
 - Run an interactive job (qsub -l)
 - Theta: Instead of aprun ... myprog.exe
 - ddt --connect aprun ... myprog.exe
 - Cooley: Instead of mpirun -f \$COBALT_NODE_FILE -n 12 a.out
 - ddt --connect -n 12 -mpiargs "-f \$COBALT_NODEFILE" a.out
- Handy tip: edit your job script then run it from the interactive command line
- Likewise with Alinea MAP
 - Theta: map --connect aprun ... myprog.exe

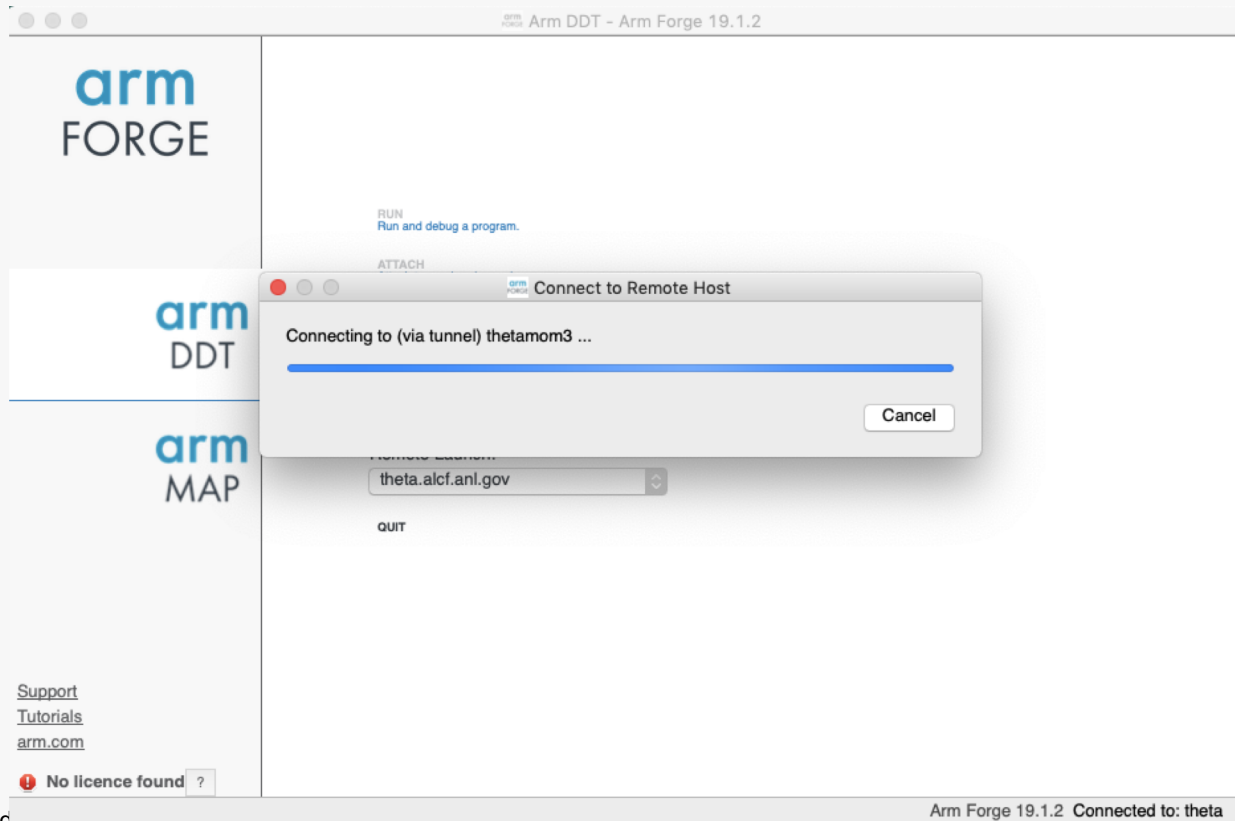
DDT Reverse Connect (1)

When `ddt --connect` starts, a reverse connect request will appear. Click *Accept*.



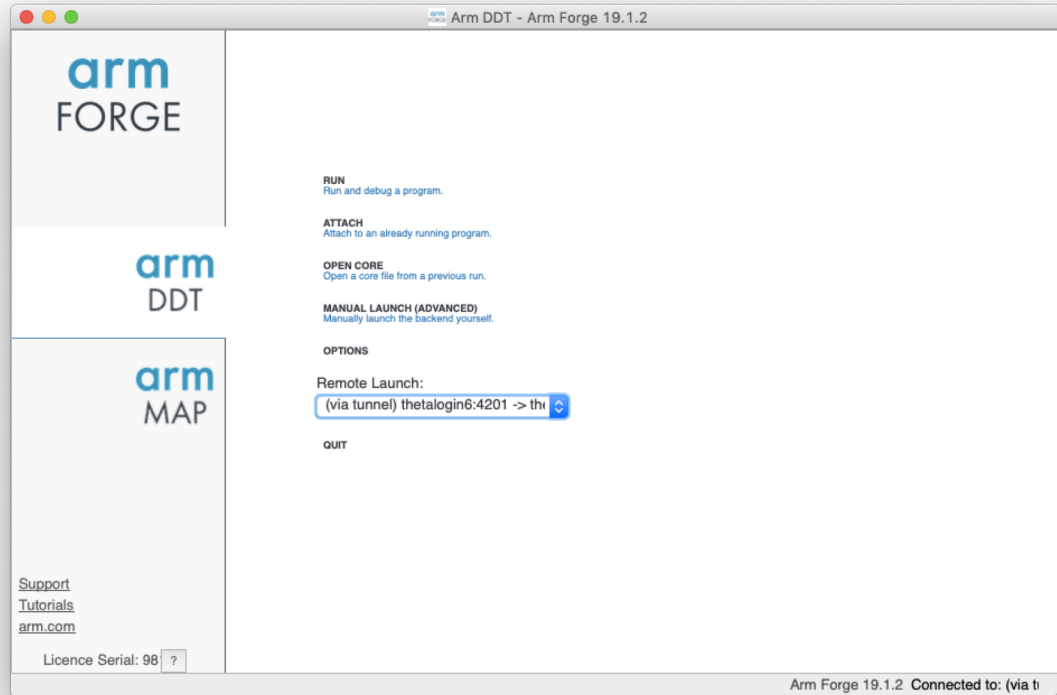
DDT Reverse Connect (2)

The connection will take a few seconds to establish.



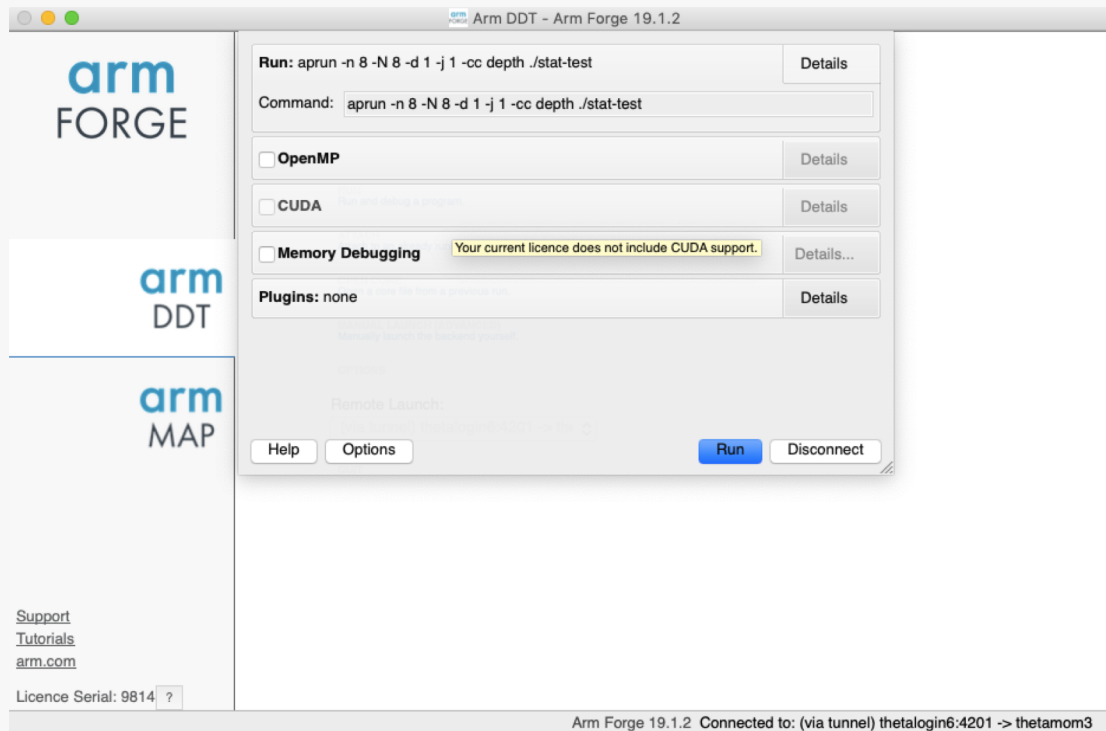
DDT Reverse Connect (3)

Briefly it will display (*via tunnel*)



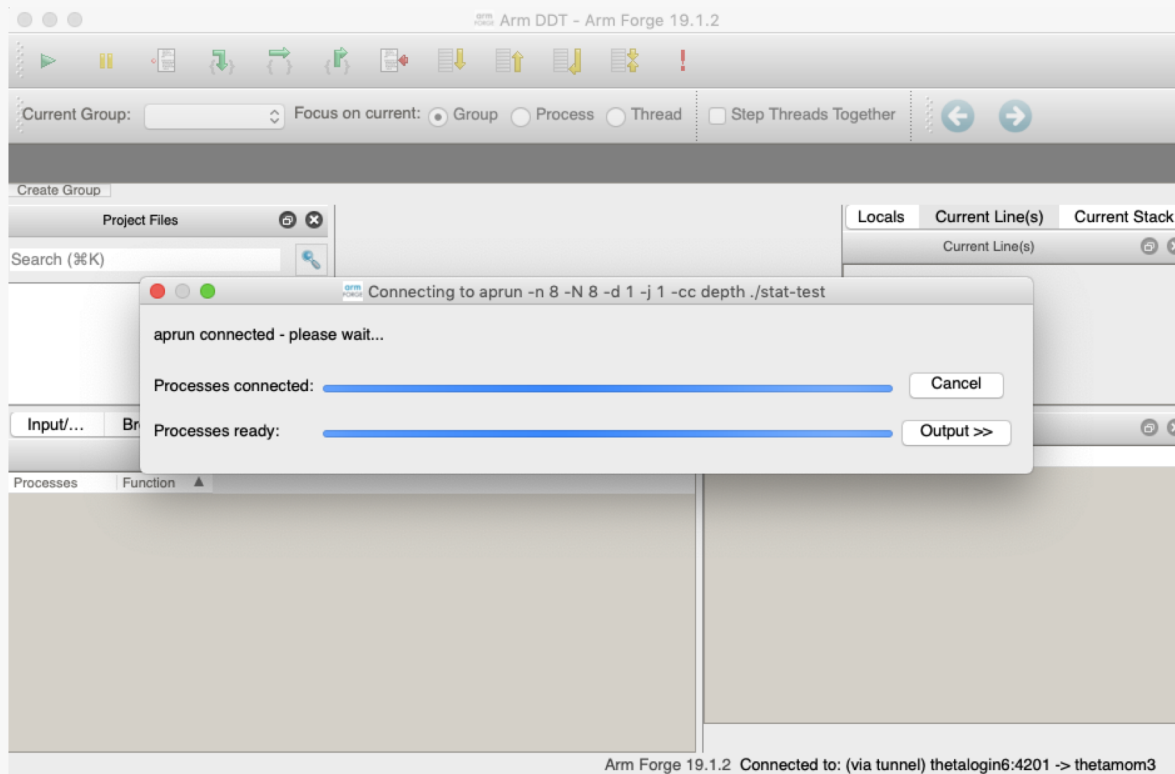
DDT Reverse Connect (4)

The Remote client is now connected to DDT running on the remote host.
Make selections as desired then click *Run*.



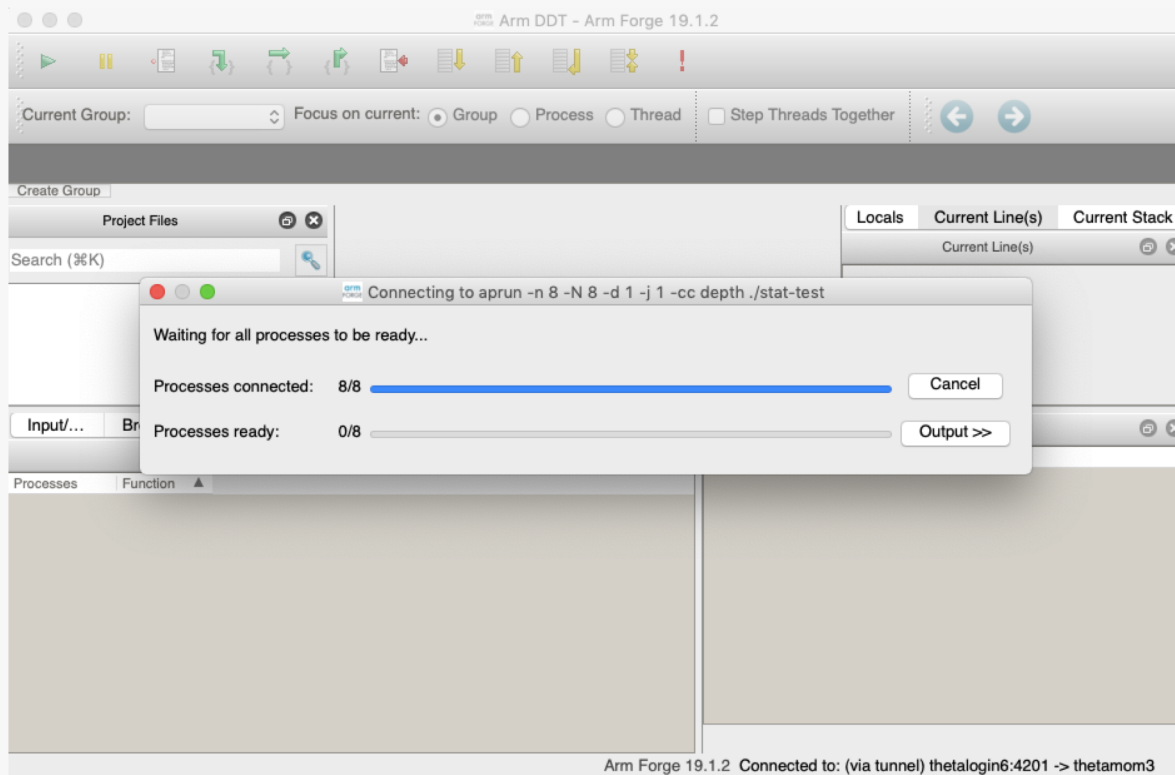
DDT Reverse Connect (5)

DDT on the remote host will now execute the aprun command and attach to it.



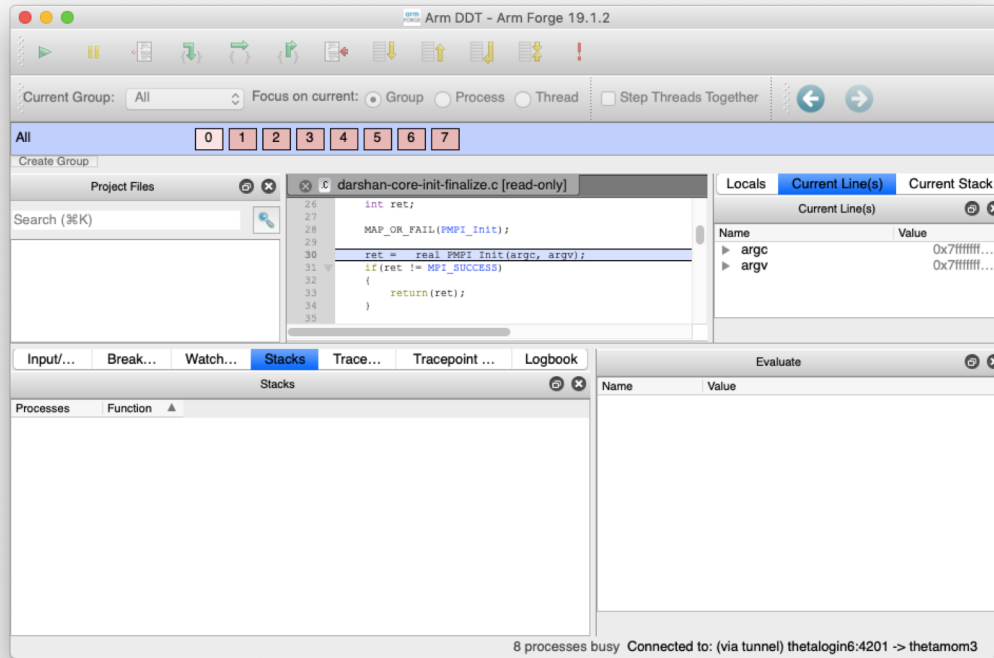
DDT Reverse Connect (6)

Progress of connecting to MPI processes.



DDT Reverse Connect (7)

Ready to debug!



Questions

See also

– <https://www.alcf.anl.gov/support-center>

– Email: support@alcf.anl.gov

