Starting DDT and MAP at NERSC

Woo-Sun Yang
NERSC User Services Group
Allinea DDT and MAP Training
May 14, 2015
Overview of today’s training

• Training on Allinea’s DDT (parallel debugger) and MAP (parallel profiler)
• Lectures and demo sessions
  – Beau Paisley, Allinea Software
• Hands-on session in the afternoon
  – You can use your own code
• Use the ‘training’ module to get training materials that Beau prepared:

```
$ module load training
$ echo $EXAMPLES  # example code directory name
/project/projectdirs/training/2015/allinea_ddt_map

$ cp -R $EXAMPLES .  # copy the entire directory
$ cd allinea_ddt_map
$ ls
```

• We will primarily use Edison for today’s demos
Allinea DDT and MAP

• **DDT (Distributed Debugging Tool)**
  – Graphical parallel debugger
  – Can be used for debugging serial, MPI, OpenMP, CAF, UPC, CUDA codes
  – Intuitive and simple to use
  – 8192 license tokens (≈MPI tasks)

• **MAP**
  – Source-level parallel code performance profiler
  – Measures floating point operations, memory usage, floating point vector instructions, CPU time, I/O, MPI communication, load balancing etc.
  – OpenMP threading support added lately
  – 512 license tokens (≈MPI tasks)

• **DDT and MAP on**
  – Edison
  – Hopper
  – sl6carver: Carver with Scientific Linux 6
  – sl5carver: Carver with Scientific Linux 5
  – Babbage (Xeon Phi)
Using DDT or MAP

• Mostly run in an interactive batch session

  $ qsub -I -v DISPLAY -lmppwidth=24,walltime=30:00 -q debug
  ...
  $ cd $PBS_O_WORKDIR
  $ module load allineatools
  $ ddt ./a.out
  $ map ./a.out

• Can run in non-interactive mode, too ("off-line" mode with DDT and command line mode with MAP)

• See

  https://www.nersc.gov/users/software/debugging-and-profiling/ddt/#toc-anchor-3
One “environment” called Allinea Forge, starting 5.0

Running DDT

Running MAP
If you are far away from NERSC

- Remote X window application (GUI) over network: slow response

- Two solutions
  - Use NX to improve the speed
    - Works with any X window applications

  - Use Allinea Forge remote client
    - Runs on your desktop/laptop
    - Connects to NERSC machine and submits a batch job
    - Displays results in real time
    - No license file required on your local desktop/laptop
Using NX
First, set up the remote client for login and using Allinea tools

(1) Select ‘Configure’ to create a configuration for a NERSC machine

(2) Create a configuration
Then, connect to a machine using remote client

(1) Select a machine

RUN
Run and debug a program.

ATTACH
Attach to an already running program.

OPEN CORE
Open a core file from a previous run.

MANUAL LAUNCH (ADVANCED)
Manually launch the backend yourself.

OPTIONS

Remote Launch:

- Off
- Configure...
- edison
- hopper
- carver

(2) Enter the passphrase for your ssh key

Enter your password for the SSH key "id_rsa".

Password:

- Remember password in my keychain

Cancel OK

or enter the NIM password

Connecting to wyang@edison.nersc.gov ...

Password:
Set up for batch job submission (for Edison, Hopper)

After clicking the ‘OPTIONS’ button...

- MPI/UPC Implementation: Cray X-Series (MPI/shmem/CAF)
- Override default aprun path: e.g. /usr/local/bin/aprun
- Debugger: Automatic (recommended)
- Create Root and Workers groups automatically
- Use shared symbol cache (not supported with this debugger/platform)
- Heterogeneous system support
- Enable CUDA software pre-emption (CUDA 5.5+)

Job Submission Settings:
- Submission template file: /usr/common/usg/allineatools/5.0.1/templates/pbs-xt4.qif
- Submit command: qsub
- Regexp for job id: \(d+\)
- Cancel command: qdel JOB_ID_TAG
- Display command: qstat
- Edit Queue Parameters...
- Quick Restart
Then, submit a job from the remote client

Working Directory needs to be set correctly

‘Submit to Queue’ must be selected; queue parameters can be adjusted
• If things go wrong (e.g., DDT using `mpirun` instead of `aprun` on Edison, Hopper), do `rm -rf ~/.allinea` and start from beginning

• See