Parallel debuggers

High Performance Scientific Computing 2018
Rio Yokota

GDB

How to compile:

```
> cd hpc_lecture/debugging
> g++ -O0 –g 00_matmul.cpp
```

How to run:

```
> gdb ./a.out
(gdb) run 512
```

How to use gdb:

`list` (Look at source code)
`break` (Set a breakpoint)
`run` (Execute the code till the breakpoint)
`next` (Execute the next line)
`step` (Execute the next line and step into fuctions)
`continue` (Continue to the next breakpoint)
`print` (Print values)
`backtrace` (Show the stack backtrace)
`quit` (Quit GDB)

https://wiki.scinet.utoronto.ca/wiki/images/c/ce/Best-practice-debug.pdf

GDB + MPI

How to use qrsh with X forwarding:

```
> qrsh -g tga-hpc-lecture -l q_node=1 -l
h_rt=0:10:00 -pty yes -display $DISPLAY -v
TERM /bin/bash
```

How to use module inside a compute node:

```
> . /etc/profile.d/modules.sh
> module load intel intel-mpi
```

How to compile:

```
> mpicxx -O0 -g 01_mpi.cpp
```

How to run:

```
> mpirun -np 2 xterm -e gdb ./a.out
```

Allinea DDT

Setup:

```
> qrsh -g tga-hpc-lecture -l q_node=1 -l
h_rt=0:10:00 -pty yes -display $DISPLAY -v
TERM /bin/bash
> . /etc/profile.d/modules.sh
> module load intel intel-mpi allinea
```

How to compile:

```
> mpicxx -O0 -g 01_mpi.cpp
```

How to run:

```
> ddt a.out
```

Reference:

http://content.allinea.com/downloads/userguide-forge.pdf

CUDA-GDB

How to compile:

```
> nvcc -Xcompiler "-O3 -fopenmp -g -O0" -g -O0 02_gpu.cu
```

How to run:

```
> cuda-gdb ./a.out
> cuda-memcheck ./a.out
> nsight
```

http://on-demand.gputechconf.com/gtc/2013/presentations/S3037-S3038-Debugging-CUDA-Apps-Linux-Mac.pdf