Debuggers

• **Fix coding errors for**
  – Wrong results
  – Program crash
  – Program hang

• **How to find them?**
  – Place print statements in what you think strategic locations
    • Difficult to know where the code fails and whether variables have incorrect values
    • Recompile whenever you make a change
    • Tedious and exhausting
  – Using debuggers for your detective work
    • Compile only once (generally)
    • Control execution pace of your program
    • Examine values using debugger’s tools
      – Visualization and statistics
    • Can identify where the code fails or hangs
Parallel debuggers on Cori and Edison

- **Parallel debuggers with a graphical user interface**
  - DDT (Distributed Debugging Tool)
  - TotalView

- **Specialized debuggers on Cori and Edison**
  - STAT (Stack Trace Analysis Tool)
    - Collect stack backtraces from all (MPI) tasks
  - ATP (Abnormal Termination Processing)
    - Collect stack backtraces from all (MPI) tasks when an application fails

- **Valgrind**
  - Suite of debugging and profiling tools
  - Best known for its detailed memory debugging (memcheck)

- **Intel Inspector**
  - Thread and memory debugging

- **Cray debuggers for comparative debugging**
  - CCDB
  - lgdb
Parallel debuggers on Cori and Edison

- **Parallel debuggers with a graphical user interface**
  - DDT (Distributed Debugging Tool)
  - TotalView

- **Specialized debuggers on Cori and Edison**
  - STAT (Stack Trace Analysis Tool)
    - Collect stack backtraces from all (MPI) tasks
  - ATP (Abnormal Termination Processing)
    - Collect stack backtraces from all (MPI) tasks when an application fails

- **Valgrind**
  - Suite of debugging and profiling tools
  - Best known for its detailed memory debugging (memcheck)

- **Intel Inspector**
  - Thread and memory debugging

- **Cray debuggers for comparative debugging**
  - CCDB
  - lgdb

Today

Some Other Tools: Training on April 24
DDT and TotalView

- GUI-based traditional parallel debuggers
- Works for C, C++, Fortran programs with MPI, OpenMP, pthreads
  - DDT supports CAF (Coarray Fortran) and UPC (Unified Parallel C), too
- Licenses
  - DDT: up to 4096 MPI tasks on Cori (Haswell and KNL) and Edison
  - TotalView: up to 512 MPI tasks on Cori (Haswell) and Edison
  - Licenses shared among users and machines
- For info
  - https://www.roguewave.com/products-services/totalview
How to build and run with DDT

$ ftn -g -O0 -o jacobi_mpi jacobi_mpi.f90  
-g for debugging symbols;  
-O0 for the Intel compiler

$ salloc -N 1 -t 30:00 -q debug -C knl  
Start an interactive batch session

$ module load allinea-forge  
Load the allinea-forge module to use DDT

$ ddt ./jacobi_mpi  
Start DDT
If you work far away from NERSC

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

• Two solutions
  – Use NX to improve the speed
    • Works for X window applications
    • https://www.nersc.gov/users/network-connections/using-nx/ (general)
    • http://portal.nersc.gov/project/mpccc/nx/NX_Tutorial/Start_Over.html (installation and quick user guide)
  – Use Arm Forge remote client
    • Runs on your desktop/laptop
    • Submit a debugging batch job from a NERSC machine and make the client reverse connect to the job
    • Displays results in real time
    • No license file required on your local desktop/laptop
    • https://www.nersc.gov/users/software/performance-and-debugging-tools/ddt#toc-anchor-5 (setup)
DDT window

For navigation, parallel stack frame view is helpful in quickly finding out where each process is executing.

To check the value of a variable, right-click on a variable or check the pane on the right.

Sparklines to quickly show variation over MPI tasks.

To evaluate expressions.
Navigating in your program

- Play/Continue
- Pause
- Add Breakpoint
- Step Into
  - To next line; if it’s a function call, enter the function
- Step Over
  - To next line in the current stack frame even if it’s a function call
- Step Out
  - Return to the caller function
- Run To Line
Breakpoints, watchpoints and tracepoints

• **Breakpoint**
  – Stops execution when a selected line (breakpoint) is reached
  – Double click on a line to create one; there are other ways, too

• **Watchpoints for variables or expressions**
  – Stops when a variable or an expression changes its value

• **Tracepoints**
  – When reached, prints what lines of codes is being executed and the listed variables

• **Can add a condition for an action point**
  – Useful inside a loop

• **Can be made active or inactive**
Many ways to check variables

- Right click on a variable for a quick summary
- Variable pane
- Evaluate pane
- Display variable values over processes (Compare across processes) or threads (Compare across threads)
- MDA (Multi-dimensional Array) Viewer
  - Visualization
  - Statistics
National Energy Research Scientific Computing Center