Debugging Tools New User Training





Woo-Sun Yang User Engagement Group, NERSC

June 21, 2019





Debuggers



• Program errors

- Program crashes
- Program hangs
- Wrong results

How to find and fix them?

- Print statements
 - Difficult to guess where and what to print
 - Recompile and submit jobs whenever you change them
 - Tedious, exhausting and time-consuming
 - Hard to extract info from output about the error, especially for parallel codes
- Debuggers
 - Compile only once (generally)
 - Control execution of your program
 - Check variables; visualize and get stats





Parallel debuggers on Cori



- Parallel debuggers with a graphical user interface
 - DDT (Distributed Debugging Tool)
 - TotalView
- Specialized debuggers
 - 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 tool, 'memcheck'
 - <u>https://docs.nersc.gov/development/performance-debugging-tools/valgrind/</u>
 - Intel Inspector

Office of

Science

- Threading and memory debugging
- <u>https://docs.nersc.gov/programming/performance-debugging-tools/inspector/</u>



- 3 -

DDT and TotalView



- GUI-based traditional parallel debuggers
- C, C++, Fortran codes with MPI, OpenMP, pthreads
- Licenses
 - DDT: up to 8192 MPI tasks on Cori
 - TotalView: up to 512 MPI tasks on Cori
 - Shared among users and machines
- For info
 - <u>https://developer.arm.com/tools-and-software/server-and-h</u> <u>pc/arm-architecture-tools/arm-forge</u>
 - <u>https://docs.nersc.gov/development/performance-debugging-t</u> <u>ools/ddt/</u>
 - <u>https://www.roguewave.com/products-services/totalview</u>
 - <u>https://docs.nersc.gov/development/performance-debugging-t</u> <u>ools/totalview/</u>





How to build and run with DDT



\$ ftn -g -00 -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
\$ module load allinea-forge
\$ ddt ./jacobi mpi

Start an interactive batch session Load the allinea-forge module to use DDT Start DDT

🖉 🔍 📉 Run		
Application: /global/cscratch1/sd/wyang/debugging/jacobi_mpi		Details
Application: //global/cscratch1/sd/wyang/debugging/jacobi_mpi		
Arguments:		•
🗖 stdin file: 📔		- 6
Working Directory:		
🔽 MPI: 16 processes, SLURM (MPMD)		Details
Number of Processes: 16		
🏳 Processes per Node 📔 👻		
Implementation: SLURM (MPMD) Change		
srun arguments		•
Г OpenMP		Details
☐ CUDA		Details
Memory Debugging		Details
☐ Submit to Queue	Configure	Parameters
Environment Variables: none		Details
Plugins: none		Details





If you work far away from NERSC



- Running X11 GUIs over network: reponses painfully slow due to intrinsically high latency and inefficient bandwidth between X11 client and server
- Two solutions
 - Use NX (NoMachine) to improve the speed
 - Works for X11 window applications
 - <u>https://docs.nersc.gov/connect/nx/</u>
 - Use Arm Forge remote client
 - Run on your desktop/laptop
 - Submit a debugging batch job on a NERSC machine and make the job connect to the client ("reverse connect")
 - Displays results in real time
 - <u>https://developer.arm.com/tools-and-software/server-and-hpc/arm-architect</u> <u>ure-tools/downloads/download-arm-forge</u> (for downloading remote clients)
 - <u>https://docs.nersc.gov/programming/performance-debugging-tools/ddt/#</u> <u>reverse-connect-using-remote-client</u> (for setup)





Arm Forge remote client settings



• • •	Remote Launch Settings	
Connection Name:	cori	
Host Name:	wyang@cori.nersc.gov wyang@cmom02.nersc.gov	
	How do I connect via a gateway (multi-hop)?	
Remote Installation Directory:	/global/common/sw/cray/cnl6/haswell/allinea-forge/default	
Remote Script	/global/common/sw/cray/cnl6/haswell/allinea-forge/remote-init	
	Always look for source files locally	
KeepAlive Packets:	Enable	
Interval:	30 seconds	0
Uncheck	Proxy through login node	
		Test Remote Launch
Help		OK Cancel

• See

https://docs.nersc.gov/development/performance-debugging-tool





DDT window

Science









- 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





Check variables



Evaluate

Cancel

Distributed Array Dimensions: None 2 How do I view distributed arrays?

- Right click on a variable for a quick summary
- Variable pane
- Evaluate pane

U.S. DEPARTMENT OF

Office of

Science

- Display variable values over processes (Compare across processes) or threads (Compare across threads)
- MDA (Multi-dimensional Array) Viewer
- Align Stack Frames Auto-update Staggered Array What does this do? Visualization Range of \$i Statistics From: 0 To: 49 🗲 Multi-Dimensional Array Viewe Display: Rows Array Expression: uNorth[\$i] Evaluate Cancel Distributed Array Dimensions: None C How do I view distributed arrays? Only show if: See Examples Visualization Staggered Array What does this do? Data Table Statistics Range of \$i Count: 50 From: 0 Not shown: 0 Errors: 0 To: 49 Aggregate: 0 Display: Rows Numerical: 50 Sum: 15.2618 Only show if: See Examples Minimum: 0 Data Table Statistics Maximum: 0.580318 Range: 0.580318 Visualize Export Eull Window Goto Mean: 0.305235 i 0 Variance: 0.030429 0.0131118195 nan: 0 2 0.0262203719 -nan: 0 3 0.0393223912 4 0.0524146073 inf 0 0.0654937625 -inf: 0 0.0785565972 <0: 0 0.0915998444 =0: 1 0 10462027 >0: 49 Help Help Close Help



TotalView





• Click 'Yes' to the question 'Process srun is a parallel job. Do you want to stop the job now?'







TotalView (cont'd)



Root window	Process window
	For navigation X srun <jacobi_mpiomp>.0</jacobi_mpiomp>
TotalView for HPC 2018.3.8	Eile Edit View Group Process Ibread Action Point Debug Tools Window Help
File Edit View Tools Help	Group (Control)
<pre>Provide address ></pre>	Kark 0: stork.jacobi_mpiomp>,0 (At Breakpoint 1) Thread 1 (12963008): jacobi_mpiomp (At Breakpoint 1) Stack Trace Stack Trace Stack Trace (F90) jacobi_mpiomp, PP=7fffffff5af0 main, PP=7ffffffff5e90
-9.1	70 call set_bc (unew, n, js, je) I o see the value of a 71 71
Configure ≤<	73 Variable, right-click on a
State of MPI tasks	75 76 16 (nyid == 0) print Add to Expression List variable to "dive" on it or
and threads;	78 ! Make the new value the Across Processes 79 80 I Source parallel do Across Threads
members denoted	81 do j=js-l.je+1 82 u(:,j) = unew(:,j) Set Breakpoint
roughly as	84 [Somp end parallel do Create Watchpoint
'rank.thread'	Action Points Threads Enable P-P+Px I-I+
U.S. DEPARTMENT OF ENERGY Office of Science	1 [/jacobi_mpiomp.f90 Disable jacobi_mpiomp+0xeb3 Breakpoints, etc. Properties For selecting MPI task and thread

STAT (Stack Trace Analysis Tool)



- Gathers stack backtraces (sequence of function calls leading up to the current function) from all (MPI) processes
 - Merge them into a single file (*.dot)
 - Results displayed as a single call tree for all processes
 - Can be useful for debugging a hanging application
 - With the info learned from STAT, can investigate further with DDT or TotalView
- Works for MPI, CAF and UPC, OpenMP





STAT (Cont'd)



- STAT commands (after loading the 'stat' module)
 - stat-cl: invokes STAT to gather stack backtraces
 - STATview: a GUI to view the results
 - STATGUI: a GUI to run STAT or view results
- For more info:
 - 'intro_stat', 'STAT', 'STATview' and 'STATGUI' man pages
 - /opt/cray/pe/stat/default/doc/stat_userguide.pdf
 - <u>https://docs.nersc.gov/development/performance-debugg</u> <u>ing-tools/stat_atp/</u>





Debug a hanging application with STAT



 If your code hangs in a consistent manner, you can use STAT to see whether some MPI ranks got stuck.

```
$ ftn -g -o jacobi mpi jacobi mpi.f90 with usual optimization flags, if any
  salloc -N 1 -t 30:00 -g debug -C knl
  srun -n 4 ... ./jacobi mpi &
[1] 53634
$ module load stat
  stat-cl -i 53634
                                 -i to get source line numbers
                                 STAT samples stack backtraces a few times
Attaching to application...
Attached!
Application already paused... ignoring request to pause
Sampling traces...
Traces sampled!
Resuming the application ...
Resumed!
Merging traces...
Traces merged!
Detaching from application...
Detached!
```

- Results written to /global/cscratch1/sd/wyang/debugging/stat/stat_results/jacobi_mpi.0000 \$ 1s -1 stat results/jacobi mpi.0000/*.dot
- -rw-r--r-- 1 wyang wyang 9028 Jun 20 10:42 stat_results/jacobi_mpi.0000/00_jacobi_mpi.0000.3D.dot

\$ STATview stat_results/jacobi_mpi.0000/00_jacobi_mpi.0000.3D.dot





Debug a hanging application with STAT (Cont'd)

U.S. DEPARTMENT OF

ERG

Office of

Science







Cray ATP (Abnormal Termination Processing)



- ATP gathers stack backtraces from all processes when the application fails
 - Invokes STAT underneath
 - Output in atpMergedBT.dot and atpMergedBT_line.dot (which shows source code line numbers), which are to be viewed with STATview
- The atp module is loaded on Cori by default, but ATP is <u>not</u> enabled; to enable:

export ATP_ENABLED=1 # sh/bash/ksh

• For more info

– 'intro_atp' man page

<u>__ing-tools/stat_atp/</u>

<u>https://docs.nersc.gov/development/performance-debugg</u>





National Energy Research Scientific Computing Center



