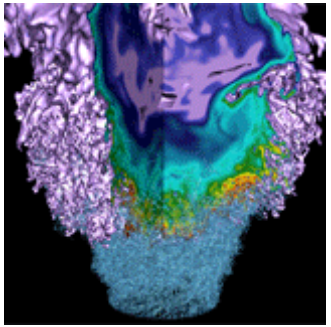
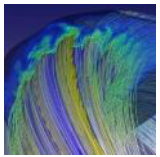
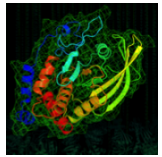
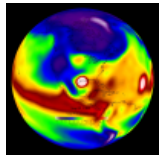
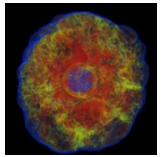
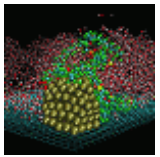
Using Intel Tools at NERSC



Charlene Yang
Application Performance Specialist

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Intel Performance Tools

- **Intel VTune Amplifier**
  - a general view of the code
  - hotspots, threading, concurrency, locksandwaits, memory-access, io
- **Intel Advisor**
  - dig deeper, focus on vectorization and threading
  - memory access pattern, data dependency, tripcounts, Roofline, prototype threading
- **Intel Trace Analyzer and Collector (ITAC)**
  - focus on MPI communication
  - load imbalance, deadlocks, idealization, …
- **Intel Inspector**
  - focus on memory and threading
  - memory errors, nondeterministic threading errors

Intel VTune Amplifier

- Use SCRATCH rather than PROJECT or HOME because they do not support mmap functionality
  - `cd $SCRATCH`

- Compile with the following settings:
  - `module swap craype-haswell craype-mic-knl`
  - `export CRAYPE_LINK_TYPE=dynamic`
  - `cc -g -O3 -debug inline-debug-info -qopt-report=3 -qopt-report-phase=vec -qopenmp`

  - Turns on -xMIC-AVX512, debugging, optimization report, OpenMP, dynamic linking

Intel VTune Amplifier

- Run the job with VTune
  - `module load vtune`
  - `salloc -q interactive -C knl -N2 -t 00:30:00 --perf=vtune`
    (use --reservation if possible)

  - Loads runtime drivers for hardware event based analyses, such as "memory-access," "advanced-hotspots," and "general-exploration"

  - `module unload darshan` (Might interfere with VTune data collection)
  - `export OMP_NUM_THREADS=4`
  - `export OMP_PROC_BIND=spread`
  - `export OMP_PLACES=threads`

Intel VTune Amplifier

- Run the job with VTune
  - `srun -n 8 -c 16 --cpu_bind=cores amplxe-cl -collect hotspots -r results/ -data-limit=0 -trace-mpi -finalization-mode=deferred -- ./a.out`
  - Two nodes, 4 MPI tasks each, 4 OpenMP threads per task, bound to core
  - Set data collection limit to xxMB, or 0 (no upper limit) -- default is 500MB
  - `trace-mpi` will produce two results folders:
    - `results.nid02506 results.nid02507 (: data.0 data.1 data.2 data.3)`
  - `finalization-mode`
    - `deferred:` only calculate binary checksum
    - `none:` skip

Intel VTune Amplifier

```
amplxe-cl -help <action>
amplxe-cl -help collect <analysis type>
```

Available Actions:

- collect
- collect-with
- command
- finalize
- help
- import
- report
- version

Available Analysis Types:

- advanced-hotspots Advanced Hotspots
- concurrency Concurrency
- disk-io Disk Input and Output
- general-exploration General Exploration
- hotspots Basic Hotspots
- hpc-performance HPC Performance Characterization
- locksandwaits Locks and Waits
- memory-access Memory Access
- memory-consumption Memory Consumption
- system-overview System Overview
- ……

Intel VTune Amplifier

- Modifiers for **`collect`** action:
  - **`[no]-allow-multiple-runs, [no]-analyze-system, data-limit, discard-raw-data,duration, finalization-mode,[no]-follow-child, knob , mrte-mode, quiet, resume-after, return-app-exitcode, ring-buffer, search-dir, start-paused, strategy, [no-]summary, target-duration-type ,target-pid, target-process, target-system, trace-mpi, no-unplugged-mode, user-data-dir, verbose`**
  - https://software.intel.com/en-us/vtune-amplifier-help-collect
- Finalize results
  - Open the GUI on a login node, **`amplxe-gui`**, “Open Results” (not Open Projects)
  - or, **`amplxe-cl -finalize -r results-nofinal.nid02507/`**

Intel VTune Amplifier

- GUI
  - login node with X forwarding
    - **`module load vtune`**
    - launch **`amplxe-gui`**
  - NoMachine
    - Web brower: https://nxcloud01.nersc.gov/nxwebplayer
    - Set up SSH: https://docs.nersc.gov/connect/nx/

- NERSC documentation
  - https://www.nersc.gov/users/software/performance-and-debugging-tools/vtune/

Intel Advisor

- Use SCRATCH rather than PROJECT or HOME because they do not support mmap functionality
  - `cd $SCRATCH`

- Compile with the following settings:
  - `module swap craype-haswell craype-mic-knl`
  - `export CRAYPE_LINK_TYPE=dynamic`
  - `cc -g -O3 -debug inline-debug-info -qopt-report=3 -qopt-report-phase=vec -qopenmp`

  - Turns on -xMIC-AVX512, debugging, optimization report, OpenMP, dynamic linking

Intel Advisor

- Run the job with Advisor
  - `salloc -q interactive -C knl -N2 -t 00:30:00 --perf=vtune`

    (use --reservation if possible; no need to load certain drivers like VTune does)

  - `module load advisor`
  - `export OMP_NUM_THREADS=4`
  - `export OMP_PROC_BIND=spread`
  - `export OMP_PLACES=threads`
  - `srun -n 8 -c 16 --cpu_bind=cores advixe-cl -collect survey -r results/ -data-limit=0 -trace-mpi -no-auto-finalize -- ./a.out`

  - `no-auto-finalize` to suppress finalization on compute nodes

Intel Advisor

- Run the job with VTune
  - **`advixe-cl -help <action>`**
  - **`advixe-cl -help collect <analysis type>`**
    - survey
    - dependencies
    - map
    - suitability
    - tripcounts
    - roofline
      - one pass of survey, and another pass of tripcounts
      - doesn't support **`-trace-mpi`**, need to use MPMD mode
- Finalize results
  - Open result with GUI on a login node, **`advixe-gui`**
  - **`advixe-cl -report <analysis type> -r results-nofinal.nid02507/`**

Intel Advisor

- Roofline Analysis
  - `srun -n1 -c272 --cpu_bind=cores advixe-cl -collect survey --project-dir results/ -data-limit=0 -no-auto-finalize -- ./stencil-knl`
  - `srun -n1 -c272 --cpu_bind=cores advixe-cl -collect tripcounts --flop --no-trip-counts --project-dir results/ -data-limit=0 -no-auto-finalize -- ./stencil-knl`
  - (On a login node) `advixe-cl -report roofline --project-dir results/`
  - MPMD mode:
    - `srun -n16 -multi-prog survey.conf`
      - `0 advixe-cl -collect survey --project-dir results/ -data-limit=0 -no-auto-finalize -- ./stencil-knl`
      - `1-15 ./stencil-knl`
    - Same for `tripcounts`

Intel Advisor

- Packing Advisor projects
  - `advixe-cl --snapshot --project-dir ./results --pack --cache-sources --cache-binaries -- profile-knl-19493575`

- NRESC documentation:
- https://www.nersc.gov/users/software/performance-and-debugging-tools/advisor/

Intel Trace Analyzer and Collector (ITAC)

- Compile with **Intel MPI and Intel compiler**:
  - `module swap craype-haswell craype-mic-knl`
  - `module load impi`
  - dynamic linking:
    - `export CRAYPE_LINK_TYPE=dynamic`
    - `mpiicc -cc=icc -qopenmp xthi.c -o xthi.x`
  - static linking:
    - `module load itac`
    - `mpiicc -cc=icc -qopenmp xthi.c -o xthi.x -L$VT_LIB_DIR -lVT $VT_ADD_LIBS`

Intel Trace Analyzer and Collector (ITAC)

- Run executables compiled with Intel MPI and Intel compiler:
  - `module load impi itac`
  - dynamic linking:
    - `export LD_PRELOAD=$VT_ROOT_ARCH/slib/libVT.so`
  - static linking: nothing
  - `export VT_LOGFILE_FORMAT=STFSINGLE # produce one single file`
  - `export VT_PCTRACE=5 # call stack`
  - `export I_MPI_FABRICS=shm:tcp # set fabric`
  - if there is OpenMP
    - `export INTEL_LIBITTNOTIFY64=$VT_ROOT_ARCH/slib/libVT.so`
    - `export KMP_FORKJOIN_FRAMES_MODE=0`
    - `export OMP_NUM_THREADS=8; export OMP_PROC_BIND=spread`
    - `export OMP_PLACES=threads`
  - `srun -n 4 ./xthi.xx`

Intel Trace Analyzer and Collector (ITAC)

- Compile with **Cray MPI and Intel compiler (i.e. PrgEnv-intel)**
  - `module swap craype-haswell craype-mic-knl`
  - dynamic linking:
    - `export CRAYPE_LINK_TYPE=dynamic`
    - `cc -qopenmp xthi.c -o xthi.x`
  - static linking:
    - `module load itac`
    - `cc -qopenmp xthi.c -o xthi.x -L$VT_LIB_DIR -lVT $VT_ADD_LIBS`

Intel Trace Analyzer and Collector (ITAC)

- Run executables compiled with Cray MPI and Intel compiler (i.e. PrgEnv-intel)
  - `module load itac`
  - dynamic linking:
    - `export LD_PRELOAD=$VT_ROOT_ARCH/slib/libVT.so`
  - static linking: nothing
  - `export VT_LOGFILE_FORMAT=STFSINGLE # produce one single file`
  - `export VT_PCTRACE=5 # call stack`
  - if there is OpenMP
    - `export INTEL_LIBITTNOTIFY64=$VT_ROOT_ARCH/slib/libVT.so`
    - `export KMP_FORKJOIN_FRAMES_MODE=0`
    - `export OMP_NUM_THREADS=8`
    - `export OMP_PROC_BIND=spread`
    - `export OMP_PLACES=threads`
  - `srun -n 4 ./xthi.xx`

Intel Trace Analyzer and Collector (ITAC)

- View results:
  - NoMachine, **`module load itac; traceanalyzer`**
- Documentation:
  - https://www.nersc.gov/users/software/performance-and-debugging-tools/intel-trace-analyzer-and-collector/
- ITAC collector libraries
  - libVT, libVTim, libVTfs, ….
  - https://software.intel.com/en-us/node/561272
- ITAC analyzer command line
  - **`traceanalyzer --cli --messageprofile -o messages.txt ./foo.stf`**
  - **`traceanalyzer --cli --collopprofile -o messages ./foo.stf`**
  - **`traceanalyzer --cli --functionprofile -o messages ./foo.stf`**
  - https://software.intel.com/en-us/node/561584

Intel Inspector

- Use SCRATCH rather than PROJECT or HOME because they do not support mmap functionality
  - `cd $SCRATCH`

- Compile with the following settings:
  - `module swap craype-haswell craype-mic-knl`
  - `export CRAYPE_LINK_TYPE=dynamic`
  - `cc -g -O3 -debug inline-debug-info -qopt-report=3 -qopt-report-phase=vec -qopenmp`

  - Turns on -xMIC-AVX512, debugging, optimization report, OpenMP, dynamic linking

Intel Inspector

- Run the job with Inspector
  - `salloc -q interactive -C knl -N2 -t 00:30:00`
    (use --reservation if possible)
  - `module load inspector/2019.up3`
  - `export OMP_NUM_THREADS=4`
  - `export OMP_PROC_BIND=spread`
  - `export OMP_PLACES=threads`
  - `srun -n 8 -c 16 --cpu_bind=cores inspxe-cl -collect ti2 -r results/ -no-auto-finalize -- ./a.out`
  - `no-auto-finalize` to suppress finalization on compute nodes
  - Finalize with GUI or `inspxe-cl -report <> -r results/`

Intel Inspector

Available analysis types for `collect` action

- mi1 Detect Leaks
- mi2 Detect Memory Problems
- mi3 Locate Memory Problems
- ti1 Detect Deadlocks
- ti2 Detect Deadlocks and Data Races
- ti3 Locate Deadlocks and Data Races

Available report types for `report` action

- summary
- problems
- observations
- status

- Documentation
- https://software.intel.com/en-us/inspector-user-guide-linux-inspxe-cl-actions-options-and-arguments
- https://www.nersc.gov/users/software/performance-and-debugging-tools/inspector/

Thank You