TEXAS ADVANCED COMPUTING CENTER
WWW.TACC.UTEXAS.EDU

TEXAS
The University of Texas at Austin

Hands-on Session

PRESENTED BY:

Jerome Vienne

```
$ ssh -X username@stampede2.tacc.utexas.edu

$ tar -xvf ~train00/lab_knl_tools_SC17.tgz
$ cd lab_knl_tools
$ ls –l
advisor_c  advisor_f  APS gprof_c  timer_c  vtune_c  vtune_f
```

Timers: Command Line

Step 1. Start Interactive Sessions with idev

To launch a thirty-minute session on a single node in the development queue, simply execute:

$ idev -m 30

To launch a two-hour session on a single node using a specific reservation, execute:

$ idev -m 120 -r <reservation_name>

More info on how to use idev can be found using the following command:

$idev –help

Timers: Command Line

$ cd timer_c

Step 2. Compile timer.c using the following commands

$ icc timer.c -o timer

&

$ icc -xMIC-AVX512 timer.c -o timer_AVX512

Step 3. Run experiment using the following commands and compare the results

$ ./timer
$ time ./timer

&

$ ./timer_AVX512
$ time ./timer_AVX512

Gprof

cd gprof_c

Step 1. Enable profiling during compilation (use -pg option)

$ icc -g -pg -o gprof_test gprof_test.c

or

$ gcc -pg -o gprof_test gprof_test.c

Step 2. Execute the binary so that profiling data is generated

$ ./gprof_test

If the profiling is enabled then on executing the program, file gmon.out will be generated.

$ ls

gmon.out gprof_test gprof_test.c

Gprof

Step 3. Run *gprof* on profiling data

$ gprof -b gprof_test gmon.out > analysis.out

This will give an human readable file. This file contains two tables:
flat profile: overview of the timing information of the functions
call graph: focuses on each function

-*b* option will suppress lot of verbose information which would be otherwise included in analysis file.

Check the contents of analysis.out for more details.

Gprof

Step 4. Below are few more examples of gprof options

Suppress printing statically declared functions with -a option

$ gprof -b -a gprof_test gmon.out > analysis.out

Print only flat profile using -p option

$ gprof -b -p gprof_test

Print info related to specific function

$ gprof -b -pStaticFunc gprof_test

Suppress printing of flat profile using -P option

$ gprof -b -P gprof_test

Print only call graph using -q option

$ gprof -b -q gprof_test

Suppress printing of call graph using -Q option

$ gprof -b -Q gprof_test

Application Performance Snapshot

$cd APS

$module load vtune

$idev –n 32 –N 1

$ibrun aps cg.C.32

It will generate a folder with different files, You will have to run a command line to generate the text and the html file.

Ex: “aps --report=/work/01538/viennej/stampede2/lab_knl_tools/APS/aps_result_20171110”

You can copy the generated html file to look at it.

Ex: scp stampede2.tacc.utexas.edu:$PATH/lab_knl_tools/APS/aps_report_45654 .

VTune

$cd vtune_c

or

$cd vtune_f

Step 1. Set up Virtual Network Computing (VNC) or Vis portal Sessions

Please follow instructions at,

https://portal.tacc.utexas.edu/user-guides/stampede2#visualization-and-virtual-network-computing-vnc-sessions

or

https://vis.tacc.utexas.edu/

VTune

Step 2. Hotspot analysis

2.1) Compile with debug symbols

$ icc -g -O2 -xMIC-AVX512 vtune_hotspot.c
or
$ ifort -g -O2 -xMIC-AVX512 vtune_hotspot.f90

2.2) Run collection on KNL deferring finalization to host (make sure <result_dir> does NOT exist)

$module load vtune

$amplxe-cl -c hotspots -r <result_dir> -search-dir ./ ./a.out

2.3) Generate reports, work with GUI

$amplxe-cl -report hotspots -r <result_dir>
or
$amplxe-gui <result_dir>

VTune

Step 3. Memory access analyis

3.1) Compile with debug symbols

$ icc -qopenmp -g -O2 -xMIC-AVX512 ./omp_mm_knl.c -lmemkind
or
$ ifort -qopenmp -g -O2 -xMIC-AVX512 ./omp_mm_knl.f90 -lmemkind

3.2) Run collection on KNL deferring finalization to host (make sure <result_dir> does NOT exist)

$ export OMP_NUM_THREADS=64
$ amplxe-cl -c memory-access -knob analyze-mem-objects=true -r <result_dir> -search-dir ./ ./a.out

3.3) Using GUI to check results

$ amplxe-gui <result_dir>

VTune

Step 4. HPC Performance Characterization

4.1) Compile with debug symbols

$ icc -qopenmp -g -O2 -xMIC-AVX512 ./omp_mm_knl.c -lmemkind
or
$ ifort -qopenmp -g -O2 -xMIC-AVX512 ./omp_mm_knl.f90 –lmemkind

4.2) Run collection on KNL deferring finalization to host (make sure <result_dir> does NOT exist)

export OMP_NUM_THREADS=64

$ amplxe-cl -c hpc-performance -r <result_dir> -search-dir ./ ./a.out

4.3) Using GUI to check results

$ amplxe-gui <result_dir>

Advisor

Advisor Lab

Step 1. Set up Virtual Network Computing (VNC) or Vis portal Sessions

Please follow instructions at,

https://portal.tacc.utexas.edu/user-guides/stampede2#visualization-and-virtual-network-computing-vnc-sessions

or

https://vis.tacc.utexas.edu/

Advisor

Step 2. Survey

2.1) Compile with debug symbols

$ icc -g -qopenmp -O2 -xMIC-AVX512 -qopt-report=5 vector_omp.c
or
$ ifort -g -qopenmp -O2 -xMIC-AVX512 -qopt-report=5 vector_omp.f90

2.2) Run survey analysis

$ export OMP_NUM_THREADS=1

$source /opt/intel/advisor_2018.1.0.523188/advixe-vars.sh

$ advixe-cl -c survey --search-dir src:=./ -- ./a.out

2.3) Generate reports, work with GUI

$ advixe-cl -report hotspots --search-dir src:=./
or
$ advixe-gui “Open Result” (or “File” -> “Open” -> “Result”)

Advisor

Step 3. Trip count

3.1) After finish survey analysis,

$ advixe-cl -c tripcounts --search-dir src:=./ -- ./a.out

3.2) Generate reports, work with GUI

$ advixe-gui “Open Result” (or “File” -> “Open” -> “Result”)