COUNTDOWN

Hands-on

Dr. Daniele Cesarini, HPC Software Engineer, CINECA

Energy Efficiency in HPC, IT4Innovation, Ostrava (CZ), 29 January 2020

EU H2020 FETHPC project ANTAREX (g.a. 671623)

EU FP7 ERC Project MULTITHERMAN (g.a.291125)

How to log in

- Establish a ssh connection
  - ssh $USER@salomon.it4i.cz

- Remarks:
  - **ssh** available on all Linux distros
  - **putty** (free) on Windows
  - **secure shell plugin** for Google Chrome!
  - login nodes are swapped to keep the load balanced

- COUNTDOWN Github repository:
- https://github.com/EEESlab/countdown

COUNTDOWN Folder

The directory for COUNTDOWN hands-on is:
/scratch/work/project/dd-20-1/countdown

You can find the following folders:

- **countdown**: source files of COUNTDOWN
- **cntd_build**: build folder of COUNTDOWN
- **cntd_install**: install folder of COUNTDOWN where you can find the library (./lib/libcntd.so)
- **job**: in this folder you can find an example of job script for COUNTDOWN
- **NPB3.4**: it contains the NAS parallel benchmarks for testing
  - Executables: ./NPB3.4/NPB3.4-MPI/bin
- **whitelist**: it contains the MSR whitelist for Salomon
- **tool**: in this folder you can find different tools to configure the environment (hack of intel_pstate, a monitor script, etc.)

Run COUNTDOWN

This batch script will run a countdown example!
/scratch/work/project/dd-20-1/countdown/job/run_cntd.job

At the end of the job you can find the report of COUNTDOWN in:

- **$HOME/cntd_output**

```
[node]$ cd /scratch/work/project/dd-20-1/countdown/job

[node]$ qsub run_cntd.job
```

COUNTDOWN Configuration

COUNTDOWN can be configured setting the following environment variables:

- **CNTD_OUT_DIR=$PATH** Output directory of report files

- **CNTD_BARRIER=[enable/on/yes/1]** Force artificial barriers on top of collective and P2P MPI primitives
- **CNTD_EAM_SLACK=[enable/on/yes/1]** Enable COUNTDOWN Slack algorithm
- **CNTD_EAM_CALL=[enable/on/yes/1]** Enable COUNTDOWN algorithm

- **CNTD_TIME_TRACE=[enable/on/yes/1]** Enable time trace for each compute node

Add «/path/to/libcntd.so» in LD_PRELOAD environment variable to instrument your application:

```
[node]$ mpirun -np $NPROCS -genv LD_PRELOAD=/path/to/countdown/lib/libcntd.so $EXE
```

Download and Compile COUNTDOWN

```
[node]$ module load icc/2019.5.281-GCC-8.3.0-2.32 \
                impi/2019.6.154-iccifort-2019.5.281-GCC-8.3.0-2.32 \
                CMake/3.14.1 \
                hwloc/1.11.12 \
                libunwind/1.2.1 \
                Libmsr/20191216

[node]$ git clone https://github.com/EEESlab/countdown.git

[node]$ mkdir cntd_build

[node]$ cd cntd_build

[node]$ cmake -DCMAKE_INSTALL_PREFIX=$INSTALL_PREFIX ../countdown

[node]$ make && make install
```

To instrument your application with countdown:

```
[node]$ mpirun -np $NPROCS \
     -genv LD_PRELOAD=$INSTALL_PREFIX/lib/libcntd.so \
     -genv CNTD_OUT_DIR=$OUTPUT_DIR \
     -genv CNTD_EAM_SLACK=true \
     -genv CNTD_TIME_TRACE=true \
     $EXE
```

Test Application - NAS Benchmark

The NAS parallel benchmark is an open-source benchmark suite which can be downloaded and build using the following commands:

```
[node]$ wget https://www.nas.nasa.gov/assets/npb/NPB3.4.tar.gz

[node]$ tar -xvzf NPB3.4.tar.gz

[node]$ cd NPB3.4/NPB3.4-MPI

[node]$ cp config/make.def.template config/make.def

[node]$ module load icc/2019.5.281-GCC-8.3.0-2.32 \
                impi/2019.6.154-iccifort-2019.5.281-GCC-8.3.0-2.32 \
                CMake/3.14.1 \
                hwloc/1.11.12 \
                libunwind/1.2.1 \
                Libmsr/20191216

[node]$ make <benchmark-name> CLASS=<class>

[node]$ ls ./bin
```

https://www.nas.nasa.gov/publications/npb.html

COUNTDOWN Reporting

COUNTDOWN implements a reporting hierarchy.

The reports are simple CSV files, you can analyze them using your favorite data analytic system (Excel, Python, Matlab, etc.).

Report files:

- **cntd_summary.csv**: it contains the global information on the job run (execution time, energy consumption, average power, frequency, cpi, ratio on application/MPI time, etc.).
- **cntd_nodes.csv**: architectural information on the compute nodes.
- **cntd_sockets.csv**: architectural information on the sockets of the compute nodes.
- **cntd_cpus.csv**: architectural information on the CPU of the compute nodes.
- **cntd_summary_mpi.csv**: information on all MPI primitives used in the applications (execution time, data exchange, etc.).
- **cntd_summary_eam.csv** : same of “MPI information” but contains only MPI primitives taken into account from the energy-aware mechanism.
- **cntd_time_trace**: a time-based report (sample at 1s) of all the nodes. Contains the same information of nodes, sockets and CPU).

Proposed Exercises

1. Compare the execution time, the energy consumption and the average power consumption of your application (or the NAS benchmark) with and without COUNTDOWN (using both approaches). This information are in the summary report.

2. Plot in a chart the power consumption of a socket with and without COUNTDOWN. This information are in the time-based report of the nodes.

3. Create a pie chart with the most time-expensive MPI primitives.