COSC 594 Homework 7:
Performance analysis using HPCToolkit

Due March 13th 2013
Gabriel Marin <gmarin@eecs.utk.edu>

ASSIGNMENT:
This assignment is intended to give you practical experience in using HPCToolkit. For this assignment, you will use the FFTW library, a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions.

See the Appendix at the end of this assignment for information about where to find the code, how to compile it and how to run it.

1. Sampling rate: The sampling rate for sampling-based profilers can affect the accuracy of performance measurements. Sampling too frequently can perturb the execution you are examining; sampling too infrequently can lead to imprecise results. When using hardware performance counters, a sampling rate should be carefully chosen after considering the frequency of the events you aim to measure. Investigate the impact of the sampling rate on the accuracy of profiles collected using \texttt{hpcrun} out of HPCToolkit. Consider sampling using hardware counters to measure cycles (event \texttt{PAPI\_TOT\_CYC}) and primary cache misses (event \texttt{PAPI\_L1\_DCM}). Is the default sampling rate used by \texttt{hpcrun} for these events appropriate? If not, what is an appropriate sampling frequency for each of these events. Justify your answers with data.

Use your findings for the second part of the assignment.

2. Understand memory hierarchy utilization: A strength of hardware performance counters is that they can measure phenomena that cannot be measured directly otherwise. Use \texttt{hpcrun} to gather data to enable you to compute the bandwidth utilized by the FFTW benchmark at each level of the memory hierarchy: between primary (L1) data cache and the processor, between L1 and secondary (L2) cache, between L2 and tertiary (L3) cache, and finally between L3 cache and main memory. (To minimize the number of experiments you run, you may want to use \texttt{hpcrun’s} ability to collect multiple events in a single execution.) What fraction of available bandwidth is used by the application at each level of the hierarchy? Does the application appear bandwidth bound at any level of the hierarchy? Are there substantially different bandwidth rates used by the three most time consuming routines?

You will want to use the Flat View tab of the viewer, and sort by the exclusive CPU cycle counts to determine the most time consuming routines.
APPENDIX:

Use one of the AMD 10H based ICL machines, e.g. *ig* or *mordor8*, for this assignment.
- *ig* uses AMD Istanbul CPUs.
- *mordor8* uses AMD Barcelona CPUs.

Look up information about the processors used in the machine that you will be using for this assignment.

A copy of the FFTW code is located on the ICL machines in

```
/iclscratch1/homes/gmarin/CS594_Homework7
```

Extract the tarball in a local directory in your home folder.
To compile it, use ‘configure’ and ‘make’, as shown below. You must tell configure to compile the files using the ‘-g’ option, so that HPCToolkit can map the performance data to the source code. Specify the compiler options as an argument to the configuration step:

```
./configure CFLAGS="-O3 -g"
```

```
make
```

These commands will compile the library, as well as a benchmark program, *bench*, found in subfolder tests. We will use the *bench* program for this assignment.

Collect data while running the benchmark with a problem size of 256x256x256.

```
./bench 256x256x256
```