

ECE571: Advanced Microprocessor Design – Homework 1

Due: Thursday 28 January 2016, 3:30PM

1. Background

- For this assignment, log into my Haswell machine as described on the account slip that I handed out in class.
- On Linux or OSX you will do the following (replace username with the one on the slip):
`ssh -p 2131 username@weaver-lab.no-ip.org`
- On a Windows machine you'll want to get a program such as `putty`, some directions can be found here:
http://web.eece.maine.edu/~vweaver/classes/ece571_2013s/using_ssh.html
- Be sure you are connecting to port 2131 (not the default port).
- Be sure to change your password using the `passwd` command once you log in.
- We will use the `401.bzip2` benchmark from the SPEC CPU 2006 benchmark suite.
- Create a document that contains the data described in the Analysis sections below. A `.pdf` or `.txt` file is preferred but I can accept MS Office format if necessary.

2. Aggregate Event Counts

- `perf` tool
 - First copy the input file to your local directory:
`cp /opt/ece571/401.bzip2/input.source .`
 - Use the `perf` tool to gather user instruction counts for `bzip2`:
`perf stat -e instructions:u /opt/ece571/401.bzip2/bzip2 -k -f ./input.source`
 - Run the benchmark 5 times and report the instruction count for each, as well as the overall average.
 - Run the benchmark 5 more times, but this time measure user cycles rather than instructions. Report the cycle count for each, as well as the overall average.
 - Now gather and report the results for `bzip2.reverse` which is the same benchmark, but compiled with the link order reversed (reverse alphabetical rather than alphabetical).
`perf stat -e instructions:u /opt/ece571/401.bzip2/bzip2.reverse -k -f ./input.source`
Gather results for instructions (5 times) and cycles (5 times) and report the individual and overall average results.
- Questions to Answer
 - (a) Are the instruction counts deterministic, or do they vary? How large is the variation?
 - (b) Are the cycle counts deterministic, or do they vary? How large is the variation?
 - (c) Does changing the link order change the instructions or cycle metrics?

3. Sampled Results

- perf
 - Use perf to gather sampled data on the benchmark:
`time perf record -e instructions /opt/ece571/401.bzip2/bzip2 -k -f ./input.source`
 - Get a report on the most used functions; report the top 5
`perf report`
 - Use perf annotate to find out which assembly instruction caused the most CPU use:
`perf annotate`
- Valgrind DBI tool
 - Use valgrind to gather sampled data, as well as time how long it takes.
`time valgrind - --tool=callgrind /opt/ece571/401.bzip2/bzip2 -k -f ./input.source`
 - Get a report on the most used functions; report the top 5
`callgrind_annotate`
- gprof
 - The bzip2.gprof binary was compiled with -pg profiling support. Gather profiling data with it, note how long it took to run.
`time /opt/ece571/401.bzip2/bzip2.gprof -k -f ./input.source`
 - Get a report on the most used functions, report the top 5
`gprof /opt/ece571/401.bzip2/bzip2.gprof`
- Questions to Answer
 - (a) Did the three different methods of gathering function CPU use return the same results?
 - (b) What were the relative speeds of the various methods of gathering the information?

4. Skid

- Re-run the perf record / perf annotate results, but use the event `instructions:pp` instead of `instructions`
- Questions to Answer:
 - (a) Which instruction was reported as taking the most time for the two cases?
 - (b) Which do you think is more likely?
 - (c) What is the cause of this difference?

5. Submitting your work.

- Create the document containing the data as well as answers to the questions asked.
- Please make sure your name appears in the document.
- e-mail the file to me by the homework deadline.