CSC469 Assignment 1 (Part B)

Abhinav Gupta (998585963), Zeeshan Qureshi (997108954)

11 Feb 2013

Experiment

All experiments were run on CDF machines. We originally attempted to run the experiments on isolated Ubuntu virtual machines with limited memory and a single CPU but it turned out that `perf` could not access the low level information it needed inside the VM, so all numbers showed as `<not counted>`.

On CDF, we first ran both, the O2 and O3 versions of `bzip2`, `lbm`, and `perlbranch` on all the `train` size data corresponding to the programs and recorded total execution times in user space with `time`. The results were as follows:

Test Program	Total Time (O2)	Total Time (O3)	Improvement (O2/O3)
`bzip2`	66.70	69.09	0.965
`lbm`	204.50	204.08	1.002
`perlbench`	29.81	29.68	1.004

Table 1: Baseline with empty environments

As shown in the table, `lbm` and `perlbench` improved by `1.002` and `1.004` while `bzip2` degraded by `0.965`. With this baseline to compare against, we began running each experiment with varying environment sizes.

A major automation issue that we encountered was `perf`'s inability to be called from a pipe or to have its output piped to another command. If `perf` is called from within a bash pipe, it crashes with the error, `Failed opening logfd: Invalid argument`. Additionally, since `perf` does not write its output to `stdout` or `stderr`, there is no easy way to pipe its output into another command to

allow scripting around it. As per a mailing list thread[1], at least one of these issues is a known bug and has been fixed.

For the remainder of the experiment, due to lack of time, we decided to forgo all but the `perlbench` test program since that was the fastest of the three. We ran `perlbench` on all four of its inputs with 5 different configurations: 0, 250, 500, 750 and 1000 bytes. Again, we decided to experiment with only 5 configurations due to time constraints. The following table shows the total number of cycles it took `perlbench-O2` and `perlbench-O3` to execute on all their inputs under different environment sizes.

Environment Size	Cycles (O2)	Cycles (O3)	Improvement (O2/O3)
0	70605781448	69989340427	1.0088
250	70893587340	69807712968	1.0155
500	70803423147	69832274573	1.0139
750	70176850821	69138993949	1.0150
1000	69868997375	68717346336	1.0167

Table 2: Number of cycles it took to run all input through `perlbench`

There was a performance improvement in each case, ranging from `1.0088` to `1.0167`. However, there does not seem to be any direct correlation between the environment size and the performance improvement in this case.

Conclusion

From the available data, we can conclude that the environment size does not normally introduce a significant measurement bias and it is possible that observed bias may be a random artefact.

It should be noted that since the experiments were run on shared CDF machines, the numbers cannot be considered clean.

[1] http://linux-kernel.2935.n7.nabble.com/BUG-perf-annotate-broken-in-pipe-mode-td592974.html