WaveScript Benchmarks Performance Report

October 30, 2007

Machine information:
Linux faith 2.6.18-4-k7 #1 SMP Wed May 9 23:42:01 UTC 2007 i686 GNU/Linux

WaveScript SVN:
Revision: 2793

WaveScope Engine SVN:
Revision: 1495

1 Microbenchmarks

This section contains various microbenchmarks that stress the implementation of particular language constructs or data types. Figure ?? contains execution time results.

![Microbenchmarks in all WS backends](image)

Per-stream-element overheads

One thing that you can see, is that currently (2007.10) the C++/XStream engine has a high per-tuple (that is, per-element) on the communication channels relative to the ML backend. The `just_timer` test stresses this, doing nothing but passing a large number of unit tuples.
2 Language Shootout Benchmarks

This is where I will accumulate some of the small benchmarks from the language shootout. Here are some per-benchmark comments:

- **fannkuch** - “pancake flipping”. This is a translation of the gcc version of the benchmark. Tests indexed access to a small array.


3 Application Benchmarks

This section includes performance results on larger programs, namely, our current applications. Presently (2007.10) the largest of these by far is the marmot application.

4 Data Representation Profiling

This section includes an analysis of the efficiency of different data representations under different backends. This should theoretically be run on different hardware platforms as well (such as the ARM-based ensboxes).
A Appendix: Additional system information

Top results before running benchmarks:

top - 13:03:57 up 92 days, 22:36, 12 users, load average: 1.00, 1.04, 1.07
Tasks: 466 total, 1 running, 465 sleeping, 0 stopped, 0 zombie
Cpu(s): 14.3%us, 1.2%sy, 3.4%ni, 80.3%id, 0.8%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 2076424k total, 1709520k used, 366904k free, 41492k buffers
Swap: 1951856k total, 1709520k used, 366904k free, 833184k cached

Top results after running benchmarks: