Speed ver. 2, July 2005
(Test results updated July 2005)

Ziheng's naïve benchmark program

Ziheng Yang

Note: Do not change or replace files included in the package.

Thanks to **Nick Goldman** and **Andrew Rambaut** for collecting test results.

Fetching and compiling the programs

The archive is http://abacus.gene.ucl.ac.uk/software/speed2.tar.gz, which contains Win32 executables as well as C source files, which you can compile for unix/linux/OSX. Look at the Makefile and type make or use the commands like the following to compile.

```
cc -o small -O4 small.c tools.c –lm
cc -o large -O4 large.c tools.c -lm
```

To run the program, type **small** or **large** at the command prompt.

Simple descriptions

Both programs test the raw CPU speed for numerical computation. Both are sequential programs and do not use multiple processors if you have any. If you are serious about testing, there are many professional benchmarks available; see, e.g., http://www.specbench.org/.

small calculates the transition probability matrix *P*(*t*) many times. It needs about 4MB of RAM.

large runs a Markov chain. It needs about 444MB of RAM. The output is like the following (I hope the output does not depend on the platform, but I am not sure that this will be the case.)

```
  5% 0.50 0.00 1.00 0.50  0.832 0.534 0.394 0.495 0.327 -2197178046.0  0:15
 10% 0.50 0.00 1.00 0.50  0.798 0.512 0.378 0.475 0.314 -2196902682.8  0:29
…
100% 0.49 0.55 0.35 0.45  0.753 0.484 0.355 0.447 0.295 -2191882540.7  4:43
```

Test Results (Updated July 2005)

The following table lists timings we have got. These are the best results for the machine/compiler, when the program is running at the foreground, and no other program is running at the same time. If you have results for fast machines, please send me an email with information for all the fields in the table.

Computer model / OS	Compiler & options	small (~4MB)	large (~444MB)
Samsung X10+ centrino 1.8GHz, 1.5GB, winXP	MSC++ VC++6 cl -O2 –Ot	1m31s	4m52s
SunFire dual AMD 64bit Opteron 250 2.4GHz Redhat 2.3	gcc 3.2.3 -march=athlon -mcpu=athlon -O4 -funroll-loops -fomit-frame-pointer -finline-functions	1m19s	2m37s
Dual AMD athlon 1.7GHz Redhat 7	gcc 2.96 as above	5m26s	8m06s
Dual 2.4GHz Intel Xeon RedHat 2.7	gcc as above	3m34s	4m02s
Dual 2.3GHz Xserve G5 OX Server 10.3.9	gcc 3.3 -fast	3m30s	3m12s