

Re: Measurement Accuracy & ANOVA

Source: <http://sci.tech-archive.net/Archive/sci.stat.math/2008-09/msg00004.html>

- *From:* RichUlrich <rich.ulrich@xxxxxxxxxxxx>
 - *Date:* Mon, 01 Sep 2008 15:23:25 -0400
-

On Mon, 1 Sep 2008 06:25:16 -0700 (PDT), Alan <jalanthomas@xxxxxxxxxxxx> wrote:

Would someone provide background and explain "the principle of scaling up by 1000"?

If doing something once is too quick to get a good measure, try doing it 5 times, or 10 times, or 1000 times. You might look for Byte Magazine from 20 years ago for examples of simple computer testing.

From the benchmarks that I used to read, the times for **single** sets

of executions were computed directly from known cycle times and the number of instructions that had to be executed. Direct measurement of Disk-seeks, etc., where the hardware performance would depend on varying physical factors, were done by looking at the average time for numerous seeks. (The disk head was 'tossed' towards in or out, towards the right address; then it needed to read a few sectors to home in on the correct one.)

"Pipes" and "caches" and other optimizations — in the hardware and in the compiler — reduce the worth of those direct measures, unless you know enough about them to take them into account.

If you wanted to know how long it will take for one "Do-loop", you could time a single pass, or compute the duration of all the instructions. They should come out about the same, on a PC from 1988. Or time 1000 passes. That also comes out as predicted, unless the "smart" compiler says that, "Okay, nothing is being done in that loop that has any variables that are used later, so I will delete the whole loop."

For a practical benchmark, you need something that looks like a

Re: Measurement Accuracy & ANOVA

practical application, and that consists of more than one try.
It probably should also compete with other threads/processes
that are active on computers today.

--
Rich Ulrich

.