

**Problem 1:** Section 2.2.2 of the run and reporting rules for SPECcpu2006, <http://www.spec.org/cpu2006/Docs/runrules.html>, specifies that the optimization flags and options used to obtain the base result must be the same for each benchmark (compiled with the same language, say C). Why must they be the same?

**Problem 2:** Section 1.2.3 of the run and reporting rules for SPECcpu2006, <http://www.spec.org/cpu2006/Docs/runrules.html>, assumes that the tester is honest. Provide an argument that many of the run and reporting rules ignore this assumption, or at best are based on the assumption that the tester is honest but sloppy or unmotivated.

**Problem 3:** Find the SPECcpu2000 CINT2000 disclosure for the fastest systems using each of the chips below. All chips implement some form or superset of IA-32 (also known as 80x86). All of the implementations are superscalar, meaning they can sustain execution of more than one instruction per cycle. In particular, an  $n$ -way superscalar processor can sustain execution of  $n$  instructions per cycle on ideal code, on real code the sustained execution rate is much lower (for reasons to be covered later in the course, such as cache misses). Some of the implementations are multi-cored. (A core is an entire processor and so a 2-core chip has two complete processors.)

- Pentium III, 1-core, 2-way
- Pentium 4, 1-core, 3-way
- Pentium Extreme, 2-core, 3-way
- Intel Core 2 Extreme X6800, 2-core, 4-way
- Opteron 256, 1-core, 3-way
- Athlon FX-62, 2-core, 3-way

(a) For each system list the following information:

- The peak (result) ratio (for the suite).
- The clock frequency.
- The gcc peak (result) run time (in seconds).
- The maximum number of instructions the system could have executed during the run of gcc assuming all cores were used.
- The maximum number of instructions the system could have executed during the run of gcc assuming one core was used.
- Execution efficiency assuming all cores were used: number of instructions executed divided by maximum number of instructions that could have been executed in the same amount of time. Assume that all systems run the same binary (executable) of gcc and make a guess at how many instructions would be executed when running the binary for the SPEC inputs.
- Execution efficiency assuming a single core was used: Same as previous value, except assume only one core used.

(b) The execution efficiency computation was based on the assumption that the number of executed instructions was the same in all systems. Identify two systems for which this was more likely to be true and two systems where this was less likely to be true.

(c) How much does a dual-core implementation improve the performance of gcc?