1.9 Real stuff: Benchmarking the Intel Core i7

> I thought [computers] would be a universally applicable idea, like a book is. But I didn't think it would develop as fast as it did, because I didn't envision we'd be able to get as many parts on a chip as we finally got. The transistor came along unexpectedly. It all happened much faster than we expected.
> *J. Presper Eckert, coinventor of ENIAC, speaking in 1991*

Each chapter has a section entitled "Real Stuff " that ties the concepts in the book with a computer you may use every day. These sections cover the technology underlying modern computers. For this first "Real Stuff" section, we look at how integrated circuits are manufactured and how performance and power are measured, with the Intel Core i7 as the example.

SPEC CPU benchmark

A computer user who runs the same programs day in and day out would be the perfect candidate to evaluate a new computer. The set of programs run would form a *workload*. To evaluate two computer systems, a user would simply compare the execution time of the workload on the two computers. Most users, however, are not in this situation. Instead, they must rely on other methods that measure the performance of a candidate computer, hoping that the methods will reflect how well the computer will perform with the user's workload. This alternative is usually followed by evaluating the computer using a set of *benchmarks*—programs specifically chosen to measure performance. The benchmarks form a workload that the user hopes will predict the performance of the actual workload. As we noted above, to make the *common case fast*, you first need to know accurately which case is common, so benchmarks play a critical role in computer architecture.

Workload: A set of programs run on a computer that is either the actual collection of applications run by a user or constructed from real programs to approximate such a mix. A typical workload specifies both the programs and the relative frequencies.

Benchmark: A program selected for use in comparing computer performance.

SPEC (***System Performance Evaluation Cooperative***) is an effort funded and supported by a number of computer vendors to create standard sets of benchmarks for modern computer systems. In 1989, SPEC originally created a benchmark set focusing on processor performance (now called SPEC89), which has evolved through five generations. The latest is SPEC CPU2006, which consists of a set of 12 integer benchmarks (CINT2006) and 17 floating-point benchmarks (CFP2006). The integer benchmarks vary from part of a C compiler to a chess program to a quantum computer simulation. The floating-point benchmarks include structured grid codes for finite element modeling, particle method codes for molecular dynamics, and sparse linear algebra codes for fluid dynamics.

The figure below describes the SPEC integer benchmarks and their execution time on the Intel Core i7 and shows the factors that explain execution time: instruction count, CPI, and clock cycle time. Note that CPI varies by more than a factor of 5.

Table 1.9.1: SPECINTC2006 benchmarks running on a 2.66GHz Intel Core i7 920 (COD Figure 1.18).

Execution time is the product of the three factors in this table: instruction count in billions, clocks per instruction (CPI), and clock cycle time in nanoseconds. SPECratio is simply the reference time, which is supplied by SPEC, divided by the measured execution time. The single number quoted as SPECINTC2006 is the geometric mean of the SPECratios.

Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2252	0.60	0.376	508	9770	19.2
Block-sorting compression	bzip2	2390	0.70	0.376	629	9650	15.4
GNU C compiler	gcc	794	1.20	0.376	358	8050	22.5
Combinatorial optimization	mcf	221	2.66	0.376	221	9120	41.2
Go game (AI)	go	1274	1.10	0.376	527	10490	19.9
Search gene sequence	hmmer	2616	0.60	0.376	590	9330	15.8
Chess game (AI)	sjeng	1948	0.80	0.376	586	12100	20.7
Quantum computer simulation	libquantum	659	0.44	0.376	109	20720	190.0
Video compression	h264avc	3793	0.50	0.376	713	22130	31.0
Discrete event simulation library	omnetpp	367	2.10	0.376	290	6250	21.5
Games/path finding	astar	1250	1.00	0.376	470	7020	14.9

XML parsing	xalancbmk	1045	0.70	0.376	275	6900	25.1
Geometric mean	-	-	-	-	-	-	25.7

To simplify the marketing of computers, SPEC decided to report a single number summarizing all 12 integer benchmarks. Dividing the execution time of a reference processor by the execution time of the evaluated computer normalizes the execution time measurements; this normalization yields a measure, called the ***SPECratio***, which has the advantage that bigger numeric results indicate faster performance. That is, the SPECratio is the inverse of execution time. A CINT2006 or CFP2006 summary measurement is obtained by taking the geometric mean of the SPECratios.

Elaboration

When comparing two computers using SPECratios, apply the geometric mean so that it gives the same relative answer no matter what computer is used to normalize the results. If we averaged the normalized execution time values with an arithmetic mean, the results would vary depending on the computer we choose as the reference.

The formula for the geometric mean is

$$\sqrt[n]{\prod_{i=1}^{n} \textbf{Execution time ratio}_i}$$

where Execution time ratio$_i$ is the execution time, normalized to the reference computer, for the ith program of a total of n in the workload, and

$$\prod_{i=1}^{n} a_i \textbf{ means the product } a_1 \times a_2 \times \ldots \times a_n$$

SPEC power benchmark

Given the increasing importance of energy and power, SPEC added a benchmark to measure power. The SPEC power benchmark reports power consumption of servers at different workload levels, divided into 10% increments, over a period of time. The figure below shows the results for a server using Intel Nehalem processors similar to the above.

Table 1.9.2: SPECpower_ssj2008 running on a dual socket 2.66GHz Intel Xeon X5650 with 16GB of DRAM and one 100GB SSD disk (COD Figure 1.19).

Target Load %	Performance (ssj_ops)	Average Power (watts)
100%	865,618	258
90%	786,688	242
80%	698,051	224
70%	607,826	204
60%	521,391	185
50%	436,757	170
40%	345,919	157
30%	262,071	146
20%	176,061	135
10%	86,784	121
0%	0	80
Overall Sum	4,787,166	1922
Σssj_ops / Σpower =		2490

SPECpower started with another SPEC benchmark for Java business applications (SPECJBB2005), which exercises the processors, caches, and main memory as well as the Java virtual machine, compiler, garbage collector, and pieces of the operating system. Performance is measured in throughput, and the units are business operations per second. Once again, to simplify the marketing of computers, SPEC boils these numbers down to one number, called "overall ssj_ops per watt." The formula for this single summarizing metric is

$$\textbf{overall ssj\_ops per watt} = \left(\sum_{i=0}^{10} \textbf{ssj\_ops}_i\right) \Big/ \left(\sum_{i=0}^{10} \textbf{power}_i\right)$$

where ssj_ops$_i$ is performance at each 10% increment and power$_i$ is power consumed at each performance level.

PARTICIPATION ACTIVITY 1.9.1: SPEC benchmark.

1) "Go game (AI)" is a benchmark.

- True
- False

2) A workload is typically a set of benchmarks.

- True
- False

3) All SPEC benchmarks have similar instruction count, CPI, and clock cycle time characteristics.

- True
- False

4) A smaller SPECratio indicates better performance.

- True
- False

5) Power is not a useful evaluation metric in the SPEC benchmarks.

- True
- False