Benchmarking

Lecture Objectives:

1) Explain the SPEC benchmarks.
2) Define Amdahl's law
3) Define MIPS

Measuring
Performance
Changes

In Class Activity

- Download the spec benchmark suite from the course website
- Install it to your computer
  - Any location is fine so long as you have write access…
- Execute it on your computer
- We'll come back to the results later on at the end of lecture…

Done already

How do we measure performance?

- What is MIPS?

Million Instructions per second

While good, it's terrible!

Problems with MIPS

- Specification of instruction rate, not complexity of the instructions performed

- Varies between programs on a single computer → Word / Excel

- MIPS may vary independently from performance ⇒ As we will se

NOP → Does nothing

Performance Problem

- I have a computer program which takes 100 seconds to execute. Of this time, 80 seconds is spent downloading data over the network. How much faster must the download be in order to obtain a program which runs 5x faster?

5x improvement

$\frac{100}{5} = 20s$ total time

Not making any changes to computation

∴ Download time would have to be 0. AKA Not possible.

Amdahl's Law

- *For over a decade prophets have voiced the contention that the organization of a single computer has reached its limits and that truly significant advances can be made only by interconnection of a multiplicity of computers in such a manner as to permit co-operative solution...The nature of this overhead (in parallelism) appears to be sequential so that it is unlikely to be amenable to parallel processing techniques. Overhead alone would then place an upper limit on throughput of five to seven times the sequential processing rate, even if the housekeeping were done in a separate processor...At any point in time it is difficult to foresee how the previous bottlenecks in a sequential computer will be effectively overcome.- Gene Amdahl*

Amdahl's Law in English...

- The performance enhancement possible with a given improvement is limited by the amount that the improved feature is used.
  - Law of diminishing returns

$$ExecutionTime_{Improved} = \frac{ExecutionTime_{AffectedByImprovement}}{AmountOfImprovement} + ExecutionTime_{Unaffected}$$

Amdahl's Law Applied

- A Program spends 40 seconds performing network transfers and 60 seconds generating reports. What improvement in performance in the report generator would be necessary to increase the speed of the program by a factor of 2?

Total base time = 100 s

Total improved time = 50 s

$$T_I = \frac{T_C}{\text{Factor}} + T_U$$

$$50\,s = \frac{60}{n} + 40$$

$$50 = \frac{60}{n} + 40$$

$$10 = \frac{60}{n}$$

$$10n = 60$$

$$n = 6$$

Amdahl's Law Corollary

- Make the common case fast…

Improve the speed of things which we do often at the expense of things we do rarely

Benchmarking

- How do you decide which computer to buy for your application?

AMD Quad-Core Opteron Vs. Intel Quad-Core Xeon Processor				
Data by SPEC.org				
CINT2006 Rates				
	Clock-speed	Number of chips	Base	Peak
Intel Xeon E5320	1.86GHz	2	67.70	80.20
AMD Opteron 2347	1.90GHz	2	72.80	82.30
AMD Opteron 2350	2.00GHz	2	77.30	88.30
AMD Opteron 2360 SE	2.50GHz	2	88.20	102.00
Intel Xeon E5335	2.00GHz	2	78.10	92.20
Intel Xeon E5345	2.33GHz	2	85.70	102.00
Intel Xeon E5355	2.66GHz	2	91.60	109.00
Intel Xeon X5365	3.00GHz	2	97.90	116.00
CFP2006 Rates				
Intel Xeon E5320	1.86GHz	2	45.1	47.7
AMD Opteron 2347	1.90GHz	2	68.5	73
AMD Opteron 2350	2.00GHz	2	72.4	77.3
AMD Opteron 2360 SE	2.50GHz	2	80.6	86.3
Intel Xeon E5335	2.00GHz	2	53.4	56.8
Intel Xeon E5345	2.33GHz	2	57.2	60.8
Intel Xeon E5355	2.66GHz	2	60.10	63.30
Intel Xeon X5365	3.00GHz	2	63.10	66.90
Intel Xeon 5400/45nm	3.20GHz	2	89.80	89.80

SPEC Benchmark

- Programs used to measure performance
  - Supposedly typical of actual workload
- Standard Performance Evaluation Corp (SPEC)
  - Develops benchmarks for CPU, I/O, Web, …
- SPEC CPU2006
  - Elapsed time to execute a selection of programs
    - Negligible I/O, so focuses on CPU performance
  - Normalize relative to reference machine
  - Summarize as geometric mean of performance ratios
    - CINT2006 (integer) and CFP2006 (floating-point)

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

CINT2006 for Opteron X4 2356

Name	Description	IC×10⁹	CPI	Tc (ns)	Exec time	Ref time	SPECratio
perl	Interpreted string processing	2,118	0.75	0.40	637	9,777	15.3
bzip2	Block-sorting compression	2,389	0.85	0.40	817	9,650	11.8
gcc	GNU C Compiler	1,050	1.72	0.47	24	8,050	11.1
mcf	Combinatorial optimization	336	10.00	0.40	1,345	9,120	6.8
go	Go game (AI)	1,658	1.09	0.40	721	10,490	14.6
hmmer	Search gene sequence	2,783	0.80	0.40	890	9,330	10.5
sjeng	Chess game (AI)	2,176	0.96	0.48	37	12,100	14.5
libquantum	Quantum computer simulation	1,623	1.61	0.40	1,047	20,720	19.8
h264avc	Video compression	3,102	0.80	0.40	993	22,130	22.3
omnetpp	Discrete event simulation	587	2.94	0.40	690	6,250	9.1
astar	Games/path finding	1,082	1.79	0.40	773	7,020	9.1
xalancbmk	XML parsing	1,058	2.70	0.40	1,143	6,900	6.0
Geometric mean							11.7

SPEC and power

- Power consumption of server at different workload levels
  - Performance: ssj_ops/sec
  - Power: Watts (Joules/sec)

Not Desktop/laptop

$$\text{Overall ssj\_ops per Watt} = \left(\sum_{i=0}^{10} \text{ssj\_ops}_i\right) \Bigg/ \left(\sum_{i=0}^{10} \text{power}_i\right)$$

operations per Watt

SPEC Power versus load
SPECpower_ssj2008 for X4

Target Load %	Performance (ssj_ops/sec)	Average Power (Watts)
100%	231,867	295
90%	211,282	286
80%	185,803	275
70%	163,427	265
60%	140,160	256
50%	118,324	246
40%	920,35	233
30%	70,500	222
20%	47,126	206
10%	23,066	180
0%	0	141
Overall sum	1,283,590	2,605
∑ssj_ops/ ∑power		493

Lots of power doing nothing

Low power at low usage?

- Look back at X4 power benchmark
  - At 100% load: 295W
  - At 50% load: 246W (83%)
  - At 10% load: 180W (61%)
- Google data center
  - Mostly operates at 10% – 50% load
  - At 100% load less than 1% of the time
- Consider designing processors to make power proportional to load

MSOE

What are we doing in the background

- Running specjvm2008
  - SPECjvm2008 is a benchmark suite for measuring the performance of a Java Runtime Environment (JRE), containing several real life applications and benchmarks focusing on core java functionality. The SPECjvm2008 workload mimics a variety of common general purpose application computations.