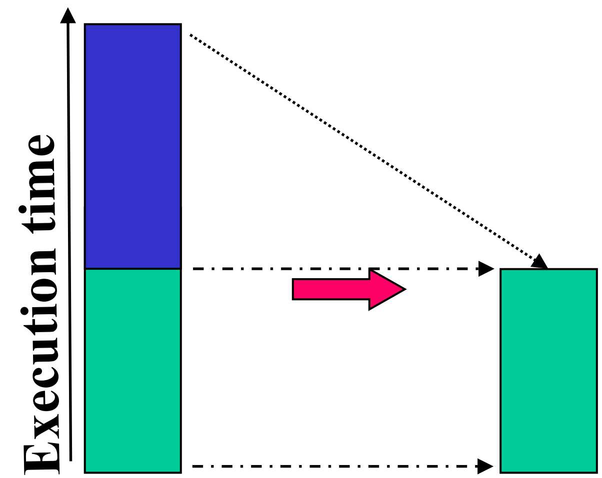
Lecture 4

14-01-2021

Amdahl's Law: Exercise 2

- A software engineer decides to rewrite a portion of a sequential program that accounts for 60% of execution time of the program:

  - **So that this portion can be run on multiple processors in parallel.**



- What is the maximum speedup that the software engineer can hope to achieve?

- $\boxed{100/40=2.5}$

Performance Measurements

- Performance measurement is important:
  - Helps us to determine if one processor (or computer) works faster than another.
  - **Intuitively, a computer has higher performance if it executes programs faster…**

How to Select A Computer System?

- Suppose you are working in a company.
- You have been asked by your manager to select a computer for some specific application:
  - **Lets say: to run a web service for your organization…**
- **How will you proceed?**

Purchasing Decision by a Novice…

- Computer A has a 2.5Ghz processor
- Computer B has a 3.2GHz processor
- Which one is faster?

B is naturally faster…

Simpleton

Obviously WRONG!

Where is the fallacy?

Comparing Computer Performance

- We say computer X is n times faster than Y, iff

$$\frac{\text{Execution time}_Y}{\text{Execution time}_X} = n$$

Relative Performance

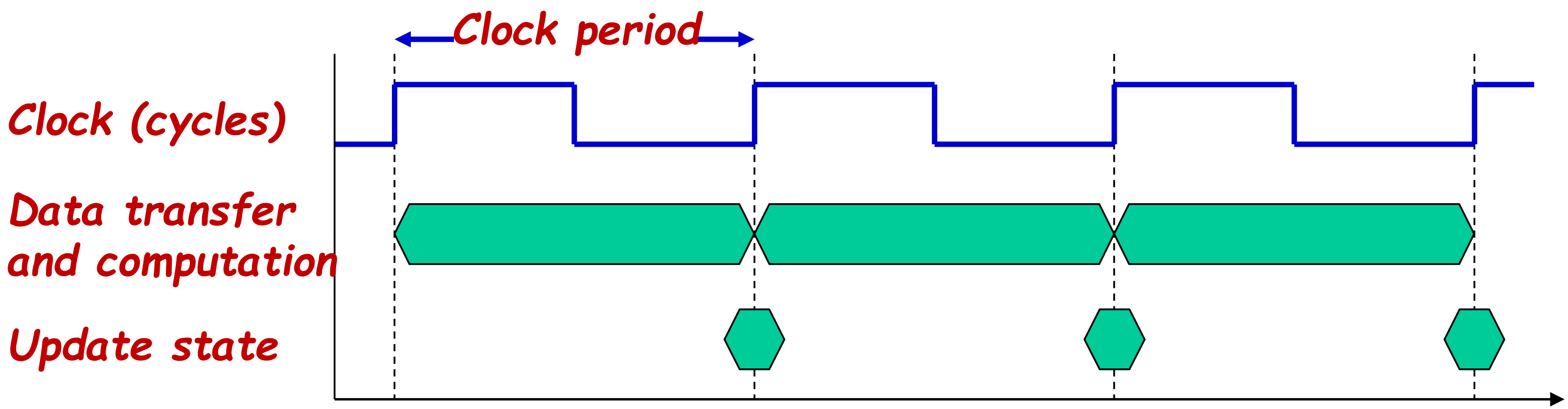
- Define Performance = 1/Execution Time
- "X is n time faster than Y"

$$\text{Performance}_X / \text{Performance}_Y = \text{Execution time}_Y / \text{Execution time}_X = n$$

- *Example: time taken to run a program*
  - *10s on A, 15s on B*
  - *Execution Time$_B$ / Execution Time$_A$ = 15s / 10s = 1.5*
  - *So A is 1.5 times faster than B*

CPU Clocking

- All operation of computer are governed by a constant-rate clock…



- *Clock period: duration of a clock cycle*
  - *e.g., 250ps = 0.25ns = 250×10^{-12}s*
- *Clock frequency (rate): cycles per second*
  - *e.g., 4.0GHz = 4000MHz = 4.0×10^{9}Hz*

CPU Time

$$\text{CPU Time} = \text{CPU Clock Cycles} \times \text{Clock Cycle Time}$$

$$= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}}$$

- Performance can be improved by:
  - Reducing number of clock cycles
  - Increasing clock rate
  - Hardware designer must often trade off clock rate against cycle count

CPU Time Exercise

- Computer A: 2 GHz clock, takes 10s CPU time
- Computer B:
  - Completes the job in 6s CPU time
  - Has faster clock, but uses 1.2 × clock cycles
  - What is Computer B's clock rate?

$$\text{Clock Rate}_B = \frac{\text{Clock Cycles}_B}{\text{CPU Time}_B} = \frac{1.2 \times \text{Clock Cycles}_A}{6\text{s}}$$

$$\text{Clock Cycles}_A = \text{CPU Time}_A \times \text{Clock Rate}_A$$

$$= 10\text{s} \times 2\text{GHz} = 20 \times 10^9$$

$$\text{Clock Rate}_B = \frac{1.2 \times 20 \times 10^9}{6\text{s}} = \frac{24 \times 10^9}{6\text{s}} = 4\text{GHz}$$

Instruction Count and CPI

$$\text{Clock Cycles} = \text{Instruction Count} \times \text{Cycles per Instruction}$$

$$\text{CPU Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time}$$

$$= \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}$$

- Instruction Count (IC) for a program:
  - Determined by program, ISA and compiler
- Average cycles per instruction (CPI):
  - Determined by CPU hardware
  - If different instructions have different CPI
    - **Average CPI affected by instruction mix**

Exercise

- Computer A: Cycle Time = 250ps, CPI = 2.0
- Computer B: Cycle Time = 500ps, CPI = 1.2
- Same ISA
- Which is faster, and by how much?

$$\text{CPU Time}_A = \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A$$
$$= \text{I} \times 2.0 \times 250\text{ps} = \text{I} \times 500\text{ps}$$
$$\text{CPU Time}_B = \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B$$
$$= \text{I} \times 1.2 \times 500\text{ps} = \text{I} \times 600\text{ps}$$
$$\frac{\text{CPU Time}_B}{\text{CPU Time}_A} = \frac{\text{I} \times 600\text{ps}}{\text{I} \times 500\text{ps}} = 1.2$$

A is faster…

…by this much

Exercise

- A given application written in Java runs in 15 seconds on a desktop processor.
  - A new Java compiler is released that requires only 0.6 as many instructions as the old compiler.
  - Unfortunately, it increases the CPI by 1.1.
- How fast can we expect the application to run using this new compiler?

Ans: 15 sec × 0.66= 9.9 sec

Clock-Rate Based Performance Measurement

- Comparing performance based on clock rates alone, is obviously meaningless:
  - **Execution time=CPI × Clock cycle time × Number of instructions**
  - Caveat:
    - A processor with a higher clock rate may actually execute programs much slower!

Exercise: Which Machine is Faster and what is the speedup?

- **Machine A for prog X**
  - Clock cycle time = 10ns/cycle
  - CPI = 2.0

- **Machine B for prog X**
  - Clock cycle time = 30ns/cycle
  - CPI = 0.5

Assume same ISA.

Let I = number of instructions in the program.

CPU clock cycles (A) = I * 2.0
CPU time (A) = CPU clock cycles * clock cycle time
= I * 2.0 * 10
= I * 20 ns

CPU clock cycles (B) = I * 0.5
CPU time (B) = CPU clock cycles * clock cycle time
= I * 0.5 * 30
= I * 15 ns

$$\frac{Performance(A)}{Performance(B)} = \frac{Execution(B)}{Execution(A)} = 0.75$$

Exercise: Calculate Overall CPI (Cycles per Instruction)

Operation	Freq	CPI(i)
ALU	40%	1
Load	27%	2
Store	13%	2
Branch	20%	5

Typical Instruction Mix

Overall CPI= 1*0.4+ 2*0.27+ 2*0.13+5*0.2

= 2.2

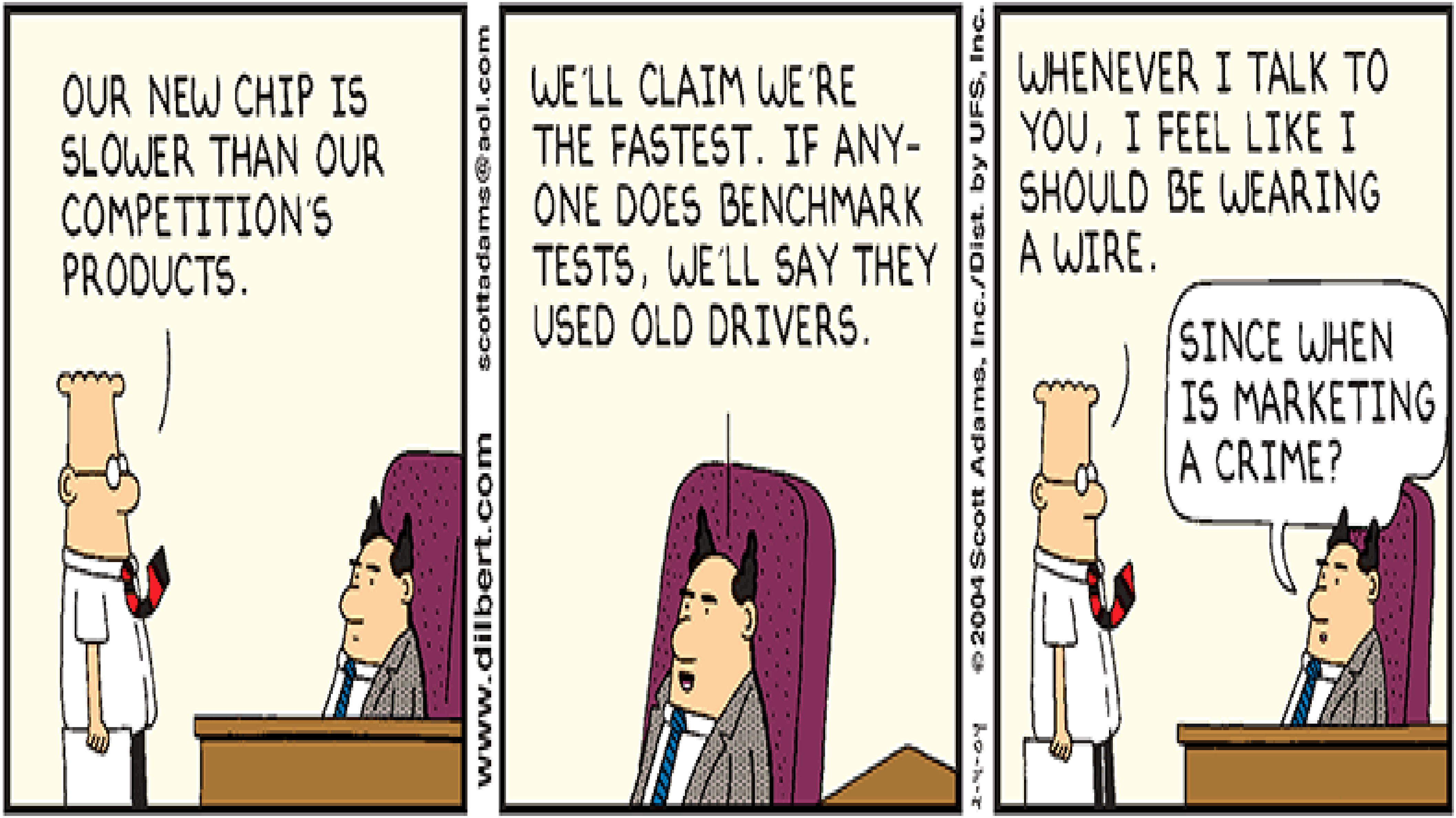
MIPS and MFLOPS

- Used extensively 30 years back.
- MIPS: millions of instructions processed per second.
- MFLOPS: Millions of FLoating point OPerations completed per Second

$$\text{MIPS} = \frac{\text{Instruction Count}}{\text{Exec. Time} \times 10^6} = \frac{\text{Clock Rate}}{\text{CPI} \times 10^6}$$

Problems with MIPS

- MIPS suffers from several severe shortcomings.
- So severe, made some one coin:
  - **"Meaningless Information about Processing Speed"**
- **A Major Problem:**
  - MIPS depends on the program being executed to measure MIPS.


OUR NEW CHIP IS SLOWER THAN OUR COMPETITION'S PRODUCTS.
scottadams@aol.com
www.dilbert.com
WE'LL CLAIM WE'RE THE FASTEST. IF ANYONE DOES BENCHMARK TESTS, WE'LL SAY THEY USED OLD DRIVERS.
© 2004 Scott Adams, Inc./Dist. by UFS, Inc.
WHENEVER I TALK TO YOU, I FEEL LIKE I SHOULD BE WEARING A WIRE.
SINCE WHEN IS MARKETING A CRIME?

Exercise 4

- A computer A-100 executes a certain benchmark program with an average CPI of 2.0.
- Another computer A-101 has the same instruction set but has an enhanced compiler.
- The compiler for A-101 generated 20% less instructions for the same benchmark program.
- The compiled code when run on A-101 achieved a CPI of 0.5 at 800MHz.
- In order for the two machines to have the same MIPS rating, what should be the clock rate of the computer A-100?

Exercise 4: Answer

- Let I be the number of instructions in the benchmark program
- A-101 executes 0.8*I instructions in 0.8*I*CPI(A-101)*C(A-101) seconds
- $= 0.8*I*0.5/(800*10^{6})$ seconds
- MIPS= 800/0.5= 1600
- MIPS of A-100=$I*10^{-6}/(I*2*C)=10^{-6}/(2*C)$
- Frequency of A-100=3200 MHz = 3.2GHz

Toy Benchmarks

- The performance of different computers can be compared by running some standard programs:
  - Quick sort, Merge sort, etc.
- But, the basic problem remains:
  - Even if a computer performs well for a toy benchmark, it may not perform well for the required applications.
  - **What can be a solution then?**

Synthetic Benchmarks

- Basic Principle: Analyze the distribution of instructions over a large number of practical programs.
- Synthesize a program that has the same instruction distribution as a typical program:
  - Need not compute something meaningful.
- Dhrystone, Khornerstone, Linpack are some of the older synthetic benchmarks:
  - More recent is **SPEC**…

Dhrystone

- Short synthetic benchmark program
  - Largely uses integer programming.
  - Based on published statistics on use of programming language features;
  - original publication in CACM 27,10 (Oct. 1984),
- **Problems:** Due to its small size (100 HLL statements, about 1.5 KB code), the memory system outside the cache is not tested;
  - **Compilers can easily optimize for Dhrystone.**

Linpack

- Developed from the "LINPACK" package of linear algebra routines.
  - Originally written in FORTRAN; a C version also exists.
- Almost all of the benchmark's time is spent in a subroutine executing the inner loop for frequent matrix operations:
  - Such as **y(i) = y(i) + a * x(i)**
- The standard version operates on 100x100 matrices;
  - There are also versions for sizes 300x300 and 1000x1000, with different optimization rules.
- **Problems:**
  - Code mostly concerns for matrix computation.
  - LINPACK is easily vectorizable on most systems.

Problems with Benchmarks

- Linpack consists of different types of operations on 300*300 matrices.
  - Optimization of this inner-loop resulted in performance improvement by a factor of 9.
- Optimizing compilers can discard 25% Dhrystone code
- Solution: **Benchmark suite**

SPEC Benchmarks

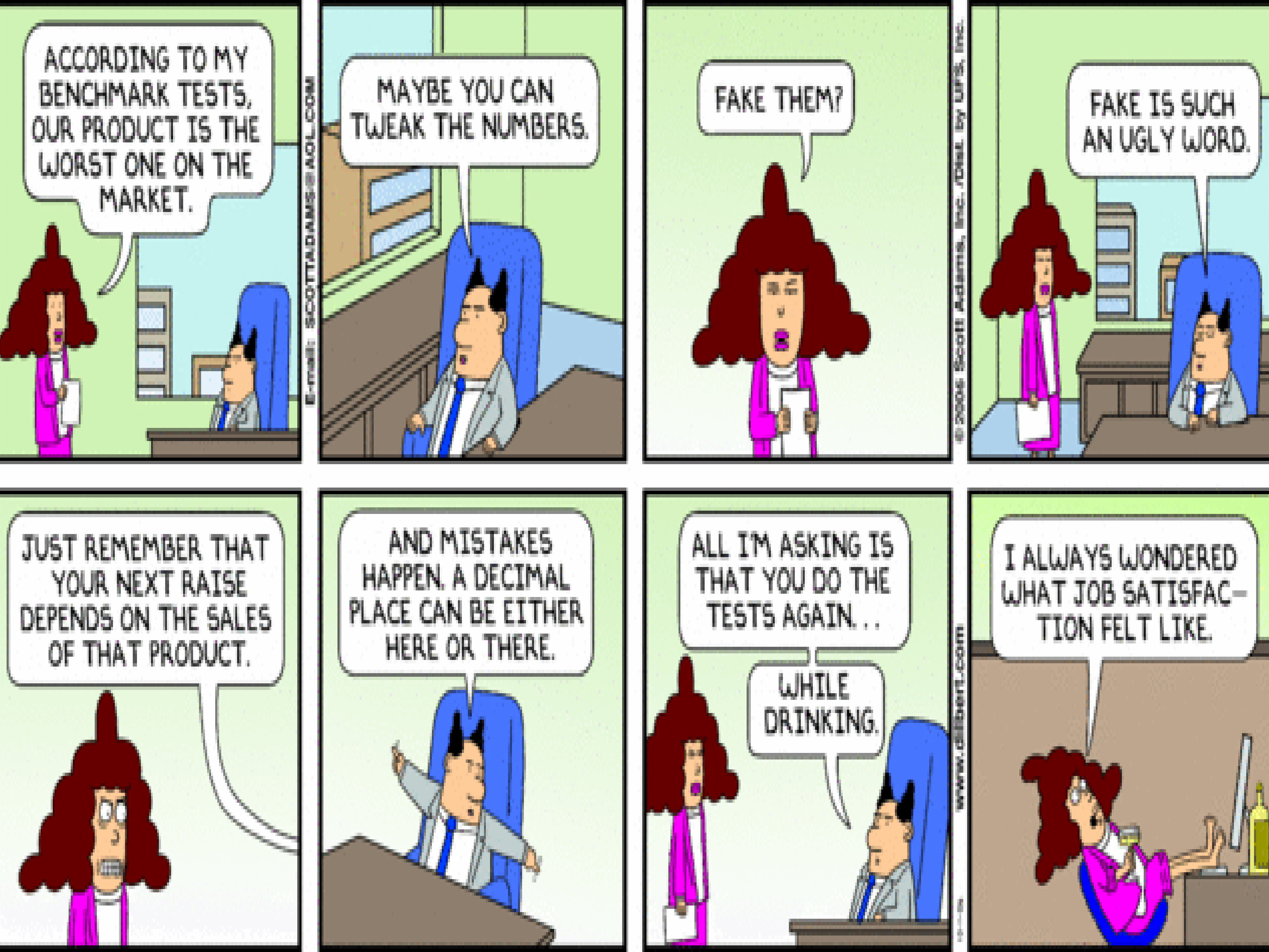
- SPEC: Standard Performance Evaluation Corporation:
  - A non-profit organization (www.spec.org)
  - **Uses a suite of programs**
- CPU-intensive benchmark for evaluating processor performance of workstation:
  - Generations: SPEC89, SPEC92, SPEC95, and SPEC2000 …
  - Emphasizes memory system performance in SPEC2000.

Other SPEC Benchmarks

- **SPECviewperf:** 3D graphics performance
  - For applications such as CAD/CAM, visualization, content creations, etc.
- **SPEC JVM98:** performance of client-side Java virtual machine.
- **SPEC JBB2000:** Server-side Java application
- **SPEC WEB2005:** evaluating WWW servers
  - Contains multiple workloads utilizing both http and https, dynamic content implemented in PHP and JSP.

Summary of SPEC Benchmarks

- CPU: CPU2006
- Graphics: SPECviewperf
- Java Client/Server: jAppServer2004
- Mail Servers: MAIL2001
- Network File System: SDS97_R1
- Power (under development)
- Web Servers: WEB2005


ACCORDING TO MY BENCHMARK TESTS, OUR PRODUCT IS THE WORST ONE ON THE MARKET.
MAYBE YOU CAN TWEAK THE NUMBERS.
FAKE THEM?
FAKE IS SUCH AN UGLY WORD.
JUST REMEMBER THAT YOUR NEXT RAISE DEPENDS ON THE SALES OF THAT PRODUCT.
AND MISTAKES HAPPEN. A DECIMAL PLACE CAN BE EITHER HERE OR THERE.
ALL I'M ASKING IS THAT YOU DO THE TESTS AGAIN. . .
WHILE DRINKING.
I ALWAYS WONDERED WHAT JOB SATISFAC- TION FELT LIKE.
E-mail: SCOTTADAMS@AOL.COM
© 2004 Scott Adams, Inc. /Dist. by UFS, Inc.
www.dilbert.com