inst.eecs.berkeley.edu/~cs61c

# CS61C : Machine Structures

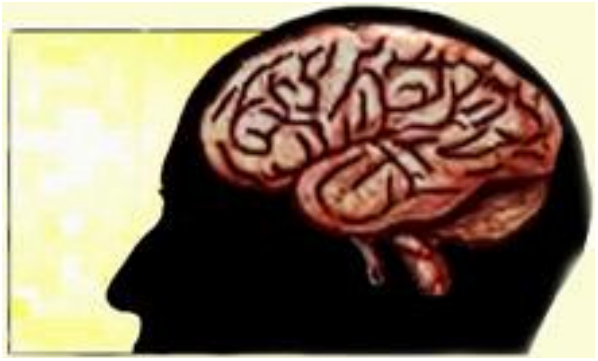
## Lecture 42
## Performance II

**2004-12-08**

**Lecturer PSOE Dan Garcia**

www.cs.berkeley.edu/~ddgarcia

**Mind control ⇒**



**Researchers have found a way to non-invasively control a cursor in 2D wearing a hat with electrodes!**

**This puts the I in I/O**

www.cnn.com/2004/TECH/12/07/computer.thought.reut/

# Review

- Latency v. Throughput
- Performance doesn't depend on any single factor: need to know Instruction Count, Clocks Per Instruction (CPI) and Clock Rate to get valid estimations
- User Time: time user needs to wait for program to execute: depends heavily on how OS switches between tasks
- CPU Time: time spent executing a single program: depends solely on design of processor (datapath, pipelining effectiveness, caches, etc.)

$$\text{CPU time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

# Example Standardized Benchmarks (1/2)

- Standard Performance Evaluation Corporation (SPEC) SPEC CPU2000
  - CINT2000 12 integer (gzip, gcc, crafty, perl, ...)
  - CFP2000 14 floating-point (swim, mesa, art, ...)
  - All relative to base machine Sun 300MHz 256Mb-RAM Ultra5_10, which gets score of 100
  - `www.spec.org/osg/cpu2000/`
  - They measure
    - System speed (SPECint2000)
    - System throughput (SPECint_rate2000)

# Example Standardized Benchmarks (2/2)

- SPEC
  - Benchmarks distributed in source code
  - Members of consortium select workload
    - 30+ companies, 40+ universities
  - Compiler, machine designers target benchmarks, so try to change every 3 years
  - The last benchmark released was SPEC 2000
    - They are still finalizing SPEC 2004

# Example PC Workload Benchmark

- PCs: Ziff-Davis Benchmark Suite
  - "Business Winstone is a system-level, application-based benchmark that measures a PC's overall performance when running today's top-selling Windows-based 32-bit applications… it doesn't mimic what these packages do; it runs real applications through a series of scripted activities and uses the time a PC takes to complete those activities to produce its performance scores.
  - Also tests for CDs, Content-creation, Audio, 3D graphics, battery life

`http://www.etestinglabs.com/benchmarks/`

# Other PC benchmarking resources

Thanks to Robert van Spyk for these leads…

- [H]ardIOCP (http://www.hardocp.com/) - "is an online site where hardware-expert-gamers outline new features and run a number of tests (including overclock limits) on cutting-edge hardware (CPU, motherboard, videoboard, modified cases, etc)."
- 3DMark and PCMark (http://www.futuremark.com/) - "are commonly used benchmarks to demonstrate the aggregate power of a system for common applications. 3DMark runs eye-candy pretty demos of games from most genres using the newest technology. PCMark tests operating system, multimedia, and office application performance (neat tests list at www.futuremark.com/products/pcmark04/?tests)."
- SiSoft Sandra (http://www.sisoftware.net/) - "Tons of standard benchmarks and also information tools. Used with the above to generate bragging rights"

# Performance Evaluation

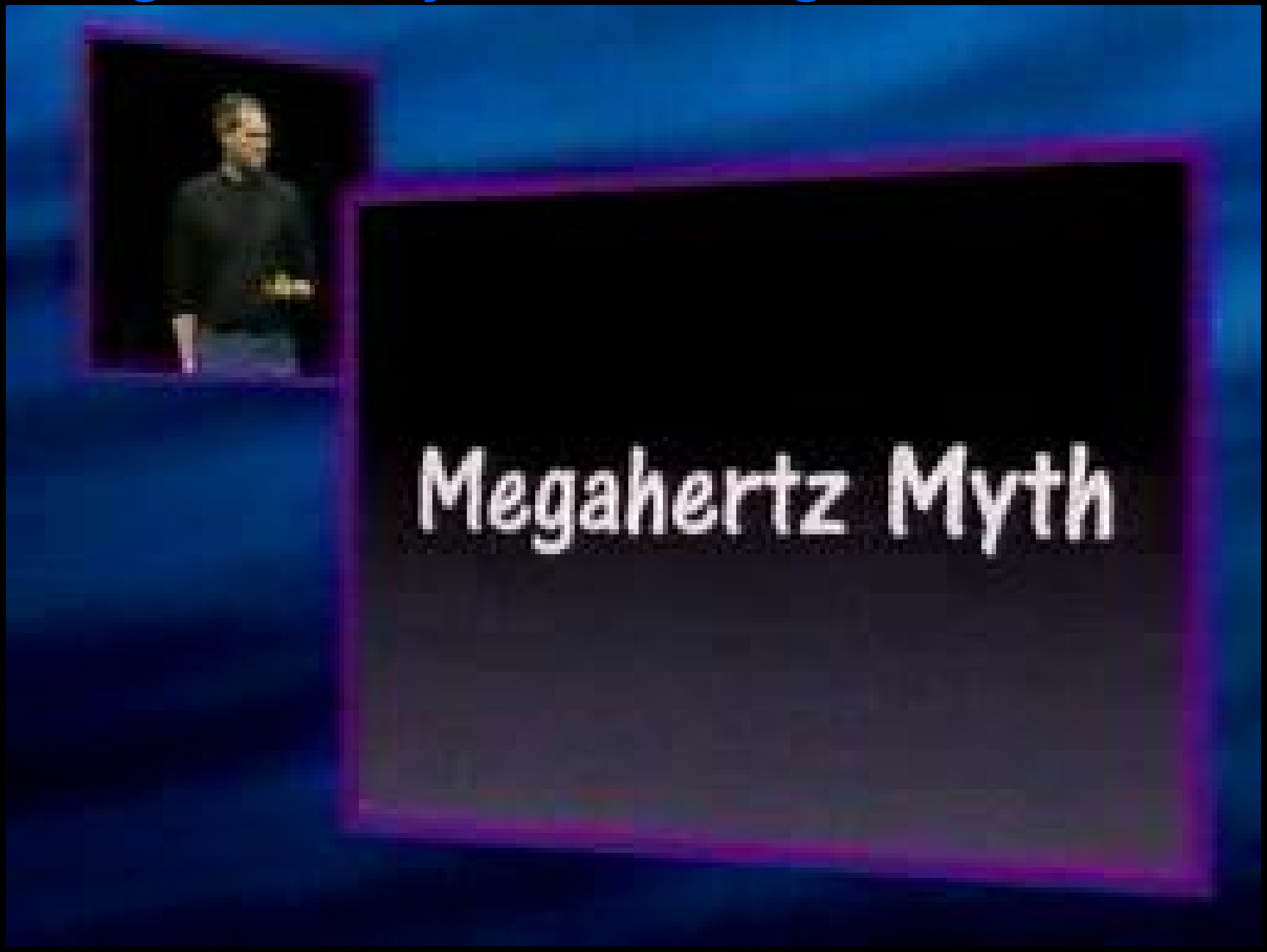
- Good products created when have:
  - Good benchmarks
  - Good ways to summarize performance
- Given sales is a function of performance relative to competition, should invest in improving product as reported by performance summary?
- If benchmarks/summary inadequate, then choose between improving product for real programs vs. improving product to get more sales; Sales almost always wins!

# Performance Evaluation: The Demo

If we're talking about performance, let's discuss the ways shady salespeople have fooled consumers (so that you don't get taken!)

5. Never let the user touch it
4. Only run the demo through a script
3. Run it on a stock machine in which "no expense was spared"
2. Preprocess all available data
1. Play a movie


# Megahertz Myth Marketing Movie

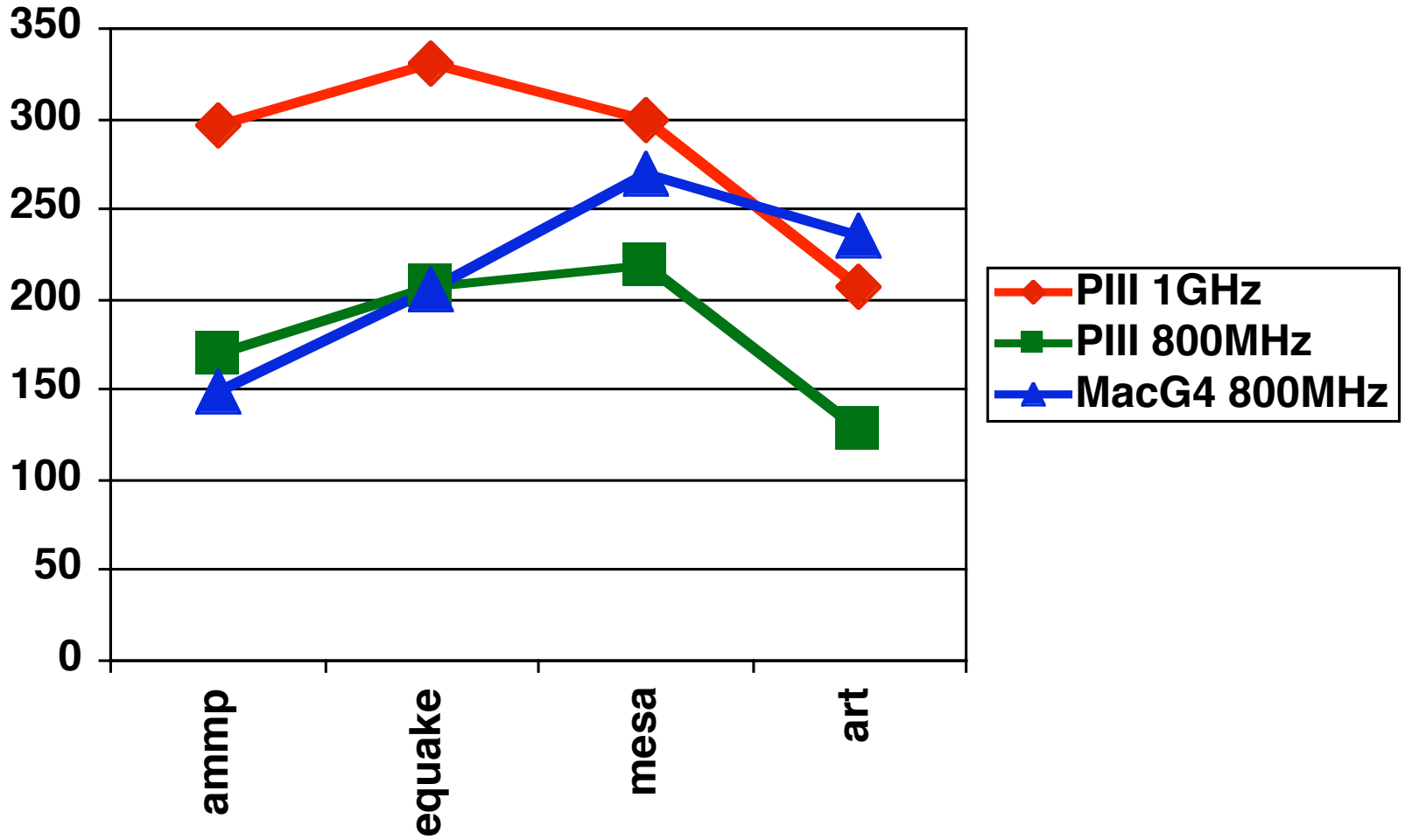


# PC / PC / Mac Showdown!!! (1/4)

- PC
  - 1 GHz Pentium III
  - 256 Mb RAM
  - 512KB L2 Cache
  - No L3
  - 133 MHz Bus
  - 20 GB Disk
  - 16MB VRAM
- PC 800MHz PIII
- Mac
  - 800 MHz PowerbookG4
  - 1 Gb RAM
    - 2 512Mb SODIMMs
  - 32KB L1Inst, L1Data
  - 256KB L2 Cache
  - 1Mb L3 Cache
  - 133 MHz Bus
  - 40 GB Disk
  - 32MB VRAM

**Let's take a look at SPEC2000 and a simulation of a real-world application.**

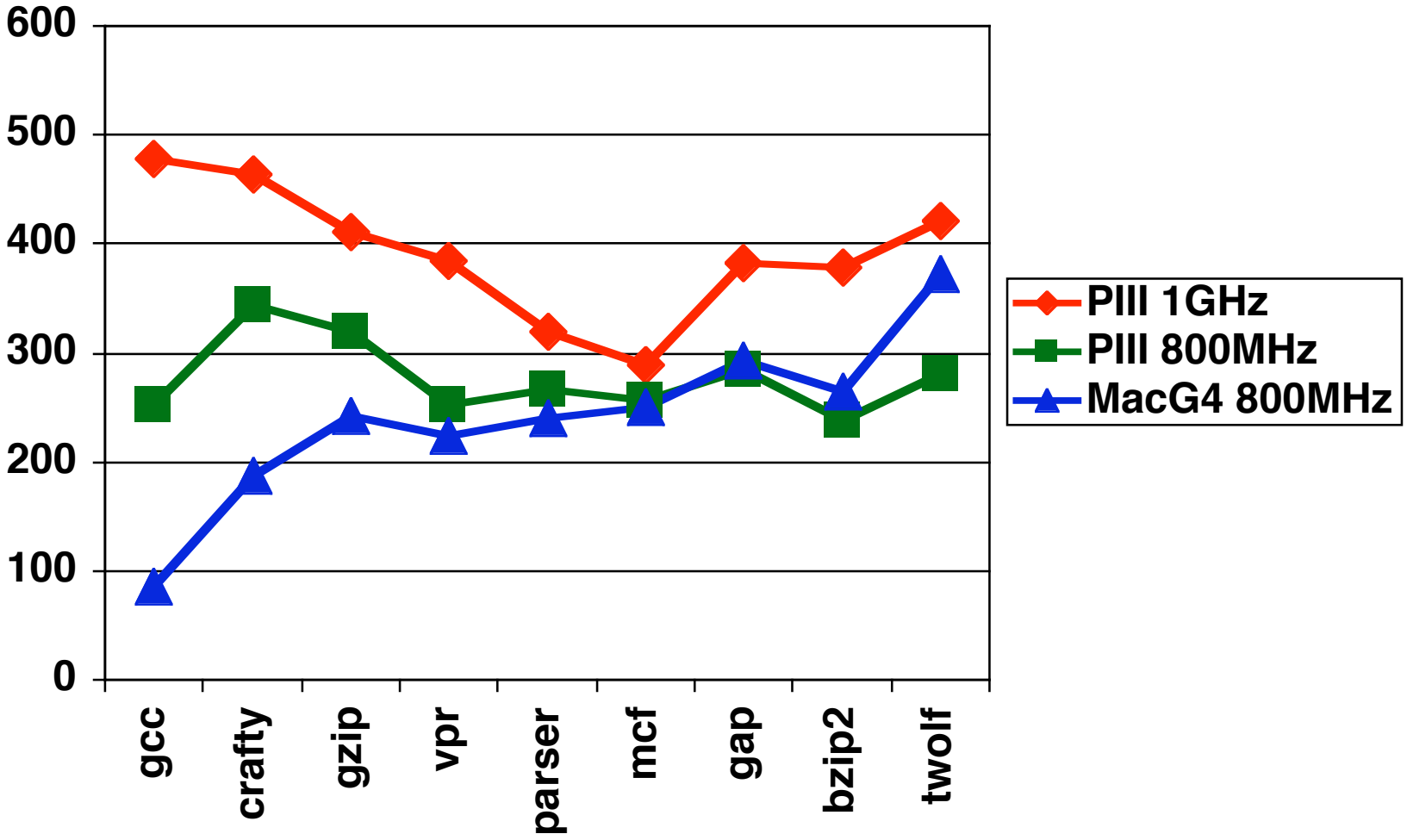
# PC / Mac Showdown!!! (2/4)




**CFP2000 (bigger better)**

**[left-to-right by G4/PIII 800MHz ratio]**

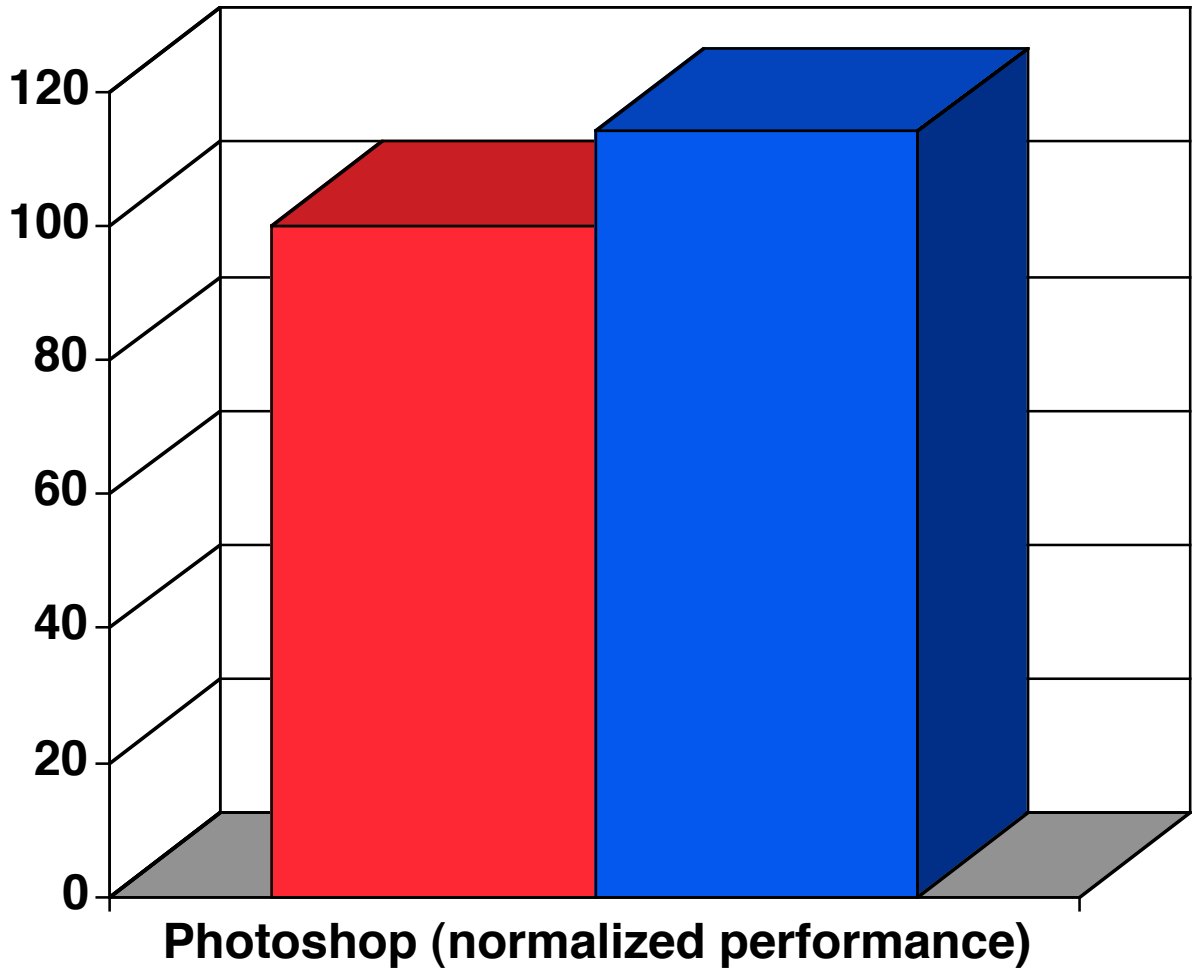
# PC / Mac Showdown!!! (3/4)




**CINT2000 (bigger better)**

**[left-to-right by G4/PIII 800MHz ratio]**

# PC / Mac Showdown!!! (4/4)




…Apple got in a heap of trouble when claiming the G5 was the "worlds fastest personal computer"

…lies, damn lies, and statistics.

**Normalized Photoshop radial blur (bigger better)**

**[Amt=10,Zoom,Best](PIII = 79sec = "100", G4= 69sec)**

# On the other end of the perf curve…

- It doesn't look like much. A drab, gray piece of plastic, about five inches long and three inches wide. A black-and-white screen, three inches by two inches, showing a few simple snippets of text. And yet this nondescript little computer may hold the key to bringing information technology to (mostly illiterate) Third World countries. Cost: US $250-$300

## Specs

- Intel Strong-ARM Chip
- 64MB RAM
- Modem
- Runs Linux!
- Stylus input (like Palm)
- Text-speech - speakers
- Room for Smart cards
- 3 AAA Batts last 4 Hrs

www.sciam.com/article.cfm?articleID=000454AE-7675-1D7E-90FB809EC5880000

# Administrivia

- Last semester's final + solutions online (thanks to Chema)
- Great talk today @ 4pm in 306 Soda: "Microprocessor Design Tradeoffs"
  - This talk reassesses advances in processor architecture in light of metrics that recognize power efficiency as the fundamental limiter to performance. We propose that in the light of these metrics many of the "advances" have been steps in the wrong direction, and we propose alternatives that can increase processor performance while simultaneously improving power efficiency

# Administrivia II

- HKN evaluations on Friday
- Final survey in lab this week
- Final exam review
  - Sunday, 2004-12-12 @ 2pm in 10 Evans
- Final exam
  - Tuesday, 2004-12-14 @ 12:30-3:30pm in 230 Hearst Gym
  - Same rules as Midterm, except you get 2 double-sided handwritten review sheets (1 from your midterm, 1 new one) + green sheet [Don't bring backpacks]

# Reality Check…The Simputer vs. Alternative?

**Simputer**

**(www.simputer.org)**

- **Cost $250**
- **3 AAA batteries/4 hrs**
- **Sell software for food**
- **Plays tetris**

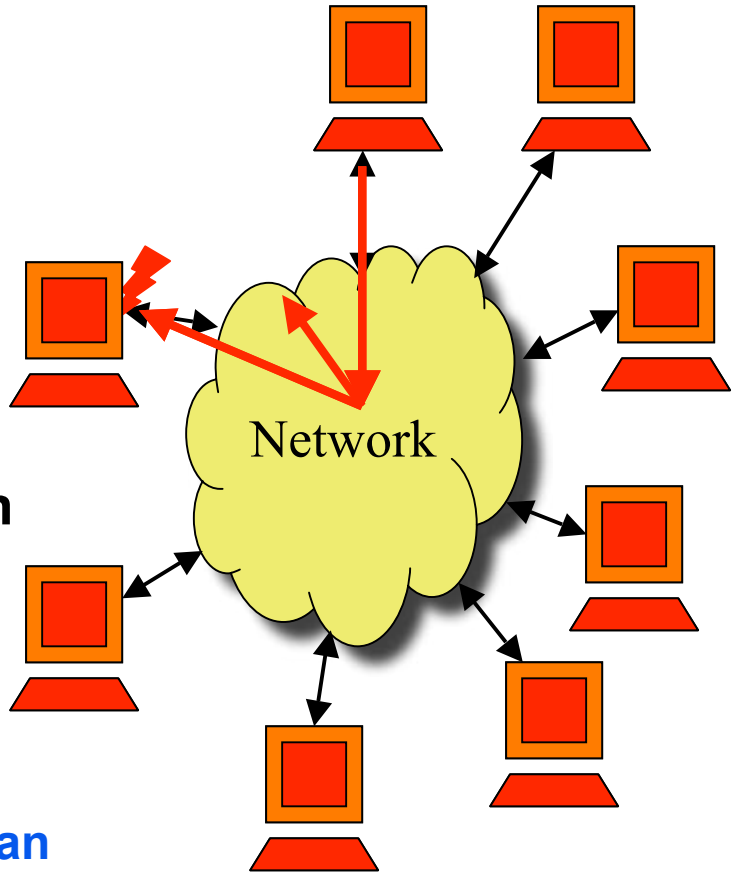
**Water buffalo**

**(www.heifer.org)**

- **Cost: $250**
- **2 Gallons of Milk/day**
- **Sell calves for food**
- **Plows fields**

# What Are Computer Worms?

- **Self replicating network programs**
  - Exploit vulnerabilities to infect remote machines
  - Victim machines continue to propagate the infection
- **Three main stages**
  1. Detect new targets
  2. Attempt to infect new targets
  3. Transfer the worm and activate the code on the victim machine
- **Often fully autonomous**
  - Spread without any user-interaction required
  - Can be very fast: Slammer infected all vulnerable hosts in 10 minutes




**Worm versus Virus:** Worm's self propagate through the network, no human interaction or exchange of files required.

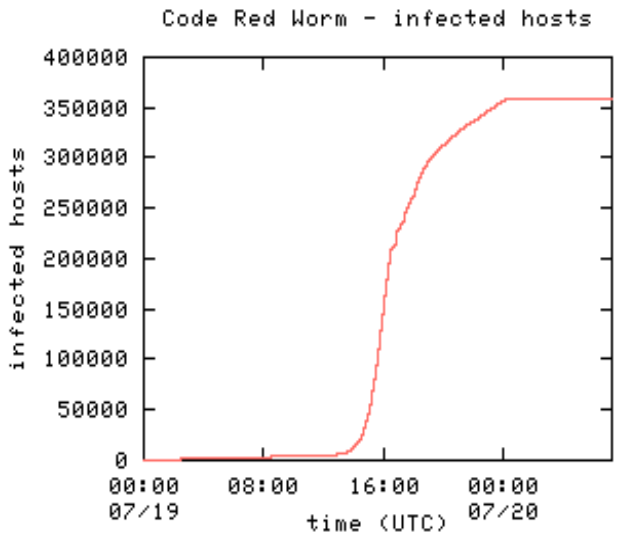
# How Do Worms Find Targets?

## Most common technique called: Random Scanning

- **Repeat Forever:**
  - **Pick a "random" IP address, if vulnerable, infect it**
    - **From single host, launch many threads to try more machine addresses simultaneously**
  - **Other techniques exist**
- **Fraction of the net infected (a):**
  - **Function of time and worm's speed**
  - **"Logistic" function**
    - **Initial growth is exponential**
- **Speed (K) depends on:**
  - **Rate of scanning**
  - **Number of vulnerable machines**
  - **Size of address space**

$$a = \frac{e^{K(t-T)}}{1 + e^{K(t-T)}}$$

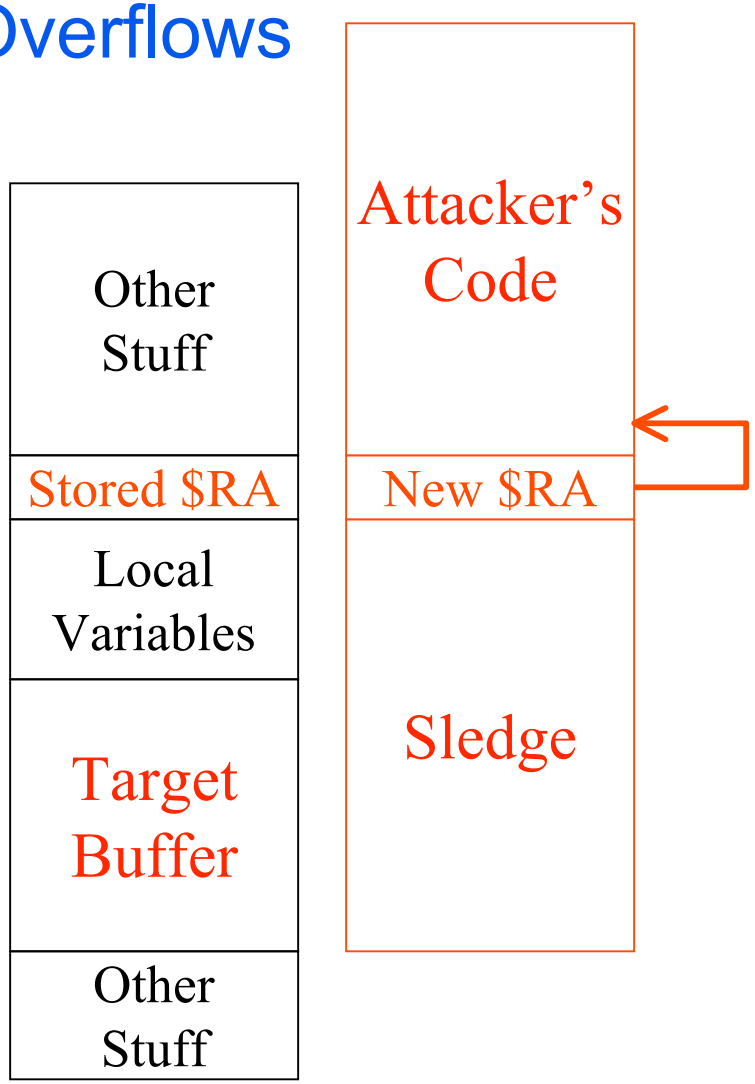
$$K = \frac{\text{Scan Rate * Vuln Machines}}{\text{Address Space Size}}$$



# How Do Worms Infect Targets?

## Most common technique: Buffer Overflows

- **The worm needs to somehow attack the victim machine**
  - **Take control of the victim**
  - **Transfer over the body of the worm**
- **Common Vulnerability: Stack Overflow**
  - **Victim program has an unchecked buffer on the stack**
  - **Attacking string overwrites the stack**
    - **Sledge → dummy data for overwriting**
    - **Overwritten return address → points to code**
    - **Injected code → Attacking code to execute**
  - **Now function returns to the attacker's code**
    - **The worm now uses this to transfer over the rest of the worm and to start running on the victim**



# Worm Conclusions

- **Example vulnerable applications (these have been, at least partially, patched):**
  - **Apache and IIS web servers. Code Red attacked IIS.**
  - **Blaster and its variants attacked Windows RPC (Remote Procedure Call) service, a "default-on" part of the OS.**
- **To date, most worms have been relatively benign. Most damage comes from flooding the network with scan messages and panic of system administrators. The day will come when a worm will carry a harmful payload:**

  **Delete files, Re-flash EPROM, (worse if host controls external devices!)**
- **What can you do?**

  *As a user:* **Patch your machine often. Do it today! (You're not just protecting yourself, but the entire network).**

  **(Many worm writers don't expose vulnerabilities themselves, but wait for MS to announce a patch, then hope that you will not get around to patching your machine.)**

  **Live behind a "firewall" – blocks traffic on most ports. Some people find this too limiting.**

  *As a programmer:* **learn to write secure software.**

# Peer Instruction

A. Performance is a stinking business; easily corruptible and you'll never hear honest reports from a company if they have a vested interest in the results.

B. Providing the right technology at the right price for people in developing countries is going to be one of the hardest tasks for HCI & Systems researchers in coming years.

C. Many in the know believe the threat from malicious Internet worms is about to explode exponentially.

| | ABC |
|---|---|
| 1: | FFF |
| 2: | FFT |
| 3: | FTF |
| 4: | FTT |
| 5: | TFF |
| 6: | TFT |
| 7: | TTF |
| 8: | TTT |

# Peer Instruction Answer

A. Where billions are on the line, few act honorably. Power and money corrupt, folks. TRUE

B. It's going to take lots of people understanding their problems and how technology could help, then figuring out how to deliver it at very low cost. TRUE

C. There is a feeling that cyberwars & cyberterrorism are inevitable… it's a scary world out there. TRUE

A. Performance is a stinking business; easily corruptible and you'll never hear honest reports from a company if they have a vested interest in the results.

B. Providing the right technology at the right price for people in developing countries is going to be one of the hardest tasks for HCI & Systems researchers in coming years.

C. Many in the know believe the threat from malicious Internet worms is about to explode exponentially.

| | ABC |
|---|---|
| 1: | FFF |
| 2: | FFT |
| 3: | FTF |
| 4: | FTT |
| 5: | TFF |
| 6: | TFT |
| 7: | TTF |
| 8: | TTT |

# "And in conclusion…"

- Benchmarks
  - Attempt to predict performance
  - Updated every few years
  - Measure everything from simulation of desktop graphics programs to battery life
- Megahertz Myth
  - MHz ≠ performance, it's just one factor
- It's non-trivial to try to help people in developing countries with technology
- Viruses have damaging potential the likes of which we can only imagine.