



Computer Systems and Networks

ECPE 170 – Jeff Shafer – University of the Pacific

Performance Measurement

Lab Schedule

Activities

- **Today / Thursday**
 - Background discussion
 - **Lab 5 – Performance Measurement**
- **Next Week**
 - **Lab 6 – Performance Optimization**

Assignments Due

- **Lab 5**
 - **Due by OCT 10th 5:00am**
- **Lab 6**
 - **Due by OCT 17th 5:00am**

Person of the Day: Bill Joy



- Co-founder of **Sun Microsystems**
- Author of **vi text editor**
- Key contributor to original BSD Unix while a Berkeley grad student
 - First open-source OS with TCP/IP

Performance Measurement



Lab 5 Goals

1. Measure program **execution time**
2. Break down program execution time by **specific subroutines / lines** of code
3. Monitor program for **memory leaks**
 - Not really “performance”, but uses same tool

Performance Measurement

- **Why is it important to measure application performance *in detail*?**

Valgrind



<http://valgrind.org/>

Valgrind Features

- **Memcheck** module – Memory error detector
- Access 1 beyond the beginning / end of your array?
- Access un-initialized memory?
- Reading/writing memory after calling free()?
- Memory leak? (Lost pointer to a malloc() block)
- **Valgrind produces a report that identifies these errors**

Valgrind Features

- **Callgrind** module – Program profiler
- Callgraph shows you what function called what other functions
- How much CPU time does each function / code line consume?
- **Valgrind produces a report that summarizes CPU usage of your program**

Valgrind Features

- **Massif** module – Heap profiler
 - Optimize your program to use less memory (by identifying where memory is being used)
- **Helgrind** module – Threading profiler
 - Bugs in multi-threaded programs are especially difficult to find!
- ... and more modules ...

Valgrind Common Uses

- **Your program runs and suddenly segfaults**
 - Recall a segfault means a memory address was accessed that doesn't exist for your program
- How do I find where this error is?
 - Valgrind can monitor your program and detect accesses outside of static variables and dynamic memory regions

Valgrind Common Uses

- **Your program gets slower and slower the longer it runs**
 - Memory leak? (Slowing running out of heap memory because you `malloc()` without ever calling `free()`)
- How do I find where this error is?
 - Valgrind can monitor your program. It can't tell you where you *should* free it, but it will tell you where you originally called `malloc()`, or where the pointer was lost

Valgrind Behind-the-Scenes

- **Just-in time compiler**
 - Your program is re-compiled onto a virtual (simulated) processor
 - Another example of a virtual machine!
- Benefit – Valgrind can observe your program running at the machine instruction level
- Drawback – Slow! (5x slower than normal)
 - *But it's still better than fixing bugs without a tool...*

Profiling Basics



Profiling Basics

- The next labs (5-7) ask you to **measure application performance** by conducting experiments
 - Execution time
 - Processor usage
 - Memory usage

- **Which of these system configuration do you think would be best in terms of producing the cleanest, most accurate, most reproducible results?**



ECPE 170

Terminal

jshafer@ecpe170: ~

```
jshafer@ecpe170:~$
```

Program to Benchmark

9:39 PM Jeff Shafer

The image shows a terminal window titled "Terminal" with a dark purple background. The prompt is "jshafer@ecpe170: ~" and the command "jshafer@ecpe170:~\$" has been entered. A large white text box is overlaid on the terminal, containing the text "Program to Benchmark". The window title bar shows "ECPE 170" and system status icons for mail, bluetooth, volume, and network. The system tray at the top right shows the time "9:39 PM" and the user name "Jeff Shafer".





Terminal

```
jshafer@ecpe170: ~
jshafer@ecpe170:~$
```

Program to Benchmark

Downloads

chrome://downloads

Search downloads

Downloads [Open downloads folder](#) [Clear all](#)

Today
Sep 18, 2012

ubuntu-12.04.1-desktop-i386.iso 476 KB/s - 36.2 MB of 695 MB, 23 mins left
<http://releases.ubuntu.com//precise/ubuntu-12.04.1-desktop-i386.iso>
[Pause](#) [Cancel](#)

Sep 13, 2012

05-c-programming.pdf Removed
http://ecs-network.serv.pacific.edu/ecpe-170/slides/05cprogramming.pdf/at_do...
[Remove from list](#)

BIOS_Acor_7-10_A_A.zip Removed

iTunes

Alone in the Cosmos
Glint — Sound in Silence
0:19 -4:22

Alone in the Cosmos
Glint
Sound in Silence

ESPN: The Worldwide Leader

espn.go.com

EDITIONS: USA DEPORTES More

CITIES: BOSTON CHICAGO DALLAS LOS ANGELES NEW YORK

ESPN

Search

MLB Full Scoreboard Soccer WNBA NFL NCAA Football Cricket myScores All Scores

Bottom 9th	Top 8th	Top 7th	F/12	Final	Final	F/10	Final	Final
BAL 2 SEA 2	COL 0 SF 4	TEX 3 LAA 10	MIN 6 CLE 5	OAK 2 DET 12	MIL 6 PIT 0	ATL 3 MIA 4	BOS 7 TB 5	CIN 3 CHC 1

myESPN NFL MLB NBA NHL NCAA FB NCAA BB NASCAR SOCCER MORE SPORTS

TOP STORIES WATCH LIVE

HEADLINES MY HEADLINES SHARES

- Escobar banned 3 games for eye-black slur
- Falcons' Turner charged with DUI, speeding
- Ex-ref blasts Goodell | Sub refs under fire
- Late Ronaldo goal jolts City in thriller | Blog

Profiling Basics

- **The best approach (directly booting Linux) may not be convenient to achieve for this class**
- **But you can **definitely** avoid the worst configuration!**
- **Keep your system simple when benchmarking**