Performance Lies My Professor Told Me:
The Case for Teaching Software Performance Engineering to Undergraduates

Robert F. Dugan Jr.
Department of Computer Science
Stonehill College
Easton, MA 02357 U.S.A
bdugan@stonehill.edu
Overview

- Motivation
- Survey
- SPE for Undergraduates
- Sample Project
- Future Work
Motivation

- Smith’s classic paper addresses the root cause of software performance problems.

- *Software Performance Engineering*: proactive approach to performance during software development cycle

- Industrial software development still has a costly “fix-it-later” approach to performance.

- One source may be the educational system producing the workforce that creates software applications.
Methodology

- Focus on U.S. postsecondary schools.
- Focus on undergraduate degree holders providing majority of workforce (20% total CS/IT degrees were graduate) – NCES
- Focus on top 24 research / doctorate schools (10% of total CS/IT Bachelor degrees) - CRA’s Taulbee Survey
- Focus on software engineering courses, not algorithm, system, and system modeling courses.
- Focus on course syllabus and lecture notes.
- 37 courses in sample.
Limitations

- On-line lecture notes only.
- Excluded: projects, exams, quizzes, reading material, textbooks, actual lecture.
- Document keyword search may have overlooked material.
- Subjective “feel” for weight of performance vs. other requirements, and presence in case studies.
- Not a random sample.
Analysis

- (64.87%) of software engineering courses spend little or no time on the subject of performance.
- (84.84%) had an inadequate or missing performance definition.
- (82.14%) provided either qualitative or no guidelines for performance requirements.
- (70.59%) treated performance as a subclass of non-functional requirements.
- (72%) advocated a reactive methodology either implicitly or explicitly.
- (74.07%) implied that either low-level system components or algorithmic efficiency had the most influence on application performance.
(16.22%) mentioned techniques for measuring application performance. No discussion of techniques for measuring throughput and resource utilization was evident.

(16.22%) provided guidance as to where to focus effort to achieve good application performance.

(27.02%) advocated some form of performance modeling. Half of these used modeling to diagnose problems in an implemented system.

Only two courses mentioned SPE. One presented SPE as a specialized discipline for database-centered / real-time applications rather than a broadly applicable discipline.
Conclusions

- Students are not learning the meaning of performance.
- Students are misled into believing that performance is a low-level system or algorithmic issue.
- Students are not learning how to measure software performance.
Students are learning to ignore performance early in the life cycle.

Optimization Quotes

Rules of Optimization:
Rule 1: Don’t do it.
Rule 2 (for experts only): Don’t do it yet.
- M.A. Jackson

"More computing sins are committed in the name of efficiency (without necessarily achieving it) than for any other single reason - including blind stupidity." - W.A. Wulf

"We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil." - Donald Knuth

Optimization 101

Reality
Hard to predict where the bottlenecks are
It's not so hard to use tools to measure what the code is doing once it is written.
Therefore, write the code you way you want to be correct and finished first, then worry about optimization.
"Premature Optimization" = evil
Classic advice from Don Knuth
Write the code to be straightforward and correct first
Maybe it's fast enough already
If not, measure to find the bottleneck
Students may be inferring that performance is one of a jungle of competing *qualitative*, non-functional requirements.
SPE for Undergraduates
Introduction

Is SPE being taught anywhere?

<table>
<thead>
<tr>
<th>School</th>
<th>Course Name</th>
<th>Course Number</th>
<th>Semester</th>
<th>Level</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carleton University</td>
<td>High-Performance Software</td>
<td>94511</td>
<td>W2003</td>
<td>G</td>
<td><a href="http://www.science.carleton.ca/courses/94511/">http://www.science.carleton.ca/courses/94511/</a></td>
</tr>
<tr>
<td>University of Colorado</td>
<td>Software Engineering of Multi-Program Systems</td>
<td>ECEN 5043</td>
<td>F2002</td>
<td>G</td>
<td><a href="http://ece-web.engr.colostate.edu/sweng5043">http://ece-web.engr.colostate.edu/sweng5043</a> info.html</td>
</tr>
<tr>
<td>University of Arkansas</td>
<td>Web Capacity Planning</td>
<td>S199</td>
<td>G</td>
<td></td>
<td><a href="http://www.cs.ua.edu/~rappo/">http://www.cs.ua.edu/~rappo/</a></td>
</tr>
<tr>
<td>Stonehill College</td>
<td>Software Performance Engineering</td>
<td>CS 399</td>
<td>S2003</td>
<td>Ug</td>
<td><a href="http://www.stonehill.edu/computerSPE/">http://www.stonehill.edu/computerSPE/</a></td>
</tr>
</tbody>
</table>

Can SPE be an undergraduate topic?

Course goal: teach students to design, build, and maintain software with good performance.
Syllabus

Textbook: *Performance Solutions* by Smith & Williams

- Introduction/Motivation (1 week)
- Performance Modeling (4 weeks)
- Measurement (1 week)
- Algorithms (2 weeks)
- Database Performance (1 week)
- Web Programming (2 weeks)
- Testing/Workload Generation (1 week)
- Research (1 week)
- Industry Visits/Talks (1 week)
- Principles/Patterns and Antipatterns (1 week)
Projects

*I hear and I forget, I see and I remember, I do and I understand* – A Chinese Proverb

- Stonehill College Website
- Online Auction System Modeling
- Instrumentation API
- Monkey Typing
- Student Registration Website
Sample Project

- College website performance problems
- New website under development, manager hopeful problems would disappear
- Class began speculating on cause of the problems (mirrors developer intuition)
- A better way: The Scientific Method
Observe and describe the performance problem. The description includes risk assessment.

- “The latest market research shows that 70% of all college-bound high-school seniors began their college search on the web, and those virtual tours became second only to actual campus visits in luring students.” – Wall Street Journal, Fall 2002
- Delay seemed to be on client side, not server side.
- Delay seemed to be network bandwidth related.
Scientific Method and SPE

Formulate a hypothesis to explain the performance problem. Usually some form of performance model that quantitatively predicts results of new observations.

- Performance problem is due to size of web page (178KB initial, 43KB refresh) and client bandwidth delays (56Kb modem: 31.73 seconds initial, 7.63 seconds refresh)
- Simple spreadsheet models of bandwidth/page size and experimental observation confirm.
- 75% of users have 56Kb or less bandwidth - Neilsen/Netratings December 2002

What is an acceptable response time?
Scientific Method and SPE

Web Page Responsiveness

- Broadband
- 56K Modem
- Both
What is the right balance between as Nick puts it in his paper, “presentation” and “performance”?

If you feel it could be in the scope of your class, I would love to have you/your students provide some solid recommendations on the balancing of presentation and performance - I know that your course focus is performance, but the real value of course lies in maximum performance with minimum impact on presentation.
Future Work

- SPE course refinements
- Improve dialog between SPE and SE:
  - software engineering courses
  - software engineering textbooks
  - senior capstone course
- Task-based guidelines for user perception of responsiveness