Lori Pollock  
Professor, CIS  
Program Analysis, Software Development & Maintenance Tools, Optimizing Compilers

'81  B.S. CS and Econ, Allegheny
'81-'86  PhD in CS, U of Pittsburgh  
Married Mark
'86-'90  Assistant Prof, Rice U  
Lauren '88; Lindsay '90
'91-  Assistant, Associate, Full Prof UD CIS  
Matt '95

Today: 3 teenagers - (19, 17, 13), 3 PhD students and a few undergraduate researchers
What I do here at UD

• Research
  - Software Engineering and Compilation Lab (Hiperspace)
    • 213 Smith Hall
  - Collaborations
    • Vijay Shanker (UD CIS), Lisa Marvel (Army Research Lab), Martin Swany (UD CIS), John Cavazos (UD CIS), Guang Gao (UD ECE)
  - Funding
    • Primarily NSF grants; some Army funding previously

• Graduate Teaching
  - CISC 672 Compilers
  - CISC 673 Program Analysis and Transformations
  - CISC 879 Software Testing and Maintenance
  - CISC 879 Software Tools and Environments
What I do outside UD

• Computing Research Association (CRA)’s Committee on the Status of Women in Computer Research (CRA-W)

• Mentoring – speaker at mentoring workshops for undergrads, grads, assistant and associate profs, and industry lab researchers

• Program committees, conference organization, NSF panels, paper reviews,… (typical of university researchers)
PhD Students in Training

Giri Sridhara
PhD

Emily Gibson Hill
PhD

Antony Danalis
PhD

And Undergraduates: Zak Fry (now at UVA), Haley Boyd, Eric Enslen
Recently Completed PhD
2007-08

David Shepherd
Postdoc, Startup

Sara Sprenkle
Assistant Prof
Washington & Lee U

Ben Breech
Postdoc, Nasa

Mike Jochen
Assistant Prof
East Stroudsburg U
Overview: Research Projects

- Natural Language Analysis of Programs
- Testing Web Applications
- Optimization of Cluster Parallel Programs
- Runtime Test Generation via Dynamic Compilers

Collaboration with Sara Sprenkle

Emily, Giri

Antony

Ben

Software Tools Testing Compilers Parallel Computing
Optimizing Cluster Parallel Programs

Research Problem - How can scientific codes be scaled to a cluster of many CPUs?

Major Challenge - Communication Costs

Approach -

An integrated system to hide communication latency

-Surveyor: Collect “knowledge” of cluster

-Compiler: analyze dependencies and transform to create maximal communication/computation overlap

-Communication Library: Use a companion library to MPI
ASPhALT: Automatic System for Parallel Application Transformations

Contribution: FIRST to cluster-optimize MPI codes
RUGRAT: RUntime GeneRAtion of Tests with Dynamic Compilers

Dynamic compilers: perform analysis and code modification at runtime

Research Question: How can dynamic compilation technology be exploited for use beyond program optimization?

Approach: Extend to analyze for:
- Software maintenance - impact analysis
- Testing of program security mechanisms and error handling
A Closer Look at a Dynamic Compiler

Dynamic Compiler

Basic block

Create basic block

translate

Modified Basic block

Execute on CPU

Analysis & optimization

code
RUGRAT Architecture

1. Test spec
2. Dynatest Generator
   - Analysis & optimization
   - Mod. Basic block
3. Dynamic Compiler
   - Basic block
   - Create basic block
4. Create basic block
5. Execute on CPU
6. Test Oracle
7. Test Report

Code flow:
- From code to create basic block, then to basic block, then to create basic block, then to execute on CPU, then to test oracle, then to test report.
Experiments Summary

• Tested variety of programs with RUGRAT
• 120+ error code handling call sites covered
  - Both application and system calls
• Increased error code coverage ~ 50% over regular test cases
  - Not all error code statements could be covered
    • Different options, etc
• Reasonable time overhead
RUGRAT Error Handling Code: Advantages and Disadvantages

Disadvantages:
- Not a perfect simulation

Advantages:
+ Adequate simulation
+ Can target system or application calls
+ Saves quite a lot of tester effort
Testing Web Applications

- Combination of
  - Stand-alone applications
  - GUIs and Database applications
  - Distributed applications

- Numerous technologies and components
Traditional Software Testing Process

Hard to obtain when testing web applications!!

Expected Results

Oracle

Actual Results

Test Cases

Replay Tool

Pass/ Fail

Test Case Generator

Test Cases

Application Implementation

Application Specification

Application Representation

User-session-based Testing
User-Session-based Testing Process

Beta Web Application (v.0.9) Deployment

Web Application Implementation (v.1.0)

Oracle

Log User Requests

Test Cases

Replay Tool

Create/Reduce Test Cases

User Sessions

User 1:
- register.jsp?name=ss&pass=tst
- login.jsp?name=ss&pass=tst
- logout.jsp

User-Sessions-based

Expected Results

Actual Results

Pass/Fail
Maintenance Testing for Web Applications

Research Problem: How can we exploit user session logging for testing of web applications after initial deployment, with minimal tester effort?

Contributions: Scalable, practical, automated structural testing framework for web applications

* Test case generation
* Test suite reduction
* Test oracles
* Test coverage criteria in terms of URLs, parameters, values
Analyzing the Names in Software

Research Problem
- 60-90% software costs are in reading and navigating large software systems to fix bugs and add new features. Can we help with automation of search, navigation, location of relevant code?
- Key: Programmers leave clues of their intent as they choose names.

Proposed Approach
- Develop, extend, and apply natural language-based analysis to the identifier names and comments

Contribution - Aid understanding, debugging, maintenance, development

Focus on actions
- Correspond to verbs
- Verbs need Direct Object
- Phrases more useful
Clients of NLPA

- FindConcept: Search Tool
- Timna: Aspect Miner
- Dora the Explorer: Program Explorer given a starting point

- NL technology used
  Synonyms, collocations, morphology, word frequencies, part-of-speech tagging, AOIG

- Evaluation indicates
  Natural language information shows real promise for taking software engineering tools to the next level of effectiveness

- Key to success
  Accurate extraction of NL clues
Overview: Research Projects

- Natural Language Analysis of Programs
- Testing Web Applications
- Optimization of Cluster Parallel Programs
- Runtime Test Generation via Dynamic Compilers

Collaboration with Sara Sprenkle

Emily, Giri

Antony

Ben

Software Tools Testing Compilers Parallel Computing