Joe T. Meehan

May, 2015

Hobbs 105
1501 Lakeside Dr.
Lynchburg, VA 24501
Voice: (434) 544-8395
E-mail: meehean.j@lynchburg.edu
Web: http://meehean-j.web.lynchburg.edu

Education

University of Wisconsin – Madison
PhD. in Computer Science, Aug. 2011
Advisers: Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau, and Miron Livny

University of Wisconsin – Madison
M.S. in Computer Science, Dec. 2005 **GPA:** 4.0

University of Wisconsin – Eau Claire
B.S. in Comprehensive Computer Science, May 2003 **GPA:** 3.67

Teaching Experience

Lynchburg College
9/11 – Present Assistant Professor of Computer Science

University of Wisconsin – Madison
Fall 2010 Guest Lecturer CS202: Introduction to Computation
Summer 2010 Lecturer CS367: Data Structures
Summer 2007 Teaching Assistant CS367: Data Structures

University of Wisconsin – Eau Claire
9/01 – 5/03 SACM Student Tutor

Employment

Lynchburg College
9/11 – Present Assistant Professor of Computer Science
- Taught wide variety of courses
- Conducted student-faculty research
- Coached ACM programming team

University of Wisconsin – Madison
9/07 – 8/11 Graduate Student Research Assistant, ADvanced Systems Laboratory (ADSL)
- Designed Harmony multiprocessor scheduling policy analysis tool. Analyzed multiprocessor scheduling policies of Linux schedulers.
- Developed CPU Futures scheduling feedback system. Creates a feedback channel between applications and CPU scheduler to avoid process starvation and allow applications to enforce scheduling goals.
- Analysis of memory thrashing in Linux mail servers.

9/04 – 8/11 Graduate Student Research Assistant, Condor Project
- Prototyped resource-awareness in Condor core components
- Developed unique process identifier frameworks
- Created distributed scheduler rapid deployment tool
- Implemented distributed scheduler migration feature

Great Lakes Higher Education (Madison, WI)

6/03 – 9/04 Part-time Software Engineer

- Member of shared software development committee.
- Developed J2EE database caching library, retrofitted software to use new framework
- Documented software architecture

Summer 2002 Software Engineer Internship

- Implemented J2EE student loan application network parser. Converts student loans from industry standard network protocol to internal business objects.

Student/Faculty Collaborative Research

High Throughput Computing using Scavenged CPU Cycles with Adam Noll

Each evening, when the last student leaves the lab, hundreds of computers become unused resources for several hours. The goal of our project is to harness these unused compute cycles to conduct scientific research. To accomplish this goal, we have installed the Condor High Throughput Computing (HTC) middleware on a test bed of 18 computers in Hobbs. Installing Condor on this test bed involved several system administration challenges. During this process we even discovered a bug in the Condor software, which the developers at the University of Wisconsin quickly fixed. To illustrate the effectiveness of using Condor to solve computationally expensive problems, we are working on a HTC solution to the game Dots and Boxes. Dot and Boxes is a two player pencil and paper game. The board is a grid of dots and players take turns connecting the dots to create boxes. The winner is the player completes the most boxes. A complete searchable solution to Dots and Boxes will allow a player to make the optimal move during each turn. Creating this solution would take many months if run on a single machine; using our Condor test bed, we expect to solve this problem in a few weeks.

Virtual Graffiti with Michael Burks

Augmented reality is a type of virtual reality that tethers the physical world to a computer-created world, enhancing the physical world with computer manipulations, i.e. adding useful graphics and interactive points. The goal of this project is to use augmented reality to create a virtual graffiti app that would allow a person "tag", place an image or text, in the augmented reality version of the world. Using a device, such as a phone or tablet, they will then be able to walk around the physical world and see their image or text "floating" in midair exactly where they placed it.

An Analysis of Rotten Tomatoes with Matthew Dwyer

Rotten Tomatoes is a well-known website that aggregates critic and user movie reviews. We have noticed that in some instances users and critics disagree about a movie's quality. Rotten Tomatoes provides an application programming interface (API) that allows computer programs to extract information from their large database. Using this API, we have collected a large body of data from their movie database. With this data we determined what percentage of movies feature a significant difference between critic and user reviews. We also hope to find some common characteristics between these movies. For examples, are critics harsher on comedies or Mathew McConaughey movies than audiences? This project includes several interesting technical challenges, most notably how to collect and manage such a large body of data. The API limits users to 10,000 queries a day. This is far too little for our purposes. Therefore to collect this data, we needed to create a local database and an automated system to add new movies to this database everyday using our allotted 10,000 queries. The large size of this local database makes performing data analysis on it difficult. To solve this problem, we devised a series of increasingly specific analyses to extract useful correlations.

Deconstructing the Android with Brandon Gannicott

This aim of this research is to deconstruct the high-level application features of the Android operating system into its standard operating system components. Specifically, we will examine I/O, process/thread architecture, and memory management.

Lynchburg College Distributed File System with Sarah Lavinder and Kevin Midkiff

Contemporary data analysis presents a new problem: many of today's data sets are so large that traditional data processing methods become too slow and inefficient. These data sets require a new approach to file storage and management. One solution is a distributed file system, which uses a computer network to allow multiple users to access and share files. Because hardware failure is the norm rather than the exception, these systems must provide efficient fault tolerance. Replication is necessary to prevent data loss and enhance accessibility. If all data is stored on one machine, transfer of data input and output weighs heavily on the system, reducing efficiency and increasing

processing time. Because the distributed file system allows multiple machines to access one large data set simultaneously, they can collaborate and achieve a solution more quickly. In this project, a distributed file system will be implemented using the machines in Hobbs 113 and 124 to enhance the Computer Science department's resources for solving complex problems.

Computational Simulation for Game-play Strategies: Creating the Ultimate Monopoly Strategy with Kristin Marstin

Computational simulation and data analysis provides useful feedback for a wide variety of both conceptual and real-life situations. A common objective in using these techniques is to examine and determine strategies for games. The board game Monopoly is an elaborate competition that combines a bit of luck with a mixture of mathematical, financial, and property management skills. Through a methodical computer simulation and an in-depth examination of the data produced, various strategies in the game of Monopoly have been made evident. These strategies are based on which properties and groups of properties are calculated to hold the best return on investment, the most efficient method of purchasing and placing houses and hotels per property, and the most effective way to increase income.

An Analysis of Resource Allocation in the Condor High Throughput Computing System with Matt Dwyer

The Condor High Throughput Computing system is designed to create a single super computer from a collection of individual machines. This presents a classic resource distribution problem: given multiple users, how should the machine resources be divided amongst them. Condor uses an outdated exponential decay model to determine resource allocation. This project attempts to highlight the short-comings of such an approach.

Automated Personal Assistant with Matt Dwyer

Having a phone on your person at all times requires some policy for when to answer it. In the past, those who could afford it had personal assistants who screened their phone calls. We propose to build an automated personal assistant. The automated assistant gently directs some callers to voicemail. Others can choose to leave a message or can override the screening and direct the assistant to put the phone call through with an urgent ring tone. This allows family members to differentiate between mundane phone calls and immediate emergencies.

Analysis of the Linux CFS CPU Scheduler with Tony McBride

The CPU scheduler for the Linux scheduler is over 8000 lines of code. In addition, several parts of the scheduler are inefficient. By examining the underlying mathematical models for the scheduler and using more appropriate data structures we believe we can reduce the amount of code and improve the efficiency of the scheduler. We also think that it will be possible to add additional features that will improve usability.

Professional Activities

12/10 – Present	Member ACM Special Interest Group on Computer Science Education (SIGCSE)
1/12 – Present	Avid reader of SIGCSE's Nifty Programming Assignments
9/11 – Present	LC Computer Science Curriculum Review
5/13	Attended Computing Frontiers Conference
Fall 2013	Participated in Lynchburg College Computer Science Program Review
2/12	Attended SIGCSE Annual Conference
2/12	Reviewed Randy Ribler's Vietnam Education Proposal
10/11	Attended Regional CS professors Lunch
9/10 – 5/11	Attended Delta Teaching Workshops and Seminars
9/08 – 8/11	Member SACM Graduates Anonymous, UW-Madison chapter
9/02 – 5/03	SACM President UW-Eau Claire chapter
1/02 – 9/02	SACM Vice-President UW-Eau Claire chapter

Service

9/11 – Present	LC Programming Team coach
9/11 – Present	Faculty Advisor, see below
1/12 – Present	Administrator of LC's High Throughput Computing Cluster
1/12 – Present	Manager of student system administrators
2/12 – Present	Interviewer LC Scholarship Competition
4/12 – Present	Moderator for Student Scholar Showcase
8/12 – Present	Information Technology and Resources Policies Committee

6/13 – 5/14	Freshman Advising
1/12 – 5/13	Faculty advisor for student system administrators
9/11 – 5/13	Lynchburg College Computer Science Program Review
Spring 2013	Student Scholar Showcase Committee
2/13	Interviewer LC Scholarship Competition
4/12	Senior Thesis Defense Committee, see below

Courses Taught

Fall ’14	Introduction to Computation (CS105)
Fall ’14	Data Structures and Abstraction I (CS241)
Fall ’14	Data Structures and Abstraction II (CS242)
Fall ’14	Senior Project (CS451)
Spring ’15	Sophomore Project (CS231)
Spring ’15	Data Structures and Abstraction II (CS242)
Spring ’15	Operating Systems (CS360)
Spring ’15	Senior Project (CS452)
Spring ’14	Sophomore Project (CS231)
Spring’ 14	Data Structures and Abstraction I (CS241)
Spring ’14	Special Topics: Distributed Systems (CS398)
Fall ’13	Introduction to Computation (CS105)
Fall ’13	Data Structures and Abstraction II (CS242)
Fall ’13	Database Management Systems (CS370)
Fall ’13	Senior Project (CS451)
Fall ’13	Senior Project (CS452)
Summer ’13	Internship (CS399)
Spring ’13	Intro. to Computer Science and Structured Programming I (CS141)
Spring ’13	Data Structures and Abstraction I (CS241)
Spring ’13	Independent Study (CS397)
Spring ’13	Senior Project (CS452)
Fall ’12	Data Structures and Abstraction II (CS242)
Fall ’12	Operating Systems (CS360)
Fall ’12	Independent Study (CS397)
Fall ’12	Internship (CS399)
Spring ’12	Intro. to Computer Science and Structured Programming II (CS142)
Spring ’12	Data Structures and Abstraction I (CS241)
Spring ’12	Senior Project (CS452)
Spring ’12	Programming Languages (CS322)
Spring ’12	Independent Study in Computer Science (CS397)
Spring ’12	Internship in Computer Science (CS399)
Fall ’11	Data Structures and Abstraction II (CS242)
Fall ’11	Database Management Systems (CS370)
Summer ’10	Introduction to Data Structures (CS367)

Awards

Spring ‘12	Putting Him/Her Through Award

Senior Thesis Defense Committee

Fall ‘12	Kara Winters, *Electronic Chessboard*
Spring ‘12	Keith Lester, *Fault-Tolerant Lab Control Software*
Spring ‘12	Brian Hudson, *A Ninety Dollar Interactive Whiteboard*
Spring ‘12	Owen Grubbs, *Committee Staffing Software*

Westover Thesis Defense Committee

2012-13	Kristin Marstin, *Computational Simulation for Game-play Strategies: Creating the Ultimate Monopoly Strategy*
2012-13	Brandon Gannicott, *Patent Warfare in the Smartphone Industry*

Research Summary

My research focuses on the scalability of CPU schedulers in server and cluster environments. The increase of multicore and SMP machines combined with the expanding set of CPU scheduling features means that CPU scheduling in commodity systems is becoming increasingly complex. Under heavy load these schedulers can suffer from pathological behavior, such as process starvation. The goal of my work is to reintroduce predictability and scalability into best-effort CPU schedulers, even under overload.

Harmony

Harmony is a technique for extracting the CPU load balancing policy from commodity operating systems. This technique combines high-level synthetic workloads with low-level instrumentation to fingerprint an operating system's multiprocessor scheduling policy. Harmony also aids in detecting performance bugs in the design and implementation of these policies.

CPU Futures

CPU Futures is a system designed to enable application control of scheduling for server workloads, even during system overload. CPU Futures contains two novel components: an in-kernel herald that anticipates application CPU performance degradation and a user-level feedback controller that responds to these predictions on behalf of the application. In combination, these two subsystems enable fine-grained application control of scheduling; with this control applications can define their own policies for avoiding or mitigating performance degradation under overload.

Refereed Publications

Uncovering CPU Load Balancing with Harmony, Computing Frontiers 2013. With Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau, and Miron Livny.

A Service Migration Case Study: Migrating the Condor Schedd, Midwest Instructional Computing Symposium 2005 (winner best student paper award). With Miron Livny.

Tech Reports

CPU Futures: Scheduler support for application management of CPU contention, University of Wisconsing Techreport #1684. With Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau, and Miron Livny. December 2010.

Logical Image Migration Based on Overlays, University of Wisconsing Techreport #1564. With Greg Quinn. June 2006.

Invited Talks

CPU Load Balancing in Multicore Systems, ADSL Team Meeting, April 12, 2010
Resolving Scheduling Conflicts with CPU Futures, ADSL Team Meeting February 3, 2010
Making Condor Environmentally Aware, Condor Week 2007
Problems in Dynamic Service Deployment, Condor Week 2006

Talks presented by students

An analysis of Rotten Tomatoes: The Movie Critic Website, VMI Undergraduate Research Symposium 2013

References

Dr. Andrea Arpaci-Dusseau
Professor of Computer Sciences
7375 Computer Sciences
University of Wisconsin-Madison
1210 West Dayton St.
Madison, WI 53706
(608) 265-6013
dusseau@cs.wisc.edu

Dr. Remzi Arpaci-Dusseau
Professor of Computer Sciences
7357 Computer Sciences
University of Wisconsin-Madison
1210 West Dayton St.
Madison, WI 53706
(608) 263-7764
remzi@cs.wisc.edu

Dr. Miron Livny
Professor of Computer Sciences
4367 Computer Sciences
University of Wisconsin-Madison
1210 West Dayton St.
Madison, WI 53706
(608) 262-0856
miron@cs.wisc.edu