Analysis and Prediction of the Dynamic Behavior of [Users,] Applications, Hosts, and Networks

Syllabus

Web Page

http://www.cs.northwestern.edu/~pdinda/predclass-s06

Instructor

Peter A. Dinda
Tech Institute L463
847-467-7859
pdinda@cs.northwestern.edu
Office hours: 2-3:30pm on Thursdays or by appointment
I am always interested in talking to you about your projects and will make time as needed to help you with them.

Location and Time

Lecture: Tech L168, T/Th 3:30-5 (will probably move to Systems Lab in Ford)

Prerequisites

Required	CS 311 or equivalent data structures course\
Required	Calculus and some linear algebra
Recommended	CS 213 or ECE 205 or similar
Recommended	Basic Statistics and Probability
Recommended	Some familiarity with Matlab, Maple, Mathematica, or S-Plus

If you're interested in this course but not sure whether you're prepared for it, please contact the instructor.

Objectives, framework, and caveats

This course should be of interest to advanced undergraduates and graduate students who are interested in computer systems research, understanding and controlling complex systems, and the interactions of different elements of computer systems, including their users.

The course is about understanding complex computer systems (including their interaction with their users) as if they were natural systems. What are the "natural laws" in this domain and how do they affect the design and implementation of systems? It focuses on how we can measure, analyze and predict the dynamic behavior of distributed computing environments, including their networks, hosts, applications, and their users. Broadly speaking, this is an area of **performance analysis** that includes **workload characterization and modeling** with the goal of providing **online** modeling and prediction for the direct benefit of applications and users. In this instance of the course we will pay particular attention to modeling and predicting human interaction with computer systems.

We will look at important, measurable properties of distributed computing environments and their users, such as network bandwidth and latency, host load and availability, and others, that vary dynamically over time. These properties drastically affect the performance of applications running on these systems. If you will, they are make up the **resource supply**.

We will also look at the measurable properties of executing applications. These dynamic properties form the **resource demand.** Matching supply and demand to optimize given performance metrics is the goal of scheduling and adaptation techniques in distributed systems. We will look at supply and demand in the context of adaptive systems, examining in particular how **predictions** of both can form the core of adaptation control algorithms, a holy grail of autonomic computing.

In this instance of the course we will pay particular attention to modeling and predicting **human interaction** with computer systems. It is the human users of a system that ultimately determine what the performance metrics are. If we can understand and predict their behavior in a quantitative way, we can go a long way to building effective adaptation systems. Furthermore, it gives us a new microscope fore understanding the humans themselves, and contemplating new kinds of systems and applications that can better support them.

For the most part, we will use an approach based on probability, statistics, and signal processing, although we will also touch on queuing theory and other approaches, such as chaotic dynamics, Markov models, and genetic programming. The course has three objectives. The first objective is for you to learn some of the theory behind measurement, analysis, and prediction. The second objective is for you to learn how this theory has been applied to computer systems in the past, and what fascinating new things were learned. The final objective is for you to become comfortable in applying the theory to his or her own data and systems, and in evaluating other methods for studying your data.

We will generally read about 2 papers or equivalent materials for each session, covering fundamental ideas and important recent results. Each paper will be formally presented to the group by a student and then discussed in a round-table manner. In parallel with the readings, students are strongly encouraged to apply what they are learning by using analytical tools such as Matlab, Maple, Mathematica, S-Plus, Prophet, and others to study real data, ideally data that they themselves, as well as the instructor, are interested in. Students will also be encouraged to use and extend on-line measurement and prediction systems such as RPS, Netlogger, Remos, and NWS.

Each student will complete a quarter-long research project. A separate handout, provided in paper only, gives a list of project ideas with strong potential. Other project ideas are possible, but require careful consultation with the instructor. The goal of the project work is to produce interesting new research results, that have solid potential for leading to publications for the student. Past iterations of this course have been very successful at producing publications and have even led to long-term research projects for the students, including two Ph.D. theses.

This is a graduate course and all students in it will be treated like graduate students. I will assume that you are interested in this material, that you can motivate yourself to learn about it, and that you will not be afraid to venture into uncharted territory (i.e., do research). The undergraduate section (395) will be graded separately, but differ primarily in that the expectations for the project will be slightly lower.

Reading

While there is no textbook for the course, **the following book is required** and is available at the campus store and via the web:

Raj Jain, *The Art of Computer Systems Performance Analysis*, Wiley, 1991.

This book provides background reference material for many of the tools we will use in this class, including measurement techniques and presentation, probability and statistics, experiment design, simulation, and basic queuing theory. While it’s not perfect, it is a very useful reference to have on your shelf even beyond this course. I will provide photocopied supplements for signal processing, time-series analysis, and other areas.

The vast majority of the reading for the course will consist of original research papers that report on studies of real computer systems, networks, and their applications. In addition to these papers, there are a number of books, papers, and other resources that are very helpful in understanding the theoretical and statistical analysis techniques that were applied to produce their results. We will read some of this material in tandem with the research papers so that you will be able to generalize the techniques and learn to apply them to your own work. Most of what we will read is available on the web, and I will hand out photocopies of what is not.

Project

Over the course of the quarter, you will apply what you learn to a project of your choice, and then document your project in a high quality paper and open presentation. A separate handout (paper only) provides a list of project ideas that have tremendous potential. Other topics are possible, but all project topics must be chosen in consultation with the instructor. Projects may be done individually or in groups. Project complexity and expectations will be tied to group size.

During your project, you will produce weekly progress reports emailed to the instructor. At the end of the quarter, you will hand in a conference-style paper documenting your progress, and you will give a conference-style talk at a public colloquium.

The expectation for graduate students is that the project will be quality work that the students would not be embarrassed to submit to a workshop. The three previous iterations of this course, totaling about 20 projects, have resulted in six publications, four long-term research projects (including two Ph.D. theses), and several conference-level drafts.

The expectation for undergraduates is that the project be something they would be proud to list on their resumes, although students are encouraged to aim high. Half of the undergraduates who have taken this course have gone on to Ph.D. programs in computer science.

Exams

There will be no exams. The final exam period may be used for presentations.

Grading

50 %	Project
10 %	Project paper and presentation
20 %	In-class paper presentations of papers
20 %	General classroom participation

Schedule

The readings and schedule may change during the course of the quarter, but these are the topics we will try to cover.

Tuesday, 3/28 Introduction

Performance Analysis and Workload Characterization
Perspective of Adaptive Systems
Statistical Tools
Queuing-centric ways of thinking about workload
Required Reading: Jain 1-4, 6, 10, 11

Thursday, 3/30 Introduction

More Statistical Tools and Queueing
Signal processing-centric ways of thinking about workload
Overview of importance concepts we will see again and again:
Power laws, heavy tails, self-similarity and long-range dependence, non-stationarity, predictability, and the implications these ideas have in provisioning, scheduling, etc.
Required Reading: Jain 12-14

Tuesday, 4/4 Processes on Hosts

Load sharing from the '80s
Exponential service time distributions lead to initial placement scheduling
Initial attempts to predict service times and other resource remands
Required Reading: Devarakonda89 (3), Eager88 (2)

Thursday, 4/6 Processes on Hosts

Load sharing from the '90s (and missed work from the '80s)
Power-law service time distributions mean process migration makes sense
Distributions of job interarrival times on supercomputers
Required Reading: Harchol-Balter96 (4), Kleban03 (5)
Optional Reading: Ott86 (1) [power law observed in '80s and ignored]

Tuesday, 4/11 More on a signal processing-centric approach to workloads

Using signals to capture aggregated events and their processing
Introduction to discrete-time signal processing
Required Reading: Oppenheimer handout, Box handout

Thursday, 4/13 Host Availability and Load

Distribution-based availability models from the Condor project
Signal-based load models
Required Reading: Mutka91 (6), Dinda99 (7),
Optional Reading: Li03 (12) [power], Dinda99-2 (9) [predictability]

Tuesday, 4/18 Heavy Tails and Self-similarity in Aggregated Network Traffic

Willinger's seminal work
How heavy-tailed arrival processes lead to self-similarity
How this impacts network design and router design
Required Reading: Willinger95 (27), Willinger95 (28), Bassingthwaighte (161)

Thursday, 4/20 More Recent Analysis of Aggregated Network Traffic

Is network traffic stationary? What does it consist of?
Required Reading: Kargagiannis04 (42), Xu05 (48)

Tuesday, 4/25 Approaches to Prediction

Signal prediction using linear and nonlinear models
Event prediction with Markov models and Hidden Markov Models
Evolving predictors using genetic programming
Required Reading: Box handout, others TBD

Thursday, 4/27 Predictability of Aggregated Network Traffic
Is network traffic predictable to any useful degree?
Required Reading: Sang00 (35), Qiao04 (44)
Optional Reading: Wolksi97 (50) [A widely used network prediction system]

Tuesday, 5/2 Power Laws of Internet Topology
How do network graphs grow? What are their characteristics?
Power laws appear in network topology graphs
Importance of remembering routing algorithms
Required Reading: Faloutsos99 (15), Li04 (25)

Thursday, 5/4 Web Workloads
Who talks to whom about what transferring how much?
Power law service time behavior on servers
Prediction of service time
Revitalization of the provably optimal SRPT scheduling discipline
Required Reading: Smith01 (86), Harchol-Balter03 (88)
Optional Reading: Lu04 (89) [SRPT with inaccurate predictions]
Lu05 (90) [Predictors for SRPT for other regimes]

Tuesday, 5/9 Documents and Queries on the Web and in P2P
Zipf's Rule and its implications
Query behavior in P2P
Required Reading: Breslau99 (84), Klemm04 (73)

Thursday, 5/11 Small Worlds and BitTorrent
How close are you to the data you want?
What is the nature of the most successful P2P protocol to date?
Required Reading: Iamnitchi04 (72), Pouwelse05 (77), Guo05 (78)

Tuesday, 5/16 Node lifetimes and other aspects of P2P systems
How long will the node you attach work?
Required Reading: Bhagwan03 (69), Leonard05 (74), Qiao06 (79)
Optional Reading: Qiao04 (71) [SRPT for P2P]

Thursday, 5/18 File systems
Contents of large scale file systems
Markov modeling accesses to predict the next file
Required Reading: Douver99 (110), Kroeger96 (108)

Tuesday, 5/23 Web and Chat Users

Required Reading: Davison04 (141), Dewes03 (137), Zhu03 (139)

Thursday, 5/25 Wireless Networks and Their Users

Required Reading: Balachandran01 (135), Schwab04 (41)

Optional Reading: Eckhardt96 (30)

Tuesday, 5/30 Direct User Feedback

Having the user tell you when it hurts and how to fix it

Measuring and understanding user comfort with resource borrowing

Scheduling under user control

Power management under user control

Required Reading: Sousa05 (144), Gupta04 (140), (146)

Optional Reading: (147) [power management]

Thursday, 6/1 Game Users [cut if no slack, maybe swap with HPC]

Required Reading: Hendersen01 (134), Chambers05 (107)

Optional Reading: Abdelkhalek01 (105)