ECE 750 (Topic XX): Mining and Analysis of Software Engineering Data

Fall (202X) Weiyi Shang

Background and Motivation

Large software systems (e.g., Amazon.com and Google's GMail) pose new challenges for software engineers and operators. These systems require near-perfect up-time while supporting millions of concurrent connections and operations. Failures and errors in such systems may bring financial and reputational repercussions. During the life cycle of such software systems, developers are focused on developing feature rich and bug-free software, while operators are focused on ensuring a failure-free and scalable operation of the software.

This course explores leading research in the mining of large software systems, discusses challenges associated with bridging the development and operation activities of large systems, highlights industrial engineering practice, and outlines future research directions. In particular, the course leverages the mining of data that is generated during the development and operation of large software systems. Students will acquire the advanced knowledge about the development and operations in the field. Once completed, students should be able to conduct research in topics related to mining software repositories and will be able to leverage the learnt techniques in other system and software engineering related research or practice.

Prerequisite

Some undergraduate students may have gain similar knowledge from work experiences or have taken similar introductory courses as part of their undergraduate degrees. Therefore, students with such knowledge may gain special permissions from the instructor. In particular, undergraduate students should gain the permission by sharing a resume and description why they are qualified for this course with the instructor.

Calendar Description

ECE 750 (Topic XX): Mining and Analysis of Software Engineering Data
Prerequisite: Undergraduate students need permission of the instructor by showing their knowledge of software engineering.

The topics of this course include challenges in developing and operating large software systems; software analysis: static code analysis and dynamic analysis; software log analysis; software performance analysis: monitoring, measuring, modeling and diagnosing system performance; mining software repositories; empirical studies on large-scale software data; and System configuration optimization.

Tentative topics and schedule

This course tentatively covers the following topics:

- Challenges in developing and operating large software systems
- Software data extraction and analysis
- Static code analysis and dynamic analysis

- Software log analysis
- Software performance analysis
- Mining software repositories
- Empirical studies with large-scale software data
- System configuration optimization

The table below outlines a possible schedule for this course over a 12-week term. During every week of the course, students will learn how to adopt existing data mining techniques in order to leverage large-scale software data during software development and operation. Slight modifications to content may occur based on the background and abilities of students.

Week(s)	**Topics**
1	Challenges of large software systems
2	Empirical studies on large-scale software data
3	Software static code analysis
4	Software dynamic analysis
5	Hands-on tutorial on software data analytics
6	Mining software repositories
7	System performance analysis
8	Software log analysis
9	Code analysis for performance improvement
10	Configuration and the end
11	In-class exam
12	Final project presentations

Grading Scheme

Grades will be based on weekly critique of assigned papers, paper presentation and discussion, assignments, in-class exam(s) and a group project involving analysis of software engineering and system data. As part of the course project, students will be expected to produce a final report detailing their techniques and findings.

Textbooks and References

The course references would include (subject to be updated year by year):

1. David Maplesden, Ewan Tempero, John Hosking and John C. Grundy, "Subsuming Methods: Finding New Optimisation Opportunities in Object-Oriented Software", ICPE 2015: Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering

2. Wei Xu, Ling Huang, Armando Fox, David Patterson and Michael Jordan, "Detecting Large-Scale System Problems by Mining Console Logs", SOSP 2009 Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles

3. Todd Mytkowicz,Amer Diwan, Matthias Hauswirth, Peter F. Sweeney, "Producing Wrong Data Without Doing Anything Obviously Wrong!", ASPLOS 2009: Proceedings of the 14th international conference on Architectural support for programming languages and operating systems

4. Sapan Bhatia, Abhishek Kumar, Marc E. Fiuczynski and Larry Peterson, "Lightweight, High-Resolution Monitoring for Troubleshooting Production Systems", OSDI 2008: Proceedings of the 8th USENIX conference on Operating systems design and implementation

Items	**% of final grade**	**Notes**
Weekly paper critique	10%	Each student needs to submit a 1-page critique report for a pre-selected research paper per week.
Paper presentation and discussion	10%	Every week, the students who present need to submit slides of presenting the pre-selected research papers. The rest of the class needs to discuss with the the presenters. The activities during discussion are considered in the grading. Each student approximately presents twice during the course.
Assignment	15%	Students are required to finish an assignment based on papers that cover the relevant knowledge taught in the class.
In-class exam	20%	Students are required to take in-class exam(s) that cover the relevant knowledge taught in the class.
Course project	45%	Each group of students needs to conduct an original project reported within 10 pages in IEEE double column format. The project will explore one or more of the themes covered in the course. Each group also needs to submit a project proposal (2 pages IEEE format). The proposal should provide a brief motivation of the project, a detailed discussion of the data and systems that will be used in the project, along with a timeline of milestones, and expected outcome.
Total	100%	

5. Sai Zhang and Michael D. Ernst, “Automated Diagnosis of Software Configuration Errors”, ICSE 2013: Proceedings of the 2013 International Conference on Software Engineering

6. Kirk Glerum, Kinshuman Kinshumann, Steve Greenberg, Gabriel Aul, Vince Orgovan, Greg Nichols, David Grant, Gretchen Loihle, and Galen Hunt, “Debugging in the (Very) Large: Ten Years of Implementation and Experience”, SOSP 2009: Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles

7. Shi Han, Yingnong Dang, Song Ge, Dongmei Zhang, and Tao Xie, “Performance Debugging in the Large via Mining Millions of Stack Traces”, ICSE 2012: Proceedings of the 34th International Conference on Software Engineering

8. Ding Yuan, Yu Luo, Xin Zhuang, Guilherme Renna Rodrigues, Xu Zhao, Yongle Zhang, Pranay U. Jain, and Michael Stumm, “Simple Testing Can Prevent Most Critical Failures: An Analysis of Production Failures in Distributed Data-Intensive Systems”, OSDI 2014: Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation

9. Ding Yuan, Soyeon Park, and Yuanyuan Zhou, “Characterizing Logging Practices in Open-Source Software”, ICSE ’12: Proceedings of the 34th International Conference on Software Engineering

10. Zuoning Yin, Xiao Ma, Jing Zheng, Yuanyuan Zhou, Lakshmi N. Bairavasundaram, Shankar Pasupathy, “An Empirical Study on Configuration Errors in Commercial and Open Source Systems”, SOSP 2011: Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles

11. Guilherme Bicalho de Padua and Weiyi Shang. Revisiting Exception Handling Practices with Exception Flow Analysis. 17th IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM 2017).

12. Guilherme Bicalho de Padua and Weiyi Shang. Studying the Prevalence of Exception Handling Anti-Patterns. 2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC 2017)

13. Zhen Ming Jiang, Ahmed E. Hassan, Parminder Flora, and Gilbert Hamann. An Automated Approach for Abstracting Execution Logs to Execution Events. Journal of Software Maintenance and Evolution: Research and Practice. August, 2008.

14. Thanh H. D. Nguyen, Bram Adams, Zhen Ming Jiang, Ahmed E. Hassan, Mohamed Nasser, Parminder Flora. Automated Detection of Performance Regressions Using Statistical Process Control Techniques. In Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering.

15. Tse-Hsun Chen, Weiyi Shang, Zhen Ming Jiang, and Ahmed E. Hassan. Detecting Performance Anti-patterns for Applications Developed Using Object-Relational Mapping. The 36th International Conference on Software Engineering (ICSE 2014)

16. Israel Herraiz Ahmed E. Hassan. Beyond Lines of Code: Do We Need More Complexity Metrics? Making Software.

17. D. Landman, A. Serebrenik and J. Vinju, "Empirical Analysis of the Relationship between CC and SLOC in a Large Corpus of Java Methods," 2014 IEEE International Conference on Software Maintenance and Evolution.

18. Guilherme B. de Pádua and Weiyi Shang. Studying the relationship between exception handling practices and post-release defects. The 15th International Conference on Mining Software Repositories (MSR 2018)

19. Weiyi Shang, Ahmed E. Hassan, Mohamed Nasser and Parminder Flora. Automated Detection of Performance Regressions Using Regression Models on Clustered Performance Counters. The 6th ACM/SPEC International Conference on Performance Engineering (ICPE 2015).

20. Shane McIntosh, Yasutaka Kamei, Bram Adams, and Ahmed E. Hassan. 2014. The impact of code review coverage and code review participation on software quality: a case study of the qt, VTK, and ITK projects. In Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014).

21. M. D. Syer, Z. M. Jiang, M. Nagappan, A. E. Hassan, M. Nasser and P. Flora, "Leveraging Performance Counters and Execution Logs to Diagnose Memory-Related Performance Issues," 2013 IEEE International Conference on Software Maintenance.

22. Qingwei Lin, Hongyu Zhang, Jian-Guang Lou, Yu Zhang, and Xuewei Chen. 2016. Log clustering based problem identification for online service systems. In Proceedings of the 38th International Conference on Software Engineering Companion (ICSE '16).

23. Jieming Zhu, Pinjia He, Qiang Fu, Hongyu Zhang, Michael R. Lyu, and Dongmei Zhang. 2015. Learning to log: helping developers make informed logging decisions. In Proceedings of the 37th International Conference on Software Engineering (ICSE '15).