Welcome to CPRE 563 Advanced Data Storage Systems

*Theme: How to build **reliable & secure** systems for data*

Course Information

- Instructor: Mai Zheng <mai@iastate.edu>
- Time: MWF, 11:00 AM - 11:50 AM, Spring 2024
- Location:
  - In-Person @HOWE 1242 (Section 1)
  - Webex Live Class (Section 2): https://iastate.webex.com/meet/celt1242howeLinks to an external site.
  - Recordings: Canvas/Echo360
- Office Hours:
  - Time: 4:00-5:00 pm, Mondays
  - Location: 349 Durham Hall
- TA: Tabassum Mahmud <tmahmud@iastate.edu>
  - TA Office Hours:
    - Time: 2:00 - 3:00 pm, Wednesdays
    - Location: Online, Webex link: https://iastate.webex.com/meet/tmahmudLinks to an external site.

Course Description

Data storage systems is a fundamental building block of modern society. The trend of big data brings exciting new challenges for data storage in terms of reliability, security, performance, etc. How to address the issues coming with the unprecedented scale and complexity is an important open question to our society. Students in computer engineering, computer science, electrical engineering, and/or other computer-related fields need to be aware of the challenges as well as the opportunities. In fact, many top institutions have identified the strong needs in data storage expertise and have dedicated courses for storage systems (e.g., "ECE 746 Storage Systems" at Carnegie Mellon University). Unfortunately, there is no equivalent course at Iowa State. CPRE 563 fills the gap.

This course focuses on how to build reliable and secure storage systems for valuable digital data (e.g., financial transactions, scientific computations, family photos). The content covers the fundamentals of data storage technologies as well as the state of the art, including storage devices, file systems, and warehouse-scale big data storage, with an emphasis on design tradeoffs for reliability and security. Specifically, the main topics include but not limited to the following:

(1) Hard disk drives

(2) Flash-based solid state drives

(3) Byte-addressable persistent memories

(4) Multi-device arrays

(5) File systems and checkers

(7) Large-scale HPC & Cloud storage

(8) Practical methodologies for analyzing and improving system **reliability, security, and performance** (e.g., Fault Injection, Program Analysis, Fuzzing, Model Checking, Symbolic Execution)

Prerequisites

No strict prerequisites, but knowledge of operating systems (e.g., CPRE 308) and C programming language is helpful; if you are not sure, please talk to the instructor.

Course Outcomes/Objectives

The course will provide the knowledge and first-hand experience on storage systems to students. Specifically, after completing this course, students will:

(1) learn the overall architecture of storage systems

(2) learn the internals of representative storage hardware;

(3) learn the internals of representative storage software;

(4) understand the design tradeoffs of storage systems for high performance, reliability, and security;

(5) gain hands-on experience on high-impact open-source systems;

(6) gain critical thinking and problem solving skills and apply them to analyze/improve real-world systems.

Course Website

On ISU Canvas.

Textbook

No textbook required. The content will be based on lecture slides as well as selected publications from renown venues.

Assignments & Grading

Students will be evaluated based on the following tasks (More details will be covered in class and on slides):

(1) Paper Review & Paper Presentation (30%):

The instructor will select a set of representative research papers. Students will be required to read the papers thoroughly and write reviews for the papers. A review template will be provided. Reviews will be graded based on the depth of understanding as well as the unique insights.

Depending on the class size, some students will have opportunities to lead the discussion of one or more paper in class (first come first serve). For each presentation, the student lead has about 30 minutes to present the problems, background, ideas, design & implementation issues, experiments, tools, results, and related work. Additionally, we will have in-depth discussion about the paper, which can be during or after the presentation. The instructor will help coordinate. Those presenters do not need to write paper reviews.

(2) Project (60%):

Students will work toward a research-oriented project either by group or individually. The goal is to do quality systems research, i.e., to add to our understanding of how to analyze and build storage systems. Suggested project ideas will be provided by the instructor, but students are strongly encouraged to come up with their own project ideas under the guidance of the instructor. Three reports are expected from each project: initial proposal, midterm report, and final report.

(3) Class Participation (10%):

Students are expected to participate in in-person/Webex/Canvas discussion of research papers and related topics.

Please refer to the course slides for more details.

Attendance

Attendance is mandatory:

-
    - Three absences are allowed. You don't need to email the instructor for approval if you haven't used up the quota.
    - Four absences will lower your grade 1 full letter, and each further absence results in your grade dropping 1 additional full letter. Seven absences will result in failure of the course if you do not withdraw.
    - Three lates and/or early leaves equal one absence.
    - After exhausting the default three-absence quota, if you need additional absences, please contact the instructor with justification & solid proof in advance. We will evaluate the request case by case.
    - Attendance is taken in class randomly throughout the semester.

Late Assignment Policy

Assignments (Paper Reviews & Project Reports) should be submitted to Canvas by the deadlines. Late submission will NOT be accepted, unless approved by the instructor in advance.

Topics & Schedule (Tentative)

Week & Date	**Tentative Topics**	**Supplementary Readings**
Week 1 Jan 17, 19	Introduction & Warmup	*How to Read a Paper (by S. Keshav) *Writing reviews for systems conferences (by T. Roscoe) *Building Secure and Reliable Systems (by Google): Chapter 1 & 2; Chapter 6, 8, 9, 13 & 15
Week 2 Jan 22, 24, 26	Storage Hardware: HDD, RAID, SSD, SmartSSD, PM;	*RAID: A Personal Recollection of How Storage Became a System (Randy H. Katz) *Design Tradeoffs for SSD Performance (USENIX ATC'08) *An Empirical Guide to the Behavior and Use of Scalable Persistent Memory (FAST '20)
Week 3 Jan 29, 31, Feb 02	Storage Software: Single-Node Storage Software	*The Design and Implementation of a Log-Structured File System (TOCS'92) *F2FS: A New File System for Flash Storage (FAST'15) *NOVA: A Log-structured File System for Hybrid Volatile/Non-volatile Main Memories (FAST'16) *An Analysis of Persistent Memory Use with WHISPER (ASPLOS'17)
Week 4 Feb 05, 07, 09	Storage Software: Distributed Storage Software	*The Google File System (SOSP'03) *The Hadoop Distributed File System (MSST'10) *HopsFS: Scaling Hierarchical File System Metadata Using NewSQL Databases (FAST'17) *λFS: A Scalable and Elastic Distributed File System Metadata Service using Serverless Functions (ASPLOS'24)
Week 5 Feb 12, 14, 16	Putting It Together: Enhancing the Reliability & Security of Storage Systems	*Why Does the Cloud Stop Computing? (SoCC'16), *Fail-Slow at Scale: Evidence of Hardware Performance Faults in Large Production Systems (FAST'18) *Towards Robust File System Checkers (TOS'18)

		*A Study of Failure Recovery and Logging of High-Performance Parallel File Systems (TOS'22) *The Art, Science, and Engineering of Fuzzing: A Survey (TSE'19) *KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs [OSDI'08] *S2E: A Platform for In-Vivo Multi-Path Analysis of Software Systems (ASPLOS'11) *DR. CHECKER: A Soundy Analysis for Linux Kernel Drivers (Security'17)
Week 6 Feb 19, 21, 23	**Project Proposal Report**	--
Week 7 Feb 26, 28, Mar 01	Project & Reading Week: [No Class due to FAST'24]	--
Week 8 Mar 04, 06, 08	**Storage Configuration** **[P01]** Do Not Blame Users for Misconfigurations (SOSP'03) **[P02]** ConfD: Analyzing Configuration Dependencies of File Systems for Fun and Profit (FAST'23)	*A Study of Linux File System Evolution (FAST'13) *Understanding and Discovering Software Configuration Dependencies in Cloud and Datacenter Systems (FSE'20) *A Study of Persistent Memory Bugs in the Linux kernel (SYSTOR'21) *Learning from Mistakes — A Comprehensive Study on Real World Concurrency Bug Characteristics (ASPLOS'08) *Fuzzing File Systems via Two-Dimensional Input Space Exploration (S&P'19) *Finding Semantic Bugs in File Systems with an Extensible Fuzzing Framework (SOSP'19)
Week 9 Mar 11, 13, 15	[SPRING BREAK]	--

Week 10 Mar 18, 20, 22	**Storage Fault Tolerance (Single-node)** **[P03]** IRON File Systems (SOSP'05) **[P04]** Torturing Databases for Fun and Profit (OSDI'14)	*Vinter: Automatic Non-Volatile Memory Crash Consistency Testing for Full Systems (ATC'22) *A Study of Database Performance Sensitivity to Experiment Settings (VLDB'22) *Towards Robust File System Checkers (FAST'18) *Fuzzing Error Handling Code using Context-Sensitive Software Fault Injection (USENIX Security'20) *Evaluating File System Reliability on Solid State Drives (ATC'19) *Specifying and Checking File System Crash-Consistency Models (ASPLOS'16)
Week 11 Mar 25, 27, 29	**Storage Fault Tolerance (Large-scale)** **[P05]** Redundancy Does Not Imply Fault Tolerance: Analysis of Distributed Storage Reactions to Single Errors and Corruptions (FAST'17) **[P06]** On Fault Tolerance, Locality, and Optimality in Locally Repairable Codes (ATC'18)	*Lessons & Actions: 10K SSD-Related Failures in Alibaba Cloud (USENIX ATC'19) *How Bad Can a Bug Get? An Empirical Analysis of Software Failures in the OpenStack Cloud Computing Platform (FSE'19) *Understanding Issue Correlations: A Case Study of the Hadoop System (SoCC'15) *Simple Testing Can Prevent Most Critical Failures: An Analysis of Production Failures in Distributed Data-Intensive Systems (OSDI'14) *Opening the chrysalis: on the real repair performance of MSR codes (FAST'16) *A Tale of Two Erasure Codes in HDFS (FAST'15) *Strong and Efficient Consistency with Consistency-Aware Durability (FAST'20) *Pinpointing Crash-Consistency Bugs in the HPC I/O Stack: A Cross-Layer Approach (SC'21)
Week 12 Apr 01, 03, 05	**Project Midterm Report**	--
Week 13	**Project Midterm Report**	--

Apr 08, 10, 12		
Week 14 Apr 15, 17, 19	**Computational Storage** **[P07]** NeSSA: Near-Storage Data Selection for Accelerated Machine Learning Training (HotStorage'23) **[P08]** Closing the B+-tree vs. LSM-tree Write Amplification Gap on Modern Storage Hardware with Built-in Transparent Compression (FAST'22)	*Breathing New Life into an Old Tree: Resolving Logging Dilemma of B+-tree on Modern Computational Storage Drives (VLDB'23) *Understanding the Performance Characteristics of Computational Storage Drives: A Case Study with SmartSSD (Electronics'21) *POLARDB Meets Computational Storage: Efficiently Support Analytical Workloads in Cloud-Native Relational Database (FAST'20) *Hardware/Software Co-Programmable Framework for Computational SSDs to Accelerate Deep Learning Service on Large-Scale Graphs (FAST'22)
Week 15 Apr 22, 24, 26	**Data Security: Ransomware** **[P09]** A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions (ACM Computing Surveys'22) **[P10]** Travelling the Hypervisor and SSD: A Tag-Based Approach Against Crypto Ransomware with Fine-Grained Data Recovery (CCS'23)	* RøB: Ransomware over Modern Web Browsers (USENIX Security'23) *Wake Up Digital Forensics' Community and Help Combat Ransomware (S&P'22) *FlashGuard: Leveraging Intrinsic Flash Properties to Defend Against Encryption Ransomware (CCS'17) * ShieldFS: a self-healing, ransomware-aware filesystem (ACSAC'16) *XRP: In-Kernel Storage Functions with eBPF (OSDI'22)

Week 16 (Prep Week) Apr 29, May 01, 03	**Mini-conference: Final Project Report**	--
Week 17 (Final Week) May 09 (Thursday) 7:30 - 9:30 am	**Mini-conference: Final Project Report**	--

Other ISU Policies

Please refer to other policies described in:

(1) "Syllabus Statements" on course Canvas;

(2) ISU CELT website: https://www.celt.iastate.edu/instructional-strategies/preparing-to-teach/how-to-create-an-effective-syllabus/recommended-iowa-state-university-syllabus-statements/

Further Information/Feedback

Please contact Dr. Mai Zheng by email.