CS6218. Principles of Programming Languages & Software Engineering Week 1: Logistics & Motivation



National University of Singapore

Logistics & Overview

© Copyright National University of Singapore. All Rights Reserved.



- ▶ Instructor: Manuel Rigger (李曼努)
- Lectures: Tue 16:00-18:00
- Location: SR_LT19
- Contact: rigger@nus.edu.sg

Module Information



CS6218 Principles of Prog. Languages & Software Engineering [2210]

Module Content

This module will focus on the latest important research in ensuring the correctness, reliability, security, and performance of data-centric systems. It will approach this topic through a software engineering and programming languages lens, providing a broad perspective by considering systems reaching from traditional relational database systems to applications such as machine learning as well as techniques reaching from automated software testing to human-centric approaches.

Find more information about this module at its homepage @ .

Basic Information

- Instructor: Manuel Rigger ₽
- Lectures: Tue 16:00-18:00
- Location: SR_LT19
- Contact: <u>rigger@nus.edu.sg</u>

https://canvas.nus.edu.sg/courses/25019

Module Information



* » CS6218: Principles of Prog. Languages & Software Engineering (2022)

CS6218: Principles of Prog. Languages & Software Engineering (2022)

Ensuring the Correctness and Reliability of Data-Centric Systems

This module will focus on the latest important research in ensuring the correctness, reliability, security, and performance of data-centric systems. It will approach this topic through a software engineering and programming languages lens, providing a broad perspective by considering systems reaching from traditional relational database systems to applications such as machine learning as well as techniques reaching from automated software testing to human-centric approaches. The module's website is still under construction and content (including the grading scheme) might still change.

Basic Information

- Instructor: Manuel Rigger
- Lectures: Tue 16:00-18:00
- Location: SR_LT19
- Contact: rigger@nus.edu.sg

https://manuelrigger.at/teaching/CS6218/

Module Contents

This module will focus on the latest important research in ensuring the correctness, reliability, security, and performance of data-centric systems. It will approach this topic through a software engineering and programming languages lens, providing a broad perspective by considering systems reaching from traditional relational database systems to applications such as machine learning as well as techniques reaching from automated software testing to human-centric approaches.

SE/PL Perspective

- Techniques: automated testing, debugging, verification, ...
- Publication venues: ICSE, ESEC/FSE, PLDI, POPL, ASPLOS, OOPSLA, …
 - But also other venues in which relevant works were published!
- Approaches: white-box and grey-box

About Me

- Enthusiastic about this topic!
 - Lab's focus (https://nus-test.github.io/)
- Designed and developed SQLancer to find logic bugs in database systems
 - Found more than 500 bugs in widely used database systems

PC-chair of <u>DBTest '22</u>, Co-organizer of Dagstuhl seminars on "Ensuring the **Reliability and Robustness of Database** Management Systems"

of Software Technologies Lab



https://github.com/sqlancer/sqlancer



Module Goals: Primary Goals

- Learn about general techniques such as differential testing, metamorphic testing, debugging techniques, programming languages & type systems, etc.
- Learn how these and other techniques can be applied in the context of data-centric systems

Module Goals: Secondary Goals

- Build skills on how to read and judge scientific projects/papers
- Develop a taste on research
- Practice giving talks and giving feedback

Grading Scheme

- 40% project
- 40% presentation
- > 20% attendance and participation

Grading Scheme

- ▶ 40% project
- 40% presentation
- > 20% attendance and participation



- Alone or groups of two (document shared effort via Git)
- Goal: gain hands-on experience
- Research-focused or implementation-focused
 - Graded accordingly
- Deliverable
 - Initial project proposal
 - Final report & project overview presentation (recording or in-person)



- > acmart with sigconf
 (Latex:\documentclass[sigconf]{acmart})
 - https://www.acm.org/publications/proceedings-template
 - Page Limits
 - Initial project proposal: ~1 page
 - Final report: 4 pages

Initial Project Proposal

- Goal: obtain some feedback from me
- Questions that should be answered
 - What is the problem being tackled?
 - Why is the problem important?
 - Is the project research-focused or implementation-focused?
 - What is the proposed solution?
 - What is the proposed timeline?
 - What is the expected outcome of the project?

Initial Project Proposal



Projects

This page is still under construction.

The goal of the project is to explore one of the module's topics in detail and gain hands-on experience with it.

Logistics

- · Projects should be done in groups of two, or alone.
- Projects should be submitted as GitHub repositories. If two students work on a project, it should be clear from the commit history that both students contributed.

Project ideas

Every group is expected to independently work on a project related to the module's focus. The projects can either be focused on the research or implementation aspect. Potential directions for high-level suggestions include the following:

- Designing and implementing a new approach for testing, verifying, or debugging data-centric systems
- Implementing a known approach in a new system
- Conducting an empirical study, that is, measuring and analyzing the status quo to gain actionable insights (i.e., what can we learn from the study that allows us to implement better systems?)

https://manuelrigger.at/teaching/CS6218/projects.html



(Self) Plagiarism

- Fine to build on previously submitted projects (e.g., for another module)
 - But: highlight the proposed differences and enhancements in the report
- No paper presentation about a paper previously presented

Final Report

Follow a typical paper structure, for example:

- Abstract
- Introduction
- Approach
- Implementation
- Evaluation
- (Related Work)
- Conclusion

Project Presentation

- 15 minutes time limit
- Give an overview of the project
- If sufficient time, we'll have in-class presentations

Grading Scheme

- 40% project
- 40% presentation
- 20% attendance and participation

Presentation

Everyone will present 1—2 papers

Number depends on the number of attendees

Subsequent Q/A and presentation feedback

Presentations: Paper Selection

https://nus-cs6218-2022.hotcrp.com/reviewprefs

- System used for paper reviewing
- We will use it only for bidding and assigning papers

More papers than attendees to account for different interests



Presentations: Other Papers

- Inherently interdisciplinary topic with many directions and strands of research
- Suggest an alternative paper that you would like to present

- "How to Give a Great Research Talk"
- By Simon Peyton Jones

Why you should listen to this talk

- Because many research talks are poor...
- ...and quite simple things can make your talks much better
- Because everyone benefits from good talks
 - Your audience benefits from your hard-won insights
 - You benefit from their informed feedback
- Because a research talk gives you access to the world's most priceless commodity: the time and attention of other people. Don't waste it!





"How to Design Talks" By Ranjit Jhala





https://www.youtube.com/watch?v=aFT79TmffPk

Andreas Zeller



Andreas Zeller is faculty at the CISPA Helmholtz Center for Information Security and professor for Software Engineering at Saarland University, both in Saarbrücken, Germany. His research on automated debugging, mining software archives, specification mining, and security testing has proven highly influential. Zeller is an ACM Fellow and holds an ACM SIGSOFT Outstanding Research Award.

☑ office-zeller@cispa.de

- +49 681 87083-1001
- @AndreasZeller

22 October 2013

Summarizing your presentation with miniature slides

by Andreas Zeller

The slide that drives me nuts

As part of my job, I listen to talks a lot. There's good talks, and there's bad talks. But nothing can drive me as crazy as making a slide like this your final slide:







London Restaurants



1.000

	Travel Cluster
Description	APP
APIs used	ACCESS-ANE INCENTION STATE

Key Findings

- Of the top 5 outliers per cluster. 26% show unadvertised (covert) behavior
- Typically ad frameworks (apploving, airpush)
- Using OC-SVM as a classifier of APIs per cluster, we could flag 56% of novel malware as such
- Current work: Dynamic API usage, information flow, user authorization

http://www.st.cs.uni-saarland.de/chabada/

What's wrong with showing "Thank you!" at the end? Or "Questions?"? Or slide like this showing up at the end, there's three problems:



both? The problem is that most talks are followed by a discussion. With a

© Copyright National University of Singapore. All Rights Reserved.

https://andreas-zeller.info/2013/10/22/summarizing-your-presentation-with.html

How could we visualize this?

Use visual elements rather than textVector graphics whenever possible

- Use own examples and figures to demonstrate a deep understanding
- Including a motivating/running example can help with organizing the presentation
- Making good slides and practicing talks needs lots of time

Designing Presentations: Visual Elements

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque porttitor, lorem quis cursus gravida, sapien augue venenatis mi, sed semper felis velit a odio. Aenean imperdiet lacus non ipsum commodo feugiat. Nunc sodales in neque id laoreet. Sed sagittis, augue eget mollis suscipit, nisi tellus ullamcorper erat, et tincidunt dui turpis eu dolor. Cras guam tortor, mattis vel ullamcorper in, tincidunt convallis urna. Sed interdum, risus at ultricies venenatis, metus felis viverra magna, at viverra nulla nulla at nisi. Ut semper molestie varius. Nunc sed bibendum eros. Curabitur ultricies, massa et imperdiet fringilla, ante tortor lacinia massa, non euismod sapien orci nec dui. Phasellus nec dignissim leo. Vivamus lacinia tellus et lorem malesuada fermentum. Sed justo tellus, tempus in portitor ut, laoreet ac mauris. Cras scelerisque libero nec risus ullamcorper, et convallis ex eleifend. Nunc dictum velit venenatis lacus auctor rhoncus. Aliguam sodales cursus metus, in rutrum nisi laoreet et. Donec pulvinar rutrum nulla, sit amet aliguet magna laoreet sed. Vestibulum mattis, odio pulvinar blandit tincidunt, magna nisi pharetra justo, pretium conseguat libero eros nec dolor. Maecenas eget nibh sem. Sed nunc leo, tincidunt sit amet condimentum ultricies, porttitor a libero. Etiam placerat blandit odio, vel viverra libero rhoncus id. Aliguam tincidunt finibus est in ullamcorper. Morbi rutrum, ante ut vestibulum elementum, tellus mi laoreet tortor, non interdum ipsum nibh sed risus. Pellentesque at purus nec orci elementum sollicitudin. Cras at metus vel augue elementum tincidunt ut volutpat ipsum. Nulla facilisi. Cras suscipit convallis ante vitae congue. Donec ultricies eros non nunc vehicula, vitae blandit velit ultrices. Mauris est nibh, pretium sed posuere sagittis, tristique vitae lectus. Integer ullamcorper sed massa et maximus. Duis ornare nec libero quis congue. Etiam et mi mollis, blandit nunc et, imperdiet sem. Vestibulum lacus sapien, sollicitudin eget semper quis, maximus at lacus. Curabitur eget ligula vitae massa porttitor eleifend. Sed sed dui vitae lacus vehicula venenatis. Nulla tempus hendrerit eros, conseguat elementum felis. Morbi aliguet sem ac negue dapibus, nec lobortis dolor lobortis. Etiam ut est nec magna mollis rutrum. Ut aliguam interdum tristigue. Pellentesque finibus ipsum et mi auctor, quis mollis dui porta. Etiam vitae nibh elit. Duis vulputate arcu elit, eget ornare sapien tristique sed. Sed imperdiet odio vitae mollis fringilla. Sed aliquet fringilla nulla vitae dapibus. Suspendisse consequat finibus lacus, sed fringilla leo molestie id. Integer non tristique odio, in pharetra nulla. Nunc varius urna a dolor vestibulum, eget fringilla lorem accumsan. Praesent non eros malesuada libero tristique rutrum. Pellentesque blandit lectus odio, vel dignissim nisi ultricies sit amet. Cras sit amet sapien nibh. Curabitur ut mi vel tellus auctor facilisis ac at odio. Vivamus vulputate lorem vel ligula cursus elementum. Sed venenatis magna at mattis elementum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliguam id lectus imperdiet, ultrices est viverra, ornare nulla. Vestibulum turpis nisi, dictum id lobortis eget, ornare ut justo. Maecenas ut tristique orci, ac viverra elit. Maecenas sit amet justo pellentesque lorem ultrices porttitor. Sed sed eros quis nisl tempor vehicula vel sed dui. Nulla bibendum consectetur est, sed dapibus massa malesuada non. Quisque euismod ex erat, nec tincidunt odio semper ut. Ut pretium magna eget nisl semper condimentum. Morbi sit amet suscipit lectus, non ullamcorper neque. Sed pharetra odio quam, et faucibus dui condimentum at. Mauris mi tortor, placerat posuere magna in, faucibus sagittis tellus. Integer mollis erat sed tortor rutrum congue. In rhoncus ultricies porta. Sed luctus ipsum vel ultricies venenatis. Donec pharetra eu massa nec rhoncus. Nunc cursus lorem nec velit faucibus viverra. Etiam sollicitudin, neque sed congue viverra, mauris dui laoreet massa, nec vestibulum orci diam non arcu. Morbi eu vestibulum mi, posuere sollicitudin sapien. Nulla elementum id turpis in dapibus. Duis fermentum orci ac sollicitudin sagittis. Vivamus iaculis tempus ex, sed aliquet lectus. Praesent ultricies lectus id leo laoreet, non fringilla quam convallis. Curabitur accumsan posuere feugiat. In nisi dui, hendrerit ac tristique ac, vehicula non quam. Curabitur id fringilla ante. Aliquam lobortis tortor ipsum. Nunc luctus, dui at molestie fringilla, nisl leo semper orci, quis venenatis justo nisl ut risus. Ut id arcu velit. Vestibulum ultricies sodales est, eu sodales augue hendrerit vitae. Pellentesque nec magna a nisi tempor dapibus non luctus arcu.

"Wall of text"





(Simple) figures

Designing Presentations: Vector Graphics



Use visual elements rather than text

- Vector graphics whenever possible
- Use own examples and figures to demonstrate a deep understanding
- Including a motivating/running example can help with organizing the presentation
- Making good slides and practicing talks needs lots of time

Giving Feedback

- Giving constructive feedback is difficult
- Frameworks can help
 - What/why
 - Feedforward
 - Feedback sandwich

Giving Feedback: What/why

- Explain what the presenter did and why it was effective (or not)
- Example 1: You used many visuals, which made the presentation lively and easy to understand.
- Example 2: You were speaking fast, which made it difficult to follow the contents.
- Issue: giving direct negative feedback can be difficult to give and digest

Giving Feedback: Feedforward

Focus on the future rather than past

- More positive
- Example: For the next talk, it would be helpful to speak slower and add some breaks.

Giving Feedback: Feedback Sandwich

- Start with some positive feedback
- Provide constructive criticism
- End on a positive note

Example: I overall liked the talk. It was clear to the audience that you put much work in designing the examples and figures. Despite this, it was difficult to follow the talk since you talked very quickly. I think by trying to add some breaks, you could make your future talks even better.

Grading Scheme

- 40% project
- 40% presentation

20% attendance and participation

Attendance

- One unexcused no-show allowed (i.e., no need to inform me)
- More for valid reasons (e.g., paper presentation at a conference)
Timeline

- Proposing additional papers: August 17
- Paper bidding: August 18
- Team composition: September 15
 - One day before the project proposal submission to set up Canvas
- Project proposal: September 16
 - Finish before recess!
- Final report & recording: TBD

Timeline

- week 02 (16/08/22): Class introduction & logistics
- week 03 (23/08/22): Reliability techniques & SQLancer
- week 04 (30/08/22): Test oracles I: differential testing
- week 05 (06/09/22): Test oracles II: metamorphic testing
- week 06 (13/09/22): Test oracles III: potpourri
- Recess
- week 07 (27/09/22): Test case generation
- week 08 (04/10/22): TBD
- week 09 (11/10/22): TBD
- week 10 (18/10/22): TBD
- week 11 (25/10/22): TBD
- week 12 (01/11/22): TBD
- week 13 (08/11/22): Backup/project presentations

Volunteers: first two presentations

- Two volunteers needed
- August 30 (two weeks)
- Focus: differential testing

https://manuelrigger.at/teaching/CS6218/readings.html#different ial-testing

Differential Testing

- Data-Oriented Differential Testing of Object-Relational Mapping Systems (ICSE '21) [GitHub] [YouTube]
- Finding bugs in Gremlin-based graph database systems via Randomized differential testing (ISSTA '21) [GitHub] [YouTube]
- DiffStream: Differential Output Testing for Stream Processing Programs (OOPSLA '20) [GitHub] [YouTube]
- APOLLO: Automatic Detection and Diagnosis of Performance Regressions in Database Systems (VLDB '19) [GitHub] [YouTube],

What About You?

What is your background and/or your (research) interests?

- Why are you taking this class?
- What are your expectations?

Interviewer: Why do you want this job? Me: I've always been passionate about being able to afford food



Ensuring the Correctness and Reliability of Data-centric Systems

What are Data-Centric Systems?

- Broad umbrella term
- System in which data is an important asset
- Heterogeneous landscape of systems with partly overlapping functionality

Examples: Database Systems

- Relational database systems
- NoSQL systems
 - Document stores
 - Graph stores
 - Key-value stores
- NewSQL systems

Examples: Database Systems

Relational database systems

- NoSQL systems
 - Document stores
 - Graph stores
 - Key-value stores
- NewSQL systems







CREATE TABLE t0(c0 INTEGER);



Table or Relation

Column or Attribute



Row or Tuple



INSERT INTO t0(c0) **VALUES** (0), (1), (2); **Database Management System** 0 Stored in t0. **100. 111** Database db

© Copyright National University of Singapore. All Rights Reserved.

Database Management Systems (DBMS) SELECT * FROM t0 WHERE ϕ ; **Database Management System** Stored in t0: (c0: INT)-Database db 0 1 2

© Copyright National University of Singapore. All Rights Reserved.



© Copyright National University of Singapore. All Rights Reserved.

Importance of SQL

- SQL is one of the most popular programming languages adopted by many data-centric systems
- Pronunciation: Sequel vs S.Q.L



Brushing Up Your SQL Knowledge?

- Advanced SQL" Lectures
- By Torsten Grust
- Recommended if you want to understand advanced SQL concepts

3 This Course

- We will explore the wide variety of guery and procedural constructs in SOL.
- How much computation can we push into the DBMS and thus towards the data?
- Where are the limits of expressiveness and pragmatics?
- Have fun along the way! 😎 We will discuss offbeat applications of SQL beyond employees-departments and TPC-H examples.³

³ The drosophila melanogaster of database research.

Advanced SQL - Chapter 01 - Video #01 - Introduction





Video lecture, part of the course "Advanced SQL", U Tübingen, summer 2020. Read by Torsten

SHOW MORE

2.19K subscribers

13.560 views · Apr 17, 2020

Grust

15:12 / 29:1

Database Systems Research Group at U Tübingen



Brushing Up Your Database Knowledge?

- "Database Systems" Lectures
- By Andy Pavlo
- Recommend if you want to understand how database systems are implemented





Examples: Database Systems

- Relational database systems
- NoSQL systems
 - Document stores
 - Graph stores
 - Key-value stores
- NewSQL systems

Graph Databases

- NodesRelationships
- Properties



(p:Person {name: "Jennifer"})-[rel:LIKES]->(g:Technology {type: "Graphs"})



- Relational database systems
- NoSQL systems
 - Document stores
 - Graph stores
 - Key-value stores
- NewSQL systems

Data Manipulation Frameworks

Examples: Data Manipulation Frameworks



Installation

Package overview

Getting started tutorials

What kind of data does pandas handle? How do I read and write tabular data? How do I select a subset of a DataFrame ?

How to create plots in pandas?

How to create new columns derived from existing columns?

How to calculate summary statistics?

How to reshape the layout of tables? How to combine data from multiple tables? How to handle time series data with ease? How to manipulate textual data? Comparison with other tools Community tutorials

Aggregating statistics grouped by category



What is the average age for male versus female Titanic passengers?

https://pandas.pydata.org/pandas-docs/stable/getting_started/intro_tutorials/06_calculate_statistics.html



- Relational database systems
- NoSQL systems
 - Document stores
 - Graph stores
 - Key-value stores
- NewSQL systems
- Data Manipulation Frameworks

Data Processing Frameworks

Examples: Data Processing

Apache Spark

Hadoop

Apache Flink

In this example, we search through the error messages in a log file.
val textFile = sc.textFile("hdfs://...") // Creates a DataFrame having a single column named "line"
val df = textFile.toDF("line")
val errors = df.filter(col("line").like("%ERROR%"))
// Counts all the errors
errors.count()
// Counts errors mentioning MySQL
errors.filter(col("line").like("%MySQL%")).count()
// Fetches the MySQL errors as an array of strings
errors.filter(col("line").like("%MySQL%")).collect()



- Relational database systems
- NoSQL systems
 - Document stores
 - Graph stores
 - Key-value stores
- NewSQL systems
- Data Manipulation Frameworks
- Data Processing Frameworks

AI Applications

AI Applications: Machine Translation



Send feedback

Overview of the Landscape by Matt Turck

MACHINE LEARNING, ARTIFICIAL INTELLIGENCE, AND DATA (MAD) LANDSCAPE 2021

INFRASTRUCTURE -		ANALYTICS	MACHINE LEARNING & ARTIFICIAL INTELLIGENCE -		APPLICATIONS - ENTERPRISE					
STORAGE LANGES HOUSE A CONTRACT OF CONTRAC	DATA MERIODISS MERIO	HPATTORIS VISUALIZATION LÖGER COMPANY AND	MI FATOBAS MI FAT	DataRobot Cata GI Muusaaperter US.A. ZILLIZ Piguozio strodigia Mul Soage ooviedy.al OYMODEL						
Non-construction Non-construction Non-construction Non-construction CONACCE Tomas Tom	DATABASE DOTADASE	Minosci Quencia Alteryx Minosci Quencia Alteryx Minosci Datamere Character Minosci Datamere Character Chord Alteres Minosci Datamere Minosci Datamere Chord Alteres Minosci Datamere Minosci Datamere Minosci Datamere Minosci Datamere Chord Alteres Minosci Datamere M	LABELLING Control Longer Control Lon	Z PRO- ION & BOSSEWA- BILITY Mobilides After Mobilides After Addricks After all Decembra trex LICON truera p.10 Artice mas WHYLABS						
MP 36. December 2012 Construction Constructi		DAIL CALLOG MITECS DOALLING Non BISCONITION Control Splunk> State Response Control Splunk> Control Trace Splunk> State Response Control Splunk> States State Trace Splunk> States State Control Splunk> States State Control Splunk> States State Control Splunk> States State Control Splunk>		SYNTHETIC MEDIA Constantial and Instances Synthesic E descript Synthesic E descript Precospience supertons Constance Fragments V VOCALD	ADVERSING SCALE					
PRALOCE SERVICE MAIL CONSERVATION Ourse conserved Stable Services Stable Services Stable Services	State State Control State Control State	UNITY SLACK Without Billion Without Billion Billion B	HORCOMMAN MARKEN @ OpenAI @ Device Vyrysgeta: @ @ OpenAI @ Device Warrens @ Or Office 		HALLOAR USES CALL CONTROL CONT					

	- OPEN SOURCE																	
FRAMEWORKS	FORMAT -	QUERY / DATA FLOW	DATA -	DATABASES	ORCHESTRATION	INFRA-	DATA OPS	STREAMING &	STAT TOOLS & -		MLOPS & INFRA	AI / MACHINE LEARNING / DEEP LEARNING	(G	SEARCH	LOGGING & MONITORING	VISUALIZATION	COLLABORATION -	SECURITY
Martine Look Martine, annes		sook soi 🎍 ዄ	ACCESS	The same frequence of the Transmission of the same	R Annale Andre P PROPERT	STRUCTURE	W. MARQUEZ	MESSAGING	LANGUAGES	pittin	milita states	Ptensorfiaw	🔏 operCV 🏬 🚓 🖪 Keras BERT 🗟 root Caffe	ekoltment Sol	E Hoans E Hoans	3 co Superset mutpl'illo	Doke	Apache Ranger KNOX
GElink YARN TEZ	ance.	MARTIN AND AND DANS	Databook	🔿 influenti presto 📎 🗇 dividi 💇 💩 Duraponal	🗢 DADETER 🦉 Flyte	Ø	Contractions	Strates Office	GE Sons I		and a state	Benand PH @ OpenAI O Pytere Laterage	theand theanth were owned that	"Eacane + Sphin	X Logitzeh Oranden Yamer	Winterdame Plant Terrorburd	- 0	tury scarras
Se Masos Moon GCDAP	0		Ny magda.	xaoca 🧭 🏔 🚳 🛤 -riak	SMEDIFLOW AND Kerter	S 14505	tobal 5	Anna 🙆 nitie	g parase (9 soly	G1040	DVC Strate	press negoti de contra de contra a		•)···	Past Corter William			and the second se
Andres Areanany HELIX	Annua lant	State Manual Marie	d cken	Same source causes Ppinot codes Faired	UNPI	1 orgo	L. Propert Kanada	Apada badama GiSamba	🤨 julà 💮	•	iter States Manual	Aaronabus (and "refer Covers (Covers)	spady goundar Allest 🖓 🔐 const	Seet 1	1	theston bakeh	ANACONDA	A redy

-		DAT	DATA DESCUIRCES					
		DAI	DATA RESOURCES					
DATA MARKETPLACES	FINANCIAL & ECONOMIC DATA	AIR/SPACE/SEA	PEOPLE / ENTITIES	LOCATION INTELLIGENCE	OTHER	DATA SERVICES	INCUBATORS & SCHOOLS	RESEARCH
& DISCOVERY	Bioemberg C THEMEON RELITERS D DEM JENES Quand OCTIMULE		Z zoominfo acxism terperion		MAGENET LANSING	O GLIWITUMBLACK O Booz Alien Hamilton		OpenAI Google Research facebook research
Constant Constant Conversion	Series Chever And Street & Sections	0.58	IPECON GINSIOOVIEW 📦 Restweet	Pacelle @esri Ander Ander		Notes -		
States and the second	xignite mass earnest predate ADC tink? Thereten	WINDWARD Massifully USA	Quantcast	A suble transmission		kagglo ElectrifAl froctokes AEXL	A DataElto galvanizo (* Herris INSIGHT	and
N narrative Detabet Service.		ExoLabs Q. Fills & Symperior	Demyst melissa Zzignal -	P cocord N obrigation	VERTORE Comscore	DataKind innoPLCKUS*	The Data Incurtagoal	AIZ ANTHROPIC Salk

Version 3.0 - November 2021

mattturck.com/data2021



с сорунула налоная онистоку ог отгуарого. Ля наука неостиса.

Why Are They Important?



© Copyright National University of Singapore. All Rights Reserved.

https://www.visualcapitalist.com/wp-content/uploads/2019/04/data-generated-each-day-full.html

Scales of Database Systems

- Embedded databases
- Standard database systems
- Data warehouses and lakehouse systems

Why Are They Important?

- Embedded databases and edge computing
- Since SQLite is used extensively in every smartphone, and there are more than 4.0 billion (4.0e9) smartphones in active use, each holding hundreds of SQLite database files, it is seems likely that there are over one trillion (1e12) SQLite databases in active use.

Why Are They Important?

 "Standard" relational database systems like PostgreSQL, MySQL, Microsoft SQL Server, Oracle used by many small and mid-sized companies



https://survey.stackoverflow.co/2022/#most-popular-technologies-database-prof
Why Are They Important?

- Companies store and process large-scale data (petabytes)
- Machine learning and business intelligence solutions
- Cloud solutions



CIDR 2022 Keynote 1: The Databricks Lakehouse Platform by Matei Zaharia

https://www.youtube.com/watch?v=LJb1tR3i9hU

Challenges

- Rapidly-growing data amounts
- Increasing reliance on data-centric solutions
- Heterogeneous landscape of data-centric solutions

Is our data storage and processing infrastructure reliable?

How Reliable is SQLite?



Design process inspired by DO-178B for **safety-critical software systems** in an **aircraft**

SQLite's test cases achieve 100% branch test coverage

SQLite (~140,000 LOC) has **640 times** as much test code as source code

Anomaly testing (out-of-memory, I/O error, power failures)

Are These Systems Correct and Reliable?







Different binary representation







https://bugs.mysql.com/bug.php?id=99122





https://bugs.mysql.com/bug.php?id=99122



We could **find the bug without having an accurate understanding** ourselves



How Can We Ensure Their Reliability?

- Automated testing
 - Differential testing
 - Metamorphic testing
 - Isolation-level testing
- Debugging
- Development environments
- Static & dynamic analysis
- Language design
- Formal verification

Kinds of Bugs

- Crash bugs
- Logic or correctness bugs
- Isolation-level bugs
- Memory management bugs
- Usability issues
- Performance bugs/missed optimization opportunities



Module Contents

@ Copyright National University of Singapore All Rights Reserved

This module will focus on the latest important research in ensuring the correctness, reliability, security, and performance of data-centric systems. It will approach this topic through a software engineering and programming languages lens, providing a broad perspective by considering systems reaching from traditional relational database systems to applications such as machine learning as well as techniques reaching from automated software testing to human-centric approaches.







https://manuelrigger.at/teaching/CS6218/