CS W186 Spring 2020
Final Written Technical Report

Introduction

The purpose of this assignment is to complete a written technical report (by yourself or with **one** partner) on a topic in database systems. The choice of topic is fairly open-ended as students can delve into any area they choose -- research, industry applications, coding, benchmark testing, non-relational models, etc.

Potential Options

The following are potential tasks you can do for this assignment, of which you will only need to choose **one**. Keep in mind you may instead create your own task as long as you clearly explain it and the task meets the requirements listed in the rubric. If you do choose one of the tasks below, make sure to specify which one.

1. **Pick a topic covered in this class** (sorting, hashing, joins, query optimization, parallel query processing, concurrency, recovery, transactions, etc.) and find **two** database system research papers that cover the topic at a more detailed level. **Compare and contrast** the systems and discuss their performance tradeoffs and relevance to topics studied in this class. The system comparisons should include both metrics discussed in this class, and ones that you learn about by reading the papers.
2. **Pick a modern relational database** that you are interested in learning more about, such as MySQL. Pick **two** major components of a DBMS (sorting, hashing, joins, query optimization, parallel query processing, concurrency, recovery, transactions, etc.) that were presented in this class, and provide an in-depth explanation of how they work in the relational database. You should refer to official documentation, source code, and academic papers to gain a thorough understanding. Explain how this database has assumptions that differ from the ones that were presented in class. What were the motivations for these differences?
3. **Pick a topic covered in this class** and find **three** related published papers. Answer the following questions for each paper. Your answers should be detailed and demonstrate critical thinking (i.e. responses should be topic-oriented, rather than surface-level - for example, don't comment on the author's writing style.)
    - What are the three biggest strengths of the paper?
    - What are three weaknesses?
    - What "new ideas" does this paper present? Do not simply restate the abstract!
    - What evidence do they offer to support these ideas?
    - What contradictory evidence, if any, is provided?
    - What are significant weaknesses in the authors' argument or issues not considered?
    - How are the ideas in the paper enabled by the workloads and infrastructure of its day?
    - In what ways is it constrained by that context?
    - Is this paper still relevant today?
    - What additional insights, if any, has later research provided into the paper's topic(s)?
    - What conclusions can you draw about the topic from the three papers? (e.g. Is there consensus? Is there a great debate in the community?)
4. **Pick a modern NoSQL database**, such as DynamoDB. Provide an in-depth comparison and contrast of relational and NoSQL databases. Consider their performance tradeoffs and workloads. Explain the high level architecture of the system that you choose. Explain what workloads the system

performs well for, and which workloads it performs poorly for. Provide details on at least two components of the system that differ from the traditional DBMS model presented in this class. Use specific technical materials, benchmarks, etc. in your report, and optionally run your own experiments on this database.

Grading Rubric

This assignment is meant to be very open-ended with a lot of flexibility for students to explore what's possible in the world of databases. In terms of grading, your report will be graded on the following criteria:

- *Novelty (20%)*
    - An exceptional report will explore new ideas beyond the scope of the class and demonstrate the student's initiative.
- *Relevance (30%)*
    - An exceptional report will relate ideas back to topics and assumptions learned in this class, while discussing any important similarities or differences between the systems studied/implemented/benchmarked. Feel free to use course content (lectures, notes, etc.) as references in your report.
- *Structure (20%)*
    - This is not meant to evaluate the writing style of your report, but to ensure that there is a logical organization to the ideas presented in the report. The final product should not just be a list of bullet points. The structure will depend on the option chosen. For example, a potential structure for Task 3 may look like:
        - Introduction
        - Paper 1
        - Paper 2
        - Paper 3
        - Conclusion
        - Reflection [See below]
    - Please make sure to include a list of references at the very end of your report.
- *Depth (20%)*
    - While there is no exact word count requirement for this assignment, a report with the expected depth of discussion should have around 1200-1500 words (4-6 pages double-spaced)
    - The list of references at the end do not contribute to the word count.
    - You should use at least 3 references that are not CSW186 course material in your report.
- *Reflection (10%)*
    - Concisely highlight 2-3 key points the reader should takeaway from the report.
    - Choose one of the following:
        - What two questions would you ask the authors of the papers, or the developers of the system you studied to gain insight? What would you learn from this?
        - Come up with two questions that you would like to study next in order to build on your research. What do you think you would learn?

- *Extra Credit (Up to 15% of the assignment weight, or 2.25% of the total course weight.)*
  There is opportunity for extra credit on this report if you complete any of the following tasks:
    - **Pick any modern database**, relational or NoSQL, and go through the process of running an instance of the system yourself and interacting with it to draw insights. Document the steps

you take to get the system set up. Discuss how easy or difficult it was to get running, and whether you would use this database as a user. Measure some performance metrics, such as latency for executing queries, or throughput of operations by finding workloads or generating workloads yourself. Graphs would be a nice addition to help visualize and measure performance. The TPC benchmarks are a good place to start. Explain any anomalies that you notice. Provide any scripts or programs that you write to extract this data.

- **Pick a topic from the class** that most interests you and spend time implementing a feature that is not already a part of the Project database. You may choose to build off of the project code, or you may start from scratch. For example, you could decide to program a 2-Phase Commit protocol with logging for recovery. Submit a design document of your implementation along with the code. Measure relevant performance metrics and discuss how they can be improved.
- For any of the other tasks, you may extend the topic by running experiments or tests to validate your report. Provide any scripts, datasets, or DB dumps that you used to complete your analysis.
- *Extra credit is not a binary option and is not guaranteed, but exceptional efforts and reports will be rewarded.*

Resources

To help get you started, we've compiled the following list of resources:

- Readings in Database Systems, 5th Edition
- http://bit.ly/BerkeleyDBPrelimList
- A live list of papers related to course content: CS W186 Spring 2020 Technical Report Papers
- CS286
- Things I Wished More Developers Knew About Databases
- Jepsen.io (Especially for NoSQL or experimental ideas)
- TPC Benchmarks

There are many additional places worth searching like the ACM Digital Library and arxiv.org. We highly encourage you to share interesting resources liberally with the course on Piazza.

A note on paper reading...

For many of you, this will be your first time reading an academic research paper. Congrats! The format of an academic paper can be a bit daunting, and feel discouraged if you need to reread something a few times to get the point. If you'd like a good, check out this paper, *How to Read a Paper*.

Submission

You may choose to work by yourself or with **one** partner on this assignment. Both partners should turn in the exact same assignment. You may discuss ideas, and share references with others, but you may not edit or review reports of others. We will have submission details out shortly. Your final report should be typed, double-spaced, and submitted to by **Monday May 11, 11:59PM PDT**. You may use slip minutes for your assignment, but may not use more than 48 hours worth of time. This is to ensure each assignment can receive peer reviews.

Peer Reviews

An important aspect of the research paper development process is peer review. After you submit your completed paper, you will then be assigned three submissions written by someone else in the class to review. You will then apply the above rubric to peer-grade the submission and provide constructive feedback. All reports and reviews will be anonymous. *Each partner must individually complete the three peer reviews.* We will audit peer reviews for accuracy and correct scores as necessary.

Peer Reviews are due **Friday, May 15 at 11:59PM PDT**. You may use slip minutes on your peer reviews.

FAQ

For more information on the reasoning behind the technical report assignment, please refer to @746 and @917 on Piazza.

A Parting Thought

This is a new experiment in CS W186, and a somewhat unique experience among computer science courses. You have learned so much this semester, and now is your chance to show it off in your own way. No need to fret about miscounting I/Os or hidden test cases. There's no compiler to yell at you! We've given you some ideas, which we hope are good starting points, but they're merely that: a start. The unknown can be a frightening place, but once you start digging, you never know what interesting gems you may find.