Project Ideas

CS 764, Fall 2020

This document is just to give you some ideas about potential project topics. You are encouraged to explore other ideas that interest you (e.g., those relate to your own research projects) but are not in this document.

1. There is significant work on join algorithms for main-memory, multi-core, multi-socket settings for both hash and sort-based algorithms. See:
   - Stefan Schuh, Xiao Chen, Jens Dittrich: An Experimental Comparison of Thirteen Relational Equi-Joins in Main Memory. SIGMOD Conference 2016: 1961-1976
   - Spyros Blanas, Yinan Li, Jignesh M. Patel: Design and evaluation of main memory hash join algorithms for multi-core CPUs. SIGMOD Conference 2011: 37-48
   - Changkyu Kim, Eric Sedlar, Jatin Chhugani, Tim Kaldewey, Anthony D. Nguyen, Andrea Di Blas, Victor W. Lee, Nadathur Satish, Pradeep Dubey: Sort vs. Hash Revisited: Fast Join Implementation on Modern Multi-Core CPUs. PVLDB 2(2): 1378-1389 (2009)

   Some of the previous work suggests that radix join has the highest performance when joining two tables. However, many real-world in-memory databases systems use non-partitioned hash joins for joining multiple tables. There are a few interesting project topics to investigate under this umbrella:
   - Conduct a survey on recent join algorithms in multiple core, or distributed system, or new hardware (e.g., GPU) and summarize the key findings of previous work.
   - Evaluate radix join vs. non-partitioned hash joins as the number of tables in the join changes. Find out which algorithm performs best under what scenario. Is it possible to combine them to achieve the best of both?

2. Non-volatile memory is poised to transform the memory hierarchy of computer systems over the next decade. It promises to offer byte-addressable permanent storage an order of magnitude larger than RAM at latencies several orders of magnitude less than flash/SSDs. How does NVM change the way we implement buffer management and database logging? What new optimizations can NVM enable for a database system? You can start with the following papers:
   - van Renen, Alexander, et al. "Managing non-volatile memory in database systems." SIGMOD 2018
   - Arulraj, Joy, Matthew Perron, and Andrew Pavlo. "Write-behind logging." VLDB 2016

3. The recent developments in machine learning is affecting the way people build database systems. Below are some of the papers in this area. You can conduct a survey of recent work in this area, or dive deeper into a particular idea trying to improve the state-of-the-art.
   - Kraska, Tim, et al. "The case for learned index structures." SIGMOD 2018
   - Kraska, Tim, et al. "Sagedb: A learned database system." (2019).
   - Pavlo, Andrew, et al. "Self-Driving Database Management Systems." CIDR. Vol. 4. 2017.
   - Van Aken, Dana, et al. "Automatic database management system tuning through large-scale machine learning." SIGMOD 2017

4. Hybrid transactional/analytical processing (HTAP) supports efficient transactional and analytical processing in a single database system. A number of such systems have been developed in multiple companies. You can conduct a survey of recently built systems, identify limitations in existing solutions, and trying to propose new idea. Below are some relevant papers:
   - Kemper, Alfons, and Thomas Neumann. "HyPer: A hybrid OLTP&OLAP main memory database system based on virtual memory snapshots." ICDE 2011
   - Özcan, Fatma, Yuanyuan Tian, and Pinar Tözün. "Hybrid transactional/analytical processing: A survey." SIGMOD 2017
   - Yang, Jiacheng, et al. "F1 Lightning: HTAP as a Service." VLDB 2020

- Huang, Dongxu, et al. "TiDB: a Raft-based HTAP database." VLDB 2020

5. Transactions have been studied by the database and system communities for multiple decades and many research problems have been investigated by generations of researchers. The advent of new hardware, systems architectures, and application demands create new challenges in transaction processing. For each of the following topics, you can conduct a survey or dive deeper into a specific topic to develop new ideas.
    - What are the common techniques to handle high-contention transactions? How do they compare in terms of effectiveness?
        - Huang, Yihe, et al. "Opportunities for optimism in contended main-memory multicore transactions." VLDB 2020
        - Tanabe, Takayuki, et al. "An Analysis of Concurrency Control Protocols for In-Memory Databases with CCBench", arXiv 2020.
        - Yu, Xiangyao, et al. "Tictoc: Time traveling optimistic concurrency control." SIGMOD 2016
    - What's the tradeoff between different concurrency control protocols in terms of tail latency?
        - Yu, Xiangyao, et al. "Staring into the abyss: An evaluation of concurrency control with one thousand cores." VLDB 2014
        - Lim, Hyeontaek, Michael Kaminsky, and David G. Andersen. "Cicada: Dependably fast multi-core in-memory transactions." SIGMOD 2017
    - How does cloud change the traditional wisdom of building transaction processing systems?
        - Verbitski, Alexandre, et al. "Amazon aurora: Design considerations for high throughput cloud-native relational databases." SIGMOD 2017
        - Verbitski, Alexandre, et al. "Amazon aurora: On avoiding distributed consensus for i/os, commits, and membership changes." SIGMOD 2018
    - What are the main disadvantages of deterministic databases today? What are the possible solutions to these problems?
        - Thomson, Alexander, et al. "Calvin: fast distributed transactions for partitioned database systems." SIGMOD 2012.
        - Abadi, Daniel J., and Jose M. Faleiro. "An overview of deterministic database systems." Communications of the ACM, 2018
        - Lu, Yi, et al. "Aria: a fast and practical deterministic OLTP database." VLDB 2020

6. GPUs are good at massively parallel computation tasks. How can GPU accelerate data analytics workloads?
    - Shanbhag, Anil, Samuel Madden, and Xiangyao Yu. "A Study of the Fundamental Performance Characteristics of GPUs and CPUs for Database Analytics." SIGMOD 2020.
    - Lutz, Clemens, et al. "Pump Up the Volume: Processing Large Data on GPUs with Fast Interconnects." SIGMOD 2020

7. Investigate data compression and decompression algorithms in CPU or GPU, or both
    - Willhalm, Thomas, et al. "SIMD-scan: ultra fast in-memory table scan using on-chip vector processing units." VLDB 2009
    - Polychroniou, Orestis, and Kenneth A. Ross. "Efficient lightweight compression alongside fast scans." DaMoN@SIGMOD 2015

8. Compare different indexing approaches used for modern systems
    - Wang, Ziqi, et al. "Building a bw-tree takes more than just buzz words." SIGMOD 2018
    - Levandoski, Justin J., David B. Lomet, and Sudipta Sengupta. "The Bw-Tree: A B-tree for new hardware platforms." ICDE, 2013.
    - O'Neil, Patrick, et al. "The log-structured merge-tree (LSM-tree)." Acta Informatica, 1996