Fundamentals of Database Systems – PROJECTS (Research)

PGDBA, First Year, 2020–2022

Deadline: February 15, 2021

Total: 10 marks

SUBMISSION INSTRUCTIONS

1. Submit a solution sketch in a single file by the deadline. The solution must be self-explanatory.
2. The solution should include the sections (if applicable): Introduction, Related Work, Terminologies and Definitions, Theory, Methods, Results, Conclusion.
3. Include names and roll numbers of all of your group members (at most 5).
4. Naming convention for your submission file (assuming M is your project number): `projM` (.docx, .doc, .pdf, .tex, etc.).
5. To submit a solution file (say `projM.docx`), ensure that it is not password protected and mail to `<assignisik@gmail.com>` with the subject line as follows: `PGDBA 2020-22 Project M`.

NOTE: The contribution must be novel and non-trivial.

Project 1: [**Optimistic Concurrency Control**] Recent research highlights the superiority of hybrid optimistic concurrency control / locking systems and variations on multiversion concurrency control. They are also useful in managing uncontended multicore transactions. However, these models suffer under varying contention and workloads. Two optimization techniques, namely commit-time updates and timestamp splitting, that can dramatically improve the high-contention performance have recently been proposed [1]. Identify a new application related to it, brainstorm further, and propose something novel.

[1] Huang, Y., Qian, W., Kohler, E., Liskov, B. and Shrira, L., 2020. Opportunities for optimism in contended main-memory multicore transactions. Proceedings of the VLDB Endowment, 13(5), pp.629-642. (Link: `https://vldb2020.org/vldb-2020-awards.html`)

Project 2: [**Joins in Big Data**] We can convert predicates on a relation into data induced predicates that apply on the joining tables using statistical details. Recent research highlights how building data induced predicates using zone-maps, which are already maintained in today's clusters, leads to sizable data skipping gains [1]. Identify limitations of these approaches and accordingly suggest further improvements.

[1] Kandula, S., Orr, L. and Chaudhuri, S., 2019. Pushing data-induced predicates through joins in big-data clusters. Proceedings of the VLDB Endowment, 13(3), pp.252-265. (Link: `https://vldb2020.org/vldb-2020-awards.html`)

Project 3: [**Scalable Search**] Searching for maximum bicliques is an important concern in diverse applications. A recent research has suggested how to address this problem with a scalable

dividide-and-conquer method [1]. This approach minimizes the computational cost by significantly reducing the bipartite graph size for each subproblem while preserving the maximum biclique satisfying certain constraints by exploring the properties of one-hop and two-hop neighbors for each vertex. Suggest an approach to make it further better.

[1] Lyu, B., Qin, L., Lin, X., Zhang, Y., Qian, Z. and Zhou, J., 2020. Maximum biclique search at billion scale. Proceedings of the VLDB Endowment, 13(9), pp.1359-1372. (Link: `https://vldb2020.org/vldb-2020-awards.html`)

Project 4: [**Benchmarking Query Engine**] A recent study has put forward a novel framework for end-to-end benchmarking and performance monitoring of query engines. It can support automated workload summarization, data anonymization, benchmark execution, monitoring, regression identification, and alerting [1]. Brainstorm on finding novel applications of the said concepts to enable comparative end-to-end benchmarking in other industrial environments.

[1] Deep, S., Gruenheid, A., Nagaraj, K., Naito, H., Naughton, J. and Viglas, S., 2020. DIAMetrics: benchmarking query engines at scale. Proceedings of the VLDB Endowment, 13(12), pp.3285-3298. (Link: `https://vldb2020.org/vldb-2020-awards.html`)

Project 5: [**Facebook Storage Engine**] Given the challenges Facebook faces in managing their user database, a recent research has suggested a novel index structure that uses less space and write amplification [1]. This work, proposing a new storage engine, has added many new features with the stadard implementation of RocksDB. Identify further challenges in this approach and suggest some novel improvements.

[1] Matsunobu, Y., Dong, S. and Lee, H., 2020. MyRocks: LSM-tree database storage engine serving Facebook's social graph. Proceedings of the VLDB Endowment, 13(12), pp.3217-3230. (Link: `https://vldb2020.org/vldb-2020-awards.html`)