Practicum in Database Systems

Project 5 intro

Project 5 overview

- Multiple possibilities for extensions:
  - Parallelize your implementation
  - Create ML-based query optimizer
  - Worst-case optimal join algorithms
- Pick one of them
  - Implement it
  - Measure performance impact
  - Submit implementation and report

Parallelization

- Current hardware trends make it imperative to leverage parallelism to achieve high performance in data processing
- Exploit data parallelism with shared memory
- Try to parallelize each physical operator
- Java offers convenient ways to leverage parallelism (e.g., parallelStream)
- See "Massively parallel sort-merge joins in main memory multi-core database systems" (no need to implement precisely the methods described in the paper)
  - https://dl.acm.org/doi/10.14778/2336664.2336678

ML Query Optimizer

- Traditional optimizers make many simplifying assumptions when optimizing queries
- Currently lots of work on leveraging machine learning for more precise optimization
- Implement simple optimizer leveraging machine learning to choose join orders
- See "Neo: a learned query optimizer" (but no need to implement method from this paper)
  - https://arxiv.org/pdf/1904.03711.pdf

Worst-Case Optimal Joins

- This project is for groups with an affinity for theory and algorithms
- Typically, database systems join multiple tables by a series of binary join operations
- Recently, this has been shown to be sub-optimal as intermediate results can be asymptotically larger than the final result
- This has led to worst-case optimal join algorithms that join multiple tables in one operation, thereby avoiding disproportionally large intermediate results
- One of the simpler worst-case optimal join algorithms is described in this paper (you may also choose to implement another one):
  - "Leap-frog trie join: a worst-case optimal join algorithm" https://arxiv.org/abs/1210.0481

Submission

- Implement the selected extension and submit corresponding code
- Benchmark your extended version on a benchmark of your choosing, submit a small (max 2 pages) report comparing performance to your phase 4 implementation
- You get points for passing test (correct query results), code readability, and a reasonably thorough experimental analysis in the report