# Project Milestone: Parallel Query Processing in PostgreSQL With Vectorization

Kai Franz

https://kai-franz.github.io/15418-project/

## Summary

So far, I have constructed microbenchmarks for simple queries that test the DBMS's ability to exploit instruction-level parallelism (ILP). I have profiled PostgreSQL running these benchmarks using VTune, which showed that there is a large number of pipeline slots that are spent core-bound on these workloads.

Additionally, I have finished building a PostgreSQL executor node that can take in tuples, store them in a vector, and output the tuples that satisfy a given predicate. I have begun integrating this vectorized executor with PostgreSQL's other executor nodes. Although there are still a few bugs to fix, I would say I am about 80% of the way towards completing this task. Although this was originally my milestone goal, I realized that I underestimated the complexity of PostgreSQL's execution engine and I will need to spend additional time on this task; however, I realized other tasks, such as supporting predicate evaluation, are simpler than I expected.

## Goals

I am still on track to reaching my 100% goal. However, after a few weeks of working on the project, I realized that the complexity of PostgreSQL's execution engine make my original 125% goal (SIMD predicate evaluation) difficult to execute by the end of the semester. Instead, I have replaced it with a goal that I think is attainable within this project's timeframe, which is to implement Permutable Compiled Queries using vectorized predicate evaluation.

## Goals

- **75%:** Implement a vectorized executor that exploits instruction-level parallelism.
- **100%:** In addition to the previous goal, add support for predicate evaluation.
- **125%:** Use Permutable Compiled Queries to accelerate predicate evaluation.

The demo that I plan to show will be a speedup graph.

## Schedule (Half-Week Increments)

- **11/30:** Integrate vectorized scan
- **12/5:** Finish integrating vectorized scan
- **12/7:** Add support for predicate evaluation
- **12/12:** Run benchmark workloads (TPC-H/TPC-DS). Optimize any bottlenecks.
- **12/14:** Integrate Permutable Compiled Queries into the vectorized executor
- **Final Report:** Have a working vectorized execution engine that supports predicate evaluation.