# Context and motivation

Nowadays big data is booming business. A way to handle all this data in a fast efficient manner is very important for this cause. That is why at this time there are a lot of query optimizers, but these optimizers need statistics about the big data to create good query plans. The problem is in many of these cases such statistics are sparse or even non-existent.   
This is why Borovica, Idreas, Ailamakki, Zukowsku, and Fraser (2015) deisgned a new method for query optimizations decisions, the process of deciding which physical operators are used and in what order. Note that these decisions can affect the response time by several orders of magnitude and are therefore crucial in creating an optimally performing database. The proposed method is called *smooth scan* and its results are compared to those of a *full table scan*, an *index scan* and a *switch scan*. The smooth scan is inspired by adaptive query processing, a technique which uses runtime feedback to modify query processing to get a better response time or more efficient CPU utilization (Deshpande, Ives, & Raman, 2007).

The motivation for this article is the severe impact several scanning methods have on the total execution time, a response time which, in this day and age, becomes increasingly important. It also seems that the costs models for performance estimations deviate increasingly as the system becomes more complex. Combined with the fact that the complexity of modern workloads and the technological shift towards cloud environments, query optimization becomes increasingly important. Because of these reasons there is enough motivation to discuss a new scanning method for query handling.

The report also discusses the trade-off between CPU operations and I/O operations. Because it is possible to do several thousands of CPU operations in the time needed for one I/O operation, this trade-off is very often beneficial for the performance. Since the durations of these operations are subject to continuous change because of the fact hardware is always being improved, this too is a motivation for writing this article.

# Research problem

The main problem addressed in the paper is that once a decision is made by the optimizer, this decision is fixed throughout the execution of a query. This can result in suboptimal plans and even a small estimation error can lead to drastically different performance results. These unpredictable results makes the system non-robust. The authors define robustness as: “the ability of a system to efficiently cope with unexpected and adverse conditions, and deliver near-optimal performance for all query inputs.” In their paper they try to achieve robust query processing.

# Claimed results

To authors implement *Smooth Scan* in PostgreSQL and claim that it achieves robust performance in a range of synthetic and real workloads, while being statistics-oblivious at the same time. Existing approaches fail to do so. This robust behaviour results in significant gains compared to when the original system makes a wrong decision, and only marginal overheads compared to when a correct decision can be made.

*🡪 We can choose one (or more) of the statements below to check, as these provide some more explicit claims.*

*- With Smooth Scan, PostgreSQL demonstrates robust performance. It does not suffer from extreme degradation and achieves good performance for all queries.*

- Smooth scan improves by a factor of 10 when compared to *index scan*  for selectivity 100%.

- Smooth scan on SSD is faster than *full scan* for selectivity lower than 20%, and it is only 10% slower compared to *full scan* for 100% selectivity.

-

# Plan for verification

# Progress

# Bibliography

Borovica, R., Idreos, S., Ailamaki, A., Zukowski, M., & Fraser, C. (2015). Smooth Scan: One access path to rule them all. *IEEE International conference on Data Engineering*.

Deshpande, A., Ives, Z., & Raman, V. (2007). Adaptive query processing. *Foundations and Trends in Databases*, *1*(1), 1-140.