Streaming systems have an advantage over query engines for graph databases with regard to data aggregation (clauses Group by and Order by), because they reduce the set of stored elements only to the aggregated values. However, streaming systems cannot perform pattern matching, unlike query engines.

In this work, we created a static graph database with the Labeled-property graph model and a query engine that performs data aggregation using the classical approach of aggregating data after pattern matching was finished. Subsequently, we adjusted query processing in the query engine by applying streaming systems methods. As the result, the query engine was able to perform data aggregation during pattern matching. The goal of this work was to test whether the above-mentioned adjustments can improve performance of query processing.

We designed and implemented a certain number of single-thread and parallel solutions for both the adjusted and the classical approach. Afterwards, we conducted experiments on real graphs with artificially generated property values in order to test performance of the created solutions. The experiments showed that there were situations where the adjusted approach had better performance. Specifically, it happened in the case of parallel solutions of Order by when sorting by property values, a single-thread solution of Group by and lastly single-thread and parallel solutions of Single group Group by (the query contains aggregation functions and no Group by).