

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

In this thesis we created a static graph database using the Labeled-property data model and a query engine that performs data aggregation using the classical approach of aggregating data as the last step of query processing 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 thesis 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 the adjusted approach and the original approach. Afterwards, we conducted experiments to test performance of the created solutions. The experiments were done on real graphs with artificially generated property values. The experiments showed that there were situations where the adjusted approach was faster.