

Mask Interconnect Latencies Using Approximations

Keywords: interconnects, analytics, speculative execution, approximate query processing

Problem: Modern analytics involve complex queries executed on systems with many hardware components that communicate over a variety of interconnects. These interconnects can become the bottleneck and delay the execution of the whole query. If data cannot be transferred fast enough from operator A to operator B, then all subsequent operators of the query plan stall and task-parallelism is decreased. While traditional query optimization strategies can reduce the costly data transfers, still there is a cost we cannot avoid and depends on the characteristics of the interconnect.

Project: In [1], Sioulas et al. propose a technique that relaxes dependencies in complex nested SQL queries. The technique takes advantage of approximate query processing and speculatively executes parts of the query in order to increase task-parallelism. In case of wrong speculations, corrective actions are incrementally applied.

The goal of this project is to extend the technique of [1] and use it in order to mask the latencies of interconnects. Given an operator that materializes its output and that receives its input through an interconnect, we can “guess” the materialized result and speculatively execute the rest of the plan before the actual data arrives. A common example is shuffling and reduce. In order to aggregate data, we have first to transfer them and group them within the network and this operation can very often become a bottleneck. Instead, we can have in the reduce side a model that approximates the input, we can guess the output of grouping and aggregation and proceed with the execution.

Plan:

1. Create a taxonomy of analytical systems with interconnects. For each category, identify a use-case where this technique could be applicable.
2. Evaluate the performance of the baseline, breakdown the execution time and identify the cost of the interconnect for each case.
3. Implement the technique for a distributed analytical engine (e.g., Spark, Presto) that shuffles data over ethernet.
4. Implement the technique in a case where a faster interconnect is used (e.g., PCIe, IB). What is the margin of performance improvement for each case? How does the bandwidth of the interconnect affect the theoretical speedup?

References

1. P. Sioulas, V. Sanca, I. Mytilinis, A. Ailamaki. “Accelerating complex analytics using speculation”. Conference on Innovative Data Systems Research (CIDR) 2021

Supervisor: Prof. Anastasia Ailamaki, anastasia.ailamaki@epfl.ch

Responsible collaborator(s): Viktor Sanca, viktor.sanca@epfl.ch

Duration: 1 semester