Rapid Data Exploration, increasing AQP accuracy with Query Compilers

Abstract

To enable Interactive Data Analytics, data has to be rapidly explorable. Through Lightweight Modular Staging (LMS) Query Compilers can be generated. Since they compile Queries into lower-level code, applying them results in a major leap in Query Engine performance. Therefore Query Compilation can be a mean to assist interactive data analytics.

I propose research on how to apply Query Compilation in scalable databases that are optimized for chunking data into samples. If query execution can be speed up, it might be possible to either evaluate more chunks or increase chunk sizes to improve Approximate Query Processing (AQP) accuracy.

Connection to the lecture

- idea starting from "Darmstadt Data Analysis Stack"
- presented by **Carsten Binnig** in his Talk "Towards Interactive Data Analytics"
- queries have to take **less than 500ms** to be usable **interactively**

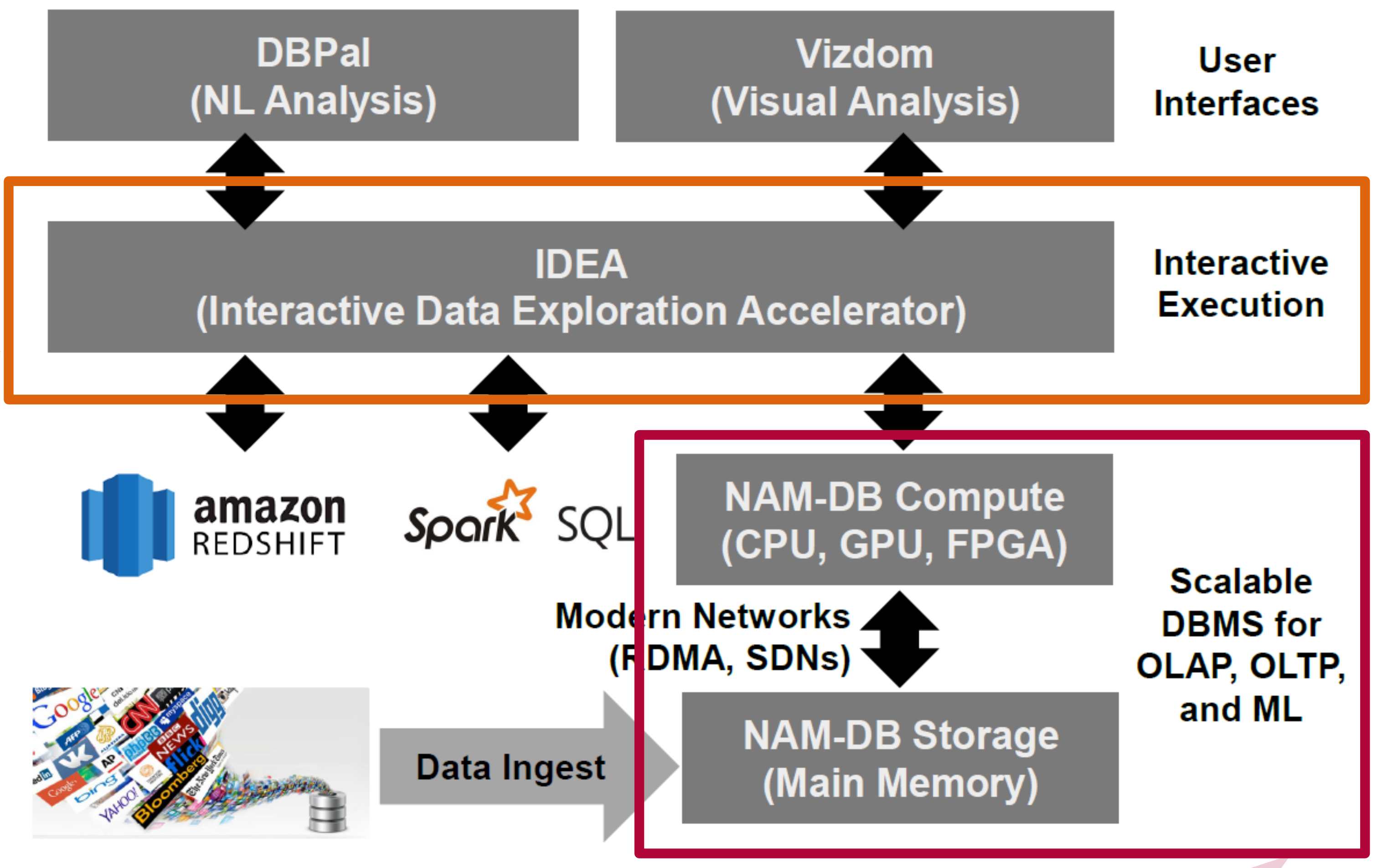
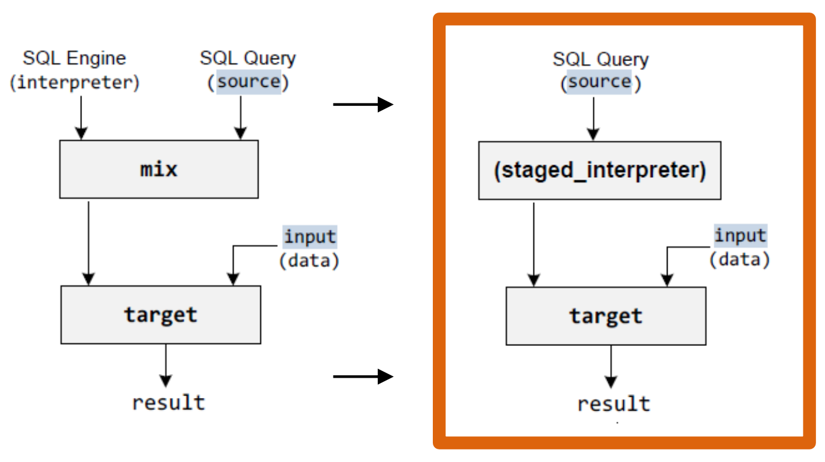


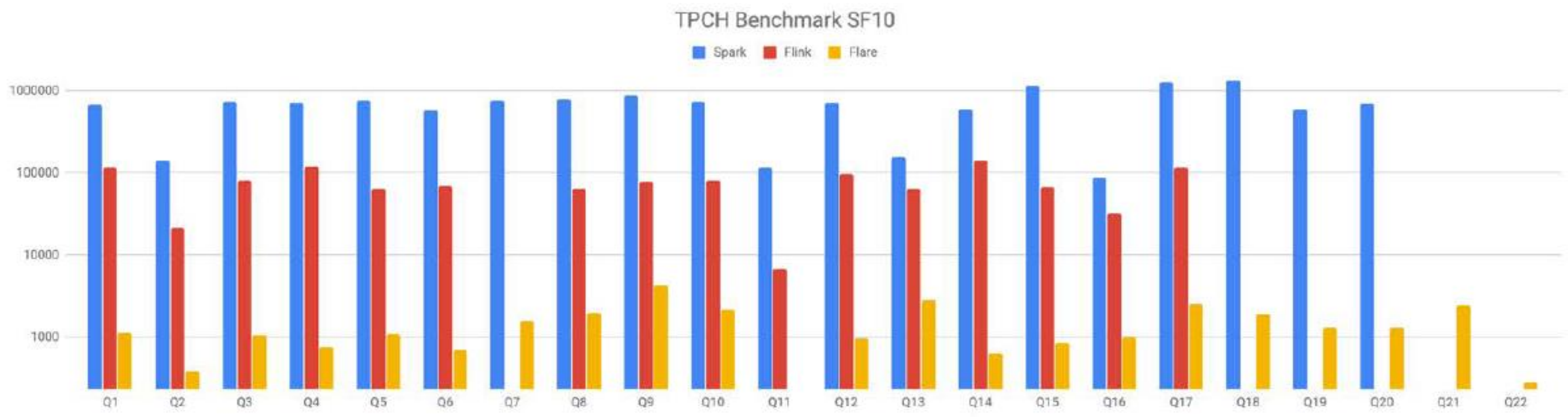
Image taken from the slides by Carsten Binnig,
Lecture "Towards Interactive Data Analytics", 05.11.2019

- utilizes AQP – queries only partial data to achieve response time
- probabilistic storage (stores variables) to use for later queries
- applys bayes theorem if possible to reduce number of queries

- **Flare** (drop-in accelerator for Spark) presented by **Tiark Rompf** in his Talk "A PL & Compiler view on Data Management and Machine Learning Systems"
- light-weight modular staging (LMS) to build "**staged interpreters**", which are query compilers



- beats plain Spark and Flink in TCPH SF10 convincingly



Images taken from the slides by Tiark Rompf,
Lecture "A PL & Compiler view on Data Management and Machine Learning Systems", 07.01.2020

Proposition

- Applying Query Compilation in Scalable DBs
- Potential Benefits:

1. enable an easy high-level interface while maintaining performance of low-level code
2. generated low-level code promises higher performance when executing look-ups
3. larger chunks can be looked at by AQP component ➔ larger sample sets
4. enable interface for machine learning library (as shown with Lantern)

Goal

Having **efficient scalable databases** that are **AQP-enabled** (support partitioning into chunks of data) to enable interactive data analytics.

Solution

Increase query efficiency by building query compilers. This also enables low-level communication with Machine Learning libraries as demonstrated by Rompf.

Stefan Reschke
Masters student

Hasso Plattner Institute, Potsdam, Germany
Lecture Series on Practical Data Engineering

E-Mail: stefan.reschke@student.hpi.de

HPI Hasso Plattner Institut