Solutions for Efficient Data Management and Data Analysis

A Hands-on Performance Benchmark of DuckDB with Python Pandas and PostgreSQL

Student

Lucien Hagmann

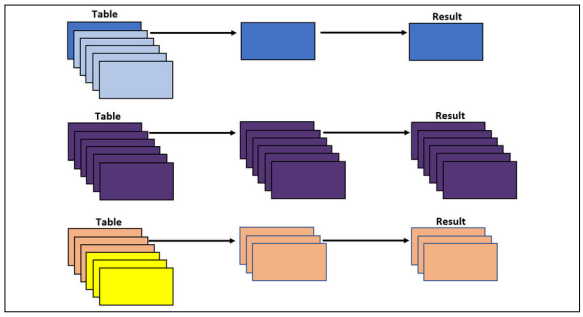
Initial Situation: In the realm of data-driven decision making, the evolution of Data Science tools, like Python Pandas, has brought with it rapid exploratory data analysis capabilities. However, these tools often fall short when it comes to a holistic approach to data management. These tools excel at in-memory operations, and data must first be exported from external sources. This limits their potential for handling massive datasets and long-term data management strategies. On the other hand, well-established data management systems such as PostgreSQL provide robust data storage and retrieval capabilities, but tend to sacrifice analytical processing efficiency. This paper addresses the challenge of balancing these aspects by introducing DuckDB, a promising solution that aims to bridge the gap between in-memory analytics and comprehensive data management. DuckDB offers the potential to combine the best of both (open source) worlds, addressing the limitations of existing tools while providing an opportunity for greater efficiency.

Approach: To investigate the claims of DuckDB's capabilities, this study delves into the technical underpinnings of various system architectures, alternative data storage structures, and programming paradigms relevant to data queries. These insights are then used to assess their impact on analytical query performance. An initial understanding of these technological differences is considered essential for the accurate interpretation of the results obtained in the practical part. The practical part of this study employs a comprehensive performance benchmark that includes two distinct scenarios. The first scenario focuses on speeding up data loading and exporting to meet the needs of data scientists who need to interact with data quickly. The second scenario emulates a TPC-H-like data warehouse setup, primarily evaluating Online Analytical Processing (OLAP) query execution times. The benchmark considers both the functional capabilities and performance aspects of DuckDB, contrasting them with Python Pandas library and PostgreSQL database.

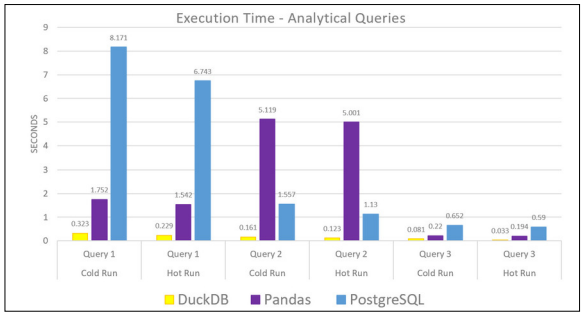
Result: The performance benchmark results demonstrate DuckDB's efficiency in addressing the dual challenges of data management and analytics. DuckDB exhibited remarkable performance in both scenarios, with much faster query execution times compared to Python Pandas and PostgreSQL. For instance, in-memory data loading and data conversion, was on average 50% faster with DuckDB than with Python Pandas. With regard to analytical queries involving different aggregates, DuckDB's execution time was approximately 30 times faster than PostgreSQL and 5 times faster than Pandas. JOIN and GROUP BY queries as well as the applied Window Function were solved 10 times faster than by PostgreSQL. Undoubtedly, the hands-on work with the three different data management solutions demonstrated the speed and flexibility of DuckDB, along with its seamless integration with data management tasks. However, the analysis also revealed a more nuanced perspective. In particular, DuckDB sometimes required more hardware resources than Python Pandas or PostgreSQL. Additionally, PostgreSQL still outperformed DuckDB by 50% when writing data to disk, highlighting the fact that DuckDB is designed as an OLAP database and does not compete with PostgreSQL in the OLTP domain. Despite these nuanced findings, DuckDB comes highly recommended as a data management solution due to its ease of setup, flexibility, and exceptional performance on analytical queries.

Execution Engines: Iterator model of PostgreSQL, materialization model of Pandas and Vector-Volcano model of

Own presentment



Measuring and comparing the execution time of analytical queries during the performance benchmark
Own presentment



The results, which are summarized in a rating table, show DuckDB as the leader in the comparison
Own presentment

Benchmark Aspects	DuckDB	Pandas	PostgreSQL
-Installation and Setup	++	++	+
-Integrability	++	+	-
-Simplicity in applying tool	++	-	++
-Graphical User Interface	-	-	++
Data Loading Performance	+	+	-
Data Converting and Exporting	++	+	-
SELECT Statement (data read)	++	-	+
UPDATE Statement (data write)	+	--	++
Indexation	+	-	+
AVG/SUM Query	++	+	-
GROUPY BY Query	++	-	+
JOINS	++	-	+

Advisor
Prof. Stefan F. Keller

Subject Area
Computer Science, Data Science