Cross Island Join Query and Optimization

Sadanand Kolhe, Gerald Balekaki
skolhe1@hawk.iit.edu, gerald.balekaki@iit.edu. github.com/sadannd/Cross-Island-Join-Query
Computer Science, Illinois Institute of Technology

Motivation/Abstract

- In contemporary operations, many organizations utilize various systems. Integrating data from these systems is vital for data analysts and database administrators, improving product functionality and reducing data management costs.
- While established methodologies facilitate the integration of data within a singular database, challenges arise when confronted with the integration of two disparate systems. Consequently, Motivated by the Bigdawg Framework, we have devised an innovative approach to seamlessly unite data sourced from two distinct database systems. This solution seeks to improve efficiency and reduce costs in managing data from diverse systems.

Description/Introduction

- In today's world of handling data, combining information from different databases is a substantial challenge [1]. This research takes a critical stride in addressing this challenge by centering its focus on the fundamental issue of seamlessly integrating data.
- As we delve into the nuances of optimizing cross-database data integration, our research endeavors to bridge the existing gap by aiming to minimize temporal disparities, thereby paving the way for enhanced efficiency and expeditious data integration processes.

Architecture of Polystore

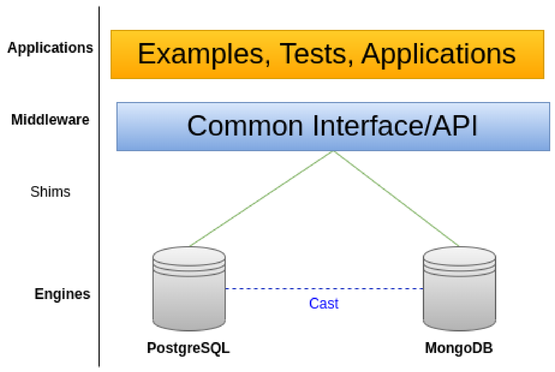

Figure 1:Polystore architecture based on Bigdawg [2]

What we did/Methodology

- In the workflow of the join query, the primary phase consisted of retrieving data from PSQL. Subsequently, data retrieval from MongoDB transpired as the second stage. Post data acquisition, a detailed cleansing process ensued, and the refined datasets were methodically stored in Hashmaps, with due consideration given to pertinent indexes for the purpose of ensuring proficient data organization.
- In the subsequent step, an iterative comparison was executed by sequentially traversing one Hashmap over another, pinpointing matching pairs. To enhance the analysis, original entries associated with these matched pairs were subsequently retrieved from initially stored Hashmaps. Importantly, the entire process was simulated for multiple users, assessing the scalability and effectiveness of the proposed methodology.

Psuedo Workflow

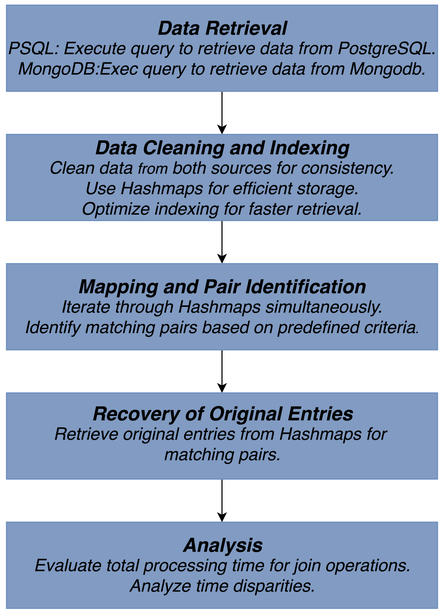

Figure 2:Workflow

Results

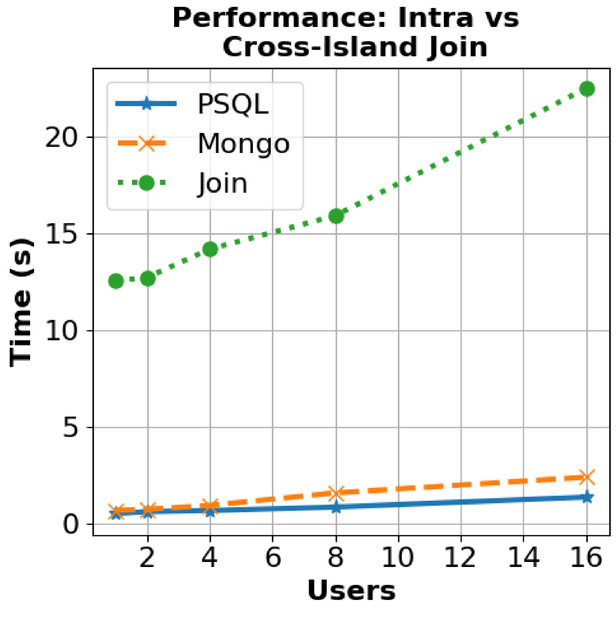

Figure 3:Results

Future Scope

- **Optimization Techniques:** Explore advanced optimization techniques to further minimize the time difference.
- **Support for Additional Database Systems:** Include popular databases such as SciDB, Accumulo, Neo4j, Cassandra catering to users with diverse data storage preferences.
- **Scalability Management:** Address scalability challenges that may arise as the system encounters an increasing number of users, larger datasets, or higher concurrent requests.

Conclusion

In conclusion, research introduces a systematic methodology for joining data from PSQL and MongoDB, addressing challenges in cross-database integration. The approach ensures precision and consistency, proving effective in simulated scenarios with multiple users. Furthermore, our findings demonstrate the inherent expense of joins across disparate storage systems.

References

[1] Ran Tan, Rada Chirkova, Vijay Gadepally, and Timothy G Mattson.
Enabling query processing across heterogeneous data models: A survey.
In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3211–3220. IEEE, 2017.

[2] Jennie Duggan, Aaron J Elmore, Michael Stonebraker, Magda Balazinska, Bill Howe, Jeremy Kepner, Sam Madden, David Maier, Tim Mattson, and Stan Zdonik.
The bigdawg polystore system.
ACM Sigmod Record, 44(2):11–16, 2015.