



Burroughs

BRIDGING THE GAP BETWEEN KAFKA AND SQL

Ainesh Sootha, Andrew Riordan, Aneesh Chakravarthula, Erika Ergart, Jason Cao, Mihira Krishnaswamy, Vandana Chari, Wyatt Klueber

Burroughs Summary

Burroughs wraps around KsqlDB, Kafka, and Postgres and is capable of:

- Translating a SQL query into a sequence of KSQL statements
- Maintaining a real time link between the KsqlDB table and the RDBMS table.

Large chunk of SQL supported:

- Joins and self-joins
- Common table expressions
- Special functions (group_concat)
- Limit clause

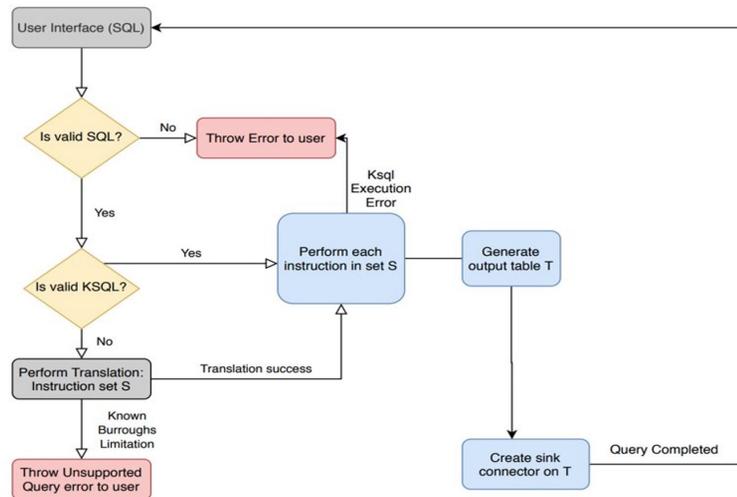
Other Features:

- Embedded producers
- Query execution tracing
- Topic deletion

Conclusions

Lessons Learned

We have learned that Kafka is a widely used real-time event streaming platform. However, Kafka has its limitations and differences. The differences between kSQL and regular SQL is one of the biggest limitations that Burroughs tries to solve. However, making sure that the SQL queries translates to the equivalent kSQL query involves thinking about utilizing a lot of intermediate kSQL commands to support more SQL commands, and accounting for the limitations in the Kafka infrastructure.



Future Goals

Building a complete product

The future goals of Burroughs are based around the primary goal of being able to use SQL on real-time data. Since we have a robust preliminary application, the future goals involve adding features and constantly updating the app to support the needs of the users.

Some future goals include:

- Adding support for complex queries like cascading aggregations
- Making Burroughs completely portable, thus allowing it to attach to any existing Kafka system
- This would also allow Burroughs to be used outside of just the Viasat use cases.
- Update Web GUI to support existing features (that are already supported by the CLI)
- Make Burroughs easily compatible with tools like Tableau

References/Acknowledgements

Special thanks to Mark Rapo, Gideon Shalev, Brian Burns, and the entire staff from Viasat for mentoring us this year! We would also like to thank Dr. Mark Daniel Ward, Ellen Gundlach, and Maggie Ann Betz for supporting us and providing resources to help advance this project.

- Kleppmann, M. (2017). *Designing data-intensive applications*. Beijing: O'Reilly.
- Nickoloff, J. (2019). *Docker in Action*. Manning Publications Company.
- Narkhede, N., Shapira, G., & Palino, T. (2017). *Kafka: The definitive guide: Real-time data and stream processing at scale*. Beijing: O'Reilly.
- Apache Kafka & Event Streaming Platform for the Enterprise. (n.d.). Retrieved November 22, 2020, from <https://www.confluent.io/>
- KsqlDB. (n.d.). Retrieved November 22, 2020, from <https://ksqldb.io/>
- Apache Calcite API. (n.d.). Retrieved March 21, 2021, from <https://calcite.apache.org/javadocAggregate/index.html>
- 84.51 The Complete Journey 2.0: <https://www.8451.com/area51>

