Confidential - Do Not Share

Arrow FlightSQL: A 20x Faster Alternative to ODBC and JDBC

Presented by James Duong of Bit Quill Technologies



Overview

- Introduction to Arrow Flight
- FlightSQL Enhancements for Arrow Flight
- JDBC Driver for FlightSQL

Introduction to Arrow Flight

Introduction to Apache Arrow

- A columnar, in-memory data format and supporting libraries.
- Supported on many languages including C++, Java, Python, Go
- Data is strongly typed. Each row has the same schema.
- Includes libraries for working with the format:
 - Computation engine utilizing SIMD operations for vectorized data analysis.
 - Interprocess communication.
 - Serialization / deserialization from file formats.
- Fully open source with a permissive license.

Arrow Adoption

Arrow powers dozens of open source & commercial technologies



10+ programming languages supported



Java, C, C++, Python, R, JavaScript, C#, Ruby, Rust, Go, ...

Apache Arrow Adoption

Apache Arrow Adoption



Why is Arrow Flight needed?

• An open protocol that the community can support.

• Designed for data in the modern world

- Older protocols (ODBC/JDBC) are row oriented and geared towards large numbers of columns and low numbers of rows.
- Arrow's columnar format is oriented towards high compressibility and large numbers of rows.

• Supports distributed computing as a client-side concept:

- A data request can return multiple endpoints to a client.
- The client can retrieve from each endpoint in parallel.

Arrow Flight

• Protocol for serialization-free transport of Arrow data

• This is particularly efficient if the client application will just work with Arrow data directly.



Distributed Computing: Single Node with Arrow Flight



Endpoint = {location, ticket}

Distributed Computing: Multiple Nodes with Arrow Flight



Arrow Flight as a Development Framework

• Includes a fully-built client library

• Includes a high-performance, scalable server

- Built on top of Google's gRPC technology and compatible with existing tooling.
- Server implementation details such as thread-pooling, asynchronous IO, request cancellation are already implemented!
- Server deployment is a matter of implementing a few RPC request handlers.

FlightSQL Extensions for Arrow Flight

Why extend Arrow Flight? It is generic by design

• Client sends a byte stream, server sends a result

- The content of the byte stream is opaque in the interface.
- It only has meaning for a particular server.
- Example Dremio interprets the byte stream to be a UTF-8 encoded SQL query string.
- Catalog information is not part of Arrow Flight's design
 - There is no RPC call to to describe how to build the byte stream the client sends.
 - Generic tools cannot be built.
- Flight is meant to serve any tabular data, not databases in particular.
- ODBC and JDBC standardize query execution and catalog access.
- Enter FlightSQL

What is FlightSQL?

• Initiative to allow databases to use Arrow Flight as the transport protocol

• Leverage the performance of Arrow and Flight for database access.

• Extended set of RPC calls to standardize a SQL interface on Flight:

- Query execution
- Prepared statements
- Database catalog metadata (tables, columns, data types).
- SQL syntax capabilities

• Generic client libraries

 A FlightSQL client application can be used against any Flight SQL server without code changes.

Common Tool Workflow



Retrieving query data

FlightSQL vs. JDBC

JDBC

- Each database vendor must implement, maintain, and distribute a driver.
- Each database vendor must implement their entire server.
- Implementation details may be closed source.
- Protocol is proprietary.

FlightSQL

- Single client that works against any FlightSQL server.
- Server implementation is part of Flight. Only RPC handlers need to be implemented.
- Flight and Arrow components are open and the community is actively improving them.
- Protocol is open and integrates with gRPC and Arrow tooling.

FlightSQL Status

• Initial version released with Arrow 7.0.0!

- Includes support for C++ and Java clients and servers
- Enhancements to column and data type metadata are under review

• Open for contributions:

- Support for additional languages (Python, Go, C#, etc.)
- More SQL features, such as transactions.

Arrow Flight SQL JDBC Driver

Why build a JDBC Driver?

• FlightSQL will take time to be adopted.

- Many BI tools already support JDBC. This provides a fast way to allow for access from these tools to FlightSQL servers.
- A driver proves that FlightSQL provides enough SQL functionality.
 - The Arrow JDBC driver was built in *parallel* with the FlightSQL libraries and protocol.
 - BI tools were tested against the Arrow driver, which in turn verifies if FlightSQL has the capabilities required to support these tools.
- Note that it is still preferable to have native FlightSQL applications to better harness new features such as multiple endpoints.

Arrow Flight SQL JDBC Driver

• A JDBC Driver built on top of FlightSqlClient libraries

- A single driver to connect to any FlightSQL server, regardless of how the server was implemented.
- Supports arbitrary server-side options as connection properties.
- State is transmitted using HTTP cookies.
- Completely open source and to be released under the Apache license.
- Functionally complete. A pull request is available under the Arrow project.
- The driver will work with any JDBC tool without code changes.

JDBC User Experience

• How does the user experience change with a single driver that works against unlimited databases?

Before Arrow JDBC

- User must download a driver for each database they want to work with.
- In the case of Tableau, there are 90+ connectors!

After Arrow JDBC

• User can download the Arrow JDBC driver and work with any database supporting FlightSQL.

Tableau Demo

Credits

- FlightSQL was built from contributions from:
 - Bit Quill Technologies
 - Dremio
 - Symbiose Ventures
 - Voltron Data



- Arrow Flight SQL Announcement: <u>https://arrow.apache.org/blog/2022/02/16/introducing-arrow-flight-sql/</u>
- Arrow Flight SQL JDBC Driver PR: <u>https://github.com/apache/arrow/pull/12254</u>

Thank You!