

Benchmarking Apache Arrow Flight

A wire-speed protocol for data transfer, querying and microservices

Tanveer Ahmad

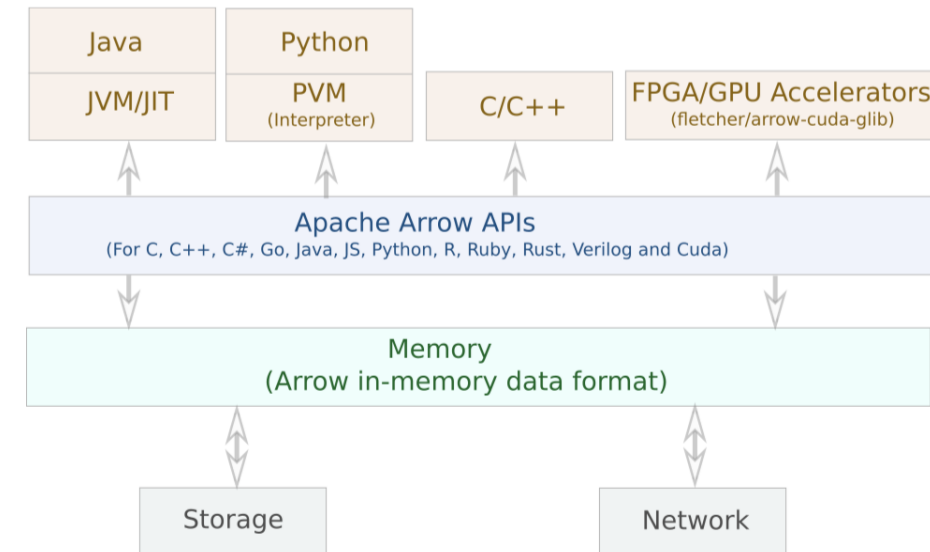
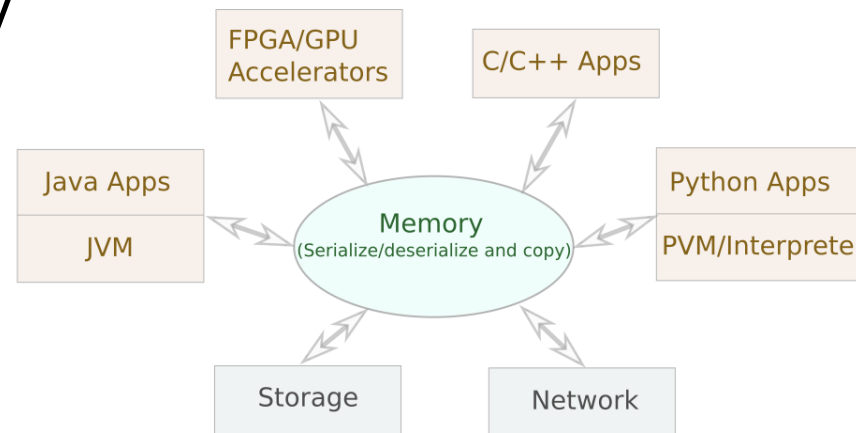
TU Delft, The Netherlands

Overview

- Apache Arrow
- Arrow Flight
- Data Transfer Benchmarks
- Use Cases
 - Transactional databases and columnar store
 - Query Subsystems
 - Microservices Integration
- Future Outlook

Apache Arrow

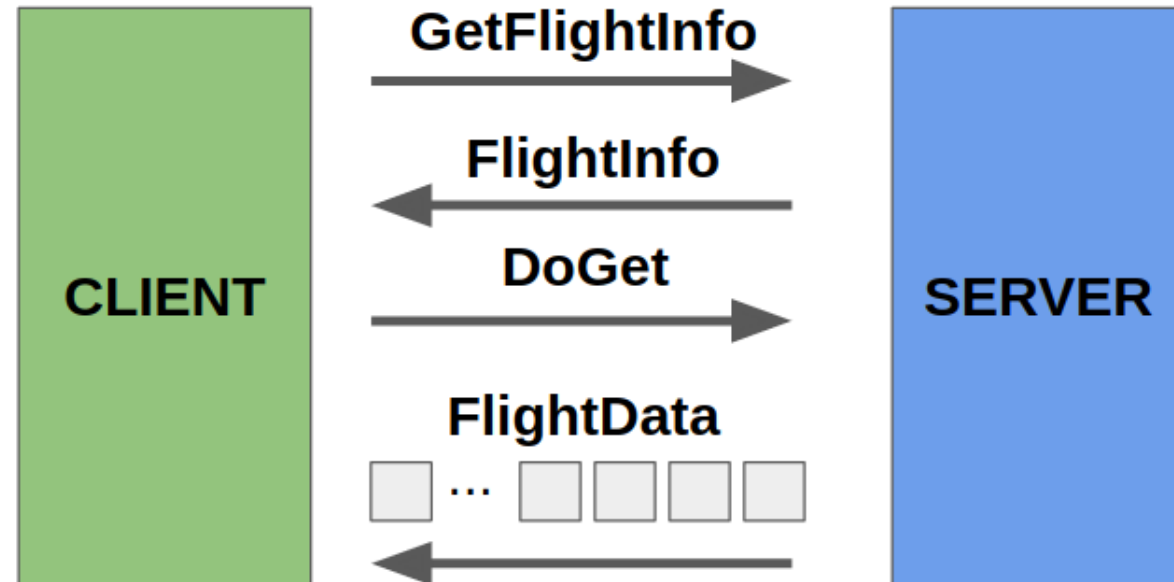
- Language-agnostic in-memory columnar data structures
- Random access
- Maximum throughput
 - Vectorized processing/SIMD
 - Pipelining
 - cache locality
- GPU/FPGA



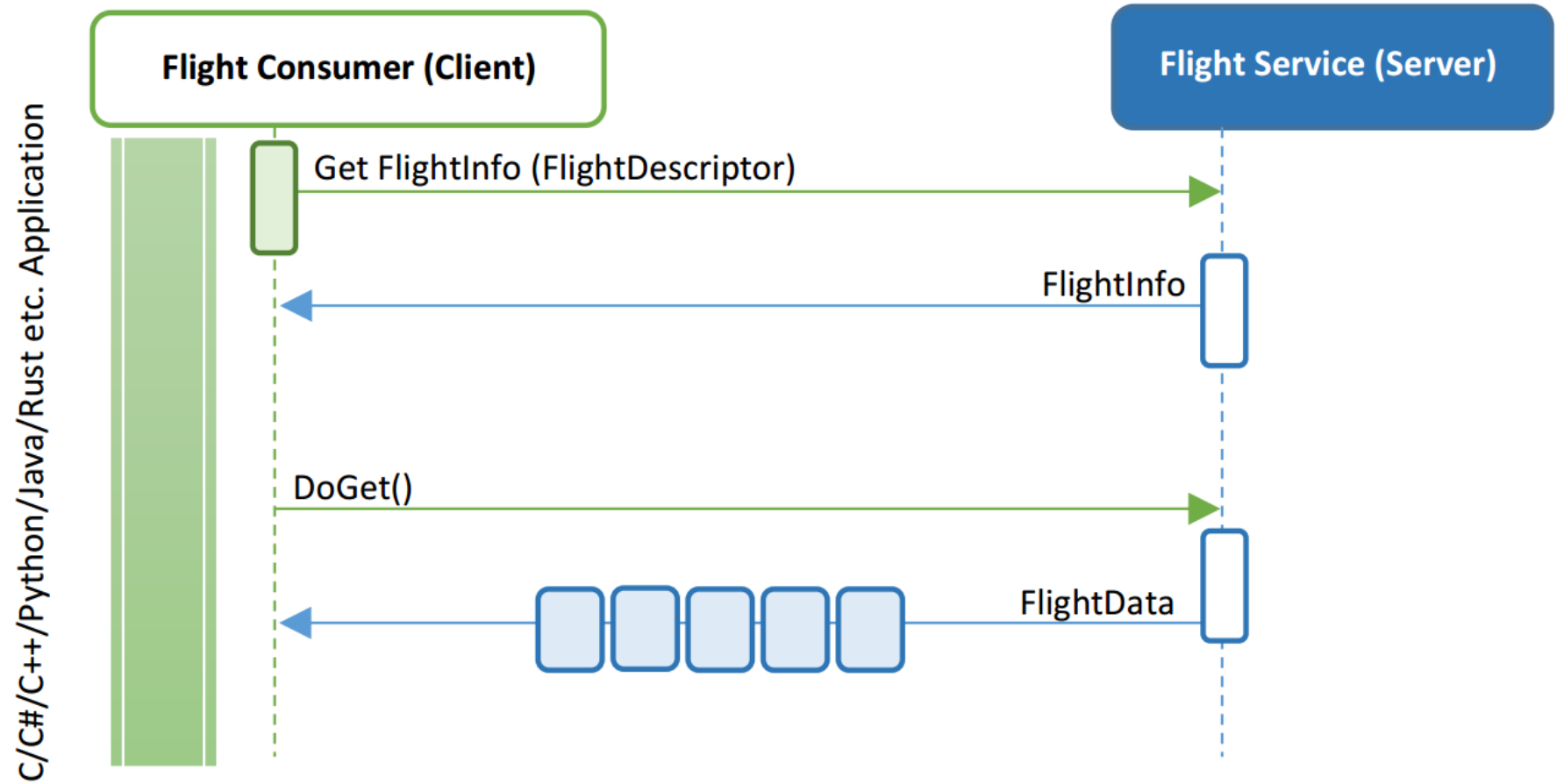
Arrow Flight

- High performance wire protocol for bulk Arrow data transfer
- Cross-platform
- Parallel
- Security

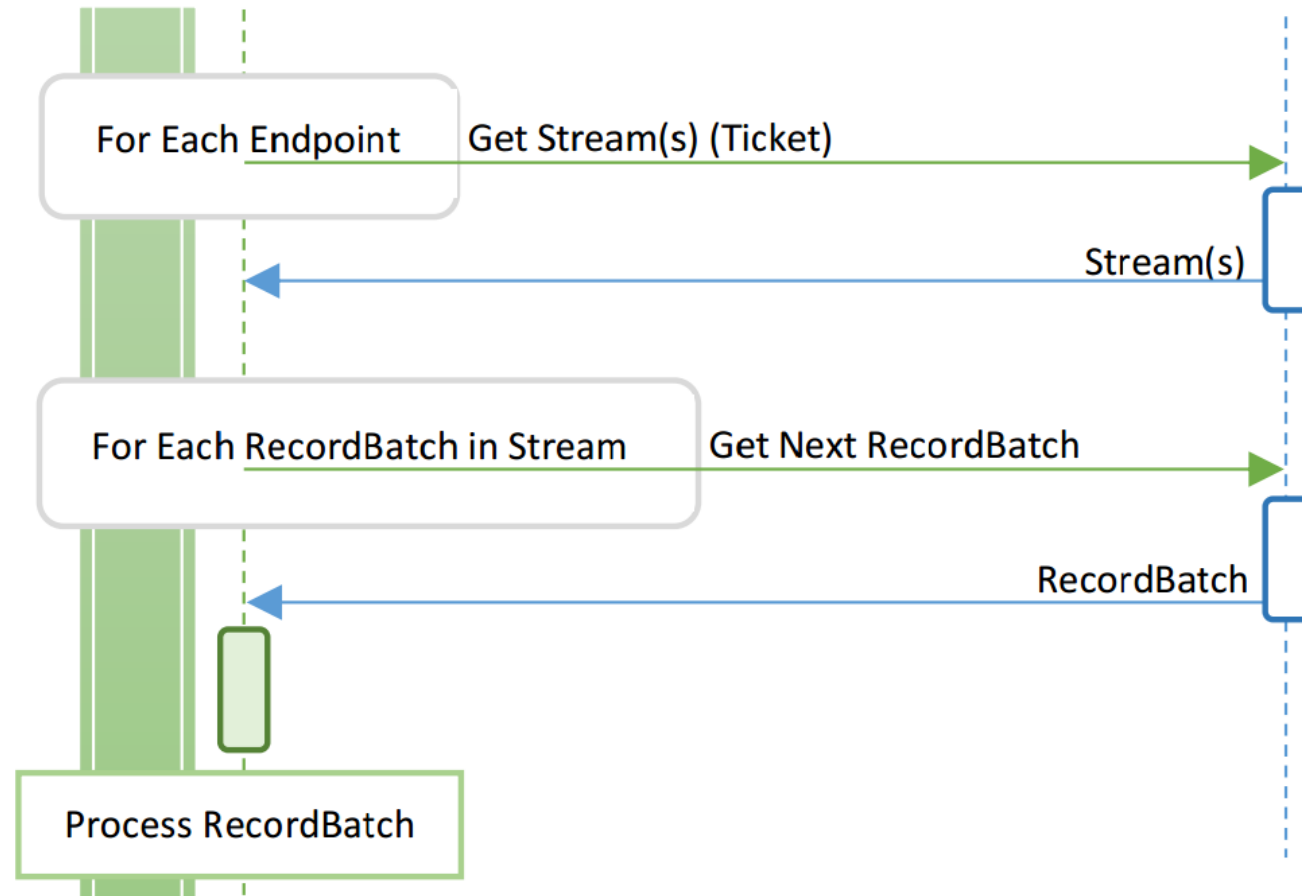
Simple Client-Server Execution Flow



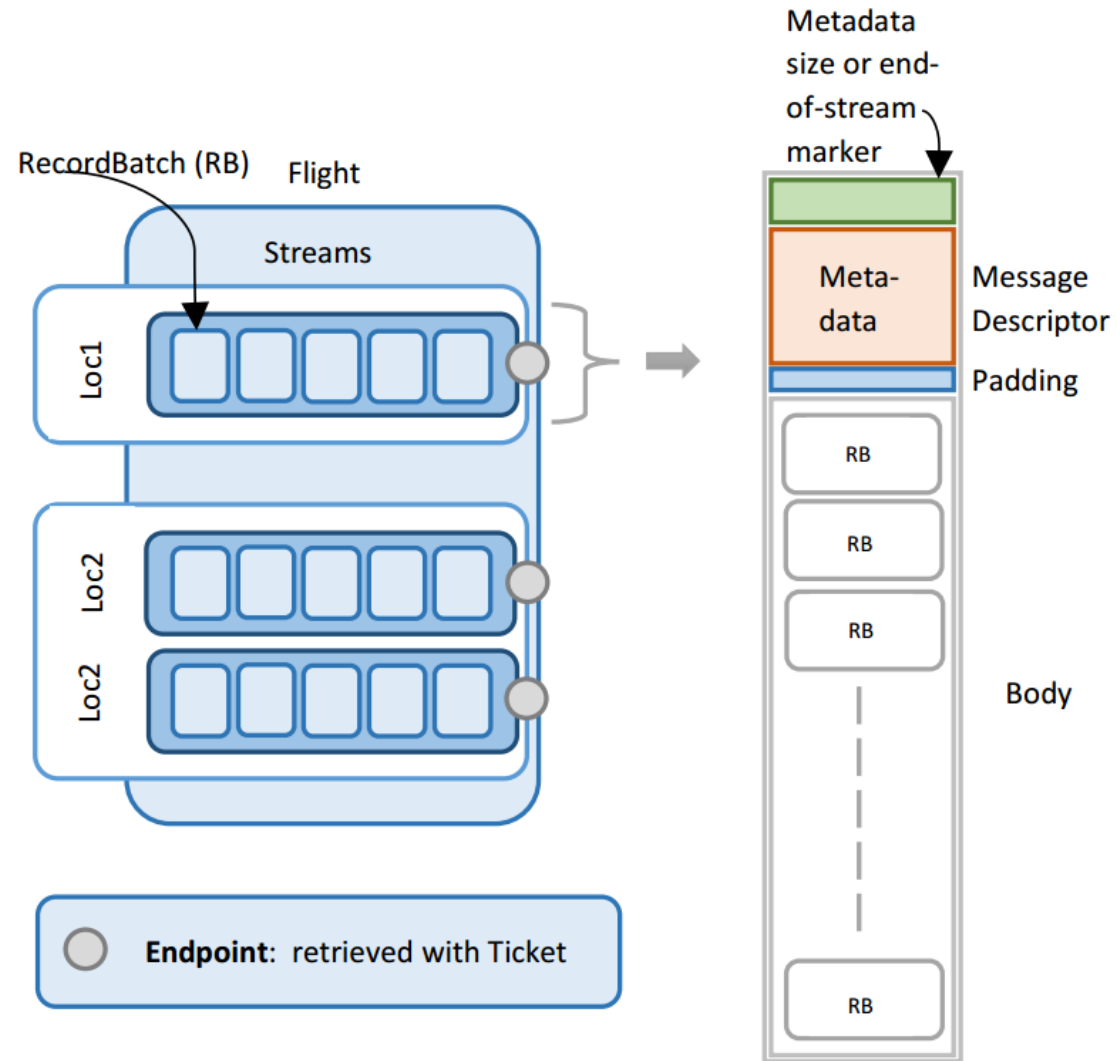
Arrow Flight



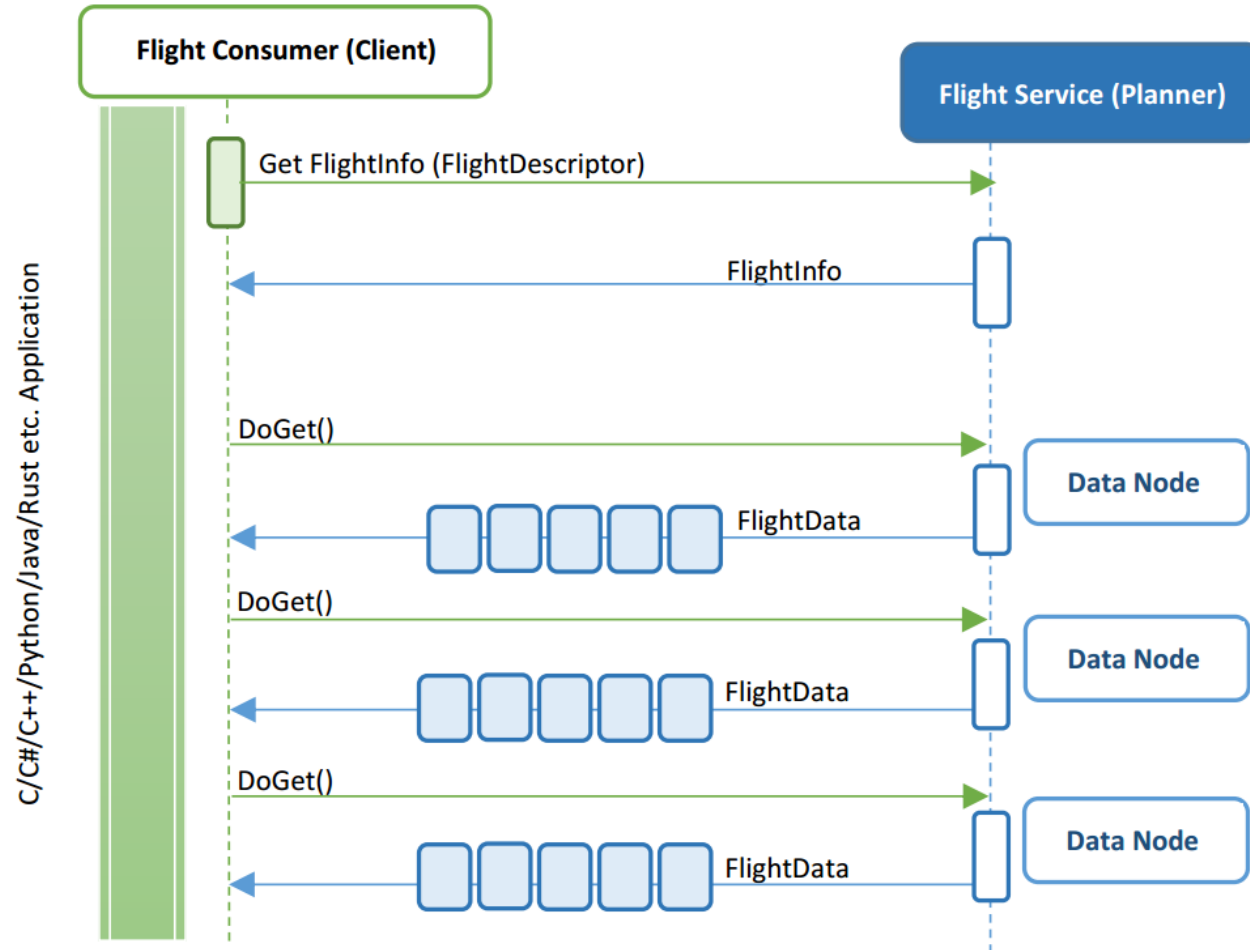
Arrow Flight



Arrow Flight

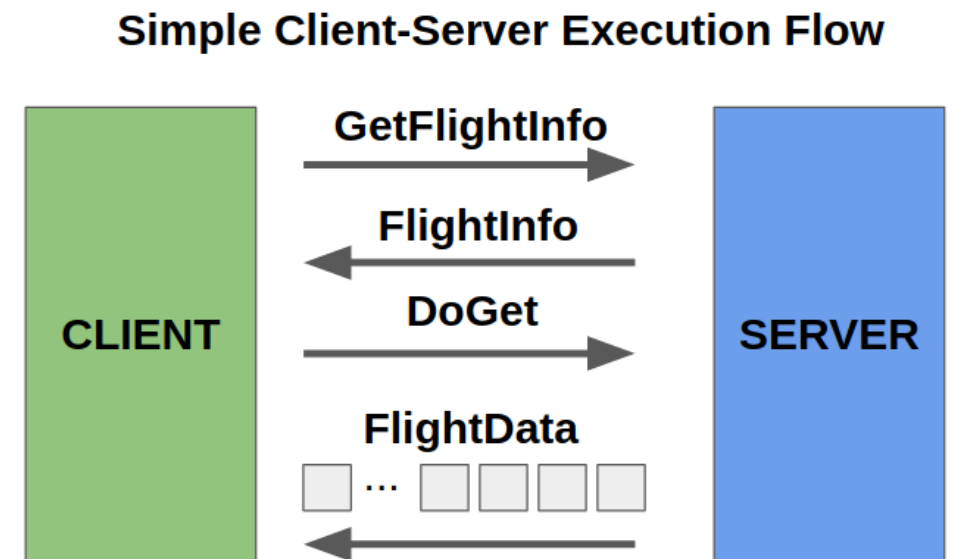


Arrow Flight



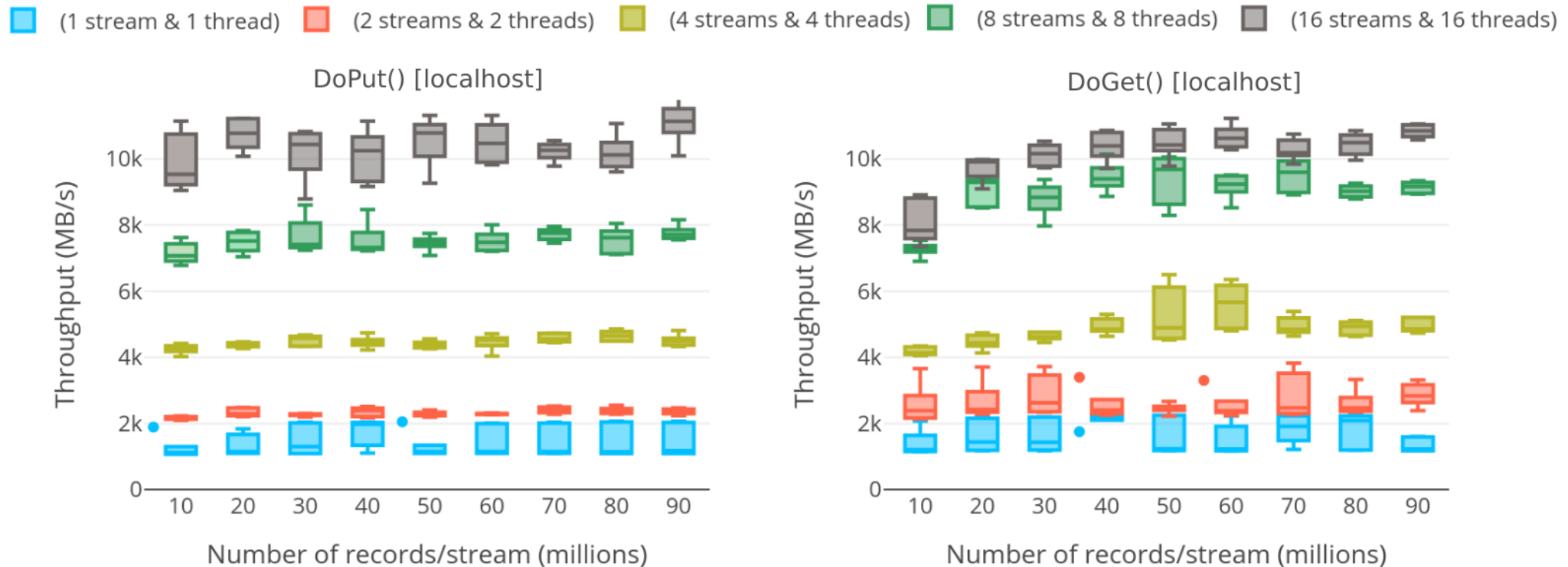
Data Transfer Benchmarks

- Localhost and client-server
- 16 physical cores (32 hyper-threading jobs)
- InfiniBand adapter 4 × FDR (56 Gbit/s inter-node bandwidth)
- 1, 2, 4, 8 and 16 streams in parallel
- Each stream having 10-90 million records



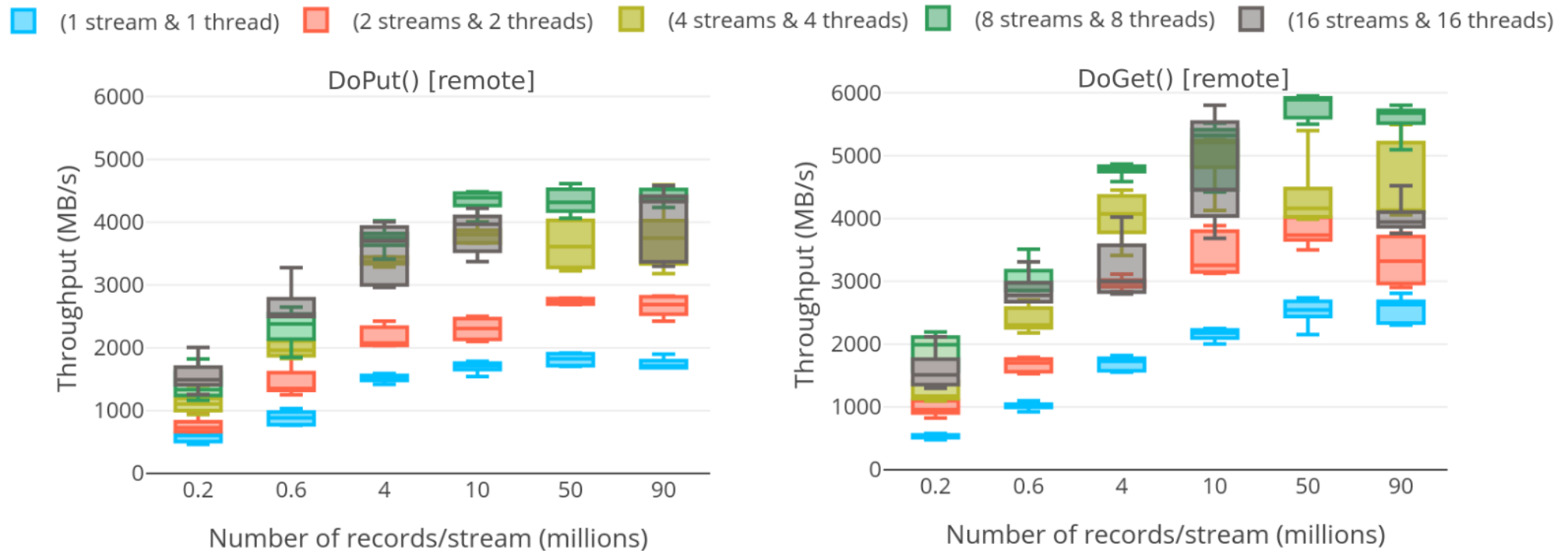
Data Transfer Benchmarks

- Localhost
- 1GB/s for single stream up to 10GB/s for 16 streams



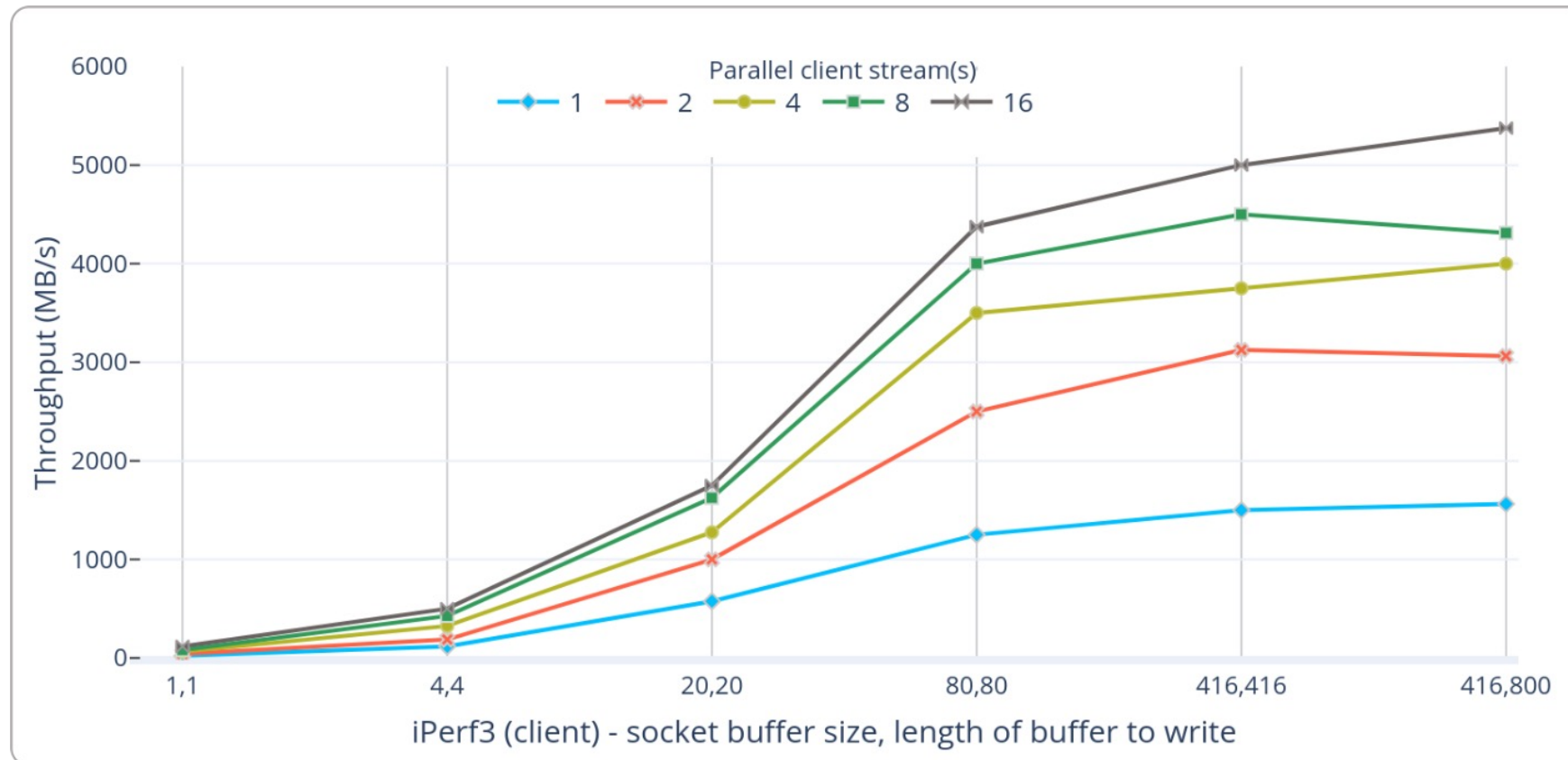
Data Transfer Benchmarks

- Remote client-server
- 6000 MB/s throughput for DoGet()
- 4800 MB/s throughput DoPut()



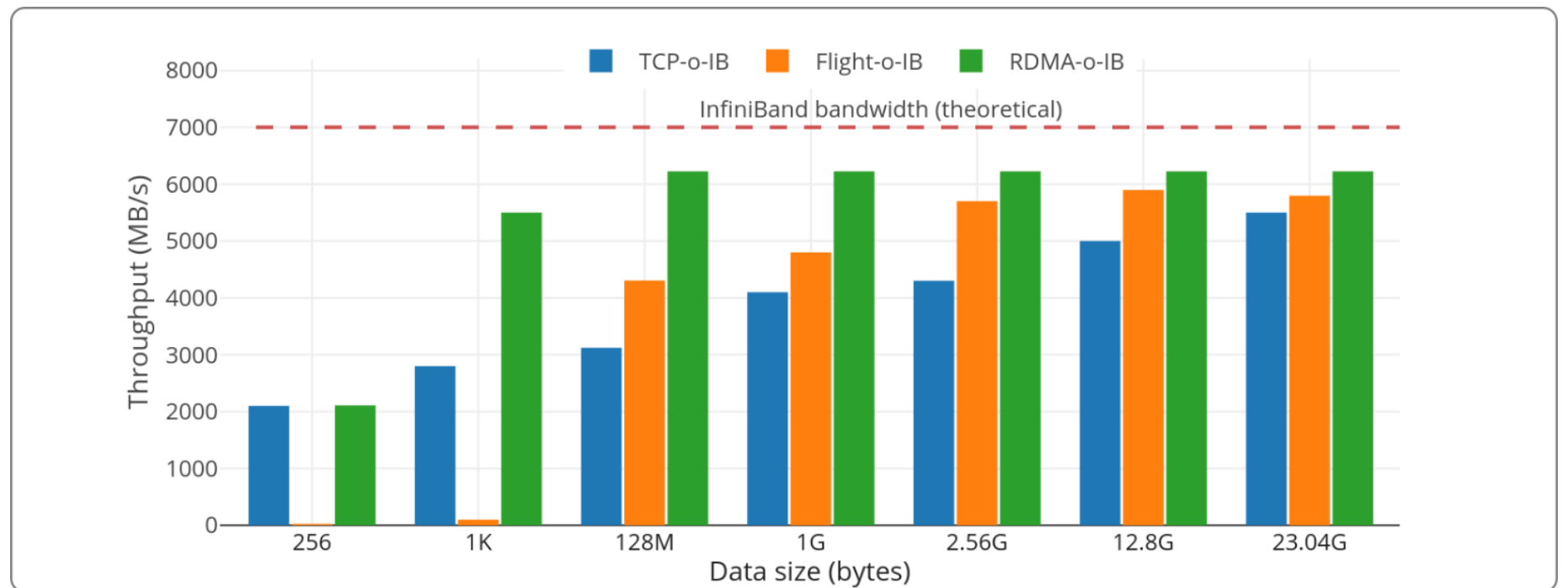
Data Transfer Benchmarks

- TCP performance on Infiniband
- 1700 MB/s (1 stream) – 5300 MB/s (16 streams)



Data Transfer Benchmarks

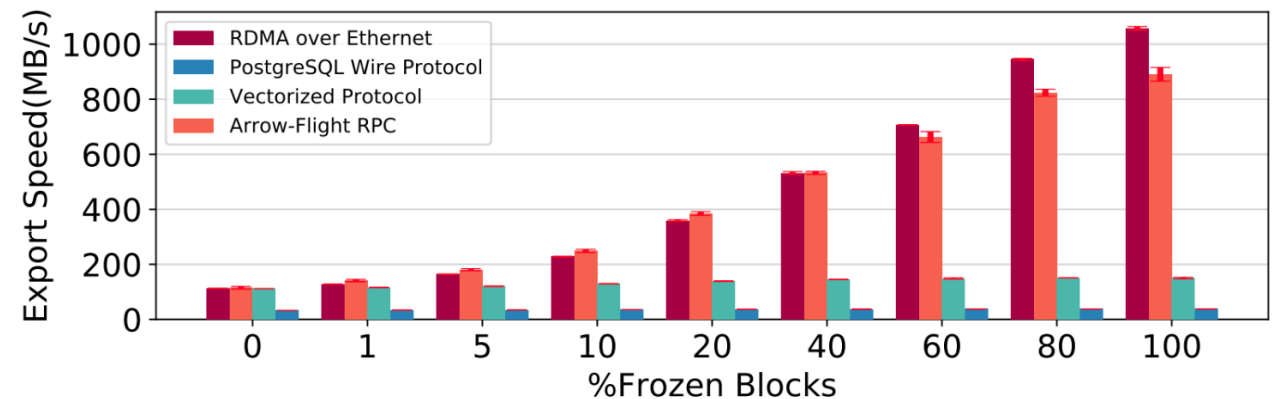
- TCP (iPerf3) -> 5.4 GB/s
- RDMA (ib verbs) -> 6.2GB/s (close to the theoretical max of 7GB/s)
- Arrow Flight -> 5.9 GB/s (95% of the RDMA bandwidth or more than 80% of the maximum achievable bandwidth)



USE CASE-I

Transactional data analytics

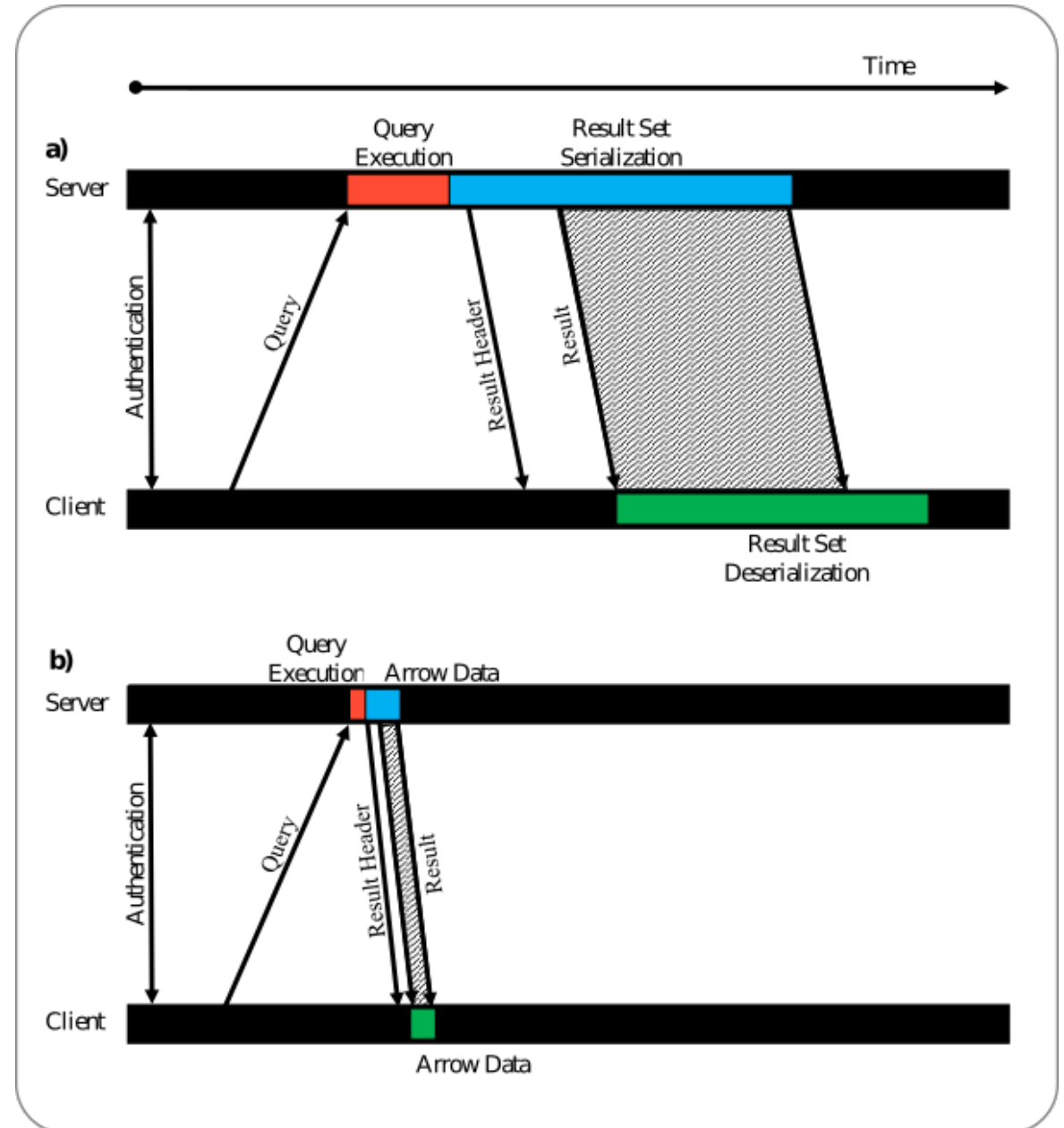
- Comparing four different export methods,
 - Client-side RDMA
 - Arrow Flight RPC
 - Vectorized wire protocol
 - Row-based PostgreSQL wire protocol.
- Flight uses up to 80% of the available bandwidth



USE CASE-II

Query Subsystems

- a) Without Flight
- b) With Flight

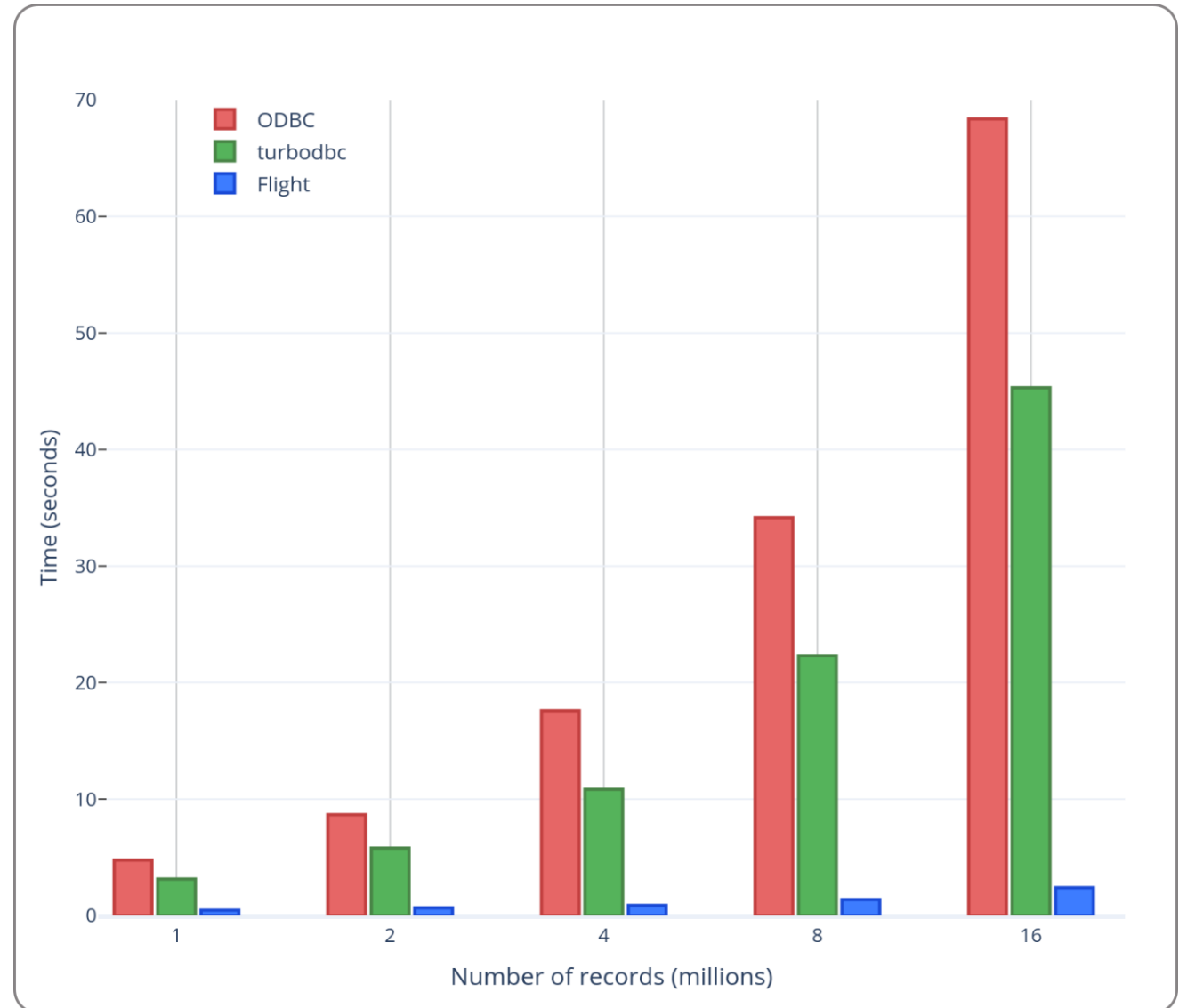


Query Subsystems

- Dremio – ODBC
- Dremio – turbodbc
- Dremio – Flight
- Data-Fusion – Flight
- Apache Arrow - FlightSQL

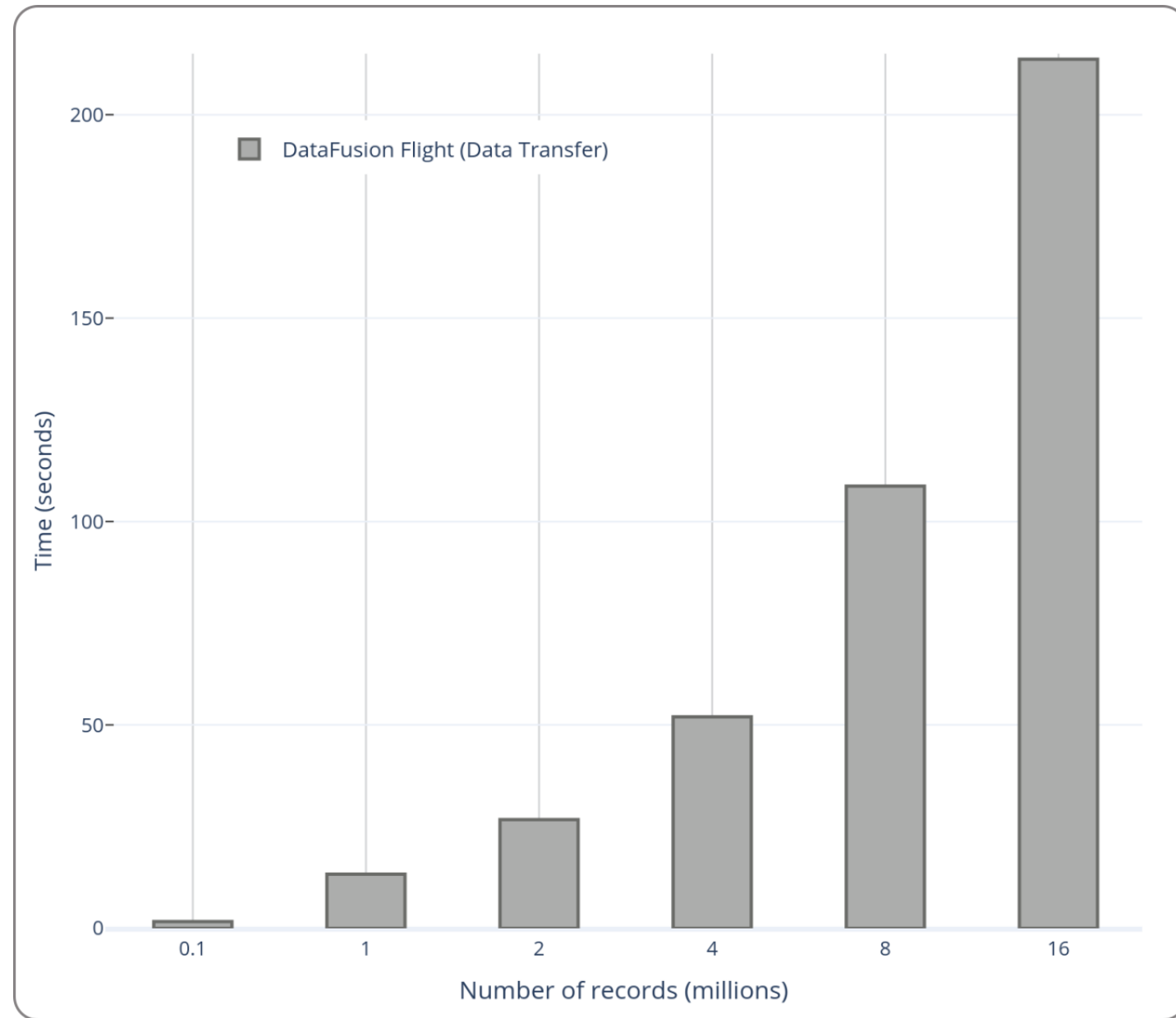
Query Subsystems

- Dremio – ODBC
- Dremio – turbodbc
- Dremio – Flight



Query Subsystems

- Dremio – ODBC
- Dremio – turbodbc
- Dremio – Flight
- Data-Fusion – Flight



Query Subsystems

- Dremio – ODBC
- Dremio – turbodbc
- Dremio – Flight
- Data-Fusion – Flight
- Apache Arrow - FlightSQL

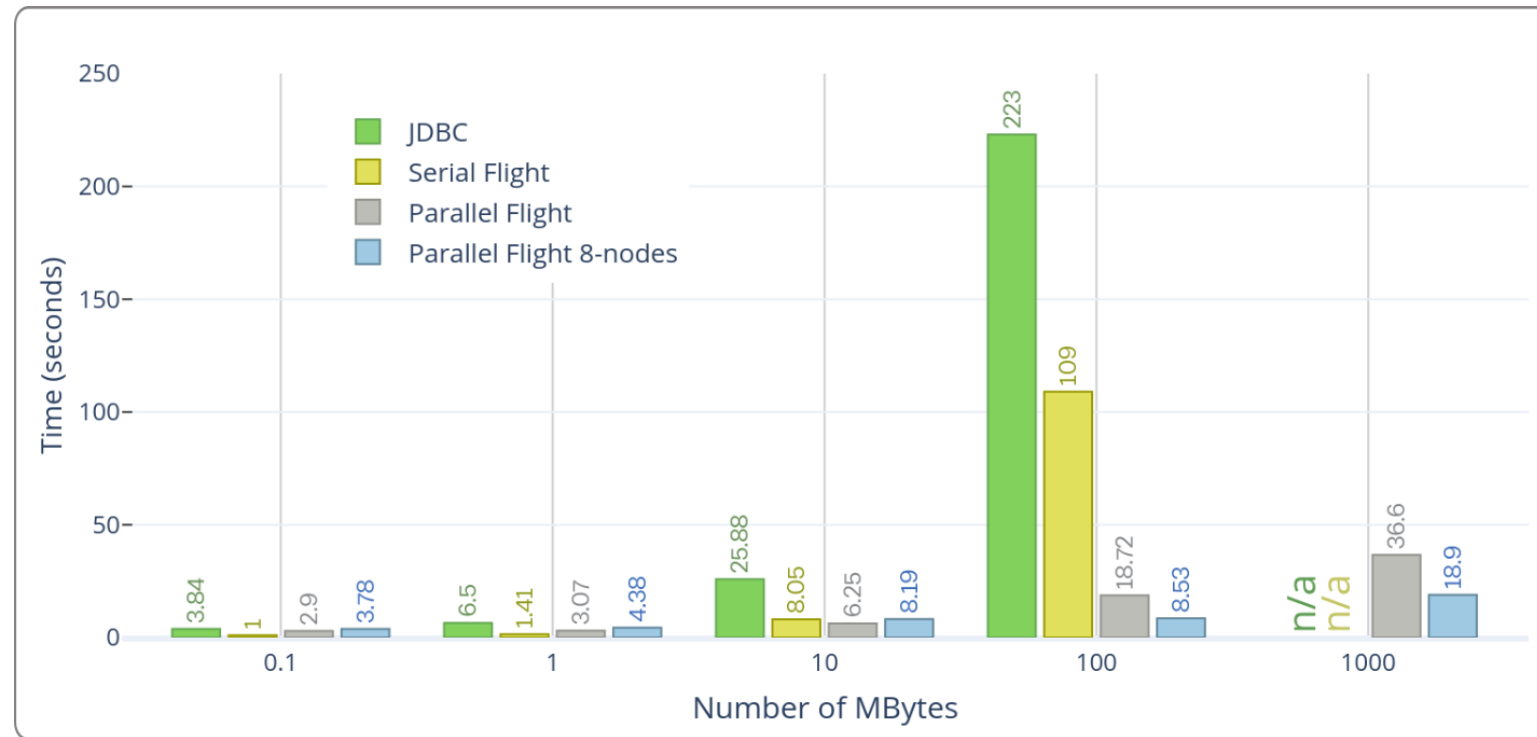
USE CASE-III

Microservices Integration

- Flight Data Microservice - Apache Spark
- Flight Data Microservice - Apache Spark/TensorFlow Clients
- XGBatch - Pandas/Dask
- FlightGrid/PyGrid - AI Models Training
- The Mesh for Data platform - Arrow/Flight module
- Magpie - A Pandas based unified data processing engine

Microservices Integration

- Flight Data Microservice - Apache Spark

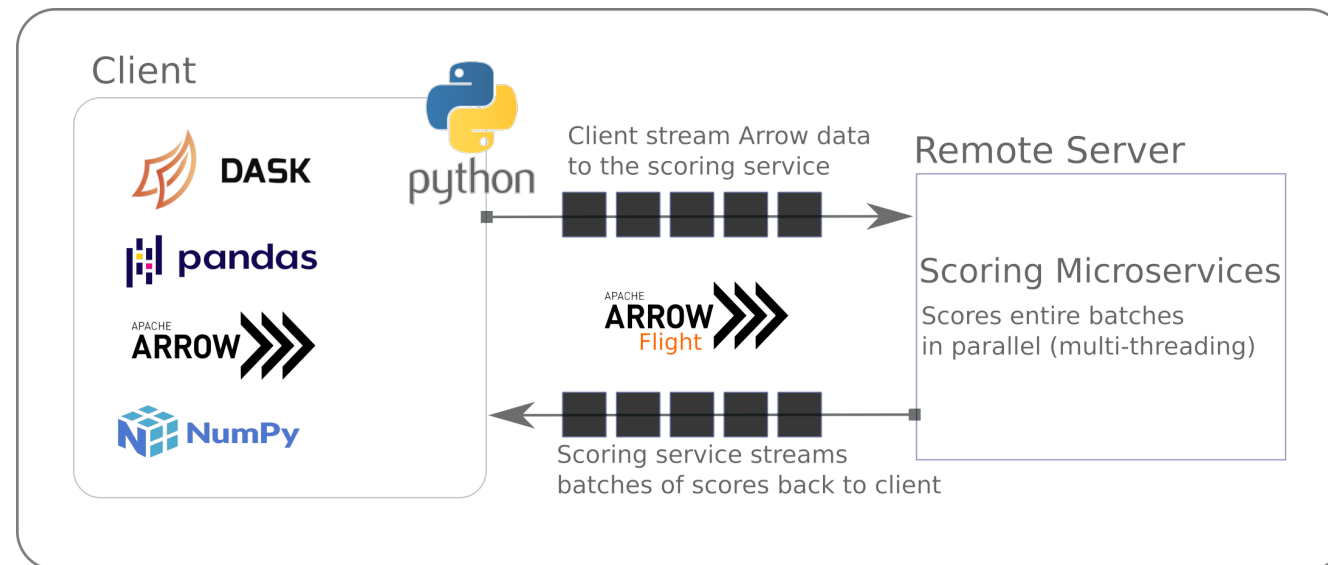


Microservices Integration

- Flight Data Microservice - Apache Spark
- Flight Data Microservice - Apache Spark/TensorFlow Clients

Microservices Integration

- Flight Data Microservice - Apache Spark
- Flight Data Microservice - Apache Spark/TensorFlow Clients
- XGBatch - Pandas/Dask



Microservices Integration

- Flight Data Microservice - Apache Spark
- Flight Data Microservice - Apache Spark/TensorFlow Clients
- XGBatch - Pandas/Dask
- FlightGrid/PyGrid - AI Models Training

Microservices Integration

- Flight Data Microservice - Apache Spark
- Flight Data Microservice - Apache Spark/TensorFlow Clients
- XGBatch - Pandas/Dask
- FlightGrid/PyGrid - AI Models Training
- The Mesh for Data platform - Arrow/Flight module

Microservices Integration

- Flight Data Microservice - Apache Spark
- Flight Data Microservice - Apache Spark/TensorFlow Clients
- XGBatch - Pandas/Dask
- FlightGrid/PyGrid - AI Models Training
- The Mesh for Data platform - Arrow/Flight module
- Magpie - A Pandas based unified data processing engine

Future Outlook

- Arrow Flight on RDMA
- FlightSQL
- Transactional and analytical workloads

Thanks!