

# Ruby + ADBC

A single API between Ruby and DBs

Sutou Kouhei

*ClearCode Inc.*

*RubyKaigi 2023*

*2023-05-13*

# Sutou Kouhei

## A president/Ruby committer

The president of ClearCode Inc.  
クリアコードの社長

## Silver Sponsors



### ClearCode Inc.

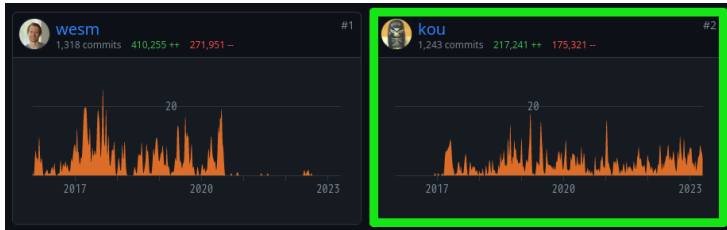
<https://www.clear-code.com/>

Free software is important in ClearCode. We develop/support software with our free software development experiences. We feed back our business experiences to free software.

# Sutou Kouhei

## The 3rd Apache Arrow PMC chair

- ✓ PMC: Project Management Committee  
Apache Arrowのプロジェクト管理委員会の3代目代表
- ✓ #2 commits (コミット数2位)



# Sutou Kouhei

## The pioneer in Ruby and ADBC

- ✓ A Ruby committer
  - ✓ Maintain some standard libraries/default gems  
標準ライブラリーとかデフォルトgemのメンテナンスをしている
- ✓ The author of Red ADBC
  - ✓ The official ADBC library for Ruby  
公式のRuby用のADBCライブラリー
- ✓ ADBC is developed by Arrow project  
ADBCはApache Arrowプロジェクトが開発している



# Sutou Kouhei

## The founder of Red Data Tools

- ✓ Provides data processing tools for Ruby

Ruby用のデータ処理ツールを提供するプロジェクト

<https://red-data-tools.github.io/>

<https://red-data-tools.github.io/ja/>

- ✓ Policies

ポリシー

- ✓ 5. Ignore criticism from outsiders

部外者からの非難は気にしない

Ignore "I use XXX for it instead of Ruby because..."

- ✓ 6. Fun!

楽しくやろう!

# Topic

話すこと

Let's use **Ruby** to  
extract and load  
large data!

大量データの読み書きにもRubyを使おうぜ！

# Embulk?

- ✓ Bulk data loader implemented with Java  
Javaで実装されたバルクデータローダー
- ✓ JRuby supported!  
JRubyサポート!

# "Embulk v0.11 is coming soon"

「Embulk v0.11 がまもなく出ます」

<https://www.embulk.org/articles/2023/04/13/embulk-v0.11-is-coming-soon.html>

*“we plan to gradually shrink our support on (J)Ruby”*

Embulk の (J)Ruby サポートは徐々に縮小していく計画です。

# Another approach: ADBC

## 別のアプローチ：ADBC

- ✓ **Arrow Database Connectivity**
- ✓ **A single API for accessing many DBs**  
各種DBにアクセスするための共通API
- ✓ **Like Active Record/Sequel in Ruby**  
Rubyで言えばActive RecordやSequelみたいなもの

# ADBC : Features

## ADBC : 特徴

### ✓ Cross-language

多言語対応

✓ Active Record needs adapters impl-ed in Ruby  
Active RecordではRubyでアダプターを実装しないといけない

✓ ADBC can use adapters impl-ed in other langs  
ADBCでは他の言語で実装されたアダプターも使える

### ✓ Optimized for large columnar data

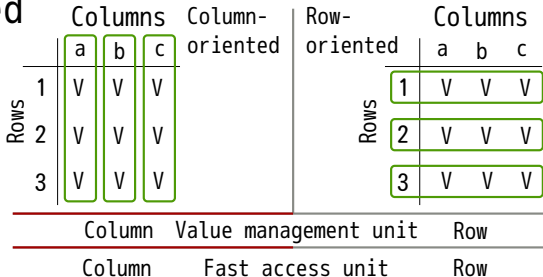
大きな列指向データに最適化

# Large column-oriented data

## 大きな列指向データ

✓ Large:  $\geq 1M$  records with 1 column  
 大きな：1カラムなら100万レコード以上

✓ Column-oriented  
 列指向



# Optimized for large columnar data

大きな列指向データに最適化

- ✓ Apache Arrow data format:  
Minimize data interchange cost!  
Apache Arrowデータフォーマット：データ交換コストがめっちゃ安い！

- ✓ Partitioned result sets:  
Fast data extract

結果セットの分割：高速なデータ読み込み

Apache Arrowフォーマットは  
なぜ速いのか

- ✓ Bulk insert:  
Fast data load

バルクインサート：高速なデータ書き込み

須藤功平

株式会社クリアコード

db tech showcase ONLINE 2020

2020-12-08

<https://slide.rabbit-shocker.org/authors/kou/db-tech-showcase-online-2020/>

Slide Arrowフォーマット

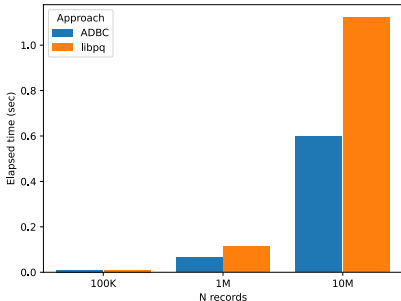
Powered by Rabbit 3.0.2



# How fast is ADBC?

## ADBCはどのくらい速いの？

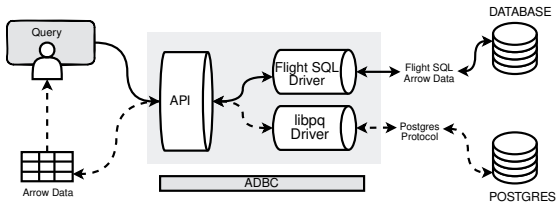
- ✓ 1 integer column  
整数値カラム1つだけ
- ✓ SELECT \* FROM x
- ✓ Lower is faster  
低いほど速い
- ✓ About 2x faster  
with 10M records  
1000万レコードで2倍ほど速い



# Architecture

## アーキテクチャー

- ✓ **Single API**  
同じAPIで使える
- ✓ **Driver per protocol**  
プロトコルごとにドライバーを用意
- ✓ **API returns Arrow data**  
レスポンスはArrowデータ



<https://arrow.apache.org/img/ADBCFlow2.svg> Apache-2.0 © 2016-2023 The Apache Software Foundation

# API

- ✓ C API
  - ✓ Bindings: GLib, Python, R, **Ruby**
- ✓ Go API
- ✓ Java API
- ✓ Rust API (WIP)

See also: <https://arrow.apache.org/arrow/0.3.0/format/specification.html>

# C API

- ✓ **AdbcDatabase**: It holds state shared by multiple connections  
複数の接続を管理
- ✓ **AdbcConnection**: It's a single, logical connection to a database  
1つの接続を管理
- ✓ **AdbcStatement**: It holds state related to query execution  
クエリーの実行を管理

See also: <https://arrow.apache.org/adbc/0.3.0/cpp/api/adbc.html>

# Ruby API: Extract

```
require "adbc"
options = {
  driver: "adbc_driver_postgresql",
  uri: "postgresql://127.0.0.1:5432/db",
}
ADBC::Database.open(**options) do |database|
  database.connect do |connection|
    connection.open_statement do |statement|
      query = "SELECT * FROM data"
      table, = statement.query(query)
    p table
    end
  end
end
```

# Ruby API: Load

```
require "adbc"
options = {
  driver: "adbc_driver_postgresql",
  uri: "postgresql://127.0.0.1:5432/db",
}
ADBC::Database.open(**options) do |database|
  database.connect do |connection|
    connection.open_statement do |statement|
      input = Arrow::Table.load("in.arrow")
      statement.ingest("table", input)
      # ...
    end
  end
end
```

# Ruby API - Active Record

# WIP

<https://github.com/red-data-tools/activerecord-odbc-adapter>

Join us! We need to improve drivers too.

# Available drivers

## 利用可能なドライバー

DB	Status
DuckDB	Beta
Flight SQL	Beta
PostgreSQL	Experimental
SQLite	Beta



# How to implement a driver

## ドライバーの実装方法

- ✓ Choose C, C++ or Go
- ✓ See the following implementations:
  - ✓ C: <https://github.com/apache/arrow-adbc/tree/main/c/driver/sqlite>
  - ✓ C++: <https://github.com/apache/arrow-adbc/tree/main/c/driver/postgresql>
  - ✓ Go (Go API): <https://github.com/apache/arrow-adbc/tree/main/go/adbc/driver/flightsql>
  - ✓ Go (C API): <https://github.com/apache/arrow-adbc/blob/main/go/adbc/pkg/flightsql/driver.go>

# Current ADBC

## 現時点のADBC

✓ 1 integer column

整数値カラム1つだけ

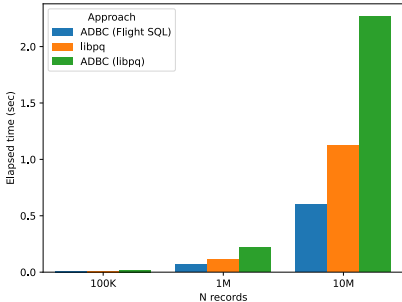
✓ SELECT \* FROM x

✓ Lower is faster

低いほど速い

✓ libpq driver  
is slow for now...

実は現時点ではlibpqドライバーは遅い...



# Flight SQL?

SQL  
on

Apache Arrow Flight

Apache Arrow Flightの上でSQLを使えるようにしたもの

# Apache Arrow Flight?

- ✓ Arrow format based fast RPC framework  
Arrowフォーマットを使った高速RPCフレームワーク
- ✓ Minimum data interchange cost!  
データ交換コストがめっちゃ安い!
- ✓ Parallel transfers  
並列転送
- ✓ Stream processing  
ストリーム処理



**Apache Arrow Flight**  
ビッグデータ用高速データ転送フレームワーク

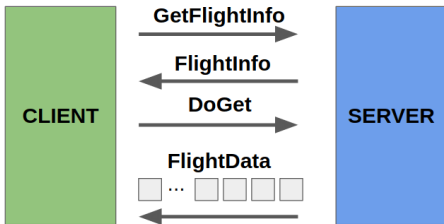
須藤功平  
株式会社クリアコード  
db tech showcase 2021  
2021-11-17

<https://slide.rabbit-shocker.org/authors/kou/db-tech-showcase-2021/>

# Simple usage

## 簡単な使い方

### Simple Client-Server Execution Flow



[https://arrow.apache.org/img/20191014\\_flight\\_simple.png](https://arrow.apache.org/img/20191014_flight_simple.png)  
Apache License 2.0 - © 2016-2021 The Apache Software Foundation

# GetFlightInfo

## ✓ Client → Server

クライアント → サーバー

## ✓ Server returns how to get data

サーバーはデータの取得方法を返す

## ✓ FlightInfo: How to get data

FlightInfo: データの取得方法

## ✓ Metadata: Schema, # of records, ...

メタデータ: スキーマ・総レコード数...

## ✓ 1+ Endpoints: Data may be distributed!

複数エンドポイント: データは複数ヶ所に分散しているかもしれない!

Simple Client-Server Execution Flow



# DoGet

- ✓ Client → Server

クライアント → サーバー

- ✓ Server returns data

サーバーはデータを返す

- ✓ Data: Record batch stream

データ: レコードバッチのストリーム

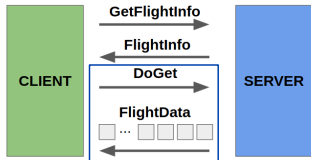
- ✓ Called as FlightData in protocol

プロトコルレベルではFlightDataと呼んでいる

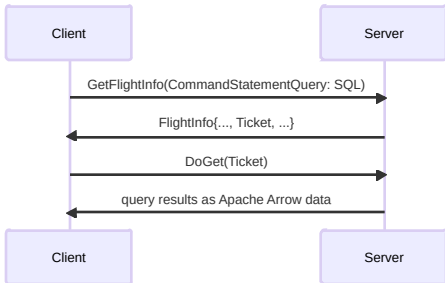
- ✓ Record batch: 0+ records

レコードバッチ: 0個以上のレコードの集まり

Simple Client-Server Execution Flow



# Apache Arrow Flight SQL



<https://arrow.apache.org/blog/2022/02/16/introducing-arrow-flight-sql/>  
Apache License 2.0 - © 2016-2023 The Apache Software Foundation



# Current ADBC

## 現時点のADBC

✓ 1 integer column

整数値カラム1つだけ

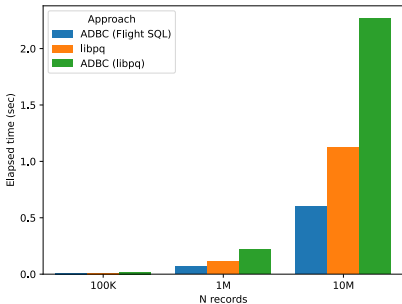
✓ SELECT \* FROM x

✓ Lower is faster

低いほど速い

✓ libpq driver  
is slow for now...

実は現時点ではlibpqドライバーは遅い...



# But can PostgreSQL talk Flight SQL?

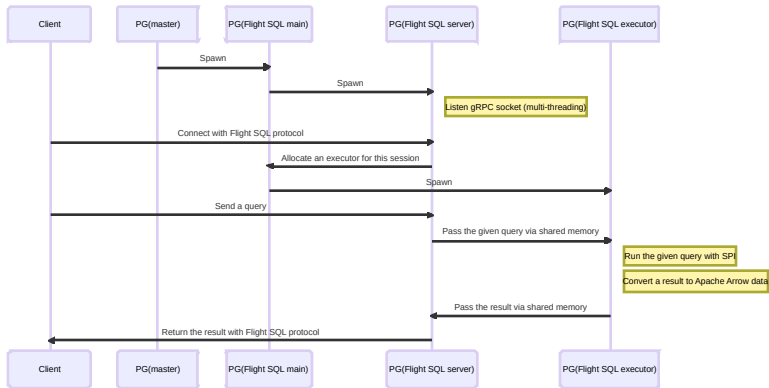
でもPostgreSQLはFlight SQLをしゃべれるの？

## Flight SQL adapter

<https://github.com/apache/arrow-flight-sql-postgresql>

I'm the author

# Architecture



# Wrap up

## まとめ

- ✓ We can use Ruby to extract and load large data by ADBC! (in a few years...)  
ADBCを使うとRubyで大量データを読み書きできるよ! (近いうちに。。。)
- ✓ PostgreSQL 'll be Flight SQL ready soon!  
すぐにPostgreSQLでFlight SQLを使えるようになるよ!
- ✓ We can use ADBC via Active Record soon  
すぐにActive Record経由でADBCを使えるようになるよ!

# Join us!

## 一緒に開発しようぜ!

- ✓ Red Data Tools: A project that provides data processing tools for Ruby

Red Data Tools : Ruby用のデータ処理ツールを提供するプロジェクト

<https://red-data-tools.github.io/>

~~<https://red-data-tools.github.io/ja/>~~

- ✓ You can implement something with us!

一緒になにか作ろうぜ!

<https://gitter.im/red-data-tools/en>

~~<https://gitter.im/red-data-tools/ja>~~

# Sponsor us?

## 資金援助しない？

- ✓ Provide XX% work time to your employee to work on Red Data Tools  
業務時間のXX%をRed Data Toolsの作業をできるようにする
- ✓ Employ a full-time Red Data Tools developer  
フルタイムのRed Data Tools開発者を雇用する
- ✓ Pay Red Data Tools continuously  
Red Data Toolsに継続的に資金を提供する  
Red Data Toolsのだれかがお金で時間を確保して作業する
- ✓ Or contact me!  
相談して！