Как Apache Arrow поможет управиться JS с большими данными?

Николай Шувалов

О себе

- Senior Full Stack Developer
- Отдел данных в билайне

- https://t.me/evelas
- @evelas

Agenda

- Расскажу о формате Arrow
- Поговорим о строковых и столбчатых форматах данных
- Посмотрим какие есть реализации и где применяется Apache Arrow
- Рассмотрим преимущества Arrow
- Почему не WASM и как взаимодействуют WASM с Apache Arrow JS
- Примеры использования Arrow JS
- Выводы

Строковой vs Столбчатый

Рассмотрим таблицу в разных форматах

phone	date	name
1	2022-05-24	foo
2	2023-01-01	bar

Сравнение других форматов популярных в вебе (CSV / JSON)

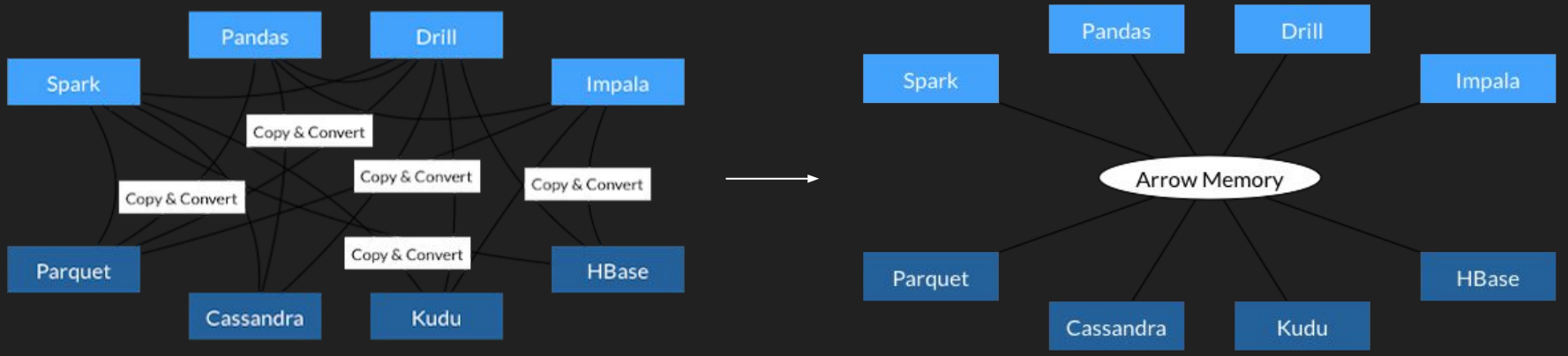
Что различает CSV / JSON от Arrow?

CSV (строковый)	phone, date, name 1, 2022-05-24, foo 2, 2023-01-01, bar
JSON (строковый)	[{“phone”: “1”, “date”: “2022-05-24”, “name”:”foo”}, {“phone”: “2”, “date”: “2023-01-01”, “name”:”bar”}]
Столбчатый	phone: [1, 2] date: [2022-06-23, 2023-01-01] name: [foo, bar]

Поддержка native number, lists, dates, etc

- Значения в text-based форматах (CSV/JSON) хранятся в string
- Некоторые значения (NaN) не заменяются в (CSV/JSON) без соглашений
- Arrow хранит типы в бинарном формате, что уменьшает расходы на парсинг и делает его более компактным

Как появился Apache Arrow



Как получился такой формат

- сериализация/десериализация, когда вы передаете данные из системы в систему, вы получаете адскую скорость производительности

- представление в памяти идентично его представлению на другом конце, и вы идентичны на любом языке, с которым работаете, поэтому память можно просто передавать из системы в систему как есть

Особенности

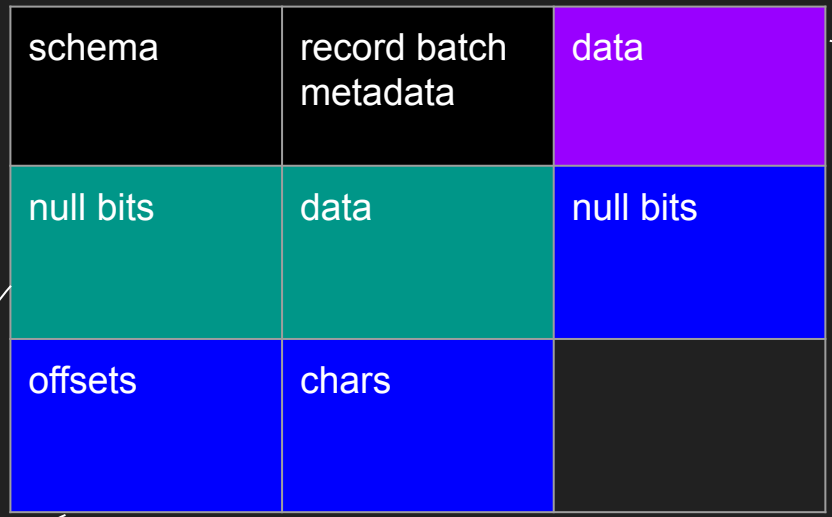
- для эффективных аналитических операций
- arrow — это «представление в памяти во время выполнения»

Цель

- Транспортировка колоночной памяти

Логический и физический уровни

phone	date	name
1	2022-05-24	foo
2	2023-01-01	bar
…	…	…
5	NULL	NULL



Наглядно

offset	data (chars)
[0,3,6]	foobar

Где применяется Apache Arrow?

- Сервис запрашивает данные у КХ
- Pandas + pyarrow
- Polars в основе использует
- В вебе

Реализации Apache Arrow

Реализация	Работает в вебе
Rust	-
C#	-
JS	+
Остальные реализации	-

Apache Arrow JS

Что включает в себя:

- Написан на TS
- Доступен для node и браузера
- Поддержка ESM / CJS
- Treeshakable

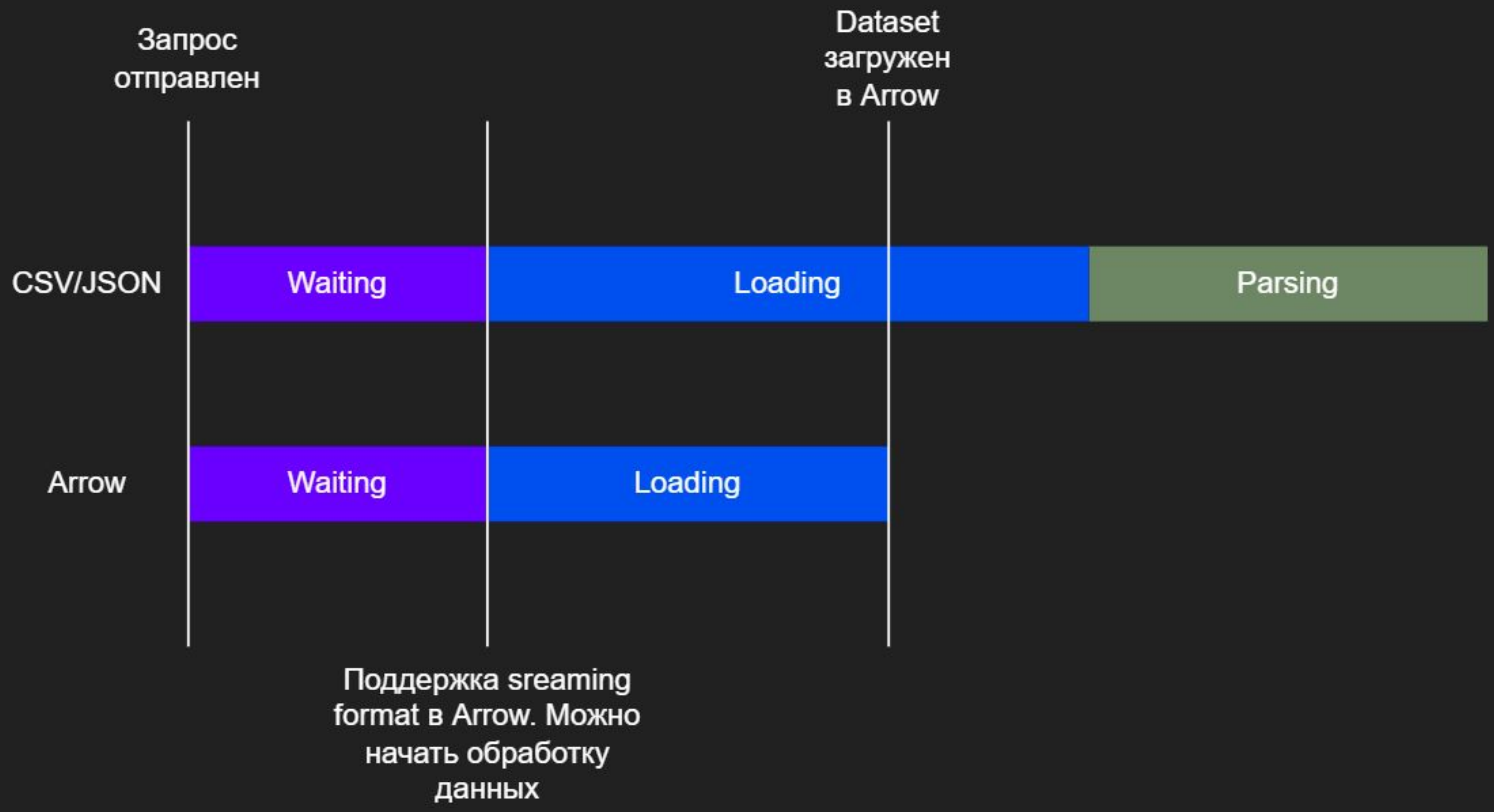
Request

Request

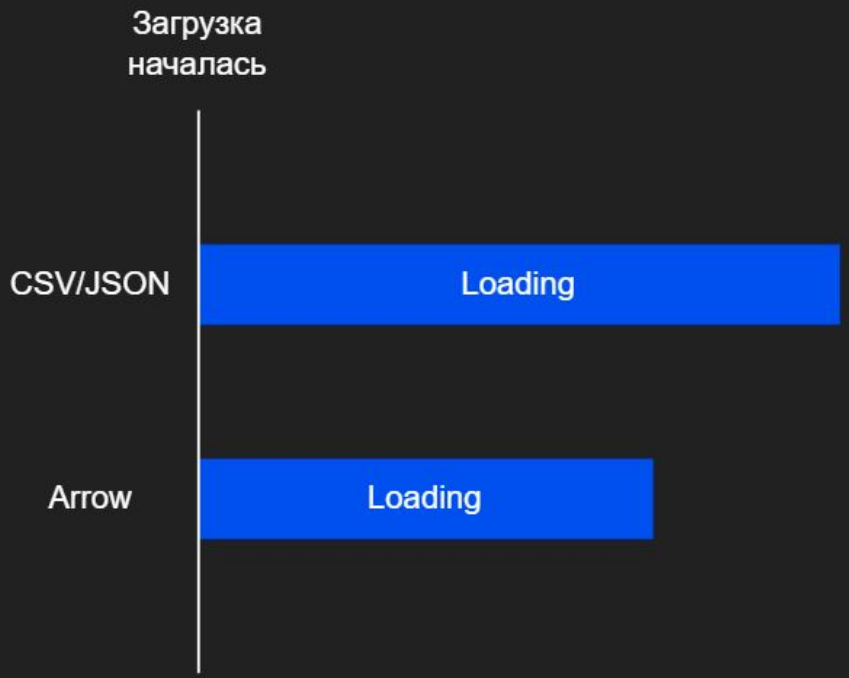
Request


Запрос отправлен
Dataset загружен в Arrow
CSV/JSON
Waiting
Loading
Arrow
Waiting
Loading

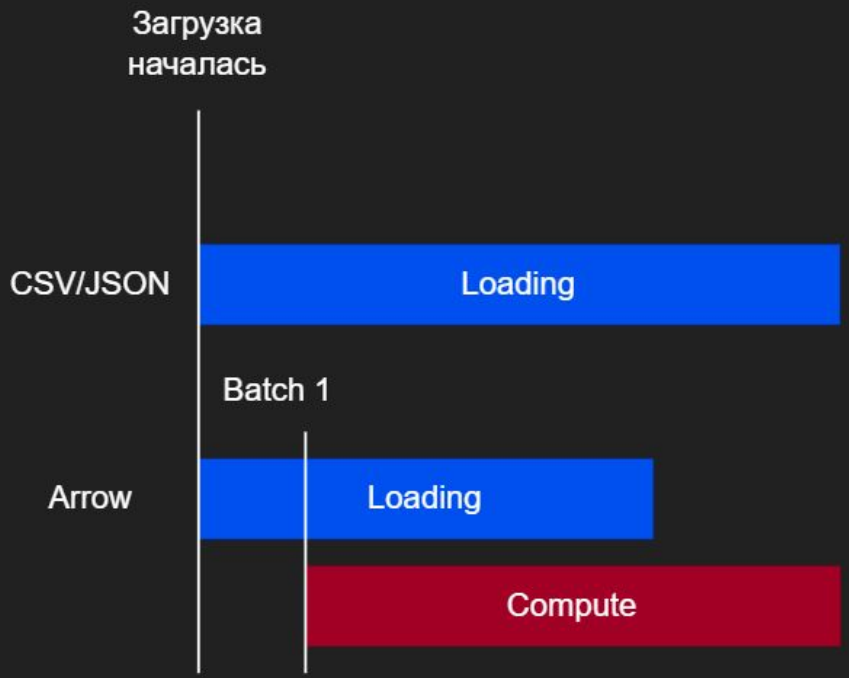
Request


Запрос отправлен
Dataset загружен в Arrow
CSV/JSON
Waiting
Loading
Parsing
Arrow
Waiting
Loading
Поддержка sreaming format в Arrow. Можно начать обработку данных

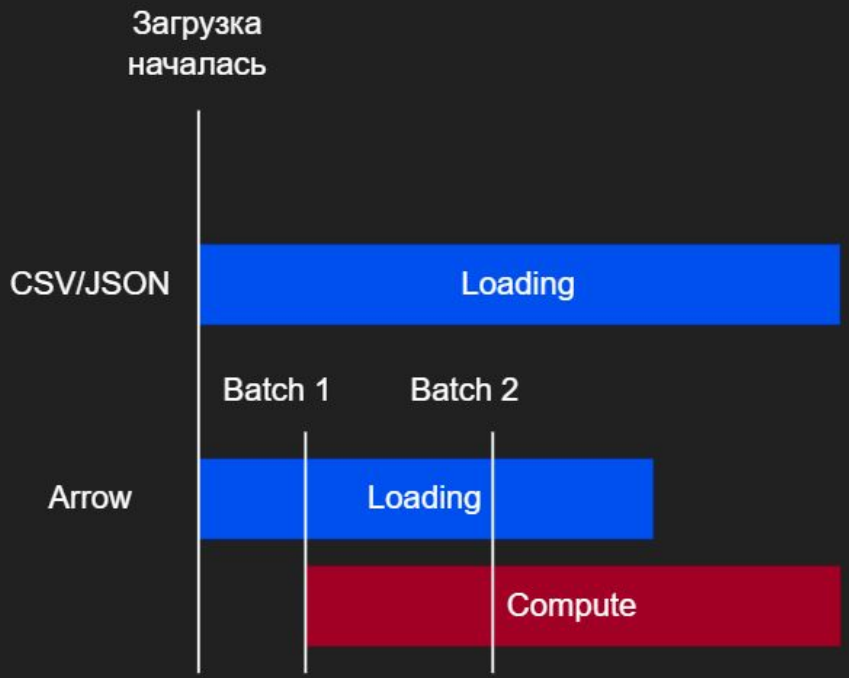
Loading



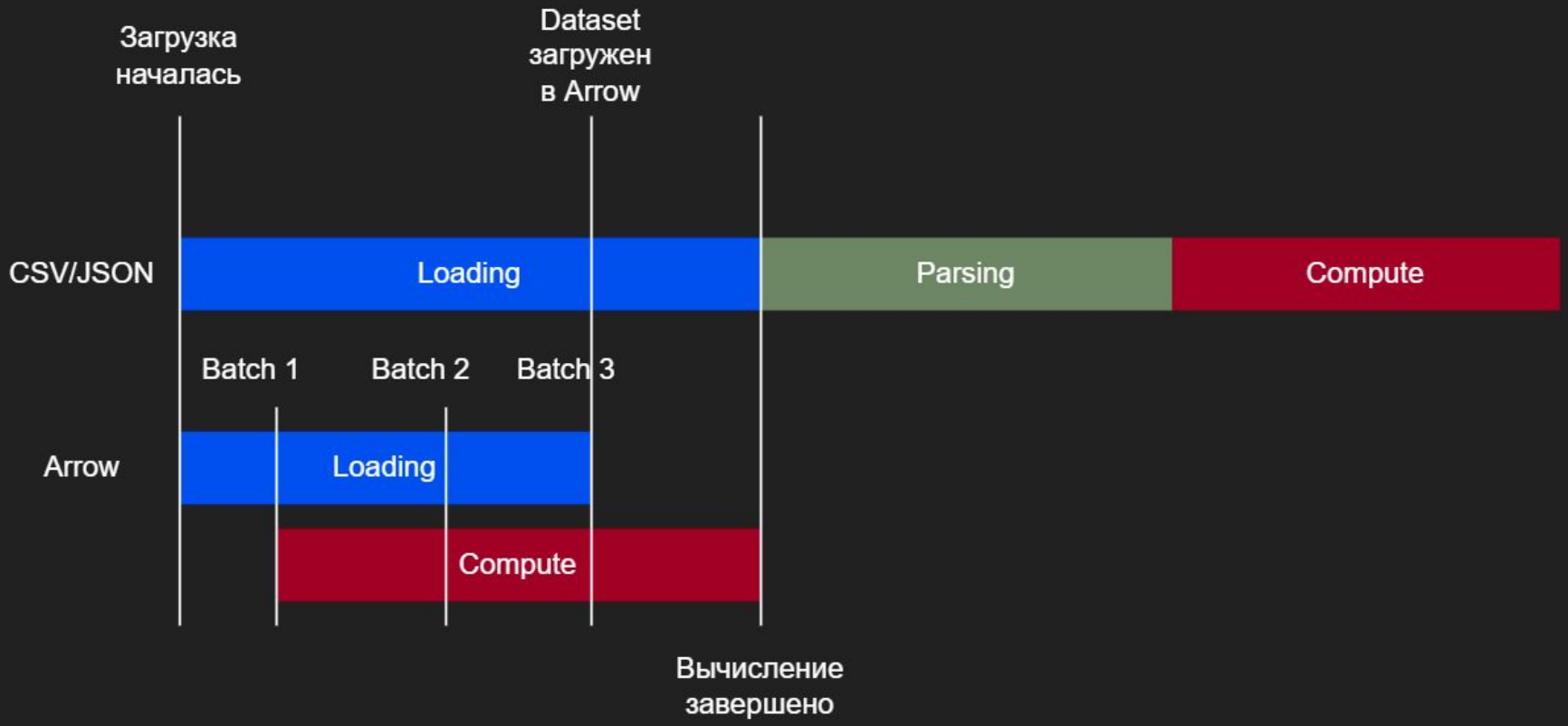
Loading


Загрузка
началась
CSV/JSON
Loading
Batch 1
Arrow
Loading
Compute

Loading


Загрузка
началась
CSV/JSON
Loading
Batch 1
Batch 2
Arrow
Loading
Compute

Loading


Загрузка началась
Dataset загружен в Arrow
CSV/JSON
Loading
Parsing
Compute
Batch 1
Batch 2
Batch 3
Arrow
Loading
Compute
Вычисление завершено

А как можно ускорить без Arrow?

Использовать protobuf:

- Protocol Buffers представляет собой двоичный формат.
- Для десериализации данных необходим отдельный .proto-файл, в котором определяется формат сообщения.

При этом все равно к данным будем применять десериализацию, а потом проводить вычисления

Немного кода

index.ts > ...

```ts
import { readFileSync } from 'fs';
import { RecordBatchFileReader } from 'apache-arrow';

async function bootstrap() {
    const reader = RecordBatchFileReader.from(readFileSync('simple.arrow'));
    //const reader = await RecordBatchFileReader.from(fetch('/data'));
    for await (const batch of reader) {
        const arrayRowsByNameColumn: AllowSharedBufferSource[] = batch.getChild('name').toArray();
        for (let row of arrayRowsByNameColumn) {
            const valueOfRow = new TextDecoder().decode(
                row,
            );
            console.log("value of col 'name': ", valueOfRow);
        }
        console.log(arrayRowsByNameColumn.length)
    }
}
bootstrap();
```

(index)	surname name	id	feature	agreement	phone
0	Uint8Array(5) [83, 109, 105, ... 2 more items] Uint8Array(4) [77, 97, 114, ... 1 more item]	8	0	0	Uint8Array(8) [90, 85, 80, ... 5 more items]
1	Uint8Array(5) [83, 109, 105, ... 2 more items] Uint8Array(4) [77, 97, 114, ... 1 more item]	1	0	0	Uint8Array(8) [90, 85, 80, ... 5 more items]
2	Uint8Array(7) [84, 104, 111, ... 4 more items] Uint8Array(4) [75, 97, 116, ... 1 more item]	2	1	0	Uint8Array(8) [90, 85, 80, ... 5 more items]
3	Uint8Array(5) [83, 109, 105, ... 2 more items] Uint8Array(3) [84, 105, 109]	3	1	0	Uint8Array(8) [90, 85, 80, ... 5 more items]
4	Uint8Array(5) [83, 109, 105, ... 2 more items] Uint8Array(3) [84, 105, 109]	4	1	0	Uint8Array(8) [90, 85, 80, ... 5 more items]
5	Uint8Array(5) [83, 109, 105, ... 2 more items] Uint8Array(3) [84, 111, 109]	5	0	0	Uint8Array(8) [90, 85, 80, ... 5 more items]
6	Uint8Array(5) [83, 109, 105, ... 2 more items] Uint8Array(4) [74, 111, 104, ... 1 more item]	6	1	0	Uint8Array(8) [90, 85, 80, ... 5 more items]
7	Uint8Array(5) [83, 109, 105, ... 2 more items] Uint8Array(5) [72, 101, 108, ... 2 more items]	7	1	0	Uint8Array(8) [90, 85, 80, ... 5 more items]

```
value of col 'name':  Mary
1
value of col 'name':  Mary
value of col 'name':  Kate
value of col 'name':  Tim
value of col 'name':  Tim
value of col 'name':  Tom
value of col 'name':  John
value of col 'name':  Helen
7
```

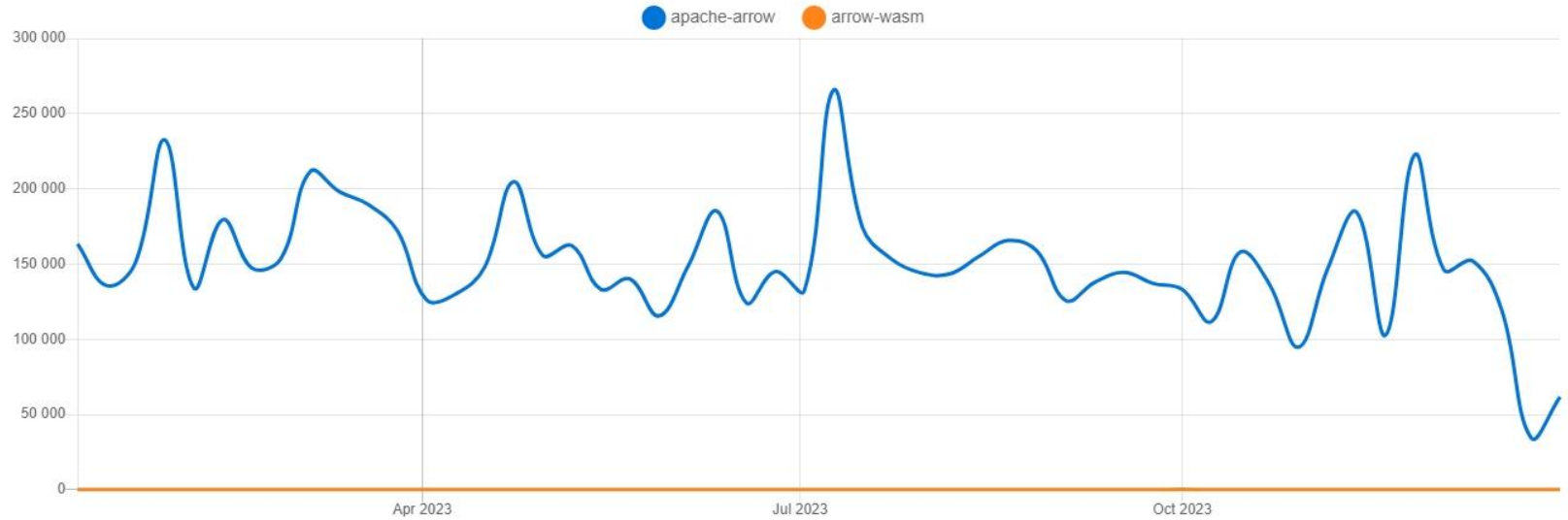
Почему не WASM?

Есть альтернатива - компилировать пакет Rust Apache Arrow в WebAssembly.

- WASM может дать хорошую производительность в приложениях с интенсивными вычислениями.
- Слишком большой и медленный (https://github.com/domoritz/arrow-wasm). (Контекст между JS и WASM оказался медленным)
- Apache Arrow JS может быть отличным инструментом для связи между контекстами WASM и JS.

apache-arrow × | arrow-wasm × | + node-parquet | + parquets | + parquetjs | + parquet | + dataframe-js | + data-forge | + pandas-js | + jskit-learn | + jsdata | + arquero

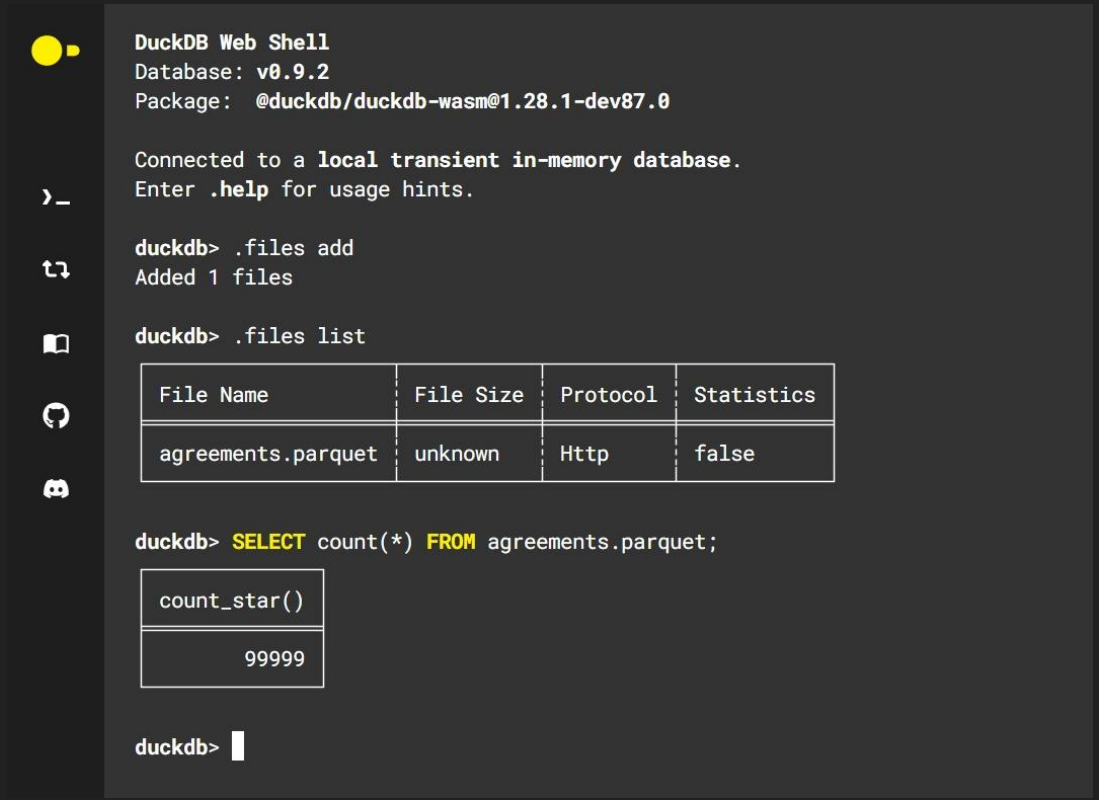
Downloads in past 1 Year



Stats

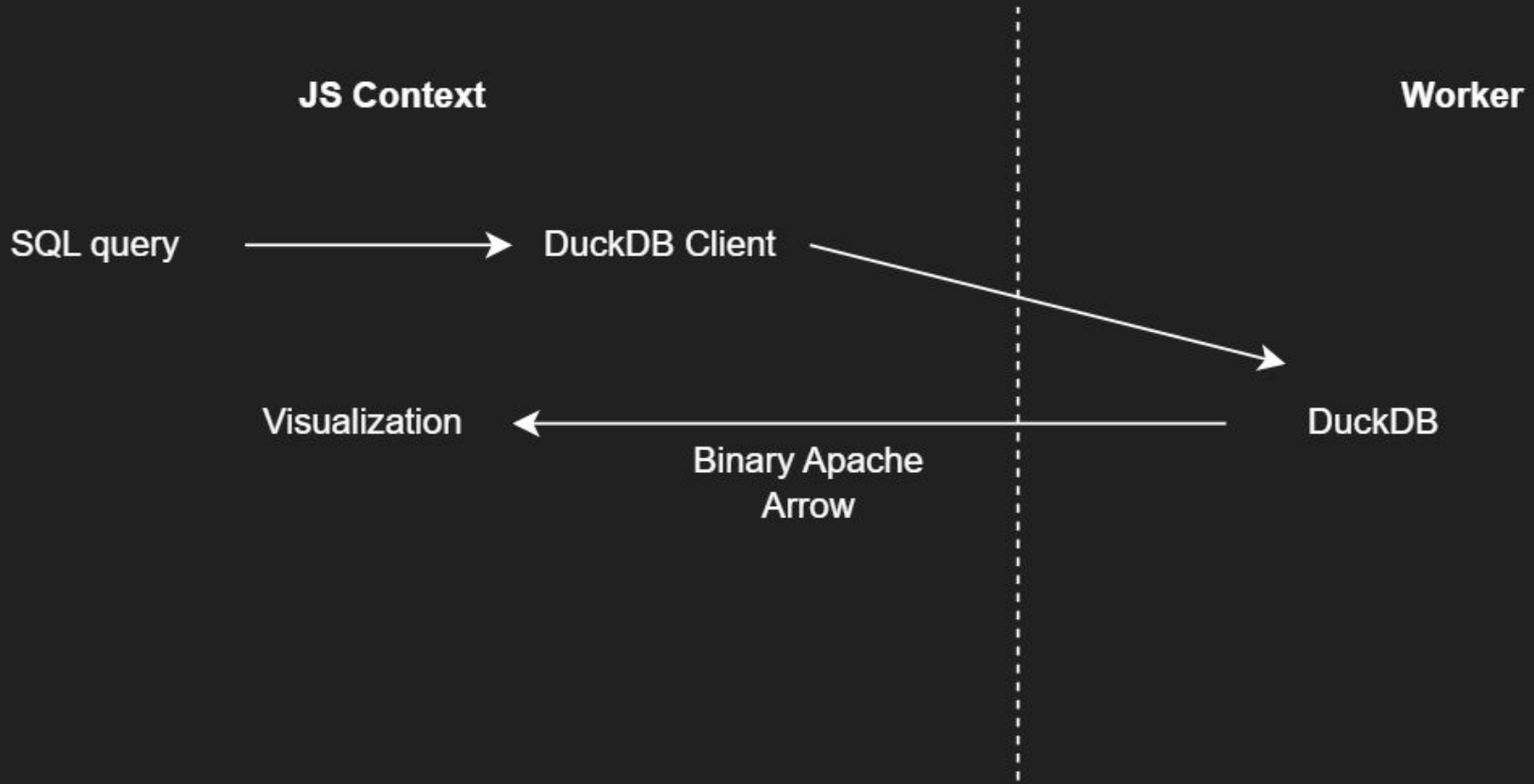
	Stars	Issues	Version	Updated	Created	Size
apache-arrow	-	-	14.0.2	20 days ago	6 years ago	minzipped size 50.7 KB
arrow-wasm	-	-	0.0.20	3 years ago	3 years ago	bundlephobia 429

DuckDB


DuckDB Web Shell
Database: v0.9.2
Package: @duckdb/duckdb-wasm@1.28.1-dev87.0
Connected to a local transient in-memory database.
Enter .help for usage hints.
duckdb> .files add
Added 1 files
duckdb> .files list
File Name
File Size
Protocol
Statistics
agreements.parquet
unknown
Http
false
duckdb> SELECT count(*) FROM agreements.parquet;
count_star()
99999
duckdb>

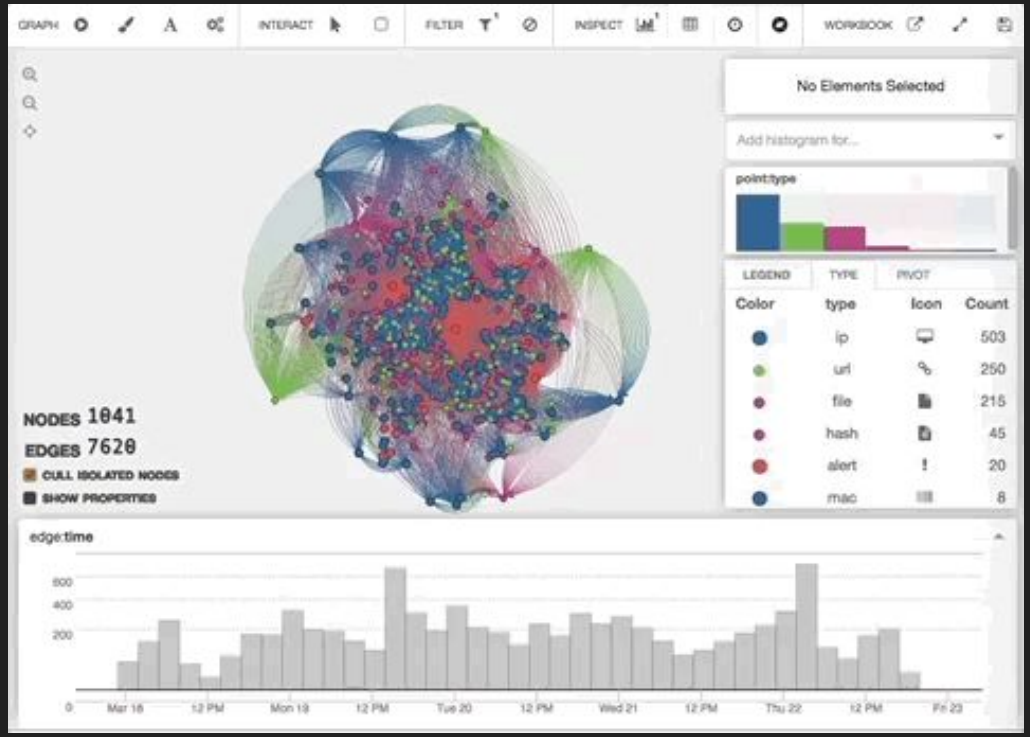
https://shell.duckdb.org/

DuckDB


JS Context
Worker
SQL query
DuckDB Client
Visualization
DuckDB
Binary Apache
Arrow

Другие примеры

- Graphistry
- Falcon
- Vega-loader
- Эффективная загрузка огромных наборов данных D3 с помощью Apache Arrow



Выводы

- Узнали о формате arrow
- Сравнили строковый и столбчатый формат
- Посмотрели загрузку данных в CSV/JSON и Arrow на временной шкале
- Затронули пример кода с чтением данных
- На схеме узнали как можно использовать реализацию Arrow в вебе

Спасибо за внимание

Вопросы