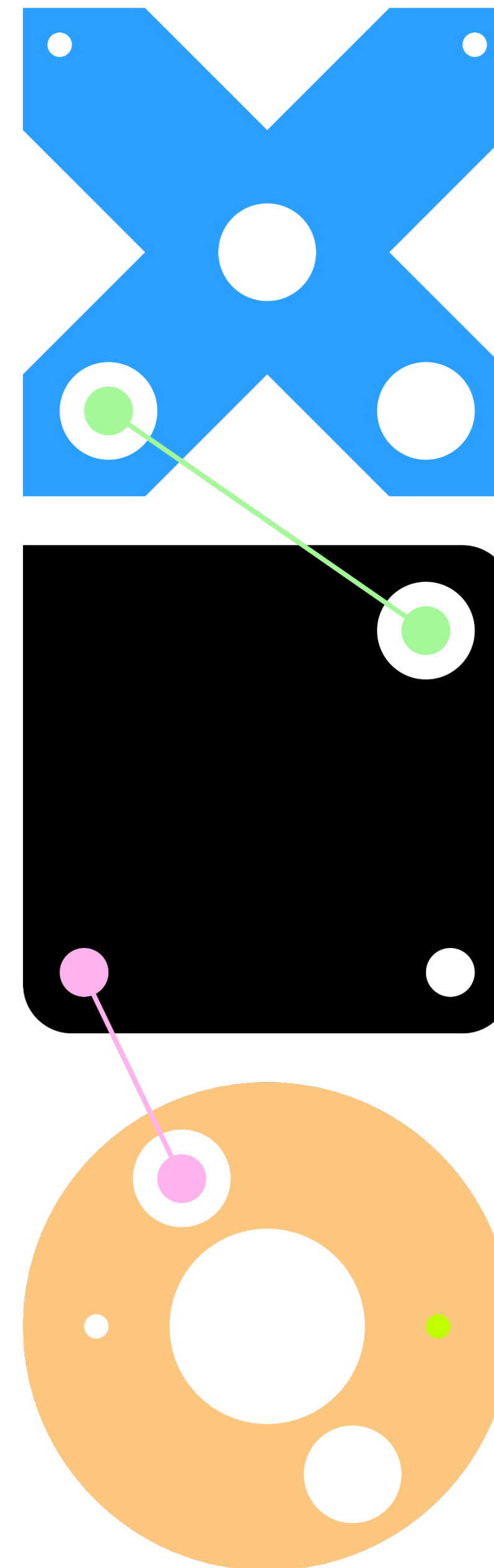


Что происходит между **SELECT** и данными

Петр Гуринов

Команда продуктовой архитектуры
Платформы Данных



Гуринов Петр

Команда продуктовой архитектуры
Платформы Данных, Yandex Cloud



Выпускник Мехмата МГУ и Бизнес-
Информатики НИУ ВШЭ

Работаю с хранилищами данных более 5 лет

Развиваю Managed Spark и другие compute-
движки в Платформе Данных

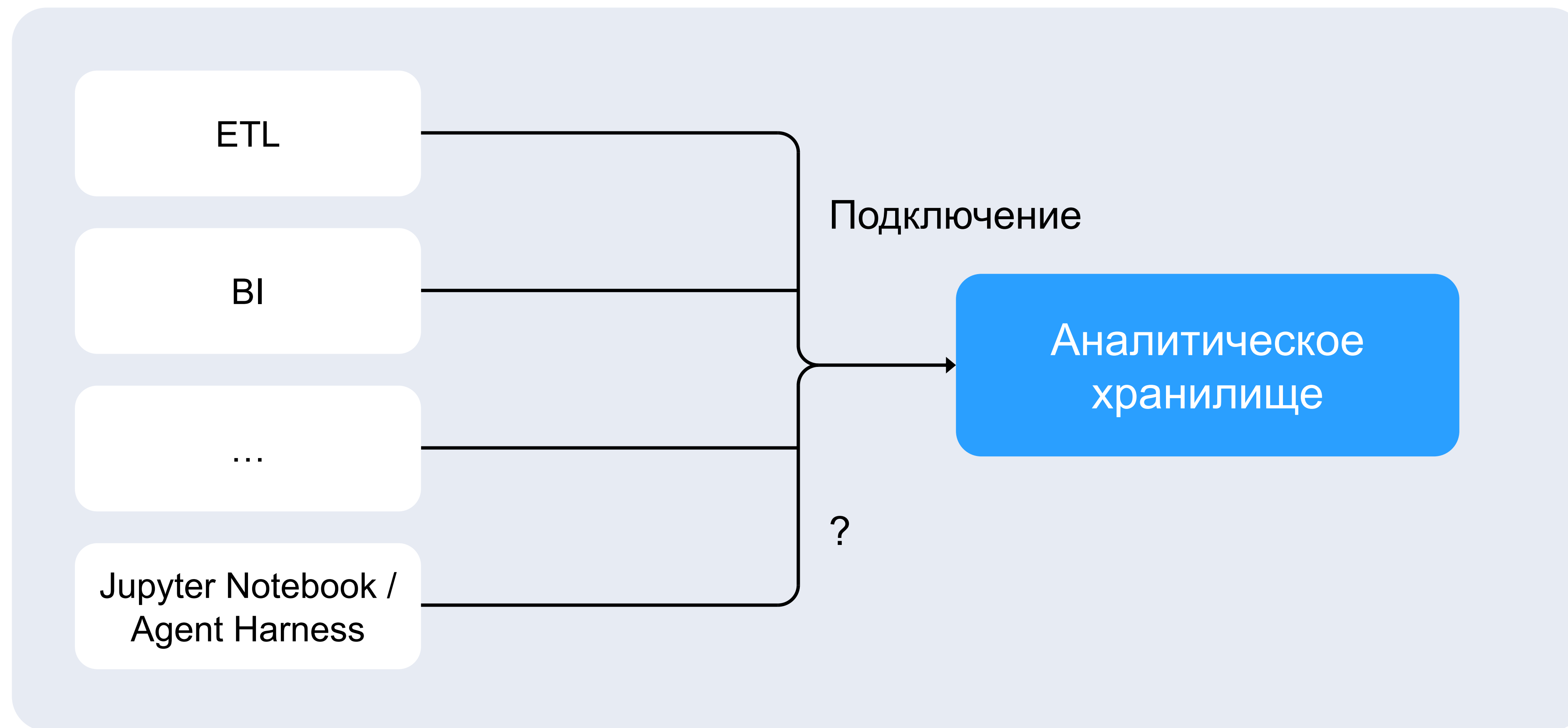
О чём поговорим

1. Введение
2. Путь данных от БД до клиента
3. Что летит по сети: wireshark
 1. в PostgreSQL
 2. в ClickHouse®
4. Намерили бенчмарком
5. Итоги

Введение

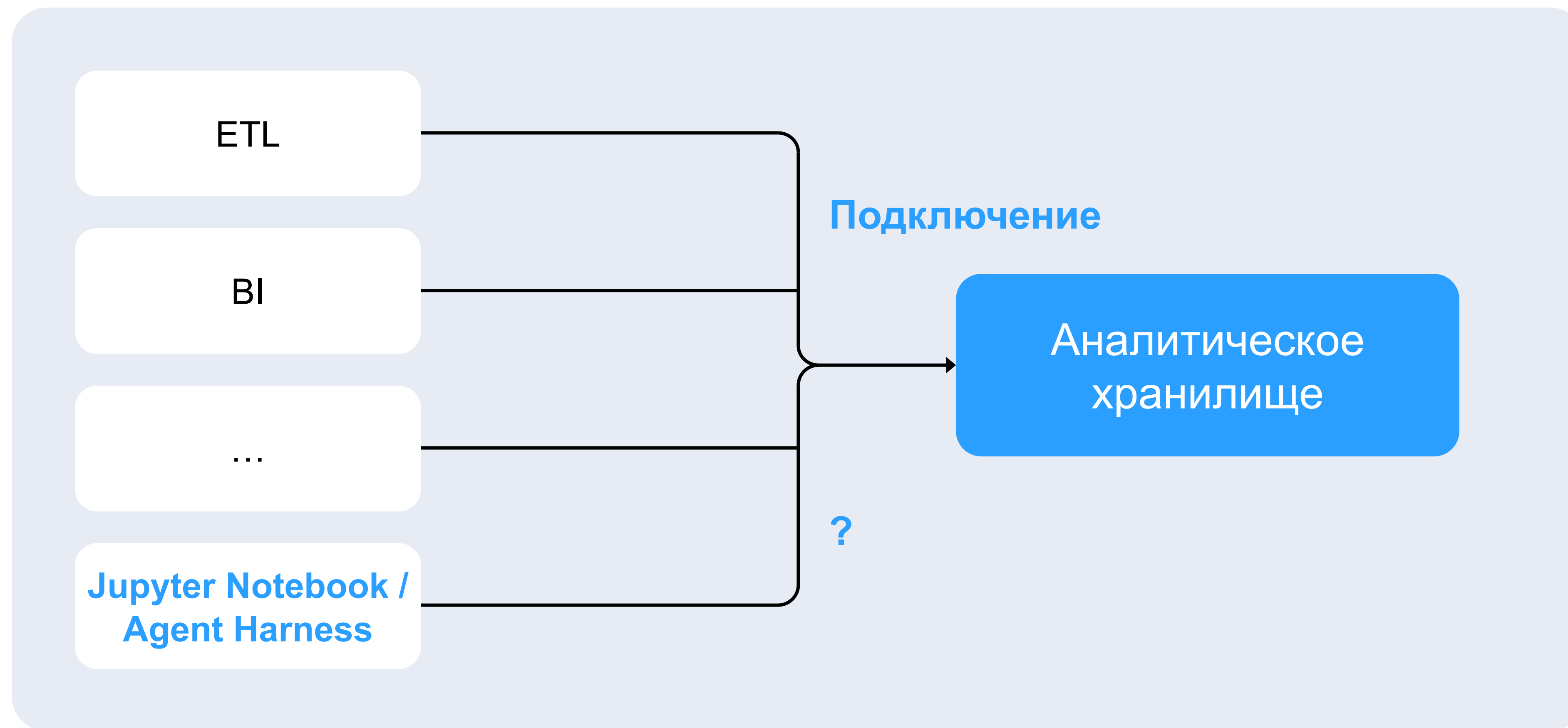
Мы проектируем компоненты

и оставляем подключение вне внимания



Мы проектируем компоненты

и оставляем подключение вне внимания



Архитектуру согласовали

Способ доставки — нет



Движок и формат
хранения выбрали



Ресурсы кластера
посчитали



Способ получения
результата скопировали
из документации

Архитектуру согласовали

Способ доставки — нет



Движок и формат хранения выбрали



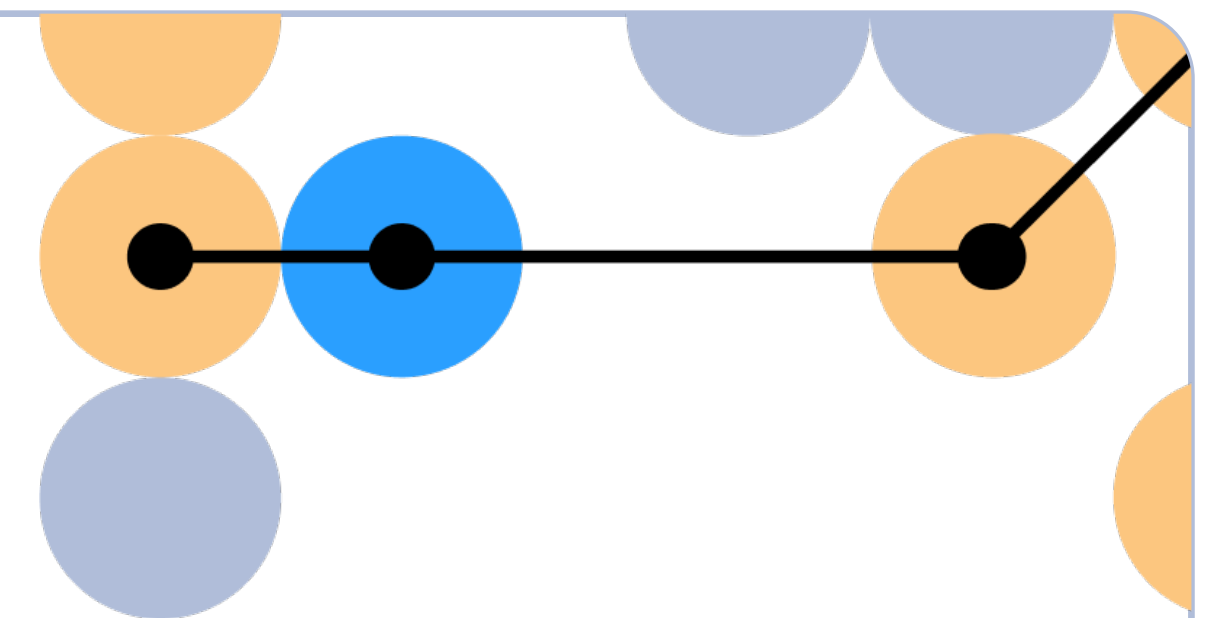
Ресурсы кластера посчитали



Способ получения результата скопировали из документации

Хотя он определяет, какой будет

- Драйвер
- Протокол
- Формат
- Способ сборки результата



«Hold My Data Hostage — A Case for Client Protocol Redesign»

Способ	Время, s	Передано, GB	Коэф. к netcat
netcat	10.25	7.19	1.0×
MySQL	101.22	7.44	9.9×
DB2	169.48	7.33	16.5×
DBMS X	189.50	6.35	18.5×
PostgreSQL	201.89	10.39	19.7×
MonetDB	209.02	8.97	20.4×
MySQL + C	391.27	2.85	38.2×
Hive	627.75	8.69	61.2×
MongoDB	686.45	43.6	67.0×

10-67×

время к netcat



Статья

PostgreSQL и ClickHouse®

PostgreSQL 18.6

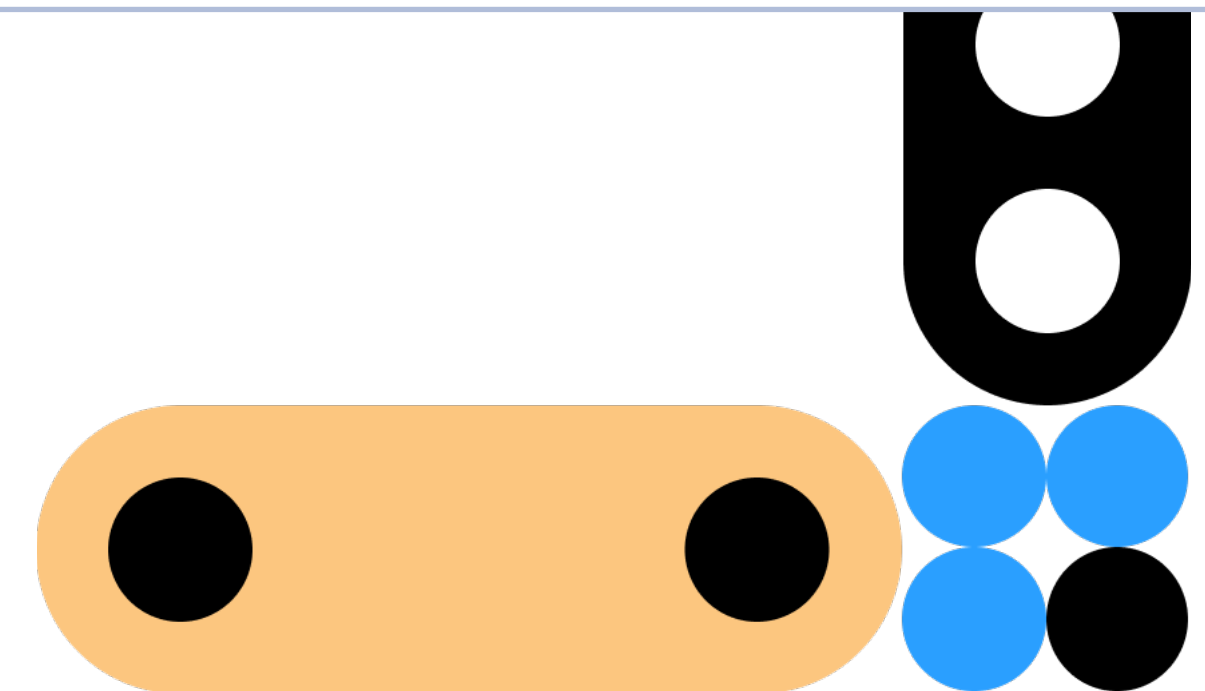
- Один основной протокол pg-wire
- Много драйверов и режимов
- Построчная модель хранения и результата

ClickHouse® 26.7

- Несколько протоколов и форматов ответа
- Колоночная модель хранения. Можно ее сохранить вплоть до Arrow на клиенте

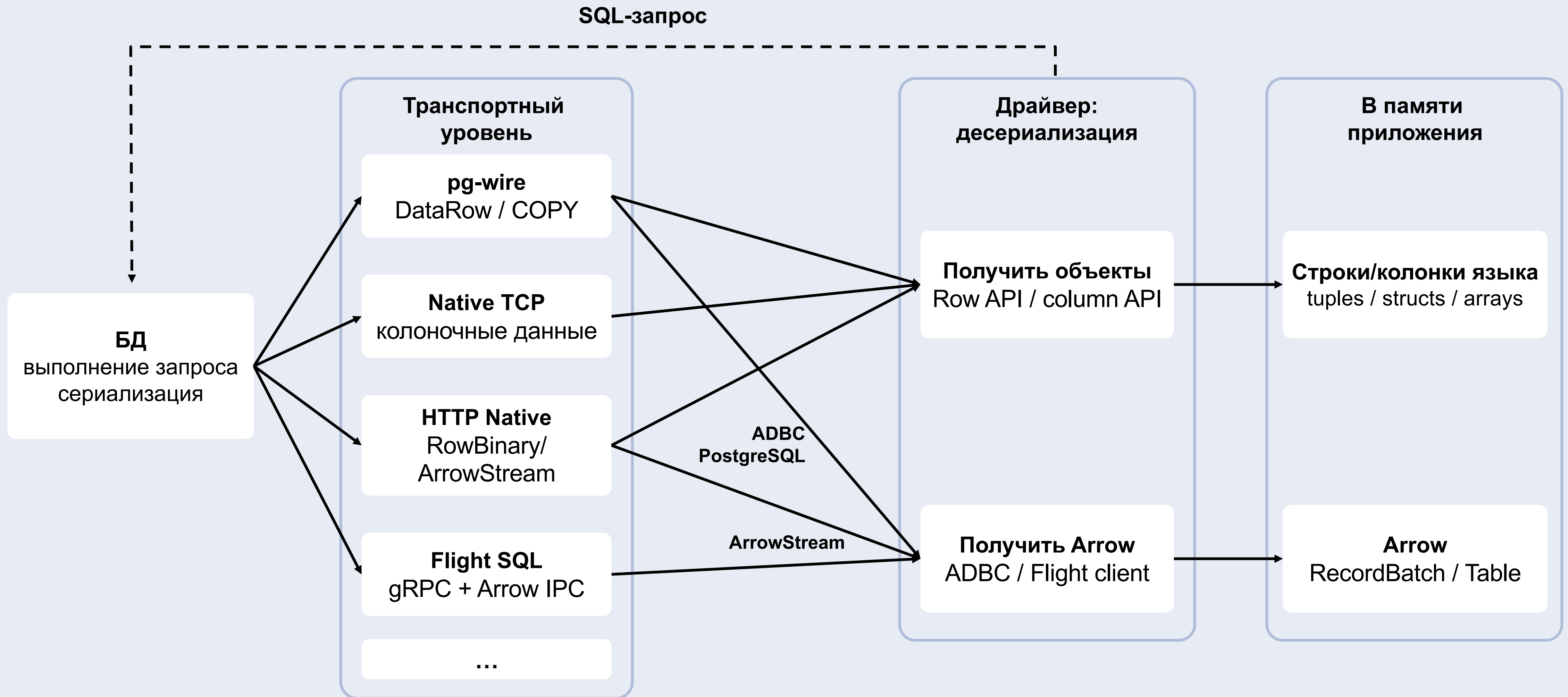


- Какие решения принимает драйвер
- Где и как меняется формат данных



Путь данных от БД до клиента

Путь запроса от БД до клиента



Одна строка определяет все уровни способа доставки

```
engine = create_engine("driver://user@host/db")  
with engine.connect() as connection:  
    df = pd.read_sql(sql, connection)
```

Что явно

- Клиент
- Подключение к базе
- SQL-запрос

Что скрыто

- Драйвер
- Протокол
- Формат
- Десериализация в DataFrame

Запрос ↔ результат

Участок	↓ Запрос	↑ Результат
Приложение	SQL и параметры	Структура данных или поток строк
Клиентский API	Выполнить запрос	Прочитать порцию или весь результат
Драйвер	Кодирует команды в сообщения	Декодирует и преобразует данные
Сеть	Сообщения выбранного протокола	Метаданные и данные выбранного формата
Серверный endpoint	Принимает и разбирает запрос	Сериализует результат
Движок БД	Планирует и выполняет запрос	Вычисляет значения результата

Формат ответа связывает сериализацию на сервере и декодирование в драйвере

Пять слоев на пути данных

Слой	За что отвечает	Примеры
API	Какие вызовы доступны приложению	DB-API, JDBC, ODBC, ADBC, Go database/sql, ...
Драйвер	Как выполнить вызовы и собрать результат	psycopg, pgJDBC, psqlODBC, ADBC PostgreSQL, clickhouse-connect, ...
Протокол обмена	Какие сообщения отправлять и в каком порядке	pg-wire, clickhouse Native TCP, HTTP, Flight SQL, ...
Формат ответа	Как закодированы данные в сообщениях	text / binary DataRow, Native, RowBinary, Arrow IPC, ...
Объект приложения	С чем работает код после получения данных	rows, tuples, structs, arrays, Arrow Table, DataFrame, ...

- **API описывает вызовы**
- **Драйвер реализует обмен и сборку результата**

Arrow, ADBC, Flight и Flight SQL

	Для чего	Как используем
Arrow	Колоночное представление данных в памяти	RecordBatch, Arrow Table
Arrow IPC	Сериализация схемы и batches	Поток или файл, ClickHouse FORMAT ArrowStream
ADBC	Клиентский API работы с БД через arrow	Драйверы ADBC postgresql и ADBC flight SQL
Flight	RPC для передачи потоков Arrow	gRPC и Arrow IPC
Flight SQL	SQL-команды и метаданные поверх Flight	Flight SQL client и совместимый endpoint

Для Flight SQL нужна поддержка на серверной стороне

**Что летит по сети:
проверим wireshark**

Для примера сгенерируем таблицу

Как участвовал в конференции

conf varchar(32)	starts_at timestamp	speaker int4
SmartData	2023-09-13 11:00:00	0
SmartData	2024-09-08 10:15:00	1
SmartData	2025-10-05 10:00:00	0
SmartData	2026-09-23 10:30:00	1

А по сети будут передаваться по-разному

```
SELECT conf, starts_at, speaker FROM smartdata ORDER BY starts_at
```

**Что летит по сети
в PostgreSQL:
проверим wireshark**

PostgreSQL → результат запроса

- pg-wire · Simple Query · Extended Query
 - └ RowDescription + DataRow: text / binary
- pg-wire · COPY OUT
 - └ CopyOutResponse + поток CopyData: text / CSV / binary
- дополнительный Flight SQL endpoint
 - └ gRPC + Arrow IPC: схема и batches
- ...

PostgreSQL → результат запроса

- `pg-wire` · `Simple Query` · `Extended Query`
 - └ `RowDescription` + `DataRow`: `text` / `binary`
- `pg-wire` · `COPY OUT`
 - └ `CopyOutResponse` + поток `CopyData`: `text` / `CSV` / `binary`
- дополнительный `Flight SQL endpoint`
 - └ `gRPC` + `Arrow IPC`: `схема` и `batches`
- ...

PostgreSQL — Simple Query

Client

PostgreSQL

Query("SELECT ...")



RowDescription



DataRow: text



DataRow: text



CommandComplete



ReadyForQuery



PostgreSQL — Simple Query

Apply a display filter ... <⌘/>

No.	Time	Source	Destination	Protocol	Length	Info
5	0.001539	127.0.0.1	127.0.0.1	PGSQL	64	>?
7	0.004677	127.0.0.1	127.0.0.1	PGSQL	57	<
9	0.005257	127.0.0.1	127.0.0.1	PGSQL	119	>?
11	0.006793	127.0.0.1	127.0.0.1	PGSQL	65	<R
13	0.006969	127.0.0.1	127.0.0.1	PGSQL	70	>p
15	0.010919	127.0.0.1	127.0.0.1	PGSQL	525	<R/S/S/S/S/S/S/S/S/S/S/S/S/S/S/K/Z
17	0.012067	127.0.0.1	127.0.0.1	PGSQL	127	>Q
19	0.012876	127.0.0.1	127.0.0.1	PGSQL	352	<T/D/D/D/D/C/Z
21	0.013514	127.0.0.1	127.0.0.1	PGSQL	61	>X

> Frame 9: Packet, 119 bytes on wire (952 bits), 119 bytes captured (952 bits) on interface lo0, i

Null/Loopback

Family: IP (2)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 55936, Dst Port: 5432, Seq: 9, Ack: 2, Len: 63

PostgreSQL

Type: Startup message

Length: 63

Protocol major version: 3

Protocol minor version: 0

Parameter name: user

Parameter value: workshop

Parameter name: database

Parameter value: workshop

Parameter name: application_name

Parameter value: psql

0000	02 00 00 00 45 00 00 73	00 00 40 00 40 06 00 00	E..s ..@.@...
0010	7f 00 00 01 7f 00 00 01	da 80 15 38 18 dd 84 90	8....
0020	98 4e 48 17 80 18 18 ec	fe 67 00 00 01 01 08 0a	·NH.....g.....
0030	aa 88 6e 37 93 0a 90 c5	00 00 00 3f 00 03 00 00	··n7.....?....
0040	75 73 65 72 00 77 6f 72	6b 73 68 6f 70 00 64 61	user·wor kshop·da
0050	74 61 62 61 73 65 00 77	6f 72 6b 73 68 6f 70 00	tabase·w orkshop·
0060	61 70 70 6c 69 63 61 74	69 6f 6e 5f 6e 61 6d 65	applicat ion_name
0070	00 70 73 71 6c 00 00		·psql··

PostgreSQL — Simple Query

Apply a display filter ... <⌘/>

No.	Time	Source	Destination	Protocol	Length	Info
5	0.001539	127.0.0.1	127.0.0.1	PGSQL	64	>?
7	0.004677	127.0.0.1	127.0.0.1	PGSQL	57	<
9	0.005257	127.0.0.1	127.0.0.1	PGSQL	119	>?
11	0.006793	127.0.0.1	127.0.0.1	PGSQL	65	<R
13	0.006969	127.0.0.1	127.0.0.1	PGSQL	70	>p
15	0.010919	127.0.0.1	127.0.0.1	PGSQL	525	<R/S/S/S/S/S/S/S/S/S/S/S/S/S/S/K/Z
17	0.012067	127.0.0.1	127.0.0.1	PGSQL	127	>Q
19	0.012876	127.0.0.1	127.0.0.1	PGSQL	352	<T/D/D/D/D/C/Z
21	0.013514	127.0.0.1	127.0.0.1	PGSQL	61	>X

> Frame 17: Packet, 127 bytes on wire (1016 bits), 127 bytes captured (1016 bits) on interface lo0

Null/Loopback

Family: IP (2)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 55936, Dst Port: 5432, Seq: 86, Ack: 480, Len: 71

PostgreSQL

Type: Simple query

Length: 70

Query: SELECT conf, starts_at, speaker FROM smartdata ORDER BY starts_at

0000 02 00 00 00 45 00 00 7b 00 00 40 00 40 06 00 00 E...{ ..@.@...

0010 7f 00 00 01 7f 00 00 01 da 80 15 38 18 dd 84 dd 8.....

0020 98 4e 49 f5 80 18 18 e5 fe 6f 00 00 01 01 08 0a .NI.....o.....

0030 aa 88 6e 3e 93 0a 90 cc 51 00 00 00 46 53 45 4c ..n>....Q...FSEL

0040 45 43 54 20 63 6f 6e 66 2c 20 73 74 61 72 74 73 ECT conf , starts

0050 5f 61 74 2c 20 73 70 65 61 6b 65 72 20 46 52 4f _at, speaker FRO

0060 4d 20 73 6d 61 72 74 64 61 74 61 20 4f 52 44 45 M smartd ata ORDE

0070 52 20 42 59 20 73 74 61 72 74 73 5f 61 74 00 R BY sta rts_at.

PostgreSQL — Simple Query

Apply a display filter ... <⌘/>

No.	Time	Source	Destination	Protocol	Length	Info
5	0.001539	127.0.0.1	127.0.0.1	PGSQL	64	>?
7	0.004677	127.0.0.1	127.0.0.1	PGSQL	57	<
9	0.005257	127.0.0.1	127.0.0.1	PGSQL	119	>?
11	0.006793	127.0.0.1	127.0.0.1	PGSQL	65	<R
13	0.006969	127.0.0.1	127.0.0.1	PGSQL	70	>p
15	0.010919	127.0.0.1	127.0.0.1	PGSQL	525	<R/S/S/S/S/S/S/S/S/S/S/S/S/S/S/K/Z
17	0.012067	127.0.0.1	127.0.0.1	PGSQL	127	>Q
19	0.012876	127.0.0.1	127.0.0.1	PGSQL	352	<T/D/D/D/D/C/Z
21	0.013514	127.0.0.1	127.0.0.1	PGSQL	61	>X

PostgreSQL

Type: Row description
Length: 83

Field count: 3

Column name: conf
Table OID: 43264
Column index: 1
Type OID: 1043
Column length: -1
Type modifier: 36
Format: Text (0)

Column name: starts_at

Column name: speaker

PostgreSQL

Type: Data row
Length: 47

Field count: 3

Column length: 9

0000	02 00 00 00 45 00 01 5c	00 00 40 00 40 06 00 00	E..\ ..@·@··
0010	7f 00 00 01 7f 00 00 01	15 38 da 80 98 4e 49 f5	 ·8··NI·
0020	18 dd 85 24 80 18 18 eb	ff 50 00 00 01 01 08 0a	...\$..... ·P····
0030	93 0a 90 cd aa 88 6e 3e	54 00 00 00 53 00 03 63	n> T··S··c
0040	6f 6e 66 00 00 00 a9 00	00 01 00 00 04 13 ff ff	onf·.....
0050	00 00 00 24 00 00 73 74	61 72 74 73 5f 61 74 00	...\$··st arts_at·
0060	00 00 a9 00 00 02 00 00	04 5a 00 08 ff ff ff ff	 ·Z····
0070	00 00 73 70 65 61 6b 65	72 00 00 00 a9 00 00 03	··speake r····
0080	00 00 00 17 00 04 ff ff	ff ff 00 00 44 00 00 00	 ···D··
0090	2f 00 03 00 00 00 09 53	6d 61 72 74 44 61 74 61	/.....S martData
00a0	00 00 00 13 32 30 32 33	2d 30 39 2d 31 33 20 31	...·2023 -09-13 1
00b0	31 3a 30 30 3a 30 30 00	00 00 01 30 44 00 00 00	1:00:00· ...0D··
00c0	2f 00 03 00 00 00 09 53	6d 61 72 74 44 61 74 61	/.....S martData
00d0	00 00 00 13 32 30 32 34	2d 30 39 2d 30 38 20 31	...·2024 -09-08 1
00e0	30 3a 31 35 3a 30 30 00	00 00 01 31 44 00 00 00	0:15:00· ...1D··
00f0	2f 00 03 00 00 00 09 53	6d 61 72 74 44 61 74 61	/.....S martData
0100	00 00 00 13 32 30 32 35	2d 31 30 2d 30 35 20 31	...·2025 -10-05 1
0110	30 3a 30 30 3a 30 30 00	00 00 01 30 44 00 00 00	0:00:00· ...0D··
0120	2f 00 03 00 00 00 09 53	6d 61 72 74 44 61 74 61	/.....S martData
0130	00 00 00 13 32 30 32 36	2d 30 39 2d 32 33 20 31	...·2026 -09-23 1
0140	30 3a 33 30 3a 30 30 00	00 00 01 31 43 00 00 00	0:30:00· ...1C··
0150	0d 53 45 4c 45 43 54 20	34 00 5a 00 00 00 05 49	·SELECT 4·Z····I

PostgreSQL — Simple Query

Apply a display filter ... <⌘/>						
No.	Time	Source	Destination	Protocol	Length	Info
5	0.001539	127.0.0.1	127.0.0.1	PGSQL	64	>?
7	0.004677	127.0.0.1	127.0.0.1	PGSQL	57	<
9	0.005257	127.0.0.1	127.0.0.1	PGSQL	119	>?
11	0.006793	127.0.0.1	127.0.0.1	PGSQL	65	<R
13	0.006969	127.0.0.1	127.0.0.1	PGSQL	70	>p
15	0.010919	127.0.0.1	127.0.0.1	PGSQL	525	<R/S/S/S/S/S/S/S/S/S/S/S/S/S/S/K/Z
17	0.012067	127.0.0.1	127.0.0.1	PGSQL	127	>Q
19	0.012876	127.0.0.1	127.0.0.1	PGSQL	352	<T/D/D/D/D/C/Z
21	0.013514	127.0.0.1	127.0.0.1	PGSQL	61	>X

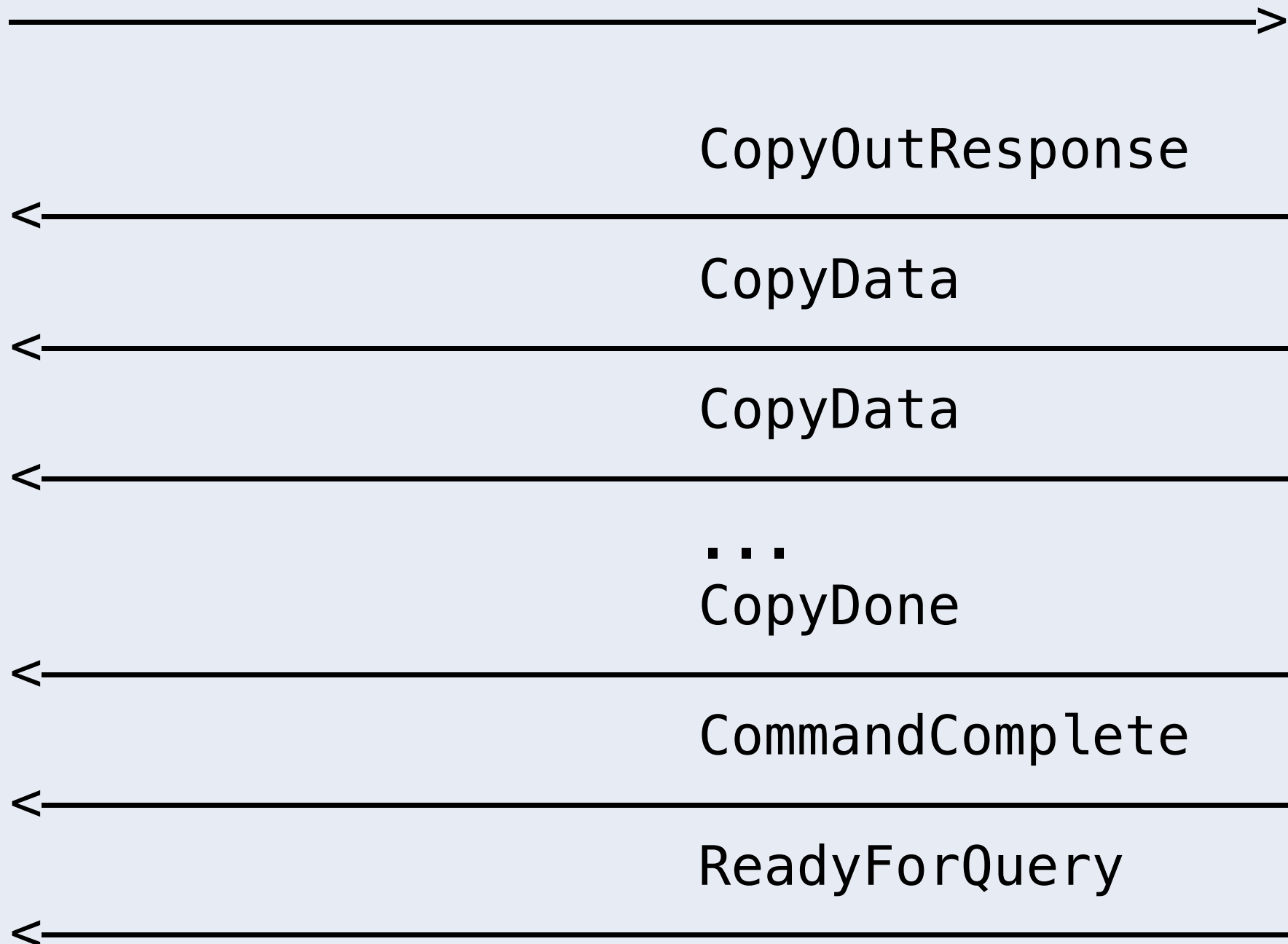
▼ PostgreSQL		0000 02 00 00 00 45 00 01 5c 00 00 40 00 40 06 00 00	E.. \ ..@..@...
Type: Data row		0010 7f 00 00 01 7f 00 00 01 15 38 da 80 98 4e 49 f5	 8...NI..
Length: 47		0020 18 dd 85 24 80 18 18 eb ff 50 00 00 01 01 08 0a	...\$.... P.....
▼ Field count: 3		0030 93 0a 90 cd aa 88 6e 3e 54 00 00 00 53 00 03 63	n> T...S..c
Column length: 9		0040 6f 6e 66 00 00 00 a9 00 00 01 00 00 04 13 ff ff	onf.....
Data: 536d61727444617461		0050 00 00 00 24 00 00 73 74 61 72 74 73 5f 61 74 00	...\$.st arts_at..
Column length: 19		0060 00 00 a9 00 00 02 00 00 04 5a 00 08 ff ff ff ff	 Z.....
Data: 323032362d30392d32332031303a33303a3030		0070 00 00 73 70 65 61 6b 65 72 00 00 00 a9 00 00 03	..speake r.....
Column length: 1		0080 00 00 00 17 00 04 ff ff ff ff 00 00 44 00 00 00	 D...
Data: 31		0090 2f 00 03 00 00 00 09 53 6d 61 72 74 44 61 74 61	/.....S martData
▼ PostgreSQL		00a0 00 00 00 13 32 30 32 33 2d 30 39 2d 31 33 20 31	...2023 -09-13 1
Type: Command completion		00b0 31 3a 30 30 3a 30 30 00 00 00 01 30 44 00 00 00	1:00:00. ...0D...
Length: 13		00c0 2f 00 03 00 00 00 09 53 6d 61 72 74 44 61 74 61	/.....S martData
Tag: SELECT 4		00d0 00 00 00 13 32 30 32 34 2d 30 39 2d 30 38 20 31	...2024 -09-08 1
▼ PostgreSQL		00e0 30 3a 31 35 3a 30 30 00 00 00 01 31 44 00 00 00	0:15:00. ...1D...
Type: Ready for query		00f0 2f 00 03 00 00 00 09 53 6d 61 72 74 44 61 74 61	/.....S martData
Length: 5		0100 00 00 00 13 32 30 32 35 2d 31 30 2d 30 35 20 31	...2025 -10-05 1
Status: Idle (73)		0110 30 3a 30 30 3a 30 30 00 00 00 01 30 44 00 00 00	0:00:00. ...0D...
		0120 2f 00 03 00 00 00 09 53 6d 61 72 74 44 61 74 61	/.....S martData
		0130 00 00 00 13 32 30 32 36 2d 30 39 2d 32 33 20 31	...2026 -09-23 1
		0140 30 3a 33 30 3a 30 30 00 00 00 01 31 43 00 00 00	0:30:00. ...1C...
		0150 0d 53 45 4c 45 43 54 20 34 00 5a 00 00 00 05 49	·SELECT 4·Z...I

PostgreSQL — COPY

Client

PostgreSQL

COPY (...) TO STDOUT



PostgreSQL — COPY

Apply a display filter ... <⌘/>

No.	Time	Source	Destination	Protocol	Length	Info
5	0.000132	127.0.0.1	127.0.0.1	PGSQL	97	>?
7	0.010767	127.0.0.1	127.0.0.1	PGSQL	65	<R
9	0.010915	127.0.0.1	127.0.0.1	PGSQL	70	>p
11	0.021685	127.0.0.1	127.0.0.1	PGSQL	521	<R/S/S/S/S/S/S/S/S/S/S/S/S/S/S/K/Z
13	0.021885	127.0.0.1	127.0.0.1	PGSQL	160	>Q
15	0.024298	127.0.0.1	127.0.0.1	PGSQL	279	<H/d/d/d/d/d/c/C/Z
17	0.024988	127.0.0.1	127.0.0.1	PGSQL	61	>X

> Frame 7: Packet, 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on interface lo0, id 0

Null/Loopback

Family: IP (2)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 5432, Dst Port: 55979, Seq: 1, Ack: 42, Len: 9

PostgreSQL

Type: Authentication request

Length: 8

Authentication type: Plaintext password (3)

0000 02 00 00 00 45 00 00 3d 00 00 40 00 40 06 00 00 E..= ..@.@...

0010 7f 00 00 01 7f 00 00 01 15 38 da ab f1 bb 54 f4 8.....T.

0020 83 13 e6 cf 80 18 18 ec fe 31 00 00 01 01 08 0a 1.....

0030 17 79 60 d7 19 ee f2 14 52 00 00 00 08 00 00 00 .y`.....R.....

0040 03

PostgreSQL — COPY

Apply a display filter ... <⌘/>						
No.	Time	Source	Destination	Protocol	Length	Info
5	0.000132	127.0.0.1	127.0.0.1	PGSQL	97	>?
7	0.010767	127.0.0.1	127.0.0.1	PGSQL	65	<R
9	0.010915	127.0.0.1	127.0.0.1	PGSQL	70	>p
11	0.021685	127.0.0.1	127.0.0.1	PGSQL	521	<R/S/S/S/S/S/S/S/S/S/S/S/S/S/S/S/K/Z
13	0.021885	127.0.0.1	127.0.0.1	PGSQL	160	>Q
15	0.024298	127.0.0.1	127.0.0.1	PGSQL	279	<H/d/d/d/d/d/c/C/Z
17	0.024988	127.0.0.1	127.0.0.1	PGSQL	61	>X

> Frame 13: Packet, 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits) on interface lo0				0000	02 00 00 00 45 00 00 9c	00 00 40 00 40 06 00 00	E... ..@·@...
Null/Loopback				0010	7f 00 00 01 7f 00 00 01	da ab 15 38 83 13 e6 dd	8.....
Family: IP (2)				0020	f1 bb 56 ce 80 18 18 e5	fe 90 00 00 01 01 08 0a	..V.....
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1				0030	19 ee f2 2a 17 79 60 e2	51 00 00 00 67 43 4f 50	...*.y`. Q...gCOP
> Transmission Control Protocol, Src Port: 55979, Dst Port: 5432, Seq: 56, Ack: 475, Len: 104				0040	59 20 28 53 45 4c 45 43	54 20 63 6f 6e 66 2c 20	Y (SELEC T conf,
PostgreSQL				0050	73 74 61 72 74 73 5f 61	74 2c 20 73 70 65 61 6b	starts_a t, speak
Type: Simple query				0060	65 72 20 46 52 4f 4d 20	73 6d 61 72 74 64 61 74	er FROM smartdat
Length: 103				0070	61 20 4f 52 44 45 52 20	42 59 20 73 74 61 72 74	a ORDER BY start
Query: COPY (SELECT conf, starts_at, speaker FROM smartdata ORDER BY starts_at) TO STDOUT (FOR				0080	73 5f 61 74 29 20 54 4f	20 53 54 44 4f 55 54 20	s_at) TO STDOUT
				0090	28 46 4f 52 4d 41 54 20	42 49 4e 41 52 59 29 00	(FORMAT BINARY)·

PostgreSQL — COPY

Apply a display filter ... <⌘/>						
No.	Time	Source	Destination	Protocol	Length	Info
5	0.000132	127.0.0.1	127.0.0.1	PGSQL	97	>?
7	0.010767	127.0.0.1	127.0.0.1	PGSQL	65	<R
9	0.010915	127.0.0.1	127.0.0.1	PGSQL	70	>p
11	0.021685	127.0.0.1	127.0.0.1	PGSQL	521	<R/S/S/S/S/S/S/S/S/S/S/S/S/S/S/S/K/Z
13	0.021885	127.0.0.1	127.0.0.1	PGSQL	160	>Q
15	0.024298	127.0.0.1	127.0.0.1	PGSQL	279	<H/d/d/d/d/d/c/C/Z
17	0.024988	127.0.0.1	127.0.0.1	PGSQL	61	>X

> Frame 15: Packet, 279 bytes on wire (2232 bits), 279 bytes captured (2232 bits) on interface lo0				0000	02 00 00 00 45 00 01 13	00 00 40 00 40 06 00 00	E... ..@·@·...
Null/Loopback				0010	7f 00 00 01 7f 00 00 01	15 38 da ab f1 bb 56 ce	8.....V·
Family: IP (2)				0020	83 13 e7 45 80 18 18 eb	ff 07 00 00 01 01 08 0a	...E.....
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1				0030	17 79 60 e4 19 ee f2 2a	48 00 00 00 0d 01 00 03	·y`.....* H.....
> Transmission Control Protocol, Src Port: 5432, Dst Port: 55979, Seq: 475, Ack: 160, Len: 223				0040	00 01 00 01 00 01 64 00	00 00 3a 50 47 43 4f 50	d· ..:PGCOP
PostgreSQL				0050	59 0a ff 0d 0a 00 00 00	00 00 00 00 00 00 00 03	Y.....
Type: CopyOut response				0060	00 00 00 09 53 6d 61 72	74 44 61 74 61 00 00 00	Smar tData...
Length: 13				0070	08 00 02 a8 3a 3a b6 0c	00 00 00 00 04 00 00 00	:.....
Format: Binary (1)				0080	00 64 00 00 00 27 00 03	00 00 00 09 53 6d 61 72	·d···'... ..Smar
> Columns: 3				0090	74 44 61 74 61 00 00 00	08 00 02 c4 97 ae 7d b1	tData... ..}
PostgreSQL				00a0	00 00 00 00 04 00 00 00	01 64 00 00 00 27 00 03	·d···'...
Type: Copy data				00b0	00 00 00 09 53 6d 61 72	74 44 61 74 61 00 00 00	Smar tData...
Length: 58				00c0	08 00 02 e3 65 2a a3 c8	00 00 00 00 04 00 00 00	e*.....
Copy data: 5047434f50590aff0d0a0000000000000000000000300000009536d617274446174610000000080002a83				00d0	00 64 00 00 00 27 00 03	00 00 00 09 53 6d 61 72	·d···'... ..Smar
PostgreSQL				00e0	74 44 61 74 61 00 00 00	08 00 02 ff 22 bb e8 fa	tData... .."
Type: Copy data				00f0	00 00 00 00 04 00 00 00	01 64 00 00 00 06 ff ff	·d.....
Length: 39				0100	63 00 00 00 04 43 00 00	00 0b 43 4f 50 59 20 34	c····C·· ··COPY 4
Copy data: 0003000000009536d617274446174610000000080002c497ae7db1000000000400000001				0110	00 5a 00 00 00 05 49		·Z····I

PostgreSQL — COPY

Apply a display filter ... <⌘/>

No.	Time	Source	Destination	Protocol	Length	Info
5	0.000132	127.0.0.1	127.0.0.1	PGSQL	97	>?
7	0.010767	127.0.0.1	127.0.0.1	PGSQL	65	<R
9	0.010915	127.0.0.1	127.0.0.1	PGSQL	70	>p
11	0.021685	127.0.0.1	127.0.0.1	PGSQL	521	<R/S/S/S/S/S/S/S/S/S/S/S/S/S/S/S/K/Z
13	0.021885	127.0.0.1	127.0.0.1	PGSQL	160	>Q
15	0.024298	127.0.0.1	127.0.0.1	PGSQL	279	<H/d/d/d/d/d/c/C/Z
17	0.024988	127.0.0.1	127.0.0.1	PGSQL	61	>X

PostgreSQL

Type: Copy data
Length: 39
Copy data: 000300000009536d617274444617461000000080002ff22bbe8fa000000000400000001

PostgreSQL

Type: Copy data
Length: 6
Copy data: ffff

PostgreSQL

Type: Copy completion
Length: 4

PostgreSQL

Type: Command completion
Length: 11
Tag: COPY 4

PostgreSQL

Type: Ready for query
Length: 5

Status: Idle (73)

0000 02 00 00 00 45 00 01 13 00 00 40 00 40 06 00 00 E...@·@·
0010 7f 00 00 01 7f 00 00 01 15 38 da ab f1 bb 56 ce 8....V·
0020 83 13 e7 45 80 18 18 eb ff 07 00 00 01 01 08 0a ...E.....
0030 17 79 60 e4 19 ee f2 2a 48 00 00 00 0d 01 00 03 ·y`.....* H.....
0040 00 01 00 01 00 01 64 00 00 00 3a 50 47 43 4f 50 d·...:PGCOP
0050 59 0a ff 0d 0a 00 00 00 00 00 00 00 00 00 00 03 Y.....
0060 00 00 00 09 53 6d 61 72 74 44 61 74 61 00 00 00 Smar tData...
0070 08 00 02 a8 3a 3a b6 0c 00 00 00 00 04 00 00 00:.....
0080 00 64 00 00 00 27 00 03 00 00 00 09 53 6d 61 72 ·d···'...Smar
0090 74 44 61 74 61 00 00 00 08 00 02 c4 97 ae 7d b1 tData...}·
00a0 00 00 00 00 04 00 00 00 01 64 00 00 00 27 00 03 d···'·
00b0 00 00 00 09 53 6d 61 72 74 44 61 74 61 00 00 00 Smar tData...
00c0 08 00 02 e3 65 2a a3 c8 00 00 00 00 04 00 00 00 e*.....
00d0 00 64 00 00 00 27 00 03 00 00 00 09 53 6d 61 72 ·d···'...Smar
00e0 74 44 61 74 61 00 00 00 08 00 02 ff 22 bb e8 fa tData..."...
00f0 00 00 00 00 04 00 00 00 01 64 00 00 00 06 ff ff d.....
0100 63 00 00 00 04 43 00 00 00 0b 43 4f 50 59 20 34 c···C···COPY 4
0110 00 5a 00 00 00 05 49 ·Z···I

PostgreSQL — ADBC

The PostgreSQL driver provides access to any database that supports the PostgreSQL wire format. It wraps libpq, the client library for PostgreSQL. The project owes credit to 0x0L's pgeon for the overall approach

```
import adbc_driver_postgresql.dbapi

uri =
    "postgresql://user:pass@localhost:5433/postgres"
with adbc_driver_postgresql.dbapi.connect(uri) as
    conn:
        pass
```



PostgreSQL – JDBC

[illegible]

За сценарий: клиент получил 61 пак. / 241 239 Б, отправил 61 пак. / 4 531 Б. Всего 245 770 Б.
TCP payload: 238 890 Б (97,2% объёма кадров). Включает служебные сообщения.

PostgreSQL – JDBC

[illegible]

За сценарий: клиент получил 61 пак. / 241 239 Б, отправил 61 пак. / 4 531 Б. Всего 245 770 Б.
TCP payload: 238 890 Б (97,2% объёма кадров). Включает служебные сообщения.

PostgreSQL – ADBC

[illegible]

За сценарий: клиент получил 61 пак. / 241 239 Б, отправил 61 пак. / 4 531 Б. Всего 245 770 Б.
TCP payload: 238 890 Б (97,2% объёма кадров). Включает служебные сообщения.

PostgreSQL – ADBC

[illegible]

PostgreSQL – ADBC

ADBC не заменил pg-wire, драйвер вызвал COPY binary

Apply a display filter ... <⌘/>

No.	Time	Source	Destination	Protocol	Length	Info
94	0.059018	127.0.0.1	127.0.0.1	PGSQL	65	<R
97	0.059053	127.0.0.1	127.0.0.1	PGSQL	70	>p
99	0.065305	127.0.0.1	127.0.0.1	PGSQL	521	<R/S/S/S/S/S/S/S/S/S/S/S/S/S/S/S/K/Z
101	0.065661	127.0.0.1	127.0.0.1	PGSQL	79	>Q
103	0.065815	127.0.0.1	127.0.0.1	PGSQL	73	<C/Z
105	0.066318	127.0.0.1	127.0.0.1	PGSQL	135	>P/S
107	0.066688	127.0.0.1	127.0.0.1	PGSQL	67	<1/Z
109	0.066740	127.0.0.1	127.0.0.1	PGSQL	68	>D/S
111	0.066917	127.0.0.1	127.0.0.1	PGSQL	153	<t/T/Z
113	0.067317	127.0.0.1	127.0.0.1	PGSQL	200	>P/B/D/E/S
115	0.067611	127.0.0.1	127.0.0.1	PGSQL	294	<1/2/n/H/d/d/d/d/d/c/C/Z
117	0.074925	127.0.0.1	127.0.0.1	PGSQL	61	>X

> Frame 113: Packet, 200 bytes on wire (1600 bits), 200 bytes captured (1600 bits) on interface lo

Null/Loopback

Family: IP (2)

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 55991, Dst Port: 5432, Seq: 170, Ack: 600, Len: 144

PostgreSQL

Type: Parse

Length: 106

Statement:

Query: COPY (SELECT conf, starts_at, speaker FROM smartdata ORDER BY starts_at) TO STDOUT (FOR

Parameters: 0

PostgreSQL

Type: Bind

Length: 14

Portal:

Statement:

Parameter formats: 0

Parameter values: 0

0000 02 00 00 00 45 00 00 c4 00 00 40 00 40 06 00 00 E... ..@.@...

0010 7f 00 00 01 7f 00 00 01 da b7 15 38 24 6b 16 fa 8\$k...

0020 d0 3f 51 8f 80 18 18 e4 fe b8 00 00 01 01 08 0a .?Q.....

0030 4b 94 fb e2 2f a8 d1 43 50 00 00 00 6a 00 43 4f K.../..C P...j.CO

0040 50 59 20 28 53 45 4c 45 43 54 20 63 6f 6e 66 2c PY (SELE CT conf,

0050 20 73 74 61 72 74 73 5f 61 74 2c 20 73 70 65 61 starts_at, spea

0060 6b 65 72 20 46 52 4f 4d 20 73 6d 61 72 74 64 61 ker FROM smartda

0070 74 61 20 4f 52 44 45 52 20 42 59 20 73 74 61 72 ta ORDER BY star

0080 74 73 5f 61 74 29 20 54 4f 20 53 54 44 4f 55 54 ts_at) T O STDOUT

0090 20 28 46 4f 52 4d 41 54 20 62 69 6e 61 72 79 29 (FORMAT binary)

00a0 00 00 00 42 00 00 00 0e 00 00 00 00 00 00 00 01 ...B.....

00b0 00 01 44 00 00 00 06 50 00 45 00 00 00 09 00 00 ...D...P .E.....

00c0 00 00 00 53 00 00 00 04 ...S....

A query string. (pgsql.query), 99 bytes

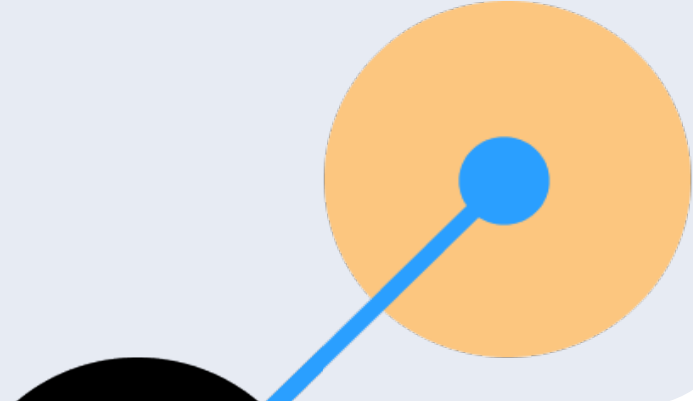
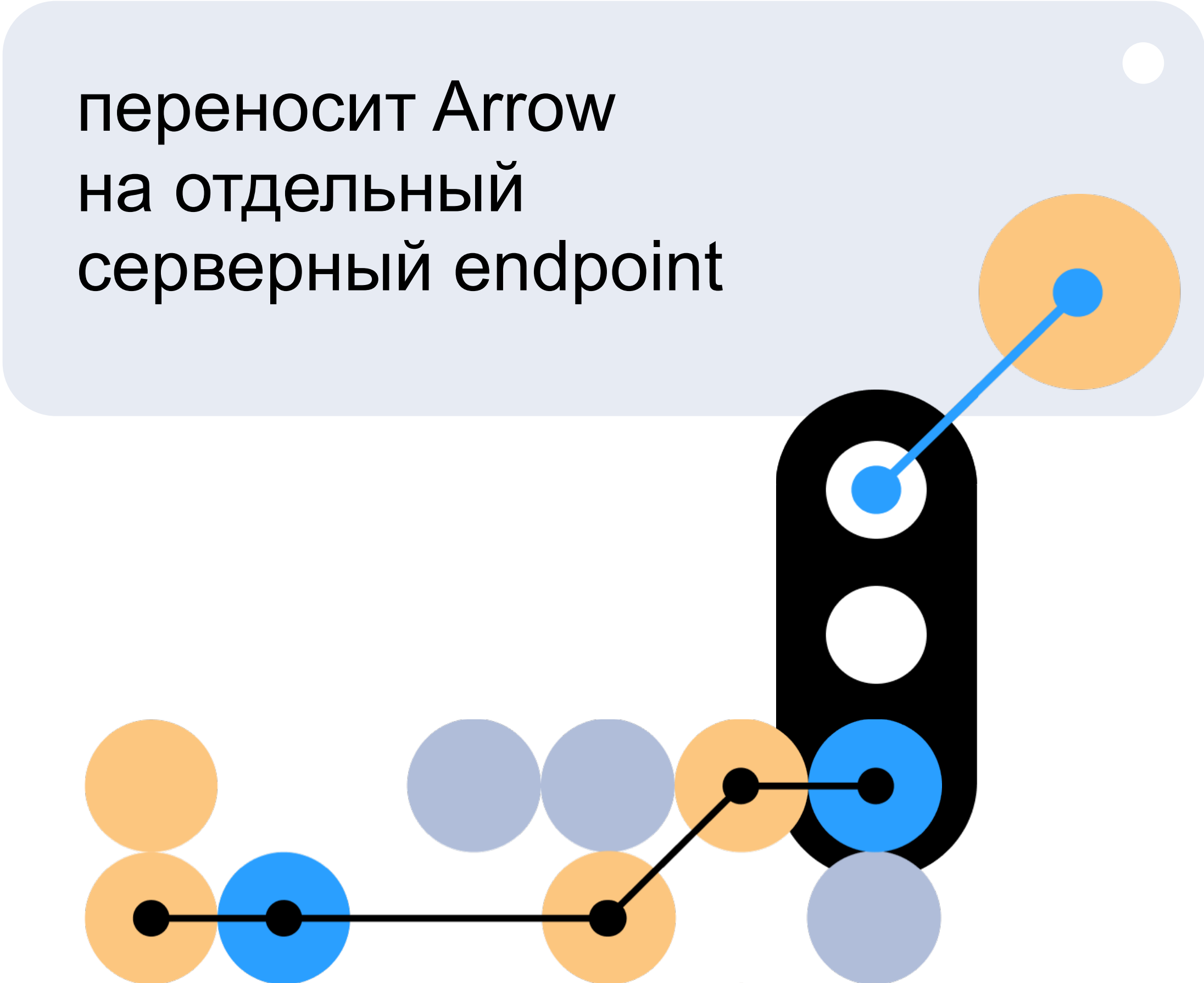
Packets: 122 · Displayed: 53 (43.4%)

Profile: Default

За сценарий: клиент получил 61 пак. / 241 239 Б, отправил 61 пак. / 4 531 Б. Всего 245 770 Б.
TCP payload: 238 890 Б (97,2% объёма кадров). Включает служебные сообщения.

37

переносит Arrow
на отдельный
серверный endpoint

A diagram illustrating a server endpoint migration. It features a black circle at the bottom left and an orange circle at the top right. A blue arrow points from the black circle to the orange circle, indicating the transfer of the 'Arrow' component to a separate server endpoint.

What is this?

Apache Arrow Flight SQL adapter for PostgreSQL is a PostgreSQL extension that adds an Apache Arrow Flight SQL endpoint to PostgreSQL.

Apache Arrow Flight SQL is a protocol to use Apache Arrow format to interact with SQL databases. You can use Apache Arrow Flight SQL instead of the PostgreSQL wire protocol to interact with PostgreSQL by Apache Arrow Flight SQL adapter for PostgreSQL.

Apache Arrow format is designed for fast typed table data exchange. If you want to get large data by **SELECT** or **INSERT/UPDATE** large data, Apache Arrow Flight SQL will be faster than the PostgreSQL wire protocol.

What is this?

Apache Arrow Flight SQL adapter for PostgreSQL is a PostgreSQL extension that adds an Apache Arrow Flight SQL endpoint to PostgreSQL.

Apache Arrow Flight SQL is a protocol to use Apache Arrow format to interact with SQL databases. You can use Apache Arrow Flight SQL instead of the PostgreSQL wire protocol to interact with PostgreSQL by Apache Arrow Flight SQL adapter for PostgreSQL.

Apache Arrow format is designed for fast typed table data exchange. If you want to get large data by **SELECT** or **INSERT/UPDATE** large data, Apache Arrow Flight SQL will be faster than the PostgreSQL wire protocol.

PostgreSQL – Flight SQL

DoGet отвечает дважды:
сначала схема Arrow, потом record batch

Current filter: http2

No.	Time	Source	Destination	Protocol	Length	Info
13	0.028621	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
15	0.028671	127.0.0.1	127.0.0.1	HTTP2	99	PING[0], WINDOW_UPDATE[1], WINDOW_UPDATE[0]
17	0.049845	127.0.0.1	127.0.0.1	HTTP2	210	HEADERS[1]: 200 OK, HEADERS[1]
19	0.051098	127.0.0.1	127.0.0.1	GRPC/...	396	HEADERS[3]: POST /arrow.flight.protocol.FlightService/GetFlightInfo, WINDOW_UPDATE[...]
21	0.063094	127.0.0.1	127.0.0.1	GRPC/...	748	HEADERS[3]: 200 OK, DATA[3] (GRPC) (PROTOBUF), HEADERS[3], WINDOW_UPDATE[0]
23	0.063180	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
25	0.063470	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
27	0.063907	127.0.0.1	127.0.0.1	GRPC/...	365	HEADERS[5]: POST /arrow.flight.protocol.FlightService/DoGet, DATA[5] (GRPC) (PROTOB...
29	0.064791	127.0.0.1	127.0.0.1	GRPC/...	369	HEADERS[5]: 200 OK, DATA[5] (GRPC) (PROTOBUF), WINDOW_UPDATE[0]
31	0.065498	127.0.0.1	127.0.0.1	GRPC/...	436	DATA[5] (GRPC) (PROTOBUF)
33	0.065848	127.0.0.1	127.0.0.1	HTTP2	66	HEADERS[5]
35	0.068482	127.0.0.1	127.0.0.1	HTTP2	82	WINDOW_UPDATE[5], WINDOW_UPDATE[0]

> Frame 27: Packet, 365 bytes on wire (2920 bits), 365 bytes captured (2920 bits) on interface lo0

Null/Loopback

Family: IP (2)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 56022, Dst Port: 15436, Seq: 836, Ack: 936, Len: 309

Source Port: 56022

Destination Port: 15436

[Stream index: 0]

[Stream Packet Number: 27]

> [Conversation completeness: Complete, WITH_DATA (31)]

[TCP Segment Len: 309]

Sequence Number: 836 (relative sequence number)

Sequence Number (raw): 49404273

[Next Sequence Number: 1145 (relative sequence number)]

Acknowledgment Number: 936 (relative ack number)

Acknowledgment number (raw): 4062757817

1000 = Header Length: 32 bytes (8)

> Flags: 0x018 (PSH, ACK)

Window: 6368

0030 98 68 4d 8d e7 13 7d ca 00 00 85 01 04 00 00 00 ·hM...}·

0040 05 40 05 3a 70 61 74 68 2a 2f 61 72 72 6f 77 2e ·@:path */arrow.

0050 66 6c 69 67 68 74 2e 70 72 6f 74 6f 63 6f 6c 2e flight.p rotocol.

0060 46 6c 69 67 68 74 53 65 72 76 69 63 65 2f 44 6f FlightSe rvice/Do

0070 47 65 74 c6 83 86 c5 c4 c3 bf c1 00 0d 61 75 74 Get..... aut

0080 68 6f 72 69 7a 61 74 69 6f 6e 1b 42 65 61 72 65 horizati on·Beare

0090 72 20 31 33 30 34 34 38 33 30 34 39 30 30 32 32 r 130448 30490022

00a0 37 34 36 31 30 30 00 15 78 2d 66 6c 69 67 68 74 746100· x-flight

00b0 2d 73 71 6c 2d 64 61 74 61 62 61 73 65 08 77 6f -sql-dat abase·wo

00c0 72 6b 73 68 6f 70 00 00 91 00 01 00 00 00 05 00 rkshop·

00d0 00 00 00 8c 0a 89 01 0a 42 74 79 70 65 2e 67 6f Btype.go

00e0 6f 67 6c 65 61 70 69 73 2e 63 6f 6d 2f 61 72 72 ogleapis .com/arr

00f0 6f 77 2e 66 6c 69 67 68 74 2e 70 72 6f 74 6f 63 ow.fligh t.protoc

0100 6f 6c 2e 73 71 6c 2e 54 69 63 6b 65 74 53 74 61 ol.sql.T icketSta

0110 74 65 6d 65 6e 74 51 75 65 72 79 12 43 0a 41 53 tementQu ery·C·AS

0120 45 4c 45 43 54 20 63 6f 6e 66 2c 20 73 74 61 72 ELECT co nf, star

0130 74 73 5f 61 74 2c 20 73 70 65 61 6b 65 72 20 46 ts_at, s peaker F

0140 52 4f 4d 20 73 6d 61 72 74 64 61 74 61 20 4f 52 ROM smar tdata OR

0150 44 45 52 20 42 59 20 73 74 61 72 74 73 5f 61 74 DER BY s tarts_at

0160 00 00 04 08 00 00 00 00 00 00 00 02 2d

Packet (365 bytes)Decompressed Header (403 bytes)

The wire type determines value format (protobuf.field.value), 137 bytes

Packets: 40 · Displayed: 16 (40.0%)

Profile: Default

За сценарий: клиент получил 20 пак. / 2 770 Б, отправил 20 пак. / 2 302 Б. Всего 5 072 Б.
TCP payload: 2 808 Б (55,4% объема кадров). Включает служебные сообщения.

39

PostgreSQL – Flight SQL

Буфером получаем данные

Current filter: http2

No.	Time	Source	Destination	Protocol	Length	Info
13	0.028621	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
15	0.028671	127.0.0.1	127.0.0.1	HTTP2	99	PING[0], WINDOW_UPDATE[1], WINDOW_UPDATE[0]
17	0.049845	127.0.0.1	127.0.0.1	HTTP2	210	HEADERS[1]: 200 OK, HEADERS[1]
19	0.051098	127.0.0.1	127.0.0.1	GRPC/...	396	HEADERS[3]: POST /arrow.flight.protocol.FlightService/GetFlightInfo, WINDOW_UPDATE[...]
21	0.063094	127.0.0.1	127.0.0.1	GRPC/...	748	HEADERS[3]: 200 OK, DATA[3] (GRPC) (PROTOBUF), HEADERS[3], WINDOW_UPDATE[0]
23	0.063180	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
25	0.063470	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
27	0.063907	127.0.0.1	127.0.0.1	GRPC/...	365	HEADERS[5]: POST /arrow.flight.protocol.FlightService/DoGet, DATA[5] (GRPC) (PROTOB...
29	0.064791	127.0.0.1	127.0.0.1	GRPC/...	369	HEADERS[5]: 200 OK, DATA[5] (GRPC) (PROTOBUF), WINDOW_UPDATE[0]
31	0.065498	127.0.0.1	127.0.0.1	GRPC/...	436	DATA[5] (GRPC) (PROTOBUF)
33	0.065848	127.0.0.1	127.0.0.1	HTTP2	66	HEADERS[5]
35	0.068482	127.0.0.1	127.0.0.1	HTTP2	82	WINDOW_UPDATE[5], WINDOW_UPDATE[0]

TCP payload (380 bytes)
[PDU Size: 380]
> HyperText Transfer Protocol 2
> GRPC Message: /arrow.flight.protocol.FlightService/DoGet, Response
✓ Protocol Buffers: /arrow.flight.protocol.FlightService/DoGet,response
 v Message: <UNKNOWN>
 v Field(2):
 [Field Name: <UNKNOWN>]
 .001 0... = Field Number: 2
 010 = Wire Type: Length-delimited (2)
 Value Length: 248
 Value [...]: 14000000000000000c0016000600050008000c000c000000000304001800000070000000000000
 v Field(1000):
 [Field Name: <UNKNOWN>]
 .001 1111 .100 0... = Field Number: 1000
 010 = Wire Type: Length-delimited (2)
 Value Length: 112
 Value: 0000000009000000120000001b0000002400000000000000536d61727444617461536d617274446174

```

0060 00 00 03 04 00 18 00 00 00 70 00 00 00 00 00 00 ..... p.....
0070 00 00 00 0a 00 18 00 0c 00 04 00 08 00 0a 00 00 ..... 
0080 00 8c 00 00 00 10 00 00 00 04 00 00 00 00 00 00 ..... 
0090 00 00 00 00 00 07 00 00 00 00 00 00 00 00 00 00 ..... 
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 
00b0 00 14 00 00 00 00 00 00 00 18 00 00 00 00 00 00 ..... 
00c0 00 24 00 00 00 00 00 00 00 40 00 00 00 00 00 00 $. .... @.....
00d0 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 ..... @.....
00e0 00 20 00 00 00 00 00 00 00 60 00 00 00 00 00 00 ..... `.....
00f0 00 00 00 00 00 00 00 00 00 60 00 00 00 00 00 00 ..... ^.....
0100 00 10 00 00 00 00 00 00 00 00 00 00 00 03 00 00 ..... 
0110 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 
0120 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 
0130 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 
0140 00 c2 3e 70 00 00 00 00 09 00 00 00 12 00 00 00 ..>p.... 
0150 1b 00 00 00 24 00 00 00 00 00 00 00 53 6d 61 72 ....$. ... Smart
0160 74 44 61 74 61 53 6d 61 72 74 44 61 74 61 53 6d tDataSma rtDataSm
0170 61 72 74 44 61 74 61 53 6d 61 72 74 44 61 74 61 artDataS martData
0180 00 00 00 00 00 ec ed 75 3b 05 06 00 00 91 b5 e9 .....u ;.....
0190 98 21 06 00 00 a8 db 65 66 40 06 00 00 da 20 f7 .!.....e f@....
01a0 23 5c 06 00 00 00 00 01 00 00 00 00 00 00 00 00 #\.....
01b0 01 00 00 00 .....

```

The wire type determines value format (protobuf.field.value), 112 bytes Packets: 40 · Displayed: 16 (40.0%) Profile: Default

За сценарий: клиент получил 20 пак. / 2 770 Б, отправил 20 пак. / 2 302 Б. Всего 5 072 Б.
TCP payload: 2 808 Б (55,4% объёма кадров). Включает служебные сообщения.

Один SELECT в PostgreSQL

Способ	Что летит по сети	Где собирается конечный результат
Simple / Extended	DataRow	Драйвер
COPY	COPY	Драйвер
ADBC PostgreSQL	COPY binary	Драйвер ADBC
Flight SQL	Arrow IPC	Сериализует результат

Протокол не всегда определяет, что именно будет передаваться по сети

**Что летит по сети
в ClickHouse®
проверим wireshark**

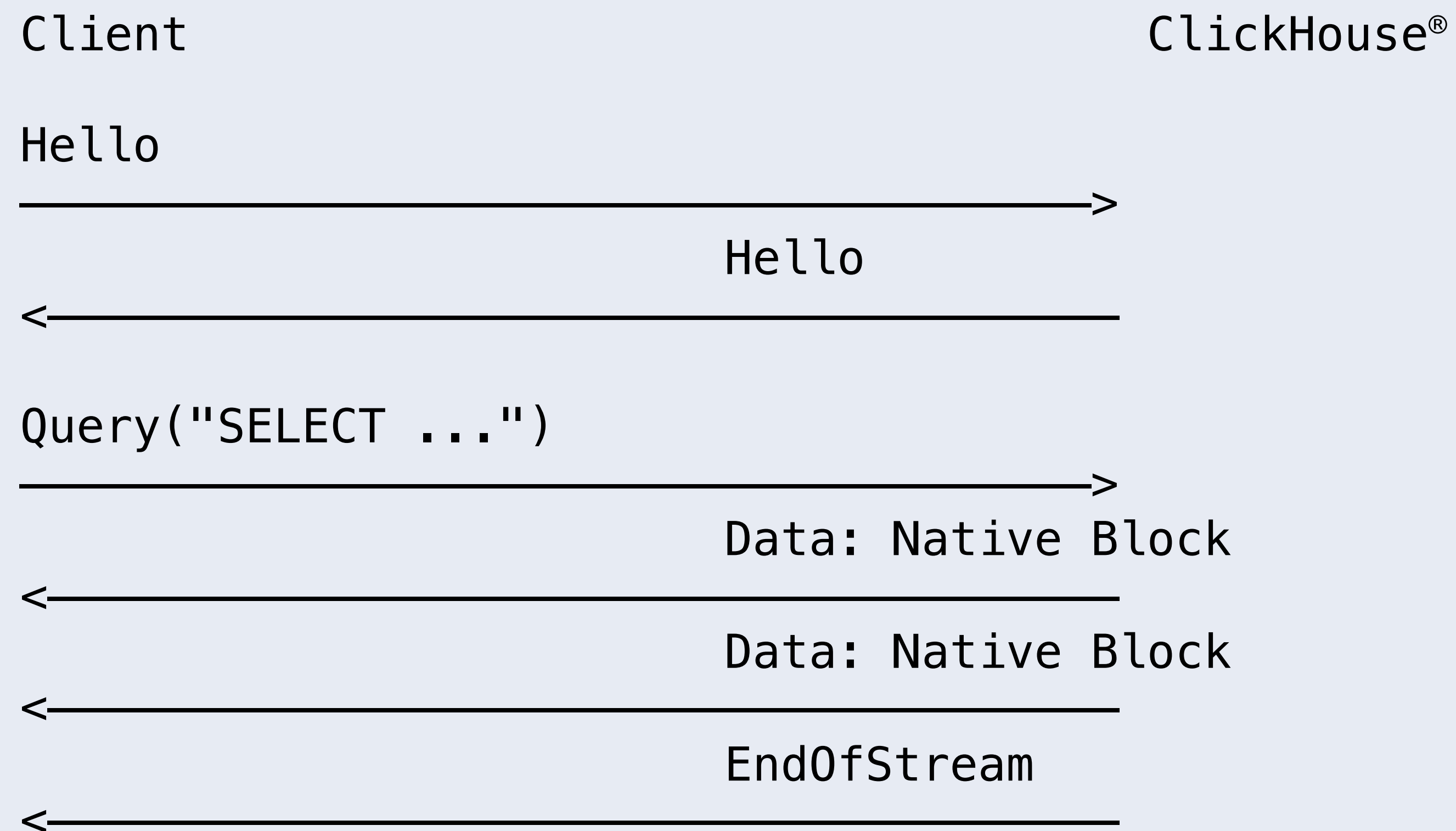
ClickHouse® → результат запроса

- Native TCP
 - └ бинарные колоночные Blocks
- HTTP – формат тела ответа выбирается через FORMAT
 - └ Native / ArrowStream / Parquet (колоночные данные)
 - └ RowBinary (бинарные строки)
 - └ TSV / CSV / JSONEachRow (текстовые строки)
 - └ JSON / JSONColumns (ответ целиком / по колонкам)
- Flight SQL
 - └ gRPC + Arrow IPC: схема и batches
- pg-wire compatibility
 - └ RowDescription + text DataRow
- ...

ClickHouse® → результат запроса

- Native TCP
 - └ бинарные колоночные Blocks
- HTTP – формат тела ответа выбирается через FORMAT
 - └ Native / ArrowStream / Parquet (колоночные данные)
 - └ RowBinary (бинарные строки)
 - └ TSV / CSV / JSONEachRow (текстовые строки)
 - └ JSON / JSONColumns (ответ целиком / по колонкам)
- Flight SQL
 - └ gRPC + Arrow IPC: схема и batches
- pg-wire compatibility
 - └ RowDescription + text DataRow
- ...

ClickHouse® – Native TCP



ClickHouse® - Native TCP

Apply a display filter ... <⌘/>

No.	Time	Source	Destination	Protocol	Length	Info
15	0.005560	127.0.0.1	127.0.0.1	TCP	221	56036 → 19000 [PSH, ACK] Seq=69 Ack=72 Win=408256 Len=165 TSval=3054924118 TSecr=65...
16	0.005573	127.0.0.1	127.0.0.1	TCP	56	19000 → 56036 [ACK] Seq=72 Ack=234 Win=408192 Len=0 TSval=658969859 TSecr=3054924118
17	0.008508	127.0.0.1	127.0.0.1	TCP	130	19000 → 56036 [PSH, ACK] Seq=72 Ack=234 Win=408192 Len=74 TSval=658969862 TSecr=305...
18	0.008547	127.0.0.1	127.0.0.1	TCP	56	56036 → 19000 [ACK] Seq=234 Ack=146 Win=408192 Len=0 TSval=3054924121 TSecr=6589698...
19	0.008908	127.0.0.1	127.0.0.1	TCP	228	19000 → 56036 [PSH, ACK] Seq=146 Ack=234 Win=408192 Len=172 TSval=658969862 TSecr=3...
20	0.008927	127.0.0.1	127.0.0.1	TCP	56	56036 → 19000 [ACK] Seq=234 Ack=318 Win=408064 Len=0 TSval=3054924121 TSecr=6589698...
21	0.008970	127.0.0.1	127.0.0.1	TCP	67	19000 → 56036 [PSH, ACK] Seq=318 Ack=234 Win=408192 Len=11 TSval=658969862 TSecr=30...
22	0.008989	127.0.0.1	127.0.0.1	TCP	56	56036 → 19000 [ACK] Seq=234 Ack=329 Win=408064 Len=0 TSval=3054924121 TSecr=6589698...
23	0.009097	127.0.0.1	127.0.0.1	TCP	2304	19000 → 56036 [PSH, ACK] Seq=329 Ack=234 Win=408192 Len=2248 TSval=658969863 TSecr=...
24	0.009117	127.0.0.1	127.0.0.1	TCP	56	56036 → 19000 [ACK] Seq=234 Ack=2577 Win=405824 Len=0 TSval=3054924122 TSecr=658969...
25	0.009364	127.0.0.1	127.0.0.1	TCP	57	19000 → 56036 [PSH, ACK] Seq=2577 Ack=234 Win=408192 Len=1 TSval=658969863 TSecr=30...
26	0.009381	127.0.0.1	127.0.0.1	TCP	56	56036 → 19000 [ACK] Seq=234 Ack=2578 Win=405824 Len=0 TSval=3054924122 TSecr=658969...

> Frame 15: Packet, 221 bytes on wire (1768 bits), 221 bytes captured (1768 bits) on interface lo0

Null/Loopback

Family: IP (2)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 56036, Dst Port: 19000, Seq: 69, Ack: 72, Len: 165

Source Port: 56036

Destination Port: 19000

[Stream index: 0]

[Stream Packet Number: 15]

> [Conversation completeness: Complete, WITH_DATA (31)]

[TCP Segment Len: 165]

Sequence Number: 69 (relative sequence number)

Sequence Number (raw): 948826631

[Next Sequence Number: 234 (relative sequence number)]

Acknowledgment Number: 72 (relative ack number)

Acknowledgment number (raw): 2548125933

1000 = Header Length: 32 bytes (8)

> Flags: 0x018 (PSH, ACK)

Window: 6270

0000 02 00 00 00 45 00 00 d9 00 00 40 00 40 06 00 00 E... ..@.@...

0010 7f 00 00 01 7f 00 00 01 da e4 4a 38 38 8d f2 07 J88...

0020 97 e1 50 ed 80 18 18 eb fe cd 00 00 01 01 08 0a ..P.....

0030 b6 16 71 56 27 47 15 03 01 00 01 00 00 09 30 2e ..qV'G... ..0.

0040 30 2e 30 2e 30 3a 30 00 00 00 00 00 00 00 00 01 0.0.0:0.

0050 04 72 6f 6f 74 0c 63 35 35 64 36 32 30 62 32 63 ..root.c5 5d620b2c

0060 66 64 11 43 6c 69 63 6b 48 6f 75 73 65 20 63 6c fd·Click House cl

0070 69 65 6e 74 1a 07 d8 a9 03 00 00 05 00 00 00 00 ient....

0080 01 01 00 00 00 00 00 01 00 00 02 00 41 53 45 4c ASEL

0090 45 43 54 20 63 6f 6e 66 2c 20 73 74 61 72 74 73 ECT conf , starts

00a0 5f 61 74 2c 20 73 70 65 61 6b 65 72 20 46 52 4f _at, spe aker FRO

00b0 4d 20 73 6d 61 72 74 64 61 74 61 20 4f 52 44 45 M smartd ata ORDE

00c0 52 20 42 59 20 73 74 61 72 74 73 5f 61 74 00 02 R BY sta rts_at..

00d0 00 01 00 02 ff ff ff ff 03 00 00 00 00

Data (data.data), 165 bytes

Packets: 30

Profile: Default

За сценарий: клиент получил 15 пак. / 3 429 Б, отправил 15 пак. / 1 085 Б. Всего 4 514 Б.
TCP payload: 2 810 Б (62,3% объема кадров). Включает служебные сообщения.

46

ClickHouse® - Native TCP

В payload подряд лежат четыре метки времени
следом четыре Int32 колонки speaker

Apply a display filter ... <36/>

No.	Time	Source	Destination	Protocol	Length	Info
10	0.004471	127.0.0.1	127.0.0.1	TCP	56	19000 → 56036 [ACK] Seq=71 Ack=68 Win=408320 Len=0 TSval=658969858 TSecr=3054924117
11	0.004967	127.0.0.1	127.0.0.1	TCP	57	56036 → 19000 [PSH, ACK] Seq=68 Ack=71 Win=408256 Len=1 TSval=3054924117 TSecr=6589...
12	0.004986	127.0.0.1	127.0.0.1	TCP	56	19000 → 56036 [ACK] Seq=71 Ack=69 Win=408320 Len=0 TSval=658969858 TSecr=3054924117
13	0.005137	127.0.0.1	127.0.0.1	TCP	57	19000 → 56036 [PSH, ACK] Seq=71 Ack=69 Win=408320 Len=1 TSval=658969859 TSecr=30549...
14	0.005153	127.0.0.1	127.0.0.1	TCP	56	56036 → 19000 [ACK] Seq=69 Ack=72 Win=408256 Len=0 TSval=3054924118 TSecr=658969859
15	0.005560	127.0.0.1	127.0.0.1	TCP	221	56036 → 19000 [PSH, ACK] Seq=69 Ack=72 Win=408256 Len=165 TSval=3054924118 TSecr=65...
16	0.005573	127.0.0.1	127.0.0.1	TCP	56	19000 → 56036 [ACK] Seq=72 Ack=234 Win=408192 Len=0 TSval=658969859 TSecr=3054924118
17	0.008508	127.0.0.1	127.0.0.1	TCP	130	19000 → 56036 [PSH, ACK] Seq=72 Ack=234 Win=408192 Len=74 TSval=658969862 TSecr=305...
18	0.008547	127.0.0.1	127.0.0.1	TCP	56	56036 → 19000 [ACK] Seq=234 Ack=146 Win=408192 Len=0 TSval=3054924121 TSecr=6589698...
19	0.008908	127.0.0.1	127.0.0.1	TCP	228	19000 → 56036 [PSH, ACK] Seq=146 Ack=234 Win=408192 Len=172 TSval=658969862 TSecr=3...
20	0.008927	127.0.0.1	127.0.0.1	TCP	56	56036 → 19000 [ACK] Seq=234 Ack=318 Win=408064 Len=0 TSval=3054924121 TSecr=6589698...
21	0.008970	127.0.0.1	127.0.0.1	TCP	67	19000 → 56036 [PSH, ACK] Seq=318 Ack=234 Win=408192 Len=11 TSval=658969862 TSecr=30...

> Frame 19: Packet, 228 bytes on wire (1824 bits), 228 bytes captured (1824 bits) on interface lo0

Null/Loopback

Family: IP (2)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 19000, Dst Port: 56036, Seq: 146, Ack: 234, Len: 172

Source Port: 19000

Destination Port: 56036

[Stream index: 0]

[Stream Packet Number: 19]

> [Conversation completeness: Complete, WITH_DATA (31)]

[TCP Segment Len: 172]

Sequence Number: 146 (relative sequence number)

Sequence Number (raw): 2548126007

[Next Sequence Number: 318 (relative sequence number)]

Acknowledgment Number: 234 (relative ack number)

Acknowledgment number (raw): 948826796

1000 = Header Length: 32 bytes (8)

> Flags: 0x018 (PSH, ACK)

Window: 6270

0000 02 00 00 00 45 00 00 e0 00 00 40 00 40 06 00 00 ...E...@.@...

0010 7f 00 00 01 7f 00 00 01 4a 38 da e4 97 e1 51 37 J8....Q7

0020 38 8d f2 ac 80 18 18 ea fe d4 00 00 01 01 08 0a 8.....

0030 27 47 15 06 b6 16 71 59 01 00 01 00 02 ff ff ff 'G....qY.....

0040 ff 03 00 00 03 04 04 63 6f 6e 66 06 53 74 72 69 c onf·Stri

0050 6e 67 00 09 53 6d 61 72 74 44 61 74 61 09 53 6d ng··Smar tData·Sm

0060 61 72 74 44 61 74 61 09 53 6d 61 72 74 44 61 74 artData· SmartDat

0070 61 09 53 6d 61 72 74 44 61 74 61 09 73 74 61 72 a·SmartD ata·star

0080 74 73 5f 61 74 14 44 61 74 65 54 69 6d 65 36 34 ts_at·Da teTime64

0090 28 36 2c 20 27 55 54 43 27 29 00 00 ec ed 75 3b (6, 'UTC ')....u;

00a0 05 06 00 00 91 b5 e9 98 21 06 00 00 a8 db 65 66!.....ef

00b0 40 06 00 00 da 20 f7 23 5c 06 00 07 73 70 65 61 @....·# \...spea

00c0 6b 65 72 05 49 6e 74 33 32 00 00 00 00 00 01 00 ker·Int3 2.....

00d0 00 00 00 00 00 00 01 00 00 00 06 04 01 80 08 00

00e0 00 01 00 00

A data segment used in reassembly of an upper-layer protocol (ULP) (tcp.segment_data), 172 bytes

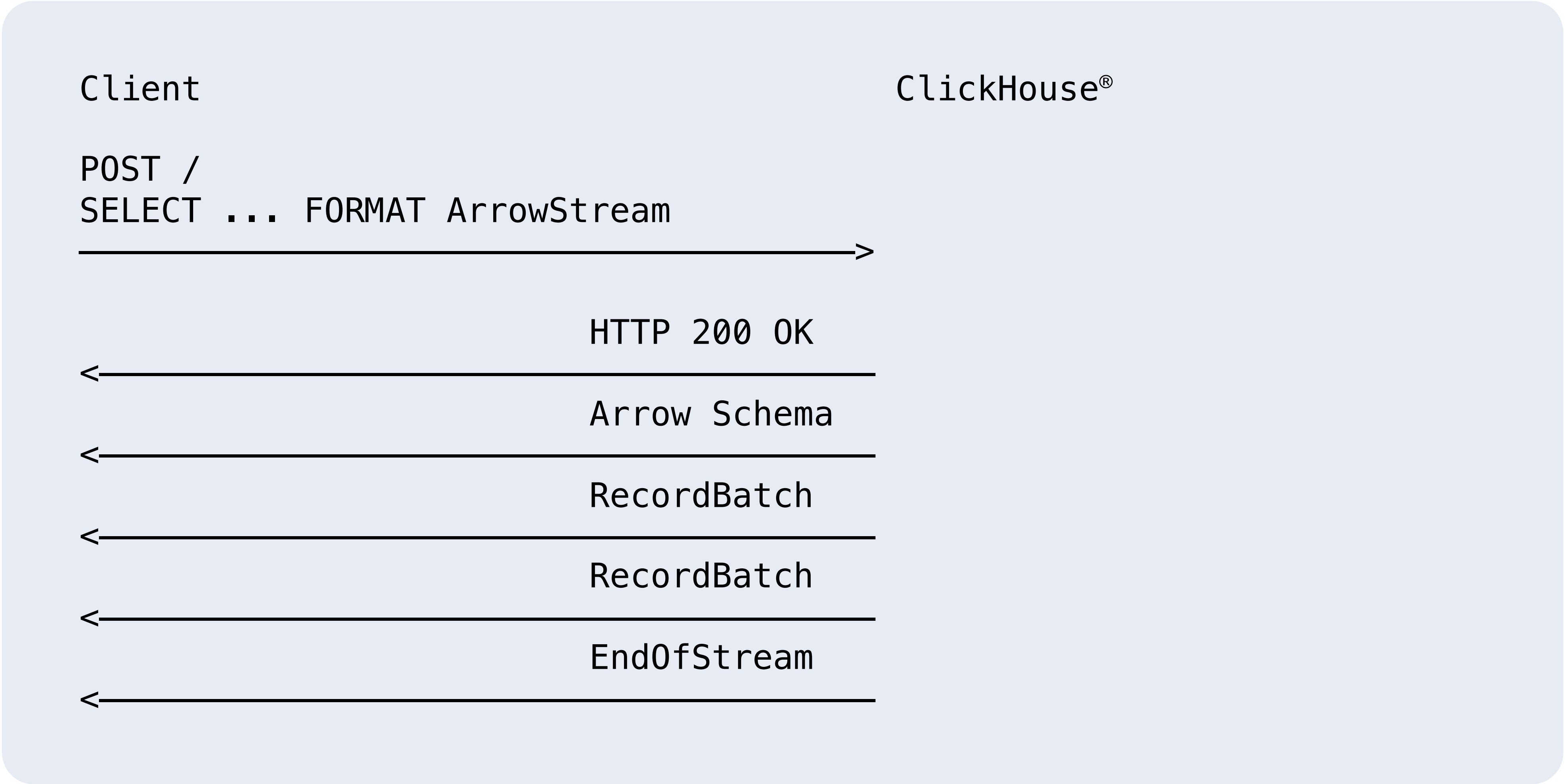
Packets: 30

Profile: Default

За сценарий: клиент получил 15 пак. / 3 429 Б, отправил 15 пак. / 1 085 Б. Всего 4 514 Б.
TCP payload: 2 810 Б (62,3% объема кадров). Включает служебные сообщения.

47

ClickHouse® – HTTP – ADBC



ClickHouse® HTTP – ADBC использует HTTP и запрашивает ArrowStream

Apply a display filter ... <⌘/>

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000177	127.0.0.1	127.0.0.1	TCP	56	56088 → 18123 [ACK] Seq=1 Ack=1 Win=408320 Len=0 TSval=1532860586 TSecr=33723881
4	0.000194	127.0.0.1	127.0.0.1	TCP	296	56088 → 18123 [PSH, ACK] Seq=1 Ack=1 Win=408320 Len=240 TSval=1532860586 TSecr=3372...
5	0.000205	127.0.0.1	127.0.0.1	TCP	56	[TCP Window Update] 18123 → 56088 [ACK] Seq=1 Ack=1 Win=408320 Len=0 TSval=33723881...
6	0.000206	127.0.0.1	127.0.0.1	HTTP	140	POST /?enable_http_compression=0 HTTP/1.1
7	0.000227	127.0.0.1	127.0.0.1	TCP	56	18123 → 56088 [ACK] Seq=1 Ack=241 Win=408128 Len=0 TSval=33723881 TSecr=1532860586
8	0.000241	127.0.0.1	127.0.0.1	TCP	56	18123 → 56088 [ACK] Seq=1 Ack=325 Win=408064 Len=0 TSval=33723881 TSecr=1532860586
9	0.013642	127.0.0.1	127.0.0.1	TCP	1540	18123 → 56088 [PSH, ACK] Seq=1 Ack=325 Win=408064 Len=1484 TSval=33723894 TSecr=153...
10	0.013686	127.0.0.1	127.0.0.1	TCP	56	56088 → 18123 [ACK] Seq=325 Ack=1485 Win=406848 Len=0 TSval=1532860599 TSecr=337238...
11	0.013733	127.0.0.1	127.0.0.1	TCP	69	18123 → 56088 [PSH, ACK] Seq=1485 Ack=325 Win=408064 Len=13 TSval=33723894 TSecr=15...
12	0.013761	127.0.0.1	127.0.0.1	TCP	56	56088 → 18123 [ACK] Seq=325 Ack=1498 Win=406848 Len=0 TSval=1532860599 TSecr=337238...
13	0.013872	127.0.0.1	127.0.0.1	HTTP	61	HTTP/1.1 200 OK
14	0.013894	127.0.0.1	127.0.0.1	TCP	56	56088 → 18123 [ACK] Seq=325 Ack=1503 Win=406848 Len=0 TSval=1532860599 TSecr=337238...

TCP segment data (84 bytes)
[2 Reassembled TCP Segments (324 bytes): #4(240), #6(84)]
Hypertext Transfer Protocol
 POST /?enable_http_compression=0 HTTP/1.1\r\n
 Host: 127.0.0.1:18123\r\n
 User-Agent: python-requests/2.34.2\r\n
 Accept-Encoding: gzip, deflate\r\n
 Accept: */*\r\n
 Connection: keep-alive\r\n
 Content-Length: 84\r\n
 Authorization: Basic d29ya3Nob3A6d29ya3Nob3A=\r\n\r\n
 [Response in frame: 13]
 [Full request URI: http://127.0.0.1:18123/?enable_http_compression=0]
 File Data: 84 bytes
 Data (84 bytes)
 Data: 53454c45435420636f6e662c207374617274735f61742c20737065616b65722046524f4d20736d6172746
 [Length: 84]

0000 02 00 00 00 45 00 00 88 00 00 40 00 40 06 00 00 ...E... ..@..@...
0010 7f 00 00 01 7f 00 00 01 db 18 46 cb 99 4f a9 a1 F..0..
0020 0f b7 28 fb 80 18 18 ec fe 7c 00 00 01 01 08 0a ..(..... ..|.....
0030 5b 5d 98 aa 02 02 95 e9 53 45 4c 45 43 54 20 63 []..... SELECT c
0040 6f 6e 66 2c 20 73 74 61 72 74 73 5f 61 74 2c 20 onf, sta rts_at,
0050 73 70 65 61 6b 65 72 20 46 52 4f 4d 20 73 6d 61 speaker FROM sma
0060 72 74 64 61 74 61 20 4f 52 44 45 52 20 42 59 20 rtdata 0 RDER BY
0070 73 74 61 72 74 73 5f 61 74 20 46 4f 52 4d 41 54 starts_a t FORMAT
0080 20 41 72 72 6f 77 53 74 72 65 61 6d ArrowSt ream

Packet (140 bytes) Reassembled TCP (324 bytes) Basic Credentials (17 bytes)

capture.pcapng Packets: 18 Profile: Default

За сценарий: клиент получил 9 пак. / 2 018 Б, отправил 9 пак. / 840 Б. Всего 2 858 Б.
TCP payload: 1 826 Б (63,9% объема кадров). Включает служебные сообщения.

49

ClickHouse® HTTP – ADBC использует HTTP и запрашивает ArrowStream

ответ приходит как Arrow IPC

Apply a display filter ... <⌘/>

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000177	127.0.0.1	127.0.0.1	TCP	56	56088 → 18123 [ACK] Seq=1 Ack=1 Win=408320 Len=0 TSval=1532860586 TSecr=33723881
4	0.000194	127.0.0.1	127.0.0.1	TCP	296	56088 → 18123 [PSH, ACK] Seq=1 Ack=1 Win=408320 Len=240 TSval=1532860586 TSecr=3372...
5	0.000205	127.0.0.1	127.0.0.1	TCP	56	[TCP Window Update] 18123 → 56088 [ACK] Seq=1 Ack=1 Win=408320 Len=0 TSval=33723881...
6	0.000206	127.0.0.1	127.0.0.1	HTTP	140	POST /?enable_http_compression=0 HTTP/1.1
7	0.000227	127.0.0.1	127.0.0.1	TCP	56	18123 → 56088 [ACK] Seq=1 Ack=241 Win=408128 Len=0 TSval=33723881 TSecr=1532860586
8	0.000241	127.0.0.1	127.0.0.1	TCP	56	18123 → 56088 [ACK] Seq=1 Ack=325 Win=408064 Len=0 TSval=33723881 TSecr=1532860586
9	0.013642	127.0.0.1	127.0.0.1	TCP	1540	18123 → 56088 [PSH, ACK] Seq=1 Ack=325 Win=408064 Len=1484 TSval=33723894 TSecr=153...
10	0.013686	127.0.0.1	127.0.0.1	TCP	56	56088 → 18123 [ACK] Seq=325 Ack=1485 Win=406848 Len=0 TSval=1532860599 TSecr=337238...
11	0.013733	127.0.0.1	127.0.0.1	TCP	69	18123 → 56088 [PSH, ACK] Seq=1485 Ack=325 Win=408064 Len=13 TSval=33723894 TSecr=15...
12	0.013761	127.0.0.1	127.0.0.1	TCP	56	56088 → 18123 [ACK] Seq=325 Ack=1498 Win=406848 Len=0 TSval=1532860599 TSecr=337238...
13	0.013872	127.0.0.1	127.0.0.1	HTTP	61	HTTP/1.1 200 OK
14	0.013894	127.0.0.1	127.0.0.1	TCP	56	56088 → 18123 [ACK] Seq=325 Ack=1503 Win=406848 Len=0 TSval=1532860599 TSecr=337238...

Content-Type: application/octet-stream\r\n
Access-Control-Expose-Headers: X-ClickHouse-Query-Id,X-ClickHouse-Summary,X-ClickHouse-Server-
X-ClickHouse-Server-Display-Name: e32daa6288e0\r\n
Transfer-Encoding: chunked\r\n
X-ClickHouse-Query-Id: 9c1b3765-a093-4380-81be-0ce84f0867fd\r\n
X-ClickHouse-Format: ArrowStream\r\n
X-ClickHouse-Timezone: UTC\r\n
Keep-Alive: timeout=30, max=9999\r\n
X-ClickHouse-Exception-Tag: zxztzpsekjoujpx\r\n
\r\n
[Request in frame: 6]
[Time since request: 13.666000 milliseconds]
[Request URI: /?enable_http_compression=0]
[Full request URI: http://127.0.0.1:18123/?enable_http_compression=0]
HTTP chunked response
Data chunk (664 octets)
Chunk size: 664 octets
Chunk data [...]: ffffffff00000001000000000000a000c000600050008000a000000000104001000000000

0330 0d 0a ff ff ff ff f0 00 00 00 10 00 00 00 00 00 ..
0340 0a 00 0c 00 06 00 05 00 08 00 0a 00 00 00 00 01
0350 04 00 10 00 00 00 0c 00 0c 00 00 00 04 00 00 00
0360 08 00 0c 00 00 00 18 00 00 00 04 00 00 00 01 00
0370 00 00 02 00 00 00 00 00 00 00 00 00 00 00 03 00
0380 00 00 84 00 00 00 38 00 00 00 04 00 00 00 94 ff
0390 ff ff 00 00 00 02 08 00 00 00 18 00 00 00 07 00
03a0 00 00 73 70 65 61 6b 65 72 00 08 00 0c 00 08 00
03b0 07 00 08 00 00 00 00 00 00 01 20 00 00 00 c4 ff
03c0 ff ff 00 00 00 0a 08 00 00 00 1c 00 00 00 09 00
03d0 00 00 73 74 61 72 74 73 5f 61 74 00 00 00 08 00
03e0 0c 00 06 00 08 00 08 00 00 00 00 00 02 00 04 00
03f0 00 00 03 00 00 00 55 54 43 00 0c 00 10 00 08 00
0400 00 00 07 00 0c 00 0c 00 00 00 00 00 00 05 08 00
0410 00 00 14 00 00 00 04 00 00 00 63 6f 6e 66 00 00
0420 00 00 04 00 04 00 04 00 00 00 ff ff ff ff 08 01
0430 00 00 14 00 00 00 00 00 00 00 0c 00 18 00 06 00
0440 05 00 08 00 0c 00 0c 00 00 00 00 03 04 00 1c 00
0450 00 00 90 00 00 00 00 00 00 00 00 00 00 00 0c 00
0460 1c 00 10 00 04 00 08 00 0c 00 0c 00 00 00 90 00

Packet (61 bytes) Reassembled TCP (1502 bytes) De-chunked entity body (672 bytes)

Chunk data (http.chunk_data), 664 bytes Packets: 18 Profile: Default

За сценарий: клиент получил 9 пак. / 2 018 Б, отправил 9 пак. / 840 Б. Всего 2 858 Б.
TCP payload: 1 826 Б (63,9% объема кадров). Включает служебные сообщения.

50

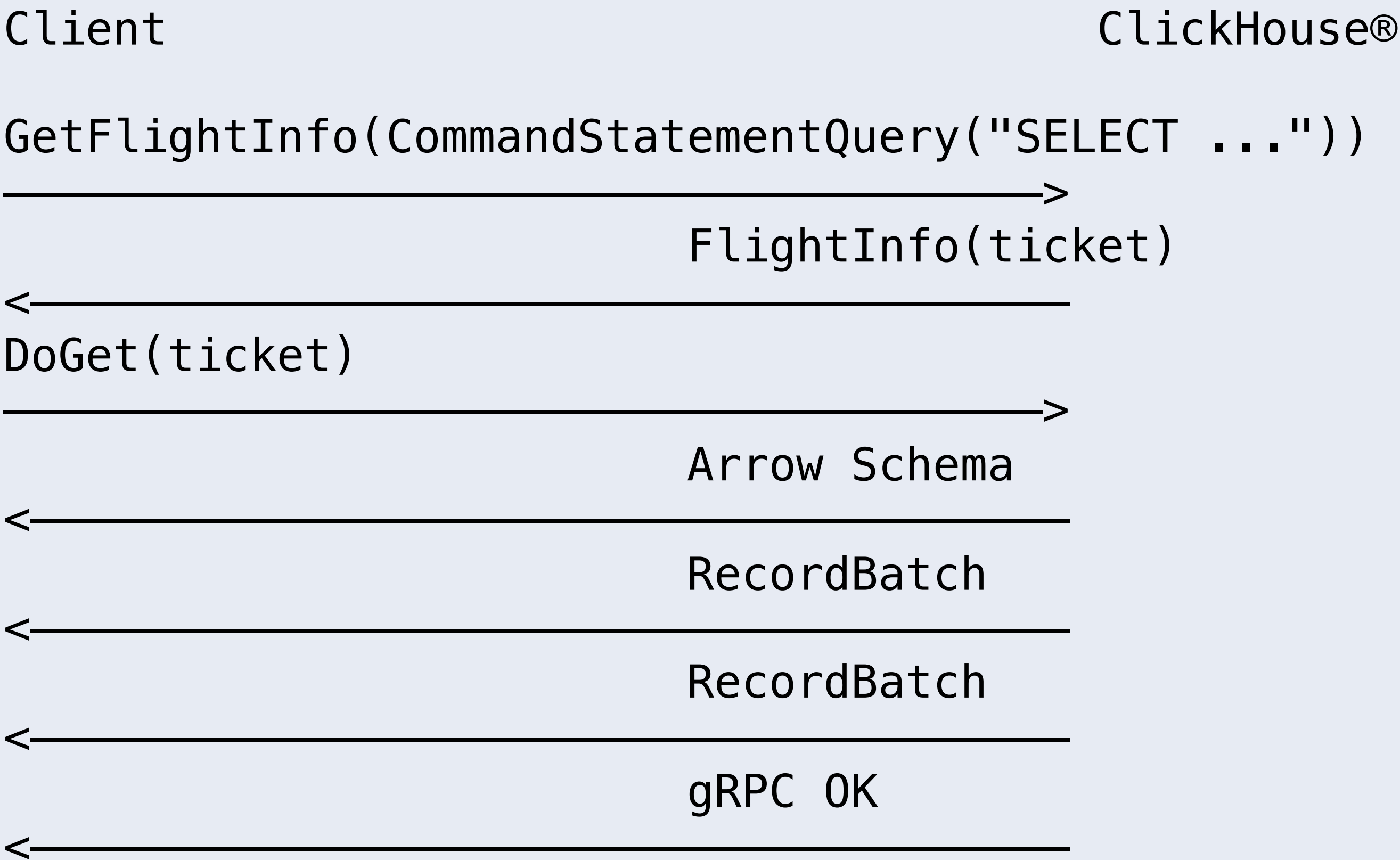
Драйверы ADBC

разные пути до Arrow

Участок	ADBC PostgreSQL	ADBC ClickHouse®	ADBC Flight SQL
Клиентский API	ADBC		
Драйвер	PostgreSQLdriver + libpq	ClickHouse® ADBC driver	Flight SQL driver
Серверная сторона	Обычный PostgreSQL	ClickHouse® HTTP endpoint	Flight SQL endpoint в БД или адаптере
Протокол / транспорт	pg-wire / TCP	HTTP(S)	Flight SQL / gRPC
Формат на сети	COPY binary → CopyData*	ArrowStream → Arrow IPC	Arrow IPC batches
Где появляется Arrow	Собирает клиентский драйвер	Сериализует ClickHouse®	Сериализует Flight SQL endpoint
Результат ADBC	Поток Arrow batches		

Протокол не всегда определяет, что именно будет передавать по сети

ClickHouse® – Flight SQL



ClickHouse® Flight SQL

GetFlightInfo -> ticket -> DoGet -> Arrow schema и batch

Current filter: http2

No.	Time	Source	Destination	Protocol	Length	Info
5	0.001526	127.0.0.1	127.0.0.1	HTTP2	138	Magic, SETTINGS[0], WINDOW_UPDATE[0]
7	0.012057	127.0.0.1	127.0.0.1	HTTP2	102	SETTINGS[0], WINDOW_UPDATE[0]
9	0.012584	127.0.0.1	127.0.0.1	GRPC/...	555	SETTINGS[0], HEADERS[1]: POST /arrow.flight.protocol.FlightService/GetFlightInfo, W...
11	0.015609	127.0.0.1	127.0.0.1	HTTP2	95	SETTINGS[0], WINDOW_UPDATE[0], PING[0]
13	0.015661	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
15	0.032771	127.0.0.1	127.0.0.1	GRPC/...	773	HEADERS[1]: 200 OK, DATA[1] (GRPC) (PROTOBUF), HEADERS[1]
17	0.032929	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
19	0.033233	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
21	0.033739	127.0.0.1	127.0.0.1	GRPC/...	242	HEADERS[3]: POST /arrow.flight.protocol.FlightService/DoGet, DATA[3] (GRPC) (PROTOB...
23	0.034765	127.0.0.1	127.0.0.1	GRPC/...	401	HEADERS[3]: 200 OK, DATA[3] (GRPC) (PROTOBUF), WINDOW_UPDATE[0]
25	0.034947	127.0.0.1	127.0.0.1	GRPC/...	436	DATA[3] (GRPC) (PROTOBUF)
27	0.034990	127.0.0.1	127.0.0.1	HTTP2	66	HEADERS[3]

> HyperText Transfer Protocol 2

> HyperText Transfer Protocol 2

> HyperText Transfer Protocol 2

> HyperText Transfer Protocol 2

> GRPC Message: /arrow.flight.protocol.FlightService/GetFlightInfo, Request

> Protocol Buffers: /arrow.flight.protocol.FlightService/GetFlightInfo,request

> Message: <UNKNOWN>

> Field(1): 2 (uint32)

> [Field Name: <UNKNOWN>]

> .000 1... = Field Number: 1

> 000 = Wire Type: varint (0)

> Value: 02

> Field(2):

> [Field Name: <UNKNOWN>]

> .001 0... = Field Number: 2

> 010 = Wire Type: Length-delimited (2)

> Value Length: 138

> Value [...]: 0a43747970652e676f66676c65617069732e636f6d2f6172726f772e666c696768742e70726f74

00f0 65 66 6c 61 74 65 2c 20 67 7a 69 70 40 0c 67 72 eflate, gzip@gr
0100 70 63 2d 74 69 6d 65 6f 75 74 03 33 30 53 40 0a pc-timeo ut·30S@·
0110 75 73 65 72 2d 61 67 65 6e 74 2b 67 72 70 63 2d user-age nt+grpc-
0120 63 2b 2b 2f 31 2e 35 31 2e 31 20 67 72 70 63 2d c++/1.51 .1 grpc-
0130 63 2f 32 39 2e 30 2e 30 20 28 6f 73 78 3b 20 63 c/29.0.0 (osx; c
0140 68 74 74 70 32 29 00 0d 61 75 74 68 6f 72 69 7a http2)·· authoriz
0150 61 74 69 6f 6e 1e 42 61 73 69 63 20 64 32 39 79 ation·Ba sic d29y
0160 61 33 4e 6f 62 33 41 36 64 32 39 79 61 33 4e 6f a3Nob3A6 d29ya3No
0170 62 33 41 3d 00 00 04 08 00 00 00 00 01 00 00 00 b3A=.....
0180 05 00 00 94 00 01 00 00 00 01 00 00 00 00 00 8f 08
0190 02 12 8a 01 0a 43 74 79 70 65 2e 67 6f 6f 67 6c Cty pe.googl
01a0 65 61 70 69 73 2e 63 6f 6d 2f 61 72 72 6f 77 2e eapis.co m/arrow.
01b0 66 6c 69 67 68 74 2e 70 72 6f 74 6f 63 6f 6c 2e flight.p rotocol.
01c0 73 71 6c 2e 43 6f 6d 6d 61 6e 64 53 74 61 74 65 sql.Comm andState
01d0 6d 65 6e 74 51 75 65 72 79 12 43 0a 41 53 45 4c mentQuer y·C·ASEL
01e0 45 43 54 20 63 6f 6e 66 2c 20 73 74 61 72 74 73 ECT conf , starts
01f0 5f 61 74 2c 20 73 70 65 61 6b 65 72 20 46 52 4f _at, spe aker FRO
0200 4d 20 73 6d 61 72 74 64 61 74 61 20 4f 52 44 45 M smartd ata ORDE
0210 52 20 42 59 20 73 74 61 72 74 73 5f 61 74 00 00 R BY sta rts_at·
0220 04 08 00 00 00 00 00 00 00 00 00 05

Packet (555 bytes)

Decompressed Header (374 bytes)

The wire type determines value format (protobuf.field.value), 138 bytes

Packets: 34 · Displayed: 13 (38.2%)

Profile: Default

ClickHouse® Flight SQL

выглядит одинаково поверх обеих баз

Current filter: http2

No.	Time	Source	Destination	Protocol	Length	Info
5	0.001526	127.0.0.1	127.0.0.1	HTTP2	138	Magic, SETTINGS[0], WINDOW_UPDATE[0]
7	0.012057	127.0.0.1	127.0.0.1	HTTP2	102	SETTINGS[0], WINDOW_UPDATE[0]
9	0.012584	127.0.0.1	127.0.0.1	GRPC/...	555	SETTINGS[0], HEADERS[1]: POST /arrow.flight.protocol.FlightService/GetFlightInfo, W...
11	0.015609	127.0.0.1	127.0.0.1	HTTP2	95	SETTINGS[0], WINDOW_UPDATE[0], PING[0]
13	0.015661	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
15	0.032771	127.0.0.1	127.0.0.1	GRPC/...	773	HEADERS[1]: 200 OK, DATA[1] (GRPC) (PROTOBUF), HEADERS[1]
17	0.032929	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
19	0.033233	127.0.0.1	127.0.0.1	HTTP2	73	PING[0]
21	0.033739	127.0.0.1	127.0.0.1	GRPC/...	242	HEADERS[3]: POST /arrow.flight.protocol.FlightService/DoGet, DATA[3] (GRPC) (PROTOB...
23	0.034765	127.0.0.1	127.0.0.1	GRPC/...	401	HEADERS[3]: 200 OK, DATA[3] (GRPC) (PROTOBUF), WINDOW_UPDATE[0]
25	0.034947	127.0.0.1	127.0.0.1	GRPC/...	436	DATA[3] (GRPC) (PROTOBUF)
27	0.034990	127.0.0.1	127.0.0.1	HTTP2	66	HEADERS[3]

[Server contiguous streams: 1]
TCP payload (380 bytes)
[PDU Size: 380]

> HyperText Transfer Protocol 2
> GRPC Message: /arrow.flight.protocol.FlightService/DoGet, Response
v Protocol Buffers: /arrow.flight.protocol.FlightService/DoGet,response
 v Message: <UNKNOWN>
 v Field(2):
 [Field Name: <UNKNOWN>]
 .001 0... = Field Number: 2
 010 = Wire Type: Length-delimited (2)
 Value Length: 248
 Value [...]: 1400000000000000c0016000600050008000c000c000000000304001800000070000000000000
 v Field(1000):
 [Field Name: <UNKNOWN>]
 .001 1111 .100 0... = Field Number: 1000
 010 = Wire Type: Length-delimited (2)
 Value Length: 112
 Value: 0000000009000000120000001b0000002400000000000000536d61727444617461536d617274446174

0060 00 00 03 04 00 18 00 00 00 70 00 00 00 00 00 00 p.....
0070 00 00 00 0a 00 18 00 0c 00 04 00 08 00 0a 00 00
0080 00 8c 00 00 00 10 00 00 00 04 00 00 00 00 00 00
0090 00 00 00 00 00 07 00 00 00 00 00 00 00 00 00 00
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0 00 14 00 00 00 00 00 00 00 18 00 00 00 00 00 00
00c0 00 24 00 00 00 00 00 00 00 40 00 00 00 00 00 00 \$. @.....
00d0 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 @.....
00e0 00 20 00 00 00 00 00 00 00 60 00 00 00 00 00 00 `.....
00f0 00 00 00 00 00 00 00 00 00 60 00 00 00 00 00 00 ^.....
0100 00 10 00 00 00 00 00 00 00 00 00 00 00 03 00 00
0110 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0120 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0130 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00>p.....
0140 00 c2 3e 70 00 00 00 00 09 00 00 00 12 00 00 00\$... ..Smart
0150 1b 00 00 00 24 00 00 00 00 00 00 00 53 6d 61 72 tDataSma rtDataSm
0160 74 44 61 74 61 53 6d 61 72 74 44 61 74 61 53 6d artDataS martData
0170 61 72 74 44 61 74 61 53 6d 61 72 74 44 61 74 61 u ;.....
0180 00 00 00 00 00 ec ed 75 3b 05 06 00 00 91 b5 e9 e f@.....
0190 98 21 06 00 00 a8 db 65 66 40 06 00 00 da 20 f7 #\.....
01a0 23 5c 06 00 00 00 00 01 00 00 00 00 00 00 00
01b0 01 00 00 00

The wire type determines value format (protobuf.field.value), 112 bytes

Packets: 34 · Displayed: 13 (38.2%)

Profile: Default

За сценарий: клиент получил 17 пак. / 2 518 Б, отправил 17 пак. / 1 791 Б. Всего 4 309 Б.
TCP payload: 2 381 Б (55,3% объёма кадров). Включает служебные сообщения.

Что намерили бенчмарком

БД и Клиент

работают на разных мощностях

Клиенты	Сеть	Серверы БД
32 vCPU 128 GB RAM Python-драйверы	Находятся в одной VPC в одной зоне	16 vCPU 64 GB RAM PostgreSQL 18.6 / ClickHouse® 26.7.5.10

Датасет: публичный ClickBench hits

Срез	Колонки	Какой паттерн эмулируем
w10	WatchID, JavaEnable, Title, EventTime, CounterID, ClientIP, RegionID, UserID, URL, Referer	Числа, строки, timestamp
w50	первые 50 колонок DDL	Смешанные числа и строки
w105	Вся таблица	Широкая аналитическая выгрузка

12 подмножеств одного датасета

в Parquet (zstd) / CSV размер* в МБ

Строки/колонки	w10	w50	w105
10 тыс.	0.5 / 11	0.6 / 12	0.8 / 15
100 тыс.	5.2 / 104	5.9 / 121	7.7 / 147
1 млн	55 / 747	62 / 905	88 / 1 178
10 млн	691 / 6 609	799 / 8 305	1 073 / 10 993

* размер с округлением

PostgreSQL

Путь	Кортежи	Arrow	pandas	polars
psycopg simple text	да	-	да	да
psycopg extended text	да	-	да	да
psycopg extended binary	да	-	да	да
COPY text	-	-	-	-
COPY binary	-	-	-	-
COPY csv	-	-	да	да
ADBC поверх COPY	-	да	да	да
ADBC без COPY	-	да	да	да
connectorx binary	-	да	да	да
Flight SQL адаптер	-	да (w10)	да (w10)	да (w10)

PostgreSQL

Строки/колонки	w10	w50	w105
10 тыс.	text 0,061 с / 5 МБ COPY CSV 0,097 с / 4 МБ ADBC 0,035 с / 4 МБ	text 0,178 с / 7 МБ COPY CSV 0,168 с / 6 МБ ADBC 0,077 с / 7 МБ	text 0,331 с / 11 МБ COPY CSV 0,271 с / 8 МБ ADBC 0,137 с / 11 МБ
100 тыс.	text 0,694 с / 48 МБ COPY CSV 0,812 с / 45 МБ ADBC 0,288 с / 45 МБ Flight SQL* 0,463 с / 41 МБ	text 1,84 с / 74 МБ COPY CSV 1,48 с / 60 МБ ADBC 0,600 с / 70 МБ	text 3,95 с / 111 МБ COPY CSV 2,37 с / 84 МБ ADBC 1,10 с / 106 МБ
1 млн	text 6,05 с / 406 МБ COPY CSV 6,86 с / 377 МБ ADBC 2,42 с / 373 МБ Flight SQL* 4,79 с / 340 МБ	text 19,9 с / 662 МБ COPY CSV 13,1 с / 524 МБ ADBC 5,10 с / 618 МБ	text 43,3 с / 1049 МБ COPY CSV 22,2 с / 782 МБ ADBC 9,34 с / 995 МБ
10 млн	text 59,4 с / 3823 МБ COPY CSV 66,7 с / 3537 МБ ADBC 25,3 с / 3479 МБ	text 197,1 с / 6381 МБ COPY CSV 124,5 с / 5007 МБ ADBC 50,3 с / 5970 МБ	text 401,3 с / 10210 МБ COPY CSV 218,7 с / 7546 МБ ADBC 89,9 с / 9751 МБ

PostgreSQL

Строки/колонки	w10	w50	w105
10 тыс.	text 0,061 с / 5 МБ COPY CSV 0,097 с / 4 МБ ADBC 0,035 с / 4 МБ	text 0,178 с / 7 МБ COPY CSV 0,168 с / 6 МБ ADBC 0,077 с / 7 МБ	text 0,331 с / 11 МБ COPY CSV 0,271 с / 8 МБ ADBC 0,137 с / 11 МБ
100 тыс.	text 0,694 с / 48 МБ COPY CSV 0,812 с / 45 МБ ADBC 0,288 с / 45 МБ Flight SQL* 0,463 с / 41 МБ	text 1,84 с / 74 МБ COPY CSV 1,48 с / 60 МБ ADBC 0,600 с / 70 МБ	text 3,95 с / 111 МБ COPY CSV 2,37 с / 84 МБ ADBC 1,10 с / 106 МБ
1 млн	text 6,05 с / 406 МБ COPY CSV 6,86 с / 377 МБ ADBC 2,42 с / 373 МБ Flight SQL* 4,79 с / 340 МБ	text 19,9 с / 662 МБ COPY CSV 13,1 с / 524 МБ ADBC 5,10 с / 618 МБ	text 43,3 с / 1049 МБ COPY CSV 22,2 с / 782 МБ ADBC 9,34 с / 995 МБ
10 млн	text 59,4 с / 3823 МБ COPY CSV 66,7 с / 3537 МБ ADBC 25,3 с / 3479 МБ	text 197,1 с / 6381 МБ COPY CSV 124,5 с / 5007 МБ ADBC 50,3 с / 5970 МБ	text 401,3 с / 10210 МБ COPY CSV 218,7 с / 7546 МБ ADBC 89,9 с / 9751 МБ

PostgreSQL

Строки/колонки		w10	w50	w105
10 тыс.		text 100% COPY CSV 158% ADBC 58%	text 100% COPY CSV 95% ADBC 44%	text 100% COPY CSV 82% ADBC 41%
100 тыс.		text 100% COPY CSV 117% ADBC 42% Flight SQL* 67%	text 100% COPY CSV 80% ADBC 33%	text 100% COPY CSV 60% ADBC 28%
1 млн		text 100% COPY CSV 113% ADBC 40% Flight SQL* 79%	text 100% COPY CSV 66% ADBC 26%	text 100% COPY CSV 51% ADBC 22%
10 млн		text 100% COPY CSV 112% ADBC 43%	text 100% COPY CSV 63% ADBC 26%	text 100% COPY CSV 55% ADBC 22%

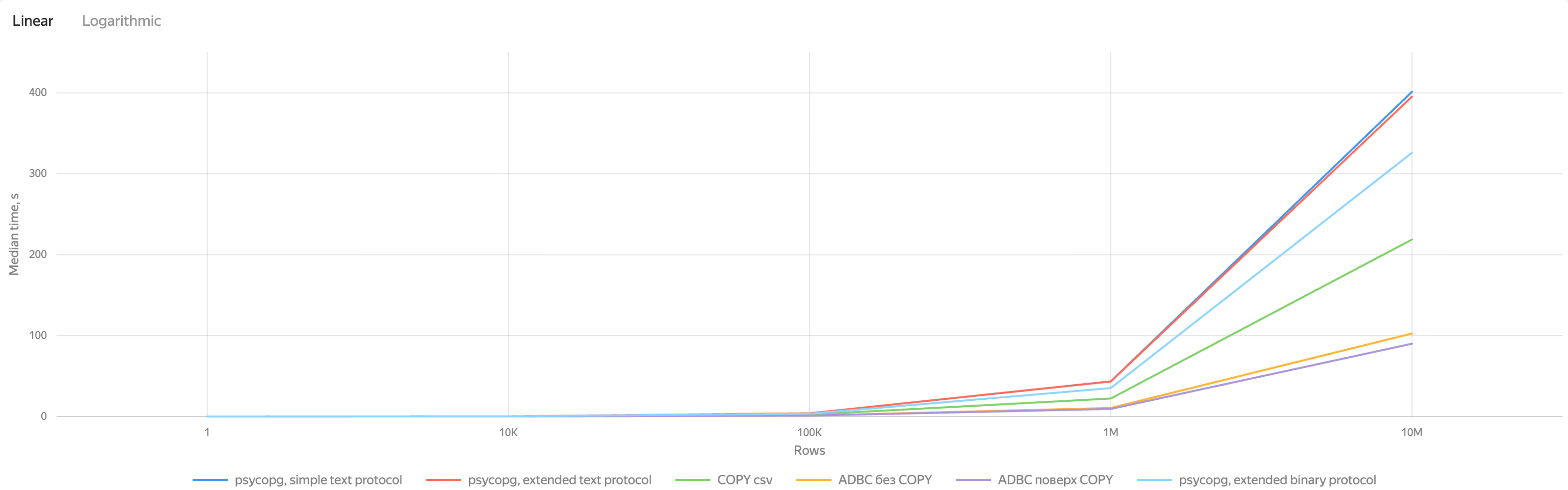
Время в % от text (100%). Меньше — быстрее. Все пути до pandas DataFrame.

PostgreSQL

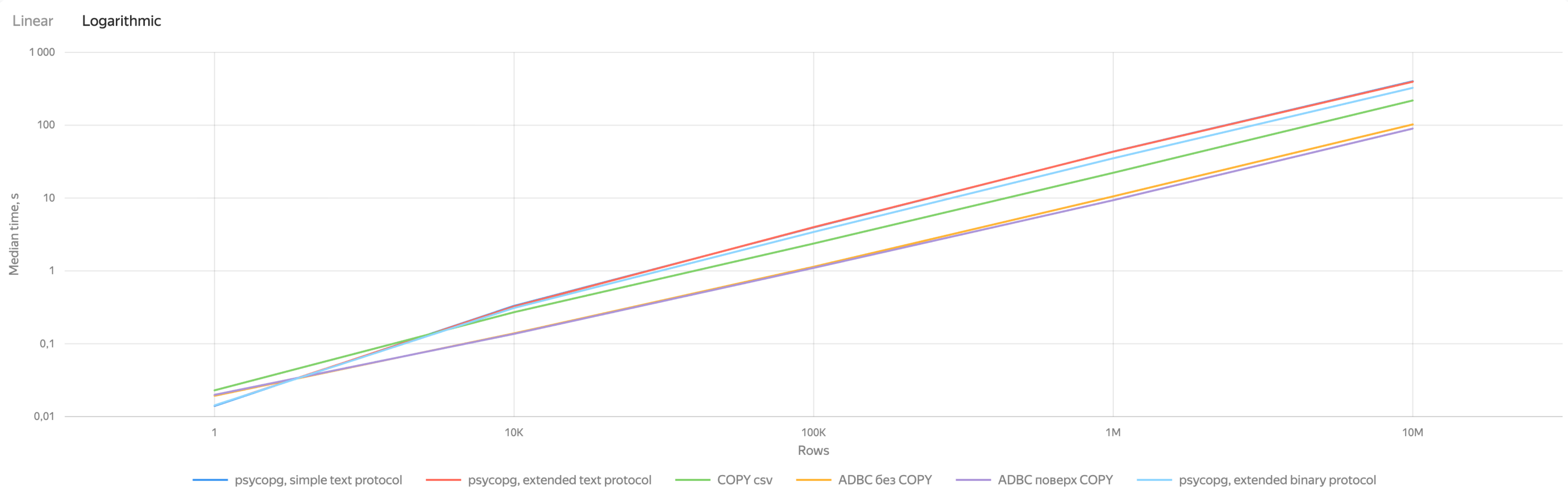
Строки/колонки		w10	w50	w105
10 тыс.		text 100% COPY CSV 158% ADBC 58%	text 100% COPY CSV 95% ADBC 44%	text 100% COPY CSV 82% ADBC 41%
100 тыс.		text 100% COPY CSV 117% ADBC 42% Flight SQL* 67%	text 100% COPY CSV 80% ADBC 33%	text 100% COPY CSV 60% ADBC 28%
1 млн		text 100% COPY CSV 113% ADBC 40% Flight SQL* 79%	text 100% COPY CSV 66% ADBC 26%	text 100% COPY CSV 51% ADBC 22%
10 млн		text 100% COPY CSV 112% ADBC 43%	text 100% COPY CSV 63% ADBC 26%	text 100% COPY CSV 55% ADBC 22%

Время в % от text (100%). Меньше — быстрее. Все пути до pandas DataFrame.

PostgreSQL



PostgreSQL



ClickHouse®

Путь	Кортежи	Arrow	pandas	polars
Native TCP, ClickHouse®-driver	да	-	да	да
HTTP Native, ClickHouse®-connect	да	-	да	да
HTTP ArrowStream	-	да	да	да
HTTP Parquet	-	да	да	да
HTTP RowBinary	-	-	-	-
HTTP CSV	-	-	да	да
HTTP JSONEachRow	да	да	да	да
ADBC ClickHouse® по HTTP	-	да	да	да
Flight SQL	-	да (w10)	да (w10)	да (w10)
pg-wire эмуляция, psycopg	да	-	да	да

Строки/колонки	w10	w50	w105
10 тыс.	Native TCP 0,049 с / 4 МБ HTTP Native 0,045 с / 4 МБ ArrowStream 0,034 с / 4 МБ	Native TCP 0,144 с / 5 МБ HTTP Native 0,090 с / 5 МБ ArrowStream 0,060 с / 5 МБ	Native TCP 0,246 с / 6 МБ HTTP Native 0,190 с / 7 МБ ArrowStream 0,092 с / 7 МБ
100 тыс.	Native TCP 0,411 с / 40 МБ HTTP Native 0,387 с / 40 МБ ArrowStream 0,173 с / 41 МБ Flight SQL 0,244 с / 41 МБ	Native TCP 1,21 с / 49 МБ HTTP Native 0,718 с / 50 МБ ArrowStream 0,289 с / 52 МБ	Native TCP 2,21 с / 60 МБ HTTP Native 1,65 с / 66 МБ ArrowStream 0,477 с / 74 МБ
1 млн	Native TCP 3,87 с / 328 МБ HTTP Native 3,27 с / 328 МБ ArrowStream 1,27 с / 336 МБ Flight SQL 1,87 с / 337 МБ	Native TCP 11,7 с / 407 МБ HTTP Native 6,16 с / 420 МБ ArrowStream 2,03 с / 444 МБ	Native TCP 20,2 с / 537 МБ HTTP Native 15,2 с / 602 МБ ArrowStream 3,39 с / 684 МБ
10 млн	Native TCP 38,8 с / 3029 МБ HTTP Native 32,9 с / 3028 МБ ArrowStream 14,5 с / 3108 МБ Flight SQL 23,37 с / 3112 МБ	Native TCP 115,5 с / 3859 МБ HTTP Native 62,3 с / 3985 МБ ArrowStream 22,5 с / 4228 МБ	Native TCP 212,3 с / 5076 МБ HTTP Native 152,9 с / 5824 МБ ArrowStream 35,6 с / 6640 МБ

Строки/колонки	w10	w50	w105
10 тыс.	Native TCP 0,049 с / 4 МБ HTTP Native 0,045 с / 4 МБ ArrowStream 0,034 с / 4 МБ	Native TCP 0,144 с / 5 МБ HTTP Native 0,090 с / 5 МБ ArrowStream 0,060 с / 5 МБ	Native TCP 0,246 с / 6 МБ HTTP Native 0,190 с / 7 МБ ArrowStream 0,092 с / 7 МБ
100 тыс.	Native TCP 0,411 с / 40 МБ HTTP Native 0,387 с / 40 МБ ArrowStream 0,173 с / 41 МБ Flight SQL 0,244 с / 41 МБ	Native TCP 1,21 с / 49 МБ HTTP Native 0,718 с / 50 МБ ArrowStream 0,289 с / 52 МБ	Native TCP 2,21 с / 60 МБ HTTP Native 1,65 с / 66 МБ ArrowStream 0,477 с / 74 МБ
1 млн	Native TCP 3,87 с / 328 МБ HTTP Native 3,27 с / 328 МБ ArrowStream 1,27 с / 336 МБ Flight SQL 1,87 с / 337 МБ	Native TCP 11,7 с / 407 МБ HTTP Native 6,16 с / 420 МБ ArrowStream 2,03 с / 444 МБ	Native TCP 20,2 с / 537 МБ HTTP Native 15,2 с / 602 МБ ArrowStream 3,39 с / 684 МБ
10 млн	Native TCP 38,8 с / 3029 МБ HTTP Native 32,9 с / 3028 МБ ArrowStream 14,5 с / 3108 МБ Flight SQL 23,37 с / 3112 МБ	Native TCP 115,5 с / 3859 МБ HTTP Native 62,3 с / 3985 МБ ArrowStream 22,5 с / 4228 МБ	Native TCP 212,3 с / 5076 МБ HTTP Native 152,9 с / 5824 МБ ArrowStream 35,6 с / 6640 МБ

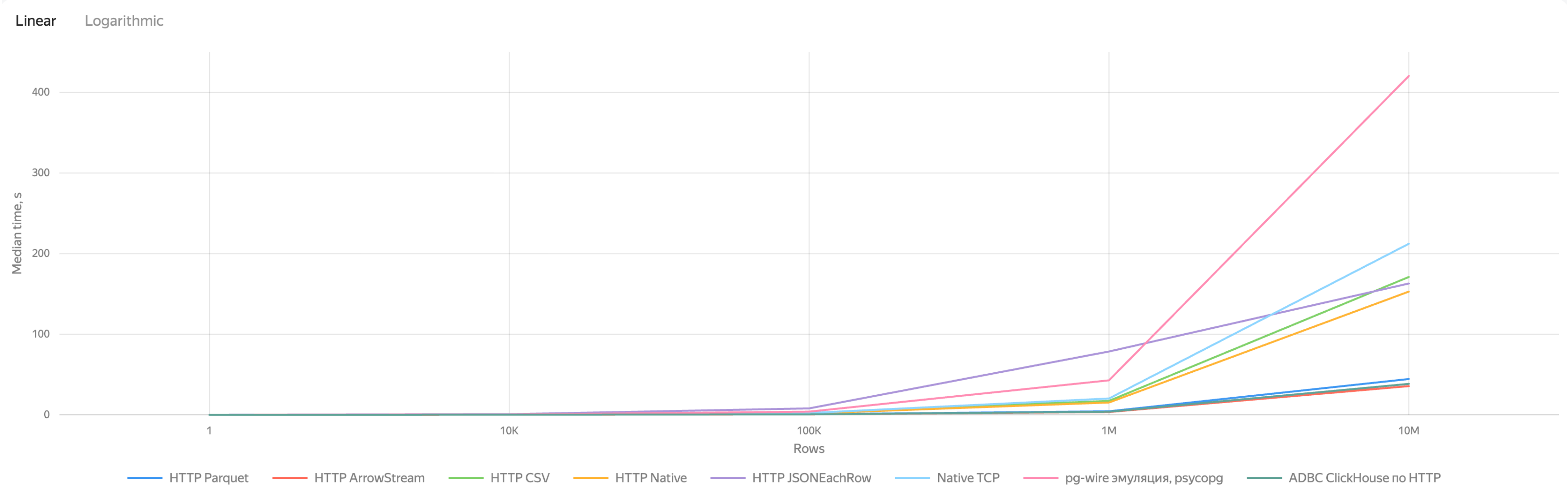
Строки/колонки		w10	w50	w105
10 тыс.		Native TCP 100% HTTP Native 92% ArrowStream 70%	Native TCP 100% HTTP Native 63% ArrowStream 42%	Native TCP 100% HTTP Native 77% ArrowStream 37%
100 тыс.		Native TCP 100% HTTP Native 94% ArrowStream 42% Flight SQL 59%	Native TCP 100% HTTP Native 59% ArrowStream 24%	Native TCP 100% HTTP Native 75% ArrowStream 22%
1 млн		Native TCP 100% HTTP Native 84% ArrowStream 33% Flight SQL 48%	Native TCP 100% HTTP Native 53% ArrowStream 17%	Native TCP 100% HTTP Native 75% ArrowStream 17%
10 млн		Native TCP 100% HTTP Native 85% ArrowStream 37% Flight SQL 60%	Native TCP 100% HTTP Native 54% ArrowStream 19%	Native TCP 100% HTTP Native 72% ArrowStream 17%

Время в % от Native TCP (100%). Меньше — быстрее. Все пути до pandas DataFrame.

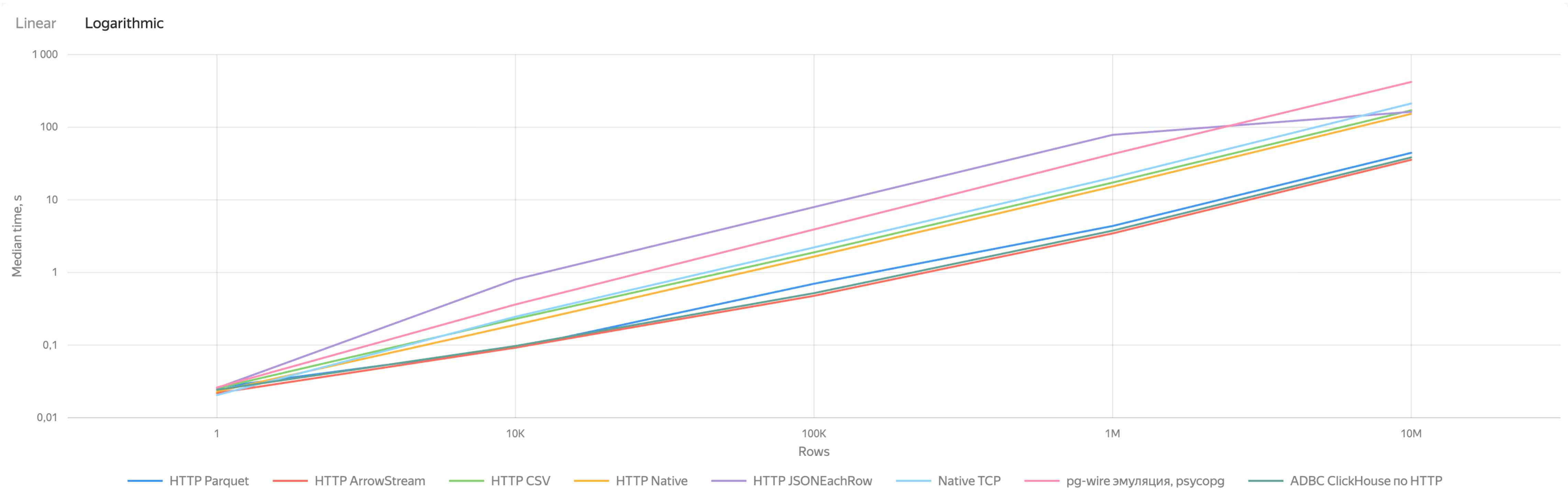
Строки/колонки		w10	w50	w105
10 тыс.		Native TCP 100% HTTP Native 92% ArrowStream 70%	Native TCP 100% HTTP Native 63% ArrowStream 42%	Native TCP 100% HTTP Native 77% ArrowStream 37%
100 тыс.		Native TCP 100% HTTP Native 94% ArrowStream 42% Flight SQL 59%	Native TCP 100% HTTP Native 59% ArrowStream 24%	Native TCP 100% HTTP Native 75% ArrowStream 22%
1 млн		Native TCP 100% HTTP Native 84% ArrowStream 33% Flight SQL 48%	Native TCP 100% HTTP Native 53% ArrowStream 17%	Native TCP 100% HTTP Native 75% ArrowStream 17%
10 млн		Native TCP 100% HTTP Native 85% ArrowStream 37% Flight SQL 60%	Native TCP 100% HTTP Native 54% ArrowStream 19%	Native TCP 100% HTTP Native 72% ArrowStream 17%

Время в % от Native TCP (100%). Меньше — быстрее. Все пути до pandas DataFrame.

ClickHouse®

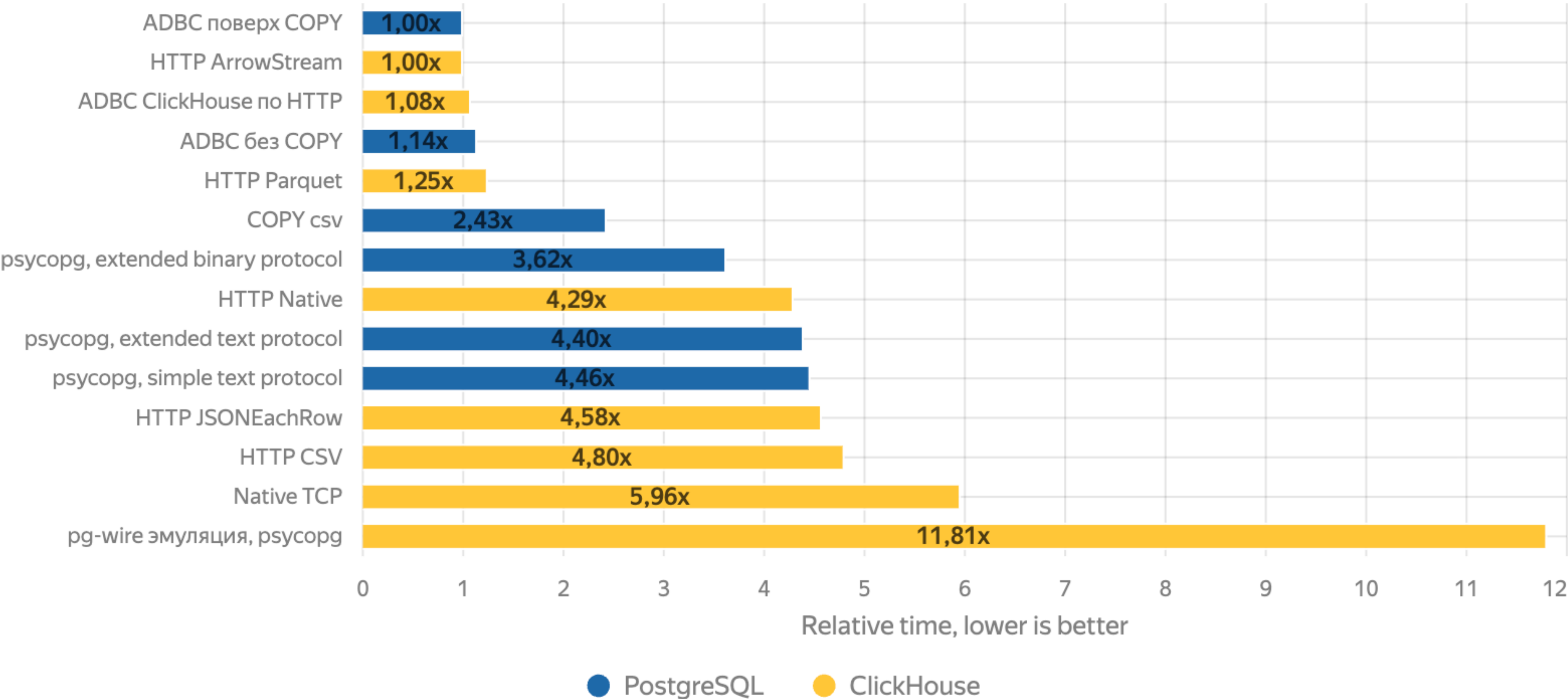


ClickHouse®



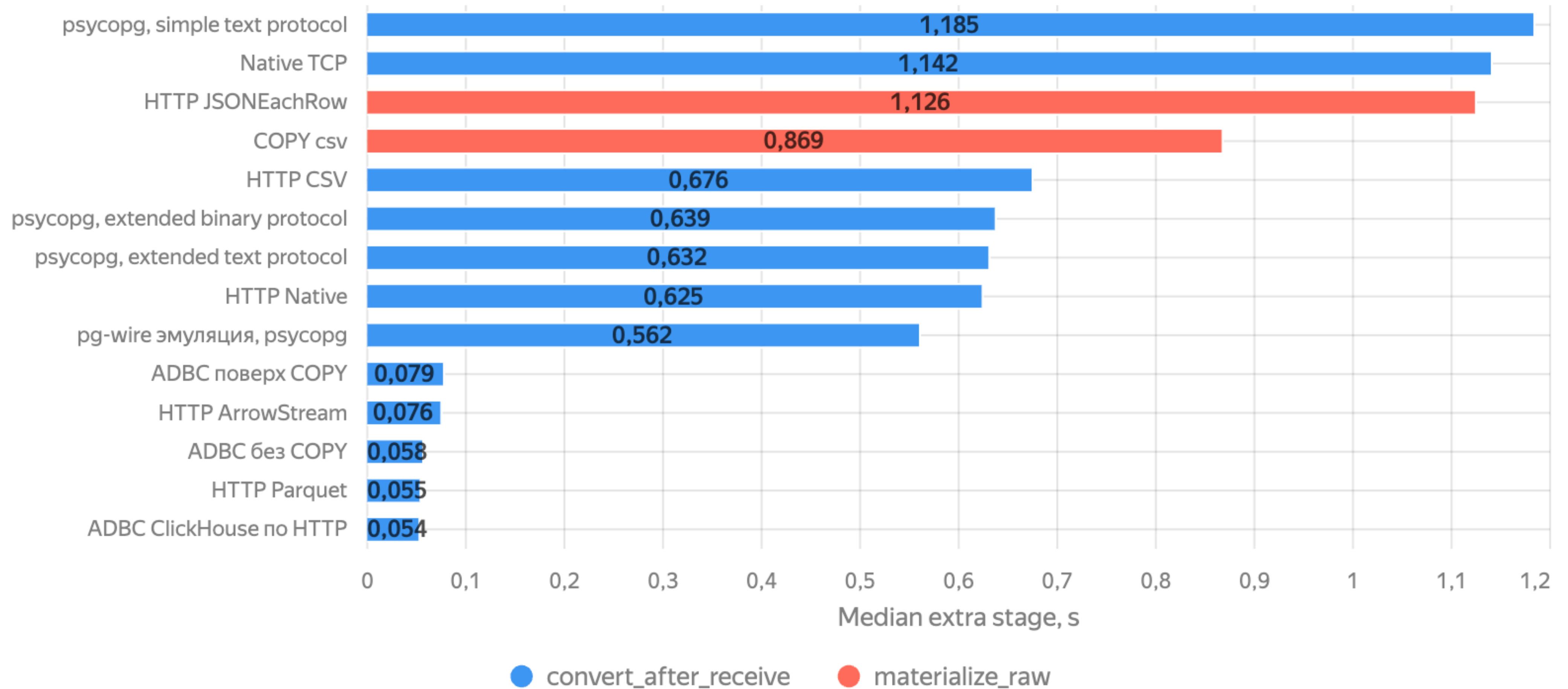
Сравним производительность

Access method rating at largest selected dataset

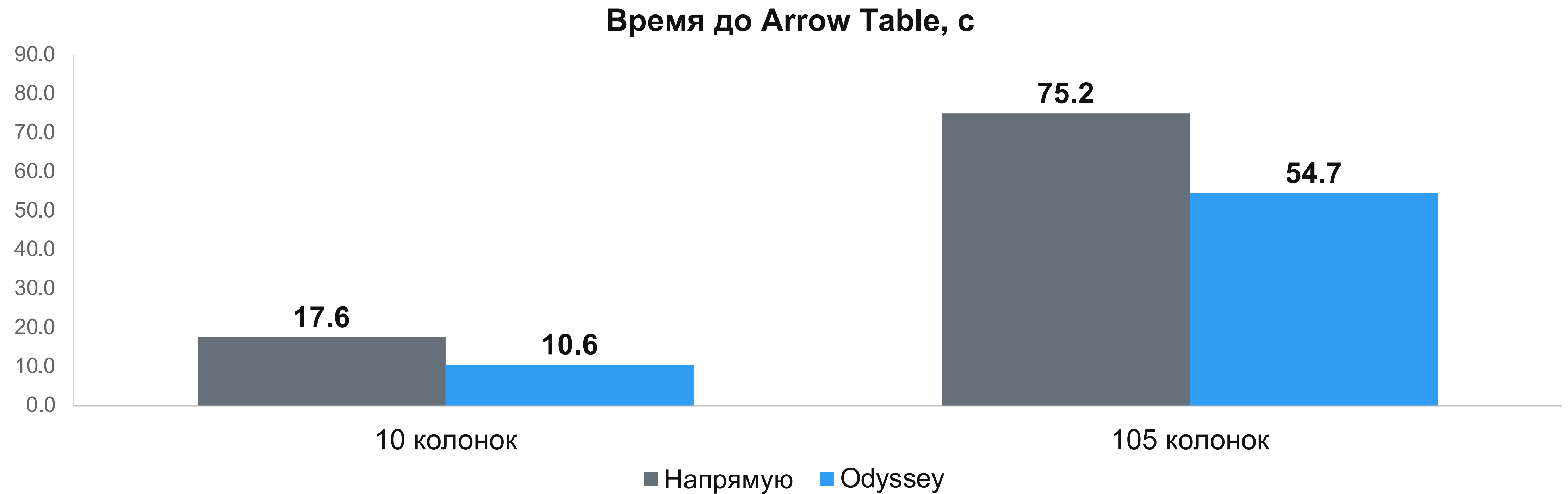


Сериализация

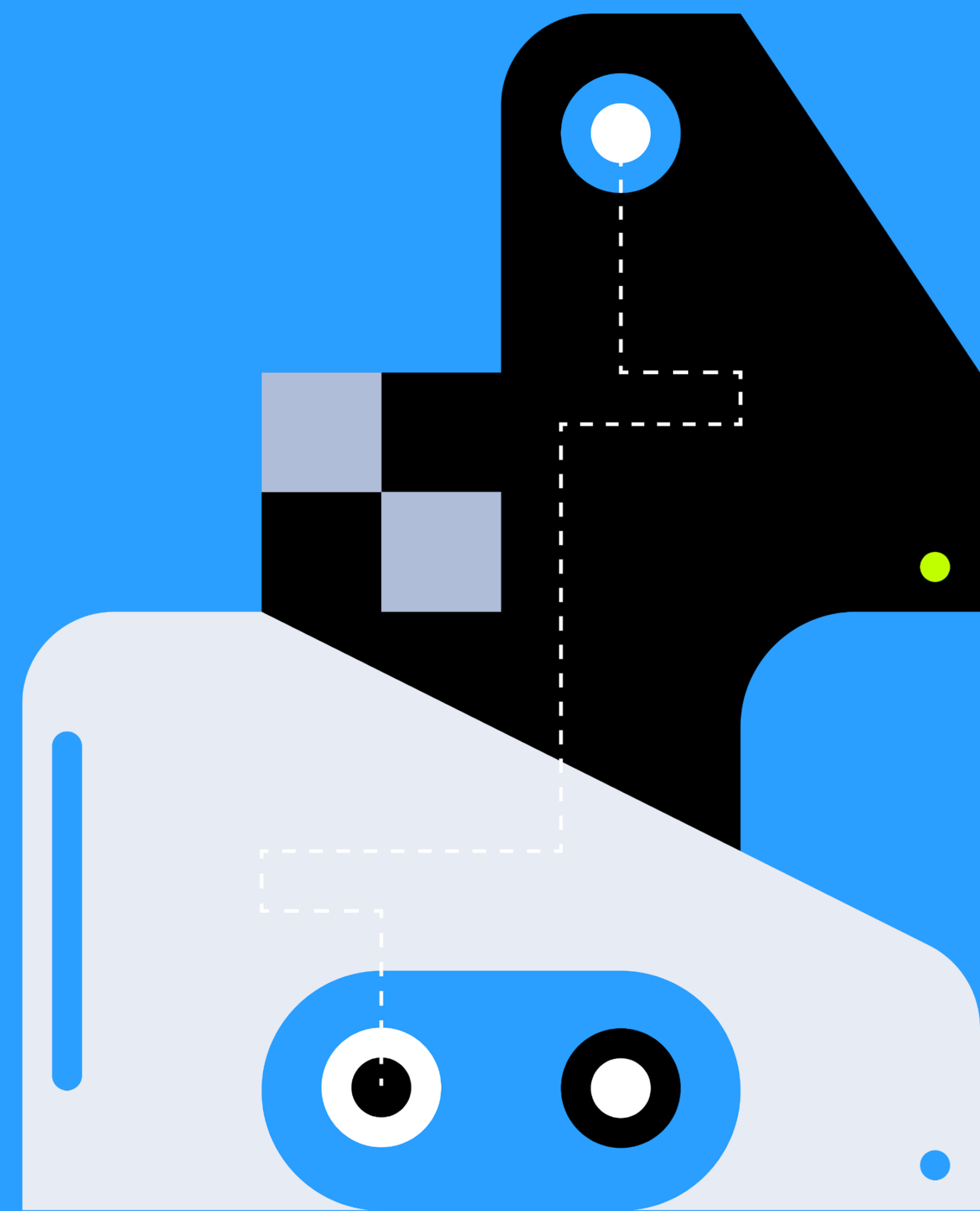
Extra stage time



Пулеры и время получения результата



Итоги



ADBC умеют нативно

Инструмент	Что именно	С какой версии
pandas	Поддерживает ADBC коннекшн	2.2.0
polars	Поддерживает ADBC коннекшн	все
DuckDB	ADBC-драйвер внутри libduckdb; расширение adbc_scanner – DuckDB как ADBC-клиент	
dbt	в dbt v2 (на Rust) ADBC стал единым слоем подключения вместо самописной логики адаптеров	dbt Core v2
Power BI и Microsoft Fabric	встроенные ODBC-драйверы коннекторов заменяются на ADBC	по умолчанию с августа 2026, ODBC убирают из сервиса к концу 2026
Apache Airflow	провайдер apache-airflow-providers- apache-arrow с AdbcHook	0.1.0, релиз 2026-09-14
Библиотеки	Python, R, Java, Go, C#, Rust и т.д.	релиз ADBC 24

Где просили, но **ADBC** не завезли

Инструмент	Тикет	Состояние
DBeaver	dbeaver/dbeaver#15535	открыт с 2022-02-17, метка «new driver»
DataGrip	YouTrack DBE-16742	Submitted с 2022-11-11, не разрешен
Trino	trinodb/trino#18038 , #24586	открыты с 2023 и 2024
SQLAlchemy	sqlalchemy#11742	закрит: мейнтейнер считает, что диалект должен жить снаружи ADBC колоночный, SQLAlchemy построчный
Apache Spark	arrow-adbc#1801	закрит как not planned ставка на Spark Connect
Excel	-	в Excel остается ODBC

Вернемся к «Hold My Data Hostage ...»

Разные операции

netcat льет готовый CSV в /dev/null, SQL-клиент строит типизированный результат.
Отношение 10× - про передачу данных по сети, не про готовый объект.

Вернемся к «Hold My Data Hostage ...»

Разные операции

netcat льет готовый CSV в /dev/null, SQL-клиент строит типизированный результат.
Отношение 10× - про передачу данных по сети, не про готовый объект.

Обоих авторов этой статьи вы знаете:

Марк Раасвельдт и Ханнес Мюлайзен - это основатели **DuckDB**

DuckDB выпустили свой протокол **Quack** в мае 2026

Quack - не **Arrow**. Своя сериализация поверх HTTP

Что важно при сравнении

Весь путь до результата

Время включает протокол, драйвер и сборку (да-да, мы проверяли сценарий с python)

Метаданные и служебные запросы

Учитывайте их время, особенно при чтении небольших результатов

А что с записью?

В этих прогонах измеряли чтение. Скорость и **корректность** записи нужно проверять отдельно

Как воспроизвести



[дашборд](#)



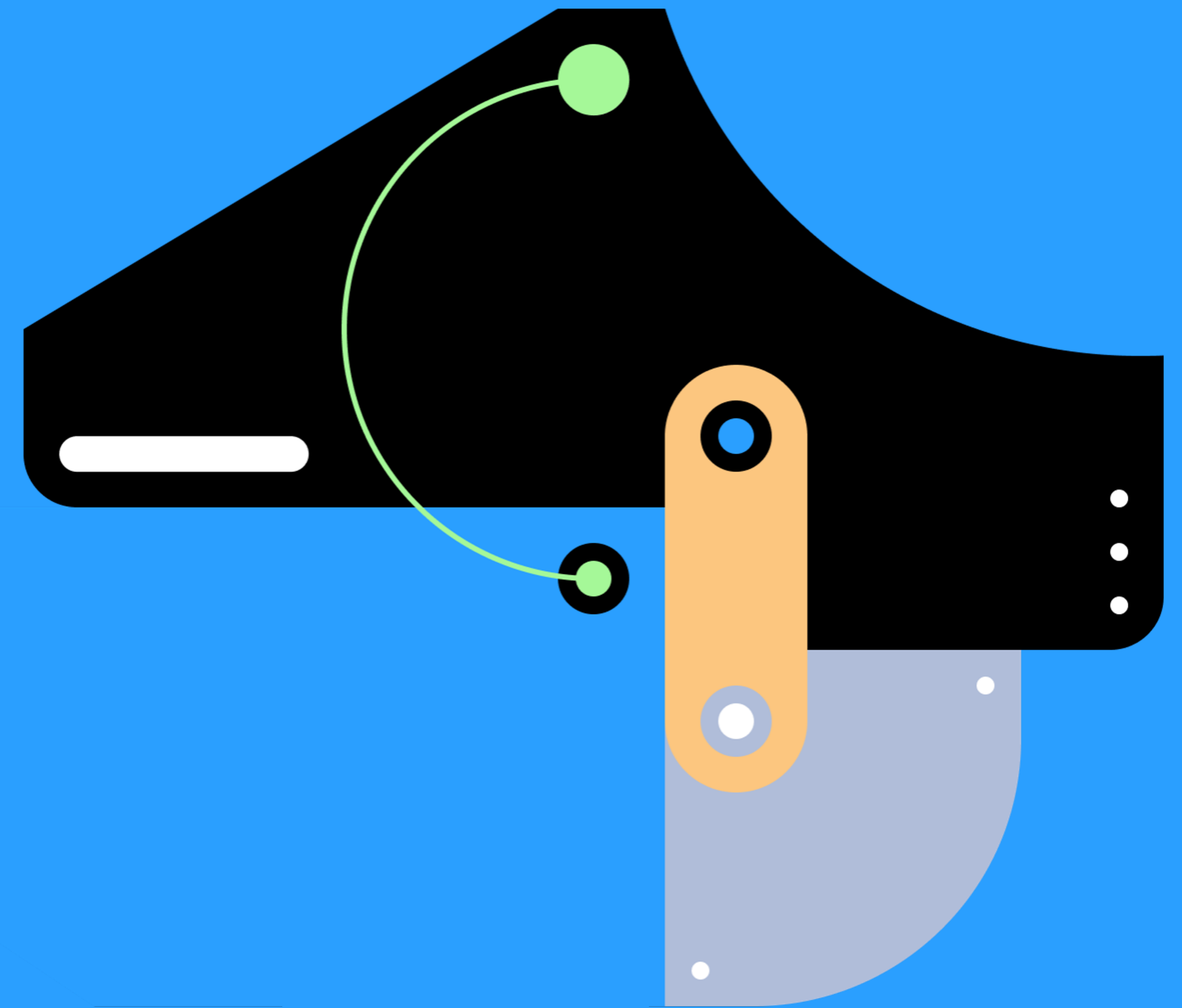
[репозиторий](#)



[данные](#)
[object storage](#)

Важен весь путь до готового результата

Проверяйте на своих данных:
время до конечной структуры с учетом приведения
типов



Петр Гуринов

Команда продуктовой архитектуры
Платформы Данных