What is the Apache DataFusion and user case in eBay

Kun Liu
Software Engineer, @eBay

About me

刘昆/ Kun Liu
liukun@apache.org
Software Engineer(Native engine/Query engine) CDT @eBay

- Open source community
  - IoTDB, TsFile @ Tsinghua (2016-2019)
  - Arrow, DataFusion @eBay (2021-Now)

- Apache IoTDB/TsFile PMC Member
- Apache Arrow/DataFusion PMC Member

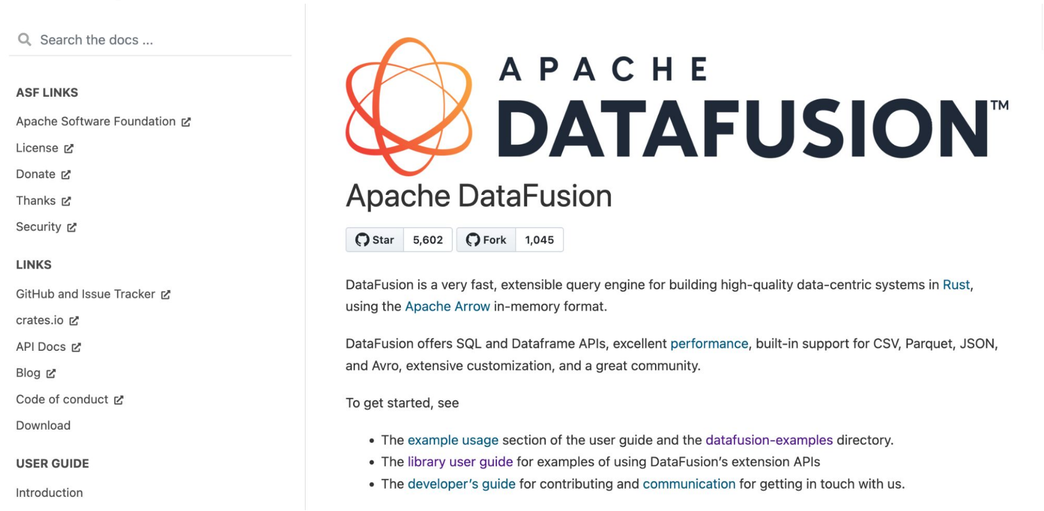
Content

- What is the Apache DataFusion
- Why we need the Apache DataFusion
- The design of Apache DataFusion
- User case of the Apache DataFusion in eBay

Part 01: What is the Apache DataFusion

What is the Apache DataFusion

- Apache DataFusion
  - Donated to Apache Arrow 2019 by @andy grove
  - Subproject of Apache Arrow before April 2024
  - **Top Level Project April 2024**



https://whimsy.apache.org/board/minutes/Arrow.html

What is the Apache DataFusion

- Apache DataFusion: Query Engine *(toolkit)*
  - Rust
  - Apache Arrow as in-memory format

What is the Apache DataFusion

- Apache DataFusion
  - **High performance**

What is the Apache DataFusion

- Apache DataFusion
  - High performance
  - **Customization and Extension**
    - easy to extend
    - easy to embed

Part 02: Why we need the Apache DataFusion

-> how to build a new database from scratch

Why we need the Apache DataFusion

How to build a new database system from scratch？

- Personal experience 
  - **Storage format: file format -> Parquet -> TsFile**

Why we need the Apache DataFusion

How to build a new database system from scratch？

- Personal experience 
  - Storage format: file format -> Parquet -> TsFile
  - **Storage engine: WAL/Mem Table -> LSM Tree engine**

Why we need the Apache DataFusion

How to build a new database system from scratch？

- Personal experience 
  - Storage format: file format -> Parquet -> TsFile
  - Storage engine: WAL/Mem Table -> LSM tree engine
  - **Query:**
    - **SQL -> SQL parser -> Logical plan -> Physical plan**
    - **Catalog/metadata management**
  - **JDBC/Client tools, other tools...**

Why we need the Apache DataFusion

How to build a new database system from scratch？

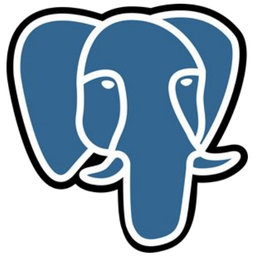
- Personal experience 
  - Storage format: file format -> Parquet -> TsFile
  - Storage engine: WAL/Mem Table -> LSM tree engine
  - Query:
    - SQL -> SQL parser -> Logical plan -> Physical plan
    - Catalog/metadata management
  - JDBC, other tools...

  .....
  - **Distributed architecture: data sharding, replica consistency(raft), distributed query**

Why we need the Apache DataFusion

Building a database or system from scratch **is hard and expensive**

- Catalog
- SQL parser
- Client
- DataType system
- File/Data storage
- Query optimizer
- Query engine
- ....



Why we need the Apache DataFusion

Building a database or system from scratch **is hard and expensive**

- Catalog
- SQL parser
- Client
- DataType system
- File/Data storage
- Query optimizer
- Query engine
- ....

The new systems are built on a foundation of fast, modular components, rather as a single tightly integrated system.

Why we need the Apache DataFusion

Andy's Take:

Databases are the second most important thing in my life, so I enjoy seeing all the developments in the last year.

My hot take on AlloyDB is that it is a neat system, and an impressive amount of engineering went into it, but I still don't know what is novel about it yet. AlloyDB's architecture is similar to Amazon Aurora and Neon, where the DBMS storage has an additional compute layer to process WAL records independently of the compute nodes. Despite already having a solid database portfolio (e.g., Spanner, BigQuery), Google Cloud felt the need to build AlloyDB to try to catch up with Amazon and Microsoft.

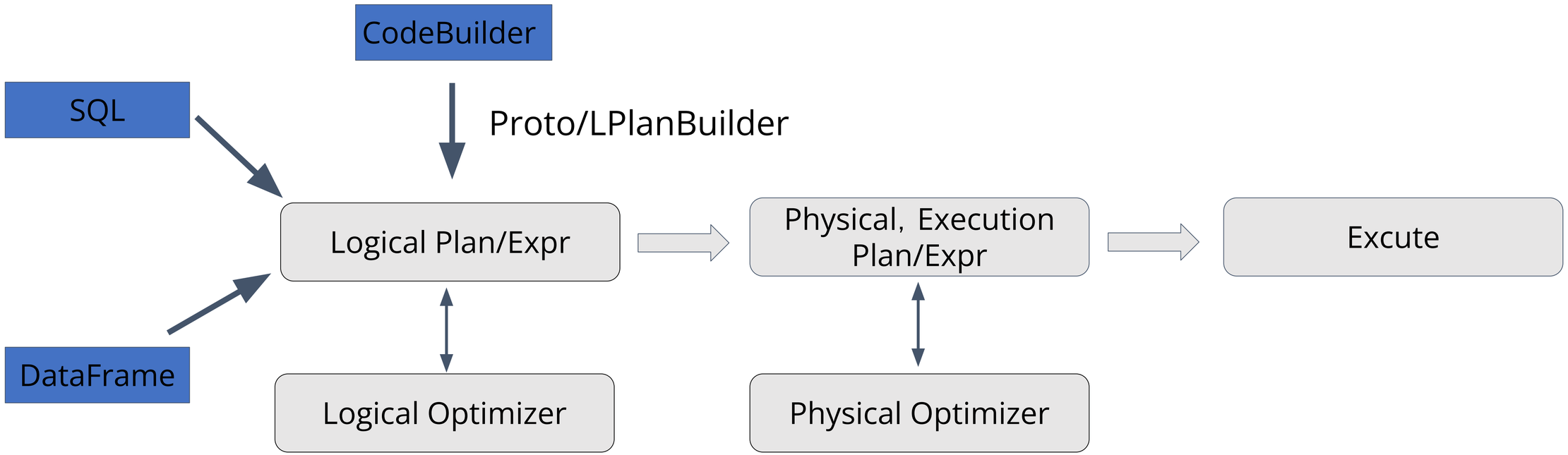
The long-term trend to watch is the proliferation of frameworks like Velox, DataFusion, and Polars. Along with projects like Substrait, the commoditization of these query execution components means that all OLAP DBMSs will be roughly equivalent in the next five years. Instead of building a new DBMS entirely from scratch or hard forking an existing system (e.g., how Firebolt forked Clickhouse), people are better off using an extensible framework like Velox. This means that every DBMS will have the same vectorized execution capabilities that were unique to Snowflake ten years ago. And since in the cloud, the storage layer is the same for everyone (e.g., Amazon controls EBS/S3), the critical differentiator between DBMS offerings will be things that are difficult to quantify, like UI/UX stuff and query optimization.

https://ottertune.com/blog/2022-databases-retrospective/

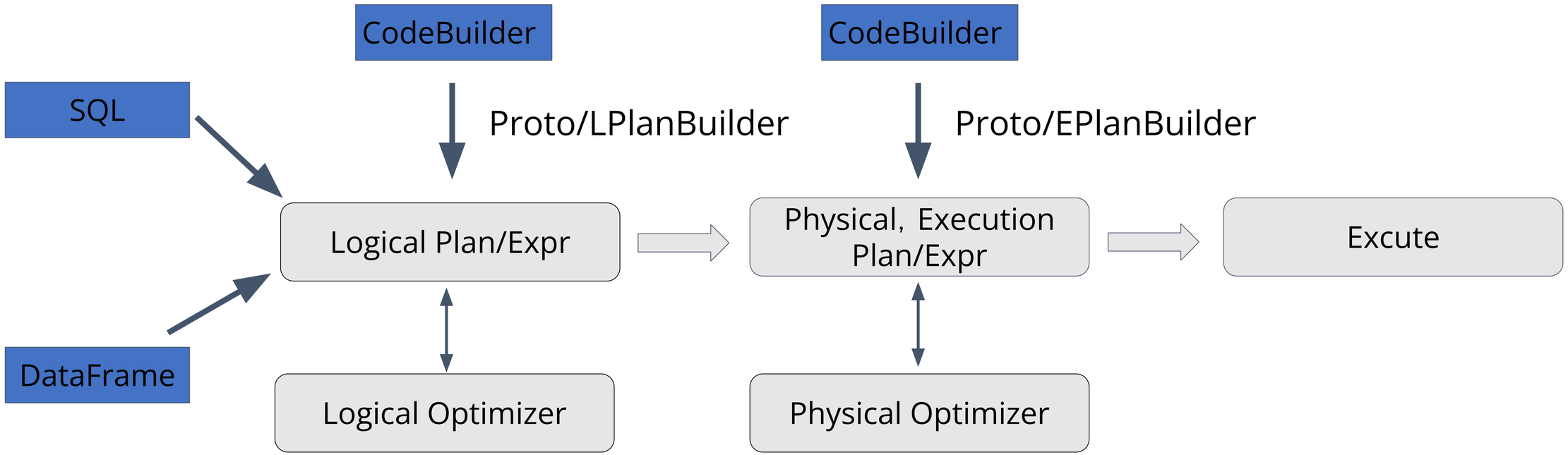
Part 03: The design of Apache DataFusion

Customization and Extension
High performance

The design of Apache DataFusion: Overview



The design of Apache DataFusion: Overview



The design of Apache DataFusion: Extension

SQL

```
SELECT COUNT(C1), C2
FROM TEST WHERE C10='hello'
GROUP BY C2
LIMIT 2;
```

DF

```
let df = ctx.from_table("TEST")
.aggregate(
vec![COL("C2")],
vec![COUNT(COL("C1"))]
)
.filter(COL("C10").eq(LIT("hello")))
.limit(0, 2);
```

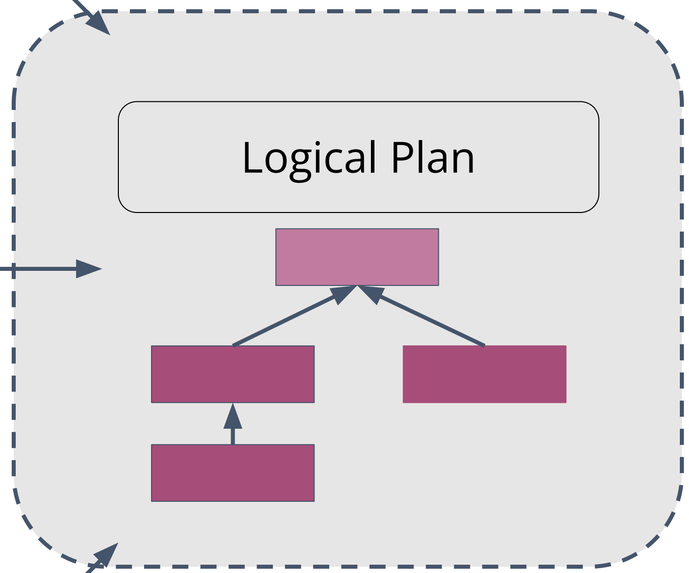
Code

```
LogicalPlanNode{
   LimitNode{
      ProjectionNode{
         AggregateNode{
            FilterNode{
               TableScanNode{ } } } } }}
```

Parser + Planner

DataFrame Builder

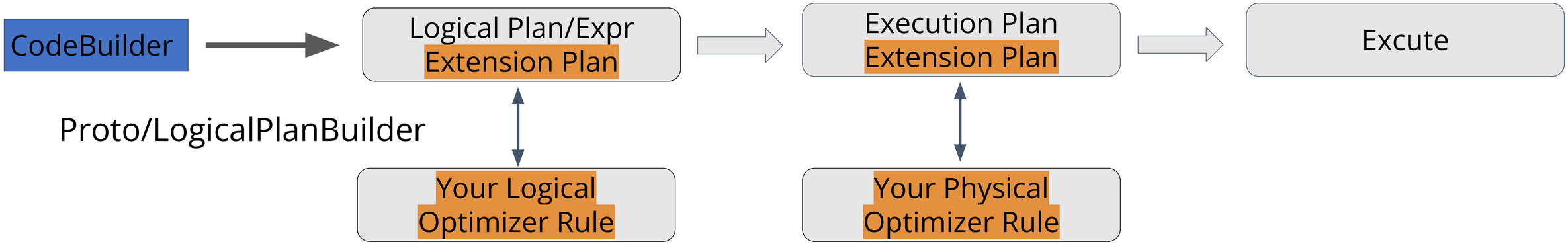
Protobuf Ser/Deser, LogicalPlanBuilder



The design of Apache DataFusion: Extension

Extensibility:

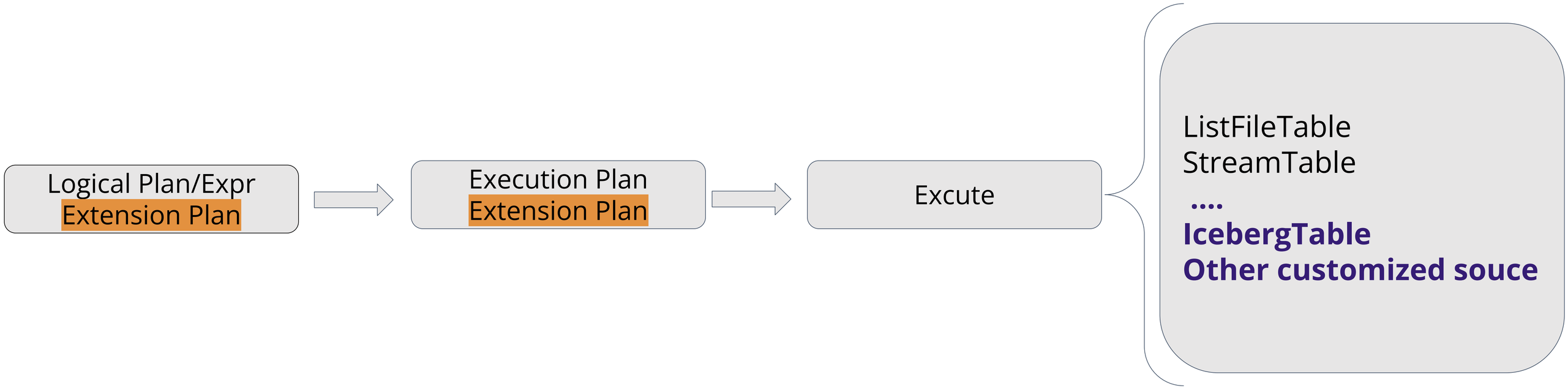
- Extension Plan Node: Add customized nodes/exprs
  - UDF/UDAF...
  - Extend Logical/Physical Plan
- Rules for optimizer
  - Add customized rule
  - Use/Skip the existing rule



The design of Apache DataFusion: Extension

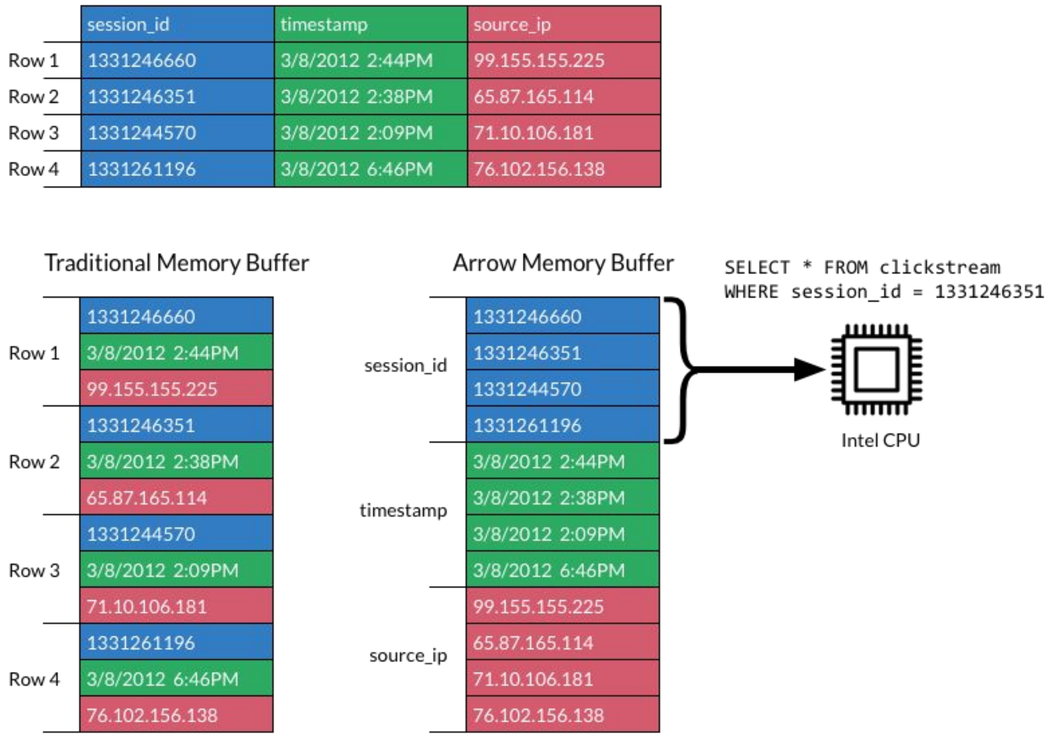
Extensibility:

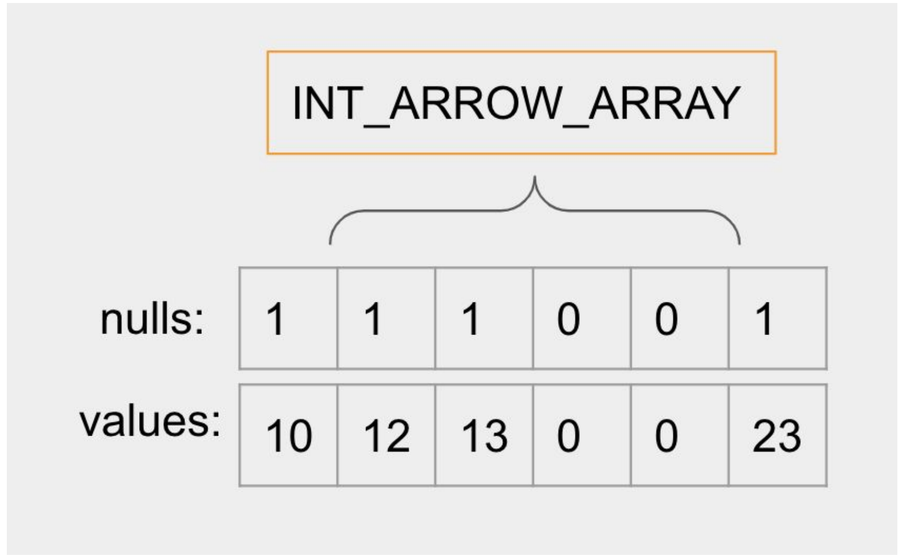
- Table provider(Datasource)
- Catalog provider(Schema)



The design of Apache DataFusion: Performance

- **Apache Arrow: in-memory columnar format**



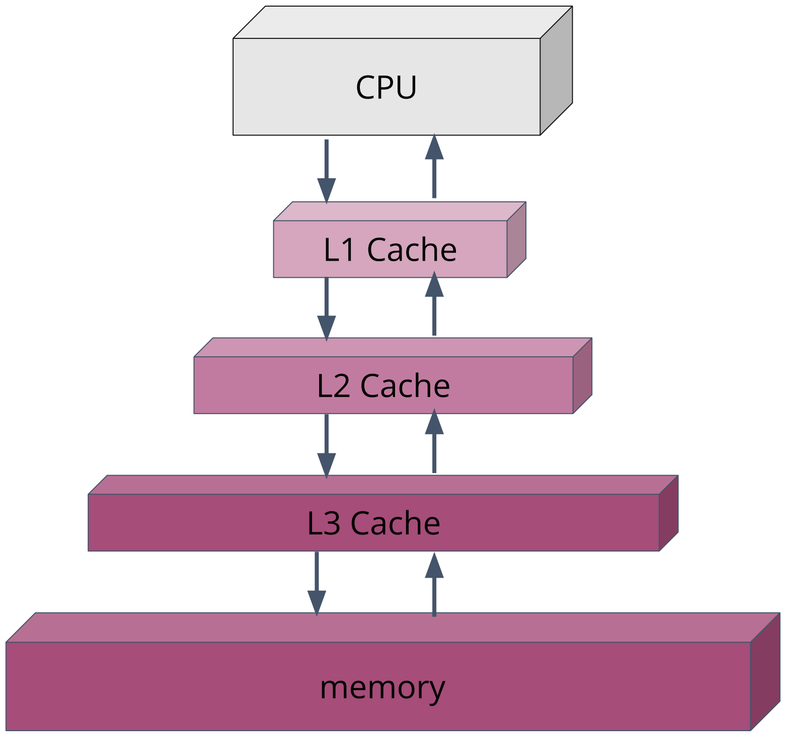


https://arrow.apache.org/overview/

The design of Apache DataFusion: Performance

- **Apache Arrow: CPU Cache-friendly/Vectorization-friendly**

```
for ( i = 0; i < max; i++) {  result[i] = a1[i] +10 }
```



	a1	a2	a3	a4	a5
r1	2	3	4	5	6
r2	1	1	1	1	1
r3	6	4	3	4	5
r4	12	4	42	2	5
r5	12	4	5	2	4

row-based memory layout

	a1	a2	a3	a4	a5
r1	2	3	4	5	6
r2	1	1	1	1	1
r3	6	4	3	4	5
r4	12	4	42	2	5
r5	12	4	5	2	4

column-based memory layout

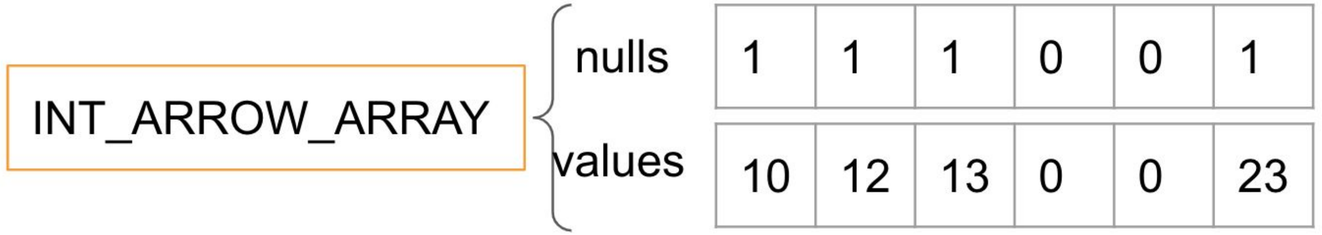
The design of Apache DataFusion: Performance

- **Apache Arrow: CPU pipelining**

SQL: select a1 + a2 from table
How to handle NULL value?

branch-predication failures

```
for(i = 0; i < max; i++) {
   if (a1[i]==NULL|| a2[i]==NULL) {
      result[i] = NULL;
   } else {
      result[i] = a1[i] + a2[i]
   }
}
```



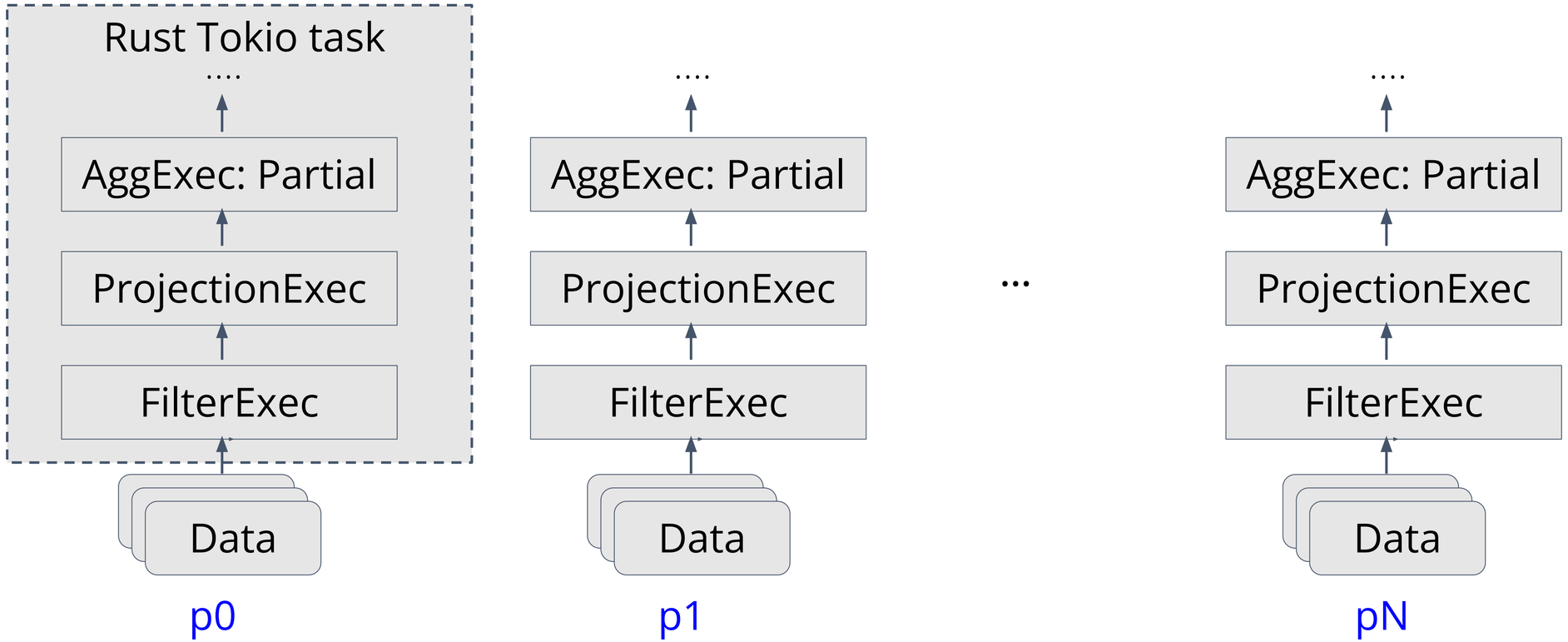

```
for(i = 0; i < max; i++) {
   result.nulls[i] = a1.nulls[i] & a2.nulls[i];
}
for (i = 0; i < max; i++) {
   result.values[i] = a1.values[i] + a2.values[i]
}
```

The design of Apache DataFusion: Performance

- Highlights about Apache DataFusion
  - **Async Scheduler:**
    - rust tokio async, avoid blocking io
  - **Parallelization:**
    - process with partitions
  - **Vectorization:**
    - batch at a time
    - vectorized expr evaluation/operator exec

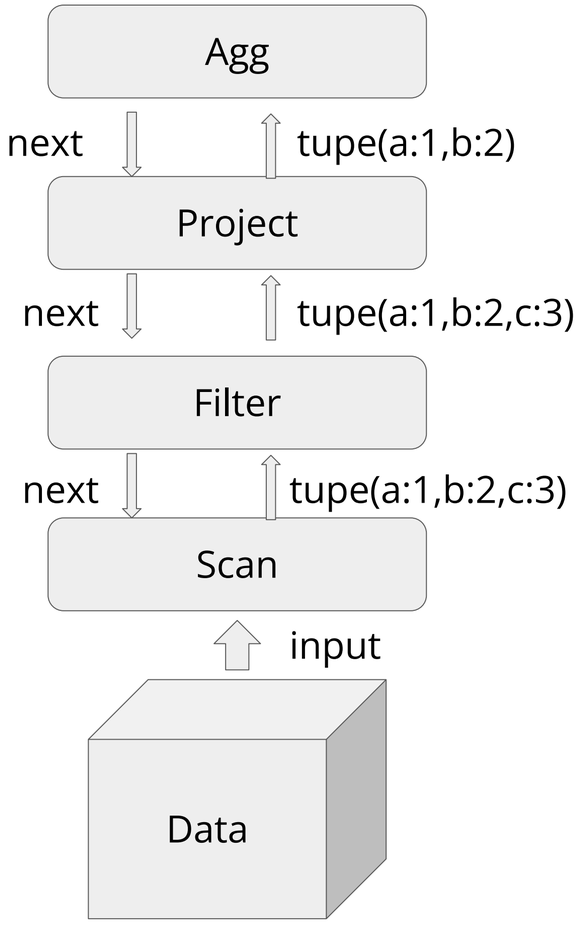
The design of Apache DataFusion: Performance

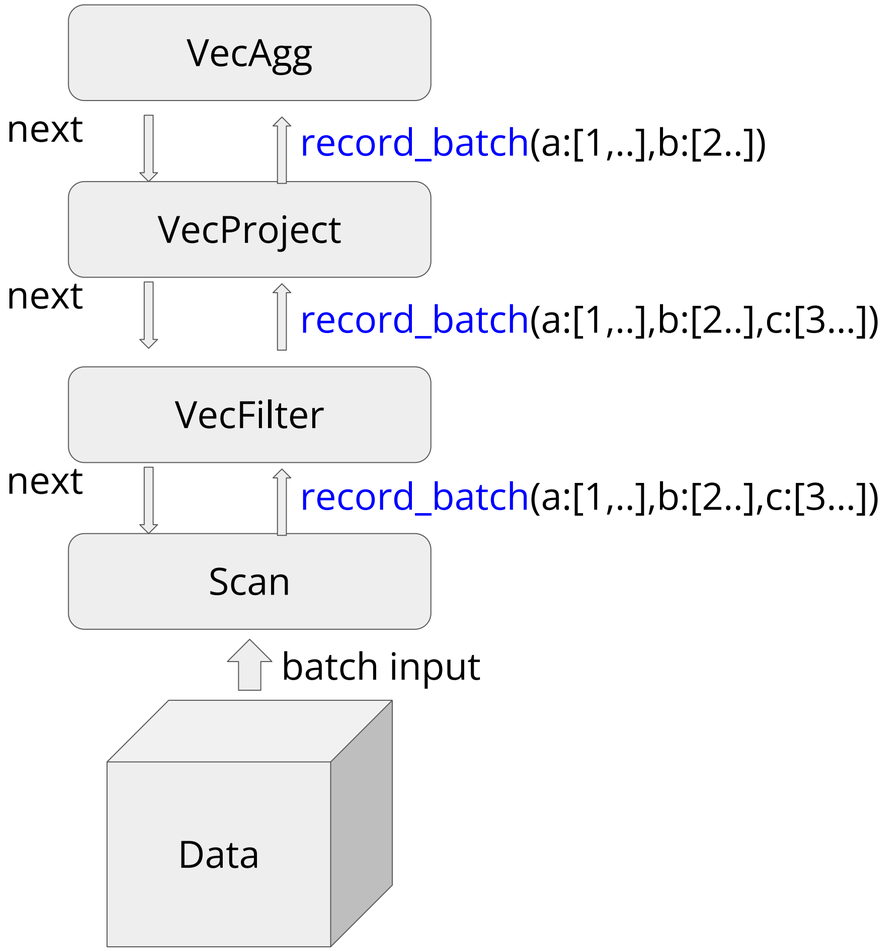
Parallelization: Partitions



The design of Apache DataFusion: Performance

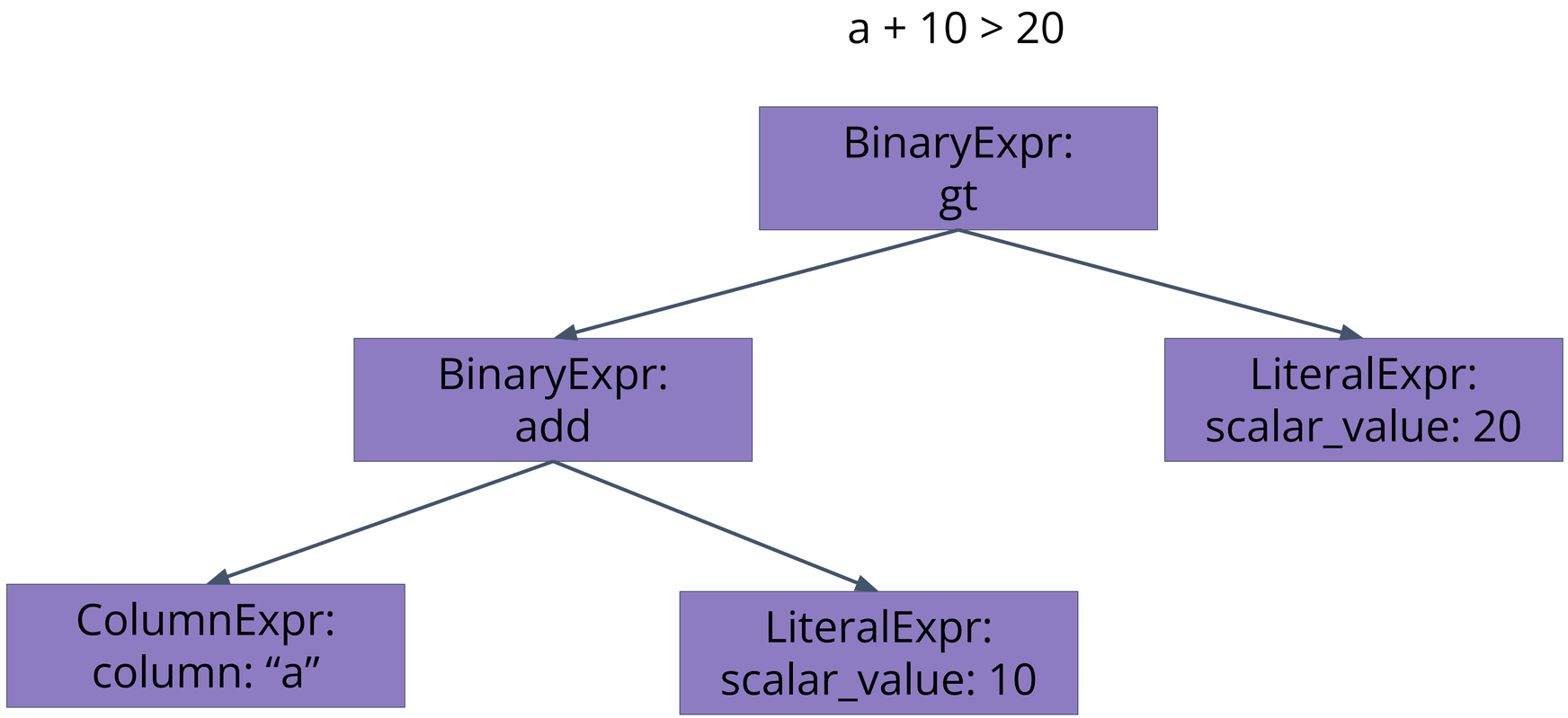
Vectorization: Batch at a time for each operator



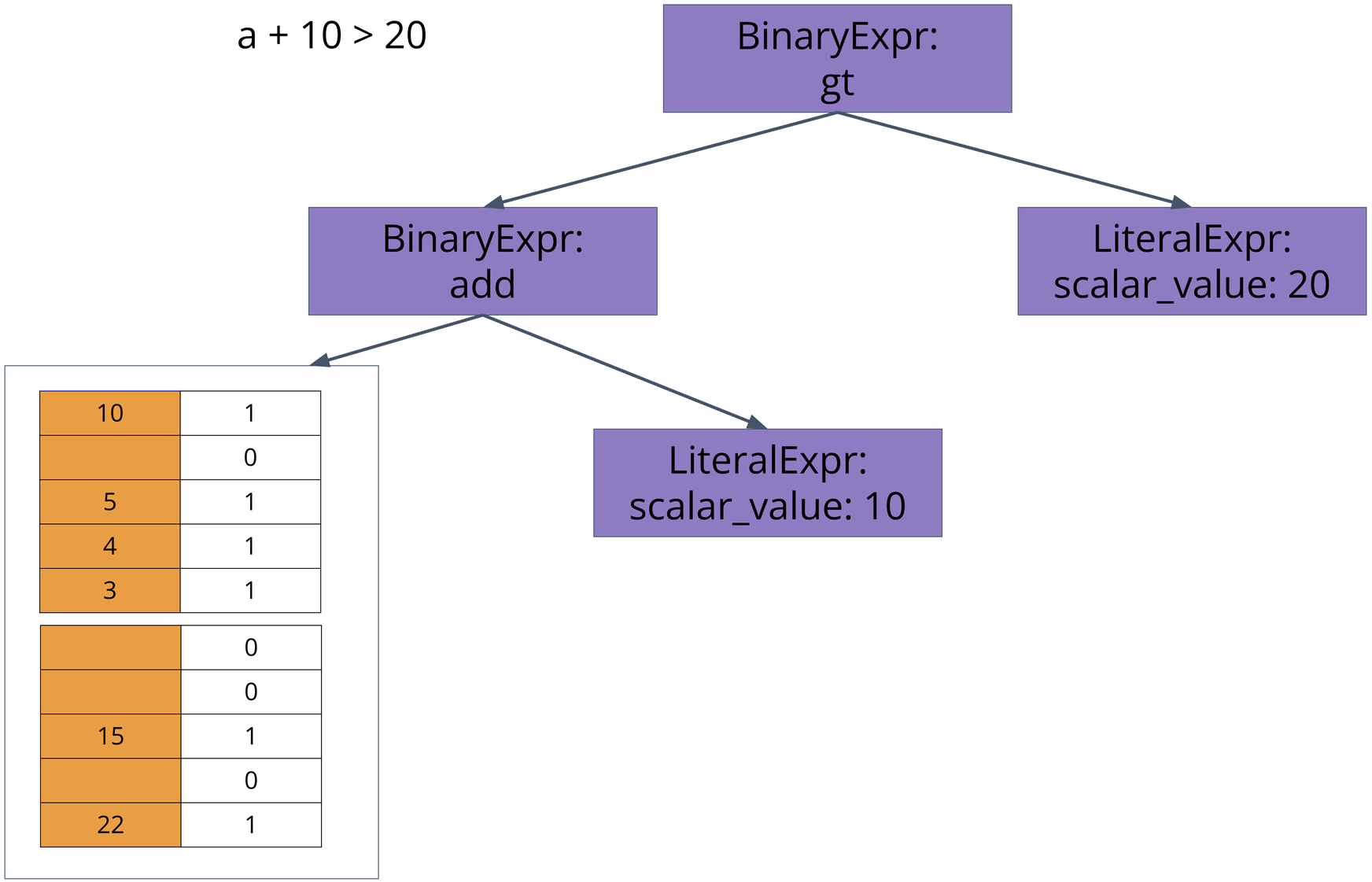


The design of Apache DataFusion: Performance

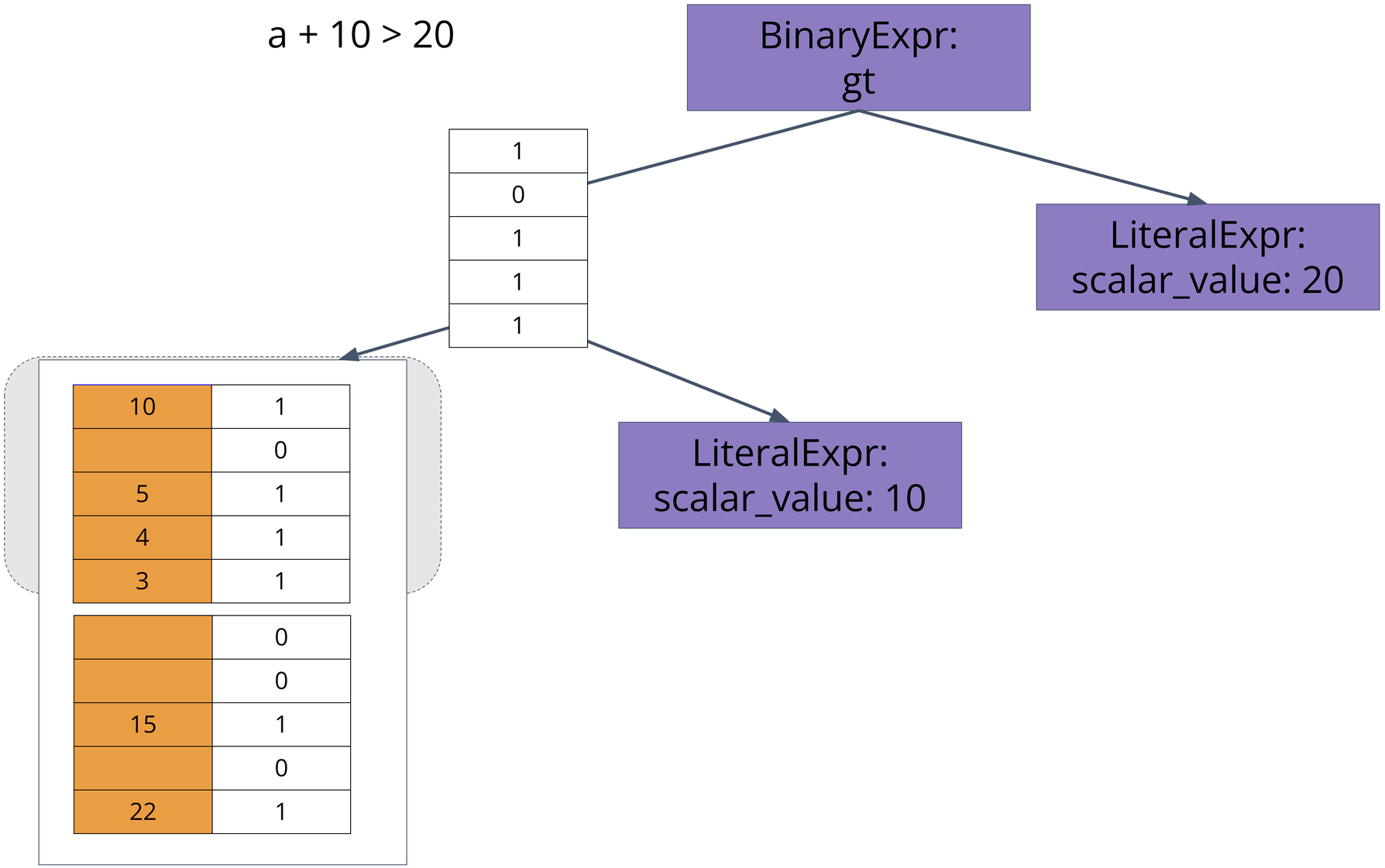
Vectorization: Expr evaluation
Arrow kernel supports some basic evaluation



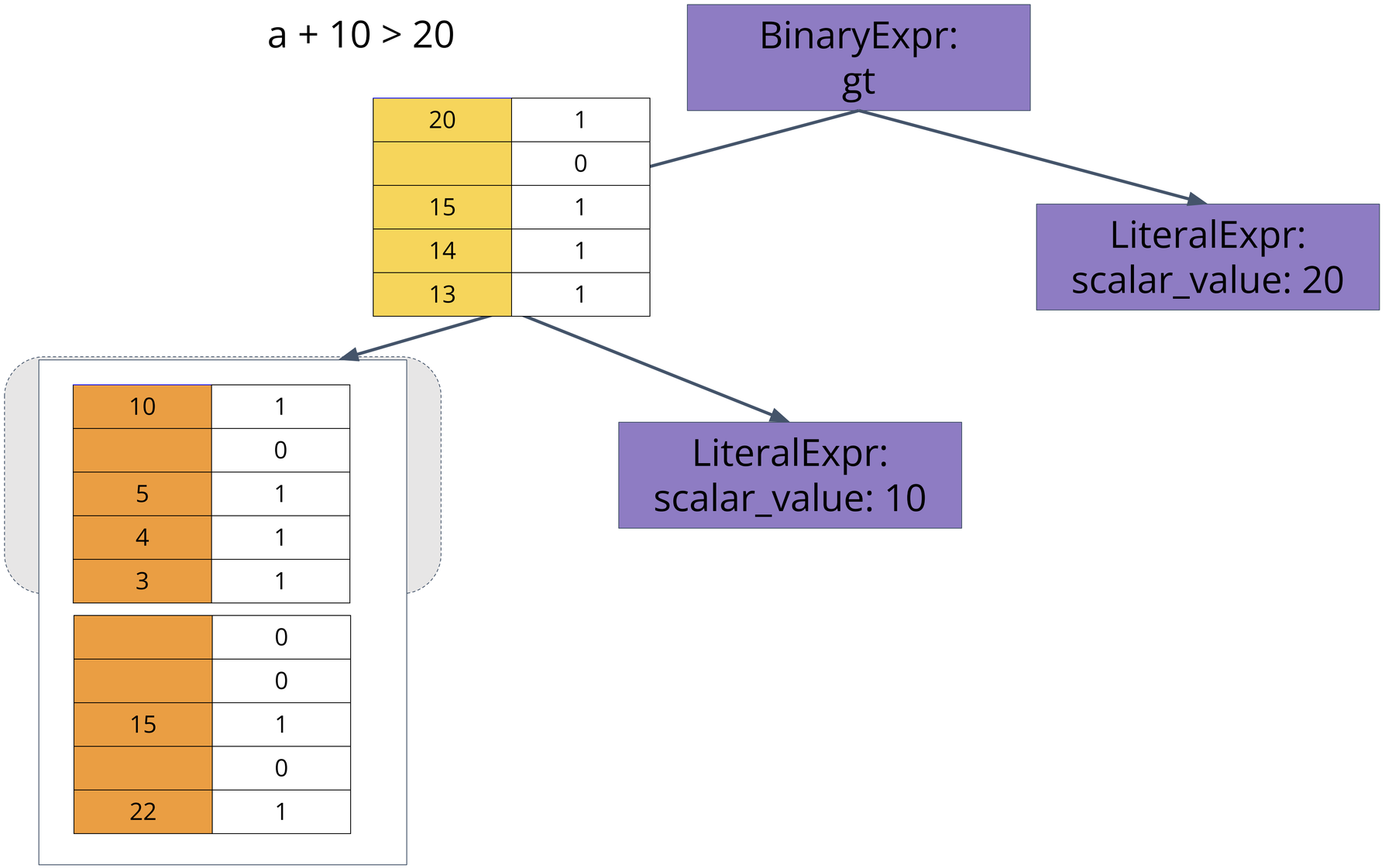
The design of Apache DataFusion: Performance



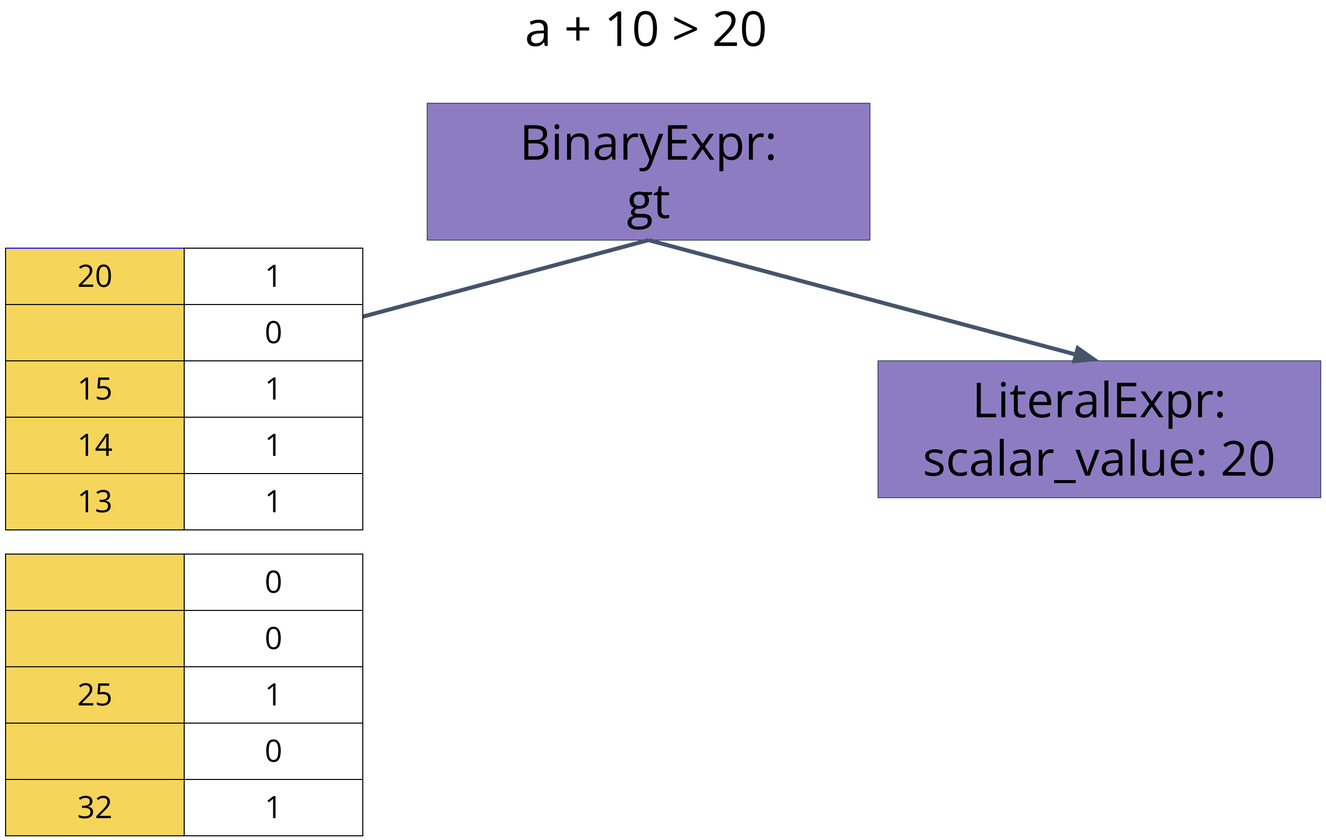
The design of Apache DataFusion: Performance



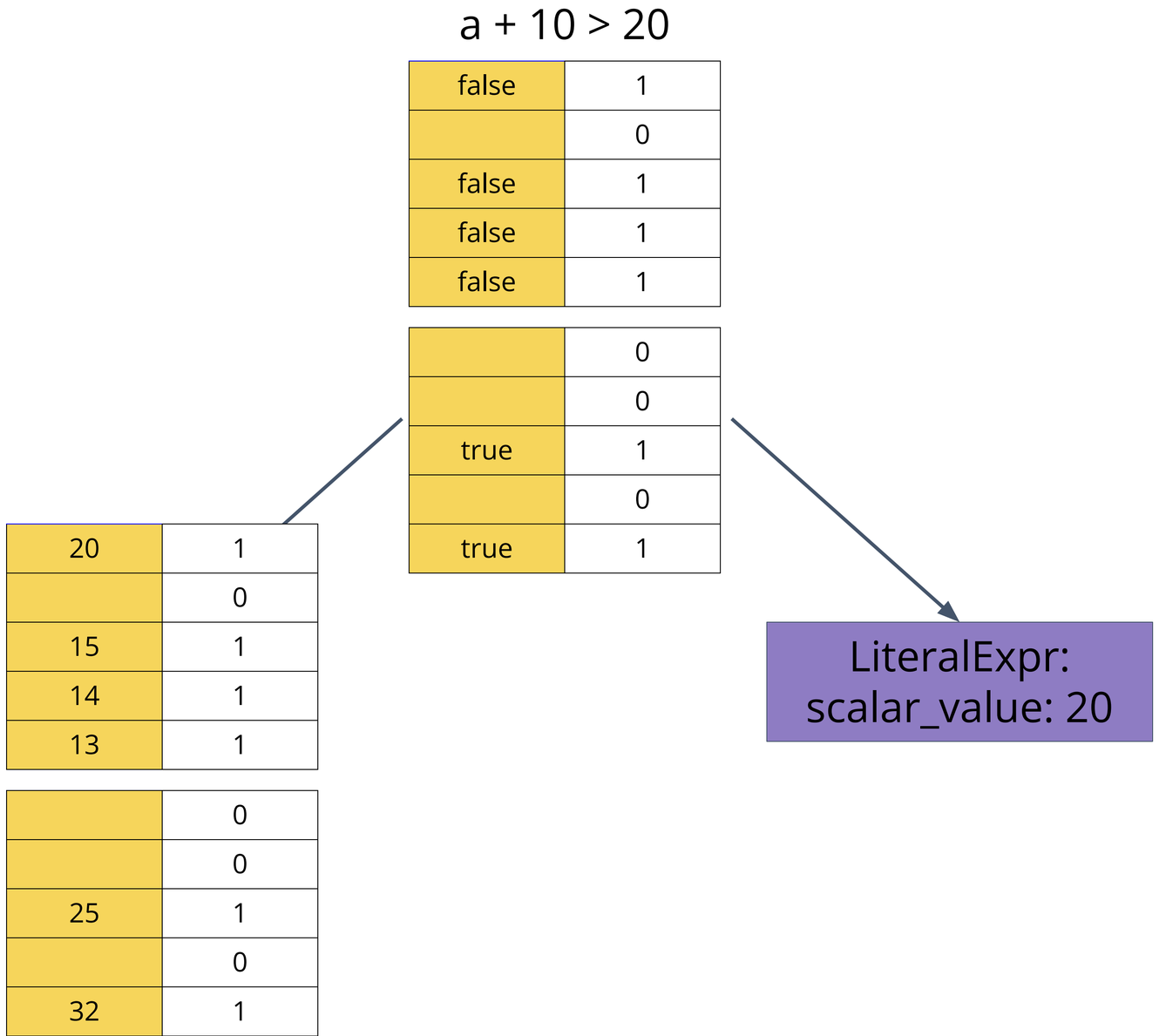
The design of Apache DataFusion: Performance



The design of Apache DataFusion: Performance



The design of Apache DataFusion: Performance



The design of Apache DataFusion: Performance

a + 10 > 20

false	1
	0
false	1
false	1
false	1

	0
	0
true	1
	0
true	1

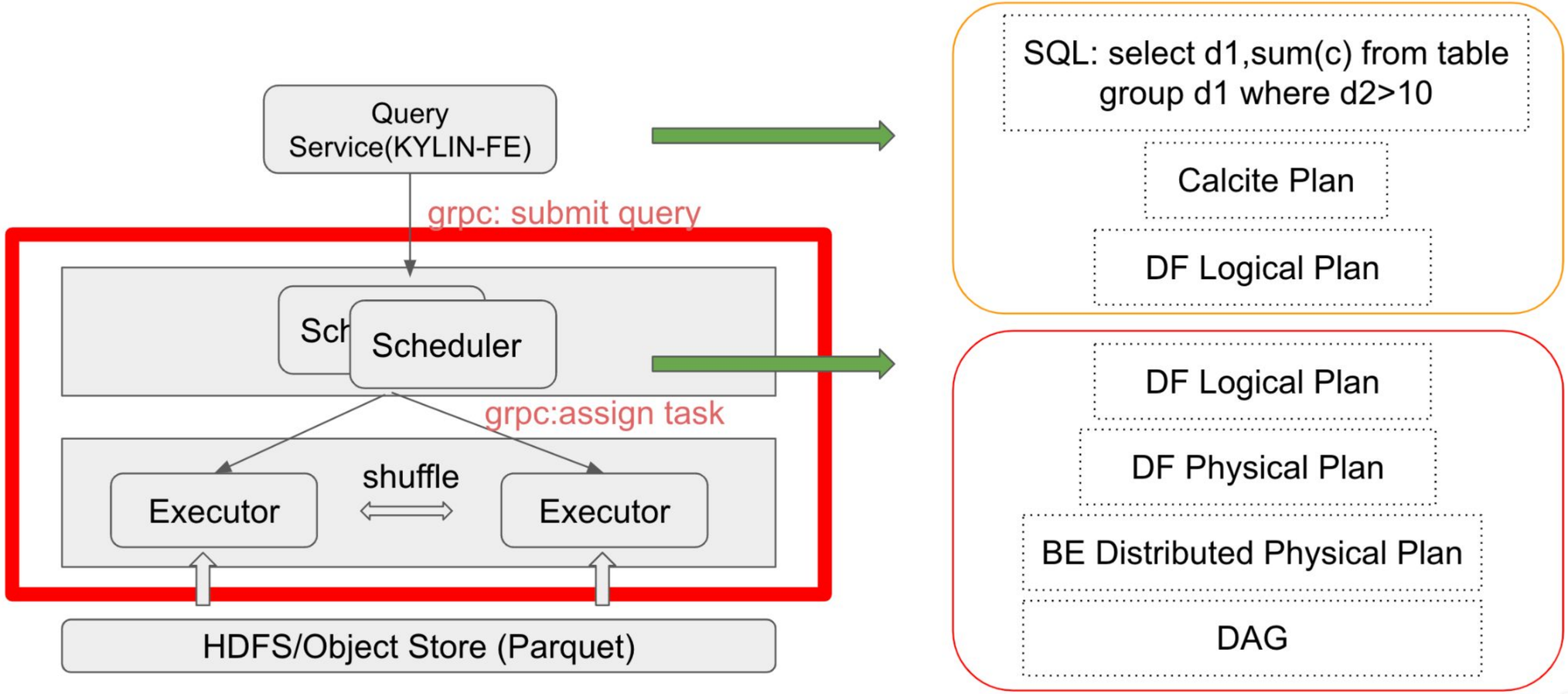
Part 04: User case of the Apache DataFusion in eBay

User case of the Apache DataFusion in eBay

- Background（2021.Q4 - 2022.Q1）
  - Shutdown Hbase cluster (planning)
  - Upgrade Kylin platform: Kylin3.1 -> Kylin5
    - Kylin5 support optional query engine
    - Community: Native query engine

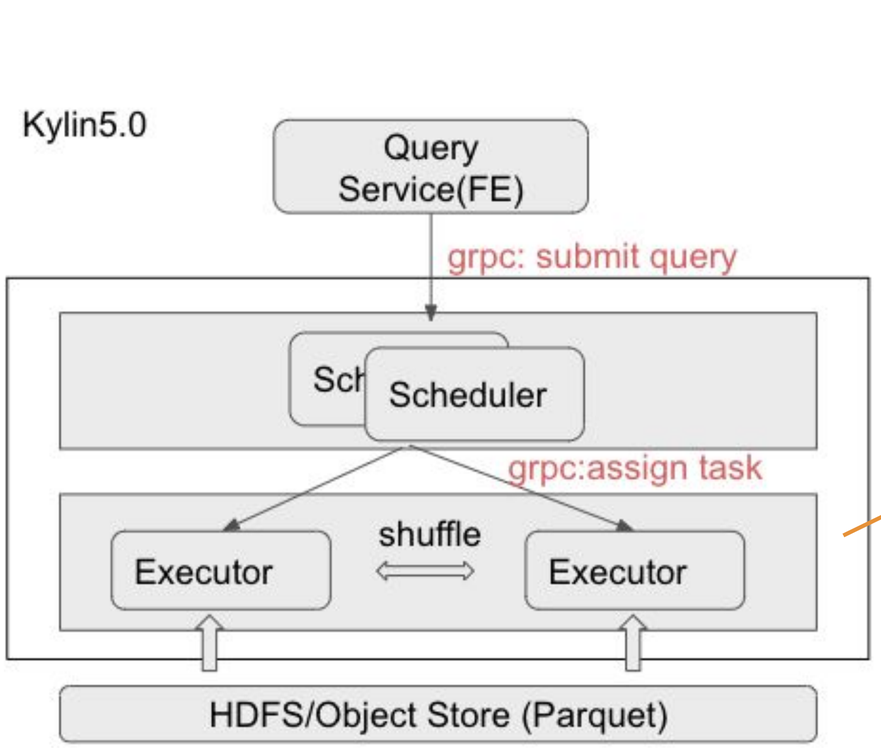
User case of the Apache DataFusion in eBay

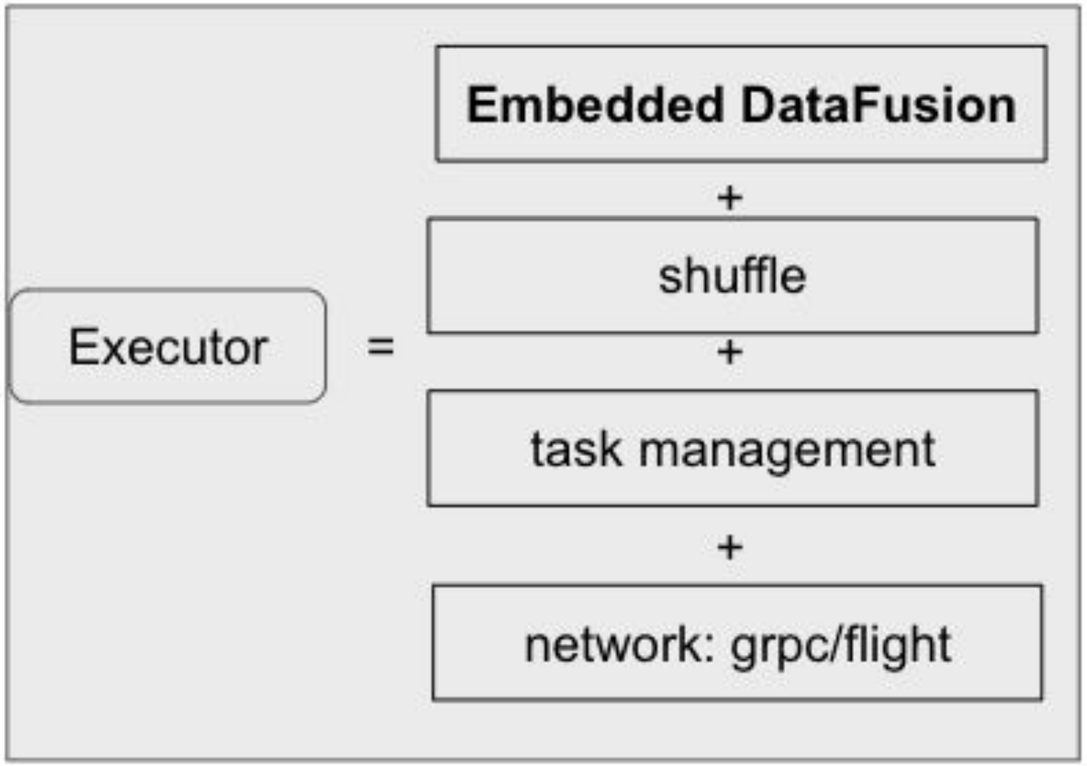
- Kylin5 + Ballista



User case of the Apache DataFusion in eBay

- Kylin5 + Ballista





User case of the Apache DataFusion in eBay

- Customization
  - Executor/DataFusion UDAF:
    - Add kylin UDAF using DataFusion UDAF framework

User case of the Apache DataFusion in eBay

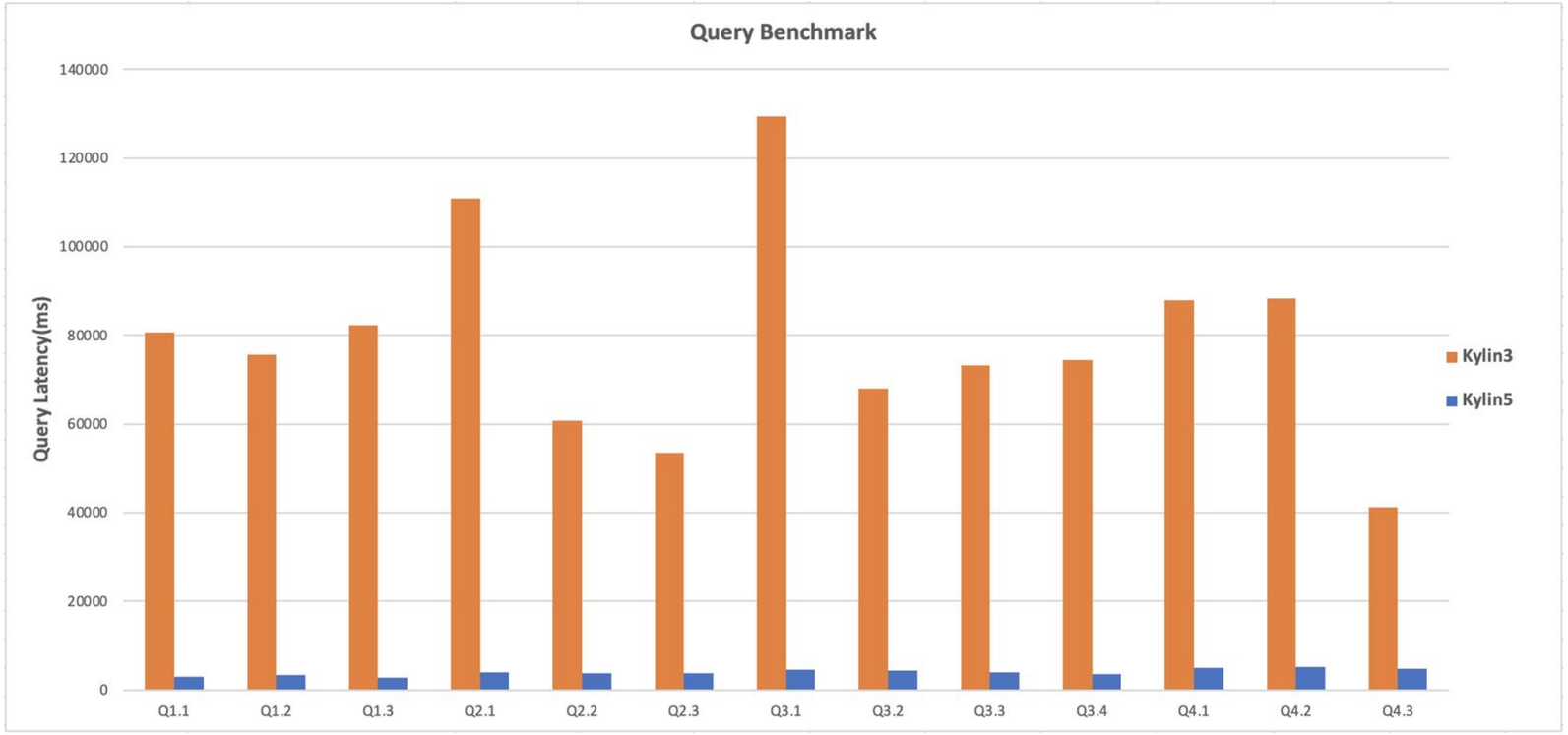
- Customization
  - Executor/DataFusion UDAF:
    - Add kylin UDAF using DataFusion UDAF framework

User case of the Apache DataFusion in eBay

- Customization
  - Ballista
    - Executor Local disk cache
    - List files cache(optional) implementation
    - File statistics cache(optional) implementation

User case of the Apache DataFusion in eBay

- Result: Kylin 5 has average **20 times** performance gain than Kylin 3.0 on SSB Test Set (1TB size, 6B rows).



Thanks