Introduction to Apache Cloudberry™ (Incubating)

Shine Zhang

Co-founder
Synx Data Labs Inc.

Disclaimer

Apache Cloudberry is an effort undergoing incubation at The Apache Software Foundation (ASF), sponsored by the Apache Incubator. Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects. While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.

Greenplum® is a registered trademark of Broadcom Inc. Synx Data Labs and SynxDB are not affiliated with, endorsed by, or sponsored by Broadcom Inc. Any references to Greenplum are for comparative, educational, and interoperability purposes only.

Shine Zhang

Co-founder, Synx Data Labs

"Do the right thing. Do what works. Be kind."

Xin "Shine" Zhang is a technology leader with deep expertise in distributed databases, AI-driven analytics, and cloud platforms. Before co-founding Synx Data Labs, he was an Engineering Tech Lead at Broadcom VMware Tanzu Data, leading enterprise data migration and next-gen database platform development.

With over 20 years of experience in MPP databases like Greenplum and PostgreSQL, Shine has been a core contributor to open-source projects, including Apache Cloudberry (Incubating), Greenplum, and Apache Geode. At Synx Data Labs, he is focused on advancing distributed data systems and AI-powered analytics solutions.

A lifelong learner and open-source contributor, Shine enjoys experimenting with AI models, mentoring engineers, and exploring new places when he's not building data platforms.

LinkedIn || shine@synxdata.com

APACHE Cloudberry™

Shine Zhang

SYNX

Xin Zhang

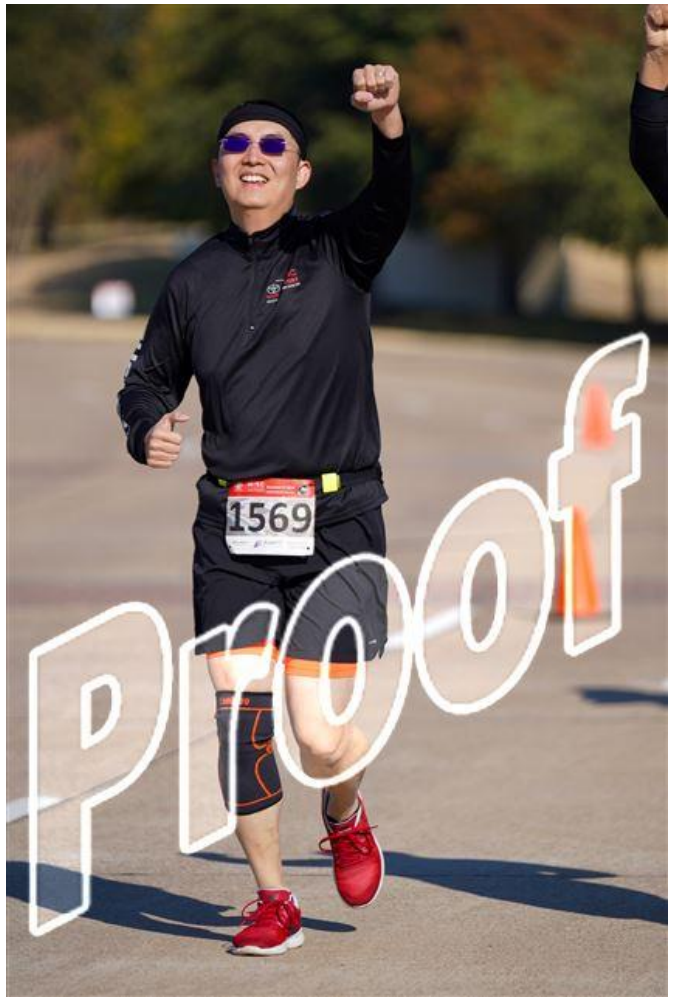
2016 Lake Sammamish Half Marathon

2:14:25

Overall Placement: 1067th of 1,749
M 30-39: 218th of 270
All Male: 575th of 753
Average Pace: 10:16

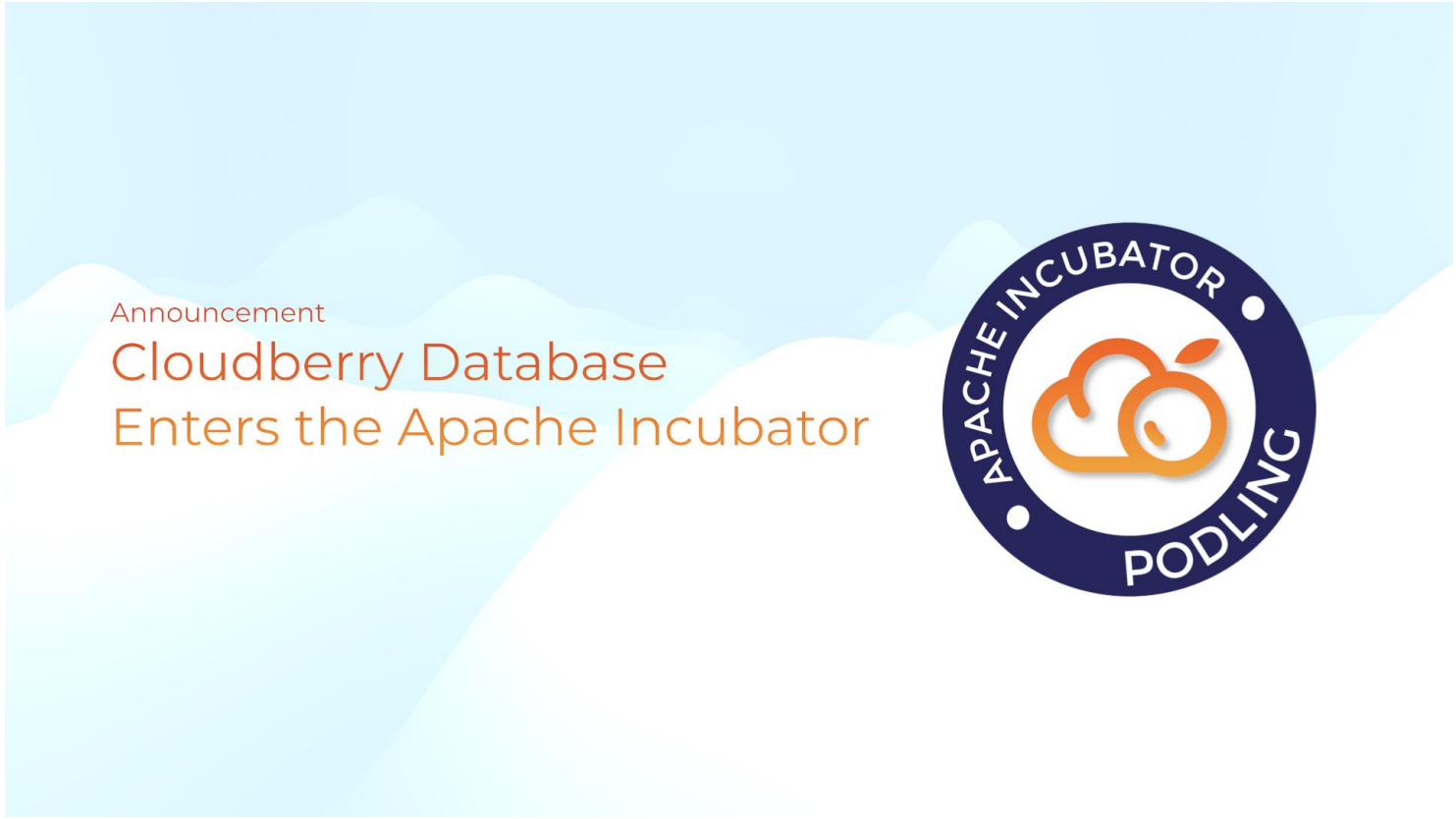
Results presented by
Lake Washington Physical Therapy
and powered by Clif

Shine Zhang



Announcement

Blog: https://s.apache.org/9xic9



Scan to read and share

Agenda

- History
- Internals
- Unique Enhancement
- Feature Comparison
- Extended Scenarios
- Future Works

History

Milestones

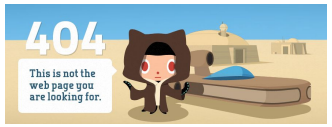
Created by the original Greenplum developers

First Commit

Early June 2022

Open Source in GitHub

June 2023



Greenplum was archived and went closed-source

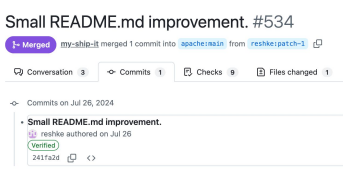
May 2024

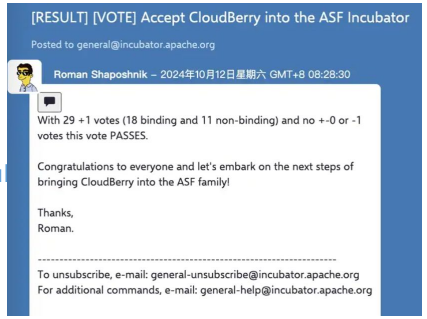
The first external contributor to GitHub

The first external discussion post on GitHub

GitHub 🌟 over 200

Early June 2024





Join the Apache Incubator on October 12, 2024

Project repos migrated to Apache on November 15, 2024

GitHub 🌟 900+

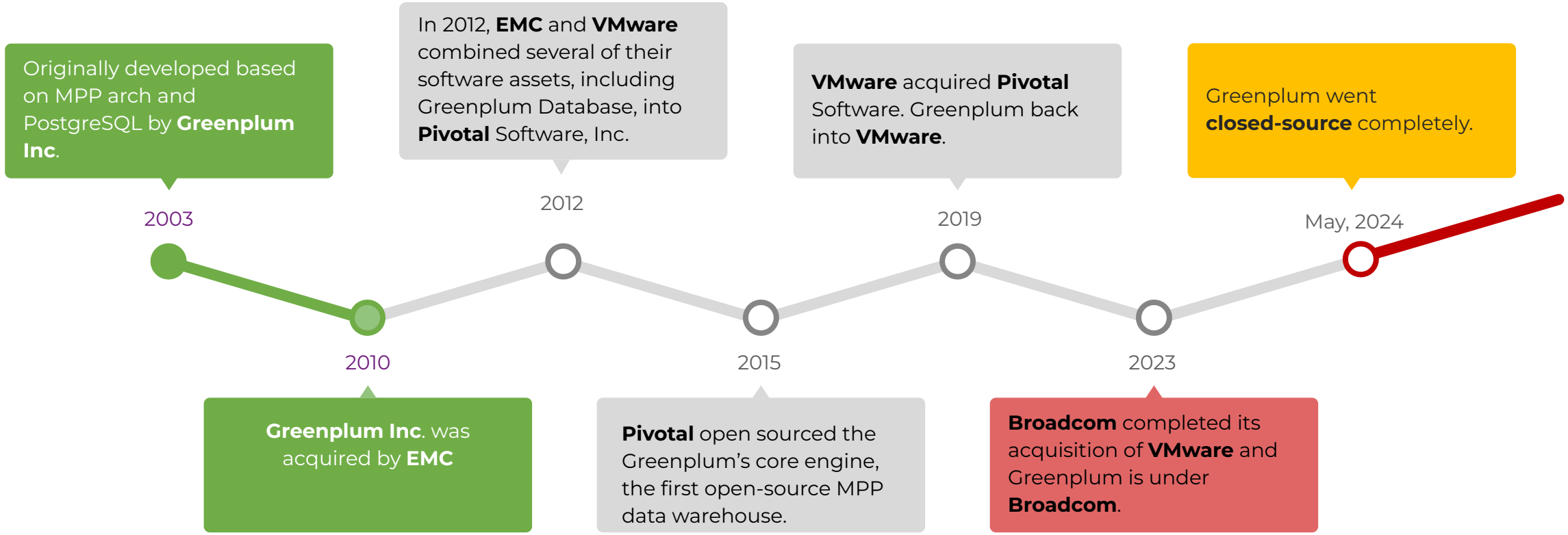
```
commit 10538adb70ac5e81abbe0cb16a9e1cf66ad808ec
Merge: d5573b984b d457cb4e8a
Author: Cloudberry <cloudberry@initial.code.dump>
Date:   Wed Jun 7 08:57:14 2023 +0000

    Initial Cloudberry code dump.
```

8,635 changed files with 1,329,364 additions and 224,191 deletions.

- GitHub - apache/cloudberry: Cloudberry Database - Open source alternative to Greenplum Database. Cre
- GitHub - apache/cloudberry-site: Website sources for Cloudberry Database
- GitHub - apache/cloudberry-bootcamp: Help you quickly try out CloudberryDB via one Docker-based Sand
- GitHub - apache/cloudberry-gpbackup: Backup Utility for CloudberryDB
- GitHub - apache/cloudberry-gpbackup-s3-plugin: S3 plugin for use with Cloudberry Database backup uti
- GitHub - apache/cloudberry-pxf: Platform Extension Framework (PXF) for Cloudberry Database
- GitHub - apache/cloudberry-go-libs: gp-common-go-libs for CloudberryDB.
- GitHub - apache/cloudberry-devops-release

History of Greenplum



Greenplum --> Apache Cloudberry!

The **ownership** of Greenplum Database has changed many times, causing concerns among ecosystem partners as well as users and developers of the open-source Greenplum community.

Greenplum Database has long **lost its innovative** momentum, lagging behind user needs in performance, cloud-native capabilities, data lakes, and other areas, failing to keep up with modern industrial trends, and its competitiveness is diminishing.

More importantly, Greenplum has always been controlled by a single vendor, and there is **no open governance** model that allows the community to participate in decision-making and voting.

#1

2022: Release product

First forked in **2022** as a branch of **Greenplum 7 Beta 3**. The product is not just a simple Greenplum replacement; it brings many advanced features and highlights, achieving multi-form data management and multi-modal analytical computation.

#2

2023: Open source

Then opened source code in **2023**, with the license under **Apache License V 2.0**. The source code is hosted on GitHub.
It is compatible with Greenplum, allowing users to use it just as they would use Greenplum.

#3

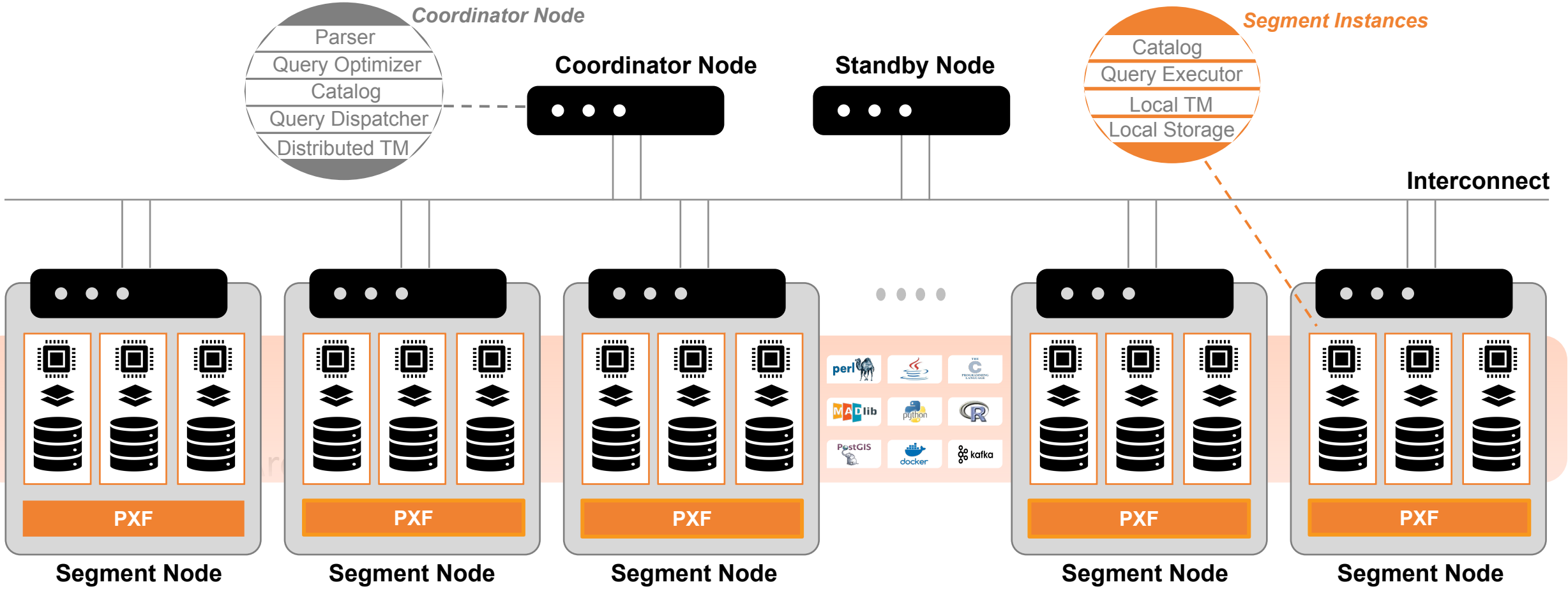
2024: Join ASF Incubator

It was voted to enter the **Apache Incubator at Oct 2024**. This would ensure that Apache Cloudberry develops along the lines of open governance and community-driven principles, thus avoiding the single-vendor control similar to that of Greenplum.

Internals

Architecture

APACHE Cloudberry™



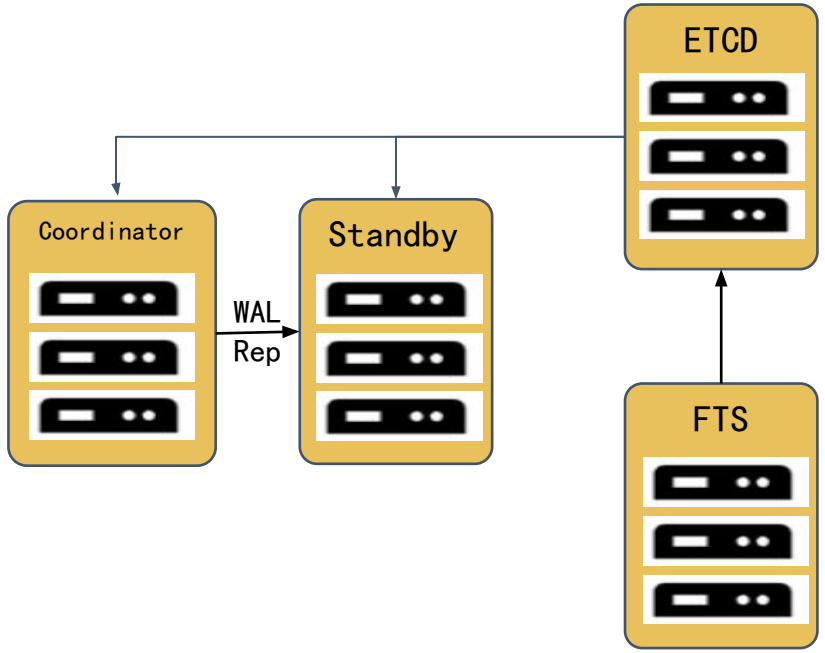
MPP Shared-Nothing Architecture, powered by PostgreSQL 14.4

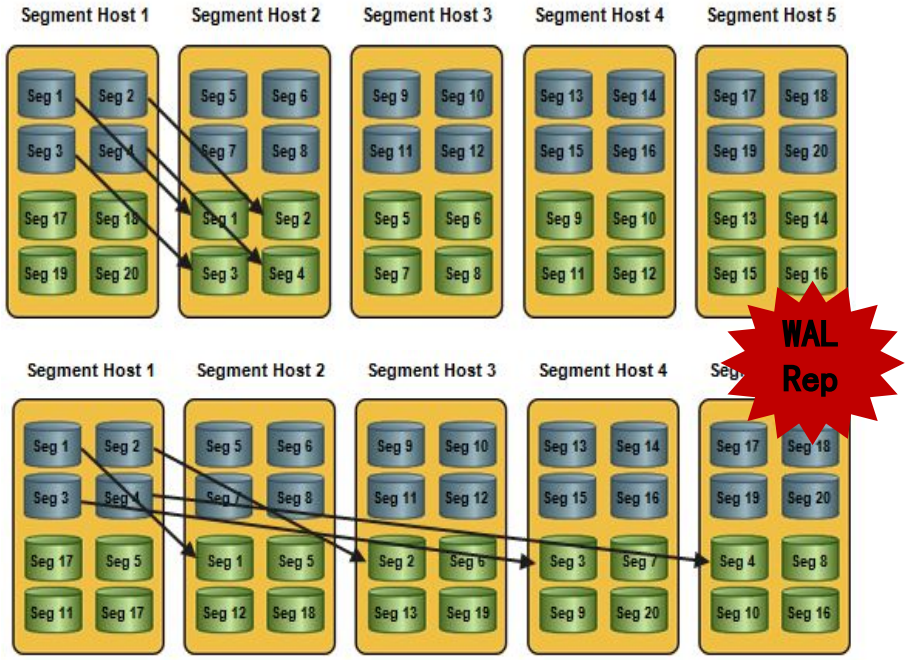
SYNX

High Availability

APACHE Cloudberry™

At the software level, Cloudberry provides a complete high-availability design for both data nodes and computing nodes





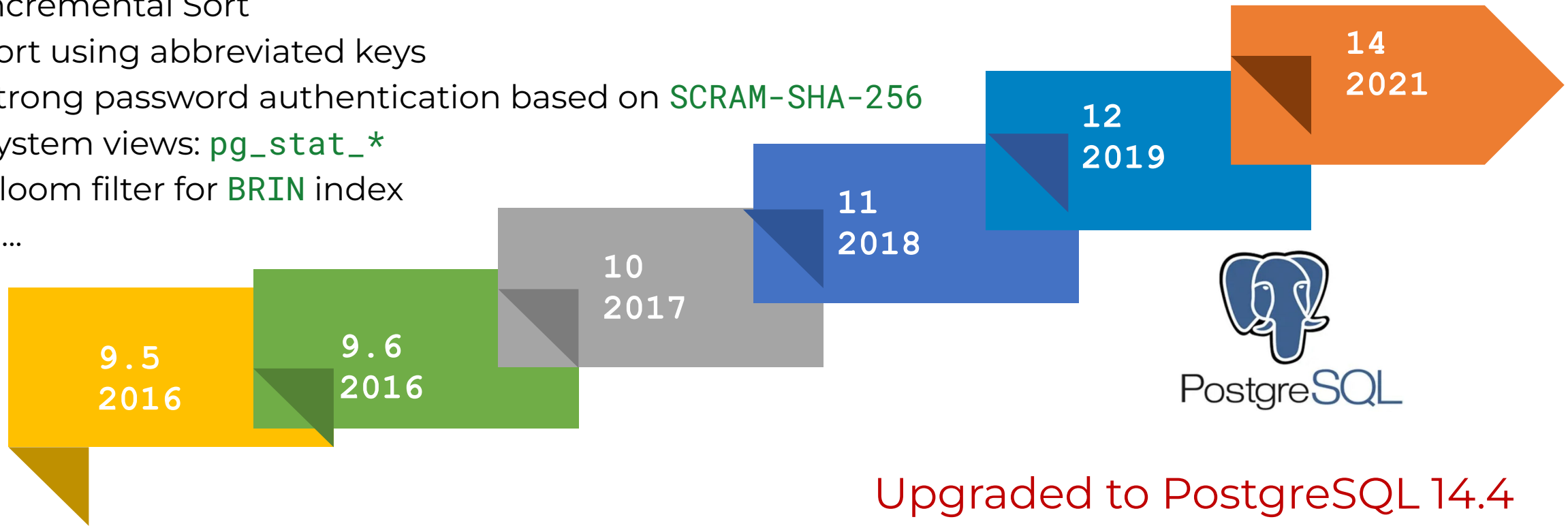
- ETCD is used to store cluster topology information and cluster status metadata information.
- Coordinator and Standby nodes keep WAL replication. When the Coordinator node fails, it automatically switches without manual intervention.
- Data nodes mirroring through WAL replication.
- two types of mirroring:
  - group mirroring (TOP: higher reliability)
  - spread mirroring (BOTTOM: higher performance)

SYNX

Postgres Kernel Upgrade

APACHE Cloudberry™

- Support multiple types: multi-range, JSON, JSONB, XML, and etc;
- UPSERT: `INSERT ... ON CONFLICT` syntax
- Hash Partitioned Table
- `REINDEX` concurrently
- Incremental Sort
- Sort using abbreviated keys
- Strong password authentication based on `SCRAM-SHA-256`
- System views: `pg_stat_*`
- Bloom filter for `BRIN` index
- .....

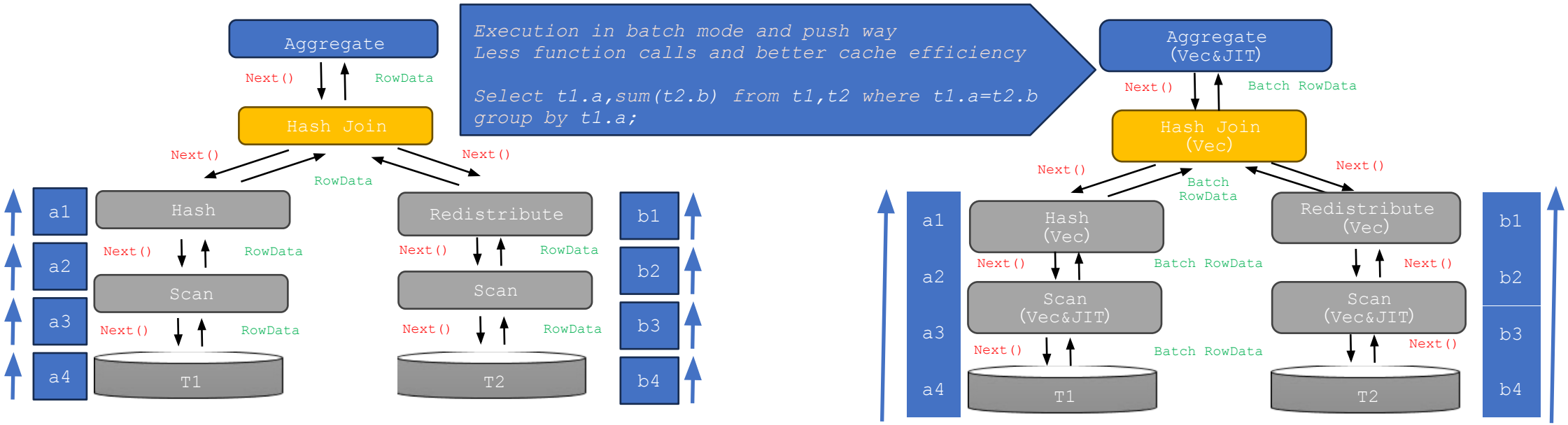


Upgraded to PostgreSQL 14.4
ANSI SQL 2011 support

SYNX

Unique Enhancement

Vectorization Execution Engine



- Default ON: setting **vector.enable_vectorization**
- **Data Types**: int2/4/8、float8、bool、char、tid、date、time、timestamp、timestamptz、varchar、text and etc
- **Operators**: Scan、Agg、ForeignScan、Result、Append、Subquery、Sequence、NestedLoopJoin、HashJoin、Sort、ForeignScan、Expresion and etc
- Leverage **threads** instead of processes to speed up

- **Batching** multiple rows with single `Next()`
- Increased **CPU** utilization
- **SIMD** with columnar storage
- Favor columnar storage **scanning**

Federated Query Across Multiple Clusters

```
EXPLAIN (COSTS OFF) SELECT * FROM fs1, fs2 WHERE fs1.a = fs2.a AND fs1.gp_
foreign_server = fs2.gp_foreign_server;
```

```
           QUERY PLAN
---
Gather Motion 3:1  (slice1; segments: 3)
   ->  Foreign Scan
         Relations: (fs1) INNER JOIN (fs2)
Optimizer: Postgres query optimizer
(4 rows)
```

- Through **`fdw`** to query between multiple Cloudberry clusters, unlike *`postgres_fdw`*, the data is not gathered by Query Dispatcher node
- Hint the optimizer through *`gp_foreign_server`* to perform join operations on local cluster

---

```
EXPLAIN (COSTS OFF) SELECT count(*) FROM fs1, fs2 WHERE fs1.a = fs2.a AND
fs1.gp_foreign_server = fs2.gp_foreign_server;
```

```
                      QUERY PLAN
---
Finalize Aggregate
   ->  Gather Motion 3:1  (slice1; segments: 3)
         ->  Foreign Scan
               Relations: Aggregate on ((fs1) INNER JOIN (fs2))
Optimizer: Postgres query optimizer

(5 rows)
```

- The optimizer will push `Join` or `Agg` operators down to the target Cloudberry cluster to reduce the amount of intermediate data transmission.

Perf Optimization: Aggregation over Join Pushdown

```
EXPLAIN SELECT o.order_id, SUM(price)
  FROM order_tbl o, order_line_tbl ol
  WHERE o.order_id = ol.order_id
  GROUP BY o.order_id;
                                        QUERY PLAN
-------------------------------------------------------------------------------------
 Gather Motion 3:1  (slice1; segments: 3)  (cost=712.89..879.56 rows=10000 width=12)
   ->  Finalize HashAggregate  (cost=712.89..746.23 rows=3333 width=12)
         Group Key: o.order_id
         ->  Hash Join  (cost=617.00..696.23 rows=3333 width=12)
               Hash Cond: (ol.order_id = o.order_id)
               ->  Partial HashAggregate  (cost=538.00..571.38 rows=3338 width=12)
                     Group Key: ol.order_id
                     ->  Seq Scan on order_line_tbl ol  (cost=0.00..371.33 rows=33333
               ->  Hash  (cost=37.33..37.33 rows=3333 width=4)
                     ->  Seq Scan on order_tbl o  (cost=0.00..37.33 rows=3333 width=4)
 Optimizer: Postgres query optimizer
```

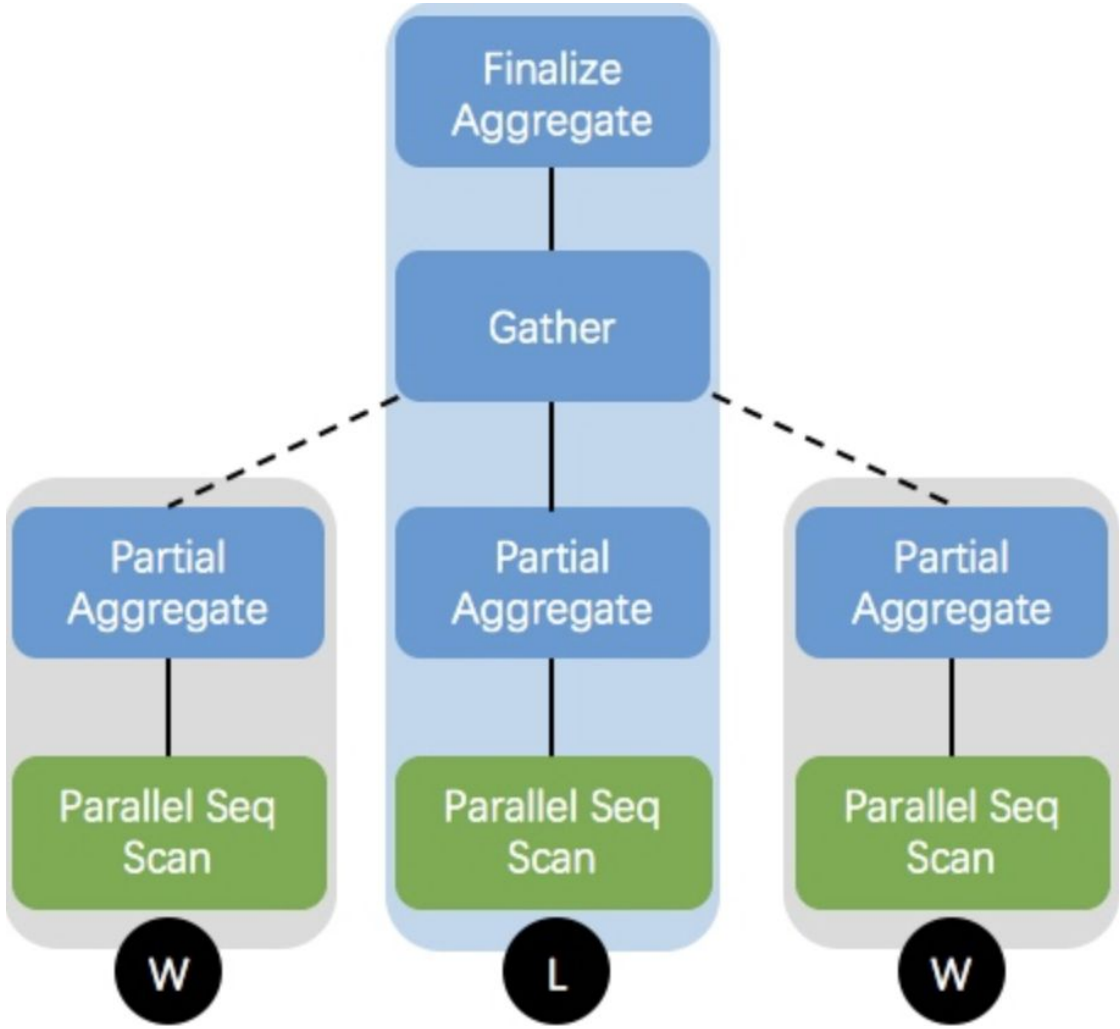
- big win for OLAP cases
- push `Agg` calculation below the `Join`, dramatically reducing the amount of data to join
- cost based decision

Parallel Execution

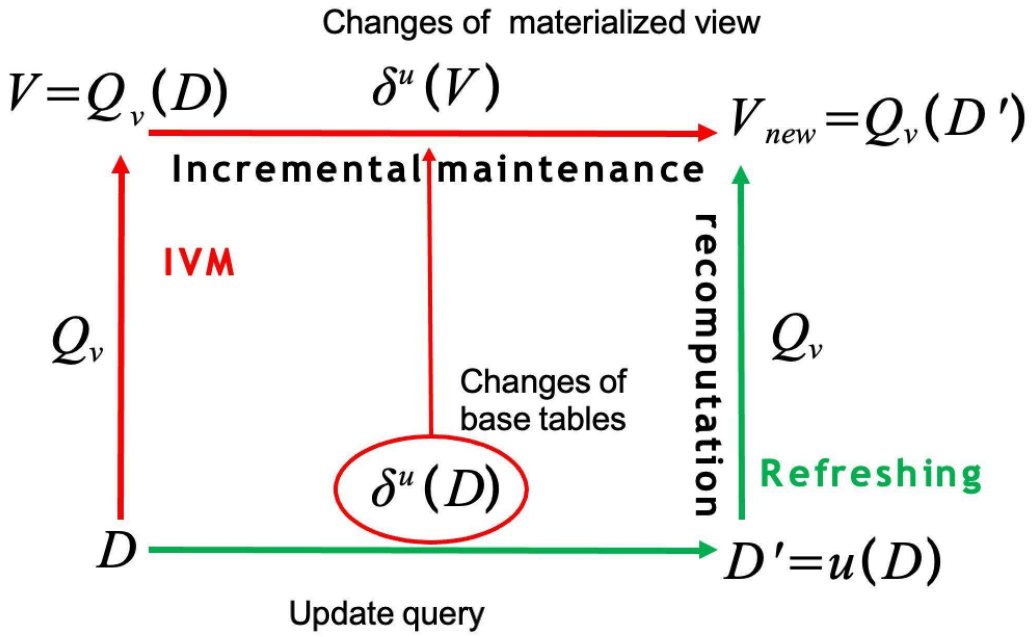
- **Multiple** CPU cores on a **single** query
- Avoid scalability issue due to high # of segments
- Dynamic DOP by data size

Supported Operators:

- sequence scan
- index scan
- index only scan
- bitmap heap scan
- append
- hash join
- nested loop join
- merge join
- ……

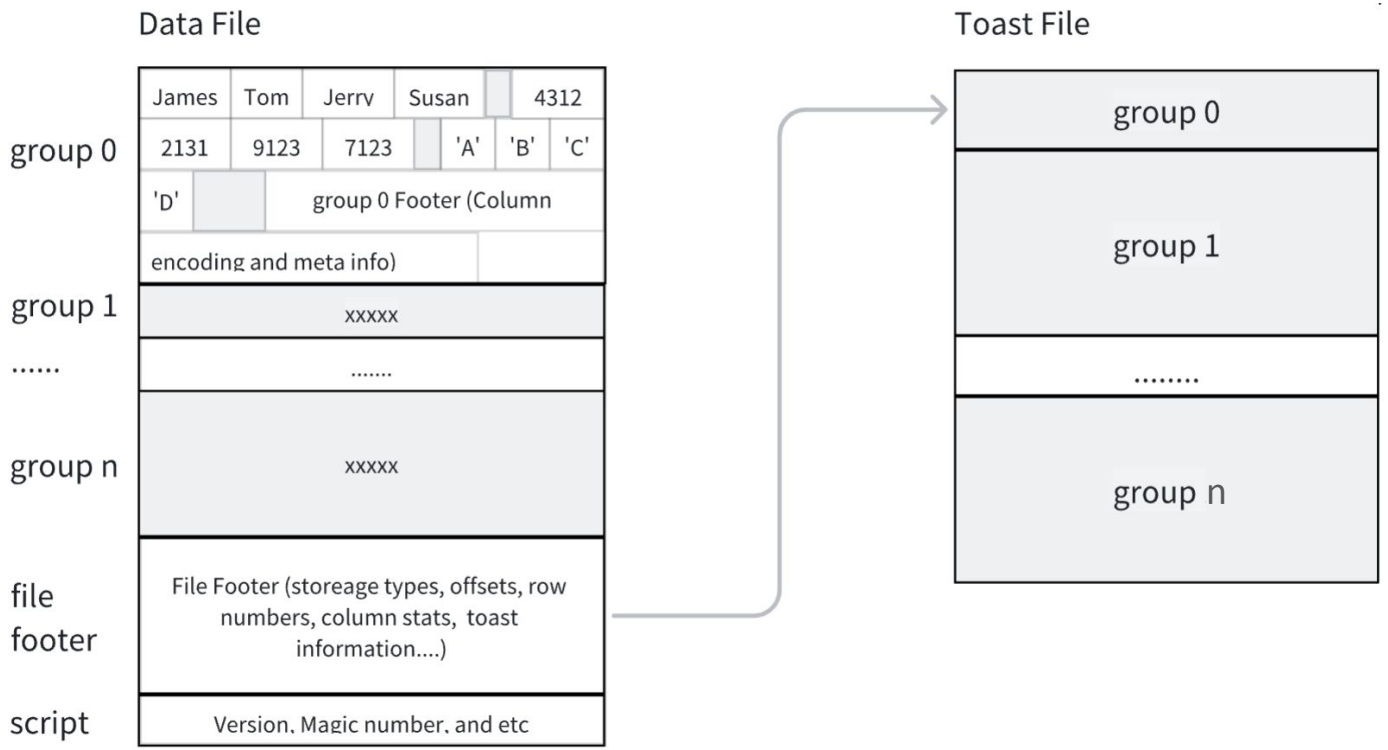


Incremental Materialized View and Query Rewrite



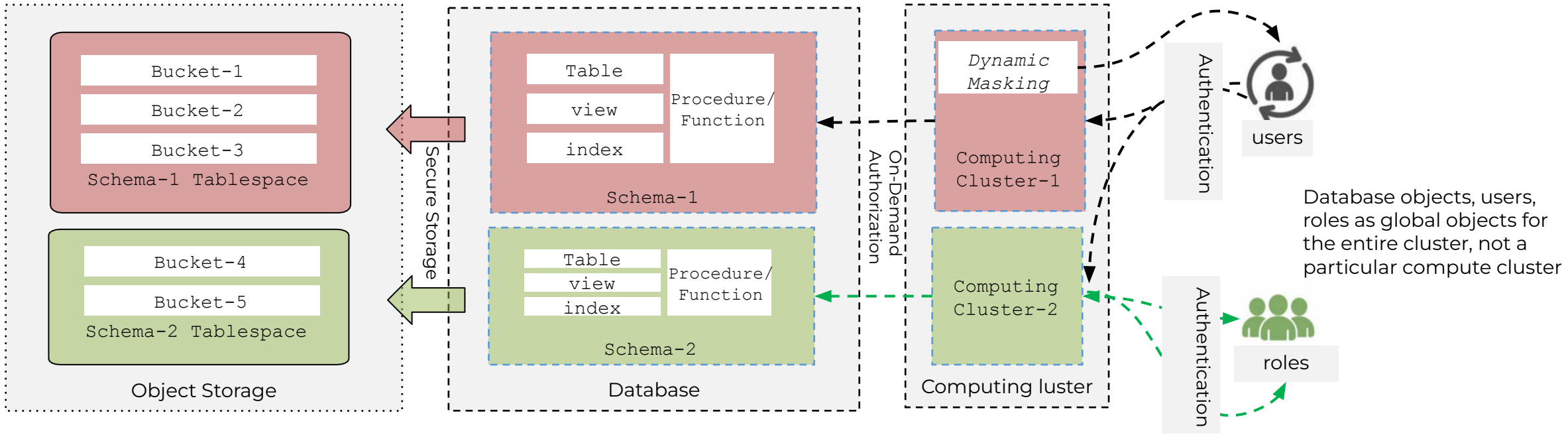
1. **Incremental** maintenance based on data updates
2. Materialized view rewriting can be **synchronous** or **asynchronous**
3. query **rewriting** rules for various statements (Filter, Join, Agg, etc.)
4. optimizer replace the "subtree" with matching materialized view
5. **cost** based decision with original alternatives
6. for **async** materialized view, the $\delta^u(D)$ is part of query rewriting

PAX: Hybrid Row-Column Storage Format



1. Filter push-down, column projection, with **runtime filter**
2. **Combined IO** request of continuous columns
3. **All** data types: fixed and variable
4. Data **compression** and **encoding**
5. File Block & Group **skip scanning**
6. **Vectorization** native support
7. Toast for **large** content data
8. **Clustering** and ordering by columns

Built-in Data Security



Secured Storage

- data encryption with **pgcrypto**:
  - ✔ Asymmetric: SM2 , RSA;
  - ✔ Symmetry: SM4, AES;
  - ✔ Hash: SM3;
- Transparent Database Encryption (**TDE**) supports: AES, SM4.

On-Demand Authorization

- User **whitelisting**
- **Hierarchical** Permissions:
  - various **permissions** (Select, Update, Execute, Ownership, etc.)
  - on multiple **levels** of objects (e.g., Schema, tables, views, etc.)
  - for individual **users**
- **Row Level** security.

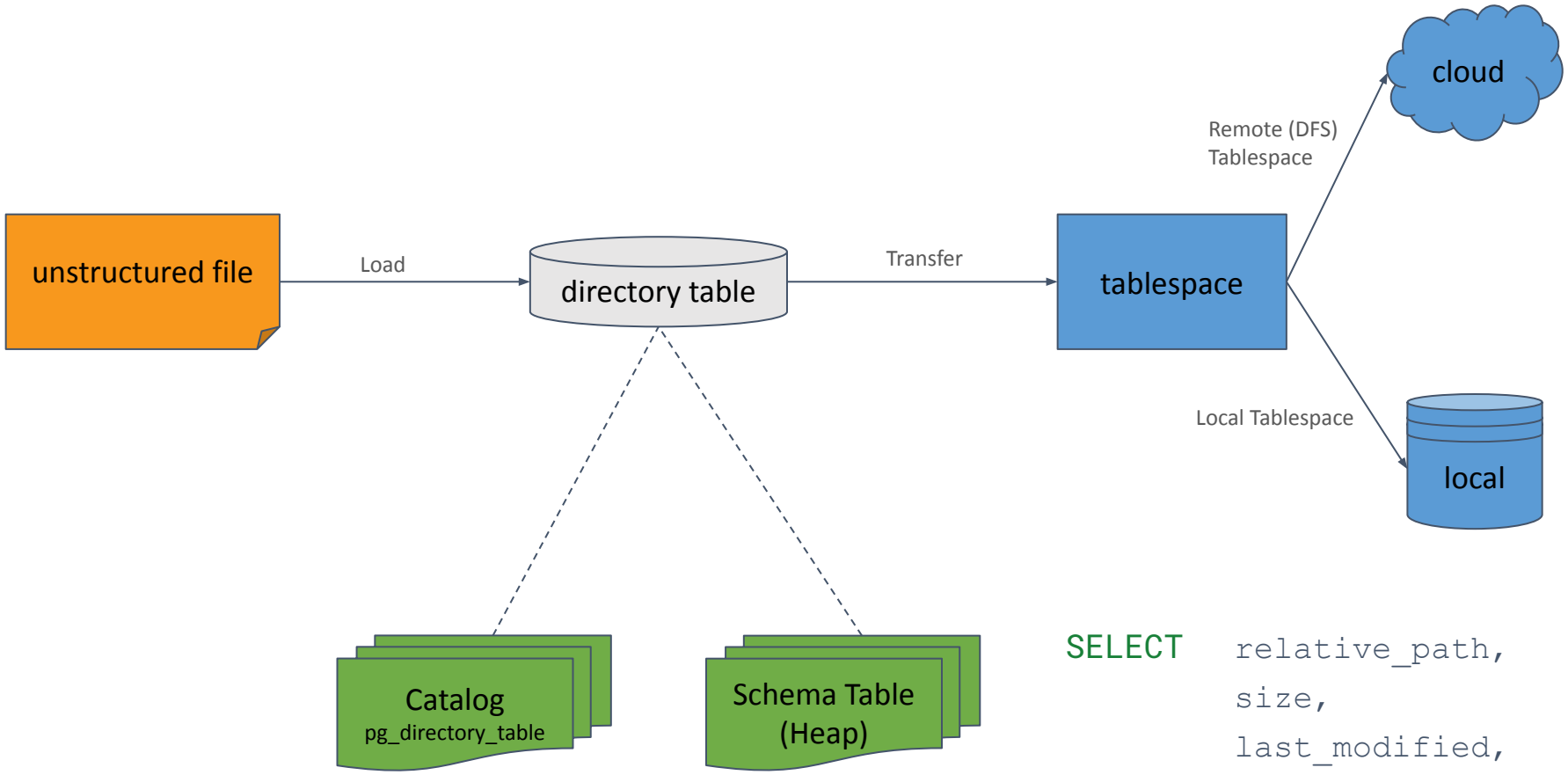
Dynamic Data Masking

- **secure data sharing** with sensitive info protection
- Random, Faking, SHA, custom functions **data masking**

Unified Authentication

- ciphertext **authentication** through:
  - ✔ MD5, SHA-256, etc.
  - ✔ Kerberos authentication
- **password strength** checking (length, alphanumeric non-alphanumeric, etc.).
- **password policies** (number of failures, fail locks, number of repeats);
- central authentication servers, e.g., **LDAP**.

Unstructured Data Management: Directory Table




```
SELECT   relative_path,
         size,
         last_modified,
         md5,
         tags,
FROM directory_table(dir_table);
```

Feature Comparison

Kernel Features

Feature names	Cloudberry Database	Greenplum
`EXPLAIN` (WAL) support	✅	❌
Multiranges	✅	❌
B-tree bottom-up index deletion	✅	❌
Covering indexes for GiST (`INCLUDE`)	✅	✅ (Upcoming)
The `range_agg` range type aggregation function	✅	❌
`CREATE ACCESS METHOD`	✅	✅ (Upcoming)
LZ4 compression for `TOAST` tables	✅	❌
JSONB subscripting	✅	❌
Configure the maximum WAL retention for replication slots	✅	❌
Verify backup integrity (`pg_verifybackup`)	✅	❌
Client can require SCRAM channel binding	✅	❌
Vacuum "emergency mode"	✅	❌
Certificate authentication with `postgres_fdw`	✅	❌
`UPSERT`	✅	✅ (Upcoming)
COPY FROM Where	✅	❌
VACUUM / ANALYZE Skip Lock Table	✅	❌
HASH partitioned table	✅	❌
CTE (SEARCH and CYCLE)	✅	❌
Procedure OUT parameters	✅	❌
`CHECK` constraints for foreign tables	✅	❌
Timeout parameter for `pg_terminate_backend`	✅	❌
Auto failover for coordinator	✅	❌
Kubernetes deployment support	✅	❌

Performance Features

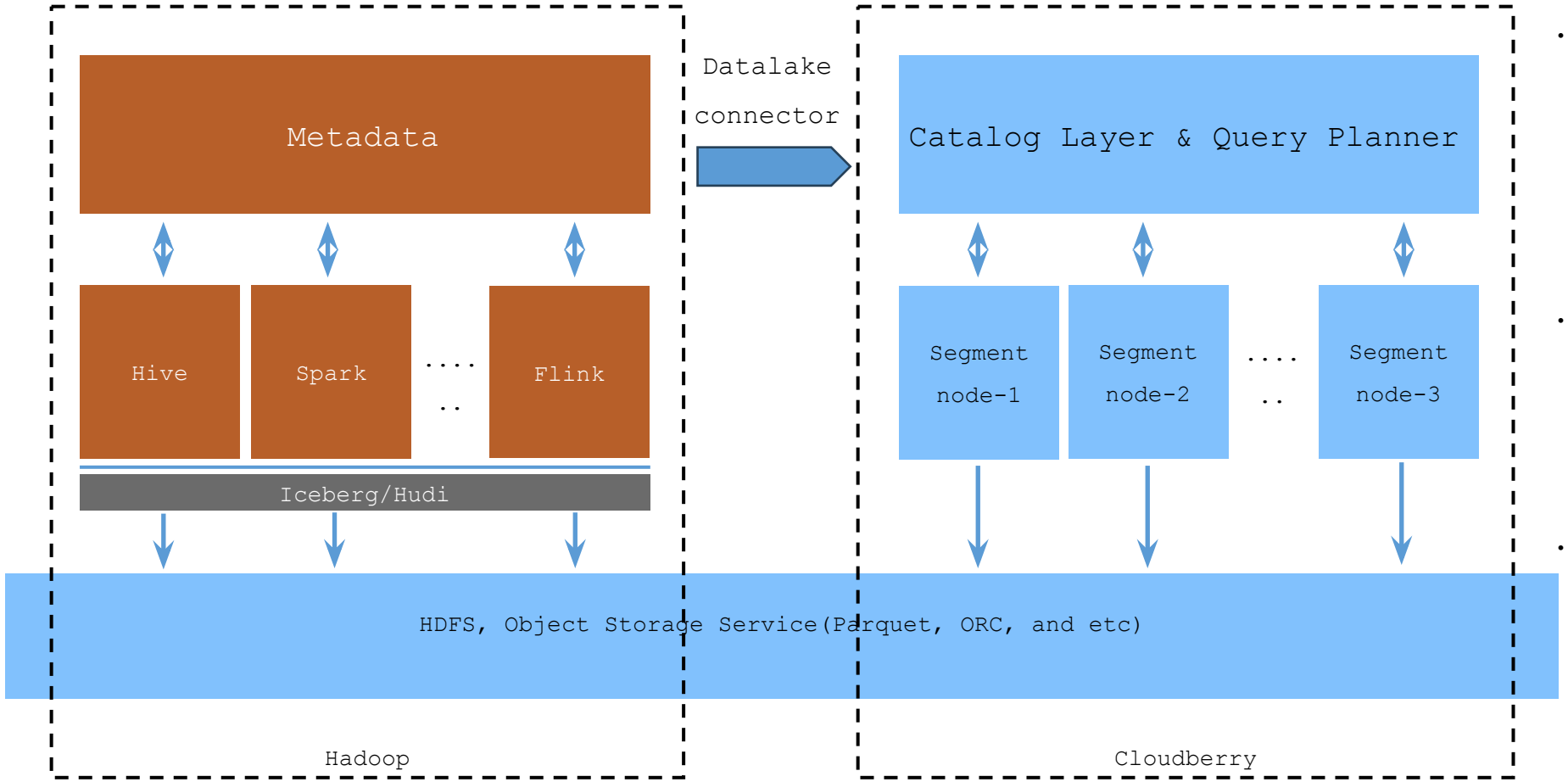
Feature names	Cloudberry Database	Greenplum
`REINDEX CONCURRENTLY`	✅	❌
Aggregation pushdown	✅	❌
`CREATE STATISTICS` - `OR` and `IN/ANY` statistics	✅	❌
Incremental sort	✅	❌
Incremental sort for window functions	✅	❌
Query pipelining	✅	❌
BRIN Index （multi-minmax, bloom）	✅	❌
Query parallelism	✅	❌
Abbrevated keys for sorting	✅	❌
Hash Index WAL support	✅	❌
`postgres_fdw` aggregation pushdown	✅	❌
No need to rewrite the whole table when adding a column	✅	❌
Runtime Filter for Join	✅	❌
Index Scan for the AppendOnly table	✅	❌

Security Features

Feature names	Cloudberry Database	Greenplum
Transparent Data Encryption (TDE)	✅	❌
Trusted extensions	✅	❌
SCRAM-SHA-256	✅	❌
Encrypted TCP/IP connection when GSSAPI	✅	❌
Row-level security policy	✅	❌

Extended Scenarios

Scenario 1: Unified Data Lake House



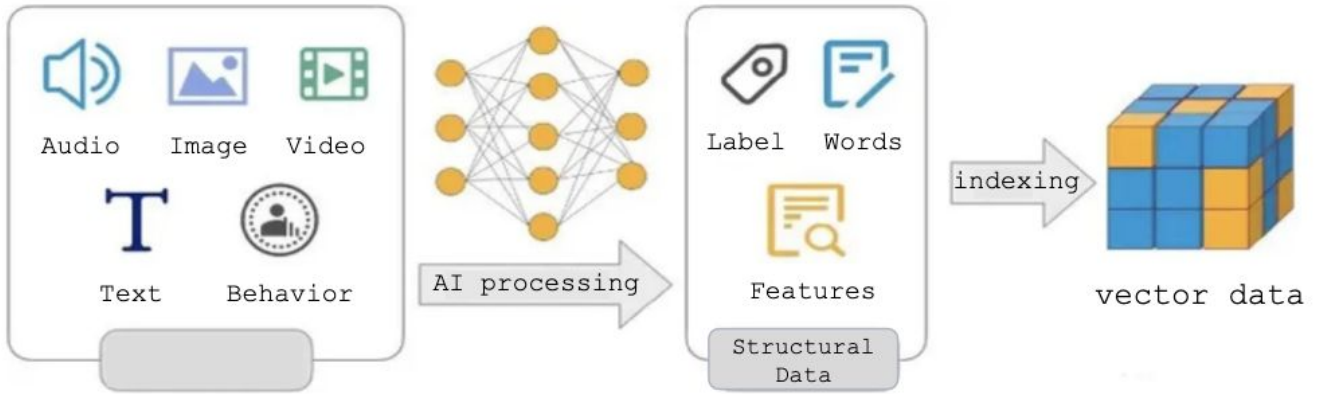
- Data Lake metadata is synced through the `Datalake connector`, which transforms data lake metadata to catalogs
- Leverages optimizer to produce best plan path to be executed.
- Executors receives the plan and follows the plan to fetch HDFS/OSS files, scan and parse.

Scenario 2: Vector Database w/ pgvector

APACHE Cloudberry™

Various distance measures: e.g. **Euclidean** distance, **cosine** distance, etc.

High-dimensional support: up to 16,000 dimensions



- **Search acceleration:** Provide efficient indexing options for vector data, such as k-nearest neighbor (k-NN) search. Even for massive datasets, users can achieve fast query execution and maintain high search accuracy
- **Based on standard SQL access**: empower SQL users using standard SQL query syntax for vector operations.
- **Robustness and security:** inherits the same level of robustness and security features, allowing users to securely store and manage vector data.

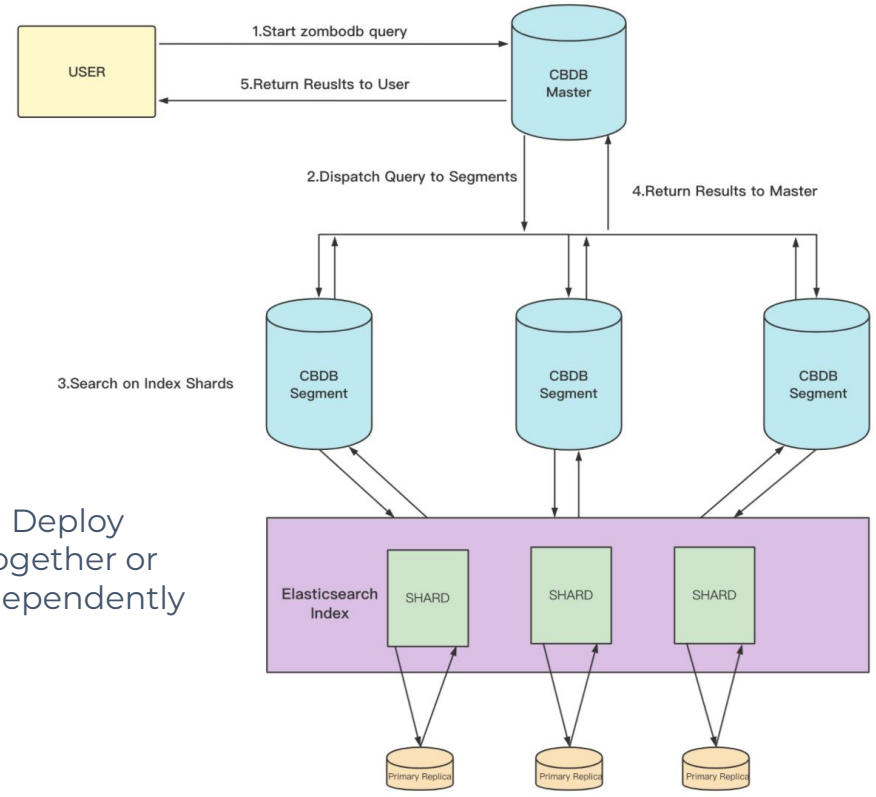
SYNX

Scenario 3: Fulltext Search Enhancement ZomboDB

- Apache Cloudberry integrates the ZomboDB plugin to enable Elasticsearch (ES) for rich full-text search and text analysis capabilities.

- Architecture:
  - ZomboDB is actually an external index of data tables. Users can create indexes (ES indexes) for ZomboDB on existing Apache Cloudberry data databases & tables by using the syntax of creating indexes
  - Support ZQL and ES JSON queries, support multi languages information segmentation and retrieval
- Use cases:
  - Knowledge Base, Intelligent Customer Service Robot
  - Transaction record analysis to optimize programmatic trading algorithms/strategies
  - Intelligent Announcement/Document Interpretation

```
INDEXING : CREATE INDEX idxproducts ON products USING
zombodb ((products.*)) WITH
(url='http://localhost:9200/');
SERACH : select * from products where products ==>
'sports,box';
```



Deploy together or independently

Future Works

Server Enhancement

- `Pluggable` MPP components, including dispatcher, interconnect, optimizer, transaction management, etc.
- Contribute `PAX` hybrid row-column storage
- Contribute `vectorization` execution engine
- Enable more `parallel operators`
- Make `ORCA` support equal features as `planner`
- Refactor the `dispatcher` logic for improved efficiency
- Materialized view and query rewrite for `external tables`
- Support `dynamic tables`
- Enable `query on standby`
- `Yeezzey storage manager` support

Tools Enhancement

Central Console

- Monitoring
  - Host processes summary
  - Compute resource monitoring
  - Workload monitoring
  - Segments recovery
- Administration:
  - user/role configuration
  - authentication
  - database object permission
  - DDL/DML/DCL auditing
- Alert configuration
  - rule based
  - `snmp` support

Central Console (Cont.)

- Tuning
  - Resource queue / group integration
  - Lock dependency graph
  - Table data skew monitor
  - System dirty pages statistics for heap
  - Self-tuning with recommendation
    - better data distribution
    - better indexing

Data Protection

- Multi-site disaster recovery support

Engineering Enhancement

Quality Assurance

- Automatic SQL generation, e.g. SQLancer
- **Chaos Monkey** testing
- Reduce ICW running time
- **In-place** database **upgrade** testing
- **Binary swap** tests for minor versions
- TPC-H/TPC-DS **benchmarking**

Release & Pipelines

- Refactor current `ICW` cases to reduce PR waiting time
- Automate building, testing, and deployment workflow w/ on GitHub Action and Docker.
- More OS distributions support, including Rocky Linux, Debian, and Ubuntu
- Pre-built Docker image support
- Support more CPU arch, including x86_64, ARM, RISC V, and LoongArch
- Ensure commit conventions through git pre-commit hooks
- Ansible playbook on cloud provider

Other Enhancement

Extensions, Ecosystem, and Tools

- DBeaver support
- PGRX support Postgres Extensions in Rust
- Enable kafka_fdw in MPP
- Integration Flink CDC to support near real-time data integration
- Lakehouse support, for example, Apache Hudi, Apache Iceberg and etc
- Patch-up tools, including pxf, gpbackup and etc
- JDBC/ODBC drivers
- Container service for executing UDFs in sandbox
- `K8S operator` for deployment

AI/ML

- pgvector continuous upgrading
- Integration with Ray to support AI/ML workloads

Welcome to join Cloudberry community

Follow us

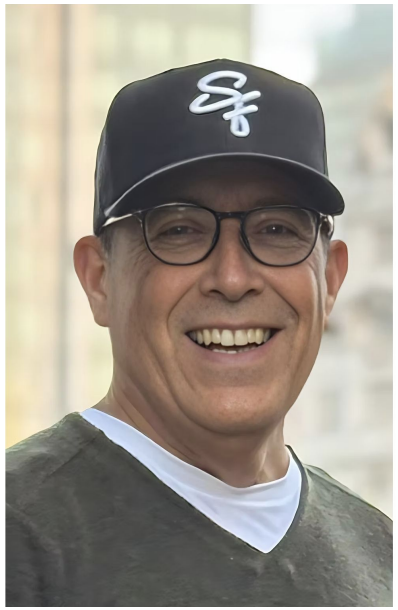
- Website: http://cloudberry.apache.org/
- X (Twitter): @ASFCloudberry
- Youtube: @ApacheCloudberry
- LinkedIn: @Apache Cloudberry
- Slack: https://apache-cloudberry.slack.com

Dev Mailing list:

- dev@cloudberry.apache.org

Sandbox: quickly try out Cloudberry

- https://github.com/apache/cloudberry-bootcamp



Ed Espino

Co-founder, Synx Data Labs

Ed Espino has spent decades building and scaling database platforms, always pushing the boundaries of MPP analytics. Before co-founding Synx Data Labs, he led engineering for Greenplum Database at VMware and Pivotal Software, shaping the future of enterprise data solutions. He's also a PPMC member of Apache Cloudberry (Incubating), helping drive innovation in real-time big data analytics. At Synx Data Labs, Ed is focused on delivering powerful, high-performance database solutions for modern businesses.

A passionate open-source explorer and lifelong learner, Ed loves solving complex problems, contributing to the community, and embracing new challenges. When he's not working on databases, you'll find him hiking the outdoors, giving back to open-source projects, or traveling the world—having walked the Camino de Santiago and explored Asia and Europe, where he has always felt at home and welcomed.

LinkedIn || eespino@synxdata.com

Shine Zhang

Co-founder, Synx Data Labs

"Do the right thing. Do what works. Be kind."

Xin "Shine" Zhang is a technology leader with deep expertise in distributed databases, AI-driven analytics, and cloud platforms. Before co-founding Synx Data Labs, he was an Engineering Tech Lead at Broadcom VMware Tanzu Data, leading enterprise data migration and next-gen database platform development.

With over 20 years of experience in MPP databases like Greenplum and PostgreSQL, Shine has been a core contributor to open-source projects, including Apache Cloudberry (Incubating), Greenplum, and Apache Geode. At Synx Data Labs, he is focused on advancing distributed data systems and AI-powered analytics solutions.

A lifelong learner and open-source contributor, Shine enjoys experimenting with AI models, mentoring engineers, and exploring new places when he's not building data platforms.

LinkedIn || shine@synxdata.com

APACHE Cloudberry™

Tushar Pednekar

Co-founder, Synx Data Labs

Tushar Pednekar is a customer success and data solutions leader with a deep background in MPP databases and enterprise analytics. Before co-founding Synx Data Labs, he led global data solutions teams at VMware, Pivotal Software, and Broadcom, driving customer success and revenue growth in large-scale database technologies.

A founding member of Apache Cloudberry (Incubating), Tushar actively contributes to the open-source data ecosystem. His expertise spans technical strategy, customer advocacy, and go-to-market execution. At Synx Data Labs, he focuses on helping businesses transition to next-generation data platforms with confidence.

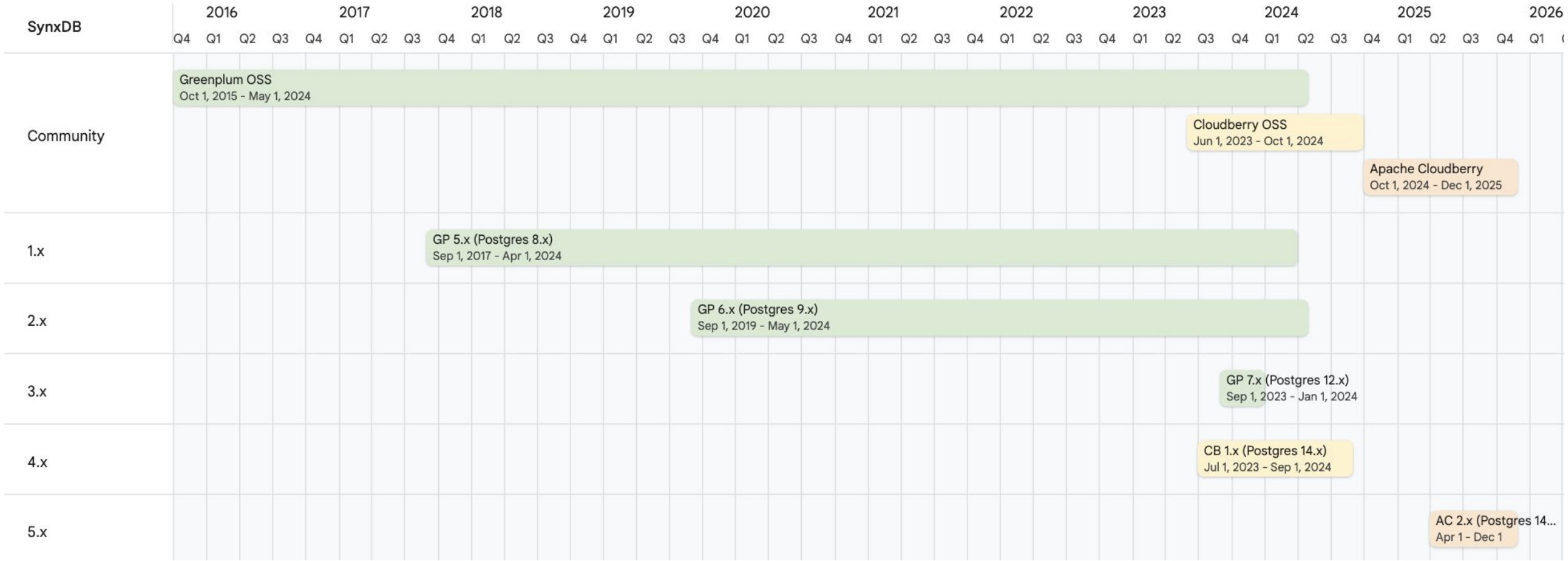
Beyond work, Tushar enjoys mentoring engineers, solving business challenges with data, and exploring new ideas in cloud and AI.

LinkedIn || tushar@synxdata.com

SYNX™

SYNX

SynxDB Past, Present and Prospect



SYNX

VLDB

ENFUSE.IO

APACHE Cloudberry™

SERVISCO

RDS TECH

HashData

system 新立資訊

webcomm 偉康科技