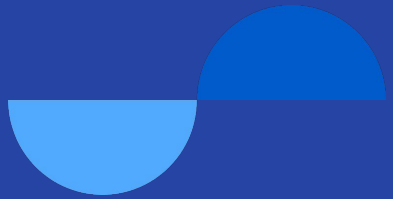
Databend

Databend

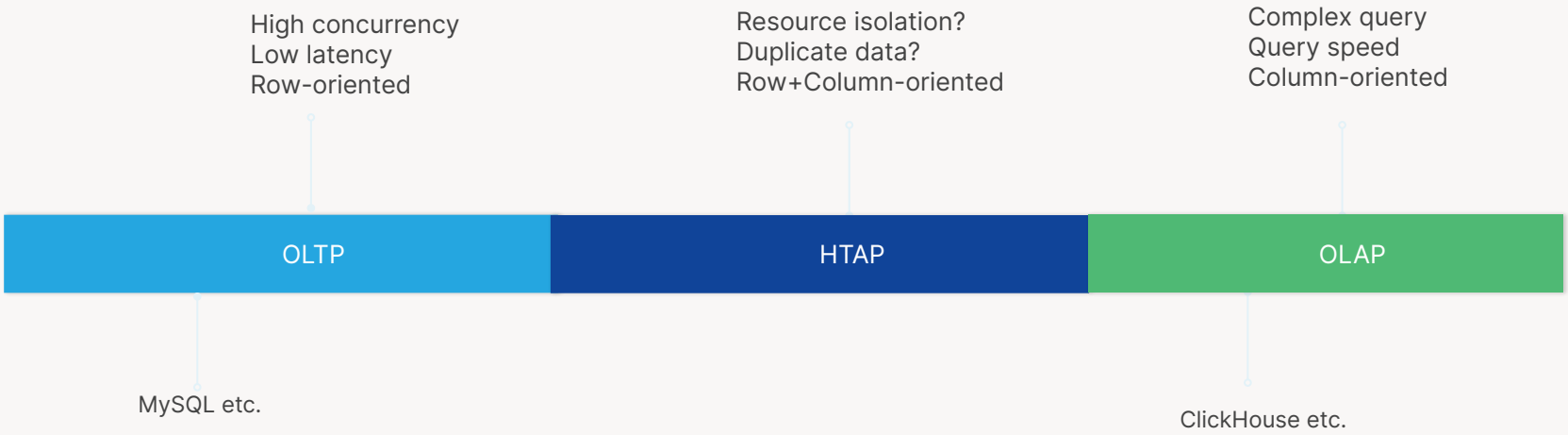
A modern warehouse with Rust for your massive-scale analytics

https://github.com/datafuselabs/databend



Database and Data Warehouse

Databend



大纲

Databend

- 大数据分析遇到了什么”新”问题？
- 传统数仓为什么无法解决这些“新”问题？
- 新一代实时弹性数仓如何设计？
- 使用 Rust 从 0 到 1 研发一款数仓是种什么体验？

Databend

Bohu TANG (张雁飞)

Co-Creator of Databend: https://github.com/datafuselabs/databend
ClickHouse and MySQL(TokuDB) 重度贡献者

Database Kernel | Distributed Database | Data Warehouse

https://bohutang.me/

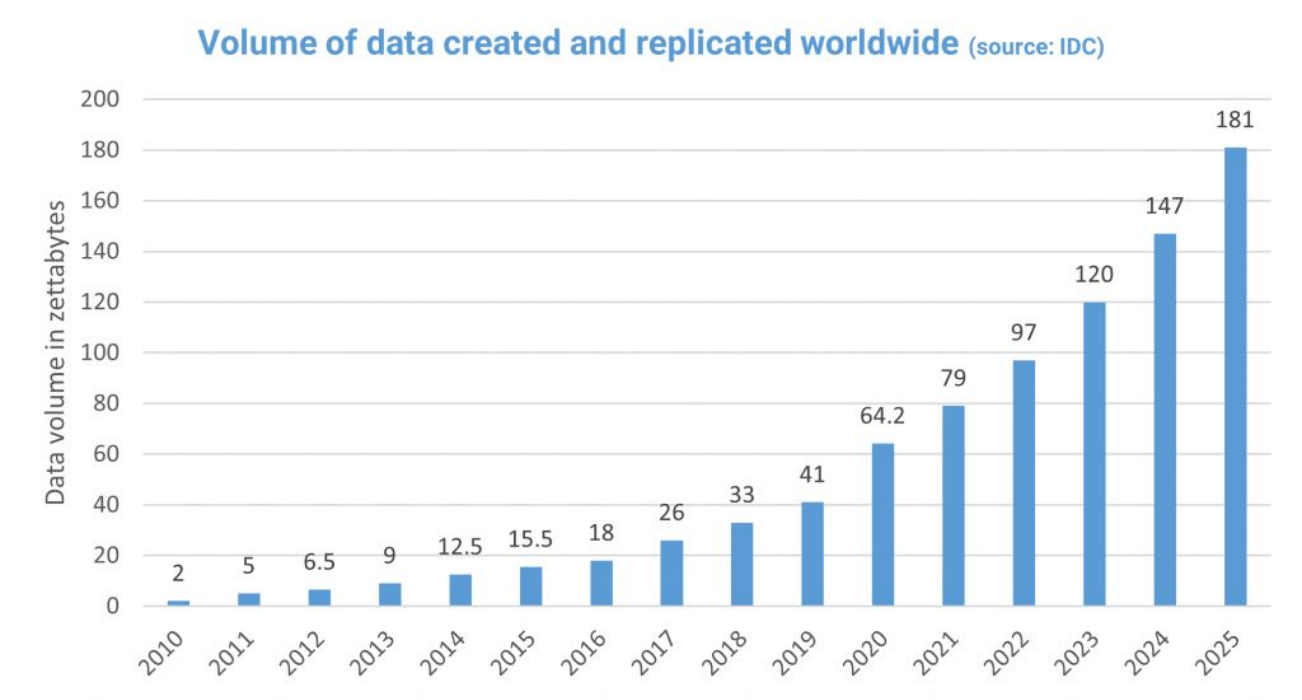
Databend

01

当今(2022)大数据新问题

全球数据指数级增长

Databend



1024PB = 1EB, 1024EB = 1ZB

当今大数据新问题 1

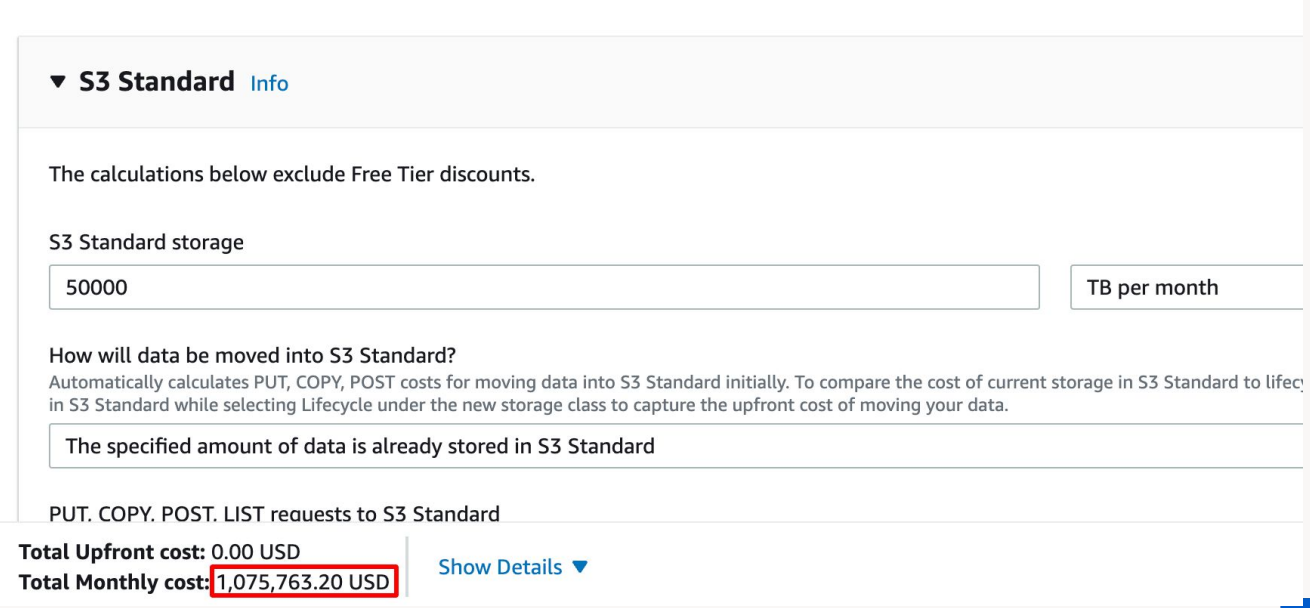
- 大数据量下的资源利用率问题, < 50%
- 物理资源常驻问题
- 大数据分析，波峰、波谷问题

Databend

当今大数据新问题 2

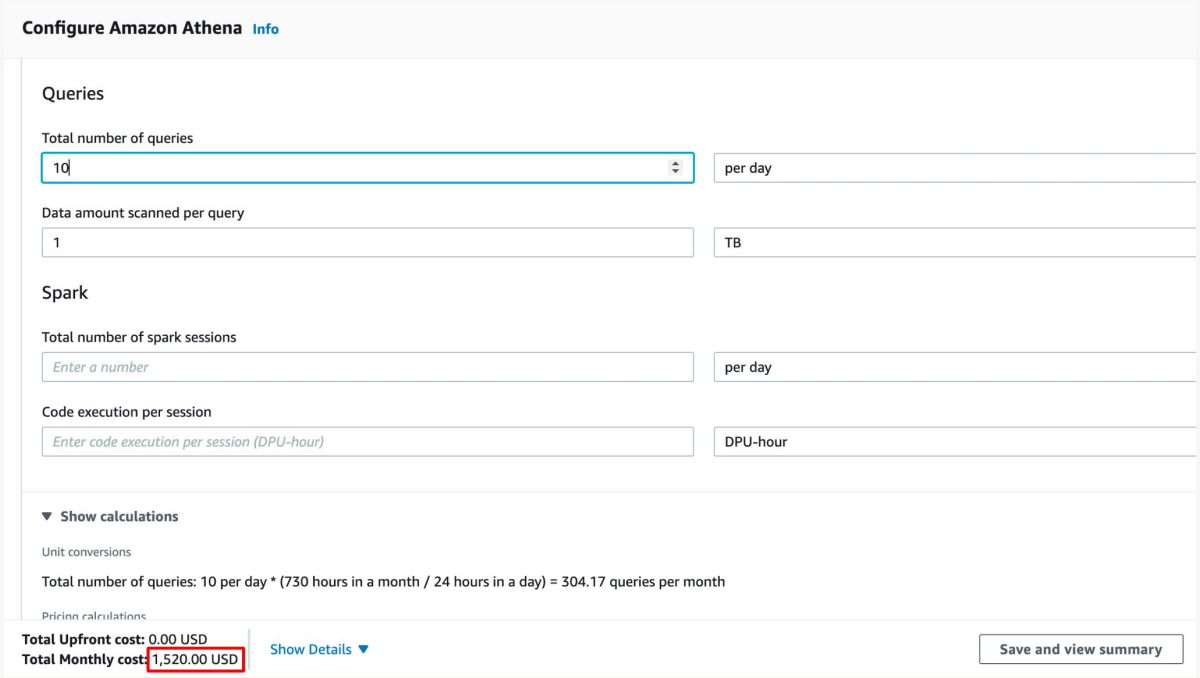
Databend

- 大数据量下的存储成本问题
- PB 级数据，每月存储成本百万美金！



当今大数据新问题 3

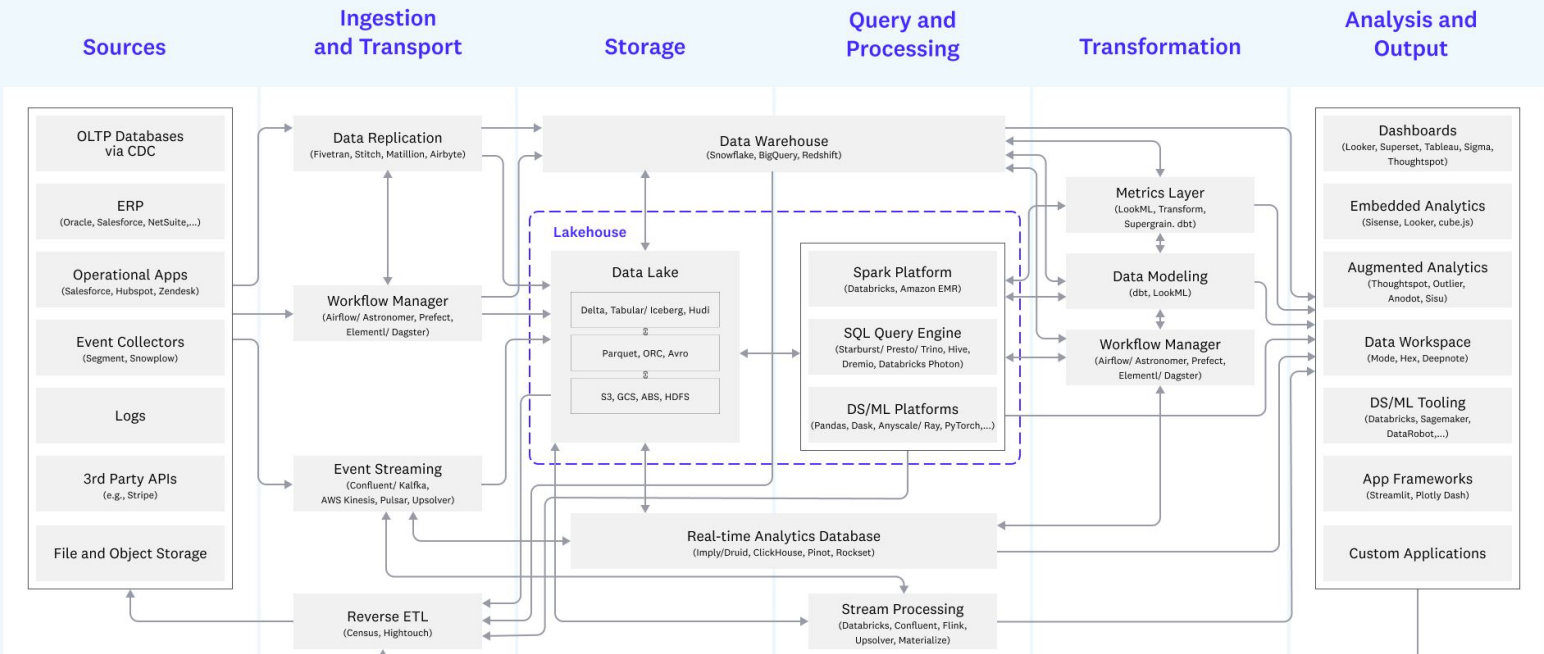
- 大数据量下的计算成本问题
- 对扫描数据量要求非常高，容易破产



当今大数据新问题 4

- 大数据量下的数据平台复杂度越来越高

Unified Data Infrastructure (2.0) (From a16z)



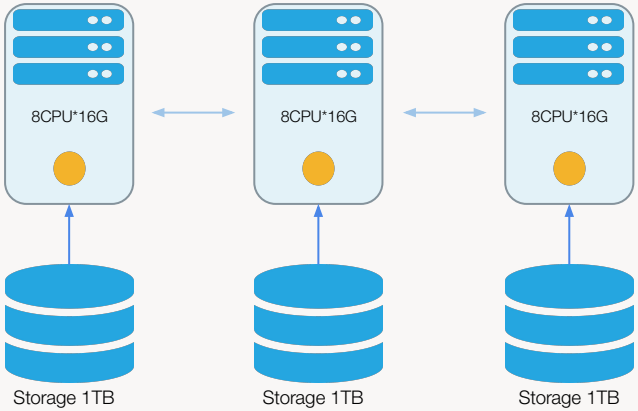
Databend

02

传统数仓架构 vs. 弹性数仓架构

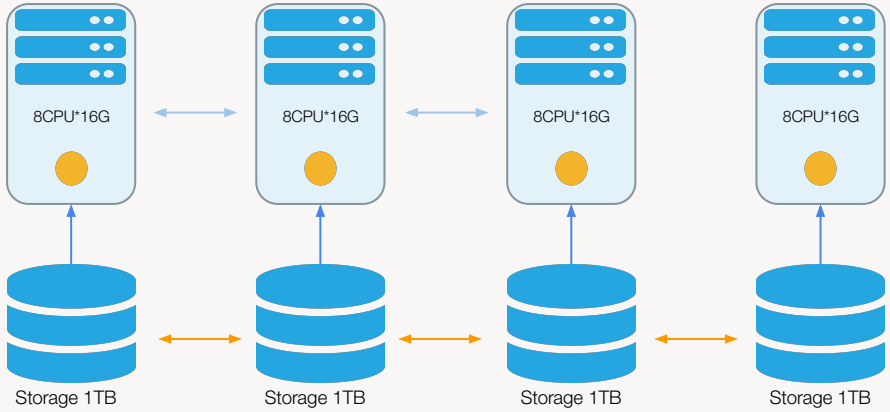
传统数仓架构

- Shared-Nothing
- 存储、计算一体
- 资源控制粒度粗



传统数仓架构

- Shared-Nothing
- 存储、计算一体
- 资源控制粒度粗



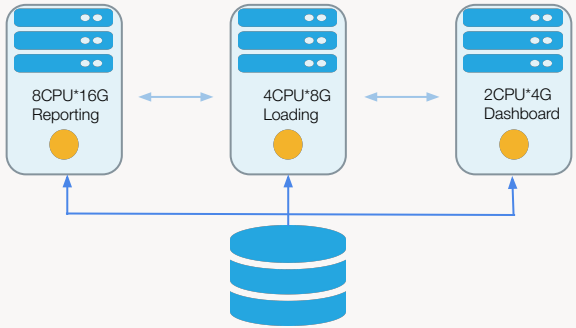
传统数仓架构

- Shared-Nothing - 弱弹性
- 存储、计算一体 - 弱弹性
- 资源控制粒度粗 - 成本高

成本(高) = Resource * Time

新一代弹性数仓架构

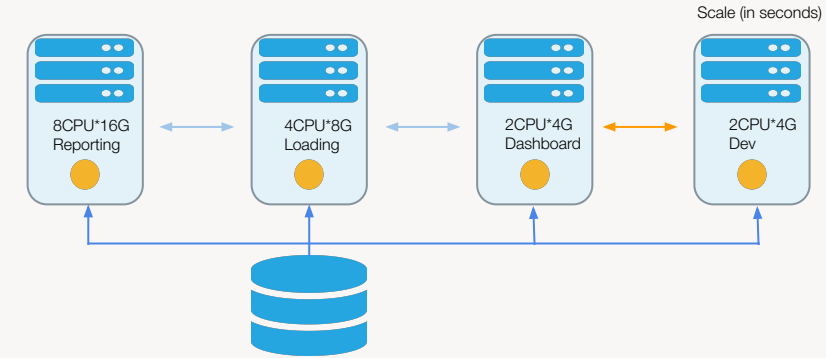
- Shared-Storage (Amazon S3, Azure Blob …)
- 真正存储、计算分离
- 实时弹性扩容和缩容
- 资源控制粒度细



AWS S3, Azure Blob, Google GCS, 阿里云 OSS, 腾讯云 COS, 华为云 OBS, IPFS ...

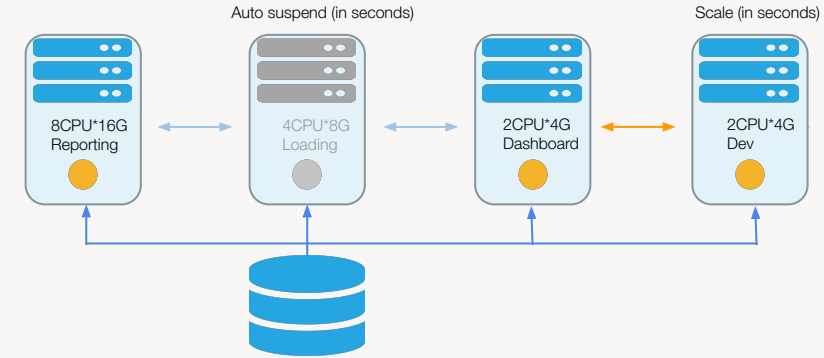
新一代弹性数仓架构

- Shared-Storage (Amazon S3, Azure Blob …)
- 真正存储、计算分离
- 实时弹性扩容和缩容
- 资源控制粒度细



新一代弹性数仓架构

- Shared-Storage (Amazon S3, Azure Blob …)
- 真正存储、计算分离
- 实时弹性扩容和缩容
- 资源控制粒度细



新一代弹性数仓架构

Databend

- Shared-Storage (Amazon S3, Azure Blob …) - 高弹性
- 真正存储、计算分离 - 高弹性
- 实时弹性扩容和缩容 - 高弹性
- 资源控制粒度细 - 成本低

成本(低) = Resource * Time

Databend

03

Databend 新一代实时弹性架构设计

ClickHouse

Databend

- OS Warehouse
- 向量化计算，细节优化到位
- Pipeline 处理器和调度器
- MergeTree 列式存储引擎
- 单机性能非常强悍
- 缺点：分布式能力较弱，运维复杂度高

[ClickHouse Group By 为什么这么快]：https://bohutang.me/2021/01/21/clickhouse-and-friends-groupby/

[ClickHouse Pipeline 处理器和调度器]：https://bohutang.me/2020/06/11/clickhouse-and-friends-processor/

[ClickHouse 存储引擎技术进化与MergeTree]: https://bohutang.me/2020/06/20/clickhouse-and-friends-merge-tree-algo/

Snowflake

- Cloud Warehouse
- 多租户，存储、计算分离
- 基于对象存储便宜介质
- 弹性能力非常强悍
- 缺点：单机性能一般，重度依赖分布式

Databend = ClickHouse + Snowflake + Rust

Databend

- 借鉴 ClickHouse 向量化计算，提升单机计算性能
- 借鉴 Snowflake 存储、计算分离思想，提升分布式计算能力
- 借鉴 Git，MVCC 列式存储引擎，Insert/Read/Delete/Update(WIP)
- 高弹性 + 强分布式，致力于解决大数据分析**成本**和**复杂度**问题
- 基于便宜的对象存储也能方便的做实时性分析
- 完全使用 Rust 研发（30w+ loc），Day1 在 Github 开源

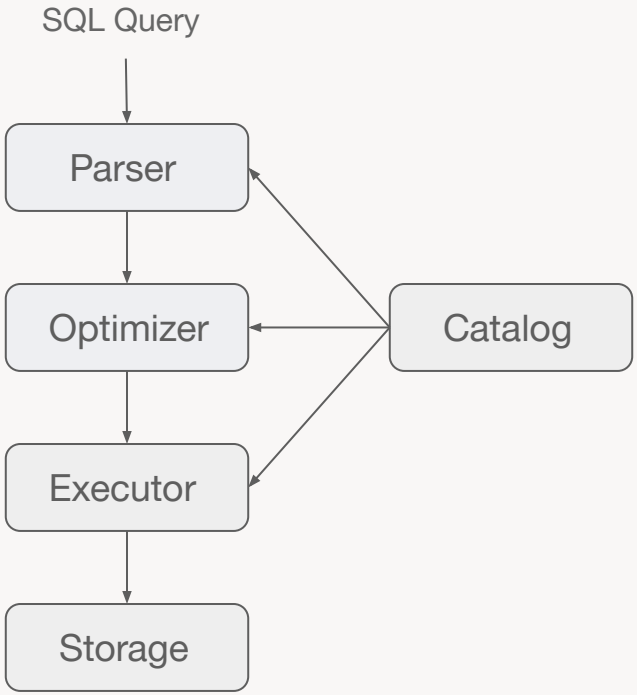
挑战

Databend

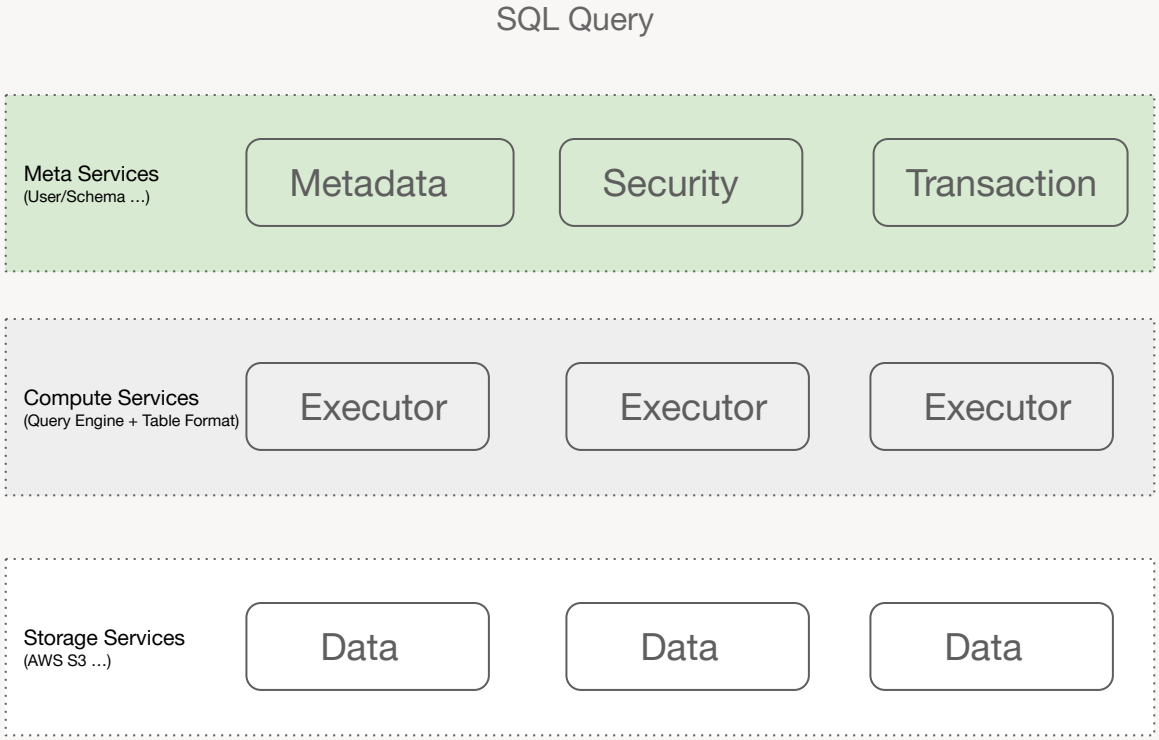
- 计算层做到高弹性，计算的状态如何管理？
- 对象存储不是为数据库而设计，高延迟和高性能如何平衡？
- 如何让系统更加智能，根据查询模式自动创建索引？
- 如何面向 Warehouse + Datalake 需求设计？

单机功能模块

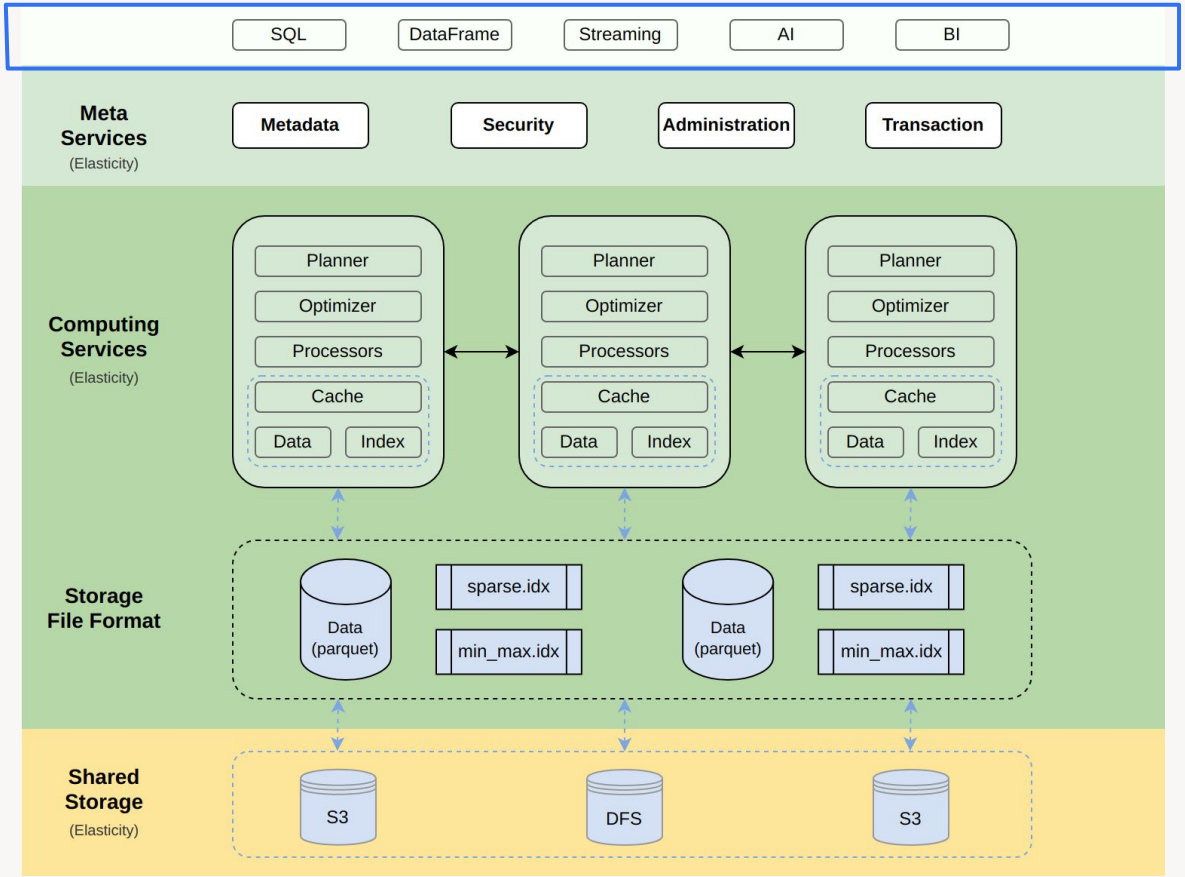
Databend



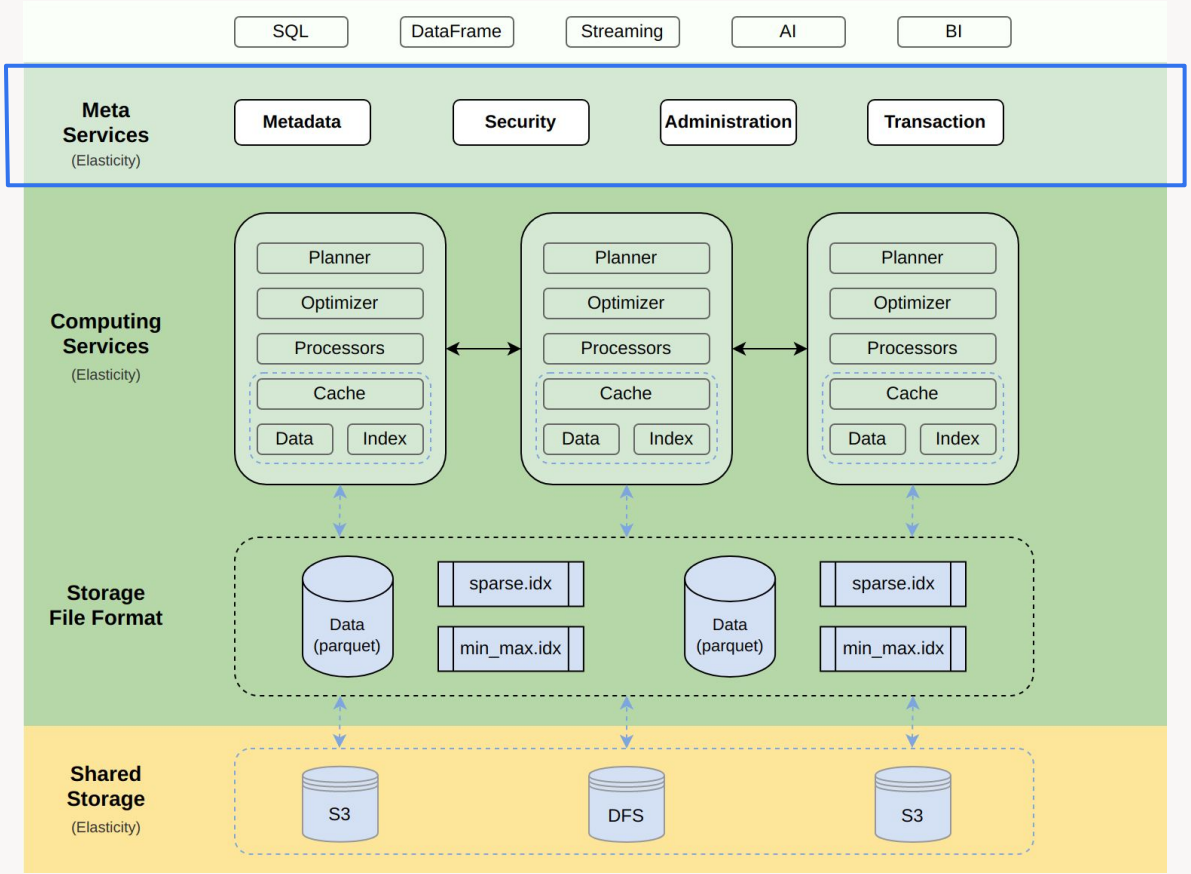
模块微服务化


SQL Query
Meta Services
(User/Schema ...)
Metadata
Security
Transaction
Compute Services
(Query Engine + Table Format)
Executor
Executor
Executor
Storage Services
(AWS S3 ...)
Data
Data
Data

Databend 架构

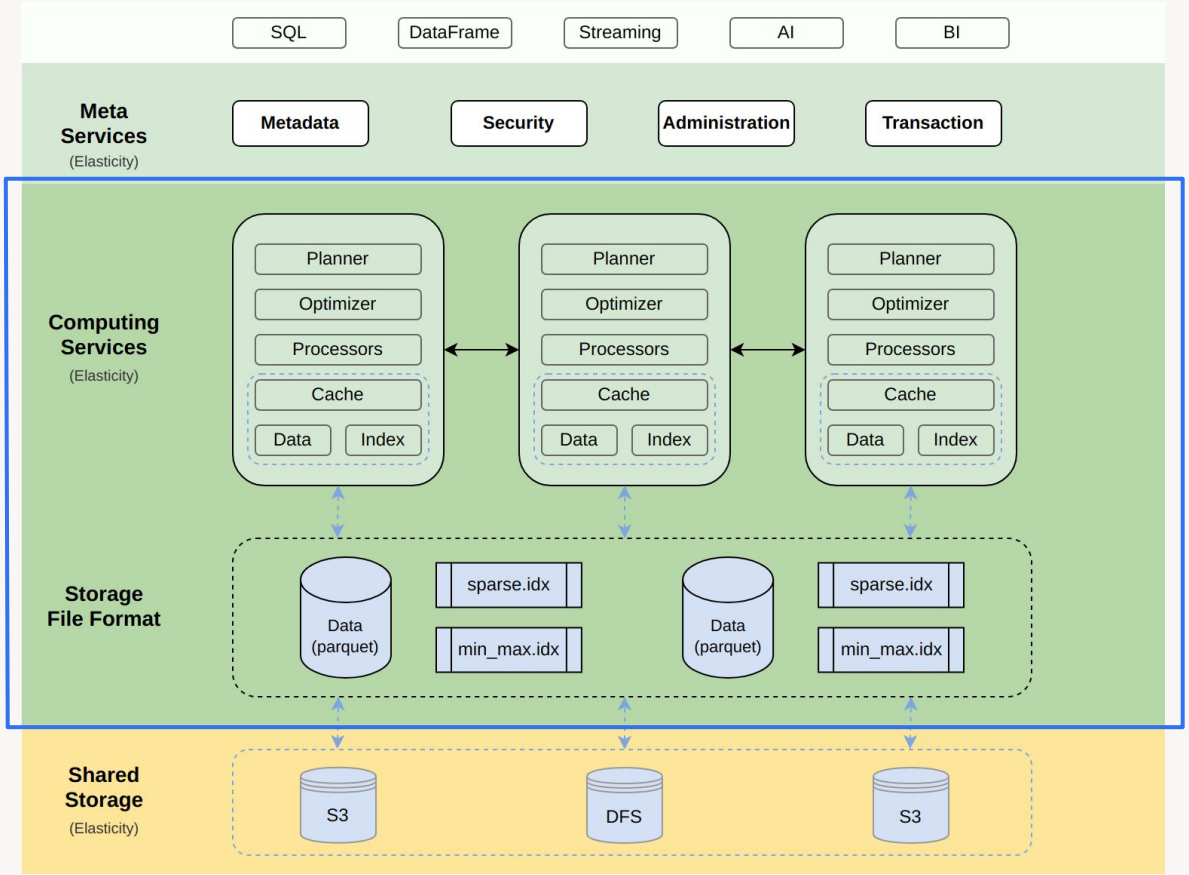

SQL
DataFrame
Streaming
AI
BI
Meta Services
(Elasticity)
Metadata
Security
Administration
Transaction
Computing Services
(Elasticity)
Planner
Optimizer
Processors
Cache
Data
Index
Planner
Optimizer
Processors
Cache
Data
Index
Planner
Optimizer
Processors
Cache
Data
Index
Storage File Format
Data (parquet)
sparse.idx
min_max.idx
Data (parquet)
sparse.idx
min_max.idx
Shared Storage
(Elasticity)
S3
DFS
S3

Databend 架构


SQL
DataFrame
Streaming
AI
BI
Meta Services
(Elasticity)
Metadata
Security
Administration
Transaction
Computing Services
(Elasticity)
Planner
Optimizer
Processors
Cache
Data
Index
Planner
Optimizer
Processors
Cache
Data
Index
Planner
Optimizer
Processors
Cache
Data
Index
Storage File Format
Data (parquet)
sparse.idx
min_max.idx
Data (parquet)
sparse.idx
min_max.idx
Shared Storage
(Elasticity)
S3
DFS
S3

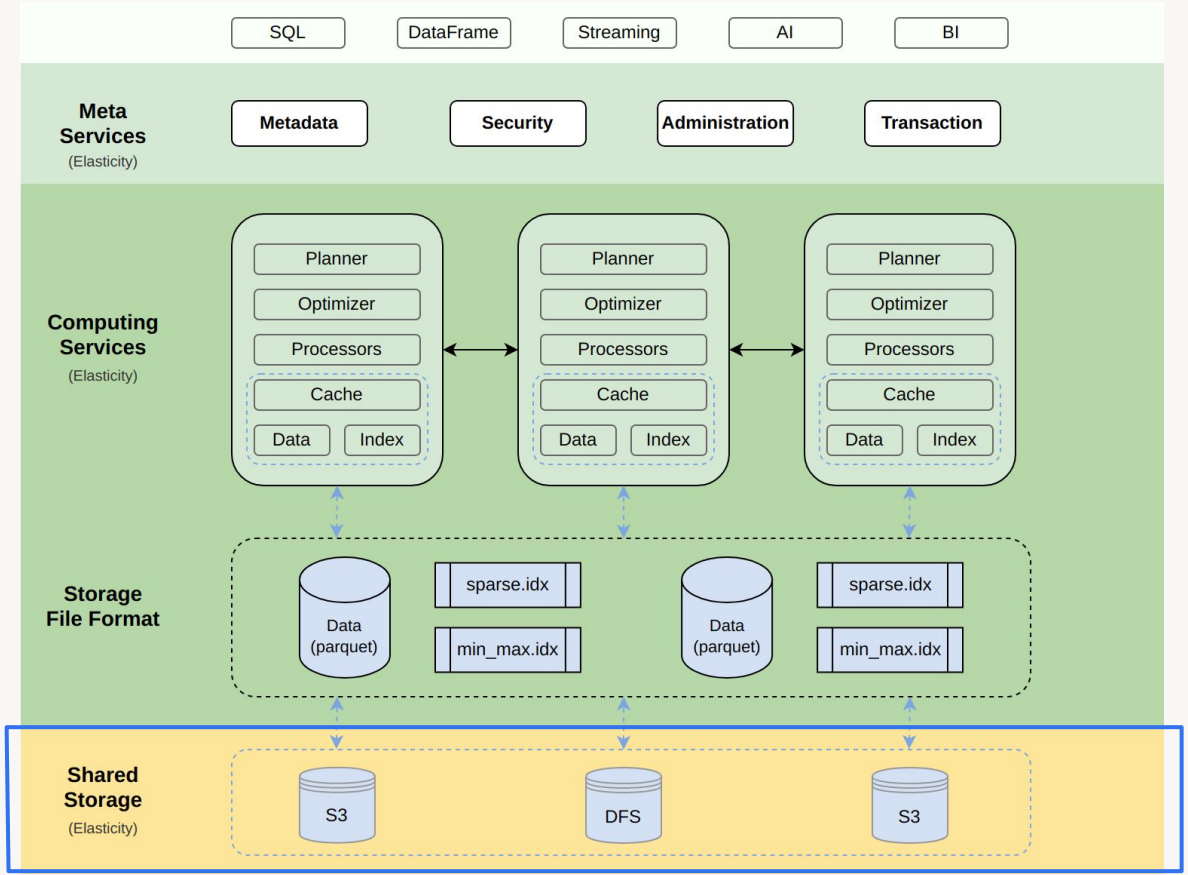
Databend 架构

Databend



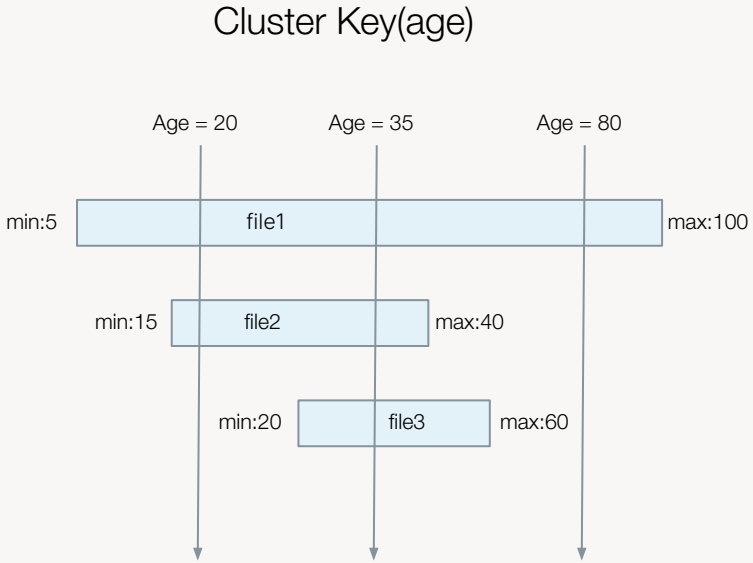
Databend 架构

Databend

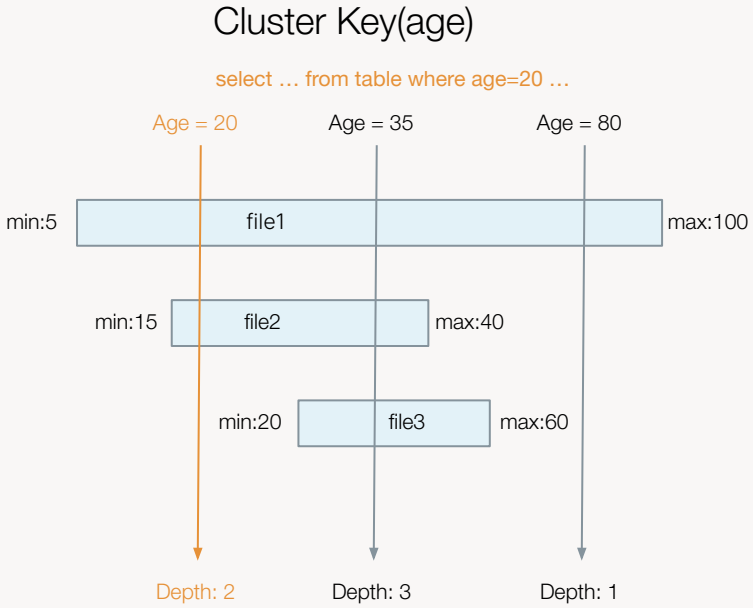


Automatic Tuning

Databend

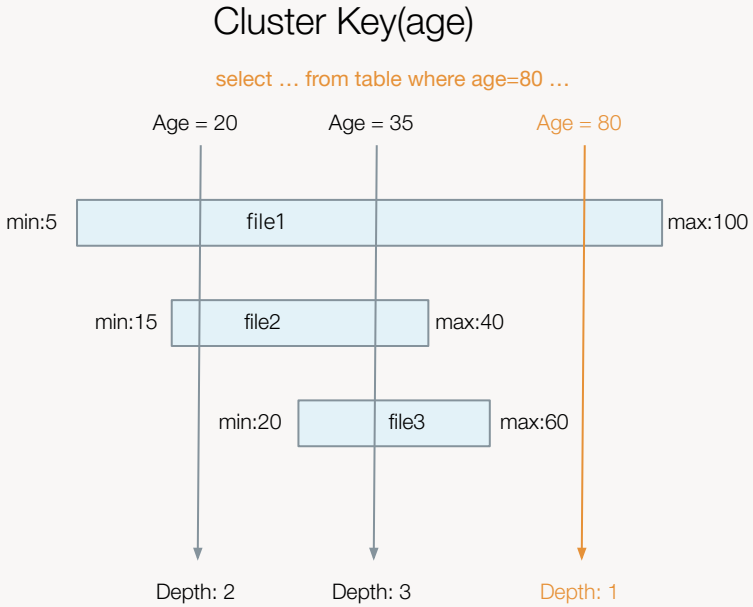


Automatic Tuning



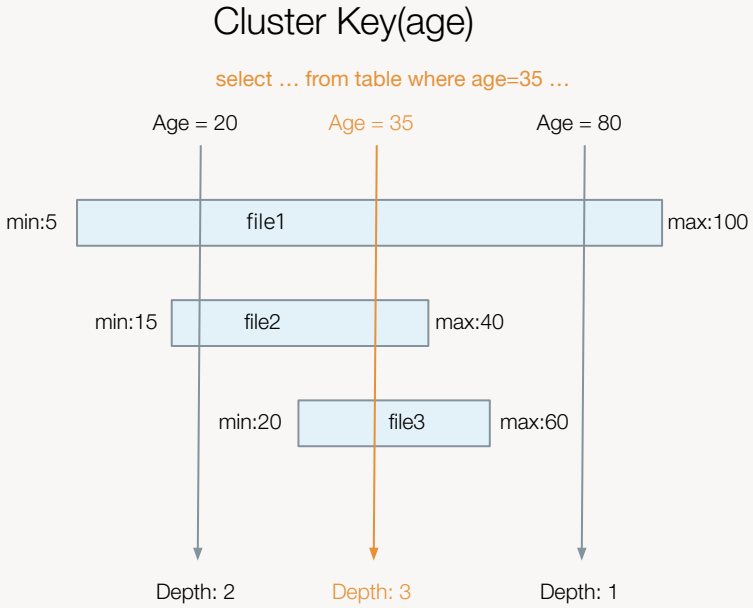
Automatic Tuning

Databend



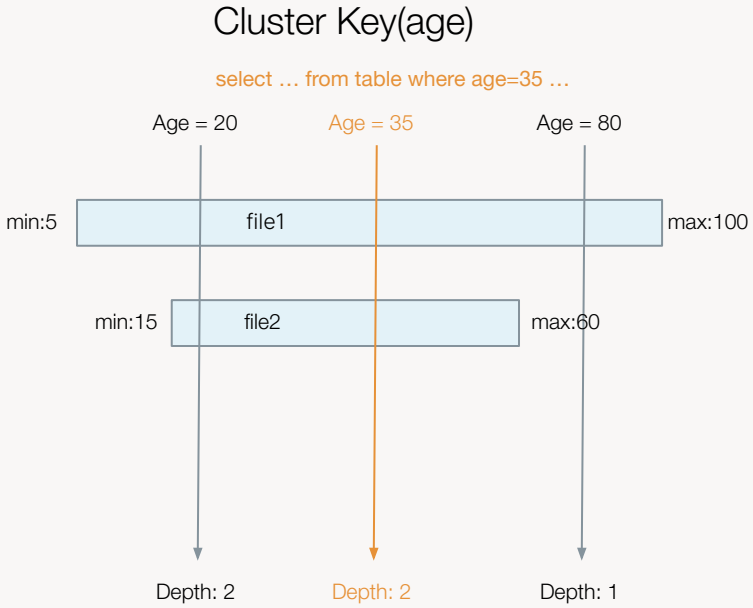
Automatic Tuning

Databend



Automatic Tuning

Databend



Databend + Hive

Databend

```
CREATE CATALOG my_hive
  TYPE=HIVE
  CONNECTION = (URL='<hive-meta-store>'
THRIFT_PROTOCOL=BINARY);
SELECT * FROM my_hive.db1.table;
```

[Multiple Catalog RFC]: https://databend.rs/doc/contributing/rfcs/multiple-catalog

Databend + Iceberg

Databend

```
CREATE CATALOG my_iceberg
  TYPE=ICEBERG
  CONNECTION = (URL='s3://my_bucket/path/to/iceberg');
SELECT * FROM my_iceberg.db1.table;
```

[Multiple Catalog RFC]: https://databend.rs/doc/contributing/rfcs/multiple-catalog

Databend

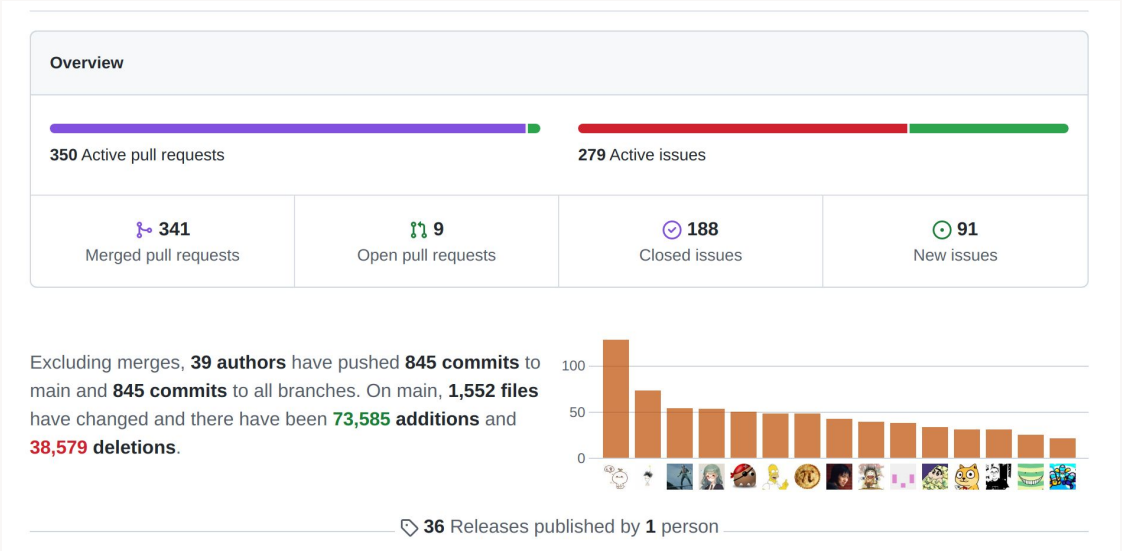
04

Databend 开源社区

Databend 开源社区

Databend

4.9K Stars	140+ Contributors	迭代非常快



https://github.com/datafuselabs/databend

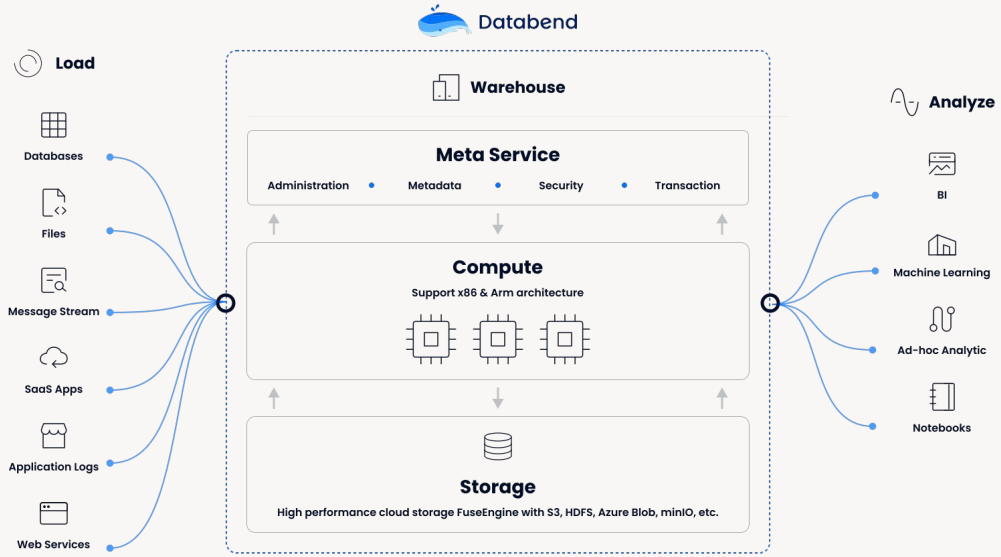
Databend 开源社区

~40 月度活跃开发者：

SAP
Yahoo
Fortinet
Shopee
PingCAP
Alibaba
Tencent
ByteDance
EMQ
快手 (湖仓一体共建)
… …

Databend 体验：On-Premises, Serverless

- On-Premises
  社区版：https://databend.rs
- Serverless Cloud
  海外(AWS)　https://app.databend.com
  国内(阿里云) https://app.databend.cn



Databend 用户

Databend

More …

Databend

Thanks