# 背景

在互联网浪潮出现之前，企业的数据量普遍不大，特别是核心的业务数据，通常一个单机的数据库就可以保存。那时候的存储并不需要复杂的架构，所有的线上请求（OLTP，Online Transactional Processing）)和后台分析（OLAP，Online Analytical Processing） 都跑在同一个数据库实例上。

随着互联网的发展，企业的业务数据量不断增多，单机数据库的容量限制制约了其在海量数据场景下的使用。因此在实际应用中，为了面对各种需求，OLTP、OLAP 在技术上分道扬镳，在很多企业架构中，这两类任务处理由不同团队完成。当物联网大数据应用不断深入，具有海量的传感器数据要求实时更新和查询，对数据库的性能要求也越来越高，此时，新的问题随之出现：

1、OLAP和OLTP系统间通常会有几分钟甚至几小时的时延，OLAP数据库和OLTP数据库之间的一致性无法保证，难以满足对分析的实时性要求很高的业务场景。

2、企业需要维护不同的数据库以便支持两类不同的任务，管理和维护成本高。

因此，能够统一支持事务处理和工作负载分析的数据库成为众多企业的需求。在此背景下，由Gartner提出的HTAP（混合事务/分析处理，Hybrid Transactional/Analytical Processing）成为希望。基于创新的计算存储框架，HTAP数据库能够在一份数据上同时支撑业务系统运行和OLAP场景，避免在传统架构中，在线与离线数据库之间大量的数据交互。此外，HTAP基于分布式架构，支持弹性扩容，可按需扩展吞吐或存储，轻松应对高并发、海量数据场景。

# 概述

## 定义

HTAP数据库（Hybrid Transaction and Analytical Process，混合事务和分析处理）。2014年Gartner的一份报告中使用混合事务分析处理（HTAP）一词描述新型的应用程序框架，以打破OLTP和OLAP之间的隔阂，既可以应用于事务型数据库场景，亦可以应用于分析型数据库场景。实现实时业务决策。这种架构具有显而易见的优势：**不但避免了繁琐且昂贵的ETL操作，而且可以更快地对最新数据进行分析**。这种快速分析数据的能力将成为未来企业的核心竞争力之一。

## 特点

此类通用平台方案具备下面优点：

通过数据整合避免信息孤岛，便于共享和统一数据管理。

基于SQL的数据集成平台可提供良好的数据独立性，使应用能专注于业务逻辑，不用关心数据的底层操作细节。

集成数据平台能提供更好的实时性和更全的数据，为业务提供更快更准的分析和决策。

能够避免各种系统之间的胶合，企业总体技术架构简单，不需要复杂的数据导入/导出等，易于管理和维护。

便于人才培养和知识共享，无须为各种专有系统培养开发、运维和管理人才。

# 技术要点

底层数据要么只有一份，要么可快速复制，并且同时满足高并发的实时更新。

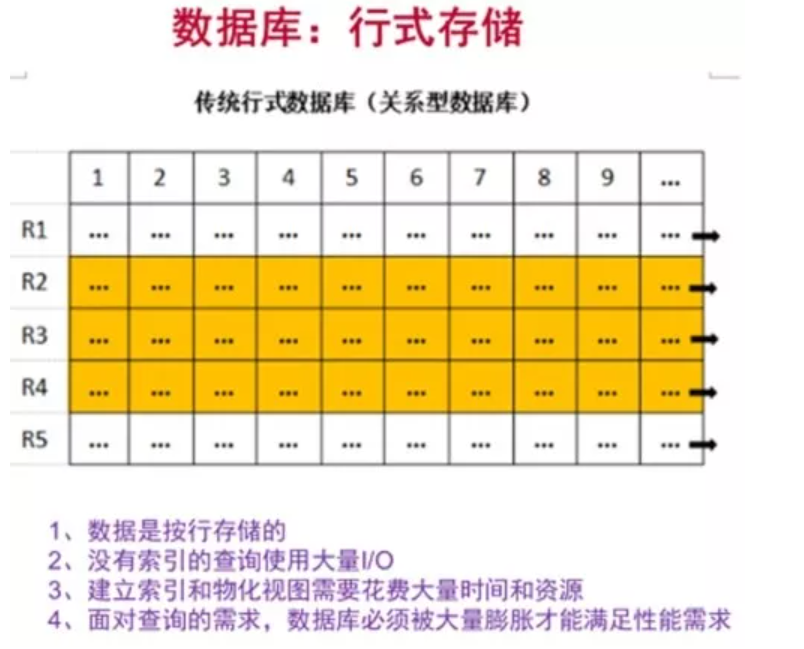
要满足海量数据的容量问题，在存储、计算都具有很好的线性扩展能力。

具有很好的优化器，可满足事务类、分析类的语句需求。

具备标准的SQL，并支持诸如二级索引、分区、列式存储、向量化计算等技术。

## 行列存储

行存储（Row-based）：对于传统的关系型数据库，比如甲骨文的OracleDB和MySQL，IBM的DB2、微软的SQL Server等，一般都是采用行存储（Row-based）行。在基于行式存储的数据库中，数据是按照行数据为基础逻辑存储单元进行存储的，一行中的数据在存储介质中以连续存储形式存在。



列式存储（Column-based）是相对于行式存储来说的，新兴的Hbase、HP Vertica、EMC Greenplum 等分布式数据库均采用列式存储。在基于列式存储的数据库中，数据是按照列为基础逻辑存储单元进行存储的，一列中的数据在存储介质中以连续存储形式存在。

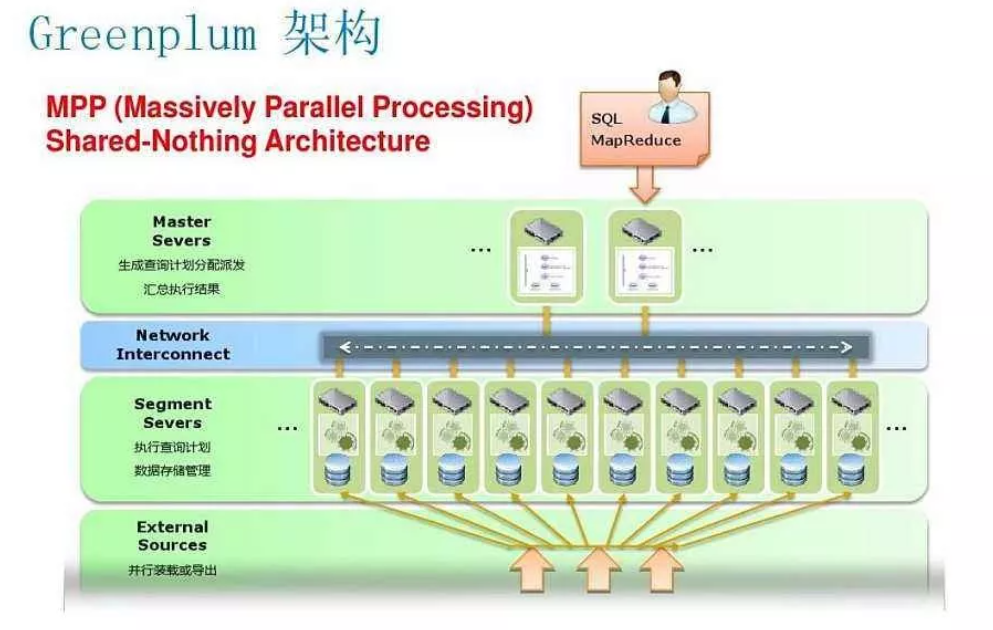


传统的行式数据库，是按照行存储的，维护大量的索引和物化视图无论是在时间（处理）还是空间（存储）面成本都很高。而列式数据库恰恰相反，列式数据库的数据是按照列存储，每一列单独存放，数据即是索引。只访问查询涉及的列，大大降低了系统I/O，每一列由一个线来处理，而且由于数据类型一致，数据特征相似，极大方便压缩。

## MPP

MPP (Massively Parallel Processing)，即大规模并行处理，在数据库非共享集群中，每个节点都有独立的磁盘存储系统和内存系统，业务数据根据数据库模型和应用特点划分到各个节点上，每台数据节点通过专用网络或者商业通用网络互相连接，彼此协同计算，作为整体提供数据库服务。非共享数据库集群有完全的可伸缩性、高可用、高性能、优秀的性价比、资源共享等优势。

简单来说，MPP是将任务并行的分散到多个服务器和节点上，在每个节点上计算完成后，将各自部分的结果汇总在一起得到最终的结果。下面以典型的MPP产品Greenplum架构为例。



## 资源隔离

OLTP、OLAP类两者对资源的使用特点不同，需要在资源层面做好隔离工作，避免相互影响。常见的通过定义资源队列的方式，指定用户分配队列，起到资源隔离的作用。

# 计算模型

## 火山模型

## 向量化引擎

# 实现方案

## TP和AP在存储/计算上都分离

TP数据导入到AP系统，有延迟，时效性不高。

增加冗余AP存储，成本增加。

增加AP系统，运维难度增加。

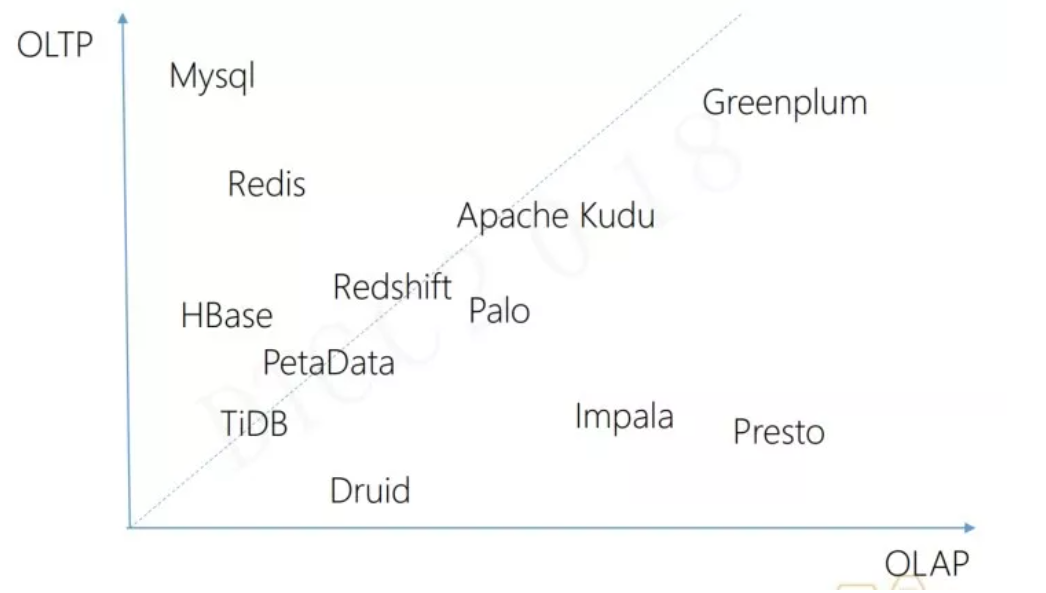
## TP和AP在存储/计算上都共享

在执行时AP、TP查询或多或少相互影响。

受限TP系统，AP比重增大时，无法快速弹性Scale Out。

## TP和AP在存储上共享，在计算上分离

# HTAP产品



目前，实现HTAP的数据库不多，主要有PingCAP的TiDB、阿里云的HybridDB for MySQL、百度的BaikalDB等。

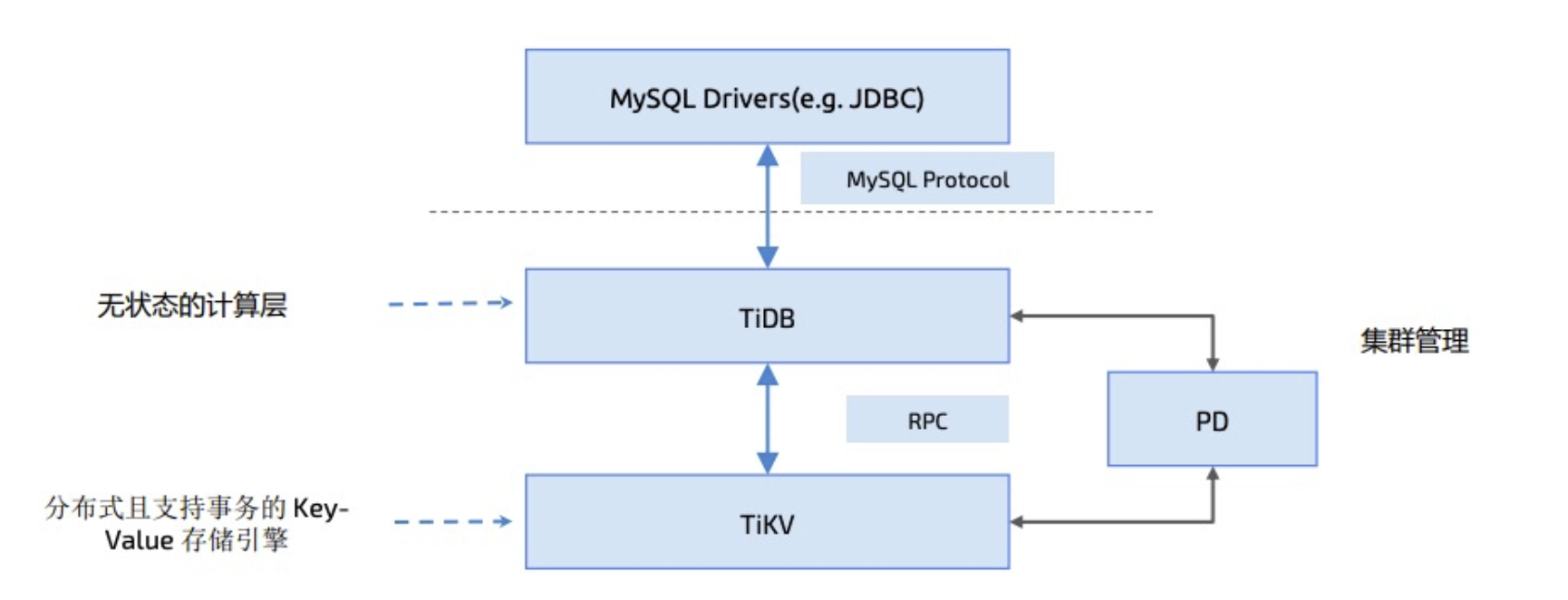
## OceanBase

## PolarDB

## TiDB

TiDB的理论基础来源于2013年Google发布的Spanner/F1论文 ，以及2014年Stanford工业级分布式一致性协议算法Raft论文。在架构上，TiDB将计算和存储层进行高度的抽象和分离，对混合负载的场景通过IO优先级队列、智能副本调度、行列混合存储等技术，使HTAP变为可能。

参考Google Spanner / F1的设计，TiDB整体架构分为上下两层：负责计算的TiDB Server和负责存储的TiKV Server，二者由集群管理模块PD Server调度和管理。



TiDB Server对应于Google F1，是一层无状态的SQL Layer，其本身并不存储数据，只负责计算。在接收到SQL请求后，该计算层会通过PD Server找到存储计算所需数据的TiKV地址，然后与TiKV Server交互获取数据，最终返回结果。在水平扩展方面，随着业务的增长，用户可以简单地添加TiDB Server节点，提高数据库整体的处理能力和吞吐。

作为整个集群的管理模块，PD（Placement Driver）主要工作有三类：一是存储集群的元信息；二是对TiKV集群进行调度和负载均衡，如数据的迁移、Raft group leader的迁移等；三是分配全局唯一且递增的事务ID。

TiKV Server是一个分布式Key Value数据库，对应于Google Spanner，支持弹性水平扩展。不同于HBase或者 BigTable那样依赖底层的分布式文件系统，TiKV Server在性能和灵活性上更好，这对于在线业务来说非常重要。随着数据量的增长，用户可以部署更多的TiKV Server节点解决数据Scale的问题。PD模块则会在TiKV节点之间以Region为单位做调度，将部分数据迁移到新加的节点上。

总体来看，TiDB具备如下核心特性：

水平线性扩展

强一致分布式事务

故障自恢复（auto -failover）的金融级高可用（非主从）

真正跨数据中心多活

据了解，TiDB对业务没有任何侵入性，能够替换传统的数据库中间件、数据库分库分表等Sharding方案。同时它也让开发运维人员不用关注数据库Scale的细节问题，专注于业务开发，大幅度提升研发生产力。

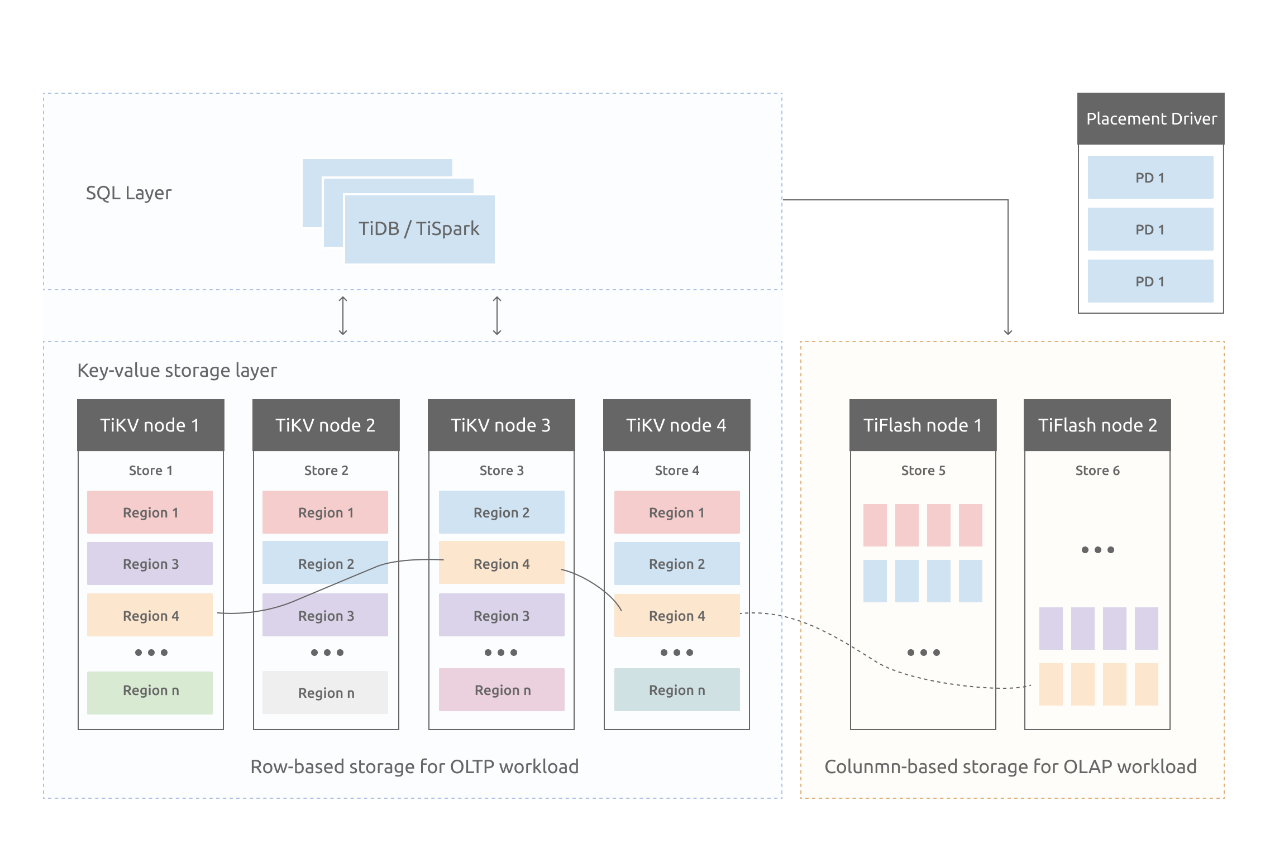
### 概述

在4.0版本上真正实现了HTAP功能。简单来说，用户可以在一套数据库上同时运行截然不同的计算负载，即联机交易的计算负载和海量数据的实时分析。此前，在数据库领域，这两种计算还不能完全放在一起，因为它们对资源的消耗、对计算本身的性能要求，以及对数据的处理方式是完全矛盾的。

通过底层数据同步及行列透明转换的技术，TiDB将TiKV面向联机交易的行存式引擎与**TiFlash面向实时分析场景的列存引擎**，转为真正的行列混合数据架构。同时，该版本在 TiDB 分布式计算层实现了透明的可根据请求自动选择行列存储引擎的能力，使高并发、低延迟的联机交易与海量数据实时分析查询计算，在 TiDB 新一代架构上完美地融合在一起。

TiFlash是TiDB HTAP形态的关键组件，它是TiKV的列存扩展，在提供了良好的隔离性的同时，也兼顾了强一致性。列存副本通过Raft Learner协议异步复制，但是在读取的时候通过Raft校对索引配合MVCC的方式获得Snapshot Isolation的一致性隔离级别。这个架构很好地解决了 HTAP 场景的隔离性以及列存同步的问题。

### 架构



上图为TiDB HTAP形态架构，其中包含TiFlash节点。

TiFlash提供列式存储，且拥有借助ClickHouse高效实现的协处理器层。除此以外，它与TiKV非常类似，依赖同样的Multi-Raft体系，以Region为单位进行数据复制和分散。

TiFlash以低消耗不阻塞TiKV写入的方式，实时复制TiKV集群中的数据，并同时提供与TiKV一样的一致性读取，且可以保证读取到最新的数据。TiFlash中的Region副本与TiKV 中完全对应，且会跟随TiKV中的Leader副本同时进行分裂与合并。

TiFlash可以兼容TiDB与TiSpark，用户可以选择使用不同的计算引擎。

TiFlash推荐使用和TiKV不同的节点以做到Workload隔离，但在无业务隔离的前提下，也可以选择与TiKV同节点部署。

TiFlash暂时无法直接接受数据写入，任何数据必须先写入TiKV再同步到TiFlash。TiFlash以learner角色接入TiDB集群，TiFlash支持表粒度的数据同步，部署后默认情况下不会同步任何数据，需要按照按表构建TiFlash副本一节完成指定表的数据同步。

TiFlash主要包含三个组件，除了主要的存储引擎组件，另外包含tiflash proxy和pd buddy组件，其中tiflash proxy主要用于处理Multi-Raft协议通信的相关工作，pd buddy负责与PD协同工作，将TiKV数据按表同步到TiFlash。

对于按表构建TiFlash副本的流程，TiDB接收到相应的DDL命令后pd buddy 组件会通过TiDB的status端口获取到需要同步的数据表信息，然后会将需要同步的数据信息发送到PD，PD根据该信息进行相关的数据调度。

### 核心特性

TiFlash主要有异步复制、一致性、智能选择、计算加速等几个核心特性。

**异步复制**

TiFlash中的副本以特殊角色（Raft Learner）进行异步的数据复制。这表示当 TiFlash节点宕机或者网络高延迟等状况发生时，TiKV的业务仍然能确保正常进行。

这套复制机制也继承了TiKV体系的自动负载均衡和高可用：并不用依赖附加的复制管道，而是直接以多对多方式接收TiKV的数据传输；且只要TiKV中数据不丢失，就可以随时恢复TiFlash的副本。

**一致性**

TiFlash提供与TiKV一样的快照隔离支持，且保证读取数据最新（确保之前写入的数据能被读取）。这个一致性是通过对数据进行复制进度校验做到的。

每次收到读取请求，TiFlash中的Region副本会向Leader副本发起进度校对（一个非常轻的RPC请求），只有当进度确保至少所包含读取请求时间戳所覆盖的数据之后才响应读取。

**智能选择**

TiDB可以自动选择使用TiFlash列存或者TiKV行存，甚至在同一查询内混合使用提供最佳查询速度。这个选择机制与TiDB选取不同索引提供查询类似：根据统计信息判断读取代价并作出合理选择。

**计算加速**

TiFlash对TiDB的计算加速分为两部分：列存本身的读取效率提升以及为TiDB分担计算。其中分担计算的原理和TiKV的协处理器一致：TiDB会将可以由存储层分担的计算下推。能否下推取决于TiFlash是否可以支持相关下推。

## TDSQL

## GoldenDB

GoldenDB目前的HTAP架构采用的是TP和AP在计算上分离，存储上共享的方式，但是DB采用的仍然都是行存储，没有采用行列混合存储。

AP引擎采用的是基于presto做二次开发。

# 发展方向

## 弱耦合/资源隔离

## 行列混合存储

## Serverless