



第十四届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA

数智赋能 共筑未来



北京国际会议中心 | 2023/8/17-19



存查算一体图数据库ArcGraph

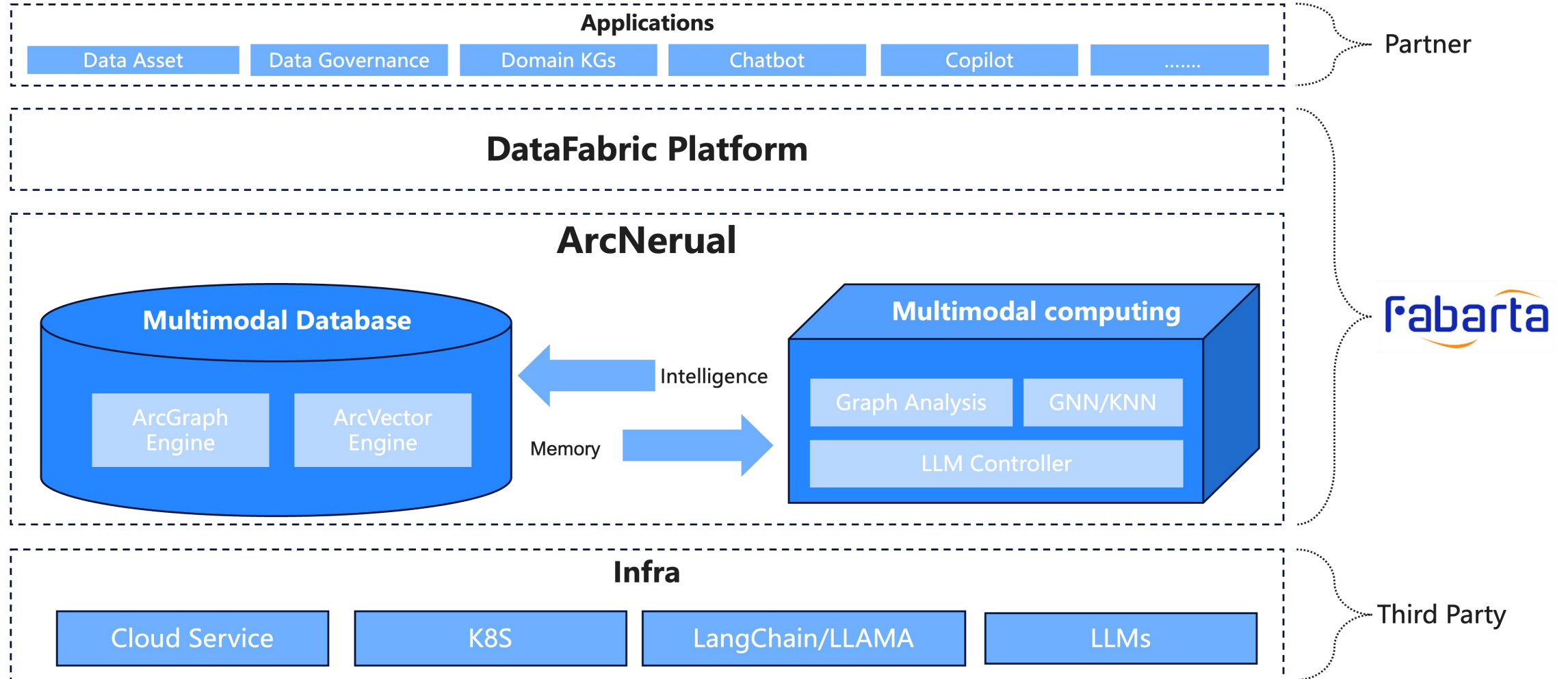
Fabarta CTO&联创 杨成虎

Fabarta产品矩阵



实现大模型时代数据、算力和模型的一体化，构建大模型时代的基础设施，帮助企业快速构建大模型时代的 AI 应用

Fabarta多模态智能引擎--ArcNeural



图数据库系统的设计挑战

数据存储 随机访问

- 图原生存储结构
- 拓扑跳转扩散读
- 多节点跨机交互
- 数据预取效率低
- 图索引结构加速
- 图结构分布式查询

混合负载

- 1-2跳，简单查询计划，需要高吞吐
- 多跳，复杂查询计划，需求低延迟
- 图计算涉及多轮数据迭代
- CPU/IO/内存资源的调度控制

图结构 扩展

- 时序图
- 图与向量
- 图与文档
- 多模态数据管理
- 多模态联邦查询

数据库 通用难题

- 分布式事务
- 查询优化器
- 数据正确性
- 数据可靠性
- 弹性扩缩容
- 行列混存
- 数据实时性
-

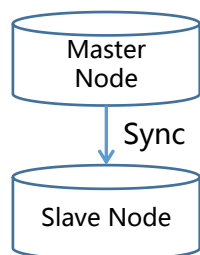
图数据库系统架构演进

图数据库架构

特点

第一代：单机

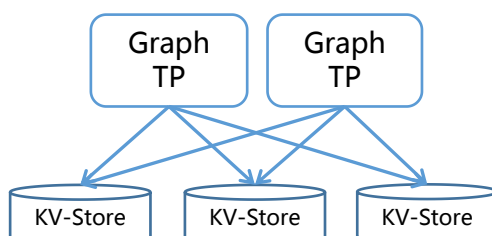
业务核心拓扑关系



- ✓ 架构简单、易于使用

第二代：分布式—TP优化

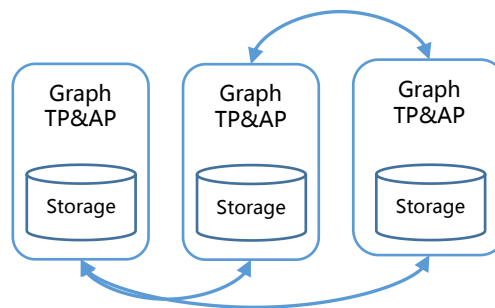
互联网高并发在线业务处理



- ✓ TP场景1-2跳性能友好

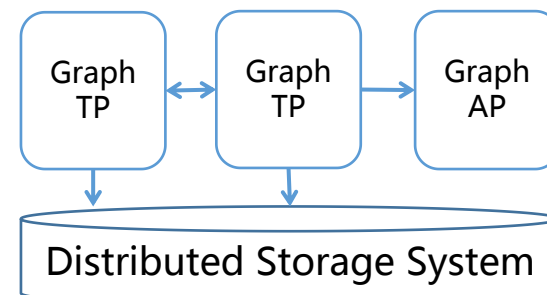
第二代：分布式—AP优化

企业数据离线分析处理



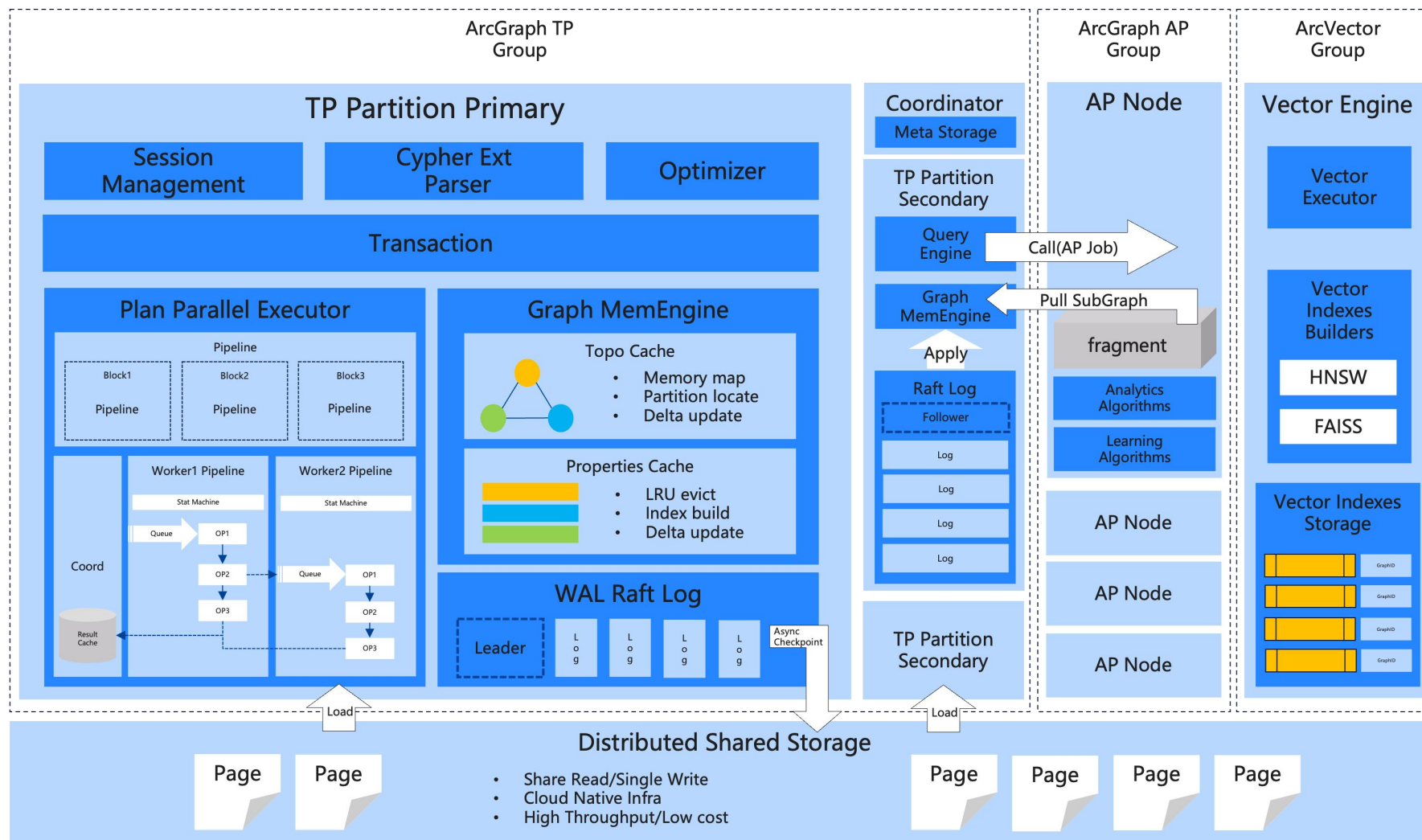
- ✓ AP场景性能友好
- ✓ TP不能独立弹性
- ✓ 存储不能独立弹性

最新一代：存算分离HTAP

DataFabric数据核心：
离在线混合、Cloud Native、AI Native

- ✓ TP/AP混合一体化，AP数据实时新鲜
- ✓ 1-3跳高吞吐、多跳低延迟
- ✓ 模块组件按需弹性

ArcGraph : 系统架构

Cloud
NativeGraph
HTAPVector
Embedded

ArcGraph : 存

内存

TOPO索引

- 常驻内存
- 结构紧凑排列
- 反向边索引

属性缓存

- LRU置换
- 索引支持

MemEngine

脏页管理

- Delta Update
- CPU Cache优化
- 无锁并发

事务管理

- MVCC
- 分布式事务

同步写入本地日志并更新内存，低延迟高吞吐。
不依赖底层存储能力即可实现上层事务功能。

- 拓扑与属性分离存储，独立优化拓扑索引查找和属性条件过滤；
- 拓扑常驻内存，属性LRU；
- 脏页；
- MemEngine针对图结构优化并发与内存空间；

本地存储

Log as Database

- Log Buffer

WAL日志

- Group Commit

- LSN管理

Raft共识

- Leader选举
- 数据多副本

异步checkpoint，并Merge Delta。解耦远端存储架构模式，降低性能依赖

- Log本地化IO写入，发挥本地磁盘高带宽优势；
- 多副本Raft，保障数据安全可靠；
- 内存副本数 ≤ Log副本数，有效降低内存成本；

远端存储

Storage Pool

存储适配器：抽象多种存储模式数据访问接口

模式1：分布式共享KV存储

Page共享存储

- 图转换为KV Page存储
- 可适配主流分布式KV存储引擎
- KV存储引擎保障数据可靠性

模式2：敏捷部署

本地磁盘

- 单机或高可用
- 多副本可读扩展

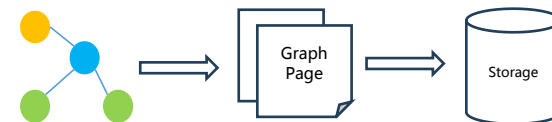
模式3：分布式块存储

共享块存储

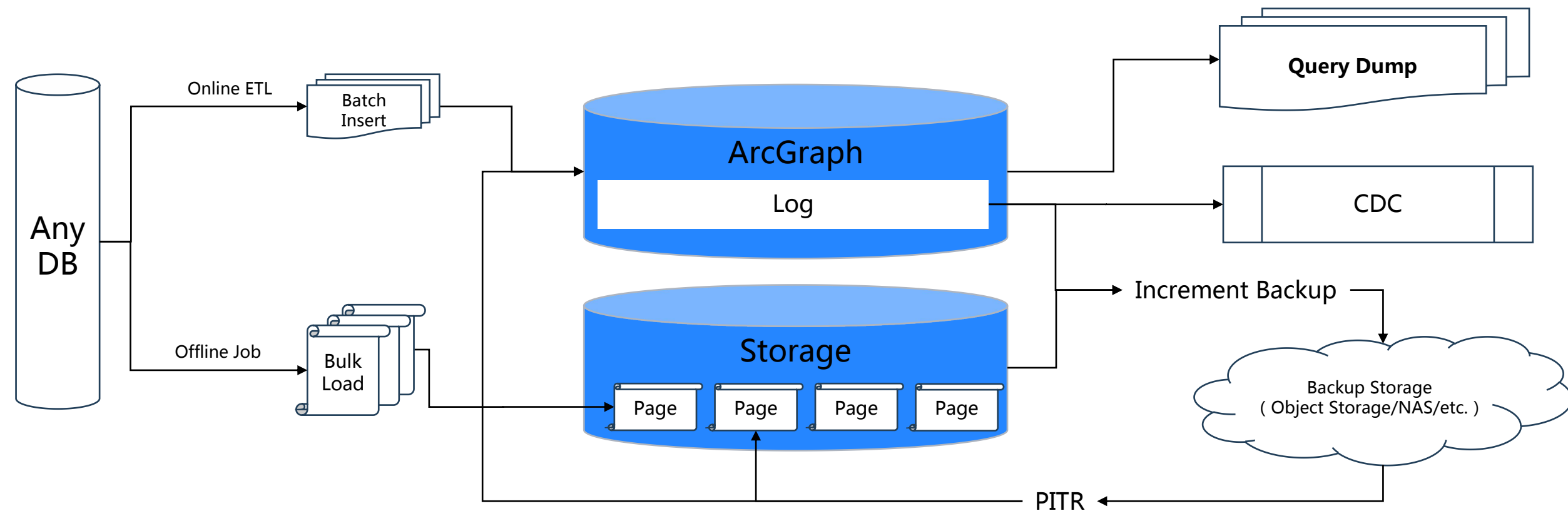
对象存储

- Cloud Native Storage
- 低成本高性能
- 面向图系统优化

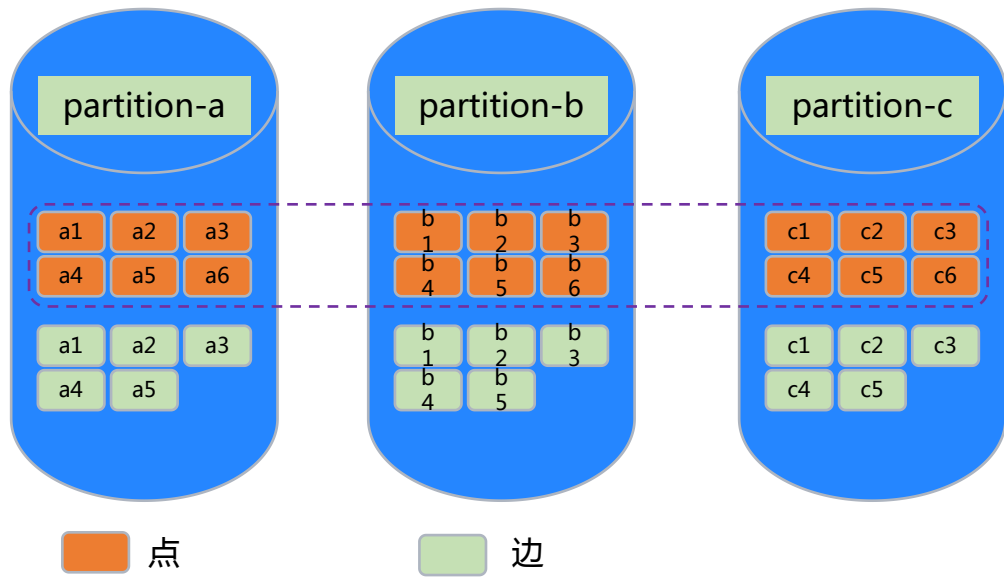
- 存算分离，计算与存储能力解耦；
- 异步Checkpoint写入外部存储；
- 按需支持多种存储结构，满足2B客户多样性的交付环境与需求；
- 图原生持久存储结构；



ArcGraph : 存—数据导入导出



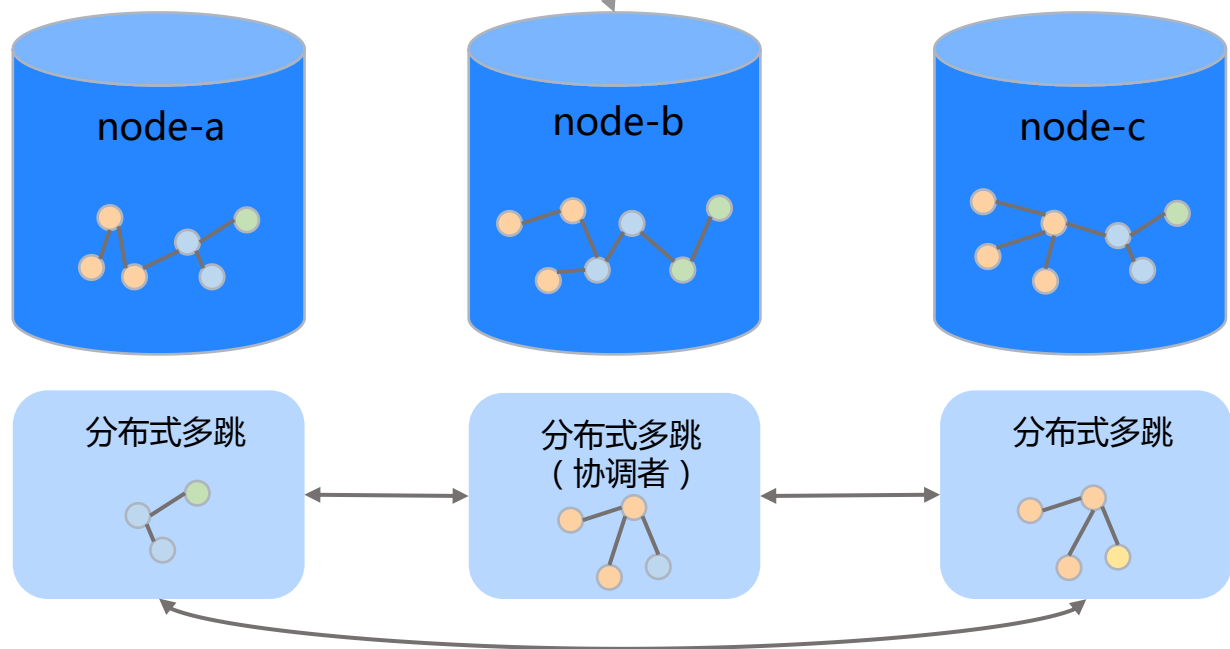
ArcGraph : 查



```
CREATE GRAPH if not exists LDBC (PARTITION_NUM=3) CHARSET=utf8
COMMENT = 'ldbc'
```

- 灵活分区策略：Hash(vertex_type), Hash(vertex_id) ...
- 基于分区策略达到点/边/索引数据的分布式分布存储；
- 边的分区采用起始点分区一致，加速多跳查询；
- 点的location信息可基于vertex_id进行计算获取；

MATCH (m: person {first_name: 'Andy'} - [e: knows: 5] - (n: person) RETURN n;



- 接收Cypher语句的计算节点进行语句的解析，优化并生成执行计划，然后基于图数据分区的分布进行plan分发；
- 计算集群中涉及的计算节点并行执行plan，比如：并行启动起始点过滤，然后开始执行多跳；
- 在多跳过程中，如果发现下一跳的起点不在当前计算节点，进行分布式通信，到目的地计算节点进行后续多跳；
- 最后，承担协调角色的节点进行结果收集，并返回给客户端；

ArcGraph : 算

TP : 低延时高并发

```
Create :  
CREATE (:Person {name: "Alice", company: "Fabarta"})  
CREATE (alice)-[:likes]->(bob)  
Query :  
MATCH (a:Person)-[:likes]->(b:Person)  
WHERE a.company = "Fabarta"  
RETURN a, b;
```

HTAP : 实时复杂计算

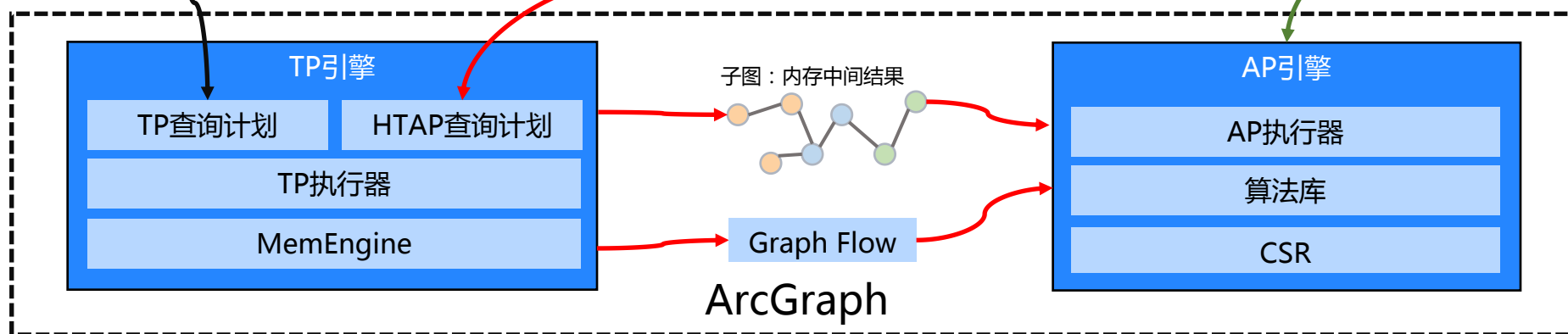
```
Query :  
MATCH (a:Person)-[:likes]->(b:Person)  
WHERE a.company = "Fabarta"  
Call pagerank((n)-[e]->(b), 0.85, 10) YIELD *  
return id,result;
```

AP : 离线超复杂计算

Data : CSV, TXT, HDFS etc.

Query:

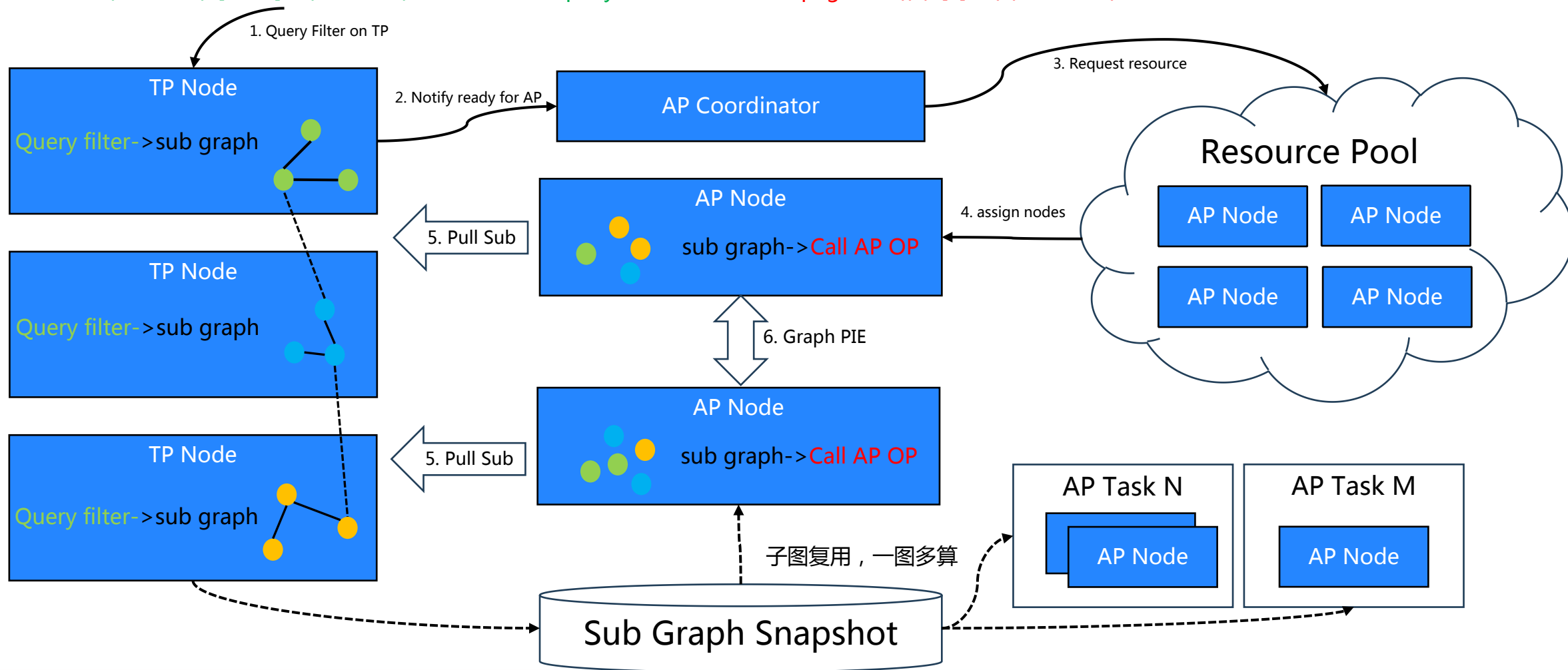
```
Call graph_matching(graph, edge_csv_path,  
vertex_csv_path) YIELD * return sub_graph_list
```



- 统一Cypher查询语句与调用入口
- AP数据实时全量/增量计算
- 旁路T+1超复杂计算
- TP/AP资源独立弹性应对混合多场景负载

ArcGraph : Serverless AP

MATCH (a:Person)-[:likes]->(b:Person) WHERE a.company = "Fabarta" Call pagerank((n)-[e]->(b), 0.85, 10) YIELD * return id,result;



ArcGraph : Graph-HTAP

	外部组合图数据库+图计算系统	ArcGraph-HTAP
性能	低于内置。图DB中的数据要先导出，再导入外挂图计算系统，并进行数据格式转换（如行转列），导致整体计算时间长。	高于外置。在引擎内部，图计算模块通过共享内存方式操作查询模块的数据，速度是SSD的1500倍左右。
资源利用率	低于内置。两套系统完全分离，导致任一系统资源空闲时，无法被另一系统及时使用。	高于外置。引擎内部的资源调度器根据任务负载合理分配系统的资源给图计算与TP模块。
横向扩展性	高于内置。可按需更灵活地扩展图计算所需资源。	低于外置。在扩展时需要同时扩展图计算及其依赖模块的资源。
运维便捷性	低于内置。需要同时对两套独立的系统及其集成方式做运维。	高于外置。只需针对一套系统做运维。
存查分析一体化的场景支持	无法支持。存查与分析为两套系统，导致存查分析被割裂为两个场景，用户使用体验差。	已支持。ArcGraph可对查询结果直接做图分析，并支持实时返回分析结果。

ArcGraph-核心特性

多模态

Graph

Vector

Json

Table

企业级数据管理

ACID

内存
引擎

多跳
并行化

短跳
高吞吐

GPU
硬件
加速

Online
DDL

企业级交付架构

弹性
云原生

分布式

Raft
多副本

多地
多中心

全面自主可控

国产化

Rust

面向合
作企业
开源

THANKS

TDDL

DistributedTable

DBproxy

HBase

PostgreSQL

SSD

MongoDB

Cassandra

GreatDB

Hyperbase

Hubble

DataCenter

VisualDataPlatform

Blockchain

ArgoDB

Distributed

DatabaseKernel

TemporalData

CloudnativeData

AIalgorithm