DTCC

数，造，未，来

第十二届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2021

2021 年 10 月 18 日 - 20 日 | 北京国际会议中心

Spark
Hadoop
OLAP
HANA
MPP
Oracle
SQL

IT168.com ChinaUnix.net ITPUB

HTAP系统的问题与主义之争

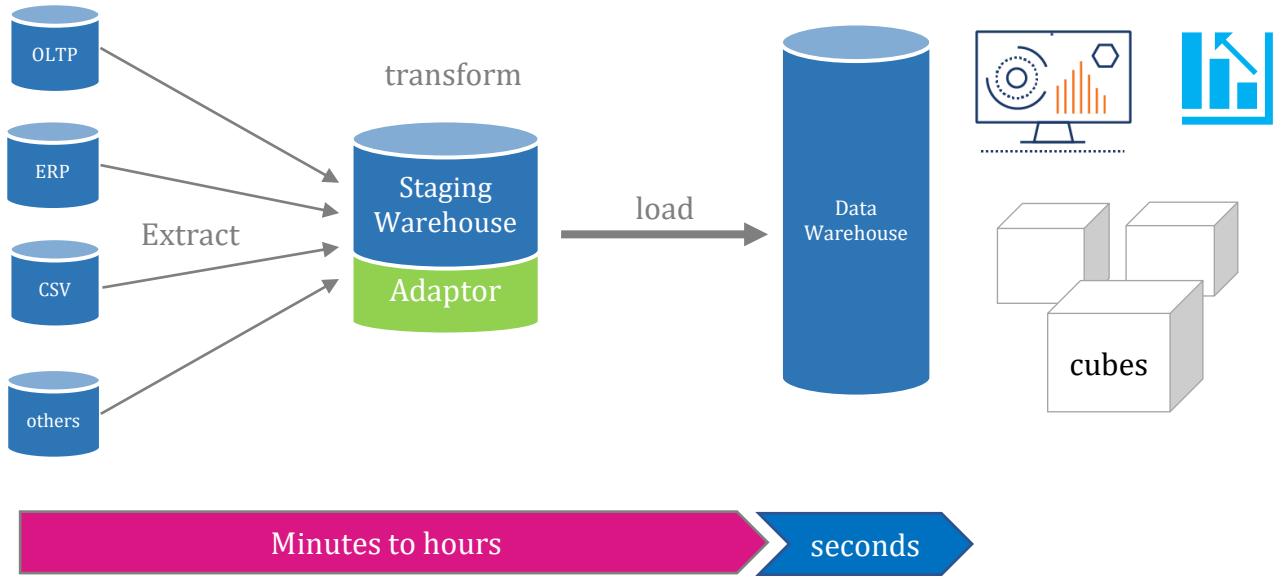
腾讯计费平台部 朱阅岸

CONTENTS

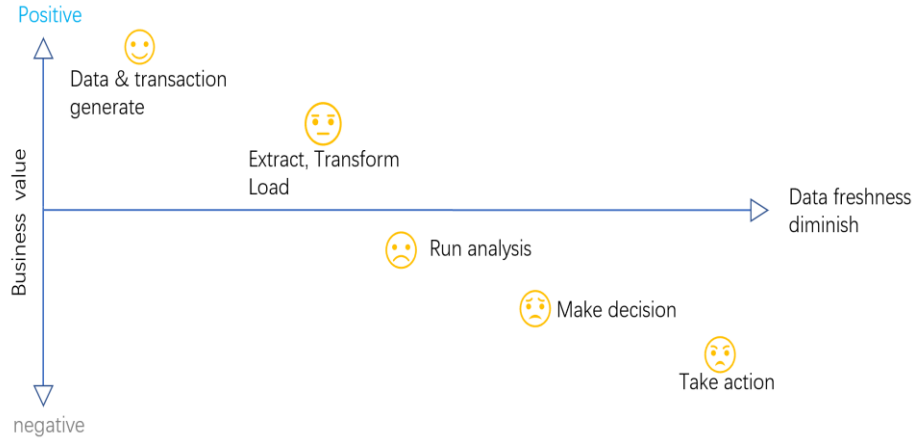
01 HTAP的定义与场景需求

问题与挑战

典型的数据处理分析框架



1. $13.01 for every $1 a company spend on analytics
   source: MIT Sloan, Nuclues Research
2. 74% of firms say they want be data-driven.
   But only 23% are successful
   source:forbes:Actionable insight: missing link between data and value
3. 2x[company are twice] likely to outperform their peers if they use advanced analytics
   source:MIT Sloan



数据价值随时间流逝而降低

问题与挑战

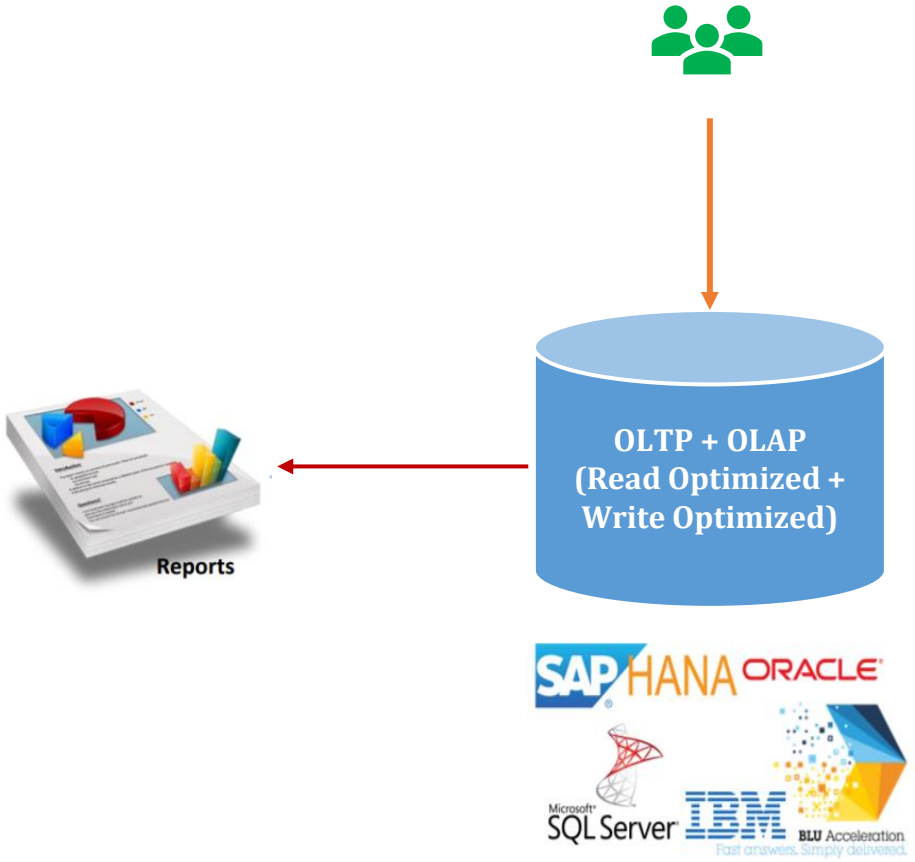
What is HTAP?

Hybrid Transaction and Analytical Processing (HTAP) is an emerging application architecture that *breaks the wall between transaction processing and analytics*. It enables more informed and "in business real time" decision making.

HTAP will empower application users to innovate via *greater situation awareness* and *improved business agility*.

This will greatly entail an upheaval in the established architectures, technologies and skills driven by means of the *modern OLAP* technologies as enablers.

— modified from Gartner 2014

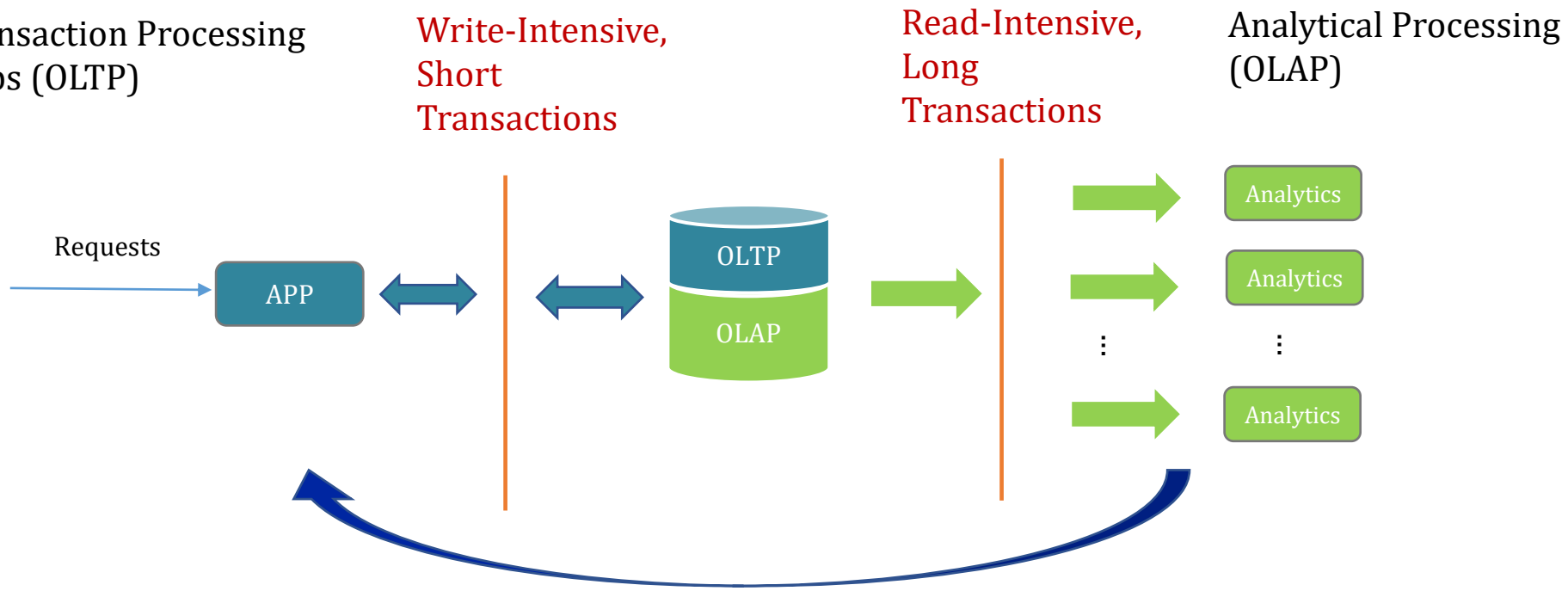


问题与挑战

The emerging HATP?

Hybrid Analytical and Transaction Processing (HATP) is the other emerging application architecture that *embraces transaction processing ability in analytical system*. It also aims to enable more informed and “in business real time” decision making.

问题与挑战



the maximum attainable OLTP throughput, when co-scheduled with a large analytical workload, is reduced by at least 3~5X.

Psaroudakis Iraklis et al. Scaling up Mixed Workloads: a Battle of Data Freshness, Flexibility, and Scheduling,TPCTC 2014

问题与挑战

摆在我们面前就是采用哪一种方案以更好、更快、更高效地实现HTAP以满足业务需要

1. 在现有系统上延伸，扩展， 满足业务需要？（问题导向）
2. 从头开始实现一个系统？（彻底的、革命性的颠覆）
3. 哪一种方法更好（问题 VS 主义）

02 HTAP系统的架构实践

HTAP系统架构实践

HTAP系统的类别

1. One size fits all 策略
   1.1 单系统单拷贝
   1.2 单系统双拷贝
2. One size doesn't fit all 模型
   2.1 共享存储上的松耦合系统
   2.2 独立存储的松耦合系统

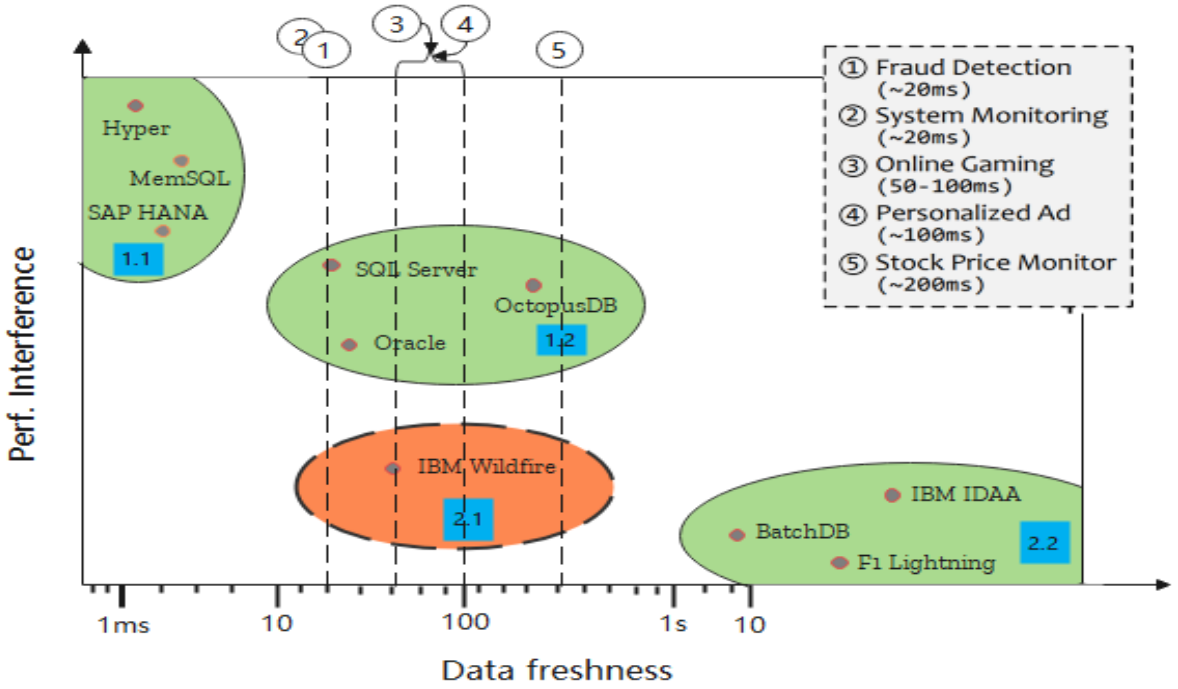
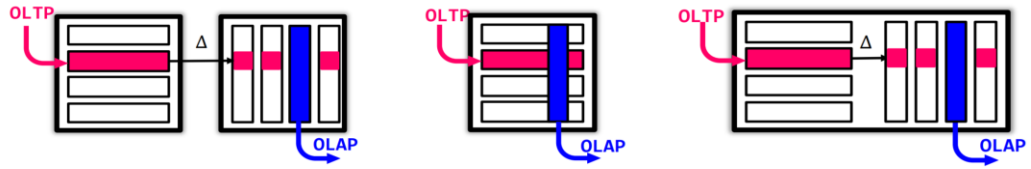


图1. HTAP系统类别



(2.2) (1.1) (1.2, 2.1)

图2. 不同HTAP系统的存储模式

Sijie Shen et al. Retrofitting High Availability Mechanism to Tame Hybrid Transaction/Analytical Processing, OSDI 2021
Özcan et al. Hybrid transactional/analytical processing: A survey, *ICMD* 2017.

单系统单拷贝之Hyper

- 查询执行模式
  - OLTP串行执行, 运行时刻无需加锁.OLAP通过系统调用fork产生子进程在事务一致性的拷贝上运行
  - 利用OS的copy-on-update技术将TP的更新操作定向到新的page上进行
  - OLAP可以通过无锁的方式并行执行; OLTP通过划分数据分片的方式达到并行执行效果
  - 利用硬件协助区分冷热数据. 热数据不做压缩, 存储在正常页面上; 冷数据压缩存储, 存放在大页面上
- 执行引擎
  - 采用向量化执行引擎, 利用LLVM生成可执行机器码
  - 以C++为轮子，驱动LLVM代码生成

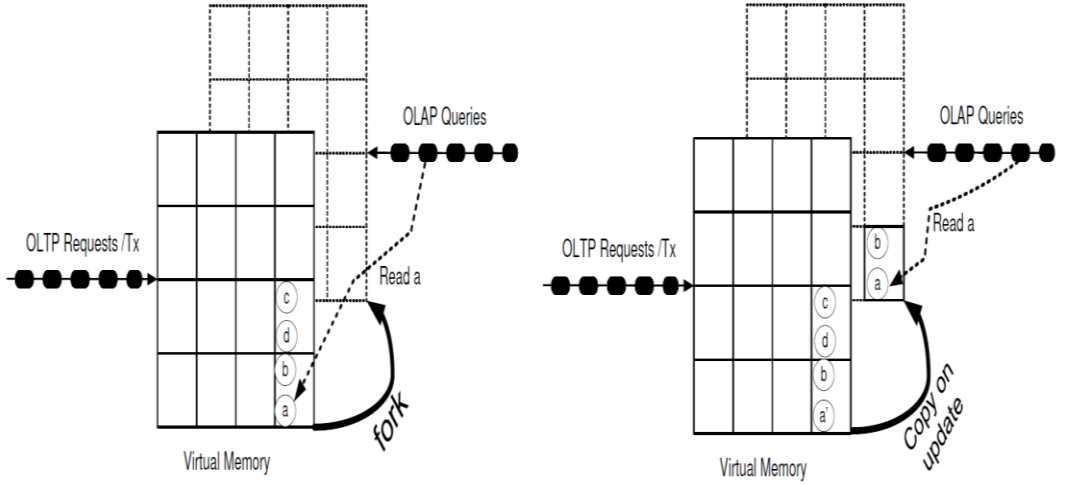


图1. Hyper系统原理

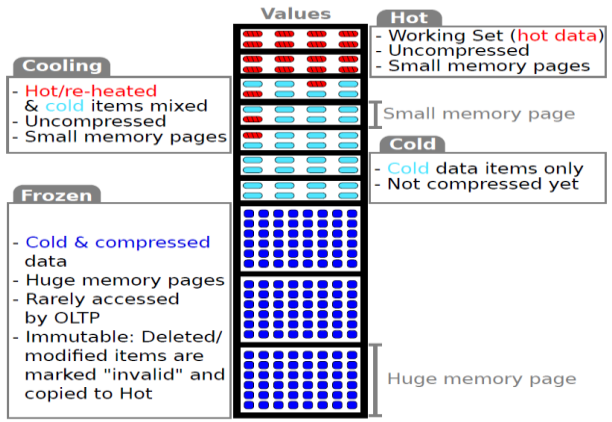


图2. 区分冷热数据

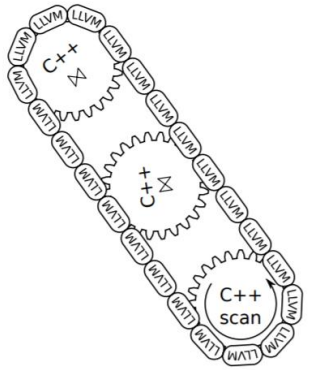


图3. 代码生成

Alfons Kemper et al. HyPer: A Hybrid OLTP&OLAP Main Memory Database System Based on Virtual Memory Snapshots, ICDE 2011
Florian Funke et al. Compacting Transactional Data in Hybrid OLTP&OLAP Databases, *VLDB* 2012.

HTAP系统架构实践

单系统单拷贝之SAP HANA

- HANA定位
  - 支持多引擎、多类工作负载的统一数据库管理平台
- 系统总体框架
  - 分层架构: 系统自顶向下由**查询编译层**与**查询执行层**构成，如图1所示
  - 查询编译层首先将各类查询字符串转换成特定领域的**抽象语法树AST**. 然后映射到Calculation Graph进行查询优化
  - 分布式执行框架构建实际的数据流, 将物理执行算子分发到特定的引擎执行. Engine Layer提供针对特定引擎的物理算子, 例如Relational Operators, Graph Operators

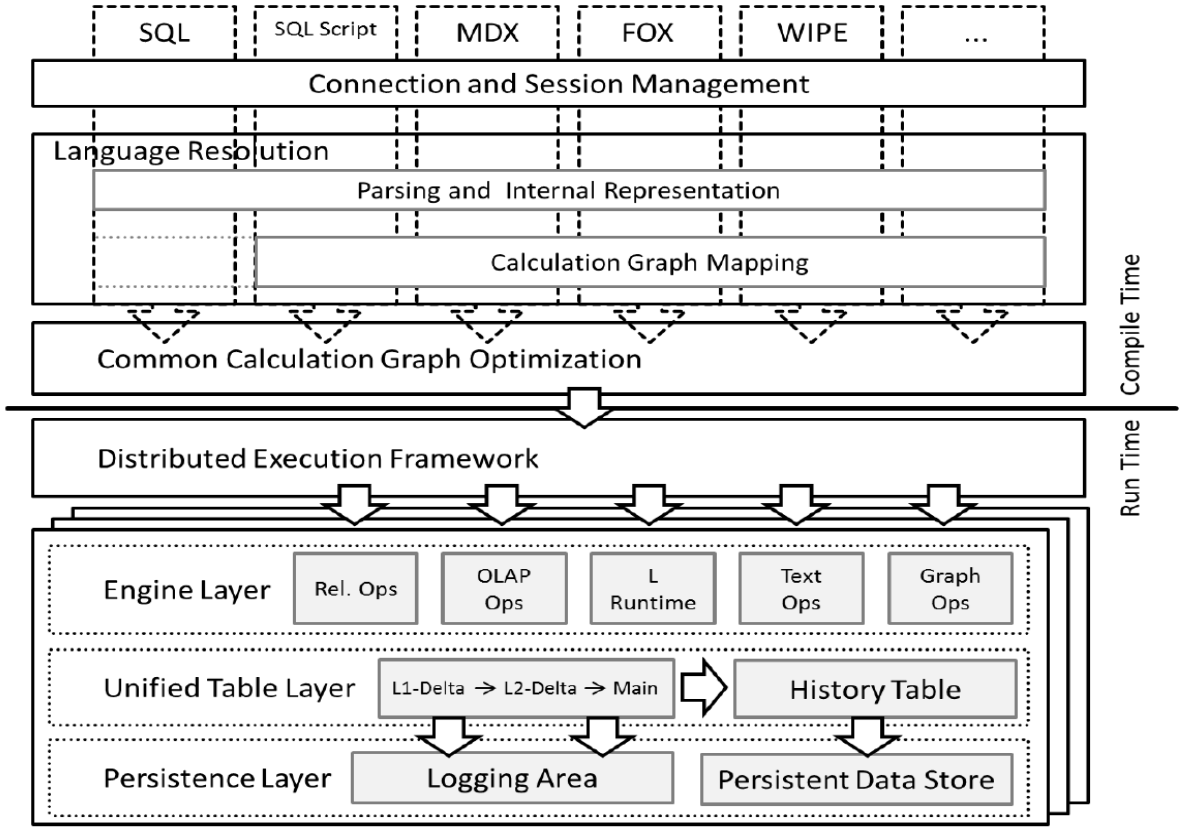


图1. 系统总体架构

Franz Färber et al. SAP HANA Database - Data Management for Modern Business Applications, SIGMOD 2011

单系统单拷贝之SAP HANA

- 存储设计
  - 向上提供统一的接口
  - 分成L1-delta, L2-delta, L3-main store三级存储, 分别对应无压缩行存储、轻量级压缩列存储、重度压缩列存储
- 合并操作
  - 合并操作前, 读操作在Main1与Delta1上进行; 合并操作开始时刻开辟另外一套缓冲区: Main2与Delta2. 此时读操作在Main1, Delta1, Delta2上进行; 最终Main1与Delta1的数据合并到Main2.
  - L1-delta采用redo日志, Main store采用影子页技术保证数据一致性与持久性

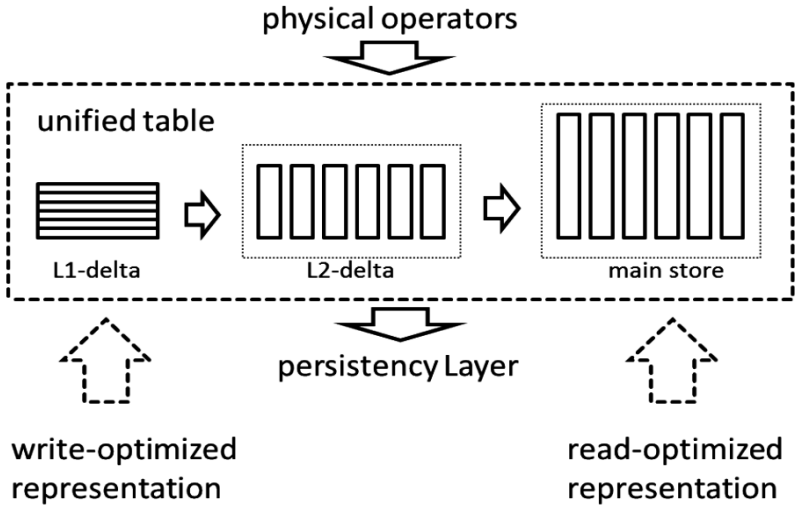


图1. 表接口与存储设计

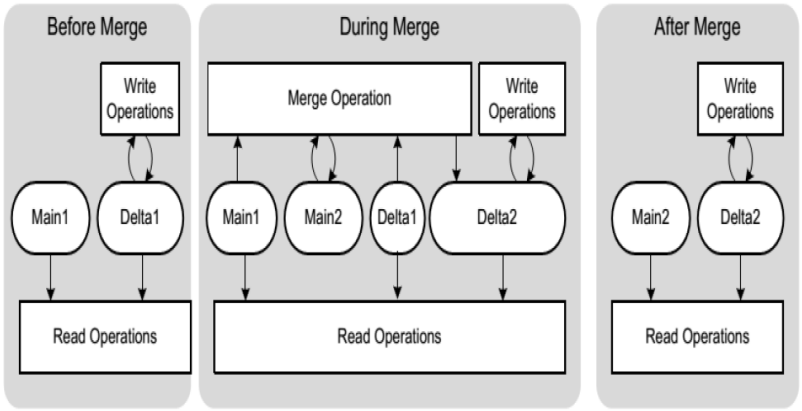


图2. 合并操作

Vishal Sikka et al. Efficient Transaction Processing in SAP HANA Database –The End of a Column Store Myth, SIGMOD 2012

单系统双拷贝之SQL Server

- 列存储索引
  - 行存储按照1,000,000条记录为单位进行水平分割, 得到Row Group. 单独对每个Row Group的每个列进行转换得到一个个segment，然后编码压缩. 一般采用数值压缩或字典压缩的轻量级压缩算法
  - 将转换后的列与字典表(Dictionary)存放在Blob类型中. 最外面的directory存放segment的位置
  - SQL Server针对这些列存储索引提供批量执行的算子, 加速分析操作.

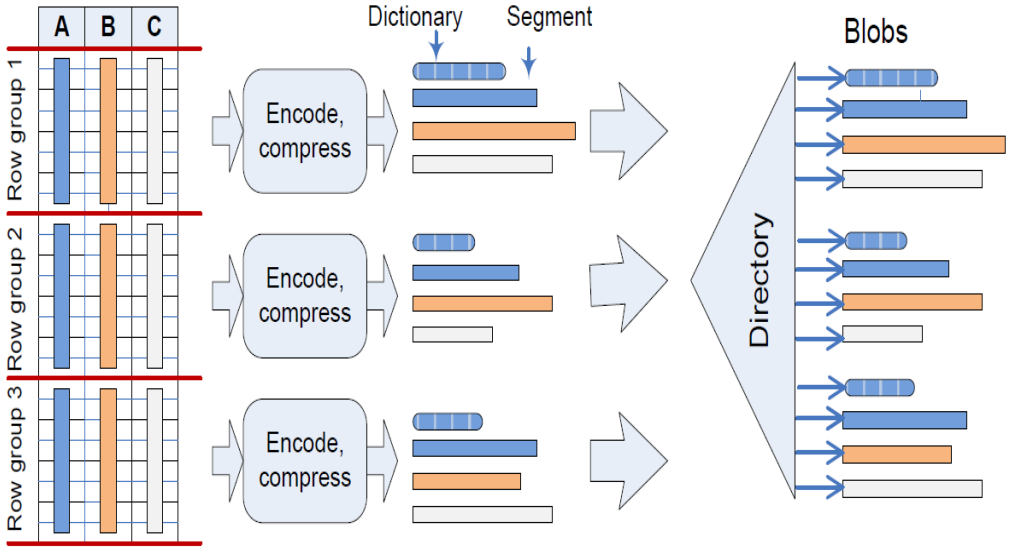


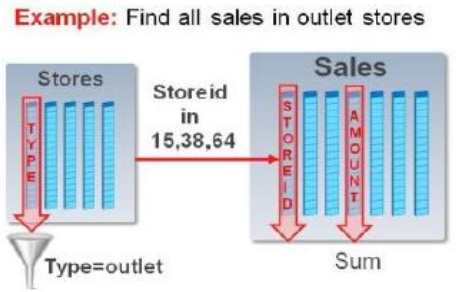
图1. 列存储索引的建立

Per-Åke Larsonet al. SQL server column store indexes, SIGMOD 2011

单系统双拷贝之Oracle

- Oracle行列共存
  - 系统将一份数据同时以行存储和列存储两种格式存放. 但是事务特性的要求不一样.
  - 灵活地指定需要加载到内存的列以及其压缩特征. 也可以在系统运行时刻更改这些列存储特性. 此外也可以利用Oracle RAC进行横向扩展
  - 利用列存储高效地生成维表上的bitmap/vector, 然后在事实表中找出相关记录, 简化某些分析场景中的复杂JOIN、GROUP BY等操作

图1. Oracle行列共存模式




```
Select Stores.id, Products.id, sum(Sales.revenue)
From Sales, Stores, Products
Where Sales.store_id = Stores.id
And Sales.product_id = Products.id
And Stores.type = "Outlet"
And Products.type = "Footwear"
Group by Stores.id, Products.id;
```

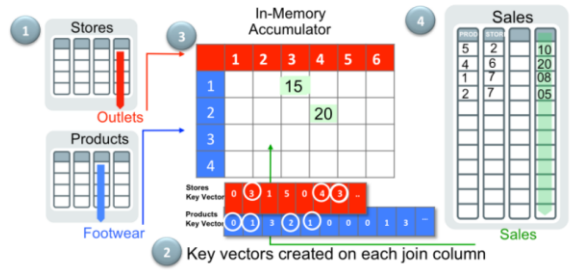


图2. 利用列存储提升业务分析场景性能

Niloy Mukherjee al. Distributed Architecture of Oracle Database In-memory, VLDB 2015

松耦合共享存储之Wildfire

- IBM Wildfire
  - 系统分成有状态(stateful)与无状态(stateless)两类节点, 分别处理事务与要求最新数据的分析型查询、可以容忍延迟的分析型查询
  - 每一个分析型引擎实例采用类似IBM BLU列存储分析引擎技术, 对接到spark executor, 提供高性能的数据分析能力 . 系统将数据分片, 分别存储到事务型节点, 加速数据写入
  - 事务型数据首先写入本地, 然后定期批量合并入共享文件系统

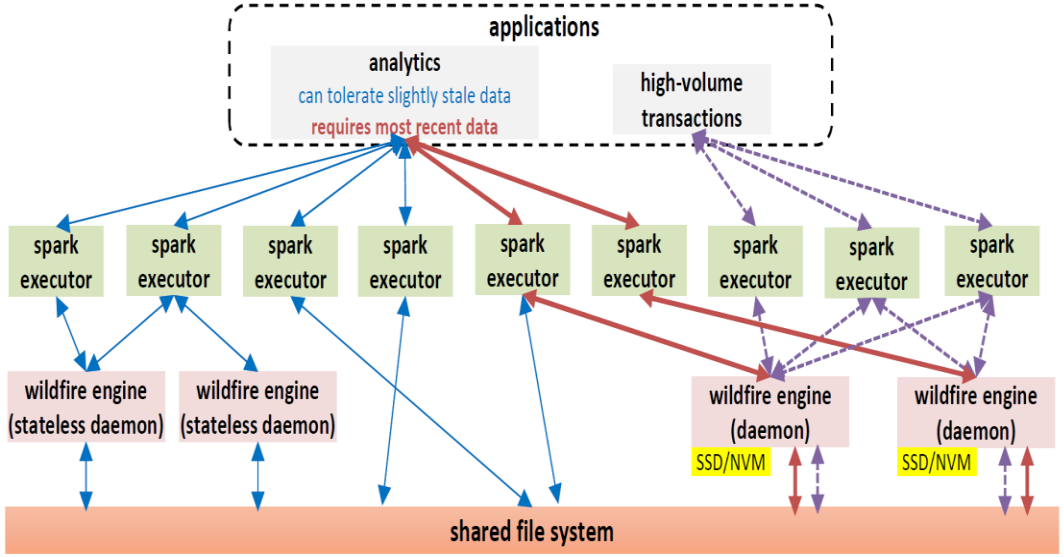


图1. wildFire架构与设计

Ronald Barber et al Evolving Databases for New-Gen Big Data Applications, CIDR 2017
Ronald Barber et al Wildfire: Concurrent Blazing Data Ingest and Analytics, SIGMOD 2016

松耦合独立存储之F1 Lightning

- Google F1 Lightning
  - 系统分成三个模块: OLTP源数据库以及对外暴露的CDC API、Lightning和F1 Query
  - Lightning包括外围changepump与存储. 开发人员在Changepump内部实现了一个适配器, 将CDC格式转换成统一的内部格式.
  - Lightning server订阅changepump, 获取相应的数据变更. Lightning采用类似LSM-tree方式维护内存与磁盘多级数据, 内存格式以行存储形式存在, 在写成磁盘阶段转变成列存储.
  - 集成F1 Query作为查询接口, 利用其快照隔离机制读取可见范围数据

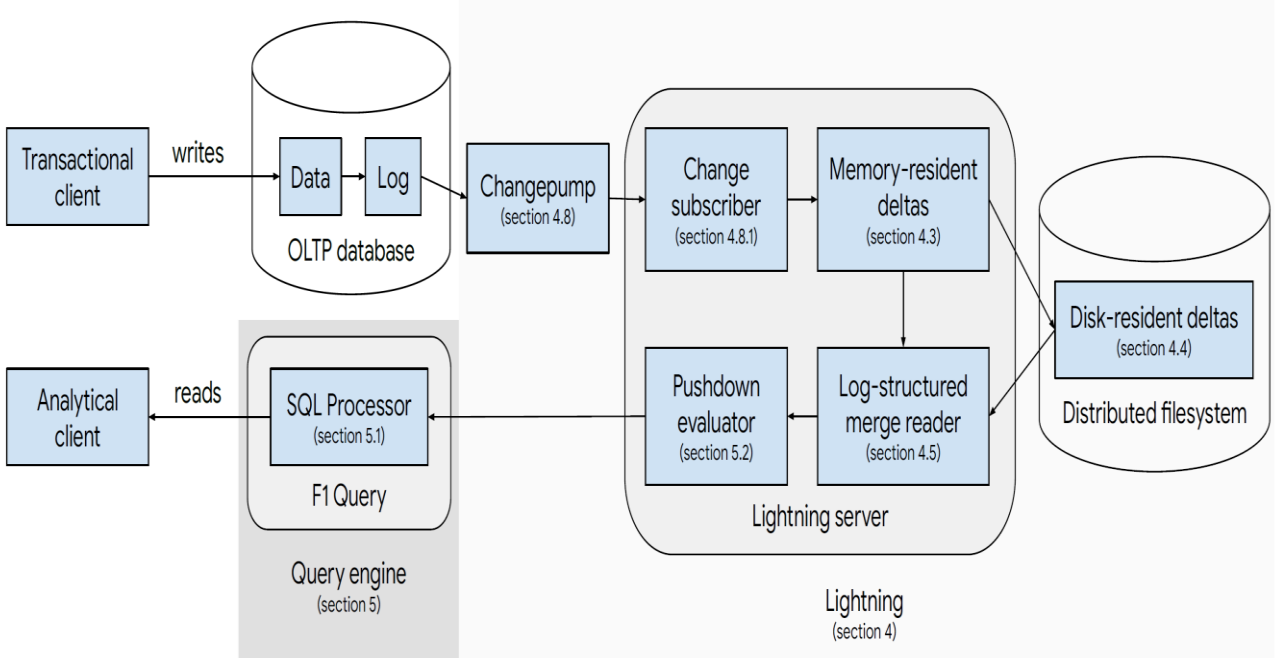


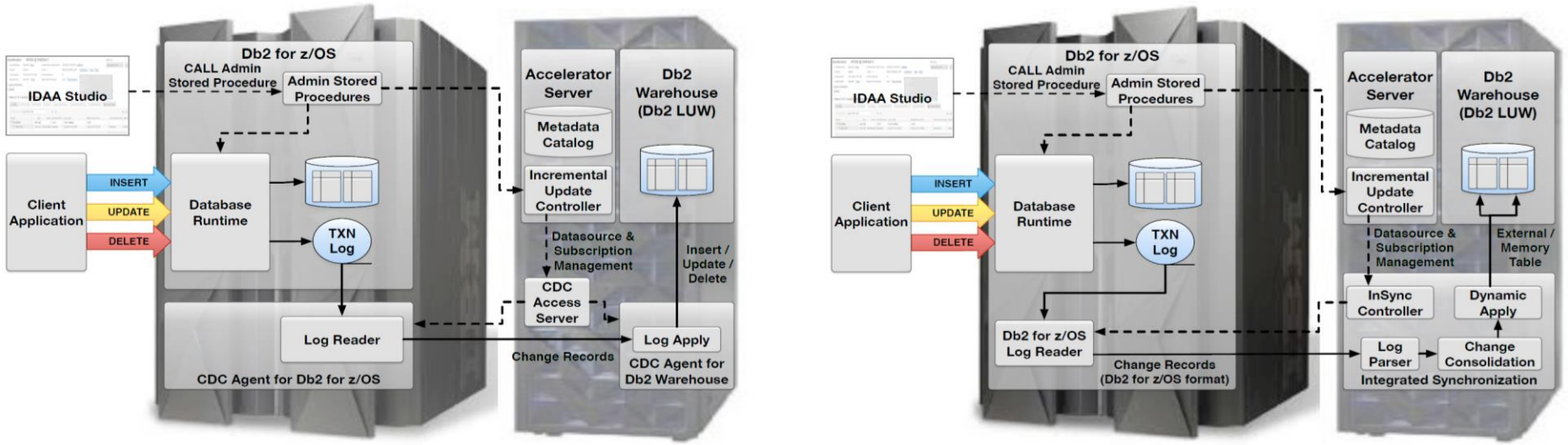
图1. Google HTAP系统

图2. 查询延迟

Jiacheng Yang et al. F1 Lightning: HTAP as a Service, VLDB 2020

松耦合独立存储之IDAA

- IBM Db2 Analytics Accelerator
  - 分析型引擎运行在IBM大型机上, 作为附件通过网络挂载到事务型引擎Db2. 对外提供统一的接口, 采用机器学习算法识别分析型查询, 然后路由到分析型引擎
  - CDC(Change Data Capture) 方案需要花费大量的时间与背景知识来维护这个外进程, 且延迟大; 而轻量级的集成同步方案能够规避上述问题, 延迟缩减~180X



CDC方案 Vs. Integrated Synchronization

Dennis Butterstein et al Replication at the Speed of Change – a Fast, Scalable Replication Solution for Near RealTime HTAP Processing, VLDB 2020

松耦合独立存储之IDAA

- IBM Db2 Analytics Accelerator
  - CDC方案在源端需要经历六个步骤: 1) **读取日志, 解密 2) 过滤无关日志, 按照LSN排序 3)** 将数据进行行列转换 4) 暂存数据直到遇上Commit或者 Rollback 日志 5) 生成CDC内部表示 6) 在发送日志之前处理回滚事务
  - 轻量级集成同步(InSync) 方案将计算密集型的3-6步骤移动到目的端, 无需等待聚集批量日志, 延迟缩减~180X

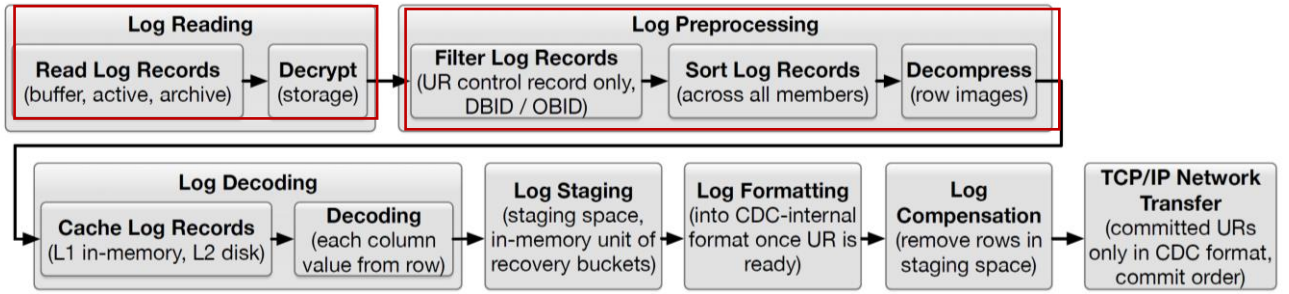


图1. CDC方案

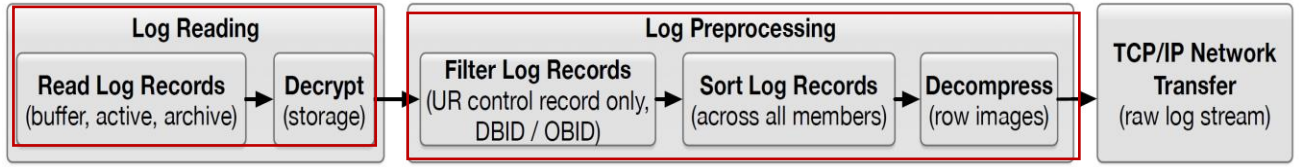
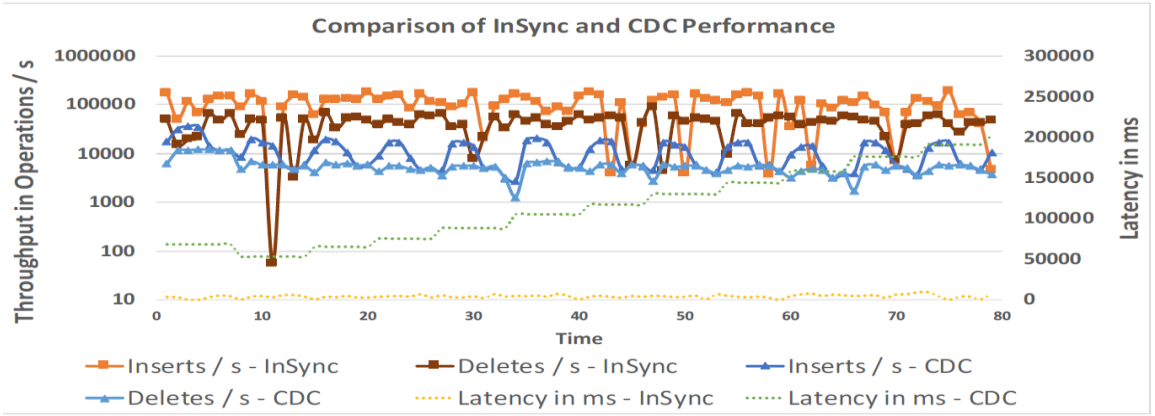
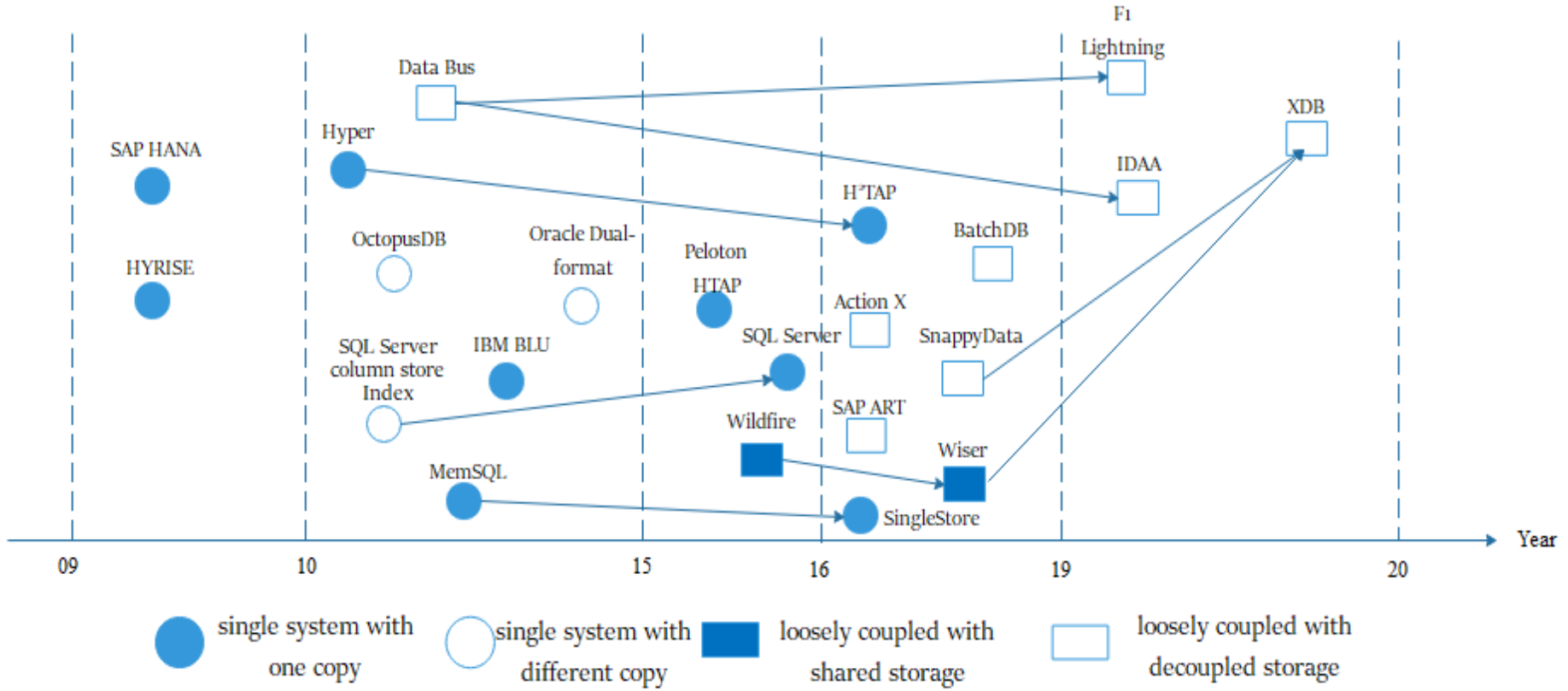


图2. InSync方案



Dennis Butterstein et al Replication at the Speed of Change – a Fast, Scalable Replication Solution for Near RealTime HTAP Processing, VLDB 2020

HTAP 技术发展一览



03 云原生对于HTAP系统的启示

云原生对于HTAP系统的启示

云原生架构与HTAP系统

1. 存算分离架构能否为HTAP的设计带来便利，解决扩缩容难题?
2. 弹性算力能否缓解HTAP中AP与TP对于资源的竞争?
3. 丰富的计算资源池与存储资源池能否针对不同的计算特征进行定制, 划分与隔离, 降低用户成本?

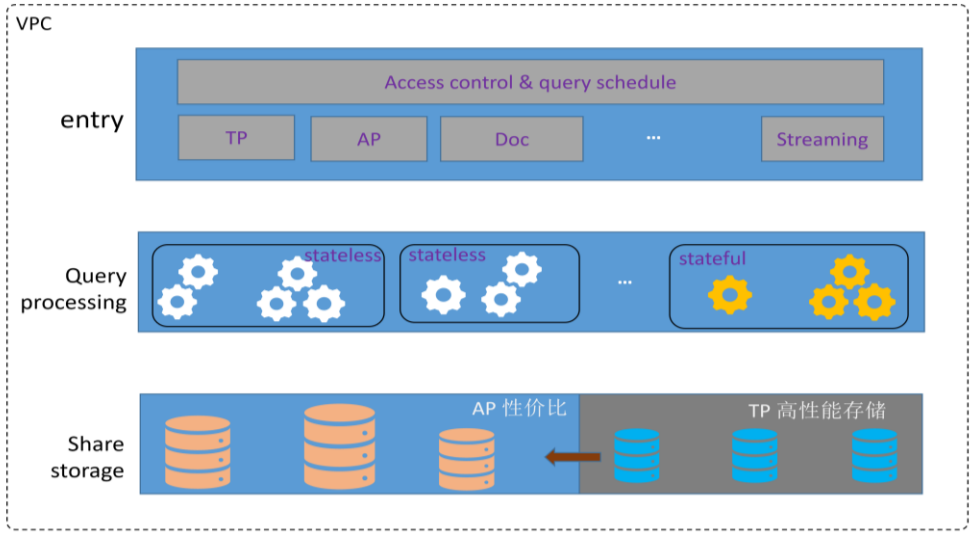


图1. a vision for cloud-native HTAP

系统	资源竞争	扩容	成本	数据可见度
SAP HANA-	▲▲▲	▲▲▲	▲▲▲	▼▼
Hyper-	▲▲▲	▲▲▲	▲▲	▼▼▼
SQL Server-	▲▲▲	▲▲▲	▲▲	▼▼
Oracle-	▲▲▲	▲▲▲	▲▲▲	▼▼
Google F1 Lightning*	▼▼▼	▼▼▼	▼▼	▲▲▲
IBM IDAA*	▼▼▼	▲▲▲	▲▲▲	▼

▲ 表现不佳
▼ 表现较优
\- one
* coupled

04 总结

总结

已有问题

TP与AP互斥的属性

架构实践

大统一系统 Vs. 松耦合

潜在机遇

云原生弹性算力天然适合


THANKS
时序
开源
自研
分布式
文档
Spark
HANA
SQL
OLAP
Oracle
Hadoop
MPP
DB2