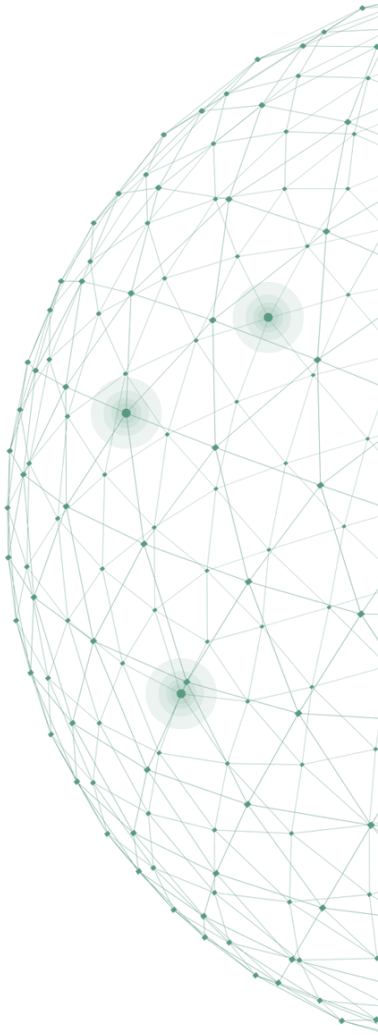
openLooKeng :
Looking Back, Looking Forward

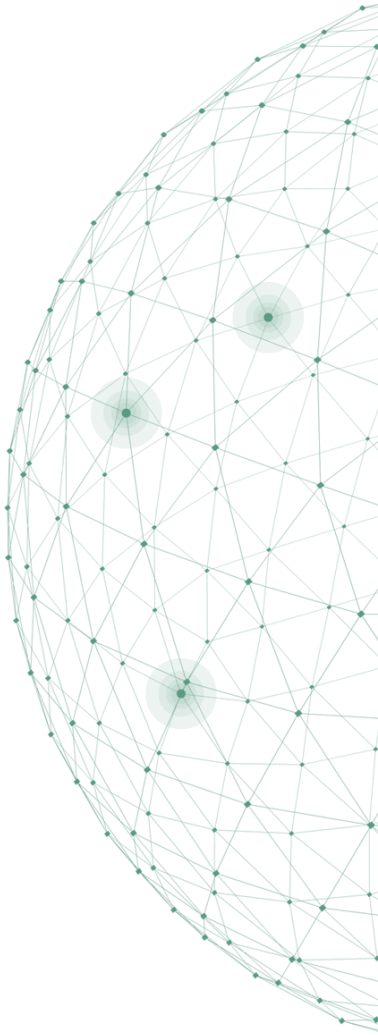
李铮

openLooKeng committer

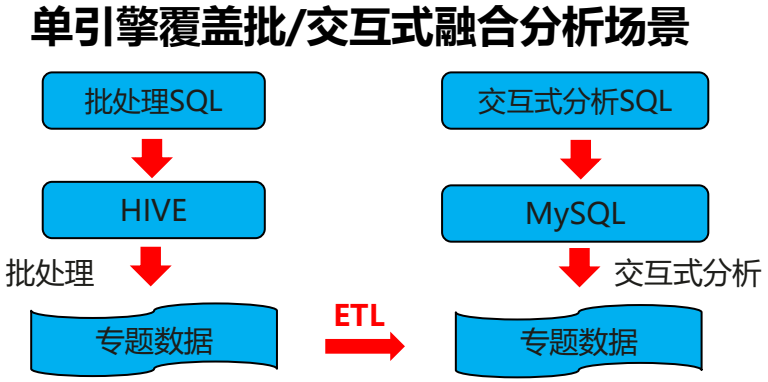
目录



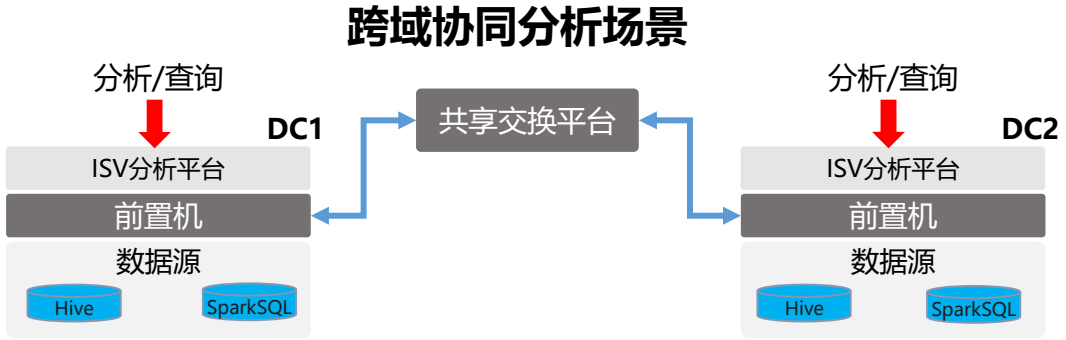
目录



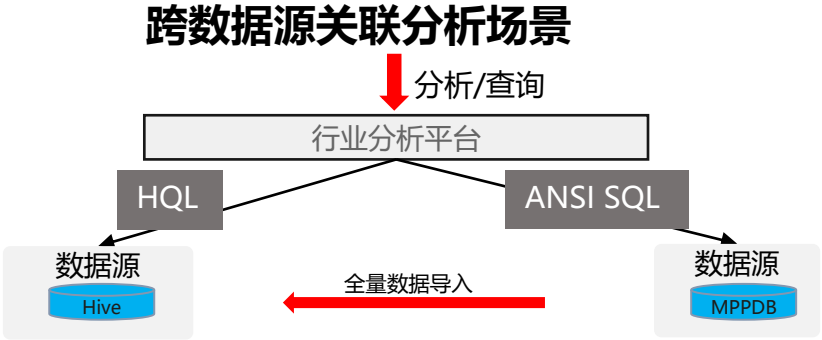
大数据分析面临的挑战



痛点1：两个烟囱，两份数据，管理复杂



痛点3：工作流人工处理，难以支撑T+0分析



痛点2：引擎接口不统一，编程模型复杂

三大需求：

- 批/交互式融合分析
- 跨数据源关联查询
- 跨域协同分析

openLooKeng-面向大数据的融合分析引擎

安平
警务大数据

政府
政务大数据 | 部委大数据

金融
金融数据湖

运营商
运营商大数据

大企业
企业数据湖

数据源

- 关系型数据
- 日志数据
- 外部数据
- 传感器(IoT)
- WEB
- 社交媒体
- 3rd party

入湖

数据使能

数据集成 | 数据开发 | 数据治理 | 虚拟数仓

计算引擎

查询引擎

批计算	流计算	融合分析	图计算	搜索
Hive, Spark	Flink	openLooKeng	GraphBase, GeoMesa, HBase	ElasticSearch

YARN

AI

机器学习 | 深度学习 | 推理引擎

数据管理

数据目录 — Catalog

数据安全 — Security

数据存储

数据存储 — **HDFS**
TXT | ORC | Parquet | Carbon

分布式存储
FS-HDFS | 对象 | 文件

管理

- 安全管理
- 租户管理
- 配置管理
- 性能管理
- 故障管理

鲲鹏服务器 | X86服务器 | 虚拟机 | 云主机

openLooKeng：统一高效的数据虚拟化融合分析引擎，让大数据变简单

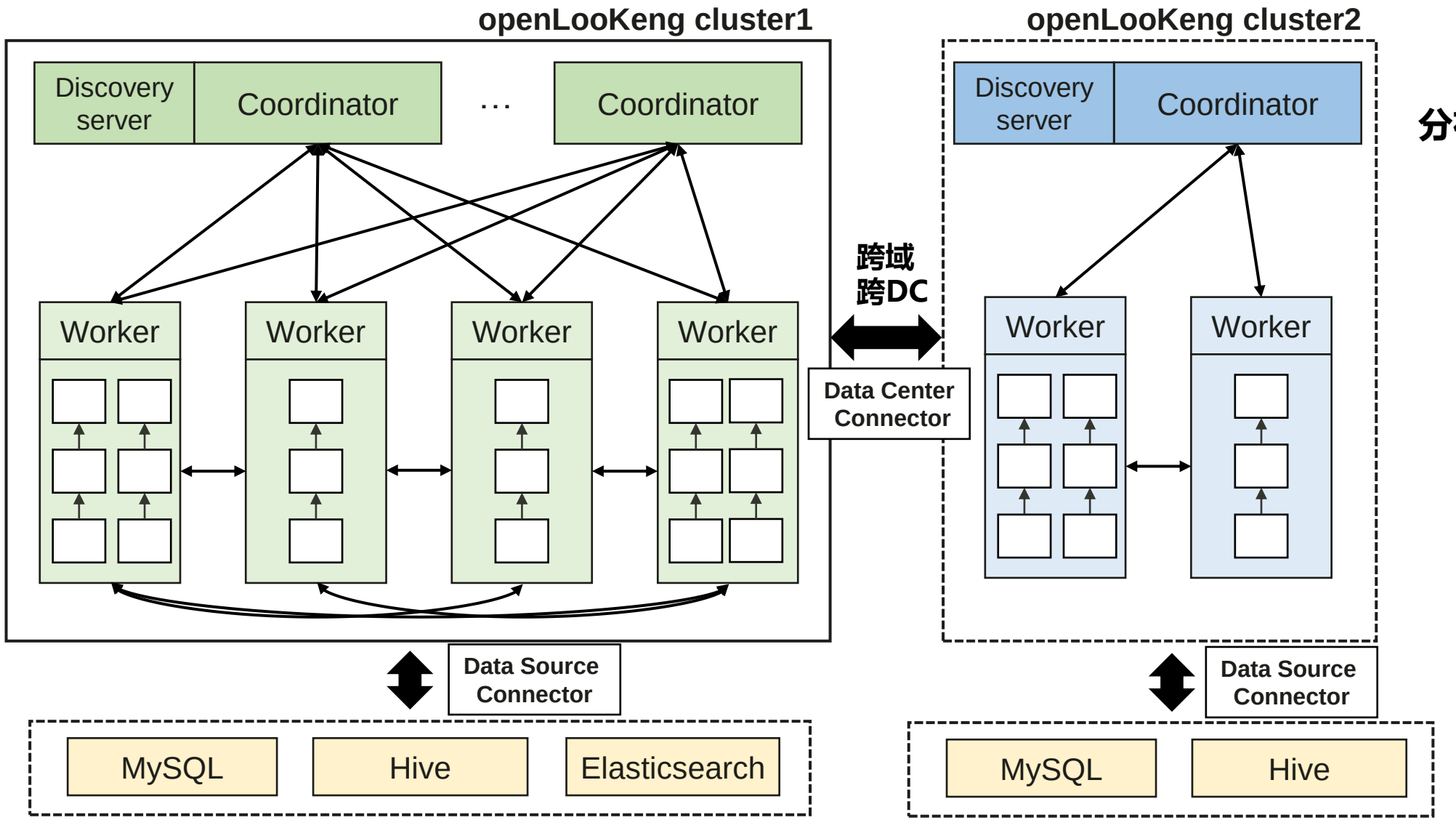
统一入口，化繁为简，单一引擎支持多场景

内核增强，高性能查询

跨源关联分析，数据消费零搬移

跨域协同计算，广域网的部署，局域网的体验

openLooKeng架构



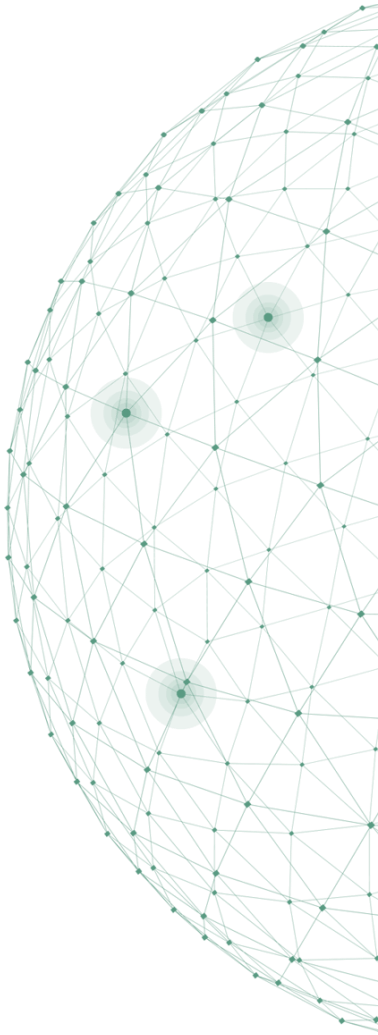
分布式处理系统，MPP架构

高可用性，无单点故障

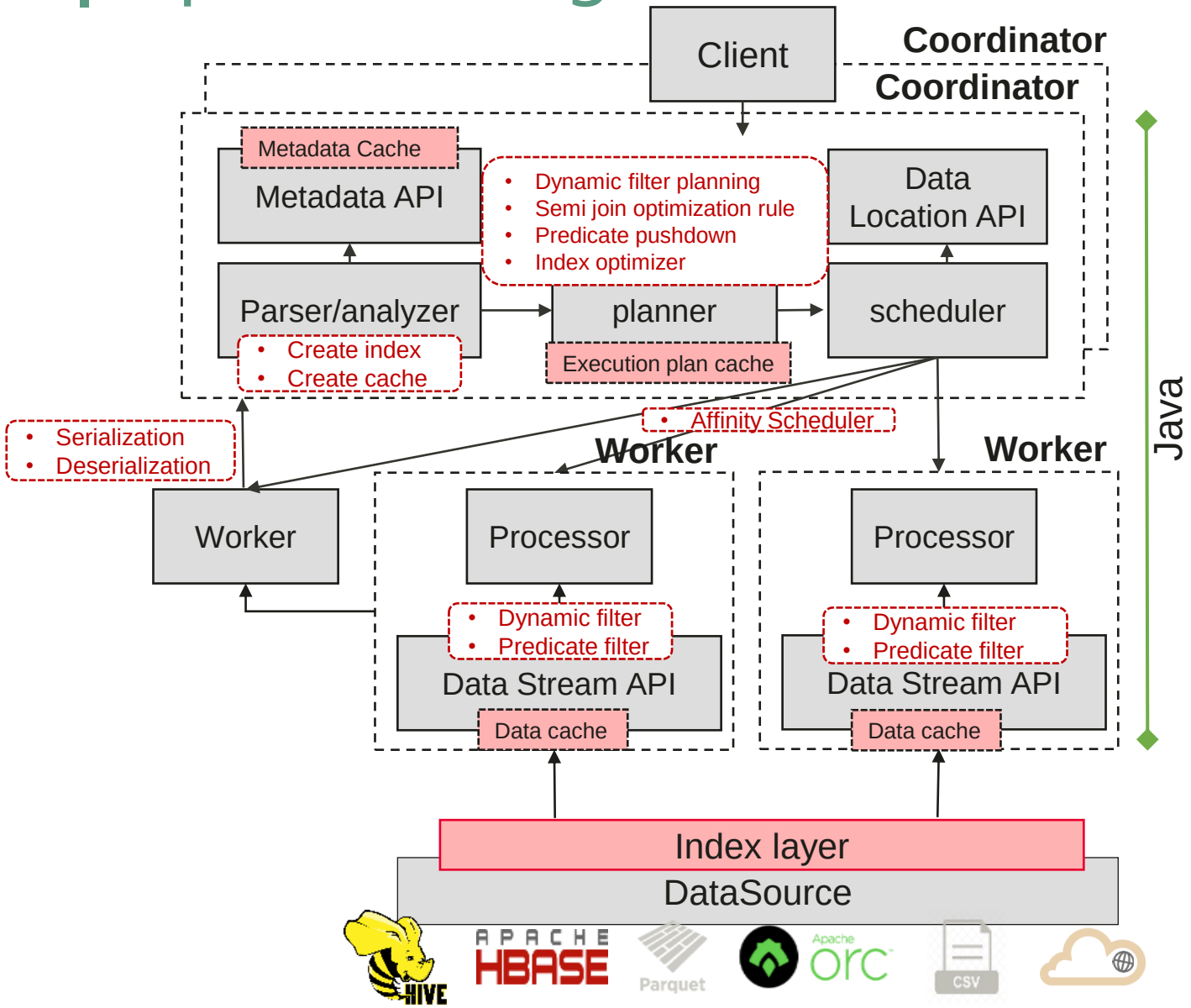
向量化列式处理引擎

基于内存的流水线处理

目录



openLooKeng交互式场景关键技术



- **数据源侧，更适应openLooKeng**
  - 分桶/分区
  - 小文件合并
  - 查询字段排序
- **引擎层，增强交互式查询能力**

  - 缓存加速：
  - 执行计划缓存
  - 元数据缓存
  - 增量列式缓存

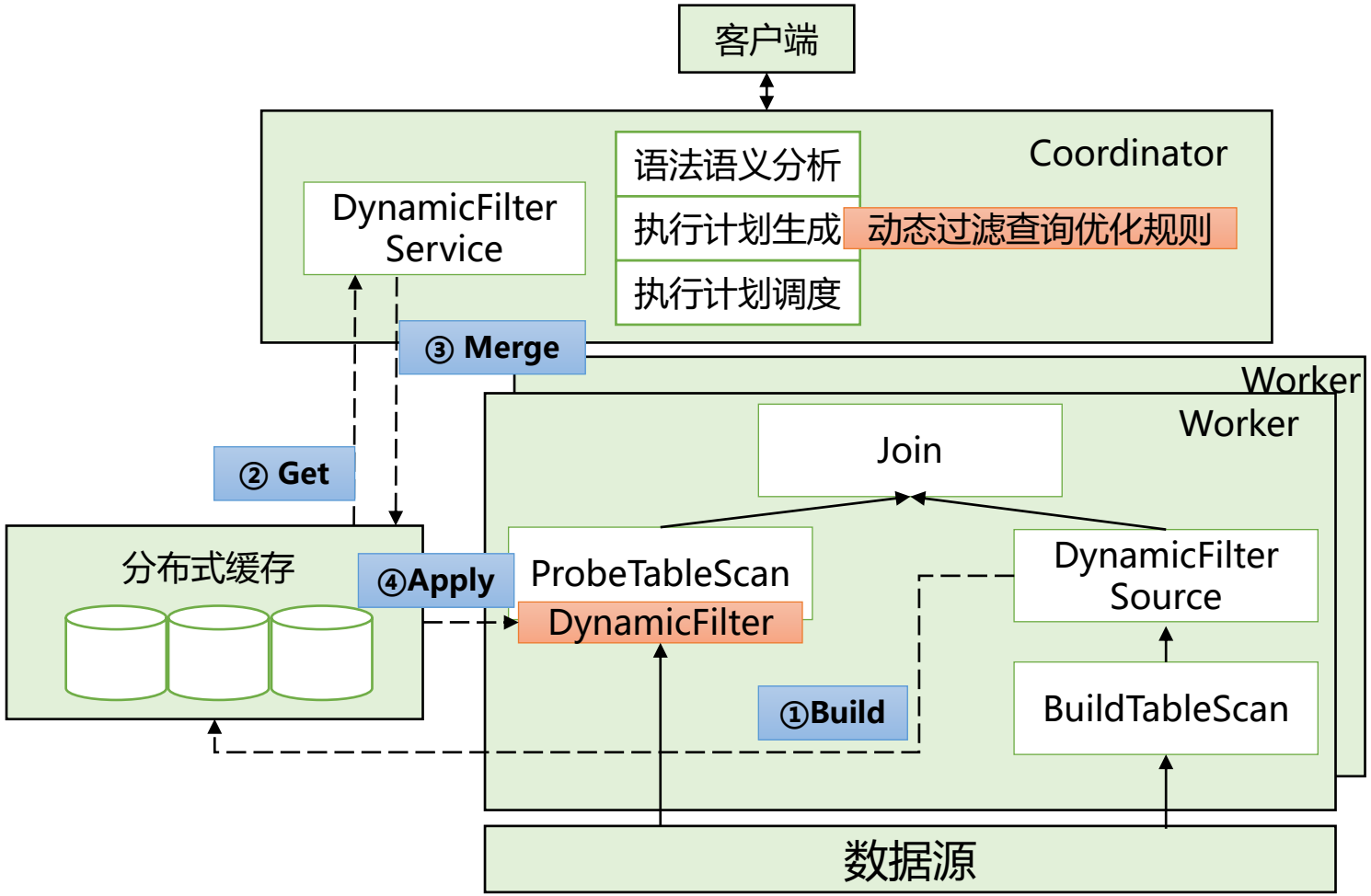
  - 优化器：
  - 谓词下推
  - **动态过滤**
  - RBO&CBO

  - 自适应调度器
- **额外层，加速交互式查询**
  - **Heuristic index layer**

    (bitmap/bloomfilter/min-max)
  - Data cache layer
  - 序列化&反序列化

Dynamic Filtering

依靠join条件以及build侧表读出的数据，运行时生成动态过滤条件（dynamic filters），应用到probe侧表的table scan阶段，从而减少参与join操作的数据量，有效地减少IO读取与网络传输

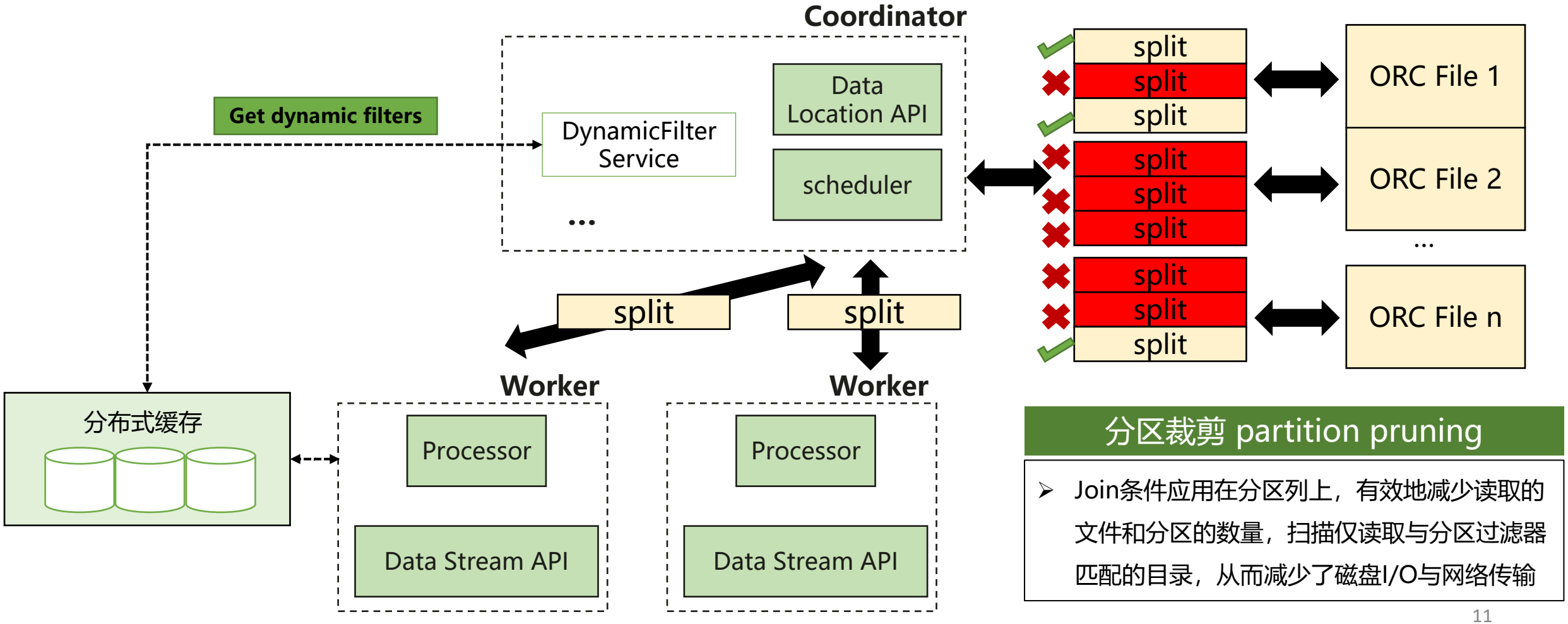


Dynamic Filtering

- 添加DynamicFIlterSource算子，搜集build侧数据
- 依赖分布式缓存进行DF的处理
- 适用于inner join & right join
- **适用于join选择率较高的场景**

Dynamic Filtering

/user/hive/warehouse/tpcds_bin_partitioned_orc_1000.db/store_sales
/**ss_sold_date_sk=2452638**/000997_0



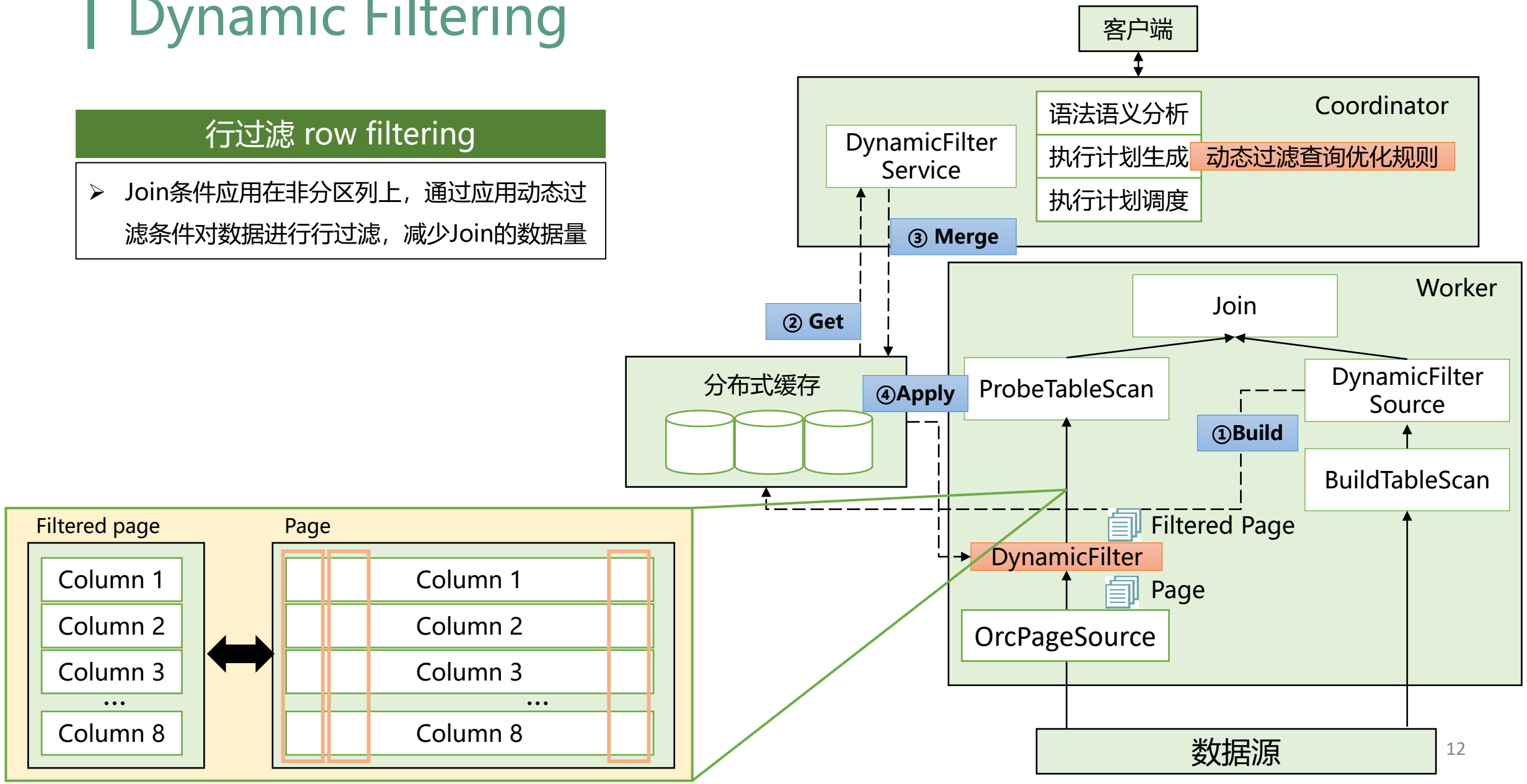
分区裁剪 partition pruning

- Join条件应用在分区列上，有效地减少读取的文件和分区的数量，扫描仅读取与分区过滤器匹配的目录，从而减少了磁盘I/O与网络传输

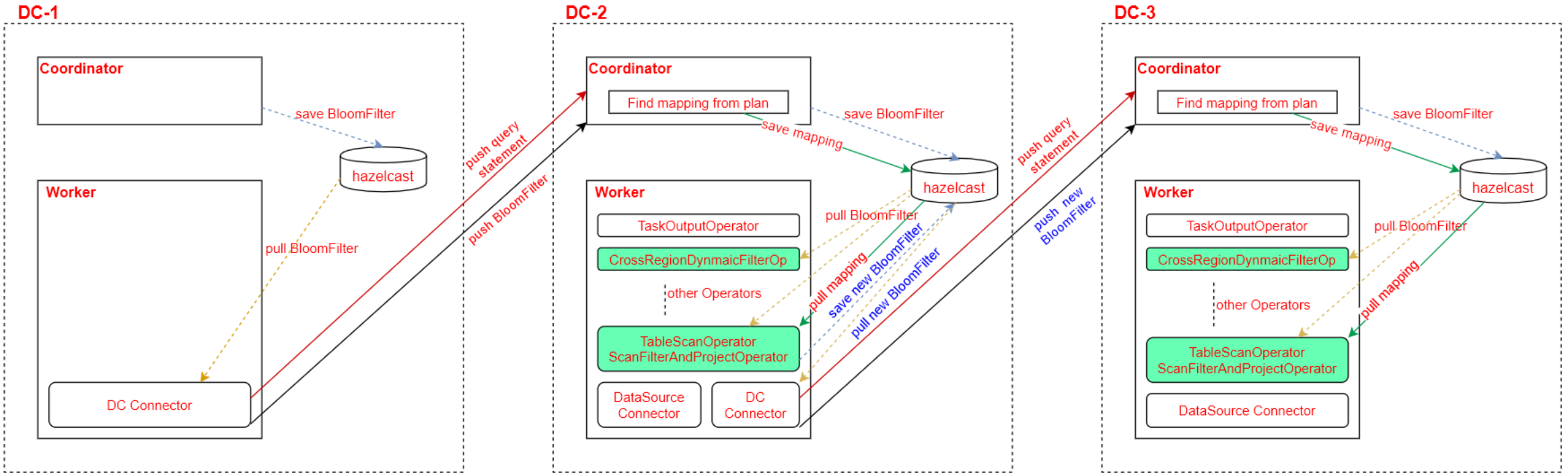
Dynamic Filtering

行过滤 row filtering

- Join条件应用在非分区列上，通过应用动态过滤条件对数据进行行过滤，减少Join的数据量



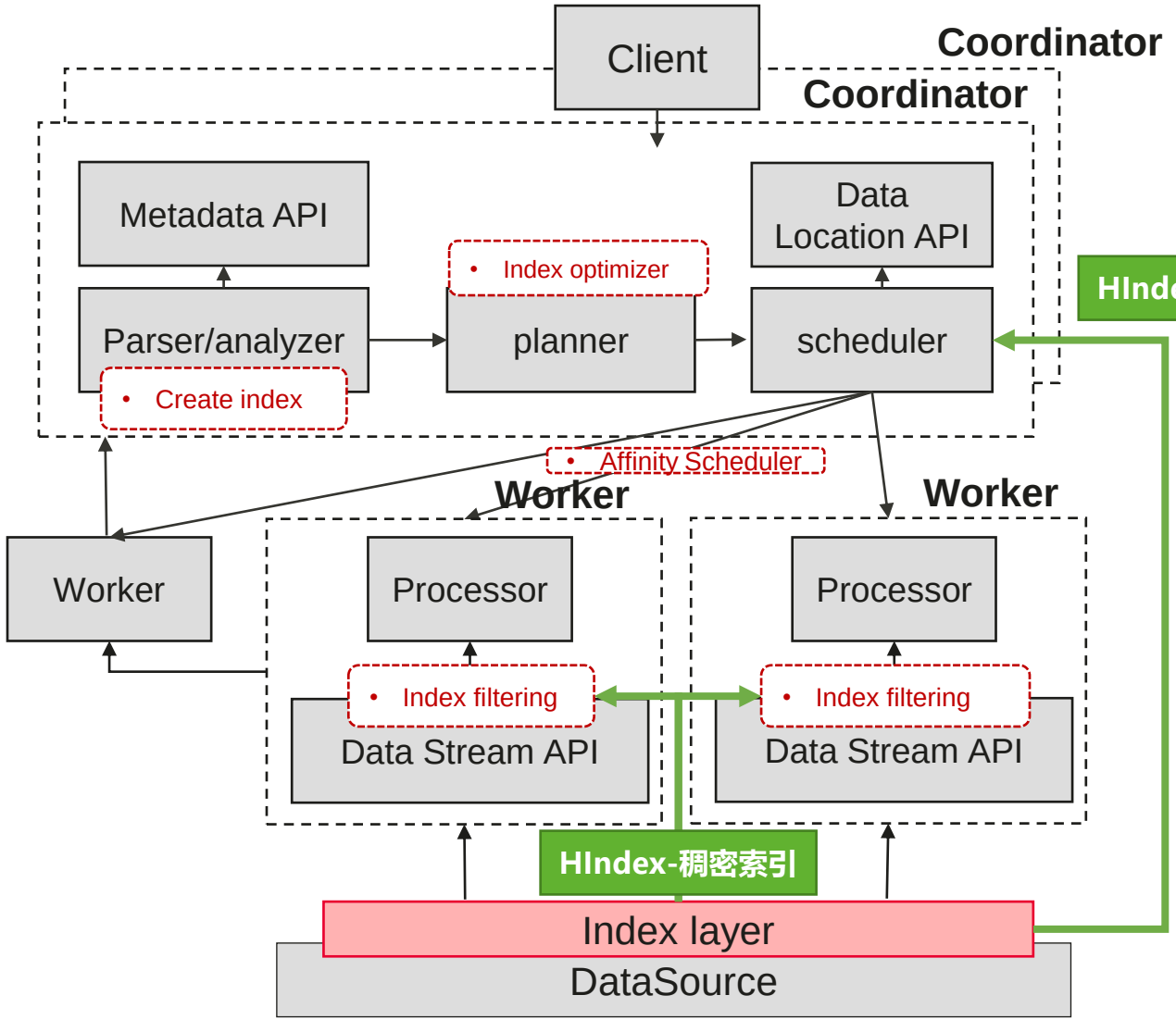
跨域Dynamic Filtering



DC-2 Coordinator： 1）将DC-1的BF filter以QueryId为Key存入到hazelcast；2）判断当前query是否存在跨域dynamic filter，存在，设置session中的cross-region-dynamic-filter；3）CN生产执行计划Plan，从Plan中Query的列名到Plan的outputSymbols的映射关系，存入hazelcast；4）判断Plan的TableScanNode是否存在DC table，如存在，则标记，可能存在继续下推BF filter的可能。

DC-2 Worker： 1) CrossRegionDynamicFilterOp从hazelcast中取出BF filter和outputSymbols，判断是否存在过滤列，存在则应用filter对Page进行过滤；2）TableScanOperator应用filter和步骤一类似；3）如果TableScanNode存在DC table，则生成新的BF filter并存入hazelcast，用于发送给下一级DC。

Heuristic index架构

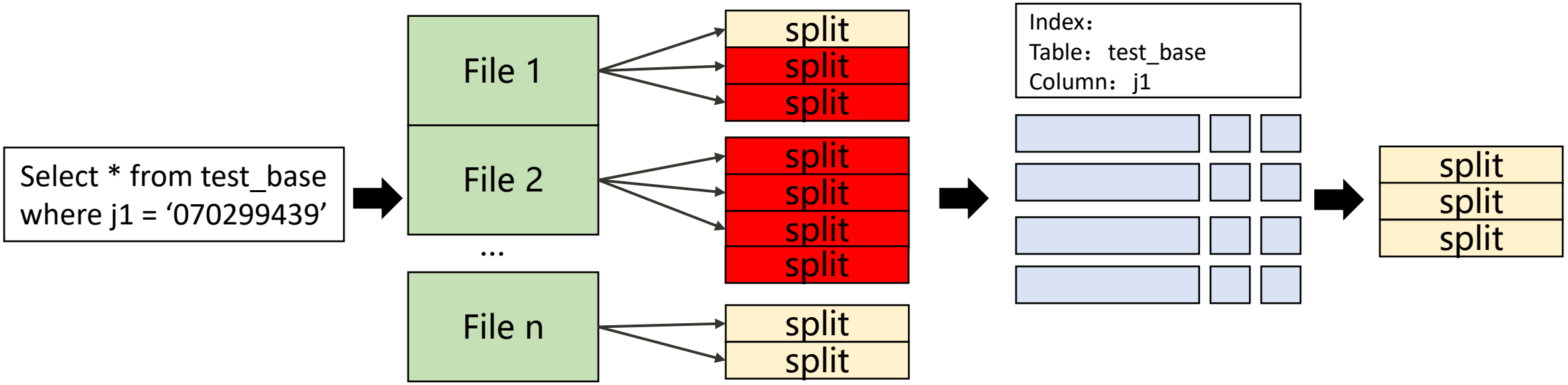


Heuristic index

- 提供统一的索引框架
- 支持多种索引结构
  - 稀疏索引：Bloomfilter、Min-Max
  - 稠密索引：Bitmap
- 任务调度阶段：
  - 裁剪Split,减少调度到Worker的任务数
  - 支持基于索引的亲和性调度
- 数据读取阶段：
  - 减少加载到计算侧内存的数据量

Heuristic index- Bloom filter索引

Bloom filter索引，确定每个split是否包含要搜索的值，并只对可能包含该值的split进行读操作

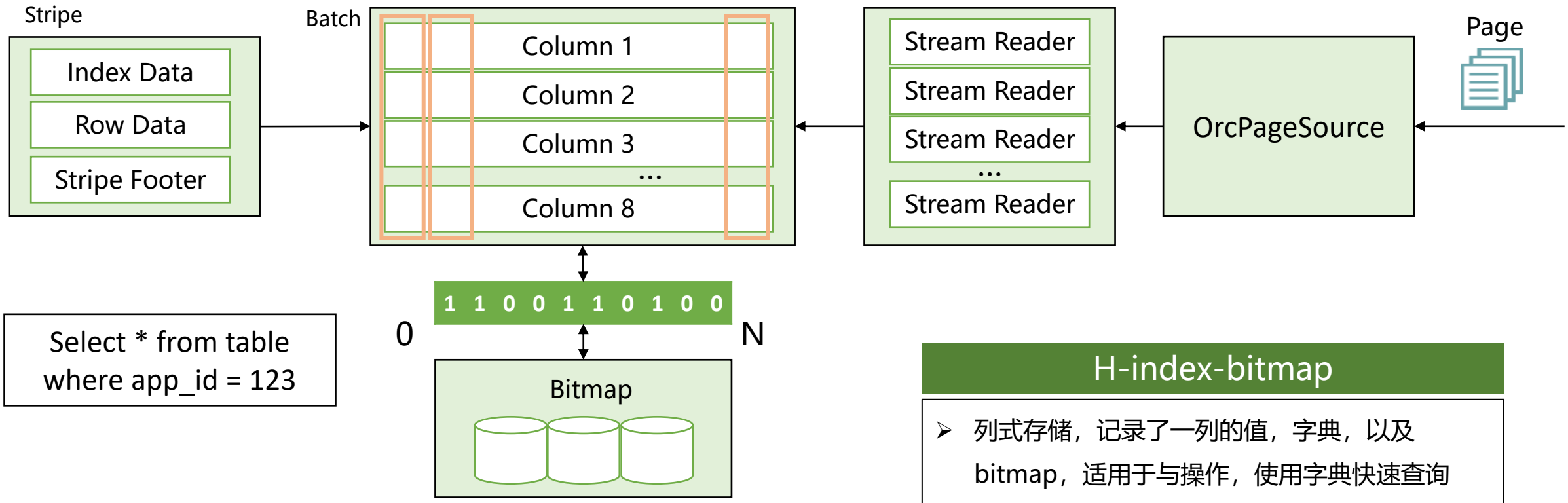


H-index-bloom filter

- 可以快速判断一个集合中有无某个值（只支持等于符号）
- 需要预先通过create index进行索引创建
- 通过在coordinator侧过滤，减少不必要的split生成与处理

Heuristic index – 位图索引

Bitmap索引，通过为谓词列保留外部位图索引，可以过滤掉与谓词不匹配的行

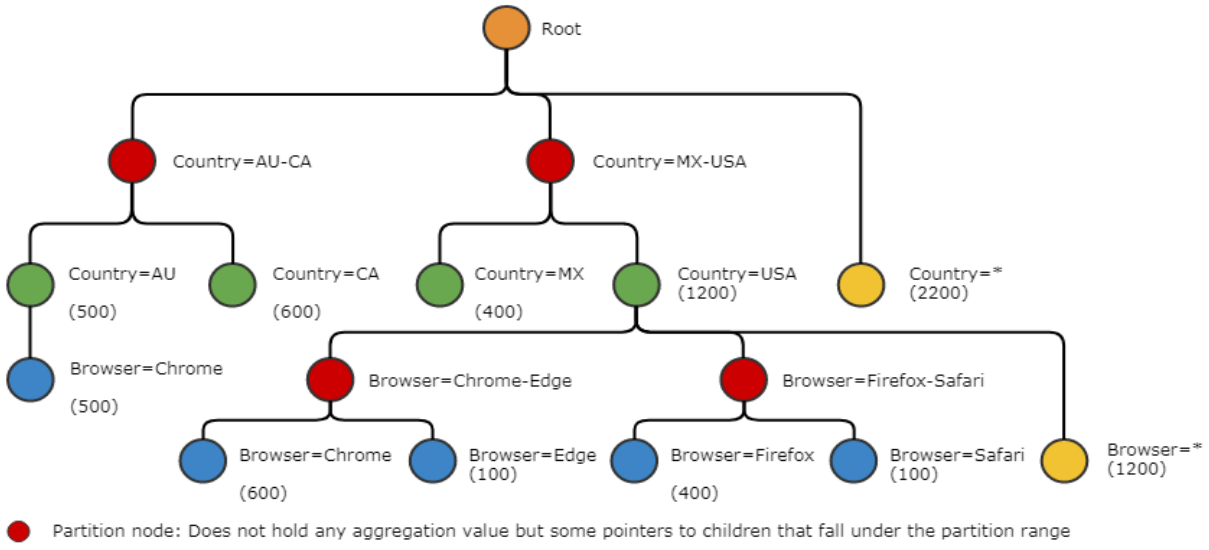


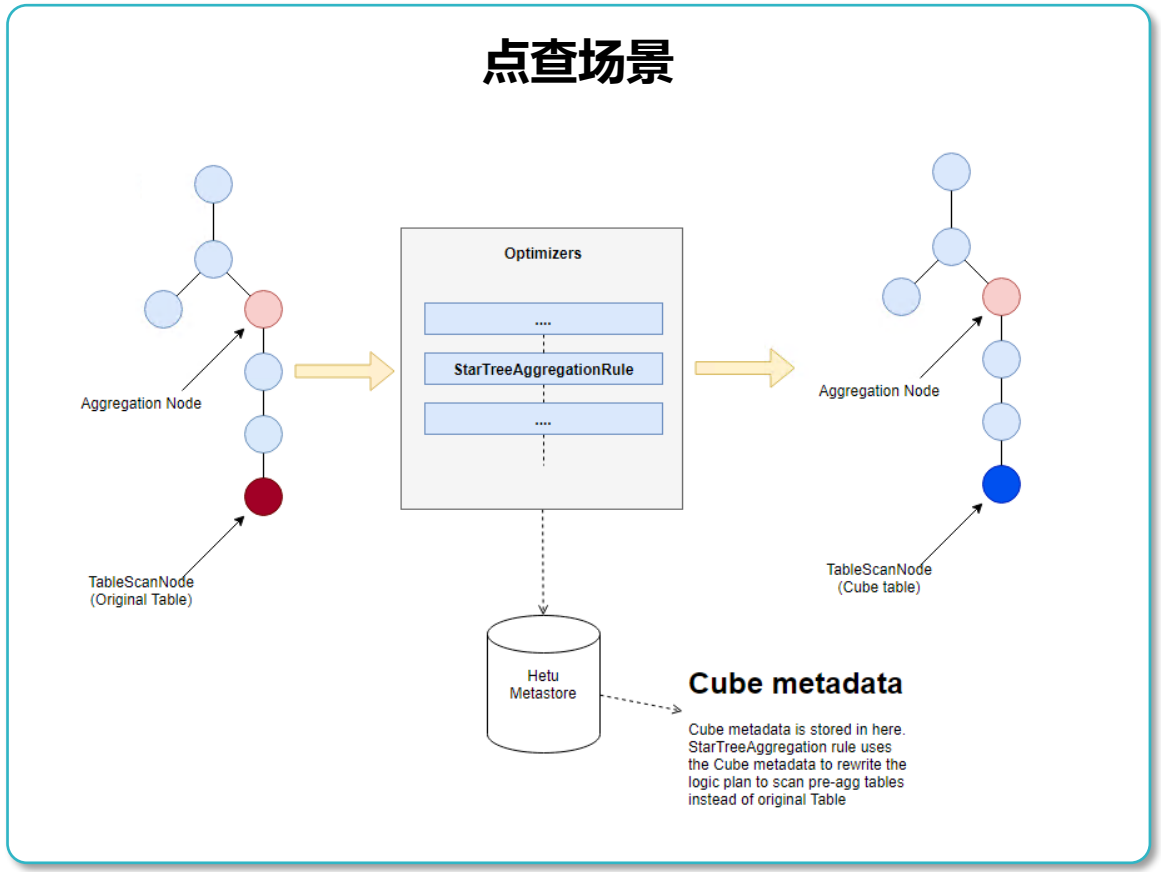
Select * from table
where app_id = 123

H-index-bitmap

- 列式存储，记录了一列的值，字典，以及bitmap，适用于与操作，使用字典快速查询
- 需要预先通过create index进行索引创建

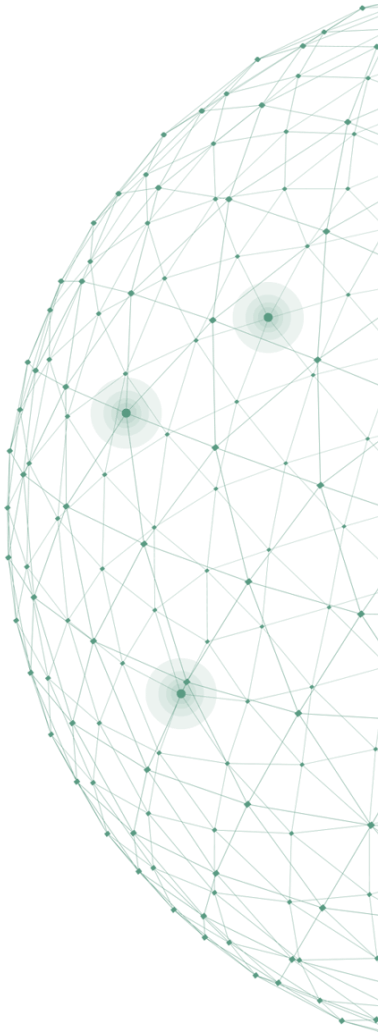
Heuristic index – StarTree index



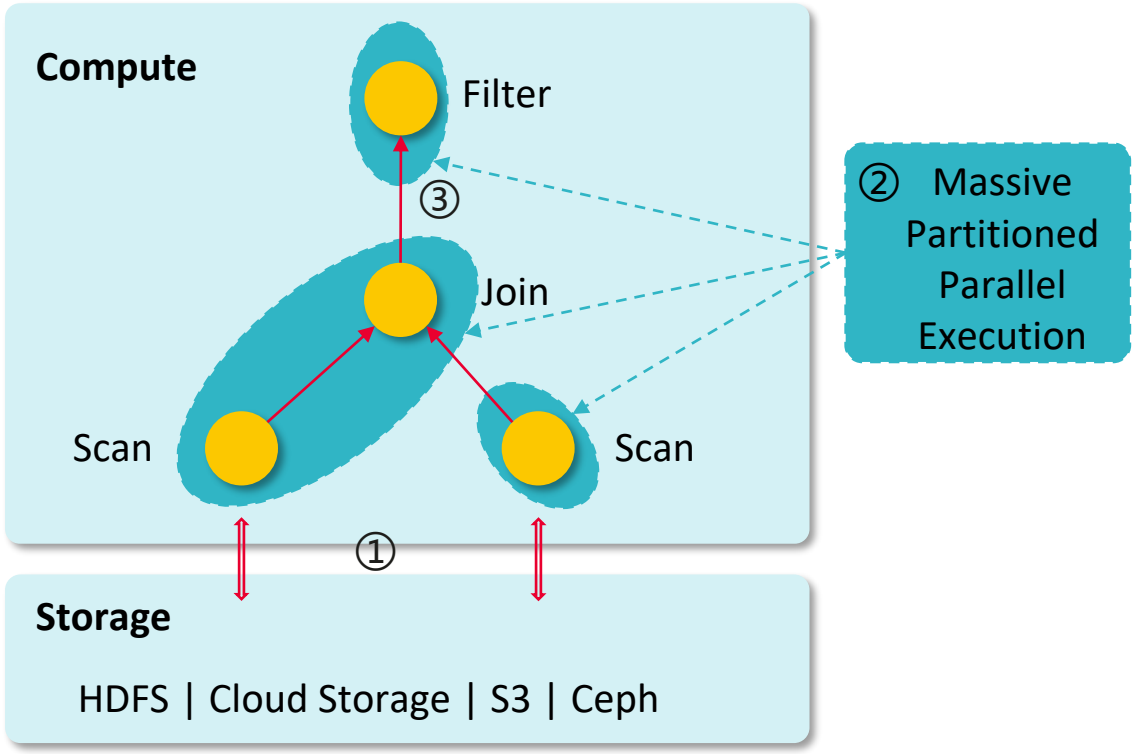


Star Tree技术：通过预聚合技术，降低高基维的查询的延迟，达到**ms级别**的查询要求

目录



从openLooKeng看大数据引擎性能的优化方向



① 数据加载：

- 存算协同：存算间缺乏有效协同

② 数据计算：

- 有效计算：算力消耗在无关控制流上
- 加速比：大规模集群的加速比小于0.5
- CPU/网络：受限数据处理逻辑无法充分利用

③ 数据交换：

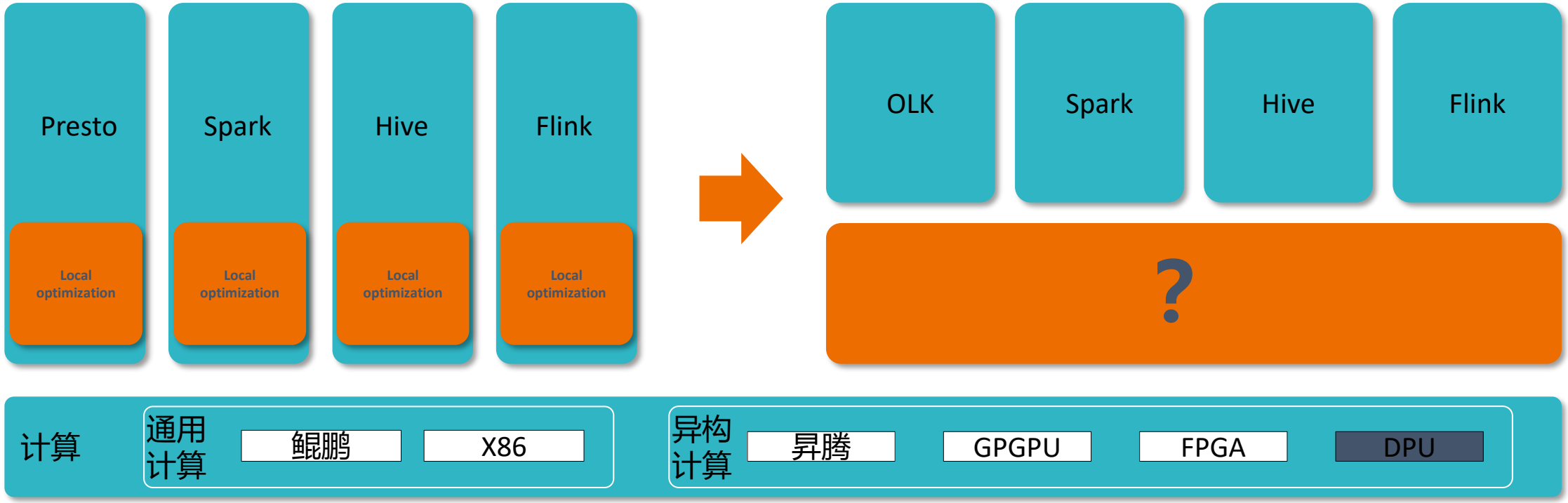
- 数据交换格式：不能根据网络特性进行自动调整
- 序列化：序列化、反序列化的性能损耗
- Zero-Copy：内存和操作系统缓冲区的数据拷贝

大数据引擎性能优化方向集中在Optimizer和Runtime两个领域

构筑进一步构建更好大数据生态?

烟囱式优化

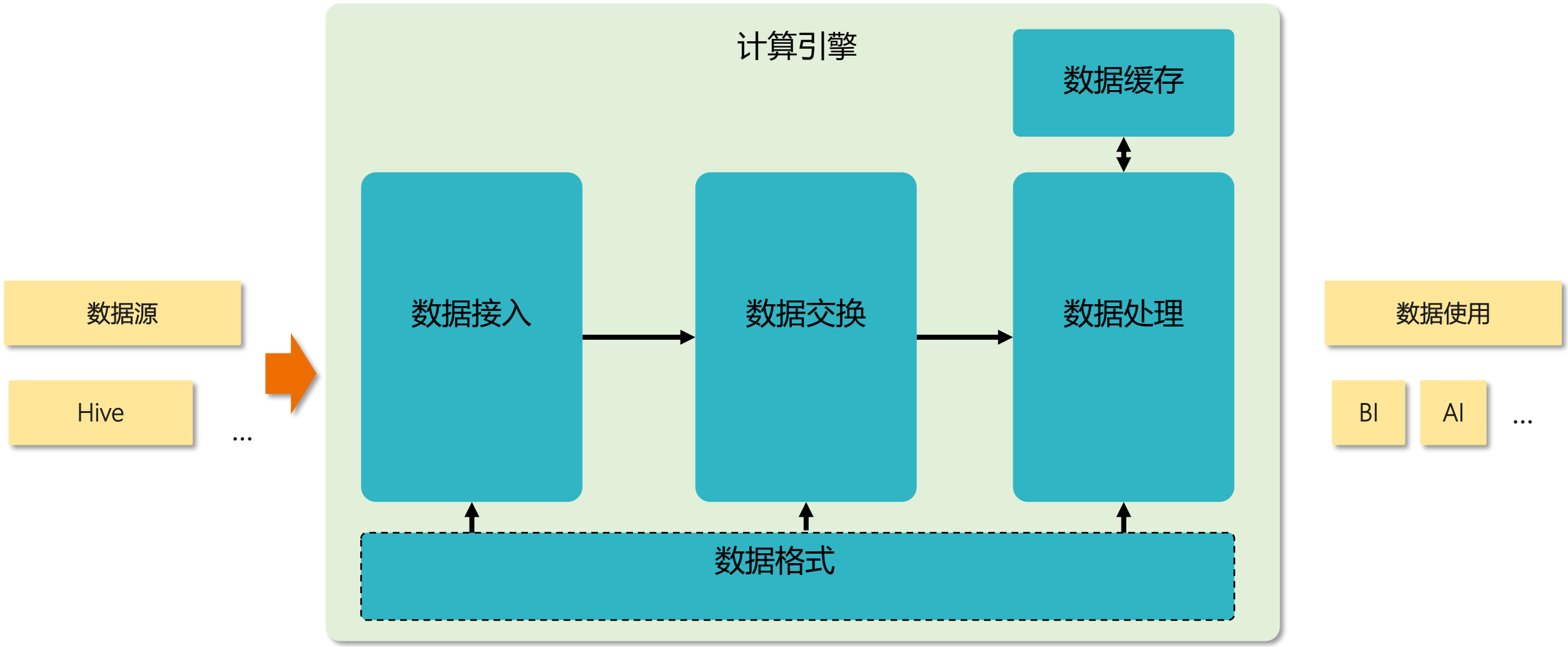
统一Runtime底座



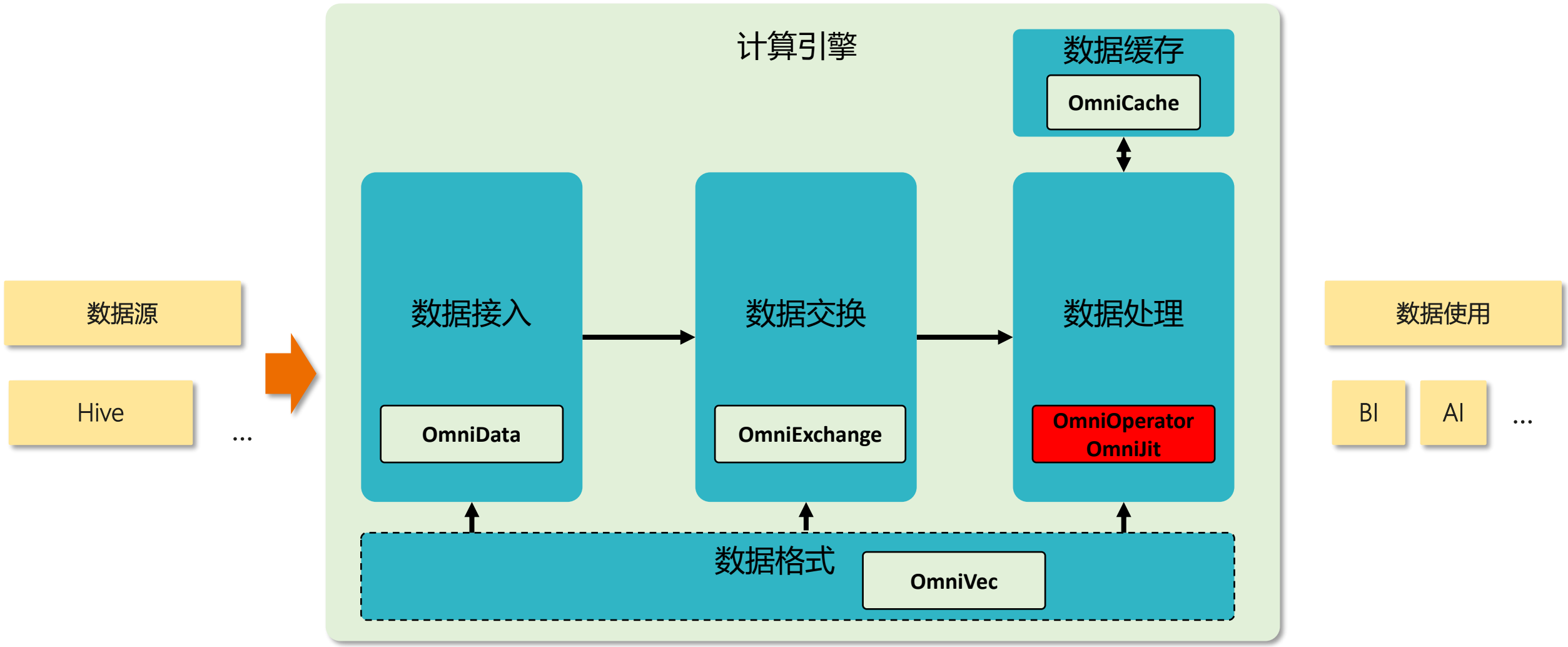
各个引擎各自垂直优化
实现native runtime算子
使用codegen对算子进行优化

一个底座支撑不同引擎，减少重复优化工作
充分挖掘通用、异构算力

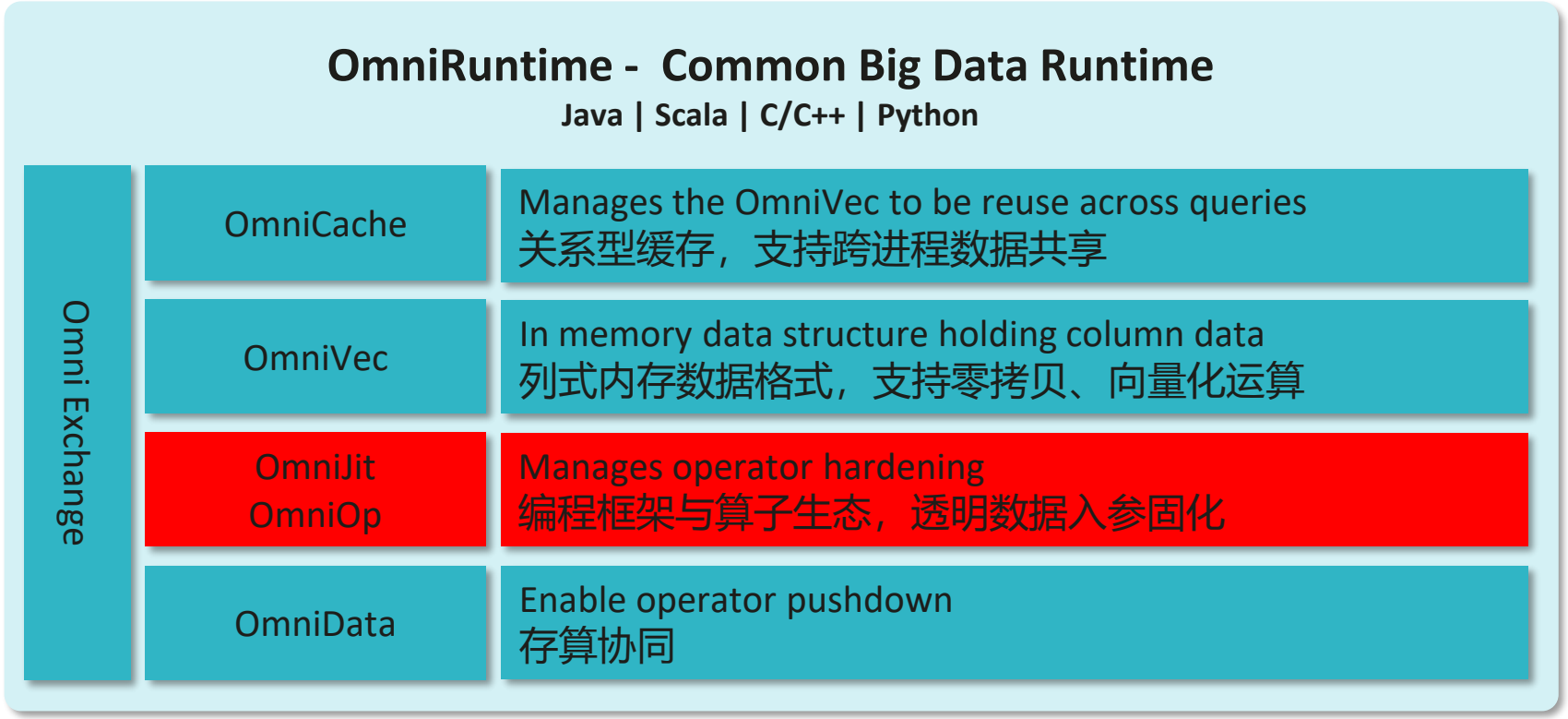
现有大数据引擎结构



OmniRuntime



OmniRuntime - 大数据的Runtime底座



多语言支持实现大数据生态完整支撑

OmniJit实现鲲鹏硬件垂直优化

Huawei JDK优化实现生态协同

数据加载：

- OmniData 存算协同，数据过滤
- OmniCache 数据重用

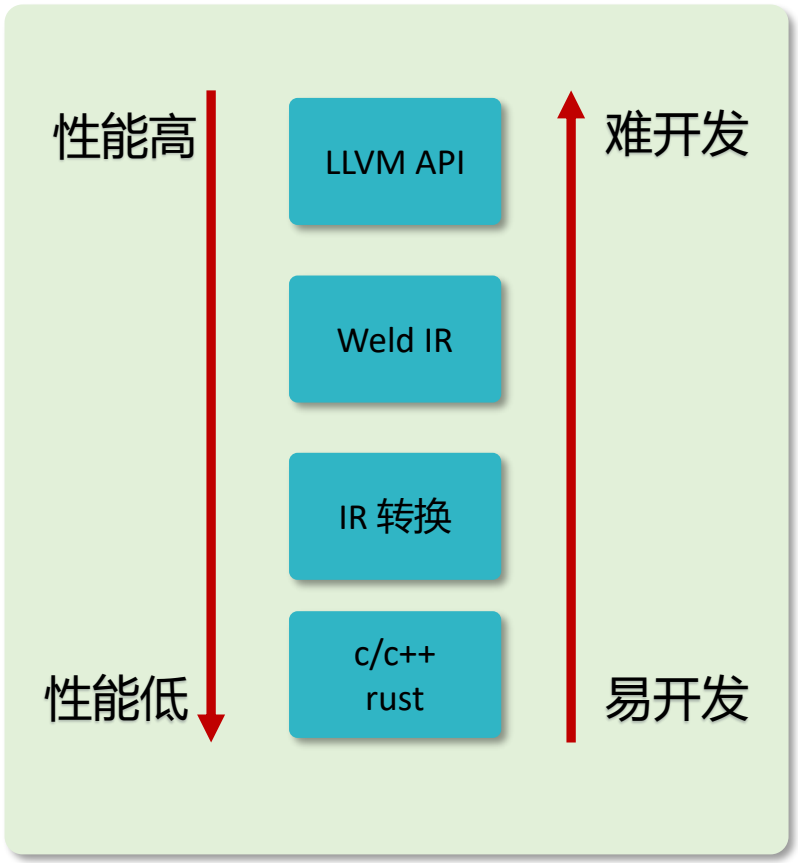
数据计算：

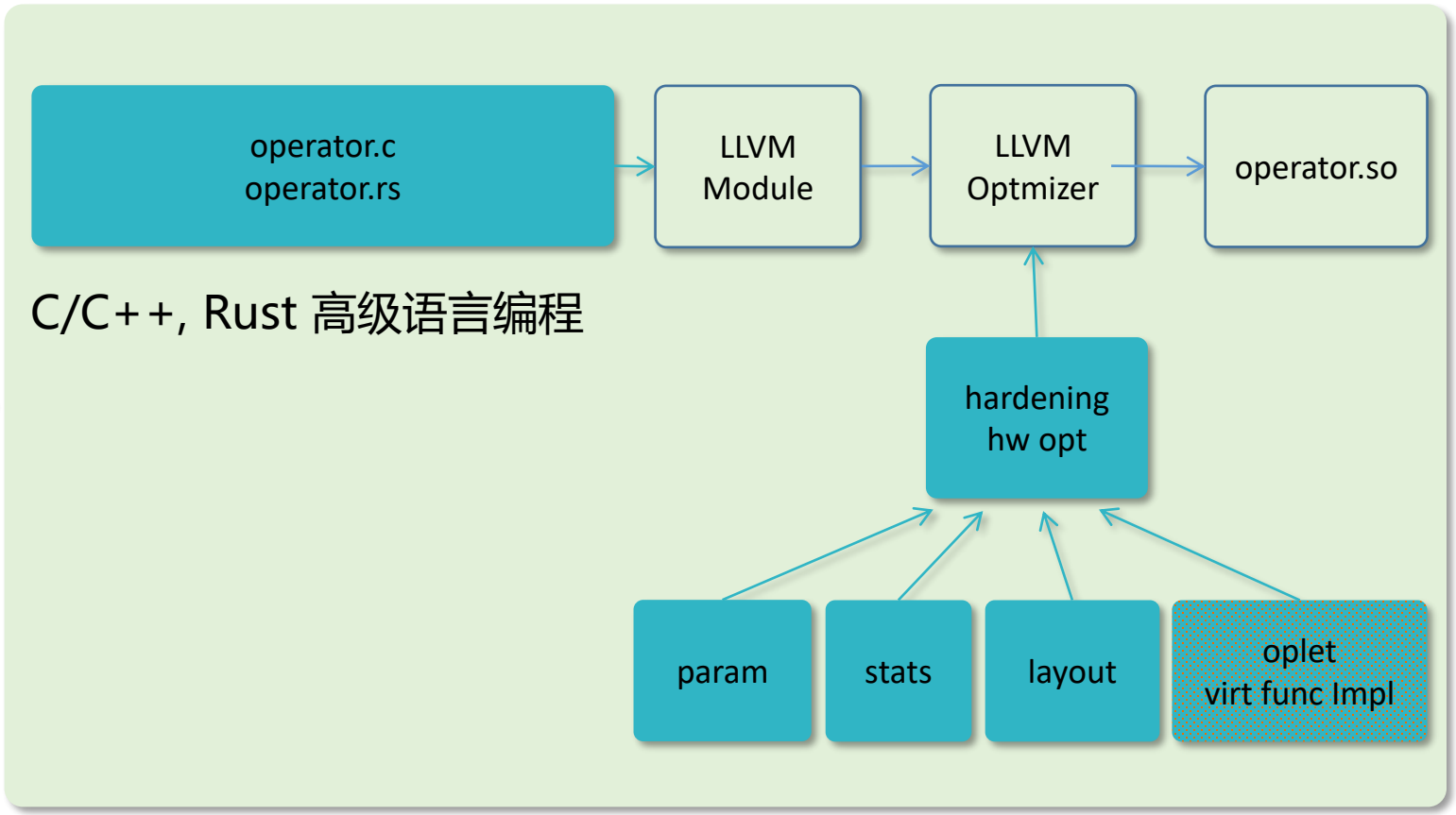
- OmniOp Native算子
- OmniJit 数据驱动控制流优化，ARM垂直优化
- OmniVec 列式内存数据格式，向量化运算

数据交换：

- OmniExchange TCP/RDMA兼容
- 高速传输，插件式Shuffle框架
- OmniVec 堆外内存，真正零拷贝

OmniRuntime关键技术 – OmniJit: Data Driven Specialization

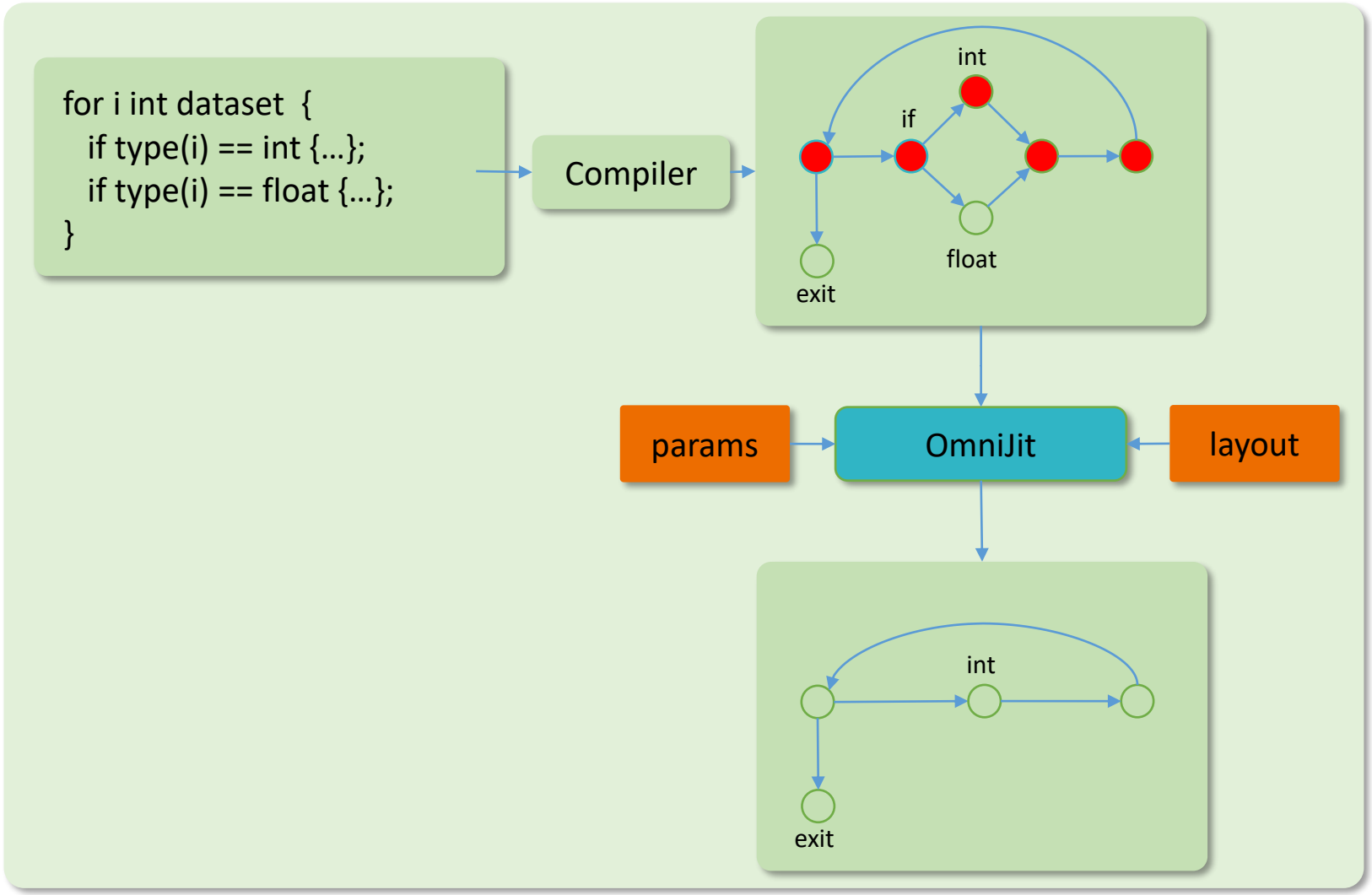


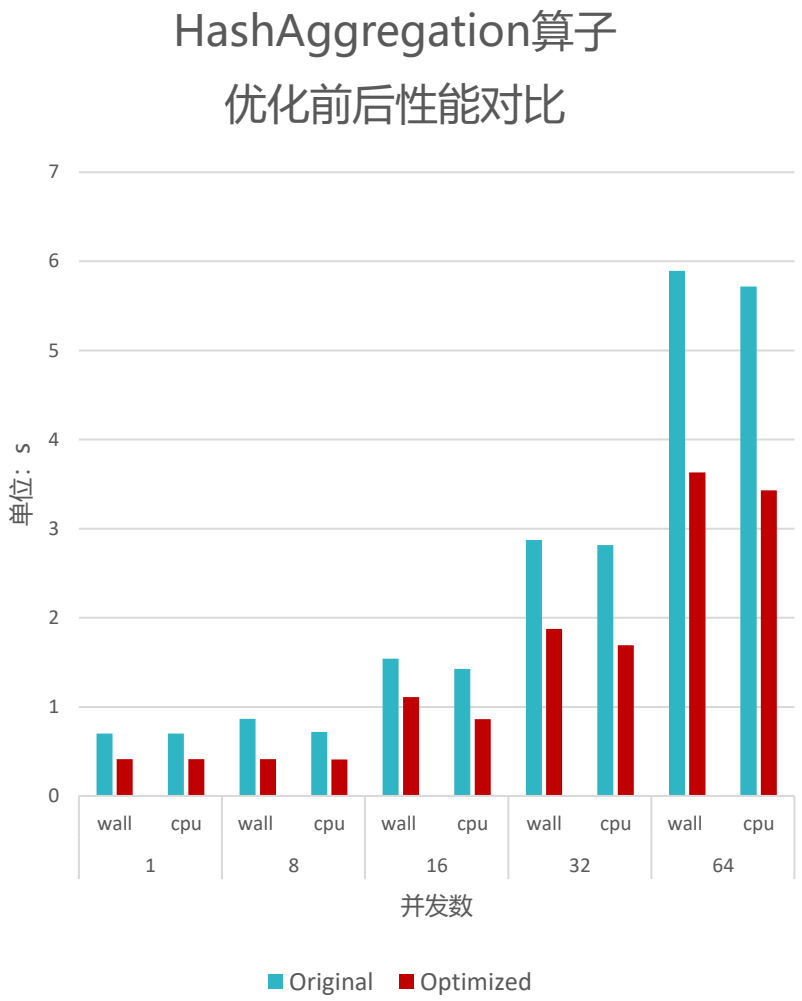


“高性能”与“易开发”不可兼得

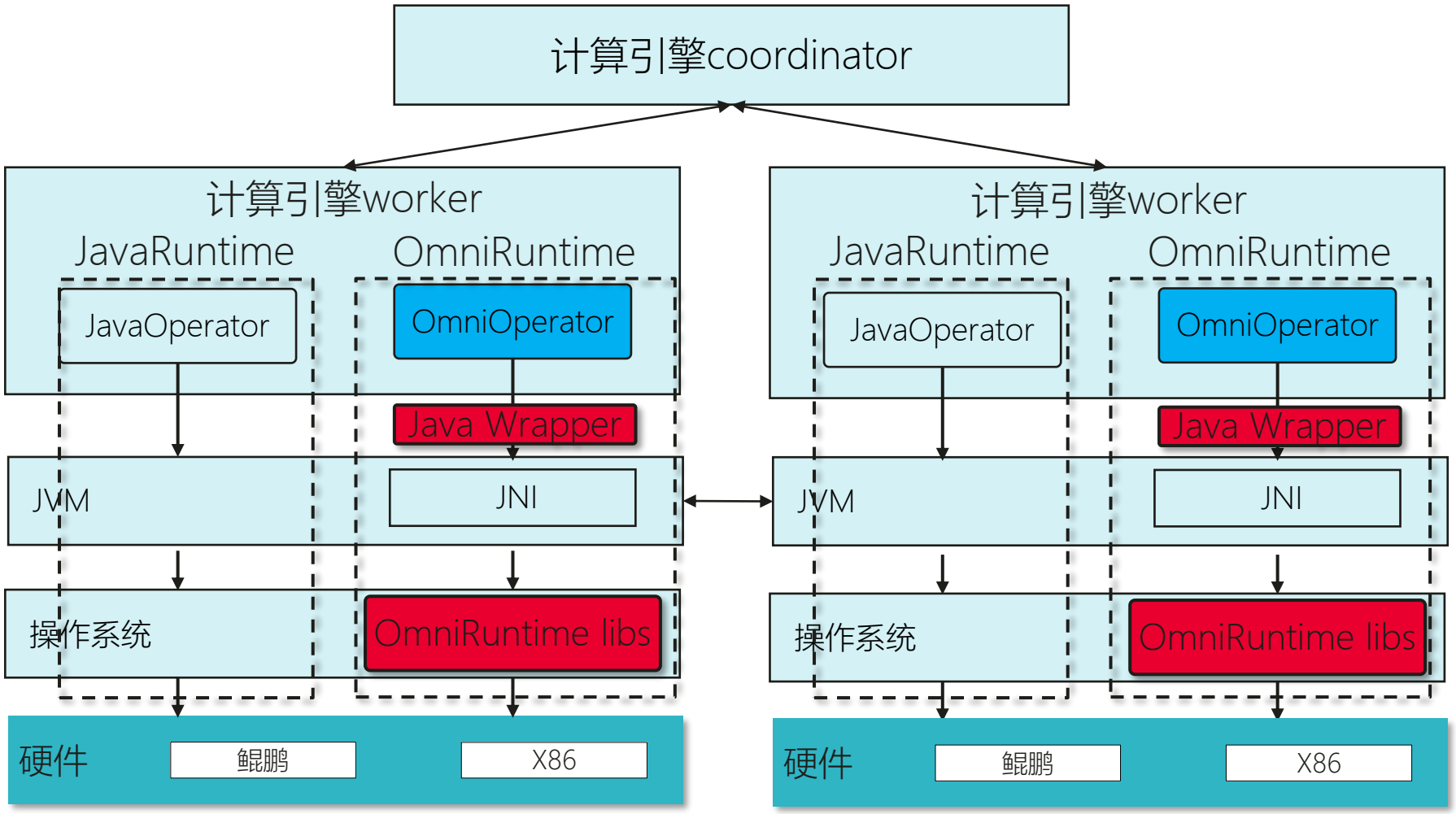
“高性能”与“易开发”兼得

OmniRuntime关键技术 – OmniJit: Data Driven Specialization





OmniRuntime大数据引擎部署图



OmniRuntime模块

引擎适配OmniRuntime模块

引擎与其他模块（解耦部分）

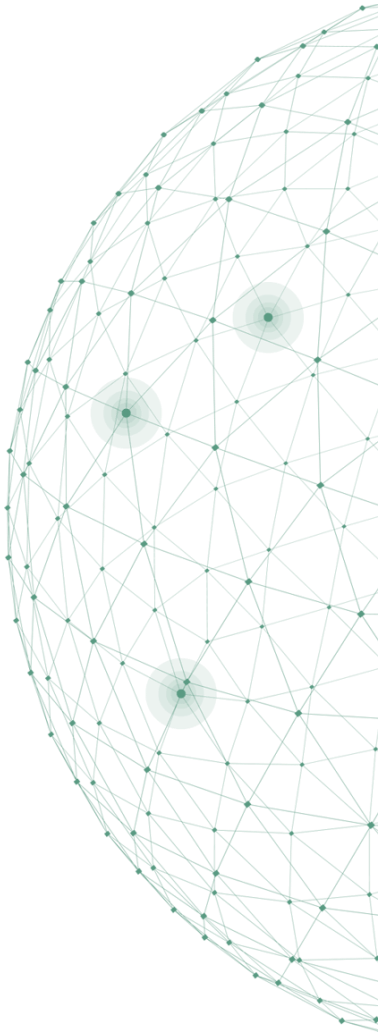
① OmniRuntime提供.so文件，jar文件，提供C++以及Java接口调用API

② 操作系统需要提供对LLVM12支持，否则需要更新操作系统

③ Java版本需要对JNI支持

④ 对于硬件解耦，支持X86与ARM，操作系统LLVM支持屏蔽底层硬件差异

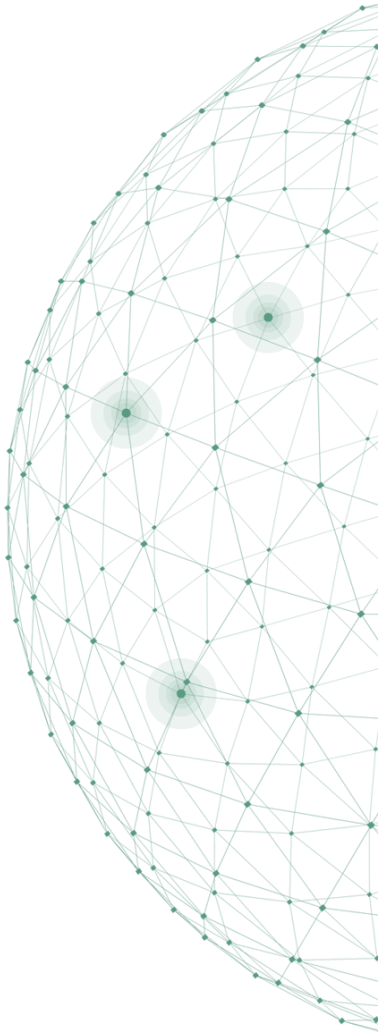
目录



总结

- openLooKeng愿景是让大数据更简单
  - 统一SQL入口，数据免搬迁
  - 高性能的交互式查询能力
  - 多源异构数据源融合分析
  - 跨域跨DC融合分析
- Looking back
  - openLooKeng从如下三个方面进行交互式查询优化：
  - 让数据源更好的适配引擎（分区/小文件合并）
  - 引擎自身优化（动态过滤/RBO/CBO/谓词下推/缓存）
  - 添加额外层加速（Heuristic index layer/Cache layer/序列化/反序列化）
- Looking forward
  - OmniRuntime构建大数据分析底座

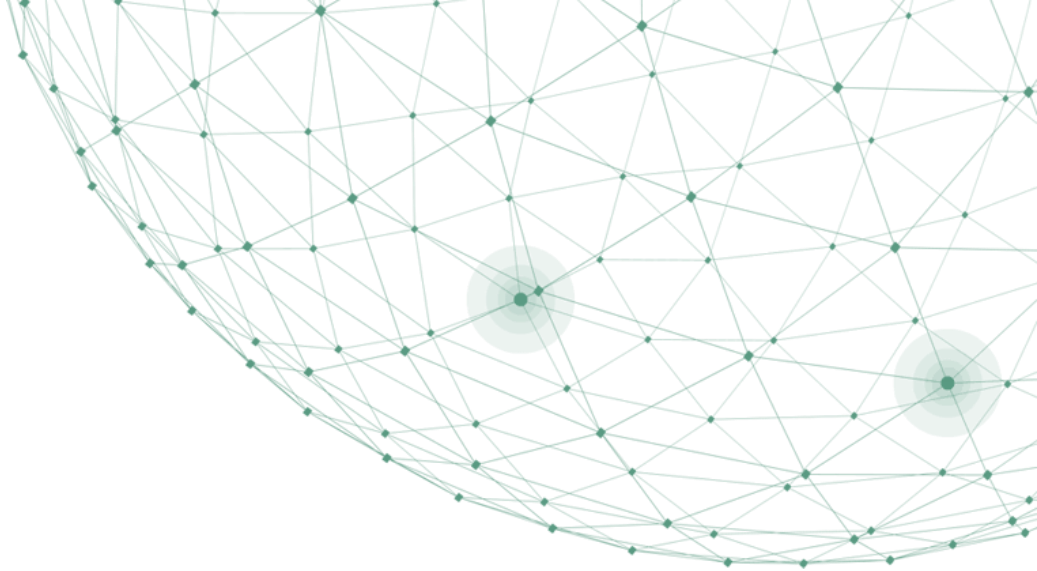
Thank you



openLooKeng, Big Data Simplified

openLooKeng微信公众号

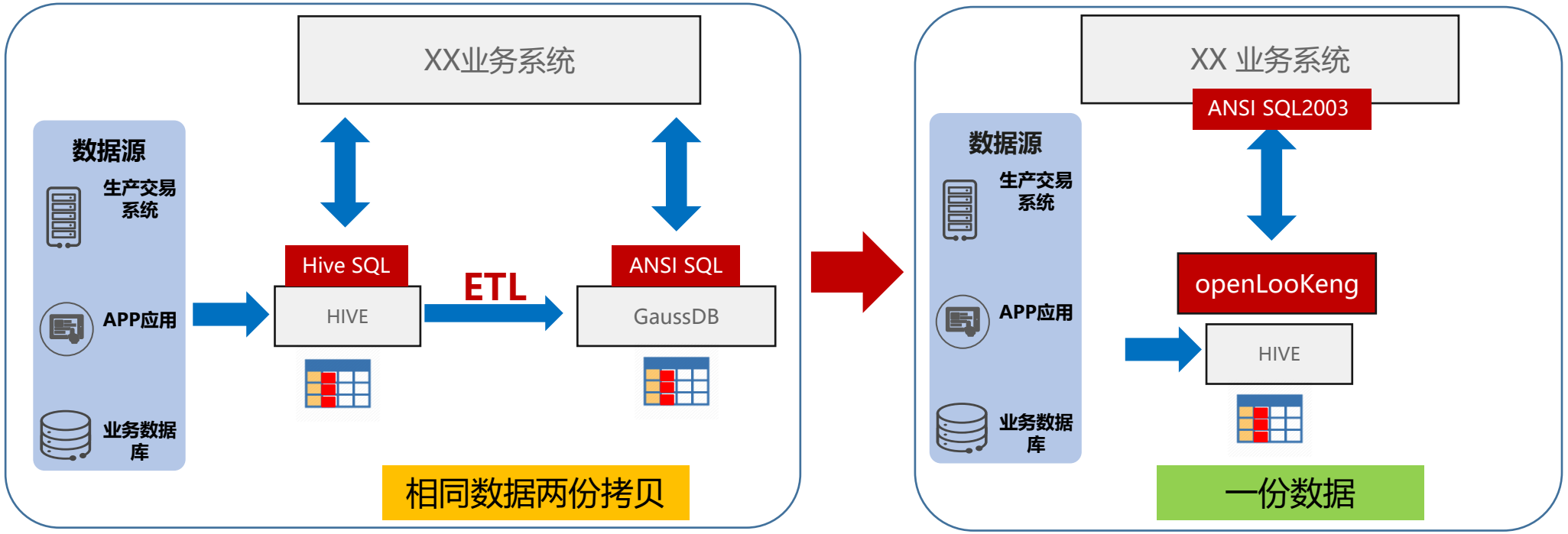
openLooKeng微信小助手



Backup

统一SQL访问，数据免搬迁：为XX客户业务减少ETL

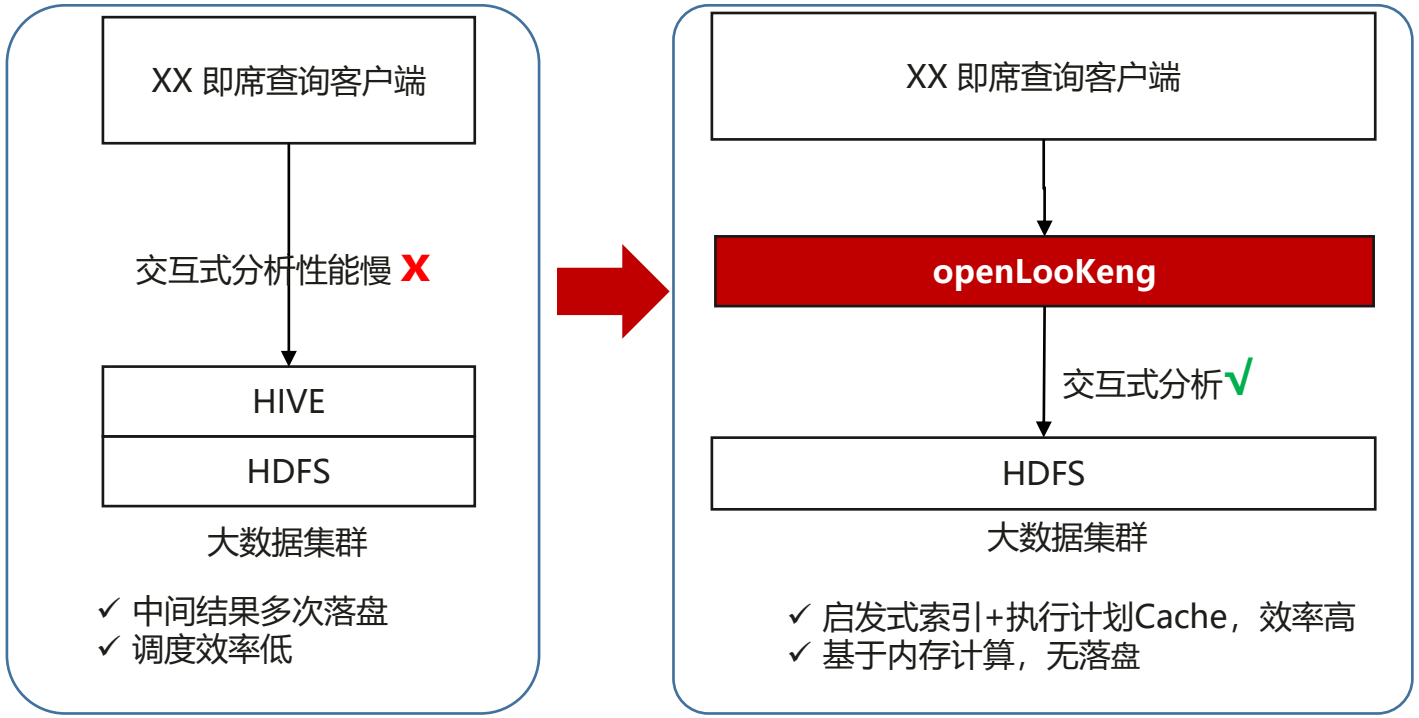
业务场景/需求	当前业务痛点	openLooKeng价值
业务对查询性能有较高要求，同时避免数据重复搬迁，减少复杂的ETL过程	HIVE SQL兼容性和交互式分析性能无法满足业务系统的要求，导致繁琐的ETL，数据重复拷贝多份	openLooKeng使用HIVE数据仓库的同一份数据，减少ETL，提升数据存储和分析效率

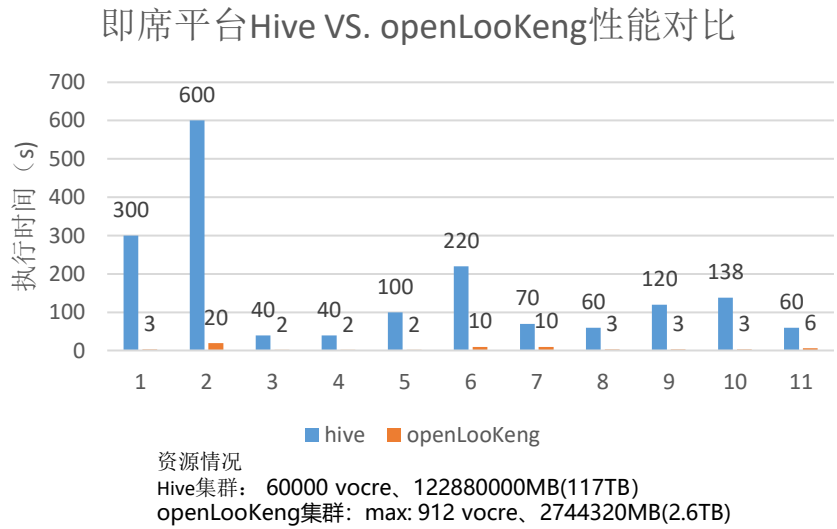


【效果】为XX客户的即席分析业务提供高性能的即席查询能力，同时避免复杂的ETL过程，减少数据重复搬迁

高性能的交互式查询：为XX客户即席服务提供秒级查询能力

业务场景/需求	当前业务痛点	openLooKeng价值
为数据分析人员提供方便快捷的即席查询能力，性能要求较高，秒级查询	通过Hive引擎来构建即席查询任务，查询时间太长（5分钟~2小时）性能不满足客户需求	openLooKeng提供秒级查询能力，结果准确率高，增强了用户对海量数据的分析能力



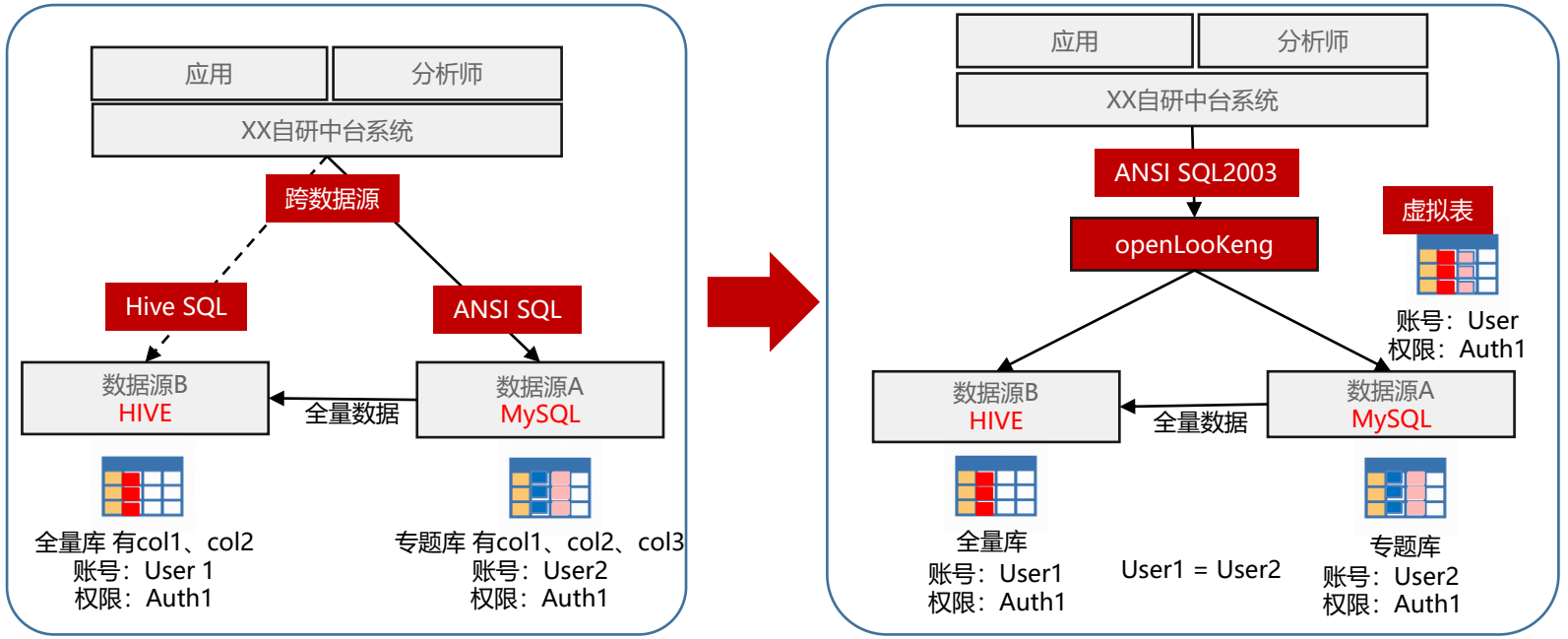


资源情况
Hive集群： 60000 vocre、122880000MB(117TB)
openLooKeng集群：max: 912 vocre、2744320MB(2.6TB)

【效果】XX即席查询平台采用openLooKeng后：不影响当前业务，实现查询加速，查询响应提升至秒级，平均性能提升2~10+倍，用户查询效率大大提升

多源异构数据融合分析：为XX客户跨源业务提供融合分析能力

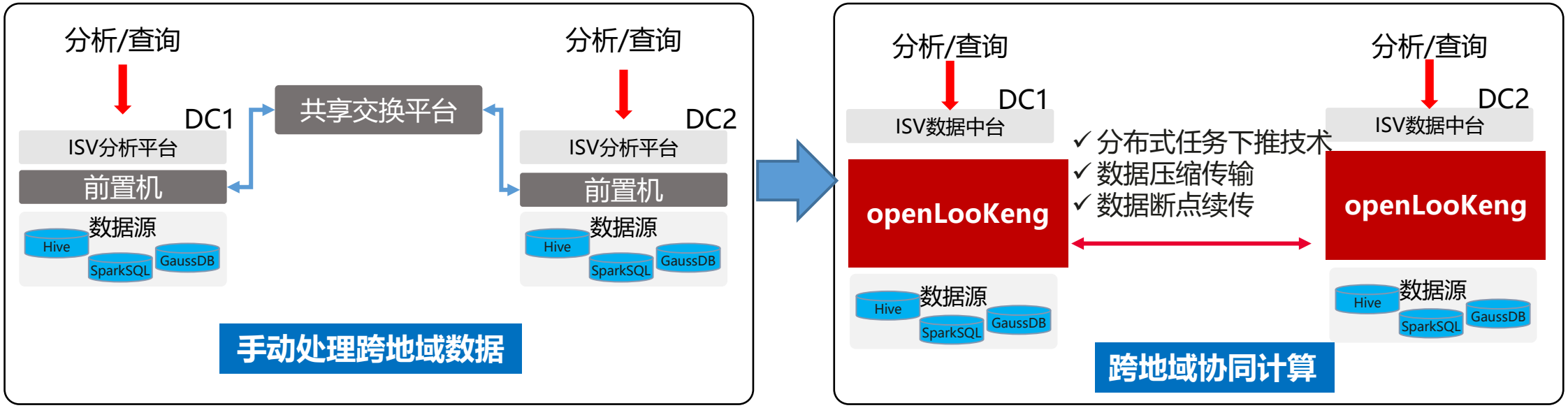
业务场景/需求	当前业务痛点	openLooKeng价值
需要查询不同数据源（Hive、Mysql等）进行关联碰撞，从而挖掘数据价值信息	Hive全量库和Mysql专题表的差异，导致应用系统开发、使用不便，数据探索效率低	屏蔽全量库/专题表的差异，简化应用系统开发/使用，提升数据探索效率，助力数据消费



【效果】openLooKeng为XX客户简化业务开发逻辑，提高对异构数据源的消费能力，增强对数据潜在价值的挖掘能力

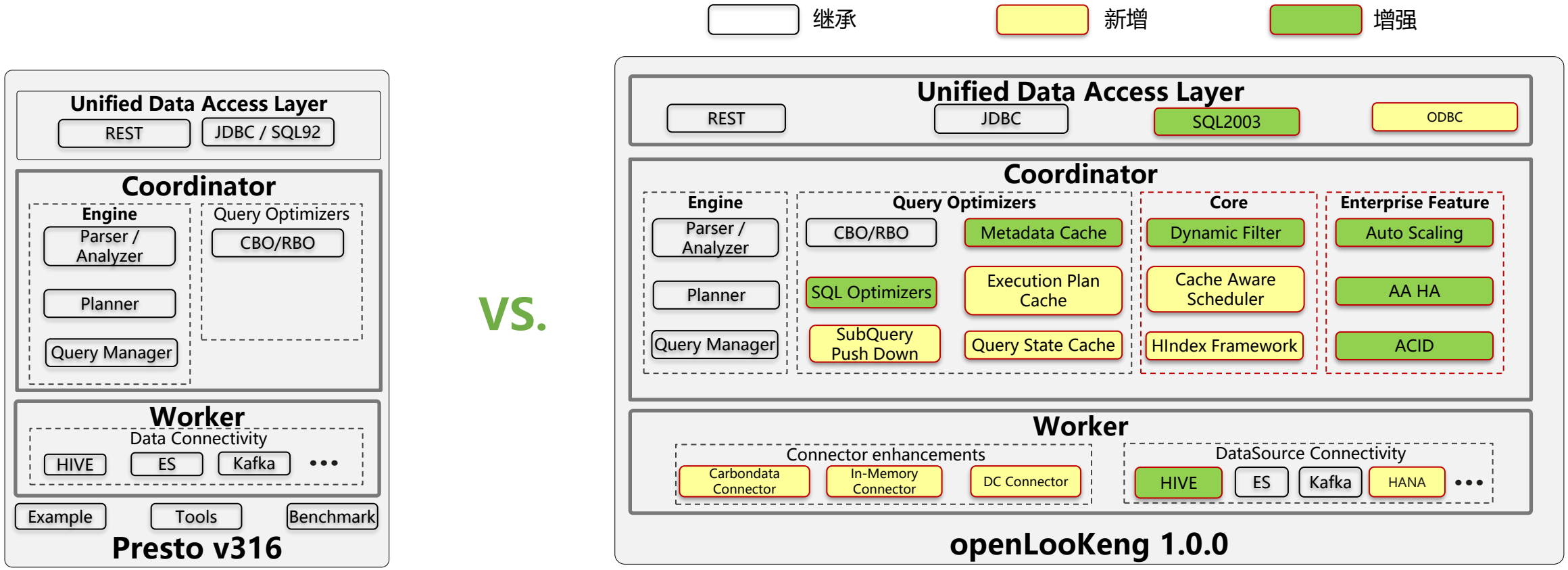
跨域跨DC融合分析：为XX客户提供跨地域协同分析能力

业务场景/需求	当前业务痛点	openLooKeng价值
打通跨省市两级数据中心数据访问，跨地市数据协查无需出差到当地处理	手动数据批量抽取数据到省中心，效率低，数据分析耗时长	openLooKeng跨地域协同分析能力强，可实现跨地域数据分析性能从天级缩短到分钟级



【效果】为XX客户提供跨地域协同分析能力，实现从天级到分钟级跨越，实现无需出差即可实现跨地市业务分析

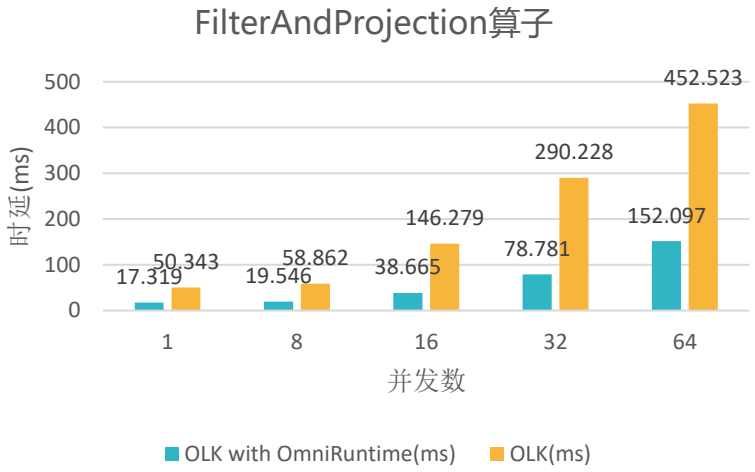
openLooKeng vs presto

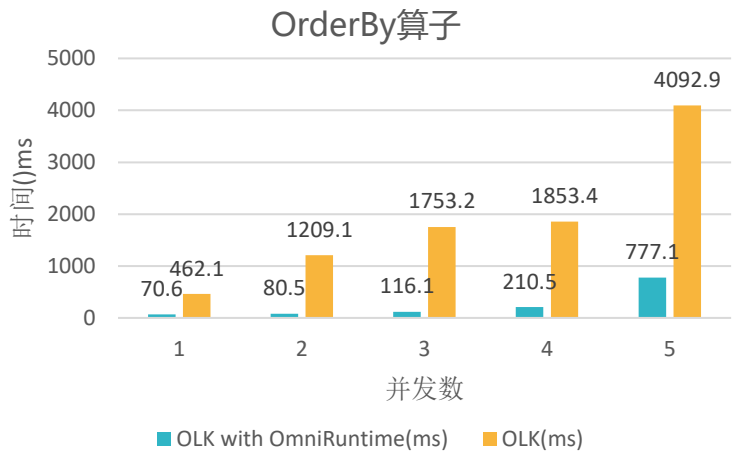


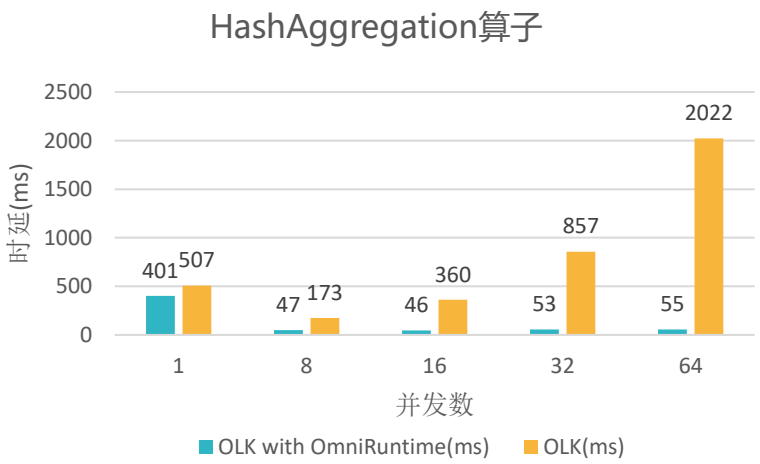
openLooKeng VS Presto 差异化竞争力

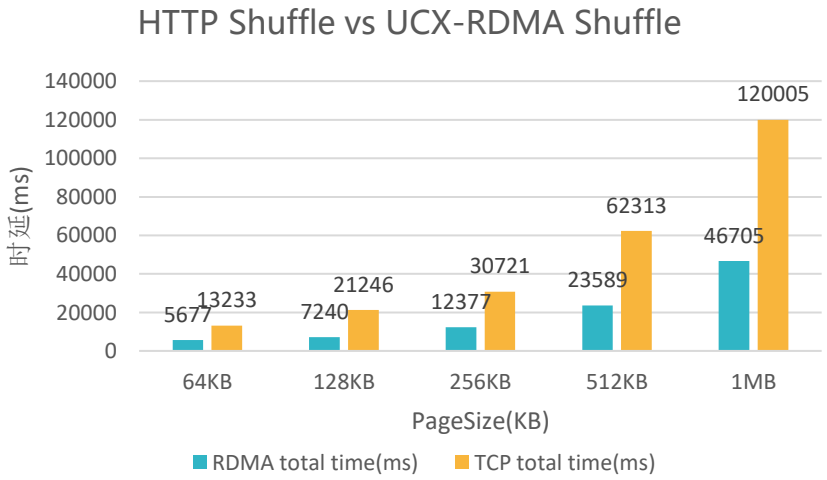
- **SQL访问层：** （1）SQL兼容性增强（92→2003）；（2）新增北向ODBC能力
- **分析引擎层：** （1）性能优化技术；（2）新增功能；（3）新增企业级特性
- **数据接入层：** （1）支持的数据源范围更大；（2）新增跨源跨DC接入能力；

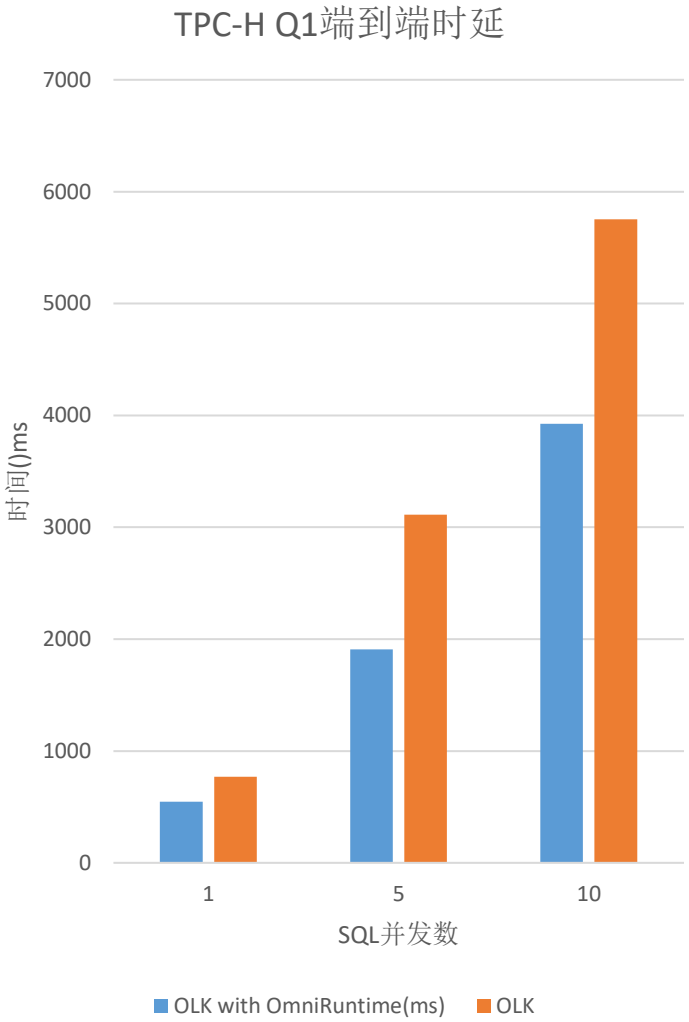
高性能算子性能对比











Memory connector 100G数据