

SequoiaDB与Spark在金融行业中的应用

王涛



Agenda

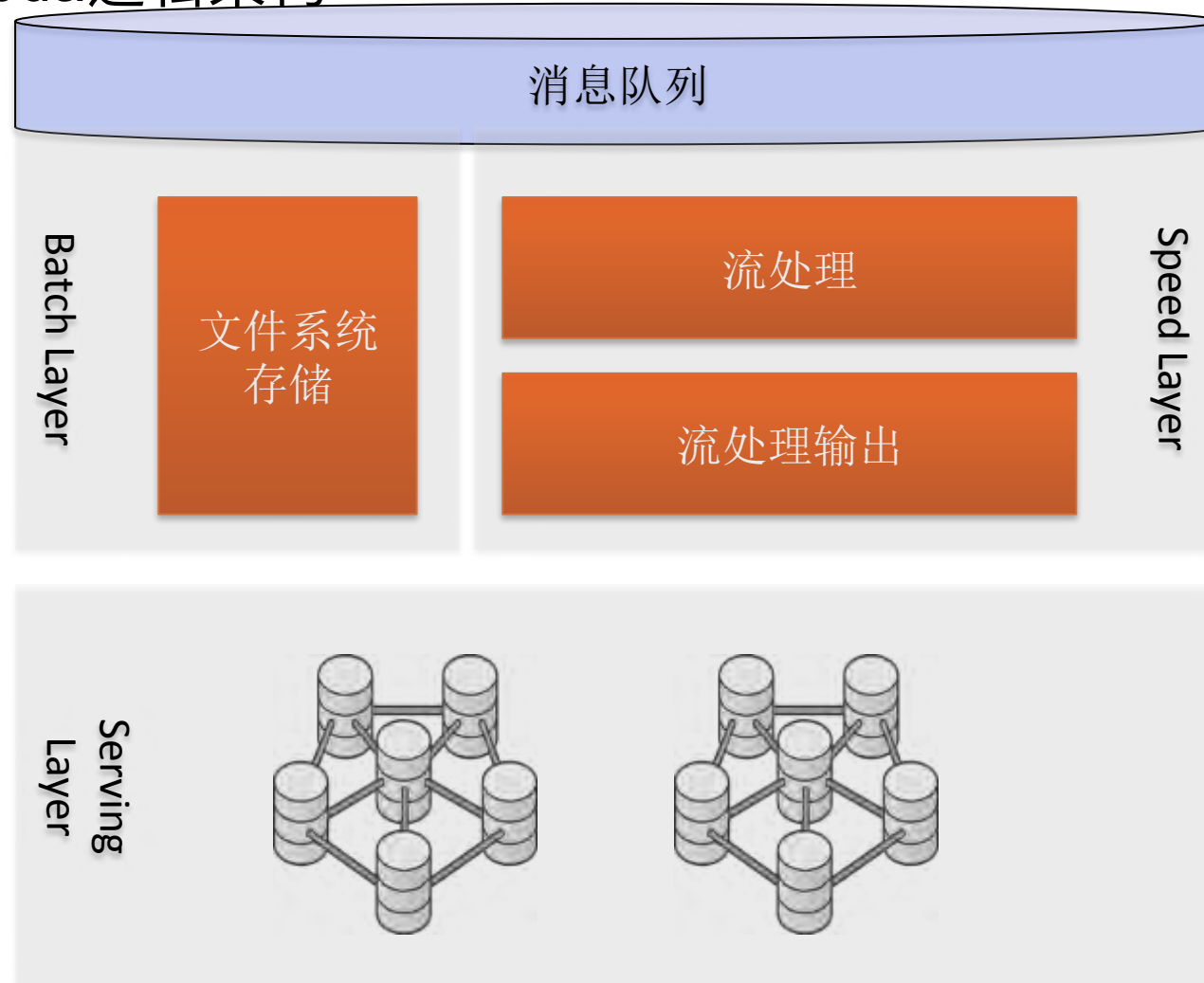
- What: 神马东西
- Why: 为什么这么做
- How: 具体怎样做的
- Where: 什么地方在用

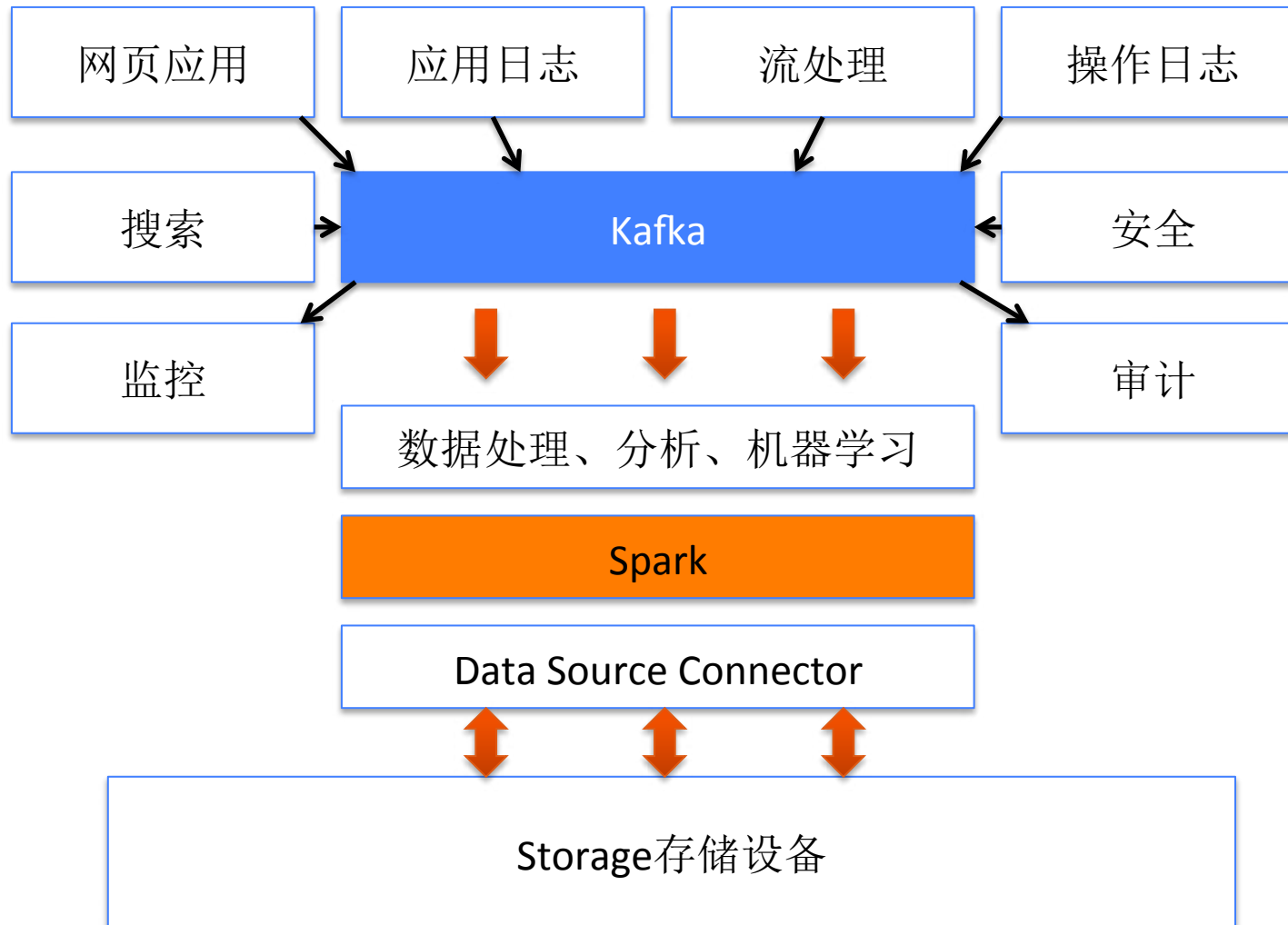


Lambda 架构

- A data-processing architecture designed to handle *massive quantities* of data by taking advantage of both batch and stream processing methods. (wikipedia)
- 一种使用批处理和流处理的方式，对海量数据进行处理的结构。
- Spark就是这样一种在同一个中间件处理框架中，有效地支持跑批和流处理的框架。

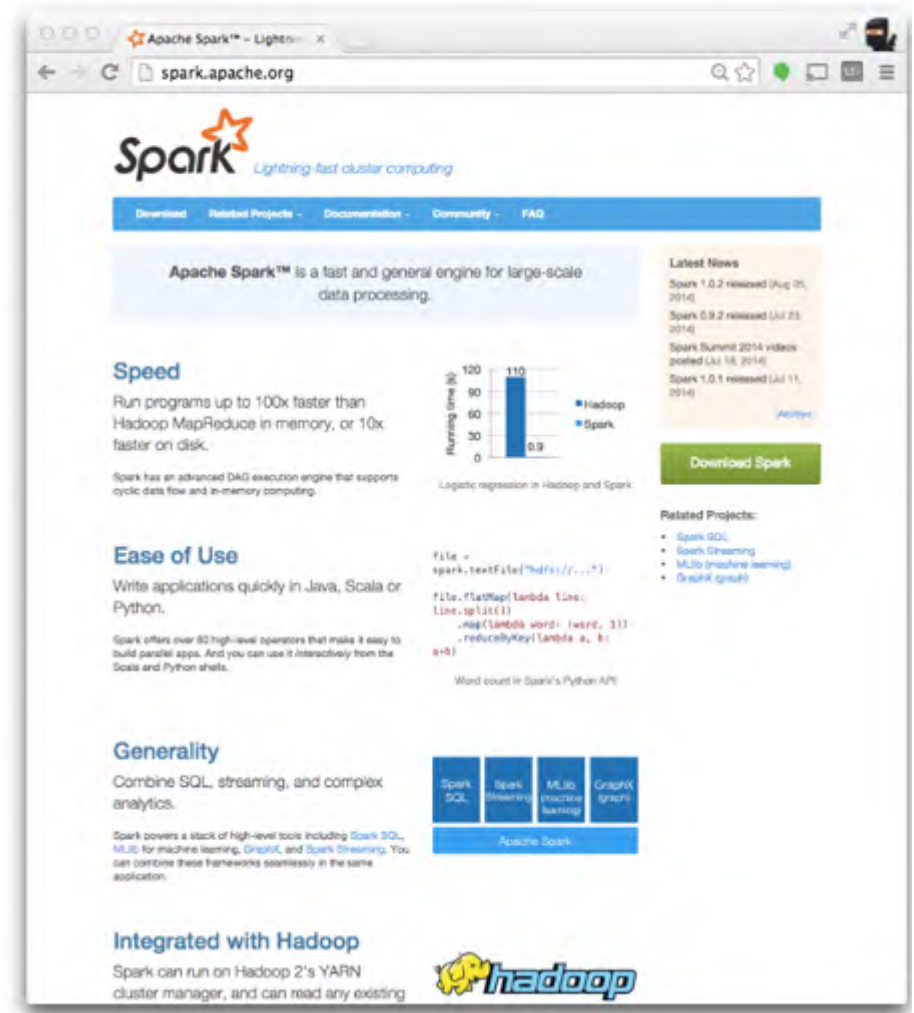
Lambda逻辑架构







- Fast, distributed, scalable and fault tolerant cluster compute system
- Enables Low-latency with complex analytics
- Developed in 2009 at UC Berkeley AMPLab, open sourced in 2010, and became a top-level Apache project in February, 2014



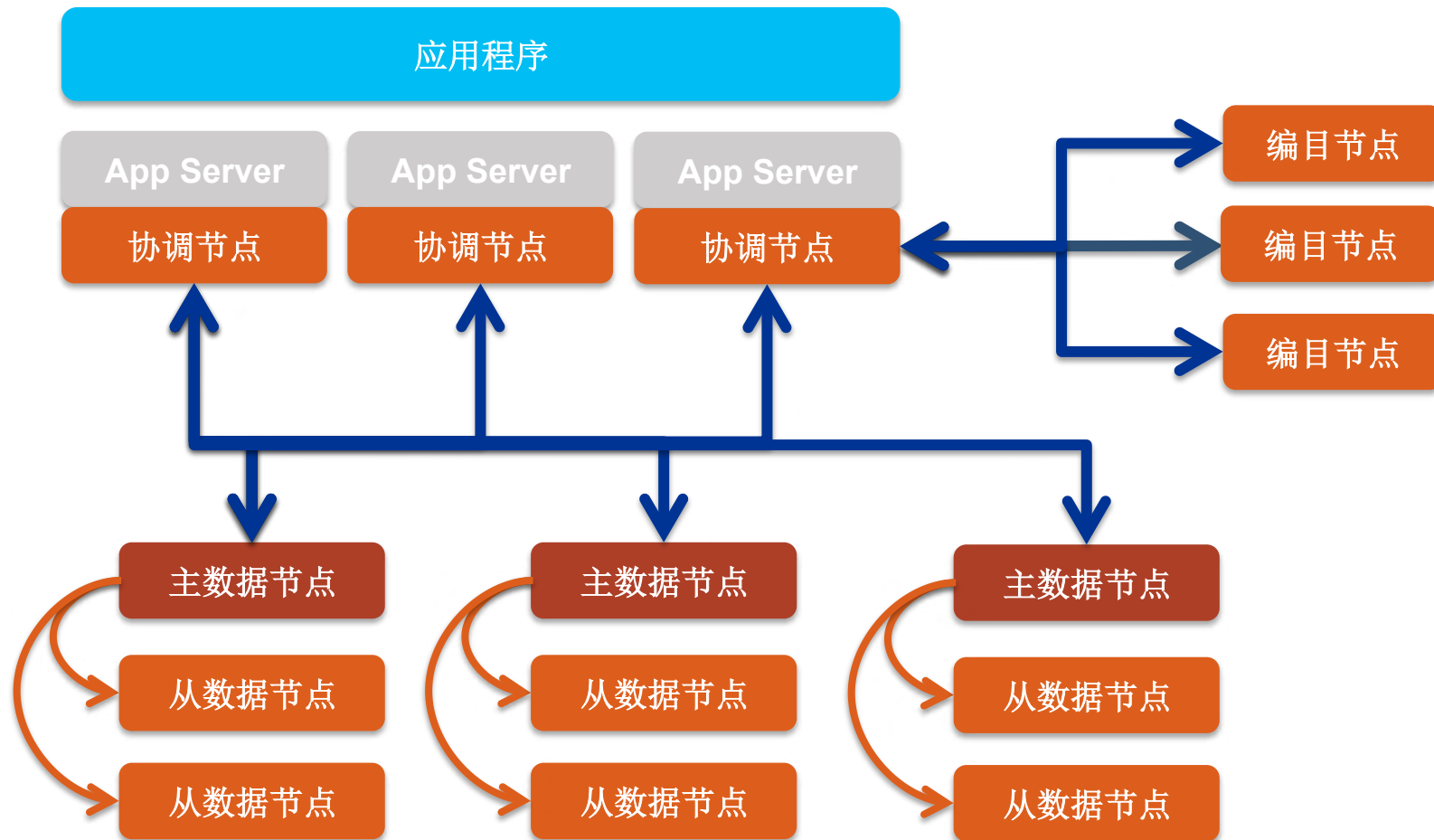
The screenshot shows the Apache Spark website homepage. The browser address bar displays 'spark.apache.org'. The page features the Spark logo and tagline 'Lightning-fast cluster computing'. A navigation menu includes 'Download', 'Related Projects', 'Documentation', 'Community', and 'FAQ'. A main banner states: 'Apache Spark™ is a fast and general engine for large-scale data processing.' Below this, there are sections for 'Speed', 'Ease of Use', 'Generality', and 'Integrated with Hadoop'. The 'Speed' section includes a bar chart comparing Hadoop (110) and Spark (0.9) running times. The 'Ease of Use' section shows code snippets for file operations. The 'Generality' section lists Spark SQL, Streaming, MLlib, and GraphX. The 'Integrated with Hadoop' section mentions YARN. A 'Latest News' sidebar lists recent releases and a 'Download Spark' button is visible.



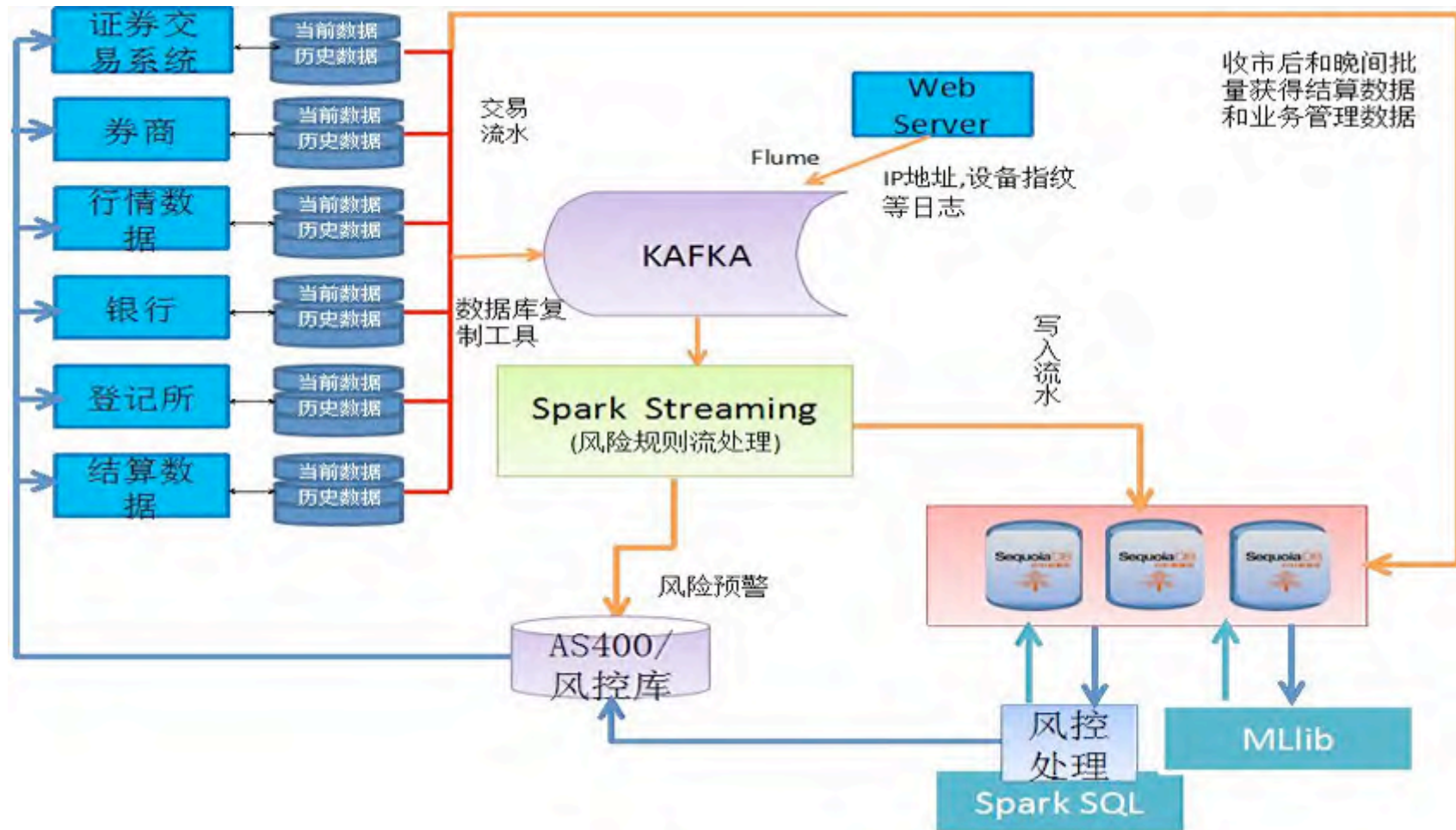
广州巨杉软件开发有限公司（简称巨杉软件）成立于2011年专注于新一代企业大数据平台研发,其核心产品SequoiaDB（巨杉数据库）是国内第一款新一代开源NoSQL数据库。



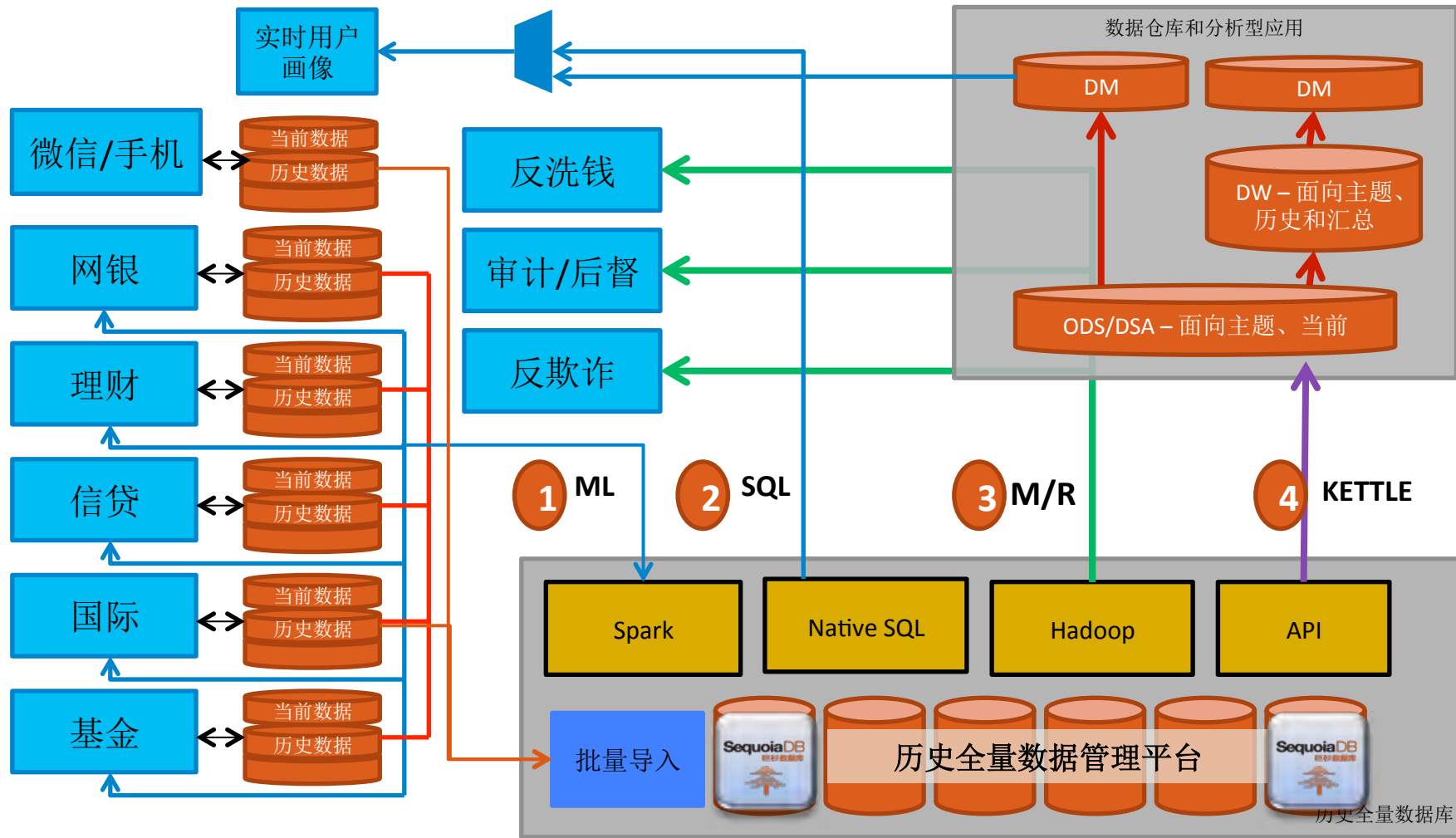
数据存储层：SequoiaDB

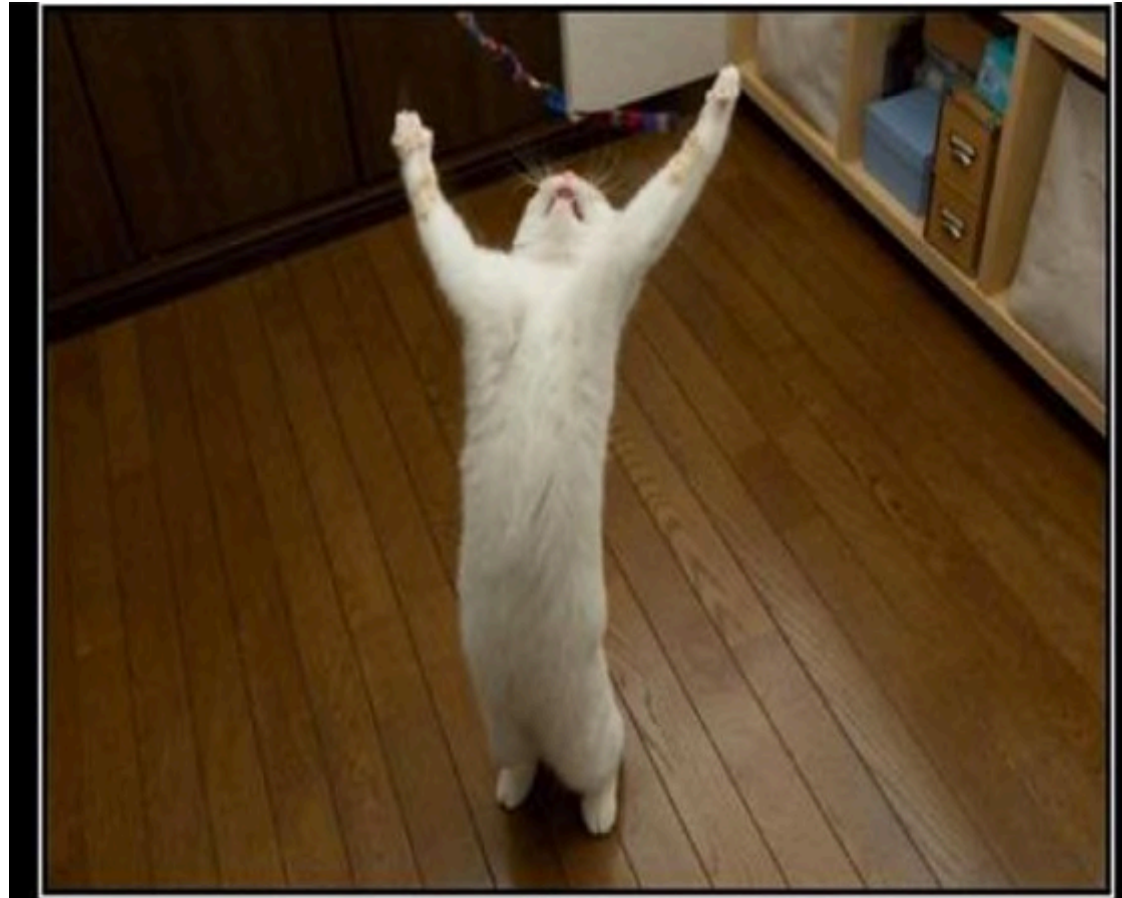


实时处理场景架构



批处理场景架构

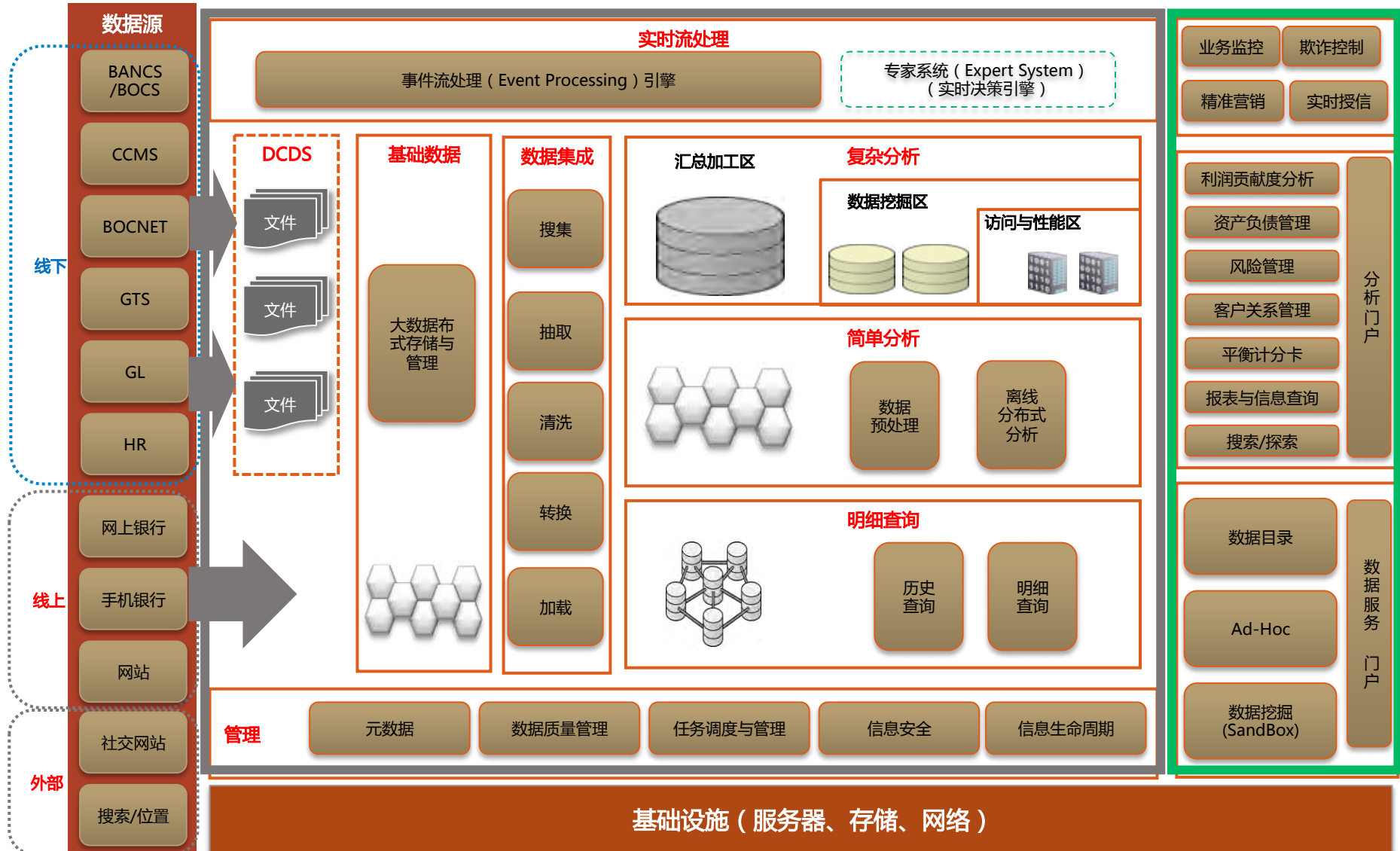




WHY???

WHY GOD WHY!!!

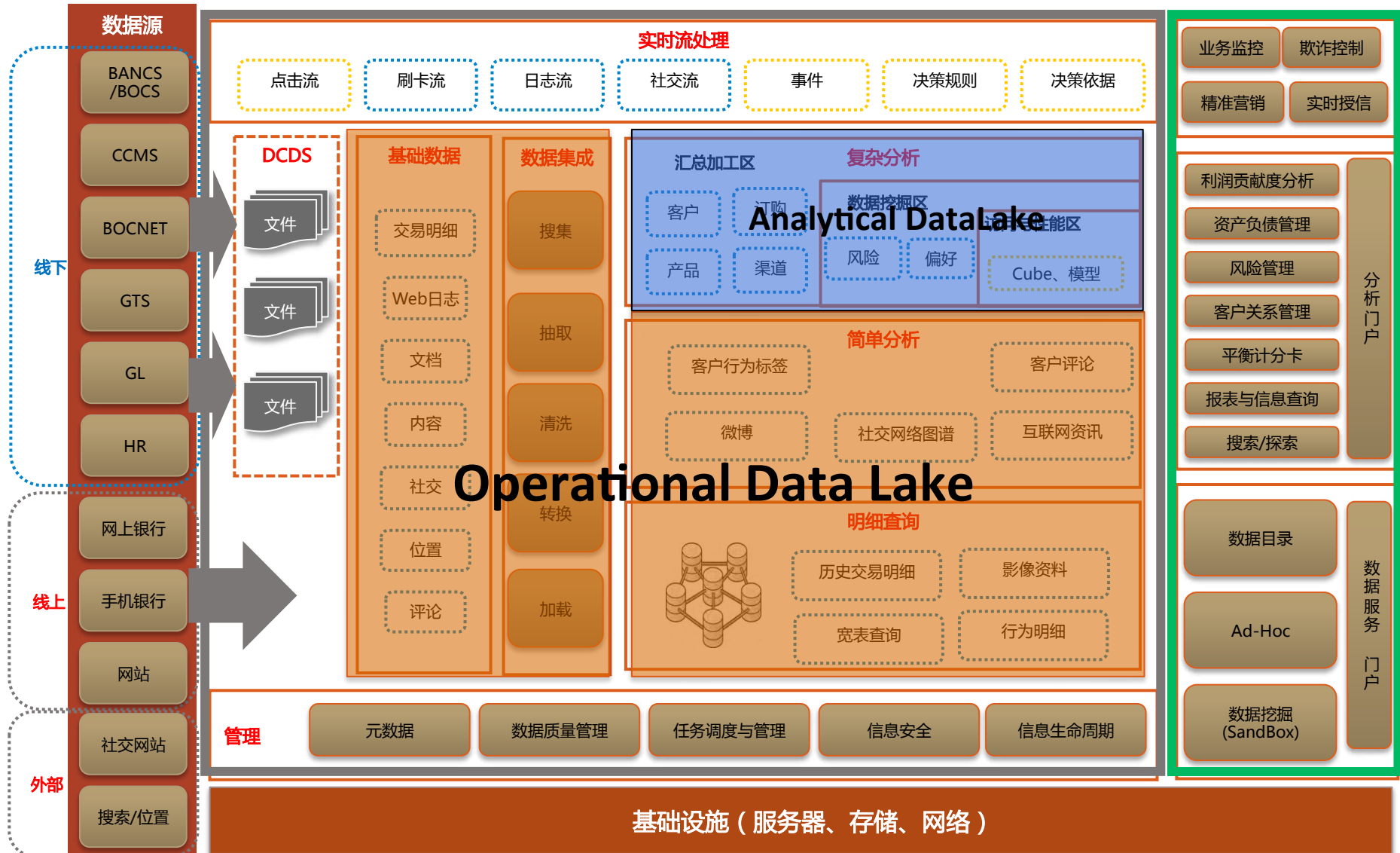
传统应用、ODS、数据仓库之间的距离在缩短



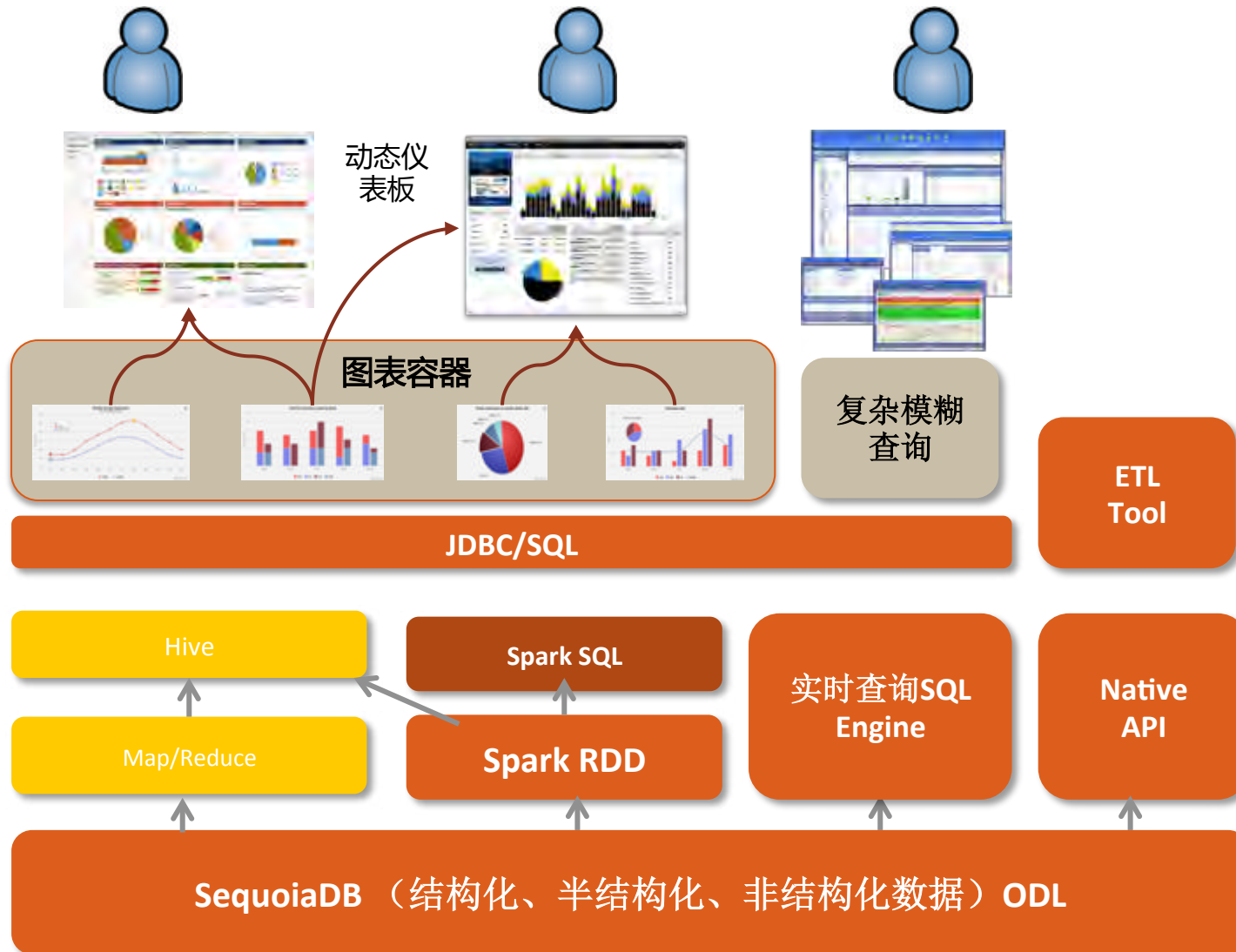
大数据平台的数据存储

- 数据结构的自适应尤其重要
- ODS (Operational Data Store)向ODL (Operational Data Lake)的转变
- 多种应用类型的支持

数据湖：DataLake



多种业务作用于ODL



SequoiaDB的5个特点 简化大数据应用开发和运维

1. 低成本管理海量数据

分布式架构, 水平扩展, 动态扩容

比传统IOE便宜!

2. 数据模型自适应

所见即所存 - 应用开发的敏捷性

与在线应用解耦

3. 支持混合业务模式

同时支持SQL和MR, 无需存两份数据

一套数据多种用途!

4. 海量数据时间序模型

按时间自动管理变化的数据模型

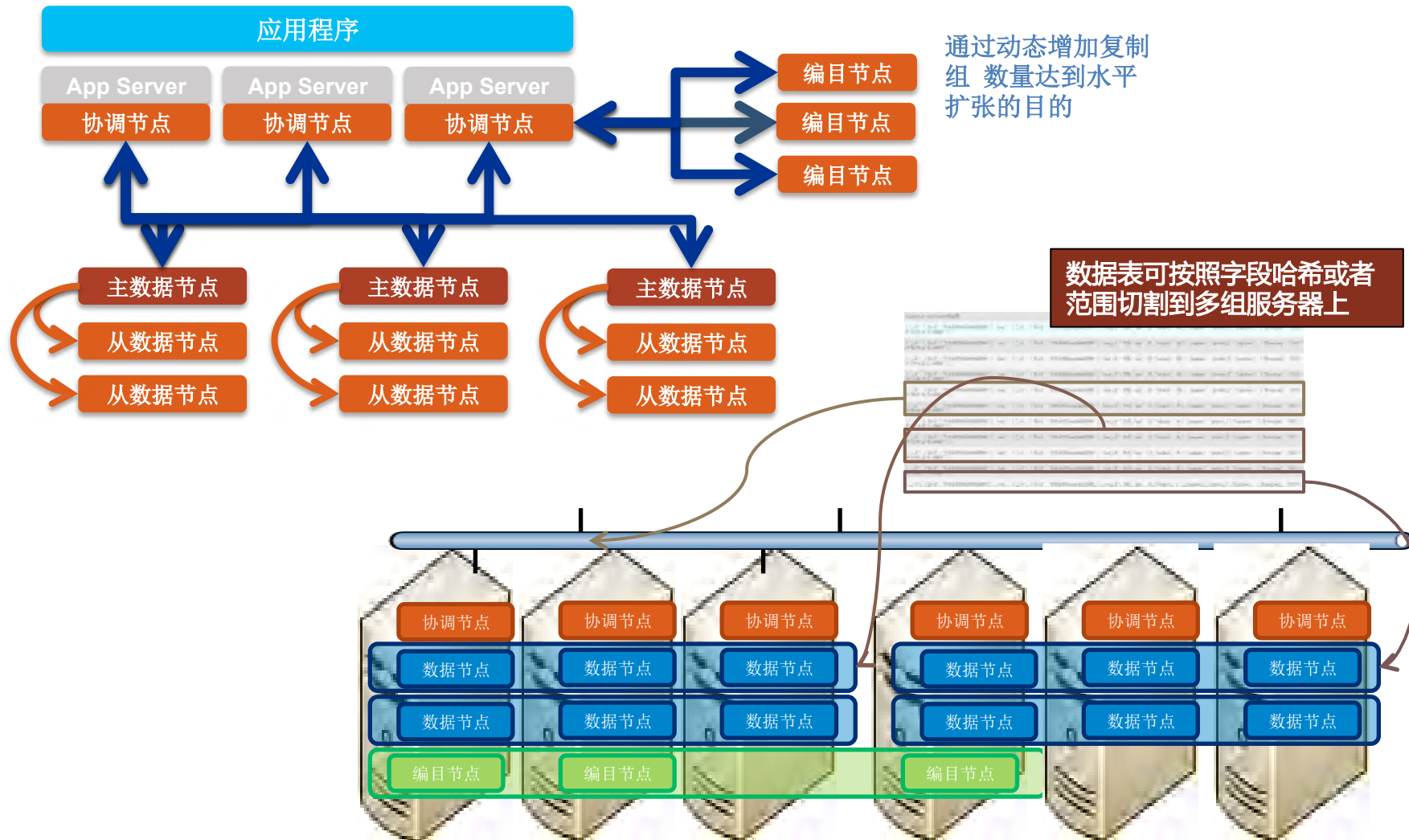
比MongoDB功能强!

5. 预集成Spark

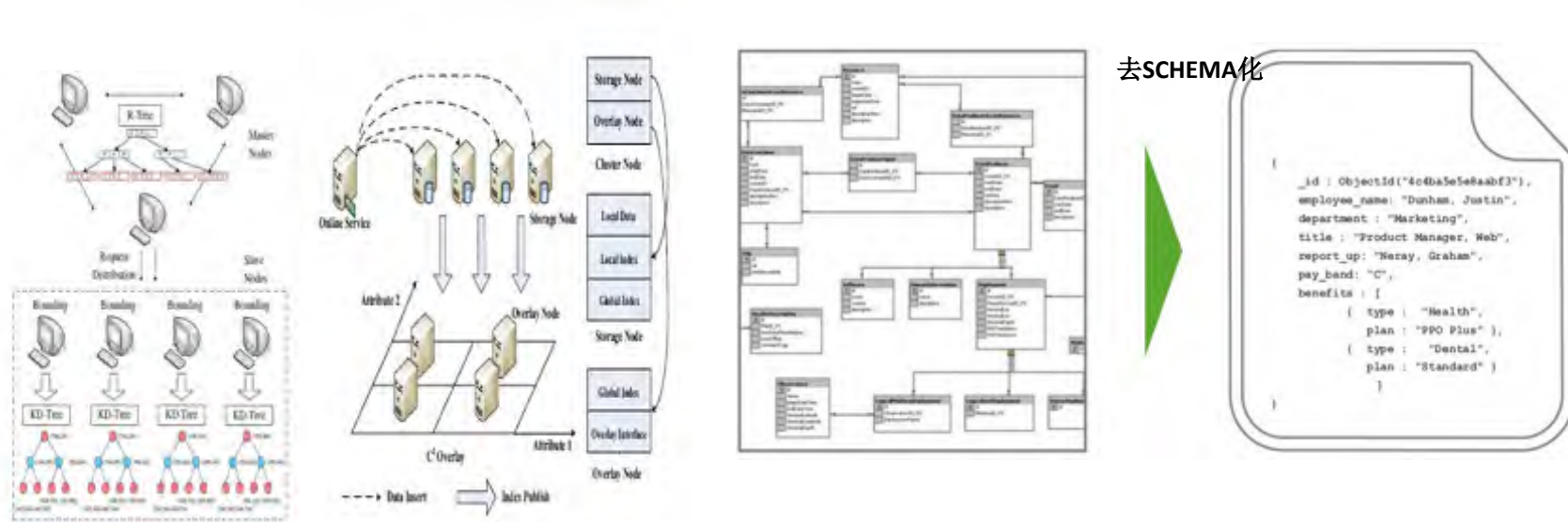
整合优化Spark, 降低开发部署成本

简化开发架构!

分布式数据管理架构



数据类型自适应



可靠性:

- 默认三备份存储, 可自由设定备份数量
- 科学的备份节点选择策略
- 自动化的数据恢复
- 多名称服务(NameNode)热备
- 高效数据验证机制

可扩展:

- 节点均使用廉价的X86设备
- 可通过新增节点来实现存储和计算能力的线性扩展
- 增删节点不影响系统的使用

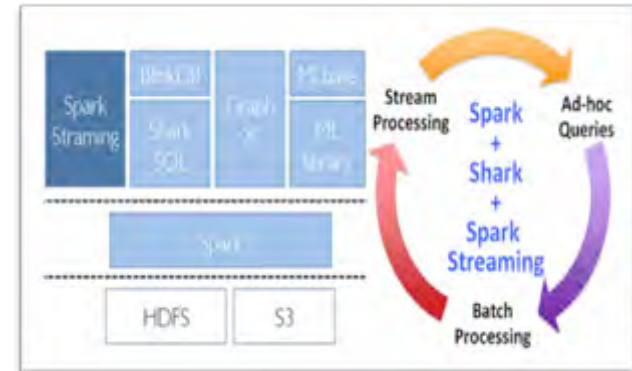
灵活的JSON (XML)文档型存储:

- 数据按对象描述(JSON)存储, 避免SCHEMA僵硬, 灵活适应变化
- 弱关系型数据存储, 同时提供SQL支持
- 非结构化数据的有序组织和管理
- 数据类型一致, 数据特征相似 - 高效压缩

高性能

- 分布并行查询, 多索引
- 只访问查询设计的列 - 大量降低系统IO
- 每一分区由一个线程来处理 - 提供更高的并发
- 针对数据汇总计算的需求, 可自定义设置数据存储的分区和分片策略

支持混合业务模式



多重计算框架:

- 实时流式计算框架
- MapReduce并行计算框架
- Spark RDD内存处理框架
- 基于内存的有向无环图(DAG)框架

丰富的数据挖掘算法库:

- 分类算法
- 聚类算法
- 关联规则
- 协同过滤
- 向量相似度计算等

多种SQL执行引擎:

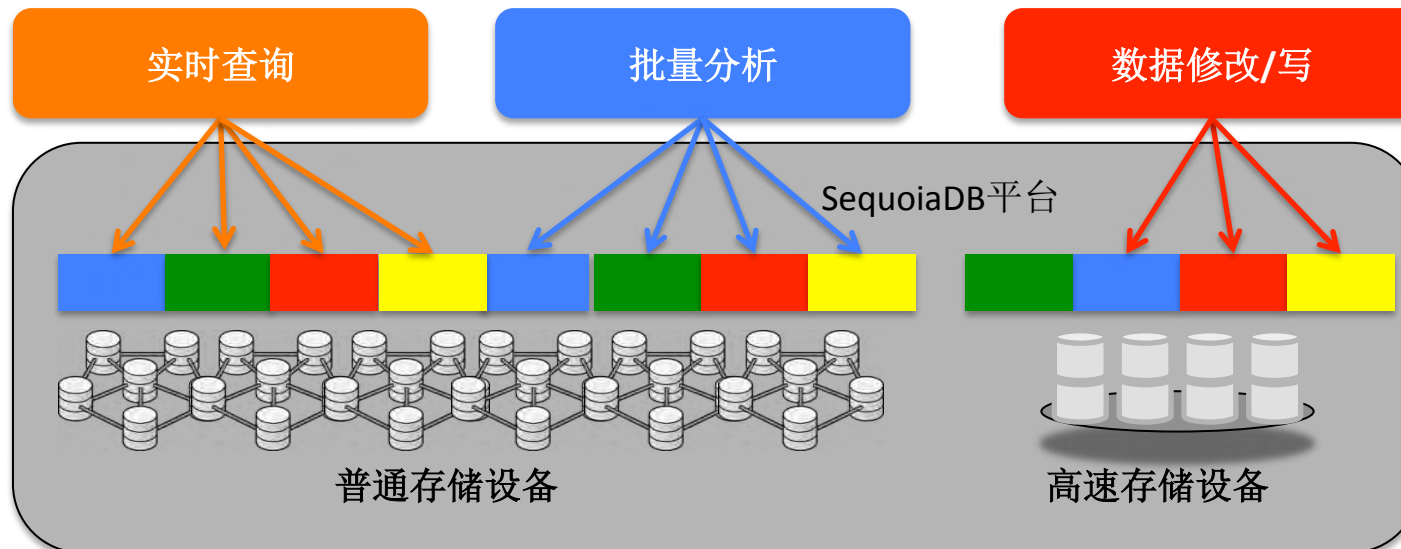
- 支持普通的标准SQL/JDBC/ODBC应用
- 基于MR架构的SQL引擎，执行高负载的汇总、勾兑等查询
- 基于MPP架构的实时SQL引擎，负责执行OLAP的查询

可线性扩展的计算性能:

- 基于廉价X86硬件设计的可扩展架构
- 任务被分布到集群上的各个节点进行
- 动态的增加节点可实现集群计算性能的线性扩展

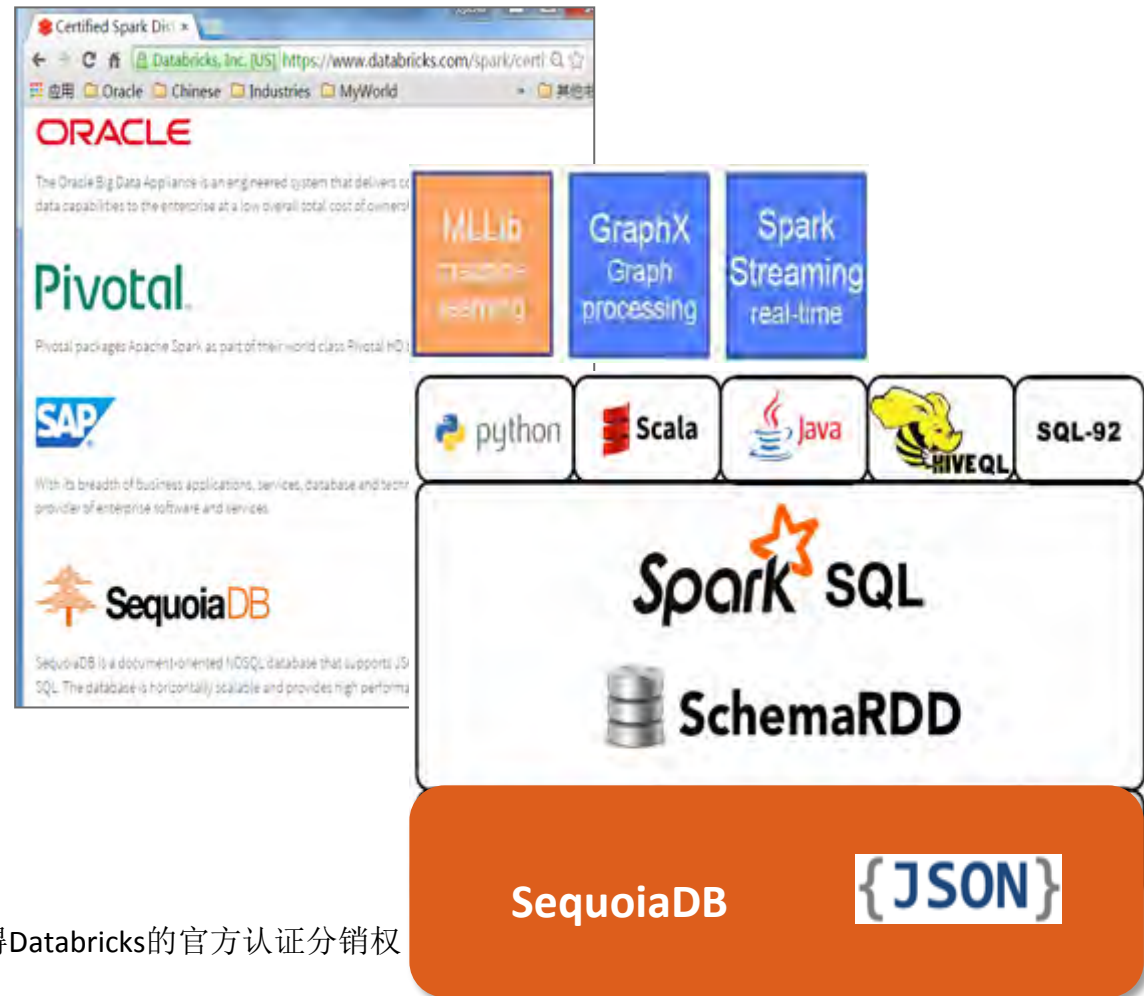
一套数据, 多种用途

- 数据在多个分布节点内可以自动分片和复制, 可以定制数据分布策略. 保证多类型业务可以运行在同一平台上, 比如:
 - 冷/热数据区分离
 - 读/写分离
 - 写交易的“强一致性”和“弱一致性”分离
 - 查询/批量分离



产品预集成 Spark

- 1, 应用层面整合:
 - 充分利用Spark内存加速机制
 - 支持用Scala开发存储过程
 - 支持RDD、transformation等操作, 减少I/O操作
- 2, 数据层面
 - Spark SQL原生方式高效访问 SequoiaDB的数据
- 3, 管理层面
 - SDB企业版提供的管理工具Admin Center可以同时管理Spark Standalone模式。



获得Databricks的官方认证分销权

轻松整合到现有Hadoop环境

1, 应用层面整合:

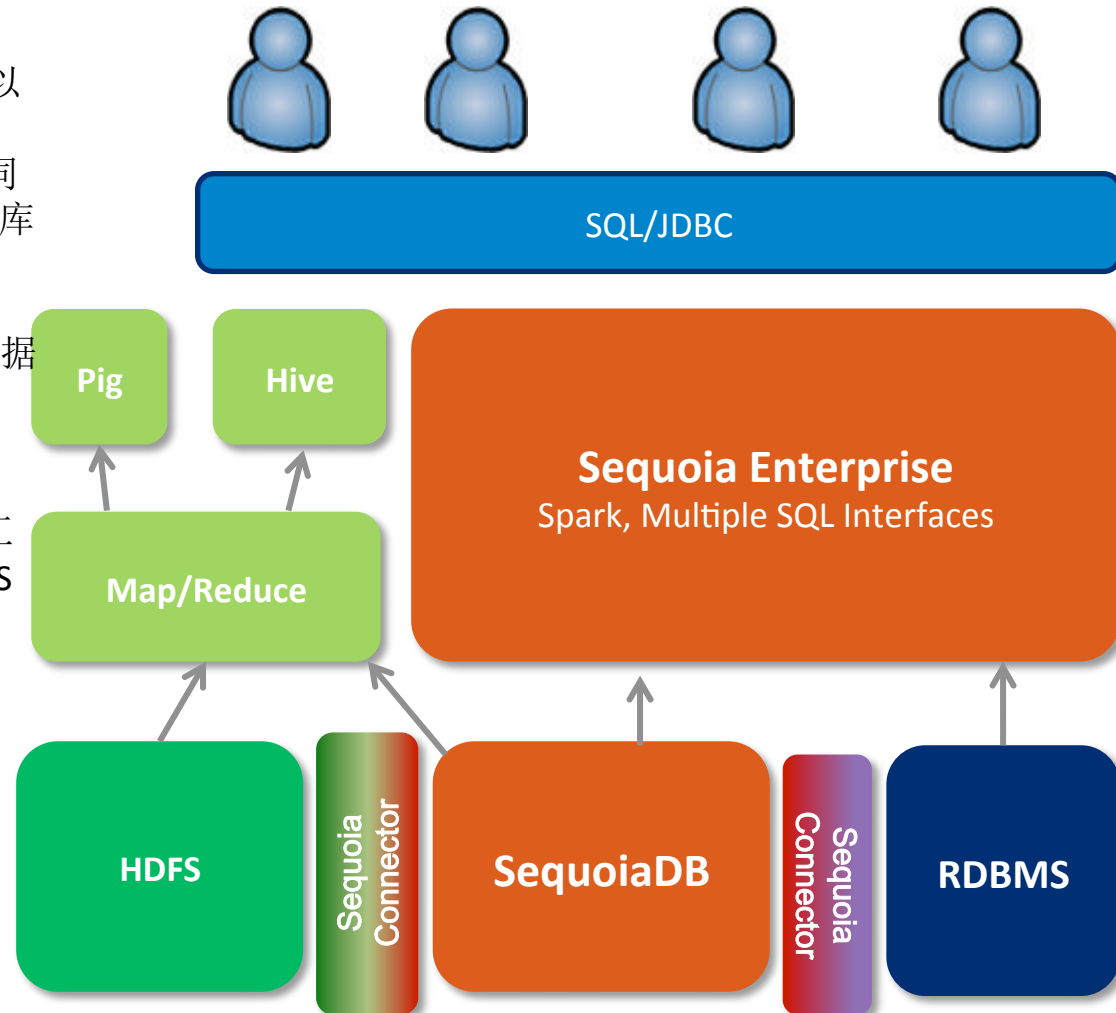
- 现有M/R和Spark代码可以同时访问SequoiaDB和HDFS/Hbase
- 现有SQL/JDBC客户可以同时访问SequoiaDB和传统关系数据库

2, 数据层面

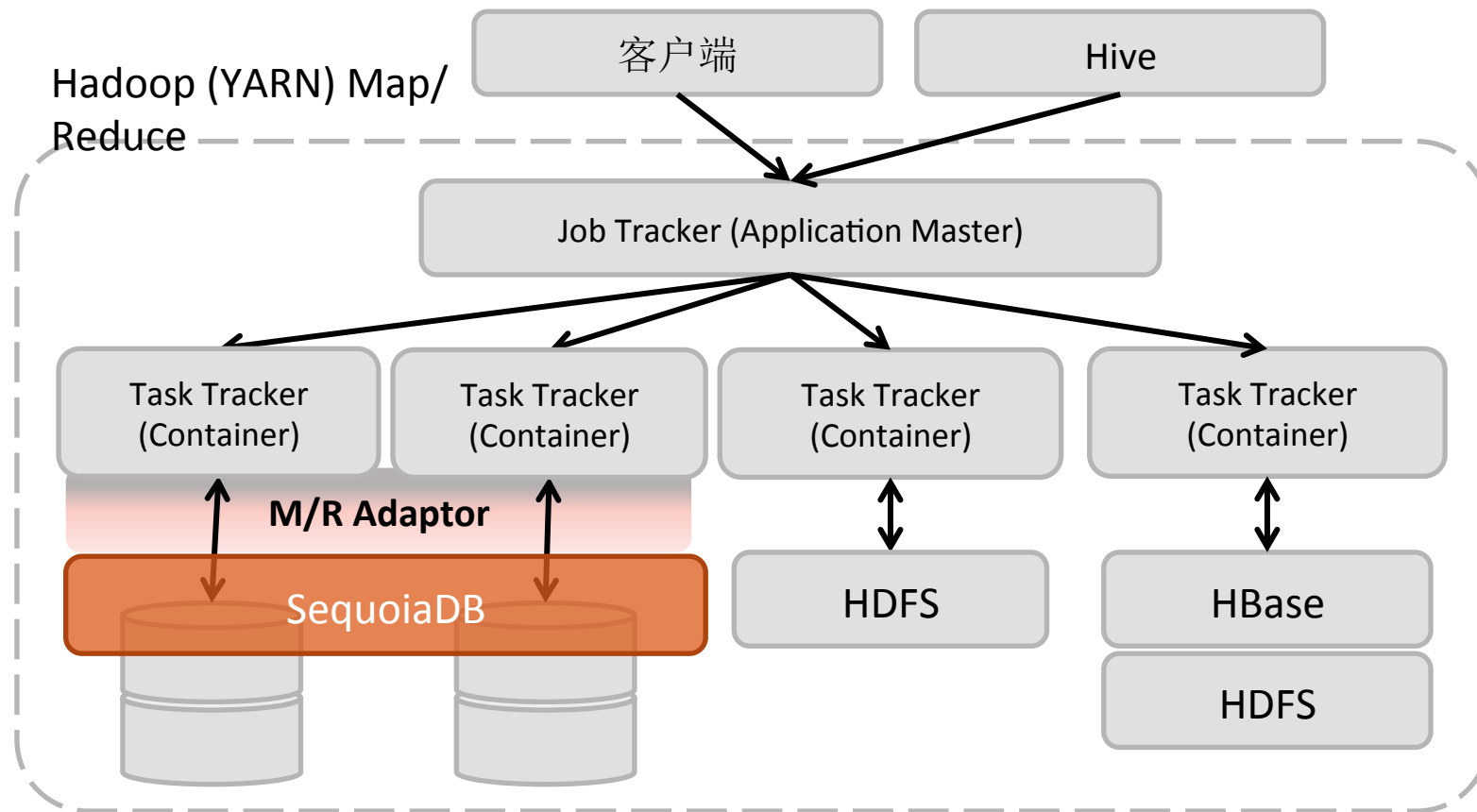
- 提供Connector在异构数据源中相互交换数据

3, 管理层面

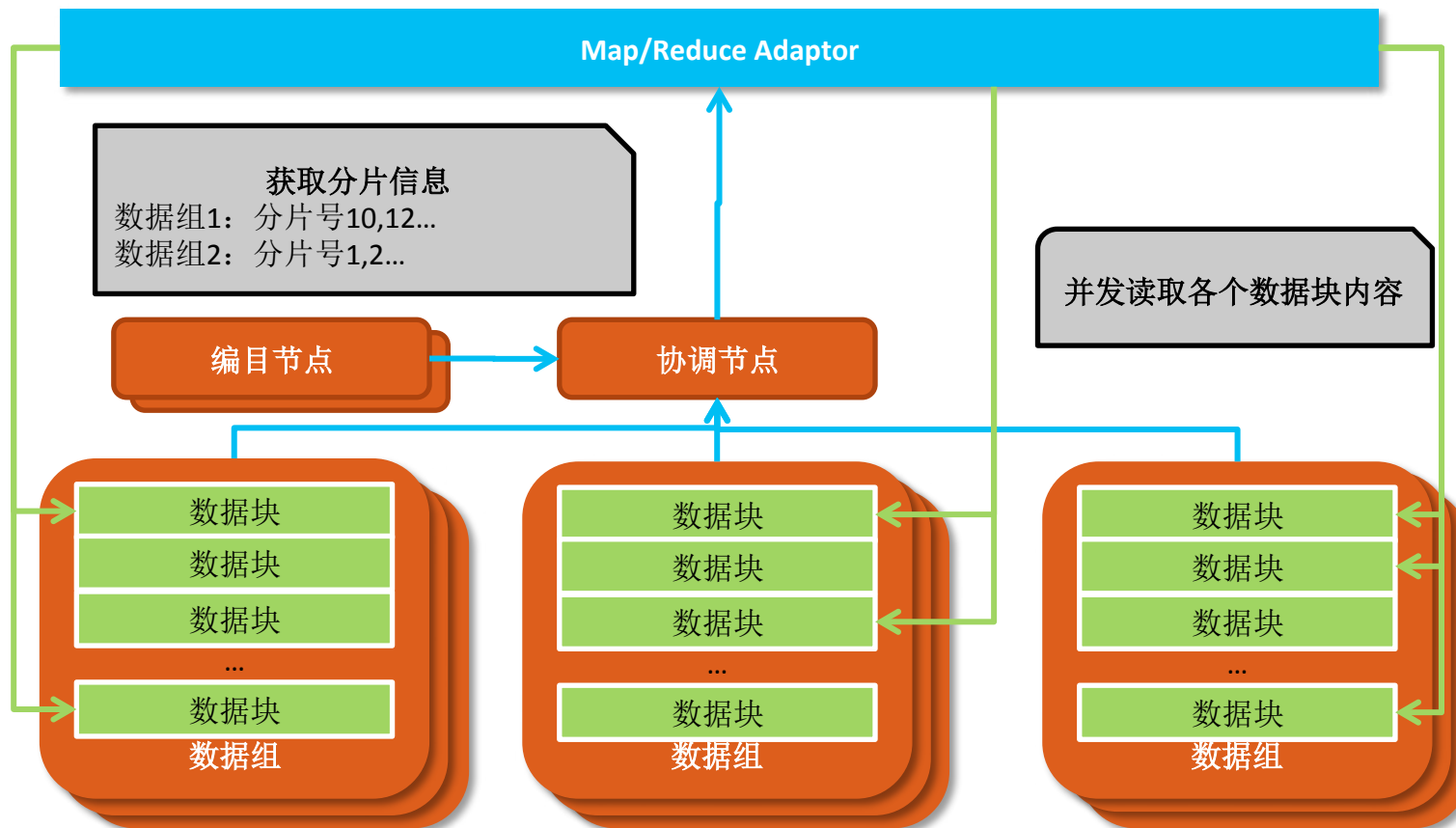
- SDB企业版提供的管理工具Admin Center可以同时管理HDFS和Spark, Hive。



轻松整合到现有Hadoop环境



针对Hadoop MapReduce的性能优化



- 1 大数据查询和导入性能相对HBASE有大幅度提升；
- 2 与Hadoop Map/Reduce 对接能发挥更大的并发优势；



Spark-SequoiaDB Connector

COMPANY

[All Posts](#)

[Partners](#)

[Events](#)

[Press Releases](#)

DEVELOPER

[All Posts](#)

[Spark](#)

[Spark SQL](#)

[Spark Streaming](#)

[MLlib](#)

[Spark Summit](#)

Search Blog 

Subscribe 



Guest blog: SequoiaDB Connector for Apache Spark

August 3, 2015 | by Tao Wang



This is a guest blog from Tao Wang at [SequoiaDB](#). He is the co-founder and CTO of SequoiaDB, leading its long-term technology vision, and is responsible for the leadership of advanced technology incubations. SequoiaDB is a JSON document-oriented transactional database.


Why We Chose Spark




SequoiaDB is a NoSQL database that has the capability to replicate data on different physical nodes and allows users to specify which "copy of data" that the application should access. It is capable of running analytical and operational workloads simultaneously on the same cluster with minimal I/O or CPU contention.

The joint solution of Apache Spark and SequoiaDB allows users to build a single platform such that a wide variety of workloads (e.g., interactive SQL and streaming) can run together on the same physical cluster.

Making SequoiaDB Work with Spark: The Spark-SequoiaDB Connector

Spark-SequoiaDB Connector



[Pull requests](#)
[Issues](#)
[Gist](#)













 SequoiaDB / **spark-sequoiadb**

 Unwatch 1
  Star 5
  Fork 1


Spark-SequoiaDB is a library that allows users to read/write data with Spark SQL from/into SequoiaDB collections. <http://www.sequoiadb.com> — [Edit](#)

 19 commits
  1 branch
  0 releases
  1 contributor


 Branch: **master**
[spark-sequoiadb / +](#)


change readme		
 wangzhonnew authored on 31 Mar	latest commit <code>db09c9eb89</code> 	
 project	add sbt-spark-package plugin	5 months ago
 sbt	remove part	5 months ago
 src	use explain	5 months ago
 LICENSE	initial commit	5 months ago
 README.md	change readme	5 months ago
 build.sbt	remove duplicate license	5 months ago
 pom.xml	add scala version	5 months ago

 **Code**


 [Issues](#) 1

 [Pull requests](#) 0


 [Wiki](#)

 [Pulse](#)

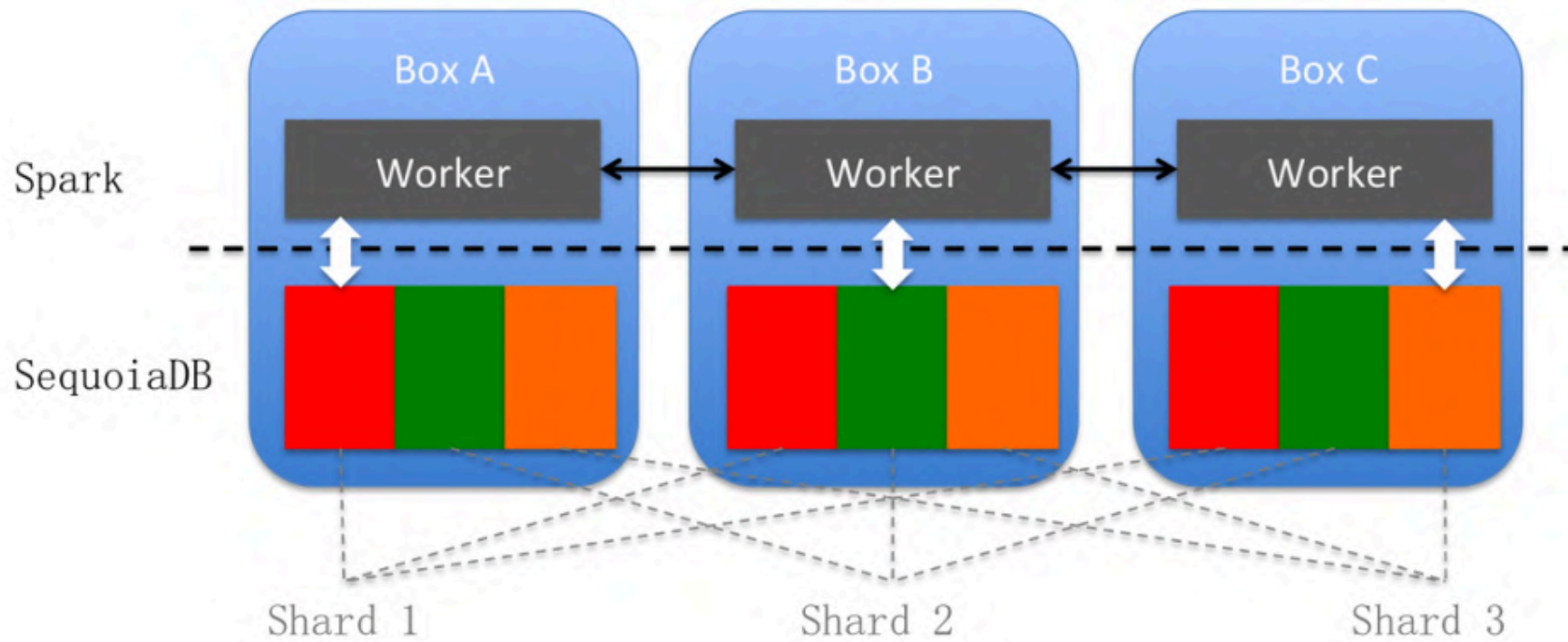
 [Graphs](#)

 [Settings](#)

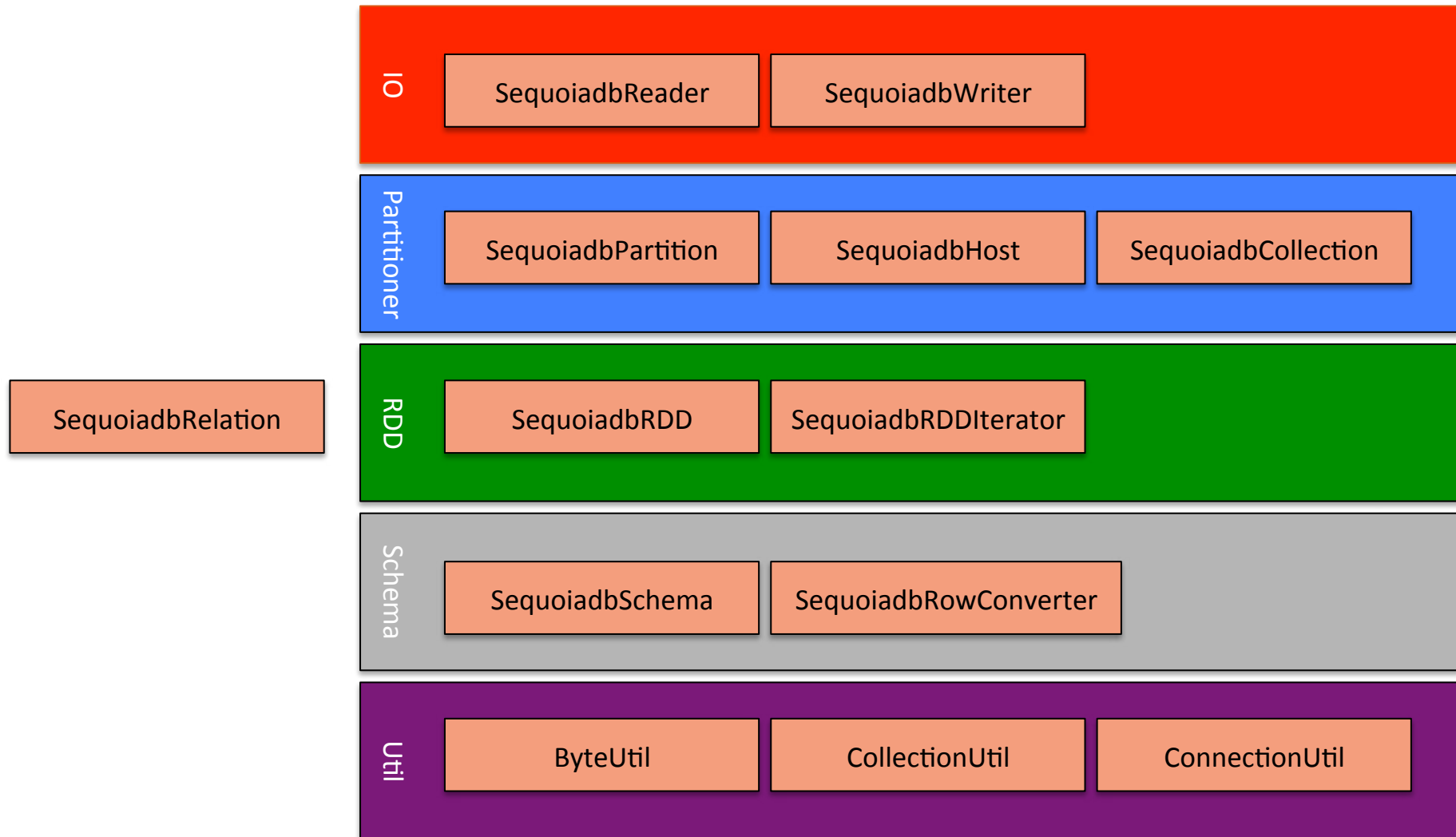
SSH clone URL
 

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#). 

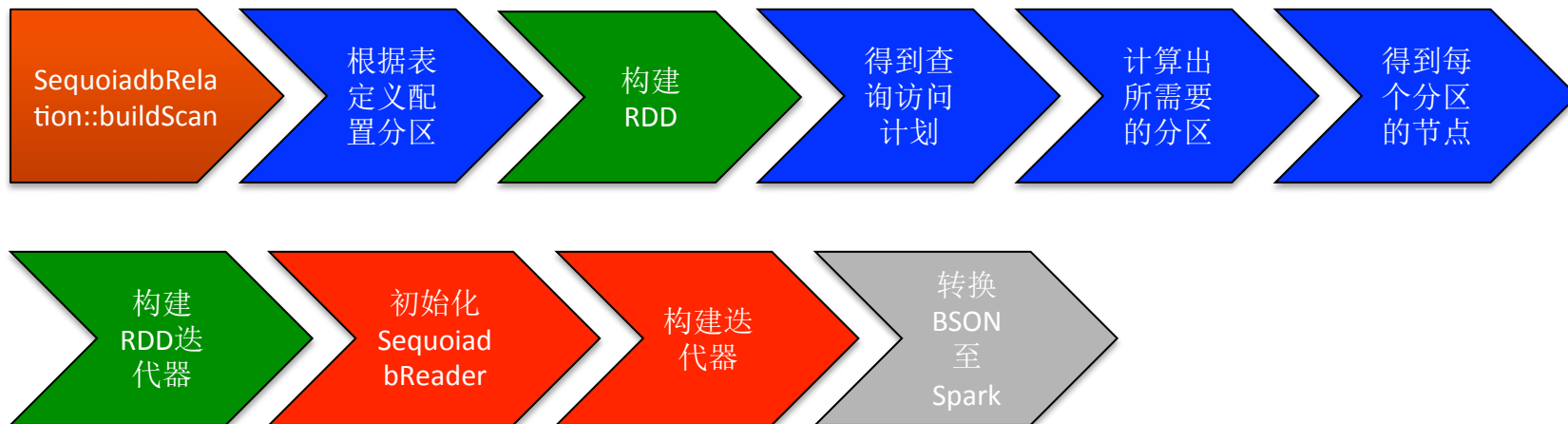
Spark/SequoiaDB联合架构



连接器结构



查询流程



SparkSQL使用方式

```
sqlContext.sql("CREATE temporary table org_department ( deptno string, deptname s  
tring, mgrno string, admrdept string, location string ) using com.sequoiadb.spark  
OPTIONS ( host 'host-60-0-16-2:50000', collectionspace 'org', collection 'depart  
ment', username 'sdbreader', password 'sdb_reader_pwd' )")  
res2: org.apache.spark.sql.DataFrame = []
```

```
sqlContext.sql("CREATE temporary table org_employee ( empno int, firstnme string,  
midinit string, lastname string, workdept string, phoneno string, hiredate date,  
job string, edlevel int, sex string, birthdate date, salary int, bonus int, comm  
int ) using com.sequoiadb.spark OPTIONS ( host 'host-60-0-16-2:50000', collectio  
nspcace 'org', collection 'employee', username 'sdb_reader', password 'sdb_reader_  
pwd' )")  
res3: org.apache.spark.sql.DataFrame = []
```

```
sqlContext.sql("select * from org_department a, org_employee b where a.deptno='D1  
1'").collect().take(3).foreach(println)  
[D11,MANUFACTURING SYSTEMS,000060,D01,null,10,CHRISTINE,I,HAAS,A00,3978,null,PRES  
,18,F,null,152750,1000,4220]  
[D11,MANUFACTURING SYSTEMS,000060,D01,null,20,MICHAEL,L,THOMPSON,B01,3476,null,MA  
NAGER,18,M,null,94250,800,3300]  
[D11,MANUFACTURING SYSTEMS,000060,D01,null,30,SALLY,A,KWAN,C01,4738,null,MANAGER,  
20,F,null,98250,800,3060]
```



WHERE?

历史数据存储

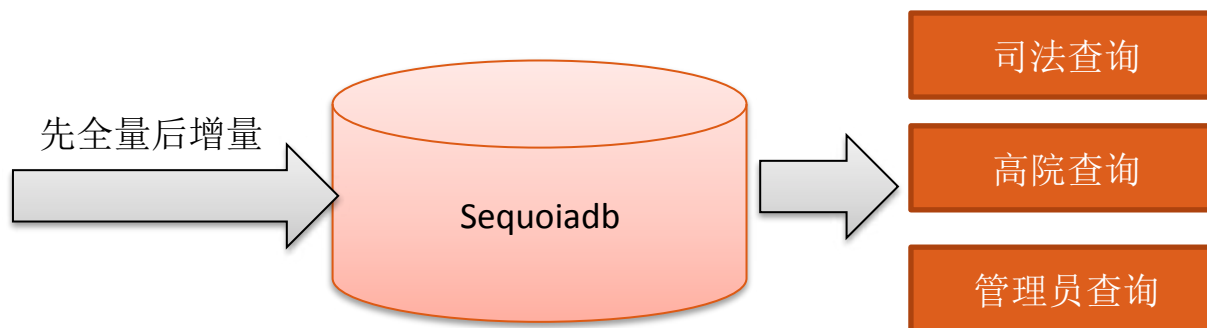
以前

- 数据量大
- 数据格式复杂
- 查询缓慢，异步请求，得到结果需要2天

现在

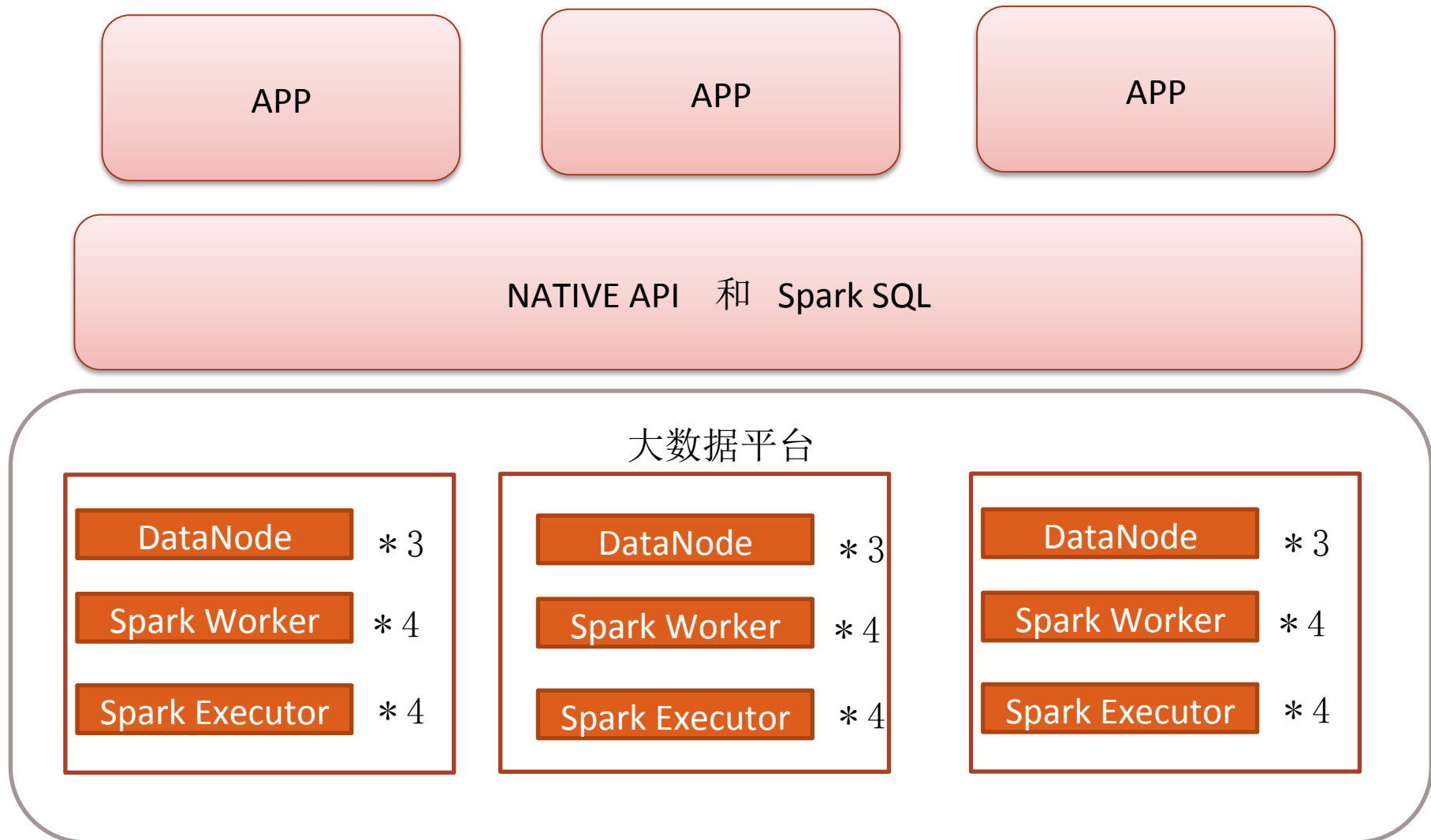
- 可以存储海量数据，动态扩容
- JSON无模式解决数据格式多变问题
- 查询秒级

海量大数据平台架构

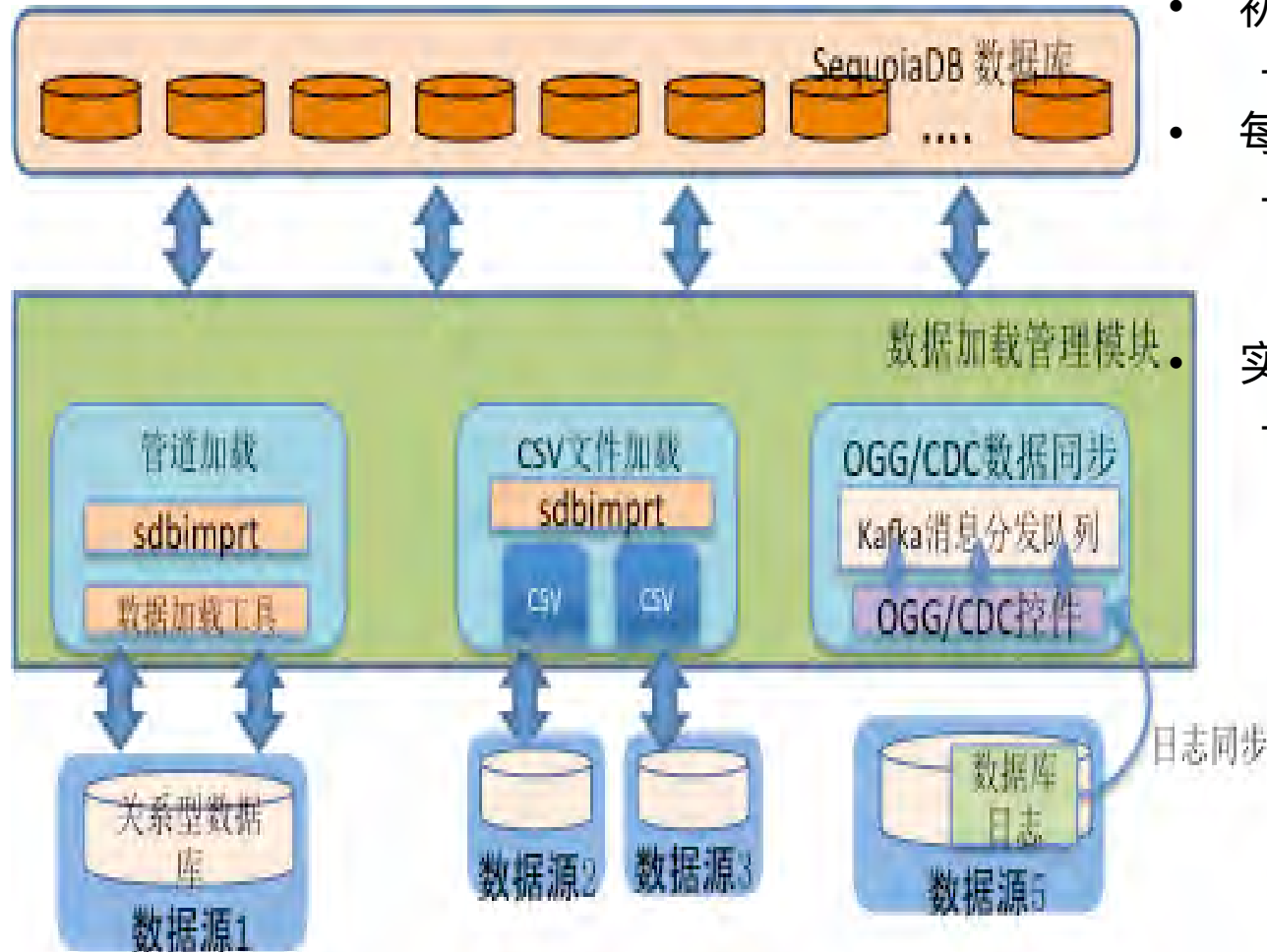


项目名称	数据量	数据大小	预计机器台数	备注
影像内容管理CM	4亿	45T影像+2T的数据	15	每台机器存放3T的影像+数据，响应性能比较紧张，根据上线的业务数量应该逐步扩容。
个人资产视图		4+T	6	实时业务，需要配置性能比较充裕的机器
核心、信用卡历史数据平台	50亿	14T(现有)+6.6T(未来2年)	12	一期配置6台，每台机器存放4T的数据最快查询15秒，批查40分钟，只能用于后台查询，预计未来需要扩容到12台以上。
网银，手机银行等历史数据		40T	24	
合计：			57	推荐最佳性能保障，我们建议机器的硬盘存储和内存比应该在1：10左右。每台机器处理2T以内的数据，3T以内的影像+数据

海量大数据物理架构



方便快捷的数据迁移方式



- 初始化全量数据加载
 - 直接读取CSV文件
- 每日增量数据
 - 通过JDBC服务直接从关系型数据库中加载到SequoiaDB中
- 实时数据同步
 - 依赖OGG/CDC工具实时解释日志文件，通过消息队列与应用程序进行数据加载

可变化的产品定义模型

JSON模型《客户产品明细表》：

CUST_NUM	PROD_CODE	SUB_PROD_CODE	CURRENCY	PROD_BAL	TIME_STAMP
客户编码	产品大类： “储蓄” “理财产品” “贵金属投资” “基金” “保险” “券商集合理财” “国债” “信托” “第三方存管” “个人贷款”	产品小类： “活期存款” “定期存款” “通知存款” “保证金存款” “其他存款”	币种	产品余额： 100000.00	交易时间

JSON模型《客户产品汇总表》

KEY	VALUE
CUST_NUM (客户编码)	{ “储蓄”：“100000.00”， “理财产品”：“100000.00” “贵金属投资”：“100000.00” “基金”：“100000.00” “保险”：“100000.00”， “券商集合理财”：“100000.00” “国债”：“100000.00” “信托”：“100000.00” “第三方存管”：“100000.00”， “个人贷款”：“100000.00” “个人总资产”： “1000000.00” “更新时间”： “201504231800” }

海量大数据平台搭建过程

项目背景

- 1、数据提取的外部需求多，投入人力成本大
- 2、现有数据提取工作模式复杂、效率低
- 3、国内新核心的投产对历史数据查询提出新的要求

系统建设方案-技术需求

数据量大，高效查询

不同条件查询需求

各历史阶段数据结构不统一

热点数据查询频度高

快速支持各业务系统接口开发

分布式结构，并行处理

多字段自由定义索引

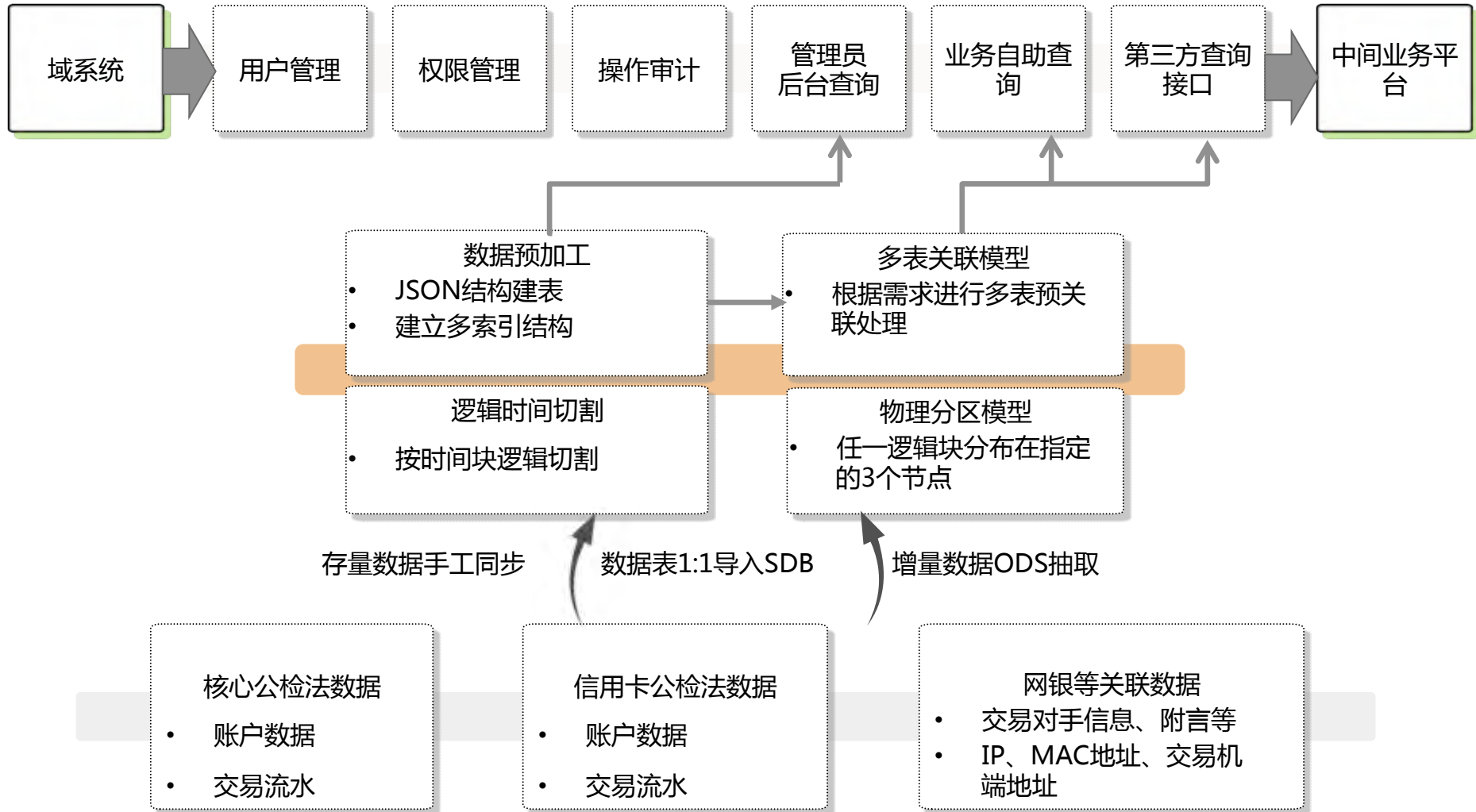
灵活应对表结构变化的数据结构

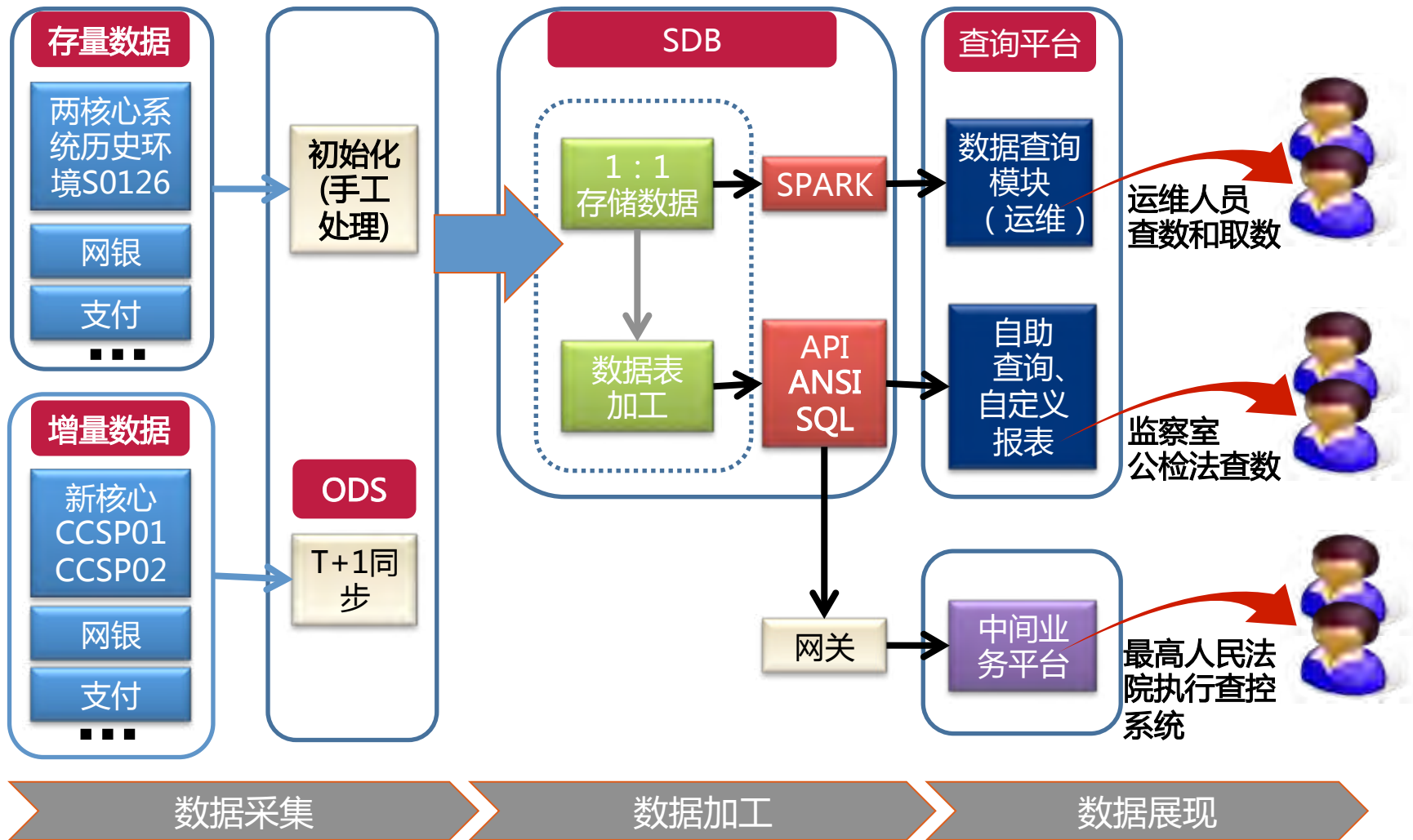
优化数据存储结构（时间性）

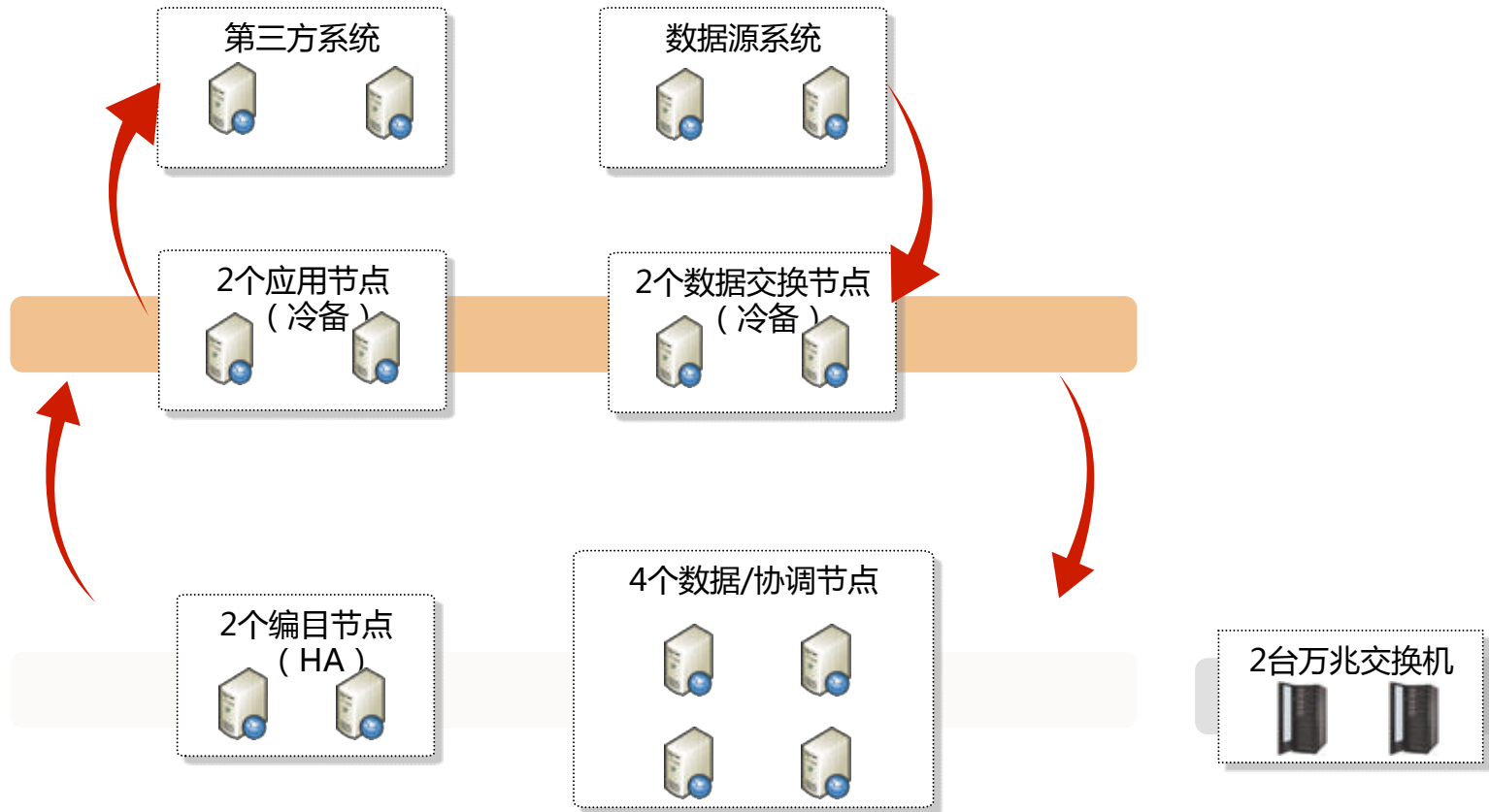
支持JDBC等通用数据查询接口

	SequoiaDB	DB2/ORACLE	Hbase + Hive
数据结构低耦合	支持	支持较差	支持较差
JDBC等通用数据查询接口	支持	支持	支持较差
分布式计算	支持	不支持	支持
多索引结构	支持	支持	支持较差
多表关联查询	支持	支持	支持较差
优化数据存储时间性	支持	支持	支持较差

计划引用巨杉数据库，满足历史数据查询平台多查询条件、表结构低依赖性的需求。

三
系统建设方案-逻辑架构






数据预处理加工

- 基于JSON格式，支持多索引结构
- 多表预关联处理

时间序模型

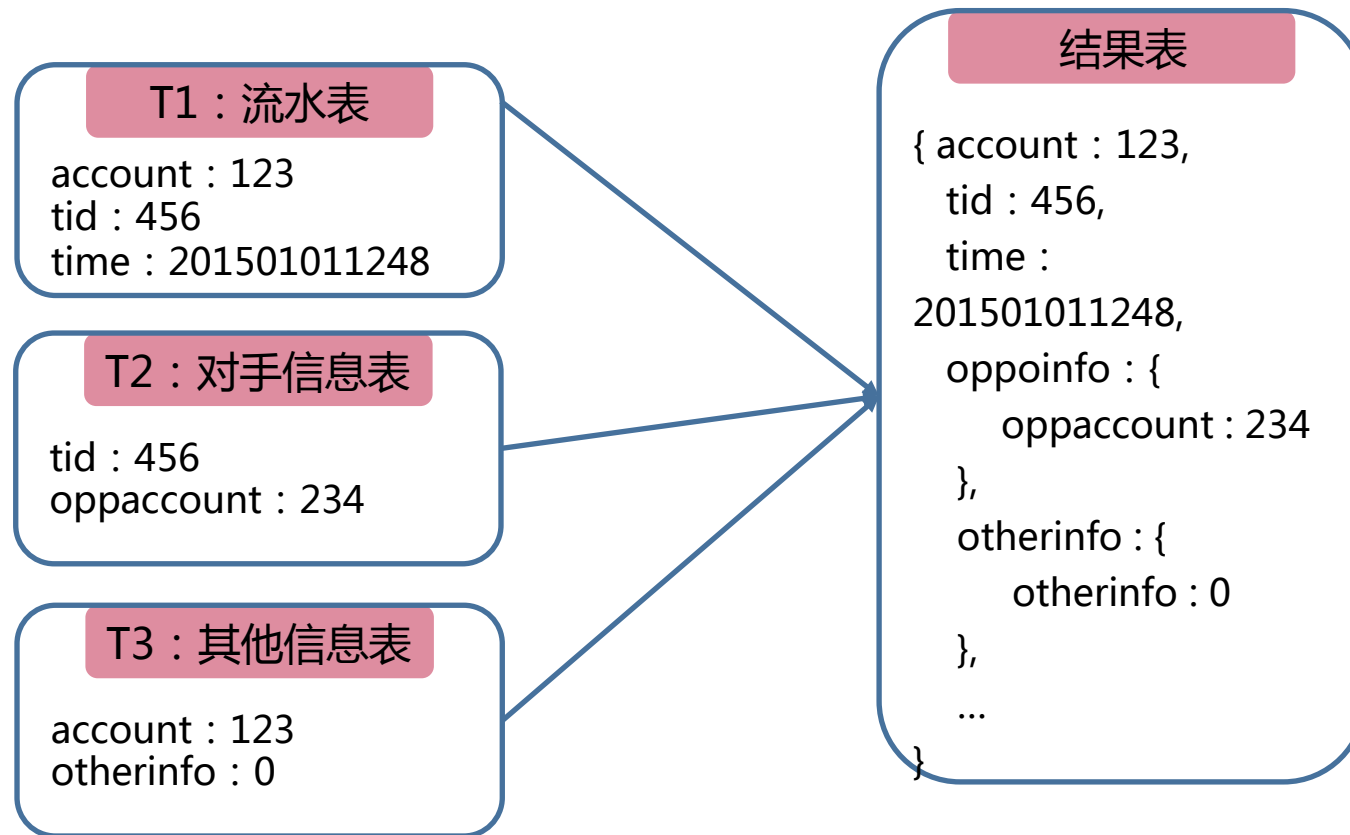
- 数据表加统一时间字段
- 旧数据的热度随着时间的推移递减
- 按时间范围段分区

数据分区模型

- 支持垂直逻辑分区和水平物理分区
- 垂直分区选择时间戳，水平分区选择唯一性较高字段

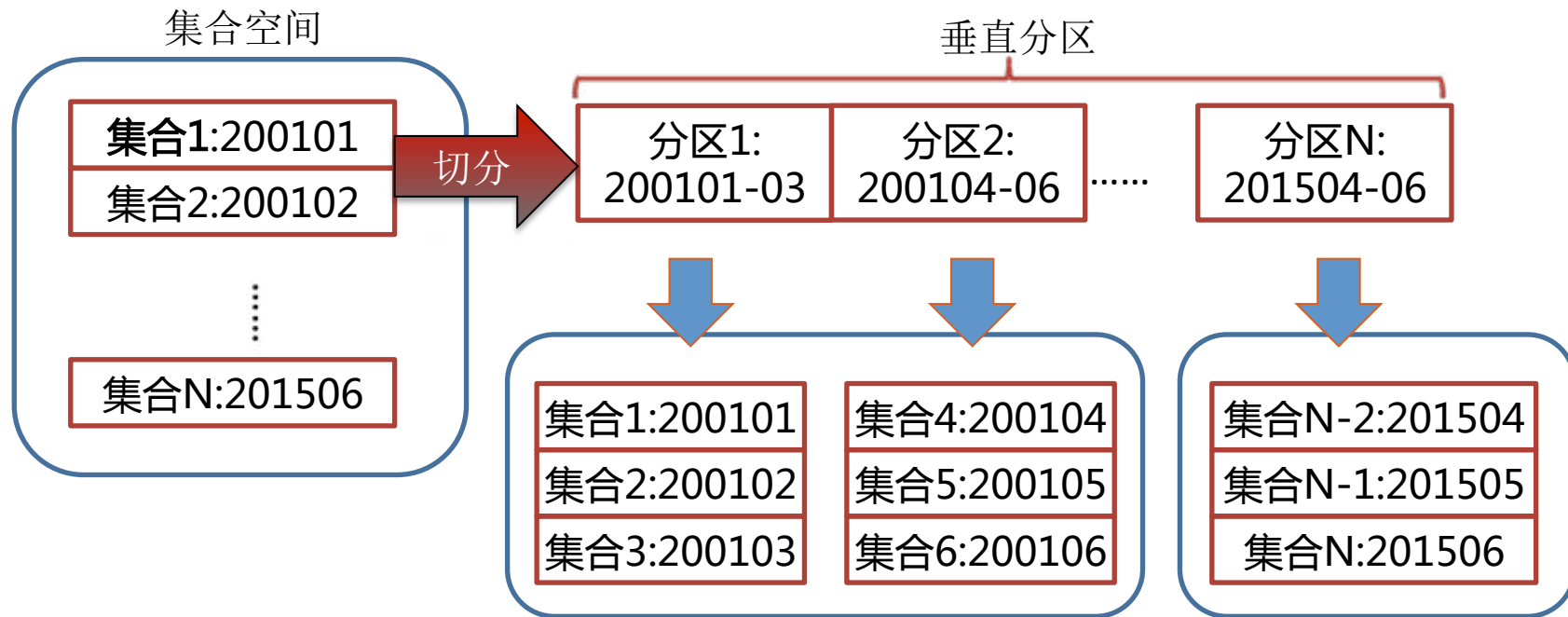
数据加工----数据表预处理

- 基于JSON结构预处理灵活应对表结构变化
- 多表预关联实现实时检索



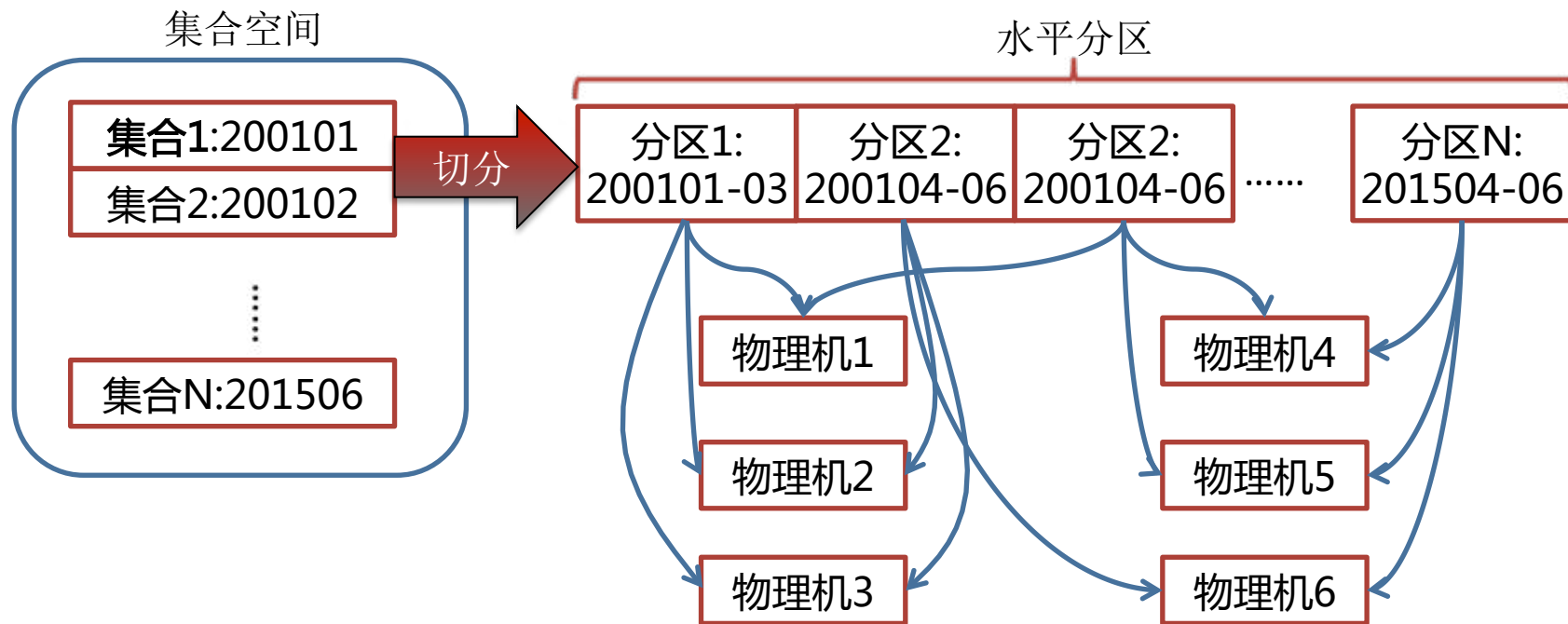
数据加工----时间序模型（垂直分区）

- 用于描述一段时间内连续产生的数据，数据表增加统一时间字段
- 按时间字段切割为多个逻辑分区
- 数据以递增为主，旧数据的热度随着时间的推移递减
- 使用集合分区（主子集合）机制可以轻松应对时间序数据

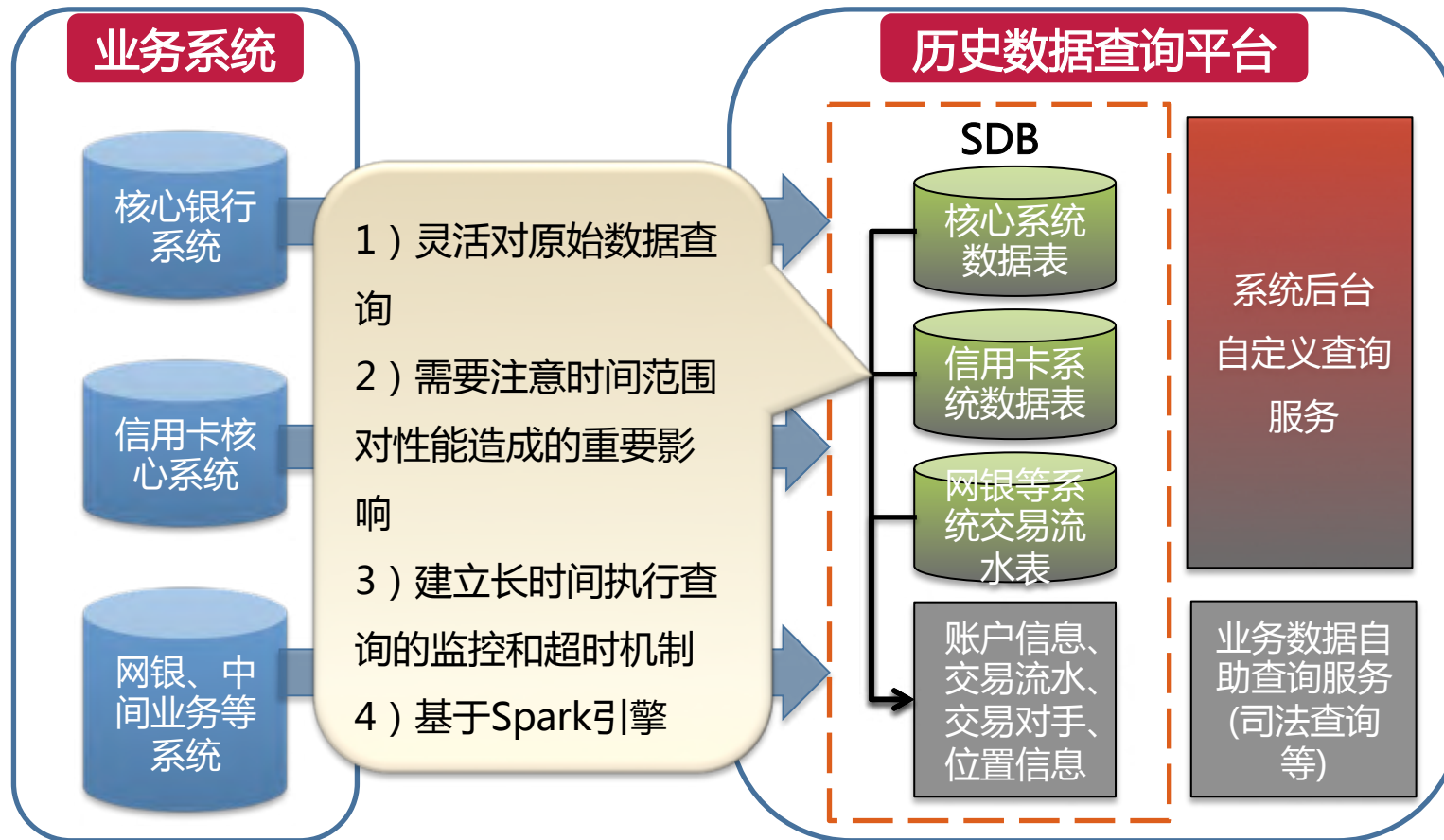


数据加工----数据水平分区模型

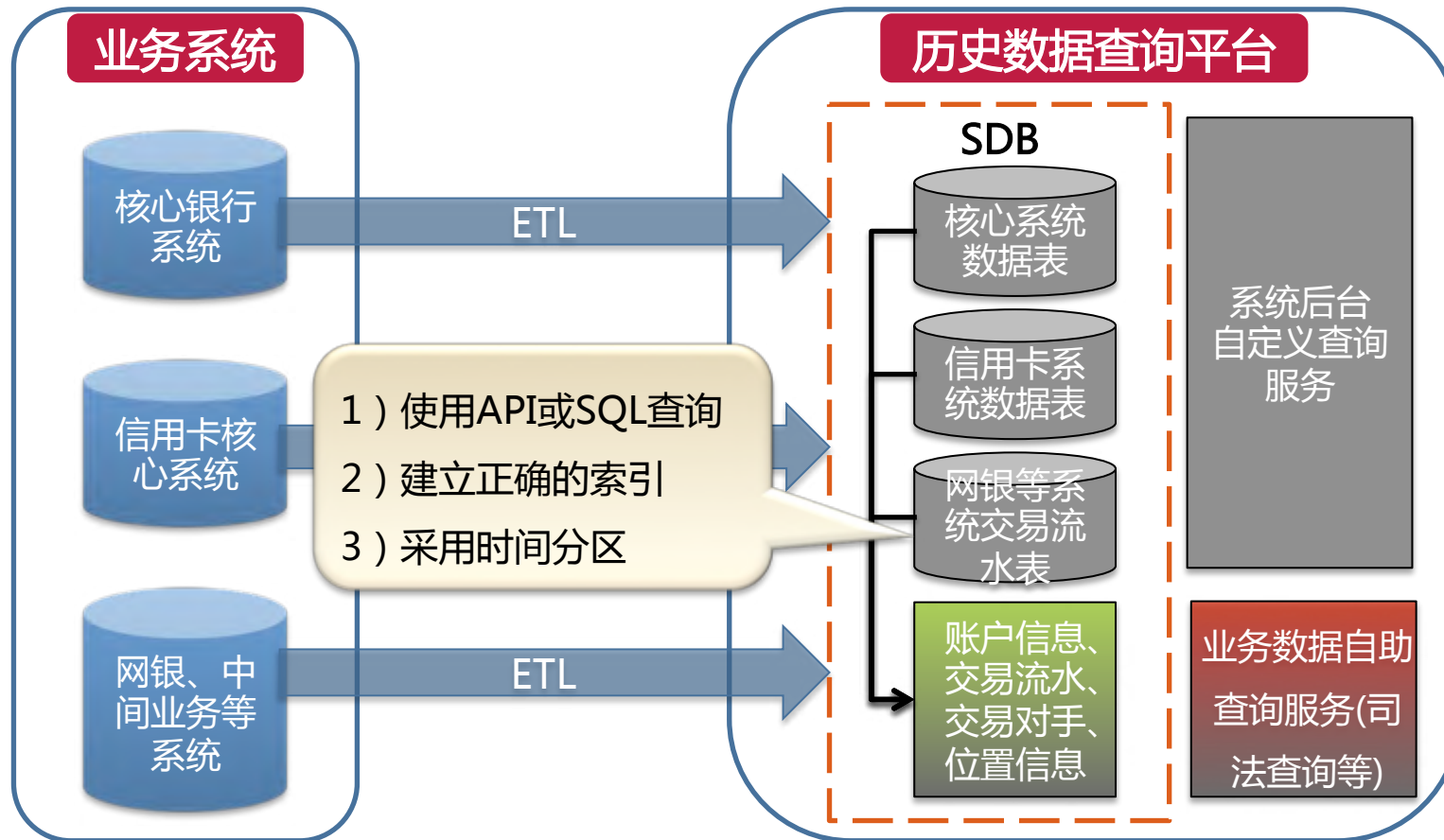
- 数据集合以3台机器为一组，进行数据复制冗余，避免单点故障
- 3台机器作为一个数据分片，多时间段集合分布在不同数据分片
- 跨时间集合查询调用不同数据分片，多台机器同时查询，提高查询效率
- 增加新的数据分片实现在线扩容



数据处理应用场景----系统后台自定义查询服务

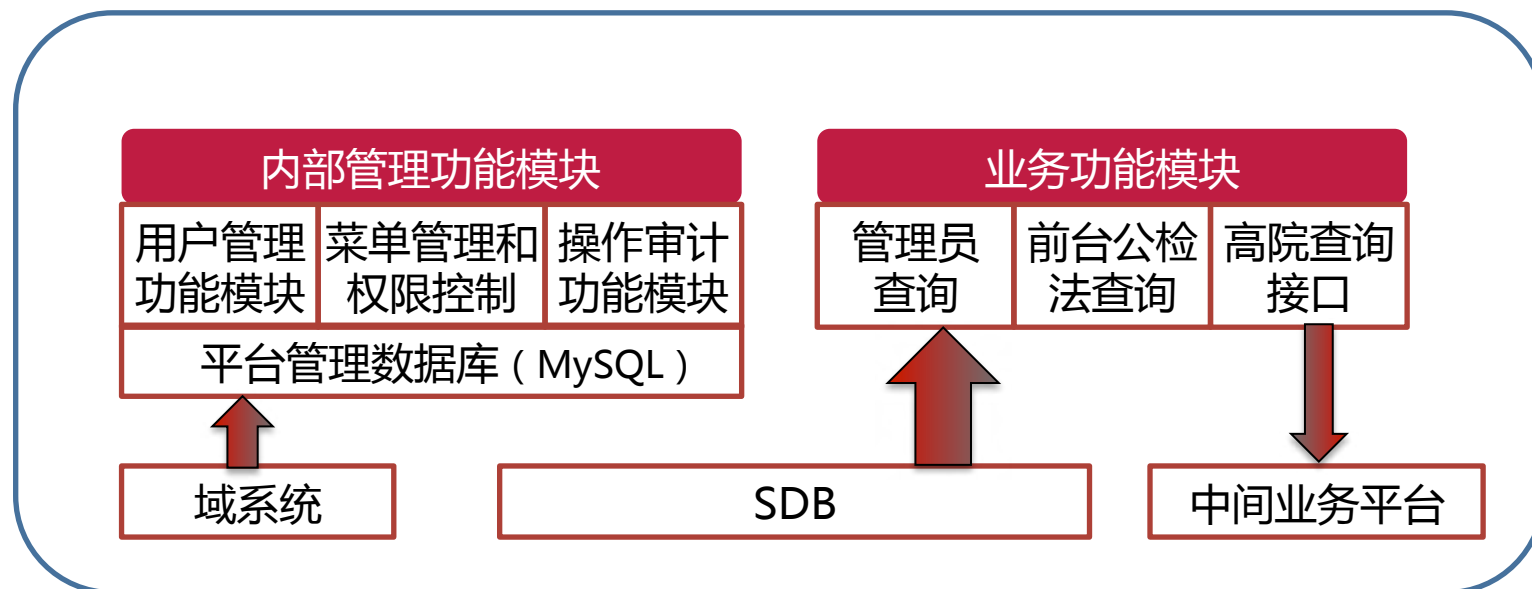


数据处理应用场景----系统后台自定义查询服务



应用平台技术方案

- 搭建统一应用平台，实现前台查询服务功能模块、后台用户管理和安全审计功能模块、接口功能模块
- 部署独立数据库(MySQL)，承担应用平台用户管理、系统配置、安全审计功能
- 采用域用户登录并对接域系统实现用户验证，控制用户权限，不允许修改历史数据
- 通过Birt报表工具实现数据展示和导出



历史数据查询平台计划存储历史数据策略：

- 存储业务系统高频交易流水数据，降低业务交易系统开销，数据保留周期根据各业务高频流水使用情况确定
- 存储大数据平台产生的客户交互明细数据，提供实时接口供第三方系统对客户交互历史的查询
- 存储影像文档，支持后台管理平台调用查询



hr_tech@sequoiadb.com

020-39022971

