大数据平台初探

阿里数据交换平台
强琦

提纲

- 大数据与云计算的关系
- 平台的场景与技术构成
- 阿里巴巴数据交换平台及其关键技术
- 大数据下的技术与商业初探

大数据与云计算的关系

- 大数据
  - 定义：Volume，Variety，Velocity，Value
  - 应用领域：政府，科技，企业，社会。。。
  - 生态：受众，开发者，平台，数据供给
  - 风险：开放与控制，个性化与隐私，。。。
  - 数据场景：见后
  - 技术：云计算，数据仓库，数据开发，数据挖掘，。。。见后

关系

	中心	数据生命周期	轴	描述
云计算	用户&计算	计算周期	纵向	强调计算能力，数据是操作对象;具备工具性；数据私有。
大数据	数据	数据本身	横向	数据作用到计算；具备可运营性，使数据可分享，可加；管理数据是重头。

大数据的数据场景

	时效要求	(对平台要求)吞吐	成本要求	服务	备注
数据服务	毫秒，秒	极高	低	数据展示	各KV们，Hbase们，…
业务(数据)	毫秒，秒	高	高	业务支撑	OLTP(DB)
数据应用	毫秒，秒	高（重）/中（轻）	高	Ad-hoc	多场景(待深度分析)
数据分析	浅层(秒)，深层(分钟)	小	中	在线/离线计算	用来支撑数据决策
深度分析	小时，天	高	低	离线计算(MR,MPI,BSP,STREAMING)	数据挖掘
数据决策	过程	小	高	决策平台(算法平台)	云端sas
工具服务	毫秒，秒	高	高	分词，地理服务等	同步模式或触发器服务(ifttt)

场景的技术说明

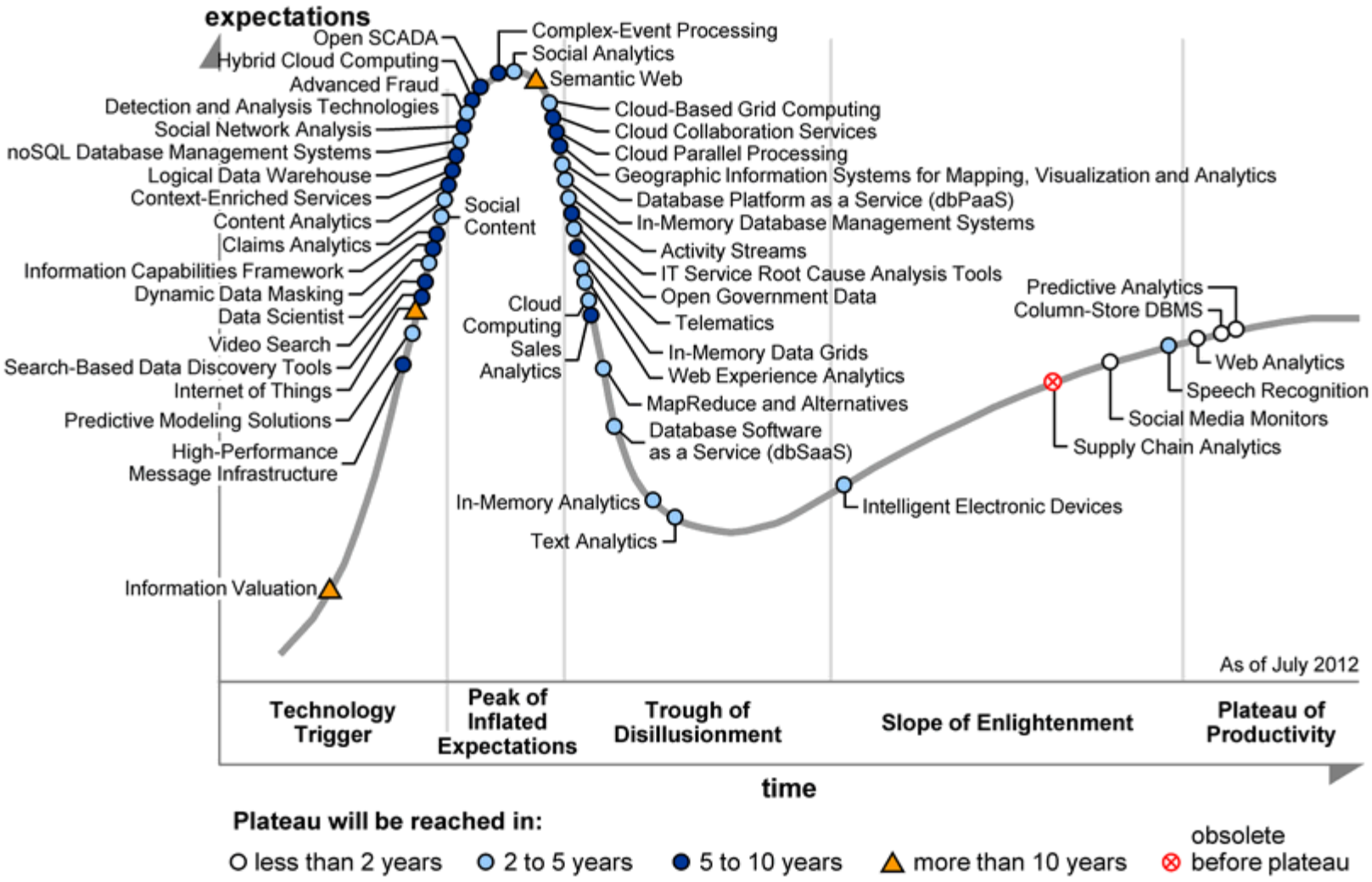
- 数据服务
  - 不同场景（写，读比例）
  - LSM-Tree
- 业务
  - OLTP，关联，事务
  - DB
- 数据应用
  - 全内存，成本敏感，compact，只读数据集
- 数据分析
  - 落地，以吞吐降成本，列存储，in process计算，dremel，impala
- 深度分析
  - 规模取胜，重在吞吐，容错机制(MR,BSP)，错峰超卖，hive（开发成本）
- 备：场景决定技术方案，不同方案服务（云）化挑战不同，high点不同。（yarn?）

技术

- 数据传输
  - 数据库日志，业务系统日志，埋点，批量同步方案，队列
- 存储
  - 块，小，大，流，kv，事务，本地计算，统一的接入层
- 计算
  - BSP（MR，HAMA）, MPI, Streaming, OLTP, OLAP, AD-HOC(real-time computing)，统一的接入层
- 展现
  - 分析可视化，数据可视化

技术

- 开发平台
  - 调度，元数据管理，数据建模，IDE
- 市场
  - 应用市场，数据市场，市场机制
- 数据管理
  - 预警，质量监控，元数据，逻辑，ODS，生命周期
- 开放
  - 安全，审计，计量，监控



- http://www.gartner.com/technology/reprints.do?id=1-1BU465T&ct=120827#h-d2e182

数据交换平台及其关键技术

	阿里	腾讯	百度	Facebook	Google	Amazon
数据规模	★★	★★	★★	★★★	★★★	★★
结构化	★★★	★	★	★★★	★	★★★
关联性	★★	★	★	★★★	★★	★
商业价值	★★★	★★	★	★★	★★	★★★

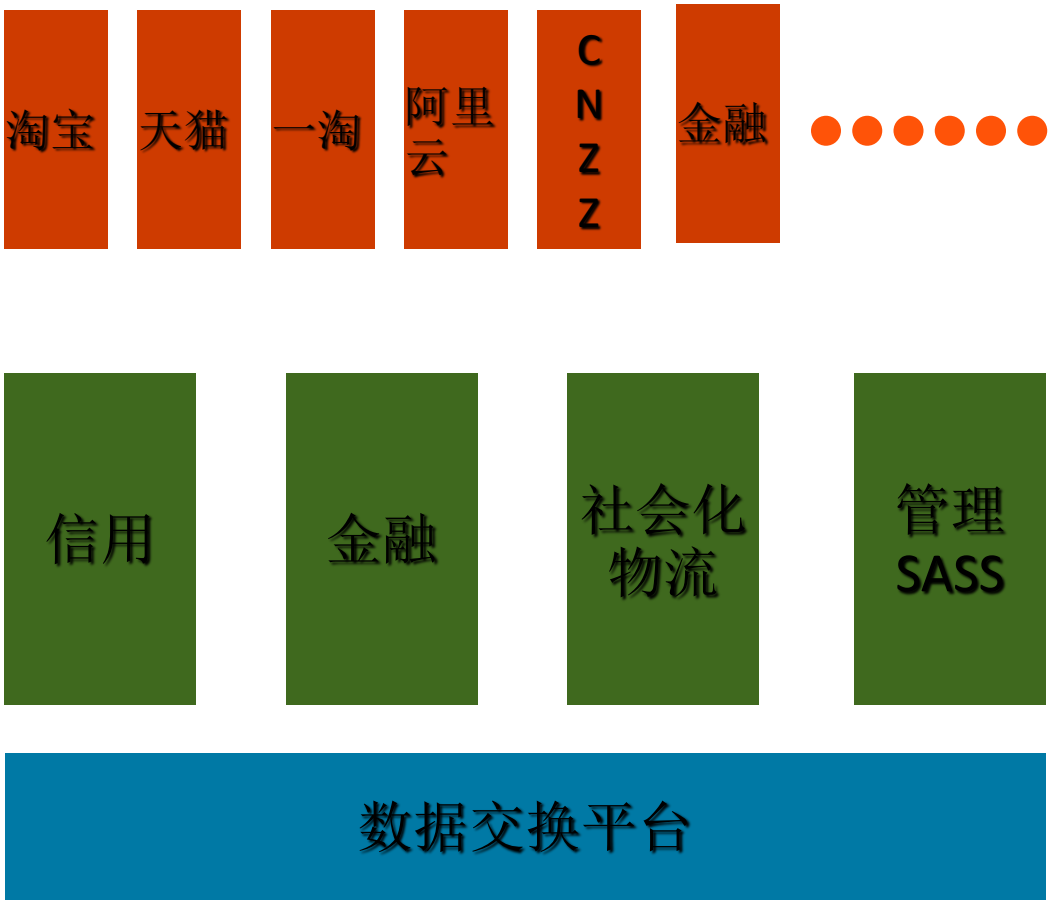
数据交换平台

- 交换
  - 只有平台
  - 只有数据
  - 有进有出
  - 做加法，甚至乘法
  - 数据作为资产的数据银行(存款者，客户，金融服务，银行)
  - 开放

数据交换平台

- 打通、整合集团数据
- 个性化服务
- 构建统一的大数据开发平台

Alibaba Map



关键技术

- ODPS
  - 开放
  - 服务化
  - 离线数据分析服务（MR,MPI,DT…）
- ODS
  - 开放与共享
  - 源头数据质量监控
  - 元数据管理

实时

	时效要求	(对平台要求)吞吐	成本要求	服务	备注
数据服务	毫秒，秒	极高	低	数据展示	各KV们，Hbase们，…
业务(数据)	毫秒，秒	高	高	业务支撑	OLTP(DB)
数据应用	毫秒，秒	高（重）/中（轻）	高	Ad-hoc	多场景(待深度分析)
数据分析	浅层(秒)，深层(分钟)	小	中	在线/离线计算	用来支撑数据决策

实时特点

- Ad-hoc computing：计算不可枚举，计算在query时发生。在线实时。这里的实时侧重query的实时计算。（数据的实时计算）
- Stream computing：计算可枚举，计算在数据发生变化时发生。离线实时。这里的实时侧重实时数据的处理。（实时数据的计算）
- Continuous Computing：计算可加（增量），大数据集的在线复杂实时计算。整体。
- 实时数据的实时计算

实时

- 数据服务
  - 重数据存储，轻计算（coprocessor）
- 业务(数据)
  - OLTP(DB)，增删 改查，事务，范式
- 数据应用
  - Memory, ssd；只读场景；复杂计算； SQL解析、成本优化器、计算引擎、存储引擎。。。

实时

- 深度分析
  - MR。以吞吐见 长，简单有效的容错机制，使其可以得以线性扩展，使错峰超卖成为可能性，以规模取胜，数据传递以跨进程方式(数据)。
- 浅度分析
  - 数据只读（非oltp，所以可对数据结 构做紧凑的设计，以对特定的查询优化）；
  - 吞吐要求不高。（这类应用面向的是运营）；
  - 时效性要求在秒到 分钟级；in-process的计算；列存储
  - 数据量巨大（要求 低成本存储方案）；
  - 非原始数据ODS。一般为加工过的 宽表。
  - Dremel&impala

Garuda

- RT OLAP (Realtime OLAP)
  - Real-Time Objects/Cube/Dimension
- 在线数据分析
  - 访问量低/半结构化/无需定义/低成本
- 在线数据应用
  - 高并发/预定义/高成本初始化/低成本复用

Garuda

- ['gɑ:rudɑ:]
- 印度神话 迦楼罗
- 中国神话 大鹏

- 最重的动物+最快的速度

- 大鹏一日同风起
- 扶摇直上九万里 —李白

场景

- **实时计算定义：**
  - 针对只读数据进行**即时**数据的获取和计算
  - 基于选择和基于扫描的结果集(候选集与全集比例)

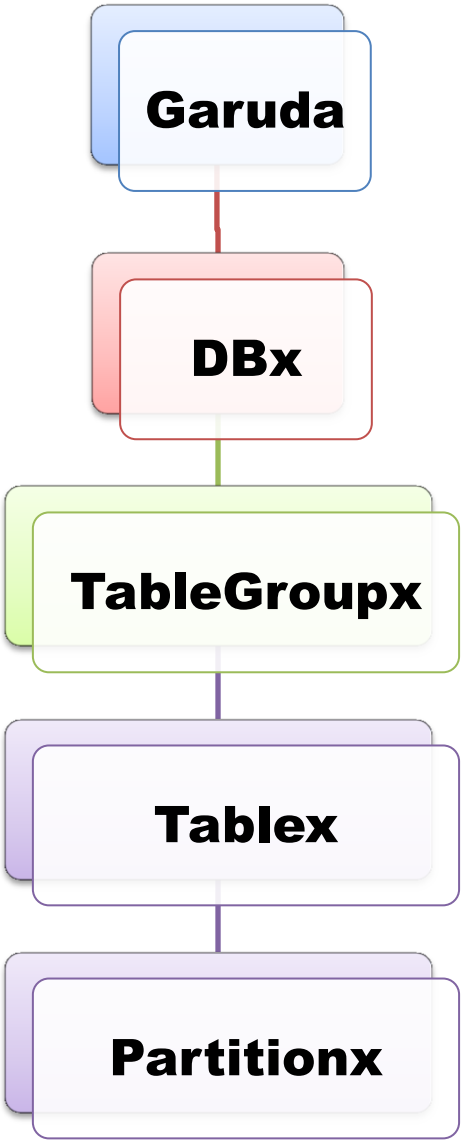
- 相关：
- RTOLAP(Realtime OLAP)
- Grid Computing
- In-memory database

特性

- Fixed/Free Schema（列存储）
- Partition/TableGroup
- 全索引
- 本地计算
- 迭代计算
- 大表Join
- 缓存
- 资源管理调度
- 可用性
- 滚动升级

Partition/TableGroup

- Parition
  - List
  - Range
  - Hash
- TableGroup
  - Join
  - PartitionGroup



选择

- 计算列/索引列(倒置)
  - 计算列 @ memory
  - 索引列 @ disk
- 索引
  - Hash
  - B+Tree
  - Skiplist
  - Bitmap
- 倒排
- 压缩
  - String?
  - PForDelta(11%)

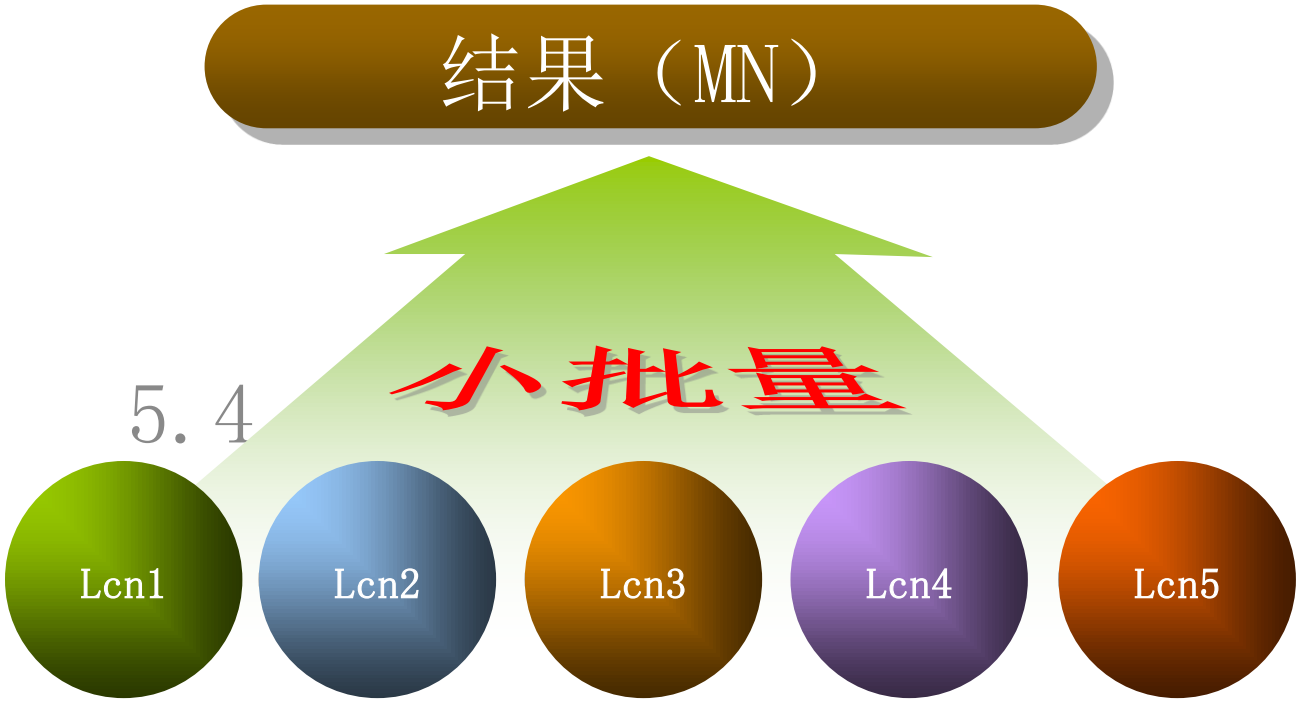
Index array(abstract)	
tree<T,int[]>	SSD
skiplist<T,int[]>	SSD
hashmap<T,int[]>	SSD
unique<T,int>	memory

本地计算

- mergeNode：
  - ✓ SQL解析
  - ✓ 路由分发
  - ✓ 结果缓存合并
  - ✓ 迭代计算

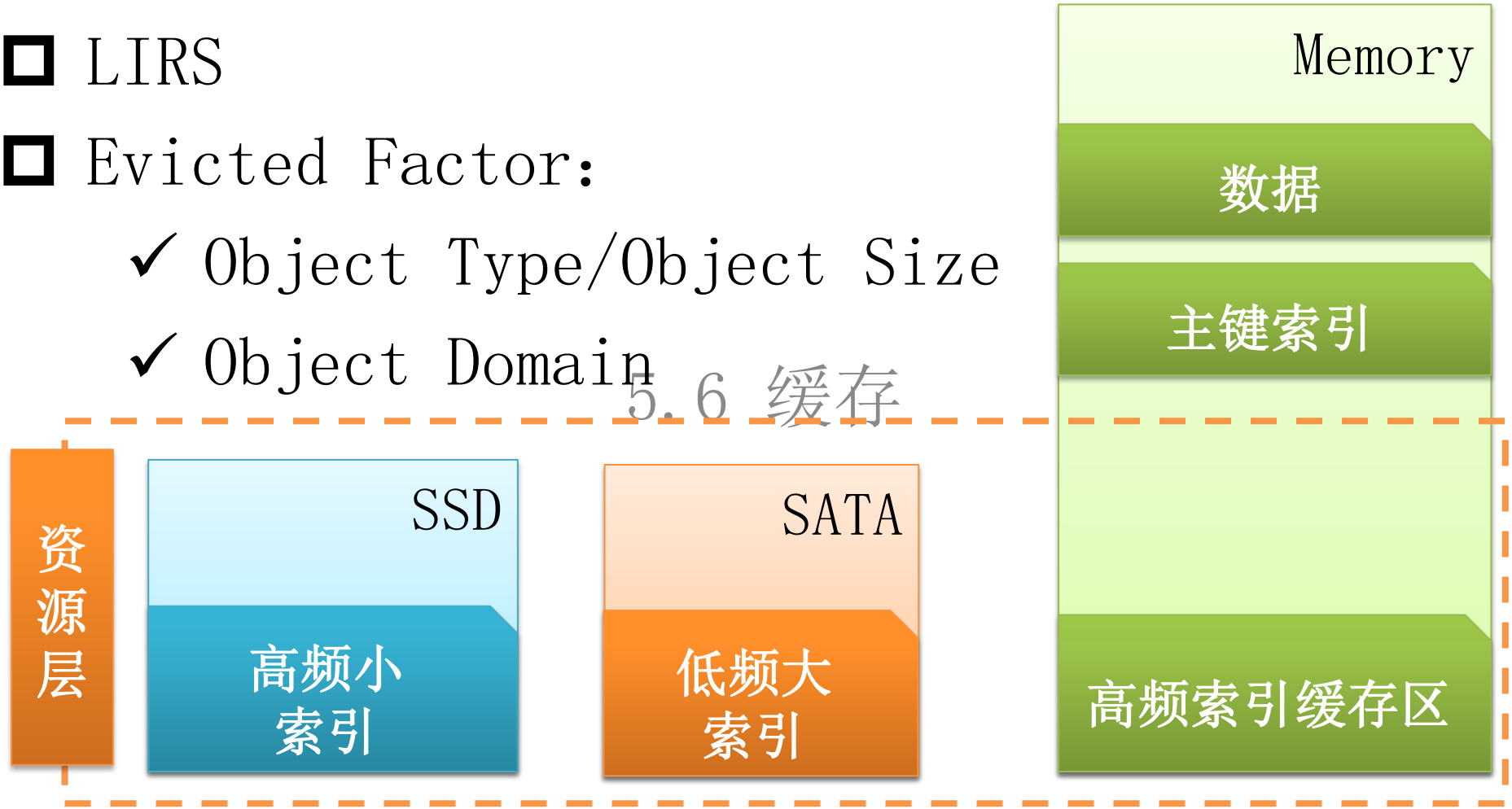
- Localnode
  - ✓ SQL解析
  - ✓ 索引查找
  - ✓ 计算

- **带宽？**



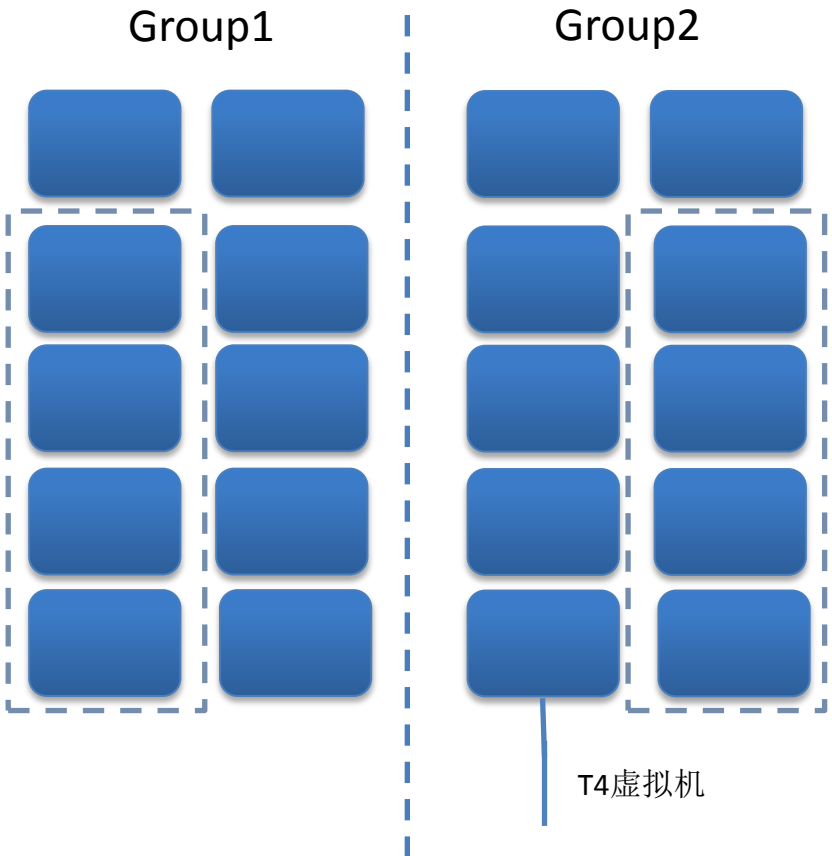
缓存

- **本地节点缓存：**
  - LIRS
  - Evicted Factor：
    - ✓ Object Type/Object Size
    - ✓ Object Domain



调度

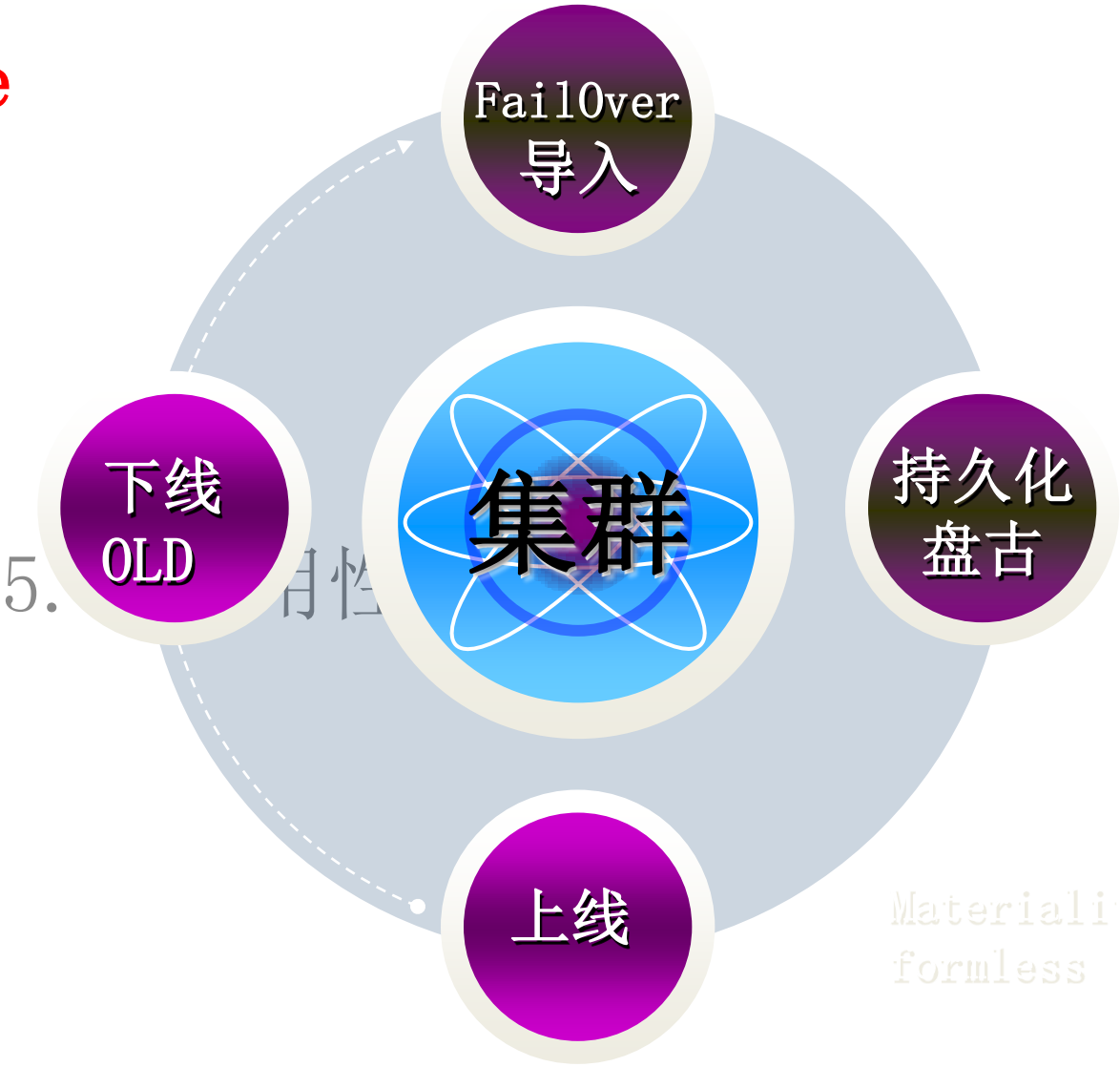
- 动态规划算法
- Monitor 服务器分布式锁（主/备）
- 可运维
- 参数：
  - 可用内存、可用磁盘（Buffer阈值）
  - 每个表占用的内存、磁盘
  - 最小可用实例数
  - 最小Failover机器数
  - 每个分区最小可用份数
  - 每个表最多保留分区数
  - 表组信息
  - 虚拟机组
  - 滚动升级
  - 整理上线
  - ….



可用性

- Failover Rotate
- 资源虚拟化(T4)

- Heartbeat
- 双机房
- 任务分布式锁
- 任务持久化
- 任务跟踪JobID
- 执行时间监控



重点

- 夯实基础
  - 存储引擎性能，成本，稳定性，运维
- 架构梳理
  - 分布式调度、SQL解析、成本优化器、计算引擎
  - 存储引擎：Memory行存储引擎、长周期引擎、检索引擎、列存储引擎等
  - 离线build&load
- 业务功能

Stream computing特点

- 流（stream）:由业务产生的有向（渠道）无界的数据流。
  - 不可控：到达时机，相关数据顺序，质量（残缺），only once，规模，上游不可控（业务改变，渠道）
  - 时效性要求：容错方案，体系架构
- 处理粒度最小：对架构影响决定性
- 处理算子对全局状态影响不同：有状态，无状态；幂等，顺序相关（偏序，全序）
- （多）输出性质不同：action，state(大多数节点为commit点，少数为commit点)

业务

- 淘宝双11直播间
  - 100亿数据
  - 多张大表join
  - 时序
  - 准确与效率
  - 消重
  - 可运营
- 移动

三个层次

- SQL
  - CREATE STREAM stream_name
  - CREATE DIM TABLE dim_name
  - CREATE CACHE TABLE AS SELECT [ALL|[col1[udf(col2),…]]] from DIMTABLE WHERE conditions WITH(cache_parameter=value[,……])
  - CREATE RESULT TABLE result_name
  - CREATE TMP TABLE tmp_tablename
  - SELECT [* | expression] [[AS] output_name] [, ...] [FROM from_item [alias] with [window(…)] [[left|full outer] join ...] on join_condition] [WHERE condition] [GROUP BY [group_expr [, ...]]|] [[UNION ALL] select] [TOP N by expression[ASC|DESC] [,……]] With(select_parameter=value[,……])
  - UDF，UDAF，UDTF

三个层次

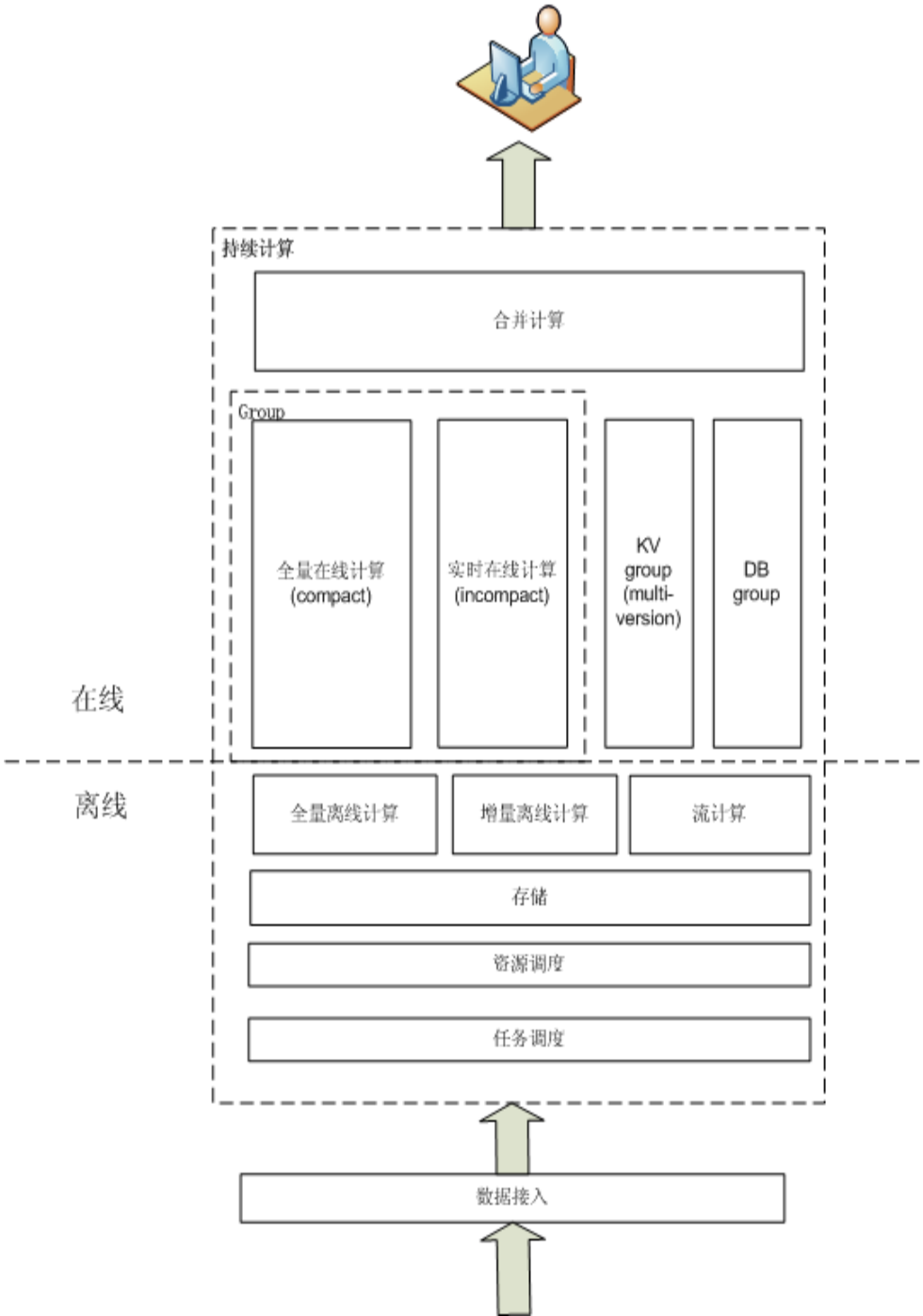
- 语义层
  - Local function(udf, udtf, udaf)
  - Shuffle
  - Aggregate
  - Merge
- sourceCode
  - 复用组件（存储层）
  - Join，topk。。。

持续计算

	批量	实时
冲击	Volume	Velocity
资源有利	累积	分摊
业务有利	覆盖	增量
延迟	高	低
成本	高	高
容错	相对简单	复杂
现有资源	多	少
计算	简单	复杂

持续计算

- Continuous Computing：计算可加（增量），大数据集的在线复杂实时计算。实时数据的实时计算。

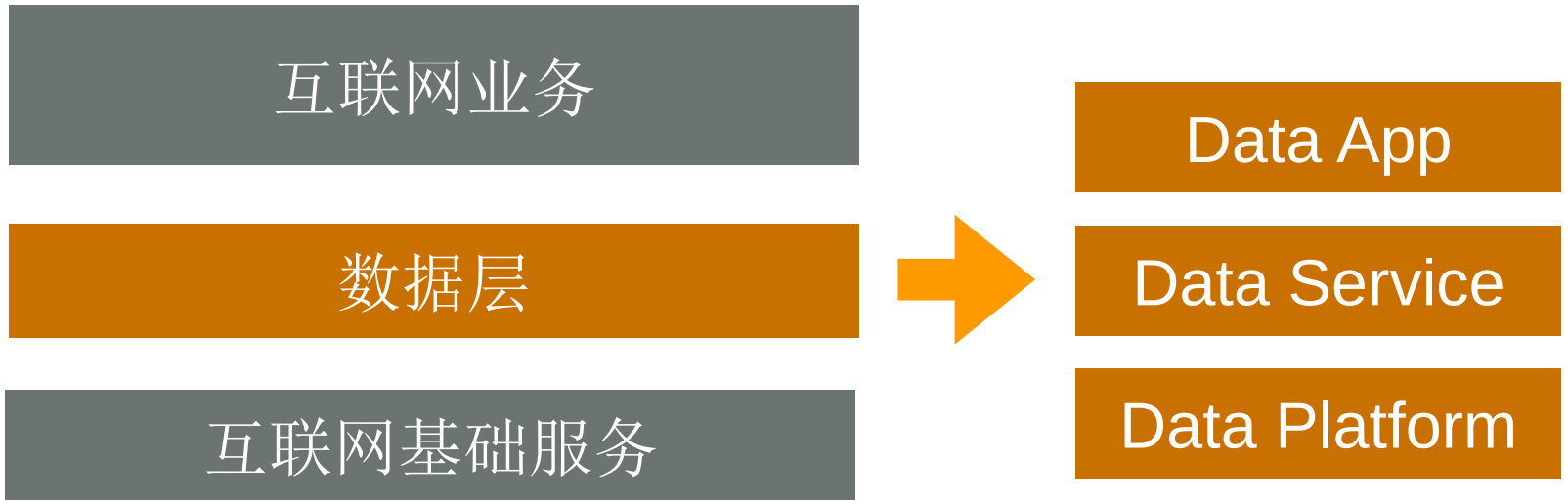


目标

- 一个开发IDE，一个入口
- 限制
  - 可加性(误差可控)
  - 局部无复杂操作
  - 局部节点无舍弃操作
  - 幂等，非幂等要同步。
  - 同构数据

- 场景
  - Compact数据集
  - （近似）增量计算
  - Read only
  - 高性能存储计算

大数据下的技术与商业初探

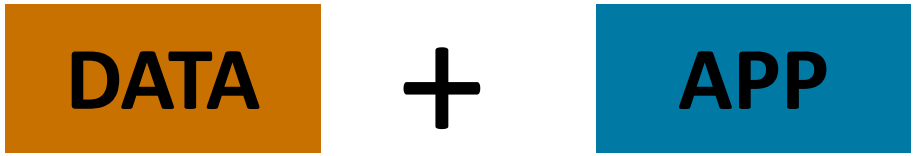


Redshift

已有Data-App

淘宝魔方　淘宝指数　个性化　金融　等待接入

数据交换平台


DATA
APP

相信生态系统的力量
相信开放的力量
Thanks

http://www.alidata.org/

和仲