淘宝网
Taobao.com

在线计算系统原理和实践

阿里巴巴集团-离哲(占超群)
@flyinweb

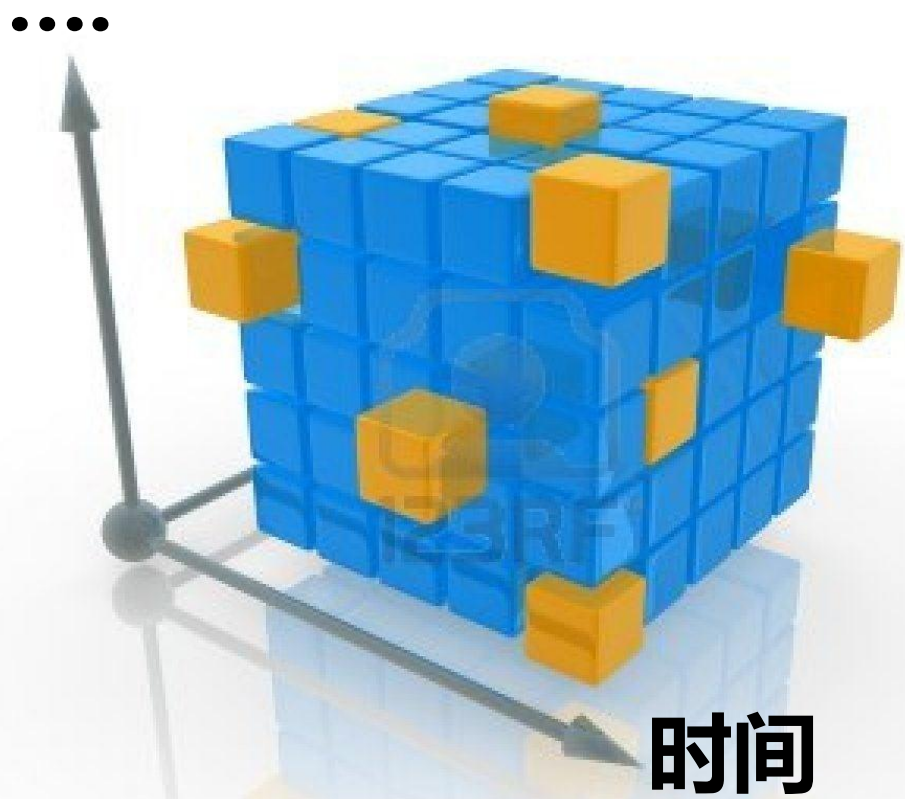
提纲

1. 场景 & 定义
2. 原理
3. 应用案例
4. 其它方案

17:01

1 场景

淘宝网 Taobao.com

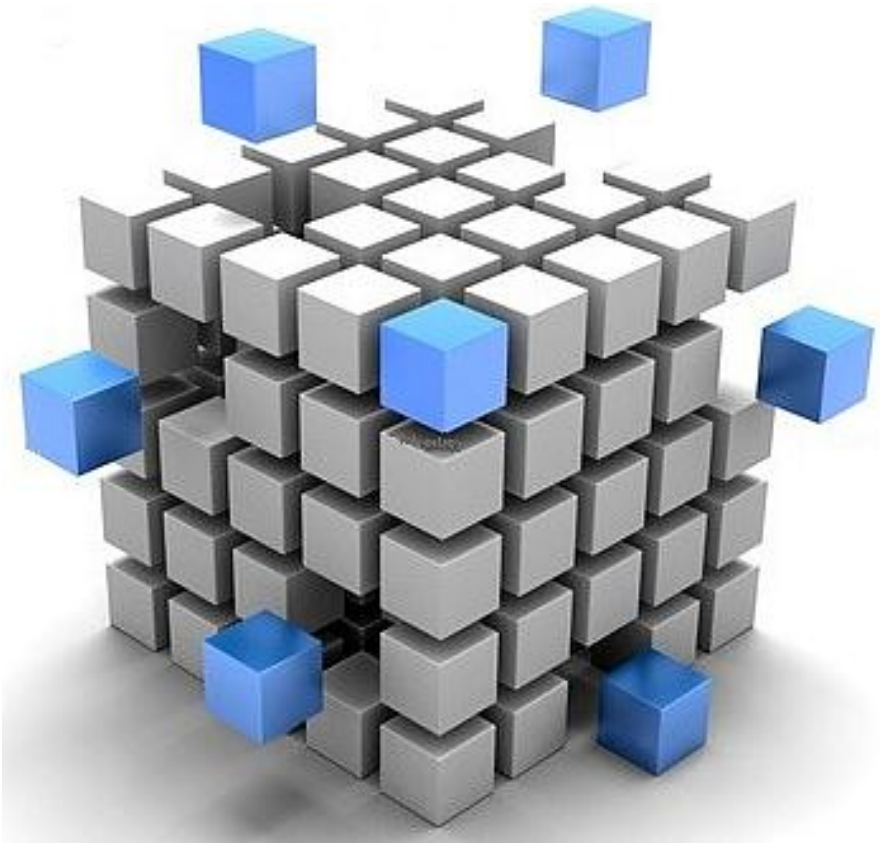


时间
省份
地市
性别
年龄

PV/UV
热门频道/栏目

17:01

1 场景



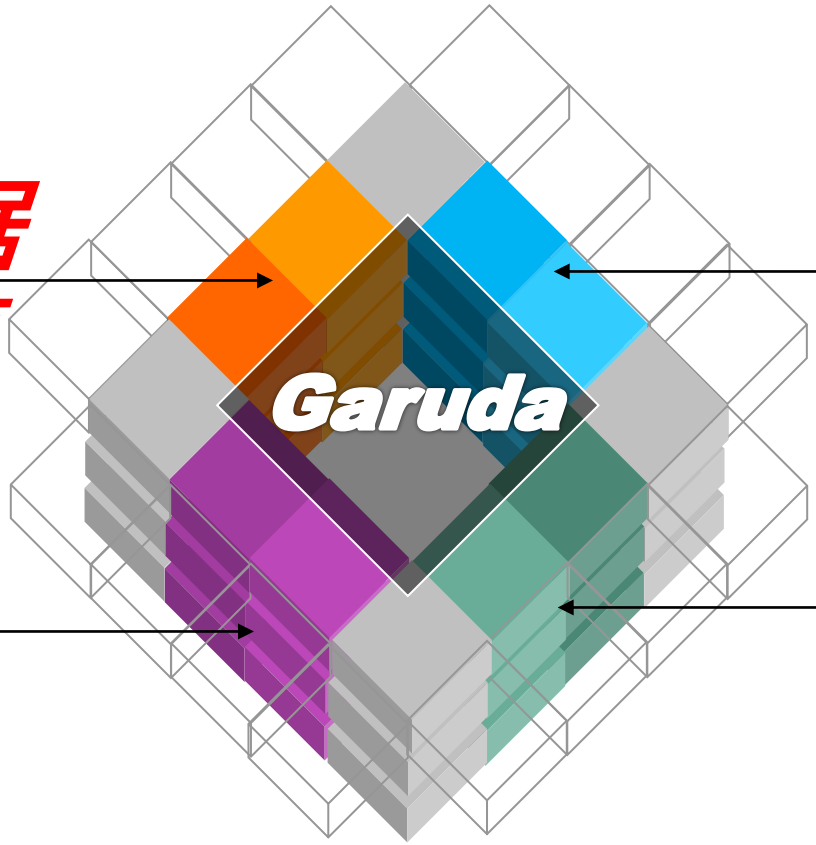
时间
省份
地市
性别
年龄
偏好

热门商品/文章
热门用户

17:01

海量数据
规则灵活
无法预算

1 场景 @ 淘宝



- 海量数据
- 无法预算

- SQL
- Schema Free

- 高并发
- 高可用

- 低延时
- 计算精确

分布式/全索引

1 在线计算定义

- **在线计算定义（相对于离线计算）：**

  针对**只读**数据进行**即时**数据的获取和计算

- 在线数据分析
  - 访问量低/半结构化/无需定义/低成本
  - 代表:
    - Google Big Query（Dremel)/Cloudera Impala（Trevni）
- 在线数据应用
  - 高并发/预定义/高成本初始化/低成本复用

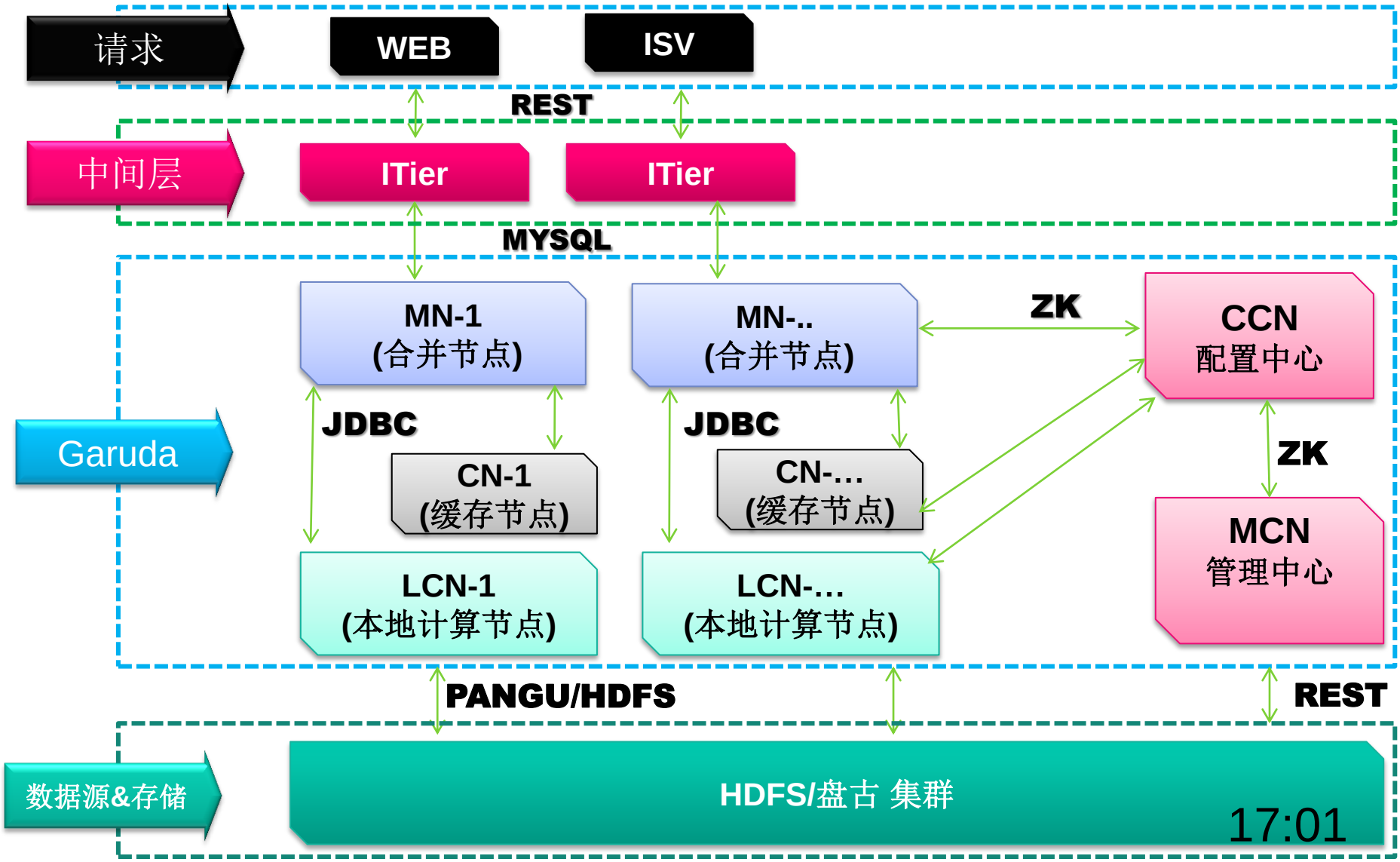
17:01

2 在线计算系统Garuda原理

- 架构
- 核心要点
- 关键特性解读

17:01

2.1 架构总览


请求
WEB
ISV
REST
中间层
ITier
ITier
MYSQL
MN-1
(合并节点)
MN-..
(合并节点)
ZK
CCN
配置中心
JDBC
JDBC
Garuda
CN-1
(缓存节点)
CN-…
(缓存节点)
ZK
MCN
管理中心
LCN-1
(本地计算节点)
LCN-…
(本地计算节点)
PANGU/HDFS
REST
数据源&存储
HDFS/盘古 集群

2.2 核心要点

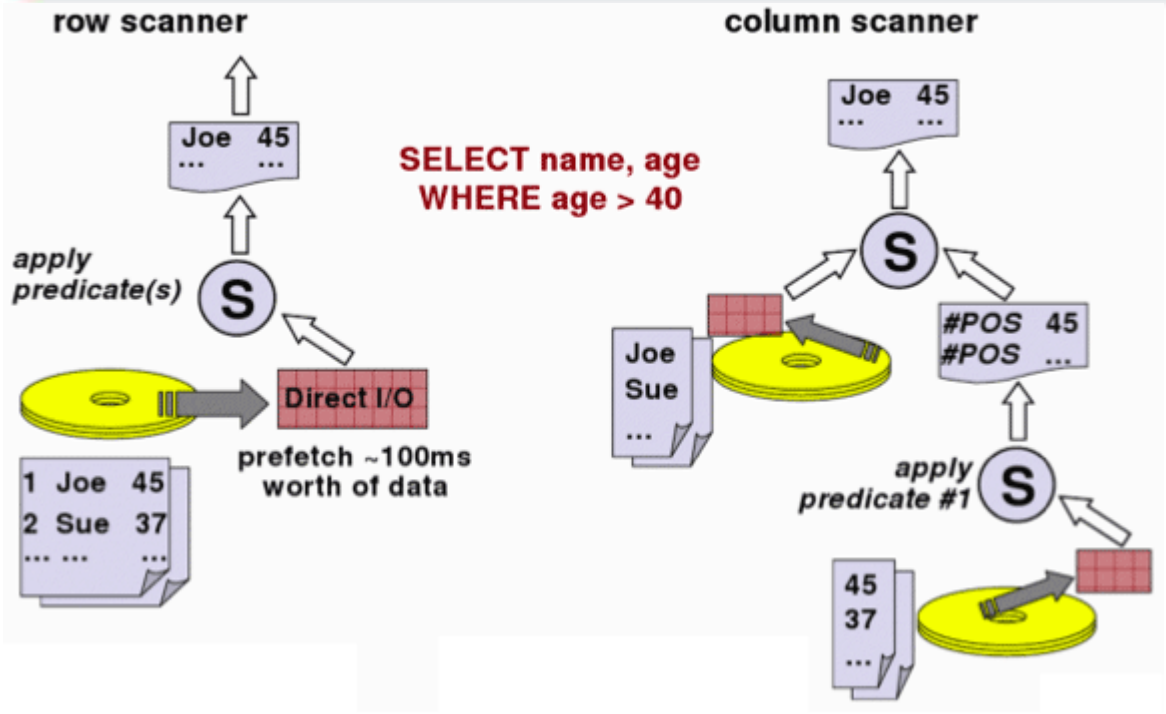
随机访问	• 内存(稳定高频数据) • SSD
业务分区 本地计算	• 避免网络交互 • 降低时延 • 提高并发
压缩	• 减少内存占用 • 降低磁盘IO
缓存	• 提高并发
Failover 加载	• 服务稳定性 • 提高可用性

2.3 关键特性

- 列存储
- 分区
- 本地计算
- 大表Join
- 缓存
- 压缩
- 资源调度 &可用性

17:01

2.3.1 列存储



存储

- 更少扫描(只选择需要的列)
- 顺序访问

压缩

- 内存压缩 38%
- 磁盘压缩 6%

2.3.2 分区

分区
- 本地计算
- 减少网络交互

分组
- 便于Join

Garuda

DBx

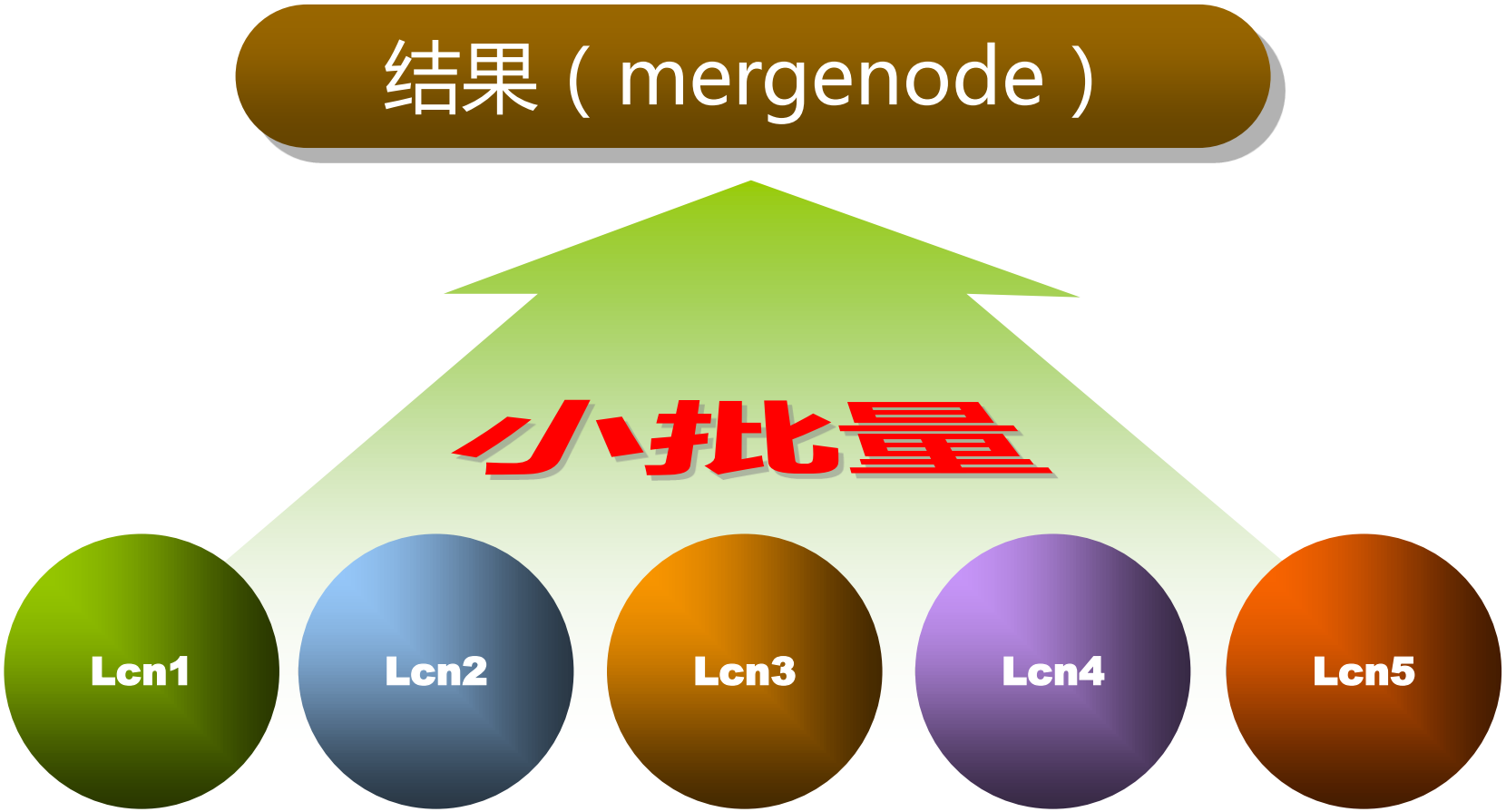
TableGroupx

Tablex

Partitionx

17:01

2.3.3 本地计算



2.3.3 本地计算-Distinct

- Count(distinct user_id)
- ....

1	0	1	0	1	0	0	0	0	1
1	2	3	4	5	6	7	8	9	10

步骤：

1）user_id 全局编码保持有序

2）计算每个分区bitset

3）bitset压缩

4）bitset求交 || 求和

2.3.3 本地计算-top k

- Select type_id, count(id) from t group by type_id order by count(id) desc limit 10
- **精确度换latency、concurrent**

Num of top-k	Size of output	Num of guaranteed	Guarantee	Precision
10	10	10	1	1
25	25	20	0.8	0.84
50	50	46	0.92	0.98
75	75	72	0.96	0.987
100	100	98	0.98	0.99

2.3.3 本地计算-其它

- **AVG**
  - Sum/Count
- **Order by rand() limit 10**
  - rand(cardinality*limit+2)

2.3.4 大表Join

- **TableGroup：**

  分区Join

- **附属表（支持M:N）：**

  存储：主表内存位置+
  自身内存位置

  加载：主表增加虚拟列

- **附加索引（支持M:N）：**

  存储：主表内存位置

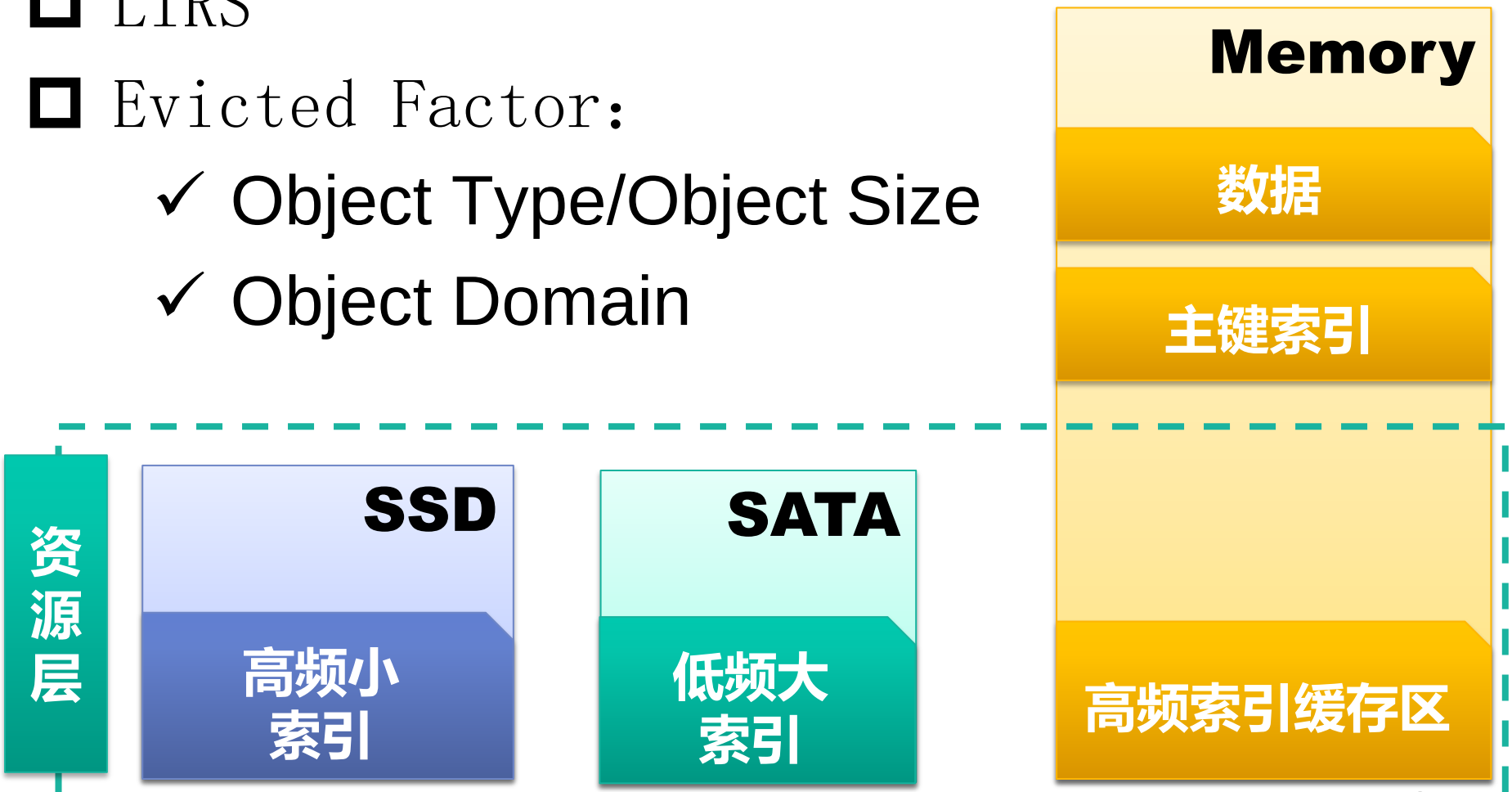
  只能用来定位和count

2.3.4 大表Join

17:01

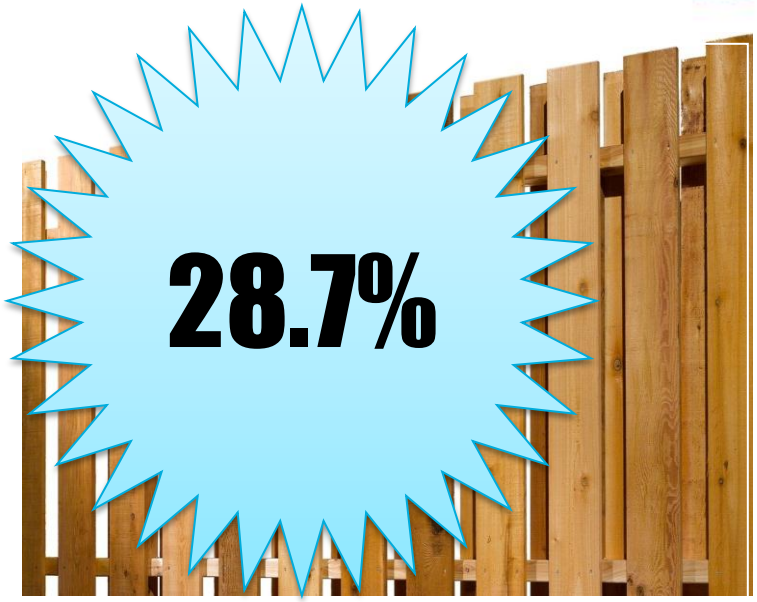
2.3.5 缓存

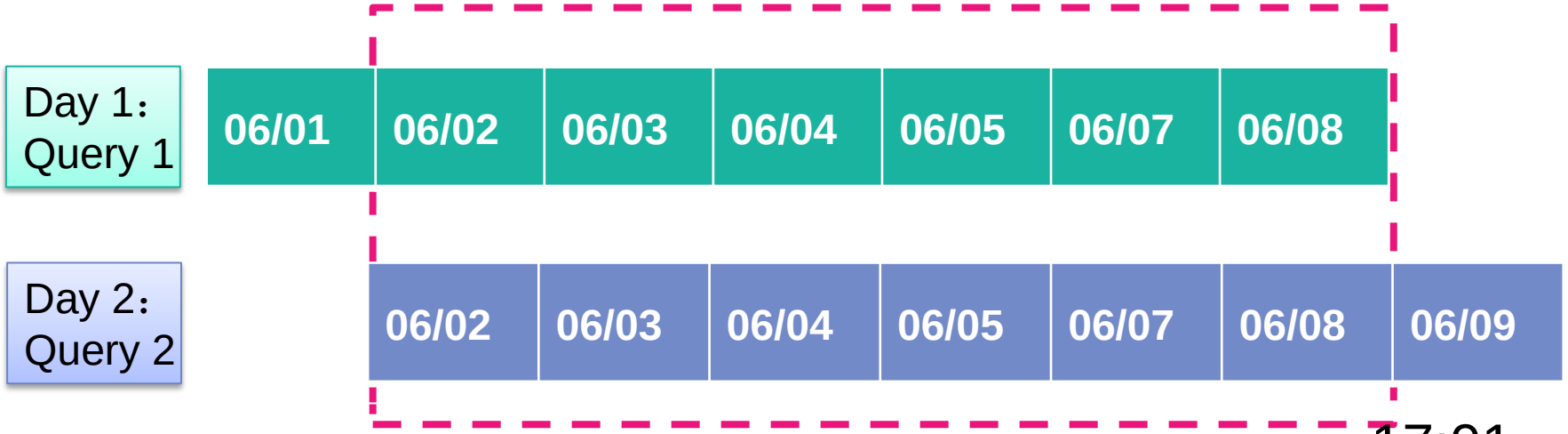
- 本地节点缓存：
  - LIRS
  - Evicted Factor：
    - ✓ Object Type/Object Size
    - ✓ Object Domain



2.3.5 缓存

- Master节点缓存：
  - LIRS
  - SQL cardinality
  - Partition result

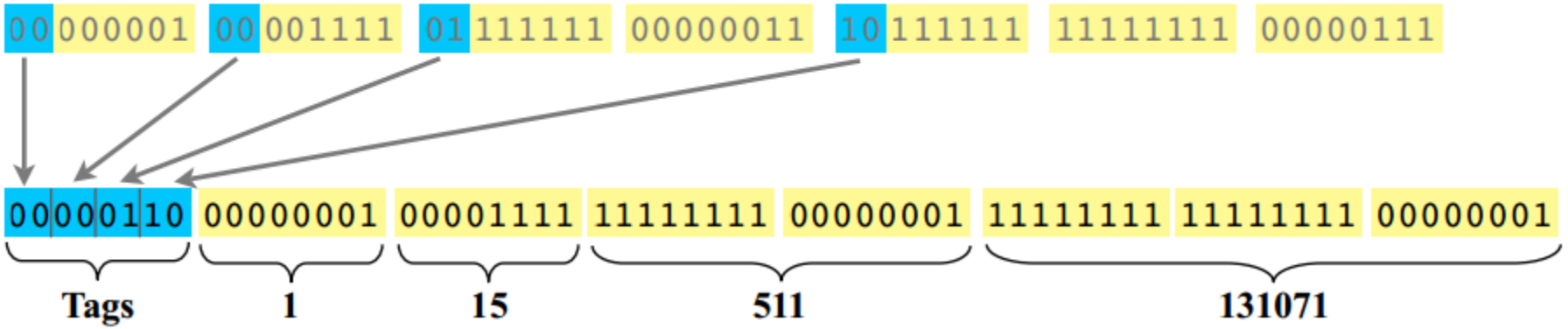




2.3.6 压缩

- 内存压缩
  - Group Varint Encoding (压缩比:37.5%)
- 磁盘压缩
  - Lz4 (2.08%,330MB/s,915MB/s)

- Idea: encode groups of 4 values in 5-17 bytes
  - Pull out 4 2-bit binary lengths into single byte prefix



17:01

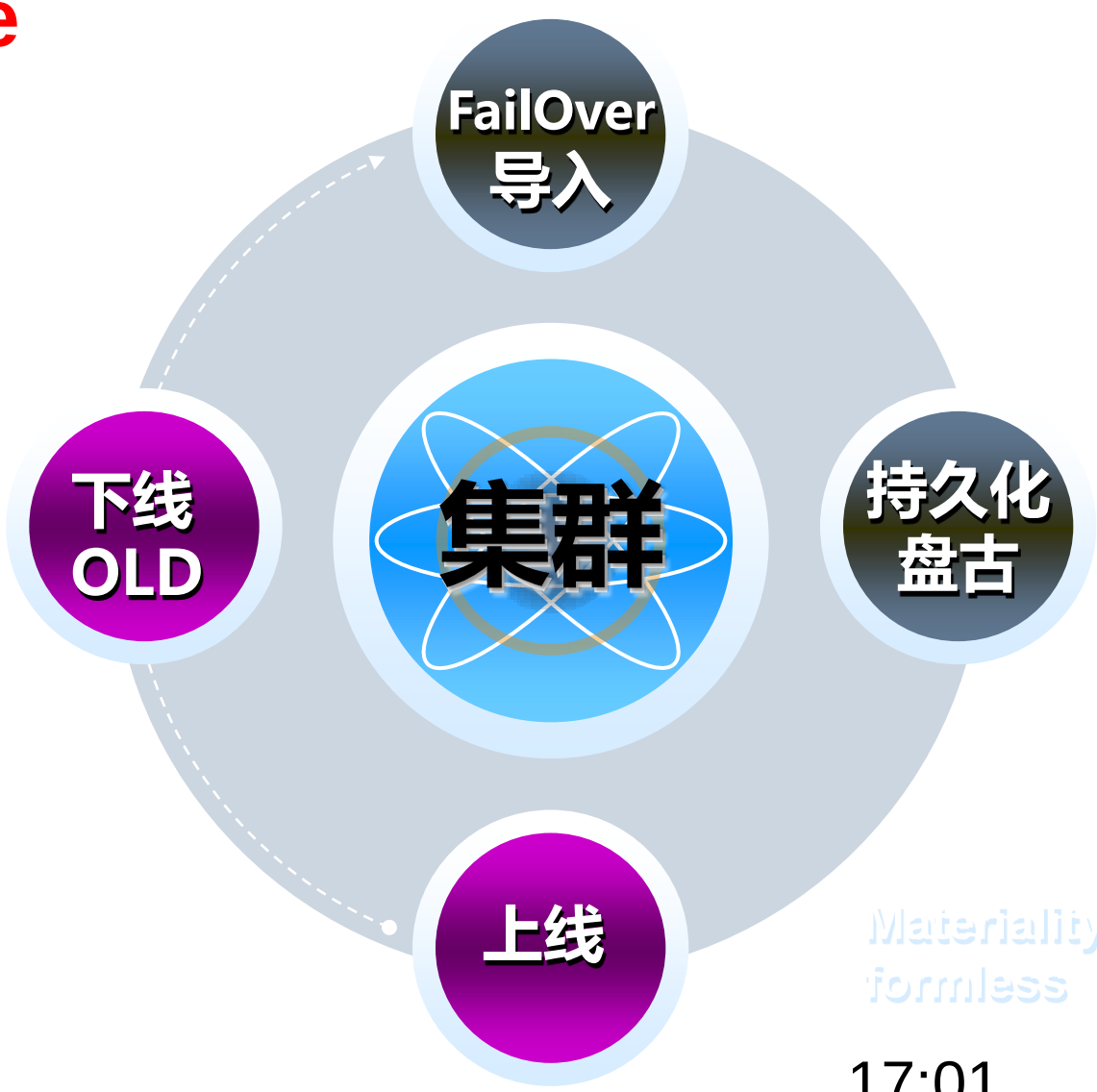
2.3.7 资源管理调度

- 动态规划算法
- Monitor 服务器分布式锁（主/备）
- 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（在线上集群）
  - 每个表最多保留分区数（Rotate）
  - 超时设置(上线/下线/导入 超时)
  - 表组信息
  - 虚拟机组
  - ….

2.3.8 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



17:01

3 应用案例

淘宝网 Taobao.com

淘宝指数 + Taobao DataCube 数据魔方

- ✓ 交易信息+搜索信息 > 360亿
- ✓ Join表三张

- ✓ QPS > 300
- ✓ RT<100ms
- ✓ AVG Compute Row/Per Query > 1亿

17:01

3 应用案例

淘宝网 Taobao.com

最近30天(2012-11-05 至 2012-12-04) 购买 羊毛衫 的主流人群是 "女性白领，中等消费的初级买家，年龄在25岁-29岁，江苏、浙江和广东共占29%"

人群定位

性别

男 27%　　女 73%

消费层级

低	偏低	中等	偏高	高
1%	8%	57%	30%	3%

买家等级

新手买家	初级买家	中等买家	资深买家	骨灰级买家
27%	48%	15%	8%	3%

身份

白领	学生
53%	7%

年龄

0	25	30	35	40	50	60
18%	25%	22%	15%	17%	3%	0%

相关品牌　相关商品　**相关属性**　+展开宝贝

名称	销量趋势	热销指数	倾向指数	人群均价
1 通勤 纯色 修身型		208,420	100	￥135
2 通勤 修身型		40,659	24	￥148
3 通勤 其它图案 修身型		31,705	18	￥168
4 甜美 纯色 修身型		30,678	16	￥146
5 纯色 修身型		18,172	12	￥127
6 通勤 条纹 修身型		25,055	11	￥136
7 通勤 纯色 直筒型		34,549	9	￥142

17:01

3 应用案例

淘宝网 Taobao.com

整体情况

行业: 手机　　属性: 直板　　最近30天

运行内存RAM
64M　2G以上　2G　1G　768M

机身内存ROM
64G以上　32G　256M　512M　128M　64G　1.5G　16g
4G　8g　2G　1G　768M

厚度
超薄(小于9mm)　薄(9mm~1cm)　普通(大于1cm)

触摸屏
电容式触摸屏　电阻式触摸屏

整体情况　热销店铺排行　热

成交概况

热销指数
206,291

成交商品数
466,358

数据图

热销指数　成交人数　成交笔数　成交商品数

77,000
75,000
73,000
71,000

17:01

3 应用案例

- **XX用户系统**

✓ 用户及关联信息>4,000,000,000

✓ Join表 2 张

✓ 列数>1000

✓ RT < 7s

✓ AVG Scan Row/Per Query > 4亿

4 其它方案

redis 1.4

redis 2.6

APACHE HBASE

17:01

4.1 Redis+Tokyo cabinet

Tokyo Cabinet 8192PiB

redis 1.4

Key	IDSet(Blob)
手机	1,3....

ID	IDData
1	Price,pay_num…

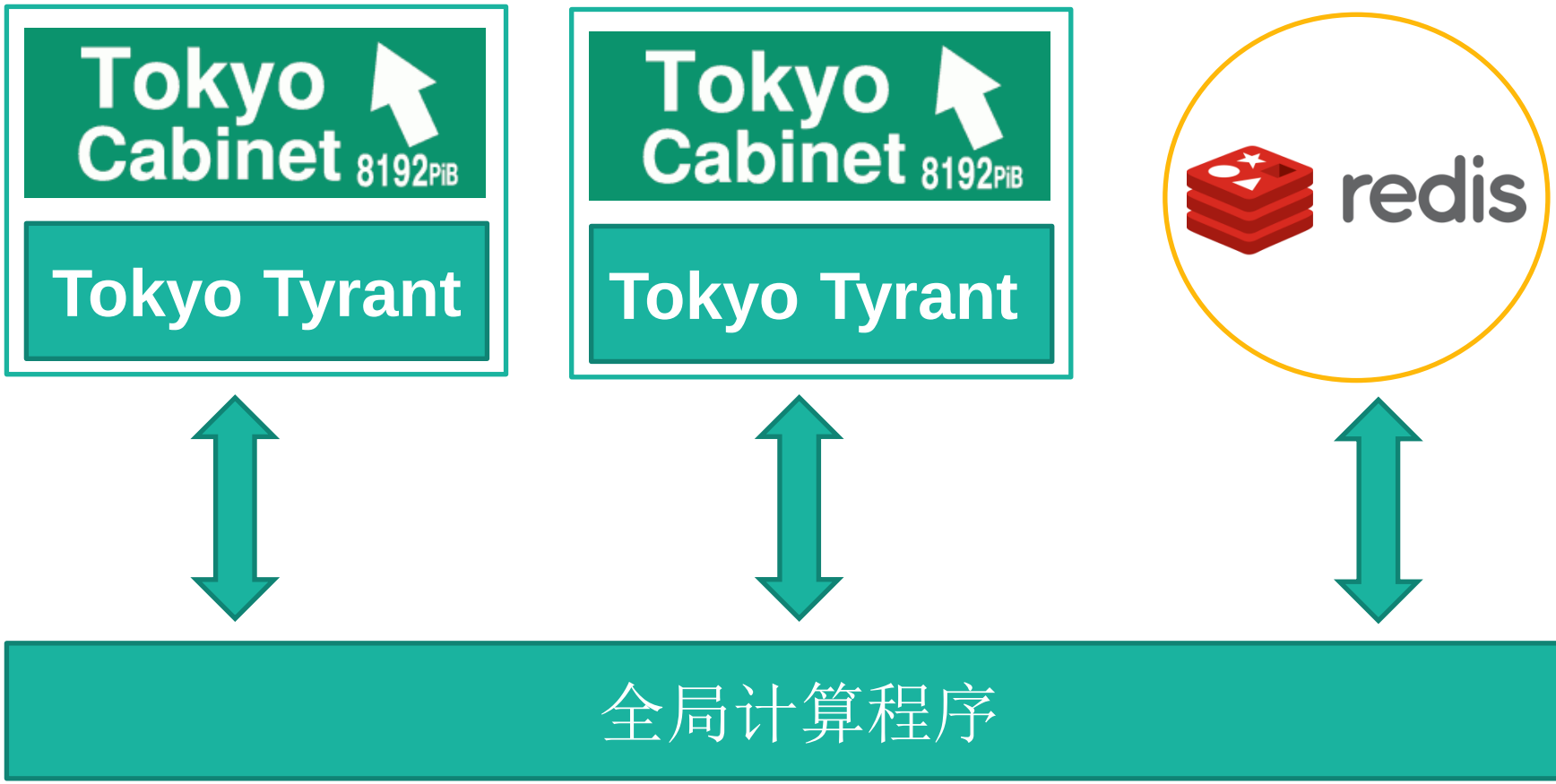
获取索引(Tokyo cabinet)

批量获取Data(Redis)

并行计算

17:01

4.1 Redis+Tokyo cabinet



17:01

4.2 Redis 2.6

index	Key	IDSet
	手机	1,3....

data	ID	hashtable
	1	Price:2,pay_num:3

每个Redis Server用lua局部计算

合并

输出

17:01

4.2 Redis 2.6

--求交索引key

```
local ids = redis.call('SINTER', 'indexKey1', 'indexKey2', 'indexKey3');
local result = {};
-- loop  data
for index, value in ipairs(ids) do
    local countCol1= count..
    local sumCol1= sum..
end
-- error first for each checking
return {false,countCol1,sumCol1};
```

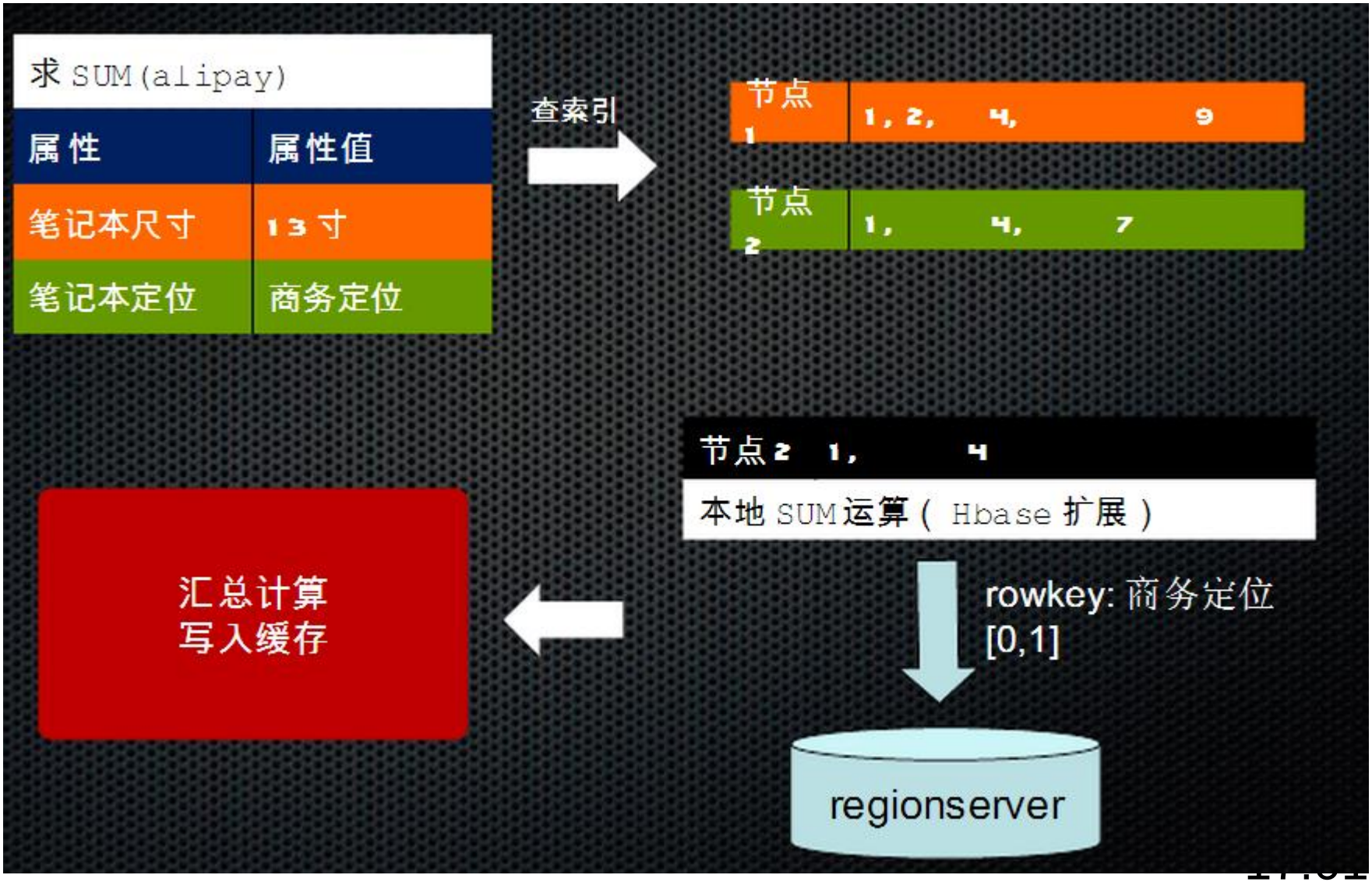
4.3 HBase

淘宝网 Taobao.com

APACHE HBASE

rowkey	列族1	列族2			
	列	列1	列2	列3	列4
浙江	1,2,5	{5455,1,144,11},{108,2,22234,12},{356,0,35355,13}	{0,1,0}	...	...

4.3 HBase


求 SUM(alipay)
属性
属性值
笔记本尺寸
13 寸
笔记本定位
商务定位
查索引
节点 1
1, 2, 4, 9
节点 2
1, 4, 7
节点2 1, 4
本地 SUM 运算（ Hbase 扩展 ）
rowkey: 商务定位
[0,1]
regionserver
汇总计算
写入缓存

4.3 HBase

- 问题 &应对：
  - 本地计算
    - 继承 HRegionServer ， HRegionInterface ，定义自己的 RegionServer 类和 RegionInterface 接口。
    - 继承 Get ，实现 Writable 接口的自定义 Get 类 (Put 同理)
    - 继承 Result ，实现 Writable 接口的自定义 Result 类
    - region 端启动时实例化自定义的 RegionServer 对象， client 端通过自定义的 RegionInterface 接口做反射代理。

4.3 HBase

- 问题 &应对：
  - 冗余量过大(最高时单个Cell超过3GB)
    - 根据Rowkey拆分,使用时rowkey scan
  - 频繁重启HBase
    - 热加载lib
  - Regionserver不均匀
    - Md5/crc32 rowkey作为rowkey header

4.4 总结

特性	Redis 1.4	Redis 2.6	HBase
QPS	低	低	低
运维	复杂	复杂	简单
并行	低	无	低
冗余	低	中	高
网络带宽	高	低	高
扩展性	低	中	低

淘宝网
Taobao.com

Q&A

这里有最具含金量的数据，
这里有完整的大数据处理平台。
欢迎加入数据平台部挑战未来!

@flyinweb
lizhe.zcq@taobao.com