Cassandra在饿了么的应用

主讲人：翟玉勇
时间：2017.06.11

概述

1. Cassandra的基本原理介绍

2. 为什么选择Cassandra

3. 饿了么Cassandra实践

4. 大数据离线平台和Cassandra的整合

Cassandra历史

BigTable

Dynamo

Cassandra概述

Cassandra最初源自FaceBook，集合了Google BigTable面向列的特性和Amazon Dynamo分布式哈希(DHT)的P2P特性于一身，具有很高的性能、可扩展性、容错、部署简单等特点。

Cassandra架构关键字

1. Gossip 点对点通信协议，用于集群之间节点交换位置和状态信息
2. Partitioner 决定如何在集群中的节点间分发数据，也就是哪个节点放止数据的第一个replica
3. Replica Strategy 决定在哪些节点放置数据的其他replica
4. Snitch 定义了复制策略用来放置replicas和路由请求所使用的拓扑信息

Gossip-节点的通信

Cassandra使用点对点通信协议Gossip在集群中的节点间交换位置和状态信息。Gossip进程每秒运行一次，与最多3个其他节点交换信息，这样所有的节点可很快的了解集群中其他节点信息。

1.种子节点

2.Cassandra故障探测

3.Cassandra故障恢复

Partitioner

Partitioner定义了数据如何在集群中的节点分布，哪个节点应该存放数据的第一份拷贝。基本上，Partitioner就是一个计算分区键token的哈希函数。

1. Partition Key 决定数据在Cassandra哪个节点上
2. Clustering Key 用于在各个分区内的排序
3. Primary Key 主键，决定数据行的唯一性

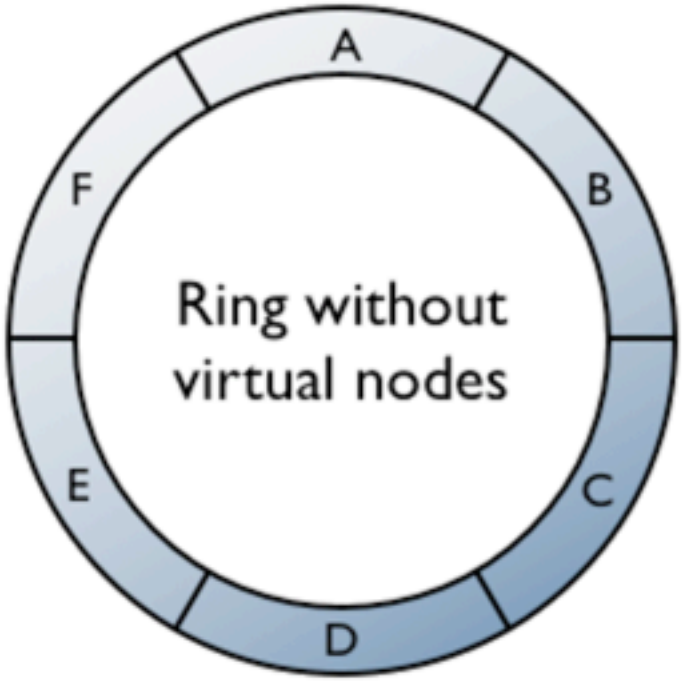
Partitioner

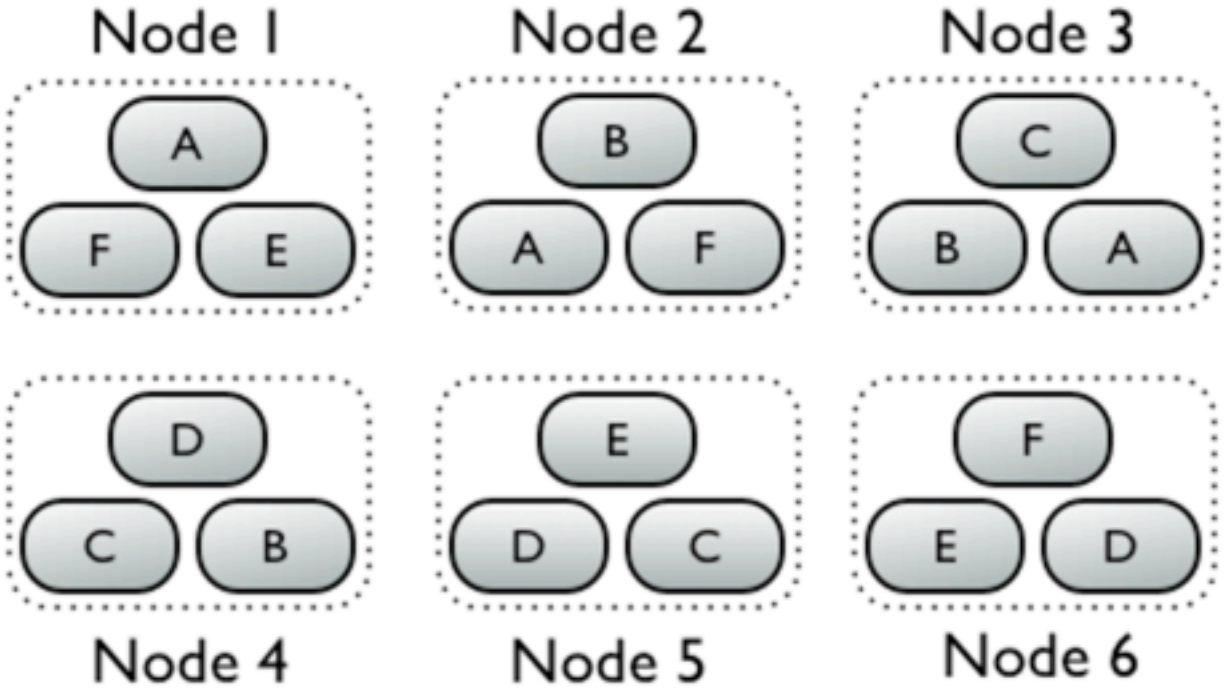
1. Key_part_one，key_part_two共同构成了primary key
2. key_part_one也就是partition key
3. key_part_two就是cluster key

```
create table table2 (
  key_part_one text,
  key_part_two int,
  data text,
  PRIMARY KEY(key_part_one, key_part_two)
);
```

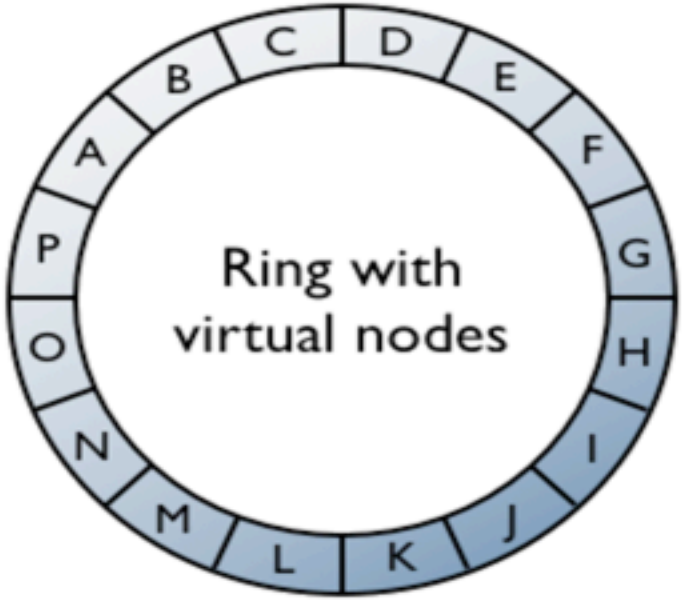
Cassandra如何根据partition key决定数据落在哪个节点？

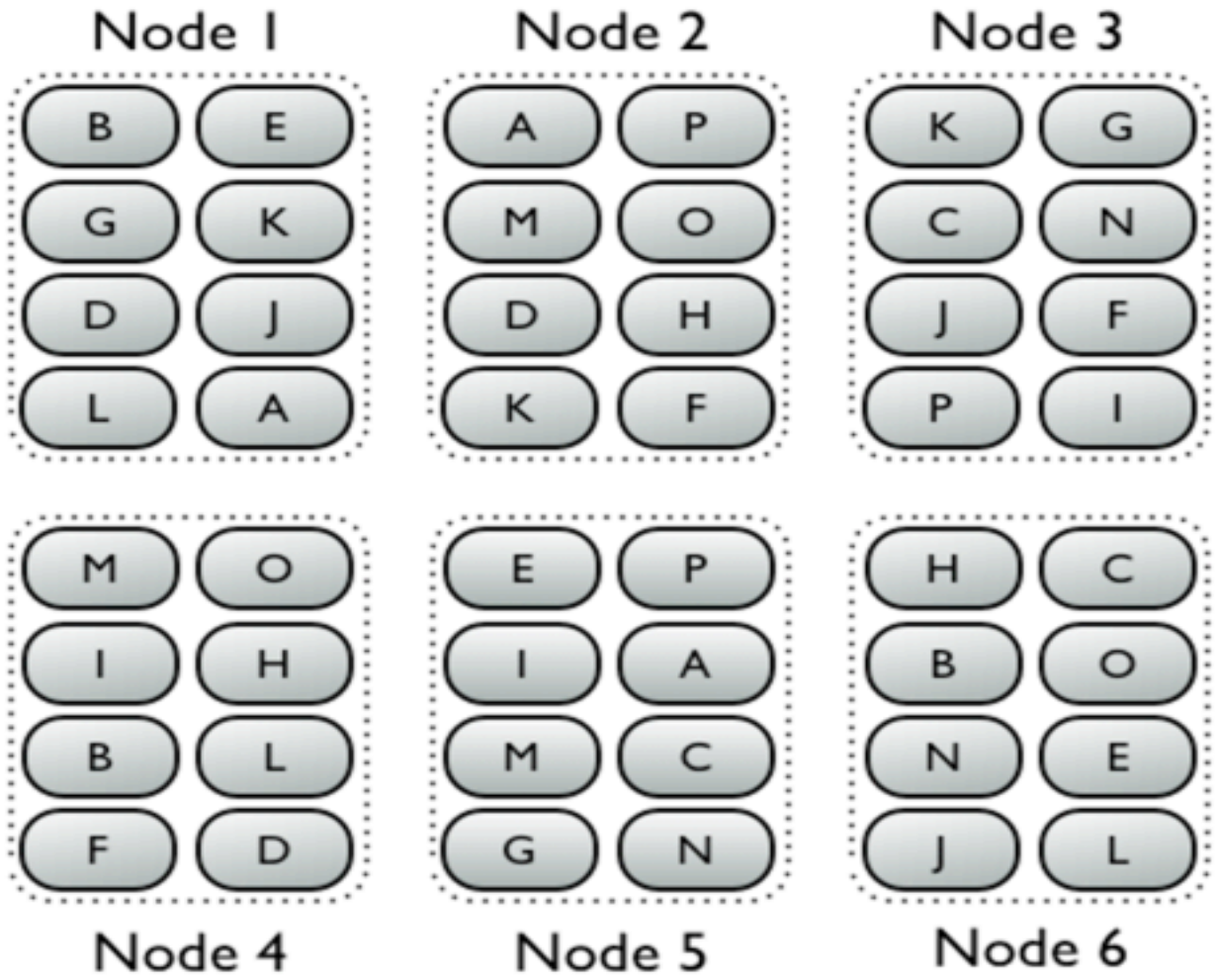
一致性哈希和虚拟节点


A
B
C
D
E
F
Ring without
virtual nodes


Node 1
A
F
E
Node 2
B
A
F
Node 3
C
B
A
D
C
B
Node 4
E
D
C
Node 5
F
E
D
Node 6

一致性哈希和虚拟节点


A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Ring with
virtual nodes


Node 1
B E
G K
D J
L A
Node 2
A P
M O
D H
K F
Node 3
K G
C N
J F
P I
M O
I H
B L
F D
Node 4
E P
I A
M C
G N
Node 5
H C
B O
N E
J L
Node 6

Replica Strategy

Cassandra在多个节点中存放replicas以保证可靠性和容错性。Replica Strategy决定放置replicas的节点，replicas的数目由复制因子确定，比如通常设置3表示每行数据有三份拷贝，每份数据存储在不同的节点。

当前可用的两种复制策略：

1.SimpleStrategy 仅用于但数据中心

```
CREATE KEYSPACE dw WITH replication = {'class': 'SimpleStrategy', 'replication_factor' : 3}
```

2.NetworkTopologyStrategy 用于多IDC场景，可指定每个IDC有多少replicas

```
CREATE KEYSPACE dw WITH replication = {'class': 'NetworkTopologyStrategy', 'DC-SH' : 2, 'DC-BG' : 2}
```

Cassandra主要的数据结构

1. Memtable
   跳表

2. SSTable

3. Bloom filter

SSTable

```
-rw-r--r-- 1 master master   59K Jun  6 08:47 rec-dm_ups_user_info-ka-669566-CompressionInfo.db
-rw-r--r-- 1 master master  161M Jun  6 08:47 rec-dm_ups_user_info-ka-669566-Data.db
-rw-r--r-- 1 master master   10 Jun  6 08:47 rec-dm_ups_user_info-ka-669566-Digest.sha1
-rw-r--r-- 1 master master  133K Jun  6 08:47 rec-dm_ups_user_info-ka-669566-Filter.db
-rw-r--r-- 1 master master  2.8M Jun  6 08:47 rec-dm_ups_user_info-ka-669566-Index.db
-rw-r--r-- 1 master master  9.7K Jun  6 08:47 rec-dm_ups_user_info-ka-669566-Statistics.db
-rw-r--r-- 1 master master   20K Jun  6 08:47 rec-dm_ups_user_info-ka-669566-Summary.db
-rw-r--r-- 1 master master   91 Jun  6 08:47 rec-dm_ups_user_info-ka-669566-TOC.txt
```

Data 真正的数据
Filter bloom filter
Index 索引文件，保存key和data数据位置的映射关系
Summary index采样数据
Statistics 存放data中columns和row个数信息
CompressionInfo 存放compression信息

CQL语言

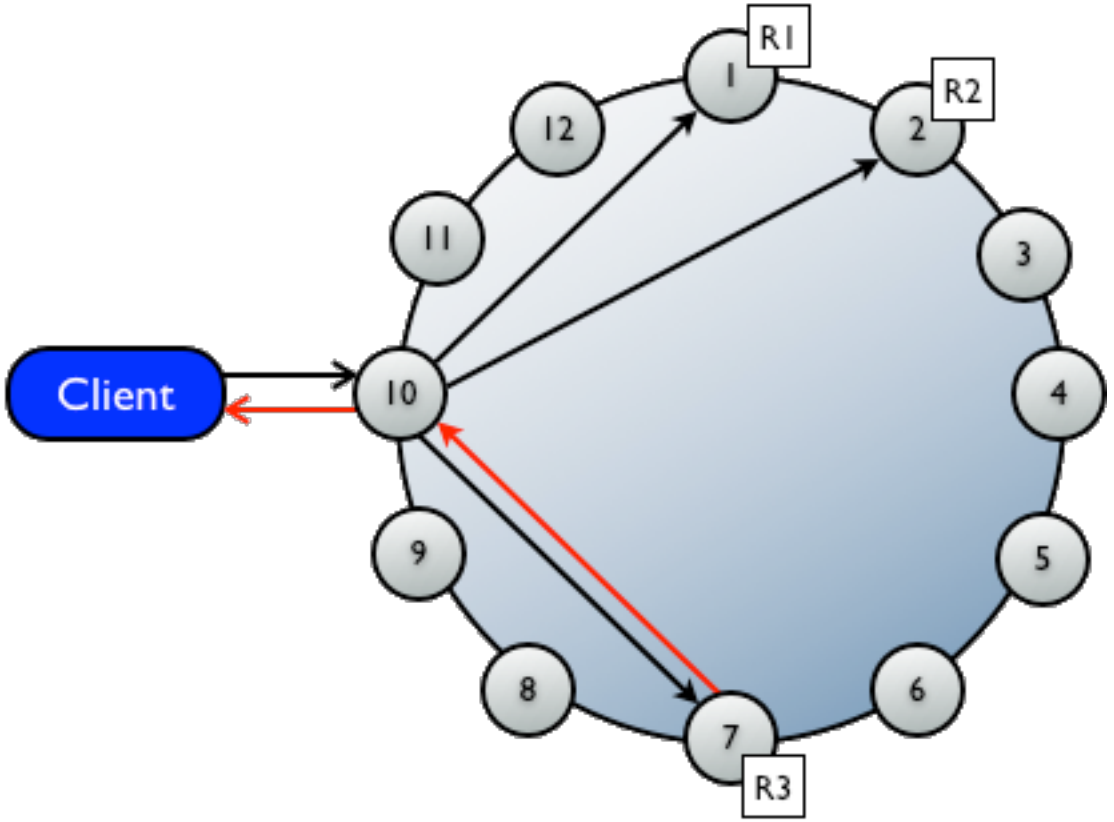
1.Cql类似于SQL

2.DDL操作create table，drop table等等

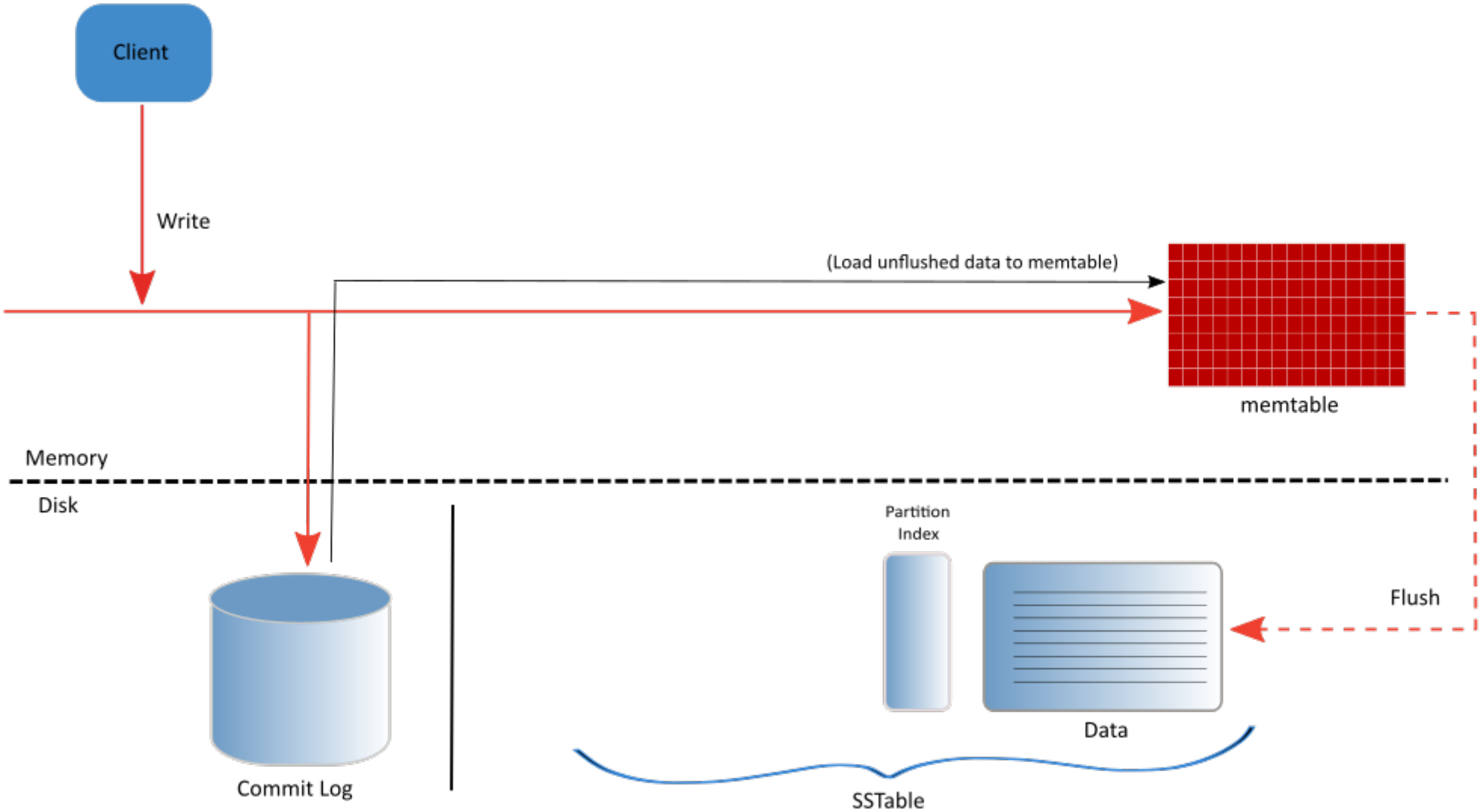
3.支持DML操作INSERT、UPDATE、DELETE等等

4.查询数据通过select

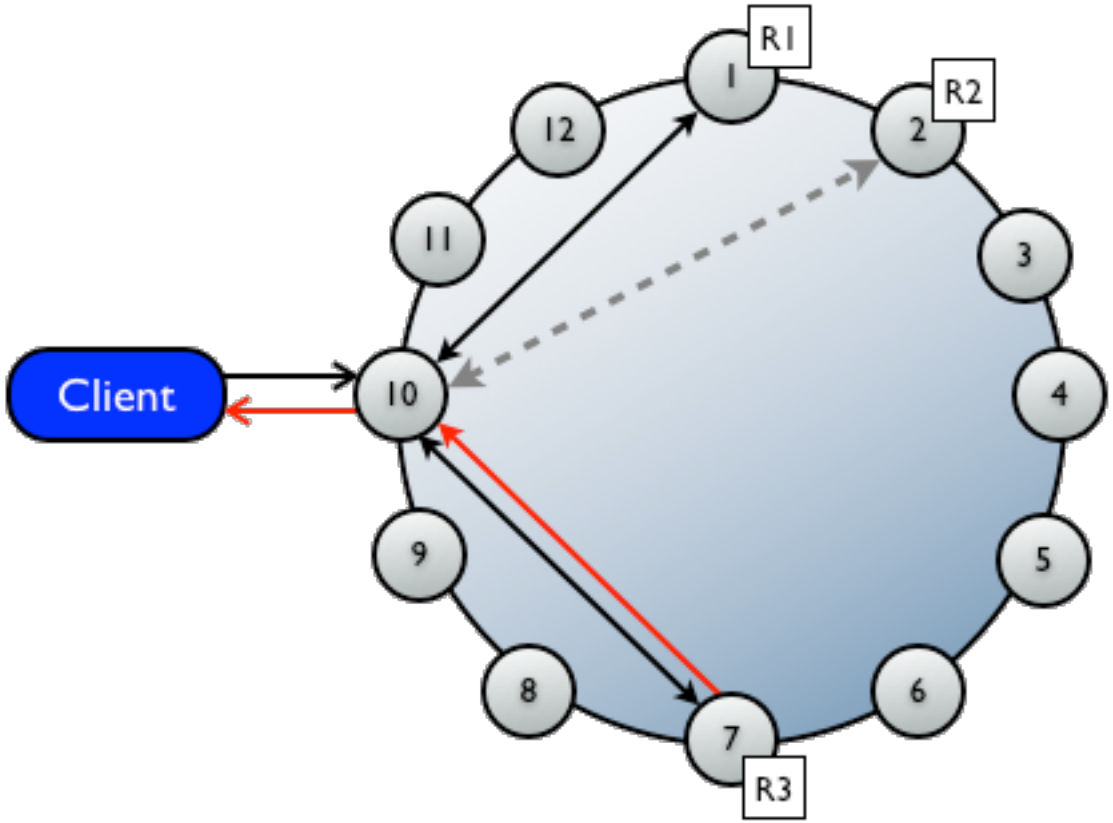
Client请求-写请求



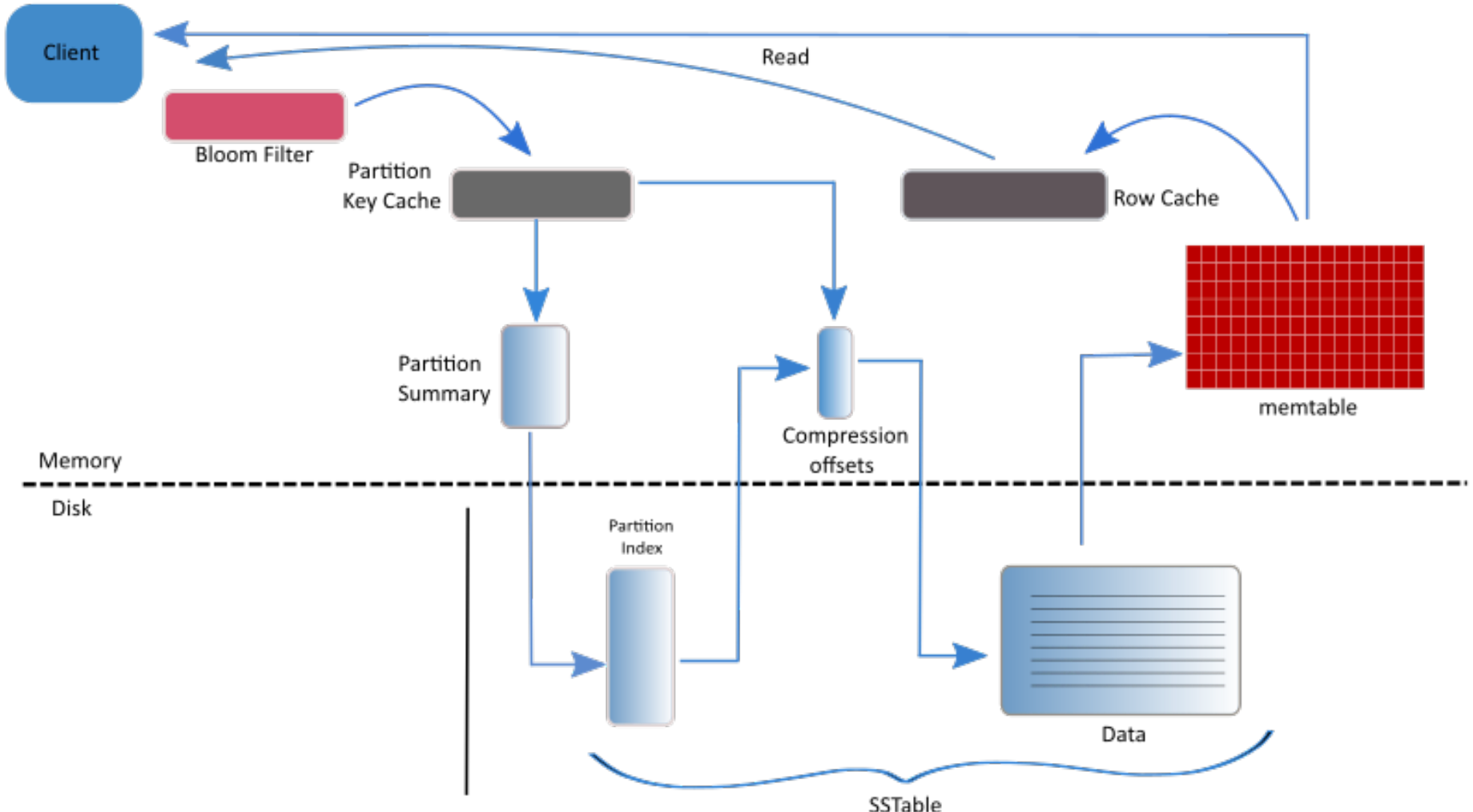
Cassandra写路径



Client请求-读请求



Cassandra读路径



Cassandra一致性保障

1. Hinted Handoff
2. Read Repair
3. Anti-Entropy Node Repair

为什么选择Cassandra

1. 运维成本
   1) 部署简单
   2) 只需要运维一个组件
   3) 监控成本低

2. 开发成本
   1) 类似sql的cql语言，对开发友好，低成本上手
   2) DataStax公司提供的强大的java client
   3) 可调节的数据一致性
   4) 异步接口

3. 适用场景
   1) Cassandra自带多idc策略
   2) 我们的业务需求

Cassandra在饿了么的实践

1. 生产应用
   1)用户画像
   2)历史订单
   3)dt.api
2. Client选择
3. 运维和监控
4. 性能调优

生产应用-用户画像

1. 5 node

2. 2. 4亿+用户数据

3. 100+用户属性

4. 每天5000万+数据更新

5. Scheme变更频繁(加字段)

6. 99%读延时3-5ms

生产应用-历史订单

1. Sata盘集群 平均响应时间小于80ms

2. 15+ node

生产应用-dt.api

饿了么大数据平台自助化数据接口平台

1.One sql one api

2.50+ Cassandra cql api

Cassandra客户端选择

1. Jdbc

2. Thrift api

3. Datastax java driver(推荐使用)

Datastax java driver

1.Sync和Async api

2.连接池

3.自动节点发现

4.自动重连机制

5.可配置的load balance和重连策略

6、query build

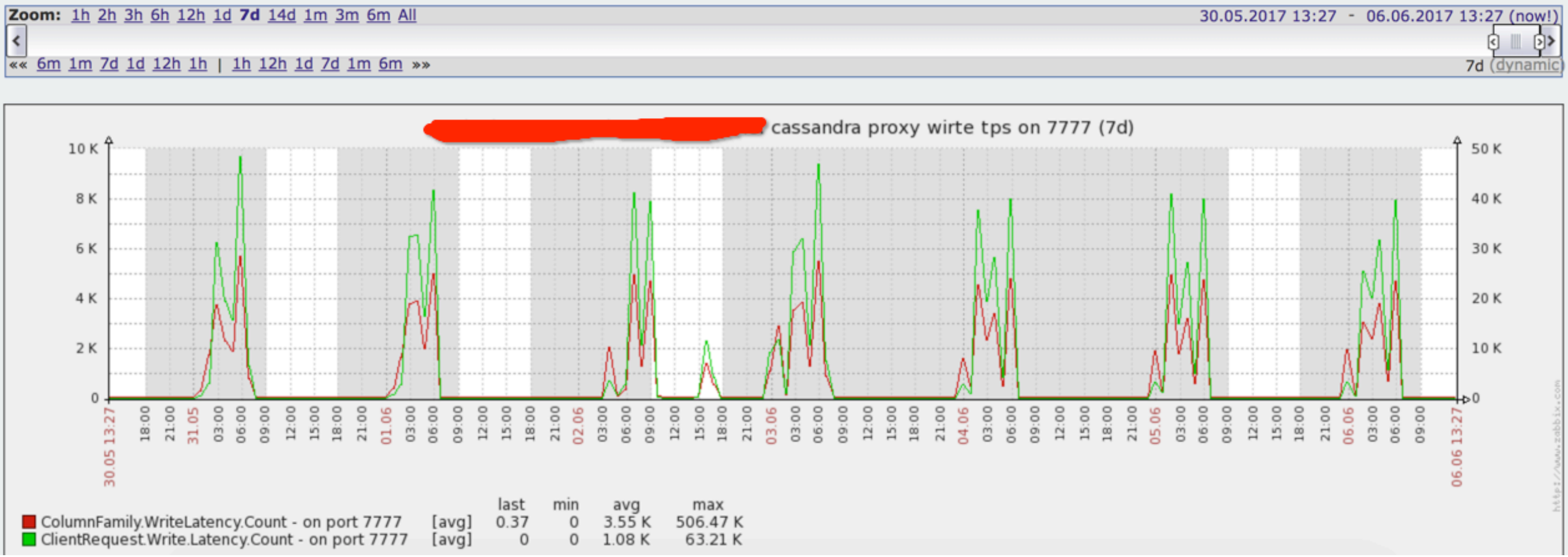
7、Object mapper

运维和监控

1. ansible自动部署

2. Zabbix监控

运维和监控


Zoom: 1h 2h 3h 6h 12h 1d 7d 14d 1m 3m 6m All
30.05.2017 13:27 - 06.06.2017 13:27 (now!)
«« 6m 1m 7d 1d 12h 1h | 1h 12h 1d 7d 1m 6m »»
7d (dynamic)
[redacted] cassandra proxy wirte tps on 7777 (7d)
10 K
8 K
6 K
4 K
2 K
0
50 K
40 K
30 K
20 K
10 K
0
30.05 13:27
31.05
01.06
02.06
03.06
04.06
05.06
06.06
06.06 13:27
last min avg max
ColumnFamily.WriteLatency.Count - on port 7777 [avg] 0.37 0 3.55 K 506.47 K
ClientRequest.Write.Latency.Count - on port 7777 [avg] 0 0 1.08 K 63.21 K

性能调优

1. 集群调优
    1)Cassandra本身参数优化设置
    2)JVM参数优化设置

2. schema设计优化

性能调优-集群调优化

集群参数设置

1. memtable_allocation_type
   heap_buffers: on heap nio buffer
   offheap_buffers: off heap(direct) nio buffers
   offheap_objects: native memory
2. concurrent_write和concurrent_read

3. Sstable compression

4. Concurrent compactor

5. memtable_flush_writers

6. Netty io线程数目

性能调优-集群调优化

JVM调优化

1. 堆的大小选择
2. 取消偏向锁

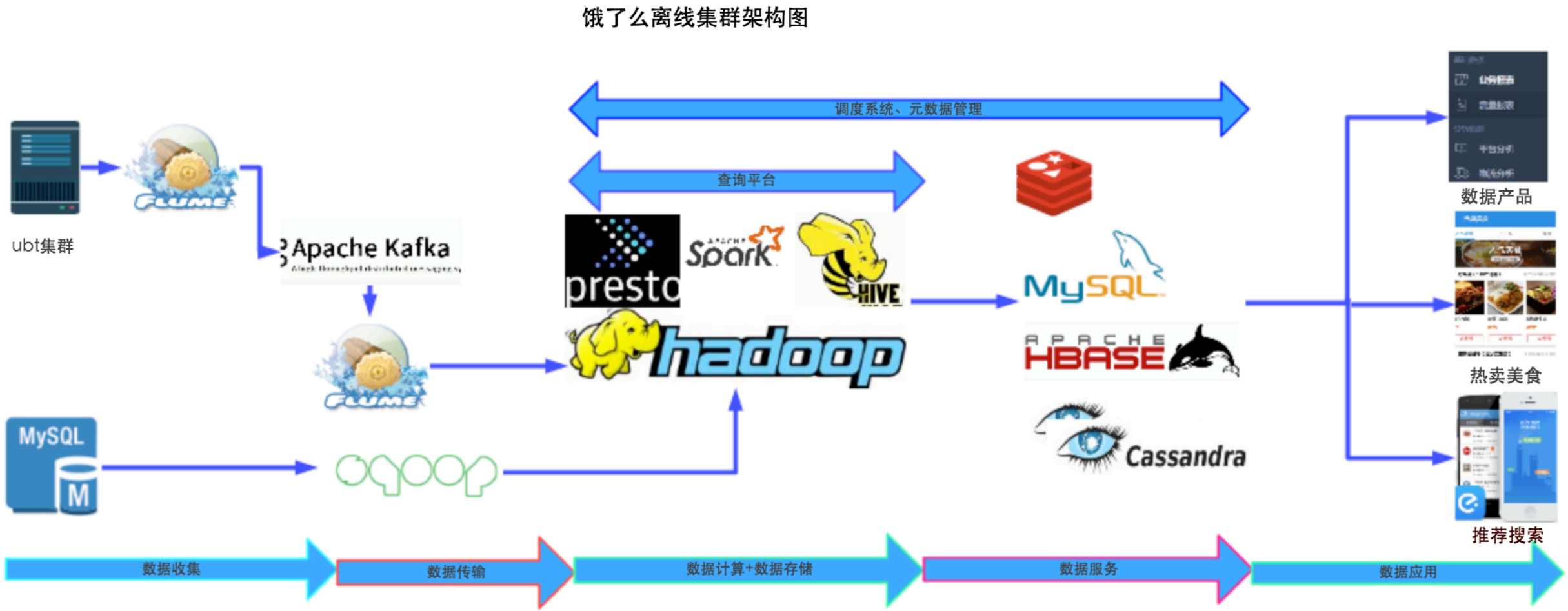
https://tobert.github.io/pages/als-cassandra-21-tuning-guide.html

性能调优-集群调优化

Scheme设计优化

1. Primary key设计，避免热点
2. 读修复关闭
3. Compaction strategy策略选择
4. Ttl设置
5. Row cache启用

大数据离线平台和Cassandra的整合



大数据离线平台和Cassandra的整合

两大数据推送Cassandra工具

1. Hive Integrate Cassandra Native Protocol
2. Hive Integrate Cassandra Bulkload

大数据离线平台和Cassandra的整合

Hive Integrate Cassandra Native Protocol

1. Hive外部表映射到Cassandra表
2. Insert Into HiveTable Select 简单快捷
3. 跨机房推送限流/限速
4. 异步写

大数据离线平台和Cassandra的整合

Hive Integrate Cassandra Native Protocol

```
CREATE EXTERNAL TABLE `hive_table`(
  `user_id` int COMMENT 'from deserializer',
  `your_name` string COMMENT 'from deserializer',
  `user_sex` string COMMENT 'from deserializer')
ROW FORMAT SERDE
  'com.eleme.bigdata.cassandra.serde.cql.CqlMapSerde'
STORED BY
  'com.eleme.bigdata.cassandra.cql.CqlStorageHandler'
WITH SERDEPROPERTIES (
  'cassandra.host'='127.0.0.1,127.0.0.2',cassandr集群ip列表
  'cassandra.username'='username', 用户名
  'cassandra.ks.name'='keyspacename', keyspace名称
  'cassandra.port'='9042', 端口
  'cassandra.table.name'='表名',
  'cassandra.password'='密码')
TBLPROPERTIES (
  'cassandra.batchmutate.size'='5', 异步写并发数目
  'cassandra.batchmodel'='asynchronous.statements', 异步写
  'cassandra.consistency.level'='QUORUM')写一致性参数
```

大数据离线平台和Cassandra的整合

Hive Integrate Cassandra Bulkload

1. hive生成Cassandra底层的SSTable文件直接load到Cassandra

2. 适用于数据快速初始化

3. 需要控制生成的SSTable大小避免Compact耗时多久

大数据离线平台和Cassandra的整合

Hive Integrate Cassandra Bulkload

```
create EXTERNAL table cassandra_table(transaction_id string, user_name string, amount double)
STORED BY '[redacted].eleme.[redacted]CqlBulkStorageHandler'
TBLPROPERTIES (
'cassandra.output.keyspace.username' = 'cassandra',
'cassandra.output.keyspace'='test',
'cassandra.output.partitioner.class'='org.apache.cassandra.dht.Murmur3Partitioner',
'cassandra.output.keyspace.passwd'='cassandra',
'mapreduce.output.basename'='cassandra',
'cassandra.output.thrift.address'='127.0.0.1',
'cassandra.output.delete.source'='true',
'cassandra.columnfamily.insert.cassandra'='insert into test.cassandra(transaction_id,user_name,amount)values(?,?,?)',
'cassandra.columnfamily.schema.cassandra'='CREATE TABLE test.cassandra (transaction_id text,user_name text PRIMARY KEY,amount double)');
```

Q&A

饿了么

饿了么技术社区 七牛云 QINIU IT大咖说 知识分享平台

THANKS !