Percent百分点

2019
05
08-10

北京新云南皇冠假日酒店

数据风云十年变迁

DTCC 第十届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2019

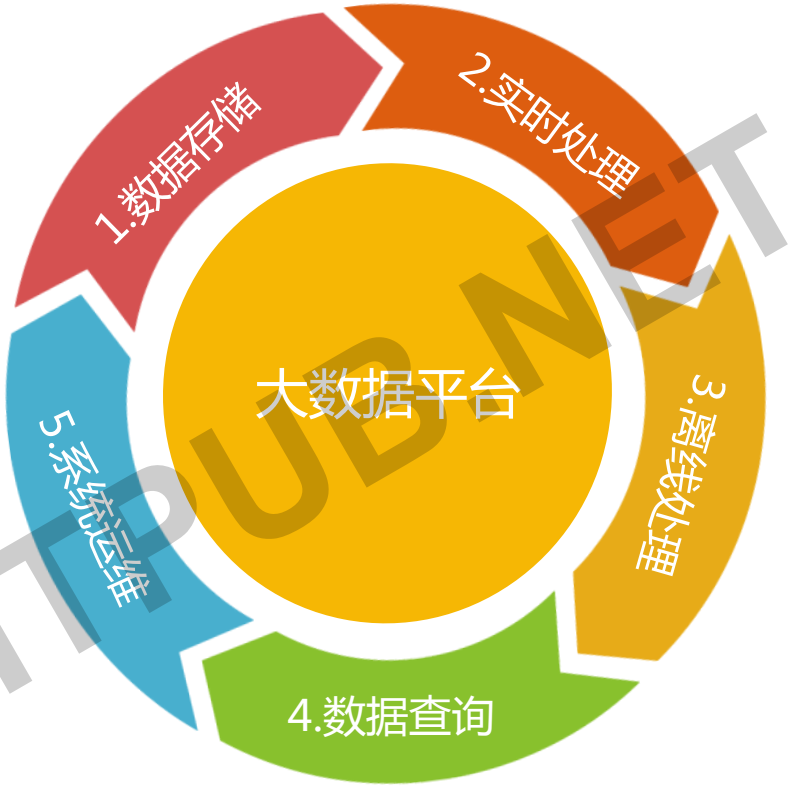
ITPUB.NET

万亿级大数据平台的建设实践

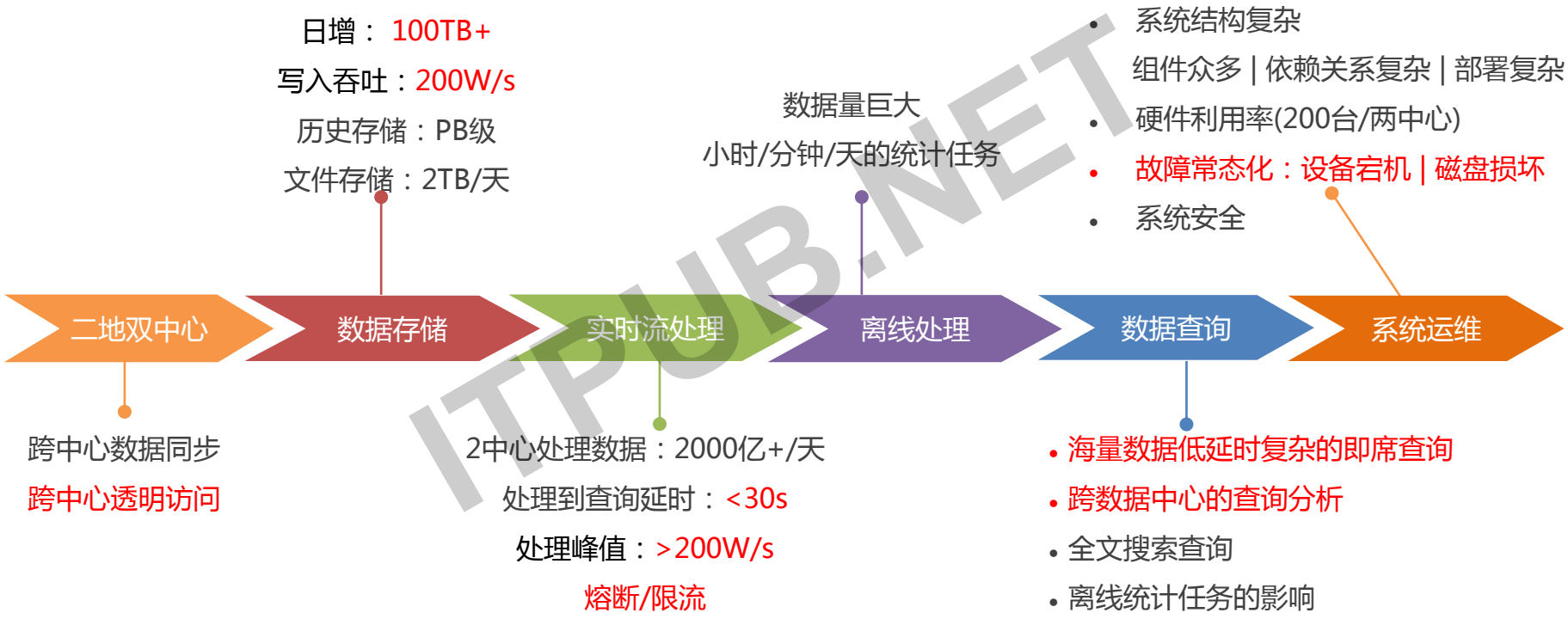
百分点 – 赵 群

目 录

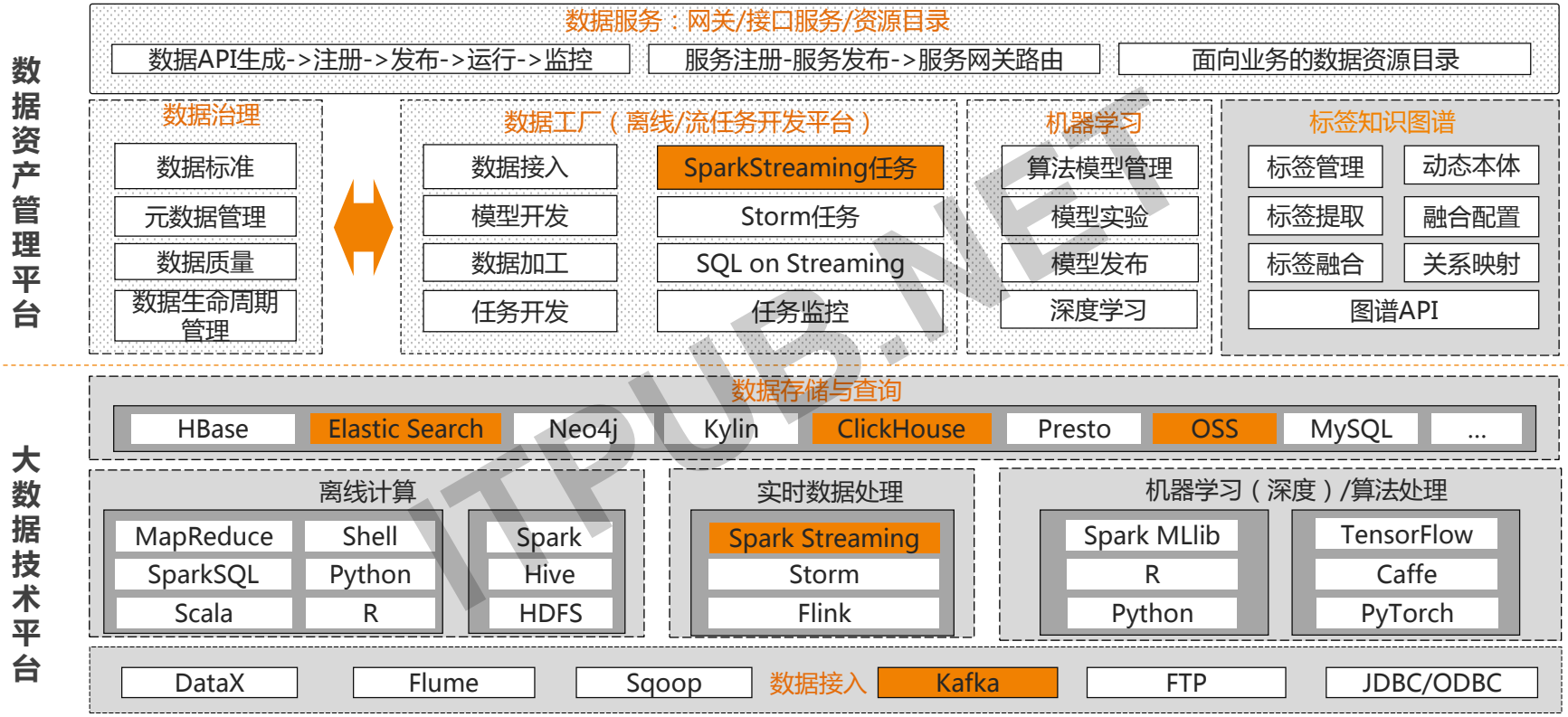
问题与挑战 – 大数据平台维度划分



问题与挑战 – 超大规模对平台提出的高要求



百分点超大规模实时数据分析典型架构



实时数据分析典型架构应对的核心组件

结构化存储：ClickHouse

消息通道 ：kafka

流处理框架： SparkStreaming

全文搜索：ElasticSearch

文件存储：OSS (HBase + Ceph)

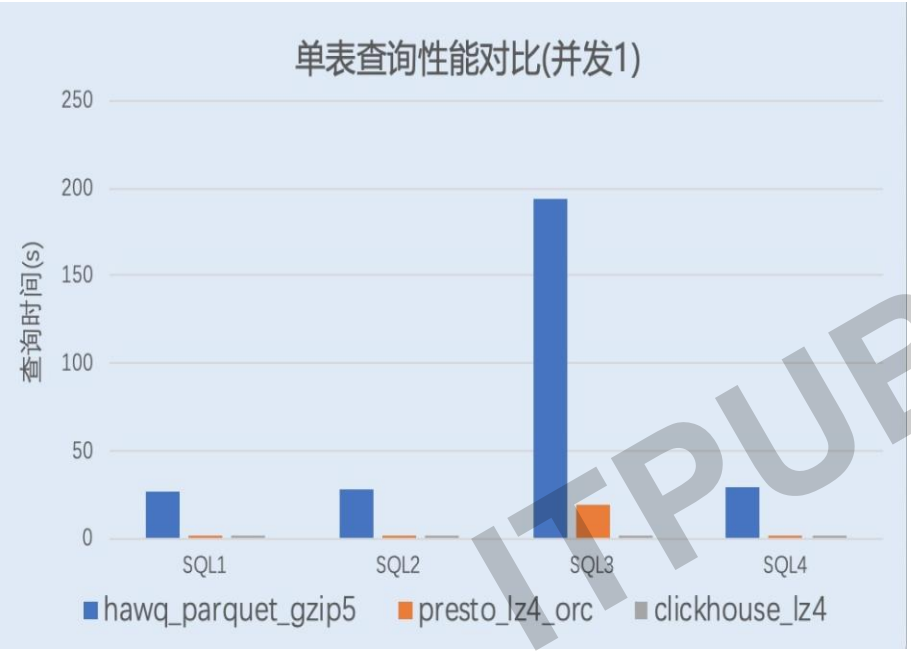
组件设计 - 存储ClickHouse

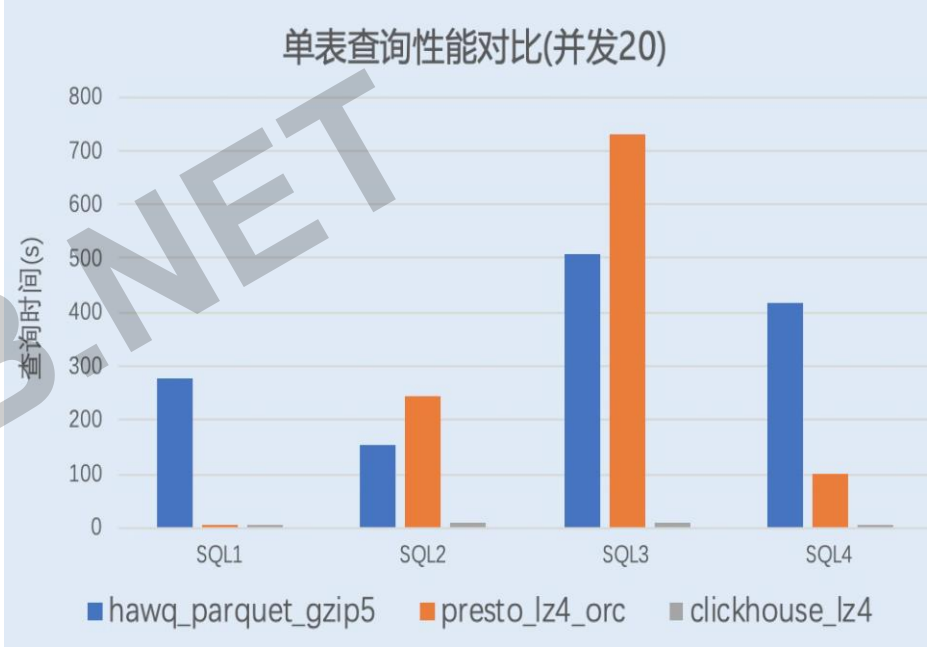
业 务：
1、超大规模的单表查询/分析；
2、有一定的并发要求；
3、实时性要求；

1. PB级的数据存储
2. 高性能的查询/分析能力
3. 低延时写入及吞吐能力
4. 数据压缩
5. 跨中心能力

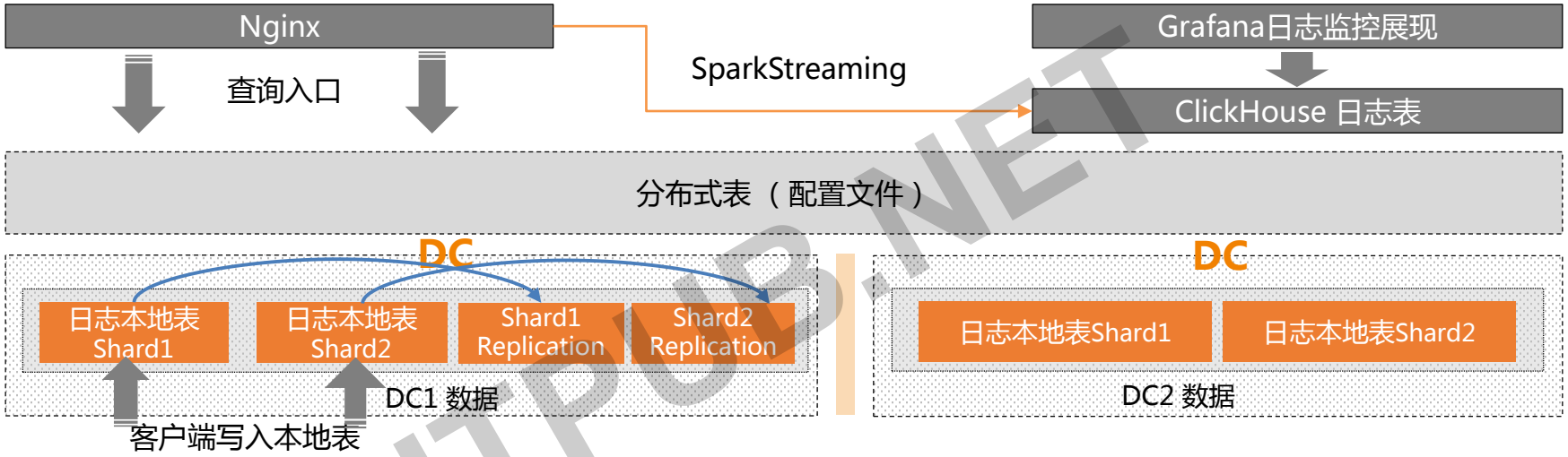
ClickHouse
Presto
HAWQ
Druid
Elastic Search

组件设计 - OLAP引擎的选型与评估





组件设计 - ClickHouse整体设计



1. ClickHouse跨中心透明访问。
2. 业务端可以查询多中心数据，也可以查询特定分中心数据。
3. 禁止分布式写。
4. 性能影响：1/4 ～1/3

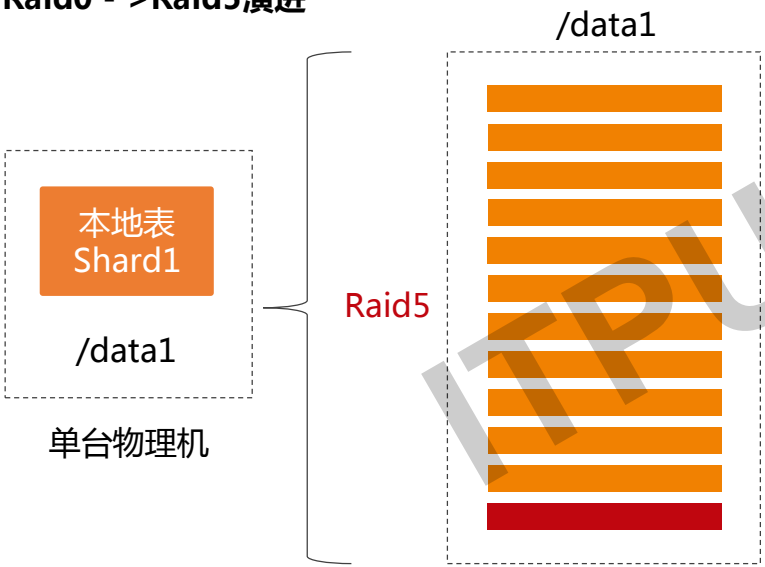
组件设计 – ClickHouse磁盘Raid的选择

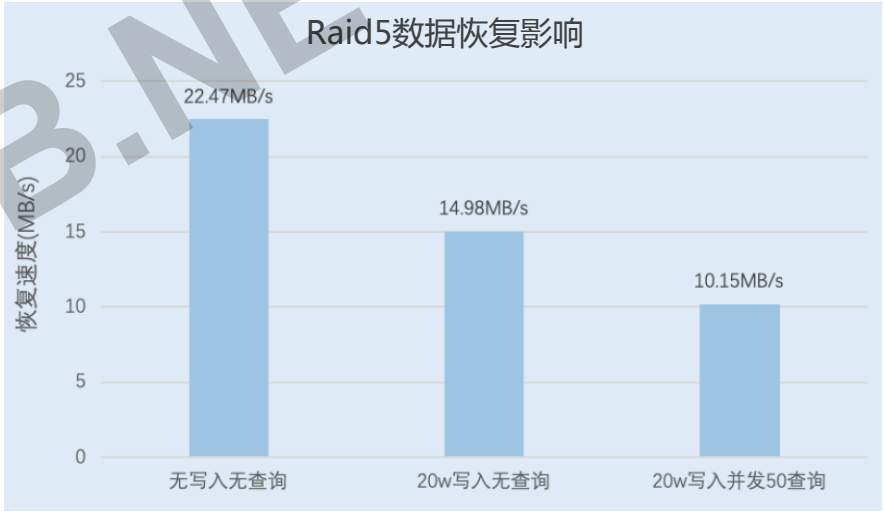
1、Raid5增加磁盘数据可靠性和读取能力

2、热备盘减少运维压力

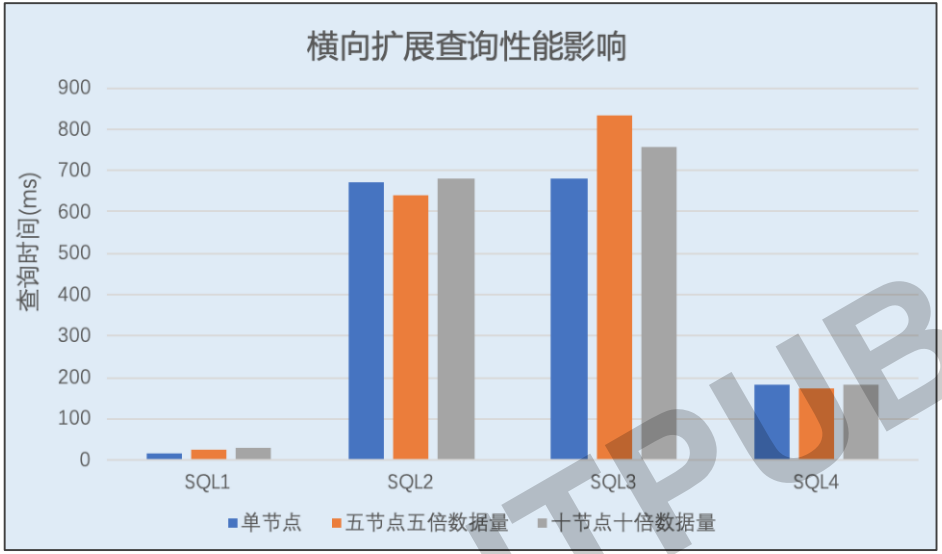
3、控制写入，保障查询性能

Raid0 - >Raid5演进

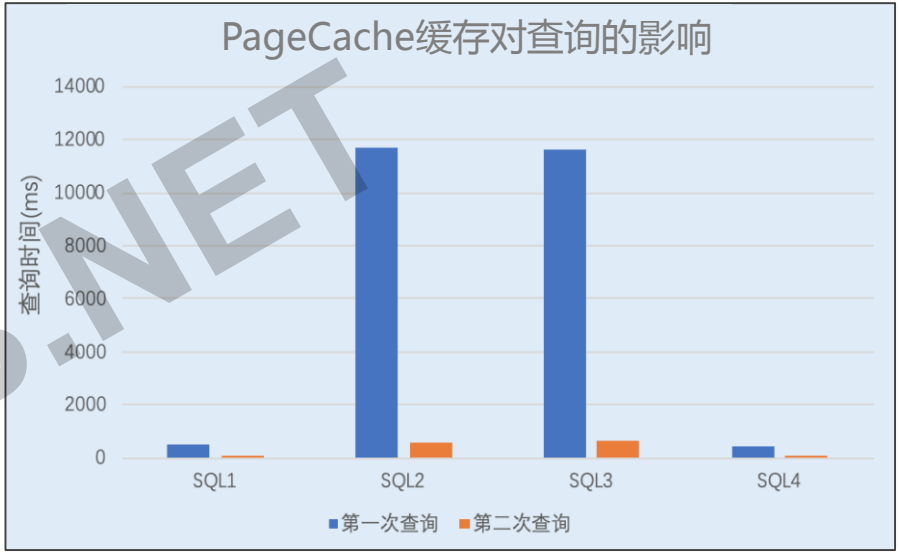




组件设计 – ClickHouse的相关测试分析



横向扩展对查询性能几乎无影响
可以基于单节点/分区评估查询性能



数据预热对查询有数量级提升
针对缓存更换条件同样生效

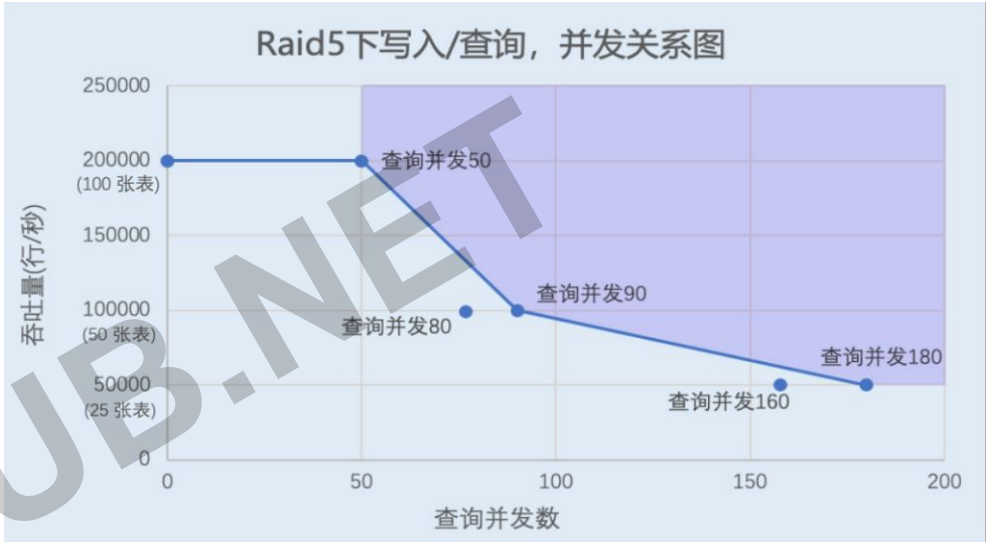
组件设计 - 如何保障ClickHouse写入的稳定性

1、20W/s （35次）提交，并发50

2、10W/s(17次)提交，并发90

3、5W/s(8次)提交，并发90

确保业务命中在安全区域



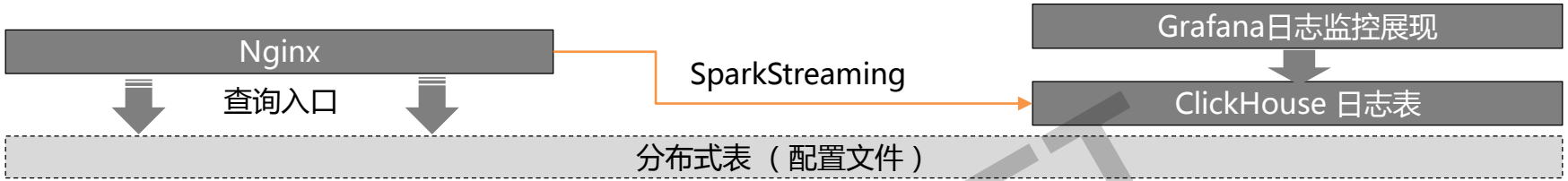
1、平衡好合并速度和Part数量的关系，一定是需要相对均衡的。

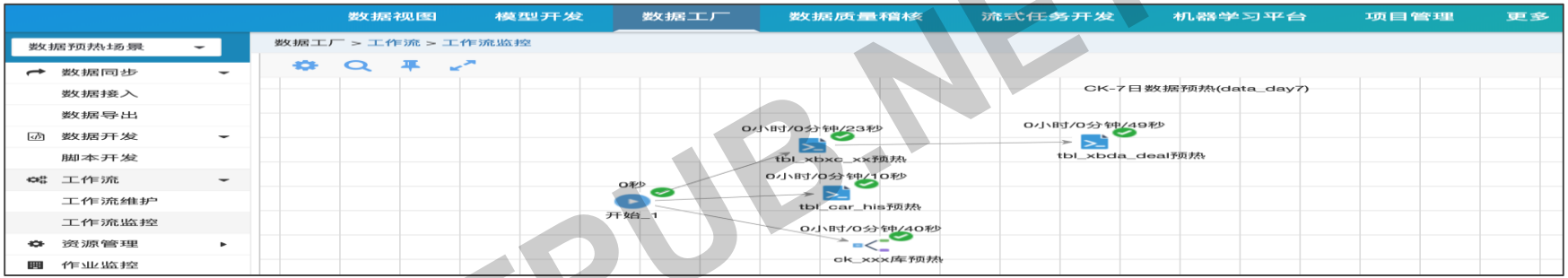
2、Part数量，实际代表着提交频率，一定是稳定，且经过估算的。

3、ClickHouse的查询和写入共同受限于Query数限制，需要分配好配额。

4、不推荐直接写入分布式表。

组件设计 - ClickHouse的查询





1、限制单条查询内存使用量和单节点查询内存使用量，预防节点Down机。

2、Query数量限制异常：控制好配额/连接池。

3、集群的Query日志，找出慢查询。我们直接通过Nginx收集了原始日志。

4、针对热数据进行查询预热。

组件设计 - 最佳实践之参数配置

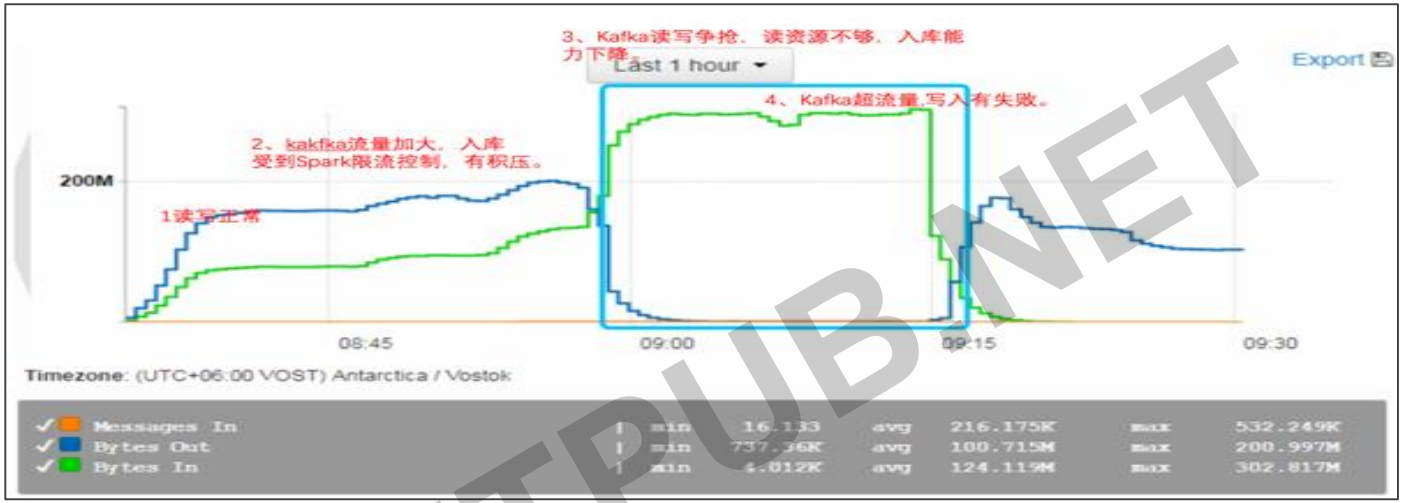
参数名称	默认值	调整后的值	参数说明	参数所在配置文件
max_memory_usage_for_all_queries	0	200G	单台服务器上所有查询的内存使用量，默认没有限制	users.xml
max_memory_usage	10G	100G	一个查询在单台服务器的最大内存使用量，默认是10GB	users.xml
max_execution_time	0	300	单次查询耗时的最长时间，单位为秒。默认没有限制	users.xml
distributed_product_mode	deny	local	默认SQL中的子查询不允许使用分布式表，修改为local表示将子查询中对分布式表的查询转换为对应的本地表	users.xml
background_pool_size	16	32	后台用于merge的线程池大小	users.xml
log_queries	0	1	system.query_log表的开关。默认值为0，不存在该表。修改为1，系统会自动创建system.query_log表，并记录每次query的日志信息	users.xml
skip_unavailable_shards	0	1	当通过分布式表查询时，遇到无效的shard是否跳过。默认值为0表示不跳过，抛异常。设置值为1表示跳过无效shard	users.xml
keep_alive_timeout	10	600	服务端与客户端保持长连接的时长，单位为秒	config.xml
max_concurrent_queries	100	150	最大支持的Query数量	config.xml
session_timeout_ms	3000	120000	ClickHouse服务和Zookeeper保持的会话时长，超过该时间Zookeeper还收到不ClickHouse的心跳信息，会将与ClickHouse的Session断开	metrika.xml

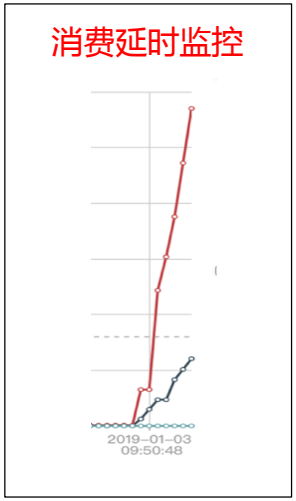
组件设计 – 消息通道Kafka

1. 解耦
2. 吞吐量
3. 数据缓冲
4. 数据路由
5. 大数据生态关系

Kafka
Pulsar

组件设计 - Kafka设计与评估思路

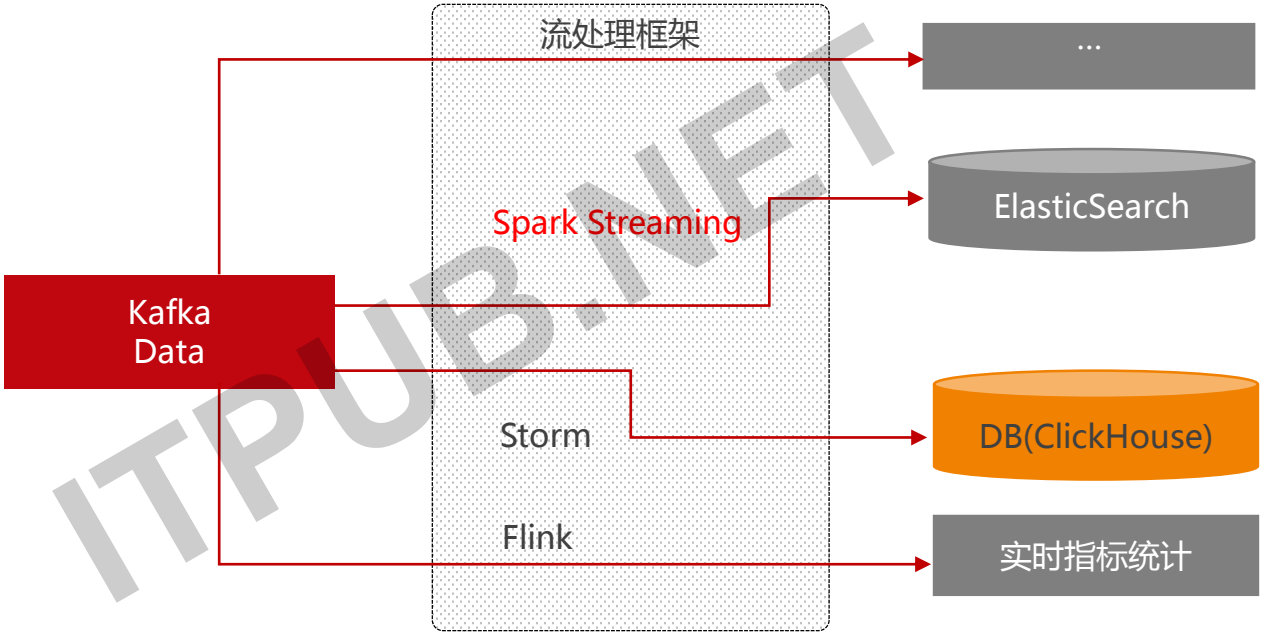




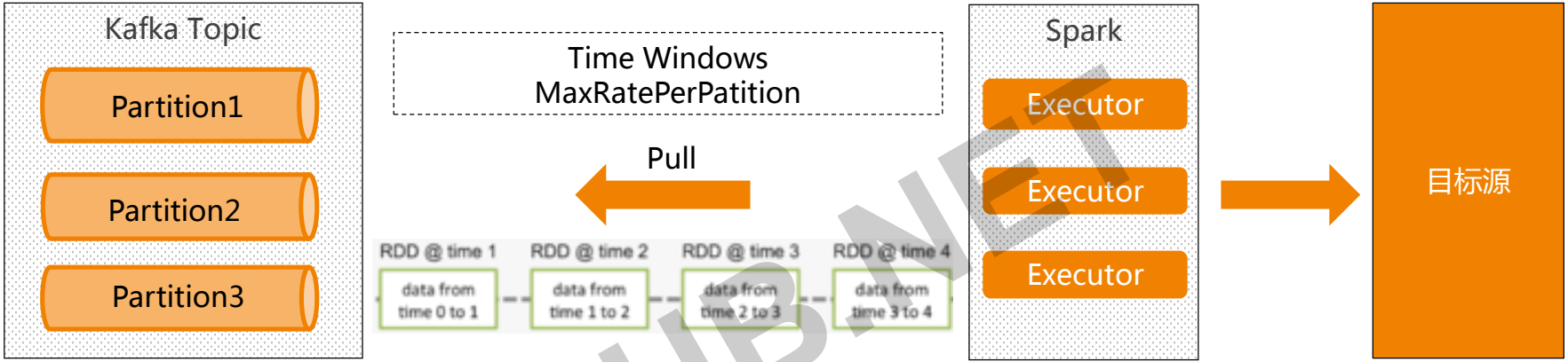
阶段一：读写正常（正常设计状态）
阶段二：流量增大，根据限流影响、高峰期存在积压（如果长期，根据业务情况扩容）
阶段三：读写争抢，读资源不够，入库能力下降（必须扩容；思考：读写配额控制 ？）
阶段四：超流量，写入失败，服务开始出现异常

组件设计 - 流处理框架SparkStreaming

- 高的吞吐量
- 稳定的处理流控制（数据量/时间）
- 计算资源的控制

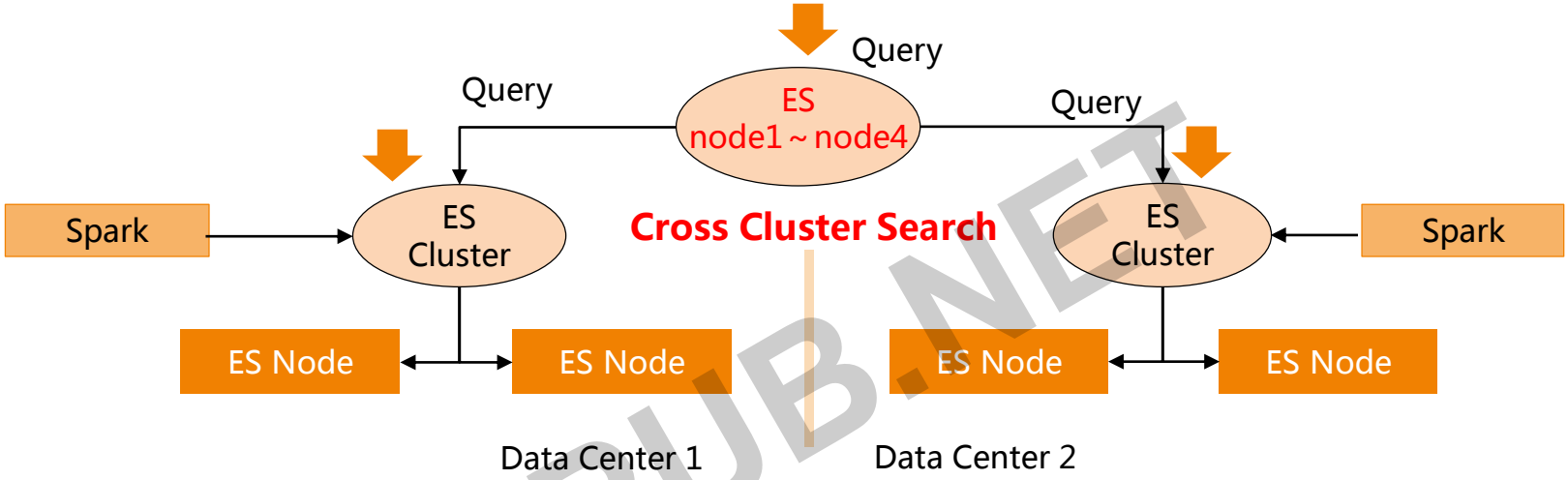


组件设计 - 流处理框架SparkStreaming



1. 时间窗口保障持续稳定提交频率。（保障对ClickHouse写入的稳定）
2. SparkStreaming反压机制，实现处理能力动态平衡。（设定合理的拉取数量）
3. Spark on Yarn 资源可控。
4. 一个Executor可以处理多个Partition,一个Partition不能同时由多个Executor处理。
5. 以写入ClickHouse为例，目前一个Executor处理在30000/s 左右。
6. 假设我们需要一个满足300W/s的处理能力。 在源读取没有瓶颈的情况下，可以 Executor数： 300 /3 = 100(个)。

组件设计 - ElasticSearch的设计



1. 性能影响：TPS：2到3倍降低
2. 配置多个Cross Cluster Search负载均衡
3. 集群Down、节点Down机器容错配置
4. 基于SparkStreaming持续稳定的时间窗口提交

组件设计 – 二进制文件存储OSS

1. 存储二进制数据
2. 友好的API支持（Http)
3. 异步调用
4. 大量的小文件
5. 写多读少

GlusterFS

HDFS

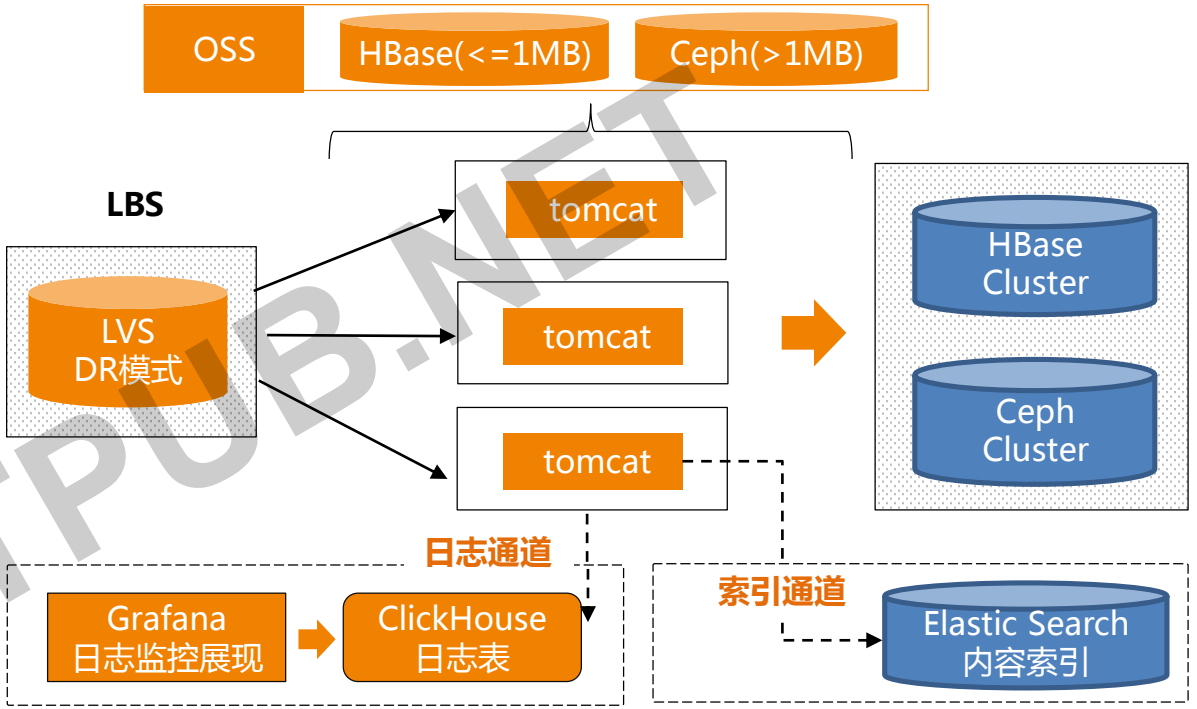
Swift

OSS(HBase + Ceph)

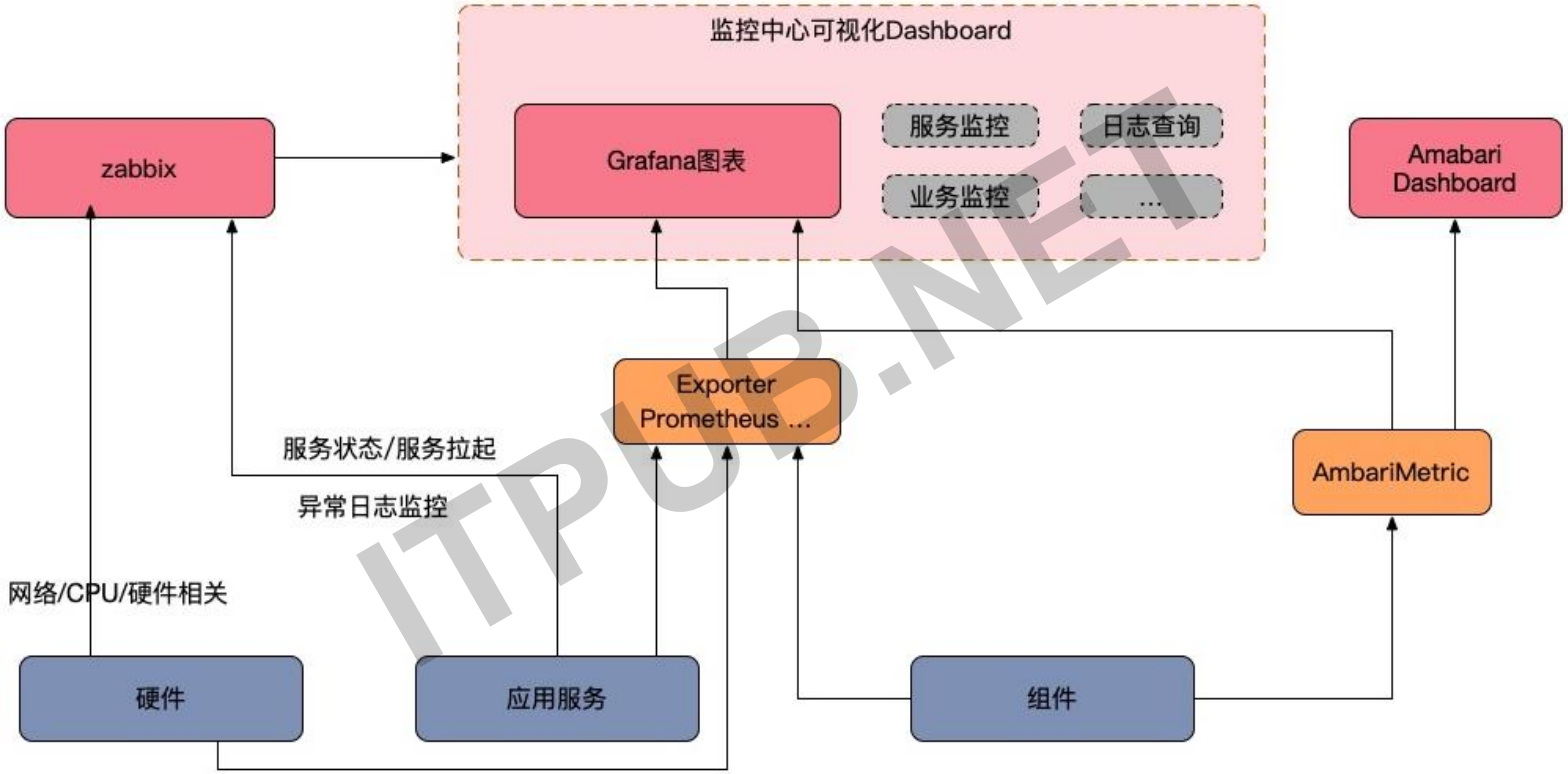
组件设计 – 存储OSS的设计

1、Hbase K-V存储支撑高并发读写。

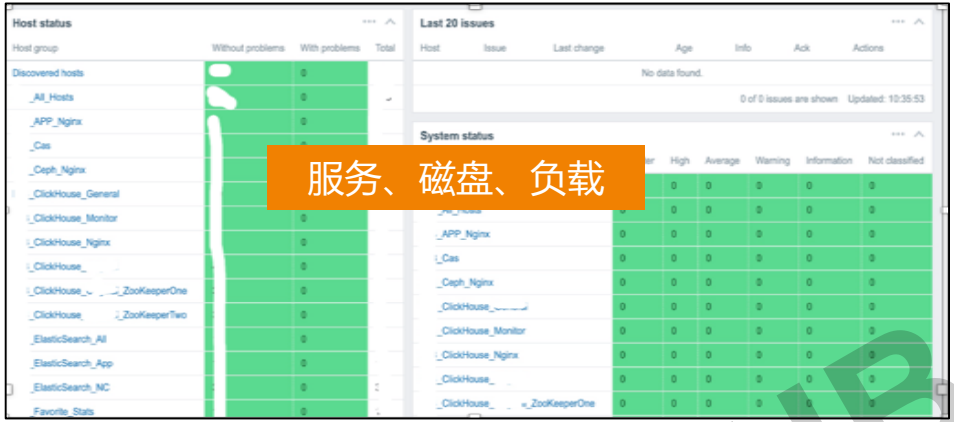
1. Ceph支撑大文件存储
2. 基于Nio通信，异步提交存储。
3. LVS支撑高吞吐量。
4. HBase和Ceph的TTL支持文件的生命周期管理

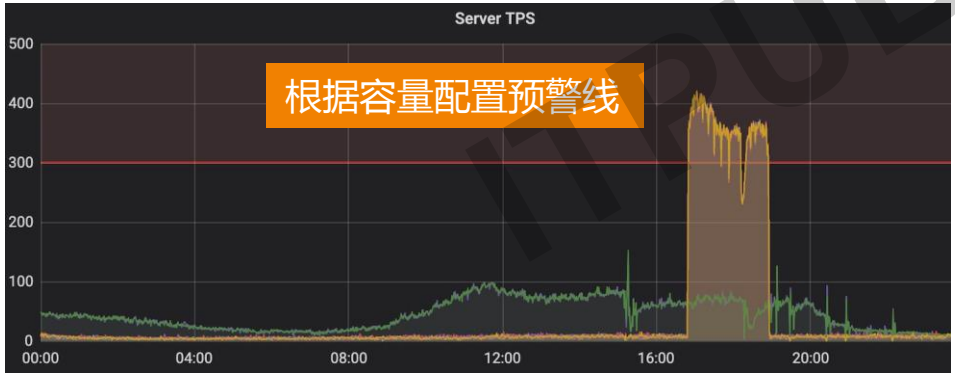


数据平台的持续运维与监控设计和实现：拥抱开源



数据平台的持续运维与监控设计和实现：拥抱开源



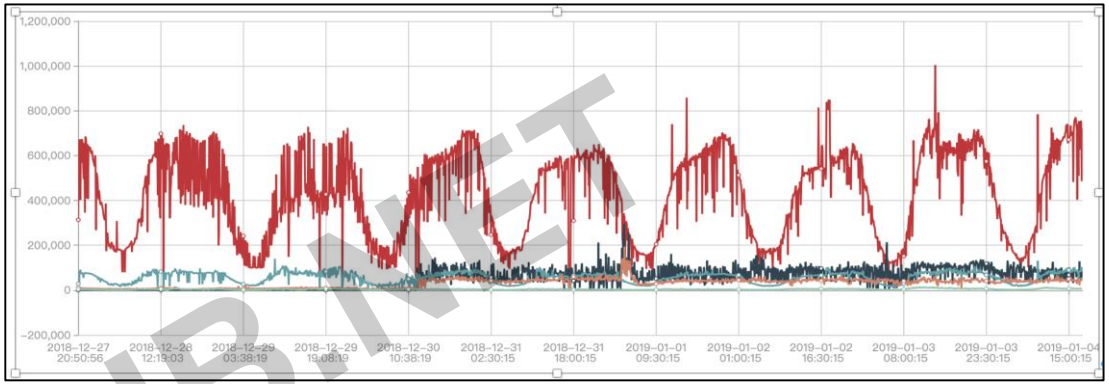


数据平台的持续运维与监控设计和实现：拥抱开源

正常状态

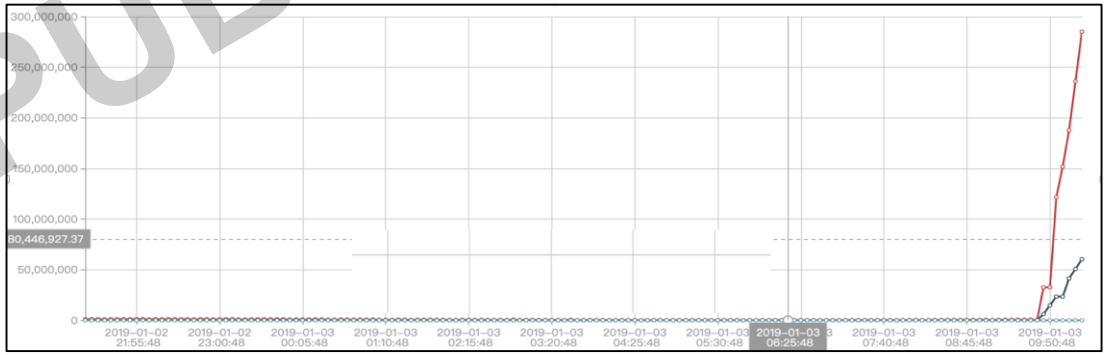
有波峰/波谷，和SparkStreaming时间窗口相关

大的波峰和波谷流量高峰造成处理量大于60W/s



异常状态

消费延时持续扩大，没有缩减。意味着处理有问题，或者需要增加处理能力。



数据平台的持续运维与监控设计和实现：拥抱开源

关于百分点

赵 群 —— 百分点研发总监，大数据平台技术负责人。2015年加入百分点，负责大数据操作系统BD-OS、数据开放服务平台、机器学习平台等多款产品的架构设计和研发；百分点是一家中国领先的数据智能技术企业，致力于帮助企业和政府机构在智能时代下，最大限度 从海量的数据资源中挖掘内在价值，在复杂多变的业务场景下，实现从数据到知识再到智能决策的演进。

Percent百分点

THANKS