



Elasticsearch在金融大数据的应用

廖晓格

2019年4月

目录

一. 总体架构

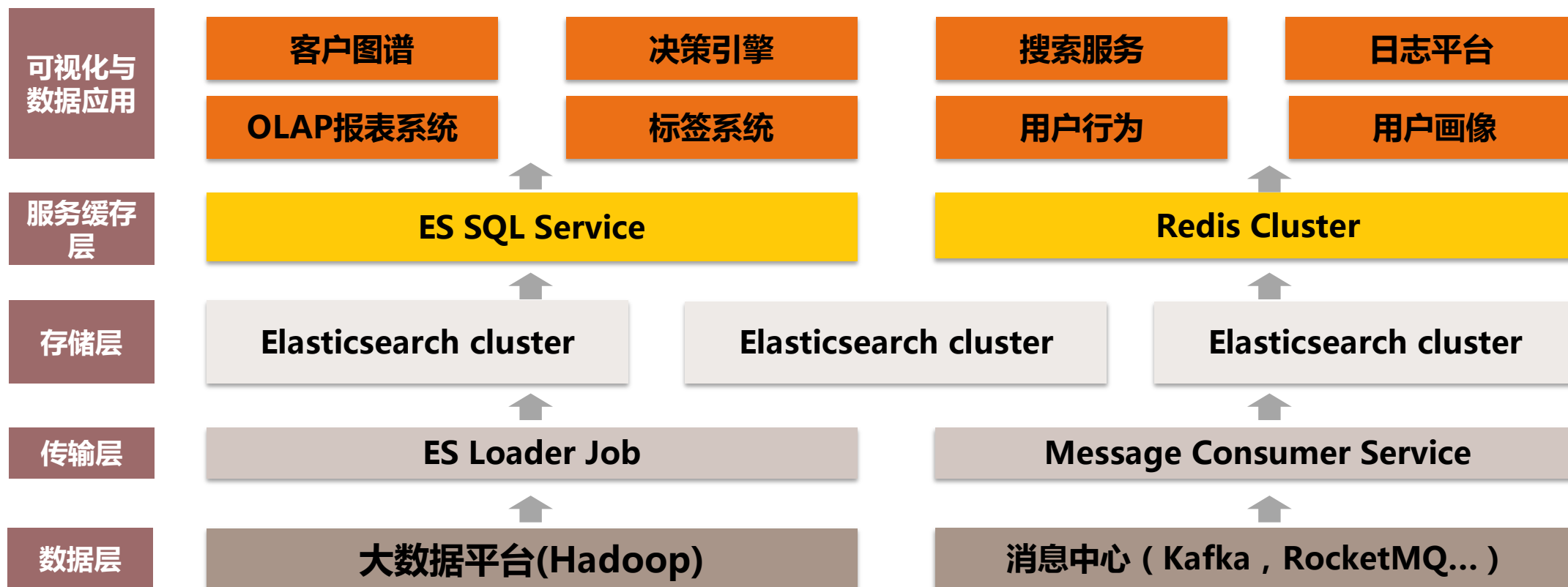
二. ES集群优化

三. 应用介绍

四. 未来规划

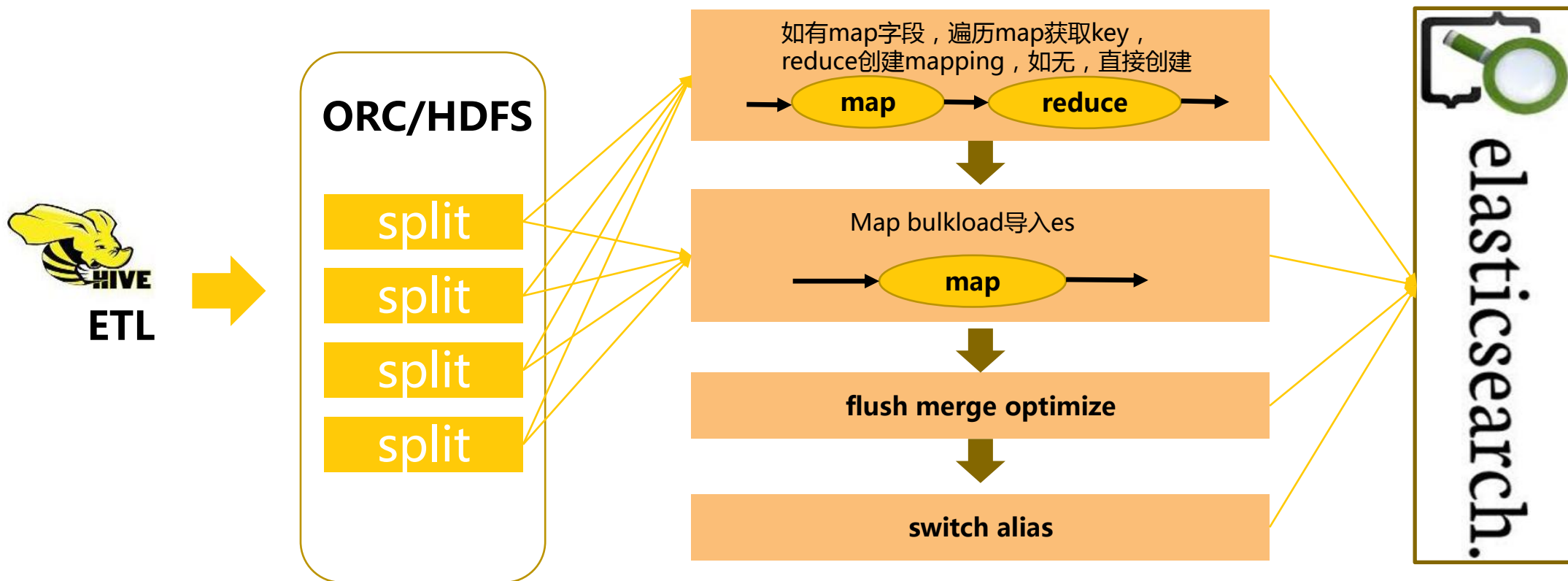
总体架构

基于Elasticsearch高效的搜索和强大的聚合特性，在大数据平台广泛应用，大量大数据服务基于elasticsearch建设，总体容量近200TB，每天新增20TB



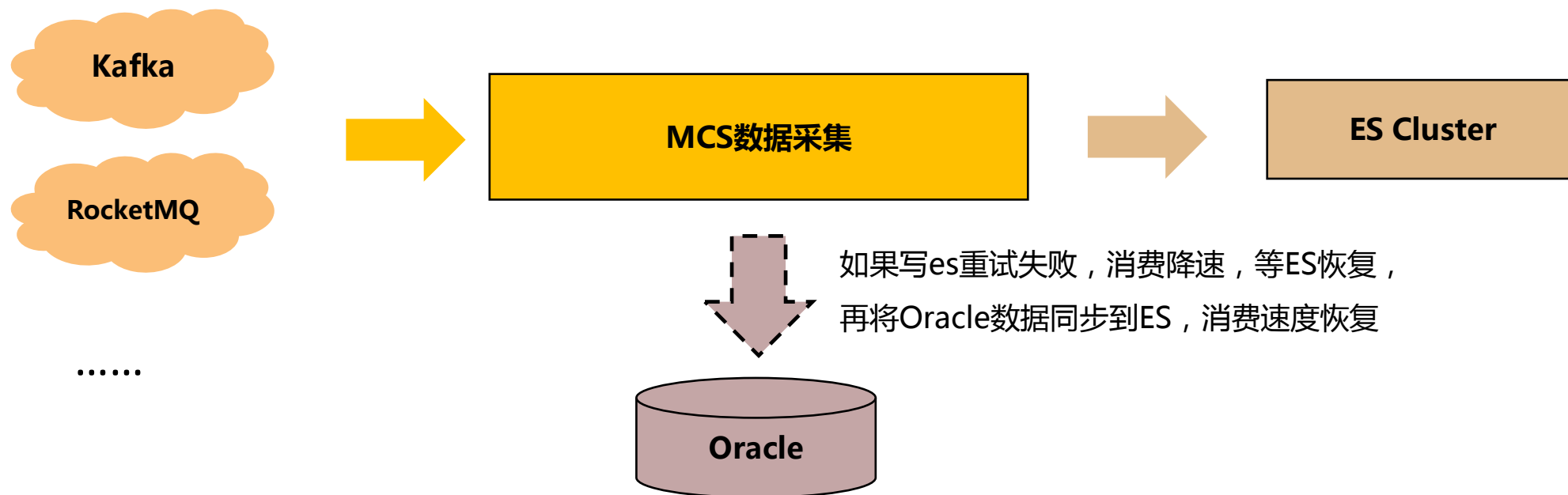
ES Loader Job

- 快速将hive表的数据导入到ES里面
- 根据hive字段创建es mapping，如果hive有map字段，则通过MapReduce创建mapping
- 限流，防止ES集群压力过大
- 优化索引，切换alias



Message Consumer Service

- 通过简单的配置完成实时消息接入，并存到Elasticsearch
- 支持ES故障无消息丢失，写降级临时将数据写到oracle
- 支持字段类型转换
- 支持消息生命周期管理
- 支持mapping新增补全

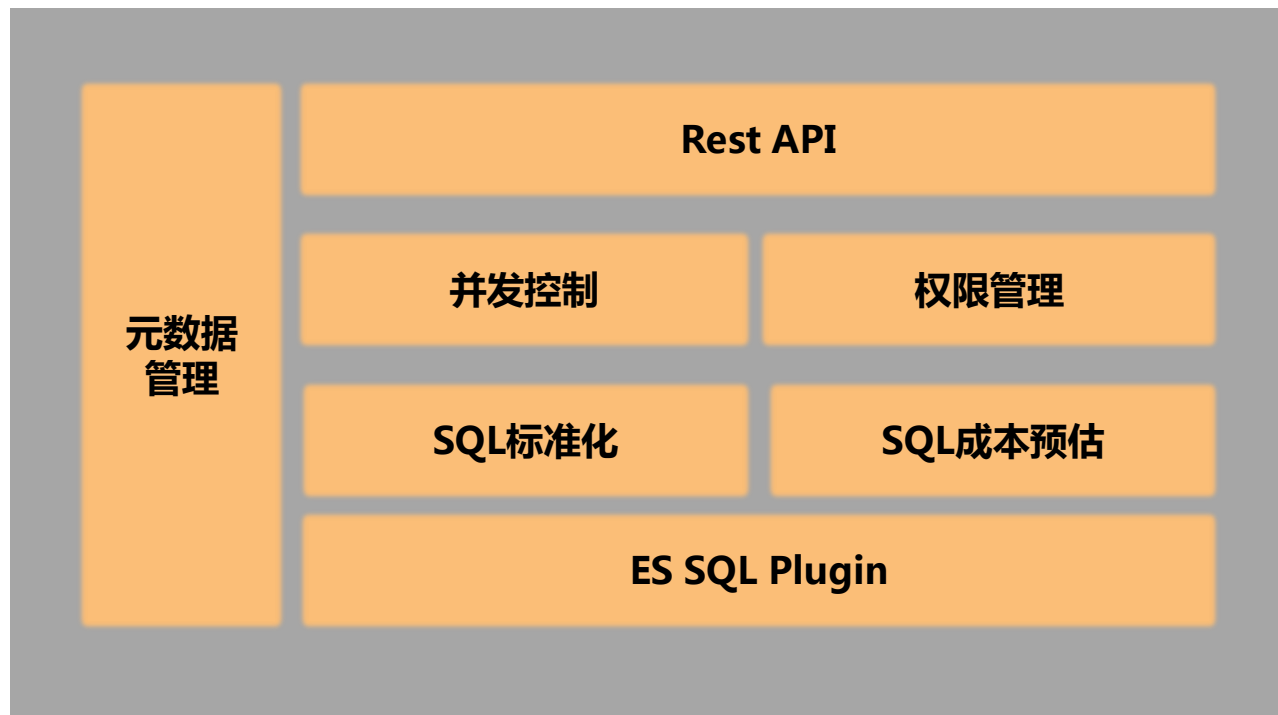


ES SQL Service

将ES SQL插件独立出来，做成服务来调用ES集群，解决如下问题：

- 避免更新插件时重启ES集群，迭代发布时只影响服务
- 方便加熔断机制
- 减小ES协调节点的压力，避免协调节点的内存使用过高

- 支持ES SQL查询（ES旧版本不支持SQL，采用<https://github.com/NLPchina/elasticsearch-sql>，并增加复杂SQL功能）
- 标准SQL支持
- 并发控制，OLAP分析的SQL进行并发控制
- 支持ES SQL权限认证
- 支持ES SQL阻断
- SQL成本预估
 - 维护基数过大的聚合
 - 会造成单节点OOM



ES SQL Service-plugin增加功能

• 语法和功能增强

ES 6.3以前不支持SQL查询，项目从开源

(<https://github.com/NLPchina/elasticsearch-sql>)基础上增加功能：

1. 新增数值/日期/字符等SQL函数
2. SQL函数内嵌任意函数和case when
3. 支持过滤条件使用SQL函数
4. 支持按SQL聚合函数排序
5. 复杂case when语句：case when嵌套函数,函数嵌套case when等
6. 空值检查，修复表达式计算null pointer exception.
7. 解决大宽表字段别名查询造成过多脚本查询限制

和性能问题 (原实现转换成脚本查询，新实现插件中做别名隐射)

• 使用场景：B+自助分析报表计算列

过滤信息

固定日期 日期范围 相对日期

年 季 月 天 年季 年月 年月天

待选

- 2016
- 2017
- 2018
- 2019

已选

计算列

新列名 当日抵押类数量

col(汽融风控超期末抵押月报(BRRPSDATA).当日抵押类数量)

校验 清空 重置

选择列

汽融风控超期末抵...

数据日期
业务分类
当日超期末抵押数量
当日抵押类数量
当日总数量

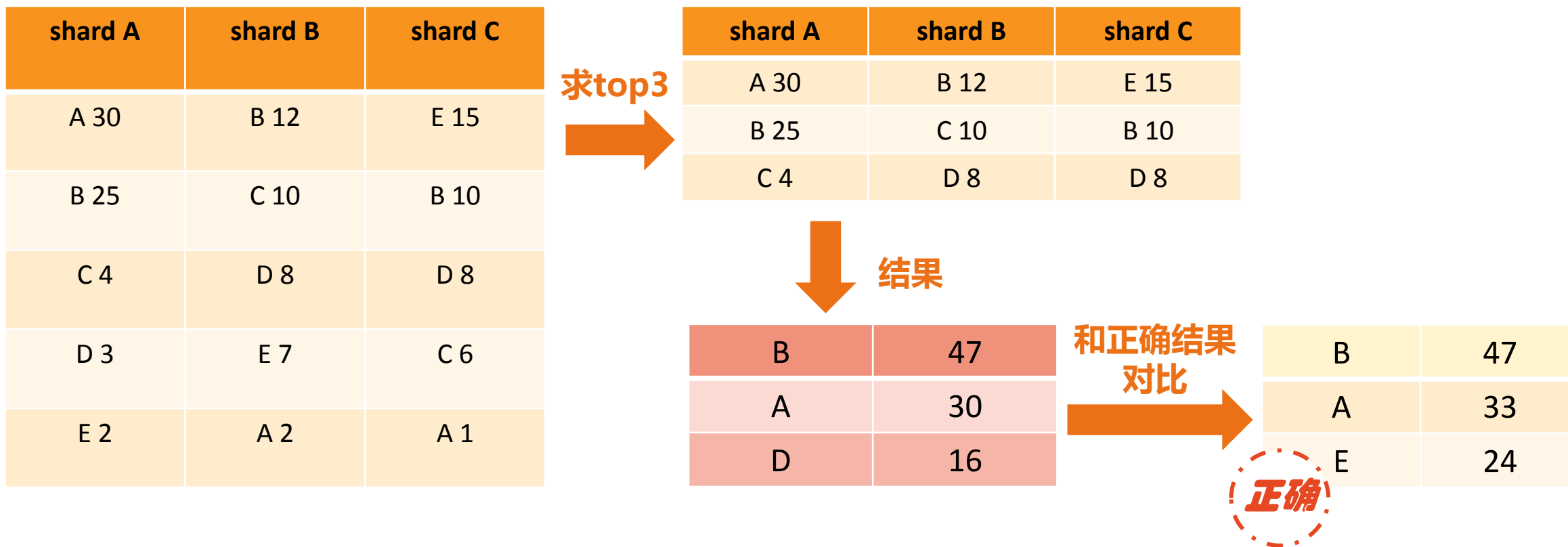
函数

- 日期函数
- 数值函数
- 字符函数
- 日期函数
- 聚合函数
- 类型转换
- 逻辑运算

函数说明

ES SQL Service-慎用terms size

ElasticSearch terms size聚合的时候，如果维度基数大于size，聚合结果求TopN可能是近似值

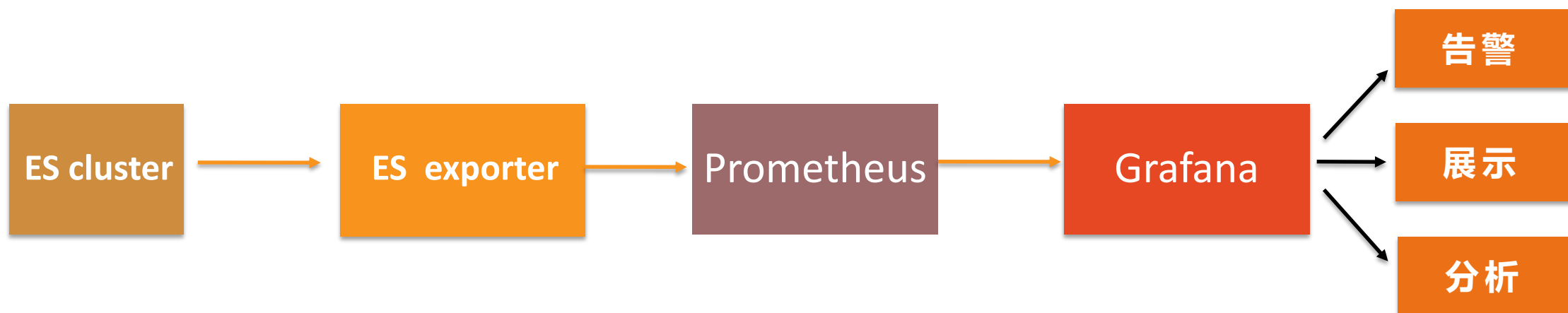
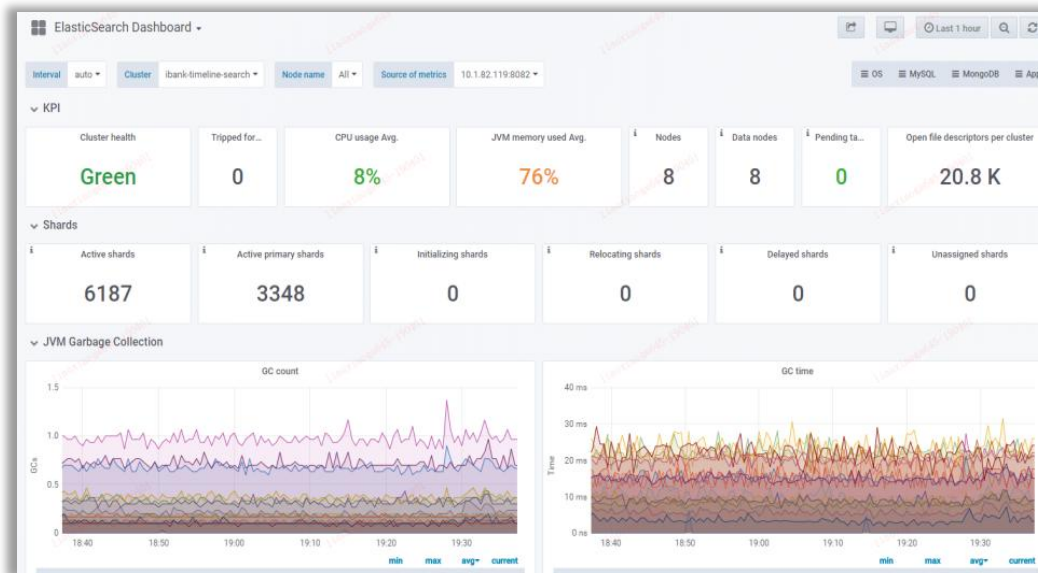


结论 : terms size & shard size 必须超过维度元素个数

Elasticsearch监控报警

- Cluster status报警
- CPU usage & load average报警
- JVM GC报警
- Disk usage报警
- Query & indexing time报警
- Thread pool queued报警

.....



目录

一．总体架构

二．*ES*集群优化

三．应用介绍

四．未来规划

索引优化

- **合理设置字段索引参数**

1. 不需要过滤时可以禁用索引 "index" : false
2. 不需要text字段的score, 可以禁用 "norms" : false
3. 不需要短语查询可以不索引positions "index_options" : "freqs"
4. 禁用全文搜索功能 _all "enabled" : false

- **采用多线程批量提交数据**

1. 使用 multiple workers/threads发送数据到ES
2. 每个bulk请求不宜过大, 避免导致OOM
3. 遇到EsRejectedExecutionException则说明IO压力过大, 需要调整线程或bulk size

- **增加Refresh间隔, 减少副本数量**

1. 增加refresh时间间隔可以避免生成过多的segment, 从而减少合并压力
2. 同步完成后再加副本可以避免主副分片同步带来的压力

- **硬件升级**

1. 使用SSD硬盘, 提高读写性能
2. 使用性能更好的CPU, 高并发
3. 使用大内存, 索引缓冲默认会占用JVM 10%的内存空间



查询优化

- **尽量避免使用script**

1. 尽量避免使用script，如要使用则可以选择painless & expressions引擎
2. 避免大量动态脚本产生，因为脚本需要编译才能执行。

- **避免大查询和大聚合**

1. 大查询或者大聚合会导致ES响应慢，还会占用大量JVM内存，从而导致其他任务堆压
2. 建议在服务层通过程序来组装业务，并及时阻断查询size或者笛卡尔积过大的请求

- **避免深度分页**

1. 深度分页导致大批数据返回到协调节点，协同节点一共会受到 $N * (From + Size)$ 条数据，然后进行排序
2. 使用 Elasticsearch scroll 高效滚动的方式来解决深度分页问题

- **使用routing**

可直接根据 routing 信息定位到某个分片查询，避免查询所有的分片，以及在协调节点上无需排序

- **加大线程池阻塞队列长度**

修改thread_pool.bulk.queue_size，默认为500，可适度调整到1000-2000

- **设置Cache参数**

1. QueryCache: 过滤查询过多则可以调大indices.queries.cache.size
2. FieldDataCache: 聚类或排序场景较多则可以调大indices.fielddata.cache.size



OS & JVM 优化

- **禁用swapping**

内存交换到磁盘对服务器性能来说是致命的，通过设置swappiness = 0来禁用该功能

- **文件描述符和MMap**

Lucene使用了大量的文件，Elasticsearch 在节点和 HTTP 客户端之间进行通信也使用了大量的套接字，这需要足够的文件描述符，设置sysctl -w vm.max_map_count=262144

- **单个节点内存不要超过32G**

1. JVM 在内存小于 32 GB 的时候会采用一个内存对象指针压缩技术
2. 大指针在主内存和各级缓存之间移动会变得缓慢
3. 机器内存大，可以采用单机多节点部署

- **至少留一半内存给LUCENE**

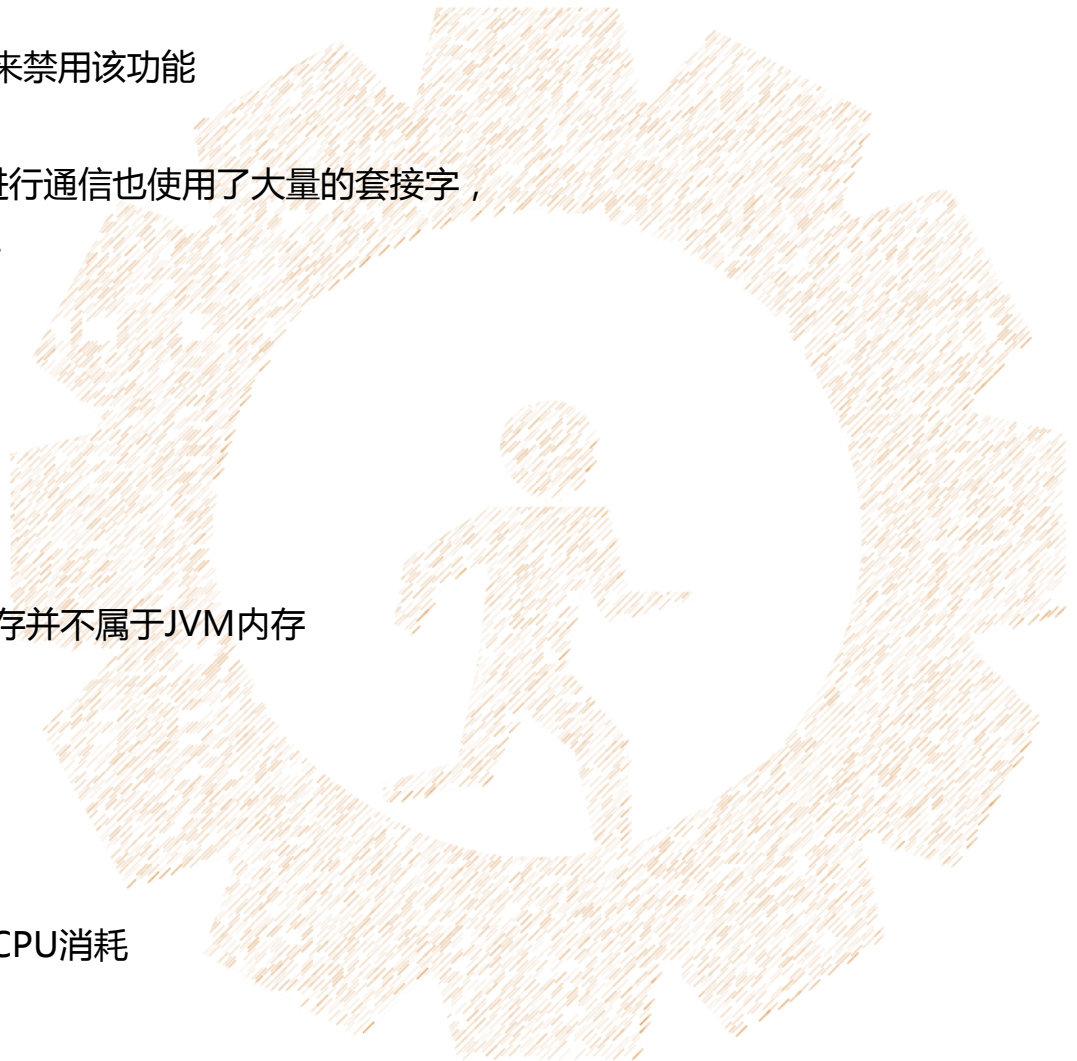
Lucene可以利用操作系统底层来缓存数据结构，以便快速访问，这些内存并不属于JVM内存

- **单机多节点部署避免主副分片被分配到同一物理机**

设置cluster.routing.allocation.same_shard.host:true

- **使用G1垃圾回收器**

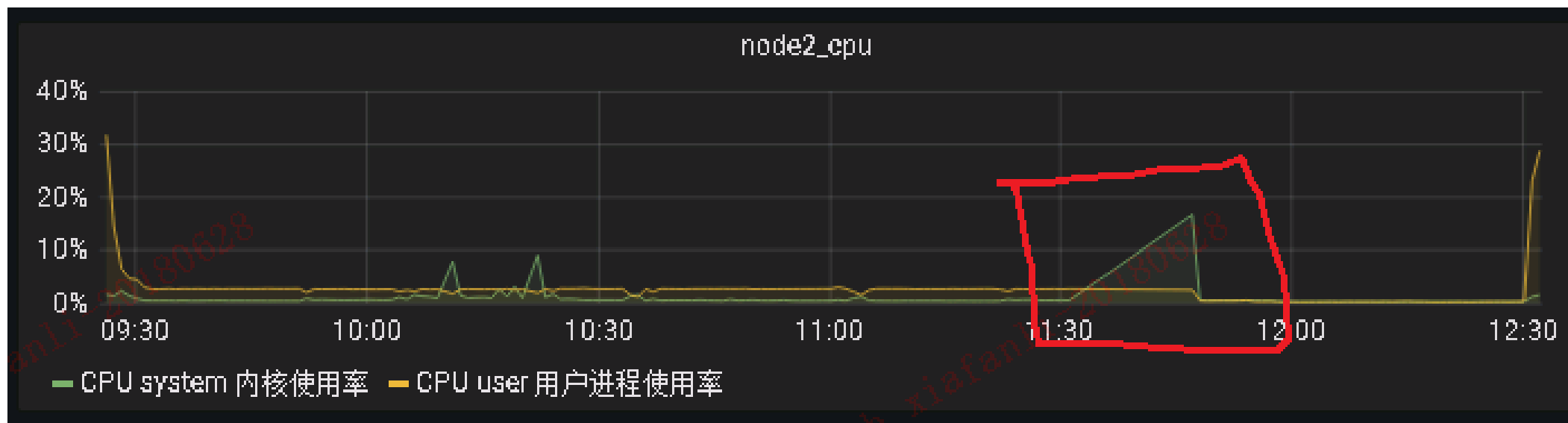
1. 存在单个索引数据非常大的集群，可以考虑使用G1替代CMS
2. 设置MaxGCPauseMillis参数减少GC停顿时间，但如果过小则会带来CPU消耗



OS & JVM 优化

- 禁用numa模式，设置vm.zone_reclaim_mode=0

Elasticsearch集群中，经常偶节点出现sys cpu占用过高问题，出现问题期间，无法登录及操作,其它活跃进程占用CPU都显示100%；压测集群过程中，出现了SYS 飙升问题，但是查看atop这个时间段的数据，也因为atop进程hang住，导致了无法采集数据



目录

一．总体架构

二．ES集群优化

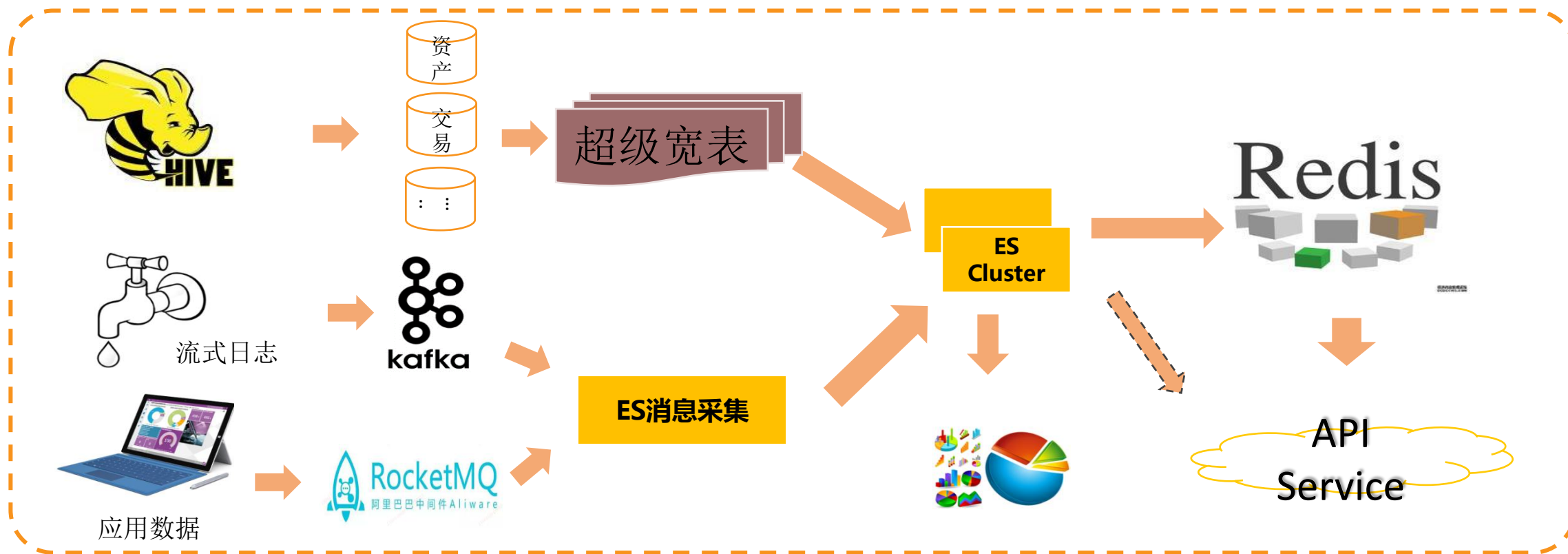
三．应用介绍

四．未来规划

标签系统

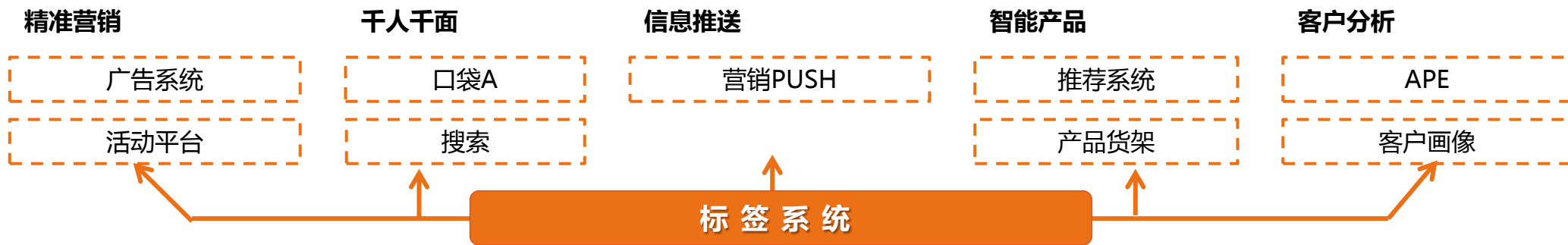
技术架构 离线、实时、流式，是完整数据中心

- **离线**：通过Hive计算，形成各个对象的超级款表，导入ES
- **实时流式日志数据**：通过ES消息采集初步聚合汇入ES
- **应用系统状态**：通过RocketMQ异步推送给指标



标签系统

根据数据，生产标签，深入了解用户和产品



客户标签库

JIANGQIANG061 下午好! 退出

客户标签 外部标签 组合标签 收藏标签

AUM

首次入金日余额

万元户近三个月AUM...

T-3个月月日均AUM

T-2个月月日均AUM

T-1个月月日均AUM

AUM余额月累计

上月月末AUM余额

客层

客户银行管理资产AU...

过滤条件设置

性别 = 1 且 客户资产分层 = '04'

拖动标签可编辑其位置 (注: 脚本优先级从左往右依次递减) 清空

且 添加数值

四则运算条件配置

首次入金日余额 * 加(+) 100 大于(>) 10

清空

生成组合标签 预览

ID	标题	名称	业务描述	所属主题
19101	加3奖励券未激活促达标 微信0830	加3奖励券未激活促达标 微信0830	加3奖励券未激活促 达标微信0830	客户 主题
19100	50元商城券未激活促达 标微信0830	50元商城券未激活促达 标微信0830	50元商城券未激活促 达标微信0830	客户 主题
19099	50元还款金领券未激活 促达标微信0830	50元还款金领券未激活 促达标微信0830	50元还款金领券未激 活促达标微信0830	客户 主题

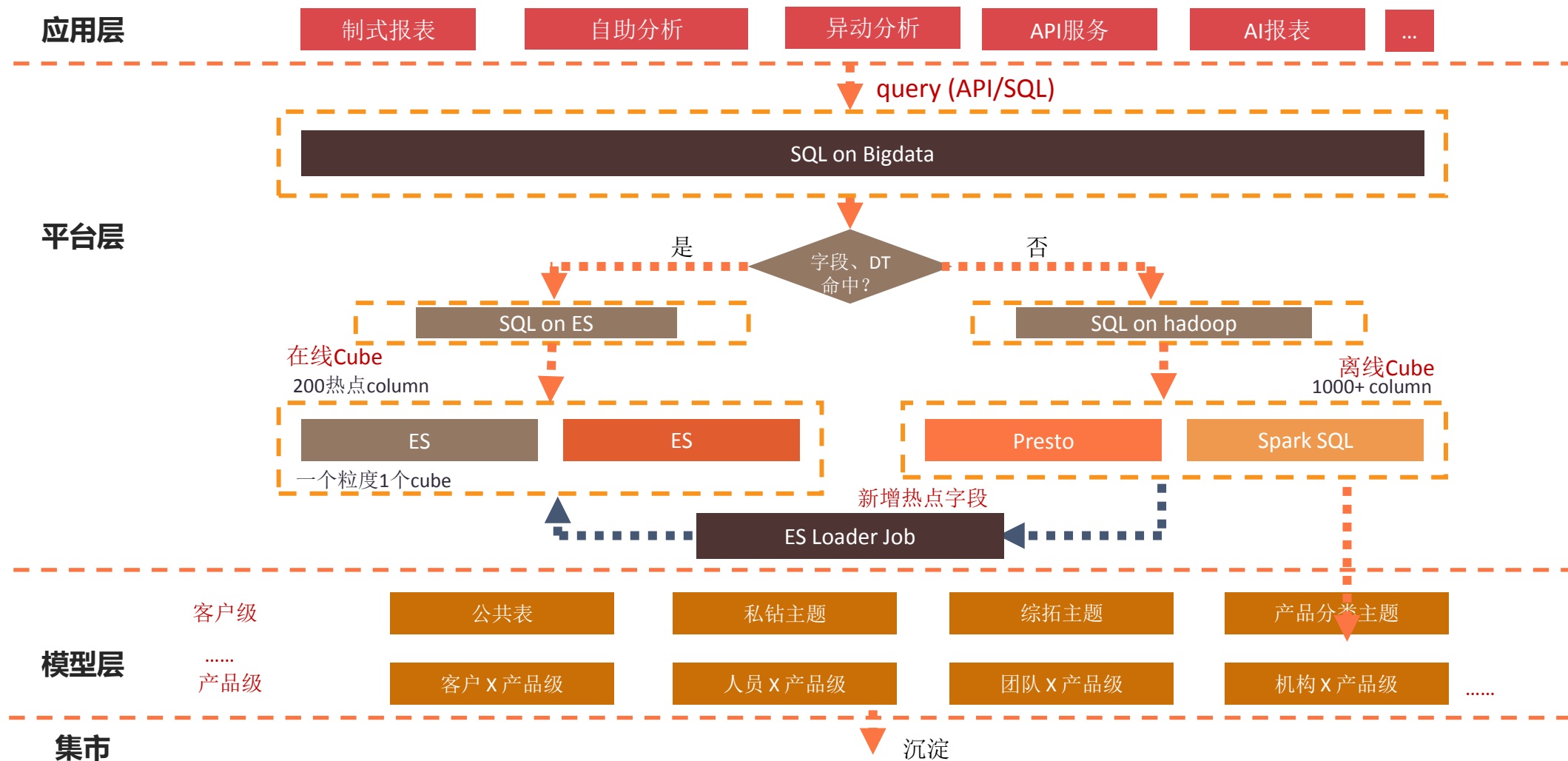
标签系统

- 标签系统目前主要建立的是客户主题标签，目前分析预测类标签30余个，客户基本维度数据600多个，业务场景标签500多个（在线300个左右）。



基于ES SQL的OLAP报表服务

金融业务极其复杂，cube的维度繁多，维度基数不大，前期kylin对维度个数限制比较大，所以采用ES构建在线热点cube，提升拖拽体验



基于ES SQL的OLAP报表服务

特点

◆自给自足

业务人员无需IT技能，
通过简单的托拉拽就能生成报表；

◆秒级计算

基于强大的计算引擎
给用户极致的查询体验；

◆丰富模型

预先准备数据模型，从业绩追踪、经营分析、客群分析等场景提供全面支持；

◆一键分享

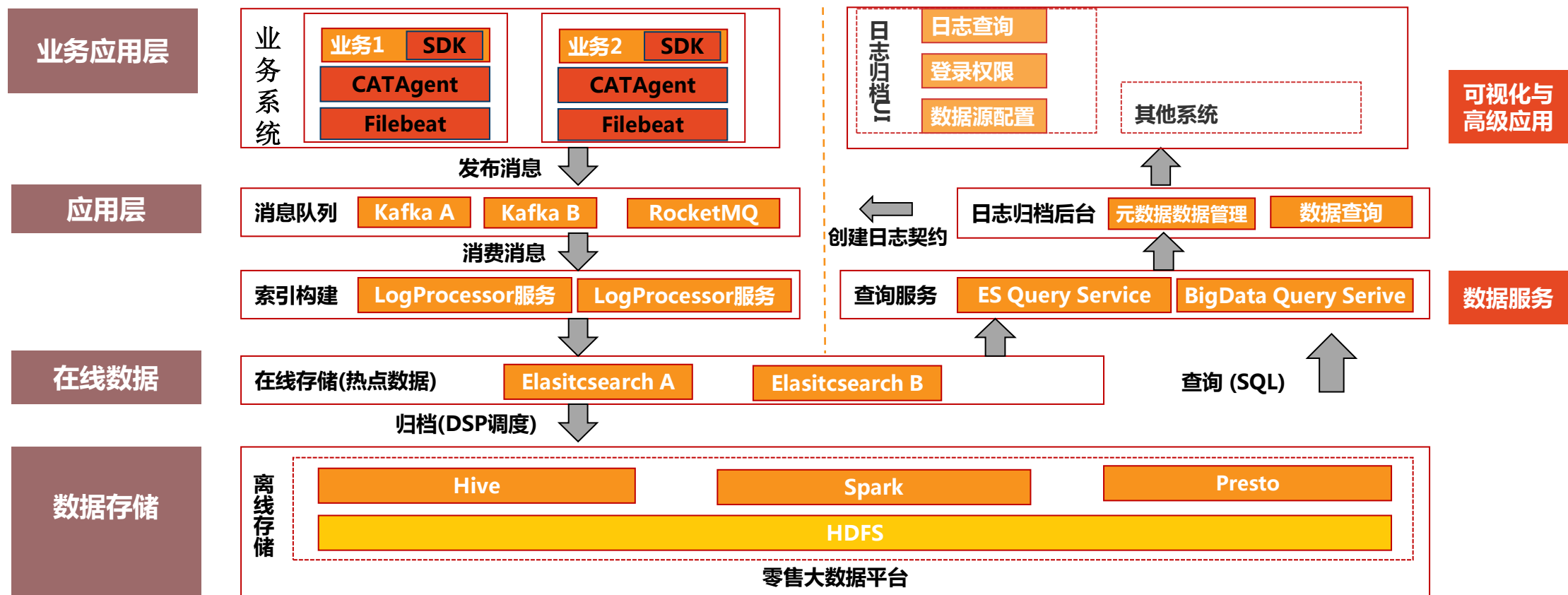
可将自己的报表和分析成果一键批量分享给其他同事；

The screenshot shows a software interface for an OLAP reporting tool. On the left is a '数据配置区' (Data Configuration Area) with a search bar and a list of data fields. The '行' (Rows) section contains '客户归属机构区域' and '客户归属机构一级'. The '列' (Columns) section contains '数据日期(年月天)', '本月客户资产...', and '存款余额(合计)'. The '度量' (Measures) section contains '存款余额(合计)'. The '过滤' (Filters) section contains '本月客户资产分层'. On the right is a pivot table titled '客户全景' (Customer Overview) with a '切换为列表' (Switch to List) button. The table has columns for '客户归属机构区域', '客户归属机构一级', and three customer categories: '零资产客户', '小额客户', and '大众千元客户'. The rows list various branches under '南区事业部' (South Area Department), such as '深圳分行汇总', '广州分行汇总', etc., with corresponding deposit balance values.

客户归属机构区域	客户归属机构一级	零资产客户	小额客户	大众千元客户
		存款余额(合计)	存款余额(合计)	存款余额(合计)
南区事业部	深圳分行汇总	6万	5万	2万
	广州分行汇总	3万	9万	1万
	东莞分行汇总	1万	6万	1万
	佛山分行汇总	1万	4万	1万
	昆明分行汇总	1万	4万	1万
	惠州分行汇总	1万	2万	1万
	海口分行汇总	1万	2万	1万
	珠海分行汇总	1万	1万	1万
	中山分行汇总	1万	2万	1万
	贵阳分行汇总	1万	7万	1万
	南宁分行汇总	1万	4万	1万
	南宁分行 (ZT) 汇		0	
上海分行汇总		1万	9万	1万
南京分行汇总		7万	8万	1万

业务日志归档平台

传统银行业务日志归档基本上是存到Oracle库的，为了减轻数据库压力和方便业务系统将日志归档，所以基于Elasticsearch将日志归档，并能提供实时日志的查询功能



目录

一．总体架构

二．ES集群优化

三．应用介绍

四．未来规划



升级版本至7.X版本， 有很多新特性

1. 稀疏性 Doc Values 的支持，可以避免稀疏字段带来的性能和硬盘空间浪费
2. 根据节点的负载高低来排序，负责高的节点接收到的任务将减少
3. 更快的查询和索引速度，故障分片恢复时间也变得更快了
4. 预排序，即在索引的时候将生成好排序信息，提升搜索或聚合的性能



MASTER节点 不再接受查询/索引请求

1. 可以将master节点独立出来，不再接受查询/索引请求，只负责管理集群元数据
2. CPU、内存和硬盘配置可以低一些，也可以采用虚拟机部署



读写分离

1. 集群分为read和write两个分区
2. index的数据放到write分区，同步完后将数据迁移到read分区
3. search请求read分区
4. 读写分离可以避免同步数据以及段合并带来集群性能抖动



金融大数据团队 期待**您**的加入



181276056@qq.com

打造金融数据新生态 助力平安零售强转型

大数据团队目标是最领先的大数据技术建设银行零售数据中心及AI智能服务平台，深入探索金融数据，为业务提供技术和数据支持，最大限度发挥银行数据的价值。



谢谢