



腾讯万亿级Elasticsearch 技术解密

姜国强

腾讯 - 云架构平台部



议程

1 ES在腾讯的应用场景

2 遇到的挑战

3 ES优化实践

4 未来规划及开源贡献

ES在腾讯的应用场景

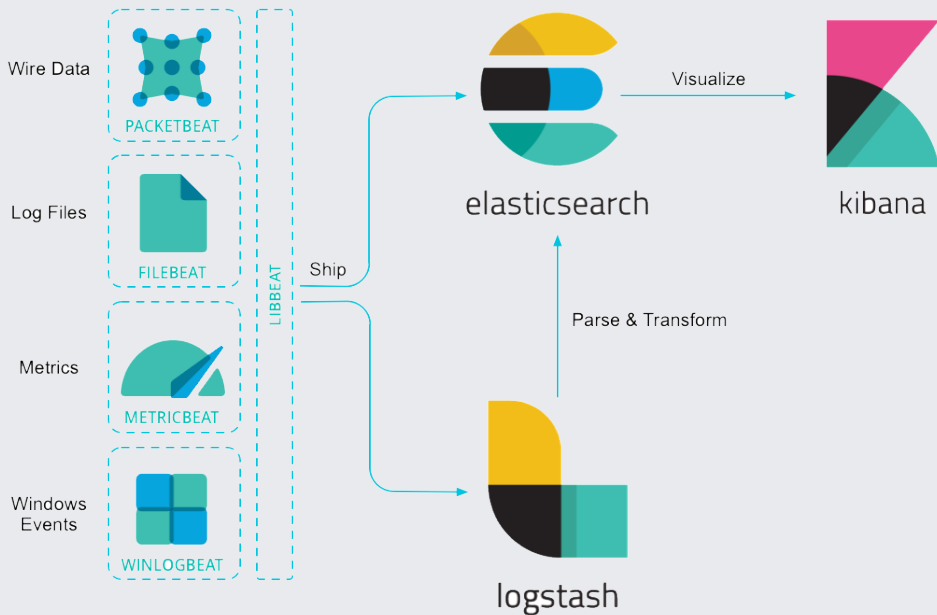
日志实时分析

典型场景

- 业务日志：用户行为
- 运营日志：慢查询、异常日志
- 审计日志
-

主要特点

- 完整的解决方案，易运维
- 实时性：从日志产生到可访问，十秒级
- 全文搜索：基于倒排索引，搜索分析灵活
- 交互式分析：千亿级日志，搜索秒级响应



搜索服务

典型场景

- 商品搜索
- APP搜索
- 站内搜索
-

主要特点

- 高性能：高并发、低延迟
- 强相关：搜索结果高度匹配用户意图
- 高可用：可用性达9999，跨机房容灾



时序数据分析

典型场景

- Metrics
- APM
- 物联网数据
-

主要特点

- 高并发写入：千万级写入
- 多维分析：多维度统计分析
- 高性能：十毫秒级

传统监控

智能硬件

应用监控

工业传感器



丰富的业务需求

公有云

丰富的场景

大量中小客户
不同的应用场景
不同的熟悉程度

内部云

超大规模

600~节点
千万级写入
十万级查询

私有云

标准化/自动化

完全隔离的环境
标准化交付
自动化运维

遇到的挑战

搜索类业务

代表场景

- 电商、站内搜索等在线服务

用户需求

- 性能更好
- 可用性更高

挑战

- 高性能
- 高可用

性能

QPS
20w+

平响延时
<20ms

P95延时
<100ms

可用性

SLA
4个9

容忍
单机故障

容忍
网络故障

时序类业务

代表场景

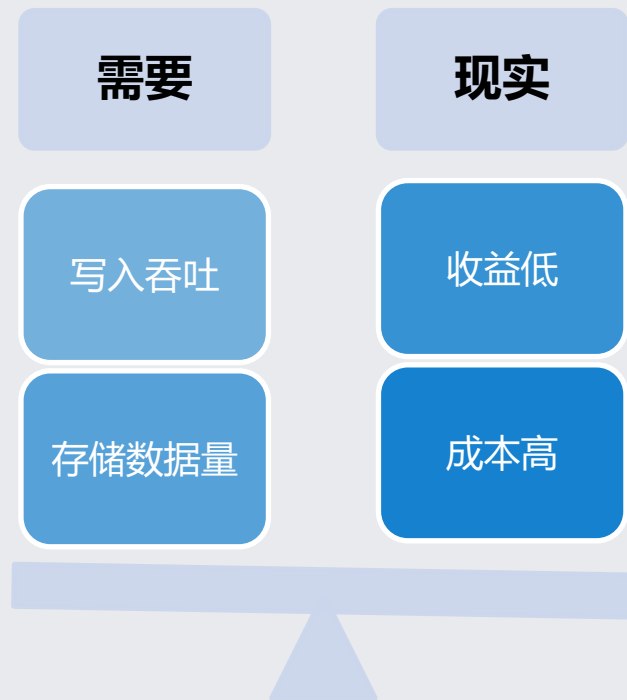
- 日志、Metric、APM等时序数据

用户需求

- 支持超高写入吞吐
- 存储海量数据

挑战

- 计算成本
- 存储成本



ES优化实践

高可用优化

健壮性不足

- 集群崩溃：异常查询、过载
- 扩展瓶颈：节点、元数据
- 集群不均：节点、硬盘

容灾方案欠缺

- 集群不可用：网络故障
- 数据丢失：自然灾害
- 误操作

系统缺陷

- 滚动升级缓慢
- Master堵塞
- 分布式死锁
-

高可用优化 - 解决方案

系统健壮性：分布式系统共性问题

- 服务限流：容忍常见异常，保障服务质量
- 可扩展性：支持千级集群、50w分片
- 集群均衡：节点、多盘间分片均衡

容灾方案

- 备份回档：保障数据可恢复
- 跨可用区容灾：容忍单机房故障
- 垃圾桶机制：实例销毁保护

缺陷修复

系统健壮性

服务限流

可扩展性

集群均衡

容灾方案

备份回档

跨可用区
容灾

垃圾桶机制

缺陷修复

滚动重启
优化

Master
堵塞

分布式
死锁

.....

高可用优化 - 服务限流

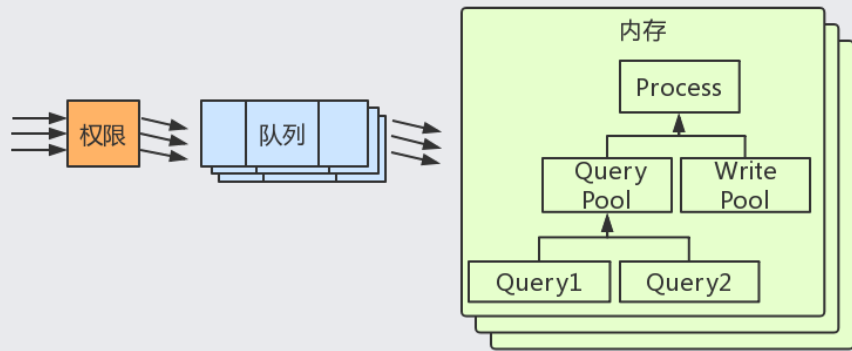
方案

- 权限：防止攻击、误操作
- 队列：优先级、拥塞优化
- 内存：全链路 + 多级内存限制 + 精准评估
- 多租户：CVM / Cgroups

优势

- 限流准确、适应性强
 - 大查询、异常查询、压力过载等
 - 单点故障、网络分区等

混沌测试



依赖	环境变量 未修改、配置	包 错误、损坏	配置 错误、误删
应用层	进程 Kill、OOM、Full GC	线程 并发、优先级	
系统层	系统 Hang	内存 脏、Swap	CPU 抢占
硬件层	硬盘 满、慢、坏、不可读写	网络 丢包、超时、断网	

成本优化

成本分析

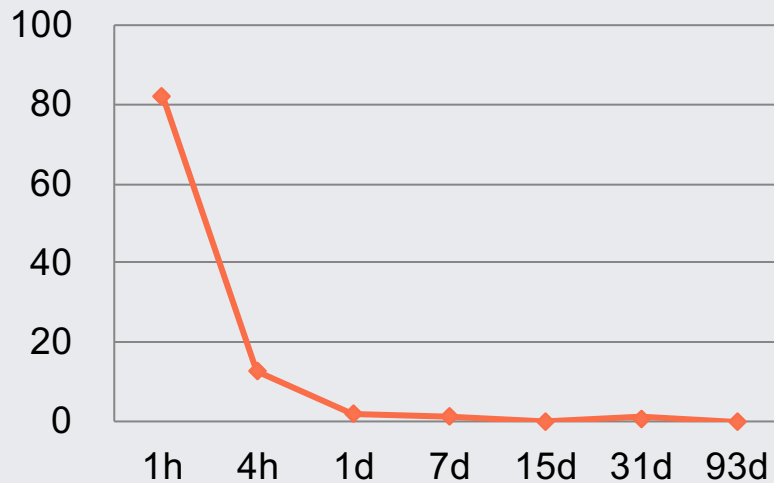
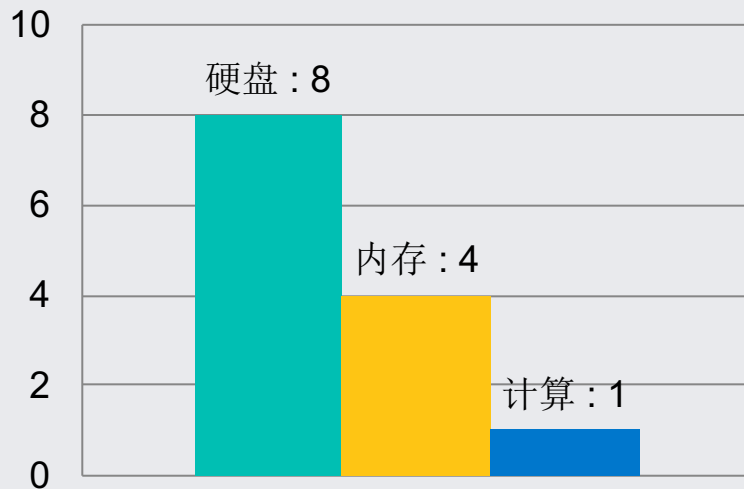
- 代表场景：日志等时序数据
- 成本比例关系

主要瓶颈

- 硬盘
- 内存

特性分析

- 冷热特性明显：访问近多远少
- 历史数据查询聚合结果
- 长时间范围查询性能差



成本优化 - 解决方案

硬盘成本

- 冷热分离：使用混合存储平衡成本、性能
- Rollup：预计算 换取 存储、性能
- 备份归档：更廉价的存储系统
- 更多优化方式：存储裁剪、生命周期管理等

内存成本

- LRU Cache：提升内存利用效率
- Off Heap：降低堆内存，提高节点规格

存储优化

冷热分离

Rollup

备份归档

存储裁剪

生命周期
管理

内存优化

LRU
Cache

Off Heap

内存池化

成本优化 - Rollup

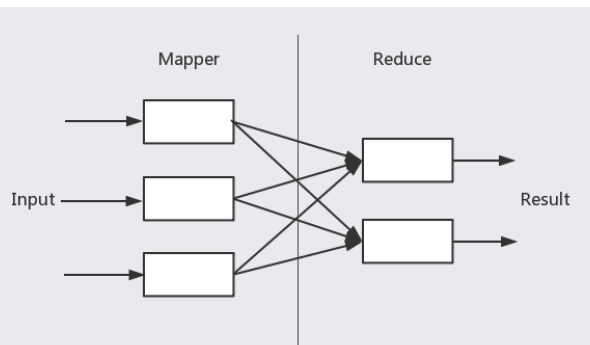
方案

- 类似 Cube / 物化视图
- 预计算：换取 成本和性能（数量级）

1	region	host	time	cpu_usage
2	gz	10.11.12.13	2017-07-01 10:00:00	20
3	bj	14.15.16.17	2017-07-01 10:00:00	20
4	gz	10.11.12.13	2017-07-01 10:00:10	30
5	bj	14.15.16.17	2017-07-01 10:00:10	40
6	gz	10.11.12.13	2017-07-01 10:00:20	25

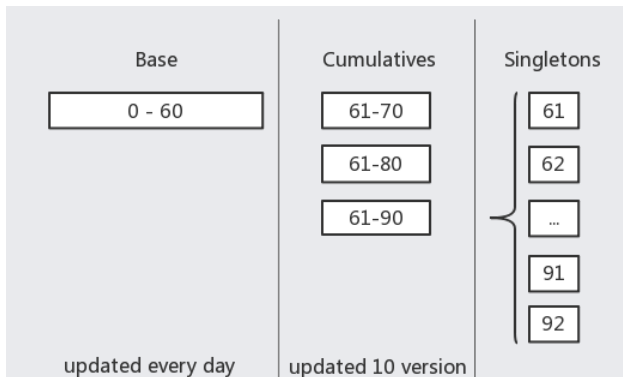


1	region	host	time	cpu_usage
2	gz	10.11.12.13	2017-07-01 10:00:00	25
3	bj	14.15.16.17	2017-07-01 10:00:00	30



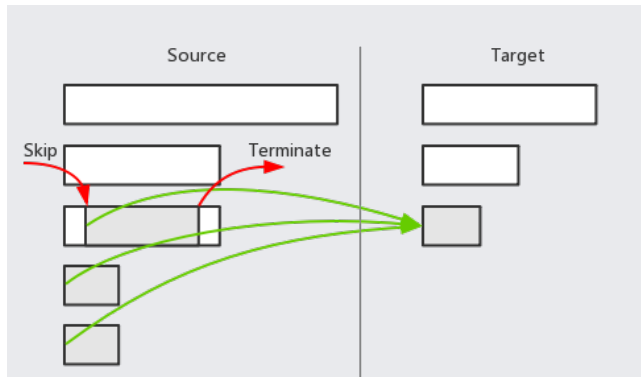
数据分析系统

- 数据无要求
- 周期性外部计算：读取全量数据
- 计算开销、维护成本高



Google MESA

- 统计数据模型，数据排序
- 底层Compact：多路归并
- 多路写入后归并，计算开销高



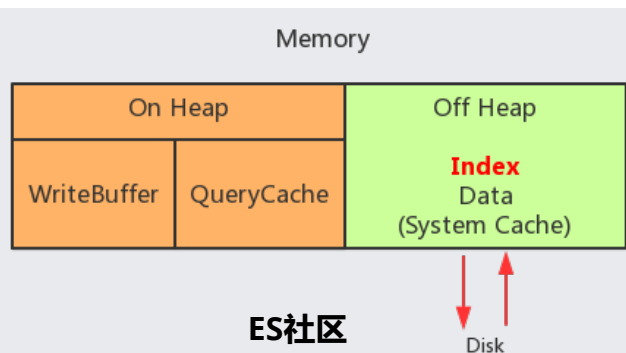
最终方案

- 数据排序
- 流式查询，多路归并
- 计算开销 < 10% 写入，内存可控

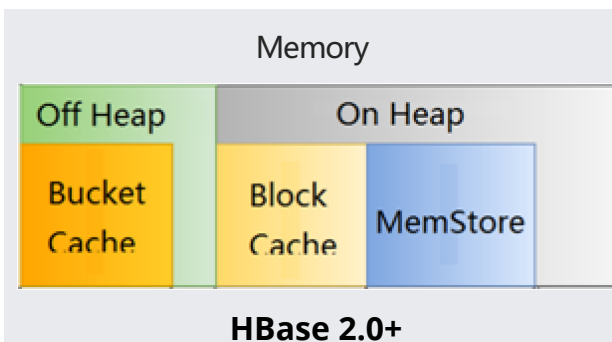
成本优化 - 内存优化

方案

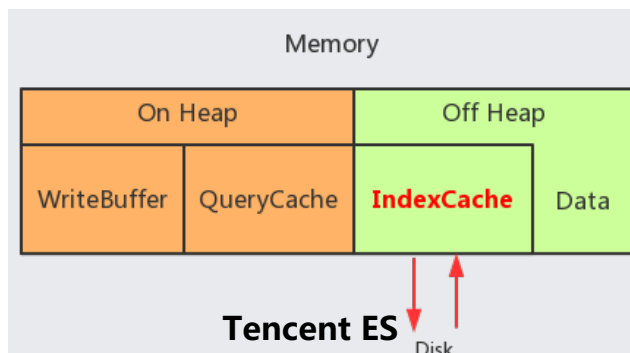
- LFU Cache提升内存利用效率，大小动态可调
- Off Heap降低堆内存使用
- 堆外访问链路优化，降低损耗



- 按需加载，依赖系统缓存
- 查询性能倍数级衰退



- 重点推介Off Heap
- 堆外内存性能接近堆内



- 内存利用效率提升80%，充分利用硬盘
- 查询性能损耗不超过2%
- 降低GC开销30%

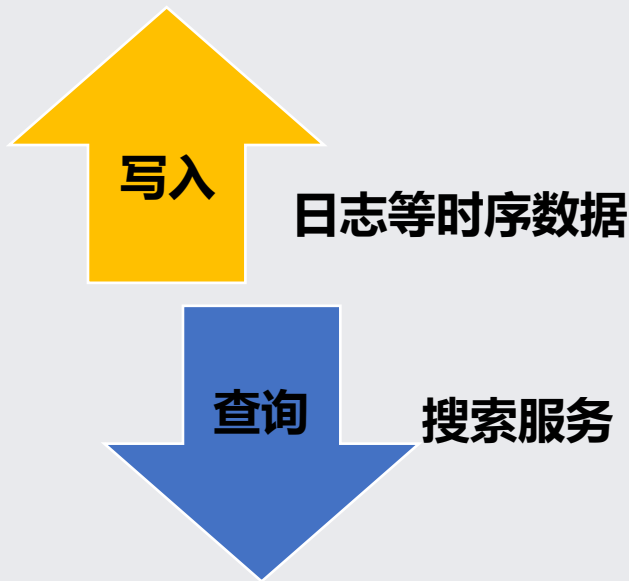
性能优化

写入

- 代表场景：日志、监控等时序数据
- 高并发写入：1000w/s级
 - 自带主键写入性能衰减1+倍
 - CPU无法充分利用

查询

- 代表场景：搜索服务
- 高并发、低延时查询：20ms，10w QPS
 - 平响需要深度调优，并发不高
 - 容易产生毛刺：GC、执行计划不优等



性能优化方案

写入：时序场景1000w+

- 优化主键去重，性能提升45%
- 优化Translog Sync机制，性能提升20%
- 向量化执行，减少分支跳转、指令Miss

查询：搜索服务平稳低延迟

- 优化Merge策略、冷数据自动Merge
- 基于索引范围的查询剪枝
- 基于CBO模型的执行计划优化

新硬件

写入优化

主键去重

资源竞争

向量化

查询优化

Merge策略

查询剪枝

执行计划

新硬件

AEP

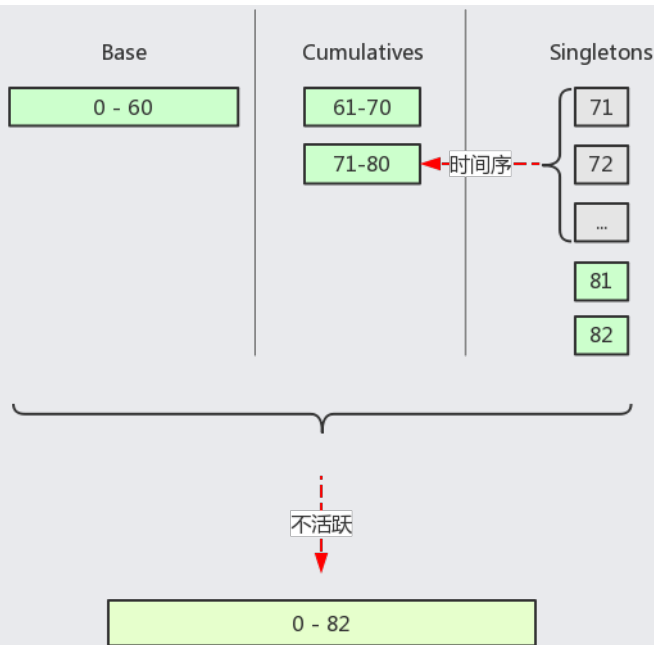
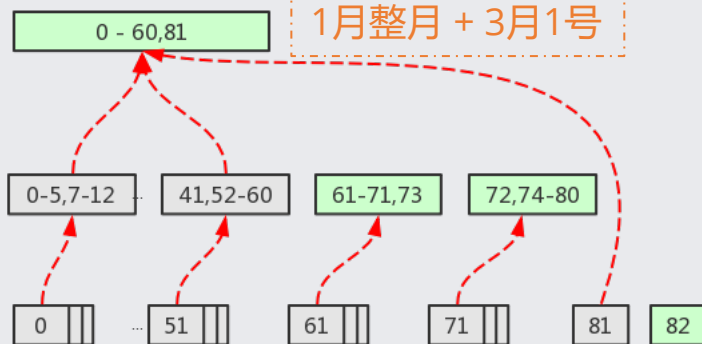
Optane

QAT

性能优化 - Merge策略

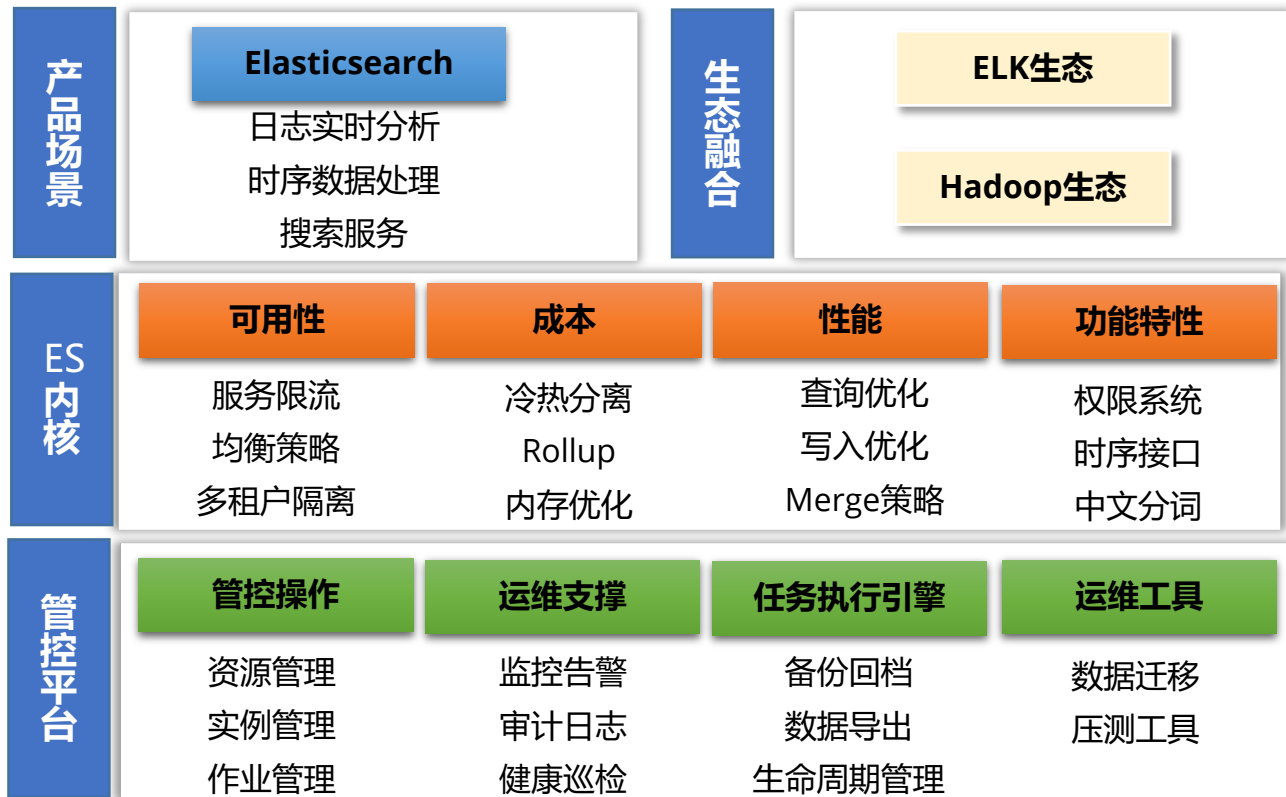
方案

- 原生策略：大小相似性 + 最大上限
- 时间序Merge**：增加时间邻近特性，方便查询裁剪
- 冷数据Merge**：提高查询性能，降低资源开销
- 效果**：搜索场景性能提升1倍



未来规划及开源贡献

产品持续建设



长期探索

大数据图谱

目标数据量递减，响应延时递减

Data Engineering

Batch



Stream



Data Discovery

Analytic

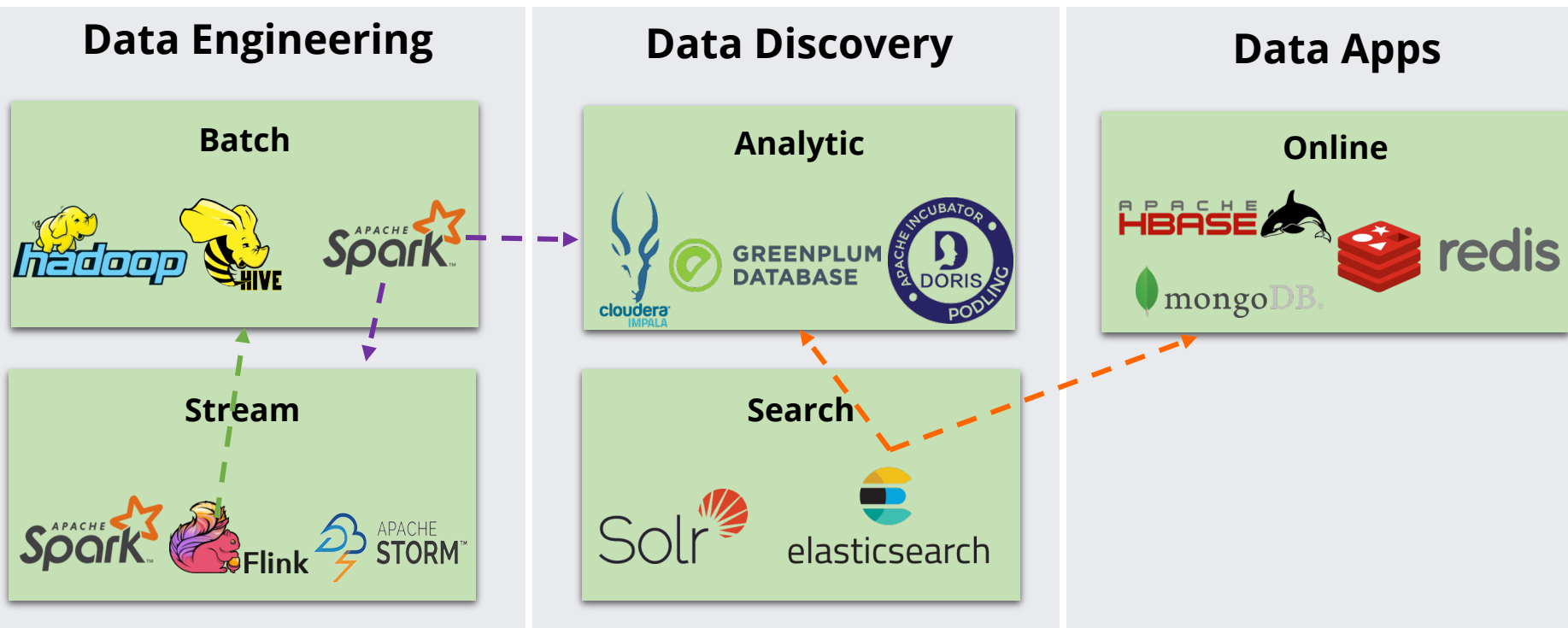


Search



Data Apps

Online



开源贡献

开源社区

- 近期提交了 10+ PR 至开源社区，其中 7 个已合并
- 被合并的 PR 涉及写入、查询、集群管理等模块
- 部分优化和官方 dev 协作完成

内部开源

- 组建公司内 Elasticsearch 开源协同小组
- 内部团队协作，共建 Elastic 生态

开源收益

- 降低分支维护成本
- 更深入的内核把控能力
- 建立在开源社区的技术影响力
- 共同推进Elastic生态持续、快速发展



We have recently made a major and ingenious improvement to Elasticsearch, which was proposed by a developer of Tencent. This improvement makes Elasticsearch Some types of write speeds have increased by about 20%, and we are very much looking forward to continuing this good relationship with Tencent Cloud.

Shay Banon  elastic

Thanks

