打造以ES为核心的系统

虞冰

2015 @ ES Shanghai Meetup

Agenda

1. 为什么以搜索为核心而不是数据库

2. 作为架构负责人需要考虑什么

3. 全公司推行前还需准备好什么

About Me

虞冰

圈内花名：小排

2005.10 ~ 2015.2

Intern ~ Director @ Baixing.com

2015.2 ~ Now

Co-Founder @ 蛋壳公寓

放心，今天不谈创业

在百姓我们做到的结果

核心集群 @Baixing

- 20+ 服务器，1.5 TB+ 内存 for JVM
- 每天1KW 条数据变更，总Doc数2亿+
- 每次数据变更后1秒左右前台可查到
- 高峰时间段 2W QPS，平均响应时间2~5ms
- 多个不同Routing规则的子集群

为什么要以“搜索”为核心打造系统?

查询复杂 & 全文搜索 & 海量数据

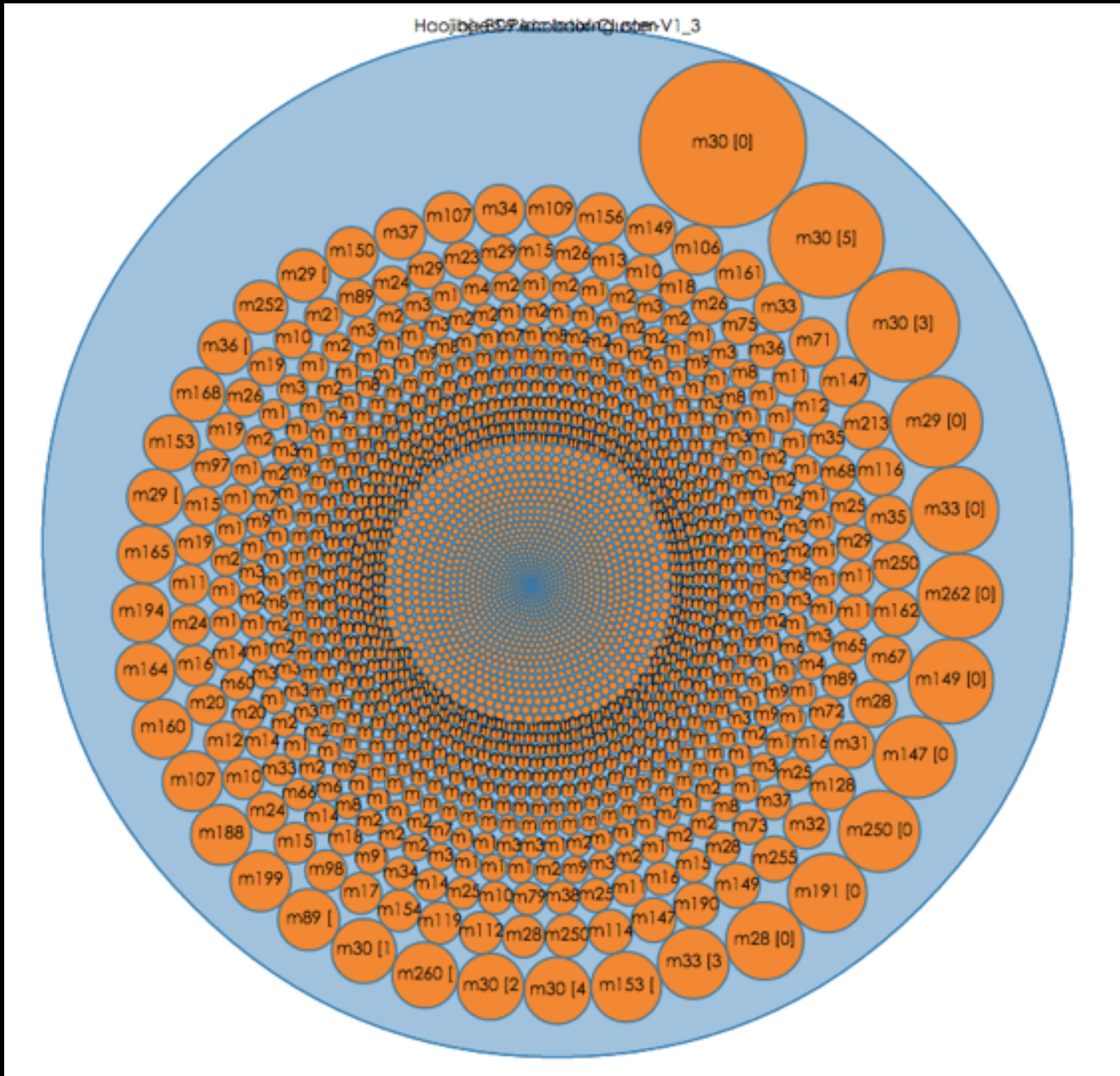
数据库性能严重跟不上！

2个最重要的问题

1. 如何确保ES高性能?
2. 如何保证整个系统稳定?

高性能

- 充分利用内存
- Routing为王
- Shard不怕多，尽可能分散。



整体高性能

=

每次查询的计算成本最小化

=

需要的节点尽量少

+

用到的Index尽量小

思考题

ES是如何从10个Shard合并出from:200, size:10 结果的?

Auto-Sharding、Distributed
很多时候并不如他们名字听起来那么神奇

两个关键问题：

慢 & 断

- **各个层面的保证：**
  - Shard => Replication 避免单点
  - Node =>
    - 热点分散到不同的Node上 （需要时关闭自动分配）
    - 尽量减少跨节点汇集，避免被故障节点牵连。
    - 独立的查询节点供外部调用 （node.data: false）
  - Index =>
    - Circuit Breakers & Snapshot
    - 多Index互备，空间换时间
  - Cluster => 多集群互备 & 快速Rebuild的能力
- **Fail Fast & 上层容错**
- **足够的内存 + JVM调优**

如何让工程师们能快速的用好ES?

降低学习成本是第一步

也是最重要的一步

写Query好难

谁能不翻手册写出对应的ES查询语句?

```
select * from users
where
city = ‘shanghai’ and age >= 18
limit 100, 20
```

理解ES的优缺点已然很难，
别让所有人重学一门DSL！

现实

- 大量ES SDK还只支持用JSON拼装查询。
- Query DSL学习门槛较高。
- 查询代码可读性差。
- 很难统一的管理和优化。

充分借鉴DB的使用经验

回想下

现在还有什么代码框架
是裸写SQL的么？

怎能少了ORM和Query Builder!

扩展原有的Query Builder，
适配ES，减少学习成本。

至此，离成功还有一步！

“你还需要(至少)一个靠谱的人长期地负责维护和优化ES。”

– 朴素的真理

如果讲到这儿还有时间

O(∩_∩)O

Routing策略

- 此问题太复杂，线下私聊吧：)

确保数据一致性

- DB Cache ES 三者的一致性
  - 在ES中只Store ID
    - 和Data的CRUD逻辑绑定，调用者无需关心
    - 后台脚本不断的检查和补漏
- 【小心】_routing字段的值变了会导致ID不唯一 @issue 3346

定期的Rebuild策略

- 原因：词典变动 & 动态字段的逻辑变动
- N天后就自动好了对于大部分需求是OK的。
- 减少运维的手动Rebuild压力。

Q&A

加好友请自报家门
:)

邮箱： icedfish@gmail.com

微博/简书： @小排虞冰