



高吞吐状况下斗鱼搜索引擎优化之路

白凡@尚德机构

2017-11-04, weibo:阿bia123

Nov 4, 2017

关于我

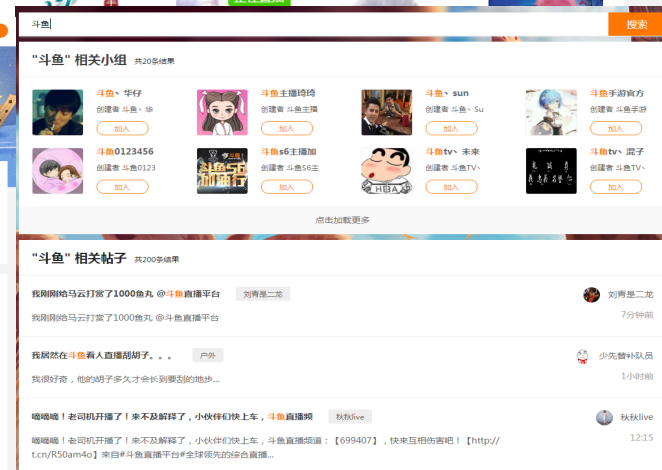
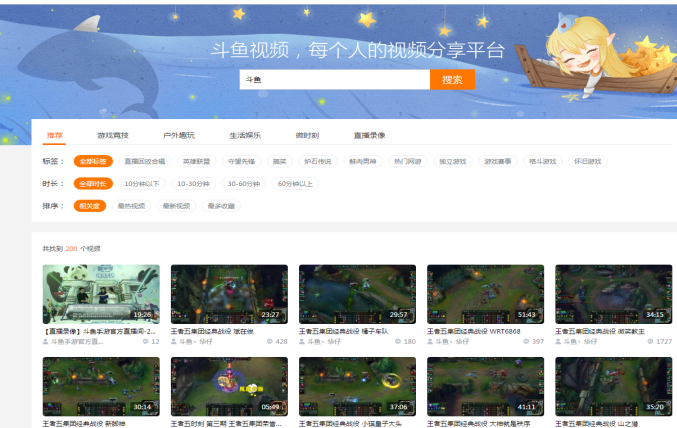
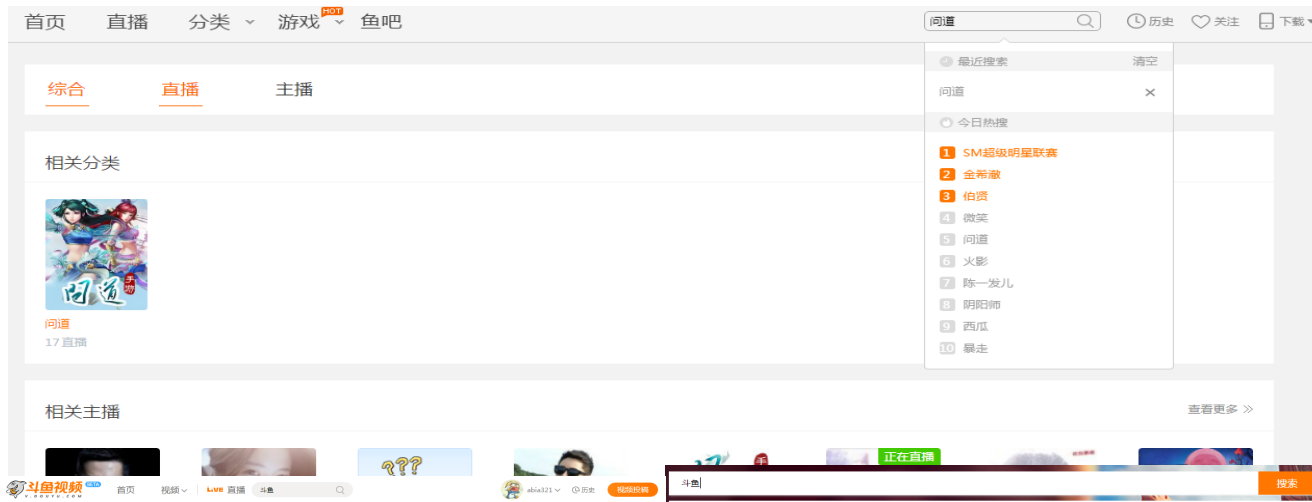
2015年1月进入斗鱼，一直从事斗鱼的搜索服务搭建。

2017年10月加入尚德机构。

主要项目包含，斗鱼主站，移动端、鱼秀、鱼吧、点播站等搜索服务。



Elasticsearch几乎覆盖整个斗鱼搜索服务



直播 搜索房间 / 主播 / 分类

取消

最近搜索

清空

微笑

陈一发儿

1209837

1414409

阴阳师

小灰灰

女流

火影

飞飞

饥荒

471330

今日热搜

1 SM超级明星联赛

2 金希澈

3 伯贤

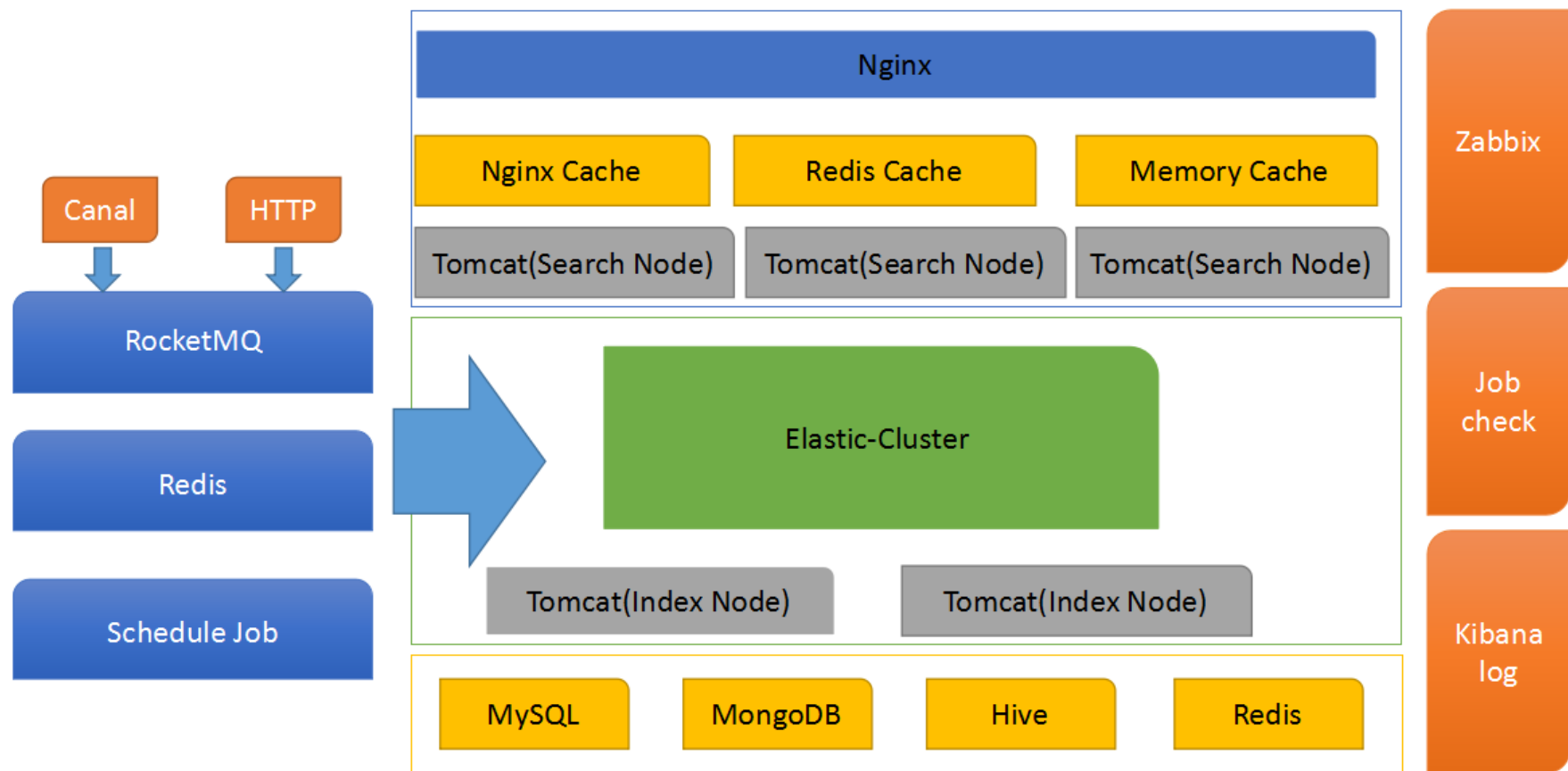
4 微笑

5 问道

6 火影

主要内容

- 技术架构
- 优化之路
- **ES**应用拓展
- 常见问题



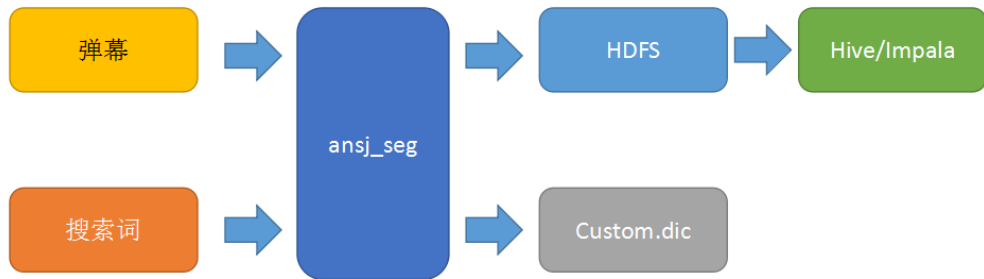
如何丰富自定义词库？

主要来源于斗鱼“弹幕”以及搜索词

采用Ansj中文分词器

HDFS，离线统计分析

Custom.dic，供索引分词



自定义分词器

场景:

斗鱼直播间有房间描述，如“五五开”的直播间，其描述可能就会有“五五开，电竞卢本伟，大神”等。且对于描述的权重比例很高。

而对于搜索业务，我们不希望搜索“五”，“开”，“卢本伟”等词能匹配到，仅在搜索“五五开”，“电竞卢本伟”时返回结果。

因此，`whitespace analyzer`加入基于“，”的 `pattern tokenizer`。实现空格和逗号分词。

创建mapping时，`analyzer`采用 `whitespace,search_analyzer`仍然使用ik，搜索使用`termQuery`

```
1
{
  "settings": {
    "analysis": {
      "analyzer": {
        "pattern_analyzer": {
          "tokenizer": "dy_pattern"
        },
        "pinyin_analyzer": {
          "tokenizer": "dy_pinyin"
        },
        "ik_smart": {
          "tokenizer": "dy_ik"
        },
        "doudou_analyzer": {
          "tokenizer": "dy_ngram"
        }
      },
      "tokenizer": {
        "dy_pinyin": {
          "type": "pinyin",
          "keep_full_pinyin": false
        },
        "dy_pattern": {
          "type": "pattern",
          "pattern": ","
        },
        "dy_ik": {
          "type": "lowercase"
        },
        "dy_ngram": {
          "type": "ngram",
          "min_gram": 1,
          "max_gram": 20,
          "token_chars": [
            "letter",
            "digit",
            "punctuation",
            "symbol"
          ]
        }
      }
    }
  }
}
```

Java API开发

BoolQuery多维度查询

维度权重比例不一，权重值配置中心统一调度

```
List<WebRoomBase> roomResult = new ArrayList<WebRoomBase>(),
//
QueryBuilder = QueryBuilders.termQuery(
    " ", sk).boost(
        (float) OceanDataCommonConfig
            .getValueInteger(" "));
//
QueryBuilder queryRoomNameFuzzy = QueryBuilders.prefixQuery(
    " ", sk);
//
QueryBuilder = QueryBuilders.matchQuery(" ", sk);
//
QueryBuilder = QueryBuilders.termQuery(" ", sk)
    .boost((float) OceanDataCommonConfig
        .getValueInteger(" "));
//
QueryBuilder = QueryBuilders.termQuery(" ", sk)
    .boost((float) OceanDataCommonConfig
        .getValueInteger(" "));
BoolQueryBuilder shouldQuery = QueryBuilders.boolQuery()
    .should( )
    .should( )
    .should( )
    .should( )
    .should( );
//
QueryBuilder queryStatus = QueryBuilders.termQuery(" ", 1);
BoolQueryBuilder mustQuery = QueryBuilders.boolQuery()
    .must(shouldQuery).must(queryStatus);
```


增量索引

- 由于斗鱼语言的异构性，**PHP,C++,Java...**，所以索引同步显得非常麻烦，在这里，分享目前斗鱼搜索的几个增量索引的方法。
- **RocketMQ**，实时性、稳定性最好，可复用，需要中间插件，比较推荐。数据源为**Canal**解析binlog，亦或前端直接通过**HTTP**请求**POST**增量数据。只要设置不同的**Group**，可共同消费。
- **Redis**队列，实时性、稳定性好，不可复用，需要中间插件。采用**Redis**队列进行消费，但不可复用，只能一个服务端消费，且如果未做持久化、一旦**Redis**挂掉，数据不可恢复。
- 定时任务，无需中间件，但实时性、稳定性不好。通过更新时间定时更新增量索引，一旦任务挂掉，时间点一过，则不可恢复。且对更新时间非常敏感，必须为写入数据库时间。

优化之路

- 集群升级，原因不外乎其二：
 1. 现有版本性能已经不能满足当前业务需求
 2. 新版本的某些接口或者功能很诱人
- 斗鱼升级主要是，业务扩展开，全站搜索ES化，PV上涨过快，搜索接口压力加大。
- 除了加机器，换SSD，水平加data node，配合ELK体系一起升级5.x
- String->text, keyword IK->ik_smart, ik_max_word
- head cerebro（前身kopf）成为独立server

优化之路

- 分词系统升级，5.x之后，分词配置不在`elasticsearch.yml`。
- 最主要用`ngram`分词替换掉了令人头疼的`wildcard`查询。
- 弊端在于会产生大量分词文件。
- 控制分片数量，太多并不意味着会更好，根据实际的性能要求调整。
- `mapping`一旦创建无法更改，建议创建一个`mapping`时，多预留备份字段。

优化之路

- 索引均匀写入磁盘，分摊磁盘I/O压力

```
#####
#
# ----- Paths -----
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /mnt/sdb, -data1,/mnt/sdc, -data1,/mnt/sdd, -data1,/mnt/sde, -data1,/mnt/sdf, -data1
#
# Path to log files:
#
#path.logs: /path/to/logs
"
```

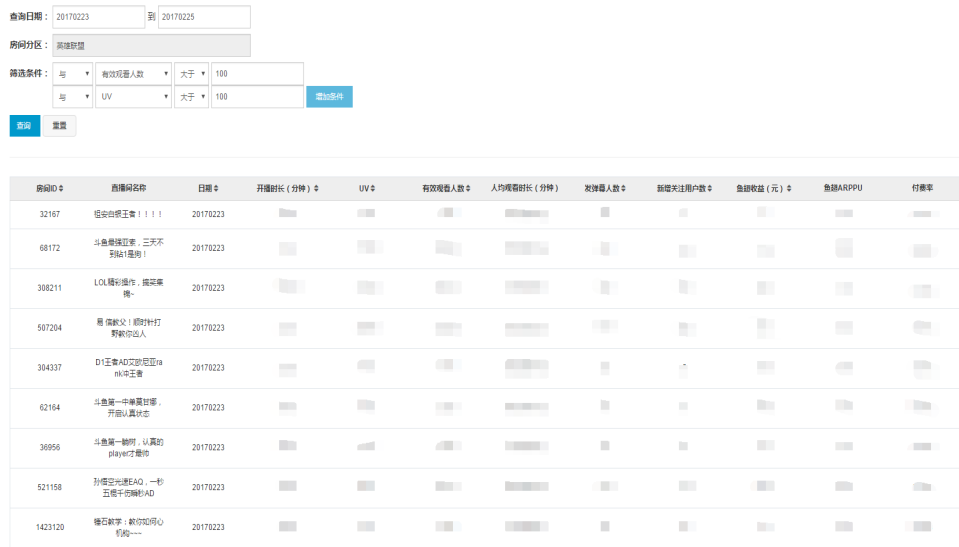
优化之路

- 记录所有慢查询，同样无法再ES.yml里配置

```
curl -XPUT 'http://[redacted]:9200/_all/_settings?preserve_existing=true' -d '{
  "index.indexing.slowlog.level" : "info",
  "index.indexing.slowlog.threshold.index.debug" : "s",
  "index.indexing.slowlog.threshold.index.info" : "s",
  "index.indexing.slowlog.threshold.index.trace" : "s",
  "index.indexing.slowlog.threshold.index.warn" : "s",
  "index.search.slowlog.threshold.fetch.debug" : "s",
  "index.search.slowlog.threshold.fetch.info" : "s",
  "index.search.slowlog.threshold.fetch.trace" : "s",
  "index.search.slowlog.threshold.fetch.warn" : "s",
  "index.search.slowlog.threshold.query.debug" : "s",
  "index.search.slowlog.threshold.query.info" : "s",
  "index.search.slowlog.threshold.query.trace" : "s",
  "index.search.slowlog.threshold.query.warn" : "s"
}'
```

ES应用拓展

- 目前主流ELK，Elasticsearch。
- Elasticsearch用于数据可视化展示，可以轻松实现多维度逻辑查询。
- Elasticsearch用于类似天梯匹配（推荐）系统，可依靠boost轻松调节业务逻辑。



常见问题

- 数据库和ES 数据同步的方案?
- 集群设置多少分片，设置多少数据节点，**master** 节点问题?
- **ES**在高并发环境下的可用性?

Questions?!

Thanks!