

推特搜索系统

Twitter Search System

主讲人 南帝老师

Keyword Search 雪花算法 kafka
冷热数据分离 Indexer
Earlybird 消息队列 HDFS
副本机制 Blender Twitter Index Service
ES-Hadoop 搜索前端服务器 Snowflake
读写分离 Lucene
Elasticsearch

Scenario 场景

设计一个推特搜索系统 Twitter Search System

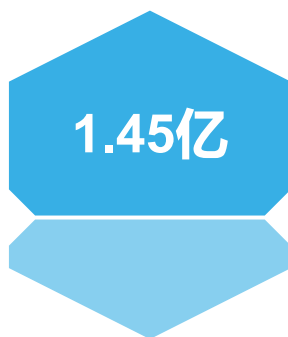
相关场景介绍

数据估算

月活跃用户(MAU)



日活跃用户(DAU)



每天新发布tweet



每天推送次数



每天处理搜索



如果一个人每10秒看一条推特，则需要158年才能看完一天内上传的所有推特。

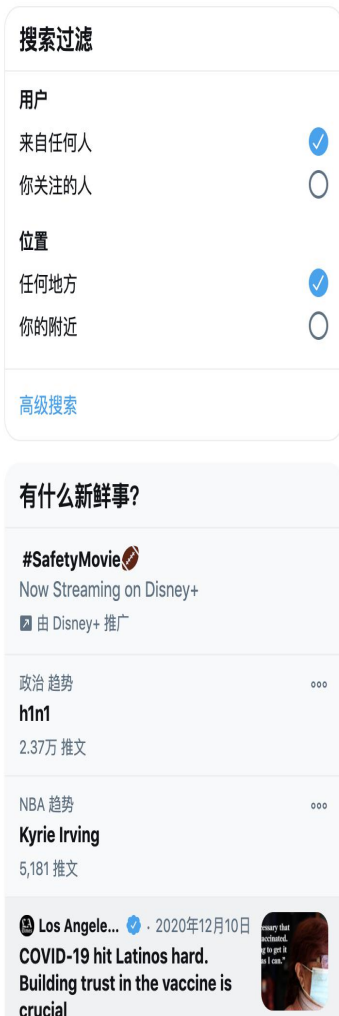
平均每秒的搜索 17K 次

数据来源：

• https://en.wikipedia.org/wiki/List_of_social_platforms_with_at_least_100_million_active_users

实时数据：

• <https://www.internetlivestats.com/twitter-statistics/>



关键字搜索

Keyword Search

相关性搜索

Relevance

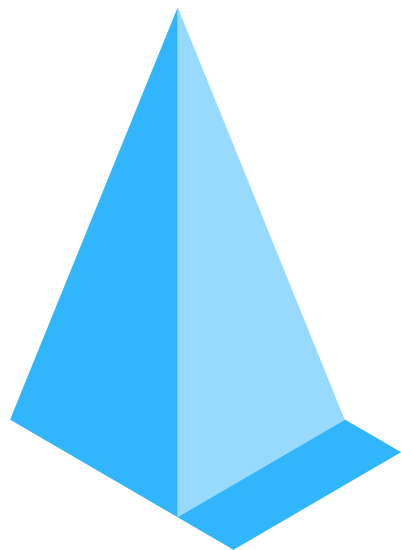
Search

分面搜索

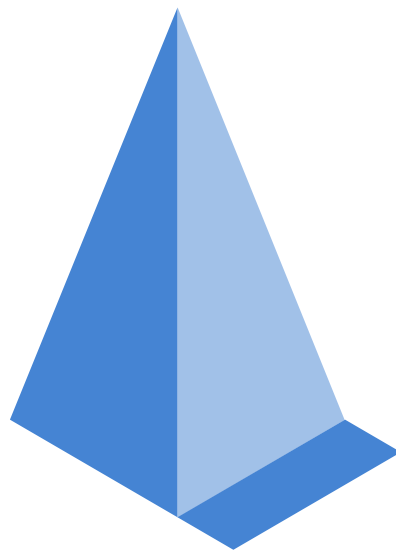
Facets Search

Server 服务

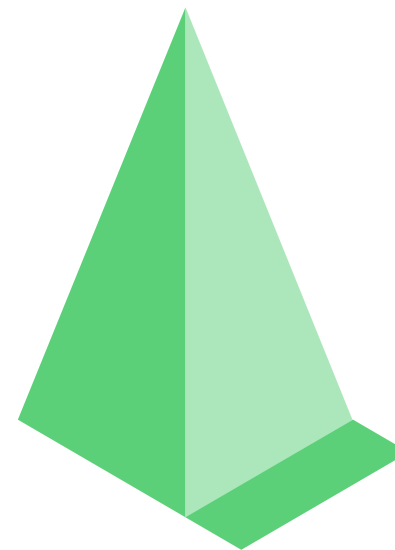
推特搜索服务架构



索引服务
Index Service



搜索服务
Search Service



排名服务
Ranking Service

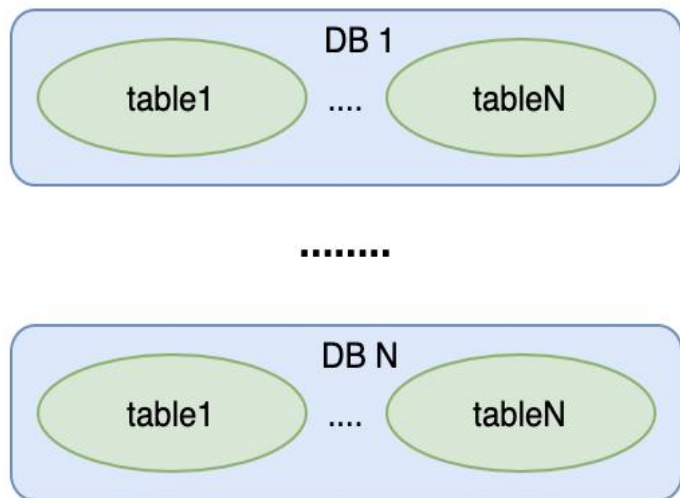
Storage 存储

了解整体存储架构

了解搜索引擎



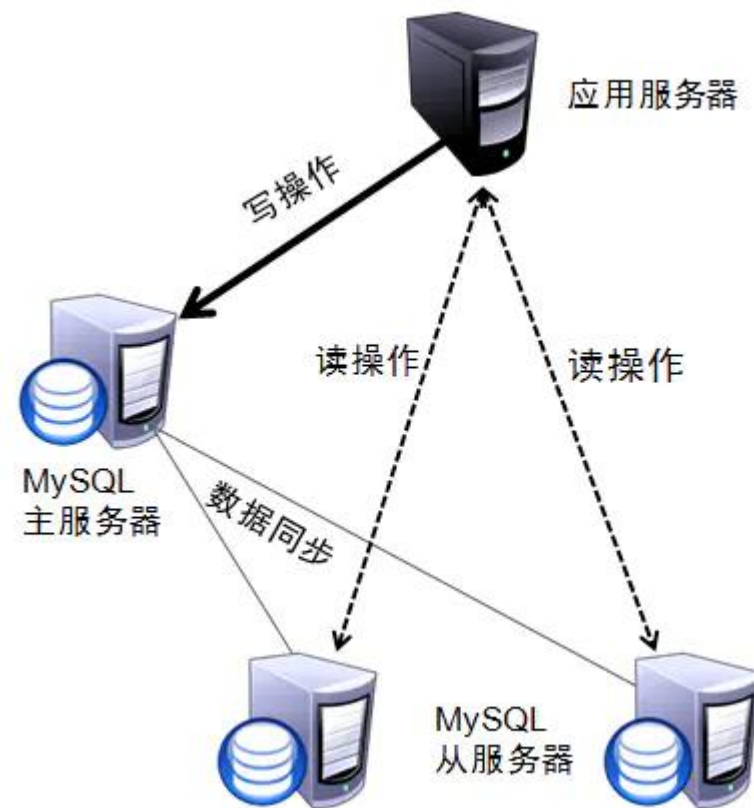
分库分表



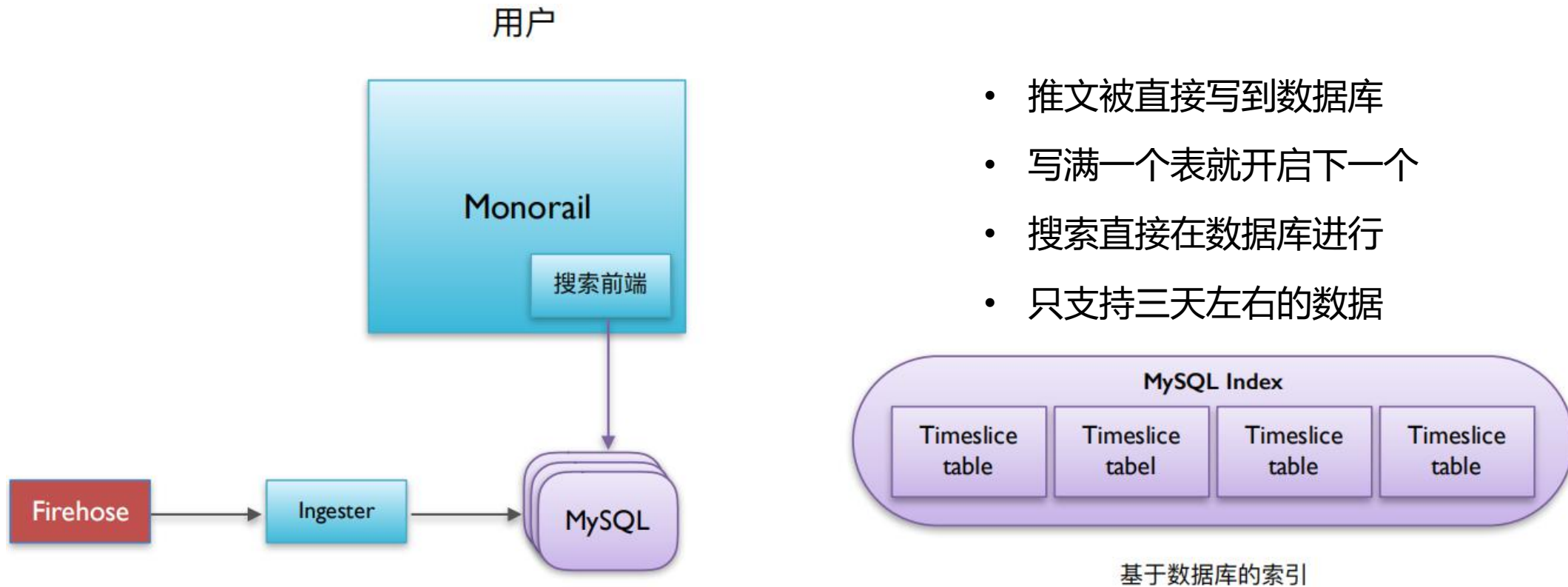
冷热数据分离



读写分离



最开始的时候 Twitter 并没有开发自己的搜索引擎，而使用 Mysql 数据库自带的搜索



在 MySQL 数据库里，twitter 基于时间做了 index，分成了多个 Timeslice table。

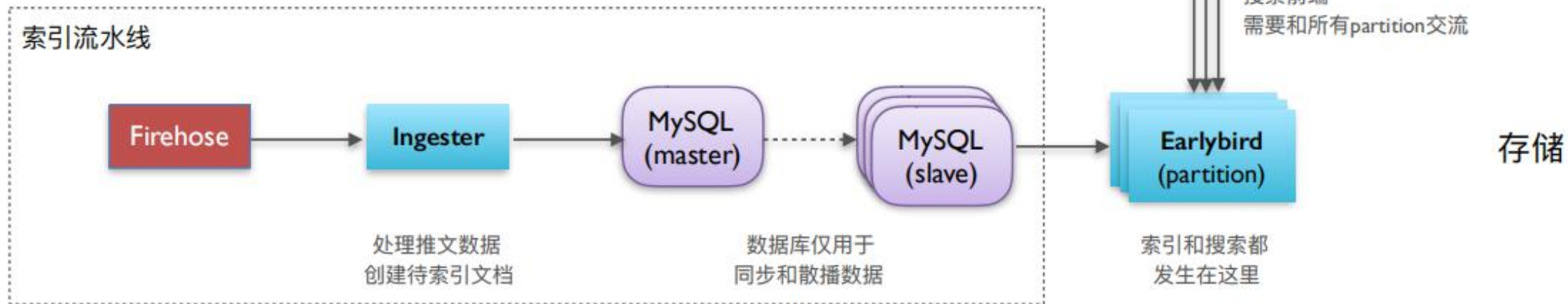


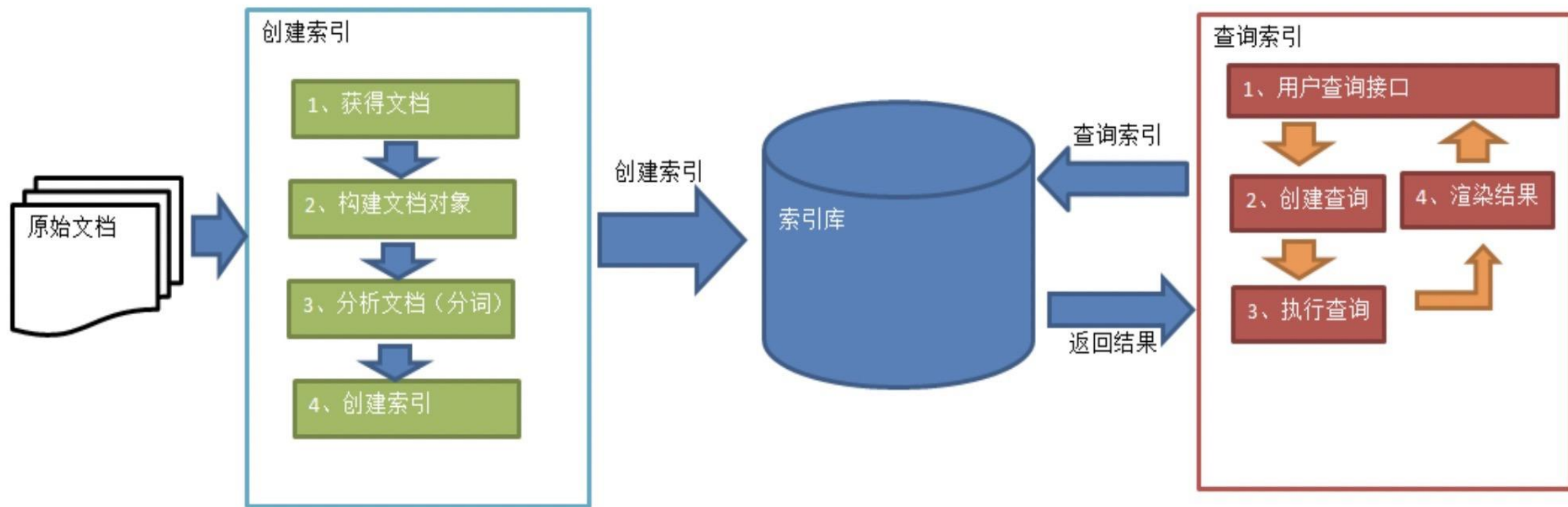
大家可以思考一下，这样做会有什么弊端？

- 由于 tweets 的数量过于庞大，每次搜索只支持三天所有的数据的查找。
- 使用MySQL搜索能支持的查找方式非常单一，并且很难扩展
- 单一的关键词进行搜索，不支持对多个关键词排列组合进行搜索，filter 过滤，模糊查询，范围搜索
- 当数据量过多的时数据压力比较大，查询速度非常慢

Twitter 团队使用 Lucene 这个开源库开发了 Earlybird 这个新一代索引服务器

- Earlybird既是一个索引器(Indexer)也是一个索引服务器(Index Server)
- MySQL现在仅用于同步和散播数据, 保存序列化的文档 (推文)
- 搜索前端代码需要和所有Earlybird partition通信





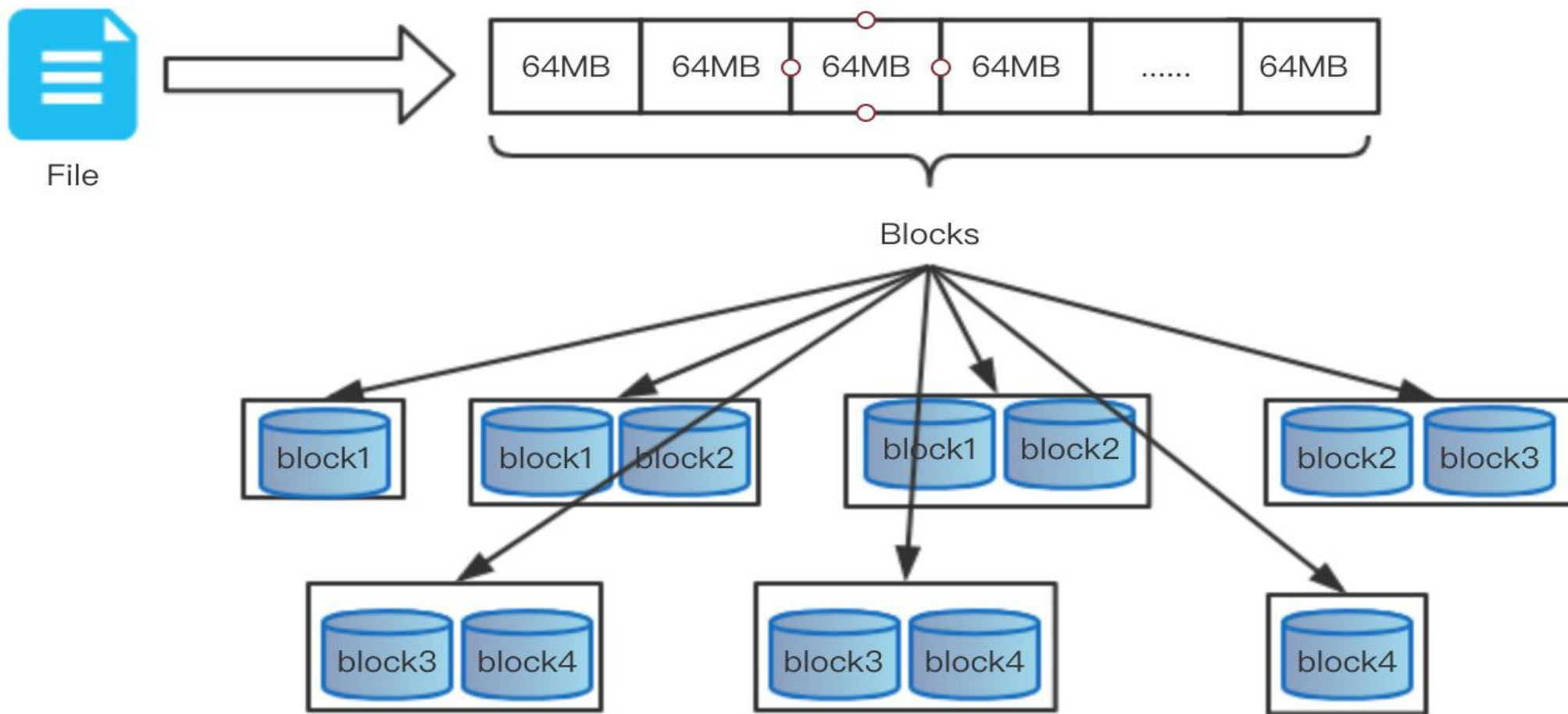


索引文件应该存在哪里？

- A. 关系型数据库 (MySQL)
- B. 非关系型数据库 (Redis)
- C. 文件系统 (HDFS)



提前对查询的内容建立索引，放到索引库中，查询的时候再从索引库中查询



生活中的索引

目 录

推荐序
前言

第 1 章 MySQL 体系结构和存储引擎 1

1.1 定义数据库和实例 1

1.2 MySQL 体系结构 3

1.3 MySQL 存储引擎 5

1.3.1 InnoDB 存储引擎 6

1.3.2 MyISAM 存储引擎 7

1.3.3 NDB 存储引擎 7

1.3.4 Memory 存储引擎 8

1.3.5 Archive 存储引擎 9

1.3.6 Federated 存储引擎 9

1.3.7 Maria 存储引擎 9

1.3.8 其他存储引擎 9

1.4 各存储引擎之间的比较 10

1.5 连接 MySQL 13

1.5.1 TCP/IP 13

1.5.2 命名管道和共享内存 15

1.5.3 UNIX 域套接字 15

1.6 小结 15

第 2 章 InnoDB 存储引擎 17

2.1 InnoDB 存储引擎概述 17

2.2 InnoDB 存储引擎的版本 18

2.3 InnoDB 体系架构 19

2.3.1 后台线程 19

2.3.2 内存 22

2.4 Checkpoint 技术 32

2.5 Master Thread 工作方式 36

2.5.1 InnoDB 1.0.x 版本之前的 Master Thread 36

2.5.2 InnoDB 1.2.x 版本之前的 Master Thread 41

2.5.3 InnoDB 1.2.x 版本的 Master Thread 45

2.6 InnoDB 关键特性 45

2.6.1 插入缓冲 46

2.6.2 两次写 53

2.6.3 自适应哈希索引 55

2.6.4 异步 IO 57

2.6.5 刷新邻接页 58

2.7 启动、关闭与恢复 58

2.8 小结 61

第 3 章 文件 62

3.1 参数文件 62

3.1.1 什么是参数 63

3.1.2 参数类型 64

3.2 日志文件 65

3.2.1 错误日志 66

3.2.2 慢查询日志 67

3.2.3 查询日志 72

3.2.4 二进制日志 73

3.3 套接字文件 83

3.4 pid 文件 83

3.5 表结构定义文件 84

3.6 InnoDB 存储引擎文件 84

3.6.1 表空间文件 85

3.6.2 重做日志文件 86

3.7 小结 90

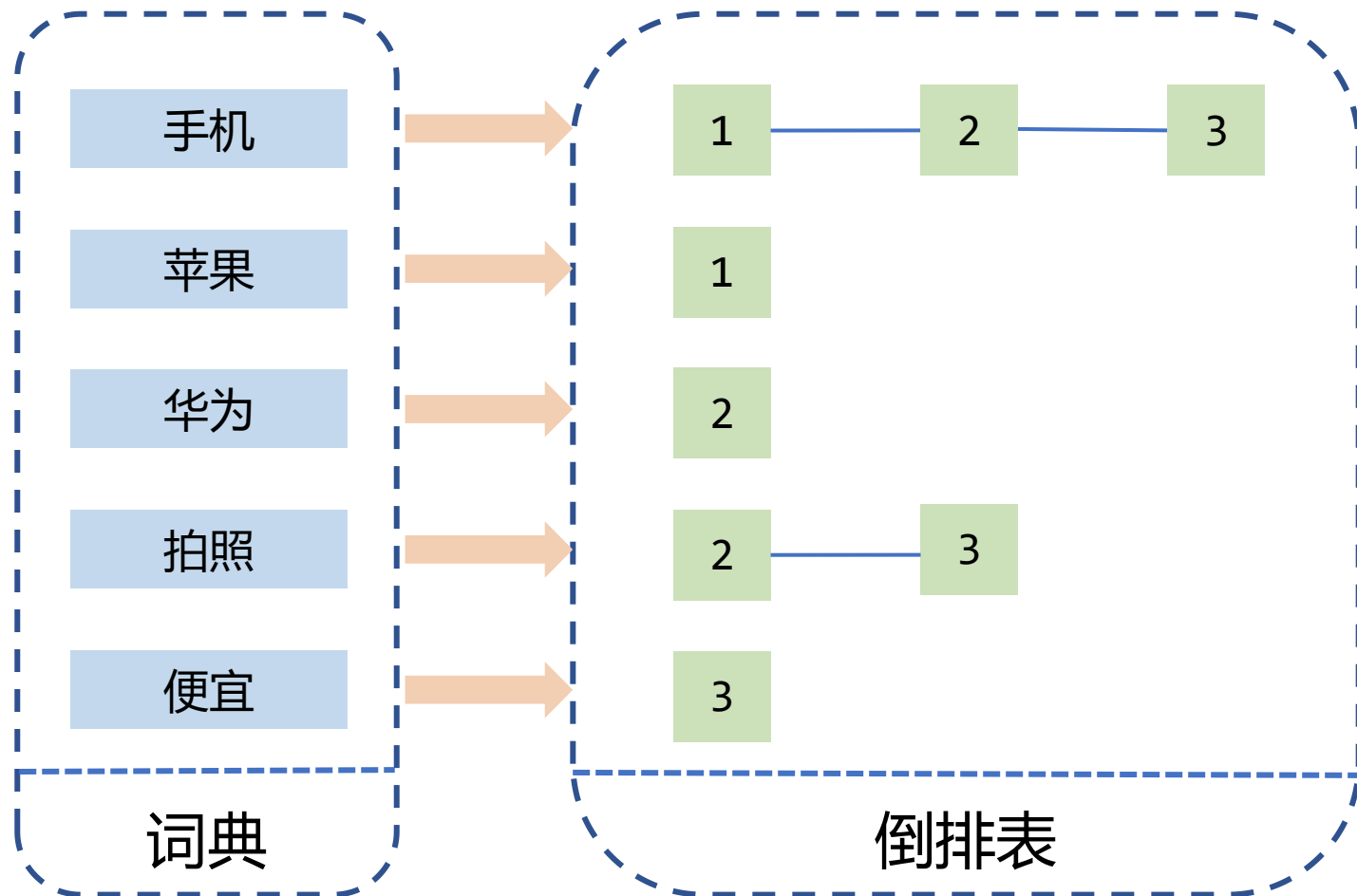
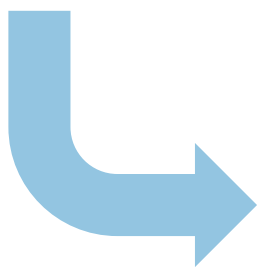
第 4 章 表 91

4.1 索引组织表 91

目录的特点

- 01 单独的结构
- 02 指向了具体内容的位置
- 03 加速了对书籍内容的查找
- 04 有些内容无法放到书籍里
- 05 书变厚了。。。

推特ID	推特内容
1	苹果手机系统流畅
2	华为手机拍照很好
3	小米手机拍照很好且便宜



倒排索引结构是根据内容（词语）找文档，倒排索引结构也叫反向索引结构，包括索引和文档两部分，索引即词汇表，它是在索引中匹配搜索关键字，由于索引内容有限并且采用固定优化算法搜索速度很快，找到了索引中的词汇，词汇与文档关联，从而最终找到了文档。

01

提取资源中关键信息， 建立索引，
对文档进行切分词组成索引

02

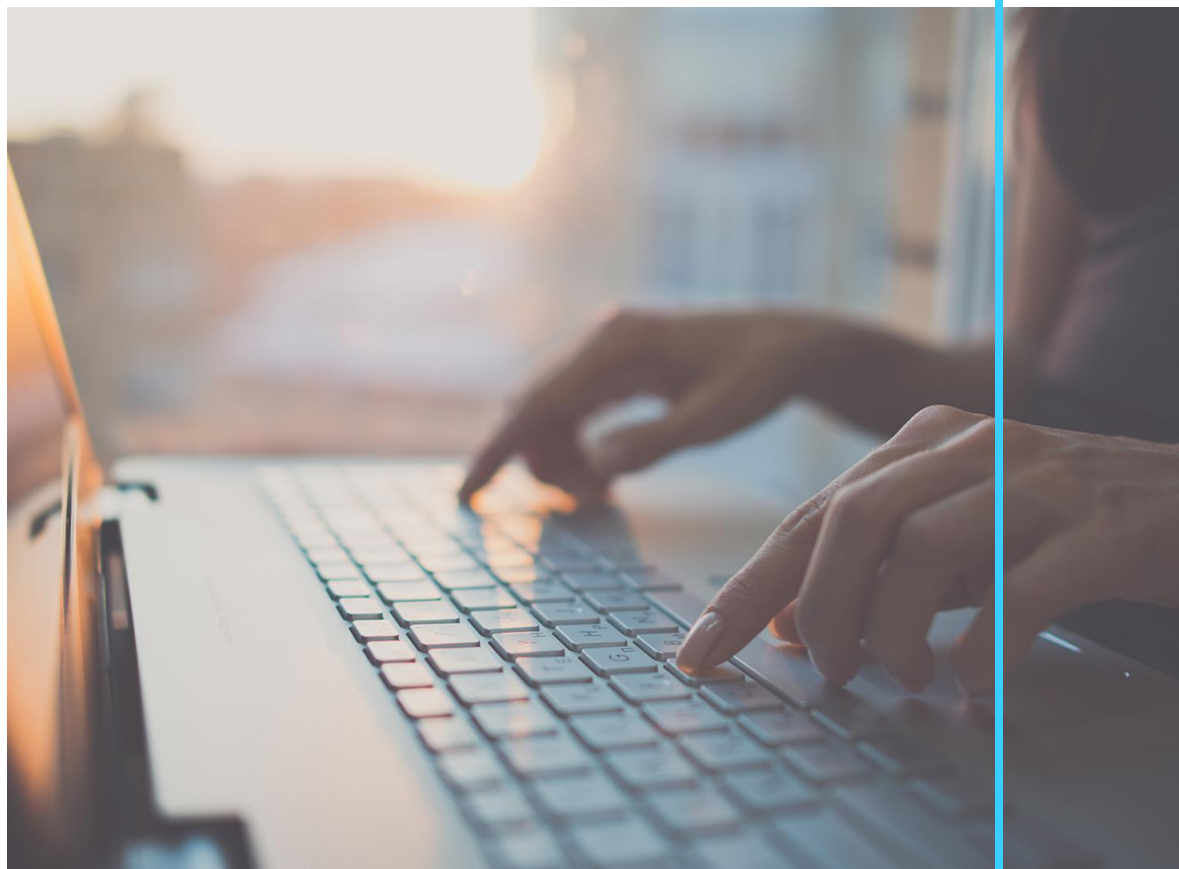
搜索时，根据关键字（目录），找
到资源的位置



- 切分词：就是将一段文本，根据一定的规则，拆分成一个一个词
将一句一句话切分成一个一个词，去掉 (a an the and or)等无意义的词, 空格，标点符号，
重复的词大写字母转为小写字母

对搜索结果排名

- 一. Twitter搜索排序策略
- 二. 存储设计



搜索关键字，自己新发布的推特搜索不出来，而排在前面的总是哪些热度很高的推特。



推文热度值

- 点赞数、收藏数、评论数、阅读数
- 热度值 = 点赞数 + 收藏数 + 评论数 + 阅读数



社交关系

- 是否是你关注
- 是否是大V



时效性

- 人气数 = 热度 + 社交关系 + 时效性
- 在存储时候将计算出来的“人气数”和索引一起存储，在搜索时根据人气数对搜索结果排序

Tweet Table

字段	内容	类型
id	id	varchar
user_id	用户	varchar
content	推特内容 (tweets)	text
create_time	创建时间	timestamp
likes	点赞次数	long
forwarding_times	转发次数	long
comment	评论次数	long

User Table

字段	内容	类型
user_id	id	varchar
user_name	用户名	varchar
email	邮箱	varchar
is_superstar	是否是明星	boolean



有问题找她(*^_^*)

《电商秒杀系统 -
Spring项目实战》

优惠券: **6B7F7E**

2周快速实现项目

提供项目源代码, 保姆教程
手把手

简历模板, 直接写入简历

30天反复回看学习

《算法面试高频题
冲刺班》

优惠券: **6B7F7E**

中小厂必考面试知识点

大厂常规面试知识点

FLAG等大厂高频面试知识点

面试高频题

老学员私教: **95折**
简历代笔

优惠券: **DD1A15**

大厂老师帮你改简历

实战面试, 算法/系统设计
均可

面试必备行为指导

项目深挖/包裹谈判

**其他优惠券欢迎私
信小娃娃(●'◡'●)**

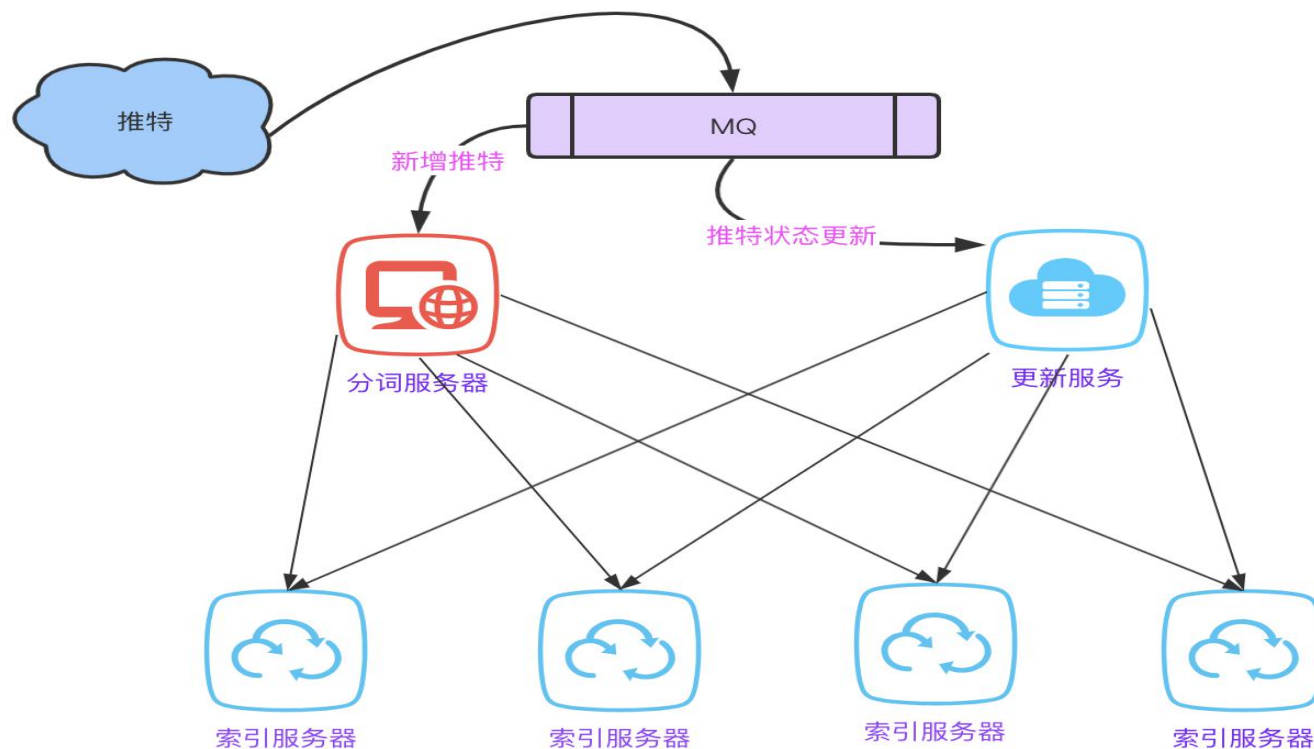


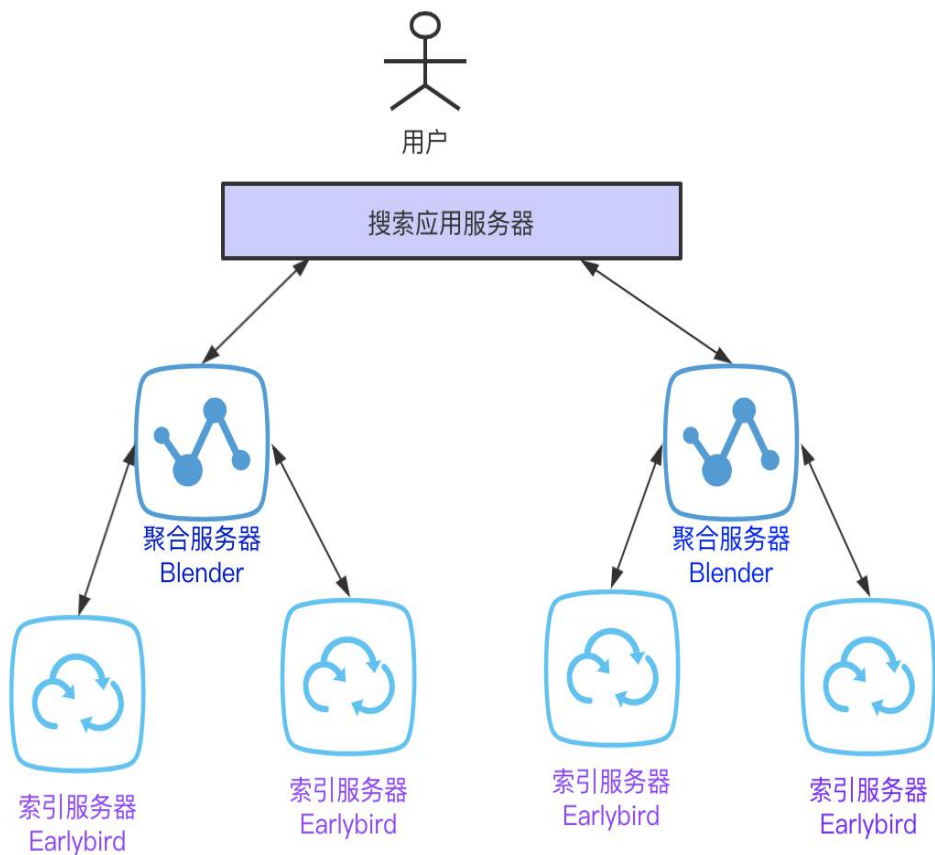
怎么加速搜索的速度，还有什么优化的地方吗

建立实时索引

将最近1周内的最新的推文建立索引库存储在内存当中
从内存中查询索引比从磁盘快很多

1. 用户发布的新 tweets 会被发送到分词服务器里面。在这里tweets 的文本被分词
2. 按照 hash 分割, tweets 被分发到各个 Earlybird(索引服务器)上, 每个 Earlybird(索引服务器)负责一部分数据, 将 tweets 实时地建立索引
3. 同时, 另外有个一个更新服务, 它推送 tweets 的动态变化信息 (例: 点赞次数, 转发次数), 动态地更新索引。





01

搜索请求

用户搜索请求首先到达 Blender (搜索前端服务器), Blender 解析请求

02

执行计算

Earlybird 服务执行相关性计算并排序。并将排序好的 tweet 列表返回给 Blender。

03

返回结果

Blender 合并各个 Earlybird 返回的列表, 并执行一些重排序 (Reranking), 然后返回给用户。

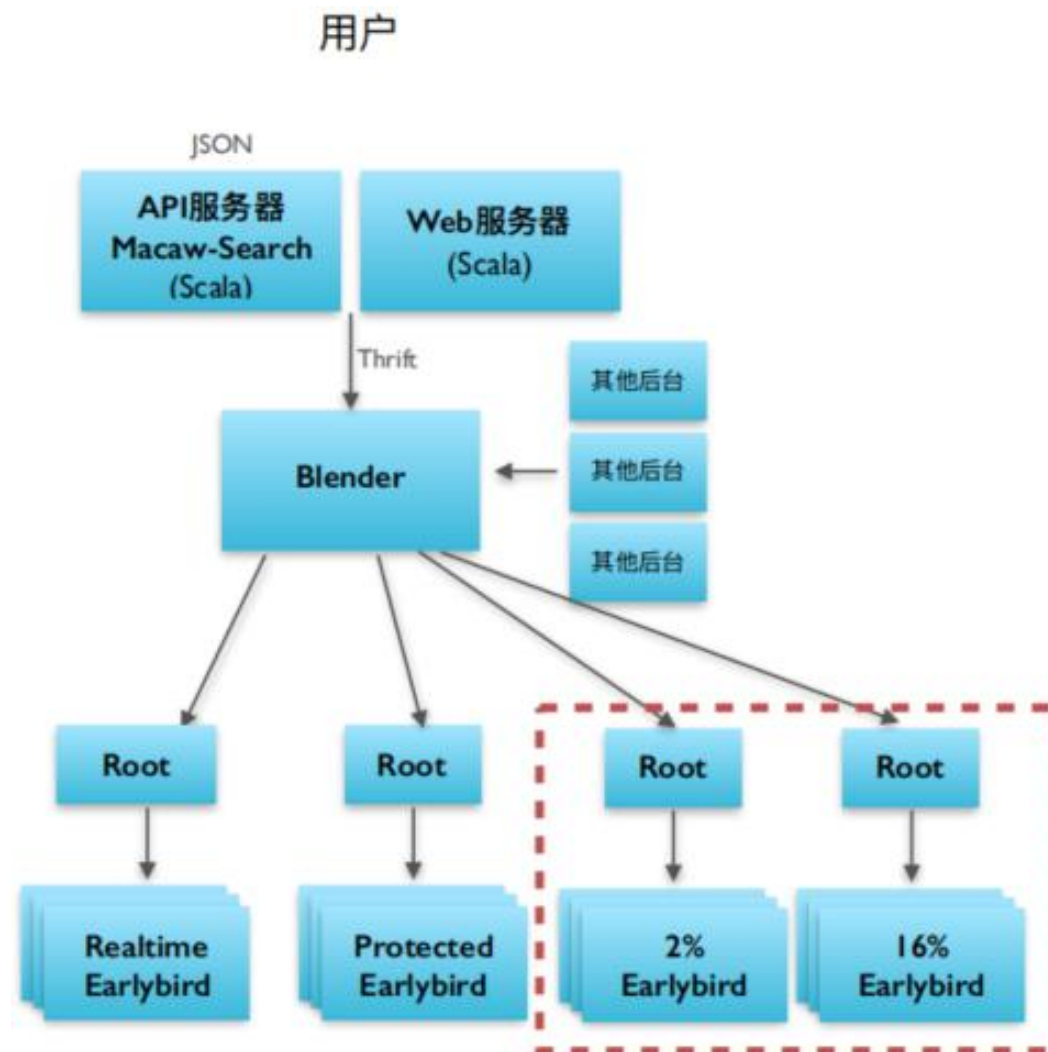


还能继续优化?



Twitter 用了一个非常巧妙的方法就是保存2%热度最高最可能被检索的 tweets 在内存中，并且保存了16%的 tweets 在 SSD 硬盘。

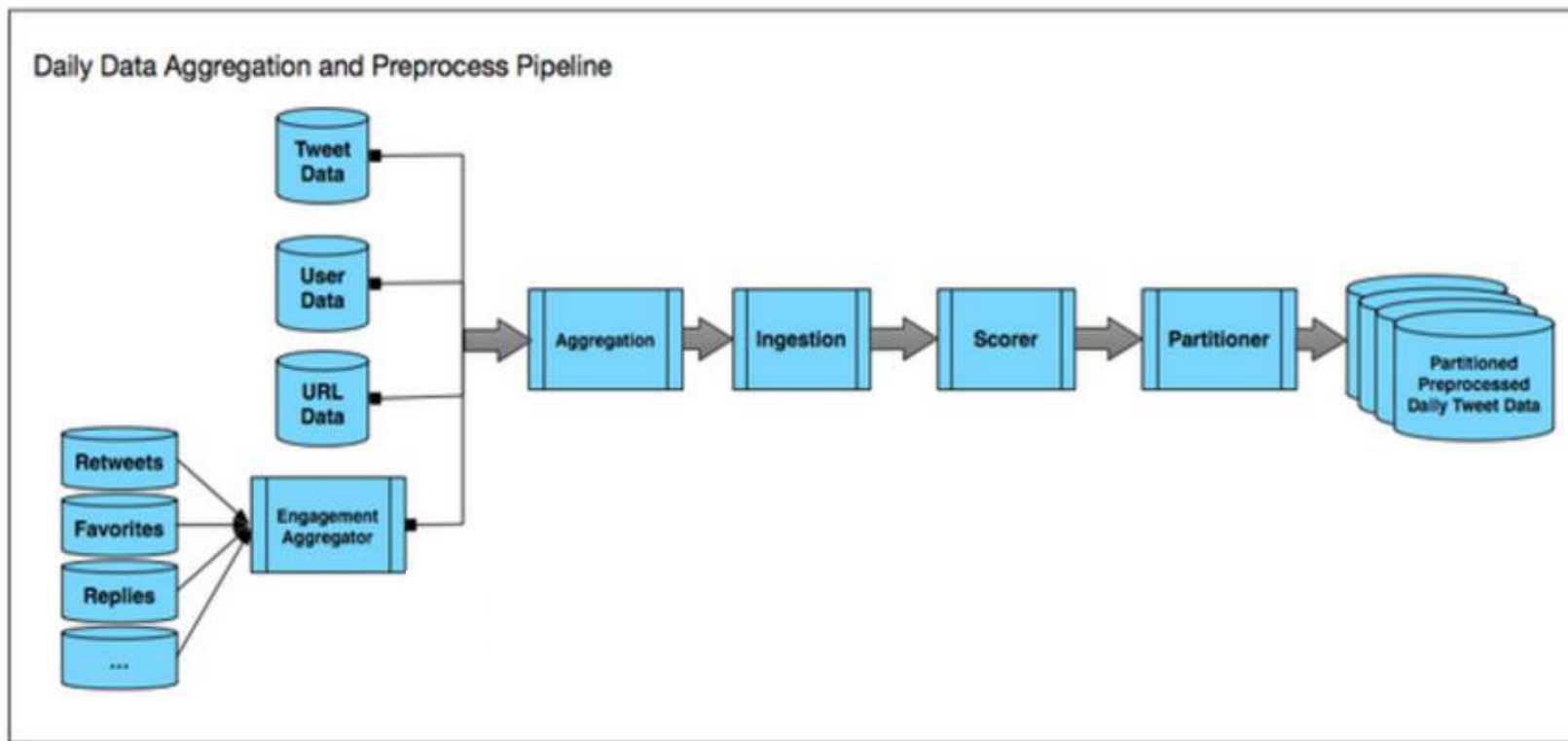




聚合：基于Tweet ID一起加入多个数据源

打分：根据推特的特征提取（转发次数，点赞次数，评论次数等）来打分,分数越高

分区存储：将数据划分为小块存储在HDFS



唯一的 tweet_id 如何生成?

传统方式

1. 时间戳 + N 位随机数
这种方法在分布式系统内会产生 ID 碰撞。

2. UUID

入数据库性能差，因为UUID是无序的。



雪花算法 (Snowflake)

- 雪花算法是 Twitter 的分布式自增 ID 算法，经测试 SnowFlake 每秒可以产生 26 万个自增可排序的 ID。
- Twitter 的 SnowFlake 生成 ID 能够按照时间有序生成。
- 在分布式系统内不会产生 ID 碰撞（由 DataCenter 和 WorkerID 做区分）并且效率较高。

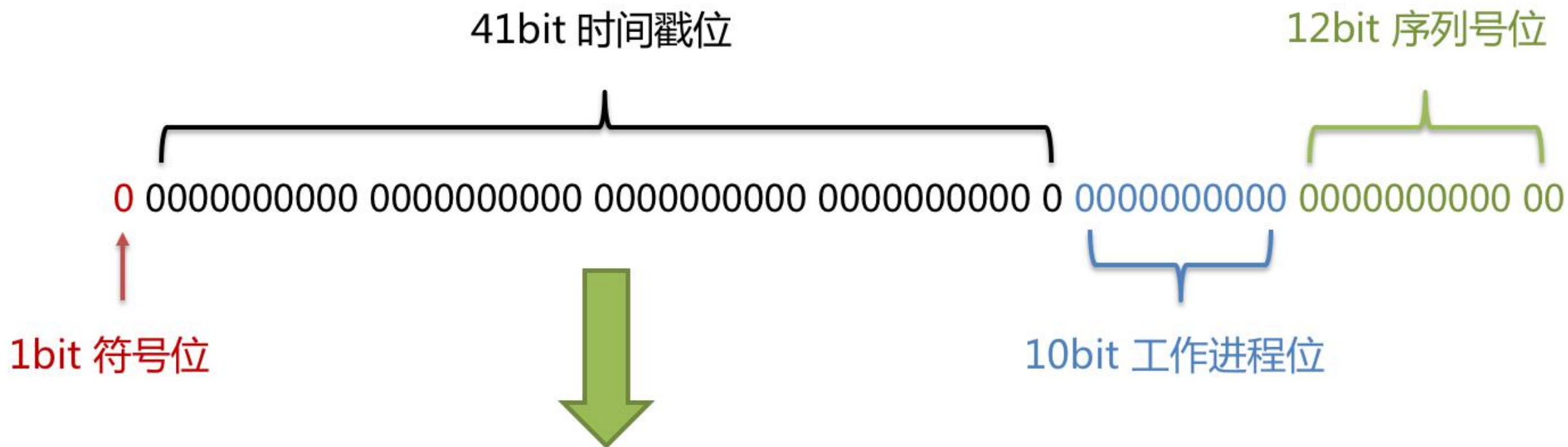
有兴趣的同学可以参考现有的开源实现

1.Scale 版详见开源项目 (Twitter 官方开源版本)

<https://github.com/twitter/snowflake>

2.Java 版中文描述

<https://github.com/souyunku/SnowFlake>



时间范围： $2^{41} / (365 * 24 * 60 * 60 * 1000L) = 69.73$ 年
工作进程数量： $2^{10} = 1024$
生成不碰撞序列的TPS： $2^{12} * 1000 = 409.6$ 万

电商网站的搜索和推特的搜索有什么区别？



电商网站的搜索和推特的搜索有什么区别？



手机

搜索

我的购物车 0

华为手机 | 手机自营 | 手机5g | 华为 | 苹果手机 | 小米 | 小米手机 | 小米10 | 苹果12 | 荣耀 | 苹果 | 华为p40 |

全部商品分类

京东时尚

美妆馆

超市

生鲜

京东国际

闪购

拍卖

金融

全部结果 > "手机"

品牌:

HUAWEI

Apple Authorized Reseller 授权经销商

小米

oppo

vivo

SAMSUNG

ONEPLUS

realme

黑鲨科技 BLACK SHARK

NOKIA

MEIZU

nubia

Lenovo

-TOUCH 天语

PHILIPS

SONY

REPUBLIC OF GAMERS

朵唯 (DOOV)

更多 多选

分类:

手机 二手手机

品牌名称:

21KE 360 8848 华为 (HUAWEI) 小米 (MI) Apple vivo OPPO 荣耀 (honor) 纽曼 (Newman) 三星 (SAMSUNG)

更多 多选

屏幕尺寸:

5.0英寸以下 5.0~5.49英寸 5.5~5.99英寸 6.0~6.24英寸 6.25-6.34英寸 6.35-6.44英寸 6.45-6.54英寸 6.55-6.64英寸 6.65-6.74英寸

更多 多选

高级选项:

运行内存 存储卡 分辨率 电池容量 后摄主摄像头 CPU型号 机身存储 摄像头数量 机身颜色 屏幕前摄组合 操作系统 热点 前摄主摄像头

2GB以下 2GB 3GB 4GB 6GB 8GB 12GB 16GB 其他

更多



支持各种维度的排序，包括支持人气、销量、信用、价格、发货地等属性的排序



支持范围查找，价格区间范围搜索



支持商品属性筛选，例如：品牌，具体参数



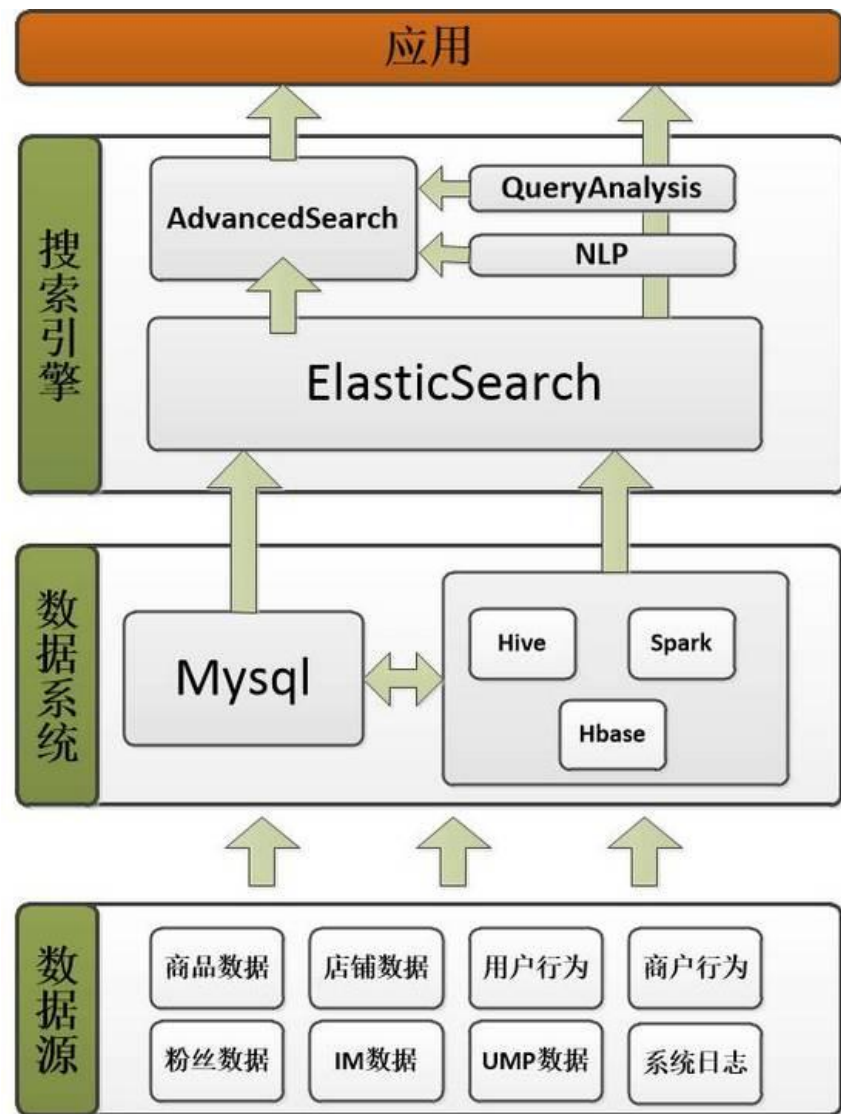
对数据的实时性要求非常高，体现在价格和库存两个方面

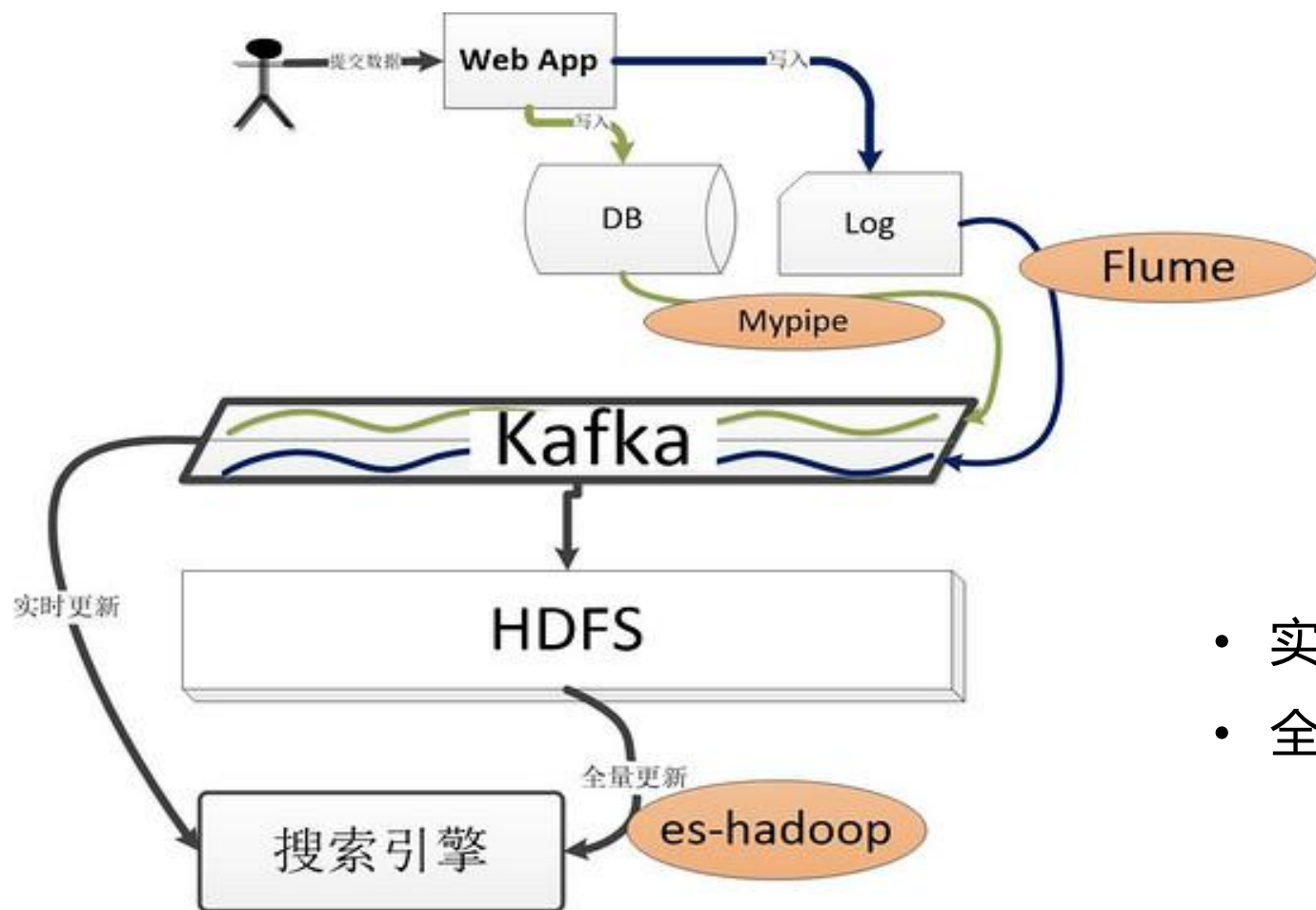


对数据准确性要求高，不能丢失商品

目前电商使用多的搜索引擎基于分布式实时引擎 **Elasticsearch(ES)**，ES 构建在开源社区最稳定成熟的索引库 **Lucence** 上。

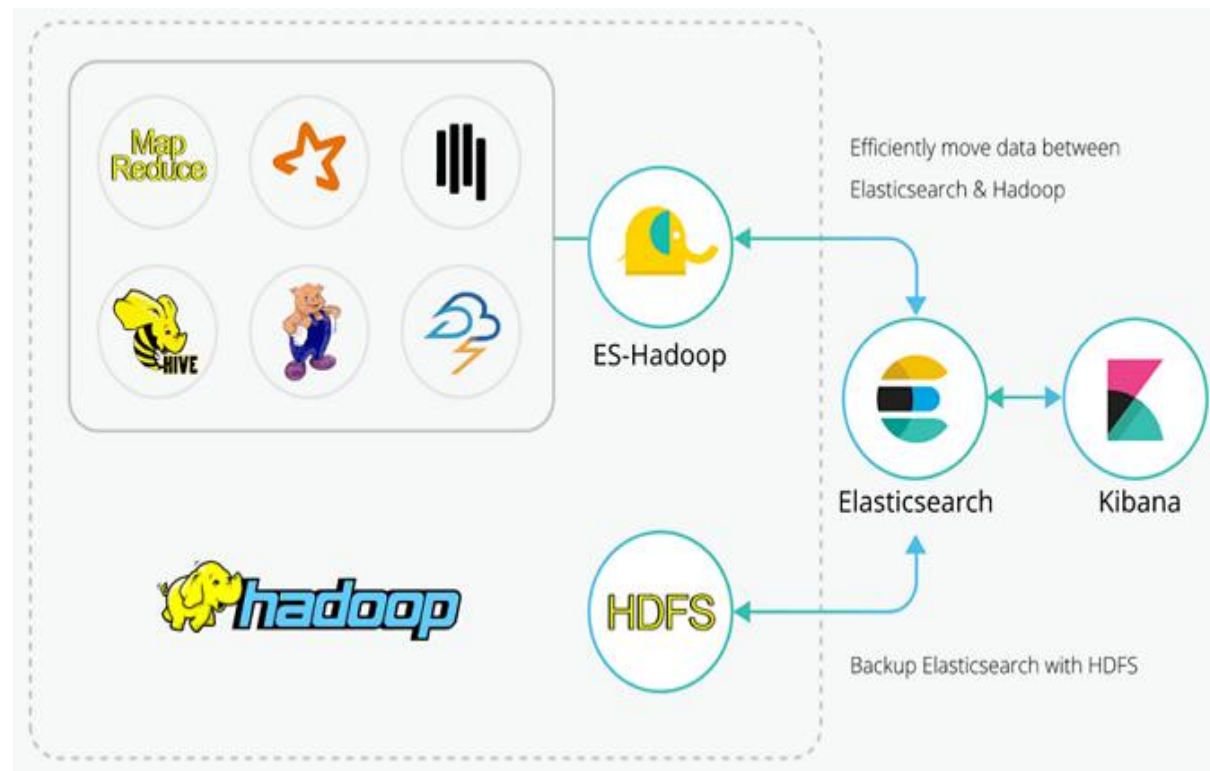
- 支持高可用, 可水平扩展
- 并有自动容错和自动伸缩的机制
- 支持还实现了 ES 与 MySQL 和 Hadoop 的无缝集成





- 实时增量的更新索引通过 **kafka** 实现
- 全量索引存储在 **HDFS** 中

ES-Hadoop(Elasticsearch for Apache Hadoop): 无缝打通了 ES 和 Hadoop 两个非常优秀的框架, 在 Hadoop 和 Elasticsearch 之间起到桥梁的作用, 完美地把 Hadoop 的批处理优势和 Elasticsearch 强大的全文检索引擎结合起来。我们既可以把 HDFS 的数据导入到 ES 里面做分析, 也可以将 ES 数据导出到 HDFS 上做备份



Lucene

搜索引擎基础，一个开放源代码的全文检索引擎工具包

Solr

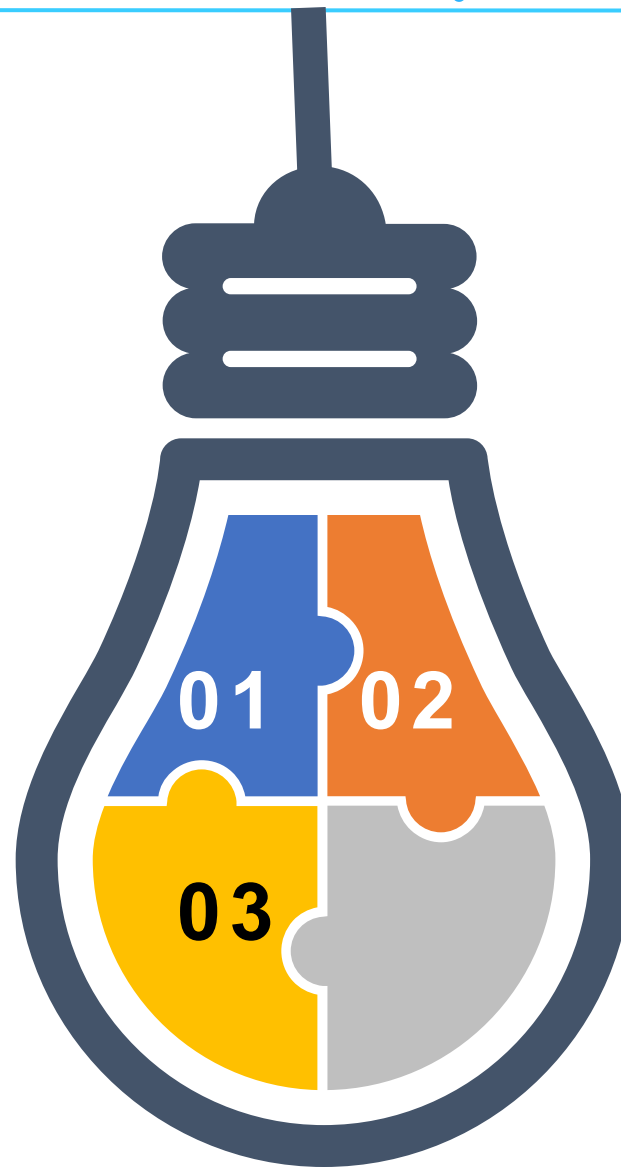
传统的搜索应用中表现好，不支持实时搜索

Elasticsearch:

支撑大规模分布式部署
实时搜索能力强，适合互联网场景

github使用ES做代码搜索

维基百科使用ES做词条搜索



Sacle 优化

热点榜单功能

搜索系统容错能力

为你推荐

美国大选

趋势

新闻

体育

娱乐

COVID-19

#SafetyMovie

Now Streaming on Disney+

由 Disney+ 推广

1 · 趋势

#fridaymorning

1.8万 推文

2 · 政治 趋势

#TrumpConceded





趋势 #LoserOfTheYear, Biden Administration

3 · 饶舌 趋势

Cudi

The Chosen One

@KidCudi · 21小时



Introducing my first very own shoe with Adidas!! The VADAWAM 326! Named after my awesome daughter Vada 🥰
❤️ Available Dec 17

显示这个主题帖

趋势 #MOTM3, #ManOnTheMoon

4 · 趋势

#FridayFeeling

1.94万 推文

5 · 趋势

Spready Mercury

People are loving the official names given to the fleet of snowplows in Scotland, from Spready Mercury to Sir Salter Scott

热度值 = 点赞数 + 收藏数 + 评论数 + 阅读数

推特这种高并发实时计算的排行榜，db 不适合，db扛不住这么大的并发量，高并发实时的排行榜天然适合用缓存来实现

版权归属于九章算法（杭州）科技有限公司，贩卖和传播盗版将被追究刑事责任

第 44 页

Redis 有序集合 Sorted Set

Sorted Set 不能重复，拥有一个权重 score 来从大到小排序。其可以排序的特点，可以应用于需要排序的场景

每一条推特是一个member，每条推特的热度值是一个 score

Zadd 命令：

- 向一个有序集合中加入一个或者多个元素及其分数

`zadd key value1 member1`

- 取出范围内的数据

`zrevrange key start end`



如何更新 Score?

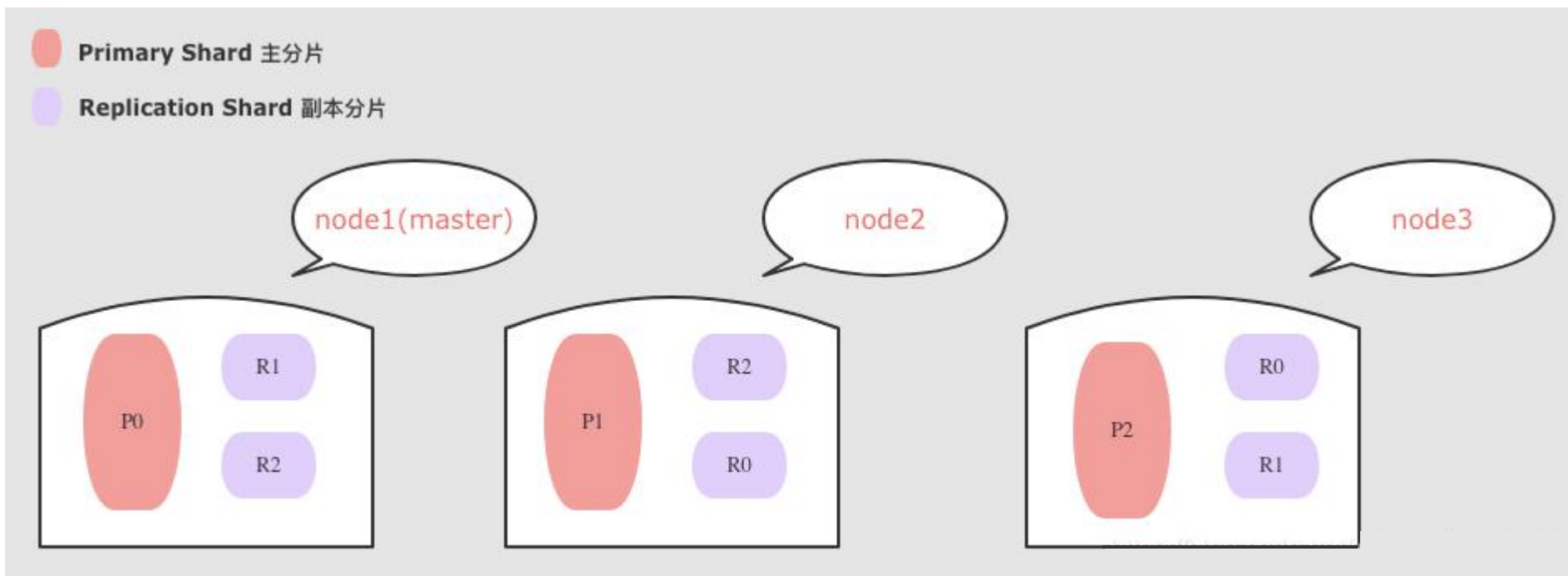
A. 实时更新

B. 每隔一段时间更新

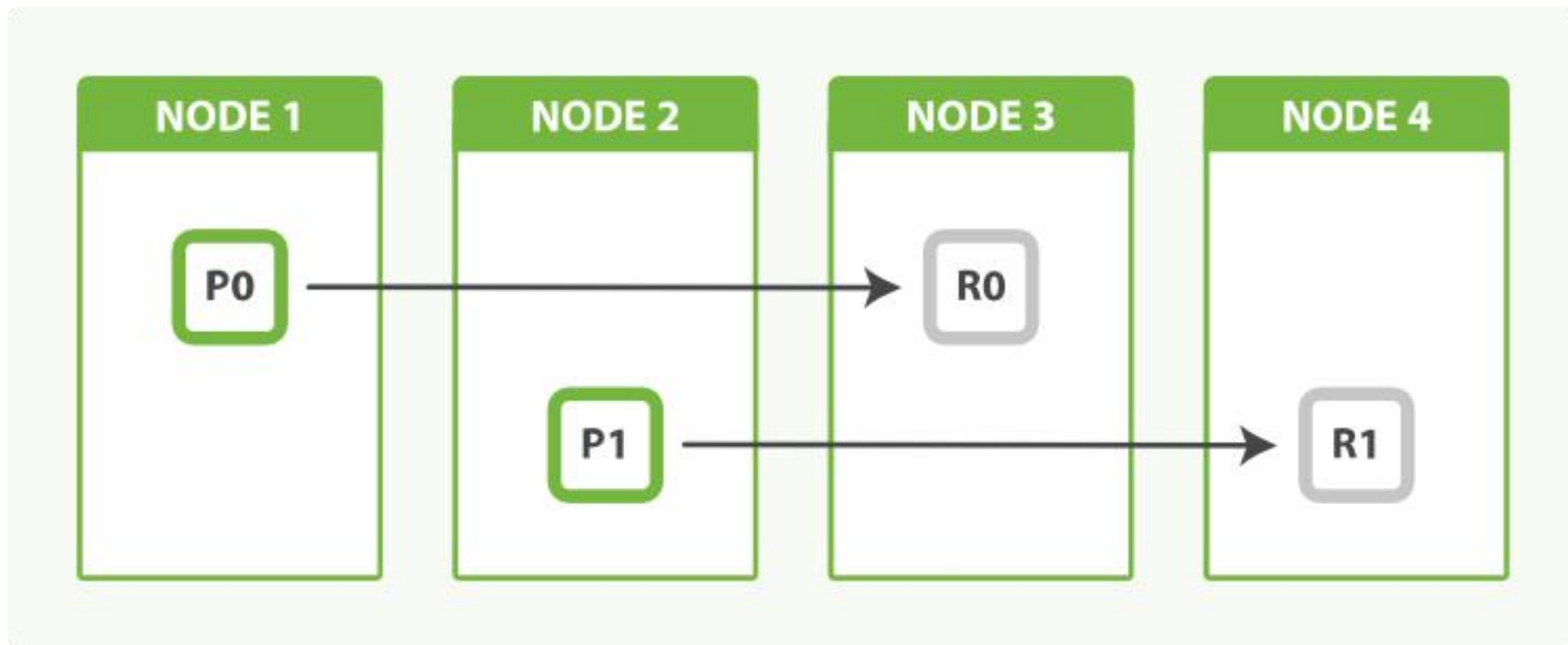


实时更新对系统压力大

- 每个索引有一个或多个分片，每个分片存储不同的数据。分片可分为主分片(primary shard)和复制分片(replica shard)，复制分片是主分片的拷贝。
- 往主分片服务器存放数据时候，会对应实时同步到备用分片服务器。
- 查询时候，所有（主、备）都进行查询。



- **故障转移**：提高系统的容错性，当某个节点某个分片损坏或丢失时可以从副本中恢复，有主分片的节点挂了，一个副本分片就会晋升为主分片
- **提高查询效率**：副本分片也提供查询能力，会自动对搜索请求进行^{load balance}负载均衡。



保留福利：**前15名**

私信班班：**【系统77】**



期末评价问卷↑

最后一节直播课啦，动动小手给老师评价一下哈~~♡ (´・ω・`) 比心

今日购课福利

- 《大厂高频算法面试特点及风格解析》
- 《背包四讲》
- 500学分&lintcode vip7天

毕业仪式打卡福利

- **毕业仪式**：于北京时间**01月24日**上午10:00在我们答疑群中举行哈~~（欢迎来抽奖）
- 在毕业仪式之前在群里打卡前面作业的同学，我们的作业奖励依然有效哈~~