Atitit 跨库查询 es mongodb区别总结

目录

[1. 数据库同步工具有没有 1](#_Toc1143)

[1.1. canal是用来实时同步mysql数据的 1](#_Toc19017)

[1.1.1. 1.首先大概介绍一下canal是干啥的？ 1](#_Toc18862)

[1.1.2. 2.canal是如何实时获取mysql数据的？ 2](#_Toc16612)

[1.1.3. Mongodb+ES如何使用 2](#_Toc836)

[2. 而聚合和查询功能 2](#_Toc27532)

[2.1. 数据结构固话 3](#_Toc28207)

[2.2. 除此之外，ES很好的支持了复杂聚合查询 3](#_Toc5826)

[3. 多字段组合查询 es更好 4](#_Toc8900)

[3.1.1. 技术栈成熟度项目背景 4](#_Toc26618)

[4. 两者的定位 5](#_Toc11605)

[5. 两者的很大区别在于源数据的存储和管理 6](#_Toc27715)

[6. 支持数据量比较 7](#_Toc22698)

[6.1. 3. 可靠性 OutOfMemory mongodb 胜出 7](#_Toc21844)

[6.2. 存储灵活性方面mongo完秒es ，需求可变 8](#_Toc8555)

****相同点****：两者都不支持SQL，没有事务支持，而且都是schema-free，都有Sharding和Replication，都有二级索引，都支持简单的聚合

**※Elasticsearch+Mongo这个架构主要针对场景：使用mongo存储海量数据，且这张表读写都很频繁。**

# 数据库同步工具有没有

## canal是用来****实时****同步mysql数据的

### **1.首先大概介绍一下canal是干啥的？**

canal是用来****实时****同步mysql数据的。对于离线任务可以通过sqoop将mysql业务库的数据导入hive数仓中计算，但是想要处理实时任务就要借助canal解析binlog日志来实现了。

官网的详细介绍： <https://github.com/alibaba/canal/wiki>

### **2.canal是如何实时获取mysql数据的？**

canal服务伪装成mysql的从节点，接收mysql主节点的binlog日志(binlog日志文件里面记录了数据库的实时操作)，然后解析binlog就知道mysql做了哪些操作。

### **Mongodb+ES如何使用**

ES支持插件，很多插件都是为了从主数据库导入到ES的，而从Mongodb导入的插件为Mongodb-river。可以参考：[How to use Elasticsearch with MongoDB?](https://stackoverflow.com/questions/23846971/how-to-use-elasticsearch-with-mongodb?rq=1)

# 而聚合和查询功能

，其实ES与Mongodb都是支持的，如果只从以上几点看，单独使用ES是可以的。

使用单独的ES，能够好的利用cpu和磁盘，数据不用在多个地方重复，cpu在不进行查询的时候可以给聚合统计使用。

如果聚合、查询、搜索的QPS非常高，那么也可以单独配置ES为搜索业务，Mongodb为存储查询聚合功能，功能上能够分离，维护上比较好。

另外，ES是基于Java的，而且在Github上很多的OOM问题，自己也遇到了OOM，所以从整个系统的稳定性上考虑Mongodb做存储、过滤和聚合，ES做搜索分工，两者独立，系统相对稳定。

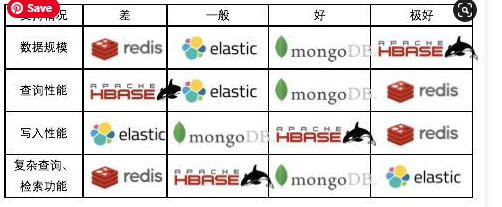
## 数据结构固话

说了这么多ES的优点，你是不是觉得ES简直万能了？可惜不是的，****ES也有很多的短处，最明显的就是字段类型无法修改、写入性能较低和高硬件资源消耗****。前边讲到ES会自动的替你建立索引，尽管这能给全文搜索以及聚合查询带来很多好处还能替你省了建索引这一麻烦事，但是这个特性也会带来一堆问题。ES需要在创建字段前要预先建立Mapping，Mapping中包含每个字段的类型信息，ES需要根据Mapping为字段建立合适的索引。由于这个Mapping的存在，ES中的字段一但建立就不能再修改类型了。（例如，你建的数据表的某个字段忘了加全文搜索，你想临时加上，但是表已经建好并且已经有很多数据了，这时候该怎么办呢？不好意思，你只能把整个数据表删了再重建一遍！）因此，ES在数据结构灵活度上高于MySQL但远不如MongoDB。ES的缺点还不止这些，自动建立索引使得ES的写入性能也收到了影响，要明显低于MongoDB，并且ES的写入还有一个更要命的问题，那就是默认1S的写入延迟，也就是说你的数据在写入后要至少等1S才能被查询到说了这么多ES的优点，你是不是觉得ES简直万能了？可惜不是的，****ES也有很多的短处，最明显的就是字段类型无法修改、写入性能较低和高硬件资源消耗****。前边讲到ES会自动的替你建立索引，尽管这能给全文搜索以及聚合查询带来很多好处还能替你省了建索引这一麻烦事，但是这个特性也会带来一堆问题。ES需要在创建字段前要预先建立Mapping，Mapping中包含每个字段的类型信息，ES需要根据Mapping为字段建立合适的索引。由于这个Mapping的存在，ES中的字段一但建立就不能再修改类型了。（例如，你建的数据表的某个字段忘了加全文搜索，你想临时加上，但是表已经建好并且已经有很多数据了，这时候该怎么办呢？不好意思，你只能把整个数据表删了再重建一遍！）因此，ES在数据结构灵活度上高于MySQL但远不如MongoDB。ES的缺点还不止这些，自动建立索引使得ES的写入性能也收到了影响，要明显低于MongoDB，并且ES的写入还有一个更要命的问题，那就是默认1S的写入延迟，也就是说你的数据在写入后要至少等1S才能被查询到

## ****除此之外，ES很好的支持了复杂聚合查询****

****复杂查询es比mongodb支持更好 ，，****

****这一特点还使得ES非常适合拿来作数据分析使用。****其实，ES还专门做了与自己配套的ELK套装，给你提供从日志收集到数据可视化分析的一条龙服务，绝对是构建高大上数据分析平台的利器。但是，ES的高成本和低写入性能这些缺点也注定了它不适合用在那些数据价值不高、对写入性能有要求、数据量大而成本受限的场景中。、



如果你的数据规模较大，对数据的读性能要求很高，数据表的结构需要经常变，有时还需要做一些聚合查询，选MongoDB；

# 多字段组合查询 es更好

因为做了很多索引更多的索引

业务系统查询操作日志记录会有很多过滤条件，且查询条件是任意组合的，现有MongoDB是不支持的，或者说所有关系型数据库都不支持，如果要支持，得创建好多组合的B+数索引，想法很不理智，这个我们已经在[《DB与ES混合之应用系统场景分析探讨》](https://mp.weixin.qq.com/s?__biz=MzI2NDExNTk5Mg==&mid=2247490796&idx=1&sn=ea9ff9bd3e244ad1eac47e083c54b983&scene=21" \l "wechat_redirect" \o "" \t "https://www.infoq.cn/article/_blank)文中探讨过，详细可以阅读；

同时主记录与从记录中有很多字符类的数据，这些数据查询即要支持精确查询，也要支持全文检索，这几个方面MongoDB功能很单一，性能也很糟糕，业务系统查询时经常超时，反倒是Elasticsearch非常合适。

### **技术栈成熟度项目背景**

分片与副本实现问题，MongoDB集合数据在设计时是需要绑定到具体的机器实例的，哪些分片分布在哪些节点上，哪些副本分布在哪些节点上，这些都需要在配置集群时就要绑定死，跟传统的关系型数据库做分库分表本质上没有什么两样，其实现在很多数据产品的集群还是这种模式偏多，比如Redis-cluster，ClickHouse等。而Elasticsearc的集群与分片和副本没有直接的绑定关系，可以任意的平衡调整，且节点的性能配置也可以很容易差异化；

操作日志数据量增加很快，单日写入超过千万条，不用多久，运维人员就需要对服务器进行扩容，且相对Elasticsearch复杂很多；

MongoDB单集合数据量超过10亿条，此情况下即使简单条件查询性能也不理想，不如Elasticsearch倒排索引快；

比较指标

|  |
| --- |
|  |
| 操作接口 | TCP | Restful(Http) |  |
| 是否支持分片 | 是 | 是 |  |
| 是否支持副本 | 是 | 是 |  |
| 选主算法 | Bully(霸凌) | Bully(霸凌) | 相比于Paxos和Raft算法实现更简单并有一定可靠性上的妥协，但是选举速度比较快 |
| 扩展难度 | 容易 | 非常容易 | es真的是我用过的扩展最方便的存储系统之一 |
| 配置难度 | 难 | 非常容易 |  |

# **两者的定位**

MongoDB和Elasticsearch都属于NoSQL大家族, 且都属于文档型数据存储

所以这两者的很多功能和特性高度重合, 但其实两者定位完全不同

MongoDB 是 ****文档型数据库****, 提供 ****数据存储和管理服务****  
Elasticsearch 是****搜索服务****, 提供 ****数据检索服务****

# 两者的很大区别在于源数据的存储和管理

* MongoDB作为一个数据库产品, 是拥有源数据管理能力的
* Elasticsearch作为一个搜索引擎, 定位是****提供数据检索服务****, 也就是说我只管查, 不管写 \_, Elasticsearch的Mapping不可变也是为此服务的, 带来的代价就是es不适合作为数据管理者, es可以从其他数据源同步数据过来提供查询, 但是不适合自己对数据进行存储和管理

es更侧重数据的查询, 各种复杂的花式查询支持的很好, 相比来说 MongoDB的查询能力就显得比较平庸了

由此可见, 对于个人, 如果你有一批数据要看, 但是不经常进行修改, 这个时候毫无疑问可以用es, 但是如果你还打算继续修改数据, 最好就是使用MongoDB，但其实对大多数人公司来讲，这两者的数据管理能力并没有多大的影响

ps: es修改Mapping的代价非常高, 所以我们一般都是把新数据重新写入一份新索引，然后直接切换读取的别名到新的索引

# 支持数据量比较

我们使用Elasticsearch存储的文档数量接近50亿（算上1份复制，接近100亿文档），总共10个数据节点和2个元数据节点（48GB内存，8核心CPU，ES使用内存达到70%），每天的文档增量大概是3000W条（速度持续增加中）。目前来看，单个文档的查询效率基本处于实时状态；对于1到2周的数据的聚合统计操作也可以在10秒之内返回结果。

但是，还有提升的空间：  
1. 对于查询单条数据的应用场景来说，我们可以使用ES的路由机制，将同一索引内的具有相同特征（比如具有相同的userid）的文档全部存储于一个节点上，这样我们之后的查询都可以直接定位到这个节点上，而不用将查询广播道所有的节点上；

## 3. 可靠性 OutOfMemory mongodb 胜出

OutOfMemory，我惨痛的回忆。当存储的数据量达到一定规模之后，你将面临许多在开发阶段无法被暴露出来的问题，例如：  
在某一个很清闲的下午，原本准备喝喝咖啡划划水的你，手贱地发了一个精心构造的搜索请求，而ES的过滤机制并没有准确地计算出这个精致的请求会消耗多少资源，然后你会一边卧槽一边眼睁睁地看着Cluster里面的节点逐个挂掉。你知道ES正在很努力地试图自救，把几个T的数据拷贝来拷贝去，但你其实根本不知道它在干什么，最后你决定挨个节点手动重启（同样不知道会造成什么后果），五个半小时后，感谢各路神佛，状态从红色变成黄色，第二天早上，黄色才变成绿色。想想如果知乎挂成这样会发生什么吧。  
这个问题长期以来一直存在，在ES的GitHub主页上有30多个Open Issue都是关于Out of Memory的，当然前期充分的压力测试某种程度上可以缓解这个问题，但也可能无限期地推迟产品上线的时间。

## 存储灵活性方面mongo完秒es ，需求可变

，用es做主存储时你要想清楚，因为  
**mapping是不可变！**  
**mapping是不可变！**  
**mapping是不可变！**  
当需求经常变时候最好用mongo。需求变化时候用es你会想死，数据量少还好，数据量多你能想像一下几亿的数据慢慢scan然后全部重建的痛苦吗？想死的心都有。

作者：老司机  
链接：https://www.zhihu.com/question/25535889/answer/101539800  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。