**有什么异常日志？**

      ESM，集群管理–集群监控–左上“节点” ，点"详情" ，可以查看各个节点的日志

ES使用最佳实践：

* 写入的数据考虑清楚是否会过期，ES擅长的不是存储而是搜索，所以一般存入ES的数据难免会随着时间删除旧数据。删除方法有两种：~~①按记录（不推荐）~~②按索引。推荐使用后者，所以需要业务根据数据特点，按天、月、季度等创建索引。
* 分片数：够用就好，不要过多不要过少。
* ES不是数据库，不建议做频繁的更新。

[date类型内置计算](https://www.elastic.co/guide/en/elasticsearch/reference/5.3/common-options.html#date-math)

**能修改集群特殊的配置吗？**比如修改分页限制，修改bool查询条件的最大子句等，一般不建议修改，超过限制很容易OOM，该优化的ES团队已经做过优化。当你想改这些参数的时候，大多情况下是考虑业务架构是否合理？如果考虑好了，非要修改，发需要业务总监回复邮件告知出了问题后果自负

怎么删除一个type下的数据？

     ①自己写查询然后删除 ②使用delete\_by\_query 插件(esm首页给的例子有[ES Java example](http://source.jd.com/app/es-example.git))  ③ 建议基于索引删除数据，新版本已不推荐使用type，应考虑多建索引。

**集群可承受的TPS是多少？**  
     这个没有具体的数字，根据不同规模集群、不同的索引结构等不同，建议根据业务评估自己的流量，压测双倍流量，若ES无法承受或满足，找ES运维协商扩容集群，不要流量暴增于平时的3倍、4倍、甚至更多的规模，ES运维会定期巡检各个集群的流量，按需缩容扩容各集群，集群暴增流量可能会导致集群不稳定，甚至挂掉。

**导数据到ES？**  
                  <http://bdp.jd.com/buffalo4/task/list.html>

* [1.ES不是什么](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-1.ES不是什么)
* [2.为什么一条数据写入后马上查询查不到？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-2.为什么一条数据写入后马上查询查不到？)
* [1.2.跨机房问题](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-1.2.跨机房问题)
* [1.3.如何解决数据写入后想马上查询到](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-1.3.如何解决数据写入后想马上查询到)
* [2. 为什么同样的query多次查询的数据不一致？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-2.为什么同样的query多次查询的数据不一致？)
* [3. bulk写入后数据条目不一致](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-3.bulk写入后数据条目不一致)
* [3.1. ES的能力边界是什么](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-3.1.ES的能力边界是什么)
* [4.es中不分词，哪些查询语句是模糊查询的？我指的模糊查询是查询 “裤子”  \*\*\*裤子\*\*\*都会查找出来](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-4.es中不分词，哪些查询语句是模糊查询的？我指的模糊查询是查询“裤子”***裤子***都会查找出来)
* [5.一个ES的index能存多少数据](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-5.一个ES的index能存多少数据)
* [6.IllegalArgumentException: number of documents in the index cannot exceed 2147483519](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-6.IllegalArgumentException:numberofdocumentsintheindexcannotexceed2147483519)
* [6.1.使用ES的条件](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-6.1.使用ES的条件)
* [7.给个ES使用文档参考](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-7.给个ES使用文档参考)
* [8.给个ES使用例子参考](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-8.给个ES使用例子参考)
* [9.ES2.1需要的jar的版本要求：](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-9.ES2.1需要的jar的版本要求：)
* [10.ES client版本问题](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-10.ESclient版本问题)
* [11.Date数据为什么差8小时](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-11.Date数据为什么差8小时)
* [12.ES内部分享的资料](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-12.ES内部分享的资料)
* [13.ES的版本是什么？索引存在了什么地方？几副本？存储容量是多少？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-13.ES的版本是什么？索引存在了什么地方？几副本？存储容量是多少？)
* [14.如何优化写性能](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-14.如何优化写性能)
* [15.如果优化读性能](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-15.如果优化读性能)
* [15.1.容量规划(科学的确定shard的数量)](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-15.1.容量规划(科学的确定shard的数量))
* [16.怎么申请服务](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-16.怎么申请服务)
* [17.我的业务很重要，我需要独立的物理机部署](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-17.我的业务很重要，我需要独立的物理机部署)
* [18.除了集群外，其他的东西为什么要申请，比如mapping的创建，修改，index的创建，删除等等。](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-18.除了集群外，其他的东西为什么要申请，比如mapping的创建，修改，index的创建，删除等等。)
* [19.用http的方式如何访问es集群](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-19.用http的方式如何访问es集群)
* [20.我的mapping合不合理](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-20.我的mapping合不合理)
* [21.mapping能否更新](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-21.mapping能否更新)
* [21.2mapping冲突](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-21.2mapping冲突)
* [22.定期创建索引](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-22.定期创建索引)
* [23.什么是template（模板）](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-23.什么是template（模板）)
* [24.ES2.1版本元字段属性](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-24.ES2.1版本元字段属性)
* [25.routing是什么？为什么要用routing？能带来什么好处？有什么坏处](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-25.routing是什么？为什么要用routing？能带来什么好处？有什么坏处)
* [26.能否对一条document指定多个rouing key,让它分配到不同的shard上去，这样我的Aquery就可以只查Ashard上，B query只查B shard](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-26.能否对一条document指定多个rouingkey,让它分配到不同的shard上去，这样我的Aquery就可以只查Ashard上，Bquery只查Bshard)
* [27.为什么申请别名？为什么要为读写申请不同的别名？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-27.为什么申请别名？为什么要为读写申请不同的别名？)
* [28.能否按天创建索引？是否提供定时删除过期索引？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-28.能否按天创建索引？是否提供定时删除过期索引？)
* [29.一个典型的错误查询用例](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-29.一个典型的错误查询用例)
* [30.ES java client](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-30.ESjavaclient)
* [31.java.lang.OutOfMemoryError: Direct buffer memory](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-31.java.lang.OutOfMemoryError:Directbuffermemory)
* [32.java.lang.OutOfMemoryError: unable to create new native thread](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-32.java.lang.OutOfMemoryError:unabletocreatenewnativethread)
* [33.ClassNotFoundException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-33.ClassNotFoundException)
* [34.NoSuchMethodError](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-34.NoSuchMethodError)
* [35.NoNodeAvailableException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-35.NoNodeAvailableException)
* [36.TransportSerializationException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-36.TransportSerializationException)
* [37.如何检查ES server的版本](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-37.如何检查ESserver的版本)
* [38.IllegalStateException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-38.IllegalStateException)
* [39.InvalidIndexNameException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-39.InvalidIndexNameException)
* [40.TypeMissingException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-40.TypeMissingException)
* [41.什么是mapping？写数据必须要创建mapping吗？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-41.什么是mapping？写数据必须要创建mapping吗？)
* [42.如何创建mapping](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-42.如何创建mapping)
* [43.是否应该完整的提交mapping申请](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-43.是否应该完整的提交mapping申请)
* [44.TooManyClauses / maxClauseCount](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-44.TooManyClauses/maxClauseCount)
* [45.EsRejectedExecutionException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-45.EsRejectedExecutionException)
* [46.深分页的问题](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-46.深分页的问题)
* [47.深分页的危害：](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-47.深分页的危害：)
* [48.Result window is too large / 修改max\_result\_window](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-48.Resultwindowistoolarge/修改max_result_window)
* [49.正确导出数据的姿势](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-49.正确导出数据的姿势)
* [50.如何处理bigint/bigdecimal类型，尤其涉及到金钱的时候](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-50.如何处理bigint/bigdecimal类型，尤其涉及到金钱的时候)
* [51.ES的默认排序规则](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-51.ES的默认排序规则)
* [52.Fielddata的危害和doc value的解救](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-52.Fielddata的危害和docvalue的解救)
* [53.不要用post filter](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-53.不要用postfilter)
* [54.ScriptException are disabled](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-54.ScriptExceptionaredisabled)
* [55.IndexNotFoundException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-55.IndexNotFoundException)
* [56.如何确定分片数](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-56.如何确定分片数)
* [57.ES维护团队后继会提供重灌数据的服务](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-57.ES维护团队后继会提供重灌数据的服务)
* [58.Index属性：Analyzed VS not\_analyzed VS No](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-58.Index属性：AnalyzedVSnot_analyzedVSNo)
* [59.MaxBytesLengthExceededException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-59.MaxBytesLengthExceededException)
* [60.Type VS New Index](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-60.TypeVSNewIndex)
* [61.\_source的意义](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-61._source的意义)
* [62.ES的update](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-62.ES的update)
* [63.VersionConflictEngineException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-63.VersionConflictEngineException)
* [64.中文分词的问题](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-64.中文分词的问题)
* [65.Query VS filter(2.X将filter合并到过滤语境中了)](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-65.QueryVSfilter(2.X将filter合并到过滤语境中了))
* [66.ES是如何实现节点容灾的](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-66.ES是如何实现节点容灾的)
* [67.ES扩容/缩容对业务的影响](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-67.ES扩容/缩容对业务的影响)
* [68.ES的TTL的坑](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-68.ES的TTL的坑)
* [69.ES TTL的使用经验](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-69.ESTTL的使用经验)
* [70.ES TTL替换方案](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-70.ESTTL替换方案)
* [71.esm管理端看到的 disk free 0GB 有没有问题？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-71.esm管理端看到的diskfree0GB有没有问题？)
* [72.ES的TPS是多少？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-72.ES的TPS是多少？)
* [73. 2.1的客户端可以正常的操作1.7的es么？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-73.2.1的客户端可以正常的操作1.7的es么？)
* [74.在index时对字段用ik分词，查询时需要指定分词器吗？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-74.在index时对字段用ik分词，查询时需要指定分词器吗？)
* [75.QueryParsingException  Caused by: org.elasticsearch.index.query.QueryParsingException: [index\_name] No field specified for term filter](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-75.QueryParsingExceptionCausedby:org.elasticsearch.index.query.QueryParsingException:[index_name]Nofieldspecifiedfortermfilter)
* [76.NumberFormatException Caused by: java.lang.NumberFormatException: For input string: "null"](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-76.NumberFormatExceptionCausedby:java.lang.NumberFormatException:Forinputstring:"null")
* [77.集群Index TPS,Search TPS与副本数的关系](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-77.集群IndexTPS,SearchTPS与副本数的关系)
* [78.IllegalArgumentException      Caused by: java.lang.IllegalArgumentException: Invalid format: ""](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-78.IllegalArgumentExceptionCausedby:java.lang.IllegalArgumentException:Invalidformat:"")
* [79.SearchParseException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-79.SearchParseException)
* [80.InvalidIndexNameException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-80.InvalidIndexNameException)
* [81.获取文档数](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-81.获取文档数)
* [82. 在ES2.X 中 delete-by-query插件的使用方法](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-82.在ES2.X中delete-by-query插件的使用方法)
* [83.MergeMappingException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-83.MergeMappingException)
* [84.如何获取线上索引最新的mapping信息？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-84.如何获取线上索引最新的mapping信息？)
* [85.ClassCastException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-85.ClassCastException)
* [86.需要ES安装包](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-86.需要ES安装包)
* [87. delete-by-query插件](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-87.delete-by-query插件)
* [88. IllegalStateException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-88.IllegalStateException)
* [89.  ReceiveTimeoutTransportException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-89.ReceiveTimeoutTransportException)
* [90.  VerifyError](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-90.VerifyError)
* [91 docker 线程数太多，或者 非docker线程多。](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-91docker线程数太多，或者非docker线程多。)
* [92  我的数据需要多大的集群？](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-92我的数据需要多大的集群？)
* [93   ES怎么查看数据](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-93ES怎么查看数据)
* [94 怎么添加别名](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-94怎么添加别名)
* [95  关于script](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-95关于script)
* [96  ES transport client的connection问题](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-96EStransportclient的connection问题)
* [97 关于inner\_hits](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-97关于inner_hits)
* [98  throttling indexing: numMergesInFlight=10, maxNumMerges=9](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-98throttlingindexing:numMergesInFlight=10,maxNumMerges=9)
* [99 SearchContextMissingException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-99SearchContextMissingException)
* [100  org.apache.hadoop.ipc.RemoteException.unwrapRemoteException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-100org.apache.hadoop.ipc.RemoteException.unwrapRemoteException)
* [101  NotXContentException  not\_x\_content\_exception](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-101NotXContentExceptionnot_x_content_exception)
* [102 .ParserMinimalBase.\_reportError(ParserMinimalBase.java:533)](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-102.ParserMinimalBase._reportError(ParserMinimalBase.java:533))
* [103. Updating Child Mapping with new field fails with NullPointerException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-103.UpdatingChildMappingwithnewfieldfailswithNullPointerException)
* [104  queryparser.classic.TokenMgrError:](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-104queryparser.classic.TokenMgrError:)
* [105 java.io.EOFException: Premature EOF: no length prefix available](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-105java.io.EOFException:PrematureEOF:nolengthprefixavailable)
* [106 StreamCorruptedException](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-106StreamCorruptedException)
* [107 esm管理端执行出现"index": "bad-request"  "reason": "no such index"](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-107esm管理端执行出现"index":"bad-request""reason":"nosuchindex")
* [108 有数据查询不到](http://jpcloud.jd.com/display/cloud/ES+FAQ#ESFAQ-108有数据查询不到)

**1.ES不是什么**

* 1. ES不是可靠的存储系统，不是数据库，它有丢数据的风险。具体请参考[ES内部分享](http://jpcloud.jd.com/pages/viewpage.action?pageId=16629275)中的ES简介这个文档

**2.为什么一条数据写入后马上查询查不到？**

* 1. ES不是实时系统，数据写入成功只是trans log成功（类似于mysql的bin log），写入成功后立刻查询查不到是正常的。因为数据此刻可能还在内存里而不是进入存储引擎里。
  2. 同理，删除一条数据后也不是马上消失
  3. 写入何时可查询？ES内部有一个后台线程，定时将内存中的一批数据写入到存储引擎，此后数据可见
  4. 默认后台线程一秒运行一次。该线程运行的越频繁，写入性能越低。运行的频率越低，写入的性能越高（不会无限高）

**1.2.跨机房问题**

a. ES官方不推荐跨机房部署。<https://www.elastic.co/blog/clustering_across_multiple_data_centers>

b. 除了上述问题外，同时我们在公司内部署双机房测试，读写性能均下降15%

c. 而且双机房部署后，由于master节点的奇数特征，势必会出现某一个机房并不能独立工作的情况

**1.3.如何解决数据写入后想马上查询到**

* 1. 写入数据后手工调用refresh这个API即可
  2. 没有免费的午餐，这个API调用的频繁会影响写入性能

**2. 为什么同样的query多次查询的数据不一致？**

  a. 由于shard的主分片和副本是由独立的节点去刷新的，刷新的频率并不同步，这样同样的query发送到不同的分片（主片和从片）上看到的数据是不同的，导致的结果是查询到的数据也不完全同步。简单说，ES是一个最终一致性系统

**3. bulk写入后数据条目不一致**

a. 对数据可靠性要求高的话，用bulk批量写入后需要展开bulkrespose的每一个条目，判断每一条数据的写入成功情况

                  ActionRequestBuilder#execute(ActionListener<Response>)，注册一个listener，在onResponse中遍历bulk中每个请求的执行情况，对于失败的请求，可以重试或者log...

**3.1. ES的能力边界是什么**

* 1. 目前已知的单ES集群可以存储PB级别的数据，不过这个就非常费劲了。TB级别数据没压力

**4.es中不分词，哪些查询语句是模糊查询的？我指的模糊查询是查询 “裤子”  \*\*\*裤子\*\*\*都会查找出来**

* 1. 有这样需求的赶紧走数据库

**5.一个ES的index能存多少数据**

* 1. ES的index分成若干shard，每个shard最大能存20亿条数据，这是个硬限制

**6.IllegalArgumentException: number of documents in the index cannot exceed 2147483519**

* 1. 超过上头的限制报的异常.一个分片最多能存 2147483519 个文档，这是ES的门槛，只能删掉部分数据，或者重新建索引。

**6.1.使用ES的条件**

* 1. 如果使用ES官方提供的jar包访问，需要JDK1.7及以上
  2. 升级JDK到1.7有问题的话，请自行用HTTP包装API

**7.给个ES使用文档参考**

* 1. <https://www.elastic.co/guide/index.html>

**8.给个ES使用例子参考**

* 1. [Demo说明](http://jpcloud.jd.com/pages/viewpage.action?pageId=15337818)

**9.ES2.1需要的jar的版本要求：**

* 1. JDK1.7及以上

**10.ES client版本问题**

* 1. 使用对应的版本访问ES server。如果ES server端的版本是1.7，那么请使用ES 1.7的client。如果ES server是2.1，请使用2.1的client
  2. 用低版本访问过程中出现问题我们不负责处理。

**11.Date数据为什么差8小时**

* 1. 时区问题：<http://josh-persistence.iteye.com/blog/2230074>    //北京时间可以表示为GMT+8:00
  2. 如何正确的写入date类型的数据到ES：.field("date", new Date(), ISODateTimeFormat.dateTime().withZone(DateTimeZone.forID("Asia/Shanghai")))

**12.ES内部分享的资料**

* 1. [ES内部分享](http://jpcloud.jd.com/pages/viewpage.action?pageId=16629275)

**13.ES的版本是什么？索引存在了什么地方？几副本？存储容量是多少？**

* 1. 目前提供的版本是2.1
  2. 索引存在linux服务器的文件系统之上（背后是文件系统，不是类似于HDFS的分布式文件系统）
  3. 副本数目由业务方根据需求自己确定
  4. 可以认为存储空间没有限制（存储空间不够的话ES服务方会动态加节点、加硬盘）

**14.如何优化写性能**

* 1. <https://www.elastic.co/blog/performance-considerations-elasticsearch-indexing>

**15.如果优化读性能**

* 1. <https://www.elastic.co/blog/found-optimizing-elasticsearch-searches>

**15.1.容量规划(科学的确定shard的数量)**

<https://www.elastic.co/guide/en/elasticsearch/guide/master/capacity-planning.html>

**16.怎么申请服务**

* 1. 我们提供一个测试环境供大家测试功能：[杰思](http://jpcloud.jd.com/pages/viewpage.action?pageId=15336328)
  2. 测试环境说明：[ES测试环境部署](http://jpcloud.jd.com/pages/viewpage.action?pageId=16618373)
  3. 线上服务申请流程请参考：  
     TODO: 王松林补充文档

**17.我的业务很重要，我需要独立的物理机部署**

* 1. 抱歉，申请物理机不在我们这，我们这只提供ES服务。只有服务，没有硬件，没有物理机，没有CPU。这些申请统统不在我们这。
  2. 但是我也理解，每个人的业务都很重要，怎么办？提供你们的QoS需求，比如需要写入的TPS是多少，查询的TP99需要保证多少等等，然后咱们一起在有限的资源上保障业务方的QoS

**18.除了集群外，其他的东西为什么要申请，比如mapping的创建，修改，index的创建，删除等等。**

* 1. 首先，目前我们做不到强制让你申请，因为所有的API都是不鉴权的，我们只能建议申请，做不到强制申请。但是，建议大家该申请的申请
  2. 但是，在2016年三季度（最晚四季度）我们会开发出ES的权限模块，此时，我们就能做到强制申请
  3. 为什么要申请？
     1. 前期一是帮助大家review明显的错误；二是为啥数据安全
     2. 后期我们维护的就不是ES了，而是ES协议兼容的存储系统，有些ES支持的我们就不支持了；有些ES不支持的我们会支持。

**19.用http的方式如何访问es集群**

* 1. ES暂时未与VIP打通，业务可自行申请VIP

**20.我的mapping合不合理**

1. 对不起，不知道
2. 要想跟我们讨论你的mapping合不合理，必须把你们的全部query都拿出来，这才有讨论的可能。
3. 什么叫全部的query？
   1. select \* from type1 where fieldA = XXX
   2. select \* from type1 where fieldA = XXX and fieldB = XXX
   3. select \* from type1 where fieldA = XXX and fieldB = XXX sort on fieldC
   4. select count( \* ) from type1 where fieldA = XXX
   5. ...

**21.mapping能否更新**

* 1. Other than where documented, **existing type and field mappings cannot be updated**. Changing the mapping would mean invalidating already indexed documents. Instead, you should create a new index with the correct mappings and reindex your data into that index.
  2. 存在的字段不能被更新和删除
  3. 不存在的字段可以添加

**21.2mapping冲突**

<https://www.elastic.co/blog/great-mapping-refactoring#conflicting-mappings>

**22.定期创建索引**

* 1. 有些业务需要定期创建索引，比如日志，订单，运单等。这种需求是通过申请一个template，然后ES服务提供方负责定期创建索引完成的

**23.什么是template（模板）**

* 1. 模板是描述表结构（mapping）和表设置(setting)的数据结构
  2. 模板特殊在它可以描述一组mapping和setting的组合。描述一组的功能是通过名字的正则表达式实现的
  3. tempate的一个例子：[TimLine图片20160715110214.png](http://jpcloud.jd.com/download/attachments/16629296/TimLine%E5%9B%BE%E7%89%8720160715110214.png?version=1&modificationDate=1468551751000&api=v2)

原理：

1. 1 在ES中创建一个索引模板，其本质就是包含mapping及分片信息，模板中有个属性  template  表示该模板用来匹配什么样的索引。比如 "**template:abc\***“，创建的所有以abc开头的索引，将会应用这个模板，一般用来匹配按 周期 创建的索引，例如 abc-2017-02-03；
2. 2 通过定时任务创建索引，菜单在 “”集群管理–定时任务配置” 创建一个定时任务，按照特定间隔创建索引。比如按1天。

**24.ES2.1版本元字段属性**

* 1. [ES2.1元字段的属性](http://jpcloud.jd.com/pages/viewpage.action?pageId=16636809)

**25.routing是什么？为什么要用routing？能带来什么好处？有什么坏处**

* 1. rouing就是hash key， 这个key决定写入/查询的分片id
  2. 如果你在写数据和查询的时候从来没有涉及到routing,那么ES在写入的时候用document的id做routing决定这条数据写到哪个分片上。查询的时候则把query下发到所有分片
  3. 为了避免把query下发到所有的分片进而提升检索性能，会有这种技巧：根据查询的特征，比如我的查询都是用用户id去查，那么在写入数据的时候可以根据用户id去决定分片；在查询的时候也用用户ID做shard key将query只下发到相关的分片上。这是routing能带来的最大和唯一的好处
  4. 但是，这是有条件的，就是你的query确实是有特征的
  5. 坏处是根据某一特定值做routing之后，可能带来数据倾斜，使得某些分片承载的数据很多而另外一些分片承担的数据很少。

**26.能否对一条document指定多个rouing key,让它分配到不同的shard上去，这样我的Aquery就可以只查Ashard上，B query只查B shard**

* 1. rouing是hash key, 你以为你用多个routing key就一定能分配到不同的shard上（没有hash冲突？！！），我只能说太navie了。
  2. 所以写数据的时候指定多个routing key这事就别想了

**27.为什么申请别名？为什么要为读写申请不同的别名？**

* 1. 请参考官方文档：<https://www.elastic.co/guide/en/elasticsearch/guide/current/index-aliases.html>

**28.能否按天创建索引？是否提供定时删除过期索引？**

* 1. 能按天创建索引
  2. 提供定时删除过期索引

**29.一个典型的错误查询用例**

* 1. search\_type[DFS\_QUERY\_THEN\_FETCH], total\_shards[32], source[{"from":0,"size":1,"query":{"term":{"waybillCode":"19966779685"}},"explain":true,"fields":["updateTime","updateTimeM","updateTimeE","tsC","tsE","tsM"]}], extra\_source[]
  2. 错误点：
     1. search type不要用DFS\_QUERY\_THEN\_FETCH，用query\_then\_fetch足够（除非能提供充足的理由）
     2. 尽可能的用filter，它快
     3. 不要用explain:true，这个是打印debug相关的信息的
     4. 不要用fields, 在client端去解码

**30.ES java client**

* 1. ES java client是线程安全的，全局构建一个即可满足读写需求，不要每次都创建ES client。否则会出现下面这个问题

**31.java.lang.OutOfMemoryError: Direct buffer memory**

* 1. 完整异常：  
     org.elasticsearch.netty.channel.socket.nio.AbstractNioSelector - Unexpected exception in the selector loop.
  2. 每次访问ES都构建新的es client即会抛出次异常。解决方案：全局构建一个client

**32.java.lang.OutOfMemoryError: unable to create new native thread**

* 1. ES进程启动时从操作系统中拿到的Max processes过小而ES开启的线程数超过该值，从而导致ES进程不能开启其他工作线程

**33.ClassNotFoundException**

* 1. 检查ES client用到的依赖包的版本是否大于等于demo里的包的版本（最好是等于）

**34.NoSuchMethodError**

* 1. 很多用户跑demo时会报出这个异常，原因是ES需要的很多jar包的版本比你们项目中使用的相同的jar包要高，此时需要升级相关jar包

**35.NoNodeAvailableException**

* 1. 查看链接的ip地址和端口是否正确。经常有人把TCP的端口错配为HTTP的端口
  2. 查看cluset name是否正确（cluster name错了也抛这个异常）
  3. 使用和ES server版本匹配的ES Client 版本
  4. 如果是在连上ES并且运行一段时间后出现，很有可能是某个链接的节点出问题了。此时请大家不要紧张，ES会自动切换到其他节点访问

**36.TransportSerializationException**

* 1. Caused by: org.elasticsearch.transport.TransportSerializationException: Failed to deserialize exception response from stream
  2. 检查ES client的版本与server的版本是否一致

**37.如何检查ES server的版本**

* 1. 

**38.IllegalStateException**

* 1. java.lang.IllegalStateException: Message not fully read (request) for requestId
  2. 检查是否有低版本的ES client访问高版本的ES server

**39.InvalidIndexNameException**

* 1. 索引的名字有如下规范，如果违反如下规范，则抛出上述异常
     1. 索引名字不能包含非法的文件名称字符（\，/，\*，?，"，<，>，|，空格，逗号 ）
     2. 索引名字不能包含#
     3. 索引名字不能以下划线\_开头
     4. 索引名称如果包含因为字母，则必须为小写

**40.TypeMissingException**

* 1. ES的index是逻辑数据库，type是逻辑表
  2. 这个异常的意思是逻辑表不存在
  3. 逻辑表如果是通过ES系统申请创建的，那么查查申请的mapping
  4. 逻辑表如果是通过程序创建的，你们自己查自己的程序

**41.什么是mapping？写数据必须要创建mapping吗？**

* 1. Mapping相当于数据库的表结构
  2. 在写数据前ES不强制要求创建mapping，因为ES有动态识别和创建的机制
  3. 非常不建议使用ES的动态识别和创建的机制，因为很多情况下这并非你所需要。推荐的做法是在写数据之前仔细的创建mapping

**42.如何创建mapping**

* 1. <http://jpcloud.jd.com/pages/viewpage.action?pageId=16624064>
  2. 测试环境创建mapping请自行创建
  3. 线上环境请在系统提交申请。

**43.是否应该完整的提交mapping申请**

* 1. 是的。虽然ES有dynamic mapping(dynamic schema)的功能，单这通常会造成很大的问题，所以我们都ES集群都是关闭该功能的，需要预先提交完整的mapping

**44.TooManyClauses / maxClauseCount**

* 1. 使用terms query, prefix query, fuzzy query, wildcard query或者range query等都会出现这个异常
  2. 上述query都是语法糖，在搜索引擎内部是转换为boolean query查询的
  3. 以terms query为了，其实是构建了多个term query为子语句的全or关系的一个boolean query
  4. 为了避免内存消耗过多和时间消耗过长，搜索引擎内部对boolean query的子语句个数是有限制的，超过这个限制就会抛出TooManyClauses
  5. 最好的解决办法是调整自己的查询语句，不要出现TooManyClauses
  6. index.query.bool.max\_clause\_count:1024  在V5中indices.query.bool.max\_clause\_count    <https://www.elastic.co/guide/en/elasticsearch/reference/5.0/breaking_50_settings_changes.html#_search_settings>

**45.EsRejectedExecutionException**

* 1. ES处理读写也是使用了常规的线程池。有一定数目的线程处理读写；超过线程的请求被放到等待队列中，如果等待队列满了，就会抛出EsRejectedExecutionException。
  2. 需要正确评估自己的读写并发量并与ES维护团队沟通，通过设置合适的线程数和等待队列长度解决
  3. ES作为存储系统，有自己的处理能力上限，业务方也需要利用消息队列等系统“削峰”保护ES
  4. 如果你碰到了队列被拒，一般来说都是批量索引请求导致的。[通过并发导入程序发送大量批量请求非常简单。越多越好嘛，对不？](https://www.elastic.co/guide/cn/elasticsearch/guide/current/_monitoring_individual_nodes.html#id-1.10.3.7.12.7.2.2)

事实上，每个集群都有它能处理的请求上限。一旦这个阈值被超过，队列会很快塞满，然后新的批量请求就被拒绝了。

这是一件 \_好事情\_ 。队列的拒绝在回压方面是有用的。它们让你知道你的集群已经在最大容量了。这比把数据塞进内存队列要来得好。增加队列大小并不能增加性能，它只是隐藏了问题。当你的集群只能每秒钟处理 10000 个文档的时候，无论队列是 100 还是 10000000 都没关系——你的集群还是只能每秒处理 10000 个文档。

队列只是隐藏了性能问题，而且带来的是真实的数据丢失的风险。在队列里的数据都是还没处理的，如果节点挂掉，这些请求都会永久的丢失。此外，队列还要消耗大量内存，这也是不理想的。

在你的应用中，优雅的处理来自满载队列的回压，才是更好的选择。当你收到拒绝响应的时候，你应该采取如下几步：

* + 1. 暂停导入线程 3–5 秒。
    2. 从批量操作的响应里提取出来被拒绝的操作。因为可能很多操作还是成功的。响应会告诉你哪些成功，哪些被拒绝了。
    3. 发送一个新的批量请求，只包含这些被拒绝过的操作。
    4. 如果依然碰到拒绝，再次从步骤 1 开始。

通过这个流程，你的代码可以很自然的适应你集群的负载，做到自动回压。

拒绝不是错误：它们只是意味着你要稍后重试。  
参考：<https://www.elastic.co/guide/cn/elasticsearch/guide/current/_monitoring_individual_nodes.html>

**46.深分页的问题**

* 1. ES处理查询的流程如下：
     1. Client需要第N到N+m条结果
     2. 接到这个请求的ES server（后继称之为协调者）向每一个数据分片所在的数据节点发送请求
     3. 每一个数据节点都需要向协调者返回(N+m)个结果
     4. 如果有n个数据分片，那么协调者拿到n \* (N+m)个结果，排序，扔掉(n-1) \* (N+m)个结果，返回给client N+m个结果
     5. 如果N是10W，100W，那么协调者的内存压力会非常大
     6. 在我们目前维护的2.1版本中，ES已经不容许N>10000了

**47.深分页的危害：**

* 1. 导致打爆节点内存引起集群整体不可用

**48.Result window is too large / 修改max\_result\_window**

1. 接上条，如果查询的N>10000,在2.X版本中就会抛出Result window is too large的异常
2. 想要修改max\_result\_window达到深分页的，请**发邮件 的标题和内容包含如下的声明方式(加粗加大、红色字体)，**并发送给部门领导审批再找ES团队执行。

标题：**申请修改 jiesi-编号  的危险参数 max\_result\_window**

内容：**我们业务方已了解修改max\_result\_window后的危害是可能导致打爆集群，造成集群整体不可用等相关危害，需要重启集群才能恢复，也深知ES团队强烈建议通过scroll解决或调整业务架构而不要修改该参数， 但我们业务方综合考虑后，也报批领导同意，愿意自行负责一切后果，希望ES团队对集群执行如下修改：**

|  |
| --- |
| PUT index/\_settings  {      "index": {          "max\_result\_window": 20000     }  } |

**49.正确导出数据的姿势**

* 1. 接上条，有人会说，我想把数据都导出，但是又会抛Result window is too large，怎么办？
  2. ES已经提供了解决方式，scroll/scan,相当于数据库的游标方式。开启了scroll/scan后，只需要指定步长即可。
  3. 具体代码请参考我们提供的ES demo
  4. scroll/scan每次返回的结果是shard Number \* size
  5. 官方demo：<https://www.elastic.co/guide/en/elasticsearch/client/java-api/2.1/java-search-scrolling.html>

**50.如何处理bigint/bigdecimal类型，尤其涉及到金钱的时候**

* 1. ES官方正在考虑支持（因为这事需要存储层支持）<https://github.com/elastic/elasticsearch/issues/17006>
  2. 咱们维护的版本只能是业务方自己处理：要么\*100转成int/long,要么就裸string

**51.ES的默认排序规则**

* 1. 在不指定任何sort的情况下，ES给出的结果仍然是稳定和有序的。因为此时ES使用了默认的排序规则
  2. 默认排序是按照ES的存储引擎（lucene）提供的对结果先打分，然后按照分数的高低排序的
  3. Lucene的打分算法：
     1. <https://lucene.apache.org/core/3_6_0/scoring.html>
  4. 如何规避打分
     1. 使用filter

**52.Fielddata的危害和doc value的解救**

* 1. 如果要对查询的结果按照某个特定字段排序或者使用了ES的聚合，那么Fielddata就出场了、
  2. 为了快速的获取排序字段的值以及为了快速的聚合某些字段的值，ES默认情况下会全量的加载该字段的全部值到内存。如果数据量大，那么ES数据节点的内存就非常吃紧，甚至OOM也时刻相伴
  3. 为了解决该问题，ES提供了doc\_value属性，如果对某字段设置了doc\_value,那么该字段被用来排序或者聚合的情况下，并不会加载到内存而是仍然从硬盘读取
  4. Doc\_value的属性对应的是ES存储引擎里的列式存储数据结构，索引读取非常快。根据ES的官方benchmark, 使用doc\_value的情况比使用fielddata仅仅慢7%左右
  5. 很多人要提说根据他们的测试，doc\_value慢的不止7%；在次并不对%多少做讨论，但是要明白，很多情况下，用doc\_value并不是时间换空间，而是用时间换生命。否则OOM了，再快有毛用~~~

**53.不要用post filter**

* 1. Post filter的提供是为了满足特定场景，99%的情况下都用不到
  2. Post filter的内部实现相当于数据库的扫表，因此非常非常慢。90%的性能问题都是由其引起

**54.ScriptException are disabled**

* 1. ES提供脚本执行引擎，为了安全，我们维护的ES集群默认都是关闭脚本引擎的，需要使用的需要在申请时说明情况

**55.IndexNotFoundException**

* 1. 查看并确认索引是否被正确创建或者被删除

**56.如何确定分片数**

* 1. ES是静态分片，一旦分片数在创建索引时确定那么后继不能修改
  2. 数据量在亿级别，8或者16分片够用，**分片数最好是2的n次方**。
  3. 如果后继数据量的增长超过创建索引的预期，那么需要创建新索引并重灌数据
  4. ES维护团队后继会提供重灌数据的服务
  5. 创建mapping是请自行制定分片数，否则创建的索引的分片数是ES的默认值。这其实并不符合需求。
  6. 副本数：一般设置为1

**57.ES维护团队后继会提供重灌数据的服务**

* 1. 如果业务特性决定能容忍一定的宕机时间，那么重灌数据对业务方是100%透明的
  2. 重灌数据的操作流程是：对现有数据做一个快照，然后从这个快照里拿数据。
  3. 如果在重灌数据的时间段只有新增数据的操作，那么可以参考这个链接做到100%无宕机无数据缺失：<https://www.elastic.co/guide/en/elasticsearch/guide/current/index-aliases.html>
  4. 如果在重灌数据的时间段有update和delete操作，那么需要业务方自己cover重灌数据这个时间段的数据变更

**58.Index属性：Analyzed VS not\_analyzed VS No**

* 1. Analyzed是使用分词，一段描述性的文字：“原产地广东佛山”就会被分词为“原产地”、“广东”、“佛山”，这样用户在查询“佛山”时就能找到这段话；如果不分词就找不到
  2. Not\_analyzed是不使用分词，一个电话号码：“010-62256225”不分词，那么用“62256225”查询不到，只能用“010-62256225”查询
  3. No是不索引，这样的内容不能被查询到
  4. 最后，要知道的一点：不管上述analyzer还是not\_analyzer, 经过这样的处理之后，这玩意就开始有个学名了：token

**59.MaxBytesLengthExceededException**

* 1. 接上述，token的最大长度是有限制的，经常会出现将一个html当做not\_analyzed的内存形成一个token，然后这样的token超过长度就抛出这个异常了。
  2. 如果抛出这个异常，要么这个字段不需要被索引（index:no），要么需要被分词(index: analyzed)

**60.Type VS New Index**

* 1. ES里提供了type，很多人以为type是物理表，一个type的数据是独立存储的；但是在ES内部并不是这样，type在ES内部仅仅是一个字段。所以在很多数据能分为独立index的情况下，不要放到一个index里用type去分。
  2. 只有嵌套类和父子类的情况下使用type才是合理的。

**61.\_source的意义**

* 1. \_source是ES提供的默认字段，会将一个document的json全部存储
  2. 现阶段，如果预估到后继有重新灌数据的场景，那么最好打开\_source并关闭所有其他字段的存储属性

**62.ES的update**

* 1. ES没有原地update的能力。所有的update都是标记删除老文档，然后重新insert一条新文档

Internally, Elasticsearch has marked the old document as deleted and added an entirely new document

Note, this operation still means full reindex of the document, it just removes some network roundtrips and reduces chances of version conflicts between the get and the index. The \_source field needs to be enabled for this feature to work.

actually Elasticsearch is following exactly the same process:

* + 1. Retrieve the JSON from the old document
    2. Change it
    3. Delete the old document
    4. Index a new document

**63.VersionConflictEngineException**

* 1. 并发update同一条document，ES内部采用了乐观并发的处理。并发修改的操作直到最后要提交是才加锁检查版本号，如果发现修改之前获取的版本号已经改变（即已经被人修改），那么会抛出这个异常。然后由用户觉得如何处理该异常

**64.中文分词的问题**

* 1. ES并不提供原生的中文分词的能力
  2. 有第三方的中文分词的插件，比如[ik](https://github.com/medcl/elasticsearch-analysis-ik)等
  3. Ik是个toy分词器，有严肃的分词需求的话，请在使用ES之前使用独立的分词器分好词后向ES写入
  4. 第三步分好词之后有两个方式写入ES：
     1. ES提供数组类型的数据结构，直接把分好的词放到数组
     2. 分好词之后用空格链接起来，然后使用ES的whitespace analyzer
  5. 查询的时候一样的处理方式：先把词在外围分好，然后构建term query，再在term query的基础上构建boolean query
  6. [查看中文分词后的效果](http://jpcloud.jd.com/pages/viewpage.action?pageId=16624064#id-常见申请mapping-查看ik中文分词效果)

**65.Query VS filter(2.X将filter合并到过滤语境中了)**

* 1. Filter的结果能被filter cache
  2. Filter的过程不参与lucene的打分
  3. {"range":{"orderCreateTime":{"from":"2016-09-20T17:29:35Z","to":null,"include\_lower":true,"include\_upper":true}}}  
     date range filter不要这么写：to":null， from":null, 因为cache不友好。

**66.ES是如何实现节点容灾的**

* 1. ES中的index，首先会进行分片，每一个分片数据一般都会有自己的副本数据，ES分配分片的策略会保证同一个分片数据和自己的副本不会分配到同一个节点上
  2. 当集群中的某一节点宕机后，ES的master在ping该节点时通过一定的策略会发现该节点不存活；此时，ES开启恢复过程，恢复的策略如下：
     1. 恢复的目标是保证集群中分片的副本数不变
     2. 如果宕机的节点上承载某分片的主分片，那么此时（恢复过程）会将该分片分配在其他节点上的某一副本提升为主分片（记住：同一分片和其副本总是不在同一节点上，那么此事是保证有对应的副本可供提升的）
     3. 根据1保证副本数不变，如果宕机的节点承载某分片的副本，那么ES会在其他非宕机节点上用主分片复制一个副本
     4. 整个过程不影响集群的读写功能；但是由于多了复制分片和迁移分片的过程，集群的读写性能受影响

**67.ES扩容/缩容对业务的影响**

* 1. 整个过程不影响集群的读写功能；但是由于多了复制分片和迁移分片的过程，集群的读写性能受影响

**68.ES的TTL的坑**

* 1. http://blog.csdn.net/fangletian1981/article/details/51644547   （感谢青龙团队方琪的分享）
  2. <https://github.com/elastic/elasticsearch/issues/18280>

**69.ES TTL的使用经验**

* 1. ES2.X对TTL已经是Deprecated, 在最新版本5.X里已经是remove了.如果使用ttl，将来不能升级。<https://www.elastic.co/guide/en/elasticsearch/reference/5.5/breaking_50_mapping_changes.html#_literal__timestamp_literal_and_literal__ttl_literal>
  2. 请尽量不要使用TTL

**70.ES TTL替换方案**

* 1. 如果有根据时间过期的需求，请在查询和删除量方面自行cover
  2. 查询方面可以构建一个时间的range query过滤掉失效的数据
  3. 定期运行数据删除的逻辑

**71.esm管理端看到的 disk free 0GB 有没有问题？**

* 1. es网关类型的节点，不占用磁盘空间，所以返回都是0GB

**72.ES的TPS是多少？**

* 1. ES的写入和查询是跟数据强相关的，我可以给你一个参考TPS，但是这个数字对你没有意义。你只能自己用自己真实的数据去测试
  2. ES的单机写入能力是几十K，然后可以认为是无限水平扩展的，也就是有N台机器，那么集群的写入能力就是N \* 几十K

**73. 2.1的客户端可以正常的操作1.7的es么？**

* 1. 当然不可以。为什么是当然？因为开源软件大版本升级不后向兼容已经是通识了
  2. 所以也就别问我2.X的client能否访问ES5.X了

**74.在index时对字段用ik分词，查询时需要指定分词器吗？**

* 1. 指定方式：mapping中加这一段："**analyzer":"ik"  ,  
     ES2.1.X参照：**[**https://github.com/medcl/elasticsearch-analysis-ik/tree/v1.7.0**](https://github.com/medcl/elasticsearch-analysis-ik/tree/v1.7.0)**ES1.7.X参照：**[**https://github.com/medcl/elasticsearch-analysis-ik/tree/v1.4.1**](https://github.com/medcl/elasticsearch-analysis-ik/tree/v1.4.1)
  2. 查询时如果还想用ik，那么就不需要了。
  3. 更完整的表述请看下面ES官方文档里的说明：
     1. Analyzers can be specified per-query, per-field or per-index. At index time, Elasticsearch will look for an analyzer in this order:
        1. The analyzer defined in the field mapping.
        2. An analyzer named default in the index settings
        3. The standard analyzer
     2. At query time, there are a few more layers
        1. The analyzer defined in a full-text query
           1. { "match\_phrase" : { "message" : { "query" : "this is a test", "analyzer" : "my\_analyzer" } } }
        2. The search\_analyzer defined in the field mapping
        3. The analyzer defined in the field mapping
        4. An analyzer named default\_search in the index settings
        5. An analyzer named default in the index settings
        6. The standard analyzer

**75.QueryParsingException  Caused by: org.elasticsearch.index.query.QueryParsingException: [index\_name] No field specified for term filter**

     查询中存在这样的语句  {"term":{"name":null}} ，以前的ES 1.7.6 支持这样，但是 ES 2.1.1 版不支持term字段的值为null。可以改为通过 exists 或者 missing 查询。

**76.NumberFormatException Caused by: java.lang.NumberFormatException: For input string: "null"**

     如果field定义为  "type": "double"，但是索引文档时，传递的是"null"，就会抛NumberFormatException。

**77.集群Index TPS,Search TPS与副本数的关系**

1. 集群写入文档TPS与副本数正相关，查询TPS与副本数无关。  
   举例：  
   1分钟向集群写入1000个文档：  
          当索引有0个副本：集群写入文档次数为1000，Index TPS为1000/60.  
          当索引有1个副本：集群写入文档次数为2000，Index TPS为2000/60.  
          当索引有2个副本：集群写入文档次数为3000，Index TPS为3000/60.

          1分钟查询1000次：  
                  当索引有0个副本：search TPS为 1000/60.  
                  当索引有1个副本：search TPS为 1000/60.  
                  当索引有2个副本：search TPS为 1000/60.

**78.IllegalArgumentException      Caused by: java.lang.IllegalArgumentException: Invalid format: ""**

    传递空字符串 ""，不满足格式要求

**79.SearchParseException**

SearchParseException  Caused by Caused by: org.elasticsearch.search.SearchParseException: [seller\_order\_20160223][0]: query[+ConstantScore(o2oOrderType:`N) +[venderId:71598](http://venderid:71598/) +[stationNo:10006391](http://stationno:10006391/) +ConstantScore(stationOrderStatus:`) +ConstantScore(printMark:`) +ConstantScore(pickMark:`)],from[0],size[12]: Parse Failure [No mapping found for [create] in order to sort on]

没有“ create ”这个field，但是使用它进行查询

**80.InvalidIndexNameException**

InvalidIndexNameException Exception [2016-07-29 02:59:50,978][DEBUG][action.admin.indices.create] [dm\_172.28.159.25:20002] [realtime\_order\*] failed to create:[realtime\_order\*] InvalidIndexNameException[Invalid index name [realtime\_order\*], must not contain the following characters [\, /, \*, ?, ", <, >, |, , ,]]   
查询的时候索引带有\, /, \*, ?, ", <, >, |, , ,] 等特殊符号

**81.获取文档数**

如果想要获取满足某一条件（获取全部可以用match\_all）的文档数，不需要使用scroll，分页等。  
HTTP API：  
"hits" 中有一个"total" 字段，表示的就是符合查询条件的所有文档数。  
JAVA API：  
searchResponse.getHits().getTotalHits()获取的就是符合查询条件的所有文档数。

**82. 在ES2.X 中 delete-by-query插件的使用方法**

见ES example <http://source.jd.com/app/es-example> 中的 DeleteByQuery 类

官方文档：<https://www.elastic.co/guide/en/elasticsearch/plugins/2.1/delete-by-query-usage.html>

注意：**非常不推荐依赖这个插件做删除操作——性能很差，对集群也有不良影响，推荐按日期去建索引——速度快、性能高。**

**83.MergeMappingException**

MergeMappingException failed to put mappings on indices [[vos-index]], type [product-type]:org.elasticsearch.index.mapper.MergeMappingException: Merge failed with failures {[mapper [category2Name] has different store\_term\_vector values, mapper [category2Name] has different  
  
因为先索引数据，然后再设置mapping，就有可能触发这种异常。先索引数据，ES会根据你的数据类型自动推测出一个mapping，你再尝试更新mapping的时候，就可能发生冲突。  
应该先创建索引，mapping，然后再索引数据。

**84.如何获取线上索引最新的mapping信息？**

集群监控-》索引-》more->找到你的索引后面有个元数据 就是

**85.ClassCastException**

Exception [2016-10-20 18:53:01,262][DEBUG][action.search.type ] [gw\_172.20.87.31:9262] All shards failed for phase: [query]:org.elasticsearch.transport.RemoteTransportException: [dm\_172.20.87.32:9263][inet[/172.20.87.32:9363]][[indices:data/read/search[phase/query](http://indicesdata/)]]  
Caused by Caused by: java.lang.ClassCastException: org.elasticsearch.index.fielddata.plain.PagedBytesIndexFieldData cannot be cast to org.elasticsearch.index.fielddata.IndexNumericFieldData  
{"error":"SearchPhaseExecutionException[Failed to execute phase [query], all shards failed; shardFailures {[B1IqBjcOSQmTrMgcnCMfGg][string\_test\_index2][0]: ClassCastException[org.elasticsearch.index.fielddata.plain.SortedSetDVOrdinalsIndexFieldData cannot be cast to org.elasticsearch.index.fielddata.IndexNumericFieldData]}{[B1IqBjcOSQmTrMgcnCMfGg][string\_test\_index2][4]: ClassCastException[org.elasticsearch.index.fielddata.plain.SortedSetDVOrdinalsIndexFieldData cannot be cast to org.elasticsearch.index.fielddata.IndexNumericFieldData]}]","status":500}   
原因：在string类型上执行数值计算。虽然可以向string类型写int数据 ，但是执行数值计算的话，肯定报错。修改mapping。

**86.需要ES安装包**

 官方下载，或从以下链接处下载：<http://source.jd.com/app/ESAgent/tree/master/end-delimiter/esagent-client/src/main/resources/bak>

**87. delete-by-query插件**

<http://source.jd.com/app/es-example/blob/master/end-delimiter/src/main/java/com/jd/es/example/deleteByQuery/DeleteByQuery.java>

不推荐依靠这个删

**88. IllegalStateException**

exception caught on transport layer [[id: 0x768964d0, /172.22.98.18:32281 => /172.28.80.12:40106]], closing connection  
java.lang.IllegalStateException: Message not fully read (request) for requestId [2187011], action [cluster/nodes/info], readerIndex [39] vs expected [57]; resetting  
at org.elasticsearch.transport.netty.MessageChannelHandler.messageReceived(MessageChannelHandler.[java:120](http://java:120/))  
at org.jboss.netty.channel.SimpleChannelUpstreamHandler.handleUpstream(SimpleChannelUpstreamHandler.java:70)  
at org.jboss.netty.channel.DefaultChannelPipeline.sendUpstream(DefaultChannelPipeline.[java:564](http://java:564/))

      客户端版本与ES版本不匹配，请保证版本一致。

**89.  ReceiveTimeoutTransportException**

 2016-11-29 10:20:19,104 [ 150931] INFO - org.elasticsearch.client.transport - [Dreaming Celestial] failed to get local cluster state for {#transport#-2}{172.28.80.30}{172.28.80.30:40001}, disconnecting...  
ReceiveTimeoutTransportException[[][172.28.80.30:40001][[cluster:monitor/state](http://clustermonitor/)] request\_id [25] timed out after [5001ms]]  
at org.elasticsearch.transport.TransportService$TimeoutHandler.run(TransportService.[java:645](http://java:645/))  
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.[java:1145](http://java:1145/))  
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.[java:615](http://java:615/))  
at java.lang.Thread.run(Thread.[java:745](http://java:745/)) 

端口配置错了。java客户端必须使用TCP端口，HTTP请求方式使用http端口，不要搞混了。

**90.  VerifyError**

Exception in thread "main" java.lang.VerifyError: (class: org/jboss/netty/channel/socket/nio/NioWorkerPool, method: newWorker signature: (Ljava/util/concurrent/Executor;)Lorg/jboss/netty/channel/socket/nio/AbstractNioWorker;) Wrong return type in function

 netty包冲突， 需要屏蔽低版本或者多余的jar包。ES 2.1.2版本java客户端需要的netty版本是 3.10.5

**91 docker 线程数太多，或者 非docker线程多。**

        第一：docker环境，Settings中设置transport.netty.worker\_count = docker核数\*2 ，默认是物理核的2倍  
        第二：保障自己客户端是单例  
        第三：保障Settings中设置 client.transport.sniff 属性为false

**92  我的数据需要多大的集群？**

        原则上原始数据是多大，就认为是存到ES是多少。若有一个副本，则翻倍。一般都有副本的。所以可以认为是2倍你原始数据大小。  
        具体占用多少，没办法确定，根据mapping结构不同，大小不同，特别\_source，\_all，是否分词等。  
        想根据已知mapping得知数据大小，在测试环境灌数据，确定。记得，测试完，尽快删除。测试环境也不要搞批量大数据，一小部分测试环境就好了，因测试环境资源有限。

**93   ES怎么查看数据**

<http://esm.jd.com/> 或test.esm.jd.com 中左边菜单，【集群管理】--【集群监控】，在页面右上角，先定位到要操作的集群，再在左边【ES插件】下选择【header】，进入页面后【数据浏览】页签

header插件显示未连接，有时候并不影响查询，直接切换到 基本查询、复合查询 即可

**94 怎么添加别名**

方式1、 单独为已存在的索引，添加别名  
     PUT {index}/\_alias/{name}  
     {index} 想要给哪个索引添加别名  
     {name} 别名的名字  
     例如我要给索引blog添加一个别名为blog\_w：  
     PUT blog/\_alias/blog\_w

方式2、 在创建索引时，添加别名，如下为blog索引添加2个别名blog\_w和blog\_r

**创建索引时，指定索引的别名**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | PUT blog  {     "mappings": {        "type": {           "properties": {              "year": {                 "type": "integer"              }           }        }     },     "aliases": {        "blog\_w": {},        "blog\_r": {}     }  } |

参考：<https://www.elastic.co/guide/en/elasticsearch/reference/5.1/indices-aliases.html#alias-adding>

**95  关于script**

为了安全，默认集群关闭script功能，若特殊情况，确实需求开启，发邮件给**ES团队的运维**([wangshixing@jd.com](mailto:wangshixing@jd.com) ，抄jes@jd.com)申请开启。  
当前，仅支持开启默认的groovy script，开启方法----配置中添加

script.inline: on  
script.indexed: on

注意：生效过程需重启。  
  
**通过script更新字段值**：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | UpdateRequest updateRequest = new UpdateRequest(INDEX, TYPE, batchInfo.getBatchId()+"");  updateRequest.routing(routing);  updateRequest.script(new Script("ctx.\_source.nowCount += " + batchInfo.getNowCount() + ";ctx.\_source.updateTime = '" + updateTime + "'"));  getClient().update(updateRequest).get();    //通过HTTP：    POST test/type1/1/\_update  {      "script" : {          "inline": "ctx.\_source.counter += params.count",          "lang": "painless",          "params" : {              "count" : 4          }      }  } |

注意：以上同时更新了2个字段的值，更新语句用英文分号分割，要放在一个Script中。

**96  ES transport client的connection问题**

构建一个ES transport client,一般来说，这个client会向这个集群的每一个节点构建13个connection

* recovery: 2 channels for the recovery of indexes
* bulk: 3 channels for low priority bulk based operations such as bulk indexing. Previously called low.
* reg: 6 channels for medium priority regular operations such as queries. Previous called med.
* state: 1 channel dedicated to state based operations such as changes to the cluster state. Previously called high.
* ping: 1 channel dedicated to pings between the instances for fault detection.

更详细的描述见这个链接：<https://www.elastic.co/blog/found-elasticsearch-networking>

**97 关于inner\_hits**

       参考：<https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-inner-hits.html>       注意：size默认返回top 3。

**98  throttling indexing: numMergesInFlight=10, maxNumMerges=9**

       es限制写入文档的速率。可能是写入太快了，es后端段合并赶不上你写入速度，请控制下写入速度。在ES 2.0.0之前，Elasticsearch以固定速率（默认为20 MB /秒）控制段合并的速率，以免段合并过快影响查询和写入。ES 2.0.0版本以后，Elasticsearch使用自适应合并IO限制，允许的IO速率会自动调节，所以您不需要触摸任何限制设置：只需使用默认设置.  
      可以通过indices.store.throttle.type: none来关闭段合并速率限制。但是即使这样设置不限制段进行合并的速率，断合并仍可能落后与你写文档的速度，导致throttling indexing。 ES这样做是为了保护索引健康，因为太多段会导致各种麻烦。  
      如果您正在使用spinning disks，您可以尝试将index.merge.scheduler.max\_thread\_count设置为1，因为spinning disks并发（concurrent ）进行段合并。  
      You should disable store throttling when using SSDs: the default is horribly low (20 MB/sec) and very easily leads to merging falling behind, and then index throttling, under heavy indexing. In ES 2.0 Lucene use adaptive IO throttle that sets itself according to merge backlog ...  
      After increasing "index.merge.scheduler.max\_thread\_count" the problem seems to be gone -- we did not see any more throttling being logged so far. <https://discuss.elastic.co/t/merge-thread-busy-even-when-refresh-interval-disabled-for-indexing/27657/8>  
      <https://www.elastic.co/blog/elasticsearch-performance-indexing-2-0>   
      <https://discuss.elastic.co/t/now-throttling-indexing/22650>

**99 SearchContextMissingException**

       查询上下文过期了。看是否是设置的query的timeout太短了，或者scroll查询传递了旧的scrollId。少量异常不影响使用可忽略。  
       注意：  
     1). context Id 每次使用最新的，不要用老的，可能能查到数据，但每个context Id 保留固定的时间，所以每次取数用最新的   
     2). scroll 的超时时间设置足够长，够处理一次响应的时间的1.5~2倍即可，当然不要过长 。注意，是scroll的超时，不是查询的超时，是2个概念，2个可以同时设置。  
     3). 应用里记得要try...catch 这样，搜不到结果也能看日常日志的原因  
     4). 找 ES团队处理，截图你 客户端请求那一块的代码，及截图在UMP上反应该方法的成功率的截图  
     5). 如果你的请求100%成功率，检查是否有其它忘记的业务模块在请求这个ES服务器，必要时可以找运维检查正在与该服务器的所有连接信息，从IP上排除有其它业务在请求   
     6). 在java环境中调用scroll有scan,scroll两种方式，scan需要取新的游标(服务端会报游标缺失，客户端无缺失)，scroll每次游标都是一样的.官方已经弃用scan方式调用（感谢Y事业部李福波提供）

**100  org.apache.hadoop.ipc.RemoteException.unwrapRemoteException**

       ES的HDFS备份功能有异常，影响的是集群的每日数据的HDFS备份(不是很紧急)。联系运维，确认对应的备份功能是否可用，按需修复备份功能，部分情况需要重启集群。  
       详细的原因，可查看ES的日志文件

**101  NotXContentException  not\_x\_content\_exception**

      when using a generic http client, make sure you set the content-type to application/json on POST requests.  
      这里有篇文章，可以参考下。  
      <http://blog.sina.com.cn/s/blog_65b8744b0102w694.html>  
      <https://stackoverflow.com/questions/35740531/elastic-search-not-x-content-exception>

**102 .ParserMinimalBase.\_reportError(ParserMinimalBase.**[**java:533**](http://java:533/)**)**

     检查查询语句的语法，一般JSON解析出错。

**103. Updating Child Mapping with new field fails with NullPointerException**

|  |
| --- |
| PUT my-index/\_mapping/child  {    "child" : {      "\_parent" : {"type" : "parent"},//ES 2.1.X 父子文档更新mapping有bug,需设置这个...      "properties" : {        "new\_field" : {"type" : "string"}      }    }  } |

**104  queryparser.classic.TokenMgrError:**

一般是查询语句里的问题。如在query  string中 / 和 / 之间的内容被当做正则的，正则的写法是有约束的....  参考  <https://stackoverflow.com/questions/17798300/lucene-queryparser-with-in-query-criteria>

**105 java.io.EOFException: Premature EOF: no length prefix available**

at org.apache.hadoop.hdfs.protocolPB.PBHelper.vintPrefixed(PBHelper.[java:2280](http://java:2280/))

影响范围：备份数据，不影响ES集群的读写。

原因：HDFS自动备份出问题了，可能当天的没有备份成功，若需要保留备份，需要运维人员执行一次HDFS备份。

路径：运营管理 – 集群维护 – 选择待操作的集群 ，进入 节点管理 – 顶部菜单“备份管理”  – 立即备份

**106 StreamCorruptedException**

       [2017-07-06 01:42:47,041][WARN ][transport.netty ] [d\_172.20.173.106:30002] exception caught on transport layer [[id: 0x32b37bc3, /172.22.214.21:59893 => /172.20.173.106:30102]], closing [connection:java.io.StreamCorruptedException](http://connectionjava.io.streamcorruptedexception/): invalid internal transport message format, got (0,5a,0,0)

  有可能端口连错了，把http端口和tcp端口搞混了。或者有人在扫描端口。不影响系统。一般可以忽略。

**107 esm管理端执行出现"index": "bad-request"  "reason": "no such index"**

       esm管理端执行GET请求，出现如下返回结果。可能是你请求的body内容过大，get方式请求会将请求内容转成URL参数，导致URL长度超过了浏览器的限制。将GET请求方式换成POST可以解决。

|  |
| --- |
| {    "error": {      "reason": "no such index",      "index\_uuid": "\_na\_",      "index": "bad-request",      "resource.type": "index\_or\_alias",      "type": "index\_not\_found\_exception",      "root\_cause": [        {          "reason": "no such index",          "index\_uuid": "\_na\_",          "index": "bad-request",          "resource.type": "index\_or\_alias",          "type": "index\_not\_found\_exception",          "resource.id": "bad-request"        }      ],      "resource.id": "bad-request"    },    "status": 404  } |

**108 有数据查询不到**

       常见问题： term查询的字段mapping是分词的、索引mapping的\_source=false, 需要特殊查询或无法返回。 咨询时，提供详细信息---集群、索引、字段，查询语句