

Elasticsearch 性能调优

作者: [QForever](#)

原文链接: <https://ld246.com/article/1573397903378>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>一、 修改当前的用户进程限制</p>

<p>登陆 Elasticsearch 的启动用户，用一下命令查看 ulimit -a

</p>

<p>上图的文件句柄 open files 为 65536，centos 系统默认的是 1024 个文件句柄。如果文件句柄完了，这就意味着操作系统会拒绝连接，意味着数据可能丢失，在 ElasticSearch 大量请求的情况下，1024 个文件句柄数量是不够的，可以改成 655360。</p>

<p>修改方法：</p>

临时修改，重启失效。

<p>ulimit -n 655360</p>

<ol start="2" >

永久修改，注销当前用户，重新登录。

<p>修改 vim /etc/security/limits.conf</p>

<p>soft nofile 655360</p>

<p>hard nofile 655360

</p>

<p>二、 关闭内存交换</p>

<p>内存交换到磁盘对服务器性能来说是致命的。</p>

<p>查看内存信息 free -h

</p>

<p>关闭内存交换</p>

<p>暂时禁用。如果我们想要在 Linux 服务器上暂时关闭，可以执行如下命令，但在服务器重启后失：
： swapoff -a</p>

<p>永久性关闭。我们可以修改 /etc/sysctl.conf，增加如下参数：</p>

<p>vm.swappiness = 1 //0-100，则表示越倾向于使用虚拟内存。</p>

<p>注意：Swappiness 设置为 1 比设置为 0 要好，因为在一些内核版本，Swappiness=0 会引发 O M 内存溢出。

</p>

<p>三、 机器内存的分配</p>

<p>官方给出了解决方案，把一半的内存分配给 Luence，另外的内存分配给 ElasticSearch，老年代新生代的内存比例为 2：1 是比较合适的，在 ElasticSearch 的配置文件 jvm.options 文件配置：</p>

<p>-Xms8g//配置堆初始化大小</p>

<p>-Xmx8g//配置堆的最大内存</p>

<p>-Xmn4g//配置新生代内存</p>

<p>确保 Xmx 和 Xms 的大小是相同的，其目的是为了能够在 Java 垃圾回收机制清理完堆区后不需

重新分隔计算堆区的大小而浪费资源，可以减轻伸缩堆大小带来的压力。 </p>

<p>四、 合理设置最小的主节点</p>

<p>minimum_master_nodes 设置及其重要，为了防止集群脑裂，这个参数应该设置为法定个数就： </p>

<p>(master 候选节点个数/ 2)+1。 </p>

<p>五、 剔除掉高负载的 Master 竞选</p>

<p>将监控到的高负载节点磁盘 IO 高的，在 elasticsearch.yml 中设置 node.master=false，然后启 elasticsearch，负载下降了写。 </p>

<p>六、 角色隔离</p>

<p>ES 集群中的数据节点负责对数据进行增、删、改、查和聚合等操作，所以对 CPU、内存和 I/O 消耗很大。在搭建 ES 集群时，我们应该对 ES 集群中的节点进行角色划分和隔离。 </p>

<p>候选主节点： </p>

<p>node.master=true</p>

<p>node.data=false</p>

<p>数据节点： </p>

<p>node.master=false</p>

<p>node.data=true

 </p>

<p>七、 修改 vm.max_map_count</p>

<p>vim /etc/sysctl.conf</p>

<p>vm.max_map_count = 655360</p>

<p>八、 定时清理过期数据</p>

<p>根据业务需求，定时清理过期的数据，可以使用 crontab 定时执行脚本实现。 </p>

<p>九、 其它优化</p>

<p>集群分片优化设置，索引优化，查询优化，索引时间精度优化。 </p>

<p>十、 elasticsearch 配置文件说明</p>

<p>cluster.name: elasticsearch: 配置 ES 的集群名称，默认值是 ES，建议改成与所存数据相关名称，ES 会自动发现在同一网段下的集群名称相同的节点。 </p>

<p>node.name: "node1": 集群中的节点名，在同一个集群中不能重复。节点的名称一旦设置，就能再改变了。当然，也可以设置成服务器的主机名称，例如 node.name:\${HOSTNAME}。 </p>

<p>node.master: true: 指定该节点是否有资格被选举成为 Master 节点，默认是 True，如果被设为 True，则只是有资格成为 Master 节点，具体能否成为 Master 节点，需要通过选举产生。 </p>

<p>node.data: true: 指定该节点是否存储索引数据，默认为 True。数据的增、删、改、查都是在 data 节点完成的。 </p>

<p>index.number_of_shards: 5: 设置索引分片个数，默认是 5 片。也可以在创建索引时设置值，具体设置为多大都要根据数据量的大小来定。如果数据量不大，则设置成 1 时效率最高。 </p>

<p>index.number_of_replicas: 1: 设置默认的索引副本个数，默认为 1 个。副本数越多，集群的可用性越好，但是写索引时需要同步的数据越多。 </p>

<p>path.conf: /path/to/conf: 设置配置文件的存储路径，默认是 ES 目录下的 Conf 文件夹。建议使用默认值。 </p>

<p>path.data: /path/to/data1,/path/to/data2: 设置索引数据多存储路径，默认是 ES 根目录下的 Data 文件夹。切记不要使用默认值，因为若 ES 进行了升级，则有可能数据全部丢失。可以用半角逗隔开设的多个存储路径，在多硬盘的服务器上设置多个存储路径是很有必要的。 </p>

<p>path.logs: /path/to/logs: 设置日志文件的存储路径，默认是 ES 根目录下的 Logs，建议修改其他地方。 </p>

<p>path.plugins: /path/to/plugins: 设置第三方插件的存放路径，默认是 ES 根目录下的 Plugins 文件夹。 </p>

<p>bootstrap.mlockall: true: 设置为 True 时可锁住内存。因为当 JVM 开始 Swap 时，ES 的效率会降低，所以要保证它不 Swap。 </p>

<p>network.bind_host: 192.168.0.1: 设置本节点绑定的 IP 地址，IP 地址类型是 IPv4 或 IPv6，认为 0.0.0.0。 </p>

<p>network.publish_host: 192.168.0.1: 设置其他节点和该节点交互的 IP 地址，如果不设置，则进行自我判断。 </p>

<p>network.host: 192.168.0.1: 用于同时设置 bind_host 和 publish_host 这两个参数。</p>
<p>http.port: 9200: 设置对外服务的 HTTP 端口，默认为 9200。ES 的节点需要配置两个端口号，一个对外提供服务的端口号，一个是集群内部使用的端口号。http.port 设置的是对外提供服务的端口号。注意，如果在一个服务器上配置多个节点，则切记对端口号进行区分。</p>
<p>transport.tcp.port: 9300: 设置集群内部的节点间交互的 TCP 端口，默认是 9300。注意，如在一个服务器配置多个节点，则切记对端口号进行区分。</p>
<p>transport.tcp.compress: true: 设置在节点间传输数据时是否压缩，默认为 False，不压缩。<p>
<p>discovery.zen.minimum_master_nodes: 1: 设置在选举 Master 节点时需要参与的最少的候选节点数，默认为 1。如果使用默认值，则当网络不稳定时可能会出现脑裂。合理的数值为 $(\text{master_eligible_nodes}/2)+1$ ，其中 master_eligible_nodes 表示集群中的候选主节点数。</p>
<p>discovery.zen.ping.timeout: 3s: 设置在集群中自动发现其他节点时 Ping 连接的超时时间，认为 3 秒。在较差的网络环境下需要设置得大一点，防止因误判该节点的存活状态而导致分片的转移</p>