集群管理

1. 集群配置：
   1. 节点：
      1. 相同角色的节点，避免使用差异较大的服务器配置，
      2. 避免使用“超大杯”服务器（SS：Super Server），比如128核CPU，1 T的内存，2T的固态硬盘。这样可能会产生较大的资源浪费。
      3. 等量的配置，使用较少的物理机好于使用较多的虚拟机。比如一个一个五台4核16G的物理机，好于10甚至11台2核8G的虚拟机，这里不仅仅是虚拟机本身可能也会消耗一部分性能的问题，也涉及数据安全的问题。
      4. 避免在同一台服务器上部署多个节点，会增加集群管理的难度。
   2. 内存：根据业务量不同，内存的需求也不同，一般生产建议不要少于16G。ES是比较依赖内存的，并且对内存的消耗也很大，内存对ES的重要性甚至是高于CPU的，所以即使是数据量不大的业务，为了保证服务的稳定性，在满足业务需求的前提下，我们仍需考虑留有不少于20%的冗余性能。一般来说，按照百万级、千万级、亿级数据的索引，我们为每个节点分配的内存为16G/32G/64G就足够了，太大的内存，性价比就不是那么高了。
   3. 磁盘：对于ES来说，磁盘可能是最重要的了，因为数据都是存储在磁盘上的，当然这里说的磁盘指的是磁盘的性能。磁盘性能往往是硬件性能的瓶颈，木桶效应中的最短板。ES应用可能要面临不间断的大量的数据读取和写入。生产环境可以考虑把节点冷热分离，“热节点”使用SSD做存储，可以大幅提高系统性能；冷数据存储在机械硬盘中，降低成本。另外，关于磁盘阵列，可以使用raid 0。
   4. CPU：CPU对计算机而言可谓是最重要的硬件，但对于ES来说，可能不是他最依赖的配置，因为提升CPU配置可能不会像提升磁盘或者内存配置带来的性能收益更直接、显著。当然也不是说CPU的性能就不重要，只不过是说，在硬件成本预算一定的前提下，应该把更多的预算花在磁盘以及内存上面。通常来说单节点cpu 4核起步，不同角色的节点对CPU的要求也不同。服务器的CPU不需要太高的单核性能，更多的核心数和线程数意味着更高的并发处理能力。现在PC的配置8核都已经普及了，更不用说服务器了。
   5. 网络： ES是天生自带分布式属性的，并且ES的分布式系统是基于对等网络的，节点与节点之间的通信十分的频繁，延迟对于ES的用户体验是致命的，所以对于ES来说，低延迟的网络是非常有必要的。因此，使用扩地域的多个数据中心的方案是非常不可取的，ES可以容忍集群夸多个机房，可以有多个内网环境，支持跨AZ部署，但是不能接受多个机房跨地域构建集群，一旦发生了网络故障，集群可能直接GG，即使能够保证服务正常运行，维护这样（跨地域单个集群）的集群带来的额外成本可能远小于它带来的额外收益。
   6. 集群规划：没有最好的配置，只有最合适的配置。
      1. 在集群搭建之前，首先你要搞清楚，你ES cluster的使用目的是什么？主要应用于哪些场景，比如是用来存储事务日志，或者是站内搜索，或者是用于数据的聚合分析。针对不同的应用场景，应该指定不同的优化方案。
      2. 集群需要多少种配置（内存型/IO型/运算型），每种配置需要多少数量，通常需要和产品运营和运维测试商定，是业务量和服务器的承载能力而定，并留有一定的余量。
      3. 一个合理的ES集群配置应不少于5台服务器，避免脑裂时无法选举出新的Master节点的情况，另外可能还需要一些其他的单独的节点，比如ELK系统中的Kibana、Logstash等。
2. 集群搭建：
   1. 过程参见0701：ES集群

curl -H "Content-Type: application/json" -XPOST http://172.16.10.184:9200/sys-tcp-2020.07.18/\_search?pretty -d ' {"query": {"match\_all": {}}}'

* 1. 重要的节点配置：
     1. cluster.name: 配置集群的名称，集群发现是通过集群名称来识别不同集群的。很重要。
     2. node.name: 节点名称
     3. path.data：数据的存储路径，必须修改为和es安装目录不同的位置，如果同一台服务器部署了多个节点，那么多个节点的路径必须不同。
     4. path.logs：日志的存储路径，规则同上。
     5. bootstrap.memory\_lock：禁用swap，只要记住，想要es快，这个一定要打开，想弄明白看0701，或者查阅文档。
     6. network.host：给当前节点绑定的ip地址
     7. http.port：服务端口号，默认9200，取值范围[9200-9300)
     8. transport.port：通信端口号，节点之间联络传输数据的端口，取值范围[9300,9400)。
     9. discovery.seed\_hosts：master候选节点的列表，注意不是及群众所有的节点，通常为数组，比如（IPV4）：["172.16.10.184:9300", "172.16.10.185:9300"]，数组中可以使用节点名称代替IP地址，不过建议使用IP地址。
     10. cluster.initial\_master\_nodes：初始化集群的master节点，就是集群首次启动的时候指定谁为默认的master。
     11. http.cors.enabled：开启跨域支持
     12. http.cors.allow-origin：设置哪些地址可以跨域，\*代表允许任何地址跨域。
  2. 其他配置：es的默认配置是一个非常合理的默认配置，绝大多数情况下是不需要修改的，如果不理解某项配置的含义，没有经过验证就贸然修改默认配置，可能造成严重的后果。比如max\_result\_window这个设置，默认值是1W，这个设置是分页数据每页最大返回的数据量，冒然修改为较大值会导致OOM。ES没有银弹，不可能通过修改某个配置从而大幅提升ES的性能，通常出厂配置里大部分设置已经是最优配置，只有少数和具体的业务相关的设置，事先无法给出最好的默认配置，这些可能是需要我们手动去设置的。关于配置文件，如果你做不到彻底明白配置的含义，不要随意修改。
  3. JVM：
     1. jvm heap分配：7.6版本默认1GB，这个值太小，很容易导致OOM。Jvm heap大小不要超过物理内存的50%，最大也不要超过32GB（compressed oop），它可用于其内部缓存的内存就越多，但可供操作系统用于文件系统缓存的内存就越少，heap过大会导致GC时间过长。修改jvm heap有两种方式：
        1. 在ES进程启动的时候加上参数：ES\_JAVA\_OPTS="-Xms512m -Xmx512m",或在jvm.options中修改-Xms和-Xmx，注意最大值和最小值要相同，避免jvm heap在运行中resize，这是一个非常耗性能的过程。
        2. 修改环境变量：ES\_HEAP\_SIZE，环境变量的优先级高于jvm.options中的设置的数值。
     2. 通常来说JVM其他配置是不需要修改的