

## 【技术贴】如何部署Apache Kylin集群实现负载均衡？

2017-03-08 apachekylin

点击上方蓝色 [apachekylin](#) 可以关注我哟

// Apache Kylin作为一个开源的OLAP-on-Hadoop引擎，由于亚秒级的强大查询能力被广大用户熟知。超高的查询性能也决定了Apache Kylin具有强大的高并发优势，同时支持良好的线性扩展。本文将介绍如何部署一个Apache Kylin集群，以及通过负载均衡提高查询并发度。

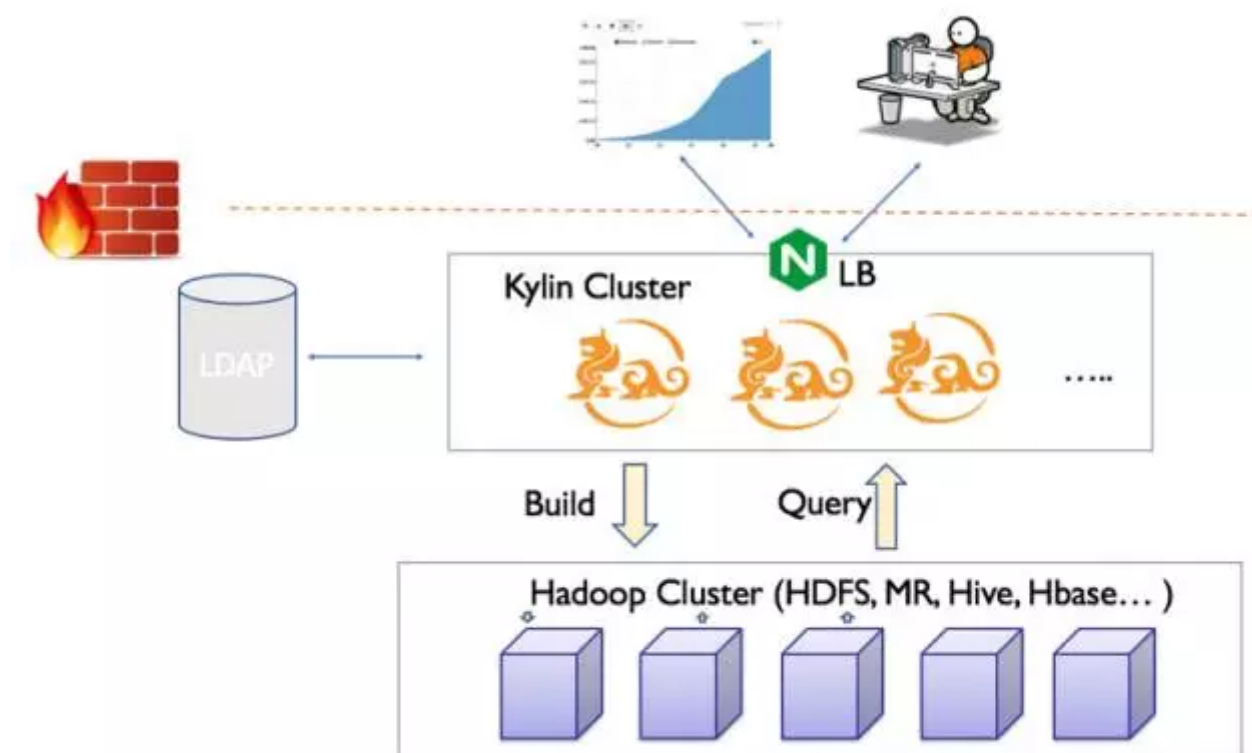
感谢Kyligence技术合伙人&高级软件架构师，KyBot技术负责人，Apache Kylin核心开发者和项目管理委员会成员（PMC），李栋撰文。



Apache Kylin™是一个开源的分布式分析引擎，提供Hadoop之上的SQL查询接口及多维分析（OLAP）能力以支持超大规模数据，能在亚秒内查询巨大的Hive表。

有关Apache Kylin的性能及并发性数据，社区上已经有很多介绍的文章，感兴趣的同学可以到Apache Kylin的官网（<http://kylin.apache.org>）进行查找。

下图所示是Apache Kylin集群部署的架构图，本文将主要以Apache Kylin 1.6为例，介绍集群部署的详细步骤。



Apache Kylin是一个无状态的服务，所有的状态信息都保存在HBase中。因此，只要把多个Kylin实例部署在同一个HBase集群上，就可以对这些 Kylin实例进行负载均衡。因此，**部署的多个Kylin实例需要满足以下条件：**

共享同一个Hadoop集群和HBase集群；

无端口冲突，最好能分别部署在不同服务器上，以做到互不影响；

共享同一个元数据库，即kylin.properties中kylin.metadata.url值相同；

只有一个Kylin实例运行任务引擎（即kylin.server.mode=all），其它 Kylin实例都作为查询引擎（即kylin.server.mode=query）

为了将外部的查询请求转发给Apache Kylin集群中的单个节点，需要部署一个负载均衡器（Load Balancer），如Nginx等。负载均衡器通过一定策略对请求进行分发，并在节点失效时重试其他节点。Kylin用户直接可以通过负载均衡器的地址进行访问。

以Nginx为例，需要为Apache Kylin站点新建一个配置文件（如kylin.conf），内容如下：

```
upstream kylin {  
    server 127.0.0.1:7070; #Kylin Server 1  
    server 127.0.0.1:17070; # Kylin Server 2  
}  
server {  
    listen    18080;  
    location / {  
        proxy_pass http://kylin;  
    }  
}
```

默认情况下，Nginx是以轮询的方式进行负载均衡，即每个请求按时间顺序逐一分配到不同的Apache Kylin实例，如果一个实例失效，会自动将其剔除。但是，默认情况下，Apache Kylin的用户Session信息是保存在本地的，当同一个用户的多个请求发送给不同Apache Kylin实例时，并不是所有的实例都能识别用户的登陆信息。因此，可以简单地配置Nginx使用ip\_hash方式，使每个请求按照客户端ip的hash结果固定地访问一个Kylin实例。

但是ip\_hash的方式可能导致Kylin实例的负载不平衡，特别是只有少量应用服务器频繁访问Kylin时会导致大部分查询请求分发给个别Kylin实例。为解决这个问题，可以通过配置Kylin将Session信息保存到Redis集群中（或MySQL、MemCache等），实现多个Kylin实例的Session共享。这样，Nginx轮询方式就不会出现Session丢失的问题了。

Apache Kylin使用Apache Tomcat作为Web服务器，简单修改Tomcat配置文件即可完成配置。  
**具体配置步骤如下：**

1. 下载Redis相关的Jar包，并放置在\$KYLIN\_HOME/tomcat/lib目录下：  
wget http://central.maven.org/maven2/redis/clients/jedis/2.0.0/jedis-2.0.0.jar  
wget http://central.maven.org/maven2/org/apache/commons/commons-pool2/2.2/commons-pool2-2.2.jar  
wget https://github.com/downloads/jcoleman/tomcat-redis-session-manager/tomcat-redis-session-manager-1.2-tomcat-7-java-7.jar
2. 修改\$KYLIN\_HOME/tomcat/context.xml，增加如下项目：  
<Valve className="com.radiadesign.catalina.session.RedisSessionHandlerValve" />  
<Manager className="com.radiadesign.catalina.session.RedisSessionManager"  
host="localhost" port="6379" database="0" maxInactiveInterval="60"/>  
其中，host和port指向所使用的Redis集群地址。

修改完毕之后，重启Kylin服务。再次访问负载均衡器，即可发现Session问题已经解决了。

综上所述，用户只需要通过简单的配置就可以实现Apache Kylin的集群部署和负载均衡，加上秒级甚至亚秒级的查询速度，在高并发高性能的应用场景依然能够带来良好的用户体验。

您可能还会想看

【技术贴】Apache Kylin v2.0.0 Beta尝鲜版上线！！

【福利帖】《Apache Kylin权威指南》正式发售

【技术贴】揭秘Apache Kylin V2.0新特性：字典编码模块的优化  
如何参与到Apache开源社区（以Apache Kylin为例）

【技术贴】Apache Kylin优化利器KyBot：降低Cube膨胀倍数

---

"Apache and Apache Kylin are either registered trademarks or trademarks of The Apache Software Foundation in the US and/or other countries. No endorsement by The Apache Software Foundation is implied by the use of these marks."

---



长按图片识别二维码关注  
Apache Kylin官方公众号



点击“阅读原文”进入Apache Kylin官网查看更多技术文章

阅读原文