(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111597536 A
(43)申请公布日 2020.08.28

(21)申请号 202010426597.1

(22)申请日 2020.05.19

(71)申请人 重庆第二师范学院
地址 400000 重庆市南岸区学府大道9号

(72)发明人 彭亚飞 韦鹏程 付仕明

(74)专利代理机构 北京国坤专利代理事务所（普通合伙）11491
代理人 赵红霞

(51)Int.Cl.
G06F 21/33(2013.01)
G06F 11/14(2006.01)

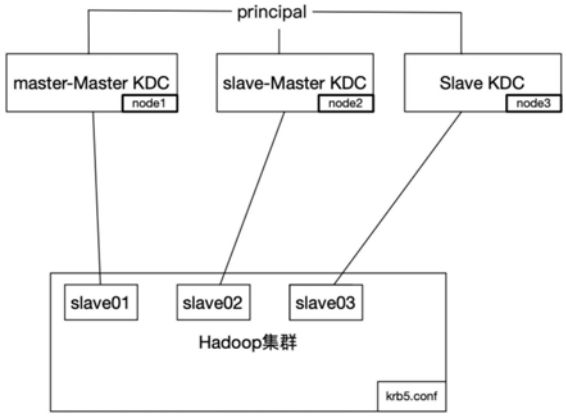
权利要求书2页 说明书4页 附图3页

(54)发明名称

一种hadoop集群kerberos高可用认证方法

(57)摘要

本发明公开了一种hadoop集群kerberos高可用认证方法，在不同的Linux服务器部署多个kerberos server；Linux定时器定时调用shell脚本对多个kerberos server的数据进行同步与备份；Hadoop在启动kerberos认证时，在krb5.conf的kdc参数配置所使用的所有kdc服务器地址；Hadoop集群中，节点通过调整kdc参数顺序对kerberos认证优先顺序进行调整，kerberos server实现多备份，当hadoop在做kerberos认证时，可以根据其认证服务器配置做调整，在大规模集群里面，可以减少因为认证带来的延时；同时，降低了kerberos server单服务器的负载，保障了集群做kerberos认证的稳定性。



CN 111597536 A

1.一种hadoop集群kerberos高可用认证方法,其特征是,包括以下步骤:

S1:在不同的Linux服务器部署多个kerberos server;

S2:Linux定时器定时调用shell脚本对相应Linux服务器内的多个kerberos server的数据相互之间进行同步与备份;

S3:Hadoop在启动kerberos认证时,在krb5.conf的kdc参数配置所使用的所有kdc服务器地址;其中,默认第一个kdc参数为kerberos server的主主(master-master)服务,第二个kdc参数为kerberos server的备主(slave-master)服务,第三个kdc参数为slave服务;Hadoop集群中,节点通过调整kdc参数顺序对kerberos认证优先顺序进行调整。

2.根据权利要求1所述的一种hadoop集群kerberos高可用认证方法,其特征是,多个所述kerberos server部署具体为:在Linux服务器安装kerberos server与kerbreos client;所述kerbreos client用于操作本地与远程的kerberos server服务principle;所述kerberos server是Hadoop集群票据分发与验证服务器。

3.根据权利要求1所述的一种hadoop集群kerberos高可用认证方法,其特征是,所述Linux定时器定时周期为5s执行一次。

4.根据权利要求1所述的一种hadoop集群kerberos高可用认证方法,所述Hadoop集群做kerberos认证时,优先向master-master对应的node1认证;当node1的kerberos server出现异常时,接着向slave-master对应的node2认证;当node1、node2均出现异常时,向slave对应的node3认证。

5.根据权利要求4所述的一种hadoop集群kerberos高可用认证方法,其特征是,所述数据的同步与备份具体为:node1、node2、node3相互备份数据;其中,node1、node2相互之间数据同步,node3全量备份master-master主机的数据。

6.根据权利要求5所述的一种hadoop集群kerberos高可用认证方法,其特征是,所述slave节点数据备份具体为:slave节点负责将master-master节点的kdc数据同步过来;同时将相应数据导出,做冷备份。

7.根据权利要求5所述的一种hadoop集群kerberos高可用认证方法,所述master-master与slave-master相互数据同步具体为:

a1:获取Linux系统hadoop启用kerberos的配置文件krb5.conf;

a2:获取本系统kerberos高可用配置目录;

a3:利用shell从krb5.conf下获取主kerberos server的Linux主机名、备kerberos server的Linux主机名;

a4:获取本Linux系统的主机名;

a5:获取本地kerberos server数据库里面所有principle信息,命名为local_principles;

a6:根据krb5.conf的kdc参数配置,获取remote主机名以及remote主机的所有principle信息,命名为remote_principles;

a7:遍历local_principals;若local_principal不在remote_principles,则同步local_principal到remote kdc服务器中;反之,则根据update_time,version信息,判断local_principal与remote_principal哪个为最新数据,然后同步到服务中;

a8:当将local_principals都遍历结束后;判断本地kdc服务是否running,如果

running,本次同步结束;如果非running,则启动起来,然后将krb5.conf中的kdc参数对调,同步到几个节点,最后结束。

一种hadoop集群kerberos高可用认证方法

技术领域

[0001] 本发明涉及计算机软件应用技术领域，更具体地说，它涉及一种hadoop集群kerberos高可用认证方法。

背景技术

[0002] Hadoop是一个由Apache基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。

[0003] Kerberos是一种网络认证协议，其设计目标是通过密匙系统为客户机/服务器应用程序提供强大的认证服务。该认证过程的实现不依赖于主机操作系统的认证，无需几本主机地址的信任，不要求网络上所有主机的物理安全，并假定网络上传输的数据包可以被任意的读取修改和插入。

[0004] Hadoop集群使用kerberos认证，是为了集群的安全性，Kerberos可以将认证的密钥在集群部署时事先放到可靠的节点上。集群运行时，集群内的节点使用密钥得到认证，只有被认证过节点才能正常使用，企图冒充的节点由于没有事先得到的密钥信息，无法与集群内部的节点通信。防止了恶意的使用或篡改Hadoop集群的问题，确保了Hadoop集群的可靠安全。

[0005] 然而，在大规模集群中，单个kerberos服务作为认证服务器，往往会出现问题：首先，有单点故障问题，当这台kerberos认证服务器出现异常，则集群上的任务启动认证会失败，会导致整个集群异常；其次，当Hadoop集群节点达到一定规模，系统任务达到一定规模的时候，会出现单kerberos认证服务器认证延迟过高或者认证异常等情况，所以，单kerberos认证服务器已无法满足大规模集群认证。因此，如何研究设计一种hadoop集群kerberos高可用认证方法是我们目前迫切需要解决的问题。

发明内容

[0006] 本发明的目的是提供一种hadoop集群kerberos高可用认证方法，kerberos server实现多备份，当hadoop在做kerberos认证时，可以根据其认证服务器配置做调整，在大规模集群里面，可以减少因为认证带来的延时；同时，降低了kerberos server单服务器的负载，保障了集群做kerberos认证的稳定性。

[0007] 本发明的上述技术目的是通过以下技术方案得以实现的：一种hadoop集群kerberos高可用认证方法，包括以下步骤：

[0008] S1：在不同的Linux服务器部署多个kerberos server；

[0009] S2：Linux定时器定时调用shell脚本对相应Linux服务器内的多个kerberos server的数据相互之间进行同步与备份；

[0010] S3：Hadoop在启动kerberos认证时，在krb5.conf的kdc参数配置所使用的所有kdc服务器地址；其中，默认第一个kdc参数为kerberos server的主主(master-master)服务，第二个kdc参数为kerberos server的备主(slave-master)服务，第三个kdc参数为slave服

务;Hadoop集群中,节点通过调整kdc参数顺序对kerberos认证优先顺序进行调整。

[0011] 优选的,多个所述kerberos server部署具体为:在Linux服务器安装kerberos server与kerbreos client;所述kerbreos client用于操作本地与远程的kerberos server服务principle;所述kerberos server是Hadoop集群票据分发与验证服务器。

[0012] 优选的,所述Linux定时器定时周期为5s执行一次。

[0013] 优选的,所述Hadoop集群做kerberos认证时,优先向master-master对应的node1认证;当node1的kerberos server出现异常时,接着向slave-master对应的node2认证;当node1、node2均出现异常时,向slave对应的node3认证。

[0014] 优选的,所述数据的同步与备份具体为:node1、node2、node3相互备份数据;其中,node1、node2相互之间数据同步,node3全量备份master-master主机的数据。

[0015] 优选的,所述slave节点数据备份具体为:slave节点负责将master-master节点的kdc数据同步过来;同时将相应数据导出,做冷备份。

[0016] 优选的,所述master-master与slave-master相互数据同步具体为:

[0017] a1:获取Linux系统hadoop启用kerberos的配置文件krb5.conf;

[0018] a2:获取本系统kerberos高可用配置目录;

[0019] a3:利用shell从krb5.conf下获取主kerberos server的Linux主机名、备kerberos server的Linux主机名;

[0020] a4:获取本Linux系统的主机名;

[0021] a5:获取本地kerberos server数据库里面所有principle信息,命名为local_principles;

[0022] a6:根据krb5.conf的kdc参数配置,获取remote主机名以及remote主机的所有principle信息,命名为remote_principles;

[0023] a7:遍历local_principals;若local_principal不在remote_principles,则同步local_principal到remote kdc服务器中;反之,则根据update_time,version信息,判断local_principal与remote_principal哪个为最新数据,然后同步到服务中;

[0024] a8:当将local_principals都遍历结束后;判断本地kdc服务是否running,如果running,本次同步结束;如果非running,则启动起来,然后将krb5.conf中的kdc参数对调,同步到几个节点,最后结束。

[0025] 与现有技术相比,本发明具有以下有益效果:

[0026] 1、通过hadoop集群部署方案,在启动kerberos认证过程中,krb5.conf配置文件根据集群的节点交叉部署kdc参数,能避免kerberos认证负载过大,同时能满足集群的kerberos认证的高可用要求;

[0027] 2、通过Kerberos同步方案,kerberos server之间数据同步通过shell调用,其kdc同步过程是根据principal的导入信息、版本号做对比进行同步;同时,根据实际场景,实现多kerberos扩充,保障集群kerberos认证的稳定性。

附图说明

[0028] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些

实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0029] 图1是本发明实施例中的整体原理结构示意图;

[0030] 图2是本发明实施例中master-master与slave-master相互数据同步的流程图;

[0031] 图3是本发明实施例中slave节点数据备份的流程图。

具体实施方式

[0032] 为了使本发明所要解决的技术问题、技术方案及有益效果更加清楚明白,以下结合附图1-3及实施例,对本发明进行进一步详细说明。

[0033] Shell是系统的用户界面,提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行。

[0034] Shell定时器是Linux服务器定时去允许一个脚本来触发一个操作。

[0035] 实施例:一种hadoop集群kerberos高可用认证方法,如图1所示,包括以下步骤:

[0036] S1:在不同的Linux服务器部署多个kerberos server。

[0037] S2:Linux定时器定时调用shell脚本对相应Linux服务器内的多个kerberos server的数据相互之间进行同步与备份。Linux定时器定时周期为5s执行一次

[0038] S3:Hadoop在启动kerberos认证时,在krb5.conf的kdc参数配置所使用的所有kdc服务器地址;其中,默认第一个kdc参数为kerberos server的主主(master-master)服务,第二个kdc参数为kerberos server的备主(slave-master)服务,第三个kdc参数为slave服务;Hadoop集群中,节点通过调整kdc参数顺序对kerberos认证优先顺序进行调整。

[0039] 多个kerberos server部署具体为:在Linux服务器安装kerberos server与kerbreos client。kerbreos client用于操作本地与远程的kerberos server服务principle。kerberos server是Hadoop集群票据分发与验证服务器。

[0040] 本发明以三个kerberos server备份方案及集群只有3个节点的情况进行详细说明。其实,真实情况下,企业级应用hadoop集群节会有100甚至更多,这个时候,kerberos server节点数系统是可以做扩充的,扩充kerberos server个数,在备份的时候,按照该发明,实现更多的kerberos备份。那么,kerberos认证的稳定性就更高,同时,规划好hadoop集群节点的krb5.conf配置的kdc参数顺序,能保障集群的kerberos认证效率更高。

[0041] 如图1所示,Hadoop集群做kerberos认证时,优先向master-master对应的node1认证;当node1的kerberos server出现异常时,接着向slave-master对应的node2认证;当node1、node2均出现异常时,向slave对应的node3认证。

[0042] 如图1所示,数据的同步与备份具体为:node1、node2、node3相互备份数据;其中,node1、node2相互之间数据同步,node3全量备份master-master主机的数据。

[0043] 如图3所示,slave节点数据备份具体为:slave节点负责将master-master节点的kdc数据同步过来;同时将相应数据导出,做冷备份。

[0044] 如图2所示,master-master与slave-master相互数据同步具体为:

[0045] a1:获取Linux系统hadoop启用kerberos的配置文件krb5.conf;

[0046] a2:获取本系统kerberos高可用配置目录;

[0047] a3:利用shell从krb5.conf下获取主kerberos server的Linux主机名、备

kerberos server的Linux主机名;

[0048] a4:获取本Linux系统的主机名;

[0049] a5:获取本地kerberos server数据库里面所有principle信息,命名为local_principles;

[0050] a6:根据krb5.conf的kdc参数配置,获取remote主机名以及remote主机的所有principle信息,命名为remote_principles;

[0051] a7:遍历local_principals;若local_principal不在remote_principles,则同步local_principal到remote kdc服务器中;反之,则根据update_time,version信息,判断local_principal与remote_principal哪个为最新数据,然后同步到服务中;

[0052] a8:当将local_principals都遍历结束后;判断本地kdc服务是否running,如果running,本次同步结束;如果非running,则启动起来,然后将krb5.conf中的kdc参数对调,同步到几个节点,最后结束。

[0053] 工作原理:通过hadoop集群部署方案,在启动kerberos认证过程中,krb5.conf配置文件根据集群的节点交叉部署kdc参数,能避免kerberos认证负载过大,同时能满足集群的kerberos认证的高可用要求;通过Kerberos同步方案,kerberos server之间数据同步通过shell调用,其kdc同步过程是根据principal的导入信息、版本号做对比进行同步;同时,根据实际场景,实现多kerberos扩充,保障集群kerberos认证的稳定性。

[0054] 本具体实施例仅仅是对本发明的解释,其并不是对本发明的限制,本领域技术人员在阅读完本说明书后可以根据需要对本实施例做出没有创造性贡献的修改,但只要在本发明的权利要求范围内都受到专利法的保护。

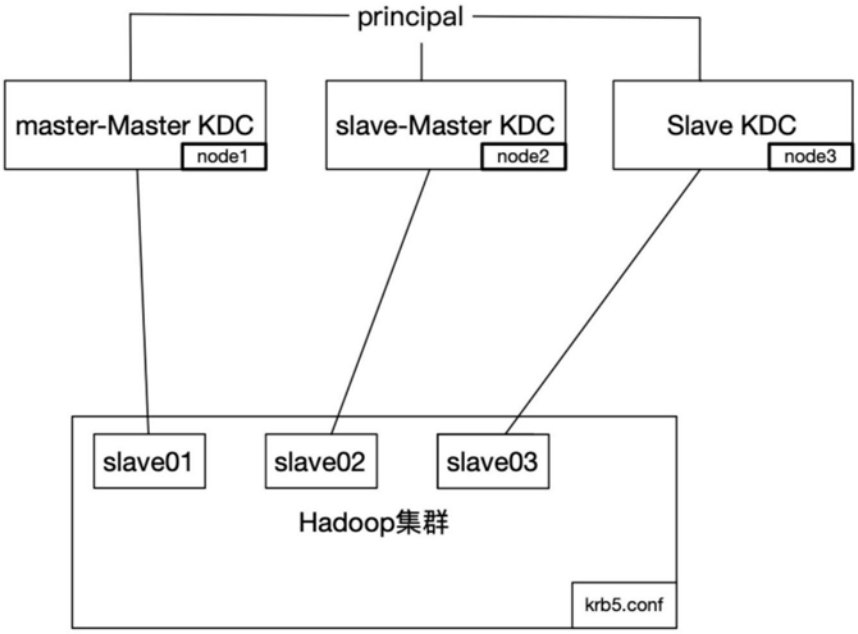

principal
master-Master KDC
node1
slave-Master KDC
node2
Slave KDC
node3
slave01
slave02
slave03
Hadoop集群
krb5.conf

图1

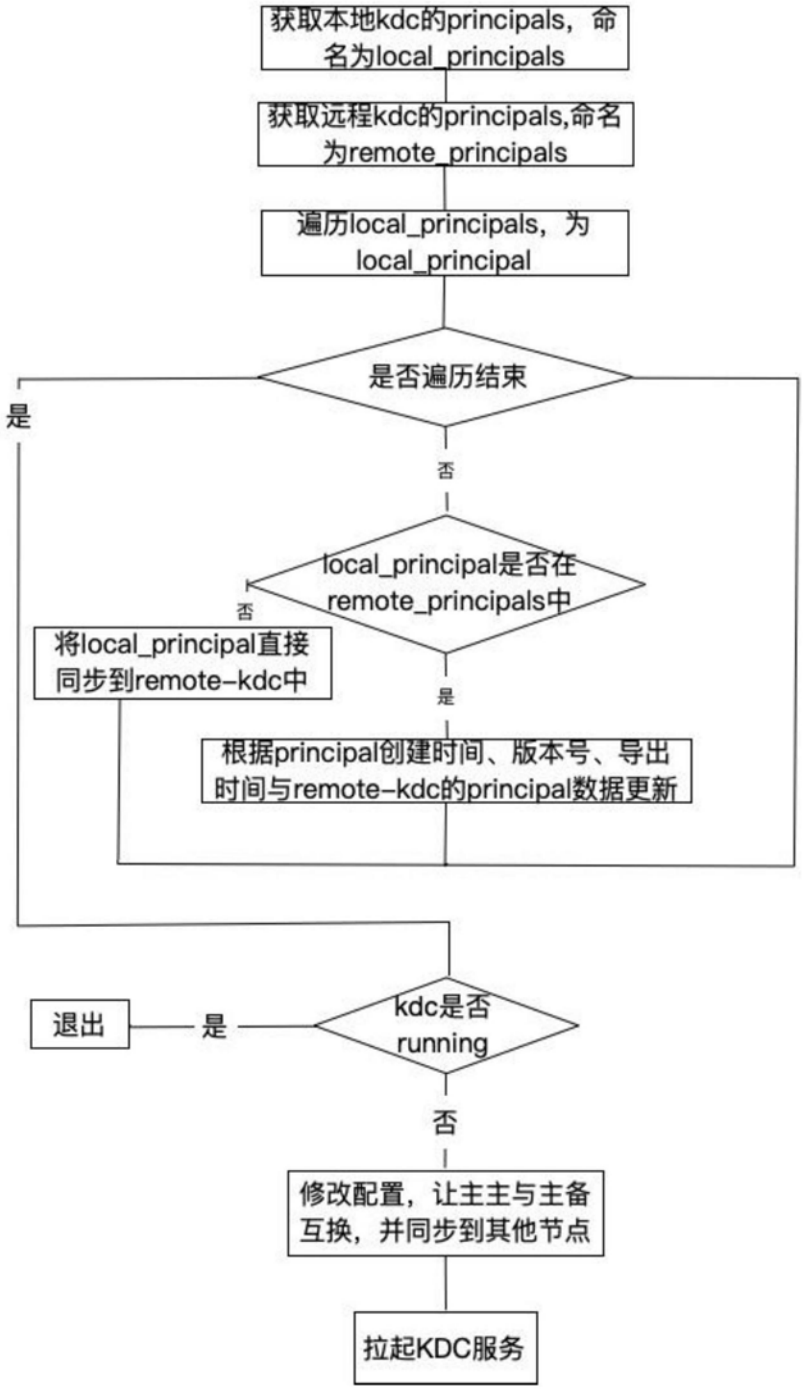

获取本地kdc的principals，命名为local_principals
获取远程kdc的principals,命名为remote_principals
遍历local_principals，为local_principal
是否遍历结束
是
否
local_principal是否在remote_principals中
否
将local_principal直接同步到remote-kdc中
是
根据principal创建时间、版本号、导出时间与remote-kdc的principal数据更新
退出
是
kdc是否running
否
修改配置，让主主与主备互换，并同步到其他节点
拉起KDC服务

图2

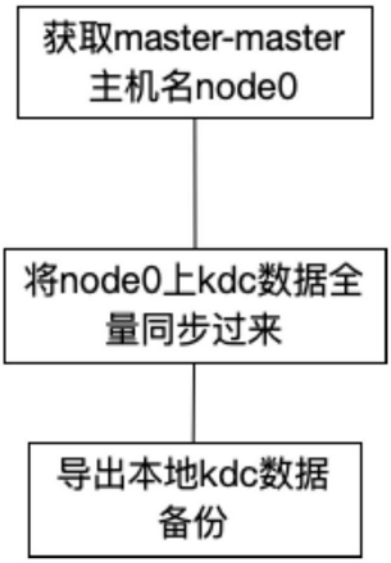

获取master-master
主机名node0
将node0上kdc数据全
量同步过来
导出本地kdc数据
备份

图3