20180321_Hadoop的HA机制

总结如下：
1- 高可用性主要是NameNode，有一个dump掉，其它的还能用
2- 同一个几圈中的一个热备的"主/备"两个冗余的NameNode(Node之间的快速转移)
3- slave负责待机状态，当活动节点对名字空间进行任何修改，它将把修改记录写到共享目录下的一个日志文件，备用节点会监听这个目录。当发现更改时，它会把修改内容同步到自己的名字空间。备用节点在故障转移时，它将保证已经读取了所有共享目录内的更改记录，保证在发生故障前的状态与活动节点保持完全一致。
4- DataNode上的配置,备用节点有最新的集群中块的位置信息，为了达到这一点，DataNode节点需要配置两个NameNode的位置，同时发送块的位置和心跳信息到两个NameNode。

5- 任何时候只有一个NameNode处于活动状态，对于HA集群的操作是至关重要的，否则两个节点之间的状态就会产生冲突、数据丢失或其他不正确的结果。为了达到这个目的，管理员必须为共享存储配置至少一个fencing方法。在宕机期间，如果不能确定之间的活动节点已经放弃活动状态，fencing进程负责中断以前的活动节点编辑存储的共享访问。这可以防止任何进一步的修改名字空间，允许新的活动节点安全地进行故障转移。

==========
主要的东西就是上面的总结。
6-
- 只有一个NameNode是Active的，并且只有这个ActiveNameNode能提供服务，改变NameSpace。以后可以考虑让StandbyNameNode提供读服务。
- 提供手动Failover，在升级过程中，Failvoer在NameNode-DataNode之间写不变的情况下才能生效。(共享配置中会有fencing方法。)
- 在之前的NameNode重新恢复之后，不能提供failback。(重新恢复之后，不能再对共享数据进行更改)
- 数据一致性比Failover更重要 (两个NameNode要确保访问的数据信息是一致的。)
- 尽量少用特殊的硬件
- HA的设置和Failover都应该保证在两者操作错误或者配置错误的时候，不得导致数据损坏
- NameNode的短期垃圾回收不应该触发Failover (对垃圾回收的正确辨别)
- DataNode会同时向NameNode Active和NameNode Standly汇报块的信息。NameNode Active和NameNode Standby通过NFS备份MetaData信息到一个磁盘上面。 (DataNode除了同时记录两个NameNode的信息，基于心跳机制也得定时向上汇报)

==========
还有几种特殊的节点：
- Secondary NameNode：它不是HA，只是阶段性的合并Edits和FsImage，以缩短集群启动时间。当NameNode失效的时候，Secondray NN 无法立刻提供服务，Secondary NN甚至无法保证数据完整性。如果NN数据丢失，在上一次合并后的文件系统的改动会丢失。
- Backup NameNode:它在内存中复制了NN的当前状态，算是Warm Standby，可仅限于此，并没有Failover等。它同样是阶段性地做CheckPoint，也无法保证数据完整性。

两种手动方式如下：

- 手动把name.dir指向NFS：这是安全的Cold Standby，可以保证元数据不丢失，但集群的恢复则完全靠手动
- FaceBook AvatarNode:FaceBook有强大的运维做后盾，所以AvatarNode只是Hot Standby，并没有自动切换，当主NN失效的时候，需要管理员确认，然后手动把对外提供服务的虚拟机IP映射到Standby NN，这样做的好处是确保不会发生脑裂的场景。

NameNode是HDFS集群的单点故障，每一个集群只有一个NameNode，如果这个机器或进程不可用，整个集群就无法使用，直到重启NameNode或者新启动一个NameNode节点 影响HDFS集群不可用主要包括以下两种情况

- 类似机器宕机这样的意外情况将导致集群不可用，只有重启NameNode之后才可使用
- 计划内的软件或硬件升级，将导致集群在短时间范围内不可用。

HDFS的高可用性(HA)就可以解决上述问题，通过提供选择运行在同一集群中的一个热备的“主/备”两个冗余NameNode，允许在机器宕机或系统维护的时候，快速转移到另一个NameNode。

HA集群

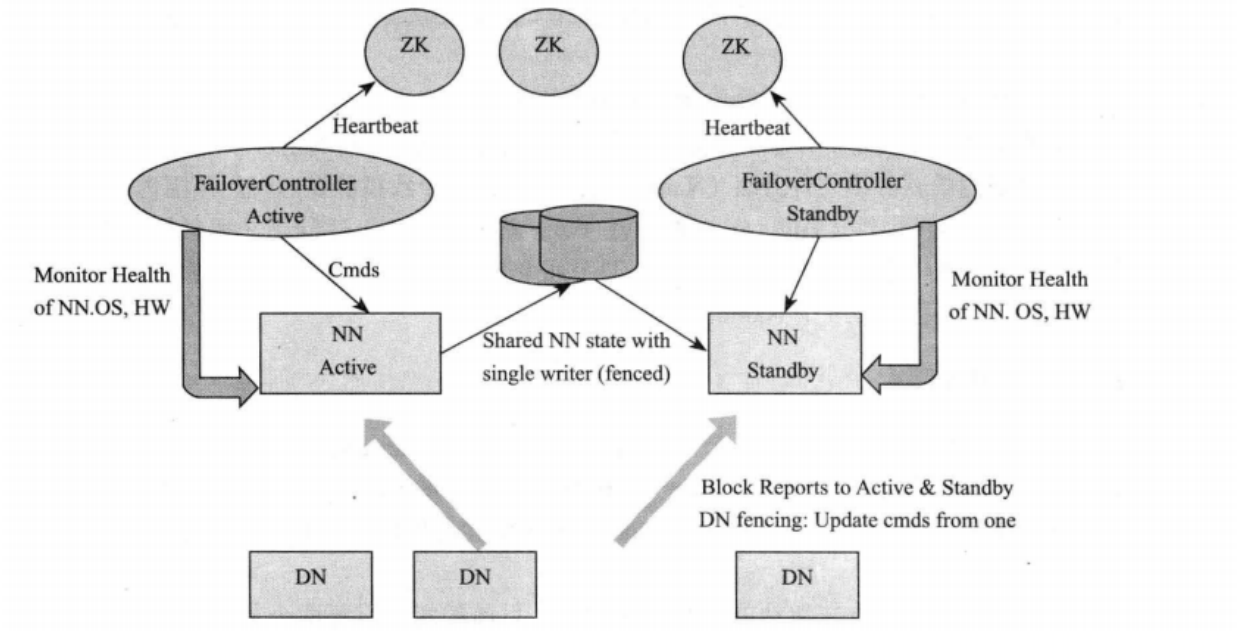
一个典型的HA集群，两个单独的机器配置为NameNode，在任何时候NmaeNode处于活动状态，另一个出于待机状态，活动的NameNode负责处理集群客户端的操作，待机时仅仅作为一个Slave,保持足够的状态，如果有必要提供一个快速的故障转移。

为了保持备用节点与活动节点状态的同步，需要两个节点同时访问一个共享存储设备（例如从NAS、NFS挂载）到一个目录。将有可能在未来版本中放款此限制。

当活动节点对名字空间进行任何修改，它将把修改记录写到共享目录下的一个日志文件，备用节点会监听这个目录。当发现更改时，它会把修改内容同步到自己的名字空间。备用节点在故障转移时，它将保证已经读取了所有共享目录内的更改记录，保证在发生故障前的状态与活动节点保持完全一致。

为了提供快速的故障转移，必须保证**备用节点有最新的集群中块的位置信息，为了达到这一点，DataNode节点需要配置两个NameNode的位置**，同时发送块的位置和心跳信息到两个NameNode。

任何时候只有一个NameNode处于活动状态，对于HA集群的操作是至关重要的，否则两个节点之间的状态就会产生冲突、数据丢失或其他不正确的结果。为了达到这个目的，管理员必须为共享存储配置至少一个fencing方法。在宕机期间，如果不能确定之间的活动节点已经放弃活动状态，fencing进程负责中断以前的活动节点编辑存储的共享访问。这可以防止任何进一步的修改名字空间，允许新的活动节点安全地进行故障转移。



HA架构原理图

架构解释如下

- 只有一个NameNode是Active的，并且只有这个ActiveNameNode能提供服务，改变NameSpace。以后可以考虑让StandbyNameNode提供读服务。
- 提供手动Failover，在升级过程中，Failvoer在NameNode-DataNode之间写不变的情况下才能生效。
- 在之前的NameNode重新恢复之后，不能提供failback。
- 数据一致性比Failover更重要
- 尽量少用特殊的硬件
- HA的设置和Failover都应该保证在两者操作错误或者配置错误的时候，不得导致数据损坏
- NameNode的短期垃圾回收不应该触发Failover
- DataNode会同时向NameNode Active和NameNode Standly汇报块的信息。NameNode Active和NameNode Standby通过NFS备份MetaData信息到一个磁盘上面。

HA机制

单点故障

- Secondary NameNode：它不是HA，只是阶段性的合并Edits和FsImage，以缩短集群启动时间。当NameNode失效的时候，Secondray NN 无法立刻提供服务，Secondary NN甚至**无法保证数据完整性**。如果NN数据丢失，在上一次合并后的文件系统的改动会丢失。
- Backup NameNode:它在内存中复制了NN的当前状态，算是Warm Standby，可仅限于此，并没有Failover等。它同样是**阶段性地做CheckPoint，也无法保证数据完整性。**
- **手动把name.dir指向NFS：这是安全的Cold Standby，可以保证元数据不丢失**，但集群的恢复则完全靠手动
- FaceBook AvatarNode:FaceBook有强大的运维做后盾，所以**AvatarNode只是Hot Standby，并没有自动切换，当主NN失效的时候，需要管理员确认，然后手动把对外提供服务的虚拟机IP映射到Standby NN，这样做的好处是确保不会发生脑裂的场景。**其某些设计思想和Hadoop 2.0里的HA非常相似，从时间上来看，Hadoop 2.0应该是借鉴了FaceBook的做法

集群容量和集群性能

单NameNode的架构使得HDFS在集群扩展性和性能上都有潜在的问题，当集群大到一定程度时后，NameNode进程使用的内存可能会达到上百GB，常用的估算公式为1GB对应1百万个块，按默认块大小计算，大概是64TB。同时，所有的元数据信息的读取和操作都需要与NNT通信，在集群规模变大后，NameNode成为性能的瓶颈。