HDFS 归档存储

- 一、介绍
- 二、存储类型和存储策略
- 三、配置使用
  - 3.1、存储策略设置
  - 3.2、Mover
  - 3.3、 hdfs缓存
- 四、测试性能

一、介绍

以前的HDFS只支持硬盘作为存储介质，HDFS-2832实现了HDFS的异构存储，使HDFS的存储介质不局限于硬盘，同时也可以存储在SSD、归档（存储空间大，计算能力弱）和内存中。HDFS缓存就是建立在异构存储的基础上，实现了将HDFS文件的某些hot files的副本存储在缓存中，从而可以达到增加读写速率，提升性能的效果。HDFS现在使用了一种名为lazy persist的策略，可以将hot files保存在缓存中，一段时间后再将这些文件的block转存到硬盘中做持久化。以后将会更加智能的根据文件使用情况判断哪些文件需要放在缓存，哪些文件可以持久化到硬盘。

- 归档存储是一种将hdfs日益增长的存储能力和计算能力相隔离的方案。
- 低价格、高存储空间和和低计算能力的节点可以用来作为集群数据的冷存储设备。
- 基于热区的数据可以移动到冷区的策略。
- 新增更多的存储节点作为冷区存储可以独立于计算能力提高集群的存储能力。
- 异构储存和归档存储的框架让hdfs可以包容其他类型的存储设备，比如SSD、内存。用户可以选择存储数据在SSD或内存中以期待获取更好的性能增益。

二、存储类型和存储策略

HDFS现在已经支持不同的存储介质：ARCHIVE, DISK, SSD and RAM_DISK。

DISK：普通硬盘

SSD：固态硬盘

ARCHIVE：高密度存储数据的介质来解决数据量的容量扩增的问题，用具有高存储密度（petabyte）和很低的计算能力的设备支持归档存储。

RAM_DISK：内存

存储策略：Hot, Warm, Cold, All_SSD, One_SSD and Lazy_Persist

根据不同的存储策略将文件存储到不同类型的存储设备中。

（1）Hot:需要存储和计算处理的数据。所有副本都存在DISK。

（2）Cold:存储需要有限的计算的数据、无需再用的数据，需要归档的数据需要从热区移动到冷区存储。当一个block是cold的，所有副本被存储在归档区。

（3）Warm:部分hot数据和部分cold数据。当一个block是warm的，它的一些副本被存在DISK,一些副本被存在归档区。

（4）All_SSD:存储所有副本在SSD。

（5）One_SSD:存储一个副本在SSD,其它副本被存在DISK。

（6）Lasy_Persist:单副本存在内存，先写在RAM_DISK,然后持久化在DISK。

策略 ID	策略名	Block Placement (n 副本)	新建文件的备选存储介质	副本的备选存储介质
15	Lasy_Persist	RAM_DISK: 1, DISK: n-1	DISK	DISK

12	All_SSD	SSD: n	DISK	DISK
10	One_SSD	SSD: 1, DISK: n-1	SSD, DISK	SSD, DISK
7	Hot (default)	DISK: n	<none>	ARCHIVE
5	Warm	DISK: 1, ARCHIVE: n-1	ARCHIVE, DISK	ARCHIVE, DISK
2	Cold	ARCHIVE: n	<none>	<none>

当存储空间足够时，block副本会根据表格第三列存储；如果第三列的存储设备类型不可用时（空间不够了），第四列和第五列（退一级所选择使用的存储类型）中的后补存储类型列表会用来替补第三列中不够的空间，进行文件和副本创建。

Lasy_Persist策略只适用于单副本block。超过一个副本的block，由于只有一个副本写到RAM_DISK不会提高整体的性能，所有副本会被写到DISK。

三、配置使用

3.1、存储策略设置

（1）配置：hdfs-site.xml

a)dfs.storage.policy.enabled=true启用存储策略功能。

b)配置属性dfs.datanode.data.dir 设置本地
存储目录,同时带上一个存储类型标签,声明此目录用的是哪种类型的存储介质.配置参考如下:

```
DISK目录：/grid/dn/disk0  ---> [DISK]file:///grid/dn/disk0
SSD目录：/grid/dn/ssd0 --->  [SSD]file:///grid/dn/ssd0
ARCHIVE目录：/grid/dn/archive0  ---> [ARCHIVE]file:///grid/dn/archive0
RAM_DISK 目录：/grid/dn/ram0  --->  [RAM_DISK]file:///grid/dn/ram0
```

如果目录前没有带上[SSD]/[DISK]/[ARCHIVE]/[RAM_DISK]这4种中的任何一种,则默认是DISK类型。

（2）相关命令：（使用hdfs用户或者超级用户）

```
[hdfs@node5 ~]$ hdfs storagepolicies
Usage: bin/hdfs storagepolicies [COMMAND]
          [-listPolicies]
          [-setStoragePolicy -path <path> -policy <policy>]
          [-getStoragePolicy -path <path>]
          [-help <command-name>]
```

```
$hdfs storagepolicies -listPolicies
```

列出所有可用的存储策略。

```
[hdfs@node5 ~]$ hdfs storagepolicies -listPolicies
Block Storage Policies:
        BlockStoragePolicy{COLD:2, storageTypes=[ARCHIVE], creationFallbacks=[], replicationFallbacks=[]}
        BlockStoragePolicy{WARM:5, storageTypes=[DISK, ARCHIVE], creationFallbacks=[DISK, ARCHIVE], replicationFallbacks=[DISK, ARCHIVE]}
        BlockStoragePolicy{HOT:7, storageTypes=[DISK], creationFallbacks=[], replicationFallbacks=[ARCHIVE]}
        BlockStoragePolicy{ONE_SSD:10, storageTypes=[SSD, DISK], creationFallbacks=[SSD, DISK], replicationFallbacks=[SSD, DISK]}
        BlockStoragePolicy{ALL_SSD:12, storageTypes=[SSD], creationFallbacks=[DISK], replicationFallbacks=[DISK]}
        BlockStoragePolicy{LAZY_PERSIST:15, storageTypes=[RAM_DISK, DISK], creationFallbacks=[DISK], replicationFallbacks=[DISK]}
[hdfs@node5 ~]$
```

```
$ hdfs storagepolicies -setStoragePolicy -path / -policy hot
```

将根目录设置为hot策略。

```
[hdfs@node5 ~]$ hdfs storagepolicies -setStoragePolicy -path / -policy hot
Set storage policy hot on /
```

```
$hdfs storagepolicies -getStoragePolicy -path /
```

查看/目录的存储策略。

```
[hdfs@node5 ~]$ hdfs storagepolicies -getStoragePolicy -path /
The storage policy of /:
BlockStoragePolicy{HOT:7, storageTypes=[DISK], creationFallbacks=[], replicationFallbacks=[ARCHIVE]}
```

注意：在hdfs上新创建一个目录，它的存储策略是未定义的，需要手动配置。

3.2、Mover

一个归档数据的数据迁移工具。扫描hdfs上的文件，检查是block放置规则是否满足存储策略。如果有block违反存储策略，会将它的副本移动到不同类型的存储设备中以满足存储策略的要求。

命令：

```
hdfs mover [-p <files/dirs> 或者-f <local file name>]
```

-p <files/dirs>	指定要迁移的hdfs文件或目录，若有多个文件或目录，以空格分隔。
-f <local file>	指定包含要迁移的HDFS文件或目录列表的本地文件。

注：如果-p和-f被省略了，默认目录是根目录。

3.3、 hdfs缓存

目前有2种缓存文件系统tmpfs和ramfs，当前版本的Hadoop只支持tmpfs，ramfs的支持还在开发中（https://issues.apache.org/jira/browse/HDFS-8584）。

tmpfs的存储空间受到Linux内核的限制，而ramfs会使用Linux系统的所有空闲内存。

tmpfs在内存不足的时候，会使用swap分区，因此，为了达到最佳性能，通常要将系统的swap分区禁掉。

tmpfs挂载

执行如下命令：

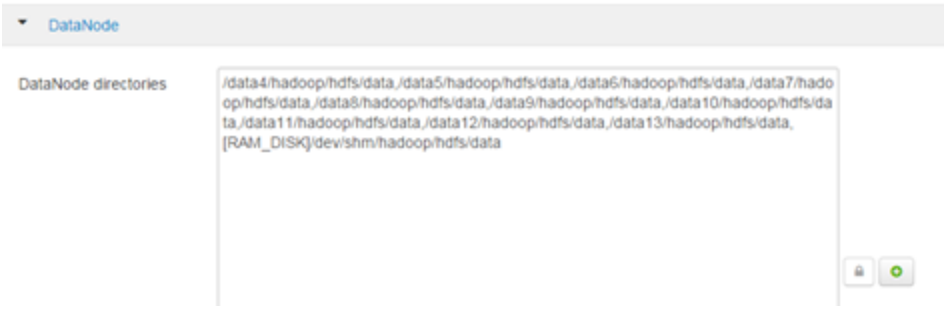
mount -t tmpfs -o size=32g tmpfs /dev/shm

同时，需要将tmpfs加入/etc/fstab中，保证机器重启自动挂载。在/etc/fstab文件加入：

tmpfs /dev/shm tmpfs defaults

如果在hdfs-site.xml中配置了dfs.datanode.max.locked.memory参数，要保证tmpfs的挂载size大于或等于该参数的值。

配置datanode目录，加入RAM_DISK



如上图所示，在hdfs-site.xml中配置dfs.datanode.data.dir参数，加入[RAM_DISK]/dev/shm/hadoop/hdfs/data目录。

注意：[RAM_DISK]标签一定要加上，否则hdfs会把tmpfs目录当成普通的[DISK]目录，这样将导致DN重启的时候，丢失数据。

配置完成后重启DN。

设置storage policy

通过执行hdfs storagepolicies -setStoragePolicy -path <path> -policy LAZY_PERSIST命令，将需要使用HDFS缓存的目录设置成LAZY_PERSIST策略。

往该目录写文件的时候，将会写一个副本在[RAM_DISK]中，其余副本通过DN的pipeline写到其他两个DN的[DISK]中。从该目录读文件的时候，如果[RAM_DISK]中有所需要的副本，那么将直接从内存读取，避免了从硬盘中读取消耗时间。

副本置换策略

当缓存用满的时候，HDFS会随机找一个[DISK]目录，将需要写入[RAM_DISK]的block临时写在[DISK]目录的Block Pool下的lazy文件夹中。而HDFS会有LazyWriter线程周期性遍历，周期由dfs.datanode.lazywriter.interval.sec参数指定，采用LRU算法，将一部分存在[RAM_DISK]中的副本持久化到[DISK]中去。

目前只支持LRU算法，该算法并不高效，以后还会支持LFU (least frequently used)算法。

四、测试性能

需求：把ssd和普通的sas盘分层存储，对比测试性能

步骤：

配置：hdfs-site.xml

（1）dfs.storage.policy.enabled=true启用存储策略功能。

（2）属性dfs.datanode.data.dir

SSD盘：[SSD]file:///grid/dn/ssd0

Sas盘（跟之前一致）：file:///data0

完成后，重启hdfs.

（3）创建测试目录并设置策略

```
$hdfs dfs -mkdir /ssd
$hdfs dfs -mkdir /sas
$hdfs storagepolicies -setStoragePolicy -path /sas  -policy Hot
$hdfs storagepolicies -setStoragePolicy -path /ssd  -policy All_SSD
```

（4）对两个目录进行hdfs写测试。