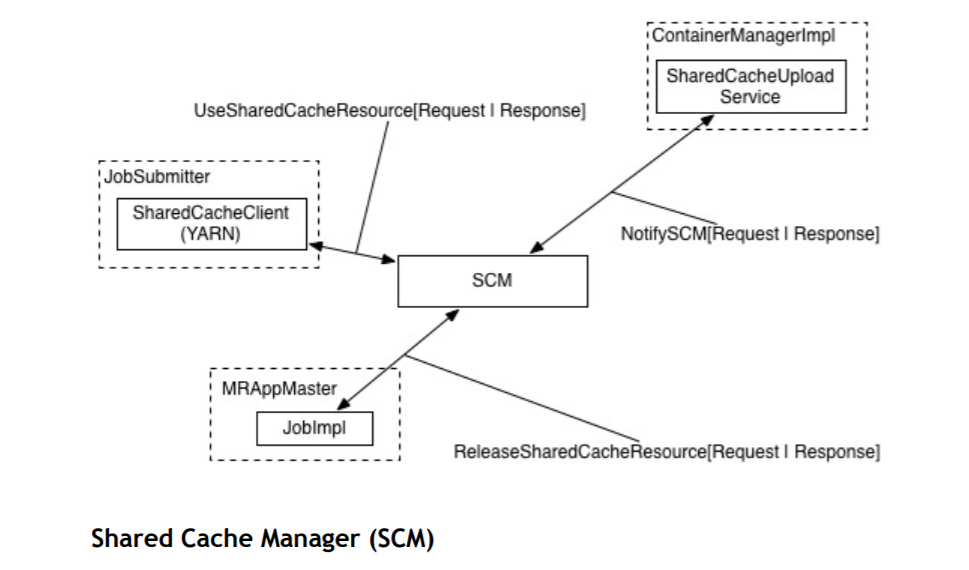
YARN Shared Cache

YARN App可以使用其他App及本App上轮执行时上传的资源，避免重复上传相同的文件，从而节省网络资源及减小应用启动时间，该功能是通过YARN Shared Cache来提供。其系统架构图如下所示：



核心组件：

* SCM，Shared Cache Service的主要组件，与客户端交互提供Shared Cache中的资源并用于缓存的清理。SCM以单独的进程运行在任意节点，避免影响YARN集群
* Localization Service，资源在本地化时，NM使用Localization Service将资源添加到Shared Cache，默认情况下Application Master Container向缓存中添加资源。YARN 应用的开发者可以通过ContainerLauchContext来显示指定Container上传哪些资源。
* Shared Cache Client，YARN应用的开发者和用户通过Shared Cache Client与Cache交互。其实现功能包括计算资源的checksum、在Shared Cache中声明应用资源、在应用生命周期使用资源等
* Shared Cache HDFS Directory，用于存储Shared Cache资源

# Shared Cache使用

1）创建Shared Cache在HDFS中的路径

*$hdfs dfs -mkdir /sharedcache //目录权限0755，所属者为share cache manager启动用户*

1. 配置yarn-site.xml

*<property>*

*<name>yarn.sharedcache.enabled</name>*

*<value>true</value>*

*</property>*

*<property>*

*<name>yarn.sharedcache.root-dir</name>*

*<value>/sharedcache</value>*

*</property>*

1. 启动SharedCache 守护进程，命令如下：

*$yarn --daemon start sharedcachemanager*

启动进程及使用端口如下：

*$ jps*

*27478 SharedCacheManager*

*$ netstat -anp|grep 27478*

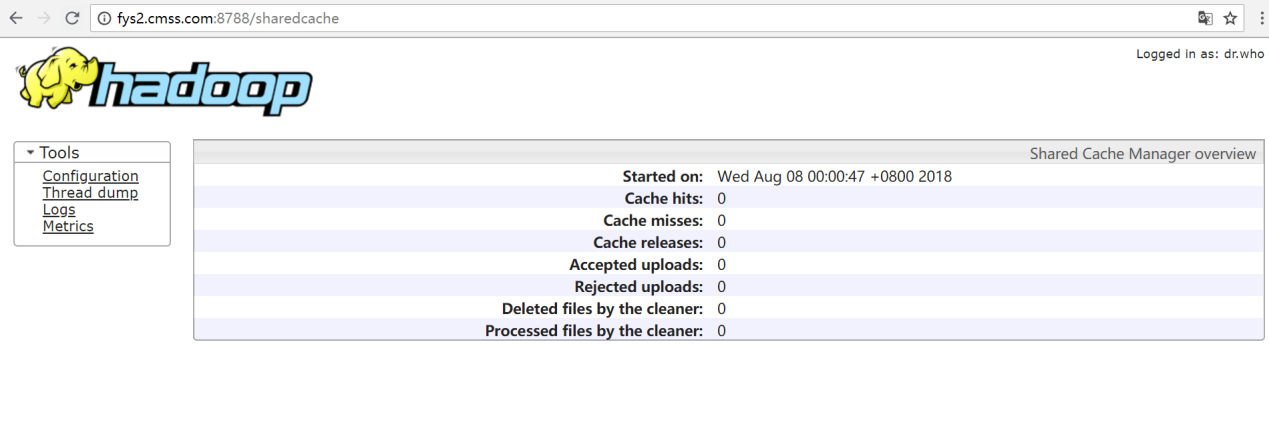
*tcp 0 0 0.0.0.0:8045 0.0.0.0:\* LISTEN 27478/java*

*tcp 0 0 0.0.0.0:8046 0.0.0.0:\* LISTEN 27478/java*

*tcp 0 0 0.0.0.0:8047 0.0.0.0:\* LISTEN 27478/java*

*tcp 0 0 0.0.0.0:8788 0.0.0.0:\* LISTEN 27478/java*

界面如下：



1. 可配置参数如下表所示：

|  |  |  |
| --- | --- | --- |
| 参数 | 描述 | 默认值 |
| yarn.sharedcache.enabled | 是否开启sharedcache | false |
| yarn.sharedcache.root-dir | sharecached的根目录 | /sharedcache |
| yarn.sharedcache.nested-level | sharedcache目录的最大层次 | 3 |
| yarn.sharedcache.store.class | SCM Store的实现类 | InMemorySCMStore |
| yarn.sharedcache.store.in-memory.staleness-period-mins | 资源的失效时间 | 10080min |
| yarn.sharedcache.store.in-memory.initial-delay-mins | CacheManager启动后cache clean的delay时间 | 10min |
| yarn.sharedcache.store.in-memory.check-period-mins | Cache的check周期 | 720min |
| yarn.sharedcache.admin.address | Shared Cache的管理端口 | 0.0.0.0:8047 |
| yarn.sharedcache.admin.thread-count | 管理线程数目 | 1 |
| yarn.sharedcache.webapp.address | Web App的端口 | 0.0.0.0:8788 |
| yarn.sharedcache.cleaner.period-mins | Cache的clean周期 | 1440 |
| yarn.sharedcache.cleaner.initial-delay-mins | CacheManager启动后Clean开启的delay周期 | 10min |
| yarn.sharedcache.cleaner.resource-sleep-ms | 处理两个Resource的sleep时间 | 0 |
| yarn.sharedcache.uploader.server.address | 与NM通信的端口 | 0.0.0.0:8046 |
| yarn.sharedcache.uploader.server.thread-count | 处理NM发送的资源处理请求的线程数目 | 50 |
| yarn.sharedcache.client-server.address | SCM的客户端端口 | 0.0.0.0:8045 |
| yarn.sharedcache.client-server.thread-count | 处理客户端请求的线程数目 | 50 |
| yarn.sharedcache.checksum.algo.impl | 计算checksum的算法 | ChecksumSHA256Impl |
| yarn.sharedcache.nm.uploader.replication.factor | NM的uploader的上传的资源副本数目 | 10 |
| yarn.sharedcache.nm.uploader.thread-count | NM Uploader的线程数目 | 20 |

# MapReduce Support

MapReduce可以通过使用YARN SharedCache，减少Job提交时间，使用步骤如下：

1）配置参数，mapred-site.xml

*property>*

*<name>mapreduce.job.sharedcache.mode</name>*

*<value>enabled</value>*

*</property>*

MapReduce用户基于resource Type上传资源，当启动时首先检查cache中存在已使用的资源，没有则异步上传该资源

1. 配置mapreduce sharedcache参数

*<property>*

*<name>mapreduce.job.cache.sharedcache.files</name>*

*<value>/data/README.txt</value>*

*</property>*

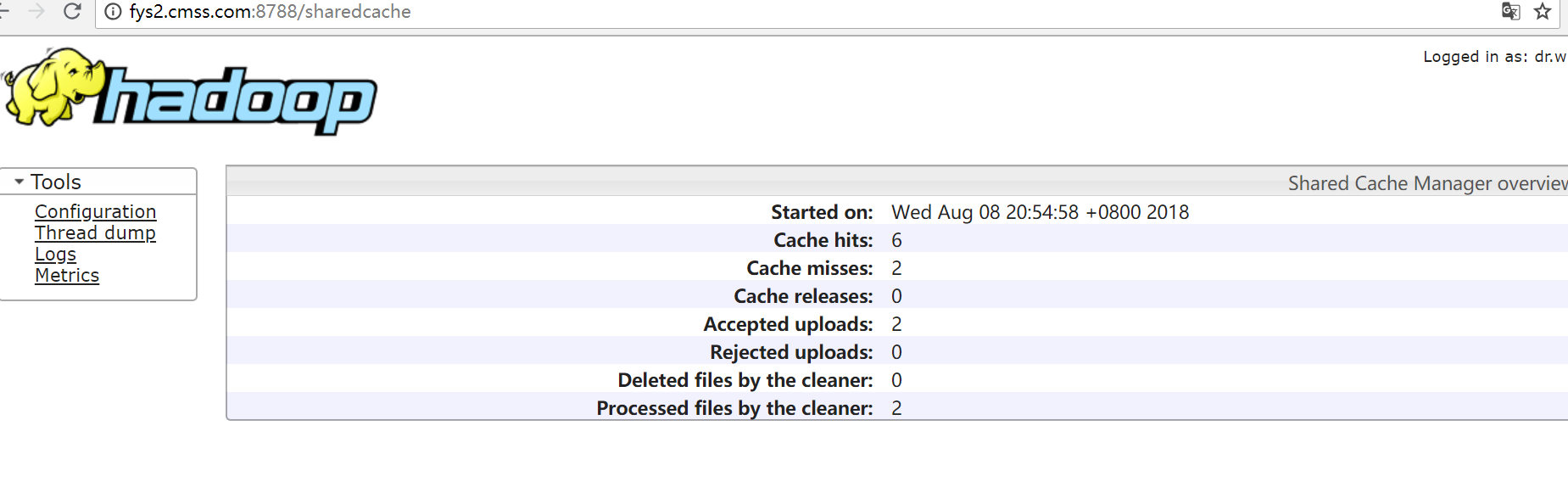
1. 提交Job作业

*hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.0-SNAPSHOT.jar pi 1 1000*

1. 执行后查看sharedcache目录，如下：

*/sharedcache/5/6/0/560c5fa5e0b8c5ad4fb527fb173571f662e8c06f1bf12307cd20e80e9586380b/README.txt*

文件上传到sharedcache目录下，查看页面如下：



https://issues.apache.org/jira/browse/YARN-1492

https://issues.apache.org/jira/browse/YARN-7282