Hadoop Static Group Mapping

在core-site.xml中配置如下：

*<property>*

*<name>hadoop.user.group.static.mapping.overrides</name>*

*<value>hive=hadoop,hive;hdfs=hadoop,hdfs</value>*

*</property>*

配置这些参数后可以绕过LDAP(获取其他group 查询机制)来获取用户组，缺点是不能提供宽松的用户管理。该参数配置后，在Groups中会将用户及用户组的对应关系写入成员变量：

*private final AtomicReference<Map<String, List<String>>> staticMapRef =*

*new AtomicReference<>();*

Group#getGroups的执行如下：

*public List<String> getGroups(final String user) throws IOException {*

*Map<String, List<String>> staticUserToGroupsMap = staticMapRef.get();*

*if (staticUserToGroupsMap != null) { //先从静态用户组映射关系中获取*

*List<String> staticMapping = staticUserToGroupsMap.get(user);*

*if (staticMapping != null) {*

*return staticMapping;*

*}*

*}*

*... //静态映射关系中没有，则从GroupProvider中拉取*

*return cache.get(user);*

*}*

staticMapRef的初始化如下，根据参数进行用户组的解析：

*private void parseStaticMapping(Configuration conf) {*

*String staticMapping = conf.get(*

*CommonConfigurationKeys.HADOOP\_USER\_GROUP\_STATIC\_OVERRIDES,*

*CommonConfigurationKeys.HADOOP\_USER\_GROUP\_STATIC\_OVERRIDES\_DEFAULT);*

*Collection<String> mappings = StringUtils.getStringCollection(staticMapping, ";");*

*Map<String, List<String>> staticUserToGroupsMap =*

*new HashMap<String, List<String>>();*

*for (String users : mappings) {*

*Collection<String> userToGroups = StringUtils.getStringCollection(users,*

*"=");*

*......*

*String user = userToGroupsArray[0];*

*List<String> groups = Collections.emptyList();*

*if (userToGroupsArray.length == 2) {*

*groups = (List<String>) StringUtils*

*.getStringCollection(userToGroupsArray[1]);*

*}*

*staticUserToGroupsMap.put(user, groups);*

*}*

*staticMapRef.set(*

*staticUserToGroupsMap.isEmpty() ? null : staticUserToGroupsMap);*

*}*