Spark任务日志过大导致磁盘溢出问题解决方案

一 问题背景

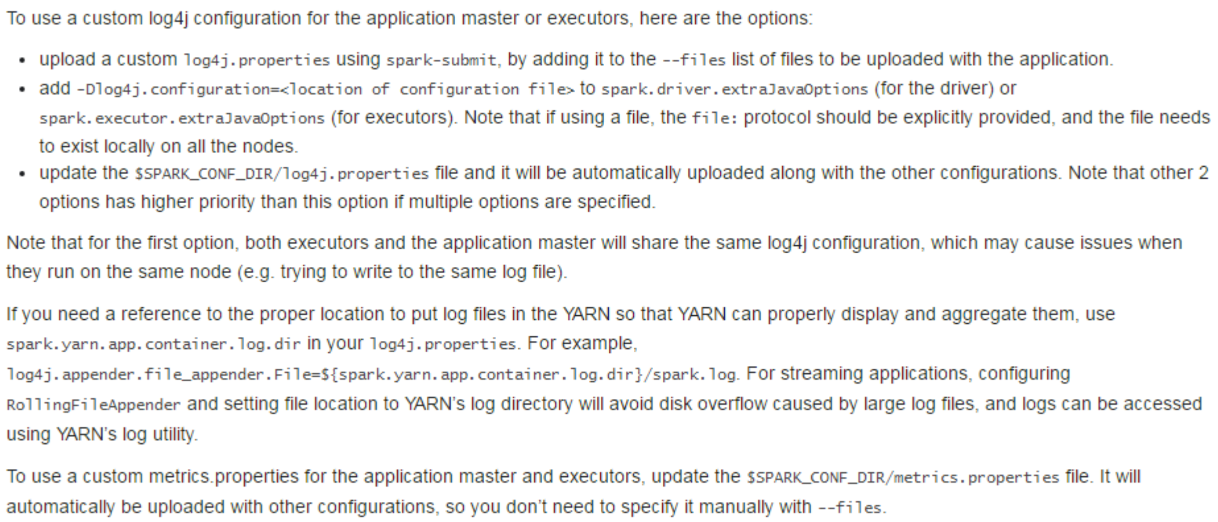
平台近期出现多次spark任务日志文件将磁盘打满，导致平台异常报警和任务失败的情况，这些任务包括Spark-Streaming任务和普通Spark任务。产生该问题的原因主要是：

Spark-Streaming任务运行时间比较长，Executor和Driver进程产生的Spark系统日志数量很大；业务用户在应用代码中使用 `System.out.print` 等输出了大量的调试信息（曾有任务运行40分钟打满100G日志文件）。以上信息全部输出在Yarn Container日志路径下的 `stdout` 和 `stderr` 里面，而Yarn本身没有对这些文件大小做限制，导致文件无限增长，最终将磁盘打满。

二 解决方案

2.1 解决思路

针对该问题，Spark官网给出了解决方案：



To use a custom log4j configuration for the application master or executors, here are the options:

- upload a custom `log4j.properties` using `spark-submit`, by adding it to the `--files` list of files to be uploaded with the application.
- add `-Dlog4j.configuration=<location of configuration file>` to `spark.driver.extraJavaOptions` (for the driver) or `spark.executor.extraJavaOptions` (for executors). Note that if using a file, the `file:` protocol should be explicitly provided, and the file needs to exist locally on all the nodes.
- update the `$SPARK_CONF_DIR/log4j.properties` file and it will be automatically uploaded along with the other configurations. Note that other 2 options has higher priority than this option if multiple options are specified.

Note that for the first option, both executors and the application master will share the same log4j configuration, which may cause issues when they run on the same node (e.g. trying to write to the same log file).

If you need a reference to the proper location to put log files in the YARN so that YARN can properly display and aggregate them, use `spark.yarn.app.container.log.dir` in your `log4j.properties`. For example, `log4j.appender.file_appender.File=${spark.yarn.app.container.log.dir}/spark.log`. For streaming applications, configuring `RollingFileAppender` and setting file location to YARN's log directory will avoid disk overflow caused by large log files, and logs can be accessed using YARN's log utility.

To use a custom metrics.properties for the application master and executors, update the `$SPARK_CONF_DIR/metrics.properties` file. It will automatically be uploaded with other configurations, so you don't need to specify it manually with `--files`.

在此基础上，结合平台Spark组件部署实际情况，制定以下方案：

- spark yarn contanier（包括executor和driver进程）单独使用一套日志机制，采用Rolling模式滚动删除，保留最新的5G日志数据：单个日志文件最大1G，最多保留5个，开启滚动删除。
- 业务用户在应用代码输出的调试信息重定向至log4j日志中，限制单个文件大小并应用滚动删除机制。

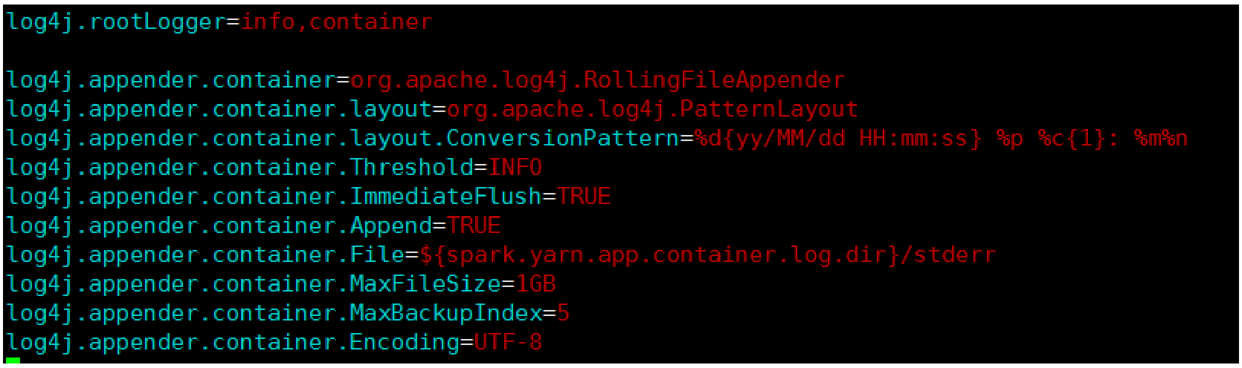
2.2 实施方案

- 自定义 `log4j.properties` 配置文件，限制文件大小，开启滚动删除。
- 将该文件放置在 `$SPARK_HOME/conf/yarn-container-log4j-conf` 文件夹下面，与 `$SPARK_HOME/conf/log4j.properties` 区分。
- 在 `spark-defaults.conf` 中通过 `spark.yarn.dist.files` 配置该文件，实现任务提交时上传，并下发供Executor和Driver进程使用。
- 增加输出信息重定向日志功能，使业务输出的调试信息重定向至log4j配置的指定日志文件。
- 在 `ApplicationMaster` 和 `CoarseGrainedExecutorBackend` 主进程中嵌入重定向函数，实现Driver进程和Executor进程业务调试信息由管道输出转变成日志输出。

- **同时，针对spark2.0以上版本，社区版本源码中Logging类为内部类，内部版本将Logging类更改为公共类，供业务调用，可替代System.out输出方式**

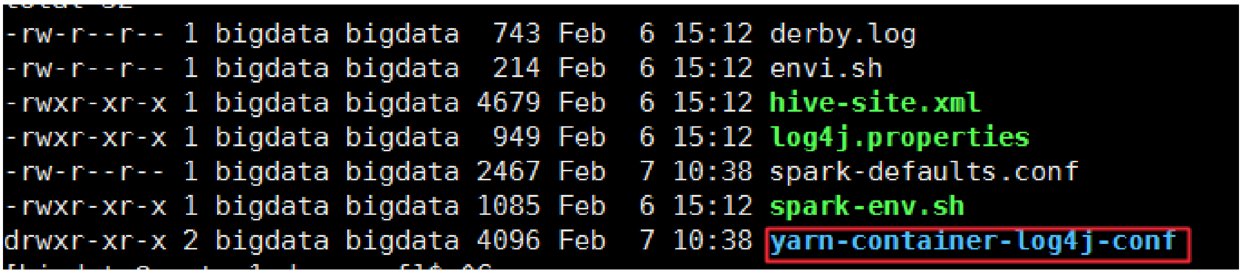
2.3 具体配置

自定义log4j.properties配置如下：



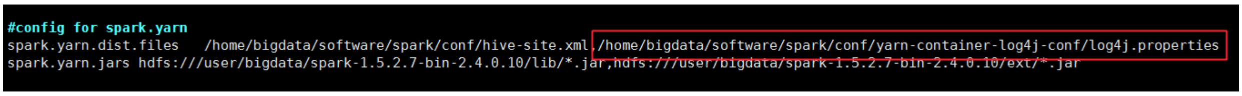
```
log4j.rootLogger=info,container

log4j.appender.container=org.apache.log4j.RollingFileAppender
log4j.appender.container.layout=org.apache.log4j.PatternLayout
log4j.appender.container.layout.ConversionPattern=%d{yy/MM/dd HH:mm:ss} %p %c{1}: %m%n
log4j.appender.container.Threshold=INFO
log4j.appender.container.ImmediateFlush=TRUE
log4j.appender.container.Append=TRUE
log4j.appender.container.File=${spark.yarn.app.container.log.dir}/stderr
log4j.appender.container.MaxFileSize=1GB
log4j.appender.container.MaxBackupIndex=5
log4j.appender.container.Encoding=UTF-8
```

文件放置在$SPARK_HOME/conf/yarn-contanier-log4j-conf/路径下：



```
-rw-r--r-- 1 bigdata bigdata  743 Feb  6 15:12 derby.log
-rw-r--r-- 1 bigdata bigdata  214 Feb  6 15:12 envi.sh
-rwxr-xr-x 1 bigdata bigdata 4679 Feb  6 15:12 hive-site.xml
-rwxr-xr-x 1 bigdata bigdata  949 Feb  6 15:12 log4j.properties
-rw-r--r-- 1 bigdata bigdata 2467 Feb  7 10:38 spark-defaults.conf
-rwxr-xr-x 1 bigdata bigdata 1085 Feb  6 15:12 spark-env.sh
drwxr-xr-x 2 bigdata bigdata 4096 Feb  7 10:38 yarn-container-log4j-conf
```

spark-defaults.conf修改配置如下：



```
#config for spark.yarn
spark.yarn.dist.files   /home/bigdata/software/spark/conf/hive-site.xml,/home/bigdata/software/spark/conf/yarn-container-log4j-conf/log4j.properties
spark.yarn.jars hdfs:///user/bigdata/spark-1.5.2.7-bin-2.4.0.10/lib/*.jar,hdfs:///user/bigdata/spark-1.5.2.7-bin-2.4.0.10/ext/*.jar
```