

In the currently per-job scenario, all the user jars and Flink system jars are set to system classpath and FlinkUserCodeClassLoader actually does not have any user jars.

System Classpath of Current PerJob Mode

Currently, the user jars are all added to system classpath when starting a PerJob cluster.

StandalonePerjob

Currently, all user job's jars are copied to FLINK_LIB_DIR. For example, all the files in FLINK_JOB_ARTIFACTS_DIR are copied to FLINK_LIB_DIR when building flink job cluster DockerImage.

YarnPerJob

Currently, the user can provide users' jars/resources in two ways. One is jar-file the other is -yt. These jars and resources are distributed by yarn with the Flink system jars/resources. All the user and system jar/resources are added to Flink system classpath, which used to startup the TM and JM, by the YarnClusterDescriptor.deployJobCluster.

All jars distributed by yarn contains:

1. System Jars
 - a. FLINK_LIB_DIR/*
 - b. PLUGIN/*
 - c. flink-dist.jar
2. System Config
 - a. flink-conf.yaml
 - b. io4j.properties
 - c. logback.xml
 - d. other security file and so on
3. User's jars/resources
 - a. JarFile
 - b. ShipDirectories
 - c. JobGraph

MesosPerjob

Currently, all the job files are copied to the FLINK_LIB_DIR.(the documentation suggested).

Separating System Class and User Class

We should not add user jars to the system classpath when starting a per-job cluster if we want to separate system classloader and user classloader.

As Till said the simpler way to add user jar into the user classloader is to add it when retrieving JobGraph.

I think we could change the ClassPathJobGraphRetriever and PackagedProgram to support all the three scenarios.

Add two fields to ClassPathJobGraphRetriever

```
public class ClassPathJobGraphRetriever {  
    // the directory that contains the jar and resource files of a job  
    public final String[] jobDirs;  
    // don't generate the jobGraph if the jobGraph already exists (deterministic)  
    @Nullable  
    public final String jobGraphFile;  
}
```

Add a new constructor to PackagedProgram

```
public class PackagedProgram{  
    // the jars in the jobDirs will be added to 'classpath'  
    // the parent path of resources in the jobDirs will be added to 'classpath'  
    // all the items added to the classpath will use the relative path because yarn mode  
will  
    public PackagedProgram(String[] jobDirs, Class<?> entryPointClass, String...  
args)  
}
```

Question:

1. Why PackagedProgram use parentFirst mode?
2. getAllLibraries() maybe have some duplicate jars. This is because extractedTempLibraries is extracted from the jarFile. Does this have some special thoughts?

StandalonePerJob

Adding two command-line arguments job-jar-paths and job-graph-path to StandaloneJobClusterEntryPoint and these arguments will be passed to the ClassPathJobGraphRetriever

YarnPerJob

1. CliFrontend

- a. Currently, the JobGraph is generated before CliFrontend decides it is Perjob mode or Session mode.
 - b. the new CliFrontend does not generate JobGraph in the per-job mode.
- 2. YarnClusterDescriptor.deployJobCluster
 - a. The system's jar/resource and user's jar/resource are to be shipped as before. But only the System's jar/resource is added to the classpath, which uses to start JM and TM. But the user's jar/resources will not add to the classpath.
 - b. user's jar
 - i. jar-file:
 - 1. this jar-file will be extracted and added to a remote directory xxx/___flink_jar_file/ and this directory will be distributed as -yt
 - ii. -yt ship_file_directory
 - 1. these files will be distributed with as before
 - c. ___flink_jar_file/ and ship_file_directory will be passed to YarnJobClusterEntrypoint
- 3. YarnJobClusterEntrypoint will use the ClassPathJobGraphRetriever
- 4. Other
 - a. two open questions
 - i. Should -t files be distributed to all nodes and be added to system classpath when starting a yarn session?
 - ii. Should (ORDER/FIRST/LAST) parameters be supported?
 - b. Different behavior in this new version
 - i. In the new version, the classpath contains all the jars/resources from -yt when the user's main is invoked.

Mesos

- 1. The user can add all the job jars to a job-jar-paths which is different from the FINK_LIB_DIR.
- 2. ./mesos-appmaster-job.sh -Djob-jar-paths=xxxx -Dinterval.jobgraph-path=yyyy
- 3. MesosJobClusterEntrypoint use the new version ClassPathJobGraphRetriever

task

- 1. ClassLoader
 - ~~a. assert non null jobClassDir~~
 - ~~b. set jobClassDir to final~~
 - c. AbstractUserClassPathJobGraphRetriever private final String jobClassDir does not use