**Executing Hadoop applications from WS-PGRADE**

**User Manual**

Version 1.3, 16 March 2016

Carlos Blanco, University of Cantabria, josecarlos.blanco@unican.es

Tamas Kiss, University of Westminster, t.kiss@westminster.ac.uk

**1. Introduction**

This user manual explains the configuration of a set of workflows that can be used to submit Hadoop applications through a WS-PGRADE portal. The workflow automatically creates a Hadoop cluster in the cloud (e.g. in OpenStack, OpenNebula or Amazon clouds) and then executes one or more Hadoop applications there. Finally, the Hadoop cluster can be destroyed freeing up the cloud resources. The user only needs to provide job input files (e.g. jar files and input data) and two configuration files for specifying cluster and job parameters.

Two methods can be used for submitting a Hadoop application through a WS-PGRADE portal:

* **Single Node Method:** The Single Node Method uses a single node workflow with a simple program to create Hadoop clusters in a cloud infrastructure, execute a Hadoop application in the cluster and retrieve results, and destroy the cluster.
* **Three Node Method**: The Three Node Method works the same way as the Single Node Method, but divides the task into three stages. The first stage creates the Hadoop cluster, the second executes the Hadoop application, and the third destroys the cluster. Each stage can be considered as a workflow node executing a particular task. The main idea behind dividing the complete process into three stages was to clearly separate these stages and allow more flexibility for the end-user when designing a complex application scenario. For example, the Three Node Method allows for reusability of the Hadoop cluster, as after cluster creation the user can add multiple execute nodes to the workflow.

**2. Prerequisites**

In order to use these Hadoop workflows a user needs **access to any WS-PGRADE/gUSE portal** configured to submit jobs to the EGI FedCloud and/or to the CloudBroker Platform. If the user uses the EGI FedCloud infrastructure, (s)he will need a grid certificate that has to be configured in the *Certificate* option under *Security* tab in the portal. Whereas, if (s)he uses CloudBroker then (s)he will require CloudBroker's credentials (username and password) that they have to be indicated in the *Cloud* option under *Security* tab. In both cases Hadoop has to be pre-deployed on the target cloud sites.

At the time of writing this user manual the following WS-PGRADE/gUSE portals fulfill the above requirements:

* EGI FedCloud Portal: https://fedcloud-gateway.lpds.sztaki.hu/

 https://guse-fedcloud-gateway.sztaki.hu/

* CloudSME Gateway: http://cloudsme-prod.lpds.sztaki.hu/liferay-portal-6.1.0/

**3. Configuration files**

Before going into details regarding the parameterization and execution of the pre-prepared workflows, this section gives a generic overview of the configuration files the user needs to prepare/modify. There are two configuration files: the cluster configuration file (cluster.cfg), and the job configuration file (job.cfg). These files have to be full in by the user to submit Hadoop workflows.

**3.1 Cluster configuration file**

The *cluster.cfg* file has the following parameters:

**Cluster section**

* + ***infrastructure***: Infrastructure tag (e.g. fedcloud, cloudbroker)
	+ ***cloud***: Cloud resource (e.g. CESNET, BIFI, RECAS-BARI)
	+ ***app***: Hadoop version (e.g. Hadoop-2.7.1)
	+ ***flavour***: Size of the VM (e.g. Small, Medium, Large and XLarge)
	+ ***nodes***: Number of Hadoop nodes
	+ ***volume***: Size of block storage in GB for each node (only available for fedcloud infrastructure)

**Credentials section**

* + ***myproxy\_server***: Hostname of myproxy server (e.g. myproxy1.egee.cesnet.cz)
	+ ***myproxy\_password***: Myproxy password
	+ ***username***: Myproxy credential name
	+ ***cloudbroker\_url***: URL of CloudBroker platform
	+ ***cloudbroker\_password***: Password of CloudBroker platform

*Please note that Myproxy information is only required in case of using the EGI FedCloud, while CloudBroker related credentials are only necessary for CloudBroker managed clouds.*

For instance, this is an EGI FedCloud cluster configuration file:

[cluster]

infrastruture=fedcloud

cloud=CESNET

app=Hadoop-2.7.1

flavour=Small

nodes=2

[credentials]

myproxy\_server=myproxy1.egee.cesnet.cz

myproxy\_password=\*\*\*\*\*\*

username=josecarlos.blanco

**3.2 Job configuration file**

The job.cfg file has these parameters:

**Job section**

* + ***input\_data***: Where to get input data from. Currently supported input data sources are *local* (uploaded directly by the user to the portal server), *s3/s3n/s3a*, *hftp*, *hdfs*, *swift*, *http/https* and *sftp/ftp*. If there are several input sources, those can be defined separated by a comma.

(e.g. local, s3n://AWSAccessKeyId:AWSSecretKey@bucket/file, http://server/file,swift://bucket/path?auth\_url=\*\*\*;tenant=\*\*\*;username=\*\*\*;password=\*\*\*, hdfs://server/file, sftp://user:password@server/file)

* + ***output\_data***: Where to transfer output data files. Currently supported output data destinations are *local*, *s3/s3n/s3a*, *hftp*, *hdfs*, *swift*, *http/https* and *sftp/ftp*. If it is needed to copy job’s outputs to several destinations, they can be defined separated by a comma.

(e.g. local, sftp://user:password@server/path, hdfs://server/path, s3://AWSAccessKeyId:AWSSecretKey@bucket)

* + ***class\_job***: The main class name of the Hadoop application to be executed on the cluster, complete with package info (e.g. org.myorg.WordCount)
	+ ***jar\_file***: jar file of the Hadoop application to be executed (e.g. WordCount.jar).
	+ ***map\_tasks***: Number of map tasks for the application.
	+ ***reduce\_tasks***: Number of reduce tasks for the application.

For instance, this is an EGI FedCloud job configuration file:

[job]

input\_data = local

output\_data = local

job\_class = WordCount

jar\_file = wordcount.jar

map\_tasks = 2

reduce\_tasks = 1

**4. How to use the Single Node Method and the Three Node Method**

* In order to assist you when creating the configuration files, we have created configuration file templates that you only need to customize. Download these configuration file templates from the [SHIWA repository](https://shiwa-repo.cpc.wmin.ac.uk/shiwa-repo/) where the workflows are available, and customize details in the *job.cfg* and *cluster.cfg* files as necessary (see Section 3).
* Copy your Hadoop application executable (jar file), **or** its source code files together with the *build.sh* script (see Section 5) to compile it, into a folder of your choice. (Please note that the solution supports two options: you either compile your Hadoop application yourself in which case you will upload your compiled *jar* file, **or** you do the compilation at execution time in the cloud for which you will need the source code and the *build.sh* script.)
* If the input data is *local,* copy your Hadoop input files into a folder called *input*, compress this folder to create a tar archive called *input.tar,* and copy this compressed file to the same folder where your executable/source code is. If your input is not *local* then input files have to be indicated in the *job.cfg* file.
* Compress the content of your folder including the job executable or its code sources and the *build.sh* script, and the compressed job input files (if input\_data is *local*) as a tar archive called *Data.tar*.
* Log in to the WS-PGRADE portal, select the *import* option under the *Workflow* tab and select *Remote SHIWA Repository*.
* From the list of public bundles, find and import the bundle named “*Hadoop\_Single\_Node*” and/or “*Hadoop\_Three\_Node*”.
* Select the *Workflow* tab and click on the *Concrete* option to find the new imported workflow.
* In the *Concrete* option you will find the deployed Hadoop application and configure the parameters as desired. In the *Job I/O* tab:
	+ For Single Node Method upload the *cluster.cfg* file to Port 0, the *job.cfg* file to Port 1 and the *Data.tar* file to Port 2 (see Fig 1). **Do not modify port 3 as that already contains the specific executables to create and destroy the Hadoop cluster and to execute your Hadoop application)**
	+ For the Three Node Method upload the *cluster.cfg* file to Port 0 in the Deploy Node, the *job.cfg* file to Port 1 and the *Data.tar* file to Port 2 in the Execute Node (see Fig 2). **Do not modify Port 1 in the Deploy Node as that already contains the specific executables to create and destroy the Hadoop cluster and to execute your Hadoop application)**
* Remember to save and upload the new configuration.
* You can now submit the workflow.

****

**Fig. 1.**Port description - Single Node Method

**Fig. 2.**Port description- Three Node Method

**5. How to compile Hadoop jobs**

Sometimes the user does not have access to Hadoop libraries to compile the *jar* file before the Hadoop job submission. In this case, (s)he can create a script called *build.sh* and put it in the same folder with the source code. In order to ease this process, the first parameter of this script will be all Hadoop library paths available in the Hadoop cluster. For instance, this is a *build.sh* script to compile the *WordCount jar* file:

 hadoop\_libraries=$1

# Compile WordCount

javac -classpath ${hadoop\_libraries} -d ./classes WordCount.java

# Create the Jar

jar -cvf wordcount.jar -C ./classes/ .

**6. How to create your own Hadoop workflow – for advanced users only**

If you want to customize the Hadoop workflow and extend it with your own workflow components then follow the next steps:

* Download Hadoop configuration files, together with the software bundle and the executable for each Hadoop Node from the Hadoop workflows available at the [*SHIWA repository*](https://shiwa-repo.cpc.wmin.ac.uk/shiwa-repo/).
* Customize details on the *job.cfg* and *cluster.cfg* configuration files.
* Copy your Hadoop job executable (jar file), **or** its source code files and the *build.sh* script (see Section 5), to the same folder.
* Copy your Hadoop job input files in a folder called input, compress this folder to create a tar archive called *input.tar* and copy this compressed file to the same folder as before (only if input\_data is *local*).
* Compress the folder as a tar file called *Data.tar*.
* Log in to the WS-PGRADE portal and create a workflow according to your application.
* Place the Create Node before the first Execute Node and place the Destroy Node after the last Execute Node (see Fig 1).
* Connect the output (channel) Port 2 of the Create Node to the input (channel) Port 0 of the first Execute Node.
* Connect the output (channel) Port 3 of the First Execute Node to the input (channel) Port 0 of the next Execute Node and repeat for every Execute Node.
* Connect the output (channel) Port 3 of the last Execute Node to the input (channel) Port 0 of the Destroy Node.
* Configure the rest Ports as it is described in the Fig 2.
* Update the executable for each node.
* Remember to save and upload the configuration.