Extras/Working with BDFS Alluxio

Alluxio (BDFS)

https://docs.oracle.com/en/cloud/paas/big-data-compute-cloud/csspc/big-data-file-system-bdfs.html (https://docs.oracle.com/en/cloud/paas/big-data-compute-cloud/csspc/big-data-file-system-bdfs.html)
and
https://www.alluxio.org/docs/master/en/index.html (https://www.alluxio.org/docs/master/en/index.html)
and
http://www.alluxio.org/docs/master/en/Configuration-Settings.html (http://www.alluxio.org/docs/master/en/Configuration-Settings.html)

Display the alluxio command line help...

```
%sh
alluxio fs
```

```
Usage: java AlluxioShell
       [cat <path>]                                       Prints the file's contents to the console.
       [checksum <Alluxio path>]                          Calculates the md5 checksum of a file in the Alluxio filesystem.
       [chgrp [-R] <group> <path>]                        Changes the group of a file or directory specified by args. Specify -R to change the group recursively.
       [chmod [-R] <mode> <path>]                         Changes the permission of a file or directory specified by args. Specify -R to change the permission recursively.
       [chown [-R] <owner> <path>]                        Changes the owner of a file or directory specified by args. Specify -R to change the owner recursively.
       [copyFromLocal <src> <remoteDst>]                  Copies a file or a directory from local filesystem to Alluxio filesystem.
       [copyToLocal <src> <localDst>]                     Copies a file or a directory from the Alluxio filesystem to the local filesystem.
       [count <path>]                                     Displays the number of files and directories matching the specified prefix.
       [cp [-R] <src> <dst>]                              Copies a file or a directory in the Alluxio filesystem. The -R flag is needed to copy directories.
       [createLineage <inputFile1,...> <outputFile1,...> [<cmd_arg1> <cmd_arg2> ...]]   Creates a lineage.
       [deleteLineage <lineageId> <cascade(true|false)>]  Deletes a lineage. If cascade is specified as true, dependent lineages will also be deleted.
       [du <path>]                                        Displays the size of the specified file or directory.
       [fileInfo <path>]                                  Displays all block info for the specified file.
       [free <path>]                                      Frees the space occupied by a file or a directory in Alluxio.
       [getCapacityBytes]                                 Gets the capacity of the Alluxio file system.
       [getUsedBytes]                                     Gets number of bytes used in the Alluxio file system.

ExitValue: 255
```

An example of listing the alluxio (BDFS) file system

```
1.00B     02-03-2018 15:54:03:180  Directory      /citibike/raw
130.33MB  02-03-2018 15:54:03:181  In Memory      /citibike/raw/201612-citibike-tripdata.csv
1.00B     02-03-2018 18:51:31:613  Directory      /citibike/modified
130.33MB  02-03-2018 18:51:31:630  In Memory      /citibike/modified/201612-citibike-tripdata.nh.csv
```

Explicitly load the data we want to work with into BDFS

```
%sh
alluxio fs load /citibike/modified/201612-citibike-tripdata.nh.csv
alluxio fs ls -R /citibike
```

```
1.00B     02-03-2018 15:54:03:180  Directory      /citibike/raw
130.33MB  02-03-2018 15:54:03:181  In Memory      /citibike/raw/201612-citibike-tripdata.csv
1.00B     02-03-2018 18:51:31:613  Directory      /citibike/modified
130.33MB  02-03-2018 18:51:31:630  In Memory      /citibike/modified/201612-citibike-tripdata.nh.csv
```

An example of listing the alluxio files system using hadoop fs

```
%sh
hadoop fs -ls swift://journeyC.default/citibike/modified
# use the below LOGGER setting to avoid lots of INFO logging from alluxio by default
export HADOOP_ROOT_LOGGER=WARN
hadoop fs -ls bdfs://localhost:19998/citibike/modified
```

```
Found 1 items
-rw-rw-rw-   1  136661199 2018-02-03 18:50 swift://journeyC.default/citibike/modified/201612-citibike-tripdata.nh.csv
Found 1 items
-rw-rw-rw-   3  136661199 2018-02-03 18:51 bdfs://localhost:19998/citibike/modified/201612-citibike-tripdata.nh.csv
```

An example of using alluxio (BDFS) versus standard object store (swift)

```
%spark

// If you get this error message:
// java.lang.IllegalStateException: Cannot call methods on a stopped SparkContext.
// Then go to the Settings tab, then click on Notebook.  Then restart the Notebook.  This will restart your SparkContext

//val swift_df = sqlContext.read.format("com.databricks.spark.csv").option("header", "true").load("swift://journeyC.default/citibike/raw/201612-citibike-tripdata.csv")
val swift_df = sqlContext.read.format("com.databricks.spark.csv").option("header", "true").load("swift://journeyC.default/citibike/modified/201612-citibike-tripdata.nh.csv")

//val bdfs_df = sqlContext.read.format("com.databricks.spark.csv").option("header", "true").option("inferSchema","true").load("bdfs://localhost:19998/citibike/raw/201612-citibike-tripdata.csv")
val bdfs_df = sqlContext.read.format("com.databricks.spark.csv").option("header", "true").option("inferSchema","true").load("bdfs://localhost:19998/citibike/modified/201612-citibike-tripdata.nh
    .csv")


{
var t0 = System.nanoTime()
println("# of rows: %s".format(
  swift_df.count()
))
var t1 = System.nanoTime()
println("Swift Count Elapsed time: " + (t1 - t0)/1000000000 + "s")
println("..")

t0 = System.nanoTime()
println("# of rows: %s".format(
  bdfs_df.count()
))
t1 = System.nanoTime()
println("BDFS Count Elapsed time: " + (t1 - t0)/1000000000 + "s")
println("..")
}
```

```
swift_df: org.apache.spark.sql.DataFrame = [528: string, 2016-12-01 00:00:04: string ... 13 more fields]
bdfs_df: org.apache.spark.sql.DataFrame = [528: int, 2016-12-01 00:00:04: timestamp ... 13 more fields]
# of rows: 812191
Swift Count Elapsed time: 7s
..
# of rows: 812191
BDFS Count Elapsed time: 1s
..
```

In order to use hive, we need to adjust an Alluxio parameter

This is needed in 18.1.2.

1.edit /u01/bdcsce/var/lib/ambari-agent/cache/stacks/HDP/2.4/services/ALLUXIO/package/templates/alluxio-site.template , and add
alluxio.underfs.object.store.mount.shared.publicly=true
2.restart alluxio via ambari

See https://www.alluxio.org/docs/master/en/Configuring-Alluxio-with-Swift.html (https://www.alluxio.org/docs/master/en/Configuring-Alluxio-with-Swift.html)

These instructions did not work in 18.1.2. You need to edit the alluxio-site.template file directly:
1.In ambari, navigate to Alluxio, then Configs.
2.Expand the custom alluixio-site section

3.Click Add Property...
4.Add this:
alluxio.underfs.object.store.mount.shared.publicly=true
5.Save the configuration
6.Restart alluxio.

Create an external hive table against BDFS (alluxio)

```
%sh

/u01/bdcsce/opt/alluxio/bin/alluxio fs chmod 777 /citibike/modified/

hive  <<EOF
DROP TABLE bike_trips_objectstore_bdfs;

CREATE external TABLE bike_trips_objectstore_bdfs (
TripDuration int,
StartTime timestamp,
StopTime timestamp,
StartStationID string,
StartStationName string,
StartStationLatitude string,
StartStationLongitude string,
EndStationID string,
EndStationName string,
EndStationLatitude string,
EndStationLongitude string,
BikeID int,
UserType string,
BirthYear int,
Gender int
)
ROW FORMAT delimited
FIELDS TERMINATED BY ','
location 'bdfs://localhost:19998/citibike/modified/';


exit;

EOF
```

```
Changed permission of /citibike/modified to 777
WARNING: Use "yarn jar" to launch YARN applications.
Logging initialized using configuration in file:/etc/hive/2.4.2.0-258/0/hive-log4j.properties
hive> DROP TABLE bike_trips_objectstore_bdfs;
OK
Time taken: 0.948 seconds
hive>
    > CREATE external TABLE bike_trips_objectstore_bdfs (
    > TripDuration int,
```

Compare the performance of Spark SQL tables on object store (swift) versus bdfs versus hdfs

```
%spark

val swift_df=spark.sql("select * from bike_trips_objectstore")
val bdfs_df=spark.sql("select * from bike_trips_objectstore_bdfs")
val hdfs_df=spark.sql("select * from bike_trips")

{
var t0 = System.nanoTime()
println("# of rows: %s".format(
  swift_df.count()
))
var t1 = System.nanoTime()
println("Swift Count Elapsed time: " + (t1 - t0)/1000000000 + "s")
println("..")

t0 = System.nanoTime()
println("# of rows: %s".format(
  bdfs_df.count()
))
t1 = System.nanoTime()
println("BDFS Count Elapsed time: " + (t1 - t0)/1000000000 + "s")
println("..")

t0 = System.nanoTime()
println("# of rows: %s".format(
  hdfs_df.count()
))
t1 = System.nanoTime()
println("HDFS Count Elapsed time: " + (t1 - t0)/1000000000 + "s")
println("..")

}
```

```
swift_df: org.apache.spark.sql.DataFrame = [tripduration: int, starttime: timestamp ... 13 more fields]
bdfs_df: org.apache.spark.sql.DataFrame = [tripduration: int, starttime: timestamp ... 13 more fields]
hdfs_df: org.apache.spark.sql.DataFrame = [tripduration: int, starttime: timestamp ... 13 more fields]
# of rows: 812192
Swift Count Elapsed time: 5s
..
# of rows: 812192
BDFS Count Elapsed time: 0s
..
# of rows: 812192
HDFS Count Elapsed time: 3s
..
```

View the Alluxio Web UI (port 19999)

The suggested way is to ssh into BDC and tunnel port 19999. Then point your local browser to http://127.0.0.1:19999/configuration (http://127.0.0.1:19999/configuration)

Example of using the Alluxio interpreter in zeppelin

```
%alluxio
help
```

```
Commands list:
        [help] - List all available commands.
        [cat <path>] - Prints the file's contents to the console.
        [chgrp [-R] <group> <path>] - Changes the group of a file or directory specified by args. Specify -R to change the group recursively.
        [chmod -R <mode> <path>] - Changes the permission of a file or directory specified by args. Specify -R to change the permission recursively.
        [chown -R <owner> <path>] - Changes the owner of a file or directory specified by args. Specify -R to change the owner recursively.
        [copyFromLocal <src> <remoteDst>] - Copies a file or a directory from local filesystem to Alluxio filesystem.
        [copyToLocal <src> <localDst>] - Copies a file or a directory from the Alluxio filesystem to the local filesystem.
        [count <path>] - Displays the number of files and directories matching the specified prefix.
        [createLineage <inputFile1,...> <outputFile1,...> [<cmd_arg1> <cmd_arg2> ...]] - Creates a lineage.
        [deleteLineage <lineageId> <cascade(true|false)>] - Deletes a lineage. If cascade is specified as true, dependent lineages will also be deleted.
        [du <path>] - Displays the size of the specified file or directory.
        [fileInfo <path>] - Displays all block info for the specified file.
        [free <file path|folder path>] - Removes the file or directory(recursively) from Alluxio memory space.
        [getCapacityBytes] - Gets the capacity of the Alluxio file system.
        [getUsedBytes] - Gets number of bytes used in the Alluxio file system.
        [listLineages] - Lists all lineages.
        [load <path>] - Loads a file or directory in Alluxio space, makes it resident in memory.
        [loadMetadata <path>] - Loads metadata for the given Alluxio path from the under file system.
        [location <path>] - Displays the list of hosts storing the specified file.
        [ls [-R] <path>] - Displays information for all files and directories directly under the specified path. Specify -R to display files and directories recursively.
        [mkdir <path1> [path2] ... [pathn]] - Creates the specified directories, including any parent directories that are required.
        [mount <alluxioPath> <ufsURI>] - Mounts a UFS path onto an Alluxio path.
        [mv <src> <dst>] - Renames a file or directory.
        [persist <alluxioPath>] - Persists a file or directory currently stored only in Alluxio to the UnderFileSystem.
        [pin <path>] - Pins the given file or directory in memory (works recursively for directories). Pinned files are never evicted from memory, unless TTL is set.
        [report <path>] - Reports to the master that a file is lost.
        [rm [-R] <path>] - Removes the specified file. Specify -R to remove file or directory recursively.
```

```
	[setTtl <path> <time to live(in milliseconds)>] - Sets a new TTL value for the file at path.
	[tail <path>] - Prints the file's last 1KB of contents to the console.
	[touch <path>] - Creates a 0 byte file. The file will be written to the under file system.
	[unmount <alluxioPath>] - Unmounts an Alluxio path.
	[unpin <path>] - Unpins the given file or folder from memory (works recursively for a directory).
\t[unsetTtl <path>] - Unsets the TTL value for the given path.
	[unpin <path>] - Unpin the given file to allow Alluxio to evict this file again. If the given path is a directory, it recursively unpins all files contained and any new files created within this directory.
```

```
%alluxio
ls -R /citibike
```

```
1.00B     02-03-2018 15:54:03:180  Directory      /citibike/raw
130.33MB  02-03-2018 15:54:03:181  In Memory      /citibike/raw/201612-citibike-tripdata.csv
1.00B     02-03-2018 18:51:31:613  Directory      /citibike/modified
130.33MB  02-03-2018 18:51:31:630  In Memory      /citibike/modified/201612-citibike-tripdata.nh.csv
```

```
%alluxio
```