



ChubaoFS

Java SDK and Hadoop Plugin Design

Oct 13, 2020

# Framework

ChuboFS hadoop

<https://github.com/chubaofs/chubaofs-hadoop.git>

ChuboFS high level java sdk

ChuboFS low level java sdk

ChubaoFS libsdk



# Implemented hdfs functions

index	name	return	function
1	mkdirs	void	mkdirs(Path f, FsPermission permission)
2	create	FSDataOutputStream	create(Path f, boolean overwrite, int bufferSize, short replication, long blockSize)
3	create	FSDataOutputStream	createNonRecursive(Path f, boolean overwrite, int bufferSize, short replication, long blockSize, Progressable progress)
4	open	FSDataInputStream	open(Path f, int bufferSize)
5	append	FSDataOutputStream	append(Path f, int bufferSize)
6	appendFile	FSDataOutputStreamBuilder	appendFile(Path path)
7	truncate	boolean	truncate(Path f, long newLength)
8	close	void	close()
9	delete	boolean	delete(Path f, boolean recursive)
10	exists	boolean	exists(Path f)
11	rename	void	rename(Path src, Path dst)
12	getFileStatus	FileStatus	getFileStatus(Path f)
13	setOwner	void	setOwner(Path p, String username, String groupname)
14	setPermission	void	setPermission(Path p, FsPermission permission)
15	setTimes	void	setTimes(Path p, long mtime, long atime)
16	listStatus	FileStatus[]	listStatus(Path f)

# High-level-java-sdk

---

## **CFSCient**

*init() throws CFSException;*

*FileStorage openFileStorage(StorageConfig config) throws CFSException;*

## **FileStorage**

*boolean mkdirs(String path, int mode, int uid, int gid) throws CFSException;*

*CFSFile open(String path, int flags, int mode, int uid, int gid) throws CFSException;*

*void truncate(String path, long newLength) throws CFSException;*

*void close() throws CFSException;*

*void rmdir(String path, boolean recursive) throws CFSException;*

*void unlink(String path) throws CFSException;*

*void rename(String src, String dst) throws CFSException;*

## **CFSFile**

*void close() throws CFSException;*

*void flush() throws CFSException;*

*void write(byte[] buff, int buffOffset, int len) throws CFSException;*

*long read(byte[] buff, int buffOffset, int len) throws CFSException;*

*void seek(long offset) throws CFSException;*

*long getFileSize();*

*long getPosition();*

# Deploy Dependence

---

## One so:

libsdk.so

- From chubaofs/libsdk

## Two jars:

chubaofs-hadoop-xx-SNAPSHOT.jar

- From chubaofs-hadoop

libchubaofs-xx-SNAPSHOT.jar

- From chubaofs/java

## Two profiles:

core-site.xml

cfs-site.xml

# core-site.xml

---

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>cfs://172.16.236.1 :8080</value>
  </property>

  <property>
    <name>fs.AbstractFileSystem.cfs.impl</name>
    <value>org.apache.hadoop.fs.ChubaoAbstractFS</value>
  </property>

  <property>
    <name>fs.cfs.impl</name>
    <value>org.apache.hadoop.hdfs.ChubaoFileSystem</value>
  </property>
</configuration>
```

# cfs-site.xml

---

```
<configuration>
  <property>
    <name>cfs.master.addr</name>
    <value>172.16.236.1:8080, 172.16.236.2:8080, 172.16.236.3:8080</value>
  </property>

  <property>
    <name>cfs.volume.name</name>
    <value>cfstest</value>
  </property>

  <property>
    <name>cfs.volume.owner</name>
    <value>cfs</value>
  </property>

  <property>
    <name>cfs.libsdk.path</name>
    <value>/root/hadoop/hadoop-3.3.0/lib/cfs/libcfs.so</value>
  </property>
</configuration>
```

# How to deploy

---

## **HDFS Shell on ChubaoFS**

1. Put the two jars to `$HADOOP_HOME/share/hadoop/common`
2. Put the two profiles to `$HADOOP_HOME/etc/hadoop`

## **YARN on ChubaoFS**

1. Put the two jars to `$HADOOP_HOME/share/hadoop/common`
2. Put the two profiles to `$HADOOP_HOME/etc/hadoop`
3. Start the yarn servers.

## **HBase on ChubaoFS**

1. Put the two jars to `$HBASE_HOME/lib`.
2. Put the two profiles to `$HBASE_HOME/conf`.
3. Modify `conf/hbase-site.xml` `` `hbase.rootdir` `cfs://172.16.236.1:8080/hbase` ``
4. Start hbase。



# About tests

---

## **Chubaofs-hadoop**

- FSMainOperationsBaseTest
- FileSystemContractBaseTest

## **Mapreduce Benchmark**

- dfsio
- mrbench
- nnbench
- terasort

## **HBase Benchmark**

- randomWrite
- sequentialWrite
- randomRead
- sequentialRead
- checkAndDelete

Note: The above tests are done in a cluster of 4 dockers, each uses 4 cpus and 8GB memory.

**Thanks!**

**FES**