&

1FileLock

FilejavaFileLockFileChannelregion

FileLock

```
    FileChannel channel = new FileOutputStream(LOG_FILE_NAME, true).getChannel();
    //
    FileLock fileLock = channel.lock(0L, Long.MAX_VALUE, true);
    fileLock = channel.lock();
    //read write file
    fileLock.release();
```

FileLockFileSystemFileSystem

2zookeeper

zookeeperzkdataDir

```
private String createLockNode(ZkClient client, String path) throws LockException {
    //
    try {
        return client.createEphemeralSequential(path, null);
    } catch (ZkNoNodeException e) {
        //
    }


    throw new LockException("Create ephemeral sequential node failed!");
}
```

```
//
String watchPath = getWatchPath(getSortedChildren(), currentNode);
final CountDownLatch latch = new CountDownLatch(1);
//
final IZkDataListener previousNodeListener = new IZkDataListener() {
    public void handleDataDeleted(String dataPath) throws Exception {
        latch.countDown();//
    }

    public void handleDataChange(String dataPath, Object data) throws Exception {
        //do nothing
    }
};


zkClient.subscribeDataChanges(watchPath, previousNodeListener);//
```

zookeeperzkzk

3

zookeeper

```
private String createLockFile() {
    int seq = 0;
    //rr
    if (!rr.exist(lockFilesPath)) {
        rr.createDirectory(lockFilesPath);
    }

    List<String> lockFiles = listLockFiles(lockFilesPath);
    try {

        if (lockFiles.size() > 0) {
            String last = lockFiles.get(lockFiles.size() - 1);
            seq = getSeq(last) + 1;
        }

        String lockFileName = lockFilesPath + lockPrefix + seq;
        if (rr.create(lockFileName)) {
            return lockFileName;
        }
    } catch (IOException e) {
        e.printStackTrace();
    }


    return null;
}
```

1WEB-INF/reportlets/WorkBook13.cpt WEB-INF/reportlets/WorkBook13.cpt/

2write-lock-01read-lock-02 read-lock-03

3

1jvm

2

4Redis

zookeeperrediszkredisSETNXSET if Not Exist

1) setnx keykey1keykey0

2setnxkv

3