package com.twitter.search.earlybird.archive.segmentbuilder;

import java.util.concurrent.atomic.AtomicBoolean;

import com.google.common.annotations.VisibleForTesting;

import com.twitter.common.base.Command;

import com.twitter.search.common.util.zktrylock.TryLock;

import com.twitter.search.earlybird.archive.ArchiveHDFSUtils;

import com.twitter.search.earlybird.index.EarlybirdSegmentFactory;

import com.twitter.search.earlybird.partition.SegmentInfo;

import com.twitter.search.earlybird.partition.SegmentSyncConfig;

public class SomeoneElseIsBuildingSegment extends SegmentBuilderSegment {

public SomeoneElseIsBuildingSegment(

SegmentInfo segmentInfo,

SegmentConfig segmentConfig,

EarlybirdSegmentFactory earlybirdSegmentFactory,

int alreadyRetriedCount,

SegmentSyncConfig sync) {

super(segmentInfo, segmentConfig, earlybirdSegmentFactory, alreadyRetriedCount, sync);

}

/\*\*

\* This method refreshes local state of a segment.

\* 1. Try to grab the ZK lock

\* 2a. if got the lock, the segment is not being built; mark segment as NOT\_BUILT\_YET.

\* 2b. otherwise, the segment is being built; keep the SOMEONE\_ELSE\_IS\_BUILDING state

\*/

@Override

public SegmentBuilderSegment handle()

throws SegmentInfoConstructionException, SegmentUpdaterException {

TryLock lock = getZooKeeperTryLock();

final AtomicBoolean alreadyBuilt = new AtomicBoolean(false);

boolean gotLock = lock.tryWithLock((Command) () -> {

// The segment might have already finished built by others

if (segmentExistsOnHdfs()) {

alreadyBuilt.set(true);

}

});

if (!gotLock) {

return this;

}

if (alreadyBuilt.get()) {

return new BuiltAndFinalizedSegment(

segmentInfo, segmentConfig, earlybirdSegmentFactory, 0, sync);

} else {

// When a segment failed building, its state might not be clean. So, it is necessary to

// create a new SegmentInfo with a clean state

SegmentInfo newSegmentInfo = createNewSegmentInfo(segmentInfo);

return new NotYetBuiltSegment(

newSegmentInfo,

segmentConfig,

earlybirdSegmentFactory,

alreadyRetriedCount + 1,

sync);

}

}

@VisibleForTesting

boolean segmentExistsOnHdfs() {

return ArchiveHDFSUtils.hasSegmentIndicesOnHDFS(sync, segmentInfo);

}

}