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

import java.util.concurrent.atomic.AtomicBoolean;

import com.google.common.base.Stopwatch;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.common.util.Clock;

import com.twitter.search.common.util.GCUtil;

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

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

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

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

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

public class NotYetBuiltSegment extends SegmentBuilderSegment {

private static final Logger LOG = LoggerFactory.getLogger(NotYetBuiltSegment.class);

public NotYetBuiltSegment(

SegmentInfo segmentInfo,

SegmentConfig segmentConfig,

EarlybirdSegmentFactory earlybirdSegmentFactory,

int alreadyRetriedCount,

SegmentSyncConfig sync) {

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

}

/\*\*

\* 1. Grab the ZK lock for this segment.

\* 2a. if lock fails, another host is updating; return the SOMEONE\_ELSE\_IS\_BUILDING state.

\* 2b. if lock succeeds, check again if the updated segment exists on HDFS.

\* 3a. if so, just move on.

\* 3b. if not, update the segment.

\* In both cases, we need to check if the segment can now be marked as BUILT\_AND\_FINALIZED.

\*/

@Override

public SegmentBuilderSegment handle()

throws SegmentUpdaterException, SegmentInfoConstructionException {

LOG.info("Handling a not yet built segment: {}", this.getSegmentName());

Stopwatch stopwatch = Stopwatch.createStarted();

TryLock lock = getZooKeeperTryLock();

// The tryWithLock can only access variables from parent class that are final. However, we

// would like to pass the process() return value to the parent class. So here we use

// AtomicBoolean reference instead of Boolean.

final AtomicBoolean successRef = new AtomicBoolean(false);

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

ArchiveSegmentUpdater updater = new ArchiveSegmentUpdater(

segmentConfig.getTryLockFactory(),

sync,

segmentConfig.getEarlybirdIndexConfig(),

Clock.SYSTEM\_CLOCK);

boolean success = updater.updateSegment(segmentInfo);

successRef.set(success);

});

if (!gotLock) {

LOG.info("cannot acquire zookeeper lock for: " + segmentInfo);

return new SomeoneElseIsBuildingSegment(

segmentInfo,

segmentConfig,

earlybirdSegmentFactory,

alreadyRetriedCount,

sync);

}

// 1. we want to make sure the heap is clean right after building a segment so that it's ready

// for us to start allocations for a new segment

// — I think we've had cases where we were seeing OOM's while building

// 2. the thing that I think it helps with is compaction (vs just organically running CMS)

// — which would clean up the heap, but may leave it in a fragmented state

// — and running a Full GC is supposed to compact the remaining tenured space.

GCUtil.runGC();

if (successRef.get()) {

LOG.info("Indexing segment {} took {}", segmentInfo, stopwatch);

LOG.info("Finished building {}", segmentInfo.getSegment().getSegmentName());

return new BuiltAndFinalizedSegment(

segmentInfo, segmentConfig, earlybirdSegmentFactory, 0, sync);

} else {

int alreadyTried = alreadyRetriedCount + 1;

String errMsg = "failed updating segment for: " + segmentInfo

+ " for " + alreadyTried + " times";

LOG.error(errMsg);

if (alreadyTried < segmentConfig.getMaxRetriesOnFailure()) {

return new NotYetBuiltSegment(

createNewSegmentInfo(segmentInfo),

segmentConfig,

earlybirdSegmentFactory,

alreadyTried,

sync);

} else {

throw new SegmentUpdaterException(errMsg);

}

}

}

}