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

import java.io.IOException;

import com.google.common.base.Preconditions;

import com.twitter.common.quantity.Amount;

import com.twitter.common.quantity.Time;

import com.twitter.search.common.database.DatabaseConfig;

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

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

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

import com.twitter.search.earlybird.common.config.EarlybirdConfig;

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

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

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

public abstract class SegmentBuilderSegment {

protected final SegmentInfo segmentInfo;

protected final SegmentConfig segmentConfig;

protected final EarlybirdSegmentFactory earlybirdSegmentFactory;

protected final int alreadyRetriedCount;

protected final SegmentSyncConfig sync;

public SegmentBuilderSegment(SegmentInfo segmentInfo,

SegmentConfig segmentConfig,

EarlybirdSegmentFactory earlybirdSegmentFactory,

int alreadyRetriedCount,

SegmentSyncConfig segmentSyncConfig) {

this.segmentConfig = segmentConfig;

this.earlybirdSegmentFactory = earlybirdSegmentFactory;

this.alreadyRetriedCount = alreadyRetriedCount;

this.sync = segmentSyncConfig;

Preconditions.checkState(segmentInfo.getSegment() instanceof ArchiveSegment);

this.segmentInfo = Preconditions.checkNotNull(segmentInfo);

}

public SegmentInfo getSegmentInfo() {

return segmentInfo;

}

public String getSegmentName() {

return segmentInfo.getSegmentName();

}

public int getAlreadyRetriedCount() {

return alreadyRetriedCount;

}

/\*\*

\* Handle the segment, potentially transitioning to a new state.

\* @return The state after handling.

\*/

public abstract SegmentBuilderSegment handle()

throws SegmentInfoConstructionException, SegmentUpdaterException;

public boolean isBuilt() {

return false;

}

@Override

public String toString() {

return "SegmentBuilderSegment{"

+ "segmentInfo=" + segmentInfo

+ ", state=" + this.getClass().getSimpleName()

+ ", alreadyRetriedCount=" + alreadyRetriedCount + '}';

}

/\*\*

\* Given a SegmentInfo, create a new one with the same time slice and partitionID but clean

\* internal state.

\*/

protected SegmentInfo createNewSegmentInfo(SegmentInfo oldSegmentInfo)

throws SegmentInfoConstructionException {

Preconditions.checkArgument(oldSegmentInfo.getSegment() instanceof ArchiveSegment);

ArchiveSegment archiveSegment = (ArchiveSegment) oldSegmentInfo.getSegment();

try {

ArchiveSegment segment = new ArchiveSegment(archiveSegment.getArchiveTimeSlice(),

archiveSegment.getHashPartitionID(), EarlybirdConfig.getMaxSegmentSize());

return new SegmentInfo(segment, earlybirdSegmentFactory, sync);

} catch (IOException e) {

throw new SegmentInfoConstructionException("Error creating new segments", e);

}

}

protected TryLock getZooKeeperTryLock() {

ZooKeeperTryLockFactory tryLockFactory = segmentConfig.getTryLockFactory();

String zkRootPath = sync.getZooKeeperSyncFullPath();

String nodeName = segmentInfo.getZkNodeName();

Amount<Long, Time> expirationTime = segmentConfig.getSegmentZKLockExpirationTime();

return tryLockFactory.createTryLock(

DatabaseConfig.getLocalHostname(),

zkRootPath,

nodeName,

expirationTime);

}

}