package com.twitter.search.earlybird.archive;

import java.io.IOException;

import java.util.concurrent.ConcurrentHashMap;

import com.google.common.base.Preconditions;

import org.apache.lucene.index.IndexWriterConfig;

import org.apache.lucene.index.KeepOnlyLastCommitDeletionPolicy;

import org.apache.lucene.index.LogByteSizeMergePolicy;

import org.apache.lucene.index.SerialMergeScheduler;

import com.twitter.decider.Decider;

import com.twitter.search.common.schema.SearchWhitespaceAnalyzer;

import com.twitter.search.common.schema.earlybird.EarlybirdCluster;

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

import com.twitter.search.core.earlybird.index.EarlybirdIndexSegmentData;

import com.twitter.search.core.earlybird.index.EarlybirdLuceneIndexSegmentData;

import com.twitter.search.earlybird.EarlybirdIndexConfig;

import com.twitter.search.earlybird.exception.CriticalExceptionHandler;

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

/\*\*

\* Base config for the top archive tweet clusters.

\*/

public abstract class ArchiveEarlybirdIndexConfig extends EarlybirdIndexConfig {

private final CloseResourceUtil resourceCloser = new CloseResourceUtil();

public ArchiveEarlybirdIndexConfig(

EarlybirdCluster cluster, Decider decider, SearchIndexingMetricSet searchIndexingMetricSet,

CriticalExceptionHandler criticalExceptionHandler) {

super(cluster, decider, searchIndexingMetricSet, criticalExceptionHandler);

}

@Override

public IndexWriterConfig newIndexWriterConfig() {

return new IndexWriterConfig(new SearchWhitespaceAnalyzer())

.setIndexDeletionPolicy(new KeepOnlyLastCommitDeletionPolicy())

.setMergeScheduler(new SerialMergeScheduler())

.setMergePolicy(new LogByteSizeMergePolicy())

.setRAMBufferSizeMB(IndexWriterConfig.DEFAULT\_RAM\_PER\_THREAD\_HARD\_LIMIT\_MB)

.setMaxBufferedDocs(IndexWriterConfig.DISABLE\_AUTO\_FLUSH)

.setOpenMode(IndexWriterConfig.OpenMode.CREATE\_OR\_APPEND);

}

@Override

public CloseResourceUtil getResourceCloser() {

return resourceCloser;

}

@Override

public EarlybirdIndexSegmentData optimize(

EarlybirdIndexSegmentData segmentData) throws IOException {

Preconditions.checkArgument(

segmentData instanceof EarlybirdLuceneIndexSegmentData,

"Expected EarlybirdLuceneIndexSegmentData but got %s",

segmentData.getClass());

EarlybirdLuceneIndexSegmentData data = (EarlybirdLuceneIndexSegmentData) segmentData;

return new EarlybirdLuceneIndexSegmentData(

data.getLuceneDirectory(),

data.getMaxSegmentSize(),

data.getTimeSliceID(),

data.getSchema(),

true, // isOptimized

data.getSyncData().getSmallestDocID(),

new ConcurrentHashMap<>(data.getPerFieldMap()),

data.getFacetCountingArray(),

data.getDocValuesManager(),

data.getDocIDToTweetIDMapper(),

data.getTimeMapper(),

data.getIndexExtensionsData());

}

}