package com.twitter.search.earlybird.partition;

import java.io.IOException;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.search.earlybird.EarlybirdStatus;

public final class SegmentOptimizer {

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

private static final String OPTIMIZING\_SEGMENT\_EVENT\_PATTERN = "optimizing segment %s";

private static final String OPTIMIZING\_SEGMENT\_GAUGE\_PATTERN = "optimizing\_segment\_%s";

private SegmentOptimizer() {

}

/\*\*

\* Optimize a segment. Returns whether optimization was successful.

\*/

public static boolean optimize(SegmentInfo segmentInfo) {

try {

return optimizeThrowing(segmentInfo);

} catch (Exception e) {

// This is a bad situation, as earlybird can't run with too many un-optimized

// segments in memory.

LOG.error("Exception while optimizing segment " + segmentInfo.getSegmentName() + ": ", e);

segmentInfo.setFailedOptimize();

return false;

}

}

public static boolean needsOptimization(SegmentInfo segmentInfo) {

return segmentInfo.isComplete() && !segmentInfo.isOptimized()

&& !segmentInfo.isFailedOptimize() && !segmentInfo.isIndexing();

}

private static boolean optimizeThrowing(SegmentInfo segmentInfo) throws IOException {

if (!needsOptimization(segmentInfo)) {

return false;

}

String gaugeName =

String.format(OPTIMIZING\_SEGMENT\_GAUGE\_PATTERN, segmentInfo.getSegmentName());

SearchIndexingMetricSet.StartupMetric metric =

new SearchIndexingMetricSet.StartupMetric(gaugeName);

String eventName =

String.format(OPTIMIZING\_SEGMENT\_EVENT\_PATTERN, segmentInfo.getSegmentName());

EarlybirdStatus.beginEvent(eventName, metric);

try {

segmentInfo.getIndexSegment().optimizeIndexes();

} finally {

EarlybirdStatus.endEvent(eventName, metric);

}

return true;

}

}