package com.twitter.search.earlybird.archive;

import java.io.IOException;

import java.util.List;

import com.google.common.annotations.VisibleForTesting;

import org.apache.lucene.index.DirectoryReader;

import org.apache.lucene.index.LeafReader;

import org.apache.lucene.index.LeafReaderContext;

import org.apache.lucene.store.Directory;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

public final class ArchiveSegmentVerifier {

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

private ArchiveSegmentVerifier() {

}

@VisibleForTesting

static boolean shouldVerifySegment(SegmentInfo segmentInfo) {

if (segmentInfo.isIndexing()) {

LOG.warn("ArchiveSegmentVerifier got segment still indexing.");

return false;

}

if (!segmentInfo.isComplete()) {

LOG.warn("ArchiveSegmentVerifyer got incomplete segment.");

return false;

}

if (!segmentInfo.isOptimized()) {

LOG.warn("ArchiveSegmentVerifyer got unoptimized segment.");

return false;

}

return true;

}

/\*\*

\* Verifies an archive segment has a sane number of leaves.

\*/

public static boolean verifySegment(SegmentInfo segmentInfo) {

if (!shouldVerifySegment(segmentInfo)) {

return false;

}

Directory directory = segmentInfo.getIndexSegment().getLuceneDirectory();

return verifyLuceneIndex(directory);

}

private static boolean verifyLuceneIndex(Directory directory) {

try {

DirectoryReader indexerReader = DirectoryReader.open(directory);

List<LeafReaderContext> leaves = indexerReader.getContext().leaves();

if (leaves.size() != 1) {

LOG.warn("Lucene index does not have exactly one segment: " + leaves.size() + " != 1. "

+ "Lucene segments should have been merged during optimization.");

return false;

}

LeafReader reader = leaves.get(0).reader();

if (reader.numDocs() <= 0) {

LOG.warn("Lucene index has no document: " + reader);

return false;

}

return true;

} catch (IOException e) {

LOG.warn("Found bad lucene index at: " + directory);

return false;

}

}

}