package com.twitter.search.earlybird.partition;

import java.util.Optional;

import java.util.concurrent.atomic.AtomicInteger;

import java.util.concurrent.atomic.AtomicLong;

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

public class SegmentIndexStats {

private EarlybirdIndexSegmentData segmentData;

private final AtomicLong indexSizeOnDiskInBytes = new AtomicLong(0);

private final AtomicInteger partialUpdateCount = new AtomicInteger(0);

private final AtomicInteger outOfOrderUpdateCount = new AtomicInteger(0);

private Optional<Integer> savedStatusCount = Optional.empty();

private Optional<Integer> savedDeletesCount = Optional.empty();

public void setSegmentData(EarlybirdIndexSegmentData segmentData) {

this.segmentData = segmentData;

}

/\*\*

\* We'd like to be able to return the last counts after we unload a segment from memory.

\*/

public void unsetSegmentDataAndSaveCounts() {

savedStatusCount = Optional.of(getStatusCount());

savedDeletesCount = Optional.of(getDeleteCount());

segmentData = null;

}

/\*\*

\* Returns the number of deletes processed by this segment.

\*/

public int getDeleteCount() {

if (segmentData != null) {

return segmentData.getDeletedDocs().numDeletions();

} else {

return savedDeletesCount.orElse(0);

}

}

/\*\*

\* Return the number of documents in this segment.

\*/

public int getStatusCount() {

if (segmentData != null) {

return segmentData.numDocs();

} else {

return savedStatusCount.orElse(0);

}

}

public long getIndexSizeOnDiskInBytes() {

return indexSizeOnDiskInBytes.get();

}

public void setIndexSizeOnDiskInBytes(long value) {

indexSizeOnDiskInBytes.set(value);

}

public int getPartialUpdateCount() {

return partialUpdateCount.get();

}

public void incrementPartialUpdateCount() {

partialUpdateCount.incrementAndGet();

}

public void setPartialUpdateCount(int value) {

partialUpdateCount.set(value);

}

public int getOutOfOrderUpdateCount() {

return outOfOrderUpdateCount.get();

}

public void incrementOutOfOrderUpdateCount() {

outOfOrderUpdateCount.incrementAndGet();

}

public void setOutOfOrderUpdateCount(int value) {

outOfOrderUpdateCount.set(value);

}

@Override

public String toString() {

StringBuilder sb = new StringBuilder();

sb.append("Indexed ").append(getStatusCount()).append(" documents, ");

sb.append(getDeleteCount()).append(" deletes, ");

sb.append(getPartialUpdateCount()).append(" partial updates, ");

sb.append(getOutOfOrderUpdateCount()).append(" out of order udpates. ");

sb.append("Index size: ").append(getIndexSizeOnDiskInBytes());

return sb.toString();

}

}