package com.twitter.search.earlybird.partition;

import com.google.common.annotations.VisibleForTesting;

import com.twitter.search.common.partitioning.base.Segment;

/\*\*

\* Representation for segment sync state, the local and hdfs file locations, as well as the

\* current in-memory sync states maintained by earlybirds.

\*/

public class SegmentSyncInfo {

// Is this segment loaded from disk?

private volatile boolean loaded = false;

// Has this segment been flushed to disk, and uploaded to HDFS if uploading is enabled?

private volatile boolean flushed = false;

// Time when the segment was flushed to local disk

private volatile long flushTimeMillis = 0;

private final Segment segment;

private final SegmentSyncConfig syncConfig;

private final String localSyncDir;

private final String hdfsFlushDir;

private final String hdfsSyncDirPrefix;

private final String hdfsUploadDirPrefix;

private final String hdfsTempFlushDir;

@VisibleForTesting

public SegmentSyncInfo(SegmentSyncConfig syncConfig, Segment segment) {

this.segment = segment;

this.syncConfig = syncConfig;

this.localSyncDir = syncConfig.getLocalSyncDirName(segment);

this.hdfsSyncDirPrefix = syncConfig.getHdfsSyncDirNamePrefix(segment);

this.hdfsUploadDirPrefix = syncConfig.getHdfsUploadDirNamePrefix(segment);

this.hdfsFlushDir = syncConfig.getHdfsFlushDirName(segment);

this.hdfsTempFlushDir = syncConfig.getHdfsTempFlushDirName(segment);

}

public boolean isLoaded() {

return loaded;

}

public boolean isFlushed() {

return flushed;

}

public long getFlushTimeMillis() {

return flushTimeMillis;

}

public String getLocalSyncDir() {

return localSyncDir;

}

public SegmentSyncConfig getSegmentSyncConfig() {

return syncConfig;

}

public String getLocalLuceneSyncDir() {

// For archive search this name depends on the end date of the segment, which can change,

// so we cannot pre-compute this in the constructor.

// This should only be used in the on-disk archive.

return syncConfig.getLocalLuceneSyncDirName(segment);

}

public String getHdfsFlushDir() {

return hdfsFlushDir;

}

public String getHdfsSyncDirPrefix() {

return hdfsSyncDirPrefix;

}

public String getHdfsUploadDirPrefix() {

return hdfsUploadDirPrefix;

}

public String getHdfsTempFlushDir() {

return hdfsTempFlushDir;

}

public void setLoaded(boolean isLoaded) {

this.loaded = isLoaded;

}

/\*\*

\* Stores the flushing state for this segment.

\*/

public void setFlushed(boolean isFlushed) {

if (isFlushed) {

this.flushTimeMillis = System.currentTimeMillis();

}

this.flushed = isFlushed;

}

/\*\*

\* Adds debug information about the loaded and flushed status of this segment to the given

\* StringBuilder.

\*/

public void addDebugInfo(StringBuilder builder) {

builder.append("[");

int startLength = builder.length();

if (loaded) {

builder.append("loaded, ");

}

if (flushed) {

builder.append("flushed, ");

}

if (startLength < builder.length()) {

builder.setLength(builder.length() - 2);

}

builder.append("]");

}

}