package com.twitter.search.earlybird.archive;

import java.io.File;

import java.io.IOException;

import org.apache.lucene.store.Directory;

import org.apache.lucene.store.FSDirectory;

import com.twitter.decider.Decider;

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

import com.twitter.search.common.util.io.flushable.DataDeserializer;

import com.twitter.search.common.util.io.flushable.FlushInfo;

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

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

import com.twitter.search.core.earlybird.index.extensions.EarlybirdIndexExtensionsFactory;

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

import com.twitter.search.earlybird.index.DocValuesBasedTimeMapper;

import com.twitter.search.earlybird.index.DocValuesBasedTweetIDMapper;

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

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

/\*\*

\* Index config for the on-disk Tweet clusters.

\*/

public class ArchiveOnDiskEarlybirdIndexConfig extends ArchiveEarlybirdIndexConfig {

public ArchiveOnDiskEarlybirdIndexConfig(

Decider decider, SearchIndexingMetricSet searchIndexingMetricSet,

CriticalExceptionHandler criticalExceptionHandler) {

super(EarlybirdCluster.FULL\_ARCHIVE, decider, searchIndexingMetricSet,

criticalExceptionHandler);

}

@Override

public boolean isIndexStoredOnDisk() {

return true;

}

@Override

public Directory newLuceneDirectory(SegmentSyncInfo segmentSyncInfo) throws IOException {

File dirPath = new File(segmentSyncInfo.getLocalLuceneSyncDir());

return FSDirectory.open(dirPath.toPath());

}

@Override

public EarlybirdIndexSegmentData newSegmentData(

int maxSegmentSize,

long timeSliceID,

Directory dir,

EarlybirdIndexExtensionsFactory extensionsFactory) {

return new EarlybirdLuceneIndexSegmentData(

dir,

maxSegmentSize,

timeSliceID,

getSchema(),

new DocValuesBasedTweetIDMapper(),

new DocValuesBasedTimeMapper(),

extensionsFactory);

}

@Override

public EarlybirdIndexSegmentData loadSegmentData(

FlushInfo flushInfo,

DataDeserializer dataInputStream,

Directory dir,

EarlybirdIndexExtensionsFactory extensionsFactory) throws IOException {

// IO Exception will be thrown if there's an error during load

return (new EarlybirdLuceneIndexSegmentData.OnDiskSegmentDataFlushHandler(

getSchema(),

dir,

extensionsFactory,

new DocValuesBasedTweetIDMapper.FlushHandler(),

new DocValuesBasedTimeMapper.FlushHandler())).load(flushInfo, dataInputStream);

}

@Override

public boolean supportOutOfOrderIndexing() {

return false;

}

}