package com.twitter.search.earlybird.segment;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import java.util.Set;

import com.twitter.common.util.Clock;

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

import com.twitter.search.common.util.io.dl.DLReaderWriterFactory;

import com.twitter.search.common.util.io.dl.SegmentDLUtil;

import com.twitter.search.earlybird.EarlybirdIndexConfig;

import com.twitter.search.earlybird.common.config.EarlybirdConfig;

/\*\*

\* An implementation of SegmentDataProvider using DistributedLog.

\*/

public class DLSegmentDataProvider implements SegmentDataProvider {

private final int hashPartitionID;

private final DLReaderWriterFactory dlFactory;

private final SegmentDataReaderSet readerSet;

public DLSegmentDataProvider(

int hashPartitionID,

EarlybirdIndexConfig earlybirdIndexConfig,

DLReaderWriterFactory dlReaderWriterFactory) throws IOException {

this(hashPartitionID, earlybirdIndexConfig, dlReaderWriterFactory,

Clock.SYSTEM\_CLOCK);

}

public DLSegmentDataProvider(

int hashPartitionID,

EarlybirdIndexConfig earlybirdIndexConfig,

DLReaderWriterFactory dlReaderWriterFactory,

Clock clock) throws IOException {

this.hashPartitionID = hashPartitionID;

this.dlFactory = dlReaderWriterFactory;

this.readerSet = new DLSegmentDataReaderSet(

dlFactory,

earlybirdIndexConfig,

clock);

}

@Override

public SegmentDataReaderSet getSegmentDataReaderSet() {

return readerSet;

}

@Override

public List<Segment> newSegmentList() throws IOException {

Set<String> segmentNames = SegmentDLUtil.getSegmentNames(dlFactory, null, hashPartitionID);

List<Segment> segmentList = new ArrayList<>(segmentNames.size());

for (String segmentName : segmentNames) {

Segment segment = Segment.fromSegmentName(segmentName, EarlybirdConfig.getMaxSegmentSize());

segmentList.add(segment);

}

// Sort the segments by ID.

Collections.sort(segmentList);

return segmentList;

}

}