package com.twitter.search.core.earlybird.index.column;

import java.io.IOException;

import com.google.common.base.Preconditions;

import org.apache.lucene.index.LeafReader;

import org.apache.lucene.index.NumericDocValues;

import com.twitter.search.core.earlybird.index.util.AllDocsIterator;

/\*\*

\* A NumericDocValues implementation that uses an AllDocsIterator to iterate through all docs, and

\* gets its values from a ColumnStrideFieldIndex instance.

\*/

public class ColumnStrideFieldDocValues extends NumericDocValues {

private final ColumnStrideFieldIndex csf;

private final AllDocsIterator iterator;

public ColumnStrideFieldDocValues(ColumnStrideFieldIndex csf, LeafReader reader)

throws IOException {

this.csf = Preconditions.checkNotNull(csf);

this.iterator = new AllDocsIterator(Preconditions.checkNotNull(reader));

}

@Override

public long longValue() {

return csf.get(docID());

}

@Override

public int docID() {

return iterator.docID();

}

@Override

public int nextDoc() throws IOException {

return iterator.nextDoc();

}

@Override

public int advance(int target) throws IOException {

return iterator.advance(target);

}

@Override

public boolean advanceExact(int target) throws IOException {

// The javadocs for advance() and advanceExact() are inconsistent. advance() allows the target

// to be smaller than the current doc ID, and requires the iterator to advance the current doc

// ID past the target, and past the current doc ID. So essentially, advance(target) returns

// max(target, currentDocId + 1). At the same time, advanceExact() is undefined if the target is

// smaller than the current do ID (or if it's an invalid doc ID), and always returns the target.

// So essentially, advanceExact(target) should always set the current doc ID to the given target

// and if target == currentDocId, then currentDocId should not be advanced. This is why we have

// these extra checks here instead of moving them to advance().

Preconditions.checkState(

target >= docID(),

"ColumnStrideFieldDocValues.advance() for field %s called with target %s, "

+ "but the current doc ID is %s.",

csf.getName(),

target,

docID());

if (target == docID()) {

return true;

}

// We don't need to check if we have a value for 'target', because a ColumnStrideFieldIndex

// instance has a value for every doc ID (though that value might be 0).

return advance(target) == target;

}

@Override

public long cost() {

return iterator.cost();

}

}