package com.twitter.search.earlybird.search.queries;

import java.io.IOException;

import java.util.Objects;

import java.util.Set;

import org.apache.lucene.index.LeafReader;

import org.apache.lucene.index.LeafReaderContext;

import org.apache.lucene.index.NumericDocValues;

import org.apache.lucene.search.BooleanClause;

import org.apache.lucene.search.BooleanQuery;

import org.apache.lucene.search.DocIdSetIterator;

import org.apache.lucene.search.IndexSearcher;

import org.apache.lucene.search.Query;

import org.apache.lucene.search.ScoreMode;

import org.apache.lucene.search.Weight;

import com.twitter.search.common.query.DefaultFilterWeight;

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

/\*\*

\* CSFDisjunctionFilter provides an efficient mechanism to query for documents that have a

\* long CSF equal to one of the provided values.

\*/

public final class CSFDisjunctionFilter extends Query {

private final String csfField;

private final Set<Long> values;

public static Query getCSFDisjunctionFilter(String csfField, Set<Long> values) {

return new BooleanQuery.Builder()

.add(new CSFDisjunctionFilter(csfField, values), BooleanClause.Occur.FILTER)

.build();

}

private CSFDisjunctionFilter(String csfField, Set<Long> values) {

this.csfField = csfField;

this.values = values;

}

@Override

public Weight createWeight(IndexSearcher searcher, ScoreMode scoreMode, float boost) {

return new DefaultFilterWeight(this) {

@Override

protected DocIdSetIterator getDocIdSetIterator(LeafReaderContext context) throws IOException {

return new CSFDisjunctionFilterDISI(context.reader(), csfField, values);

}

};

}

@Override

public int hashCode() {

return (csfField == null ? 0 : csfField.hashCode()) \* 17

+ (values == null ? 0 : values.hashCode());

}

@Override

public boolean equals(Object obj) {

if (!(obj instanceof CSFDisjunctionFilter)) {

return false;

}

CSFDisjunctionFilter filter = CSFDisjunctionFilter.class.cast(obj);

return Objects.equals(csfField, filter.csfField) && Objects.equals(values, filter.values);

}

@Override

public String toString(String field) {

return "CSFDisjunctionFilter:" + csfField + ",count:" + values.size();

}

private static final class CSFDisjunctionFilterDISI extends RangeFilterDISI {

private final NumericDocValues docValues;

private final Set<Long> values;

private CSFDisjunctionFilterDISI(LeafReader reader, String csfField, Set<Long> values)

throws IOException {

super(reader);

this.values = values;

this.docValues = reader.getNumericDocValues(csfField);

}

@Override

protected boolean shouldReturnDoc() throws IOException {

return docValues.advanceExact(docID()) && values.contains(docValues.longValue());

}

}

}