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

import org.apache.lucene.util.CloseableThreadLocal;

import com.twitter.search.common.search.QueryCostProvider;

public class QueryCostTracker implements QueryCostProvider {

public static enum CostType {

// For the realtime segment we track how many posting list blocks

// are accessed during the lifetime of one query.

LOAD\_REALTIME\_POSTING\_BLOCK(1),

// Number of optimized posting list blocks

LOAD\_OPTIMIZED\_POSTING\_BLOCK(1);

private final double cost;

private CostType(double cost) {

this.cost = cost;

}

}

private static final CloseableThreadLocal<QueryCostTracker> TRACKERS

= new CloseableThreadLocal<QueryCostTracker>() {

@Override protected QueryCostTracker initialValue() {

return new QueryCostTracker();

}

};

public static QueryCostTracker getTracker() {

return TRACKERS.get();

}

private double totalCost;

public void track(CostType costType) {

totalCost += costType.cost;

}

public void reset() {

totalCost = 0;

}

@Override

public double getTotalCost() {

return totalCost;

}

}