package com.twitter.search.earlybird\_root.mergers;

import java.util.ArrayList;

import java.util.List;

import com.twitter.search.earlybird.thrift.EarlybirdResponse;

import com.twitter.search.earlybird.thrift.EarlybirdResponseCode;

import com.twitter.search.earlybird.thrift.TierResponse;

public final class TierResponseAccumulator extends ResponseAccumulator {

private static final String TARGET\_TYPE\_TIER = "tier";

private final List<TierResponse> tierResponses = new ArrayList<>();

// Total number of partitions the request was sent to, across all tiers.

private int totalPartitionsQueriedInAllTiers = 0;

// Among the above partitions, the number of them that returned successful responses.

private int totalSuccessfulPartitionsInAllTiers = 0;

@Override

public String getNameForLogging(int responseIndex, int numTotalResponses) {

return TARGET\_TYPE\_TIER + (numTotalResponses - responseIndex);

}

@Override

public String getNameForEarlybirdResponseCodeStats(int responseIndex, int numTotalResponses) {

return TARGET\_TYPE\_TIER + (numTotalResponses - responseIndex);

}

@Override

protected boolean isMergingAcrossTiers() {

return true;

}

@Override

public boolean shouldEarlyTerminateMerge(EarlyTerminateTierMergePredicate merger) {

if (foundError()) {

return true;

}

int numResults = 0;

for (EarlybirdResponse resp : getSuccessResponses()) {

if (resp.isSetSearchResults()) {

numResults += resp.getSearchResults().getResultsSize();

}

}

return merger.shouldEarlyTerminateTierMerge(numResults, foundEarlyTermination());

}

@Override

public void handleSkippedResponse(EarlybirdResponseCode responseCode) {

tierResponses.add(new TierResponse()

.setNumPartitions(0)

.setNumSuccessfulPartitions(0)

.setTierResponseCode(responseCode));

}

@Override

public void handleErrorResponse(EarlybirdResponse response) {

// TierResponse, which is only returned if merging results from different tiers.

TierResponse tr = new TierResponse();

if (response != null) {

if (response.isSetResponseCode()) {

tr.setTierResponseCode(response.getResponseCode());

} else {

tr.setTierResponseCode(EarlybirdResponseCode.TRANSIENT\_ERROR);

}

tr.setNumPartitions(response.getNumPartitions());

tr.setNumSuccessfulPartitions(0);

totalPartitionsQueriedInAllTiers += response.getNumPartitions();

} else {

tr.setTierResponseCode(EarlybirdResponseCode.TRANSIENT\_ERROR)

.setNumPartitions(0)

.setNumSuccessfulPartitions(0);

}

tierResponses.add(tr);

}

@Override

public AccumulatedResponses.PartitionCounts getPartitionCounts() {

return new AccumulatedResponses.PartitionCounts(totalPartitionsQueriedInAllTiers,

totalSuccessfulPartitionsInAllTiers, tierResponses);

}

@Override

public void extraSuccessfulResponseHandler(EarlybirdResponse response) {

// Record tier stats.

totalPartitionsQueriedInAllTiers += response.getNumPartitions();

totalSuccessfulPartitionsInAllTiers += response.getNumSuccessfulPartitions();

tierResponses.add(new TierResponse()

.setNumPartitions(response.getNumPartitions())

.setNumSuccessfulPartitions(response.getNumSuccessfulPartitions())

.setTierResponseCode(EarlybirdResponseCode.SUCCESS));

}

}