package com.twitter.search.earlybird.stats;

import java.util.concurrent.TimeUnit;

import com.twitter.search.common.metrics.SearchCounter;

import com.twitter.search.common.metrics.SearchRateCounter;

import com.twitter.search.common.metrics.SearchRequestStats;

/\*\*

\* SearchRequestStats with earlybird-specific additional stats.

\*/

public final class EarlybirdRPCStats {

private final SearchRequestStats requestStats;

// Number of queries that were terminated early.

private final SearchCounter earlyTerminatedRequests;

// We do not count client error in the response error rate, but track it separately.

private final SearchRateCounter responseClientErrors;

public EarlybirdRPCStats(String name) {

requestStats = SearchRequestStats.export(name, TimeUnit.MICROSECONDS, true, true);

earlyTerminatedRequests = SearchCounter.export(name + "\_early\_terminated");

responseClientErrors = SearchRateCounter.export(name + "\_client\_error");

}

public long getRequestRate() {

return (long) (double) requestStats.getRequestRate().read();

}

public long getAverageLatency() {

return (long) (double) requestStats.getTimerStats().read();

}

/\*\*

\* Records a completed earlybird request.

\* @param latencyUs how long the request took to complete, in microseconds.

\* @param resultsCount how many results were returned.

\* @param success whether the request was successful or not.

\* @param earlyTerminated whether the request terminated early or not.

\* @param clientError whether the request failure is caused by client errors

\*/

public void requestComplete(long latencyUs, long resultsCount, boolean success,

boolean earlyTerminated, boolean clientError) {

// We treat client errors as successes for top-line metrics to prevent bad client requests (like

// malformed queries) from dropping our success rate and generating alerts.

requestStats.requestComplete(latencyUs, resultsCount, success || clientError);

if (earlyTerminated) {

earlyTerminatedRequests.increment();

}

if (clientError) {

responseClientErrors.increment();

}

}

}