package com.twitter.search.earlybird.stats;

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

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

public class SegmentSyncStats {

private static final String CPU\_TOTAL = "\_cpu\_total\_";

private static final String CPU\_USER = "\_cpu\_user\_mode\_";

private static final String CPU\_SYS = "\_cpu\_system\_mode\_";

private final SearchCounter segmentSyncLatency;

private final SearchCounter segmentSyncLatencyCpuTotal;

private final SearchCounter segmentSyncLatencyCpuUserMode;

private final SearchCounter segmentSyncLatencyCpuSystemMode;

private final SearchCounter segmentSyncCount;

private final SearchCounter segmentErrorCount;

private SegmentSyncStats(SearchCounter segmentSyncLatency,

SearchCounter segmentSyncLatencyCpuTotal,

SearchCounter segmentSyncLatencyCpuUserMode,

SearchCounter segmentSyncLatencyCpuSystemMode,

SearchCounter segmentSyncCount,

SearchCounter segmentErrorCount) {

this.segmentSyncLatency = segmentSyncLatency;

this.segmentSyncLatencyCpuTotal = segmentSyncLatencyCpuTotal;

this.segmentSyncLatencyCpuUserMode = segmentSyncLatencyCpuUserMode;

this.segmentSyncLatencyCpuSystemMode = segmentSyncLatencyCpuSystemMode;

this.segmentSyncCount = segmentSyncCount;

this.segmentErrorCount = segmentErrorCount;

}

/\*\*

\* Creates a new set of stats for the given segment sync action.

\* @param action the name to be used for the sync stats.

\*/

public SegmentSyncStats(String action) {

this(SearchCounter.export("segment\_" + action + "\_latency\_ms"),

SearchCounter.export("segment\_" + action + "\_latency" + CPU\_TOTAL + "ms"),

SearchCounter.export("segment\_" + action + "\_latency" + CPU\_USER + "ms"),

SearchCounter.export("segment\_" + action + "\_latency" + CPU\_SYS + "ms"),

SearchCounter.export("segment\_" + action + "\_count"),

SearchCounter.export("segment\_" + action + "\_error\_count"));

}

/\*\*

\* Records a completed action using the specified timer.

\*/

public void actionComplete(Timer timer) {

segmentSyncCount.increment();

segmentSyncLatency.add(timer.getElapsed());

segmentSyncLatencyCpuTotal.add(timer.getElapsedCpuTotal());

segmentSyncLatencyCpuUserMode.add(timer.getElapsedCpuUserMode());

segmentSyncLatencyCpuSystemMode.add(timer.getElapsedCpuSystemMode());

}

public void recordError() {

segmentErrorCount.increment();

}

}