package com.twitter.search.earlybird\_root;

import java.util.Map;

import java.util.Random;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.search.common.root.PartitionLoggingSupport;

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

import com.twitter.search.earlybird\_root.common.EarlybirdRequestContext;

public class EarlybirdServicePartitionLoggingSupport

extends PartitionLoggingSupport.DefaultPartitionLoggingSupport<EarlybirdRequestContext> {

private static final Logger PARTITION\_LOG = LoggerFactory.getLogger("partitionLogger");

private static final long LATENCY\_LOG\_PARTITIONS\_THRESHOLD\_MS = 500;

private static final double FRACTION\_OF\_REQUESTS\_TO\_LOG = 1.0 / 500.0;

private final Random random = new Random();

@Override

public void logPartitionLatencies(EarlybirdRequestContext requestContext,

String tierName,

Map<Integer, Long> partitionLatenciesMicros,

long latencyMs) {

String logReason = null;

if (random.nextFloat() <= FRACTION\_OF\_REQUESTS\_TO\_LOG) {

logReason = "randomSample";

} else if (latencyMs > LATENCY\_LOG\_PARTITIONS\_THRESHOLD\_MS) {

logReason = "slow";

}

EarlybirdRequest request = requestContext.getRequest();

if (logReason != null && request.isSetSearchQuery()) {

PARTITION\_LOG.info("{};{};{};{};{};{}", tierName, logReason, latencyMs,

partitionLatenciesMicros, request.getClientRequestID(),

request.getSearchQuery().getSerializedQuery());

}

}

}