package com.twitter.recos.hose.common

import com.twitter.finagle.stats.{Stat, StatsReceiver}

import com.twitter.recos.internal.thriftscala.RecosHoseMessage

import java.util.concurrent.Semaphore

trait EdgeCollector {

def addEdge(message: RecosHoseMessage): Unit

}

/\*\*

\* The class consumes incoming edges and inserts them into a buffer of a specified bufferSize.

\* Once the buffer is full of edges, it is written to a concurrently linked queue where the size is bounded by queuelimit.

\*/

case class BufferedEdgeCollector(

bufferSize: Int,

queue: java.util.Queue[Array[RecosHoseMessage]],

queuelimit: Semaphore,

statsReceiver: StatsReceiver)

extends EdgeCollector {

private var buffer = new Array[RecosHoseMessage](bufferSize)

private var index = 0

private val queueAddCounter = statsReceiver.counter("queueAdd")

override def addEdge(message: RecosHoseMessage): Unit = {

buffer(index) = message

index = index + 1

if (index >= bufferSize) {

val oldBuffer = buffer

buffer = new Array[RecosHoseMessage](bufferSize)

index = 0

Stat.time(statsReceiver.stat("waitEnqueue")) {

queuelimit.acquireUninterruptibly()

}

queue.add(oldBuffer)

queueAddCounter.incr()

}

}

}