package com.twitter.recos.user\_user\_graph

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.finatra.kafka.consumers.FinagleKafkaConsumerBuilder

import com.twitter.graphjet.bipartite.NodeMetadataLeftIndexedPowerLawMultiSegmentBipartiteGraph

import com.twitter.graphjet.bipartite.segment.NodeMetadataLeftIndexedBipartiteGraphSegment

import com.twitter.recos.hose.common.UnifiedGraphWriter

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

import com.twitter.recos.util.Action

case class UserUserGraphWriter(

shardId: String,

env: String,

hosename: String,

bufferSize: Int,

kafkaConsumerBuilder: FinagleKafkaConsumerBuilder[String, RecosHoseMessage],

clientId: String,

statsReceiver: StatsReceiver)

extends UnifiedGraphWriter[

NodeMetadataLeftIndexedBipartiteGraphSegment,

NodeMetadataLeftIndexedPowerLawMultiSegmentBipartiteGraph

] {

// The max throughput for each kafka consumer is around 25MB/s

// Use 3 processors for 75MB/s catch-up speed.

val consumerNum: Int = 3

// Leave 2 Segments for live writer

val catchupWriterNum: Int = RecosConfig.maxNumSegments - 2

import UserUserGraphWriter.\_

private def getEdgeType(action: Byte): Byte = {

if (action == Action.Follow.id) {

UserEdgeTypeMask.FOLLOW

} else if (action == Action.Mention.id) {

UserEdgeTypeMask.MENTION

} else if (action == Action.MediaTag.id) {

UserEdgeTypeMask.MEDIATAG

} else {

throw new IllegalArgumentException("getEdgeType: Illegal edge type argument " + action)

}

}

/\*\*

\* Adds a RecosHoseMessage to the graph. used by live writer to insert edges to the

\* current segment

\*/

override def addEdgeToGraph(

graph: NodeMetadataLeftIndexedPowerLawMultiSegmentBipartiteGraph,

recosHoseMessage: RecosHoseMessage

): Unit = {

graph.addEdge(

recosHoseMessage.leftId,

recosHoseMessage.rightId,

getEdgeType(recosHoseMessage.action),

recosHoseMessage.edgeMetadata.getOrElse(0L),

EMTPY\_NODE\_METADATA,

EMTPY\_NODE\_METADATA

)

}

/\*\*

\* Adds a RecosHoseMessage to the given segment in the graph. Used by catch up writers to

\* insert edges to non-current (old) segments

\*/

override def addEdgeToSegment(

segment: NodeMetadataLeftIndexedBipartiteGraphSegment,

recosHoseMessage: RecosHoseMessage

): Unit = {

segment.addEdge(

recosHoseMessage.leftId,

recosHoseMessage.rightId,

getEdgeType(recosHoseMessage.action),

recosHoseMessage.edgeMetadata.getOrElse(0L),

EMTPY\_NODE\_METADATA,

EMTPY\_NODE\_METADATA

)

}

}

private object UserUserGraphWriter {

final val EMTPY\_NODE\_METADATA = new Array[Array[Int]](1)

}