package com.twitter.interaction\_graph.scio.agg\_flock

import com.spotify.scio.values.SCollection

import com.twitter.algebird.Min

import com.twitter.flockdb.tools.datasets.flock.thriftscala.FlockEdge

import com.twitter.interaction\_graph.scio.common.InteractionGraphRawInput

import com.twitter.interaction\_graph.thriftscala.FeatureName

import java.time.Instant

import java.time.temporal.ChronoUnit

import org.joda.time.Interval

object InteractionGraphAggFlockUtil {

def getFlockFeatures(

edges: SCollection[FlockEdge],

featureName: FeatureName,

dateInterval: Interval

): SCollection[InteractionGraphRawInput] = {

edges

.withName(s"${featureName.toString} - Converting flock edge to interaction graph input")

.map { edge =>

// NOTE: getUpdatedAt gives time in the seconds resolution

// Because we use .extend() when reading the data source, the updatedAt time might be larger than the dateRange.

// We need to cap them, otherwise, DateUtil.diffDays gives incorrect results.

val start = (edge.updatedAt \* 1000L).min(dateInterval.getEnd.toInstant.getMillis)

val end = dateInterval.getStart.toInstant.getMillis

val age = ChronoUnit.DAYS.between(

Instant.ofEpochMilli(start),

Instant.ofEpochMilli(end)

) + 1

InteractionGraphRawInput(edge.sourceId, edge.destinationId, featureName, age.toInt, 1.0)

}

}

def getMutualFollowFeature(

flockFollowFeature: SCollection[InteractionGraphRawInput]

): SCollection[InteractionGraphRawInput] = {

flockFollowFeature

.withName("Convert FlockFollows to Mutual Follows")

.map { input =>

val sourceId = input.src

val destId = input.dst

if (sourceId < destId) {

Tuple2(sourceId, destId) -> Tuple2(Set(true), Min(input.age)) // true means follow

} else {

Tuple2(destId, sourceId) -> Tuple2(Set(false), Min(input.age)) // false means followed\_by

}

}

.sumByKey

.flatMap {

case ((id1, id2), (followSet, minAge)) if followSet.size == 2 =>

val age = minAge.get

Seq(

InteractionGraphRawInput(id1, id2, FeatureName.NumMutualFollows, age, 1.0),

InteractionGraphRawInput(id2, id1, FeatureName.NumMutualFollows, age, 1.0))

case \_ =>

Nil

}

}

}