package com.twitter.simclusters\_v2.summingbird.common

import com.twitter.simclusters\_v2.common.ClusterId

import com.twitter.simclusters\_v2.thriftscala.{

ClustersUserIsInterestedIn,

ClustersWithScores,

Scores

}

object SimClustersInterestedInUtil {

private final val EmptyClustersWithScores = ClustersWithScores()

case class InterestedInScores(

favScore: Double,

clusterNormalizedFavScore: Double,

clusterNormalizedFollowScore: Double,

clusterNormalizedLogFavScore: Double)

def topClustersWithScores(

userInterests: ClustersUserIsInterestedIn

): Seq[(ClusterId, InterestedInScores)] = {

userInterests.clusterIdToScores.toSeq.map {

case (clusterId, scores) =>

val favScore = scores.favScore.getOrElse(0.0)

val normalizedFavScore = scores.favScoreClusterNormalizedOnly.getOrElse(0.0)

val normalizedFollowScore = scores.followScoreClusterNormalizedOnly.getOrElse(0.0)

val normalizedLogFavScore = scores.logFavScoreClusterNormalizedOnly.getOrElse(0.0)

(

clusterId,

InterestedInScores(

favScore,

normalizedFavScore,

normalizedFollowScore,

normalizedLogFavScore))

}

}

def buildClusterWithScores(

clusterScores: Seq[(ClusterId, InterestedInScores)],

timeInMs: Double,

favScoreThresholdForUserInterest: Double

)(

implicit thriftDecayedValueMonoid: ThriftDecayedValueMonoid

): ClustersWithScores = {

val scoresMap = clusterScores.collect {

case (

clusterId,

InterestedInScores(

favScore,

\_,

\_,

clusterNormalizedLogFavScore))

// NOTE: the threshold is on favScore, and the computation is on normalizedFavScore

// This threshold reduces the number of unique keys in the cache by 80%,

// based on offline analysis

if favScore >= favScoreThresholdForUserInterest =>

val favClusterNormalized8HrHalfLifeScoreOpt =

Some(thriftDecayedValueMonoid.build(clusterNormalizedLogFavScore, timeInMs))

clusterId -> Scores(favClusterNormalized8HrHalfLifeScore = favClusterNormalized8HrHalfLifeScoreOpt)

}.toMap

if (scoresMap.nonEmpty) {

ClustersWithScores(Some(scoresMap))

} else {

EmptyClustersWithScores

}

}

}