package com.twitter.simclusters\_v2.hdfs\_sources

import com.twitter.dal.client.dataset.KeyValDALDataset

import com.twitter.scalding.{DateOps, DateRange, Days, TypedPipe}

import com.twitter.scalding\_internal.dalv2.DAL

import com.twitter.scalding\_internal.dalv2.remote\_access.{ExplicitLocation, ProcAtla}

import com.twitter.scalding\_internal.multiformat.format.keyval.KeyVal

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

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

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

import java.util.TimeZone

object InterestedInSources {

private val ModelVersionInterestedInDatasetMap: Map[ModelVersion, KeyValDALDataset[

KeyVal[UserId, ClustersUserIsInterestedIn]

]] = Map(

ModelVersion.Model20m145kDec11 -> SimclustersV2InterestedInScalaDataset,

ModelVersion.Model20m145kUpdated -> SimclustersV2InterestedIn20M145KUpdatedScalaDataset,

ModelVersion.Model20m145k2020 -> SimclustersV2InterestedIn20M145K2020ScalaDataset

)

/\*\*

\* Internal version, not PDP compliant, not to be used outside simclusters\_v2

\* Reads 20M145KDec11 production InterestedIn data from atla-proc, with a 14-day extended window

\*/

private[simclusters\_v2] def simClustersRawInterestedInDec11Source(

dateRange: DateRange,

timeZone: TimeZone

): TypedPipe[(UserId, ClustersUserIsInterestedIn)] = {

DAL

.readMostRecentSnapshot(

SimclustersV2RawInterestedIn20M145KDec11ScalaDataset,

dateRange.prepend(Days(14)(timeZone))

)

.withRemoteReadPolicy(ExplicitLocation(ProcAtla))

.toTypedPipe

.map {

case KeyVal(userId, clustersUserIsInterestedIn) =>

(userId, clustersUserIsInterestedIn)

}

}

/\*\*

\* Internal version, not PDP compliant, not to be used outside simclusters\_v2

\* Reads 20M145KUpdated InterestedIn data from atla-proc, with a 14-day extended window

\*/

private[simclusters\_v2] def simClustersRawInterestedInUpdatedSource(

dateRange: DateRange,

timeZone: TimeZone

): TypedPipe[(UserId, ClustersUserIsInterestedIn)] = {

DAL

.readMostRecentSnapshot(

SimclustersV2RawInterestedIn20M145KUpdatedScalaDataset,

dateRange.prepend(Days(14)(timeZone))

)

.withRemoteReadPolicy(ExplicitLocation(ProcAtla))

.toTypedPipe.map {

case KeyVal(userId, clustersUserIsInterestedIn) =>

(userId, clustersUserIsInterestedIn)

}

}

/\*\*

\* Internal version, not PDP compliant, not to be used outside simclusters\_v2

\* Reads 20M145K2020 InterestedIn data from atla-proc, with a 14-day extended window

\*/

private[simclusters\_v2] def simClustersRawInterestedIn2020Source(

dateRange: DateRange,

timeZone: TimeZone

): TypedPipe[(UserId, ClustersUserIsInterestedIn)] = {

DAL

.readMostRecentSnapshot(

SimclustersV2RawInterestedIn20M145K2020ScalaDataset,

dateRange.prepend(Days(14)(timeZone))

)

.withRemoteReadPolicy(ExplicitLocation(ProcAtla))

.toTypedPipe.map {

case KeyVal(userId, clustersUserIsInterestedIn) =>

(userId, clustersUserIsInterestedIn)

}

}

private[simclusters\_v2] def simClustersRawInterestedInLite2020Source(

dateRange: DateRange,

timeZone: TimeZone

): TypedPipe[(UserId, ClustersUserIsInterestedIn)] = {

DAL

.readMostRecentSnapshot(

SimclustersV2RawInterestedInLite20M145K2020ScalaDataset,

dateRange.extend(Days(14)(timeZone)))

.withRemoteReadPolicy(ExplicitLocation(ProcAtla))

.toTypedPipe.map {

case KeyVal(userId, clustersUserIsInterestedIn) =>

(userId, clustersUserIsInterestedIn)

}

}

/\*\*

\* Reads 20M145KDec11 production InterestedIn data from atla-proc, with a 14-day extended window

\*/

def simClustersInterestedInDec11Source(

dateRange: DateRange,

timeZone: TimeZone

): TypedPipe[(UserId, ClustersUserIsInterestedIn)] = {

DAL

.readMostRecentSnapshot(

SimclustersV2InterestedInScalaDataset,

dateRange.prepend(Days(14)(timeZone)))

.withRemoteReadPolicy(ExplicitLocation(ProcAtla))

.toTypedPipe.map {

case KeyVal(userId, clustersUserIsInterestedIn) =>

(userId, clustersUserIsInterestedIn)

}

}

/\*\*

\* Reads 20M145KUpdated InterestedIn data from atla-proc, with a 14-day extended window

\*/

def simClustersInterestedInUpdatedSource(

dateRange: DateRange,

timeZone: TimeZone

): TypedPipe[(UserId, ClustersUserIsInterestedIn)] = {

DAL

.readMostRecentSnapshot(

SimclustersV2InterestedIn20M145KUpdatedScalaDataset,

dateRange.prepend(Days(14)(timeZone))

)

.withRemoteReadPolicy(ExplicitLocation(ProcAtla))

.toTypedPipe.map {

case KeyVal(userId, clustersUserIsInterestedIn) =>

(userId, clustersUserIsInterestedIn)

}

}

/\*\*

\* Reads 20M145K2020 InterestedIn data from atla-proc, with a 14-day extended window

\*/

def simClustersInterestedIn2020Source(

dateRange: DateRange,

timeZone: TimeZone

): TypedPipe[(UserId, ClustersUserIsInterestedIn)] = {

DAL

.readMostRecentSnapshot(

SimclustersV2InterestedIn20M145K2020ScalaDataset,

dateRange.prepend(Days(14)(timeZone))

)

.withRemoteReadPolicy(ExplicitLocation(ProcAtla))

.toTypedPipe.map {

case KeyVal(userId, clustersUserIsInterestedIn) =>

(userId, clustersUserIsInterestedIn)

}

}

/\*\*

\* Reads InterestedIn data based on ModelVersion from atla-proc, with a 14-day extended window

\*/

def simClustersInterestedInSource(

modelVersion: ModelVersion,

dateRange: DateRange,

timeZone: TimeZone

): TypedPipe[(UserId, ClustersUserIsInterestedIn)] = {

DAL

.readMostRecentSnapshot(

ModelVersionInterestedInDatasetMap(modelVersion),

dateRange.prepend(Days(14)(timeZone))

)

.withRemoteReadPolicy(ExplicitLocation(ProcAtla))

.toTypedPipe.map {

case KeyVal(userId, clustersUserIsInterestedIn) =>

(userId, clustersUserIsInterestedIn)

}

}

}