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

import com.spotify.scio.ScioContext

import com.spotify.scio.values.SCollection

import com.twitter.beam.io.dal.DAL

import com.twitter.beam.io.dal.DAL.ReadOptions

import com.twitter.beam.job.ServiceIdentifierOptions

import com.twitter.dal.client.dataset.SnapshotDALDatasetBase

import com.twitter.dal.client.dataset.TimePartitionedDALDataset

import com.twitter.interaction\_graph.scio.agg\_address\_book.InteractionGraphAggAddressBookEdgeSnapshotScalaDataset

import com.twitter.interaction\_graph.scio.agg\_address\_book.InteractionGraphAggAddressBookVertexSnapshotScalaDataset

import com.twitter.interaction\_graph.scio.agg\_client\_event\_logs.InteractionGraphAggClientEventLogsEdgeDailyScalaDataset

import com.twitter.interaction\_graph.scio.agg\_client\_event\_logs.InteractionGraphAggClientEventLogsVertexDailyScalaDataset

import com.twitter.interaction\_graph.scio.agg\_direct\_interactions.InteractionGraphAggDirectInteractionsEdgeDailyScalaDataset

import com.twitter.interaction\_graph.scio.agg\_direct\_interactions.InteractionGraphAggDirectInteractionsVertexDailyScalaDataset

import com.twitter.interaction\_graph.scio.agg\_flock.InteractionGraphAggFlockEdgeSnapshotScalaDataset

import com.twitter.interaction\_graph.scio.agg\_flock.InteractionGraphAggFlockVertexSnapshotScalaDataset

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

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

import com.twitter.statebird.v2.thriftscala.Environment

import com.twitter.usersource.snapshot.flat.UsersourceFlatScalaDataset

import com.twitter.usersource.snapshot.flat.thriftscala.FlatUser

import com.twitter.util.Duration

import org.joda.time.Interval

case class InteractionGraphAggregationSource(

pipelineOptions: InteractionGraphAggregationOption

)(

implicit sc: ScioContext) {

val dalEnvironment: String = pipelineOptions

.as(classOf[ServiceIdentifierOptions])

.getEnvironment()

def readDALDataset[T: Manifest](

dataset: TimePartitionedDALDataset[T],

interval: Interval,

dalEnvironment: String,

projections: Option[Seq[String]] = None

)(

implicit sc: ScioContext,

): SCollection[T] = {

sc.customInput(

s"Reading ${dataset.role.name}.${dataset.logicalName}",

DAL.read[T](

dataset = dataset,

interval = interval,

environmentOverride = Environment.valueOf(dalEnvironment),

readOptions = ReadOptions(projections)

)

)

}

def readMostRecentSnapshotDALDataset[T: Manifest](

dataset: SnapshotDALDatasetBase[T],

dateInterval: Interval,

dalEnvironment: String,

projections: Option[Seq[String]] = None

)(

implicit sc: ScioContext,

): SCollection[T] = {

sc.customInput(

s"Reading most recent snapshot ${dataset.role.name}.${dataset.logicalName}",

DAL.readMostRecentSnapshot[T](

dataset,

dateInterval,

Environment.valueOf(dalEnvironment),

readOptions = ReadOptions(projections)

)

)

}

def readMostRecentSnapshotNoOlderThanDALDataset[T: Manifest](

dataset: SnapshotDALDatasetBase[T],

noOlderThan: Duration,

dalEnvironment: String,

projections: Option[Seq[String]] = None

)(

implicit sc: ScioContext,

): SCollection[T] = {

sc.customInput(

s"Reading most recent snapshot ${dataset.role.name}.${dataset.logicalName}",

DAL.readMostRecentSnapshotNoOlderThan[T](

dataset,

noOlderThan,

environmentOverride = Environment.valueOf(dalEnvironment),

readOptions = ReadOptions(projections)

)

)

}

def readAddressBookFeatures(): (SCollection[Edge], SCollection[Vertex]) = {

val edges = readMostRecentSnapshotNoOlderThanDALDataset[Edge](

dataset = InteractionGraphAggAddressBookEdgeSnapshotScalaDataset,

noOlderThan = Duration.fromDays(5),

dalEnvironment = dalEnvironment,

)

val vertex = readMostRecentSnapshotNoOlderThanDALDataset[Vertex](

dataset = InteractionGraphAggAddressBookVertexSnapshotScalaDataset,

noOlderThan = Duration.fromDays(5),

dalEnvironment = dalEnvironment,

)

(edges, vertex)

}

def readClientEventLogsFeatures(

dateInterval: Interval

): (SCollection[Edge], SCollection[Vertex]) = {

val edges = readDALDataset[Edge](

dataset = InteractionGraphAggClientEventLogsEdgeDailyScalaDataset,

dalEnvironment = dalEnvironment,

interval = dateInterval

)

val vertex = readDALDataset[Vertex](

dataset = InteractionGraphAggClientEventLogsVertexDailyScalaDataset,

dalEnvironment = dalEnvironment,

interval = dateInterval

)

(edges, vertex)

}

def readDirectInteractionsFeatures(

dateInterval: Interval

): (SCollection[Edge], SCollection[Vertex]) = {

val edges = readDALDataset[Edge](

dataset = InteractionGraphAggDirectInteractionsEdgeDailyScalaDataset,

dalEnvironment = dalEnvironment,

interval = dateInterval

)

val vertex = readDALDataset[Vertex](

dataset = InteractionGraphAggDirectInteractionsVertexDailyScalaDataset,

dalEnvironment = dalEnvironment,

interval = dateInterval

)

(edges, vertex)

}

def readFlockFeatures(): (SCollection[Edge], SCollection[Vertex]) = {

val edges = readMostRecentSnapshotNoOlderThanDALDataset[Edge](

dataset = InteractionGraphAggFlockEdgeSnapshotScalaDataset,

noOlderThan = Duration.fromDays(5),

dalEnvironment = dalEnvironment,

)

val vertex = readMostRecentSnapshotNoOlderThanDALDataset[Vertex](

dataset = InteractionGraphAggFlockVertexSnapshotScalaDataset,

noOlderThan = Duration.fromDays(5),

dalEnvironment = dalEnvironment,

)

(edges, vertex)

}

def readAggregatedFeatures(dateInterval: Interval): (SCollection[Edge], SCollection[Vertex]) = {

val edges = readMostRecentSnapshotDALDataset[Edge](

dataset = InteractionGraphHistoryAggregatedEdgeSnapshotScalaDataset,

dalEnvironment = dalEnvironment,

dateInterval = dateInterval

)

val vertex = readMostRecentSnapshotDALDataset[Vertex](

dataset = InteractionGraphHistoryAggregatedVertexSnapshotScalaDataset,

dalEnvironment = dalEnvironment,

dateInterval = dateInterval

)

(edges, vertex)

}

def readFlatUsers(): SCollection[FlatUser] =

readMostRecentSnapshotNoOlderThanDALDataset[FlatUser](

dataset = UsersourceFlatScalaDataset,

noOlderThan = Duration.fromDays(5),

dalEnvironment = dalEnvironment,

projections = Some(Seq("id", "valid\_user"))

)

}