package com.twitter.frigate.pushservice.store

import com.twitter.frigate.pushservice.params.PushQPSLimitConstants.SocialGraphServiceBatchSize

import com.twitter.hermit.predicate.socialgraph.RelationEdge

import com.twitter.storehaus.ReadableStore

import com.twitter.util.Future

case class SocialGraphServiceProcessStore(edgeStore: ReadableStore[RelationEdge, Boolean])

extends ReadableStore[RelationEdge, Boolean] {

override def multiGet[T <: RelationEdge](

relationEdges: Set[T]

): Map[T, Future[Option[Boolean]]] = {

val splitSet = relationEdges.grouped(SocialGraphServiceBatchSize).toSet

splitSet

.map { relationship =>

edgeStore.multiGet(relationship)

}.foldLeft(Map.empty[T, Future[Option[Boolean]]]) { (map1, map2) =>

map1 ++ map2

}

}

}