package com.twitter.simclusters\_v2.common

import com.twitter.decider.Decider

import com.twitter.servo.decider.{DeciderGateBuilder, DeciderKeyName}

import com.twitter.servo.util.Gate

class DeciderGateBuilderWithIdHashing(decider: Decider) extends DeciderGateBuilder(decider) {

def idGateWithHashing[T](key: DeciderKeyName): Gate[T] = {

val feature = keyToFeature(key)

// Only if the decider is neither fully on / off is the object hashed

// This does require an additional call to get the decider availability but that is comparatively cheaper

val convertToHash: T => Long = (obj: T) => {

val availability = feature.availability.getOrElse(0)

if (availability == 10000 || availability == 0) availability

else obj.hashCode

}

idGate(key).contramap[T](convertToHash)

}

}