package com.twitter.tweetypie

package service

package observer

import com.twitter.servo.exception.thriftscala.ClientError

import com.twitter.snowflake.id.SnowflakeId

import com.twitter.tweetypie.thriftscala.GetTweetCountsRequest

import com.twitter.tweetypie.thriftscala.GetTweetCountsResult

private[service] object GetTweetCountsObserver {

type Type = ObserveExchange[GetTweetCountsRequest, Seq[GetTweetCountsResult]]

def observeExchange(stats: StatsReceiver): Effect[Type] = {

val resultStateStats = ResultStateStats(stats)

Effect {

case (request, response) =>

response match {

case Return(\_) | Throw(ClientError(\_)) =>

resultStateStats.success(request.tweetIds.size)

case Throw(\_) =>

resultStateStats.failed(request.tweetIds.size)

}

}

}

def observeResults(stats: StatsReceiver): Effect[Seq[GetTweetCountsResult]] = {

val retweetCounter = stats.counter("retweets")

val replyCounter = stats.counter("replies")

val favoriteCounter = stats.counter("favorites")

Effect { counts =>

counts.foreach { c =>

if (c.retweetCount.isDefined) retweetCounter.incr()

if (c.replyCount.isDefined) replyCounter.incr()

if (c.favoriteCount.isDefined) favoriteCounter.incr()

}

}

}

def observeRequest(stats: StatsReceiver): Effect[GetTweetCountsRequest] = {

val requestSizesStat = stats.stat("request\_size")

val optionsScope = stats.scope("options")

val includeRetweetCounter = optionsScope.counter("retweet\_counts")

val includeReplyCounter = optionsScope.counter("reply\_counts")

val includeFavoriteCounter = optionsScope.counter("favorite\_counts")

val tweetAgeStat = stats.stat("tweet\_age\_seconds")

Effect { request =>

val size = request.tweetIds.size

requestSizesStat.add(size)

// Measure Tweet.get\_tweet\_counts tweet age of requested Tweets.

// Tweet counts are stored in cache, falling back to TFlock on cache misses.

// Track client TweetId age to understand how that affects clients response latencies.

for {

id <- request.tweetIds

timestamp <- SnowflakeId.timeFromIdOpt(id)

age = Time.now.since(timestamp)

} tweetAgeStat.add(age.inSeconds)

if (request.includeRetweetCount) includeRetweetCounter.incr(size)

if (request.includeReplyCount) includeReplyCounter.incr(size)

if (request.includeFavoriteCount) includeFavoriteCounter.incr(size)

}

}

}