package com.twitter.servo.util

import com.twitter.finagle.stats.{StatsReceiver, Stat}

import com.twitter.util.Future

object LogarithmicallyBucketedTimer {

val LatencyStatName = "latency\_ms"

}

/\*\*

\* helper to bucket timings by quantity. it produces base10 and baseE log buckets.

\*/

class LogarithmicallyBucketedTimer(

statsReceiver: StatsReceiver,

prefix: String = LogarithmicallyBucketedTimer.LatencyStatName) {

protected[this] def base10Key(count: Int) =

prefix + "\_log\_10\_" + math.floor(math.log10(count)).toInt

protected[this] def baseEKey(count: Int) =

prefix + "\_log\_E\_" + math.floor(math.log(count)).toInt

/\*\*

\* takes the base10 and baseE logs of the count, adds timings to the

\* appropriate buckets

\*/

def apply[T](count: Int = 0)(f: => Future[T]) = {

Stat.timeFuture(statsReceiver.stat(prefix)) {

// only bucketize for positive, non-zero counts

if (count > 0) {

Stat.timeFuture(statsReceiver.stat(base10Key(count))) {

Stat.timeFuture(statsReceiver.stat(baseEKey(count))) {

f

}

}

} else {

f

}

}

}

}