MAG(i)C Pseudo-random Number Generator

This pseudo-random number generator generates integers between 1 and 1,000,000. This feature is implemented on two platforms in two different languages, Java and Python on Google App Engine instances and Google Compute VMs.

The software include in this repository meets the following requirements:

- For every 1000 numbers generated, at least 750 of them are unique.
- Numbers are displayed only in Arabic format with no leading zeroes.
- Webpage will display one number at a time in plaintext. To retrieve a new number, the page must be refreshed.

Platform Specific Information

Google App Engine

When implemented on Google App Engine, the site auto-scales to accommodate for the load that the site is experiencing. This is desirable, however be aware that if the site starts experiencing lots of traffic, that it will start to cost more money. It might be a good idea to set a daily spending limit to prevent any problems from arising.

Google Compute VM

Setup for Google Compute VMs is much more involved than that of App Engine, however it is easier to predict costs and optimize the server as all aspects of the server are configurable.

Language Specific Information

Python

The Python version of this app was written using Flask, a micro web-framework that itself is written in Python. Flask was used in order to create a simple and lightweight web experience that is also easily scalable up to very large scale projects.

For some context on how small and lightweight the random number generator site is, here is all of the core functionality of the site:

```
app = Flask(__name__)

@app.route('/')
def hello():
    return str(random.randint(0,1000000))
```

Additionally, the Python implementation uses the Web Server Gateway Interface (WSGI) calling convention to handle requests between the flask framework and the web server we are using, Nginx.

Nginx is more than capable of hosting this simple site, but it was selected for its simplicity, flexibility, and excellent performance under high load. Do note that in the App Engine implementation Nginx is not explicitly implemented, but is the default web server for App Engine applications.

Java

Performance

The Python VM version on a Google Compute Engine VM n1-standard-2 instance is capable of handling a little over 3000 requests/sec, and in testing seemed limited only by CPU power. Here is a `wrk` benchmark on aforementioned instance.

```
MacBook-Pro:~ andrewm$ wrk -t2 -c400 -d60s --latency http://pyvm.andrewreaganm.me/
Running 1m test @ http://pyvm.andrewreaganm.me/
  2 threads and 400 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency    73.69ms   40.35ms   1.63s    93.79%
    Req/Sec     1.72k   301.03     2.30k    87.73%
  Latency Distribution
     50%   65.09ms
     75%   80.65ms
     90%   99.09ms
     99%  212.33ms
  202983 requests in 1.00m, 34.05MB read
  Socket errors: connect 151, read 0, write 0, timeout 0
Requests/sec:   3378.26
Transfer/sec:    580.25KB
```