Distributed Performance testing on Cloud using FunkLoad

Team 9: Phase II Deliverable

Team Members

Rohan Shah, Ankush Jain, Mayank Gupta, Rishabh Ladha

Brief description/overview

To use Funkload - a web-testing tool - to provide a generic and usable web interface for basic functional and performance testing on the web.

1. Functional Testing

This feature will check for working of all the links in the URL provided by the user. If some URLs are not functioning (i.e. a 404 error is generated) then the user will be alerted about it.

2. Distributed Load Testing

It allows web developers to know how their applications respond to heavy traffic and uncover glitches that might not be otherwise apparent.

3. Front-end

The front-end will provide a user friendly interface for a user to interact with this application.

Tools used

1. FunkLoad
2. Apache2
3. Python
4. HTML/CSS/JavaScript for the frontend

Work Done So far

Section 1 : Sample Test Run

- A sample WebApp having 2 pages is setup using apache2 server.
- Pages contain basic html controls such as anchor tags as of now.
- Using FunkLoad simple demo test case has executed on this web app.
- This demo tests includes running a functional test and a benchmarking test.
- At the end using FunkLoad utility detailed report of both the test cases han generated.

Section 2: Spider

A spider-like tool that grabs all the links in a given website has been made in Python using urllib2. If the website is large, the tool may take some time, in which case it can be configured to test the links on the homepage only.

Code Description

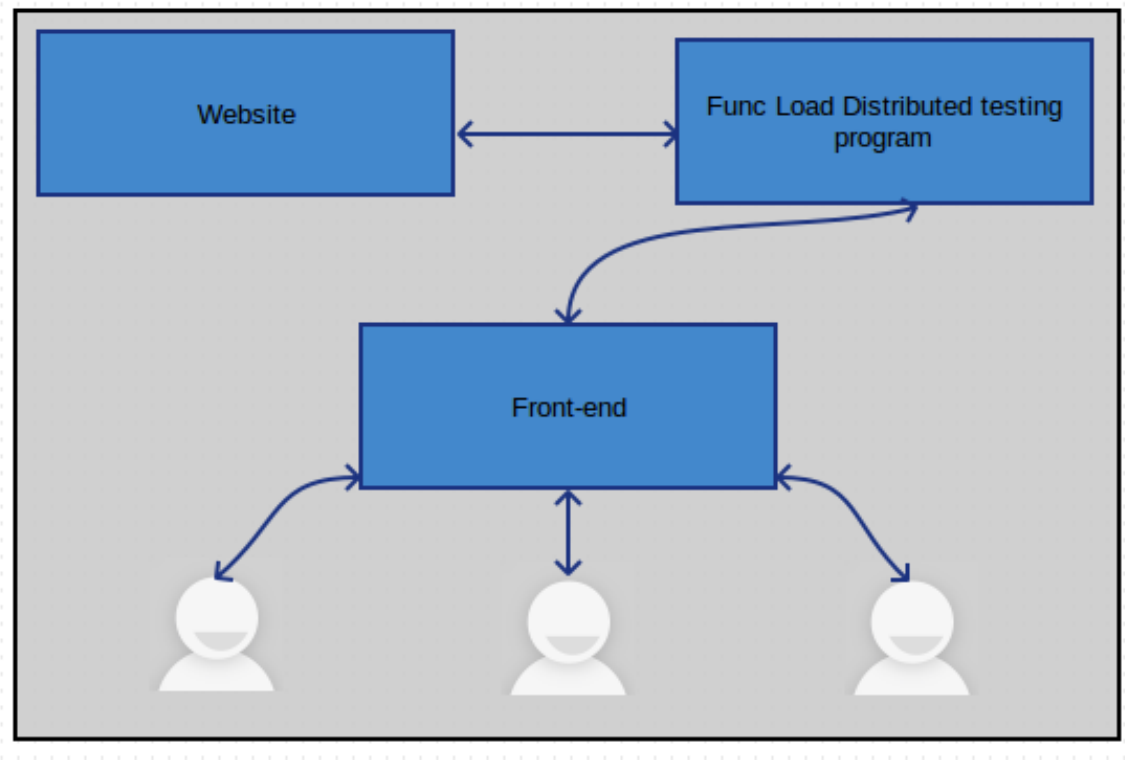
Section 1 : Sample Test Run

1. *class Simple(FunkLoadTestCase)*
   "Simple" is the name of the test case and it inherits inbuilt class "*FunkLoadTestCase*".
2. *def setUp(self)*
   This is the function which will take care of all the initial setup part before running any test case.
3. *def test_simple(self)*
   This is the actual test case where steps to perform test execution will be specified.

Section 2 : Website spider

This module is responsible for the functional testing part, in conjunction with Funkload. Given a website URL, the module will recursively check all the links in the website (like a spider) until no more unique links can be found. The FunkLoad application will check if accessing any link generated by the spider generates a 404 error, thus providing a form of functional testing for the website.

Architecture



Issues faced

1. Managing funkload daemons across distributed set of computing nodes may be tricky. This section has not yet been fully explored.

2. Obtaining all the links from a website. Doing this recursively can be very time consuming if the website is large. However, if proper coverage of the website is required, there is no other solution.

Link to Code

https://github.com/mayank93/funkloadApp