UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Benchmarking Service
Test Reports

ProCoders

Bongani Tshela - 14134790

Harris Leshaba - 15312144

Joseph Letsoalo - 15043844

Minal Pramlall - 13288157

Contents

1 Purpose.

This document outlines the various activities performed as part of testing of the algorithm bench-marking system application.

2 System Overview.

The Benchmarking service will be to provide users with the capability of testing viability between algorithms with graphical data backed by performed tests. Programmers are taught multiple ways to complete a task, and with a benchmarking tool at their disposal, they will be able to make informed decisions with which is the best algorithm to use among the ones that they have considered. The service will make use of microkernels, a super-lightweight operating system that will handle user requests and provide results. The system will be run off a cloud infrastructure service such as Google Cloud Compute Engine, to ensure the service is completely online, and thus completely accessible from anywhere with an internet connection. The user will communicate with the system through a website interface, they can submit multiple algorithms to benchmark and multiple datasets for them to be tested against. The system will benchmark all of these concurrently and provide the user with graphs and other visual output on the results. The system will be released into the open-source domain upon completion (Per request of the client).

3 Types of testing.

1. Unit Testing :
We are using usnit testing in Java [Junit] to test and verify each and every function implemented.
2. Intergration Testing :
We are using Sahi Pro to test both the functions in web scripting and the intergrated application or system.

4 Functional Requirements or Modules Testing.

4.1 User Management Module.

User registration and log-in :
The functions for user registration and verification were written in PHP and are being tested with sahi pro. The functions work properly with all test conditions we put it in. The module queries the database successfully and returns and displays relevant errors when results are not found.

4.2 Bench-marking and VM initializing Module.

Algorithm bench-marking :
Given the correct parameters [Test data and an actual algorithm], this module works fine under certain conditions [The module is still in development]. The module is written in java and being tested with Junit. The module will report code errors and infinite loops or non functioning codes.
VM-Initializing :
Module still under development. This module is written in Java and other scripting languages to initialize and queue virtual Osv machines to be used by the system.

4.3 Display module

The module is witten in PHP and other web scripting language. Given the results (Json strings) from the bench-marking module, the module graphically represents the results to make it easy for the use to see their results. The module was tested with Sahi Pro (Screenshots to follow) and it works fine under all test conditions it was put in.

4.4 Access Module

5 Non-Functional requirements Tested.

1. Scalability - This is a big part of this project. Scalability has been tested and we are in a process of making the project even more scalable.
2. Security -
3. Performance .

6 Conclusion.

The testing of the bench-marking is a work in progress and some modules will still be properly tested. Screen-shots and test data will be posted shortly.