



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

BENCHMARKING SERVICE
SOFTWARE REQUIREMENTS SPECIFICATION

ProCODERS

BONGANI TSHELA - 14134790

HARRIS LESHABA - 15312144

JOSEPH LETSOALO - 15043844

MINAL PRAMLALL - 13288157

Contents

Table Of Contents	1
1 Introduction	2
1.1 Purpose	2
1.2 Scope	2
1.3 Overview	2
2 Overall Description	2
2.1 Product Perspective	2
2.1.1 User Interfaces	2
2.1.2 Hardware Interfaces	3
2.1.3 Software Interfaces	3
2.1.4 Communications Interfaces	3
2.1.5 Memory	3
2.2 User Characteristics	3
2.3 Constrains	3
3 Specific Requirements	3
3.1 External Interface Requirements	3
3.2 Functional Requirements	4
3.3 Performance Requirements	4

1 Introduction

1.1 Purpose

This document aims to unambiguously define the requirements of the microbenchmarking web service. The document describes the uses of the product including objectives and goals. This document is intended for the development team, to enable all members of the team to have the same understanding of what should be done. The client can also use the document to verify whether the development team understood the needs and goals of the user, pertaining to the use of the service.

1.2 Scope

Product Name: ProBenchmarking Probenchmarking is an online microbenchmarking service that enable users to benchmark thier algorithms on different platforms. The users can measure how their algorithms use resouces of the system. They can measure how the algorithms uses a variety of system performance attributes such as CPU time, elapsed time, memory usage, power consumption and heat generation.

1.3 Overview

2 Overall Description

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).

2.1 Product Perspective

A benchmarking service is not easily accessible so this system will be able to provide for that need. Team ProCoder will undertake the development of this service, and provide a reliable and convenient system that will be available for public use.

2.1.1 User Interfaces

The Benchmarking system will have a website interface that the user will be interacting with, it will use modern design and an easy-to-use interface that will allow the user to interact without difficulty and also be pleasing to the eye.

The user will be given the option if they want to register to the site or not, and will then be directed to the input page where they can submit their algorithms and datasets that they want to be benchmarked. The system will perform the benchmarking and return the results to the user in the form of graphs and other visual representation, if the user had decided to register to the site then they will be given the option to save their benchmarking results should they want to revisit it at a later time.

2.1.2 Hardware Interfaces

The benchmarking system will be able to run on internet browsers, the latest browsers and ones compatible with HTML5 is recommended.

2.1.3 Software Interfaces

The benchmarking system will be run on a cloud platform, using MySQL database technologies to store user data.

2.1.4 Communications Interfaces

The benchmarking service will make use of HTTP for the user to communicate with the website interface of the system.

2.1.5 Memory

The microkernel technology has been chosen as it is a Virtual Machine that is attractively lightweight on memory, with an estimated 20MB on overheads, the size of the VM will look to be approximately 80MB in size. The resource allocation to the VM is determined based on the request so that it will only be allocated the amount of resources to suit the request.

2.2 User Characteristics

The web service will be used by people who are relatively knowledge about computer programming and the internal organization of computer systems, that is, they are familiar with the computer jargon that will be used. The service will be used by developers, researchers, teachers and students in the computer programming world. The benchmarking service should be used in a generic way, without intricate configuration, to welcome usage from a large potential user base.

2.3 Constrains

3 Specific Requirements

3.1 External Interface Requirements

The user will communicate with the web interface, providing test algorithms and datasets as inputs.

The Access module will receive the user inputs, validate correctness, and then send it to the benchmarking module.

Error checking will consist of ensuring the algorithms and datasets are compatible, as well as the code that is going to be executed is not harmful to the system (Eg. Infinite loops).

The Benchmarking module will boot up a virtual machine to respond to this request, and send the user input through as parameters.

The Access module will receive the results from the Benchmarking module, and then send it to the Display module.

The Display module receives the results from the Access module and then draws visual representation on the Web Interface

If the user is a free or registered is determined at this stage and will determine how extensively drawn the output will be for the user.

If the user is registered, then they will have the option to save their benchmarking data for later viewing, which will be performed by the Access module.

A MySQL database will be utilized for storing user data.

3.2 Functional Requirements

The system should enable the user to upload source code

The system should verify that the code is complete, no errors

The system should let the user choose the performance attribute to measure

The system should run the algorithm

The system should provide feed back to the user

The system should let user create,read, update and delete and account

3.3 Performance Requirements

Efficiency

Time Behaviour

The benchmarking service should provide a response in at most 10sec, provided there are no delays pertaining to the internet connection.

Resource Behaviour

The platform the system will be running on should use minimal resources (isolated machines) so that the side-effects that are not a concern of the specified benchmark is minimized.

Scalability

The system should be able to handle a growing amount of work or be extended to accomodate growth. The total output should be able to increase when resources are added.

Reliability

The system should be able to compile and benchmark any algorithm written in java. The system should be available to any device that has a browser.

Usability

The system should be easy to use without any need for local configuration.