A BenchMarking Tool

by

Rahul Gupta

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

May, 2012

A BenchMarking Tool

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
Master of Technology (M.Tech)

by
Rahul Gupta

Supervised By
Dr. Amey Karkare

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

May, 2012

CERTIFICATE

It is certified that the work contained in this thesis entitled
"*A BenchMarking Tool*",
by *Rahul Gupta(Roll No. 10111030)*, has been carried out under my
supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Amey Karkare)
Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur
Kanpur-208016

May, 2012

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor Dr. Amey karkare for his support and encouragement. He provided me freedom to experiment with new ideas and inculcated the techniques required to understand and progress my work. I am grateful for his patient guidance and advice in giving a proper direction to my efforts.

I would like to thank many friends with whom I had many enjoyable as well as thought-provoking discussions.

Last, but not the least, I would like to thank my parents who have been a constant source of support and encouragement. Their belief in me survived even as mine failed.

Rahul Gupta

Dedicated to
my parents

Abstract

Bench-marked various programming languages (C, C++, JAVA, Python, Cython, Haskell) and to complete a 3-dimensional matrix with the three dimensions as Fields(Graph Theory, Web Programming, Numerical Computability, NP-Hard etc.), Languages and Attributes(such as Compile time, Memory Usage, Security, Debugging etc.). Finally, we aim to develop a GUI tool to manifest the results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 2

Background

Chapter 3

The Problem

3.0.1 Benchmarking

We know that our main aim was to speed-up Python. We want to use it's dynamic features, rich libraries, simplicity and along with that want to increase it's speed. So, In order to know the possible reasons why Python is slower than C started with benchmarking the following algorithms.

3.0.2 Problems Used for Benchmarking

Josephous Problem

Longest Common Subsequence

Pollard's rho Algorithm

3.0.3 Computer Algebra System Software

These are the results

- Python is interpreted, while C is compiled
- Python has no primitives data types
- List can handle more than one type of data,, so additional field is required
- But development time in python is less than as compared to C

Chapter 4

Approach

Introduction

Chapter 5

Implementation Details

Introduction

Chapter 6

Experimentation Results

Basic						
	C	**C++**	**Python**	**Cython**	**JAVA**	**Haskell**
Line Of Code						
Ease Of Writing						
Debugging						
Compile Time						
Run Time						
Memory Usage						
Security						

Graph Theory						
	C	**C++**	**Python**	**Cython**	**JAVA**	**Haskell**
Line Of Code						
Ease Of Writing						
Debugging						
Compile Time						
Run Time						
Memory Usage						
Security						

Numerical Computing						
	C	**C++**	**Python**	**Cython**	**JAVA**	**Haskell**
Line Of Code						
Ease Of Writing						
Debugging						
Compile Time						
Run Time						
Memory Usage						
Security						

Web Programming						
	C	**C++**	**Python**	**Cython**	**JAVA**	**Haskell**
Line Of Code						
Ease Of Writing						
Debugging						
Compile Time						
Run Time						
Memory Usage						
Security						

NP-Hard						
	C	**C++**	**Python**	**Cython**	**JAVA**	**Haskell**
Line Of Code						
Ease Of Writing						
Debugging						
Compile Time						
Run Time						
Memory Usage						
Security						

Chapter 7

Conclusion

Introduction