 doi:10.20944/preprints201909.0322.v1

COMPARATIVE ANALYSIS OF COMPILER PERFORMANCES AND PROGRAM EFFICIENCY

Bukie, Paul Tawo[1], Udeze, Chinedu Leonard[2], Obono, Iwara Ofem[3] and Edim, Bassey Edim[4].

1: bukiepaultawo@unical.edu.ng, 2: udezechinedu@unical.edu.ng, 3: onedoublezero@yahoo.com, 4:edime@unical.edu.ng

Department of Computer Science, University of Calabar, Calabar, Cross River State, Nigeria.

ABSTRACT

With the existence of several programming languages such as C/C++, Java, C#, LISP, Prolog, Python, Simula, F#, Go, Haskell, Scala, Ruby, Dart, Swift, Groovy etc. and diverse paradigms like structured, object-oriented, list, aspect-oriented, service-oriented, web, mobile and logic programming, there is a need to perform an exhaustive comparative analysis of diverse compilers and environments before making a choice of implementation technology in software engineering. Optimization of compilers helps to reduce execution time by making use of high speed processor registers, thereby, eliminating redundant computation. This paper reports some series of performance analysis done with some popular programming languages including Java, C++, Python and PHP. Programs involving recursive and iterative functions like factorial of large numbers and binary search of large arrays were run on the various platforms with the execution time recorded in milliseconds and represented in a chart. This can aid in making a selection of the appropriate language to use for a given application domain.

KEYWORDS

Java Virtual Machine (JVM), High level programming languages, High Performance Computing (HPC), PHP Framework, Compiler.

INTRODUCTION

A popular high level programming language like Java uses the Java Virtual Machines (JVM) to execute bytecodes. This is one of the most widely accepted languages for teaching in schools and for enterprise application development. In comparing languages for performance, we consider some criteria like simplicity, write-ability using program constructs and API, reliability and error handling, support for appropriate data types, availability and cost of installation, market value, community support and documentation, OS/platform independence, availability of libraries and coverage for major topics [1]. Strongly typed languages like Java and C# provide more support for data types than some older languages like GW BASIC, Pascal, C and JavaScript. In dynamically typed languages like JavaScript, PHP and Python, variables are not declared to be of any specific type, but associate with the type of the value they hold during execution. JavaScript

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2019 doi:10.20944/preprints201909.0322.v1

has a number of extensions Angular-JS, Node-JS, Socket.IO, Express-JS, Fire Base, D3 etc. while Java extensions are available for GUI, web programming, hardware interfacing and Bluetooth technology with toolkits such as AJAX, Servlets and Java Server Pages. Mobile Application development can be done in C# using the XARMIN, in Java using Android and J2ME and with JavaScript using Win Js and Phone Gap API.

RELATED WORKS

According to [2], when a method is first called in Java, the object-oriented language uses Just-In-Time compilers integrated into the Java Virtual Machine to compile bytecodes into native binary codes. They further stated that due to encapsulation, object-oriented languages have a large number of method calls. They conducted a set of experiment at the University of South Alabama using four sorting algorithms to test for runtime efficiency including Bubble sort, Insertion sort, Recursive Quicksort and Heap sort using Java and C++. The first two algorithms had $O(n^2)$ while the last two had O(nlogn) complexity. This test implemented identical versions of Java and C++ algorithms. The C++ used "-O2" optimization switch for speed of execution while Java used the HotSpot – a JIT technology – with on-the-fly runtime optimizations. The results of the test displayed the $O(n^2)$ algorithms with performance ratio range of 2.5 to 3 while the O(nlogn) algorithms ranged from 1.4 to 1.6. In all the cases, the Interpreted Java was the slowest, while the HotSpot-enabled Java was faster due to optimization, but C++ programs were relatively the fastest.

[3] identified two of the major challenges facing the computer field which can be solved by compiler research, including: the cost of programming multi-core processors and the security and reliability of complex software systems. Based on their publication, in the late 1950s, when the field of compiling was still new, its focus was on the translation of high level language programs into machine codes, and the optimization of space and time requirements of programs. From their view, code optimization helps in avoiding redundant computations, allocating registers, working on instruction-level parallelism and enhancing locality. This delivers a great level of performance in compilers. One of the ways of achieving program optimization is by making parallel programming mainstream. One of the key ways of achieving this is by interactivity which might involve incorporation of compilers into IDEs, redesigning compiler algorithms and optimization.

Another way of achieving program optimization is by developing a rigorous approach to architecture-specific optimization. Part of their recommendations is that in order to achieve new programming languages that could be suitable for expressing computation, there should be a description of algorithms and data structures. Recent research at MIT, University of Tennessee, University of California Berkeley, Carnegie Mellon University etc. has demonstrated a new offline approach called "autotuning" which defines a search space for the implementations of a program for better optimization. This facilitates the development of libraries for numeric and symbolic algorithms. Correctness of programs and security add to the overall efficiency of the

 doi:10.20944/preprints201909.0322.v1

algorithms implemented. Debugging aims at developing engineering technique to detect and eliminate defects in order to bring programming process and software engineering models in conformity with engineering standards in aeronautical, automotive and electronic engineering.

[4] of University of Hail, Saudi Arabia described high level programming languages as languages that are close to natural language and provide an abstraction from the enclosed firmware and the hardware of the machine. These languages can be classified into functional, procedural and object-oriented. Compilation time is the amount of time taken for the compiler to convert the source code into an executable file while execution time is the amount of time taken to receive input and generate output during execution.

Research paper by [5] stated that it is necessary to do much optimization on intermediate code since the optimizer can be developed once and used subsequently without alterations for filtrations on the back end and front end. The code to be optimized is usually placed on the Left-Hand Side (LHS) of the compiler optimization. From their description, a compiler can work on both hard and soft problems. In a hard problem, a simple mistake can threaten the correctness of the compiler. On the other hand, in a soft problem, a mistake potentially affects the resource usage of the compiler or the program being compiled but does not affect the correctness of the compiler. Examples of soft problems include the decision of which type of optimization to use and the order in which to run them. Continuous methods such as machine learning and combinatorial optimization can be used to handle soft problems. Research shows that testing can hardly flush out compiler errors and bugs. Translation validation tools are more effective in detecting bugs. It is an important use case for formal semantics as a proof that a compiler's output refines its input. Based on the conclusion of this study, compilers can be enhanced by withdrawing hand-written parts and substituting them with parts obtained using formal semantics, automated theorem provers and data-driven tools [5].

[6] of Assam Engineering Institute India outlined the issues developers face in software engineering, including: stability of code, platform independence, code optimization, programming style and standard, exception handling, unused variables, array boundary checks and software consistency. Testing of developed software solutions can be done manually (this is costly) or with automated testing tools. Examples of tools for testing some C-based languages like C/C++ and Java are JUnit, Java Coding Standard Checher (JCSC), Cppcheck, Appperfect and Simplectest. There are several criteria for testing which includes: efficiency, portability, usability, reliability, functionality, maintainability, vendor support, pricing etc.

Compilers pass through several phases in generating an object code which includes lexical analysis (conversion of a sequence of characters into a sequence of tokens), syntactic analysis (a process of analyzing a string of symbols), parsing (top-down or bottom-up), optimal storage management and code generation [7]. [8] conducted a research on how two languages like Smalltalk/X and Java can be integrated into a JVM like STX:LIBJAVA. This is possible by compiling one language into another's intermediate code. It can also be achieved by running two

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2019 doi:10.20944/preprints201909.0322.v1

virtual machines alongside and using some communication channels. The two languages can be executed in the same virtual machine using similar object representation. These two languages in this case are object-oriented, high-level, and class-based. They have automatic memory management and use message passing for communication between objects. However, they differ in class access, field access, selector matching, method overloading, protocol matching, exception handling and synchronization. Such integrations have runtime-level integrations and language-level integration. Some projects like Apache Tomcat, Saxon and Groovy support Java runtime. JRuby is another project that integrates two languages Ruby and Java into a single programming environment. This research revealed that STX:LIBJAVA is relatively slower than Oracle JVM in running the same code, and is implemented in production and research on language interoperability.

Based on the publication by [9] of Riga Technical University Latvia, all the components of the Phalcon PHP Framework are developed with the C programming language, and there are various versions of it for Windows, Linux and Mac. This framework has low server resource expenses with relative speed and high performance. They further described another framework, the Symfony2 used for fast web application development and database management with DBMS like MySQL, PostgreSQL and SQLite. From experiments, PhalcomPHP is much faster than Symfony2 in handling requests. [10] of Morocco conducted a series of tests to perform evaluation analysis on some PHP Frameworks like Laravel, Symfony and Codeigniter based on some criteria like memory usage, response time, request per second and number of files required on each framework. From the test results, Laravel had the most request per second while symphony had least; Symfony had the most memory usage while Laravel had the least; Symfony had the longest response time while Laravel had the least; and Laravel also required the most number of files for the web pages in the framework while Symfony required the least. Hence, it can be deduced from the test that Laravel surpasses the other MVC frameworks; it can be implemented for rapid application development of large scale systems. CodeIgniter on the other hand, is good for simple and short projects.

[11] of Sardar Patel University Gujarat, India described a collaborative language ‘One’ which can implement knowledge-bases. It was described as an extensible language that has ambiguity and conflict management mechanisms with natural-language-like syntax. This language has been implemented with Java with some classes like LexerenUS for lexical analysis, Parser for parsing, Assembler, Execution for bytecode execution, VirtualMachine, Memory for the virtual machine, Heap, Type for data types, SymbolTable, KnowledgeBase, Loader, etc. This research reported that this language permits developers to write program in one high level language and generate code for several languages and frameworks. According to [12] of Osmania University Hyderabad India, compilers enable three basic optimization paths, namely: (i) Classical optimization and procedure in-lining, (ii) Super-block optimization and loop unrolling and (iii) Hyper-block optimization. They further stated that two grammars are equivalent if they define the same language and can generate similar sentences. From their view, natural language

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2019 doi:10.20944/preprints201909.0322.v1

processing techniques can generate language specific information using machine learning and generic parsing instead of using grammar and transformation rules.

[13] of University of Alabama Birmingham, USA classified Domain-specific languages as languages geared towards a specific nature of problem and not general purpose software engineering e.g. SQL. They described the Google Query Language (GQL) designed for advanced Google Search. The lexical and syntax analysis of this compiler is integrated into one stage, which can generate lexer, parser and AST for the classes.

COMPARATIVE ANALYSIS OF PROGRAMMING LANGUAGES

Table 1: Development year of various high level programming languages

S/No	High level language	Year of development
1	C	1971
2	C++	1983
3	Java	1995
4	Python	1991
5	PHP	1994
6	BASIC	1964
7	COBOL	1959
8	ADA	1979
9	Pascal	1970
10	Fortran	1954
11	VB.NET	2001
12	Lisp	1958
13	Smalltalk	1969
14	Perl	1987
15	C#	2000

Programming languages can be compared in terms of syntax, compilation time, execution time, machine dependency, lines of code and efficiency.

Java: Older versions of this language seems to have a longer execution time relative to a language like C++; although, the implementations improve steadily with the use of Just-In-Time (JIT) compilation and areas such as garbage collection routines. It is a language that has both interpreters and compilers with the former being relatively slower. Java and C++ have many syntactic and semantic similarities. [2] found out that Java execution time is approximately 2 times slower than the C++ execution time and over 3 times slower than the C/C++ groups put together. Results from an experiment conducted by [14] showed that Java executed a particular algorithm in 13 clock cycles while C++ method executed in 12 clock cycles, a slow-down of about 8.3%. He equally demonstrated that object creation in Java could be similar to creating

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2019 doi:10.20944/preprints201909.0322.v1

objects using the *new* operator in C++ or using the *malloc* statement in C. The memory operations are done with the heap in all three cases.

The syntax of Java is derived from C/C++, although the former is almost totally object-oriented. Most of its primitive types are functions, the values are mostly objects and the variables belong to classes. The programs are platform independent. The programs are first compiled to an intermediate language called bytecodes by the JVM, and then to machine language codes. It can be described as a portable language. This language makes use of the automatic garbage collector as a background thread to ensure that memory is available on demand in order to achieve high performance. It has a reliable security system. Programmers can use a suite of APIs to achieve high level security. The JVM, as a security manager also removes un-trusted codes from the Operating System. [15] of University of A Coruna, Spain identified the options in Java for HPC as: shared-memory programming, java sockets, Remote Model Invocation (RMI) and message-passing. It uses threads and some language extensions for promoting parallel programming paradigms. Sockets are low-level interfaces used in network programming for data streaming. There are several projects on Java message-passing such as: MPJava, JCluster, Parallel Java, P2P-MPI, MPJ/Ibis, JMPI etc. They gave a classification of some current research in Java for HPC including: libraries for data parallelism; advancement of Java message-passing primitives, development of low-level message-passing devices and automations in MPJ algorithms.

C++: It is notable for its efficiency as an object-oriented language; even though it supports some elements of functional programming. [16] claims that, “the speed gap between Java and C++ can be explained with the slower memory management techniques in Java Virtual Machines.” It started as an extended improvement on the C language. It makes use of header files and preprocessor directives. Its program execution usually starts from the *main()* function. It is a very fast programming language as most of its modules are in machine language, and can run efficiently on specific machines. It includes a collection of both high level and low level features making it easier for hardware implementation. It is system dependent; although specific libraries and system utilities can be used for the implementation. In terms of security, this language is vulnerable to pointer hacks. It is less near to natural language when compared with Java and C#, with C# being the closest to human language of the three. However, C++ takes the least number of lines of code, with C# having the maximum. C++ lacks memory management efficiency relative to some other languages like Java.

C#: This is a general-purpose object-oriented language that requires .NET framework installation on the machine before execution. The types are also broken down into classes and objects as in Java. Most of the syntactic structure is inherited from C/C++. The basic structure is made up of classes with attributes, methods, namespace declarations and other statements. As .NET framework is a Microsoft product, it is not fully platform independent. Originally, it is well-supported by the Windows OS; although, there are some third party framework like MONO that tries to run ASP.NET on Linux. C# has high memory consumption as it is relatively slower than other C/C++-based languages. The slow speed can also be attributed to its runtime

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2019 doi:10.20944/preprints201909.0322.v1

optimization techniques. For high performance reasons, it is advisable to select data type in C# before using it. Performance can also be enhanced by: using *fo*r loop instead of *foreach* loop, selecting structure instead of class, using *Stringbilde*r instead of *String* for concatenation purposes and using direct assignment of data members to classes.

PHP: PHP simply means Hypertext Preprocessor, a server-side scripting language used for web development. This language follows the C-style syntax but uses the $ symbol for variable naming. It promotes object-oriented programming and online database management. It supports polymorphism and inheritance like other OOP languages. It is a very fast scripting language that can be integrated with other web-based languages like HTML, SQL, JavaScript, XML etc. According to [17] of Concordia University Canada, the source codes of PHP are usually interpreted to machine object codes on the web server. It is dynamically typed and weak which implies that there is no need for variable declaration before usage. The variables are of no specific data type. Eclipse IDE can be used to edit PHP codes to integrate it with HTML tags. PHP codes run in a root directly in a compatible server like WAMP or XAMMP. There are various PHP programming framework such ZEND engine, CakePHP2, CodeIgniter, Symfony2, Yii and PhalconPHP. Such frameworks have built-in functions; they have security advantages, they save cost of web development and have support forums that can provide answers to developers. According to [9], PHP is used for the server-side scripting of 81.7% of the websites with known programming language.

Python: This is a dynamically typed language that is less verbose than C-based languages. It heavily borrowed keyword arguments, exception model and module system from the Modula-3 language. It has two broad versions, namely: 2.x versions and 3.x versions. It supports programming paradigms such as functional, procedural, reflective, imperative and object-oriented approaches. It supports inheritance and polymorphism but lacks good encapsulation mechanism as all the classes are public without private or protected members. Python compiles to an intermediary code which is in turn interpreted by the Python Runtime Environment (PRE) to machine code. The PRE uses the garbage collector to handle all allocation and de-allocation of memory. Due to its dynamic typing, there is no guarantee that the program will run for many data types just because it ran in one. Hence, some errors based on data types are only detected during runtime. There are several open-source packages of Python compilers and interpreters including CPython, Pypy, Jython, Iron python and Skulpt. Python can be run on several IDEs like IDLE (an IDE that comes pre-installed with the reference implementation), PyCharm, PyDev, Aptana Studio and Komodo Edit.

Code Snippets for a set of high level programming languages.

Given a set of n positive integers X_1, X_2, X_3, …, X_n to compute the series, such that:

$$\sum_i^n \; X_i! = X_1! + X_2! + X_3! + \ldots + X_n! \text{ , } i=1,2,3, \ldots, n$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2019 doi:10.20944/preprints201909.0322.v1

This problem involves a loop (summation) and a recursive function (factorial). Thus, it can be used to compare the runtime efficiency of some high level programming languages.

Using a test case of n=15 positive integer numbers. The following sequence of data values were used consistently on these languages:

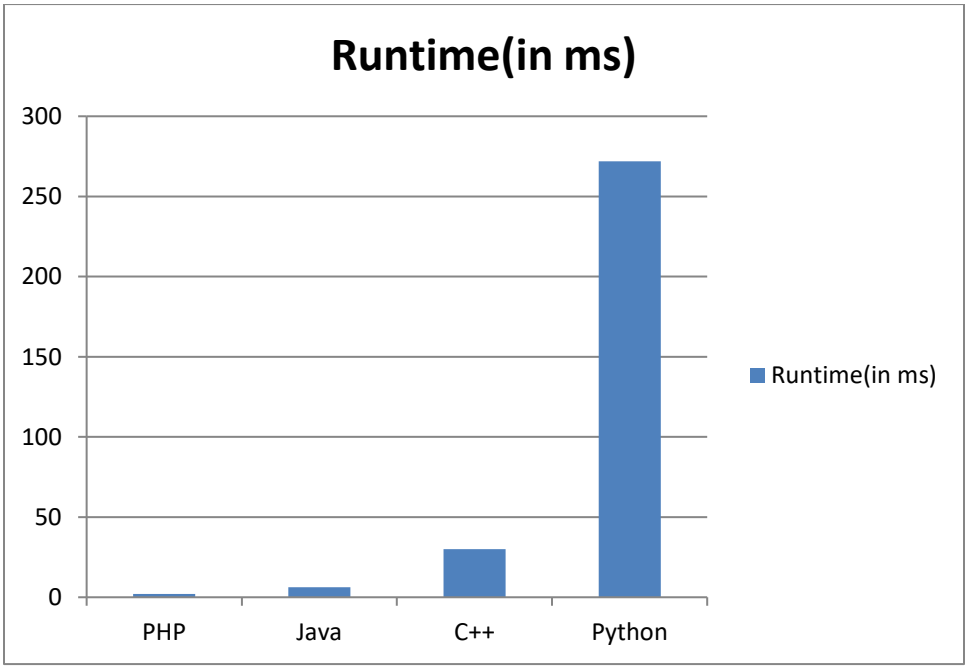
34, 45, 54, 32, 45, 65, 54, 44, 33, 21, 34, 54, 65, 54 and 31.

All the compilers gave an equivalent value of 1.64953×10^{91} as the sum of the factorials of the numbers with different execution time. The execution times were measured within the program from the first statement after the input to the last statement after the output.

Language	**Execution time (in milliseconds)**	**IDE used**
Java	6.2	Netbeans 8.0.2
PHP	2.04	WAMP Server/Komodo Edit
Python	272.00	IDLE 3.6.0
C++	30.00	Dev C++ 5.11

Table 2: Execution time of various programming languages.

Fig. 1: Graphical Representation of Runtime in milliseconds using bar chart



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2019 doi:10.20944/preprints201909.0322.v1

C++ program snippet

```
#include<iostream>
#include<cmath>
#include<ctime>
#include<cstdio>

using namespace std;
double N[100];
double fac(double n){     if(n<=0) return 1;  else return (n * fac(n-1)); }//
factorial function

int main(){
       int num;
       cout<<"How many numbers do you want to handle?\t";
       cin>>num; //number of positive integers
       for(int i=0; i<num; i++){
              cout<<"Enter the value of N"<<(i+1)<<"\t";
              cin>>N[i]; //array of positive n integers
       }
       clock_t start = clock();
       double sum = 0;
    for(int i=0; i<num; i++){
       sum += fac(N[i]); //summation of factorials
    }
    for(int i=0; i<num; i++){
       cout<<"N"<<(i+1)<<" = "<<N[i]<<"\t"<<endl; //display of integers
    }
    cout<<"The sum of the factorials = "<<sum<<endl; //value of sum of factorials
    clock_t end = clock();

    printf("The execution time = %.1f milliseconds",(float)(end-start));//execution
time in milliseconds
    return 0;
}
```

Java Program snippet

```
1 package compilers;
 2 import javax.swing.JOptionPane;
 3 public class Compilers {
 4    static double[] N;
 5    public static double fac(double n){ //factorial function
 6       if(n<=1) return 1;
 7       else return (n* fac(n-1));
 8    }
 9
10    public static void main(String[] args) { //main function
11       int num;
12       num = Integer.parseInt(JOptionPane.showInputDialog("How many integers do you want to handle"));
13       N = new double[num]; //number of integers
14       //input of integer values
15       for(int i=0; i<num; i++){
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2019 doi:10.20944/preprints201909.0322.v1

```
16          N[i] = Double.parseDouble(JOptionPane.showInputDialog("Enter the positive integer value of
N"+(i+1)));
17       }
18       double start = System.nanoTime();
19       double sum = 0;
20       for(int i=0; i<num; i++){
21         sum += fac(N[i]); //summation of factorials
22       }
23       for(int i=0; i<num; i++){
24         System.out.println("N"+(i+1)+" = "+N[i]+"\t"); //display of integers
25       }
26       System.out.println("The sum of the products = "+sum);
27       double end = System.nanoTime();//calculation of execution time
28       System.out.println(String.format("The execution time = %.1f milliseconds",(end-start)/1000000));
29    }
30
31 }
32
```

Python Program snippet

```
import array as arr
import time

def fac(n): #factorial function
    if n<=1:
        return 1
    else:
        return (n * fac(n-1))

def main(): #definition of main function
    N = arr.array('f')
    num = int(input("how many integers do you want to handle? "))
    #input of integer values
    for x in range(num):
        temp = float(input("Please enter a positive integer for N%i
"%(x+1)))
        N.append(temp)

    start = time.time()
    sum = 0
    for x in range(num): #computation of factorial summation
        sum += fac(N[x])
    #output of values
    for x in range(num):
        print ("N%i ="%(x+1), N[x], "\n ")
    print ("The sum of the factorials = ", sum)

    end = time.time()
    due = (end-start)*1000 #computation of runtime
    print ("duration of execution = %f milliseconds"%due)

main()
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2019 doi:10.20944/preprints201909.0322.v1

PHP code snippet

```
function fac($n){ //factorial function
   if($n<=1) return 1;
   else return ($n * fac($n-1));
}
if ($outer >1){ //reading of values from file

            $matrics[$track] = trim($data[1]);
            $track++;

          }
            $outer++ ;
  }
  $start = microtime(true) * 1000;
  $sum = 0; //computation of factorial sum
   for($i=0; $i<$num; $i++){
      $sum += fac($matrics[$i]);
   }
   echo '<center><table style="overflow: auto"><tr><td><h2>'.$num.' EXCEL FILE
VALUES</h2></td></tr>';
   for($i=0; $i<$num; $i++){  //display of values on the browser
      echo '<tr><td>'.$matrics[$i].'</td></tr>';
   }
   echo '<tr><td>The sum of the factorials = '.$sum.'</td></tr>'; //sum of factorials
   $end = microtime(true) * 1000;
   echo '<tr><td>The execution time = '.($end - $start).' milliseconds</td></tr>'; //execution time
   $message = "<center><h3>File uploaded successfully</h3></center>";
                              echo '<tr><td>'.$message.'</td></tr>';
   echo '</table></center>';
```

Comparison of Compilers using Binary Search Program

Given a program that can perform Binary search on an array of all positive odd integers from 1 to a specific limit X entered at runtime, to search for any integer n also entered during execution.

Taking a Worst case: X=25000000 and n = 4500001;

An average case: X= 5000000 and n = 2000005;

And a best case: X= 500000 and n = 300006

The outcomes for the four languages are shown below:

All the execution times were recorded in milliseconds.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2019 doi:10.20944/preprints201909.0322.v1

LANGUAGE	BEST	AVERAGE	WORST
PYTHON	124.8	873.6	4405.25
PHP	66.78	773.87	3278.36
JAVA	4.7	16	75
C++	16	94	422

Table 3: Execution time for the Binary Search algorithm using four languages

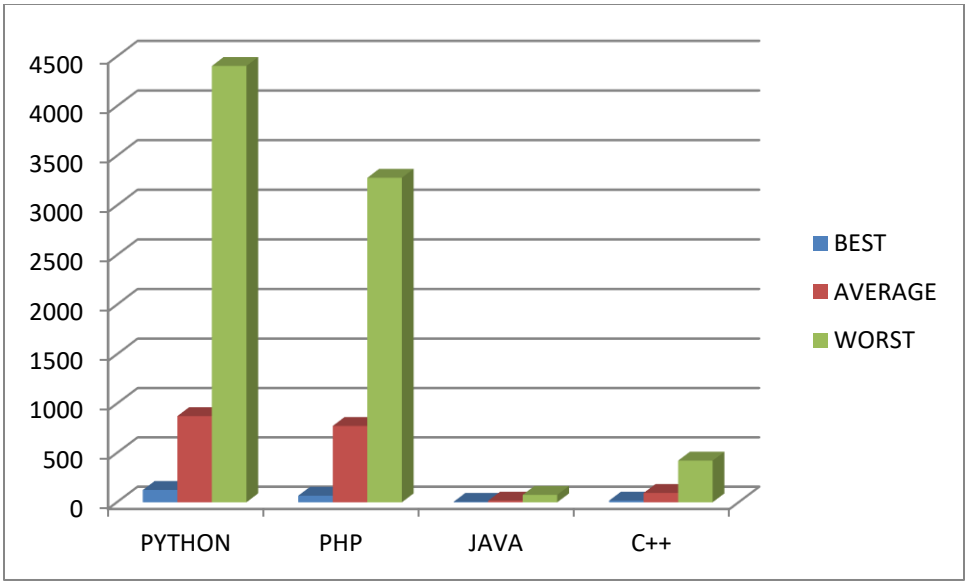


Fig. 2: Graphical Representation of the execution time for the four programming languages.

APPLICATIONS OF LANGUAGES

Java can be used for large-scale applications with efficient memory management and networking functions. It can be used for desktop applications, web-based systems, distributed computing environments and High Performance Computing [15]. PHP is used as a server-side language for developing web applications and easily works with MySQL for the management of databases. It fits perfectly for large-scale open source solutions, enterprise CMS and CRM systems. It can easily be embedded to HTML/CSS to display information on the browser. Python is also a scripting language used in building standalone applications. It supports Rapid Application Development and is suitable for prototyping. It can be used for website development but does not function properly with real-time/embedded applications. Different languages have interfaces for database communication; C# uses the ADO.NET, Java uses the JDBC while JavaScript can connect to some varieties like MySQL, SQL Server or MongoDB.

CONCLUSION

Comparative analysis shows that Java is most preferable for highly secured systems, C++ on the other hand is better for hardware implementation while C# can be implemented in high speed and easy-to-use applications. Operating Systems concepts such as memory management, I/O, process management, multithreading and semaphores are supported by C-based languages like Java and C#. They also support network programming concepts like Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) sockets. JavaScript can be used for both client-side (e.g. J-Query, Bootstrap, Angular-Js etc) and server-side (using Express-Js) web programming while C and Pascal uses the Common Gateway Interface (CGI) programming which is very cumbersome. It is deducible from this research that languages such as GW BASIC, Pascal and Fortran are gradually phasing out and cannot be used easily for implementing some cutting-edge technologies in cloud computing, OOP, machine learning, Big Data and Artificial Intelligence.

REFERENCES

[1] Ghazala, S. S. and Noman, I. A Qualitative Study of Major Programming Languages: Teaching Programming Languages to Computer Science Students. *Journal of Information and Communication Technology*, 2016, 10(1): 24-34. Retrieved from https://www.researchgate.net/publication/306438454

[2] Wentworth, S. and Langan, D. Performance Evaluation: Java vs C++. Computer Science Department, University of South Alabama 36688, 2000.

[3] Hall, M., Padua D. and Pingali K. Compiler Research: the Next 50 Years. *Communications of the ACM*, 2009, 52(2).

[4] Zahida, P. and Nazish, F. Performance Comparison of Most Common High Level Programming Languages. *International Journal of Computing Academic Research (IJCAR*), 2016, 5(5): 246 – 258.

[5] Lopes, P. N. and Regehr J. Future Directions for Optimizing Compilers, 2018.

[6] Hitesh, T., Plabita, B. and Khataniar G. P. Teaching Automated Test Data: Generation Tools for C, C++, and Java Programs. *International Journal of Computer Science and Information Technology (IJCSIT)*, 2013, 5(1).

[7] Mahak, J., Nidhi, S. and Neha M. Compiler Basic Design and Construction. *International Journal of Computer Science and Mobile Computing* (IJCSMC), 2014, 3(10): 850-852.

[8] Marcel, H., Jan, K., Jan, V. and Claus G. On the Integration of Smalltalk and Java. *Science of Computer Programming 96*, 2014, 17-33. Retrieved from www.elsevier.com/locate/scico

[9] Natalya, P. and Victoria, B. Analysis and Practical application of PHP Frameworks in Development of Web Information Systems. *Procedia Computer Science 104*, 2017, 51-56. Retrieved from www.sciencedirect.com

[10] Majida, L., Khaoula, B., Samira, K. and Mohamed, L. K. A Comparative Study of PHP Framework Performance. In Proc. t*he 12th International Conference Interdisciplinary in Engineering: Procedia Manufacturing 32*, 2019, pp. 864 – 871. Retrieved from www.elsevier.com/locate/procedia

[11] Jignesh, V. S. An Implementation of a Knowledge-based Programming language with Dynamic Grammar. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2016, 6(5), ISSN: 2277 128X. Retrieved from www.ijarcsse.com

[12] Raju, C., Thirupathi, M. and Arvind, T. Analysis of Parsing Techniques and Survey on Compiler Applications. *International Journal of Computer Science and Mobile Computing (IJCSMC)*, 2013, 2(10): 115-125, ISSN 2320-088X. Retrieved from www.ijcsmc.com

[13] Xiaoqing, W., Barrett, B., Jeff, G. and Marjan, M. Applying Object-orientation and Aspect-orientation in Teaching Domain-specific Language Implementation. *Journal of Circuits, Sytems and Computers (JCSC)*, 2005, 21(2).

[14] Mangione, C. Performance tests show Java as fast as C++. *Javaworld*, 1998, 3(2), retrieved from http://www.javaworld.com/javaworld/jw−02−1998/jw−02−jperf_p.html

[15] Guillermo, T. L., Sabela, R., Roberto, R. E., Juan T. and Ramon D. Java in High Performance Cmputing Arena: Research, practice and experience. *Science of Computer Programming 78*, 2013, pp. 425-444. Retrieved from www.elsevier.com/locate/scico

[16] Klemm, R. Practical Guidelines for boosting Java Server Performance. In *Proceedings of the ACM, 1999 Conference on Java Grande*, 1999, pp. 25-34.

[17] Zakari, A., Oualid, E. H., Kaushik S. and Chitrang, P. Comparative Studies of Six Programming Languages. Concordia University Montreal, Canada, 2015.