

An Empirical Analysis of C#, PHP, JAVA, JSP and ASP.Net regarding performance analysis based on CPU utilization

Md. Ahsan Arif*

Mohammad Shahazzat Hossain**

Nazmun Nahar***

Mst. Dilruba Khatun****

Abstract

The development of software is the most important part of a computer system. Presently, digital system gives a tremendous leadership in the corporate sector. To meet this demand, users face lack of speed and reliability in the modern digital system. Improvement of the performance of computer and software has been a topic of a great interest for the last era. The researches on both the hardware and software technology and compiler optimization have been applied in an effort to make the execution of software applications faster than before. Many developers have opted to adopt different programming languages replacing the old ones in accordance with their own choices. Those researches focus on the ways for assessing the effectiveness of the integrated programming language in the performance of computer system and utilization of resources. The researches also met two experimentation aims, which are extensive study of relevant writing and the implementation of applied enquiry. This was carried out through a case study with a software application that is developed by the integrated programming language. The major key findings of this analysis are: 1. Quality Algorithm 2. Selection of Suitable Programming Languages 3. Expert Writing Code and Tuned SQL query. Findings of these researchers show that the programming language has a huge effect on CPU or software performance in both the web and desktop. These observations can hold the interest of the developers and software architectural designers, who have been designing the system for executing the programs. After doing a comparative study amongst all targeted languages, we can say that JAVA/JSP has the best performance than any other regarding CPU uses, memory uses and execution time.

Keywords: *Software Engineering, Application of Software, Performance of Computer System, Performance of Programming Language, Algorithm, Integrated Programming Language, SQL query.*

* Assistant Professor, Department of Computer Science and Engineering, Asian University Bangladesh Dhaka-1230. E-mail: mdahsanarif@yahoo.com

** Programmer, ICIT, Bangladesh University of Engineering and Technology.

*** Administrator, US Software Ltd

**** Web Developer, Liveoutsourcing Ltd

I. INTRODUCTION

The software engineers and professionals tend to pay too little attention to the broader context of their work. They seem to be unaware about choosing suitable programming languages to build up the application software. They are quite flexible in the integrated programming language selection properly. This has some unfortunate consequences. We can get some necessary concepts down through the article about the issues associated with application software performance.

We can exemplify the Standish Group quoted from the CIO Journal, December 2003 Issue, entitled "By the numbers available, the software quality stinks." The Standish Group reported in 1999 that 74% of all projects were not successful. According to a survey by Standish Group in 2002, only 34 % of the software development was successful. Around 38 billion US dollars were lost in every annum due to software failure and one of the major reasons for that was high failure rate of poor software quality. Typical software projects are often late due to over budget, poor quality, and difficult tracking. Design of tools and implementation are the major reasons of poor software quality.

IBM Research has significantly increased in its investment in the language design and implementation, tooling, developing applications, as well as working. It found in closely observation of over 25 universities around the world for improving the language and its implementation, development of course materials, and applications. In the recent past three years, IBM Research awarded over \$750K to grow the community for the language. An X10 Birds-of-a-Feather session in October 2010 ACM SPLASH conference drew over 100 researchers (video related). The 2nd X10 workshop would be co-located with PLDI'12 that was held in Beijing, China in June, 2012. For many years, the increase of number of transistors on a chip as predicted by Moore's Law has resulted in higher clock speeds, and thus having a greater performance; without any change of software. This has enabled the creation of advanced performance-hungry applications and an increase of software developer's productivity through high-level programming languages that are enabled by these performances that increases day by day.

From above, it has become clear that these papers have been exposed in a particular term, which is related to the software engineering as well as improvement of software performances. Our requirements are not fulfilled in all of them clearly. We are introduced with all the modern day programming languages. Software developers have built up the software by such programming languages, in which they feel at home. But does it really exercise greater influence on the computer performance and computer resources? If so, how much would it impact on the computer performance and computer resources? These questions are raised by the practitioners in software development organization, members of software engineering and researchers themselves. Therefore, many debates are conventional among them. To find out the correct solution of these questions, we have commissioned this study. At first, we have designed a tool by different programming languages for both desktop and web based applications. We have already mentioned that these tools have been designed with the monthly salary report generation system of Asian University of Bangladesh (AUB) because of requirement of more space and more time. In connection of space and time, the speed has to be considered a necessary term here as well. The most critical schema among all the schemas is given below. This study has confirmed that the programming language is the most responsible for influencing the overall performance of a computer system.

II. BACKGROUND STUDY

Both the application performance and programming language performance are the burning questions all over the world. Many researchers already have their tremendous job on the both topics. They are tried to figure out performance in different angles such as programming language [7], SQL query quality [2,3], database, data access technology [43,44], software [8,16], hardware [15] etc. We know that many programming languages are common at present. Therefore, there are many opinions, arguments and criticisms among computer scientists, software developers, hardware manufacturers and general users as well as programmers about the affecting terms of computer system performance and utilization of computer resources. But it is very critical to make a strong decision.

In this study, we have proposed to measure a database application performance, which is written through different programming languages with OOP paradigm regarding execution time, CPU uses and memory uses.

III. RESEARCH METHODOLOGY

Our research methodology requires a gathered relevant data from the specified records and compiling it with database in order to analyze the research issues. We have already mentioned that our toolkit has been designed with partial payroll system, which can manipulate huge data. The aim of manipulating huge data is to give pressure to hardware system so that we can get some values of CPU usages, which may be remarkable. We have considered the measuring Computer Performance, Impacts, Technology, Algorithmic complexity and Data Management for calculating fruitful results. Before implementation of our toolkit, we considered a standard model of Hardware System, Windows Systems, Compiler, Server and the version of Programming Languages. We have selected in such a manner that all the things are not so backdated and having not so much high-ended. In fact, all the things are average usage ranked in the world.

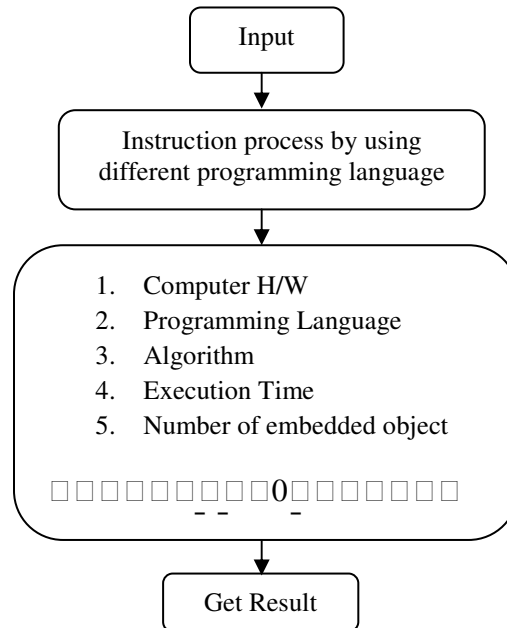
Processor	Intel(R) Core(TM)2 Duo 2.00 GHz
Memory	2GB DDR2 400 MHz
Hard Disk Drive	500GB 7200 RPM
Graphics	Graphics Intel(R) 965 Express 384.0 MB
Platform	Windows XP service pack 3

Figure 1: Hardware and Software Systems

Programming Language	Compiler	Server
JAVA (jdk 1.6)	JAVAC	Apache tomcat 7.0
C# v3.5	CSC	IIS 5.0
PHP v5.3.8	interpreter	Apache/2.2.21

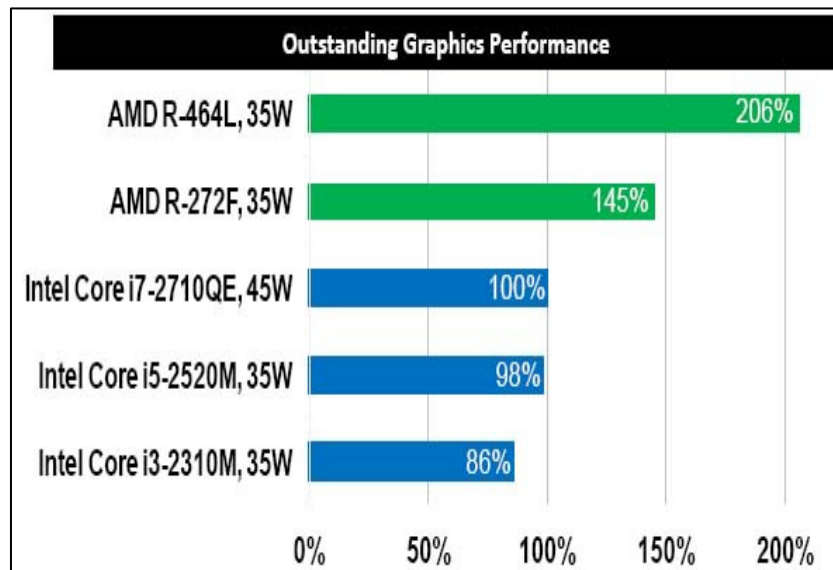
Figure 2: Programming Languages, Compiler, version and server

Here, we mentioned a general method of finding responsible terms of computer and software performance in fig 2.1. Under the section of Research Methodology (3.1 to 3.6), we have proved it scientifically.



A. COMPUTER PERFORMANCE

Cache, main memory, and compilers also play a role. Software applications also have different performance requirements [22]. A model of high-performance computers is derived from instruction timing formulas, with compensation for pipeline and cache memory effects. [21] As the computer performance depends on CPU [26], System RAM [27], HDD and Graphics, a performance graph of Graphic card is shown below, which has been proved in previous researches. [25]



B. IMPACTS

Each factor affects performance to a different degree and the overall effect is harder to predict accurately. Following are the three terms which are most responsible for working on CPU performance [38, 2, 3, and 22] that impacts factor(s) mainly depend(s) on Algorithm, SQL Query Quality and Response time.

a. Algorithm

In Mathematics and Computer Science, an algorithm is a step-by-step procedure for calculations [19, 20]. We considered the three most frequently encountered and measurable metrics of an algorithm. They are shown below:

- **Speed or running time:** The time of an Algorithm takes for an execution. [20]
- **Space:** The memory used by the algorithm during its operation. [20]
- **Memory:** It is often possible to make an algorithm faster by increasing the space of memory. This might be one of the solutions. Sometimes, it creates a situation of hardware changes. On the other hand, expensive calculation can be managed by programming recalculation. This is a very common method of improving speed. Some programming languages often add special features to support it, such as C++'s 'mutable' keyword. [20]

b. SQL Query Quality

Sometimes, the project contains many queries in different ways inside the source file. A lot of impacts are found depending on the programming language. [2] The queries are either in isolated form, or embedded within the host language. Finally, Query Quality depends on the following terms [2]:

- Number of parameters
- The context of the variables
- Equality compared to the other queries
- Number of tables used
- Inner queries
- Number of queries combined by the UNION keyword
- The number of depending queries
- The number of joints used

During this research, we used the below mentioned 3 (three) queries and the reasons is explained here in Method-I (3.2.2.1) and we found the alternative that is Method-II (3.2.2.2). It is more faster method (Method-II) than any other methods. The reason of avoiding Method-II is explained sequentially.

b. a. Method-I

```
SELECT DISTINCT e_id FROM Thesis ORDER BY e_id ASC;
```

The above SQL query collects all distinct employee-ids from the database into thesis table according to ascending order.

```
SELECT DISTINCT io_date FROM thesis WHERE e_id='.$e_id.' ORDER BY io_date ASC;
```

The 2nd SQL query gathers all distinct employees' attendance date according to employee-id (. e_id) from database into thesis table as ascending order.

```
SELECT DISTINCT io_time FROM thesis WHERE e_id= $e\_id$  AND
io_date=to_date('$io_date_twhp','DD-MON-YY') ORDER BY io_time ASC;
```

Finally, the 3rd query pick employee's attendance time according to employee-id and employee's attendance date from database into thesis table as ascending order. Therefore, one by one using of the 3 queries create an environment for preparing the required result. In this case, we must take help from the procedural language to move one by one record so that we can calculate total of all entities, which have been picked by the 3rd query. In this case, current algorithmic complexity is $O(n^3)$. The manipulation algorithm is stated in section 3.4.

b.b. Method-II

We can take id and date/time of all employees in a single query instead of the previous method. This time the attendance summary sheet of 10,000 employees for 20 working days, a huge quantity of data will be loaded in the primary memory. In that case, manipulations use looping technique that will increase complexity and memory uses. If there is a client server based architecture, the server will be loaded due to share of primary memory and a huge traffic of Network.

Secondly, the possible technique is in an array. If we run three proposed SQL query accordingly and insert all related values into different array than multiple complexities, which will arise in manipulation. Looping technique will also be used in server and not to share any memory for manipulating data inside the array. Therefore, client machine is responsible for all calculations. If we use this technique, the complexity will be increased in $O(n^6)$.

Another possible and effective technique is server side scripting. If we have done the manipulation through the database programming with trigger, the complexity will be less. The performance will also be better than before. Multi-tier application is popular today due to this technique. As we are measuring the application performance of programming language, we are skipping the backend database programming.

c. Response / Execution Time

Response or execution time is the most important factor on computer performance [24]. It depends on the Number of line, Number of embedded object and Number of page. We have considered these factors while developing our performance measurement toolkit.

C. CPU FORMULA

Among the scientific experiments, the methodologies for data analysis may involve complex calculations and it may cause for writing sophisticated computer programs. We have built up the toolkit using c# language for measuring computer system performance. We have used the following CPU formula [1].

$$\text{CPU}_{\text{time}} = (\text{Kernel time} + \text{User time}) * 100 / \text{Elapsed time}$$

This formula creates result which depends on the input of Kernel time i.e. User time and Elapsed time. The latest computer system architecture still depends on the above CPU uses formula.

D. ALGORITHMIC COMPLEXITY

Algorithm is also an essential portion of the CPU performance. It depends on programming. We proposed an algorithm for generating employee's attendance report among different programming languages, which already have been included in our toolkit.

In our proposed algorithm, we have used 3 while loops (1st loop) used for collecting employee-id and it repeats in 3 to 14 lines' calculation, 2nd inner while loop used for collecting attendance-date and it repeats in 5 to 13 lines and last inner (3rd loop) while the loop used for collecting attendance-time and it repeats 7 lines. Remaining other lines used for initialization, calculate total working-hour, total present-hour, absent-hour and single-entry etc. We have mentioned here a report on function, which proves our minimized and optimized code writing.

Function: REPORT ()

1. [Initialize] Set DAI=0, ct0=0,ct_end=0,ct9=090000,ct10=100000,ct16=160000, ct17=170000, twh=Total working hour, tr=Total ID, TWHP[], TWHA[], in_miss[], out_miss[], SE[].
2. while ID != NULL repeat steps 3 to 14
3. [Initialize] Set ta[], count_tdp=0, arr_twha[twh,00,00].
4. while Date != NULL repeat steps 5 to 13
5. [Initialize] Set dd=date, time_save [], a=0.
6. while Time != NULL repeat steps 7
7. Set time_save[a]=Time
8. a=a+1 [Increment counter]
9. [End of step 6 loop]
10. If ct0<ct9 then Set time_save[0]="09:00:00"
11. [End of if structure]
12. If ct_end>ct17 then Set time_save[end]="17:00:00"
13. [End of if structure]
14. If a = 1 then Set SE[DAI] +=1
15. [End of if structure]
16. Set Counter_tdp = Counter_tdp + 1
17. td[] = time_diff (time_save[end], time_save[0])
18. ta[] = time_add (ta,td)
19. If ct0>ct10 then Set in_miss[DAI] +=1
20. [End of if structure]
21. If ct_end<ct16 then Set out_miss[DAI] +=1
22. [End of if structure]
23. [End of step 4 loop]
24. Set TWHP[DAI]=ta, td1= time_diff (arr_twha, ta), TWHA[DAI]=td1, TDA=twd – count_tdp, DAI= DAI + 1 [Increment counter]
25. Exit.

a. Complexity:

$$\begin{aligned}
 &1+n+1+n^2+1+n^3+1+1+1+1 \\
 &+1+1+1+1 \\
 &=11+n+n^2+n^3 \\
 &=O(n^3)
 \end{aligned}$$

E. DATA MANAGEMENT

We have used oracle10g database and our database maintains the status of RDBMS, Client-Server database architect system and it is up to 3rd level normalized design.

a. Data Access Technology

All programming has different kinds of data access technology. We selected the best one for each language. C# has three data access technology such as ADO.Net (ActiveX Data Object), ADO, OLE DB (Object Linking and Embedding Database). OLE DB is the best for us because it provides high performance, raw manipulative power and system level programming. OJDBC (Oracle Java Database Connectivity) Driver for java is better than java Open Database Connectivity (ODBC) Bridge [10]. Our last programming language is PHP, we have used OCI8 (Oracle Call Interface) data access technology for connecting oracle10g [43, 44].

F. DEVELOPMENT

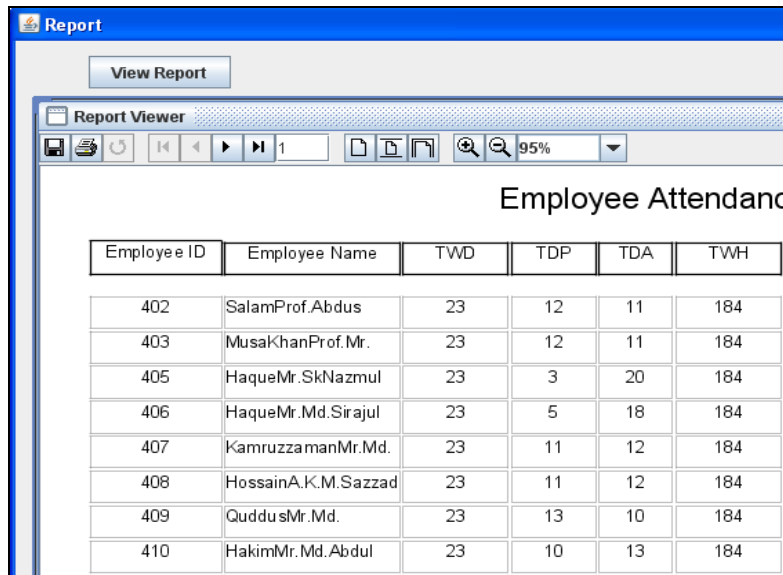
To get result from our research, we need to develop same application with the same algorithm for each programming language such as Java, C# (c sharp), Asp.Net, PHP and JSP. There are two types of applications, first- desktop and second- the web. These two types of applications are commonly used for desktop computer or laptop but web applications are growing rapidly as we considered both the types of applications for our research work.

We have developed a sample payroll reporting system for proving with practical data not merely analytical. All the programming languages, technologies, applications and other particulars are used and stated below:

a. Desktop Application

Java Technology: NetBeans IDE v7.0, JDK 1.6, IReport v4.5.0

Application (Screenshot):

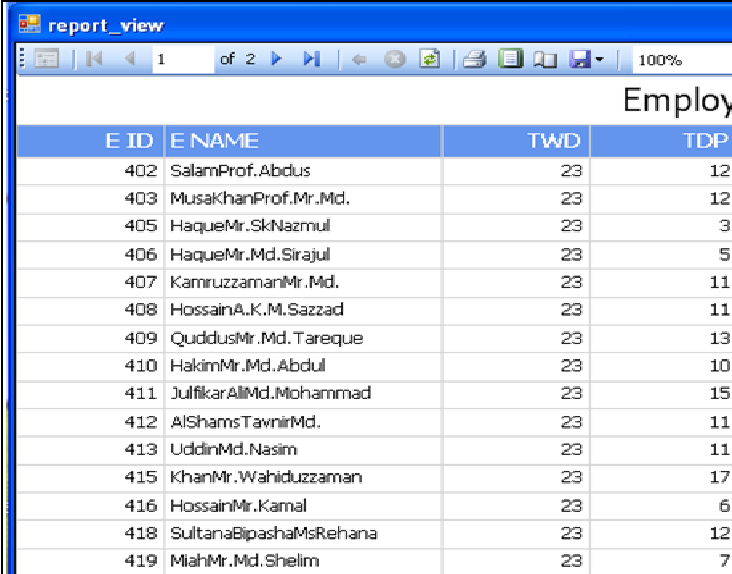


Employee ID	Employee Name	TWD	TDP	TDA	TWH
402	SalamProf.Abdus	23	12	11	184
403	MusaKhanProf.Mr.	23	12	11	184
405	HaqueMr.SkNazmul	23	3	20	184
406	HaqueMr.Md.Sirajul	23	5	18	184
407	KamruzzamanMr.Md.	23	11	12	184
408	HossainA.K.M.Sazzad	23	11	12	184
409	QuddusMr.Md.	23	13	10	184
410	HakimMr.Md.Abdul	23	10	13	184

Figure 5: JAVA Application (Screenshot)

C# (C sharp) Technology: Visual Studio 2008, Microsoft Report Viewer, .Net framework 3.5

Application (Screenshot):



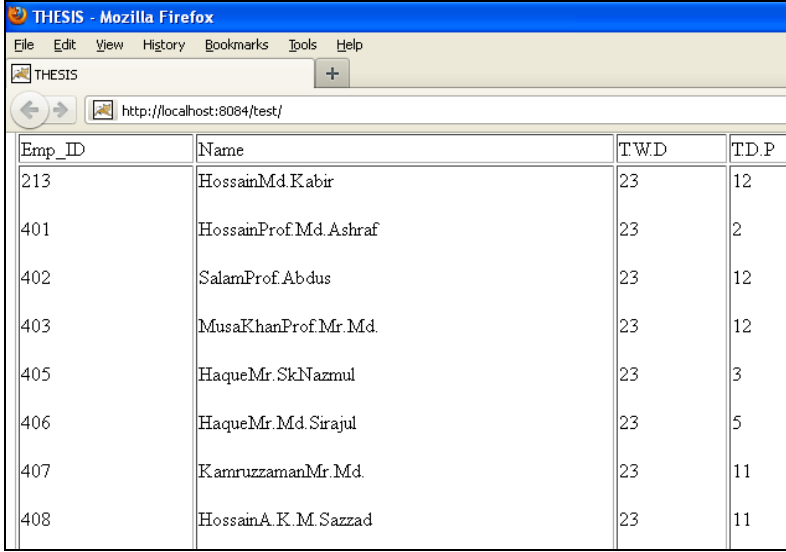
E ID	E NAME	T.W.D	T.D.P
402	SalamProf.Abdus	23	12
403	MusakhanProf.Mr.Md.	23	12
405	HaqueMr.SkNazmul	23	3
406	HaqueMr.Md.Sirajul	23	5
407	KamruzzamanMr.Md.	23	11
408	HossainA.K.M.Sazzad	23	11
409	QuddusMr.Md.Tareque	23	13
410	HakimMr.Md.Abdul	23	10
411	JulfikarAlIMd.Mohammad	23	15
412	AlShamsTavnirMd.	23	11
413	UddinMd.Nasim	23	11
415	KhanMr.Wahiduzzaman	23	17
416	HossainMr.Kamal	23	6
418	SultanaBipashaMsRehana	23	12
419	MiahMr.Md.Shelim	23	7

Figure 6: C# application (Screenshot)

b. Web Application

JSP Technology: NetBeans IDE v7.0, Apache Tomcat v7.0 Server, HTML, Mozilla Firefox 5.0, JDK 1.6

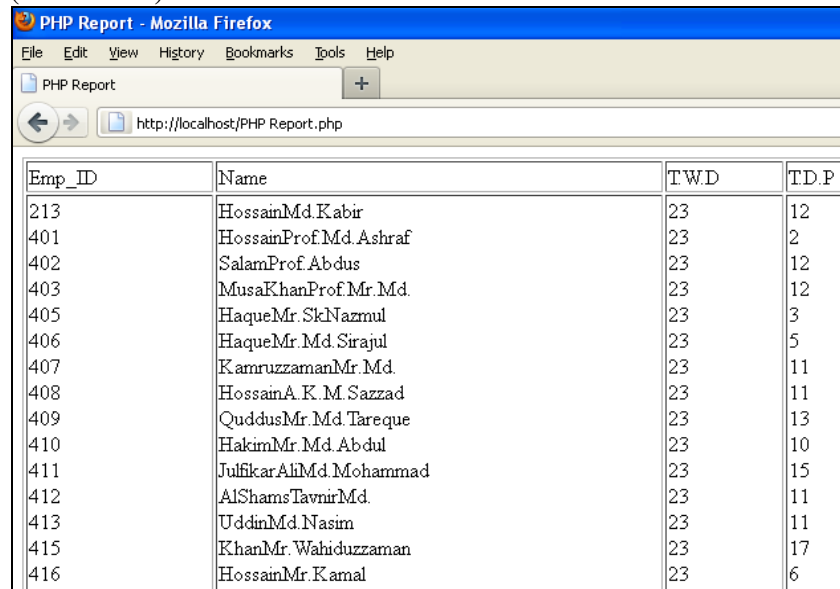
Application (Screenshot):



Emp_ID	Name	T.W.D	T.D.P
213	HossainMd.Kabir	23	12
401	HossainProf.Md.Ashraf	23	2
402	SalamProf.Abdus	23	12
403	MusaKhanProf.Mr.Md.	23	12
405	HaqueMr.SkNazmul	23	3
406	HaqueMr.Md.Sirajul	23	5
407	KamruzzamanMr.Md.	23	11
408	HossainA.K.M.Sazzad	23	11

Figure 7: JSP Application (Screenshot)

PHP Technology: Notepad++, Apache Server, HTML, Mozilla Firefox 5.0
Application (Screenshot):

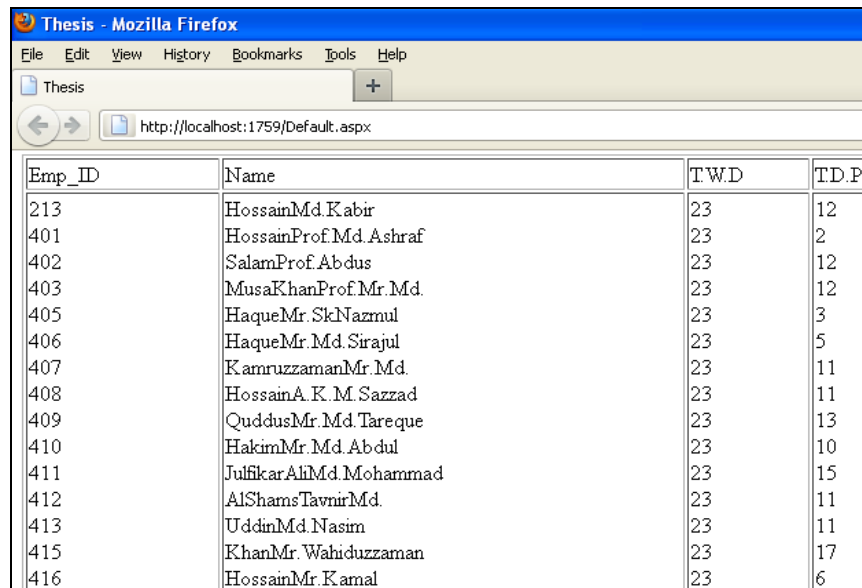


Emp_ID	Name	T.W.D	T.D.P
213	HossainMd.Kabir	23	12
401	HossainProf.Md.Ashraf	23	2
402	SalamProf.Abdus	23	12
403	MusaKhanProf.Mr.Md.	23	12
405	HaqueMr.SkNazmul	23	3
406	HaqueMr.Md.Sirajul	23	5
407	KamruzzamanMr.Md.	23	11
408	HossainA.K.M.Sazzad	23	11
409	QuddusMr.Md.Tareque	23	13
410	HakimMr.Md.Abdul	23	10
411	JulfikarAliMd.Mohammad	23	15
412	AlShamsTavnirMd.	23	11
413	UddinMd.Nasim	23	11
415	KhanMr.Wahiduzzaman	23	17
416	HossainMr.Kamal	23	6

Figure 8: PHP Application (Screenshot)

ASP.net Technology: Visual Studio 2008 IDE, C#, .Net Framework 3.5, HTML, IIS Server, Mozilla Firefox 5.0.

Application (Screenshot):

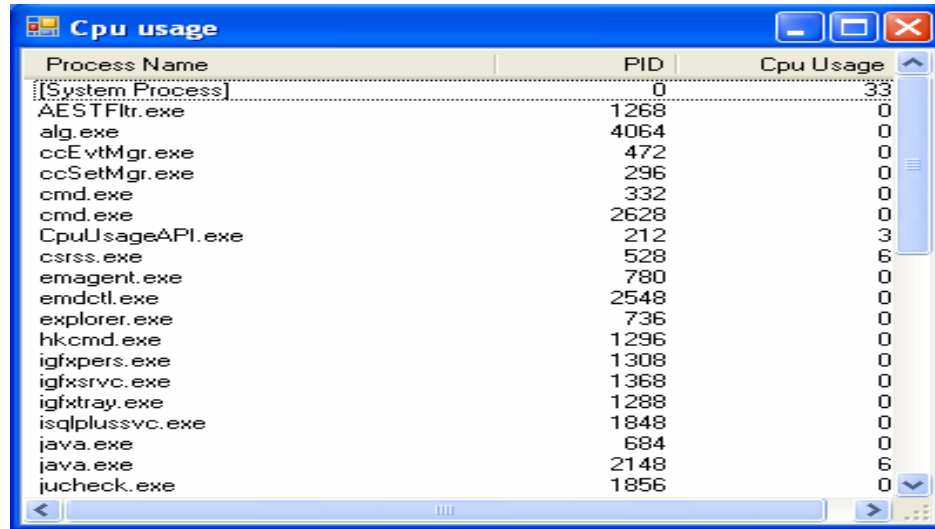


Emp_ID	Name	T.W.D	T.D.P
213	HossainMd.Kabir	23	12
401	HossainProf.Md.Ashraf	23	2
402	SalamProf.Abdus	23	12
403	MusaKhanProf.Mr.Md.	23	12
405	HaqueMr.SkNazmul	23	3
406	HaqueMr.Md.Sirajul	23	5
407	KamruzzamanMr.Md.	23	11
408	HossainA.K.M.Sazzad	23	11
409	QuddusMr.Md.Tareque	23	13
410	HakimMr.Md.Abdul	23	10
411	JulfikarAliMd.Mohammad	23	15
412	AlShamsTavnirMd.	23	11
413	UddinMd.Nasim	23	11
415	KhanMr.Wahiduzzaman	23	17
416	HossainMr.Kamal	23	6

Figure 9: ASP.net Application (Screenshot)

c. Performance measurement tool

Performance is the most important issue for all applications for calculating our application performance. We also developed another tool. This can calculate CPU usages time and memory usages.



Process Name	PID	Cpu Usage
[System Process]	0	33
AESTFltr.exe	1268	0
alg.exe	4064	0
ccEvtMgr.exe	472	0
ccSetMgr.exe	296	0
cmd.exe	332	0
cmd.exe	2628	0
CpuUsageAPI.exe	212	3
csrss.exe	528	6
emagent.exe	780	0
emdctl.exe	2548	0
explorer.exe	736	0
hkcmd.exe	1296	0
igfxpers.exe	1308	0
igfxsvc.exe	1368	0
igfxtray.exe	1288	0
isqlplussvc.exe	1848	0
java.exe	684	0
java.exe	2148	6
jucheck.exe	1856	0

G. PROPOSED METHOD

In modern computer science, performance is the biggest issue for the both applications and the computer system. Performance means, how much time and memory is actually used when a program is run on it, this depends on the machine, compiler, etc. as well as the source code [32]. Total existence of programming languages are almost 596 (Five hundred and ninety six) [29]. Some of these are currently used and some are historical. Currently, the most popular programming languages are Java, PHP and C#. They have different types of compiler architectures and programming language paradigms so that there are some issues of application or CPU performance. We have been investigating all the issues, which are explained in the above section. It influences on CPU performance depending on programming language. We have classified our application in two parts; firstly, desktop and another, secondly, is the web. The web applications have been growing rapidly [30] than desktop application. We have developed a sample payroll reporting system using object oriented programming (OOP) paradigm for both desktop (Java, C#) and web version (JSP, ASP.Net, PHP). The same algorithm is considered for each programming language for solving this problem. We also have developed a toolkit to calculate CPU usages time on windows platform using C#. We considered the resources allocated by our toolkit while getting the result. Afterwards, it investigates CPU usages time, process execution time and memory usages according to our proposed algorithm. Our toolkit is followed by all kind of previous researches regarding CPU performance. We also published our toolkit in international standard journal, which has been subscribed under all the famous Search Engine. Our result is included in the below result (6) section:

H. RESULTS

a. Memory

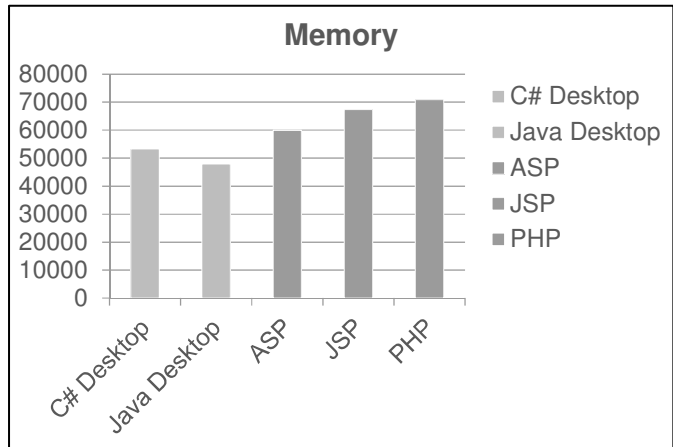
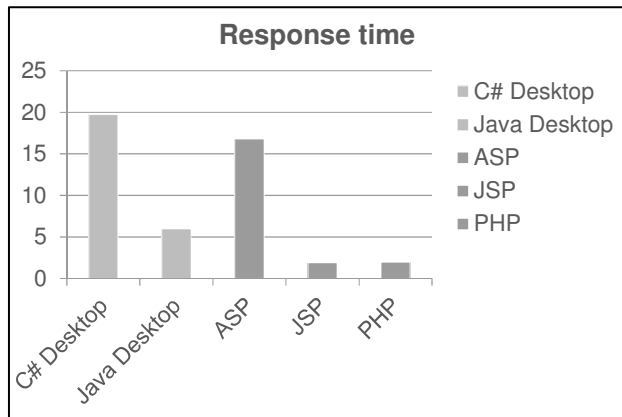


Figure 11: Memory Uses

b. Execution/Response Time

Figure 13: Execution Time



c. CPU Uses

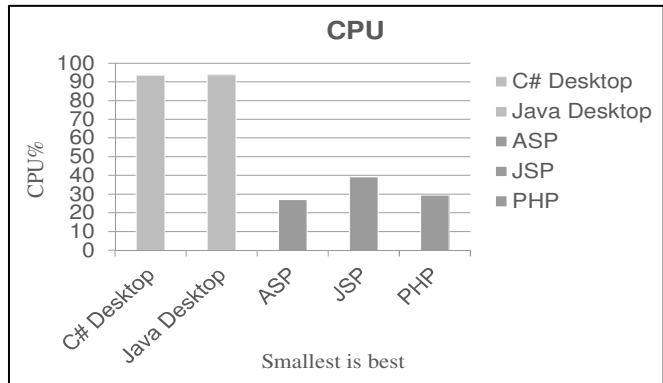


Figure 14: CPU Uses

d. Overall Performance

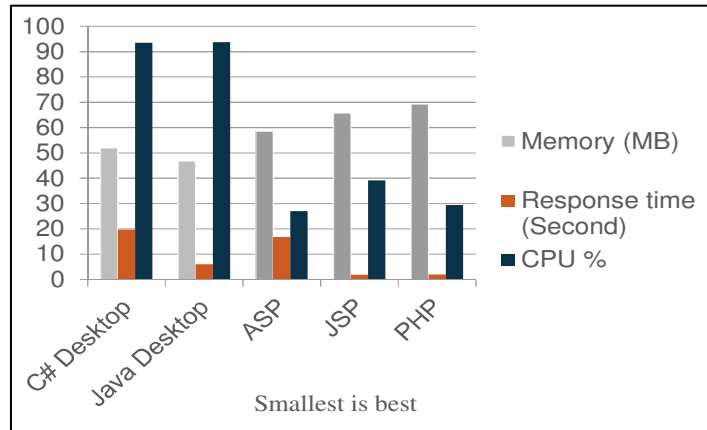


Figure 15: Overall performance

The above four result graph is the representation of real numeric value of output of our developed toolkit. Here short/small bar of a graph representing the best result. It means less influence from programming language to CPU.

I. THE APPROACH OF COLLECTING DATA FOR MEASUREMENTS

Our research work presents a scientific way to discern how much CPU works for an application; is really consuming. It concerns that, we are not working with the Embedded Application. We are considering the Customized Application with Database System and this is the reason that we does not depend on Windows Task Manager or any other 3rd party Software/Measurement Tools. We have written a program using C# which can take manipulatable data from kernel (KERNEL32.DLL). The below equations (figure 16) 1 through 4 is representing our coding concepts.

- 1 $U = 100\% - (\% \text{ of time spent in idle task})$
- 2 $\% \text{ time in the idle task} = \frac{(\text{average period of background task with no load}) * 100\%}{(\text{average period of background task with some load})}$
- 3 $\% \text{ time in the idle task} = \frac{(\text{BG Loop counts measured during 25 ms}) * 100\%}{(\text{BG Loop counts expected in an unloaded CPU})}$
- 4 $\text{Actual \%} = \frac{\text{Scaled Percent} * (100\%)}{\text{max value 8 bit value}}$

Figure 16: Equations

The .Net framework has been designed with some system library which is a certified tool for kernel programming [45]. Another important issue is Physical Memory usage. Our collections of each process memory usage are from “Process.WorkingSet64 Property” [45]. This property generates the amount of physical memory, in bytes, allocated for the associated process.

Our aim is to get Response or Execution time using programming output, we set timer before and after our Algorithm/Function to get our desired execution time and the formula of calculation explained at section C.

The work flow of our tool is illustrated below using a Flowchart (Figure 17) according to the exact algorithm stated in Section D.

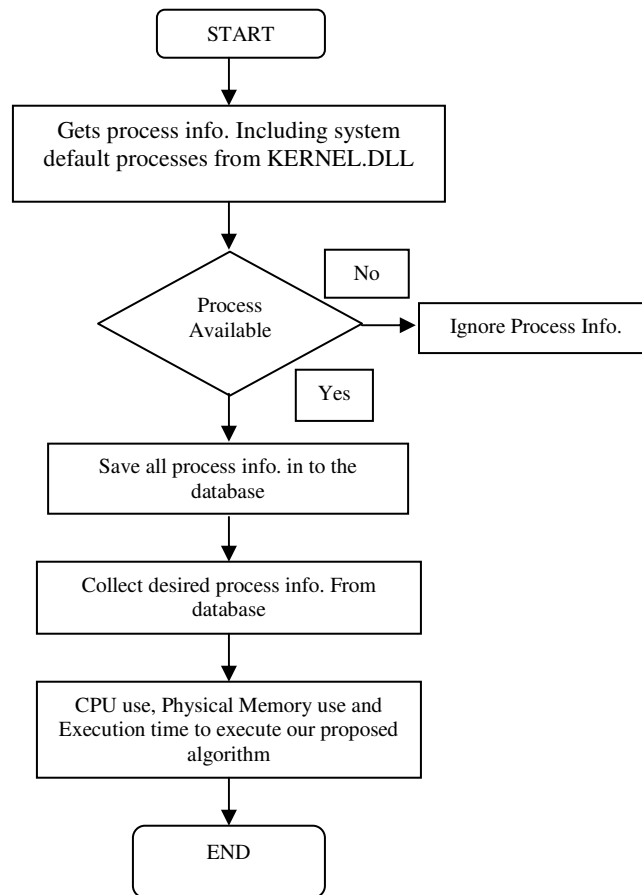


Figure 17: Programming Data Flow

J. CONCLUSION

After doing a comparative study amongst all targeted languages, we can say that JAVA/JSP has the best performance than any other regarding CPU uses, memory uses and execution time.

References:

- [1] *Computer Architecture a Quantitative Approach* [Second Edition] writer John L Hennessy & David a Patterson
- [2] *Quality metrics for SQL queries embedded in host languages*, Huib J. van den Brink, Institute of Information and Computing Sciences, Utrecht University, hjbrink@cs.uu.nl
- [3] *Quality Assessment for Embedded SQL*, Huib van den Brink, Utrecht University, The Netherlands, Email: hjbrink@cs.uu.nl
- [4] Performance Comparison of Alternative Solutions For Web-To-Database Applications, AMANDA W. WU*, HAIBO WANG# AND DAWN WILKINS
- [5] Software Performance Engineering: What Can It Do For You? *Dr. Smith's CMG 2011 Michelson Session Presentation, CMG, Washington National Harbor, Dec. 2011.*
- [6] Software Performance Testing, Xiang Gan, Helsinki 26.09.2006, Seminar paper, University of Helsinki, Department of Computer Science.
- [7] <http://www.cherrystonesoftware.com/doc/AlgorithmicPerformance.pdf> [19-07-2012 2:19 AM]
- [8] Best Practices for Software Performance Engineering, Connie U. Smith, Ph.D., Performance Engineering Services, PO Box 2640, Santa Fe, New Mexico, 87504-2640 (505) 988-3811, <http://www.perfeng.com/>
- [9] Design and Performance Modeling of Component Interconnection Patterns for Distributed Software Architectures, Hassan Gomaa, Dept. of Information and Software Engineering, School of Information Technology and Engineering, George Mason University, 4400 University Drive, Fairfax, VA 220300-4444, USA, hgomaa@gmu.edu, +1-703-993-1652.
- [10] Java programming for high-performance numerical computing, IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598, USA, IBM Systems Journal.
- [11] An empirical comparison of C, C++, Java, Perl, Python, Rexx, and Tcl, Lutz Prechelt (prechelt@ira.uka.de), Fakultät für Informatik, Universität Karlsruhe, D-76128 Karlsruhe, Germany, +49/721/608-4068, Fax: +49/721/608-7343, <http://www.wipd.ira.uka.de/~prechelt/>
- [12] Quantifying Behavioral Differences Between C and C++ Programs, Brad Calder, Dirk Grunwald, and Benjamin Zorn, Department of Computer Science, Campus Box 430, University of Colorado Boulder, CO 80309-0430 USA.
- [13] On the Application of Measurement Theory in Software Engineering, Lionel Briand, Centre de Recherche Informatique de Montréal (CRIM), Software Engineering Group, 1801 McGill College av., Montréal, PQ, H3A 2H4, Canada, e-mail: lbriand@crim.ca.
- [14] Performance Analysis of Information Technology Application of Cooperative Organizations in Cavite, Dr. Ronald L. Pancho, Professor, College of Business Administration, De La Salle University-Dasmari.
- [15] A High-Performance Micro architecture with Hardware-Programmable Function Units, Rahul Razdan and Michael D. Smith, Harvard University, Cambridge, MA 02138, Digital Equipment Corporation, Hudson, MA 01742.
- [16] SOFTWARE PERFORMANCE ENGINEERING, Connie U. Smith¹ and Lloyd G. Williams², ¹Performance Engineering Services, PO Box 2640, Santa Fe, NM 87504, www.perfeng.com, ²Software Engineering Research, 264 Ridgeview Lane, Boulder, Co 80302, (303) 938-847.
- [17] An Introduction to Performance Testing, Chih-wi Ho, Laurie Williams, North Carolina State University, dright@acm.org, williams@csc.ncsu.edu

-
- [18] The Future of Software Performance Engineering, Murray Woodside, Greg Franks, Dorina C. Petriu, Carleton University, Ottawa, Canada. {cmw | greg | petriu}@sce.carleton.ca
- [19] <http://en.wikipedia.org/wiki/Algorithm> [July-10-2012 03:11 PM]
- [20] http://en.wikipedia.org/wiki/Algorithmic_efficiency [July-10-2012 03:17 PM]
- [21] An instruction timing model of CPU performance, Bernard L. Peuto Zilog, Inc., Cupertino, California, Leonard J. Shustek Stanford Linear Accelerator Center and Computer Science Department, Stanford University, Stanford, California.
- [22] Measuring CPU performance in the New Millennium, Henning, J.L. Compaq Comput. Corp., USA.
- [23] <http://www.comentum.com/php-vs-asp.net-comparison.html> [july-10-12 12:30 PM]
- [24] <http://www.websiteoptimization.com/speed/tweak/web-page-performance-thesis/> [19-07-2012 11:47pm]
- [25] http://www.cpu-world.com/news_2012/2012051001_AMD_readies_R-series_embedded_APUs.html [20-07-2012 2:30am]
- [26] http://en.wikipedia.org/wiki/Central_processing_unit [20-07-2012 3:40am]
- [27] <http://www.computermemoryupgrade.net/memory-influence-on-performance.html> [20-07-2012 3:55am]
- [28] <http://xlinux.nist.gov/dads/HTML/bigOnotation.html> [20-07-2012 4:24am]
- [29] http://en.wikipedia.org/wiki/List_of_programming_languages [22-07-2012 5:12pm]
- [30] <http://valums.com/web-apps/> [22-07-2012 6:21pm]
- [31] http://en.wikipedia.org/wiki/Big_O_notation [23-07-2012 4:46pm]
- [32] <http://pages.cs.wisc.edu/~vernon/cs367/notes/3.COMPLEXITY.html> [23-07-2012 5:31pm]
- [33] http://en.wikipedia.org/wiki/Object-oriented_programming [25-07-2012 2:57pm]
- [34] http://en.wikipedia.org/wiki/Relational_database_management_system [25-07-2012 8:03pm]
- [35] Toward a Verified Relational Database Management System, Gregory Malecha, Greg Morrisett, Avraham Shinnar, Ryan Wisnesky, Harvard University, Cambridge, MA, USA, fgmalecha, greg, shinnar, ryan@cs.harvard.edu
- [36] http://en.wikipedia.org/wiki/Client%E2%80%93server_model [25-07-2012 8:45pm]
- [37] <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> [10-08-2012 4:35 pm]
- [38] Loop Recognition in C++/JAVA/GO/Scala, Robert Hundt, Google, 1600 Amphitheatre Parkway, Mountain View, CA, 94043, Rhundt@google.com
- [39] <http://www.favbrowser.com/chrome-vs-opera-vs-firefox-vs-internet-explorer-vs-safari/> [28-09-2012 1:26 am]
- [40] <http://lifehacker.com/5917714/browser-speed-tests-chrome-19-firefox-13-internet-explorer-9-and-opera-1164> [28-09-2012 1:25 am]
- [41] A Brief Tutorial in Traditional vs. OO Programming using java, Richard A. Johnson, CIS Dept., Southwest Missouri State University Springfield, MO, 65804, USA, richardjohnson@smsu.edu
- [42] A Comparative Analysis of Structured and OO programming Methods, Asagba, Prince Oghenekaro; Ogheneovo, Edward E. CPN, MNCS. Department of Computer Science, University of Port Harcourt, Nigeria. pasagba@yahoo.com, edward_ogheneovo@yahoo.com, 08056023566.
- [43] <http://en.wikipedia.org/wiki/ODBC> [28-09-2012 1:26 am]
- [44] <http://msdn.microsoft.com/en-us/library/aa290751%28v=VS.71%29.aspx> [28-09-2012 1:28 am]
- [45] <http://msdn.microsoft.com/en-us/library/system.diagnostics.process.workingset64.aspx> [2013-08-03 09:47 pm]