

The Legend “JAVA” Vs “C++” Programming Languages

Nelapatla Swetha¹, Mulagani Supriya², Ravula Kartheek³

B.Tech Students, Computer Science & Engineering, Rise Krishna Sai Gandhi Group of Institutions, Ongole, India^{1,2}

Assistant Professor, Computer Science & Engineering, Rise Krishna Sai Gandhi Group of Institutions, Ongole, India³

Abstract: This paper develop an examination between two principle programming's utilized as a part of programming applications that are Java what's more, C++, the examination operation incorporates the time expected to play out some calculation i.e. speed of operation, adaptability to altering some code, and proficiency. Similar code is utilized to contrast between the two programming with figure out which one is better. It is found that C++ needs less time to execute similar code contrasting and Java. Java needs around 10% overabundance time to execute similar code portion contrasting with C++.

Keywords: C++; Java; Code Flexibility, Algorithm Speed.

I. INTRODUCTION

Java and C++ are the most utilized dialects as a part of program-ming for the vast majority of software engineers and framework fashioners. Java has a structure called an "Interface". Java interface is practically indistinguishable to a C++ class that has only immaculate virtual capacities. Intrinsic in java is not proficient from more than one base class; regardless of the possibility that the base classes have only dynamic strategies or unadulterated virtual functions. The contrasts amongst Java and C++ can be abridged as in Table (1).

The time expected to execute some code, calculation, program or finish framework program is considered critical in any programming dialect, in this paper the time expected to execute some same code in both Java and C++ is utilized to look at between such two dialects. The base time expected to execution is favorable position be-cause it reflects what amount is the dialect is capable and proficient. The base time of execution means more speed of execution which is the principle objective of any fashioner or developer.

Numerous inquires about and ponders talked about this issue, Lutz Prechelt, (1999), examined the relative effectiveness of Java projects, specifically in contrast with settled execution dialects, for example, C or C++. Java is of-ten considered moderate also, memory-concentrated. Most benchmarks however think about just a solitary implementation of a program in, say, C++ to one usage in Java, dismissing the likelihood that option implementations may look at in an unexpected way. Conversely, the current article exhibits a correlation of 40 diverse implementations of similar program, composed by 40 different developers. The between individual program variances are bigger than those between the dialects and the execution hole amongst Java and different dialects is as yet contracting quickly.

Peter Sestoft, 2010, they think about the numeric performance of C, C# and Java on three little cases. Matured dialects for example, C# and Java are less demanding and more secure to use than customary dialects, for example, C or C++ while controlling element information structures, graphical UIs, etc., Dirk E. et al. (2011), talked about the RC++ bundle streamlines coordinating C++ code with R.

It gives a reliable C++ class chain of command that maps different sorts of R items (vectors, grids, capacities, situations,) to devoted C++ classes. Question interchange amongst R and C++ is overseen by basic, adaptable and extensible ideas which incorporate wide support for C++ Standard Format Library colloquialisms. C++ code can be arranged, connected what's more, stacked on the y, or included through bundles. Adaptable blunder and exemption code taking care of is given. RC++ significantly brings down the hindrance for software engineers needing to consolidate C++ code with R...

Table 1. Contrasts amongst Java and C++

Item	JAVA	C++
1	Java interface is not a class.	While C++ interface is a class.
2	Functions declared within Java interface cannot be implemented using that interface and have no member variables.	In C++ the functions can be implemented using inheritance and there are many options of such implementation using regular inheritance between two variables A and B if we need two copies or one copy then the virtual inheritance can be used.
3	The Clock in Java is an interface.	Where as in C++ it was a class with nothing but pure virtual functions. However the Subject class is quite different.
4	Java uses garbage collection. Garbage collection is a scheme of memory management that automatically frees blocks of memory sometime after all references to that memory have been redirected.	The new object is referred to by the variable "c". Note that "c" is rather like a reference variable in C++, also C++ is often criticized for its lack of GC. However, many people have added garbage collectors to C++.
5	Java does not have templates, which is of some concern to any programmer. In Java, one cannot create a type-safe container. All containers in Java can hold any kind of object. This can lead to some ugly problems.	Templates are a wonderful feature of C++.

Michi H. et al. 2012, thought about between Windows Correspondence Establishment and Java: Remote Technique Summon which are at present observed as real contenders in the middleware space, execution is regularly taken as the sole assessment standard, regardless of the way that performance is stand out of numerous variables that impact the decision of middleware. They gave an execution and versatility correlation of the three middleware stages, and talked about when execution and adaptability matter and when they do, excluding their presumable effect close by different variables on the general cost of a venture. At long last, for those applications that for sure require elite and adaptability, the article calls attention to a couple of strategies you can use to get the greatest value for your money.

II. RESULTS AND DISCOURSE

From last contrasting amongst Java and C++, the primary concern that may uncover which programming is better is an ideal opportunity to execute similar calculation.

So for instance consider the accompanying Java code:

```
public class RealTime {
    public void Do()
    {
        //must complete in 500 µs
        Clock c = new Clock;
        //might collects!
        // diddle with clock for 100 µs } }
```

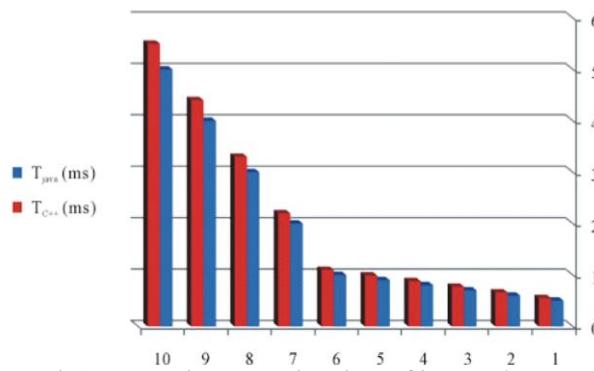


Fig1: comparing execution time of java and c++

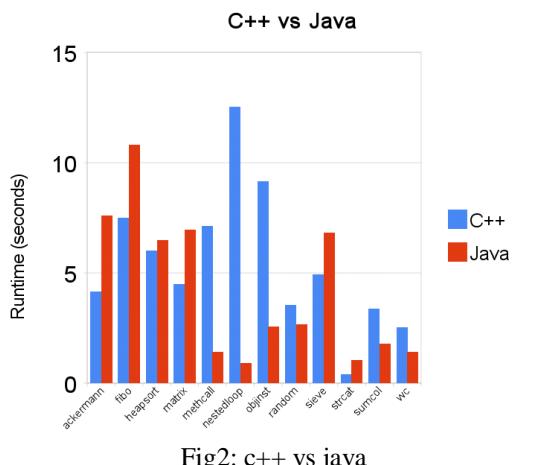


Fig2: c++ vs java

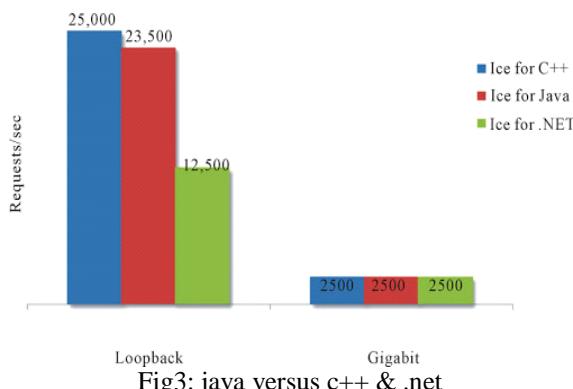


Fig3: java versus c++ & .net

On the off chance that this code is executed utilizing Java and C++, Java takes 500 μ s to be finished such calculation. This is an ordinary limitation in a hard continuous framework. Those capacities that call {Real Time. Do()} depend on the way that it will take no more drawn out than 500 μ s to execute. While similar capacity takes only 450 miniaturized scale seconds to be executed utilizing C++. The objective of the analysis is to measure the time expected to execute similar code on both Java and C++.

$$T_{Java} = 1.10^* T_{C++}$$

T_{Java}: time expected to execute some given code utilizing Java SW.

T_{C++}: time expected to execute some given code utilizing C++ SW.

To contrast the present study and past studies, Figure 2 contrasting Java runtime for various calculations and that of C++, unmistakably the Java runtime is more in practically calculations than that of C++.

Figure 3 contrasting between Java asks for/sec and that for C++ and .NET, unmistakably the C++ asks for per unit of time is more which let us know that C++ is more productive.

III. CONCLUSION

Java is an intense dialect. While C++ has a generally simple time to be learned, and will find that the software engineers appreciate utilizing it. It is noticed that a couple of issues with the dialect in the above exchange. Dialect plan dependably includes a few drawbacks or inadequacies that disappoint somebody. C++ is a fascinating dialect that empowers us to compose codes effectively with more adaptability and with little time expected to execute some code comparing to Java.

REFERENCES

- [1]. J. E. Moreira, S. P. Midkiff, M. Gupta, P. V. Artigas, M. Snir and R. D. Lawrence, "Java Programming for High Performance Numerical Computing," IBM Systems Journal, Vol. 39, No. 1, 2000, pp. 21-56. doi.org/10.1147/sj.391.0021

- [2]. S. Peter, "Numeric Performance in C, C# and Java Peter Sestoft," IT University of Copenhagen, Copenhagen, 2010.
- [3]. E. Dirk and F. Romain, "Rcpp: Seamless R and C++ Integration," 2010. <http://dirk.eddelbuettel.com/>
- [4]. P. Lutz "Comparing Java vs C/C++ Efficiency Differences to Inter-Personal Differences," Communications of the ACM, Faculty at fur Informatic University, Karlsruhe, 1999.
- [5]. W. Peng, M. Sam, M. J. Moreira and G. Manish, "Efficient Support for Complex Numbers in Java," ACM 1999 Java Grande Conference, San Francisco, 12-14 June 1999, pp. 109-118.
- [6]. H. Michi and S. Mark, "Choosing Middleware: Why Performance and Scalability Do (and Do Not) Matter," ZeroC, Inc., Palm Beach Gardens, 2011.
- [7]. J. Glenn, C. M. Clement, S. Q. Snell and G. Vladimir, "Design Issues for Efficient Implementation of MPI in Java," ACM 1999 Java Grande Conference, San Francisco, 12-14 June 1999, pp. 58-65.
- [8]. Volume 3, Issue 3, March 2015, Comparison between Object Oriented Programming Languages: Java and C++ International Journal of Advance Research in Computer Science and Management Studies, www.ijarcms.com

BIOGRAPHIES



Nelapatla Swetha Presently pursuing her B.Tech in CSE Department" in Rise Krishna Sai Gandhi Group of Institutions, Ongole, Prakasam District, A.P, India. Affiliated to Jawaharlal Nehru Technological University, Kakinada. Approved by AICTE, New Delhi. Her research interests are computer networks, OOPS etc.



Mulagani Supriya Presently pursuing her B.Tech in CSE Department" in Rise Krishna Sai Gandhi Group of Institutions, Ongole, Prakasam District, A.P, India. Affiliated to Jawaharlal Nehru Technological University, Kakinada. Approved by AICTE, New Delhi. Her research interests are computer networks, OOPS etc.



Ravula Kartheek, Presently Working as an "Assistant Professor in CSE Department" in Rise Krishna Sai Gandhi Group of Institutions, Ongole, Prakasam District, A.P, India. Affiliated to Jawaharlal Nehru Technological University, Kakinada. Approved by AICTE, New Delhi. His B.Tech completed at VRS & YRN College of Engineering & Technology, Chirala, Prakasam District, A.P, and India. His M.Tech completed in Chirala College of Engineering & Technology, Chirala. His research interests are network security, Computer Networks (wireless Networks), OOPS, Web Designing etc.