Language Comparison in Various Domains

Anupriya Jain*
Seema Sharma**
Sachin Sharma***

Vivekananda Journal of Research
July - December 2023, Vol. 13, Issue 2, 25-34
ISSN 2319-8702(Print)
ISSN 2456-7574(Online)
Peer Reviewed Refereed Journal
©Vivekananda Institute of Professional Studies-TC
https://vips.edu/journal/
VIPS
योग: कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

ABSTRACT

This paper explores various languages used in different domains of computer science. The number of languages has grown in the past two decades and their use cases have grown with them too. This paper tries to explore these domains and the prominent languages being used in them. Developers may find this paper useful in choosing a language for their projects. Beginners in the field of computer science can also learn about the myriad use cases of the language they are learning. We have some packages of R and Python that we use in Machine Learning. This paper studies the use of different languages like C, C++, Rust, R, and Python. The paper attempts to foreground the power and the capabilities of all these languages in various domains. The paper also includes a table comparing all these languages on the basis of their respective features. After reading and searching about all these languages we can conclude that every language has some pros and cons according to its features.

INTRODUCTION

Developers usually find learning a new language a little cumbersome and time-consuming. Given the vast resources available on the internet, it's not always possible to capture relevant information in a short period of time. This paper presents a comparison of some popular programming languages and the domains in which they are popular. This is in no way exhaustive research on the subject, but it explores the various domains in which some of these languages excel. Different use cases of popular languages are explained using

* Professor, MRIIRS, Faridabad. Email : Anupriya.sca@mriu.edu.in
** Associate Professor, MRIIRS, Faridabad. Email : seema.sca@mriu.edu.in
*** Associate Professor, MRIIRS, Faridabad. Email : sachin.sca@mriu.edu.in

various domains. Beginner developers will find this information useful in choosing the language for their application.

DOMAINS

1 Embedded Systems

These systems require low-power consumption devices and hardware. Due to their lack of hardware capacity, they require clever resource allocation and a Programming language that doesn't use up much of the resources from the hardware.

Table 1.1 Features of Embedded Systems Language

Embedded Systems Domain Features	Languages		
	C	Rust	C++
Size (Memory)	It gives access to even the most low-level system components with the help of its built-in pointer. It takes up very little memory.	It is very easy to manage memory as it is very flexible and allows programmers to allocate memory in both dynamic and static ways.	It also offers access to low-level resources like C and makes sure the memory remains free from unnecessary system calls.
Speed	It is very fast and stable.	Not that fast as it takes quite some time to compile.	It is slower than C as it first has to compile code into machine code and then execute it.
Portability	It is simple to port the embedded programs as they can run on any IDE.	It allows the developer to port their programs across a wide range of systems including small micro-controllers.	C++ codes will have portability if the compiler is ported as the codes can only execute on the C++ compiler.

Implementation	C follows a loose data typing policy making it quite suitable.	Rust follows a similar syntax to C++ and hence can be integrated into other existing C/C++ code bases.	The object-oriented nature of C++ helps programmers write the most complex embedded system without overflowing memory.
Secure	It is a secure language.	Rust mainly focuses on type-safety and memory safety and achieves this via principles like ownership and borrowing.	C++ is more secure than C due to its usage of verbose casting, templates, etc.

1.1 C

C has changed the world of programming unlike any other language and is still relevant today. Most of the Linux operating systems are written in C and the original UNIX was entirely written in C.C is portable and efficient. C gives a level of control over hardware like no other language. Memory manipulations are so much easier. When it is required to access arbitrary memory in system applications like operating systems or at the hardware/ software boundary, as in a microcontroller, C is very effective and accurate in manipulating memory locations. C is highly portable, so much so that many modern compilers and interpreters for different languages are written in C such as Python, Ruby, and PHP. Some languages convert their intermediate code to C instead of converting it into machine code. C gives a high level of control over the hardware which makes it easier to write system-level applications in it. It is used in various Embedded Systems.

Embedded systems have limited memory and low storage. C is perfect for such systems since it takes much less memory than any other language. C doesn't require converting code into bytecode like other languages, which makes it faster than other languages. The microcontroller's internal architecture also affects its power consumption. The usage of Pointers enables programs written in C to manipulate memory directly. Support for C is provided out of the box in many OSs. No external libraries need to be installed. It can run on any IDE. C has high support since it is the longest language in the ecosystem.

1.2 Rust

Rust is a low-level programming language developed originally at Mozilla Research by Graydon Hoare. It first appeared around February 2010. It is an open-source language. Rust is a multi-paradigm, general-purpose programming language. Like other low-level programming languages, Rust also doesn't have a garbage collector but it achieves memory safety using two principles: Ownership and Borrowing.

Rust maintains high control over memory and other hardware which enables it to perform well in Embedded Systems, Kernel Programming, and all other tasks that a low-level language can perform. The main difference between Rust and other low-level programming languages is that Rust provides type-safety and memory safety. Using rust over C in developing applications for Microcontroller becomes much easier since there is type and memory safety and Rust has a well-documented official package manager, Cargo, while C lacks an official package manager.

2 High-Performance Computing

This section defines High-performance Computing as anything which requires processing millions of requests at a very high rate. All this includes Machine Learning Algorithms, Game Engines, Cloud and Distributed Systems, High-Frequency Trading, etc.

2.1 C++

C++ excels wherever heavy computation is required to be done in a short amount of time. C++ being a successor of C has all the functionalities and features of C. On top of that, it has classes that make it all the more object-oriented. Classes provide a better way to model real-world objects in C++ which was not possible in C. Extending on the C, it has an almost similar execution time but with added features. C++ borrows most of its functionality from C. Most machine learning libraries like tensor flow, Keras, etc use C++ in their backends simply because of its fast execution time. C++ has rich support for various data structures and algorithms, which makes it easier to develop applications where heavy use of data structures is required. Standard Template Library of C++ provides native execution speed for various data structures. Yes, high-performance C++ code is highly appreciated in the finance industry. It translates to identifying and capturing trades before anyone else. It can also mean you use fewer compute resources to run heavy tasks like Monte Carlo simulations or bond evaluations.

Libraries such as task-flow provide a library for parallel programming.[6] CPP-task flow is written in C++. This library uses a task-graph model to implement parallel programming. The motivation for choosing C++ is the control it provides over memory and threads. Both of which are important factors in parallel programming. Conduit[7] is another library for performing High-Performance Computing Tasks in C++. It uses “best-effort computing” to ease the difficulties faced because of the synchronization of memory and processes. C++ provides the necessary control over the hardware required by such a library.

Another library is Kokkos Core[8], this library provides a model to write high-performance applications which are portable to major HPC platforms. Both data management and parallel code execution are facilitated by it using abstractions.

2.2 Rust

Developed in 2010, Rust is a newcomer to the field of High-Performance Computing (HPC). But even in this short-time that it has been around in the community, it has already grown to tackle various challenges put forth by HPC. The advantage it has over other languages in this domain is the memory safety and type-safety features which are supported out of the box and with the support of external libraries such as GhostCell, it also enables it to overcome the problem of aliasing over mutability, which all the other languages support. Rust also avoids many race conditions owing to its architecture [3]. Rust provides a library for parallelism, Rayon [5]. It makes it easier to convert sequential code into parallel code. It also guarantees data-race freedom.

Many data structures required in High-performance computing are not supported out of the box such as support for complex numbers or multi-dimensional arrays but a lot of libraries support these features which are written by developers in the community, including *num-complex* and *ndarray,* and *nalgebra*.

Libraries such as *cross-beam and parking_lot* provide support for optimized low-level synchronization[4].

Rust has a feasible future in HPC as discussed by Michal Sudwoj [4]. Although there is considerable work to be done in GPU programming. More support for dynamic data structures needs to be provided.

3 Machine Learning

Machine Learning is a type of learning which enables computers to learn about certain patterns or behaviors by going through sample data. A machine learning model trains on labeled sample data and produces output for unlabelled test data with a certain accuracy. The accuracy of the model depends on the type of algorithm chosen, the training time, and the number of parameters chosen for which the model was trained.

Table 1.2 Features of Machine Learning Languages

Machine Learning Domain Features	Languages		
	Python	**R**	**Java**
Description	Python is scalable and open-source. Easy syntax, in-built functions, and wide package support.	R is an open-source programming language. The style of programming is very easy.	Java is a high-level, object-oriented programming language with features like easy debugging, ease of use, graphical representation of data, etc.
Flexibility	It is flexible in the sense that it supports the procedural, functional, object-oriented, and imperative style of programming permitting developers to use any approach they like.	R is highly flexible in the sense that it can work with other tools, also allowing the developers to decide which tool to use for which task.	Java is flexible as applications developed in its coding system can run on any operating system.
Speed	Python allows for extremely fast prototyping speeds but is still slower than Java, C/C++, etc.	R programs are slower than Python.	Java being a compiled language is faster and more reliable than Python.
Portability	Python is a cross-platform language as most of the operating systems come with a pre-installed Python interpreter so its programs can run on different systems such as LINUX, UNIX, etc.	It offers cross-platform compatibility.	Java uses JVM(Java virtual machine) with the help of which code can be written and executed on multi-platforms.

Ideal Usage	Python is considered ideal for fields such as prototyping, scientific computing, sentiment analysis, Natural Language Processing(NLP), and data science.	R offers a diverse array of machine learning techniques such as data visualization, data analysis, models evaluation, etc.	Java is fast executing so it works best for speed-critical machine learning projects.
Application	Python is one of the main languages at Google and is used by Intel, IBM, Pixar, NASA, etc.	Google, Facebook, Microsoft, Mozilla, etc. are using R.	Java is used in data science frameworks such as Fink, Hadoop, Hive, and Spark.
Packages	NumPy, Scikit, Keras, TensorFlow, Sci-Py, Seaborn, Matplotlib, etc.	CARAT, MICE, randomFOREST, dplyr, tidyr, etc.	Weka, Mallet, Apache mahout, Deeplearning4j, etc.

3.1 R

R behaves like a tool, and it is used to make sense of big data. R is useful in research because it processes a large amount of data in less time**.** R is open-source, all of its work and steps are developed by programmers over. You can use various packages in CRAN.

R Machine Learning Packages:-

CARAT:This package performs as Classification and Regression Training. These goals mix training and model prediction. In this package, users can run different algorithms for different problems.

Random Forest:- Random Forest uses multiple random trees to get accurate results. It combines multiple plants to make a forest. Random Forest is mainly used in Machine Learning. Random Forest is also used in the fields of classification and regression. It includes the addition of a variety of trees and then its observation is entered into the decision tree.

Rpart: From viewpoints for recursive partitioning and regression. The purpose of this package is Classification or Regression. This includes a two-stage process. By using binary trees, we can show the output.

KernLab: This package is mainly used for SVM, kernel feature analysis, Gaussian process, and a spectral clustering algorithm and many more methods. KernLab is widely used in SVM methods. In the KernLab package, many problems like clustering, classification, and regression are so difficult to solve. So, KernLab is used to solve these problems.

Nnet: It is the oldest package used in data science. Scientists use this package when there is a need to use an artificial neural network (ANN). This package is widely used and it is easy to implement. This package is restricted to the single layer of nodes in neural networks.

DPLYR: This package is widely used in data science. This package is also used for data manipulation. It provides feasible, fast, and stable functions for data handling. This contains some set of functions like mutate, select, filter, and arrange and this comes under the DPLYR package.

GGPlot2: This package is used for plotting commands. It plots the commands from the given datasets. Ggplot2 is another widely used package in data science. It is the most sophisticated framework among all the R packages available.

Word Cloud: This word cloud specifies the involvement of thousands of words in a single image. Contrariwise, it is a visualization of text data.

Tidyr: This package is also widely used in data science. The purpose of this package in data science is to tidy up the data. The variable is placed in columns and the observation is placed in the rows and the value is placed in the cell.

4.2 Python

Artificial language works in machine learning and has the ability to work and learn without being openly programmed. As we know, that Python is a general-purpose language, so it is able to do any task of machine learning, and then it enables us to make prototypes so we can test any products and algorithms of machine learning.

Data science relies on visionary capability and many more other functions of machine learning (ML) algorithms. Python is the most powerful programming language and it is very useful for scientific computing. The vast accessibility of ML libraries accessible to Python users make it a more attractive solution to illuminate the massive amount of data that is available today.

Some of the Python packages are: -

Matplotlib: As we all know that matplotlib is the oldest package in python that we can use in machine learning. With the help of matplotlib, we can understand a large volume of data through various techniques. It is the standard tool for NLP in Python.

Pandas: We use this for analyzing data. It provides high -performance and easy-to-access data tools for python programming language. We can easily clean our data by using this tool.

Scikit-Learn: This library is widely used by the programmer. We can use this library to select the best tools for our machine learning. Basically, it is built on top of the matplotlib, NumPy, and SciPy, and this library has several tools for data analysis and data mining tasks.

Seaborn: We use this library for making statistical graphs in Python. Basically, we use this package for the visualization of data It is built on top of matplotlib.

NumPy: It is a very popular library in python which is used to add multi-dimensional arrays and matrix processing to Python. It's a large collection of high-level mathematical functions. It is mainly used for scientific computing.

Keras: Keras is a very popular ML for Python. It is used to give fast results and it is able to develop with a focus on enabling fast experimentation. It is mainly designed for humans. It makes Keras easy to learn and easy to use.

SciPy: It is a numerical processing library. It has multiple built-in functions for mathematical problems. It has multiple modules for optimization, linear algebra, integration, and statistics.

Pytorch: It is used for deep learning and it is the most popular library function in Python. This is a Machine learning library that is implemented in C and wrapped in Lua. It was developed by Facebook. Now many organizations use this like Twitter, Salesforce, and many others.

TensorFlow: TensorFlow includes many libraries and it was developed by Google. This is an open-source library and is used for high-performance numerical computation. It is a framework used for creating and running computations involving tensors.

CONCLUSION

The domains in which the discussed languages operate are developing day by day and with them, the requirement for people to have advanced knowledge of these languages is also increasing. The study concludes that all these domains have requirements for their own languages and much work is to be done in the optimization of existing languages and the development of new languages. The huge support for JavaScript shows that it will remain at the top of the developing world for some time. While Rust is an upcoming contender for systems development and high-performance computing, C++ still holds a majority in that domain.

REFERENCES

https://stackoverflow.com/questions/36136201/how-does-rust-guarantee-memory-safety-and-prevent-segfa ults

Costanzo, M., Rucci, E., Naiouf, M., & De Giusti, A. D. (2021). Performance vs programming effort between rust and C on multicore architectures: Case study in N-body. *arXiv preprint arXiv:2107.11912*. https://doi.org/10.1109/CLEI53233.2021.9640225

Sudwoj, M. (2020). *Rust programming language in the high-performance computing environment*. *Bachelor's* [Thesis]. ETH Zürich.

https://docs.rs/rayon/1.0.3/rayon/

Lin, C. X., Huang, T. W., Guo, G., & Wong, M. D. F. (2019, October). A modern C++ parallel task programming library. In *Proceedings of the 27th ACM International Conference on Multimedia* (pp. 2284–2287). https://doi.org/10.1145/3343031.3350537

Moreno, M. A., Papa, S. R., & Ofria, C. (2021). Conduit: AC++ library for best-effort high-performance computing. *arXiv preprint arXiv:2105.10486*. https://doi.org/10.1145/3449726.3463205

https://github.com/kokkos/kokkos

Matsakis, N. D., & Klock, F. S. II. (2014). The rust language. In *Proceedings of the 2014 ACM SIGAda Annual Conference on High Integrity Language Technology (Hilti 14)*. *Academic Medicine, New York, NY, United States*, (103–104)

https://analyticsindiamag.com/top-12-r-packages-for-machine-learning-in-2020/

https://cran.r-project.org/view=MachineLearning

https://www.upgrad.com/blog/top-python-libraries-for-machine-learning/

https://www.activestate.com/blog/top-10-python-machine-learning-packages/