



ONE CALCULATION about JVM and PVM

Prof.S.Mohan,Dept. of Computer Science & Prof. J.Jebaraj,Dept of Mathematics
VSB College of Engineering Technical Campus,Kinathukadavu,Coimbatore,
Tamil Nadu,INDIA.

ABSTRACT

Virtual machines mean simulated system. The physical system within the imaginary as same as. In python does not have JIT (just – in – time) compiler. Instead of AOT (Ahead – of – time) compiler. JIT is dynamic development and default. But AOT performance and production environments pre – compile time.JIT compiler is complexity to extend to core implementation and uses cython. But JIT compiler used languages JAVA and C#. Then JIT are used in run time environment of the languages in Python and JAVA (JRE) in JAVA but PYPY (PYPL) in Python. So defends different OS and CPU architecture used in JVM and PVM like as JIT and AOT compilation.

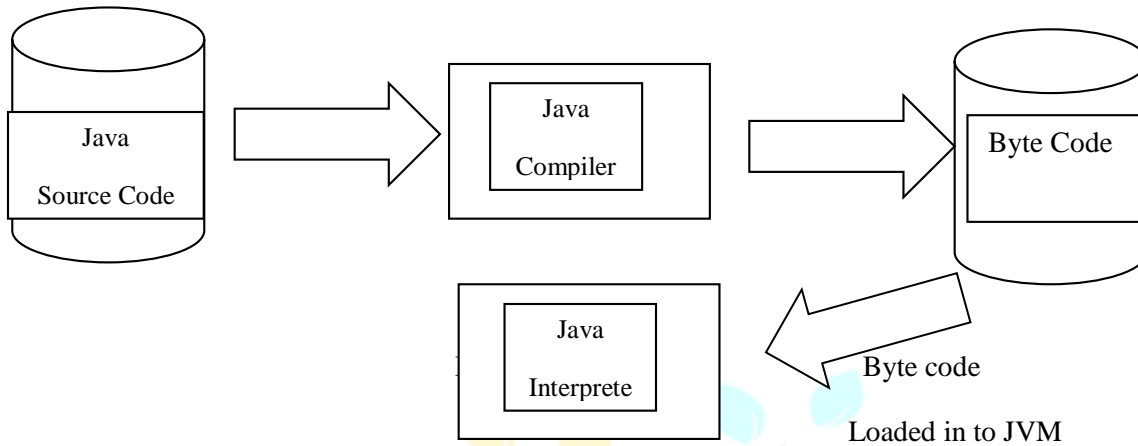
Key: JIT (just – in – time), AOT (Ahead – of – time), cython (C and Python), JRE(Java Runtime Environment), PYPY - PYPL (Python Programming language)

INTRODUCTION

The both JVM and PVM are having the interpreter and compiler. Both are platform (OS) independent translated by byte – code. But compiler to translate the source code in to machine code. The interpreter also does the same process but in different ways. Why need to interpreter while working to compiler? Because error feedback does not give accurate by compiler but interpreter gives.

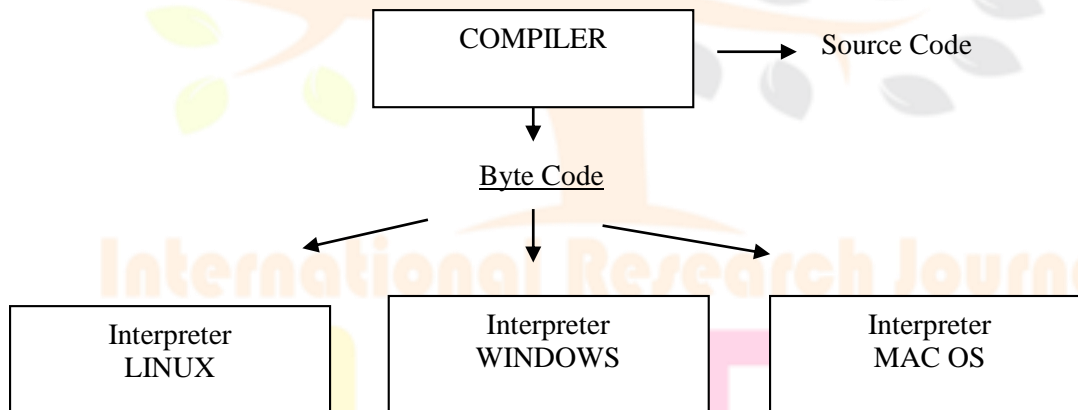
JIT compiler means IO – Bond but it has start up time delay. Optimization of JIT performs. It processes in runtime environment AOT is pre - compilation of the process. Because it is compiled to different CPU. So, it is called cross compiler. A program translates from object to sources that is de – compiler. The transpiler means source – to – source compiler. One language format to convert another one language is called transpilar. So, it is also called transpilar

JVM (Java Virtual Machine)



In other languages Python and JAVA Compiler that convert source code specific machine code. This machine code was OS dependent that varied from one OS to another OS. So they had to design compiler different operating system too difficult. Sun micro system initialize by making a platform independent. Byte code and specific JVM or interpreter.

Why need interpreter?



Similarly, Python uses interpreter as well as the compiler. Translate much similar to Byte code instruction to CPU. They are executed by software called a virtual machine. These are not emulated entire operating system in CPU execution environment.

PVM (Python Virtual Machine)

Python simply run X.PY. Python compiler entirely different from JAVA

- i) JAVA: Run the compiler turn to source code in to compiled classes. Java is called compiler language used – JIT Compiler.
- ii) PYTHON: Run and interpreter from source code into translate. So, Python is interpreter language PYPY in another implementation faster than Cython. Compile language missing the Python interactive

mode(prompt) from JAVA. Both are executed by virtual machine in JAVA and Python. .PYC compiled Python Byte code VM – dependent.

It is varied from Java virtual machine to Python virtual machine. Both are having compiler and interpreter. But also have transpiler and Byte code translator OS independent and language independent and Byte code translator compiler and interpreter also. Because it is differing from JIT and AOT with in JVM and PVM.

JVM and PVM

Just – in – Time compilation (JIT)

JVM is also known as dynamic translation in JIT compiler. That is also known as prior to execution to traditional translation adaptation and optimization best suited dynamic programming languages. JIT compiler source code or byte code to machine code and execution it. Output directly in to memory. But arbitrary (individually) memory can not be executed. Because it does the process at runtime environment.

Ahead – of – Time compilation (AOT)

Its dynamic recompilation source virtual machines. By computing during execution of the system can generate the code to reflect the program runtime environment not available to traditional static compile. Adoptive optimization recompilation performs based on the current execution file. Treat of between JIT compiling and interpreting instructions AOT. Intermediate languages such as JAVA byte code equaling to DOT net CIL (Common intermediate languages). Native code depends or machine code. It is useful to embedded devices. AOT standard native compiler. Because it is precompiled process.

What is different between JIT & AOT?

In JIT and AOT are do at runtime environment. Because JIT is a run time compilation during execution program. But AOT is recompilation of byte code in common intermediate code generator. IT is start up time delay more optimization of JIT performs individually delay will increases will inverse fully. Application code initially interpreter reducing compilation time is significant. Starting initial code interpreter static execution can be collected before compilation. It gives the better optimization.

In AOT compiled program library complex to considerable function of runtime environment. Saving disk space, memory and starting time, embedded devices. It can most case procedure machine of machine optimization code. AOT compiled perform complex and advised code optimization. JIT will consider more too costly cannot usually perform some optimization possible in JIT.

Summary

In JVM and PVM generally are not emulated. But it simulated. Because it is virtual machine work as a physical system as usually work as imaginary. Some process but not emulated. In JVM is the intermediate code generated by byte code. Also same as PVM. But one thing JVM have JIT compilation at runtime. Because it is executed by runtime.

AOT is pre-compilation in byte code to translate depends operating system first. Then it will take compilation in byte code process. Use is interpreter. Because interpret to translate line by line. In AOT is recompilation also available but not in JIT compiler. Because it has better cache utilization. Byte code is rearranged execution code. JIT is after giving the language verses of the speed. In static compile can also take profile instruction as input. Many optimizations are only feasible at runtime.

FUTURE SCOPE

In JAVA and C# used JIT compiler but does not involved in Python virtual machine instead of AOT is fixed with PyPy. It is also precompiled at runtime in JIT is after execution compilation is recompilation is taken. But not in JIT compiler. Because Python compiler is language in dependent also.

So we can do JAVA programing in Python compiler slowly speed to the JIT compiler. Because it has AOT compiler. But not have that feature in JAVA. Because both OOPS languages. So, we need the static compiler performance in JAVA to do the Python compiler the speed will be slow compare JIT compiler. Both are having virtual machine to the same process. But it is different from the compilation and interpretation JIT and AOT. That is differ both in virtual machine.

REFERENCES

- [1] Sun Y, Zhang W. Overview of real-time java computing. J Comput Sci Eng. 2013;
- [2] Overview of Java application configuration frameworks. Int J Open Inf Technol. 2013;
- [3] Glover J, Lazzarini V, Timoney J. Python for Audio Signal Processing. Int J Speech Technol. 2017;
- [4] Travis E. Oliphant (Brigham Young University). Python for Scientific Computing Python Overview. Comput Sci Eng. 2007;
- [5] Fayyad, U., Piatetsky-Shapiro, G., & Smyth P. Python for Scientific Computing Python Overview. Comput Sci Eng. 2007;
- [6] Tyagi VK, Goel R, Singh M, Kumar S. Modeling and analysis of a closed loop supply chain with uncertain lead time in the perspective of inventory management. Int J Sci Technol Res. 2020;
- [7] Kadiyala A, Kumar A. Applications of Python to evaluate environmental data science problems. Environmental Progress and Sustainable Energy. 2017
- [8] How the JAVA virtual machine works by Bikash shaw

