JYTHON- THE CHOICE OF FUTURE

Trishant Malik [1], Sadhana Gopal [2], Seema Devi [3]

[1,2,3]*CSE/Dronacharya College of Engineering/MDU(India)*

ABSTRACT

The world has witnessed a significant and emerging change in computers and their need in our work and lives. The invention of Charles Babbage has now become an essential part of almost every sector, be it industries, education, medical etc. The journey from First generation of computers till now has brought and forgot various programming languages. 1GL(First Generation Language) might be considered as first programming language for computers, a machine level language involving no compiler or translator to program the computer. A large number of developers would have never heard of it. There could be few more similar programming languages which were given existence but people never got familiar with them. The past and present if seen at a glance then we can say that introducing high level programming languages was the significant milestone achieved in the field of computer science. JAVA then can be considered as the big fish in the ocean of high level languages. Its key features i.e. object oriented nature, concurrency, platform independency, security made it stand out in the market without ant notable competition from last many years until Python was introduced in the market. Python's productive and dynamically typing feature has earned it a strong position in the market somewhat similar to JAVA. However, none of them should be considered better since each of them have their own pros and cons. In this paper, we write about the programming language that might become more popular than any one of them in the near future. Jython a java implementation of Python is a programming language holding advantages and disadvantages of both JAVA as well as Python.

Keywords: Jython, Java, Python, Comparison among the three, JNI(Jython Native Interface)

I. INTRODUCTION

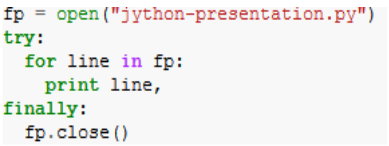
The introduction of Jython begins with research work of Jim Hugunin. While pursuing his masters at MIT he was about to finish his thesis related to superconductor-semiconductor junction considering it as building block of a quantum computer. He had to deal with numeric analysis and calculation for which he was using MATLAB initially but except for numerical data some other functionalities had to be added in his work for which he found MATLAB wasn't sufficient and along with MATLAB he used Python and C language to finish his work and after finishing his work he started working on a numeric extension of Python named NumPy (written in Python and C). Later in 1997 while pursuing his PhD he compared Numeric Python's performance to various programming languages and realised that Java could be a better replacement of C and Python programs could be successfully translated to byte codes of Java and decided to create an aesthetic match of Java and Python which in later years resulted in Jython. Jython's initial releasecame in January 2001 then in 2005 Python Software

Foundation conferred Jython however the development was really slow. In 2008 Sun Micro System hired Ted Leung and Frank Wierzbicki to work on Jython and Jython 2.5 was released in 2009than its stable release Jython 2.5.3 came in August 2012. On 22-August-2014 Jython 2.7 Beta 3 has been released with improved features and bug fixes.

Figure1: Jython Essentials

II. COMPARISON BETWEEN JAVA, PYTHON AND JYTHON

In comparison with Java and Python, Jython is seemingly an efficient language. Jython is a Java implementation of Python that combines expressive power with clarity. Jyhton is freely available for commercial and non-commercial purposes. While implementing Jython in embedded scripting, Java programmers can add the Jython libraries to their system to allow the end users to write simple or complicated scripts that add functionality to the application. Also, Jython provides an interactive interpreter that can be used to interact with Java packages or with running Java applications, through which programmers can experiment and debug any Java system using Jython. Python programs are usually ten times shorter than the equivalent Java program. This translates directly to increased programmer productivity.

Java	*Python*	*Jython*
Speed The performance of Java programs is nearly ten times faster than any other programming language.	**Speed** Python programs are implemented faster than Jython programs, but are slower than Java.	**Speed** With the new stand alone implementation of Jython. Jython is expected to show the same trait as of Java in terms of speed
Productivity In general comparisons Java typically has ten times more code length. For eg: Java Code for file read 	**Productivity** Python is more productive than Java	**Productivity** Jython follows Python in terms of productivity hence is more productive than Java For eg: Jython Code for file read 
Static Typed Language Java is a static typed language For example: str-"String value"; NOTE: This will produce an error if str is not declared before using it.	**Dynamic Typed Language** Without declaring str we can use it in Python	**Both** In some situations static typed language is preferable in others dynamic typed language. Thus Jython is going to be the choice in both situations as it supports both
Multi Threaded Environment Java provides a multi threaded environment which is one of the	**GIL**	**Multi threaded Environment** Jython being a Java

significant advantages of Java		implementation can be used under multi threadedenviornment
Pre-defined Data Structure Although Java is a well structured language possessing various predefined types in comparison to Python it doesn't have some pre-defined objects	**Pre-defined Data Structure** Python is more rich in data structures (List, Dicts, Tuples etc. all are objects)	**Pre-defined Data Structure** Jython users get availability of both data structures Java as well as Python.

Table1: Comparison between Java Python and Jython

III. STRENGTH & WEAKNESS OF JYTHON

Strength

- Prototyping can be done very easily and effectively with Jython.
- Jython implements Beans properties.
- Jython can bind together the libraries which are already written in Java.
- Jython is an excellent option for embedded scripting.
- It allows abstraction of classes.
- It also supports a full oriented programming model.
- Clean and easy to read syntax
- Module-based organisation.
- Extended reach and ease of distribution.

Weakness

- Since Jython is a java implementation thus C libraries Python without JNI(Jython Native Interface) cannot be used in it.
- COM libraries without JNI cannot be accessed
- Several UNIX functionalities cannot be accessed in Jython due to absence of JNI.

IV. CONCLUSION

Jython is a scripting language of unparalleled ease of development with an operating environment for which many powerful tools have been written. The combination can be immensely valuable for programmers, enabling

you to cut development time significantly while still retaining the ability to use the existing Java tools, such as applets and servlets. Just as Python was originally conceived as glue to bind together other programs, Jython acts as glue to help you get the most of Java-based tools. In particular, Jython excels at simplifying the use of complex Java programming libraries and APIs, such as the Swing graphical Interface toolkit or the JDBC Database connectivity API.

REFERENCES

[1]. Hugunin, Jim (March 2002). "Story of Jython". Retrieved 2009-06-05.

[2]. Leung, Fred (2008-03-03). "The Sun is going to shine on Python". Retrieved 2008-03-03.

[3]. Nutter, Charles (2008-01-03). "Jython's Back, Baby!". Retrieved 2008-02-09.

[4]. Baker, Jim (2008-01-03). "Django on Jython: Minding the Gap". Retrieved 2008-02-17.

[5]. Baker, Jim (2008-06-24). "Flipping the 2.5 Bit for Jython". Retrieved 2008-07-12.

[6]. Wierzbicki, Frank (2008-07-15). "Jython 2.5 Alpha Released!". Retrieved 2008-07-16.