An Interesting Investigation towards Understanding of [OpenJIT Compiler Framework + ImageJ - Java Imaging Software] Interaction with JikesRVM [RVM- Research Virtual Machine] in the Context of [Byte Code Engineering Library(BCEL) + IoT + High Performance Computing (HPC)] related Java based Heterogeneous Image Processing Environments – A Simple Suggestion & Technical Notes.

[“OpenJIT is by no means a toy or partially functional prototype”]

[Bytecode Engineering & its related issues are often ignored by many** - Bytecode is one of the most important features of java that aids in cross-platform execution **]

Nirmal Tej Kumar

Independent Consultant Informatics/Imaging/Photonics/Nanotechnology/HPC – R&D.
R&D Collaborator USA/UK/Israel/Japan/Germany/BRICS Group of Nations.
Current Member ante Inst,UTD,Dallas,TX,USA.
email id hmfg2014@gmail.com

[I] Inspiration & Introduction :

“ *OpenJIT* is an open-ended, reflective JIT compiler framework for Java being researched and developed in a joint project by Tokyo Inst. Tech. and Fujitsu Ltd. “

[Source : *https://dl.acm.org/citation.cfm?id=713479*]

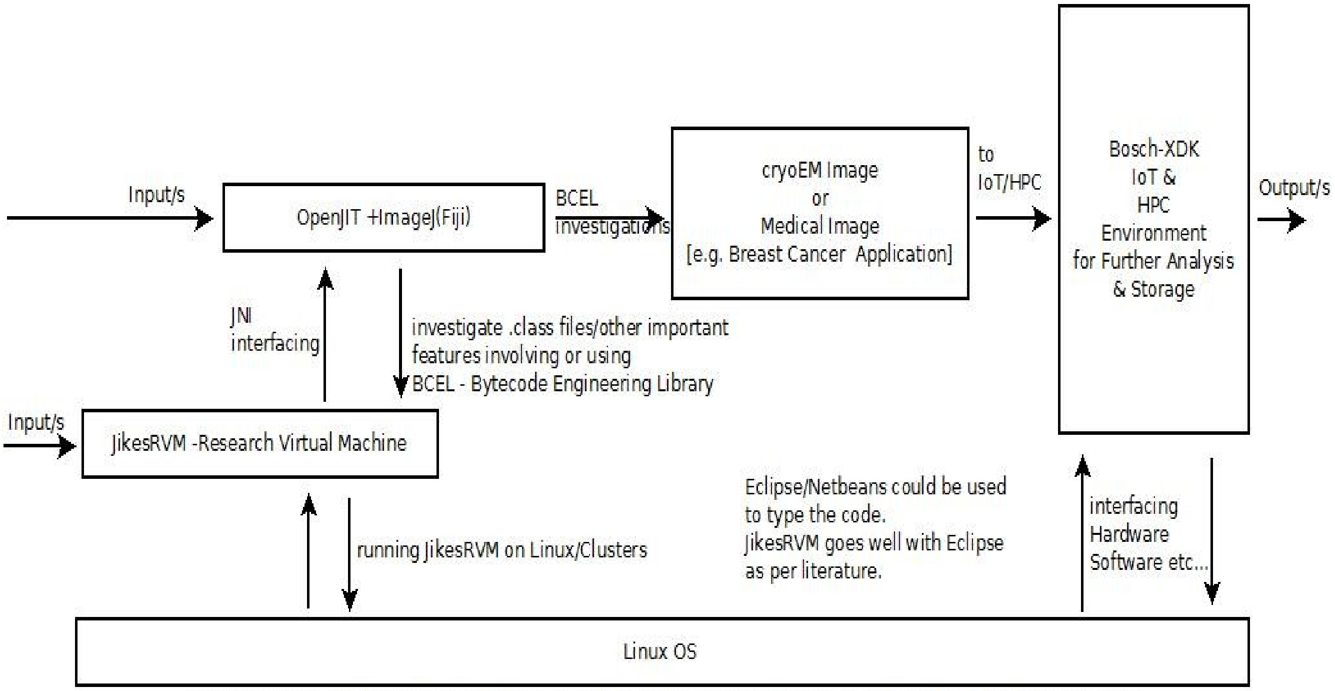
“ An open platform for scientific image analysis. ... ***ImageJ*** **is an open source image processing program designed for scientific multidimensional images.** ***ImageJ*** **is highly extensible, with thousands of plugins and scripts for performing a wide variety of tasks, and a large user community. “**

[Source : *https://imagej.net/Welcome]*

https://www.knime.com/community/imagej
https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1934-z
https://www.nature.com › nature methods › historical commentary

https://www.future-science.com/doi/full/10.2144/000114351
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.7279Manipulating Java Class Files with BCEL - Part One : Hello World! - http://www.geekyarticles.com/2011/08/manipulating-java-class-files-with-bcel.html
https://www.geeksforgeeks.org/just-in-time-compiler/ - Nice Explanation.

[II] OpenJIT-ImageJ-JikesRVM [RVM - Research Virtual Machine] Informatics Framework :


Input/s
OpenJIT +ImageJ(Fiji)
BCEL
investigations
cryoEM Image
or
Medical Image
[e.g. Breast Cancer Application]
to
IoT/HPC
Bosch-XDK
IoT &
HPC
Environment
for Further Analysis
& Storage
Output/s
JNI
interfacing
investigate .class files/other important
features involving or using
BCEL - Bytecode Engineering Library
Input/s
JikesRVM -Research Virtual Machine
running JikesRVM on Linux/Clusters
Eclipse/Netbeans could be used
to type the code.
JikesRVM goes well with Eclipse
as per literature.
interfacing
Hardware
Software etc...
Linux OS

Approximate OpenJIT-ImageJ-JikesRVM-BCEL Based Image Processing Informatics Framework - A Simple Suggestion.
Testing in progress -Please Check & Satify Yourselves
Thanks - Dr.Nirmal
[Not endorsing any commercial products used in our R&D project - there could be alternative approaches]

Figure I – Our Total Overview of R&D presented. Approximate Suggestion for ‘Non–commercial’ R&D Only. Actual Implementation will vary to large extent. Java suits IoT/HPC interfacing very well hence this simple & humble presentation. [Please Note : We have used ImageJ, Fiji could be used as well]

Some direct useful links for readers :

[a] https://www.semanticscholar.org/author/Nirmal-Tej-Kumar/12354503/suggest

[b] http://vixra.org/author/nirmal_tej_kumar

[c] http://vixra.org/author/nirmal

[d] http://vixra.org/author/n_t_kumar

[e] http://vixra.org/author/d_n_t_kumar [Good information on Image Processing being used by researchers internationally]

{ Written in free style to inspire & encourage others in these related domains. Testing in Progress. Thanks for understanding – Dr.Nirmal. }

[III] Related R&D Information on Software Used :

[a] Information on OpenJIT can be found on http://www.openjit.org/.

[b] *https://pdfs.semanticscholar.org/8d39/c85e6c97891c143ae6e3fcc9b905e9b4c240.pdf*

[c] https://www.usenix.org/conference/jvm-01/openjit-2-design-and-implementation-application-framework-jit-compilers

[d] *https://www.jikesrvm.org/*

[e] *https://commons.apache.org/bcel/*

[f] https://github.com/apache/commons-bcel

[g] *https://www.ibm.com › Learn › Java development*

[h] *https://dzone.com/articles/byte-code-engineering-1*

[i] *https://www.archlinux.org/packages/extra/any/java-bcel*

[j] *www.java2s.com › Jar File Download › bcel*

[k] http://dmakarov.github.io/work/guide/

[l] https://www.researchgate.net/publication/
3821514_cJVM_A_single_system_image_of_a_JVM_on_a_cluster
{ DOI: 10.1109/ICPP.1999.797382 · Source: IEEE Xplore }

[m] https://openjdk.java.net/groups/compiler/

[n] *https://imagej.nih.gov/ij/download.html && https://imagej.nih.gov/ij/*

[o] http://openimaj.org/ && https://fiji.sc/

[p] https://commons.apache.org/components.html

[q] **“Just in time compiler in Java -** The **JIT compiler**. The **Just-In-Time** (**JIT**) **compiler** is a component of the runtime environment that improves the performance of **Java**™ applications by **compiling** bytecodes to native machine code at run **time**. ... When a method has been compiled, the JVM calls the compiled code of that method directly instead of interpreting it.”
https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.2019/com.ibm.java.vm.80.doc/docs/jit_overview.html

[IV] Acknowledgment/s :

“Special Thanks” to all who made this happen in my LIFE. Non-Profit Academic R&D Only.
Please read instructions completely before using OpenJIT/JikesRVM Software/BCEL.

[V] Reference/s :

[1] *matsu-www.is.titech.ac.jp/sites/default/files/.../CTT100451809.pdf*

[2] *www.dotnet-guide.com/jit2.html*

[3] *https://dblp.uni-trier.de/pers/hd/m/Maruyama:Fuyuhiko*

[4] *https://marcusdenker.de/old/OpenJit.html*

[5] https://dblp.org/pers/hd/s/Shimura:Kouya

THE END.