Java & AI
OpenJDK projects

JCP Working Group: Java & AI
Created for Java Community Members to share in their communities
Prepared by Zoran Sevarac, Deep Netts Technologies

About

These slides are created to promote and share information about AI related work within Open JDK projects.

Use them to disseminate information in your Development teams, communities, Java User Groups (JUG) and Java related events.

We continuously evolve Java to meet your future app development needs

Java has 30 years of experience evolving with the latest tech trends

Data-centric World

Amber

Continuously improve developer productivity through evolution of the Java language.

ZGC

Create a scalable low-latency garbage collector capable of handling terabyte heaps.

Cloud-powered World

Loom

Massively scale lightweight threads, making concurrency simple again.

Leyden

Improve the start-up time and time to peak performance of cloud applications.

AI-driven World

Babylon

Extend the reach of Java including to machine learning models and GPUs

Panama

Safe and efficient interoperation with native libraries and native Java.

Valhalla

Unify primitives and classes to improve productivity and performance.

Java and AI - the big picture

Java Platform

- New Java platform features enable safe, scalable and performant usage of native libraries, foreign platforms and specialized hardware within Java.
- They will enable Java to be the platform of choice for running AI workloads, offering powerful AI development platform
- These features are being developed as **OpenJDK projects**.

Ecosystem

- These features will enhance development of Java based AI frameworks, libraries and tools, that will provide specific AI related features.

Open JDK Projects

- Panama - Interconnecting JVM and native code (C, Python, ..), and simplifying calls to AI and native libraries.
  - Vector API - Using CPU vector operations to accelerate mathematical operations used for AI/machine learning (part of Panama)
- Babylon - Extend the reach of Java to foreign programming models such as GPUs (code in Java, convert to hardware specific parallel code in runtime)
  - HAT (Heterogenious Acceleration Toolkit) How Babylon and Panama Enable Java GPGPU
- Valhalla Combining abstractions of the object-oriented programming with the performance characteristics of primitive types.

All together, they make Java platform faster, more suitable for numerical computing, and easier to use hardware acceleration.

Project Panama

Foreign Function and Memory API

API that simplifies native function calls in Java (c, Tensorflow, Python)
jextract - open source tool to create initial Java methods from .h files

How it will influence AI development:
Simplify development and usage of native libraries used in machine learning (mostly highly efficient numeric computation), and AI/ML libraries directly

Note: It's not out of the box solution for AI, since generated code needs to be refactored, and wrapped into more Java-like API. Otherwise it will look like a C api with lot's of native memory pointers (MemorySegment classes).

However it completely removes need for writing and compiling C code (as it was the case with JNI)

https://openjdk.org/projects/panama/

Vector API

What is the Java Vector API?

The **Java Vector API** is a feature in Java that helps your programs **run faster** by doing **many operations at the same time** — especially for **numbers and math**.

This is done using a technique called **SIMD** (Single Instruction, Multiple Data). Instead of adding numbers one by one, SIMD can add many numbers at once — like doing multiple calculator operations in a single step.

Why is it useful?

It's especially helpful in:

- Image processing
- Machine learning
- Physics simulations
- Anything that deals with big arrays of numbers

https://openjdk.org/jeps/469

Babylon

What is Project Babylon?

Project Babylon is effort to **make it easier for Java developers to write code that runs efficiently on GPUs and AI hardware**, which are often used for training and running large AI models.

How is it used?

Java developers will be able to:

- Write regular Java code
- Use new language features to **offload compute-heavy tasks to the GPU**.
- Still enjoy Java's safety, portability, and tooling—but with **AI-grade performance**.

Project Babylon = Java + GPUs + AI acceleration

It helps Java developers build high-performance AI apps **without switching to other languages** like C or CUDA.

Valhalla

Aims to unify primitive types and classes, to improve performance and preserve productivity.
Enhance the Java language and JVM with data layout and type system features, so Java can better handle performance-sensitive workloads in areas like numerical computing and high-performance collections.

Its main goals are to make Java's data model more flexible without breaking existing code, and to reduce the overhead of Java’s traditional object model when you don’t need it.

How it will help AI?

It will unlock high performance numerical computing for Java, while preserving key object oriented benefits.

HAT (Heterogenious Acceleration Toolkit)

In Simple Terms: Normally, Java runs on the CPU. But today's computers also have powerful GPUs and other chips (like NPUs or FPGAs) that can handle certain tasks much faster than the CPU — especially for AI, machine learning, and scientific computing.
JHAT helps Java code "offload" some of that heavy work to those accelerators, without needing to write low-level code in C or CUDA.

What JHAT Does: Analyzes your Java code to find parts that can benefit from acceleration (like matrix operations). Compiles those parts to run on accelerators (GPU, etc.). Manages communication between your Java code and the hardware behind the scenes.

Why It's Important: Brings Java closer to the performance of native AI/ML frameworks.
Lets Java developers use high-performance hardware without leaving the Java ecosystem.
Helps modernize Java for the AI era.

https://cr.openjdk.org/~psandoz/conferences/2024-JVMLS/JAVA_BABYLON_HAT-JVMLS-24-08-05.pdf

Approaches for Using Java for AI

1. **Use as a Web service - serve model as web service**
   You can build a model in any language, using a model is language agnostic,
   But might have potential issues with latency and scalability.

2. **Use wrappers or native libraries from Java**
   Portability issues with native dependencies, memory/stability issues, limited scalability, distribution and maintenance overhead.

3. **Use Java native AI libraries**
   Highly scalable, low latency, easy to use, integrate with existing development and production environment, and distribute on large scale.
   Out of the box models mostly not available.

Java AI Ecosystem

JSR 381

Tornado VM

DJL

Deep Netts

Langchain4j

SpringAI

Helidon – 4.1 AI Features

Wanna help? - Call To Action

- You don’t have to become Open JDK contributor, just join the mailing list and share feedback
- Which Open JDK features are relevant to your favorite library?
- Have you tried them? Let us know your experience with the Early Access builds of the project(s)
- Be very specific about your experience and use case - what and how, with examples
- Start initial discussion on Open JDK mailing lists
- Prompts for the mailing lists- prepare questions, what would be helpful, what do we want to know

Feedback on the OpenJDK projects

As a part of our efforts to support and promote Java Platform for AI development, in the JCP Java & AI Working Group https://github.com/jcp-org/Java-and-AI-Working-Group

- How do you see <OpenJDKProject> relates to AI and do you have any specific feedback you want to incorporate?
- Are there any examples of this?
- Were there any AI related discussions on the project's mailing list that you would like to point out?
- What would be ideal feedback from the community related to trying EA builds of <OpenJDKProject> ? Have you received any feedback that would be a good example of this?
- Is there any specific topic or prompt for feedback that you would like us to start on the mailing lists?
- What information might be able to gather through the JCP Java & AI working group could be helpful to the project?