

# Onwards and upwards: Presto's JDK upgrade plan

# Presto is outdated

Outdated dependencies  
currently in the Presto build<sup>1</sup>

88%

Years since the official release  
of Java 8<sup>2</sup>

10

Source: 1. According to the aggregate report from the maven  
versions plugin on PrestoDB master as of 2024/12/03.

502/507 dependencies can be upgraded

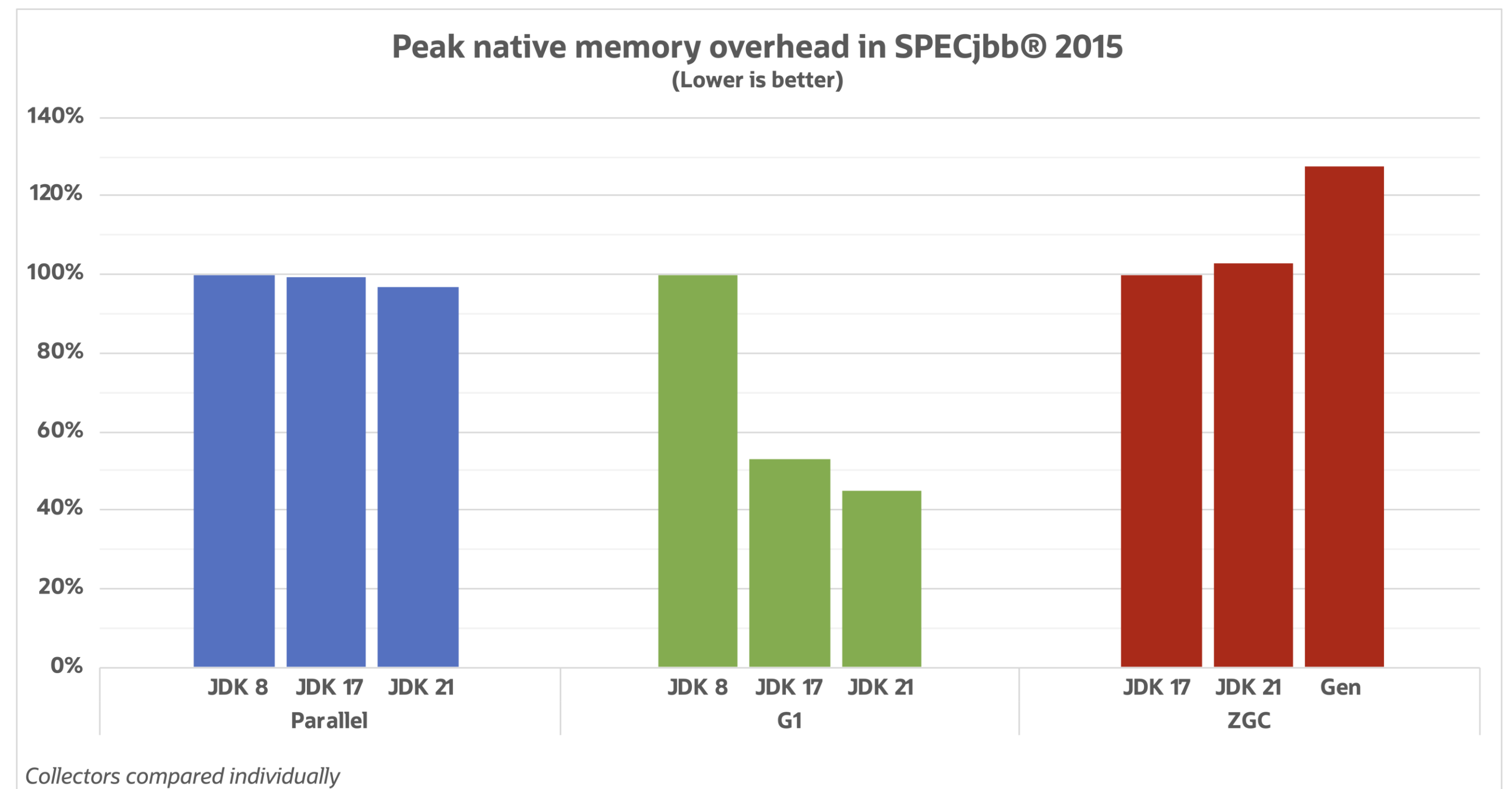
Source: 2. [https://en.wikipedia.org/wiki/Java\\_version\\_history](https://en.wikipedia.org/wiki/Java_version_history)

# JDK upgrades bear fruit

- New language features improve developer productivity & performance
  - Sealed types
  - Pattern matching
  - Record classes
  - Multi-line text blocks
  - Improvements to NPE error messages
  - Virtual threads (async IO!)
- Enhanced APIs for performance & safety
  - Foreign memory API
  - Vector API for SIMD

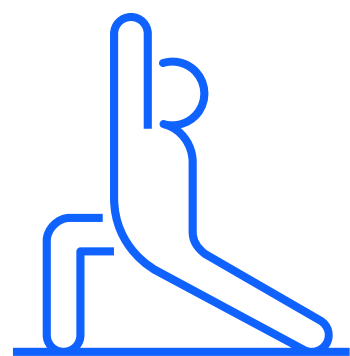
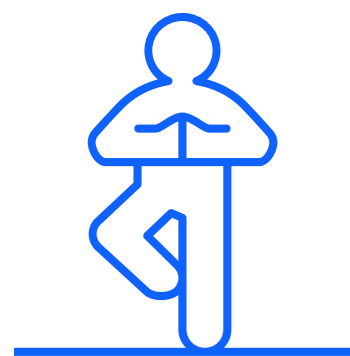
# JDK upgrades bear fruit

- GC improvements
  - ZGC GA in Java 17+
  - Latency improvements
  - Throughput improvements



Courtesy of Stefan Johanssen, Oracle: <https://kstefanj.github.io/2023/12/13/jdk-21-the-gcs-keep-getting-better.html>

# Current Java migration Plan



1

Dec-Jan '25, Presto 291

Code Compatibility: 8  
Runtime Compatibility: 8 & 17

2

Feb-March, Presto 292

Code Compatibility: 8  
Runtime Compatibility: 8 & 17

Deprecate Java 8 for runtime  
and build.

3

Apr-May, Presto 293

Code Compatibility: 17  
Runtime Compatibility: 17, 21

Add support for new language  
features up to 17. Drop support  
for Java 8

4

Aug-Sept, Presto 295+?

Roadmap planning for  
upgrading to Java 21-level  
bytecode and support for the  
next LTS Java release (Java 25,  
Sept '25)

# Get Involved!

Lots of work to accomplish the migration.

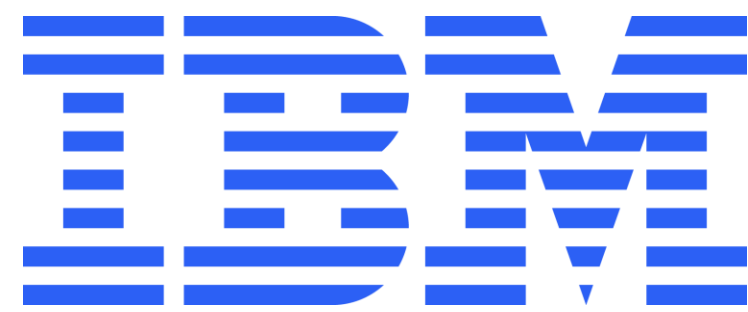
- Feature branch for dual 8/17 support:
  - **upgradeJava17**
  - build & most tests passing
- Feature branch for 17-level bytecode & code modernization:
  - **upgradeJava17Bytecode**
  - Build mostly, passing, still need to run tests


Plenty of work to be done!


How to help:


1. Run your workloads with JDK 17 and report issues
2. Inform your teams of the impending upgrade schedule
  - a. Plan & prioritize accordingly
3. Performance testing using Java 17 w/ new GC
  - a. Communicate results in TSC meetings, or via slack
4. Contribute to the migration efforts on GitHub:
  - a. <https://github.com/orgs/prestodb/projects/31>
  - b. [#jdk21-upgrade](#) in PrestoDB Slack





 #jdk21-upgrade

 <https://github.com/orgs/prestodb/projects/31>

 zac@ibm.com