ORACLE

GraalVM

Native Image: GraalVM Enterprise Edition vs. GraalVM Community Edition

GraalVM Edition feature and benefit comparison

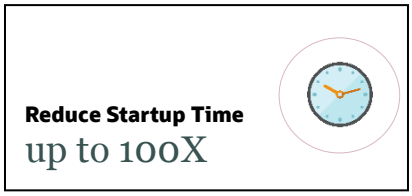
April, 2023, Version 1.0

Public

Introduction

This document highlights additional Native Image capabilities only available in the Enterprise Edition of GraalVM.

GraalVM Native Image compiles Java and other Java Virtual Machine (JVM) language applications **ahead-of-time** to generate native executables for Linux, Windows, and macOS. Native executables run with almost instantaneous startup times and use very little CPU and memory resources—making them ideal for microservices and other modern containerized workloads. For this reason, GraalVM is supported by popular microservice framework vendors such as Micronaut, Spring Boot, Helidon, and Quarkus.



GraalVM Availability and Licensing

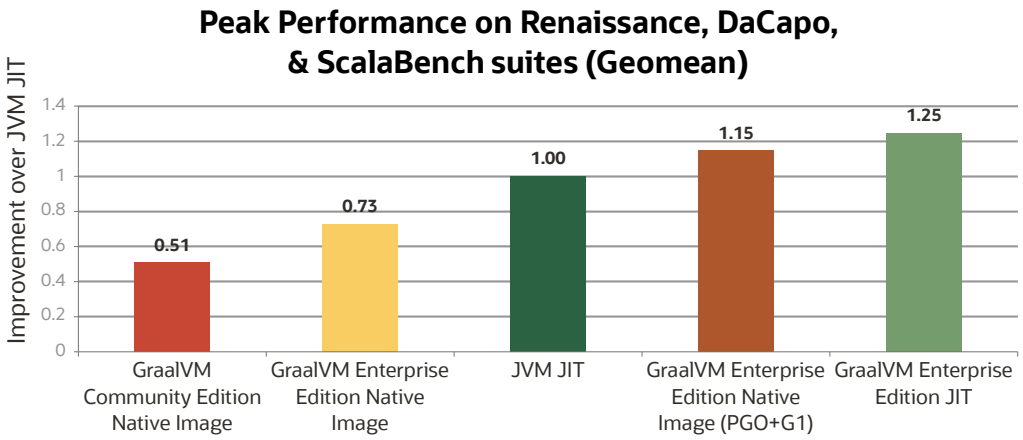
- GraalVM Community Edition is GPL licensed open source software.
- GraalVM Enterprise Edition is available free of charge with your Java SE subscription license.
- GraalVM Enterprise Edition may be used free of charge on Oracle Cloud Infrastructure.

Optimal Application Performance

GraalVM Enterprise Edition provides improved efficiency with greater peak performance and a reduced memory footprint. In fact, using GraalVM Enterprise Edition you can see an application performance improvement of up to 125% over GraalVM Community Edition. These improved performance metrics rely on sophisticated compiler optimization phases, aggressive code inlining, advanced partial escape analysis, and other techniques that produce extremely efficient machine code and avoid heap memory allocations. Fewer heap objects lead to reduced time required for garbage collection, which frees up CPU, which in turn leads to increased application throughput.

Profile-Guided Optimization for Higher Throughput

Profile-guided optimization (PGO) results in additional performance gains, faster startup times, and minimal memory footprint. With PGO, you can collect profiling data about your application in advance and then provide it to Native Image, which uses this information to optimize the resulting native executable. PGO makes it possible for applications to achieve the same or better throughput for an application deployed as a native executable compared with the same application deployed on the JVM with Just-In-Time (JIT) compilation. The figure below illustrates that on a broad set of benchmarks, native executables compiled using GraalVM Enterprise Edition using both PGO and the G1 garbage collector run up to 15% faster than on the JVM with the default C2 JIT compiler. Out of the box performance for native executables compiled using GraalVM Community Edition is approximately 50% slower than JIT compilation. Native executables compiled using GraalVM Enterprise Edition can be significantly faster than their GraalVM Community Edition counterparts.



Lower Resource Requirements

A native executable compiled with GraalVM generally has a much lower memory requirement than its corresponding Java application because it doesn't include any JIT compilation infrastructure (such as the compiler itself, compilation data structures, metaspace class files, profile data cache, and dynamic code cache). Native executables compiled with GraalVM Enterprise Edition require the least amount of memory thanks to optimizations such as compressed pointers. Optional use of the G1 garbage collector (only available in GraalVM Enterprise Edition) increases memory needs as it trades space for speed, but still delivers a significant reduction in memory usage.

High Performance Garbage Collector Options

For managing the Java heap, GraalVM provides several garbage collector (GC) options. Serial GC is the default GC in both GraalVM Community and Enterprise Editions. It is optimized for low memory footprint and small Java heap sizes. The G1 garbage collector, only available in GraalVM Enterprise Edition, is a multi-threaded GC that is optimized to reduce "stop-the-world" pauses, and therefore improve latency, while achieving high throughput. To enable G1, use the option `--gc=G1`. Currently, G1 can only be used in native executables that are built on Linux for x64. Having multiple options for GC enables you to configure and tune the performance needs of your application based on its specific characteristics.

Enterprise Security

Software supply chains remain vulnerable to exploitation as researchers and adversaries frequently discover security vulnerabilities in widely used software libraries. When designing hardware, engineers often assemble a bill of materials (BOM) that itemizes all the individual components in a design. Each entry may contain information relevant for auditing and assembling a design, such as the manufacturer or layout constraints. In contrast, it is often difficult to identify the origin of a piece of software, which may be represented as an executable file, a compressed archive, or a layered filesystem run by a container orchestration engine. This can make it difficult for developers and operators to periodically review the integrity of their software supply chain and limit the attack surface available to adversaries.

A Software Bill of Materials (SBOM) is one way to capture and document the provenance of software components used in an application. Using GraalVM Enterprise Edition, you can generate and embed an SBOM in a native executable—the SBOM can be extracted and examined to scan for known vulnerabilities. With support for the CycloneDX SBOM format and tool support from Syft and Gryft, native executables compiled using GraalVM Enterprise Edition are compatible with common security scanners making them easy to interrogate and check for published CVEs (Common Vulnerabilities and Exposures).

Production Support

GraalVM Enterprise Edition provides full production support via your Java SE subscription. Our 24x7 support includes access to quarterly performance, scalability, and security updates and allows you to log and resolve GraalVM issues quickly and efficiently. Oracle's production support reduces time to resolution, mitigates risk, and minimizes support costs, which enables maximum application uptime and reduces exposure.

Summary

Today, many enterprises continue to innovate to retain customers and modernize applications while controlling IT expenses. GraalVM is an enabling technology that improves the success of these IT initiatives. While both GraalVM editions offer notable improvements and benefits, GraalVM Enterprise Edition offers significant and additional capabilities that warrant its use in the most performance conscious innovative enterprises.

GraalVM Enterprise Edition has been under development in Oracle Labs for 10 years. It has a dedicated support and development staff and is a strategic component of Oracle's Java development plans. Using GraalVM Enterprise Edition, Java clients will continue to be at the forefront of Java innovation, and in a position to provide the most efficient and current solutions to their customers.

ORACLE

GraalVM

Appendix

Summary—High Value Capabilities *only* available in Enterprise Edition

Feature	Description	GraalVM Community Edition	GraalVM Enterprise Edition
Production Support	• Access to quarterly performance, scalability, and security updates.	No	Yes
High-Performance Garbage Collector Options	• Option to use G1 garbage collector (currently Linux x64 only).	No	Yes
Profile Guided Optimization	• Collect profiling data in advance; and then provide the data to Native Image, which uses it to further optimize the performance of the resulting native executable. • Additional performance gain, higher throughput. • Combines the performance benefits of the GraalVM JIT compiler with the "startup" and "footprint" benefits of the GraalVM AOT compiler.	No	Yes
Advanced Optimizations	• Additional patented optimization techniques which enable GraalVM to generate faster code and eliminate unnecessary memory usage, which lowers memory needs and reduces garbage collection overhead.	No	Yes
Advanced Tuning Options	• Command-line options to improve many aspects of performance.	No	Yes

ORACLE

GraalVM

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle