new relic

2023年 Javaエコシステムの現状

最も人気のあるプログラミング言語の一つを徹底解説

レポート / 2023年4月

目次

03 概要

04 Java 17のユーザーが1年間で430%増加

05 Java 14は、非LTSバージョンで最も普及

06 Amazon - 現在、最も人気のあるJDKベンダー

07 周辺のものすべてを支配するコンテナ

- コンテナでの計算設定
- コンテナでのメモリ設定

09 ガベージイン・ガベージアウト

10 手法

11 New Relicについて

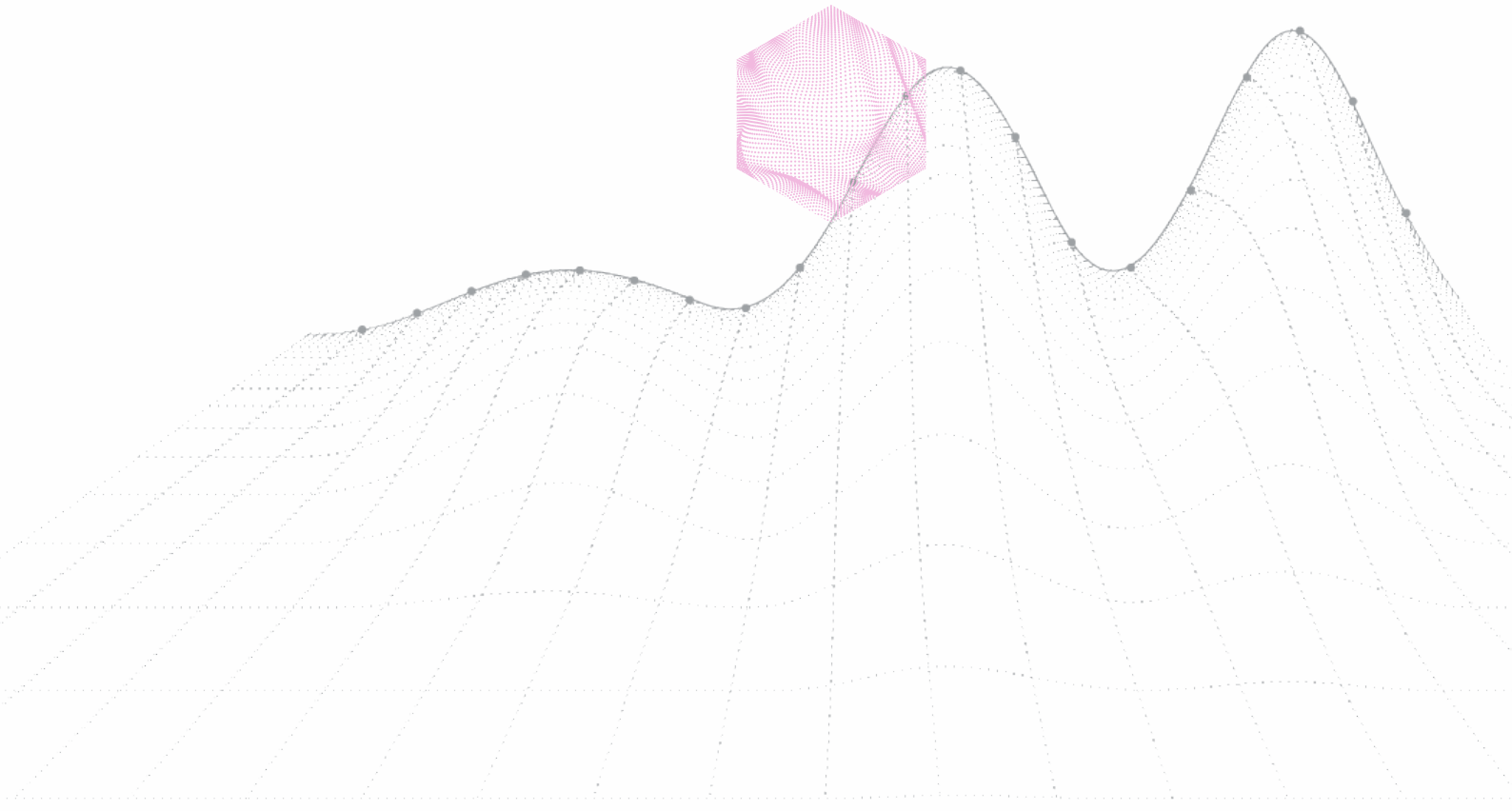
概要

Sun Microsystem社は1996年にJavaプログラミング言語をリリースし、最新のマルチメディアアプリケーションを開発する上で、よりポータブルでインタラクティブな方法を提供しています。それ以来、Javaは最も人気のあるプログラミング言語の一つとなっています。Javaはプラットフォームに依存せず、何千ものライブラリを提供し、サポートも充実しているため、ほぼすべての主要産業や経済分野で使用されています。

New Relicは過去数年間、Javaエコシステムに注目し、新しいバージョンのリリースとコンテナの普及によるJavaの使用方法の変化を明らかにしてきました。2023年のレポートでは、Javaエコシステムの現状についての内容とインサイトを提供します。

以下のカテゴリーについて検証しました。

- 実運用で最も使用されているJavaのバージョン
- 最も人気のあるJDKベンダー
- コンテナの台頭
- 最も一般的なヒープサイズ設定
- 最もよく使われるガベージコレクションアルゴリズム



Java 17の採用が1年間で430%増加

Javaの長期サポート（LTS）は2〜3年ごとにリリースされており、四半期ごとのリリースでは、安定性、セキュリティ、パフォーマンスの更新のみが提供されています。

現在、56%以上のアプリケーションが実運用でJava 11を使用しています（2022年の48%および2020年の11%から増加）。Java 8は僅差で2位、実運用でそのバージョンを使用しているアプリケーションの33%（2022年の46%から減少）を獲得しています。

Java 11は2年連続でトップの座を維持していますが、Java 17の採用率は、Java 11導入時の数字をはるかに上回っています。現在、9%以上のアプリケーションが実運用でJava 17を使用しています（2022年の1%未満から上昇）、1年間で430%の成長率を表しています。Java 11がそのレベルに近づくには何年もかかりました。

Java 7のサポートが2022年に終了したため、0.28%のみのアプリケーションが実運用でJava 7を使用しています。Java 7を使用しているアプリケーションのほとんどは、アップグレードされていないレガシーアプリケーションです。

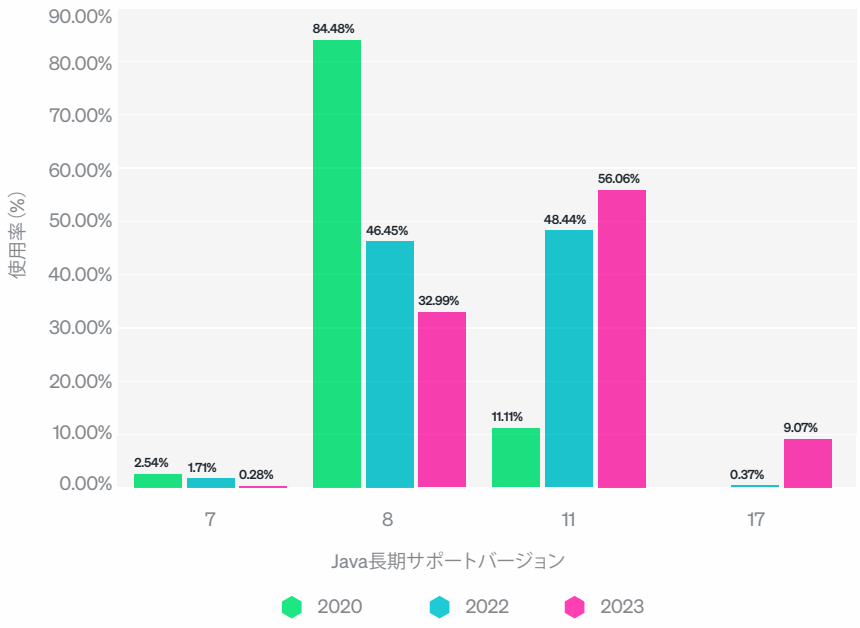


図01. Java LTSバージョンの割合

Java 14は、非LTSバージョンで最も普及

Java 9から、プラットフォームのリリースパターンが変更されました。およそ半年ごとにJavaの新しいバージョンが利用可能になりますが、次のリリースまでしかサポートされません。その意図は、新しい機能をより頻繁に利用できるようにするためです。

暫定的な非LTSバージョンのJavaの採用率は、実運用中の LTSバージョンと比較して非常に低く、非LTSバージョンのJavaを使用しているアプリケーションはわずか1.6%に過ぎません（2022年の2.7%から減少）。

LTS以外のバージョンの使用の減少に影響を与えている要因には、次のようなものが挙げられると考えられます。

- サポート不足
- 機能の魅力が十分に理解されていない
- 次のLTSがリリースされるまでの期間

バージョン8とバージョン11の間では、次のLTSバージョンのリリース時期が明確ではありませんでした。現在ではタイムラインが確定しており、2〜3年短縮されています。次のLTSバージョンは23ではなく、21になると予想されています。

使用されている非LTSのJavaバージョンの中で、最も利用されているのはJava 14（0.57%、2022年の0.95%から低下）で、Java 15は僅差で2位（0.44%、2022年の0.70%から低下）。

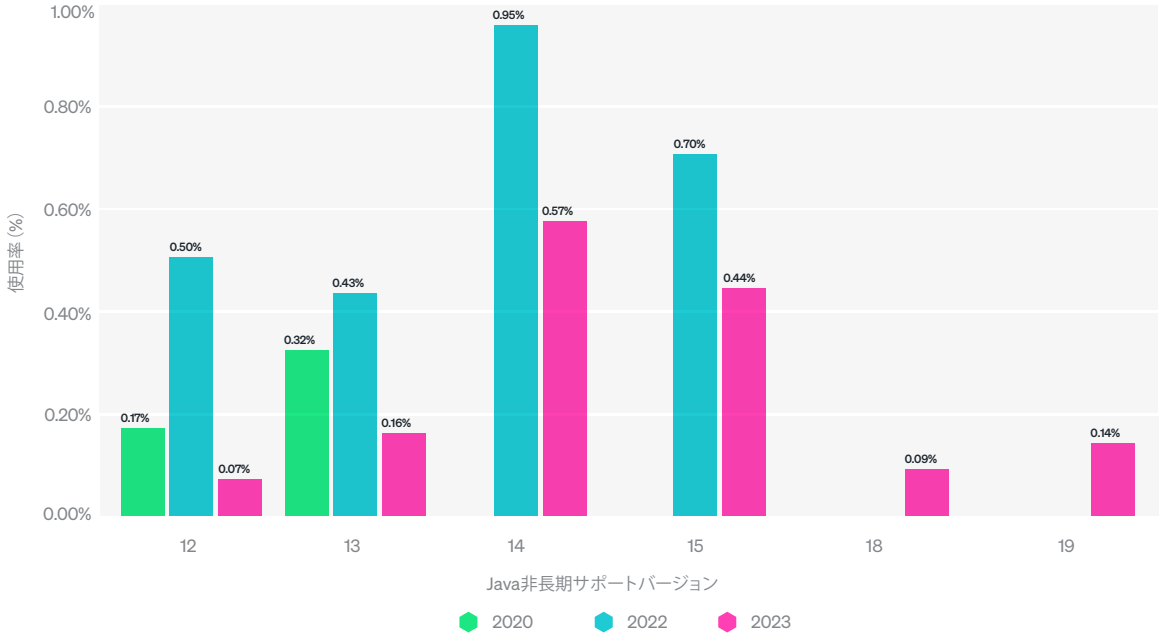


図02. 使用中のJava非LTSバージョンの割合

Amazon―現在、最も人気のあるJDKベンダー

近年、利用されているJava Developer Kit（JDK）ディストリビューションのソースが変化しています。かつて多くの開発者がオラクルからJDKを入手していたのに対し、OpenJDKプロジェクトでJavaがオープンソース化されたことで、豊富な選択肢が生まれました。

2020年には、オラクルが最も人気のあるJDKベンダーで、Java市場のおよそ75%を占めていました。JDK 11ディストリビューションのライセンスがより制限されるようになり（Java 17でよりオープンなスタンスに戻る前）、オラクルのバイナリから離れる顕著な動きがありました。オラクルは首位は維持しているものの、2022年には34%あったシェアが2023年には28%に落ち込んでいます。

Amazonの利用は市場の31%にまで劇的に増加し（2020年2.18%、2022年22%から上昇）、最も人気のあるJDKベンダーになりました。

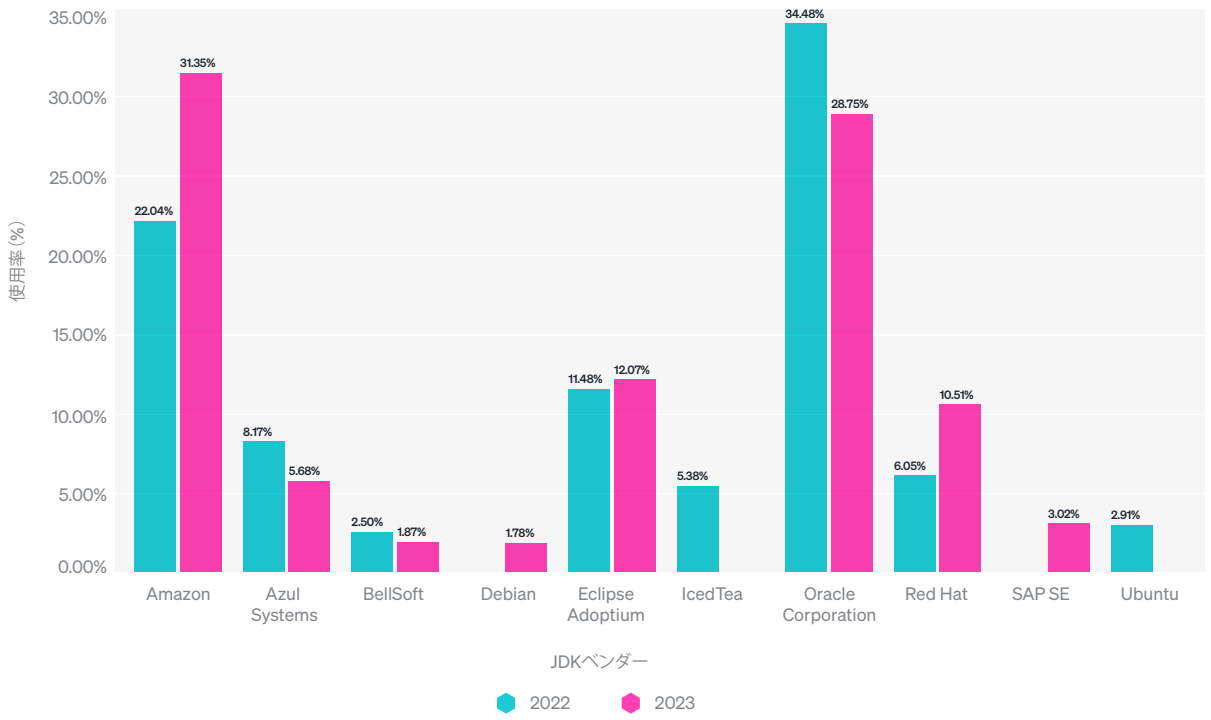


図03. 使用中のJDKベンダーの割合

周辺のものすべてを支配するコンテナ

アプリケーションのコンテナ化が主流になっており、New Relicにレポートするアプリケーションの70%以上が、コンテナからレポートしています。

コンテナでの計算設定

コンテナは、エンジニアリングチームが計算とメモリのリソースをどのように割り当てるかについて、影響を与えます。例として、New Relicのデータによると、コンテナ内では、4コア未満で実行するアプリケーションの割合が非常に高いことを示しています。

エンジニアリングチームはコンテナのシングルコア設定から離れ、使用率はわずか36%（2022年の42%から低下）で、マルチコア設定に移行しており、29%以上が8コア設定を使用（2022年の20%から増加）しています。

通常、エンジニアリングチームは、コンテナをデプロイすることが多いクラウド環境では、より小規模な計算設定を使用します。しかし、この傾向はアプリケーションによっては思わぬ問題を引き起こし、コンフィギュレーション低下の一因となっている可能性も考えられます。例えば、チームが1つのCPUのみで実行されている場合、ガベージコレクターを明示的に設定したとしても、期待通りのガベージコレクターを取得できない可能性があります。

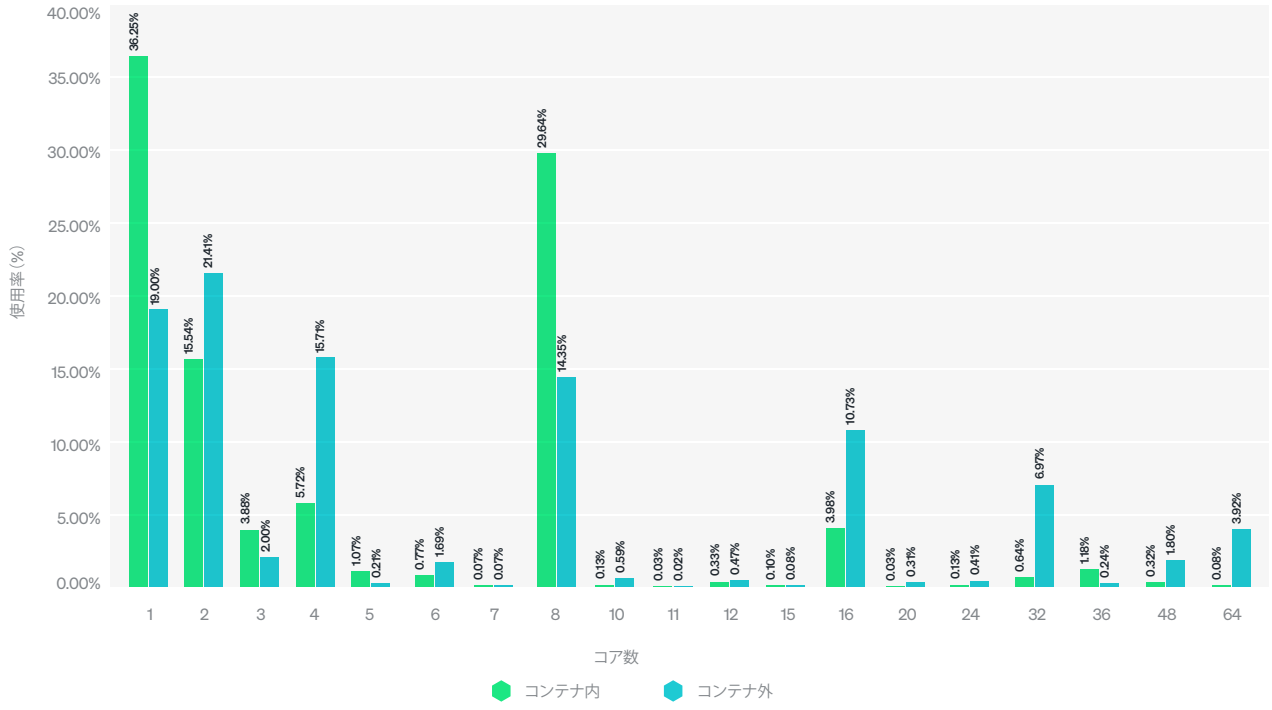


図04. コンテナ内外で実行するJavaアプリケーションのコア数別の割合

コンテナでのメモリ設定

メモリ設定の比較でも同様の傾向が見られ、コンテナ内のインスタンスが小さくなる傾向が見られます。コンテナをデプロイするという性質上、制限がより厳密に適用されるため、開発者は自身のフットプリントをより意識するようになります。

Java 9では、コンテナとの連携を改善するための多くの機能が導入されました。`-Xmx`による正確なヒープサイズの指定を置き換える`-XX:MaxRAMPercentage`起動フラグなどです。Java仮想マシン（JVM）はコンテナのメモリ制限を認識しているため、`-XX:MaxRAMPercentage` でJVMをコンテナサイズに簡単にスケーリングします。

New Relicのデータによると、コンテナ化されたアプリケーションの30%がJVMメモリの上限を`XX:MaxRAMPercentage`フラグによって明示的に（2022年の9%から増加）要求しているため、このフラグは普及しつつあります。

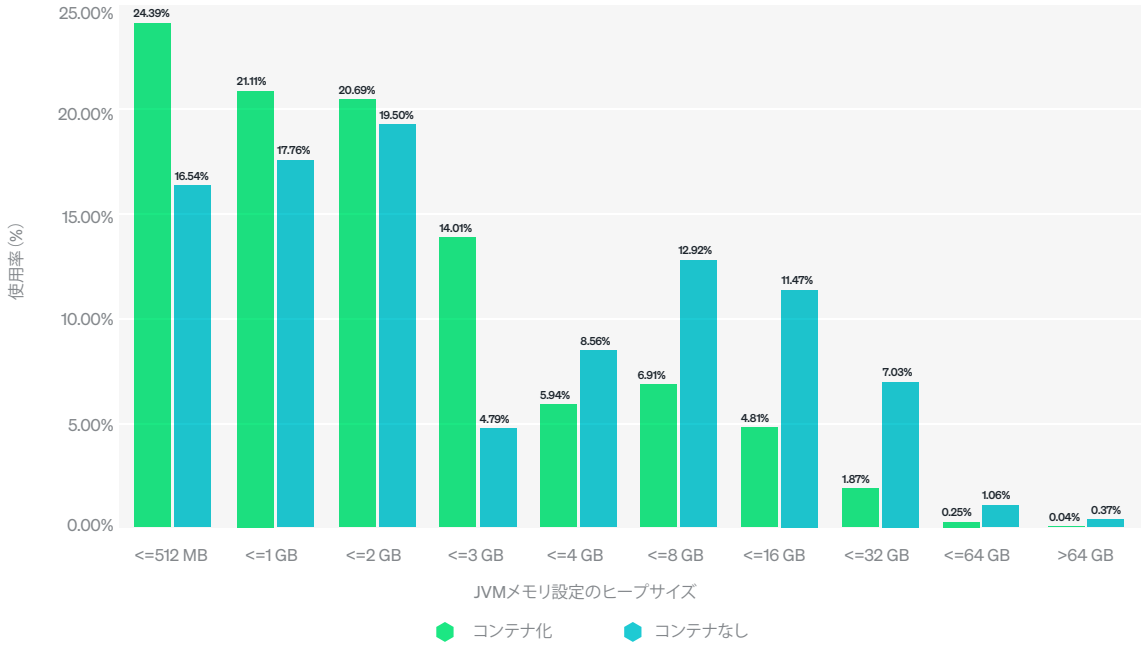


図05. JVMメモリ設定のヒープサイズ別の、コンテナを使用する場合と使用しない場合のJavaアプリケーションの割合

ガベージイン・ガベージアウト

自動ガベージコレクションは、ヒープメモリを調べて、使用中のオブジェクトと使用されていないオブジェクトを特定し、未使用のオブジェクトを削除するプロセスです。JVMのパフォーマンスにおいて中心的な役割を考えると、ガベージコレクションは、依然としてJavaコミュニティで注目されています。

New Relicのデータによると、Garbage-First (G1) ガベージコレクターは、Java 11以降のバージョンを使用しているユーザーの間で依然として絶大な人気があり、顧客の65%が使用しています。G1の主な利点の1つは、大きな領域を一度にクリアするのではなく、小さな領域をクリアすることです。これにより、収集プロセスが最適化されます。また、実行がフリーズすることはめったになく、若い世代と古い世代の両方を同時に収集できるため、エンジニアにとって優れた標準設定になっています。

Java 8以降に登場した他の実験的なガベージコレクター（ZGCとShenandoah）は、実運用システムでの使用は多くありません。どちらも実運用対応のリリースがありますが、一般的な処理での使用はまだごくわずかです。

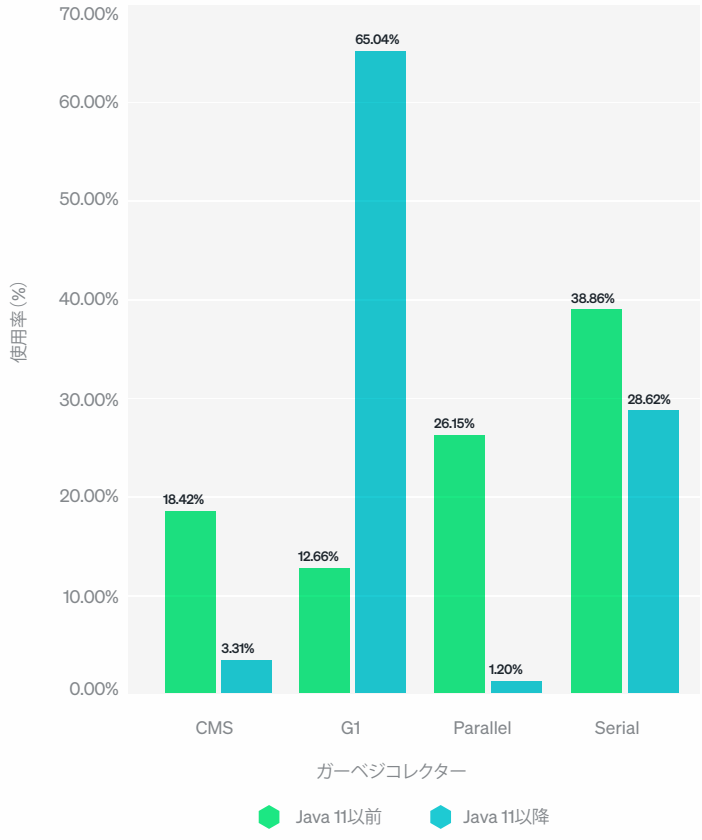


図06. Javaバージョン別の使用中のガベージコレクターの割合

手法

本レポートのデータは、2022年1月にNew Relicにレポートしたアプリケーションのみから取り出したものであり、Javaの使用状況の全体像を示すものではありません。New Relicは、適切なデータを匿名化し、意図的に粗視化することで、Javaエコシステムの一般的な概要を提供します。攻撃者やその他の悪意のある者の役に立つような詳細情報は、意図的に本レポートから除外されています。

さっそくNew RelicでJavaデータのモニターを始めましょう。

Javaクイックスタートをインストール

New Relicについて

New Relicは、オブザーバビリティのリーダーとして、優れたソフトウェアの計画、構築、デプロイ、運用に対するデータ駆動型のアプローチによりエンジニアを支援しています。New Relicは、メトリクス、イベント、ログ、トレースからなる全テレメトリが集約された唯一の統合データプラットフォームを、強力なフルスタック分析ツールと組み合わせて提供し、データに基づくエンジニアのベストパフォーマンスを可能にします。

New Relicは、直感的で予測可能な業界初の使用量ベースの価格体系によって提供され、計画サイクルタイム、変更失敗率、リリース頻度、平均解決時間（MTTR）の改善を支援することにより、エンジニアに高い費用対効果をもたらします。これにより、世界をリードする大企業や成長著しいスタートアップ企業のアップタイムと信頼性、運用効率の向上を助け、イノベーションと成長を加速させる優れたカスタマーエクスペリエンスの創出を支援します。

new relic.