New Relic

Rapport

État des lieux 2022 de l'écosystème Java

Une analyse approfondie des langages de programmation les plus populaires

Avril 2022

Table des matières

Présentation

L'industrie logicielle moderne est vaste et les langages de programmation ne manquent pas. Java est incroyablement populaire auprès des développeurs de logiciels, et il est utilisé dans quasiment tous les secteurs industriels et économiques majeurs, parce qu'il fonctionne sur n'importe quelle plateforme et peut passer aisément d'un ordinateur à un autre ; de plus, il est bien pris en charge.

En mars 2020, New Relic a publié son premier rapport sur l'état des lieux de l'écosystème Java basé sur des données rassemblées à partir d'applications fournissant des données de performance. La sortie récente de Java 17, la première version avec prise en charge à long terme (LTS) depuis Java 11, permet de réexaminer ces données. Pour créer ce rapport, New Relic a anonymisé et rendu intentionnellement grossières les données appropriées afin de donner une idée générale de l'écosystème Java. Toute information détaillée susceptible de pouvoir aider des pirates et d'autres personnes malintentionnées a été délibérément exclue du rapport.

Ce rapport a pour but de fournir un contexte et des informations sur l'état actuel de l'écosystème Java.

Les catégories suivantes ont été examinées :

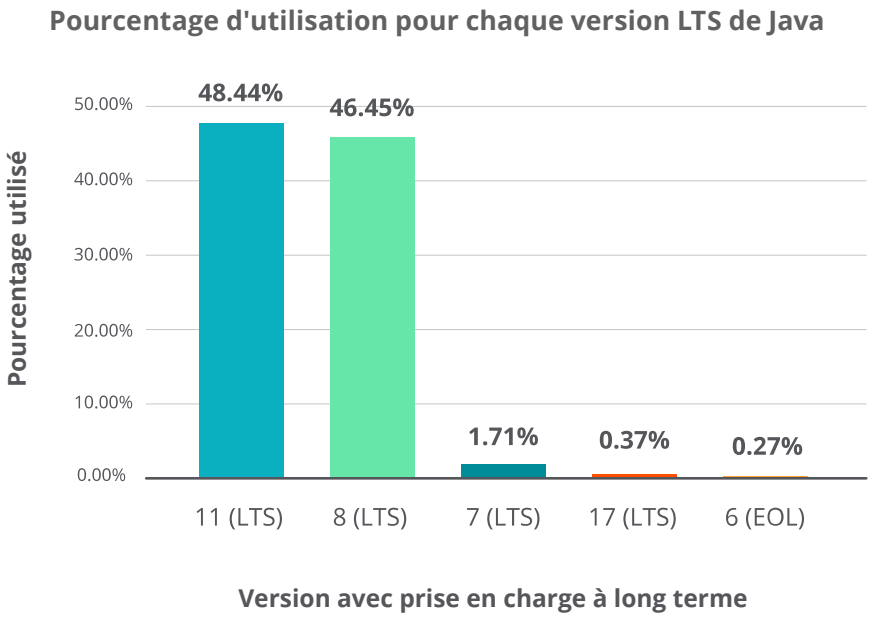
- Version la plus utilisée en production
- Fournisseurs les plus populaires
- Montée en flèche des conteneurs
- Configurations des tailles de tas les plus courantes
- Algorithmes de garbage collection les plus utilisés

Java 11 est la nouvelle norme

En 2020, alors que Java 11 était disponible depuis plus d'un an, la grande majorité des applications étaient toujours exécutées sur Java 8 (84,48 %). Depuis, la tendance s'est inversée entre ces deux versions LTS. Plus de 48 % des applications utilisent maintenant Java 11 en production (contre 11,11 % en 2020), suivi de près par Java 8, utilisé par 46,45 % des applications en production.

Java 17 ne fait pas partie du palmarès, mais, dans les quelques mois qui ont suivi sa sortie, il a déjà dépassé les versions Java 6, Java 10 et Java 16.

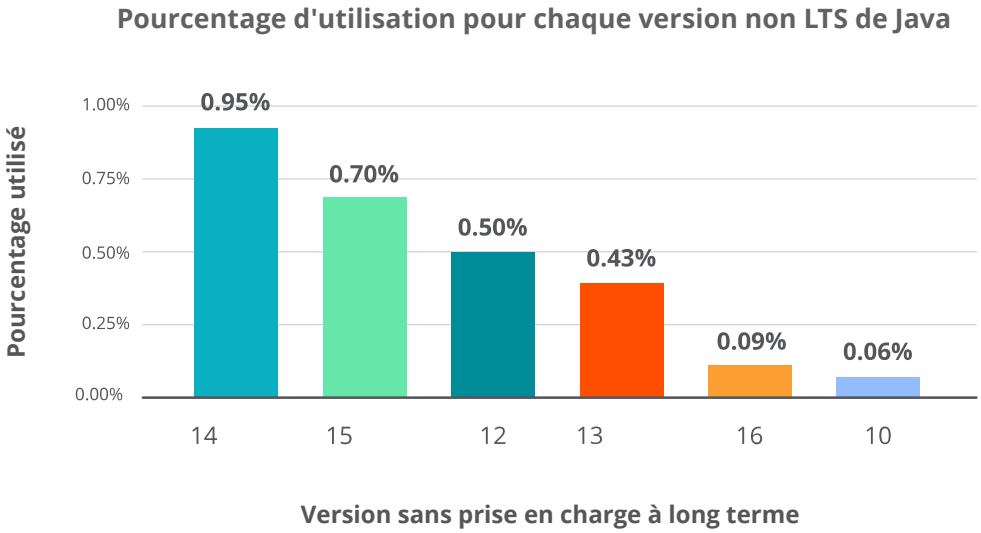
La prise en charge de Java 7 se termine en 2022, et, pourtant, 1,71 % des applications l'utilisent toujours en production. Java 6, quant à lui, n'est plus pris en charge, mais 0,27 % des applications l'utilisent. La plupart des applications qui utilisent Java 6 et Java 7 sont des applications anciennes qui n'ont pas été mises à niveau.



Java 14 est la version non LTS la plus populaire

À partir de Java 9, le rythme de sortie pour la plateforme a changé. Tous les six mois, une nouvelle version de Java était disponible, mais ces versions étaient prises en charge uniquement jusqu'à la sortie de la version suivante. Le but était d'offrir de nouvelles fonctionnalités plus souvent.

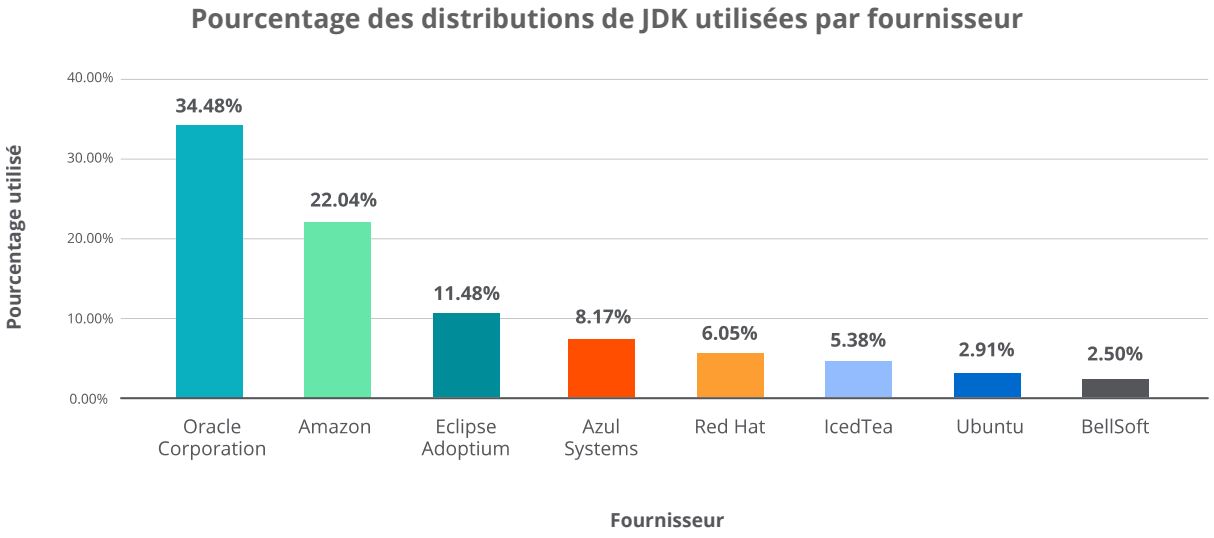
Cependant, le taux d'adoption de versions non LTS provisoires de Java reste extrêmement faible par rapport aux versions LTS en production puisque 2,7 % seulement des applications utilisent des versions non LTS de Java. Si des fournisseurs tels qu'Azul Systems fournissent des correctifs sur certaines versions non LTS, ce n'est pas le cas pour la plupart des fournisseurs. Ceci explique sans doute la réticence à effectuer une mise à niveau. Parmi les versions non LTS de Java utilisées, Java 14 est la plus populaire, et Java 10 et Java 16 sont au bas du tableau.



Oracle en perte de popularité, Amazon en hausse

Ces dernières années, la source des distributions de JDK (Java Developer Kit) utilisées a changé. Auparavant, bon nombre de développeurs obtenaient leurs JDK auprès d'Oracle, mais le nouvel accès open source de Java dans le projet OpenJDK offre maintenant une multitude d'options.

Le tableau ci-dessous montre la distance prise avec les exécutables Oracle suite à la restriction de l'octroi de licences pour la distribution de JDK 11 (avant le retour vers une position plus ouverte avec Java 17). En 2020, Oracle était le fournisseur le plus populaire, occupant 75 % environ du marché Java. Si Oracle demeure en première position, sa part a diminué de moitié. Amazon a grimpé de façon spectaculaire pour occuper 22 % du marché (contre 2,18 % en 2020).



Depuis novembre 2021, nous avons observé une inversion intéressante de la tendance de ces chiffres outre la distanciation générale avec Oracle. Avant la sortie de Java 17, Eclipse Adoptium et Amazon occupaient des positions inverses quasiment identiques dans ce classement.

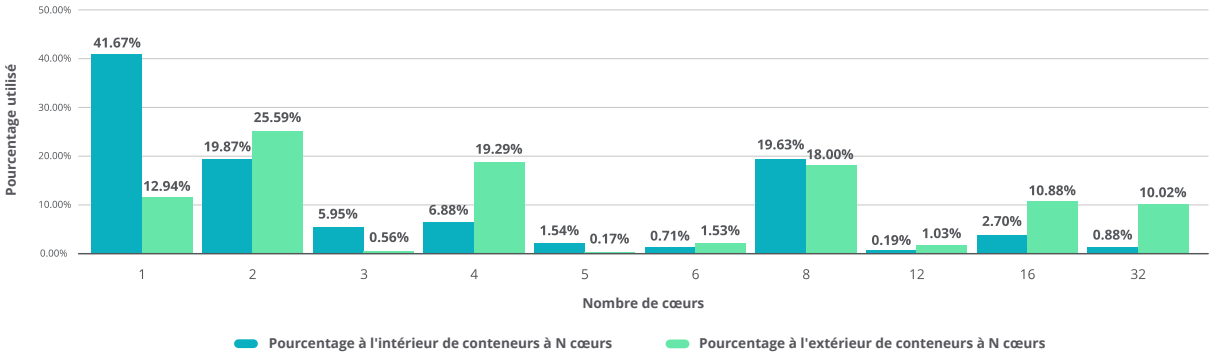
Les conteneurs au cœur de tout ce qui nous entoure

Les applications en conteneurs sont de plus en plus populaires et les données d'applications Java de New Relic en témoignent. Plus de 70 % des applications Java communiquant avec New Relic le font depuis un conteneur.

Paramètres de calculs intensifs dans des conteneurs

Les conteneurs impactent la façon dont les ressources de calculs intensifs et de mémoire sont attribuées. Ainsi, les données New Relic montrent un pourcentage largement supérieur d'applications exécutées avec moins de quatre cœurs lorsqu'elles sont dans des conteneurs.

Pourcentage des applications exécutées à l'intérieur et à l'extérieur de conteneurs par nombre de cœurs

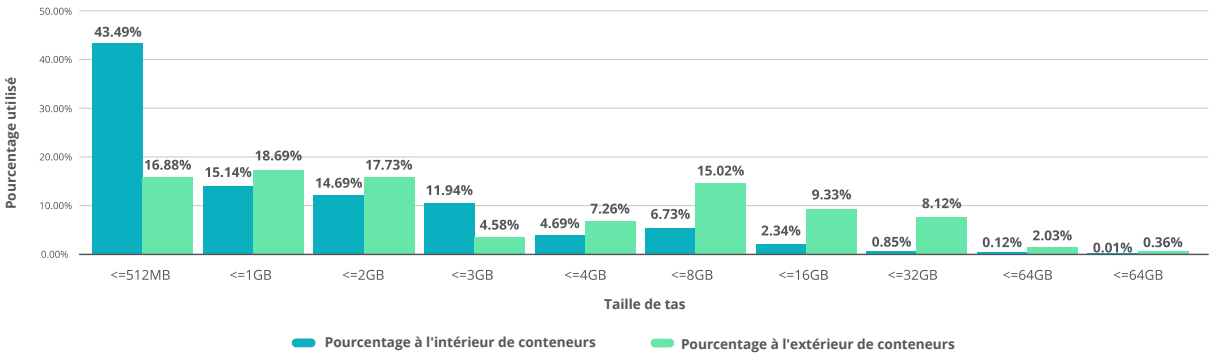


Il est tout à fait logique de rechercher une exécution à plus petite échelle dans des environnements cloud où des conteneurs sont souvent déployés. Mais cette tendance peut s'accompagner de problèmes inattendus pour certaines applications. En particulier, la plupart des avantages concomitants du ramasse-miettes G1 par défaut sur des machines virtuelles Java (JVM) disparaissent lorsque l'exécution se fait sur moins de deux cœurs. Toutes ces instances à cœur unique pourraient tout aussi bien utiliser le ramasse-miettes de série, et en payer le coût en termes de performance, mais beaucoup ne le savent sans doute pas.

Paramètres de mémoire dans des conteneurs

Des tendances similaires apparaissent lorsque l'on compare les paramètres de mémoire, avec un penchant pour des instances plus petites dans des conteneurs. Les données New Relic montrent que 80 % seulement des applications en conteneurs demandent explicitement une limite supérieure sur la mémoire JVM via les indicateurs -Xmx ou -XX:MaxRAMPercentage. Les fonctionnalités de détection de conteneurs dans le JVM, depuis la version 9, ont sans doute mis fin à ce risque de sécurité pour les applications tant que le JVM est le seul processus exécuté dans chaque conteneur.

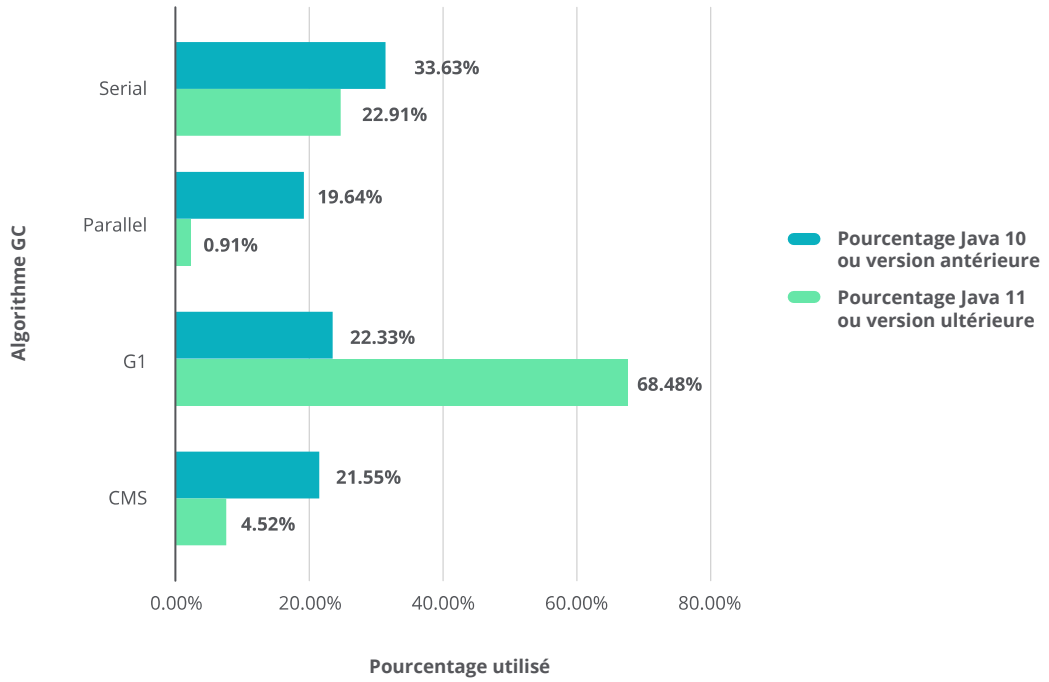
Pourcentage d'allocation de la mémoire selon la taille des tas exécutés à l'intérieur ou à l'extérieur de conteneurs



À données inexactes, résultats erronés

Étant donné son rôle essentiel dans la performance JVM, la récupération de la mémoire du système (Garbage collection, ou GC, en anglais) reste très discutée dans la communauté. Les données New Relic montrent des changements marqués dans l'utilisation du ramasse-miettes après Java 8. Cela n'est pas surprenant compte tenu des valeurs par défaut mises à jour et des performances accrues du ramasse-miettes G1 sur Java 11 et les versions ultérieures.

Pourcentage d'algorithmes GC utilisés pour Java 8 ou versions antérieures par rapport à Java 11 ou versions ultérieures



G1 est le grand favori pour ceux qui ont abandonné Java 8. D'autres ramasse-miettes expérimentaux apparus après Java 8 (ZGC et Shenandoah) restent peu utilisés dans les systèmes de production, mais cela n'est pas surprenant puisque ceux-ci n'ont atteint un statut adapté à la production que récemment.

Méthodologie

Les données sur lesquelles s'appuie ce rapport ont été extraites entièrement d'applications communiquant avec New Relic en janvier 2022 et ne fournissent pas une vue d'ensemble de l'utilisation de Java. New Relic a anonymisé et rendu intentionnellement grossières les données appropriées afin de donner une idée générale de l'écosystème Java. Toute information détaillée susceptible de pouvoir aider des pirates et d'autres personnes malintentionnées a été délibérément exclue du rapport.

À propos de New Relic

En tant que leader de l'observabilité, New Relic permet aux ingénieurs d'avoir une approche datadriven de la planification, du développement, du déploiement et de l'exécution d'excellents logiciels. La plateforme d'observabilité New Relic One propose la seule plateforme unifiée et uniformisée qui permette aux ingénieurs d'obtenir toutes les données télémétriques (métriques, événements, logs et traces) avec les outils d'analyse full-stack les plus puissants qui soient, ce qui les aide à découvrir non seulement les problèmes, mais surtout leurs raisons. Le modèle de tarification intuitif et prévisible de New Relic est le seul du secteur à être basé sur l'utilisation, ce qui permet aux ingénieurs d'obtenir plus pour leur argent et d'améliorer la planification des différents cycles, de réduire les taux d'échecs dus aux modifications, d'accélérer la fréquence de sortie des nouvelles versions et de réduire les temps moyens de résolution des problèmes. Tous ces avantages aident les meilleures marques au monde, dont AB InBev, Banco Internacional, Chegg, Gojek, Signify Health, TopGolf, World Fuel Services (WFS) et Zalora à améliorer les temps de disponibilité et la fiabilité, à stimuler l'efficacité opérationnelle et à assurer une expérience client exceptionnelle, ce qui encourage l'innovation et la croissance.

Commencez à monitorer vos données Java dès aujourd'hui.
Installez le quickstart Java.