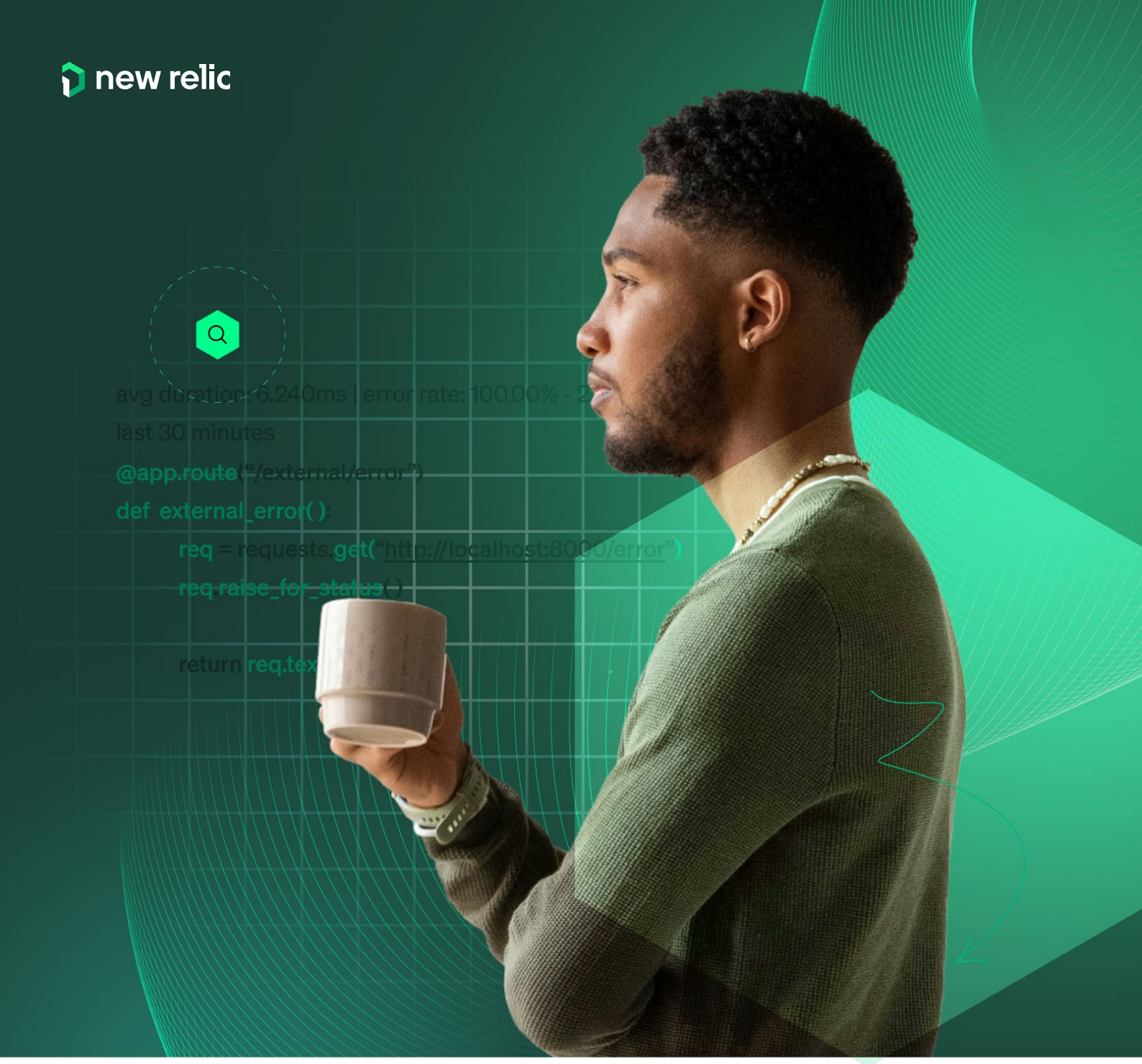
new relic



État des lieux 2024 de l'écosystème Java

Une perspective approfondie des langages de programmation les plus populaires

Rapport / Avril 2024

Table des matières

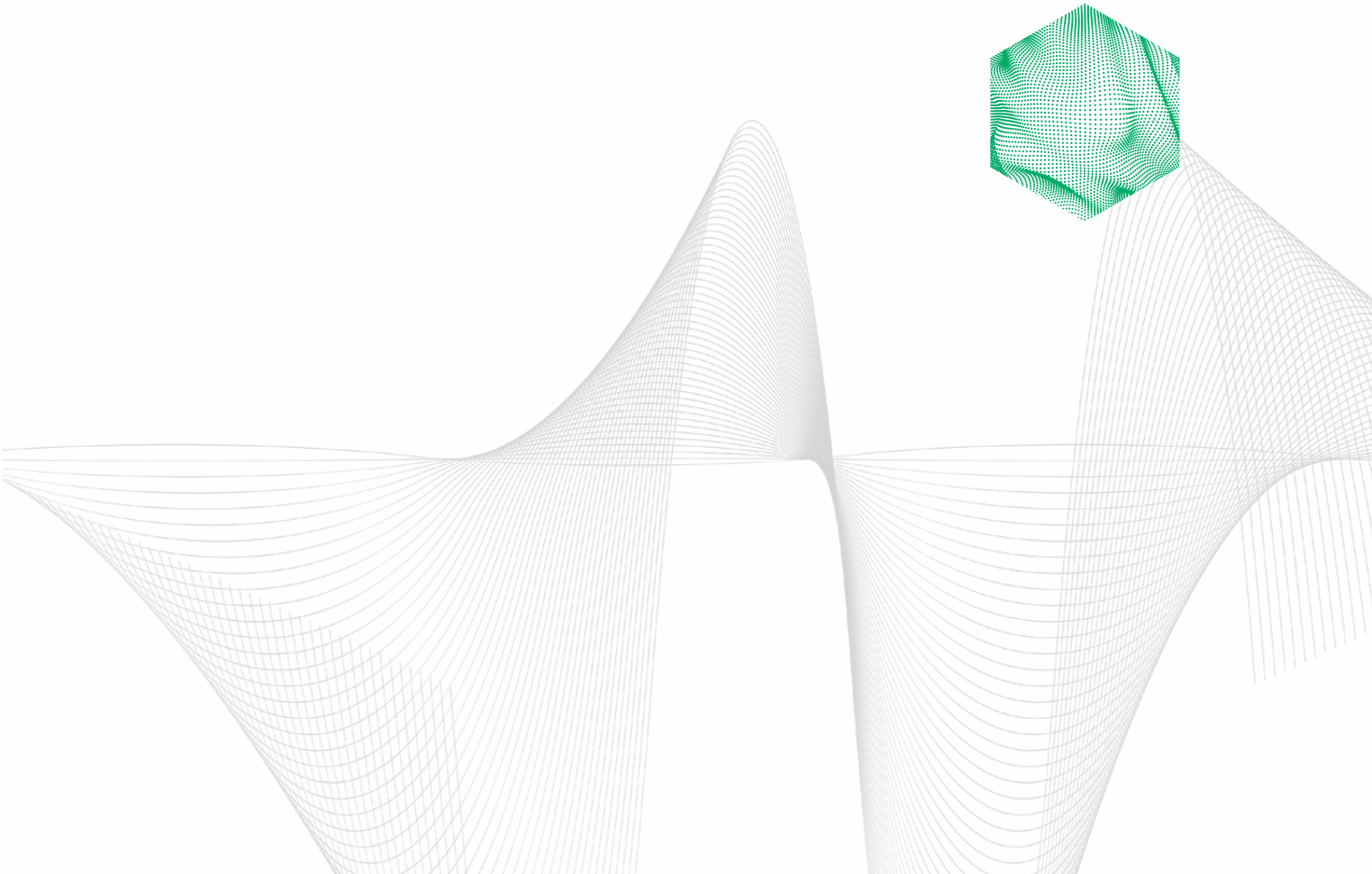
Présentation

New Relic analyse l'écosystème Java depuis plusieurs années pour découvrir les changements au niveau de l'utilisation qu'en font les développeurs, notamment : la façon dont ils adoptent Java 21 à un rythme plus rapide (ou pas) que les autres versions de Java, les tendances du kit pour les développeurs Java (JDK) soutenu par les fournisseurs et la communauté, et les types de questions liées à Java que posent les développeurs qui utilisent l'assistant d'IA générative pour l'observabilité de New Relic.

Ce rapport annuel fournit le contexte et les détails de l'état des lieux actuel de l'écosystème Java, qui sont basés sur les données provenant de centaines de milliers d'applications transmises à New Relic chaque mois.

Le rapport 2024 examine les éléments suivants :

- La plupart des versions de Java les plus utilisées en production
- Les fournisseurs de JDK les plus populaires
- L'utilisation du calcul et de la mémoire dans les applications Java
- Les frameworks et bibliothèques Java les plus populaires pour le logging, le chiffrement et les bases de données
- Les types de questions et de demandes liées à Java les plus courants chez les développeurs



Accélération de l'adoption des nouvelles versions Java

Étudions tout d'abord les versions les plus utilisées en production.

Autrefois, Oracle ne produisait que des nouvelles versions contenant des mises à jour et changements majeurs dans le JDK. Aujourd'hui, les nouvelles versions Java d'Oracle sont publiées tous les six mois — en mars et septembre généralement — et chaque version contient quelques nouvelles fonctionnalités et correctifs. Tous les deux ans, Oracle présente une nouvelle version avec support à long terme (LTS) de Java contenant des mises à jour pour aider à améliorer la stabilité, la sécurité et les performances, ce que les développeurs citent souvent comme étant l'une des motivations les plus importantes pour passer à la version supérieure de Java.

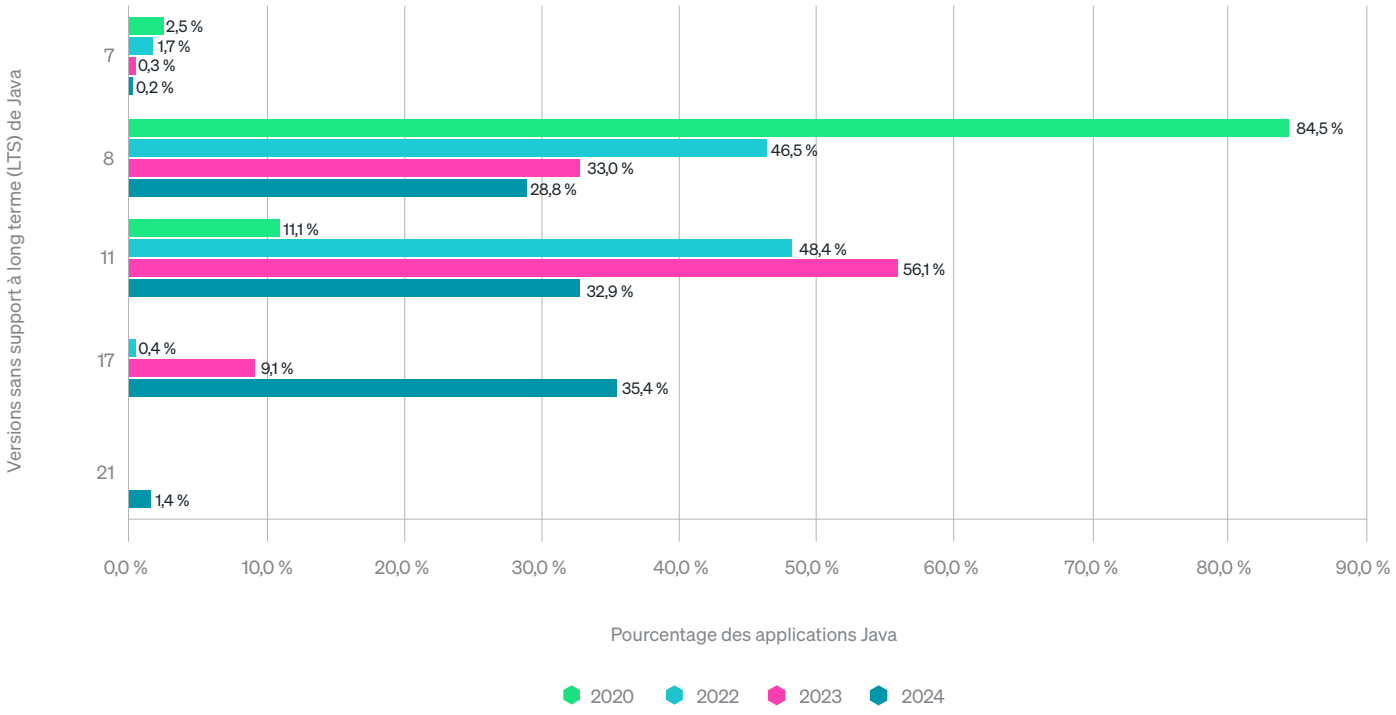
Oracle a publié Java 21 en septembre 2023. Ce jalon est important pour Java et présente des améliorations notables au niveau des fonctionnalités d'aperçu comme les fils (threads) virtuels et les bibliothèques mises à niveau, ainsi que des avancées au niveau de la syntaxe qui placent Java sur un pied d'égalité avec un grand nombre de langages plus modernes.

Au cours des six mois suivant la sortie de Java 21, 1,4 % des applications monitorées par New Relic l'utilisait. Pour placer ce phénomène dans son contexte, il suffit de dire qu'au cours des six mois suivant la sortie de Java 17, seulement 0,37 % des applications l'utilisaient. Une différence de 287 % de moins pour Java 17 alors que son taux d'adoption dépassait déjà de loin ce que le monde des développeurs avait vu lors de la sortie de Java 11. Environ un dixième (9 %) des applications utilisaient Java 17 en production en 2023. Ce chiffre est passé aujourd'hui à 35 %, ce qui représente un taux de croissance de presque 300 % en un an. Pour comparaison, il a fallu des années avant que Java 11 se rapproche de ce niveau.

35 %

des applications utilisent Java 17

Moins de 2 % des applications utilisaient les versions sans LTS de Java, ce qui est logique puisqu'elles ne sont généralement pas utilisées en production.



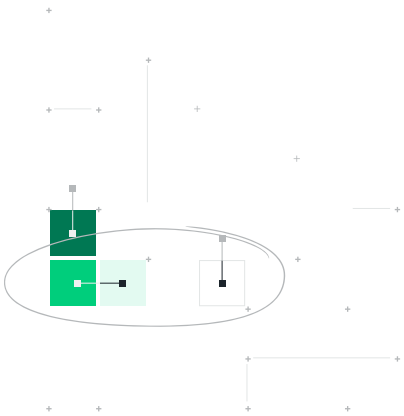
Adoption annuelle des versions de Java avec support à long terme (LTS)

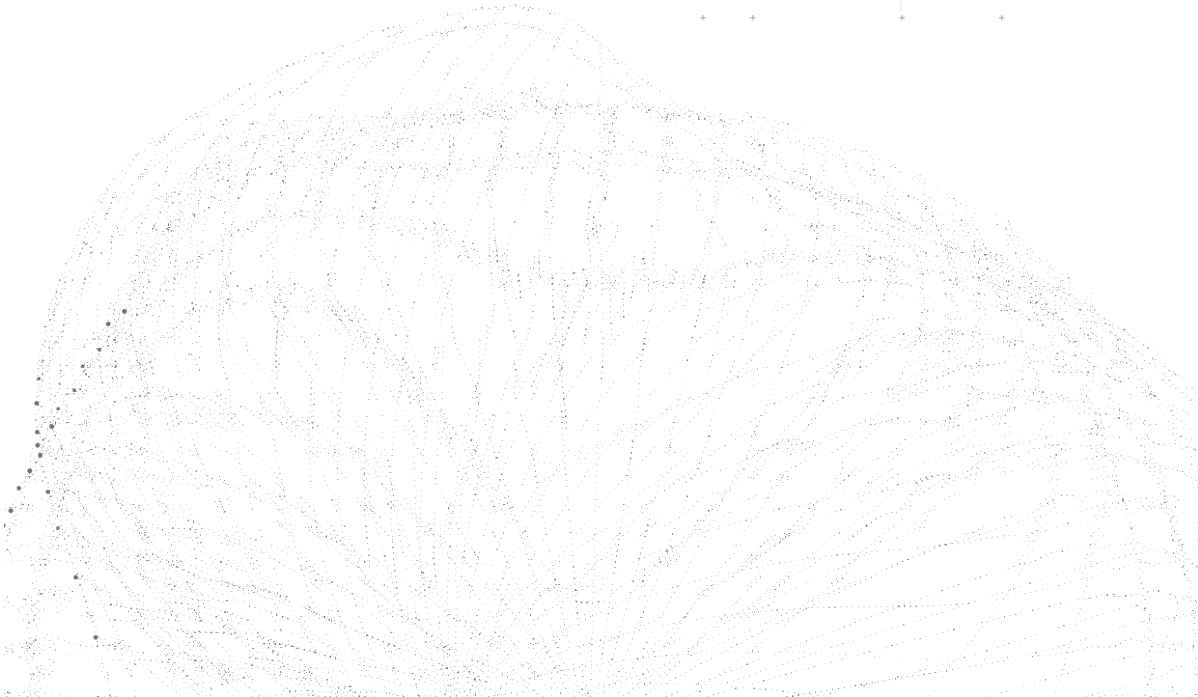
La popularité croissante d'Eclipse Adoptium chez les fournisseurs du JDK

Examinons maintenant les fournisseurs de JDK les plus populaires.

Autrefois, Java était une « source fermée » ou exclusive, ce qui signifiait que les développeurs ne pouvaient télécharger le JDK qu'auprès de Sun Microsystems. Mais environ 10 ans après la sortie de la première version de Java par Sun Microsystems, Oracle a publié la première version de Java en « source ouverte » (ou open source). OpenJDK la maintient et permet aux communautés de développeurs et fournisseurs comme Microsoft et Amazon de maintenir d'autres versions du JDK.

En 2020, Oracle était le fournisseur de JDK le plus populaire avec environ 75 % du marché de Java. Toutefois, depuis le mouvement d'éloignement notable des fichiers binaires d'Oracle en raison de la licence plus restrictive du JDK 11, et malgré un retour à une attitude plus ouverte avec Java 17, nous constatons un déclin régulier d'année en année. Et si Oracle retenait la première place en 2022 avec 34 %, la société est passée à 29 % en 2023 et elle se situe aujourd'hui à 21 %, soit une baisse de 28 % en un an.



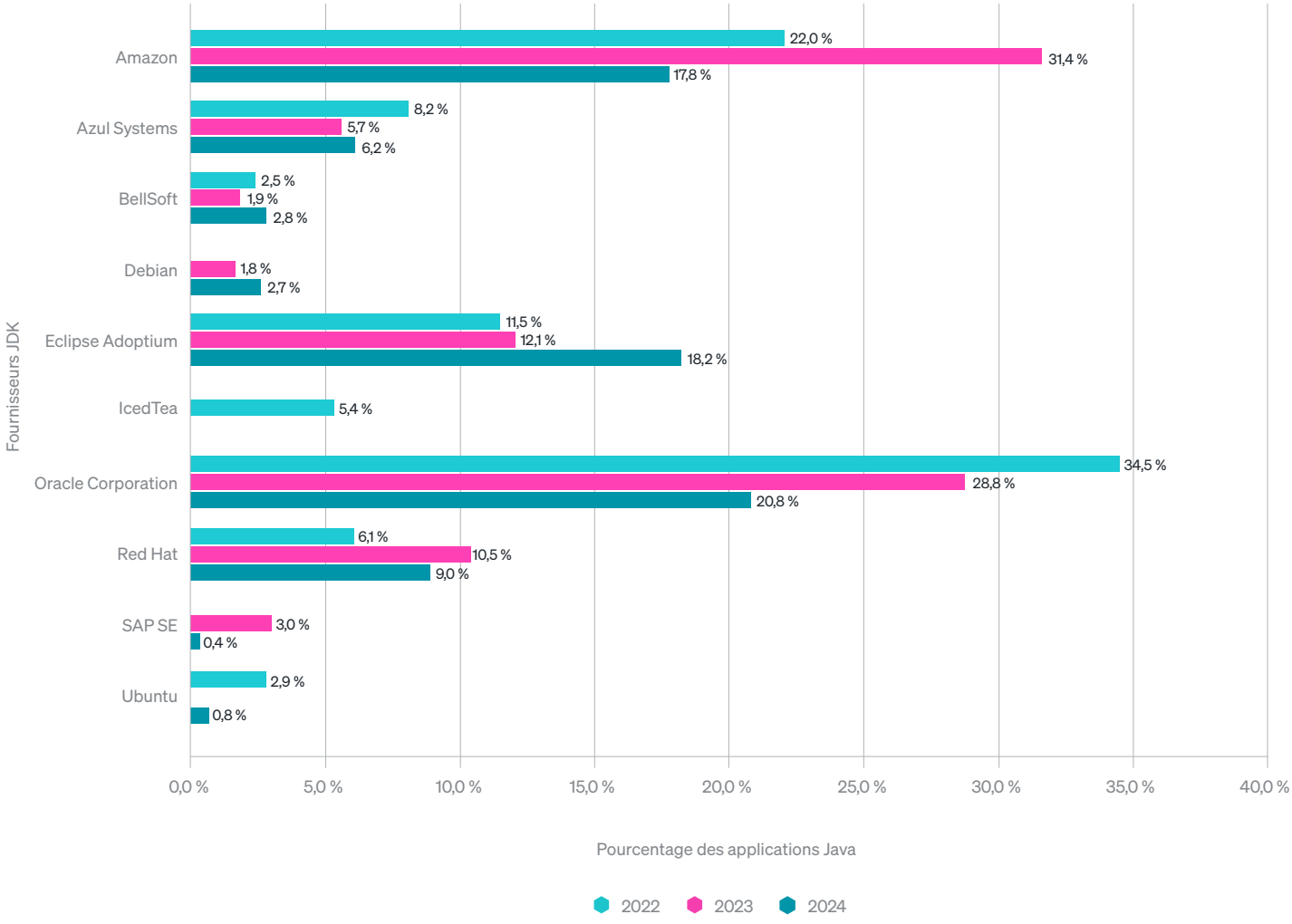


L'utilisation d'Amazon est passée à 31 % de parts de marché en 2023 (alors qu'elle était à 2,2 % en 2020 et à 22 % en 2022), toutefois, elle a subi une baisse de 43 % d'une année sur l'autre en passant à 18 % en 2024.

L'étoile montante cette année est Eclipse Adoptium, dont l'adoption a augmenté de 50 % d'une année sur l'autre en passant de 12 à 18 %. Étant donné qu'Eclipse Adoptium est géré par la communauté, ce JDK tend à être mis à jour plus fréquemment que les JDK d'Oracle et d'Amazon.

18 %

utilisent le JDK d'Eclipse Adoptium



Fournisseurs de JDK les plus populaires par an

Les configurations les plus courantes pour les applications Java

Nous avons ensuite examiné l'utilisation des Garbage Collectors (ou ramasse-miettes), du calcul et de la mémoire dans les applications Java.

43 %

utilisent le Garbage Collector G1

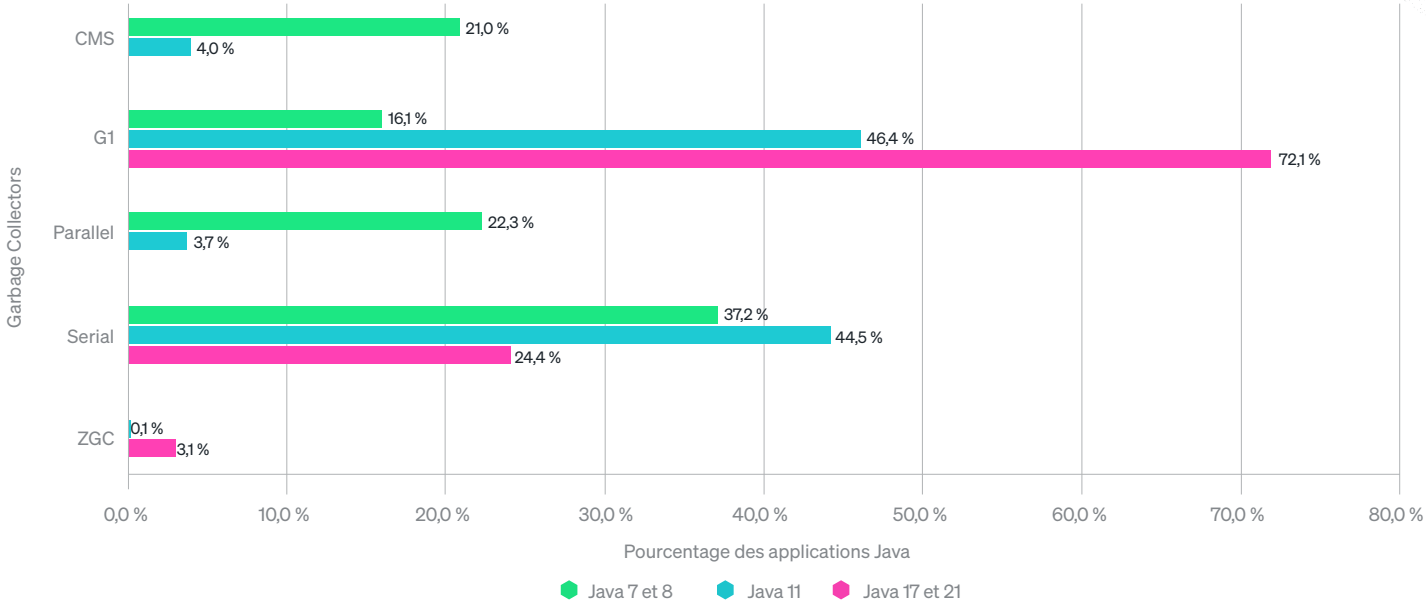
À données inexactes, résultats erronés

Les Garbage Collectors (GC) de Java sont des composants de gestion de la mémoire utilisés pour empêcher les fuites de mémoire, optimiser l'utilisation de la mémoire et assurer les performances globales et la stabilité des applications Java.

Certaines versions de Java ne prennent en charge que quelques GC spécifiques, et, en conséquence, les types de GC utilisés par les développeurs dépendent partiellement de leur version de Java.

Depuis Java 11, le Garbage-First (G1) a été le GC par défaut. Le fait que G1 était le ramasse-miette par défaut explique peut-être pourquoi 43 % des clients l'utilisent et pourquoi il y a eu une grand ruée vers l'adoption de Java 11, 17 et 21 par rapport aux versions Java 7 et 8. En outre, l'un des principaux avantages du G1 est qu'il libère de plus petites zones au lieu de libérer beaucoup d'espace à la fois, ce qui optimise le processus de collecte. Il arrête aussi rarement l'exécution et peut collecter les générations *Young* (jeune) et *Old* (vieille) en même temps, ce qui en fait une excellente option par défaut pour les développeurs.

Le deuxième GC le plus populaire est Serial (37 %). Il est idéal pour les applications ou les systèmes qui tournent sur un seul processeur ou lorsqu'un grand nombre de machines virtuelles Java (JVM) tourne sur le même appareil. Par rapport aux GC plus complexes, il utilise aussi moins de ressources en CPU et en mémoire, ce qui le rend compatible avec les environnements ayant des ressources limitées.



Garbage Collectors utilisés par la version LTS de Java

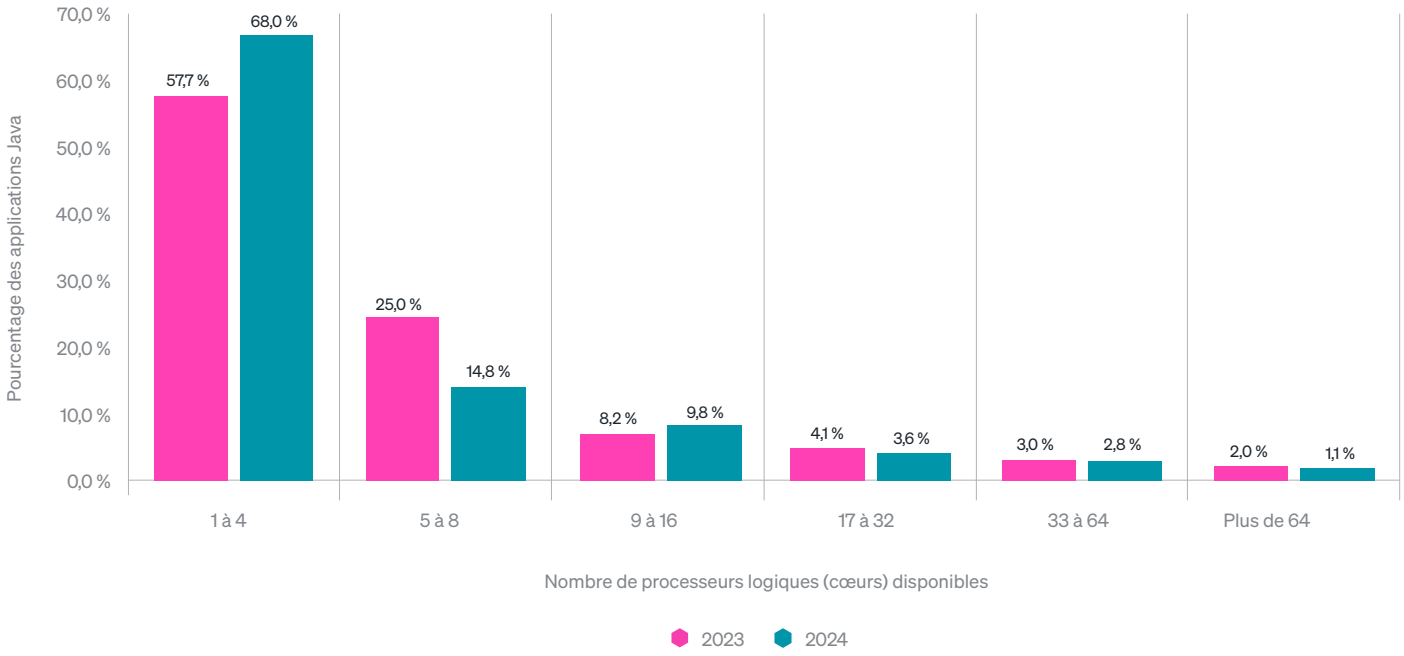
Paramètres de calcul et de mémoire

68 %

utilisent 1 à 4 cœurs

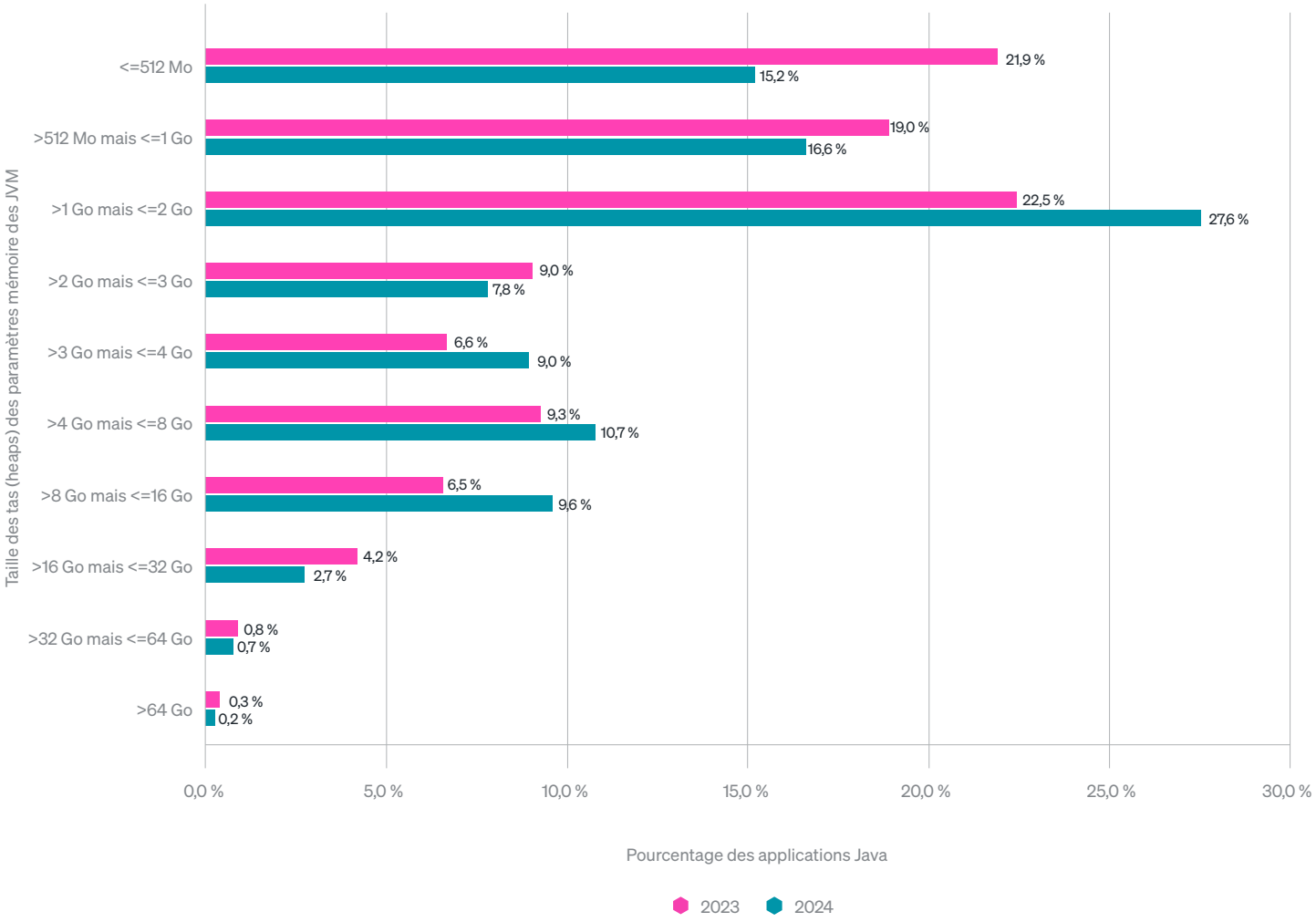
Les données de New Relic montrent une augmentation de 18 % d'une année sur l'autre dans les applications exécutées avec quatre cœurs ou moins.

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 Garbage Collector G1 par défaut sur les JVM récentes disparaissent lorsque l'exécution se fait sur moins de deux cœurs. Toutes ces instances à cœur unique pourraient tout aussi bien utiliser le GC Serial, et payer le coût des performances de cette option.



Nombre de processeurs logiques (cœurs) disponibles par application Java en 2023 et 2024

Si l'on regarde les paramètres mémoire de la JVM, 32 % des applications Java utilisent 1 Go ou moins et 68 % utilisent plus de 1 Go. Cela équivaut à une augmentation de 15 % d'une année sur l'autre pour les applications utilisant plus de 1 Go de mémoire.



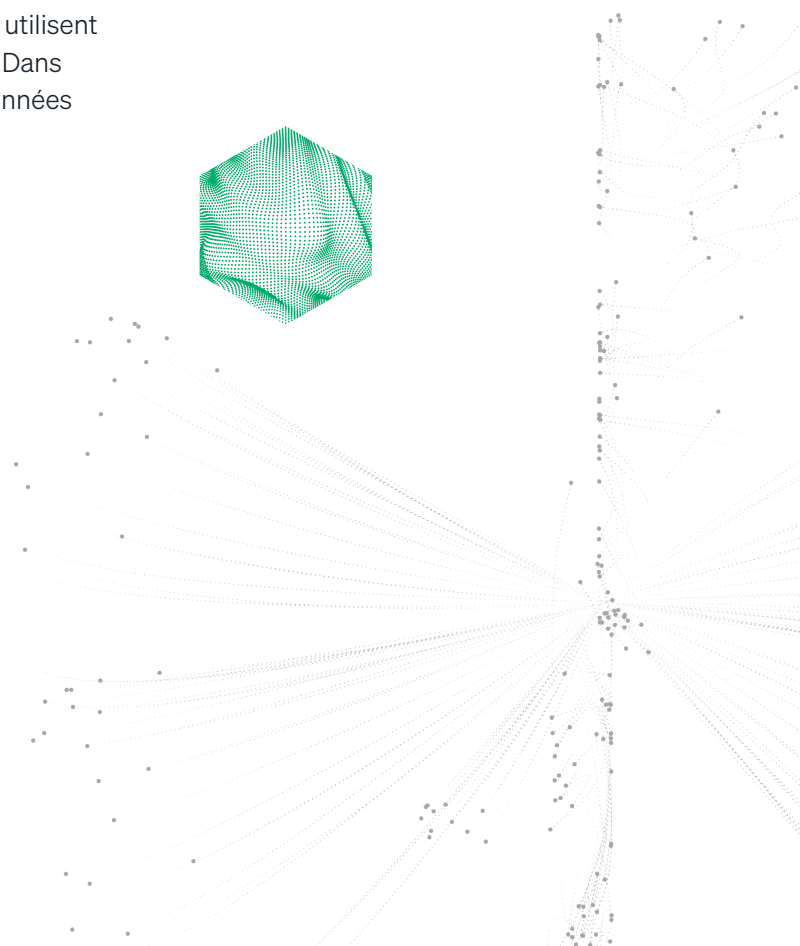
Taille des tas (heaps) des paramètres mémoire des JVM par an

Les frameworks et bibliothèques Java populaires pour le logging, le chiffrement et les bases de données

Tous les développeurs Java utilisent des frameworks et des bibliothèques pour rationaliser le processus de développement d'applications en utilisant le code pré-écrit pour les tâches courantes comme le développement web et la connectivité des bases de données. Les bibliothèques sont utilisées pour améliorer la fonctionnalité d'une application. Nous examinons ici les frameworks et bibliothèques les plus populaires pour le logging, le chiffrement et la connectivité des bases de données.

Log4j est le framework de logging le plus populaire pour les applications Java

Toute application ou système logiciel peut comporter des bogues et des problèmes dans des environnements de test ou de production et les développeurs utilisent des outils de logging pour résoudre les problèmes. Toutefois, le logging est seulement utile lorsqu'il fournit les informations requises à partir des messages de log sans impacter négativement les performances de l'application. Les développeurs de logiciels utilisent divers frameworks de logging pour résoudre ces problèmes. Dans les faits, 91 % des applications Java qui transmettent leurs données à New Relic utilisent les frameworks de logging.

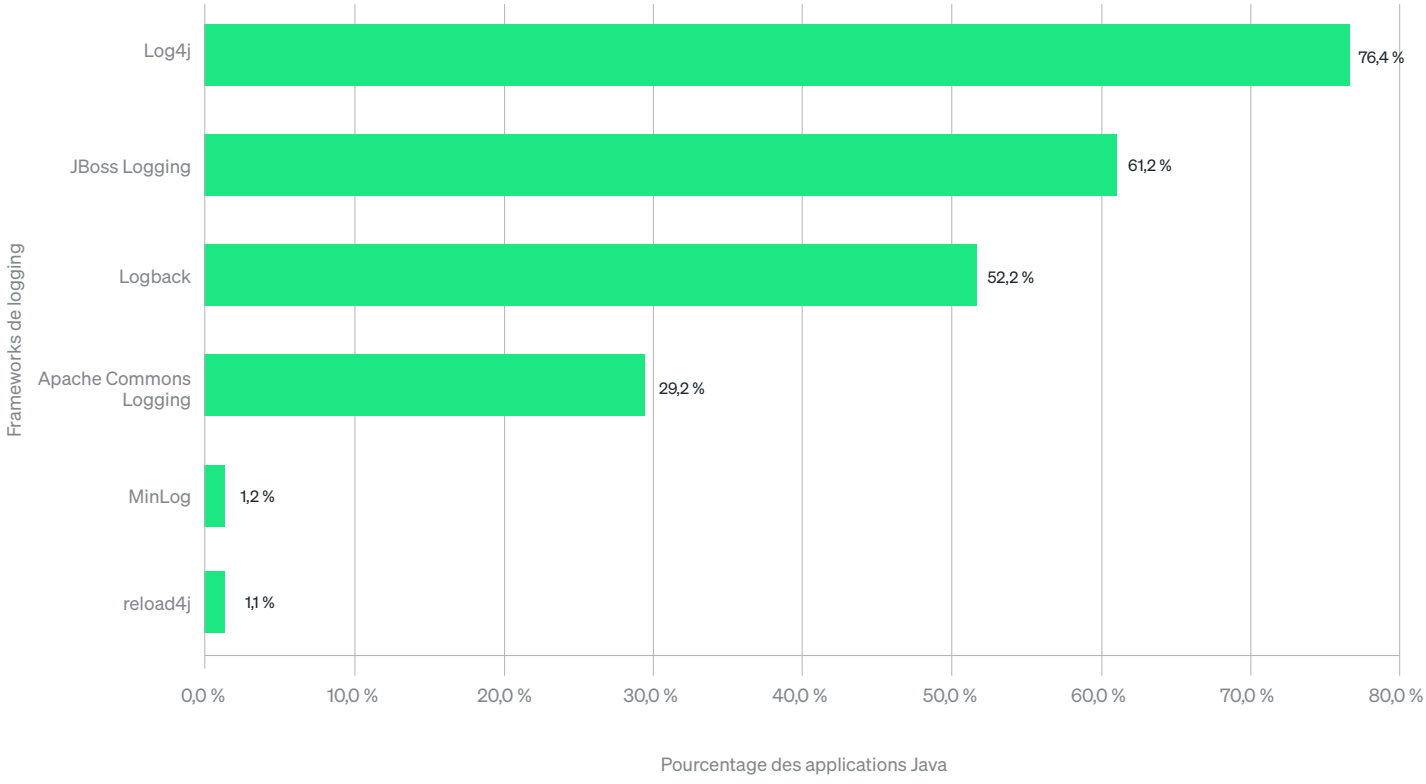


Le framework de logging le plus utilisé est Log4j avec 76 % d'applications Java qui s'en servent, suivi par JBoss Logging (61 %) et Logback (52 %).

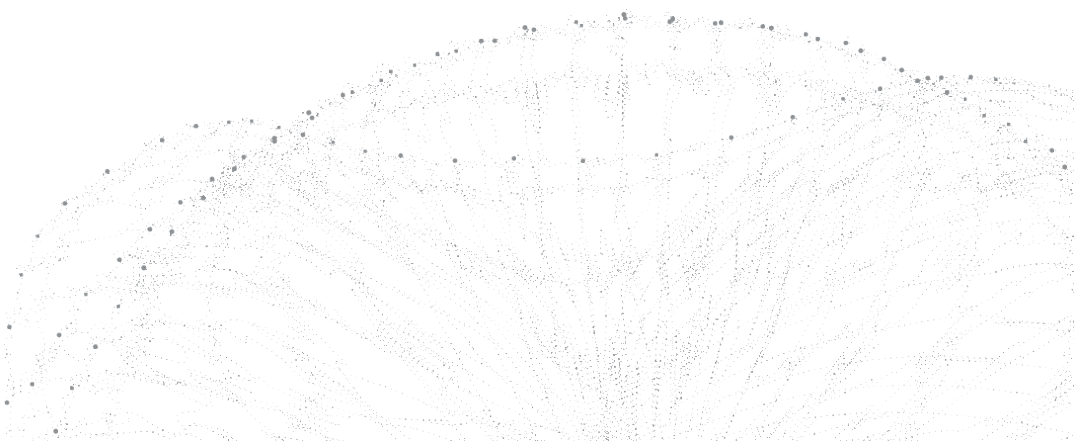
En outre, la plupart des développeurs Java (83 %) comptent sur SLF4j, qui est un framework agissant en tant qu'abstraction pour les autres types de frameworks de logging Java. SLF4j permet aux développeurs d'utiliser le framework de logging de leur choix et aux applications de passer à n'importe quel framework de logging Java indifféremment sans impacter les implémentations ni apporter de modifications. En raison de cette fonctionnalité, SLF4j permet aux applications d'être indépendantes, et donne ainsi une plus grande flexibilité et plus de portabilité pour le logging sur toutes les parties du système. Cela signifie également que les applications Java peuvent utiliser plus d'un framework de logging.

76 %

utilisent le framework de logging Log4j



Framework de logging le plus populaire pour les applications Java



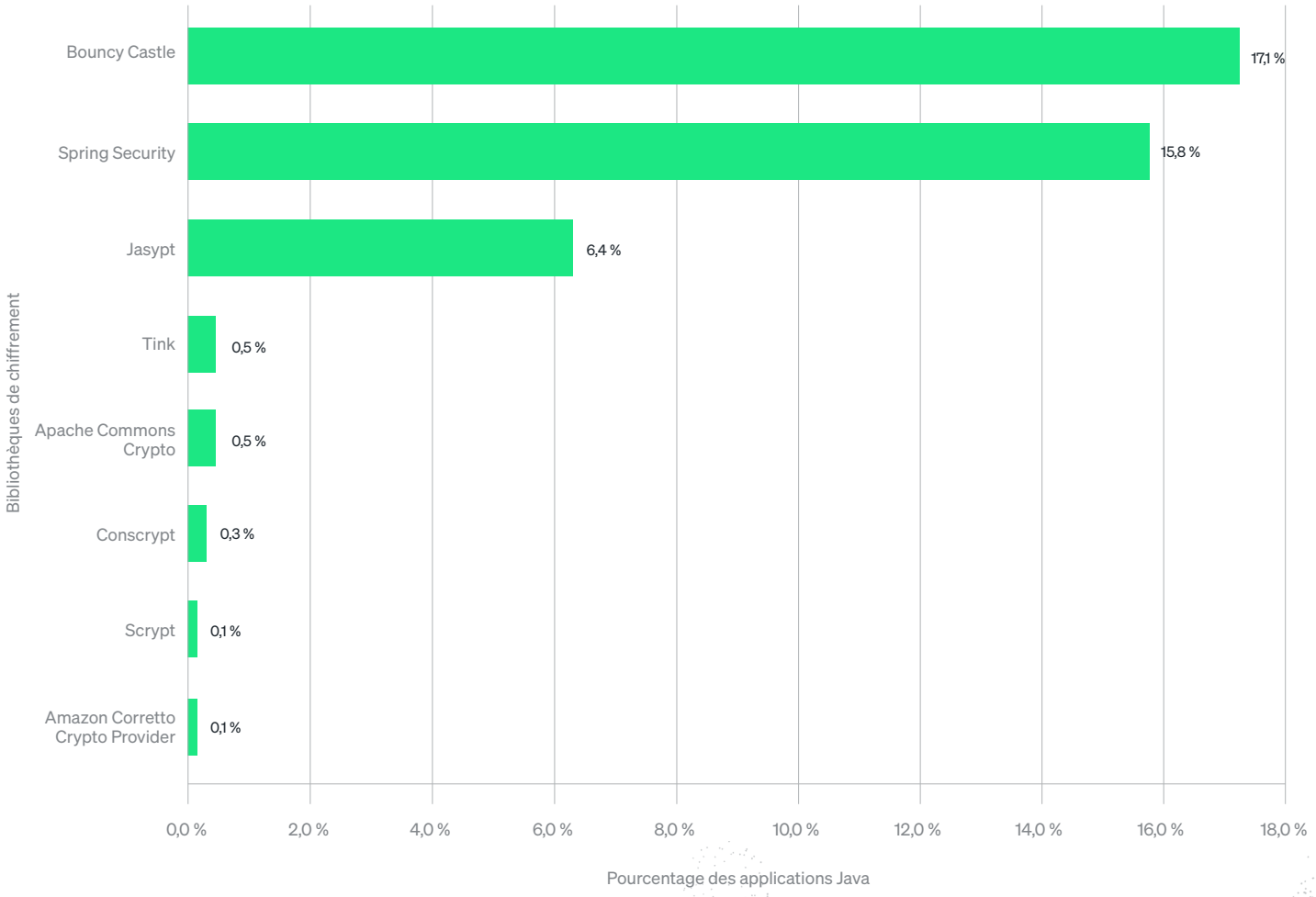
Bouncy Castle est la bibliothèque de chiffrement la plus populaire pour les applications Java

17 %

utilisent la bibliothèque de chiffrement Bouncy Castle

Plus d'un tiers (41 %) des applications Java envoyant les données vers New Relic utilisent des bibliothèques de chiffrement : 17 % utilisent Bouncy Castle, 16 % utilisent Spring Security et 6 % utilisent Jasypt. Bouncy Castle est depuis de nombreuses années une bibliothèque populaire pour le chiffrement en raison de ses suites et utilitaires de cryptographie.

Bien que seulement 0,09 % des développeurs utilisent la bibliothèque ACCP (Amazon Corretto Crypto Provider), nous prévoyons que plus d'applications l'utiliseront dans un futur proche, car les entreprises et les développeurs veulent consolider les fournisseurs et parce qu'elle fournit généralement de meilleures performances.



Bibliothèques de chiffrement les plus populaires pour les applications Java

Oracle est le système de base de données le plus populaire pour les applications Java

17 %

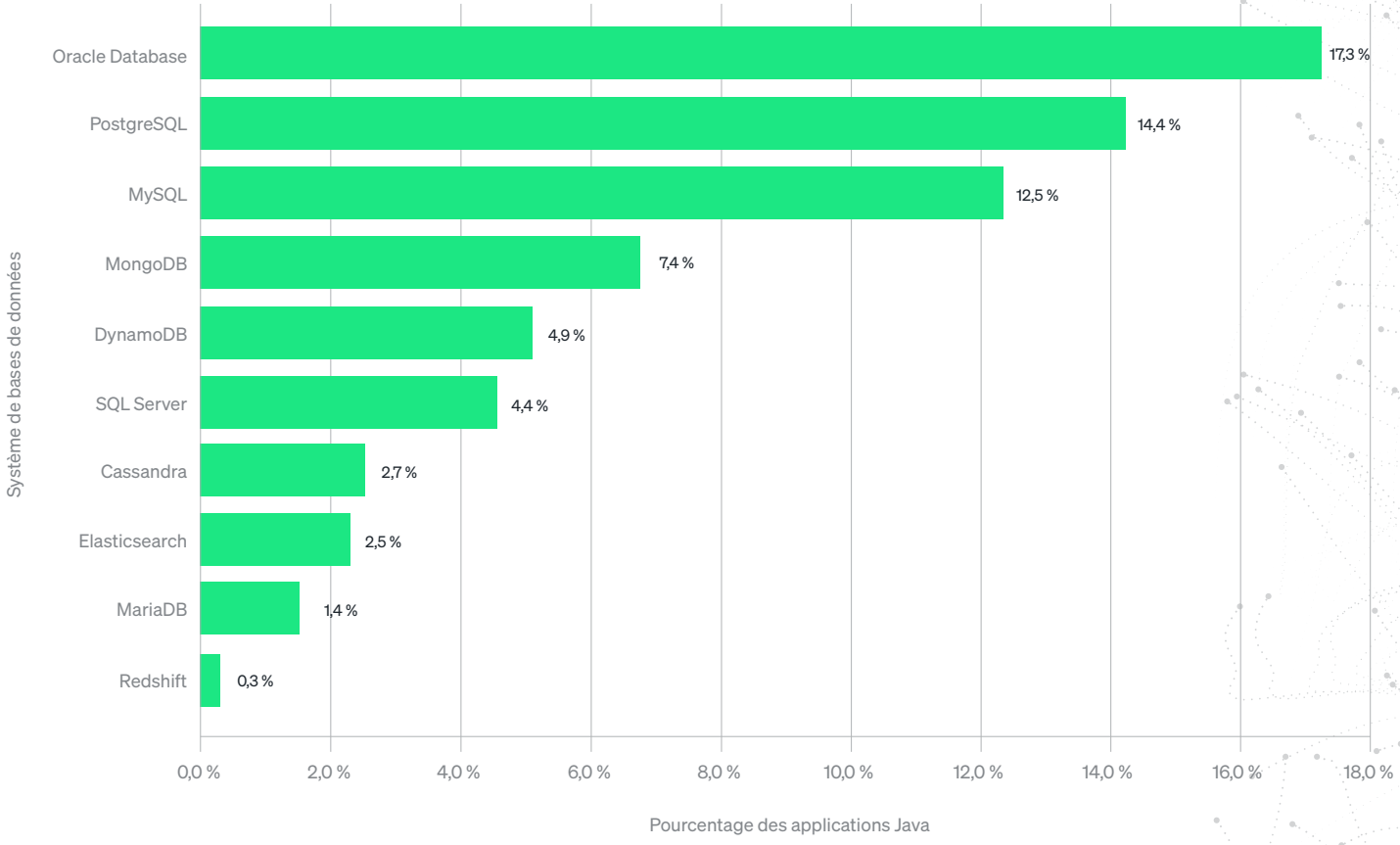
utilisent Oracle Database

En ce qui concerne les bases de données, Oracle Database est la plus largement utilisée, avec 17 % des applications Java transmettant leurs données à New Relic qui l'utilisent. Oracle Database est connue pour son évolutivité (ou scalabilité) et sa capacité à gérer de larges quantités de données rapidement et efficacement. En tant que telle, elle a tendance à être le système de base de données préféré des entreprises. En outre, elle offre une assistance client et un ensemble d'outils robustes.

Le deuxième système de base de données le plus populaire est PostgreSQL, avec 14 % des applications Java transmettant leurs données à New Relic qui l'utilisent. Bien qu'Oracle Database soit géré directement par Oracle et disponible par le biais d'une licence, PostgreSQL est une base de données open source en utilisation libre qui est préférée pour la gestion des opérations et des requêtes complexes en lecture seule.

MySQL est troisième, avec 13 % des applications Java qui l'utilisent. Il s'agit également d'une base de données open source. Elle offre moins de fonctionnalités qu'Oracle Database et PostgreSQL, ce qui la rend plus stable et plus rapide lors du traitement, surtout s'il s'agit du traitement de requêtes en lecture seule.

PostgresSQL a gagné en popularité au cours de la dernière année, car il se conforme davantage aux spécifications SQL et prend en charge plus de fonctionnalités que MySQL. MySQL demeure un excellent système de base de données, mais nous pouvons voir que sa popularité diminue alors que celle de PostgresSQL augmente.



Systèmes de base de données les plus populaires pour les applications Java

Les types de questions et de demandes liées à Java les plus courants chez les développeurs

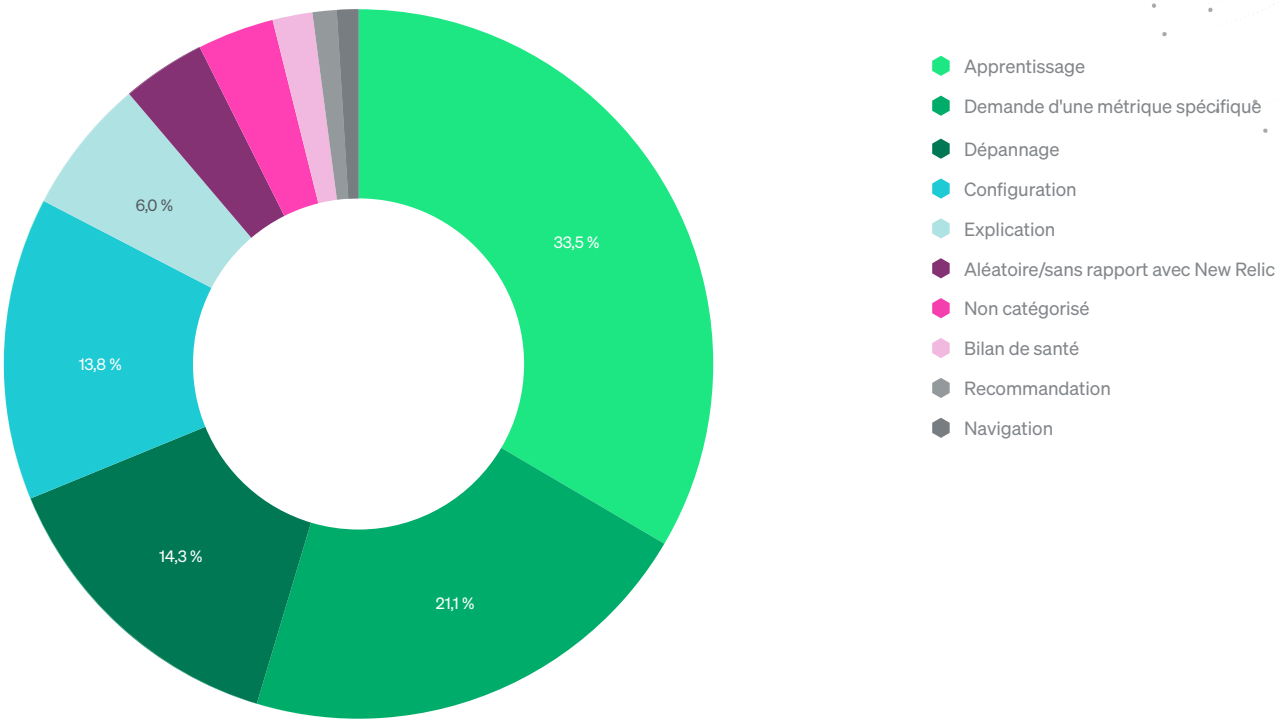
34 %

des questions et demandes liées à Java portent sur comment faire quelque chose

Nous avons également examiné le type de questions et de demandes liées à Java provenant des développeurs par le biais de l'assistant d'IA générative pour l'observabilité New Relic AI. Depuis Janvier 2024, les données montrent que 34 % des 483 questions liées à Java étaient des questions d'apprentissage (Comment...?), 21 % étaient relatives à l'interrogation d'une métrique spécifique, 14 % concernaient la configuration et un autre 14 % le dépannage.

Voici quelques exemples de questions et demandes :

- Comment me connecter aux appels d'API dans Java Spring Boot ?
- Si le framework Java est passé de Tomcat à JBoss, dois-je reconfigurer l'agent ?
- Création d'un dashboard pour l'utilisation de la mémoire du conteneur et l'utilisation du heap de la JVM.
- Cette application monitorée utilise le framework Vaadin dans Java. Elle consomme une grande quantité de mémoire. Qu'est-ce que ça peut être ?
- Dans JVM, que signifie « G1 Eden Space heap usage » ?



Types de questions et de demandes liées à Java les plus courants provenant des développeurs qui utilisent New Relic AI

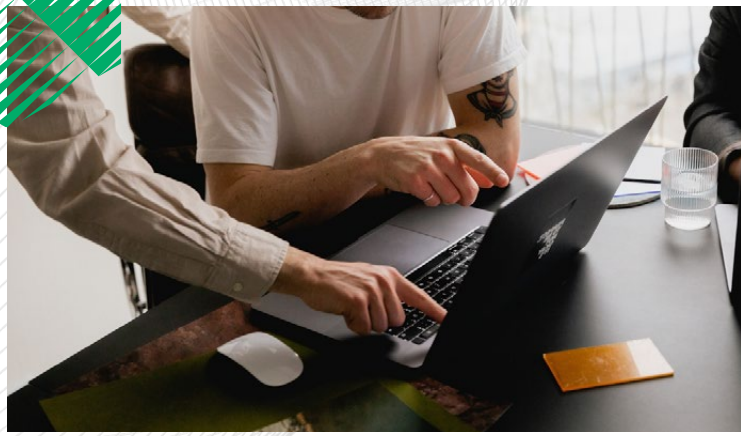
Méthodologie

Ce rapport est basé sur les données rassemblées à partir de centaines de milliers d'applications et envoyées à New Relic, qui apportent des informations sur les performances. Il ne fournit donc pas un portrait complet de l'utilisation de Java. Toutes les données ont été collectées en 2024.

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 d'aider des pirates et d'autres personnes malintentionnées a été délibérément exclue du rapport.

Démarrez le monitoring de vos données Java avec New Relic dès aujourd'hui !

Installer le quickstart Java



À propos de New Relic

En tant que leader de l'observabilité, New Relic permet aux ingénieurs d'avoir une approche data-driven de la planification, du développement, du déploiement et de l'exécution d'excellents logiciels. New Relic propose la seule plateforme unifiée et uniformisée avec toutes les données télémétriques (métriques, événements, logs et traces) associées à de puissants outils d'analyse full-stack pour aider les ingénieurs à donner le meilleur d'eux-mêmes en s'appuyant sur des données et non sur des opinions.

Le modèle de tarification intuitif et prévisible de New Relic est le premier du secteur à être basé sur l'utilisation, ce qui permet aux ingénieurs d'obtenir plus de leur investissement en les aidant à améliorer la planification des différents cycles, diminuer les taux d'échecs des modifications, optimiser la fréquence de publication des nouvelles versions et réduire les temps moyens de résolution des problèmes (MTTR). Ceci permet aux meilleures marques du monde et aux start-up en hyper-croissance d'améliorer les temps de disponibilité, la fiabilité et l'efficacité opérationnelle et de donner à leurs clients une expérience exceptionnelle qui stimule l'innovation et la croissance.

new relic.