

Pourquoi le tracing distribué est-il essentiel pour l'APM ?

Réduisez le MTTR dans des environnements d'application distribués complexes

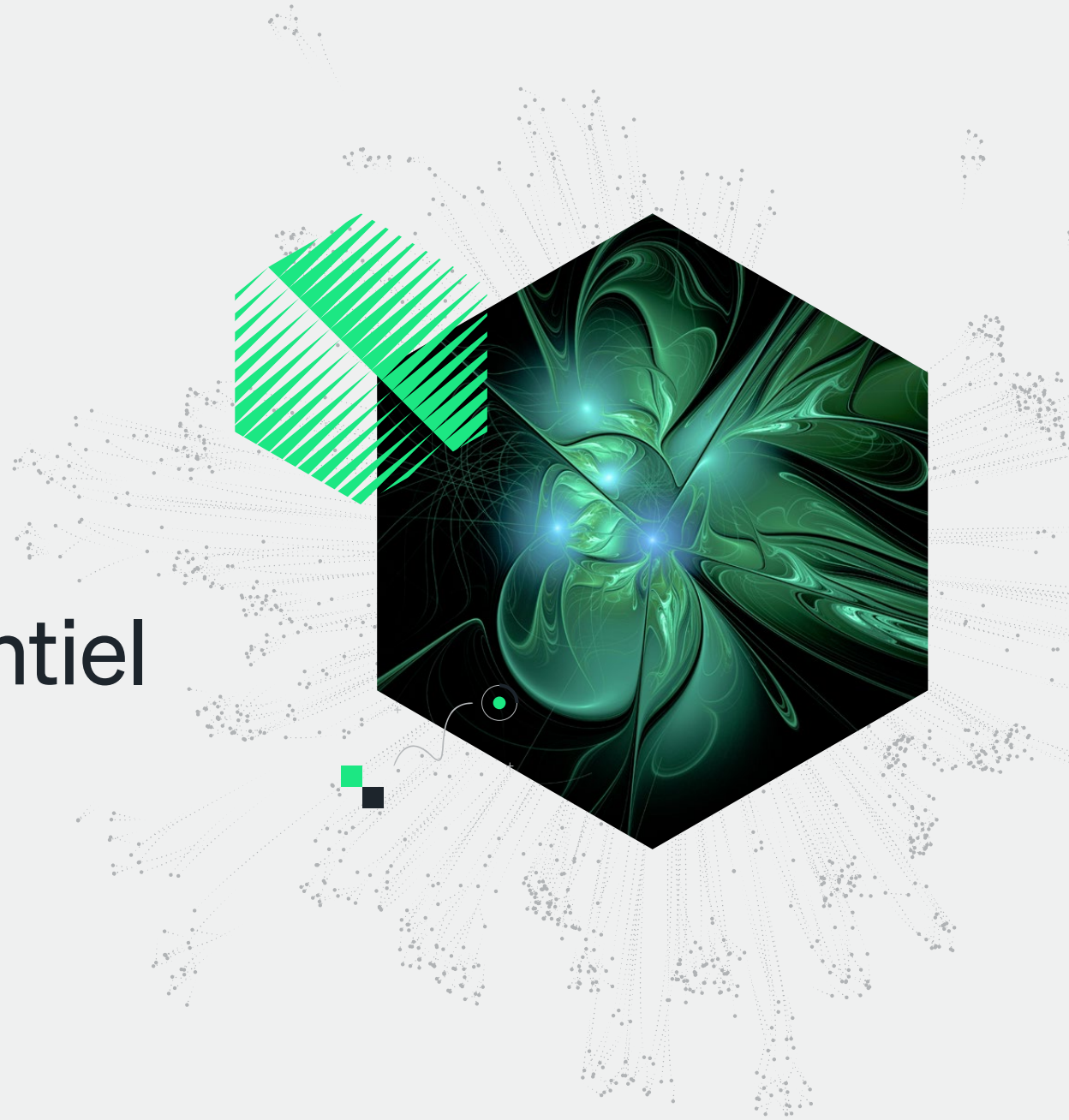


Table des matières



03 Élagage de la complexité

04 Traçage du chemin passant dans les systèmes distribués

- 06 › Quand utiliser les traces ?
- 07 › Comment fonctionnent les traces ?
- 08 › Savoir tirer les conclusions
- 09 › Pourquoi les organisations ont-elles besoin du tracing distribué ?

10 Plus grande visibilité sur le pipeline de données

- 10 › Échantillonnage efficace en début de workflow
- 11 › Traces exploitables avec échantillonnage en fin de workflow
- 13 › Analyse et visualisation

14 Allègement du fardeau de gestion

15 L'échantillonnage au début ou à la fin ? Les deux !

16 Étapes suivantes

17 À propos de New Relic

Élagage de la complexité

Les environnements et les architectures logiciels modernes, comme les microservices, ont le potentiel d'accélérer le développement d'applications. Mais dans de nombreuses organisations, les équipes IT sont confrontées à un environnement complexe, qui rend difficile le diagnostic et la résolution des problèmes de performance et des erreurs avant qu'ils n'impactent la fiabilité et l'expérience des clients.

Les environnements de microservices peuvent inclure quelques dizaines à plusieurs centaines de services, rendant ainsi difficile la tâche de déterminer les chemins de requêtes et de diagnostiquer les problèmes. Et le fardeau du monitoring des performances des applications (APM) ne fait qu'augmenter avec l'orchestration, l'automatisation, et les CI/CD pour les déploiements logiciels fréquents. Sans une instrumentation adéquate du monitoring, les organisations risquent d'imposer à leurs équipes des recherches répétées pour trouver des réponses dans des systèmes distribués, ce qui augmente le temps moyen de résolution (MTTR) et réduit le temps de développement de logiciels innovants.

L'observabilité élague la complexité des logiciels et fournit une visibilité de bout en bout qui permet aux équipes de résoudre les problèmes plus rapidement, de travailler plus intelligemment et de créer de meilleures expériences numériques pour leurs clients. L'observabilité crée le contexte et des informations exploitables en combinant notamment quatre types essentiels de données d'observabilité : métriques, événements, logs et traces (MELT).

Les traces — ou plus précisément les traces distribuées — sont essentielles pour les équipes logicielles qui ont réalisé une transition (ou qui l'envisagent) vers le cloud et qui ont adopté des architectures de microservices. C'est parce que le tracing distribué est la meilleure façon de comprendre rapidement ce qui arrive aux requêtes à mesure qu'elles passent par les microservices qui composent les applications distribuées.

Les cadres, ingénieurs DevOps, propriétaires de produits, SRE (ingénieurs de fiabilité du site), chefs d'équipe logiciels, ou d'autres parties prenantes peuvent utiliser le tracing distribué pour trouver les goulots d'étranglement ou les erreurs et gagner du terrain pour résoudre les problèmes plus rapidement.



Traçage du chemin passant dans les systèmes distribués

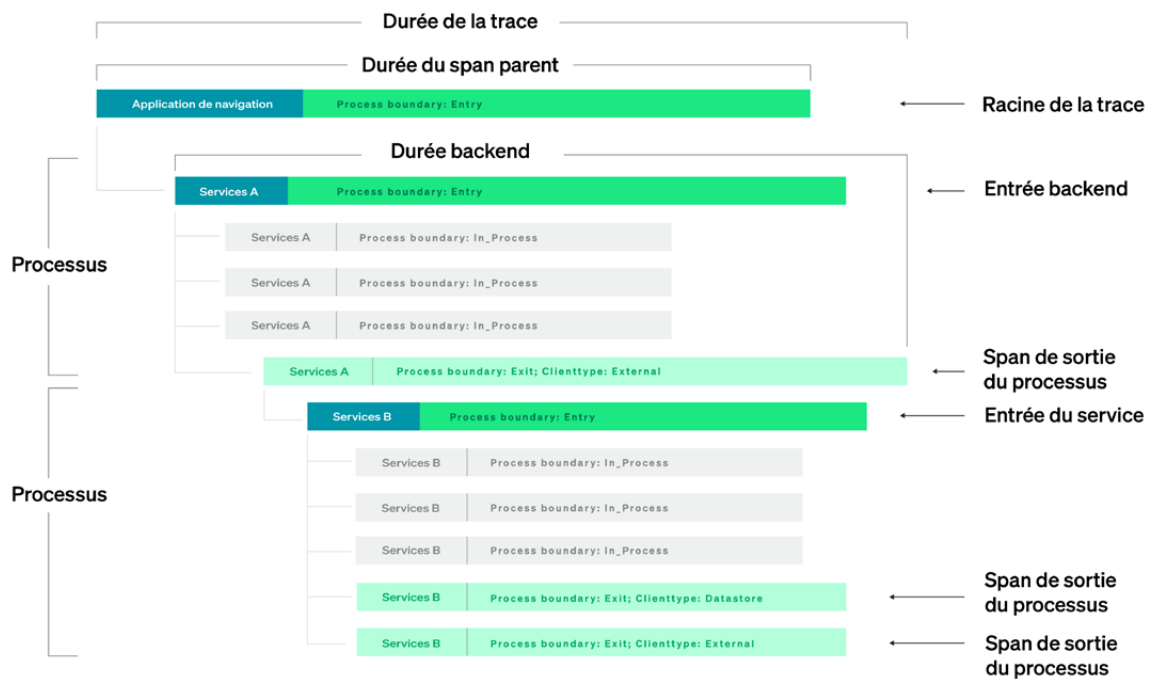
Le tracing distribué est désormais monnaie courante pour l'exploitation et le monitoring des environnements d'application modernes. Lorsque les équipes font le monitoring des logiciels et des performances système pour l'observabilité, le tracing permet de monitorer et d'analyser les requêtes au fur et à mesure qu'elles se propagent via un environnement distribué et passe d'un service à l'autre.

Le tracing distribué est la capacité à tracer une solution pour suivre et observer les demandes de service à mesure qu'elles passent dans les systèmes distribués en collectant les données pendant que les requêtes d'un service à l'autre. Les données de trace aide les équipes à comprendre le flux de requêtes via l'environnement de microservices et à trouver les endroits où les pannes ou les problèmes de performances se sont produits sur le système, et pourquoi.

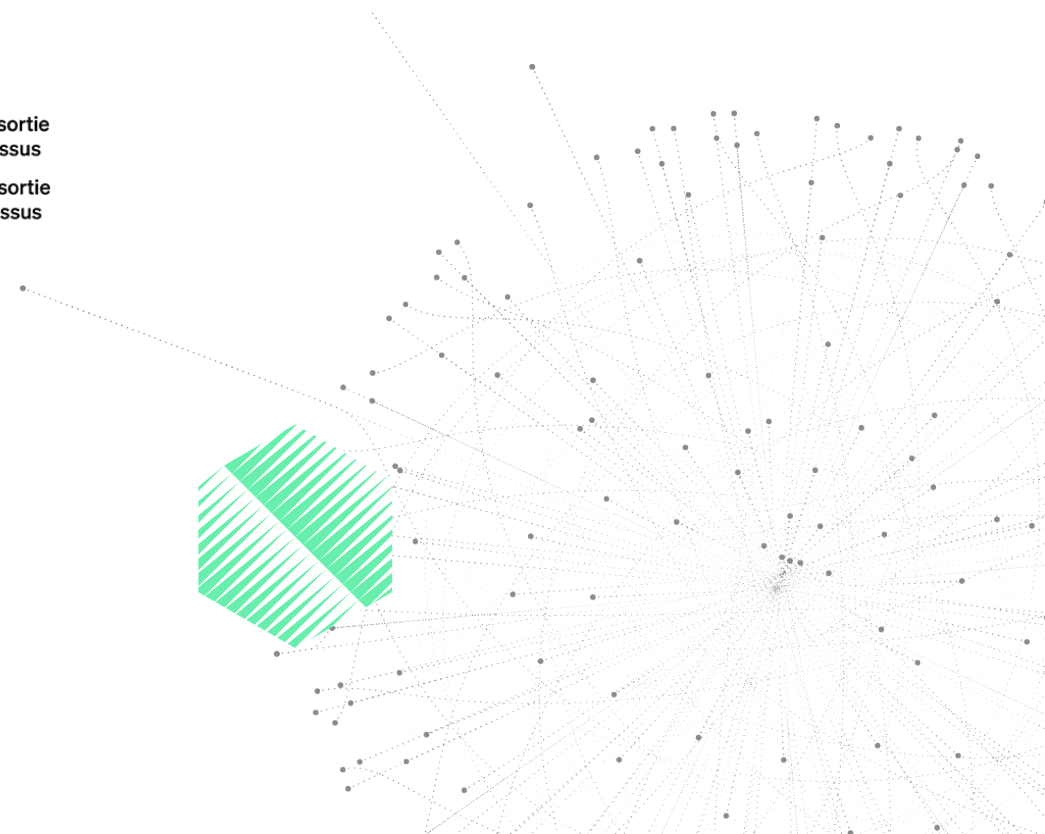
Lorsque les équipes instrumentent les systèmes pour le tracing distribué, toutes les transactions génèrent la télémétrie de la trace, de l'utilisateur frontend aux appels de base de données du backend. Par exemple, quand des clients cliquent sur un panier pour un achat sur une application e-commerce, cette requête est envoyée vers plusieurs services frontend et backend sur plusieurs conteneurs, dans des environnements serverless, sur des machines virtuelles, avec différents prestataires cloud, sur site, ou une combinaison quelconque de tout cela. La requête peut comprendre le service d'inventaire pour s'assurer que l'inventaire est disponible, le service de paiement et le service d'expédition, puis finalement, la requête se termine et retourne à l'utilisateur. À chaque fois que la requête passe d'un service à un autre, elle émet un span avec la télémétrie de tracing. Une fois que la requête est terminée, les spans sont assemblés pour créer une trace complète du parcours de la requête dans le système.

Avec le tracing distribué, les équipes peuvent :

- Tracer le parcours d'une requête à travers le système complexe.
- Comprendre les dépendances de service en amont et en aval
- Découvrir la latence des composants sur ce parcours.
- Comprendre où se trouvent les goulots d'étranglement sur le chemin de la requête.
- Voir et analyser où se produisent les erreurs dans la transaction au niveau du service individuel.



Les diagrammes de dispersion et en cascade indiquent le temps pris par chaque requête à chaque étape sur les différents services d'application



Quand utiliser les traces ?

En général, le tracing distribué est la meilleure façon pour les équipes DevOps, Ops, IT et SRE d'obtenir rapidement des réponses à des questions spécifiques dans des environnements où les logiciels sont distribués ou dépendent d'architectures serverless. Dès qu'une requête implique quelques microservices, il est essentiel de pouvoir visualiser comment les différents services fonctionnent ensemble.

Les données de trace fournissent le contexte de ce qui se passe sur l'ensemble de l'application et entre les services et les entités. S'il n'y avait que des événements bruts pour chaque service isolément, il n'y aurait aucun moyen de reconstruire une chaîne unique entre les services d'une transaction particulière.

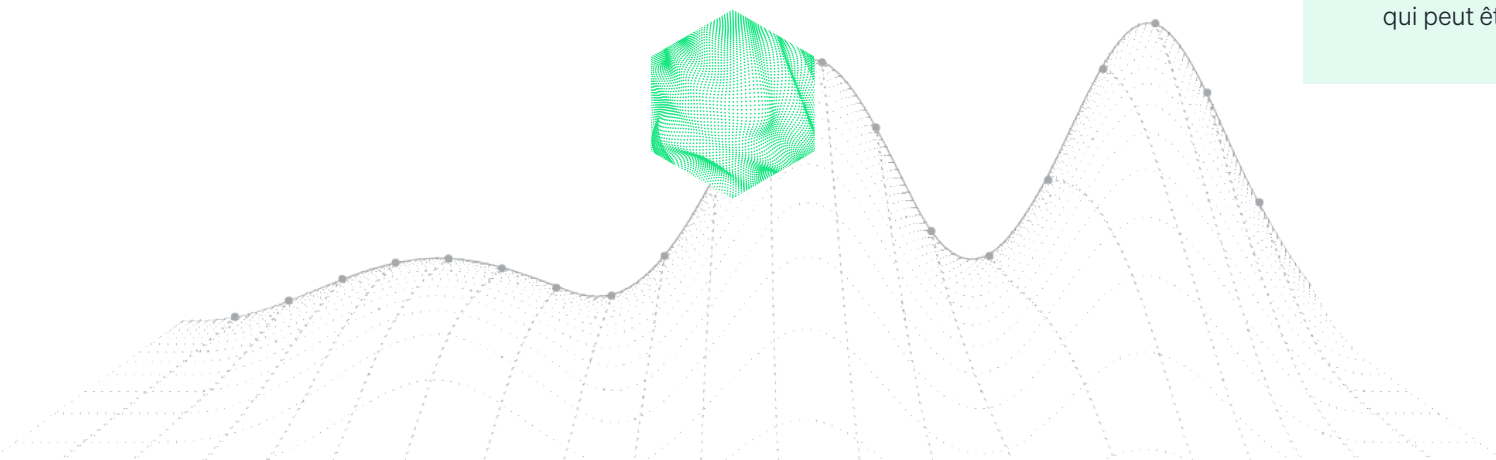
Les applications appellent souvent de nombreuses autres applications en fonction de la tâche qu'elles essaient d'accomplir et elles traitent aussi souvent les données en parallèle. Pour ces raisons, la chaîne d'appels peut être inconciliable et le timing incertain pour la corrélation. La seule façon d'avoir une chaîne d'appels convergents consiste à faire passer le contexte de la trace entre chaque service afin d'identifier exclusivement une seule transaction sur toute la chaîne.

Cela signifie que les équipes doivent utiliser le tracing distribué pour obtenir des réponses à des questions telles que :

- Quel est l'état de santé (ou l'intégrité) des services qui constituent un système distribué ?
- Quelle est la cause profonde des erreurs et des défauts au sein d'un système distribué ?
- Où se trouvent les goulots d'étranglement de performance qui pourraient impacter l'expérience des clients ?
- Quels services contiennent du code problématique ou inefficace que les équipes devraient prioriser pour l'optimisation ?

Petit guide sur la terminologie du tracing distribué :

- Une **transaction** correspond aux fonctions et appels de méthodes qui constituent cette unité de tâche dans une application logicielle. Elle démarre lorsque la méthode est appelée et se termine lorsque celle-ci revient ou rencontre une erreur.
- Une **requête** est la façon dont les applications, microservices et fonctions communiquent ensemble.
- Une **trace** correspond aux données de performance sur les requêtes à mesure qu'elles traversent les microservices.
- Un **span** représente les opérations ou les segments qui font partie d'une trace.
- Un **span parent** est le premier span d'une trace.
- Un **span enfant** est un span ultérieur qui peut être imbriqué.



Comment fonctionnent les traces ?

Lorsqu'elles sont rassemblées, les traces forment des événements spéciaux appelés « spans », qui permettent de faire le suivi d'une chaîne causale dans un écosystème de microservices pour une seule transaction. Pour réaliser les spans, chaque service échange l'un avec l'autre des identifiants de corrélation, appelés « contexte de trace ». Ce contexte de trace est utilisé pour ajouter des attributs au span.

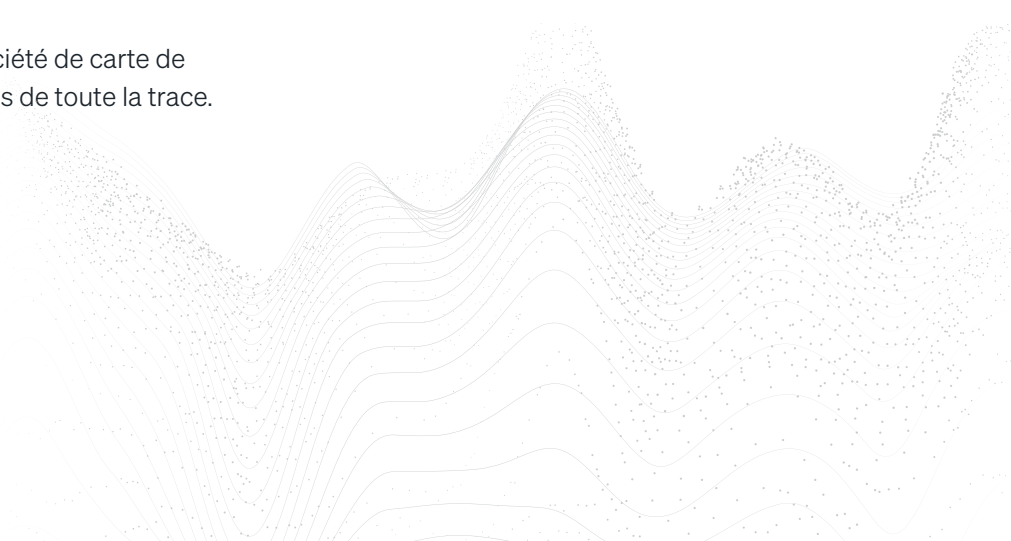
Horodatage	EventType	TraceID	SpanID	ParentID	ServiceID	Durée
8/11/2022 15:34:23	Span	2ec68b32	aaa111	bbb111	Distributeur automatique	23
8/11/2022 15:34:22	Span	2ec68b32	bbb111	aaa111	Backend du distributeur automatique	18
8/11/2022 15:34:20	Span	2ec68b32	ccc111	bbb111	Société de carte de crédit	15
8/11/2022 15:34:19	Span	2ec68b32	ddd111	ccc111	Banque émettrice	3

Exemple d'une trace distribuée composée des spans d'une transaction par carte de crédit

Dans le tableau ci-dessus, l'horodatage et les données de durée indiquent que le service de la société de carte de crédit est le plus lent de la transaction et prend 12 des 23 secondes, soit plus de la moitié du temps de toute la trace.



Comment avons-nous obtenu 12 secondes ? Le span pour contacter la banque émettrice est appelé un span enfant. Le span qui contacte la société de carte de crédit est son parent. Ainsi, si la requête à la banque a pris trois secondes, la société de carte de crédit a pris 15 secondes, et si l'on soustrait l'enfant du parent, il a fallu 12 secondes pour traiter la transaction sur la carte de crédit.



Savoir tirer les conclusions

Au début de leur avancée vers des applications distribuées, les organisations se sont rapidement rendu compte qu'elles avaient besoin de visibilité sur les microservices individuels isolément et sur tout le flux de requêtes. Cette migration est la raison pour laquelle le tracing distribué est devenu une bonne pratique pour obtenir la visibilité nécessaire sur tout ce qui se passe. Et la combinaison des traces avec les trois autres types essentiels de données télémétriques (métriques, événements et logs) donne aux équipes un tableau complet de l'environnement et des performances de leurs logiciels et leur permet une observabilité de bout en bout.

Le tracing distribué exige aussi le contexte de trace. Cette exigence signifie qu'il faut attribuer un identifiant unique à chaque requête, et à chaque étape dans la trace, l'encodage de ces informations contextuelles, et le passage (ou la propagation) du contexte encodé d'un service à l'autre pendant que la requête avance dans l'environnement d'une application. Ce processus permet à l'outil de tracing distribué de corréler chaque étape d'une trace, dans l'ordre correct, ainsi que d'autres informations nécessaires pour monitorer et faire le suivi des performances.

Une seule trace capture généralement les données sur :

- Les spans (nom du service, nom de l'opération, durée et autres métadonnées)
- Les erreurs
- La durée des opérations importantes au sein de chaque service (tel que les appels de méthode et fonctions internes)
- Les attributs personnalisés



W3C Trace Context est devenu la norme de propagation du contexte de trace traversant les limites de processus. Il permet à tous les traceurs et agents conformes à la norme de participer dans une trace, avec les données de trace qui sont propagées du premier au dernier services. De nombreux fournisseurs d'observabilité, dont New Relic, prennent pleinement en charge la norme W3C Trace Context.



Pourquoi les organisations ont-elles besoin du tracing distribué ?

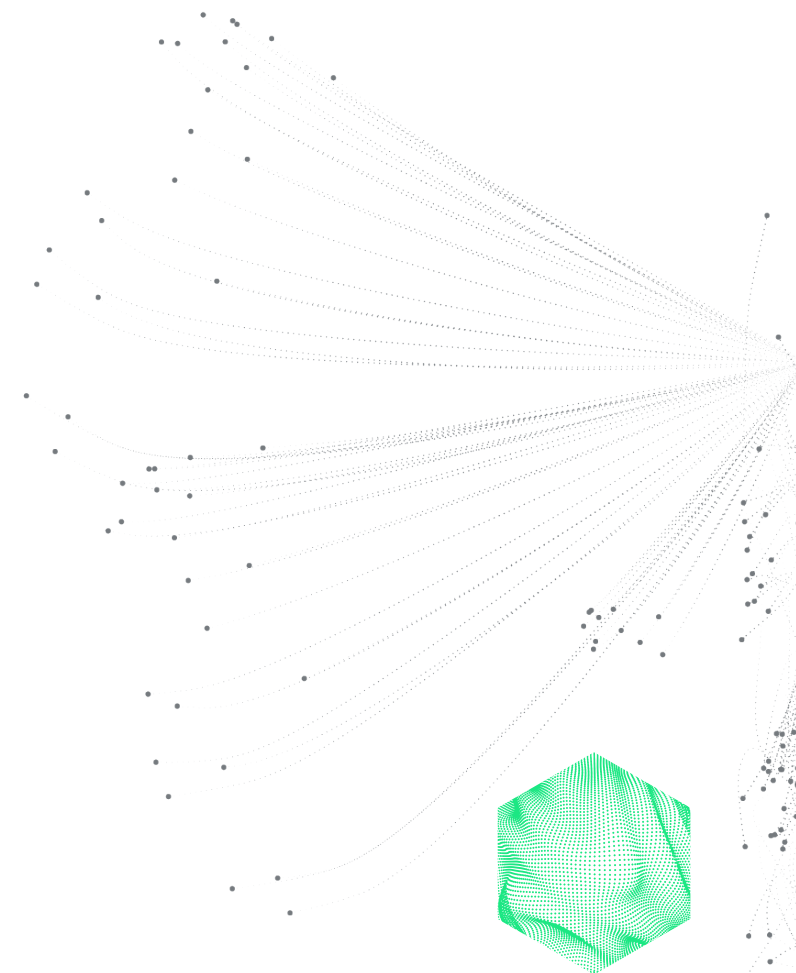
Alors que les nouvelles technologies et pratiques (cloud, microservices, conteneurs, fonctions serverless, DevOps, SRE, etc.) favorisent une plus grande vélocité et permettent un passage plus fluide des phases d'écriture du code à la production des logiciels, elles présentent également de nouveaux défis.

- Davantage de points de défaillance dans le stack d'applications
- Augmentation du MTTR en raison de la complexité de l'environnement de l'application
- Moins de temps d'innovation pour les équipes qui doivent consacrer de longues heures au diagnostic des problèmes

Par exemple, une requête lente peut impacter l'expérience d'un groupe de clients. Cette requête est distribuée sur de multiples microservices et fonctions serverless. Plusieurs équipes possèdent et monitorent les différents services impliqués dans la requête, mais aucune n'a signalé de problèmes de performance avec ses microservices. Sans possibilité de voir les performances de toute la requête sur les différents services, il est quasiment impossible de repérer précisément où et pourquoi la latence est élevée ni de savoir quelle équipe doit prendre le problème en charge. Dans le cadre de la stratégie d'observabilité de bout en bout, le tracing distribué relève les défis des environnements d'applications modernes.

En comprenant en profondeur les performances de chaque service — en amont et en aval — les équipes IT peuvent plus efficacement et plus rapidement :

- Identifier et résoudre les problèmes afin de minimiser l'impact sur l'expérience des clients et les résultats de l'entreprise.
- Mesurer l'état global de santé du système et comprendre l'effet des changements sur l'expérience des clients.
- Prioriser les zones à forte valeur pour les améliorer et ainsi optimiser l'expérience numérique des clients.
- Innover constamment en toute confiance pour surpasser la concurrence.



Plus grande visibilité sur le pipeline de données

Le tracing distribué exige le reporting et le traitement de la télémétrie de tracing. Le volume de données de la trace peut croître de manière exponentielle dans le temps, car le volume de requêtes augmente avec le déploiement par les équipes de nouveaux microservices dans l'environnement.

Pour cette raison, de nombreuses organisations utilisent l'échantillonnage des données pour gérer la complexité et les coûts associés à la transmission de l'activité des traces. Dans l'idéal les données échantillonnées représentent les caractéristiques de l'ensemble complet des données.

Les équipes IT ont besoin d'avoir la flexibilité nécessaire pour choisir un échantillonnage en début ou en fin de workflow afin de répondre aux exigences de monitoring pour chaque application.

Échantillonnage efficace en début de workflow

L'échantillonnage en début de workflow collecte et stocke les données de trace de manière aléatoire pendant que le span parent (le premier span) est traité pour faire le suivi et analyser ce qui se passe au niveau de la transaction sur tous les services qu'elle touche. En règle générale, l'échantillonnage en début de workflow se déroule au sein de l'agent chargé de la collecte de données télémétriques de trace en effectuant la sélection aléatoire des traces à échantillonner pour l'analyse. Les décisions d'échantillonnage se produisent avant que les traces ne soient terminées. Étant donné qu'il n'y a aucun moyen de savoir quelle trace peut rencontrer un problème, il est possible que les équipes ne voient pas certaines traces qui contiennent des processus anormalement lents ou des erreurs.

L'échantillonnage en début de workflow est une bonne option pour obtenir un échantillon statistique global des requêtes qui passent dans un système distribué. Il réussit bien à trouver les traces contenant des erreurs ou subissant une latence dans les applications avec un volume plus faible de transactions et d'environnements et une combinaison d'architectures basées sur un monolithe et des microservices. L'échantillonnage en début de workflow est une méthode efficace pour prélever une large quantité de données de trace en temps réel et il a peu ou aucun impact sur les performances des applications.

Avantages de l'échantillonnage en début de workflow

- Fonctionne bien pour les applications avec un débit de transaction plus faible
- Mise en route rapide et simple
- Convient aux environnements monolithiques et microservices, avec toujours une prédominance des monolithes actuellement
- Peu ou pas d'impact sur les performances des applications
- Une solution peu coûteuse pour l'envoi des données de trace à des fournisseurs tiers
- L'échantillonnage statistique assure une transparence suffisante sur le système distribué

Limites de l'échantillonnage en début de workflow

- Les traces sont échantillonnées de manière aléatoire
- L'échantillonnage se produit avant qu'une trace ait terminé son parcours dans les différents services, et il n'y a donc aucun moyen de savoir à l'avance quelle trace a rencontré un problème.
- Dans les systèmes à haut débit, les traces avec des erreurs ou une latence anormale peuvent ne pas être échantillonnées et donc manquer.

Traces exploitables avec échantillonnage en fin de workflow

Le tracing distribué avec l'échantillonnage en fin de workflow aide les équipes IT à résoudre les problèmes dans des systèmes hautement distribués de gros volume où les équipes doivent observer toute la télémétrie de trace et échantillonner les traces qui contiennent des erreurs ou une latence anormale. L'échantillonnage en fin de workflow collecte toutes les informations sur cette trace une fois qu'elle est terminée.

L'échantillonnage en fin de workflow est un plus, mais c'est surtout un must lorsque les équipes ont besoin du meilleur niveau de granularité pour le dépannage.

Certaines organisations ont besoin de leur outil de tracing distribué pour observer et analyser chaque span (chaque passage d'un service à un autre) et faire remonter les traces plus exploitables pour le dépannage parce que les temps d'arrêt pourraient coûter des millions de dollars, surtout au cours de pics.

Par exemple, une organisation avec une charge d'environ trois millions de spans par minute voit des pics de 300 millions de spans par minute lors du lancement d'un nouveau produit. L'échantillonnage en début de workflow classique ne convient pas pour ce type d'organisation dont le volume de transactions est élevé.

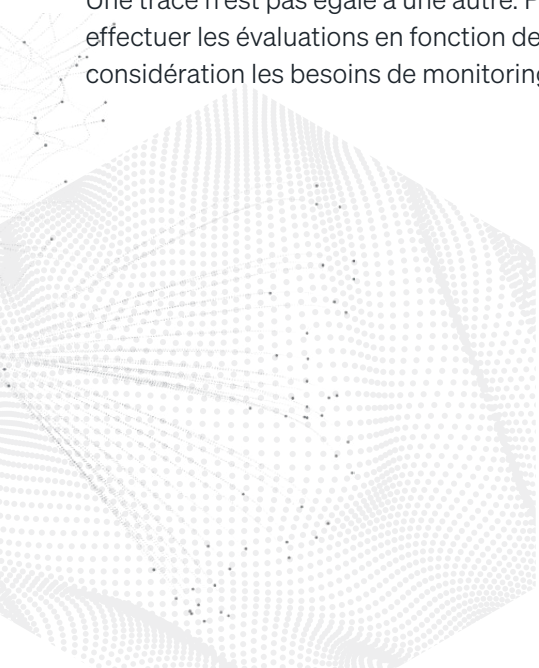
Une trace n'est pas égale à une autre. Pour choisir la meilleure méthode d'échantillonnage, les équipes doivent effectuer les évaluations en fonction des cas d'utilisation et de l'analyse du rapport coûts/avantages et prendre en considération les besoins de monitoring de chaque application.

Avantages de l'échantillonnage en fin de workflow

- Observe et analyse 100 % des traces
- Prélève des échantillons une fois que les traces sont terminées
- Permet de visualiser les traces contenant des erreurs ou un ralentissement atypique plus rapidement

Limites de l'échantillonnage en fin de workflow

- Peut exiger des passerelles, proxies et satellites supplémentaires pour exécuter le logiciel d'échantillonnage
- Dans certains cas, exige l'exécution de certaines tâches lourdes pour gérer et faire évoluer les logiciels tiers.
- Entraîne des frais supplémentaires pour la transmission et le stockage de plus de données.





Échantillonnage en début de workflow (haut) et échantillonnage en fin de workflow (bas)

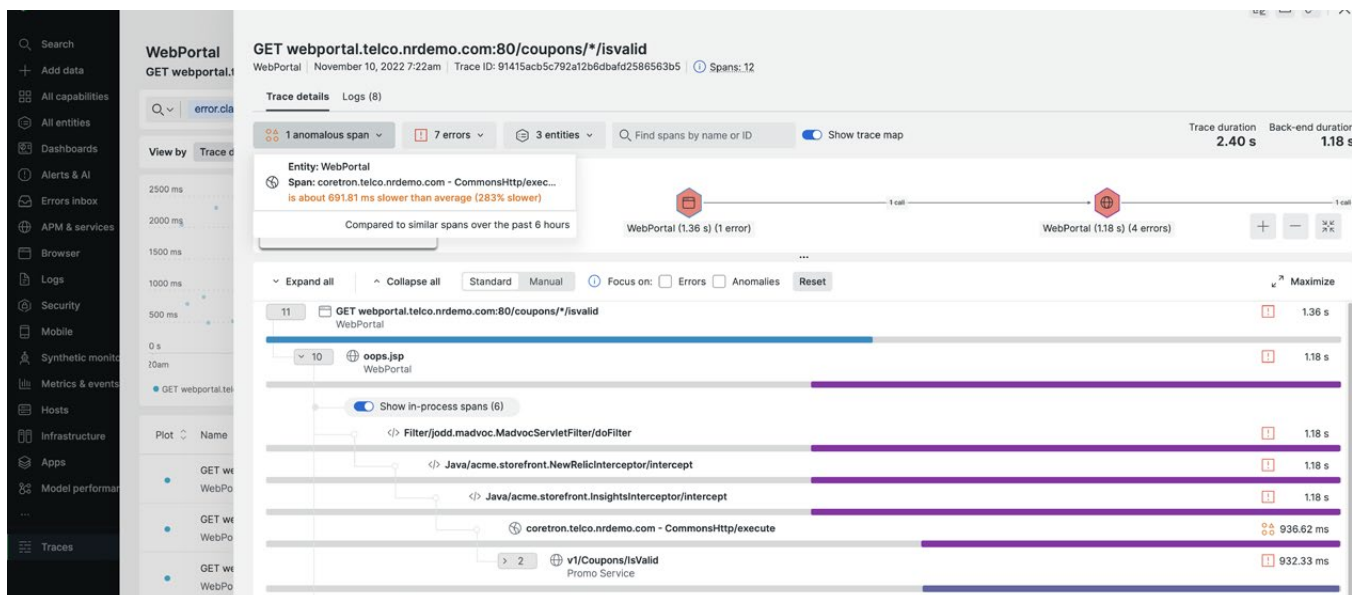


Analyse et visualisation

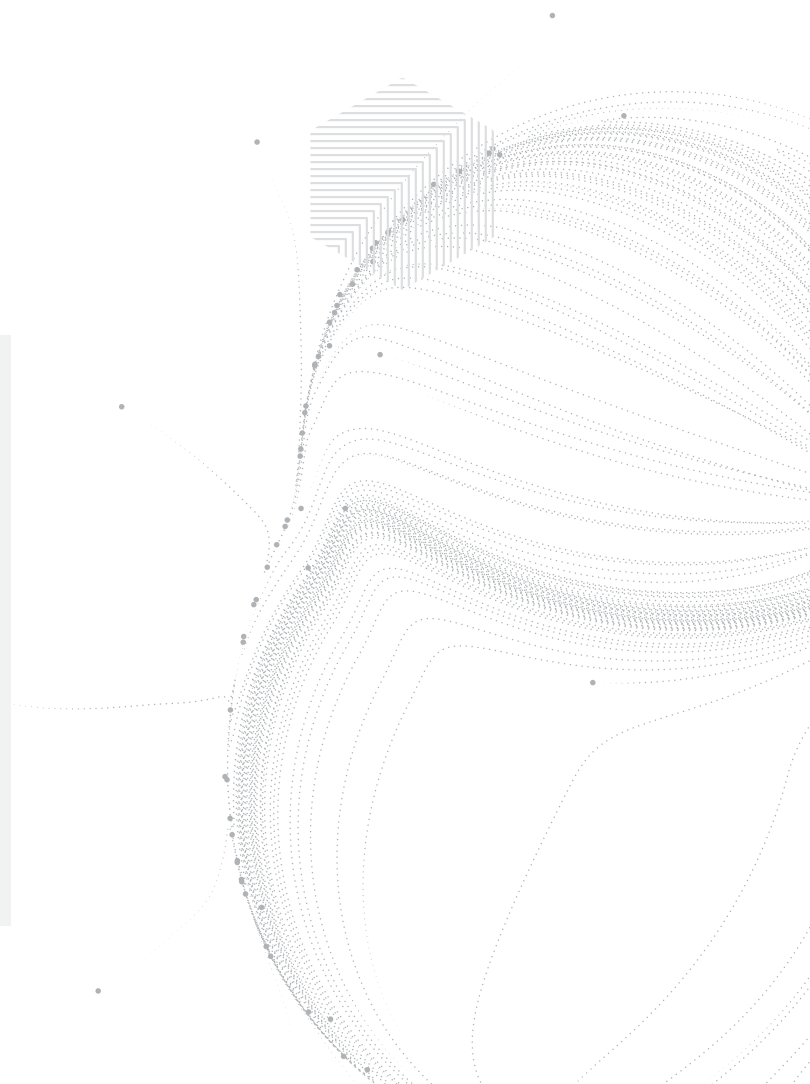
La collecte de données de traces constitue une perte de temps si les équipes IT ne disposent pas d'un moyen simple d'analyser et de visualiser les données sur des architectures complexes. Une plateforme d'observabilité complète permet aux équipes de voir toutes les données de télémétrie et d'activité au même endroit. Elle fournit également le contexte dont elles ont besoin pour comprendre la signification, prendre les mesures adéquates rapidement et utiliser les données de manière significative.

La visualisation d'une trace distribuée est, dans l'idéal, une structure en arborescence. La visualisation doit inclure les spans enfants qui font référence à un span parent et permettent aux équipes de voir quels spans ont une latence élevée et des erreurs au sein d'une trace. Cela permet également aux équipes de voir les détails exacts de l'erreur et les services qui sont lents en ayant des attributs détaillés pour trouver des erreurs et les résoudre rapidement.

Les prestataires d'observabilité comme New Relic utilisent cette structure de visualisation pour le dépannage et l'analyse.



Traçage distribué de New Relic



Allègement du fardeau de gestion

Le dépannage des systèmes distribués est un cas classique où l'on cherche l'aiguille dans une botte de foin. L'instrumentation des systèmes pour le traçage, puis la collecte et la visualisation des données peut être une tâche intensive et complexe à implémenter. Les solutions SaaS intégralement gérées permettent aux équipes d'éliminer la lourde tâche de déploiement, de gestion et d'évolution des passerelles ou satellites tiers pour la collecte de données.

La plateforme d'observabilité New Relic facilite l'instrumentation des applications avec un seul déploiement d'agent pour la plupart des langages de programmation et frameworks. Les équipes peuvent également utiliser des outils et des normes d'instrumentation open source pour instrumenter les environnements. OpenTelemetry est considéré comme la norme de l'instrumentation et de la collecte des données télémétriques open source.

La plateforme New Relic offre également un service d'échantillonnage en fin de workflow pleinement géré qui observe et analyse 100 % des spans sur un système distribué et fournit des visualisations pour les traces avec des erreurs ou une latence anormale afin que les équipes puissent rapidement identifier et résoudre les problèmes.

La plateforme observe chaque span et fournit les métriques, les données d'erreur et les traces essentielles dans une seule vue. Elle fournit des informations vitales en enregistrant les données les plus exploitables sur la plateforme New Relic. Le résultat est une visibilité inégalée sur des systèmes distribués, qui permet aux ingénieurs de comprendre l'impact de la latence ou des erreurs en aval avec des métriques détaillées et descendre jusqu'aux données de trace enregistrées pour trouver les traces les plus pertinentes.

Le tracing distribué est inclus dans New Relic APM, avec le transfert des données à faible latence et faible coût à partir des agents New Relic ou de l'instrumentation au sein des fonctions serverless, ou de toute autre source de données y compris l'instrumentation de tiers.

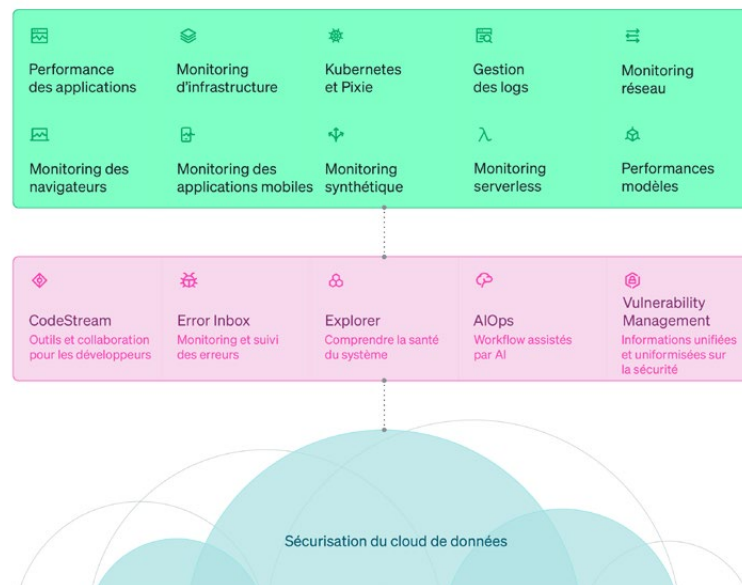
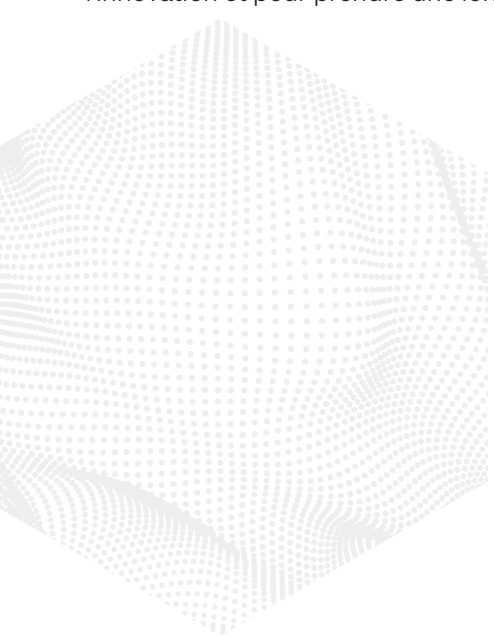
Avec New Relic, vous pouvez :

- Profiter d'un service cloud local pleinement géré qui évolue à la demande.
- Observer et analyser 100 % des traces sur les systèmes distribués.
- Visualiser les traces les plus exploitables qui contiennent des erreurs ou une latence anormale.
- Éliminer les tâches de déploiement, de gestion, de prise en charge et d'évolutivité de passerelles, ou de satellites tiers dans des environnements.
- Tirer parti de la prise en charge intégrale de l'instrumentation et des normes open source pour la télémétrie des traces
- Réduire le coût des frais de sortie des données des workloads de proximité jusqu'au cloud.
- Résoudre les problèmes plus efficacement.
- Réduire le temps moyen de détection (MTTD) et le délai moyen de résolution (MTTR) grâce à des traces exploitables haute fidélité.
- Permettre aux ingénieurs et aux développeurs de se concentrer sur ce qui est important, comme par exemple le développement de nouvelles fonctionnalités.

L'échantillonnage au début ou à la fin ? Les deux !

New Relic offre des options flexibles pour le tracing distribué afin que les équipes puissent décider d'échantillonner en début ou en fin de workload au niveau de l'application New Relic. Pour les applications critiques où les équipes doivent observer et analyser chaque trace, ils peuvent sélectionner l'échantillonnage en fin de workload sans se préoccuper de gérer l'infrastructure d'échantillonnage.

New Relic est le seul fournisseur d'observabilité qui donne aux équipes IT la flexibilité de sélectionner le tracing distribué avec un échantillonnage en début ou fin de workflow pleinement géré. Avec moins de choses à gérer, il y a plus de place pour l'innovation et pour prendre une longueur d'avance sur la concurrence.



La plateforme d'observabilité New Relic incorpore la gestion des logs, l'APM, le tracing distribué, le monitoring d'infrastructure, serverless, mobile, du navigateur, synthétique, Kubernetes, et plus encore.

Étapes suivantes

Pour commencer à utiliser New Relic APM, [inscrivez-vous](#) pour obtenir un compte gratuit dès aujourd'hui. Les comptes gratuits comprennent 100 Go/mois d'ingestion des données, un utilisateur Full Platform et un nombre illimité d'utilisateurs Basic.

Vous possédez déjà un compte New Relic ? Le démarrage avec le tracing distribué de New Relic APM est simple ; utilisez simplement notre tout dernier agent APM. Découvrez les options de paramétrage du tracing distribué.

Démarrer maintenant



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 uniformisée avec toutes les données télémétriques (métriques, événements, logs et traces) et les 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, les taux d'échecs des modifications, la fréquence de publication des nouvelles versions et 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.

