Engineer

Latence

Attente entre une requête et le début de la réponse

Ce que c’est

La latence est le temps qu’un système met à commencer à répondre : du départ d’une requête du client au retour du premier octet de la réponse. Les ingénieurs parlent aussi de délai, de retard ou, pour un trajet réseau mesuré à l’aller et au retour, de temps aller-retour (RTT). En performance web, la même idée s’appelle temps jusqu’au premier octet (TTFB). Le temps de réponse est plus large : il va jusqu’au dernier octet.

Il est facile de la confondre avec le débit et la bande passante, mais les trois répondent à des questions différentes. La bande passante est le nombre de bits qu’une liaison peut transporter par seconde. Le débit est la quantité de travail qu’un système termine par seconde. La latence est l’attente d’une unité de travail. Un traitement nocturne par lots peut avoir un excellent débit et une latence désastreuse, et un serveur de discussion l’inverse.

Chaque milliseconde vient de l’une de ces quelques sources :

- **la propagation**, le trajet du signal : dans une fibre optique, la lumière parcourt environ 200 km par milliseconde ; un trajet de 1 000 km coûte donc au moins 10 ms aller-retour avant toute exécution logicielle ;
- **la transmission**, l’envoi des octets sur la liaison, important pour de grandes charges utiles sur des liaisons lentes ;
- **la mise en file d’attente**, derrière d’autres travaux dans un tampon de routeur, un pool de threads, un pool de connexions ou un verrou, seule composante qui augmente avec la charge ;
- **le traitement**, le travail lui-même : la requête, le modèle de document, le chiffrement ;

- **les allers-retours de protocole**, la conversation avant le premier octet utile. Une nouvelle connexion HTTPS sur TCP exige d'abord une négociation TCP puis TLS, et TLS 1.3 a réduit sa négociation complète à un seul aller-retour.

Le groupe IP Performance Metrics de l'IETF a défini précisément la version réseau dans la RFC 2681 : le délai aller-retour va du premier bit du paquet envoyé au dernier bit de la réponse reçue. Deux de ses choix s'appliquent à tout service. Une réponse qui n'arrive jamais est comptée comme indéfinie, informellement infinie, plutôt que discrètement exclue de l'échantillon. Les statistiques décrites sont les percentiles, la médiane et le minimum. Le groupe observe aussi que le minimum ne reflète que la propagation et la transmission, tandis que tout ce qui le dépasse signale la congestion : cette distinction vaut pour une API comme pour un chemin entre routeurs.

Pour une requête web unique, curl décompose l'attente en étapes :

```
curl -s -o /dev/null "$URL" -w '
dns        %{time_namelookup}
connect    %{time_connect}
tls        %{time_appconnect}
first byte %{time_starttransfer}
total      %{time_total}
'
```

Chaque valeur est un nombre de secondes depuis le départ : les écarts donnent donc les coûts de la résolution du nom, de l'ouverture de connexion, de la négociation TLS, puis du temps de réflexion du serveur jusqu'au premier octet.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce que la latence ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Une supervision et des alertes qui signalent les incidents tant qu'ils restent mineurs, avec les chiffres qui montrent à quelle fréquence ils le restent.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Premier tableau de bord d'observabilité — première vue de l'entreprise sur ses propres systèmes

Premier tableau de bord d'observabilité Contactez-nous →

Pourquoi c’est important

La latence est ce que l’utilisateur ressent ; le débit est ce que reflète la facture du serveur. Quand une page est lente, il faut décider où investir. Les cinq composantes ci-dessus exigent des remèdes différents ; quatre montrent pourquoi acheter du matériel n’est pas un remède général. La distance se traite en se rapprochant : cache, CDN, région proche des utilisateurs. Les allers-retours se traitent en les réduisant : réutilisation des connexions, requêtes regroupées, chargements en parallèle. Le temps de traitement se réduit en effectuant moins de travail : index, meilleur plan de requête. Ajouter de la capacité peut réduire l’attente ; des processeurs plus rapides, davantage de mémoire ou de meilleures entrées-sorties peuvent aussi réduire le temps de traitement lorsque ces ressources constituent le goulet d’étranglement. Un serveur plus puissant n’est pas automatiquement plus rapide : mesurez où le temps est passé avant de choisir le matériel.

Les moyennes minimisent le problème, car les utilisateurs rencontrent la queue de distribution. Une page qui appelle dix services backend en parallèle attend le plus lent. Si chaque appel a, indépendamment des autres, une probabilité de 1 % d’être lent, la part des chargements rencontrant au moins un appel lent est de $1 - 0{,}99^{10}$, soit environ 9,6 %. Un retard rare dans les parties devient courant dans l’ensemble.

C’est pourquoi les objectifs sérieux s’écrivent en percentiles, par exemple 99 % des requêtes servies en 300 ms au maximum sur 30 jours, et pourquoi le 99e percentile est généralement le chiffre à surveiller. La médiane dit comment le service se comporte un bon jour ; les percentiles élevés disent ce que ressentent les clients sur le point de partir.

À quoi cela ressemble en pratique

Dans une application cartographique, la latence du rendu des couches et des requêtes sur les données de localisation nuisait aux utilisateurs comme à la fiabilité du backend. Le travail portait sur le traitement des requêtes spatiales d’un backend utilisant une ancienne base Microsoft SQL Server. Le passage à PostgreSQL avec PostGIS a réduit le temps de réponse moyen de 2,5 secondes à moins de 250 millisecondes, améliorant d’un facteur 10 les performances des requêtes spatiales. Le travail comprenait des index GiST et SP-GiST sur les colonnes spatiales, plus de 50 millions d’enregistrements migrés et une configuration ajustée aux accès concurrents ; la charge CPU du backend a baissé de 65 %.

La latence plafonne aussi le débit lorsque plusieurs opérations se disputent une même ligne. Une note sur la vente du dernier billet une seule fois situe le plafond pour un article unique à environ le nombre de ventes qu'un verrou permet, divisé par sa durée de détention. Tout travail lent sous ce verrou abaisse le plafond : un appel de paiement effectué sans relâcher le verrou le ramène à environ deux ventes par seconde. Même une mesure de sûreté a un prix ici. Un serveur de secours synchrone PostgreSQL ajoute de la latence à chaque validation, allongeant précisément la durée de détention du verrou par laquelle on divise pour obtenir ce plafond.

Pour trouver où passe le temps, la supervision développée en interne pour une plateforme enregistre la latence de l'API dans sa propre base et trace les requêtes avec OpenTelemetry : une requête lente peut ainsi être suivie dans le frontend, l'API et la base de données, plutôt que faire l'objet de suppositions.

Où les problèmes surviennent

La moyenne cache la queue de distribution. Une moyenne de 100 ms reste compatible avec une requête sur cinquante prenant trois secondes. Enregistrez la latence en histogramme et lisez ses percentiles. Attention à leur combinaison : faire la moyenne des 99es percentiles de cinq serveurs ne donne pas le 99e percentile du parc. Prometheus l'explique dans sa documentation : les quantiles calculés par un résumé ne peuvent pas être agrégés, tandis que les effectifs des classes d'histogramme peuvent être additionnés avant le calcul du quantile.

Le chronomètre démarre au mauvais endroit. Un serveur qui commence à mesurer au début de son gestionnaire ne comptabilise pas le temps passé dans l'équilibreur de charge, la file d'acceptation ou l'attente d'une connexion du pool. Les générateurs de charge qui attendent chaque réponse avant d'envoyer la requête suivante en envoient moins précisément lorsque le système se bloque : le blocage apparaît donc à peine dans les résultats. Mesurez aussi depuis le client et laissez le serveur indiquer sa part dans un en-tête standard :

```
Server-Timing: db;dur=53, app;dur=47.2
```

La conversation multiplie les échanges. Une page qui exécute une requête par ligne d'une liste paie un aller-retour par ligne. Avec une base dans une autre région et 30 ms par aller-retour, cent lignes coûtent trois secondes avant le

moindre travail réel des requêtes. Une requête avec jointure ou un appel groupé ne paie la distance qu'une fois.

Les nouvelles tentatives multiplient la charge. Un client qui abandonne après une seconde et réessaie trois fois multiplie par quatre le trafic d'un serveur lent, le ralentissant davantage. Donnez une échéance globale à chaque requête, espacez exponentiellement les tentatives avec une variation aléatoire et plafonnez le nombre de nouvelles tentatives.

Pour aller plus loin

- RFC 2681 : mesure du délai aller-retour pour IPPM — le délai aller-retour défini au bit près, avec un traitement explicite des réponses perdues et des percentiles.
- RFC 7679 : mesure du délai aller simple pour IP Performance Metrics — l'équivalent en aller simple, publié comme standard Internet.
- Server Timing — l'en-tête du W3C permettant au serveur d'indiquer au navigateur où son temps est passé.
- Histogrammes et résumés — pourquoi les quantiles d'un résumé ne se combinent pas, contrairement aux effectifs des classes d'histogramme.
- RFC 8446 : TLS 1.3 — la conception de la négociation, dont l'aller-retour supprimé.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce que la latence ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/latency/