Latence des nombres

Points Clés

- Il semble probable que la vidéo discute des chiffres de latence standard que les programmeurs devraient connaître, en se basant sur son titre et le contenu en ligne associé.
- La recherche suggère que ces chiffres incluent des temps pour des opérations telles que l'accès au cache L1 (0,5 ns) et les allers-retours réseau (jusqu'à 150 ms), variant selon le matériel.
- Les preuves penchent en faveur de ces chiffres étant approximatifs, avec des mises à jour reflétant les avancées technologiques, notamment pour les SSDs et les réseaux.

Introduction

La vidéo "Latency Numbers Programmer Should Know: Crash Course System Design #1"couvre probablement les chiffres de latence essentiels pour les opérations informatiques, cruciaux pour la conception de systèmes. Ces chiffres aident les programmeurs à comprendre les impacts sur les performances et à optimiser les systèmes.

Chiffres de Latence et Leur Importance

La latence est le délai entre le lancement et l'achèvement d'une opération, comme l'accès à la mémoire ou l'envoi de données sur un réseau. La vidéo liste probablement des latences typiques, telles que : - Référence au cache L1 à 0,5 nanosecondes (ns), l'accès mémoire le plus rapide. - Un aller-retour au sein du même centre de données à 500 microsecondes (us) ou 0,5 millisecondes (ms), affectant les systèmes distribués.

Ces chiffres, bien qu'approximatifs, guident les décisions en conception de systèmes, comme le choix entre la mémoire et le stockage sur disque.

Contexte dans la Conception de Systèmes

Comprendre ces latences aide à optimiser le code, à faire des compromis et à améliorer l'expérience utilisateur. Par exemple, savoir qu'une recherche de disque prend 10 ms peut influencer la conception de la base de données pour minimiser ces opérations.

Détail Inattendu

Un aspect intéressant est la manière dont ces chiffres, comme les temps de lecture SSD, se sont améliorés avec la technologie, tandis que les latences CPU de base comme l'accès au cache L1 restent stables, montrant l'impact inégal de l'évolution du matériel.

Note de Sondage : Analyse Détaillée des Chiffres de Latence de la Vidéo

Cette note fournit une exploration complète des chiffres de latence probablement discutés dans la vidéo “Latency Numbers Programmer Should Know: Crash Course System Design #1”, basée sur le contenu en ligne disponible et les ressources associées. L’analyse vise à synthétiser des informations pour les programmeurs et les concepteurs de systèmes, offrant à la fois un résumé et des insights détaillés sur la signification de ces chiffres.

Contexte et Contexte La vidéo, accessible sur YouTube, fait partie d’une série sur la conception de systèmes, se concentrant sur les chiffres de latence critiques pour les programmeurs. La latence, définie comme le délai entre le lancement et l’achèvement d’une opération, est cruciale pour comprendre les performances du système. Étant donné le titre de la vidéo et les recherches associées, il semble qu’elle couvre les chiffres de latence standard popularisés par des figures comme Jeff Dean de Google, souvent référencées dans les communautés de programmation.

Les recherches en ligne ont révélé plusieurs ressources discutant de ces chiffres, y compris un GitHub Gist intitulé “Latency Numbers Every Programmer Should Know”(GitHub Gist) et un article Medium de 2023 (Medium Article). Ces sources, ainsi qu’un article High Scalability de 2013 (High Scalability), ont fourni une base pour compiler le contenu probable de la vidéo.

Compilation des Chiffres de Latence Sur la base des informations recueillies, le tableau suivant résume les chiffres de latence standard, probablement discutés dans la vidéo, avec des explications pour chaque opération :

Opération	Latence (ns)	Latence (us)	Latence (ms)	Explication
Référence au cache L1	0,5	-	-	Accéder aux données dans le cache de niveau 1, la mémoire la plus rapide près du CPU.
Mauvaise prédiction de branche	5	-	-	Pénalité lorsque le CPU prédit incorrectement une branche conditionnelle.
Référence au cache L2	7	-	-	Accéder aux données dans le cache de niveau 2, plus grand que L1 mais plus lent.
Verrouillage/déverrouillage de mutex	25	-	-	Temps pour acquérir et libérer un mutex dans les programmes multithreads.
Référence à la mémoire principale	100	-	-	Accéder aux données de la mémoire vive principale (RAM).

Opération	Latence (ns)	Latence (us)	Latence (ms)	Explication
Compresser 1K octets avec Zippy	10,000	10	-	Temps pour compresser 1 kilo-octet en utilisant l’algorithme Zippy.
Envoyer 1 KB d’octets sur un réseau 1 Gbps	10,000	10	-	Temps pour transmettre 1 kilo-octet sur un réseau à 1 gigabit par seconde.
Lire 4 KB aléatoirement à partir d’un SSD	150,000	150	-	Lecture aléatoire de 4 kilo-octets à partir d’un disque à semi-conducteurs.
Lire 1 Mo séquentiellement à partir de la mémoire	250,000	250	-	Lecture séquentielle de 1 méga-octet à partir de la mémoire principale.
Aller-retour au sein du même centre de données	500,000	500	0,5	Temps d’aller-retour réseau au sein du même centre de données.
Lire 1 Mo séquentiellement à partir d’un SSD	1,000,000	1,000	1	Lecture séquentielle de 1 méga-octet à partir d’un SSD.
Recherche de disque dur	10,000,000	10,000	10	Temps pour qu’un disque dur cherche une nouvelle position.
Lire 1 Mo séquentiellement à partir du disque	20,000,000	20,000	20	Lecture séquentielle de 1 méga-octet à partir d’un disque dur.
Envoyer un paquet CA->Pays-Bas->CA	150,000,000	150,000	150	Temps d’aller-retour pour un paquet réseau de Californie aux Pays-Bas.

Ces chiffres, principalement de 2012 avec quelques mises à jour, reflètent les performances matérielles typiques, avec des variations notées dans les discussions récentes, notamment pour les SSDs et les réseaux en raison des avancées technologiques.

Analyse et Implications Les chiffres de latence ne sont pas fixes et peuvent varier en fonction du matériel spécifique et des configurations. Par exemple, un article de blog de 2020 par Ivan Pesin (Pesin Space) a noté que les latences de disque et de réseau se sont améliorées grâce à de meilleurs SSDs (NVMe) et des réseaux plus rapides (10/100Gb), mais les latences CPU de base comme l’accès au cache L1 restent stables. Cette évolution inégale souligne l’importance du contexte dans la conception de systèmes.

En pratique, ces chiffres guident plusieurs aspects : - **Optimisation des Performances** : Minimiser les opérations à haute latence, comme les recherches de disque (10 ms), peut considérablement améliorer la vitesse de l’application. Par exemple, mettre en cache les données fréquemment accédées en mémoire (250 us pour une lecture de 1 Mo) plutôt que sur disque peut réduire les temps d’attente. - **Décisions de Compromis** : Les concepteurs de systèmes font souvent face à des choix, comme utiliser des caches en mémoire ou des bases de données. Savoir qu’une référence à la mémoire principale (100 ns) est 200 fois plus rapide qu’une référence au cache L1 (0,5 ns) peut informer ces décisions. - **Expérience Utilisateur** : Dans les applications web, les latences réseau, comme un aller-retour de centre de données (500 us), peuvent affecter les temps de chargement des pages, impactant la satisfaction de l’utilisateur. Un article

de blog Vercel de 2024 (Vercel Blog) a souligné cela pour le développement frontend, notant comment les cascades réseau peuvent accumuler la latence.

Contexte Historique et Mises à Jour Les chiffres originaux, attribués à Jeff Dean et popularisés par Peter Norvig, datent d'environ 2010, avec des mises à jour par des chercheurs comme Colin Scott (Interactive Latencies). Un article Medium de 2019 par Dan Hon (Dan Hon Medium) a ajouté des latences humoristiques mais pertinentes, comme le redémarrage d'un MacBook Pro (90 secondes), illustrant des retards plus larges liés à la technologie. Cependant, les chiffres de latence de base ont vu peu de changements, le GitHub Gist suggérant qu'ils restent "quite similar"jusqu'en 2023, en raison des limitations physiques.

Conclusion et Recommandations Pour les programmeurs et les concepteurs de systèmes, mémoriser ces chiffres de latence fournit un modèle mental pour l'ajustement des performances. Ils doivent être traités comme des directives, avec des benchmarks réels effectués pour un matériel spécifique. Rester à jour, notamment dans les technologies émergentes comme le calcul quantique ou les réseaux 5G, sera crucial. Des ressources comme le GitHub Gist et l'article Medium offrent des points de départ pour une exploration plus approfondie.

Cette analyse, fondée sur le contenu probable de la vidéo et complétée par une recherche en ligne approfondie, souligne la pertinence durable des chiffres de latence dans l'informatique, avec un appel à s'adapter aux changements technologiques pour une conception de système optimale.

Citations Clés

- Latency Numbers Every Programmer Should Know GitHub Gist
- Latency Numbers Programmer Should Know YouTube Video
- Updated Latency Numbers Medium Article
- More Numbers Every Awesome Programmer Must Know High Scalability
- Latency Numbers Every Web Developer Should Know Vercel Blog
- Latency Numbers Every Engineer Should Know Pesin Space Blog
- More Latency Numbers Every Programmer Should Know Dan Hon Medium
- Numbers Every Programmer Should Know By Year Interactive Latencies