Latenz Zahlen

Wichtige Punkte

- Es scheint wahrscheinlich, dass das Video Standard-Latenzwerte behandelt, die Programmierer kennen sollten, basierend auf seinem Titel und dem dazugehörigen Online-Content.
- Forschung legt nahe, dass diese Werte Zeiten für Operationen wie L1-Cache-Zugriff (0,5 ns) und Netzwerk-Roundtrips (bis zu 150 ms) umfassen, die je nach Hardware variieren.
- Die Beweise deuten darauf hin, dass diese Zahlen ungefähr sind, mit Aktualisierungen, die technologische Fortschritte, insbesondere bei SSDs und Netzwerken, widerspiegeln.

Einführung

Das Video “Latency Numbers Programmer Should Know: Crash Course System Design #1”behandelt wahrscheinlich wesentliche Latenzwerte für Computeroperationen, die für das Systemdesign entscheidend sind. Diese Zahlen helfen Programmierern, die Auswirkungen auf die Leistung zu verstehen und Systeme zu optimieren.

Latenzwerte und ihre Bedeutung

Latenz ist die Verzögerung zwischen dem Initiieren und dem Abschließen einer Operation, wie z.B. dem Zugriff auf den Speicher oder dem Senden von Daten über ein Netzwerk. Das Video listet wahrscheinlich typische Latenzen auf, wie z.B.: - L1-Cache-Referenz bei 0,5 Nanosekunden (ns), der schnellste Speicherzugriff. - Eine Roundtrip innerhalb desselben Rechenzentrums bei 500 Mikrosekunden (us) oder 0,5 Millisekunden (ms), was verteilte Systeme beeinflusst.

Diese Zahlen, obwohl ungefähr, leiten Entscheidungen im Systemdesign, wie die Wahl zwischen Speicher und Festplattenspeicher.

Kontext im Systemdesign

Das Verständnis dieser Latenzen hilft bei der Optimierung von Code, der Abwägung von Kompromissen und der Verbesserung der Benutzererfahrung. Zum Beispiel kann das Wissen, dass ein Festplattenzugriff 10 ms dauert, das Datenbankdesign beeinflussen, um solche Operationen zu minimieren.

Unerwartetes Detail

Ein interessanter Aspekt ist, wie sich diese Zahlen, wie z.B. SSD-Lesezeiten, mit der Technologie verbessert haben, während die Kern-CPU-Latenzen wie der L1-Cache-Zugriff stabil geblieben sind, was den ungleichen Einfluss der Hardwareentwicklung zeigt.

Umfragehinweis: Detaillierte Analyse der Latenzwerte aus dem Video

Dieser Hinweis bietet eine umfassende Untersuchung der Latenzwerte, die wahrscheinlich im Video "Latency Numbers Programmer Should Know: Crash Course System Design #1"behandelt werden, basierend auf verfügbarem Online-Content und verwandten Ressourcen. Die Analyse zielt darauf ab, Informationen für Programmierer und Systemdesigner zu synthetisieren, und bietet sowohl eine Zusammenfassung als auch detaillierte Einblicke in die Bedeutung dieser Zahlen.

Hintergrund und Kontext Das Video, das auf YouTube zugänglich ist, ist Teil einer Serie zum Systemdesign und konzentriert sich auf Latenzwerte, die für Programmierer entscheidend sind. Latenz, definiert als die Zeitverzögerung zwischen dem Initiieren und dem Abschließen einer Operation, ist entscheidend für das Verständnis der Systemleistung. Angesichts des Titels des Videos und verwandter Suchen scheint es, dass es Standard-Latenzwerte abdeckt, die von Persönlichkeiten wie Jeff Dean von Google populär gemacht wurden und oft in Programmierergemeinschaften zitiert werden.

Online-Suchen ergaben mehrere Ressourcen, die diese Zahlen diskutieren, einschließlich eines GitHub Gist mit dem Titel "Latency Numbers Every Programmer Should Know"(GitHub Gist) und einem Medium-Artikel aus dem Jahr 2023 (Medium Artikel). Diese Quellen, zusammen mit einem High Scalability Beitrag aus dem Jahr 2013 (High Scalability), bildeten die Grundlage für die Zusammenstellung des wahrscheinlichsten Inhalts des Videos.

Zusammenstellung der Latenzwerte Basierend auf den gesammelten Informationen fasst die folgende Tabelle die Standard-Latenzwerte zusammen, die wahrscheinlich im Video besprochen werden, mit Erklärungen für jede Operation:

Operation	Latenz (ns)	Latenz (us)	Latenz (ms)	Erklärung
L1-Cache-Referenz	0,5	-	-	Zugriff auf Daten im Level 1 Cache, der schnellste Speicher nahe der CPU.
Branch mispredict	5	-	-	Strafe, wenn die CPU einen bedingten Zweig falsch vorhersagt.
L2-Cache-Referenz	7	-	-	Zugriff auf Daten im Level 2 Cache, größer als L1, aber langsamer.
Mutex lock/unlock	25	-	-	Zeit zum Erhalten und Freigeben eines Mutex in mehrthreadigen Programmen.
Hauptspeicher-Referenz	100	-	-	Zugriff auf Daten aus dem Hauptspeicher (RAM).
Komprimieren von 1 KB mit Zippy	10,000	10	-	Zeit zum Komprimieren von 1 Kilobyte mit dem Zippy-Algorithmus.

Operation	Latenz (ns)	Latenz (us)	Latenz (ms)	Erklärung
Senden von 1 KB über 1 Gbps Netzwerk	10,000	10	-	Zeit zum Übertragen von 1 Kilobyte über ein 1 Gigabit pro Sekunde Netzwerk.
Lesen von 4 KB zufällig von SSD	150,000	150	-	Zufälliges Lesen von 4 Kilobytes von einer Festplatte.
Lesen von 1 MB sequenziell aus dem Speicher	250,000	250	-	Sequenzielles Lesen von 1 Megabyte aus dem Hauptspeicher.
Roundtrip innerhalb desselben Rechenzentrums	500,000	500	0,5	Netzwerk-Roundtrip-Zeit innerhalb desselben Rechenzentrums.
Lesen von 1 MB sequenziell von SSD	1,000,000	1,000	1	Sequenzielles Lesen von 1 Megabyte von einer SSD.
HDD suchen	10,000,000	10,000	10	Zeit für eine Festplatte, um eine neue Position zu suchen.
Lesen von 1 MB sequenziell von der Festplatte	20,000,000	20,000	20	Sequenzielles Lesen von 1 Megabyte von einer Festplatte.
Senden eines Pakets CA->Niederlande->CA	150,000,000	150,000	150	Roundtrip-Zeit für ein Netzwerkpaket von Kalifornien zu den Niederlanden.

Diese Zahlen, hauptsächlich aus dem Jahr 2012 mit einigen Aktualisierungen, spiegeln die typische Hardwareleistung wider, wobei Variationen in jüngsten Diskussionen, insbesondere bei SSDs und Netzwerken, aufgrund technologischer Fortschritte festgestellt wurden.

Analyse und Implikationen Die Latenzwerte sind nicht fest und können je nach spezifischer Hardware und Konfiguration variieren. Zum Beispiel bemerkte ein Blogpost von Ivan Pesin aus dem Jahr 2020 (Pesin Space), dass sich die Latenzen von Festplatten und Netzwerken dank besserer SSDs (NVMe) und schnellerer Netzwerke (10/100Gb) verbessert haben, während die Kern-CPU-Latenzen wie der L1-Cache-Zugriff stabil geblieben sind. Diese ungleiche Entwicklung unterstreicht die Bedeutung des Kontextes im Systemdesign.

In der Praxis leiten diese Zahlen mehrere Aspekte: - **Leistungsoptimierung**: Das Minimieren von Operationen mit hoher Latenz, wie Festplattenzugriffe (10 ms), kann die Anwendungsgeschwindigkeit erheblich verbessern. Zum Beispiel kann das Cachen häufig zugreifender Daten im Speicher (250 us für 1 MB Lesen) anstelle der Festplatte Wartezeiten reduzieren. - **Abwägungsentscheidungen**: Systemdesigner stehen oft vor Entscheidungen, wie der Verwendung von In-Speicher-Caches anstelle von Datenbanken. Das Wissen, dass ein Hauptspeicherzugriff (100 ns) 200 Mal schneller ist als ein L1-Cache-Zugriff (0,5 ns), kann solche Entscheidungen beeinflussen. - **Benutzererfahrung**: Bei Webanwendungen können Netzwerklatenzen, wie ein Rechenzentrums-Roundtrip (500 us), die Seitenladezeiten beeinflussen und die Benutzerzufriedenheit beeinträchtigen. Ein Vercel-Blogpost aus dem Jahr 2024 (Vercel Blog) betonte dies für die Frontend-Entwicklung und bemerkte, wie Netzwerkwasserfälle die Latenz kumulieren können.

Historischer Kontext und Aktualisierungen Die ursprünglichen Zahlen, die Jeff Dean zugeschrieben werden und von Peter Norvig populär gemacht wurden, stammen aus etwa 2010, mit Aktualisierungen durch Forscher wie Colin Scott (Interactive Latencies). Ein Medium-Post von Dan Hon aus dem Jahr 2019 (Dan Hon Medium) fügte humorvolle, aber relevante Latenzen hinzu, wie das Neustarten eines MacBook Pro (90 Sekunden), was breitere technologische Verzögerungen illustriert. Allerdings haben sich die Kernlatenzwerte nur minimal verändert, wobei der GitHub Gist vorschlägt, dass sie “noch ziemlich ähnlich”sind, basierend auf physikalischen Einschränkungen, bis 2023.

Schlussfolgerung und Empfehlungen Für Programmierer und Systemdesigner bietet das Auswendiglernen dieser Latenzwerte ein mentales Modell für die Leistungsanpassung. Sie sollten als Richtlinien behandelt werden, mit tatsächlichen Benchmarks für spezifische Hardware. Auf dem Laufenden bleiben, insbesondere bei aufkommenden Technologien wie Quantencomputing oder 5G-Netzwerken, wird entscheidend sein. Ressourcen wie der GitHub Gist und der Medium-Artikel bieten Ausgangspunkte für weitere Erkundungen.

Diese Analyse, die auf dem wahrscheinlichsten Inhalt des Videos basiert und durch umfangreiche Online-Recherchen ergänzt wird, unterstreicht die bleibende Relevanz von Latenzzahlen in der Informatik, mit einem Aufruf, sich an technologische Veränderungen anzupassen, um ein optimales Systemdesign zu gewährleisten.

Wichtige Zitate

- 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