Engineer

Latenz

⏱ Wartezeit von einer Anfrage bis zum Beginn der Antwort

Was es ist

Latenz ist die Zeit, die ein System braucht, um mit der Antwort zu beginnen: vom Absenden einer Anfrage durch den Client bis zur Rückkehr des ersten Antwortbytes. Ingenieure sprechen auch von Verzögerung, Lag oder, bei einem in beide Richtungen gemessenen Netzwerkpfad, von Round-Trip Time (RTT). Bei der Webperformance heißt dieselbe Idee Time to First Byte (TTFB). Die Antwortzeit ist die umfassendere Größe und reicht bis zum letzten Byte der Antwort.

Latenz wird leicht mit Durchsatz und Bandbreite verwechselt. Die drei beantworten unterschiedliche Fragen. Bandbreite gibt an, wie viele Bits eine Verbindung pro Sekunde transportieren kann. Durchsatz beschreibt, wie viel Arbeit ein System pro Sekunde abschließt. Latenz sagt, wie lange eine einzelne Arbeitseinheit wartet. Ein nächtlicher Stapelverarbeitungslauf kann hervorragenden Durchsatz bei schlechter Latenz haben; bei einem Chatserver kann es umgekehrt sein.

Jede Millisekunde stammt aus einer von wenigen Quellen:

- **Signalausbreitung**, also das Überwinden der Entfernung: Licht in Glasfaser legt ungefähr 200 km pro Millisekunde zurück. Eine Strecke von 1.000 km kostet deshalb für Hin- und Rückweg mindestens 10 ms, bevor irgendeine Software läuft;
- **Übertragung**, also die Bytes auf die Leitung zu bringen. Das fällt bei großen Datenmengen auf langsamen Verbindungen ins Gewicht;
- **Warten in Warteschlangen**, hinter anderer Arbeit in einem Routerpuffer, Thread-Pool, Verbindungspool oder an einer Sperre: die eine Komponente, die mit der Last wächst;
- **Verarbeitung**, die eigentliche Arbeit: die Abfrage, das Template, die Verschlüsselung;

- **Protokoll-Roundtrips**, das Gespräch vor dem ersten nützlichen Byte. Eine neue HTTPS-Verbindung über TCP braucht zuerst einen TCP-Handshake und einen TLS-Handshake. TLS 1.3 verkürzte seinen vollständigen Handshake auf einen einzigen Roundtrip.

Die IP-Performance-Metrics-Gruppe der IETF hat die Netzwerkvariante in RFC 2681 genau festgelegt: Die Roundtrip-Verzögerung reicht vom ersten gesendeten Bit eines Pakets bis zum letzten empfangenen Bit der Antwort. Zwei ihrer Entscheidungen lassen sich auf jeden Dienst übertragen. Eine nie eintreffende Antwort gilt als undefiniert, umgangssprachlich unendlich, statt stillschweigend aus der Stichprobe entfernt zu werden. Und die beschriebenen statistischen Kennzahlen sind Perzentile, Median und Minimum. Der RFC hält außerdem fest, dass das Minimum nur Signalausbreitung und Übertragung widerspiegelt, während alles darüber auf Überlastung hinweist. Diese Unterscheidung gilt für eine API ebenso wie für einen Pfad zwischen Routern.

Für eine einzelne Webanfrage kann curl die Wartezeit in ihre Abschnitte zerlegen:

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

Jeder Wert zählt Sekunden seit dem Start. Die Abstände dazwischen sind deshalb die jeweiligen Kosten: Namensauflösung, Verbindungsaufbau, TLS-Handshake und anschließend die eigene Denkzeit des Servers bis zum ersten Byte.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist Latenz?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Monitoring und Alarme melden sich, solange ein Fehler noch klein ist; Kennzahlen zeigen, wie oft das gelingt.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Das erste Observability-Dashboard aufgebaut — der erste Einblick des Unternehmens in die eigenen Systeme

Das erste Observability-Dashboard aufgebaut Kontakt aufnehmen →

Warum es wichtig ist

Latenz ist das, was ein Nutzer spürt; der Durchsatz spiegelt sich in der Serverrechnung wider. Ist eine Seite langsam, stellt sich die Frage, wofür Geld ausgegeben werden soll. Die fünf Komponenten oben brauchen unterschiedliche Lösungen. Vier davon zeigen, warum der Kauf von Hardware kein allgemeines Heilmittel ist. Entfernung lässt sich durch größere Nähe verringern: mit einem Cache, einem CDN oder einer Region nahe den Nutzern. Roundtrips verringert man, indem man weniger davon braucht: Verbindungen wiederverwenden, Abfragen bündeln, parallel abrufen. Verarbeitungszeit verringert man durch weniger Arbeit: mit einem Index oder einem besseren Abfrageplan. Zusätzliche Kapazität kann Wartezeiten in Warteschlangen verringern. Schnellere Prozessoren, mehr Speicher oder bessere Ein- und Ausgabe können auch die Verarbeitungszeit verkürzen, wenn diese Ressourcen den Engpass bilden. Ein größerer Server ist nicht automatisch schneller: Messen Sie vor der Hardwarewahl, wo die Zeit vergeht.

Mittelwerte lassen das Problem kleiner aussehen, als es ist, weil Nutzer die langsamen Ausreißer erleben. Eine Seite, die zehn Backend-Dienste parallel aufruft, wartet auf den langsamsten. Hat jeder Aufruf unabhängig von den anderen eine Wahrscheinlichkeit von 1 %, langsam zu sein, treffen $1 - 0{,}99^{10}$, also ungefähr 9,6 % der Seitenaufrufe, auf mindestens einen langsamen Aufruf. Eine seltene Verzögerung in den Teilen wird im Ganzen häufig.

Deshalb werden ernst gemeinte Ziele als Perzentile formuliert, etwa innerhalb eines Zeitraums von 30 Tagen 99 % der Anfragen binnen 300 ms zu beantworten, und deshalb ist das 99. Perzentil gewöhnlich der Wert, den man beobachten sollte. Der Median zeigt, wie sich der Dienst an einem guten Tag anfühlt. Die hohen Perzentile zeigen, wie er sich für die Kunden anfühlt, die kurz vor dem Absprung stehen.

Wie es in der Praxis aussieht

Bei einer geografischen Kartenanwendung beeinträchtigte die Latenz beim Rendern von Kartenebenen und beim Abfragen von Standortdaten sowohl die Nutzer als auch die Zuverlässigkeit des Backends. Die Arbeit konzentrierte sich

auf die Verarbeitung räumlicher Abfragen in einem Backend, das eine Altdatenbank mit Microsoft SQL Server verwendete. Der Wechsel zu PostgreSQL mit PostGIS senkte die durchschnittliche Antwortzeit von 2,5 Sekunden auf unter 250 Millisekunden, eine 10-fache Verbesserung der Leistung räumlicher Abfragen. Dahinter standen GiST- und SP-GiST-Indizes auf den räumlichen Spalten, mehr als 50 Millionen migrierte Datensätze und eine auf gleichzeitige Zugriffe abgestimmte Konfiguration. Die CPU-Last im Backend sank um 65 %.

Latenz begrenzt auch den Durchsatz, wenn mehrere Zugriffe um dieselbe Zeile konkurrieren. Eine Notiz über den genau einmaligen Verkauf der letzten Eintrittskarte gibt die Obergrenze für einen einzelnen Artikel näherungsweise als Anzahl der Verkäufe pro Sperrvorgang, geteilt durch die Haltedauer dieser Sperre, an. Alles Langsame innerhalb der Sperre senkt diese Grenze: Ein Zahlungsaufruf während des Haltens drückt sie auf ungefähr zwei Verkäufe pro Sekunde. Selbst eine Sicherheitsmaßnahme hat hier ihren Preis. Ein synchroner Standby in PostgreSQL fügt jedem Commit Latenz hinzu und verlängert damit genau die Haltedauer der Sperre, durch die bei dieser Obergrenze geteilt wird.

Um herauszufinden, wo die Zeit bleibt, zeichnet das eigene Monitoring einer Plattform API-Latenzen in der eigenen Datenbank auf und verfolgt Anfragen über OpenTelemetry. Eine langsame Anfrage lässt sich damit über Frontend, API und Datenbank hinweg verfolgen, statt ihre Ursache zu erraten.

Wo es schiefgeht

Der Durchschnitt verbirgt die langsamen Ausreißer. Ein Mittelwert von 100 ms ist damit vereinbar, dass jede fünfzigste Anfrage drei Sekunden braucht. Erfassen Sie Latenzen als Histogramm und lesen Sie dessen Perzentile. Achten Sie dann auf die Zusammenführung: Die 99. Perzentile von fünf Servern lassen sich nicht zum 99. Perzentil des gesamten Bestands mitteln. Prometheus erläutert dies in seiner eigenen Dokumentation: Von einer Summary berechnete Quantile lassen sich nicht aggregieren. Die Klassen eines Histogramms können dagegen zuerst summiert und das Quantil anschließend bestimmt werden.

Die Zeitmessung beginnt am falschen Punkt. Ein Server, der erst ab dem Start seines Handlers misst, übersieht die Zeit im Lastverteiler, in der Accept-Warteschlange oder beim Warten auf eine Verbindung aus einem Pool. Lastgeneratoren, die vor jeder nächsten Anfrage auf die Antwort warten, senden ausgerechnet dann weniger Anfragen, wenn das System stockt. Das Stocken

ist in den Ergebnissen deshalb kaum zu erkennen. Messen Sie auch vom Client aus und lassen Sie den Server seinen Anteil in einem standardisierten Header melden:

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

Das Gespräch ist zu kleinteilig. Eine Seite, die für jede Zeile einer Liste eine Abfrage ausführt, bezahlt je Zeile einen Roundtrip. Steht die Datenbank in einer anderen Region mit 30 ms pro Roundtrip, kosten hundert Zeilen drei Sekunden, bevor irgendeine Abfrage echte Arbeit erledigt. Eine Abfrage mit einem Join oder ein gebündelter Aufruf bezahlt die Entfernung nur einmal.

Wiederholungsversuche vervielfachen die Last. Ein Client, der nach einer Sekunde aufgibt und es dreimal erneut versucht, vervierfacht den Datenverkehr zu einem langsamen Server und macht ihn damit noch langsamer. Geben Sie jeder Anfrage eine Gesamtfrist, vergrößern Sie die Abstände zwischen Versuchen exponentiell mit einer zufälligen Streuung und begrenzen Sie die Zahl der Wiederholungsversuche.

Weiterführende Lektüre

- RFC 2681: Metrik für die Roundtrip-Verzögerung für IPPM — bitgenau definierte Roundtrip-Verzögerung mit ausdrücklich berücksichtigten verlorenen Antworten und Perzentilen.
- RFC 7679: Metrik für die Einwegverzögerung für IP Performance Metrics — das Gegenstück für eine Richtung, veröffentlicht als Internetstandard.
- Server Timing — der W3C-Header, mit dem ein Server dem Browser mitteilt, wo seine Zeit geblieben ist.
- Histogramme und Summaries — warum sich Quantile einer Summary nicht zusammenführen lassen, die Klassen eines Histogramms dagegen schon.
- RFC 8446: TLS 1.3 — der Aufbau des Handshakes einschließlich des eingesparten Roundtrips.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist Latenz?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/latency/