Engineer

Затримка

очікування між запитом і початком відповіді

Що це таке

Затримка (англійською latency) — це час, за який система починає відповідати: від миті, коли запит залишає клієнта, до повернення першого байта відповіді. Коли мережевий шлях вимірюють туди й назад, кажуть про час обміну, round-trip time (RTT), а у вебпродуктивності те саме називають time to first byte (TTFB). Час відповіді — ширше поняття: він триває до останнього байта.

Затримку легко сплутати з пропускною здатністю та смугою пропускання, але ці три величини відповідають на різні питання. Смуга — скільки бітів за секунду здатен передати канал. Пропускна здатність — скільки роботи система завершує за секунду. Затримка — скільки чекає одна одиниця роботи. Нічний пакетний процес може мати чудову пропускну здатність і жахливу затримку, а чат-сервер — навпаки.

Кожна мілісекунда береться з одного з кількох джерел:

- **поширення**, шлях сигналу через відстань: світло в оптичному волокні долає приблизно 200 км за мілісекунду, тож маршрут завдовжки 1 000 км коштує щонайменше 10 мс туди й назад ще до того, як запрацює будь-яка програма;
- **передавання**, виведення байтів у канал, що відчутно для великих обсягів на повільних лініях;
- **черга**, очікування за іншою роботою в буфері маршрутизатора, пулі потоків, пулі з’єднань чи на блокуванні — єдина складова, що росте з навантаженням;
- **обробка**, сама робота: запит до бази, шаблон, шифрування;
- **обміни протоколу**, розмова до першого корисного байта. Нове HTTPS-з’єднання через TCP спершу потребує рукостискання TCP і рукостискання TLS, а TLS 1.3 скоротив повне рукостискання до одного обміну.

Робоча група IETF з метрик продуктивності IP точно визначила мережевий варіант у RFC 2681: затримка обміну триває від першого надісланого біта до останнього біта отриманої відповіді. Два рішення звідти варто перенести на будь-який сервіс. Відповідь, яка так і не прийшла, вважається невизначеною, неформально — нескінченною, а не тихо випадає з вибірки. А статистики, які описує документ, — це перцентилі, медіана й мінімум. Він також зауважує, що мінімум відображає лише поширення й передавання, а все понад мінімум свідчить про перевантаження. Цей поділ так само працює для API, як і для шляху між маршрутизаторами.

Для окремого вебзапиту curl розкладе очікування на етапи:

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

Кожне число — секунди від початку, тож різниця між сусідніми і є ціною етапу: пошук імені, встановлення з’єднання, рукостискання TLS і, нарешті, власний час роздумів сервера до першого байта.

Застосуйте поняття до рішення

Застосуйте «Що таке затримка?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Моніторинг і сповіщення, які попереджають, поки збій ще малий, і цифри, що показують, як часто так і залишається.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Запустила перший дашборд спостережуваності — перший огляд власних систем компанії

Запустила перший дашборд спостережуваності Зв'язатися з нами →

Чому це важливо

Затримку відчуває користувач; пропускна здатність відображається в рахунку за сервери. Коли сторінка повільна, потрібно вирішити, куди вкласти гроші. П’ять наведених вище складників потребують різних рішень; чотири з них показують, чому купівля обладнання не є універсальним розв’язанням. Відстань зменшують, наближаючи систему до користувачів: кеш, CDN, регіон

поруч із ними. Кількість обмінів скорочують: повторно використовують з'єднання, групують запити, отримують дані паралельно. Обробку прискорюють, виконуючи менше роботи: індекс, кращий план запиту. Додаткова потужність може зменшити очікування в черзі; швидший процесор, більший обсяг пам'яті чи швидше введення-виведення також можуть скоротити час обробки, якщо саме ці ресурси є вузьким місцем. Потужніший сервер не обов'язково буде швидшим: перш ніж обирати обладнання, виміряйте, на що йде час.

Середні значення применшують проблему, бо користувачі стикаються з хвостом розподілу. Сторінка, що паралельно звертається до десяти бекенд-сервісів, чекає на найповільніший. Якщо кожен виклик незалежно від інших повільний в 1% випадків, то $1 - 0{,}99^{10}$, тобто близько 9,6% завантажень сторінки, натрапляють щонайменше на один повільний виклик. Рідкісна затримка в частинах стає звичною в цілому.

Тому серйозні цілі записують у перцентилях, наприклад: 99% запитів отримують відповідь протягом 300 мс за 30 днів. І тому 99-й перцентиль зазвичай і є тим числом, за яким варто стежити. Медіана показує, як сервіс почувається в добрий день, а високі перцентилі — яким він здається клієнтам, що вже збираються піти.

Як це виглядає на практиці

У картографічному застосунку затримки під час рендерингу шарів карти й запитів до даних про місцезнаходження шкодили і користувачам, і надійності бекенду. Робота зосереджувалася на обробці просторових запитів у бекенді, що використовував стару базу даних Microsoft SQL Server. Перехід на PostgreSQL із PostGIS скоротив середній час відповіді з 2,5 секунди до менш ніж 250 мілісекунд, тобто підвищив продуктивність просторових запитів у 10 разів. За цим стояли індекси GiST і SP-GiST на просторових стовпцях, понад 50 мільйонів перенесених записів і конфігурація, налаштована на одночасний доступ; навантаження на процесор бекенду впало на 65%.

Затримка також обмежує пропускну здатність там, де за один рядок змагаються багато запитів. Нотатка про те, як продати останній квиток рівно один раз, оцінює стелю для одного товару як кількість продажів за одне захоплення блокування, поділену на час його утримання. Усе повільне всередині блокування цю стелю знижує: виклик оплати під час утримання

опускає її приблизно до двох продажів на секунду. Навіть захисний механізм тут має ціну. Синхронний standby у PostgreSQL додає затримку до кожного коміту, а отже, подовжує саме той час утримання, на який ділить формула.

А щоб з'ясувати, куди йде час, власний моніторинг на одній платформі записує затримку API у власну базу даних і трасує запити через OpenTelemetry. Тож повільний запит можна простежити крізь фронтенд, API і базу даних, а не вгадувати.

Де виникають проблеми

Середнє ховає хвіст. Середні 100 мс цілком сумісні з тим, що кожен п'ятдесятий запит триває три секунди. Записуйте затримку як гістограму й читайте з неї перцентилі. І стежте за тим, як їх поєднують: 99-ті перцентилі п'яти серверів не можна усереднити до 99-го перцентиля всього парку. Документація Prometheus прямо про це каже: квантилі, які обчислює summary, агрегувати не можна, а кошики гістограми можна спершу підсумувати й лише тоді обчислити квантиль.

Секундомір вмикають не там. Сервер, що міряє від початку роботи обробника, пропускає час, який запит провів у балансувальнику навантаження, у черзі прийому з'єднань чи в очікуванні з'єднання з пулу. Генератори навантаження, які чекають на кожну відповідь, перш ніж надіслати наступний запит, надсилають менше запитів саме тоді, коли система застрягає, і застій майже не видно в результатах. Вимірюйте й на боці клієнта, а сервер нехай повідомляє свою частку в стандартному заголовку:

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

Розмова надто балакуча. Сторінка, що виконує по одному запиту на кожен рядок списку, платить по одному обміну за рядок. Якщо база в іншому регіоні й обмін коштує 30 мс, то сто рядків забирають три секунди ще до того, як будь-який запит зробить справжню роботу. Один запит з join або один пакетний виклик платить за відстань один раз.

Повторні спроби множать навантаження. Клієнт, що здається через секунду й пробує ще тричі, перетворює один повільний сервер на вчетверо більший трафік, і сервер від цього стає ще повільнішим. Дайте кожному запиту загальний дедлайн, збільшуйте паузу між спробами експоненційно й з випадковим розкидом і обмежте кількість повторів.

Що почитати далі

- RFC 2681: A Round-trip Delay Metric for IPPM — затримка обміну, визначена до біта, з явними правилами для втрачених відповідей і перцентилів.
- RFC 7679: A One-Way Delay Metric for IP Performance Metrics — відповідник для затримки в один бік, ухвалений як Internet Standard.
- Server Timing — заголовок W3C, яким сервер повідомляє браузеру, на що пішов час.
- Histograms and summaries — чому квантилі із summary не поєднуються, а кошики гістограми поєднуються.
- RFC 8446: TLS 1.3 — будова рукостискання, зокрема обмін, який воно заощаджує.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке затримка?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/latency/