Engineer

CDN

сервери поблизу читача, що віддають копії

Що це таке

CDN, або мережа доставлення контенту (content delivery network), — це парк серверів, розкиданих по багатьох містах. Вона стоїть між читачами та сервером, на якому сайт працює насправді, — його називають джерелом, або origin, — і роздає копії того, що публікує джерело. Читач у Сіднеї, який відкриває сторінку з джерелом у Торонто, отримує її з машини в Сіднеї чи поблизу, і запит може взагалі не перетинати Тихий океан.

Два механізми спрямовують кожного читача до найближчого сервера. Деякі мережі дають різну відповідь DNS залежно від того, звідки надійшло запитання. Інші оголошують ту саму IP-адресу з усіх точок присутності, а маршрутизація інтернету сама доводить кожен пакет до найближчої. Так чи інакше, сервер, що відповідає, зазвичай завершує TLS від імені сайту: він має сертифікат для домену й читає кожен запит відкритим текстом, перш ніж вирішити, що з ним робити.

Далі працює спільний кеш. Якщо в сервера вже є свіжа копія, він відповідає одразу, і це влучання (hit). Якщо немає, він забирає відповідь у джерела, зберігає копію й відповідає, і це промах (miss). Скільки часу копія лишається свіжою, вирішує джерело в заголовках відповіді, які визначає стандарт кешування HTTP:

```
# Таблиця стилів із хешем у назві: під цією назвою вміст не зміниться.
Cache-Control: public, max-age=31536000, immutable

# Сторінка: браузери щоразу перепитують, CDN тримає її десять хвилин.
Cache-Control: no-cache
CDN-Cache-Control: max-age=600
```

```
# Сторінка облікового запису: жоден кеш не може зберігати копію.
Cache-Control: private, no-store
```

`max-age` задає час життя в секундах. `immutable` каже браузеру не перевіряти копію повторно протягом цього часу. `no-cache` дозволяє кешу зберегти відповідь, але вимагає звернутися до джерела перед повторним використанням, а `private` узагалі не пускає відповідь у спільні кеші. `CDN-Cache-Control` читають лише CDN, тож сайт може дозволити периферії мережі тримати сторінку довше, ніж браузерам.

Та сама інфраструктура сьогодні переносить набагато більше, ніж статичні файли. Більшість CDN також передають джерелу як проксі запити, які не можна кешувати, поглинають потоки ворожого трафіку й запускають невеликі програми на периферії. Тому термін охоплює широкий набір сервісів, що виросли з тієї самої ідеї: віддавати копії поруч із читачем.

Застосуйте поняття до рішення

Застосуйте «Що таке CDN?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Сервери під те навантаження, яке у вас справді є, перенесені в хмару чи назад, із порахованим наперед місячним рахунком.

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. Вела 20 застосунків Docker у Google Cloud і AWS — 20 контейнеризованих застосунків розгорнуто й поставлено під моніторинг

Вела 20 застосунків Docker у Google Cloud і AWS Зв'язатися з нами →

Чому це важливо

Відстань забирає час, якого не поверне жодне налаштування сервера. Світло в оптоволокні долає приблизно 200 кілометрів за мілісекунду, тож дорога туди й назад між Європою та Австралією займає значно більше ніж 100 мілісекунд ще до того, як будь-який сервер щось зробив, а нове захищене з’єднання витрачає одну чи дві такі подорожі, перш ніж буде надіслано перший запит. Копія поблизу прибирає більшу частину цього очікування для кожного читача, далекого від джерела.

Друга причина — навантаження. Коли більшість запитів влучає в кеш, джерело бачить лише малу частку трафіку, а раптовий сплеск падає на мережу, збудовану для сплесків, а не на одну машину. Із `stale-if-error`

периферія може й далі віддавати останню справну копію, поки джерело не працює, тож коротка аварія лишається непомітною для багатьох читачів.

Не всі витрати видно в рахунку. CDN перед сайтом — це ще одна компанія посеред кожного відвідування. Вона бачить IP-адресу кожного читача, кожну URL-адресу, яку той запитує, і, оскільки завершує TLS, сам вміст обміну, а обробляє все це за власною політикою. Це також спільна залежність: коли у великого провайдера погана година, тисячі не пов'язаних між собою сайтів переживають її разом із ним.

Давній аргумент на користь бібліотек і шрифтів із популярного публічного CDN теж ослаб. Браузери тепер тримають окремий кеш для кожного сайту, тож файл, завантажений для одного сайту, для наступного завантажується знову, і спільна копія приносить запит до третьої сторони без жодного виграшу від кешу.

Сайту, чиї читачі здебільшого в одному регіоні й який працює на швидкій машині в тому самому регіоні, CDN може бути зовсім не потрібен. Сайту з аудиторією по всьому світу, важкими медіафайлами чи трафіком, що надходить ривками, він зазвичай потрібен.

Як це виглядає на практиці

На цьому сайті питання CDN вирішили так: спробували й вимкнули. Робота компанії з DNS фіксує, що дві вебадреси вже обслуговувалися через мережу доставлення контенту провайдера, усе працювало, і все ж від неї відмовилися. Із проксі попереду інша компанія обробляла адресу кожного відвідувача й кожну URL-адресу, яку той запитував, раніше за сам сайт і за власною політикою, а не за політикою компанії. Для бізнесу, чия відмітна обіцянка полягає в тому, що ніхто не спостерігає, сторінка оцінює це як гірший обмін, ніж атаки, від яких проксі захищає, а аргументи записано, щоб ніхто випадково не відкрив це питання знову.

Та сама позиція видна в тому, що завантажують сторінки. Сторінка подяк називає п'ять видів файлів, які доходять до браузера читача, і пояснює, чому шрифти надходять із власної машини сайту: вебшрифт із чужого CDN дає третій стороні дивитися, як ви читаєте. Cloudflare відповідає за DNS сайту й більше ні за що, сайт свідомо не пропускається через проксі, і читач, який відкриє панель мережі в браузері, нарахує один хост.

Де CDN справді на своєму місці, нотатки так і кажуть. Нотатка про розпродаж обмеженого товару доводить, що відмова покупцеві має бути найдешевшою дією системи. Коли покупців мільйон, а одиниць десять тисяч, 990 000 запитів мусять отримати відмову. Серед способів позбутися їх якомога раніше нотатка називає кімнату очікування, шлюз, розрахований на кількість товару, обмеження частоти запитів для кожного облікового запису та прапорець `sold_out`, який CDN віддає, ні в кого нічого не перепитуючи. Порада щодо навантажувального тестування підходить до того самого з іншого боку: коли шлях «розпродано» становить 99% трафіку, цьому трафіку місце на периферії.

Де виникають проблеми

Периферія робить рівно те, що кажуть заголовки та її конфігурація, а це не завжди те, що мали на увазі.

- Чужу сторінку показують наступному відвідувачеві. Відповідь з обліковим записом, кошиком чи привітанням потрапляє в кеш, бо ніщо не позначило її як приватну. Встановлення cookie не зупиняє кешування, на що прямо вказує стандарт кешування HTTP, тож наступний читач отримує сторінку першого. Позначайте кожну персональну відповідь `private` або `no-store` і кажіть CDN оминати кеш, коли запит несе сесійну cookie.
- Розгортання залежить від очищення кешу. Новий HTML уже працює, а деякі вузли ще тримають старий файл стилів чи навпаки, і сторінки показуються зламаними, доки не спливе кожна копія. Очищення доходить до всіх точок не одразу й у різних провайдерів працює по-різному. Додавайте хеш вмісту до назви кожного файлу, кешуйте такі файли на рік як `immutable`, а коротке життя лишайте тільки HTML.
- Ключ кешу хибний в один чи в інший бік. Параметри відстеження на кшталт `utm_source` перетворюють кожне поширене посилання на окремий запис у кеші, і частка влучань обвалюється. Якщо ж у ключі бракує заголовка, що змінює відповідь, як-от `Accept-Language`, усі отримують мову одного читача. Відкидайте параметри, які джерело ігнорує, і перелічуйте у `Vary` заголовки, які воно читає.
- Популярний об’єкт застаріває, і джерело заливає запитами. Тисячі читачів одночасно отримують промах, і кожен їхній запит іде далі. Об’єднання запитів у CDN і `stale-while-revalidate`, що віддає стару копію, поки один запит забирає нову, тримають навантаження на джерело рівним.

- Джерело відповідає кожному, хто його знайде. Захист на периферії мало що важить, якщо зловмисник може дістатися адреси джерела напряму, а її часто видають старі записи DNS, заголовки листів чи забутий піддомен. Приймайте трафік на джерелі лише від CDN — за діапазоном мережевих адрес або через автентифіковане з'єднання.

Що почитати далі

- RFC 6707, розділ 1.1: термінологія CDN — визначення CDN, сурогатних серверів і маршрутизації запитів від IETF.
- RFC 9111: HTTP Caching — що можуть зберігати спільні кеші, як довго і що означає `private`.
- RFC 9213: Targeted HTTP Cache Control — поле `CDN-Cache-Control`, адресоване лише CDN.
- RFC 5861: HTTP Cache-Control Extensions for Stale Content — `stale-while-revalidate` і `stale-if-error`.
- RFC 8246: HTTP Immutable Responses — як файл із хешем у назві уникає повторних перевірок.

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 CDN?» 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/cdn/