Engineer

CDN

servere nær læseren, der udleverer kopier

Hvad det er

Et CDN, et content delivery network, er en flåde af servere fordelt over mange byer. Det står mellem læserne og den server, et site faktisk kører på, kaldet origin-serveren, og udleverer kopier af det, origin udgiver. En læser i Sydney, der beder om en side, hvis origin står i Toronto, får den fra en maskine i eller nær Sydney, og forespørgslen behøver aldrig at krydse Stillehavet.

To mekanismer sender hver læser til en server i nærheden. Nogle netværk giver forskellige DNS-svar alt efter, hvor spørgsmålet kommer fra. Andre annoncerer den samme IP-adresse fra alle lokationer og lader internettets routing føre hver pakke til den nærmeste. I begge tilfælde afslutter den server, der svarer, som regel TLS på sitets vegne. Den har altså et certifikat til domænet og læser hver forespørgsel i klartekst, før den beslutter, hvad der skal ske med den.

Resten er en delt caches arbejde. Har serveren allerede en frisk kopi, svarer den med det samme, og det kaldes et hit. Har den ikke, henter den svaret hos origin, gemmer en kopi og svarer, og det kaldes et miss. Hvor længe en kopi regnes for frisk, bestemmer origin i svarheadere, som standarden for HTTP-caching definerer:

```
# Et stylesheet med hash i navnet: ændrer sig aldrig under det navn.
Cache-Control: public, max-age=31536000, immutable

# En side: browsere spørger hver gang, CDN'et gemmer den ti minutter.
Cache-Control: no-cache
CDN-Cache-Control: max-age=600

# En kontoside: ingen cache må gemme en kopi.
Cache-Control: private, no-store
```

`max-age` angiver en levetid i sekunder. `immutable` beder browseren om ikke at genvalidere i den tid. `no-cache` tillader en cache at gemme svaret, men kræver, at den spørger origin, før den genbruger det, mens `private` holder svaret helt ude af delte caches. `CDN-Cache-Control` læses kun af CDN'er, så et site kan lade kanten beholde en side længere end browserne.

Det samme maskineri bærer i dag langt mere end statiske filer. De fleste CDN'er sender også forespørgsler, der ikke kan caches, videre til origin som proxy, opfanger store mængder fjendtlig trafik og kører små programmer ude ved kanten. Derfor dækker ordet en bred vifte af tjenester, der alle udspringer af den samme idé: at servere kopier tæt på læseren.

Brug begrebet i en beslutning

Brug »Hvad er et CDN?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Servere dimensioneret efter det, I rent faktisk bruger, flyttet op i skyen eller hjem igen, med den månedlige regning regnet ud først.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har drevet 20 Docker-apps på Google Cloud og AWS — 20 containeriserede applikationer udrullet og overvåget

Har drevet 20 Docker-apps på Google Cloud og AWS Kontakt os →

Hvorfor det betyder noget

Afstand koster tid, som ingen tuning af serveren kan hente igen. Lys i fiber tilbagelægger omkring 200 kilometer pr. millisekund, så en tur frem og tilbage mellem Europa og Australien tager et godt stykke over 100 millisekunder, før nogen server har udført noget arbejde, og en ny sikker forbindelse bruger en eller to af de ture, før den første forespørgsel overhovedet bliver sendt. En kopi i nærheden fjerner det meste af ventetiden for alle læsere langt fra origin.

Belastning er den anden grund. Når de fleste forespørgsler er hits, ser origin kun en lille del af trafikken, og en pludselig spids rammer et netværk, der er bygget til spidser, frem for én maskine. Med `stale-if-error` kan kanten blive ved med at levere den sidste gode kopi, mens origin er nede, så et kort nedbrud forbliver usynligt for mange læsere.

Prisen står ikke på fakturaen. Et CDN foran et site er én virksomhed mere midt i hvert eneste besøg. Det ser hver læsers IP-adresse, hver URL, der bliver bedt om, og, fordi det afslutter TLS, selve indholdet af udvekslingen, og det håndterer det

hele under sin egen politik. Det er også en fælles afhængighed: har en stor udbyder problemer en times tid, har tusindvis af sites, der ellers intet har med hinanden at gøre, de samme problemer.

Det gamle argument for at hente biblioteker og fonte fra et populært offentligt CDN er også blevet svagere. Browsere holder nu en separat cache for hvert site, så en fil, der er hentet til ét site, hentes igen til det næste, og den delte kopi giver en forespørgsel til en tredjepart uden nogen cachegevinst at vise for det.

Et site, hvis læsere mest befinder sig i én region, og som betjenes fra en hurtig maskine i samme region, har måske slet ikke brug for et CDN. Et site med et publikum over hele verden, tunge medier eller trafik, der kommer i ryk, har det som regel.

Sådan ser det ud i praksis

På dette site blev CDN-spørgsmålet afgjort ved at prøve et og slå det fra igen. Virksomhedens DNS-arbejde fortæller, at de to webflader blev sendt gennem udbyderens indholdsnetværk, at det virkede, og at det alligevel blev slået fra igen. Med proxyen på plads håndterede en anden virksomhed hver besøgendes adresse og hver URL, de bad om, før sitet selv gjorde, og under den virksomheds politik frem for sitets egen. For en forretning, hvis kendetegn er, at ingen kigger med, vurderer siden det som en dårligere handel end de angreb, proxyen beskytter imod, og begrundelsen står skrevet ned, så ingen tager spørgsmålet op igen ved et uheld.

Samme holdning ses i det, siderne henter. Kolofonen nævner de fem slags filer, der når en læsers browser, og forklarer, hvorfor skrifterne kommer fra sitets egen maskine: en webfont hentet fra en andens CDN lader en tredjepart se dig læse. Cloudflare besvarer sitets DNS og gør ikke mere end det, sitet er bevidst ikke proxyet, og den læser, der åbner browserens netværksfane, tæller én vært.

Hvor et CDN gør gavn, siger noterne det. Noten om lynudsalg argumenterer for, at det at afvise en køber skal være det billigste, systemet gør. Med en million købere og ti tusind enheder skal 990.000 forespørgsler afvises. Blandt de måder, noten nævner til at skille sig af med dem tidligt, er et venteværelse, en port dimensioneret efter lageret, en hastighedsgrænse pr. konto og et `sold_out` -flag, som CDN'et udleverer uden at spørge noget bag sig. Rådet om belastningstest siger det samme fra den anden side: når udsolgt-stien er 99 % af trafikken, hører den trafik hjemme ved kanten.

Hvor det går galt

Kanten gør præcis, hvad headerne og dens konfiguration siger, og det er ikke altid det, der var ment.

- En andens side bliver vist for den næste besøgende. Et svar, der viser en konto, en kurv eller en hilsen, bliver cachet, fordi intet markerede det som privat. At sætte en cookie stopper ikke caching, som standarden for HTTP-caching gør opmærksom på, så den næste læser får den førstes side. Markér alle personlige svar med `private` eller `no-store`, og bed CDN'et gå uden om cachen, når der følger en sessionscookie med.
- En udrulning afhænger af en purge. Ny HTML går i luften, mens nogle kanter stadig har det gamle stylesheet, eller omvendt, og siderne vises i stykker, indtil hver kopi udløber. En purge tager tid om at nå alle lokationer og fungerer forskelligt fra udbyder til udbyder. Sæt en hash af indholdet ind i hver fils navn, cache de filer et år som `immutable`, og hold kun HTML'en kortlivet.
- Cachenøglen er forkert i den ene eller den anden retning. Sporingsparametre som `utm_source` gør hvert delt link til en separat post i cachen, og hitraten styrtdykker. Udelades en header, der ændrer svaret, for eksempel `Accept-Language`, får alle én læsers sprog. Fjern parametre, som origin ignorerer, og nævn i `Vary` de headere, den læser.
- Et populært objekt udløber, og origin bliver oversvømmet. Tusindvis af læsere rammer et miss i samme øjeblik, og hver eneste af deres forespørgsler går igennem. Sammenlægning af forespørgsler i CDN'et og `stale-while-revalidate`, der leverer den gamle kopi, mens én forespørgsel henter den nye, holder belastningen på origin jævn.
- Origin svarer alle, der finder den. Beskyttelse ved kanten betyder ikke meget, hvis angribere kan nå origin-adressen direkte, og gamle DNS-poster, mailheadere eller et glemt underdomæne afslører den ofte. Tag kun imod trafik på origin fra CDN'et, enten efter netværksområde eller over en godkendt forbindelse.

Læs mere

- RFC 6707, afsnit 1.1: CDN-terminologi — IETF's definitioner af et CDN, dets surrogater og routingen af forespørgsler.
- RFC 9111: HTTP Caching — hvad delte caches må gemme, hvor længe, og hvad `private` betyder.
- RFC 9213: Targeted HTTP Cache Control — feltet `CDN-Cache-Control`, som kun taler til CDN'er.

- RFC 5861: HTTP Cache-Control Extensions for Stale Content — `stale-while-revalidate` og `stale-if-error` .
- RFC 8246: HTTP Immutable Responses — hvordan en fil med hash i navnet slipper for at blive kontrolleret igen.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er et CDN?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/cdn/