Engineer

CDN

servere nær leseren som deler ut kopier

Hva det er

Et CDN, kort for *content delivery network* og på norsk et innholdsleveringsnettverk eller bare et innholdsnettverk, er en flåte av servere spredt over mange byer. Den står mellom leserne og serveren et nettsted egentlig kjører på, opphavsserveren (*origin*), og deler ut kopier av det opphavsserveren publiserer. En leser i Sydney som ber om en side med opphavsserver i Toronto, får den fra en maskin i eller nær Sydney, og forespørselen trenger kanskje aldri å krysse Stillehavet.

To mekanismer styrer hver leser til en server i nærheten. Noen nettverk gir et annet DNS-svar avhengig av hvor spørsmålet kommer fra. Andre annonserer den samme IP-adressen fra hvert sted og lar rutingen på internett bære hver pakke til det nærmeste. Uansett avslutter serveren som svarer, som regel TLS på vegne av nettstedet. Det betyr at den har et sertifikat for domenet og leser hver forespørsel i klartekst før den bestemmer seg for hva den skal gjøre med den.

Det den gjør etterpå, er jobben til en delt cache. Har den allerede en fersk kopi, svarer den med en gang, og det er et treff. Har den ikke det, henter den svaret fra opphavsserveren, beholder en kopi og svarer, og det er en bom. Hvor lenge en kopi regnes som fersk, bestemmer opphavsserveren i svarheadere som standarden for HTTP-caching definerer:

```
# A fingerprinted stylesheet: never changes under this name.
Cache-Control: public, max-age=31536000, immutable

# A page: browsers check back each time, the CDN keeps it ten minutes.
Cache-Control: no-cache
CDN-Cache-Control: max-age=600
```

```
# An account page: no cache may keep a copy.
Cache-Control: private, no-store
```

`max-age` gir en levetid i sekunder. `immutable` ber nettleseren la være å revalidere i den tiden. `no-cache` lar en cache lagre et svar, men krever at den sjekker med opphavsserveren før svaret brukes igjen, mens `private` holder det helt utenfor delte cacher. `CDN-Cache-Control` leses bare av CDN-er, så et nettsted kan la kanten holde en side lenger enn nettleserne gjør.

Det samme maskineriet bærer i dag mye mer enn statiske filer. De fleste CDN-er videresender også forespørsler som ikke kan caches, til opphavsserveren som en proxy, tar imot flommer av fiendtlig trafikk og kjører små programmer ute på kanten. Derfor dekker begrepet et bredt spekter av tjenester som alle starter fra den samme ideen: å servere kopier nær leseren.

Bruk begrepet i en beslutning

Bruk «Hva er CDN?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Servere dimensjonert etter det du faktisk bruker, flyttet inn i skyen eller ut av den igjen, med månedsregningen regnet ut på forhånd.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Driftet 20 Docker-apper på Google Cloud og AWS — 20 containeriserte applikasjoner produksjonssatt og overvåket

Driftet 20 Docker-apper på Google Cloud og AWS Ta kontakt →

Hvorfor det er viktig

Avstand koster tid som ingen mengde finjustering av serveren kan vinne tilbake. Lys i fiber går omtrent to hundre kilometer per millisekund, så en tur fram og tilbake mellom Europa og Australia tar godt over hundre millisekunder før noen server har gjort noe arbeid. En ny sikker forbindelse bruker én eller to slike runder før den første forespørselen blir sendt. Å servere fra en kopi i nærheten fjerner det meste av den ventetiden for hver leser som sitter langt fra opphavsserveren.

Last er den andre grunnen. Når de fleste forespørslene er treff, ser opphavsserveren bare en liten del av trafikken, og en plutselig topp lander på et nettverk bygd for topper og ikke på én maskin. Med `stale-if-error` kan kanten fortsette å servere den siste gode kopien mens opphavsserveren er nede, så et kort avbrudd forblir usynlig for mange lesere.

Kostnaden er mindre åpenbar enn fakturaen. Et CDN foran et nettsted er enda et selskap midt i hvert besøk. Det ser hver lesers IP-adresse og hver URL de ber om, og siden det avslutter TLS, ser det også innholdet i utvekslingen, og det håndterer alt sammen etter sine egne retningslinjer. Det er også en delt avhengighet: når en stor leverandør har en dårlig time, har tusenvis av nettsteder som ellers ikke har noe med hverandre å gjøre, den samtidig.

Det gamle argumentet for å laste biblioteker og fonter fra et populært offentlig CDN har også svekket seg. Nettlesere holder nå en egen cache for hvert nettsted, så en fil som er hentet for ett nettsted, hentes på nytt for det neste. Den delte kopien gir en forespørsel til en tredjepart uten noen cachegevinst å vise til.

Et nettsted der leserne stort sett sitter i én region, og som serveres fra en rask maskin i samme region, trenger kanskje rett og slett ikke et CDN. Et nettsted med et publikum over hele verden, tunge medier eller trafikk som kommer i støt, trenger som regel et.

Slik ser det ut i praksis

På dette nettstedet ble CDN-spørsmålet avgjort ved å prøve et og slå det av. Selskapets DNS-arbeid dokumenterer at proxying av de to webflatene gjennom leverandørens innholdsnettverk ble bygget, virket og ble så slått av igjen. Med proxyen på plass håndterte et annet selskap hver besøkendes adresse og hver URL de ba om, før nettstedet selv gjorde det, og etter det selskapets retningslinjer og ikke etter dette selskapets egne. For en bedrift som skiller seg ut ved å hevde at ingen ser på, vurderer siden det som en dårligere byttehandel enn angrepene proxyen beskytter mot. Begrunnelsen er skrevet ned, så ingen åpner spørsmålet igjen ved et uhell.

Det samme standpunktet viser seg i hva sidene laster. Kolofonen navngir de fem typene filer som når fram til leserens nettleser, og forklarer hvorfor skrifttypene kommer fra nettstedets egen maskin: en webfont fra noen andres CDN er en tredjepart som ser deg lese. Cloudflare svarer på DNS for nettstedet og gjør ingenting mer, nettstedet går med vilje ikke gjennom deres proxy, og en leser som åpner nettverkspanelet i nettleseren, teller én vert.

Der et CDN gjør seg fortjent til plassen, sier notatene fra. Notatet om lynsalg argumenterer for at å avvise en kjøper må være det billigste systemet gjør. Med en million kjøpere og ti tusen enheter må 990 000 forespørsler mislykkes. Blant

måtene notatet nevner for å kvitte seg med dem tidlig, er et venterom, en port som slipper inn omtrent like mange kjøpere som det finnes enheter, en hastighetsbegrensning per konto og et `sold_out` -flagg som CDN-et serverer uten å spørre noe bak seg. Rådet om lasttesting sier det samme fra den andre siden: når utsolgt-stien er 99 % av trafikken, hører den trafikken hjemme på kanten.

Hvor det går galt

Kanten gjør nøyaktig det headerne og konfigurasjonen sier, og det er ikke alltid det som var ment.

- Neste besøkende får servert en annen persons side. Et svar som viser en konto, en handlekurv eller en hilsen, blir cachet fordi ingenting merket det som privat. Å sette en informasjonskapsel stopper ikke caching, som standarden for HTTP-caching påpeker, så neste leser får den første leserens side. Merk hvert svar som gjelder én bruker, med `private` eller `no-store` , og be CDN-et gå utenom cachen når en øktcookie er til stede.
- En produksjonssetting er avhengig av en tømming. Ny HTML går live mens noen kanter fortsatt holder det gamle stilarket, eller omvendt, og sidene vises ødelagt til hver kopi har gått ut. Tømminger bruker tid på å nå hvert sted og varierer fra leverandør til leverandør. Legg en hash av innholdet i filnavnet til hver ressurs, cache de filene i et år som `immutable` , og gi bare HTML-en kort levetid.
- Cachenøkkelen bommer i den ene eller den andre retningen. Sporingsparametere som `utm_source` gjør hver delte lenke til en egen oppføring i cachen, og treffraten stuper. Utelater nøkkelen en header som endrer svaret, som `Accept-Language` , får alle servert én lesers språk. Fjern parametere opphavsserveren overser, og navngi headerne den leser, i `Vary` .
- Et populært objekt går ut, og opphavsserveren oversvømmes. Tusenvis av lesere bommer i samme øyeblikk, og hver eneste av de forespørslene slipper gjennom. Sammenslåing av forespørsler i CDN-et og `stale-while-revalidate` , som serverer den gamle kopien mens én enkelt forespørsel henter den nye, holder lasten på opphavsserveren jevn.
- Opphavsserveren svarer alle som finner den. Beskyttelse ute på kanten betyr lite hvis angripere kan nå adressen til opphavsserveren direkte, og gamle DNS-poster, e-posthoder eller et glemt underdomene røper den ofte. Ta bare imot trafikk på opphavsserveren fra CDN-et, etter nettverksområde eller gjennom en autentisert forbindelse.

Les mer

- RFC 6707, section 1.1: CDN terminology — IETFs definisjoner av et CDN, stedfortrederserverne (*surrogates*) og ruting av forespørsler.
- RFC 9111: HTTP Caching — hva delte cacher kan lagre, hvor lenge, og hva `private` betyr.
- RFC 9213: Targeted HTTP Cache Control — feltet `CDN-Cache-Control`, som bare henvender seg 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 fingeravtrykk i navnet slipper å bli sjekket på nytt.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er CDN?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/cdn/