Engineer

CDN

読者の近くでコピーを配信するサーバー群

どのようなものか

CDNはcontent delivery network、コンテンツ配信ネットワークの略で、多数の都市へ分散したサーバー群です。読者と、サイトが実際に動くオリジンと呼ぶサーバーの間に入り、オリジンが公開するもののコピーを配信します。シドニーの読者が、トロントをオリジンとするページを求めると、シドニーかその近くのマシンから取得し、要求が太平洋をまったく渡らないこともあります。

近くのサーバーへ各読者を案内する仕組みは、二つあります。問い合わせ元の場所に応じて、異なるDNSの答えを返すネットワークもあれば、すべての拠点から同じIPアドレスへの経路を広告し、インターネットの経路制御で、各パケットを最も近い拠点へ送るものもあります。どちらの場合も、応答するサーバーは通常、サイトに代わってTLSを終端します。つまり、そのドメインの証明書を持ち、各要求を平文で読んでから、どう処理するかを決めます。

次に行うのは、共有キャッシュの仕事です。新鮮なコピーをすでに持っていれば、直ちに応答します。これがヒットです。なければ、オリジンから応答を取得し、コピーを保持してから答えます。これがミスです。コピーがいつまで新鮮かを決めるのは、オリジンです。HTTPキャッシュ標準が定める応答ヘッダーに記述します。

```
# 内容のハッシュを名前に含むスタイルシート：この名前の内容は変わらない。
Cache-Control: public, max-age=31536000, immutable

# ページ：ブラウザーは毎回再確認し、CDNは10分間保持する。
Cache-Control: no-cache
CDN-Cache-Control: max-age=600

# アカウントのページ：どのキャッシュにもコピーを保存させない。
Cache-Control: private, no-store
```

`max-age` は、有効期間を秒で示します。 `immutable` は、その間は再検証しないようブラウザーへ伝えます。 `no-cache` は、応答の保存を認めますが、再利用前にオリジンへ確認させます。 `private` は、共有キャッシュへの保存自体を認めません。 `CDN-Cache-Control` を読むのはCDNだけなので、ブラウザーより長く、エッジでページを保持できます。

同じ仕組みは、今では静的ファイル以外も広く扱います。ほとんどのCDNは、キャッシュできない要求をオリジンへ中継し、悪意ある大量のトラフィックを吸収し、エッジで小さなプログラムも実行します。そのため、この語は、読者の近くでコピーを提供するという同じ考えから始まった、幅広いサービスを指します。

概念を判断に生かす

「CDNとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 実際の使用量に合わせてサーバーを構成し、月額費用を事前に把握したうえで、クラウドへの移行もクラウドからの移転も進めます。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 Google Cloud・AWSでDockerアプリ20個を運用 — 20 デプロイし監視したコンテナアプリケーション

Google Cloud・AWSでDockerアプリ20個を運用 お問い合わせ →

なぜ重要なのか

距離が生む待ち時間は、サーバーをいくら調整しても短縮できません。光ファイバー内の光は、一ミリ秒に約二百キロメートル進むため、ヨーロッパとオーストラリアの往復には、どのサーバーもまだ何も処理していなくても、百ミリ秒を大きく超える時間がかかります。新しい安全な接続では、最初の要求を送る前に、その往復を一回か二回行います。近くのコピーから配信すれば、オリジンから遠いすべての読者について、その待ち時間の大半を取り除けます。

二つ目の理由は負荷です。大半の要求がヒットすれば、オリジンに届くのはトラフィックの小さな割合だけです。急増も、一台のマシンではなく、急増に対応するよう構築したネットワークが受け止めます。 `stale-if-error` を使えば、オリジンの停止中も、エッジで最後の正常なコピーを提供し続けられるため、短い停止は、多くの読者からは見えません。

代償は、請求書ほど分かりやすくはありません。サイトの前段にCDNを置くと、訪問のたびに別の会社が間に入ります。その会社には、読者のIPアドレス、要求するすべてのURL、さらにTLSを終端するため、通信の内容も見えます。すべて

を、その会社自身の方針で扱います。また、共通の依存先にもなります。大きな事業者で一時間ほど問題が起きると、無関係な何千ものサイトが、一緒に影響を受けます。

広く使われる公開CDNからライブラリーやフォントを読み込むという、以前の根拠も弱くなりました。現在のブラウザーは、サイトごとに別のキャッシュを持つため、一つのサイトで取得したファイルも、次のサイトで取得し直します。その共有コピーは、キャッシュ上の利点を伴わない、第三者への要求を増やします。

読者の大半が一つの地域にいて、その地域の高速なマシンから配信するサイトなら、そもそもCDNが不要かもしれません。読者が世界中にいる場合や、容量の大きいメディアを扱う場合、トラフィックが突発的に増える場合は、通常、CDNが必要です。

実際の活用例

このサイトでは、CDNを試してから無効にする、という結論を出しました。自社のDNSの作業記録は、二つのウェブ公開先を、事業者のコンテンツネットワークでプロキシする構成を構築し、動作した後、再び無効にしたことを記しています。プロキシがあると、すべての訪問者のアドレスと要求URLを、サイトより先に別の会社が、その会社の方針で扱い、このサイトの方針では扱えません。誰にも見られず読めることを特徴とする事業にとって、そのページは、プロキシが防ぐ攻撃よりも、その代償のほうが大きいと判断しています。誰かが偶然、同じ問いを持ち出さないよう、理由も書き残しています。

ページが読み込むものにも、同じ立場が表れます。クレジットのページは、読者のブラウザーへ届く五種類のファイルを挙げ、書体をサイト自身のマシンから配信する理由を説明します。他者のCDNからのウェブフォントは、第三者が読書を見ていることになります。Cloudflareが行うのは、サイトのDNSへの応答だけです。サイトの配信には意図的にプロキシを使わず、ブラウザーのネットワークパネルを開けば、読者は一つのホストだけを数えられます。

CDNが役立つ場面では、ノートもそう述べています。短時間の限定販売についてのノートは、買い手を断る処理を、システム内で最も負担の小さいものにする必要があると論じます。百万人の買い手と一万個の在庫なら、990,000件の要求を断る必要があります。早い段階で負荷を落とす方法として、待合室、在庫数に合わせた入場制限、アカウントごとのレート制限、背後の何にも問い合わせずCDNが返す `sold_out` フラグを挙げます。負荷テストへの助言も、反対側から同じ点を示します。売り切れ時の処理がトラフィックの99%なら、そのトラフィックは、エッジで処理すべきです。

問題が起きるところ

エッジは、ヘッダーと設定に書かれたとおりに動きます。それが、意図したことと一致するとは限りません。

- 他人のページを、次の訪問者へ提供します。アカウント、買い物かご、挨拶を示す応答が、プライベートと指定されていないため、キャッシュされます。HTTPキャッシュ標準が指摘するとおり、Cookieを設定してもキャッシュは止まらないため、次の読者に、最初の読者のページが届きます。利用者別のすべての応答に `private` か `no-store` を付け、セッションCookieがあるときはキャッシュを迂回するよう、CDNにも指示します。
- デプロイが、パージに依存します。新しいHTMLを公開しても、一部のエッジが古いスタイルシートを保持するか、その逆が起き、すべてのコピーの期限が切れるまで、ページの表示が壊れます。パージが全拠点へ届くには時間がかかり、事業者によっても異なります。各素材のファイル名に内容のハッシュを入れ、 `immutable` で一年間キャッシュし、HTMLだけを短い有効期間にします。
- キャッシュキーが、過剰にも不足にもなります。 `utm_source` などの追跡パラメーターの違いで、共有するリンクごとに別のエントリーができ、ヒット率が崩れます。 `Accept-Language` のように応答を変えるヘッダーを省くと、一人の言語を全員へ返します。オリジンが無視するパラメーターは取り除き、読むヘッダーは `Vary` に指定します。
- 人気のあるデータの期限が切れ、オリジンへ要求が殺到します。何千人もの読者の要求が同時にキャッシュミスとなり、そのすべてがオリジンへ届きます。CDNで要求を一つにまとめ、新しいコピーを一つの要求で取得する間は古いコピーを返す `stale-while-revalidate` を使えば、オリジンの負荷を一定に保てます。
- アドレスを見つけた誰にでも、オリジンが応答します。攻撃者が直接アクセスできれば、エッジでの防御の意味は小さくなります。古いDNSレコード、メールヘッダー、忘れられたサブドメインが、アドレスを明かすこともよくあります。ネットワークアドレスの範囲か、認証済みの接続によって、CDNからのトラフィックだけをオリジンで受け付けます。

参考資料

- RFC 6707、第1.1節：CDNの用語：CDN、代理配信サーバー、要求の振り分けに関するIETFの定義です。
- RFC 9111：HTTP Caching：共有キャッシュに保存できるもの、その期間、 `private` の意味を説明します。
- RFC 9213：Targeted HTTP Cache Control：CDNだけに伝える `CDN-Cache-Control` フィールドを説明します。

- RFC 5861：HTTP Cache-Control Extensions for Stale Content：`stale-while-revalidate` と `stale-if-error` を説明します。
- RFC 8246：HTTP Immutable Responses：内容のフィンガープリントを名前に持つファイルで、再検証を避ける仕組みを説明します。

用語集

実際の制約に合わせて選ぶ

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「CDNとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/cdn/