

Projeto CDN

1st Ricardo Sá
DCC
FCUP
Porto, Portugal
up202408644@up.pt

2nd Xavier Santos
DCC
FCUP
Porto, Portugal
up202108894@up.pt

3rd Rui Santos
DCC
FCUP
Porto, Portugal
up202109728@up.pt

Abstract—A evolução da Internet tem ocorrido maioritariamente de forma descentralizada. Esta descentralização foi, e é, uma característica fundamental para o seu crescimento e evolução de forma “orgânica”. Todavia, a inerente falta de gestão tornou difícil lidar sistematicamente com problemas relacionados com desempenho; paralelamente, o aumento exponencial de utilizadores levou à sobrecarga da largura de banda disponível na rede e à diminuição da capacidade dos servidores em responder a pedidos, o que afeta a qualidade dos serviços prestados.

Uma Rede de Distribuição de Conteúdos (CDN) pode ser definida, de maneira simples, como um conjunto de servidores interligados, cujo principal objetivo é mitigar os efeitos da latência e a da perda de pacotes, recorrendo à aproximação do conteúdo dos utilizadores finais.

Este artigo explora a viabilidade da implementação de uma CDN baseada em DNS. Abordamos os princípios fundamentais das CDNs, os tipos existentes, e os objetivos deste projeto, com foco na utilização do DNS para otimizar a distribuição de conteúdo.

Foi desenvolvido um protótipo de CDN que utiliza DNS para direcionar o tráfego, testado em ambientes virtualizados e na cloud.

I. OVERVIEW

Uma rede de distribuição de conteúdos (CDN) é um grupo de servidores distribuídos geograficamente que armazena o conteúdo em *cache* junto dos utilizadores finais. Uma CDN permite a transferência rápida de ativos necessários para carregar conteúdos na Internet (e.g. páginas HTML, ficheiros JavaScript, folhas de estilo, imagens e vídeos).

Para além de melhorar a performance no acesso a recursos na Web, uma CDN pode também providenciar certos aspetos de segurança, nomeadamente contra ataques DoS.

A. Como funciona uma CDN?

Uma CDN é constituída por três componentes principais:

- **Servidores origem:** Os servidores de origem (SO) contêm as versões originais do conteúdo e funcionam como fonte principal de informação. As alterações/adições de conteúdo são realizadas nestes servidores. Um SO pode ser gerido pelo próprio fornecedor do conteúdo, ou pode ser hospedado na infraestrutura de terceiros (e.g. AWS S3 da Amazon, Google Cloud Storage).
- **Servidores réplica:** Os servidores réplica (SR) dispersos pelo mundo, armazenam conteúdo transferido do(s) SO e são responsáveis por entregar esse conteúdo aos

utilizadores mais próximos. Quando um utilizador quer aceder a um conteúdo, o mecanismo de distribuição redireciona-o para um SR geograficamente mais próximo dele. Se a réplica não tem o conteúdo em *cache* (ou a mesma estiver desatualizada), o pedido é reencaminhado para o SO e o conteúdo da *cache* é atualizado.

- **Mecanismo de distribuição:** o mecanismo de distribuição (MD) é responsável por redirecionar pedidos (destinados a CDN) para o SR que se encontre mais próximo da sua origem. Para além da proximidade, este mecanismo pode ter em conta outros aspetos, como por exemplo quando está a decidir para qual réplica é que vai ser reencaminhado o pedido (e.g. carga, estado da rede, etc...).

Para demonstrar o funcionamento de uma CDN, suponhamos a seguinte situação: um cliente localizado em Nova Iorque pretende aceder ao *website* de uma loja em Londres, que é hospedado num servidor algures no Reino Unido (ver Fig. 1). Sem CDN, o pedido do cliente teria um atraso relativamente elevado, visto que o pedido teria que atravessar o Oceano Atlântico. Se o servidor no Reino Unido utilizar uma CDN e criar réplicas em diversas localizações, o atraso associado aos pedidos reduz significativamente, especialmente se existir uma réplica geograficamente mais próxima do nosso cliente. As réplicas criadas contêm uma *cache* em que guardam o conteúdo do servidor localizado em Londres e são responsáveis por entregá-lo o mais próximo possível da localização do cliente (neste caso, em Nova Iorque).

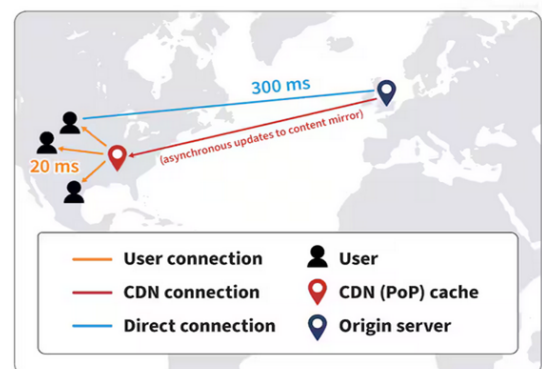


Fig. 1: Rede CDN

B. Funções essenciais de uma CDN

Para melhorar a distribuição de conteúdo, as CDNs executam três funções essenciais:

- **Redução de latência:** A latência (ou *delay*) é o atraso com que o cliente se depara ao tentar aceder a uma página *web* ou um vídeo em *streaming* antes de estes serem totalmente carregados no dispositivo (i.e. o atraso desde da origem até ao destino). Tipicamente as CDNs aliviam a latência reduzindo a distância física que o conteúdo precisa de percorrer para chegar até ao cliente, criando réplicas mais próximas dos mesmos.
- **Fiabilidade e redundância:** Estes permitem que um serviço se mantenha ativo, mesmo na presença de falhas de *hardware/software*. No caso das CDNs, o conteúdo é tipicamente distribuído por diversos **SR** e, mesmo que um deles falhe, existem outros que garantem redundância suficiente para manter o funcionamento do sistema.
- **Segurança dos dados:** A segurança dos dados é um aspeto fundamental em qualquer serviço *web*. No caso das CDNs podemos utilizar certificados TLS/SSL para criar um canal de comunicação seguro entre os nós que fazem parte de uma CDN.

1) Latência – Como é que uma CDN diminui latência?

Os serviços providenciados por uma CDN permitem reduzir a latência da seguinte forma:

- A distribuição geográfica associada a uma CDN permite reduzir significativamente a distância entre utilizadores e os recursos fornecidos por um *website*. Em vez de um utilizador contactar um **SO**, que pode encontrar-se no ponto oposto do globo, contacta a réplica geograficamente mais próxima (utilizando o **MD** fornecido pela CDN). Desta forma obtém-se uma menor latência e o serviço prestado é de maior qualidade.
- As otimizações de *hardware* e *software*, como o controlo de carga eficiente e os SSD, podem ajudar os dados a chegar ao utilizador rapidamente.
- As CDNs podem utilizar técnicas como compressão para diminuir o tamanho dos ficheiros e a latência no acesso aos mesmos.
- As CDNs podem acelerar os *websites* que utilizam certificados TLS/SSL, de forma a otimizar a reutilização de ligações e a ativação de um início falso do TLS.

2) Fiabilidade e redundância – Como é que uma CDN mantém um website sempre online?

A presença *online* ‘constante’ é um componente crítico para qualquer *website* na Internet. As falhas de *hardware* e os picos de tráfego, que advêm de ataques maliciosos ou apenas de um aumento tráfego, têm a capacidade de comprometer o estado do servidor *web* e impedir os utilizadores de aceder a um *website* ou serviço. Uma CDN completa possui várias características que minimizam o tempo de inatividade:

- O controlo de carga distribui o tráfego de rede uniformemente por vários servidores, o que facilita a gestão de aumentos abruptos de tráfego.
- O *failover* inteligente oferece um serviço contínuo mesmo que um ou mais servidores CDN fiquem *offline* devido a falhas de *hardware*. Este pode ainda redistribuir o tráfego para outros servidores que estejam operacionais.

3) Segurança dos Dados – Como é que uma CDN protege os dados?

A segurança da informação é parte integrante de uma CDN, que pode manter um *website* protegido com novos certificados TLS/SSL que irão garantir um elevado padrão de autenticação, encriptação e integridade.

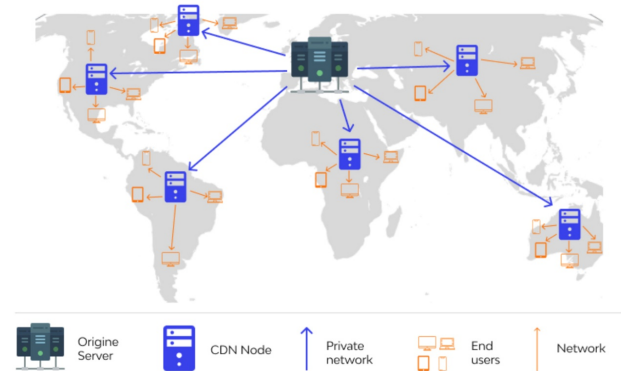


Fig. 2: Segurança da Rede CDN

C. Quais são os benefícios de utilizar uma CDN?

Apesar dos benefícios associados a uma CDN variarem conforme a implementação da mesma, estes podem ser descritos em quatro componentes principais:

- **Menor latência:** a distribuição de conteúdo para a réplica mais próxima permite reduzir significativamente a latência no acesso ao conteúdo *web*.
- **Disponibilidade e redundância de conteúdo:** Devido à sua arquitetura distribuída, as CDN são capazes de absorver grandes volumes de tráfego e garantir a continuidade do serviço em caso de falhas de *hardware/software*.
- **Segurança:** Uma rede CDN pode melhorar a segurança, fornecendo mitigação de ataques DoS, melhorias nos certificados de segurança, e outras otimizações.

II. TIPOS DE CDN

Embora todo o tipo de CDN partilhe funcionalidades e componentes, podemos distingui-las pelo **mecanismo de distribuição** que utilizam.

De seguida temos uma breve descrição sobre alguns tipos de CDN:

- **Multiplexação do Cliente:** Neste mecanismo, o cliente (ou um *proxy server* perto do cliente) recebe o endereço de um conjunto de **SR** e escolhe um para enviar um

pedido. Geralmente, este esquema impõe *overhead* adicional no envio de um conjunto de candidatos de **SR** para o cliente quando é pedido o conteúdo. Além disso, devido à pouca informação geral, o cliente pode escolher um servidor com carga elevada, o que pode levar a sobrecarga e, conseqüentemente, um aumento da latência.

- **Redirecionamento de pedidos HTTP:** Este é o tipo de CDN mais simples. Neste esquema, os pedidos vão diretamente para o servidor de origem, que faz um HTTP Redirect para um novo URL (i.e. uma réplica). Como o **SO** é o único ponto responsável por redirecionar pedidos, isto pode levar a problemas com SPOF e *throttling*.
- **DNS:** Este esquema utiliza o serviço de resolução de nomes providenciado por DNS para encaminhar os pedidos de conteúdo do cliente para a réplica mais adequada (tipicamente tendo em conta parâmetros como localização geográfica e *load balancing* de pedidos).
- **Anycasting:** Esta técnica utiliza um endereço IP ou URL *anycast* para identificar um grupo de servidores que fornecem o mesmo serviço. Quando um cliente comunica com um desses servidores, os pacotes são enviados com o endereço *anycast* no campo de destino. Os *routers* encaminham estes pacotes para o servidor mais próximo ou mais adequado dentro do grupo, com base em métricas de encaminhamento. Esta capacidade de encaminhamento *anycast* pode ser integrada na infraestrutura de encaminhamento da Internet existente, permitindo a distribuição eficiente de pedidos para CDNs. Além disso, a arquitetura *anycast* oferece alta escalabilidade, adaptando-se ao crescimento do tráfego da Internet.
- **Encaminhamento Peer-to-Peer (P2P):** Sistemas P2P estão a ser amplamente implementados na Internet para a distribuição de conteúdos. Os nós participantes num sistema P2P geralmente pertencem a diferentes entidades e eles próprios constituem uma rede *ad-hoc*. Como esta está em constante mudança, nenhum nó tem a informação global completa. O principal problema desta solução, que hoje em dia ainda é investigado, consiste em arranjar uma forma distribuída e eficiente de encaminhar pedidos sem que haja sobrecarga de propagação de informação de encaminhamento.

III. PROJETO CDN

A. Objetivos do Projeto

- Implementar uma Rede CDN simples, constituída por um **SO** e pelo menos dois **SR**. O **MD** que vamos utilizar é o DNS (Ver Figura 3).
- Configurar o servidor DNS para reencaminhar o tráfego para o SR mais próximo do cliente.
- Assegurar que os nós de *cache* armazenem corretamente os conteúdos estáticos (imagens, CSS, JS, etc.).
- Realizar testes de desempenho, simulando diferentes cenários de latência da rede, incluindo conexões virtuais com atrasos controlados para simular distância geográfica.

- Comparar o desempenho com e sem CDN (tempo de carregamento, latência, etc.).
- Observar a forma como a CDN lida com múltiplos pedidos em simultâneo.
- Fazer uma demonstração prática da resolução DNS e distribuição de conteúdo via CDN.
- Repetir o processo num ambiente realista, fazendo uso dos recursos da cloud.

B. Software de simulação

O GNS3 (Graphical Network Simulator-3) permite criar redes complexas com recurso a uma interface gráfica, com a possibilidade de integrar *routers*, *switches*, servidores e clientes, sendo ideal para simulações e para servir como prova de conceito.

Na implementação deste projeto utilizamos o GNS3. Desta forma podemos simular uma CDN baseada em DNS num ambiente virtual.

C. Idealização de um Protótipo

O protótipo que idealizamos pretende estabelecer uma arquitetura funcional de CDN que permita distribuir conteúdos estáticos utilizando o DNS como **MD**. Para isso, serão configurados dois nós de *cache* (**SR**), localizados em diferentes regiões geográficas. Estes nós terão como função armazenar localmente os conteúdos estáticos, reduzindo a carga sobre o **SO** e diminuindo o tempo de resposta para os utilizadores.

Uma parte fundamental deste projeto passa pela configuração de um sistema de DNS que permita redirecionar os pedidos de conteúdo para o nó de *cache* mais próximo do utilizador, com base na sua localização ou na latência da ligação. Com o GNS3, é possível simular diferentes cenários de rede, incluindo *delays* de comunicação entre dispositivos, o que permite testar a eficácia da CDN em contextos mais realistas.

Através das ferramentas que o GNS3 disponibiliza, será possível realizar uma avaliação de desempenho da solução implementada, comparando os tempos de resposta com e sem a utilização da CDN, bem como a eficiência na resolução DNS e distribuição do tráfego de rede.

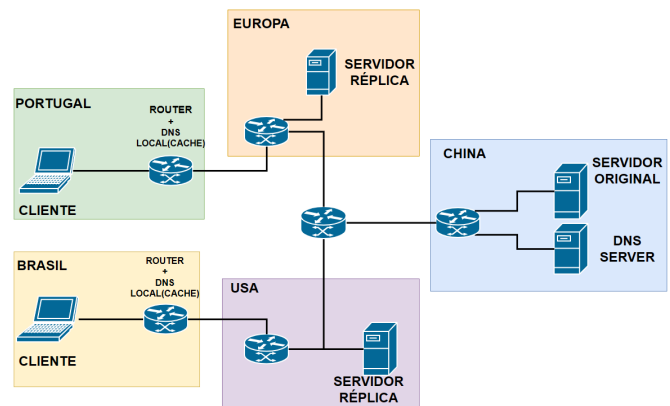


Fig. 3: Esquema do projeto CDN

Fig. 7: Configuração da view USA

```

GNU nano 7.2          admredes.pt-usa.zone
$ORIGIN admredes.pt.
$TTL 86400

@      SOA      dns up202408644.fc.up.pt. (
                                2025112803
                                3h
                                15
                                1w
                                3h
                                )
      NS       dns
      MX       10      mail

cdnnode A      192.2.2.2
router  A      172.16.0.1
;SERVIDORES
dns     A      172.16.0.2
mail    A      172.16.0.7
;ALIAS
www     CNAME   dns

```

Fig. 8: Configuração da zona USA

0) Zona default

Para pedidos com origem noutro destino, definimos uma view default, que devolve o Ip do SO que se encontra na China (ver Fig. 9 e Fig. 10).

```

view "other" {
    match-clients { any; };
    recursion no;
    zone "admredes.pt" IN {
        type master;
        file "master/admredes.pt-other.zone";
    };
};

```

Fig. 9: Configuração da view default

```

GNU nano 7.2          admredes.pt-other.zone
$ORIGIN admredes.pt.
$TTL 86400

@      SOA      dns up202408644.fc.up.pt. (
                                2025112803
                                3h
                                15
                                1w
                                3h
                                )
      NS       dns
      MX       10      mail

cdnnode A      192.3.3.2
router  A      172.16.0.1
;SERVIDORES
dns     A      172.16.0.2
mail    A      172.16.0.7
;ALIAS
www     CNAME   dns

```

Fig. 10: Configuração da zona default

C. Configuração Servidor Origem e Réplicas

Para construir a CDN em si decidimos utilizar o NGINX. A instalação e respetiva configuração do NGINX foi feita de forma separada quer no SO quer nos SR. Nos SR, a lógica implementada, lida com o pedido procurando o recurso localmente. Se esse recurso não for encontrado, o pedido é então encaminhado ao SO, que irá fornecer-lo (caso exista), e essa resposta é posteriormente guardada em cache no respetivo SR (ver Fig. 11).

```

GNU nano 7.2          /etc/nginx/nginx.conf
server {
    listen      192.1.1.2:80;

    server_name _;
    location / {

        proxy_cache my_cache;
        proxy_pass http://192.3.3.2:80;
        proxy_cache_valid 200 202 10m;
        proxy_cache_valid 404 1m;
        proxy_cache_use_stale error timeout updating;
        add_header X-Cache-Status $upstream_cache_status;
    }
    include /etc/nginx/default.d/*.conf;
}

```

Fig. 11: Configuração NGINX servidor réplica Europa

Para o SO, apenas definimos no ficheiro de configuração o caminho onde os recursos, que vão ser distribuídos para os clientes, estão (ver Fig. 12).

```

server {
    #listen      192.3.3.2:80;
    listen 80;
    server_name _;
    root      /var/www/html/;
    index index.html index.htm;

    location / {
        try_files $uri $uri/ =404;
    }

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;
}

```

Fig. 12: Configuração NGINX servidor Origem

D. Configuração dos Routers

Para possibilitar a comunicação entre todos os nós da rede, configuramos o protocolo OSPF. Para além disso, tal como mencionado anteriormente, configuramos um delay e largura de banda nas interfaces dos routers para tornar o ambiente de simulação mais realista.

E. Análise de Resultados

Para o procedimento deste teste utilizamos o comando `curl -w "@curl-format.txt" -o /dev/null -s "http://cdnnode.admredes.pt/200MB-TESTFILE.ORG.pdf"` nos terminais, sendo o `curl-format.txt` um ficheiro de texto onde irá ser possível imprimir e visualizar os resultados dos tempos obtidos na obtenção do ficheiro.

Observando os resultados na tabela I que foram obtidos apartir dos terminais, podemos concluir que o caching feito por o SR permite obter o ficheiro de forma significativamente mais rápida, quando comparado com obtenção do ficheiro a partir do SO.

	Servidor Réplicas (ficheiro em cache)	Servidor Origem
Terminal em Portugal	73s	539s
Terminal no Brasil	60s	179s

TABLE I: Tempos obtidos na transferência de um ficheiro de 200MB a partir de ambos os terminais (Brasil e Portugal)

Podemos concluir que a CDN auxiliou na redução do tempo de distribuição de conteúdo para os clientes. Consequente-

mente também reduziu o tempo de espera do cliente para receber estes ficheiros.

É fundamental destacar que, estando num ambiente virtualizado, a repetição dos testes irá produzir resultados consistentes, um comportamento característico que contrasta com a variabilidade esperada em sistemas não virtualizados.

V. CDN BASEADA EM DNS NA CLOUD

Visando resultados mais representativos de um cenário real de entrega de conteúdo, optamos por implementar uma CDN na cloud, utilizando a plataforma Azure e seus serviços dedicados.

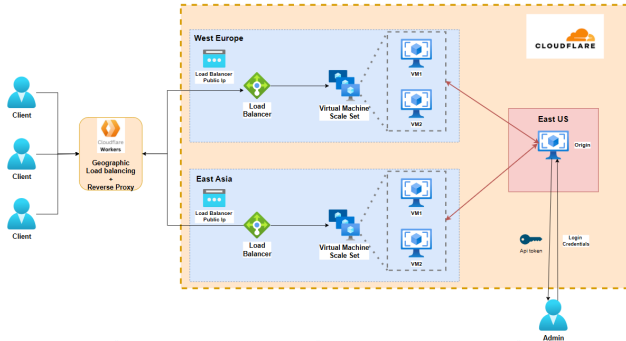


Fig. 13: Arquitetura CDN baseada em Cloud

A Figura 13 ilustra a arquitetura proposta para CDN baseada em DNS na cloud. A arquitetura está estrategicamente dividida em dois componentes principais: o Frontend e o Backend. O **Frontend** utiliza Cloudflare Workers para load balancing geográfico e funcionalidades de reverse proxy. A infraestrutura de **Backend**, está presente em diferentes regiões geográficas (West EU e East Asia no nosso caso), compreende load balancer regionais com endereços IP públicos, VMSS contendo os **SR** e um **SO** localizado na região East US.

Para construir a CDN em si decidimos utilizar uma configuração NGINX semelhante a que tínhamos no ambiente virtual. Para resolução de nomes, utilizamos a infraestrutura da cloudflare, e utilizamos um domínio válido que obtemos a parte.

Para além de um ambiente mais realista temos a Cloudflare a providenciar diferentes aspetos de segurança como: mitigação de ataques DDoS, Web Application Firewall (WAF) básico, Certificados SSL/TLS, conexão HTTPS, etc...

A. Resultados

Para os testes utilizamos VM com tamanho **Standard_B2ms** [6] para todas as VMs instanciadas (**SR** e **SO**). No **SO** utilizamos um SSD com 128 Gb e no **SR** utilizamos um SSD com 32 Gb.

Utilizando o computador pessoal e uma rede doméstica (WiFi de 2,4 GHz com velocidades médias de download de 40-50 MBps), fizemos testes de desempenho para o cliente da UE. Os resultados podem ser vistos na tabela II

	MISS	HIT
200 Mb	≈ 25.87s	≈ 22.29s
20 Mb	≈ 6.35s	≈ 3.33s
4 Kb	≈ 0.99s	≈ 0.43s

TABLE II: Tempos aproximados de cache MISS e cache HIT (WEST EU)

Ao contrário do teste do cliente da UE numa rede local, a avaliação do cliente da Ásia utilizou uma VM simples do Azure situada na mesma região que o VMSS e o load balancer. Isso resultou num desempenho significativamente melhor, principalmente porque os nós de cache estão geograficamente mais próximo do cliente e a VM do Azure oferecia uma conexão de rede muito mais confiável e rápida. Os resultados podem ser vistos na tabela III.

	MISS	HIT
200 Mb	≈ 16.40s	≈ 2.48s
20 Mb	≈ 3.83s	≈ 0.46s
4 Kb	≈ 0.99s	≈ 0.34s

TABLE III: Tempos aproximados de cache MISS e cache HIT (East Asia)

VI. CONCLUSÃO

Com este relatório foi possível explicar o funcionamento de uma CDN com recurso a um exemplo real, estabelecer os princípios fundamentais das CDNs e delinear os seus componentes num protótipo. Esse protótipo foi implementado com sucesso, em ambiente virtual e na cloud.

Resultados de testes em ambientes virtualizados e na cloud permitem afirmar a eficácia de uma CDN no que consta a redução significativa dos tempos de transferência e da espera do cliente. As descobertas de ambos os ambientes sublinham a importância das CDNs na infraestrutura atual da internet.

REFERENCES

- [1] Cloudflare, "What is a content delivery network (CDN)? — How do CDNs work?". [Online]. Disponível: <https://www.cloudflare.com/learning/cdn/what-is-a-cdn/>.
- [2] Akamai, "O que é uma rede de entrega de conteúdo (CDN - Content Delivery Network)?" [Online]. Disponível: <https://www.akamai.com/pt/glossary/what-is-a-cdn>.
- [3] G. Peng, "CDN: Content Distribution Network", 2018, pp. 3-5
- [4] <https://instances.vantage.sh/azure/vm/b2ms>
- [5] <https://github.com/RS181/TAR>
- [6] "Azure Virtual Machine: Standard B2ms". [Online]. Disponível: <https://instances.vantage.sh/azure/vm/b2ms>.
- [7] "RS181/TAR". [Online]. Disponível: <https://github.com/RS181/TAR>.