**Compression in HTTP**

A compressão é uma forma importante de aumentar o desempenho de um site. Para alguns documentos, uma redução de até 70% no tamanho reduz a necessidade de largura de banda. Com o passar dos anos, os algoritmos também se tornaram mais eficientes, e novos algoritmos são suportados por clientes e servidores.

Na prática, os desenvolvedores web não precisam implementar mecanismos de compressão; tanto navegadores quanto servidores já os têm implementados, mas precisam garantir que o servidor esteja configurado corretamente. A compressão ocorre em três níveis diferentes:

* Primeiro, alguns formatos de arquivo são compactados com métodos otimizados específicos,
* então, a compactação geral pode ocorrer no nível HTTP (o recurso é transmitido compactado de ponta a ponta),
* e, finalmente, a compactação pode ser definida no nível de conexão, entre dois nós de uma conexão HTTP.

**File format compression**

Cada tipo de dado possui alguma redundância, ou seja, desperdício de espaço. Se o texto normalmente pode apresentar até 60% de redundância, essa taxa pode ser muito maior para outras mídias, como áudio e vídeo. Ao contrário do texto, esses outros tipos de mídia usam muito espaço para armazenar seus dados, e a necessidade de otimizar o armazenamento e recuperar espaço era evidente desde o início. Engenheiros desenvolveram o algoritmo de compactação otimizado usado por formatos de arquivo projetados para essa finalidade específica. Os algoritmos de compactação usados ​​para arquivos podem ser agrupados em duas grandes categorias:

* Loss-less compression (Compressão sem perdas), onde o ciclo de compressão-descompressão não altera os dados recuperados. Eles correspondem (byte a byte) ao original. Para imagens, GIF ou PNG usam compressão sem perdas.
* Lossy compression (Compressão com perdas), onde o ciclo altera os dados originais de forma (esperançosamente) imperceptível para o usuário. Formatos de vídeo na web apresentam perdas; o formato de imagem JPEG também apresenta perdas.

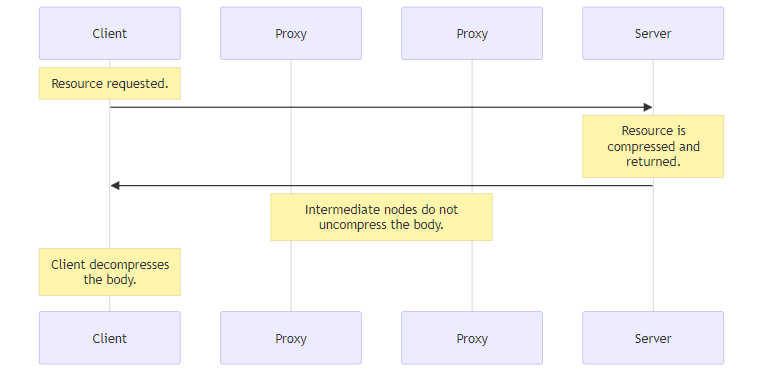
Alguns formatos podem ser usados ​​para compressão com ou sem perdas, como o webp, e geralmente o algoritmo com perdas pode ser configurado para comprimir mais ou menos, o que, naturalmente, resulta em menor ou maior qualidade. Para um melhor desempenho de um site, o ideal é comprimir o máximo possível, mantendo um nível aceitável de qualidade. Para imagens, uma imagem gerada por uma ferramenta pode não estar otimizada o suficiente para a web; recomenda-se o uso de ferramentas que comprimam o máximo possível com a qualidade necessária. Existem inúmeras ferramentas especializadas para isso.

Algoritmos de compressão com perdas geralmente são mais eficientes do que aqueles sem perdas.

Observação: como a compactação funciona melhor em um tipo específico de arquivo, geralmente não há necessidade de compactá-lo uma segunda vez. Na verdade, isso costuma ser contraproducente, pois o custo da sobrecarga (algoritmos geralmente exigem um dicionário que aumenta o tamanho inicial) pode ser maior do que o ganho extra na compactação, resultando em um arquivo maior. Não use as duas técnicas a seguir para arquivos compactados.

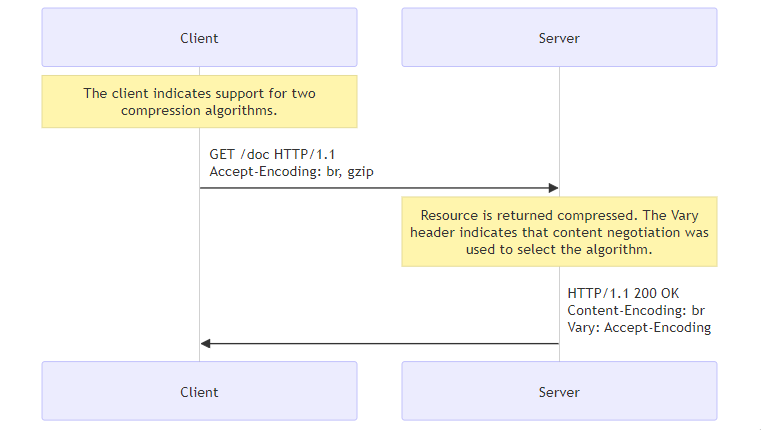
**End-to-end compression (Compressão de ponta a ponta)**

Em termos de compressão, a compressão de ponta a ponta é onde residem as maiores melhorias de desempenho dos sites. A compressão de ponta a ponta refere-se à compressão do corpo de uma mensagem, realizada pelo servidor e que permanece inalterada até chegar ao cliente. Quaisquer que sejam os nós intermediários, eles deixam o corpo intacto.



Todos os navegadores e servidores modernos suportam essa tecnologia, e a única coisa a ser negociada é o algoritmo de compactação a ser usado. Esses algoritmos são otimizados para texto. Na década de 1990, a tecnologia de compactação avançava em ritmo acelerado e vários algoritmos sucessivos foram adicionados ao conjunto de opções possíveis. Hoje em dia, apenas dois são relevantes: gzip, o mais comum, e br, o novo concorrente.

Para selecionar o algoritmo a ser usado, navegadores e servidores utilizam negociação proativa de conteúdo. O navegador envia um cabeçalho Accept-Encoding com o algoritmo suportado e sua ordem de precedência; o servidor escolhe um, usa-o para compactar o corpo da resposta e usa o cabeçalho Content-Encoding para informar ao navegador o algoritmo escolhido. Como a negociação de conteúdo foi usada para escolher uma representação com base em sua codificação, o servidor deve enviar um cabeçalho Vary contendo pelo menos Accept-Encoding junto com esse cabeçalho na resposta; dessa forma, os caches poderão armazenar em cache as diferentes representações do recurso.



Como a compactação traz melhorias significativas de desempenho, é recomendável ativá-la para todos os arquivos, exceto aqueles já compactados, como imagens, arquivos de áudio e vídeos.

O Apache suporta compactação e usa mod\_deflate; para Nginx há ngx\_http\_gzip\_module; para IIS, o elemento <httpCompression>.

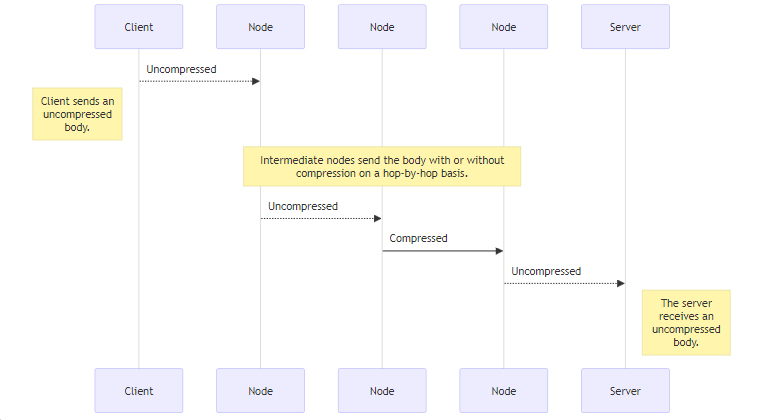
**Compression dictionary transport**

Formatos de compactação modernos, como a compactação Brotli e a compactação Zstandard, podem usar dicionários de dados usados ​​com frequência para aumentar ainda mais a compactação, em vez de apenas referenciar aqueles contidos no arquivo que está sendo compactado. Normalmente, para respostas HTTP, isso usa o dicionário estático predefinido incluído nesse formato (por exemplo, o dicionário estático Brotli está disponível no código-fonte).

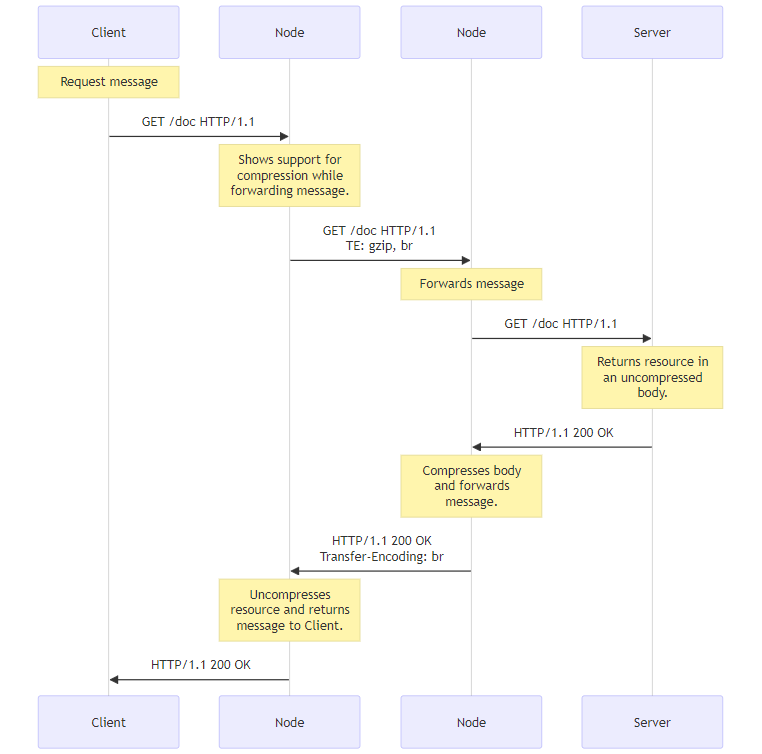
O Transporte de Dicionário de Compressão permite que um desenvolvedor especifique um recurso que pode ser usado como dicionário para solicitações futuras. Este pode ser um arquivo de dicionário específico ou um recurso existente (por exemplo, usar app.v1.js como dicionário ao baixar app.v2.js). Isso normalmente melhora a compressão e, portanto, o tempo de carregamento. No exemplo app.vX.js, a maior parte do download consistiria apenas no delta entre as duas versões, e os bytes comuns poderiam ser referenciados a partir do arquivo app.v1.js original que já foi baixado.

**Hop-by-hop compression (Compressão salto a salto)**

A compressão salto a salto, embora semelhante à compressão ponta a ponta, difere em um elemento fundamental: a compressão não ocorre no recurso no servidor, criando uma representação específica que é então transmitida, mas no corpo da mensagem entre quaisquer dois nós no caminho entre o cliente e o servidor. Conexões entre nós intermediários sucessivos podem aplicar uma compressão diferente.



Para isso, o HTTP usa um mecanismo semelhante à negociação de conteúdo para compressão de ponta a ponta: o nó que transmite a solicitação anuncia sua vontade usando o cabeçalho TE e o outro nó escolhe o método adequado, o aplica e indica sua escolha com o cabeçalho Transfer-Encoding.



Na prática, a compressão hop-by-hop é transparente para o servidor e o cliente e raramente é utilizada. TE e Codificação de Transferência são usados ​​principalmente para enviar uma resposta em blocos, permitindo iniciar a transmissão de um recurso sem conhecer seu comprimento.

Observe que o uso de codificação de transferência e compactação no nível do salto é tão raro que a maioria dos servidores, como Apache, Nginx ou IIS, não tem uma maneira fácil de configurá-los. Essa configuração geralmente ocorre no nível do proxy.