



HAProxy

Powering Your Uptime

ALOHA Load-Balancer - Application Note

Compression HTTP

Document version: v1.1

Last update: 3 juin 2014

EMEA Headquarters

3, rue du petit robinson

ZAC des Metz

78350 Jouy-en-Josas

France

<http://www.haproxy.com/>

Objectif

La compression HTTP est une technique qui permet de réduire la taille des réponses en les compressant. Elle a plusieurs intérêts :

- chargement des objets plus rapides : moins d'octets à transférer
- réduit les coûts de bande passante
- réduit l'empreinte dans les serveurs de cache de contenu

La compression est déjà largement déployée sur les serveurs webs tels que Apache, nginx et IIS.

Limitations

Afin de fonctionner, le client doit annoncer quels algorithmes de compression il supporte et le serveur doit être compatible avec au moins l'un de ces algorithmes.

Difficulté



Versions concernées

- Aloha 5.5 et supérieur

Changelog

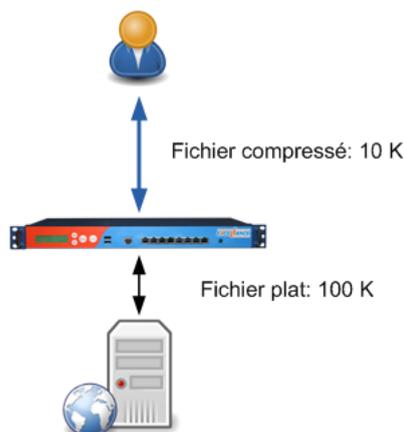
Version	Description
1.1	2014-06-02 <ul style="list-style-type: none">- Mise à jour du thème HAProxy Tech.- Changements mineurs
1.0	2013-01-03 Version initiale

Contexte

L'**ALOHA** Load-Balancer peut compresser les réponses en lieu et place des serveurs ou peut compresser à la volée les réponses qui ne l'auraient pas été par les serveurs (mais qui auraient clairement dûs l'être).
The **ALOHA** adapte dynamiquement le taux de compression en fonction de sa charge.

Schéma

Le schéma ci-dessous montre comment se passe la compression lorsqu'elle est effectuée sur l'**ALOHA** Load-Balancer :



HAProxy et la compression

La compression est autorisée par le client à travers le header HTTP **Accept-Encoding** de la requête : pas de header, pas de compression.

HAProxy analyse les flux en temps réel. Si le serveur supporte la compression HTTP, alors **HAProxy** laissera passer les réponses compressées telles quelles. Si le serveur ne supporte pas la compression HTTP, alors **HAProxy** compressera la réponse (en conformité avec la requête du client).

Lorsque **HAProxy** est configuré pour décharger la compression des serveurs (offload), il supprime le header **Accept-Encoding**, pour empêcher le serveur de compresser la réponse.

HAProxy ne va pas compresser dans les cas suivants :

- Le client n'a pas annoncé un algorithme compris par **HAProxy**
- Le réponse n'est pas annoncée en HTTP/1.1
- Le status code HTTP de la réponse n'est pas 200
- La réponse ne contient ni l'en-tête "**Transfer-Encoding** : chunked" ni **Content-Length**
- Le **Content-Type** de la réponse est "multipart"
- La requête contient l'en-tête "Cache-control : no-transform"
- L'en-tête **User-Agent** de la requête contient "Mozilla/4" mais pas "MSIE 6 with XP SP2" ou "MSIE 7" ou supérieur
- La réponse est déjà compressée



La compression ne change pas l'en-tête **Etag**

Pour le moment **HAProxy** supporte la compression **gzip**. **deflate** est aussi supporté mais ne doit pas être utilisé en production : cet algorithme est implémenté de différentes manières côté client et sont mutuellement incompatible.

Configuration

Utilisation courante

La configuration ci-dessous s'applique quand les serveurs compressent déjà les objets. L'**ALOHA** compressera tout ce qui aura été omis par les serveurs.

Cette directive peut être ajoutée dans les sections **default**, **frontend** ou **backend**.

```
compression algo gzip
compression type text/html text/plain text/css text/javascript
```

Déchargement de compression

La configuration ci-dessous s'applique lorsque l'on souhaite que la compression soit effectuée uniquement par l'**ALOHA**. Cette directive peut être ajoutée dans les sections **default**, **frontend** ou **backend**.

```
compression algo gzip
compression offload
compression type text/html text/plain text/css text/javascript
```

Ce mode est utilisé pour empêcher certains serveurs buggés de corrompre les réponses.