

1. Introduction

Varnish est un reverse proxy HTTP principalement utilisé comme accélérateur web. Varnish conserve (cache) des fichiers (ou des fragments) en mémoire pour réduire le temps de réponse et la consommation de bande passante. Il peut également faire office de : Pare-feu, Bouclier anti-DDoS, Répartiteur de charge et contrôleur d'accès. Varnish se présente sous la forme d'une suite logicielle, déclinée en deux distributions :

Varnish Cache :

- Version Open Source (BSD)
- Suite logicielle en ligne de commande
- Support de SSL/TLS (frontend uniquement)

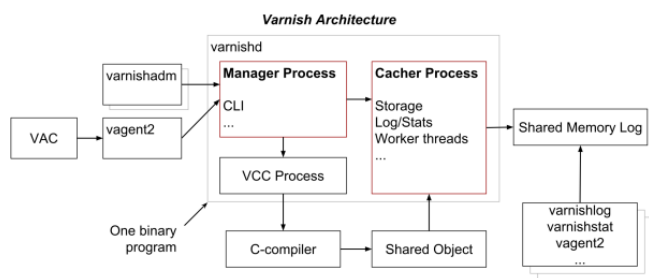
Varnish Plus :

- Version commerciale de la suite;
- Prix de base: 31'000\$;
- Support client.
- Support de SSL/TLS (frontend et backend);
- Varnish Administration Console (VAC);
- Varnish High Availability (VHA);
- Varnish Custom Statistics (VCS).

2. Fonctionnement

Le protocole HTTP prévoit des entêtes pour émettre des directives de contrôle du caching des objets tels que vu durant le cours. Varnish se base donc sur ces entêtes HTTP des réponses, pour déterminer le comportement de leur mise en cache.

3. Architecture



Manager:

- Applique les changements de configuration;
- Délègue les tâches au Cachier et au VCC;
- Surveille Varnish;
- Fournit une invite de commande.

VCC:

- Traduit le VCL en C (Qui est ensuite traité par un compilateur traditionnel, tel que gcc).

Cacher:

- Écoute les requêtes clientes;
- Gère les worker threads;
- Sauve les objets en cache;
- Log le trafic;
- Met à jour les compteurs pour les statistiques.

4. VCL

Le Varnish Configuration Language (VCL) est le langage de configuration de Varnish. Il est notamment utilisé pour écrire ses politiques de caching. En interne, il est traduit en C.

Exemples d'utilisation:

Forcer la mise en cache de toutes les images, au format .jpg, pendant 5 minutes :

```
sub vcl_fetch {
    if (req.url ~ "\.jpg$") {
        set beresp.ttl = 300;
    }
}

sub vcl_recv {
    if (req.url ~ "^/index\.html" ||
        req.url ~ "^/$") { return(pass); }
}
```

5. Utilitaires

Plusieurs utilitaires sont disponibles dans Varnish :

- varnishadm : permet de contrôler une instance varnish qui s'exécute
- varnishstat : affiche des statistiques relatives au caching
- vagent 2 : agent vagrant permettant de faire des requête REST afin de contrôler à distance et monitorer varnish
- varnishtest : permet de simuler une architecture. En voici un exemple :

```
varnishtest "#1029"

server s1 {
    rxreq
    expect req.url == "/bar"
    txresp -gzipbody {[bar]}
} -start

client c1 {
    txreq -url "/bar" -hdr "Accept-Encoding: gzip"
    rxresp
    gunzip
    expect resp.bodylen == 5
} -run
```

6. Déploiement

L'architecture la plus simple est d'avoir un client, une machine dédiée à Varnish et un serveur web. Le client interagit uniquement avec le proxy. :

