

---

# Web caching with Varnish

— Robin Gaudin, Lev Pozniakoff,  
Axel Vallon et Nicolas Viotti —

---

# Organisation de la présentation

1. Introduction: Qu'est ce que Varnish ?
2. Avantages et inconvénients
3. Caractéristiques techniques
4. VCL
5. Comparatif
6. Démonstration
7. Conclusion

# Introduction: Qu'est ce que Varnish ?

- Serveur de caching http
- Reverse proxy
- License BSD
- Configuration via langage propre à Varnish: VCL
- Accélère les web app
- Installation devant le serveur HTTP

## Avantages

- Temps de chargement court
- Soulagement du serveur Web
- Prise en charge des balises ESI\*
- L'OS stocke les contenus sur le disque dur
- Load Balancer avec Round Robin
- Diverses possibilités de configurations grâce à VCL

## Inconvénients

- Pour les systèmes sans ESI, aucune optimisation possible
- Complexité, peut générer des bugs sur le site
- Aucune prise en charge par TLS/SSL voire HTTPS
- Configuration chronophage et complexe
- Seulement sous UNIX

```
<!DOCTYPE html><html>
  <body>
    <!-- ... some content -->

    <!-- Embed the content of another page here -->
    <esi:include src="http://..." />

    <!-- ... some content -->
  </body></html>
```

- \*Voici un balisage ESI permettant de générer une page dynamiquement à partir d'une autre

# Caractéristiques techniques

- Caching
  - Reverse proxy cache



- Scalability
  - Possibilité d'auto-scaling
  - Sky, Tesla, Twitch, Facebook, Pinterest, ...

# VCL (Varnish Configuration Language)

- Langage de programmation dédié
- Traduit en langage C
- Compilé et chargé dynamiquement par le binaire varnishd
- Apporte beaucoup de flexibilité
- 10 sous-routines, mais 2 réellement utiles :
  - `vcl_recv`
  - `vcl_fetch`

# VCL (Varnish Configuration Language)

- vcl\_recv :

exécutée lorsqu'une requête HTTP envoyée par le client a été reçue et analysée par Varnish

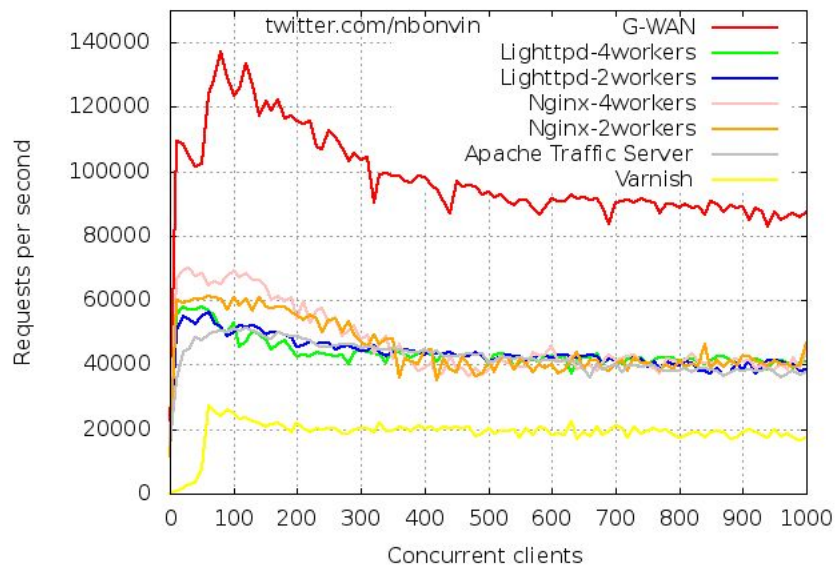
- vcl\_fetch (ou vcl\_backend\_response) :

exécutée lorsque Varnish a fini de réceptionner la réponse du backend à la requête précédemment transmise

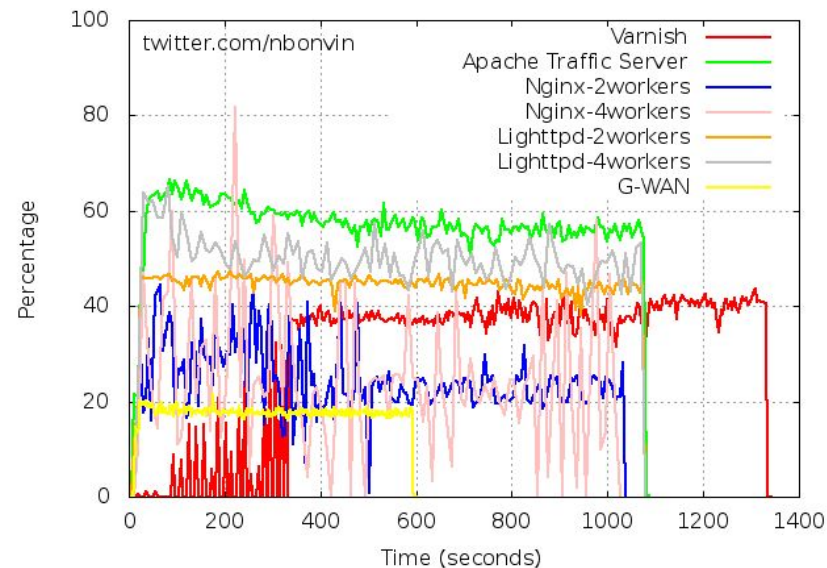
```
sub vcl_recv {  
    if(req.http.host == "static.monsite.fr"){  
        set req.backend = static1;  
    }  
    else if(req.http.host ~ "(www.)?monsite.fr"){  
        set req.http.host = "www.monsite.fr";  
        set req.backend = php;  
    }  
    if(req.http.authorization || req.request == "POST" || req.http.cookie){  
        return(pass);  
    }  
    return(lookup);  
}  
sub vcl_fetch {  
    if(req.http.host == "www.monsite.fr" && !req.url ~ "^/admin"){  
        unset beresp.http.set-cookie;  
    }  
}
```

# Comparatif

Average RPS



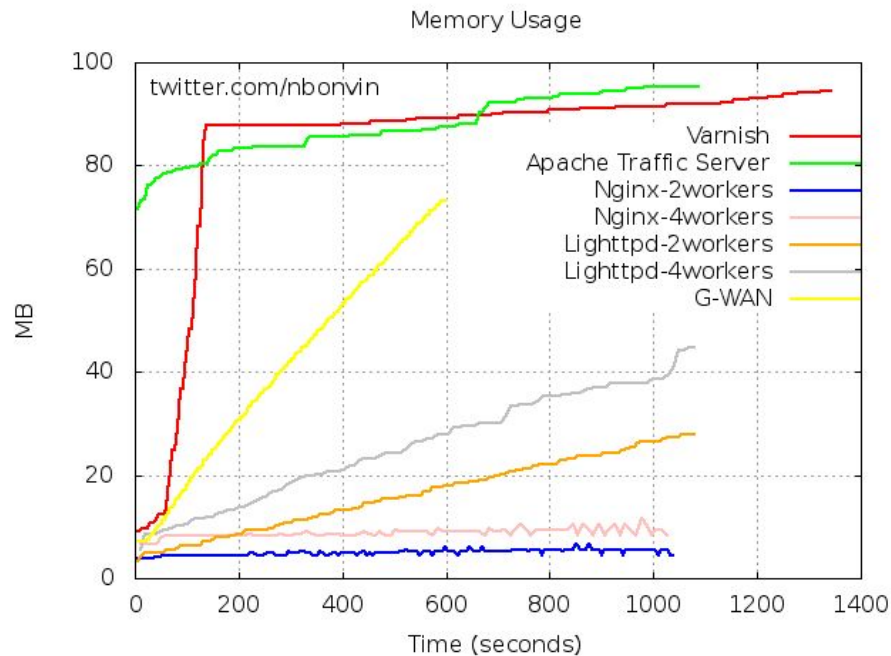
CPU Usage



<http://gwan.com/benchmark>



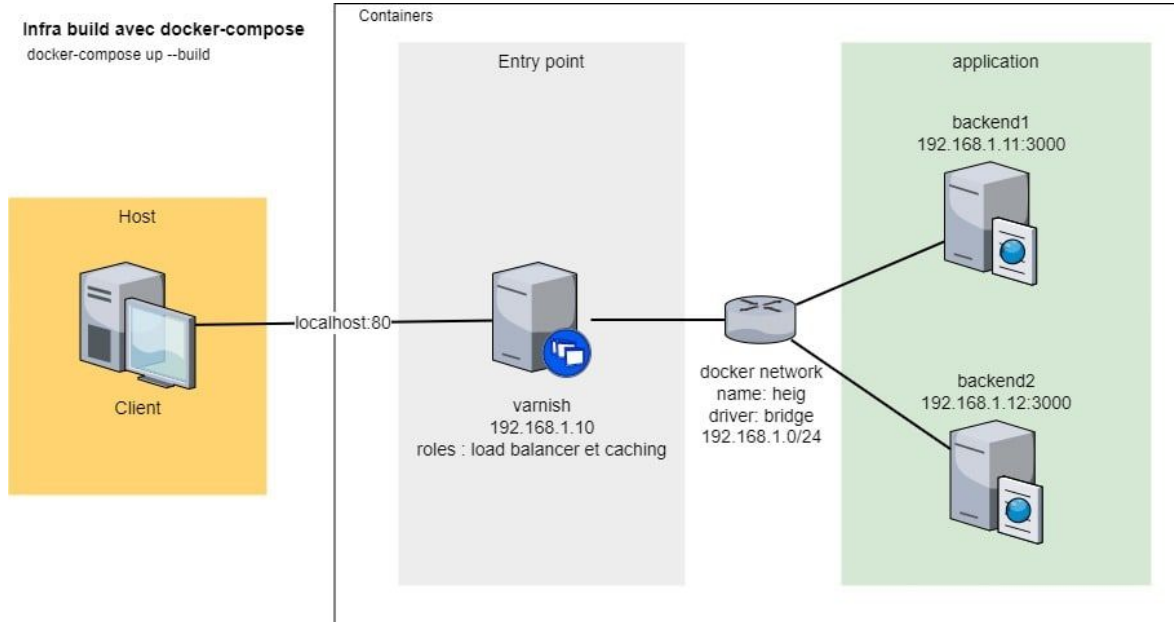
# Comparatif (2)



Competitors: A stolen slide,  
probably not up-to-date!

	ATS	HAproxy	nginx	Squid	Varnish	mod_proxy
Worker Threads	Y	N	N	N	Y	Y
Multi-Process	N	Y	Y	N	Y	Y
Event-driven	Y	Y	Y	Y	N	N?
Plugin APIs	Y	N	Y	part	Y	Y
Forward Proxy	Y	N	N	Y	N	Y
Reverse Proxy	Y	Y	Y	Y	Y	Y
Transp. Proxy	Y	Y	N	Y	N	N
Load Balancer	part	Y	Y	Y	Y	Y
Cache	Y	N	Y	Y	Y	Y
ESI	soon	N	N	Y	Y	N
ICP	Y	N	N	Y	N	N
Keep-Alive	Y	N	Y	Y	Y	Y
SSL	Y	N	Y	Y	N	Y

# Démonstration



<https://github.com/Semestre5/AIT-presentation-2>

# Conclusion

- + Gratuit
- + Compétitif
- + Performant et scalable
- ~ Flexibilité supérieure mais complexe grâce à VCL
- Pas de support SSL intégré
- Seulement UNIX/LINUX/BSD
- Seulement HTTP