Transfert de session TLS

Stage de L3 à l'Irit – ENSEEIHT

Encadré par Daniel Hagimont avec l'aide de Brice Ekane

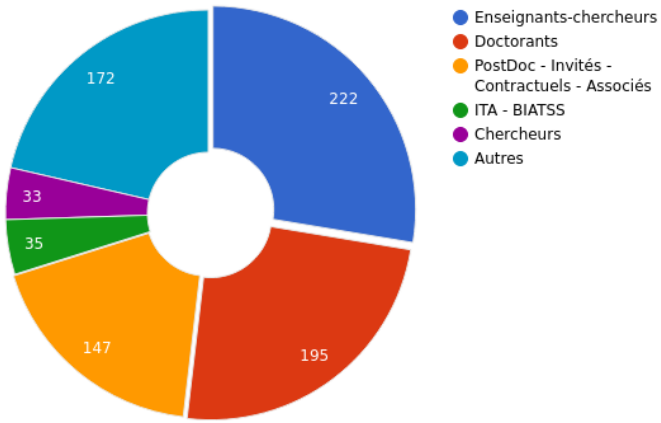
Adrien Vannson

13 juin 2022 – 22 juillet 2022

Contexte : l'Irit, l'équipe Sepia

L'Irit :

- Institut de Recherche en Informatique de Toulouse – UMR 5505
- 7 sites différents dont l'Enseeiht
- Environ 600 membres

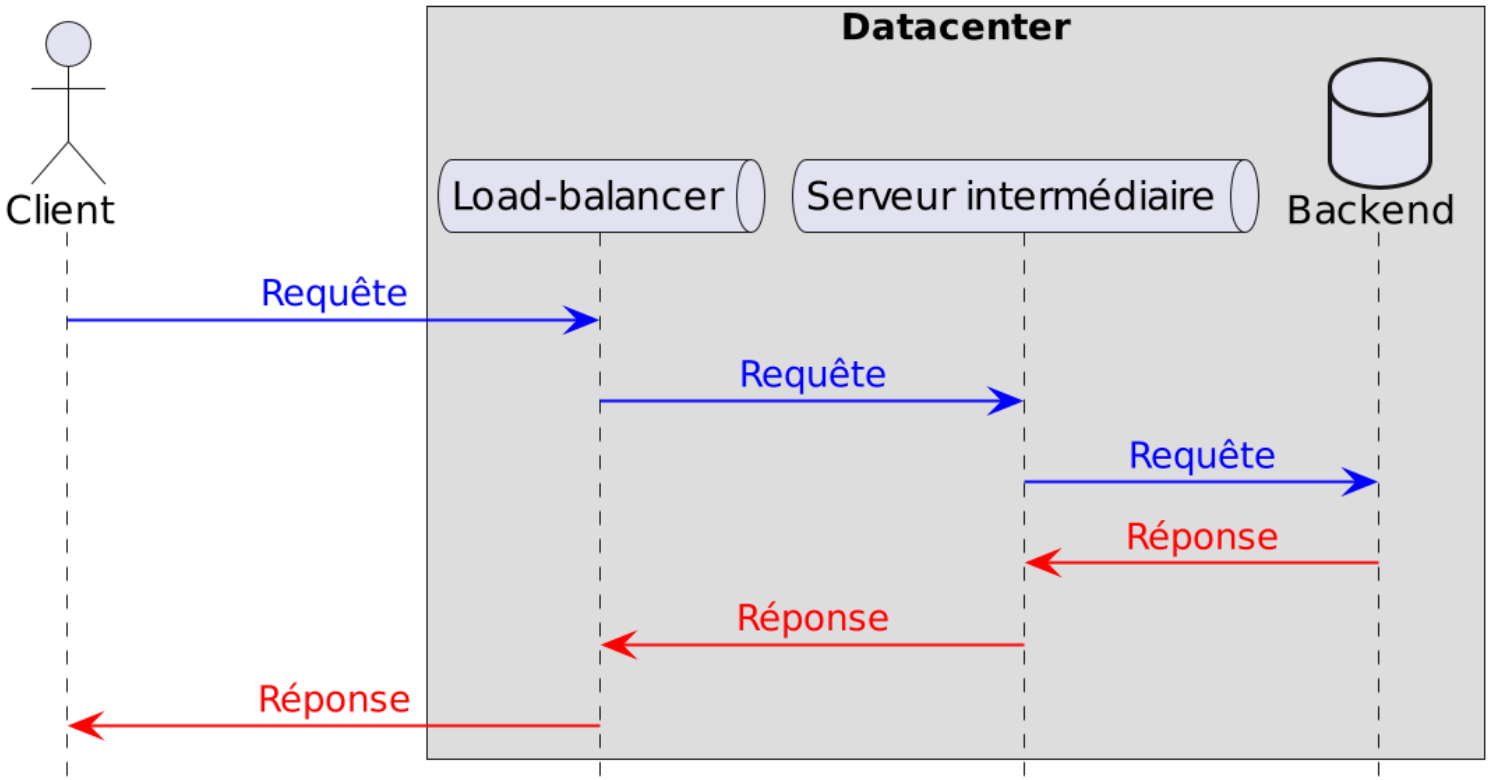


L'équipe Sepia :

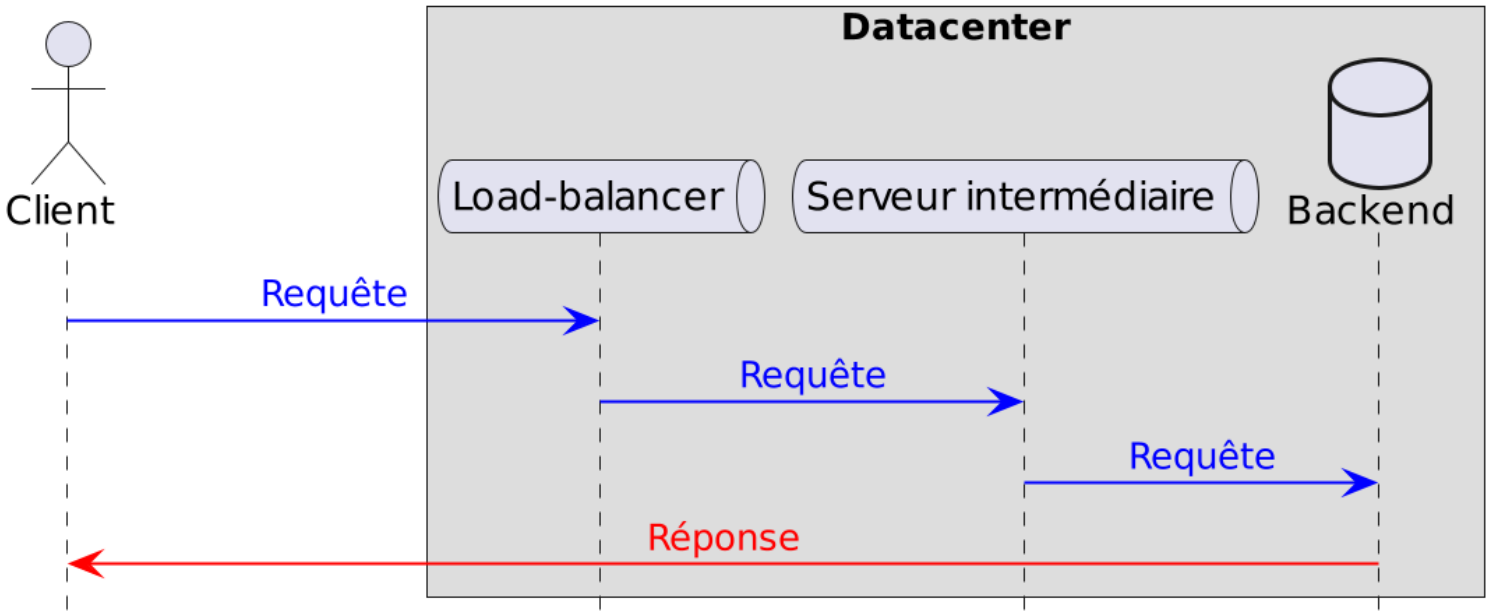
- 10 membres permanents
- Actuellement 30 membres au total
- S'intéresse à la gestion de ressources au sein des datacenters

Plan

Présentation du problème



DiSC

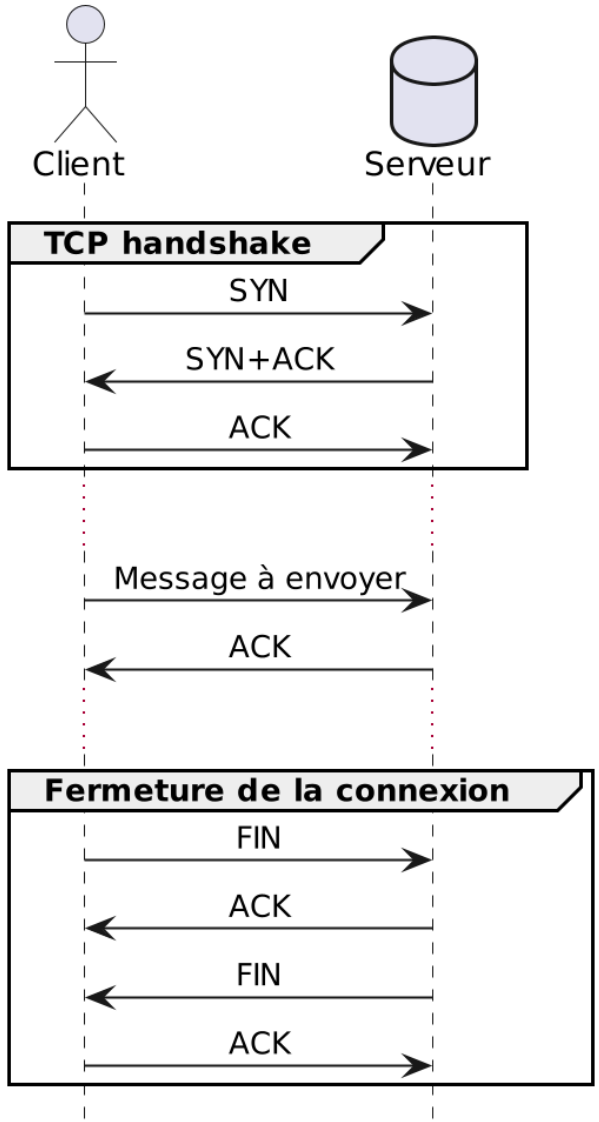


- *Distributed Shared Connection*
- Renvoyer les données directement depuis le backend de manière transparente pour le client (*spoofing*)
- Ne fonctionne pas si la connexion est chiffrée avec TLS, car le backend ne possède pas les informations de chiffrement du load-balancer
  → C'est l'objectif du stage !

Plan

Le protocole TCP

- Assure la fiabilité :
  - Établissement d'une connexion
  - Aquittement des paquets
  - Sommes de contrôle
  - Retransmission

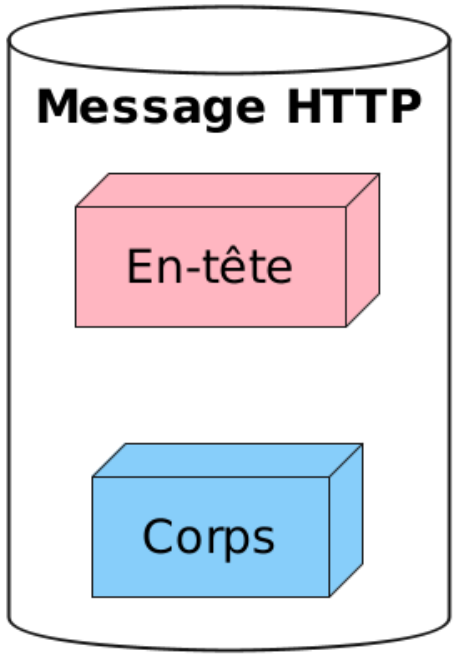


Le protocole TLS

- Successeur de SSL
- Assure la sécurité :
  - Authentification du serveur
  - Chiffrement des échanges
  - Intégrité des données

Le protocole HTTP

- Hypertext Transfer Protocol
- Développé pour le web
- HTTPS : variante utilisant TLS



Plan

Réalisation d'un prototype

Comment partager une session TLS entre plusieurs machines ?

- Avec OpenSSL ?
  - Piste suggérée initialement par mes encadrants...

Réalisation d'un prototype

Comment partager une session TLS entre plusieurs machines ?

- Avec OpenSSL ?
  - Piste suggérée initialement par mes encadrants...
  - ... mais ça ne fonctionne pas

Réalisation d'un prototype

Comment partager une session TLS entre plusieurs machines ?

- Avec OpenSSL ?
  - Piste suggérée initialement par mes encadrants...
  - ... mais ça ne fonctionne pas
- TLSe
  - Bibliothèque implémentant TLS
  - Utilisée dans Prism
  - Permet l'exportation de session

Réalisation d'un prototype

Comment partager une session TLS entre plusieurs machines ?

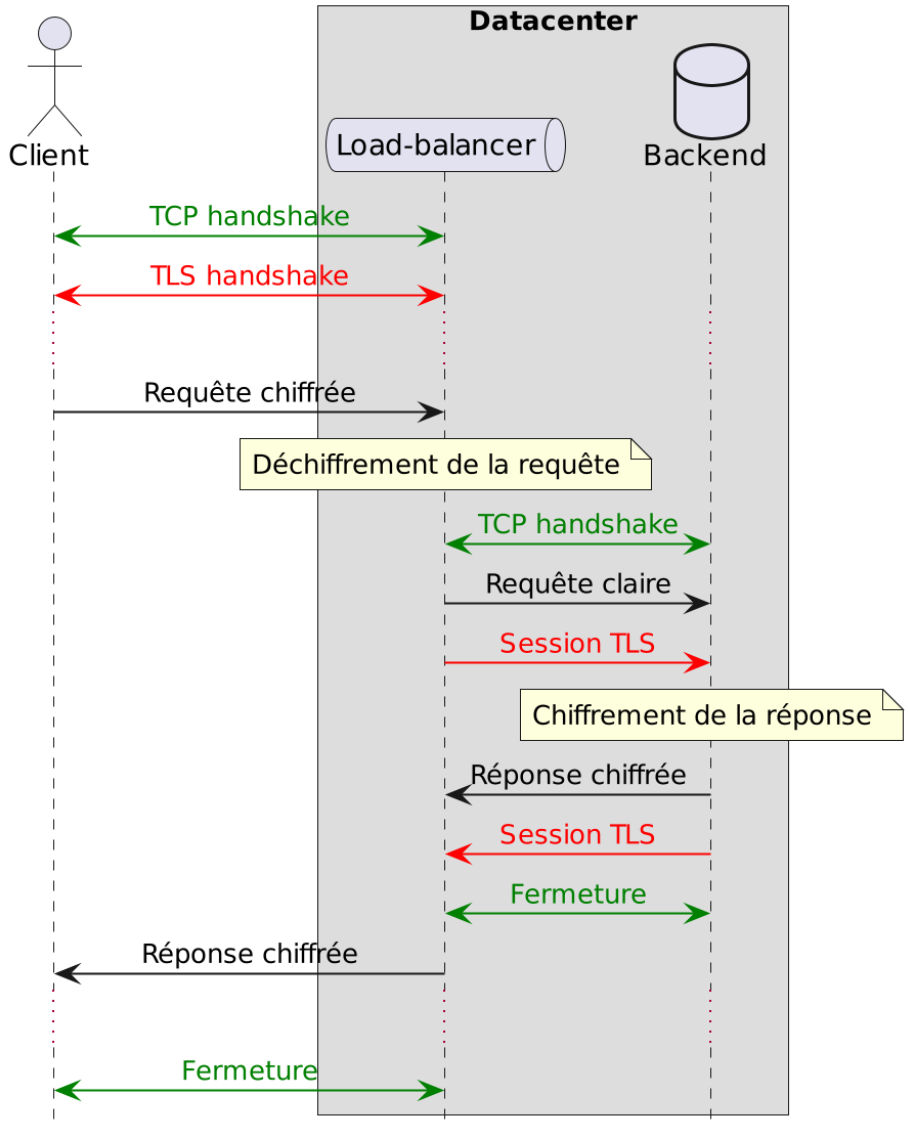
- Avec OpenSSL ?
  - Piste suggérée initialement par mes encadrants...
  - ... mais ça ne fonctionne pas
- TLSe
  - Bibliothèque implémentant TLS
  - Utilisée dans Prism
  - Permet l'exportation de session
  - ... mais petit projet, ne semble pas fiable, n'implémente pas toute l'API d'OpenSSL

Réalisation d'un prototype

Comment partager une session TLS entre plusieurs machines ?

- Avec OpenSSL ?
  - Piste suggérée initialement par mes encadrants...
  - ... mais ça ne fonctionne pas
- TLSe
  - Bibliothèque implémentant TLS
  - Utilisée dans Prism
  - Permet l'exportation de session
  - ... mais petit projet, ne semble pas fiable, n'implémente pas toute l'API d'OpenSSL
- WolfSSL
  - Bibliothèque implémentant TLS
  - Beaucoup utilisée, mise à jour régulièrement
  - Permet l'exportation de session

Prototype


Datacenter
Client
Load-balancer
Backend
TCP handshake
TLS handshake
Requête chiffrée
Déchiffrement de la requête
TCP handshake
Requête claire
Session TLS
Chiffrement de la réponse
Réponse chiffrée
Session TLS
Fermeture
Réponse chiffrée
Fermeture

Plan

Intégration à Nginx dans le load-balancer

- Nginx :
  - Serveur web, proxy inversé
  - Premier ou deuxième serveur web le plus utilisé au monde
- DiSC utilise Nginx $\rightarrow$ intégration nécessaire
- Nginx utilise OpenSSL...

Intégration à Nginx dans le load-balancer

- Nginx :
  - Serveur web, proxy inversé
  - Premier ou deuxième serveur web le plus utilisé au monde
- DiSC utilise Nginx $\rightarrow$ intégration nécessaire
- Nginx utilise OpenSSL...
- $\rightarrow$ Recompilation de Nginx avec WolfSSL (API compatible disponible)

Intégration à Nginx dans le load-balancer

- Nginx :
  - Serveur web, proxy inversé
  - Premier ou deuxième serveur web le plus utilisé au monde
- DiSC utilise Nginx $\rightarrow$ intégration nécessaire
- Nginx utilise OpenSSL...
- $\rightarrow$ Recompilation de Nginx avec WolfSSL (API compatible disponible)
- Première approche : étendre Nginx avec un filtre
  - Problème : le load-balancer doit renvoyer au client un `Content-Length` non nul sans envoyer de données après, ce qui faisait bloquer Nginx... N'a pas abouti

Intégration à Nginx dans le load-balancer

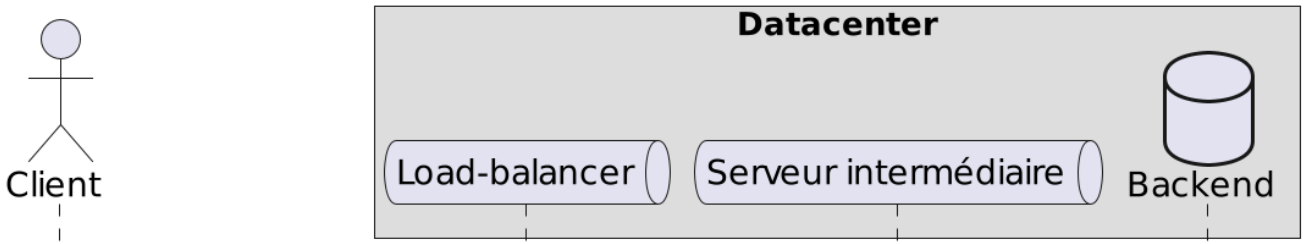
- Nginx :
  - Serveur web, proxy inversé
  - Premier ou deuxième serveur web le plus utilisé au monde
- DiSC utilise Nginx → intégration nécessaire
- Nginx utilise OpenSSL...
- → Recompilation de Nginx avec WolfSSL (API compatible disponible)
- Première approche : étendre Nginx avec un filtre
  - Problème : le load-balancer doit renvoyer au client un `Content-Length` non nul sans envoyer de données après, ce qui faisait bloquer Nginx... N'a pas abouti
- Deuxième approche : modifier directement le code source
  - Recherche de l'appel à `SSL_write` : dans `ngx_ssl_write`
  - Lorsqu'un header est reçu, modifier certains champs et transmettre la session TLS au backend

Plan

Intégration

CloudLab

- Intégration dans le projet DiSC
- Évaluation sur CloudLAB
- Quatre machines :
  - Client
  - Load-balancer (Nginx)
  - Serveur intermédiaire (Nginx)
  - Backend (Nginx)



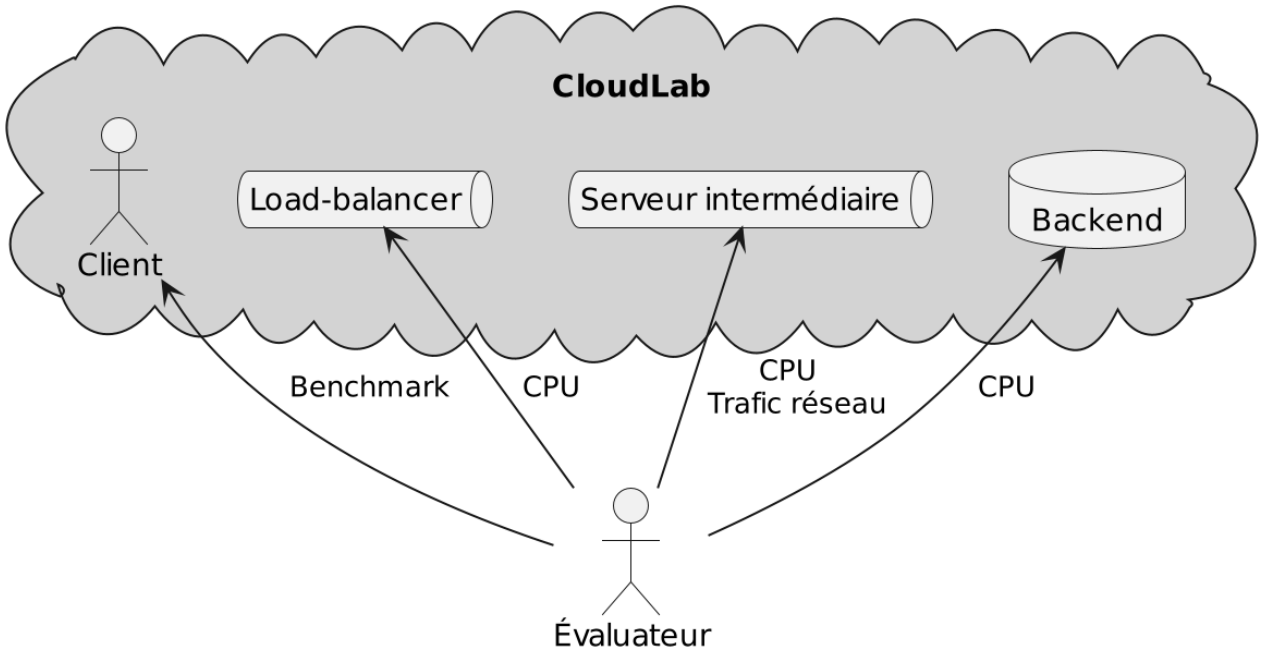
Objectifs

- Constater que les données sont envoyées directement depuis le backend
- Mesurer :
  - La baisse de la charge des serveurs bypassés
  - L'augmentation de la charge du backend liée au chiffrement
  - (le surcoût lié à TLS)

Données à mesurer

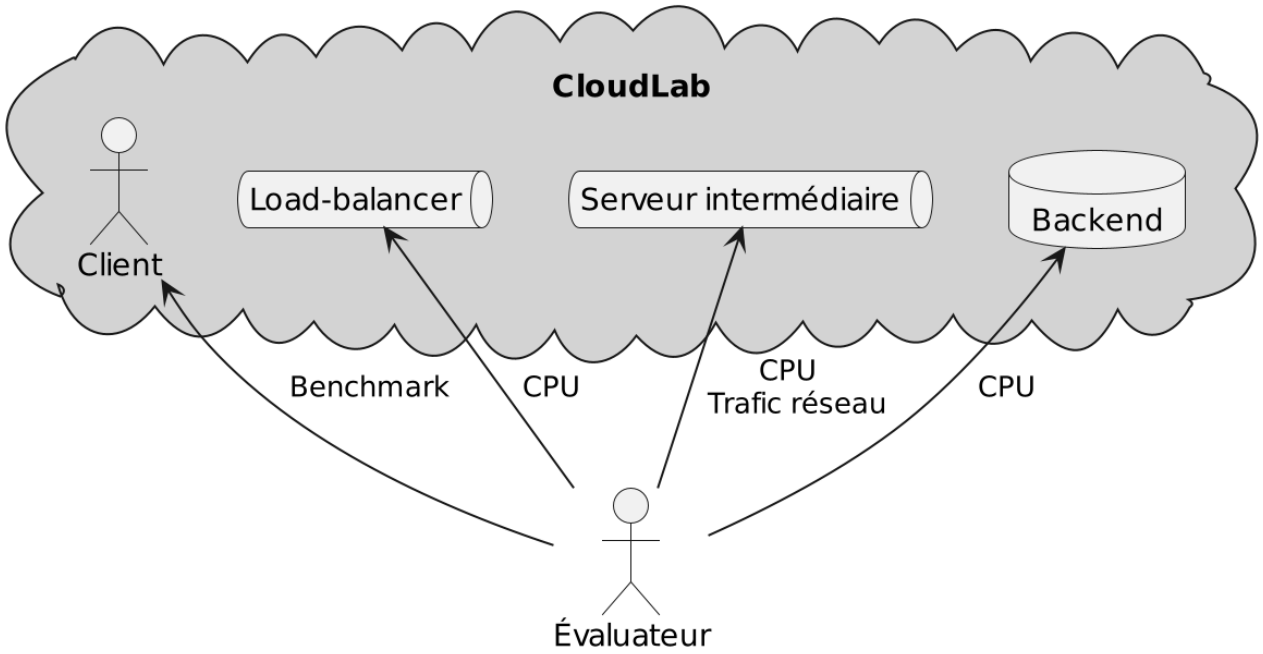
- Load-balancer :
  - Temps de calcul
- Serveur intermédiaire :
  - Temps de calcul
  - Données envoyées sur le réseau
- Backend :
  - Temps de calcul

Mesure des données



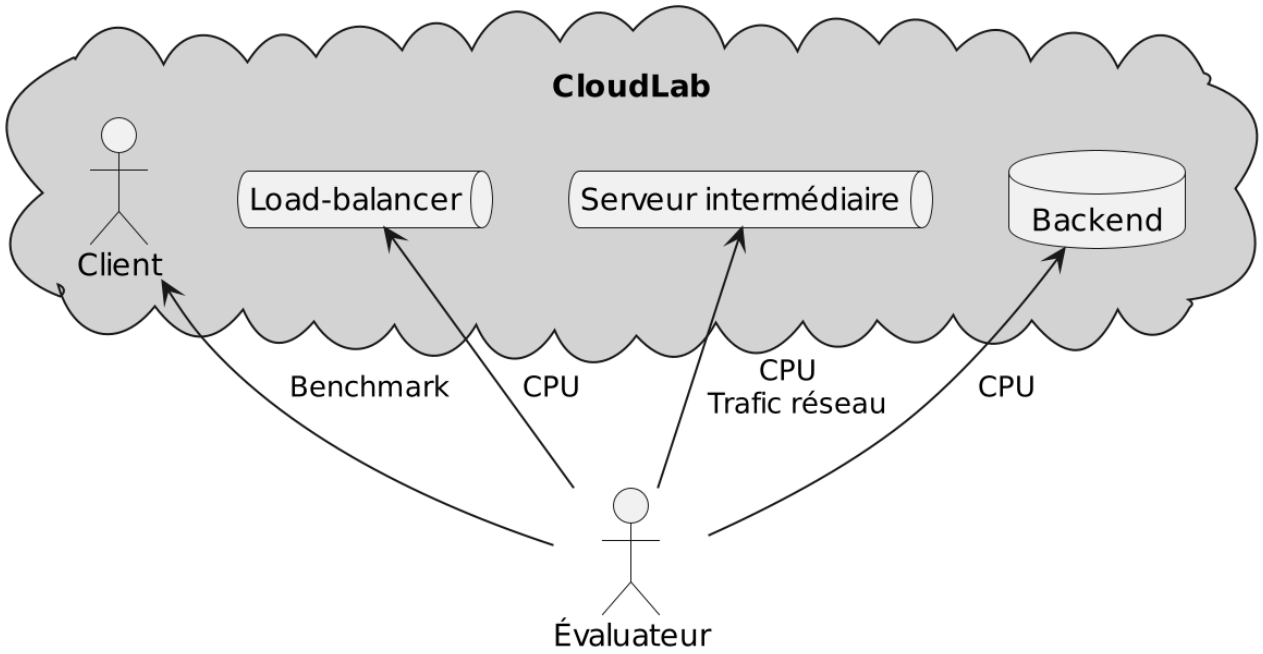
- Script ouvrant une connection SSH sur chaque machine
- Plusieurs centaines de requêtes avec `curl`

Mesure des données



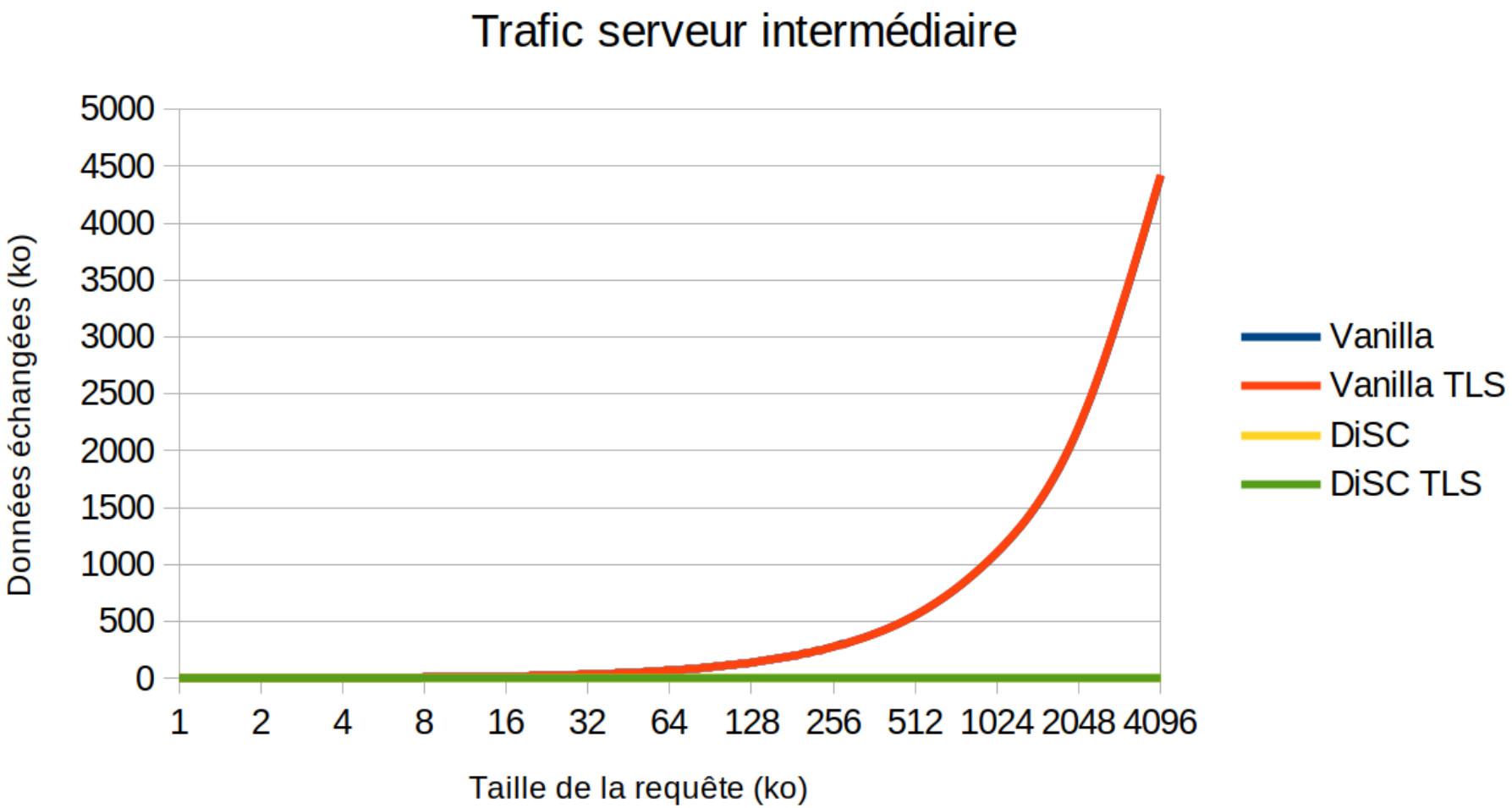
- Script ouvrant une connection SSH sur chaque machine
- Plusieurs centaines de requêtes avec `curl`
- Mesure du temps de calcul :
  - Temps CPU : `/proc/pid/stat`
  - Fréquence du processeur fixée à 800 MHz

Mesure des données

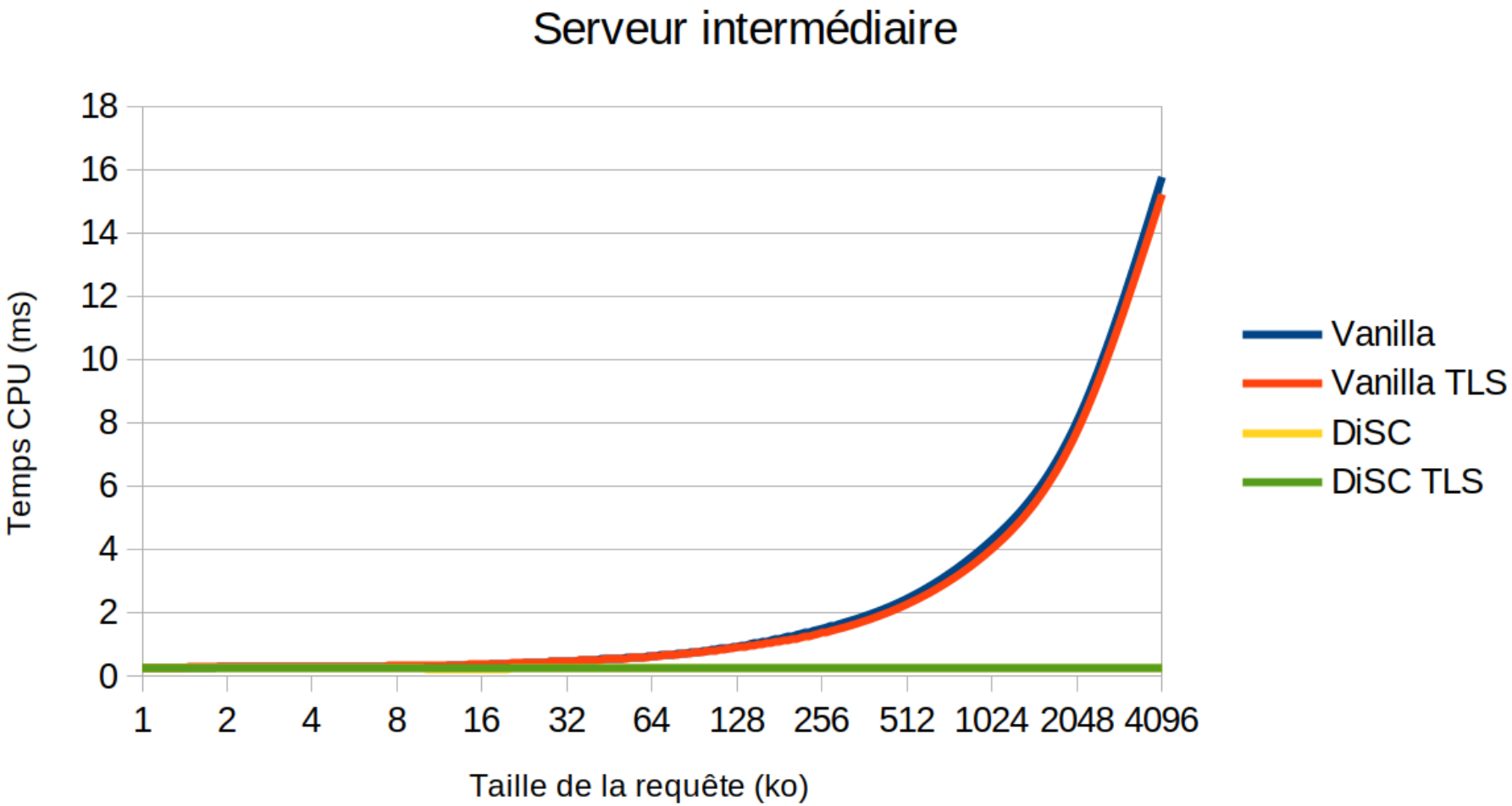


- Script ouvrant une connection SSH sur chaque machine
- Plusieurs centaines de requêtes avec `curl`
- Mesure du temps de calcul :
  - Temps CPU : `/proc/pid/stat`
  - Fréquence du processeur fixée à 800 MHz
- Mesure du trafic réseau : `ifconfig ens1f0`

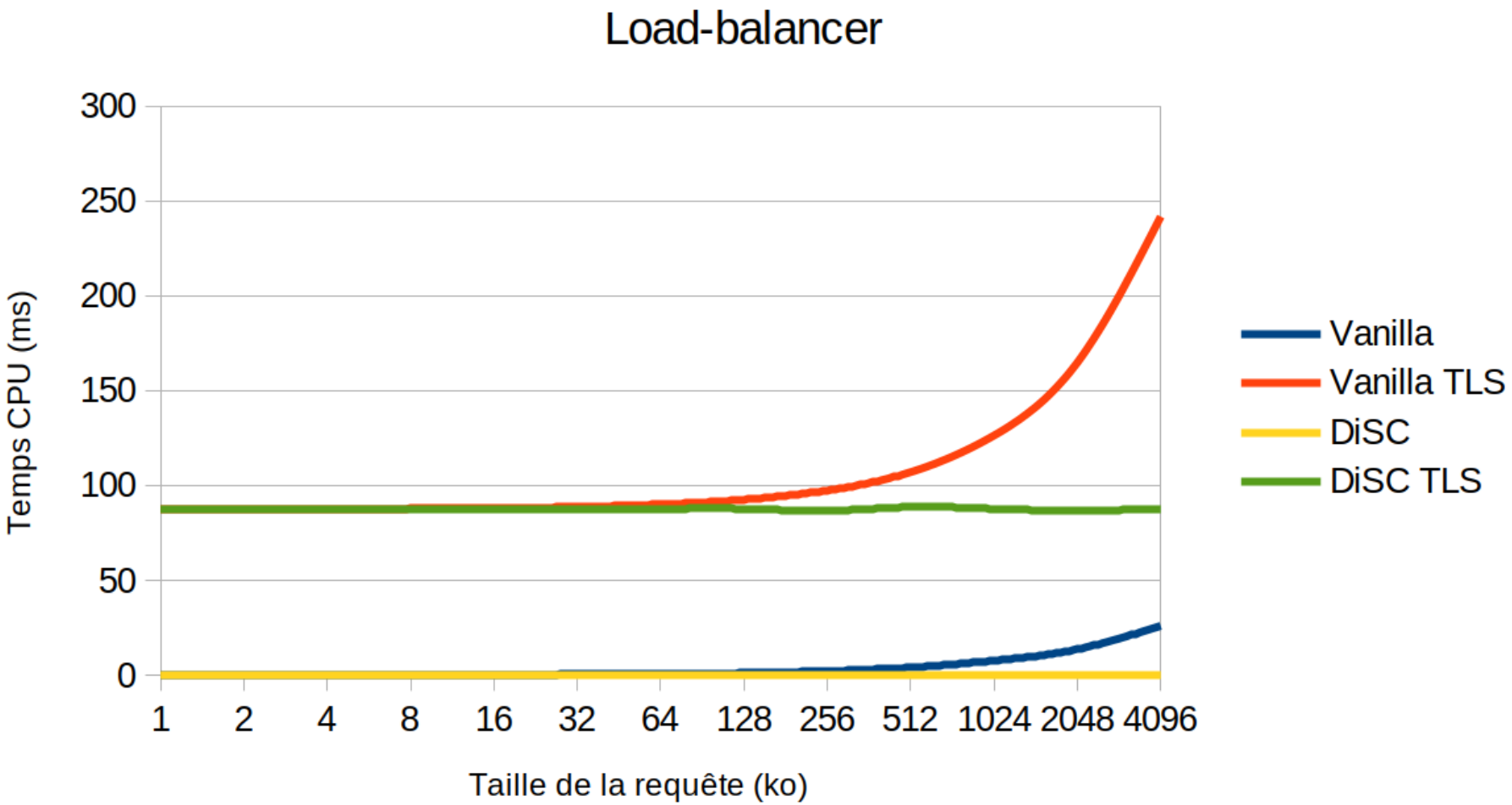
Résultats


Trafic serveur intermédiaire
Données échangées (ko)
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
Taille de la requête (ko)
Vanilla
Vanilla TLS
DiSC
DiSC TLS

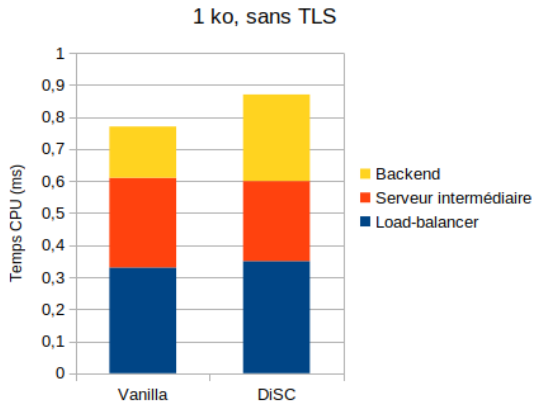
Résultats

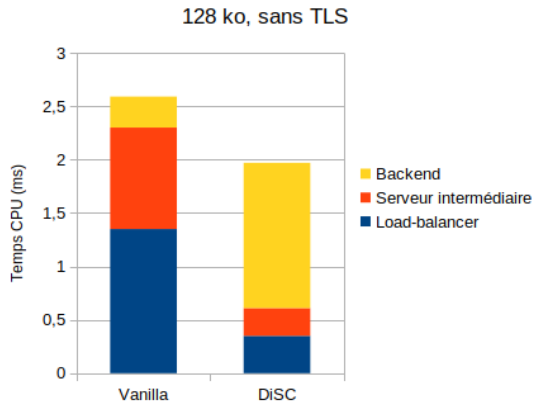

Serveur intermédiaire
Temps CPU (ms)
18
16
14
12
10
8
6
4
2
0
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
Taille de la requête (ko)
Vanilla
Vanilla TLS
DiSC
DiSC TLS

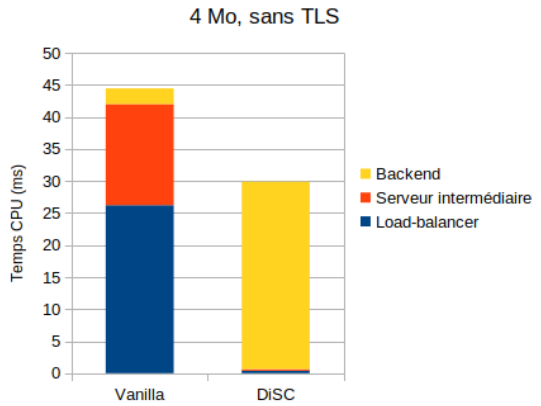
Résultats

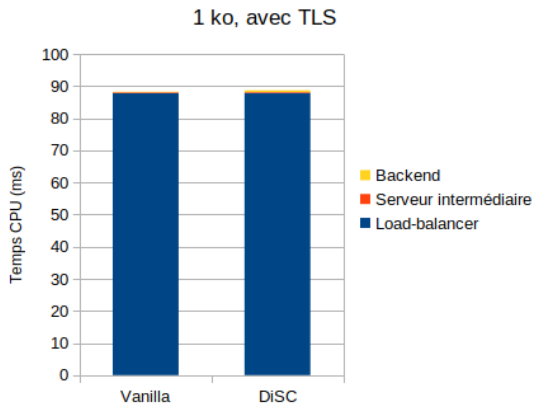

Load-balancer
Temps CPU (ms)
300
250
200
150
100
50
0
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
Taille de la requête (ko)
Vanilla
Vanilla TLS
DiSC
DiSC TLS

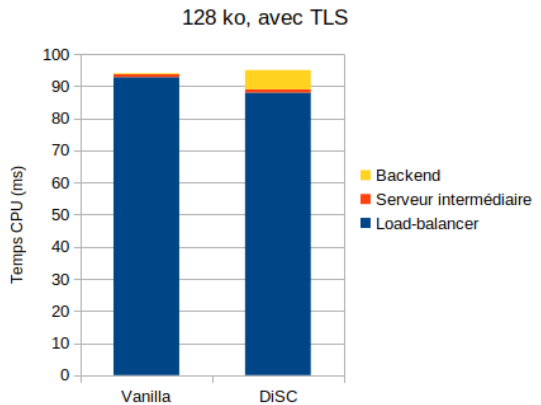
Résultats

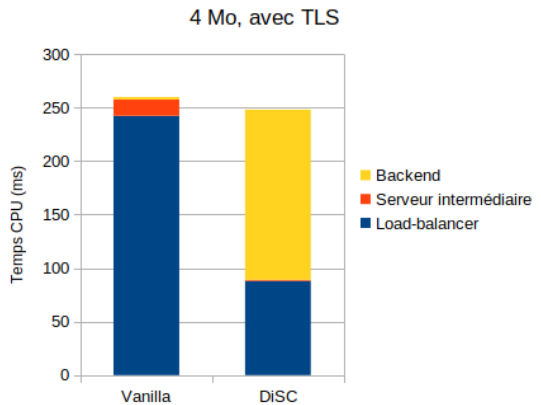

1 ko, sans TLS
Temps CPU (ms)
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
Vanilla
DiSC
Backend
Serveur intermédiaire
Load-balancer


128 ko, sans TLS
Temps CPU (ms)
3
2,5
2
1,5
1
0,5
0
Vanilla
DiSC
Backend
Serveur intermédiaire
Load-balancer


4 Mo, sans TLS
Temps CPU (ms)
50
45
40
35
30
25
20
15
10
5
0
Vanilla
DiSC
Backend
Serveur intermédiaire
Load-balancer


1 ko, avec TLS
Temps CPU (ms)
100
90
80
70
60
50
40
30
20
10
0
Vanilla
DiSC
Backend
Serveur intermédiaire
Load-balancer


128 ko, avec TLS
Temps CPU (ms)
100
90
80
70
60
50
40
30
20
10
0
Vanilla
DiSC
Backend
Serveur intermédiaire
Load-balancer


4 Mo, avec TLS
Temps CPU (ms)
300
250
200
150
100
50
0
Vanilla
DiSC
Backend
Serveur intermédiaire
Load-balancer

Plan

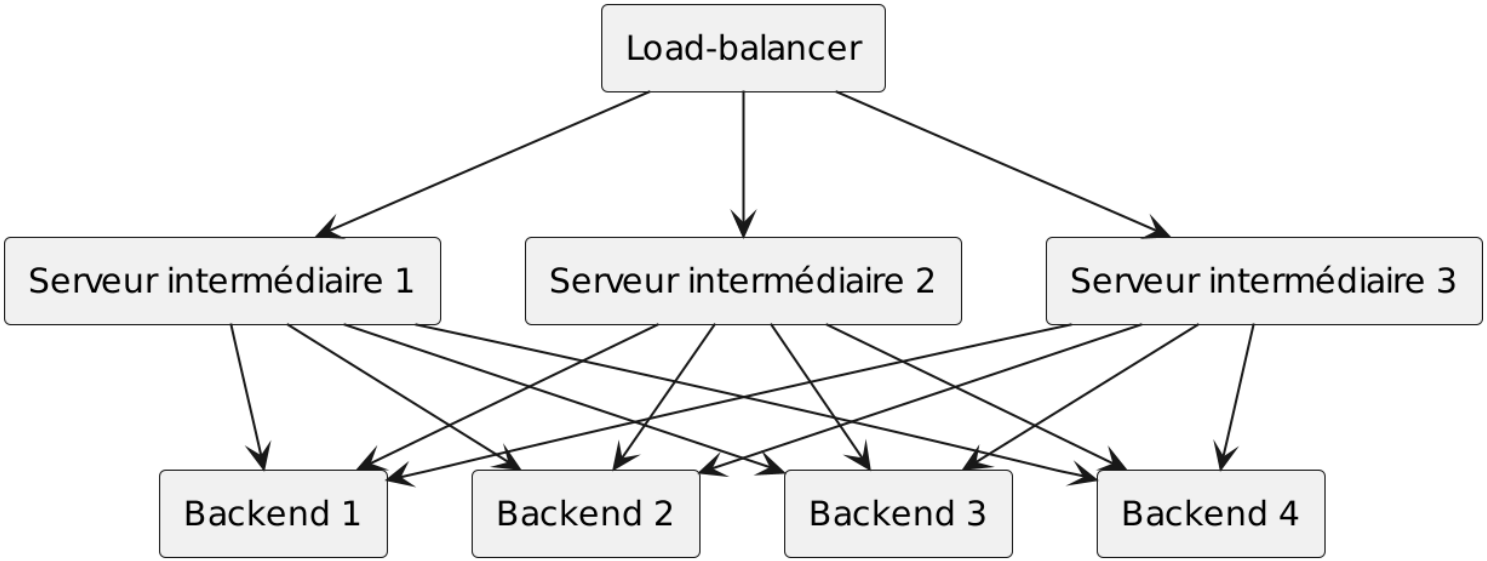
Conclusion

- Mon stage a permis de :
  - Concevoir une méthode de transfert de session TLS
  - L'intégrer à Nginx
  - Évaluer le gain de performance obtenu

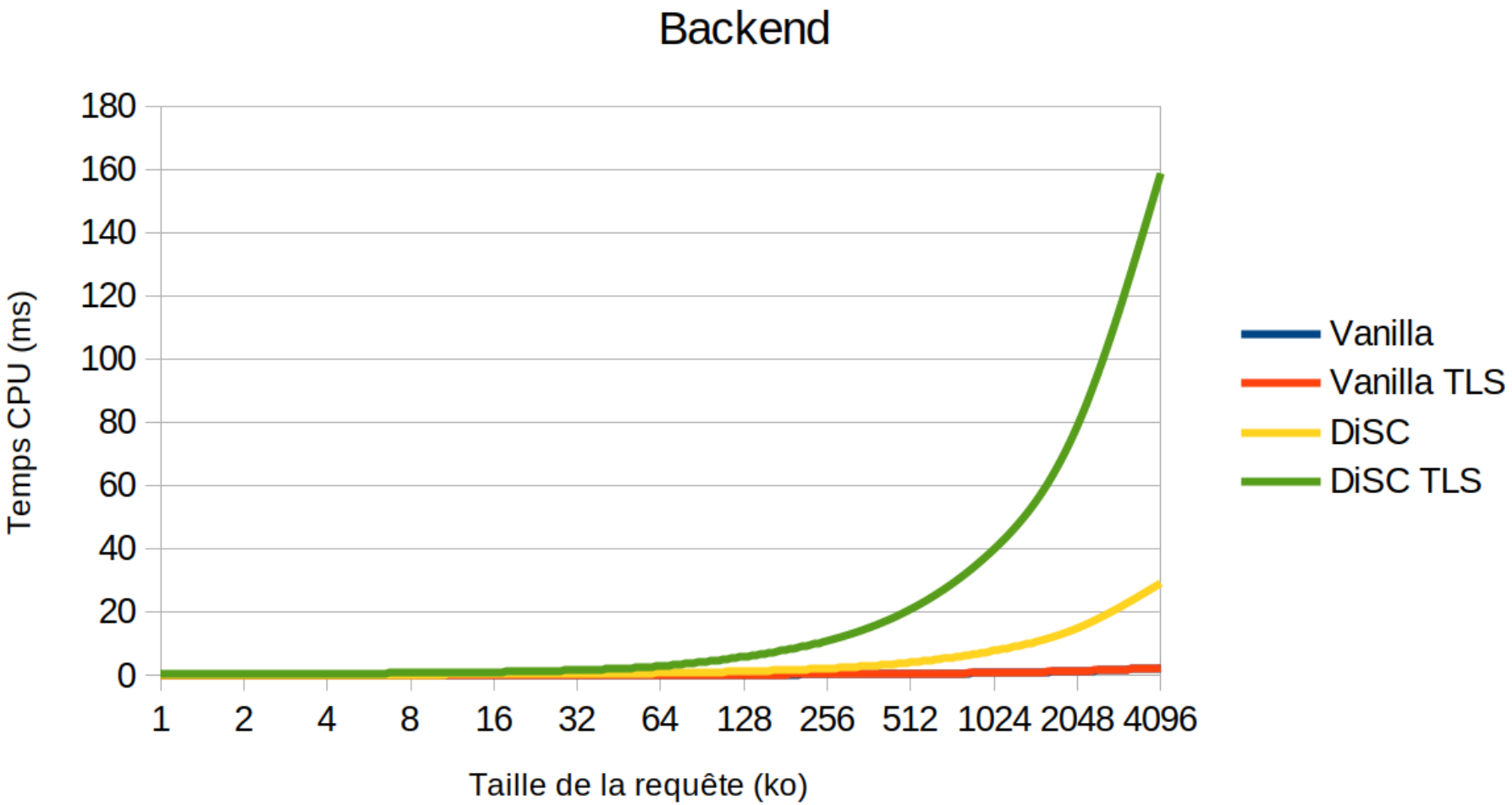
Les protocoles réseaux

Couche application	Message	HTTP, FTP, ...
Couche transport	Segment	**TCP**, UDP
Couche réseaux	Datagramme	IP
Couche liaison	Trame	
Couche physique	Données	

Architecture multitiers



Résultats


Backend
Temps CPU (ms)
180
160
140
120
100
80
60
40
20
0
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
Taille de la requête (ko)
Vanilla
Vanilla TLS
DiSC
DiSC TLS