

DMA – Labo 1

Auteurs : Bastian Cholet, Kevin Ferati

Partie 1

Méthodologie de statistiques

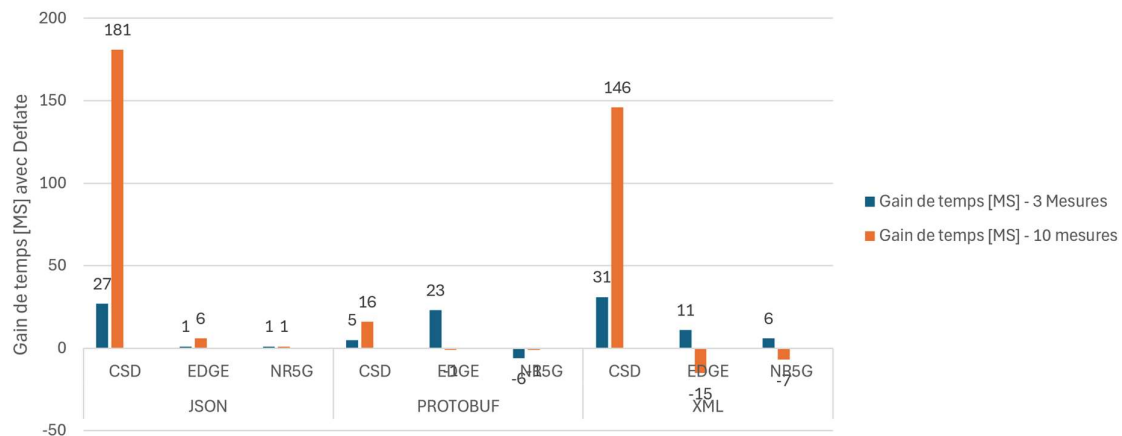
Nous avons testé 3 vitesses de transmission différentes : CSD, EDGE et 5G. Nous avons mesuré le temps de réponse et de traitement en effectuant la moyenne de 10 requêtes pour chaque format (JSON, XML, Protobuf), compression activée ou non, pour chaque réseau ci-dessus. Chaque requête contenait 3, 10, 50 et 100 mesures différentes.

Pour chaque tuple (Format, Réseau, Nombre de mesure, Compression activée), nous obtenons ainsi une moyenne de temps de réponse et de traitement sur 10 requêtes. Ensuite, nous récupérons la différence entre le temps avec deflate ou non d'activé.

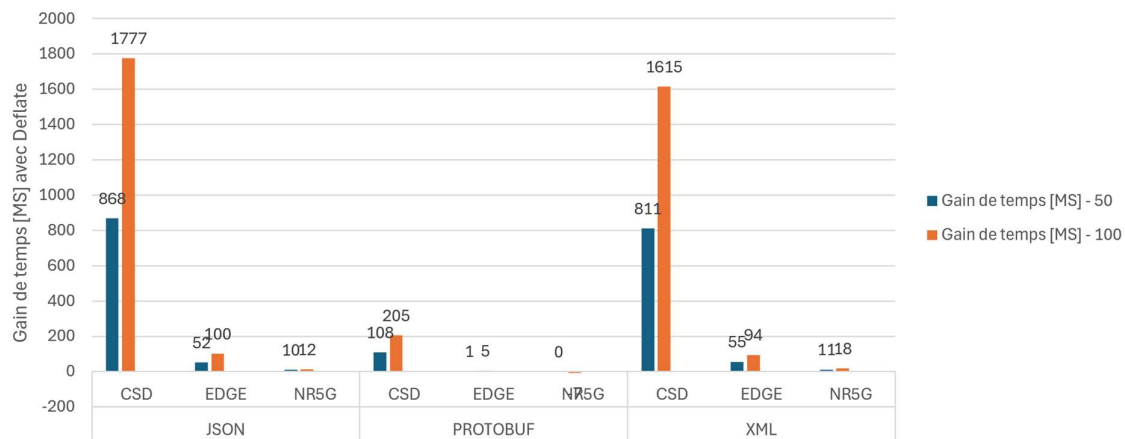
Avant chaque série de mesure, *un* warmup consistant en une requête dans l'API est faite afin de ne pas être biaisé, première exécution étant toujours plus lente-

Mesures et observations

Voici les gains observés pour des petites données :



Voici ceux avec les grandes données (50, 100 mesures):



Observations :

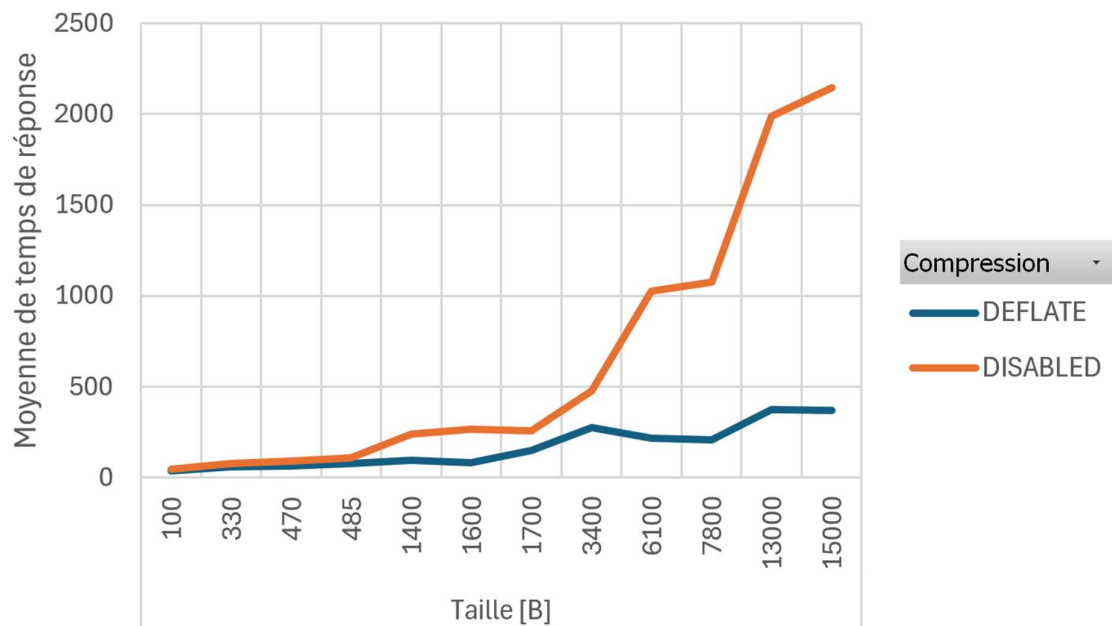
- Plus un réseau est lent, plus la compression devient rentable. Nous pouvons effectivement remarquer que les plus gros gains de temps se font sur le réseau CSD, mais peu dans le cas des réseaux 5G.
- La compression est utile dans le cas de volumes de données importants. Les gains avec protobuf quasiment toujours faibles (car le format est optimisé) mais ceux avec XML et JSON restent conséquents. Dans les volumes faibles, l'*overhead* induit par (dé)compression peut impacter les résultats finaux (par exemple, protobuf pour 3 et 10 mesures).

Nous pouvons nous en convaincre en regardant simplement le tableau suivant :

Réseau et compression	Somme des moyennes des temps de réponse
CSD	
DEFLATE	2006
DISABLED	7796
EDGE	
DEFLATE	638
DISABLED	970
NR5G	
DEFLATE	574
DISABLED	612

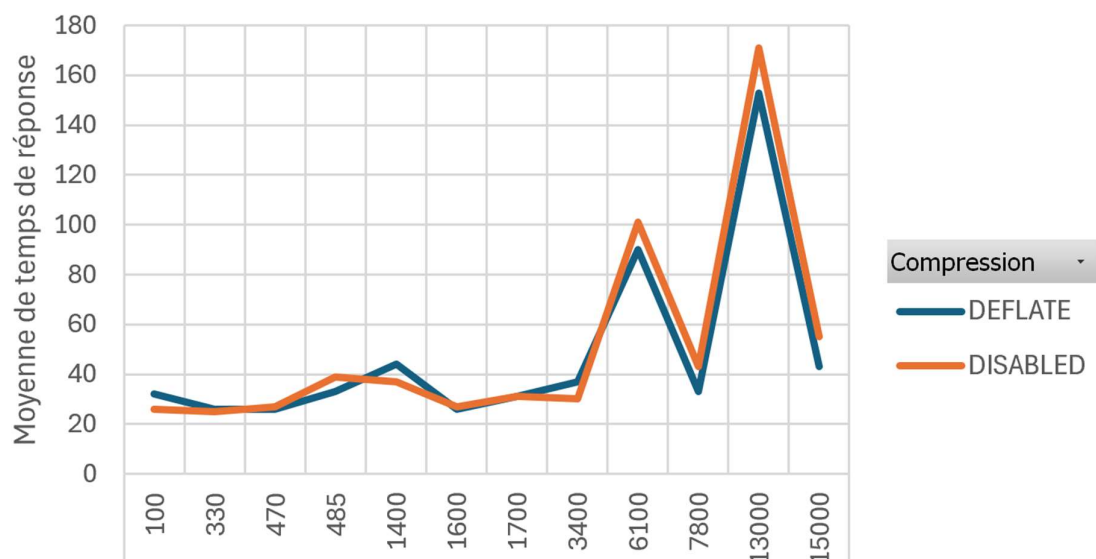
Plus un réseau est lent, plus la compression sera efficace, dans le cas précédent.

Quant à la question de savoir à partir de quand la compression est utile, voici deux courbes comparant, pour une certaine liste de données, le temps d'exécution en MS lors de la compression ou non pour le réseau CSD :



Dans ce réseau, nous pouvons observer qu'il existe déjà un gain de temps aux alentours de 500B. La vraie différence se fait aux alentours de 3KB.

Voici le même schéma pour la 5G :



La très haute vitesse compense la compression ou non. Il faudrait tester avec des tailles beaucoup plus grandes, afin de s'en assurer.

Partie 2.1

La récupération de l'ensemble des auteurs ou de leurs livres est problématique dans un contexte mobile, notamment lors d'une connexion internet de faible débit. La requête de tous les auteurs est trop volumineuse et devraient plutôt intégrer un système de pagination pour segmenter la taille de la réponse.

Une autre possibilité serait la mise en cache de ces requêtes. On pourrait imaginer n'effectuer la récupération ou la mise à jour des auteurs que lorsque le téléphone accède à un point d'accès wifi.

Partie 3.1

Il permet d'identifier l'appareil sur lequel l'app est installée. Il est généré à l'installation, la réinstallation et la mise à jour de l'app.

Dans le cadre d'une app comme WhatsApp, un token est généré à l'installation de l'app sur chacun des appareils où l'app est installée. Ce token permet ainsi d'identifier et de rediriger le message vers les bons appareils.

Puisque chaque appareil va générer un token, il convient de les stocker dans une base de donnée sur le serveur. Ainsi, dès la réception d'un message, le serveur pourra redistribuer la notification à l'ensemble des appareils. Il convient bien sûr de maintenir cette DB à jour notamment lorsque l'app est réinstallée sur un appareil déjà enregistré.