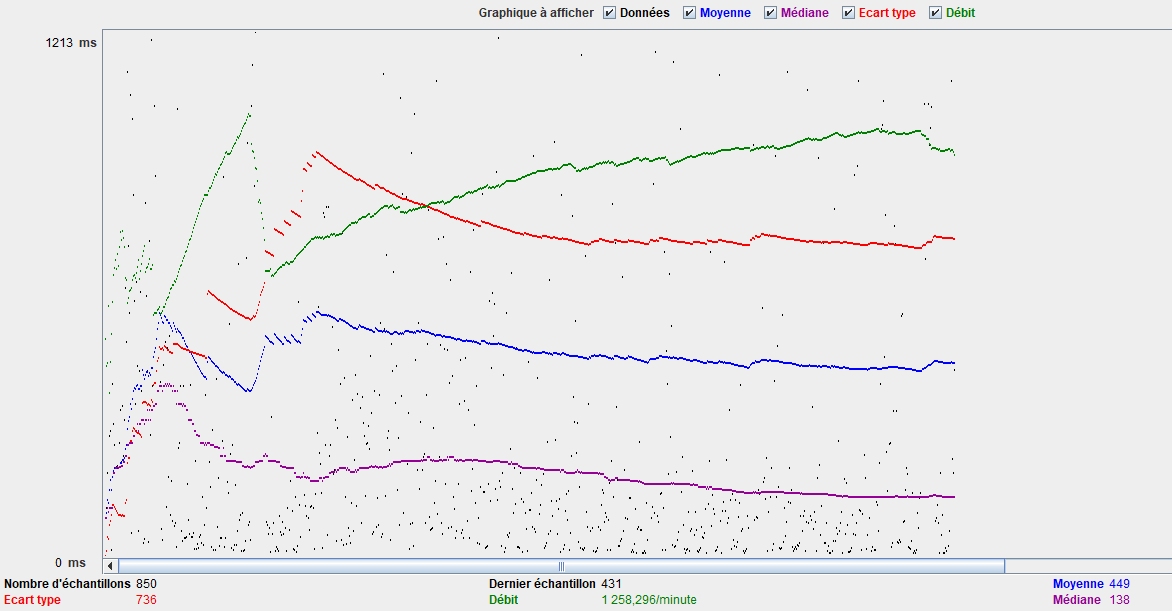
Tests de charge

Afin de tester le site Gizza, on a utilisé l’outil d’injection de charge Apache Jmeter.

On a commencé par établir un scénario fonctionnel qui se déroule comme tel : l’utilisateur commence par charger la page d'accueil, s’authentifie, sélectionne une pizza et enfin valide sa commande. Ensuite, on a établi 2 scénarios de tir de charge : le premier avec 10 utilisateur qui exécutent 5 itérations avec une monté en charge de 1s, le second avec 100 utilisateurs qui exécutent 50 itérations avec une monté en charge de 30s.

le résultat du premier tir de charge est représenté dans le graphe suivant :

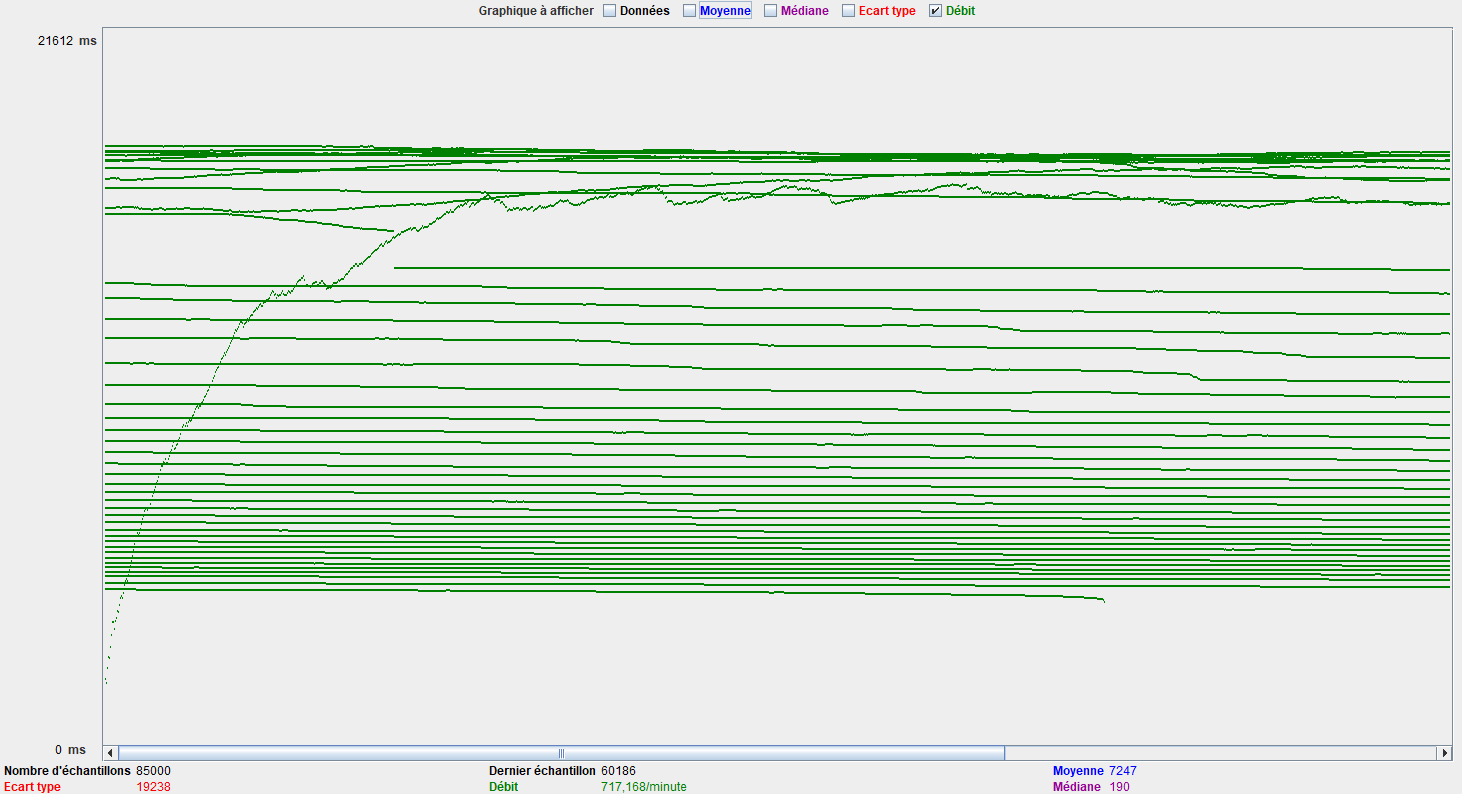
Graphe de résultats du tir de charge 10x5

on constate que le débit augmente progressivement avec l’augmentation du nombre d’utilisateurs et qui se stabilise sans baisse brutal, même chose pour la moyenne du temps de réponse (qui est en Milliseconde) qui augmente et ensuite se stabilise. Le test a duré 3 minutes.

Les résultats du deuxième test sont présentés dans le tableau et les graphe suivant:

le tableau représente les résultats consolidé, on peut distinguer le nombre total de requêtes qui a été réalisé qui s'élève à 85000 avec un taux d'erreur de 5.15% sur la totalité mais un taux de presque 40%.

les graphe suivants montrent le débit ainsi que le temps de réponse moyen. Les deux courbe évoluent de manière proportionnelle. Le débit augmente avec l'augmentation du nombre d'utilisateurs, au bout de quelques itérations la courbe fait une cassure qui correspond au début de l’apparition des erreur. on peut le constater même sur le graphe du temps moyen de réponse.

La cause des erreurs a étés identifié. il s’agit d’une relation ManyToMany qui charge a chaque fois les commandes reliés au produit et donc la taille de l’objet augmente avec l’augmentation du nombre de commande. Cette erreur a été corrigé par la suite. 

Graphe du debit



Graphe de la moyenne du temps de réponse

