

Truc - Machin

Performances - Simulation d'un Serveur

Jean-David Fiquet

Louis Pequignot

21 novembre 2015

Sommaire

I	Description du Simulateur	3
1	Simulation d'un serveur	3
2	Architecture	3
II	Analyse des résultats	4

I Description du Simulateur

1 Simulation d'un serveur

Le but de ce TP est d'implémenter un serveur web. Plus particulièrement de simuler un flux de client se connectant au serveur et d'analyser la réaction du temps de réponse en fonction de la fréquence des clients.

Le cycle de vie d'un client est décrite ainsi :

- Un client envoie une requête de type 1 au serveur.
- Le serveur reçoit la requête et la traite. La durée du traitement varie de manière uniforme entre 0ms et 30ms. Il envoie ensuite la réponse.
- A la réception de la réponse, le client envoie une requête de type 2 au serveur.
- Le serveur reçoit cette deuxième requête et la traite. La durée du traitement de la requête 2 suit une répartition exponentielle de paramètre $1/10 \text{ ms}^{-1}$. Il envoie ensuite la réponse.
- Le client se déconnecte dès qu'il reçoit la réponse.

Le délai de communication est fixé à 6ms entre le client et le serveur. Les clients se connectent selon un processus de Poisson de paramètre λ .

On note que le temps moyen de réponse est au moins égale à 49ms. Les délais de communication sont fixés pour les 4 messages envoyés. S'ajoute ensuite les moyennes des deux calculs. La requête 1 dure en moyenne 15ms. La requête 2 dure en moyenne 10ms.

$$4*6 + 15 + 10 = 49\text{ms}$$

2 Architecture

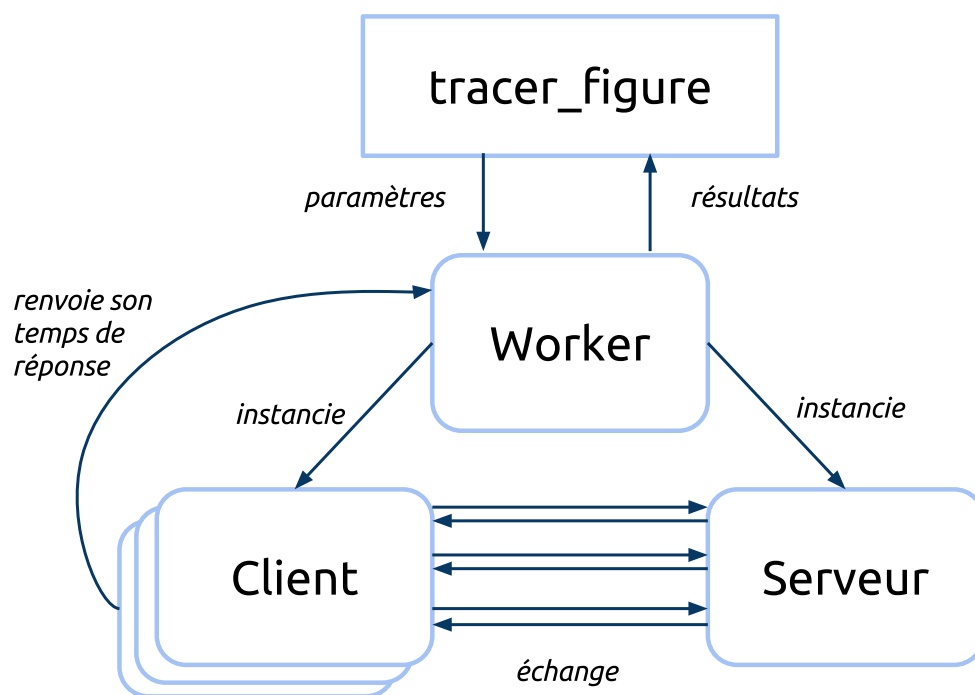


FIGURE 1 – Architecture du simulateur

Le simulateur est programmé en **Ruby**. Il est composé de trois objets et d'un script. Le script *tracer_figure* est lancé pour effectuer les calculs et tracer les courbes, il est exécuté et s'appuie

sur les objets Ruby pour calculer les temps de réponses. Le script instancie un objet *Worker* qui va se charger de gérer les clients et le serveur. Le *Worker* nécessite des paramètres en entrée : Le nombre de clients et le paramètre λ du processus de Poisson pour définir la fréquence de connexion des clients. Le *Worker* va ensuite instancier le *Serveur* qui recevra les requêtes des *Clients* et leur renverra en réponse le temps écoulé par son calcul. Le *Worker* va aussi instancier plusieurs *Clients* en fonction du nombre d'échantillon voulu et les connecter au *Serveur* à des temps dépendant du processus de Poisson. Une fois les échanges entre *Clients* et *Serveur* terminés, chaque *Client* va renvoyer son temps de réponse.

II Analyse des résultats