

Rapport TER

Plate-forme de calcul distribué en JavaScript

Taoukilite Ahmed El Mahdi

April 3, 2017

1 Présentation du projet

1.1 Contexte

Le projet consiste à concevoir une plateforme de calcul distribuée, basé sur le volontariat, et qui permettrait au chercheurs d'avoir la puissance calculs de nombreux ordinateurs personnels dans le monde entier. Cela est fait grâce à l'exploitation des navigateurs internet des volontaires qui joueront le rôle des workers en exécutant du code javascript qui reçoivent de la plateforme. Les calculs seront effectués sur les machines volontaires et les résultats seront renvoyés à la plate-forme.

1.2 Objectifs

L'objectif de ce projet est de développer en premier temps une interface web qui permettra dans un premier temps de lancer un calcul avec un nombre de jobs, à partir d'un code javascript saisi par le chercheur (Figure 1). En effet un chercheur accèdera à l'interface web puis saisira le code des jobs donc le code qui sera exécuté sur chaque volontaire qui participe au calcul, la plate-forme créera le nombre de jobs choisis par le chercheur en envoyant des messages contenant le code javascript et l'id du job (Correlation id) aux serveurs de file de messages.

La plateforme contiendra une deuxième interface web qui est l'interface volontaire (Worker), cet interface jouera le rôle de l'exécuteur en effet le volontaire se rendra sur le site de la plateforme, puis ainsi commencera à participer au calcul en consommant les jobs créés par le chercheur. Ces jobs sont extraits du serveur de la file de messages par le côté serveur de la plateforme ensuite ces jobs sont envoyés aux volontaires (Workers) qui décortiqueront le message en retrouvant le code et l'id du job, puis le code javascript extrait sera évalué puis exécuté en fonction des paramètres en entrées (l'id du job) ou pas.

Un volontaire participe au calcul dès qu'il se connecte sur l'interface volontaire de la plateforme, et se déconnecte quand il ferme l'interface.

2 Fonctionnement

2.1 Acteurs

La plate-forme (Figure 2) contient 4 types d'acteurs:

- Un acteur de type serveur web (type NodeJS) pour gérer les chercheurs et les volontaires, qui effectue les opérations suivantes:
 - Interactions avec l'interface chercheur :
 - * Récupération du code JavaScript du calcul + le nombre de jobs à créer.
 - * Création et génération des jobs d'un calcul.
 - * Dépôt de ces jobs dans la file de messages dédiée aux jobs.
 - * Récupération des résultats des jobs à partir de la file des résultats (un calcul se termine dès que les réponses de tous les jobs sont reçues).
 - Interactions avec l'interface volontaire :
 - * Récupération des jobs à partir de la file des jobs.

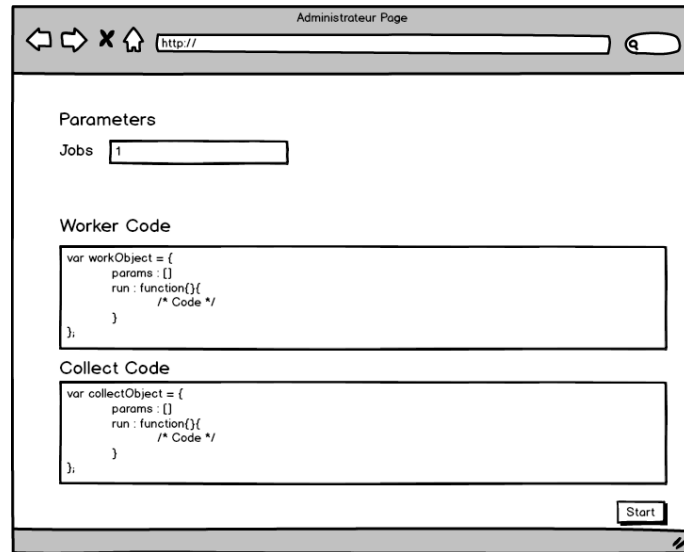


Figure 1: Prototype d'interface chercheur

- * Envoie des jobs aux volontaires.
- * Récupération des résultats du job exécuter par le volontaire.
- * Dépôts des résultats dans la file des résultats
- Un serveur web (type NodeJS) pour gérer les chercheurs, qui effectue les opérations suivantes:
 - *
 - * Génération et dépôt des jobs à la file des jobs.
 - * Récupération des résultats du calcul a partir la file des résultats.

2.2 Architecture de la plate-forme

2.3 Annexe

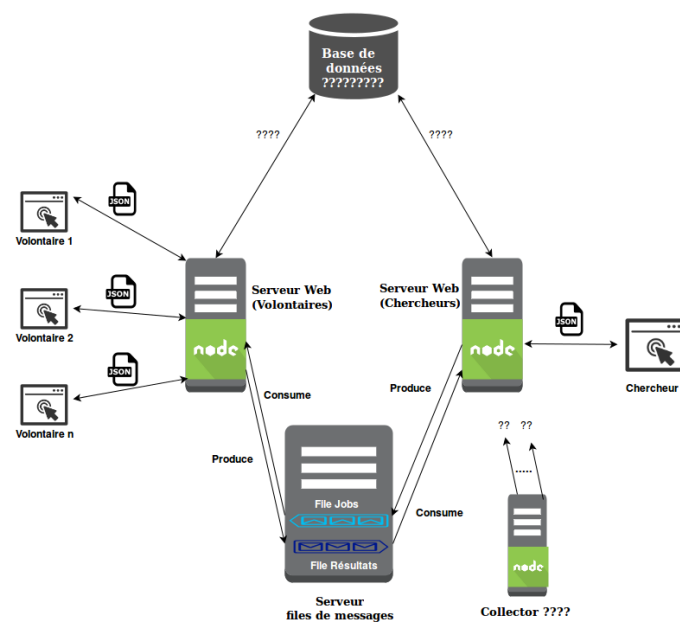


Figure 2: Architecture générale de la plate-forme