



# RAPPORT TER

## Plate-forme de calcul distribué en JavaScript

Taoukilite Ahmed El Mahdi

April 7, 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 de lancer un calcul avec un nombre de jobs, à partir d'un code javascript saisi par le chercheur. 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) au serveur de file de messages.

La plateforme contiendra une deuxième interface web qui est l'interface volontaire (Worker), cette 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 d'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

La plate-forme contient 4 types d'acteurs:

- Un serveur web 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 dédié au jobs.
    - \* Récupération des résultats des jobs à partir la file des résultats (un calcul se termine dès qu'on recoivent la totalité des résultats des jobs).
  - Interactions avec l'interface Volontaire :
    - \* Récupération des jobs à partir de la file des jobs.
    - \* Envoie des jobs aux volontaires.
    - \* Récupération des résultats du job exécuter par le volontaire.
    - \* Dépôts des résultats dans la file des résultats
- Un serveur file de messages :
  - Se rendre sur le site de la plateforme.
  - Remplissage du formulaire du calcul.
  - Lancement du calcul.
- Un ou plusieurs Volontaires : Navigateurs Web.
  - Se rendre sur le site de la plateforme
  - Connexion a la plateforme
  - Récupération des jobs
  - Décortiquage des jobs: récupération de l'id, récupération du code Javascript.
  - Exécution des jobs, renvoie des resultats
- Un ou plusieurs Chercheurs : Navigateurs Web.
  - Se rendre sur le site de la plateforme.
  - Remplissage du formulaire du calcul.
  - Lancement du calcul.

Administrateur Page

Parameters

Jobs

Worker Code

```
var workObject = {
  params : []
  run : function(){
    /* Code */
  }
};
```

Collect Code

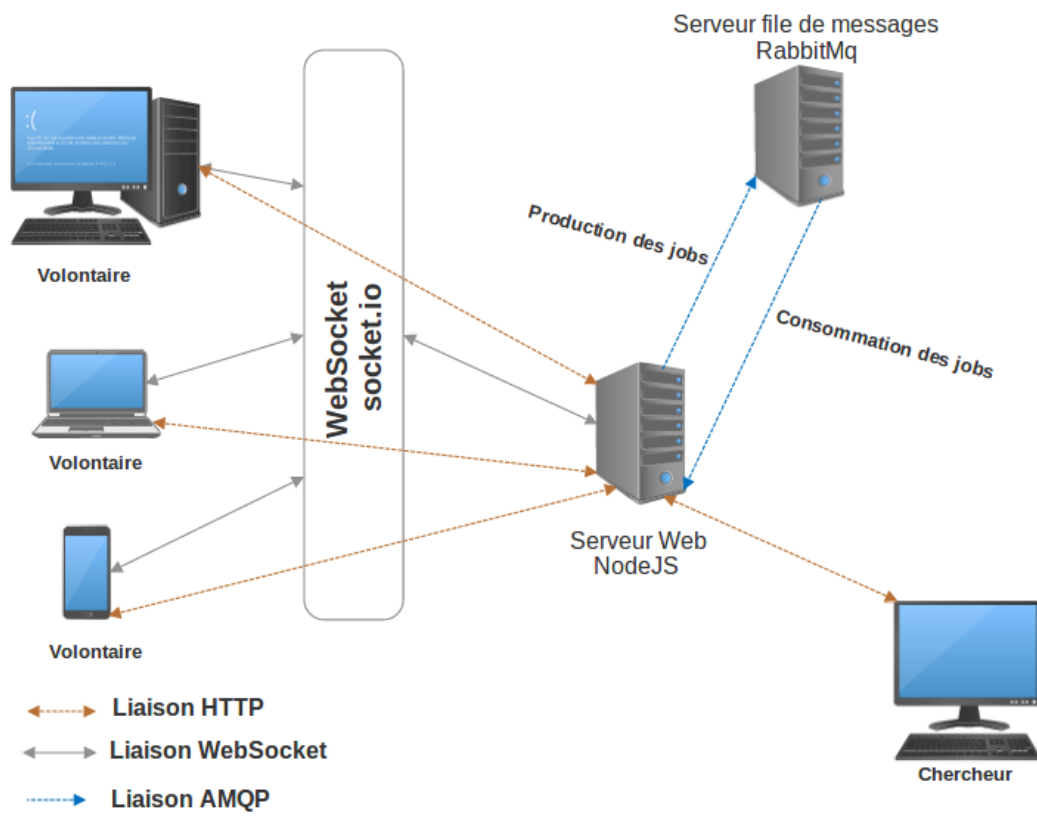
```
var collectObject = {
  params : []
  run : function(){
    /* Code */
  }
};
```

Start

### 3 Technique

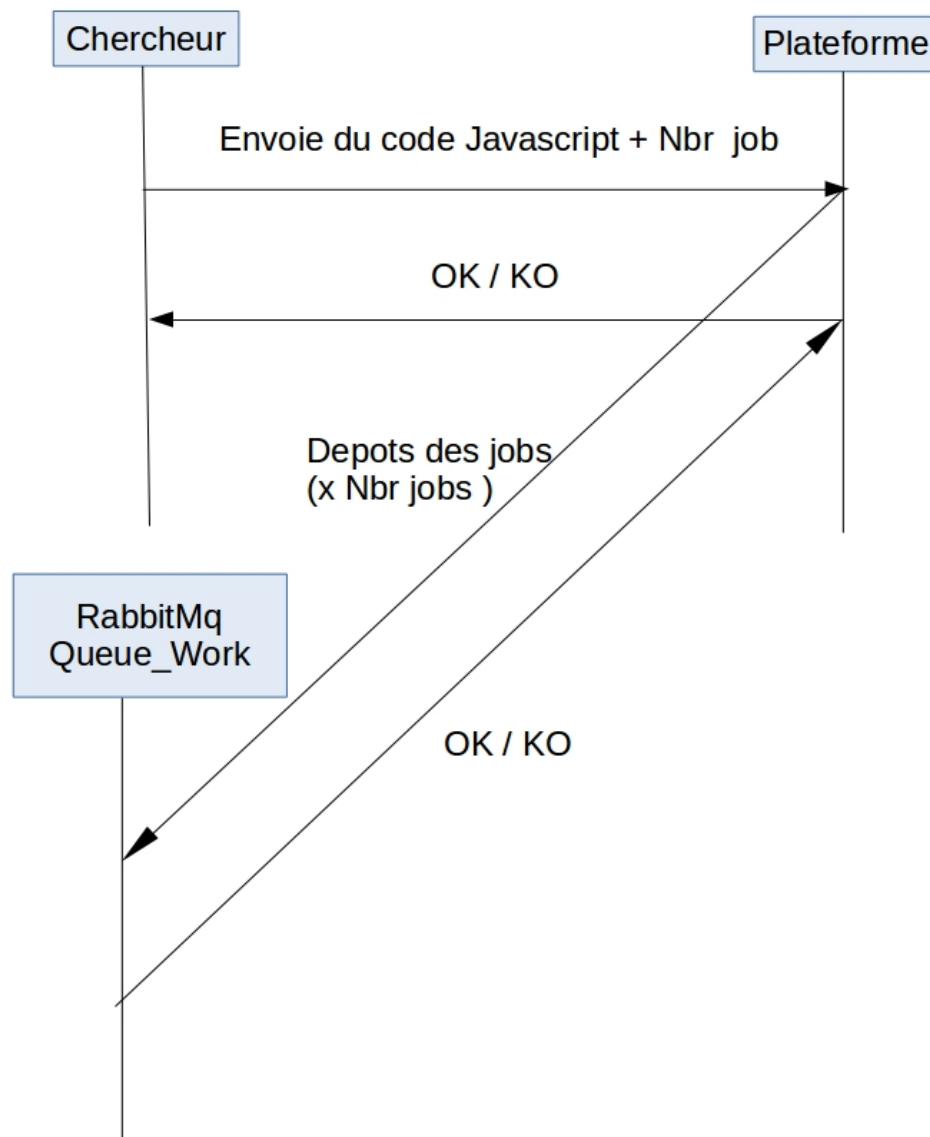
La plateforme gère des calcul distribué qui seront fait en Javascript, pour cela on a opté pour le Javascript dans le coté serveur a fin qu'il y une compatibilité entre langages Front-end et Back-end. La plateforme utilise les technologies suivantes :

- Back-end : On utilise NodeJS comme serveur web, ensuite Express qui est un framework qui permet de développer des application web en Javascript plus rapidement, il permet aussi de gérer les routes URL ainsi que les requetes HTTP/HTTPS.
- Front-end : les interfaces chercheur et volontaire sont conçus en HTML/Javascript/CSS, pour l'interface Chercheur les communication sont faite en HTTP, parcontre dans l'interface Volontaire la premier requete pour se rendre sur le site de la plateforme se fait par HTTP classic, puis pour toute les communication prochaines en bascule vers les websockets.
- Middleware : L'utilisation d'un middleware type serveur de file de messages, ce serveur permettra de stocker les jobs créer, et aussi les résultats des jobs reçus de la part des volontaires, il permettra aussi d'avoir une bonne scalabilité de notre application en ayant la possibilité d'ajouter des instance du serveur web pour augmenter les performances de la plateforme.

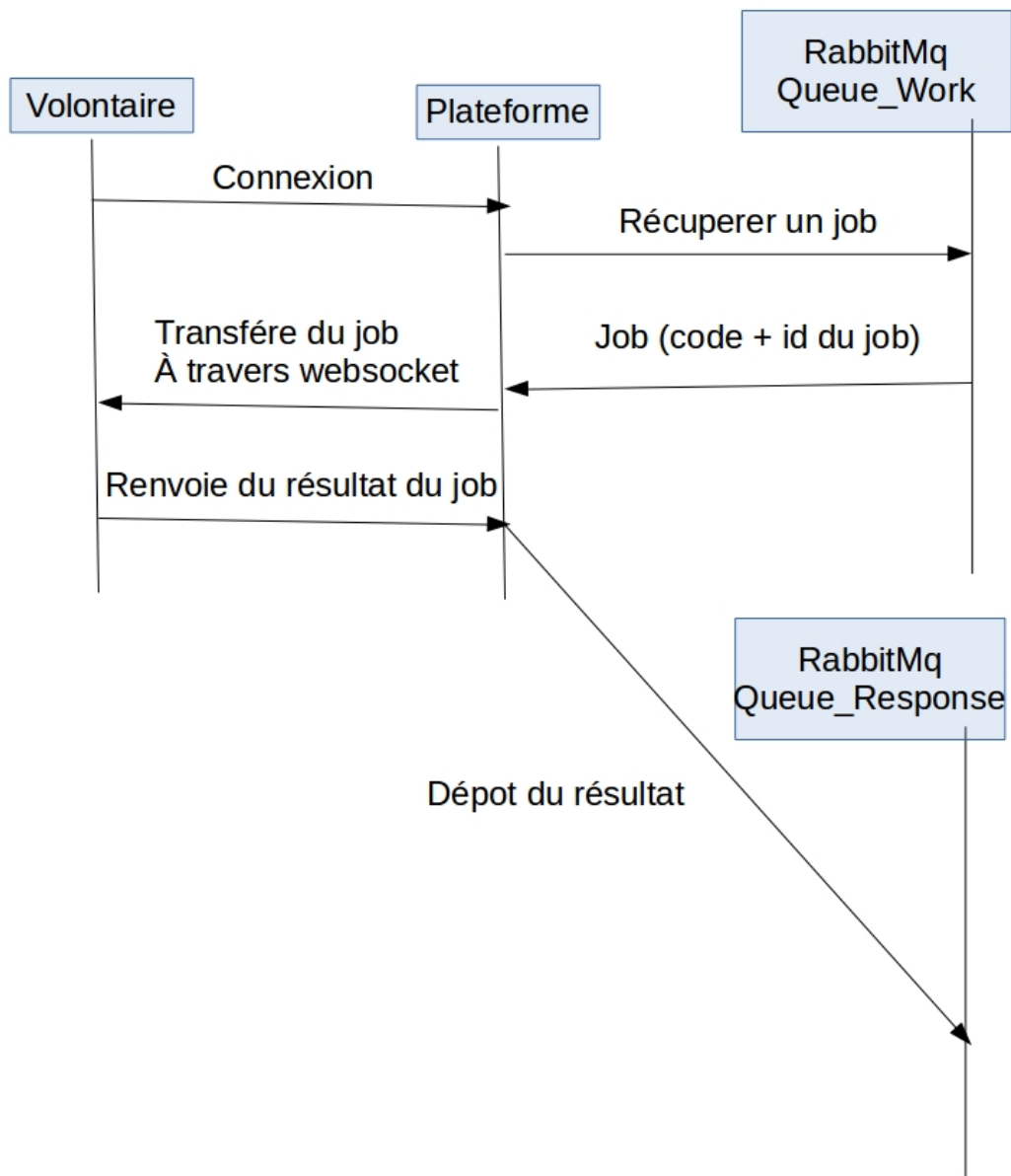


## 4 Scénarios, Plan et Execution

### 4.1 Creation et depots des jobs



## 4.2 Envoie des jobs et récupération des résultats



## 4.3 Execution

Pour lancer la plateforme il suffit de lancer le script WWW avec la commande `./WWW` (`chmod 755 WWW`), ce script s'occupe de lancer le serveur et puis il se met en attente sur le port 3000, donc pour se rendre sur l'interface chercheur il faut visiter l'URL <http://localhost:3000/calcpnl>.

Pour l'interface volontaire qui permet de participer au calcul, il faut se rendre sur l'URL <http://localhost:3000/>, cette interface est une page blanche, mais qui communique avec la plateforme en continue et en temps réel grâce au websockets, en effet cette page reçoit des jobs les exécute ensuite elle renvoie le résultat et ainsi de suite jusqu'à qu'il n'y a plus de jobs.

#### 4.4 Plan

