

Roussel Gildas
Qiu Lingling
Guerry Camille
Lasselin Henri
Griet Thibaud

Rapport technique

Ce rapport comprend tous les aspects techniques du projet Chaosmonkey fait par le groupe J2emépa3.

Pour l'envoi de requête, nous utilisons *Ajax* afin de communiquer avec le serveur Tomcat. Pour cela nous avons implémenté des fonctions *request* dans nos fichiers *Javascript* (*monitor.js* et *simulator.js*). Ajax envoie la requête vers le côté serveur avec des paramètres. Le serveur traite les paramètres et les renvoie au client. Ce dernier les récupère afin de modifier son contenu partiellement (seules les données à modifier sont traitées).

Pour toute requête au serveur, hormis la première connexion du client, nous utilisons cette méthode. Par exemple, nous utilisons cette méthode pour simuler des pannes via les formulaires de la page *simulator* ou encore pour récupérer les données issues de la base de données *Sql*.

Nous utilisons également cette méthode pour le rafraîchissement de la page.

Cependant pour le rafraîchissement instantané de la page *monitor*, nous avons envisagé plusieurs moyens d'y parvenir :

1/ Le client envoie sans cesse des requêtes au serveur qui lui répond s'il y a eu du changement ou pas (panne ou non) : il répond dans tous les cas. Bien que cette méthode soit fonctionnelle, le problème réside dans l'occupation excessive du processeur (la plupart du temps inutile).

2/ Le client envoie une requête avec un long délai. Lorsque le serveur a un message à envoyer au client, il y répond. A l'expiration ou au traitement de la requête, une nouvelle requête est envoyée. Le problème est que le serveur doit mettre en attente la requête.

3/ On utilise alors un Websocket. Un canal bidirectionnel est créé entre le client et le serveur lors de la connexion du client. Ils peuvent envoyer des messages quand ils veulent : dans notre cas, lorsqu'une nouvelle panne est ajoutée. Nous avons pris cette solution car elle permet de ne solliciter le serveur seulement lorsqu'il y a un message.

Pour la génération du graphe, nous avons utilisé le *Javascript* avec *Canvas*. La difficulté est de trouver une taille idéale de page qui s'adapte.