

Supercalculateur mobile distribué

Par Jérémie Brault, Alexandre Brulé, Florian Fichant Le Fur, Pierre Serrut et Hugo Vandenbunder.

Keywords

Supercalculateur, mobile, système distribué

Technologies et Frameworks utilisés

Android, Android Studios, Java ou C#, SQL, et JSON

Modèle économique

Le client paye un prix selon le temps de calcul nécessaire, ainsi qu'un surcoût associé à l'utilisation de l'application et du réseau. Les pairs ayant participé à un calcul fini reçoivent un paiement au prorata du temps de calcul délégué et de la quantité de données traitée.

Fonctionnement général de l'application

Un utilisateur installe l'application, et à son lancement il rejoint le réseau centralisé par un serveur, qui maintient une liste de pairs disponibles. Cet entretien se fait grâce à des requêtes périodiquement envoyées par un utilisateur au serveur pour lui signaler qu'il est toujours disponible. Autrement, celui-ci peut lui indiquer qu'il souhaite se déconnecter.

Un utilisateur peut envoyer une demande de calcul au serveur. Celui-ci s'occupe de fragmenter le calcul, et d'envoyer une sous-partie à autant de pairs que nécessaire. Ceux-ci exécutent le sous-calcul reçu, et renvoient leur sous-résultat au serveur. Ce dernier recompose alors le résultat final et l'envoie à l'utilisateur ayant fait la requête initiale.

Un utilisateur recevant une demande de calcul n'est pas prévenu autre que par une icône dans sa barre de statut et n'a pas accès aux détails du calcul. Cependant, s'il souhaite se retirer du réseau au cours d'un calcul, il peut notifier le serveur avant de se déconnecter.

L'utilisateur peut modifier des préférences de contexte, et, lorsqu'il n'est pas en calcul, se déconnecter du réseau. Ces préférences permettent au téléphone de son utilisateur de signaler au serveur qu'il est disponible ou non le cas échéant. Parmi les éléments de contextes identifiés nous listons : le niveau de batterie et la connectivité actuelle.

A la réception d'une partie d'un calcul, l'application va envoyer un message au serveur afin d'en valider la bonne réception ainsi que l'acceptation. Du point de vue serveur, si cette validation n'arrive pas dans un temps prédéterminé, il considère que la connexion est perdue et va alors envoyer la requête à un autre pair. De plus, au cours d'un calcul, le serveur envoie périodiquement des requêtes aux utilisateurs auxquels il a adressé un sous-calcul. Celles-ci lui permettent de distinguer un utilisateur toujours en calcul d'un autre n'étant plus disponible sans qu'il ait été notifié. Dans le second cas, le serveur renvoie cette requête à un autre pair disponible du réseau.

Un utilisateur ayant fait une demande de calcul peut être sollicité dans le processus de calcul dont il est l'initiateur.

Le serveur maintient pour chaque utilisateur un score indiquant la fiabilité qu'il lui attribue. Ce score augmente pour chaque résultat de sous-calcul reçu, et diminue lorsque le serveur doit ré-adresser un sous-calcul. Ce score permet au serveur de choisir à quels utilisateurs envoyer la requête de sous-calcul.

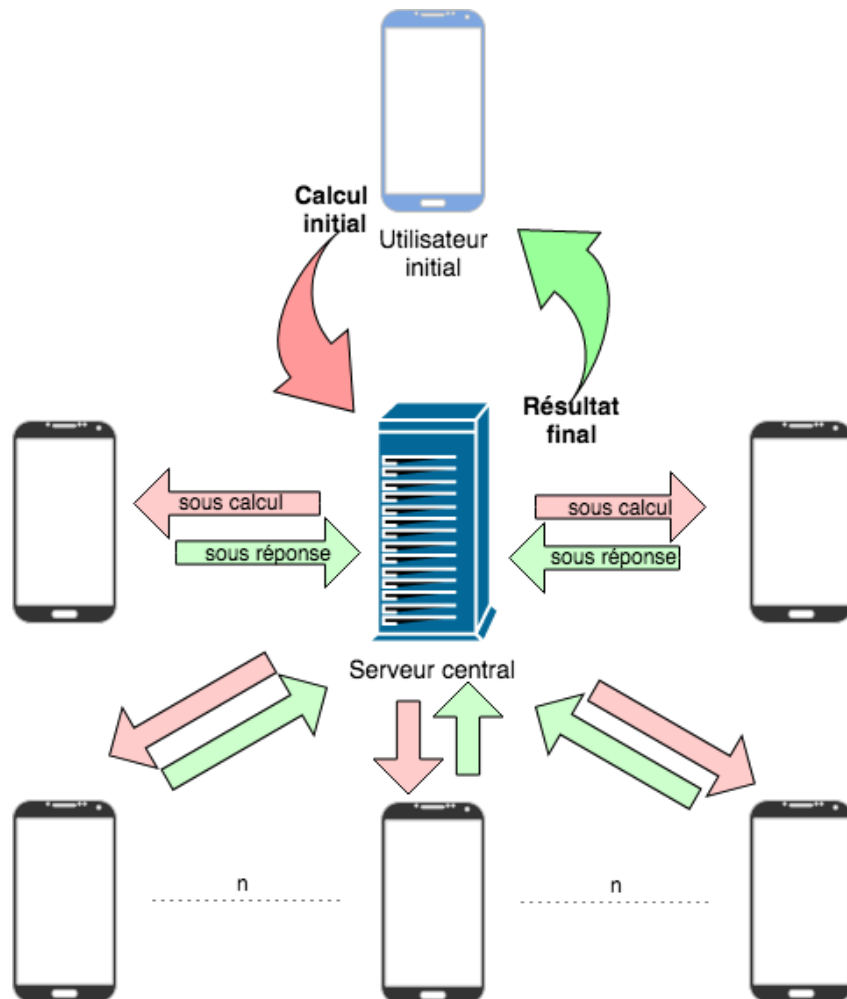


Schéma de fonctionnement de l'application

Hypothèses

Nous supposons que le calcul fourni est cohérent, fractionnable, et solvable.

Dans le cas d'un utilisateur ne répondant pas alors qu'un sous-calcul lui a été envoyé, nous estimons que la connexion a été rompue et le sous-calcul est envoyé à un autre pair.

Une requête de calcul est envoyée au travers d'un fichier présent sur le téléphone de son utilisateur.

Pour commencer nous nous limitons à des problèmes mathématiques simples (type addition, soustraction, multiplication et division). Le but étant de prouver la faisabilité d'un réseau et de la distribution d'un problème mathématique. Le fractionnement d'un problème et sa solvabilité sont des sujets à parts entières et le temps de réalisation du projet ne nous permet pas de nous concentrer sur ces aspects.

Distribution des tâches de l'application

Utilisateur faisant la demande

- Envoi d'une requête au serveur
- Réception de la réponse

Utilisateur lambda

- Attente et envoi de requête indiquant la disponibilité
- Réception d'une demande de sous-calcul par le serveur
- Calcul du résultat intermédiaire
- Envoi de la réponse intermédiaire au serveur
- gestion de contexte

Serveur

- Maintien de la liste des pairs disponibles
- Maintien de la liste des pairs en calcul
- Réception d'une requête de calcul
- Fragmentation du calcul
- Envoi des demandes de calculs intermédiaires
- Réception des réponses intermédiaires
- Formulation du résultat final
- Envoi de la réponse finale à l'utilisateur initial

Responsabilités au sein du projet

Les responsabilités détaillées par la suite sont principalement indicatives puisque chacun participera au développement de plusieurs composants du projet.

Chef de projet : Hugo Vandebunder

Responsable mobile : Florian Fichant Le Fur

Responsable fragmentation des calculs : Pierre Serrut

Responsable base de données : Jérémie Brault

Responsable serveur : Hugo Vandebunder

Améliorations futures

- Plusieurs types de calculs
- Envois de requête de calcul depuis web