Projet de systèmes distribués : simulateur de blockchain

1. Travail demandé

Le travail demandé comportera une implantation (Java, C, C++ ou autres). Un court rapport devra accompagner ce projet pour décrire les choix de votre implantation. Le projet devra utiliser une librairie de programmation distribuée (RPC, RMI, Corba ou autre). Un jeu de test présentera un exemple d'utilisation de votre programme. L'ensemble devra fonctionner sur les machines de l'UFR (turing ou autre).

2. Réalisation du projet

- Le projet se fera seul ou en binôme. L'évaluation sera différente pour les projets réalisés seuls et les projets réalisés en binôme.

- En cas d'ambiguïté, précisez dans le rapport votre interprétation personnelle. Toute solution cohérente et correctement justifiée pourra être acceptée.

3. Sujet

Les blockchain permettent un enregistrement distribué d'une suite d'opérations. Des opérations sont enregistrées lors de la création d'un nouveau bloc. Les nouveaux blocs sont ensuite distribués aux différents nœuds. Une opération est considérée comme définitivement enregistrée lorsqu'elle est dans une chaîne de blocs d'une majorité de nœuds.

Il s'agira dans ce projet de réaliser une plate-forme de simulation de l'évolution d'une blockchain. Deux types de nœuds seront présents dans le réseau : des nœuds blocs chargées de gérer la blockchain qui mettront à jour la blockchain et qui s'échangeront les blocs et les demandes des participants, et les nœuds participant qui s'inscriront auprès des nœuds blocs.
Les nœuds blocs seront régulièrement récompensés en points blockchain par les nœuds blocs auprès desquels ils sont inscrits. Ils pourront échanger des points entre eux. Les échanges sont enregistrés dans la chaîne de blocs. Il sera possible pour un nœud bloc d'augmenter le mérite d'un nœud participant, la fraction de blocs qu'il recevra sera augmentée en conséquence.

Les nœuds blocs reçoivent régulièrement des demandes des nœuds participants qu'ils mettent en attente. Les nœuds blocs génèrent régulièrement des nouveaux blocs. Lors de la création d'un nouveau bloc, les demandes en attentes sont ajoutées au bloc créé. Le nouveau bloc contiendra aussi le résultat d'une fonction de hachage appliquée au bloc précédent. Ce hash verrouille les blocs précédent : il sera alors possible d'identifier toute modification des blocs précédents en comparant le hash stocké au résultat de la fonction de hachage appliquée au bloc précédent. Il ne sera plus possible de modifier un bloc sans modifier tous les blocs suivant, la chaîne sera donc grandement modifiée par rapport aux chaînes de blocs enregistrées dans les autres nœuds et sera considérée comme non valide par les autres nœuds.

Un nœud participant peut s'inscrire, et donc être relié, à un (ou plusieurs) nœud bloc. Un nœud bloc peut refuser une inscription d'un nœud participant. Un nœud participant peut s'inscrire à nœud bloc, demander combien il possède de fraction de bloc et envoyer une demande de transfert de bloc à un autre nœud dont il connaît l'existence.

Un nœud bloc est relié à différents nœuds blocs voisins. Un nœud bloc retransmet à ses nœuds voisins les demandes des nœuds participants qu'il reçoit et qui doivent être inscrits dans la chaîne de bloc. Il transmet aussi à ses voisins les nouveaux blocs qu'il crée. Il peut demander tout ou une partie de la chaîne de blocs à ses nœuds voisins. Un nœud bloc peut augmenter le mérite d'un nœud participant auquel il est relié pour augment la quantité de fraction de bloc transmis à chaque participant auquel il est relié lors de la création d'un nouveau bloc.

Lorsqu'un nœud possède deux chaînes de blocs de longueurs différentes, il considère comme valide la chaîne de bloc la plus longue et ne tient plus compte de l'autre chaîne de blocs. Lorsqu'un bloc nœud reçoit deux nouvelles chaînes de bloc de longueurs identiques, il considère comme valide la première chaîne de bloc qu'il reçoit. Si un nœud bloc reçoit un bloc contenant des opérations qu'il lui reste à traiter, il supprime de la liste des opérations à traiter les opérations contenues dans le nouveau boc.

Il sera possible de visualiser à un instant la chaîne de bloc, c'est à dire le nombre de blocs et le contenu de chaque bloc.

Lors de la réalisation du système, vous pourrez prendre en compte les cas suivants. La note sera fonction du nombre de fonctionnalités implémentées et de la difficulté des problèmes traités.

- Comment sont créés les nouveaux blocs ? La solution la plus simple à mettre en place est que chaque nouveau bloc est créé au bout d'un temps aléatoire compris entre `n` et `n+k` par chacun des blocs nœuds. Il est possible de mettre en place une solution où les nœuds blocs doivent résoudre un problème pour pouvoir créer un nouveau bloc. Classiquement, le problème à résoudre est trouver un code à ajouter au bloc de telle sorte qu'une fonction de hachage appliquée au nouveau bloc (entête plus ensemble des opérations à enregistrée dans le bloc) commencent par un suite de `M` zéros. Plus `M` est grand, plus il faudra prendre de

temps pour trouver une nouvelle solution. M peut dans ce cas augmenter avec le temps pour rendre le problème de plus en plus difficile.

- Vous pouvez mettre en place des vérifications pour vous assurer qu'un nœud bloc mal intentionné ne puisse pas corrompre la chaîne de bloc.

- Vous pouvez chiffrer les opérations des nœuds participants (utilisation de clés publiques/clés privées) pour que les nœuds blocs ne puissent pas corrompre les opérations des nœuds participants.

- Vous pouvez mettre en place des tâches que devront réaliser les nœuds participants pour augmenter leur mérité auprès des nœuds blocs. Les nœuds blocs pourront diminuer le mérite de nœuds participant n'ayant pas réaliser de tâche ou n'ayant pas échangées de fractions de blocs depuis un certains temps.

Votre application permettra de lancer X blocs participants et Y nœuds blocs et de lancer une simulation de déroulement d'évolution de la chaîne de blocs.

La visualisation du déroulement d'une simulation pourra être réalisée à postériori (il n'est pas nécessaire qu'elle soit faite en temps réel), c'est à dire après la fin de la partie. Elle pourra être générée à partir d'un programme indépendant, utilisant un enregistrement des actions des différents joueurs/producteurs.

4. Remise du projet

La remise du projet se fera par moodle au plus tard pour le 14 mai 2018. Une soutenance sera organisée lors de la séance de travaux pratiques du 14 mai.
L'archive rendue devra contenir les sources, le fichier de compilation (makefile), les jeux de test et le rapport.

Documentation :

- partie 1 de https://openclassrooms.com/courses/comprendre-le-bitcoin-et-la-blockchain
- l'article original définissant le bitcoin, la plus connue des blockchain : https://bitcoin.org/bitcoin.pdf