Projet de TPEA

Scrabblos Java

Firat MERSIN
Alexis BELANGER
Nicolas MATTON

1er novembre 2019

1 Introduction

Nous souhaitons représenter un jeu de mots sous forme de blockchain. Pour ce faire nous aurons des Auteurs qui publient des lettres, et des Politiciens qui, avec les lettres produites créeront des mots sous forme de block qui viendront donc compléter leur blockchain après validation par un algorithme de consensus en Proof-of-Work.

Notre première approche sera en Peer-to-Peer, tour par tour. Les auteurs commenceront la partie. Durant le tour des auteurs, chaque auteur publie une lettre, puis vient le tour des politiciens qui essaieront de composer un mot avec les lettres à leur disposition puis le publieront sous forme d'un block et après validation par l'algorithme de consensus, les auteurs et les politiciens ajouteront ce block à leur blockchain. La partie se finira lorsque une blockchain atteint 21 blocks, soit 20 mots trouvés par les politiciens et qui ont été accepté par les autres joueurs. Par la suite nous aurons la même approche, mais cette fois si en "roue libre", les auteurs pourront proposer des lettres n'importe quand tout comme les politiciens qui pourront proposer des mots.

Pour la partie en roue libre et la partie avec Serveur, nous donnerons nos idées de fonctionnement et d'implémentation.
Le code qui est fournit se base sur l'approche Peer-to-Peer en tour par tour.

2 Installation et exécution

Une fois le code téléchargé, rendez vous à la racine du projet puis tapez les commandes suivantes dans l'ordre :

— javac src/block/*.java src/client/*.java src/data/Data.java src/launch/Launcher.java src/round/Round.java src/Words/*.java
— cd src
— java launch.Launcher

3 Structures

Un block est composé de cette façon :

- **String** previousHashId : hash de l'Id du block précédent dans la blockchain
- **ArrayList<Data>** word : Liste de Datas (lettres) composant le mot
- **String** hashId : hash de l'Id du block
- **String** politicianHashId : hash de l'Id du politicien ayant créé le block

Une lettre est représentée par une structure Data composée de cette facon :

- **char** letter : la lettre
- **String** authorHashId : hash de l'Id de l'auteur ayant publié la lettre
- **Block** lastBlockInChain : dernier block de la blockchain de l'auteur (utilisé pour signer la lettre)

La blockchain est réprésentée sous forme d'une liste de blocks, elle est initialisée avec un block "genese" pour permettre aux auteurs de signer leurs premières lettres avec ce block, en attendant qu'un premier mot soit trouvé par les politiciens. Les politiciens utilisent également le hash de l'Id du block "genese" pour initialiser le champ previousHashId du premier block qu'ils vont créer.
L'ensemble des mots est stocké dans un Patricia Trie, facilitant la recherche d'un mot existant a partir des lettres données.

4 Choix des mots

Les mots ne doivent contenir qu'une unique lettre par auteur. Pour respecter cette condition, les politiciens (qui sont tous honnêtes) choisissent aléatoirement une unique lettre par auteur parmi toutes les lettres valides qu'ils ont à leur disposition. Ensuite ils essaient de former un mot valide de taille minimum imposée (la difficulté). Cette difficulté varie tout au long de l'exécution, elle diminue si les politiciens ne trouvent aucun mot valide et augmente sinon.

Pour un politicien, une lettre est valide seulement si elle est signée par un block correspondant au dernier block de sa blockchain.

5 Peer-to-Peer

N'ayant pas de fournisseur de sac de lettres comme c'est le cas avec le serveur, nous donnons à chaque auteur un sac contenant un certain nombre de lettres minuscules pendant la phase d'initialisation du jeu.

5.1 Tour par tour

Un tour se déroule en deux étapes, le jeu des auteurs et le jeu des politiciens. Un auteur ne peut donc pas proposer une lettre pendant la phase de jeu des politiciens et inversement.

Une partie en tour par tour se déroule comme ceci : les auteurs vont tous proposer une et une seule lettre, avant de réveiller les politiciens et de s'endormir à leur tour. Une fois réveillés, les politiciens vont entamer leur phase de jeu en récupérant les lettres reçues pendant le tour des auteurs et en cherchant un mot valide (mot de taille supérieure ou égale à la difficulté, ne contenant pas plus d'une lettre venant d'un même auteur et correctement signé à partir du $hashId$ du block précédent). Ces derniers vont ajouter ce mot à l'état de la blockchain qu'ils connaissent, et l'envoyer à tous les autres participants qui vont l'accepter ou non selon l'algorithme de consensus.

Avec ce fonctionnement, on est donc sûr qu'un auteur n'injecte qu'une seule lettre aux politiciens par tour car une fois une lettre envoyée, un auteur va s'endormir sans avoir la possibilité de retourner à l'étape d'un envoi de lettre avant le prochain tour.

5.2 Roue libre

Pour le jeu en roue libre, il n'y a plus de phases de jeu Auteurs et Politiciens, chaque acteur du jeu peut participer à n'importe quel moment, il faut donc être vigilant avec la signature des lettres utilisées par les politiciens. En effet, là où au tour par tour la liste des lettres disponibles étaient nettoyée à chaque tour en ne gardant que les lettres correctement signées, on peut maintenant avoir dans un pool de lettres de politicien des lettres "anciennes" qui ne peuvent plus être utilisées dans les nouveaux mots. Les politiciens ne proposant que des mots valides au tour par tour, nous gardons le même principe de construction pour le mode roue libre.

5.3 Consensus

5.3.1 Principe

Nous avons choisi un algorithme en *proof-of-work*

Notre algorithme de consensus fonctionne comme ceci : lors du tour des politiciens, ces derniers créent leur mot valide respectif, les ajoutent à leur blockchain, et envoient cette dernière à tous les autres participants (auteurs et politiciens). Les blockchains reçues sont stockées dans une liste *buffer*. Avant la fin de l'exécution du tour des politiciens, ces derniers vont choisir de prendre comme blockchain la blockchain ayant le meilleur score global dans le buffer. Les auteurs feront de même en se réveillant.
Le score global d'une blockchain est donné par la somme des points de toutes les lettres présentent. Les points associés à une lettre sont les mêmes qu'au Scrabble.

5.3.2 Idée de *proof-of-stake*

Une possibilité d'implémentation d'un algorithme en *proof-of-stake* serait qu'a chaque tour de politicien, un leader soit élu avec une probabilité proportionelle à son score actuel, ce dernier aurait pour tâche de choisir le block à ajouter et à le propager aux autres participants. Le leader aurait donc intérêt à choisir le mot le plus rentable pour lui en terme de score.

5.3.3 En cas d'attaque

L'algorithme de choix de la "bonne blockchain" est fait de sorte que tous les acteurs aient la même blockchain à la fin d'un tour, cependant, si un attaquant en rôle de politicien arrive avec un autre consensus, il peut décider de choisir le mot augmentant le plus son score personnel (comportement égoïste comparé aux autres qui maximisent le score total de la chaîne) et ainsi créer un fork. Mais en faisant cela, il n'aura plus le même état de blockchain que les autres, ses mots seront donc systématiquement refusés.

Cependant, avec l'idée de *proof-of-stake* il est possible qu'un attaquant avec un consensus différent soit désigné comme leader et décide de placer son block dans la blockchain et non le meilleur block possible, le tout en gardant la possibilité de proposer des mots valides par la suite.

6 Idée d'adaptation pour le serveur

Pour l'utilisation du serveur, les politiciens devraient à chaque création de block b envoyer ce dernier au serveur pour qu'il soit transmis à tous les participants, ces derniers récupèrent le *wordpool* du serveur pour reconstruire la blockchain à laquelle appartient le block b pour voir si le block b est effectivement valide et calculer le score total de la blockchain.
Les méthodes d'envoi et de réception de données des auteurs et des politiciens devraient donc être changées.

6.1 Consensus sur le choix du block

Comme pour le Peer-to-peer, les politiciens et les auteurs vont choisir la blockchain donnant le meilleur score total parmi toutes les blockchains qui ont été contruitent suite à la reception des différents block b proposés à ce tour. L'idée d'algorithme en *proof-of-stake* serait également le même que pour le peer-to-peer.