Scrabblos

Mathias Bourgoin, Julien Tesson

9 novembre 2020

1 Présentation

L'objectif du projet est d'implanter un jeu de mots sous la forme d'une blockchain. Nous considérerons deux types d'acteurs : des **Auteurs** (clients), chargés d'injecter des lettres dans la chaine et des **Politiciens** (producteurs de blocs) chargés de les agréger sous la forme de mots. Dans un premier temps nous considérerons qu'il existe une autorité centrale avec laquelle communiquent Auteurs et Politiciens.

1.1 Schéma général d'une partie :

Nous vous proposons deux manières d'implanter ce jeu :

— soit au tour par tour, en laissant l'autorité centrale décider des changements de tours

— soit en temps réel (en roue libre), où Auteurs et Politiciens devront eux-mêmes s'organiser entre eux

Dans les deux cas, nous vous proposons d'utiliser le serveur central **scrabblos-server** fourni (disponible sur le dépôt gitlab [1] du projet). L'ensemble des communications s'effectue au format JSON, une description détaillée des communications possibles avec le serveur est fournie dans un document dédié disponible sur le dépôt du projet.

1.1.1 En tour par tour

— En début de partie, les Auteurs vont s'enregistrer auprès de l'autorité centrale et récupérer leur ensemble de lettres de départ.

— À chaque tour, les auteurs peuvent chacun soumettre une unique lettre qui sera (ou non) intégrée dans un mot par les politiciens.

1. `https://gitlab.com/julien.t/projet-p6/`

À chaque lettre sont associés des points et un auteur remporte ces points si sa lettre a été intégrée à un mot sur la chaine. Le serveur central stockera toutes les lettres soumises par les auteurs dans son **letterpool** (dont l'utilisation est précisée dans le document dédié au serveur central).

— À chaque tour, les politiciens sont en compétition pour produire un mot valide. C'est-à-dire :

1. Faisant référence (par son hash) à un mot prédécesseur valide
2. Ne contenant que des lettres injectées avec une référence au mot prédécesseur
3. Ne contenant pas plus d'une lettre venant d'un même client
4. Existant dans un dictionnaire

Une fois qu'ils ont réussi à produire un mot, ils doivent le proposer au serveur central qui le stockera dans son **wordpool** et le propagera à tous les participants (Auteurs ou Politiciens).

— Un tour est défini par une durée pendant laquelle Auteurs et politiciens peuvent effectuer leurs actions

— Après un nombre de tours défini par le serveur central, la partie se termine et les vainqueurs, meilleur auteur et meilleur politicien, sont déterminés par consensus (à vous de définir ce consensus).

1.1.2 En roue libre

— On abandonne ici le tour par tour, les politiciens peuvent produire un mot n'importe quand, au-dessus d'un mot déjà proposé.

1.2 1. Le serveur central

Nous vous fournissons le code du serveur central. Il est capable de répondre à plusieurs messages. En particulier il est capable de recevoir/répondre aux messages suivants :

— enregistrement d'un nouvel auteur
— démarrage d'une communication continue avec l'auteur/politicien connecté
— retourner l'état actuel de son bassin de lettres (letter pool)
— retourner l'état actuel de son bassin de mot (word pool)
— retourner un état incrémental de ses deux bassins
— injecter une nouvelle lettre d'un auteur
— injecter un nouveau mot d'un politicien

Plus d'informations sur les messages (en particulier leur format précis) sont disponibles dans le document dédié disponible sur le dépôt du serveur central.

Attention : Le serveur central ne réalise pas le consensus, il ne sert que de relais pour simplifier les communications entre auteurs et politiciens. C'est aux auteurs/politiciens d'implanter un algorithme de consensus permettant d'assurer que les règles du jeu sont respectées et qui, en fin de partie, permettra de déterminer l'auteur et le politicien vainqueurs de la partie.

1.3 2. Les Auteurs

Les auteurs doivent injecter une lettre par mot. En tour par tour, le serveur central passera au tour suivant lorsque tous les auteurs enregistrés auront soumis une lettre (ou lorsqu'un timeout défini au lancement du serveur aura été atteint).

Comme le serveur ne réalise pas l'algorithme de consensus, c'est aux auteurs de vérifier quels mots ont été produits par les politiciens et décider duquel d'entre eux définit la tête de la chaîne (le dernier mot inclus dans la chaine). C'est ce mot qu'ils utiliseront pour signer leur prochaine lettre injectée.

Le serveur n'implémente pas les règles du jeu : c'est à vous de décider par exemple de ce qu'il advient des lettres déjà soumises sur le serveur, mais non incluses dans un bloc précédent.

Les auteurs gagnent des points lorsqu'une de leurs lettres est incluse dans un mot de la chaîne. Vous pourrez utiliser un système de points similaire à celui du scrabble pour pondérer les points associés à chaque lettre.

1.4 3. Les Politiciens

Les politiciens doivent injecter des mots valides c'est-à-dire satisfaisant les contraintes précisées dans la description du jeu **au tour par tour** ci-dessus.

Les mots ne doivent en particulier contenir que des lettres signées avec le même mot, et jamais plus d'une lettre soumise par un même auteur. On appellera le mot utilisé pour signer l'ensemble des lettres choisies par le politicien, son prédécesseur. Comme le serveur central n'implante pas le consensus, il ne connait pas le dernier mot valide de la chaine, mais il connait l'ensemble des mots soumis par tous les politiciens depuis le début de la partie.

C'est au politicien, avec cette information, et avec l'ensemble des lettres soumises par les auteurs, de décider quel mot il va utiliser comme prédéces-

seur pour soumettre un nouveau mot.

Si besoin, pour valider les mots, nous vous fournissons, sur le dépôt du serveur central, 4 dictionnaires de mots (au format texte brut avec 1 mot par ligne) :

— dict_100000_1_10.txt contient 100000 mots de 1 à 10 caractères
— dict_100000_5_15.txt contient 100000 mots de 5 à 15 caractères
— dict_100000_25_75.txt contient 100000 mots de 25 à 75 caractères
— dict_100000_50_200.txt contient 100000 mots de 50 à 200 caractères

Le nombre de points attribués à une lettre est sa valeur au scrabble français. [2]

Les politiciens gagnent des points si leur mot est inclus dans la chaîne. Ils gagnent alors la somme de tous les points des lettres du mot soumis.

1.5 Questions

Question 0

Comment s'assurer que l'auteur n'injecte pas plusieurs lettres pour un bloc ?

Question 1

Implanter un Auteur et un Politicien pour la version tour à tour. Vous pourrez vous appuyer sur les fonctions contenues dans le fichier `src/client_utils.ml`.

Le code de l'Auteur en version tour par tour est fourni dans le fichier `src/author.ml`. Il conviendra de l'adapter si besoin et de compléter les fonctions du fichier `src/consensus.ml` nécéssaires à son bon fonctionnement.

Le squelette du code du Politicien est fourni dans le fichier `src/politicien.ml`. À vous de le compléter.

Contactez-nous si vous avez besoin de changer la structure des messages échangés.

Question 2

Décrivez dans le rapport votre implantation des fonctions du fichier `src/consensus.ml`.

2. `https://fr.wikipedia.org/wiki/Lettres_du_Scrabble`

Question 3

Implanter un Auteur et un Politicien pour la version "en roue libre" avec serveur central. Quelles modifications devez vous apporter à votre algorithme de consensus ?

Question 4

Votre implantation est-elle resistante aux attaques ? Si non lesquelles et quelles modifications faudrait-il envisager pour y résister ?

Question 5

Comment modifieriez-vous le système pour ne plus utiliser le serveur central (en PoW ou Pos, au choix).

Implantez cette solution.

Question 6

Ce système s'appuie-t-il plutôt sur un algorithme de consensus à Proof-of-Work ou à Proof-of-Stake ? Comment modifieriez-vous ce système pour implanter un algorithme de consensus basé sur l'autre modèle ?

Évaluation

Ce projet sera réalisé par groupe de 3. Pensez à nous informer à l'adresse mathias.bourgoin@nomadic-labs.com de la composition de votre groupe ainsi que de l'adresse du dépôt (git, svn, etc.) que vous utilisez pour ce projet.

Le projet sera évalué sur la base d'une soutenance par groupe. Il sera aussi attendu un rapport de **au moins** 3 pages détaillant votre implantation et vos choix ainsi que vos réponses aux questions ci-dessus.

Lors de la soutenance, vous expliquerez :

— comment vous obtenez un consensus entre les acteurs du jeu.
— est-ce qu'un attaquant peut faire changer le consensus en sa faveur ?
— si oui, quelle mesure pourrait-on prendre pour l'en empêcher ? Si non, qu'est-ce qui l'en empêche.

Compte tenu du temps limité pour mener à bien l'implantation du projet, il est acceptable que l'implémentation ne soit pas très robuste, mais vous

devez être en mesure d'expliquer les défaillances, et de proposer des pistes d'amélioration.