Projet Informatique: Analysis of the Star Wars Social Graph

Nikolay Ionanov

7 avril 2017

1 Structure generale

Le but de ce problème c'est de definir k-neighbourhood pour un caracter specifié à partir de Wookieepedia, qui contient environ 135000 pages. Pour faire ça il va nous falloir effectuer le BFS parallel, ayant parsé préalablement le wookiepedia. On defini ainsi l'execution suivante de notre progamme
1) D'abord il faut effectuer le proprocessing. C'est à dire extraire tous les caracters presents dans ce wiki. Cette procedure se fait de manière sequentiel, car le parsing de JSON ne se parallelise pas.
2) Créer les threads pour affectuer le BFS. Chaque thread doit parser la page html de chaque caracter pour extraire les liens associés.
3) Organiser lle recherche niveau par niveau

2 Construction des requettes

L'API de wiki (wookieepedia dans notre cas) permet d'obtenir l'information necessaire (noms des characters, tous les liens sur une page particulière) sous la forme de JSON. Pour acceder l'ensemble des characters on peut remarque que dans le code source que toutes les pages des characters sont marquées avec le template suivante "Template ACharacter". Donc la requette suivante est utilisée : `http://starwars.wikia.com/api.php?format=json&action=query&list=embeddedin&eititle=Template%3ACharacter&eilimit=500` Ensuite pour obtenir les liens sur une page particulière on utilise `http://starwars.wikia.com/api.php?action=query&format=json&prop=links&titles=page_title&pllimit=500` Pour telecharger le contenu de la requette on utilise

```
URI request = new URI(base_uri.replace(" ", "%20"));
String res = Request.Get(request)
  .connectTimeout(5000)
  .socketTimeout(5000)
.execute().returnContent().asString();
```

Pour acceder le contenu de requete POST html, fluent-hc de APACHE est utilisé.

3 Structure de données

Pour affectuer le BFS on a besoin d'une structure des données qui implemente l'interface d'une queue. Cette structure doit être thread-safe et prendre en compte que on fait le bfs "niveau par niveau".

```
public class SecureQueue<E> {

        private LinkedList<E> currentLevel;
        private LinkedList<E> nextLevel;
```

```
        private final Lock lock;

        public SecureQueue(){
                this.currentLevel = new LinkedList<E>();
                this.nextLevel = new LinkedList<E>();
                this.lock = new ReentrantLock();
        }

        public void enqueue(E element)
        {
                lock.lock();
                try{
                        this.nextLevel.add(element);
                }finally{
                        lock.unlock();
                }
        }

        public E dequeue()
        {
                lock.lock();
                try{
                        return this.currentLevel.removeFirst();
                }finally{
                        lock.unlock();
                }
        }

        public void proceed(){
                lock.lock();
                try{
                        this.currentLevel = this.nextLevel;
                        this.nextLevel = new LinkedList<E>();
                }finally{
                        lock.unlock();
                }
        }

        public int size(){
                lock.lock();
                try{
                        return this.currentLevel.size();
                }
                finally{
                        lock.unlock();
                }
        }
}
```

Le parallelisme est assuré par des locks. Pour garantir la separation des niveaux de recherche, la structure est implementé avec deux LinkedList. La première sert à stocker les characters de niveaux presente. Les thread de recherche vident cette list pour remplir la liste de niveau suivante.

4 PBFS

La structure des données SequreQueue nous permet de realiser le BFS parallel. Chaque thread prend un caracter de la queue et fait une requette GET pour obtenir les liens associées avec ce caracter.Ensuite il verifie si le lien apartient à l'ensemble des caracters et si ce lien n'a pas encore été consideré, il note la profondeur et l'ecrit dans le fichier.

5 Les resulats