HE-Arc

Projet d'automne No 212

Docker Hub Taxonomy - Journal de travail

Auteurs :
Pedretti Maël

Responsables :
Pasin Marcelo
Schiavoni Valerio

24 janvier 2018

Le projet Docker Hub Taxonomy est réalisé dans le cadre du Travail d'Automne, un module de 3ème année de Bachelor au sein de la Haute-École Arc — Ingénierie, section Développement Logiciel et Multimédia. Le présent document décrit le journal de travail et développe le suivi du projet.

haute école
neuchâtel berne jura
arc ingénierie
saint-imier le locle delémont

Table des matières

1. Introduction

Afin de pouvoir tracer le travail effectué, voici un journal de travail regroupant les points majeurs de l'avancement du projet.

2. Suivi du travail

2.1 14.09.2017

Première rencontre avec Marcelo Pasin et Valerio Schiavoni Prise en main de docker Recherche sur les API Essai de crawl du site web en python avec un crawler hyper basique

-> page d'index et page linkées parcourues

2.2 19.09.2017

Bref rendez-vous avec Marcelo Pasin Essai de récupération de données via l'api REST docker

-> pas concluent

2.3 24.09.2017

Recherche sur l'api REST

-> Toujours pas trouvé de moyen de récupérer les dockerfiles

2.4 26.09.2017

Recherche sur l'api REST

-> Toujours pareil

Discussion avec Marcelo et Valerio

-> Retour au crawl HTML

Réussite dans la récupération des dockerfiles depuis les liens de la page Explorer !

2.5 11.10.2017

Essai de débugage du crawler, toujours des bugs

2.6 12.10.2017

Idem

2.7 13.10.2017

Idem Améliorations dans le code

2.8 14.10.2017

Idem

2.9 15.10.2017

Réussite de débugage du crawler.

-> il est maintenant possible de crawler depuis la page explorer et de récupérer tous les paquets et leur dépendance dans les dockerfiles

2.10 17.10.2017

Discussion avec Valerio

-> utiliser la fonction de recherche pour crawler plus de paquets

-> Ajouter des compteurs pour 4 symboles différents pages et images

-> Mesurer durée de chaque round de crawling

2.11 24.10.2017

Ajout d'un compteur de temps et essai de crawl multiprocessing

2.12 31.10.2017

Fin du multiprocessing pour explore et package pages

2.13 07.11.2017

Crawling et parsing des dockerfile dans des classes et process séparées Ajout d'un script de test pour la qualité du code.

2.14 14.11.2017

Il est maintenant possible de construire un graphe à partir des fichiers de logs pour afficher les statistiques

2.15 21.11.2017

Discussion avec Marcelo :

-> Ajouter comme node le hash de la commande qui ajoute des dépendances dans le dockerfile

-> graphviz : juste écrire "dép -> arrivée"

Recherches sur graphviz et sur comment implémenter au mieux cette récupération de graphes.

2.16 23.11.2017

Recherches supplémentaires sur graphiz et ses outils permettant d'afficher des graphes.

2.17 28.11.2017

Implémentation du process search page et intégration dans le multiprocessing.

2.18 4.12.2017

Fin de l'implémentation du log du graph au format dot. Plusieurs problèmes apparaissent.

2.19 5.12.2017

Recherche et implémentation du script permettant d'afficher les graphes à partir des fichier dot.

2.20 12.12.2017

Implémentation d'un process responsable d'écrire le log des statistiques afin de ne plus avoir de problèmes d'accès et d'avoir des timestamps triés par ordre croissant.

2.21 16.12.2017

Implémentation d'un process resposable d'écrire le graphe dans un fichier dot. Évite d'autre problèmes d'accès.

2.22 17.12.2017

Nettoyage du code.

2.23 26.12.2017

Recherche concernant les technologies permettant d'afficher le graphe de manière plus attirante que les outils Graphviz.

-> rien de très concluant

2.24 29.12.2017

Idem

2.25 04.01.2017

Essai d'une technologie Javascript. Découverte de problèmes liés au format dot.

2.26 05.01.2017

Essai différents avec la même technologie, rien de concluant.

2.27 09.01.2017

Essai avec d'autres frameworks javascript mêlés à des scripts python pour changer le format du graph, rien ne marche.

2.28 16.01.2017

Toujours pas d'avance

2.29 17.01.2017

Découverte du package pyveplot et génération d'un premier graphe de test suivant l'exemple fourni. ça marche !

2.30 19.01.2017

Réussite dans l'affichage des données receuillis sur Docker Hub de deux manières différentes. D'une part l'affichage avec les points positionnés de manière aléatoire sur les axes. D'autre part l'affichage avec les points positionnés par rapport à leur poids (les plus lourds, qui ont le plus de liens, sont vers tirés vers l'extérieur)

2.31 20.01.2017

Début du rapport

2.32 22.01.2017

Rapport

2.33 23.01.2017

Rapport

2.34 24.01.2017

Journal de travail et rapport.