

LOG8371: Ingénierie de la qualité en logiciel

Hiver 2019

TP2 : Efficacité et performance

Remarque :

Les travaux pratiques constituent une partie importante du cours et ont pour objectif de vous pousser à concevoir des plans d'assurance de qualité des logiciels, à élaborer des stratégies de test et à vous servir des différents outils disponibles pour évaluer la qualité des logiciels selon des critères donnés. Il vous est recommandé de prendre ces travaux au sérieux et de faire appel à votre créativité et à votre pensée critique pour mieux les réussir. La collaboration avec vos collègues est permise durant et en dehors des séances de laboratoire, cependant les règlements relatifs au plagiat restent tout de même applicables en tout temps.

I) Objectifs du TP :

Les objectifs de ce deuxième TP sont de maîtriser:

- La compréhension et la définition des cibles de la performance logicielle.
- La mesure de l'utilisation des ressources du point de vue du logiciel (profiling) et de l'infrastructure (monitoring).
- Le déploiement automatique d'un service en utilisant des conteneurs.
- Le contrôle automatique de l'allocation dynamique des ressources (autoscaling).

II) Énoncé :

Q1) Mettez à jour le plan de qualité pour ajouter les critères et sous-critères qui portent sur l'efficacité et la performance de Weka. Mettez à jour les objectifs, les tests et tous les détails nécessaires. Concentrez-vous sur les 5+1 fonctionnalités que vous avez spécifiées au TP1. Soumettez la nouvelle version du plan dans l'entrepôt soumis. (20 points)

Q2) Déployez une version REST de Weka en utilisant des conteneurs Docker. Clonez cette version de Weka dans votre entrepôt et créez des images Docker qui sont nécessaires pour déployer le logiciel. Soumettez un petit manuel avec les commandes pour déployer Weka en Docker. Dans le vidéo de la question Q6, vous devez démontrez votre déploiement en Docker. (5 points)

Weka REST : <https://github.com/jguwekarest/jguwekarest>

Q3) Profilez votre version de Weka en utilisant un outil comme JProfiler. Rapportez les résultats sur la consommation des ressources (CPU, RAM etc.). Rapportez ces résultats par rapport aux différents cas de charge (nombre de requêtes et de données). Préparez au moins 4 scenarios (un de charge réduite, un de charge moyenne, un de charge augmentée, et un de charge augmentée exceptionnelle). Soumettez un petit manuel d'usage du JProfiler pour votre projet en incluant les étapes pour la configuration et l'exécution de l'outil, les données d'entrée (aussi les scenarios spécifiés) et les résultats. (20 points)

JProfiler : <https://www.ej-technologies.com/products/jprofiler/overview.html>

Q4) Préparez et exécutez des load tests pour votre version de Weka. Utilisez les scenarios créés pour la question 3. Utilisez l'outil JMeter pour exécuter les load tests. Rapportez sur les scenarios ou les objectifs ne sont pas possible à atteindre. Soumettez un petit manuel d'usage du JMeter pour votre projet en incluant les étapes pour la configuration et l'exécution de l'outil, les données d'entrée (aussi les scenarios spécifiés) et les résultats (20 points)

JMeter : <http://jmeter.apache.org/usermanual/get-started.html>

Q5) Configurez le monitoring pour Docker. Définissez les seuils de CPU entre lesquels les objectifs de performance sont garantis. Implémentez la capacité d'adapter l'infrastructure Docker de Weka pour répondre à l'augmentation de charge (en ajoutant des conteneurs) ou à la réduction de charge (en retirant des conteneurs). Utilisez la méthode « scale » de docker-compose pour faire l'adaptation. (20 points)

Q6) Vidéo : Démontrez le déploiement de Weka REST en Docker. Démontrez l'exécution du système avec peu de ressources et une charge de travail augmenté. Démontrez les métriques problématiques pour l'utilisation des ressources. Exécuter

la commande « scale » pour ajouter des ressources et redémontrez les métriques corrigées. (15 points)

Docker-compose/scale : <https://docs.docker.com/compose/reference/scale/>

Docker monitoring : <https://rancher.com/comparing-monitoring-options-for-docker-deployments/>

III) Remise :

Le travail doit être fait par équipe de 3-5 personnes et doit être remis via Moodle au plus tard le :

➤ 21 Mars avant 23h59 pour les trois groupes.

Veuillez envoyer un fichier de type *.zip qui va contenir le nom et membres de votre équipe, le lien pour le dépôt Github (vous pouvez utiliser le même comme dans le TP1), le lien pour le vidéo YouTube et votre plan de qualité. Le fichier portera le nom :

log8371_TP2_NomÉquipe.zip

Les travaux en retard seront pénalisés de 10 % par jour de retard. (Aucun travail ne sera accepté après 4 jours de retard. Si votre dépôt ne respecte pas la nomenclature définie ci-dessus, 0.5 point de pénalité sera appliqué). Tous les livrables doivent être de haute qualité et être traité en façon professionnelle.

Bonne chance.