

Monographie - Exploitation des journaux (logs) d'application

Cartouche 2

- Code du sujet : 40Z
- Code de l'alliance : 2020A40Z (aaaa : année, s : saison, nnx : code du sujet) - exemple : 2019A60A
- Liste des équipes
- E16 - CASIment l'UML
 - Larcher, théo (aerodynamic-saucepan)
 - Billard, baptiste (BaptBillard)
 - Lebeaud, simon ({pseudo Github membre 1-2})
- E04 - Equipe B
 - Lacroix, Daphné (dalacroix)
 - Lardière, Nina (nlardiere)
 - Delaunay, Julien (jdelaunay)
- E07 - Equipe C
 - Benkassou, Ali (khazit)
 - Leclerc, Tanguy (LeclercTanguy)
 - Marie, Joran (Joran MARIE)
- E08 - Equipe D
 - De Laroque, Florent (FlorentDL)
 - Marie, Leo (lmarie01)
 - Toutain, Aurélien ## Partie A

A0 : Introduction.

- L'objectif de ce sujet est de trouver des techniques permettant d'analyser efficacement des journaux (*logs*) d'une application dont le volume est très important. Ce projet permettra d'apporter des éléments de solution à un besoin réel et actuel d'entreprises devant manipuler des bases de données volumineuses (*Big Data*) et marchera de paire avec une application, choisie par l'équipe, qui permettra de générer les données (les *logs*).
Ce projet n'a donc pas pour but d'apporter une amélioration à un processus de traitement de données mais bien à fournir des outils nouveaux d'analyse.
- Les tests des solutions trouvées seront conduits par l'outil **Gatling** qui permet de stresser une application par une montée en charge importante.
- L'application choisie par l'équipe est : 60R : flux réseaux sociaux OU 60B : annuaire contacts ASI OU Application spécifique.

A1. Glossaire et Mots-clés.

- Big Data
- Base de données
- Logs
- Analyse
- ELK (Elasticsearch, Logstash, Kibana)
- Rsyslog
- Fichiers RRD
- Gatling
- Stress, montée en charge
- Devops

A2. Webographie

- Wikipédia
- Simplilearn
- Datamation
- Sodifrance (Gatling)
- ElasticSearch

A3. Bibliographie

A4. Acteurs

Apache Solr

Apache Software Foundation est une organisation à but non lucratif créée en juin 1999 dans le Delaware aux États-Unis par Brian Behlendorf, Roy Fielding et Ben Laurie. Elle développe ses logiciels en open-source, en mode collaboratif.

Sa plateforme logicielle de moteur de recherche Solr (prononcée “Solar”), basée sur le moteur d’indexation et de recherche de texte **Lucene**.

Le développement de Solr débute chez CNET, mais ces derniers décident très vite de rendre le code public : en janvier 2006, la première version de la plateforme est publiée, et son développement est confié à Apache. Ces derniers améliorent alors le moteur, notamment en 2008 : ils améliorent largement ses performances et le dotent d’une recherche distribuée.

Les projets **Lucene** et **Solr** sont fusionnés en 2011 : le développement est commun, les distributions sont séparées mais un tronc commun permettant de ne disposer que d’une seule version est vite créé. La recherche distribuée est nettement améliorée ; **Solr** est aujourd’hui utilisée notamment par **Adobe**, **Instagram**, **Netflix**, **eBay** et **Goldman Sachs**.

Solr permet l'intégration des bases de données, le surlignage des termes trouvés, l'auto-suggestion et la recherche full-text.

ElasticSearch

Elasticsearch inc. est une entreprise centrée sur la recherche fondée par Steven Schuurman, Uri Boness et Simon Willnauer.

L'entreprise est basée sur le succès de l'outil **Elasticsearch**, un logiciel open source, fonctionnant sur Lucene pour l'indexation et la recherche de données. Suite au succès d'**Elasticsearch** deux autres projets sont créés ; Logstash et Kibana. Logstash est un outil informatique de collecte, analyse et stockage de logs, c'est un outil d'ingestion open source. Il est généralement associé avec **Elasticsearch**, moteur de recherche distribué, et Kibana, interface d'Elasticsearch. En effet Kibana est une interface utilisateur permettant de visualiser les logs. Ces trois outils ; **Elasticsearch**, Kibana et Logstash forment la solution EKL. L'entreprise met toute la puissance de la recherche au service de ses clients pour les aider à explorer et analyser leurs données, que ce soit le cours des actions, les flux twitter ou les logs Apache.

Algolia

Algolia est une start up française. Lors de sa création en 2012 par Nicolas Dessaigne et Julien Lemoine, ils ont commencé à travailler sur la recherche hors ligne pour les téléphones mobiles.

Aujourd'hui Algolia propose une plateforme de recherche en tant que serveur « search as a service » qui est comme un moteur de recherche hébergé en externe pour faire des recherches sur site avec des résultats rapides et concrets. **Google** ou **Microsoft** proposent déjà cela, mais ils explorent et indexent le web de manière générale. Algolia propose, quant à lui, de n'indexer que les sites de leurs clients ce qui rend les recherches plus simples. Ils utilisent **RESTful JSON API** pour que les données du client soient push vers Algolia. La zone de recherche est ensuite ajoutée aux pages web du client. Comme la recherche se fait sur un site précis et non sur le web, Algolia peut proposer des recherches intelligentes et plus spécifiques au site car ils en connaissent la structure et la sémantique. Ils sont également très attentifs sur leur capacité à fournir des fonctionnalités instantanées, multi-plateformes et tolérantes aux fautes de frappe. Leur solution n'est toutefois pas open-source.

Algolia fournit son service via diverses API. Il prend en charge plusieurs langages tels que Python, Ruby, PHP, JavaScript, Java, Go, C #, Scala et les systèmes d'exploitation Android et iOS. Leur solution est également intégrable sur Ruby on Rails, Symfony, Django et Laravel.

A5. Facteurs qualité

- Consistency
- Accuracy
- Error Tolerance

A6. Indicateurs qualité

- Consistency
- Task effectiveness
- Task Time
- Accuracy
- Number of failures
- Number of faults
- Error tolerance
- Number of memory related errors
- Number of inaccurate computations encountered by users

A7. Références théoriques

Nous allons dans cette partie faire le point sur les patterns appliqués au DevOps issus de <http://blog.devopsguys.com/2030/02/20/twelve-devops-anti-patterns/> et <https://www.slideshare.net/realgenekim/2012-velocity-london-devops-patterns-distilled>

Ces patterns indiquent la marche à suivre quand on réalise du DevOps.

Les 3 principes du DevOps

Systems thinking (Dev to Ops)

Le premier principe consiste à comprendre le flux de travail et à toujours chercher à l'augmenter. Il ne faut jamais autoriser des optimisations locales qui peuvent causer des dégradations globales et il faut parvenir à une compréhension approfondie du système.

Amplify feedback loops (Ops to Dev)

Ce principe consiste à comprendre et répondre aux besoins des clients internes et externes. Pour cela il faut de nombreux feedbacks qui doivent être conscits. Il faudra créer et intégrer les connaissances où elles sont nécessaires.

Culture of continual experimentation and learning

Le troisième principe consiste à appliquer les deux premiers en continu, afin de créer un environnement qui permet de l'apprentissage et de la compréhension en continu.

DevOps Areas

Nous allons maintenant voir les 4 thématiques du DevOps.

Area 1: Extend delivery to production

Cette thématique a pour but principal de se concentrer sur l'aspect "entreprise" avec une vue de bout en bout du projet. Cela permet d'engager la conversation entre les différents acteurs pour avancer.

Les différentes étapes sont:

- Établir la confiance entre les équipes avec des meetings.
- Comprendre le goulot d'étranglement, la limite du processus.
- L'intégration continue.
- La livraison continue.
- Intégrer les autres rôles dans le processus (management, assurance qualité ...).

Area 2: Extend operations feedback to project

Cette thématique a pour but de produire un feedback et de donner de la visibilité sur le travail réalisé. Cela afin de montrer les améliorations de l'organisation.

Pour cela il faut:

- Des objectifs et des instructions de fonctionnement clairs.
- Une prise de conscience commune de la situation.

Area 3: Embed Project knowledge into Operations

Cette thématique a pour but de raccourcir et d'amplifier les feedback, de créer de la connaissance et des capacités où il y en a besoin et s'assurer de l'optimisation du système.

Area 4: Embed Operations knowledge into Project

Cette thématique a pour but d'apprendre aux développeurs à réfléchir comme les opérateurs, de gérer la dette technique et d'avoir une vision de bout-en-bout du projet.

Partie B

B1. Approche technique

B2. Solutions technologiques concurrentes

B3. Solutions retenues

B4. Liste de métriques

B5. Découpage (WBS) pour réalisation des prototypes

Partie C

C1. Architecture solution W

C2. Architecture solution X

C3. Architecture solution Y

C4. Architecture solution Z