Contribuer à un projet logiciel libre

Retour d'expérience avec Hadoop

Grigori Rybkine

Laboratoire de l'Accélérateur Linéaire
Orsay

Séminaire Service Informatique, 11 avril 2018

Plan

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Plan

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Plan

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Plan

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Plan

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Plan

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Plan

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Plan

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Plan

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Plan

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Apache Hadoop

- Dans le projet EventIndex de l'expérience ATLAS nous utilisons le logiciel du projet Apache Hadoop — un framework qui permet le traitement distribué de gros ensembles de données.
- Hadoop comprend les modules :
  - Hadoop Common
  - Hadoop Distributed File System (HDFS)
  - Hadoop YARN (Yet Another Resource Negotiator)
  - Hadoop MapReduce (Un système, basé sur YARN, pour le traitement parallèle de gros ensembles de données).
- Hadoop Common supporte des formats de fichier de base comme `SequenceFile`, `MapFile` (deux `SequenceFile`s — l'un avec les données, l'autre étant l'index pour l'accès aléatoire aux données). Nous utilisons le dernier pour stocker et interroger les données.

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

EventIndex et Hadoop

- L'espace occupé par les données croît et nous manquons d'espace de temps en temps.
- J'ai exploré les sous-formats de compression différents de `SequenceFile` et trouvé que l'utilisation du plus efficace d'entre eux — compression par bloc — pourrait, dans notre cas, réduire la taille d'un facteur 10.
- Cependant, j'ai également découvert que les requêtes de l'accès aléatoire ne fonctionnaient pas sur nos fichiers quand ils étaient compressés par bloc.
- Grâce à la disponibilité du code source, j'ai été capable d'examiner le problème complètement et localisé un bogue dans une méthode de `MapFile`.
- Après avoir corrigé le problème et maintenant utilisant la version corrigée pour EventIndex, j'ai décidé de soumettre le patch au projet Hadoop.

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Environnement de production et de test

- Hadoop utilise Git comme système de gestion de version de code source.
- Le moyen le plus simple d'obtenir un environnement de production et de test avec tous les outils appropriés est d'utiliser la configuration Docker fournie.
  - Cela nécessite une version de Docker installée et exécutée en tant qu'un utilisateur normal.
- Un script est fourni pour générer et exécuter l'image du Docker.
- La construction et l'exécution se sont déroulées sans accroc, sauf que l'arborescence du code source montée n'était pas accessible depuis le conteneur — je l'ai réglé assez rapidement, plus sur la diapositive 11.
- Actuellement l'environnement est
  - OS : Ubuntu 16.04.3 LTS
  - java : openjdk version "1.8.0_151"
  - mvn : Apache Maven 3.3.9 (moteur de production)

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Contribuer une correction de bogue

- Déposer un rapport de bogue dans JIRA, système de ticket du projet.
- Modifier le code source en ajoutant la correction.
  - Code Java doit être formaté selon les conventions de Sun (avec une exception).
  - Contributions doivent passer les tests unitaires existants.
- Nouveaux tests unitaires doivent être fournis pour demontrer les bogues et corrections. Hadoop utilise JUnit v4 comme framework de test.
- Assurez-vous qu'aucun nouvel avertissement du compilateur javac n'est introduit par votre patch.
- Fournir un patch par l'une des manières suivantes :
  - Créer et attacher un diff dans ASF JIRA.
  - Créer un pull request dans GitHub.

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Tests de patch automatisés

- Lorsque vous pensez que votre patch est prêt à être committé, sélectionnez le lien Submit Patch dans le ticket JIRA du problème.
- Les patches soumis seront testés automatiquement par Jenkins, moteur de l'integration continue du projet.
- La configuration du Jenkins utilise Apache Yetus qui
  - automatiquement vérifie les nouvelles contributions par rapport à une variété d'exigences acceptées par la communauté
  - aide les release managers à générer une documentation de release basée sur les informations fournies par les outils de suivi des problèmes de la communauté et les dépôts sources
- Dès la fin des tests, Jenkins/Yetus ajoutera un message de succès ("+1") ou d'échec ("-1") à votre rapport de problème dans JIRA.

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Prochaines étapes

- Une fois un commentaire "+1" est reçu depuis le système de test de patch automatisé et un reviewer de code a défini le drapeau Reviewed dans le ticket JIRA du problème
  - un committer doit ensuite l'évaluer en quelques jours et soit le committer, soit le rejeter avec une explication.
- Si votre patch reçoit un "-1" depuis le système de test Jenkins/Yetus
  - sélectionnez le lien Cancel Patch dans le ticket JIRA du problème
  - téléversez un nouveau patch avec des corrections nécessaires
  - sélectionnez le lien Submit Patch encore.

Patch de code Java

genericqa added a comment - 10/Jan/18 15:51

-1 overall

Vote	Subsystem	Runtime	Comment
0	reexec	9m 51s	Docker mode activated.
			Prechecks
+1	@author	0m 0s	The patch does not contain any @author tags.
+1	test4tests	0m 0s	The patch appears to include 1 new or modified test files.
			trunk Compile Tests
+1	mvninstall	17m 14s	trunk passed
+1	compile	12m 49s	trunk passed
+1	checkstyle	0m 38s	trunk passed
+1	mvnsite	1m 5s	trunk passed
+1	shadedclient	11m 35s	branch has no errors when building and testing our client artifacts.
+1	findbugs	1m 27s	trunk passed
+1	javadoc	0m 52s	trunk passed
			Patch Compile Tests
+1	mvninstall	0m 43s	the patch passed
+1	compile	11m 50s	the patch passed
-1	javac	11m 50s	root generated 1 new + 1240 unchanged - 0 fixed = 1241 total (was 1240)
-0	checkstyle	0m 39s	hadoop-common-project/hadoop-common: The patch generated 17 new + 125 unchanged - 1 fixed = 142 total (was 126)
+1	mvnsite	1m 2s	the patch passed
-1	whitespace	0m 0s	The patch 24 line(s) with tabs.
+1	shadedclient	9m 40s	patch has no errors when building and testing our client artifacts.
+1	findbugs	1m 35s	the patch passed
+1	javadoc	0m 52s	the patch passed
			Other Tests
+1	unit	8m 11s	hadoop-common in the patch passed.
+1	asflicense	0m 33s	The patch does not generate ASF License warnings.
		90m 26s	

Résultats des tests pour le premier patch `MapFile.java` que j'ai fourni dans HADOOP-15151 :

```
javac:
[WARNING] ... [deprecation] cleanup(Log,Clos
eable...) in IOUtils has been deprecated

checkstyle:
Line is longer than 80 characters (found 83)
. [LineLength]
'if' construct must use '{}'s. [NeedBraces]
Name 'SIZE' must match pattern
'^[a-z][a-zA-Z0-9]*$'. [LocalFinalVariableNa
File contains tab characters (this is the
first instance). [FileTabCharacter]
'for' construct must use '{}'s. [NeedBraces]

whitespace:
File contains tab characters (instead of
spaces).
```

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Test checkstyle de code Java

Le test `checkstyle` est effectué avec l'outil Checkstyle, également sur Github, qui

- aide les programmeurs à écrire du code Java conforme à une norme de codage
- est par défaut compatible avec le Guide de style Google Java et les conventions Sun Code, mais est hautement configurable
- est intégré dans le processus de production via maven-checkstyle-plugin

licence : GNU LGPL v2.1

Patch de script shell

genericqa added a comment - 01/Feb/18 14:21

-1 overall

Vote	Subsystem	Runtime	Comment
0	reexec	14m 52s	Docker mode activated.
			Prechecks
+1	@author	0m 0s	The patch does not contain any @author tags.
-1	test4tests	0m 0s	The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch.
			trunk Compile Tests
+1	mvninstall	16m 43s	trunk passed
-1	mvnsite	2m 0s	root in trunk failed.
+1	shadedclient	10m 6s	branch has no errors when building and testing our client artifacts.
			Patch Compile Tests
-1	mvnsite	5m 5s	root in the patch failed.
-1	shellcheck	0m 0s	The patch generated 4 new + 0 unchanged - 0 fixed = 4 total (was 0)
+1	shelldocs	0m 12s	There were no new shelldocs issues.
-1	whitespace	0m 0s	The patch 5 line(s) with tabs.
+1	shadedclient	10m 34s	patch has no errors when building and testing our client artifacts.
			Other Tests
-1	unit	20m 46s	root in the patch failed.
+1	asflicense	0m 25s	The patch does not generate ASF License warnings.
		81m 7s	

L'autre patch que j'ai soumis était pour le script `start-build-env.sh` — pour résoudre le problème de l'accès aux répertoires montés par le Docker sur les systèmes avec SELinux[a] activé mentionné sur la diapositive 5. Comme nous pouvons le voir sur la gauche, les tests Jenkins/Yetus pour HADOOP-15195 ont abouti au échec de `test4tests`, ce qui était dû à de nouveaux tests manquants, et également au échec de `shellcheck` avec les codes d'erreur :

- SC2086 : Double quote to prevent globbing and word splitting.
- SC2012 : Use find instead of ls to better handle non-alphanumeric filenames.

a. Security-Enhanced Linux

ShellCheck pour code shell

Le test `shellcheck` est effectué avec l'outil ShellCheck d'analysis de script shell, également sur Github, qui souligne et clarifie

- problèmes de syntaxe typiques du débutant qui provoquent un shell à donner des messages d'erreur cryptiques
- problèmes sémantiques de niveau intermédiaire typiques qui provoquent un comportement étrange et contre-intuitif
- avertissements subtils, cas limites d'exécution et pièges qui peuvent provoquer l'échec d'un script qui marche d'un utilisateur avancé dans des circonstances futures

licence : GNU General Public License, version 3

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Tests de code shell

Quant à `test4tests`, le reviewer a finalement demandé d'envisager d'écrire des tests unitaires bats pour le nouveau code shell.

- BATS signifie Bash Automated Testing System
  - Bats est un framework de test conforme à TAP pour Bash.
  - Il fournit un moyen simple de vérifier que les programmes UNIX que vous écrivez se comportent comme prévu.
- TAP signifie Test Anything Protocol
- les résultats des tests bats du nouveau code shell ressemblent à

```
1..2
ok 1 start-build-env.sh (Docker without z mount option)
ok 2 start-build-env.sh (Docker with z mount option)
```

Logiciel libre et open source

- Le logiciel du projet Hadoop est publié sous Licence Apache 2.0
- C'est une licence de logiciel libre d'après la définition du logiciel libre : Un programme est un logiciel libre si vous, en tant qu'utilisateur de ce programme, avez les quatre libertés essentielles :
  0. d'exécuter le programme comme vous voulez, pour n'importe quel usage ;
  1. d'étudier le fonctionnement du programme, et de le modifier pour qu'il effectue vos tâches informatiques comme vous le souhaitez ;
  2. de redistribuer des copies, donc d'aider votre voisin ;
  3. de distribuer aux autres des copies de vos versions modifiées ; en faisant cela, vous donnez à toute la communauté une possibilité de profiter de vos changements ;

l'accès au code source est une condition nécessaire pour la liberté 1 et la liberté 3.

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Logiciel libre et open source

En pratique

- Dans notre utilisation de Hadoop, toutes les quatre libertés se sont avérées absolument nécessaires.
- Logiciel libre et open source décrivent presque la même catégorie de logiciel, mais voir l'article de Richard Stallman sur les différences des positions.

Conclusion

- Hadoop est disponible sous une licence de logiciel libre/open source, avec le code source accessible via un dépôt Git
- Hadoop a créé une infrastructure efficace et une ambiance stimulante pour les contributeurs du projet avec
  - l'accès facile à l'environnement complet de production et de test basé sur Docker et un système d'exploitation moderne
  - l'utilisation de compilateurs et d'outils de production modernes — Java 8, Maven 3.3
  - l'utilisation intensive des tests unitaires et des outils avancés d'assurance de la qualité du code — JUnit et Checkstyle pour code Java, BATS et Shellcheck pour code shell
  - l'utilisation systématique et cohérente d'un outil de suivi des problèmes — JIRA
  - le système de test de contributions automatisé — Yetus
  - reviewers de contributions experts et amicaux
- Contributeurs peuvent apprendre et adopter (certaines des) pratiques et outils du projet Hadoop

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE