eBook Gratuit

APPRENEZ apache-pig

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#apache-pig

Table des matières

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: apache-pig

It is an unofficial and free apache-pig ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official apache-pig.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Commencer avec apache-pig

Remarques

Cette section fournit une vue d'ensemble de ce qu'est apache-pig et pourquoi un développeur peut vouloir l'utiliser.

Il convient également de mentionner tous les grands sujets dans Apache-pig et de les relier aux sujets connexes. La documentation pour apache-pig étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Examples

Installation ou configuration

Linux

Exigences (r0.16.0)

Obligatoire

Selon la documentation `Apache-Pig` , il ne prend en charge que `Unix` `Windows` exploitation `Unix` et `Windows` .

- Hadoop 0.23.X, 1.X ou 2.X
- Java 1.6 ou versions ultérieures installées et variable d'environnement JAVA_HOME définie sur répertoire d'installation Java

Optionnel

- Python 2.7 ou plus (UDF Python)
- Ant 1.8 (pour les builds)

Téléchargez la dernière version de Pig

Téléchargez la dernière version de pig depuis http://pig.apache.org/releases.html#Download

Installation

```
mkdir Pig
cd Downloads/
tar zxvf pig-(latest-version).tar.gz
tar zxvf pig-(latest-version).tar.gz
mv pig-(latest-version).tar.gz/* /home/Pig/
```

Configuration

Après avoir installé Apache Pig, vous devez le configurer.

Ouvrez le fichier .bashrc

```
vim ~/.bashrc
```

Dans le fichier .bashrc, définissez les variables suivantes -

```
export PIG_HOME = /home/Pig
export PATH  = PATH:/home/Pig/bin
```

enregistrer le fichier et recharger bashrc dans l'environnement en utilisant

```
. ~/.bashrc
```

Vérification de la version du cochon

```
pig –version
```

Si l'installation réussit, la commande ci-dessus affiche le numéro de version de Pig installé.

Tester l'installation du cochon

```
pig –h
```

Cela devrait afficher toutes les commandes possibles associées à pig

Votre cochon est maintenant installé localement et vous pouvez l'exécuter en utilisant un paramètre local comme

```
pig –x local
```

Connexion à Hadoop

Si Hadoop1.x ou 2.x est installé sur le cluster et que la variable d'environnement HADOOP_HOME est configurée.

vous pouvez connecter pig à Hadoop en ajoutant la ligne dans le fichier .bashrc comme avant

```
export PIG_CLASSPATH = $HADOOP_HOME/conf
```

Porc Courir

Modes d'exécution

Vous pouvez exécuter Pig en utilisant la commande `pig` *(bin / pig)* ou en exécutant le fichier `jar` *(java -cp pig.jar)*

`PIG` scripts `PIG` peuvent être exécutés dans 3 modes différents:

- **Mode local**

```
pig -x local ...
```

- **Mode Mapreduce** (mode par défaut)

```
pig -x mapreduce ...
     (or)
pig ...
```

- **Tez Local Mode**

```
pig -x tez ...
```

Mode interactif

Le cochon peut être exécuté en mode interactif à l'aide du shell `Grunt` . Les instructions et commandes Latin Pig peuvent être entrées de manière interactive dans ce shell.

Exemple

```
$ pig -x <mode> <enter>
grunt>
```

`Mode` peut être l'un des modes d'exécution comme expliqué dans la section précédente.

Temps différé

Le cochon peut également être exécuté en mode batch. `.pig` fichier `.pig` contenant une liste d'instructions et de commandes de porc est fourni.

Exemple

```
$ pig -x <mode> <script.pig>
grunt>
```

De même, le `Mode` peut être l'un des modes d'exécution, comme expliqué dans la section précédente.

Exemple de compte de mots dans Pig

Fichier d'entrée

```
Mary had a little lamb
its fleece was white as snow
and everywhere that Mary went
the lamb was sure to go.
```

Code de nombre de mots de cochon

```
-- Load input from the file named Mary, and call the single
-- field in the record 'line'.
input = load 'mary' as (line);

-- TOKENIZE splits the line into a field for each word.
-- flatten will take the collection of records returned by
-- TOKENIZE and produce a separate record for each one, calling the single
-- field in the record word.
words = foreach input generate flatten(TOKENIZE(line)) as word;

-- Now group them together by each word.
grpd = group words by word;

-- Count them.
cntd = foreach grpd generate group, COUNT(words);

-- Print out the results.
dump cntd;
```

Sortie

```
Mary,2
had,1
a,1
little,1
lamb,2
its,1
fleece,1
was,2
white,1
as,1
snow,1
and,1
everywhere,1
that,1
went,1
the,1
sure,1
to,1
go,1
```

Qu'est-ce que le cochon?

Pig fournit un moteur pour exécuter des flux de données en parallèle sur Hadoop. Il comprend un langage, Pig Latin, pour exprimer ces flux de données. Pig Latin inclut des opérateurs pour de nombreuses opérations de données traditionnelles (jointure, tri, filtrage, etc.), ainsi que la possibilité pour les utilisateurs de développer leurs propres fonctions de lecture, de traitement et d'écriture de données. Pig est un projet open source Apache. Cela signifie que les utilisateurs sont libres de le télécharger en tant que source ou binaire, de l'utiliser pour eux-mêmes, d'y contribuer et, selon les termes de la licence Apache, de l'utiliser dans leurs produits et de le modifier comme ils l'entendent.

Lire Commencer avec apache-pig en ligne: https://riptutorial.com/fr/apache-

pig/topic/3244/commencer-avec-apache-pig

Chapitre 2: Fonctionnement du cube

Examples

CUBE de base

Considérant le cas suivant:

Nous avons des données d'événement utilisateur avec 4 dimensions:

1. Seau de test A / B (prod / test)
2. Type de client (web / mobile)
3. module (commande / rapport)
4. événement (clic / vue)

```
test    mobile    order_module    click
prod    web    order_module    view
prod    mobile    order_module    click
```

Un système de rapports peut vouloir signaler des mesures pour différentes combinaisons, telles que:

1. Quel est le nombre total de **clics** de **l'** utilisateur **mobile** ?
2. Quel est le nombre total de **clics** provenant de différents groupes de tests?
3. ce qui est la **vue** de la page **web** total **order_module** dans le seau **prod?**

Comme vous pouvez le voir, il existe de nombreuses combinaisons. Ce n'est pas tout à fait efficace en temps si nous ne stockons que les plus petites métriques de granularité et que nous les déplions ensuite lors de la réception d'une requête. Une solution consiste donc à **pré-calculer TOUTES les combinaisons** .

Voici comment procéder avec l'opération CUBE de PIG.

```
example = LOAD './cube.example' AS (product:chararray, client:chararray, module:chararray,
action:chararray);

cubed_data = CUBE example BY CUBE(product, client, module, action);

final_data = FOREACH cubed_data GENERATE $0, COUNT_STAR($1);

dump final_data;
```

Il produira des résultats de toutes les combinaisons et de tous les comptes. Voir le résultat du vidage précédent - avec ces statistiques, nous pourrions répondre aux questions précédentes avec une réponse directe. Aucune autre agrégation nécessaire.

```
((prod,web,order_module,view),1)
((prod,web,order_module,),1)
```

```
((prod,web,,view),1)
((prod,web,,),1)
((prod,mobile,order_module,click),1)
((prod,mobile,order_module,),1)
((prod,mobile,,click),1)
((prod,mobile,,),1)
((prod,,order_module,view),1)
((prod,,order_module,click),1)
((prod,,order_module,),2)
((prod,,,view),1)
((prod,,,click),1)
((prod,,,),2)
((test,mobile,order_module,click),1)
((test,mobile,order_module,),1)
((test,mobile,,click),1)
((test,mobile,,),1)
((test,,order_module,click),1)
((test,,order_module,),1)
((test,,,click),1)
((test,,,),1)
((,web,order_module,view),1)
((,web,order_module,),1)
((,web,,view),1)
((,web,,),1)
((,mobile,order_module,click),2)
((,mobile,order_module,),2)
((,mobile,,click),2)
((,mobile,,),2)
((,,order_module,view),1)
((,,order_module,click),2)
((,,order_module,),3)
((,,,view),1)
((,,,click),2)
((,,,),3)
```

Lire Fonctionnement du cube en ligne: https://riptutorial.com/fr/apache-pig/topic/6120/fonctionnement-du-cube

Chapitre 3: Opérateur de chargement

Examples

Chargement des données boursières

Supposons les données boursières suivantes stockées dans `HDFS` . C'est un fichier `CSV` avec des champs: **Symbol, Date, Open, High, Close et Volume** .

```
ABT,20160106,42.310001,42.98,42.209999,42.560001,5906000
BAC,20160201,14.05,14.09,13.8,13.96,105739400
CAS,20160129,1.9,1.97,1.83,1.84,34500
DCA,20160129,3.46,3.54,3.46,3.51,84600
ECL,20160114,103.480003,105.400002,102.480003,104.82,1485000
FAF,20160201,34.040001,34.82,33.939999,34.639999,1222600
TYL,20160201,156.070007,159.550003,155.690002,158.259995,177100
UTL,20160201,38.610001,39.889999,38.57,39.27,119500
VTR,20160128,54.09,54.73,53.549999,53.790001,2441300
WWE,20160201,17.629999,18,17.27,17.799999,734100
XRX,20160104,10.41,10.43,10.13,10.3,9122600
YUM,20160104,71.32,72.25,70.639999,72.209999,3466300
ZTR,20160104,12.1,12.14,11.98,12.11,60200
```

Exemple 1 Une simple instruction `LOAD` pour les données ci-dessus ressemblerait à ceci:

```
stocks = load '/user/pig/stock.txt' using PigStorage(',') as
             (sym:chararray, date:int, open:float, high:float, low:float,
              close:float, vol:int);
```

Chargement de données depuis ElasticSearch

Avant de vérifier la syntaxe spécifique, voyons comment configurer votre environnement pour charger les plug-ins nécessaires.

Installer

Pour charger des données directement depuis ElasticSearch, vous devez télécharger le plug `elasticsearch-hadoop` in `elasticsearch-hadoop` . Vous avez différentes façons de le faire fonctionner, pour une configuration rapide, vous pouvez faire ce qui suit.

Pour que cela fonctionne, vous devez mettre le fichier jar `elasticsearch-hadoop-<version>.jar` dans un dossier du noeud sur lequel vous avez installé le serveur. Dans mon cas, il est très courant de chercher sur Google - je devais aussi ajouter `commons-httpclient-<version>.jar` dans ce dossier.

Ensuite, vous pouvez exécuter `pig` en mode shell (appelé `grunt`) en tapant simplement `pig` sur la console. Ensuite, vous devez charger ces deux pots de la façon suivante:

```
REGISTER /path/to/jars/commons-httpclient-<version>.jar;
REGISTER /path/to/jars/elasticsearch-hadoop-<version>.jar;
```

Vous êtes maintenant prêt à écrire du code.

Exemple

Vérifions la syntaxe pour charger les données d'un cas complexe.

```
DATA = LOAD 'my_index/log' USING org.elasticsearch.hadoop.pig.EsStorage(
'es.nodes=https://server1:port1,https://server2:port2,https://server3:port3',
'es.query=?q=*',
'es.net.ssl=true',
'es.net.http.auth.user=user',
'es.net.http.auth.pass=pass',
'es.net.ssl.keystore.type=JKS',
'es.net.ssl.truststore.location=file:///path/to/truststore.jks',
'es.net.ssl.truststore.pass=pass');
```

C'est l'exemple complet, maintenant analysons-le pas à pas.

- `es.nodes` contient la liste des nœuds de votre cluster ElasticSearch. Vous devez spécifier vos nœuds en tant que liste séparée par des virgules, avec le port associé.
- `es.query` contient la requête qui sera soumise à ElasticSearch pour récupérer des données. Vous pouvez également placer une requête au format DSL, mais veillez à ce que seule la partie correspondante de la requête soit prise en compte! Si vous essayez de limiter le nombre de champs via la requête DSL, cela ne fonctionnera pas: pour y parvenir, vous devez utiliser le paramètre `es.read.source.filter` . exemple d'une requête DSL: `'es.query = { "query":{ "match_all":{} } }'`
- `es.net.ssl=true` est explicite, vous devez également fournir les informations de connexion à ElasticSearch avec `es.net.http.auth.user` et `es.net.http.auth.pass` .
- `es.net.ssl.keystore.type` si vous avez besoin d'un fichier de clés certifiées, vous pouvez sélectionner ici le type. Dans le paramètre `es.net.ssl.truststore.location` , définissez l'emplacement du fichier, veillez à ajouter `file://` préfixe `file://` et, dans le paramètre `es.net.ssl.truststore.pass` , définissez le mot de passe du fichier de clés certifiées. fichier.

Quelques réglages utiles

- `es.read.source.filter=field1,field2,field3` vous permet de ne récupérer que les champs spécifiés dans ElasticSearch (dans cet exemple trois).
- `es.output.json=true` permet d' `es.output.json=true` données dans un format clé-valeur (JSON). Le réglage sur `false` renverra les données au format CSV (par défaut).

Lire Opérateur de chargement en ligne: https://riptutorial.com/fr/apache-pig/topic/7257/operateur-de-chargement

Crédits

S. No	Chapitres	Contributeurs
1	Commencer avec apache-pig	Bhavesh, Brian Armstrong, Community, DhiwaTdG, Mzzzzzz, pratiklodha
2	Fonctionnement du cube	Wenzhong
3	Opérateur de chargement	Andrea Romagnoli, DhiwaTdG