Parallélisme, systèmes distribués et grilles

Manciaux Romain, Hanser Florian

Master informatique, M2 ILC

4 Mars 2019

UFR Mathématique Informatique Strasbourg

UNIVERSITÉ DE STRASBOURG

Table des matières

I. Prélude

Ce rapport présente le projet final de parallélisme, systèmes distribués et grilles. Les sources de ce projet sont disponibles sur ce dépôt git : https://github.com/PamplemousseMR/BeerReduce. Ce dépôt contient un fichier « README.md » décrivant comment compiler le projet et comment exécuter les binaires générées.

II. Données

Le jeu de données est le suivant : https://www.kaggle.com/jtrofe/beer-recipes#recipeData.csv. Ce jeu de données est une liste des bières faites maison (*homemade*) avec leur origine, leur type de brassage et des informations telles que la quantité de sucre d'amorçage utilisée, la couleur ou l'amertume des bières. Il contient 75 000 entrées provenant du site Brewer's Friend.

Les données sont réparties en deux fichiers. Le premier fichier, StyleData.csv, contient la correspondance entre l'identifiant de la bière et le nom du type de bière. Les types de bières sont répertoriés en sept grandes familles, à savoir : Ale, IPA, Stout, Lager, Porter, Bittler et Cider.

Nom de la colonne	Description
Style	Nom du type de bière
StyleId	Identifiant du type de bière

Le second fichier, RecipeData.csv, est le fichier principal comprenant 75 000 recettes faites maison et postées par les utilisateurs du site Brewer's Friend. Chaque bière est décrite grâce à son nom, son pourcentage d'alcool, mais aussi grâce à d'autres attributs décrivant son goût, tel que l'IBU pour mesurer l'amertume de la bière.

Nom de la colonne	Description
BeerID	Identifiant de la recette
Name	Nom de la bière
URL	Adresse de la recette sur le site https://www.brewersfriend.com
Style	Nom du type de bière
StyleID	Identifiant du type de bière
Size(L)	Quantité de bière produite avec la recette
OG	Poids spécifique du moût avant la fermentation
FG	Poids spécifique du moût après la fermentation
ABV	Pourcentage d'alcool
IBU	International Bittering Units pour mesurer l'amertume
Color	Couleur de la bière, de la plus claire à la plus foncée, ex : 40 = noir
BoilSize	Quantité de liquide à ébullition
BoilTime	Temps durant lequel le moût est bouilli
BoilGravity	Poids spécifique du moût avant ébullition
Efficiency	Efficacité de l'extraction du moût de bière - extraire les sucres du grain pendant le moût
MashThickness	Quantité d'eau par livre de graine
SugarScale	Échelle pour déterminer la concentration de solides dissous dans le moût
BrewMethod	Techniques de brassage
PitchRate	Levure ajoutée au fermenteur - M cellules / ml / deg P
PrimaryTemp	Température au stade de fermentation
PrimingMethod	Méthode d'amorçage
PrimingAmount	Quantité de sucre d'amorçage utilisée
UserId	Identifiant utilisateur du site

III. Première question

Le premier algorithme cherche à répondre à la question suivante : ***pour chaque famille de bières (API, ALE), par quelle(s) méthode(s) de brassage obtient-on la plus grande quantité de bières les plus sucrées ?***

1. L'algorithme

L'algorithme répondant à cette question se situe dans « BestIPA.java ». Il est réalisé à l'aide de deux passes de map-reduce détaillées par la suite.

La première passe réalise une étape d'association qui affecte pour chaque style de bière les données utiles pour la suite de l'algorithme, à savoir : le taux de sucre, la méthode de brassage et le nom de la bière. L'étape de réduction cherche donc le taux de sucre le plus élevé dans la liste des bières (pour chacun des styles) et enregistre dans un fichier temporaire le type d'extraction et le nom de la bière dans une liste qui possède comme clef le style de bière.

La seconde passe permet d'obtenir le type d'extraction par laquelle le plus de bières ont été produites. L'association a sensiblement le même but que la passe précédente, mais le réducteur lui, recherche le type de brassage qui propose le plus d'occurrence et enregistre le nom des bières correspondantes.

2. Résultat

Le programme retourne les valeurs suivantes :

```
ALE      extract      : Black IPA, Buck Hork, ...
BITTER   extract      : ESB - BB, The Innkeeper
CIDER    All Grain    : Eplecider, Megan's Medicine, ...
IPA      extract      : D's Hoppin IPA, Liquid Cheg, ...
LAGER    extract      : Sur sirup og honning, Yeast Starter, ...
PORTER   Partial Mash : Yuengling, PB Porter
STOUT    extract      : Imperial milk stout, The Imperial Peanut, ...
UNKNOWN  All Grain    : Mjýd, Oaked Acerglyn (Maple Mead), ...
```

Nous pouvons observer qu'en moyenne, la méthode d'extraction utilisée est « Extract Brewing ». En effet, c'est la forme de brassage utilisée par la plupart des nouveaux brasseurs. Il est donc tout à fait normal de trouver ce résultat en priorité. Néanmoins, les deux autres méthodes sont plus avancées techniquement et c'est grâce à celles-ci que nous pouvons obtenir les bières les plus sucrées.

IV. Seconde question

Dans la suite de ce document, une tranche de couleurs permet de définir un intervalle de couleurs. Dans les données, les couleurs sont classifiées par des nombres décimaux. Nous avons donc décidé d'établir des intervalles dont le pas est de 0.25. Ainsi, la tranche de couleur 11.50 correspond à toutes les couleurs comprises dans l'intervalle `]11.25 ; 11.50]`.

Le second algorithme cherche à répondre aux questions suivantes : ***pour chaque famille de bières (API, ALE) quelles sont les bières ayant consommées le moins d'eau par livre de graine dans leur tranche de couleurs et combien sont-elles ?***

1. L'algorithme

L'algorithme répondant à cette question se situe dans « DarkestBeer.java ». Il est réalisé à l'aide de deux passes de map-reduce détaillées par la suite.

La première passe réalise une étape d'association qui affecte pour chaque tranche de couleurs les données utiles pour la suite de l'algorithme, à savoir : la quantité d'eau par livre de graine, la famille de la bière et le nom de la bière. L'étape de réduction cherche pour une tranche de couleurs la ou les bières ayant consommées le moins d'eau par livre de graine. Les noms des bières, ainsi que leurs familles sont donc restitués pour la tranche de couleurs.

Nous avons donc en sortie de la première passe une liste pour chaque tranche de couleurs de toutes les bières ayant consommées le moins d'eau par livre de graine.

La seconde passe récupère cette sortie pour extraire de leur tranche de couleurs la liste des bières et de leurs familles ayant consommées le moins d'eau par livre de graine et pour associer à la famille le nom de la bière en question. L'étape de réduction compte pour chaque famille le nombre de bières sélectionnées et restitue le nombre et les noms de chaque bière.

2. Résultat

Le programme retourne les valeurs suivantes :

```
ALE      64  : Agave Brown Ale, Cream of Three Crops,…
BITTER   6   : Red Squirrel Best Bitter, Bitter Bride ESB,…
CIDER    4   : Cider, Raspberry Port,…
IPA      47  : N.E N.Z IPA, Heinýpýý Double IPA,…
LAGER    14  : Roasted Chestnut, Oat stotu,…
PORTER   46  : FABB Old Ale Partigyle Porter, Amidst the Black 2.0,…
STOUT    90  : Black Beauty, RIS + Pump,…
UNKNOWN  116 : And the Beat Gose On, St. Louis Biýre de Garde,…
```

Grâce au tableau suivant (Tableau 1), nous pouvons constater dans un premier temps que le résultat peut être utilisé à des fins d'analyse. Effectivement nous pouvons observer que la fréquence d'apparition d'une famille dans le jeu de données n'influe en rien son apparition dans le résultat.

Tableau 1

Famille	Fréquence d'apparition de la famille dans la liste totale	Résultat de l'algorithme	Fréquence du résultat	Evolution de la fréquence
CIDER	0,3%	4	1,0%	3.33
BITTER	3,4%	6	1,6%	0.47
PORTER	3,7%	46	11,9%	3.21
LAGER	5,6%	14	3,6%	0.64
STOUT	8,1%	90	23,3%	2.88
IPA	23,1%	47	12,1%	0.52
ALE	25,5%	64	16,5%	0.65
UNKNOW	30,5%	116	30,0%	0.98

Dans un second temps, nous pouvons remarquer la famille de bières qui produit le plus de bières ayant consommées le moins d'eau par livre de graine pour sa tranche de couleurs est la famille STOUT.

De manière plus générale, nous pouvons également relever une forte augmentation de la fréquence pour les familles STOUT, PORTER et CIDER. Nous pouvons donc conclure que ce sont ces trois familles qui produisent le plus de bières ayant consommées le moins d'eau par livre de graine.

V. Conclusion

Cet exercice était intéressant, car nous avons pu d'une part utiliser la méthode MapReduce et d'autre part nous rendre compte que cette utilisation permet une résolution plus rapide de tels problèmes. En effet, manipuler un grand nombre de données en les distribuant dans un cluster de machines est beaucoup plus rapide qu'un algorithme séquentiel. De plus, Hadoop nous permet de réaliser ce type d'algorithme sur des ordinateurs personnels sans avoir à modifier le code.

Nous avons remarqué que nous pouvions partir de n'importe quel jeu de données pour se poser tout type de questions, puis y répondre aisément grâce à Hadoop. En effet, Hadoop se caractérise par sa prise en main facile. En partant de notre jeu de données relatif aux bières, qui a priori ne soulève aucune interrogation, nous avons pu ainsi dégager deux problématiques et y répondre. Cependant, si se poser des questions et y répondre étaient aisés, encore fallait-il trouver deux questions pertinentes. Ceci a pu nous poser difficulté.