Accélération par pré-agrégations des accès complexes et répétitifs aux Big Data

Nabil El malki

Institut de Recherche en Informatique de Toulouse, Univ. Toulouse III Paul Sabatier
118 Route de Narbonne
F-31062 TOULOUSE CEDEX 9

Nabil.El-Malki@irit.fr

MOTS-CLÉS : *apprentissage automatique, analyse des données, agrégation, k-moyennes*

KEYWORDS: *machine learning, big data analytics, k-means, aggregation*

ENCADREMENT. *Olivier Teste et Franck Ravat*

1. Contexte

L'humanité produit des quantités de données numérisées dans des proportions et avec un rythme sans commune mesure avec le passé. Ces masses de données, désignées communément comme Big Data, sont entreposées dans des clusters de stockage où les données sont plus ou moins structurées. Ces masses de données sont ensuite exploitées par des analystes (« data scientists ») qui utilisent des chaînes complexes de traitements, afin d'extraire les phénomènes contenus dans les masses de données. Ces traitements consistent à explorer les données, les classifier suivant des approches supervisées, semi-supervisées ou encore non supervisées. Par exemple, les données radars dans l'aviation civile sont stockées sous forme binaire par traces radars reconstituées. Un accès répétitif consiste à extraire toutes les trajectoires dans une fenêtre spatio-temporelle de l'espace aérien 3D. Un tel traitement réclame de nombreux accès aux données brutes pour constituer une réponse. L'aspect répétitif est induit notamment lorsque plusieurs requêtes demandent des calculs élémentaires communs répétés sur les données brutes. Par exemple, calculer le nombre de trajectoires par semaine dans une fenêtre de l'espace aérien revient à agréger 7 calculs de trajectoires quotidiennes.

2. État de l'art

Dans cette section nous nous intéressons uniquement aux travaux relatifs aux accès complexes et répétitifs appliqués à des données massives.

Dans le domaine de l'analyse statistique, les auteurs des travaux de (Wasay *et al.*, 2017) proposent un système baptisé Datacanopy qui repose sur un cache intelligent destiné à l'analyse statistique exploratoire. Datacanopy pré-calcule et pré-agrège des calculs statistiques pour éviter les accès répétitifs aux données de base. Pour ce faire, il décompose les calculs en opérations élémentaires et les données de base en blocs unitaires (chunks). Ces chunks sont stockés dans un arbre binaire agrégeant de manière récursive les calculs des feuilles jusqu'à la racine. La construction d'un tel arbre dépend des requêtes utilisateur. Cette contrainte conduit l'approche Datacanopy à supporter uniquement les requêtes respectant cet ordonnancement.

Dans le domaine de l'apprentissage automatique, les méthodes, telles que k-means, ont recours à des accès répétitifs aux données de base. Le but de k-means est de diviser l'ensemble des individus Xi en un certain nombre de classes homogènes défini préalablement par un utilisateur. Cette méthode repose sur un agorithme itératif consistant à intégrer ou déplacer des points dans des classes. L'utilisation de la méthode k-means nécessite un temps d'exécution proportionnel au produit du nombre de classes et du nombre de points par itération. Ce temps d'exécution total est coûteux en terme de calcul, en particulier pour les grands ensembles de données. Par conséquent, l'algorithme de clustering k-means ne peut pas satisfaire le besoin en temps de réponse rapide pour certaines applications. K-means (Lloyd, 1982), effectue des répétitions des calculs de distance et de moyenne sur les mêmes blocs de données. Plusieurs extensions de la version standard du k-means ont été proposées pour accélérer les temps d'exécution :

– Accélération par la parallélisation de l'algorithme via le MapReduce (Li *et al.*, 2015) ou le MPI (Zhang *et al.*, 2013) qui sont des modèles de programmation conçus pour traiter des volumes importants de données de manière parallélisée et distribuée.

– Accélération par réduction du nombre de calculs à effectuer. Les algorithmes d'Elkan (Elkan, 2003) et de Hamerly (Hamerly, 2010) se basent sur la propriété de l'inégalité triangulaire pour éviter de calculer à chaque itération la distance entre un point donné et tous les centres de gravité des classes.

– Accélération par organisation ou structuration des données. Dans les travaux (Hung *et al.*, 2005), les auteurs proposent un algorithme accélérant k-means par découpage du jeu de données en blocs unitaires égaux. Ceux-ci contiennent au moins un individu. Ensuite k-means est déroulé sur les centres de gravité de ces blocs et non sur les points contenus.

Ces extensions accélèrent en temps d'exécution k-means standard mais n'utilisent pas l'approche de pré-agrégation des résultats intermédiaires utilisés dans d'autres contextes et qui pourrait offrir des perspectives d'amélioration intéressantes.

3. Problématique

Les chaînes de traitements induisent des accès parfois complexes et répétitifs aux données, alourdissant sensiblement ces traitements (Wasay *et al.*, 2017). La répétition est provoquée par les méthodes qui sont généralement peu optimisées. Ainsi ils ne conservent pas les calculs communs entre deux requêtes. Par exemple, calculer une variance et une covariance sur une même colonne numérique nécessite de calculer à chaque fois la somme des valeurs contenues dans la colonne. La complexité d'accès est due à la multiplicité des architectures de stockage du big data (BD Nosql, polystore...)(Chevalier *et al.*, 2015). L'objectif de nos travaux est de proposer une approche pour optimiser les accès répétitifs, en considérant différentes architectures, basée sur des pré-calculs agrégeant les calculs intermédiaires répétés.

4. Actions réalisées

Dans un premier temps, nous nous intéressons aux algorithmes d'apprentissage de données (machine learning), plus particulièrement, à l'algorithme de k-means, l'un des algorithmes de clustering couramment utilisé. Nous avons mené deux actions visant à répondre au problème de la répétitivité des accès dans k-means :

– Proposition d'une structure arborescente stockant des pré-agrégats (des moyennes) pour être utilisé par k-means. Lors de l'exécution de ce dernier, il n'effectue pas d'opérations de moyennes mais il récupère les résultats des opérations depuis une structure de données arborescente. Cette solution permet à k-means de ne pas parcourir les données de base pour calculer les moyennes pour réduire le temps de son exécution. La structure est une composition de plusieurs sous-structures dans lesquelles chacune ne stocke que les moyennes des ensembles ayant le même nombre d'entités de base. Les sous-structures dites M2 (regroupement de deux entités de base) et M3 (regroupement de trois entités de base) sont calculées à partir des données de base. Par contre la sous-structure M4 est calculée à partir de M2, celle de M5 à partir de M2 et M3, celle de M6 à partir de M3 et ainsi de suite. Il existe plusieurs chemins de constructions des sous-structures, à titre d'exemple M6 peut être construite de deux M3 ou bien de trois M2.

– L'autre action est la réduction de l'espace de calcul des moyennes, c'est-à-dire que l'on ne calcule a priori que les moyennes susceptibles d'être requises par k-means.

5. Actions futures

L'objectif du projet de thèse est d'aller au-delà de la simple utilisation des outils existants permettant d'explorer et d'analyser les masses de données. La thèse abordera plusieurs aspects scientifiques non résolus dans la littérature :

– Modélisation des données. Pour permettre la manipulation des données, il conviendra dans un premier temps de définir le modèle de représentation des données.

L'enjeu de cette phase est notamment de prendre en compte la grande variabilité des données dans le contexte des Big Data (approche « schemaless »). La modélisation des données a pour enjeu de permettre la définition rigoureuse des composants de base manipulés par les requêtes des utilisateurs, et de définir les mécanismes garantissant des pré-agrégations cohérentes de ces composants.

– Noyau algébrique d'opérateurs élémentaires. Ces accès seront décomposés sur la base d'un noyau algébrique d'opérations élémentaires additives permettant de modéliser la chaîne de traitements complexes par composition de ces opérations élémentaires. Les calculs élémentaires répétés pourront alors être pré-calculés par le système de gestion du Big Data accédé sous la forme de pré-agrégats.

– Apprentissage automatique des pré-agrégats. Des mécanismes d'apprentissage automatique pourront être introduits dans le système. Le but de cet apprentissage sera de permettre une maintenance prédictive des pré-agrégats en fonction de l'évolution des traitements effectués par les utilisateurs. Ces mécanismes confèreront au système des capacités d'adaptation automatique en fonction de l'évolution des traitements mais également de la masse de données.

Nos expérimentations pourront s'appuyer sur la plateforme OSIRIM de l'IRIT, offrant une baie de stockage massif (36 disques de 3To) et d'un cluster de calcul de 640 cœurs, étendus par cartes GPU.

6. Bibliographie

Chevalier M., El Malki M., Kopliku A., Teste O., Tournier R., "Implementation of multidimensional databases with document-oriented NoSQL", *Int. Conf. on Big Data Analytics and Knowledge Discovery, Dawak'15*, p. 379–390, 2015.

Elkan C., "Using the Triangle Inequality to Accelerate k-Means", *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*p. 147–153, 2003.

Hamerly G., "Making k -means even faster", *2010 SIAM international conference on data mining (SDM 2010)*p. 130–140, 2010.

Hung M.-C., Wu J., Chang J.-H., "An Efficient k-Means Clustering Algorithm Using Simple Partitioning", *Journal of Information Science and Engineering 21*, vol. 1177, p. 1157–1177, 2005.

Li Z., Song X., "K-means Clustering Optimization Algorithm Based on MapReduce", , , p. 198–203, 2015.

Lloyd S. P., "Least Squares Quantization in PCM", *IEEE Transactions on Information Theory*, vol. 28, n° 2, p. 129–137, 1982.

Wasay A., Wei X., Dayan N., Idreos S., "Data Canopy", *Proceedings of the 2017 ACM International Conference on Management of Data - SIGMOD '17*p. 557–572, 2017.

Zhang J., Wu G., Hu X., Li S., Hao S., "A Parallel Clustering Algorithm with MPI – MKmeans", *Journal of Computers*, vol. 8, n° 1, p. 10–17, 2013.