

ED STIC - Proposition de Sujets de Thèse pour la campagne d'Allocation de thèses 2015

Axe Sophi@Stic :

Titre du sujet :

Mention de thèse :

HDR Directeur de thèse inscrit à l'ED STIC :

Co-encadrant de thèse éventuel :

Nom :

Prénom :

Email :

Téléphone :

Email de contact pour ce sujet :

Laboratoire d'accueil :

Description du sujet :

1 Contexte et motivation

L'avènement du Big Data a vu une explosion de modèles et d'environnements permettant le traitement d'énormes quantités de données. MapReduce [3] a longtemps été le modèle de prédilection, du fait notamment de l'implémentation open source Hadoop [1]. Cependant, du fait de la grande diversité des données, il a vite montré ses limites. Son approche en traitement par lot (batch) le rend inadapté à des données dynamiques. Son modèle de données, de type clé-valeur, complique, par exemple, l'analyse de grands graphs. Un modèle vertex-centric [5] ou graph-centric[8] est en général préféré. Si les données sont disponibles sous

forme de flux, il est nécessaire d'avoir des environnements très réactifs comme Storm[9] afin d'avoir un traitement proche du temps réel. Cette distinction batch-stream tend à disparaître car une architecture de haut niveau, appelée Lambda Architecture [2], décrit comment combiner à la fois du traitement de flux et du traitement de type batch dans un même environnement.

Dans la majorité des cas le modèle est implémenté sur un nombre limité d'environnements ce qui impose des contraintes fortes sur l'utilisateur. Pour palier à cela, plusieurs approches ont été suivies. La première consiste à proposer des modèles plus génériques et plus flexibles comme par exemple les Resilient and Distributed Dataset de Spark[10]. Une autre consiste à introduire des langages de plus haut niveau comme Pig Latin[6]. Enfin, plus récemment une approche utilisant des workflows qui sont ensuite exécutés sur n'importe quel environnement a montré des résultats intéressants [4].

Le point commun de tout ces travaux est qu'ils supposent un environnement essentiellement statique, le seul aspect dynamique concernant souvent la gestion d'éventuelles pannes à l'exécution. Or en pratique rien n'est vraiment statique. Les données traitées peuvent varier considérablement en taille et en complexité, ce qui implique que l'utilisateur doit estimer les ressources dont il aura besoin à priori afin d'obtenir de bonnes performances.

De nombreux travaux autour de la performance de jobs Hadoop ont montré que c'était possible mais non trivial [7]. De plus, les contraintes énergétiques font qu'il peut être aussi utile de restreindre le nombre de machines utilisées au cours d'un calcul ou au contraire ajouter des machines intrinsèquement parallèles qui présentent une meilleure efficacité énergétique. En pratique, il y a donc de la dynamique introduite par les données mais aussi par l'environnement.

2 Objectifs et plan

Ce travail de thèse s'intéresse à la dynamique dans les Lambda Architecture. Le candidat pourra étudier de nouveaux modèles de programmation permettant de tenir compte de la dynamique dès la conception du code métier.

Combiné à une approche autonome pour l'usage des ressources, le but est de pouvoir assurer une qualité de service selon des critères comme le temps d'exécution ou la consommation énergétique.

Dans un premier temps, nous souhaitons étudier l'ajout ou la suppression dynamiques de noeuds dans un cluster Spark ou Storm en fonction de besoins spécifiés par l'utilisateur. L'idée est de passer par un Cloud afin d'avoir

accès à des ressources qui ne sont pas localement disponibles. Ensuite ce mécanisme pourra être étendu afin de permettre l'élasticité durant l'exécution du job et non plus lors de sa soumission.

Des expérimentations nous permettront ensuite de quantifier l'impact du modèle de programmation sur le coût de reconfiguration à l'exécution et proposer des améliorations. À plus long terme, il sera intéressant

d'étendre et généraliser ces résultats pour des Lambda Architecture.

References

- [1] Hadoop <https://hadoop.apache.org/>.
- [2] Lambda architecture <http://lambda-architecture.net/>.
- [3] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.
- [4] Ionel Gog, Malte Schwarzkopf, Natacha Crooks, Matthew P. Grosvenor, Allen Clement, and Steven Hand. Musketeer: All for one, one for all in data processing systems. In *Proceedings of the Tenth European Conference on Computer Systems, EuroSys '15*, pages 2:1–2:16, New York, NY, USA, 2015. ACM.
- [5] Grzegorz Malewicz, Matthew H. Austern, Aart J.C Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: A system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10*, pages 135–146, New York, NY, USA, 2010. ACM.
- [6] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: A not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08*, pages 1099–1110, New York, NY, USA, 2008. ACM.
- [7] Ge Song, Zide Meng, Fabrice Huet, Frederic Magoules, Lei Yu, and Xuelian Lin. A Hadoop MapReduce Performance Prediction Method. In *HPCC 2013*, pages 820–825, Zhangjiajie, China, November 2013.
- [8] Yuanyuan Tian, Andrey Balmin, Severin Andreas Corsten, Shirish Tatikonda, and John McPherson. From "think like a vertex" to "think like a graph". *Proc. VLDB Endow.*, 7(3):193–204, November 2013.
- [9] Ankit Toshniwal, Siddarth Taneja, Amit Shukla, Karthik Ramasamy, Jignesh M. Patel, Sanjeev Kulkarni, Jason Jackson, Krishna Gade, Maosong Fu, Jake Donham, Nikunj Bhagat, Sailesh Mittal, and Dmitriy Ryaboy. Storm@twitter. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, SIGMOD '14*, pages 147–156, New York, NY, USA, 2014. ACM.
- [10] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing, HotCloud'10*, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.

English version: