

Année 2018-2019
Proposition de sujet de stage M2

CALCULS LOCAUX POUR DONNÉES DISTRIBUÉES

LABORATOIRES : LIRIS

Bât Nautibus, campus de la Doua
25 avenue Pierre de Coubertin
69100 Villeurbanne France

LIP6

Sorbonne Université
4 place Jussieu
75005 Paris

PERSONNES ENCADRANT LE STAGE :

| | | | |
|---------|--|----------------|-------------------|
| | Emmanuel Coquery | Xavier Urbain | Sébastien Tixeuil |
| Tél. : | 04 72 44 58 25 | 04 27 46 57 07 | 01 44 27 87 62 |
| Email : | {emmanuel.coquery, xavier.urbain}@univ-lyon1.fr sebastien.tixeuil@lip6.fr | | |

MOTS-CLEFS : algorithmique distribuée, démonstration automatique, bases de données.

CONTEXTE ET OBJECTIFS SCIENTIFIQUES

Les systèmes de gestion de données de type No/NoSQL se sont de nos jours imposés pour la gestion des données dans le cadre du BigData. La volumétrie mise en œuvre dans ces systèmes nécessite l'utilisation de plusieurs machines, ne serait-ce que pour le stockage brut des données.

Lors de l'utilisation d'un système de gestion de données, on peut souhaiter bénéficier d'un certain nombre de propriétés. Une première propriété est la cohérence des données, à savoir que les lectures reflètent les écritures qui les précèdent. Une deuxième propriété est la disponibilité, c'est-à-dire que le système répondra aux différentes requêtes qui lui sont adressées en un temps fini. Une troisième est que, si on utilise un système de gestion des données distribué, ce dernier soit tolérant aux partitions, c'est-à-dire résistant aux pertes de messages entre les nœuds du réseau. Un théorème connu [2] établit cependant l'impossibilité d'obtenir ces trois propriétés en même temps. On s'intéresse donc à des compromis, en général en affaiblissant la notion de cohérence.

Une des caractéristiques de l'écosystème No/NoSQL est la diversité des solutions disponibles et leur côté ad hoc, correspondant à des compromis variés en termes de cohérence [1]. À cause de cette diversité, il est intéressant de pouvoir automatiser les approches de formalisation et de preuve du bon comportement de ces systèmes.

On s'intéresse à l'application d'un outil théorique et pratique puissant pour l'étude des entrepôts distribués : les calculs locaux, lesquels vérifient de bonnes propriétés pour la compréhension, l'étude et l'analyse des systèmes distribués. Ils sont en effet suffisamment intuitifs pour permettre le développement de programmes et complètement formels, autorisant ainsi une démarche de certification. On peut les représenter par des réseaux de processeurs n'ayant accès qu'à des ressources locales, c'est-à-dire les états des voisins et des canaux qui y mènent. Le comportement d'un algorithme est donc déterminé par l'état initial du réseau, ses propriétés locales et globales, et enfin les primitives de communication. Les calculs locaux sont en fait des *systèmes d'interactions locales*. Parmi ces modèles, les systèmes de *réécriture de graphes étiquetés* de Litovsky, Métivier et Mosbah [4, 3] bénéficient du haut niveau d'abstraction apporté par la réécriture et ont permis l'obtention d'importants résultats (par exemple un algorithme d'élection de leader totalement asynchrone ne nécessitant que des messages de taille polynomiale). Ce sont eux que nous nous proposons d'utiliser dans ce projet.

TRAVAUX

Ce stage concerne l'expression et l'analyse, en termes de réécritures (de graphes puis de termes), des notions pertinentes pour les données distribuées.

À des fins d'automatisation, une partie de ces traductions sera implantée dans une boîte à outils logicielle pour systèmes de réécriture.

On étudiera les systèmes produits afin de développer des techniques automatiques de preuve de propriété (par exemple de type *terminaison*).

Ces travaux se poursuivent en thèse, une piste parmi d'autres étant la vérification formelle des propriétés et donc la validation formelle de solutions de cohérence à terme.

CONTEXT AND SCIENTIFIC GOALS

No/NewSQL-like Database Management Systems are nowadays the norm when it comes to BigData management. In such a demanding context, several machines are required, if only for storing raw data.

Users of such DBMS may ask for it to enjoy some desirable properties. A first one is *Data consistency*, ensuring that reads conform to preceding writes. A second one is *availability*, ensuring that any request is fulfilled in finite time. A third one is *partition resilience*, that is robustness to the loss of messages between nodes in the network. However, a well-known theorem [2] states that these three properties cannot coexist. One may then compromise by weakening the notion of consistency, leading to, for instance, *eventual consistency*.

The No/NewSQL ecosystem proposes various solutions, remarkable for their diversity and a quite *ad hoc* aspect; they address different approaches for the weakening of consistency [1]. Automating those approaches, and hence verifying the correct behaviour of such DBMS is thus important, if not crucial.

To study properties of distributed data stores, we focus on a theoretical and practical model : *local calculi*, which have shown invaluable for the analysis of distributed systems. They indeed enjoy an intuitive formalism, as well as a completely formal aspect, enabling a certification approach. Roughly, they consist in networks of processors that can access local-only resources, that is the states of their neighbours and the relevant communication channels. the behaviour of an algorithm is thus determined by the network's initial state, its local and global properties, and communication primitives. Local calculi are in fact *local interaction systems*; amongst those, *graph relabelling systems* [3, 4] enjoy the high level of abstraction brought by rewriting, and yield important results (*e.g.* an asynchronous leader election requiring for polynomial-size messages only).

This internship is dedicated to expressing and analysing relevant concepts and properties of distributed databases, in terms of (first graph-, then term-) rewriting.

The focus will be on the *automation* of the proof for some of those properties (for example, but not limited to, various notions of termination in that particular context). Hence, implantation into a rewriting toolbox, and analysis, research for, and design of dedicated proof techniques are expected.

This work can extend into a PhD subject, an example of research path being the formal certification of the proposed approaches and, thus, automation and formal validation of wide-scale solutions to eventual consistency.

COMPÉTENCES :

- Démonstration automatique et fondements de la programmation.
- Programmation fonctionnelle (Ocaml est un plus)
- Algorithmique distribuée.

Références

- [1] Philip A. Bernstein and Sudipto Das. Rethinking eventual consistency. In Kenneth A. Ross, Divesh Srivastava, and Dimitris Papadias, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, pages 923–928. ACM, 2013.
- [2] Seth Gilbert and Nancy A. Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2) :51–59, 2002.
- [3] Igor Litovsky and Yves Métivier. Computing Trees with Graph Rewriting Systems with Priorities. *Tree Automata and Languages*, pages 115–139, 1992.
- [4] Igor Litovsky, Yves Métivier, and Éric Sopena. Graph Relabelling Systems and Distributed Algorithms. In Hartmut Ehrig, Hans-Jörg Kreowski, Ugo Montanari, and Grzegorz Rozenberg, editors, *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 3, pages 1–56. World Scientific, 1999.