

## Progress Report and CSI Report – 2018-19

---

### Administrative information

**PhD student:** Matthieu Nicolas

**Current PhD year in 2018-19:** 2A

**Laboratory:** Loria

**Supervisor(s):** Olivier Perrin et Gérald Oster

**Title of the PhD thesis:** Efficient (re)naming in Conflict-free Replicated Data Types (CRDTs)

**Fundings for the PhD thesis:** Contrat doctoral

#### Members of the CSI:

- **scientific member:** Stephan Merz
- **auxiliary member:** Ye-Qiong Song

## Progress report

### Short description of the subject:

Dans le cadre de ma thèse, j'étudie et travaille sur les Conflict-free Replicated Data Types (CRDTs). Les CRDTs sont de nouvelles spécifications des types abstraits de données, tels que l'*Ensemble* ou la *Séquence*. Contrairement aux spécifications traditionnelles, les CRDTs sont conçus pour supporter nativement les modifications concurrentes. Pour ce faire, ces structures de données intègrent un mécanisme de résolution de conflits directement au sein de leur spécification. Cette spécificité rend les CRDTs particulièrement adaptés pour concevoir des systèmes distribués hautement disponibles dans lesquels différents noeuds répliquent et modifient une même donnée sans aucune coordination.

Pour résoudre les conflits de manière déterministe, les CRDTs utilisent généralement des identifiants qu'ils associent aux éléments stockés au sein de la structure de données. Cependant, selon le type de CRDT, les identifiants doivent respecter un ensemble de contraintes telles qu'être unique ou appartenir à un espace dense. Dans certains cas, ces contraintes empêchent de borner la taille des identifiants. La taille des identifiants croît alors continuellement avec le nombre de modifications effectuées.

Ces identifiants représentent donc un surcoût lié à l'utilisation des CRDTs par rapport aux structures de données traditionnelles. Ce surcoût, parfois non-borné, peut décourager l'adoption des CRDTs dans les systèmes distribués. Le but de cette thèse est de proposer des solutions pour pallier ce problème. Une approche possible est d'étudier ce sujet comme étant un cas particulier du problème du renommage. Il convient alors de proposer un mécanisme permettant de renommer les identifiants pour réduire leur taille, sans introduire la nécessité d'une coordination entre les différents noeuds.

### Main results of the year:

**Travaux précédents.** Au cours de ma 1ère année, j'ai étudié principalement un CRDT donné, *LogootSplit*. Ce CRDT permet de représenter une séquence répliquée et est notamment utilisé dans des applications d'édition collaborative. *LogootSplit* souffre particulièrement du problème de croissance non-bornée des identifiants qu'il utilise.

Afin d'adresser ce problème, j'ai proposé une première version d'un nouveau CRDT, *RenamableLogootSplit*. Ce CRDT associe *LogootSplit* à un mécanisme de renommage. Ce mécanisme vise à réduire le surcoût de *LogootSplit* en réallouant ponctuellement des identifiants plus petits à l'ensemble des éléments stockés au sein de la séquence. Pour gérer les modifications concurrentes à un renommage, *RenamableLogootSplit* utilise des fonctions de transformation. Ces fonctions permettent de transformer les modifications avant de les appliquer à la copie locale pour prendre en compte l'effet du renommage.

Cette première version était néanmoins incomplète: elle n'était pas capable d'intégrer des renommages concurrents. Pour l'utiliser en l'état, il était donc nécessaire d'empêcher des renommages concurrents d'être générés, notamment en reposant sur un mécanisme de coordination.

Les mécanismes de coordination étant coûteux et peu adaptés aux systèmes dynamiques, il s'agissait du défaut principal de ce prototype.

**Travaux en cours.** Cette année, j'ai poursuivi mes travaux sur *RenamableLogootSplit* en concevant et implémentant une nouvelle version de ce dernier. La principale évolution de cette nouvelle version est la capacité de gérer des renommages concurrents. Pour cela, *RenamableLogootSplit* établit un ordre de priorité entre des renommages concurrents. Pour déterminer la priorité d'un renommage par rapport à un autre, différentes méthodes sont possibles. Par exemple, nous pouvons établir un ordre entre deux renommages en comparant leur couple *(identifiant de l'auteur, valeur courante de l'horloge logique)* respectif. Une autre approche possible pour créer un ordre repose sur l'utilisation de métriques telles que la taille du document renommé ou le nombre de modifications effectuées précédemment. En se basant sur l'ordre établi, *RenamableLogootSplit* n'applique en finalité qu'un seul renommage d'un ensemble de renommages concurrents. De nouvelles fonctions de transformation permettent d'annuler l'effet d'un renommage précédemment appliqué, s'il s'avère finalement qu'il existe un renommage concurrent prioritaire.

Cette solution présente toutefois un défaut. Afin que les différentes fonctions de transformation soient déterministes, elles reposent sur l'état sur lequel s'est appliqué le renommage à prendre en compte ou à annuler pour décider de la transformation à effectuer. Il est donc nécessaire de conserver les différents états renommés tant qu'ils peuvent être requis pour transformer une modification concurrente ou qu'un renommage concurrent prioritaire peut être reçu. J'ai donc étudié *RenamableLogootSplit* pour identifier les conditions nécessaires pour pouvoir supprimer définitivement les états renommés et concevoir un mécanisme de *garbage collection* adapté.

Avec l'aide d'un ingénieur de notre équipe, nous sommes en train de mettre en place plusieurs expériences pour valider cette contribution et la comparer à l'existant. La première expérience a été conçue pour pouvoir simuler une collaboration à grande échelle à l'aide de bots, en utilisant soit *LogootSplit*, soit *RenamableLogootSplit*. Les objectifs de cette expérience sont multiples. Tout d'abord, elle nous permet d'éprouver le bon fonctionnement de l'application en vérifiant que les différents noeuds convergent bel et bien à terme. Nous utilisons aussi cette expérience pour générer et collecter un grand nombre d'états différents, qu'ils s'agissent d'états intermédiaires ou finaux. Ces états sont utilisés dans le cadre d'une seconde expérience pour mesurer la consommation en mémoire du *CRDT*, mais aussi le temps d'intégration des modifications. Ceci nous permet de suivre et de comparer l'évolution de ces données au fur et à mesure de la progression de la collaboration pour chaque *CRDT*.

Finalement, afin de présenter cette contribution à la communauté scientifique, je rédige actuellement un article décrivant son fonctionnement et comparant ses performances par rapport à l'existant.

### **Plan for next year:**

L'objectif de cette année est de finir la rédaction et de publier l'article présentant *RenamableLogootSplit*.

Afin de compléter la partie validation de cet article, nous allons poursuivre les expériences

que nous menons pour effectuer des mesures de performances de *RenamableLogootSplit* sur différents aspects: temps d'intégration du renommage, temps d'intégration des modifications concurrentes à un renommage, évolution de la mémoire utilisée par la structure de données... Ces mesures seront utilisées pour comparer les performances globales de *RenamableLogootSplit* par rapport à l'existant.

Pour garantir la convergence à terme des différents noeuds du système mais aussi pour garantir le respect de l'intention de l'utilisateur, les fonctions de transformation doivent vérifier plusieurs propriétés. Elles doivent notamment préserver l'unicité des identifiants ainsi que l'ordre existant entre les identifiants. Ces fonctions étant critiques pour le bon fonctionnement du système, je souhaite prouver formellement leur correction.

Concernant la suite de mes travaux, j'étudie actuellement la littérature sur les CRDTs pour identifier d'autres structures de données qui bénéficieraient d'une adaptation du mécanisme de renommage proposé pour *LogootSplit* ou, le cas échéant, pour identifier d'autres problématiques de recherche liées à l'utilisation des CRDTs dans les systèmes distribués.

### **Publications:**

- [1] Matthieu Nicolas. Efficient renaming in CRDTs. In *Middleware 2018 - 19th ACM/IFIP International Middleware Conference (Doctoral Symposium)*, Rennes, France, December 2018.

### **Project after the thesis:**

J'envisage actuellement soit de poursuivre une carrière académique en tant qu'enseignant-chercheur, soit de m'orienter vers une carrière d'ingénieur R&D au sein de l'académie ou de l'industrie.

### **Scientific and professional modules validated:**

#### **Scientific modules**

- Réplication et cohérence des données
- Participation à l'école d'été SATIS 2018

#### **Professional modules**

- Fi4 152 E Sauveteur Secouriste du Travail (SST)
- Fi4 162 C Formation à la communication orale et corporelle en milieu professionnel
- Fi4 282 Outils numériques pour la pédagogie (plateforme Arche, studio professeur)
- Fi4 305 Culture de l'intégrité scientifique
- PA1.1 MDD 14 - Eléments d'innovations pédagogiques

### **Date and signature of the PhD student**

.....

## Opinion of the supervisors

Opinion on this progress report: .....

Agreement for an additional year? .....

Date of the defense: .....

Date and signature of the PhD supervisors

.....

## Report of the Comité de Suivi Individuel

### Is the student confident with the PhD progression?

Partiellement: les travaux sur la conception d'un algorithme de renommage pour la structure de données *LogootSplit*, utilisée dans le cadre de l'édition collaborative, a pris plus de temps que prévu, en particulier pour gérer des renommages concurrents dans un système totalement non coordonné. Cet algorithme a été présenté lors du symposium doctoral associé à la conférence *Middleware 2018* mais n'a pas encore été publié formellement. Matthieu se pose quelques questions : comment valoriser ce résultat au mieux, comment le généraliser à d'autres structures de données et plus généralement sur la suite de la thèse.

### Is the PhD on good tracks?

Le travail accompli me semble être substantiel. Matthieu a fait deux contributions : d'abord l'algorithme de renommage pour rendre plus compacte la structure de données représentant un document partagé suite à ses modifications par des utilisateurs distribués, puis la gestion de renommages concurrents sans coordinateur. La première contribution devrait intéresser toute la communauté car elle semble répondre à un défi de performance (des évaluations plus poussées de performance sont en préparation). L'intérêt de la deuxième contribution est peut-être plus académique mais le défi algorithmique est réel. Un (ou deux) article(s) décrivant ces contributions devraient être soumis sans tarder.

Pour la suite de la thèse, il semble être envisagé d'ouvrir un nouvel axe de recherche afin d'élargir le travail sur les CRDT et d'en trouver d'autres applications. Cela est probablement une bonne idée afin de ne pas rester braqué sur un algorithme particulier. Je recommande en conséquence de boucler rapidement les travaux sur *RenamableLogootSplit* – en particulier en ce qui concerne l'étude de performance. L'idée d'une preuve formelle de l'algorithme devrait être abandonnée et un projet concret pour ce nouvel axe devrait être élaboré avant l'été.

Dans tous les cas, une soutenance à la fin de la 3ème année paraît peu réaliste, et Matthieu en est d'accord.

### Is the professional project sound?

Le projet professionnel reste encore flou: Matthieu semble s'orienter plutôt vers une carrière dans l'industrie après avoir été attiré par une carrière académique dans un premier temps.

### Conclusion:

Je ne vois pas de problème pour une réinscription en 3A et j'encourage les parties prenantes à définir des objectifs réalistes pour la suite de la thèse.

**Date:** 4 juin 2019

### Signatures:

PhD student

Two members of the CSI

.....

.....