Ré-identification sans coordination dans les types de données répliquées sans conflits

Matthieu Nicolas (matthieu.nicolas@loria.fr)

20 décembre 2022

Rapporteurs :	Hanifa Boucheneb	Professeure, Polytechnique Montréal
	Davide Frey	Chargé de recherche, HdR, Inria Rennes Bretagne-Atlantique
Examinateurs :	Hala Skaf-Molli	Maîtresse de conférences, HdR, Nantes Université, LS2N
	Stephan Merz	Directeur de Recherche, Inria Nancy - Grand Est
Encadrants :	Olivier Perrin	Professeur des Universités, Université de Lorraine, LORIA
	Gérald Oster	Maître de conférences, Université de Lorraine, LORIA

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Inria

cnrs
dépasser les frontières

Applications collaboratives

TODO : Voir comment représenter une appli collaboratice à ce stade

- Un système collaboratif est un système supportant ses utilisateur-rices dans leurs processus de collaboration pour la réalisation de tâches.

Démocratisation des applications collaboratives

TODO : Voir comment illustrer ce point. Screenshots et nombre d'utilisateurs en-dessous ?

Avantages d'une architecture basée sur le cloud…

TODO : Représenter collaboration via appli basée sur le cloud

- Disponibilité : Répond aux utilisateur-rices
- Tolérance aux pannes : Fonctionne malgré pannes
- Capacité de passage à l'échelle : Supporte activité massive

… et ses limites

TODO : Illustrer chacune des propriétés

- Confidentialité
- Souveraineté
- Pérennité
- Résistance à la censure

Pouvons-nous concevoir des applications collaboratives satisfaisant l'ensemble de ces propriétés ?

Applications collaboratives pair-à-pair [1]

TODO : Illustrer une appli P2P

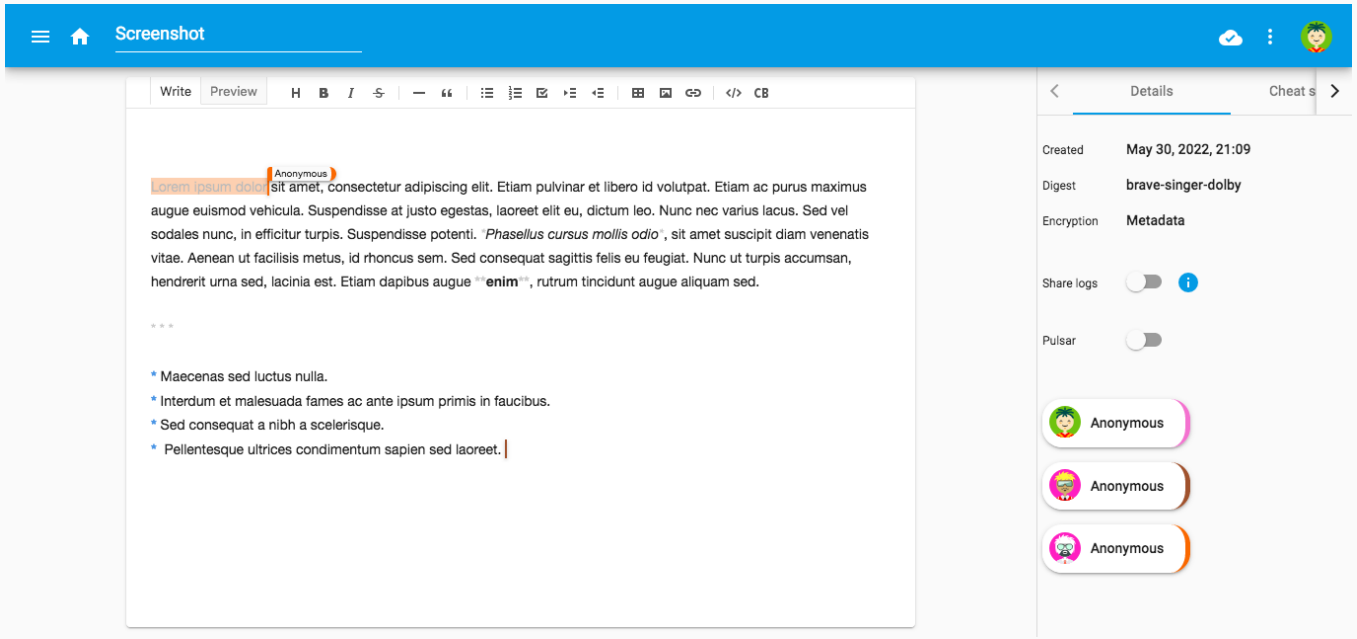
Problématiques

En l'absence d'autorités centrales, comment

- résoudre les conflits de modifications ?
- authentifier les utilisateur-rices ?
- sécuriser les communications ?

[1]. Kleppmann et al., « Local-First Software : You Own Your Data, in Spite of the Cloud ».

MUTE *



- Éditeur de texte collaboratif P2P temps réel chiffré de bout en bout
- Permet à l'équipe d'étudier et contribuer sur les problématiques des applications Local-First Software (LFS)

*. Disponible à : `https://mutehost.loria.fr`

Mes contributions

TODO : À retravailler pour faire apparaître les problématiques ?
TODO : Ajouter comparaison des modèles de sync et approches pour CRDTs pour le type Séquence ?

- Conception d'un nouveau Conflict-free Replicated Data Type (CRDT) pour le type Séquence
- Implémentation et intégration de CRDTs dans MUTE

Type de données *Séquence*

TODO : Représenter une séquence, puis une exécution provoquant un conflit

- Représente une collection ordonnée et de taille dynamique d'éléments
- Deux opérations de modifications, *insert* et *remove*
- Modifications ne commutent généralement pas

Un mécanisme de résolution de conflits est nécessaire pour résoudre les divergences

Conflict-free Replicated Data Types (CRDTs) [2]

- Nouvelles spécifications des types de données
- Types de données répliquées
  - Permettent modifications sans coordination
  - Garantissent la convergence forte

Convergence forte

Ensemble des noeuds ayant intégrés le même ensemble de modifications obtient des états équivalents, sans nécessiter d'actions ou messages supplémentaires

[2]. Shapiro et al., « Conflict-Free Replicated Data Types ».

CRDTs et théorie des treillis [3]

- Reposent sur la théorie des treillis

TODO : Représenter un sup-demi-treillis

- États ordonnés par relation $\leq$, avec état minimal $\bot$
- Modification d'un état génère état supérieur ou égal d'après $\leq$
- Existe fonction qui fusionne deux états et retourne leur borne supérieure

[3]. Davey et al., *Introduction to Lattices and Order*.

La fonction de fusion d’états est un mécanisme de résolution de conflits automatique

Caractéristiques d'un CRDT

Ce qui définit un CRDT

- Sémantique de son mécanisme de résolution de conflits
- Modèle de synchronisation utilisé

Se focalise sur

- Spécification forte des séquences répliquées avec entrelacements [4]
- Synchronisation par opérations

[4]. Attiya et al., « Specification and space complexity of collaborative text editing ».

LogootSplit [5]

- CRDT pour le type Séquence
- Appartient à l'approche à identifiants densément ordonnés

[5]. André et al., « Supporting Adaptable Granularity of Changes for Massive-Scale Collaborative Editing ».

Approche à identifiants densément ordonnés

- Assigne un identifiant de position à chaque élément de la séquence
- Utilise les identifiants des éléments pour les ordonner les uns par rapport aux autres

Propriétés des identifiants de position [6]

- Unique
- Immuable
- Ordonnable par une relation d'ordre strict total $<_{id}$
- Appartenant à un espace dense

[6]. Preguica et al., « A Commutative Replicated Data Type for Cooperative Editing ».

Identifiant LogootSplit

- Composé d'un ou plusieurs tuples $pos_{offset}^{nodeId\ nodeSeq}$
- TODO : Ajouter animations pour expliciter le rôle de chaque composant

Exemple :

$$i_0^{A1} r_3^{B1} \cdots m_0^{A2}$$

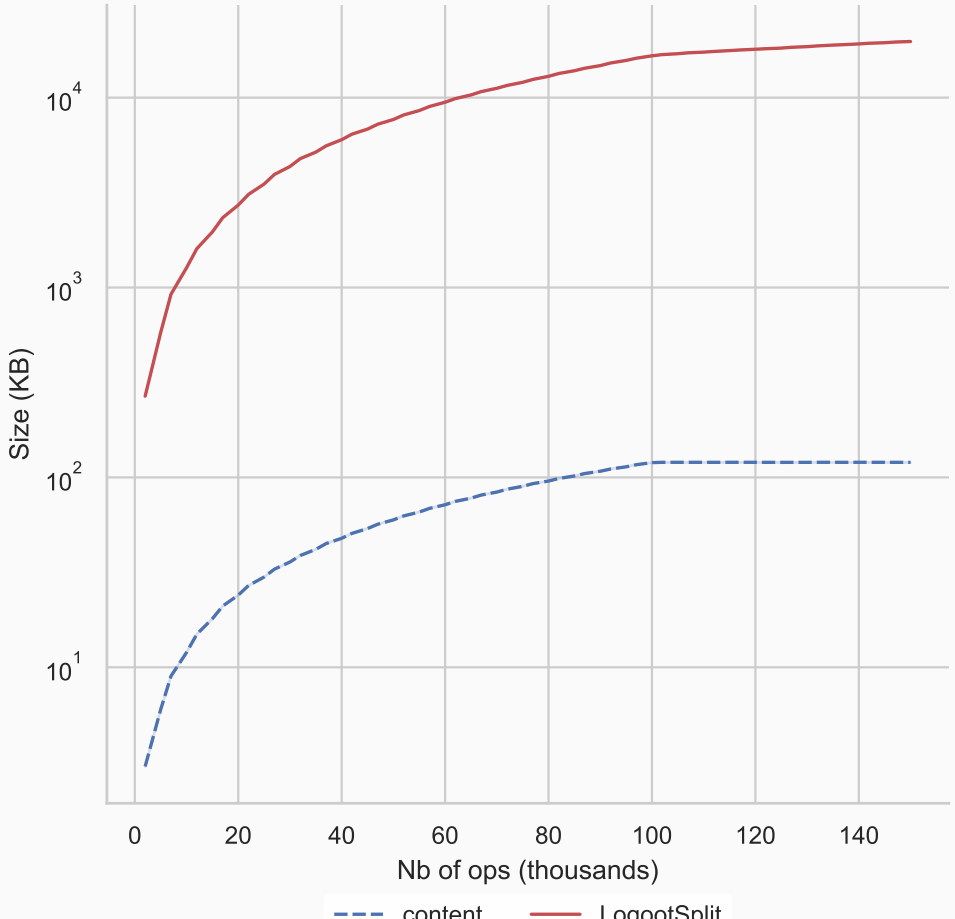
???

TODO : Reprendre exemple de divergence, mais avec LogootSplit pour montrer convergence

Limites de LogootSplit

- Croissance non-bornée de la taille des identifiants
- Fragmentation en blocs courts

Taille du contenu comparé à la taille de la séquence LogootSplit



1% contenu, 99% métadonnées

Mitigation du surcoût des CRDTs pour le type Séquence

L'approche core-nebula [7]

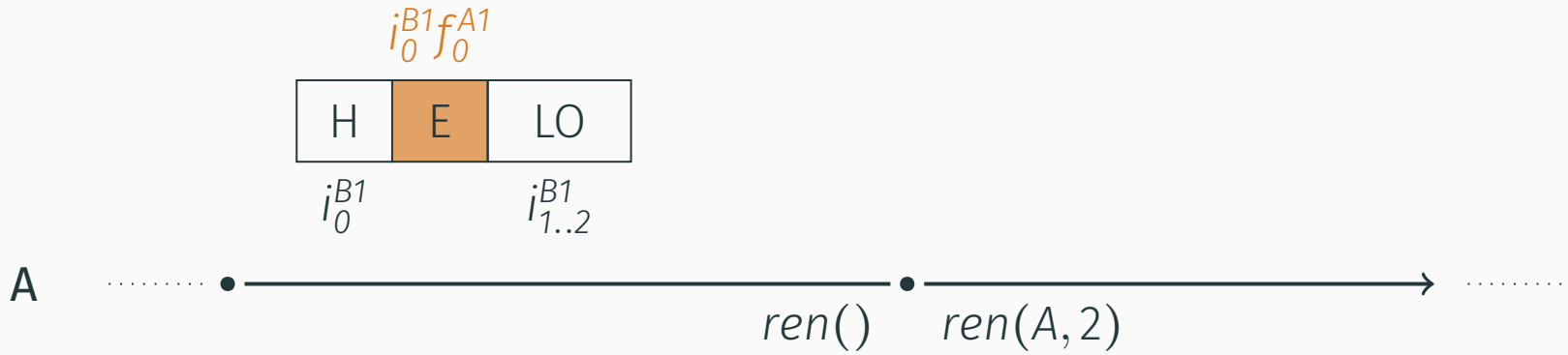
- Ré-assigne des identifiants courts aux éléments, c.-à-d. les *renomme*
- Nécessite de transformer les opérations *insert* et *remove* concurrentes...
- ...et consensus pour effectuer le renommage

Inadaptée aux systèmes P2P sujets au churn

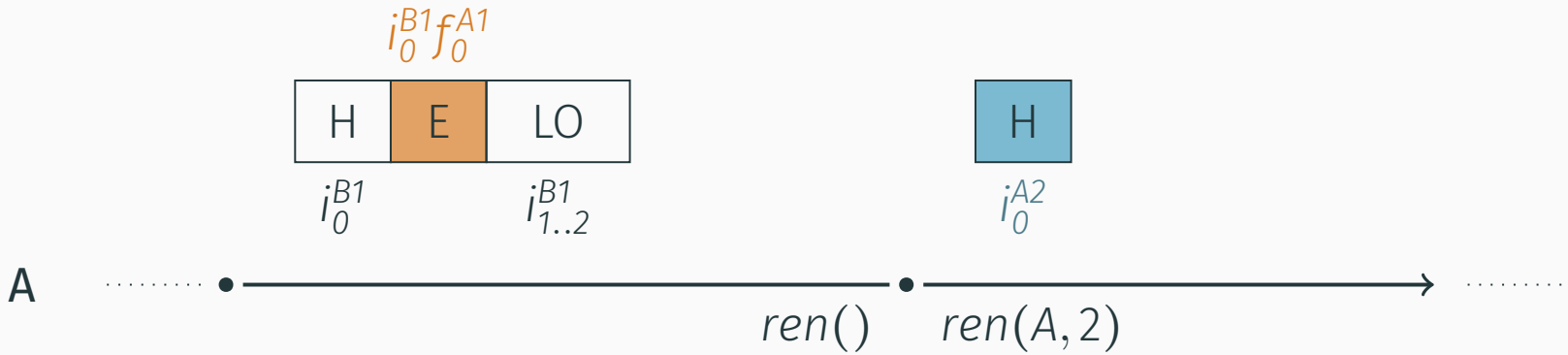
[7]. Zawirski et al., « Asynchronous rebalancing of a replicated tree ».

Pouvons-nous proposer un mécanisme de réduction du surcoût des CRDTs pour le type Séquence adapté aux applications LFS, c.-à-d. sans coordination synchrone ?

Opération *rename*

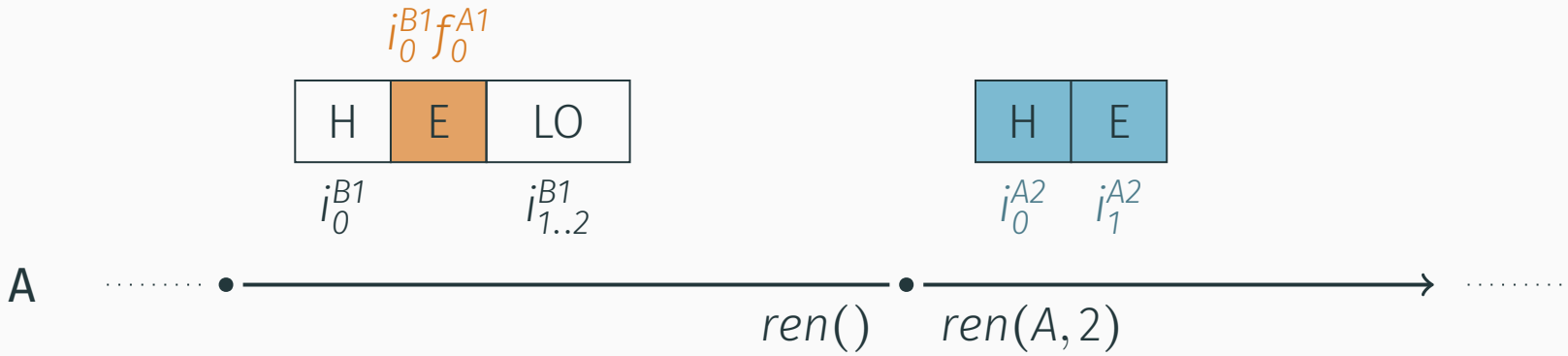


Opération *rename*



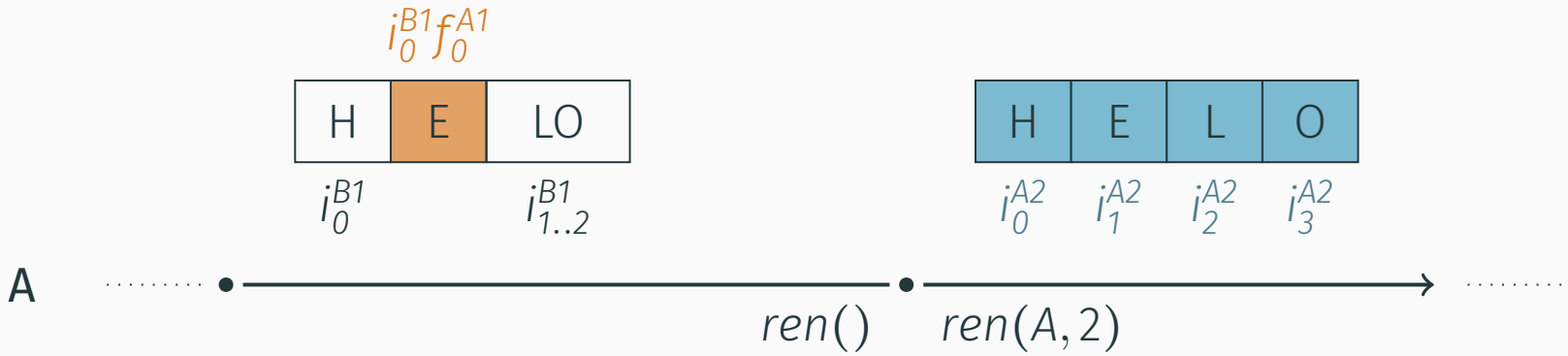
- Génère nouvel identifiant pour le 1er élément : $i_0^{B1} \rightarrow i_0^{A2}$

Opération *rename*



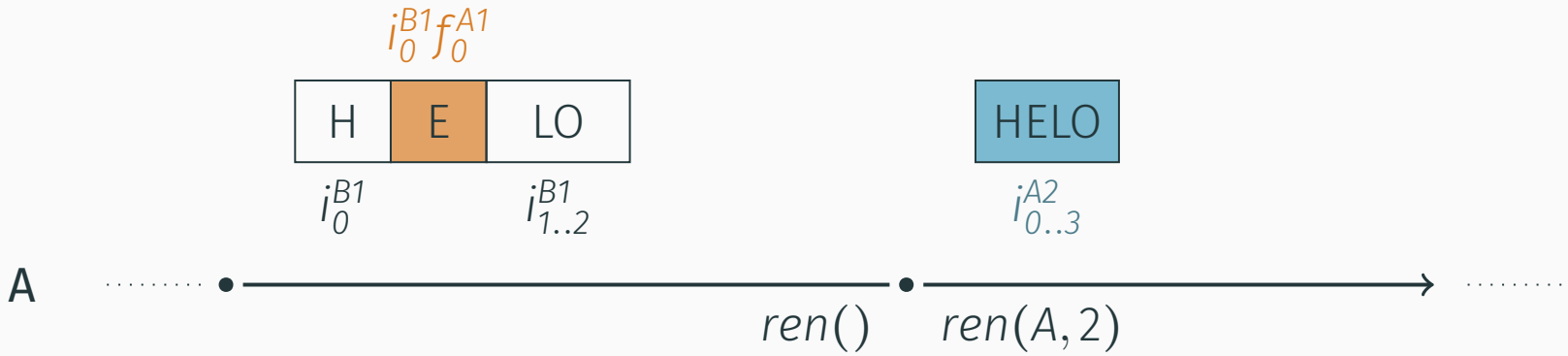
- Génère nouvel identifiant pour le 1er élément : $i_0^{B1} \rightarrow i_0^{A2}$
- Génère identifiants contigus pour éléments suivants : i_1^{A2}, i_2^{A2}...

Opération *rename*



- Génère nouvel identifiant pour le 1er élément : $i_0^{B1} \rightarrow i_0^{A2}$
- Génère identifiants contigus pour éléments suivants : i_1^{A2}, i_2^{A2}...

Opération *rename*



- Génère nouvel identifiant pour le 1er élément : $i_0^{B1} \rightarrow i_0^{A2}$
- Génère identifiants contigus pour éléments suivants : i_1^{A2}, i_2^{A2}...

Thanks for your attention, any questions?