

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

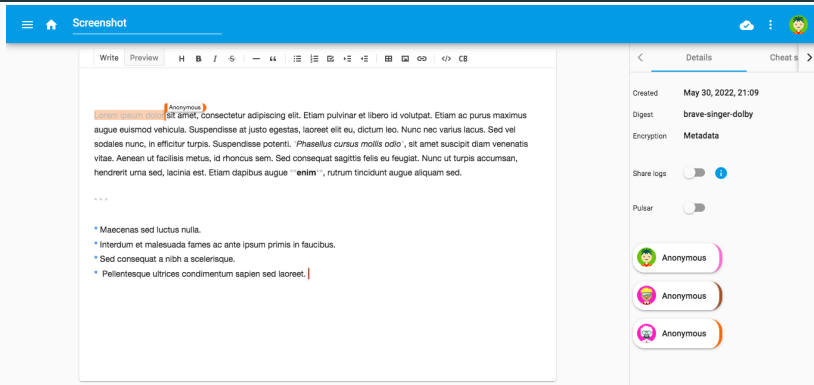
---

Matthieu Nicolas ([matthieu.nicolas@loria.fr](mailto:matthieu.nicolas@loria.fr))

20 décembre 2022

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

# MUTE<sup>\*</sup>, un exemple de Local-First Software (LFS)<sup>[1]</sup>

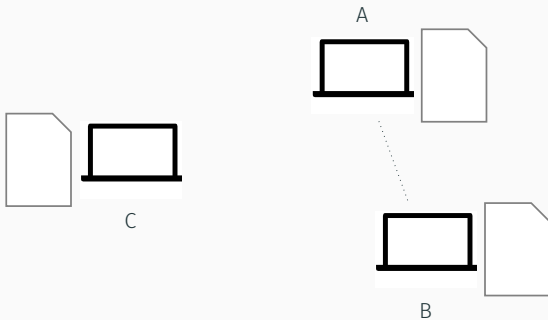


- Application pair-à-pair
- Permet de rédiger collaborativement des documents texte
- Garantit la confidentialité & souveraineté des données

\*. Disponible à : <https://mutehost.loria.fr>

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

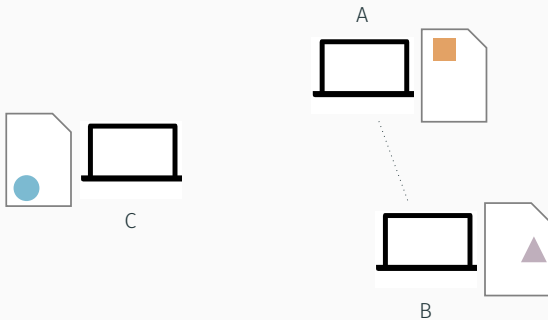
# Réplication dans applications collaboratives pair-à-pair



---

[2]. TERRY et al., « Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System ».

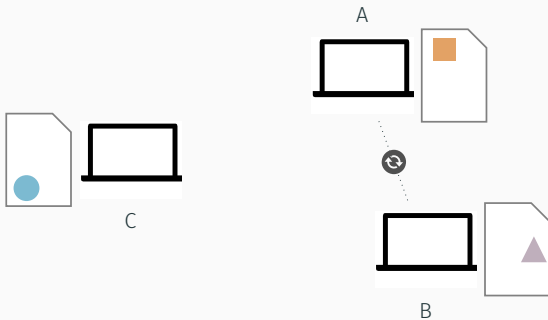
# Réplication dans applications collaboratives pair-à-pair



---

[2]. TERRY et al., « Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System ».

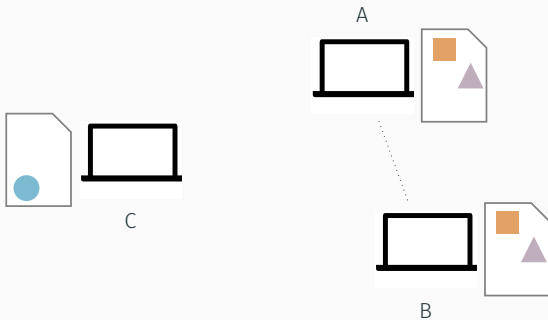
# Réplication dans applications collaboratives pair-à-pair



---

[2]. TERRY et al., « Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System ».

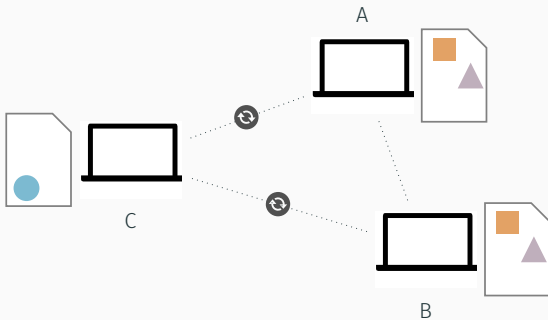
# Réplication dans applications collaboratives pair-à-pair



---

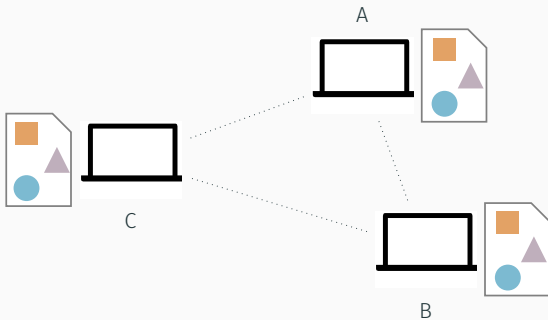
[2]. TERRY et al., « Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System ».

# Réplication dans applications collaboratives pair-à-pair



[2]. TERRY et al., « Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System ».

# Réplication dans applications collaboratives pair-à-pair

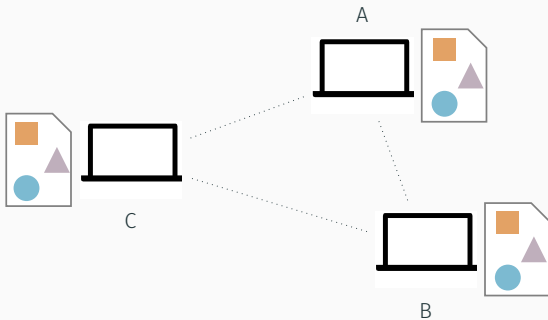


---

[2]. TERRY et al., « Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System ».



# Réplication dans applications collaboratives pair-à-pair

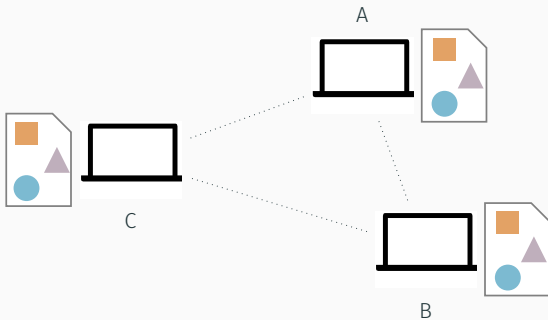


- Doit garantir **convergence à terme** <sup>[2]</sup>...
- ...malgré ordres différents d'intégration des modifications

---

[2]. TERRY et al., « Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System ».

# Réplication dans applications collaboratives pair-à-pair



- Doit garantir **convergence à terme** <sup>[2]</sup>...
- ...malgré ordres différents d'intégration des modifications

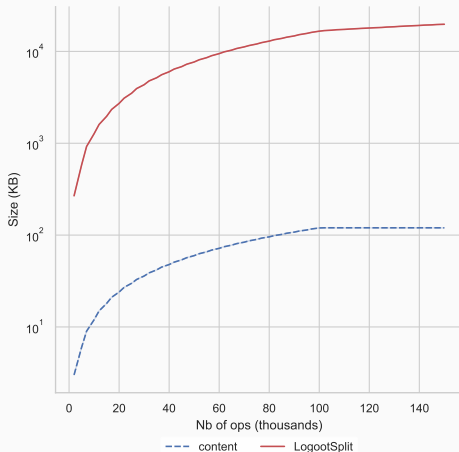
**Nécessite des mécanismes de résolution de conflits**

---

[2]. TERRY et al., « Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System ».

# Évaluation de MUTE

## Taille du texte comparée à taille de la séquence répliquée

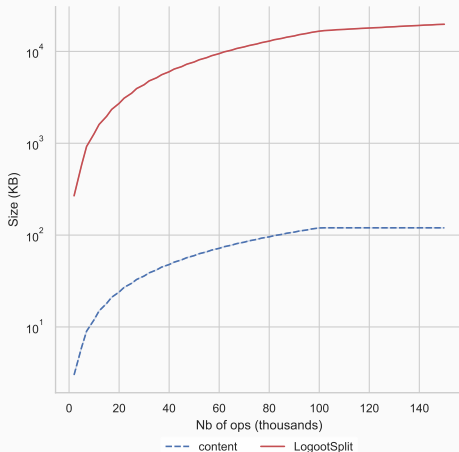


### Constat

- 1% contenu...
- ...99% métadonnées

# Évaluation de MUTE

## Taille du texte comparée à taille de la séquence répliquée



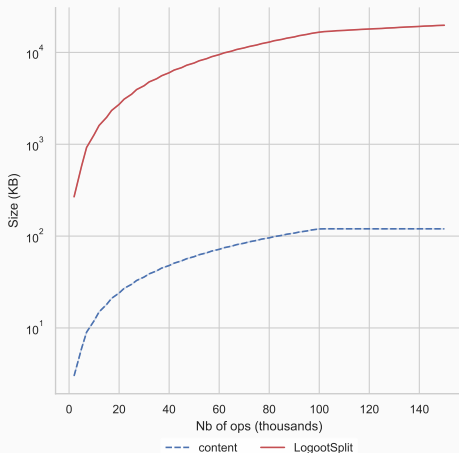
### Constat

- 1% contenu...
- ...99% métadonnées

Et ça augmente!

# Évaluation de MUTE

## Taille du texte comparée à taille de la séquence répliquée



### Constat

- 1% contenu...
- ...99% métadonnées

Et ça augmente!

### Impact

- Surcoût **mémoire**...
- ...mais aussi surcoût en **calculs** et en **bande-passante**

Comment peut-on réduire le surcoût des  
mécanismes de résolution de conflits dans les  
applications pair-à-pair ?

# Conflict-free Replicated Data Types (CRDTs)<sup>[3]</sup>

- Nouvelles spécifications des types de données, e.g. *Ensemble* ou *Séquence*
- Incorpore nativement mécanisme de résolution de conflits

---

[3]. SHAPIRO et al., « Conflict-Free Replicated Data Types ».

# Conflict-free Replicated Data Types (CRDTs)<sup>[3]</sup>

- Nouvelles spécifications des types de données, e.g. *Ensemble* ou *Séquence*
- Incorpore nativement mécanisme de résolution de conflits

## Propriétés des CRDTs

- Permettent modifications **sans coordination**
- Garantissent la **convergence forte**

---

[3]. SHAPIRO et al., « Conflict-Free Replicated Data Types ».



# Conflict-free Replicated Data Types (CRDTs)<sup>[3]</sup>

- Nouvelles spécifications des types de données, e.g. *Ensemble* ou *Séquence*
- Incorpore nativement mécanisme de résolution de conflits

## Propriétés des CRDTs

- Permettent modifications **sans coordination**
- Garantissent la **convergence forte**

## Convergence forte

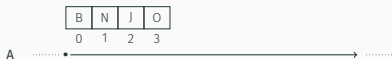
Ensemble des noeuds ayant intégrés le même ensemble de modifications obtient des états équivalents, sans nécessiter d'actions ou messages supplémentaires

---

[3]. SHAPIRO et al., « Conflict-Free Replicated Data Types ».

# CRDTs pour le type Séquence

## Type Séquence séquentiel



---

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

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

## Type Séquence séquentiel



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

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

# CRDTs pour le type Séquence

## Type Séquence séquentiel



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

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

# CRDTs pour le type Séquence

## Type Séquence séquentiel



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

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

# CRDTs pour le type Séquence

## Type Séquence séquentiel



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

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

# CRDTs pour le type Séquence

## Type Séquence séquentiel



- Changements des indices est **source de conflits**

---

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

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

# CRDTs pour le type Séquence

## Type Séquence séquentiel



## CRDTs pour Séquence



- Changements des indices est **source de conflits**
- Assignent des **identifiants de position**<sup>[4]</sup> à chaque élément
- Permettent d'**ordonner les éléments**

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

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



# CRDTs pour le type Séquence

## Type Séquence séquentiel



## CRDTs pour Séquence



- Changements des indices est **source de conflits**
- Assignent des **identifiants de position**<sup>[4]</sup> à chaque élément
- Permettent d'**ordonner les éléments**

$$id_0 <_{id} id_1 <_{id} id_2 <_{id} id_3$$

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

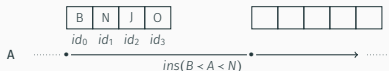
[5]. ANDRÉ et al., « Supporting Adaptable Granularity of Changes for Massive-Scale Collaborative Editing ».

# CRDTs pour le type Séquence

## Type Séquence séquentiel



## CRDTs pour Séquence



- Changements des indices est **source de conflits**
- Assignent des **identifiants de position**<sup>[4]</sup> à chaque élément
- Permettent d'**ordonner les éléments**

$$id_0 <_{id} id_1 <_{id} id_2 <_{id} id_3$$

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

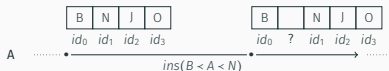
[5]. ANDRÉ et al., « Supporting Adaptable Granularity of Changes for Massive-Scale Collaborative Editing ».

# CRDTs pour le type Séquence

## Type Séquence séquentiel



## CRDTs pour Séquence



- Changements des indices est **source de conflits**
- Assignent des **identifiants de position**<sup>[4]</sup> à chaque élément
- Permettent d'**ordonner les éléments**

$$id_0 <_{id} id_1 <_{id} id_2 <_{id} id_3$$

- Nécessaire que les identifiants appartiennent à un **espace dense**

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

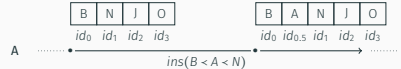
[5]. ANDRÉ et al., « Supporting Adaptable Granularity of Changes for Massive-Scale Collaborative Editing ».

# CRDTs pour le type Séquence

## Type Séquence séquentiel



## CRDTs pour Séquence



- Changements des indices est **source de conflits**
- Assignent des **identifiants de position**<sup>[4]</sup> à chaque élément
- Permettent d'**ordonner les éléments**

$$id_0 <_{id} id_1 <_{id} id_2 <_{id} id_3$$

- Nécessaire que les identifiants appartiennent à un **espace dense**

$$id_0 <_{id} id_{0.5} <_{id} id_1$$

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

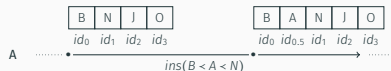
[5]. ANDRÉ et al., « Supporting Adaptable Granularity of Changes for Massive-Scale Collaborative Editing ».

# CRDTs pour le type Séquence

## Type Séquence séquentiel



## CRDTs pour Séquence



- Changements des indices est **source de conflits**
- Assignent des **identifiants de position**<sup>[4]</sup> à chaque élément
- Permettent d'**ordonner les éléments**

$$id_0 <_{id} id_1 <_{id} id_2 <_{id} id_3$$

- Nécessaire que les identifiants appartiennent à un **espace dense**

$$id_0 <_{id} id_{0.5} <_{id} id_1$$

Utilise LogootSplit<sup>[5]</sup> comme base

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

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

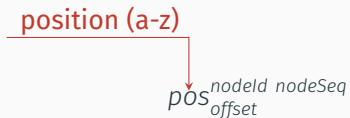
## Identifiant

- Composé d'un ou plusieurs tuples de la forme

$$pos_{offset}^{nodeId \ nodeSeq}$$

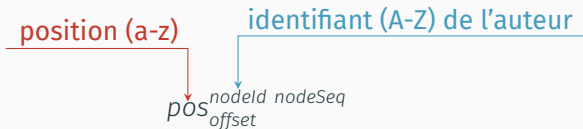
## Identifiant

- Composé d'un ou plusieurs tuples de la forme



## Identifiant

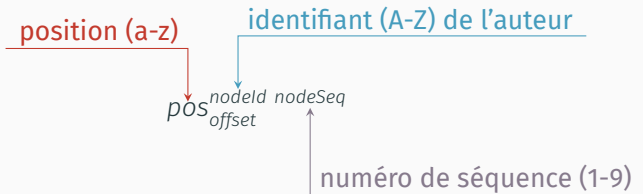
- Composé d'un ou plusieurs tuples de la forme





## Identifiant

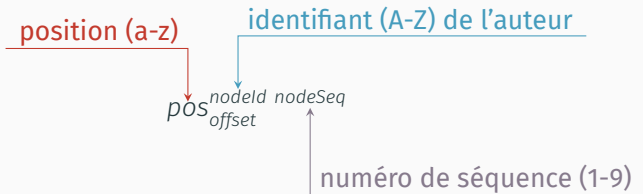
- Composé d'un ou plusieurs tuples de la forme



# Identifiant LogootSplit

## Identifiant

- Composé d'un ou plusieurs tuples de la forme



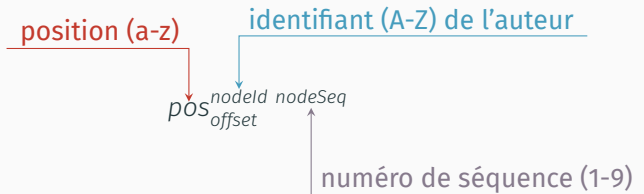
## Exemples

$d_0^{F5}$

# Identifiant LogootSplit

## Identifiant

- Composé d'un ou plusieurs tuples de la forme



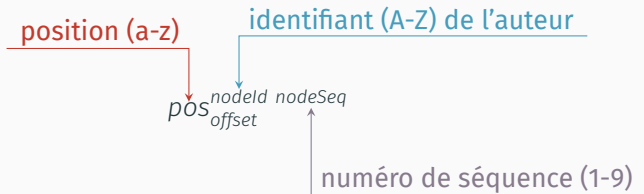
## Exemples

$$d_0^{F5} <_{id} m_0^{C1}$$

# Identifiant LogootSplit

## Identifiant

- Composé d'un ou plusieurs tuples de la forme



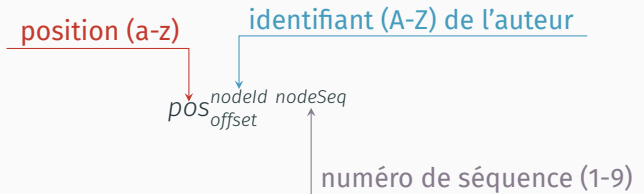
## Exemples

$$d_0^{F5} <_{id} m_0^{C1} <_{id} m_0^{C1} f_0^{E1}$$

# Identifiant LogootSplit

## Identifiant

- Composé d'un ou plusieurs tuples de la forme



## Exemples

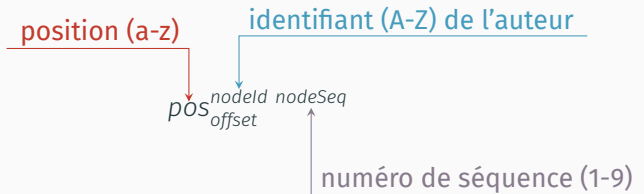
$$d_0^{F5} <_{id} m_0^{C1} <_{id} m_0^{C1} f_0^{E1}$$

$$i_0^{B1} <_{id} ? <_{id} i_1^{B1}$$

# Identifiant LogootSplit

## Identifiant

- Composé d'un ou plusieurs tuples de la forme



## Exemples

$$d_0^{F5} <_{id} m_0^{C1} <_{id} m_0^{C1} f_0^{E1}$$

$$i_0^{B1} <_{id} i_0^{B1} f_0^{A1} <_{id} i_1^{B1}$$

- Coûteux de stocker les identifiants de chaque élément

B	A	N	J	O
$m_0^{C1}$	$m_1^{C1}$	$m_2^{C1}$	$m_3^{C1}$	$m_4^{C1}$

# Bloc LogootSplit

- Coûteux de stocker les identifiants de chaque élément

B	A	N	J	O
$m_0^{C1}$	$m_1^{C1}$	$m_2^{C1}$	$m_3^{C1}$	$m_4^{C1}$

- Aggrège en un **bloc** éléments ayant **identifiants contigus**

## Identifiants contigus

Deux identifiants sont contigus si et seulement si les deux identifiants sont identiques à l'exception de leur dernier offset et que leur derniers offsets sont consécutifs.



# Bloc LogootSplit

- Coûteux de stocker les identifiants de chaque élément

B	A	N	J	O
$m_0^{C1}$	$m_1^{C1}$	$m_2^{C1}$	$m_3^{C1}$	$m_4^{C1}$

- Aggrège en un **bloc** éléments ayant **identifiants contigus**

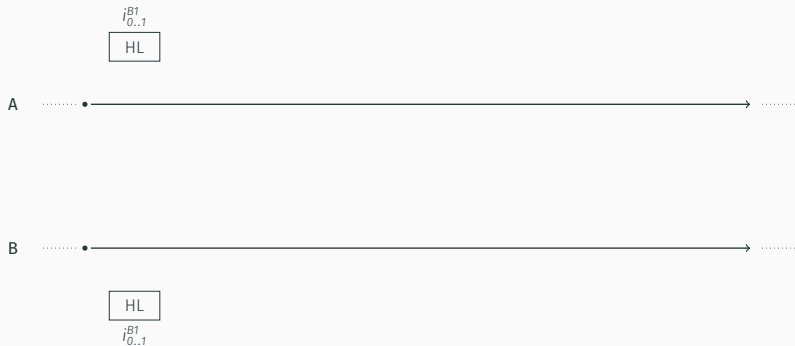
## Identifiants contigus

Deux identifiants sont contigus si et seulement si les deux identifiants sont identiques à l'exception de leur dernier offset et que leur derniers offsets sont consécutifs.

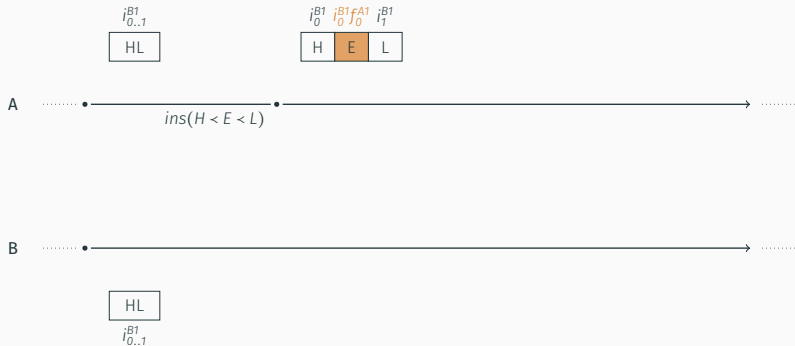
- Note l'intervalle d'identifiants d'un bloc :  $pos_{begin..end}^{nodeId nodeSeq}$

BANJO
$m_{0..4}^{C1}$

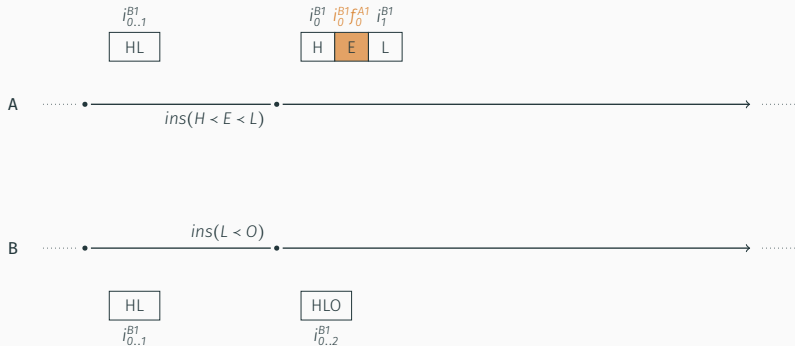
# Exemple insertions concurrentes



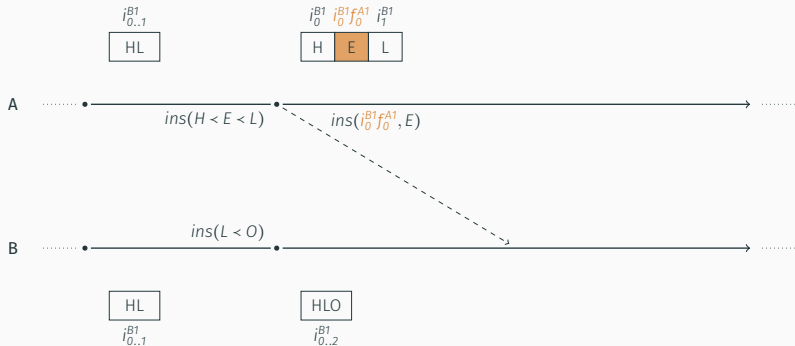
# Exemple insertions concurrentes



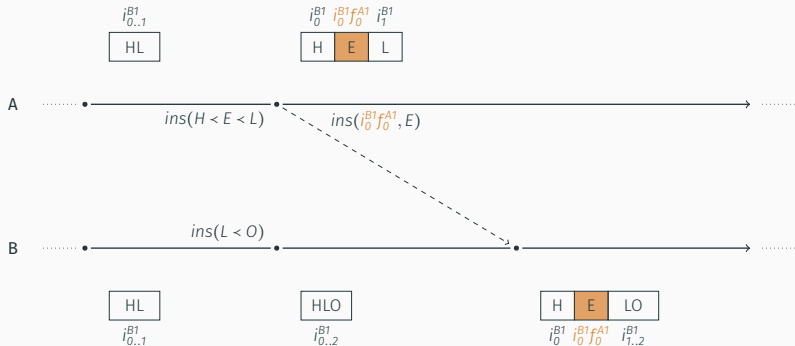
# Exemple insertions concurrentes



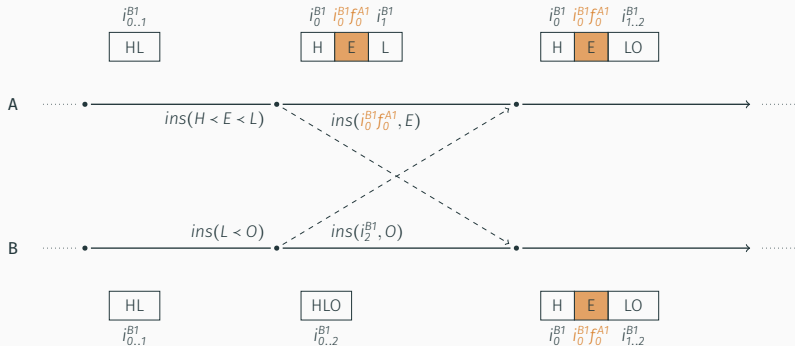
# Exemple insertions concurrentes



# Exemple insertions concurrentes



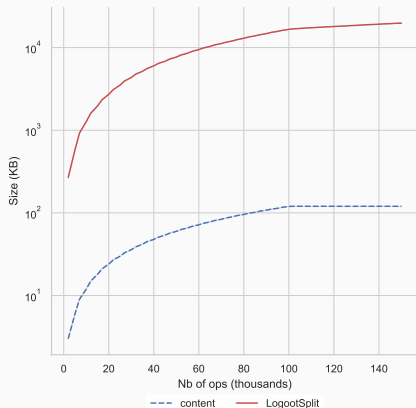
# Exemple insertions concurrentes



# Limites de LogootSplit

## Sources de la croissance des métadonnées

- Augmentation non-bornée de la taille des identifiants
- Fragmentation de la séquence en un nombre croissant de blocs



Diminution des performances du point de vue **mémoire, calculs et bande-passante**

**Figure 1** – Taille du contenu comparée à la taille de la séquence LogootSplit



## L'approche core-nebula<sup>[6]</sup>, pour Treedoc

- Ré-assigne des identifiants courts aux éléments, c.-à-d. les *renomme*
- Transforme les opérations *insert* et *remove* concurrentes...

---

[6]. ZAWIRSKI et al., « Asynchronous rebalancing of a replicated tree ».

## L'approche core-nebula<sup>[6]</sup>, pour Treedoc

- Ré-assigne des identifiants courts aux éléments, c.-à-d. les *renomme*
- Transforme les opérations *insert* et *remove* concurrentes...
- ...mais ne supportent pas opérations *rename* concurrentes

---

[6]. ZAWIRSKI et al., « Asynchronous rebalancing of a replicated tree ».

## L'approche core-nebula<sup>[6]</sup>, pour Treedoc

- Ré-assigne des identifiants courts aux éléments, c.-à-d. les *renomme*
- Transforme les opérations *insert* et *remove* concurrentes...
- ...mais ne supportent pas opérations *rename* concurrentes

Inadaptée aux applications pair-à-pair

---

[6]. ZAWIRSKI et al., « Asynchronous rebalancing of a replicated tree ».

## Proposition

Mécanisme de renommage supportant les  
renommages concurrents