Sauvegarde et dissémination dans MoSAIC

Ludovic Courtès

LAAS-CNRS

- **Sauvegarde des données : résumé**
  - Soumission à MobiHoc : plan de l'article
  - Contraintes sur les mécanismes de stockage
  - Solutions proposées issues de l'état de l'art
  - Solutions proposées : fragmentation & indexation
  - Évaluation des solutions : cas de test
  - Évaluation des solutions : conclusion
- Codes d'effacement
- Nouvelles pistes de recherche

Soumission à MobiHoc : plan de l'article

- Implications des contraintes MoSAIC sur les **mécanismes de stockage**
- État de l'art des **mécanismes de stockage sous-jacent au stockage réparti**
- **Architecture** du prototype
- **Évaluation expérimentale** de ces techniques
- Pistes de recherche à venir

Contraintes sur les mécanismes de stockage

- **Contraintes imposées par MoSAIC**
  - absence de relations de **confiance** entre participants
  - espace de stockage et surtout bande passante **limités**
  - interactions **éphémères et imprévisibles**
- **Implications**
  - **« imputabilité »** de la consommation de ressources
  - **chiffrement**, **détection d'erreurs** dans les données (accidentelles *et* malicieuses)
  - **redondance**
  - **compression** des données
  - limitation de la **taille des fragments** de données
  - **atomicité & cohérence** du « support » de stockage

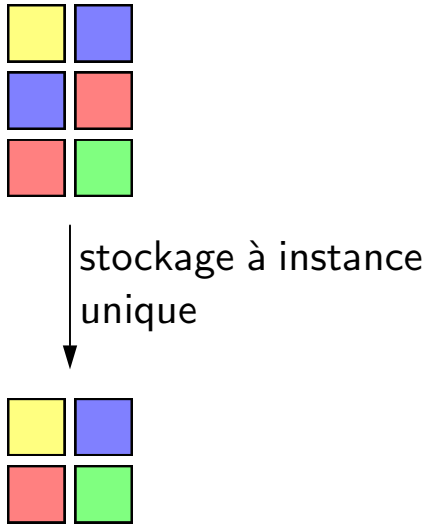
Solutions proposées issues de l'état de l'art

Détection d'erreurs

- fonctions de hachage cryptographiques (SHA*, Whirlpool, etc.)

Compression

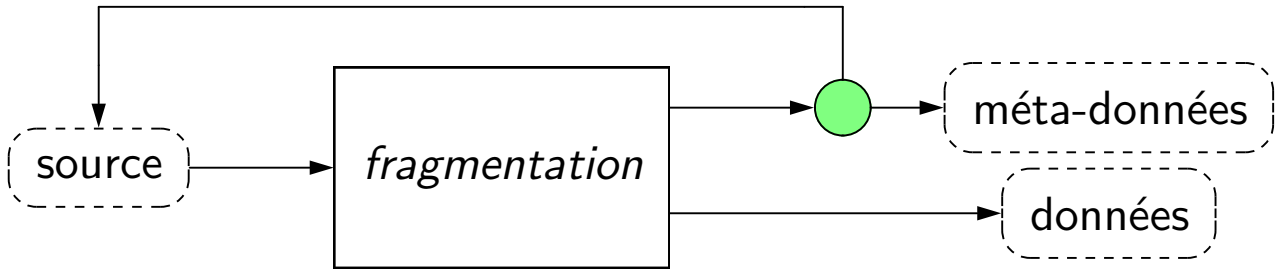
- stockage à instance unique
- découpage fonction du contenu
- compression sans perte « classique »
- voire compression à perte spécifique (images, sons, etc.)



Solutions proposées : fragmentation & indexation

Fragmentation

- fragmentation
- indexation de blocs individuels
- indexation de séquences de blocs $\Rightarrow$ **méta-données**



Évaluation des solutions : cas de test

1. **Fichiers en « écriture seule »**
   - ensemble de fichiers PostScript
   - peu voire pas de ressemblance entre eux
   - format « verbeux », non compact

2. **Versions successives de fichiers textes**
   - ensemble de fichiers sources C
   - beaucoup de ressemblances
   - format non compact

3. **Un fichier texte**
   - boîte aux lettres (format `mbox`)
   - format non compact

Évaluation des solutions : conclusion

Critères d'évaluation

- **taux de compression**
- **débit lors du traitement** ≈ consommation CPU

Résultats

- **compression « classique »** (type `gzip`) indispensable
- **stockage à instance unique** bénéfique et « gratuit »
- **découpage fonction du contenu (algo. de Manber)** limité

- Sauvegarde des données : résumé
- **Codes d'effacement**
  - Codes d'effacement : définitions
  - Codes d'effacement : exemples
  - Impact sur la disponibilité et le coût de stockage
  - Codes d'effacement et disponibilité
- Nouvelles pistes de recherche

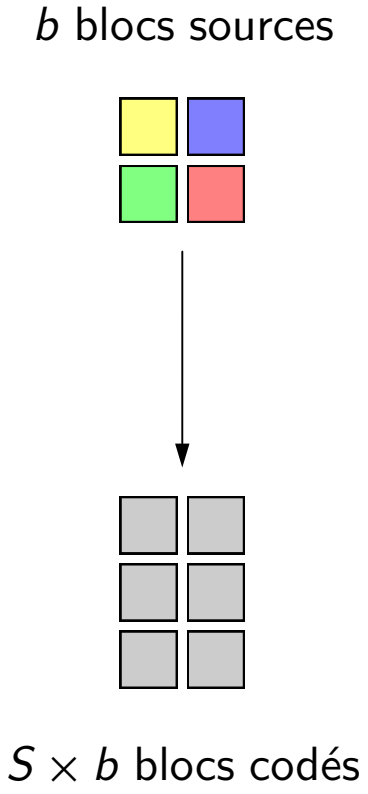
Codes d'effacement : définitions

Définition

- message de b blocs $\rightarrow b \times S$ blocs
- m blocs suffisent pour recouvrir le message, $b < m < S \times b$
- $S \in \Re$: « **amplification** » (ou *stretch factor*, *overhead*)
- **défaillances tolérées** : $S \times b - m$

Codes optimaux (*Maximum Distance Separable*)

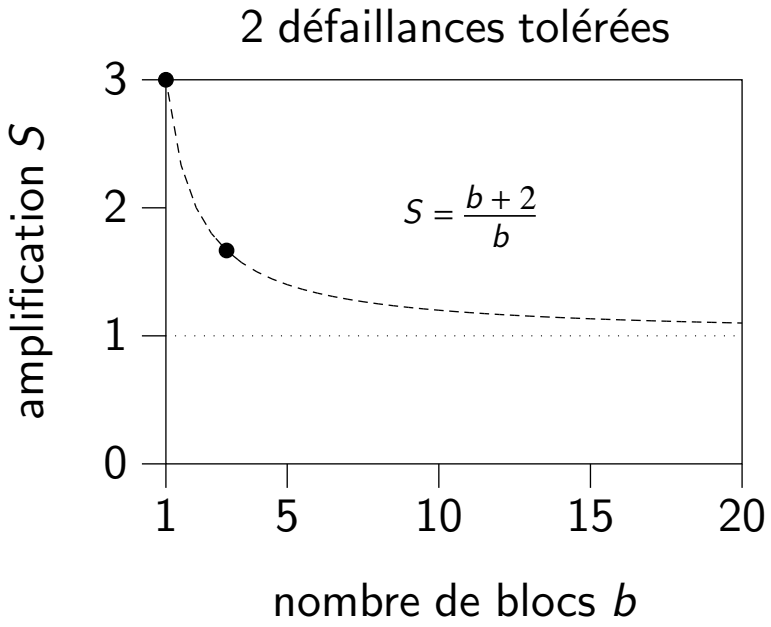
- quand $m = b \Longrightarrow$ défaillances tolérées : $S \times b - b$
- notation : code $(n,k) \Leftrightarrow S = \frac{n}{k}$ et $n - k$ défaillances tolérées



Codes d'effacement : exemples

Exemples

- code (3,1) : 2 défaillances tolérées, $S = 3$
- code (5,3) : 2 défaillances tolérées, $S = 1.67$



Impact sur la disponibilité et le coût de stockage

Disponibilité d'une donnée *D*

- **Hypothèse** : $S \times b$ fragments, chacun stocké chez un contributeur différent
- μ : probabilité qu'un contributeur soit disponible en un instant quelconque
- $A(D) = \sum_{i=b}^{S \times b} C_i^{S \times b} \times \mu^i \times (1-\mu)^{S \times b - i}$

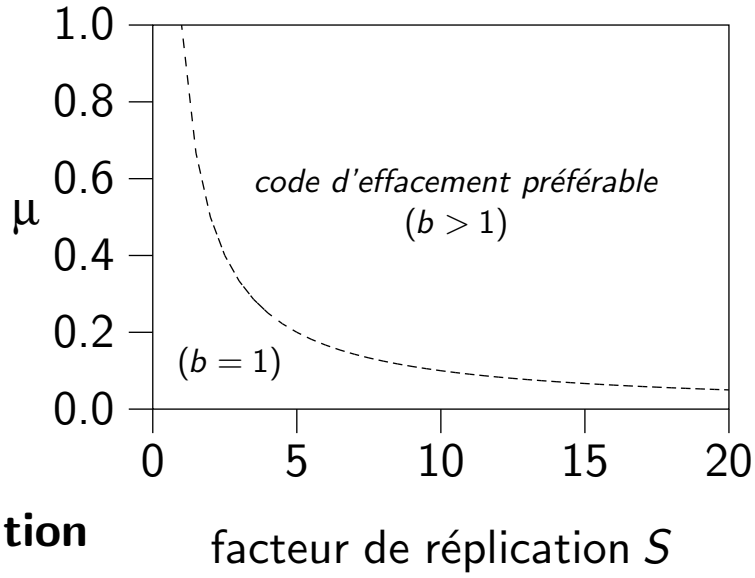
Problèmes

- impact de μ sur la disponibilité des données ?
- impact du nombre de blocs b ?
- impact du facteur de réplication S ?

Codes d'effacement et disponibilité

Conclusions [Lin et al. 2004]

- Si $S \times \mu > 1$, alors choisir b « grand »
- Si $S \times \mu \leq 1$, alors choisir $b = 1$



Problématiques

- **évaluation dynamique** de μ
- impact de la **fragmentation et de la dissémination des fragments** sur la disponibilité des données

- Sauvegarde des données : résumé
- Codes d'effacement
- **Nouvelles pistes de recherche**
  - Évaluation des protocoles choisis
  - Évaluation : une fois les données distribuées
  - Évaluation : l'algorithme de dissémination
  - Résultats escomptés

Évaluation des protocoles choisis

Impact de l'algorithme de dissémination

- **politiques** possibles ?
- impact sur la **disponibilité** ?
- impact sur le **coût en ressources** ? (avec stockage ≈ énergie)

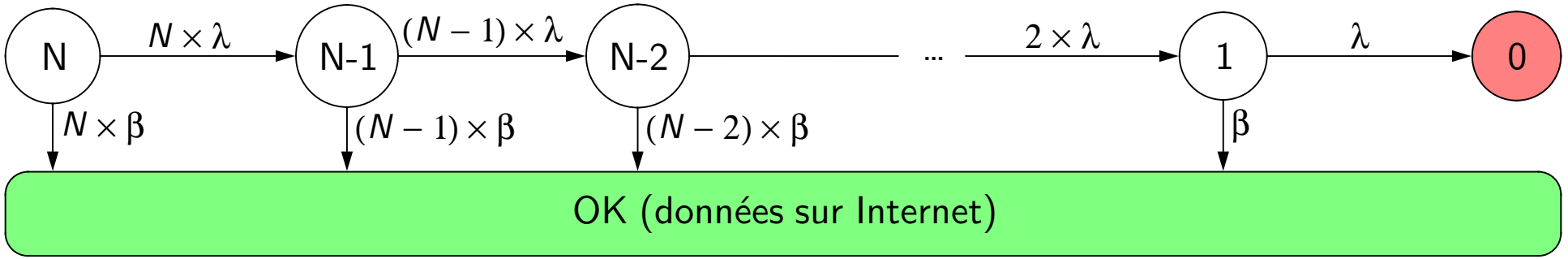
Méthodologie

- évaluation **analytique**
- à partir d'un **automate** représentant les scénarios de dissémination

Évaluation : une fois les données distribuées

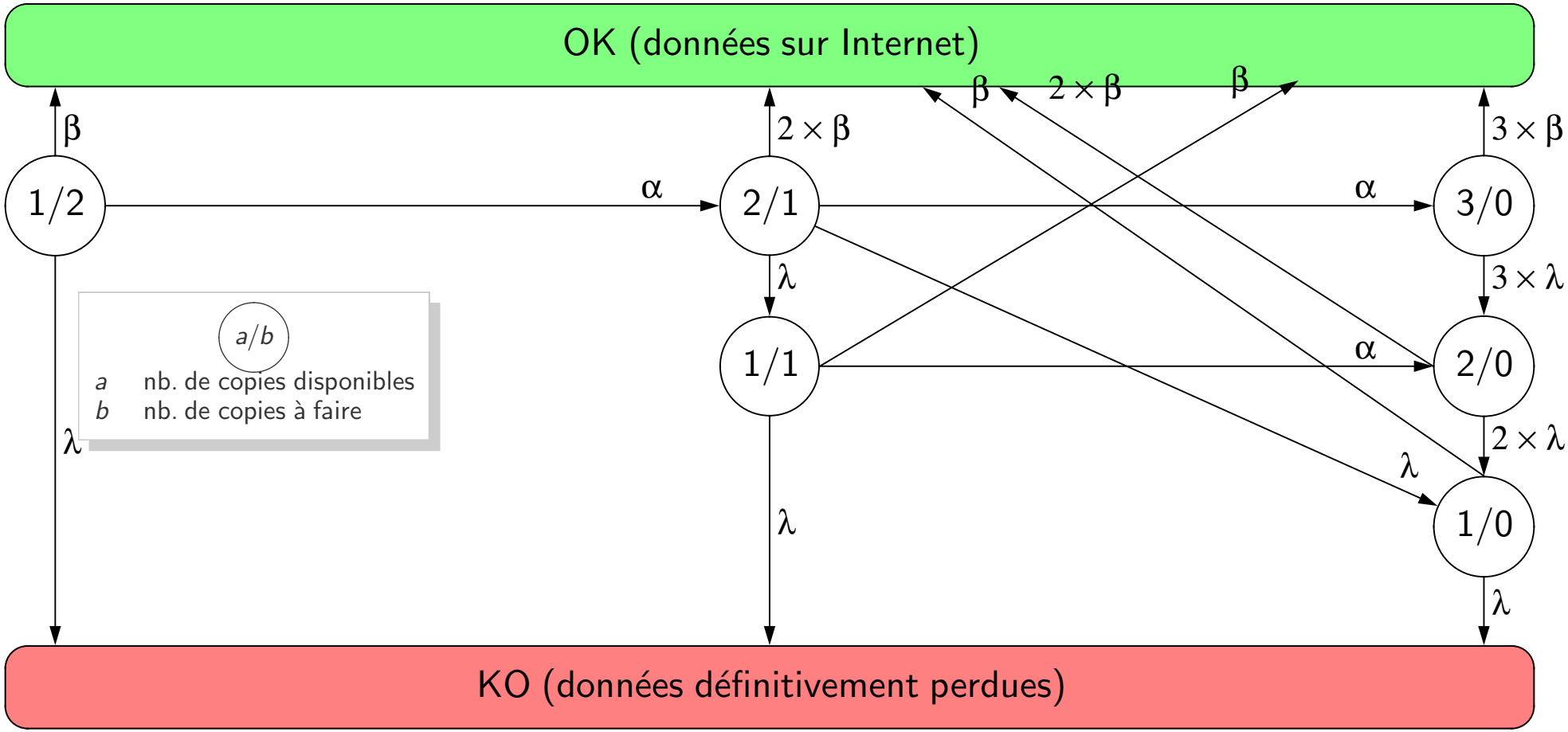
Hypothèses

- réplication simple, N copies distribuées
- λ : taux de défaillance d'un contributeur
- β : taux de connexion à Internet d'un contributeur



Évaluation : l'algorithme de dissémination

Et avant les *N* copies ?



Résultats escomptés

Paramètres

- temps moyen entre deux rencontres
- temps moyen entre deux connexions Internet
- temps moyen entre deux défaillances de contributeurs

Résultats

- probabilité de réussite/échec
- temps moyen pour arriver à l'état OK
- et si on utilise des codes d'effacement ?

Fin

Questions ?