CONTRIBUTIONS À LA RÉPLICATION DE DONNÉES DANS LES SYSTÈMES DISTRIBUÉS À GRANDE ÉCHELLE

Sébastien Monnet

Séminaire LISTIC
Le 12 Juillet 2016

Contexte : gestion de données dans les systèmes distribués à grande échelle

- **Motivations**
  - Masse de données
  - Besoin de systèmes de gestion de données fiables, performants, cohérents
- **Architectures visées**
  - Systèmes pair-à-pair
  - Centres de données/*clouds*
- **Problématiques**
  - Grande échelle (grand nombre de machines/nœuds)
  - Grande dynamicité
  - Virtualisation (fragmentation des ressources)

Réplication de données dans les systèmes distribués à grande échelle

- **Réplication : technique clé**

- **Trois axes de recherche**
  - Tolérance aux fautes (stockage fiable)
  - Performances des accès aux données
  - Cohérence des données

Axe 1 : Tolérance aux fautes

- **Plusieurs copies des données**
  - En cas de panne il reste des copies disponibles
- **Problèmes**
  - Quel degré de réplication (combien de copies) ?
  - Comment placer les copies des données ?
  - Comment maintenir le degré de réplication ?
    - Quand déclencher les réparations ?

Tolérance aux fautes : contributions

- **Impact du placement sur la pérennité**
  - RelaxDHT [SSS 09, TAAS 12] (Sergey Legtchenko)
  - SPLAD [NCA 15] (Véronique Simon)
- **Détection de fautes**
  - RepFD [ICPP 15] (Maxime Véron)
    - Détecteur de fautes collaboratif
    - Adapté aux environnements dynamiques

Axe 2 : Performance des accès

- **Placement des copies**
  - Localité des accès
  - Affinités des données (proximité sémantique)
- **Adaptation du nombre de copies à la popularité**
  - Meilleure répartition de la charge sur les serveurs
- **Problèmes**
  - Comment adapter dynamiquement les ressources en fonction de la charge ?
  - Comment localiser efficacement les données ?
  - Comment faire face à la fragmentation de ressources due à la virtualisation ?

Performance des accès : contributions

- **Adaptation de la réplication en fonction de l'utilisation**
  - AREN [ICPADS 12, ICPADS 13] (Gutemberg Da Silva Silvestre)
    - Utiliser le potentiel des périphériques de bordure de réseaux (box, mini centres de données)
  - POPS [ICPADS 14] (Karine Pires)
    - Prise en compte de la popularité des flux vidéos pour l'allocation de ressources
- **Mutualisation de la mémoire des machines virtuelles**
  - PUMA [SYSTOR 15] (Maxime Lorrillere)
    - Permet à une machine virtuelle d'étendre son cache
- **Localisation efficace pour des distributions non-uniformes**
  - DONUT [DSN 10, SRDS 11] (Sergey Legtchenko)

Axe 3 : Réplication et cohérence

- **Mises à jour concurrentes**
  - Des copies d'une même donnée peuvent diverger
- **Problèmes**
  - Comment assurer la cohérence des données répliquées
  - Quel modèle/protocole de cohérence

Réplication et cohérence : contributions

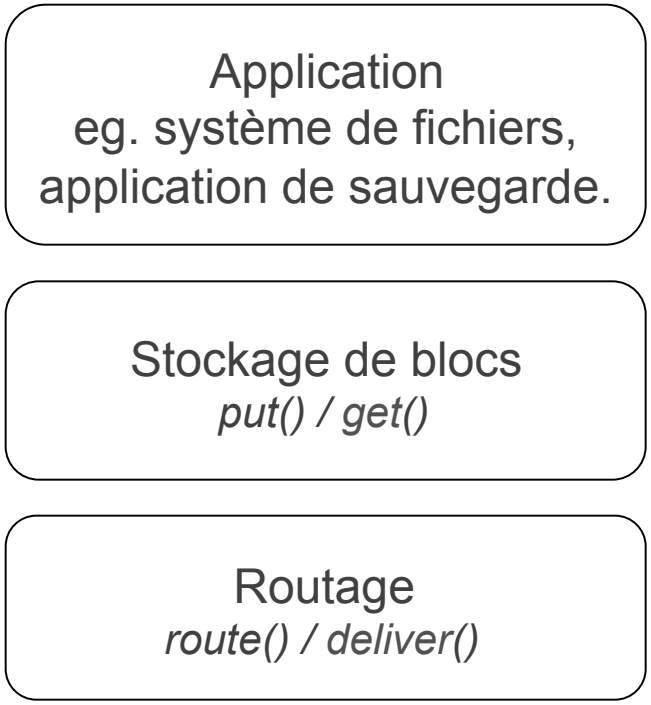
- **Bases de données répliquées**
  - Gargamel [ICPADS 12] (Pierpaolo Cincilla)
    - Gestion des conflits en amont (procédures stockées)
      - Exécution parallèle des transactions indépendantes
      - Exécution séquentielle des transactions en conflit potentiel
    - Repose sur
      - Un classificateur de transactions
      - Un ordonnanceur (maintien du graphe des conflits)

IMPACT DU PLACEMENT DES DONNÉES SUR LEUR PÉRENNITÉ

RelaxDHT et SPLAD

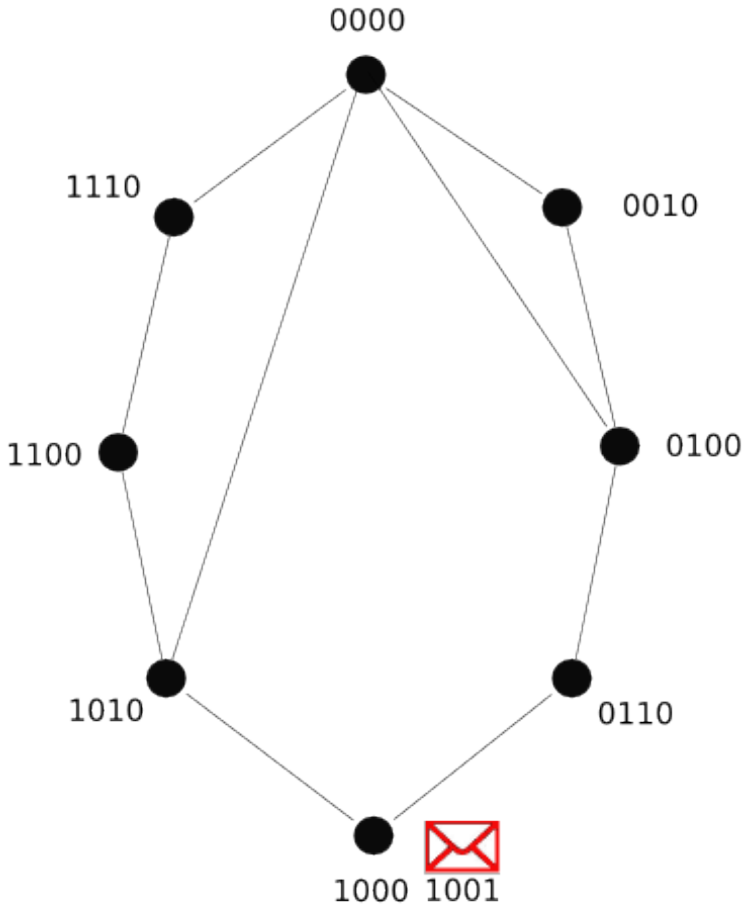
Contexte : les tables de hachage distribuées (DHT)

- **Interface simple**
  - *put/get*
- **Localisation/routage efficace**
- **Maintien du degré de réplication**
- **Nombreuses variantes**
  - PAST, DHASH, Can, Tapestry, …
- **Mauvaise tolérance au *churn***



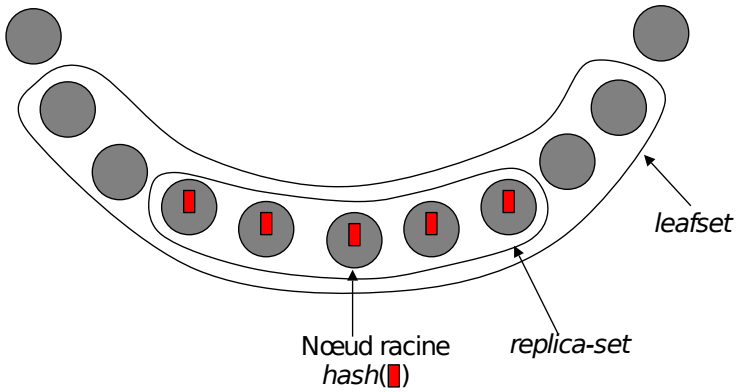
Fonctionnement des DHTs (PAST, DHASH)

- **Un identifiant par nœud**
  - Un anneau logique
  - Un routage à base de clé (KBR)
- **Un identifiant par donnée**
- **Nœud racine d'une donnée (*root*)**
  - Nœud ayant l'identifiant le plus proche
- **Voisinage logique de la racine : le *leafset***

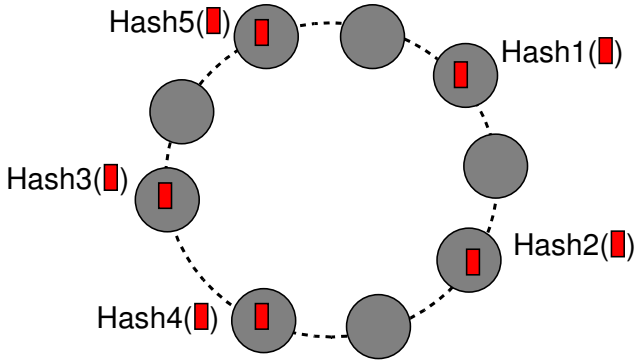


Placements des copies dans les DHTs

- **Deux principales familles**
  - Contigu (nœuds adjacents à la racine)
    - Au sein du *leafset (leafset-based)*
  - Non-contigu
    - À base de clés multiples, …



- ***Leafset-based* =>**
  maintenance indépendante du nombre de nœuds/blocs



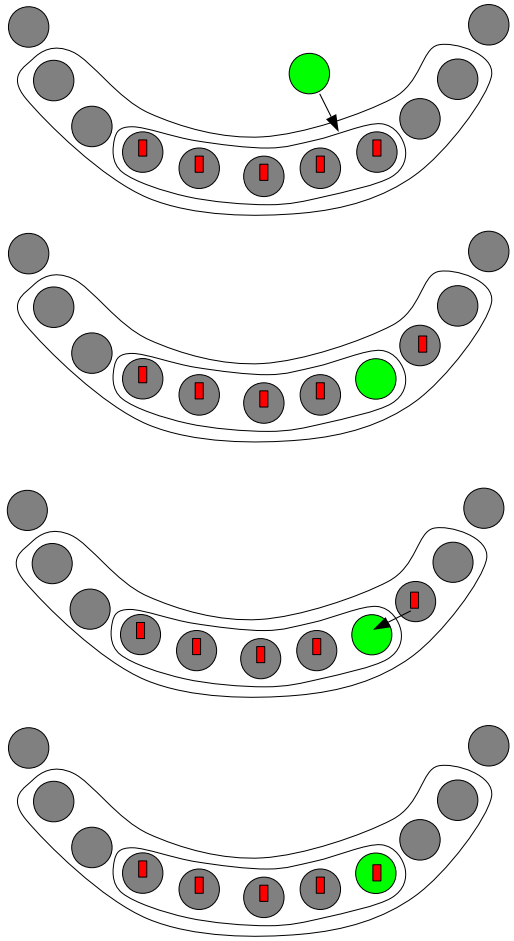
Les DHTs face au *churn*
Mouvements « inutiles »

- **Insertion d'un nœud**
  - Racine de nombreuses données
  - Au sein de plusieurs replica-sets

  => brise la contiguïté

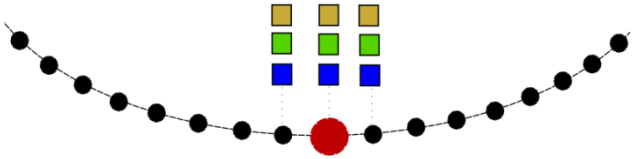
- **Respect des contraintes de placement**

  => Besoin de déplacer des données

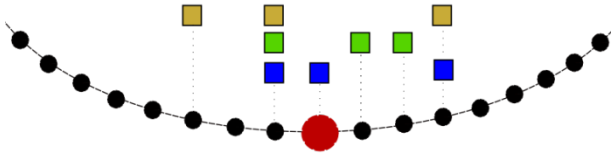


Les DHTs face au *churn*
Contenus similaires

- **Placement contigu**
  - Contenus similaires sur les nœuds
  - Peu de sources disponibles pour les réparations

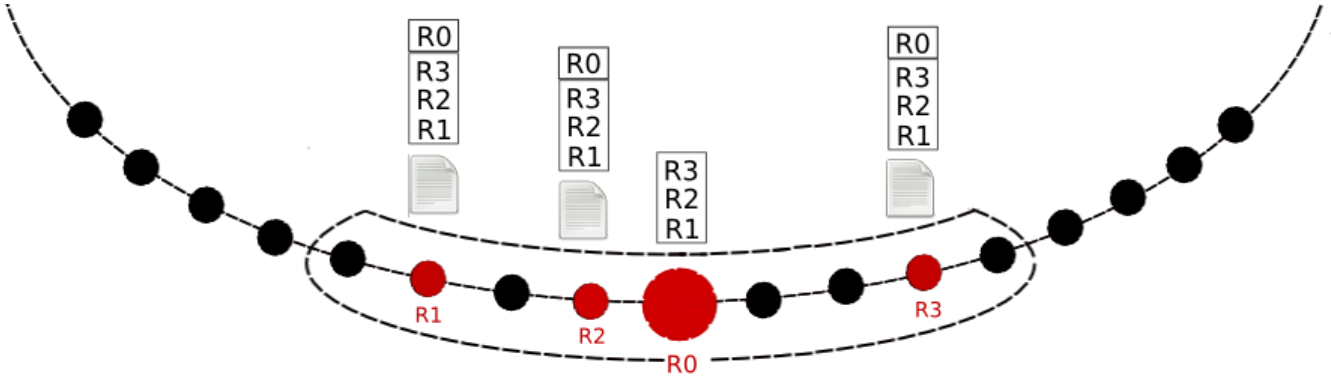


- **Données éparpillées**
  - Plus de sources et destinations pour les transferts
  - Réparations plus rapides
  - Moins de pertes de données



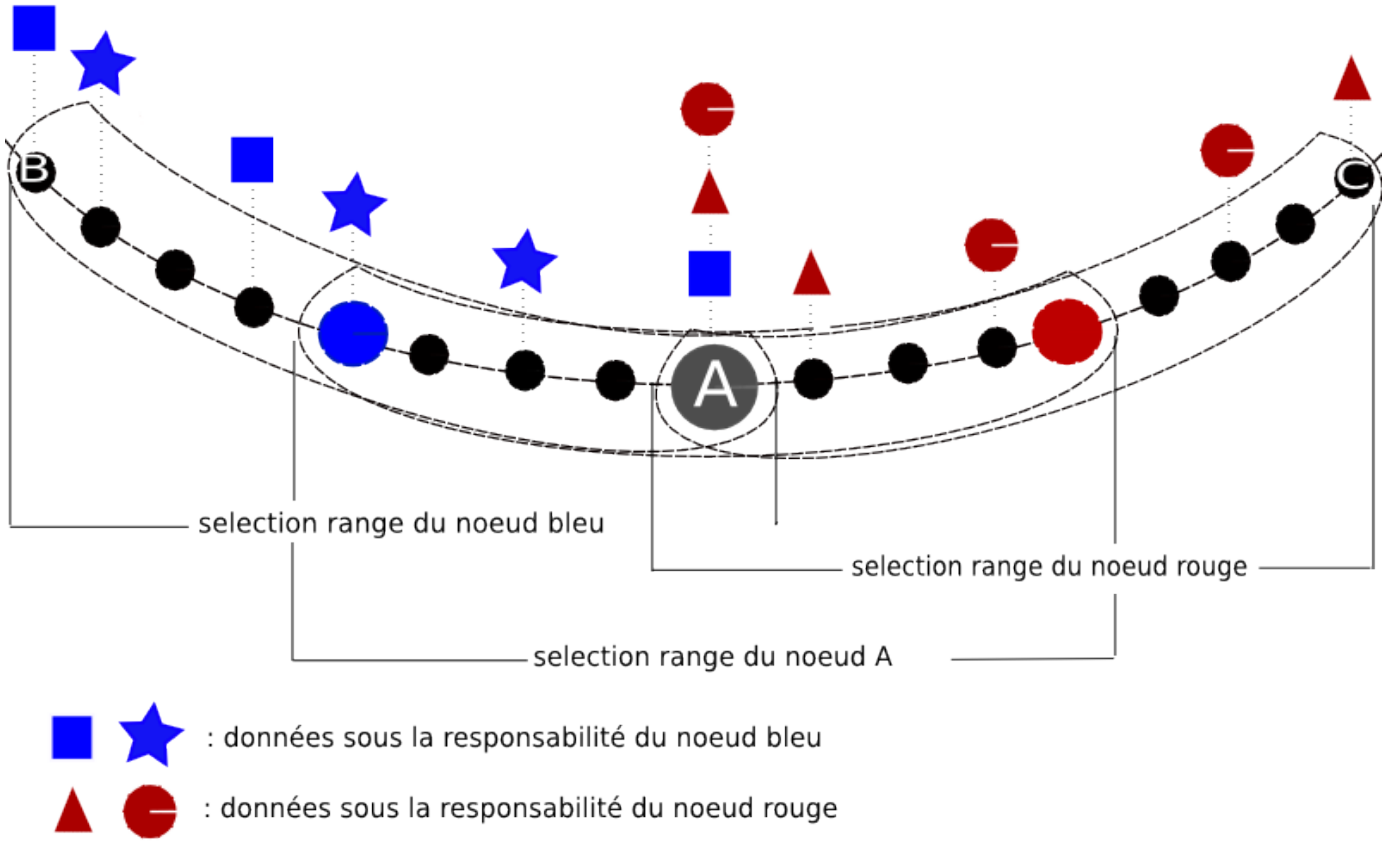
RelaxDHT : relâcher les contraintes de placement

- **Métadonnées**
  - Stockées par la racine et les noeuds hébergeurs
  - Pour chaque donnée : liste des hébergeurs et identifiant de la racine
- **Protocole de maintenance périodique**
  - Nœud racine responsable de vérifier la présence des hébergeurs
  - Nœuds hébergeurs responsables de vérifier la présence de la racine
  - Maintenance du *leafset* utilisée comme un détecteur de fautes
- **Gains**
  - Limitation des mouvements « inutiles »
  - Meilleure parallélisation des transferts



SPLAD : éparpillement et placement des données

- **Étude de la pérennité à long terme**
- **Notion d'espace de sélection (*selection range*) réglable**

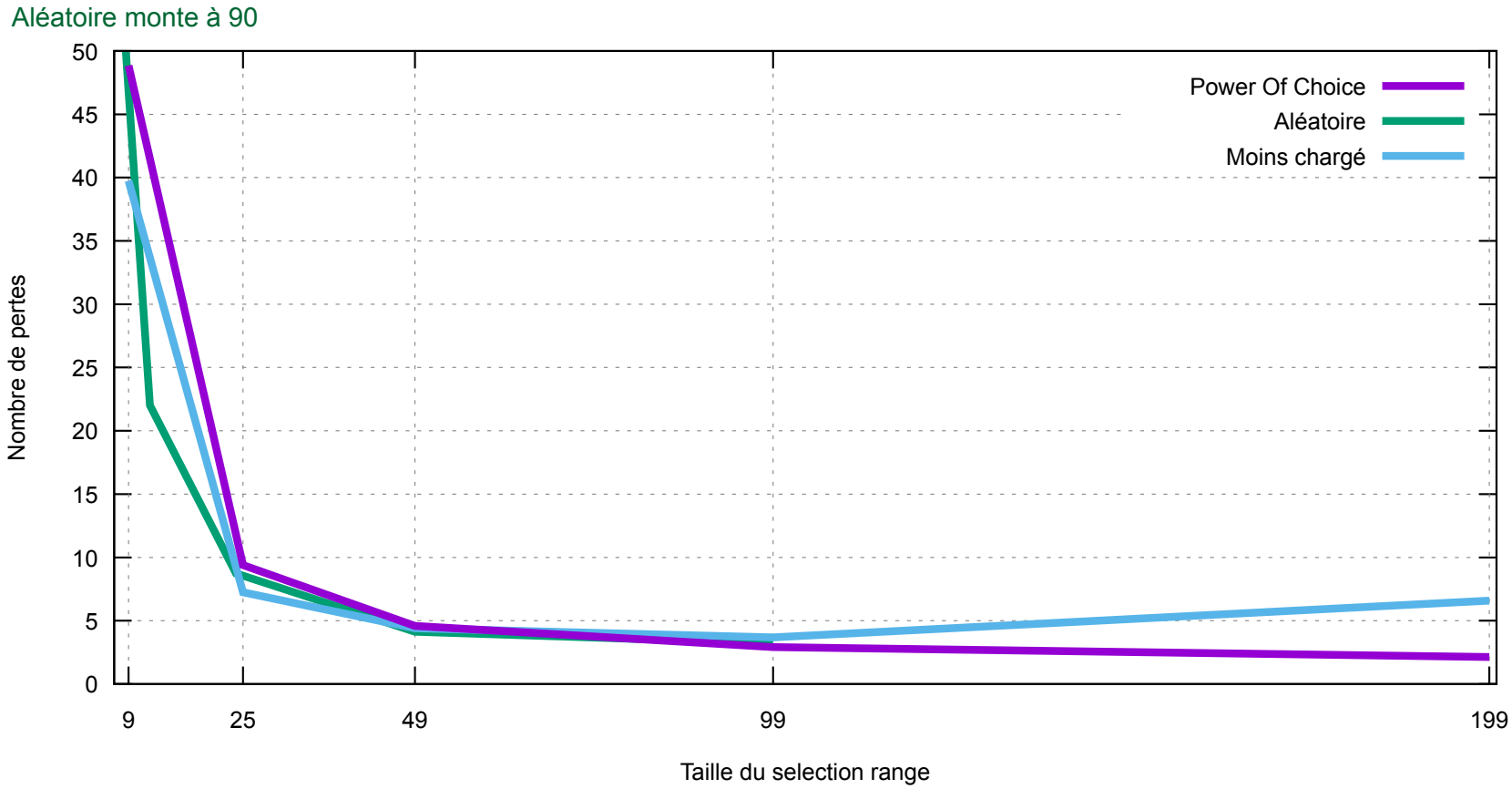


Placement au sein des *selection ranges*

- **Comment choisir un nœud hébergeur ?**
  - Aléatoire uniforme
    - Facile à mettre en œuvre
    - « Vieux » nœuds surchargés
  - Le moins chargé
    - Bonne répartition de la charge
    - Effet de bord : congestion réseau (surtout pour les grands *selection ranges*)
  - *Power of two choices* (le moins chargé parmi deux choisis aléatoirement)
    - Facile à mettre en œuvre
    - Bonne répartition de la charge

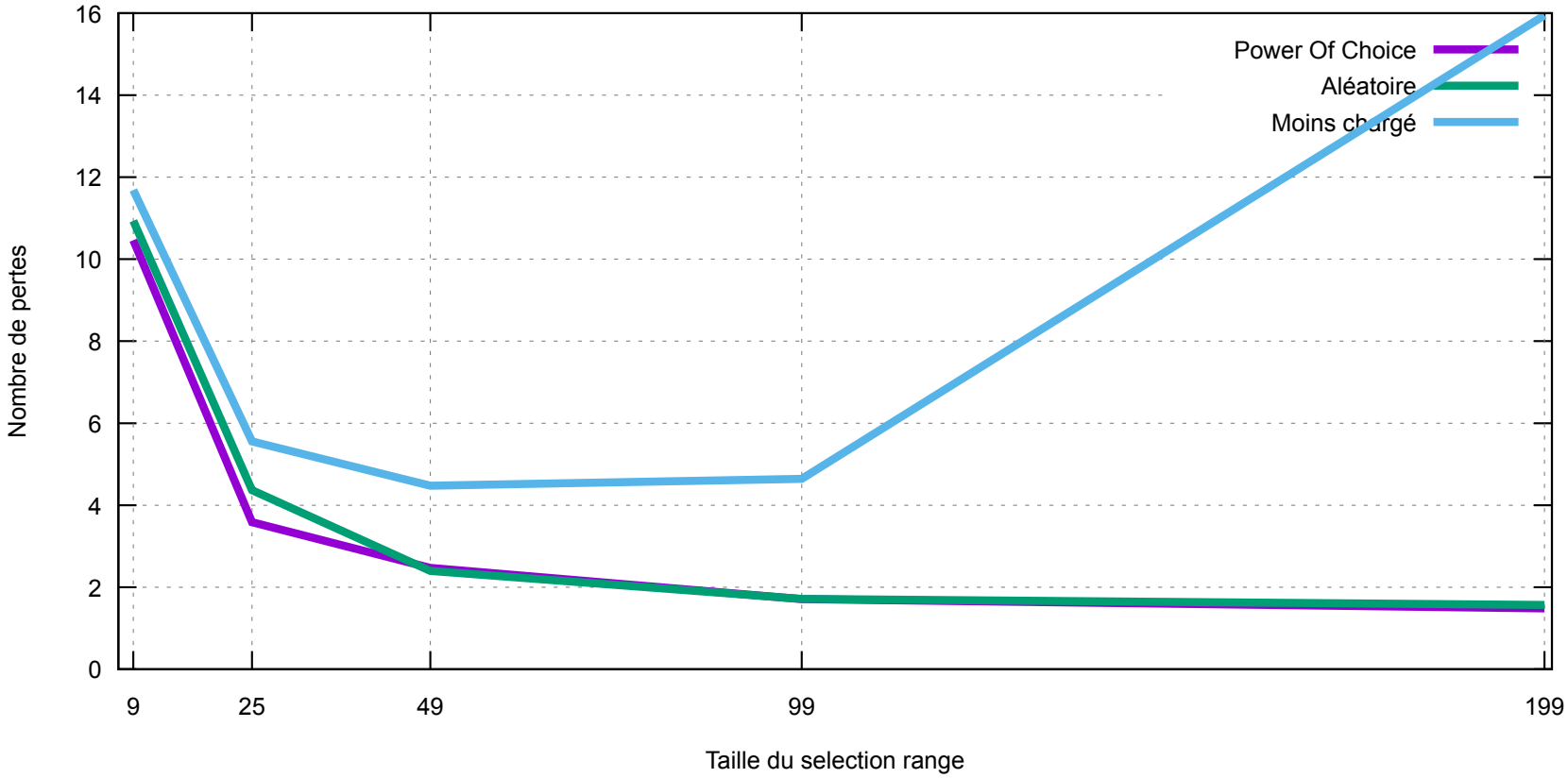
Impact de la taille du *selection range* et de la politique de placement

10.000 blocs de données, bande passante **asymétrique** (10Mb/1Mb)



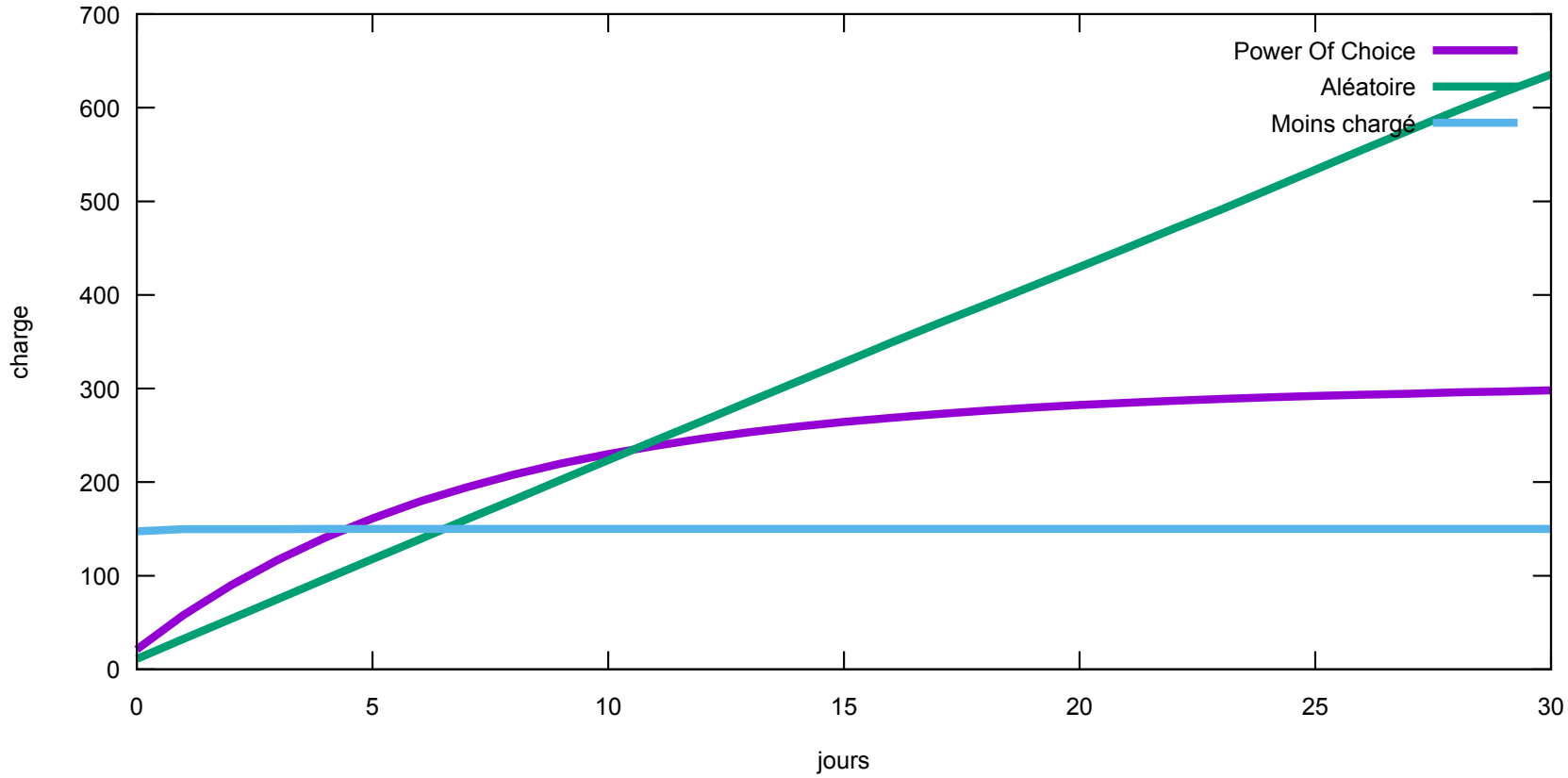
Impact de la taille du *selection range* et de la politique de placement

10.000 blocs de données, bande passante **symétrique** (5,5Mb)



Évolution de la charge de stockage

10.000 blocs de données - *selection ranges* de 200 nœuds



Impact du placement sur la pérennité
Leçons

- **Placement libre des copies**
  - Limite les mouvements inutiles
  - Prise en compte d'autres critères (performance)
- **Éparpillement des données**
  - Réparations plus rapides => moins de pertes
  - Attention : coût de la maintenance
- **Répartition de la charge de stockage**
  - Évite un déséquilibre du système
- **Autres facteurs à prendre en considération**
  - Fautes corrélées
  - Fiabilité des nœuds
  - Contraintes de performances/cohérence

CONCLUSIONS

&

PERSPECTIVES

Conclusions

- **Réplication de données**
  - Combien ?
  - Où ?
  - Quand ?
  - Comment ?
- **Nécessité de prendre en compte conjointement**
  - Tolérance aux fautes
  - Performance
  - Cohérence

⇒ **Adapter la réplication aux besoins applicatifs**

Intégration au sein du LISTIC
Groupe RSLR (*Réseaux et Systèmes Logiciels)*

- **Systèmes répartis à grande échelle**
  - ⇒ Vers des architectures hybrides (*cloud* –périphériques utilisateurs)
  - ⇒ Vers une prise en compte des coûts énergétiques
- **Expertise en gestion de données**
- **Intégration aux thématiques émergentes du groupe**
  - Objets Sages (Flavien Vernier, Ilham Alloui)
    - ⇒ Application au réparti: adaptation aux pics de charge
  - Modèle virtuel de terrain -- MVT (Frédéric Pourraz, Hervé Verjus)
    - ⇒ Stockage, interactions périphériques/*cloud*

Intégration au sein du LISTIC
CIT et CODE

- **Collaborations possibles avec les autres groupes**
  - CIT (*Connaissance, Images et Télédétection*)
    - ⇒ Réalité augmentée: répartition des traitements périphériques mobiles/*cloud*
  - CODE (*COmbinaison et DÉcision*)
    - ⇒ Gestion des données adaptée aux traitements des *Big Data*