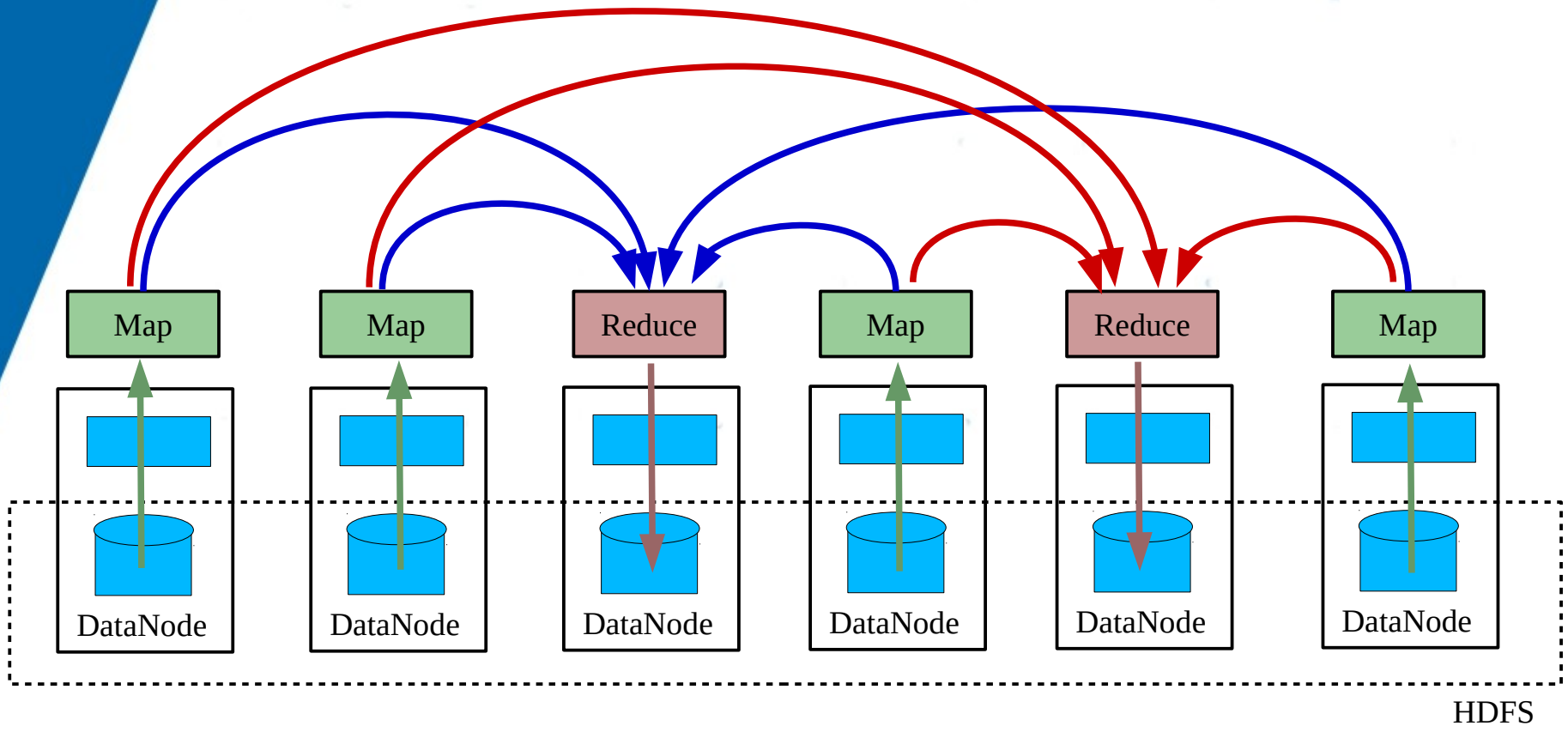
Hidoop – partie 2

Daniel Hagimont
Philippe Mauran

Rappel : schéma général



Objectifs

- 1ère étape
  - Consolider Hidoop
  - Votre plate-forme doit être fonctionnelle et fiable
  - Terminer première étude de scalabilité (WordCount)
- 2ième étape
  - Un thème au choix :
    - Tolérance aux pannes
    - Optimisation des performances
  - Différentes applications

Consolider Hidoop

- Doit fonctionner en réparti et être fiable pour de gros fichiers
  - Démonstration visée
    - 5-10 machines
    - 4-8 Go
- Quelques aberrations que j’ai observées

  ...

Consolider Hidoop

Aberrations

- HdfsWrite est une opération d’installation des données
  - Pas appelée dans startJob()
  - Pas incluse dans les mesures
- On ne charge pas un fichier entier pour l’envoyer
  - On lit et on envoie au fur et à mesure
- Les blocs sont gros
  - On ne charge pas un block en entier
- On crée des threads coté serveur
  - Pas de thread : exécution séquentielle
  - Threads coté client : ne passe pas à l’échelle
- Ne pas stocker les blocs sur NFS
  - Limite le parallélisme

Consolider Hidoop

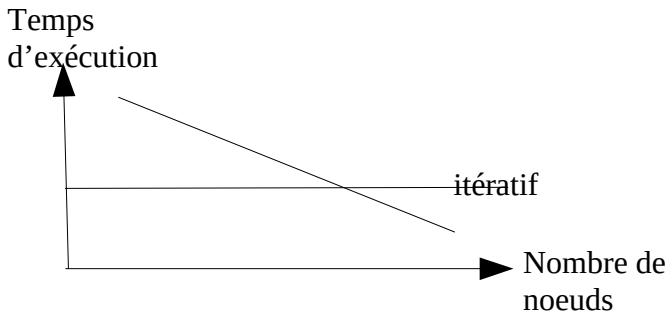
- WordCount
  - Vous devez pouvoir montrer une tendance et un speedup

	itératif	Hidoop
overhead	0	- - -
Speedup //	0	+ + +

Lequel est le plus grand

- A l'extrême
  - Avec un très grand fichier et une très grande taille de bloc
    - Très longs traitements en parallèle
    - Très peu d'overhead

Toujours essayer de prévoir le résultat avant les mesures, on cherche à vérifier une hypothèse



Tolérance aux pannes

- Résister à différents types de pannes
  - Pannes fail-stop : un élément tombe et ne revient pas (pas de panne intermittente)
  - Détection des pannes : mécanisme de heart-beat
  - Panne de processus plutot que machine
    - Panne HdfsServer
      - Réplication des blocks pour disponibilité
        - Gérer la disponibilité des blocks (NameNode)
      - Panne pendant une copie (HdfsWrite) : gérer la cohérence
      - Panne pendant une exécution : relancer les maps impactés (arrêter les maps, résultats indisponibles)
    - Panne Worker
      - Gérer la disponibilité des Workers
      - Relancer les maps impactés
    - Panne startJob
      - Reprendre l'exécution sans ré-exécution de tous les maps
- Défi
  - Exécuter un processus qui tue aléatoire des processus en cours d'exécution (startJob, HdfsServer, Worker)

Optimisation des performances

- Diagnostic / amélioration
  - Recherche des possibilités d'amélioration des performances
    - Profilage de votre système
      - Pleins d'outils de profilage dans l'environnement Java (journalise le temps passé dans chaque méthode)
    - Proposer des améliorations de l'implémentation
      - Sans changements stratégiques (ex : sérialisation ou byte[])
      - Avec changements stratégiques (ci-dessous)
- Améliorations stratégiques
  - Envoyer directement les données des maps vers le reducer
    - Plutôt que de les stocker dans un fichier local et faire un HdfsRead
      - En particulier si les maps ne se terminent pas en même temps
      - Le reducer peut commencer le travail avant
  - Gérer plusieurs reducers s'exécutant sur les slaves
    - Une fonction sur les clés (K) détermine le reducer qui reçoit
    - Un reducer génère un résultat final localement
  - Exploiter les coeurs d'une machine (dans un Worker)
- Défi
  - Un fichier de 8Go, 5 machines, obtenir le meilleur temps d'exécution

Différentes applications

- Différentes applications
  - On a différents goulots d'étranglement en fonction de la quantité de CPU utilisée dans l'application (IO bound ou CPU bound)
    - CPU bound : les IO attendent
    - IO bound : le CPU attend
  - Exemples d’applications
    - Algorithme X de Knuth
    - Page ranking
    - Décimales de Pi
  - À priori, page ranking est IO bound et X de Knuth est CPU bound.
- Etudier les tendances

Suivi du projet

- Inscription sous moodle
- 5 séances de suivi (une semaine sur 2)
  - Une semaine
    - Consolidation
    - Choix de l'option (tolérance aux pannes, optimisation des performances)
    - Choix des contributions (spécification)
- Rendu final (fin mars)
- Une séance de restitution /bilan (mi-avril)