Google File System

Recherche avancée
Outils linguistiques

Recherche Google | J'ai de la chance

Sommaire

1. **Google en chiffres**

2. **Les raisons d'être de GFS**

3. **Les grands principes de GFS**
   - **L'architecture**
   - **L'accès de fichier en lecture**
   - **L'accès de fichier en écriture**
   - **Bilan**

4. **Les Evolutions et Alternatives**

5. **Conclusion & Questions**

Google en Chiffre (1)

1996 : Larry Page et Sergey Brin mettent au point un moteur de recherche (BackRub).

1998 : - BackRub devient Google
- L'entreprise Google voit le jour.

2000 : 10 nouvelles langues pour le moteur de recherche (dont le Français)

2001 : - Larry et Sergey font appel à Eric Schmidt pour diriger la société.
- 27 nouvelles langues

2004 : ~ 100 langues

2008: ~ 150 langues

Google en Chiffre (2)

	2000	2004	2009
Chiffre d'affaire*	78	3 189	23 659
Bénéfices*	12	399	6 524
Nbre d'employés	364	8 763	20 304

* En millions de dollars

Google en Chiffre (3)

Nombre de pages indexées par Google :

- 1998 : 26 millions
- 2000 : 1 milliard
- 2008 : 1 000 milliards

Nombre de serveurs :

- 2000 : 6 000 serveurs
- 2003 : 15 000 serveurs
- 2005 : 100 000 serveurs
- 2006 : 450 000 serveurs
- 2007 : 1 million de serveurs
- 2008 : 2 millions de serveurs
- 2010 : > 2% du parc mondial

Hum… Où peuvent-ils les cacher … ?

Google en Chiffre (4)

USA : 20 Datacenters
Europe : 12 Datacenters
Ailleurs: 6 Datacenters

=> ~ 50 000 serveurs par Datacenter

Pourquoi une telle quantité de serveurs ?

- Données liées à l'indexation du web
- Multiplication des services Google:
  - Gmail
  - Google Books
  - Google Documents
  - GDrive
  - ect…
- Gestion des pannes (redondance)
- … Et bien sur, la gestion de la publicité ($)

En 2008, Google gérait 20 Pétaoctets de données par jour !

Rappel:

1 Pétaoctet = 1 000 Téraoctets

1 Téraoctet = 1 000 Gigaoctets

En 2010 : ~~Certainement~~ plus !

Les problématiques de Google :

- Pouvoir stocker tout pleins de fichiers
- … de taille (très) importante
  (potentiellement plusieurs dizaines de giga)
- … disponible sur le réseau.
- et assurant une haute disponibilité des informations.

Comment faire ?

Un système de fichier par Google, pour Google.

=> GFS (Google File System)

GFS en quelques mots:

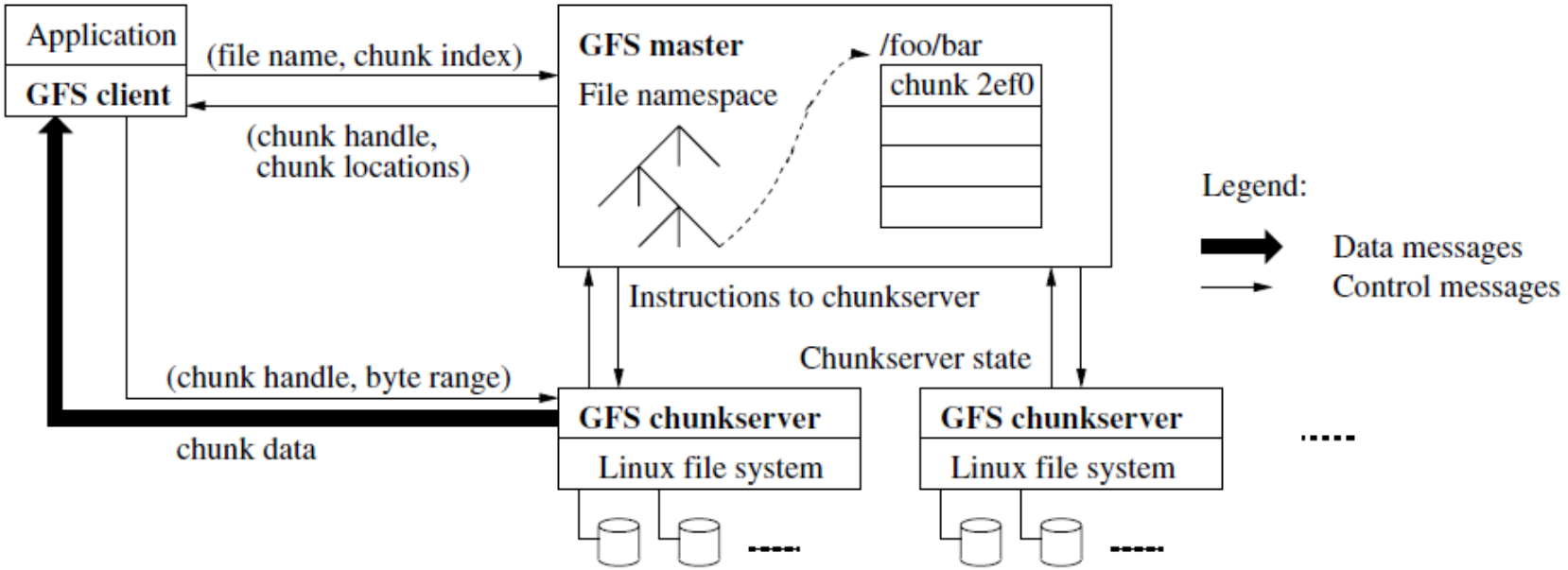
- **c'est un système de fichiers distribués**
- **Présent dans les Datacenters de Google depuis 2001**
- **Optimisé pour le stockage de données**
- **Les fichiers sont découpés en morceaux (Chunks), puis répartis sur plusieurs machines.**

Architecture (1)

Plusieurs clients accèdent aux fichiers d'un cluster GFS

Un Cluster GFS, c'est :

- **Un Master Server (Répliqué pour assurer un backup)**
- **Des Chunks Servers (jusqu'à plusieurs milliers)**



Architecture (2)

Les Opérations classiques sur GFS :

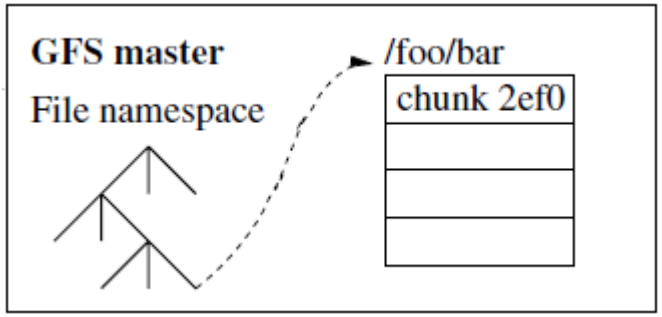
- **Create / Delete**
- **Open / Close**
- **Read / Write**

Mais aussi :

- **Append**
- **Snapshot**

Le Master Server (1)

- **Il tient à jours les Metadatas en RAM :**
  - Les Namespaces (arborescence des répertoires)
  - Le mapping: fichiers -> Chunks
  - La localisation des Chunks Servers
  - Informations de contrôle
- **Garbage Collection des Chunks obsolètes**
- **Migration des chunks entre Chunks Server :**
  - Réplication
  - Load Balancing



Euh… En RAM, ça veut dire que si le server crash, on perd tout ?

Le Master Server (2)

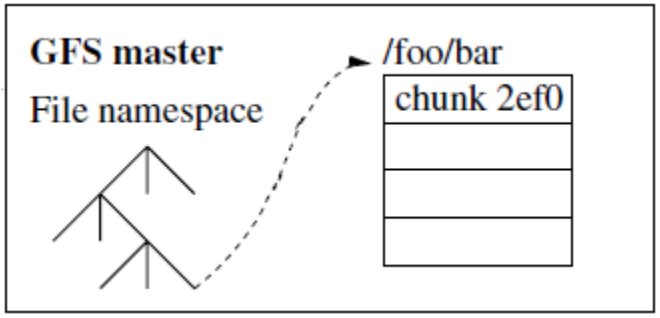
• **Il y a un Check Point, qui contient les informations du dernier état stable.**

• **Chaque opération liée aux Metadatas des Namespaces, et des mapping fichiers -> chuncks est écrite dans un fichier Operation Log.**

• **Le fichier Operation Log est écrit sur le disque local, et à distance.**

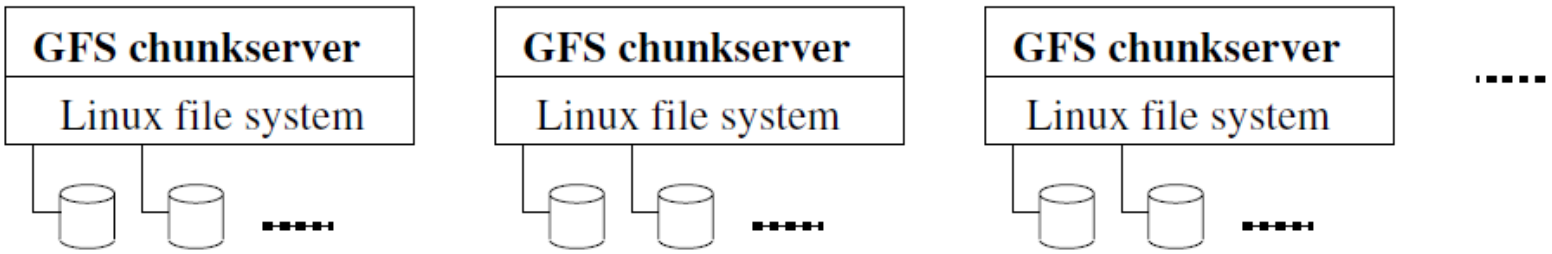
• **En cas de plantage, le serveur rejoue les dernières opérations après un état stable.**

• **Les Metadatas liées à la localisation des Chunks Server sont uniquement en RAM**



Les Chuncks Servers

- **Contient les Chunks**
- **Un Chunk est répliqué 3 fois dans le cluster (configurable)**
- **Taille des Chunks : 64Mo**
- **Chunks identifiés par un ID unique de 64 bits (Chunk Handle)**
- **Les Chunks sont stockés sur le système de fichier de la machine**



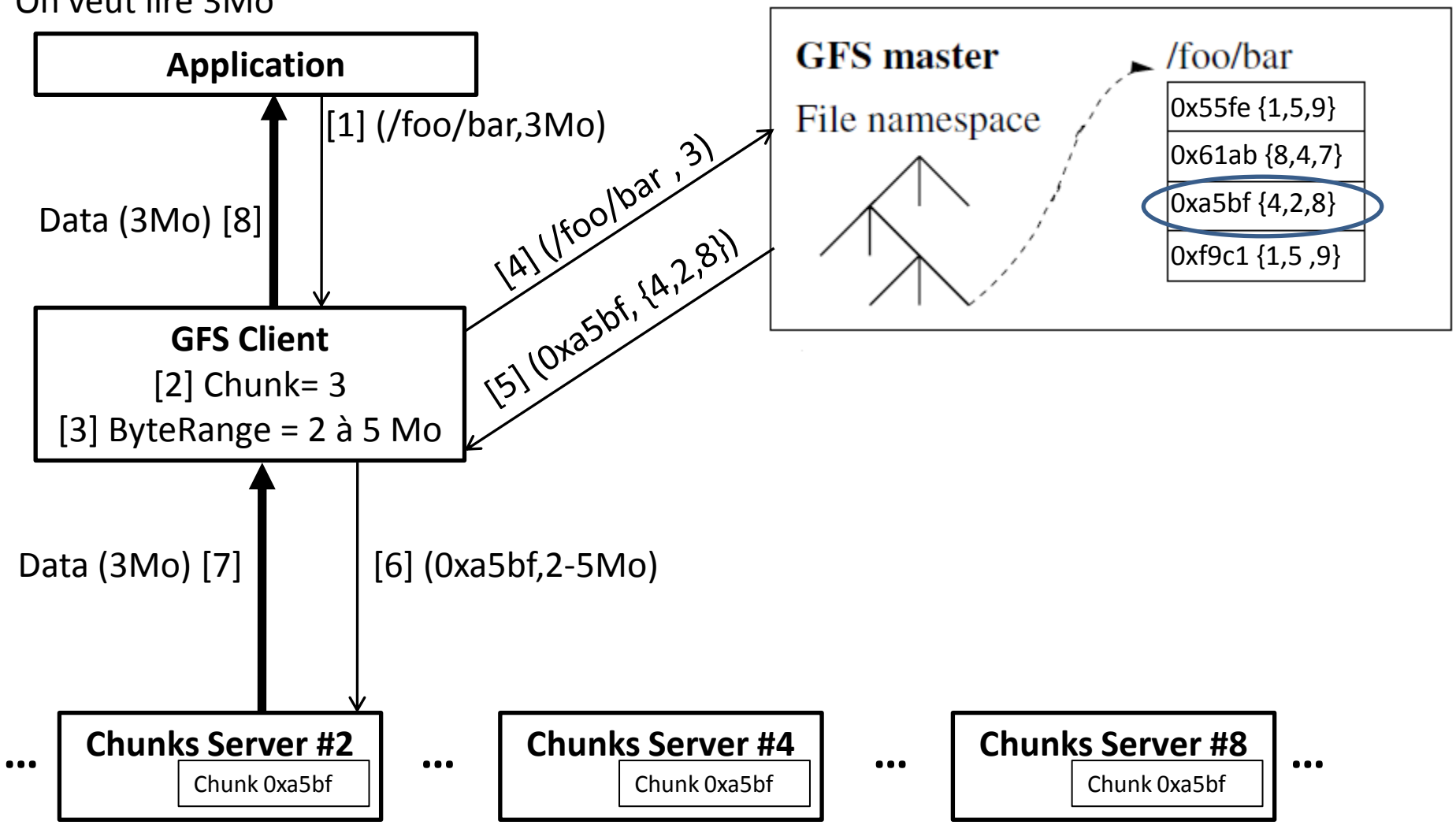
Read (1)

Pré-requis: le fichier est ouvert , et la position est connue

1. **l'application transmet au client GFS, le nom du fichier et le nombre de bit à lire**
2. **Le client GFS calcul le Chunk concerné (⌜ position/64Mo ⌝)**
3. **Le client GFS calcul le Byte Range ([position - ⌞position/64Mo⌟ *64Mo] à [position - ⌞position/64Mo⌟ *64Mo +taille])**
4. **Le client demande au Master Server l'emplacement du Chunk (nom du fichier + n° du Chunk)**
5. **Le master renvoie le Chunk Handle et les localisations du Chunk**
6. **Le client transmet à un des Chunks Servers l'ID du Chunk et le Byte Range désiré**
7. **Le Chunk Server envoie les données au client**
8. **Le client transmet les données à l'application**

Read (2)

Fichier ouvert : /foo/bar
Position : 130 Mo
On veut lire 3Mo



Write (1)

Pré-requis: le fichier est ouvert , et la position est connue

1. **l'application transmet au client GFS, le nom du fichier et le nombre de bit à lire**
2. **Le client GFS calcul le Chunk concerné et le Byte Range**
3. **Le client demande au Master Server l'emplacement du Chunk**
4. **Le master renvoie le Chunk Handle et les localisations du Chunk (primaire + secondaires)**
5. **Le client envoie les données à écrire à chaque réplicas (stockées dans un buffer)**

Write (2)

6. **Le client envoie un Write au Server Chunks primaire (Chunk Handle + Byte Range)**

7. **Le Chunks Server primaire détermine l'ordre de sérialisation des données du buffer et les écrit dans le Chunk en suivant cet ordre**

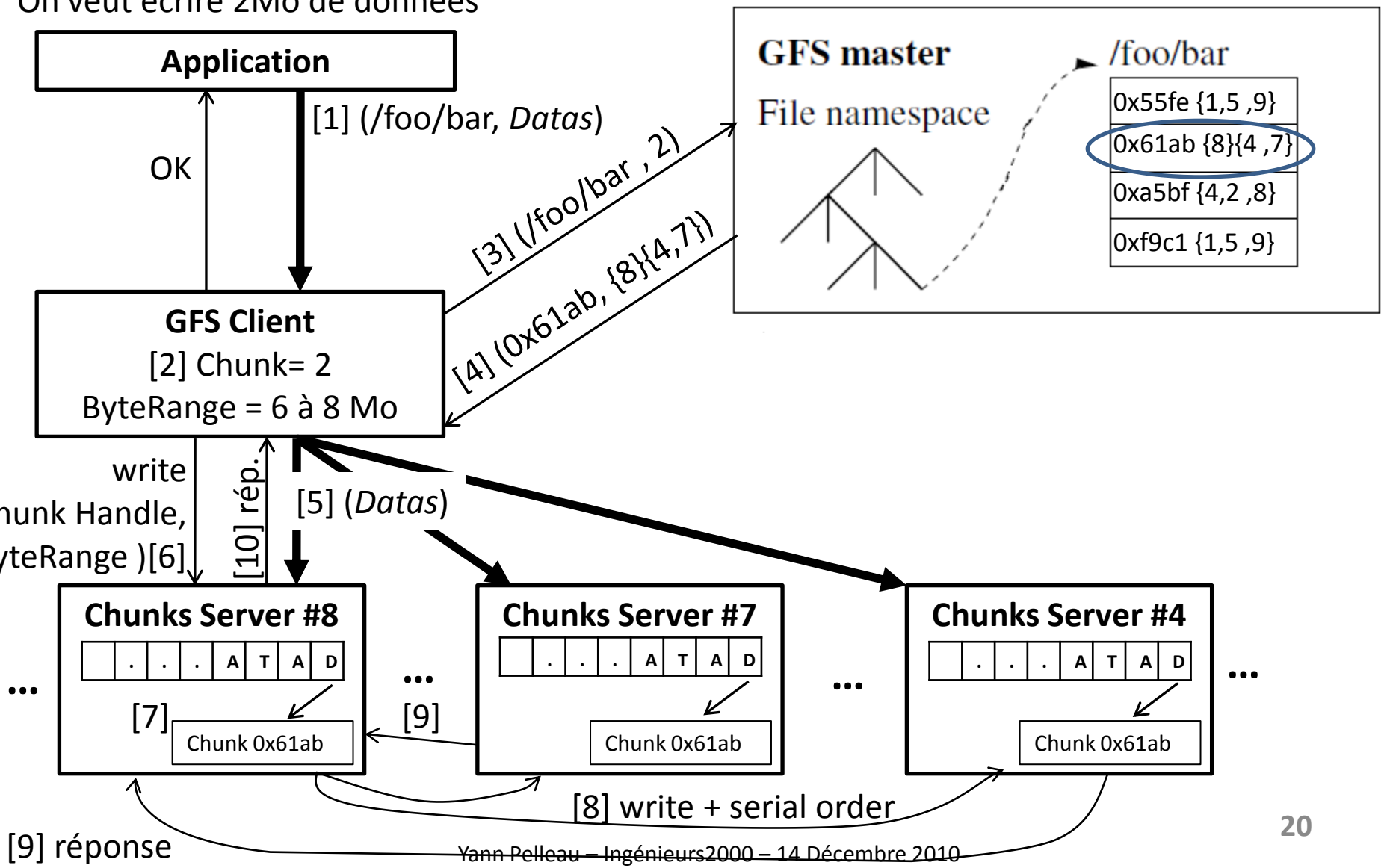
8. **Le Chunks Server primaire transmet l'ordre de sérialisation aux Chunks Servers secondaires pour qu'ils exécutent le Write**

9. **Les Chunks Servers secondaires répondent au primaire**

10. **Le Chunks Servers primaire répond au client**

Write (3)

Fichier ouvert : /foo/bar
Position : 70 Mo
On veut écrire 2Mo de données



Autres Opérations

Append :

- **Très utilisé chez Google**
- **C'est un Write dans le dernier chunk**
- **Pas de Byte Range**
- **Si le Chunk atteint 64 Mo, il faut continuer dans un nouveau Chunk**

Snapshot :

- **Permet de créer une copie d'un fichier ou d'une arborescence**
- **Le Master Server s'assure qu'il n'y pas de modification en cours.**
- **Le Master Server log l'opération**
- **Les Metadatas du Master Server sont dupliquées**
- **Le Master Server demande aux Chunks Servers de dupliquer les Chunks**

BenchMark

En 2003 :

Used for research and development.

Used for production data processing.

Cluster	A	B
Chunkservers	342	227
Available disk space	72 TB	180 TB
Used disk space	55 TB	155 TB
Number of Files	735 k	737 k
Number of Dead files	22 k	232 k
Number of Chunks	992 k	1550 k
Metadata at chunkservers	13 GB	21 GB
Metadata at master	48 MB	60 MB

Cluster	A	B
Read rate (last minute)	583 MB/s	380 MB/s
Read rate (last hour)	562 MB/s	384 MB/s
Read rate (since restart)	589 MB/s	49 MB/s
Write rate (last minute)	1 MB/s	101 MB/s
Write rate (last hour)	2 MB/s	117 MB/s
Write rate (since restart)	25 MB/s	13 MB/s
Master ops (last minute)	325 Ops/s	533 Ops/s
Master ops (last hour)	381 Ops/s	518 Ops/s
Master ops (since restart)	202 Ops/s	347 Ops/s

Point sur GFS

- Les clients peuvent faire des Read/Write/Append en parallèle
- Haute Tolérance aux pannes
- Performant
- Matériel standard, et donc pas cher.

- Pas optimisé pour les petits fichiers
- un seul Master Server
- “High sustained bandwidth is more important than low latency”

GFS 2 (alias Colossus)

Quand ?

En 2009

Pourquoi ?

Corriger les différents problèmes/contraintes:

- Un seul master

- Chunk de 64Mo

- Les latences

Comment :

- **Master Server distribué**
- **Taille des Chunks : 1Mo**

Hadoop

- **Projet Apache**
- **C'est un Système de fichier Distribué d'inspirant de GFS**
- **Les organisations utilisant Hadoop:**
  - **Yahoo**
  - **Facebook**
  - **Powerset (Microsoft Bing)**
  - **IBM**
- **Inclut BigTable et MapReduce**
- **Programmé en Java ☺**

Conclusion

- **GFS a dépassé les espérances de Google**
- **GFS a permis a Google de se développer rapidement**
- **Les concepts de GFS sont utilisé par un grand nombre de concurrents**
- **Finalement GFS n'est pas si compliqué que ça. Il fallait juste y penser…**

Questions