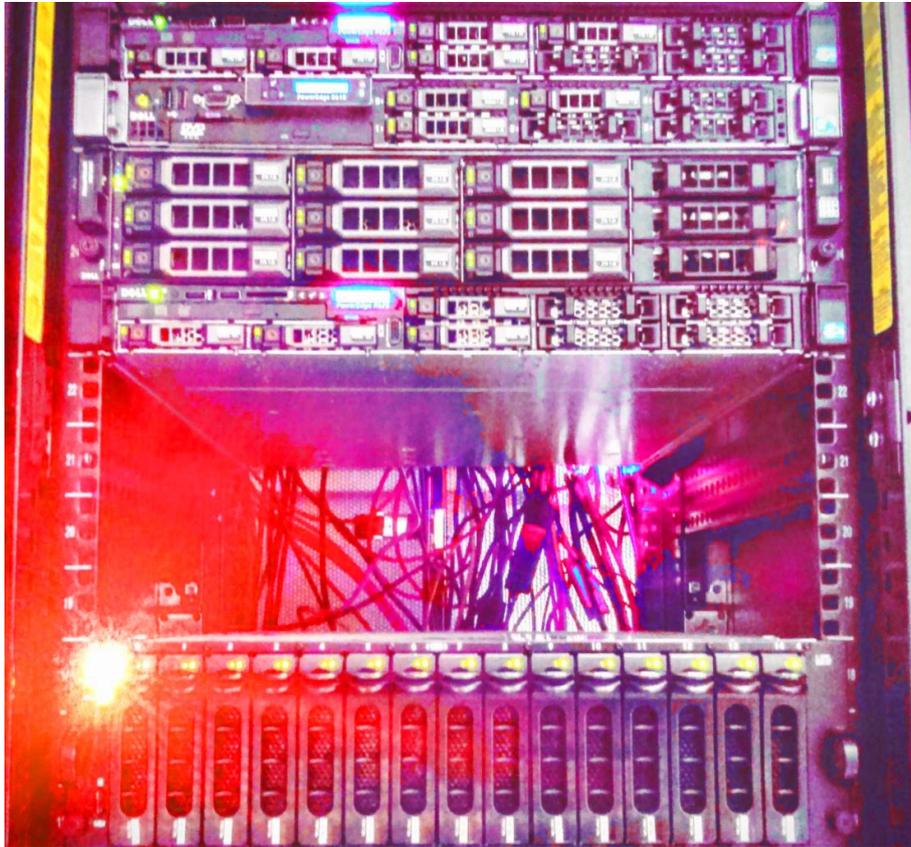
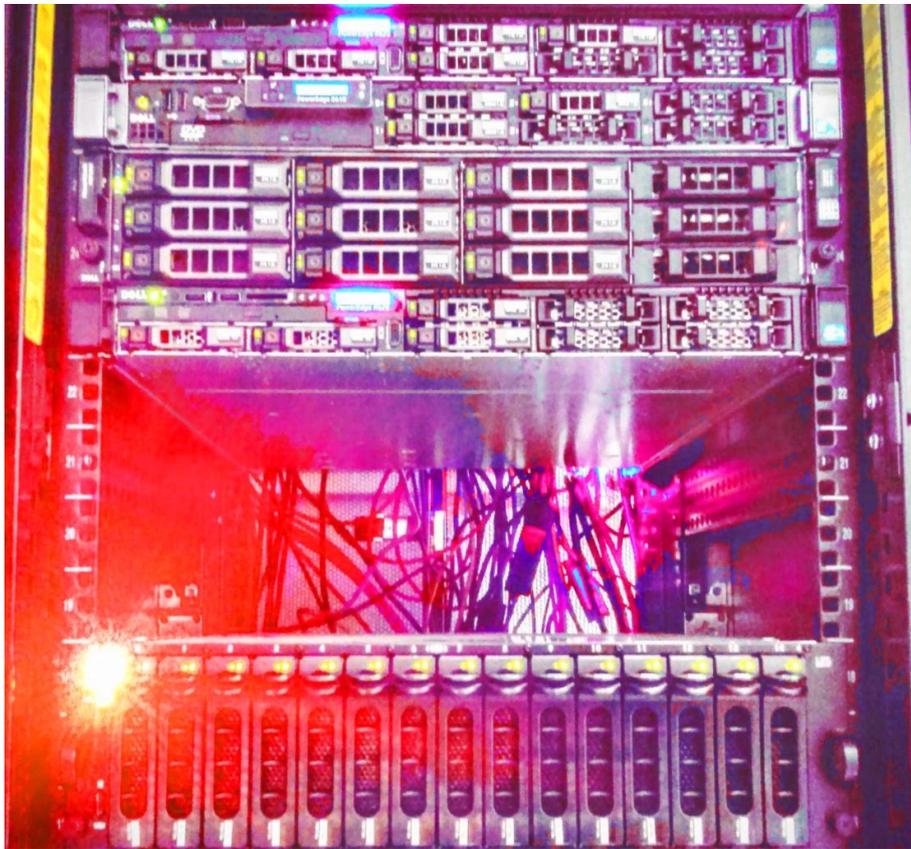


# Systemes de fichiers réseau & stockage objet



*Un système de fichier réseau fournit une abstraction au **système d'exploitation** tandis que le stockage objet fournit une abstraction à une **application**.*

# Systemes de fichiers réseau & stockage objet



- **Plan**

- Introduction
- Topologie réseau type
- Virtual File System → VFS dans l'espace noyau
- Système de fichiers dans l'espace utilisateur → FUSE
- Appels de procédures distants (RPC)
- Network File System → NFS
- Server Message Block → SMB
- Stockage Objet → Ceph

- **Objectifs**

- Définitions, vocabulaire et usages

# Introduction

- **Avantages**

- Partage

- Fichiers disponibles à grande échelle → Nombre important de clients avec des systèmes divers
- Partage depuis un point unique
  - Diminution du nombre de copies
  - Moins d'incohérences

- Gestion des disques

- Système NAS → pas de file d'attente unique par périphérique en mode bloc
- Extension facile des volumes (virtuels ou non) par ajout au serveur NFS → LVM

- Architecture

- Gestion des verrous et des files d'attentes côté serveur → côté client avec les solutions SAN
- *Snapshots* → au niveau système de fichier (LVM)
- Sauvegardes → au niveau serveur NFS uniquement

# Introduction

- **Inconvénients**

- Limitation technique

- Lancement système impossible sur NAS
- Coût CPU plus important côté hyperviseur → logiciel de communication avec le serveur NAS

- Performances

- Latence possible → usages applications transactionnelles
- Pas de *multipath* avec un système NAS

- Architecture

- Manque de nouvelles fonctions liées à la virtualisation
- Contraintes de sécurité → autorisations d'accès depuis les clients
- Accès sensibles aux ruptures de connexions réseau

# Topologie type



## Réseau IP «frontal»

- Hôte ↔ hôte
- Application ↔ système de fichiers
- Client ↔ Serveur
- NFS / SMB
- NAS

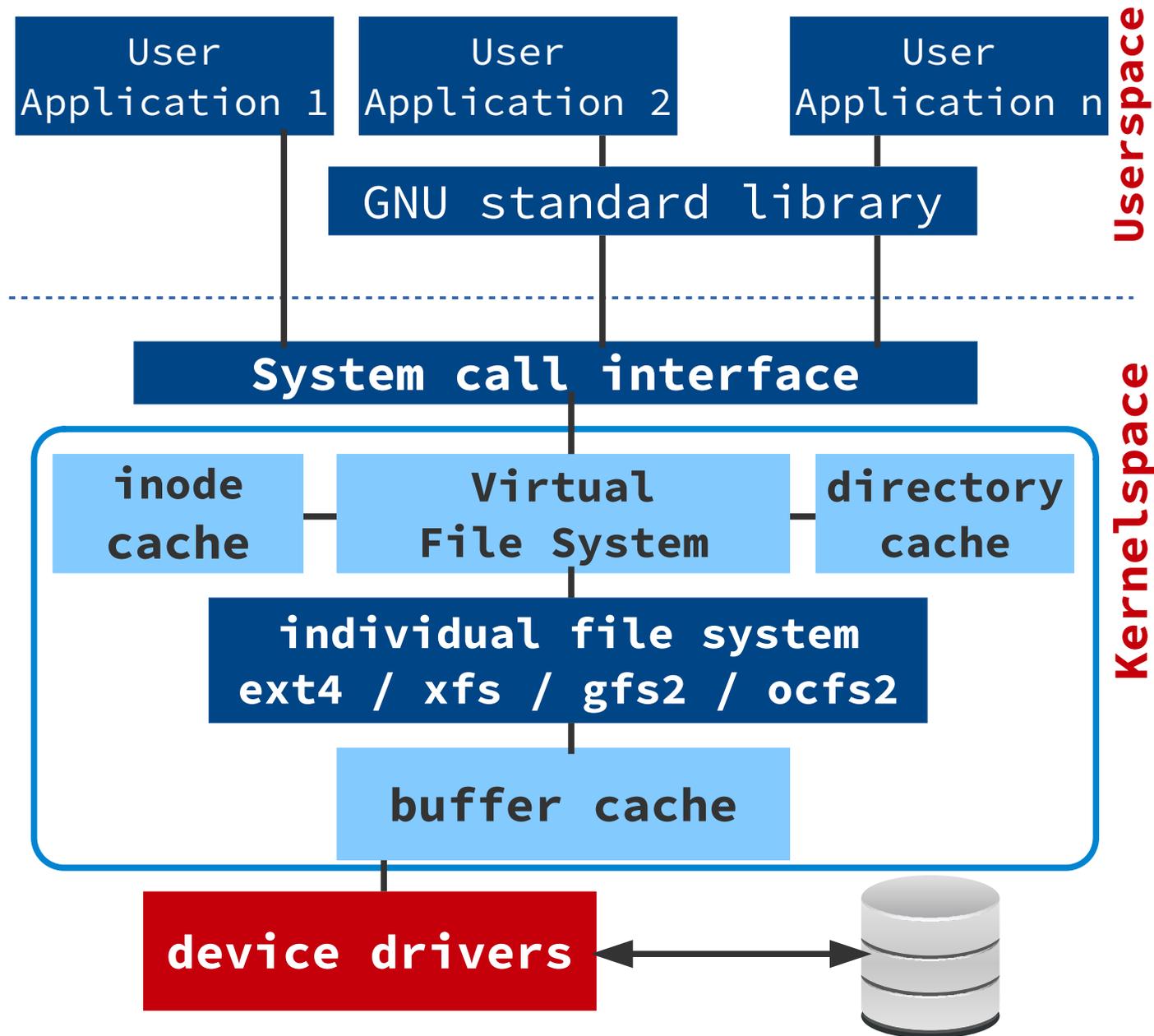
## Réseau de stockage «dorsal»

- Hôte ↔ stockage
- Système de fichiers ↔ périphérique
- Application ↔ périphérique
- VFS noyau
- SAN

# Systeme de fichiers virtuel (VFS)

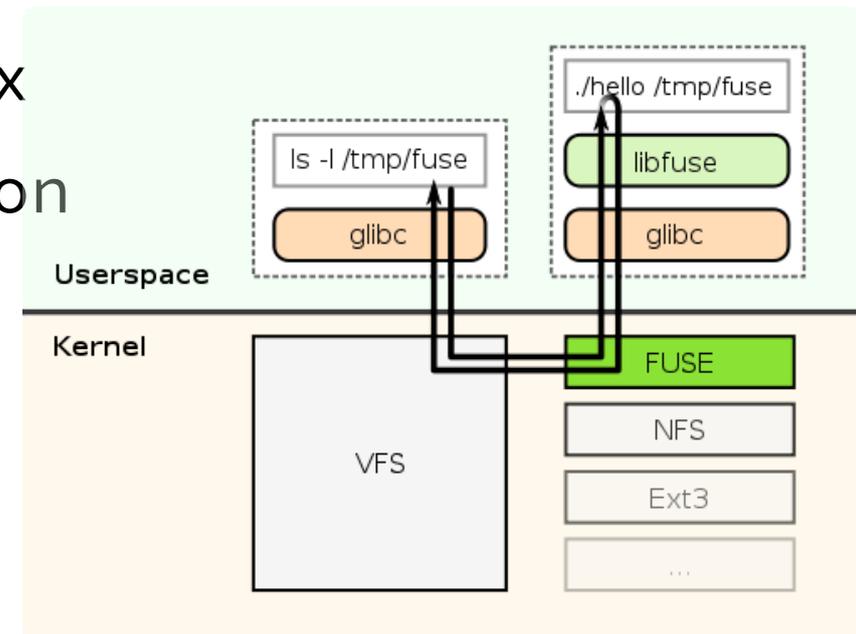
- **Qu'est-ce qu'un système de fichiers virtuel (VFS) ?**
  - Un service du noyau
  - Une interface entre les applications ou processus et les dispositifs de stockage
  - Une bibliothèque standard
- **Quel est le but du VFS ?**
  - Un accès transparent au stockage vis-à-vis des utilisateurs et/ou des applications
  - Une gestion uniforme du contrôle d'accès aux fichiers et répertoires
  - Un système de nommage cohérent entre fichiers locaux et réseau
  - Des performances d'accès uniformes → cache

# Systeme de fichiers virtuel (VFS)



# Systeme de fichiers dans l'espace utilisateur

- **Étape intermédiaire avant le stockage objet → bibliothèque FUSE**
  - Portabilités entre noyaux
  - Interface de programmation
    - Tous les langages
    - Toutes les applications
  - Avantages
    - Performances
    - Mise au point dans l'espace utilisateur
  - Inconvénient
    - Métadonnées système uniquement



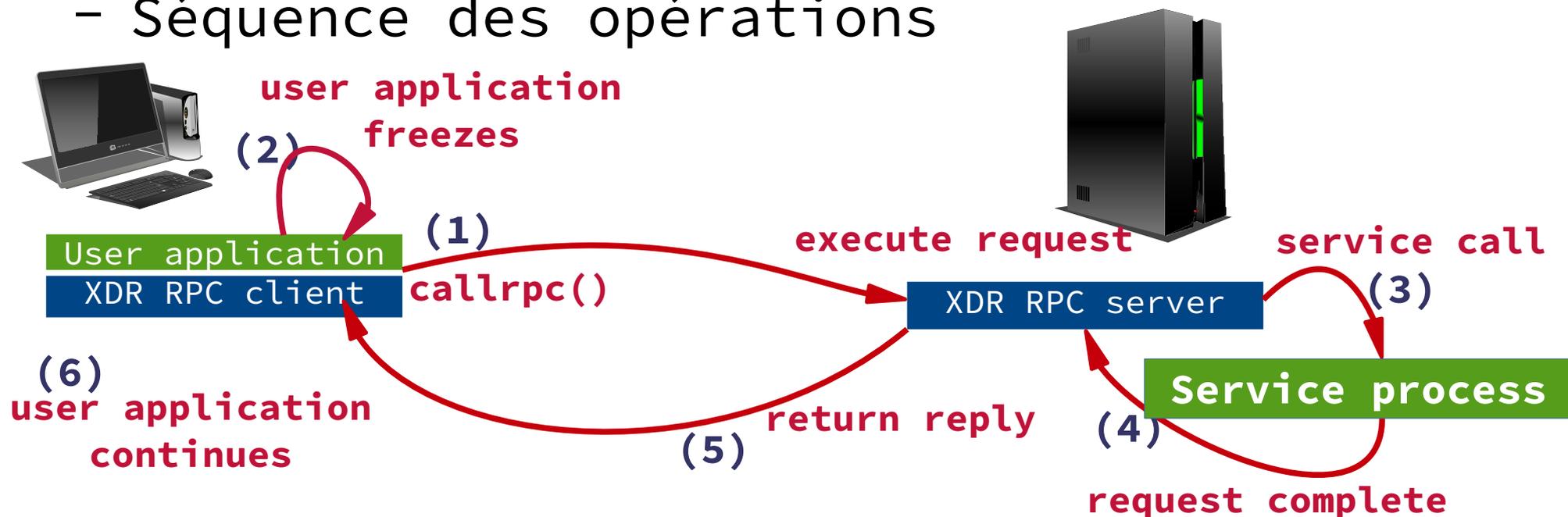
# Mécanisme d'accès réseau

- **Appel de procédure distant**

- *Remote Procedure Call* (RPC)

- Extension des appels de procédure locaux
- Exécution de code sur un hôte distant

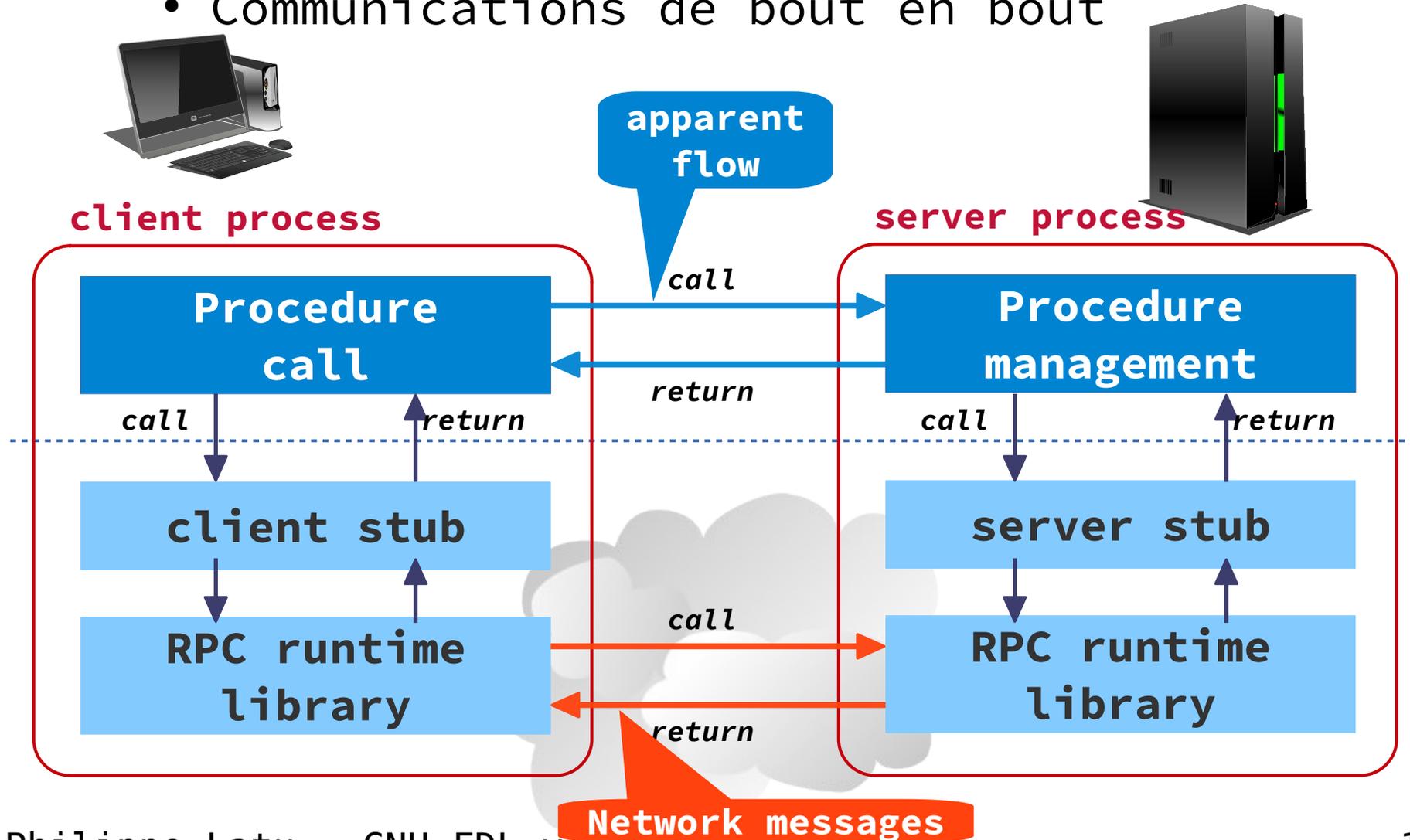
- Séquence des opérations



# Mécanisme d'accès réseau

## • Modélisation des flux RPC

- Communications de bout en bout



# Network File System (NFS)

- **NFSv2 1983**

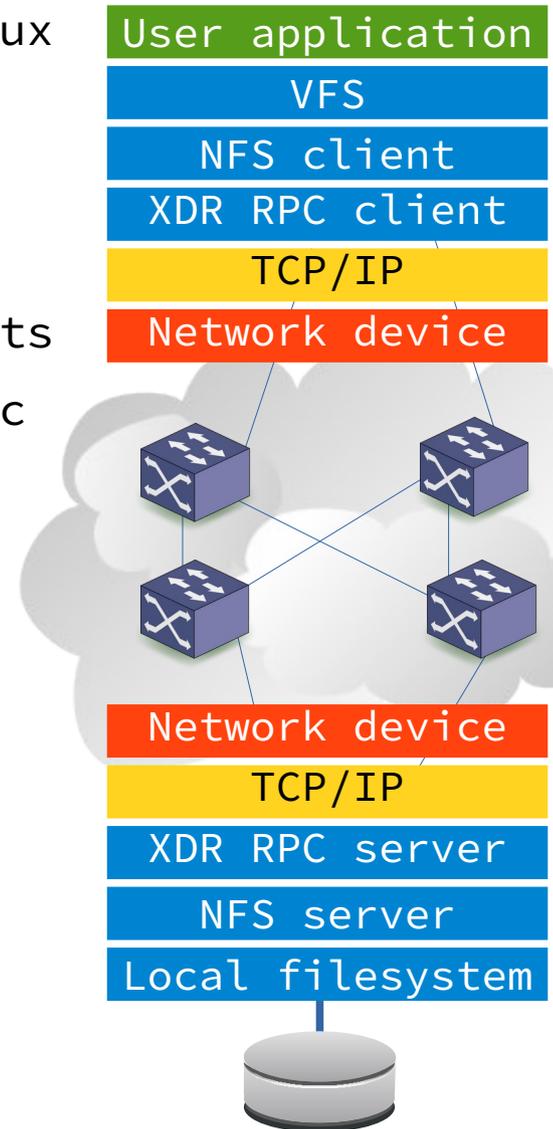
- Protocole de transport UDP uniquement sur réseaux locaux
- Performances médiocres en écriture

- **NFSv3 1995**

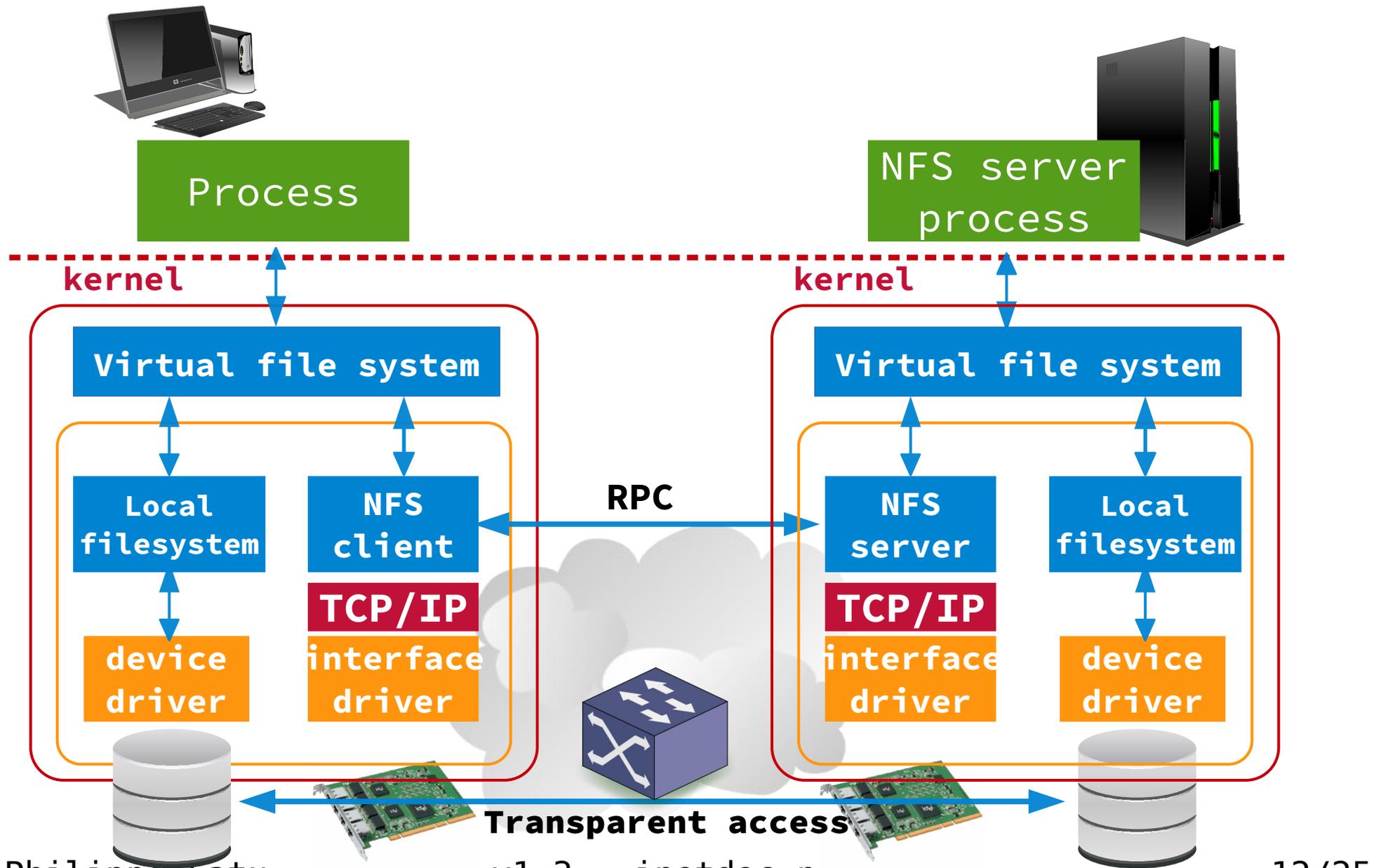
- Exportation de systèmes de fichiers POSIX 64 bits
- Protocoles de transport UDP et TCP toujours avec multiplexage de ports
- Performances améliorées en écriture

- **NFSv4 2003 → mise à jour 2015**

- Réduction des temps de latence
- Communications sur un port unique → tcp/2049 (filtrage plus facile)
- Appels de procédures groupés → *compound* RPC
- Chiffrements des flux

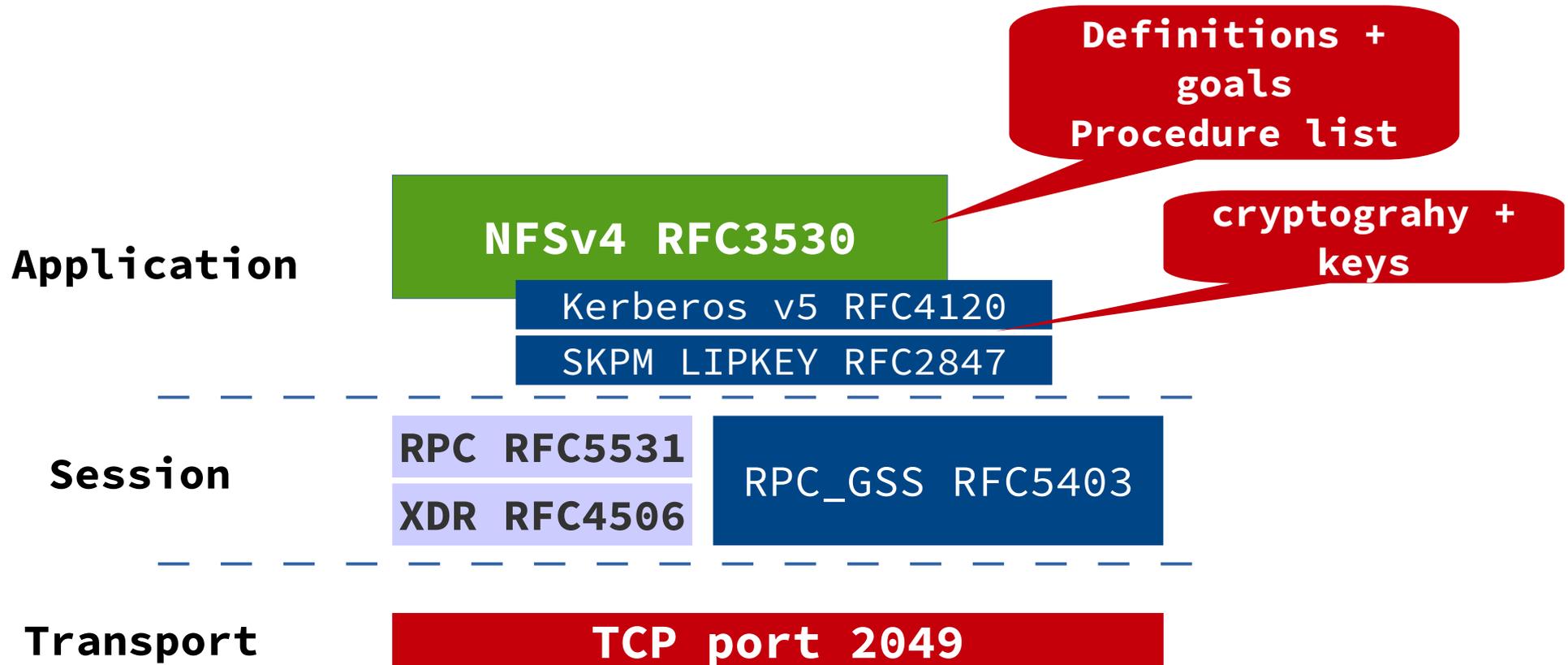


# Network File System (NFS)



# Network File System (NFS)

- **Pile des protocoles NFSv4.x**



# Network File System (NFS)

- **Fonctions principales**

- Système de fichiers distribué → accès aux fichiers distant identiques aux accès locaux
- Modèle client/serveur
  - Serveur → répertoires locaux accessibles aux clients
  - Clients → montage des répertoires distants
- Hiérarchique par nature → chaînes de répertoires et fichiers

- **Usages**

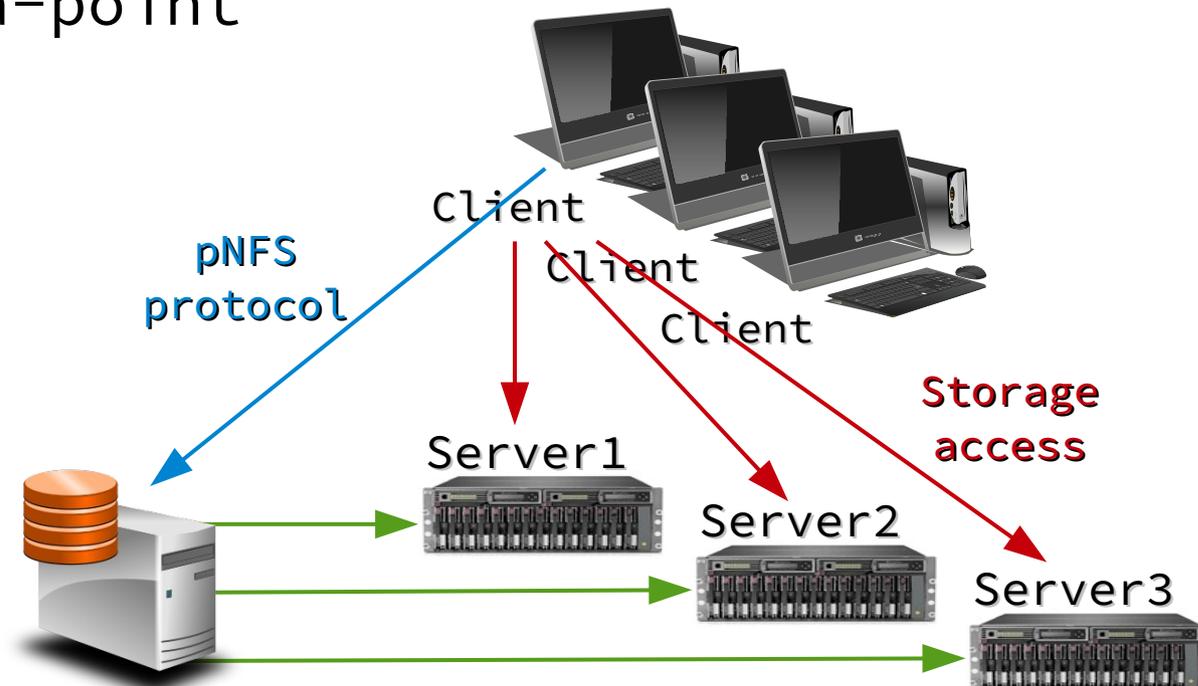
- Partage de répertoires : utilisateurs, données et applications qui sont exécutées localement
- Bases de données et VM datastores
- VMware supporte NFSv4.1 en mode client pour le stockage de fichiers VMDKs
- Amazon supporte NFSv4.0 à travers AWS Elastic File System (EFS)

# Network File System (NFS)

- **Évolutions du protocole**

- *Parallel* NFS (pNFS)

- Aggrégation de serveurs NFS autonomes
    - Relation entre client et serveur de type point-à-point



# Network File System (NFS)

- **Évolutions du protocole**

- *Parallel* NFS (pNFS)

- Client → lecture ou écriture d'un fichier
    - Serveur → autorisation d'accès au fichier
    - Pagination (*stripe map*) du fichier fournie au client
    - Client → lecture ou écriture directe sur tous les serveurs

- Plus de goulot d'étranglement

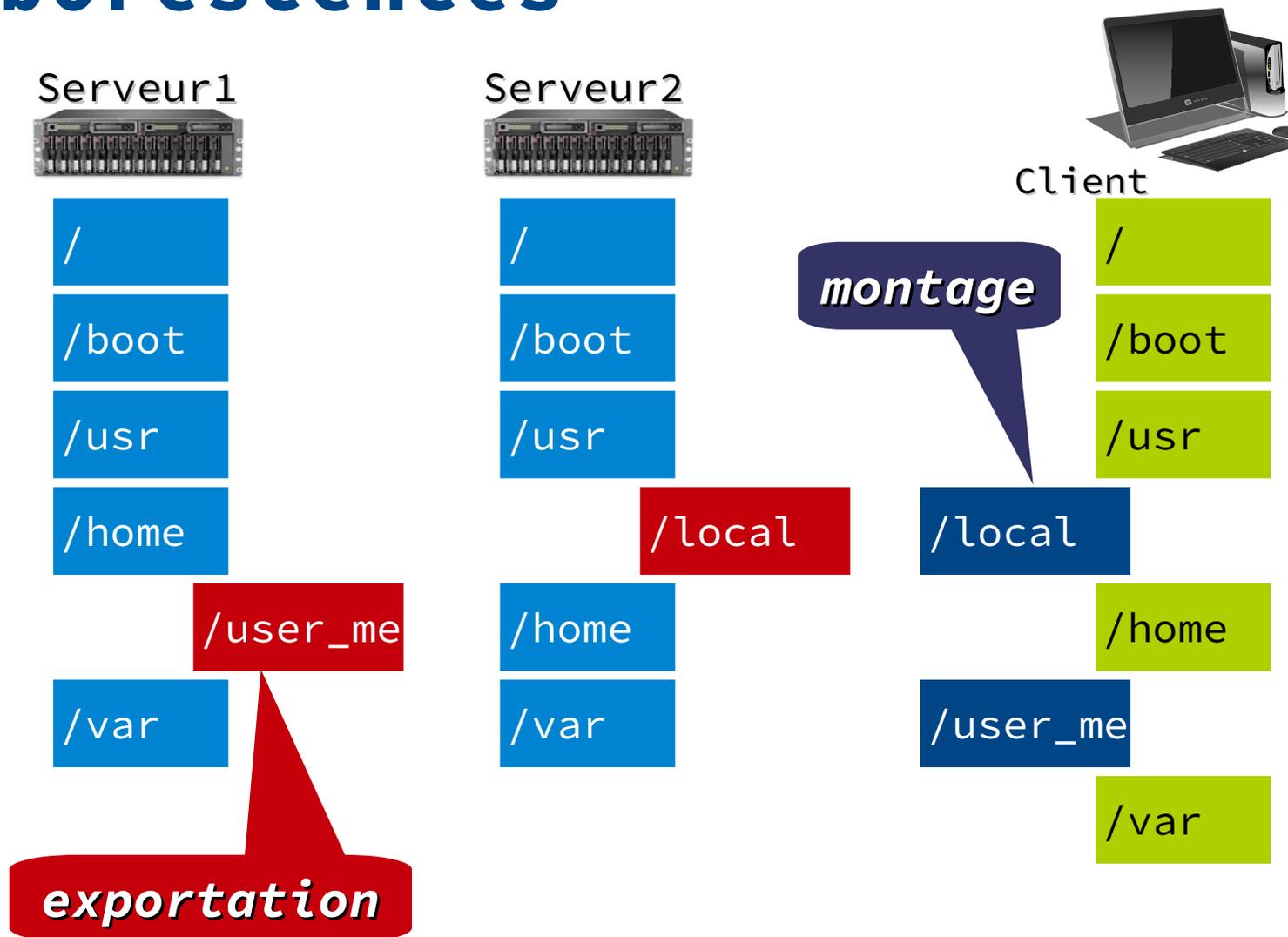
- Plus de relation point-à-point avec le serveur
    - Amélioration des accès aux fichiers de grande taille

- Gestion améliorée

- Répartition de charge entre clients et serveurs
    - Espace de nommage unique préservé

# Network File System (NFS)

- **Arborescences**



# Server Message Block (SMB)

- **Fonctions principales**

- Mode de partage standard des systèmes Microsoft
- SMB 3.0 → Windows Server 2012 → optimisé pour applications serveur
- Améliorations SMB 3.0
  - Performances → clusters
  - Tolérance aux pannes
  - Sécurité → algorithmes AES

- **Usages**

- Intégration Hyper-V et produits Windows Server
- Bases de données SQL Server

- **SMB Direct**

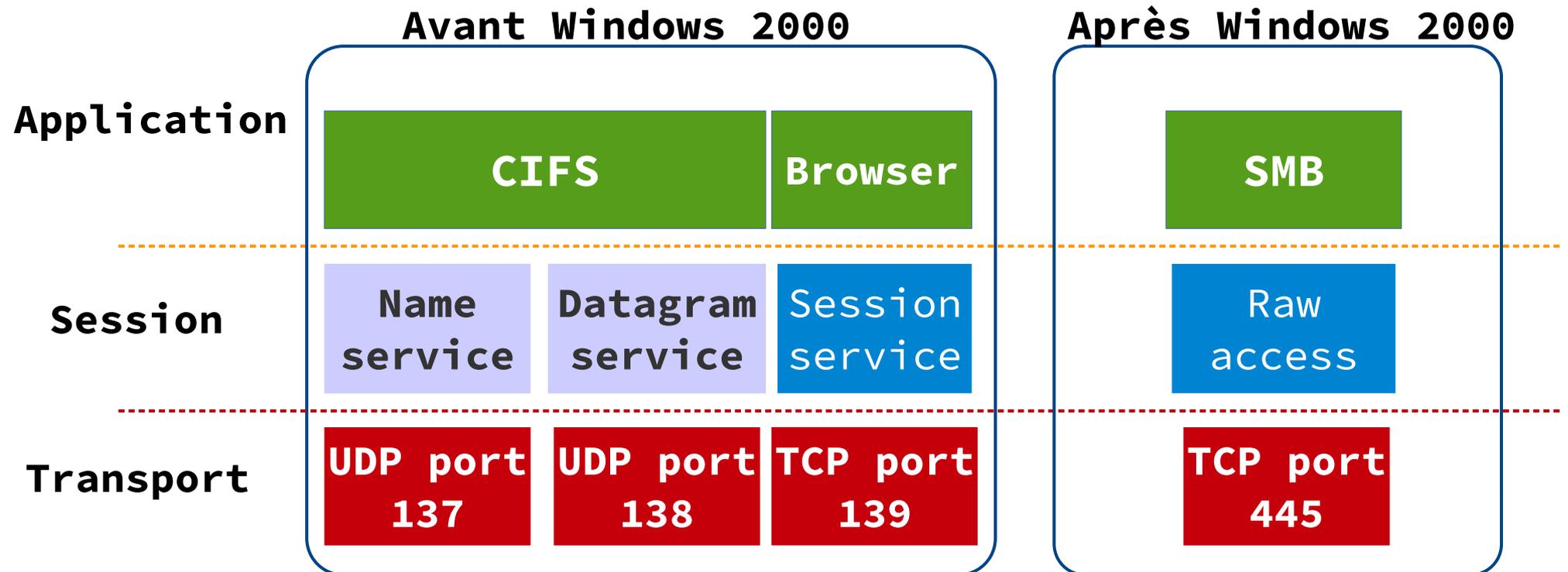
- *Remote Direct Memory Access* (RDMA)
- Canaux d'accès «directs» de la mémoire d'un hôte à l'autre à travers le réseau → Hyper-V

- **SMB Multichannel**

- Distribution des flux entre interfaces multiples

# Server Message Block (SMB)

- Pile des protocoles SMB – CIFS**



# Object storage

- **Stockage en mode objet**
  - Trouver/Chercher des données à partir d'expressions rationnelles
  - *Cloud storage* → transformation en base de données
  - Plus de *Cloud* se développe → plus on peut trouver d'informations
  - Plus les Métadonnées sont importantes → plus on peut formuler de requêtes complexes (*Big Data*)
    - Exemple → recherche de données similaires à l'aide de la classification des métadonnées



<b>Nutrition Facts</b>	
<b>8 servings per container</b>	
Serving size	2/3 cup (55g)
<hr/>	
Amount per 2/3 cup	<b>Calories 230</b>
<hr/>	
% DV*	
12%	<b>Total Fat</b> 8g
5%	<b>Saturated Fat</b> 1g
	<b>Trans Fat</b> 0g
0%	<b>Cholesterol</b> 0mg
7%	<b>Sodium</b> 160mg
12%	<b>Total Carbs</b> 37g
14%	<b>Dietary Fiber</b> 4g
	<b>Sugars</b> 1g
	<b>Added Sugars</b> 0g
	<b>Protein</b> 3g
<hr/>	
10%	<b>Vitamin D</b> 2mcg
20%	<b>Calcium</b> 260mg
45%	<b>Iron</b> 8mg
5%	<b>Potassium</b> 235mg
<small>* Footnote on Daily Values (DV) and calories reference to be inserted here.</small>	

# Object storage

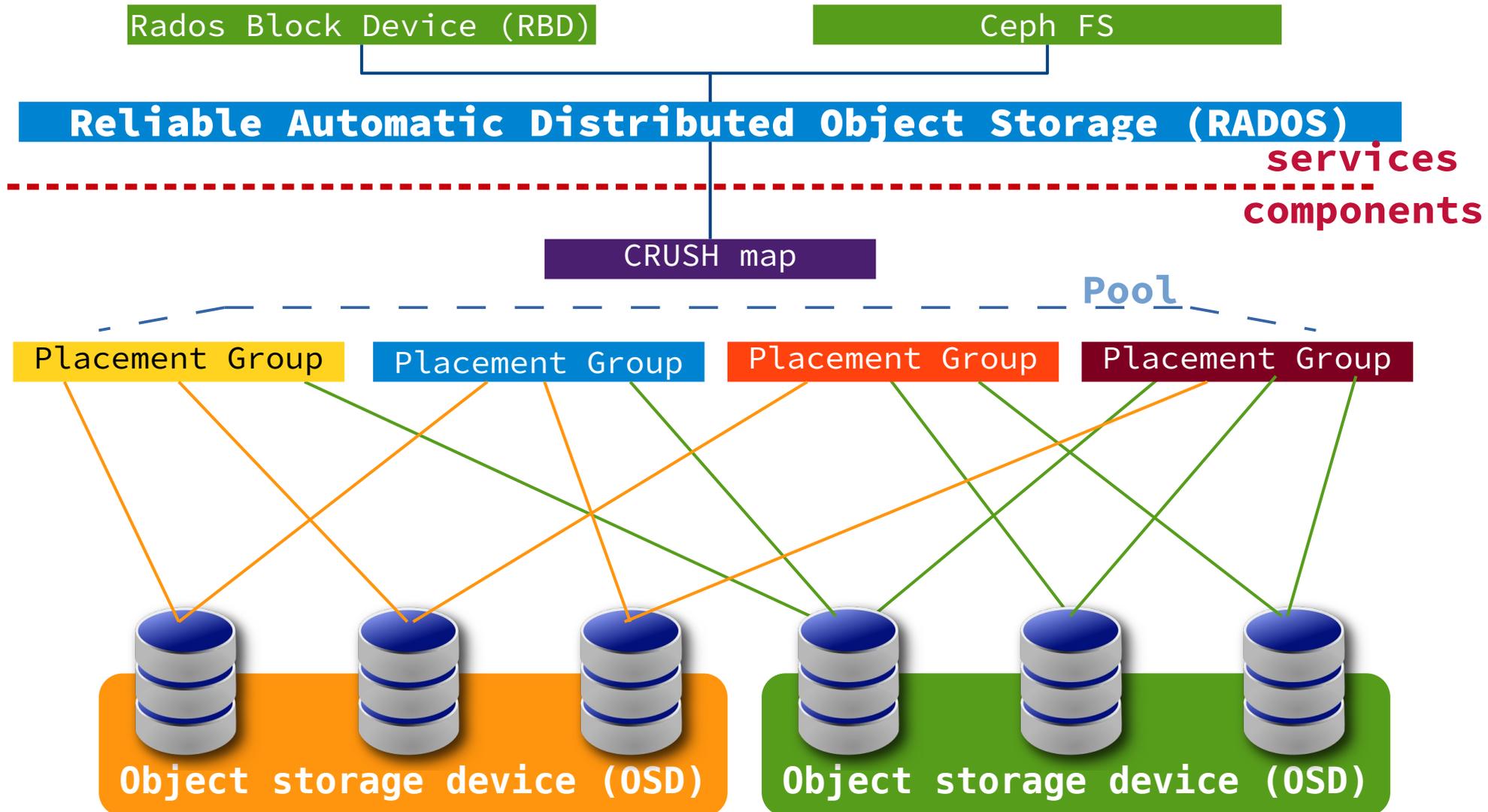
- **Stocker et récupérer un objet**

- Un serveur stocke un objet sur un seul nœud → algorithmes de sélection basés sur la topologie
- Politique de stockage → algorithmes basés sur l'occupation des nœuds, les performances et les métadonnées système
- Schémas de nommage cohérent → accès global
- Réplication → entre nœuds et/ou sites en fonction des lectures



<b>Nutrition Facts</b>	
<b>8 servings per container</b>	
<small>Serving size</small>	<small>2/3 cup (55g)</small>
<hr/>	
<small>Amount per 2/3 cup</small>	<b>Calories 230</b>
<hr/>	
<small>% DV*</small>	
<b>12%</b>	<b>Total Fat</b> 8g
<b>5%</b>	<b>Saturated Fat</b> 1g
	<b>Trans Fat</b> 0g
<b>0%</b>	<b>Cholesterol</b> 0mg
<b>7%</b>	<b>Sodium</b> 160mg
<b>12%</b>	<b>Total Carbs</b> 37g
<b>14%</b>	<b>Dietary Fiber</b> 4g
	<b>Sugars</b> 1g
	<b>Added Sugars</b> 0g
	<b>Protein</b> 3g
<hr/>	
<b>10%</b>	<b>Vitamin D</b> 2mcg
<b>20%</b>	<b>Calcium</b> 260mg
<b>45%</b>	<b>Iron</b> 8mg
<b>5%</b>	<b>Potassium</b> 235mg
<small>* Footnote on Daily Values (DV) and calories reference to be inserted here.</small>	

# Architecture CEPH



# Architecture CEPH

- **Object Storage Devices (OSDs)**

- Analogie avec les volumes de sockage
- OSDs multiples → RAID inutile

- **Pools**

- Regroupement de *Placement Groups*
- Stratégie de performance → *tiering*

- **CRUSH maps**

- Distribution des objets vers les OSDs
- Une correspondance par *Pool*

- **RADOS**

- Transformation → données/objets → stockage partagé performant

- **RDB**

- Périphérique de type bloc → découpage en *stripes*

# Synthèse

- **NAS**

- Fichiers → vue du système d'exploitation
- Partage réseau avec un bon rapport performance / coût

- **Stockage objet**

- Objets → vue des applications
- Métadonnées → recherches et requêtes complexes
- Algorithmes / stratégies de distribution du stockage à grande échelle → *Cloud*

# Ressources

- **Stockage réseau**

- `Ciscolive.com BRKINI-1011`

- **Introduction à CEPH**

- `http://docs.ceph.com/docs/jewel/start/intro/`

- **Samba experience**

- `https://sambaxp.org/`