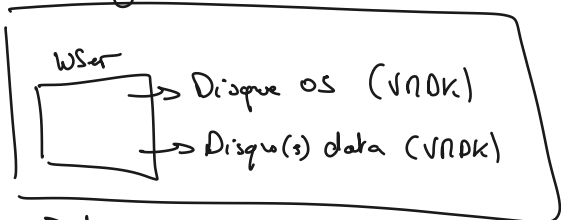


# Bonjour tout le monde



Data . VMDK

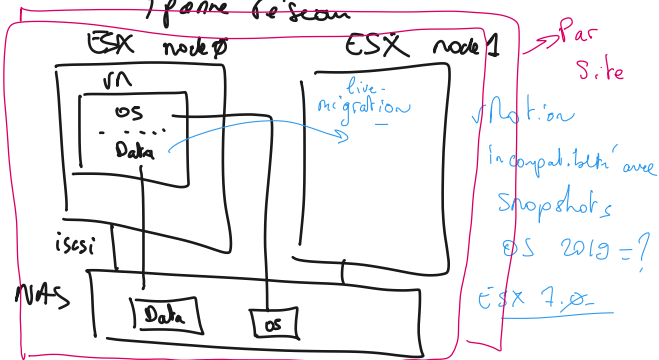
OS → local host  
VM ESX.



FS → autonome OK

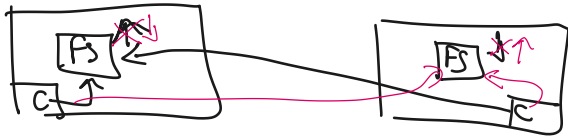
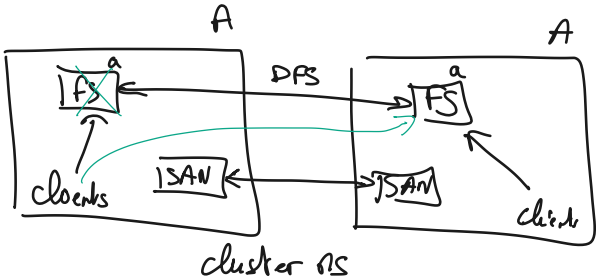
→ rendre Tolérant vis à vis host ESX

~~1 panne de scan~~



DC1

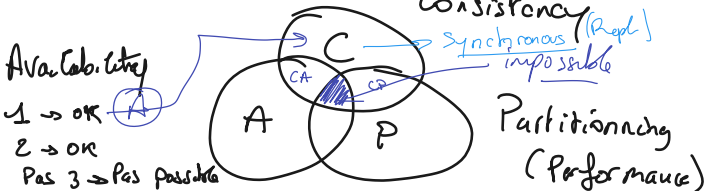
DC2

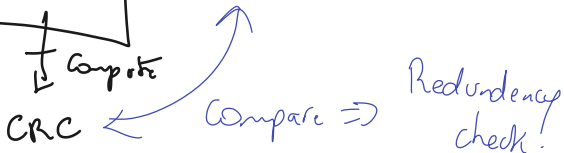
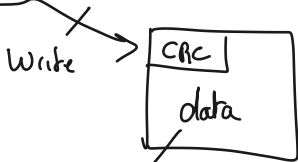
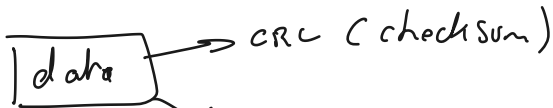


Coherence  $\Leftrightarrow$  disponibilité  
 Synchrones  $\Leftrightarrow$  asynchrone

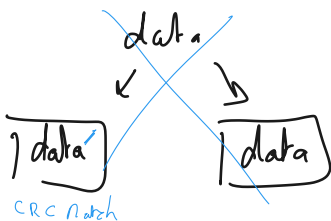
En Base de données

Théorème de CAP





Repair:



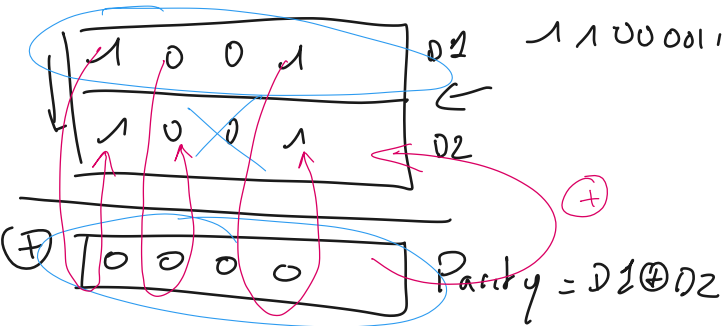
RAID ~~6~~



Q AND  
1 OR

XOR ← ⊕  
⊕ XOR

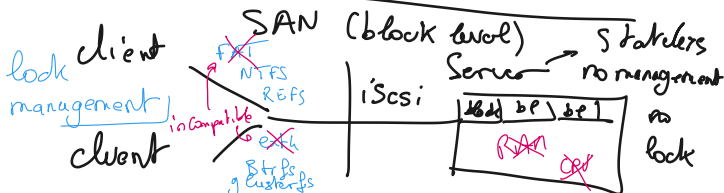
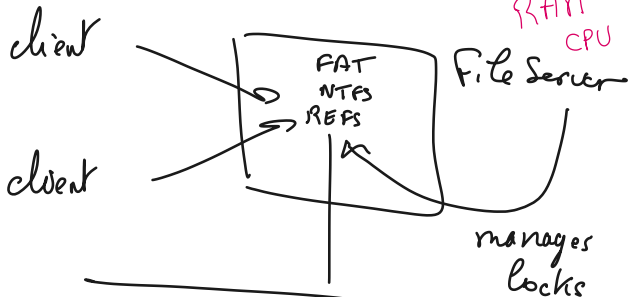
0 ⊕ 0 = 0  
1 ⊕ 1 = 0  
1 ⊕ 0 = 1  
0 ⊕ 1 = 1



SMB/NFS

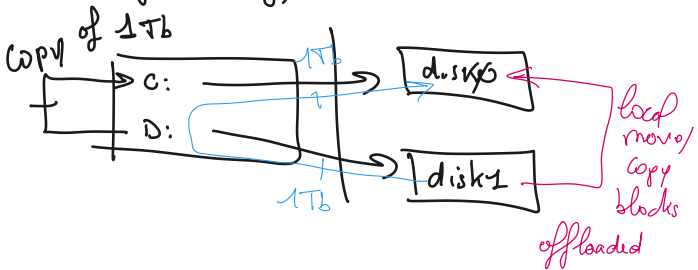
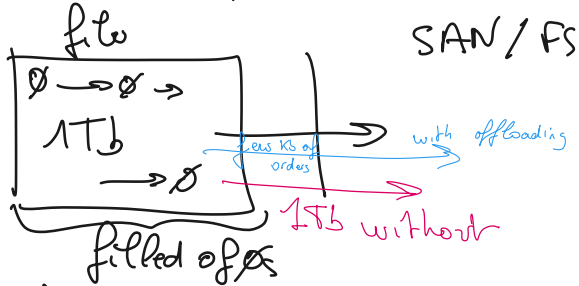
Client / Server

Server Process



# offloading

client (Database)



ESXi vSphere :

Chaque node du cluster Windows ( WSFC ) doit être sur un host unique du cluster VMWare. **only one windows cluster node per vSphere cluster host !**

**NOT Supported by vSphere with WSFC :**

- Live Storage vMotion support
- Fault Tolerance
- N-Port ID Virtualization (NPIV)
- Mixed versions of ESXi hosts in a vSphere cluster

Physical Disk



Partitions / Volumes

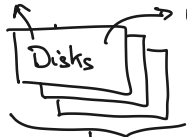


File System / Format  
NTFS / ReFS



Letter

local STORAGE POOL



Storage Pool



Virtual Disk  
- Size - Mirror  
- RAID Scheme - Parity  
- thin / thick



Volumes



Format



Letter

file Server workload  
→ ReFS  
- more resilient  
- less performance

SQL workload  
→ NTFS  
- Resiliency bring Syt  
- performant.

Mise à jour de la conf pour être compatible avec VMWare are les cluster FCI :

Lecteurs de DVD en IDE

Ne mettre que des disques SATA sur les bandes :

SRV1 : 0:0 0:1 0:2

SRV2 : 1:0 1:1 1:2

SRV3 : 2:0 2:1 2:2

SRV4: 3:0 3:1 3:2

Désactiver le firewall sur toutes les machines ( pour SQL Plus tard )

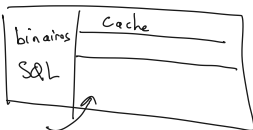
Article gestion des tiers :

<https://charbelnemnom.com/how-to-create-a-multi-resilient-volume-with-refs-on-standalone-server-in-windows-server-2016-ws2016-hyperv-storagespaces/>

Page Spl.      Mémoire

Instance = RAM + Disque

Process



- Common global

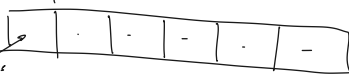
Page (8ko) → Types de Page

NDF → fichier Primaire → - Page de MetaData  
NDF → " (1) Secondaires → - Page de Sommaire  
BDD

Disques

Data  
Types

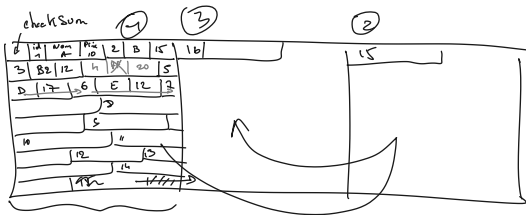
log



statique

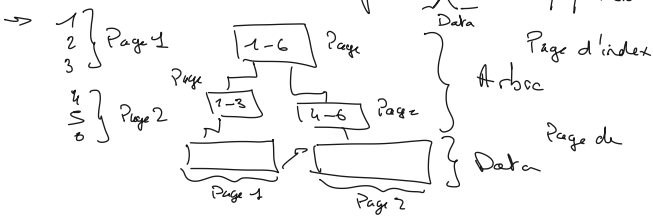


en ajout

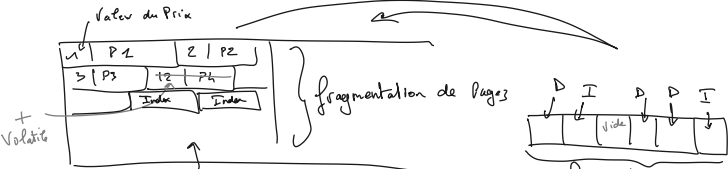


Donnée

Primary Key  $\rightarrow$  Index cluster  $\rightarrow$  Index BTree +  
 feuille Triées Physiquement  
 Data



Index ordre.



Valeur d'index  $\rightarrow$  N° de Page  $\rightarrow$  d'index  
 ou  $\rightarrow$  de donnée } Pointeurs  
 valeur clef primaire

Index  $\rightarrow$  Triés (donnée pas forcément)  
 ex: Prix de Produits





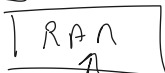
Objectif: - Limiter les IO  
 - Persister / Ne pas perdre de données.

→ Combinaison Data / Log

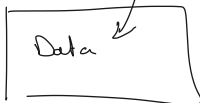
Journal / Log



- Petites écritures  
 append only



R/W volumineux



Atomicity → transaction

Consistency

Isolation

Durability

Disques locaux:

Log → HDD RAID 10

Data → SSD RAID (S/6)

bon en lecture

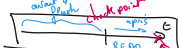
mauvais / Moyen en écriture -

check point

RAN

flush data

logs



Plantage Serveur

Partie des blocs incohérents

de faire les blocs de données en fait à chaque check point



# Importance de temps

→ équivaut au "fichier de Pagination"

→ quand pas assez de RAM

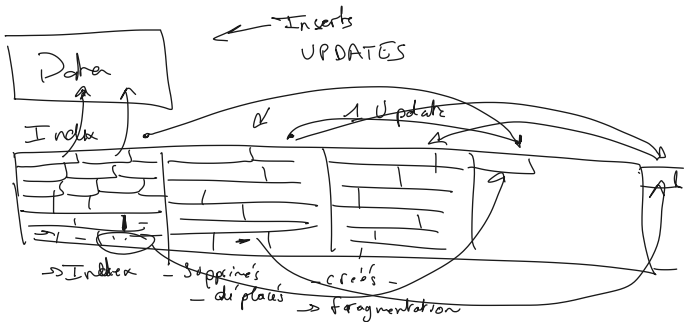
ex1 - Objets temporaires non partagés } en RAM  
create table #ma table

Si volume Temp > RAM → persiste dans temps

ex2:

Requête Tri sans Index sur Table de  
100 Go

lecture → Tri → bufferisation → envoi des lignes  
RAM sinon dans temps



						50%

marge

80% = Consomme 2x plus de payes

33% = " 3x