

Machines virtuelles et systèmes d'exploitation

INF34207 – Séance du
7 février 2024

Lise Boudreault, chargée de cours

Plan du cours:

1. Exercice récapitulatif et solutions (mémoires)
2. Machines virtuelles
3. Les systèmes à base de conteneurs

1. Exercice récapitulatif et solutions (mémoires)

Exercice récapitulatif :

Soit une mémoire centrale de 48 k d'octets adressable par 8 bits, on dispose des puces mémoires suivantes : 8k*16, 32k*4, 8k*8, 8k*2 et 16k*32

1. Proposer une organisation de cette mémoire
2. Déterminer l'adresse de fin et de début de cette mémoire
3. Déterminer l'adresse de début et de fin de chaque zone.

1. Exercice récapitulatif: rappel notions

Bus de données, bus de contrôle et bus d'adresses

- **Le bus de données :** Le bus de données est un bus bidirectionnel, son rôle consiste à transporter les données à lire ou à écrire de la mémoire centrale vers le microprocesseur ou du microprocesseur vers la mémoire centrale. Lors d'une lecture, c'est la mémoire qui envoie un mot sur le bus (le contenu de l'emplacement demandé) ; lors d'une écriture, c'est le processeur qui envoie la donnée. La taille du bus de données est égale au TCM.

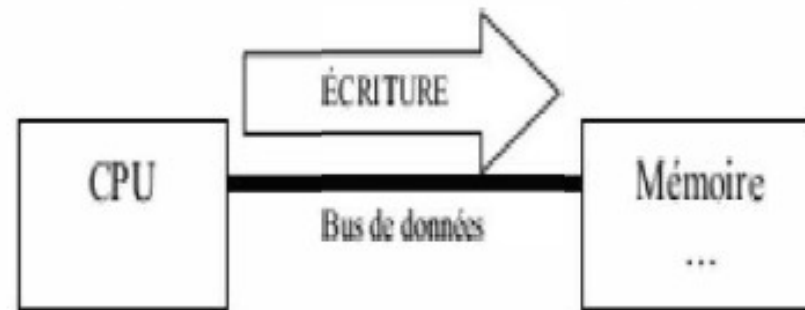
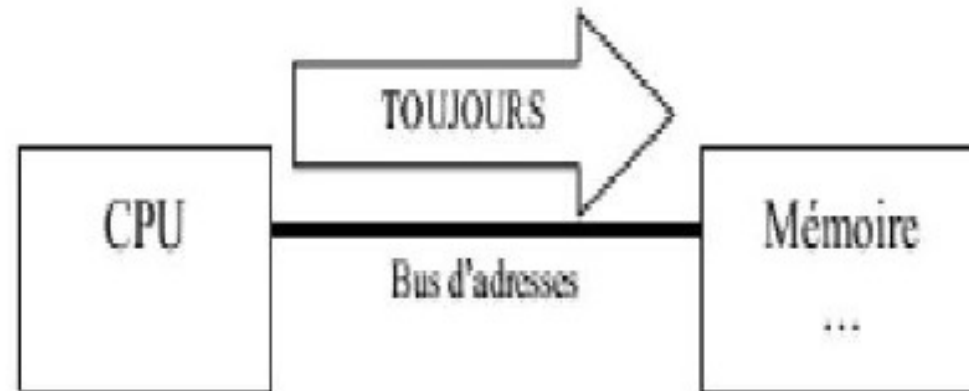


FIGURE 3 OPERATION D'ECRITURE AVEC LE BUS DE DONNEES

1. Exercice récapitulatif: rappel notions

- **Le bus d'adresses** : Le bus d'adresse est un bus unidirectionnel, son rôle consiste à transporter l'adresse de la case mémoire à lire ou à écrire. Seul le processeur envoie des adresses. Il est composé de m fils ; on utilise donc des adresses de m bits. La mémoire peut posséder au maximum 2^m emplacements (adresses 0 à $2^m - 1$).



1. Exercice récapitulatif: rappel notions

L'adresse basse dite aussi adresse de début : représente l'adresse de début de la case numéro 1 elle est représentée en binaire par une suite de m bits (nombre de bits d'adresse) de 0

L'adresse haute ou aussi l'adresse de fin : représente l'adresse de début de la case numéro $N-1$ elle est représentée en binaire par une suite de m bits (nombre de bits d'adresse) de 1

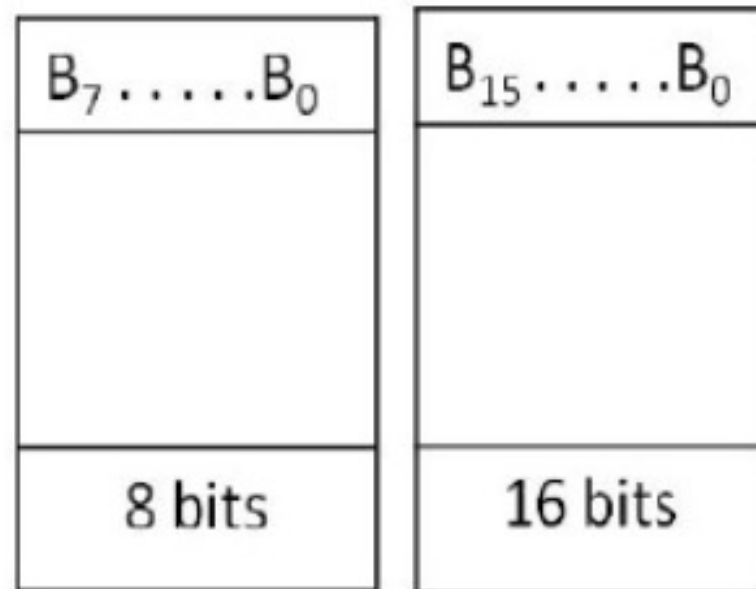
1. Exercice récapitulatif: rappel notions

La mémoire centrale peut être vue comme un large tableau divisé en emplacements (cases) de taille fixe utilisés pour stocker instructions et données. La taille d'un emplacement mémoire noté TCM pourrait être quelconque (toujours multiple de 2) doit être la même pour toutes les cases d'un ordinateur. En fait, la plupart des ordinateurs utilisent des emplacements mémoire d'un octet (1 byte = 8 bits).

Tightly Coupled Memory (TCM) provides low-latency memory accesses that the core can use without the unpredictability of access time that is a feature of caches

1. Exercice récapitulatif: rappel notions

Exemple : une mémoire centrale avec une TCM= 8 bits et une autre avec TCM=16 bits

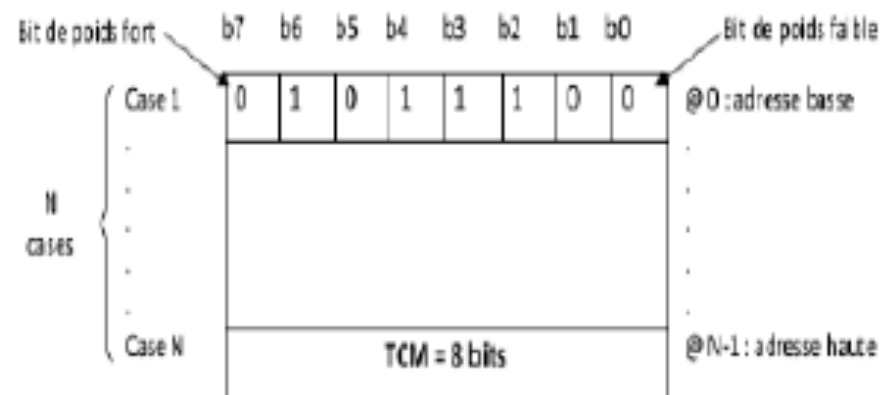


Chaque case mémoire est référencée par un numéro unique appelé: **adresse mémoire** (généralement écrite en hexadécimal). Les adresses sont séquentielles (consécutives).

1. Exercice récapitulatif: rappel notions

Exemple D'UNE STRUCTURE DE MEMOIRE CENTRALE AVEC TCM = 8 BITS

Toutes les adresses des cases mémoires nécessitent le même nombre de bits pour les écrire : m bits, ou m est considéré comme le nombre de bits d'adresse (nombre de lignes d'adresse). Le nombre de bit m dépend du nombre des cases dans la mémoire centrale. Si la mémoire contient N cases alors les adresses de ces cases sont numérotées de 0 à $N-1$.



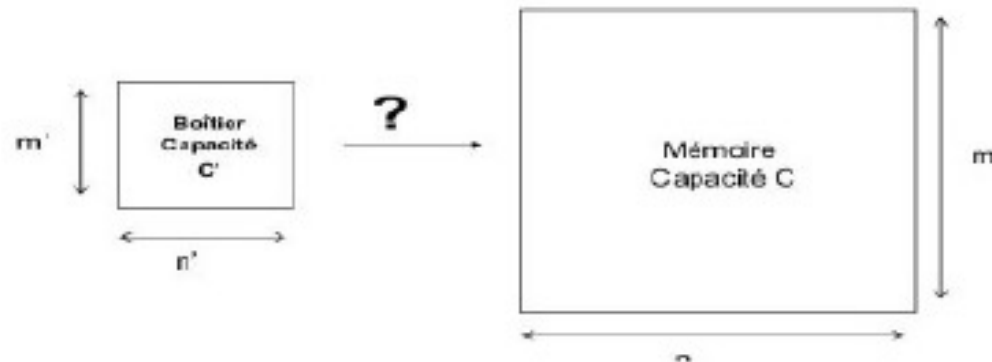
1. Exercice récapitulatif: rappel notions

Conception d'une mémoire centrale

Le problème avec la mémoire centrale c'est qu'on veut réaliser une mémoire de capacité C , mais nous disposons uniquement de boîtiers de taille inférieure.

Les techniques d'intégration ne permettent pas d'obtenir des boîtiers ayant des capacités ou des formats suffisants pour toutes les applications. Il est alors nécessaire d'associer plusieurs boîtiers pour augmenter la longueur des mots ou le nombre de mots. L'association de plusieurs blocs peut permettre d'améliorer les performances temporelles de la mémoire en faisant fonctionner plusieurs blocs en parallèle.

$$\text{Nombre de puces} = \text{Taille mémoire} / \text{Taille puce}$$



1. Exercice récapitulatif: rappel notions

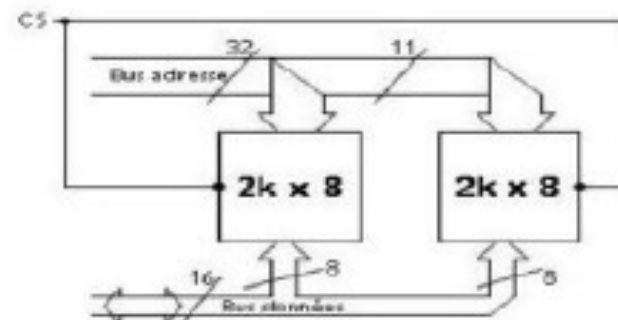
Assemblage Horizontal

(puces en Parallèle placées sur la même ligne → nombre de puces parallèles = nombre de colonnes)
Augmentation de taille case mémoire

On effectue une mise en parallèle des blocs mémoires élémentaire afin d'augmenter la largeur des mots, c'est-à-dire la largeur du bus de données, le bus d'adresse commun aux blocs élémentaire restant inchangé. On pourrait par exemple associer en parallèle 4 blocs mémoires de 8 bits pour former des mots mémoires de 32 bits.

Pour déterminer le nombre de boîtier nécessaire pour obtenir la taille de la case de la mémoire M (extension cases ou extension colonnes) ou les puces placées en parallèles on applique la formule suivante.

$$\text{Nombre de puces parallèles} = TCM / TCP$$



1. Exercice récapitulatif: rappel notions

Assemblage Vertical

(Puces en série placées sur la même colonne \rightarrow nombre de puces séries = nombre de lignes) : Augmentation espace d'adressage

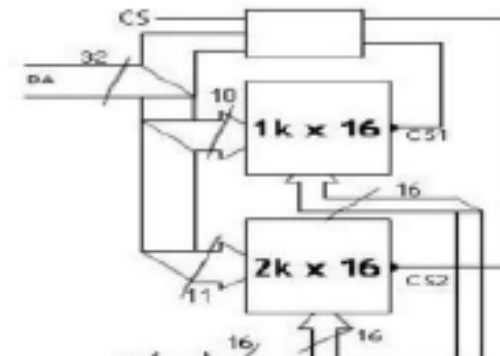
On effectue une association série des blocs mémoires pour augmenter le nombre de mots mémoires, c'est-à-dire la capacité de la zone adressable. Pour ce faire on augmente la largeur du bus d'adresse.

Pour déterminer le nombre de puces en série (extension lignes) nous appliquons la formule suivante :

$$\text{Nombre de puces séries} = \text{Nbr total de puces} / \text{Nbr de puces parallèles}$$

Ou

$$\text{Nombre de puces séries} = \text{espace d'@ Zone} / \text{espace d'@ Zone puce}$$



1. Exercice récapitulatif: rappel notions

Pour une mémoire avec 5 ligne d'adresse

@début=@basse = m '0' = 5 '0' = 000002 = 00/H

@fin=@haute = m '1' = 5 '1' = 111112 = 1F/H

On effet on peut représenter le nombre de cases mémoire par 2^m cases, on dit que le nombre de case mémoire directement adressable est de 2^m (espace d'adressage directe)

1. Exercice récapitulatif: rappel notions

la taille de la mémoire centrale noté **TMC** dépend du l'espace d'adressage et de la taille d'une casse mémoire on obtien donc la formule suivante :

$$\text{Taille de la mémoire centrale} = \text{espace d'adressage} \times \text{TCM}$$

$$\text{TMC} = 2^m \times \text{TCM}$$

La taille de la mémoire centrale est généralement exprimée en kilo ø, Méga ø, Giga ø, rappelons ces mesures

| | |
|------------|---------------------------------------|
| 1 K (Kilo) | $2^{10} = 1024$ bites |
| 1 M (Mega) | $2^{20} = 1048\ 576$ bits |
| 1 G (Giga) | $2^{30} = 1\ 073\ 741\ 824$ bits |
| 1 T (Téra) | $2^{40} = 1\ 099\ 511\ 627\ 776$ bits |

1. Exercice récapitulatif et solutions

$$1. \text{TMC} = 48\text{k} \varnothing$$

$$\text{TCM} = 1 \varnothing$$

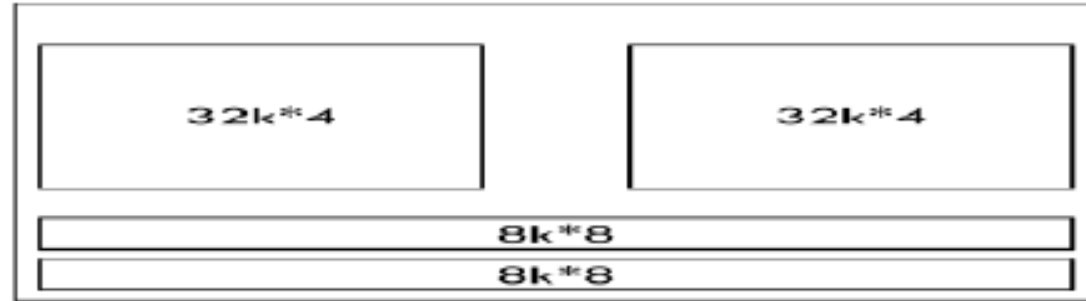
D'où nombre de cases = 48 k cases ; $2^{15} < 48\text{ k} < 2^{16}$ d'où $m = 16$ lignes d'adresse Plusieurs organisations sont possibles sauf que les puces $8\text{k} * 16$ et $16\text{k} * 32$ ne peuvent pas être utilisées car $\text{TCM} < \text{TCP}$

On peut travailler avec l'espace d'adressage

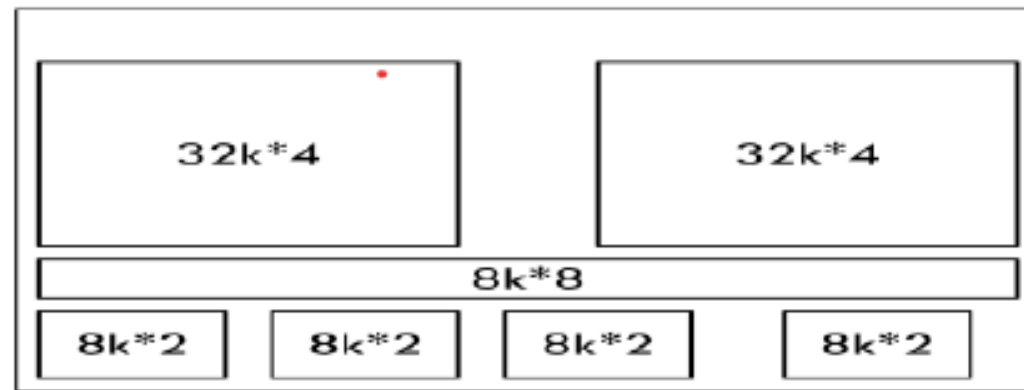
48k/ 32k on trouve une ligne de puces de type $32\text{k} * 4$ en série et il reste 16k de cases de l'espace d'adressage qu'on peut soit occuper par 2 lignes de puces $8\text{k} * 8$ ou $8\text{k} * 2$ ou un mélange des deux,

1.1 Proposer une organisation de la mémoire

Organisation 1



Organisation 2



1.2 Déterminer l'adresse de début et de fin de cette mémoire

2. On va considérer la deuxième organisation

@début MC = 16 '0' = 0000

@fin MC = 16 '1' = FFFF

1.3 Déterminer l'adresse de début et de fin de chaque zone

3. Zone1

@début = @début MC = 0000

$m_{Z1} = ?$ N= 32k cases d'où $N = 2^{15}$ d'où $m_{Z1} = 15$

@fin = @début + 15 '1' 0000+ 7FFF = 7FFF

Zone 2

@début = @fin Z1 +1 = 7FFF+1 = 8000

$m_{Z2} = ?$ N= 8k cases d'où $N = 2^{13}$ d'où $m_{Z2} = 13$

@fin = @début + 13 '1' 8000+ 1FFF = 9FFF

Zone 3

@début = @fin Z2 +1 = 9FFF+1 = A000

$m_{Z3} = ?$ N= 8k cases d'où $N = 2^{13}$ d'où $m_{Z3} = 13$

@fin = @début + 13 '1' A000+ 1FFF = BFFF

Zone Vide

@début = @fin Z3 +1 = BFFF+1 = C000

@fin = @fin MC = FFFF

2. Machines virtuelles et processeurs

Pourquoi les processeurs à plusieurs cœurs ?

Pour répondre à des besoins variés:

- Naviguer et accéder à plusieurs applications sur Internet
- Jouer à des jeux vidéo (gaming) en réseau;
- Produire des montages vidéo;
- Développer des applications énergivore, en réseau pair-à-pair (peer-to-peer) comme le minage d'une cryptomonnaie pour la validation d'une transaction et son écriture dans un grand livre « comptable » (blockchain)

2. Machines virtuelles et processus



Processeurs Intel Xeon W-3175X et AMD Ryzen Threadripper à plusieurs « cœurs » (core) pour des applications hautement « threadées » et gourmandes en ressources informatiques

2. Les processeurs à plusieurs coeurs

les processeurs à plusieurs « cœurs » d'exécution ont permis de créer des environnements de virtualisation

La compétition est féroce entre fabricants de processeurs:

Processeurs

Intel Xeon W-3175X (2019-2022)

AMD Ryzen Threadripper (2018-....)

En 2024 ?

2. Qu'est-ce que le threading?

Un ***thread*** ou **fil** (traduction normalisés par [ISO/CEI 2382-7:2000](#)¹ (autres appellations connues : **processus léger**, **fil d'exécution**, **fil d'instruction**, **processus allégé**, **filet d'exécution**², **exétron**, **tâche**, voire **unité d'exécution**³ ou **unité de traitement**^{4,5}[\[réf. nécessaire\]](#)) est similaire à un [processus](#) car tous deux représentent l'exécution d'un ensemble d'[instructions](#) du [langage machine](#) d'un [processeur](#).

Source: wikipedia, [https://fr.wikipedia.org/wiki/Thread_\(informatique\)](https://fr.wikipedia.org/wiki/Thread_(informatique))

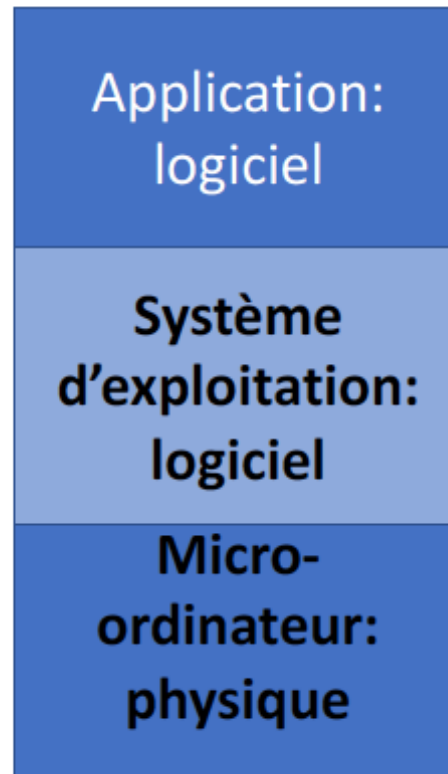
2. Qu'est-ce que le threading?

Du point de vue de l'utilisateur, ces exécutions semblent se dérouler en parallèle. Toutefois, là où chaque processus possède sa propre mémoire virtuelle, les *threads* d'un même processus se partagent sa mémoire virtuelle. En revanche, tous les *threads* possèdent leur propre pile d'exécution.

Source: wikipedia, [https://fr.wikipedia.org/wiki/Thread \(informatique\)](https://fr.wikipedia.org/wiki/Thread_(informatique))

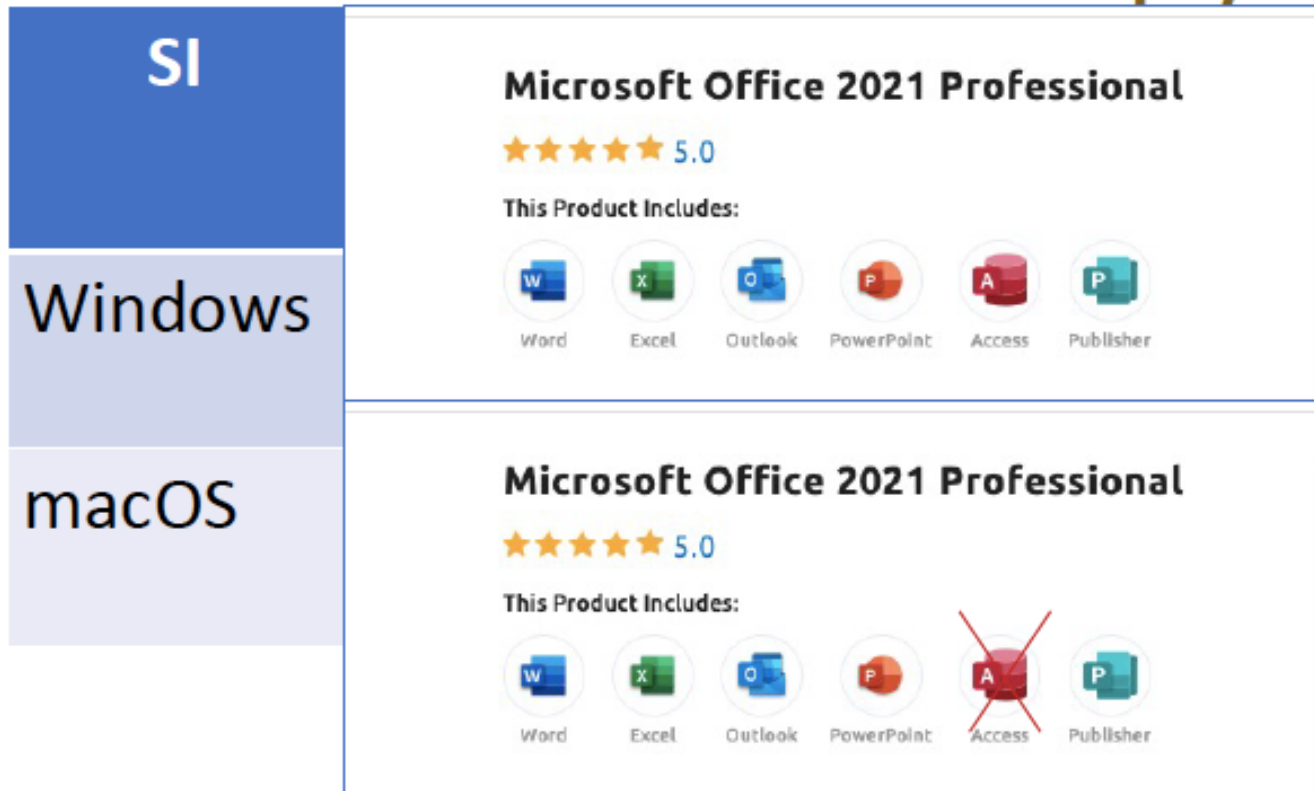
2. AVANT les machines virtuelles

Avant les processeurs à plusieurs « coeurs »: les composants d'un système « traditionnel »



2. AVANT les machines virtuelles

Environnement traditionnel: un système d'exploitation (SI) permet à des logiciels de fonctionner dans un environnement « physique »



2. AVEC les machines virtuelles

Pourquoi la virtualisation?

Pour permettre à des systèmes d'exploitation de cohabiter ...

...et aux applications prévues pour des systèmes d'exploitation différents (ex: Windows, macOS, Linux) de fonctionner sur le même ordinateur

2. Machines virtuelles (MV/VM)

Machines virtuelles (MV)

Parallels Desktop, VMWare, VirtualBox sont des logiciels qui permettent de créer une machine virtuelle (MV) sur un ordinateur Apple. La machine virtuelle fonctionne comme une application Mac, elle est contrôlée par le système d'exploitation macOS.

2. La virtualisation

Machines virtuelles (MV)

La machine virtuelle permet d'installer/exécuter le système d'exploitation Windows en même temps qu'elle permet aux applications Mac de fonctionner, comme Apple Mail et Safari.

2. La virtualisation

Machines virtuelles : petite histoire

- Concept de la machine virtuelle créé par IBM 1960s
- OBJECTIF:
 - optimiser l'utilisation d'un ordinateur
 - éviter la sous-consommation de temps machine
- 1998: VMware (ouvre la voie au concept de middleware)
- 1999: VMware workstation 1.0
- 2008: Microsoft avec Hyper V
- **En 2022**, omniprésentes dans les systèmes d'information
- Les architectures technologiques qui évoluent très vite

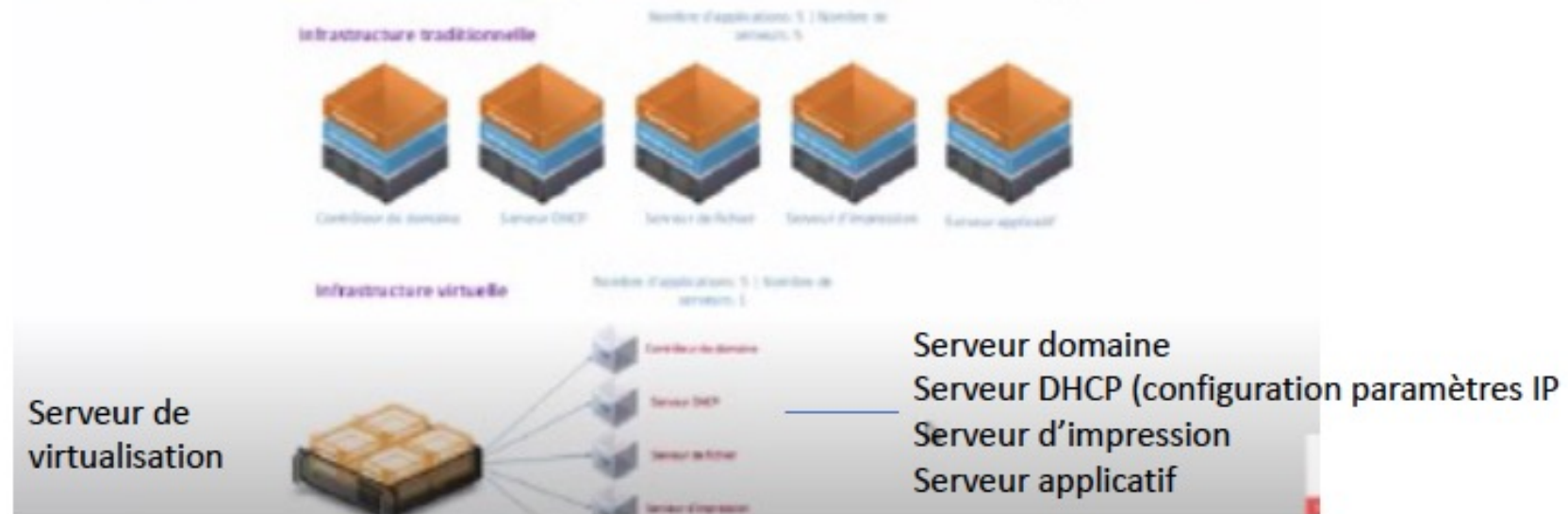
2. La virtualisation

En 2024 ?

2. La virtualisation

Machines virtuelles : avantages

- Consolidation des serveurs: (équation un serveur = plusieurs applications)

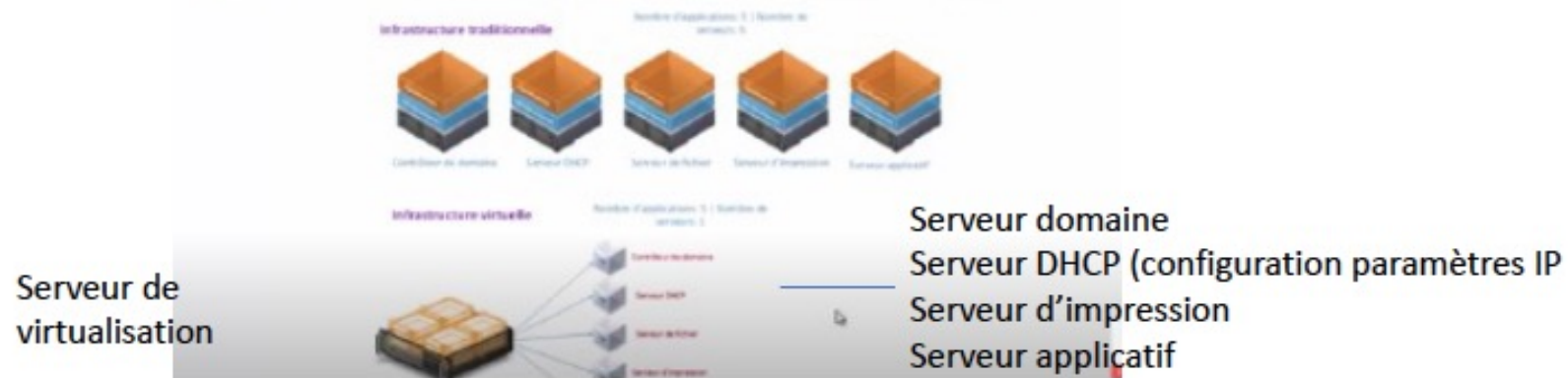


2. La virtualisation

Machine virtuelle : principe de base

Permettre à plusieurs serveurs de cohabiter sur une machine

- Consolidation des serveurs: (équation un serveur = plusieurs applications)



2. La virtualisation

Machines virtuelles : avantages

- Plusieurs machines virtuelles peuvent cohabiter
- Gain en énergie, par exemple, 1 serveur plutôt que 8
- Coûts réduits en équipements
- Optimisation de l'utilisation des ressources du processeur
- La récupération par les copies de sécurité est simplifiée
- Flexibilité des opérations sur une seule machine
- Facilite l'administration centralisée plutôt que décentralisée

2. La virtualisation

Machines virtuelles : inconvénients

- Les coûts sont élevés
- Attendre le retour sur investissement sur le long terme pour atteindre un seuil de rentabilité
- Performance diminuée lorsqu'il y a plusieurs bases de données
- Technologies complexes qui nécessite une expertise en TI
- Organisation du travail: besoins importants de formation continue car les technologies évoluent rapidement

3. La virtualisation et les systèmes à base de conteneurs

La virtualisation permet de répliquer tout un centre de données (datacenter) en faisant fie de l'infrastructure matérielle cible. La problématique de l'administration n'est plus de fournir un environnement cible identique à l'environnement source mais de mettre à disposition des ressources (CPU, RAM, Stockage) au moins équivalentes.

Source: <https://www.orange-business.com/fr/blogs/cloud-computing/reflexions/les-apports-des-technologies-de-la-virtualisation-pour-la-mise-en-oeuvre-dun-pra>

3. La virtualisation et les systèmes à base de conteneurs

Dans une infrastructure informatique décentralisée, la virtualisation permet de reconstruire les différents sites sur un seul site en ne se préoccupant pas d'avoir les mêmes équipements entre les sites sources et le site de récupération en cas de désastre ou d'un bri majeur.

3. La virtualisation et les systèmes à base de conteneurs

L'infrastructure secondaire pour la récupération risque d'être sous utilisée.

D'une part la virtualisation permet de ne pas mettre en œuvre le même nombre de serveurs que le site nominal en consolidant n machines virtuelles sur un seul serveur physique.

D'autre part, l'infrastructure de virtualisation du site de repli peut être utilisée pour exécuter des serveurs ou des services non critiques (machines de développement, laboratoire, environnement de recette, etc.). Ces ressources non critiques peuvent être libérées lors de l'activation du PRA.

3. La virtualisation et les systèmes à base de conteneurs

- **PRA** : Plan de reprise d'activité
- Un plan de reprise d'activité est un document qui permet à une entreprise de prévoir, par anticipation, les démarches à entreprendre pour reconstruire et remettre en route un système **informatique** en cas de sinistre important du centre **informatique**.
- Source: <https://www.journaldunet.fr/business/dictionnaire-economique-et-financier>

3. La virtualisation et les systèmes à base de conteneurs

- 1. Les processeurs virtuels dans un processeur Intel
- 2. Les processeurs AMD Ryzen Threadripper
- 3. La virtualisation du MAC OS:
 - Parallels Desktop VMWare Fusion
- 4. Système à base de conteneurs: Docker



Je vous remercie de votre attention

Rappel: Travail #1 remise 14 février 23h59

Préparation pour la semaine prochaine:
Révision des notions vues aux séances précédentes

Bonne semaine!