Virtualization level 2

Bonne gestion d’une architecture virtualisée avec ESXi

1 : CPU

Quelle est la problématique ?

- Avantages de la virtualisation (consolidation des serveurs) :

  Economie du matériel, de l'énergie, des surfaces, …

  Souplesse (ajouter, tester, supprimer, …) d'exploitation

  Compatibilité avec serveurs existants (Windows, Linux, Solaris, …)

- Challenges

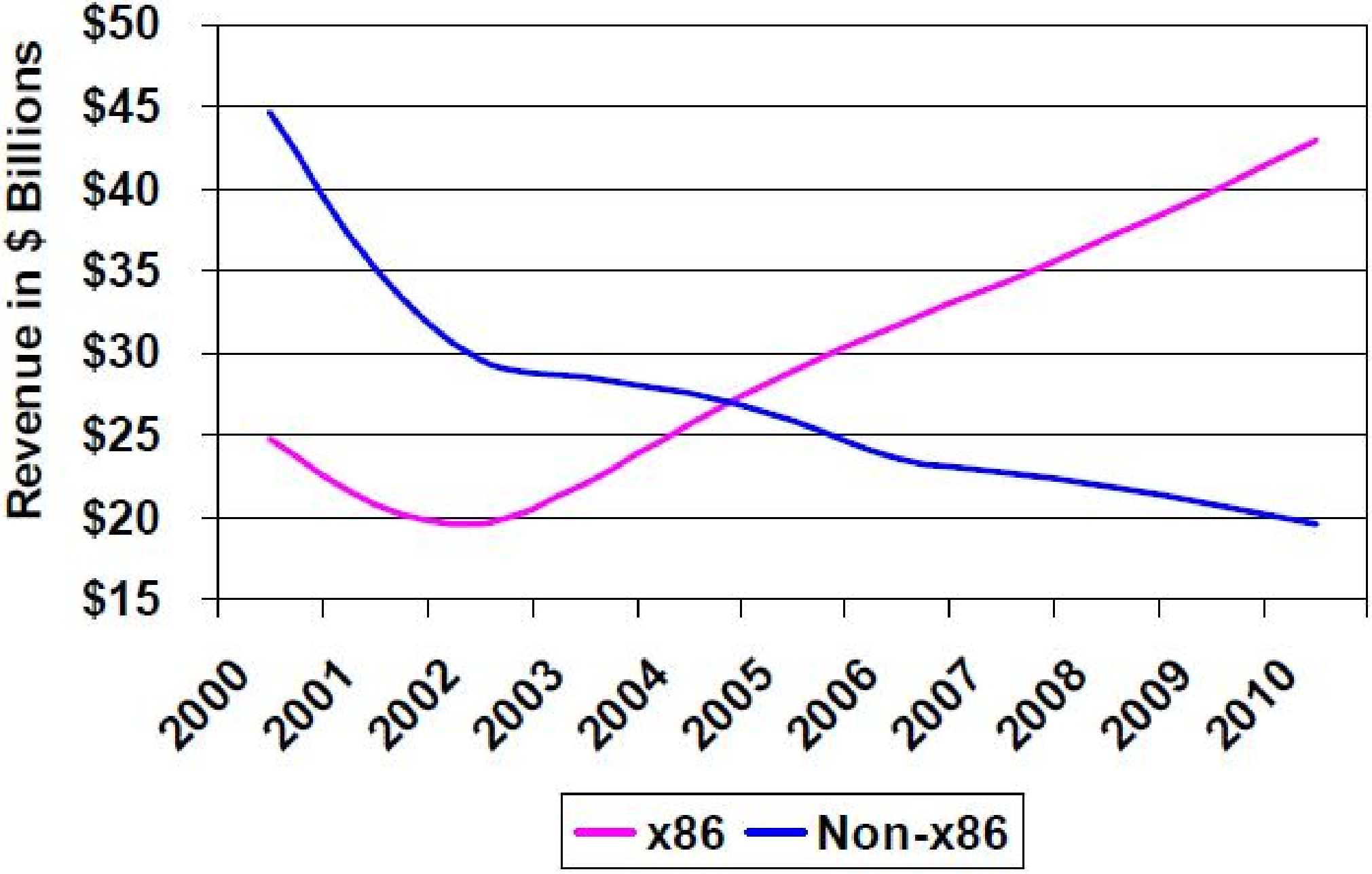
  Trouver le meilleur compromis entre coût et performances : charge CPUs, optimisation de l'occupation RAM, choix du système de stockage, prise en compte des exigences de sécurité

- Inconvénients

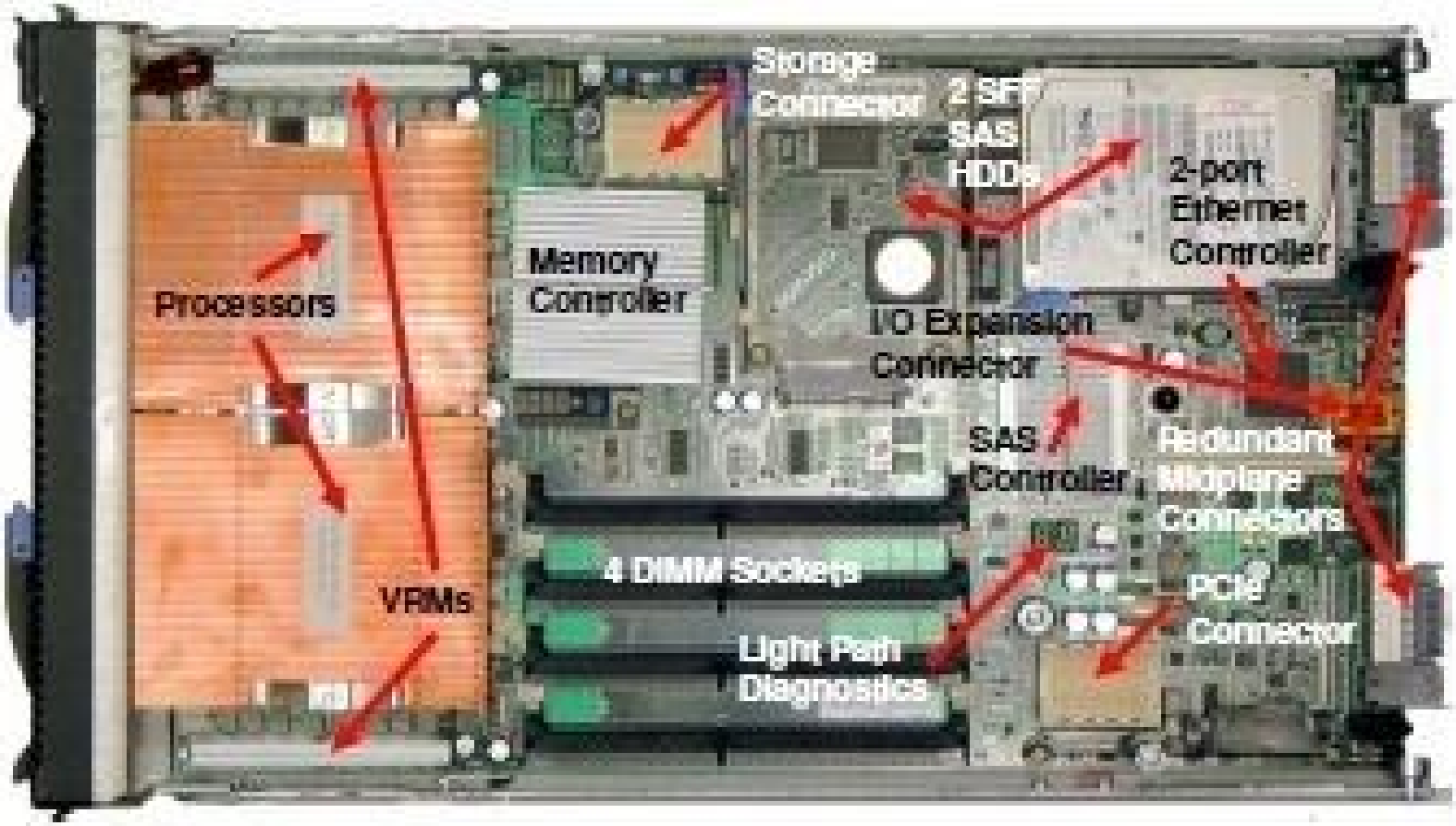
  Difficulté d'agir lors d'un problème (affirmation VMware)

  Niveau de sécurité (disponibilité, isolation entre VMs, …) à valider

x86 servers the fastest growing (source IBM)



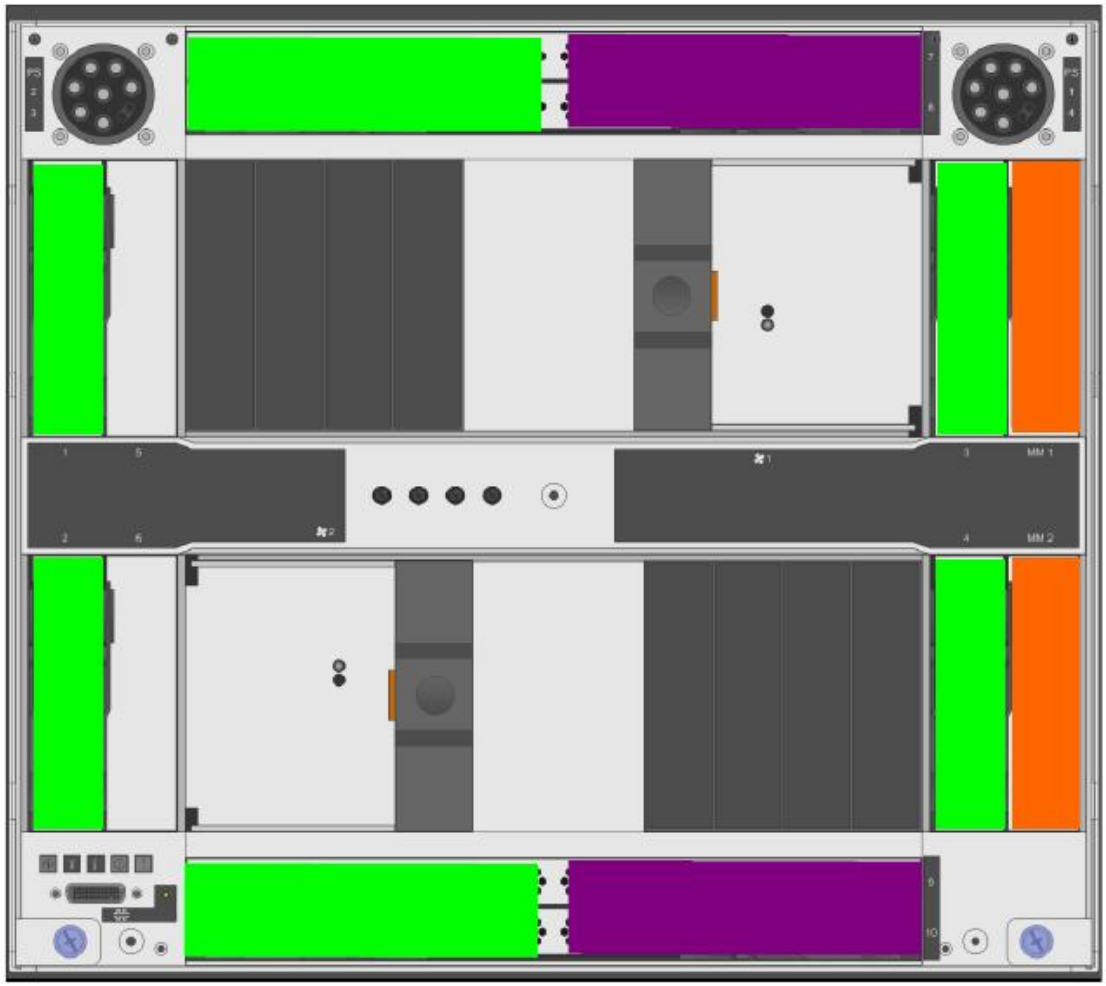
IBM Blade HS21



- CPU = 2 x Intel Xeon CPU E5430 @ 2.66GHz → 8 cores
- RAM = 32 GB
- No Hard Disk
- 6 ports Ethernet 1 Gbit/s
- 2 ports FC *(Fibre Channel)* 4 Gbit/s

IBM Blade Center H (châssis)

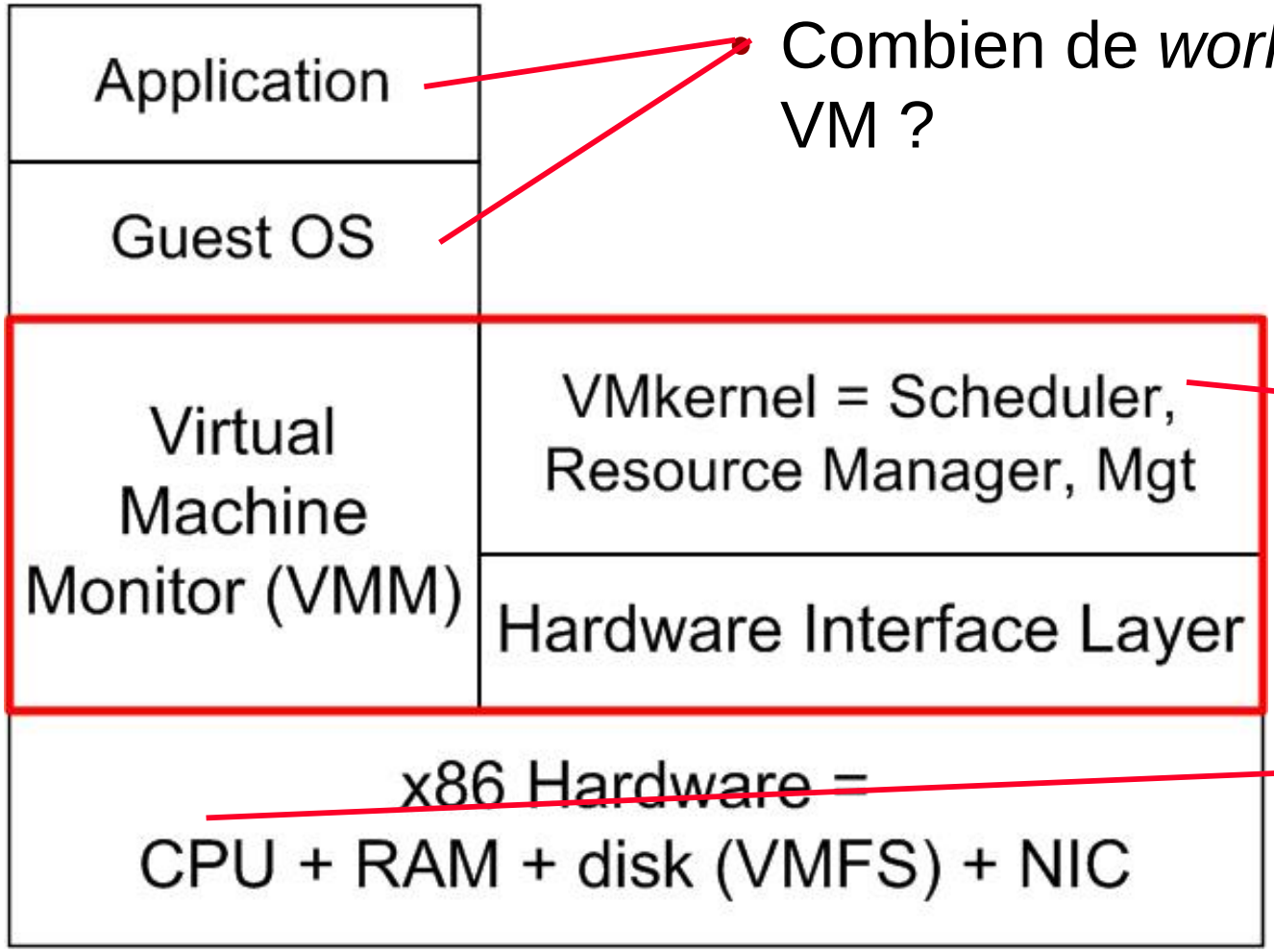
- 14 lames *(blade)* au max



- 6 commutateurs Ethernet (vert)
- 2 *fabrics Fibre Channel* (violet)
- 2 module de management (orange)

- http://www.lenovo.com/transactions/pdf/lenovo-ibm-x86-acquisition-ppt.pdf

Architecture ESXi : scheduler, worlds, pCPU ?



- Combien de *worlds* sont nécessaires par VM ?
- Quels sont les *worlds* ?
  Schedulable entities
- Comment se fait le lien avec le(les) pCPUs ?
- pCPU = physical CPU

Esxtop répond aux questions précédentes

```
 6:03:39pm up  2:13, 152 worlds; CPU load average: 0.51, 0.51, 0.51
PCPU USED(%):   1.1 100.0 AVG:  50.5
PCPU UTIL(%):   1.1 100.0 AVG:  50.6

    ID    GID NAME             NWLD   %USED    %RUN    %SYS   %WAIT    %RDY   %IDLE
 10136   2652 vmx                1    0.02    0.02    0.00   99.93    0.01    0.00
 10138   2652 mks:Nostalgia      1    0.10    0.09    0.00   99.79    0.07    0.00
 10139   2652 vcpu-0:Nostalgi    1   99.92   99.94    0.00    0.00    0.01    0.00
```

- PC Asus comprenant 2 pCPU

- Option V → view only VM + option e → expand

  3 *worlds* pour Nostalgia

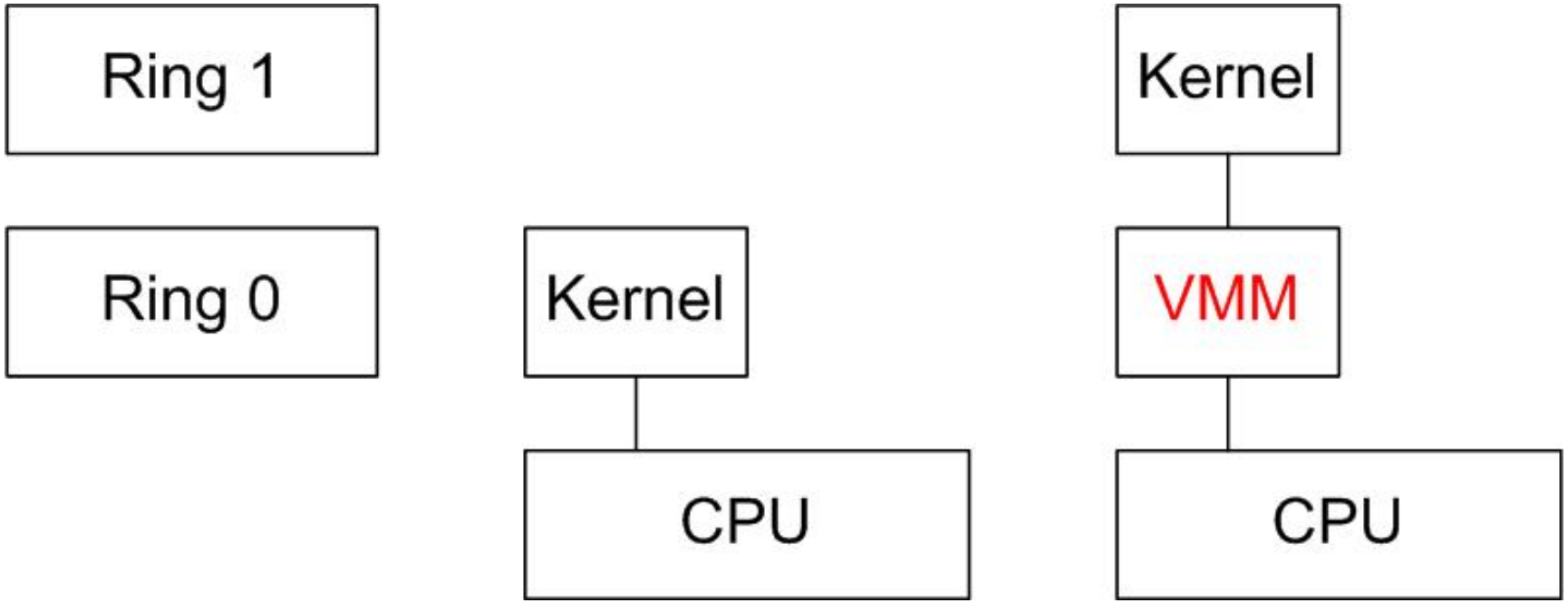
  vcpu-0:Nostalgia → processus associé à la ressource vCPU

  mks:Nostalgia → MKS assists mouse/keyboard/screen virtualization

  vmx → semble correspondre à VMM

Virtual Machine Monitor

- Code qui virtualise tout le matériel : CPU, RAM, …
- Garantit l'isolation entre les VMs
- Intercepte *(trap & emulate, hw assist)* les instructions privilégiées



Esxtop : comparaison Nostalgia – ubuntu (2 vCPU)

ID	GID	NAME	NWLD	%USED	%RUN	%SYS	%WAIT	%RDY	%IDLE
10136	2652	vmx	1	0.02	0.02	0.00	99.93	0.01	0.00
10138	2652	mks:Nostalgia	1	0.10	0.09	0.00	99.79	0.07	0.00
10139	2652	vcpu-0:Nostalgi	1	99.92	99.94	0.00	0.00	0.01	0.00
19118	8707	vmx	1	0.02	0.02	0.00	99.95	0.00	0.00
19174	8707	mks:ubuntu_serv	1	0.08	0.08	0.00	99.83	0.07	0.00
19175	8707	vcpu-0:ubuntu_s	1	0.11	0.11	0.00	99.75	0.12	99.66
19176	8707	vcpu-1:ubuntu_s	1	0.02	0.02	0.00	96.18	3.78	96.08

- GID 2652 (3 *worlds*) correspond à Nostalgia
- GID 8707 (4 worlds) correspond à ubuntu
  nb de *worlds* vCPU = nb de vCPU

- Nostalgia consomme toutes les ressources mises à disposition
  vCPU-0 = 99% Used

- http://communities.vmware.com/docs/DOC-9279
  Excellent document qui décrit chaque compteur

Idle

- Sous Linux et Windows, un *thread* (processus) en cours d'exécution (état = RUNNING) va utiliser l'instruction HLT *(halt)* pour terminer son exécution et passer dans l'état WAITING

  http://en.wikipedia.org/wiki/HLT

- Cette instruction met en veille le CPU (qui ne fait plus rien → économie d'énergie) jusqu'à la prochaine interruption

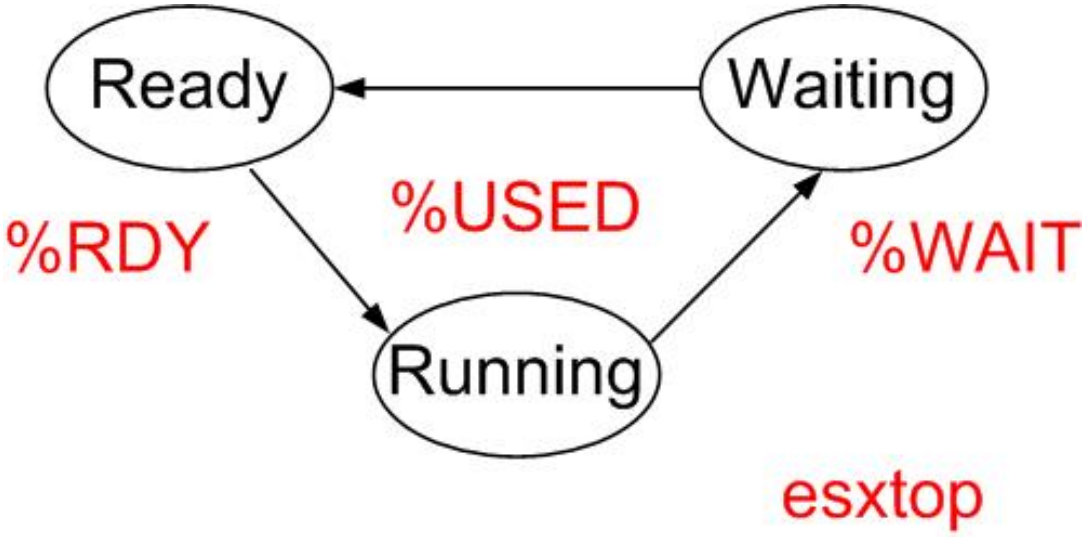
- L'ordonnanceur *(scheduler)* est réveillé périodiquement (100 Hz par exemple)
- Le processus *Idle* est exécuté lorsque tous les autres processus sont dans l'état WAITING

  http://en.wikipedia.org/wiki/System_Idle_Process

Scheduler

- Périodiquement, l'ordonnanceur *(scheduler)* doit allouer une ressource pCPU au processus (VM - vCPU) qui le demande

- Chaque processus (VM) se trouve dans 1 des états suivants



- Running (%USED) → la VM est en cours d'exécution
- Waiting (%WAIT) → la VM attend un événement (ou "exécute" idle)
- Ready (%RDY) → la VM attend la ressource CPU

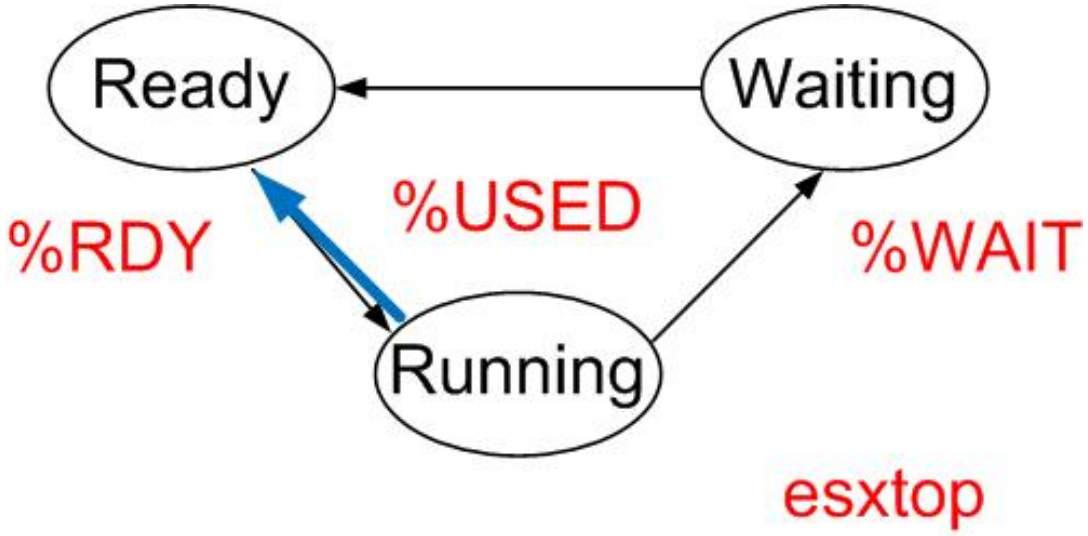
ID	GID	NAME	NWLD	%USED	%RUN	%SYS	%WAIT	%RDY	%IDLE
7838	7838	XP	3	0.55	0.72	0.00	299.29	0.03	99.47

Charge utile du(des) pCPU

- Quel est le **niveau de charge utile des pCPU ?**

- Y a-t-il contention (%RDY > 5%) ?

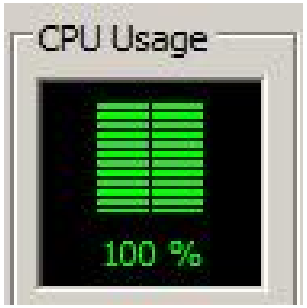
  Σ CPU_VMi > ressources disponibles = 6 GHz (2 CPU de 3 GHz)

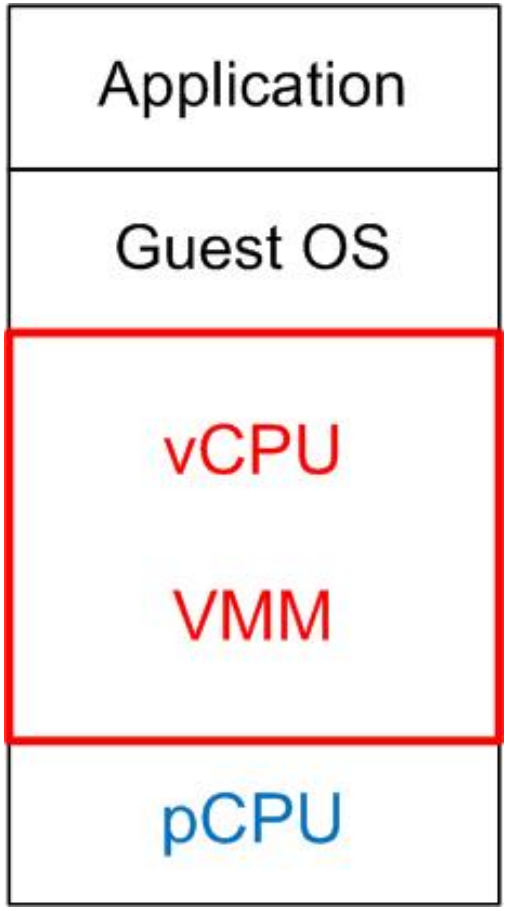


- Idle renseigne sur le taux de charge des pCPU

  → scénario avec 3 VM Nostalgia capable de consommer 6 GHz

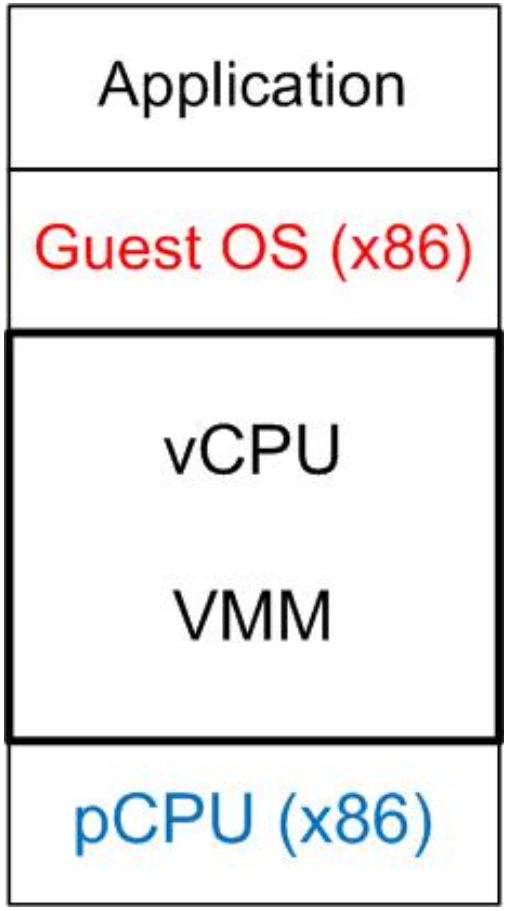
pCPU à 20% & vCPU à 100% (§1) ?





- Application *Password Cracking* qui charge le CPU au maximum
- Guest OS (XP) croit utiliser pCPU; il voit en fait vCPU
- VMM *(Virtual Machine Monitor)* présente à l'OS vCPU = une virtualisation du CPU
- Limitation à 20%
- pCPU chargé à 20%

Virtualisation & Emulation



- La virtualisation exige une couche VMM *(Virtual Machine Monitor)* capable d'émuler le matériel
- Pour rester **performante**, cette émulation impose des versions binaires des *Guest OS (Linux, Windows, …)* **compatibles** avec pCPU (x86)
- La problématique est donc différente des produits comme Bochs (http://bochs.sourceforge.net/) = *cross platform IA-32 emulator* ou Qemu (http://www.qemu.org/)

Labo level 2 §1-2 : Process, CPU (45 min)

1 Processus affichés (15') par esxtop

VM Nostalgia et ubuntu_server

2 Charges CPU (20') mesurées avec esxtop, vSphere, Task Manager

John The Ripper pour cracker un mot de passe → VM XP_T_Crack

Limiter la charge CPU

Affichages vSphere

- Temps réel avec intervalle de 20 s

	Rollup	Units	
• **CPU**			
Usage	Average	Percent	→ moyenne en %
Usage in MHz	Average	MHz	→ moyenne en MHz
Idle	Summation	Millisecond	→ somme en ms
• **Memory**			
Consumed	Average	Kilobytes	→ moyenne en Kbyte
• **Values**	Latest	Maximum	Minimum Average

Virtualization level 2

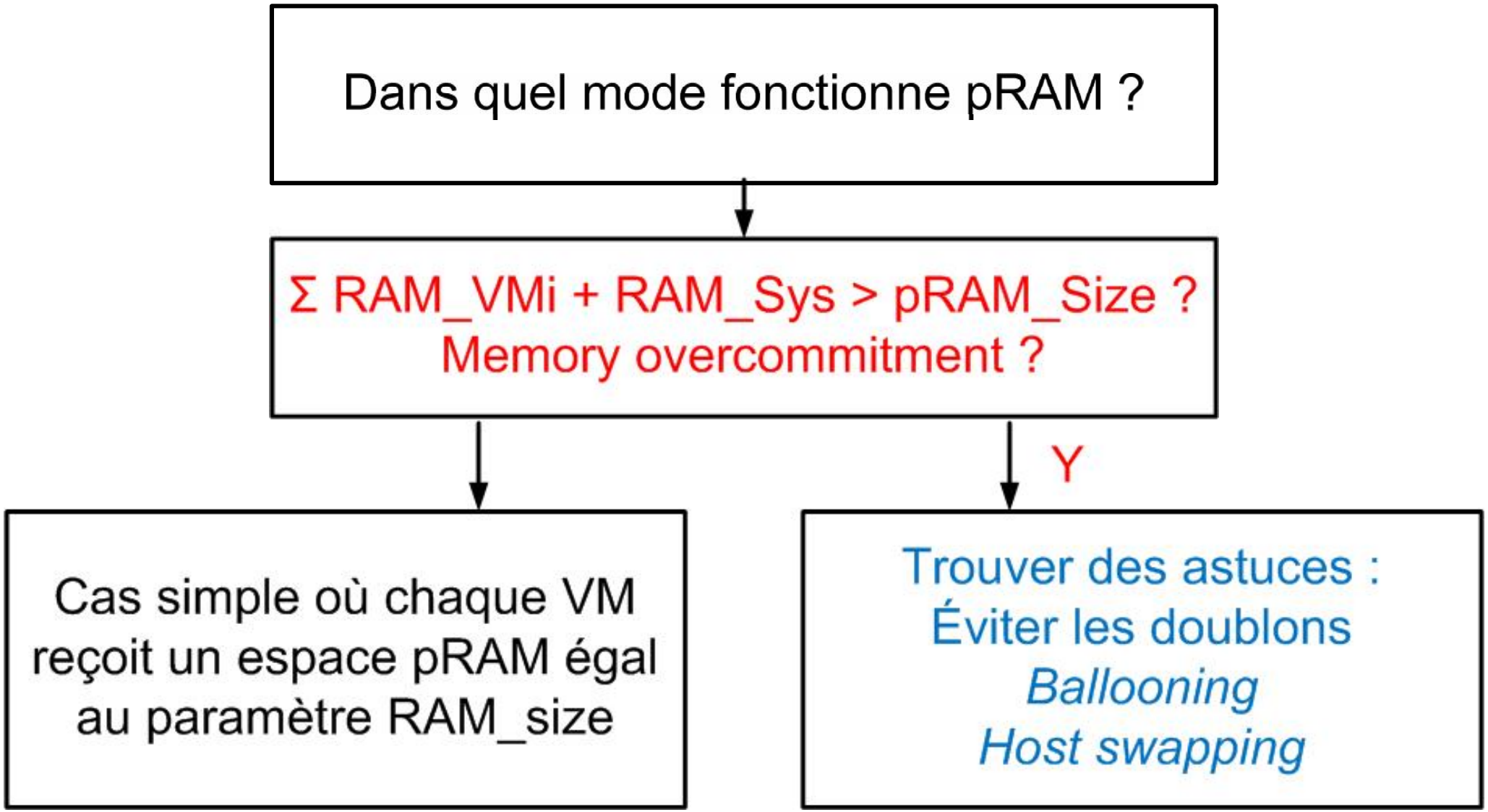
Bonne gestion d’une architecture virtualisée avec ESXi

2 : RAM – VMware Tools

Synthèse du labo précédent

- Combien de worlds (processus) par VM ? Utilité ?
- Comment la VM XP_T_Crack se comporte-t-elle au niveau charge CPU ?
- Y a-t-il un effet en limitant à 20% des ressources CPU ?
- Comment l'ordonnanceur (scheduler) utilise-t-il les 2 cœurs ?
- Quel est le time slice ? Voir question 2c

Modes de fonctionnement de la mémoire



Consolider le plus de serveurs

Occupation RAM – Bilan mémoire

- Comment dimensionner pRAM ?
- Quel est l’espace mémoire occupé par ESXi ?
- Quel est l’espace mémoire occupé par chaque VM ?
- Différences entre vRAM et pRAM ?
- Quels sont les outils au niveau hyperviseur ESXi ?
- Quels sont les outils au niveau VM ?
- Méthodologie de mesure

top & htop

```
top - 12:03:33 up 20 min,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  59 total,   1 running,  58 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  2.6%sy,  0.0%ni, 97.4%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    509264k total,    55188k used,   454076k free,    10580k buffers
Swap:   397304k total,        0k used,   397304k free,    28820k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
  862 labotd    20   0  2416 1144  920 R  2.6  0.2   0:00.64 top
```

```
CPU[|                                  1.3%]     Tasks: 20 total, 1 running
Mem[|||                              15/497MB]     Load average: 0.00 0.05 0.02
Swp[                                  0/387MB]     Uptime: 00:37:28

  PID USER      PRI  NI  VIRT   RES   SHR S CPU%  MEM%   TIME+  Command
  870 labotd     20   0  2588  1224   992 R  0.0   0.2  0:00.44 htop
    1 root       20   0  2760  1624  1212 S  0.0   0.3  0:00.22 /sbin/init
```

Linux – free

```
eig@ubuntu-server:~$ free -t
             total       used       free     shared    buffers     cached
Mem:        515440      34932     480508          0       2860      14284
-/+ buffers/cache:      17788     497652
Swap:       409616          0     409616
Total:      925056      34932     890124
```

Affiche (niveau *kernel*) en kilobytes (-m pour MByte) :

- **free** espace mémoire libre
- used = buffers + cached + buffers/cache used → espace utilisé
- total = used + free

- Utilisation du *swap*

- la colonne shared est obsolète

Linux (sans/avec virtualisation)

- Ubuntu Server 8.04.3 LTS 32bits

- Résultats de la commande free

après *Boot*	total (MB)	used (MB)	free (MB)
Physique (réfence)	2065	44.5	2021
VM	2076	35.8	2040

après 1 jour	total (MB)	used (MB)	free (MB)
Physique (réfence)	2065	133	1932
VM	2076	128	1947

Paramètres mémoire selon esxtop

```
 7:13:21am up  3:30, 143 worlds; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM  /MB:  4081   total:   612     vmk,   223 other,   3245 free
VMKMEM/MB:  3779 managed:   226 minfree,   524 rsvd,   2920 ursvd,  high state
PSHARE/MB:  1280  shared,     2  common:  1278 saving
SWAP  /MB:     0    curr,     0  target:                  0.00 r/s,   0.00 w/s
MEMCTL/MB:     0    curr,     0  target,   139 max
```

7:13:21am → heure système

3:30 → up time

PMEM /MB: 4081 total → espace RAM vu par ESXi

612 vmk → espace utilisé par le noyau

3245 free

PSHARE/MB: 1280 shared → gain dû à ***Page Sharing***

Paramètres mémoire selon vSphere

- *Summary*

- *Summary-VM*

- Beaucoup (trop=20) de paramètres disponibles : *Shared common*, Granted, Consumed, Balloon, Shared, Usage, Swap in, Active, Zero, Heap, Swap out rate, State, Unreserved, Reserved capacity, Overhead, Memory used by Vmkernel, Swap in rate, Swap out, Available heap memory, Swap used

- Distinguer entre niveaux physique (pRAM) et virtuel (vRAM)

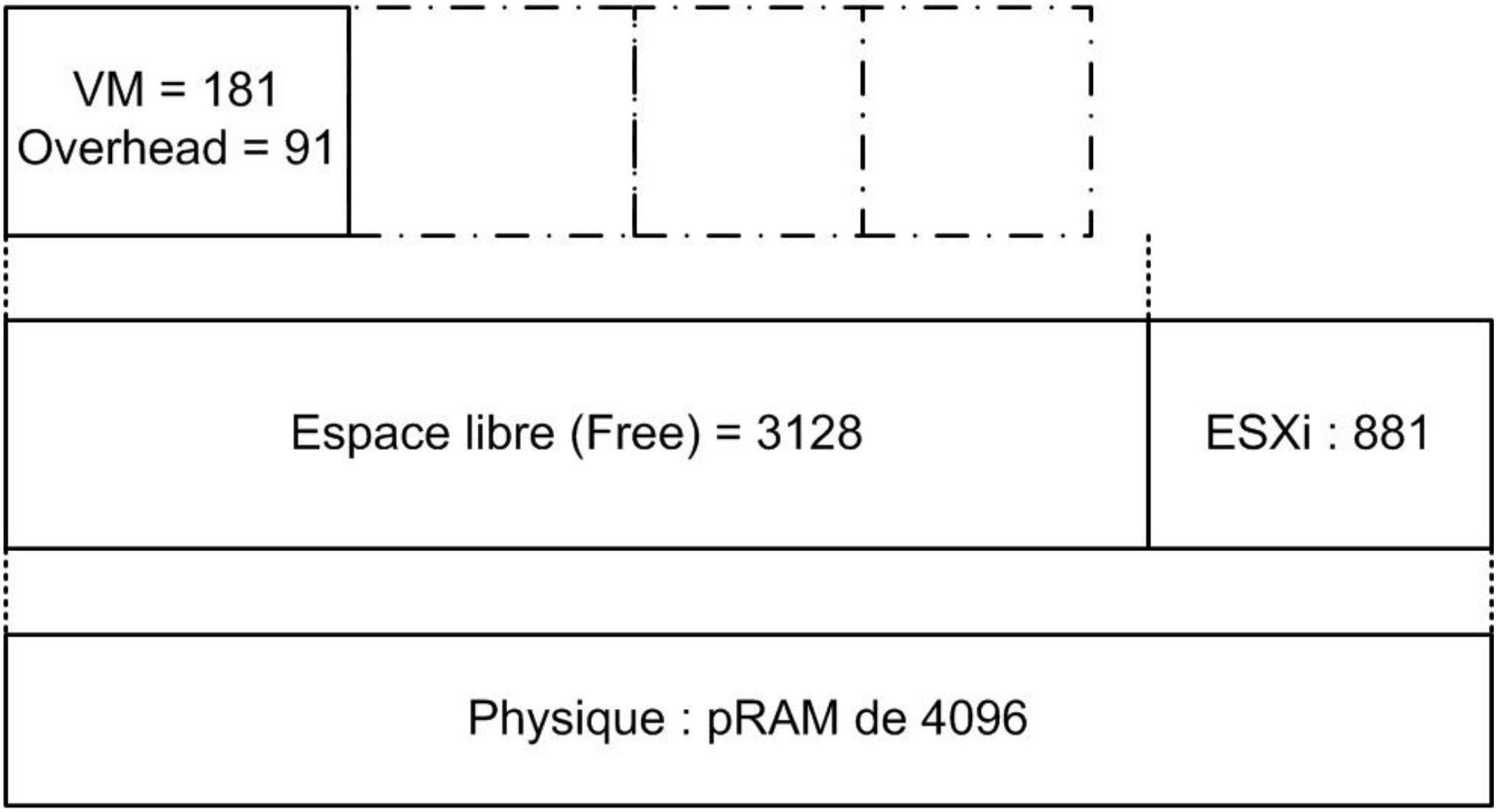
- Distinguer les mécanismes *Sharing, …, Balloon, Swap, …*

Affichages vSphere

- Temps réel avec intervalle de 20 s

	Rollup	Units	
CPU			
Usage	Average	Percent	→ moyenne en %
Usage in MHz	Average	MHz	→ moyenne en MHz
Idle	Summation	Millisecond	→ somme en ms
Memory			
Consumed	Average	Kilobytes	→ moyenne en Kbyte
Values	Latest	Maximum	Minimum Average

Bilan mémoire Linux en Mbyte (Labo §3.1 – 3.2)

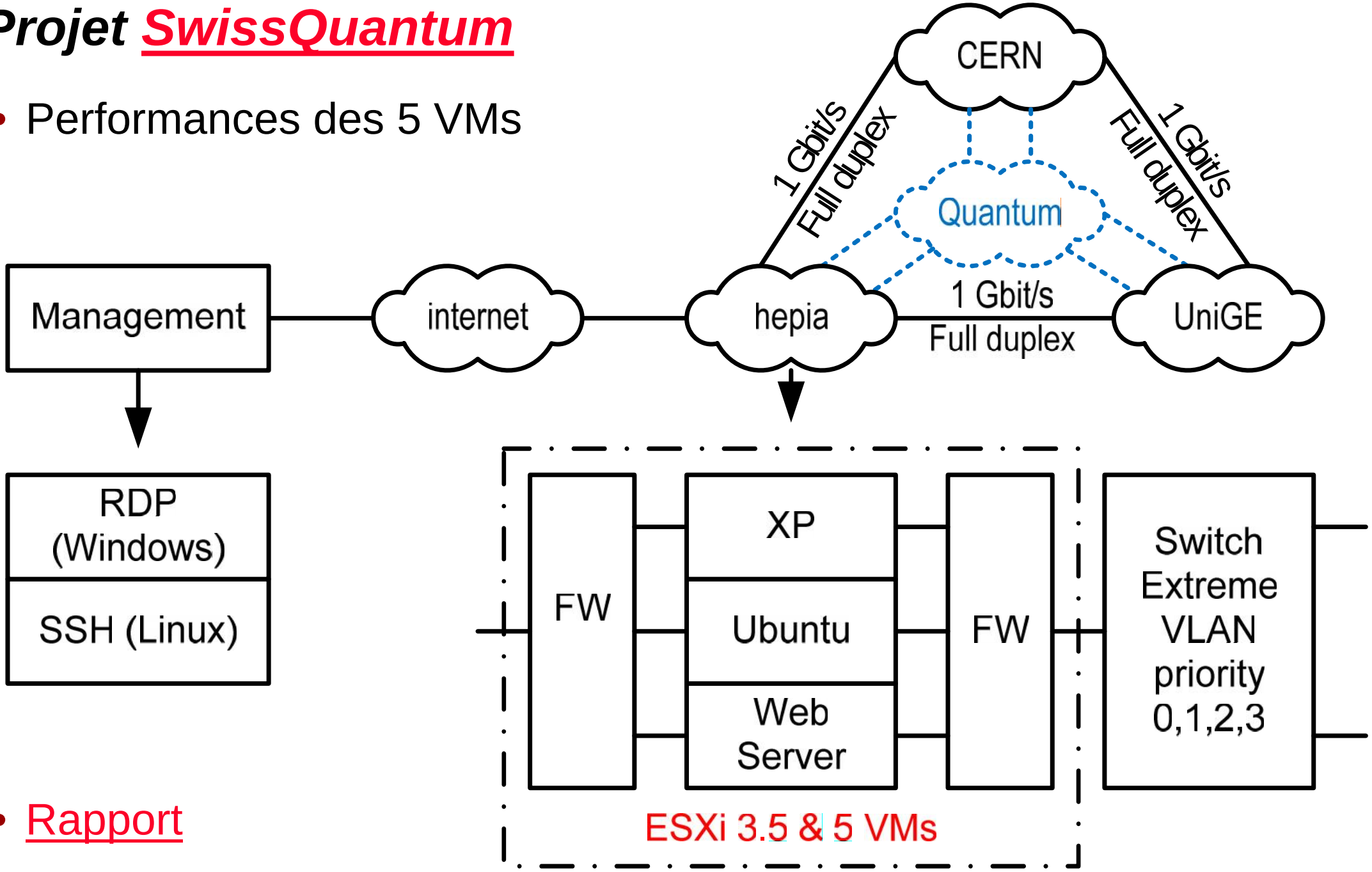


Bilan mémoire Linux (suite)

• Taille physique	4096 (=4GB RAM)	
		Outils
• Espace vu par ESXi	3991	vSphere + esxtop
• Espace libre	3128	esxtop
• Espace occupé	881	vSphere-ESXi
• Espace système	635	esxtop
• VM	181	vSphere-VM
• Overhead	91	vSphere-VM

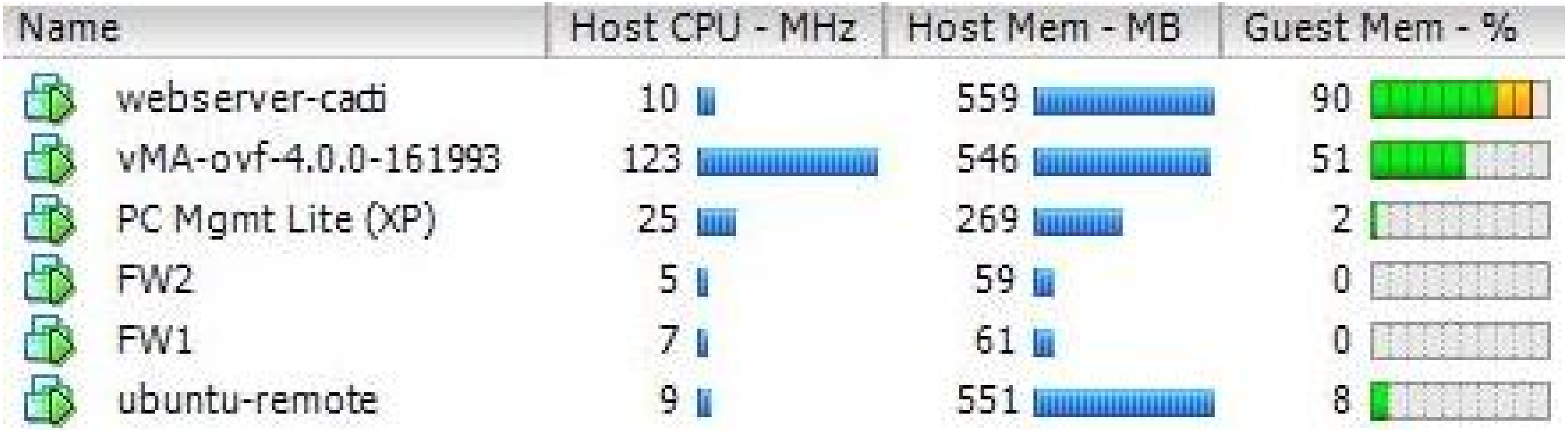
Projet SwissQuantum

- Performances des 5 VMs



- Rapport

Solution



Name	Host CPU - MHz	Host Mem - MB	Guest Mem - %
webserver-cacti	10	559	90
vMA-ovf-4.0.0-161993	123	546	51
PC Mgmt Lite (XP)	25	269	2
FW2	5	59	0
FW1	7	61	0
ubuntu-remote	9	551	8

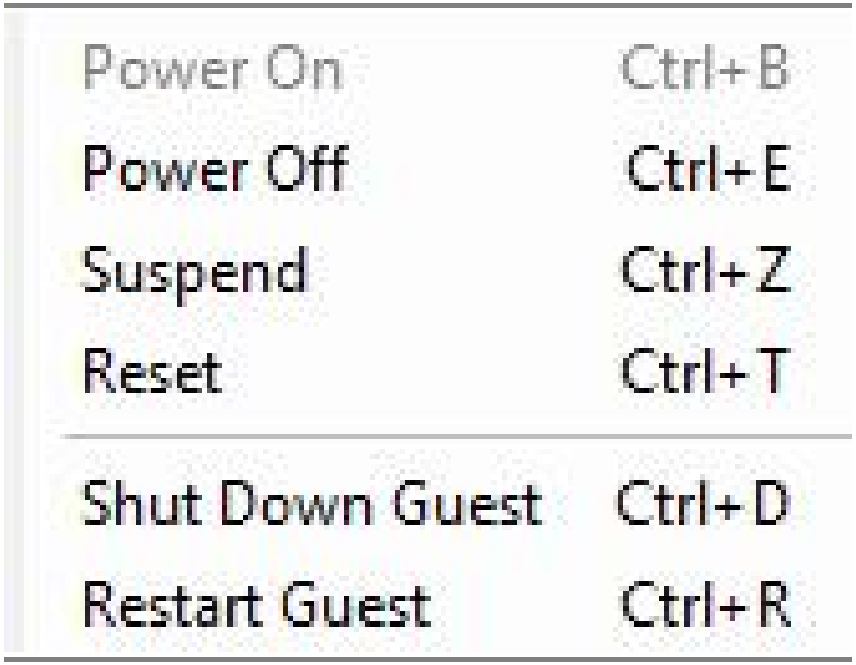
- esxtop → free = 1600 MB
- Affichage parfois lent de l'image dû à *Swap Guest (webserver)*
- Augmenter la taille de *Memory Size* dans les paramètres de la VM de 512 à 1024 MB

VMware Tools : descriptif

- Exécutables destinés à améliorer les **performances** et la **convivialité** (plus besoin de Ctrl+Alt pour quitter la fenêtre GUI)
- Il est recommandé de les installer bien qu'une VM puisse fonctionner sans ces outils
- **Pilotes** *(drivers)* pour le confort (écran, souris, partage des dossiers) et les **performances (vNIC + *balloon*)**
- **Services** *(daemons)* → ***Heartbeat*** (battement du cœur) périodique à vSphere
- Windows → objet dans la barre des taches *(taskbar)*
- Voir **VMware Tools Help**

Power Off / Shut Down Guest

- *Power Off* = ***hard*** *power off*
  équivalent à couper l'alimentation d'une machine physique
- La partie inférieure est proposée si les *VMware Tools* ont été installés



- *Shut Down Guest* = ***soft*** *power off*
  effectue un arrêt propre du système

VMware Tools

- Ces outils sont présents dans le système ESXi pour les OS suivants

```
# cd /usr/lib/vmware/isoimages/
# ls
freebsd.iso        netware.iso        linux.iso
solaris.iso        windows.iso
```

- Les fichiers sont copiés dans
  C:\ProgramFiles\VMware\VMwareTools
  /etc/vmware-tools

- Installation (Windows & Linux) → Labo
- Mises à jour (faciles pour Windows, complexes sous Linux → recompiler le noyau)

Labo level 2 §3 : RAM (45 min)

3 Occupation RAM (45') mesurée avec esxtop, vSphere et Task Man

3.1 Linux 32 bit → VM ubuntu_server

3.2 Evolution dans le temps

3.3 Comparaison avec XP 32 bit → VM XP_Tools

Virtualization level 2

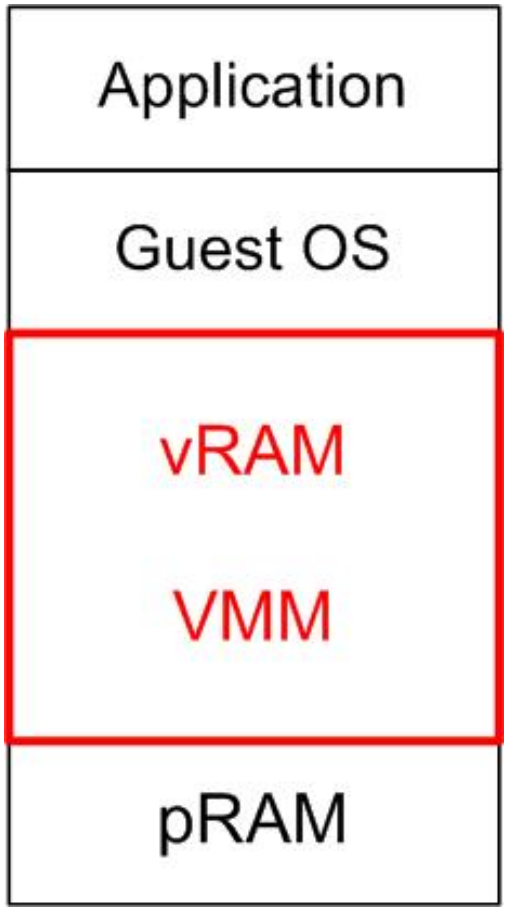
Bonne gestion d’une architecture virtualisée avec ESXi

3 : Sharing, Balloon, Swap

Synthèse du labo précédent

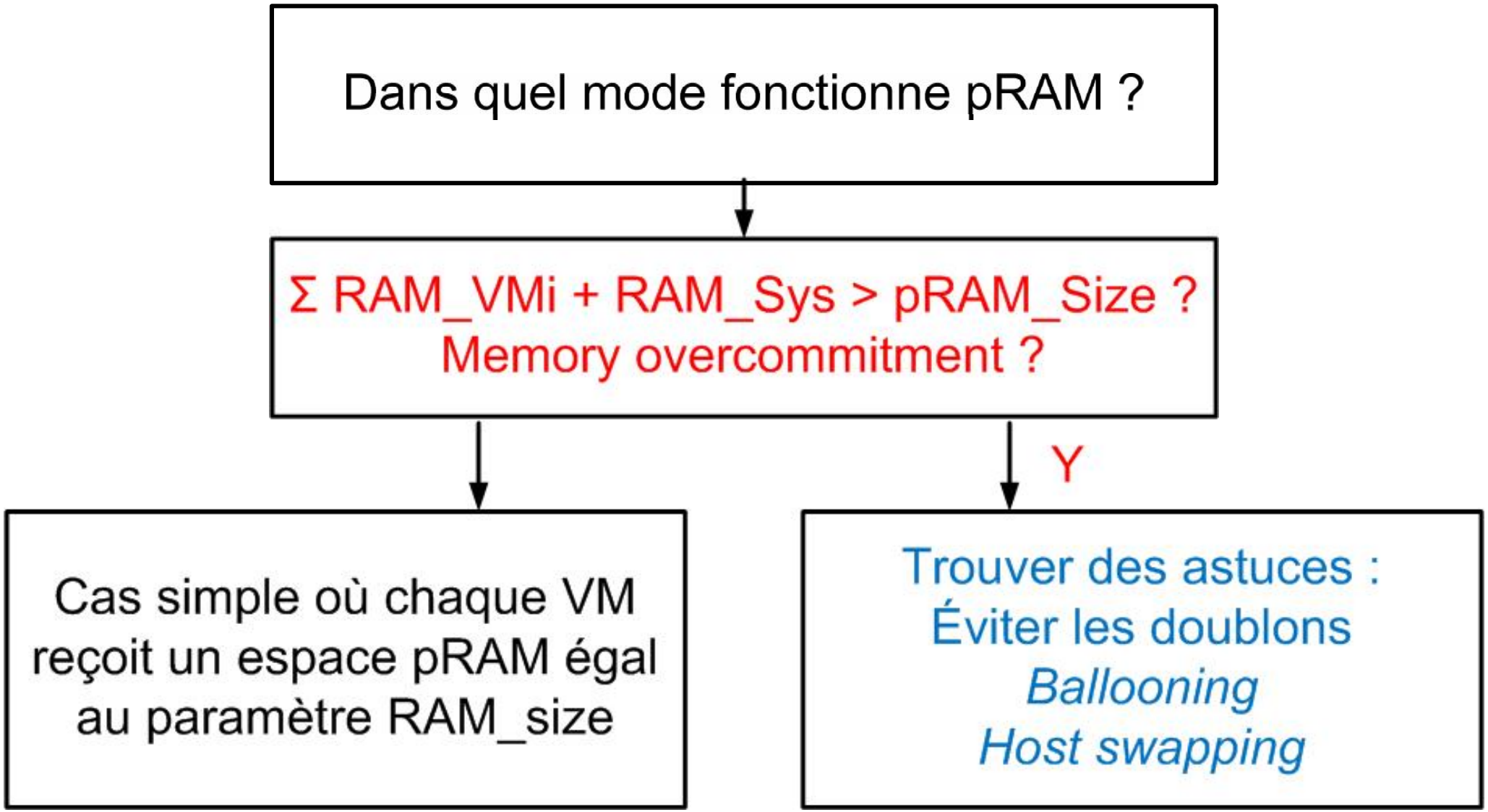
- Expliquer les résultats obtenus à la question 3j

Gestion mémoires (OS – VMM – RAM)



- VMM *(Virtual Machine Monitor)* présente à l'OS
  vRAM = une virtualisation de la RAM
- OS (Linux, Windows, …) croit posséder la RAM; il utilise en réalité vRAM
- Il utilise une adresse linéaire de 32 bit
  Espace adressable de 4 Gbyte pour CPU-32 bit
- VMM est seul à gérer pRAM = mémoire physique selon une technique connue = pagination !
- OS croit utiliser son mécanisme de pagination alors que l'accès à pRAM n'est contrôlé que par VMM

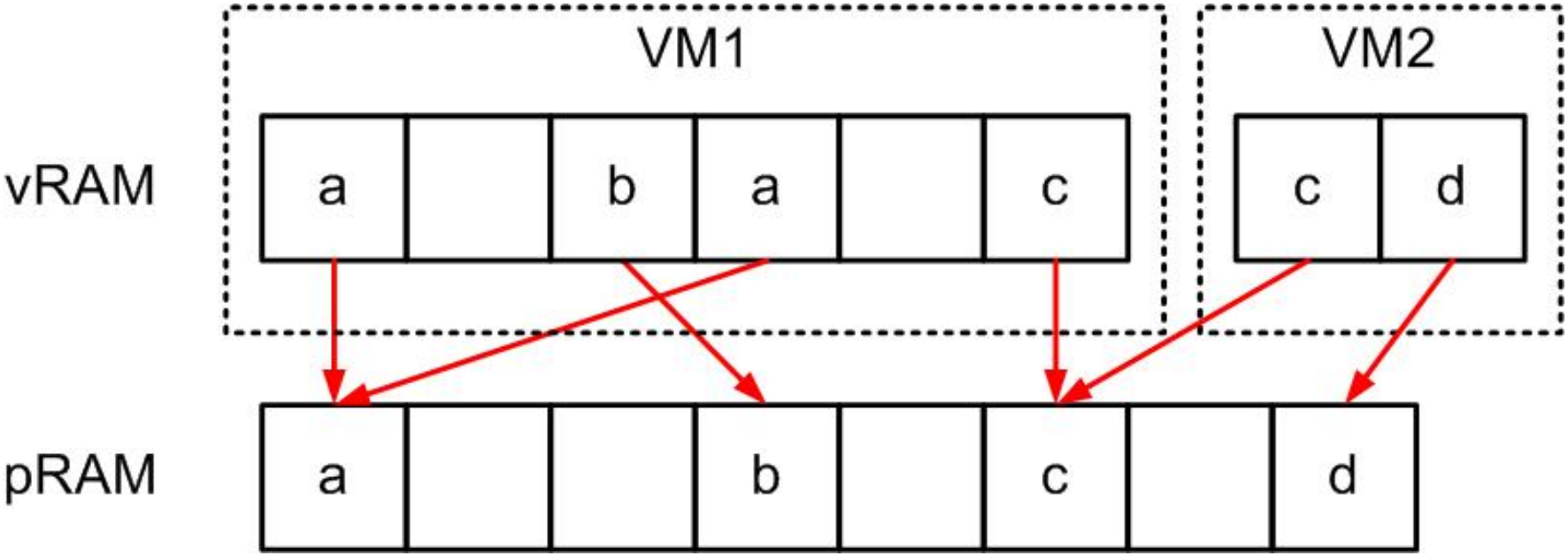
Modes de fonctionnement de la mémoire



Consolider le plus de serveurs

Memory Sharing

- Chaque carré représente une page de **4 KByte**



- **vRAM_VM1** = 4 pages → ***Memory Granted*** = 16 KByte
- **vRAM_VM2** = 2 pages → ***Memory Granted*** = 8 Kbyte
- **pRAM_VM1** = 2.5 pages → ***Memory Consumed*** = 10 KByte
- **pRAM_VM2** = 1.5 pages → ***Memory Consumed*** = 6 KByte

Memory Sharing implementation

- Les pages identiques sont identifiées à partir de leur contenu et d'une fonction de hachage
- En cas de détection d'un doublon, la page conservée est marquée *copy-on-write*
- En cas d'écriture, une exception est produite afin de créer une copie
- Faible impact (< 1%) sur la charge CPU
- Ce mécanisme reste invisible aux divers *Guest OS*

Gain obtenu avec XP 32 bit

- **1 VM** XP 32 bit (sans VMware Tools)

  Sans *Page Sharing* → *Memory Consumed* = 1060 Mbyte

  Avec **140 MByte**

- **2 VM** XP 32 bit (sans VMware Tools)

  Avec *Page Sharing* → *Memory Consumed* = **280 MByte**

- *Memory Size* = 1024 MByte
- Gain relatif important = 140/1060 = 13%
- Occupation mémoire proche entre XP (140) et Ubuntu server (120)

Gain obtenu avec Server 2008 64 bit avec VMware Tools

- **1 VM**

  Sans *Page Sharing* → *Memory Consumed* = 2085 Mbyte

  Avec **370 MByte**

- **2 VM**

  Avec *Page Sharing* → *Memory Consumed* = **740 MByte**

- *Memory Size* = 2048 MByte
- Gain relatif important = 370/2085 = 18%

Gain obtenu avec Ubuntu server 32 bit

- **1 VM** Ubuntu server 32 bit (sans VMware Tools)

  Sans *Page Sharing* → *Memory Consumed* = 110-120 Mbyte

  Avec **idem**

- **2 VM** Ubuntu server 32 bit (sans VMware Tools)

  Sans *Page Sharing* → *Memory Consumed* = 2 x (110-120) Mbyte

  Avec *Page Sharing* → *Memory Consumed* = **idem**

- *Memory Size* = 512 MByte
- Résultat identique avec VMware Tools
- Bizarrement, aucun gain !
- Explication → OS utilise des ***larges pages* > 4 kByte**

  http://www.tdeig.ch/vmware/TPS.pdf

Augmentation et diminution de l'espace RAM

- Imaginons une VM qui occupe 100 pages de pRAM qui ont été allouées par ESX

- Que se passe-t-il si cette VM a besoin de pages supplémentaires (pour pouvoir démarrer une nouvelle application) ?

  La VM va vouloir accéder à une adresse linéaire hors de l'espace alloué de 100 pages

  ESX va allouer une page supplémentaire, une page …

- Que se passe-t-il si cette VM n'utilise plus cet espace supplémentaire ?

  Risque qu'il soit gaspillé car ESX ne voit (facilement) que les augmentations *(allocate)*; pas les diminutions *(deallocate)* !!!

Ballooning

- ESX ne voit que les demandes d'allocation mémoire !
- ESX cherche donc à récupérer la mémoire physique pRAM
- L'astuce développé par VMware dans *ballooning* consiste à trouver le moyen d'identifier des espaces pRAM inutilisés (par une ou des VMs) pour les réallouer dynamiquement à une VM

Balloon implementation

- Les VMware Tools proposent le pilote vmmemctl *(balloon* driver) pour Linux, Windows, FreeBSD, …

- Il est capable d'occuper un espace mémoire variable (balon) dans chaque VM, via des appels systèmes get_free_page (Linux) & AllocatePages (Win)

- Il communique avec ESX qui va lui indiquer l'action à entreprendre → demande N pages libres, récupère N pages, réallouer N pages

- En mode *overcommitment of memory*, ESX optimise en permanence l'occupation RAM

Labo §5 : Balloon

5 *Balloon* (30')

Désactiver *Page Sharing*

Mesures avec esxtop & vSphere

VM XP_Tools

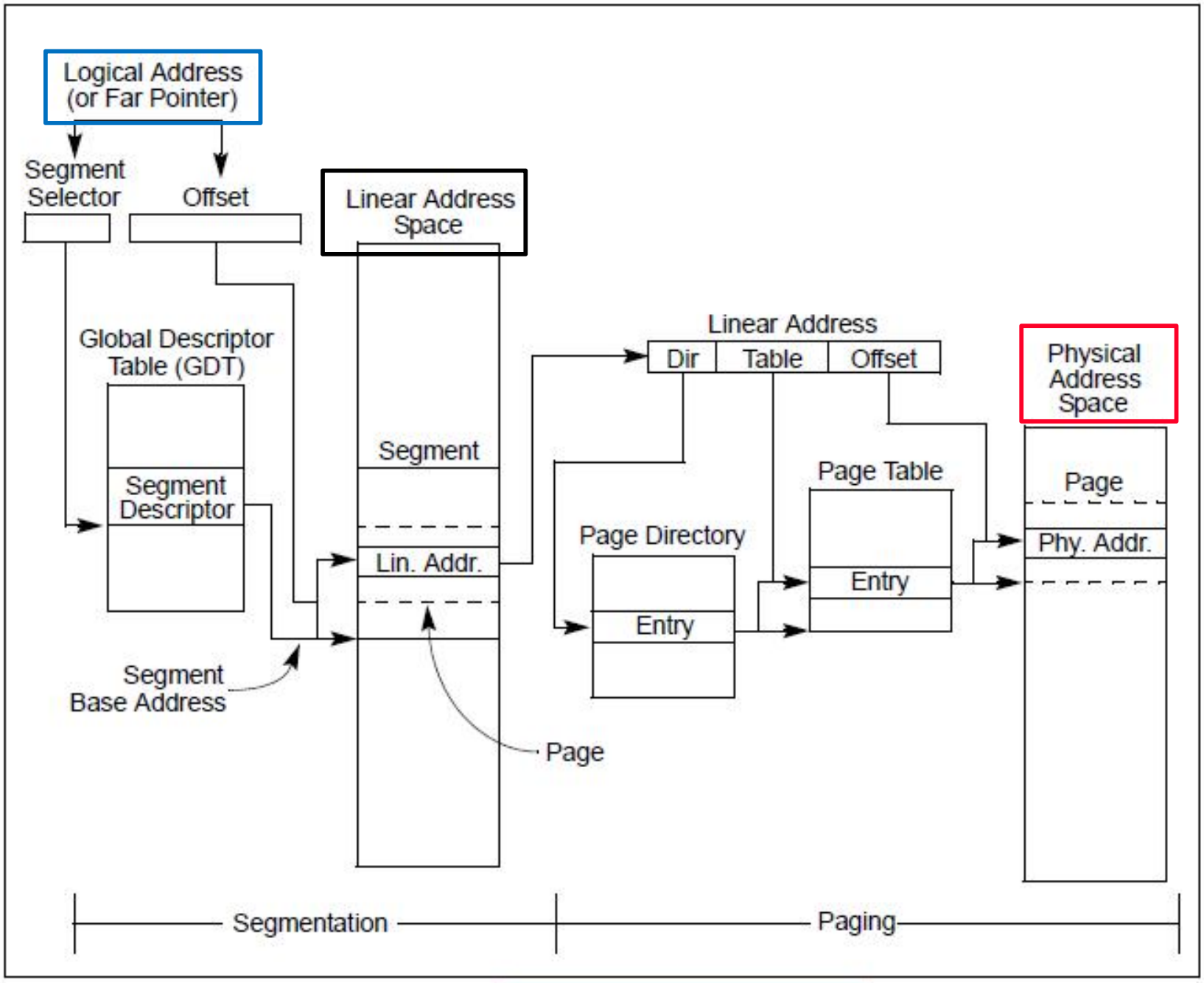
VM ubuntu_T_kernel
Simuler une demande mémoire avec **`./kernel-compile`**

Segmentation and Paging

Logical Address =
segment (16 bit)
+ offset (16 bit)

Linear Address
Space (32 bit)
= 4 GByte

Physical Address
to RAM hardware



3 types d'adresse (logique – linéaire – physique)

- Adresse logique

  Au niveau logiciel (code assembleur, code C++ compilé, …), le jeu d'instructions x86 utilise un segment et un déplacement *(offset)*

- Adresse linéaire

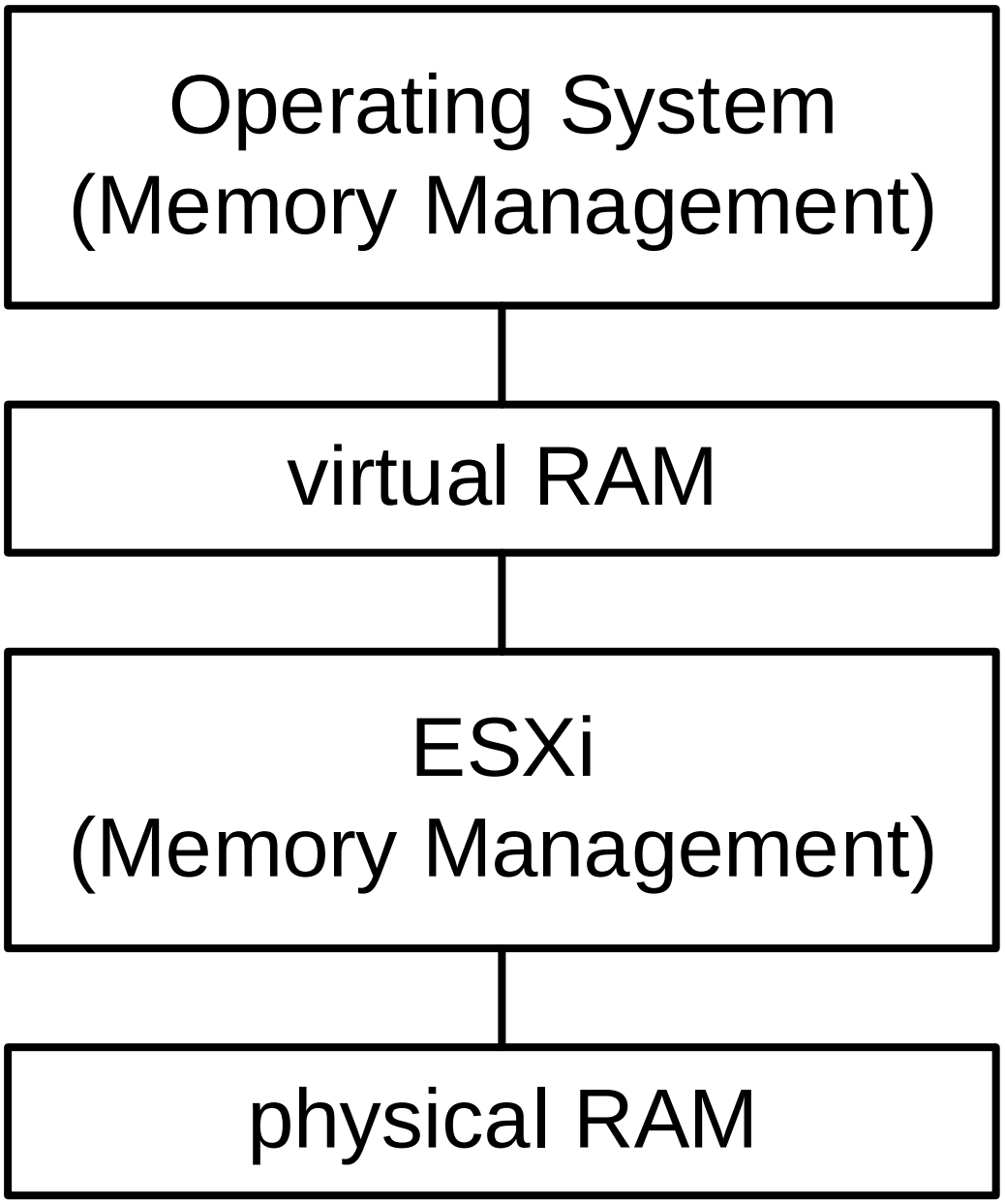
  Un processeur IA-32 est capable d'adresser 4 Gbyte

- L'unité de pagination traduit les adresses linéaires en adresses physiques et comprend un cache *(Translation Look-Aside Buffer)*

- Adresse physique générée par les signaux électriques sur le composant mémoire RAM

- L'espace physique est découpée en page *(frame)*

Pagination gérée par OS et par ESXi



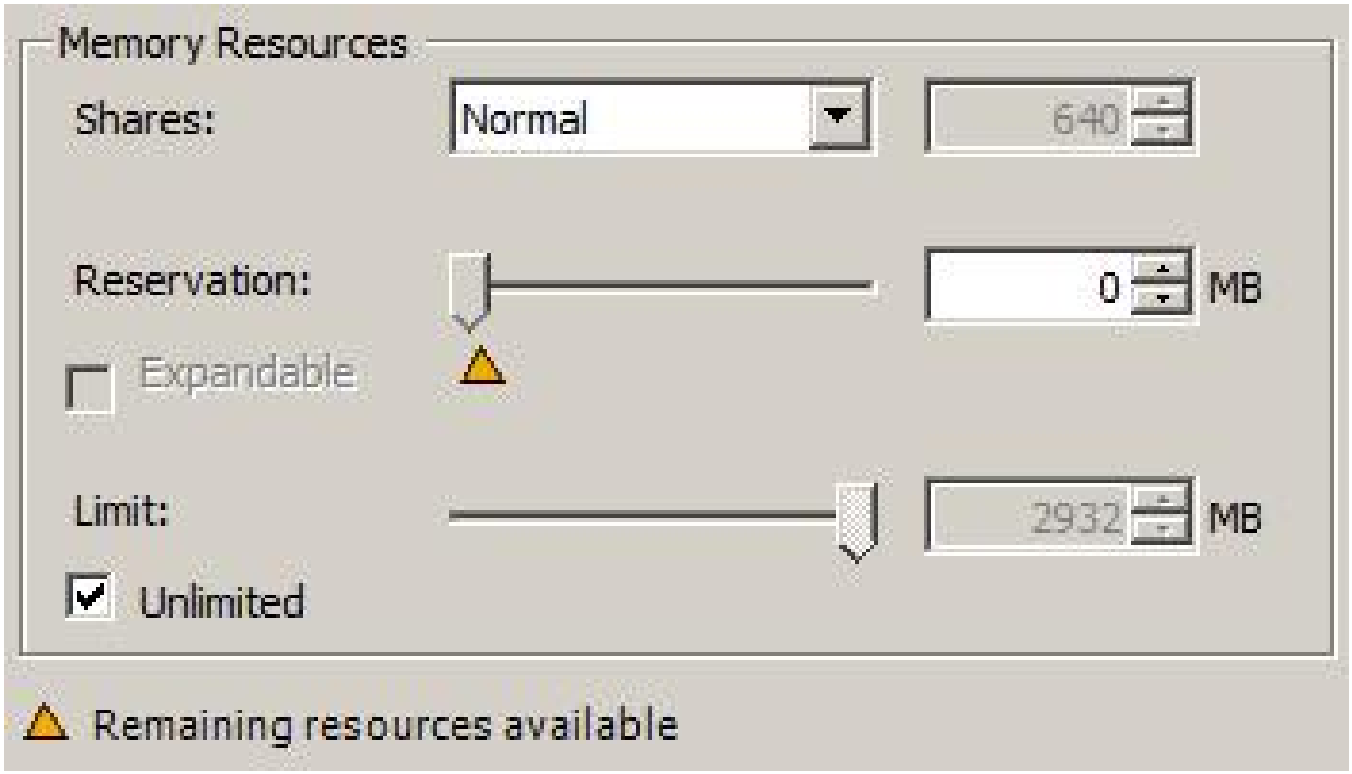
Le gestionnaire mémoire de Windows/Linux assure la pagination avec un éventuel swap = transfert RAM → disque

Idem pour le gestionnaire mémoire de ESXi

Swap

- Lors du démarrage de la VM, ESX crée un fichier swap **`VM.vswp`**
- Au besoin, ESX peut rapidement récupérer des pages en pRAM qu'il copie dans ce fichier de swap
- Cette opération est transparente à l'OS
- Elle reste pénalisante en matière de performances
- OS peut être amené à utiliser son mécanisme de swap ; ce qui peut entraîner une pagination croisée *(double paging problem)*

Memory Manager : allocation



- Espace physique non réservé est alloué en fonction des priorités relatives *(share)* de chaque VM
- Possibilité de réserver un espace physique (sans réallocation possible)
- Possibilité de limiter l'espace physique

esxtop

```
 3:26:21pm up 12 min, 101 worlds; MEM overcommit avg: 0.00, 0.00, 0.00
 heure act uptime                  requested / managed  1min  5min 15min

PMEM  /MB: 4083   total:   302     vmk,   247 other,   3533 free
physRAM    taille          système ESX                   libre

VMKMEM/MB  3877 managed:   232 minfree,   386 rsvd,   3369 ursvd, high state
Mémoire système                                                  %pRAM free

PSHARE/MB    94  shared,    34  common:    61 saving
Doublons

SWAP  /MB
Effectué par ESX

MEMCTL/MB
Balloon
```

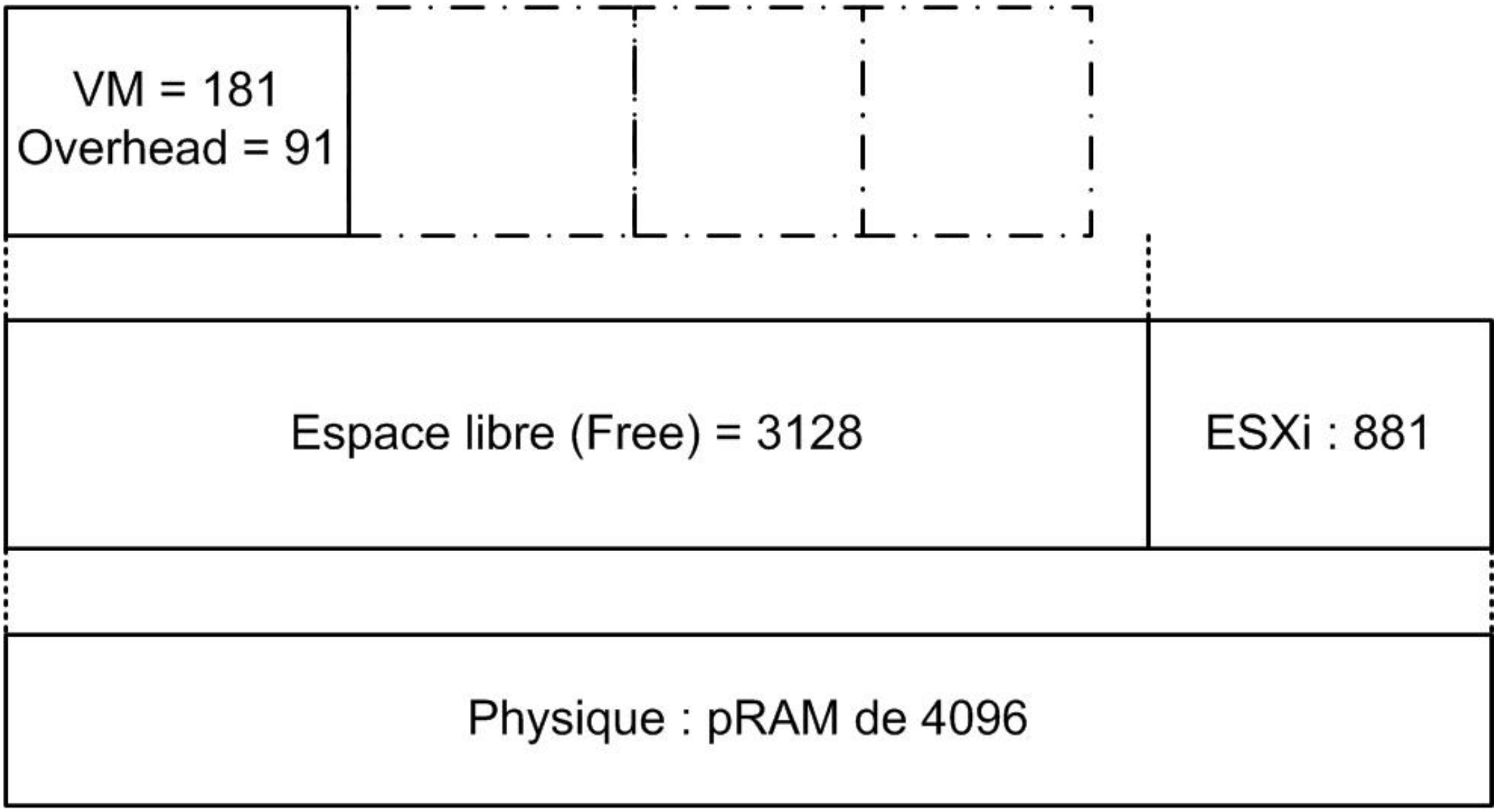
Best Practices for Memory Overcommitment

- Utiliser les mécanismes *Sharing* (par défaut) & *Ballooning* (VMtools)
- Donner une valeur appropriée à VM_memory_size

  Valeur supérieure à la valeur moyenne pour pourvoir absorber des demandes d'allocation et éviter la pagination par l'OS
- Utiliser la réservation avec parcimonie; il peut être nécessaire d'en réserver un minimum pour éviter le risque de se la faire voler
- Conserver la valeur *unlimited* sauf pour des cas particuliers

  even if host free memory is in the high state, memory reclamation is still mandatory when a virtual machine's memory usage exceeds its specified memory limit.
- Utiliser au besoin des priorités relatives avec *shares*

Bilan mémoire Linux en Mbyte (Labo §3.1 – 3.2)



Principaux compteurs mémoire

- **Libre** **PMEM : free (esxtop)**
- **Utilisé** **Consumed (vSphere)**
  espace pRAM utilisé par ESXi (global / VM)

- Shared PSHARE (esxtop), *shared* (vSphere)
- Balloon MEMCTL (esxtop), *balloon* (vSphere)
- Swapped SWAP (esxtop), swap (vSphere)

- Overhead espace pRAM supplémentaire pour gérer cette VM
- … beaucoup d'autres compteurs ! Réelle utilité ?

- http://communities.vmware.com/docs/DOC-10398 Local

Labo §6 : Swap (30 min)

6 Swap (30')

VM ubuntu_T_kernel

Effectuer §5.2 afin de désactiver le mécanisme de Page Sharing

6.2 Swap Guest (Linux)

6.3 Swap Host (ESXi)

6.4 Swap Host (ESXi) contrôlé via les priorités

Virtualization level 2

Bonne gestion d’une architecture virtualisée avec ESXi

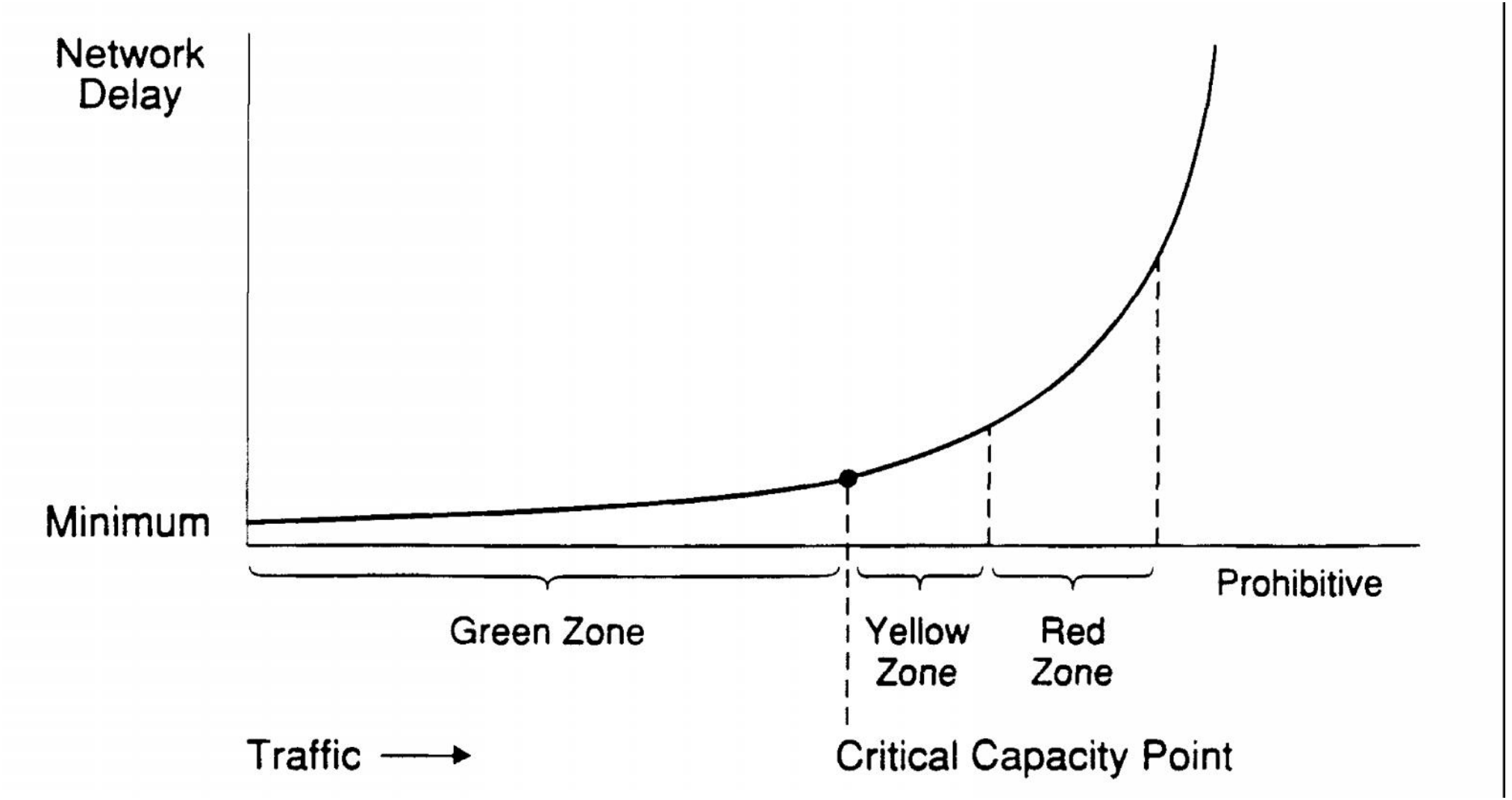
4 : Performances

Synthèse du labo précédent

- Que signifie Swap ?
- Pourquoi XP effectue-t-il du swap au §6.2 ?
- Pourquoi ESXi effectue-t-il du swap au §6.3 ?
- Quel est l'intérêt pratique du scénario joué au §6.4 ?

Symptôme

- L’utilisateur constate une augmentation du temps de réponse



Problématique (cas vécu)

- L’entreprise X (qui fabrique des ordinateurs) a besoin d’un nouveau serveur de base de données à usage interne (Genève)

- Sur la base du cahier des charges,
  le matériel (CPU-RAM-HD-Ethernet) est choisi,
  ainsi que le système d’exploitation et l’application DB

- Les performances réseau (commutateurs Ethernet) sont mesurées et respectent le cahier des charges

- Alors que chaque partie (matériel – logiciel – réseau) respecte le cahier des charges, le résultat final de son intégration se révèle vite inutilisable en raison de performances insuffisantes !

Modèle en couches

- L'analyse des performances est souvent un exercice complexe !

- Identifier les flux → producteur – consommateur

  HD → RAM, RAM → CPU → réseau, …, cache, …

- Identifier le goulet d'étranglement *(bottleneck)*

  Quel est le maillon (CPU – RAM – SAN – Network) critique ou sous-dimensionné ?

- Corriger si nécessaire

- Valider chaque couche (Ethernet, …, Oracle) avec des scénarios et des outils qui tiennent compte des exigences initiales → **références**

Partages des ressources

- La consolidation des serveurs dans une architecture virtualisée consiste à **partager les ressources** disponibles entre les différentes VMs

- L'évolution des télécom. est similaire : des capacités garanties de 3.1 kHz ou 64 kbit/s ont été remplacées par des canaux (internet) de communication offrant un service *best effort*

- Ce type de service ne pose pas de problème tant que l'utilisateur a la patience d'attendre ou que les performances du réseau sont suffisantes

- Applications critiques aux temps, telles la vidéo ou la téléphonie, sensibles au temps de transit (latence), à la gigue ainsi qu'à la bande passante

Classification au niveau réseau (commutateur, routeur)

- ***Elastic traffic (with flow control)***

  → modèle *Controlled Load Services* (rfc 2211) 1997

  s'appliquent à des réseaux *best effort* auxquels on ajoute :

  Le contrôle d'accès

  Différents niveaux de priorité *(low – medium – high)* → *DiffServ*

  La gestion de la congestion (perte des paquets)

- ***Inelastic traffic (real time)***

  → modèle *Guaranteed Services* (rfc 2212) 1997

  Réservation de la ressource (RNIS : canal de 64 kbit/s, …),

  protocole RSVP

Service Level Agreement (SLA)

- Déterminer les principaux paramètres du SLA à définir dans une relation entreprise – fournisseur d'accès à internet (ISP)

- Démo http://www.tdeig.ch/shinken/Bilgin_RTB.pdf p24,25,28,34,58
  http://www.tdeig.ch/shinken/Schaub_RPA.pdf
  http://www.shinken-monitoring.org/
  slides 73-75

Performance Troubleshooting

- **Complexité**

  Les systèmes d'information deviennent toujours plus complexes

  La virtualisation ajoute une complexité supplémentaire

  On ne maîtrise bien que ce que l'on comprend

  Plus un système est simple; plus il est facile à sécuriser

- **Qui** signale le problème ? utilisateur, …
- Quel est le **symptôme** ? c'est lent, …
- Est-il **reproductible** ? produit mature → *next slide*
- Le **quantifier** mesurer le temps de réponse, …
- Le comparer avec un éventuel SLA *(Service Level Agreements)*
- Trouver la **cause**

Common Criteria EAL4+ Certification

- VMware vSphere 4.0 and VMware vCenter Server 4.0 achieved Common Criteria certification at Evaluation Assurance Level 4 (EAL4+) under the Common Criteria Evaluation and Certification Scheme (CCS). Common Criteria is an international set of guidelines (ISO 15408) that provides a common framework for evaluating security features and capabilities of Information Technology (IT) security products, and EAL4+ is the highest assurance level that is recognized globally by all signatories under the Common Criteria Recognition Agreement (CCRA).

- Full story here:

http://www.vmware.com/company/news/releases/common-criteria-certification-vsphere.html

- http://fr.wikipedia.org/wiki/Crit%C3%A8res_communs
- Cas vécus par Vincent Udriot en 2016 : lancer 10 fois un script

  http://www.tdeig.ch/virtual/Udriot_RTB.pdf p67

Scalability (capacité à évoluer)

- Idéalement, une architecture virtualisée devrait être évolutive pour lui permettre de faire face à des utilisateurs et des charges supplémentaires (montée en charge) sans en impacter les performances → il suffit d'augmenter certaines ressources

- Remarque : combien de changements subi le SI de l'entreprise par an ?

- La montée en charge peut être prévisible (acquisition d'une startup de 10 personnes dans 2 mois)
- Elle peut aussi être massive dans un interval de temps court

  → comment faire face à une demande instantanée 100 x supérieure ?

Méthodologie

- Anticiper les problèmes

  Idéalement : vous détectez un problème avant les utilisateurs

- Connaître le **profil** de chaque VM qui va devenir une **référence**

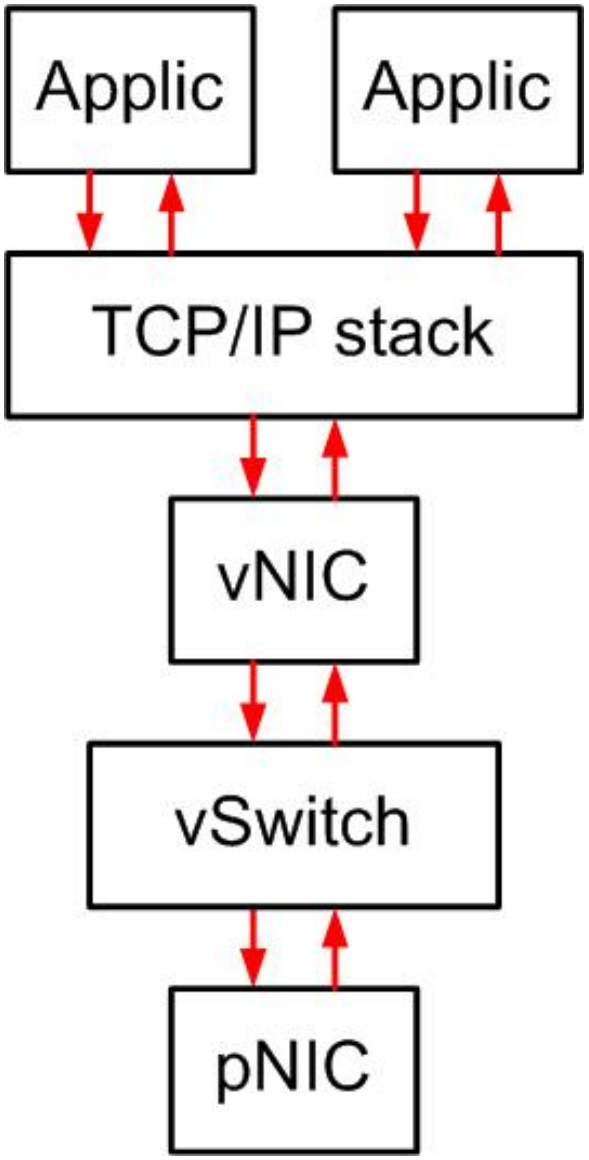
  Le compléter éventuellement avec des mesures du *Guest OS*

- Penser à le suivre périodiquement ; à le mettre à jour

  10 utilisateurs au début – 20 aujourd'hui – X demain

- Profil = rapport de mesures **(min – moyen – max)** des principaux paramètres CPU – RAM – *Storage* – *Network*

Network problems ?



- Un système virtualisé comprend diverses entités (processus, matériel) indépendantes interconnectées à l'aide de FIFO
- FIFO = *First In First Out* = tampon de taille fixe entre producteur et consommateur
- Un producteur qui écrit dans un tampon plein produit un *dropped packet*
- ↓ *Tx dropped packet*
- ↑ *Rx dropped packet*

Conseils VMware

- http://communities.vmware.com/docs/DOC-10352
- *VMware Tools* présents et à jour dans chaque VM
- Charge CPU → %RDY, Idle
- Charge RAM → *overcommitment*

  espace réservé = *overhead* + …

  léger pris en charge par le mécanisme *balloon*

  fort exigeant l'aide du disque *(swap)*
- *Storage* → débit utile
- *Network* → paquets perdus

Labo 7-9 (40 min)

7 Mesures de charge (CPU, réseau) avec vSphere et esxtop (20')

Générateur de charge httperf → CPU & network

VM Ubuntu Webserver

VM Ubuntu Desktop

9 Réseau (20')

Mesure du débit utile entre 2 VMs avec l'outil netserver

9.1 2 x VM XP

9.2 Gain (de 3) apporté par le pilote des VMware Tools

→ 2 x VM XP_Tools

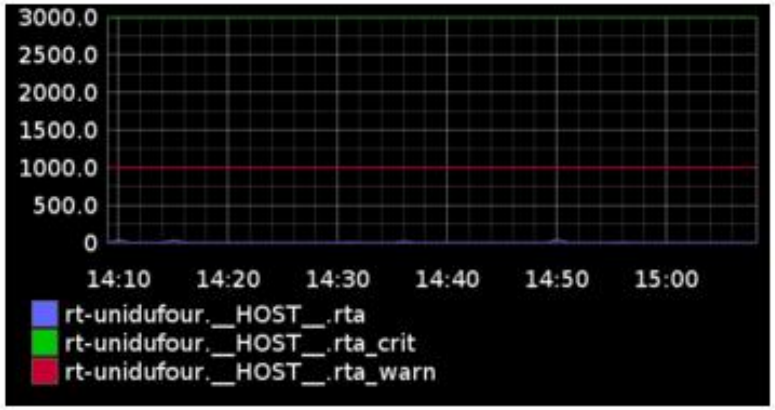
9.3 Linux → gain de 20-30

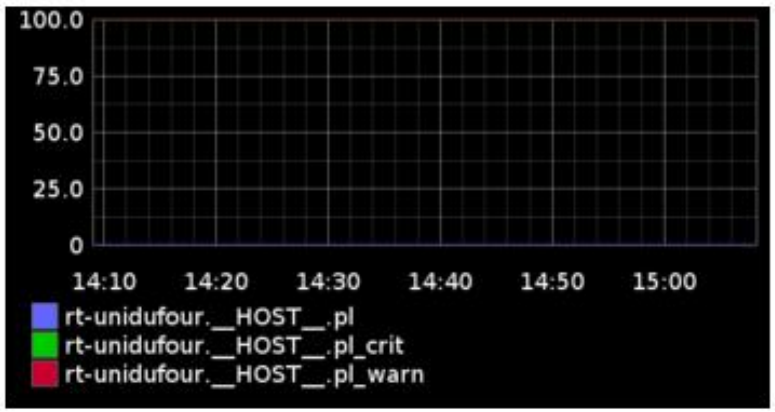
→ 2 x VM Ubuntu Desktop

Home / Dashboard

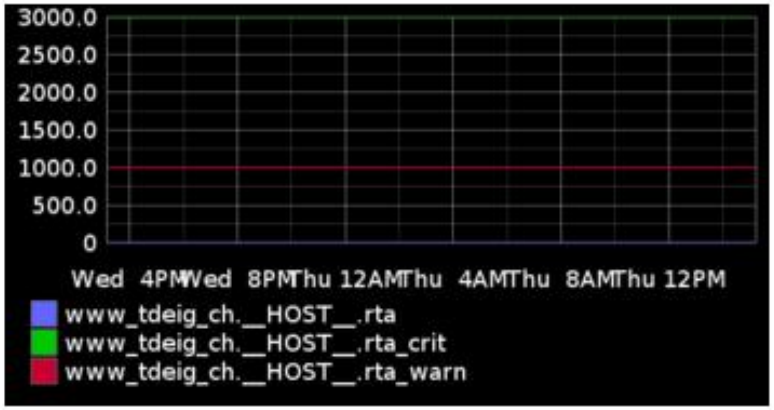
+ Hosts + Services

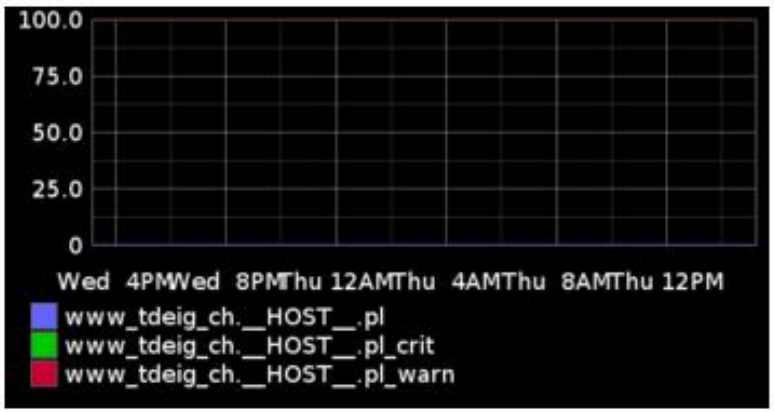
Element graphs for rt-unidufour





Element graphs for www.tdeig.ch



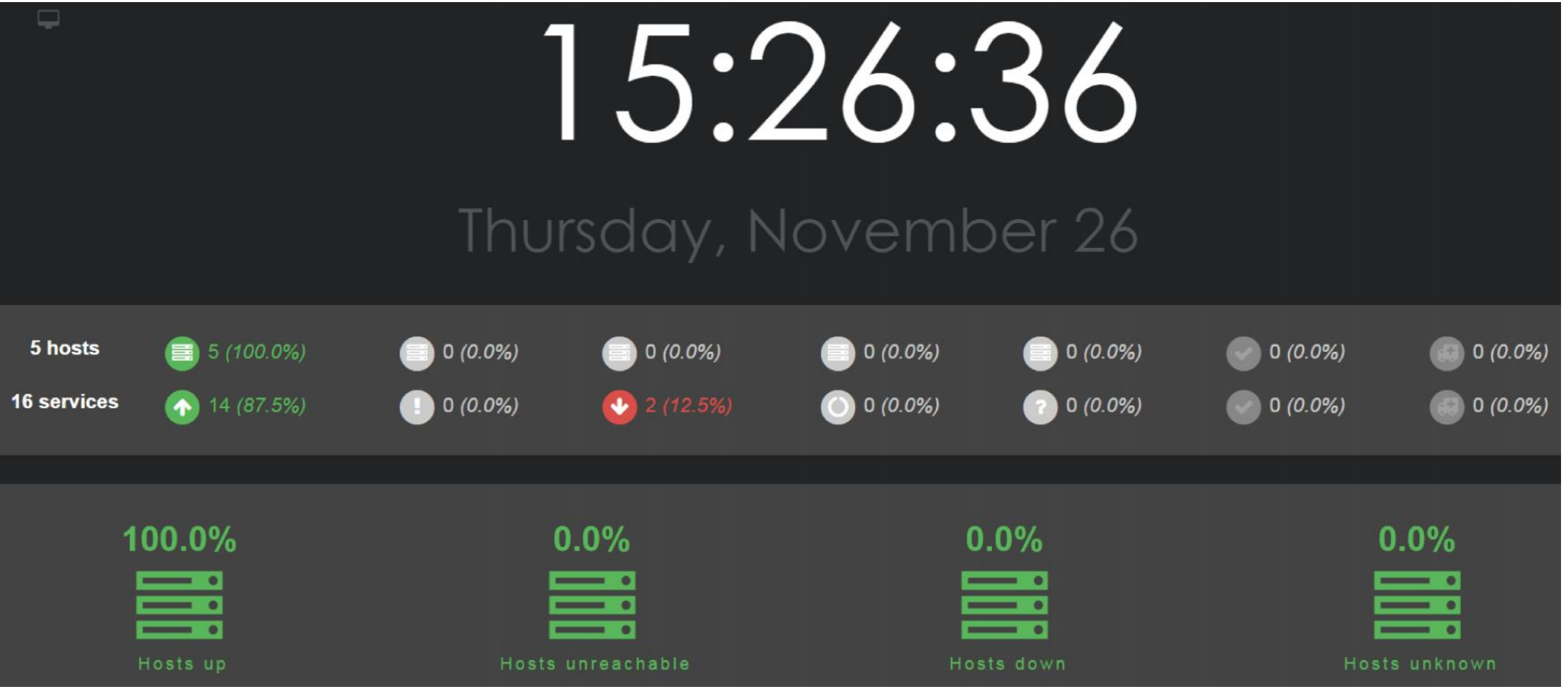


Select all elements

Business impact: Normal

5 elements

Host	Service	State	Duration
ca.tdeig.ch		UP	2w 3h
rt-unidufour		UP	5d 20h
srv-shinken		UP	3w 6d
tdeig.ch		UP	2w 3h
www.tdeig.ch		UP	2w 3h


15:26:36
Thursday, November 26
5 hosts
16 services
5 (100.0%)
0 (0.0%)
0 (0.0%)
0 (0.0%)
0 (0.0%)
0 (0.0%)
0 (0.0%)
14 (87.5%)
0 (0.0%)
2 (12.5%)
0 (0.0%)
0 (0.0%)
0 (0.0%)
0 (0.0%)
100.0%
Hosts up
0.0%
Hosts unreachable
0.0%
Hosts down
0.0%
Hosts unknown