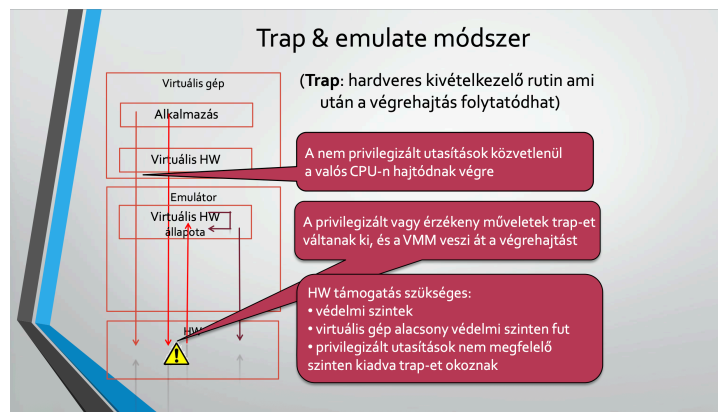


1. Mi az a memória felfújás?

- **Memória felfújás (memory ballooning)**
 - Ha kifogy a host memóriája, akkor „elvesz” a vendégtől
 - Egy ágens vagy driver a vendég kernelben (paravirtualizációs szemléletmód) elkezd memóriát foglalni a VMM utasítására. A VMM az ágens által „foglalt” memórialapok mögé nem is allokal fizikai memóriát, így nyer vissza helyet
 - Egyrészt a vendég fel fog adni a disk cache-ből,
 - Másrészt el fog kezdeni kilapozni a saját swap területre, elkerüli, hogy a host is swappeljen

2. Írja le mi a lényege a Trap&Emulate módszernek!

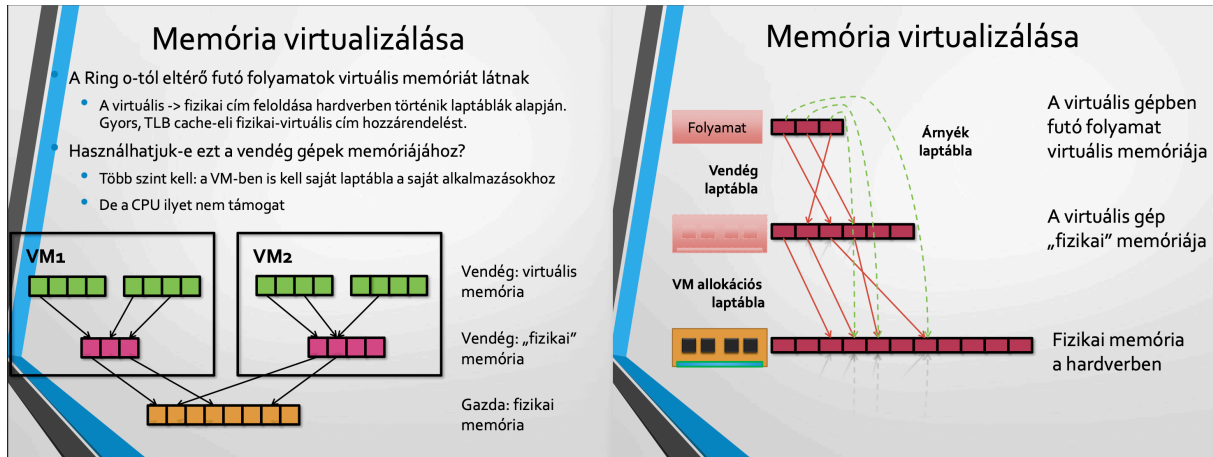


3. Mit jelent a JIT, és mi a feladata?

- **„Éppen időben” fordító (JIT compiler „just in time”) –** végrehajtás előtt egy fordító feldolgozza a virtuális gép soron következő utasításait, és kódot generál belőle, ami az eredetivel ekvivalens viselkedést mutat
 - Közvetlenül futtatható kódot generál, cache-elési eljárásokkal gyors lehet
 - Nehéz implementálni, nem hordozható
 - Izolációt nem automatikusan biztosítja
 - Példa: QEMU x86 emulátor (és Java VM, .NET CLR is ilyen)

4. Virtuális gépek esetén hogyan van megoldva a valódi fizikai memória címzése, és mivel több a hagyományos memóriacímzésnél?

(Röviden összefoglalva az árnyéktábla használata a kulcs, ami azért kellett, mivel az alpból két szintes memóriacímzést ami egy hagyományos rendszeren működik nem tudunk tovább növelni, ahhoz hardveres támogatás kell – ezen felül nyilván megoldása válogatja mi kezeli ezt, hardveres támogatásnál az árnyéktáblát a rendszer tudja kezelni stb. De itt csak az árnyéktábla használata a lényeg)



5. Hogyan épül fel egy RAID 01 megoldás? Írjon le róla annyi jellemzőt, amennyit csak tud!



(pluszba lehet gondolni az írási és olvasási sebességre is, ami habár direkt RAID 01-nél kifejtve nincs, a RAID 1 és RAID 0-hoz viszont igen, ez alapján egy várható teljesítményjavítást is lehet írni gondolatmenettel, indokolva, hogy ez meg az ennyit tud, így a RAID 01 összességében ideálisan ennyit tudhat. Összes jellemzőhöz még hozzátartozik, hogy mennyire redundáns, tehát mennyi lemezvesztést bír el! Illetve az is, hogy mi határozza meg a használható tárhelymennyiséget, akár egy példán keresztül)



6. Adja meg a felhő rendszerek csoportosítását cél szerint, és röviden jellemezze őket!

- Infrastruktúra mint szolgáltatás (IaaS)
- Platform mint szolgáltatás (PaaS)
- Szoftver mint szolgáltatás (SaaS)

(plusz a rövid összefoglalás mindegyikhez)

7. Hasonlítsa össze a konténer és a hypervisor alapú virtualizációt!

- Hypervisor
 - Egy fizikai, számos virtuális hardver
 - Vegyes operációs rendszerek
 - Kisebb sűrűség
 - Kisebb teljesítmény
- OS szintű virtualizáció
 - Egy fizikai hardver, nincs virtuális
 - Egy kernel, userspace példányok
 - Nagy sűrűség
 - Közel natív teljesítmény

(más különbség is elfogadható, a kernel használata pl. ilyen amit gyakran láttam. Viszont csak a két módszer külön-külön általános jellemzése itt nem elég! Konkrét különbségekre kell kitérni adott szempontból.)

8. Mi a különbség szimuláció és emuláció között?

Szimuláció és emuláció

Ekvivalencia biztosítása

- **Szimuláció:** szoftverrel modellezzük a processzor belső működését (különböző mélységig megtartva a valósághűséget). Fizikailag sohasem fut a virtuális gép kódja a CPU-n.
- **Emuláció:** helyettesítjük a VMM felett futtatandó programot/annak egy részletét egy másikkal, ami végül pontosan ugyanazt az eredményt adja (de akár teljesen más futási utat bejárva, elkerülve a privilegiált utasítások használatát)



9. A VM kernelének milyen módszerekkel tudjuk megoldani azt, hogy módosítsa laptábláját? Röviden írja le az elvüket is!

Memória virtualizálása

- Mi van, ha vendég kernel módosítani akarja a laptábláját?
 - Megfelelően frissíteni kell az árnyék táblát is
- Három megoldás:
 1. Természetesen Trap and emulate, de hogyan?
 - Read-only-ra állítjuk a vendég kernel számára látható laptáblákat, ha azt módosítani akarja, akkor jön a kivétel, átkerül a vezérlés a VMM-hez ami biztonságosan elvégzi a módosítást az árnyék táblán is
 2. „természetesen?! trap and emulate?”
 - A vendég kernel egyszerűen ne maga akarja módosítani a laptáblát, *kérje meg* a VMM-et erre ☺
 - Paravirtualizációs megoldás
 3. Hardveres kiegészítés több szintű laptáblák kezelésére
 - Core i7 és Phenom processzoroktól kezdve van (EPT / RVI)
 - Az egész árnyék tábla frissítési problémát hardveresen lekezel
 - Ezzel együtt hardveres virt. gyorsabb lehet a szoftveres virtualizálásnál

10. Adja meg az NIST által megfogalmazott 5 alapelvet mely egy felhő infrastruktúra jellemzője! (csak felsorolás)



Az NIST körülírta – még ha csak vázlatosan is – a felhő alapú rendszerek leglényegesebb tulajdonságait

- Igény szerinti önkiszolgálás (On-demand self service)
- nagyteljesítményű hálózat (Broad network access)
- Erőforrás készletekre épülnek (Resource pool)
- teljes és gyors rugalmasság (Rapid elasticity)
- mérhető adatokat biztosítanak (Measured Service)