SISTEMAS VIRTUALIZACION Y SEGURIDAD

MARZO 2022

PREGUNTA 1 INTRODUCCION Y RENDIMIENTO (1 PUNTOS)

a) ¿De las diversas configuraciones de *SpecCPU* cuando **es** ventajoso disponer de múltiples CPUs en el sistema bajo test (i.e. añadir más CPUs **mejora** el comportamiento)?
b) ¿Cuáles son las cargas/aplicaciones del *benchmark* SpecWeb2005 y qué intentan modelar?
c) ¿Para un servidor determinado, con cuantos almacenes es previsible el *benchmark* SpecJbb 2005 alcance su máximo rendimiento?
d) ¿Por qué se emplea la media geométrica al calcular el rendimiento promedio de las aplicaciones para calcular el SpecMark en SPECint y SPECfp?

PREGUNTA 2 CLOUDS PUBLICOS Y CONTAINERS (1.5 PUNTO)

a) ¿En que condiciones disponer de servidores *on-premises* **no** es económicamente ventajoso con respecto a emplear servidores en el *cloud?*
b) ¿Por qué una máquina virtual es mas segura que un *container*? ¿Qué soluciones conoces para paliar el problema en los *containers*?
c) ¿Qué utilidad tienen los *cgroups* con los *containers*?
d) ¿Por qué es más ágil un *container* que una máquina virtual?
e) ¿Cómo se puede ejecutar un contenedor *docker* en un host OSX?

PREGUNTA 3 MÁQUINAS VIRTUALES (1.5 PUNTOS)

a) ¿Qué diferencia una máquina virtual de un simulador?
b) ¿Qué es una instrucción crítica?
c) ¿Cuál es la diferencia entre un *World-switch*, de acuerdo con la terminología de VMWare, y un cambio de contexto?
d) ¿Para que sirve la tabla de páginas en la sombra? ¿Cuántas hay por máquina virtual?
e) ¿Por qué MIPS no es virtualizable?

PREGUNTA 4 SOPORTE ARQUITECTURAL VIRTUALIZACION MEM Y CPU (1.5 PUNTOS)

a) ¿Por qué diferentes implementaciones de x86 las instrucciones asociadas a VT-x tienen un rendimiento diferente (tendiendo a ser más rápidas en los procesadores más recientes)?
b) ¿Cómo logra *Vt-x* evitar el problema de las instrucciones criticas sin romper la compatibilidad hacia atrás (i.e., romper el ISA)?
c) ¿Qué mecanismo se usa para comunicar el hipervisor con el hardware en el caso de VT-x?
d) ¿Por qué no es posible aplicar las ideas de *Xen* en *VMWare* (teniendo en cuenta los objetivos de diseño de ambos)?
e) ¿Por qué en las primeras versiones de VT-x, las soluciones software (como la paravirtualización en Xen) podía ser más rapidas que la asistencia hardware?

PREGUNTA 5 ENTRADA SALIDA (1.5 PUNTOS)

a) ¿Por qué un dispositivo paravirtualizado va a ser habitualmente más rápido que uno emulado?
b) En un servidor moderno, ¿Dónde se encuentra la IOMMU y cuál es su utilidad?

(….)

c) Explicar brevemente como funciona la interposición de entrada salida y por que no es necesaria en un dispositivo de I/O y sistema con SRIOV
d) ¿Qué papel juega *Qemu* en una máquina virtual *kvm*?
e) ¿Por qué una maquina virtual usando un dispositivo SRIOV puede sufrir mas interrupciones que el mismo dispositivo en *bare-metal*? ¿Qué consecuencias tienen esas interrupciones?

PREGUNTA 6 INTRO SEGURIDAD (1.5 PUNTOS)

a) ¿En que consiste un ataque *Rowhammer*? ¿Qué mecanismos se pueden emplear para evitar sus consecuencias?
b) ¿Cuál es la diferencia entre confidencialidad e integridad?
c) ¿Cuál es la diferencia entre *secure hashing* y encriptación?
d) ¿Qué tipo de arquitectura de seguridad en memoria emplearías si lo único confiable fuese tu propia aplicación? (i.e. ni el OS ni hipervisor son confiables).
e) ¿Cómo puede limitar el fabricante de un dispositivo que tipo y versión del sistema operativo podemos instalar en el (p.ej. como hace Apple)?

PREGUNTA 7 PRACTICO SEGURIDAD (1.5 PUNTOS)

a) Explicar como funcionan los ataques tipo Spectre y justificar su peligrosidad. ¿Qué soluciones existen para defendernos de ellos?
b) ¿Por qué es necesario emplear *cifrado* y *hashing* en memoria para garantizar que su contenido sea seguro?
c) En un sistema multiprocesador, ¿Cómo podemos lograr que todos los controladores de memoria empleen la misma clave de cifrado para una máquina virtual dada?
d) Citar un ejemplo de utilidad del *sealing* y explicar cómo funciona.
e) ¿Cuál es la diferencia entre un *Side channel* y un *Covert Channel*? Citar un ejemplo de cada caso.