Efficient Memory Virtualization

Reducing Dimensionality of Nested Page Walks

Rodrigo Fernández Silió

Universidad de Cantabria

Índice

Virtualización

La virtualización es una tecnología que permite la creación, gestión, administración y ejecución de máquinas virtuales (VM, por sus siglas en inglés, Virtual Machine) mediante el uso de un monitor de máquina virtual (VMM, por sus siglas en inglés, Virtual Machine Monitor), también conocido como hipervisor.

Virtualización

Beneficios de la virtualización:

- Gestión de recursos
- Consolidación de servidores
- Seguridad
- Tolerancia a fallos

Sobrecarga de la virtualización

Los beneficios de la virtualización, sin embargo, vienen acompañados de sobrecargas en procesamiento, E/S y memoria.

Afortunadamente, los avances en hardware para virtualización (por ejemplo, la virtualización de E/S en hardware) han reducido considerablemente estas sobrecargas.

Sin embargo, las sobrecargas de la virtualización de memoria no son universalmente bajas.

Sobrecarga de la virtualización

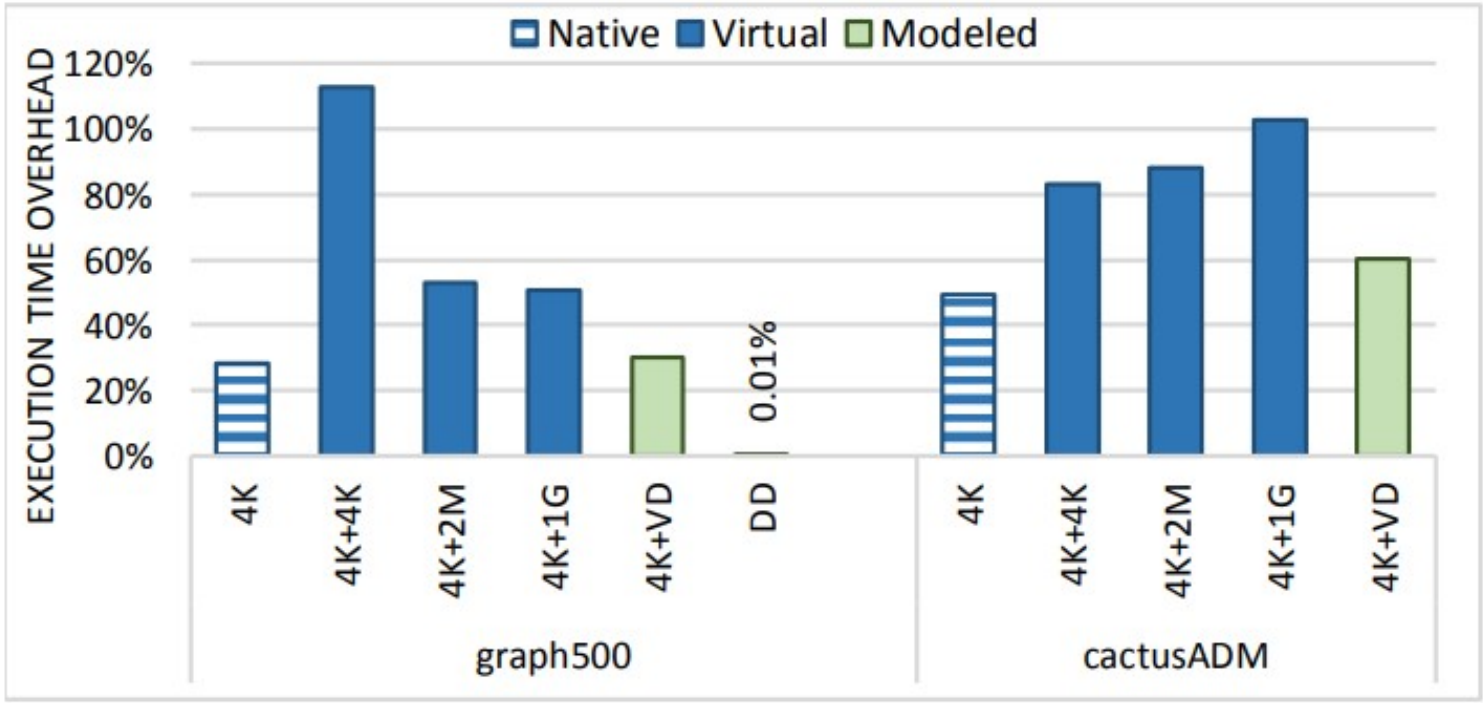

Figura: Sobrecarga asociada con la memoria virtual para las diferentes cargas de trabajo y configuraciones seleccionadas

Se observa que los costos aumentan drásticamente con la virtualización y permanecen altos incluso cuando el VMM utiliza páginas más grandes.

Tres nuevos modos de virtualización

Introducción de nuevo hardware que soporta 3 nuevos modos de virtualización para reducir las sobrecargas de la traducción de direcciones virtualizadas.

Este HW es una extensión de los segmentos directos (DD, por sus siglas en inglés, direct segments), previamente propuestos para disminuir los fallos de la TLB en sistemas no virtualizados.

Dos nuevas técnicas que mejoran los segmentos directos

- **self-ballooning**

  Esta técnica emplea el ballooning y el hotplug para crear memoria física contigua en la VM a partir de memoria física fragmentada en la VM.

- **escape filter**

  Esta técnica permite la existencia de huecos en los segmentos directos, evitando que una única página defectuosa impida crear el segmento directo.

Índice

Monitor de máquina virtual (VVM)

Un monitor de máquina virtual es una capa de software que permite la creación, gestión, administración y ejecución de máquinas virtuales.

Ejemplos de VMM:

- KVM
- Xen
- VMware

Máquina virtual (VM)

Una máquina virtual es una abstracción de un entorno de cómputo completo a través de la virtualización combinada del procesador, la memoria y los componentes de E/S de una computadora.

Evolución de la virtualización

En las etapas tempranas de la virtualización, se utilizaban métodos puramente software.

A medida que la virtualización ganó popularidad, se introdujo el soporte de hardware.

La virtualización de las unidades de gestión de memoria siguió la misma progresión.

Shadow paging

El VMM construye la Shadow Page Table utilizando la información de la tabla de páginas del guest, que permite la traducción de gVA a gPA, y de la tabla de direcciones del host, que posibilita la traducción de gPA hPA.

La Shadow Page Table permite que en un fallo de la TLB se realice un page walk estándar de una dimensión.

Sin embargo, asegurar la coherencia de la Shadow Page Table ante cambios en las tablas de páginas del guest o del host implica costes sustanciales.

2D page walk

En la actualidad, la mayoría de los procesadores soportan el 2D page walk en hardware.

Sin embargo, este enfoque no es perfecto, ya que las referencias a la memoria de la tabla de páginas pueden aumentar hasta un total de 24.

Hacen falta 5 referencias para traducir la raíz y cada uno de los 4 niveles de la tabla de páginas del guest, más 4 referencias para obtener la dirección física final. En total: $5 \cdot 4 + 4 = 24$

2D page walk

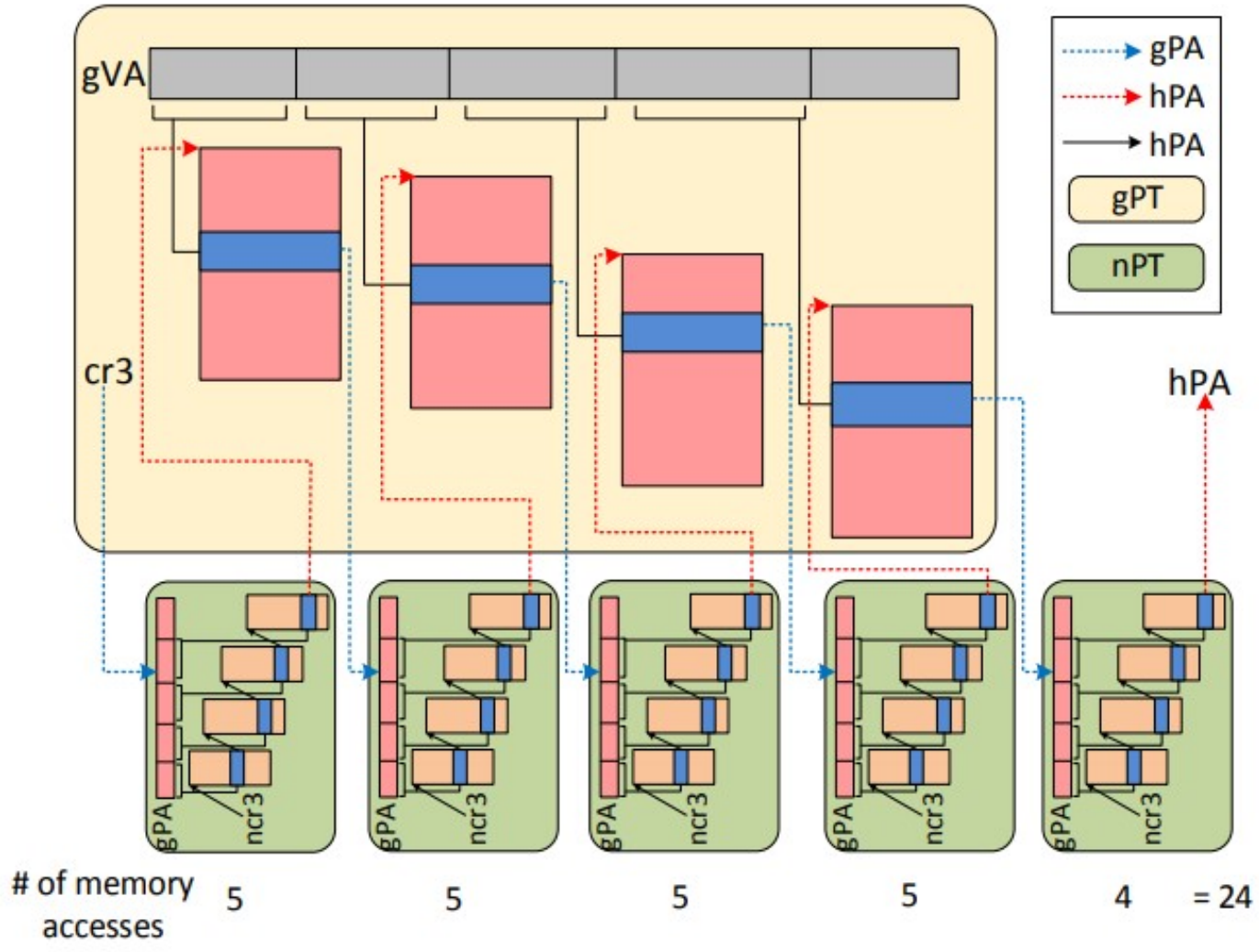

Figura: 2D page walk

Segmentos directos (Direct segments)

Los segmentos directos utilizan una forma de segmentación junto con paginación para eliminar en gran medida la sobrecarga de memoria virtual en cargas de trabajo con grandes requisitos de memoria.

Un segmento directo mapea un amplio rango de direcciones virtuales contiguas a direcciones físicas contiguas. Este mapeo se realiza utilizando solo tres registros, BASE, LIMIT y OFFSET, donde BASE y LIMIT representan el inicio y el fin del espacio de direcciones virtuales contiguas, y OFFSET es la diferencia entre las direcciones virtuales y físicas del segmento directo.

Segmentos directos (Direct segments)

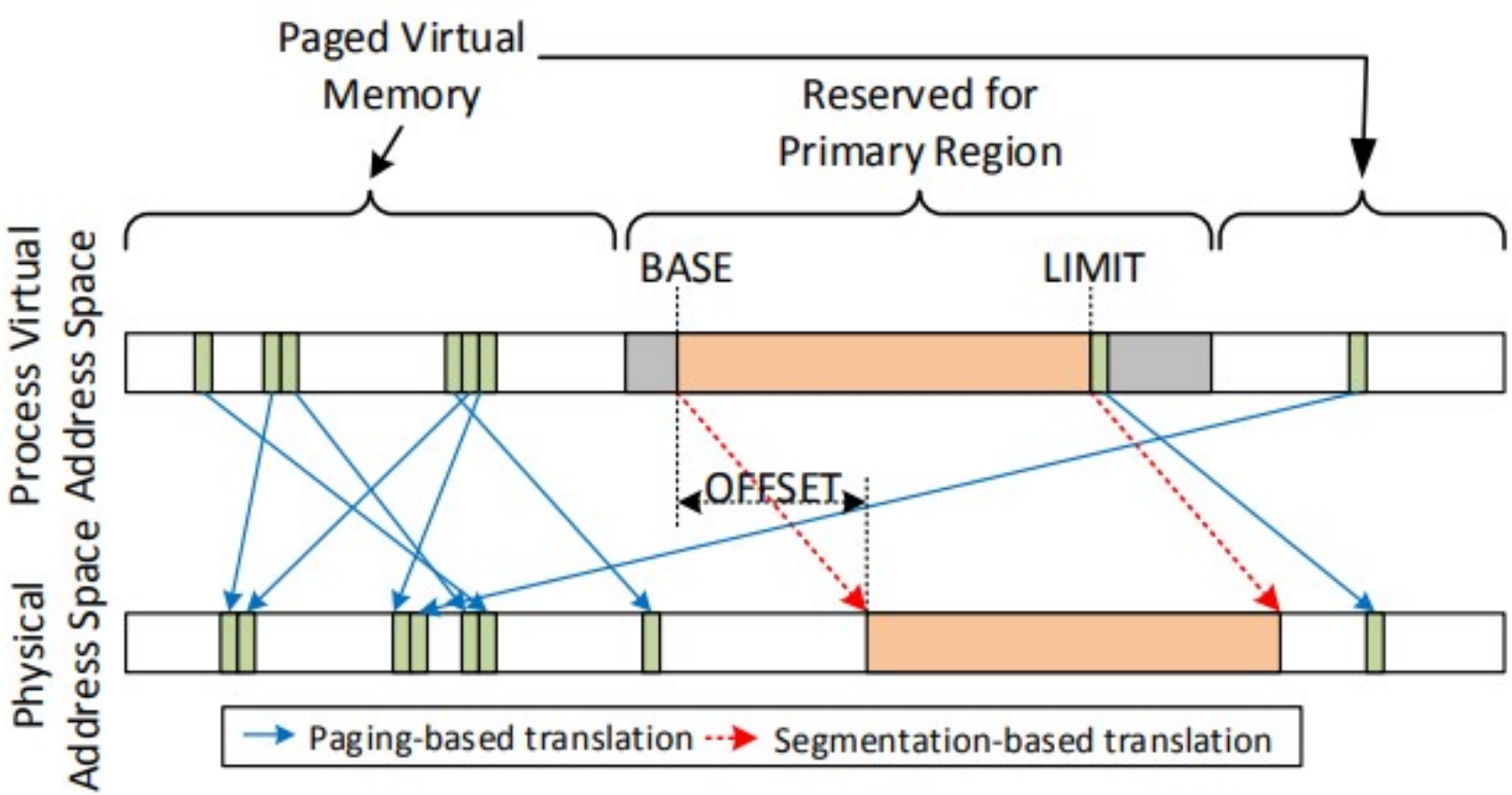

Figura: Espacio de direcciones utilizando un segmento directo.

Índice

Modos de traducción de direcciones

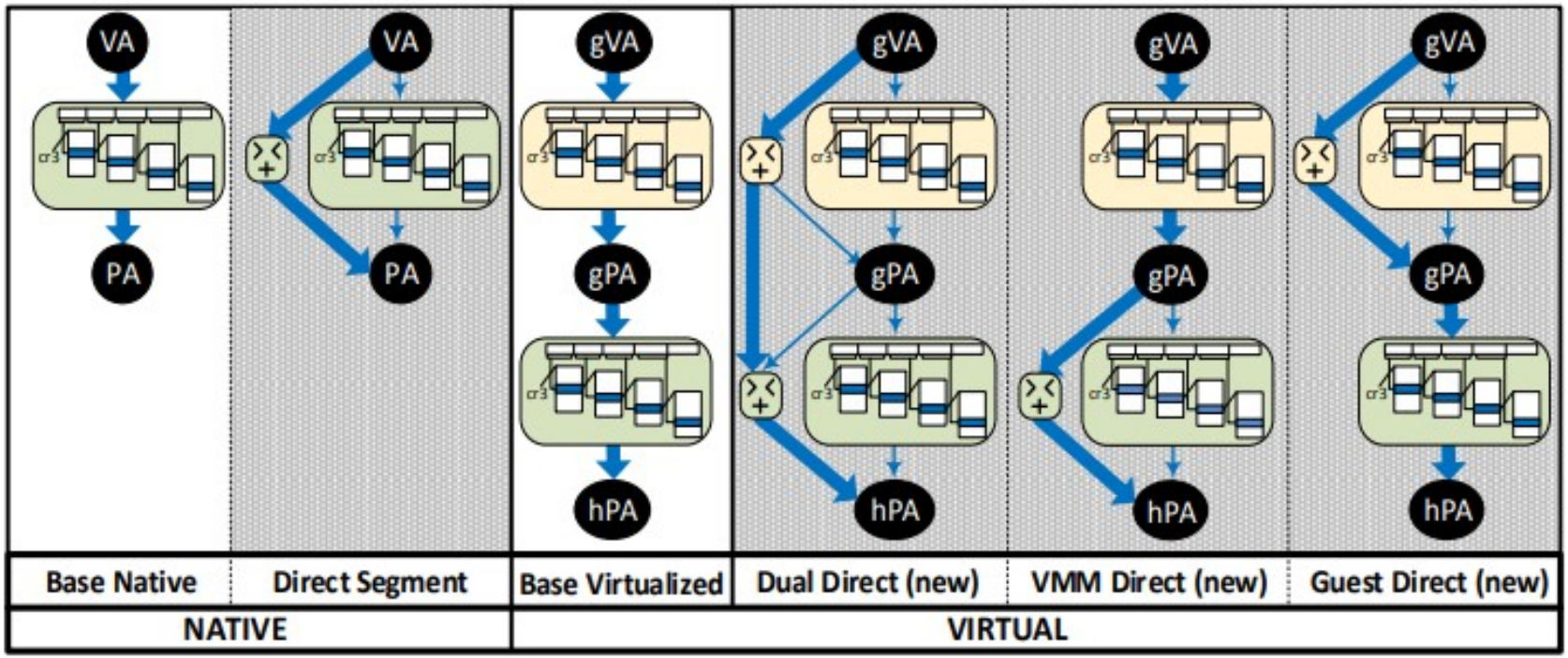

Figura: Modos de traducción de direcciones

Tenemos dos modos base (nativo y virtualizado), un modo de segmento directo no virtualizado (sombreado) y tres nuevos modos virtualizados (sombreados).

Registros:

- BASE_g, LIMIT_g, OFFSET_g
- BASE_v, LIMIT_v, OFFSET_v

El modo Dual Direct consulta ambos registros para eludir completamente el page walk.

Además, se realiza una modificación en el hardware del page walk para aplanar una o dos dimensiones la búsqueda, según el modo seleccionado.

Diseño hardware propuesto

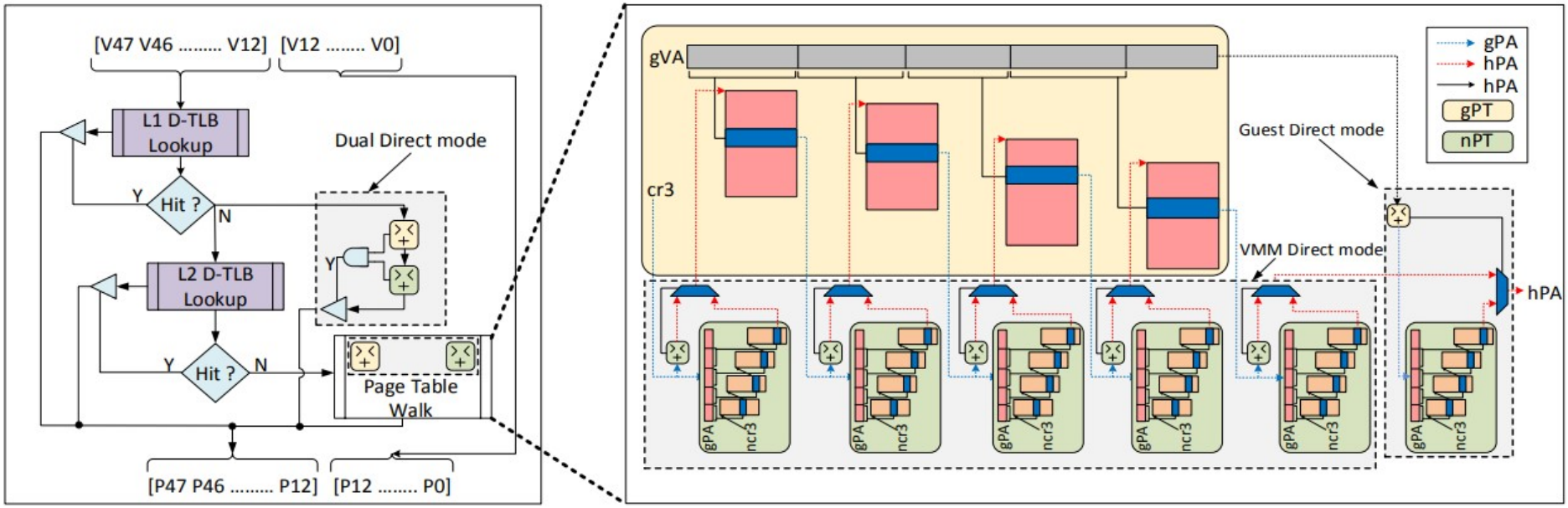

Figura: Diseño hardware propuesto

Dual Direct

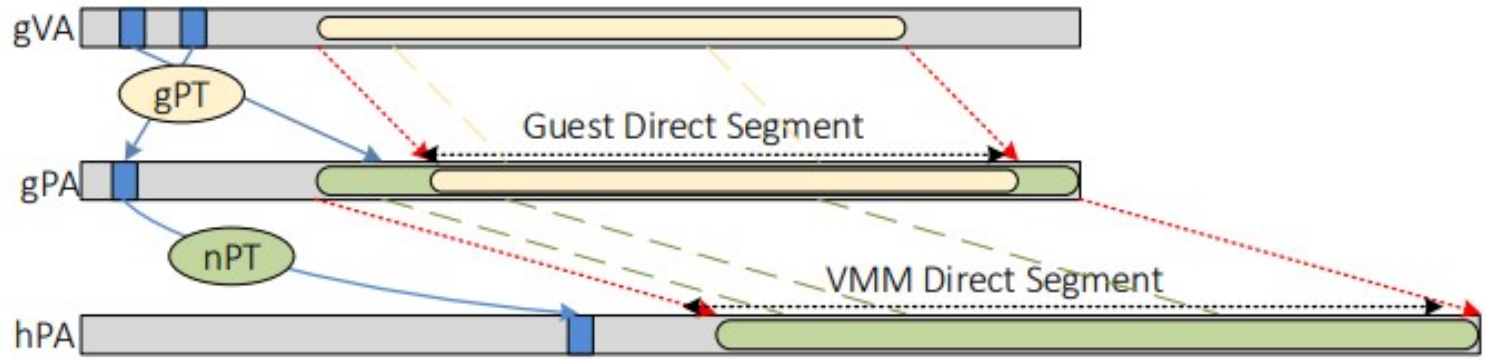

Figura: Estructura de memoria para el modo Dual Direct

El modo Dual Direct busca una page walk de dimensión cero. Utiliza dos niveles de segmentos directos: uno, llamado segmento del guest, para (la mayoría de) traducciones del primer nivel (gVA->gPA), y el otro, llamado segmento del VMM, para (la mayoría de) traducciones del segundo nivel (gPA->hPA).

VMM Direct

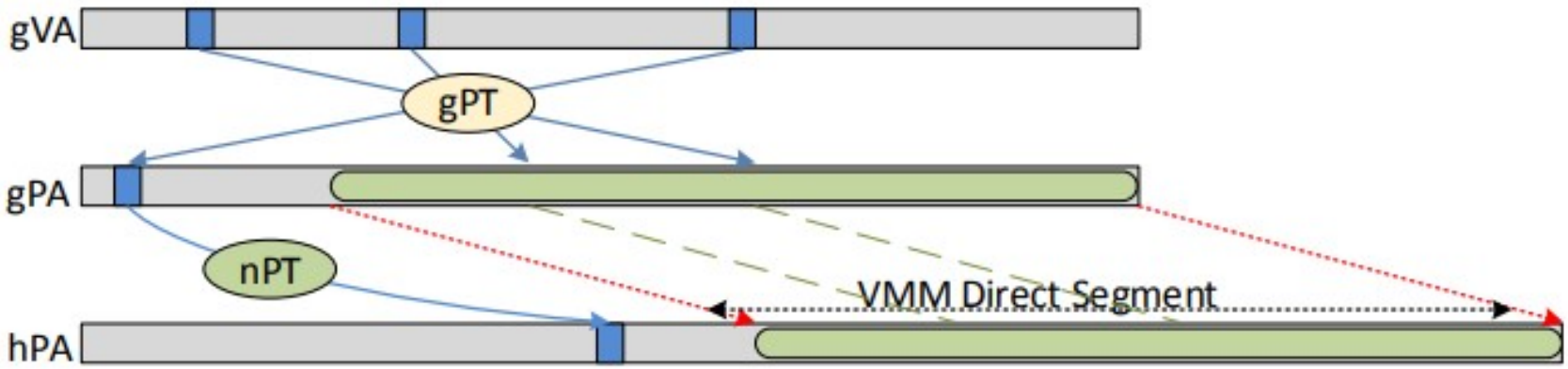

Figura: Estructura de memoria para el modo VMM Direct

El modo VMM Direct busca un page walk de una dimensión sin cambios en la aplicación ni en el sistema operativo invitado. Utiliza la paginación para el primer nivel de traducción de direcciones (gVA->gPA) y un segmento directo para (la mayoría de) el segundo nivel (gPA->hPA).

Guest Direct

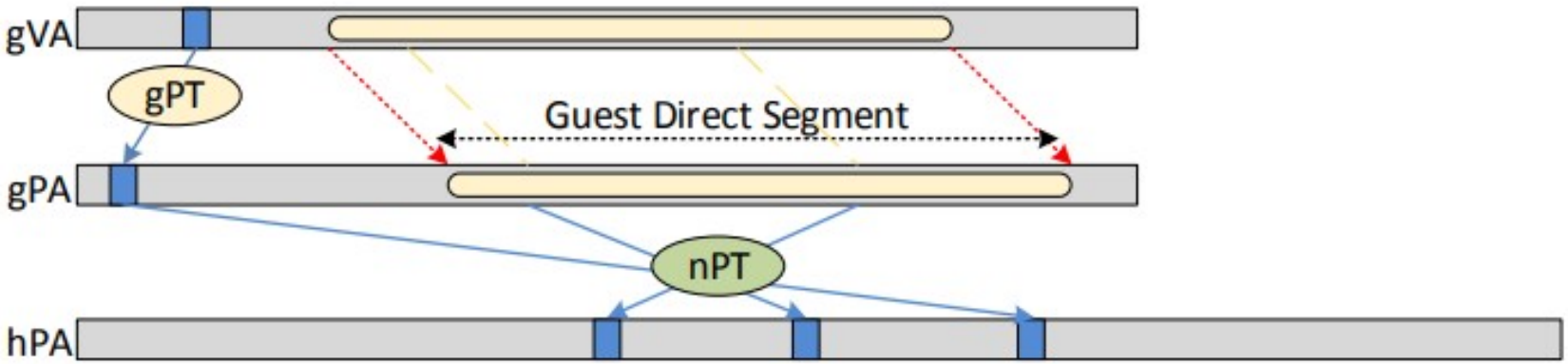

Figura: Estructura de memoria para el modo Guest Direct

El modo Guest Direct busca un page walk. Utiliza el segmento directo para (la mayoría de) traducciones del primer nivel (gVA->gPA) y paginación para el segundo nivel (gPA->hPA).

Índice

Fragmentación de memoria

La fragmentación de la memoria física del invitado y del host puede impedir la creación de segmentos

A continuación se presentan algunas formas de reducir la fragmentación de la memoria para permitir la creación de segmentos directos.

- Self-ballooning
- Reclaiming I/O gap memory
- Memory compaction

Self-ballooning

Para solucionar la fragmentación de la memoria física del guest y facilitar la creación rápida de segmentos del sistema operativo invitado, los autores proponen la técnica del self-ballooning.

El objetivo de esta técnica es proporcionar rápidamente memoria física contigua a partir de memoria física fragmentada, sin los costes asociados con la compactación de memoria.

Self-ballooning

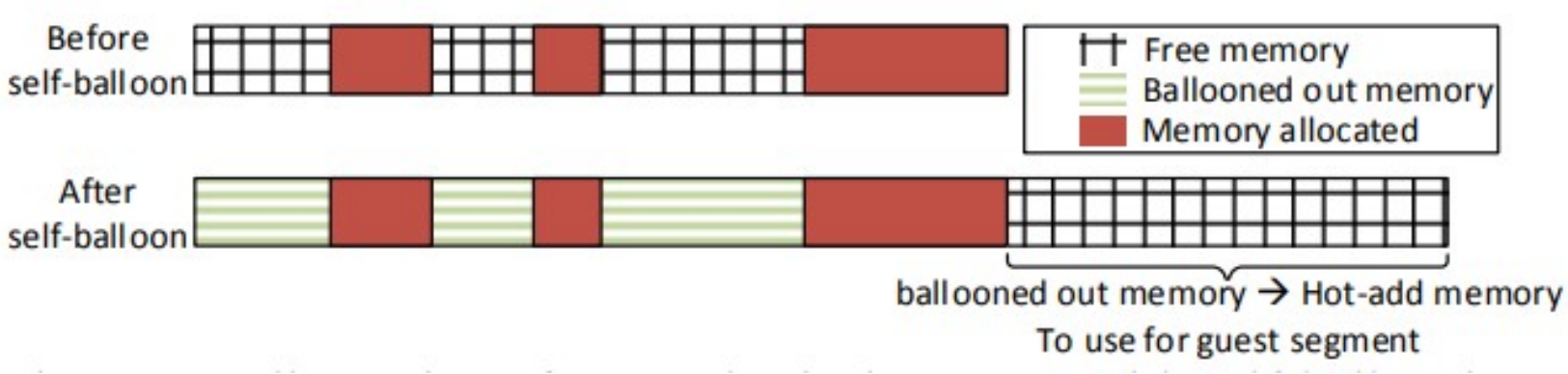

Figura: Self-ballooning

1. El balloon driver asigna la memoria no utilizada dentro del espacio de direcciones de la VM a un grupo de memoria reservada, de modo que no esté disponible para otros procesos en la VM.
2. El balloon driver pasa estas páginas al VMM, que utiliza el memory hotplug para agregar la misma cantidad de memoria nuevamente a la VM.

Problema: Fragmentación debido al I/O Gap en x86-64

- I/O Gap en x86-64: Espacio de 1 GB al final de la dirección física de 32 bits reservado para E/S mapeada en memoria.
- Problema: La presencia del I/O Gap divide las direcciones entre la región antes y después de este espacio, impidiendo un único segmento directo para toda la memoria física del invitado.

Solución: Hot-Unplug

- Hot-Unplug: Esta técnica posibilita la eliminación de la de la memoria física del invitado que precede al I/O Gap, al mismo tiempo que amplía la capacidad de memoria física del invitado en la misma cantidad después del I/O Gap.
- Beneficio: Permite que un solo segmento directo mapee casi toda la memoria física.

Compactación de memoria

- Para resolver el problema de la fragmentación de la memoria física del host, aprovechamos la técnica más lenta de compactación de memoria.
- La técnica de compactación de memoria reubica gradualmente las páginas y crea un segmento directo para el VMM.

Índice

Escape filter

- El escape filter es una nueva técnica propuesta por los autores para enfrentar el desafío de las páginas defectuosas al utilizar segmentos directos.
- El término "páginas defectuosas" se refiere a aquellas páginas de memoria que presentan algún tipo de error o fallo. Pueden deberse a diversas razones, como fallas en hardware, errores de lectura/escritura, etc.

Problema con las páginas defectuosas

- En sistemas operativos comunes, las páginas defectuosas no son un gran problema, generalmente se colocan en una lista de páginas defectuosas y simplemente se evita su uso.
- Sin embargo, con segmentos directos, una sola página defectuosa puede evitar la creación de un segmento directo considerable.

La solución: escape filter

Esta técnica permite la existencia de huecos en segmentos directos. Si una dirección está en un hueco, "escapa" de la traducción basada en segmentos y utiliza la paginación convencional.

En resumen, esta técnica aporta la capacidad de mantener segmentos extensos incluso cuando algunas de sus páginas presentan son defectuosas.

Índice

Metodología de la evaluación

- Problema: La larga duración de la carga de trabajo y el gran tamaño de la memoria utilizada provocan que realizar una simulación completa sea imposible.
- Solución: Implica por utilizar *performance counters* a nivel de hardware y realizar modificaciones en el Monitor de Máquina Virtual y en el kernel del sistema operativo invitado.
  - Se utilizan los *performance counters* para medir la cantidad de fallos en la TLB.
  - Se utilizan las modificaciones en el Monitor de Máquina Virtual y el kernel del sistema operativo invitado para conocer la dirección virtual del invitado y la dirección física del invitado para un fallo en la TLB.

Sobrecarga asociada con la memoria virtual

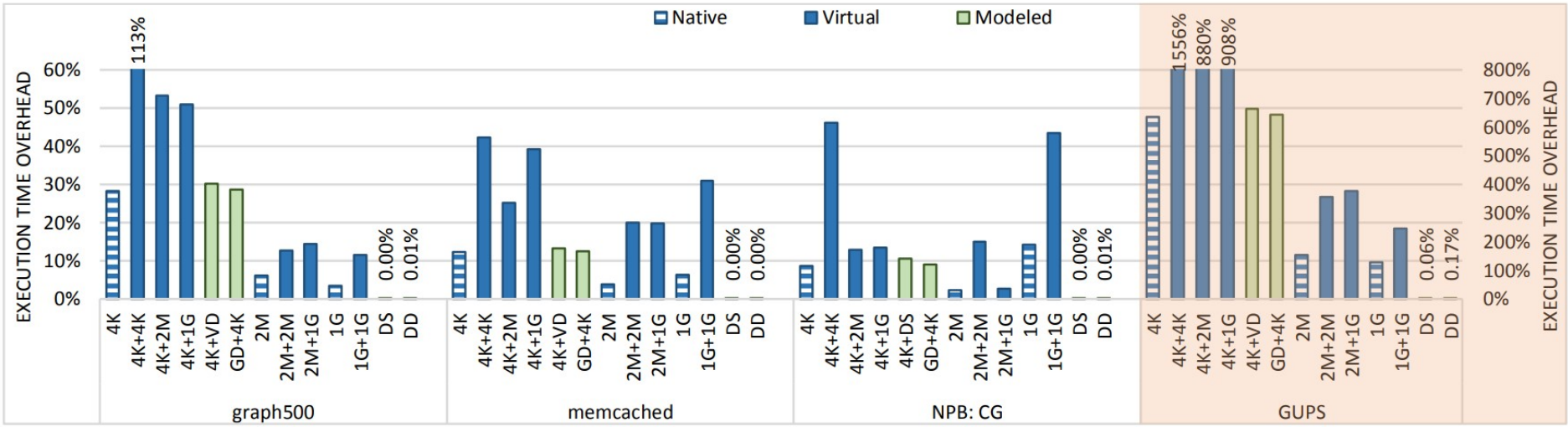

Figura: Sobrecarga asociada con la memoria virtual para las diferentes big-memory workloads y configuraciones seleccionadas

Sobrecarga asociada con la memoria virtual

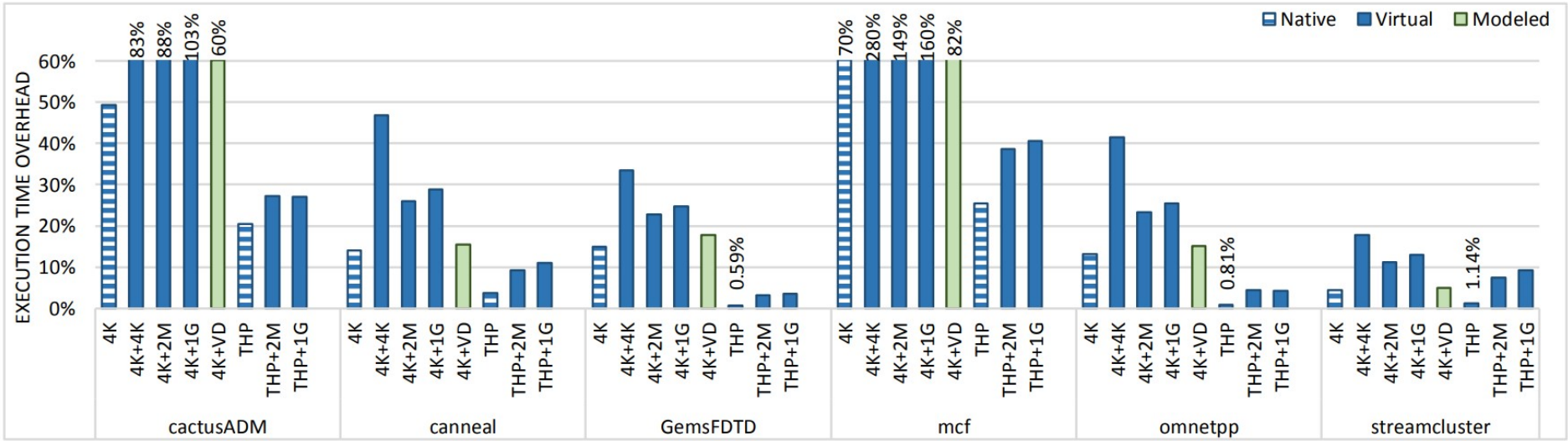

Figura: Sobrecarga asociada con la memoria virtual para las diferentes compute workloads y configuraciones seleccionadas

Impacto del nuevo HW en el consumo de energía

El nuevo hardware introduce un consumo adicional de energía.

Pero reduce el tiempo de ejecución y, por tanto, reduce la energía estática consumida por el sistema.

Los autores esperan que la disminución en el consumo estático compense el aumento derivado del nuevo hardware.

Índice

Evaluación del artículo

Este artículo parece ser una continuación del trabajo presentado por los mismos autores hace un año.

Es más, pienso que los autores se percataron que este nuevo artículo no alcanzaba el mismo nivel que el anterior.

En un intento de mejorar el artículo, los autores han introducido conceptos adicionales; algunos de estos, como el self-ballooning, aportan coherencia al tema, mientras que otros, como el escape filter, podrían haberse integrado mejor en el artículo anterior.

Además, la presencia de excesivas pruebas da la impresión de ser un añadido con el único propósito de rellenar.

Contribución única y original

En cuanto a la originalidad, el artículo presenta limitaciones.

La idea de los segmentos directos ya había sido propuesta anteriormente; aquí, simplemente se aplica a la virtualización.

Además, el desarrollo del hardware ya fue abordado en el artículo previo, y esta entrega es una adaptación más que una contribución completamente nueva.

Coherencia y organización

La inclusión de conceptos adicionales en el artículo puede dar la sensación de desorganización y puede distraer.

Puede desviar la atención del lector de la verdadera idea propuesta.

Justificación de las nuevas técnicas?

La introducción del self-ballooning parece estar justificada en el contexto del artículo actual, ya que aborda específicamente la fragmentación de la memoria física en el guest.

Sin embargo, la inclusión del escape filter podría ser cuestionada, ya que podría haber sido más pertinente en el artículo anterior.

Calidad de las pruebas

Se han llevado a cabo numerosas pruebas de rendimiento; sin embargo, los resultados no respaldan de manera concluyente la propuesta de los autores.

Esta falta de respaldo sustancial da la sensación de que estas pruebas se han incluido con el propósito de rellenar el artículo, más que para fortalecer la propuesta indicada.

Evaluación del artículo

Aunque no alcanza el mismo nivel que el artículo del año pasado, este trabajo sigue ofreciendo una contribución original y significativa al campo.

A pesar de algunas dudas sobre la necesidad de ciertas pruebas, la metodología empleada es rigurosa y apropiada.

La revisión del artículo no solo abarca el contenido científico, sino también la gramática y ortografía, demostrando un cuidado en la presentación del trabajo.

La estructura del artículo se mantiene lo más clara y lógica posible a pesar de la complejidad introducida por la variedad de conceptos presentados.

Se citan y se utilizan referencias de manera precisa y actualizada.

Hay una gran cantidad de figuras e imágenes que facilitan la compresión al lector.

Conclusión de la evaluación

En resumen, este paper es un trabajo muy sólido, respaldado esto por su publicación en la *International Symposium on Microarchitecture* (MICRO), una conferencia de gran nivel.

Sin embargo, es importante señalar que no alcanza el mismo nivel que el artículo del año anterior, el cual fue presentado en la conferencia *Architectural Support for Programming Languages and Operating Systems* (APLOS), una conferencia que se sitúa un escalón por encima en términos de reconocimiento y prestigio.