

Tema 13: Sistema de Entrada/Salida

Arquitectura de Computadoras

Ing. Nicolás Majorel Padilla (npadilla@herrera.unt.edu.ar)

<http://microprocesadores.unt.edu.ar/arqcom/>

¿Cuáles temas veremos?



- ▶ Primera parte: **Generalidades**
 - ▶ Introducción. Conceptos generales.
 - ▶ *Performance, Price & Power.*
 - ▶ Conceptos de **Paralelismo** y *Pipelining*.
- ▶ Segunda parte: **Procesadores**
 - ▶ ISA: qué es, cómo se diseña.
 - ▶ Diseños de procesadores para *performance*.
- ▶ Tercera parte: **Subsistema de Memoria**
 - ▶ Memoria Caché.
 - ▶ Memoria Virtual.
- ▶ Cuarta parte: **Sistema completo**
 - ▶ Subsistemas de Entrada/Salida y de Comunicaciones.
 - ▶ Consumo de energía.
- ▶ Quinta parte: **Nuevos paradigmas**
 - ▶ Arquitecturas específicas.
 - ▶ Seguridad.

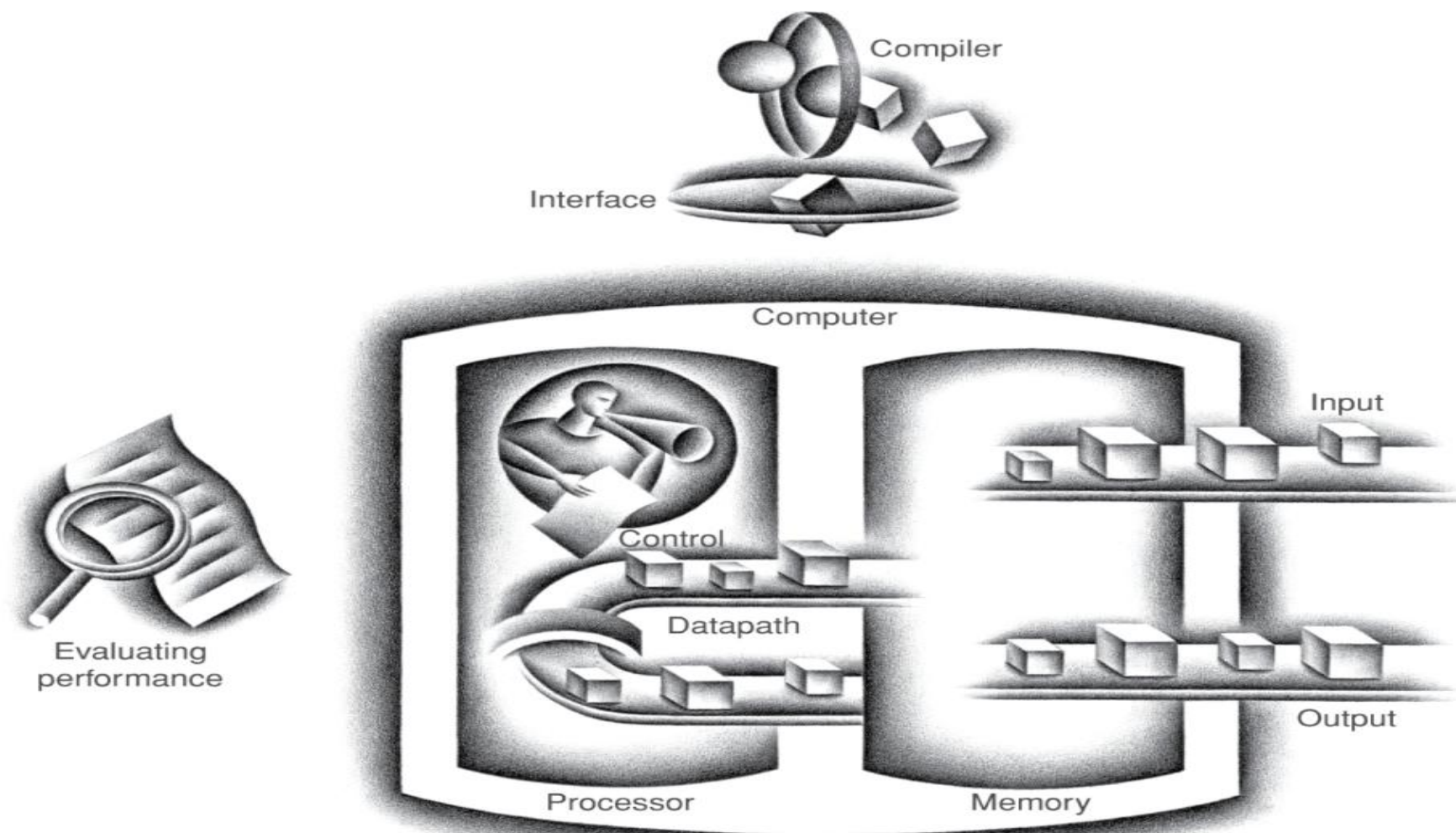
Temas que veremos

- ▶ Esquema básico del Sistema de I/O.
- ▶ Rol del Sistema Operativo.
- ▶ Delegación de la responsabilidad de I/O del CPU.
- ▶ Métricas de Performance en I/O.
- ▶ Ejemplos de dispositivos de I/O:
 - ▶ Discos magnéticos y de estado sólido.
- ▶ Nuevas tecnologías y tendencias.

Lectura recomendada

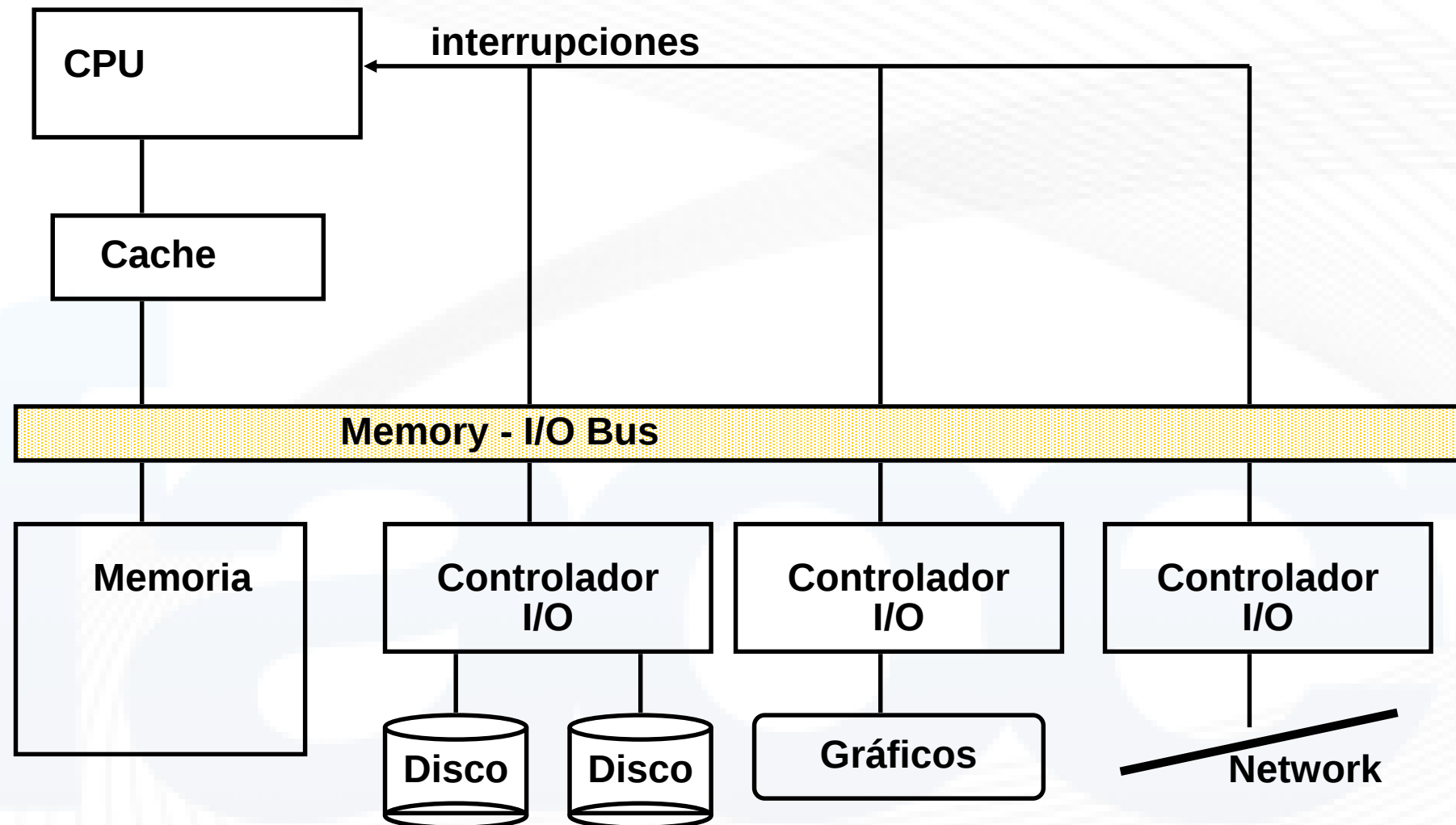
- ▶ Computer Organization and Design, RISC-V Edition (2da ed, 2021):
 - ▶ Sección 1.4: *Under the Covers*
 - ▶ Sección 5.2: *Memory Technologies*
 - ▶ Sección 5.5: *Dependable Memory Hierarchy*
 - ▶ Sección 5.11: *Redundant Array of Inexpensive Disks*
 - ▶ Sección 6.10: *Communicating to the Outside World – Cluster Networking*

Repaso: Partes de una computadora



- ▶ Ya vimos sobre procesadores, sus caminos de datos y su control, y sobre el sistema de memoria.
- ▶ Ahora pasamos el foco al Sistema de Entrada y Salida.

Esquema básico del Sistema de I/O



- ▶ Existencia de **múltiples dispositivos**, con características y comportamientos **muy diferentes**, conectados a uno o más buses.
- ▶ En este tema nos enfocaremos en aspectos globales de I/O y en ejemplos de dispositivos. En el siguiente, en las conexiones.

Responsabilidades del SO

- ▶ El Sistema Operativo es la interface entre el hardware de I/O y el programa de usuario que requiere una operación de I/O.
- ▶ Sus responsabilidades surgen de 3 características:
 - ▶ El sistema de I/O es compartido por múltiples programas usando el procesador (**multiprogramación**).
 - ▶ El sistema de I/O usa generalmente **interrupciones** para comunicar información de la operación de I/O.
 - ▶ Y las interrupciones deben ser manejadas por el SO, ya que se ejecutan en **modo supervisor**.
 - ▶ El control de bajo nivel de los dispositivos de I/O es complejo.
 - ▶ Porque debe manejar eventos concurrentes.
 - ▶ Porque los requisitos son muy específicos del dispositivo en cuestión.

El SO deberá proveer...

- ▶ **Protección** de recursos de I/O compartidos.
 - ▶ Garantizar que un programa de usuario sólo acceda a los recursos a los que tiene permisos.
- ▶ **Abstracción** en el uso de dispositivos de I/O.
 - ▶ Proveer rutinas que manejen la operación de bajo nivel de los distintos dispositivos.
- ▶ **Manejo de interrupciones** generadas por los dispositivos de I/O.
- ▶ **Acceso equitativo** a los recursos de I/O compartidos.
 - ▶ Todos los programas de usuario deben tener un buen servicio.
- ▶ **Organización** de los accesos a los recursos de I/O compartidos.
 - ▶ Para mejorar la productividad total del sistema.

Drivers de dispositivos

- ▶ El SO provee todas las características anteriores incorporando porciones de software que sirven de interfaz con cada dispositivo específico.
- ▶ Estos pedazos de software son conocidos como “drivers”.
 - ▶ Debería haber uno por cada dispositivo que se pretende conectar al sistema.
- ▶ El driver se encarga de copiar datos desde el espacio de direcciones del programa de usuario al espacio de direcciones del SO.
- ▶ Y también de manejar la comunicación entre el SO y el dispositivo.

Requisitos de comunicación

- ▶ El SO debe ser capaz de comunicarse con el sistema de I/O.
 - ▶ Y al mismo tiempo evitar que los programas de usuario se comuniquen directamente con el sistema de I/O. *¿Por qué?*
- ▶ Tres tipos de comunicación:
 - ▶ El SO debe ser capaz de enviar comandos al dispositivo de I/O.
 - ▶ El dispositivo de I/O debe ser capaz de notificar al SO cuando haya completado una operación (o si ocurrió un error).
 - ▶ Se deben transferir los datos entre el dispositivo de I/O y la Memoria principal.

Comunicación 1: SO → I/O

- ▶ Dos métodos son los más usados.
- ▶ **Instrucciones especiales de I/O.**
 - ▶ Especifican el dispositivo y el comando a enviar.
 - ▶ Ej: SEND(id_disco1, dato)
 - ▶ El identificador de dispositivo suele enviarse por líneas de control.
 - ▶ El comando suele enviarse por líneas de datos.
- ▶ **I/O mapeada en memoria.**
 - ▶ Se asignan áreas del espacio de memoria a dispositivos de I/O.
 - ▶ Los accesos a estas direcciones son interpretados como comandos por los dispositivos.
 - ▶ Ej: sw x10, 0x800(zero)

Comunicación 1: SO → I/O

- ▶ Las instrucciones especiales tienen la ventaja que directamente se ejecutan en modo supervisor.
 - ▶ A pesar de ello, no son muy populares.
- ▶ *¿Por qué es muchísimo más común la I/O mapeada en memoria?*
 - ▶ Porque permite usar el **mismo mecanismo de protección que ya se implementó para Memoria Virtual**.
 - ▶ Los programas de usuario no tienen acceso a estas direcciones.

Comunicación 2: I/O → SO

- ▶ También son dos los métodos más usados.
- ▶ **Polling (encuestas)**
 - ▶ Cada dispositivo de I/O se asocia a un registro de estado (mapeado en memoria o no).
 - ▶ El dispositivo pone información en este registro.
 - ▶ El SO periódicamente revisa este registro para ver si hay datos listos, o si ocurrió un error.
 - ▶ Si los datos están listos, leerlos y continuar.
 - ▶ Si no, permanecer en el lazo e intentar nuevamente más tarde.
 - ▶ **Ventajas:** simplicidad, total control sobre la operación.
 - ▶ **Desventaja:** mucho overhead, puede consumir muchos ciclos de CPU al revisar y que no haya modificaciones.

Comunicación 2: I/O → SO

► Interrupciones

- ▶ El procesador está ocupado en sus asuntos hasta que un dispositivo está listo y requiere su atención mediante una interrupción.
- ▶ El procesador guarda el estado (PC y otros registros) y salta a la rutina de servicio de interrupción correspondiente.
- ▶ Al terminar, continúa con lo que estaba haciendo.
- ▶ **Ventaja:** eficiencia, el programa de usuario sólo se detiene durante la transferencia.
- ▶ **Desventajas:** requiere hw especial en el dispositivo para generar la interrupción, y además en el procesador para detectar la interrupción y poder guardar el estado automáticamente.

¿Polling o Interrupciones?

- ▶ Polling es más rápido:
 - ▶ El compilador sabe cuáles registros se usan en la encuesta, no necesita guardar estado.
 - ▶ Se evitan overheads innecesarios (vaciado de pipeline, prioridades, etc).
- ▶ Polling es más lento:
 - ▶ El overhead de la encuesta ocurre haya o no datos listos.
 - ▶ Un dispositivo podría esperar mucho tiempo hasta que es atendido (por tener muchos dispositivos, por hacer cálculos intermedios).
- ▶ Como casi siempre, hay un compromiso:
 - ▶ Eventos frecuentes y regulares, con polling.
 - ▶ En otro caso, interrupciones.

Comunicación 3: I/O → Memoria

- ▶ Para enviar los datos del dispositivo de I/O a Memoria, sería ideal evitar que el procesador pierda tiempo.
- ▶ Conviene delegar la transferencia en un **controlador DMA** (*Direct Memory Access*).
 - ▶ Externo al procesador, toma posesión del bus de memoria.
 - ▶ Transfiere datos de a bloques, sin intervención del procesador.
 - ▶ El procesador envía la dirección inicial del bloque a transferir, el dispositivo en cuestión, la longitud del bloque, y un comando para que empiece.
 - ▶ El DMA toma control del bus y genera todas las señales correspondientes para hacer la transferencia.
 - ▶ Al terminar, genera una interrupción al procesador.

Comunicación 3: I/O → Memoria

- ▶ Si al controlador DMA se le agregan funciones adicionales, se convierte en un **IOP (*Input Output Processor*)**.
 - ▶ El procesador emite una instrucción al IOP donde le indica cuál es el dispositivo objetivo, y cuál es la dirección donde están el resto de los comandos.
 - ▶ El IOP busca sus propios comandos y realiza todas las transferencias a memorias necesarias, siempre en segundo plano.
 - ▶ Al terminar todas las transferencias, interrumpe al procesador principal.
 - ▶ Al igual que el DMA, el IOP “roba ciclos” al procesador: accede a memoria cuando no la usa el procesador.
- ▶ Hoy en día, este modelo es bastante usado. **¿Dónde?**

Consideraciones de Diseño de I/O

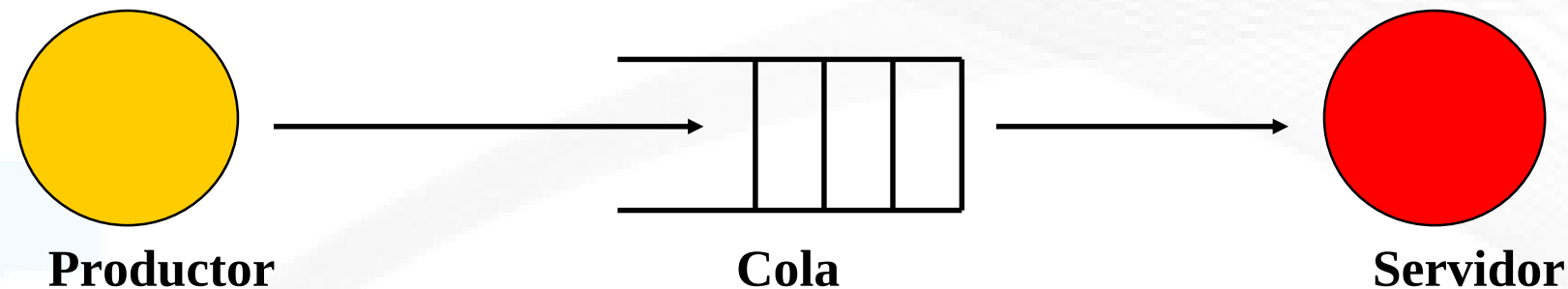
- ▶ Siempre, lo primordial es **performance**.
- ▶ Pero ahora entran en consideración nuevos factores:
 - ▶ **Escalabilidad**.
 - ▶ ¿Habrá algún problema si conectamos un nuevo disco al sistema?
 - ▶ **Tolerancia a fallos**.
 - ▶ Si falla una placa de red, ¿afectará al resto de la red?
- ▶ Dos métricas más que veremos en instantes.

Performance de I/O

- ▶ Depende de múltiples factores, por lo que no es fácil de medir:
 - ▶ El procesador y su jerarquía de memoria (cachés, mem. principal).
 - ▶ Dispositivos de I/O, sus controladores y su firmware.
 - ▶ Interconexiones.
 - ▶ La eficiencia con que se usan los dispositivos de I/O.
- ▶ Hay métricas específicas para distintos tipos y usos de computadoras.
 - ▶ Bases de Datos y transacciones: www.tpc.org
 - ▶ Sistemas web: SPECWeb.
 - ▶ Supercomputadoras: ExaFLOPs.
- ▶ Dos métricas comunes (las de siempre):
 - ▶ Productividad: **Ancho de banda** – [MB/s] u [Ops/s]
 - ▶ Latencia: **Tiempo de respuesta** – [s]

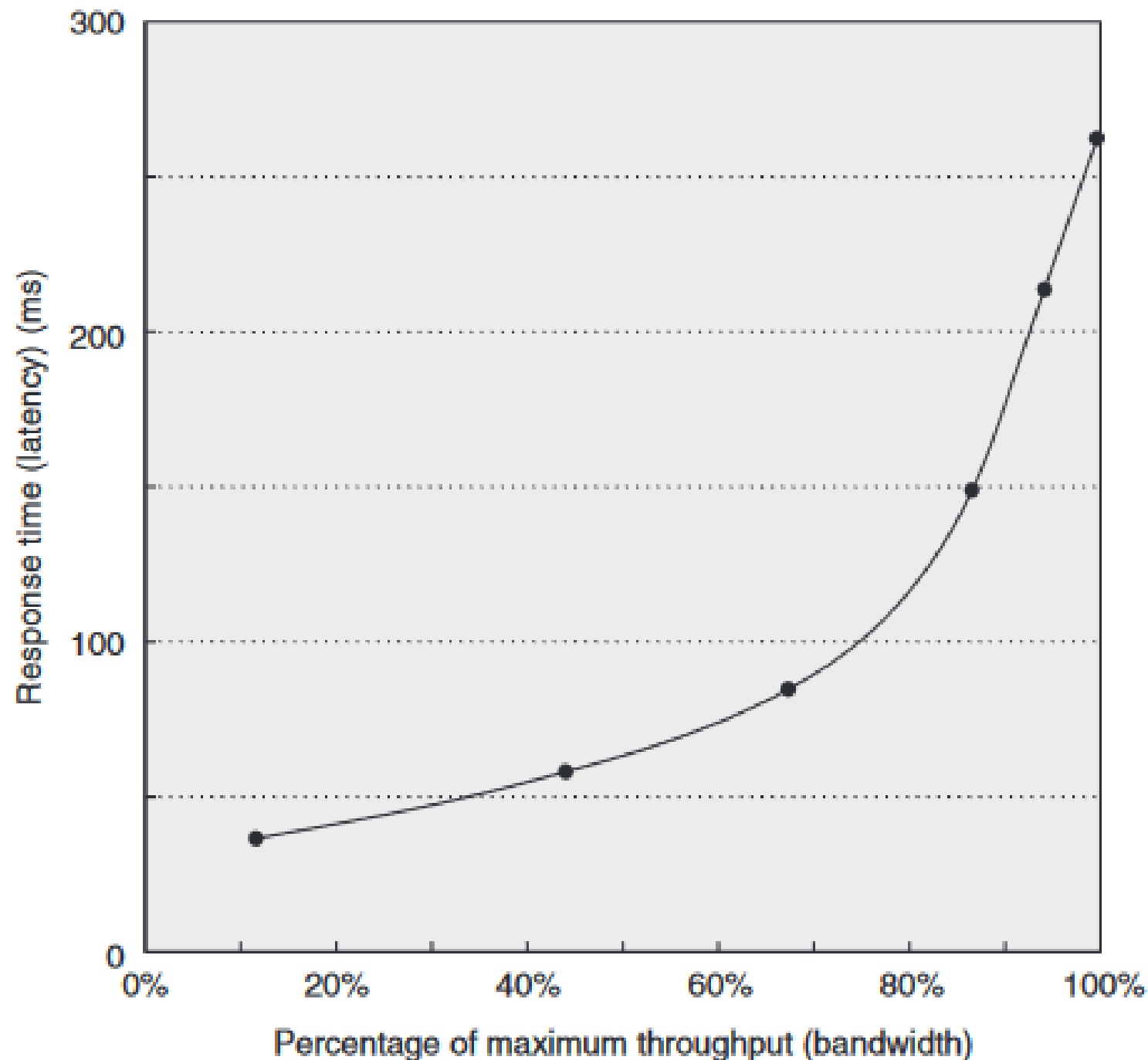
Latencia vs. Productividad

- ▶ La productividad y la Latencia están relacionadas mediante el modelo productor-consumidor.



- ▶ La productividad es el número de operaciones que completa el Servidor por unidad de tiempo.
 - ▶ Idealmente, el Servidor no debería detenerse nunca.
 - ▶ Y para ello, la cola no debería estar nunca vacía.
- ▶ La Latencia comienza cuando una tarea se pone en la cola, y finaliza cuando el Servidor la termina.
 - ▶ Idealmente, la cola debería estar vacía.
 - ▶ Así el Servidor puede responder inmediatamente.

Latencia vs. Productividad



- ▶ Nótese que con latencia mínima, la productividad es de un 10% aprox.
- ▶ Y con productividad máxima, la latencia es... ¡7 veces mayor!
- ▶ Nuevamente, un compromiso.
 - ▶ Se intenta que ningún recurso compartido esté ocupado más del 70-80% del tiempo.

Latencia vs. Productividad

- ▶ La Productividad de I/O puede mejorarse sin sacrificar la latencia.
 - ▶ En general, agregando más hardware.
 - ▶ Ej: poniendo otro servidor.
- ▶ Sin embargo, la latencia es mucho más difícil de reducir.
 - ▶ A fin de cuentas, está limitada por tecnología.
- ▶ Agregar hardware para mejorar productividad, nos conduce a tener en cuenta dos factores adicionales.

Confiabilidad y Disponibilidad

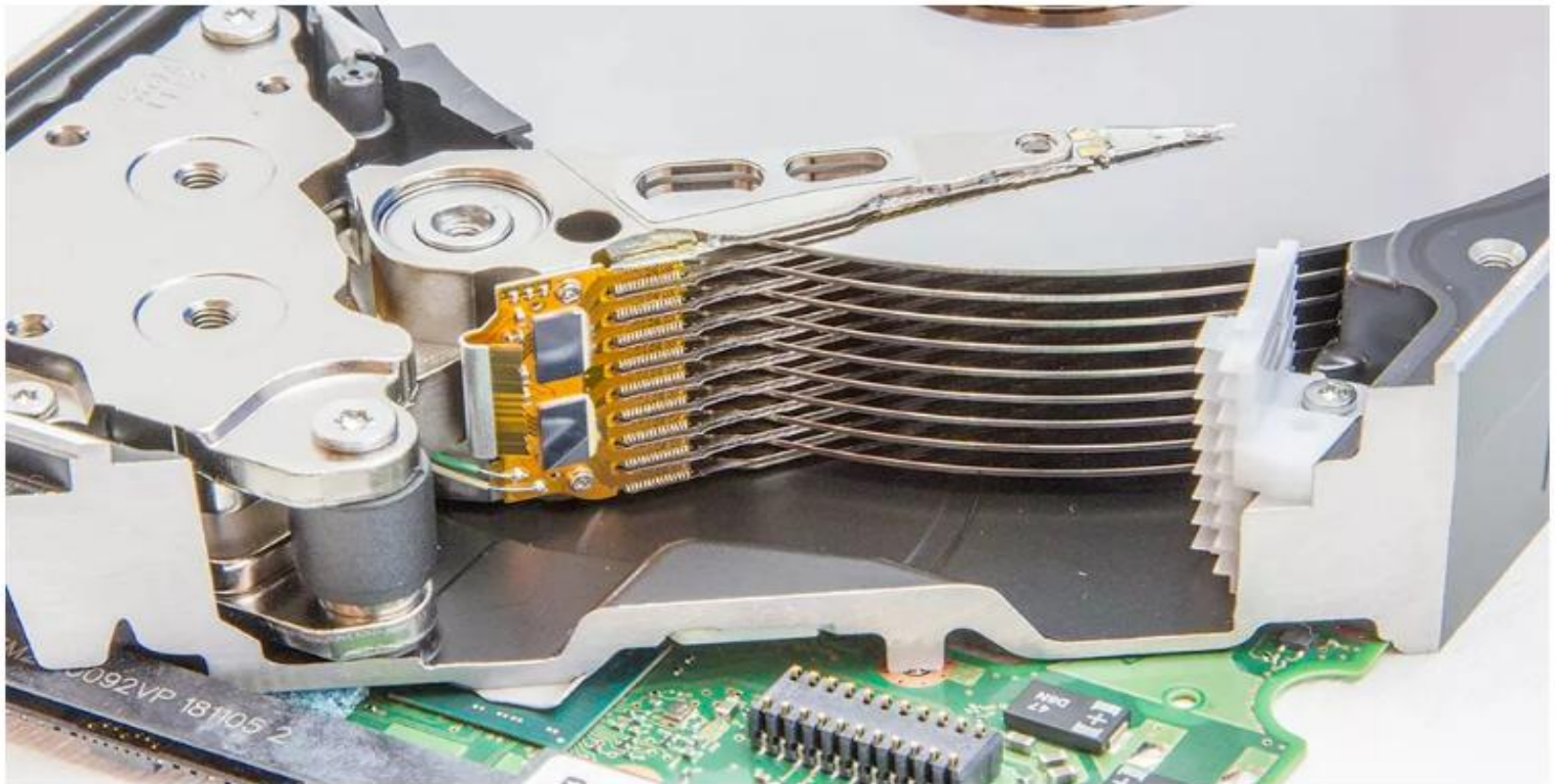
- ▶ **Confiabilidad (*reliability*)**: Tiempo hasta que se produce una caída del servicio.
 - ▶ MTTF (Mean Time To Failure), medida en años.
 - ▶ AFR (Annual Failure Rate).
- ▶ **Disponibilidad (*availability*)**: porcentaje de tiempo que el servicio está disponible.
 - ▶ MTTR (Mean Time To Repair), medida en horas.
 - ▶ $D = \text{MTTF} / (\text{MTTF} + \text{MTTR})$
 - ▶ Usualmente, se mide en “cantidad de nueves”. Ej: 99,99%.
- ▶ Normalmente, la disponibilidad se mejora agregando hardware.
 - ▶ Pero eso suele disminuir la confiabilidad.
 - ▶ Ej: un avión con dos motores.
- ▶ La confiabilidad sólo se mejora con mejores componentes, o con menos componentes.

Dispositivos de I/O: Discos Magnéticos



- ▶ En general, son el último nivel de la Jerarquía de Memoria.
 - ▶ El último dispositivo electromecánico que aún se usa.
- ▶ Propósito: almacenamiento no volátil de larga duración.
- ▶ Son grandes, baratos (\$/byte) y lentos.
- ▶ Desde 1980, su capacidad aumentó 10^6 x y su tiempo de acceso sólo 10x.

Detalles constructivos

- ▶ Placas rotatorias rígidas recubiertas de una superficie magnética.
 - ▶ Cada placa puede almacenar datos en ambas superficies.
 - ▶ Cada superficie tiene vinculado un cabezal para lectura/escritura.
 - ▶ El cabezal se ubica en el extremo de un brazo mecánico móvil que mueve todos los cabezales juntos, radialmente.
- 
- (Image credit: Western Digital)
- ▶ Cada superficie se divide en anillos concéntricos que se denominan **pistas**.
 - ▶ Cada pista está dividida en **sectores**, que son la mínima unidad de transferencia de los discos.
 - ▶ Cada sector contiene un número fijo de bytes (Ej: 4096 B, desde 2011).
 - ▶ Las pistas exteriores poseen más sectores (**densidad de grabación constante**).
 - ▶ Para los más curiosos: <https://www.edn.com/peeking-inside-an-hdd/>

Operación de lectura/escritura

- ▶ El brazo se mueve hasta ubicar el cabezal sobre la pista deseada. Esto se denomina *seek*.
 - ▶ El tiempo promedio necesario para el posicionamiento se denomina **seek time** y tiene una duración aproximada de **3-4 ms**.
- ▶ Una vez posicionado el cabezal sobre la pista, se espera que el disco gire hasta que el sector deseado se ubique bajo el cabezal. Esto se denomina **latencia rotacional**.
 - ▶ A 10.000 rpm, son **6 ms**, aunque en promedio, se considera media vuelta.
- ▶ Finalmente, se lee/escribe el sector, lo cual demora un cierto **tiempo de transferencia**.
 - ▶ Con una tasa de transferencia de 100 MB/s, tiene una duración aproximada de **40 us**.

Tiempo de Acceso de Discos Magnéticos

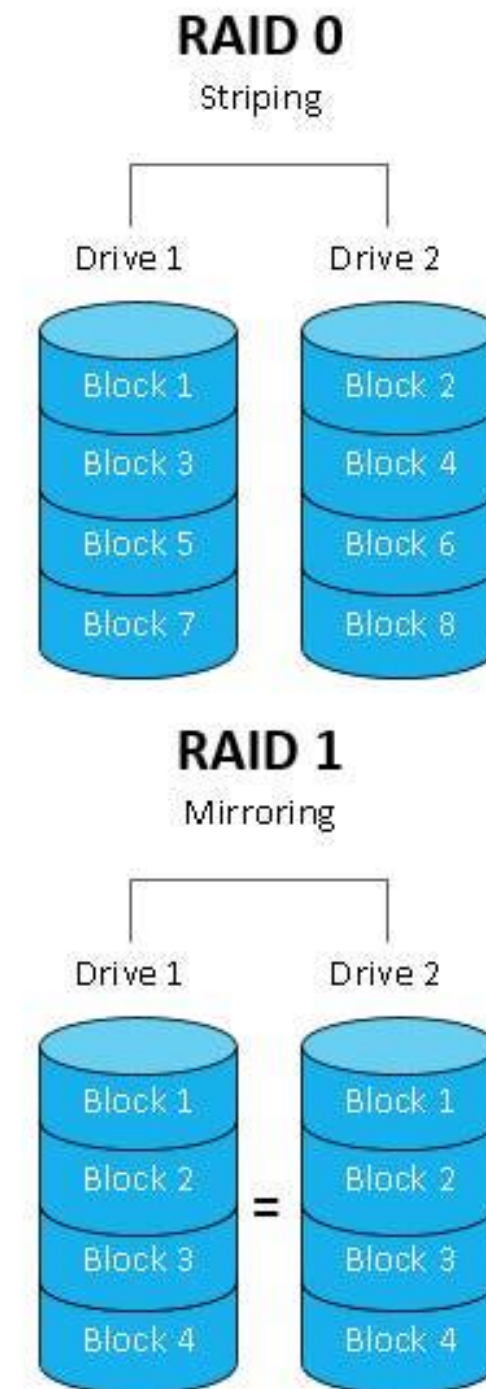
- ▶ **AMAT(Disco) = seek time + latencia + transferencia + retardo del controlador**
 - ▶ El retardo del controlador ronda típicamente entre 100 y 200 us.
- ▶ *¿Qué les parece este AMAT comparado con todos los que venimos viendo en temas anteriores?*
- ▶ *¿Cuántos ciclos serían para un microprocesador con una frecuencia de reloj de 3 GHz?*
- ▶ *¿Cuáles son los componentes del cálculo del AMAT que más influyen en el resultado?*
- ▶ Como el tiempo de transferencia es mucho menor que el de seek y el de latencia suelen ponerse varios discos en simultáneo para aumentar la productividad.

RAID de Discos Magnéticos

- ▶ Para aumentar la productividad de los discos magnéticos se propuso organizarlos en forma de un arreglo.
 - ▶ Muchos discos pequeños y baratos, con los datos distribuidos entre ellos.
 - ▶ De esta manera, se puede acceder a los discos en paralelo o en *interleaving*, aumentando así la productividad.
 - ▶ Pero al aumentar la cantidad de discos, baja la confiabilidad.
- ▶ Para que la baja de confiabilidad no produzca una baja de disponibilidad, se agrega **redundancia** a los datos.
 - ▶ Si algún disco falla, la información puede ser reconstruida.
 - ▶ Esta terminó siendo la principal razón de su popularidad.
- ▶ RAID: ***Redundant Array of Inexpensive Disks***
 - ▶ Varias configuraciones posibles, para distintos casos de uso.

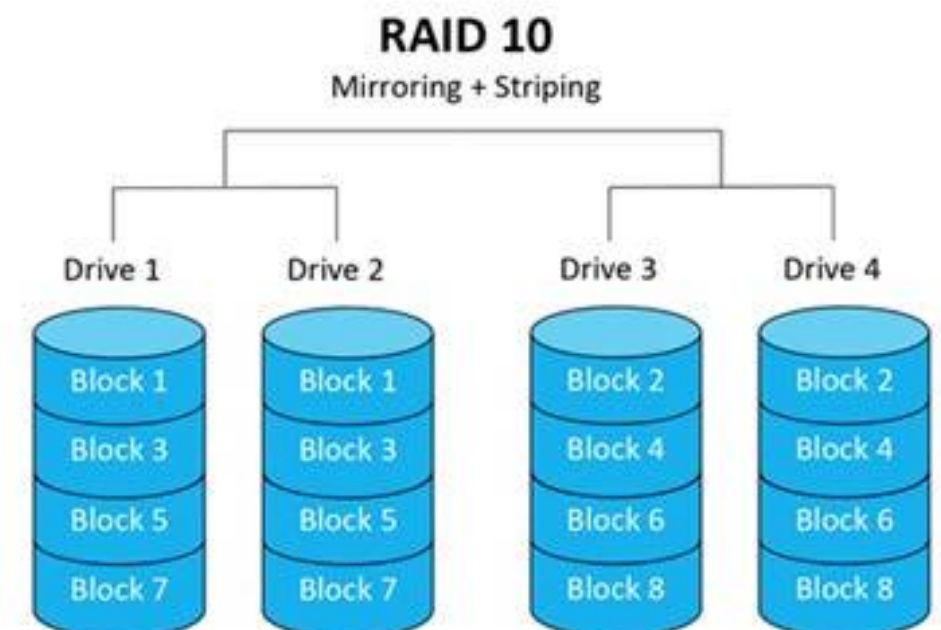
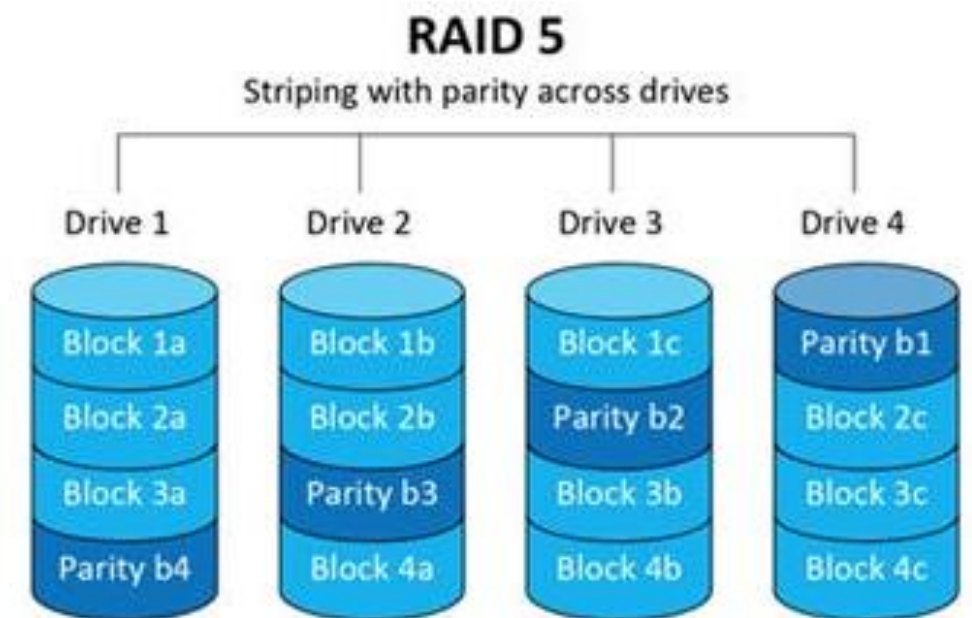
RAID de Discos Magnéticos

- ▶ **RAID 0 (*striping*):**
 - ▶ Performance sin redundancia.
 - ▶ Si se rompe un disco, se pierde la información.
- ▶ **RAID 1 (*mirroring*):**
 - ▶ Redundancia sin performance.
 - ▶ Necesita el doble de discos.
 - ▶ Si se rompe un disco, se usa el espejo.



RAID de Discos Magnéticos

- ▶ **RAID 5 (*distributed parity*):**
 - ▶ Performance y redundancia.
 - ▶ Utiliza un disco extra.
 - ▶ Calcula la paridad de los datos divididos, y la distribuye alternadamente entre todos los discos.
 - ▶ Si se rompe un disco (solamente uno), permite reconstruir la información.
 - ▶ Últimamente reemplazado por RAID 6, que tolera el fallo simultáneo de dos discos.
- ▶ **RAID 1+0 (*striped mirrors*):**
 - ▶ Performance y redundancia.
 - ▶ Se crean pares de discos espejados, y a esos pares se los divide.
 - ▶ Requiere mínimamente de 4 discos.



RAID de Discos Magnéticos

- ▶ En grandes datacenters, RAID deja de ser rentable.
 - ▶ Porque al usar discos muy grandes, la reconstrucción por una falla puede demorar mucho tiempo, aumentando la probabilidad de nuevas fallas.
- ▶ Se lo reemplaza por una técnica denominada *Erasure Coding*.
 - ▶ No usa paridad, sino algoritmos de corrección de errores de tipo Reed-Solomon.
 - ▶ Los verán en Transmisión de Datos.
 - ▶ Dividen un archivo en k partes, le agregan m partes adicionales de redundancia, y lo reparten en $k+m$ discos.
 - ▶ Soporta que el fallo simultáneo de m discos.
- ▶ Provee mucha más tolerancia a fallos con no tanto costo adicional.
- ▶ Necesita mayor procesamiento, generando mayor latencia.
- ▶ Muy usado en almacenamiento de gran escala.
 - ▶ Drive, Dropbox, Instagram, Netflix, etc.

Dispositivos de I/O: Memorias FLASH

- ▶ Hoy en día, las memorias FLASH son empleadas masivamente.
 - ▶ Pen drives, tarjetas de memoria SD, celulares, tablets.
 - ▶ Microcontroladores.
 - ▶ Discos de estado sólido (SSD).
- ▶ Impulsada principalmente por la economía de escala.
- ▶ Celdas de memoria agrupadas en páginas, a su vez agrupadas en bloques de hasta 256 páginas.
 - ▶ **No se escribe de a bytes**, sino de a páginas completas.
 - ▶ Y los datos se borran de a bloques completos.
 - ▶ Una escritura de una página demora aprox. **200 us**.

Memorias FLASH: funcionamiento

- ▶ Tienen como una compuerta flotante que no está conectada.
 - ▶ Permite almacenar electrones durante mucho tiempo.
- ▶ Para escribir un nuevo valor, se aplica una tensión alta en esta compuerta flotante.
 - ▶ La aplicación sucesiva de estas altas tensiones termina deteriorando la compuerta.
 - ▶ Por lo tanto, la cantidad de escrituras que se realizan en una memoria de este tipo está limitada.
- ▶ Esta característica introduce una nueva métrica para memorias FLASH: **duración o resistencia** (*endurance*).
 - ▶ Usualmente se mide en cantidad de TeraBytes que pueden ser escritos (*TBW*), o en ciclos de escritura/borrado (*P/E cycles*).
- ▶ Para los más curiosos:
<https://www.extremetech.com/extreme/210492-extremetech-explains-how-do-ssds-work>

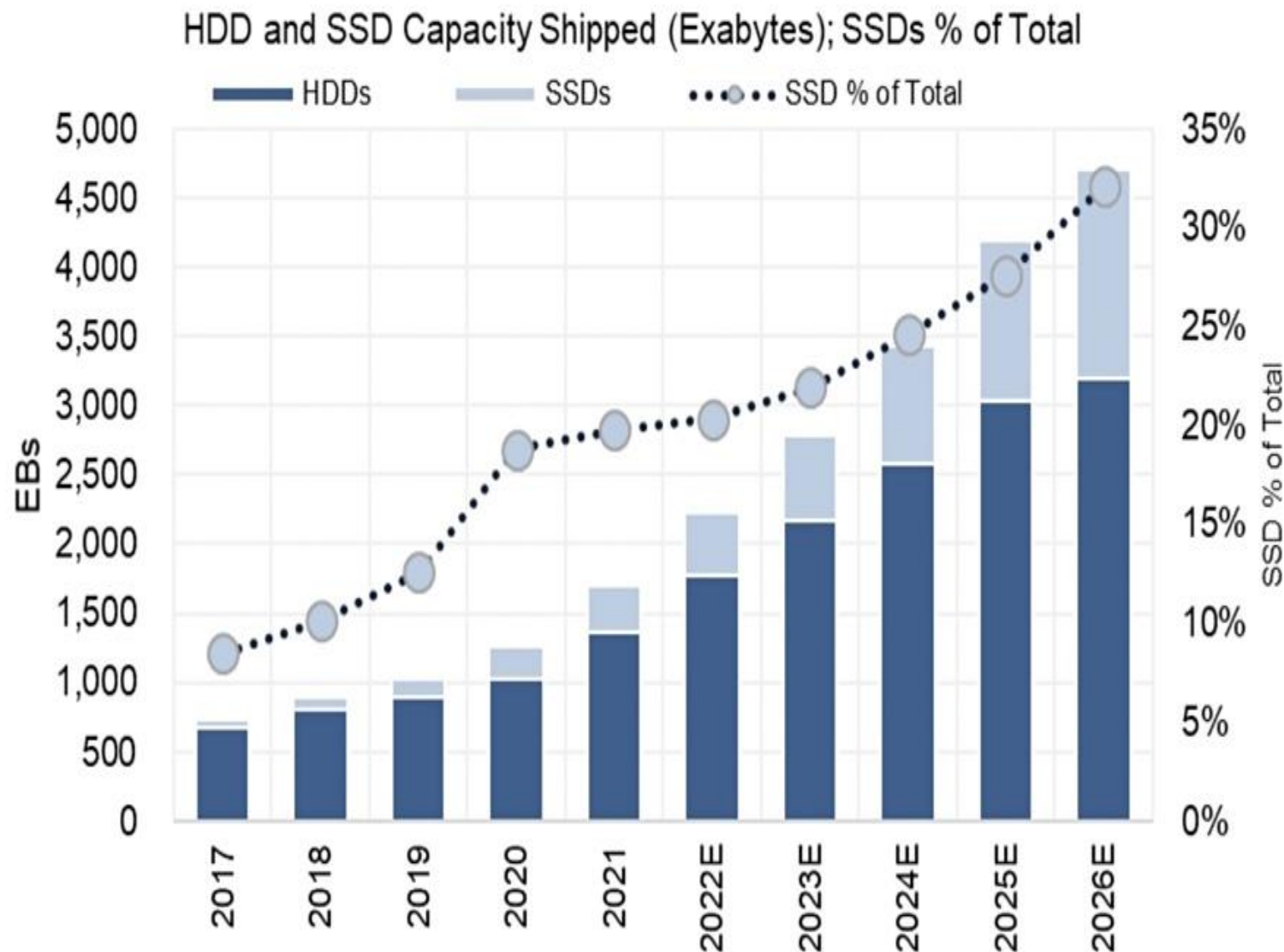
Discos SSD vs. Discos Magnéticos

- ▶ Ventajas de los SSD
 - ▶ No tienen componentes mecánicos.
 - ▶ **Enorme reducción en AMAT.** Entre 100 y 1000 veces más rápidos.
 - ▶ Mejor resistencia a golpes e impactos.
 - ▶ Menor tiempo de transferencia
 - ▶ Más rápida en lecturas, similar en escrituras.
- ▶ Desventajas de los SSD
 - ▶ Cantidad limitada de escrituras.
 - ▶ Mayores latencias a medida que aumenta su capacidad.
 - ▶ **¡Mayor consumo de energía!** Entre un 40% y 60% mayor, dependiendo de la tarea.
 - ▶ Fuente: <https://www.tomshardware.com/news/firm-claims-hdds-can-use-less-power-than-ssds>
 - ▶ Más caros (no sólo en \$, sino en \$/byte).
- ▶ *¿Podrán los discos SSD reemplazar a los HDD?*

Discos SSD vs. Discos Magnéticos

- ▶ Según reportes anuales de la organización IDC:
 - ▶ En 2020 se crearon 64.2 ZiB (10^{21} Bytes) de datos.
 - ▶ Se espera un 23% de crecimiento anual para los próximos 5 años.
 - ▶ No todos esos datos se almacenan.
 - ▶ En el mismo año, el almacenamiento global instalado fue de 6.7 ZiB.
 - ▶ Se espera un 19% de crecimiento anual para los próximos 5 años.
 - ▶ **El 65% de esa cantidad, son HDD.** Se espera que baje a 54% en 2024.
- ▶ *“The greatest thing ever for the HDD industry was the smartphone.” R. Anderson (WD, 2021)*
- ▶ Actualmente, las IA generan un montón de texto, imágenes y videos que deben ser almacenados en algún lado.
- ▶ *“You don’t have AI without data, and you don’t have data without storage.” K. Sennesael (WD, 2025)*

Discos SSD vs. Discos Magnéticos



Source: Gartner; Wells Fargo Securities, LLC.

- ▶ Comparando \$/GB, datos tomados de Amazon.com:

Año	HDD	SSD	HDD/SSD
2013	0,04	0,72	18
2018	0,02	0,24	12
2021	0,02	0,10	5
2024	0,02	0,08	4

- ▶ La tendencia indica que los precios (\$/GB) se van acercando.
 - ▶ Algunas estimaciones indican el crossover entre 2028 y 2033.
- ▶ En 2026, los precios de ambos están en aumento, por la IA.

Discos SSD vs. Discos Magnéticos

- ▶ Los HDD no pueden competir en performance con los SSD, pero sí en otras métricas, generalmente relacionadas a la capacidad.
 - ▶ Por ejemplo, para un datacenter pasar de cinco HDD de 4 TB a uno de 20 TB ahorra racks, espacio, cableado, energía, etc.
 - ▶ Los que sí corren riesgo de desaparecer son los HDD “premium”, o de “alta performance”.
- ▶ Nuevas tecnologías para HDD en pleno desarrollo:
 - ▶ Buscan incrementar densidad de grabación (capacidad), cantidad de platos por HDD (capacidad) y multiactuadores (transferencia).
 - ▶ Se proyectan HDD de 120 TB para 2028.
- ▶ La limitación en la vida útil de los SSD hace poco probable que desaparezcan los HDD.
 - ▶ Inclusive, algunos sugieren que, en el largo plazo, son los SSD los que serán reemplazados por una tecnología mejor.

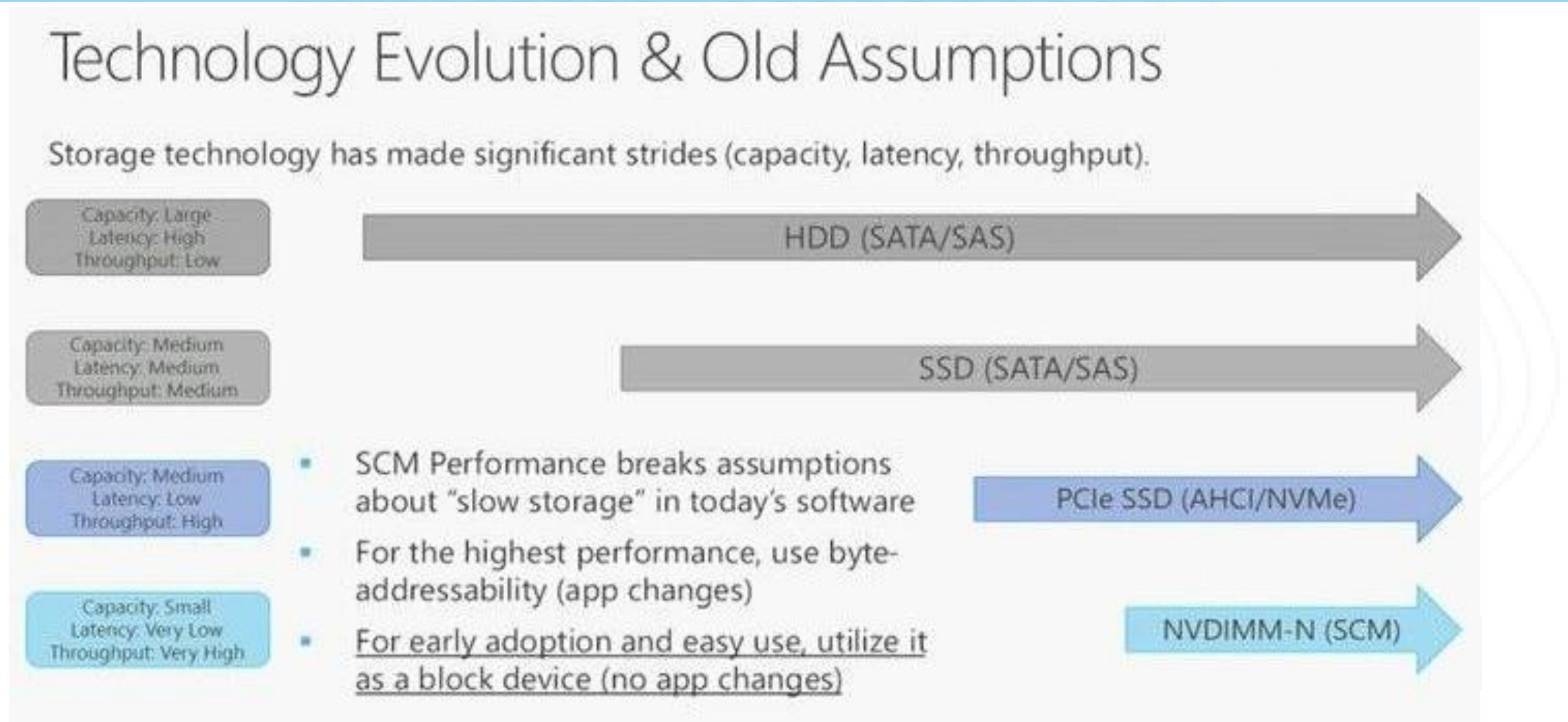
Discos SSD vs. Discos Magnéticos

- ▶ La solución actual a la disyuntiva: convivencia.
 - ▶ Técnicamente conocida como ***Tiering*** de discos.
- ▶ El SO clasifica al almacenamiento en capas (*tiers*) según su rendimiento, y mueve automáticamente los bloques de datos según su frecuencia de acceso:
 - ▶ Tier 0/1, capa caliente: SSD rápidos para los datos más accedidos.
 - ▶ Tier 2, capa tibia: SSD lentos (SATA) para datos frecuentes pero no críticos.
 - ▶ Tier 3, capa fría: HDD para datos históricos.
- ▶ Nuevamente, **Jerarquía** (Gran idea en Arquitectura de Computadoras).

Tendencias actuales

- ▶ Hay una **brecha de performance entre Memoria e I/O**.
 - ▶ Los discos SSD son 1000 veces más lentos.
- ▶ Nuevas aplicaciones requieren nuevas capacidades de I/O:
 - ▶ En *Machine Learning* y computación científica, el cuello de botella está en I/O.
 - ▶ Los datos son procesados mucho más rápidamente de lo que son traídos.
 - ▶ Generando muchísimas paradas en los procesadores.
- ▶ Nuevas métricas están surgiendo
 - ▶ Se habla de IOPS (Operaciones de I/O por segundo), en contraste a MIPS o FLOPS.
- ▶ Muchas investigaciones en nuevos métodos de almacenamiento (fibra de vidrio, ADN).
 - ▶ Buscan mayor capacidad, mayor duración y menor costo/byte.
- ▶ **La evolución de una tecnología suele mejorar el ancho de banda, y sólo nuevas tecnologías mejoran la latencia.**

Memorias RAM no Volátiles



- ▶ SCM es Storage-Class Memory, también NVM (Non Volatile Memory) o PMEM (Persistent Memory).
- ▶ Las primeras se conectan al bus PCIe, las más nuevas directamente al bus de memoria.
- ▶ NVDIMMs, basadas en DIMM + NAND para otorgarles persistencia.

Memorias RAM no Volátiles

- ▶ ¡Requieren nuevos modelos de programación!
 - ▶ Normalmente, *¿quién hace los accesos a memoria no volátil?*
 - ▶ Casi toda la jerarquía de memorias vista es volátil.
- ▶ ¡Pero generan casos de uso aún no imaginados!
 - ▶ La separación entre Memoria e I/O se está tornando difusa.
 - ▶ Generan nuevos cuellos de botella (SO, protocolos de redes).
- ▶ Requieren todo un ecosistema para que funcionen.
 - ▶ Hardware, Software, Estándares, Plataformas.
- ▶ Representan el mayor avance tecnológico en memorias desde hace 50 años.
 - ▶ Hacen falta varios años más hasta que podamos entender completamente cómo aprovecharlas al máximo.

Intel Optane Memory

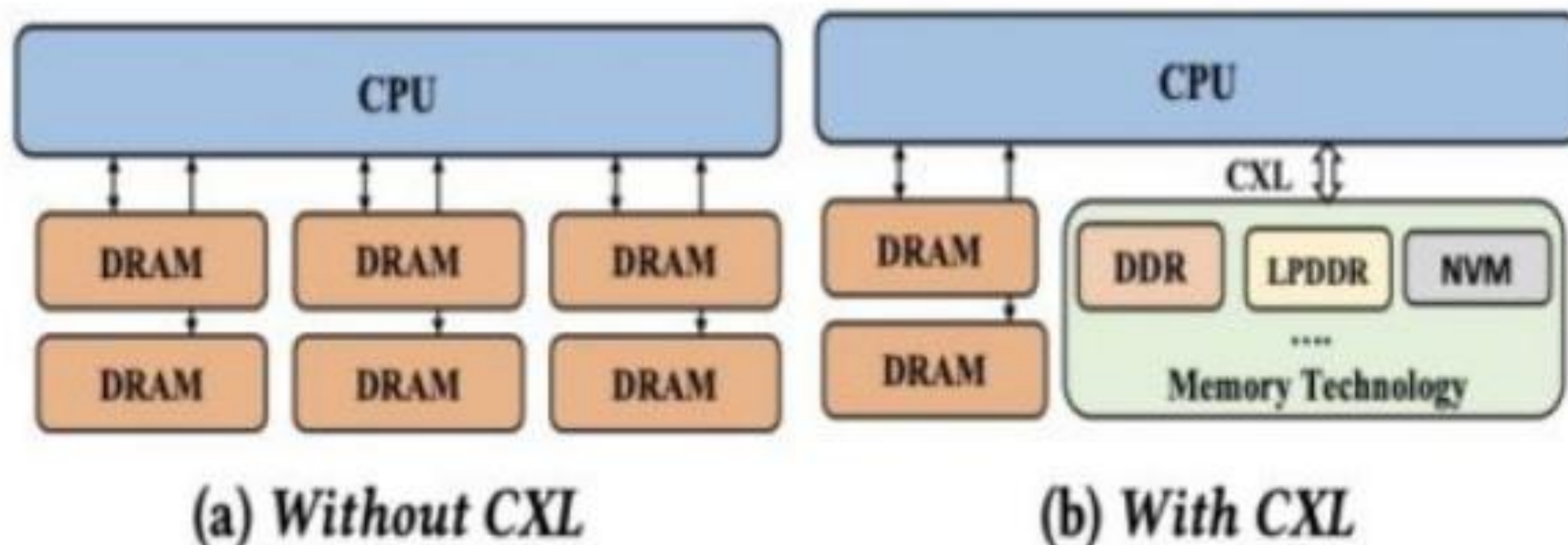
- ▶ Anunciada en 2015, surgió en 2019, y fue **dada de baja en 2022.**
- ▶ Basada en memorias 3D XPoint, un avance tecnológico.
 - ▶ Buscaba cerrar la brecha entre las DRAM y las FLASH actuales.
 - ▶ 100x más rápidas que FLASH, 10x más lentas que DRAM.
 - ▶ 1000x más escrituras que las FLASH.
- ▶ No supo encontrar un mercado.
 - ▶ No pudieron reemplazar ni a las DRAM (por velocidad), ni a las FLASH (por costo).
 - ▶ No pudieron despegarse de los procesadores Xeon.

UltraRAM

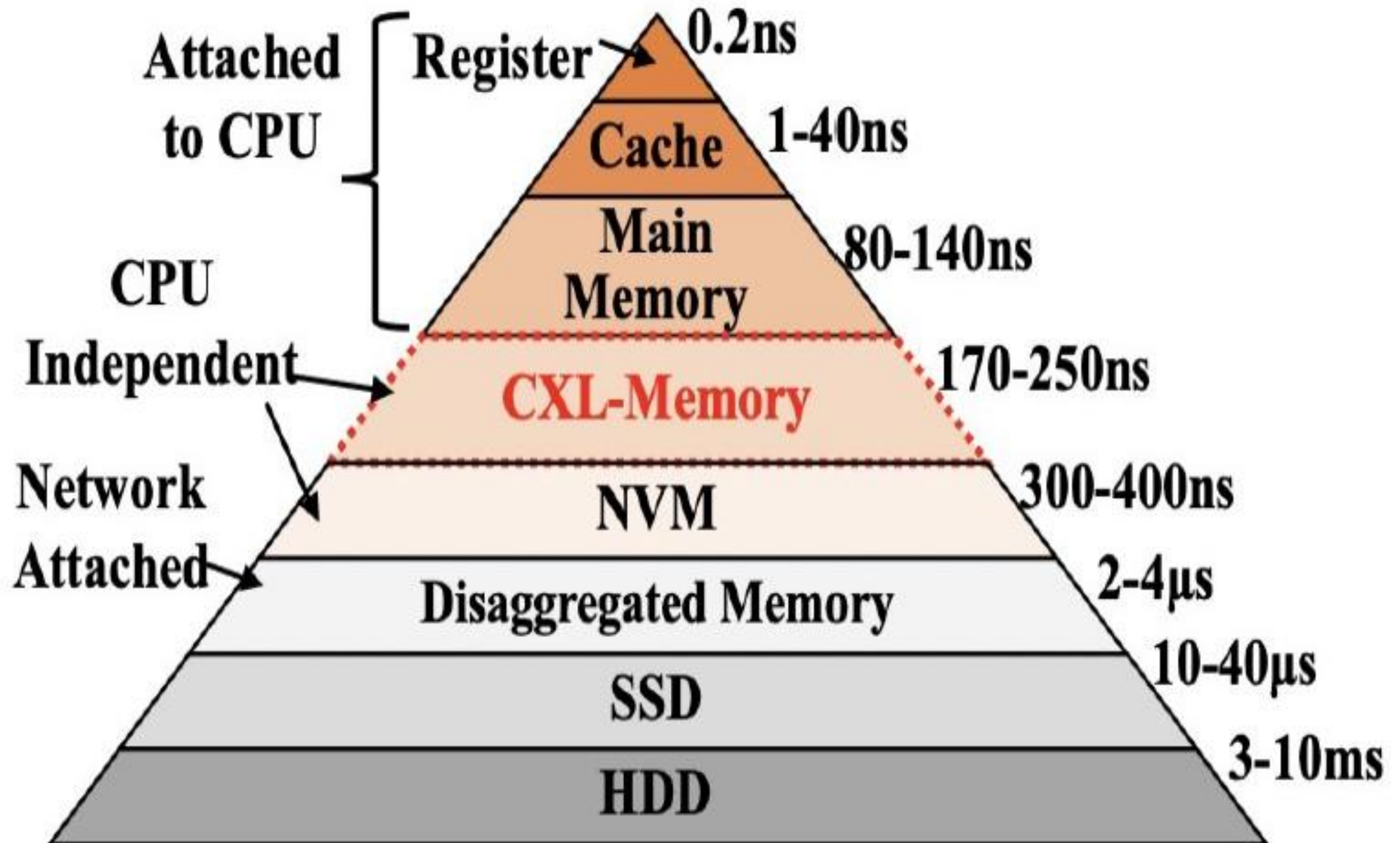
- ▶ Investigación de la Universidad de Lancaster (UK), de enero de 2022.
 - ▶ Aún no en producción. Se espera para 2033, si todo va bien.
- ▶ Tecnología de memoria no volátil con la velocidad, eficiencia energética y *endurance* de una DRAM.
 - ▶ 4000x mayor *endurance* que una FLASH (aprox. 1000 años).
 - ▶ Latencia 10x menor que una DRAM, del orden de los pocos ns.
 - ▶ 100x eficiencia energética que una DRAM, 1000x que una FLASH.
- ▶ Promesa de romper la brecha entre memoria principal y Entrada/Salida.
 - ▶ Achicar la cantidad de niveles de la jerarquía de memoria.
 - ▶ Objetivo: tener 1 TiB de UltraRAM, y que sea tanto memoria primaria como almacenamiento.
- ▶ Posible problema: ni una mención sobre el costo aún.
- ▶ Posible problema: si no es igual o más rápida que la DRAM, difícil que cumpla sus promesas.

CXL

- ▶ Es un bus, así que lo veremos en el Tema siguiente.
- ▶ A partir de su versión 2.0 (del 2020) introduce el concepto de “*memory pooling*”, mediante el cual permite acceder a módulos de DRAM remotamente a través de la red.
 - ▶ Ampliaría a DDR, permitiendo aumentar notablemente la capacidad, sin aumentar tanto la latencia.
 - ▶ Permitiría manejar idénticamente distintas tecnologías de memorias, como distintos DDR, LPDDR, HBM y NVM, de diferentes costos y performances.
 - ▶ De esta manera, solucionaría la *Memory Capacity Wall* y la *Memory Bandwidth Wall*.



Jerarquía de memoria propuesta



Resumen final

- ▶ Un sistema de I/O está compuesto por múltiples dispositivos, muy diferentes entre sí.
- ▶ El SO se encarga de manejar el subsistema de I/O.
- ▶ Usa I/O mapeada en memoria, con interrupciones, y con la ayuda de un DMA.
- ▶ Aunque siempre prima la performance, también son importantes la escalabilidad, la tolerancia a fallos, la confiabilidad y la disponibilidad.
 - ▶ Performance: Ancho de banda y Tiempo de Respuesta.
 - ▶ Compromiso entre productividad y latencia.
- ▶ Métricas adecuadas dependen del tipo de computadora y su uso.

Resumen final

- ▶ Discos magnéticos:
 - ▶ Tres componentes principales de su AMAT: seek time, latencia rotacional, tiempo de transferencia.
 - ▶ La transferencia es mucho menor que los otros dos, que son mecánicos.
- ▶ RAID: múltiples discos para mejorar productividad, con redundancia para mejorar tolerancia a fallos.
- ▶ Discos SSD basados en Memorias FLASH.
 - ▶ Mucha mejor performance, más caros.
 - ▶ Limitaciones en las escrituras.
- ▶ Nuevas memorias RAM no volátiles.
 - ▶ Reciente aparición, muchas promesas y expectativas.