

**SEGURIDAD Y ALTA  
DISPONIBILIDAD**

**UD 6**

**21 ASIR**

**27/02/2011**

**VICEN MORALES**

## INDICE UD6.- Implantación de soluciones de Alta Disponibilidad

### •Análisis de configuraciones de alta disponibilidad:

- Alta disponibilidad:

Concepto.

Funcionamiento ininterrumpido.

Integridad de datos y recuperación de servicio.

- Soluciones de alta disponibilidad:

Servidores redundantes. RAID.

Sistemas de «clusters».

SAN, NAS, FiberChannel.

Balancedores de carga.

- Instalación y configuración de soluciones de alta disponibilidad:

- Virtualización de sistemas:

Posibilidades de la virtualización de sistemas.

Herramientas para la virtualización.

Configuración y utilización de máquinas virtuales

## UD 6: Implantación de soluciones de Alta Disponibilidad

### •Análisis de configuraciones de alta disponibilidad:

- Alta disponibilidad:

Concepto.

**Alta disponibilidad** (High availability) es un protocolo de diseño del sistema y su implementación asociada que asegura un cierto grado absoluto de continuidad operacional durante un período de medición dado. Disponibilidad se refiere a la habilidad de la comunidad de usuarios para acceder al sistema, someter nuevos trabajos, actualizar o alterar trabajos existentes o recoger los resultados de trabajos previos. Si un usuario no puede acceder al sistema se dice que está no disponible. El término tiempo de inactividad (downtime) es usado para definir cuándo el sistema no está disponible

Funcionamiento ininterrumpido.

Integridad de datos y recuperación de servicio.

- Soluciones de alta disponibilidad:

Servidores redundantes. RAID.

Los **sistemas redundantes**, en ingeniería de computadores, son aquellos en los que se repiten aquellos datos o hardware de carácter crítico que se quiere asegurar ante los posibles fallos que puedan surgir por su uso continuado.

Se presenta como una solución a los problemas de protección y confiabilidad. Este tipo de sistemas se encarga de realizar el mismo proceso en más de una estación, ya que si por algún motivo alguna dejara de funcionar o colapsara, inmediatamente otro tendría que ocupar su lugar y realizar las tareas del anterior.

Las técnicas de redundancia han sido usadas por la industria militar y aeroespacial por muchos años para alcanzar una alta confiabilidad. Una base de datos replicada es un ejemplo de sistema distribuido redundante.

En informática, el acrónimo **RAID** (del inglés «conjunto redundante de discos independientes», anteriormente conocido como **Redundant Array of Inexpensive Disks**, «conjunto redundante de discos baratos») hace referencia a un sistema de almacenamiento que usa múltiples discos duros o SSD entre los que se distribuyen o replican los datos. Dependiendo de su configuración (a la que suele llamarse «nivel»), los beneficios de un RAID respecto a un único disco son uno o varios de los siguientes: mayor integridad, mayor tolerancia a fallos, mayor throughput (rendimiento) y mayor capacidad. En sus implementaciones originales, su ventaja clave era la habilidad de

combinar varios dispositivos de bajo coste y tecnología más antigua en un conjunto que ofrecía mayor capacidad, fiabilidad, velocidad o una combinación de éstas que un solo dispositivo de última generación y coste más alto.

En el nivel más simple, un RAID combina varios discos duros en una sola unidad lógica. Así, en lugar de ver varios discos duros diferentes, el sistema operativo ve uno solo. Los RAID suelen usarse en servidores y normalmente (aunque no es necesario) se implementan con unidades de disco de la misma capacidad. Debido al decremento en el precio de los discos duros y la mayor disponibilidad de las opciones RAID incluidas en los chipsets de las placas base, los RAID se encuentran también como opción en las computadoras personales más avanzadas. Esto es especialmente frecuente en las computadoras dedicadas a tareas intensivas y que requiera asegurar la integridad de los datos en caso de fallo del sistema. Esta característica no está obviamente disponible en los sistemas RAID por software, que suelen presentar por tanto el problema de reconstruir el conjunto de discos cuando el sistema es reiniciado tras un fallo para asegurar la integridad de los datos. Por el contrario, los sistemas basados en software son mucho más flexibles (permitiendo, por ejemplo, construir RAID de particiones en lugar de discos completos y agrupar en un mismo RAID discos conectados en varias controladoras) y los basados en hardware añaden un punto de fallo más al sistema (la controladora RAID).

Todas las implementaciones pueden soportar el uso de uno o más discos de reserva (*hot spare*), unidades preinstaladas que pueden usarse inmediatamente (y casi siempre automáticamente) tras el fallo de un disco del RAID. Esto reduce el tiempo del período de reparación al acortar el tiempo de reconstrucción del RAID.

Los niveles RAID más comúnmente usados son:

- RAID 0: Conjunto dividido
- RAID 1: Conjunto en espejo
- RAID 5: Conjunto dividido con paridad distribuida

### **RAID 0 (Data Striping)**

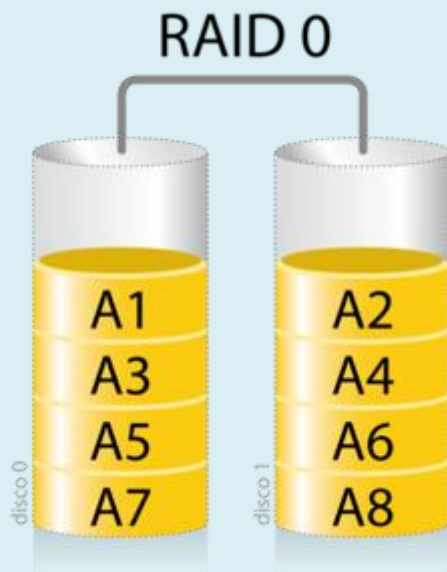


Diagrama de una configuración RAID 0.

Un **RAID 0** (también llamado **conjunto dividido** o **volumen dividido**) distribuye los datos equitativamente entre dos o más discos sin información de paridad que proporcione redundancia. Es importante señalar que el RAID 0 no era uno de los niveles RAID originales y que no es redundante. El RAID 0 se usa normalmente para incrementar el rendimiento, aunque también puede utilizarse como forma de crear un pequeño número de grandes discos virtuales a partir de un gran número de pequeños discos físicos. Un RAID 0 puede ser creado con discos de diferentes tamaños, pero el espacio de almacenamiento añadido al conjunto estará limitado por el tamaño del disco más pequeño (por ejemplo, si un disco de 300 GB se divide con uno de 100 GB, el tamaño del conjunto resultante será sólo de 200 GB, ya que cada disco aporta 100GB). Una buena implementación de un RAID 0 dividirá las operaciones de lectura y escritura en bloques de igual tamaño, por lo que distribuirá la información equitativamente entre los dos discos. También es posible crear un RAID 0 con más de dos discos, si bien, la fiabilidad del conjunto será igual a la fiabilidad media de cada disco entre el número de discos del conjunto; es decir, la fiabilidad total —medida como MTTF o MTBF— es (aproximadamente) inversamente proporcional al número de discos del conjunto (pues para que el conjunto falle es suficiente con que lo haga *cualquiera* de sus discos).

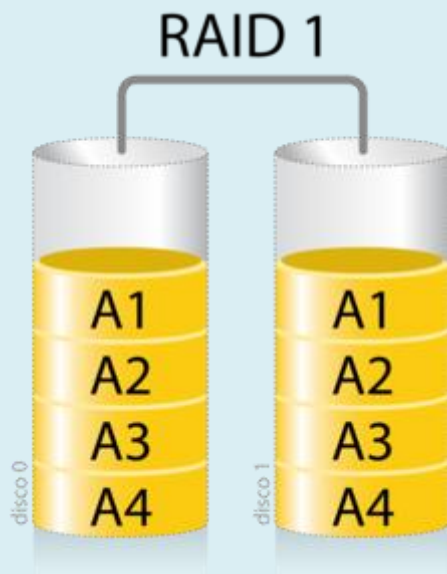


Diagrama de una configuración RAID 1.

### RAID 1

Un **RAID 1** crea una copia exacta (o **espejo**) de un conjunto de datos en dos o más discos. Esto resulta útil cuando el rendimiento en lectura es más importante que la capacidad. Un conjunto RAID 1 sólo puede ser tan grande como el más pequeño de sus discos. Un RAID 1 clásico consiste en dos discos en espejo, lo que incrementa exponencialmente la fiabilidad respecto a un solo disco; es decir, la probabilidad de fallo del conjunto es igual al producto de las probabilidades de fallo de cada uno de los discos (pues para que el conjunto falle es necesario que lo hagan *todos* sus discos).

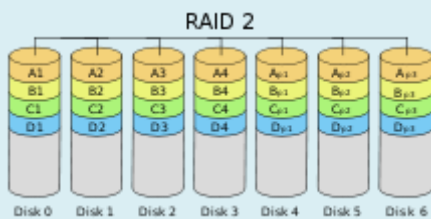
Adicionalmente, dado que todos los datos están en dos o más discos, con hardware habitualmente independiente, el rendimiento de lectura se incrementa aproximadamente como múltiplo lineal del número de copias; es decir, un RAID 1 puede estar leyendo simultáneamente dos datos diferentes en dos discos diferentes, por lo que su rendimiento se duplica. Para maximizar los beneficios sobre el rendimiento del RAID 1 se recomienda el uso de controladoras de disco independientes, una para cada disco (práctica que algunos denominan *splitting* o *duplexing*).

Como en el RAID 0, el tiempo medio de lectura se reduce, ya que los sectores a buscar pueden dividirse entre los discos, bajando el tiempo de búsqueda y subiendo la tasa de transferencia, con el único límite de la velocidad soportada por la controladora RAID. Sin embargo, muchas tarjetas RAID 1 IDE antiguas leen sólo de un disco de la pareja, por lo que su rendimiento es igual al de un único disco. Algunas implementaciones RAID 1 antiguas también leen de ambos discos simultáneamente y comparan los datos para detectar errores. La detección y corrección de errores en los discos duros modernos hacen esta práctica poco útil.

Al escribir, el conjunto se comporta como un único disco, dado que los datos deben ser escritos en todos los discos del RAID 1. Por tanto, el rendimiento no mejora.

El RAID 1 tiene muchas ventajas de administración. Por ejemplo, en algunos entornos 24/7, es posible «dividir el espejo»: marcar un disco como inactivo, hacer una copia de seguridad de dicho disco y luego «reconstruir» el espejo. Esto requiere que la aplicación de gestión del conjunto soporte la recuperación de los datos del disco en el momento de la división. Este procedimiento es menos crítico que la presencia de una característica de snapshot en algunos sistemas de archivos, en la que se reserva algún espacio para los cambios, presentando una vista estática en un punto temporal dado del sistema de archivos. Alternativamente, un conjunto de discos puede ser almacenado de forma parecida a como se hace con las tradicionales cintas.

**RAID 2**



**RAID 2**

Un **RAID 2** divide los datos a nivel de bits en lugar de a nivel de bloques y usa un código de Hamming para la corrección de errores. Los discos son sincronizados por la controladora para funcionar al unísono. Éste es el único nivel RAID original que actualmente no se usa. Permite tasas de transferencias extremadamente altas.

Teóricamente, un RAID 2 necesitaría 39 discos en un sistema informático moderno: 32 se usarían para almacenar los bits individuales que forman cada palabra y 7 se usarían para la corrección de errores.

**RAID 3**

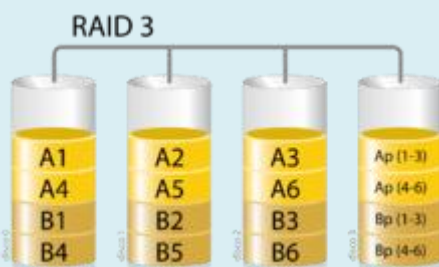


Diagrama de una configuración RAID 3. Cada número representa un byte de datos; cada columna, un disco.

Un **RAID 3** usa división a nivel de bytes con un disco de paridad dedicado. El RAID 3 se usa rara vez en la práctica. Uno de sus efectos secundarios es que normalmente no puede atender varias peticiones simultáneas, debido a que por definición cualquier simple bloque de datos se dividirá por todos los miembros del conjunto, residiendo la misma dirección dentro de cada uno de ellos. Así, cualquier operación de lectura o

escritura exige activar todos los discos del conjunto, suele ser un poco lento porque se producen cuellos de botella. Son discos paralelos pero no son independientes (no se puede leer y escribir al mismo tiempo).

En el ejemplo del gráfico, una petición del bloque «A» formado por los bytes A1 a A6 requeriría que los tres discos de datos buscaran el comienzo (A1) y devolvieran su contenido. Una petición simultánea del bloque «B» tendría que esperar a que la anterior concluyese.

**RAID 4**

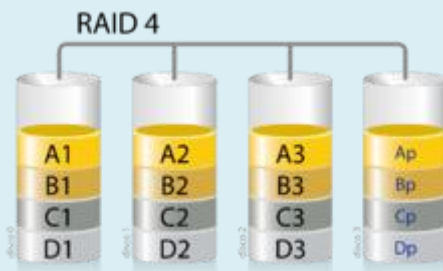


Diagrama de una configuración RAID 4. Cada número representa un bloque de datos; cada columna, un disco.

Un **RAID 4**, también conocido como **IDA (acceso independiente con discos dedicados a la paridad)** usa división a nivel de bloques con un disco de paridad dedicado. Necesita un mínimo de 3 discos físicos. El RAID 4 es parecido al RAID 3 excepto porque divide a nivel de bloques en lugar de a nivel de bytes. Esto permite que cada miembro del conjunto funcione independientemente cuando se solicita un único bloque. Si la controladora de disco lo permite, un conjunto RAID 4 puede servir varias peticiones de lectura simultáneamente. En principio también sería posible servir varias peticiones de escritura simultáneamente, pero al estar toda la información de paridad en un solo disco, éste se convertiría en el cuello de botella del conjunto.

En el gráfico de ejemplo anterior, una petición del bloque «A1» sería servida por el disco 0. Una petición simultánea del bloque «B1» tendría que esperar, pero una petición de «B2» podría atenderse concurrentemente.

**RAID 5**

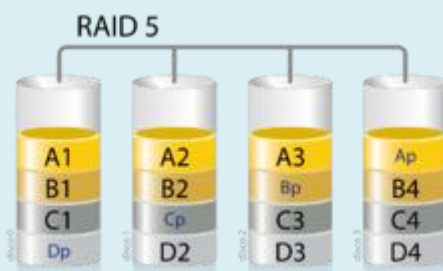






Diagrama de una configuración RAID 5.

Un **RAID 5** usa división de datos a nivel de bloques distribuyendo la información de paridad entre todos los discos miembros del conjunto. El RAID 5 ha logrado popularidad gracias a su bajo coste de redundancia. Generalmente, el RAID 5 se implementa con soporte hardware para el cálculo de la paridad. RAID 5 necesitará un mínimo de 3 discos para ser implementado.

En el gráfico de ejemplo anterior, una petición de lectura del bloque «A1» sería servida por el disco 0. Una petición de lectura simultánea del bloque «B1» tendría que esperar, pero una petición de lectura de «B2» podría atenderse concurrentemente ya que sería servida por el disco 1.

Cada vez que un bloque de datos se escribe en un RAID 5, se genera un bloque de paridad dentro de la misma división (*stripe*). Un bloque se compone a menudo de muchos sectores consecutivos de disco. Una serie de bloques (un bloque de cada uno de los discos del conjunto) recibe el nombre colectivo de división (*stripe*). Si otro bloque, o alguna porción de un bloque, es escrita en esa misma división, el bloque de paridad (o una parte del mismo) es recalculada y vuelta a escribir. El disco utilizado por el bloque de paridad está escalonado de una división a la siguiente, de ahí el término «bloques de paridad distribuidos». Las escrituras en un RAID 5 son costosas en términos de operaciones de disco y tráfico entre los discos y la controladora.

Los bloques de paridad no se leen en las operaciones de lectura de datos, ya que esto sería una sobrecarga innecesaria y disminuiría el rendimiento. Sin embargo, los bloques de paridad se leen cuando la lectura de un sector de datos provoca un error de CRC. En este caso, el sector en la misma posición relativa dentro de cada uno de los bloques de datos restantes en la división y dentro del bloque de paridad en la división se utiliza para reconstruir el sector erróneo. El error CRC se oculta así al resto del sistema. De la misma forma, si falla un disco del conjunto, los bloques de paridad de los restantes discos son combinados matemáticamente con los bloques de datos de los restantes discos para reconstruir los datos del disco que ha fallado «al vuelo».

Lo anterior se denomina a veces Modo Interino de Recuperación de Datos (*Interim Data Recovery Mode*). El sistema sabe que un disco ha fallado, pero sólo con el fin de que el sistema operativo pueda notificar al administrador que una unidad necesita ser reemplazada: las aplicaciones en ejecución siguen funcionando ajenas al fallo. Las lecturas y escrituras continúan normalmente en el conjunto de discos, aunque con alguna degradación de rendimiento. La diferencia entre el RAID 4 y el RAID 5 es que, en el Modo Interno de Recuperación de Datos, el RAID 5 puede ser ligeramente más rápido, debido a que, cuando el CRC y la paridad están en el disco que falló, los cálculos no tienen que realizarse, mientras que en el RAID 4, si uno de los discos de datos falla, los cálculos tienen que ser realizados en cada acceso.

El RAID 5 requiere al menos tres unidades de disco para ser implementado. El fallo de un segundo disco provoca la pérdida completa de los datos.

El número máximo de discos en un grupo de redundancia RAID 5 es teóricamente ilimitado, pero en la práctica es común limitar el número de unidades. Los inconvenientes de usar grupos de redundancia mayores son una mayor probabilidad de fallo simultáneo de dos discos, un mayor tiempo de reconstrucción y una mayor probabilidad de hallar un sector irrecuperable durante una reconstrucción. A medida que el número de discos en un conjunto RAID 5 crece, el MTBF (tiempo medio entre fallos) puede ser más bajo que el de un único disco. Esto sucede cuando la probabilidad de que falle un segundo disco en los  $N-1$  discos restantes de un conjunto en el que ha fallado un disco en el tiempo necesario para detectar, reemplazar y recrear dicho disco es mayor que la probabilidad de fallo de un único disco. Una alternativa que proporciona una protección de paridad dual, permitiendo así mayor número de discos por grupo, es el RAID 6.

Algunos vendedores RAID evitan montar discos de los mismos lotes en un grupo de redundancia para minimizar la probabilidad de fallos simultáneos al principio y el final de su vida útil.

Las implementaciones RAID 5 presentan un rendimiento malo cuando se someten a cargas de trabajo que incluyen muchas escrituras más pequeñas que el tamaño de una división (*stripe*). Esto se debe a que la paridad debe ser actualizada para cada escritura, lo que exige realizar secuencias de lectura, modificación y escritura tanto para el bloque de datos como para el de paridad. Implementaciones más complejas incluyen a menudo cachés de escritura no volátiles para reducir este problema de rendimiento.

En el caso de un fallo del sistema cuando hay escrituras activas, la paridad de una división (*stripe*) puede quedar en un estado inconsistente con los datos. Si esto no se detecta y repara antes de que un disco o bloque falle, pueden perderse datos debido a que se usará una paridad incorrecta para reconstruir el bloque perdido en dicha división. Esta potencial vulnerabilidad se conoce a veces como «agujero de escritura». Son comunes el uso de caché no volátiles y otras técnicas para reducir la probabilidad de ocurrencia de esta vulnerabilidad.

### RAID 6

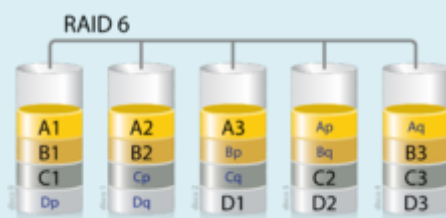


Diagrama de una configuración RAID 6. Cada número representa un bloque de datos; cada columna, un disco;  $p$  y  $q$ , códigos Reed-Solomon.

Un **RAID 6** amplía el nivel RAID 5 añadiendo otro bloque de paridad, por lo que divide los datos a nivel de bloques y distribuye los dos bloques de paridad entre todos los miembros del conjunto. El RAID 6 no era uno de los niveles RAID originales.

El RAID 6 puede ser considerado un caso especial de código Reed-Solomon.<sup>1</sup> El RAID 6, siendo un caso degenerado, exige sólo sumas en el Campo de galois. Dado que se está operando sobre bits, lo que se usa es un campo binario de Galois ( $GF(2^m)$ ). En las representaciones cíclicas de los campos binarios de Galois, la suma se calcula con un simple XOR.

Tras comprender el RAID 6 como caso especial de un código Reed-Solomon, se puede ver que es posible ampliar este enfoque para generar redundancia simplemente

produciendo otro código, típicamente un polinomio en  $GF(2^8)$  ( $m = 8$  significa que estamos operando sobre bytes). Al añadir códigos adicionales es posible alcanzar cualquier número de discos redundantes, y recuperarse de un fallo de ese mismo número de discos en cualquier puntos del conjunto, pero en el nivel RAID 6 se usan dos únicos códigos.

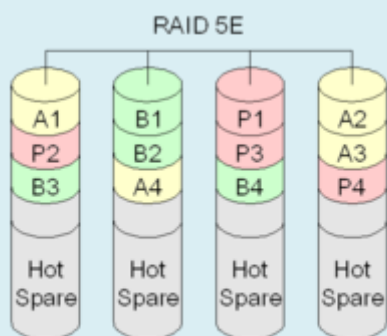
Al igual que en el RAID 5, en el RAID 6 la paridad se distribuye en divisiones (*stripes*), con los bloques de paridad en un lugar diferente en cada división.

El RAID 6 es ineficiente cuando se usa un pequeño número de discos pero a medida que el conjunto crece y se dispone de más discos la pérdida en capacidad de almacenamiento se hace menos importante, creciendo al mismo tiempo la probabilidad de que dos discos fallen simultáneamente. El RAID 6 proporciona protección contra fallos dobles de discos y contra fallos cuando se está reconstruyendo un disco. En caso de que sólo tengamos un conjunto puede ser más adecuado que usar un RAID 5 con un disco de reserva (*hot spare*).

La capacidad de datos de un conjunto RAID 6 es  $n-2$ , siendo  $n$  el número total de discos del conjunto.

Un RAID 6 no penaliza el rendimiento de las operaciones de lectura, pero sí el de las de escritura debido al proceso que exigen los cálculos adicionales de paridad. Esta penalización puede minimizarse agrupando las escrituras en el menor número posible de divisiones (*stripes*), lo que puede lograrse mediante el uso de un sistema de archivos WAFL.

**RAID 5E y RAID 6E**



## RAID 5E

Se suele llamar RAID 5E y RAID 6E a las variantes de RAID 5 y RAID 6 que incluyen discos de reserva. Estos discos pueden estar conectados y preparados (*hot spare*) o en espera (*standby spare*). En los RAID 5E y RAID 6E, los discos de reserva están disponibles para cualquiera de las unidades miembro. No suponen mejora alguna del rendimiento, pero sí se minimiza el tiempo de reconstrucción (en el caso de los discos *hot spare*) y las labores de administración cuando se producen fallos. Un disco de reserva no es realmente parte del conjunto hasta que un disco falla y el conjunto se reconstruye sobre el de reserva.

### Niveles RAID anidados

Muchas controladoras permiten anidar niveles RAID, es decir, que un RAID pueda usarse como elemento básico de otro en lugar de discos físicos. Resulta instructivo pensar en estos conjuntos como capas dispuestas unas sobre otras, con los discos físicos en la inferior.

Los RAID anidados se indican normalmente uniendo en un solo número los correspondientes a los niveles RAID usados, añadiendo a veces un «+» entre ellos. Por ejemplo, el RAID 10 (o RAID 1+0) consiste conceptualmente en múltiples conjuntos de nivel 1 almacenados en discos físicos con un nivel 0 encima, agrupando los anteriores niveles 1. En el caso del RAID 0+1 se usa más esta forma que RAID 01 para evitar la confusión con el RAID 1. Sin embargo, cuando el conjunto de más alto nivel es un RAID 0 (como en el RAID 10 y en el RAID 50), la mayoría de los vendedores eligen omitir el «+», a pesar de que RAID 5+0 sea más informativo.

Al anidar niveles RAID, se suele combinar un nivel RAID que proporcione redundancia con un RAID 0 que aumenta el rendimiento. Con estas configuraciones es preferible tener el RAID 0 como nivel más alto y los conjuntos redundantes debajo, porque así será necesario reconstruir menos discos cuando uno falle. (Así, el RAID 10 es preferible al RAID 0+1 aunque las ventajas administrativas de «dividir el espejo» del RAID 1 se perderían.)

Los niveles RAID anidados más comúnmente usados son:

- [RAID 0+1](#): Un espejo de divisiones
- [RAID 1+0](#): Una división de espejos
- [RAID 30](#): Una división de niveles RAID con paridad dedicada
- [RAID 100](#): Una división de una división de espejos
- [RAID 10+1](#): Un Espejo de espejos

### RAID 0+1

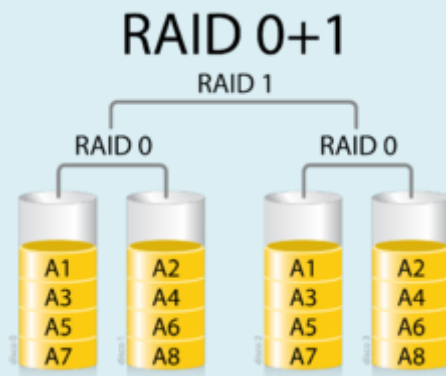


Diagrama de una configuración RAID 0+1.

Un **RAID 0+1** (también llamado **RAID 01**, que no debe confundirse con RAID 1) es un RAID usado para replicar y compartir datos entre varios discos. La diferencia entre un RAID 0+1 y un RAID 1+0 es la localización de cada nivel RAID dentro del conjunto final: un RAID 0+1 es un espejo de divisiones.

Como puede verse en el diagrama, primero se crean dos conjuntos RAID 0 (dividiendo los datos en discos) y luego, sobre los anteriores, se crea un conjunto RAID 1 (realizando un espejo de los anteriores). La ventaja de un RAID 0+1 es que cuando un disco duro falla, los datos perdidos pueden ser copiados del otro conjunto de nivel 0 para reconstruir el conjunto global. Sin embargo, añadir un disco duro adicional en una división, es obligatorio añadir otro al de la otra división para equilibrar el tamaño del conjunto.

Además, el RAID 0+1 no es tan robusto como un RAID 1+0, no pudiendo tolerar dos fallos simultáneos de discos salvo que sean en la misma división. Es decir, cuando un disco falla, la otra división se convierte en un punto de fallo único. Además, cuando se sustituye el disco que falló, se necesita que todos los discos del conjunto participen en la reconstrucción de los datos.

Con la cada vez mayor capacidad de las unidades de discos (liderada por las unidades serial ATA), el riesgo de fallo de los discos es cada vez mayor. Además, las tecnologías de corrección de errores de bit no han sido capaces de mantener el ritmo de rápido incremento de las capacidades de los discos, provocando un mayor riesgo de hallar errores físicos irrecuperables.

Dados estos cada vez mayores riesgos del RAID 0+1 (y su vulnerabilidad ante los fallos dobles simultáneos), muchos entornos empresariales críticos están empezando a evaluar configuraciones RAID más tolerantes a fallos que añaden un mecanismo de paridad subyacente. Entre los más prometedores están los enfoques híbridos como el RAID 0+1+5 (espejo sobre paridad única) o RAID 0+1+6 (espejo sobre paridad dual). Son los más habituales por las empresas.

### RAID 1+0

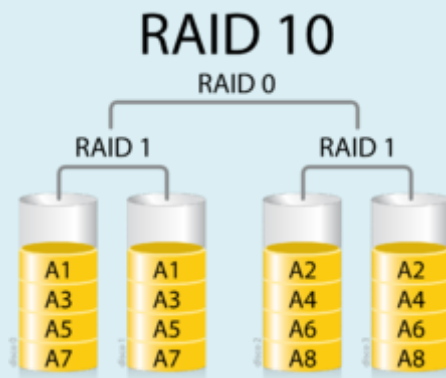


Diagrama de una configuración RAID 10.

Un **RAID 1+0**, a veces llamado **RAID 10**, es parecido a un RAID 0+1 con la excepción de que los niveles RAID que lo forman se invierte: el RAID 10 es una división de espejos.<sup>2</sup>

En cada división RAID 1 pueden fallar todos los discos salvo uno sin que se pierdan datos. Sin embargo, si los discos que han fallado no se reemplazan, el restante pasa a ser un punto único de fallo para todo el conjunto. Si ese disco falla entonces, se perderán todos los datos del conjunto completo. Como en el caso del RAID 0+1, si un disco que ha fallado no se reemplaza, entonces un solo error de medio irrecuperable que ocurra en el disco espejado resultaría en pérdida de datos.

Debido a estos mayores riesgos del RAID 1+0, muchos entornos empresariales críticos están empezando a evaluar configuraciones RAID más tolerantes a fallos que añaden un mecanismo de paridad subyacente. Entre los más prometedores están los enfoques híbridos como el RAID 0+1+5 (espejo sobre paridad única) o RAID 0+1+6 (espejo sobre paridad dual).

El RAID 10 es a menudo la mejor elección para bases de datos de altas prestaciones, debido a que la ausencia de cálculos de paridad proporciona mayor velocidad de escritura.

**RAID 30**

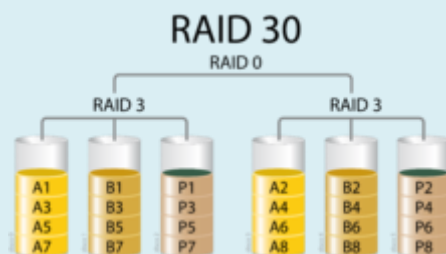


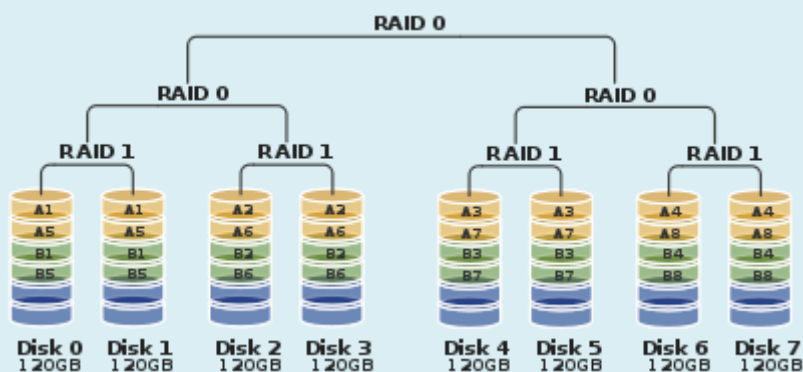
Diagrama de una configuración RAID 30.

El **RAID 30** o división con conjunto de paridad dedicado es una combinación de un RAID 3 y un RAID 0. El RAID 30 proporciona tasas de transferencia elevadas combinadas con una alta fiabilidad a cambio de un coste de implementación muy alto.

La mejor forma de construir un RAID 30 es combinar dos conjuntos RAID 3 con los datos divididos en ambos conjuntos. El RAID 30 trocea los datos en bloque más pequeños y los divide en cada conjunto RAID 3, que a su vez lo divide en trozos aún menores, calcula la paridad aplicando un XOR a cada uno y los escriben en todos los discos del conjunto salvo en uno, donde se almacena la información de paridad. El tamaño de cada bloque se decide en el momento de construir el RAID. Etc...

El RAID 30 permite que falle un disco de cada conjunto RAID 3. Hasta que estos discos que fallaron sean reemplazados, los otros discos de cada conjunto que sufrió el fallo son puntos únicos de fallo para el conjunto RAID 30 completo. En otras palabras, si alguno de ellos falla se perderán todos los datos del conjunto. El tiempo de recuperación necesario (detectar y responder al fallo del disco y reconstruir el conjunto sobre el disco nuevo) representa un periodo de vulnerabilidad para el RAID.

### RAID 100



RAID 100.

Un **RAID 100**, a veces llamado también **RAID 10+0**, es una división de conjuntos RAID 10. El RAID 100 es un ejemplo de «RAID cuadrado», un RAID en el que conjuntos divididos son a su vez divididos conjuntamente de nuevo.

Todos los discos menos unos podrían fallar en cada RAID 1 sin perder datos. Sin embargo, el disco restante de un RAID 1 se convierte así en un punto único de fallo para el conjunto degradado. A menudo el nivel superior de división se hace por software. Algunos vendedores llaman a este nivel más alto un *MetaLun* o *Soft Stripe*.

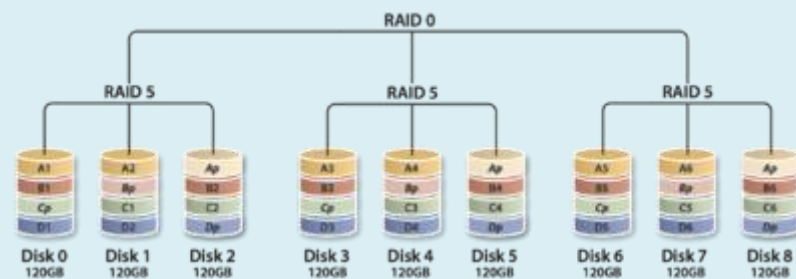
Los principales beneficios de un RAID 100 (y de los RAID cuadrado en general) sobre un único nivel RAID son mejor rendimiento para lecturas aleatorias y la mitigación de los puntos calientes de riesgo en el conjunto. Por estas razones, el RAID 100 es a menudo la mejor elección para bases de datos muy grandes, donde el conjunto software subyacente limita la cantidad de discos físicos permitidos en cada conjunto estándar. Implementar niveles RAID anidados permite eliminar virtualmente el límite de unidades físicas en un único volumen lógico.

### RAID 10+1

Un **RAID 10+1**, es un reflejo de dos RAID 10. Se utiliza en la llamados **Network RAID** que aceptan algunas cabinas de datos. Es un sistema de alta disponibilidad por red, lo que permite la replicación de datos entre cabinas a nivel de RAID, con lo cual se simplifica ampliamente la gestión de repliación de cabinas.

El RAID 10+1, tratándose de espejos de RAID10 que tienen una gran velocidad de acceso, hace que el rendimiento sea muy aceptable, siempre y cuando se respete el requerimiento de 2ms de latencia como máximo.

### RAID 50



RAID 50.

Un **RAID 50**, a veces llamado también **RAID 5+0**, combina la división a nivel de bloques de un RAID 0 con la paridad distribuida de un RAID 5, siendo pues un conjunto RAID 0 dividido de elementos RAID 5.

Un disco de cada conjunto RAID 5 puede fallar sin que se pierdan datos. Sin embargo, si el disco que falla no se reemplaza, los discos restantes de dicho conjunto se convierten en un punto único de fallo para todo el conjunto. Si uno falla, todos los datos del conjunto global se pierden. El tiempo necesario para recuperar (detectar y responder al fallo de disco y reconstruir el conjunto sobre el nuevo disco) representa un periodo de vulnerabilidad del conjunto RAID.

La configuración de los conjuntos RAID repercute sobre la tolerancia a fallos general. Una configuración de tres conjuntos RAID 5 de siete discos cada uno tiene la mayor capacidad y eficiencia de almacenamiento, pero sólo puede tolerar un máximo de tres fallos potenciales de disco. Debido a que la fiabilidad del sistema depende del rápido reemplazo de los discos averiados para que el conjunto pueda reconstruirse, es común construir conjuntos RAID 5 de seis discos con un disco de reserva en línea (*hot spare*) que permite empezar de inmediato la reconstrucción en caso de fallo del conjunto. Esto no soluciona el problema de que el conjunto sufre un estrés máximo durante la reconstrucción dado que es necesario leer cada bit, justo cuando es más vulnerable. Una configuración de siete conjuntos RAID 5 de tres discos cada uno puede tolerar hasta siete fallos de disco pero tiene menor capacidad y eficiencia de almacenamiento.

El RAID 50 mejora el rendimiento del RAID 5, especialmente en escritura, y proporciona mejor tolerancia a fallos que un nivel RAID único. Este nivel se



recomienda para aplicaciones que necesitan gran tolerancia a fallos, capacidad y rendimiento de búsqueda aleatoria.

A medida que el número de unidades del conjunto RAID 50 crece y la capacidad de los discos aumenta, el tiempo de recuperación lo hace también.

### Niveles RAID propietarios

Aunque todas las implementaciones de RAID difieren en algún grado de la especificación idealizada, algunas compañías han desarrollado implementaciones RAID completamente propietarias que difieren sustancialmente de todas las demás.

### RAID 50EE

Himperia utiliza el **RAID 50EE** en el **ZStore 3212L**. Se trata de un RAID 0 de dos *pools*, cada uno de ellos con RAID 5EE (7+1+1). Tolera el fallo simultáneo de dos discos, y hasta 4 discos no simultáneos. El tiempo de reconstrucción se reduce al mínimo, gracias al RAID 5EE. Y se mejora el rendimiento gracias al RAID 0.

### Paridad doble

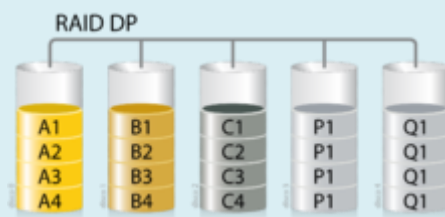


Diagrama una configuración RAID de doble paridad.

Una adición frecuente a los niveles RAID existente es la **paridad doble**, a veces implementada y conocida como **paridad diagonal**. Como en el RAID 6, hay dos conjuntos de información de chequeo de paridad, pero a diferencia de aquél, el segundo conjunto no es otro conjunto de puntos calculado sobre un síndrome polinomial diferente para los mismos grupos de bloques de datos, sino que se calcula la paridad extra a partir de un grupo diferente de bloques de datos. Por ejemplo, sobre el gráfico tanto el RAID 5 como el RAID 6 calcularían la paridad sobre todos los bloques de la letra A para generar uno o dos bloques de paridad. Sin embargo, es bastante fácil calcular la paridad contra múltiples grupos de bloques, en lugar de sólo sobre los bloques de la letra A: puede calcularse la paridad sobre los bloques de la letra A y un grupo permutado de bloques.

De nuevo sobre el ejemplo, los bloques Q son los de la paridad doble. El bloque Q2 se calcularía como  $A2 \text{ xor } B3 \text{ xor } P3$ , mientras el bloque Q3 se calcularía como  $A3 \text{ xor } P2 \text{ xor } C1$  y el Q1 sería  $A1 \text{ xor } B2 \text{ xor } C3$ . Debido a que los bloques de paridad doble se distribuyen correctamente, es posible reconstruir dos discos de datos que fallen mediante recuperación iterativa. Por ejemplo, B2 podría recuperarse sin usar ninguno

de los bloques  $x_1$  ni  $x_2$  mediante el cálculo de  $B_3 \text{ xor } P_3 \text{ xor } Q_2 = A_2$ , luego  $A_2 \text{ xor } A_3 \text{ xor } P_1 = A_1$ , y finalmente  $A_1 \text{ xor } C_3 \text{ xor } Q_1 = B_2$ .

No es recomendable que el sistema de paridad doble funcione en modo degradado debido a su bajo rendimiento.

**RAID 1.5**

**RAID 1.5** es un nivel RAID propietario de HighPoint a veces incorrectamente denominado RAID 15. Por la poca información disponible, parece ser una implementación correcta de un RAID 1. Cuando se lee, los datos se recuperan de ambos discos simultáneamente y la mayoría del trabajo se hace en hardware en lugar de en el controlador software.

**RAID 7**

**RAID 7** es una marca registrada de Storage Computer Corporation, que añade cachés a un RAID 3 o RAID 4 para mejorar el rendimiento.

**RAID S o RAID de paridad**

**RAID S** es un sistema RAID de paridad distribuida propietario de EMC Corporation usado en sus sistemas de almacenamiento Symmetrix. Cada volumen reside en un único disco físico, y se combinan arbitrariamente varios volúmenes para el cálculo de paridad. EMC llamaba originalmente a esta característica RAID S y luego la rebautizó **RAID de paridad** (*Parity RAID*) para su plataforma Symmetrix DMX. EMC ofrece también actualmente un RAID 5 estándar para el Symmetrix DMX.

**Matrix RAID**

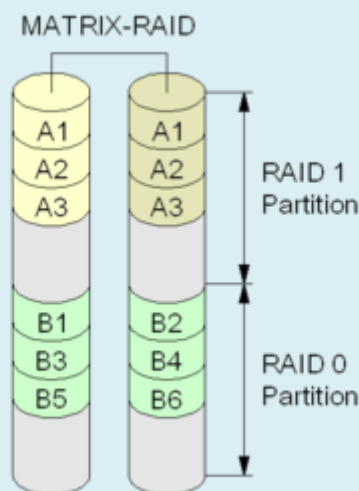


Diagrama una configuración Matriz RAID.

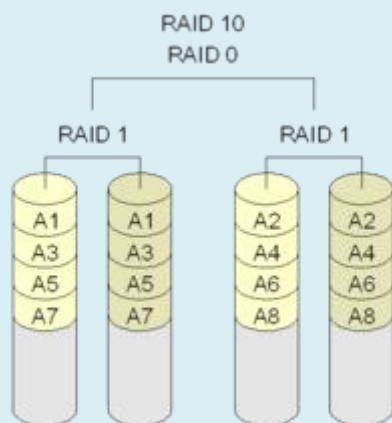
**Matrix RAID** ('matriz RAID') es una característica que apareció por vez primera en la BIOS RAID Intel ICH6R. No es un nuevo nivel RAID.

El Matrix RAID utiliza dos o más discos físicos, asignando partes de idéntico tamaño de cada uno de ellos diferentes niveles de RAID. Así, por ejemplo, sobre 4 discos de un total de 600GB, se pueden usar 200 en raid 0, 200 en raid 10 y 200 en raid 5.

Actualmente, la mayoría de los otros productos RAID BIOS de gama baja sólo permiten que un disco participen en un único conjunto.

Este producto está dirigido a los usuarios domésticos, proporcionando una zona segura (la sección RAID 1) para documentos y otros archivos que se desean almacenar redundantemente y una zona más rápida (la sección RAID 0) para el sistema operativo, aplicaciones, etcétera.

### Linux MD RAID 10



### RAID 10

La controladora RAID software del kernel de Linux (llamada **md**, de *multiple disk*, 'disco múltiple') puede ser usada para construir un conjunto RAID 1+0 clásico, pero también permite un único nivel RAID 10 con algunas extensiones interesantes.

En particular, soporta un espejado de  $k$  bloques en  $n$  unidades cuando  $k$  no es divisible por  $n$ . Esto se hace repitiendo cada bloque  $k$  veces al escribirlo en un conjunto RAID 0 subyacente de  $n$  unidades. Evidentemente esto equivale a la configuración RAID 10 estándar.

Linux también permite crear otras configuraciones RAID usando la controladora *md* (niveles 0, 1, 4, 5 y 6) además de otros usos no RAID como almacenamiento multirruta y LVM2.

### IBM ServeRAID 1E



Diagrama una configuración RAID 1E.

La serie de adaptadores IBM ServeRAID soportan un espejado doble de un número arbitrario de discos, como se ilustra en el gráfico.

Esta configuración es tolerante a fallos de unidades no adyacentes. Otros sistemas de almacenamiento como el StorEdge T3 de Sun soportan también este modo.

### RAID Z

El sistema de archivos ZFS de Sun Microsystems implementa un esquema de redundancia integrado parecido al RAID 5 que se denomina **RAID Z**. Esta configuración evita el «agujero de escritura» del RAID 5<sup>4</sup> y la necesidad de la secuencia leer-modificar-escribir para operaciones de escrituras pequeñas efectuando sólo escrituras de divisiones (*stripes*) completas, espejando los bloques pequeños en lugar de protegerlos con el cálculo de paridad, lo que resulta posible gracias a que el sistema de archivos conoce la estructura de almacenamiento subyacente y puede gestionar el espacio adicional cuando lo necesita.

### Posibilidades de RAID

#### Lo que RAID puede hacer

- RAID puede mejorar el *uptime*. Los niveles RAID 1, 0+1 o 10, 5 y 6 (sus variantes, como el 50) permiten que un disco falle mecánicamente y que aun así los datos del conjunto sigan siendo accesibles para los usuarios. En lugar de exigir que se realice una restauración costosa en tiempo desde una cinta, DVD o algún otro medio de respaldo lento, un RAID permite que los datos se recuperen en un disco de reemplazo a partir de los restantes discos del conjunto, mientras al mismo tiempo permanece disponible para los usuarios en un modo degradado. Esto es muy valorado por las empresas, ya que el tiempo de no disponibilidad suele tener graves repercusiones. Para usuarios domésticos, puede permitir el ahorro del tiempo de restauración de volúmenes grandes, que requerirían varios DVD o cintas para las copias de seguridad.

- RAID puede mejorar el rendimiento de **ciertas** aplicaciones. Los niveles RAID 0, 5 y 6 usan variantes de división (*striping*) de datos, lo que permite que varios discos atiendan simultáneamente las operaciones de lectura lineales, aumentando la tasa de transferencia sostenida. Las aplicaciones de escritorio que trabajan con archivos grandes, como la edición de vídeo e imágenes, se benefician de esta mejora. También es útil para las operaciones de copia de respaldo de disco a disco. Además, si se usa un RAID 1 o un RAID basado en división con un tamaño de bloque lo suficientemente grande se logran mejoras de rendimiento para patrones de acceso que implique múltiples lecturas simultáneas (por ejemplo, bases de datos multiusuario).

### Lo que RAID no puede hacer

- RAID no protege los datos. Un conjunto RAID tiene un sistema de archivos, lo que supone un punto único de fallo al ser vulnerable a una amplia variedad de riesgos aparte del fallo físico de disco, por lo que RAID no evita la pérdida de datos por estas causas. RAID no impedirá que un virus destruya los datos, que éstos se corrompan, que sufran la modificación o borrado accidental por parte del usuario ni que un fallo físico en otro componente del sistema afecten a los datos.
- RAID no simplifica la recuperación de un desastre. Cuando se trabaja con un solo disco, éste es accesible normalmente mediante un controlador ATA o SCSI incluido en la mayoría de los sistemas operativos. Sin embargo, las controladoras RAID necesitan controladores software específicos. Las herramientas de recuperación que trabajan con discos simples en controladoras genéricas necesitarán controladores especiales para acceder a los datos de los conjuntos RAID. Si estas herramientas no los soportan, los datos serán inaccesibles para ellas.
- RAID no mejora el rendimiento de **todas** las aplicaciones. Esto resulta especialmente cierto en las configuraciones típicas de escritorio. La mayoría de aplicaciones de escritorio y videojuegos hacen énfasis en la estrategia de buffering y los tiempos de búsqueda de los discos. Una mayor tasa de transferencia sostenida supone poco beneficio para los usuarios de estas aplicaciones, al ser la mayoría de los archivos a los que se accede muy pequeños. La división de discos de un RAID 0 mejora el rendimiento de transferencia lineal pero no lo demás, lo que hace que la mayoría de las aplicaciones de escritorio y juegos no muestren mejora alguna, salvo excepciones. Para estos usos, lo mejor es comprar un disco más grande, rápido y caro en lugar de dos discos más lentos y pequeños en una configuración RAID 0.
- RAID no facilita el traslado a un sistema nuevo. Cuando se usa un solo disco, es relativamente fácil trasladar el disco a un sistema nuevo: basta con conectarlo, si cuenta con la misma interfaz. Con un RAID no es tan sencillo: la BIOS RAID debe ser capaz de leer los metadatos de los miembros del conjunto para reconocerlo adecuadamente y hacerlo disponible al sistema operativo. Dado que los distintos fabricantes de controladoras RAID usan diferentes formatos de metadatos (incluso controladoras de un mismo fabricante son incompatibles

si corresponden a series diferentes) es virtualmente imposible mover un conjunto RAID a una controladora diferente, por lo que suele ser necesario mover también la controladora. Esto resulta imposible en aquellos sistemas donde está integrada en la placa base. Esta limitación puede obviarse con el uso de RAID por software, que a su vez añaden otras diferentes (especialmente relacionadas con el rendimiento).

## Sistemas de «clusters».

El término **cluster** (a veces españolizado como **clúster**) se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de hardwares comunes y que se comportan como si fuesen una única computadora.

Hoy en día desempeñan un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno.

La tecnología de clústeres ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

El cómputo con clústeres surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran.

Simplemente, un clúster es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio.

Los clústeres son usualmente empleados para mejorar el rendimiento y/o la disponibilidad por encima de la que es provista por un solo computador típicamente siendo más económico que computadores individuales de rapidez y disponibilidad comparables.

De un clúster se espera que presente combinaciones de los siguientes servicios:

1. Alto rendimiento
2. Alta disponibilidad
3. Balanceo de carga
4. Escalabilidad

La construcción de los ordenadores del clúster es más fácil y económica debido a su flexibilidad: pueden tener todos la misma configuración de hardware y sistema operativo (clúster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (clúster semihomogéneo), o tener diferente hardware y sistema operativo (clúster heterogéneo), lo que hace más fácil y económica su construcción.

Para que un clúster funcione como tal, no basta solo con conectar entre sí los ordenadores, sino que es necesario proveer un sistema de manejo del clúster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

## SAN, NAS, FiberChannel.

Una **red de área de almacenamiento**, en inglés **SAN** (*storage area network*), es una red concebida para conectar servidores, matrices (arrays) de discos y librerías de soporte. Principalmente, está basada en tecnología **fibre channel** y más recientemente en **iSCSI**. Su función es la de conectar de manera rápida, segura y fiable los distintos elementos que la conforman.

Una red SAN se distingue de otros modos de almacenamiento en red por el modo de acceso a bajo nivel. El tipo de tráfico en una SAN es muy similar al de los discos duros como ATA, SATA y SCSI. En otros métodos de almacenamiento, (como SMB o NFS), el servidor solicita un determinado fichero, p.ej. "/home/usuario/wikipedia". En una SAN el servidor solicita "el bloque 6000 del disco 4". La mayoría de las SAN actuales usan el protocolo SCSI para acceder a los datos de la SAN, aunque no usen interfaces físicas SCSI. Este tipo de redes de datos se han utilizado y se utilizan tradicionalmente en grandes main frames como en IBM, SUN o HP. Aunque recientemente con la incorporación de Microsoft se ha empezado a utilizar en máquinas con sistemas operativos Microsoft.

Una SAN es una red de almacenamiento dedicada que proporciona acceso de nivel de bloque a LUNs. Un LUN, o número de unidad lógica, es un disco virtual proporcionado por la SAN. El administrador del sistema tiene el mismo acceso y los derechos a la LUN como si fuera un disco directamente conectado a la misma. El administrador puede particionar y formatear el disco en cualquier medio que él elija.

Dos protocolos de red utilizados en una SAN son Fibre Channel e iSCSI. Una red de canal de fibra es muy rápida y no está agobiada por el tráfico de la red LAN de la empresa. Sin embargo, es muy cara. Las tarjetas de canal de fibra óptica cuestan alrededor de \$ 1000.00 USD cada una. También requieren conmutadores especiales de canal de fibra. iSCSI es una nueva tecnología que envía comandos SCSI sobre una red TCP / IP. Este método no es tan rápido como una red Fibre Channel, pero ahorra costes, ya que utiliza un hardware de red menos costoso.

A partir de desastres como lo fue el "martes negro" en el año 2001 la gente de TI, han tomado acciones al respecto, con servicios de cómo recuperarse ante un desastre, cómo recuperar miles de datos y lograr la continuidad del negocio, una de las opciones es contar con la Red de área de almacenamiento, sin embargo las compañías se pueden enfrentar a cientos de ataques, por lo que es necesario contar con un plan en caso de contingencia; es de vital importancia que el sitio dónde se encuentre la Red de almacenamiento, se encuentre en un área geográfica distinta a dónde se ubican los servidores que contienen la información crítica; además se trata de un modelo centralizado fácil de administrar, puede tener un bajo costo de expansión y administración, lo que la hace una red fácilmente escalable; fiabilidad, debido a que se hace más sencillo aplicar ciertas políticas para proteger a la red.

**NAS** (del inglés *Network Attached Storage*) es el nombre dado a una tecnología de almacenamiento dedicada a compartir la capacidad de almacenamiento de un computador (Servidor) con ordenadores personales o servidores clientes a través de una red (normalmente TCP/IP), haciendo uso de un Sistema Operativo optimizado para dar acceso con los protocolos CIFS, NFS, FTP o TFTP.

Generalmente, los sistemas NAS son dispositivos de almacenamiento específicos a los que se accede desde los equipos a través de protocolos de red (normalmente TCP/IP). También se podría considerar un sistema NAS a un servidor (Linux, Windows, ...) que comparte sus unidades por red, pero la definición suele aplicarse a sistemas específicos.

Los protocolos de comunicaciones NAS están basados en ficheros por lo que el cliente solicita el fichero completo al servidor y lo maneja localmente, están por ello orientados a información almacenada en ficheros de pequeño tamaño y gran cantidad. Los protocolos usados son protocolos de compartición de ficheros como NFS o Microsoft Common Internet File System (CIFS).

Muchos sistemas NAS cuentan con uno o más dispositivos de almacenamiento para incrementar su capacidad total. Frecuentemente, estos dispositivos están dispuestos en RAID (*Redundant Arrays of Independent Disks*) o contenedores de almacenamiento redundante.

**Fiber Channel Protocol (FCP)** es un protocolo de transporte para la **transmisión de comandos SCSI sobre redes Fiber Channel**. Muy utilizado y extendido.

Bien, ya sabemos que estamos hablando de redes de almacenamiento basadas en conexiones de fibra óptica, sobre las que circula información a través del protocolo FCP (Fiber Channel Protocol), principalmente intercambio de comandos SCSI entre los dispositivos conectados a dicha red de fibra. Ahora bien, ¿Qué posible topologías o formas de conexión existen con Fiber Channel (FC)?

La tecnología Fiber Channel (FC) ofrece **tres posibles topologías**:



- **Conexión punto a punto (FC-P2P, Point-to-Point).** Se utiliza en soluciones de **almacenamiento DAS**, en las cuales, se conecta una cabina de almacenamiento directamente a las tarjetas HBA del servidor (o servidores, veáse el caso de un **Cluster**). Bueno, puede ser una cabina de almacenamiento, un robot de cintas, etc.
- **Bucle arbitrado (FC-AL).** Permite conectar **hasta 126 dispositivos en anillo, compartiendo el ancho de banda**, de forma análoga a las redes Token Ring. Que yo sepa, no tiene mucho uso.
- **Red conmutada (FC-SW, Switch Fabric).** Aprovecha la utilización de conmutadores o switches Fiber Channel (FC) para la conexión de múltiples dispositivos, sin compartir el ancho de banda (siempre que el switch tenga suficiente capacidad de conmutación, claro), soportando **hasta 15,5 millones de nodos**, y manteniendo todo el ancho de banda. Se utiliza habitualmente en arquitecturas de almacenamiento SAN, utilizándose el término de **Switch Fabric**, por la capacidad de este tipo de redes de reenviar los paquetes entre switches, desde el origen al destino, por el puerto apropiado.

**Suelen utilizarse múltiples swiches, y múltiples puertos** tanto en la cabina de almacenamiento (Storage) y Librería de Cintas, como en los equipos cliente (Servidores Windows, UNIX o Mainframe), de tal modo, que puedan definirse **múltiples caminos** entre el equipo cliente (Servidor Windows, UNIX o Mainframe) y el destino, la cabina de Almacenamiento (Storage) o Librería de Cintas. Esto proporciona un mecanismo de Alta Disponibilidad, vital en las actuales infraestructuras de almacenamiento SAN corporativas. Suele apoyarse en protocolos para la gestión de caminos múltiples al almacenamiento, como es el caso de **MPIO (Multi Path Input Output) y SecurePath** (protocolo propietario de HP) , y gracias también a la configuraciones de **LUN Masking y Zoning**.

## Balancedores de carga.

Un **balanceador de carga** fundamentalmente es un dispositivo de hardware o software que se pone al frente de un conjunto de servidores que atienden una aplicación y, tal como su nombre lo indica, asigna o balancea las solicitudes que llegan de los clientes a los servidores usando algún algoritmo (desde un simple Round Robin hasta algoritmos más sofisticados).

El **balance o balanceo de carga** es un concepto usado en informática que se refiere a la técnica usada para compartir el trabajo a realizar entre varios procesos, ordenadores, discos u otros recursos. Está íntimamente ligado a los sistemas de multiprocesamiento, o que hacen uso de más de una unidad de procesamiento para realizar labores útiles.

El *balance de carga* se mantiene gracias a un *algoritmo* que divide de la manera más equitativa posible el trabajo, para evitar los así denominados *cuellos de botella*.

### **Balance de carga en servidores web**

Uno de los principales problemas de los mayores sitios web en Internet es cómo gestionar las solicitudes de un gran número de usuarios. Se trata de un problema de escalabilidad que surge con el continuo crecimiento del número de usuarios activos en el sistema.

Este servicio se puede brindar tanto con un enrutador como con una computadora con dos placas de red y software específico.

Hay balanceadores de carga tipo round-robin (uno a uno) y por pesos (que son capaces de saber cuál de los nodos está más libre y lanzarle la petición). El más conocido es LVS, sin embargo hay otros, como el Red Hat Piranha.

Y en la plataforma para Windows Server se tiene al ISA Server (Microsoft Internet Security and Acceleration Server).

Existen softwares para el balance de carga, como "Wingate" en donde se pueden añadir dos redes y no es tan difícil de configurar.

### **Cluster de balanceo de carga**

Un **clúster de balanceo de carga** o de **cómputo adaptativo** está compuesto por uno o más ordenadores (llamados *nodos*) que actúan como frontend del cluster, y que se ocupan de repartir las peticiones de servicio que reciba el cluster, a otros ordenadores del cluster que forman el back-end de éste. Un tipo concreto de cluster cuya función es repartir la carga de proceso entre los nodos en lugar de los servicios es el cluster openMosix.

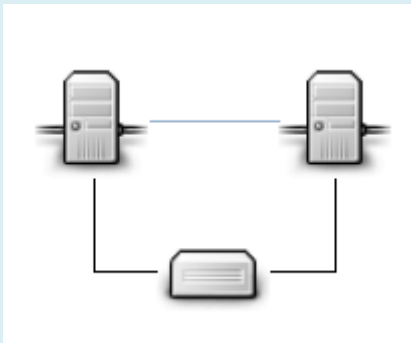
Las características más destacadas de este tipo de cluster son:

- Se puede ampliar su capacidad fácilmente añadiendo más ordenadores al cluster.
- Robustez. Ante la caída de alguno de los ordenadores del cluster el servicio se puede ver mermado, pero mientras haya ordenadores en funcionamiento, éstos seguirán dando servicio.

## - Instalación y configuración de soluciones de alta disponibilidad:

En el mundo empresarial existen muchas aplicaciones que dada su naturaleza crítica deben proporcionar un servicio ininterrumpido de 24 horas al día, 7 días a la semana. Para conseguir estos niveles de disponibilidad se utiliza una configuración avanzada de hardware y software denominada en su conjunto Cluster De Alta Disponibilidad (HA, High Availability).

Esto implica que, en entornos de producción con cargas elevadas de trabajo, una parada del servicio tenga consecuencias mínimas para los usuarios, que podrán seguir trabajando de forma transparente, ya que este sistema tiene la capacidad de ofrecer el servicio asignado a las máquinas de forma continua, incluso en caso de fallo de una de ellas.



La Solución de Alta Disponibilidad de Linalco le ofrece:

- Tolerancia a fallos
- Escalabilidad
- Flexibilidad
- **Tranquilidad**

¿Qué servidores puedo configurar en Alta Disponibilidad?

Esta Solución es capaz de otorgar Alta Disponibilidad a la mayoría de los aplicativos que funcionan bajo servidores Linux:

- **Servidores basados en Software Libre.**  
(Apache, Postfix, MySQL, PostgreSQL, Iptables, Squid, Samba, etc.)
- **Servidores basado en Software Propietario.**  
(Oracle, Informix, desarrollos propietarios, etc.)

## - Virtualización de sistemas:

Posibilidades de la virtualización de sistemas.



Aunque no todo el mundo es consciente de sus posibilidades, la virtualización puede llegar a **solucionar muchos problemas de compatibilidad** que afrontan los usuarios a la hora de migrar de sistemas, pero también es útil para **probar otras alternativas** en el terreno de los **sistemas operativos** o para experimentar con versiones aún en estado muy preliminar. Éstas son algunas de las razones por las que disponer de una aplicación de virtualización en nuestro ordenador es interesante, pero hay muchas más, porque los casos de uso son prácticamente ilimitados. La virtualización está muy de moda en servidores, pero también está convirtiéndose en un recurso muy interesante para los usuarios convencionales de informática.

### Tipos de virtualización

En la actualidad, este campo está dividido en un gran número de soluciones, algunas más adecuadas que otras dependiendo de los distintos escenarios. De hecho, la virtualización que se aplica en empresas es poco parecida a la que podremos disfrutar los usuarios de soluciones como VirtualBox o VMware, de modo que es conveniente explicar las diferentes tecnologías.

- **Virtualización completa:** Dentro de este grupo de soluciones tenemos a su vez otros tres subgrupos que se dirigen a distintos escenarios. En primer lugar, hay una capa de **emulación hardware** que permite acceder a los recursos desde la plataforma virtualizada. Este sistema se usa por ejemplo en **KQEMU**. En segundo lugar, si se opta por **traducción binaria**, se emula un conjunto de instrucciones (por ejemplo, el de la plataforma x86 que usamos en la mayoría de ordenadores de 32 bits). Un desarrollo clásico dentro de este grupo es **VirtualBox**. Finalmente, a la **virtualización clásica** pertenecen desarrollos como **KVM**, que hace uso del kernel Linux. Este grupo

aprovecha el soporte de la virtualización nativa que aportan las tecnologías **Intel VT y AMD-V**.



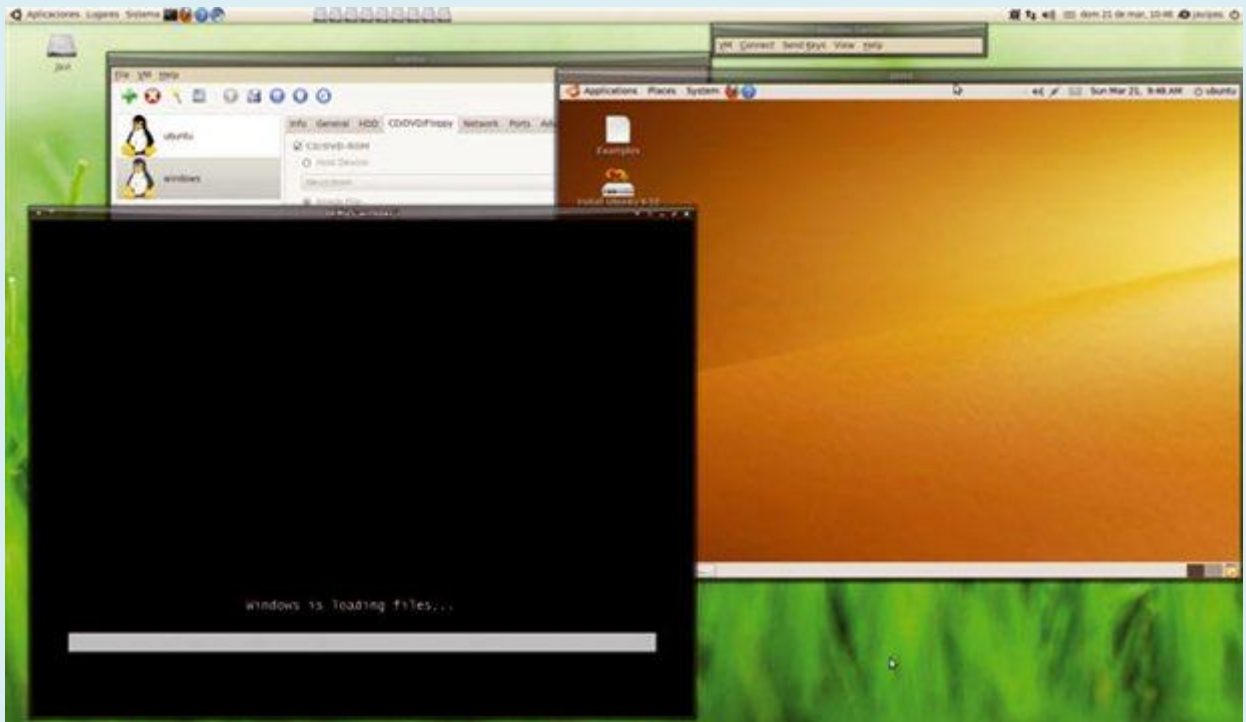
- **Paravirtualización:** Estos sistemas ofrecen mejor comportamiento nativo (el rendimiento de la máquina se acerca al de un sistema instalado de forma convencional en un equipo físico) gracias a una interfaz software que es similar —aunque no idéntico— al del hardware utilizado en la plataforma. **Xen** es uno de los ejemplos clásicos de esta aproximación.

- **Virtualización a nivel de sistema operativo (Contextualización/Contenedores):** En este tipo de técnica, el propio núcleo del sistema operativo permite crear múltiples instancias aisladas del sistema operativo nativo en lugar de una sola. Estas instancias (que a menudo se conocen como contenedores) tienen entidad propia y son vistas como un servidor independiente. Ejemplos muy conocidos son **Linux-VServer** y **OpenVZ**.

Es interesante tener en cuenta también que en estas categorías puede haber cierto uso de técnicas de **emulación parcial**. Sucede en desarrollos como **KQEMU**, **KVM** o **VirtualBox**, mientras que en **OpenVZ** o **Xen** no hay ningún tipo de emulación implicada y la virtualización es «pura». Y precisamente el tipo de aproximación al problema también determina la carga que esas capas de virtualización aportan al rendimiento y la eficiencia final del sistema virtualizado.

Como decíamos, hay soluciones que tienen un **rendimiento en máquina virtual que prácticamente es el mismo que el de las soluciones nativas**; es decir, daría igual instalarlas en un servidor separado o en una máquina virtual. Sin embargo, otras propuestas —precisamente, las protagonistas en esta comparativa— están más

orientadas al usuario final, y aunque tienen ciertas ventajas, también disponen de una eficiencia menor: el sistema operativo instalado en una máquina real siempre se comportará sensiblemente mejor.



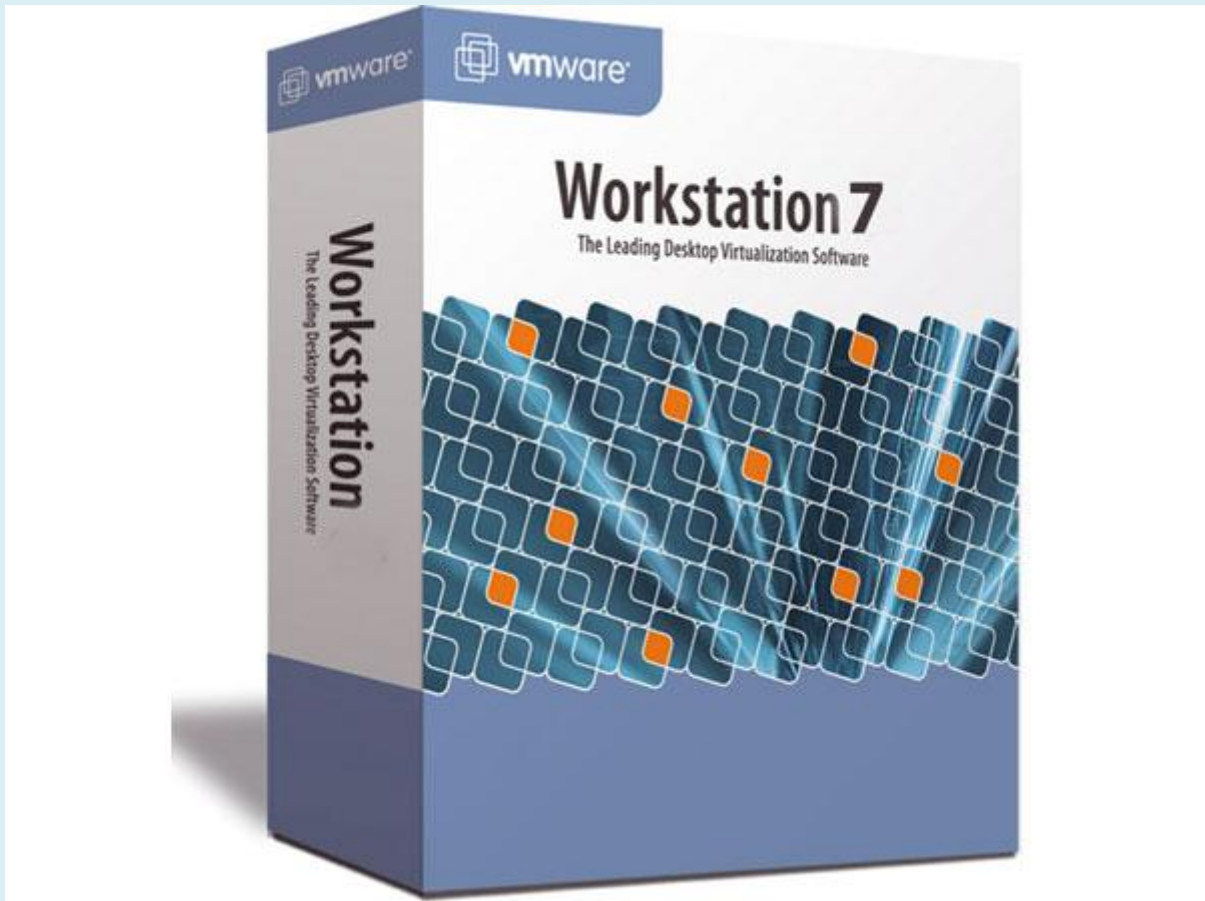
### ¿Para qué sirven?

Las aplicaciones prácticas de la instalación de un software de virtualización son múltiples, pero hay que tener claro qué se puede hacer y qué no con este tipo de soluciones. El escenario más común es el de poder **ejecutar dos o más sistemas operativos distintos al mismo tiempo**. Muchos usuarios están interesados, e incluso obligados por cuestiones de trabajo, en iniciar una sesión en dos sistemas distintos en los que corren aplicaciones diferentes.

Por un lado, está el uso casi **por diversión**, en el que por ejemplo instalaríamos en nuestro sistema operativo Windows una máquina virtual con Ubuntu **para probar** esta distribución Linux. Aunque hay muchos métodos de evaluar el funcionamiento de soluciones Linux –LiveCDs o Wubi, por ejemplo–, las máquinas virtuales son especialmente interesantes para ese propósito de prueba sin tener que preocuparnos de posibles descuidos o de la eliminación de datos de forma involuntaria. Con las máquinas virtuales ese riesgo desaparece totalmente.

El segundo escenario clásico es el de la **ejecución de aplicaciones** que necesitamos para trabajar y **que no funcionan en nuestro entorno habitual**. Es lo que ocurre cuando, por ejemplo, un usuario de Mac OS X está forzado a trabajar con cierta versión de cierta aplicación que solo funciona bajo Windows. En ese caso, llegan al rescate soluciones como las **VMware, Parallels o VirtualBox**, que permiten instalar esas máquinas virtuales con Windows para acceder a las aplicaciones que necesitábamos sin problemas y, como en el caso anterior, sin riesgos.

En este último escenario hay que destacar otra ayuda muy interesante: los **modos de coherencia**, que proporcionan soluciones como **VMware Fusion** o **Parallels Desktop**, que permiten que el escritorio de Mac OS X y el de Windows se entremezclen, de forma que el usuario no tenga que preocuparse nunca por si una aplicación está ejecutándose en el **sistema operativo anfitrión** (el «bueno», el que arrancamos cuando encendemos el ordenador) o en el **sistema operativo invitado** (que es el que se ejecuta bajo la máquina virtual).



### Sus limitaciones

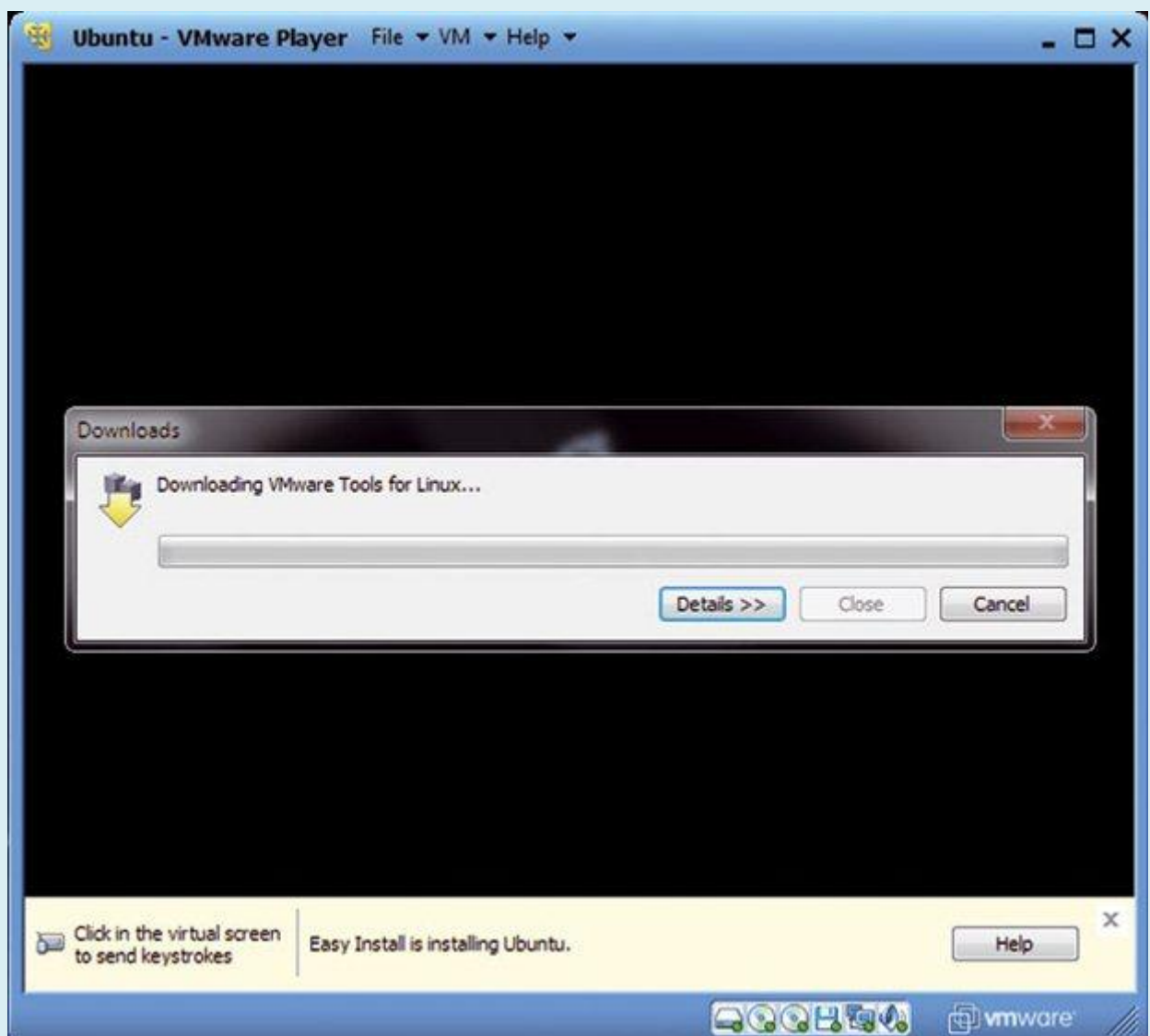
Aunque gracias a las extensiones de los procesadores de Intel y AMD el rendimiento de los sistemas que corren bajo una máquina virtual son excelentes, hay que tener en cuenta que dicha producción nunca será tan notable como la que pueda ofrecer una instalación nativa, y aquí entran en juego los distintos tipos de virtualización de los que hablábamos anteriormente. Mientras que **la virtualización a nivel de sistema operativo** con conceptos como contenedores es muy **eficiente en el uso de recursos** (se habla de un 97% del rendimiento del SO real), en otras soluciones menos especializadas ese rendimiento puede ser menor y estar entre el 70% y el 90% según la situación.

Eso puede ser un obstáculo, aunque afortunadamente las máquinas virtuales permiten afinar el uso de recursos: **podremos establecer cuánta memoria y cuánto disco duro le asignamos a cada máquina virtual**. Pero además, si contamos con un procesador

multinúcleo –algo muy extendido hoy en día–, podremos hacer que también la máquina virtual aproveche esos dos, cuatro o más cores para sus propias tareas, convirtiendo nuestra máquina virtual en una solución SMP (*Symmetric MultiProcessing*).

Ese **soporte hardware**, no obstante, **no es completo**. Y es que las tecnologías de virtualización no podrán sacar todo el partido a nuestro hardware. Un apartado claro en el que esto se demuestra es **en aplicaciones gráficas** exigentes y, sobre todo, en **juegos**. Que nadie piense que teniendo Linux e instalando Windows 7 en una máquina virtual tendrá acceso a aplicaciones y juegos DirectX 11. Esa API no es compatible con los últimos desarrollos de virtualización.

De hecho, en VirtualBox o VMware se permite la ejecución de aplicaciones y juegos OpenGL 2.0 y Direct3D 8/9, pero nada de Direct3D 10/11. No esperéis grandes rendimientos de esos juegos, aunque sí es posible que disfrutéis, por ejemplo, de títulos de estrategia o RPGs gracias a su menores requerimientos gráficos. En definitiva, si queréis ejecutar juegos de otras plataformas, la virtualización no es ni mucho menos la opción más adecuada.



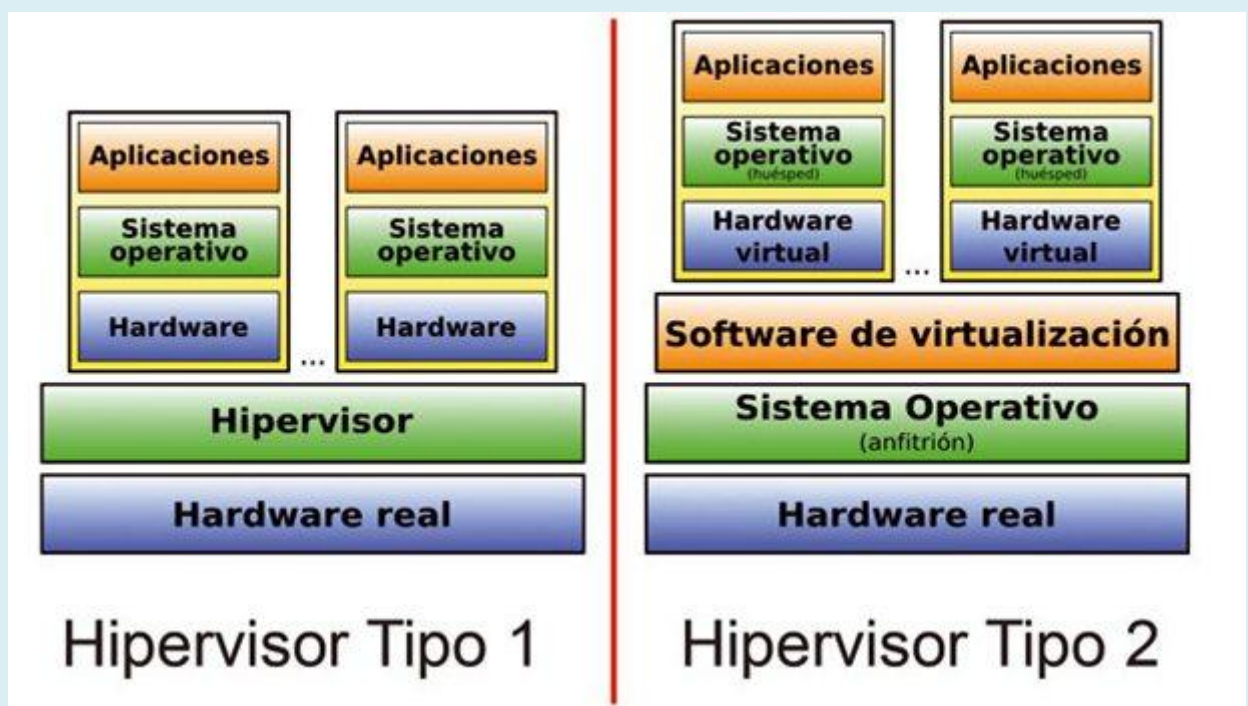


**Hipervisor, clave de la virtualización**

Este componente software, que también suele conocerse como **monitor de máquina virtual**, es el encargado de que se puedan utilizar múltiples sistemas operativos en un equipo. En la actualidad los hipervisores se pueden clasificar en dos grandes grupos:

- **Tipo 1 Nativo, bare-metal o unhosted.** Este tipo de hipervisores **se ejecutan directamente sobre el hardware real** del equipo para controlar ese hardware y monitorizar los sistemas operativos virtualizados. Las máquinas virtuales que corren de forma simultánea lo hacen en otro nivel que está por encima del hipervisor. Ejemplos: VMware ESXi, VMware ESX, Xen, Citrix XenServer y Microsoft Hyper-V Server.
- **Tipo 2 Hosted:** En este caso, la aplicación hipervisora o monitora **se ejecuta sobre un sistema operativo convencional** para luego virtualizar diversos sistemas operativos. La virtualización se sitúa en una capa más alejada del hardware y el rendimiento del hipervisor es menor que en los anteriores. Ejemplos: Sun VirtualBox, Sun VirtualBox OSE, VMware Workstation, VMware Server, VMware Player, QEMU, Microsoft Virtual PC y Microsoft Virtual Server.

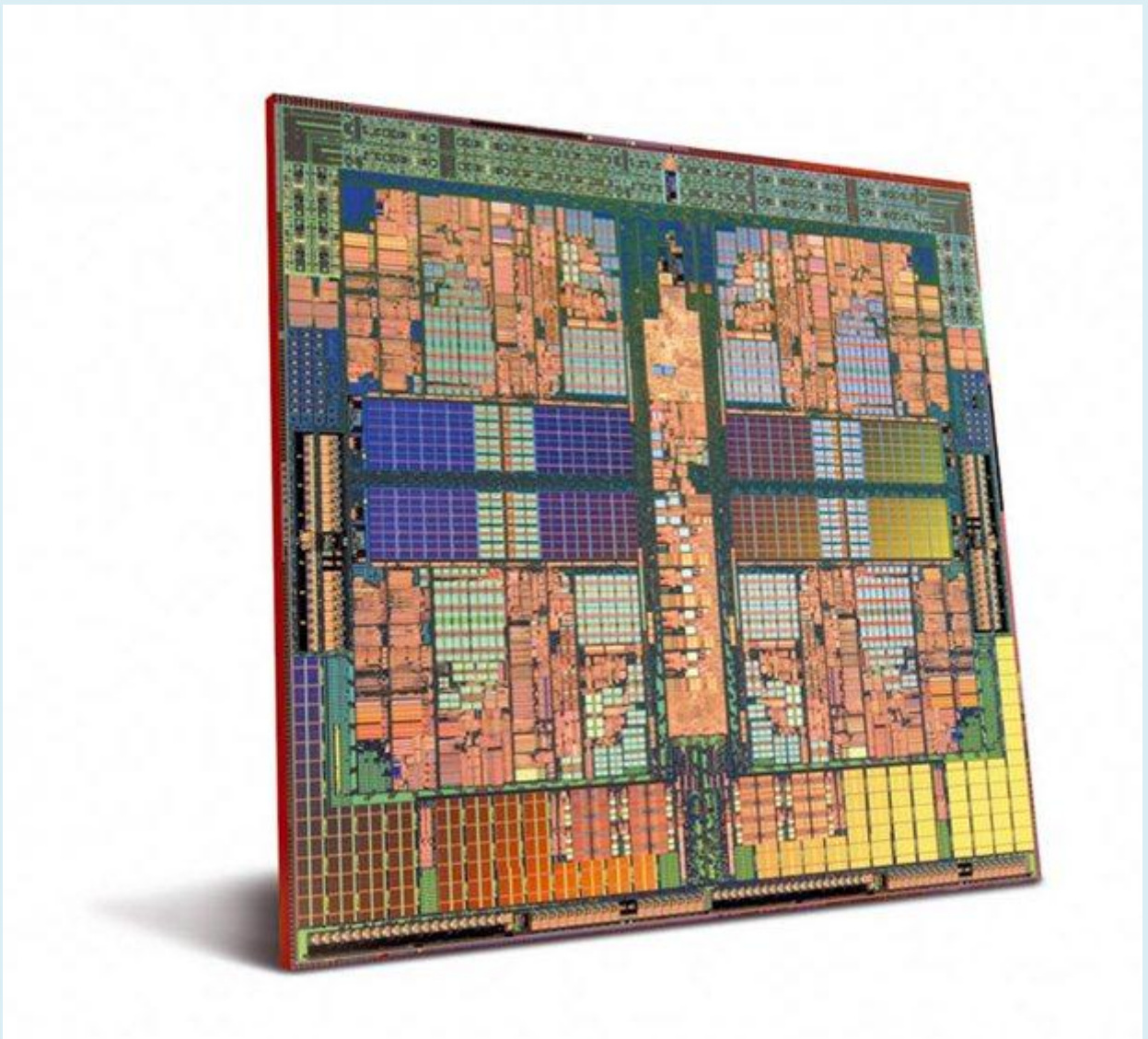
Aunque algunos desarrollos se pueden clasificar fácilmente en uno u otro grupo, hay cierto debate a la hora de englobar otras soluciones como hipervisores de Tipo 1 (teóricamente más eficientes) o de Tipo 2 (más orientados al usuario final). Por ejemplo, los desarrolladores de Xen afirman que KVM es un hipervisor de Tipo 2, como también podría serlo Microsoft Virtual Server. Sin embargo, cada hipervisor aprovecha ciertas fortalezas (en el caso de KVM, el kernel Linux y su versatilidad) y, por lo tanto, englobar a cada desarrollo en uno u otro grupo es para muchos una cuestión meramente académica.



### Tu procesador te ayuda

Hace tiempo que tanto Intel como AMD fabrican procesadores con una serie de extensiones que son vitales para acelerar los procesos de virtualización, y para soportar algunas características más que interesantes. En el caso de AMD, su **tecnología AMD-V**, que fue desarrollada con el nombre en clave de Pacifica, está presente en la práctica totalidad de sus microprocesadores actuales (los Sempron suelen no estar preparados con esta capacidad).

Por su parte, también **Intel** presentó hace tiempo su propia propuesta al respecto, la tecnología llamada **VT-x** (antes conocida como Vanderpool) y que de nuevo está presente en casi todos los microprocesadores actuales, aunque cuidado, no en todos. Si vais a aprovecharos de los programas de virtualización, deberíais asegurarnos de que el procesador de vuestro ordenador dispone de dichas extensiones, algo fácil de comprobar en los sitios web de los fabricantes.



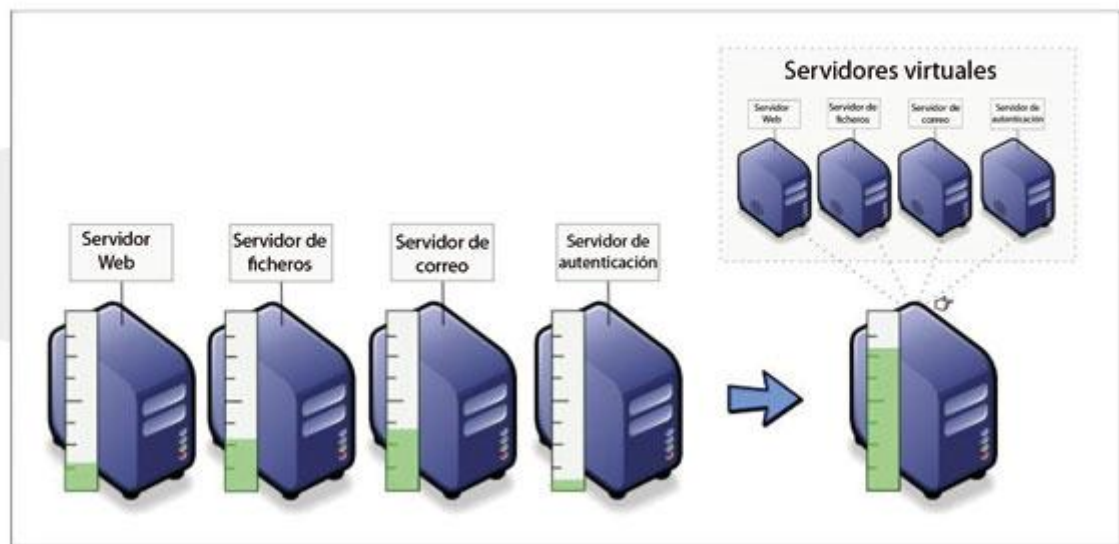
### Virtualización en empresas, otro mundo

Aunque en esta comparativa nos hemos centrado en las soluciones para el usuario final, las tecnologías de virtualización tienen especial protagonismo en el mundo corporativo, donde temas como la **consolidación de servidores**, la posibilidad de **balancear cargas** o la **virtualización de escritorios** (que ofrece una nueva visión de los Thin Clients tradicionales) están a la orden del día.

Las soluciones de empresas son desarrolladas por compañías muy experimentadas como **XenSource**, **VMware**, **Citrix**, **Microsoft** o **Red Hat** (que se apoya en KVM tras la compra de Qumranet, aunque antes lo hacía en Xen) y las implantaciones son más complejas, como también lo son los productos, que no tienen demasiado que ver con las soluciones para usuario final.

## ¿Por qué virtualizar una infraestructura en servidores?

La consolidación de servidores es el argumento tradicional



### Diez razones por las que virtualizar

Es muy fácil encontrar razones para aprovechar este tipo de soluciones, pero a continuación os proponemos 10 de ellas.

1. Reducción de los costes de compra y mantenimiento
2. Compatibilidad con aplicaciones y sistemas operativos antiguos
3. Pruebas en entornos aislados y seguros que no afectan al resto de componentes
4. Entorno de desarrollo barato para programadores software en distintas plataformas de forma simultánea
5. Gestión y control centralizado de recursos

6. Procedimientos de *backup* y restauración muy sencillos
7. Migraciones en vivo de máquinas virtuales entre distintos PCs o servidores físicos
8. Fácil recuperación/reinicio en caso de posibles caídas
9. Alta disponibilidad
10. Fácil puesta en marcha de servicios de Cloud Computing

### La opinión de PC Actual: más fácil y barato que nunca

Hace unos años, la virtualización era una tecnología a la que era muy difícil acceder, tanto por las pocas opciones como por el precio de las soluciones disponibles. Prácticamente, VMware no tenía rival en un mercado que poco a poco fue ganando adeptos debido a los beneficios que proporciona en consolidación de servidores y control de gastos. Curiosamente, esas ventajas no solo se traducen en beneficios para la empresa, ya que pronto se pudieron extender a máquinas de escritorio como PCs y portátiles. El mercado empezó a animarse y, hoy en día, las tendencias favorecen tanto a empresas como a usuarios finales.

### Tres claros destacados

Es evidente que existen **tres grandes protagonistas** en este segmento, al menos cuando hablamos de **usuario final**. **VMware** compite con sus productos Workstation, Player e incluso Server –un buen paso previo (y gratuito) para empresas que quieren comenzar a explorar este terreno–, mientras que VirtualBox y Parallels disponen de sus propias soluciones con una única diferencia en el caso de **Parallels**: se trata de una **solución de pago**, mientras que los desarrollos de **VMware y VirtualBox tienen coste cero**.

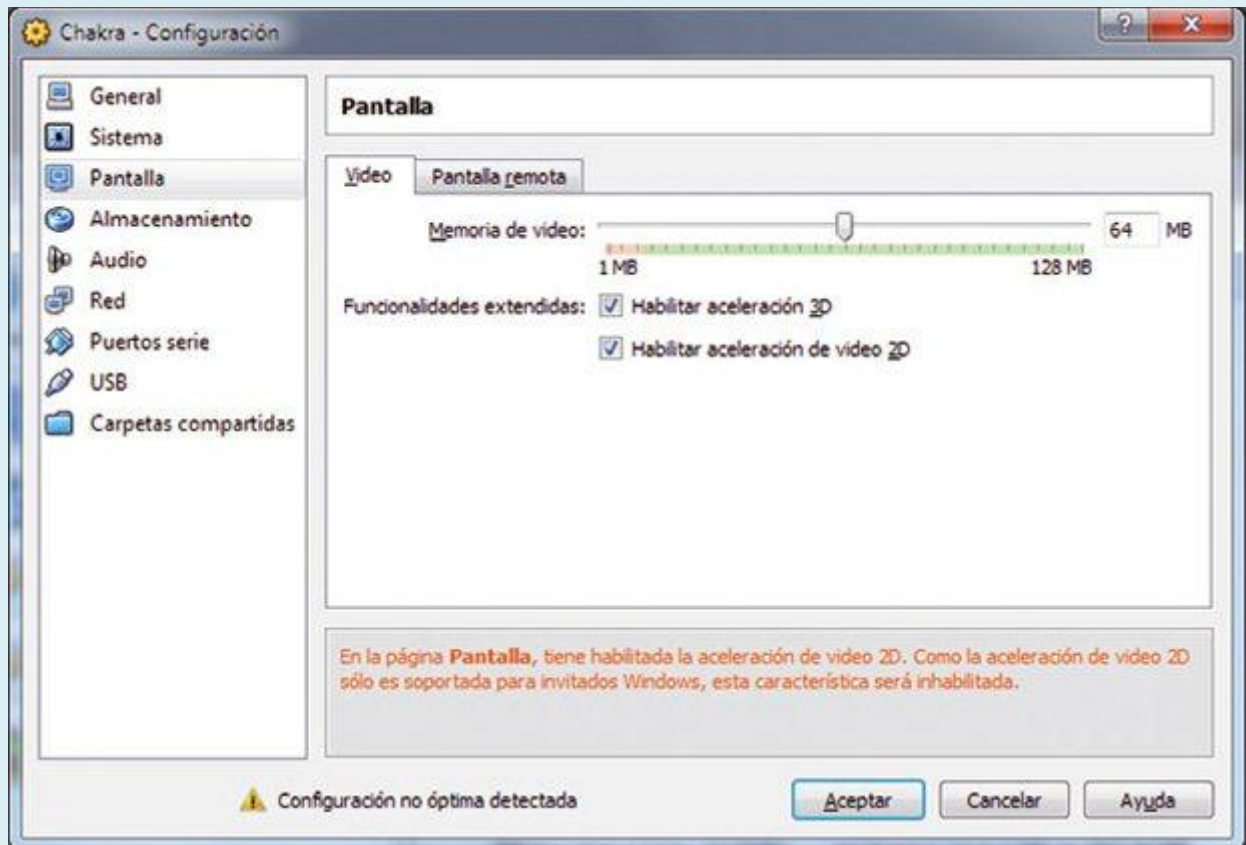
De hecho, ese ha sido el único factor que ha perjudicado a Parallels, que quizás en la plataforma Windows no está tan pulido como sus competidores, pero que está claro que en Mac OS X compite de tú a tú con VMware Fusion. Es difícil elegir un **vencedor** entre VMware Player y VirtualBox, pero nos hemos decantado por **VirtualBox** por una ventaja poco tangible para muchos pero que nosotros valoramos: se trata de un desarrollo **Open Source**, y esa libertad y filosofía hacen que la balanza se desequilibre a favor del desarrollo de Sun/Oracle.



### En Linux, otro mundo

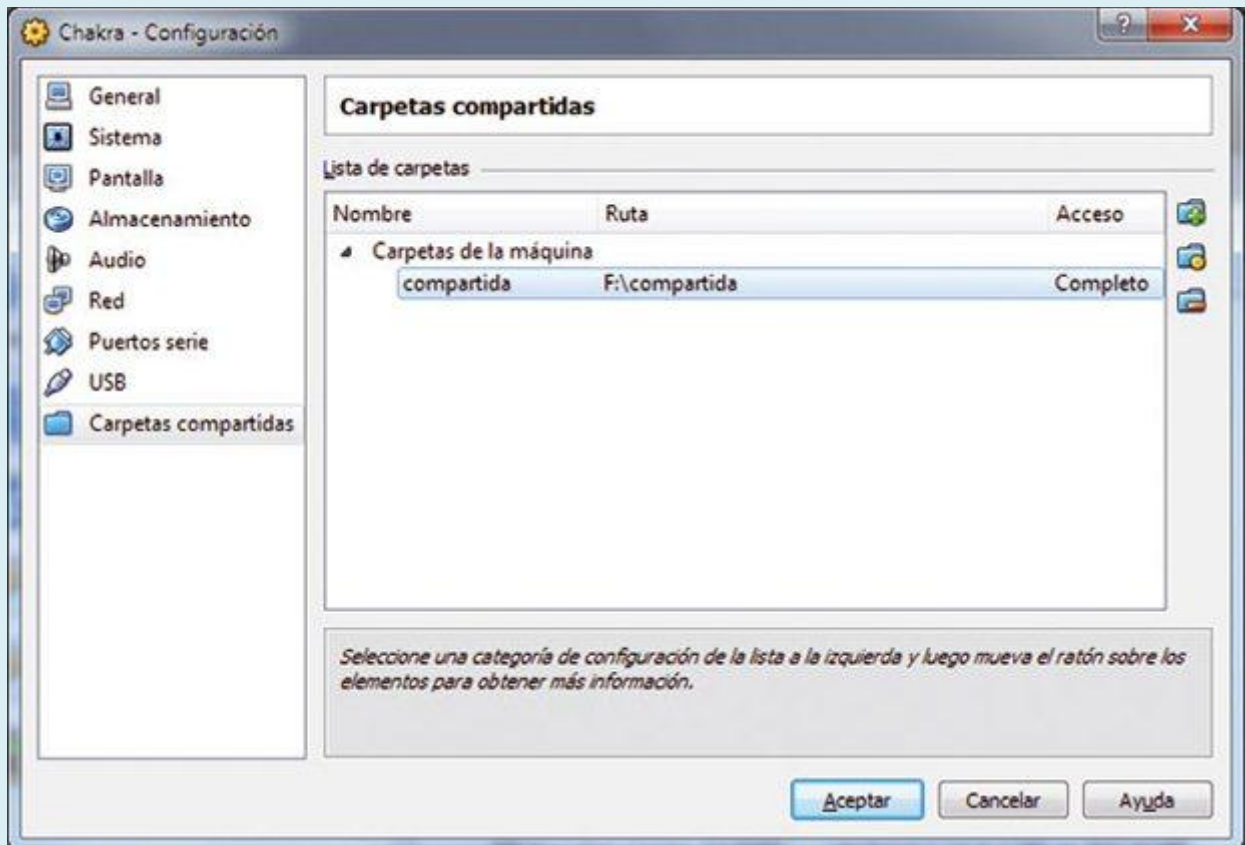
Del resto de soluciones comentadas habría que dejar a un lado **XP Mode**, un desarrollo de Microsoft que ha sabido aprovechar las ventajas de Virtual PC para ofrecer **una aplicación práctica a la hora de migrar de Windows XP a Windows 7**. Sin embargo, es **demasiado específico**, y no tiene apenas nada que ver con las tres propuestas más destacadas, y mucho menos con las alternativas del mundo Linux.

KVM, Xen y Linux-VServer son tres formas distintas de acercarnos a la virtualización de sistemas, cada una con sus ventajas e inconvenientes. **KVM** es la más «cercana» al **usuario final**, ya que existen ciertos asistentes que permiten crear y gestionar máquinas virtuales como con VirtualBox, VMware o Parallels. En **Xen** también es posible acceder a tales paneles de control, pero su puesta en marcha es **más compleja**, como la de **Linux-VServer**, ambas **poco aptas para usuarios noveles** y para los que no se hayan «peleado» con el kernel Linux. Sin embargo, los beneficios en rendimiento son muy significativos, lo que precisamente justifica el **éxito en grandes implantaciones**. Si nos restringimos a los usuarios finales, lo tenemos claro: VirtualBox es nuestra preferida.



### Lo mejor: Muchos avances

El rendimiento de estas soluciones se ha mejorado gracias al soporte de extensiones de virtualización por parte de Intel y AMD, pero estos desarrollos también han sabido ir mejorando su soporte hardware –vital para una emulación eficiente–. Así, hoy en día, los modos de coherencia o de integración de escritorios proporcionan una serie de posibilidades muy llamativas para el usuario final. El soporte de modos 3D –aunque no para jugar a juegos en máquinas virtuales– y de todo tipo de periféricos nos permiten explotar dichas tecnologías para resolver situaciones de migración, pruebas o consolidación.



### Lo peor: Algunas limitaciones

En general, nos hemos encontrado con que las opciones de compartir carpetas entre las máquinas anfitrión y las máquinas invitadas son limitadas y, salvo en VMware, la configuración de dichas carpetas es algo confusa. También es cierto que las herramientas de integración de las máquinas virtuales (VMware Tools, VirtualBox Guest Additions...) no están disponibles para todos los sistemas operativos invitados, lo que impide un funcionamiento más suave para estas soluciones si queremos probar un sistema operativo menos extendido o soportado.

### Herramientas para la virtualización.



[Parallels Desktop 4, con un desarrollo muy «mac»](#)

Fabricante: **Parallels** / Precio: **79,99€**

Aunque Parallels apuesta especialmente fuerte por el mundo Mac, que ya cuenta con la versión Desktop 5, hace tiempo que sus desarrolladores también ofrecen su aplicación de virtualización tanto para Windows como para Linux



### [VirtualBox 3.1.4 satisface todas las necesidades](#)

Fabricante: **Oracle** / Precio: **0€**

VirtualBox es un desarrollo que Sun (recientemente adquirida por Oracle) ha cuidado y mimado especialmente, y eso ha permitido contar con una aplicación sorprendentemente eficiente y versátil



### [VMware Player 3.0.1, perfecto para usuarios noveles](#)

Fabricante: **VMware** / Precio: **0€**

VMware Player es una herramienta que hasta no hace mucho tenía una seria limitación: no era posible crear máquinas virtuales directamente en la aplicación. Sin embargo, ese problema se erradicó en las últimas versiones, que han convertido a esta aplicación en una hermana pequeña de VMware Workstation





[KVM, solución de virtualización para Linux](#)

Fabricante: **KVM** / Precio: **0€**

Kernel-based Virtual Machine (KVM) es un desarrollo que actúa como un módulo del kernel que proporciona la infraestructura básica de virtualización, además de un módulo adicional para el procesador

```

jv@pragmatico:~$ sudo ./linux/virtual-server.py
Fixing usability of /proc entries for Linux-VServer
Starting Linux-VServer in background
jv@pragmatico:~$ sudo vserver njesta enter
# cat /proc/mounts | ls -la
Total 76
drwxr-xr-x 29 root root 4096 Mar 22 08:18 .
drwxr-xr-x 29 root root 4096 Mar 22 08:18 ..
drwxr-xr-x 2 root root 4096 Mar 22 08:17 bin
drwxr-xr-x 2 root root 4096 Apr 25 04:41 boot
drwxr-xr-x 4 root root 4096 Mar 22 08:24 dev
drwxr-xr-x 39 root root 4096 Mar 22 08:18 etc
drwxr-xr-x 2 root root 4096 Apr 25 04:41 home
drwxr-xr-x 9 root root 4096 Mar 22 08:17 lib
drwxr-xr-x 2 root root 4096 Mar 22 08:18 media
drwxr-xr-x 2 root root 4096 Apr 25 04:41 mnt
drwxr-xr-x 2 root root 4096 Mar 22 08:18 opt
    
```

- 

[Linux-VServer, una alternativa Open Source](#)

Fabricante: **Linux-VServer** / Precio: **0€**

El sistema operativo creado por Sun ha sido alabado por su concepción de la gestión de los llamados containers, máquinas virtuales aisladas e independientes que permiten ejecutar varias instancias de uno o varios sistemas operativos de una forma muy controlada



- 

[Xen, muy eficiente en entornos corporativos](#)

Fabricante: **Xen** / Precio: **0€**

Xen fue uno de los primeros desarrollos Open Source que puso en jaque a las propuestas empresariales que ya estaban asentadas. Su tecnología de paravirtualización permite obtener rendimientos muy notables y cercanos a los sistemas nativos



- 

[XP Mode para Windows 7, migración de XP a W7 gracias a la virtualización](#)

Fabricante: **Microsoft** / Precio: **0€**

Esta herramienta se sitúa como una opción adicional a la hora de ejecutar aplicaciones de Windows XP que no acaban de funcionar bien en Windows 7 y que tampoco se ajustan a las técnicas de compatibilidad que se pueden establecer en las propiedades de los archivos ejecutables

Configuración y utilización de máquinas virtuales