



Universidad Nacional de La Matanza

**DEPARTAMENTO DE
INGENIERÍA E INVESTIGACIONES TECNOLÓGICAS**

Sistemas Operativos

Trabajo Practico Nro. 3:

Diagnósticos, Virtualización y Storage

Profesora a cargo: Villamayor, Alexis

Integrantes: Galasso, Agustina

Benitez, Sabrina

Roccetti, Carla

Di Ludovico, Maximiliano

Famiglietti, Damián

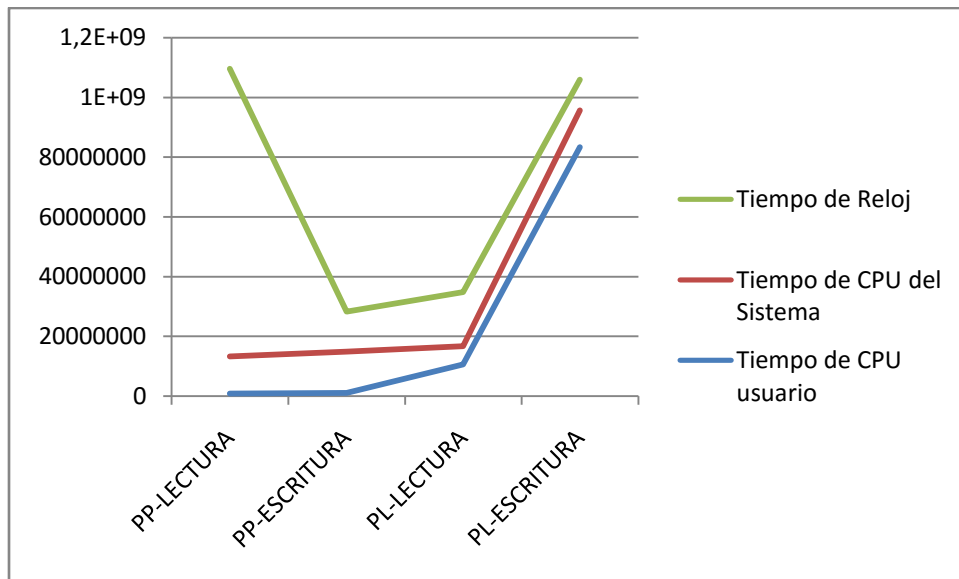
**1er CUATRIMESTRE - AÑO
2016**

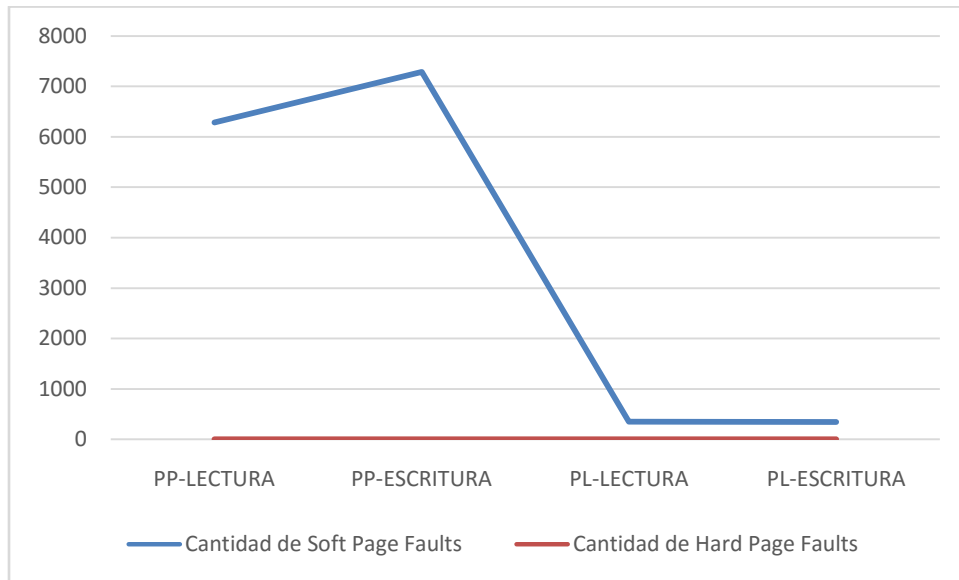
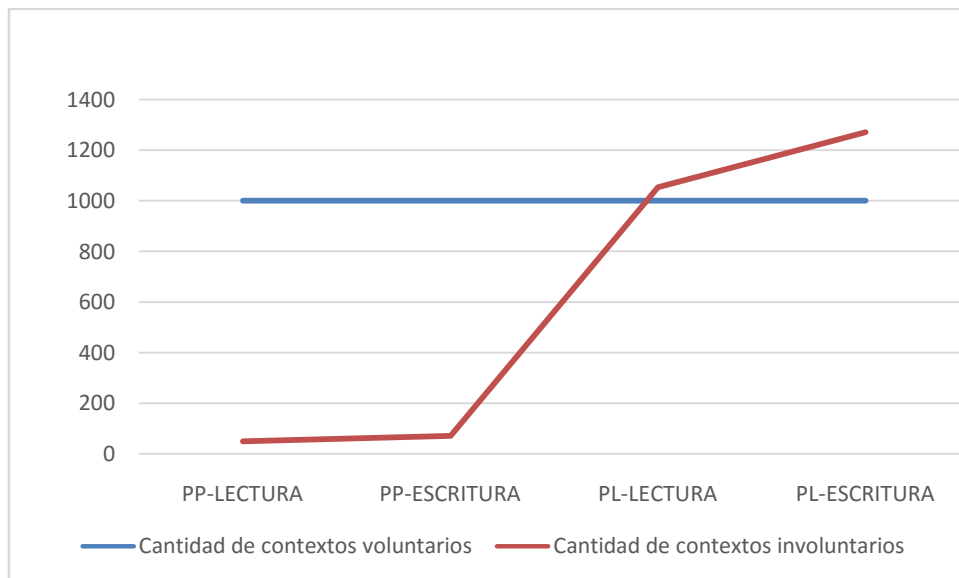
Ejercicio 1- Performance de procesos pesados y livianos

Una vez creadas las unidades de procedimiento y ejecutadas de a una por vez, obtuvimos los siguientes resultados:

CASOS	Tiempo de CPU usuario	Tiempo de CPU del Sistema	Tiempo de Reloj	Cantidad de Soft Page Faults	Cantidad de Hard Page Faults	Cantidad de señales recibidas	Cantidad de contextos voluntarios	Cantidad de contextos involuntarios
PP-LECTURA	8670000	124132000	962483086	6286	0	0	1000	49
PP-ESCRITURA	10292000	138941000	133541281	7289	0	0	1000	70
PL-LECTURA	105762000	61069000	180816821	350	0	0	1000	1053
PL-ESCRITURA	832868000	123386000	102490700	344	0	0	1000	1271

Comparación respecto a los tiempos de CPU usuario, CPU del sistema y el tiempo de reloj



Comparación respecto a la cantidad de Soft Page Faults y Hard Page Faults**Comparación respecto a la cantidad de contextos voluntarios e involuntarios**

Conclusiones

La principal diferencia entre un proceso pesado y uno liviano es que el primero es cualquier programa en ejecución y un hilo es una unidad básica de utilización de la CPU, y consiste en un contador de programa, un juego de registros y un espacio de pila. Los hilos permiten la ejecución concurrente de varias secuencias de instrucciones asociadas a diferentes funciones dentro de un mismo proceso, compartiendo un mismo espacio de direcciones y las mismas estructuras de datos del núcleo (variables globales, espacio de direcciones, ficheros abiertos, etc.). En cambio un proceso pesado es igual a una tarea con un solo hilo y al tener cada uno su propio espacio de direccionamiento, la comunicación entre ellos y los cambios de contexto resultan costosos. Por lo tanto la performance (en lectura principalmente) de los hilos es mucho mayor ya que al compartir memoria y recursos, se pueden comunicar entre sí sin invocar el núcleo del SO, haciendo más rápida la comunicación entre procesos, lo que genera menos overhead. Se tarda menos tiempo en crear un nuevo hilo de un proceso que ya existe, en terminarlo, y en hacer un cambio de contexto entre hilos de un mismo proceso (su TCB es más pequeño que los PCB de los procesos).

La ventaja de los procesos pesados es que garantizan protección. Si un proceso falla, los demás procesos continúan sin problemas. En cambio si un thread falla, esto causa la falla de todos los demás threads que comparten el mismo procesador.

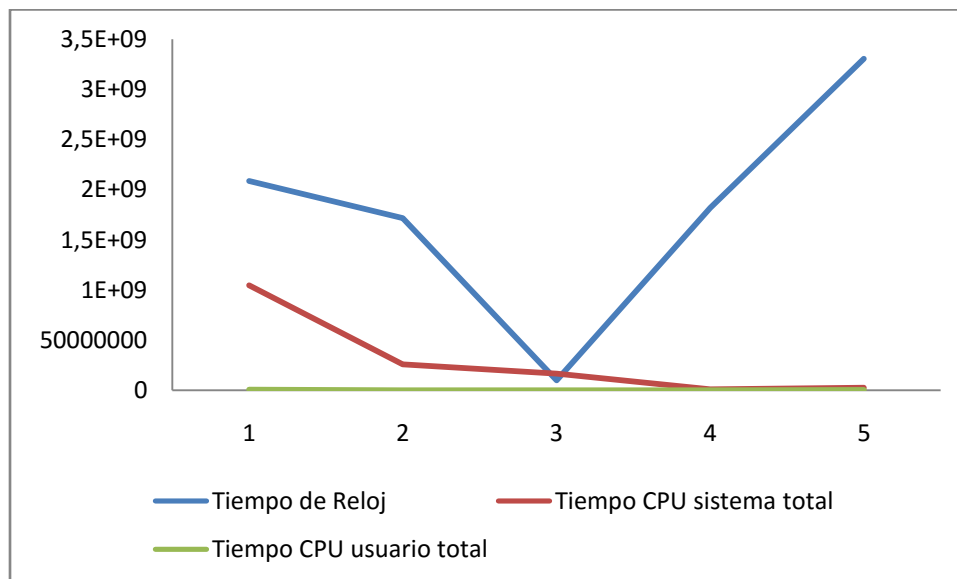
La cantidad de Soft Page Faults generada por los Procesos Livianos es menor con respecto a los Procesos Pesados, tanto en la lectura como en la escritura. Esto se debe a que al compartir memoria con el proceso padre, el Proceso Liviano solo debe apuntar a las direcciones del padre y genera páginas por los valores que declaró en su código. En cambio el Proceso Pesado copia todos los valores del vector para crear el nuevo proceso, a raíz de esto, se crean más páginas nuevas, y esto produce más page fault que en el Liviano.

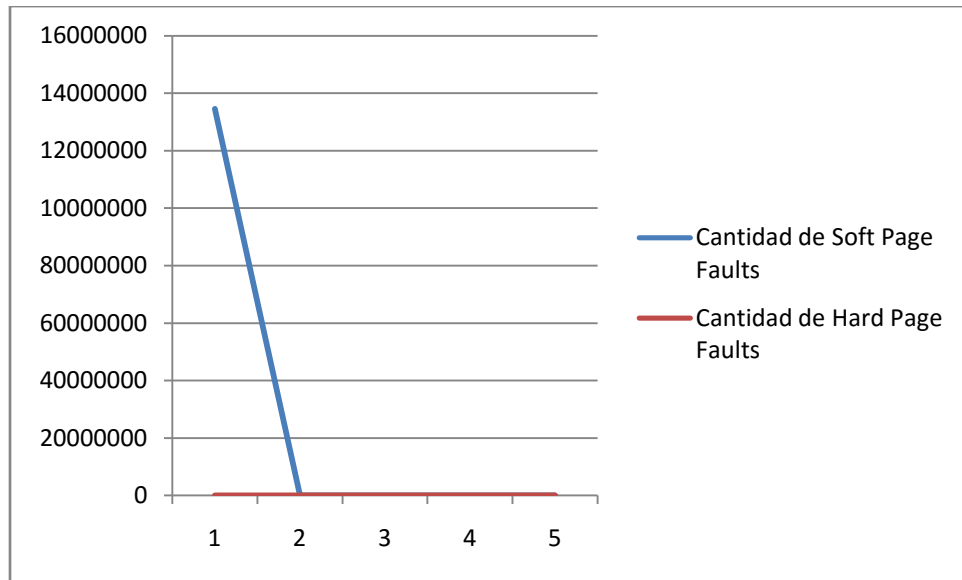
Cuando ocurre un cambio de contexto voluntario el proceso realiza una llamada al sistema que implica esperar por un evento y se cambia su estado de en "ejecución" a "bloqueado". Esto ocurre por un motivo de eficiencia en el uso del procesador. Por lo contrario, en un cambio involuntario el S.O. le quita la UCP al proceso y este cambia su estado de en "ejecución" a "listo". Esto ocurre para realizar un reparto del procesador. Con respecto a los cambios de contextos voluntarios, ambos procesos arrojan un valor de 1000, lo cual serían uno por cada thread o fork invocado. En cambio, la cantidad de cambios de contextos involuntarios ronda entre 40 y 1280 aproximadamente. Como se puede observar en la tabla, el Proceso Liviano obtuvo cantidades mucho mayores que el Proceso Pesado.

Ejercicio 2- Performance de datos compartidos

Una vez creadas las unidades de procedimiento y ejecutadas de a una por vez, obtuvimos los siguientes resultados:

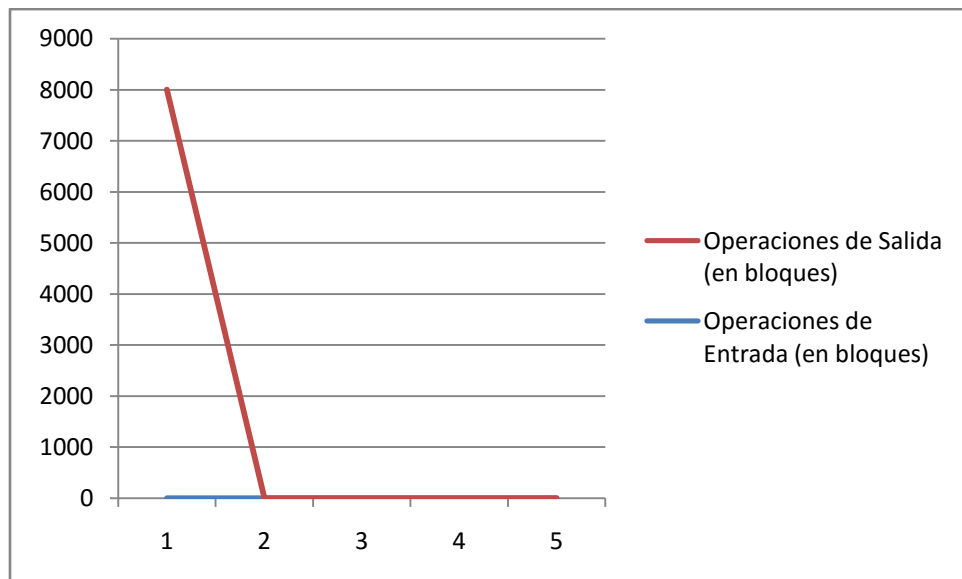
MÉTRICA	CASO 1	CASO 2	CASO 3	CASO 4	CASO 5
Tiempo de Reloj	2084352028 ns	1715872568 ns	99345500 ns	1817629223ns	3302941723 ns
Tiempo CPU sistema total	1048000000 ns	260000000 ns	168000000 ns	12000000 ns	28001000 ns
Tiempo CPU usuario total	8000000 ns	0	0	0	4000000 ns
Cantidad de Soft Page Faults	134524916	0	222	250	249
Cantidad de Hard Page Faults	0	0	0	0	0
Operaciones de Entrada (en bloques)	0	0	0	0	0
Operaciones de Salida (en bloques)	8000	0	0	0	0
Mensajes IPC enviados	0	0	0	0	0
Mensajes IPC recibidos	0	0	0	0	0





Hard page fault: error de página que se produce cuando la página no se encuentra en la memoria física.

Soft page fault: se produce cuando la página está residente en otro lugar en la memoria. Por ejemplo, la página puede estar en el conjunto de trabajo de otro proceso.



Conclusiones

En el caso 3, la memoria compartida entre los dos procesos, un programa la ve exactamente de la misma forma que si fuera una memoria normal, las paginas se encuentran en memoria y las mismas no necesitan realizar entradas y salidas al disco. Además, las operaciones de lectura y escritura (tiempo de usuario) son muy rápidas ya que requiere llamadas al sistema (tiempo de sistemas) solo para la inicialización.

En cambio, en el Caso 1, el acceso a los ficheros es lento debido a que tiene que leer y escribir en el disco, realizando muchas operaciones de entrada y salida y consecuentemente el traslado de los datos a memoria (soft page fault) y efectuar llamadas al sistema.

En el Caso 2, para la comunicación a través de una tubería FIFO, no es adecuado para trabajo interactivos ya que retrasa todos los trabajos debido a su estructura (encola un mensaje detrás del otro), también hay problemas de tiempos y coordinación en una implementación de este tipo.

Mientras que para los Casos 4 y 5 son más lentos debido a los tiempos de comunicación (conexión, protocolos, servicios del sistema), tiempo de respuesta y manejo de los datos en el sistema (servidor o cliente).

Para la comunicación en entornos distribuidos es posible con el método Socket (cliente-servidor) en máquinas distintas.

Estas son algunas de las características que tiene que tener la instalación:

- . Que ambas máquinas estén conectadas bajo una misma red LAN;
- . Que el cliente tiene que conocer la IP de la máquina del servidor;
- . Que ambos tienen que trabajar bajo la habilitación del mismo número de puerto;
- . Que ambas máquinas, tanto la del cliente como la del servidor, tengan diferente IP;

Acotación: Esto se puede lograr tanto en máquinas que tengan instalado el sistema operativo Linux de forma nativa como usando máquina virtual.

Se usa el protocolo **TCP** del protocolo **TCP/IP**, para gestionar la conexión.

Comunicación entre procesosóInterProcessCommunication (IPC) es el conjunto de mecanismos claves en la compartición de datos (intercomunicación) y sincronización entre distintos procesos. Los procesos que se ejecutan concurrentemente pueden ser independientes o cooperativos. Los procesos independientes cuando no pueden alterar o verse alterados por otros procesos que están en ejecución. En cambio, los procesos cooperativos pueden verse afectados y afectar a los demás procesos.

Hay varios métodos de IPC disponibles:

Archivo

- Es simplemente un flujo unidimensional de bits.
- Es tratado por el sistema operativo como una única unidad lógica.
- Depende del software que se ejecuta.

FIFOs (pipes con nombre):

- Las tuberías con nombre existen en el sistema de archivos como un archivo de dispositivo especial.
- Los procesos de diferentes padres pueden compartir datos mediante una tubería con nombre.
- Una vez finalizadas todas las operaciones de E/S, la tubería con nombre permanece en el sistema de archivos para un uso posterior.

Segmentos de memoria compartida

- La memoria que habitualmente puede direccionar un proceso a través de su espacio de direcciones virtuales es local a ese proceso.
- Cualquier intento de direccionar esa memoria desde otro proceso provoca una violación de segmento.
- La memoria compartida permite a dos o más procesos compartir datos a través de un segmento de memoria.

Sockets

- Es el extremo de una conexión y tiene asociado un nombre.
- Dos procesos que se quieren comunicar entre si se ponen de acuerdo en dicho nombre

¿Qué pasa si queremos comunicar dos procesos en computadoras distintas?

- En vez de usar nombres de archivo para establecer la comunicación, se usa la dirección IP y un número de puerto.
- Determinados servicios usan puertos estándar: 80 para HTTP, 22 para SSH, etc.

Tipos de comunicación:

TCP: Garantiza la llegada de los datos, el orden relativo y la integridad de los mismos, etc.

UDP: Se envían paquetes independientes uno de otro. No hay garantías de ningún tipo respecto de su arribo, ni la corrección de los datos que llevan.

Otro de los conceptos involucrados es:

La mutua Exclusión:

Es la comunicación requerida entre dos o más procesos que se están ejecutando en paralelo y que necesitan a la vez el uso de un recurso no compartible. Consiste en asignar el recurso no compartible a sólo uno de los procesos, mientras que los otros deben permanecer a la espera hasta que finalice la utilización de dicho recurso por el proceso al que se le asignó. Cuando este proceso termine, el recurso será asignado a uno de los procesos en espera. Se asegura el correcto uso del recurso.

Región Crítica:

Es la sección del código de un proceso el cual accede a un recurso compartido que no debe ser utilizado por más de un recurso a la vez. Por lo general termina en un tiempo determinado. Solamente un proceso puede estar en su sección crítica a la vez. El método más común para prevenir que dos procesos entren a su sección crítica al mismo tiempo es la exclusión mutua. Se utiliza para sincronizar el acceso a la región crítica el método de Semáforos de tipo Mutex que permiten o no el ingreso a la región crítica.

Ejercicio 3- Performance de tecnologías de almacenamiento

Una vez creadas las unidades de procedimiento y ejecutadas de a una por vez, obtuvimos los siguientes resultados:

CASO	TIEMPO RELOJ	TIEMPO USUARIO	TIEMPO SISTEMA	BLOQUES ENTRADA	BLOQUES SALIDA
RAID 0 – NAS	2475631593	204000000	1401032704	0	16000
RAID 0 – SAN	4161608747 ns	20000000 ns	852000000 ns	0	8000
RAID 1 – NAS	2928854943	132000000	1653032704	0	16000
RAID 1 – SAN	1558754700 ns	52000000 ns	796000000 ns	0	8000
RAID 5 – NAS	2705041933	196000000	1729032704	0	16000
RAID 5 – SAN	3647184786 ns	12000000 ns	792000000 ns	0	8000
RAID 6 – NAS	149935192	328000000	213934592	0	16000
RAID 6 – SAN	779464039 ns	20000000 ns	832000000 ns	0	8000
RAID 10 – NAS	3315212085	68000000	1525032704	0	16000
RAID 10 – SAN	22109376 ns	16000000 ns	788000000 ns	0	8000

Conclusiones

Analice el comportamiento y los datos obtenidos en cada uno de los casos, compare los resultados entre los mismos y explique a qué se deben las diferencias y similitudes en los comportamientos de cada uno de los RAID según la forma en los que fueron compartidos.

RAID0:

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.

RAID1:

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 de escritura no mejora. Sin embargo, dispone de un mayor rendimiento de lectura, puesto que pueden leerse ambos discos al mismo tiempo.

RAID5:

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). 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. Las escrituras en un RAID 5 son costosas en términos de operaciones de disco y tráfico entre los discos y la controladora. La velocidad de transferencia tampoco es buena.

RAID6:

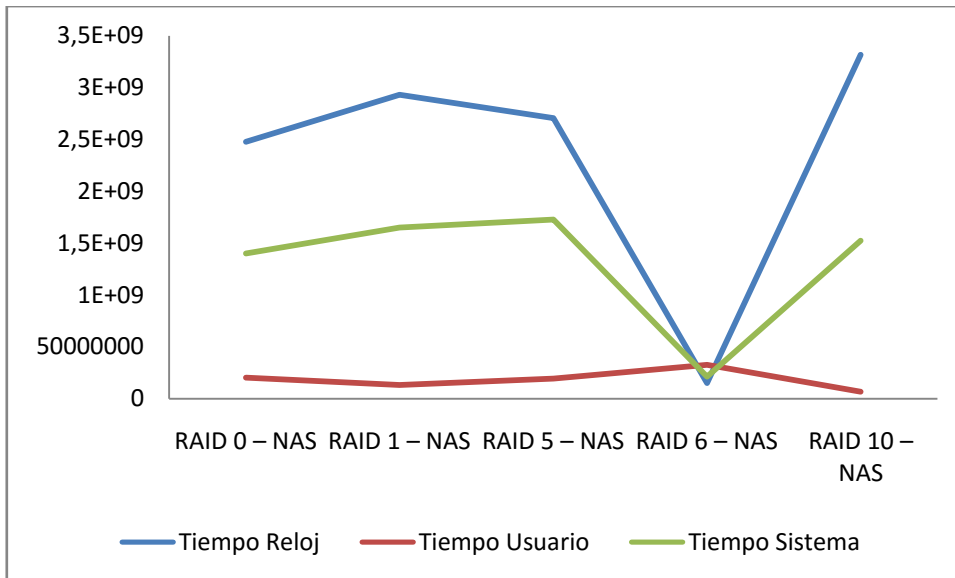
Un **RAID 6** amplía el nivel RAID 5 añadiendo otro bloque de paridad. Un RAID 6 no penaliza el rendimiento de las operaciones de lectura, pero sí el de las de escritura debido al proceso que exigen los cálculos adicionales de paridad.

RAID10:

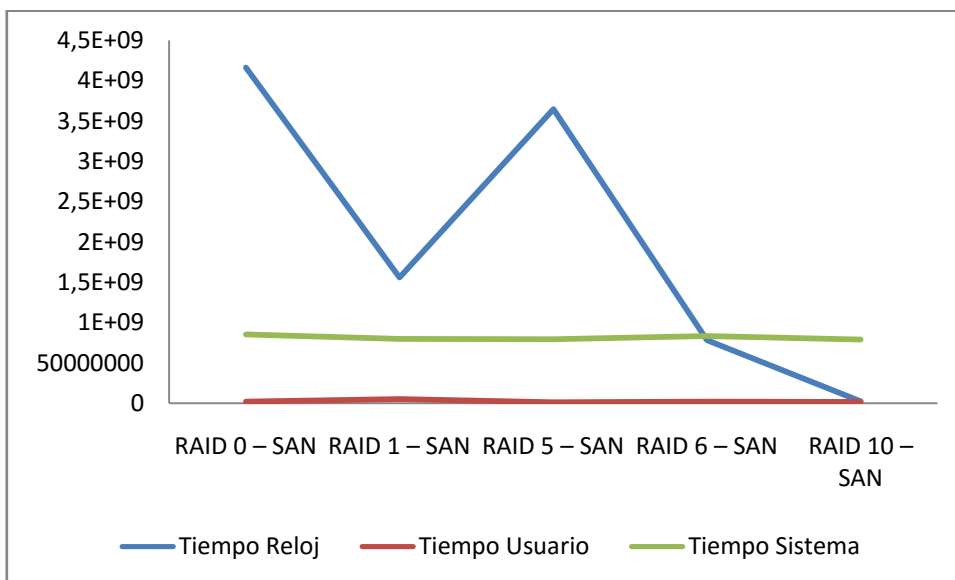
La ventaja de utilizar una **RAID 10** es disponer de la redundancia de la RAID 1 y del nivel de rendimiento de la RAID 0. Los datos no necesitan realizar procesos de regeneración de la información de la paridad porque ésta se copia de la otra unidad replicada. Las velocidades de escritura y lectura se multiplican por 3 en un array de 6 discos.

Diferencias de métricas en tecnologías de almacenamiento San y Nas

La principal diferencia entre ambas estructuras es la forma en que se mapean, se realizan las conexiones y como se resuelven las peticiones. Luego vienen los distintos niveles de raid con sus distintas características y configuraciones y valores de performance. Lo que hace que a la hora de implementar un tipo de almacenamiento raid se decida por alguno de ellos en particular, son las necesidades de los usuarios (tiempos lectura, tiempos de escritura, recuperación de datos, tiempos de realización del sistema o de los programas, etc).



En el gráfico se observa que **RAID 6** mejora considerablemente el tiempo de acceso a datos y provee un mayor rendimiento de E/S. Además proporciona un alto rendimiento en aplicaciones de velocidad de demanda interactiva pero con un costo considerable en lecturas y escrituras (tiempo de usuario).



En cambio en una red SAN, RAID 10 mejora el rendimiento y la rapidez de los tiempos ya que se trata de una norma orientada a bloques de datos (el equipo puede obtener datos de más de un disco a la vez ya que los datos se graban en forma de bandas en dos o más discos).

Anexo:

Tiempo de reloj: tiempo aproximado de CPU que transcurrió desde que nuestro programa fue iniciado hasta que finalizó. Dicho tiempo representado en un valor de tipo `clock_t`: un valor entero que indica una cantidad de "tics" de reloj.

Tiempo de usuario: tiempo transcurrido desde el principio hasta el final de un periodo durante la ejecución de una aplicación. Por ejemplo cuando un programa recorre una matriz, se acumula tiempo de CPU de usuario.

Tiempo de sistema: es el tiempo total de un trabajo en el sistema incluyendo el tiempo de servicio (tiempo que tarda el sistema en completar una tarea). Por ejemplo el tiempo total es la combinación de la cantidad de tiempo que la CPU tarda en realizar alguna acción para un programa y la cantidad de tiempo que la CPU tarda en realizar llamadas al sistema para el kernel.

¿A qué se debe que se haya solicitado que los discos virtuales se definan como FixedSize?


Al crear un disco de este tipo, se requiere disponer de espacio suficiente que aparecerá en la máquina virtual invitada. Es decir, si creamos un Disco Virtual de tamaño fijo de 200MB en el sistema de ficheros del servidor, en realidad ocupa 200 MB de espacio de almacenamiento físico en el disco duro del equipo, incluso cuando el disco virtual no tiene datos almacenados en él. La ventaja de usar este tipo de discos es una mejora en el rendimiento de la máquina virtual y además evita el problema de la Fragmentación, pero tienen el inconveniente que ocupan todo el espacio aunque interiormente tengan poca o ninguna información.

Además, cuando los datos deben ser escritos en un disco de expansión fija, el sistema operativo ya tiene asignado un espacio libre en la unidad física que contiene el archivo de disco duro virtual y luego actualiza la tabla de consulta para que los datos pueden ser escritos. Esto provee una mejora de los tiempos de administración de datos ya que proporciona una previsibilidad y que no se quedará sin espacio de almacenamiento de forma inesperada en el host.


Demuestre a través de imágenes (capturas de pantallas) del producto SAN & NAS qué RAIDs pudieron soportar tolerancia a fallos (y de cuantos discos) sin generar la pérdida de la lun completa. Todas las imágenes deben ser incluidas en un documento tipo Word con una breve explicación sobre lo que se está demostrando.

RAID0: al eliminar uno de los discos, la carpeta compartida se elimina por completo debido a que se elimina el grupo completo a causa de la configuración del mismo en el servidor openfiler (no redundante). Se emplea toda la capacidad de los discos. No tiene integridad de datos lo que implica que un error en uno de los discos produzca la pérdida total de los datos.

Software RAID Management

 No existing RAID arrays were found.

Create a new RAID array


 No existing RAID partitions were found, or all existing RAID partitions are used. You can [create new RAID partitions](#).

RAID1:

En cambio vemos que al eliminar un disco de RAID1 la carpeta sigue compartida en pero levemente degradada:

Software RAID Management

Array	Level	Array Size	Device Size	State	Synchronization	Manage	Add	Used In	Delete
/dev/md1	RAID-1	202.88 MB	202.88 MB	Clean & degraded	Not started	View members	All RAID partitions are used	raid1 VG	In use



Member devices of array "/dev/md1" : RAID-1

Number	Member	Device	Faulty	Active	Synchronized	Spare	Remove
0	0	/dev/sda1	NO	YES	YES	NO	Member
1	1	/dev/sdb1	NO	YES	YES	NO	Member
2	2	/dev/sdc1	NO	YES	YES	NO	Member
3	3	/dev/sdd1	NO	YES	YES	NO	Member
4	4	/dev/sde1	NO	YES	YES	NO	Member
5	5		NO	NO	NO	NO	Removed

[Home](#) · [Documentation](#) · [Support](#) · [Website](#) · [License](#) · [Log Out](#)

```
root@localhost:/media/RAIDONAS# ls
archivo.txt servidor servidor.c
```

Repetimos para 2 discos:

Software RAID Management									
Array	Level	Array Size	Device Size	State	Synchronization	Manage	Add	Used In	Delete
/dev/md1	RAID-1	202.88 MB	202.88 MB	Clean & degraded	Not started	View members	All RAID partitions are used	raid1 VG	In use

Member devices of array "/dev/md1" : RAID-1							
Number	Member	Device	Faulty	Active	Synchronized	Spare	Remove
0	0	/dev/sda1	NO	YES	YES	NO	Member
1	1	/dev/sdb1	NO	YES	YES	NO	Member
2	2	/dev/sdc1	NO	YES	YES	NO	Member
3	3	/dev/sdd1	NO	YES	YES	NO	Member
4	4		NO	NO	NO	NO	Removed
5	5		NO	NO	NO	NO	Removed

Repetimos para 3 discos:

Software RAID Management									
Array	Level	Array Size	Device Size	State	Synchronization	Manage	Add	Used In	Delete
/dev/md1	RAID-1	202.88 MB	202.88 MB	Clean & degraded	Not started	View members	All RAID partitions are used	raid1 VG	In use

Member devices of array "/dev/md1" : RAID-1							
Number	Member	Device	Faulty	Active	Synchronized	Spare	Remove
0	0	/dev/sda1	NO	YES	YES	NO	Member
1	1	/dev/sdb1	NO	YES	YES	NO	Member
2	2	/dev/sdc1	NO	YES	YES	NO	Member
3	3		NO	NO	NO	NO	Removed
4	4		NO	NO	NO	NO	Removed
5	5		NO	NO	NO	NO	Removed

Repetimos para 4 discos:

Software RAID Management									
Array	Level	Array Size	Device Size	State	Synchronization	Manage	Add	Used In	Delete
/dev/md1	RAID-1	202.88 MB	202.88 MB	Clean & degraded	Not started	View members	All RAID partitions are used	raid1 VG	In use


Member devices of array "/dev/md1" : RAID-1							
Number	Member	Device	Faulty	Active	Synchronized	Spare	Remove
0	0	/dev/sda1	NO	YES	YES	NO	Member
1	1	/dev/sdb1	NO	YES	YES	NO	Member
2	2		NO	NO	NO	NO	Removed
3	3		NO	NO	NO	NO	Removed
4	4		NO	NO	NO	NO	Removed
5	5		NO	NO	NO	NO	Removed


Repetimos para 5 discos:

Software RAID Management									
Array	Level	Array Size	Device Size	State	Synchronization	Manage	Add	Used In	Delete
/dev/md1	RAID-1	202.88 MB	202.88 MB	Clean & degraded	Not started	View members	All RAID partitions are used	raid1 VG	In use

Member devices of array "/dev/md1" : RAID-1							
Number	Member	Device	Faulty	Active	Synchronized	Spare	Remove
0	0	/dev/sda1	NO	YES	YES	NO	Member
1	1		NO	NO	NO	NO	Removed
2	2		NO	NO	NO	NO	Removed
3	3		NO	NO	NO	NO	Removed
4	4		NO	NO	NO	NO	Removed
5	5		NO	NO	NO	NO	Removed

Y observamos que al eliminar el sexto disco perdemos toda la carpeta compartida y por lo tanto se elimina todo el grupo de discos (es debido a la configuración en forma de espejo).

Software RAID Management	
	No existing RAID arrays were found.

Create a new RAID array	
	No existing RAID partitions were found, or all existing RAID partitions are used. You can create new RAID partitions .

RAID5:

Repetimos los mismos pasos que RAID1:

Vemos que al eliminar un disco de RAID5 la carpeta sigue compartida en pero levemente degradada:

Software RAID Management									
Array	Level	Array Size	Device Size	State	Synchronization	Manage	Add	Used In	Delete
/dev/md2	RAID-5	1014.38 MB	202.88 MB	Clean & degraded	Not started	View members	All RAID partitions are used	raid5 VG	In use

Member devices of array "/dev/md2" : RAID-5							
Number	Member	Device	Faulty	Active	Synchronized	Spare	Remove
0	0	/dev/sda1	NO	YES	YES	NO	Member
1	1	/dev/sdb1	NO	YES	YES	NO	Member
2	2	/dev/sdc1	NO	YES	YES	NO	Member
3	3	/dev/sdd1	NO	YES	YES	NO	Member
4	4	/dev/sde1	NO	YES	YES	NO	Member
5	5		NO	NO	NO	NO	Removed

```
root@localhost:/media/RAIDONAS# ls
archivo.txt  servidor  servidor.c
```

Repetimos para 2discos:

Software RAID Management	
	No existing RAID arrays were found.

Y observamos que al eliminar el segundo disco perdemos toda la carpeta compartida y por lo tanto se elimina todo el grupo de discos (ya que solo soporta la perdida de hasta un disco).

RAID6:

Repetimos los mismos pasos que RAID1:

Vemos que al eliminar un disco de RAID6 la carpeta sigue compartida en pero levemente degradada:

Software RAID Management

Array	Level	Array Size	Device Size	State	Synchronization	Manage	Add	Used In	Delete
/dev/md3	RAID-6	811.50 MB	202.88 MB	Clean & degraded	Not started	View members	All RAID partitions are used	raid6 VG	In use

Member devices of array "/dev/md3" : RAID-6

Number	Member	Device	Faulty	Active	Synchronized	Spare	Remove
0	0	/dev/sda1	NO	YES	YES	NO	Member
1	1	/dev/sdb1	NO	YES	YES	NO	Member
2	2	/dev/sdc1	NO	YES	YES	NO	Member
3	3	/dev/sdd1	NO	YES	YES	NO	Member
4	4	/dev/sde1	NO	YES	YES	NO	Member
5	5		NO	NO	NO	NO	Removed

[Home](#) · [Documentation](#) · [Support](#) · [Website](#) · [License](#) · [Log Out](#)

```
root@localhost:/media/RAIDONAS# ls
archivo.txt  servidor  servidor.c
```

Repetimos para 2 discos:

Software RAID Management

Array	Level	Array Size	Device Size	State	Synchronization	Manage	Add	Used In	Delete
/dev/md3	RAID-6	811.50 MB	202.88 MB	Clean & degraded	Not started	View members	All RAID partitions are used	raid6 VG	In use


Member devices of array "/dev/md3" : RAID-6

Number	Member	Device	Faulty	Active	Synchronized	Spare	Remove
0	0	/dev/sda1	NO	YES	YES	NO	Member
1	1	/dev/sdb1	NO	YES	YES	NO	Member
2	2	/dev/sdc1	NO	YES	YES	NO	Member
3	3	/dev/sdd1	NO	YES	YES	NO	Member
4	4		NO	NO	NO	NO	Removed
5	5		NO	NO	NO	NO	Removed

[Home](#) · [Documentation](#) · [Support](#) · [Website](#) · [License](#) · [Log Out](#)

Repetimos para 3 discos:

Software RAID Management



No existing RAID arrays were found.

Y observamos que al eliminar el tercer disco perdemos toda la carpeta compartida y por lo tanto se elimina todo el grupo de discos (debido a su redundancia dual de doble paridad soporta la pérdida de un disco más que raid 5).

RAID10:

Repetimos los mismos pasos que RAID1:

Vemos que al eliminar un disco de RAID10 la carpeta sigue compartida en pero levemente degradada:

Software RAID Management

Array	Level	Array Size	Device Size	State	Synchronization	Manage	Add	Used In	Delete
/dev/md4	RAID-10	608.62 MB	202.86 MB	Clean & degraded	Not started	View members	All RAID partitions are used	raid10 VG	In use

Member devices of array "/dev/md4" : RAID-10

Number	Member	Device	Faulty	Active	Synchronized	Spare	Remove
0	0	/dev/sda1	NO	YES	YES	NO	Member
1	1	/dev/sdb1	NO	YES	YES	NO	Member
2	2	/dev/sdc1	NO	YES	YES	NO	Member
3	3	/dev/sdd1	NO	YES	YES	NO	Member
4	4	/dev/sde1	NO	YES	YES	NO	Member
5	5		NO	NO	NO	NO	Removed

[Home](#)
[Documentation](#)
[Support](#)
[Website](#)
[License](#)
[Feedback](#)

```
root@localhost:/media/RAIDONAS# ls
archivo.txt  servidor  servidor.c
```

Repetimos para 2discos:

Software RAID Management



No existing RAID arrays were found.

Y observamos que al eliminar el segundo disco perdemos toda la carpeta compartida y por lo tanto se elimina todo el grupo de discos (debido a las características heredadas de Raid 0).

Indique qué conceptos teóricos están involucrados en cada uno de los casos anteriores.

RAID 0 (Data Striping)

Distribuye los datos equitativamente entre dos o más discos (usualmente se ocupa el mismo espacio en dos o más discos) sin información de paridad que proporcione redundancia. El RAID 0 se usa normalmente para proporcionar un alto rendimiento de lectura ya que los datos se recuperan de dos o más discos de forma paralela, aunque un mismo fichero solo está presente una vez en el conjunto. También es posible crear un RAID 0 con más de dos discos pero reduce el tiempo medio entre fallo por lo que el sistema fallido no se repara y en consecuencia se produce la pérdida total de los datos.

RAID 1 (Mirroring)

Este es el nivel RAID más básico en cuanto a “Data Mirroring” se refiere. Recordemos que el “Mirroring” consiste en crear una copia exacta (o espejo) de un conjunto de datos en dos o más discos. En el RAID 1, los datos del disco primario son duplicados en otro disco. Mediante esta operación no se mejora el rendimiento del sistema pero si un disco falla tendrá una copia de seguridad en el segundo disco. Puede perder hasta la mitad de los discos (espejado).

RAID5 (PARIDAD)

Un RAID con paridad (por ejemplo un RAID-5), usa división de datos a nivel de bloques, distribuyendo la información de paridad entre todos los discos miembros del conjunto.

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).

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 utilizan 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.

El sistema sabe que un disco ha fallado, pero sólo con el fin de 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. El Modo Degradado se “activa” cuando uno de los discos se

corrompe o sale de línea por fallo físico, y no se comunica correctamente con el resto de los discos en el arreglo.

Una vez que el arreglo pasa a Modo Degradado, se pone una carga muy grande en el resto de los discos. La mayoría de las veces, una vez que un disco falla, el resto tienen que compensar esa falta y trabajan más duro, reduciendo su rendimiento y acortando su vida útil. RAID 5 necesitará un mínimo de 3 discos para ser implementado. El fallo de otro disco provoca la pérdida completa de los datos.

RAID6 (EXTRA PARIDAD)

El sistema RAID6 es equivalente a RAID5 pero se utilizan dos discos para CRC. El resultado es que pueden fallar dos discos simultáneamente y aun así el RAID sigue funcionando. Requiere al menos cuatro discos, y la capacidad ofrecida será la de $n-2$ discos. Suele ser la aconsejable cuando el número de discos es elevado, dado que aporta una seguridad extra a cambio de un coste de almacenamiento cada vez más marginal. RAID 6 puede perder 2 discos, un fallo de otro disco provoca la pérdida completa de los datos.

RAID 10

Está configurado de forma que el RAID 0 está dividido en dos RAID 1. Un RAID 10 requiere un mínimo de cuatro discos, y secciona los datos en pares duplicados. Si dos discos en el mismo par duplicado falla, todos los datos se perderán porque no hay paridad.

La gran ventaja del RAID 10 es que todos los discos menos uno en cada RAID 1 podrían fallar sin que se produjera ninguna pérdida de datos.