



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

**Ingeniería Técnica de Telecomunicación
(Especialidad: Sistemas de Telecomunicación)**

**ESTUDIO, DISEÑO E IMPLEMENTACIÓN
DE UN
SERVIDOR DE ALMACENAMIENTO REMOTO
MULTIPROTOCOLO SOBRE PLATAFORMA VIRTUAL**

PROYECTO FIN DE CARRERA

Autor: Jorge Valenzuela Jiménez

Tutor: Jesús Carretero Pérez

Julio 2010

Contenido

1.	Introducción.....	11
1.1	Introducción al almacenamiento de datos.....	11
1.2	Motivaciones y objetivos del proyecto	15
1.3	Contenido del documento	17
2.	Estado del arte	19
2.1	Estudio de soluciones actuales.....	19
2.1.1	Storage Area Network	19
2.1.2	Network Attached Storage.....	29
2.1.3	Ip storage	31
2.2	Ejemplos de sistemas actuales	36
3.	Definición del entorno	38
3.1	Estudio del software de virtualización a utilizar.....	38
3.1.1	Evolución en la tecnología de virtualización.....	38
3.1.2	Tipos de virtualización	39
3.1.3	Características generales de VirtualBox.....	41
3.1.4	VirtualBox frente al resto: Ventajas e inconvenientes	43
3.1.5	Conclusiones.....	44
3.2	Estudio del Sistema Operativo a utilizar.....	44
3.2.1	Estado actual del mundo “Linux”.....	44
3.2.2	OpenSolaris: Estado Actual.....	45
3.2.3	OpenSolaris: Ventajas e inconvenientes.....	47
3.2.4	Requisitos de instalación	47
3.2.5	Conclusiones.....	48
3.3	Estudio del software para una solución iscsi	48
3.3.1	Introducción a los target iSCSI.....	48
3.3.2	Proyectos Open Source para implementar target iSCSI.....	49
3.4	Instalación del software de Base.....	51
3.4.1	Instalación de VirtualBox.....	51
3.4.2	Instalación de OpenSolaris	62
4.	Diseño arquitectónico del servidor de almacenamiento.....	80

4.1	Componentes necesarios para implementar la solución	80
4.2	Creación y configuración de Logical Unit.....	81
4.2.1	Tipos de Logical Unit.....	82
4.2.2	Ventajas de utilizar ZFS frente al resto de opciones	82
4.2.3	Creación de una LU utilizando ZFS	83
4.3	Configuración servidor NFS	90
4.4	Configuración servidor SMB.....	91
5.	Estudio y Configuración de la parte cliente para utilizar iSCSI.....	94
5.1	Instalación sobre SO Windows.....	94
5.1.1	Estado actual de la parte iSCSI para Windows	94
5.1.2	Instalación.....	94
5.1.3	Pruebas	96
5.1.4	Conclusiones.....	98
5.2	Instalación sobre SO Linux.....	99
5.2.1	Estado actual de los proyectos para implementar initators en Linux	99
5.2.2	Instalación.....	99
5.2.3	Pruebas	101
5.2.4	Conclusiones.....	103
5.3	Instalación sobre SO OpenSolaris	103
5.3.1	Instalación.....	103
5.3.2	Pruebas	103
5.3.3	Conclusiones.....	104
6.	Performance de los sistemas.....	105
6.1	Performance de acceso desde cliente Windows	105
6.2	Performance de acceso desde cliente Linux	108
6.3	Performance de acceso desde cliente OpenSolaris	109
6.4	Conclusiones	111
7.	Conclusiones y presupuesto	113
7.1	Conclusiones del proyecto	113
7.2	Mejoras futuras	114
7.2.1	Procedimiento de estimación de recursos.....	114

7.2.2	Descripción del equipo de trabajo	115
7.2.3	Estudio del presupuesto	115
8.	Bibliografía.....	117

Tabla de ilustraciones

1. Evolución de la capacidad y coste en almacenamiento	12
2. Diagrama DAS.	13
3. Diagrama SAN.	14
4. Diagrama NAS.	15
5. Ejemplo estructura SAN.	19
6. Diagrama P-P.	20
7. Diagrama A-L.	21
8. Diagrama Fabric.	21
9. Diferencias entre topologías SAN.	22
10. FC Layers.	23
11. Layer 0.	23
12. Campos de un Frame.	25
13. Diagrama de Control de Flujo.	27
14. Control de Flujo Clase 1.	28
15. Control de flujo clase 2.	28
16. Conexión iFCP.	32
17. Bloques iSCSI.	33
18. Diagrama RDMA.	35
19. Diagrama de bloques FCoE.	35
20. Full Virtualizer.	39
21. Paravirtualización.	39
22. Virtualización en SO.	40
23. Cuota de mercado en SO.	45
24. Cronología OpenSolaris.	46
25. Crossbow Project.	47
26. iSCSi Cliente/Servidor.	48
27. Diagrama bloques.	50
28. OpenSolaris Target.	51
29. Archivo Ejecutable.	52
30. Advertencia Seguridad.	52

31. Bienvenida.....	53
32. Licencia.....	53
33. Instalador. Componentes.....	54
34. Instalador. Opciones escritorio.....	54
35. Advertencia interfaces red.....	55
36. Progreso.....	55
37. Instalación finalizada.....	56
38. VB Registro.....	57
39. VB Inicio.....	57
40. VB Nueva.....	58
41. VB Identificación Máquina.....	58
42. VB Ram.....	59
43. VB Disco Duro.....	59
44. VB Tipo Almacenamiento.....	60
45. VB Tamaño Disco.....	61
46. VB Presentacion VM.....	61
47. VB Almacenamiento.....	62
48. VB Añadir DVD.....	63
49. VB Añadir ISO.....	64
50. VB ISO Solaris.....	65
51. Solaris Instalación 1.....	65
52. Solaris Instalación 2.....	66
53. Solaris Instalación 3.....	66
54. Solaris Instalación 4.....	67
55. Solaris Instalación 5.....	67
56. Solaris Instalación 6.....	68
57. Solaris Instalación 7.....	68
58. Solaris Instalación 8.....	69
59. Solaris Instalación 9.....	70
60. Solaris Instalación 10.....	70
61. Solaris Instalación 11.....	71

62. Solaris Instalación 12.	71
63. Solaris Instalación 13.	72
64. Solaris Instalación 14.	72
65. Solaris Instalación 15.	73
66. Solaris Instalación 16.	73
67. Solaris Instalación 17.	74
68. Solaris Instalación 18.	74
69. Solaris Instalación 19.	75
70. Solaris Instalación 20.	75
71. Solaris Instalación 21.	76
72. Solaris Instalación 22.	76
73. Solaris Instalación 23.	77
74. Solaris Instalación 24.	77
75. Solaris Instalación 25.	78
76. Solaris Instalación 26.	78
77. Solaris Instalación 27.	79
78. Solaris Instalación Progreso.	79
79. Servicios STMF.	82
80. Asignar discos LU.	84
81. VB discos LU.	85
82. Solaris Format.	85
83. VB Añadir Controladora SCSI.	86
84. Solaris Añadir nuevos discos.	86
85. Nuevo Zpool.	87
86. Crear Vista LU.	87
87. Mapeado de LU.	89
88. Activar Servicio iSCSI tgt.	89
89. Configurar NFS.	91
90. Crear FS Samba.	92
91. Configurar Samba.	92
92. Microsoft Initiator 1.	95

93. Microsoft Initiator 2.	95
94. Configurar Win Initiator 1.	96
95. Configurar Win Initiator 2.	96
96. Configurar Win Initiator 3.	97
97. Configurar Win Initiator 4.	97
98. Configurar Win Initiator 5.	98
99. Initiator Linux 1.	99
100. Initiator Linux 2.	100
101. NFS Linux 1.	100
102. Initiator Linux 3.	101
103. Initiator Linux 4.	101
104. Initiator Linux 6.	102
105. NFS Linux 2.	102
106. Opensolaris Initiator 1.	103
107. Opensolaris 2.	104
108. Performance tamaño bloque automático.	105
109. Performance tamaño bloque 512KB.	106
110. Performance tamaño bloque 1 KB.	106
111. Performance NFS.	107
112. Performance Samba.	107
113. Performance iSCSI.	108
114. Performance NFS.	108
115. Performance iSCSI 1.	109
116. Performance iSCSI 2.	109
117. Performance iSCSI 3.	109
118. Performance iSCSI 4.	110
119. Performance NFS 1.	110
120. Performance NFS 2.	110
121. Performance NFS 3.	111
122. Performance NFS 4.	111
123. Desglose Equipo de trabajo.	115

1. Introducción

1.1 Introducción al almacenamiento de datos

La tecnología avanza a un ritmo vertiginoso y con ella las necesidades de almacenamiento de datos. Vivimos en la era de la digitalización y ello conlleva que toda la información deba almacenarse en soportes informáticos.

Si miramos la evolución desde el lado de los usuarios y comparásemos hoy las necesidades actuales de un usuario medio, con las de hace 3 o 4 años, veríamos que la necesidad de almacenamiento ha crecido fuertemente, impulsada principalmente por la digitalización de la información. Cada vez necesitamos dispositivos con más capacidad de almacenar información. Es común observar como en poco tiempo las cámaras fotográficas son capaces de guardar instantáneas con mayor calidad, o las cámaras de video capaces de grabar en alta definición, lo que implica ficheros de mayor tamaño y por tanto mayor espacio de almacenamiento.

Si pasamos al mundo empresarial, se puede hablar de dos épocas. Antes de aparecer Internet y después.

Antes de llegar Internet a los usuarios, había empresas con necesidades de almacenamiento muy elevadas (sobre todo empresas grandes). Los presupuestos que se asignaban a los departamentos de IT eran elevados y por tanto existían soluciones de alta disponibilidad a las que sólo podían acceder ciertas corporaciones, pero que proporcionaban las necesidades que estas demandaban (Alta disponibilidad, alto rendimiento, tolerancia a fallos y escalabilidad).

Con la llegada de Internet, llegaron nuevas aplicaciones y miles de nuevas empresas. Muchas de ellas eran pequeñas empresas, pero con necesidades elevadas de espacio para almacenar grandes cantidades de información.

Estas necesidades surgen por la propia naturaleza de las aplicaciones aparecidas, como pueden ser:

- E-commerce
- Banca Online
- Discos duros virtuales para usuarios
- E-mail
- Multimedia (audio, video)
- Blogs
- Redes sociales
- Foros

Desde aquel momento, la gran mayoría de empresas comenzó a tener las mismas necesidades que las grandes empresas, pero las soluciones que existían en el mercado hasta entonces eran costosas.

Las empresas grandes con abultados presupuestos para su departamento de TI podían permitirse comprar sistemas redundados, con copia remota entre datacenters para disponer de planes de disaster recovery, y que les permitiera tener siempre accesibles sus aplicaciones vitales.

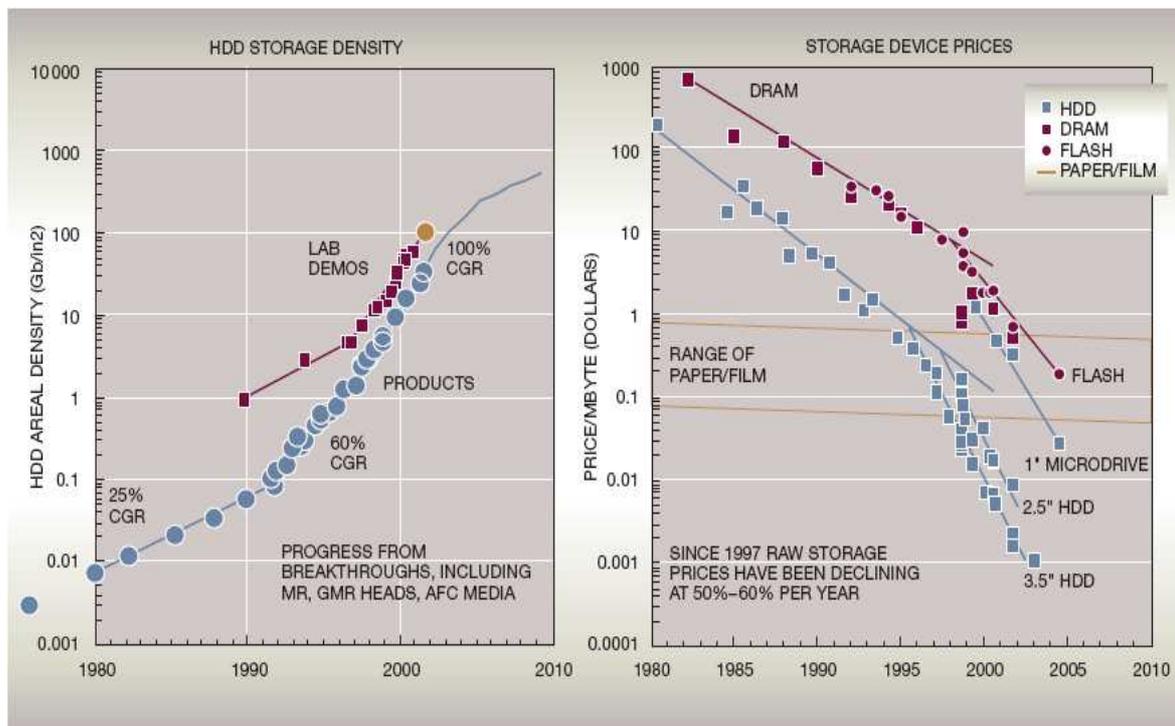
Sin embargo empresas no tan grandes o con presupuestos ajustados para sus departamentos de TI no podían costear soluciones de este tipo, aunque el acceso a la información también fuese vital para el negocio.

Entonces comenzaron a aparecer nuevos fabricantes y nuevas soluciones (sistemas y topologías de red) que han permitido que actualmente existan en el mercado una gran cantidad de opciones disponibles para diseñar datacenters cubriendo las demandas actuales del mercado y pudiendo obtener costes más contenidos.

Los sistemas actuales aún habiendo sufrido una gran evolución siguen basándose en el disco duro como unidad mínima (al menos en su mayoría). Sobre estas unidades mínimas se añaden capas hardware y software que ayudan a dotar a los sistemas de mayor rendimiento (utilizando nuevas formas de acceso en escritura y en lectura, disminuyendo los tiempos de acceso a disco) y además facilitan las labores de administración diarias, evitando que puedan ocurrir errores humanos.

En términos económicos, con la evolución de la tecnología el coste de los HDD (Hard Disk drive) ha ido disminuyendo, mientras que su densidad ha ido aumentando, lo que ha implicado que en los sistemas actuales el menor coste corresponde curiosamente a las propias unidades de almacenamiento.

En la siguiente gráfica podemos ver cómo ha sido esta evolución en el tiempo, tanto en capacidad, como en coste:



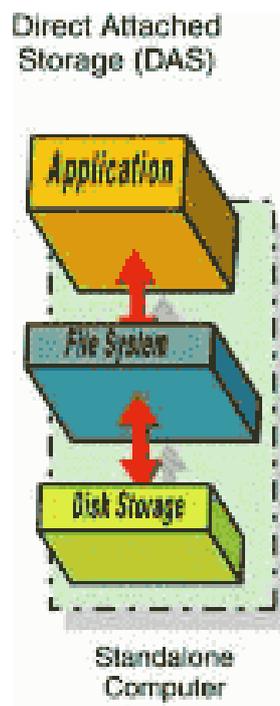
1. Evolución de la capacidad y coste en almacenamiento .

Respecto a las topologías de conexión la evolución ha sucedido de manera más lenta debido principalmente al coste.

El coste de incorporar elementos nuevos no es muy alto, es posible comprar un elemento que incluya alguna novedad tecnológica e incorporarlo en nuestros sistemas en producción (normalmente incorporan retrocompatibilidad para poder ser utilizados con sistemas menos evolucionados). Sin embargo, la decisión de basar nuestros sistemas de almacenamiento en una topología u otra será la responsable de la evolución de nuestros sistemas, ya que normalmente los sistemas son distintos para cada topología. No es lo mismo un sistema de almacenamiento por fibra óptica que por red, y un cambio de topología normalmente implica un cambio en toda la infraestructura, por lo que el adoptar una topología u otra repercutirá en el futuro en toda la plataforma, por eso es importante conocer cómo se comporta cada una.

La forma tradicional de conectar un servidor a un almacenamiento, se conoce como DAS (Direct Attached Network) y es la más sencilla de todas, ya que el servidor se conecta directamente a un HDD o a un array (tipo scsi por ejemplo).

Todas las aplicaciones que corren sobre el servidor hacen sus peticiones de datos directamente al filesystem.



2. Diagrama DAS.

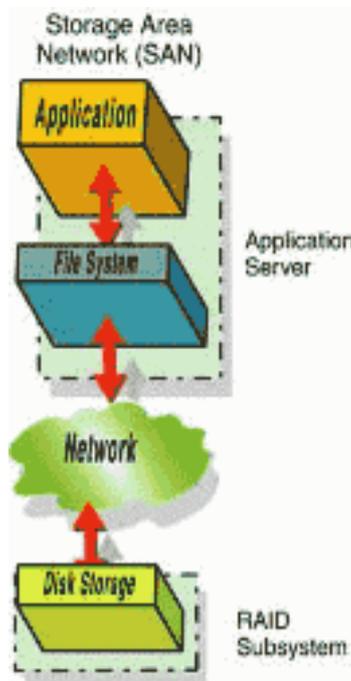
Los protocolos más utilizados en DAS son SCSI (Small Computer System Interface), SAS (Serial Attached SCSI) y FC (Fibre Channel).

La principal desventaja de utilizar DAS radica en la incapacidad para compartir datos o recursos no utilizados con otros servidores, lo que dota a esta tecnología de muy poca flexibilidad. Es por ello que aparecen las redes SAN (Storage Area Network), para poder ofrecer almacenamiento compartido a un grupo de servidores.

Las redes SAN son redes de almacenamiento **dedicadas** que proporciona acceso de nivel de bloque a LUNs .

Una LUN, o número de unidad lógica, es un disco “virtual” proporcionado por la SAN.

Una vez el disco es configurado por el servidor, la forma de acceso a los datos por las aplicaciones es transparente (como en la topología DAS).



3. Diagrama SAN.

Las redes SAN utilizan como medio físico de transmisión mayoritariamente fibra óptica, lo que le proporciona un gran ancho de banda.

La mayoría de las SAN usan el protocolo SCSI para la comunicación entre los servidores y los dispositivos de almacenamiento, aunque no se haga uso del interfaz físico de bajo nivel. En su lugar se emplea una capa de mapeo, como el estándar FCP.

Con este estándar es posible salvar las limitaciones que impone el protocolo scsi en cuanto al número máximo de dispositivos conectados, permitiendo conectar a cada puerto tantas LUN como sea necesario dependiendo de las necesidades de crecimiento.

Algunas ventajas que tienen las redes SAN frente a las redes DAS son:

- SAN provee duplicación de acceso a datos a través de múltiples caminos.
- SAN provee tecnología de redundancia en la escritura de datos

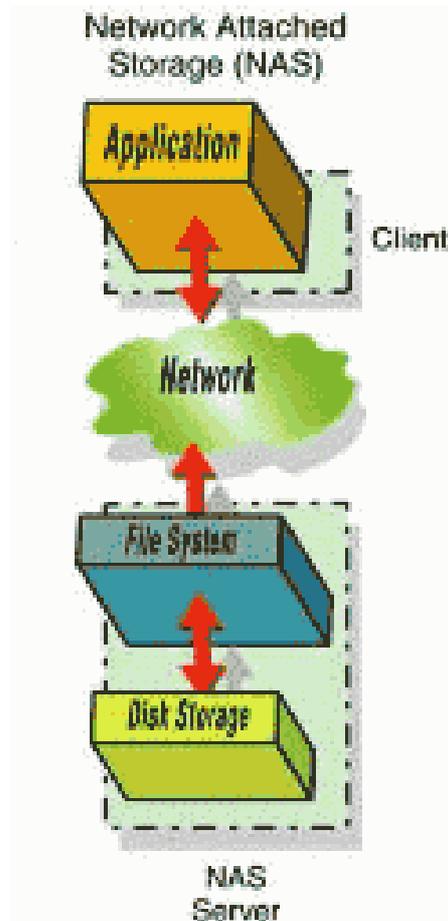
El principal inconveniente con el que cuentan las redes SAN es el coste elevado de los elementos que forman su infraestructura. Aún siendo la tecnología adoptada mayoritariamente por todas las empresas,

El último tipo de topología en aparecer son las redes NAS.

Las redes NAS comparten la capacidad de almacenamiento a través de una red TCP/IP utilizando normalmente los protocolos CIFS y NFS.

En la tecnología NAS, las aplicaciones y programas de usuario hacen las peticiones de datos a los sistemas de ficheros de manera remota mediante protocolos CIFS y NFS, y

el almacenamiento es local al sistema de ficheros. Sin embargo, DAS y SAN realizan las peticiones de datos directamente al sistema de ficheros.



4. Diagrama NAS.

NAS resulta muy útil para proporcionar el almacenamiento centralizado a servidores clientes en entornos con grandes cantidades de datos. Permite habilitar fácilmente sistemas de ficheros compartidos entre distintos clientes.

Las ventajas que ofrecen las topologías NAS sobre DAS y SAN son la capacidad de compartir las unidades, un menor coste, la utilización de la misma infraestructura de red y una gestión más sencilla.

Por el contrario, NAS tiene un menor rendimiento y fiabilidad por el uso compartido de las comunicaciones. Los protocolos de comunicaciones NAS son protocolos de compartición de ficheros como NFS, Samba o CIFS (Common Internet File System).

1.2 Motivaciones y objetivos del proyecto

Dentro de los proyectos enfocados a conseguir redes de almacenamiento sostenibles tanto en calidad, coste y eficiencia energética, aparecen los que intentan conseguir el acceso a dispositivos como en SAN a través de redes de bajo coste como es la red Ethernet.

Los principales proyectos abiertos actualmente son iSCSI (Internet SCSI), FCoE (Fibre Channel over Ethernet) y AoE (Ata over Ethernet).

Isesi es una extensión del protocolo scsi que utiliza TCP/IP para sus transferencias de datos. Al contrario que otros protocolos de red diseñados para almacenamiento, como por ejemplo el canal de fibra (que es la base de la mayor parte de las redes de áreas de almacenamiento), solamente requiere una simple interfaz ethernet (o cualquier otra red compatible TCP/IP) para funcionar.

Esto permite una solución de almacenamiento centralizada de bajo coste sin la necesidad de realizar inversiones costosas ni sufrir las habituales incompatibilidades asociadas a las soluciones canal de fibra para redes de área de almacenamiento.

El problema que parece tener iscsi se enfoca sobre todo en el rendimiento, ya que se ve afectado por la sobrecarga que generan las transmisiones TCP/IP, aunque cada vez este problema está más mitigado debido a las redes Gigabit Ethernet, lo que ha permitido una gran aceptación de la tecnología iSCSI en las empresas.

Sobre FcoE se puede decir que es un protocolo para encapsular paquetes Fibre Channel sobre redes Ethernet.

La mayoría de los datacenter , utilizan Ethernet para redes TCP / IP y Fibre Channel para redes de área de almacenamiento (SAN).

Con FCoE, Fibre Channel se convierte en otro protocolo de red que se transporta a través de Ethernet, además del tráfico IP.

El problema que tiene el protocolo FCoE es que su tráfico no viaja por encima de la capa IP de la pila de protocolos, lo que hace que su tráfico no sea enrutable por los switch y routers de la red. Son necesarios switches y routers específicos que soporten FCoE para enrutar el tráfico.

AoE es un protocolo diseñado para acceso de alto rendimiento a dispositivos SATA sobre redes Ethernet. La desventaja que tiene sobre iSCSI es la misma que FCoE. Su tráfico no puede ser enrutado por la electrónica de red de las redes TCP/IP.

La motivación principal de este proyecto es la de implementar un servidor de almacenamiento que utilice iSCSI para proporcionar espacio de almacenamiento centralizado a un conjunto de clientes con diversos Sistemas Operativos, de manera que tengamos una plataforma lo más variada posible, ya que iscsi es una tecnología que cada vez goza de más aceptación en los datacenter .

Además, con algunos desarrollos actuales, es posible diseñar un servidor iSCSi utilizando soluciones open-source, lo que nos puede ayudar a conseguir un sistema centralizado con un coste menor que utilizar una solución cerrada de algún fabricante.

En último lugar nos servirá para realizar una comparativa en términos de rendimiento frente a otras soluciones ofrecidas en el mercado, que nos permitan ver qué características interesantes pueden ofrecernos otras soluciones.

Para que el sistema de almacenamiento sea un servidor lo más completo posible se desea dotar al sistema de capacidades NAS (NFS y Samba).

Todo el diseño está basado en software open-source, lo que hace de nuestra solución una alternativa versátil y muy económica.

1.3 Contenido del documento

Este documento se compone de una serie de capítulos que pertenecen a las distintas fases de desarrollo del proyecto y que son los siguientes:

Capítulo I – Introducción

Se trata de una presentación que permitirá ir adentrándose en los conceptos sobre almacenamiento de datos, evidenciando las tendencias futuras de la tecnología y mostrando un pequeño resumen de los distintos tipos de redes. Además, se contemplan las motivaciones para realizar este proyecto y los objetivos a alcanzar.

Capítulo II - Estado del arte

En este capítulo se realiza un breve recorrido histórico mostrando las tecnologías utilizadas años atrás y la evolución hasta las tecnologías actuales y futuras. Además nos adentramos en los paradigmas utilizados en la actualidad para mostrar una visión técnica y tener una visión completa de los actuales sistemas de almacenamiento.

Capítulo III – Definición del entorno

Brevemente se estudiarán las técnicas existentes en virtualización de sistemas y se presentarán algunos ejemplos utilizados actualmente en el mundo empresarial. Se estudiarán los pros y los contras de cada una de las técnicas de virtualización de manera que se obtenga el conocimiento suficiente para poder optar por una solución dependiendo de las características necesarias.

En este mismo capítulo realizaremos un pequeño estudio sobre sistemas operativos y veremos las características de los más utilizados empresarialmente para tener una visión global de cómo se comporta cada sistema.

Una vez presentados los conceptos de virtualización y de sistemas operativos, en este capítulo se procederá a la instalación detallada de ambos.

Capítulo IV – Diseño del sistema de almacenamiento

En este capítulo se realizará una breve introducción la gestión de volúmenes y se implementará sobre el software elegido realizando todos los pasos para obtener un volumen gestionando el almacenamiento disponible. Además se documentará la instalación y configuración de Samba y Nfs.

Capítulo V - Cliente iSCSI

En este apartado del proyecto se realiza el estudio de las diferentes opciones existentes actualmente para implementar clientes iSCSI en la mayoría de las plataformas utilizadas en entornos reales. Después del estudio se procede a la instalación en cada una de las plataformas y a la configuración con respecto al servidor de almacenamiento.

Capítulo VI – Performance

Este capítulo pretende ser un pequeño estudio sobre las diferencias de rendimiento dependiendo de la plataforma utilizada como cliente y de la forma de acceso a los datos bien sea por iSCSI o bien sea a nivel de sistema de ficheros mediante Samba y/o Nfs.

Capítulo VII – Conclusiones y presupuesto

El último capítulo del proyecto pretende mostrar las ventajas y desventajas en cuanto a la elección de la tecnología de acceso a los datos. También pretende mostrar una pequeña muestra de posibles actualizaciones/mejoras que pueden realizarse a la implementación elegida.

En la última parte del capítulo se muestra el presupuesto detallado con la explicación en tiempo y coste de cada uno de los roles necesarios para llevar a cabo el proyecto.

2. Estado del arte

2.1 Estudio de soluciones actuales

En la actualidad existen numerosas tecnologías para cubrir las distintas necesidades que cada tipo de negocio demanda de los servicios de IT. Dentro de estas tecnologías cobran especial importancia las tecnologías que dan soporte al almacenamiento de datos.

En este proyecto nos vamos a centrar en el estudio de las 2 topologías de redes de almacenamiento que mayoritariamente están implantadas en el mundo empresarial: SAN y NAS. Después estudiaremos por extensión las SAN basadas sobre transporte IP.

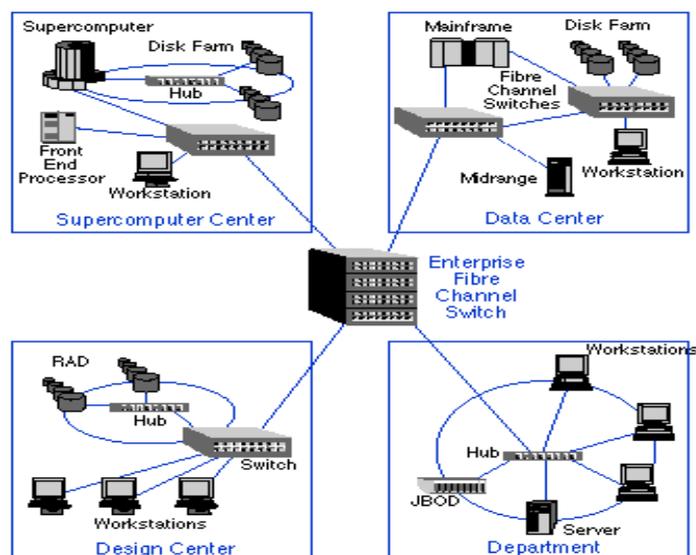
2.1.1 Storage Area Network

Las redes SAN fueron originalmente creadas para utilizar el estándar Fibre Channel.

Fibre Channel es una tecnología madura y segura para las comunicaciones de alta velocidad y además es la base para más del 90% de todas las instalaciones de SAN en el mundo.

Fibre Channel en la actualidad no sólo se aplica en el área de almacenamiento sino que funciona igual de bien en redes, vídeo y muchas otras aplicaciones, como por ejemplo en la industria aeroespacial. Además Fibre Channel es la tecnología mayoritariamente aceptada para soportar las siguientes aplicaciones:

- Grandes Bases de Datos y “warehouse”
- Almacenamiento de alto rendimiento
- Sistemas de backup y recuperación
- Clusters de servidores
- Backbone entre campus
- Redes de audio/video digital



5. Ejemplo estructura SAN.

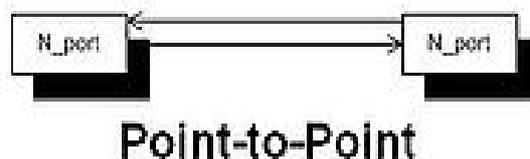
En la terminología de FC es muy común hablar de puertos. Un puerto es cualquier entidad que pueda comunicarse activamente sobre la red SAN y no tiene que ser obligatoriamente un hardware port. Además cada puerto se define unívocamente por su WWN (World Wide Name) similar a la dirección MAC en Ethernet.

Hay definidos varios tipos de puertos dependiendo de su función:

- **N_port:** es un puerto directo sobre un nodo (i.e servidor o un disco) utilizados en topología FC-P2P o FC-SW. También se le conoce como Node Port.
- **NL_port** Es un puerto sobre un nodo utilizando topología FC-AL. También es conocido como Node Loop Port
- **F_port** Es un puerto de un switch (fabric) el cual conecta con con un nodo con conexión punto a punto (i.e, con un N_Port). También se le conoce como Fabric Port. Un F_port no soporta topología en loop. FL_port es un puerto de switch que se conecta a una topología loop (i.e. NL_ports). También se le conoce como Fabric Loop Port.
- **E_port** es la conexión entre 2 switches FC. También se le conoce como Puerto de Expansión. Cuando los E_ports entre 2 switches forman un link, este link es conocido como ISL (Inter-Switch Link)
- **EX_port** es la conexión entre un route FC y un Switch FC. En el lado del switch este se ve como un E_port, pero en el lado del router es un EX_Port.
- **TE_port** es un añadido de Cisco a Fibre Channel y ahora ya pertenece al estándar. Es una extensión de ISL. El Te_Port provee no sólo las funciones estándar de un E_port, sino que permite routing de multiples VSAN (Virtual SAN). También se le conoce como Trunking E_Port (entendiendo trunking como un conjunto de ISL).

Existen 3 topologías principales de conexión en Fibre Channel dependiendo de cómo se conecten los puertos entre si:

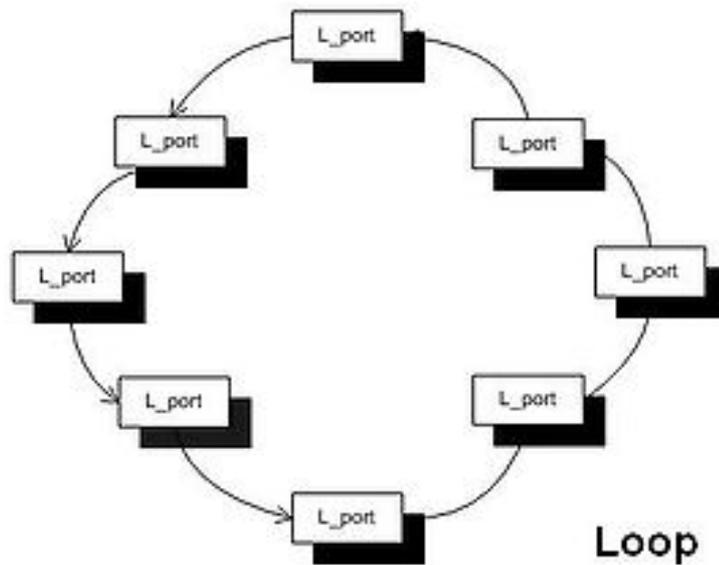
- **Point-to-Point (FC-P2P):** Dos dispositivos son conectados entre si directamente. Esta topología es la más simple y tiene una conectividad muy limitada.



6. Diagrama P-P.

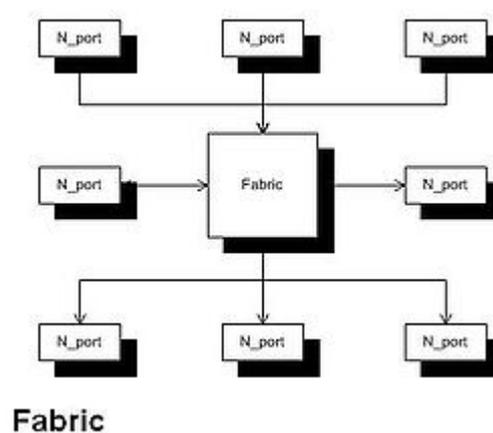
- **Arbitrated Loop (FC-AL):** En esta topología todos los dispositivos están conectados en anillo, similar a una red Token ring. El hecho de añadir o quitar dispositivos al anillo causa que toda la actividad del anillo sea interrumpida. Igualmente el fallo de uno de los dispositivos conectados al anillo, interrumpen el funcionamiento del mismo. Existen concentradores FC que conectan

múltiples dispositivos entre si y en caso de fallo pueden puentear el puerto fallido.



7. Diagrama A-L.

- **Switched fabric (FC-SW):** Todos los dispositivos o “anillos” de dispositivos son conectados a Switches Fibre Channel. Las ventajas de esta tipología sobre FC-P2P o FC-AL son:
 - Los switches administran el estado del fabric y proveen conexiones optimizadas.
 - El flujo de datos entre 2 puertos, sólo pasa a través de switches, no por otros puertos.
 - El fallo de un puerto está aislado y no afecta a ningún otro puerto.



8. Diagrama Fabric.

En la siguiente figura podemos ver las principales diferencias entre el uso de una topología y otra:

Attribute	Point-to-Point	Arbitrated loop	Switched fabric
Max ports	2	127	$\sim 16777216 (2^{24})$
Address size	N/A	8-bit ALPA	24-bit port ID
Side effect of port failure	N/A	Loop fails (until port bypassed)	N/A
Mixing different link rates	N/A	No	Yes
Frame delivery	In order	In order	Not guaranteed
Access to medium	Dedicated	Arbitrated	Dedicated

9. Diferencias entre topologías SAN.

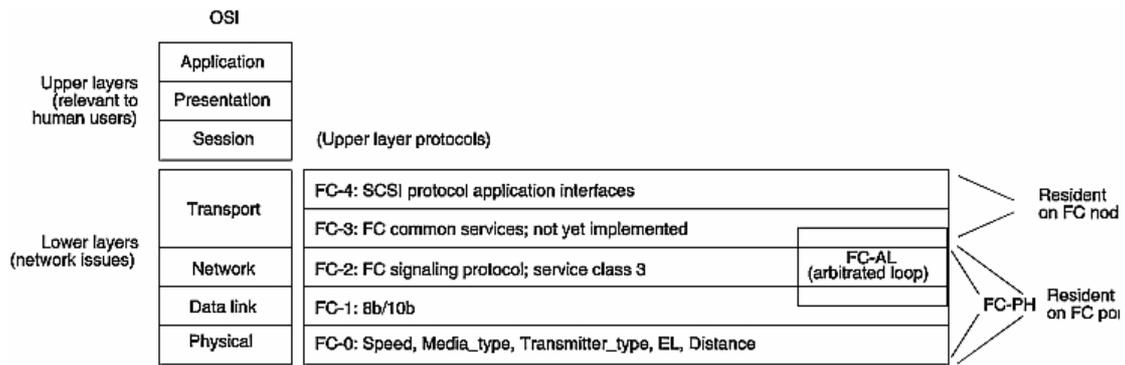
2.1.1.1 El estándar FC

El estándar FC comenzó su desarrollo en 1988 principalmente para ser utilizado en el campo de la supercomputación, pero acabó convirtiéndose en el estándar utilizado para las redes de almacenamiento en el mundo empresarial. En un principio se pensó para simplificar las funciones del sistema HIPPI pero el estándar acabó soportando diversos protocolos de transmisión de datos como SCSI, HIPPI, Ethernet, IP y ATM. A pesar de lo que su nombre indica puede utilizar como medio de transmisión fibra óptica o pares de cobre. De ahí que se nombre “fibre” y no fiber, ya que aunque originalmente fue diseñado para trabajar sobre fibra óptica, al habilitarse el soporte sobre cobre se decidió cambiar la palabra fiber por el término francés “fibre” para reducir la asociación con la fibra óptica.

El protocolo fibre channel está dividido en 5 capas las cuales guardan algunas similitudes con el modelo de red OSI. Estas 5 capas son:

- **FC0** La capa física, que incluye los cables, la óptica de la fibra, conectores.
- **FC1** La capa de enlace de datos, que implementa la codificación y decodificación de las señales.
- **FC2** La capa de red, definida por el estándar FC-PI-2, que constituye el núcleo de Fibre Channel y define los protocolos principales.
- **FC3** La capa de servicios comunes, la cual eventualmente puede implementar funciones de encriptado o redundancia.
- **FC4** La capa de “mapeo” de protocolos. En esta capa, protocolos como SCSI son encapsulados dentro una unidad de información para ser enviada a la FC2.

Las capas FC0, FC1 y FC2 son conocidas como la capa física de Fibre Channel (FC-PH).



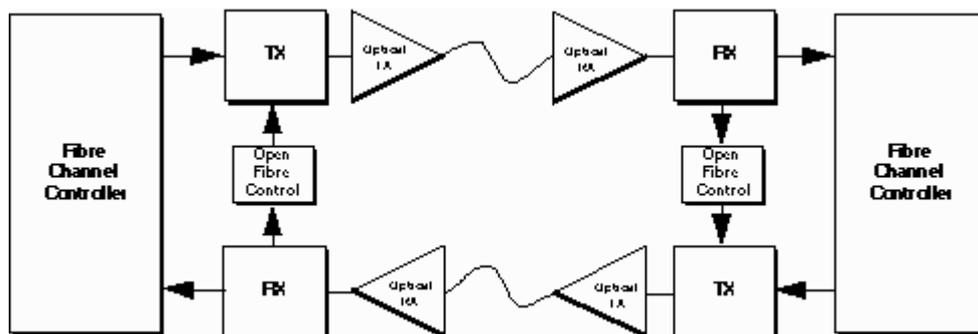
10. FC Layers.

2.1.1.1.1 FC-0 layer

El nivel más bajo del estándar FC define la conexión física, incluyendo cable, conectores y parámetros ópticos y eléctricos para gran variedad de velocidades.

Este nivel está diseñado para proveer alta flexibilidad y permitir el uso de un gran número de tecnologías, permitiendo que una ruta de comunicación entre 2 nodos, puede realizarse con un link sobre diferentes tecnologías. Por ejemplo en destino una señal puede transmitirse sobre pares de cobre y ser convertida para transmitirse sobre una fibra monomodo para cubrir grandes distancias.

Además, debido a que FC utiliza a menudo láser entre links, implementa un sistema de seguridad para que los niveles de potencia óptica no superen los definidos por las normas de seguridad láser (en entornos de computación debe cumplir la clase 1, que es la más exigente) de exposición de usuarios a radiaciones láser. Este sistema se denomina OFC (Open Fibre Control) y básicamente consiste en asegurar que ambos extremos de un link de fibra óptica están conectados antes de que las señales de láser comiencen a ser transmitidas, para evitar que el láser emita cuando se detecta que el link de fibra está abierto.



11. Layer 0.

2.1.1.1.2 FC-1 layer

Esta capa define el protocolo de transmisión, mantiene el control de DC y provee métodos de alineación de trama e integración de pulsos que permitan recuperar la información del reloj (sincronismo).

Se utiliza el código de línea 8b/10b que es un tipo de codificación para transmisión de datos en líneas de alta velocidad que codifica 8 bits de datos sobre una trama de 10 bits

(símbolo). Los 5 bits más bajos de datos son codificados dentro de un grupo de 6 bits (5b/6b) y los 3 bits más altos son codificados dentro de un grupo de 4 bits (3b/4b).

8b/10b surgió debido a la alta velocidad de transferencia de datos, que hace que los bits transmitidos se mantengan durante muy poco tiempo en la línea, siendo necesario disponer de relojes muy precisos para sincronizar el conjunto emisor/receptor. Sin embargo, emisor y receptor se pueden sincronizar fácilmente cuando los bits que llegan por la línea cambian rápidamente. Atendiendo a este hecho, la codificación 8b10b transforma las cadenas de 8 bits (hasta 256 valores) en cadenas de 10 bits (hasta 1024 valores) con la restricción de no tener más de cinco ceros o cinco unos seguidos. Gracias a esta codificación, no es necesario disponer de un reloj tan preciso, ya que la sincronización entre emisor y receptor se realiza ayudándose de los bits que son transmitidos entre ellos.

Gracias a los 1024 valores posibles con 10 bits, el sistema sólo utiliza aquellos que tienen un número similar de ceros y unos, existiendo estas tres posibilidades:

- 5 unos y 5 ceros: código con disparidad neutra
- 6 unos y 4 ceros: código con disparidad positiva
- 4 unos y 6 ceros: código con disparidad negativa

El objetivo es que se cancele la disparidad, de esta forma, el nivel de la componente continua de la señal eléctrica es nulo permanentemente.

Algunos ejemplos de sistemas que utilizan codificación de línea 8b/10b son:

- Serial Advanced Technology Attachment
- Serial Attached SCSI
- Fibre Channel
- PCI Express
- Gigabit Ethernet
- Infiniband
- DvB
- HDMI

2.1.1.1.3 FC-2 Layer

En esta capa se definen los diferentes mecanismos para controlar la transmisión de los 3 tipos de servicios que soporta, además de la señalización para el control de trama.

Para implementar los servicios de esta capa, en el estándar se definen 7 bloques:

- a) Ordered Set
- b) Frame
- c) Sequence
- d) Exchange

- e) Protocol
- f) Flow Control
- g) Service Classes

Ordered Set

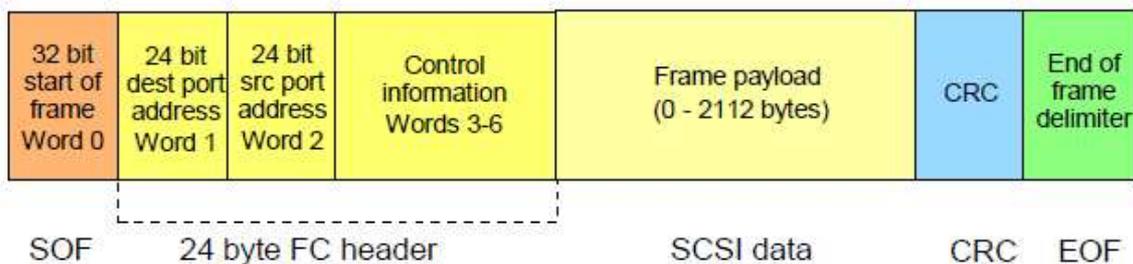
El “ordered set” consiste en la transmisión de 4 bytes que contienen datos junto con algunos caracteres que tienen un significado especial. Ordered Set posibilita la sincronización de trama y siempre comienza por el carácter espacial “K28.5”. Los 3 principales tipos de “Ordered Sets” están definidos por el protocolo de señalización

Los “ordered set” para delimitación de trama (Start-of-Frame i.e SOF y End-of-Frame i.e EOF) siempre van precediendo o terminando el contenido de un Frame.

Hay múltiples delimitadores SOF y EOF definidos por las secuencias de control dependiendo de la topología de red utilizada.

Frame

Los frame son el bloque básico para una conexión FC. Los Frames contienen la información que debe transmitirse (payload), la dirección de los puertos de origen y destino e información para control del link. Los frame son comúnmente catalogados como Frame de datos (Data Frame) y Frame de control.



12. Campos de un Frame.

Sequence

Una secuencia está formada por un conjunto de 1 o más frame contiguos transmitidos de manera unidireccional desde un puerto a otro. Cada frame dentro de una secuencia es numerado inequívocamente con un número de secuencia. La recuperación de errores es controlada por un nivel superior que controla los límites de la secuencia.

Exchange

Un “Exchange” se compone de 1 o más secuencias no concurrentes para una sola operación. Los intercambios pueden ser unidireccionales o bidireccionales entre 2 puertos. Dentro de un mismo Exchange, sólo una secuencia puede estar activa al mismo tiempo, pero secuencias de diferentes Exchanges si pueden estar activas concurrentemente.

Protocol

Este bloque de protocolos se refiere a los servicios ofrecidos por FC. Los protocolos pueden ser específicos de servicios de capas superiores para los cuales FC provee su

propio set de protocolos para administrar el entorno operativo de una transferencia de datos. Los siguientes protocolos son los especificados por el estándar:

- Protocolos de secuencias simples y detección de link(Ordered Set)
- Fabric Login protocol: The interchanging of Service Parameters of an N_Port with the fabric.
- N_Port Login protocol: Before performing data transfer, the N_Port interchanges its Service Parameters with another N_Port.
- Data transfer protocol describes the methods of transferring Upper Layer Protocol (ULP) data using the Flow control management of Fibre Channel (see chapter 5.6).
- N_Port Logout Protocol is performed when an N_Port requests removal of its Service Parameters from the other N_Port. This may be used to free up resources at the connected N_Port.

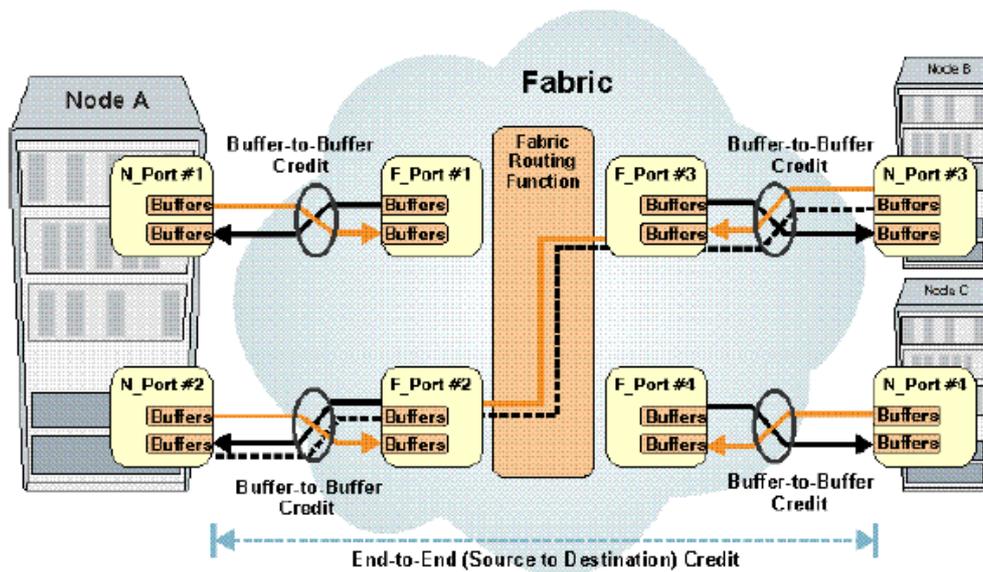
Flow control

El control de Flujo es el proceso encargado de que los frame entre N_Ports y N_port y el fabric, no saturen el receptor. Se utilizan 2 contadores BB_Credit_CNT y EE_Credit_CNT y ambos son inicializados a cero durante la fase de login.

El tipo de control de flujo utilizado depende de las clases de servicio utilizadas. La clase 1 utiliza control de flujo end-to-end. La clase 2 utiliza sólo buffer-to-buffer y la clase 2, utiliza ambos. En todos los tipos el control de flujo se controla por un mensaje “Sequence Initiator “(emisor) y un Sequence Recipient (destino) utilizando Credit y Cred_CNT. El Credit_CNT representa el número de frame de datos que aún no han sido confirmados (ACK) por el Sequence Recipient.

- El control de flujo extremo a extremo, defino el flujo de frames entre N_Ports. En este caso, el Sequence Recipient es responsable de aceptar los frame de datos válidos con frames de ACK. Cuando el número de buffers es insuficiente para el Frame que llega, se envía un Fram “Busy”, y cuando un Frame se recibe con error, se envía un “Reject” frame.
- El control de flujo buffer-to-buffer es administrado entro un N_Port y un F_Port o entre N_ports en topologías punto a punto. Cada puerto es responsable de la negociación durante el Fabric Login.

En la siguiente figura podemos ver el flujo de comunicación así como el mecanismo de control de flujo utilizado en cada lado de la comunicación:



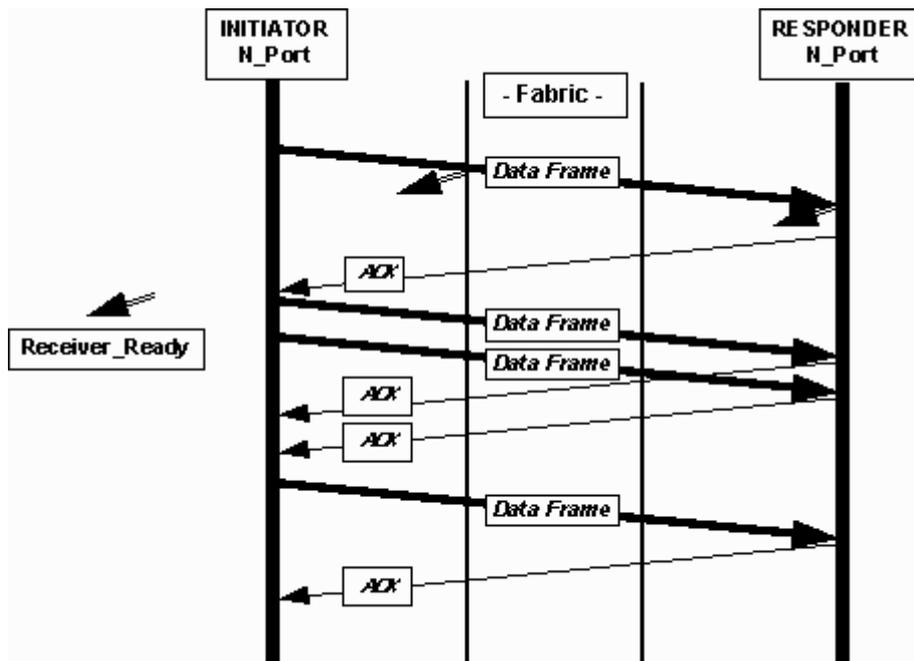
13. Diagrama de Control de Flujo.

Service Classes

Para asegurar la transmisión de diferentes tipos de tráfico, FC define 3 clases de servicios. El usuario selecciona la clase de servicio dependiendo de las características de sus aplicaciones definiendo parámetros como la longitud de paquetes y la duración de transmisión.

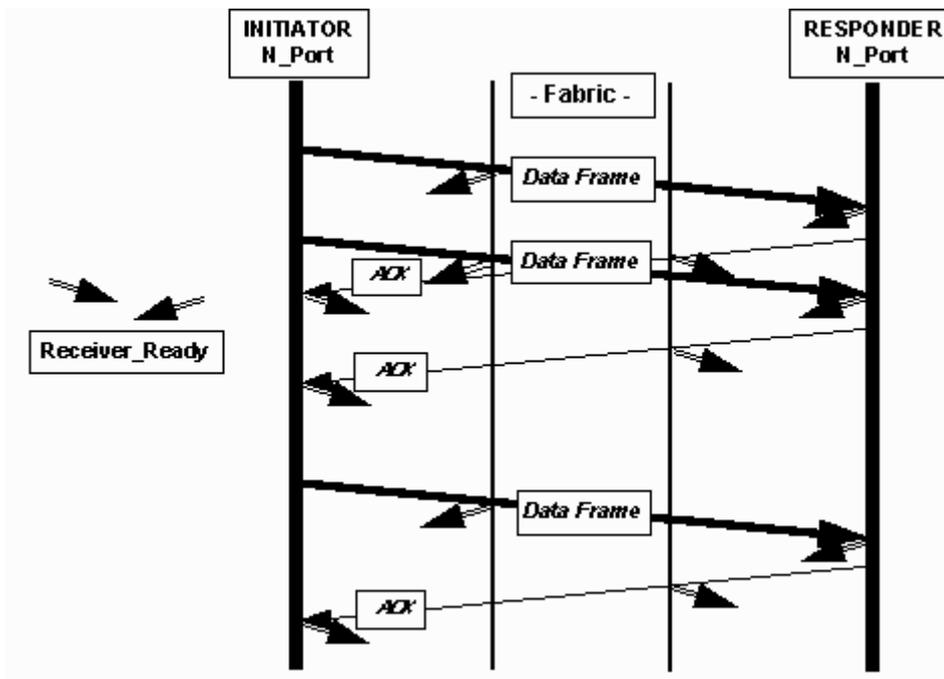
El servicio de clase 1 es un servicio que provee conexiones dedicadas. Es el equivalente a una conexión física dedicada. Una vez establecida, una conexión de clase 1 es mantenida y garantizada por el fabric. Este servicio garantiza el máximo ancho de banda entre 2 N_ports y es la mejor elección para soportar transacciones con alta capacidad. En esta conexión, los Frame son entregados al receptor en el mismo orden que se transmitieron.

La siguiente ilustración muestra la gestión de control de flujo en una conexión de clase 1.



14. Control de Flujo Clase 1.

El servicio de clase 2 es un “Frame-switched”, es decir no orientado a conexión. Este servicio permite compartir el ancho de banda multiplexando frame desde múltiples emisores sobre el mismo canal o canales. El fabric no garantiza el orden de entrega y los frame pueden ser entregados desordenados. Las clases 1 y 2 envían frame de aceptación (Acknowledgment Frames). Si la entrega no puede realizarse por un problema de congestión, un Busy Frame es devuelto y el envío se reintenta. En la siguiente ilustración podemos ver el flujo de mensajes.



15. Control de flujo clase 2.

La clase de servicio 3, es idéntica a la clase 2 excepto porque el envío de frame no es confirmado (no se envían Ack Frame). Ese servicio es útil para transmisiones en tiempo real donde los tiempos son clave.

El estándar FC también define un modo de servicio opcional llamado “intermix”. Intermix es una opción de la clase de servicio tipo 1, en la cual se garantiza una cantidad de ancho de banda para los frame de tipo 1, y los frame de clase 2 y clase 3 se multiplexan en el canal sólo cuando hay suficiente ancho de banda disponible para compartir el canal.

2.1.1.1.4 FC-3 Layer

La capa 3 de FC define los mecanismos necesarios para prestar servicios avanzados tales como:

- **Striping** – Para aumentar el ancho de banda utilizando múltiples N_Ports en paralelo utilizados como si fuesen un único canal de transmisión.
- **Hunt groups** – La capacidad de que más de un puerto pueda responder a la misma dirección de alias (se define un alias). Esto mejora la eficiencia ya que disminuye la posibilidad de llegar a un N_Port ocupado.
- **Multicast** – La posibilidad de realizar una sólo transmisión a múltiples puertos de destino. Esto incluye el envío a todos los N_Port conectados a un fabric (broadcast) o sólo a un subconjunto de N_ports del fabric.

2.1.1.1.5 FC-4 Layer

La capa 4 de FC es el nivel más alto en la estructura definida por el estándar. Ésta define los interfaces de aplicaciones que pueden ejecutarse sobre FC y especifica las reglas para mapear protocolos de capa superior utilizando los niveles más bajos definidos en FC. Fibre channel está adaptada para poder transportar datos de red y de canal y permite que sean transportados de manera concurrente sobre la misma interfaz física.

Los siguientes protocolos de red y de canal están soportados por FC:

- Small Computer System Interface (SCSI)
- Intelligent Peripheral Interface (IPI)
- High Performance Parallel Interface (HIPPI) Framing Protocol
- Internet Protocol (IP)
- ATM Adaptation Layer for computer data (AAL5)
- Link Encapsulation (FC-LE)
- Single Byte Command Code Set Mapping (SBCCS)
- IEEE 802.2

2.1.2 Network Attached Storage

NAS (Network Attached Storage) es el nombre de la tecnología de almacenamiento que basa su funcionamiento en la capacidad de compartir el almacenamiento de un servidor

con ordenadores personales o servidores clientes a través de una red (usualmente es TCP/IP) con una arquitectura tipo cliente/servidor.

Generalmente los sistemas NAS son dispositivos dedicados a los que se accede desde los equipos a través de protocolos de red (como SMB, NFS , FTP y HTTP) basados en ficheros, i.e, el cliente solicita el fichero completo al servidor y lo maneja localmente, a diferencia de las redes SAN en el que su protocolo está orientado a dispositivos de bloque.

Generalmente constan de un dispositivo simple conocido como “nas box” o “nas head” que actúa como interfaz entre el almacenamiento y los clientes. Los clientes siempre se conectan al “nas head” a través de una conexión ethernet.

Debido a la capacidad multiprotocolo y a la reducción del sistema operativo para ser embebido en el hardware, los dispositivos NAS tienen sus limitaciones en comparación con DAS y SAN.

Si el dispositivo NAS está ocupado atendiendo peticiones de muchos usuarios y con mucha carga por operaciones de E/S, su rendimiento baja y afecta a todos los clientes.

Es importante señalar que un servidor NAS es en sí mismo un servidor, con CPU, placa base, memoria RAM, etc. y su fiabilidad está supeditada a su diseño, dependiendo de si dispone de controladores redundantes, fuentes de alimentación redundantes, teniendo en cuenta que los sistemas NAS tienen su aparición como alternativa para reducir el coste asociado de los servidores de archivos tradicionales.

Los 2 protocolos más utilizados con la tecnología NAS son NFS y CIFS, y ambos se basan en una arquitectura cliente/servidor. Estos 2 protocolos además son bastante más antiguos que los dispositivos NAS, ya que ambos fueron desarrollados sobre la década de los 80.

2.1.2.1 NFS

NFS es un protocolo de sistemas de ficheros en red que originalmente fue desarrollado por SUN Microsystems en 1984, y permitía que servidores clientes pudieran acceder a ficheros a través de la red como si el filesystem fuese local. NFS está construido sobre Open Network Computing Remote Procedure Call (ONC RPC) y es un estándar abierto definido en las RFC, por lo que cualquiera puede implementarlo.

Existen varias versiones de NFS, desde su desarrollo hasta la actualidad:

- NFS Original o NFSv1 (1984): Esta versión solo fue utilizada por Sun para propósitos experimentales. Cuando el equipo de desarrollo fue añadiendo cambios a la versión 1, Sun decidió presentar la versión 2.
- NFSv2 (1989): La versión 2 del protocolo originalmente trabajaba completamente sobre UDP. Esto implica que no era orientado a conexión.
- NFSv3 (1995): En esta versión se añaden nuevas funcionalidades debido a la evolución tecnológica del hardware.
 - Soporte para ficheros de más de 2GB
 - Soporte para escritura asíncrona sobre el servidor, mejorando el rendimiento

- Atributos adicionales para los ficheros
 - Una operación REaddirPlus para obtener los identificadores de archivo y los atributos, junto con los nombres de archivo cuando se escanea un directorio
- NFSv4 (2000): En esta versión se incluyen mejoras en el rendimiento, mejoras sobre seguridad obligatorias y se introduce un protocolo de estado. Esta versión es la primera versión desarrollada por el IETF después de que Sun Microsystems abandonara el desarrollo. En esta versión, una de las características más importantes es el soporte para proveer acceso a sistemas de ficheros distribuidos (Parallel file system).

2.1.2.2 SMB

SMB (Server Message Block) es un protocolo de red que permite compartir archivos e impresoras entre nodos de una red. Es utilizado principalmente por ordenadores que corren sistema operativo Windows. Fue desarrollado originalmente por IBM pero la versión utilizada hoy en día es la modificada por Windows, CIFS. Microsoft renombró SMB a CIFS (common Internet File System) en 1998.

Para el mundo Unix existe la implementación libre del protocolo SMB, conocida como Samba.

2.1.3 Ip storage

En la última década, la industria de las tecnologías de almacenamiento se ha esforzado por desarrollar alternativas a las redes de almacenamiento FC basadas en Ethernet e IP, que comúnmente se conoce como IP Storage.

En relación con el almacenamiento IP, FC tiene sus ventajas debido a que es muy eficiente, en parte debido a su pila de protocolos. Sin embargo TCP/IP es aparentemente menos eficiente debido a la sobrecarga de datos en las distintas capas del protocolo, lo que aumenta la complejidad de utilización para la utilización en redes de almacenamiento.

Los distintos protocolos que forman la familia de IP Storage incluyen Fibre Channel sobre IP (FCIP), Internet SCSI (iSCSI), Fibre Channel over Ethernet (FCoE), Ata over Ethernet e Internet Fibre Channel (iFCP).

Aunque los protocolos son muy diferentes entre sí todos ellos buscan una meta común: el transporte de dispositivos en modo bloque a través de una red IP. Esto tiene una ventaja directa, y es el aprovechamiento de la gran infraestructura de redes Ethernet TCP/IP existente en la actualidad y el avance en las limitaciones geográficas derivadas de la tecnología en las redes SAN.

2.1.3.1 FCIP

Fibre Channel over IP (también conocido como Fibre Channel Tunneling o Storage Tunneling) define los mecanismos necesarios para realizar la transmisión de frames Fibre Channel sin modificar a través de la red IP. Debido a que la mayoría de las organizaciones ya tienen una infraestructura IP, esta tecnología se convierte en un atractivo para conectar redes SAN separadas geográficamente a un costo bajo a través de redes LAN, MAN o WAN. Para ello se utilizan equipos conversores (Edge Devices

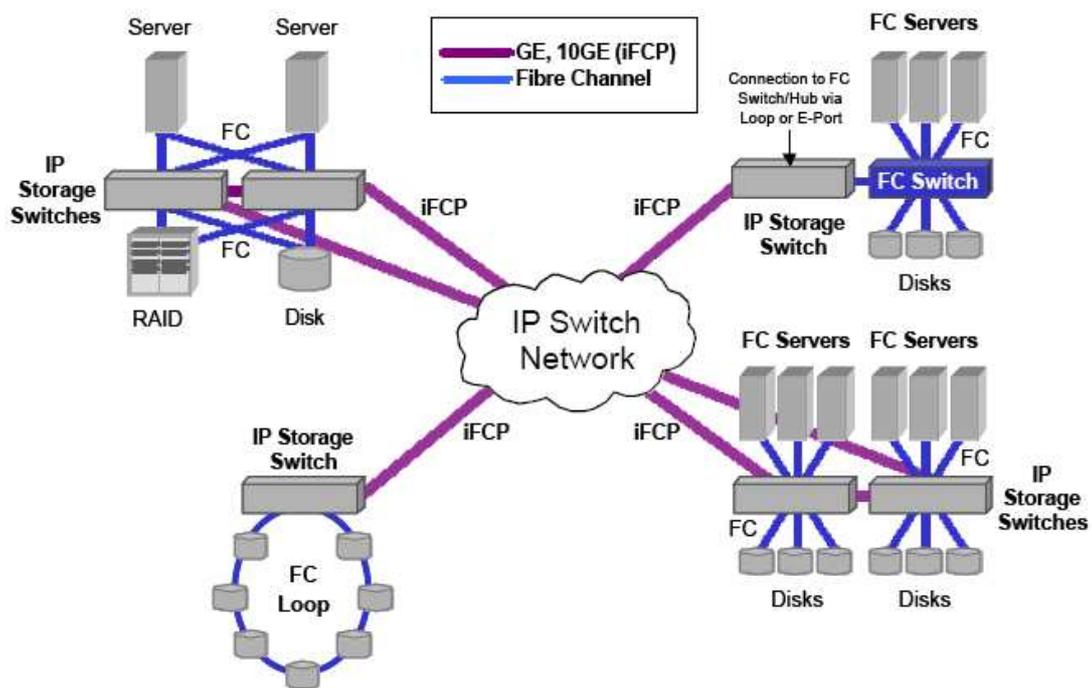
o FCIP Gateway) entre las redes SAN que se quieren interconectar. Estos dispositivos se limitan a encapsular y reenviar las tramas FC vía TCP/IP a través de un socket o túnel de forma totalmente transparente. Los servicios TCP/IP se utilizan para establecer la conexión, realizar el control de congestión y gestión así como el control y recuperación de errores, pero no afectan a los servicios del fabric FC.

Otro de los puntos importantes de FCIP es que no reemplaza FC con IP sino que simplemente permite el despliegue de nuevos fabric utilizando túneles IP convirtiendo varias redes SAN separadas geográficamente en una única red SAN unificada.

2.1.3.2 iFCP

La especificación iFCP define a este como un protocolo Gateway-to-Gateway para la implementación de una red FC fabric en la cual los switches y routers TCP/IP reemplacen a los componentes FC. La principal diferencia existente entre FCIP e iFCP es que en iFCP la capa de transporte se sustituye por Gigabit Ethernet y TCP/IP.

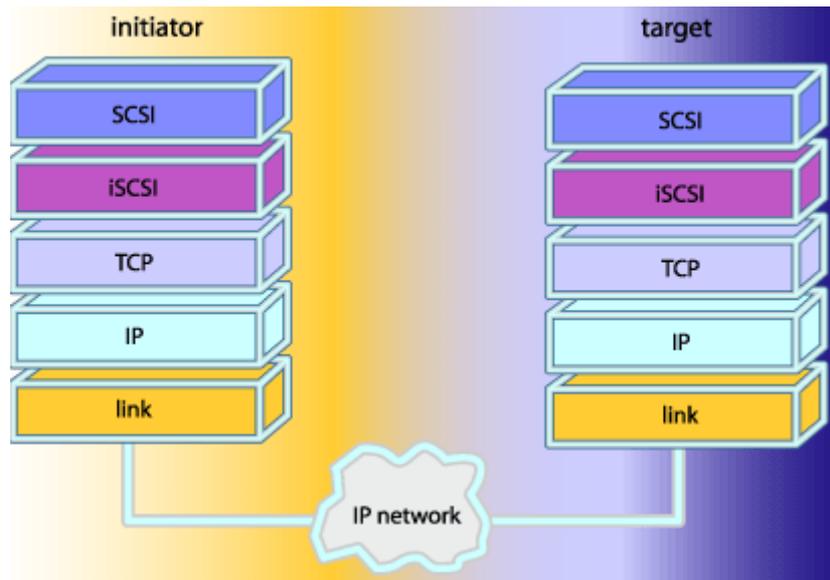
Con iFCP los dispositivos Fibre Channel se conectan a un gateway o switch iFCP y cada sesión FC se termina en el gateway local y se convierte a una sesión TCP/IP a través de iFCP. Un segundo gateway o switch recibe la sesión iFCP e inicia una sesión FC.



16. Conexión iFCP.

2.1.3.3 iSCSI

iSCSI (Internet SCSI) es un estándar oficial que permite la utilización del protocolo SCSI sobre redes TCP/IP. Al contrario que otros protocolos de red para almacenamiento (como FC) solamente requiere una interfaz Ethernet para funcionar. Esto permite tener una solución de almacenamiento centralizada de bajo coste, sin la necesidad de realizar inversiones costosas ni sufrir las habituales incompatibilidades asociadas a las soluciones FC.



17. Bloques iSCSI.

Dentro de esta especificación hay varios conceptos importantes que hay que conocer para comprender cómo funciona iSCSI:

- a) **INITIATOR:** Se llama initiator al cliente SCSI y este tiene el mismo objetivo que si fuese un bus adaptador iscsi solo que en vez de estar cableado, envía los comandos scsi a través de la red TCP/IP. Además el initiator se divide en 2 grandes grupos:
 - Software initiator: Un software initiator utiliza código para implementar iSCSI. Típicamente se implementa como un driver de dispositivo que utiliza una tarjeta de red existente (NIC) y la pila de protocolos existente para emular el dispositivo SCSI. Los software initiators están disponibles en la mayoría de sistemas operativos y es el tipo más utilizado en computación
 - Hardware initiator: Un hardware initiator utiliza un hardware dedicado en combinación con software (firmware) corriendo sobre el hardware dedicado para implementar iSCSI. Un hardware initiator mueve la sobrecarga del procesamiento iSCSI, TCP e interrupciones gestionándolo el propio hardware y liberando al servidor de esta carga computacional.
- b) **Host Bus Adapter:** Comúnmente conocido como HBA es la implementación de un Hardware Initiator. Una HBA típica se compone de una combinación de un NIC Gigabit (o 10 G) y algún tipo de motor para tratar el tráfico iSCSI.
- c) **TARGET:** En la especificación se hace referencia al target como un recurso de almacenamiento ubicado en un servidor iSCSI y mostrado a un cierto número de initiators. En general el target representa un disco duro que trabaja sobre la red IP. Al igual que ocurre con los initiators, el software para crear un target iSCSI está disponible para la mayoría de los sistemas operativos.

- d) **LUN:** En terminología SCSI, una Lun (Logical Unit Number) representa un dispositivo scsi individual direccionable que es parte de un dispositivo scsi Target. En un entorno SCSI, las Lun's son esencialmente los números de dispositivos de disco. Un initiator negocia con un target para establecer conexión con una LUN. El resultado es una conexión iSCSI que emula una conexión a un dispositivo SCSI. Los initiators tratan las Lun iSCSI de la misma forma que si fuesen discos SCSI o IDE.
- e) **ADDRESSING:** iSCSI trata 3 tipos de direccionamiento (referidos tanto al initiator como al target):
- **IQN (iSCSI Qualified Name):** Puede tener hasta 255 caracteres de longitud, con el siguiente formato:

Iqn.year-mo.reverse_domain_name:unique_name

Year-mo representa el año y el mes en que se registró el dominio

Reversed_domain_name es el nombre del dominio oficial en orden inverso

unique_name es cualquier nombre que queramos utilizar. Típicamente se coge el nombre del servidor.

- **EUI (Extended Unique Identifier):** Este formato permite a una “naming authority” a utilizar los identificadores IEEE EUI-64 para construir nombres iSCSI. Siempre empiezan con “eui” seguidos de un código definido por el IEEE en código ASCII.
- **NAA (T11 Network Address Authority):** Define un formato para construir identificadores globales únicos con el formato NAA. Siempre son del tipo “naa.”, seguido de un identificador de NAA representado en código ASCII. Este tipo apareció para proveer compatibilidad con las convenciones de nombres utilizadas en tecnologías FC y SAS

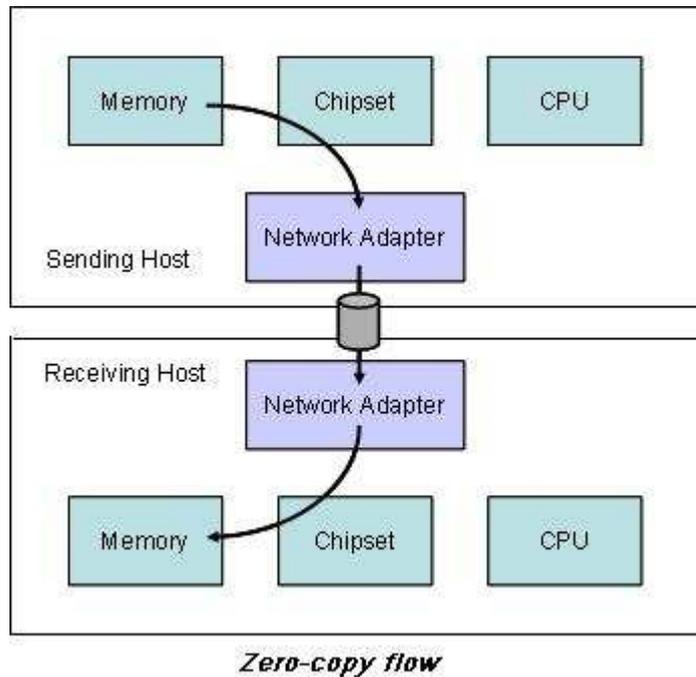
2.1.3.3.1 ISER

Una de los últimos desarrollos aparecidos para mejorar el rendimiento de iSCSI es el llamado iSER (iSCSi Extensions for RDMA).

RDMA (Remote direct memory access) es un acceso directo a memoria desde la memoria de un servidor a otro sin implicar a ninguno de sus sistemas operativos.

Esto permite un alto “throughput”, baja latencia en la red y es utilizado principalmente en clusters tipo parallel. RDMA soporta zero-copy (operaciones en las cuales la CPU no realiza ninguna tarea de copiar datos entre una memoria y otro área) permitiendo que el adaptador de red transfiera datos directamente desde o hacia la memoria, eliminando la necesidad de copiar datos entre la memoria de aplicaciones y los buffer del sistema operativo. Estas transferencias no requieren tiempo de CPU, caché o cambios de contexto por lo que aumenta mucho el rendimiento. Cuando una aplicación realizad una lectura o escritura RDMA, los datos se entregan directamente a la red reduciendo la latencia y permitiendo transferencias de mensajes mucho más rápidas.

El protocolo ISER mapea el protocolo iSCSI sobre una red que provee servicios RDMA. Esto permite que los datos sean transferidos directamente dentro de los buffer SCSI I/O sin copias intermedias.

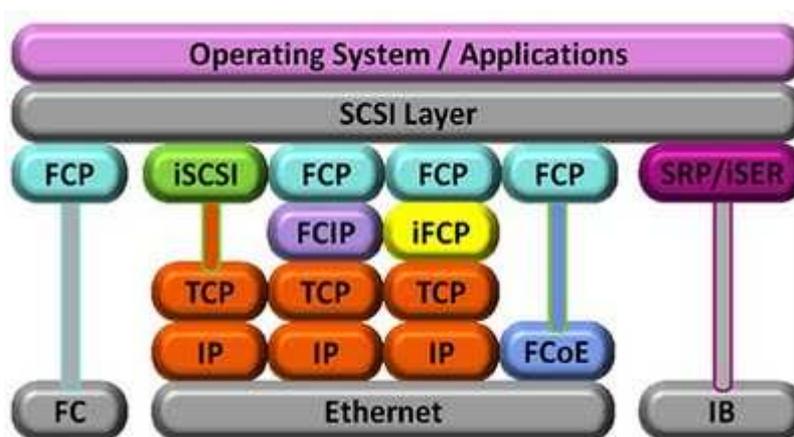


18. Diagrama RDMA.

2.1.3.4 FCOE

Fcoe es el último estándar en aparecer y permite transportar nativamente FC sobre Ethernet. La especificación de Fcoe reemplaza las capas FC0 y FC1 de la pila FC con Ethernet.

Con Fcoe, Fibre Channel se convierte en otro protocolo de red más, y en contraste con iSCSI que opera por encima de TCP/IP Fcoe opera directamente sobre Ethernet.



19. Diagrama de bloques FCoE.

2.2 Ejemplos de sistemas actuales

Actualmente existen varios fabricantes que se han especializado en ofertar soluciones de storage para satisfacer las demandas del almacenamiento de datos existente en la actualidad. Además, tal y como sucede en otros aspectos del IT podemos elegir entre soluciones de menos calidad y más económicas como soluciones tecnológicamente punteras pero mucho más costosas. Esto se traduce en una amplia gama de productos las cuales abarcan la mayoría de las necesidades actuales (desde soluciones para pequeñas y medianas empresas como soluciones para grandes multinacionales).

Los productos que podríamos denominar de gama alta, teniendo en cuenta la aceptación empresarial y las características ofertadas son los que ofrece el fabricante EMC² seguido de NetApp.

En EMC² la familia de productos orientados al almacenamiento NAS se denominan “*celerra*”. Dentro de la gama *celerra* todos ofrecen además de las capacidades NAS, capacidades iSCSI.

La gama de acceso *celerra* comienza con el modelo *Celerra NS-120* (permite escalabilidad de la plataforma hasta 120 discos). Además ofrece la característica MPFS (Multi-path FileSystem) que aumenta el rendimiento de forma considerable.

El tope de gama en la familia *Celerra* está en el modelo *NS-960*, que además de iSCSI ofrece almacenamiento SAN mediante FC. Además dispone de mecanismos de alta disponibilidad teniendo varios dispositivos en cluster que pueden realizar failover el uno sobre el otro en caso de fallo hardware.

En Netapp la familia de productos orientados al almacenamiento NAS se denominan “*FAS*”, y dentro de toda la gama *FAS* todos los productos ofrecen capacidades iSCSI.

El tope de gama en la familia *FAS* se encuentra en los modelos *FAS60X0*. Todos ofrecen además capacidades FC y métodos para proporcionar alta disponibilidad (posibilidad de dispositivos en cluster).

La gama de acceso en NetApp se encuentra en los dispositivos *FAS20X0*. Además de disponer de menos capacidad de almacenamiento, se abaratan costes en el hardware de control, ya que disponen de bastante menos memoria RAM que las gamas medias y/o altas.

Si las necesidades son menos exigentes existen fabricantes menos conocidos en el mercado con productos de acceso económicamente más accesibles que las soluciones de EMC² y/o NetApp. Por ejemplo se encuentran productos de Buffalo Technology que ofrecen características bastante razonables y económicamente no son soluciones especialmente costosas.

En Buffalo Technology no todos los productos dedicados a almacenamiento NAS proveen capacidad SAN mediante iSCSI. Para encontrar un producto que ofrezca iSCSI hay que ver la familia “*TeraStation iSCSI*”. Dentro de esta familia tenemos productos con capacidad desde los 2TB hasta los 8TB del tope de gama, proporcionando tasas de transferencia de hasta los 92MB/s en situaciones ideales.

Además, en los últimos años han aparecido proyectos opensource dedicados a convertir un servidor, en un servidor de almacenamiento como un sistema embebido utilizando software opensource (utilizan como sistema operativo distribuciones de Linux

modificadas). Los 2 proyectos más importantes en los cuales se sigue trabajando son “FreeNas” que está basado en FreeBSD, y OpenFiler que utiliza una distribución de Linux denominada “rPath” y que se utiliza para la implementación de sistemas embebidos (Openfiler, SugarCRM, AsteriskNOW).

Ambos proyectos están enfocados a convertir un servidor en una solución de almacenamiento multiprotocolo de bajo coste.

3. Definición del entorno

3.1 Estudio del software de virtualización a utilizar

Cuando hablamos de virtualización, nos referimos a la abstracción de los recursos de una computadora, llamada Hypervisor o VMM (Virtual Machine Monitor) que crea una capa de la abstracción entre el hardware de la maquina física (host) y el sistema operativo de la maquina virtual (virtual machine, guest), siendo un medio para crear una versión virtual de un dispositivo o recurso, como un servidor, un dispositivo de almacenamiento, una red o incluso un sistema operativo, donde se divide el recurso en uno o más entornos de ejecución.

Esta capa de software (VMM) maneja, gestiona y arbitra los cuatro recursos principales de una computadora (CPU, Memoria, Red, Almacenamiento) y así puede repartir dinámicamente dichos recursos entre todas las maquinas virtuales definidas en el computador central de modo que nos permite tener varios ordenadores virtuales ejecutándose sobre el mismo ordenador físico.

3.1.1 Evolución en la tecnología de virtualización

El término “virtualización” es un concepto desarrollado en la década de 1960 por IBM como una manera de particionar de forma “lógica” el hardware de mainframe en distintas máquinas virtuales. Estas particiones permitían ejecutar múltiples aplicaciones y procesos al mismo tiempo. Debido a que los recursos de mainframe eran muy costosos, la virtualización se diseñó para poder aprovechar plenamente la inversión.

La virtualización se abandonó durante los años 1980 y 1990, cuando las aplicaciones cliente-servidor y los servidores x86 de bajo costo y de escritorio se utilizaron en entornos distribuidos. La amplia adopción de Windows y la aparición de Linux como sistemas operativos en servidores estableció en la década de 1990 la arquitectura x86 como estándar en la industria. El crecimiento en servidores x86 y despliegues de escritorio han arrastrado a la industria a infrautilizar la plataforma apareciendo nuevos retos a lograr:

- Baja utilización de la infraestructura. Típicos despliegues de servidores x86 que logran una utilización media de sólo un 10% al 15% de la capacidad total. Las organizaciones suelen ejecutar una aplicación por servidor para evitar el riesgo de vulnerabilidades en una aplicación a la existencia de otra solicitud en el mismo servidor.
- Aumentan los costos de mantenimiento de la infraestructura física. La mayoría de la infraestructura debe seguir funcionando en todo momento, resultando en un elevado consumo de energía, refrigeración e instalaciones.
- Aumentan los costos de administración. Las organizaciones dedican mucho tiempo y recursos en tareas manuales asociadas con el mantenimiento de servidores, y por lo tanto requieren más personal para completar estas tareas.

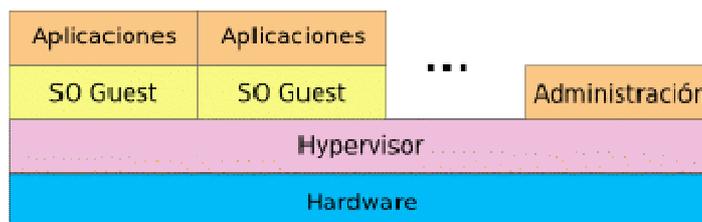
En 1999, VMware introdujo la virtualización para los sistemas x86 para abordar muchos de estos problemas.

3.1.2 Tipos de virtualización

En la actualidad cuando se trata de la virtualización, no hay sólo una manera de hacerlo. De hecho, hay varias maneras de lograr el mismo resultado a través de los diferentes niveles de abstracción. A continuación veremos los 3 modelos de virtualización más comunes.

3.1.2.1 Modelo de máquina virtual (o full virtualizer)

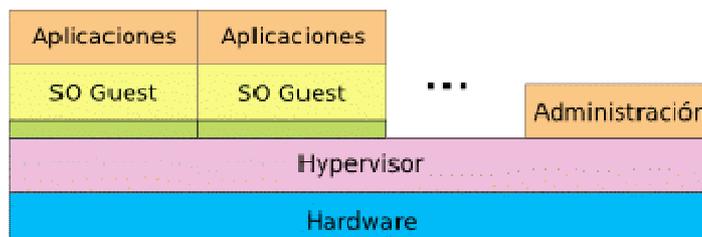
El modelo de máquina virtual está basado en la arquitectura cliente/servidor, donde cada cliente funciona como una imagen virtual de la capa hardware. Este modelo permite que el sistema operativo cliente funcione sin modificaciones. Además permite al administrador crear diferentes sistemas clientes con sistemas operativos independientes entre sí. La ventaja principal de este modelo radica en el desconocimiento por parte de los sistemas huésped del sistema hardware real sobre el que está instalado. Sin embargo, realmente todos los sistemas virtuales hacen uso de recursos hardware físicos. Estos recursos son administrados por un sistema denominado hypervisor que coordina las instrucciones CPU. El hypervisor es denominado comúnmente monitor de máquina virtual (VMM) y es el encargado de validar todas las peticiones e instrucciones de los sistemas virtuales a la CPU, supervisando todas las ejecuciones que requieran cambios de privilegios. Dos sistemas típicos de servidores virtuales son Vmware y VirtualBox.



20. Full Virtualizer.

3.1.2.2 Modelo de máquina paravirtual

El modelo de máquina paravirtual (PVM) se basa, como el modelo anterior, en la arquitectura cliente/servidor, incluyendo también la necesidad de contar con un sistema monitor. Sin embargo, en este caso, el VMM accede y modifica el código del sistema operativo del sistema huésped. Esta modificación se conoce como porting. El porting sirve de soporte al VMM para que pueda realizar llamadas al sistema directamente. Al igual que las máquinas virtuales, los sistemas paravirtuales son capaces de soportar diferentes sistemas operativos instalados en el hardware real. Modelos de máquinas paravirtuales son UML y XEN.



21. Paravirtualización.

3.1.2.3 Modelo de virtualización a nivel de sistema operativo

La virtualización a nivel de sistema operativo se diferencia de las anteriores en que, en este caso, no existe un sistema cliente/servidor propiamente dicho. En este modelo el sistema principal exporta la funcionalidad del sistema operativo desde su propio núcleo. Por esta razón, los sistemas virtuales usan el mismo sistema operativo que el nativo (aunque en la mayoría de los casos pueden instalar distintas distribuciones). Esta arquitectura elimina las llamadas del sistema entre capas, lo que favorece una reducción importante en el uso de CPU. Además, al compartir los ficheros binarios y librerías comunes del sistema en la misma máquina, la posibilidad de escalado es mucho mayor, permitiendo que un mismo servidor virtual sea capaz de dar servicio a un gran número de clientes al mismo tiempo. Ejemplos de sistemas que usan virtualización a nivel de sistema operativo son Virtuozzo y Solaris (zonas o containers)



22. Virtualización en SO.

En la actualidad existen diversas soluciones de virtualización disponibles actualmente, tanto gratuitas como de pago. Las más conocidas son las siguientes:

- Bochs: Es un emulador de procesadores x86 y AMD64 con licencia de software abierto. Bochs puede ejecutarse en distintos sistemas operativos, incluyendo Linux, Windows o incluso la Xbox. Puede además simular varios sistemas operativos como DOS, Windows o Linux.
- Microsoft Virtual PC: suite de virtualización de Microsoft para Windows y para MacOS. VirtualPC emula un PC estándar y todo el hardware asociado.
- Parallels Workstation: software de virtualización de la empresa Parallels Incorporation para procesadores Intel x86.
- QEMU: aplicación de software libre que implementa un emulador de procesador y que incluye un acelerador que permite incrementar la velocidad de las máquinas virtuales.
- Virtual Iron: otra aplicación de virtualización que ha sido de las primeras en aprovechar las capacidades específicas de virtualización de los nuevos procesadores Intel y AMD.
- VMWare: un completo conjunto de aplicaciones de virtualización, con herramientas de pago orientadas a la empresa y otras gratuitas más orientadas al uso personal.
- Xen: Una herramienta muy usada en la comunidad Linux puesto que hasta hace poco tiempo sólo podía usar Linux/Unix como sistema anfitrión. Desde la versión Xen 3.0 se puede instalar en Windows.

- VirtualBox: una herramienta para Windows, Linux y MacOS liberada bajo licencia GPL y con un rendimiento más elevado que el resto de aplicaciones.

3.1.3 Características generales de VirtualBox

VirtualBox es un virtualizador (tipo full virtualizer) para hardware x86 y AMD64/Intel64 dirigido tanto a empresas como usuarios personales.

Algunas de las principales características son:

- Modularidad: VirtualBox tiene un diseño extremadamente modular con interfaces bien definidas de programación interna.
- Descripciones de Virtual Machine en XML: Las opciones de configuración de las Virtual Machine se almacenan enteramente en ficheros XML y son independientes del hardware local, por lo que pueden ser portadas fácilmente a otros equipos.
- Software adicional para los Sistemas operativos Guest: Virtualbox tiene software especial que puede ser instalado dentro de Windows, Linux y Solaris (virtuales) para lograr una integración y un rendimiento mayor. Entre las características proporcionadas por este software adicional encontramos la integración puntero del ratón, soluciones arbitrarias de pantalla (por ejemplo cambiar el tamaño de la ventana del sistema operativo guest).
- Carpetas compartidas: VirtualBox permite declarar ciertos directorios del sistema operativo host como “shared folders” para el intercambio de datos entre el sistema operativo host y guest o entre los guest.
- Soporte integrado iSCSI: Esta característica única permite conectar una máquina virtual directamente a un servidor de almacenamiento iSCSI sin pasar por el sistema Host.
- Multigeneración de Snapshot jerarquizados: Virtualbox puede guardar snapshots del estado de la máquina virtual. Se puede ir hacia atrás en el tiempo y recuperar la máquina virtual al estado anterior.
- Arranque por red PXE: Las tarjetas de red virtuales de VirtualBox soportan totalmente el arranque remoto a través de la Preboot Execution Environment (PXE).
- Controladores virtuales USB: VirtualBox implementa un controlador USB virtual y permite conectar dispositivos USB de manera arbitraria a los sistemas operativos guest sin tener que instalar controladores específicos en el host.
- Protocolo de Escritorio Remoto (RDP): A diferencia de cualquier otro software de virtualización, VirtualBox es totalmente compatible con el estándar Remote Desk Protocol. Una máquina virtual puede actuar como un servidor RDP, lo que permite “ejecutar” la máquina virtual de manera remota en un cliente ligero que solo muestra los datos de RDP
- USB sobre RDP: Otra característica única de virtualbox es que la máquina servidor de RDP puede acceder a los dispositivos USB conectados del cliente RDP.

3.1.3.1 Sistemas operativos “anfitriones” soportados

- Windows hosts:
 - Windows Xp 32-bit (Todos los Service Pack)
 - Windows Server 2003 32-bit
 - Windows Vista (32-bit y 64-bit)
 - Windows Server 2008 (32-bit y 64-bit)
 - Windows 7 (32-bit y 64-bit)
- Mac OS X hosts:
 - 10.5 (Leopard, 32-bit)
 - 10.6 (Snow Leopard, 32-bit y 64-bit)
- Linux Hosts (32-bit y 64-bit) soporta entre otros:
 - Debian GNU/Linux 3.1 (“sarge”), 4.0 (“etch”) y 5.0 (“lenny”)
 - Fedora Core 4 hasta Fedora Core 11
 - Gentoo Linux
 - RedHat Enterprise Linux 4 y 5
 - Suse Linux 9 y 10 y OpenSuse 10.3, 11.0 y 11.11
 - Ubuntu 6.06 (“Dapper Drake”), 6.10 (“Edgy Eft”), 7.04 (“Feisty Fawn”), 7.10 (“Gutsy Gibbon”), 8.04 (“Hardy Heron”), 8.10 (“Intrepid Ibex”) y 9.04 (“Jaunty Jackalope”)
 - Mandriva 2007.1, 2008.0 y 2009.1
- Solaris Hosts (32-bit y 64-bit):
 - OpenSolaris (2008.05 y superior)
 - Solaris 10 (update 5 y superior)

3.1.3.2 Sistemas operativos “invitados” soportados

- Windows NT 4.0: Están soportados todas las versiones y service pack sin embargo existen algunos problemas con service pack muy antiguos. Se recomienda instalar el service pack 6a. VirtualBox provee guest additions.
- Windows 2000, XP, 2003 Server, Vista, 2008 Server y Windows 7: Todas las versiones y service pack están soportados, incluidas las versiones de 64-bit. También hay disponibles Guest Additions.
- Dos, Windows 3.x, Windows 95, Windows 98 y Windows Me: Hay pocos test realizados sobre estas versiones. No hay disponibles Guest Additions para estas versiones.
- Linux kernel 2.4: Soporte limitado

- Linux kernel 2.6: Todas las versiones están totalmente soportadas (tanto en 32-bit como en 64-bit). Hay disponibles Guest Additions. Es muy recomendable utilizar la versión de kernel 2.6.13 o superior para obtener el rendimiento más alto.
- Solaris10 y OpenSolaris: Soporte total tanto en 32-bit como en 64. Hay disponibles Guest Additions.
- FreeBSD: Requiere virtualizar hardware para soportar esta versión. No hay disponibles Guest Additions.
- OpenBSD: Requiere virtualizar hardware para estar soportado y sólo a partir de la versión 3.7.
- OS/2 Warp 4.5: Requiere virtualizar hardware para estar soportado. Tiene disponibles Guest Additions con unas características muy limitadas.

3.1.4 VirtualBox frente al resto: Ventajas e inconvenientes

Con respecto a los software de virtualización el mercado se ha decantado mayoritariamente por 3 fabricantes. El primero de ellos (y muy separado del resto) es EMC con Vmware. Después le sigue Microsoft, que se ha convertido en un serio competidor a través de su hipervisor Hyper-V. En tercer lugar aparece Citrix el cual tiene una base grande de clientes con la adquisición de XenSource.

VirtualBox parte de la desventaja de que a día de hoy no se ha aceptado su utilización en el mundo empresarial. Sin embargo la aceptación en ordenadores personales crece día a día, sobre todo para usuarios de Linux, que cada vez más eligen virtualbox como software de virtualización.

Virtualbox posee varias ventajas frente al resto de los software de virtualización. La primera y más importante es que es software open source. Económicamente es una característica a tener en cuenta, ya que el coste de las licencias de software suele ser bastante elevado.

Otra de las principales ventajas con las que cuenta es su amplio abanico de sistemas operativos que soporta (tanto host o “anfitrión” como guest o “invitado”). Esto implica que sea una aplicación muy potente en entornos que requieran integración entre diversidad de sistemas operativos.

En cuanto a rendimiento también se han realizado numerosas comparativas con respecto a Vmware que muestran resultados concluyentes:

- Consume muchos menos recursos, tanto de espacio en disco como de RAM
- No es necesario recompilar kernel para soportar nuevas características
- El arranque de las máquinas virtuales es mucho más rápido en VirtualBox
- Provee soporte Integrado para iSCSI
- Provee soporte integrado para RDP

3.1.5 Conclusiones

La capacidad de aprovechar al máximo el hardware disponible ofrece una gran cantidad de posibilidades a nivel empresarial y a nivel doméstico. Las ventajas de utilizar la tecnología de virtualización son muchas:

- Consolidación de servidores: convertir muchos servidores físicos en virtuales. De este modo se aprovecha el hardware disponible de la mejor manera posible.
- Recuperación ante desastres: las máquinas virtuales se pueden salvar muy fácilmente, y además su estado se puede almacenar, por lo que en caso de desastre se puede recuperar la información con rapidez.
- Pruebas de aplicaciones: en muchas ocasiones se necesita un entorno limpio para probar una aplicación. Usar una máquina virtual permite instalar un sistema operativo desde cero, probar la aplicación y luego eliminar la máquina.
- Ejecución de entornos completos sin instalación ni configuración: la posibilidad de descargar máquinas virtuales desde Internet permite ahorrar tiempo en instalaciones y configuraciones. Existen muchas máquinas virtuales con servidores LAMP (Linux, Apache, MySQL y PHP) completos listos para ser usados, máquinas con gestores de contenidos, wikis, etc., gratuitos y funcionales desde el primer momento.
- Aplicaciones portátiles: con el uso de las máquinas virtuales se pueden tener PCs completos listos para usar en dispositivos USB, lo que puede ser de mucha utilidad para tener un entorno privado y usarlo en cualquier PC.

3.2 Estudio del Sistema Operativo a utilizar

El sistema operativo es una de las partes más importantes en un equipo informático, ya que es el responsable de gestionar y coordinar los recursos hardware y el acceso a ellos. De él dependen entre otros, el rendimiento y seguridad de los equipos informáticos.

Hoy en día la cantidad de sistemas operativos que podemos encontrar es elevada por lo que la elección de un sistema u otro debe realizarse dependiendo de la función que vaya a desempeñar el equipo.

Para este proyecto nos centraremos más en el mundo Unix y Linux, ya que actualmente en el mundo empresarial la gran mayoría de servidores sobre los que corren aplicaciones críticas están gobernados por sistemas operativos Unix y/o Linux.

3.2.1 Estado actual del mundo “Linux”

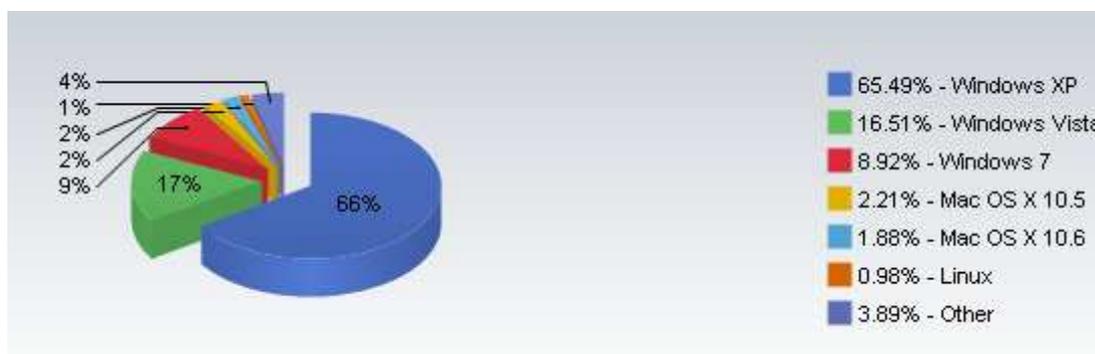
Linux (GNU/Linux) es el término empleado para referirse al sistema operativo libre similar a Unix. Es un sistema operativo desarrollado como “software libre”, es decir, todo el código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la licencia GPL (General Public License).

A pesar de que Linux es el núcleo del sistema operativo, la gran mayoría de herramientas que constituyen el sistema operativo son herramientas desarrolladas por el proyecto GNU.

A las variantes de esta unión de programas o herramientas y el núcleo, se les denominan distribuciones. El objetivo de estas distribuciones es generar sistemas operativos que cumplan con las necesidades de un determinado grupo de usuarios o unas determinadas especificaciones.

- En general, las distribuciones Linux pueden ser:
- Comerciales o no comerciales
- Ser completamente libres o incluir software privativo
- Diseñadas para uso empresarial o uso en el hogar
- Diseñadas para servidores, escritorios o dispositivos empotrados.

A pesar de todas las posibilidades que puede brindar utilizar un sistema operativo con kernel Linux, Windows sigue teniendo la mayoría de la cuota de mercado, tal y como podemos ver en el siguiente gráfico:



23. Cuota de mercado en SO.

En los últimos años este mercado está creciendo debido a las ventajas que ofrece la utilización de Linux en entornos empresariales:

- Estabilidad
- Seguridad
- independencia de proveedores
- rapidez de desarrollo en nuevas tecnologías

En la actualidad hay varias empresas que comercializan soluciones basadas en GNU/Linux: IBM, Novell (Suse), RedHat (RHEL), así como miles de pymes que ofrecen productos y servicios basados en esta tecnología.

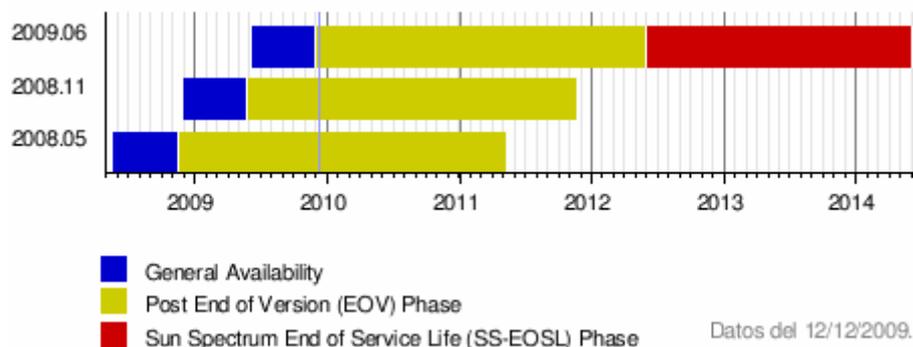
3.2.2 OpenSolaris: Estado Actual

OpenSolaris es un sistema operativo libre publicado en 2005 a partir de la versión privativa de Solaris de Sun Microsystems. Deriva del código base del System V (igual que Solaris) aunque gran parte ha sido modificado desde las versiones de inicio de la década de los 90.

Históricamente, la apertura del código fuente de Solaris comenzó a principios de 2004 cuando se formó un equipo multidisciplinario para considerar todos los aspectos del proyecto: la licencia, modelos de negocio, administración, co-desarrollo y análisis del

código fuente. Lo primero que se liberó fue d-trace, una herramienta utilizada por los administradores para depurar problemas y errores sistemáticos en el sistema operativo y sus aplicaciones.

Las versiones aparecidas según la línea de tiempos se pueden ver en la siguiente ilustración:



24. Cronología OpenSolaris.

En opensolaris, al igual que ocurre en el ámbito de Linux, también existen las distribuciones, algunas de ellas respaldadas por ingenieros de Sun y otras totalmente independientes:

- Belenix: Desarrollada por ingenieros de India Engineering Centre de Sun y mantenida por la comunidad OpenSolaris. Está orientada a entornos de escritorio y está orientada cubrir la demanda de usuarios de escritorio KDE.
- Nexenta OS: Es la primera distribución que combina las librerías GCC y herramientas GNU con el kernel SunOS de OpenSolaris.
- Opensolaris Release: Es la distribución apoyada oficialmente por Sun y que aglutina a toda la comunidad de desarrolladores de OpenSolaris. Es orientada a escritorio pero se puede adaptar fácilmente a servidores.
- Polaris: Es una versión experimental para PowerPC.

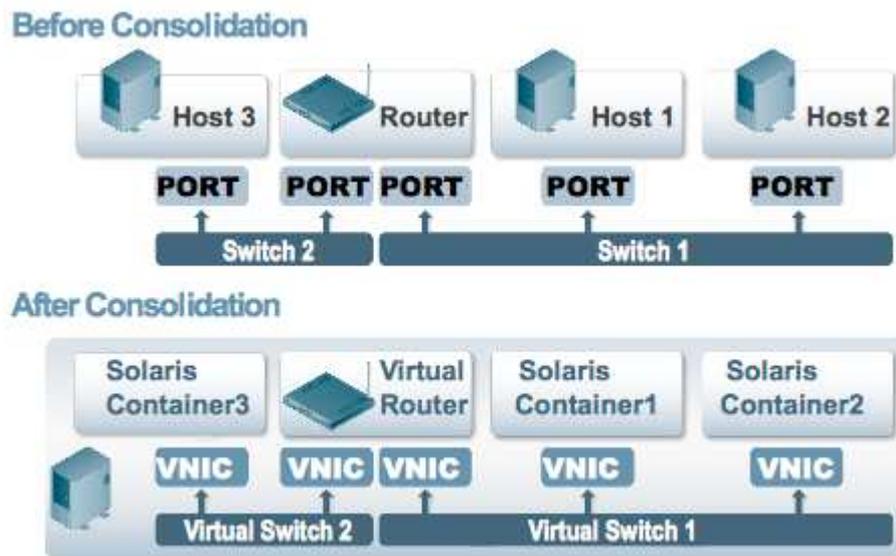
En la actualidad, tras la adquisición de Sun Microsystems por parte de Oracle existían dudas sobre su continuidad como software libre, aunque recientemente se ha anunciado que seguirá estando disponible bajo código abierto y además recibirá el respaldo para el desarrollo de nuevas versiones.

De momento la última release oficial es la 2009.06, aunque se espera una nueva release próxima que será 2010.3. Ésta release es la primera publicación oficial en llegar a plataformas SPARC.

Algunas de las novedades que aporta esta release son:

- Project Crossbow: Permite la virtualización de redes y gestión de recursos. Utilizando el bloque básico VNIC (Virtual Network Interface Controllers) es posible consolidar todo el entorno de computación para crear escenarios de certificación e incluso implantarlos. Las funciones de gestión incluidas en el proyecto Crossbow permiten fijar límites de ancho de banda en las NIC/VNIC, el establecimiento de prioridades de tráfico y la asignación de límites de recursos de CPU para las NIC/VNICs. Esta nueva arquitectura aporta nuevas

características que permiten trabajar eficazmente con las tarjetas de red de última generación manteniendo compatibilidad con NIC más antiguas. Entre estas características destaca la capacidad de cambiar en caliente de modo “interrupciones” a “polling mode” en altos volúmenes de tráfico.



25. Crossbow Project.

- Project Comstar: El framework ofrecido por el proyecto comstar permite convertir cualquier host Solaris en un target SCSI. Con este software es posible conectar un host a cualquier dispositivo SCSI a través de una red de transporte IP con acceso concurrente a las LUN y un punto único de gestión.

3.2.3 OpenSolaris: Ventajas e inconvenientes

Una de las principales ventajas que aporta la utilización de opensolaris es la facilidad para portar cualquier implementación a servidores de producción que corren solaris10 ya que posee las características del mismo. La próxima release de Solaris estará basada en Opensolaris.

Además podemos utilizar las herramientas avanzadas que contiene Solaris 10, como por ejemplo ZFS, D-trace y containers (o zonas).

La desventaja más importante con la que cuenta aún opensolaris es que puede no ser compatible con algún tipo de hardware en ordenadores domésticos.

3.2.4 Requisitos de instalación

Procesador compatible x86 (Intel y AMD)

Compatibilidad con procesadores x64

Al menos 512 MB de Ram, aunque se recomienda 768 MB

3 GB de espacio en disco aunque la recomendación son 7GB

La lista de hardware compatible es muy extensa y puede ser consultada en el siguiente vínculo:

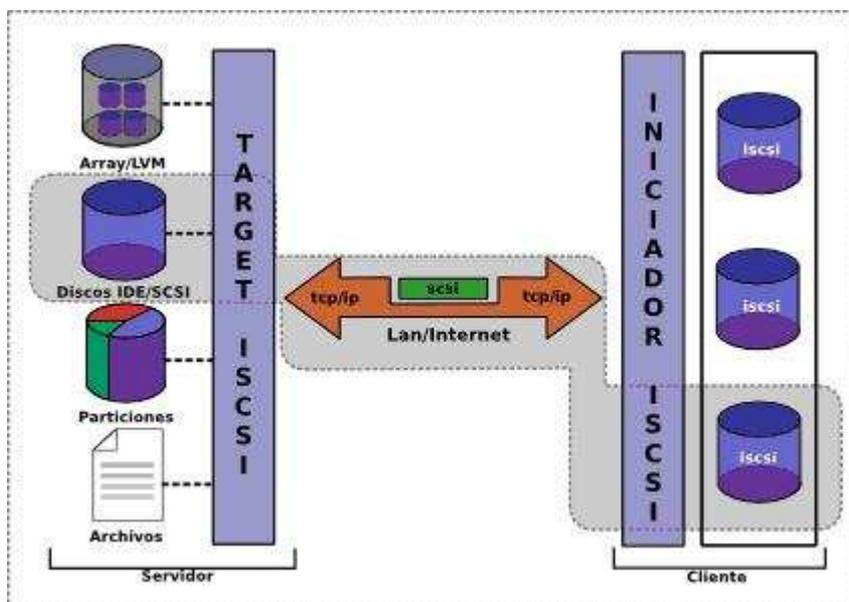
3.2.5 Conclusiones

Dentro del mundo “open source” podemos encontrar muchos proyectos abiertos de los cuales algunos se convertirán en herramientas utilizadas en el futuro y otros ni siquiera llegarán a terminar. Con los sistemas operativos denominados “libres” a día de hoy hay cientos de posibilidades entre las que elegir aunque sólo algunas son realmente eficaces y están probadas como para ser utilizadas en entornos de producción. Dentro de estos sistemas operativos está incluido opensolaris, el cual es una buena opción ya que permite la utilización de características avanzadas (por ejemplo ZFS, D-trace) y aporta el valor añadido de que sus release para producción son réplicas de las versiones de la comunidad, y por lo tanto es posible realizar una implementación de un proyecto sobre opensolaris y fácilmente portarlo a cualquier servidor corriendo Solaris 10.

3.3 Estudio del software para una solución iscsi

iSCSi es una extensión de SCSI para conectar los dispositivos utilizando la red TCP/IP, importando todo el dispositivo hardware en la parte cliente de manera que este lo ve como un dispositivo SCSI más. La utilización de iSCSI se utiliza para centralizar el almacenamiento en disco e independizar la información de los servidores.

En el siguiente diagrama se puede ver la arquitectura iSCSI que es de tipo cliente/servidor.



26. iSCSi Cliente/Servidor.

La parte del servidor se denomina target iSCSI y hay diferentes herramientas que permiten realizar este cometido.

3.3.1 Introducción a los target iSCSI

El target iSCSI es la parte servidor. Un target puede ofrecer uno o más recursos iSCSI por la red. En las soluciones Unix para iSCSI, no hace falta que el dispositivo a exportar

sea necesariamente un disco SCSI; medios de almacenamiento de distinta naturaleza se pueden usar, como por ejemplo:

- Particiones RAID
- Particiones LVM
- Discos enteros
- Particiones comunes
- Archivos
- Dispositivos de CD, cintas, etc.

3.3.2 Proyectos Open Source para implementar target iSCSI

Para poder implementar target iscsi existen numerosas posibilidades entre las cuales elegir, dependiendo del sistema operativo servidor (Windows, Linux, Unix). En este proyecto nos centraremos en ver las posibilidades existentes tanto en el mundo Linux como en Unix.

3.3.2.1 iSCSI Enterprise Target (IET)

iSCSI Enterprise Target es un software desarrollado para implementar un sistema de almacenamiento iSCSI en Linux.

- Algunas de sus características son:
- No necesita instalar parches de kernel
- Soporta SMP (Symmetric Multi-Processing).
- Soporta Kernel 2.6.x
- Se pueden añadir y borrar dinámicamente targets, volúmenes y cuentas para autenticación

Este proyecto aún se mantiene y la última release lanzada es la 1.4.19 (15/11/2009).

3.3.2.2 Open-iSCSI

Open-iSCSI es un proyecto lanzado para la implementación multiplataforma de la RFC3720 y alto rendimiento en las tasas de transferencia.

Las características que aporta son:

- Alto rendimiento: 450MB/s en lectura y 450MB/s en escritura para un tamaño de bloque de 64KB.
- Mínimo consumo de Cpu en modo kernel
- Mayor utilización del espacio de usuario

La desventaja que tiene este proyecto es que aún no existe una versión estable. Solo existe la versión “semi-estable” 2.0-871

3.3.2.3 SCST (Scsi target subsystem for Linux)

SCST es una implementación de un subsistema target para Linux. Proporciona una interfaz unificada y consistente entre los driver scsi y el kernel así como entre el kernel y los controladores del backend de almacenamiento.

Además, permite la creación de dispositivos de almacenamiento avanzados que pueden proporcionar características como replicación, deduplicación, alta disponibilidad y copias de seguridad automática. Además puede generar dispositivos virtuales como librerías de cintas virtuales (VTL). Permite utilizar cualquier tipo de enlace que soporte SCSI, incluyendo Fibre Channel, iSCSI, SAS, Infiniband y Wide SCSI.

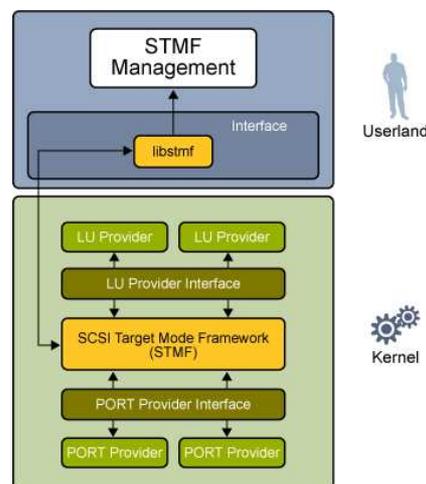
3.3.2.4 Comstar (OpenSolaris Common Multiprotocol SCSI Target)

Comstar es un software que permite convertir cualquier host opensolaris en un target SCSI que puede ser accedido sobre una red TCP/IP por cualquier iniciador.

Las características más importantes de Comstar son:

- Integración con OpenSolaris: Comstar aprovecha el conjunto de API's de OpenSolaris y las plataformas soportadas
- Basado en estándar: Comstar sigue las recomendaciones T10 para las interfaces SCSI. Soporta Target Port Group Support (TPGS) para asegurar capacidad multipathing.
- Mapeo de LUN: Permite asignar permisos a cada LUN mediante la definición de hosts y target groups.
- Incremento del rendimiento: Comstar utiliza transferencias múltiples en paralelo para los comandos SCSI mejorando el rendimiento general.
- Provisión On-Demand: Permite aumentar y disminuir el espacio de almacenamiento dinámicamente según las necesidades, con un tamaño máximo de 8 exabytes (8192 petabytes).

Comstar facilita la administración de los target a través de módulos funcionales independientes que se agrupan en un Framework conocido como STMF (Scsi Target Mode Framework).

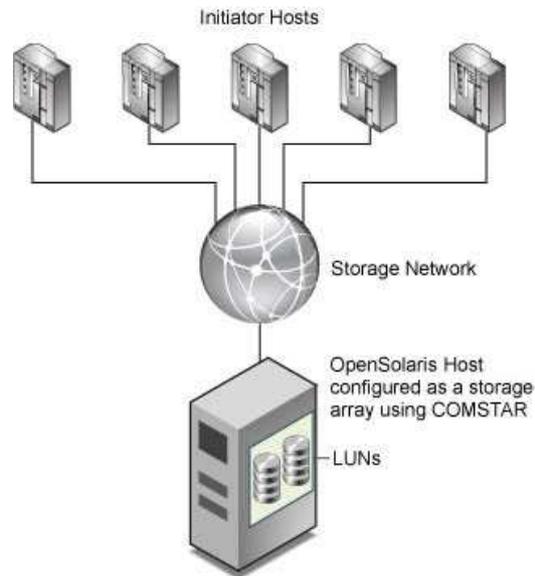


27. Diagrama bloques.

Además soporta host initiator de las siguientes plataformas:

- Solaris 10
- Windows
- Linux
- Vmware ESX

En la siguiente ilustración el host OpenSolaris está configurado como un target SCSI utilizando Comstar.



28. OpenSolaris Target..

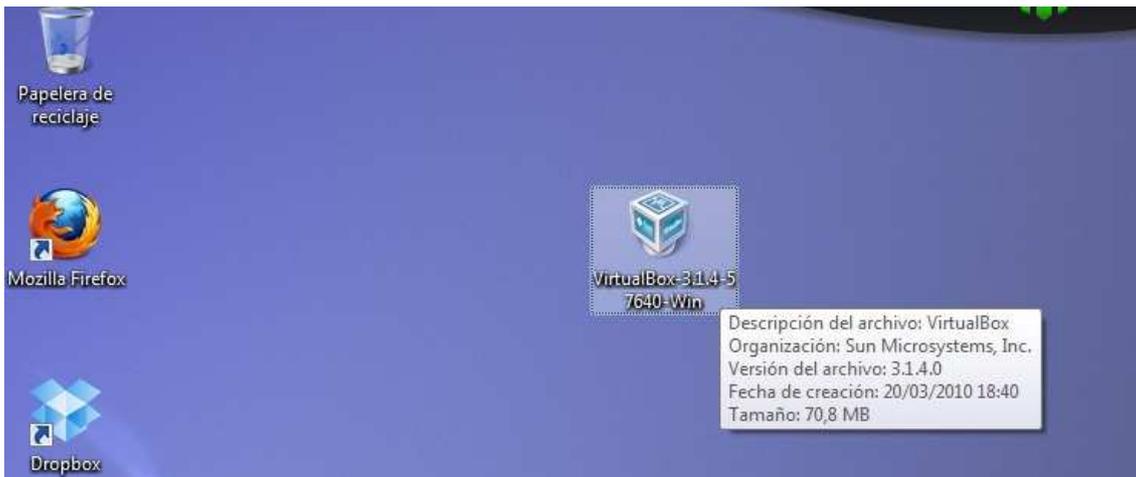
3.4 Instalación del software de Base

La plataforma sobre la cual instalamos el software de base tiene las siguientes características:

- Sistema Operativo: Windows 7 Home Premium
- Procesador: Intel Core 2 Duo T6600 2,2 Ghz
- Memoria Ram: 4 GB

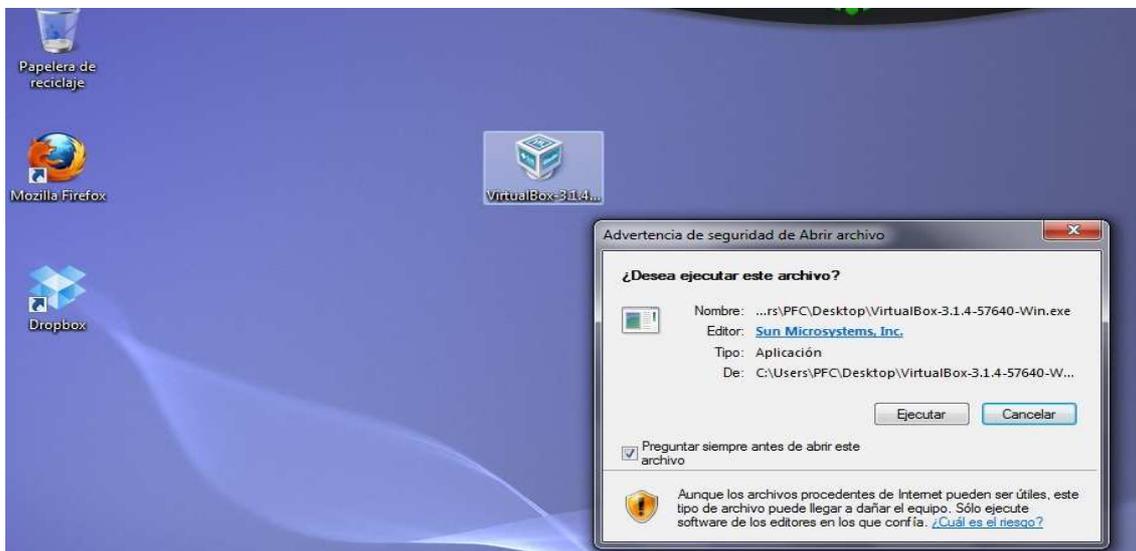
3.4.1 Instalación de VirtualBox

Primero descargamos la última versión de VirtualBox (actualmente es la release 3.1.4 lanzada el 12 de Febrero de 2010) para hosts Windows. Una vez tengamos el archivo instalador lo ejecutamos.



29. Archivo Ejecutable.

Aparecerá la advertencia de seguridad, en la cual pincharemos en ejecutar.



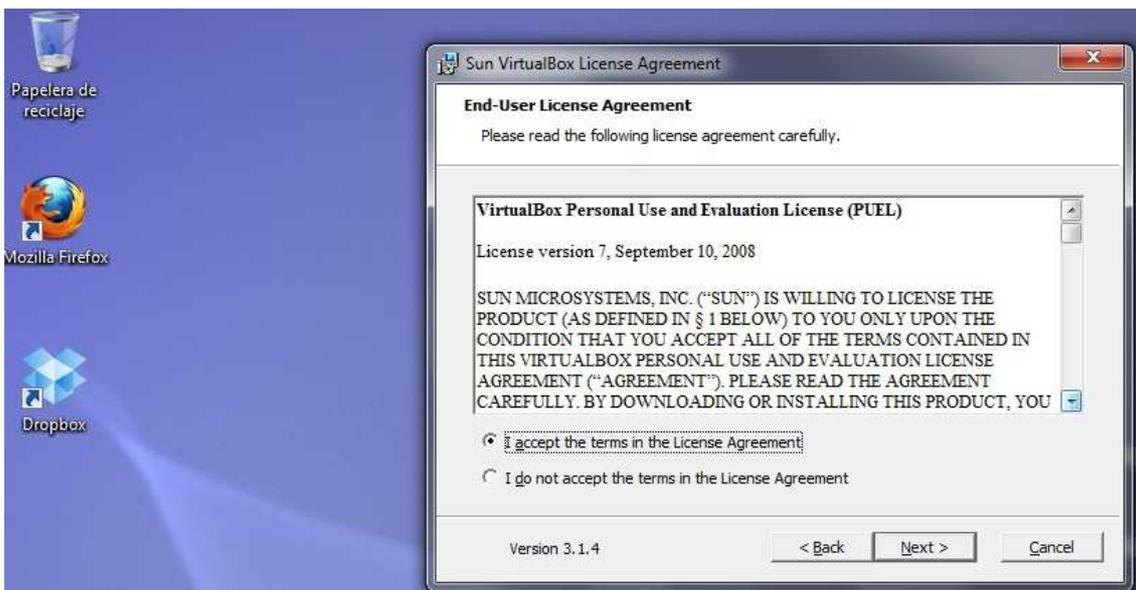
30. Advertencia Seguridad.

En la siguiente pantalla el instalador nos dará la bienvenida. Pinchamos en el botón Next.



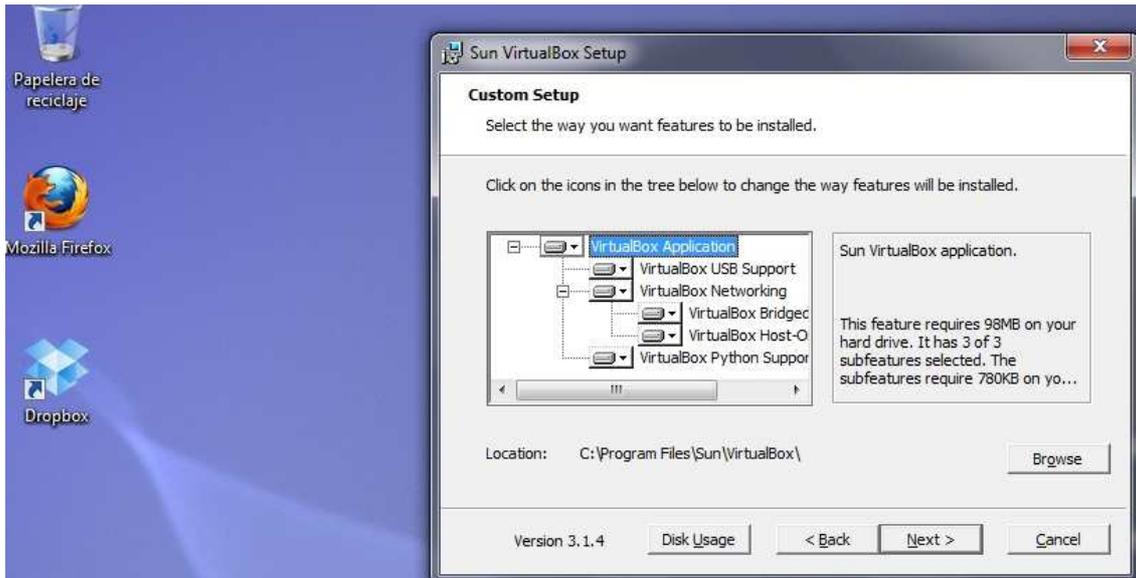
31. Bienvenida.

A continuación aparecerán los términos de licencia, los cuales debemos aceptar para proseguir con la instalación y pinchar en el botón Next.



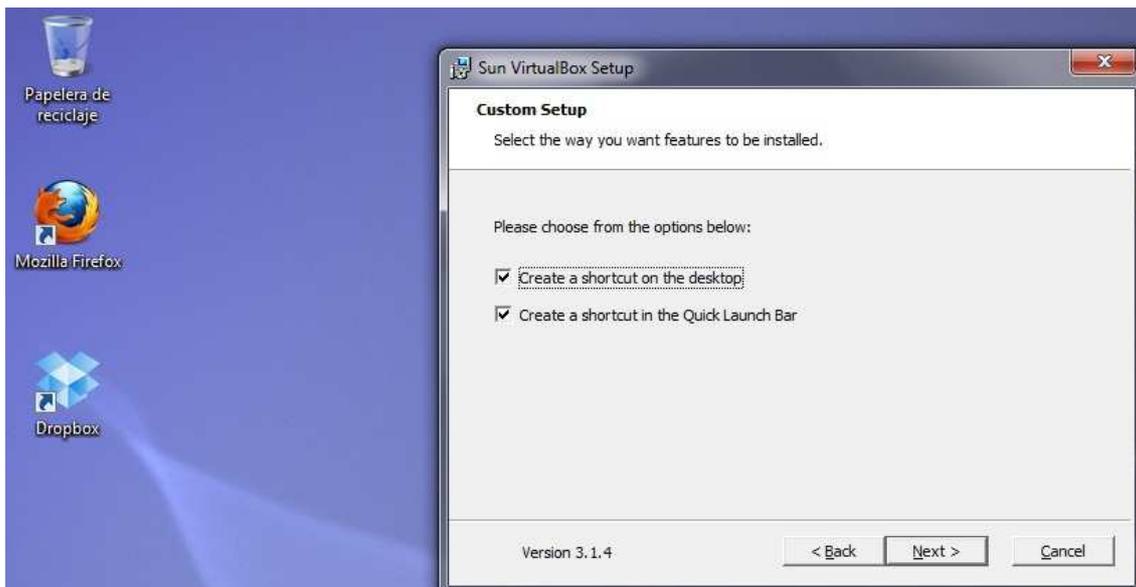
32. Licencia.

La siguiente ventana nos permite indicar qué elementos queremos instalar en este momento y cuales no. Por defecto está marcado para instalar todo. Dejamos la instalación por defecto y pinchamos en Next.



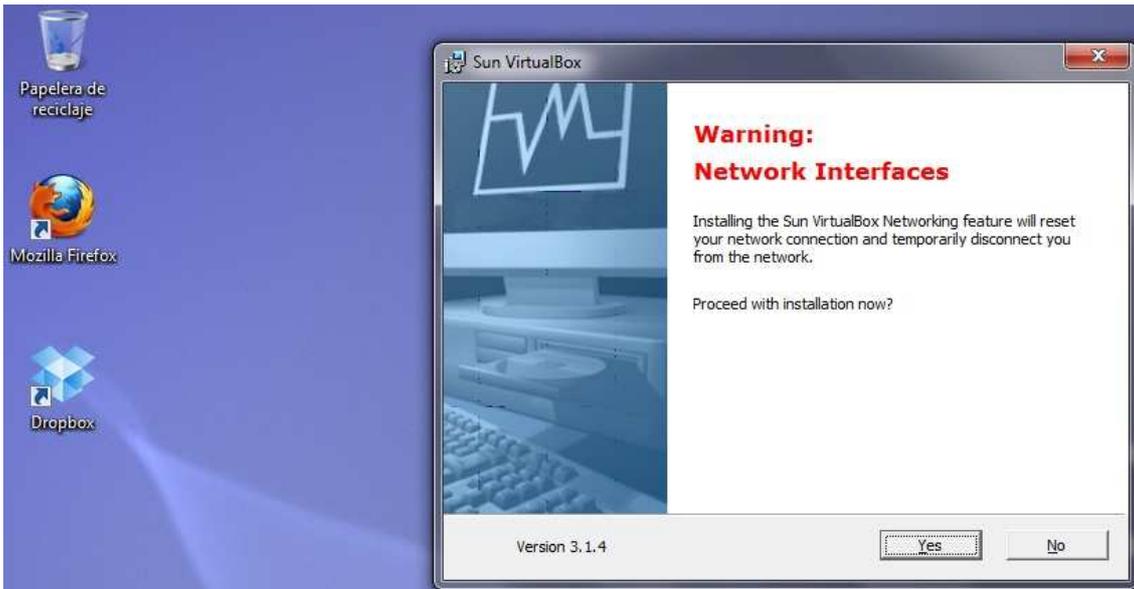
33. Instalador. Componentes.

La siguiente pantalla nos permite elegir si queremos accesos directos en el escritorio y en la barra de inicio rápido. Lo dejamos por defecto y pinchamos Next.



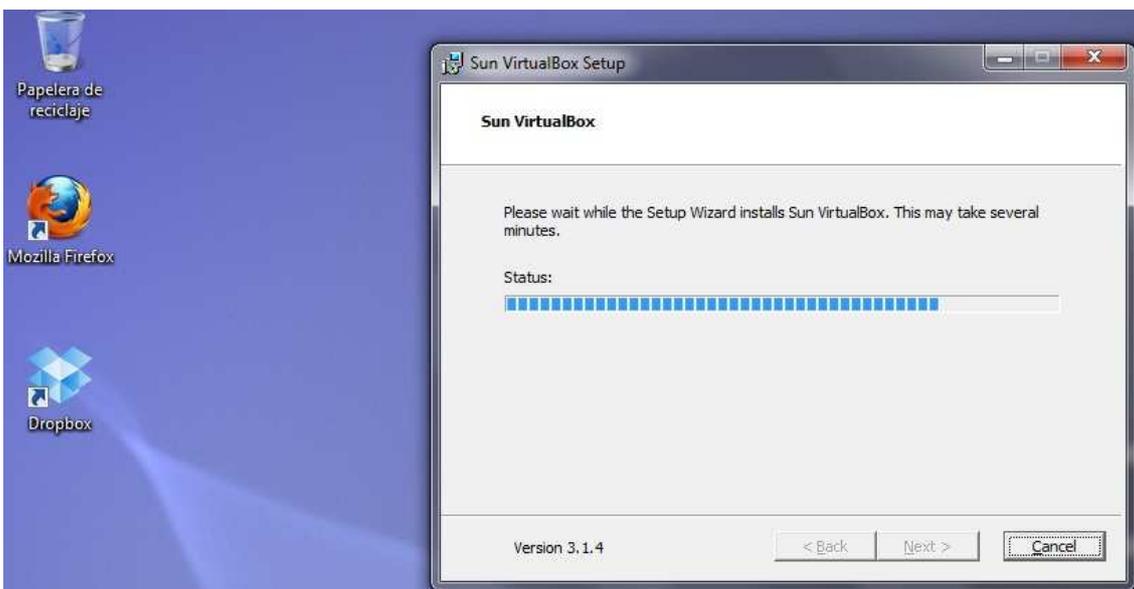
34. Instalador. Opciones escritorio.

Por último, la siguiente ventana nos avisa de que perderemos la conexión de red temporalmente, ya que virtualbox necesita configurar los interfaces de red virtuales. Pinchamos en Yes para continuar la instalación y en la siguiente ventana, pulsamos Install.



35. Advertencia interfaces red.

La instalación comenzará y nos mostrará una barra con el estado de la misma.



36. Progreso.

Una vez termine la instalación correctamente el instalador nos mostrará la ventana final en la cual debemos pulsar finish.



37. Instalación finalizada.

En este momento la instalación de virtualbox ha concluido y podemos comenzar a crear nuestra máquina virtual para instalar OpenSolaris.

3.4.1.1 Creación de máquina virtual en VirtualBox

Para crear la máquina virtual podemos pinchar en el acceso directo que el instalador de virtualbox ha dejado en el escritorio.

La primera vez que se inicia virtualbox aparecerá una ventana de registro en la cual se brinda la posibilidad de registrarse para recibir alertas de noticias y actualizaciones sobre virtualbox y para que la comunidad que desarrolla virtualbox pueda recibir estadísticas sobre el uso que ayuden a mejorar el software. Si no se desea registrarse basta con pulsar cancelar.



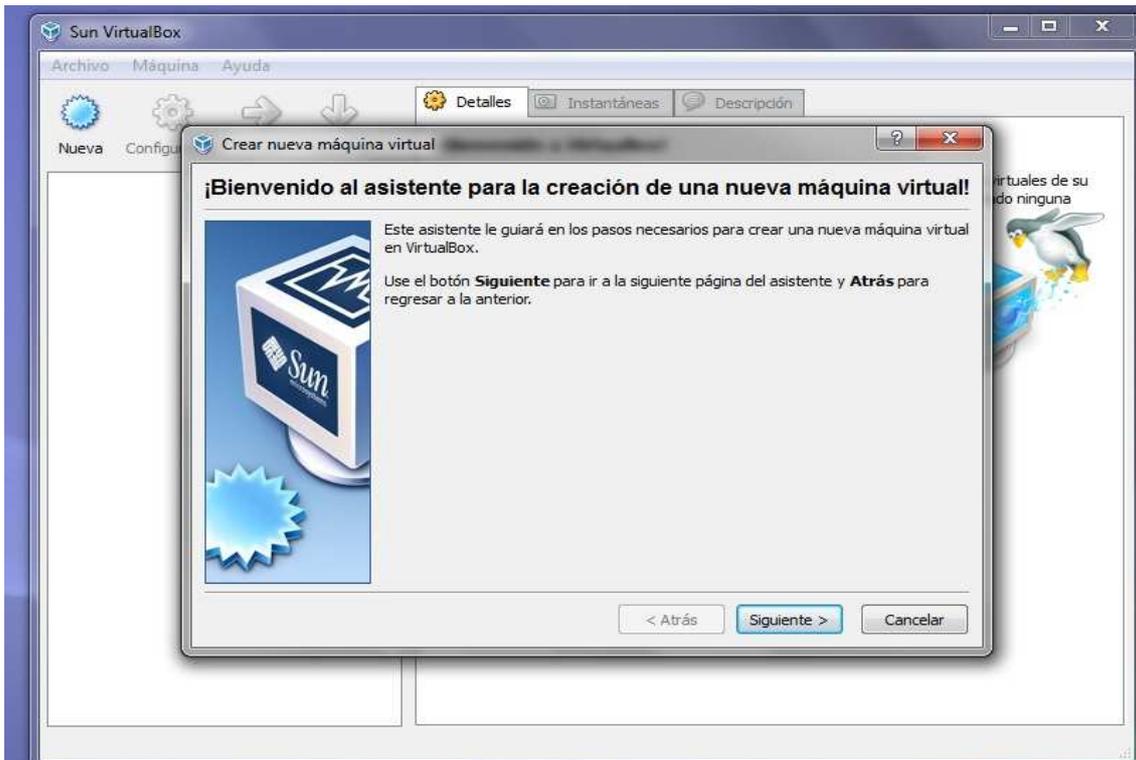
38. VB Registro.

A continuación se presentará la interfaz principal de VirtualBox. Es una interfaz muy intuitiva y sencilla y nos permite crear una máquina virtual con mucha facilidad. Pinchamos en el botón “Nueva”.



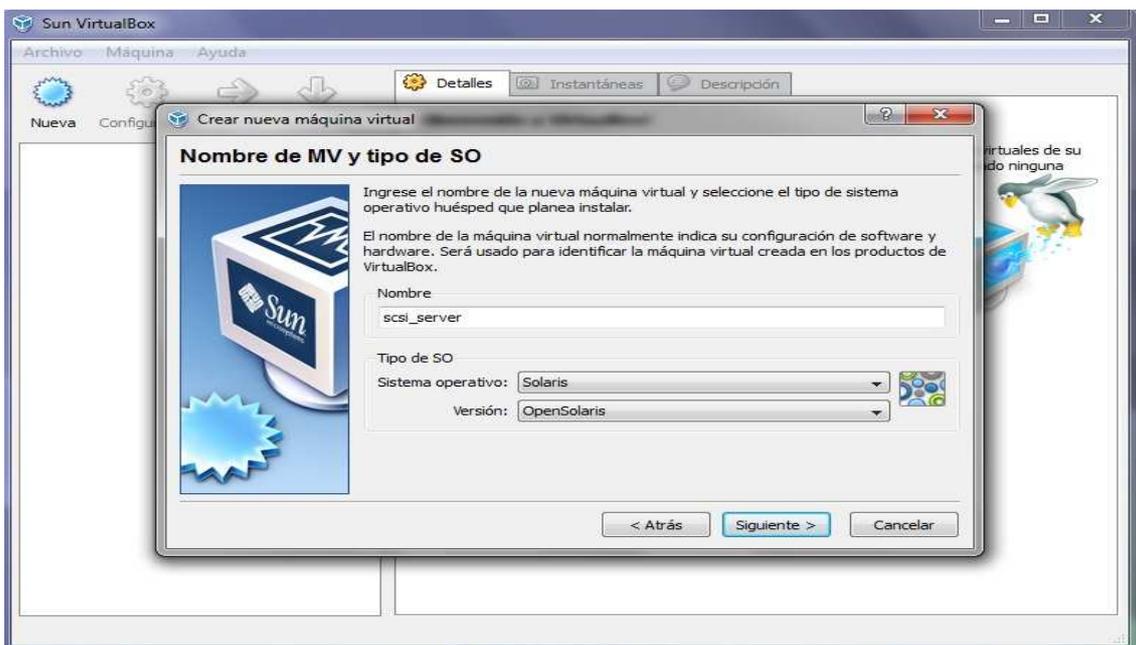
39. VB Inicio.

Al pinchar en el botón “Nueva” aparecerá el asistente para crear una nueva máquina virtual. Pinchamos en siguiente.



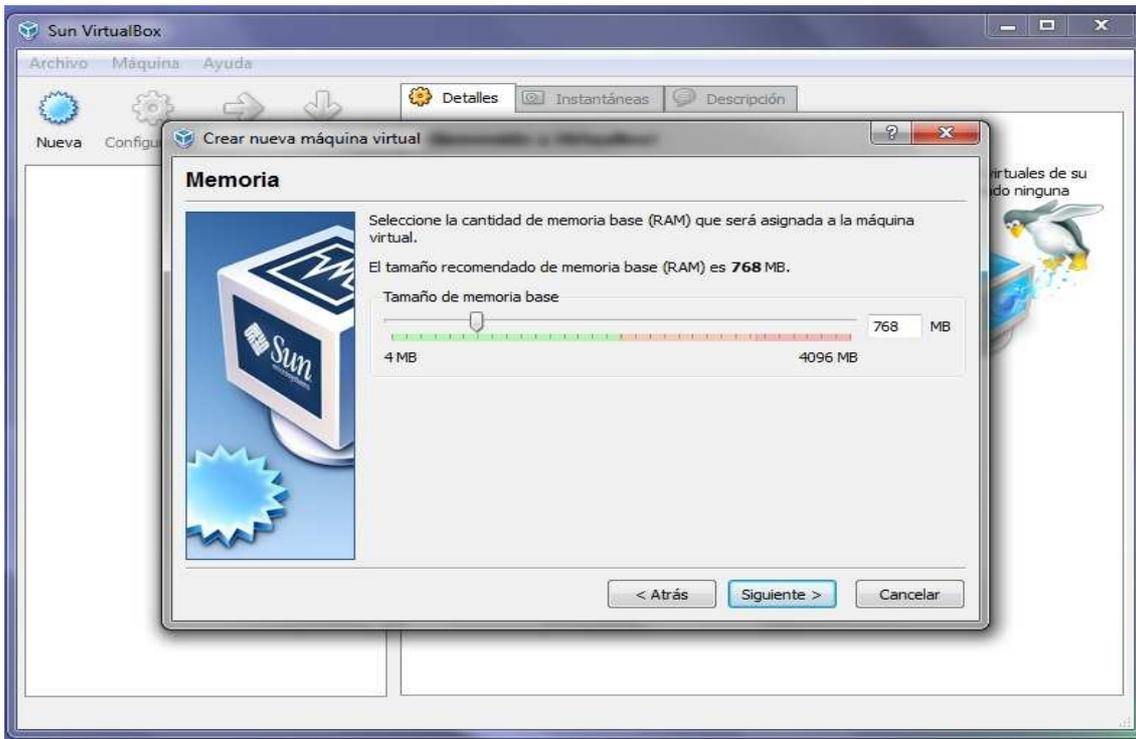
40. VB Nueva.

La siguiente pantalla nos pide el nombre con el que identificar la máquina virtual. Además con el objeto de añadir características específicas para cada sistema operativo “guest” debemos identificar qué sistema operativo correrá sobre la máquina virtual. Una vez estén rellenos todos los campos del asistente, pulsamos en siguiente.



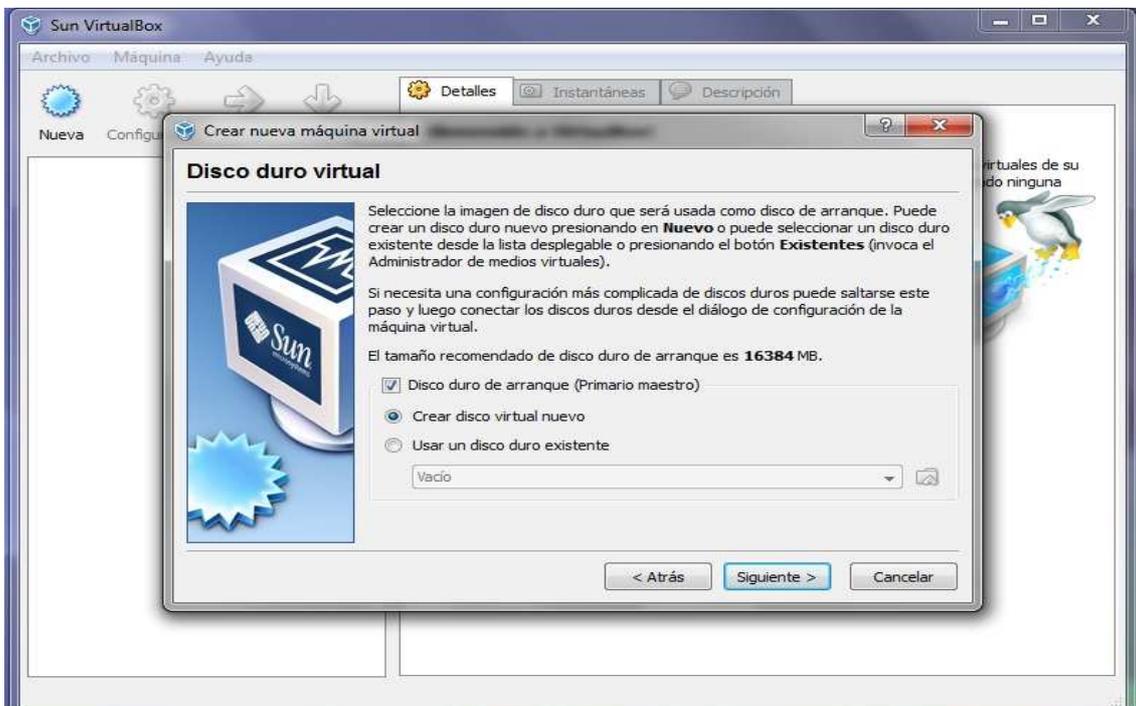
41. VB Identificación Máquina.

En la siguiente ventana del asistente, seleccionaremos la cantidad de memoria Ram que asignaremos a la máquina virtual.



42. VB Ram.

En la siguiente ventana se da la opción de crear un nuevo disco duro virtual o añadir uno ya existente. Esto es útil si tenemos varias máquinas virtuales creadas en nuestro servidor. En este caso le indicamos que vamos a crear un disco duro virtual nuevo.



43. VB Disco Duro.

En la siguiente ventana podemos seleccionar como queremos que virtualbox gestione el espacio del disco duro virtual. Podemos elegir 2 opciones:

- Almacenamiento de expansión dinámica: Asignamos una cantidad de almacenamiento (por ejemplo 16GB) pero virtualbox no coge los 16 GB del disco físico, sino que va creciendo según las necesidades hasta 16GB. Esta opción es útil cuando queremos reservar una cantidad de almacenamiento pero a priori no se conoce si va a ser utilizado el total.
- Almacenamiento de tamaño fijo: Con esta opción virtualbox reserva el espacio que seleccionemos del disco duro físico (aunque no se utilice todo).

Para el disco duro donde instalaremos el Sistema Operativo vamos a utilizar expansión dinámica.



44. VB Tipo Almacenamiento.

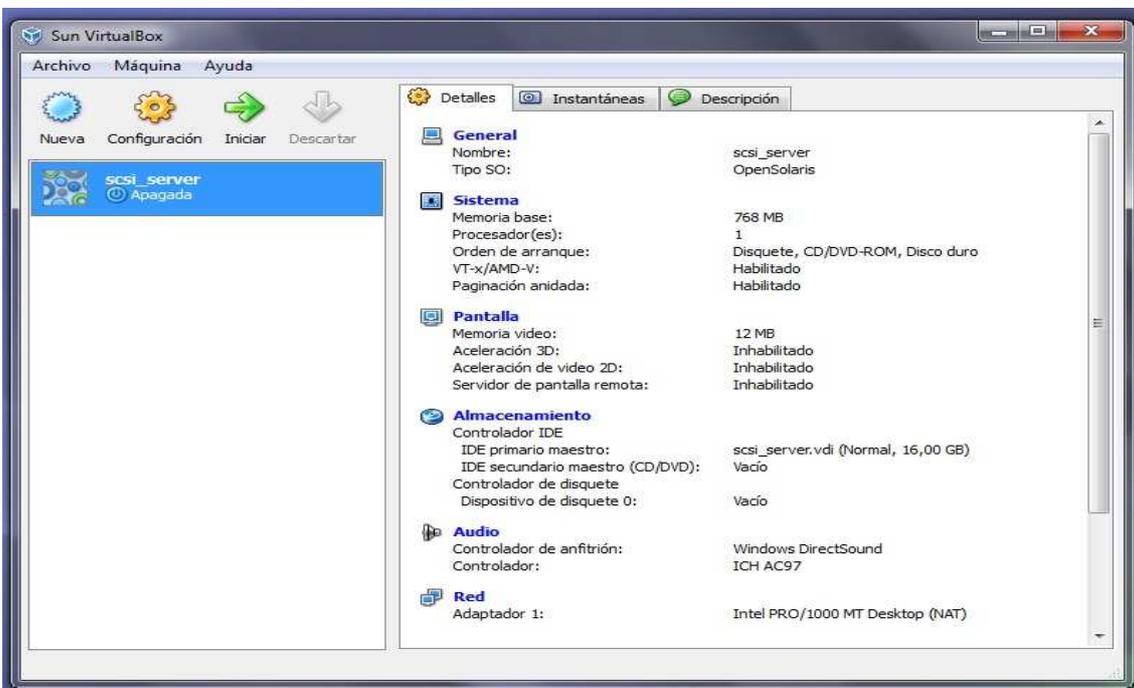
En la siguiente ventana asignamos el nombre con el cual identificaremos el disco virtual dentro de virtualbox así como el tamaño máximo asignado al mismo.



45. VB Tamaño Disco.

A continuación nos aparecerán 2 ventanas de confirmación. Pulsamos Terminar en ambas. Con estos pasos la configuración básica de la máquina virtual ha finalizado.

La máquina virtual creada aparece en la interfaz principal desde la cual podemos además modificar cualquier parámetro. Algunos se pueden cambiar en caliente (con la máquina virtual arrancada) y otros sólo son modificables en frío (es necesario parar la máquina virtual para cambiarlos).

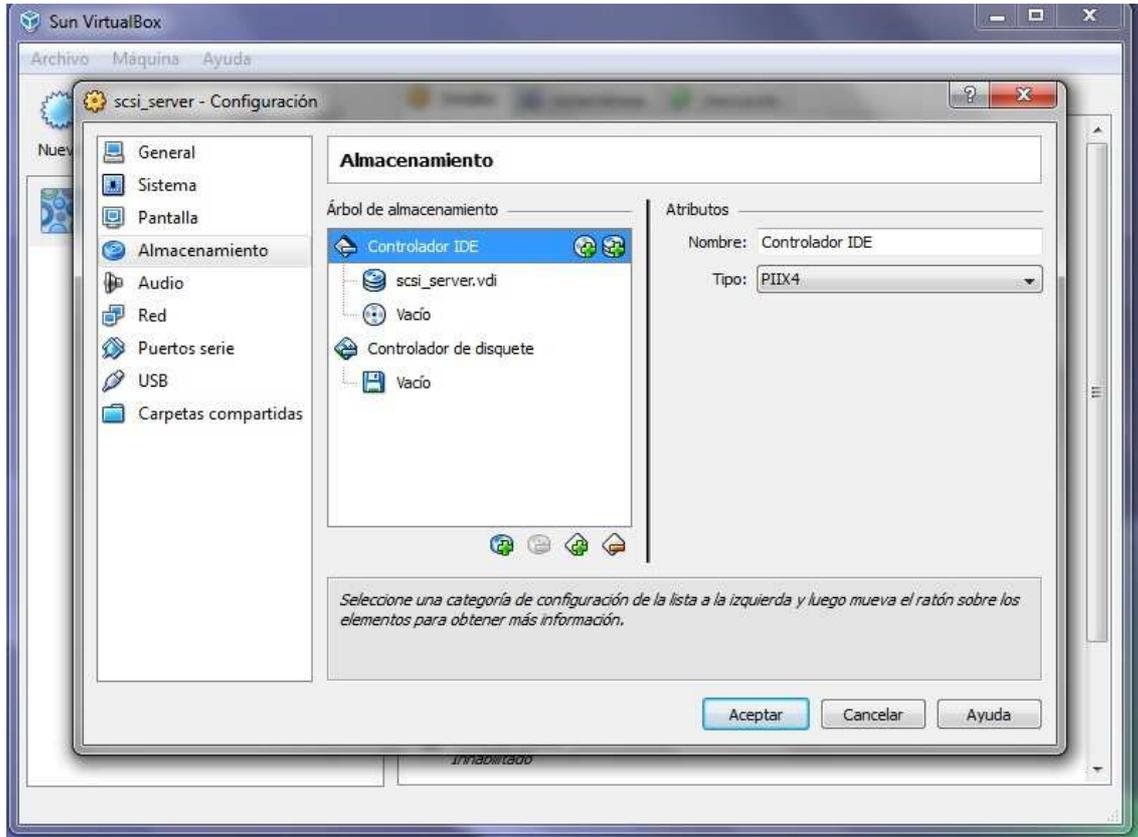


46. VB Presentacion VM.

3.4.2 Instalación de OpenSolaris

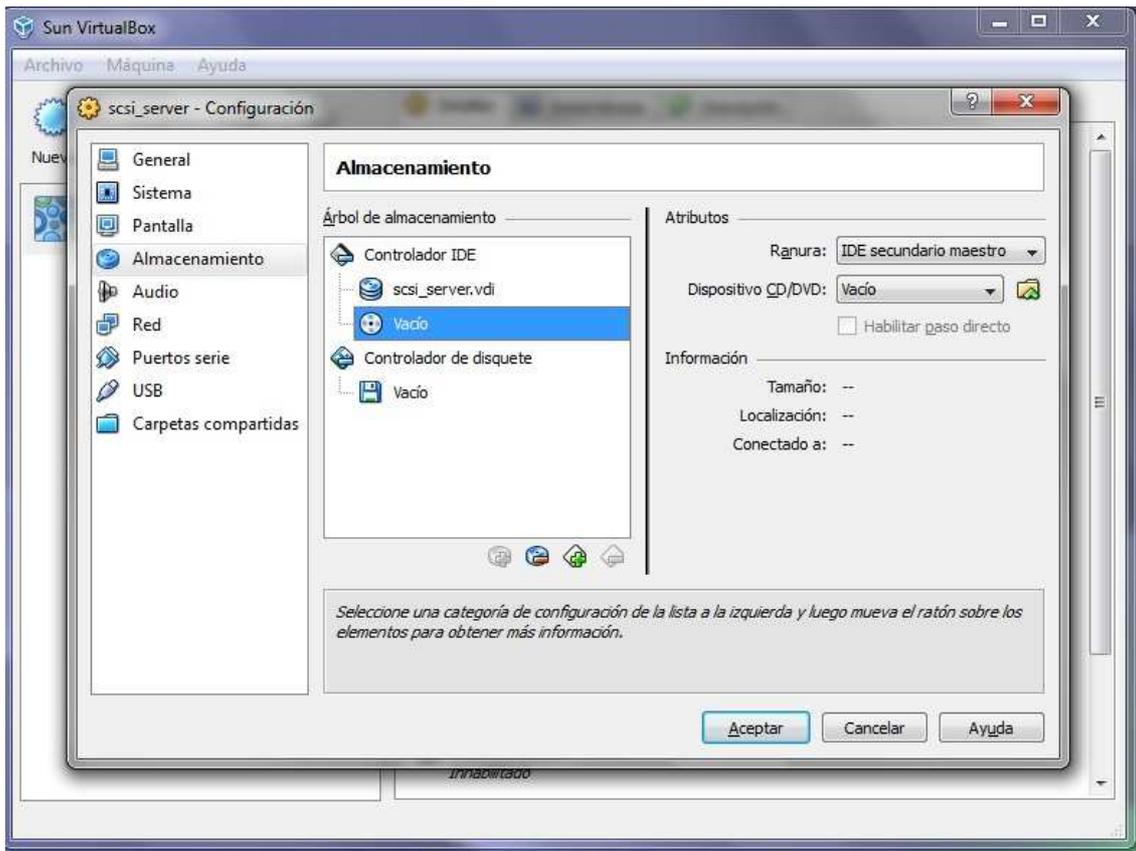
Para realizar la instalación de opensolaris es necesario acceder a la web <http://hub.opensolaris.org> y descargar la versión requerida. Para la realización de este proyecto se ha trabajado con la versión Express Community build 130.

Una vez descargada hay que indicar a virtualbox que la utilice como medio de arranque de la máquina virtual. Para ello se da doble click en la ventana de presentación en la parte de almacenamiento y aparece una ventana como la de la siguiente ilustración.



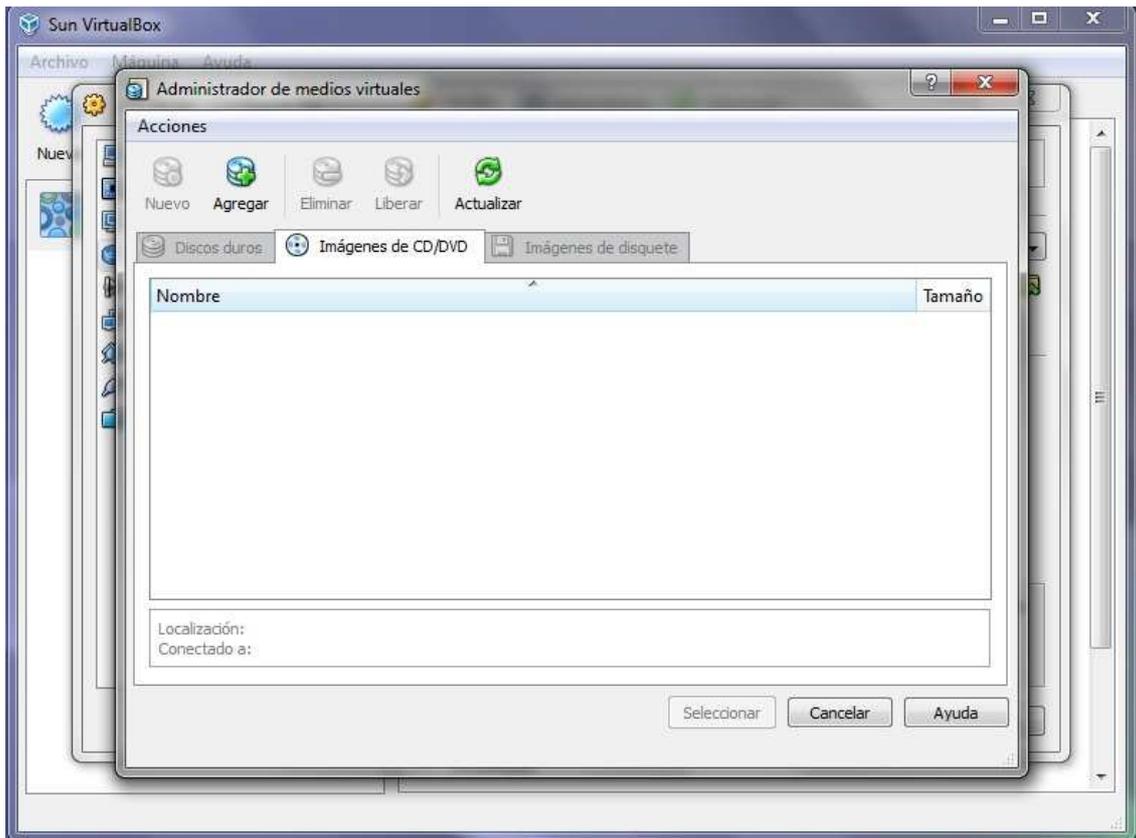
47. VB Almacenamiento.

Esta ventana muestra los dispositivos virtuales de almacenamiento que tiene configurados la máquina virtual. Para instalar opensolaris necesitamos asignar la imagen descargada al CDROM. Para ello marcamos el icono del CD en la ventana “árbol de almacenamiento” y nos aparecerá el siguiente menú a la derecha de la pantalla:



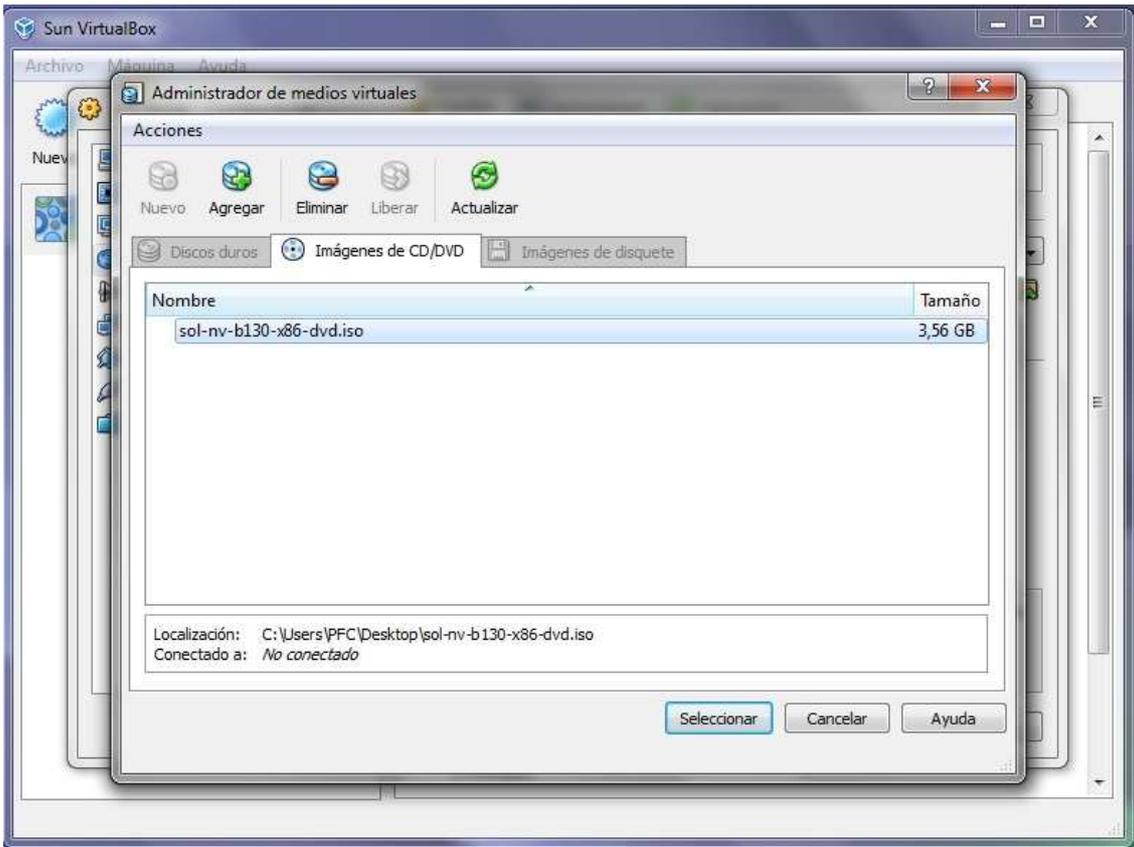
48. VB Añadir DVD.

Ahora pinchamos en la carpeta de la derecha de la opción Dispositivo CD/DVD que marca “vacío” y se abrirá la ventana de gestión de medios virtuales.



49. VB Añadir ISO.

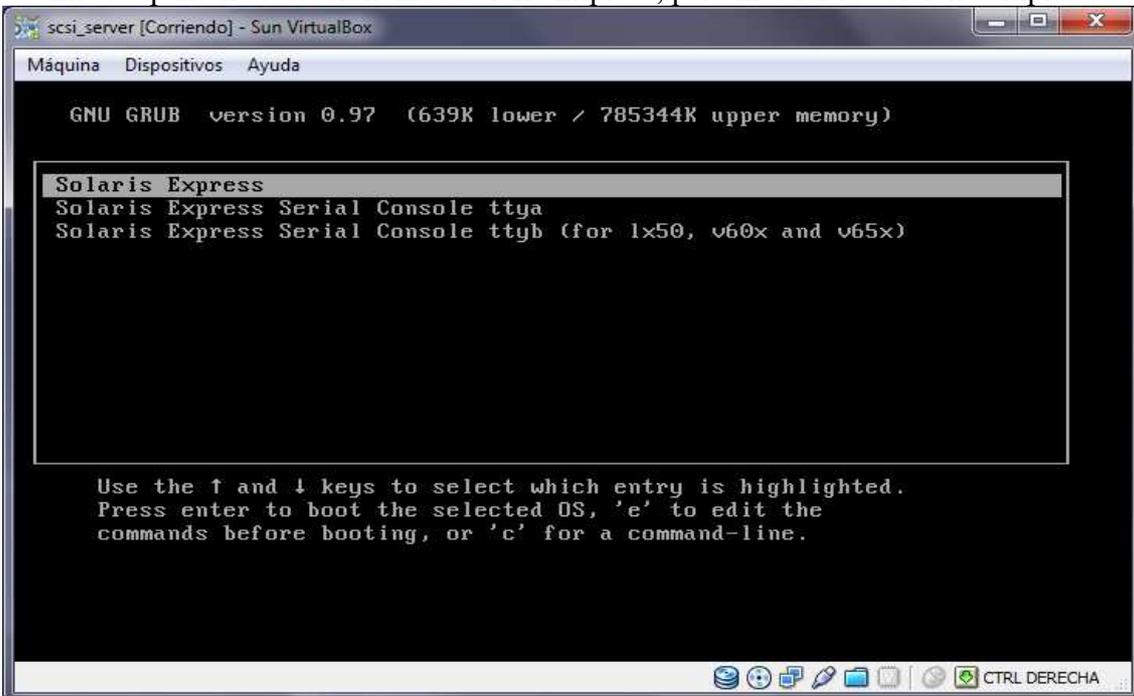
En esta ventana pinchamos en agregar y se selecciona la imagen descargada de opensolaris.



50. VB ISO Solaris.

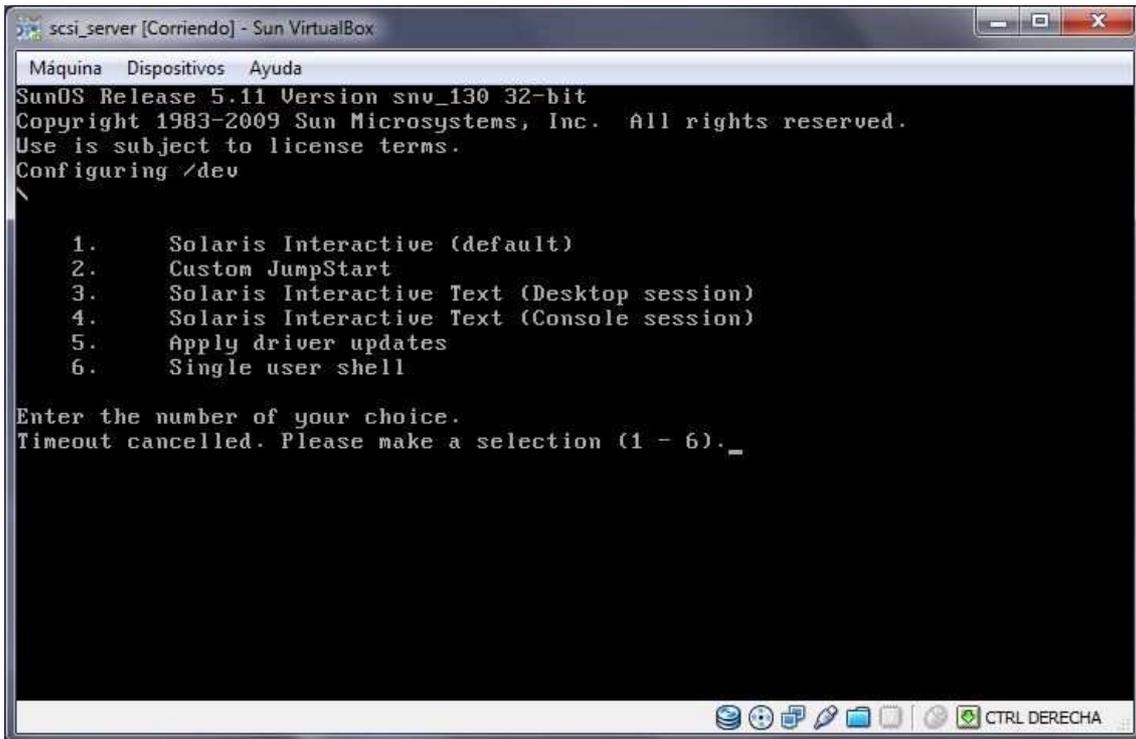
Una vez realizada la asignación de la imagen, se procede a arrancar la máquina virtual desde la ventana principal de virtualbox.

En esta pantalla seleccionamos Solaris Express, para instalar la versión completa.



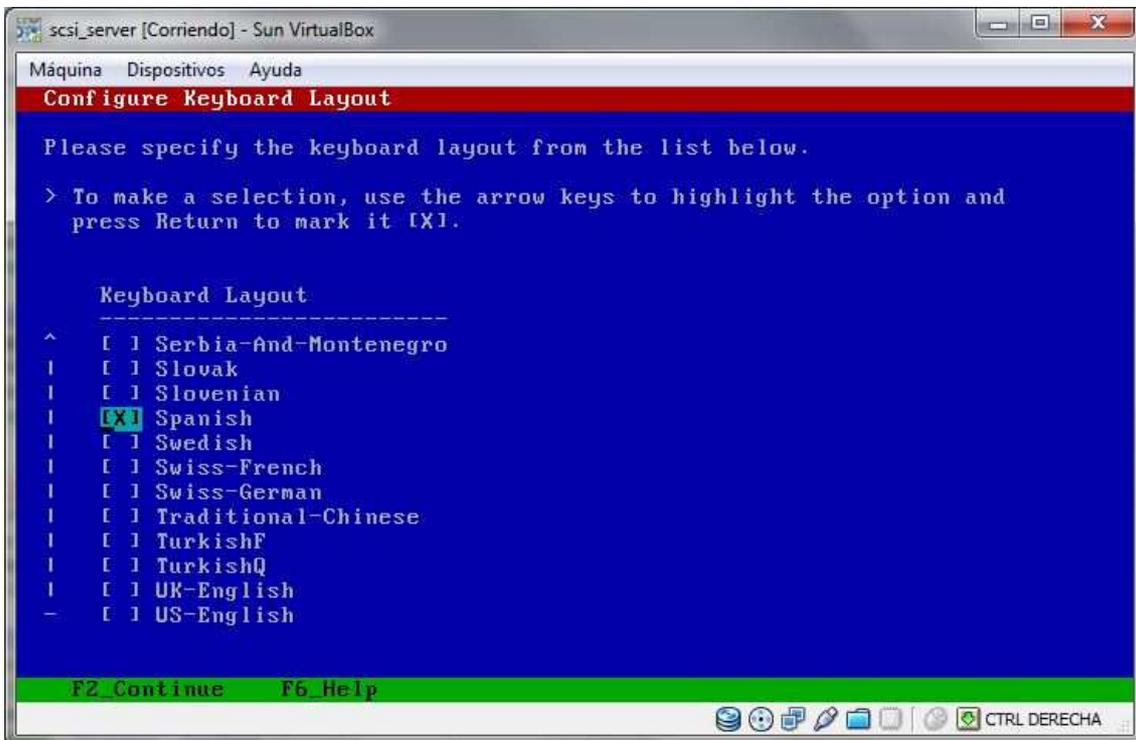
51. Solaris Instalación 1.

Ahora configurará los dispositivos y en unos minutos se presentará la siguiente pantalla en la cual se selecciona la opción 1.



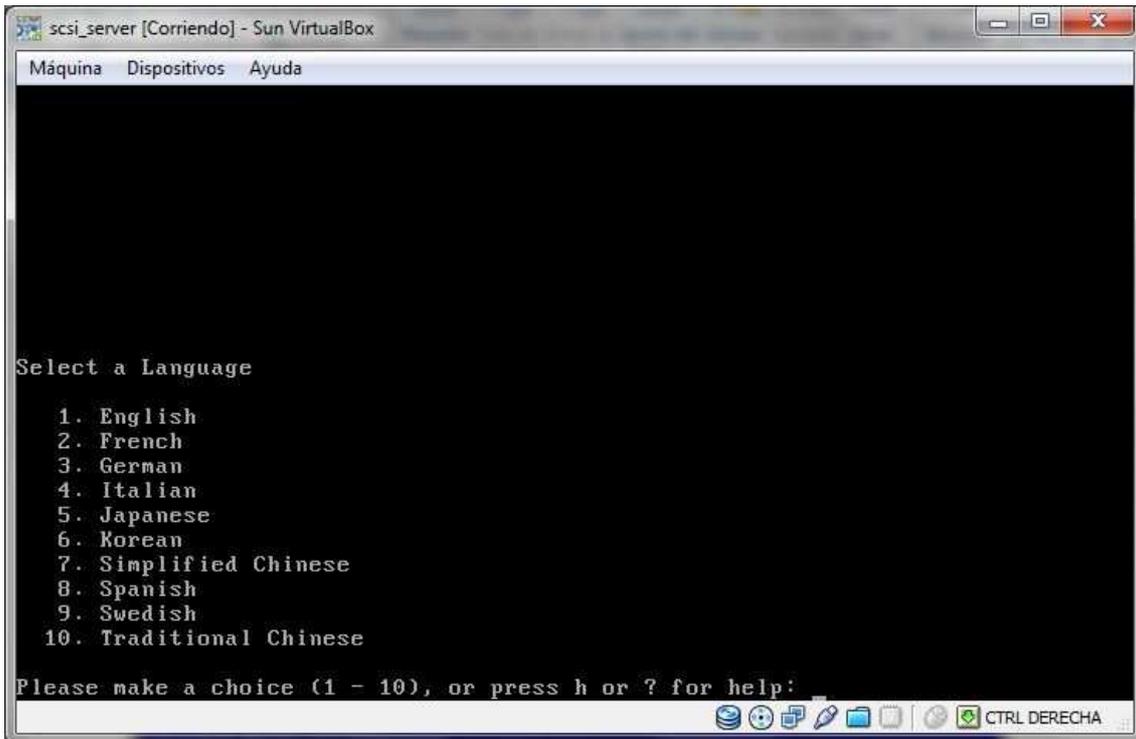
52. Solaris Instalación 2.

En la siguiente ventana es necesario elegir el tipo de teclado.



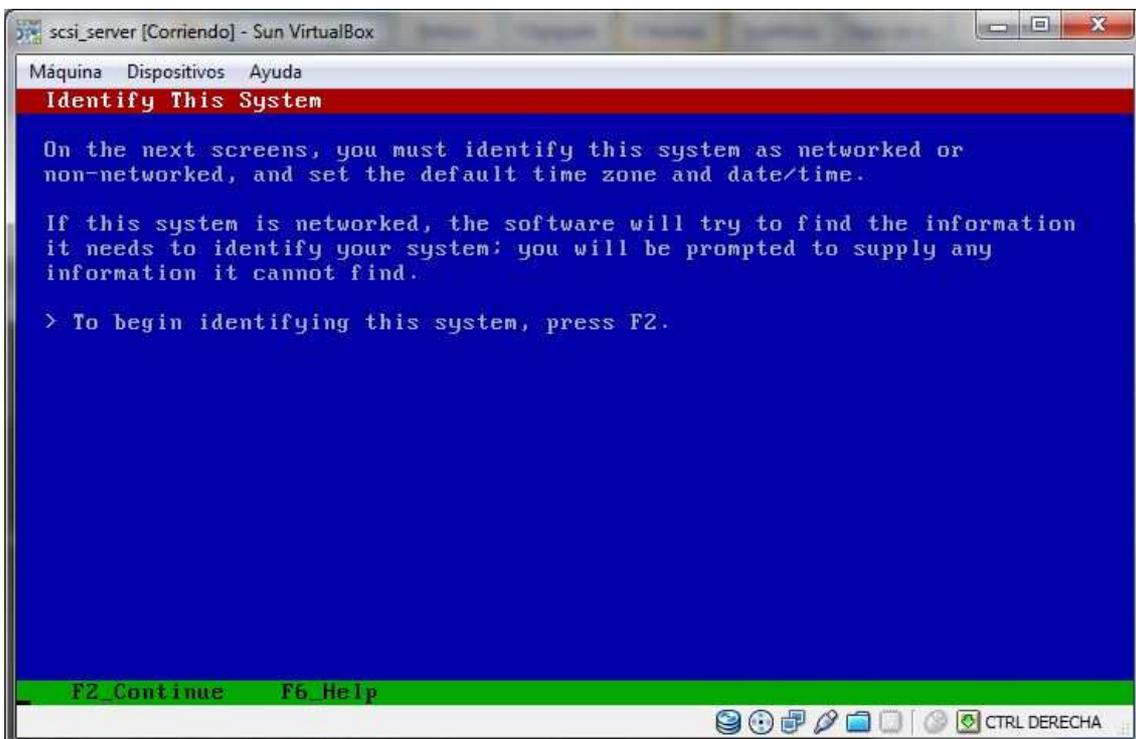
53. Solaris Instalación 3.

En la siguiente ventana nos solicitará el idioma a utilizar. Elegiremos inglés ya que empresarialmente no se suele utilizar en español.



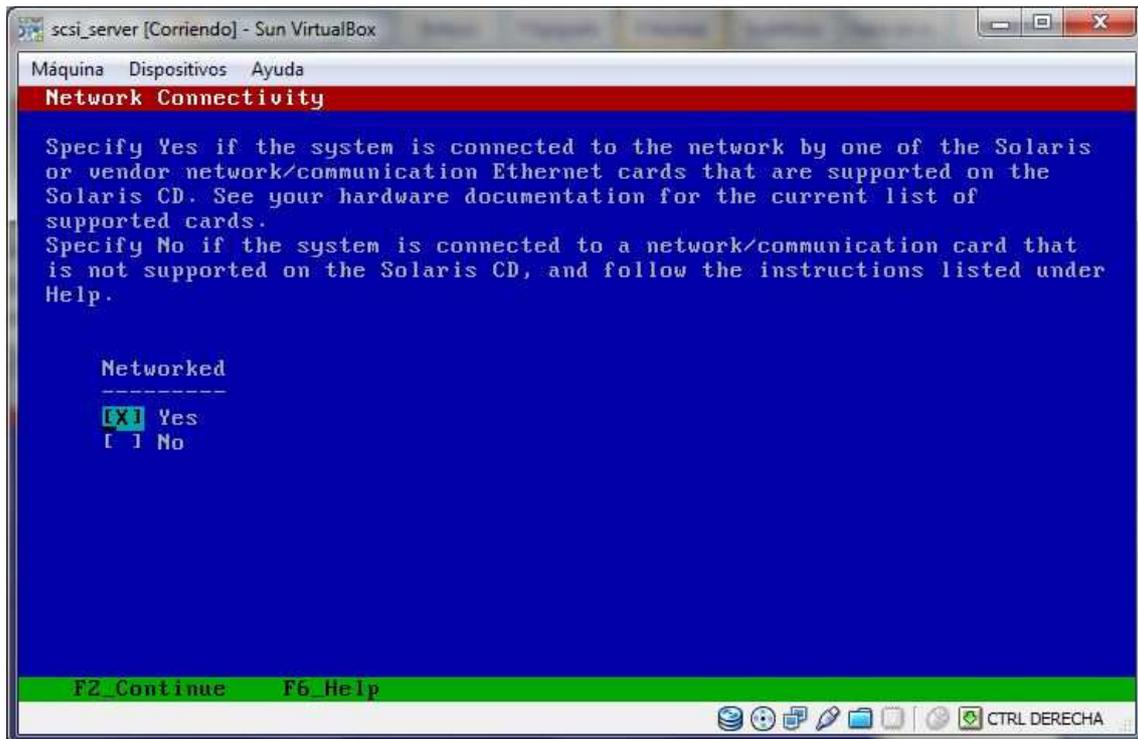
54. Solaris Instalación 4.

Tras estas opciones, nos aparecerá una ventana en la cual nos indican que se pulse F2 para comenzar la identificación del sistema. Pulsamos F2.



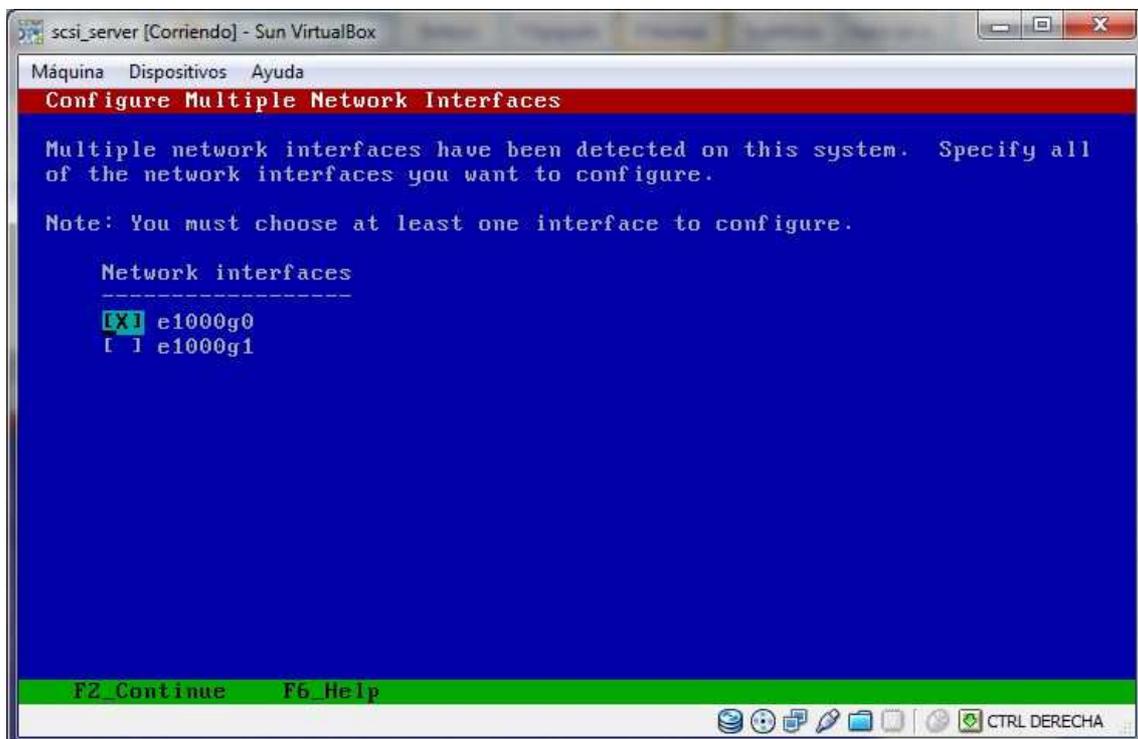
55. Solaris Instalación 5.

Nos solita indicar si el sistema estará conectado a una red. Seleccionamos “yes”



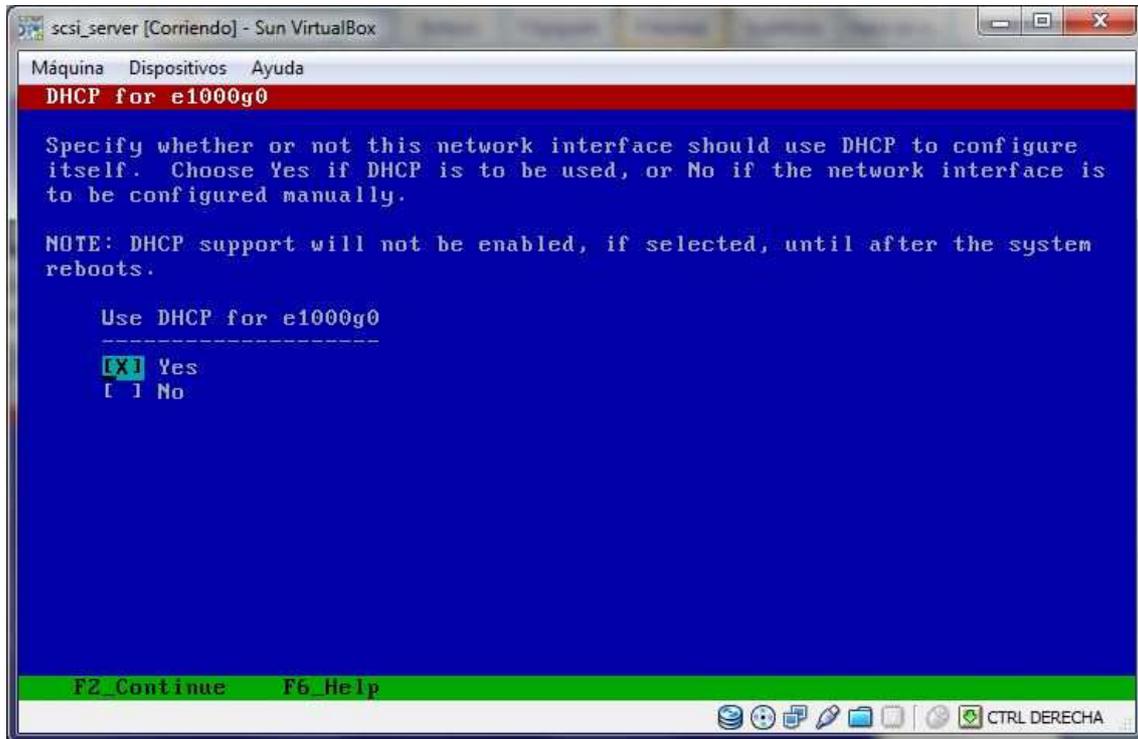
56. Solaris Instalación 6.

En la siguiente ventana nos solicita indicar qué interfaz queremos configurar. Desde VB podemos añadir y quitar interfaces. Seleccionamos el primero.



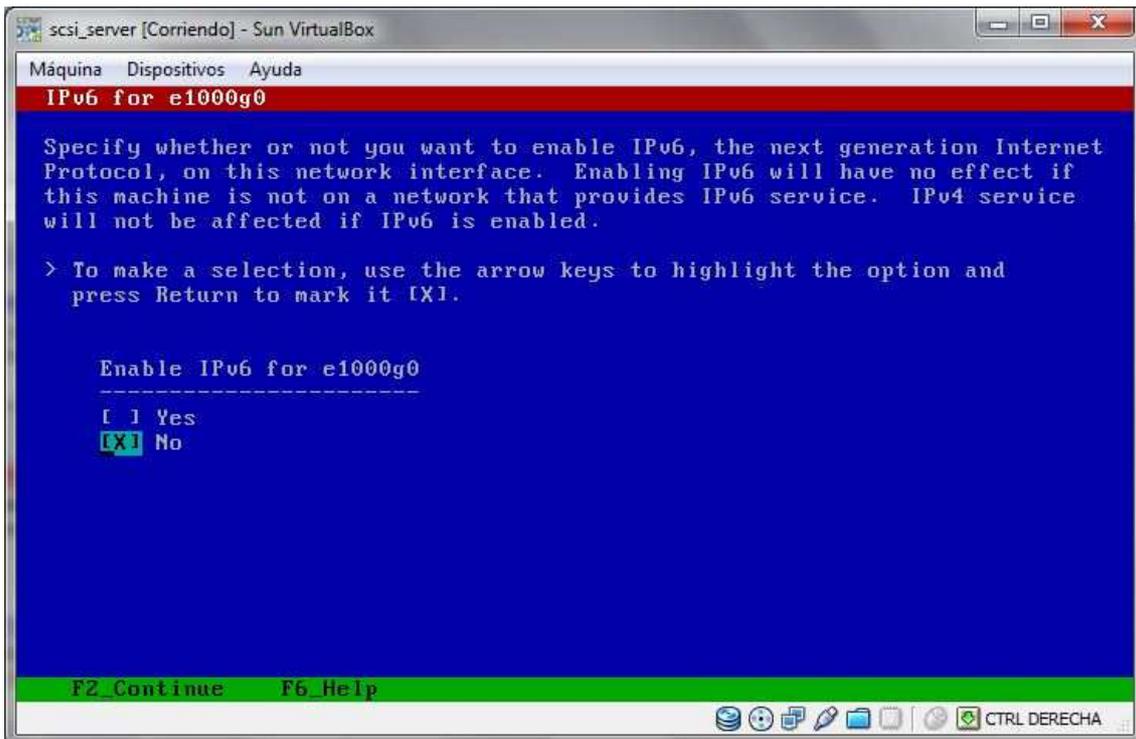
57. Solaris Instalación 7.

En la siguiente ventana marcamos “Use DHCP” ya que el PC host está conectado a internet a través de un router con DHCP configurado. De esta manera se nos asignará una IP en el primer inicio de la máquina y siempre se nos asignará la misma IP ya que esta IP el router la asocia a cada número de MAC.



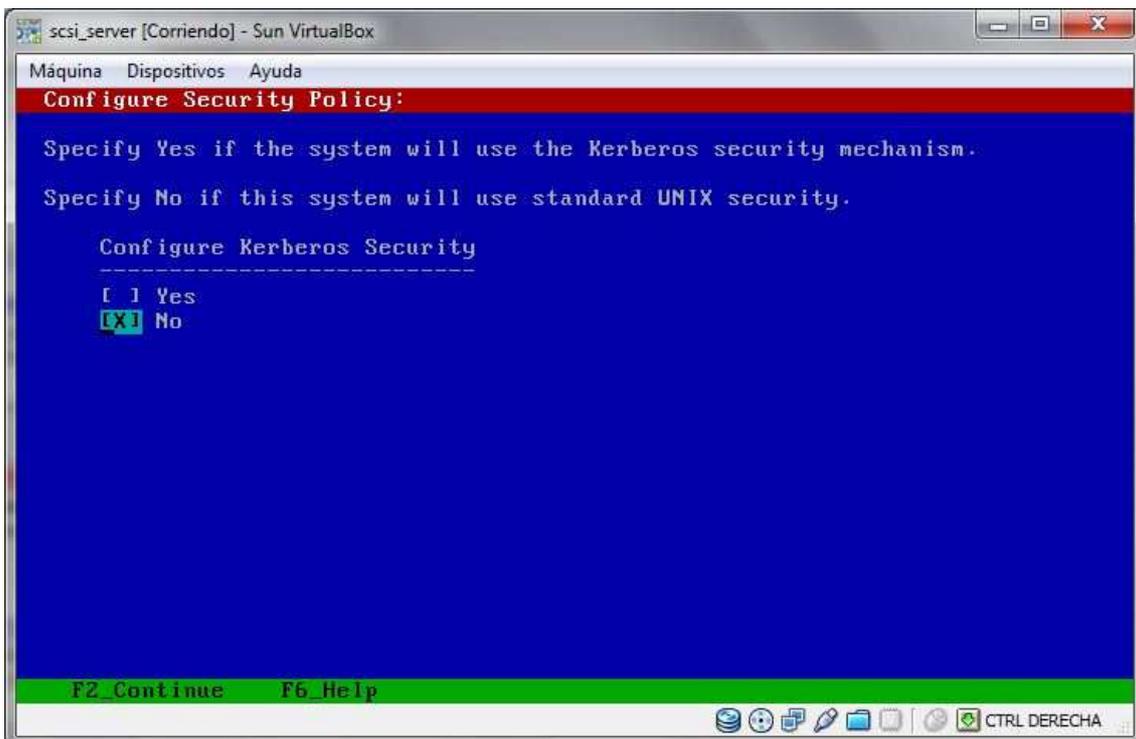
58. Solaris Instalación 8.

En la siguiente ventana indicamos que no queremos configurar IPv6.



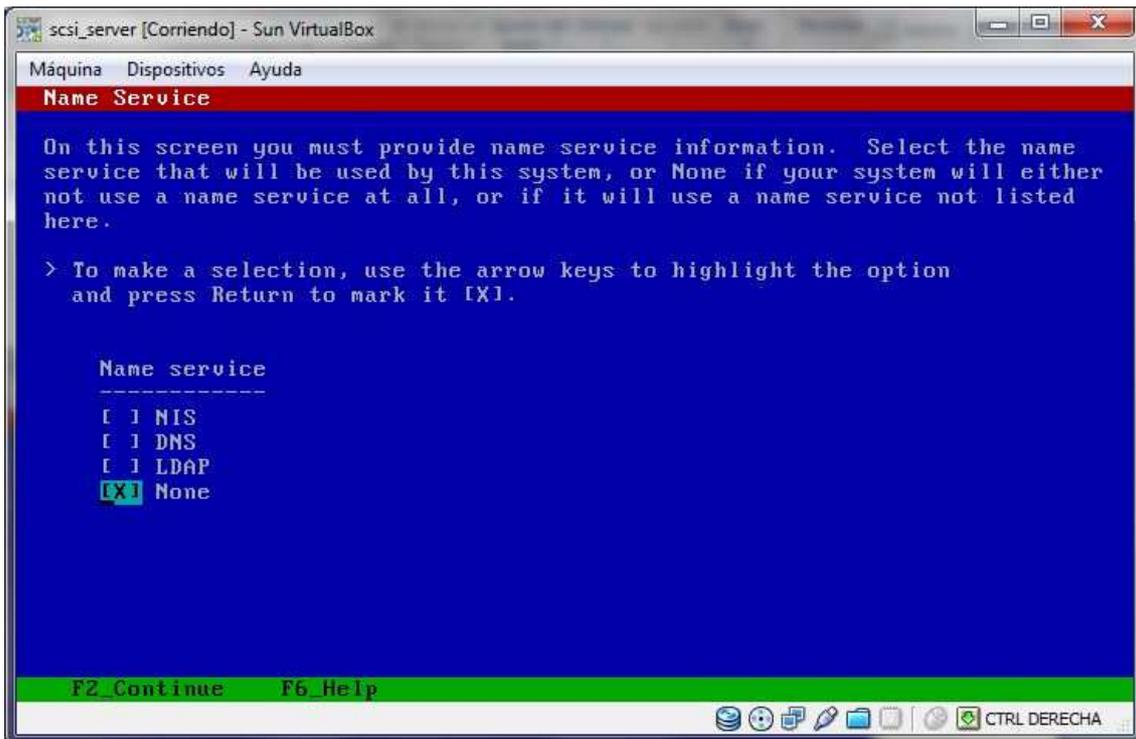
59. Solaris Instalación 9.

En la siguiente ventana indicamos que no queremos configurar Kerberos.



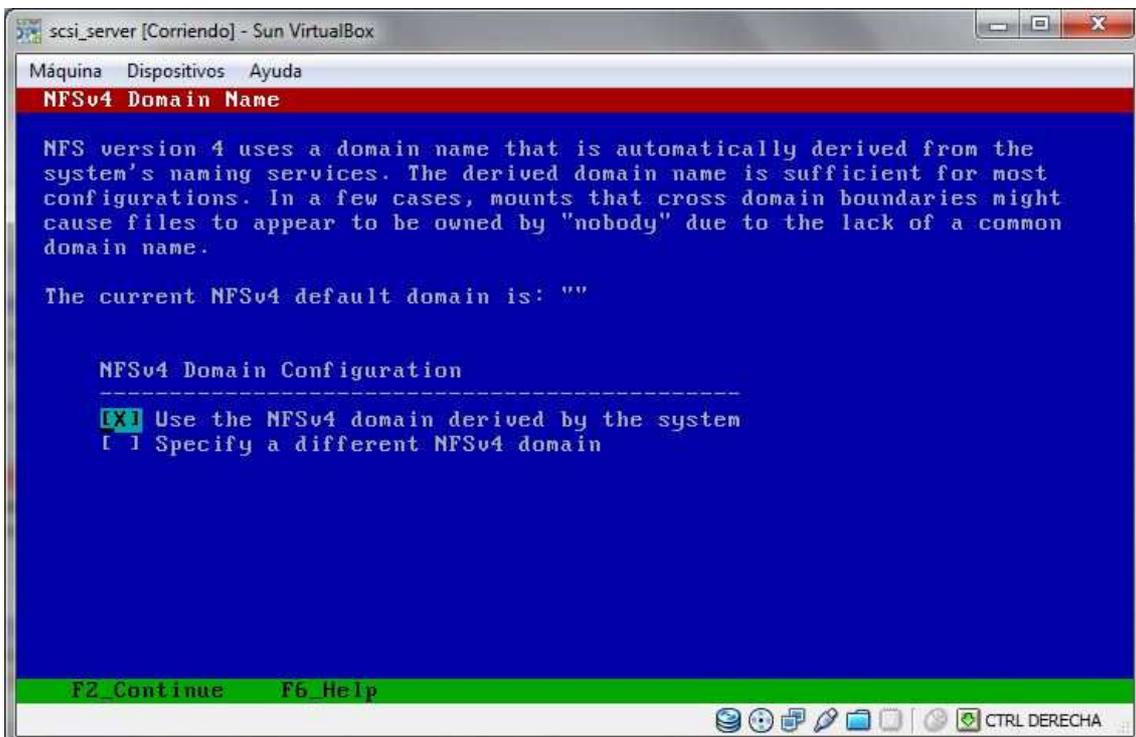
60. Solaris Instalación 10.

En la siguiente ventana indicamos que no queremos configurar ningún servicio de nombres.



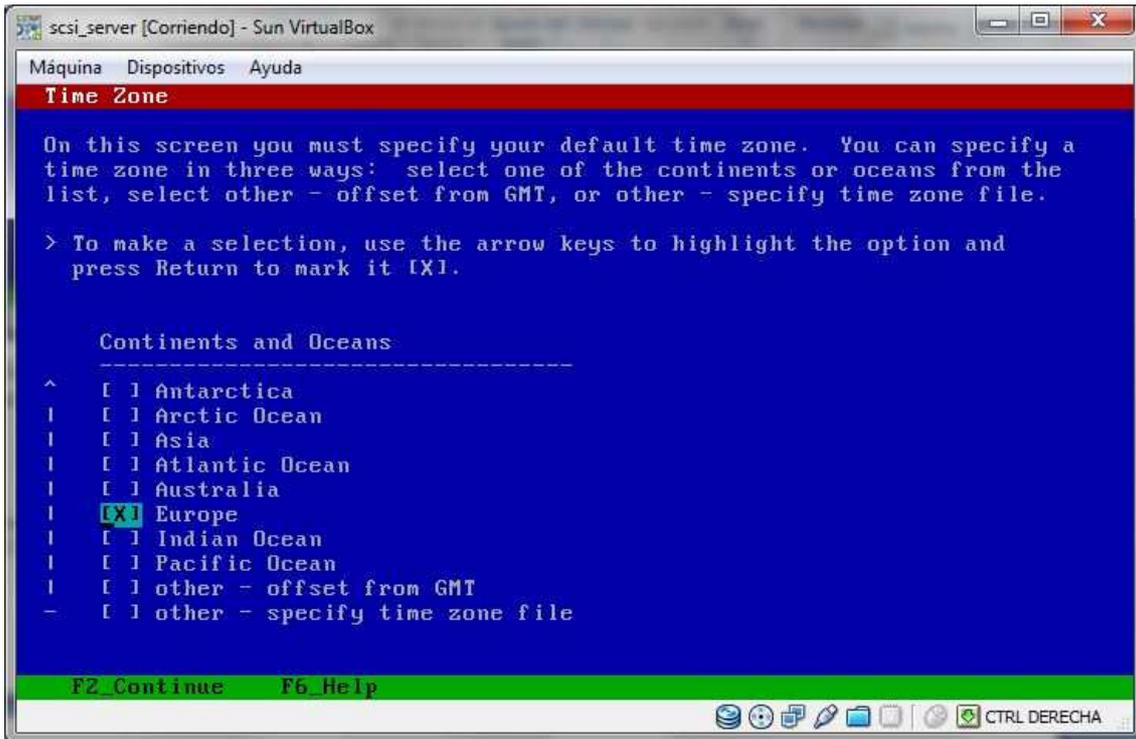
61. Solaris Instalación 11.

En la siguiente ventana de opciones indicamos que el nombre de dominio para NFS sea el del sistema por defecto.

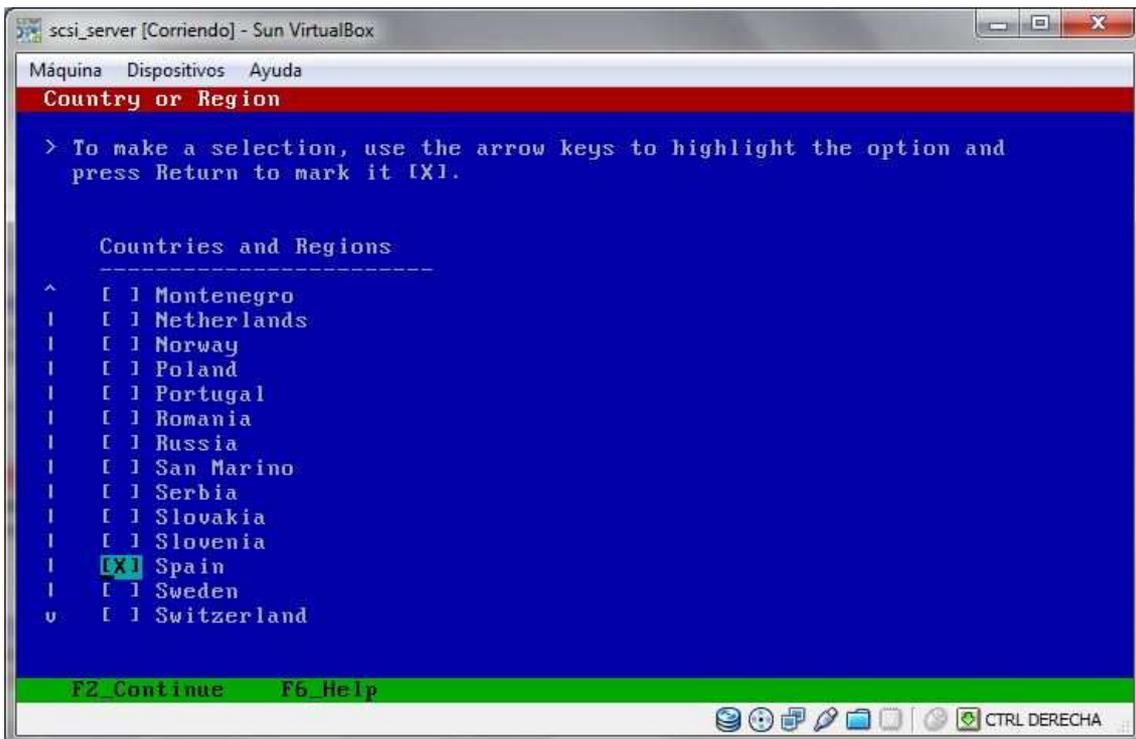


62. Solaris Instalación 12.

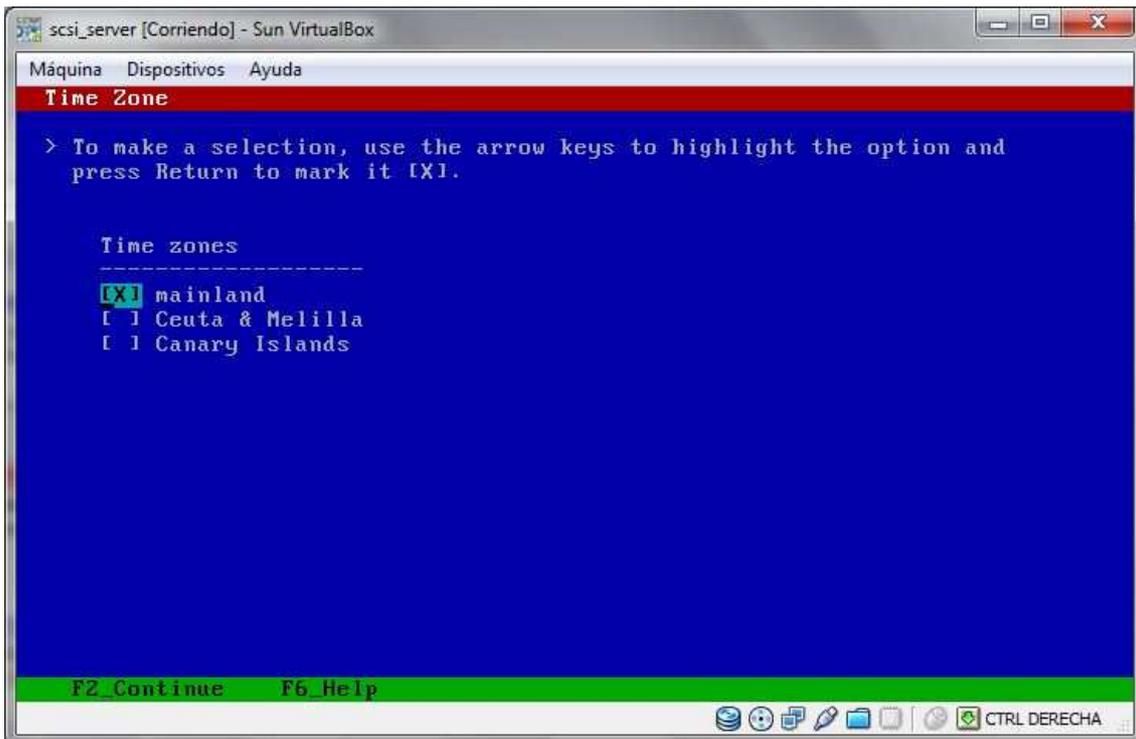
En la siguiente ventana seleccionamos el continente en el cual se encuentra el equipo para configurar la zona horaria.



63. Solaris Instalación 13.

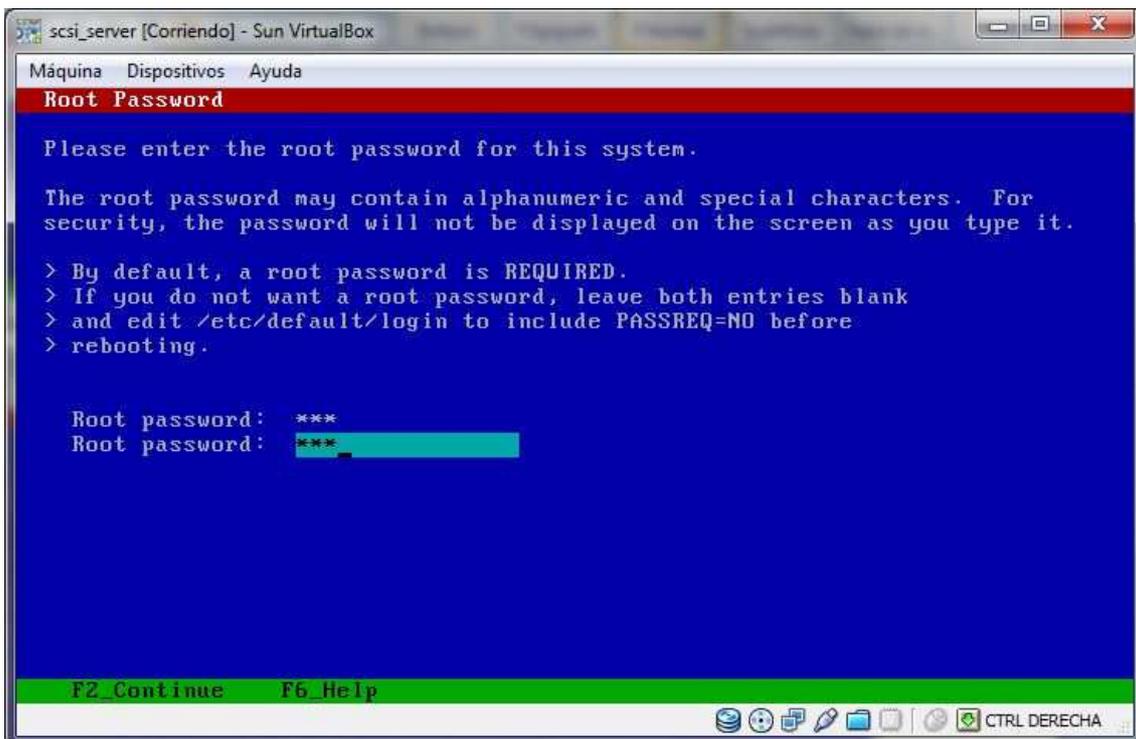


64. Solaris Instalación 14.



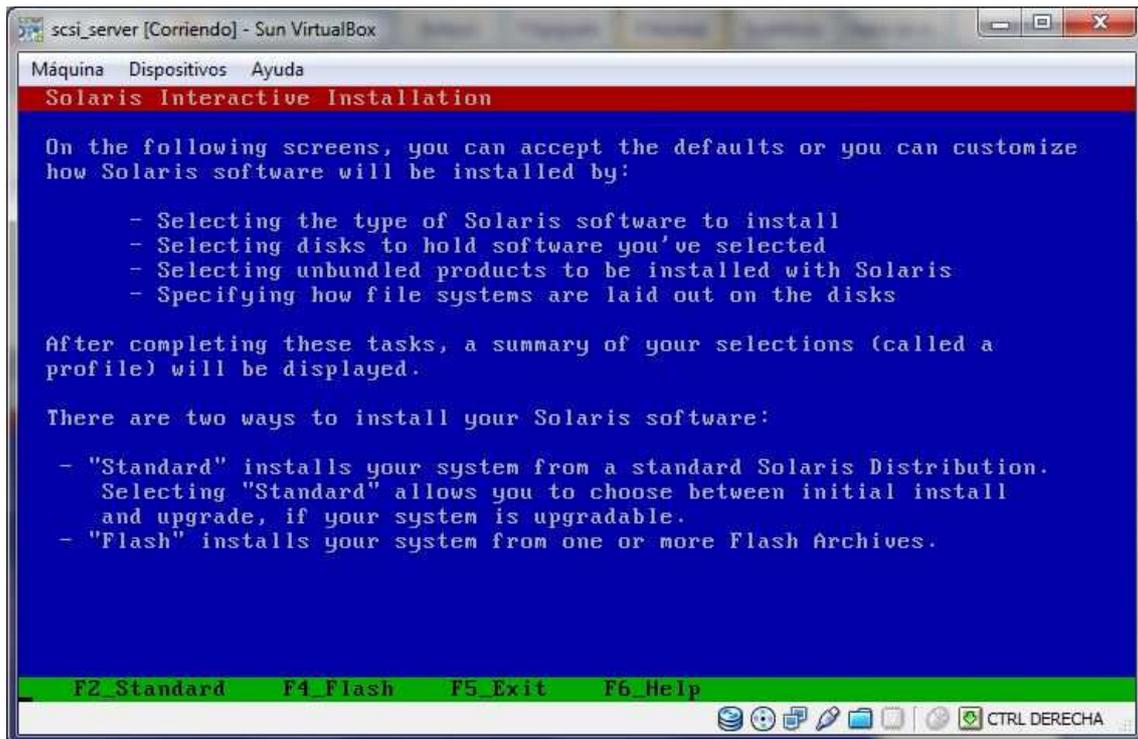
65. Solaris Instalación 15.

Tras la configuración del uso horario, introducimos una password para el usuario root.



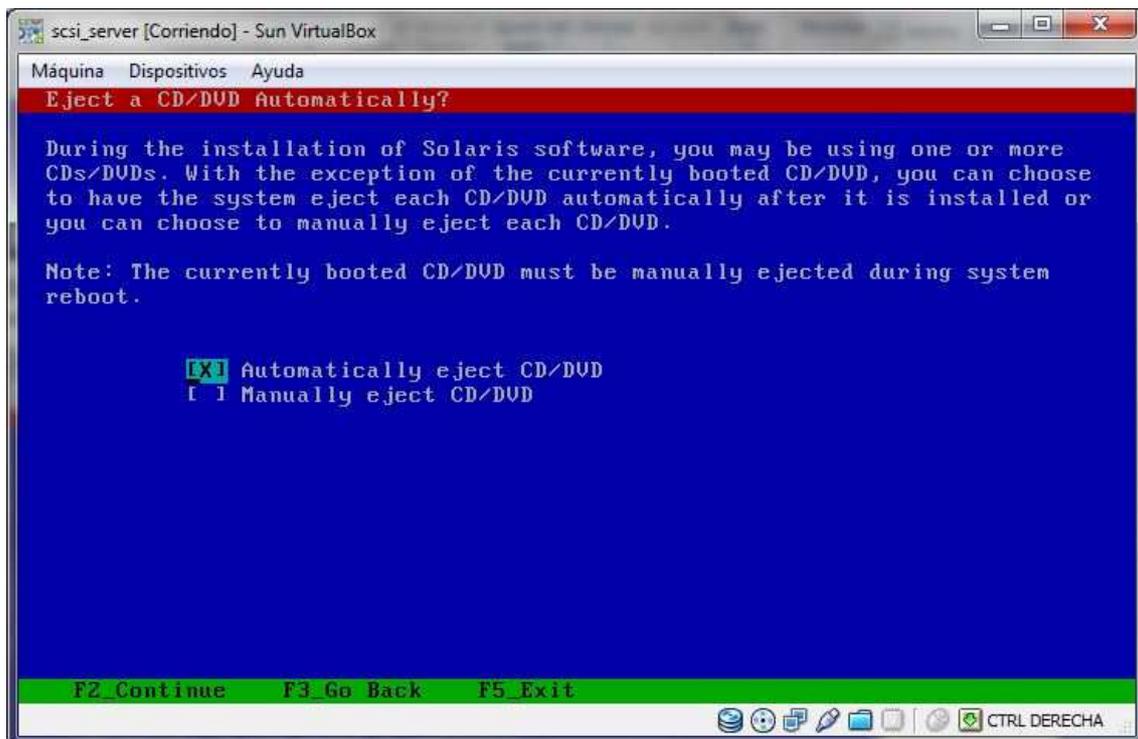
66. Solaris Instalación 16.

Tras estos pasos, la identificación del sistema está terminada. Ahora el instalador solicitará el tipo de instalación que deseamos realizar. Elegiremos estándar.



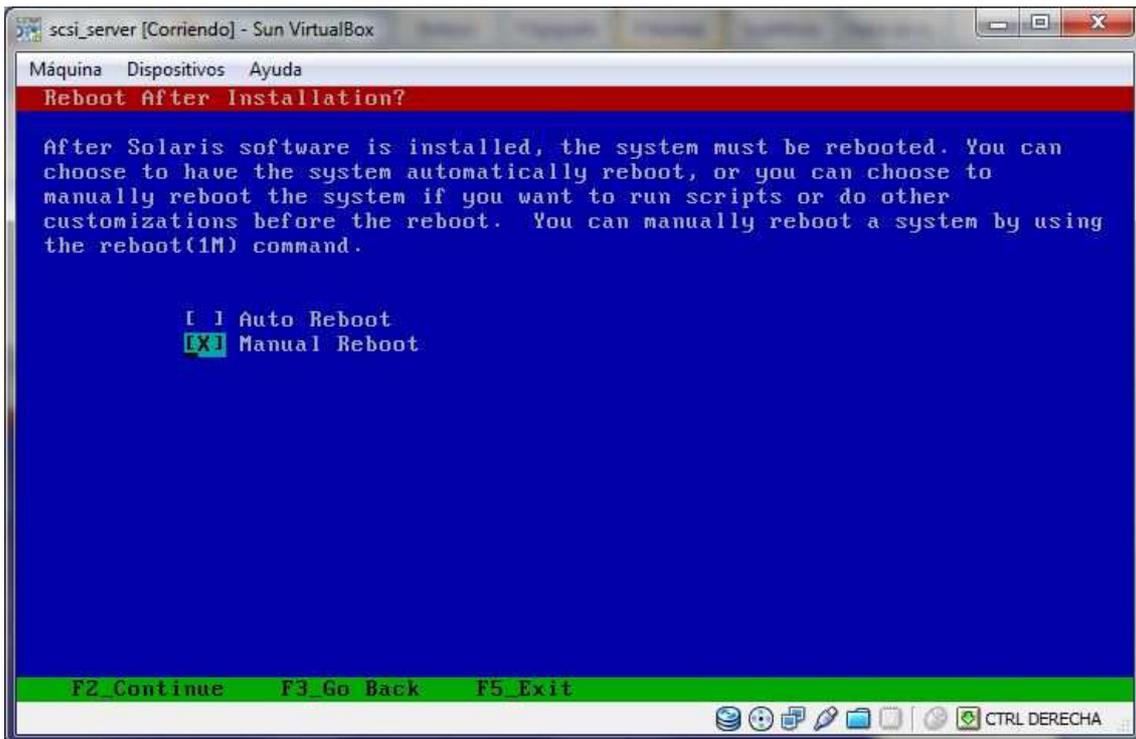
67. Solaris Instalación 17.

Le indicamos que extraiga automáticamente el CD al terminar la instalación.



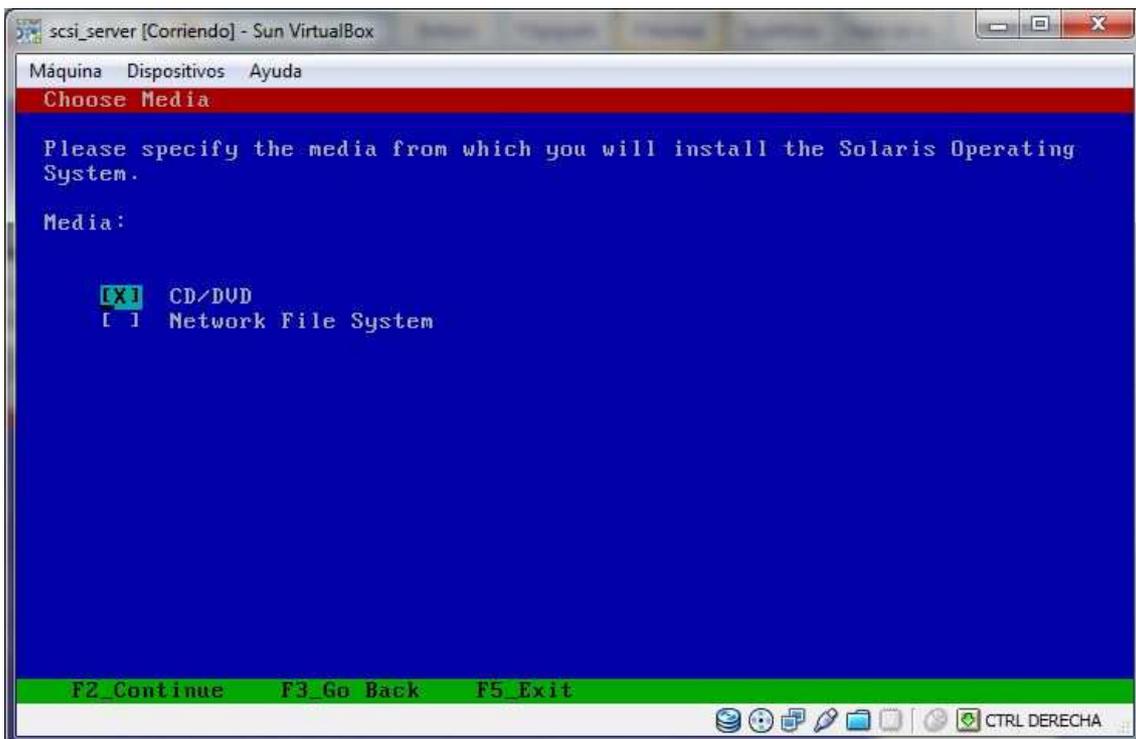
68. Solaris Instalación 18.

Le indicamos que el reboot tras la instalación no queremos que sea automático. Queremos realizarlo nosotros.



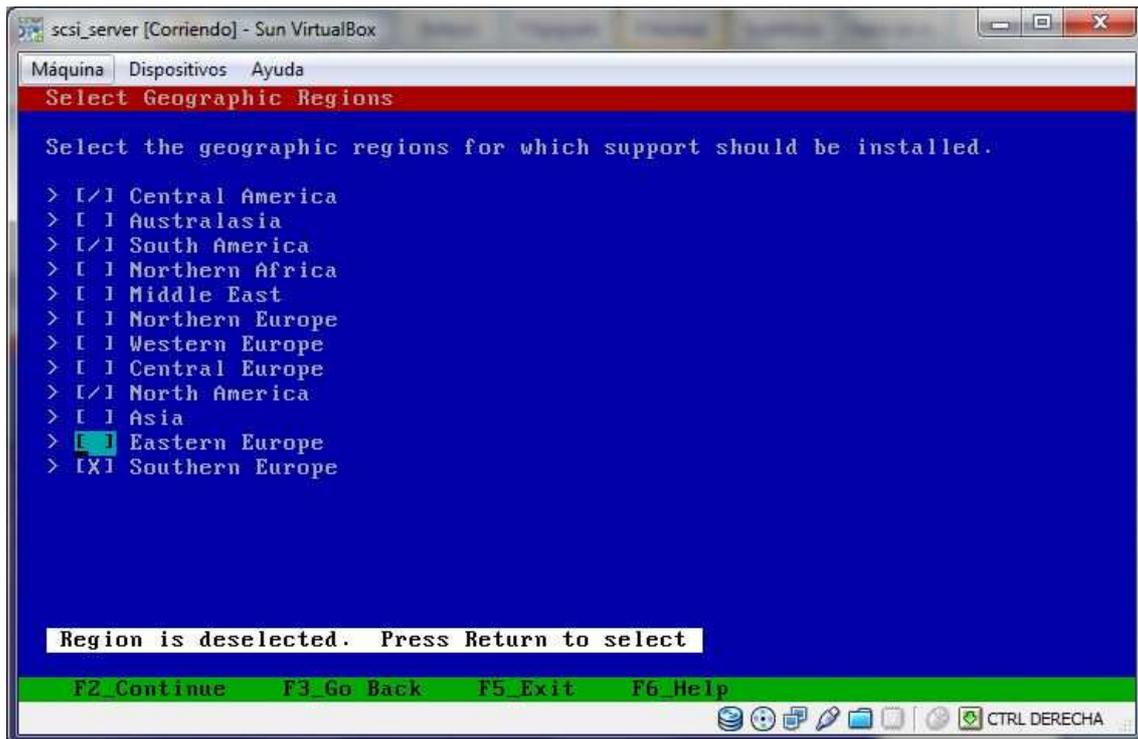
69. Solaris Instalación 19.

Ahora seleccionamos el CD/DVD como origen de los datos para la instalación.



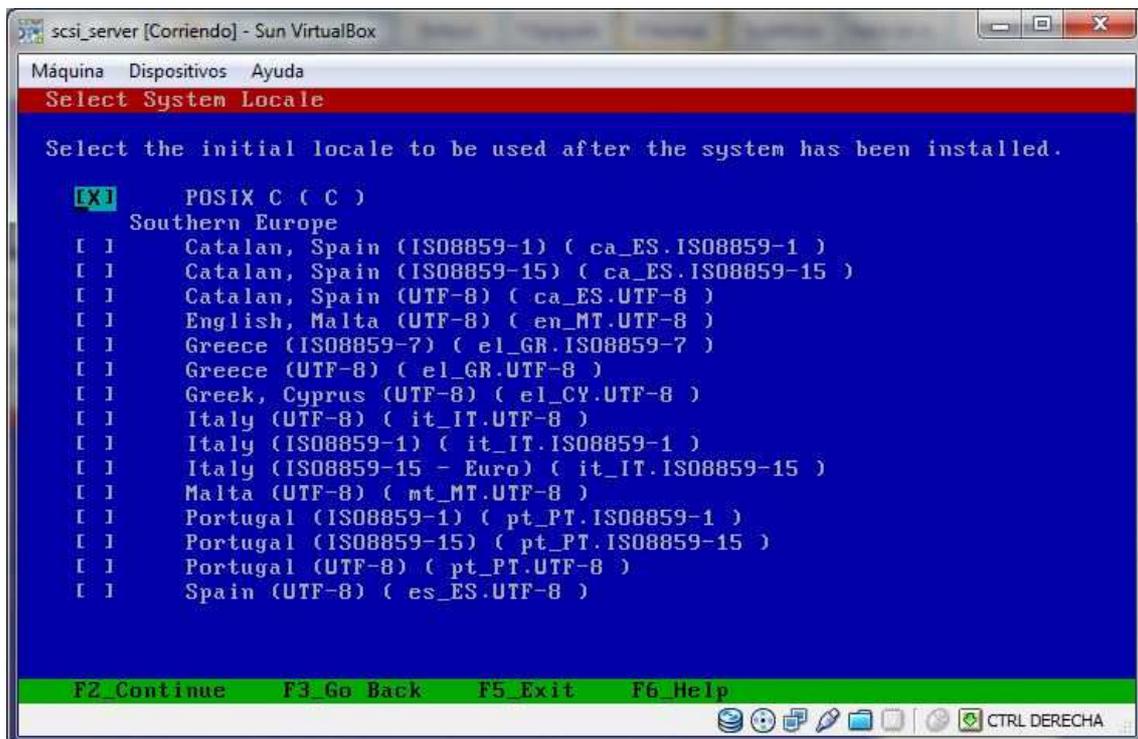
70. Solaris Instalación 20.

Lo siguiente es aceptar los términos de licencia. Una vez aceptados nos solicita elegir la región para la cual será instalado el soporte (fuentes, signos, símbolos.....)



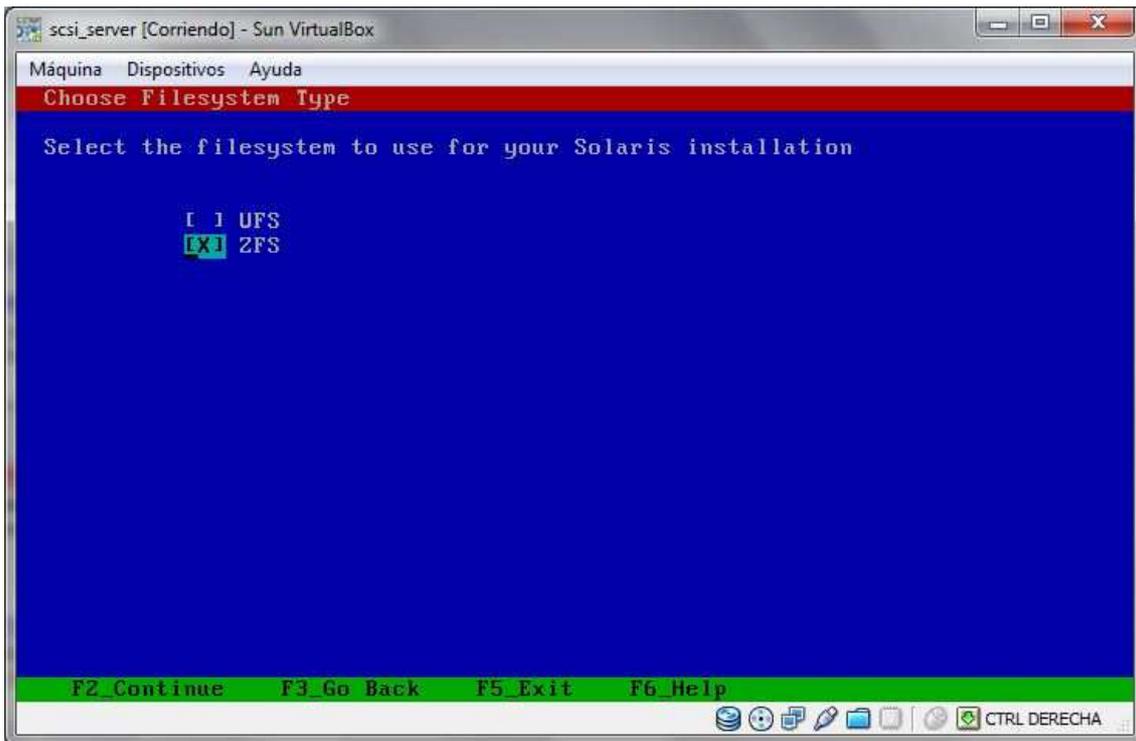
71. Solaris Instalación 21.

Elegimos Poxis C.



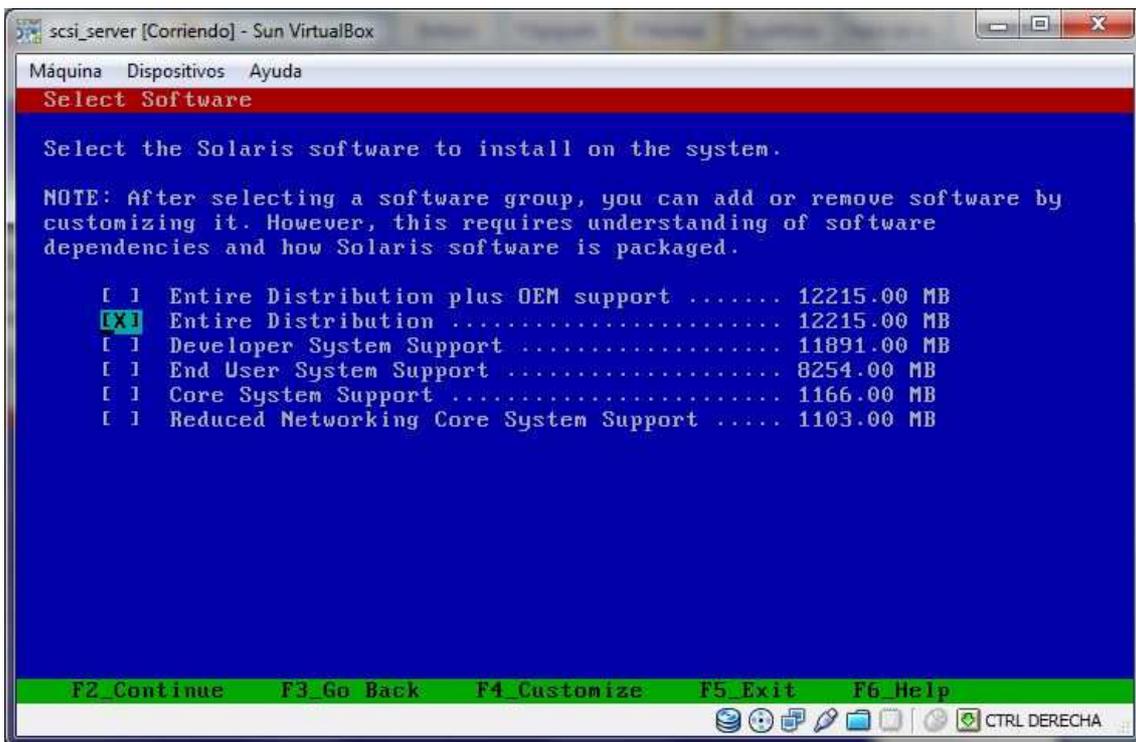
72. Solaris Instalación 22.

Tras esto, pregunta en qué tipo de Filesystem queremos instalar el sistema operativo. Las nuevas versiones de Solaris permiten arrancar desde ZFS por lo que seleccionamos éste.



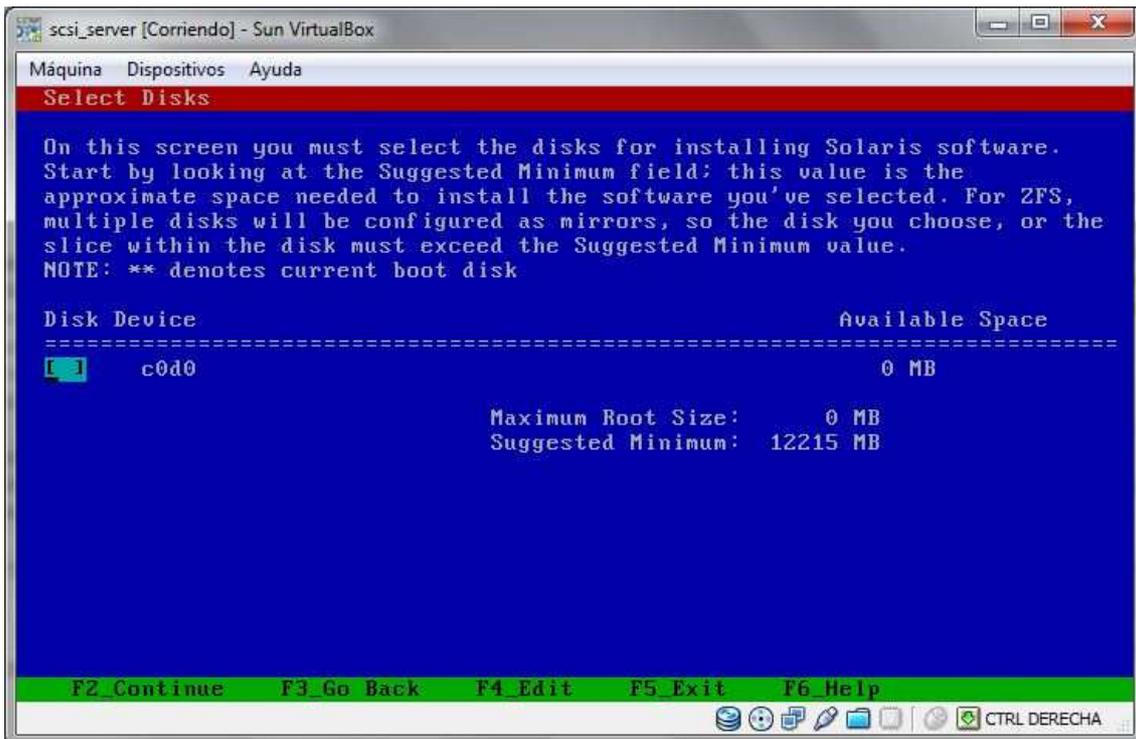
73. Solaris Instalación 23.

Marcamos que queremos instalar la distribución entera.



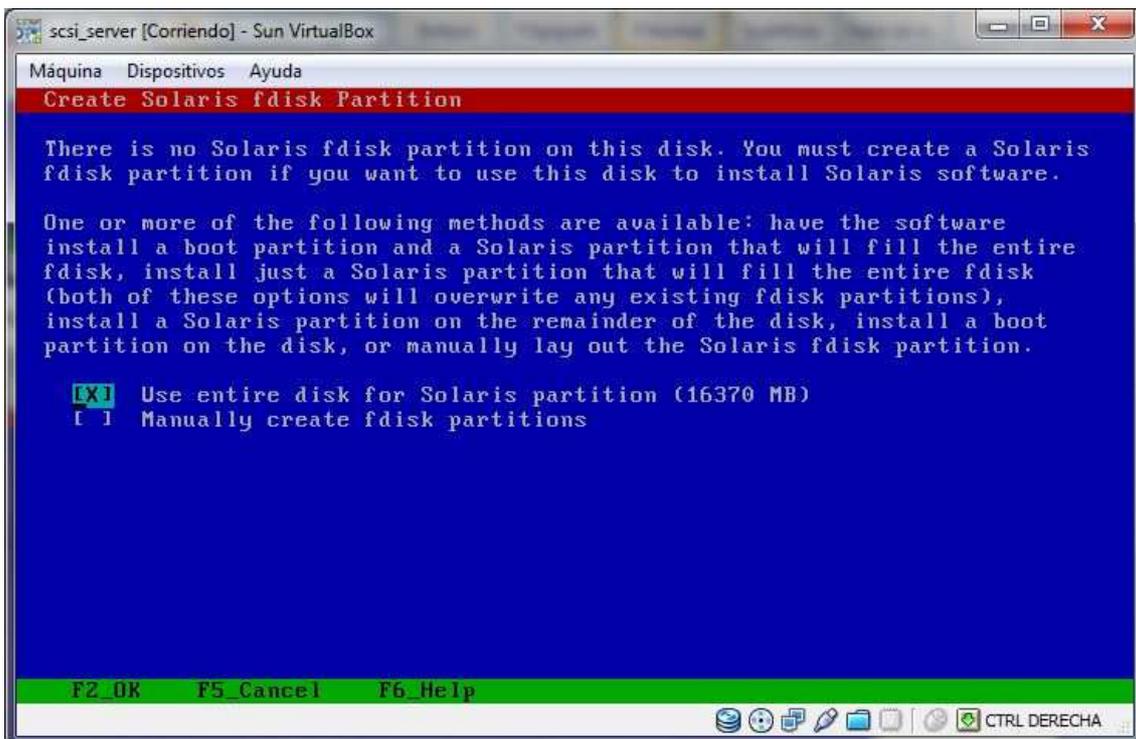
74. Solaris Instalación 24.

Ahora seleccionamos el disco sobre el cual instalar el sistema operativo.



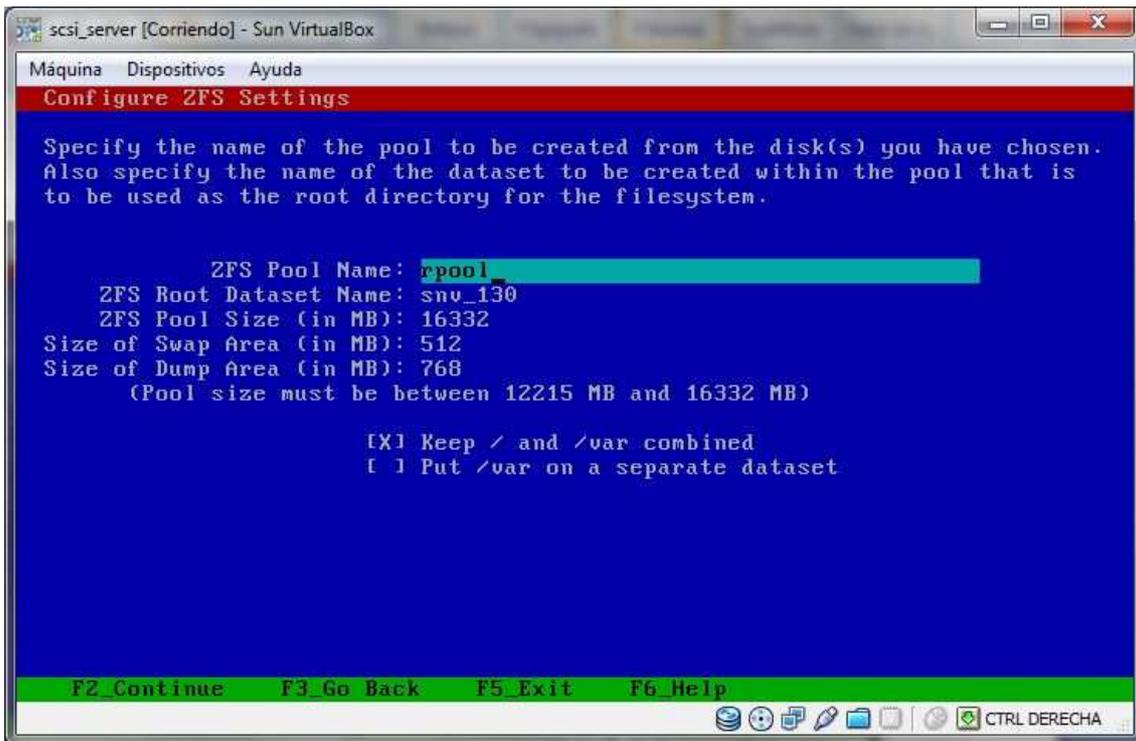
75. Solaris Instalación 25.

Indicamos que utilice el disco entero, sin particiones.



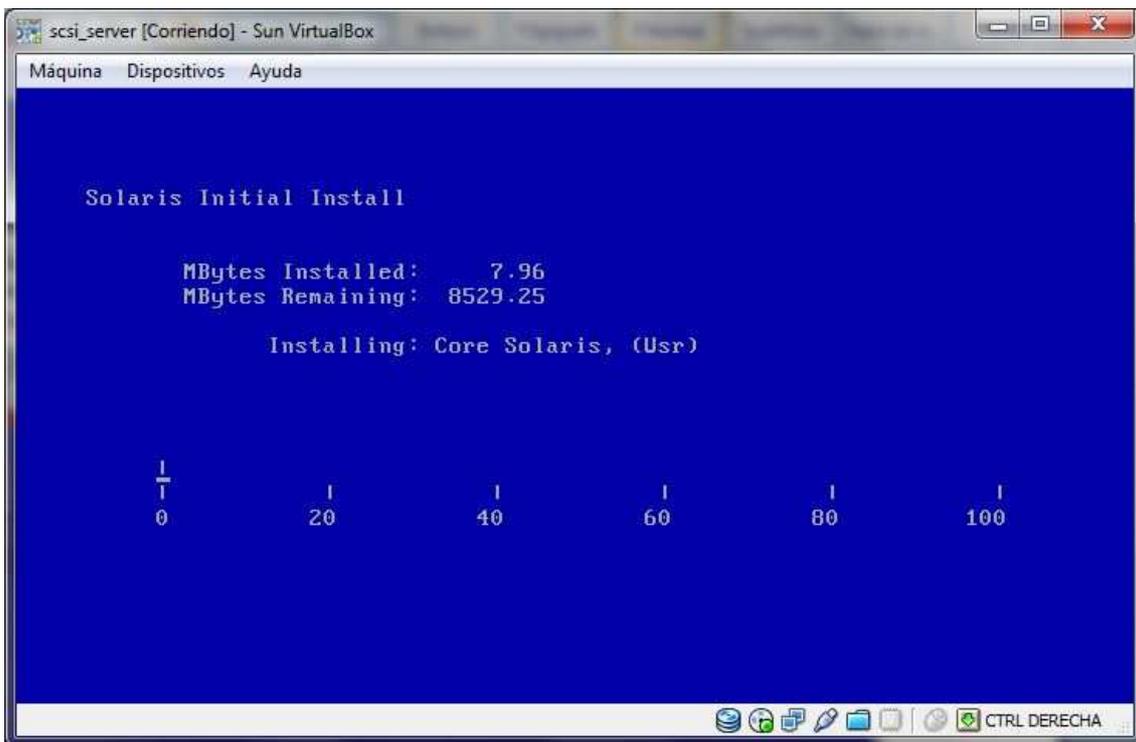
76. Solaris Instalación 26.

Seleccionamos el nombre que queremos que tenga el pool de ZFS y la cantidad de espacio asignada para swap y para dump (si la máquina tiene un fallo, se registra un core que se guarda en esta área).



77. Solaris Instalación 27.

Aceptamos el resto de pantallas y comenzará la instalación.



78. Solaris Instalación Progreso.

4. Diseño arquitectónico del servidor de almacenamiento.

En este capítulo vamos a definir los elementos necesarios para que el servidor de almacenamiento sea totalmente funcional.

4.1 Componentes necesarios para implementar la solución.

La solución adoptada en este proyecto se caracteriza principalmente por permitir utilizar 3 protocolos distintos para presentar almacenamiento externo: NFS, Samba e iSCSI.

NFS y Samba vienen instalados en todos los sistemas operativos actuales, o al menos, está disponible para instalarlo. Sin embargo la parte más complicada es decidir cómo implementar el target iSCSI. La ventaja de utilizar opensolaris frente al resto es que dispone del software Comstar incluido, lo que nos permitirá gestionar la parte iSCSI del servidor.

Aunque NFS, Samba e iSCSI sean protocolos muy diferentes (sobre todo con iSCSI), todos tienen en común que necesitan almacenamiento local en el servidor para poder prestar servicio. Para ello, la parte más importante a decidir en nuestro servidor será cómo gestionar el almacenamiento local, dependiendo de qué características se quieran destacar (Alta disponibilidad, alto rendimiento, etc).

Para gestionar el espacio de almacenamiento local comenzaremos por decidir cómo y de qué manera gestionar los discos locales. Podemos utilizar la tecnología Raid para proveer de redundancia a los datos y en caso de fallo no exista pérdida de datos. Esta característica es muy importante ya que para los servidores que utilicen el almacenamiento externo presentado por el servidor, debe ser transparente qué ocurran fallos en disco. Es más, si es posible deberemos garantizar al máximo posible la continuidad de los datos.

La elección de los diferentes niveles de Raid dependerá de las necesidades en cuanto a seguridad, velocidad, capacidad y coste. Cada nivel Raid ofrece una combinación específica de tolerancia a fallos, rendimiento y coste. La mayoría de los niveles Raid pueden satisfacer de manera efectiva uno o dos de estos criterios.

- Raid 0: Transferencia de datos más alta, pero no tiene tolerancia a fallos
- Raid 1: Mirroring. Tolerancia a fallos ya que utiliza mínimo 2 discos, los cuales contienen la misma información. Penaliza el hecho de perder el espacio de un disco para copiar el otro.
- Raid 0+1 o Raid10: Es la combinación de los 2 anteriores y permite altas tasas de transferencia y tolerancia a fallos. Se necesita un número par de discos y mínimo 4.
- Raid 2: Acceso paralelo con discos especializados. Redundancia a través de código Hamming. Debido a que necesita características especiales en los discos apenas se utiliza.

- Raid 3: Acceso síncrono con un disco dedicado a paridad. Son necesarios al menos 3 discos para implementarlo. Penaliza su rendimiento en transacción ya que todos los discos operan a la vez.
- Raid 4: Acceso independiente con un disco dedicado a paridad. La ventaja que tiene frente a Raid 3 es el acceso de manera individual a cada disco.
- Raid 5: Acceso independiente con paridad distribuida. Este tipo de array optimiza la capacidad del sistema (permite la utilización de hasta el 80% de la capacidad el conjunto de discos) y además provee tolerancia a fallos.

Por lo tanto, para nuestra aplicación lo más lógico sería utilizar Raid5, ya que es el nivel de Raid más eficaz y el de uso preferente para las aplicaciones empresariales. Comparado con otros niveles de Raid con tolerancia a fallos, Raid 5 ofrece la mejor relación rendimiento-coste en un entorno con varios discos.

Además, si en vez de trabajar en un entorno virtualizado, trabajásemos en una plataforma hardware empresarial, sería conveniente realizar un estudio para evitar puntos únicos de fallo. Si se va a utilizar un Raid 5 con 4 discos, es necesario evitar que los 4 discos estén conectados a la misma controladora. La situación ideal sería disponer de 4 controladoras y cada disco conectado a una controladora. Pero sería suficiente con poder dividir los discos entre el número de controladoras para disminuir las posibilidades de fallos en caso de la pérdida de una controladora por problemas hardware.

El último componente necesario para implementar el servidor de almacenamiento es un gestor de volúmenes.

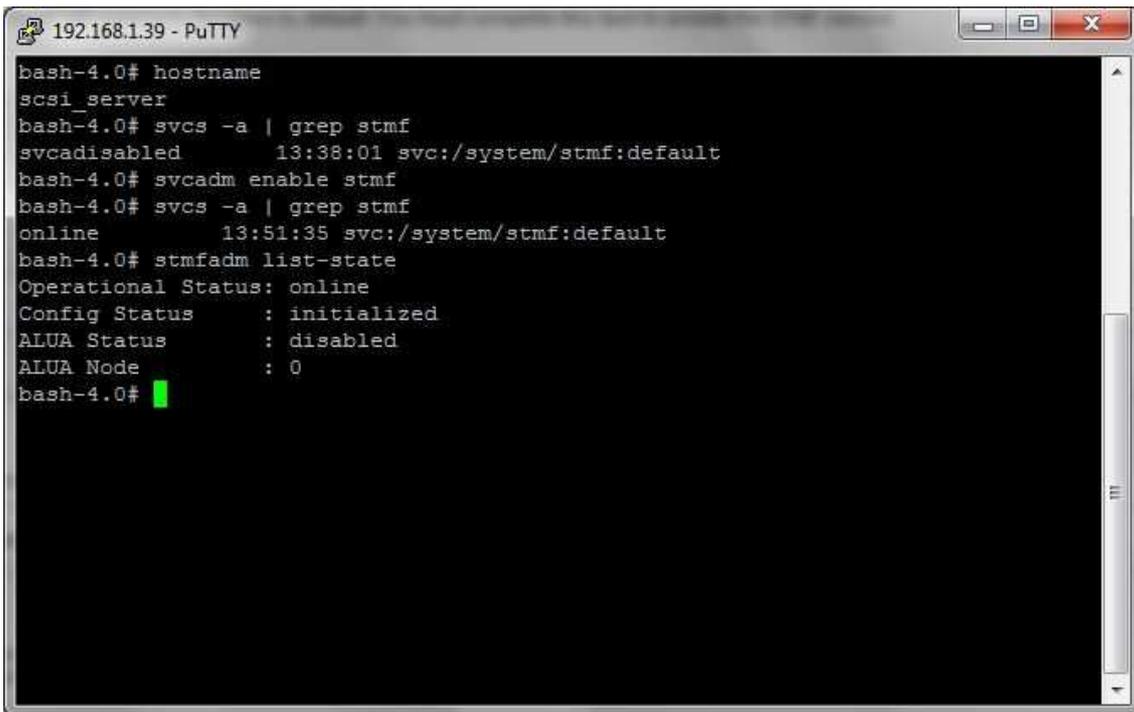
Un gestor de volúmenes es un software que nos permite añadir una capa por encima de los discos, de manera que la administración y gestión del espacio de almacenamiento sea más sencilla, así como poder evitar las limitaciones físicas de los sistemas operativos y/o los discos y gestionar de forma software los tipos de Raid, sin tener que tener una tarjeta para realizarlo por hardware. Existen varios tipos de gestores de volúmenes, los más conocidos son Veritas Volume Manager (Vxvm), Logical Volume Manager (LVM) y Solaris Volume Manager. La ventaja que aporta utilizar opensolaris como sistema operativo es la posibilidad de utilizar ZFS, que como veremos más adelante es más potente que cualquier de los gestores de volúmenes actuales.

4.2 Creación y configuración de Logical Unit

Antes de crear las Logical Unit es necesario activar el scsi target mode framework (stmf) para poder gestionar las Lun scsi.

Comstar utiliza el stmf para almacenar sus datos persistentes (configuración actual) tales como Logical Unit Mapping, definiciones de Hosts Groups y definiciones de Targets Groups.

Stmf corre como un servicio de Solaris, por lo que se administra mediante la línea de comandos con el comando svcadm. Para ver su estado podemos utilizar svcs. En la siguiente pantalla se puede ver cómo por defecto está deshabilitado y se habilita mediante la línea de comandos.



```
192.168.139 - PuTTY
bash-4.0# hostname
scsi_server
bash-4.0# svcs -a | grep stmf
svcadisabled      13:38:01 svc:/system/stmf:default
bash-4.0# svcadm enable stmf
bash-4.0# svcs -a | grep stmf
online           13:51:35 svc:/system/stmf:default
bash-4.0# stmfadm list-state
Operational Status: online
Config Status      : initialized
ALUA Status       : disabled
ALUA Node         : 0
bash-4.0#
```

79. Servicios STMF.

Para la creación de Logical Unit, se utiliza el framework sbd (scsi block device) que en línea de comandos se administra con el comando sbdadm.

4.2.1 Tipos de Logical Unit

A través de Comstar, se pueden crear 4 tipos diferentes de Logical Unit.

- Logical Unit basada en fichero: Consiste en crear un fichero mediante mkfile, y exportarlo como si fuese un dispositivo Scsi.
- Logical Unit basada en Thin-Provisioned: Thin-Provisioned posibilita que se cree una Logical Unit de 100GB por ejemplo sin utilizar todo el espacio asignado en el momento. Se definen los 100GB que será lo que se indicará al initiator que hay de espacio libre pero el espacio real es sólo el que el initiator escribe. Se utiliza la expansión dinámica para ir asignando el espacio al initiator.
- Logical Unit basada en partición de disco: Ésta es la forma más simple de todas. Consiste en utilizar una partición de un disco como target.
- Logical Unit utilizando un volumen ZFS: Zfs combina las funcionalidades de un File System y un Gestor de volúmenes. Este tipo de Logical Unit es la que permite mayor versatilidad.

4.2.2 Ventajas de utilizar ZFS frente al resto de opciones

La primera de las ventajas con la que cuenta ZFS frente al resto es la capacidad. Los límites de ZFS están diseñados para ser tan grandes que no se encuentren nunca en la práctica.

Algunos de los límites teóricos son:

- Tamaño máximo de un sistema de ficheros: 16 exabytes

- Tamaño máximo de un fichero: 16 exabytes
- Tamaño máximo de cualquier atributo: 16 exabytes
- Número de snapshots en cualquier sistema de ficheros: 2^{48}
- Número de ficheros en un sistema de ficheros: 2^{48}
- Tamaño máximo de un zpool: 3 x 1023 petabytes
- Número de dispositivos en un zpool: 2^{64}
- Número máximo de zpools por sistema: 2^{64}

Actualmente ZFS sólo está disponible para Solaris (y Opensolaris) aunque ya hay versiones experimentales en Linux y MacOS.

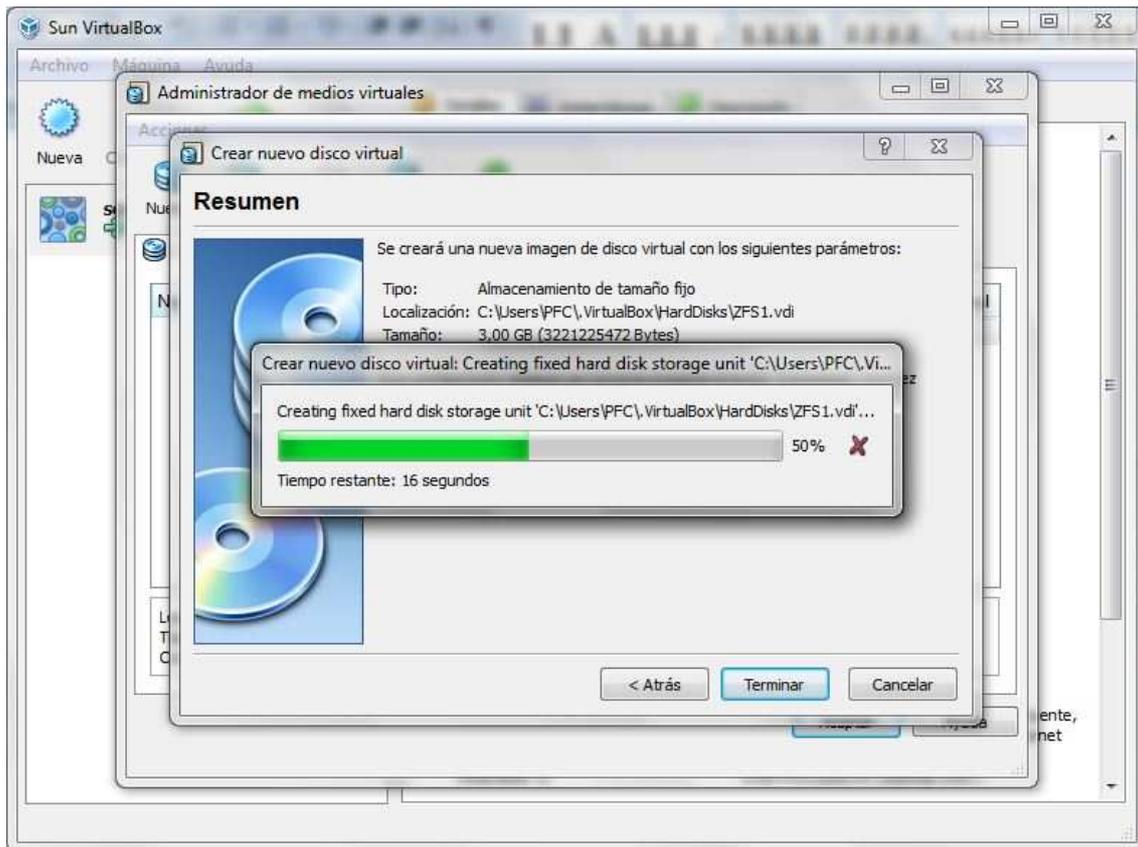
Otra de las ventajas que posee ZFS consiste en su mecanismo de auto-reparación que se basa en hacer un hash de los datos previo a la escritura en el bloque lógico del pool. Una vez escritos los datos se verifica el CRC para comprobar que se han escrito correctamente. En la lectura se realiza la misma operación y si no hay concordancia procede a buscar el bloque en el mirror o calcular los datos a través del sistema de paridad empleado, corrigiendo los datos del bloque dañado.

Otra característica de ZFS son los snapshots, que no son más que instantáneas en un momento concreto del sistema de ficheros. Estas instantáneas permiten hacer backups en caliente sin necesidad de desmontar los sistemas de ficheros así como recuperarse ante errores.

Por todas estas características y muchas más es por lo que ZFS se está convirtiendo en un gestor de volúmenes cada vez más utilizado y un futuro referente para los distintos fabricantes de servidores.

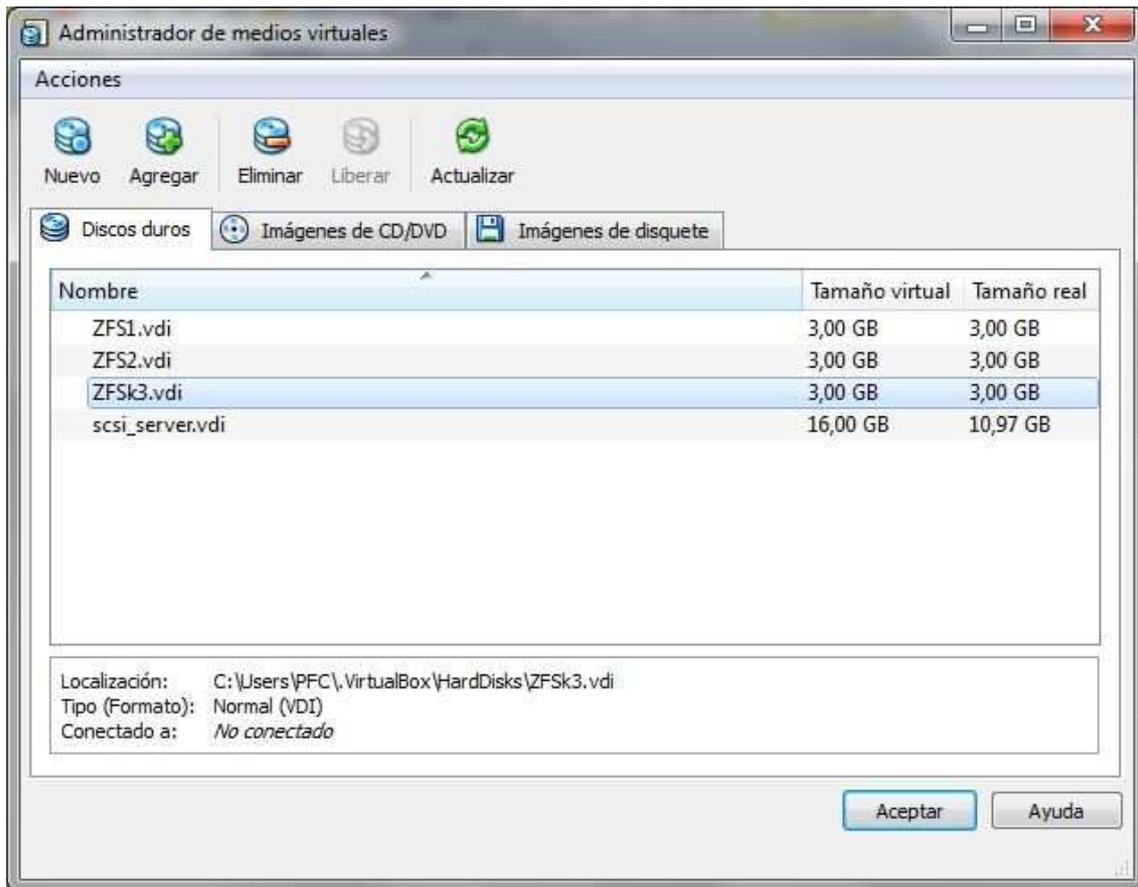
4.2.3 Creación de una LU utilizando ZFS

Para crear una Logical Unit utilizando ZFS utilizaremos 3 discos nuevos virtuales que añadiremos a la máquina virtual. La operación de crear los discos duros virtuales se puede realizar en caliente, sin necesidad de parar ninguna máquina virtual. Para ello vamos a la ventana principal de Virtual Box y pinchamos en la pestaña Archivo. Una vez ahí, pinchamos en “administrador de medios virtuales” que nos abrirá la ventana para gestionar los Discos duros virtuales. Desde esta ventana creamos 3 discos virtuales nuevos de 3 GB cada uno.



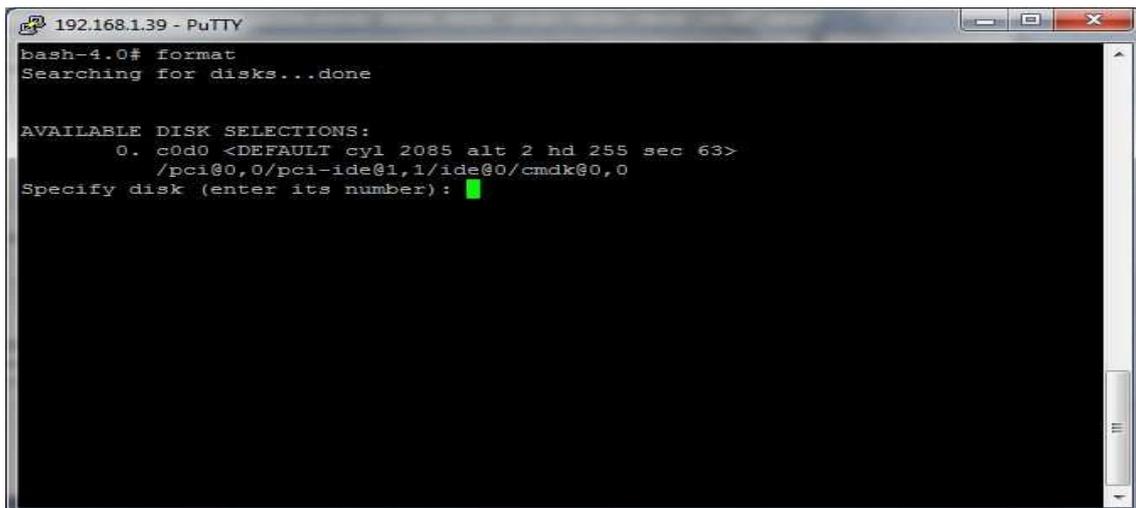
80. Asignar discos LU.

Tras terminar esta operación deberá quedar la siguiente configuración:



81. VB discos LU.

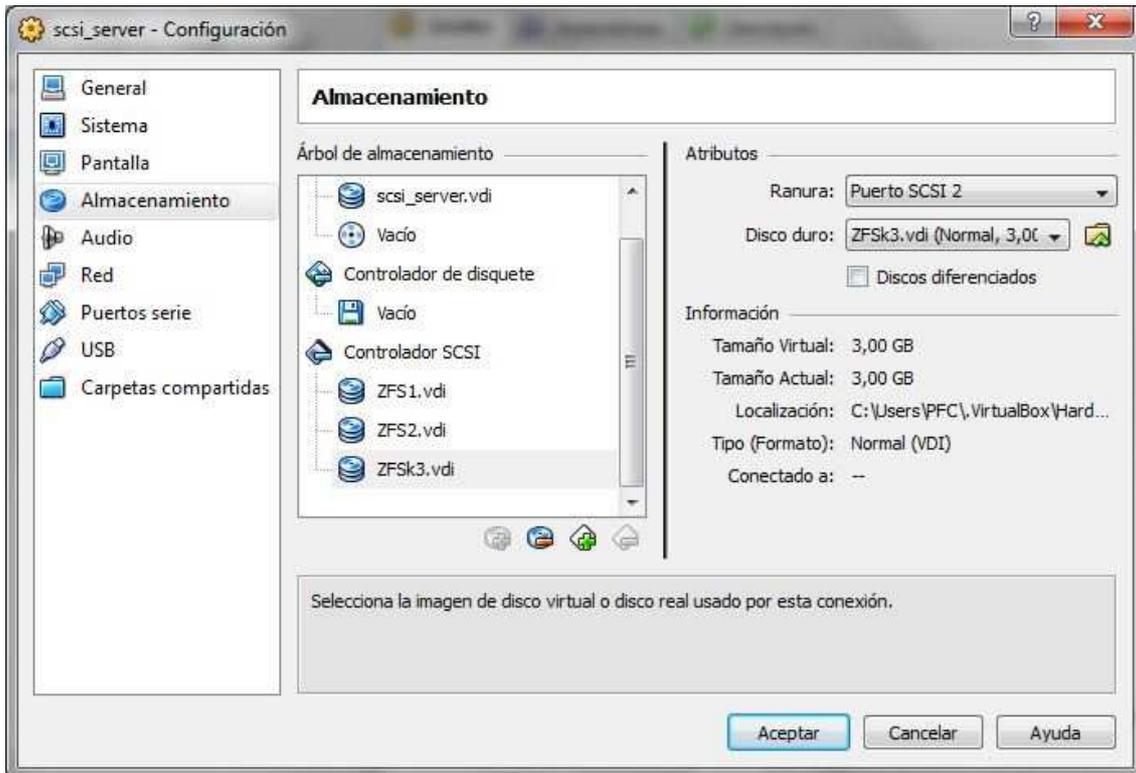
Ahora es necesario asignarlos a la máquina. Para ello debemos parar la máquina ya que virtualbox no nos permite asignarlos en caliente. Si ejecutamos format desde la línea de comandos podemos ver los discos físicos que el sistema está viendo asignados.



82. Solaris Format.

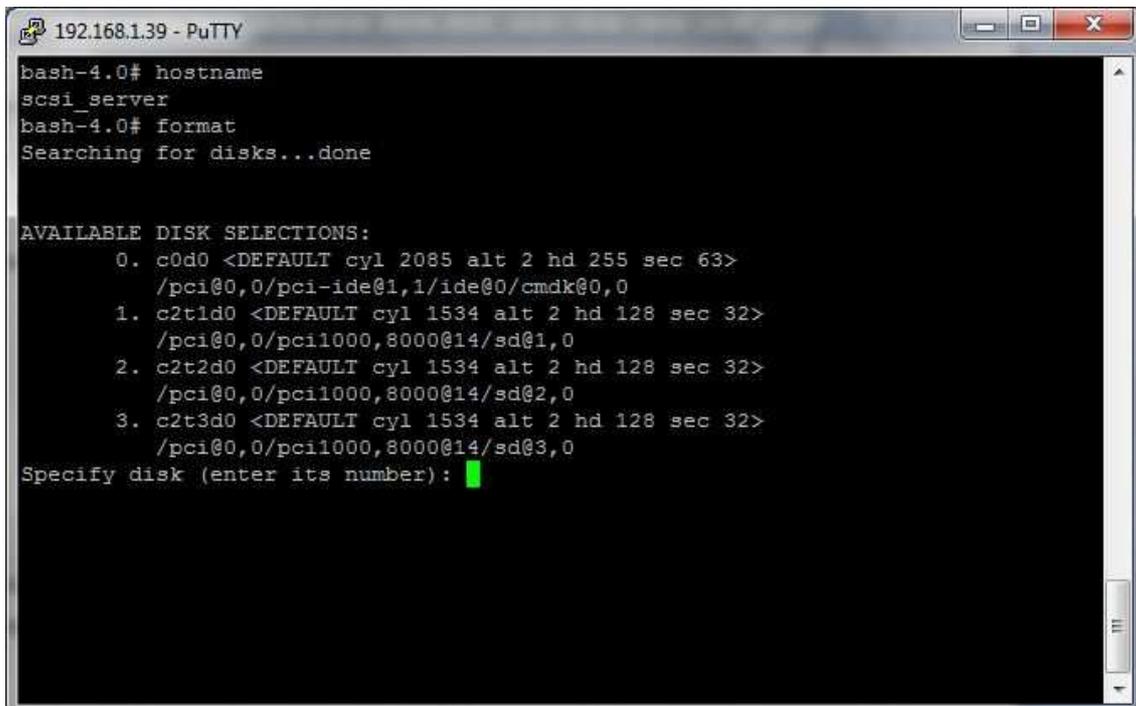
Paramos la máquina y una vez parada desde la ventana principal de VirtualBox, seleccionamos nuestra máquina virtual y pinchamos 2 veces sobre la opción

“almacenamiento”. Una vez estemos en este menú, añadimos una controladora SCSI y le añadiremos los 3 discos. Quedará una configuración como la de la siguiente imagen:



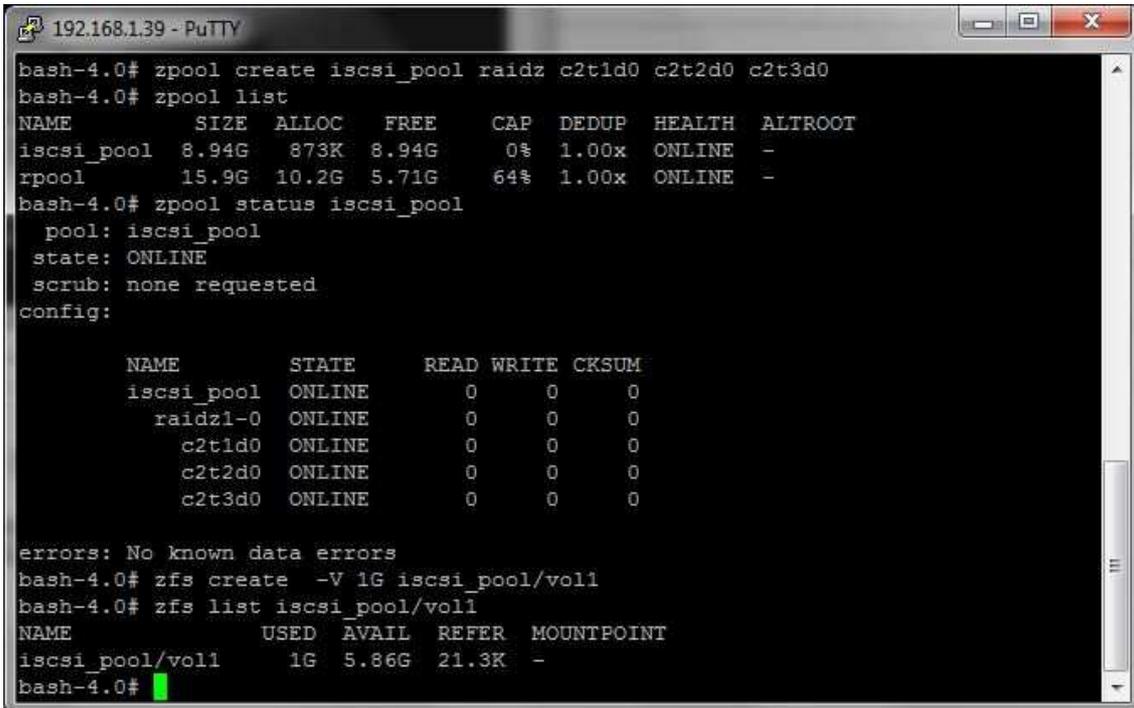
83. VB Añadir Controladora SCSI.

Ahora arrancamos de nuevo la máquina virtual y comprobamos que se ven 3 nuevos discos.



84. Solaris Añadir nuevos discos.

En la siguiente pantalla, creamos un nuevo pool ZFS utilizando raidz como tecnología de redundancia de datos. Una vez está el pool creado, se crea 1 volumen de tamaño 1 GB.



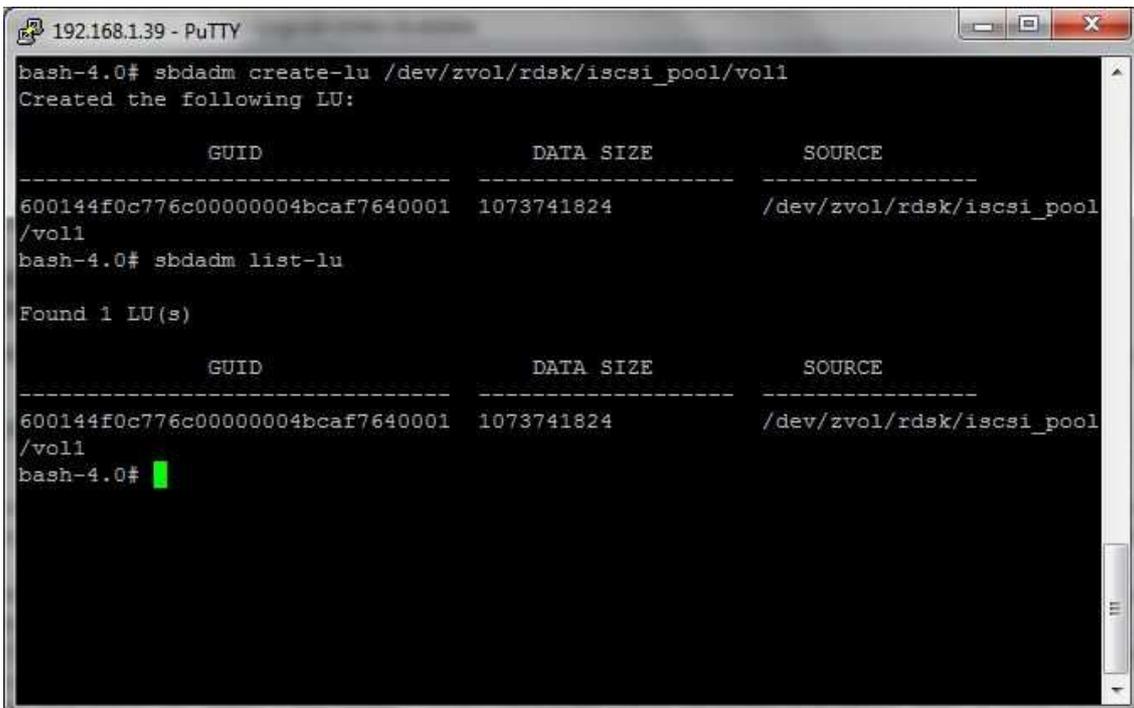
```
192.168.139 - PuTTY
bash-4.0# zpool create iscsi_pool raidz c2t1d0 c2t2d0 c2t3d0
bash-4.0# zpool list
NAME          SIZE  ALLOC  FREE   CAP  DEDUP  HEALTH  ALTROOT
iscsi_pool    8.94G  873K   8.94G   0%   1.00x  ONLINE  -
rpool        15.9G  10.2G  5.71G   64%   1.00x  ONLINE  -
bash-4.0# zpool status iscsi_pool
pool: iscsi_pool
state: ONLINE
scrub: none requested
config:

    NAME          STATE      READ  WRITE  CKSUM
    iscsi_pool    ONLINE          0     0     0
      raidz1-0    ONLINE          0     0     0
        c2t1d0    ONLINE          0     0     0
        c2t2d0    ONLINE          0     0     0
        c2t3d0    ONLINE          0     0     0

errors: No known data errors
bash-4.0# zfs create -V 1G iscsi_pool/voll
bash-4.0# zfs list iscsi_pool/voll
NAME          USED  AVAIL  REFER  MOUNTPOINT
iscsi_pool/voll  1G   5.86G  21.3K  -
bash-4.0#
```

85. Nuevo Zpool.

El siguiente paso consiste en dar visibilidad de la nueva LU creada al framework Comstar. Para ello primero hay que crear un nuevo scsi block device utilizando la utilidad sbdadm.



```
192.168.139 - PuTTY
bash-4.0# sbdadm create-lu /dev/zvol/rdisk/iscsi_pool/voll
Created the following LU:

-----
GUID                                DATA SIZE    SOURCE
-----
600144f0c776c00000004bcaf7640001  1073741824   /dev/zvol/rdisk/iscsi_pool
/voll
bash-4.0# sbdadm list-lu

Found 1 LU(s)

-----
GUID                                DATA SIZE    SOURCE
-----
600144f0c776c00000004bcaf7640001  1073741824   /dev/zvol/rdisk/iscsi_pool
/voll
bash-4.0#
```

86. Crear Vista LU.

En este momento tenemos un dispositivo de bloques scsi creado que necesitamos mapear para que pueda ser accesible por los hosts initiator (esto es obligatorio tanto para iSCSI como para Fibre Channel y FCoE).

Podemos realizar 2 tipos de mapeados :

- **Mapeado Simple:** Mediante este mapeado se exporta la LU a través de todos los puertos de todos los initiator, lo que hace que la LU sea visible para todos los initiator.
- **Mapeado Selectivo:** Mediante este mapeado se puede seleccionar qué hosts initiator tendrán visibilidad sobre la LU. Este tipo de mapeado requiere definir 3 parámetros:
 - **Host Groups:** Es el nombre que se da a un set de host initiator que tienen permitido el acceso a una misma LU.
 - **Target Groups:** Es el nombre que se da a un set de puertos target que exportar el mismo set de LU's al mismo set de Host Groups.
 - **LU View's:** Una vista es una capa creada entre la LU y el Host Initiator. Requiere que se indique Host Group, Target Group, número de LU y el LU ID.

Es importante no confundir un Target Group con un Target Portal Group (TPG). Un TPG es una lista de direcciones IP por las que escucha un iSCSI target.

Para realizar las pruebas del servidor de almacenamiento utilizaremos un mapeado simple de forma que podemos realizar pruebas con distintos servidores.

Para hacer la LU visible a todos los hosts simplemente es necesario obtener el GUID (Global Unique Identification) y crear una vista omitiendo el Host Group y el Target Group. De esta manera el sistema entiende que se quiere dar visibilidad a todos los host y define la LU mediante un mapeado simple.

```

192.168.1.39 - PuTTY
bash-4.0# sbdadm list-lu

Found 1 LU(s)

-----
GUID                                DATA SIZE                            SOURCE
-----
600144f0c776c00000004bcaf7640001  1073741824                            /dev/zvol/rdisk/iscsi_pool
/voll
bash-4.0# stmfadm list-lu -v
LU Name: 600144F0C776C00000004BCAF7640001
Operational Status: Online
Provider Name       : sbd
Alias               : /dev/zvol/rdisk/iscsi_pool/voll
View Entry Count   : 0
Data File          : /dev/zvol/rdisk/iscsi_pool/voll
Meta File          : not set
Size               : 1073741824
Block Size        : 512
Management URL    : not set
Vendor ID         : SUN
Product ID        : COMSTAR
Serial Num        : not set
Write Protect     : Disabled
Writeback Cache   : Enabled
Access State      : Active
bash-4.0# stmfadm add-view 600144f0c776c00000004bcaf7640001
bash-4.0# stmfadm list-lu -v
LU Name: 600144F0C776C00000004BCAF7640001
Operational Status: Online
Provider Name       : sbd
Alias               : /dev/zvol/rdisk/iscsi_pool/voll
View Entry Count   : 1
Data File          : /dev/zvol/rdisk/iscsi_pool/voll
Meta File          : not set
Size               : 1073741824
Block Size        : 512
Management URL    : not set
Vendor ID         : SUN
Product ID        : COMSTAR
Serial Num        : not set
Write Protect     : Disabled
Writeback Cache   : Enabled
Access State      : Active

```

87. Mapeado de LU.

El siguiente paso consisten en activar el servicio de iscsi target de Solaris y crear un iscsi target port. Es importante que el servicio iscsitgt esté desactivado para no interferir con Comstar. Para configurarlo en modo static discovery no es necesario realizar ningún paso adicional.

```

192.168.1.39 - PuTTY
bash-4.0# svcs -a | grep iscsi
disabled    13:48:54 svc:/network/iscsi/target:default
disabled    13:49:05 svc:/system/iscsitgt:default
online      13:49:19 svc:/network/iscsi/initiator:default
bash-4.0# svcadm enable /network/iscsi/target
bash-4.0# svcs -a | grep iscsi
disabled    13:49:05 svc:/system/iscsitgt:default
online      13:49:19 svc:/network/iscsi/initiator:default
online      15:43:52 svc:/network/iscsi/target:default
bash-4.0# itadm create-target
Target iqn.1986-03.com.sun:02:35e508ed-a490-6762-9c19-a5c50abc81ce successfully
created
bash-4.0# itadm list-target
TARGET NAME                                STATE    SESSIONS
iqn.1986-03.com.sun:02:35e508ed-a490-6762-9c19-a5c50abc81ce  online  0
bash-4.0#

```

88. Activar Servicio iSCSI tgt.

Por último tenemos que configurar el método de descubrimiento del target iSCSI.

Existen 3 métodos de descubrimiento de targets iSCSI:

- **Static Discovery:** Se utiliza una dirección estática para referirse al target. Este método es óptimo si el host iSCSI tiene un número pequeño de targets, o para restringir el acceso a un initiator.
- **Dynamic Discovery utilizando el iSNS:** Los targets se descubren interactuando con uno o más servidores iSNS. Este método se utiliza para descubrir targets utilizando una configuración lo más pequeña posible. Es obligatorio tener al menos un servidor iSNS en la red para poder utilizar este método.
- **SendTargets Discovery:** Permite especificar una Ip y un puerto para acceder al target.

Es importante no configurar sobre un target iSCSI los métodos static discovery y dynamic discovery , ya que pueden causar problemas de rendimiento.

4.3 Configuración servidor NFS

Para configurar el servidor de NFS sobre OpenSolaris es necesario antes de nada crear el fichero `/etc/defaultdomain` y reiniciar el servidor. Una vez el servidor haya reiniciado creamos un nuevo volumen. Para este proyecto hemos creado el volumen `iscsi_pool/nfs_vol1` con tamaño de 1G para exportarlo mediante el protocolo NFS.

Una vez hecho esto, procedemos a crear un FileSystem sobre el volumen creado y lo montamos sobre el punto de montaje elegido sobre el que colgarán todos los filesystem exportados. En este proyecto utilizaremos como directorio padre `/export`. Una vez esté el filesystem montado, procedemos a insertar una línea en el fichero `/etc/dfs/dfstab` en el cual indicamos lo que queremos exportar. Se arranca el servicio `nfs_server` (con la utilidad `svcadm` de opensolaris) y se ejecuta el comando `shareall`.

```
192.168.137 - PuTTY
bash-4.0# newfs /dev/zvol/rdisk/iscsi_pool/nfs_voll
newfs: construct a new file system /dev/zvol/rdisk/iscsi_pool/nfs_voll: (y/n)? y
Warning: 4130 sector(s) in last cylinder unallocated
/dev/zvol/rdisk/iscsi_pool/nfs_voll: 2097118 sectors in 342 cylinders of 48 t
racks, 128 sectors
1024.0MB in 25 cyl groups (14 c/g, 42.00MB/g, 20160 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 86176, 172320, 258464, 344608, 430752, 516896, 603040, 689184, 775328,
1292192, 1378336, 1464480, 1550624, 1636768, 1722912, 1809056, 1895200,
1981344, 2067488
bash-4.0# mount /dev/zvol/dsk/iscsi_pool/nfs_voll /export/share1
bash-4.0# vi /etc/dfs/dfstab
bash-4.0# grep nfs_vol /etc/dfs/dfstab
bash-4.0# grep share1 /etc/dfs/dfstab
share -F nfs -o anon=0,sec=sys,rw /export/share1
bash-4.0# svcadm enable nfs_server
svcadm: Pattern 'nfs_server' doesn't match any instances
bash-4.0# svcadm enable svc:/network/nfs/server:default
bash-4.0# shareall
bash-4.0# share
- /export/share1 anon=0,sec=sys,rw ""
bash-4.0#
```

89. Configurar NFS.

Para que el filesystem se monte en /export/share1 de forma persistente con los reinicios de la máquina es necesario añadir una entrada en el fichero vfstab.

4.4 Configuración servidor SMB

Configurando Samba en el servidor de almacenamiento podemos proveer servicios de ficheros y de impresoras para clientes Windows, aunque ya no es estrictamente necesario ya que es posible instalar clientes NFS en los clientes Windows.

La configuración del servidor samba se puede realizar de 2 maneras:

- Por la línea de comandos
- A través de la web de administración de samba, también conocida como “swat”.

Para este proyecto utilizaremos la configuración por línea de comandos debido a que en entornos de producción es muy común no tener terminales en los cuales levantar entornos gráficos y por lo tanto es necesario conocer la línea de comandos para poder trabajar en los servidores.

El primer paso consiste en crear el filesystem al igual que se ha realizado en pasos anteriores

```
192.168.134 - PuTTY
bash-4.0# zfs create iscsi_pool/samba
bash-4.0# mkdir /export/samba
bash-4.0# zfs set mountpoint=/export/samba iscsi_pool/samba
bash-4.0# df -h | grep samba
iscsi_pool/samba      5.9G   28K   2.9G    1%   /export/samba
bash-4.0#
```

90. Crear FS Samba .

Una vez tenemos el filesystem que se quiere exportar mediante Samba, procedemos a arrancar el servicio CIFS con la utilidad svcadm. El siguiente paso consiste en indicar al gestor de volúmenes (ZFS) que ese filesystem se va a exportar a través de samba y además crear un nombre para el workgroup.

```
192.168.135 - PuTTY
bash-4.0# svcadm enable -r smb/server
svcadm: svc:/milestone/network depends on svc:/network/physical, which has multiple instances.
bash-4.0# ps -ef | grep smb
root    556    511    0 19:48:08 ?                0:00 /usr/sfw/sbin/smbd -D
root    511     1    0 19:48:05 ?                0:00 /usr/sfw/sbin/smbd -D
bash-4.0# zfs set sharesmb=on iscsi_pool/samba
bash-4.0# sharemgr show -vp
default nfs=(
  /export/share1          nfs=(anon="0") nfs:sys=(rw="*")
)
zfs
  zfs/iscsi_pool/samba smb=(
    iscsi_pool_samba=/export/samba
)
bash-4.0# smbadm join -w solcifs
After joining solcifs the smb service will be restarted automatically.
Would you like to continue? [no]: yes
bash-4.0#
```

91. Configurar Samba .

Lo siguiente que se debe realizar es dar de alta el módulo PAM para la autenticación de usuarios. Para ello hay que añadir la siguiente línea al final del fichero /etc/pam.conf

Other password required pam_smb_passwd.so.1 nowarn

Una vez dado de alta el módulo PAM, debemos cambiar de nuevo la password del usuario root para que utilice este módulo y se genere una password que pueda ser utilizada tanto por Windows como por Solaris.

5. Estudio y Configuración de la parte cliente para utilizar iSCSI

En este capítulo se realizará el estudio e implementación de soluciones para que los distintos clientes existentes en un entorno de producción real puedan acceder a almacenamiento SAN mediante iSCSI.

Es importante destacar que en este proyecto no se pueden probar la solución adoptada sobre sistemas operativos no virtualizables (HP-UX, AIX) debido a que son sistemas operativos que sólo pueden correr sobre un hardware determinado y principalmente propietario.

5.1 Instalación sobre SO Windows

Para este proyecto se va a utilizar como cliente el sistema operativo Windows Xp Profesional 2009 SP3.

5.1.1 Estado actual de la parte iSCSI para Windows

En la actualidad para implementar un iscsi initiator en el mundo Windows, existe un driver oficial desarrollado por Microsoft que nos permite conectarnos con una red SAN a través de iSCSI. El driver se denomina “*Microsoft iSCSI Software Initiator*” y es compatible con arquitecturas x86 y x64(incluyendo Intel y Amd) y plataformas IA64.

El Microsoft iSCSI Software Initiator soporta como versión mínima para instalación Windows 2000 SP4 y es válido para Windows 2003 Server y Windows Xp. No se especifica que sea válido para Windows 7 ni para Windows server 2008 porque estas 2 versiones ya vienen con soporte para iSCSI incluido en la instalación.

Además del driver oficial de Microsoft existe una herramienta free llamada StartPort (de la empresa RocketDivision) que permite dotar al SO Windows de las siguientes capacidades:

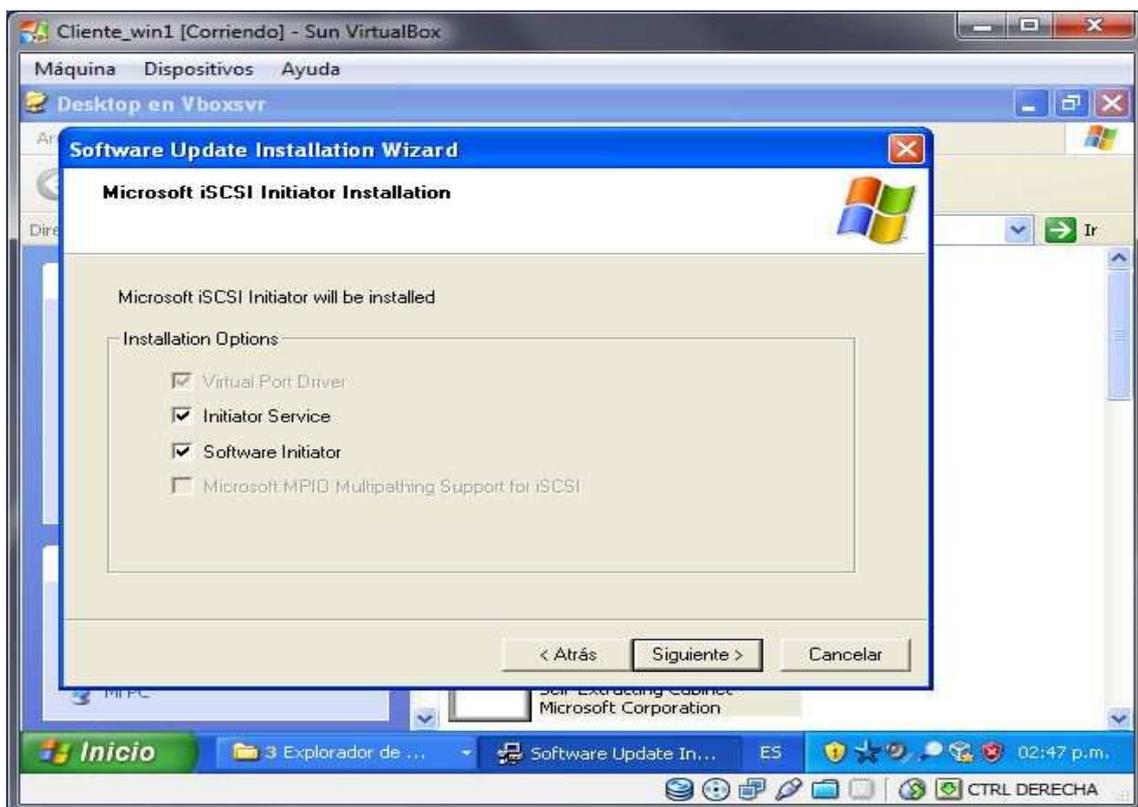
- iSCSI Initiator
- FcoE Initiator
- AoE Initiator

5.1.2 Instalación

Para instalar Microsoft iSCSi Software Initiator los pasos son similares a los de cualquier instalación sobre sistema operativo Windows.



92. Microsoft Initiator 1.



93. Microsoft Initiator 2.

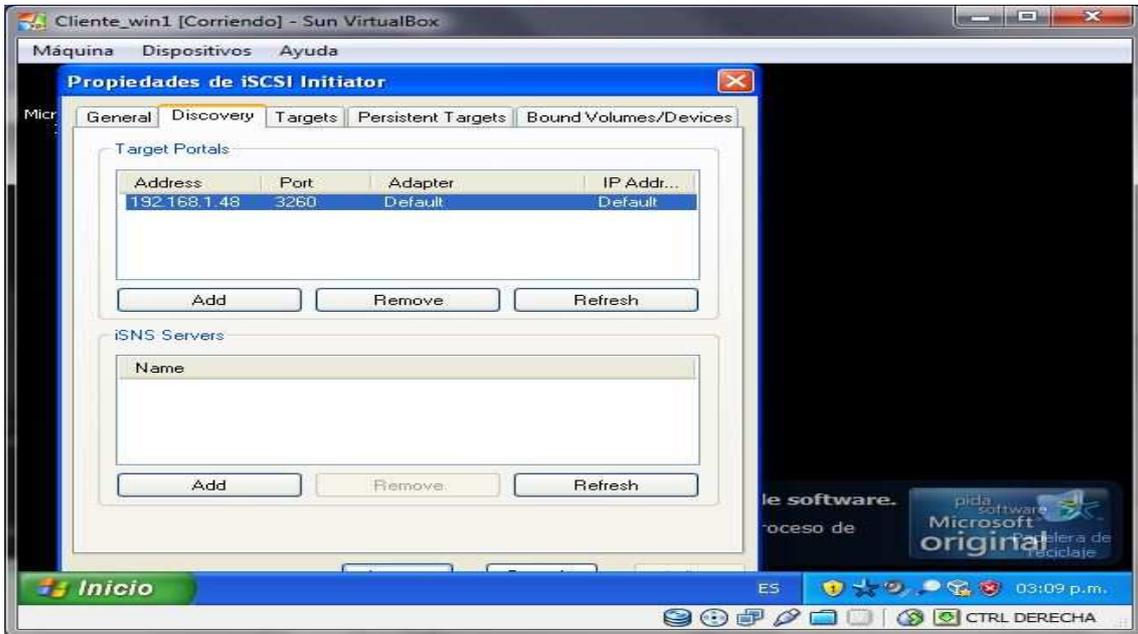
A partir de aquí la instalación es automática.

Para la instalación del software de NFS es necesario instalar el software de Microsoft “Windows Services for Unix” que se descarga de la página de Microsoft al igual que el software de iSCSI Initiator. Para este proyecto se ha instalado la versión 3.5.

La instalación de este software es totalmente automática. Sólo es necesario descomprimir el software e instalar pinchando en el icono “Setup”.

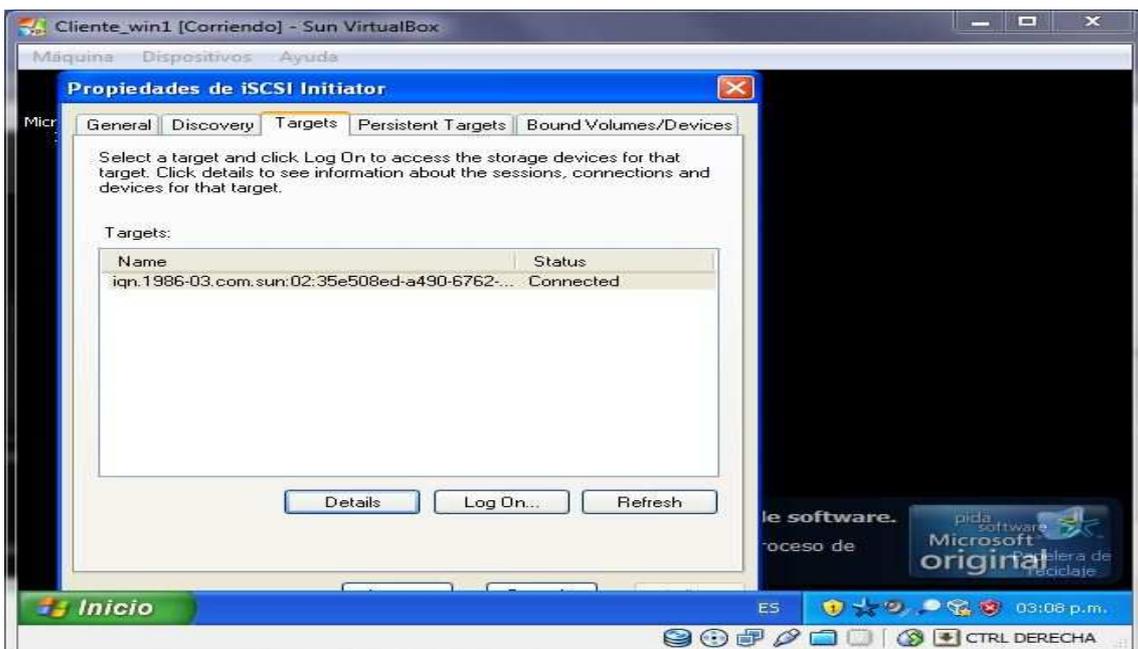
5.1.3 Pruebas

Para probar la conexión con una LUN iSCSI debemos configurar el acceso a la misma. Desde el panel de control se pincha en el icono iSCSI Initiator y en la pestaña discovery añadimos el target.



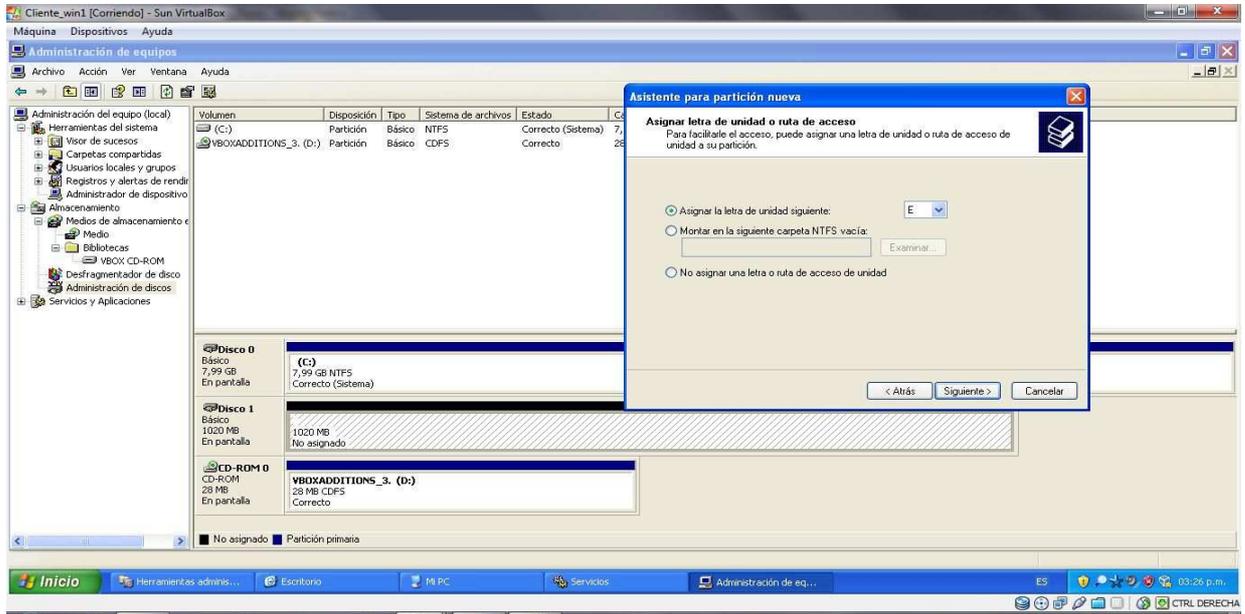
94. Configurar Win Initiator 1.

Una vez añadido el target, en la pestaña targets deberá aparecer el iqn accesible y se deberá pulsar el botón Log On para realizar el proceso de Login sobre la LUN.



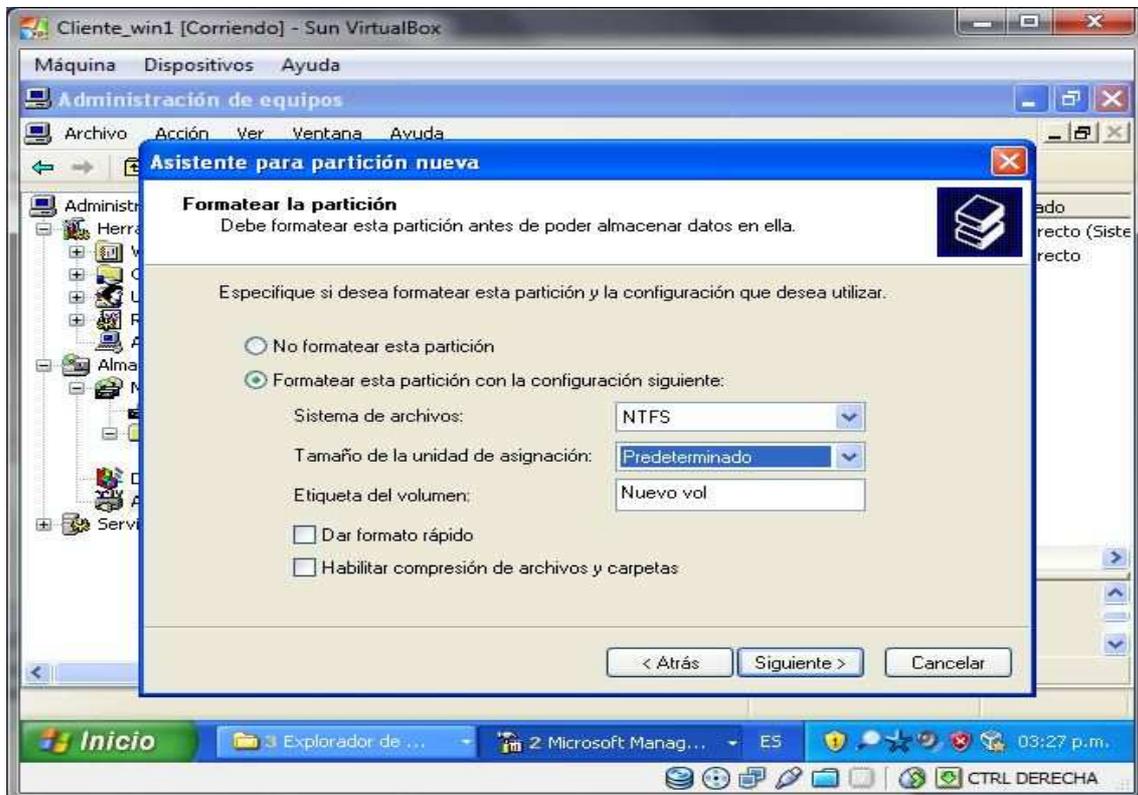
95. Configurar Win Initiator 2.

Una vez la Lun aparece en estado “Connected” se procede a crear la unidad en el sistema operativo. Para ello en el panel de control debemos ir al icono Herramientas Administrativas. Dentro de este subprograma debemos ir al icono Administración de Equipos y desplegar en el explorador de la izquierda el desplegable “Almacenamiento” y “Administración de Discos”.



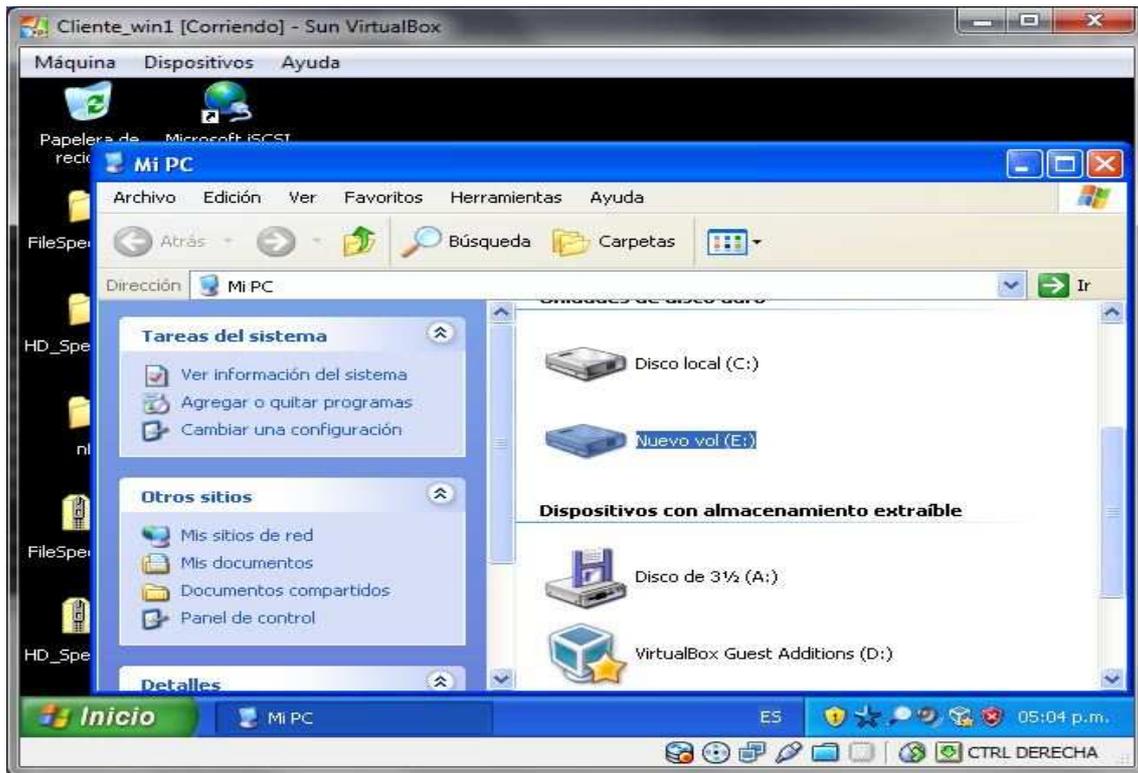
96. Configurar Win Initiator 3.

Ahora se particiona el disco para que el sistema operativo pueda verlo.



97. Configurar Win Initiator 4.

Y por último se comprueba que existe acceso a la LUN desde el icono “Mi Pc”.



98. Configurar Win Initiator 5.

Para la configuración de la unidad de Samba y de NFS, una vez instalados los Windows Services for Unix, se realiza de igual manera para ambas. Basta con abrir el icono “Mi Pc” e ir a la pestaña herramientas y dentro de ella “Conectar a Unidad de red”.

5.1.4 Conclusiones

La instalación del Software del iSCSI Initiator en Windows resulta sencilla ya que se realiza como cualquier instalación normal de software sobre este sistema operativo. Mediante una interfaz gráfica se eligen los componentes a instalar, se acepta la licencia y se instala. Es importante señalar que la instalación requiere un reinicio del sistema.

La parte de configuración iSCSI resulta algo más compleja. Resulta útil seguir algún manual para conseguir que el sistema operativo vea el disco ya que hay que realizar varios pasos (poco intuitivos) para conseguir que el sistema operativo descubra el disco. Una vez el disco está particionado se consigue el objetivo principal que es que para las aplicaciones sea transparente que es un disco iSCSI y que éstas puedan acceder a él como si fuese un disco local del sistema.

El resto de la configuración (Samba y NFS) es extremadamente sencilla una vez instalados los Windows Services for Unix (requiere reinicio del sistema) ya que se realizan las 2 de igual manera, indicando nombre del servidor que exporta el recurso o la IP y el nombre del recurso exportado. De esta manera ya aparece en Windows como una unidad de red.

5.2 Instalación sobre SO Linux

Para las pruebas sobre un cliente Linux se ha decidido utilizar Fedora, ya que es la versión libre de RedHat y por lo tanto las pruebas realizadas pueden ser extrapoladas a máquinas en producción corriendo bajo RedHat.

5.2.1 Estado actual de los proyectos para implementar inicators en Linux

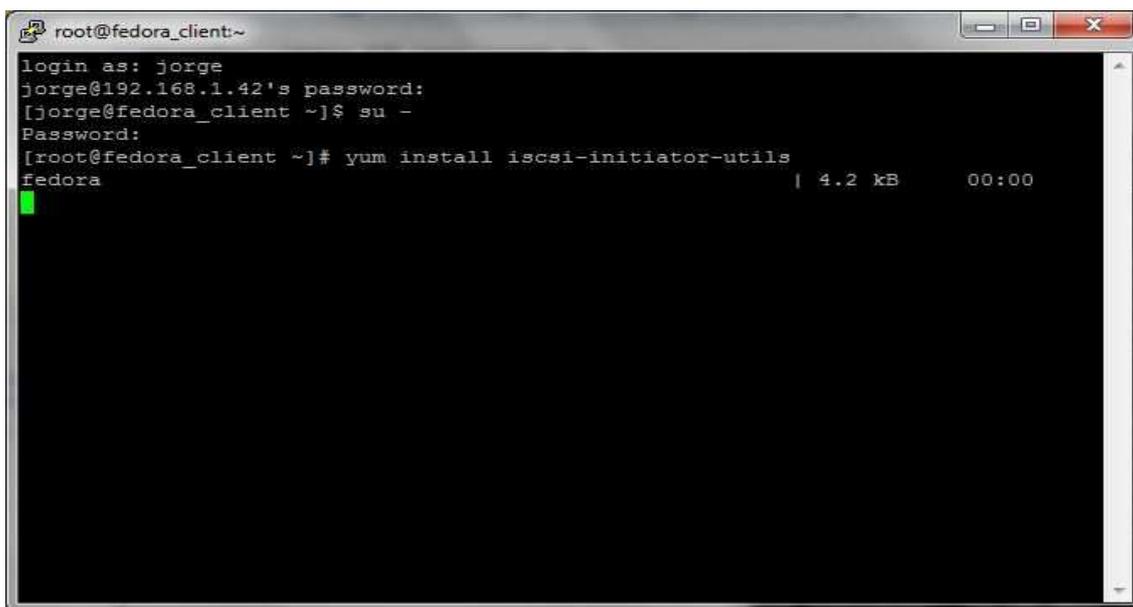
Los proyectos actuales para generar un initiator software se basan en el proyecto opensource open-iscsi.

En la propia página del proyecto hablan de la última release como “semi estable” y es la versión 2.0.871.

Para poder utilizar el iscsi initiator que implementa open-iscsi, es necesario que la versión de Linux utilizada utilice la versión de kernel 2.6.16 o superior. Las versiones del kernel 2.6.14 y 2.6.15 están parcialmente soportadas aunque con problemas conocidos y documentados en el propio paquete open-iscsi.

5.2.2 Instalación

Para la instalación del iSCSI Initiator en Linux se instalará el paquete iscsi-initiator-utils.



```
root@fedora_client:~  
login as: jorge  
jorge@192.168.1.42's password:  
[jorge@fedora_client ~]$ su -  
Password:  
[root@fedora_client ~]# yum install iscsi-initiator-utils  
fedora | 4.2 kB 00:00
```

99. Initiator Linux 1.

El siguiente paso consiste en poner la variable node.startup del fichero /etc/iscsi/iscsid.conf en automático.

```
root@fedora_client:~  
#####  
# open-iscsi can create a session and bind it to a NIC/HBA.  
# To set this up see the example iface config file.  
  
#####  
# Startup settings  
#####  
  
# To request that the iscsi initd scripts startup a session set to "automatic".  
node.startup = automatic  
#  
# To manually startup the session set to "manual". The default is automatic.  
node.startup = automatic  
  
# *****  
# CHAP Settings  
# *****  
  
# To enable CHAP authentication set node.session.auth.authmethod  
# to CHAP. The default is None.  
#node.session.auth.authmethod = CHAP  
  
# To set a CHAP username and password for initiator
```

100. Initiator Linux 2.

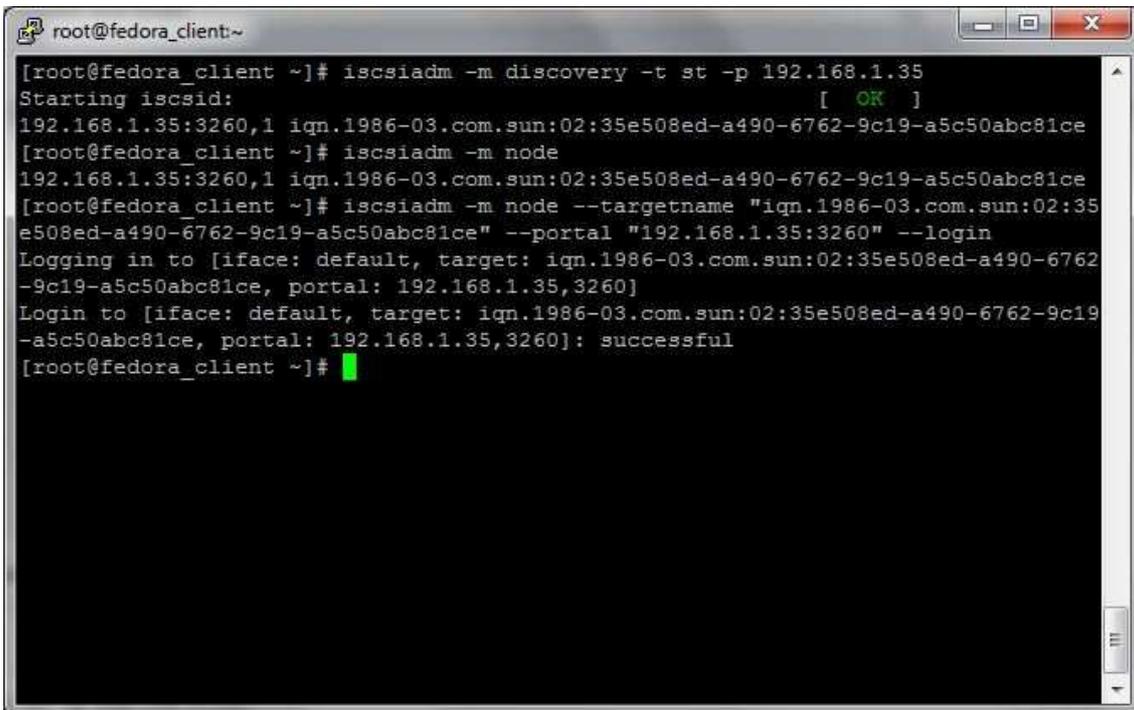
Para la parte de NFS es necesario instalar el paquete nfs-utils. Una vez instalado es necesario arrancar el servicio rpcbind y el nfs statd (servicio nfslock).

```
root@fedora_client:~  
[root@fedora_client ~]# service rpcbind start  
Starting rpcbind: [ OK ]  
[root@fedora_client ~]# service nfslock start  
Starting NFS statd: [ OK ]  
[root@fedora_client ~]# rpcinfo -p  
program vers proto port service  
100000 4 tcp 111 portmapper  
100000 3 tcp 111 portmapper  
100000 2 tcp 111 portmapper  
100000 4 udp 111 portmapper  
100000 3 udp 111 portmapper  
100000 2 udp 111 portmapper  
100024 1 udp 34694 status  
100024 1 tcp 46453 status  
[root@fedora_client ~]#
```

101. NFS Linux 1.

5.2.3 Pruebas

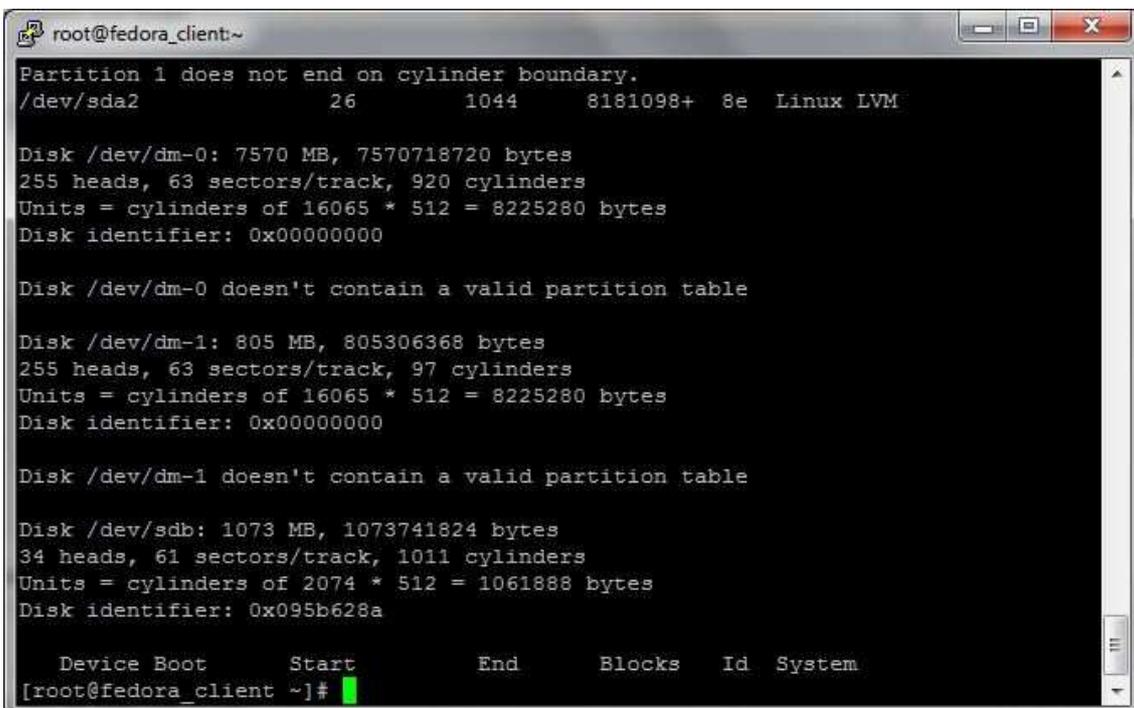
Para la parte iSCSI, una vez instalado el paquete con las utilidades el siguiente paso consiste en descubrir el target y realizar Login en al LUN.



```
root@fedora_client~  
[root@fedora_client ~]# iscsiadm -m discovery -t st -p 192.168.1.35  
Starting iscsid: [ OK ]  
192.168.1.35:3260,1 iqn.1986-03.com.sun:02:35e508ed-a490-6762-9c19-a5c50abc81ce  
[root@fedora_client ~]# iscsiadm -m node  
192.168.1.35:3260,1 iqn.1986-03.com.sun:02:35e508ed-a490-6762-9c19-a5c50abc81ce  
[root@fedora_client ~]# iscsiadm -m node --targetname "iqn.1986-03.com.sun:02:35e508ed-a490-6762-9c19-a5c50abc81ce" --portal "192.168.1.35:3260" --login  
Logging in to [iface: default, target: iqn.1986-03.com.sun:02:35e508ed-a490-6762-9c19-a5c50abc81ce, portal: 192.168.1.35,3260]  
Login to [iface: default, target: iqn.1986-03.com.sun:02:35e508ed-a490-6762-9c19-a5c50abc81ce, portal: 192.168.1.35,3260]: successful  
[root@fedora_client ~]#
```

102. Initiator Linux 3.

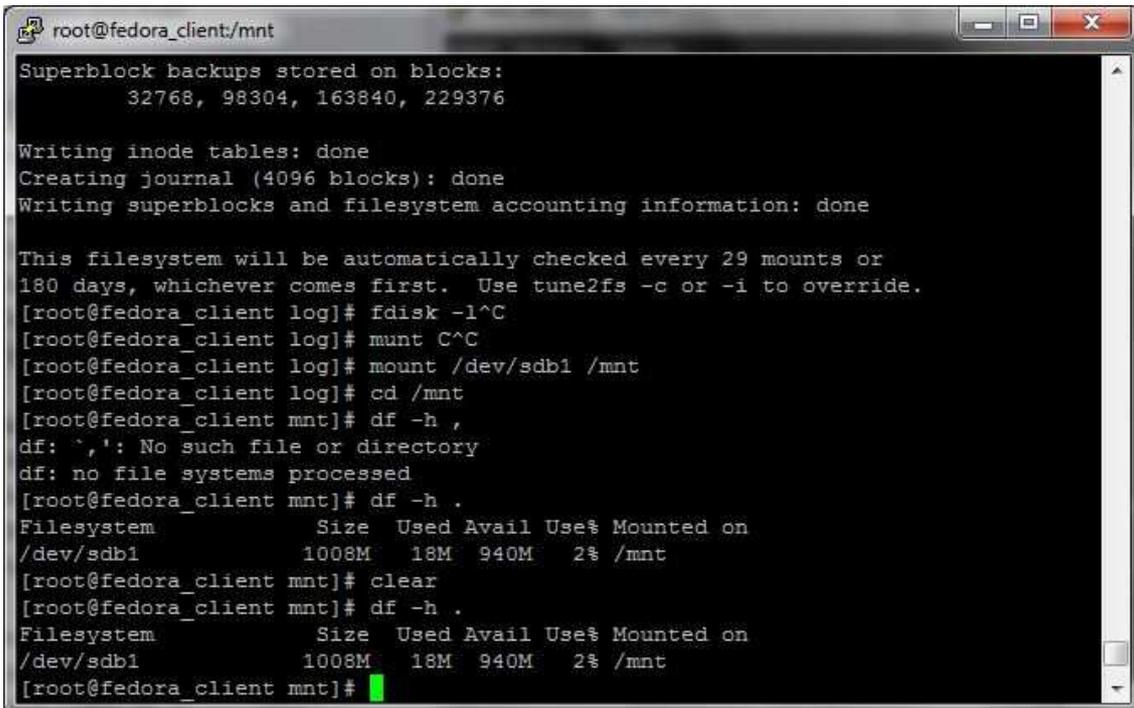
Una vez el proceso de Login sobre la Lun termina satisfactoriamente el sistema ya debería ver un nuevo disco. Para verlo utilizamos el comando fdisk.



```
root@fedora_client~  
Partition 1 does not end on cylinder boundary.  
/dev/sda2          26          1044          8181098+   8e   Linux LVM  
  
Disk /dev/dm-0: 7570 MB, 7570718720 bytes  
255 heads, 63 sectors/track, 920 cylinders  
Units = cylinders of 16065 * 512 = 8225280 bytes  
Disk identifier: 0x00000000  
  
Disk /dev/dm-0 doesn't contain a valid partition table  
  
Disk /dev/dm-1: 805 MB, 805306368 bytes  
255 heads, 63 sectors/track, 97 cylinders  
Units = cylinders of 16065 * 512 = 8225280 bytes  
Disk identifier: 0x00000000  
  
Disk /dev/dm-1 doesn't contain a valid partition table  
  
Disk /dev/sdb: 1073 MB, 1073741824 bytes  
34 heads, 61 sectors/track, 1011 cylinders  
Units = cylinders of 2074 * 512 = 1061888 bytes  
Disk identifier: 0x095b628a  
  
Device Boot      Start          End      Blocks   Id  System  
[root@fedora_client ~]#
```

103. Initiator Linux 4.

Como el sistema ya ve el disco, podemos formatearlo para crear un filesystem y montarlo en el sistema.



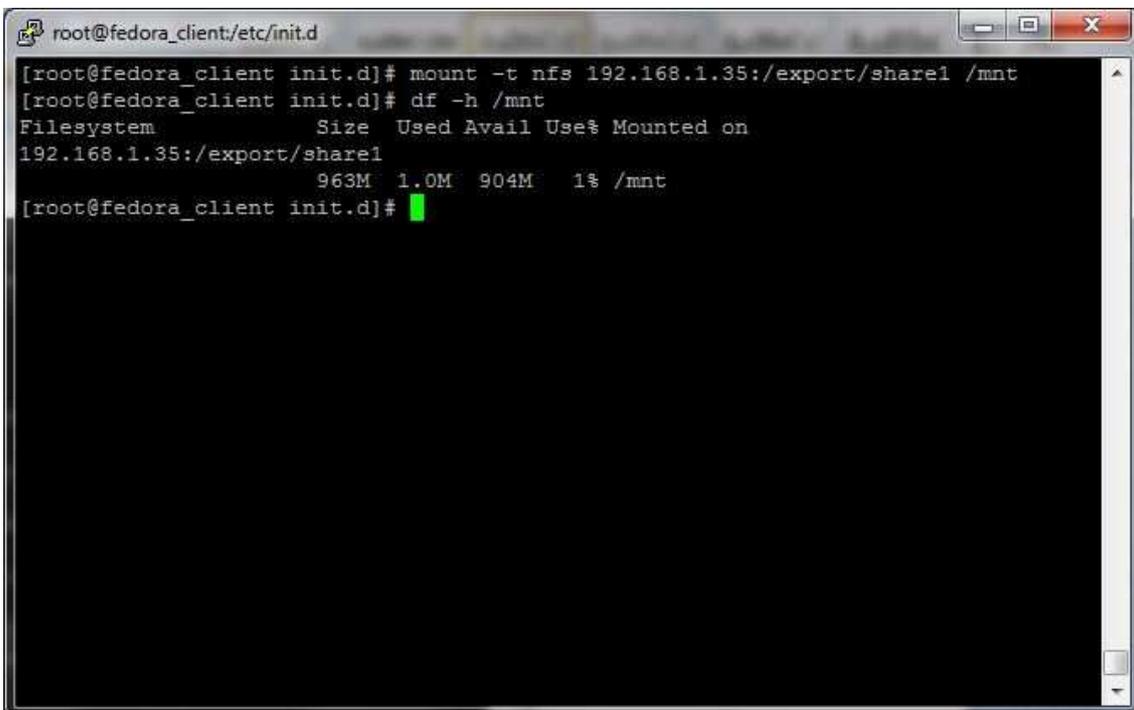
```
root@fedora_client/mnt
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 29 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@fedora_client log]# fdisk -l^C
[root@fedora_client log]# munt C^C
[root@fedora_client log]# mount /dev/sdb1 /mnt
[root@fedora_client log]# cd /mnt
[root@fedora_client mnt]# df -h ,
df: `,': No such file or directory
df: no file systems processed
[root@fedora_client mnt]# df -h .
Filesystem                Size      Used Avail Use% Mounted on
/dev/sdb1                  1008M    18M   940M   2% /mnt
[root@fedora_client mnt]# clear
[root@fedora_client mnt]# df -h .
Filesystem                Size      Used Avail Use% Mounted on
/dev/sdb1                  1008M    18M   940M   2% /mnt
[root@fedora_client mnt]#
```

104. Initiator Linux 6.

La parte NFS no tiene ninguna complejidad. Simplemente dándole al comando mount el tipo de filesystem, el nombre del host y el recurso exportado es posible montarlo en el sistema.



```
root@fedora_client/etc/init.d
[root@fedora_client init.d]# mount -t nfs 192.168.1.35:/export/share1 /mnt
[root@fedora_client init.d]# df -h /mnt
Filesystem                Size      Used Avail Use% Mounted on
192.168.1.35:/export/share1
                             963M    1.0M   904M   1% /mnt
[root@fedora_client init.d]#
```

105. NFS Linux 2.

5.2.4 Conclusiones

El proyecto open source para generar el iscsi initiator en Linux es sencillo de instalar. No requiere una configuración complicada (basta con descomentar una línea en un fichero de configuración) y permite comenzar a trabajar con almacenamiento iSCSI de forma rápida, casi inmediata. Además, al contrario que el software para Windows, no requiere de un reinicio del sistema tras su instalación lo que evita tener tiempos de parada en entornos de producción y aumentar la disponibilidad del sistema (medida muy importante en los entornos productivos).

La instalación y configuración de NFS es igual de sencilla. Tampoco requiere de reinicio tras su instalación y el montaje de recursos exportados por un servidor es inmediato.

5.3 Instalación sobre SO OpenSolaris

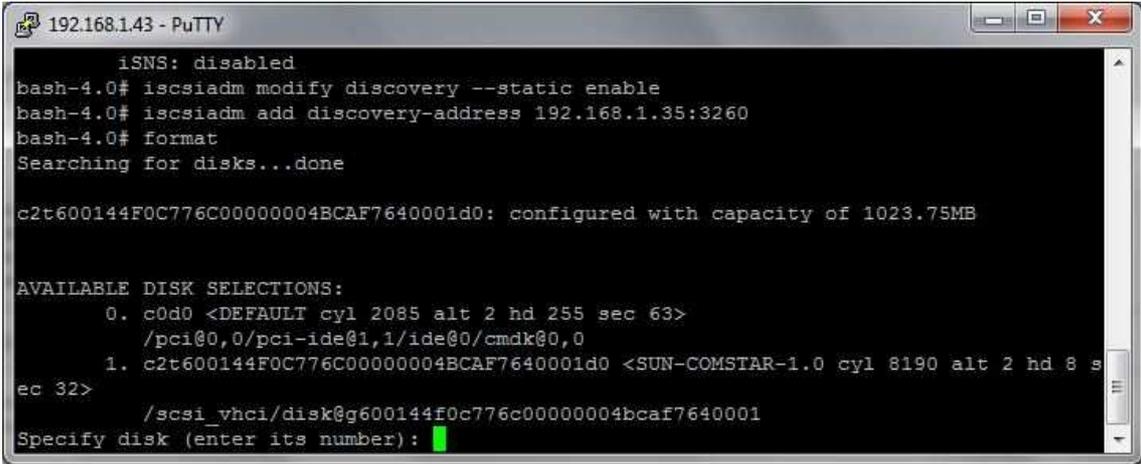
Para implementar un initiator sobre sistema operativo Solaris, utilizaremos Comstar (al igual que la implementación del target).

5.3.1 Instalación

La instalación de comstar se elige en la fase de instalación del sistema operativo donde se decide si se instala la distribución entera del sistema operativo o personalizar la instalación con los componentes que se deseen. Aún así, si en la fase de instalación no se hubiera seleccionado comstar, se pueden descargar los paquetes desde los repositorios de open solaris e instalarlos con el comando pkg.

5.3.2 Pruebas

Para configurar el initiator sobre Opensolaris primero debemos activar los mensajes de descubrimiento, y después darle la IP del servidor iSCSI.



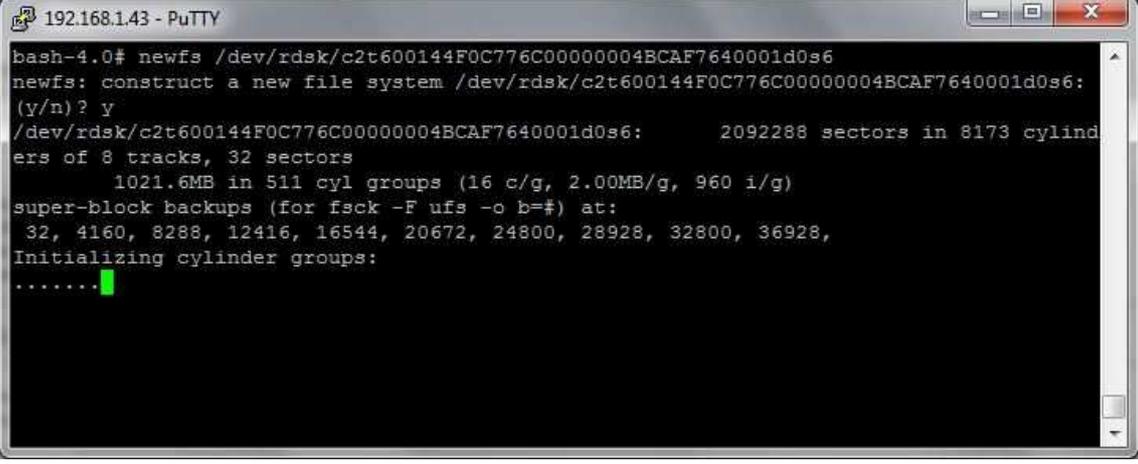
```
192.168.1.43 - PuTTY
iSNS: disabled
bash-4.0# iscsiadm modify discovery --static enable
bash-4.0# iscsiadm add discovery-address 192.168.1.35:3260
bash-4.0# format
Searching for disks...done

c2t600144F0C776C00000004BCAF7640001d0: configured with capacity of 1023.75MB

AVAILABLE DISK SELECTIONS:
  0. c0d0 <DEFAULT cyl 2085 alt 2 hd 255 sec 63>
    /pci@0,0/pci-ide@1,1/ide@0/cmdk@0,0
  1. c2t600144F0C776C00000004BCAF7640001d0 <SUN-COMSTAR-1.0 cyl 8190 alt 2 hd 8 s
    ec 32>
    /scsi_vhci/disk@g600144f0c776c00000004bcaf7640001
Specify disk (enter its number): █
```

106. Opensolaris Initiator 1.

Una vez el sistema ya ve el disco se puede crear un filesystem para montarlo en el sistema.



```
192.168.143 - PuTTY
bash-4.0# newfs /dev/rdisk/c2t600144F0C776C00000004BCAF7640001d0s6
newfs: construct a new file system /dev/rdisk/c2t600144F0C776C00000004BCAF7640001d0s6:
(y/n)? y
/dev/rdisk/c2t600144F0C776C00000004BCAF7640001d0s6:    2092288 sectors in 8173 cylinders of 8 tracks, 32 sectors
    1021.6MB in 511 cyl groups (16 c/g, 2.00MB/g, 960 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
    32, 4160, 8288, 12416, 16544, 20672, 24800, 28928, 32800, 36928,
Initializing cylinder groups:
.....█
```

107. Opensolaris 2.

5.3.3 Conclusiones

La instalación en Opensolaris no es necesaria pues viene incluida en la distribución del sistema operativo. Esto hace que no sea necesario instalar ningún componente adicional para poder beneficiarse del almacenamiento externo.

Con respecto a la configuración del dispositivo, también resulta muy sencilla. Además, existe gran cantidad de documentación en la página de la comunidad de desarrolladores que sirve de guía para las distintas configuraciones que queramos o necesitemos implementar.

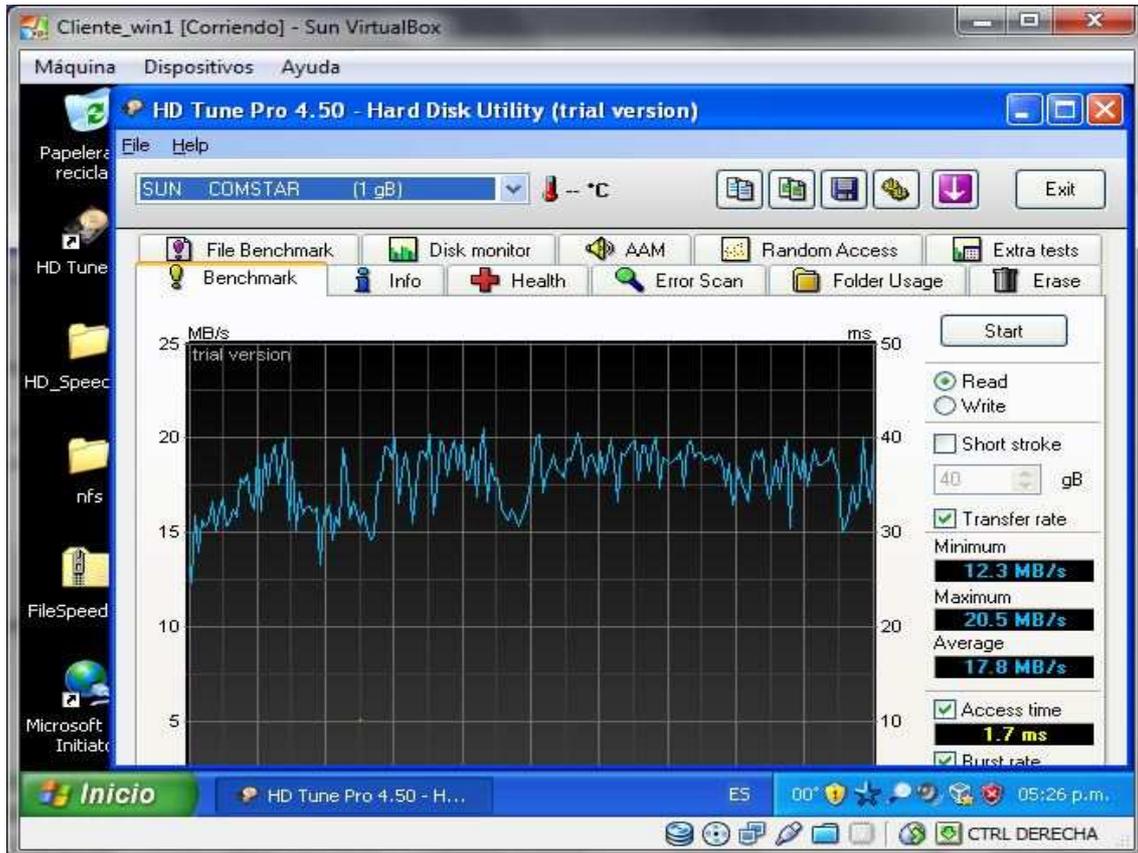
El descubrimiento de los dispositivos presentados por el target se realiza una vez indicada la ip y ejecutando el comando devfsadm (igual que se hace en solaris con discos de san) lo que hace que su administración sea muy sencilla (no son necesarios conocimientos adicionales para la implementación y administración).

6. Performance de los sistemas

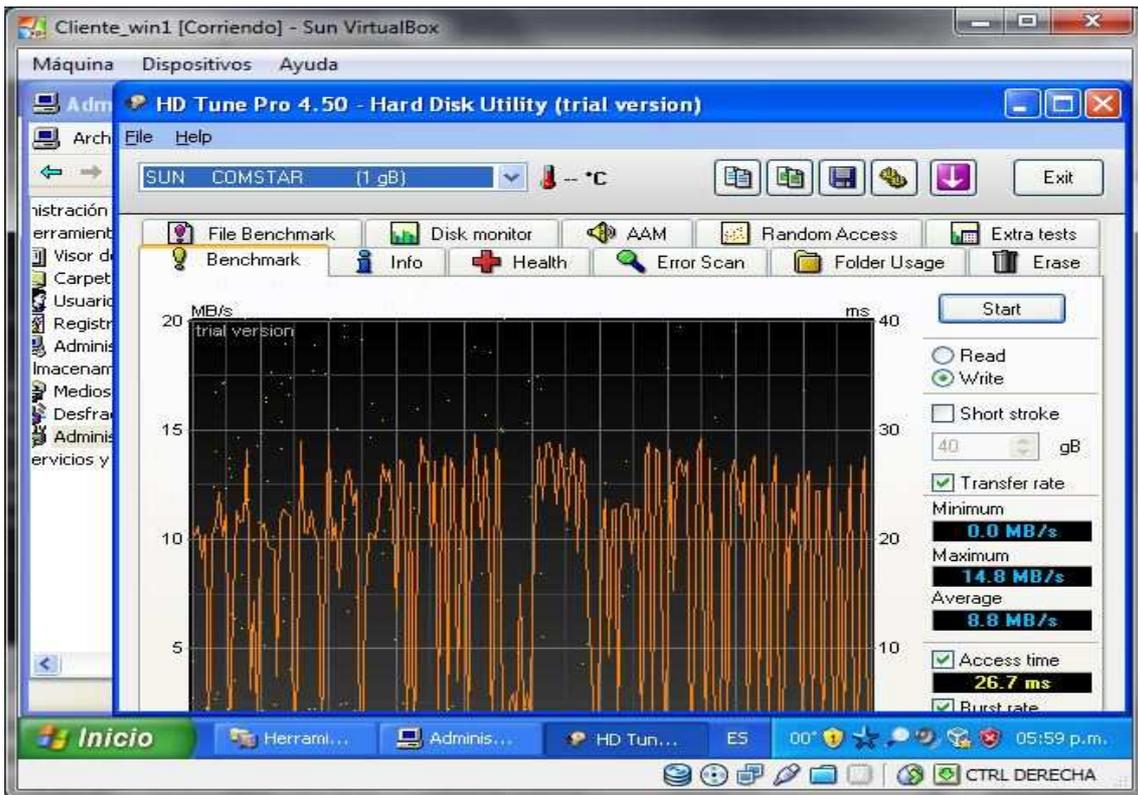
6.1 Performance de acceso desde cliente Windows

Con el sistema operativo Windows realizaremos las pruebas utilizando protocolos NFS, Samba e iSCSI.

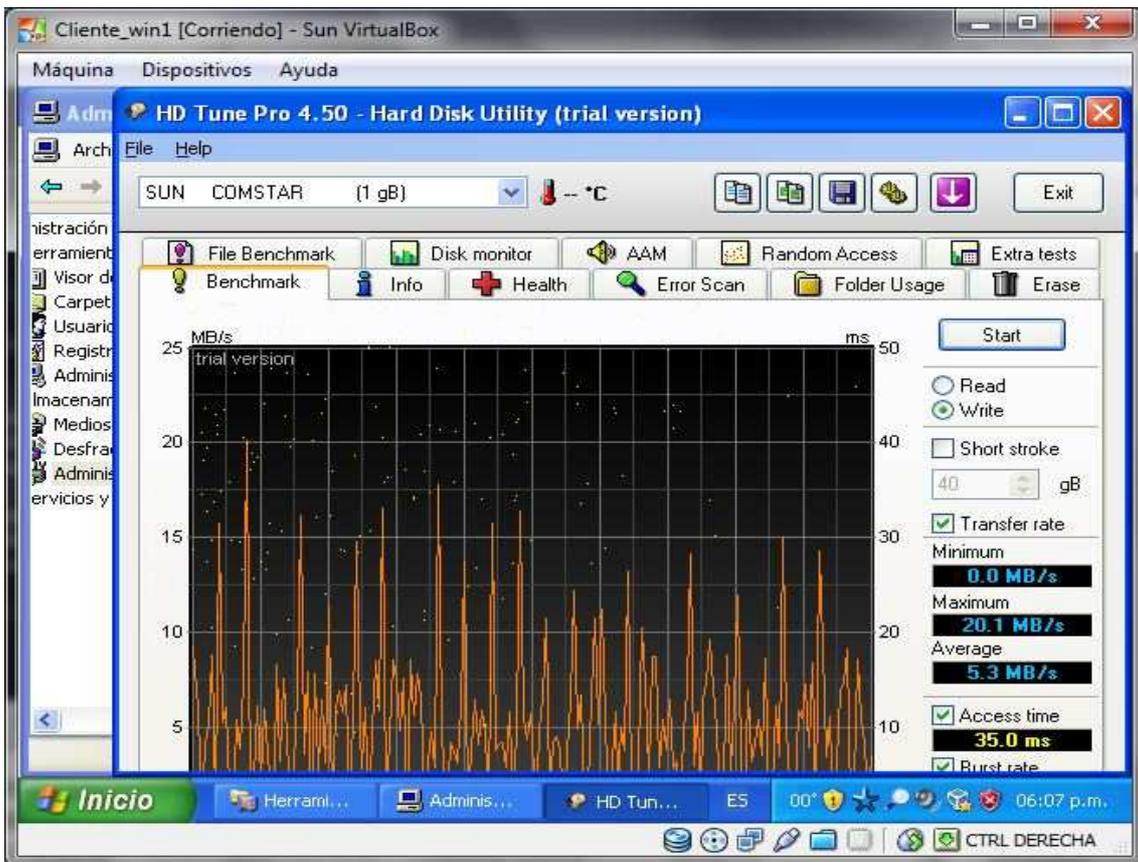
Las primeras pruebas las vamos a realizar con iSCSI y variando el tamaño de bloque.



108. Performance tamaño bloque automático.

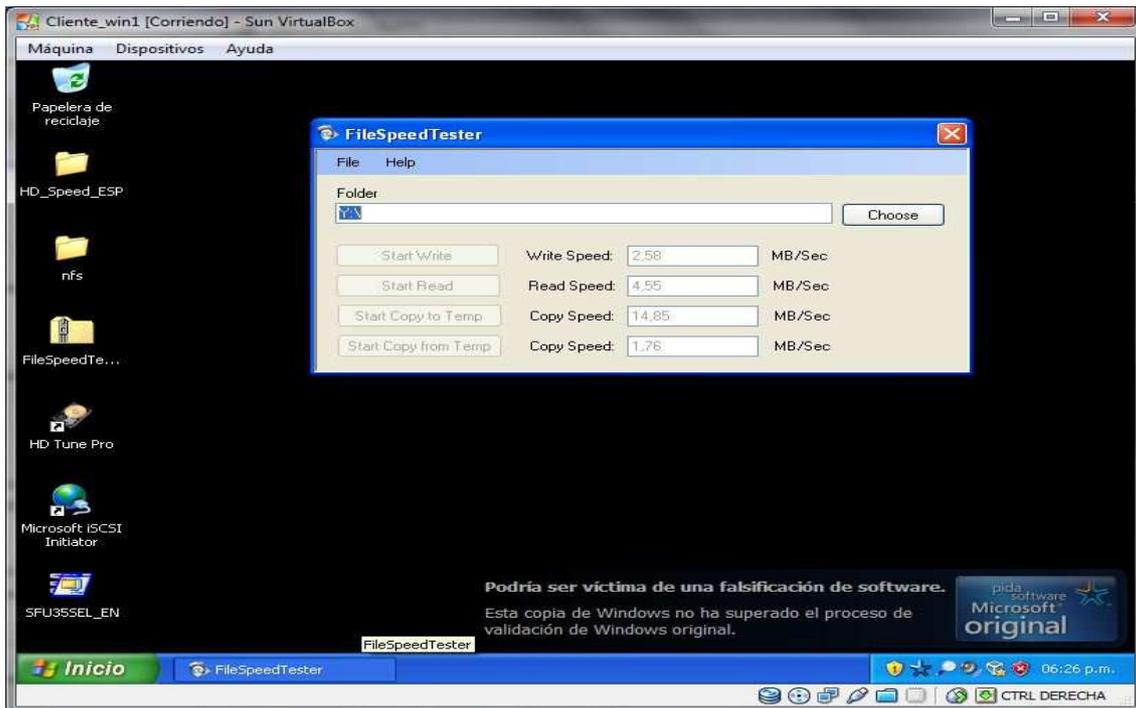


109. Performance tamaño bloque 512KB.



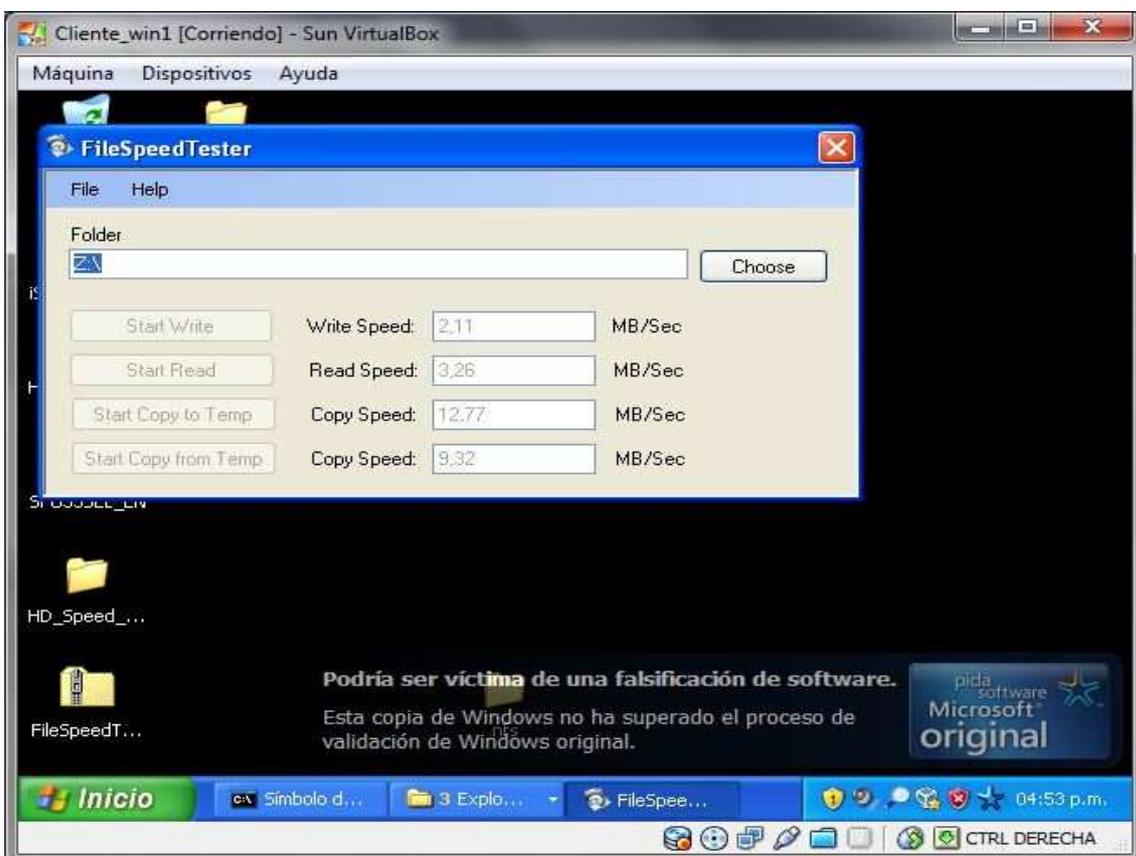
110. Performance tamaño bloque 1 KB.

La siguiente prueba que realizaremos es sobre la unidad NFS.



111. Performance NFS.

Y por último realizamos la prueba sobre la unidad montada utilizando Samba (CIFS).



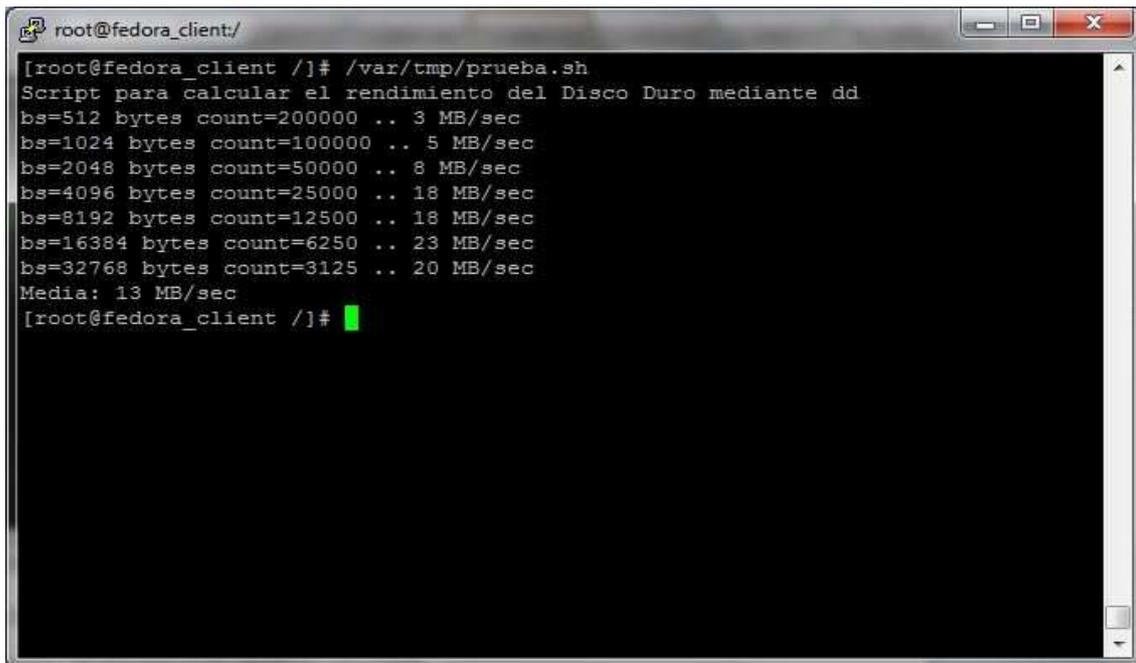
112. Performance Samba.

6.2 Performance de acceso desde cliente Linux

Para realizar las pruebas sobre Linux solo utilizaremos NFS e iSCSI, ya que samba está orientado principalmente a exportar recursos para sistemas Windows (aunque tiene más funcionales aparte de esta).

Para realizar estas pruebas se ha realizado un script que mediante el comando dd, realiza 2 pruebas por cada tamaño de bloque indicado. Al final, calcula la media de velocidad.

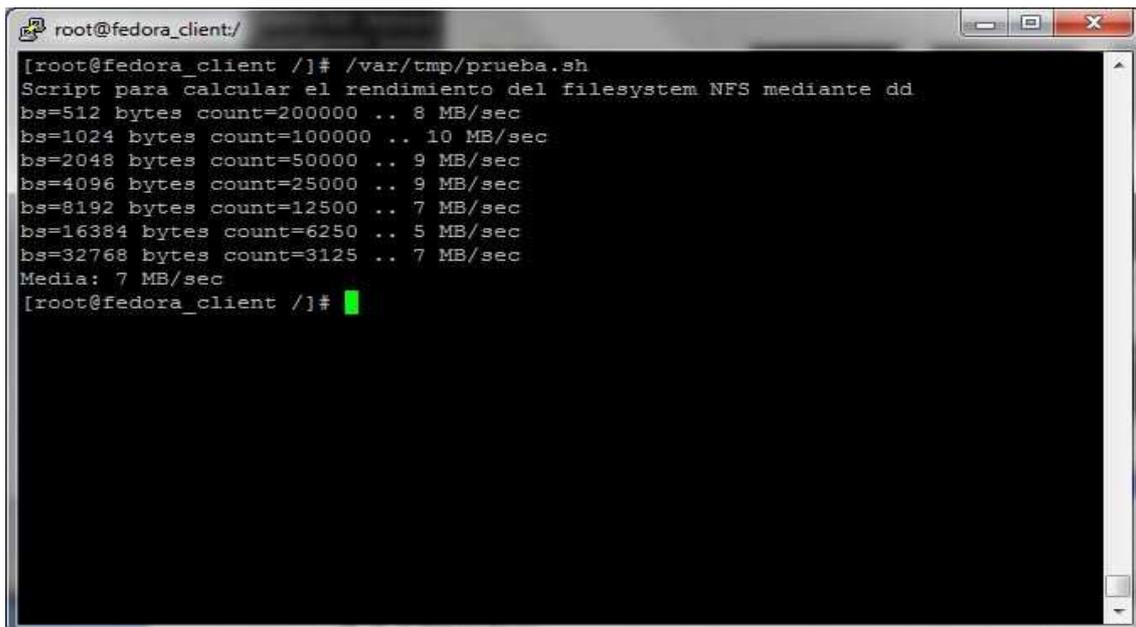
En primer lugar comenzamos con la prueba sobre el disco iSCSI.



```
root@fedora_client/
[root@fedora_client /]# /var/tmp/prueba.sh
Script para calcular el rendimiento del Disco Duro mediante dd
bs=512 bytes count=200000 .. 3 MB/sec
bs=1024 bytes count=100000 .. 5 MB/sec
bs=2048 bytes count=50000 .. 8 MB/sec
bs=4096 bytes count=25000 .. 18 MB/sec
bs=8192 bytes count=12500 .. 18 MB/sec
bs=16384 bytes count=6250 .. 23 MB/sec
bs=32768 bytes count=3125 .. 20 MB/sec
Media: 13 MB/sec
[root@fedora_client /]#
```

113. Performance iSCSI.

A continuación realizamos la prueba sobre el filesystem montado a través de NFS.



```
root@fedora_client/
[root@fedora_client /]# /var/tmp/prueba.sh
Script para calcular el rendimiento del filesystem NFS mediante dd
bs=512 bytes count=200000 .. 8 MB/sec
bs=1024 bytes count=100000 .. 10 MB/sec
bs=2048 bytes count=50000 .. 9 MB/sec
bs=4096 bytes count=25000 .. 9 MB/sec
bs=8192 bytes count=12500 .. 7 MB/sec
bs=16384 bytes count=6250 .. 5 MB/sec
bs=32768 bytes count=3125 .. 7 MB/sec
Media: 7 MB/sec
[root@fedora_client /]#
```

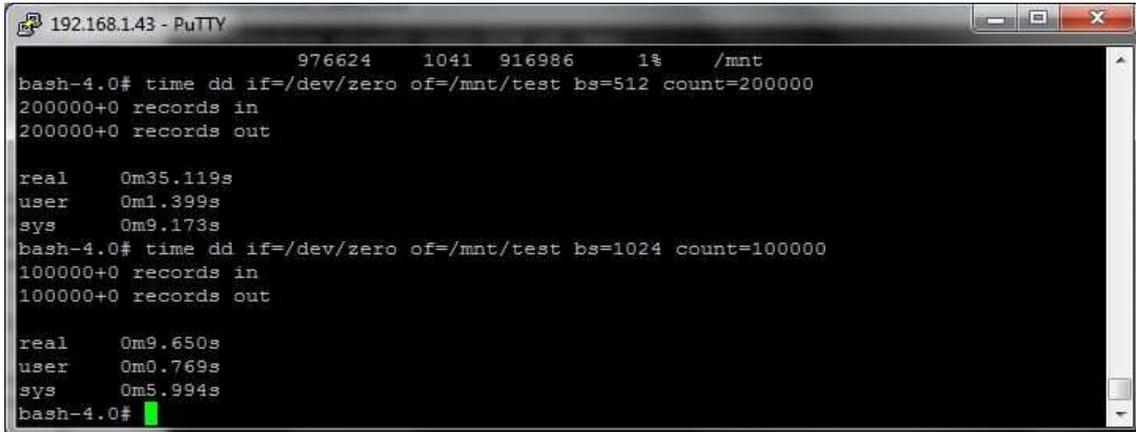
114. Performance NFS.

6.3 Performance de acceso desde cliente OpenSolaris

Para OpenSolaris al igual que para Linux, solo realizaremos las pruebas sobre protocolo NFS e iSCSI.

El comando dd en solaris no devuelve la velocidad de escritura en Mb, por lo que utilizamos dd junto con el comando time para conseguir la velocidad de transferencia.

Comenzaremos con las pruebas sobre iSCSI.

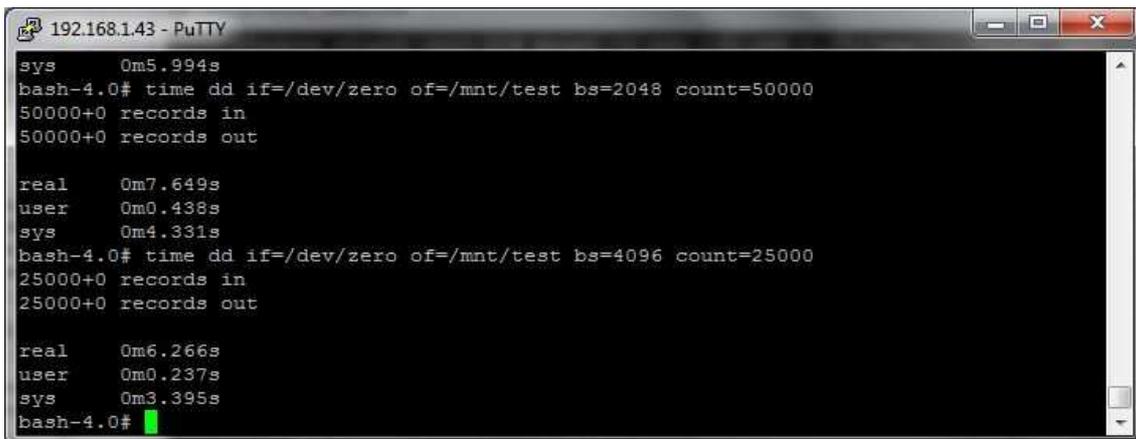


```
192.168.1.43 - PuTTY
976624 1041 916986 1# /mnt
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=512 count=200000
200000+0 records in
200000+0 records out

real    0m35.119s
user    0m1.399s
sys     0m9.173s
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=1024 count=100000
100000+0 records in
100000+0 records out

real    0m9.650s
user    0m0.769s
sys     0m5.994s
bash-4.0#
```

115. Performance iSCSI 1.

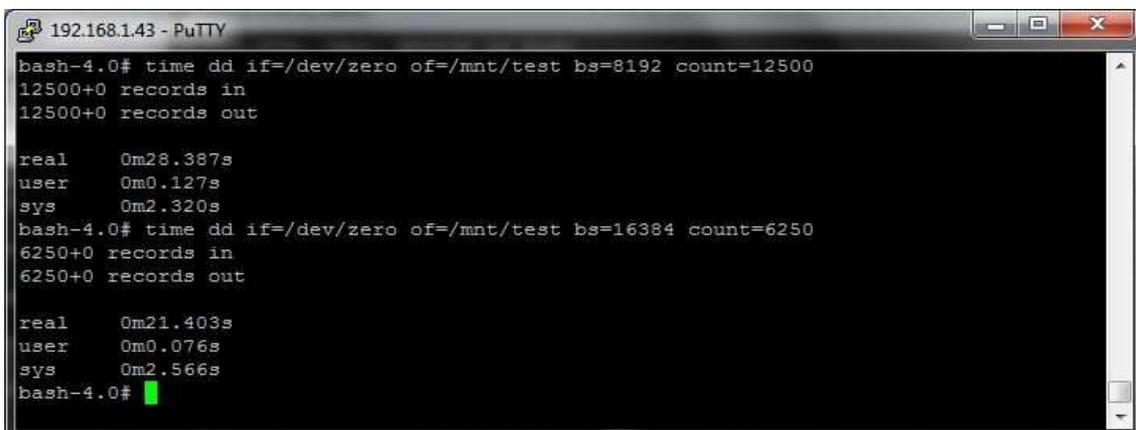


```
192.168.1.43 - PuTTY
sys     0m5.994s
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=2048 count=50000
50000+0 records in
50000+0 records out

real    0m7.649s
user    0m0.438s
sys     0m4.331s
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=4096 count=25000
25000+0 records in
25000+0 records out

real    0m6.266s
user    0m0.237s
sys     0m3.395s
bash-4.0#
```

116. Performance iSCSI 2.

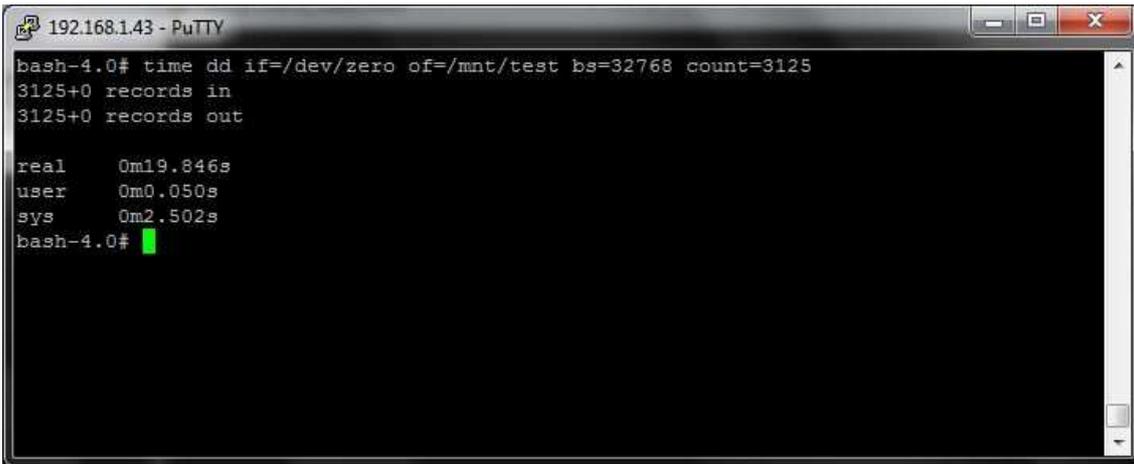


```
192.168.1.43 - PuTTY
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=8192 count=12500
12500+0 records in
12500+0 records out

real    0m28.387s
user    0m0.127s
sys     0m2.320s
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=16384 count=6250
6250+0 records in
6250+0 records out

real    0m21.403s
user    0m0.076s
sys     0m2.566s
bash-4.0#
```

117. Performance iSCSI 3.

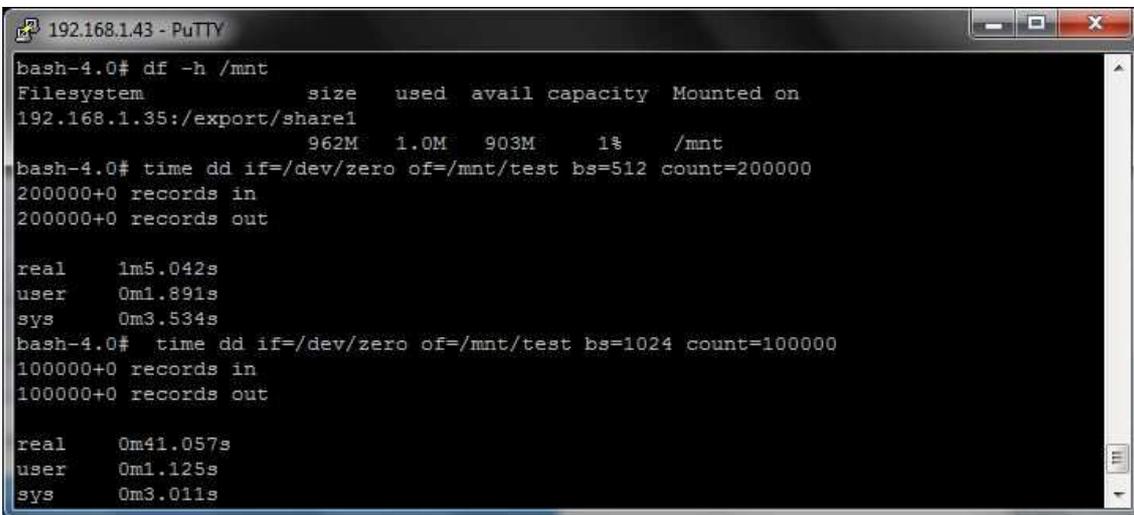


```
192.168.1.43 - PuTTY
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=32768 count=3125
3125+0 records in
3125+0 records out

real    0m19.846s
user    0m0.050s
sys     0m2.502s
bash-4.0#
```

118. Performance iSCS 4.

A continuación realizamos las pruebas sobre el filesystem montado con NFS.

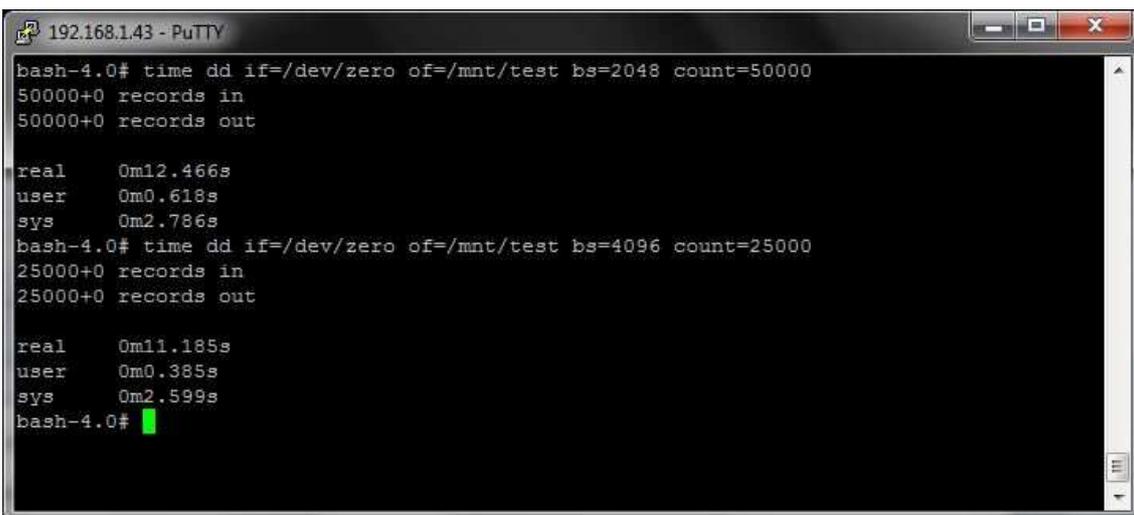


```
192.168.1.43 - PuTTY
bash-4.0# df -h /mnt
Filesystem              size  used  avail capacity  Mounted on
192.168.1.35:/export/share1
                        962M  1.0M   903M    1%    /mnt
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=512 count=200000
200000+0 records in
200000+0 records out

real    1m5.042s
user    0m1.891s
sys     0m3.534s
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=1024 count=100000
100000+0 records in
100000+0 records out

real    0m41.057s
user    0m1.125s
sys     0m3.011s
```

119. Performance NFS 1.



```
192.168.1.43 - PuTTY
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=2048 count=50000
50000+0 records in
50000+0 records out

real    0m12.466s
user    0m0.618s
sys     0m2.786s
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=4096 count=25000
25000+0 records in
25000+0 records out

real    0m11.185s
user    0m0.385s
sys     0m2.599s
bash-4.0#
```

120. Performance NFS 2.

```

192.168.1.43 - PuTTY
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=8192 count=12500
12500+0 records in
12500+0 records out

real    0m9.937s
user    0m0.223s
sys     0m2.542s
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=16384 count=6250
6250+0 records in
6250+0 records out

real    0m11.049s
user    0m0.143s
sys     0m2.441s
bash-4.0# █

```

121. Performance NFS 3.

```

192.168.1.43 - PuTTY
bash-4.0# time dd if=/dev/zero of=/mnt/test bs=32768 count=3125
3125+0 records in
3125+0 records out

real    0m33.953s
user    0m0.106s
sys     0m2.378s
bash-4.0# █

```

122. Performance NFS 4.

6.4 Conclusiones

En primer lugar para poder extraer conclusiones es necesario entender que todo el entorno está realizado sobre un Laptop, utilizando un disco duro común y conectado a una red doméstica. Por lo tanto los resultados que se extraen de las pruebas no pueden ser interpretados como si de un entorno real se tratase. Aún así, nos permite obtener diversas conclusiones de cómo se comportan diferentes sistemas operativos en el acceso a los medios virtuales, así como estudiar cómo influyen los resultados en función de los parámetros utilizados en el acceso a disco.

Para poder evaluar estos resultados se ha realizado la siguiente tabla para el protocolo iSCSI:

Tamaño Bloque	Windows	Linux	Solaris
512 Bytes	8,8 MB/s	3 MB/s	2,84 MB/s
1024 Bytes	5,3 MB/s	5 MB/s	10,36 MB/s

2048 Bytes	-	8 MB/s	13 MB/s
4096 Bytes	-	18 MB/s	16 MB/s
8192 Bytes	-	18 MB/s	3,5 MB/s
16384 Bytes	-	23 MB/s	4,67 MB/s
32768 Bytes	-	20 MB/s	5 MB/s

El primer dato reseñable es que en las pruebas, los mejores resultados trabajando con Windows se han obtenido utilizando un tamaño de bloque automático (el sistema operativo decide cuál es el mejor en cada situación). De esta manera se consiguió una velocidad máxima de escritura de 17,8 MB/s. Con tamaños de bloque específicos, las velocidades de transferencia eran mucho más bajas.

Con respecto a Linux y Solaris se puede observar que los mejores resultados se obtienen utilizando tamaños de bloque mayores del que el sistema operativo utiliza por defecto. Esto indica que, si una aplicación es muy sensible a los tiempos de espera por operaciones de i/o, se puede realizar tuning para obtener los mejores resultados de velocidad. Además en el sistema operativo OpenSolaris se observa como llegado a un umbral de tamaño de disco no se mejora la velocidad, sino que esta cae de forma abrupta. Este comportamiento es diferente al de Linux, el cual mantiene más o menos una velocidad aunque se aumente el tamaño de bloque. Estas diferencias vienen dadas por los diferentes mecanismos que utiliza el kernel del sistema operativo para gestionar las peticiones (caché, prioridades, tamaño de cachés, etc).

Con respecto a la utilización del protocolo NFS, tanto en Linux como en Opensolaris la velocidad también es dependiente del tamaño de bloque que se utiliza en la escritura del filesystem. La velocidad máxima de transferencia que se ha alcanzado en ambos sistemas es de 10 MB/s. Aunque es diferente topología de almacenamiento (NAS vs SAN) se puede ver que el rendimiento en términos de velocidad es casi del doble en iSCSI.

En el caso del sistema operativo Windows, las pruebas realizadas sobre Samba y NFS han devuelto casi los mismos resultados: 2MB/s y 3MB/s respectivamente. Ambos valores son tasas de transferencia bajas si se comparan con los resultados obtenidos con Linux y OpenSolaris, lo que nos indica que la forma en la que trabaja con estos protocolos no es todo lo eficiente que puede llegar a ser sobre Unix.

7. Conclusiones y presupuesto

7.1 Conclusiones del proyecto

Durante la realización del proyecto se ha tenido la oportunidad de entrar en el estudio a fondo de las actuales tecnologías de la información utilizadas en los entornos empresariales. Además se ha profundizado en el estudio de un sistema Unix como es OpenSolaris lo que ha permitido adquirir experiencia en su administración permitiendo conocer la utilización de los comandos más comunes. De esta forma, se ha podido ver que los sistemas Unix son comúnmente utilizados en las empresas sobre todo para aplicaciones denominadas de “misión crítica” es decir, que necesitan estar disponibles idealmente el 100% del tiempo. Otra de las conclusiones que se pueden sacar tras la realización de este proyecto es la penetración que tiene el software denominado “libre” en las empresas. Cada vez es mayor el número de empresas que comienzan a instalar nuevas máquinas con alguna distribución de Linux e incluso algunas poco a poco van migrando sus aplicaciones a las máquinas Linux.

Respecto a las denominadas técnicas de “virtualización” se ha podido comprobar como se extiende su utilización de manera vertiginosa. Cada vez es más común por los departamentos de TI comprar servidores con arquitectura Intel y crear máquinas virtuales. De momento las máquinas virtuales en entornos de producción aún no son utilizadas masivamente y se reserva su utilización a entornos de desarrollo y certificación principalmente por la rapidez y sencillez a la hora de montar un nuevo entorno para comenzar el desarrollo de una aplicación o certificación.

Si nos fijamos en la parte del almacenamiento de datos vemos que las necesidades de almacenamiento de datos siguen creciendo de manera muy rápida. Los fabricantes intentan consolidar en el mismo dispositivo todos los protocolos de acceso a almacenamiento posible, de manera que puedan cubrir las demandas de cualquier cliente y además proporcionar gran capacidad de almacenamiento en dispositivos que ocupen cada vez menos espacio y consuman menos en los centro de procesado de datos.

Como conclusión final, después del estudio de tecnologías llevado a cabo para realizar este proyecto, parece que la tendencia en el futuro será utilizar el software libre de manera masiva tanto por los usuarios como las empresas. Las aplicaciones también se están desarrollando para correr en los sistemas operativos “libres” y se desarrollan drivers para que estos sistemas operativos sean capaces de manejar cualquier dispositivo. Sobre el hardware se puede concluir que cada vez más se están utilizando tecnologías de virtualización tanto por sencillez, rapidez y ahorro, ya que con 1 sólo servidor se puede dar servicio a distintas demandas (por ejemplo una máquina virtual con Windows para frontales web, 2 máquinas virtuales Linux con MySQL, etc). La nueva forma de gestionar los servidores implica que las necesidades de almacenamiento crecen ya que se desarrollan aplicaciones y se ponen en producción con mayor rapidez y por lo tanto el almacenamiento de datos crece igual de rápido. Los nuevos dispositivos de almacenamiento deben ser versátiles en cuanto a las necesidades de crecimiento y además proveer mecanismos de seguridad suficientes para asegurar la consistencia de los datos.

7.2 Mejoras futuras

Con la realización de este proyecto fin de carrera y tras la primera toma de contacto con las nuevas tecnologías de transporte y almacenamiento de datos, surgen nuevas posibilidades y proyectos a realizar para implementar soluciones mejoradas a la actual.

La primera actualización a este proyecto debería ser su implementación en un servidor real. Para poder explorar las capacidades que puede ofrecernos la implementación realizada sería conveniente que el servidor estuviera conectado a una SAN de manera que los tiempos en el acceso a los datos (tanto en lectura como en escritura) sean muchísimo más pequeños que el rendimiento que se ha podido obtener con la implementación realizada. Además sería muy interesante poder realizar pruebas de iSCSI a través de tarjetas infiniband (principalmente utilizadas en entornos de supercomputación).

La segunda de las actualizaciones a llevar a cabo debería estar enfocada a la mejora de la seguridad en el acceso al almacenamiento utilizando “selective mapping”. En esta actualización se definirían reglas de acceso a la LUN para cada host.

Como tercera posible mejora resulta muy interesante añadir a la implementación actual el protocolo FcoE. De esta manera la solución resulta más versátil además de poder beneficiarse de las características que ofrece FcoE en cuanto a velocidad de acceso a los datos.

Y la última actualización a desarrollar pasaría por implementar una solución de alta disponibilidad de manera que aunque el servidor que trabaja como servidor de almacenamiento sufra una caída, el servicio no se vea parado o degradado y el cliente obtenga siempre una disponibilidad del 99,99% del tiempo. Para esta actualización se podría pensar en utilizar la versión libre de Sun Cluster, denominada Open High Availability Cluster que cuenta con la ventaja de que los agentes creados en la plataforma opensource pueden ser exportados a las máquinas de producción (sin soporte).

7.2.1 Procedimiento de estimación de recursos

Este proyecto se ha focalizado en el estudio, búsqueda e implementación de una solución iSCSI basada en software opensource. Hay que tener en cuenta que se ha utilizado una plataforma virtualizada sobre un PC de sobremesa, lo que hace que el rendimiento se pueda medir entre tecnologías (NFS y/o SMB vs iSCSI) pero no se pueden extrapolar datos a un entorno de producción. En un entorno productivo, corriendo OpenSolaris sobre un servidor Intel o Sparc y con una red dedicada, las tasas de transferencia serían mucho más elevadas proveyendo cifras más parecidas a las soluciones de fabricantes más costosas que la solución que en este proyecto se ha implantado y que se puede portar en cualquier momento a un entorno productivo.

Por tanto la estimación del presupuesto de este proyecto se realiza en base al tiempo dedicado en investigación, análisis e implementación y pruebas sobre una maqueta que trata de verificar que se cumplen las especificaciones requeridas y que goza de la estabilidad suficiente para ser implantada con garantías en un entorno productivo.

La estimación de recursos tendrá en cuenta por tanto los siguientes puntos:

- Equipo humano involucrado en todas y cada una de las fases del proyecto especificando las funciones individuales de cada individuo y las horas de trabajo reportadas por cada uno.
- Duración del proyecto
- Recursos hardware necesarios para implementar la maqueta
- Recursos software necesarios para su implementación

7.2.2 Descripción del equipo de trabajo

Puesto	Funciones
Jefe de Proyecto	Es la figura clave durante toda la duración del proyecto con autoridad para dirigir el equipo asignado. Se encarga del seguimiento del mismo así como de la toma de decisiones que hagan que el proyecto llegue a su término de la forma estipulada en el contrato.
Arquitectos de Sistemas	Son los encargados de estudiar las necesidades del cliente y buscar la solución tecnológica ajustada en coste y rendimiento para satisfacer la demanda solicitada. Además permanecerán en comunicación con los analistas de sistemas durante la etapa de implantación para solventar cualquier problema no previsto en la etapa de diseño.
Analistas de Sistemas	Son los encargados de implementar el diseño realizado por los arquitectos, así como de reportar cualquier problema que surja en esta etapa a los mismos.
Ingenieros de Pruebas de Sistemas	Son los encargados de, en un entorno de pruebas realizar la simulación del entorno de producción para poder testear las funcionalidades y el correcto comportamiento del sistema antes de su puesta en producción.

123. Desglose Equipo de trabajo.

7.2.3 Estudio del presupuesto

En este punto aparece reflejado el coste total del proyecto (sin incluir su implementación en producción) en función de las horas necesarias para llevar a cabo cada una de las etapas. En coste por hora se ha incluido lo siguiente:

- Gastos de personal (según el rol desempeñado)
- Viajes, dietas y otros gastos
- Material de Oficina
- Margen de beneficio

- Equipos informáticos necesarios
- Licencias de Software

Las etapas han sido codificadas de acuerdo al siguiente esquema:

- Fase I- Estudio del estado del Arte
- Fase II-Análisis de requisitos de usuario
- Fase III – Definición del entorno
- Fase IV - Diseño arquitectónico
- Fase V- Preparación y realización de pruebas

En la siguiente figura podemos ver el presupuesto desglosado por rol y por horas.

	Jefe de Proyecto	Arquitecto de Sistemas	Analista de Sistemas	Ingeniero Pruebas de Sistemas	Total
Fase I	10	20	0	0	30
Fase II	10	20	0	0	30
Fase III	15	20	40	0	75
Fase IV	15	10	40	5	70
Fase V	10	5	10	20	45
Total Horas	60	75	90	25	250
Precio Hora	40 €	35 €	31 €	25 €	-
Precio Total	2400 €	2625 €	2790 €	625 €	8440 €

8. Bibliografía

- [1]: TATE, Jon. Introduction to Storage Area Networks [Online]. Disponible en Web: <http://www.redbooks.ibm.com/redbooks/pdfs/sg245470.pdf>
- [2]: MEGGYESY, Zoltán. Fibre Channel Overview [Online]. Disponible en Web: <http://hsi.web.cern.ch/HSI/fcs/spec/overview.htm>
- [3]: DALE, David. Server and Storage Consolidation with iSCSI Arrays [Online]. Disponible en Web: http://www.snia.org/education/tutorials/2010/fall/storman/DavidDale_Consolidation_with_iSCSI_Arrays.pdf
- [4]: IETF. iSCSI Extensions for Remote Direct Memory Access (RDMA) [Online]. Disponible en Web: <http://tools.ietf.org/html/rfc5046>
- [5]: LOWE, Scott. iSCSI is the future of storage [Online]. Disponible en Web: <http://blogs.techrepublic.com.com/datacenter/?p=461>
- [6]: Sun Microsystems. OracleVM/virtualbox End User Manual [Online]. Disponible en web: <http://dlc.sun.com.edgesuite.net/virtualbox/3.2.10/UserManual.pdf>
- [7]: KHURSHUDOV, Andrei. The Essential Guide to Computer Data Storage (Editorial Prentice-Hall). ISBN 0-13-092739-2
- [8]: CLARK, Tom. IP Sans: A guide to iSCSI, iFCP and FCIP Protocols for Storage Area Networks [Online]. Disponible en Web: <http://my.safaribooksonline.com/0201752778>
- [9]: GILAD, Shainer. Why Compromise? [Online]. Disponible en Web: <http://www.hpcwire.com/features/17888274.html>