

Proyecto Fin de Carrera

DISEÑO INTEGRAL DE PLATAFORMA DE VIRTUALIZACIÓN

Título:	DISEÑO INTEGRAL DE PLATAFORMA DE VIRTUALIZACIÓN
Titulación:	INGENIERÍA INFORMÁTICA
Estudiante:	RICARDO MARTÍN MANSO
Consultor:	VICTOR CARCELER
Fecha:	16 de diciembre de 2010

Agradecimientos

Agradecer a Victor Carceler sus consejos y su guía en el desarrollo del proyecto.

A mi pareja Clara, agradecerla también, la paciencia que ha tenido conmigo sacrificando sus horas de ocio para proporcionarme el ambiente de tranquilidad necesario para poder sacar adelante el reto que ha supuesto titularme.

A mis amigos por la comprensión que han tenido especialmente estos dos últimos años.

Por último, quiero dedicar el esfuerzo realizado durante toda la carrera que ha tenido como colofón este proyecto a mis padres y a mi hermana María por el cariño y el apoyo con el que me han premiado toda mi vida.

A todos ellos: gracias!

RESUMEN

El objetivo de este proyecto es el diseño, razonado y apoyado en datos objetivos, de una plataforma de virtualización basada en VMWARE ESXi (versión gratuita de VMWARE), orientada al segmento de empresas y organizaciones que por su tamaño podrían beneficiarse de un entorno de servidores virtualizados pero que por su presupuesto no pueden acceder a tecnologías punteras de implantación.

El enfoque del proyecto será integral en el sentido que trabajá tanto la plataforma física de implantación, estudiando las tecnologías disponibles y comparándolas con las más caras y onerosas de las plataformas empresariales al uso, la configuración del software necesario, el desarrollo de una plataforma de gestión libre (que voy a llamar Free Virtual Center, FVC), el desarrollo de una solución de backup adecuada a las necesidades particulares de la virtualización (implementará nativamente tecnologías de deduplicación) y, por último, un análisis de costes medioambientales y económicos comparados con una plataforma física tradicional.

Al finalizar el proyecto, deberíamos contar con un diseño tanto a nivel físico, como tecnológico y de software capaces de ofrecer a un precio muy competitivo las ventajas de las virtualización a nivel empresarial a organizaciones y empresas con un presupuesto limitado.

Índice de contenido

1. INTRODUCCIÓN.....	7
1.1. Justificación y contexto del PFC	7
1.2. Objetivos del PFC.....	7
1.3. Enfoque y método seguido.....	8
1.4. Planificación del proyecto.....	9
1.4.1 Resultados esperados.....	9
1.4.2 Alcance y entregables.....	9
1.4.3 Plazos.....	11
1.4.5 Hitos y Actividades del proyecto.....	12
1.4.6 Roles y Recursos Asignados.....	13
1.4.7 Estimación de esfuerzo: Tareas/Roles.....	13
1.4.8 Total de Horas/Roles.....	14
1.4.9 Total de Horas/Personal.....	14
1.4.10 Gestión de Riesgos.....	14
1.5. Productos obtenidos	15
2. ESTUDIO DE LAS TECNOLOGÍAS DE IMPLANTACIÓN.....	16
2.1 Definición de los objetivos y alcance del estudio.....	16
2.2 Acceso local a disco.....	17
2.2.1 Modelado de los escenarios para las comparativas sintéticas.....	18
2.2.2 Entorno de pruebas.....	18
2.2.4 Software para las pruebas.....	19
2.2.5 Codificación de las pruebas.....	20
2.2.6 Resultado de las pruebas.....	21
2.3 Acceso a disco remoto: SAN.....	23
2.4 Desempeño tecnologías de red.....	24
2.5 Acceso a servidor de archivos	24
2.6 Rendimiento en cálculos y acceso a memoria. Maquina física vs Maquina virtual.....	25
2.8 Conclusiones.....	26
3 DISEÑO FINAL DE DESPLIEGUE.....	27
3.1 Diseño servidores	27
3.1.1 Servidor repositorio de máquinas virtuales (VMDK).....	27
3.1.2 Servidor de backup.....	30
3.1.3 Servidor de imágenes ISOS.....	30
3.1.4 Servidor de archivos de usuarios.....	30
3.1.5 Servidor de bases de datos.....	31
3.1.6 Servidor FVC.....	31
3.1.7 Servidores ESXi.....	32
3.1.8 Diseño infraestructura física.....	33
3.1.9 CONCLUSIONES.....	34
3.2 Diseño elementos de apoyo.....	35
3.2.1 Red de comunicaciones internas.....	35
3.2.2 Conexionado eléctrico.....	35
3.2.3 SAI.....	36
3.2.4 Potencia frigorífica.....	36
4 DESARROLLO DEL SOFTWARE (FVC).....	37
4.1 Introducción.....	37

4.2	Objetivos.....	37
4.3	Alcance y entregables.....	38
4.4	Gestión de la configuración.....	38
4.5	Diagrama de despliegue.....	39
4.6	Casos de uso.....	40
4.7	Decisiones de diseño.....	41
4.8	Prototipo de pantallas.....	42
4.9	Diagramas de componentes.....	48
4.10	Diagramas de clases.....	48
4.11	Diagrama de base de datos.....	51
4.12	Diagramas de actividad.....	52
5.	DESARROLLO DE HERRAMIENTA DE BACKUP DEDUPLICADO (AD).....	53
5.1	Introducción.....	53
5.2	Objetivo de proyecto de desarrollo de la herramienta de backup deduplicado.....	54
5.3	Alcance del proyecto.....	55
5.4	Estudio previo de las herramientas de backup	56
5.4.1	Introducción.....	56
5.4.2	Herramientas del sistema operativo: XCOPY y ROBOCOPY.....	57
	Estudio teórico de rendimiento máximo y pruebas de laboratorio.....	59
5.4.3	Herramienta de compresión de datos: 7z (software libre).....	61
5.4.4	Conclusión global.....	64
5.4.5	Uso eficiente del ancho de banda de red en backups remotos.....	65
5.4.6	Copias a nivel de bloques de datos: Snapshots.....	67
5.4.7	Deduplicación a nivel del sistema de archivos.....	69
5.4.8	Propuesta arquitectura para optimizar backup.....	70
5.5	DESARROLLO DE LA HERRAMIENTA.....	76
5.5.1	Casos de uso.....	76
5.5.2	Decisiones de diseño.....	79
5.5.3	Gestión de la configuración.....	79
5.5.4	Diagrama de clases	80
5.5.5	Estructuras de datos internas al archivo AD.....	81
5.5.6	Detalle del algoritmo de deduplicación.....	81
5.5.7	Estudio teórico de rendimiento máximo y pruebas de laboratorio.....	84
5.5.8	Rendimiento y pruebas.....	85
5.5.9	Conclusiones.....	86
6.	VALORACIÓN ECONÓMICA y de CONSUMO ENERGÉTICO.....	87
7.	CONCLUSIONES.....	88
8.	GLOSARIO.....	90
9.	BIBLIOGRAFÍA.....	90

Índice de ilustraciones

Ilustración 1: Escenario despliegue 1.....	28
Ilustración 2: Escenario despliegue 2.....	28
Ilustración 3: Escenario despliegue 3.....	28
Ilustración 4: Infraestructura física a nivel de servidores - I.....	33
Ilustración 5: Infraestructura física a nivel de servidores - II.....	34
Ilustración 6: Diagrama de despliegue.....	39
Ilustración 7: Diagrama de casos de uso.....	40
Ilustración 8: Prototipo pantalla principal del FVC.....	42
Ilustración 9: Prototipo pantalla gestion del pool.....	44
Ilustración 10: Prototipo pantalla gestion del host.....	45
Ilustración 11: Prototipo pantalla gestion maquina virtual.....	47
Ilustración 12: Diagrama de componentes.....	48
Ilustración 13: Diagrama de clases:presentacion.....	48
Ilustración 14: Diagrama de clases:Bussines logic.....	49
Ilustración 15: Diagrama de clases:Integración.....	50
Ilustración 16: Diagrama de bases de datos.....	51
Ilustración 17: Backup de VM (en caliente).....	52
Ilustración 18: VMOTION de VM (migración en caliente).....	52
Ilustración 19: Clonado de VM (en frío).....	52
Ilustración 20: HA (alta disponibilidad).....	52
Ilustración 21: Escenario de backup 1.....	70
Ilustración 22: Escenario de backup 2.....	72
Ilustración 23: Escenario backup 3.....	73
Ilustración 24: Escenario backup 4.....	74
Ilustración 25: Escenario backup 5.....	74
Ilustración 26: Diagrama de clases de AD.....	80
Ilustración 27: Diagrama de estructuras de datos en disco de AD.....	81
Ilustración 28: Deduplicación: bloques en disco.....	83

1. INTRODUCCIÓN

1.1. Justificación y contexto del PFC

En el ámbito de las TIC, continuamente surgen nuevas tecnologías, maduran otras y desaparecen muchas. De vez en cuando, aparecen tecnologías y productos que marcan hitos históricos y que por su utilidad, versatilidad o economía se consolidan en el mercado de forma duradera y estable: Pensemos, por ejemplo, en la aparición de los interfaces gráficos de usuario, la revolución de los 32 bits, el XML, etc.

Las tecnologías de virtualización, en su conjunto, pertenecen a esta categoría de productos. Llevan ya el suficiente tiempo en el mercado como para poder asegurar que es consenso generalizado entre los profesionales del ramo que ofrecen ventajas incuestionables sobre la implantación de Centros de procesos de datos (CPD) basados exclusivamente en servidores físicos.

Si bien las ventajas competitivas que ofrece la virtualización son ya incontestables también es cierto que los proyectos de virtualización se enfrentan a menudo a retos importantes y costosos en plazos en lo referente a la implantación física de dichas plataformas y en la gestión del entorno una vez puesto en producción.

1.2. Objetivos del PFC

El objetivo de este proyecto es el diseño, razonado y apoyado por datos objetivos, de una plataforma para la implantación de un sistema de virtualización basado en VMWARE ESXi (versión gratuita de VMWARE), orientada al segmento de empresas y organizaciones que por su tamaño podrían beneficiarse de un entorno de servidores virtualizados pero que por su presupuesto no pueden acceder a tecnologías punteras de implantación.

1.3. Enfoque y método seguido

El enfoque del proyecto será integral en el sentido que trabajá tanto la plataforma física de implantación, estudiando las tecnologías disponibles y comparándolas con las más caras y onerosas de las plataformas empresariales al uso, la configuración del software necesario, el desarrollo de una plataforma de gestión libre (que voy a llamar Free Virtual Center, FVC), frente al producto que vende VMWARE para soluciones empresariales (Virtual Center, VC), el desarrollo de una solución de backup adecuada a las necesidades particulares de la virtualización (implementará nativamente tecnologías de deduplicación) y, por último, un análisis de costes medioambientales y económicos comparados con una plataforma física tradicional.

1.4. Planificación del proyecto

1.4.1 Resultados esperados

Al finalizar el proyecto, deberíamos contar con un diseño tanto a nivel físico, como tecnológico y de software capaces de ofrecer a un precio muy competitivo las ventajas de la virtualización a nivel empresarial a organizaciones y empresas con un presupuesto limitado.

1.4.2 Alcance y entregables

El proyecto se divide en cinco grandes bloques con objetivos y alcances particulares:

Bloque 1: Estudio comparativo de las tecnologías de implantación.

Se realizarán estudios comparativos de todas las tecnologías necesarias para la implantación de una solución de virtualización de las características de coste y rendimiento adecuadas. Se confeccionará un informe completo de las pruebas realizadas y un informe de conclusiones en el que se justifique ,con datos objetivos, un diseño concreto, a nivel de tecnologías de implantación, de la plataforma de virtualización.

Bloque 2: Diseño de la plataforma hardware

Basado, en parte, en las conclusiones del bloque 1, se diseñará la plataforma hardware adecuada para soportar el despliegue de la solución de virtualización.

Se entregará, al final de este bloque, un diseño arquitectónico con elección de servidores, elementos de apoyo, alimentación eléctrica y potencia frigorífica adecuadas.

Bloque 3:Implantación de todo el sistema

Implantación del sistema basado en máquinas virtuales para pool de servidores ESXi, servidor NAS/SAN, servidor NFS/ISOS, servidor de backup y Free Virtual Center.

Las máquinas virtuales resultantes serán ejecutables en el mismo entorno que propongo, o, en un VMWARE WorkStation.

Bloque 4: Desarrollo de la herramienta software: Free Virtual Center (FVC)

VMWARE proporciona, para la versión Enterprise de sus productos, un software llamado Virtual Center (VC) que es capaz de una forma unificada de trabajar con toda la infraestructura virtual de una organización gestionando características avanzadas que en las versiones gratuitas de sus productos no están soportadas oficialmente.

El objetivo del desarrollo del Free Virtual Center (FVC) es proporcionar para las versiones gratuitas de VMWARE ESXi, funcionalidades lo más parecidas posibles a las proporcionadas por el VC, e incluso, en algunos temas más avanzadas o integradas con el entorno unificado. La herramienta o entorno a desarrollar está enfocada a proporcionar a empresas y organizaciones las funcionalidades que no se ofrecen en la versión gratuita y, por tanto, complementará los interfaces de acceso web a una sola máquina física que ofrece VMWARE y que aportan mucha funcionalidad para máquinas independientes pero están intencionadamente limitados cuando queremos disfrutar de una plataforma multihost y con características empresariales interesantes.

Se consideran objetivos mínimos alcanzables los siguientes:

Funcionalidad similar a la proporcionada por VC: (en lo posible)

- Visión unificada de todas las máquinas físicas y virtuales del entorno ESXi
- Plantillas/clonado
- Alta disponibilidad (HA)
- Migración de máquinas entre máquinas físicas (VMOTION)
- Monitorización centralizada. Avisos a responsables en caso de incidencia

Mejoras respecto a VC:

- Backup centralizado con deduplicación

Objetivo opcional: Si los plazos lo hicieran posible, se desarrollará la funcionalidad necesaria para implementar el workflow de peticiones de máquinas virtuales por parte de usuarios internos a nuestra organización, integrando así la plataforma de sistemas con las necesidades del resto de la organización con un enfoque basado en servicios (ITIL v3).

Entregables: Software y documentación del FVC (JAVA sobre TOMCAT) y software de deduplicación (API y aplicación en modo consola standalone).

Bloque 5: Estudio económico de toda la plataforma vs plataforma física.

Informe comparativo de la solución propuesta versus una basada en servidores físicos desde el punto de vista del coste económico y energético.

1.4.3 Plazos

El plazo para la terminación del proyecto es de 69 días: Del 25/10 al 1/1/2011.

1.4.5 Hitos y Actividades del proyecto

Diagrama de Gantt de los estudios a realizar:

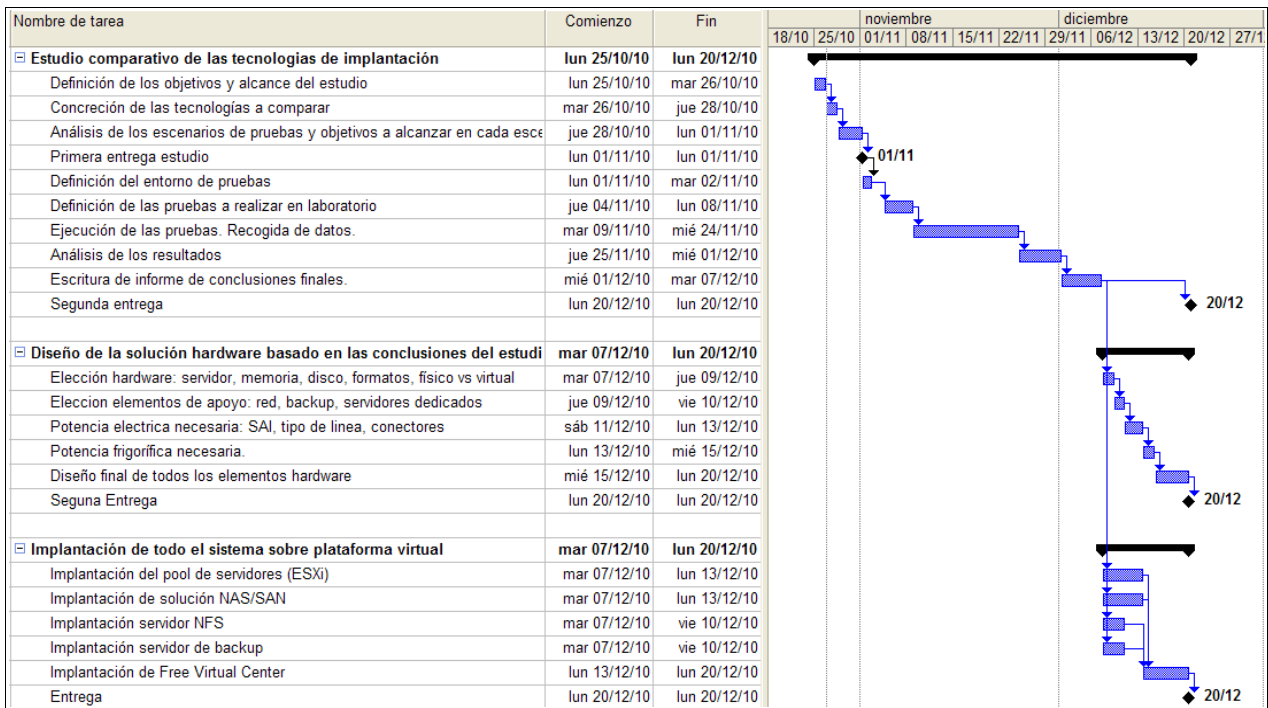
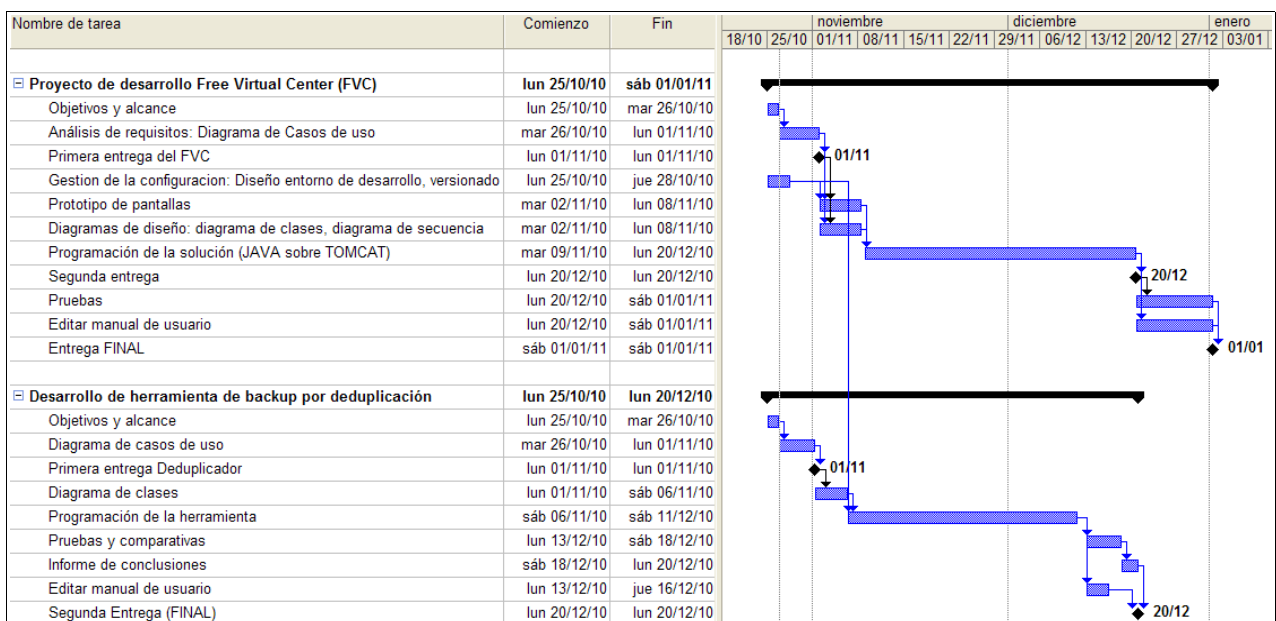


Diagrama de Gantt de los trabajos de desarrollo:



1.4.6 Roles y Recursos Asignados

SIGLAS	ROLES	PERSONA ASIGNADA
TS	TÉCNICO DE SISTEMAS	RICARDO
AS	ANALISTA DE SISTEMAS	RICARDO
AF	ANALISTA FUNCIONAL	RICARDO
DE	DESARROLLADOR	RICARDO

1.4.7 Estimación de esfuerzo: Tareas/Roles

Para la estimación de esfuerzo se parte de los siguientes supuestos:

1.- El personal del equipo (RICARDO), dedicará a este proyecto un máximo de 4 horas diarias, 6 días a la semana. Por lo tanto, solo se considerarán festivos los domingos.

2.- Solapamiento de tareas y actividades:

Del 25/10 al 6/12 se trabajará en 3 actividades en paralelo, dedicando el 33% de la jornada a cada actividad.

Del 7/12 al 20/12 se trabajará en 4 actividades en paralelo, dedicando el 25% de la jornada a cada una.

Del 21/12 al 1/1 se trabajará en una sola actividad a la vez.

3.- Cuando se solapen tareas, dentro de una actividad, se repartirá el esfuerzo equitativamente entre cada una de ellas.

ACTIVIDAD	TS	AS	AF	DE
Activ. 1	$14 * 4/3 = 18.66$ h	$24 * 4/3 = 32$ h		
Activ. 2		$12 * 4/4 = 12$ h		
Activ. 3	$13 * 4/4 = 13$ h			
Activ. 4		$2 * 4/3 = 2.66$ h	$12 * 4/3 = 16$ h $10 * 4/1 = 40$ h	$25 * 4/3 = 33.33$ h $10 * 4/4 = 10$ h
Activ. 5	$4 * 4/4 = 4$ h		$12 * 4/3 = 16$ h $4 * 4/4 = 4$ h	$26 * 4/3 = 37.33$ h $4 * 4/4 = 4$ h

1.4.8 Total de Horas/Roles

ROLES	HORAS
TS	35.66 horas
AS	46.66 horas
AF	76 horas
DE	81.66 horas
TOTAL:	240 horas

1.4.9 Total de Horas/Personal

PERSONAL	HORAS
RICARDO	240 horas (60 días laborables a 4 horas/día)

1.4.10 Gestión de Riesgos

RIESGO	PROBABILIDAD	ACCIÓN CORRECTORA
Alguno de los productos a evaluar presente dificultades de instalación en el HW disponible.	50%	Del 1 al 10 de enero existe un colchón de horas para absorber retrasos en general en el proyecto.
El nuevo API para VMWARE ESXi cambie aspectos fundamentales respecto a la version conocida	25%	En último caso, se desarrollaría sobre la versión ya experimentada.
Dificultad para trabajar 4 horas al día, 6 días a la semana por solapes con otras obligaciones.	33%	Se trabajaría los domingos.

1.5. Productos obtenidos

Los cinco grandes bloques en los que se ha planteado este proyecto han producido, cada uno, resultados y desarrollos que respaldan, forman o dan utilidad al diseño global de la plataforma de virtualización propuesta.

Los estudios de tecnologías nos han dado información sobre el rendimiento real de cada posibilidad técnica planteada, sus pros y sus contras y nos han llevado finalmente a un diseño en el que se ha optimizado cada uno de los parámetros deseados: alta disponibilidad, rendimiento, precio, etc.

Basada en estos estudios, se ha concretado una plataforma de virtualización, al alcance de cualquier persona/empresa, que se acerca lo más posible a las mucho más onerosas plataformas empresariales de alto rendimiento.

Algunos productos software clave no están disponibles de forma gratuita por lo que un objetivo importante de este proyecto era desarrollar software a media para aportar dichas funcionalidades de forma económica.

Dichos desarrollos se han concretado en una herramienta de gestión centralizada de la arquitectura virtual, basada en JAVA y tecnologías WEB que permite realizar muchas de las funcionalidades de la versión de pago de VMWARE de forma gratuita y en algunos casos más sencilla.

Apoyando dicho software se ha diseñado también una herramienta en modo consola/api java que permite hacer backups deduplicados, problema o necesidad ésta que se plantea a todo el que se sumerge en este mundo de la virtualización y que supone un gran problema diario.

Por último, se ha hecho un estudio económico y medioambiental de las ventajas de la virtualización sobre plataforma física tradicional tanto a nivel de costes como en pro de la idea del *Green Computer*.

En resumen, tal como se planteó el objetivo clave del proyecto, se ha realizado el diseño integral de toda la plataforma a todos los niveles: hardware, software, tecnologías de implantación, desarrollo del software necesario y estudio económico.

2. ESTUDIO DE LAS TECNOLOGÍAS DE IMPLANTACIÓN

2.1 Definición de los objetivos y alcance del estudio

Se realizarán estudios comparativos de todas las tecnologías necesarias para la implantación de una solución de virtualización de las características de coste y rendimiento adecuadas. Se confeccionará un informe completo de las pruebas realizadas y un informe de conclusiones en el que se justifique ,con datos objetivos, un diseño concreto, a nivel de tecnologías de implantación, de la plataforma de virtualización.

Para cada tecnología de implantación se estudiará sus comportamiento frente a tres parámetros que se valorarán en el orden siguiente:

- Alta disponibilidad: Redundancia/Robustez frente a pérdida de datos e interrupción del servicio
- Economía: La solución debe satisfacer el resto de parámetros a un coste asumible en el escenario planteado para el proyecto.
- Rendimiento: Una vez satisfechas las características anteriores, debe alcanzarse el máximo rendimiento y, en todo caso, conocer la diferencia de éste respecto a tecnologías punteras en el mercado.

2.2 Acceso local a disco

Se van a estudiar aquí las diferentes opciones a la hora de implementar el acceso local a disco.

Concretamente, se trabajará en cuatro escenarios de uso:

- Almacenamiento de las máquinas virtuales (VMDK)
- Almacenamiento de los ficheros de backup en disco
- Almacenamiento de grandes ficheros estáticos: ISO, instalaciones
- Almacenamiento de archivos de usuario

Para cada uno de los escenarios anteriores se tratará de identificar que tecnología de las enumeradas a continuación es la más adecuada:

- RAID: NoRaid, Raid0, Raid1, Raid5, Raid10
- RAID HARDWARE vs RAID SOFTWARE
- Máquina Física vs Máquina Virtual
- SAS vs SATA II(doméstico)

Los puntos anteriores se testearán para los sistemas operativos:

- Windows Server 2003 Standar Edition R2
- Solaris 10 64 bits for x86 (INTEL y AMD)
- ESXi 4.0

2.2.1 Modelado de los escenarios para las comparativas sintéticas

- Almacenamiento de las máquinas virtuales (VMDK)
En el uso normal, puede modelarse como accesos aleatorios a bloques pequeños (64K) y grandes (8MB).
Cuando hacemos backup, se modelaría como accesos secuenciales a bloques grandes
- Almacenamiento de los ficheros de backup en disco
Se modelaría como escrituras secuenciales de bloques grandes
- Almacenamiento de grandes ficheros estáticos: ISO, instalaciones
Lecturas secuenciales de bloques grandes
- Almacenamiento de archivos de usuario
En el uso normal, accesos aleatorios a bloques pequeños (64K) y grandes (8M)
Durante el proceso de backup, accesos aleatorios a bloques grandes (8M)

2.2.2 Entorno de pruebas

Las pruebas de acceso a disco local las vamos a realizar en:

- Dos servidores HP Proliant 360 G5, con 16 GB de ram, procesador DUAL XEON E5440 2.83 Ghz 4 CORESxCPU, y cuatro discos duros SAS 10K, de 72 GB. Controladora de disco RAID: Smart Array P400i.
- PC doméstico: Procesador Phenom II, 4 CORES, 4 GB de ram, 1 disco duro ATA100 de 160 GB y 2x discos SATAII de 750GB. RAID integrado en placa base ASUS.

2.2.4 Software para las pruebas

- HDTUNE PRO (trial): Programa para testar diferentes parámetros del acceso a disco, disponible para Windows.

Para cada prueba observaremos 3 parámetros:

- Random Seek (64KB),
- Random Seek (8MB)
- Sequential Middle

- ROBOCOPY: utilidad del sistema operativo

- Copiaremos un conjunto de datos de prueba, de 23 GB de tamaño, repartido en 2 ficheros de 4 GB y resto de ficheros de pequeño tamaño de una carpeta de un disco a otra carpeta del mismo disco.

Antes de ejecutar esta prueba siempre se formateará el disco.

- Esta prueba hace incapié en accesos aleatorios de pequeño y gran tamaño en la carpeta origen y accesos secuenciales de gran tamaño en carpeta destino.

- teshHD: Programa realizado en C y que escribe un fichero del tamaño especificado. Compilado para windows.

- Las pruebas se realizarán con un fichero de 50 GB.
- Antes de ejecutar esta prueba siempre se formatea el disco.
- Esta prueba hace incapie en accesos secuenciales de gran tamaño.

- Copy en Solaris

```
cp -rf /disco2/Copia /disco2/Copia2
```

Conjunto de datos idéntico al de ROBOCOPY

- Creación fichero en Solaris

```
mkfile 50g /disco2/fichero
```

2.2.5 Codificación de las pruebas

Clave	Prueba
Windows:	
P1	HD TUNE PRO - Random Seek (64KB)
P2	HD TUNE PRO - Ramdom Seek (8MB)
P3	HD TUNE PRO - Sequential Midle
P4	ROBOCOPY
P5	teshHD
Solaris:	
P6	Copy en Solaris, sistema de archivos UFS
P7	Crear fichero en Solaris, sistema de archivos UFS
ESXi 4.0	
P8	P1 sobre windows virtualizado
P9	P2 sobre windows virtualizado
P10	P3 sobre windows virtualizado
P11	P4 sobre windows virtualizado
P12	P5 sobre windows virtualizado

Ejecución de las pruebas: Cada prueba se repite cinco veces, en diferentes momentos del día, siempre sobre discos recién formateados. Los resultados expresan el promedio en caso de que los datos obtenidos las cinco repeticiones se agrupen razonablemente sobre dicho promedio y se expresa como “-” en caso de que la prueba o no haya podido hacerse o los resultados presenten una dispersión que haga desaconsejable calcular el promedio.

2.2.6 Resultado de las pruebas

Windows Server 2003 R2 - RAID POR HARDWARE					
	NoRaid	Raid0	Raid1	Raid5	Raid10
P1	-	2 MB/s	2.4 MB/s	2.1 MB/s	2.4 MB/s
P2	-	105 MB/s	92 MB/s	117 MB/s	131 MB/s
P3	-	148 MB/s	74 MB/s	83 MB/s	149 MB/s
P4	18 min	11 min	16 min	39 min	11 min
P5	12 min	8 min	14 min	11 min	8 min

Windows Server 2003 R2 - RAID POR SOFTWARE					
	NoRaid	Raid0	Raid1	Raid5	Raid10
P4	18 min	18 min	22 min	51 min (*)	No soportado
P5	12 min	6 min	12 min	120 min (*)	No soportado

Solaris 10 - RAID POR HARDWARE					
	NoRaid	Raid0	Raid1	Raid5	Raid10
P6	32 min	17 min	31 min	23 min	18 min
P7	28 min	11 min	26 min	20 min	14 min

Solaris 10 - RAID POR SOFTWARE					
	NoRaid	Raid0	Raid1	Raid5	Raid10
P6	32 min	23 min	51 min	39 min	24 min
P7	28 min	10 min	18 min	18 min	11 min

Conclusiones:

- El mejor RAID para almacenamiento local datos es, sin duda, el RAID 10, ya que es el más rápido junto con el RAID 0 pero aventaja a éste en que además ofrece redundancia de datos por lo que si se estropea un disco podemos sustituirlo antes de perder datos, ventaja de la que no disfrutamos con RAID 0.
- El RAID por hardware aventaja en algunas pruebas al RAID por software en un 33%.
- Tanto Windows como Solaris 10 han sido estables utilizando cada RAID si bien Windows ha sido globalmente más rápido en las pruebas realizadas. Desventaja: no soporta R10 por SW.

ESXi - RAID POR HARDWARE					
	NoRaid	Raid0	Raid1	Raid5	Raid10
P8	2.5 MB/s	2.5 MB/s	2.5 MB/s	(*)	2.5 MB/s
P9	65 MB/s	94 MB/s	52 MB/s	(*)	92 MB/s
P10	68 MB/s	113 MB/s	68 MB/s	(*)	85 MB/s
P11	33 min	21 min	26 min	(*)	22 min
P12	20 min	15 min	19 min	(*)	16 min

(*) RAID5, por hardware, ha dado unos rendimientos anormalmente bajos en las máquinas HP donde se ha probado. Posiblemente se deba a algún bug de las controladores hardware/drivers que montan: P400 y P600.

Conclusión: El mejor RAID para almacenamiento local de máquinas, es sin duda, el RAID 10, ya que es el más rápido junto con el RAID 0 pero aventaja a éste en que además ofrece redundancia de datos por lo que si se estropea un disco podemos sustituirlo antes de perder datos, ventaja de la que no disfrutamos con RAID 0. Además, la reconstrucción de un RAID 10 también es más rápida que en el caso de RAID5.

RAID 10 es el mejor para almacenamiento local de máquinas virtuales.

2.3 Acceso a disco remoto: SAN

Se ha movido una máquina virtual, la utilizada en la prueba anterior, a un almacenamiento remoto y se han hecho las pruebas para diferentes tecnologías de comunicación entre la cabina remota (EMC Clariion CX4-320) y el servidor físico donde está instalado el ESXi que pone en funcionamiento dicha máquina virtual.

VM Windows Server 2003 R2 (virtualizado) en cabina SAN				
	ISCSI a 1 Gbps	ISCSI a 10 Gbps	FiberChannel a 4Gbps	FCoE a 10Gbps
P11	19 min	16 min	19 min	16 min
P12	15 min	11 min	13 min	13 min

VM Windows Server 2003 R2 (virtualizado) en un NFS, red a 1Gbps		
	Servidor NFS: Solaris físico RAID 10 - HW	Servidor NFS: VM Solaris, sobre ESXi, RAID 10 - HW
P11	34 min	46 min
P12	69 min	90 min

VM Windows Server 2003 R2 (virtualizado) en un NFS/iSCSI OPENFILER(ext3), red a 1Gbps		
	Servidor NFS: Openfiler RAID 10 - HW	Servidor iSCSI: Openfiler RAID 10 - HW
P11	55 min	28 min
P12	100 min	40 min

VM Windows Server 2003 R2 (virtualizado) en un NFS/iSCSI OPENFILER(ext3), red a 1Gbps		
	Servidor NFS: Openfiler RAID 10 - SW	Servidor iSCSI: Openfiler RAID 10 - SW
P11	59 min	38 min
P12	100 min	43 min

Conclusiones:

- El acceso remoto a SAN ha sido muy rápido, tanto o más que el acceso local a disco, lo que se explica porque las cabinas de disco usan muchos discos en paralelo por lo que la sobrecarga de tiempo que perdemos por el acceso por red lo ganamos en el enorme ancho de banda en disco y los grandes cachés.
- Cuando utilizamos tecnologías de más bajo coste y hacemos uso de almacenamiento en menor número de discos: 4 para RAID 10, el rendimiento a decaído a un modesto 33-50% del conseguido en local.
- ISCSI servido por OPENFILER ha dado unos tiempos bastante aceptables y mejores que NFS.
- Solaris ha sido el mejor sirviendo NFS.(Versión 3 en Solaris, 2 en Windows)

2.4 Desempeño tecnologías de red

Se han realizado las pruebas P4 y P5 desde un servidor Windows 2003 R2 físico tomando como destino un servidor Windows virtualizado y sin virtualizar utilizando conexiones a 1Gbps y 10Gbps

Windows físico a VM Windows Server 2003 R2 (virtualizado) por CIFS		
	1 Gbps	10 Gbps
P4	12 min	11.30 min
P5	20 min	19 min

Windows físico a Windows Server 2003 R2 (físico) por CIFS		
	1 Gbps	10 Gbps
P4	9.30 min	9.25 min
P5	15 min	15 min

Conclusiones:

- El uso de red a 10Gbps no ha desempeñado ningún papel relevante en esta prueba lo que quiere decir que con la mucho más económica red a 1Gbps tenemos suficiente ancho de banda para muchas aplicaciones que no llegan a saturar el enlace de red.

2.5 Acceso a servidor de archivos

Windows físico a servidor de archivos (físico), RAID 10, red a 1Gbps				
	OPENFILER NFS	OPENFILER CIFS	WINDOWS FÍSICO CIFS	VM WINDOWS CIFS
P4	16 min	13 min	9,30 min	12 min
P5	38 min	55 min	15 min	20 min

Conclusiones:

- Para servir archivos en una red windows, utilizando CIFS, el mejor sistema ha sido Windows 2003 tanto físico como virtualizado.
- Por tanto, los servidores de archivos se montarán preferiblemente como máquinas virtuales windows 2003.

2.6 Rendimiento en cálculos y acceso a memoria. Máquina física vs Máquina virtual

Todas las pruebas se han realizado con un software llamado Dr.Hardware sobre máquinas Windows 2003.

Test:	Acceso memoria	CPU	FPU
Máquina física:	4985	4872	2025
Máquina virtual:	4396	4420	2018

Por tanto, la pérdida de rendimiento en cálculos de CPU ronda el 7% cuando virtualizamos con esta solución. En el acceso a memoria la diferencia es del 12%.

Basándome en estadísticas anuales en instalaciones de 300 servidores virtuales sobre 15 máquinas físicas, he calculado los siguientes datos que pueden ser de utilidad para el dimensionamiento de infraestructuras:

Uso de memoria física: He estimado que el uso real de memoria física de los equipos, estadísticamente, será la suma de la memoria asignada a cada máquina virtual dividida entre 1.5.

Por ejemplo: Si la suma de la memoria que hemos asignado a nuestras máquinas virtuales es de 60 GB la memoria física con la que deberemos dotar a nuestro pool de máquinas físicas VMWARE será de $60/1.5=40$ GB.

Uso de CPU. Con una metodología de cálculo similar se llega a la conclusión que en el caso del uso de CPU real respecto a número de CPU virtuales definidas para nuestras VM, en promedio, tendremos un factor de 16. Es decir, cada CPU física soportará perfectamente hasta 16 virtuales. Siempre hablando estadísticamente: en implantaciones concretas el uso de CPU de una máquina varía en un rango muy amplio casi del 0 a 100%, por lo que esta estimación debe tomarse como el promedio global.

2.8 Conclusiones

Con elementos constructivos de bajo coste, del segmento de mercado doméstico, puede construirse con ayuda del software adecuado una infraestructura de procesamiento de alta disponibilidad adecuada para una gran cantidad de negocios.

Tengamos en cuenta que si bien con estos elementos de menor coste no alcanzamos las prestaciones de las soluciones de más alta gama el hardware doméstico actual es tan potente que será posible dotar de virtualización a un gran número de organizaciones y empresas tal como nos planteamos al iniciar este proyecto.

Parte esencial de una arquitectura como la aquí descrita lo sustenta un repositorio de máquinas virtuales en remoto tal que permita que éstas sean accedidas desde múltiples nodos de procesamiento. Hemos visto que este objetivo lo podemos alcanzar con tecnologías de bajo coste: NFS e ISCSI soportadas por varios sistemas operativos gratuitos sobre una plataforma hardware estándar y una red de 1Gbps.

También, hemos constatado que el acceso a través de ISCSI es más eficiente que a través de NFS para archivos y máquinas virtuales de acceso intensivo a disco por lo que preferiremos este protocolo en caso de que sea posible y especialmente para la solución de backup que mueve diariamente gran cantidad de datos.

Por último, hemos comprobado que virtualizar algunos tipos de servidores penaliza el rendimiento lo suficientemente poco como para plantearnos virtualizar la mayor parte de nuestra infraestructura con lo que conseguiremos disfrutar de los mayores ahorros de costes económicos y medioambientales.

En todo caso, se ha sido lo más abierto posible en lo que a elección de tecnologías se refiere y eso ha hecho que se elija para algunos componentes software libre (openfiler), software gratuito (vmware esxi) y software propietario (windows 2003).

3 DISEÑO FINAL DE DESPLIEGUE

3.1 Diseño servidores

3.1.1 Servidor repositorio de máquinas virtuales (VMDK)

Si queremos disfrutar de características avanzadas y muy deseables en pro de la alta disponibilidad tales como (HA) y Migración de máquinas en caliente (VMOTION) es totalmente necesario que las máquinas virtuales de alojen en un almacenamiento remoto y no en los discos duros locales de los ESX.

De esa forma, si un ESX falla las máquinas virtuales que corrían en él pueden inmediatamente ponerse a funcionar en cualquier otro ESX con acceso al mismo almacenamiento.

El software de virtualización elegido, ESXi 4.0, permite utilizar como repositorio remoto almacenes de datos NFS, iSCSI, Fiber Channel y FcoE.

Si descartamos las dos últimas tecnologías por razón del precio y teniendo en cuenta que en los capítulos anteriores se ha comprobado que en la práctica la ganancia de rendimiento es muy cuestionable tenemos que nuestras posibilidades se reducen a NFS e iSCSI.

Tecnología	Ventajas	Desventajas
NFS	Más robusto a fallos breves de conexión Disponible para todos los sistema soperativos	Peor rendimiento en ficheros grandes.
iSCSI	Más rápido para transferencia de bloques y ficheros grandes.	Necesita driver para trabajar con VMFS

(*) VMFS: Sistema de ficheros de VMWARE.

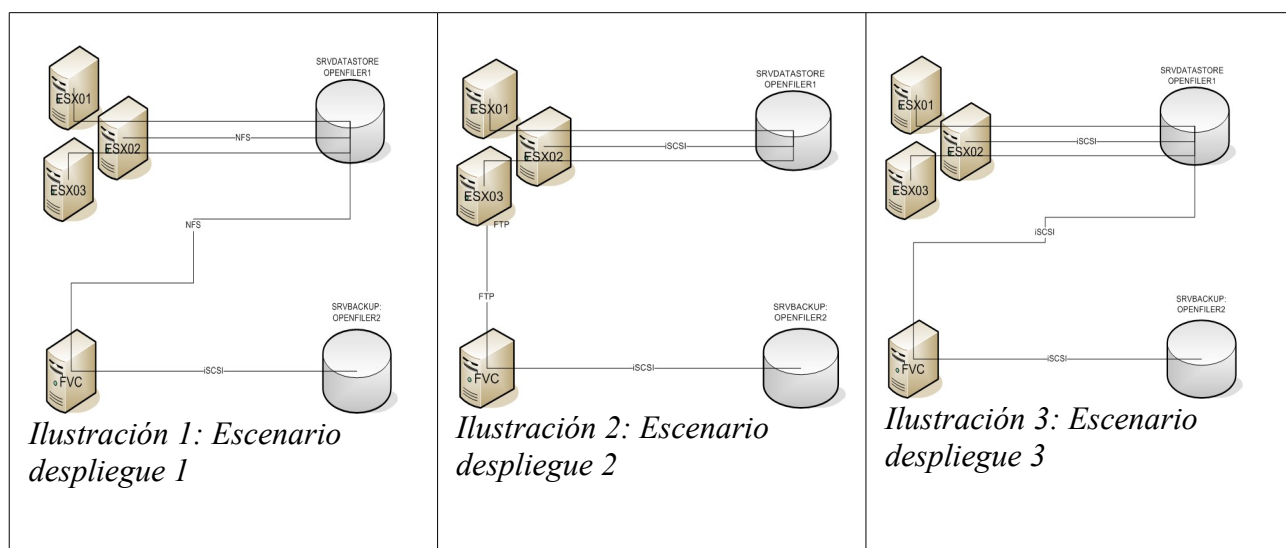
Como servidor de datos vamos a elegir a OPENFILER, software libre especializado en la provisión de servidores de datos a los cuales se puede acceder haciendo uso de múltiples tecnologías ha demostrado en las pruebas anteriores gran estabilidad y facilidad de uso. Ha superado todas las pruebas ofreciendo un rendimiento apropiado y una flexibilidad admirable.

Dicho software se puede instalar en casi cualquier PC y servidor y es capaz de trabajar a nivel de disco tanto con RAID software como con RAID hardware basado en multitud de controladoras avanzadas de disco. También se puede instalar virtualizado, tal como he probado en este mismo proyecto, si bien no parece aconsejable por razones de rendimiento cuando se usa como repositorio de máquinas virtuales.

Para la plataforma aquí diseñada se va a optar por RAID por software ya que nos evita comprar controladoras de disco onerosas y eso nos permite diseñar nuestro sistema con total libertad de elección de servidores. No obstante, si se optara por controladoras de disco con funciones de RAID por hardware se puede optar, por ejemplo, por una Adaptec 2405, producto económico de Adaptec que se puede adquirir por unos 200€.

Teniendo en cuenta que es requisito indispensable que nuestro almacén de datos sea lo más robusto posible a pérdidas físicas de los mismos y que a la vez querríamos conjugar esta necesidad con el máximo nivel de prestaciones posible tenemos que la tecnología RAID más apropiada sería la de RAID 10 ya que aporta redundancia de datos y velocidades comparables a la del rápido RAID 0.

En cuanto a la tecnología y protocolo de acceso remoto al almacén de datos se pueden plantear los tres escenarios siguientes:



En el escenario 1:

- Los servidores de proceso, ESXi, acceden al repositorio de datos a través de NFS
- El servidor de FVC/Backup accede al repositorio a través de NFS y al repositorio de backup a través de iSCSI.

Ventajas:

- NFS disponible para cualquier sistema operativo. Disponible también el Windows 2003.
- Acceso al repositorio de backup por iSCSI. (más rápido)

Desventajas:

- Acceso al repositorio por NFS, más lento que iSCSI.

En el escenario 2:

- Los ESXi acceden al repositorio a través de iSCSI
- y, el FVC accede a los ficheros de las máquinas virtuales por medio de un FTP que se habilita en uno o en todos los ESXi que publica los datos del repositorio.

Ventajas:

- Buena velocidad de acceso de los ESXi al repositorio.

Desventajas:

- Mayor lentitud del backup ya que los datos pasan del repositorio a los ESXi y de éstos por FTP al servidor FVC/Backup.
- Necesario realizar ajustes en los ESXi para poder publicar datos por FTP.

En el escenario 3:

- Todos los servidores acceden a los datos por iSCSI.

Desventaja:

- Necesario driver VMFS. Solo disponible para ESXi.
- Existe una utilidad llamada `fvmfs_r95_dist.zip` que permite acceder por medio de un software libre java a una unidad VMFS pero soporta solo un pequeño número de comandos lo que dificulta el desarrollo de herramientas que trabajen fluidamente con los datos.

El laboratorio se ha montado haciendo uso del escenario 1, todas las máquinas acceden al repositorio por NFS y el servidor FVC/Backup accede al repositorio de respaldo por medio del más rápido iSCSI.

Esta arquitectura ha demostrado ser perfectamente válida y rinde bien en todos los escenarios de prueba, si bien, si las máquinas virtuales que vamos a montar van a tener un acceso muy intensivo a grandes bloques de datos, por ejemplo, por virtualizar servidores de bases de datos o de archivos habría que plantearse montar el escenario 2 que proporciona mayor rendimiento de acceso a los datos en producción y tiene como única pega una pequeña penalización en los tiempos de backup y la necesidad de configurar algún ESXi para que sirva de puente hacia los datos por medio de publicar el repositorio a través de FTP.

En los ANEXOS finales se incluye información detallada de como montar:

- Acceso iSCSI en Windows.
- Acceso NFS en Windows.
- Como instalar OPENFILER
- Como publicar datos por iSCSI y NFS en OPENFILER.

Por último, el servidor dedicado a esta tarea se convierte en un componente crítico por lo que debe ser asegurado también con una fuente de alimentación redundante.

3.1.2 Servidor de backup

Siguiendo razonamientos similares que los utilizados para diseñar el servidor repositorio de datos principal el servidor de backup se montará utilizando OPENFILER, se dará acceso a los datos por medio de iSCSI y físicamente residirá en 4 discos montados en RAID 10 para maximizar a la vez la alta disponibilidad y el rendimiento.

(*) RAID 5 podría ser una opción por su mejor aprovechamiento de espacio de disco pero teniendo en cuenta el bajo precio de los HD actuales no parece adecuado sacrificar velocidad del backup por ganar algo de espacio en disco. Además, el uso de la herramienta de backup aquí desarrollada (AD) minimiza el espacio de disco necesario para mantener gran número de copias de los datos.

Este servidor también será dotado con fuente de alimentación redundante.

3.1.3 Servidor de imágenes ISOS

Para servir imágenes ISO se va a utilizar NFS ya que este protocolo permite el acceso desde multitud de sistemas operativos sin necesidad de instalaciones específicas. Dicho repositorio ISO almacenará imágenes estáticas de instalaciones de sistemas operativos: ESXi, Windows, Solaris, y cuantos software sean necesarios para las tareas de instalación del día a día.

Con el fin de mejorar el performance general en vez de usar una máquina virtual para esta labor se va a utilizar el mismo OPENFILER que mantiene el repositorio de máquinas virtuales configurándolo de tal forma que publique un espacio en disco que contenga las imágenes ISO y al que se acceda por NFS.

3.1.4 Servidor de archivos de usuarios

OPENFILER cuenta con la posibilidad de servir datos por CIFS, el protocolo de compartición de archivos de windows, y se puede integrar con el directorio activo para una seguridad centralizada por lo que es una opción a tener en cuenta en lo que a servidor de archivos se refiere.

En los ANEXOS finales aporto los pasos necesarios para configurarlo de esta manera.

No obstante, el uso para esta función de máquinas virtuales aporta grandes beneficios: ahorro de

costes, menor consumo energético, las incidencias no dejan fuera de servicio al servidor de archivos tanto tiempo como si se tratara de máquina física, alta disponibilidad, etc.

Por ello, el servicio de archivos lo voy a montar aquí como máquina virtual y en el sistema operativo que prefiera el usuario final teniendo en cuenta el resto de su red por lo que se montaría como servidor windows, linux+samba, linux, openfiler virtualizado, etc.

3.1.5 Servidor de bases de datos

Los servidores de bases de datos virtualizados, en mi experiencia, funcionan muy bien al menos: Oracle 10 y 11g, SqlServer 2005 y 2007 y MySQL.

No obstante, hay que ser conscientes de algunas limitaciones:

La virtualización de servidores de bases de datos degrada el rendimiento de éstos en un 15% (según pruebas realizadas con servidor SqlServer 2005)

Las tecnologías de cluster no son perfectamente compatibles con la idiosincrasia de la virtualización y se aconseja no clusterizar, al menos con los servicios de cluster de microsoft, máquinas virtuales con VMWARE.

Por lo tanto, yo propongo aquí sí virtualizar los servidores de bases de datos siempre y cuando se monten como servidores standalone y no como máquinas integradas en un cluster y siempre que contemos con potencia adecuada de las máquinas físicas de forma que se compense el menor rendimiento por estar virtualizadas.

Asumiendo ese compromiso, virtualizar los servidores de bases de datos aporta todo el resto de beneficios de la virtualización.

3.1.6 Servidor FVC

Desde dicho servidor se comandará el resto del sistema y hará las funciones de supervisor del backup. Podría montarse virtualizado pero es conveniente hacerlo en máquina física para maximizar el rendimiento del backup, y para evitar sucesos de “pescadilla se muerde la cola” donde es necesario usar el FVC para levantar máquinas virtuales pero a su vez el propio FVC es también una máquina virtual.

En cuanto a disco local es necesario montar el FVC sobre RAID 1 para mejorar la alta

disponibilidad y disminuir la probabilidad de que esta importante máquina quede fuera de servicio.

Tanto si se monta sobre Windows, como si lo hace sobre LINUX o SOLARIS no es necesario adquirir controladora RAID hardware ya que como han demostrado en las pruebas estos sistemas operativos se valen por si mismos estupendamente a la hora de montar RAID por software.

Con el mismo fin debería dotarse al servidor de FVC de fuente redundante ya que es el otro elemento que junto al disco duro más suele fallar en un servidor.

3.1.7 Servidores ESXi

ESXi se instala sin mayores complicaciones en gran variedad de equipos pero a nivel de disco tiene algunas limitaciones importantes: No es capaz de montar RAID por software, ni FAKE RAID, por lo que si queremos disfrutar de redundancia de datos necesitamos montar una controladora adecuada. Tarea ésta que no es trivial ya que no todas están soportadas. (Intentando soslayar este inconveniente he investigado al posibilidad de arranque sin discos, por PXE, y si bien he conseguido arrancar clusters de DEBIAN, incluso cualquier ISO arrancable no he podido montar ESXi de esta forma).

El modelo 2405 de Adaptac sí lo está y aunque no da un rendimiento especial si permite montar RAID 1 lo que aportaría mayor robusted a la solución diseñada.

En la propuesta siguiente sí establezco dos discos por servidor ESXi con la idea de montar RAID 1 si bien no sería un locura montar solo un disco y prescindir de RAID y por tanto de la necesidad de adquirir costosas controladoras. Tengamos en cuenta que el acceso a disco de los ESXi cuando las máquinas virtuales están en remoto es muy escaso y de perderse un disco solo afectaría al rendimiento (porque no se podría hacer uso de la partición de swap) pero no necesariamente se quedaría instantáneamente fuera de servicio la máquina ESXi.

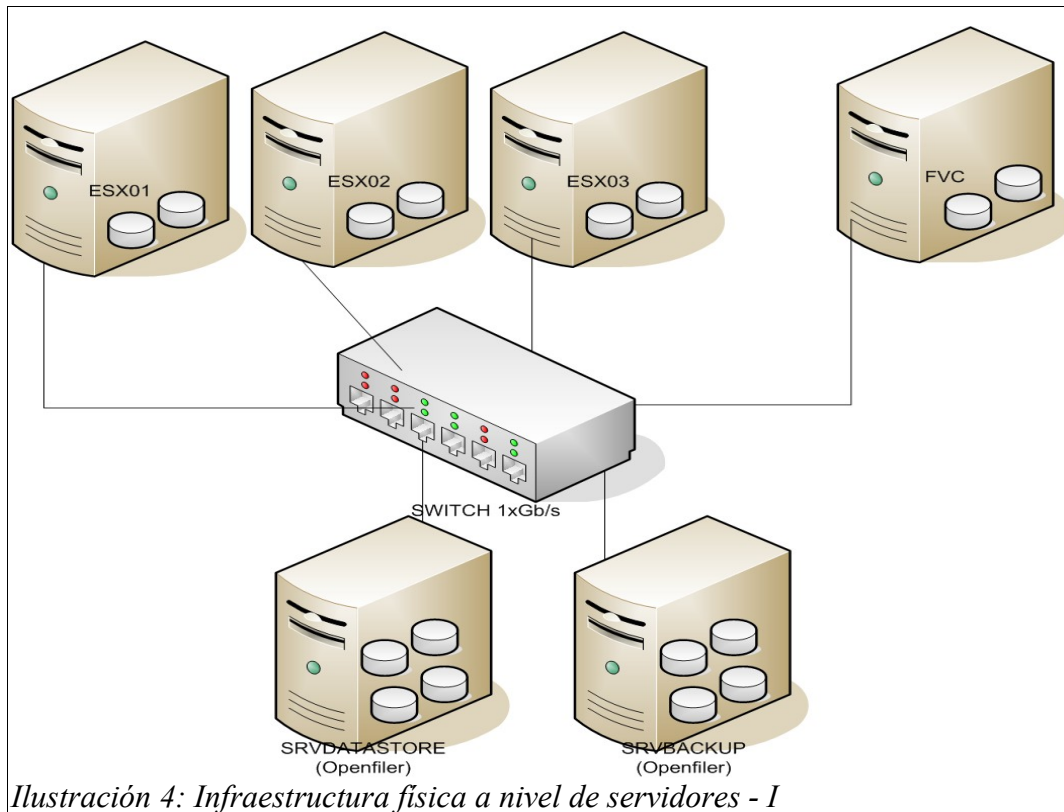
En cuanto a la fuente de alimentación su inclusión o no en los servidores ESXi es una cuestión puramente económica: Es necesaria: No, pero muy aconsejable sí. El precio de una fuente redundante con dos módulos ronda los 500€ lo que encarece la adquisición de servidores de esta gama.

En cuanto a memoria ram una de las lecciones que antes se aprende cuando se adentra uno en el mundo de la virtualización es que es necesario contar con grandes cantidades de memoria ram. Las máquinas aquí presentadas suponiendo que se doten con CPU de domésticas, de 4 cores, se montarían con 16-32 GB de memoria. (Pnematécnico: 4GB-8GB por CORE)

3.1.8 Diseño infraestructura física

Teniendo en cuenta todo lo anterior, se proponen las siguientes infraestructuras físicas para la implantación de los servidores:

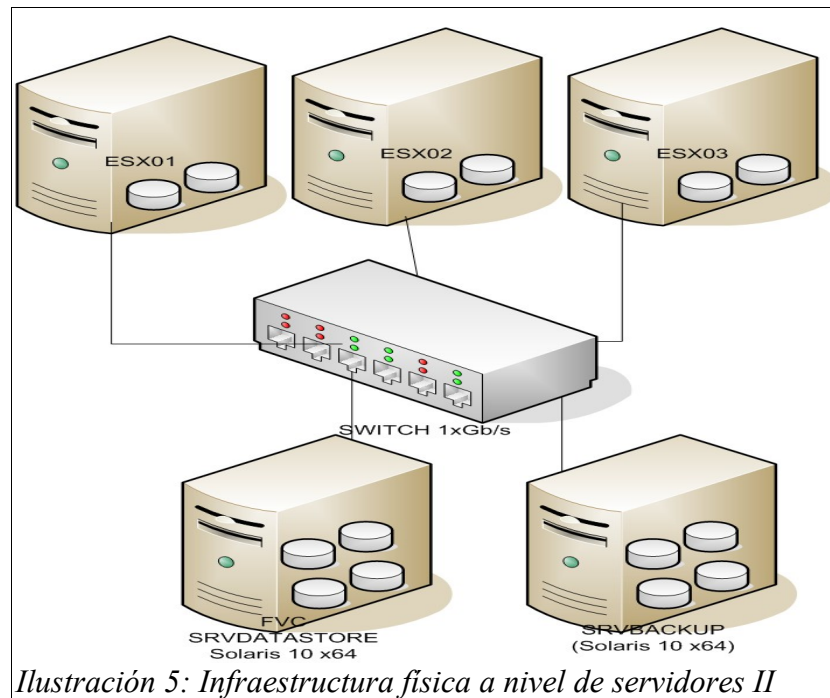
Propuesta I:



La infraestructura física I tiene como ventajas que:

- El servidor FVC es independiente del sistema de almacenamiento por lo que se tiene mucha flexibilidad a la hora de elegir el hardware tanto de éste como de la máquina que soporta el almacenamiento.
- El almacenamiento puede montarse sobre Openfiler, software libre que se instala fácilmente sobre multitud de hardware distinto incluso sobre placas base MicroATM, de ultra bajo consumo:10W permitiendo montar todo el sistema descrito sobre 3 cajas rack de 19” con 2 placas ATM/MicroATM cada una:
 - Caja 1: ESX01 y ESX02
 - Caja 2: ESX03 y FVC
 - Caja 3: SRV DATASTORE y SRV BACKUP

Propuesta II:



La infraestructura física II, presenta como mayores ventajas que:

- El FVC accede a los discos de datos para hacer el backup a velocidad de acceso local lo que acelera notablemente las copias de seguridad.
- Además, por estar montado sobre Solaris 10, presenta mayor rendimiento en el acceso al datastore por NFS desde los ESX tal como han demostrado las pruebas realizadas. Por ello, este diseño se considera de mayor rendimiento que el anterior.

Desventajas:

- Solaris no es software libre,
- No es tan sencillo de instalar y administrar que Openfiler.
- Tiene mayores necesidades de CPU, no corre sobre C3 de ultra bajo consumo.

3.1.9 CONCLUSIONES

Si es posible instalar Solaris 10 x64 en nuestro hardware y no queremos centrarnos solo en software libre la propuesta II es la que mejor rendimiento nos va a ofrecer.

El laboratorio lo he montado siguiendo la propuesta I para garantizar la flexibilidad del FVC pero el código desarrollado funcionaría sin cambios en la propuesta II.

3.2 Diseño elementos de apoyo

3.2.1 Red de comunicaciones internas

Una red de 1Gbps se considera perfectamente dimensionada para los componentes aquí utilizados y es difícil que se vea saturada en ninguna aplicación.

No obstante, hay algunas mejoras muy económicas y que aumentan al performance a la vez que la disponibilidad. A saber: Puede configurarse VMWARE ESXi de forma que utilice varios enlaces físicos de red como si fuera uno solo, de forma que se consiga disponer del ancho de banda conjunto de todas las tarjetas de red agrupadas y en caso de incidencia el tráfico de red se redirecciona por la que está OK de forma que no se producen cortes.

Para esta implantación se va a configurar cada VMWARE ESXi para usar dos tarjetas de red de 1Gbps por cada host, en modo BEACON PROBING que aporta la máxima fiabilidad en redundancia de enlaces.

Por su parte, los equipos windows utilizados se van a configurar con dos tarjetas de red de 1Gbps haciendo uso de la capacidad de Microsoft llamada TEAM que permite agrupar varios enlaces físicos en uno solo.

El precio de un switch de 1Gbps de 12 bocas no supera los 150€ y de 200€ en caso de 24 bocas.

Las tarjetas de red de 1Gbps se añadirían individualmente a cada Host para completar la tarjeta que ya suele venir integrada. El precio ronda unos 10€/tarjeta.

3.2.2 Conexión eléctrico

Los equipos aquí tratados tienen un consumo máximo que ronda los 400/500W por equipo lo que nos da una potencia máxima conjunta de 3 KW.

Las regletas de conexión que usaremos deberán estar dimensionadas como mínimo para esa potencia con un cable conductor de 16mm de cobre.

3.2.3 SAI

El equipo SAI que asegurará el suministro deberá estar dimensionado para esta potencia y soportar dicho consumo por un período mínimo de 15 minutos, suficiente para un apagado controlado.

Existe un gran rango de calidades de equipos SAI en el mercado dependiendo del modo de conversión que utilicen, del tiempo de reacción en caso de corte de suministro, etc.

Aquí vamos a dotar a nuestra infraestructura de un SAI modelo DLP500 de la empresa Enerdata.

Precio: 1000€ para el modelo enrackable.

3.2.4 Potencia frigorífica

Los equipos aquí tratados no tienen unas necesidades de refrigeración muy estrictas ya que la baja potencia total con la que trabajan hace que sea posible disipar la energía que producen en forma de calor por medios convencionales y muchas veces pasivos. No obstante, en caso de que se instalen en una habitación o recintos cerrados será obligatorio dotar a dicho recinto con unidades de refrigeración adecuadas: un extractor para los meses fríos y una unidad de aire acondicionado para el resto del año. Modelos y tecnologías existen muchas pero un nemotecnico adecuado para calcular a *grosso modo* que potencia deben tener los equipos de aire acondicionado para disipar una potencia X nos lo da la ecuación: $\text{Potencia frigorífica} = \text{potencia equipos} / 3$.

De esta forma, vemos que tendríamos que contar con un equipo de aire acondicionado de unos 1000W para mantener la temperatura en márgenes adecuados todo el año, por mucho calor que haga en el ambiente exterior.

4 DESARROLLO DEL SOFTWARE (FVC)

4.1 Introducción

VMWARE proporciona, para la versión Enterprise de sus productos, un software llamado Virtual Center (VC) que es capaz de trabajar de una forma unificada con toda la infraestructura virtual de una organización gestionando características avanzadas que en las versiones gratuitas de sus productos no están soportadas oficialmente.

El objetivo del desarrollo del Free Virtual Center (FVC) es proporcionar para las versiones gratuitas de VMWARE ESXi, funcionalidades lo más parecidas posibles a las proporcionadas por el VC, e incluso, en algunos temas más avanzadas o integradas con el entorno unificado.

Complementará los interfaces de acceso web a una sola máquina física que ofrece VMWARE y que aportan mucha funcionalidad para máquinas independientes pero están intencionadamente limitados cuando queremos disfrutar de una plataforma multihost y con características empresariales interesantes.

4.2 Objetivos

Se consideran objetivos mínimos alcanzables los siguientes:

Funcionalidad similar a la proporcionada por VC: (en lo posible)

- Visión unificada de todas las máquinas físicas y virtuales del entorno ESXi
- Plantillas/clonado
- Alta disponibilidad (HA)
- Migración de máquinas virtuales entre máquinas físicas (en caliente: VMOTION)

Mejoras respecto a VC:

- Backup centralizado con deduplicación
- Monitorización centralizada. Avisos a responsables en caso de incidencia

4.3 Alcance y entregables

Al finalizar el proyecto, se deberá contar con la documentación de análisis y diseño de la herramienta, el software desarrollado, la base de datos, servidor de correo y otros elementos interrelacionados con el sistema.

4.4 Gestión de la configuración

Como entorno de desarrollo se usará NetBeans en su versión 6.7.1 y como repositorio de código se usará Subversion. Todos los días se hará backup de la máquina de desarrollo que estará montada como máquina virtual Windows 2003, y correrá sobre VMWARE WORKSTATION y/o VMWARE ESXi 4.0.

Los diagramas de análisis y diseño de la aplicación serán realizados con las herramientas ArgoUML 0.30.2 y Visio 2003.

Para la redacción de la documentación se usará OpenOffice, versión 2.0.3.

Los gráficos serán tratados con GIMP 2.2.

Como API para acceso a funcionalidades VMWARE: VIX (VMware-vix-1.10.1-266898.exe)

Como servidor de aplicaciones se usará TOMCAT 6.0.

JAVAMAIL1.4.2

HSERVER2, servidor de correo.

4.5 Diagrama de despliegue

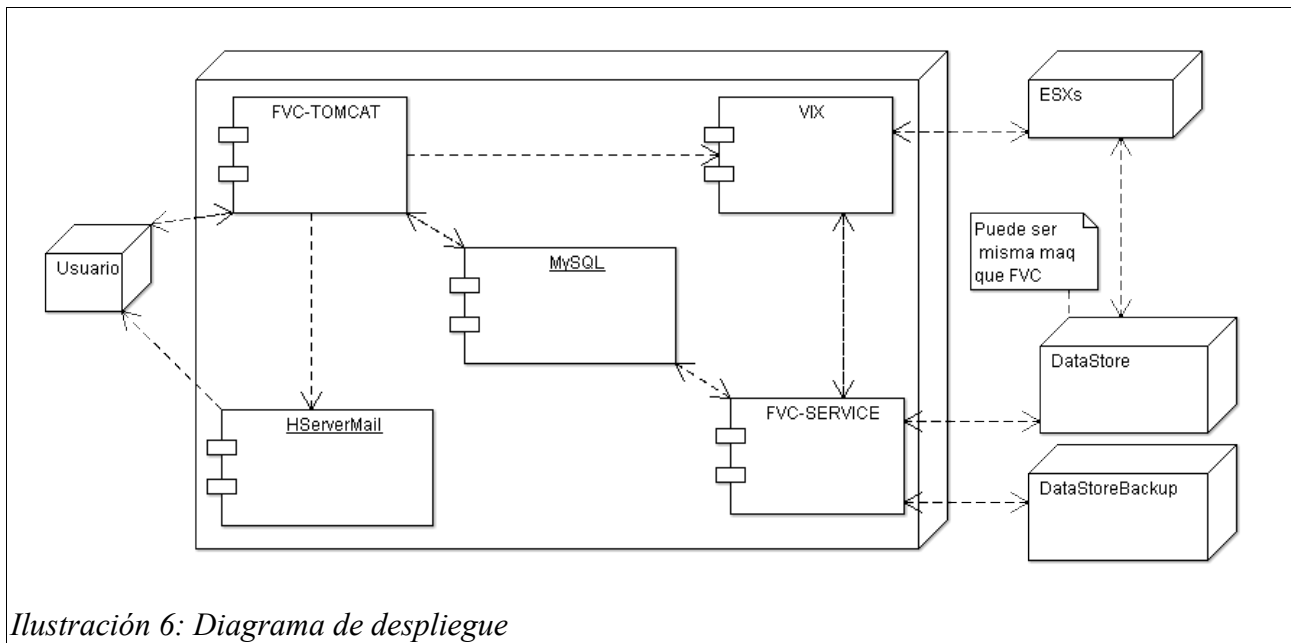


Ilustración 6: Diagrama de despliegue

La aplicación principal, el **FVC**, residirá en un servidor de aplicaciones, Tomcat, y será el interface principal con el usuario.

Para enviar mensajes a éste además de vía web se hará uso de un **servidor de correo** embebido en la máquina del **FVC**.

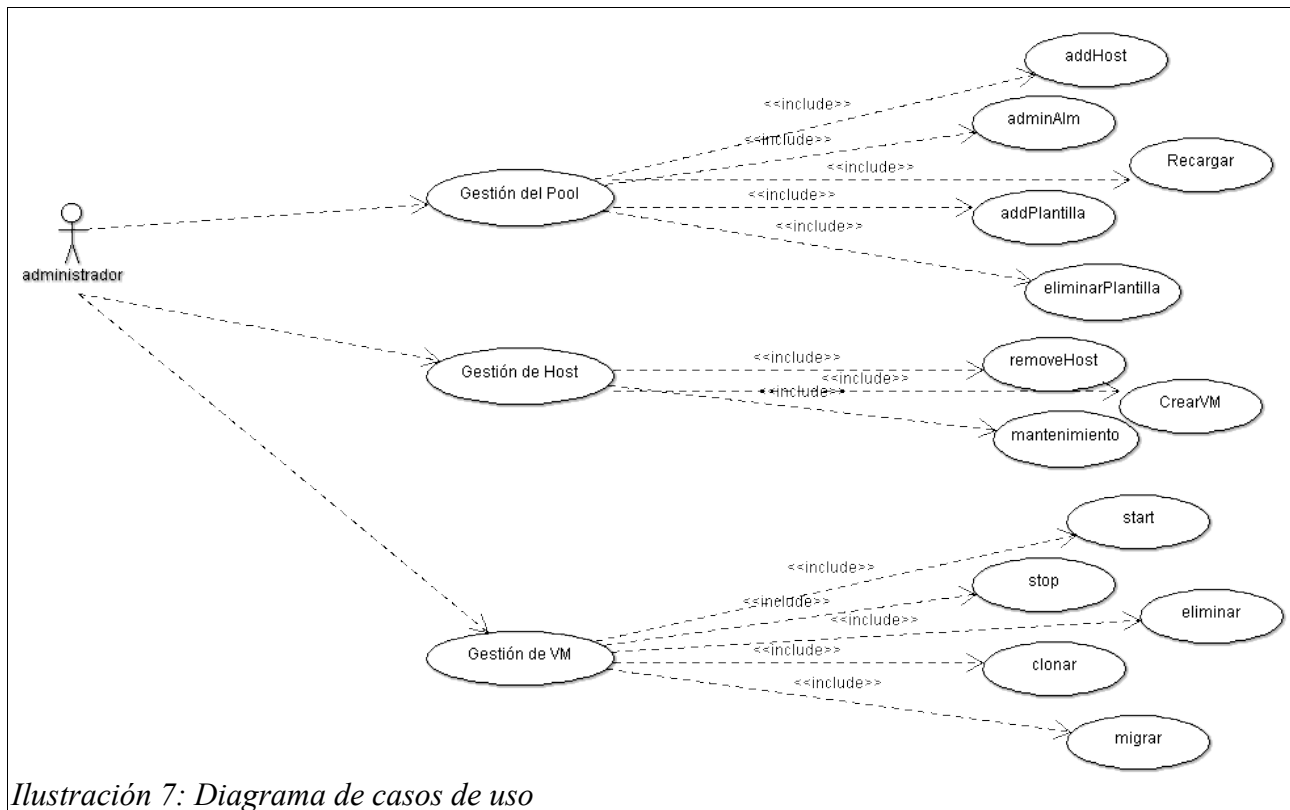
Todos los datos se guardarán, igualmente, en una bases de datos local, **MySQL**, que hará de repositorio de configuración, eventos, etc.

Las órdenes a la infraestructura **VMWARE** se cursarán a través de un componente llamado **VIX**, que proporciona **VMWARE** gratuitamente (previo registro).

El componente **FVC-SERVICE** correrá como aplicación en modo consola (para depuración) o como servicio para monitorizar constantemente la plataforma y para realizar los backups.

Por último, los **datastores** podrán residir en máquinas distintas a la del **FVC** o en la misma, dependiendo del modelo de arquitectura elegido y de los recursos hardware con los que contemos.

4.6 Casos de uso



La **Gestión del Pool** se encarga de mantener todos los datos necesarios para la gestión diaria de éste: email del responsable, nombre y ubicación de las plantillas, rutas de máquinas virtuales y de la carpeta de backup, etc..

Para plantillas podremos: dar de alta nuevas, modificar las existentes y eliminar plantilla.

Podremos dar de alta un nuevo host: máquina física que componen nuestro pool.

Se incluye aquí también la visualización de los eventos del pool.

La **Gestión de Host**, permite mantener los datos de éste: nombre, ip, usuario, contraseña, etc...

Debe permitir eliminar el host del pool.

Debe permitir crear una nueva máquina virtual que correrá sobre él.

Y se proporcionará aquí también la funcionalidad de pasar a mantenimiento el host, para que el sistema vacíe éste de VM y permita así su apagado para poder realizar tareas de mantenimiento o traslados. También se proporcionará la funcionalidad pareja: Salir de mantenimiento para declarar que el Host ya está dispuesto a aceptar máquinas virtuales.

La **Gestión de VM** permitirá realizar las acciones que sean de aplicación a éstas, como puede ser: arrancar, parar, eliminar, clonar y migrar.

Clonar: realizar una copia independiente de una máquina virtual en otra máquina nueva.

Migrar: Cambia de máquina física la máquina virtual. Si esta operación la hacemos con la máquina virtual funcionando (en caliente) llamaremos a la funcionalidad: VMOTION.

4.7 Decisiones de diseño

La aplicación se va a implementar en JAVA, utilizando para ello la versión 1.6 de JDK, correrá sobre un servidor de aplicaciones Tomcat 6.0 y hará un uso intensivo de AJAX con el fin de actualizar asíncronamente la pantalla del usuario.

Quizás, lo más complicado de la aplicación será la necesidad de implementar mecanismos que desacoplen las acciones invocadas por el usuario con la realización de las mismas sobre la plataforma virtual ya que hay operaciones muy lentas: varios minutos, incompatibles con un uso ágil de la aplicación.

Por ello, se va a implementar internamente un mecanismo de ejecución asíncrona de forma que las peticiones del usuario son contestadas inmediatamente con mensajes del tipo “Ej. Arrancando...” , el sistema realiza y comprueba las acciones en segundo plano, y cuando están realizadas el interface del usuario se refresca con la información actualizada “Ej. Disponible”.

El sistema en su conjunto se ha pensado teniendo en mente una caja negra que ofrece servicios, un aplicance, que contendría el servidor de aplicaciones, el servidor de correo, el servidor de bases de datos y el api VIX. Por ello, todas las llamadas a estos elementos se realizan localmente y permiten relativa poca configuración ya que se asume siempre su existencia y localización.

Por ejemplo, el sistema da por hecho que en la máquina local al FVC existirá un MySQL, con una base de datos llamada FVC, usuario y clave prefijados, etc...

Por último, se ha preferido no hacer uso ni de framework de presentación ni de persistencia ya que el diseño sencillo que sigue la solución parece que permite afrontar ésta directamente sin necesidad de implementar mecanismos más potentes.

No obstante, sí se ha hecho uso de funcionalidades de pool de conexiones tal como se explica en detalle en los diagramas de clases y su explicación anexa.

A nivel de navegador cliente el programa va a hacer uso de características de CSS 3.0 por lo que se visualizará correctamente en navegadores que hagan un uso correcto del estándar: ej. Firefox a partir de la versión 2.8.

El funcionamiento en Chrome y Safari es prácticamente similar y en IE 8.0 es usable pero el aspecto empeora bastante.

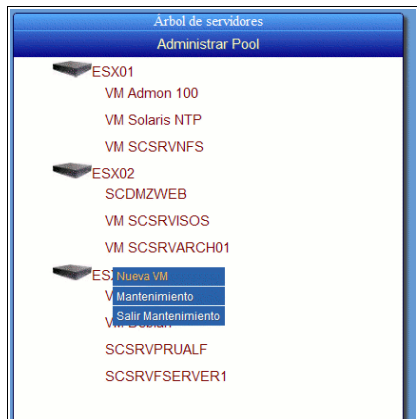
4.8 Prototipo de pantallas

La pantalla se compone de dos áreas: la izquierda presentará un árbol con las máquinas físicas y virtuales que componen nuestro pool y la derecha variará dependiendo de la función que se esté realizando. Podemos desplegar/contraer cada rama del árbol pinchando sobre la máquina física correspondiente.

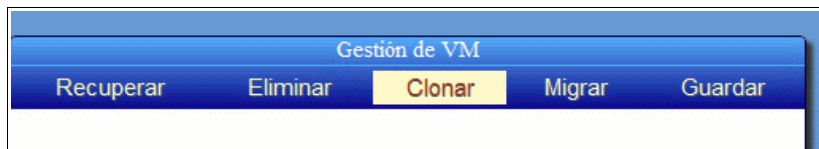


Ilustración 8: Prototipo pantalla principal del FVC

Para seleccionar un elemento haremos click sobre él o pulsaremos botón derecho del ratón. En este último caso aparecerá un menú contextual con las acciones que podremos realizar sobre el elemento. Se detalla menú contextual para HOST y para VM:



En la parte superior de cada área contamos con un menú que presentará también las acciones a realizar sobre el elemento. Cuando pasamos con el ratón sobre algún ítem del menú éste se resaltará en otro color.



La pantalla de **Gestión del Pool** presentará el siguiente aspecto:

Datastore indica donde se almacenan las VM, la ruta desde el FVC.

Responsable almacena el email de la persona que adminsitra el FVC.

Hora backup indica cuando deberá empezar éste.

Path backup indica la ruta, desde el FVC, donde residirá el fichero AD con los backups.

Si pulsamos guardar la base de datos se actualiza con el contenido actual de las cajas de texto.

Para dar de alta una plantilla nueva basta con escribir en la última línea del apartado plantillas.

Por último, para borrar una plantilla se de-seleccionará la casilla de su izquierda.

Gestión del Pool

Nuevo Host Recargar Guardar

DataStore:

Responsable:

Hora backup:

Path backup:

Plantilla

<input checked="" type="checkbox"/>	<input type="text" value="Windows 2003"/>	<input type="text" value="D:\plantillas\Windows2003w"/>
<input checked="" type="checkbox"/>	<input type="text" value="Solaris 10 x64"/>	<input type="text" value="D:\plantillas\Solaris\solaris11"/>
<input checked="" type="checkbox"/>	<input type="text" value="Oracle Solaris 11"/>	<input type="text" value="D:\plantillas\Solaris\solaris11"/>
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>

Eventos:

Evento	Hora/Fecha
Arrancando... VM Admon 100	10:27 06/01/11
Parando... scsrvmegamatcher09	16:34 05/01/11
Arrancando... scsrvNFSpxe	16:34 05/01/11
Arrancando... VM Gestion 02	14:28 05/01/11

Ilustración 9: Prototipo pantalla gestion del pool

Si pulsamos sobre Nuevo Host nos aparecerá una pantalla para poder introducir los datos necesarios:

Añadir HOST

Añadir al Pool Cancelar

Nombre:

IP:

Usuario:

Password:

Desde la pantalla de **Gestión de Host** tenemos acceso a los datos de éste, su estado actual, los últimos eventos que se hayan producido sobre la máquina o sobre las máquinas virtuales que residan en él, podremos pasar a mantenimiento la máquina física (dejarla vacía de virtuales para poder apagarla , repararla, ampliarla...) y crear una nueva VM.

Gestión del HOST

Crear VM Eliminar Host Mantenimiento Guardar

Nombre: ESX02 Disponible

IP: 10.0.0.2

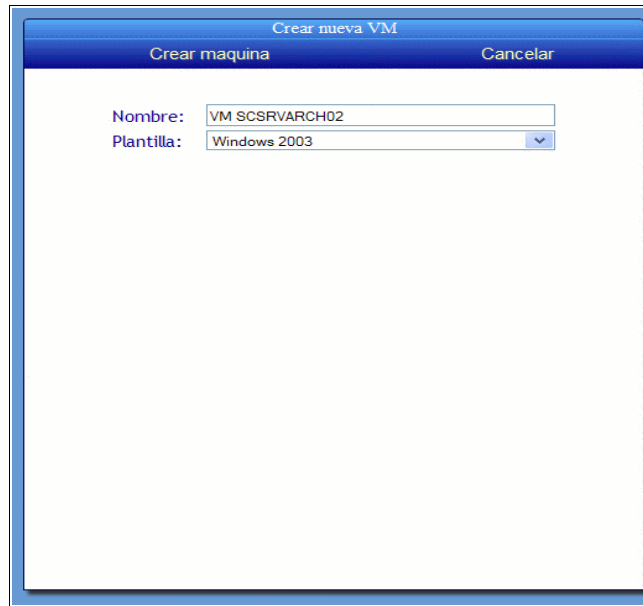
Usuario: root

Password: firefox1025

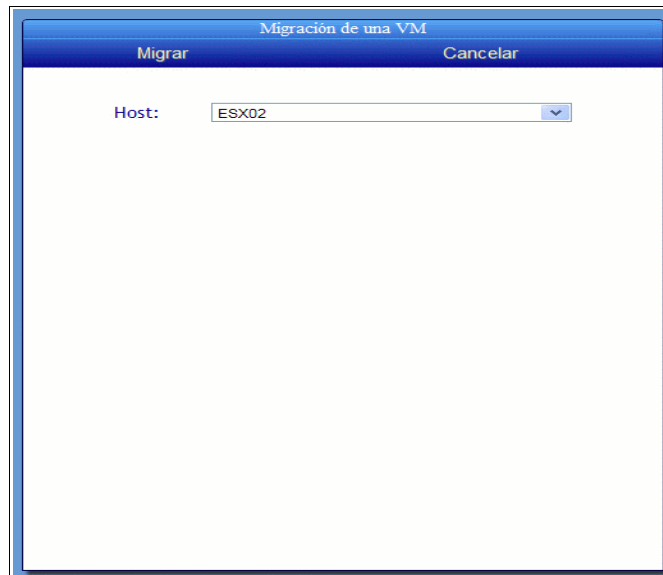
Evento	Hora
Arrancando... VM SCSRVISOS	14:28 05/01/11
MODO NORMAL en ESX02	10:57 05/01/11
MIGRANDO VM SCSRVISOS	19:31 04/01/11
MIGRANDO VM SCSRVISOS	19:30 04/01/11
Parando... VM SCSRVARC01	19:29 04/01/11
MIGRANDO VM SCSRVARC01	19:27 04/01/11
MIGRANDO VM SCSRVISOS	19:26 04/01/11
MIGRANDO VM SCSRVISOS	19:26 04/01/11
MIGRANDO VM SCSRVISOS	19:25 04/01/11
Arrancando... VM SCSRVISOS	19:21 04/01/11

Ilustración 10: Prototipo pantalla gestion del host

Crear una nueva máquina virtual es tan sencillo como elegir una plantilla sobre la cual se basará y elegir el nombre de la nueva máquina virtual.



La migración de una máquina virtual requiere que establezcamos la máquina física hacia donde queremos migrar la virtual:



Por último, la pantalla de **Gestión de máquina virtual** presentará el siguiente aspecto: Se podrá establecer el nombre de la máquina, su IP, un email del responsable, ver o recuperar cualquiera de los últimos backup y observar los últimos eventos que haya sufrido la máquina virtual.

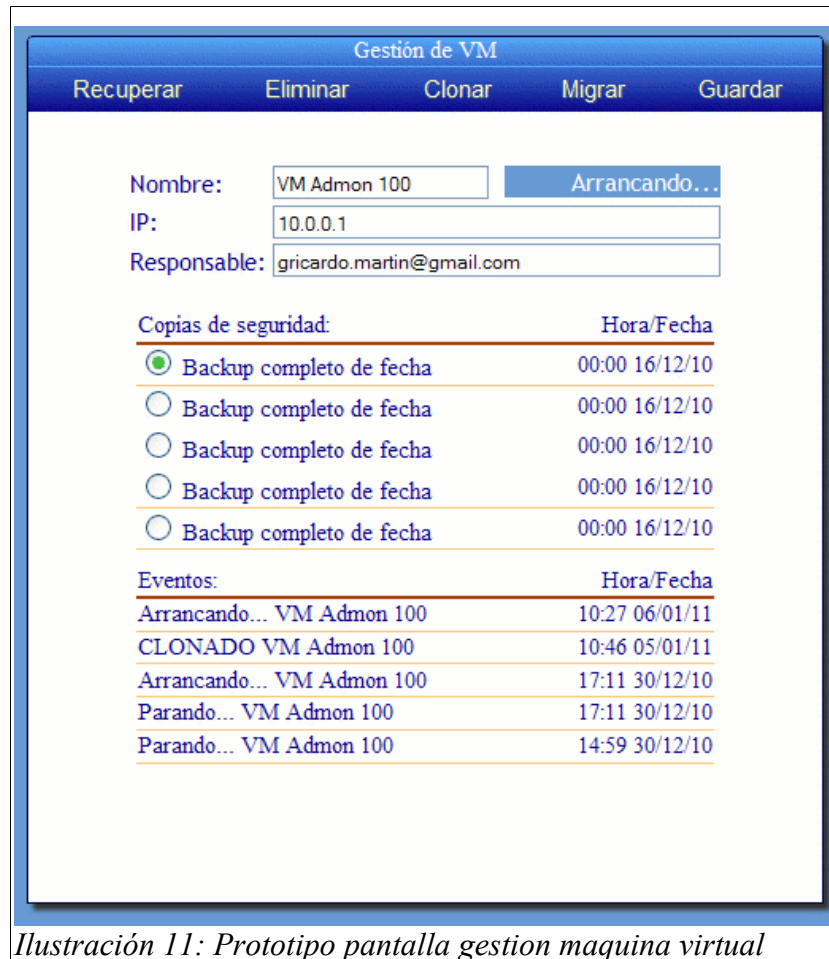
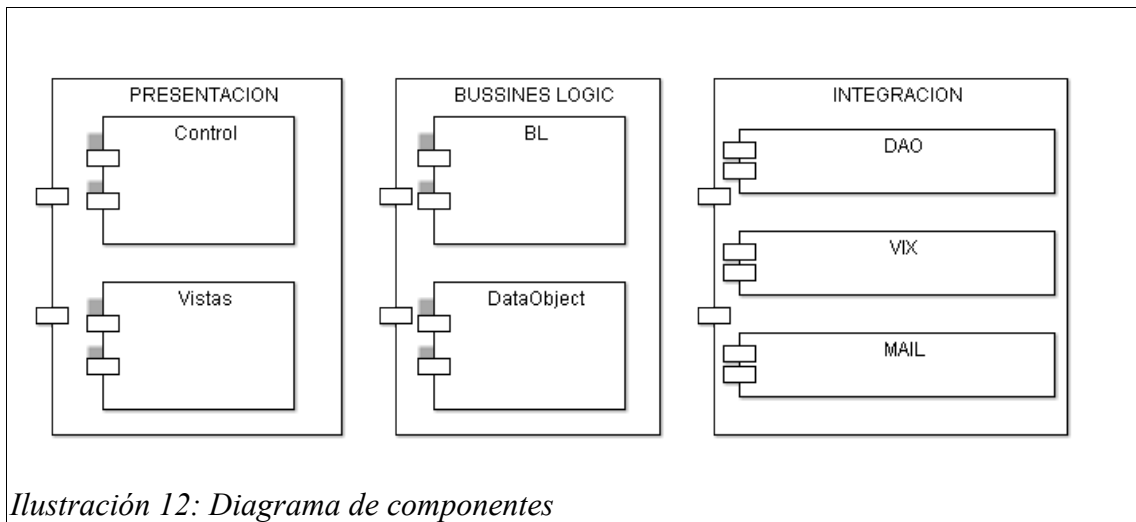


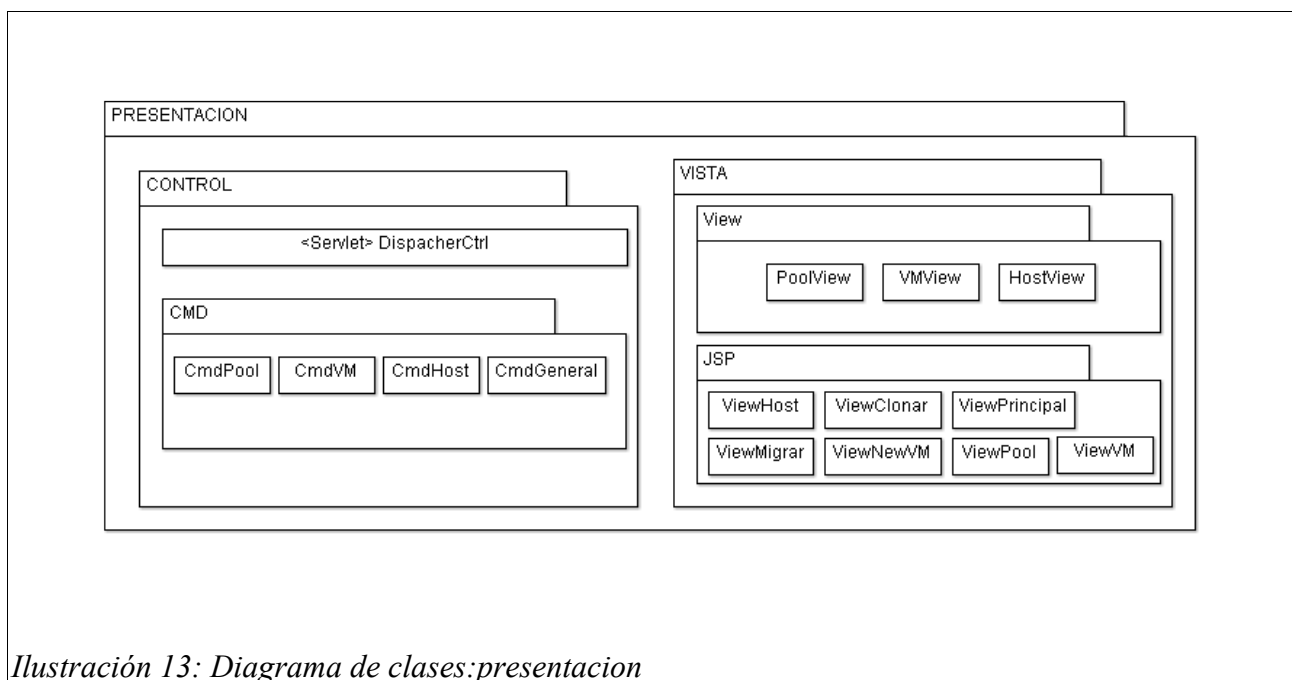
Ilustración 11: Prototipo pantalla gestion maquina virtual

4.9 Diagramas de componentes



Se va a seguir el patrón de diseño, MVC, Modelo Vista Controlador, dividiendo así las funcionalidades del proyecto en capa de presentacion: control+vista, lógica de negocio e integración.

4.10 Diagramas de clases



La **capa de presentación** está dividida a su vez en Control y en Vista.

Para el Control se ha seguido un patrón llamado “FrontController” que se basa en proporcionar un único servlet para unificar las llamadas. También se hace uso del patrón “Acción/Command” que permite organizar mucho más convenientemente las acciones desencadenadas por cada orden llegada desde el interface del usuario.

Para la vista se han proporcionado unas clases: PoolView, HostView y VMView que tienen como responsabilidad establecer los atributos necesarios para la presentación propiamente dicha que se realiza por medio de páginas JSP.

La actualización de pantalla se realizará siempre por medio de tecnologías AJAX, asíncronas, pintando solo las zonas de la pantalla necesarias y actualizando periódicamente ésta.

La simplicidad del diseño ha permitido afrontar éste directamente sin hacer uso de framework de presentación.

La capa de Lógica de Negocio sigue un patrón “Fachada”, siendo el único interface de métodos de negocio con el exterior la clase FVCBL.

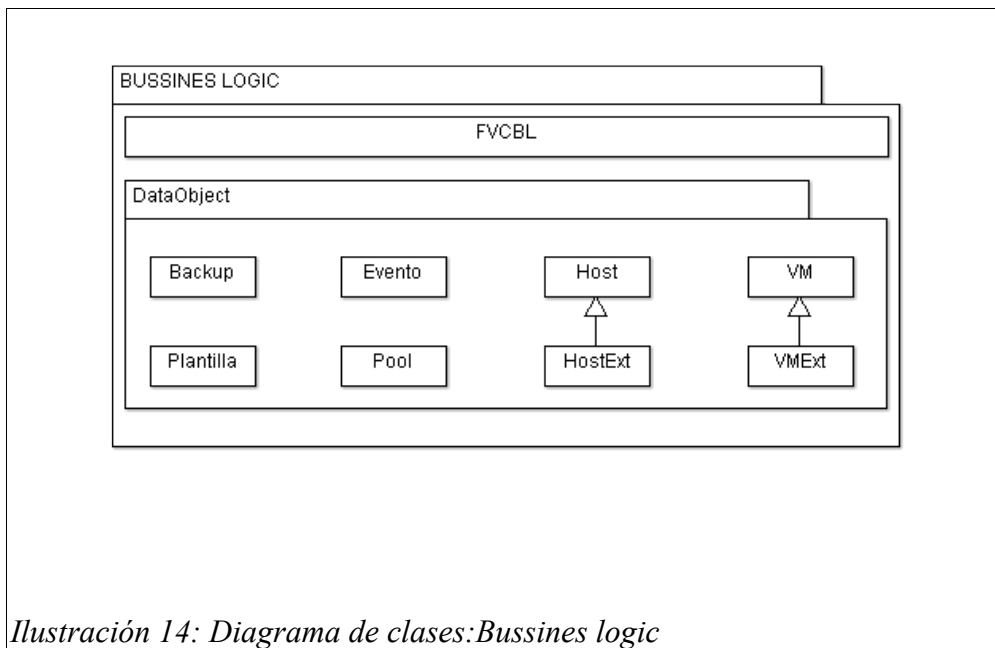


Ilustración 14: Diagrama de clases: Bussines logic

La **capa de integración** se ha realizado en base a objetos POJO para el acceso a base de datos, no siendo necesario el uso de framework de persistencia ni de tecnologías bean.

Todas las clases DAO heredan de una llamada BaseDAO que proporciona algunos servicios a las clases especializadas y sobre todo proporciona el mecanismo de pool de conexiones embebido.

Este pool consta de un diseño muy simple que se basa en el siguiente algoritmo:

Cuando se solicita una conexión si se tiene alguna cacheada y es válida se devuelve ésta.

En caso contrario se crea nueva.

Cuando se devuelve una conexión, si no se ha superado el límite máximo del caché, se almacena para nuevos usos y en caso contrario se cierra.

Este diseño permite hacer uso de cuantas conexiones necesite nuestro sistema sin más límite que el establecido en la base de datos y por otro lado necesita pocos recursos para almacenar y gestionar el pool de conexiones. Salvo en casos excepcionales de concurrencia este esquema consigue fácilmente reutilizar las conexiones salvando así recursos y optimizando el tiempo de respuesta.

Se ha preferido este diseño frente al uso de pool de conexiones del servidor de aplicaciones para hacer la aplicación más transportable.

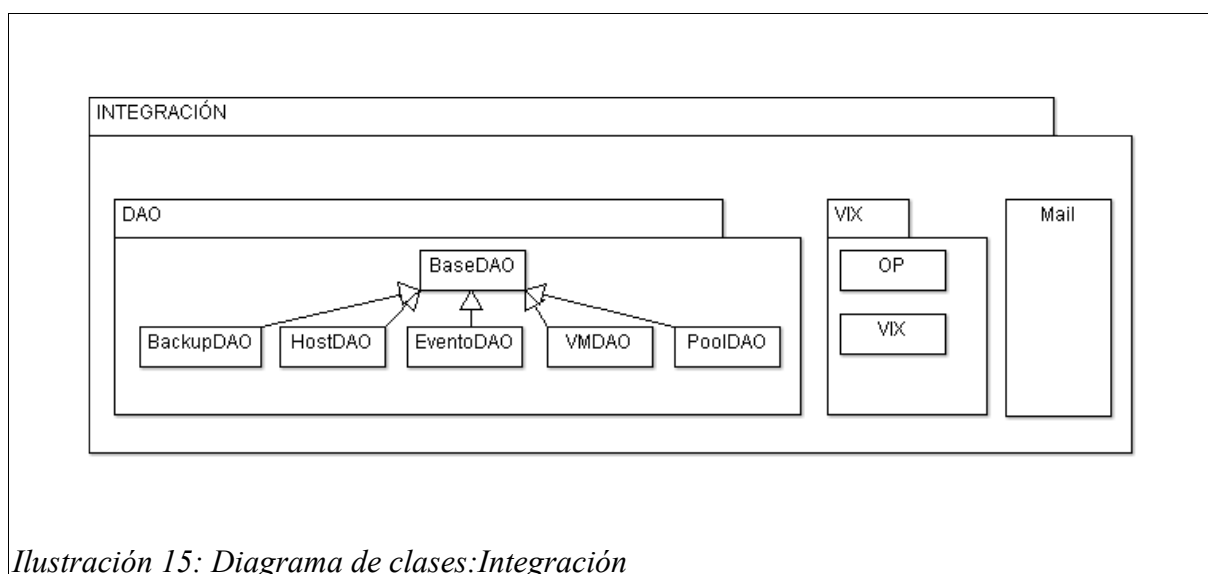


Ilustración 15: Diagrama de clases: Integración

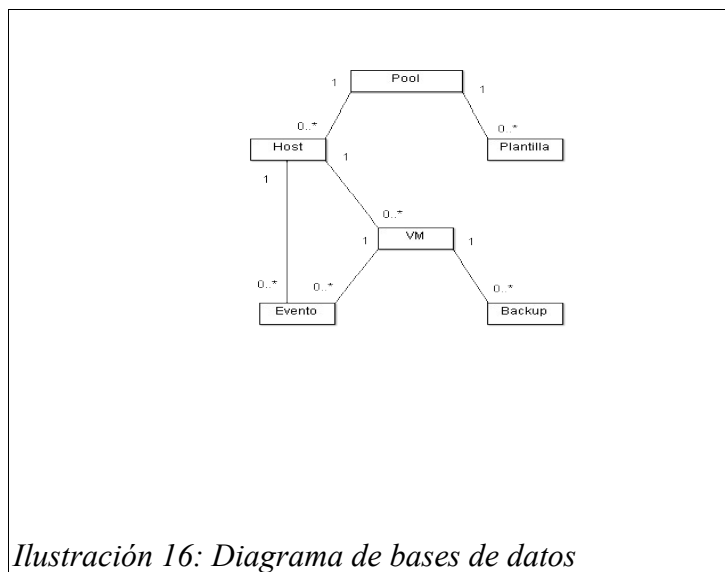
Otro componente a destacar es la clase VIX que encapsula las llamadas a VMWARE haciendo uso de un API del mismo nombre.

En caso de decidir en un futuro implementar el control de VMWARE con otra tecnología o incluso usar otro sistema de virtualización solo sería necesario cambiar dicha clase.

Por último, una de las funcionalidades que se ha implementado es una cola de operaciones asíncronas de forma que las instrucciones al API VIX se van despachando en orden, pero en segundo plano respecto a la llamada invocadora.

La clase UtilMail se utiliza para envió de correos a los responsables tanto de las máquinas físicas (administrador del pool) como de cada máquina virtual.

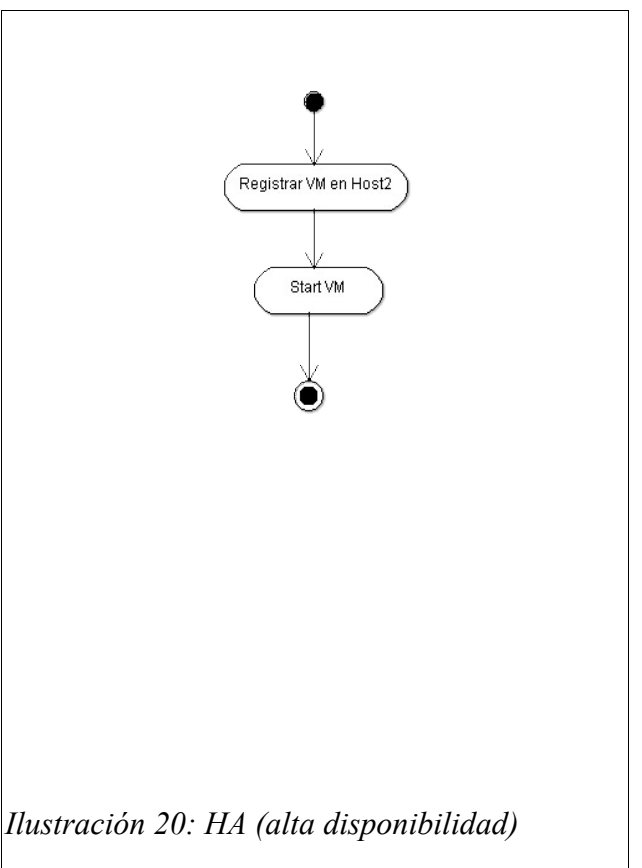
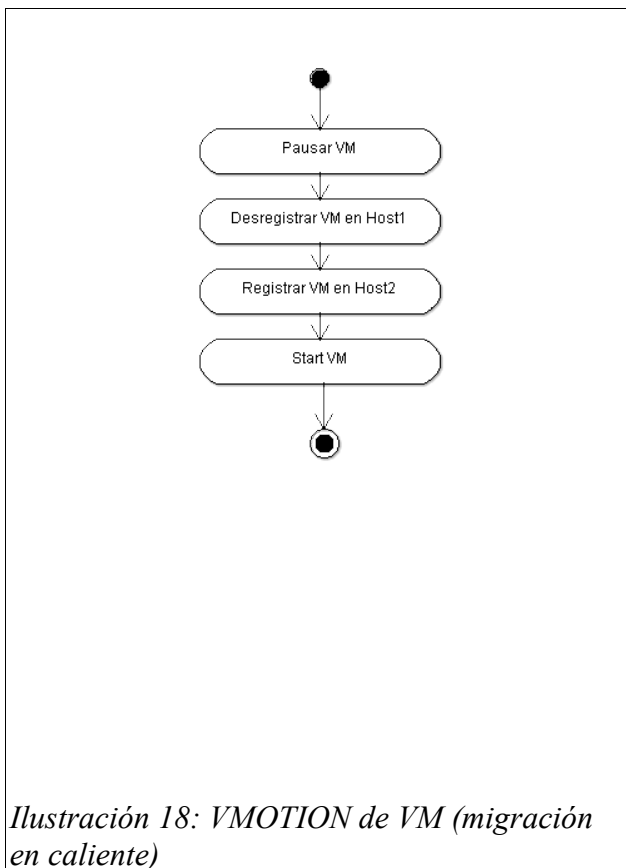
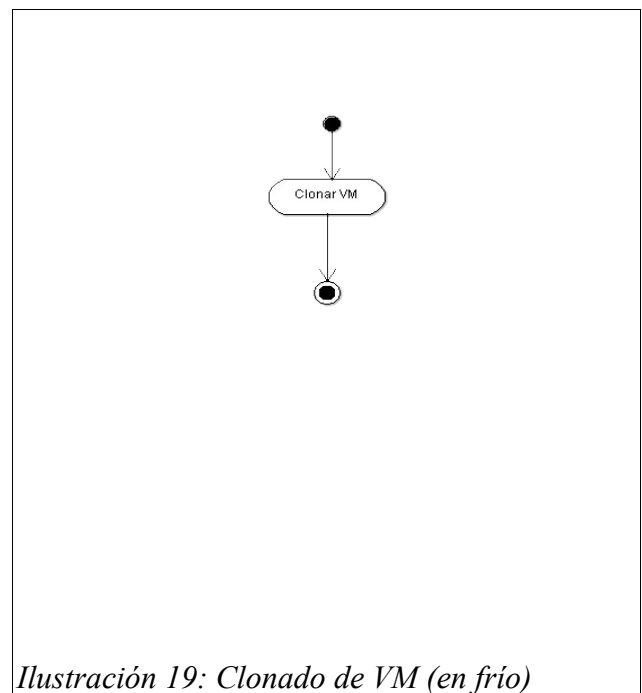
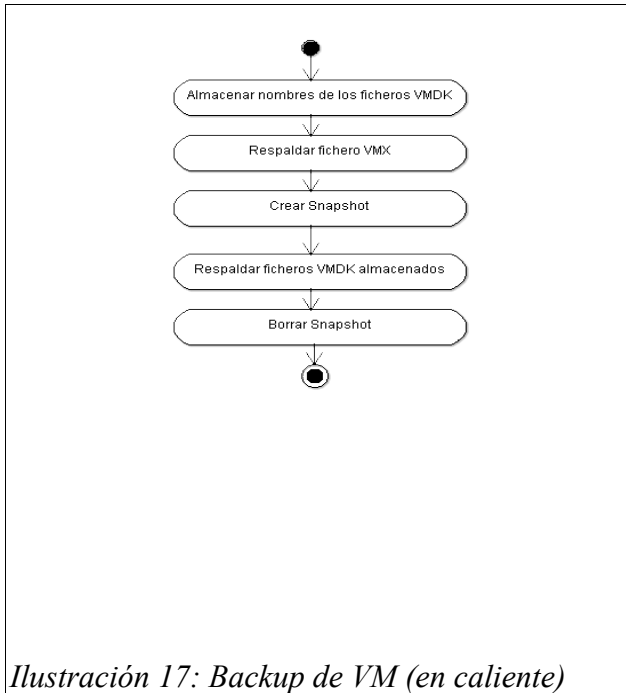
4.11 Diagrama de base de datos



La base de datos sigue un modelo muy sencillo y se basa en unas pocas tablas, creadas en un MySQL local, pero que al no usar ninguna funcionalidad particular de éste podrían ser implementadas por otros gestores de bases de datos.

Todas las tablas cuentan con una clave principal, de nomenclatura Idxxx, autoincremental.

4.12 Diagramas de actividad



5. DESARROLLO DE HERRAMIENTA DE BACKUP DEDUPLICADO (AD)

5.1 Introducción

Las incidencias que afectan a toda plataforma informática pueden dejar fuera de servicio un sistema durante muchas horas y exigir el trabajo de muchos técnicos para su reparación. Son muy costosas y tal como se ha discutido en el capítulo I, dedicado a la plataforma hardware, debemos diseñar nuestra infraestructura lo más robusta posible para disminuir la probabilidad de ocurrencia o el impacto si ésta sucede.

La pérdida de datos, no solo afecta a nuestra continuidad de servicio si no que puede producir pérdidas de información no recuperable nunca más. Si no tenemos adecuadamente respaldados nuestros datos antes o después tendremos graves pérdidas.

Los sistemas de backup han ido evolucionando a lo largo del tiempo para dar soluciones al respaldo de datos y han dado lugar a estrategias que de una forma u otra se siguen en todas las organizaciones de cierto tamaño.

Habitualmente, la estrategia de backup consiste en realizar unas copias completas de nuestro conjunto de datos, con cierta frecuencia, por ejemplo, una vez a la semana. Estas copias se llaman FULL (completas). Con frecuencia se suelen planificar en domingo por ser el día que suele estar más libre o con menos carga de trabajo el centro de datos.

En el día a día se sigue una estrategia de copia incremental, es decir, se copian solo los datos que han cambiado desde la última copia FULL.

Las copias semanales se guardan un tiempo, por ejemplo, un mes y las copias mensuales se guardan un año. El tiempo que se guarda una copia desde que se hizo hasta que se recicla su espacio para copias futuras se llama período de retención.

Así, ante una pérdida de datos, contaríamos con una copia diaria de los últimos 7 días, 1 copia semanal de las 3 últimas semanas, 11 copias mensuales y varias anuales.

La estrategia se completa con armarios ignífugos para cierta información y copias que se sacan fuera del edificio ya sea en soporte digital o por medio de fibra óptica a un centro de respaldo. Con el fin de asegurar la supervivencia de los datos en caso de desastre total.

El tiempo que se tarda en hacer la copia de todos nuestros datos, se denomina ventana de backup y suele ser el factor limitante al tamaño de los datos que queremos respaldar incluso por encima del espacio necesario de almacenamiento. Habitualmente, la ventana de backup es de 8 horas que coinciden con el horario nocturno.

5.2 Objetivo de proyecto de desarrollo de la herramienta de backup deduplicado

Los métodos tradicionales de backup presentan una muy baja eficiencia cuando en enfrentan al reto de salvaguardar máquinas virtuales por dos razones principalmente:

1.- Los datos de las máquinas virtuales se concentran en grandes ficheros del orden de gigabytes que representan discos duros enteros. Con modificar un solo bit de un gran fichero de este tipo las utilidades de backup tradicionales detectan fichero modificado y salvaguardan todo el fichero.

2.- Por ser datos binarios se prestan muy poco o nada a la compresión de datos, por lo que las utilidades y estrategias que piensan en utilizar la compresión para disminuir el tamaño de los datos almacenados no consiguen reducir el espacio ocupado y, como veremos más adelante, incremental el tiempo de backup de forma excesiva.

Existen en el mercado utilidades de backup incremental a nivel de bloque o como lo llaman otros backup deduplicado. Dichas utilidades muy frecuentemente asociadas a hardware especializado son muy costosas de adquirir y no siempre ofrecen el rendimiento adecuado.

Además, en general, son poco amigables y se integran regular o mal con scripts de backup que en grandes instalaciones superan en uso y prestaciones a herramientas en entorno puramente gráfico ya que en estas instalaciones el proceso de backup se debe integrar con comandos lanzados a las cabinas de almacenamiento, a scripts de aplicaciones concretas, etc.

El objetivo de este proyecto será desarrollar una utilidad capaz de salvaguardar de una forma sencilla todo tipo de datos pero especialmente datos de virtualización

5.3 Alcance del proyecto

El software desarrollado debe integrarse vía un API específico con cualquier otra aplicación (JAVA) y en concreto con el FVC desarrollado como proyecto “hermano”.

Además, se proveerá una aplicación en modo consola para poder utilizar la utilidad en modo comando desde cualquier script.

La utilidad automatizará también la gestión de los períodos de retención de las copias.

Se realizarán estudios y pruebas para comparar eficacia (en espacio y en tiempos) con otros herramientas de backup al uso.

Nombre de la utilidad a desarrollar: AD (...de Archivado Deduplicado)

5.4 Estudio previo de las herramientas de backup

5.4.1 Introducción

Con el fin de diseñar un sistema de backup adecuado a la infraestructura de virtualización que estamos contemplando se ha realizado un estudio de herramientas útiles, gratuitas y/o libres para la realización de los backups.

Para cada herramienta se han estudiado sus capacidades, su rendimiento en tiempo y en espacio ocupado.

Se ha montado un laboratorio consistente en un PC con:

- DISCO C: 160GB, ATA100, en NTFS.
- DISCO D: 2x DISCO SATA II de 750GB, montados en RAID0 y NTFS
- 4GB de RAM
- AMD PHENOM II 3GB, 4 cores.

Datos a respaldar:

106GB de máquinas virtuales, (13 máquinas, en promedio 8GB/máquina) alojadas en el disco D:

Backups al disco [c:\backups](#)

El PC se usa para desarrollo, no como servidor de sistemas de producción, por lo que en una misma jornada laboral no se utilizan, en promedio, más de 3 máquinas. Como veremos, este dato es importante de cara a los respaldos parciales: Incremental y diferencial diarios.

Con el fin de poder hacer cálculos teóricos sobre rendimientos máximos de los sistemas de backup que se estudiarán a continuación se ha testado dicha plataforma para obtener datos sobre el rendimiento de los discos:

	DISCO C	DISCO D (RAID 0)
Acceso aleatorio, bloques pequeños (64 kb)	1 MB/s	1.1 MB/s
Acceso aleatorio, bloques grandes (8 mb)	48 MB/s	83 MB/s
Acceso secuencial (bloques grandes)	62 MB/s	100 MB/s

Para hacer las pruebas, se ha empleado la herramienta HdTuneProfessional ya tratada en el capítulo I de este proyecto.

5.4.2 Herramientas del sistema operativo: XCOPY y ROBOCOPY

Los sistemas operativos Windows vienen con un utilidad muy potente para la copia de directorios llamada XCOPY, además, podemos descargar gratuitamente de Microsoft el Kit de recursos(rktools.exe) que añade ROBOCOPY muy similar a la anterior pero más potente.

De las decenas de parámetros con que cuentan el más importante, de cara a los backups, es el parámetro /M que permite a estas utilidades hacer uso del bit de archivado (A) con que cuentan los sistemas de archivos y que se activa cuando un archivo es nuevo o ha sido modificado. Las herramientas de backup, en general, respaldan los archivos que tienen este bit activado y lo desactivan, estado en el que permanece hasta que el usuario/sistema operativo modifica el archivo lo que provoca que el bit se vuelva a activar.

Backup FULL: (Planificado los domingos a las 9:00 de la mañana)

REM Activar el bit de archivado en todos los datos, aplicar recursivamente:

Attrib +A D:\ /S /D

REM Copiar al respaldo todos los archivos con el bit de archivado activado.

REM Además, desactiva el bit de archivado y no pide confirmación en sobrescrituras, etc.

ROBOCOPY D:\ C:\Backups /M /MIR /Y /Q

REM Vaciar directorio diario:

DEL DIARIO /S /Q

Backup Incremental: (Planificado todos los días, menos los domingos, a las 24:00)

```
ROBOCOPY D:\ C:\Backups\DIARIO /M /MIR / Y /Q
```

Podemos complementar ambos con un nombre de archivo/directorio que contenga la fecha del sistema: (planificado una vez al mes)

```
for /f "tokens=1,2,3,4 delims=/ " %%i in ('date /t') do set d=%%k%%j%%i
```

```
set destino=C:\Backups\%d%
```

```
ROBOCOPY D:\ C:\Backups\MENSUALES\%destino%
```

Con esta estrategia, contamos con una copia semanal (en [c:\backup](#)) un directorio diario (donde tenemos los cambios desde la última semana, llamada copia diferencial) y en el directorio mensuales (un directorio por mes: 20100101, 20100201, 20100301, etc...)

Podemos refinar el diseño con todo tipo de automatismos como el borrado de directorios mensuales cuando superan una antigüedad, etc...

Además de un mayor número de funcionalidades ROBOCOPY, con su parámetro /MIR y/ /PURGE nos provee una funcionalidad que XCOPY no tiene que consiste en que si del conjunto original de datos borramos un archivo ROBOCOPY lo borrará también de los datos de respaldo. De esta forma, los datos copiados se convierten en un espejo de los datos originales.

Combiando estas ordenes con otras de red podemos hacer copia a un servidor de backup de los servidores de archivos que queramos:

```
NET USE \\servidor\datos clave USER:LOCAL/administrador
```

```
ROBOCOPY \\servidor\datos c:\backups\servidor\
```

```
NET USE \\servidor\datos /delete
```

Una limitación que tienen estas utilidades consiste en que no pueden copiar ficheros abiertos en modo exclusivo por otro programa.

Resumen funcionalidades para backup de XCOPY y ROBOCOPY:

	Copia FULL	Copia dif./incr.	Purga ficheros borrados	Comprime	Encripta
XCOPY	Sí	Sí	No	No	No
ROBOCOPY	Sí	Sí	Sí	No	No

En caso de querer copiar archivos abiertos en modo exclusivo por otro programa podemos hacer uso de la utilidad VSHADOW, del propio Microsoft.

NT BACKUP, utilidad con entorno gráfico provista también con los sistemas Microsoft hace uso de la herramienta anterior. No se estudia aquí porque está muy limitada respecto a los comandos de ordenes anteriores y solo esta indicada para monopuesto.

Estudio teórico de rendimiento máximo y pruebas de laboratorio

Realizar un backup FULL con ROBOCOPY, en la plataforma de laboratorio descrita, debería ocupar el mismo espacio que los datos originales (106GB) y consumir unos 51 minutos:

$$106\text{GB} = 108.544 \text{ MB}$$

$$108544/83 \text{ (83 MB/s en LECTURA ALEATORIA bloques grandes, disco D)}=1307 \text{ s}$$

$$108544/62 \text{ (62 MB/s en ESCRITURA secuencial, disco C)}=1751 \text{ s}$$

$$\text{Total: } 1307\text{s}+1751\text{s}=51 \text{ minutos}$$

Realizar un Backup INCREMENTAL ocuparía el mismo espacio que los ficheros originales modificados, en promedio, 3 maquinas virtuales, $3 \times 8 = 24$ GB de datos (diarias)

En tiempo, se puede despreciar el tiempo que se tarda en seleccionar los ficheros con el bit de archivado activado ya que al realizarse vía directorios no tarda en comparación un tiempo apreciable. El tiempo de copia sería: $24 \times 1024/83 + 24 \times 1024/62 = 12$ minutos aproximadamente.

Resumen:

	Ocupacion FULL	Ocupacion DIARIA	Tiempo FULL	Tiempo DIARIA	RENDIMIENTO
ROBOCOPY	106 GB	24 GB	51 min	12 min	35'5 MB/s

(Las pruebas realizadas con backups diarios, durante varios meses, confirman la exactitud de estos cálculos)

Conclusiones

MUY RÁPIDA(*) en todos los casos: FULL e INCREMENTAL.

Espacio ocupado: Excesivo en las copias diarias!

Si estimamos que solo se modifican 3 VM al día = 24 GB

De esos 24 GB si pensamos que solo se habrá modificado un 1%, tenemos que para 0.24 GB modificados al día estamos guardando 24 GB!

(*): El método es muy rápido si escribimos los datos de backup a un medio rápido, por ejemplo un HD, si empleáramos un HD USB externo la copia ya no podría considerarse rápida ya que escribir 24 GB de datos en un medio lento, para salvaguardar solo 0.24 GB de cambios no es nada eficiente.

5.4.3 Herramienta de compresión de datos: 7z (software libre)

7Z es una de las herramientas de compresión más potentes que conozco si no la que más.

Es de gran utilidad en multitud de escenarios y es, a mi modo de ver, una excelente opción para respaldar ciertos tipos de datos.

Dicho ésto, contra todo pronóstico, en escenarios de respaldo de virtualización falla estrepitosamente y no es una herramienta de elección por las razones que aportaré a continuación.

Realizar una copia, comprimida, de un directorio de datos con 7z en línea de comandos es tan simple como:

```
7z a c:\archivo.7z d:\datos
```

si queremos una copia diferencial podemos escribir:

```
7za u c:\archivo.7z d:\datos -ms=off -mx=9 -t7z -u- -up0q3r2x2y2z0w2!c:\diferencial.7z
```

Resumen funcionalidades para backup de 7Z:

	Copia FULL	Copia dif./incr.	Purga ficheros borrados	Comprime	Encripta
7Z	Sí	Sí. Diferencial	Sí	Sí	Sí

Para el supuesto utilizado en las pruebas, 3 maquinas virtuales modificadas al dia, 24 GB, 7Z respalda 24 GB ya que funciona a nivel de fichero.

Durante un mes se ha realizado diariamente una copia diferencial y semanalmente una FULL.

Estos son los resultados:

	Ocupacion FULL	Ocupacion DIARIA	Tiempo FULL	Tiempo DIARIA	RENDIMIENTO
7Z	Beta * 106 GB	Beta * 24 GB	16 horas	4 horas	1'8 MB/s

Donde beta se ha estimado en 0.98. Es decir, practicamente, no se consigue ninguna ganancia de espacio con la compresión de datos.

Conclusiones:

EXTREMADAMENTE LENTA en todos los casos: FULL e INCREMENTAL.

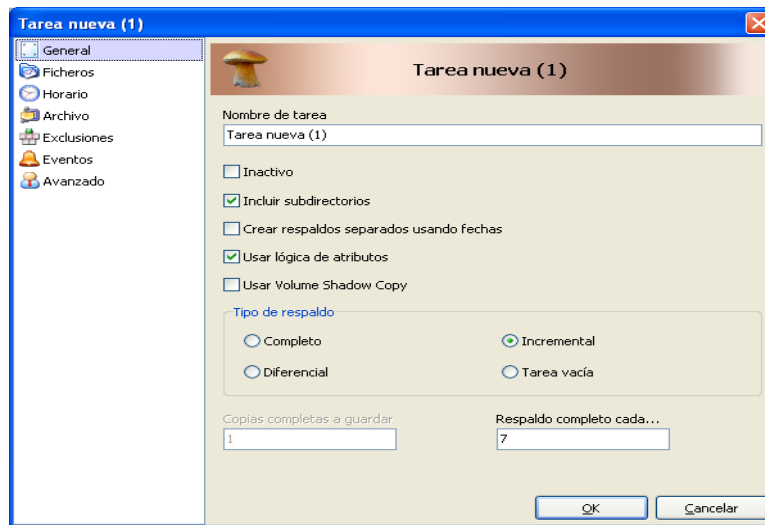
Espacio ocupado: Excesivo en las copias diarias!

Si estimamos que solo se modifican 3 VM al día = 24 GB

De esos 24 GB si pensamos que solo se abrá modificado un ¿1%?, tenemos que para 0.24 GB modificados al día estamos guardando 23.5 GB!

Herramienta de backup: COBIAN(software libre)

COBIAN, es una herramienta con interface gráfico de una calidad y robustez excelentes.



Permite todo tipo de configuraciones, lanzamiento de scripts, y formatos de backups.

Permite hacer copias con y sin compresión, y planificar las mismas desde la propia herramienta que se puede instalar como un servicio de windows.

Por supuesto, cubre también la posibilidad de copias incrementales, diferenciales, shadow copy para ficheros abiertos en exclusivo, etc...

Las pruebas hechas a la herramienta han resultado muy satisfactorias: Es fácil de usar, el formato utilizado para las copias es una directorio normal de windows en el caso de no comprimidas y un archivo 7Z en el caso de las comprimidas, se puede parametrizar con todo tipo de configuraciones, e tc..

Resumen funcionalidades para backup de COBIAN:

	Copia FULL	Copia dif./incr.	Purga ficheros borrados	Comprime	Encripta
COBIAN	Sí	Sí.	Limitada a modo espejo	Sí	Sí

Las pruebas realizadas se han llevado a cabo sobre COBIAN BACKUP BOLETUS 10.1.

Se han realizado test en modo comprimido y no comprimido y los resultados son muy parecidos a ROBOCOPY y 7Z respectivamente:

	Ocupacion FULL	Ocupacion DIARIA	Tiempo FULL	Tiempo DIARIA	RENDIMIENTO
COBIAN com. 7z	Beta * 106 GB	Beta * 24 GB	16 horas	4 horas	1'8 MB/s
COBIAN Sin compresión	106 GB	24 GB	46 min	10.5 min	35'5 MB/s

Donde beta se ha estimado en 0.98. Es decir, prácticamente, no se consigue ninguna ganancia de espacio con la compresión de datos.

Conclusiones:

MUY RÁPIDO en modo no comprimido, pero muy costoso en espacio.

MUY LENTO, en modo comprimido, y poca mejora de espacio.

Si estimamos que solo se modifican 3 VM al día = 24 GB

De esos 24 GB si pensamos que solo se habrá modificado un ¿1%?, tenemos que para 0.24 GB modificados al día estamos guardando 24GB - 23.5 GB!

5.4.4 Conclusión global

Sería de desear una herramienta que fuera capaz de realizar las copias a la velocidad de una copia no comprimida (similar a ROBOCOPY) y que solo procesara los bloques de datos modificados y no todo el archivo, con lo que la ganancia de espacio en escenarios de respaldo de virtualización sería espectacular.

5.4.5 Uso eficiente del ancho de banda de red en backups remotos

Las pruebas de laboratorio previas han demostrado que una red local rápida de al menos 1Gbps, combinada con unos discos del servidor/cabina de datos configurados en RAID 10 permiten acceder a datos en remoto a una velocidad al menos tan buena como al disco duro en local.

Para movimientos masivos de datos, en una red local, el cuello de botella de los discos origen y destino es un factor limitante más importante que el ancho de banda de red (siempre presumiendo una red de al menos 1Gbps).

No obstante, se han desarrollado algunas herramientas que deberían hacer más eficiente el uso de red para transmitir grandes ficheros. Una de estas herramientas es rsync, disponible tanto en sistemas unix/linux como en windows.

RSYNC está diseñada de tal forma que a la hora de transmitir un archivo entre dos máquinas calcula la firma de los bloques que lo forman y solo transmite los bloques que han cambiado entre una y otra siendo así una forma muy eficiente de tener sincronizados dos archivos o directorios entre dos máquinas.

Siguiendo la filosofía general de todo el proyecto se ha querido probar en la práctica el rendimiento de esta herramienta con el fin de contar con datos objetivos de su utilidad, para ello se han diseñado las pruebas del apartado siguiente.

En los apartados siguientes se volverá a hacer mención de esta herramienta y se propondrán escenarios de uso.

Pruebas realizadas con Rsync (en Oracle Solaris 11 Express):

Entorno de pruebas: Rsync viene ya incluido en Oracle Solaris 11 Express, no así en versiones anteriores por lo que de usarse en Solaris 10 habría que compilar los fuentes de rsync con gcc y algunas librerías auxiliares.

Se han montado dos máquinas virtuales Solaris 11, con 4GB de espacio libre en disco, 4 GB de ram , 2 CPU y sistema de ficheros ZFS, versión 31. Corriendo en VMWARE ESX 4.0, sobre servidores físicos diferentes conectados a través de un enlace de red de 1Gbps.

Se han diseñado unas pruebas con el objetivo de comparar rendimiento entre un copiado normal de ficheros y un rsync entre las dos máquinas, para un fichero de 700MB.

PR1: Copiado por primera vez del fichero

PR2: Cambiar n bloques en posiciones aleatorias de éste y posterior sincronización con destino

	Copiado	RSync
PR1	30 segundos. 10% CPU	30 segundos. Enviados 700MB. 15% CPU
PR2	30 segundos. 10% CPU	20 segundos. Enviados 200 KB. 40 % CPU

Conclusiones:

La herramienta ha demostrado utilidad para la transferencia de cambios en ficheros y viendo la mayor eficacia que consigue en uso de banda de red está claro que en redes más lentas y usando arreglos de disco más rápidos donde el cuello de botella sean las comunicaciones esta herramienta puede representar una considerable mejora respecto a otras soluciones.

Por ello, para el caso de unir CPD principal con Centro de respaldo o sedes remotas con oficinas centrales esta herramienta sería de elección para la sincronización de ficheros y directorios.

Por contra, para la comunicación entre servidores físicamente cercanos y conectados por una red rápida y disco relativamente lento las mejoras se hacen menos evidentes.

5.4.6 Copias a nivel de bloques de datos: Snapshots

La tecnología de snapshots, instantáneas de disco, es soportada por la gran mayoría de cabinas de disco profesionales, por Openfiler e, incluso, por Solaris 10 x64, para el sistema de archivos VFS.

Consisten éstos en sacar instantáneas del estado del disco de forma que podamos eventualmente echar para atrás cambios o recuperar totalmente el estado de un servidor en caso de incidencia.

Su utilidad como Sistema de Recuperación ante Desastres, es decir, ante pérdidas totales de un servidor o de un sistema de archivos está fuera de toda duda y si las combinamos con la replicación en otra cabina o sistema de archivos en remoto conseguimos un sistema muy robusto ante ese tipo de incidencias.

No obstante, para el respaldo de datos de usuario y para el caso concreto que nos ocupa: respaldo de máquinas virtuales presentan algunas desventajas que paso a discutir:

- No se prestan bien a la recuperación de una sola parte de un snapshot, por ejemplo, recuperar solo una máquina virtual al estado que tenía antes de ayer. Generalmente, exigen la recuperación total de la instantánea o, en todo caso, la recuperación de toda la instantánea en un lugar distinto al de los datos originales y posterior copiado a mano sobre los archivos a recuperar. Pensemos en el escenario de que un usuario nos pide recuperar solo un archivo de ayer, no parece lógico, ni eficaz, recuperar 1 TB de datos para sólo acceder finalmente a un solo archivo. Lo mismo pasa con ficheros vmdk, podemos recuperar solo un servidor virtual y no los 50 que podamos tener en el sistema de almacenamiento.
- Tal como se comentará más adelante, para poder hacer backups de maquinas virtuales VMWARE en caliente es necesario seguir unos pasos concretos que pasan por lanzar un snapshot a nivel de VMWARE sobre la máquina. De esta forma, se garantiza que los archivos VMDK se quedan en un estado consistente mientras la maquina en ejecución escribe las modificaciones en otros ficheros. Cuando acaba el proceso de copia se consolidan los datos del snapshot y se borra éste.

El estado de snapshot de VMWARE debe utilizarse para este tipo de tareas pero se debe considerar temporal y se debe volver al modo normal lo antes posible ya que por un lado el rendimiento de la máquina cuando utiliza snapshot decrece y por otro corremos el riesgo de llenar el sistema de disco produciendo un desastre total en los servidores que afecte. Pensemos en un sistema que utiliza una cabina de disco y reserva una LUN de 500 GB para maquinas virtuales. En dicha LUN podemos tener 45 maquinas de 10 GB, en total 450 GB

de espacio ocupado de forma permanente. Si ponemos a trabajar dichas máquinas en modo snapshot el espacio ocupado empezará a crecer ya que los cambios no sobrescriben los bloques originales. De esta forma, podemos quedarnos sin espacio en disco, máxime si coincide el uso de los snapshot con procesos como la defragmentación de las máquinas virtuales proceso que reescribe gran parte del contenido de sus discos VMDK.

El sistema ideal, por tanto, consiste en ir máquina a máquina haciendo un snapshot, realizando la copia de sus ficheros y consolidando el snapshot antes de pasar a la siguiente máquina. Este proceso paulatino encaja mal con la posibilidad nativa de snapshot de los sistemas contemplados aquí.

- Generalmente, los snapshots solo son accesibles en la máquina que los realizó y como mucho en otra similar que utilice algún procedimiento de replicación. Son difícilmente, transportables entre máquinas y entre sistemas de almacenamiento externo.

Además, cuando hacemos un snapshot no copiamos realmente todos los datos, solo se copian los modificados en lugar distinto al original esto es muy eficaz en una sola máquina donde tenemos accesibles los bloques antiguos y los nuevos pero si queremos sacar la instantánea a un almacenamiento externo deberemos copiar el 100% de los datos a respaldar lo que resta eficacia al sistema.

- Causan en mayor o menor medida cierta degradación del funcionamiento normal de las máquinas virtuales y archivos de datos de usuario en producción por lo que todos los sistemas imponen ciertas limitaciones en lo que a número y periodo de retención atañe. Excepción hecha del ZFS de Solaris que en esta apartado no impone límites en la práctica (2^{64} snapshots).

5.4.7 Deduplicación a nivel del sistema de archivos

Otra tecnología cada día más presente en dispositivos y programas es la llamada deduplicación de datos. De las soluciones contempladas aquí solo soporta este sistema a nivel de sistemas de archivos Solaris cuando utiliza ZFS, a partir de la versión 20 de ZFS. En las propuestas que se van a presentar en el punto siguiente se hace uso de esta tecnología.

Se han diseñado unas pruebas para comprobar el funcionamiento de la deduplicación de ZFS:

Entorno de pruebas similar al usado para Rsync, activando Dedup.

PD1: Deduplicación de bloques intrafichero. Archivo de 700 MB, con contenido trivial:relleno a ceros. Se trata de investigar si el sistema es capaz de detectar similitudes entre los bloques, ahorrar espacio y todo ello de una forma eficiente.

PD2: Deduplicación de bloques intrasistema de ficheros. Bloques pertenecientes a diferentes archivos en diferentes directorios. Para ello se ha subido un archivo de 700MB con contenido aleatorio, un ISO, a diferentes carpetas y con distinto nombre.

PD2.1, PD2.2 y PD2.3 Primera/segunda/tercera subida del archivo a la Carpeta A, B y C.

PD2.4 Subida del archivo concatenación de los 3 primeros.

PD3: Por último, se ha concatenado al archivo original un archivo de 100 bytes con la finalidad de desplazar el contenido de los bloques del fichero y verificar si Solaris/ZFS era capaz de seguir aprovechando las similitudes entre bloques.

	Espacio total ocupado	Tiempo
PD1	380 KB	22 segundos
PD2.1	700 MB	52 segundos
PD2.2	701 MB	32 segundos
PD2.3	702 MB	32 segundos
PD2.4	2'5 GB	97 segundos
PD3	2'7 GB	35 segundos

Conclusiones:

La deduplicación de ZFS/Solaris 11 es adecuada en uso de espacio en disco cuando existen bloques similares del mismo o de otros ficheros perfectamente alineados.

En cuanto a la eficiencia, como puede verse, no penaliza excesivamente los tiempos.

5.4.8 Propuesta arquitectura para optimizar backup

Vistas así las cosas, nos damos cuenta de que es necesario una herramienta o estrategia de backup que permita almacenar instantáneas del estado de nuestras máquinas, que dichas instantáneas contengan archivos en estado consistente a nivel de VMWARE, que se puedan almacenar indefinidamente por un largo período de tiempo compatible con nuestra política de retención de backups, por ejemplo:

- Copias anuales con retención 5 años.
- Copias mensuales con retención un año.
- Copias diarias con retención un mes.

Que dichas copias puedan almacenarse en un servidor distinto al que contenga las originales y eventualmente sacarse a cinta/dispositivo extraíble o enviarse telemáticamente a un centro remoto.

La herramienta de backup debe poder permitir recuperar parcialmente un contenido concreto para una fecha arbitraria y, por último, debe hacer todas estas tareas de una forma que afecte lo menos posible al entorno de producción.

En los párrafos siguientes discuto la utilidad de cada uno de estos sistemas: backup a nivel de bloque, deduplicación a nivel de sistemas de archivos, rsync, etc. para varios escenarios dados.

Escenario 1: Una LUN con todas las VM, servidas por Solaris
con iSCSI y SNAPSHOT de ZFS

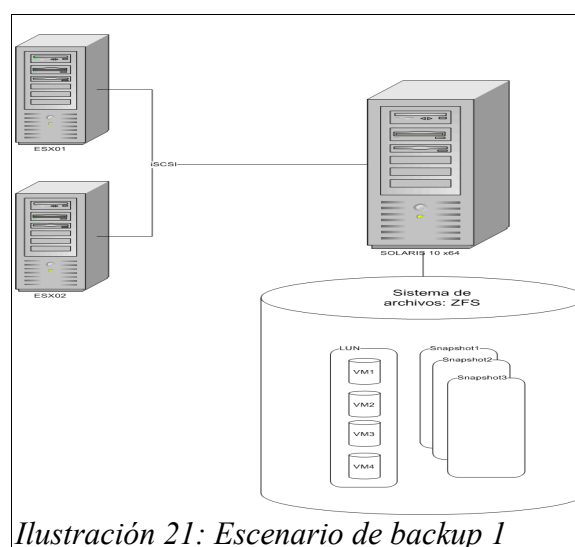


Ilustración 21: Escenario de backup 1

En el **escenario 1**, se plantea un servidor de datos basado en Solaris 10 x64, con sistema de archivos ZFS que publica por iSCSI una única LUN donde se almacenan todas las máquinas virtuales. Cada x tiempo se hace un snapshot de dicha LUN para lo cual hace falta poner todas las máquinas virtuales que contiene en estado snapshot del vmware previamente.

Problemas:

La LUN que contiene las VM es vista por Solaris como un único archivo. Además, dicha LUN está formateada por los ESX con el sistema de archivos VMFS sistema propio de VMWARE y que Solaris no entiende. Por tanto, a la hora de recuperar no habría forma de extraer solo una maquina virtual de la instantánea de forma sencilla. En instalaciones moderadamente grandes, este requisito es obligatorio y de uso frecuente por lo que por esta razón se descarta esta arquitectura.

Una variación de este sistema sería crear una LUN para cada máquina virtual, de esta forma sí podríamos hacer recuperaciones de un solo servidor virtual.

Problema:

La gestión de una LUN por máquina virtual es inasumible: Pensemos que cada vez que creáramos una máquina virtual nueva tendríamos que crear una LUN expofeso, conectarla con cada uno de los ESX de nuestro pool, uno por uno, adaptar los scripts necesarios de backup y otras operaciones, configurar sus copias snapshot, etc... Esta arquitectura, por tanto, también es descartada.

Escenario 2: VMs servidas por Solaris por NFS y SNAPSHOT de ZFS

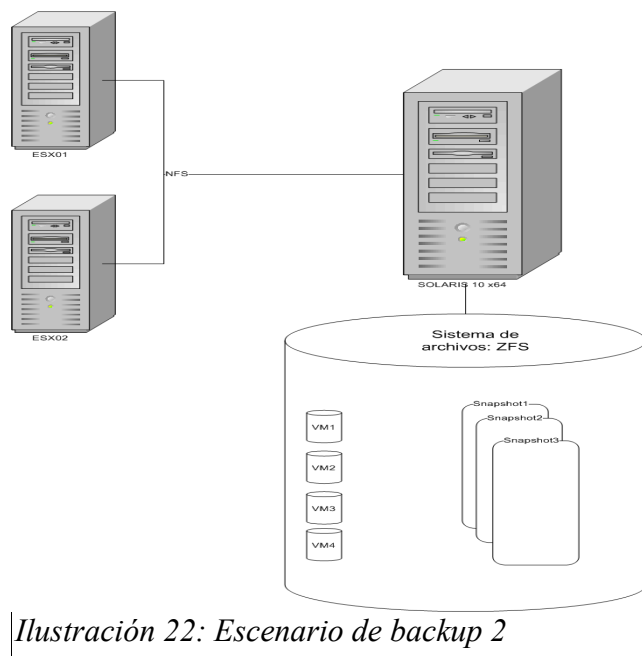


Ilustración 22: Escenario de backup 2

En el **escenario 2** las máquinas virtuales son servidas por un Solaris 10 x64, a través de NFS.

Ventajas:

Los ficheros de VM residen ahora nativamente en el sistema de archivos de Solaris, VFS, y si realizamos backup de ellas con snapshot de ZFS luego seremos capaces de recuperarlas individualmente. Si combinamos lo anterior con una segunda máquina Solaris que replique por medio de rsync el sistema de disco de la primera podemos contar con una solución aceptable.

Problemas:

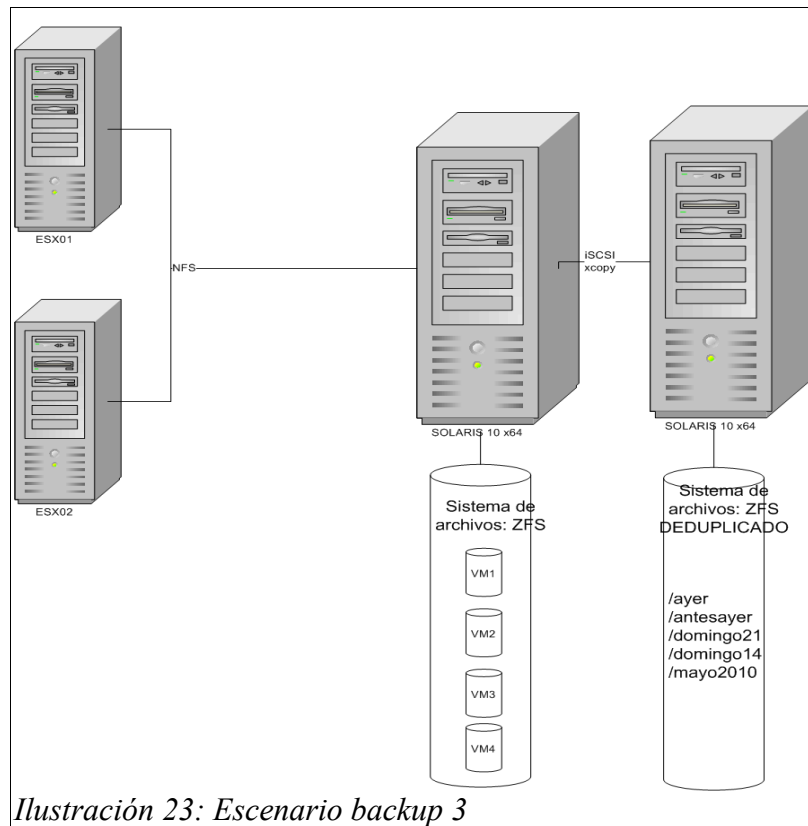
Dificultad para hacer backup de cada máquina virtual de una en una, procedimiento deseable para VMWARE. Aquí, pondríamos todas en modo snapshot, haríamos la foto de disco, etc...

La recuperación es muy engorrosa por la dificultad de montar una snapshot en un lugar diferente al original y recuperar solo una parte de su contenido. Además, sería necesario arreglar “a mano” los archivos de configuración de las maquinas virtuales recuperadas para que hagan referencia a los vmdk correctos.

Es complicado, pero posible, sacar fuera del sistema las copias: a cinta, armario ignifugo, etc.

Conclusión: Esta arquitectura es aceptable y por lo tanto la voy a llamar **Backup - Opción I**

Escenario 3: VMs servidas por Solaris
por NFS y copia por iSCSI a destino deduplicado



El **escenario 3** plantea un procedimiento sencillo que cumple casi todos los requisitos impuestos a la arquitectura buscada: Permite lanzar copias de ficheros individuales de forma negociada con VMWARE para que estén uno a uno en modo snapshot, utiliza para comunicación iSCSI mucho más potente que NFS, permite movilidad de las copias entre soportes (perdiendo la deduplicación) y el tamaño de la copia destino no es excesivo confiando la deduplicación al sistema de archivos ZFS.

La recuperación de máquinas individuales es sencilla y directa.

A este escenario lo voy a llamar **Backup - Opción II**

Escenario 4: VMs servidas por Solaris
por NFS y copia por iSCSI a destino AD

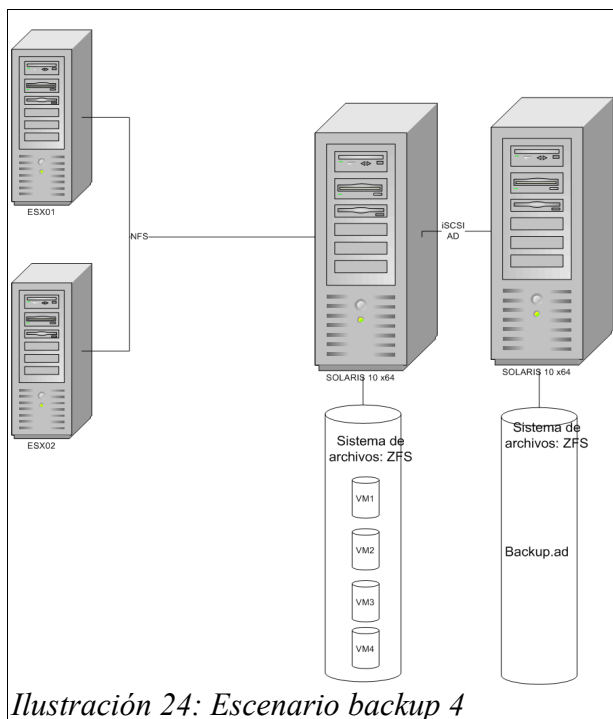


Ilustración 24: Escenario backup 4

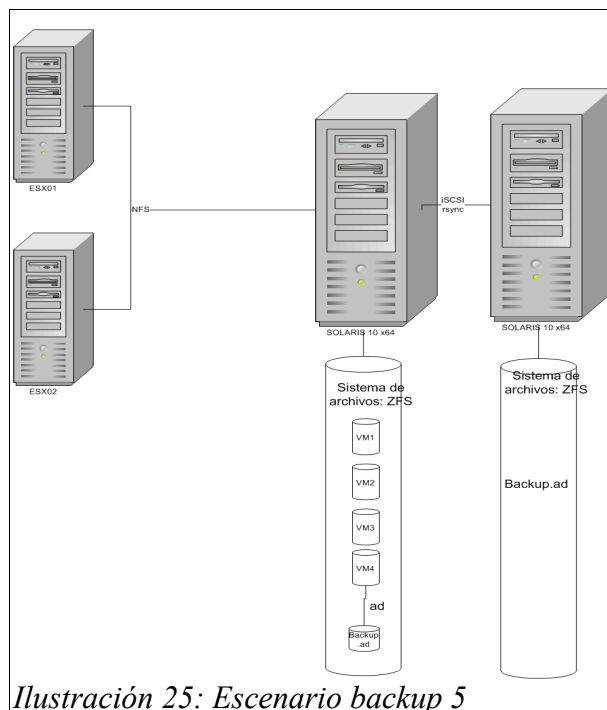


Ilustración 25: Escenario backup 5

El **escenario 4** sustituye el copiado de los ficheros por la herramienta de deduplicado creada como parte de este proyecto y perfectamente integrada con el procedimiento de backup progresivo de las máquinas virtuales. El acceso a la máquina de backup se realiza por iSCSI que es mucho más óptimo para esta tarea que NFS, el fichero deduplicado puede moverse a cualquier otro soporte: cinta, usb, etc..el espacio ocupado es mínimo y el rendimiento tal como han demostrado las pruebas es similar al copiado directo.

La recuperación se realiza también de forma directa y perfectamente controlada.

El **escenario 5** supone una variación menor del 4, consistente en realizar la copia deduplicada a un archivo local a la máquina 1 y después sincronizar dicho archivo con otro similar residente en la máquina de backup.

Para que este escenario suponga alguna ventaja en rendimiento el disco/raid donde resida el fichero backup.ad debe ser distinto al que contenga los datos originales. De esta forma se puede disminuir un poco el tiempo de backup para cada máquina virtual y después realizar la sincronización con la

de backup sin tantas restricciones de tiempo.

No obstante, este escenario supone un mayor coste económico y de energía por la necesidad de adquirir más discos duros y no aporta grandes ventajas en rendimiento cuando hablamos de conexiones de red a 1Gbps con posibilidad de hacer bonding entre varios enlaces económicos a un 1Gbps, incluso dedicar algunas líneas ethernet solo para hacer backup entre ambas máquinas.

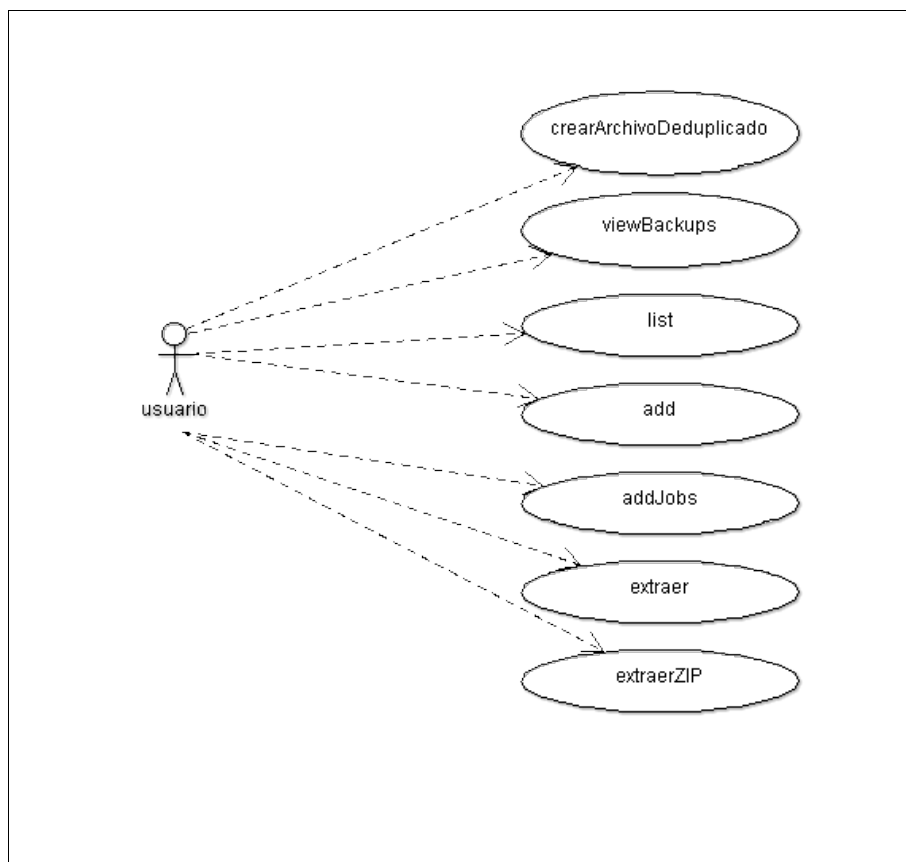
Por los motivos anteriores, voy a llamar al escenario 4 **Backup – Opción 3** y al escenario 5 lo voy a descartar porque no aporta grandes ventajas y sin embargo si aporta complejidad.

Opción de backup:	Conclusiones:
Opción I VMs servidas por Solaris por NFS y SNAPSHOT de ZFS	Dificultades para recuperaciones parciales Mala coordinación con procedimiento de vmware Difícilmente transportable
Opción II VMs servidas por Solaris por NFS y copia por iSCSI a destino deduplicado	No transportable fuera del sistema sin perder la deduplicación de disco. (Ej. Copias externas a eSATA) Dependiente de un sistema operativo (Solaris) y, de un sistema de ficheros concreto(ZFS)
Opción III VMs servidas por Solaris por NFS y copia por iSCSI a destino AD	Cumple al 100% los requisitos buscados. ES LA OPCIÓN ELEGIDA

5.5 DESARROLLO DE LA HERRAMIENTA

5.5.1 Casos de uso

Diagrama de casos de uso de la herramienta AD (Archivado Deduplicado)



AD path\archivo CREATE [periodoRetencion]

Crearé el archivo “archivo”, y configurará el periodo de retención mínimo al establecido en el parámetro. De no especificarse se tomará el periodo de retención igual a 15 días.

El sistema llevará en todo momento control de cuando empezó a usarse un archivo para hacer backup y hasta cuando hay que conservarlo como mínimo.

Cuando hacemos un backup en un “archivo.ad” el sistema procesa el periodo de retención y si el archivo se considera caducado automáticamente se crea el archivo “archivo.bak.ad” y se inicia un nuevo archivo “archivo.ad”. De esta forma se garantiza que siempre tengamos disponible como mínimo el número de días establecido por periodo retención.

AD path\archivo [VIEWBACKUPS]

Si utilizamos el comando VIEWBACKUPS o simplemente invocamos la utilidad sin ningún comando se nos presentará en pantalla un listado de todos los backups que contiene el archivo.

AD path/archivo [backupcode] LIST [path][filtro] [-R]

El comando LIST visualizará todos los archivos y directorios que contenga un backup dado. En caso de no especificar ninguno se listará el último backup que contenga el archivo.

Además, puede filtrarse la salida, para visualizar solo directorios o ficheros concretos que cumplan ciertas condiciones.

Por último, puede desactivarse la recursividad con la cláusula -R.

Máscaras de filtro permitidas:

*.extension ejemplo: *.pdf

nombre.* ejemplo: proyecto.*

Importante: los directorios siempre deben especificarse acabados en una barra “\”.

AD path/archivo ADD [etiqueta] path[filtro] [-R]

Añadirá un backup que contendrá una carpeta etiquetada como “etiqueta”, y que respaldará archivos y directorio que se encuentren en el “path” proporcionado y que cumplan el filtro establecido.

En caso de no establecer la etiqueta los archivos colgarán de la raíz del archivo ad.

AD path/archivo ADDJOBS archivo.jobs

Permite editar un archivo de texto en un formato adecuado y hacer de golpe backups de todos los trabajos definidos en dicho fichero. Importante: En caso de que alguna ruta contenga espacios debera entrecomillarse.

El formato a utilizar en “archivo.jobs” será: (un trabajo por fila)

etiqueta path[filtro] [-R]

etiqueta path[filtro] [-R]

etiqueta path[filtro] [-R]

Ejemplo: trabajos.jobs

VM c:\maquinas\

PERSONAL c:\datos*.doc

PROY c:\proyecto\proyecto.* -R

Importante: En caso de que alguna ruta contenga espacios debera entrecomillarse.

AD path/archivo [backupcode] EXTRAER [etiqueta][filtro] pathdestino [-R]

Extraerá todos los directorios y archivos que se encuentren colgando de etiqueta y que cumplan el filtro provisto y creará dicha estructura en el pathdestino.

Importante: En caso de que alguna ruta contenga espacios deberá entrecomillarse.

AD path/archivo [backupcode] EXTRAERZIP [etiqueta][filtro] archivo.zip [-R]

Extraerá todos los directorios y archivos que se encuentren colgando de etiqueta y que cumplan el filtro provisto y creará un archivo.zip con dicha estructura.

Importante: En caso de que alguna ruta contenga espacios deberá entrecomillarse.

5.5.2 Decisiones de diseño

El lenguaje de implementación elegido será JAVA buscando con ello la máxima compatibilidad y portabilidad del software.

La aplicación funcionará en modo CONSOLA y dispondrá también de un API para ser integrada en aplicaciones.

El código resultante de dicho desarrollo será publicado con licencia GPL para darle la mayor difusión posible, permitir que otros disfruten de lo aquí conseguido tal como yo he hecho uso de software de terceros en muchas ocasiones y facilitar que otras personas interesadas en continuar con dicho trabajo puedan hacerlo sin cortapisas ningunas.

Sería estupendo que algún futuro proyecto de fin de carrera de la Universidad pudiera continuar este trabajo. Apunto aquí mismo dos líneas de trabajo interesantes: ampliar el software para permitir ficheros AD encriptados y desarrollar un agente para copias en remoto eficientes en ancho de banda de red.

5.5.3 Gestión de la configuración

Como entorno de desarrollo se usará NetBeans en su versión 6.7.1 y como repositorio de código se usará Subversion. Todos los días se hará backup de la máquina de desarrollo que estará montada como máquina virtual Windows 2003, y correrá sobre VMWARE WORKSTATION y/o VMWARE ESXi 4.0.

Los diagramas de análisis y diseño de la aplicación serán realizados con las herramientas ArgoUML y Visio 2003.

Para la redacción de la documentación se usará openoffice, versión 3.2

Los gráficos serán tratados con GIMP 2.2.

5.5.4 Diagrama de clases

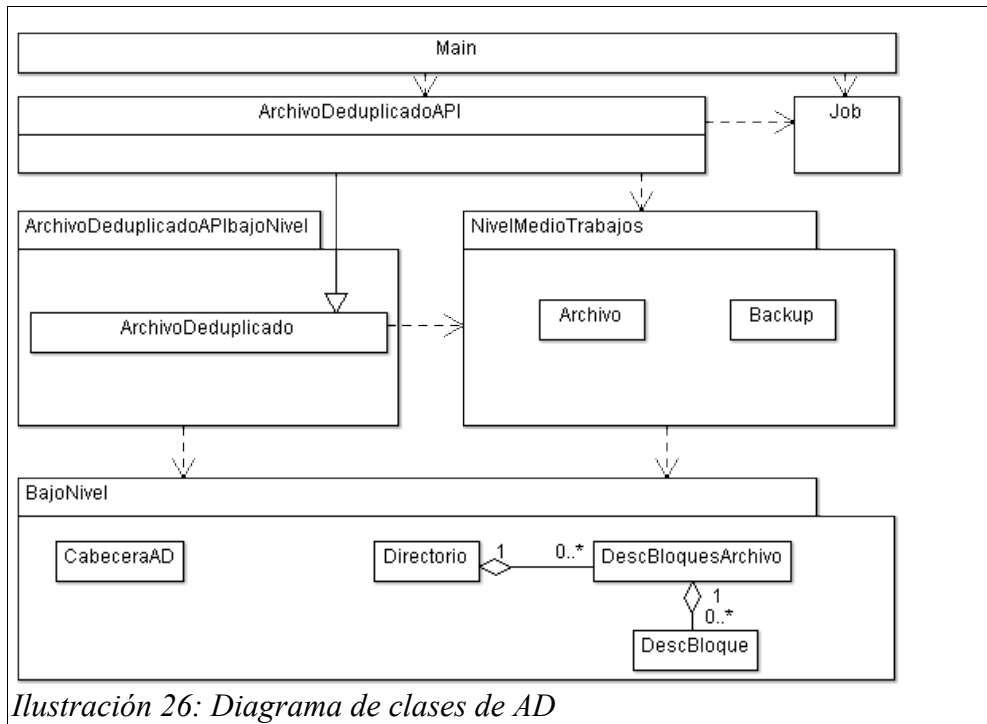


Ilustración 26: Diagrama de clases de AD

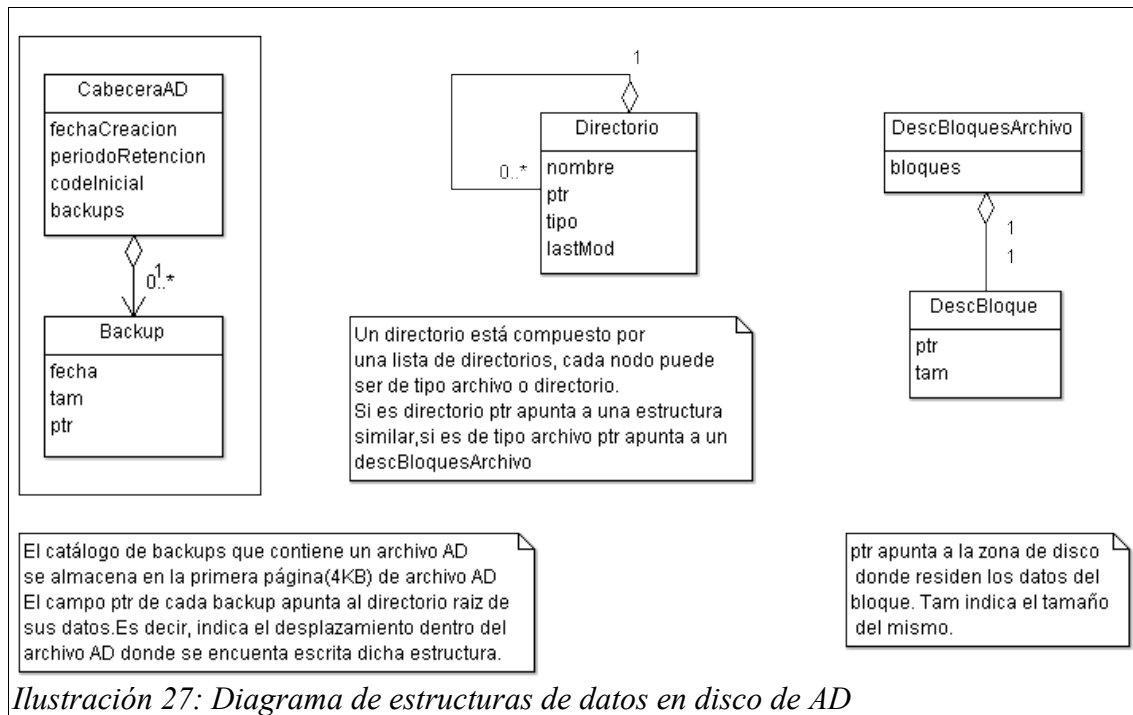
Se ha hecho uso de un diseño organizado por capas de mayor a menor abstracción y de mayor a menor distancia lógica a los datos brutos grabados en el disco.

La clase **Main**, que implementa la funcionalidad de consola, se dedica a decodificar los parámetros recibidos como parámetros y realiza las operaciones llamando a funciones de la clase **ArchivoDeduplicadoAPI** utilizando la clase **JOB** como estructura de datos que encapsula trabajos de backup.

La clase **ArchivoDeduplicadoAPI** expone toda la funcionalidad por medio de funciones de alto nivel que pueden ser llamadas tanto desde la consola como desde un hipotético software que pudiera hacer uso de esta clase con llamadas directas. Extiende la clase **ArchivoDeduplicado** con la finalidad de especializar las interfaces de llamada aligerando el número de parámetros necesarios para cada operación.

Por último, **ArchivoDeduplicado** se apoya en las clases de más bajo nivel que son las que implementan las lecturas y escrituras de disco de los datos residentes en estructuras de datos en memoria.

5.5.5 Estructuras de datos internas al archivo AD



5.5.6 Detalle del algoritmo de deduplicación

El algoritmo de deduplicado diseñado se basa en el principio de localidad, esto es, se presupone que las modificaciones de datos dentro un archivo suceden generalmente en zonas agrupadas dentro de éste, por ejemplo, en un archivo vmdk que guarda datos de una maquina virtual cuando dentro de ésta se modifica un fichero eso se plasma en datos modificados en una o varias zonas dentro del archivo vmdk. Por ello, cuando se detecta un dato modificado se marca toda una página de tamaño 4KB como modificada, de esta forma, se evita una fragmentación excesiva de datos, tal como se produciría si permitimos bloques de tamaño arbitrariamente pequeños.

A la hora de buscar bloques de datos similares a los nuevos que queremos incorporar nos ceñimos a los pertenecientes a la última copia del mismo archivo. De esta forma, gastamos pocos ciclos de CPU y trabajo de disco en buscar bloques en ficheros y en posiciones que difícilmente puedan coincidir con los buscados. Pensemos que la herramienta está pensada principalmente para grandes archivos como VMDK y es en estos escenarios donde difícilmente encontraremos bloques repetidos en ficheros distintos a la última copia del actual ya que aún conteniendo datos similares, por ejemplo, dll de windows almacenados en discos virtualizados, es casi imposible que residan en la

misma zona del archivo, ya que la fragmentación interna a éste mueve continuamente los datos de sitio, efecto que se agrava si los ficheros de maquinas virtuales, imágenes, voz, etc...han sido encriptados. Se ha preferido sacrificar una posible mayor deduplicación a costa de ganar en tiempo de proceso y carga de la CPU y el disco.

Se sopesó la posibilidad de utilizar para las comparaciones mecanismos de HASH almacenando para cada bloque en el sistema su firma para compararla con los nuevos a incorporar. No obstante, se desechó la idea porque en los escenarios previstos si bien se modifica una cantidad relativamente pequeña de los datos de un archivo estas modificaciones están repartidas por todo éste por lo que para poder utilizar mecanismos de hash eficientemente habría que calcular y almacenar las firmas de cada bloque de digamos 4KB para compararla con cada bloque del mismo tamaño a actualizar en el sistema. El cálculo de estas firmas supone un gran esfuerzo computacional y el almacenamiento de cada una de ellas para pequeños bloques con el fin de evitarse su cálculo en el futuro representa también una gran carga de trabajo. Por último, un gran problema que presenta la comparación por hash consiste en que si bien la probabilidad de que dos bloques que tengan la misma firma sean realmente distintos es muy pequeña en la realidad no es nula por lo que no se puede asegurar que la copia de seguridad que tenemos sea fideligna al 100%. En entornos de bajo ancho de banda de red, por ejemplo, copias por internet, puede ser interesante el uso de firmas por bloque. Pero en el escenario aquí contemplado, se ha considerado más conveniente, la comparación directa de bloques.

El proceso que sigue el algoritmo es el siguiente:

El deduplicador va recorriendo todas los directorios y archivos que le indica el job correspondiente.

Para cada archivo que encuentra comprueba si existe en sus estructuras de datos, si no existe, se incorpora al archivo deduplicado como un único bloque de tamaño igual al tamaño original del archivo.

Si el archivo existe, se comprueba la fecha de última modificación, si es igual a la almacenada en la última copia se apunta el puntero del archivo a los descriptores de bloques ya presentes en el disco apuntando por tanto a los datos ya almacenados por lo que no consume espacio adicional.

Por último, si el archivo existe y ha sido modificado con posterioridad a la última copia se recorre cada uno de los bloques de datos que formaban el archivo copiado con anterioridad. Si se encuentra alguna discrepancia en los datos se copian los nuevos datos al final del archivo AD y se actualiza el descriptor de bloques de archivo para que apunte a los datos originales en los bloques que se mantienen iguales y apunte a bloques nuevos para las zonas del archivo que han cambiado. Una vez que se encuentra una discrepancia se copian en el archivo AD como mínimo 4KB del archivo a

actualizar evitando así el crecimiento excesivo en número de bloques. Este proceso va fragmentando los bloques necesarios para almacenar un archivo proceso que podría reducir la eficacia de los procesos de incorporación, actualización y extracción del archivo. No obstante, este efecto negativo se palia por medio un proceso inverso consistente éste en la consolidación de bloques cuando éstos son distintos. Es decir, si el sistema se encuentra con dos o más bloques seguidos del archivo original que presentan disparidades con el que se está actualizando se almacenan los nuevos datos seguidos al final del archivo AD y se describe todo el espacio con un único bloque. Gracias al principio de localidad, ya comentado, muchos de los cambios dentro de un archivo se comprueba en la práctica, que se producen habitualmente en las mismas zonas hecho que también ayuda a mantener baja la fragmentación de los datos.

En caso de que el archivo a actualizar sea mayor que el almacenado se añade al final del archivo AD los datos nuevos y se describen los mismos en la tabla de bloques.

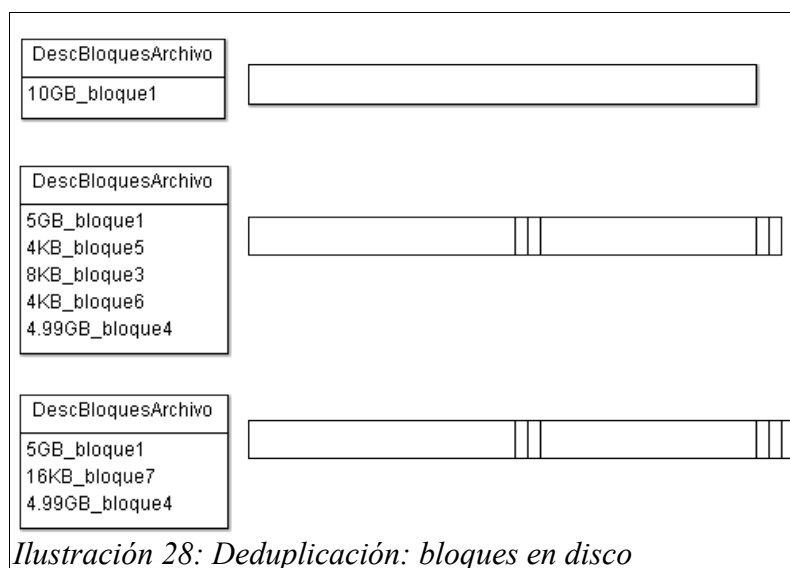


Ilustración 28: Deduplicación: bloques en disco

La figura anterior ejemplifica dicho proceso.

En el paso 1, primera copia, se incorpora al fichero AD un archivo nuevo de 10GB y que es descrito por un único bloque de 10GB.

En el paso 2, segundo backup del fichero, se detecta que han cambiado dos pequeños bloques de 4KB con 8KB que se mantienen iguales entre medias. Se añade al fichero AD los 8KB de datos nuevos y se modifica el descriptor de bloques para que apunte adecuadamente a éstos.

En un tercer backup del fichero, se detecta que han cambiado 16KB de datos en zonas solapadas con los anteriores (principio de localidad): Se añaden al archivo los 16 KB nuevos, y se actualiza la tabla de bloques consolidando bloques.

No se han implementado políticas de recuperación de huecos, si bien la rotación de archivos entre AD y BAK.AD automáticamente elimina en parte dicha necesidad. No obstante, el gran ahorro de espacio que produce esta tecnología justifica la pequeña pérdida que se produce por no recuperar huecos en el archivo AD.

5.5.7 Estudio teórico de rendimiento máximo y pruebas de laboratorio

Realizar un backup FULL con AD, en la plataforma de laboratorio descrita, debería ocupar el mismo espacio que los datos originales (106GB) y consumir unos 51 minutos:

106GB = 108.544 MB

108544/83 (83 MB/s en LECTURA ALEATORIA bloques grandes, disco D)=1307 s

108544/62 (62 MB/s en ESCRITURA secuencial, disco C)=1751 s

Total: 1307s+1751s=51 minutos

Realizar un Backup INCREMENTAL ocuparía el mismo espacio que los datos originales modificados, en promedio, 3 maquinas virtuales, 1% de cada máquina modificado al día: $3 \times 8 \times 1\% = 0.24$ GB de datos (diarias) incrementado en alguna cantidad ya que el software almacena los bloques nuevos en bloques de 4K aunque solo haya un bit cambiado en cada bloque.

En tiempo, se puede despreciar el tiempo que se tarda en seleccionar los ficheros modificados.

El sistema desarrollado necesita comparar el fichero modificado con los bloques de datos del mismo fichero la última vez que se copió por lo que debe leer el fichero original entero y los bloques almacenados enteros.

La lectura del fichero original se puede modelar como una lectura secuencial de grandes bloques y la lectura de los bloques almacenados se puede modelar como lectura aleatoria de bloques grandes.

Por lo tanto, el tiempo de lecturas será:

El tiempo de copia sería: $24 \times 1024 / 100 + 24 \times 1024 / 48 = 12.5$ minutos aproximadamente.

El tiempo de escritura, para 0.24 GB se puede considerar despreciable para este supuesto.

Resumen estimaciones teóricas:

	Ocupacion FULL	Ocupacion DIARIA	Tiempo FULL	Tiempo DIARIA	RENDIMIENTO
AD	106 GB	Alfa * 0.24 GB	51 min	12.5 min	35'5 MB/s

5.5.8 Rendimiento y pruebas

Además de las múltiples pruebas que se han realizado a lo largo del desarrollo y que han verificado la fiabilidad de la herramienta, es decir, la coincidencia perfecta entre los datos a respaldar y el contenido que recuperamos cuando extraemos los datos de la copia, se han realizado pruebas de caja blanca con cada uno de los parámetros que puede recibir la herramienta en modo consola.

No obstante, la prueba definitiva ha consistido en realizar diariamente, por un periodo de 3 semanas copias nocturnas de un conjunto de datos formado por maquinas virtuales, similar al descrito en el punto 5.4.1 de este capítulo.

Dichas copias se han realizado con el software Cobian, en modo no comprimido (ya que según se ha establecido por las pruebas aquí realizadas la compresión aumenta enormemente el tiempo de copia y sin embargo no consigue grandes mejoras en lo que a espacio se refiere con este tipo de datos y seguidamente con el software desarrollado AD.

Se ha comparado diariamente el tiempo de copia y el espacio necesario para ambas herramientas.

Al finalizar el periodo de tres semanas se ha procedido a comparar los datos respaldados por ambas herramientas buscando discrepancias de un solo bit entre ellas. Es decir, se ha recuperado en un directorio temporal, los datos del backup del día X y se han comparado con los extraídos de la otra herramienta de backup por medio de la herramienta de sistema operativo FC en modo binario.

Muestro a continuación los excelentes resultados obtenidos:

HERRAMIENTA	DATOS ORIGEN/TIEMPO	RENDIMIENTO	ESPACIO DESTINO
COBIAN (FULL)	108 GB 51 minutos	2.1 GB/min 35.8 MB/s	108 GB
COBIAN (INCRE DIARIO)	21 GB 10 minutos	2.1 GB/min 35.8 MB/s	21 GB
AD (FULL)	108 GB 35 minutos	3 GB/min 51 MB/s	108 GB
AD (INCRE DIARIO)	21 GB 21 minutos	1 GB/min 17 MB/s	150-1000 MB

A final del período de prueba de tres semanas la copia Cobian ocupa un espacio aproximado en disco de 500GB frente a los 118GB de la copia deduplicada. Lógicamente, dicha diferencia sería aún mucho mayor en caso de seguir almacenando copias por un período de 2-3 meses.

Nótese que el espacio ocupado por el backup deduplicado es el tamaño original de una sola copia de los datos más un pequeño incremento diario, siendo la suma de dichos backup incrementales de solo 10GB frente a la copia COBIAN donde a los 108 GB originales se han añadido 420GB en concepto de copia incremental, es decir, una proporción de 1 contra 42. Se puede decir entonces que AD consigue una deduplicación de **40x**.

5.5.9 Conclusiones

El desarrollo de esta herramienta ha resultado especialmente gratificante ya que se han alcanzado con gran éxito los objetivos propuestos y se ha conseguido, por tanto, una herramienta muy útil para el respaldo de datos en general y de virtualización en particular.

A nivel de ganancia de espacio en disco no tiene rival en lo que al resto de herramientas de backup probadas se refiere. En este aspecto, el resultado no ha podido ser más satisfactorio.

En cuanto al rendimiento en tiempo las copias FULL y la incorporación de ficheros de los que no se disponga de copia previa se realiza un 33% más rápido que con la copia más rápida a nivel de sistema operativo (ROBOCOPY) lo que supone un tiempo excelente.

El rendimiento temporal para ficheros de los que ya disponemos de copia previa donde, por tanto, la herramienta debe realizar las labores de comparación necesarias para descubrir que bloques de datos han cambiado, se realiza a un tercio del rendimiento anterior por lo que la copia de este tipo de ficheros lleva más o menos el doble de tiempo que la copia con ROBOCOPY si bien se mantiene dentro de parámetros razonables y en todo caso a una enorme distancia de herramientas que intentan comprimir los datos por otros medios.

En cuanto a la facilidad de gestión de las copias, el tener todos los backups perfectamente controlados desde un único catálogo permite una recuperación simple y segura frente a las de otros softwares donde se complica el asunto con copias FULL y copias incrementales que deben ser aplicadas en el orden correcto hasta recuperar la totalidad de los ficheros que queremos extraer.

Por último, las pruebas realizadas teniendo como destino un dispositivo USB, lento en comparación con un disco duro SATA II interno, pero muy frecuente en copias de seguridad domésticas y de pymes han arrojado unos tiempos excelentes: copia full a usb con mismo rendimiento que un robocopy y copia incremental (ficheros ya presentes en el fichero AD) un 20% más lentos, con una ocupación en disco mínima!

6. VALORACIÓN ECONÓMICA y de CONSUMO ENERGÉTICO

Esta sección la voy a enfocar desde el punto de vista de la comparación entre el coste estimado de adquisición de una arquitectura como la aquí propuesta y lo que costaría adquirir y mantener la infraestructura haciendo uso solo de servidores físicos. Las ventajas de la virtualización son de sobra reconocidas y abarcan muchos más aspectos que los que se van a tratar aquí: Tiempo de puesta en marcha de proyectos, ahorro en licencias, facilidades de gestión y automatización, etc..

Las capacidades limitadas de las máquinas propuestas para la virtualización permiten estimar, prudentemente, que se podría dar servicio con 4 máquinas físicas a $4 \times 15 = 60$ máquinas virtuales.

Si tuviéramos que adquirir 60 servidores físicos, con todos los costes asociados: racks, SAI, refrigeración, espacio físico, potencia eléctrica contratada, discos duros, switches, etc... y garantizar un mantenimiento de los mismos el coste sería muchísimo mayor. Si además queremos alta disponibilidad equiparable deberemos duplicar muchos de estos equipos por lo que para los cálculos siguientes voy a considerar 100 servidores físicos equivalentes. *A grosso modo:*

Adquisición	Tradicional	Virtualización
Adquisición servidores y almacenamiento	$100 \times 1000 = 100.000\text{€}$	$6 \times 1000 = 6.000\text{€}$
Armarios	$10 \times 500 = 5000 \text{ €}$	$1 \times 500 = 500\text{€}$
PDU/SAI/regletas/consolas KVM	$10 \times 1800 = 18.000\text{€}$	$1 \times 1800 = 1800\text{€}$
Refrigeración	$16 \text{ KW} = 10.000\text{€}$	$1 \text{ KW} = 2000\text{€}$
TOTAL:	133.000,00 €	10.300,00 €

(*) No se tendrá en cuenta el espacio necesario, obra civil, sistema anti-incendios, etc..

Gastos operativos/coste medioambiental	Tradicional	Virtualización
Mantenimiento servidores	$100 \times 50\text{€} = 5000\text{€}$	$6 \times 50\text{€} = 300\text{€}$
Consumo eléctrico	$100 \times 700\text{W} = 70 \text{ KW}$ Consumo: 600 MWh	$6 \times 700\text{W} = 4.2 \text{ KW}$ Consumo: 36 MWh
TOTAL:	120.000,00 €	7.200,00 €

(*) Incluye factor de corrección por rendimiento SAI y refrigeración

7. CONCLUSIONES

Al principio de este proyecto se planteó el desarrollo fundamentado con datos objetivos y experimentales de una plataforma de virtualización basada en la versión gratuita de VMWARE, VMWARE ESX 4.0. El enfoque dado al diseño ha sido integral en el sentido de abarcar todos los componentes tanto hardware, como software, tanto tecnológico como económico con el fin de conseguir un producto/idea completo.

El fin último, sería acercar la virtualización a empresas que por su tamaño les resultaría demasiado oneroso la adquisición de productos y tecnologías de alto nivel pero que por sus necesidades pueden verse muy beneficiadas en el uso de la virtualización. Este tipo de organizaciones, muy a menudo, carecen no solo de los recursos económicos sin no también, de los recursos humanos y de conocimientos necesarios para abordar con éxito este tipo de proyectos por lo que un diseño integral, casi de caja negra o appliance puede resultar clave para su aceptación. (CPD in a BOX)

A lo largo del proyecto, se han realizado pruebas de hardware y tecnologías de base necesarias para solucionar problemas comunes a muchas arquitecturas. En todo momento, se ha intentado mantener una filosofía lo más abierta e integradora posible entre el mundo del software libre, el mundo del software propietario y el software gratuito.

En lo referente al hardware se ha cuestionado o , al menos comparado, la utilidad de tecnologías punteras y caras frente a material doméstico de consumo. Intentando obtener al mínimo precio, el mejor rendimiento comparado con el que se puede obtener invirtiendo mucho más dinero.

En lo referente al desarrollo de software para apoyar la arquitectura diseñada se han realizado dos aportaciones: Por un lado una herramienta de backup deduplicado de datos, realizada partiendo de cero y que ha obtenido unos resultados excepcionales y que en algunos escenarios puede resultar equiparable a otras de uso profesional y de precios, asociadas a HW, no inferiores a 50.000€.

El desarrollo central, por otro lado, ha sido el llamado FVC, Free Virtual Center, del que tengo la intención de publicar con licencia AGPL, que ha alcanzado los objetivos los objetivos planificados: Proveyendo características de gestión avanzadas, visión unificada de máquinas, backup centralizado, monitorización, VMOTION, despliegue de plantillas y en resumen proporciona a la versión gratuita de VMWARE funcionalidades muy similares, y en algunos casos superiores, a la versión de pago del mismo.

Por último y, no menos importante, como en cualquier otro proyecto he aprendido bastantes cosas durante su desarrollo: He interiorizado que JAVA puede ser utilizado para el desarrollo y la automatización de herramientas de sistemas, he profundizado en las posibilidades de AJAX, he quedado gratamente sorprendido por el uso y las posibilidades del sistema de archivos ZFS, he visto como los razonamientos teóricos son precisos limitando el rendimiento máximo de las tecnologías pero el rendimiento mínimo viene influido por multitud de factores, bugs, implementaciones, cuellos de botella y, por último, he “sufrido en mis carnes” la planificación excesivamente optimista en lo que ha objetivos a cubrir se refiere del proyecto que ha obligado, para respetar los hitos de entrega, a trabajar muchas más horas de las planificadas inicialmente, lo me ha enseñado a valorar los tiempos de forma más prudente.

8. GLOSARIO

VMFS: Sistema de archivos empleado por VMWARE. Permite uso en clusters.

NFS: Sistema de archivos en red empleado especialmente por sistemas UNIX.

ISCSI: Tecnología de acceso a SAN consistente en encapsular peticiones SCSI en tramas IP.

FTP: Protocolo de transferencia de ficheros.

RAID: Familia de técnicas de redundancia de discos.

ZFS: Sistema avanzado de archivos utilizado por Solaris.

DEDUPLICACIÓN: Reducción del espacio de almacenamiento aprovechando la similitud entre bloques de datos.

OPENFILER: Distribución de LINUX diseñada como servidor de archivos, SAN, NAS y otros.

CIFS: Protocolo de compartición de archivos de Windows.

ESXi: Versión gratuita de VMWARE ESX

VMOTION: Migración, en caliente, de una máquina virtual. Sin pérdida de servicio.

HA – ALTA DISPONIBILIDAD: Técnicas enfocadas a garantizar la continuidad del servicio ante incidencias de hardware y/o software.

9. BIBLIOGRAFÍA

Temarios y manuales de algunas de las asignaturas de la carrera especialmente de ISCD, de PIS y de METODOLOGÍA.

Documentación oficial de VMWARE de las versiones 3.5, 4.0 y 4.1.

Manual VMWARE VIX.

Multitud de fuentes en internet, especialmente, al menos para visiones globales: wikipedia.

Documentación de Sun: docs.sun.com/app/docs

Thinking of Java version 3

Documentación recibida en cursos del INAP sobre JAVA y sobre VMWARE.