

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“EVALUACIÓN DE LOS SISTEMAS DE DEDUPLICACIÓN OPENDEDUP (SDFS) Y ZFS PARA OPTIMIZAR EL SISTEMA DE ALMACENAMIENTO EN UN SERVIDOR DE BACKUPS”**

Trabajo de titulación presentado para optar al grado académico de:

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**AUTOR:** ANIBAL FERNANDO MANCHENO SAMANIEGO

**TUTOR:** ING. MARCO VINICIO RAMOS VALENCIA MSc.

Riobamba-Ecuador

2015

**©2015**, Anibal Fernando Mancheno Samaniego

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

El Tribunal del Trabajo de Titulación certifica que: El trabajo de investigación: EVALUACIÓN DE LOS SISTEMAS DE DEDUPLICACIÓN OPENDEDUP (SDSF) Y ZFS PARA OPTIMIZAR EL SISTEMA DE ALMACENAMIENTO EN UN SERVIDOR DE BACKUPS*,* de responsabilidad del señor Anibal Fernando Mancheno Samaniego, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

**NOMBRES FIRMA FECHA**

ING. GONZALO SAMANIEGO

**DECANO \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

DR. JULIO SANTILLÁN

**DIRECTOR DE ESCUELA \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

ING. VINICIO RAMOS MSc.

**DIRECTOR DE TESIS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

ING. GERMANIA VELOZ

**MIEMBRO DEL TRIBUNAL \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NOTA DE LA TESIS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Yo, Anibal Fernando Mancheno Samaniego soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del Trabajo de Titulación pertenece a la Escuela Superior Politécnica de Chimborazo.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ANIBAL FERNANDO MANCHENO SAMANIEGO

###### DEDICATORIA

Dedico este trabajo de titulación con mucho cariño a mis padres ANIBAL y ANTONIETA, de quienes siempre he recibido apoyo incondicional en todas las etapas de mi vida, alentándome a seguir siempre adelante y a no desmayar pese a las dificultades que sobrevengan.

Anibal

###### AGRADECIMIENTO

Agradezco primeramente a DIOS, por todas las bendiciones que ha derramado en mi vida, por ayudarme a progresar en lo temporal y espiritual, por las pruebas que me dio, sé que fueron necesarias en la preparación del camino que me llevara a culminar con éxito mi carrera profesional. A mis amorosos padres ANIBAL y ANTONIETA, porque sin su ayuda, apoyo y sabios consejos me hubiese sido difícil alcanzar esta meta, ellos son mi mayor inspiración, ejemplo de vida y superación. A VANESSA, por su apoyo y ayudada constante, por alentarme a seguir siempre adelante en los momentos de dificultad. A mi amiga PAMELA, quien me demostró lo que es una verdadera amistad y fue parte importante en la realización de este logro. A FERNANDO, EDISON, NARCISA, ANITA Y JORGE por sus palabras de apoyo y motivación que me empujaron para culminar con este último peldaño, así como también a DIEGO, JOFFRE, GUSTAVO y a todos mis amigos y compañeros quienes formaron parte de este proceso, de quienes aprendí también a esforzarme día a día para dar lo mejor de mí. A mis profesores, sin duda alguna ellos fueron los escultores que me moldearon durante mi carrera universitaria, de ellos adquirí los conocimientos y valores necesarios para escalar los peldaños que me llevarían a culminar esta etapa de mi vida. Gracias a todos por hacer este sueño realidad.

Anibal

**CONTENIDO**

[DERECHO DE AUTOR ii](#_Toc433868798)

[CERTIFICACIÓN iii](#_Toc433868798)

[DECLARACIÓN DE RESPONSABILIDAD iv](#_Toc433868798)

[DEDICATORIA v](#_Toc433868798)

[AGRADECIMIENTO vi](#_Toc433868798)

[TABLA DE CONTENIDO vii](#_Toc433868798)

[INDICE DE TABLAS x](#_Toc433868798)

[INDICE DE ILUSTRACIONES xi](#_Toc433868798)

[INDICE DE ANEXOS](#_Toc433868796)

[RESUMEN xiii](#_Toc433868797)

[SUMARY xiv](#_Toc433868797)

[INTRODUCCIÓN 1](#_Toc433868798)

**CAPÍTULO I**

**MARCO TEÓRICO**

[**1.1. Introducción a la deduplicación de archivos** 7](#_Toc433868807)

[***1.1.1. Definiciones*** 7](#_Toc433868808)

[***1.1.2. Características*** 8](#_Toc433868809)

[**1.2. Ventajas y desventajas de la deduplicación de datos** 9](#_Toc433868810)

[***1.2.1. Ventajas*** 9](#_Toc433868811)

[***1.2.2. Desventajas*** 9](#_Toc433868812)

[**1.3. Ratios de deduplicación** 10](#_Toc433868813)

[***1.3.1. Definición*** 10](#_Toc433868814)

[***1.3.2. ¿Cómo se calculan?*** 10](#_Toc433868815)

[***1.3.3. Factores que influyen en los ratios de deduplicación*** 13](#_Toc433868816)

[**1.4. Tipos de deduplicación** 14](#_Toc433868817)

[***1.4.2. Deduplicación In-line (Origen)*** 14](#_Toc433868818)

[***1.4.3. Deduplicación Out-of-line (Destino)*** 14](#_Toc433868819)

[**1.5. Niveles de granularidad** 15](#_Toc433868820)

[***1.5.1. Deduplicación a nivel de archivo*** 15](#_Toc433868821)

[***1.5.2. Deduplicación a nivel de bloque*** 16](#_Toc433868822)

[*1.5.2.1. Bloques de longitud fija* 16](#_Toc433868822)7

[*1.5.2.2. Bloques de longitud variable* 16](#_Toc433868822)8

[***1.5.3. Deduplicación a nivel de byte*** 18](#_Toc433868823)

[**1.6. Opendedup (SDFS)** 19](#_Toc433868824)

[**1.7. ZFS** 20](#_Toc433868825)

[***1.7.1. Ventajas*** 20](#_Toc433868826)

[***1.7.2. ¿Cuál es el Costo?*** 21](#_Toc433868827)

[***1.7.3. ¿Cuánta memoria se necesitan para mantener la tabla dedup ZFS?*** 22](#_Toc433868828)

[**1.8. Sistema operativo Linux** 22](#_Toc433868829)

[***1.8.1. Origen*** 22](#_Toc433868830)

[**1.9. Métodos de almacenamiento de deduplicación** 26](#_Toc433868831)

[***1.9.1. Deduplicación de longitud variable*** 26](#_Toc433868832)

[*1.9.1.1. ¿Quién elige la deduplicación de longitud variable y por qué?* 26](#_Toc433868832)

[*1.9.1.2. ¿Cómo funciona la deduplicación de longitud variable?* 26](#_Toc433868832)

[*1.9.1.3. Beneficios de la deduplicación de longitud variable* 26](#_Toc433868832)

[***1.9.2. Deduplicación de longitud fija*** 27](#_Toc433868833)

[*1.9.2.1. Limitaciones* 27](#_Toc433868833)

[**1.10. Disponibilidad de Deduplicación** 27](#_Toc433868834)

[**1.11. Tablas Hash** 28](#_Toc433868835)

[**1.12. Algoritmo de Compresión LZ4**. 28](#_Toc433868836)

[**CAPITULO II**](#_Toc433868837)

**MARCO METODOLÓGICO**

[**2.1. Introducción** 30](#_Toc433868838)

[**2.2. Sistemas de archivo deduplicados** 30](#_Toc433868839)

[***2.2.1. Instalación y configuración del sistema operativo Linux Mint con el soporte para ZFS y SDFS***……..…………………...………………………………………...31](#_Toc433868840)

[*2.2.1.1. Instalación del soporte para ZFS desde el repositorio oficial de ZFS nativo para Ubuntu*……….…………………………………….………………………...….3](#_Toc433868840)2

[*2.2.1.2. Espacio para SDFS*……….………….………………….……………………….…...3](#_Toc433868840)5

[**2.3. Establecimiento de los parámetros para la elección de un sistema de archivos deduplicado para la implementación en un servidor de backups** 36](#_Toc433868870)

**CAPÍTULO III**

**MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS DE RESULTADOS**

[**3.1. Estudio comparativo de sistemas de archivos para deduplicación** 39](#_Toc433868871)

[***3.1.1. Estudio a priori*** 39](#_Toc433868872)

[***3.1.2. Estudio a posteriori*** 42](#_Toc433868873)

[*3.1.2.1. Medición en el entorno de funcionamiento* 42](#_Toc433868873)

[*3.1.2.2. Medición de características de funcionamiento SDFS y ZFS* 42](#_Toc433868873)5

[***3.1.3. Resultados obtenidos*** 47](#_Toc433868874)

[**3.2. Planteamiento del prototipo** 48](#_Toc433868875)

[***3.2.1. Diseño del escenario de pruebas*** 48](#_Toc433868876)

[*3.2.1.1. Requerimientos hardware* 48](#_Toc433868876)

[*3.2.1.2. Configuración de la red* 48](#_Toc433868876)

[*3.2.1.3. Requerimientos software* 48](#_Toc433868876)9

[***3.2.2. Implementación del escenario de pruebas*** 49](#_Toc433868877)

[*3.2.2.1. Instalación y configuración de ZFS* 49](#_Toc433868877)

[*3.2.2.2. Instalación y configuración de Bacula*](#_Toc433868877) 50

[CONCLUSIONES](#_Toc433868880) 53

[RECOMENDACIONES](#_Toc433868880) 54

[GLOSARIO](#_Toc433868878)

[BIBLIOGRAFÍA](#_Toc433868879)

[ANEXOS](#_Toc433868880)

###### INDICE DE TABLAS

[**Tabla 1-1:** Relación de la reducción de ratios y porcentajes. 12](#_Toc433874490)

[**Tabla 2-3:** Parámetros para la elección a priori de un sistema de archivos deduplicado. 39](#_Toc433874491)

[**Tabla 3-3:** Parámetros ponderados. 40](#_Toc433874492)

[**Tabla 4-3:** Medidas de ponderación. 41](#_Toc433874493)

[**Tabla 5-3:** Análisis a posteriori de parámetros de funcionamiento. 43](#_Toc433874494)

[**Tabla 6-3:** Ponderación de puntos de uso. 44](#_Toc433874495)

[**Tabla 7-3:** Análisis de características de funcionamiento. 45](#_Toc433874496)

[**Tabla 8-3:** Ponderación análisis de características de funcionamiento. 46](#_Toc433874497)

[**Tabla 9-3:** Resumen de resultados. 47](#_Toc433874498)

[**Tabla 10-3:** Requerimientos software para la implementación de ZFS. 49](#_Toc433874499)

###### INDICE DE ILUSTRACIONES

[**Figura 1-1.** Proceso general de la tecnología de la deduplicación.](#_Toc433873015) 8

[**Figura 2-1.** Proceso para calcular los ratios de deduplicación.](#_Toc433873015) 11

[**Figura 3-1.** Porcentajes de reducción en relación a los ratios.](#_Toc433873015) 12

[**Figura 4-1.** Deduplicación In-Line.](#_Toc433873015) 14

[**Figura 5-1.** Deduplicación Out-of-Line.](#_Toc433873016) 15

[**Figura 6-1.** Deduplicación a nivel de archivo (SIS).](#_Toc433873017) 16

[**Figura 7-1.** Deduplicación a nivel de bloque.](#_Toc433873018) 17

[**Figura 8-1.** Deduplicación a nivel de byte.](#_Toc433873019) 19

[**Figura 9-2.** Ventana de inicio de sesión.](#_Toc433873019) 31

[**Figura 10-2.** Particiones del disco duro.](#_Toc433873019) 33

[**Figura 11-3.** Gráfico análisis de cumplimiento condiciones a priori entre SDFS y ZFS.](#_Toc433873019) 42

[**Figura 12-3.** Gráfico comparación A Posteriori entre SDFS y ZFS.](#_Toc433873019) 44

[**Figura 13-3.** Gráfico de soporte de funciones.](#_Toc433873019) 46

# INDICE DE ANEXOS

**Anexo A.** Instalación y configuración del servidor Bacula.

**Anexo B.** Uso de Bacula.

**Anexo C.** Mensaje de Linus Benedict Torvalds.

**Anexo D.** Antes y después de la deduplicación con ZFS.

[**Anexo E.** Gráfico del uso de Linux en Servidores Enero 2011 – Octubre 2013](#_Toc433873014).

[**Anexo F.** Gráfico del uso de Linux en diferentes áreas.](#_Toc433873014)

[**Anexo G.** Pruebas realizadas del antes y después con la deduplicación ZFS en archivos de imagen, audio y video.](#_Toc433873014)

# RESUMEN

El objetivo es realizar una evaluación de los sistemas de deduplicación Opendedup (SDFS) y ZFS, con la finalidad de escoger el de mejores prestaciones para la fundamentación de un prototipo a ser usado a futuro en entornos de servidores de backup más económicos y eficientes. Para el desarrollo se levantó un escenario de pruebas para evaluar los sistemas propuestos. Mediante la aplicación de herramientas de licenciamiento libre como Virtualbox, Bacula, etc, se llevó a cabo la instalación de las soluciones de deduplicación bajo una distribución Linux Mint 17.1 virtualizada con la herramienta Virtualbox, montada sobre una máquina física, que además contiene el servidor Bacula mediante el cual se generarán y almacenarán los backups en nuestros ambientes deduplicados por SDFS y ZFS. Una vez levantado el escenario de pruebas, se sometió a un entorno de funcionamiento, en el cuál se tomaron las métricas especificadas en los capítulos 2 y 3 para realizar el análisis comparativo. Los resultados obtenidos demostraron que SDFS tuvo un cumplimiento del 40% en la comparación a priori frente al 65% de ZFS, además SDFS obtuvo 400 puntos de uso en la comparación a posteriori, mientras que ZFS solo tuvo 194.74 puntos de uso, por último SDFS logró 2 puntos en comparación de funcionalidades superado por los 5 puntos de ZFS. Se concluye que ZFS, ofrece mejores prestaciones en cuanto a la optimización del almacenamiento en un servidor de backups, pese a las dificultades presentadas durante su instalación, siendo su método de deduplicación superior a SDSF basado en la cantidad de disco economizado. Se recomienda que los backups a ser generados sean incrementales, en formato archivo, debido a que en este tipo de datos es más evidente el porcentaje de deduplicación, al tener datos base hacia quienes apuntar.

**Palabras claves:** <SOFTWARE [LINUX MINT 17.1]>, <SOFTWARE [SDFS]>, <SOFTWARE [ZFS]>, <SOFTWARE [VIRTUALBOX]>, <SOFTWARE [BACULA]>, <SISTEMAS>, <PROTOTIPO>

**SUMMARY**

An evaluation of Opendedup (SDFS) and ZFS deduplication systems was carried out in order to get the best benefits for a prototype to be used in the future in more economical and efficient backup server areas. Test scenery was used to evaluate the proposed systems. The deduplication solution installation with Linux distribution 17.1 virtualized with Virtualbox tool, on a physical machine that contains Bacula server was carried out by means of the application of free-license tools such as Virtualbox, Bacula, etc. By means Bacula server backups will be generated and kept in deduplicated by SDFS and ZFS. When the scenery of tests was used, it was put down to running to take the metrics described in the chapters II and III in order to carry out the comparative analysis. The results demonstrated that SDFS accomplished with 40% before comparing to 65% of ZFS. Besides SDFS got 400 points of use after comparing while ZFS only got 194.74 points of use. SDFS got 2 points compared to the functions overcoming 5 points of ZFS. It is concluded that ZFS offers good benefits to optimize the storing in a backup server despite the obstacles in the installation so, the deduplication method was higher than SDFS based in the quantity of the economized disk. It is recommended that these backups should be increased in file format because in this kind of data the percentage of deduplication is more evident when having data basis.

**Key words:** <SOFTWARE [LINUX MINT 17.1]>, <SOFTWARE [SDFS]>, <SOFTWARE [ZFS]>, <SOFTWARE [VIRTUALBOX]>, <SOFTWARE [BACULA]>, <SISTEMAS>, <PROTOTIPO>

# 

# INTRODUCCIÓN

##### 

##### Antecedentes

## Planteamiento del problema

### Descripción del objeto de estudio

Un servidor de backups es un servidor que tiene un software dedicado a la extracción y organización de información para el aseguramiento de la misma, tratando de prevenir la pérdida de información por motivos tanto internos como externos.

Una de las características primordiales en este tipo de servidores ha sido siempre la gran cantidad de espacio que se requiere, para almacenar todas las versiones que se generan de los archivos a través del tiempo.

Dependiendo de las necesidades del negocio un servidor de backups puede implementarse en un entorno local o en un entorno remoto, variando los mismos por su naturaleza en el rendimiento al momento de realizar las copias de archivos.

#### Situación actual del objeto de estudio

En la actualidad los servidores de backups proveen una solución efectiva a problemas como perdida de información por eventos internos o externos, permitiéndonos recuperar parte de la misma e inclusive en muchos casos, la totalidad de la misma tomándola de un punto anterior en el tiempo.

Actualmente el manejo de información se está volviendo uno de los puntos fundamentales en cualquier negocio, teniendo negocios que inclusive se dedican totalmente al manejo de la misma como Google o Facebook.

Dentro de lo cual, las herramientas de backups han ido tomando importancia y evolucionando a través del tiempo, hasta llegar a ser servidores completamente dedicados al manejo de los mismos.

Al realizarse respaldos periódicos de los datos, la tendencia de los mismos es a utilizar un gran espacio en disco, el cual nos da una limitante en cuanto a la cantidad de información respaldada en muchos de los casos.

Como respuesta a esta necesidad nace la tecnología de backups para un ahorro drástico en cuanto a almacenamiento se refiere, aunque hoy en día aún más importante, produce una reducción en cuanto a uso de datos al subir y bajar información de un servidor de backups online, siendo mucho más eficiente subir backups incrementales que duplicar la información.

En este entorno, aparecen empresas y tecnologías enfocadas a esta implementación, obteniendo así una gran gama de soluciones para implementar en un entorno de backups deduplicado.

Por esto es importante la evaluación de diferentes alternativas previo a la elección de un servidor de aplicaciones para la futura implantación de un sistema con estas características. Este proceso de análisis debe estar enfocado a los distintos parámetros recomendados por expertos y empresas proveedoras de servidores, siempre buscando que el conjunto de características de un producto como tal se ajusten a las capacidades ideales que la empresa requiere. Entre los parámetros que se toman en cuenta, actualmente, antes de la elección de una tecnología para un servidor de backups deduplicado, se plantean los siguientes:

* Rapidez y rendimiento antes, durante y después la generación de un backup.
* Existencia de información actual en las páginas oficiales.
* Existencia de manuales y foros para solventar dudas.
* Existencia de ejemplos documentados sobre la utilización y funcionamiento de la tecnología
* Compatibilidad con otros sistemas de archivos NO DEDUPLICADOS.
* Funcionamiento multiplataforma.
* Integración con herramientas para servidores de backup.
* Documentación sobre integración con herramientas para automatizar backups.

Para la implementación del servidor de backups y basados en el **Anexo E,** se decide realizar la implementación en Linux Mint 17.1 basada en Debian.

En los ámbitos de uso de Linux, cabe mencionar la reseña hecha en la web de hipertextual.com, en la cual el usuario “andewarrs” manifiesta los porcentajes de uso de Linux en las diferentes áreas, lo cual se resume en el **Anexo F.**

#### Análisis del objeto de estudio

#### 

* Servidor de BACKUPS
* ZFS
* OPENDEDUP (SDFS)

### Formulación del problema

¿Qué sistema de archivos deduplicado presenta mejores características para la implementación en un servidor de backups?

### Sistematización

* ¿Qué es backup?
* ¿Qué es Servidor de Backups?
* ¿Qué es un sistema de Archivos?
* ¿Sistemas de Archivos Decuplicados?
* ¿Aplicaciones para manejar servidores de backup?
* ¿Qué es ZFS?
* ¿Qué es SDFS?
* ¿Cuál es la mejor opción ZFS O SDFS?

### Justificación del trabajo de estudio

#### Justificación Teórica

La implementación de un sistema de deduplicación, se constituye en una solución proactiva para las empresas e instituciones que guardan una cantidad ingente de datos en sus equipos y principalmente servidores. Estos últimos requieren nuestra especial atención debido al importante trabajo que realizan. Los servidores son los que se encargan de almacenar, manejar y procesar todas las peticiones realizadas por los clientes. La presencia de contenidos duplicados en el sistema de almacenamiento de un servidor provoca una disminución considerable del espacio disponible en disco.

Por lo general, un servidor almacena información sensible que se necesita que esté respaldada, pero este proceso de respaldo requiere una gran cantidad de espacio de almacenamiento disponible. Si se implementara un proceso de deduplicación que permita contraer el espacio que se necesita consumir para guardar esta información, entonces se lograría generar un ahorro significativo del espacio útil en el dispositivo de almacenamiento.

Una de las situaciones más frecuentes donde se encuentra información duplicada es en los sistemas de copias de seguridad. Un servidor de backup crea respaldos periódicos de la información importante en una organización.

En Ecuador la mayoría de instituciones respaldan su información, con la finalidad de asegurarla. Este proceso implica que la probabilidad de tener datos duplicados aumente y el espacio útil en disco disminuya. Se propone aplicar una solución de deduplicación que se encargue de eliminar datos redundantes con la finalidad de optimizar el sistema de almacenamiento en un servidor de backups.

Actualmente, el uso de servidores de respaldo en la nube está en un gran auge, por lo cual cada uno de los bits que contiene nuestro archivo representa tiempo y costos en cuanto a uso de internet, con los nuevos planes de internet ilimitados, el uso deja de ser un problema, pero el tiempo que involucra subir y bajar la gran cantidad de datos, es un factor trascendental.

Al tener archivos deduplicados, la cantidad de datos a subir y bajar reduce considerablemente, dado que no se suben datos repetidos.

Del lado del proveedor de servicios de backup, por la naturaleza incremental de los backup, nos vemos en un escenario inundado de archivos repetidos, los cuales si deben ser almacenados individualmente, causan un uso de espacio innecesario, comprometiendo así el funcionamiento de nuestro negocio y disparando los gastos en espacio de almacenamiento.

#### Justificación Aplicativa

En las empresas los datos crecen a un ritmo acelerado y uno de los principales motivos por los que se da ese crecimiento son las constantes copias de seguridad que se realizan como medida de seguridad, lo cual da a la vez como resultado información duplicada. Debido a esto, la mayoría de fabricantes de backups implantan en sus soluciones esta tecnología que permite tener un ahorro considerable de espacio en los sistemas de almacenamiento. (Dominguez, 2014, http://www.whitebearsolutions.com/la-deduplicacion-en-el-backup/).

La mayoría de las instituciones apuntan por una solución que les garantice el respaldo de toda su información sin tener la preocupación de que estos datos se dupliquen y resten espacio útil en el dispositivo de almacenamiento. La deduplicación ofrece el entorno ideal para esto y se adapta a las necesidades requeridas.

Para poder llevar a cabo la implementación del prototipo de una solución de deduplicación en un servidor de backup, es necesario hacer uso de una herramienta de copias de seguridad como *AMANDA, BACULA, BACKUP MANAGER*, *etc.*, que permitan levantar un servidor de respaldo en un entorno LINUX. El uso de la herramienta adecuada combinada con los sistemas de deduplicación propuestos, proporcionarán el entorno ideal para evaluar el desempeño de dichos sistemas.

Actualmente existe software de deduplicación licenciado como *ACRONIS*. Sin embargo para el presente trabajo se hará uso de soluciones de deduplicación open source debido a su licenciamiento libre y por ser estables (Slideshare, 2011, http://www.slideshare.net/RedWireServices/open-source-data-deduplication-8869930). La elección de los sistemas de deduplicación planteados Opendedup (SDFS) y ZFS se estableció por ser actualmente dos de las soluciones más populares que permiten deduplicación en línea(Hoekstra y Daan, 2012, https://www.os3.nl/\_media/2011-2012/courses/lia/berry\_hoekstra\_daan\_wagenaar\_-\_lia-report-final.pdf), a diferencia de sistemas libres como *BTRFS, BACKUP PC, etc*.

Debido al difícil acceso que se tiene a un servidor de BACKUPS real dentro de una empresa por la información sensible que maneja, se plantea el levantamiento de un servidor de pruebas en una máquina física que simule las características de un servidor de backups real, del cual se realizarán las copias de seguridad para que posteriormente sean procesadas por el sistema de deduplicación empleado (Opendedup y ZFS) y así determinar el ahorro conseguido en el sistema de almacenamiento.

##### **Objetivos**

## **Objetivo general**

Evaluar los sistemas de deduplicación Opendedup (SDFS) y ZFS para optimizar el sistema de almacenamiento en un servidor de backups.

## **Objetivos específicos**

* Estudiar la tecnología de deduplicación (Opendedup y ZFS), sus características, beneficios y desventajas.
* Recopilar información sobre los sistemas de deduplicación opendedup y ZFS, y demás herramientas a emplearse a fin de levantar un escenario de pruebas en un entorno Linux.
* Evaluar los sistemas de deduplicación opendedup y ZFS implementados en el escenario de pruebas.
* Implementar un prototipo con el sistema de deduplicación que mejores prestaciones ofrezca.

##### Métodos y técnicas

El presente proyecto comprende un tema de investigación, por lo tanto los métodos a utilizarse son el método científico y experimental, ya que se apegan a los lineamientos de los procesos. A continuación presenta la siguiente estructuración:

* **Observación:** Reconocimiento del problema que se estudiará.
* **Planteamiento del problema:** Interrogante a solucionarse.
* **Experimentación:** Procedimiento de implementación en base a conocimiento teórico.
* **Análisis de los resultados:** Obtener nueva información posterior a la experimentación para su evaluación.
* **Conclusiones:** Se resume la información obtenida y se concluye en base a la verificación de los objetivos inicialmente planteados.

**CAPÍTULO I**

##### MARCO TEÓRICO

## Introducción a la deduplicación de archivos

### Definiciones

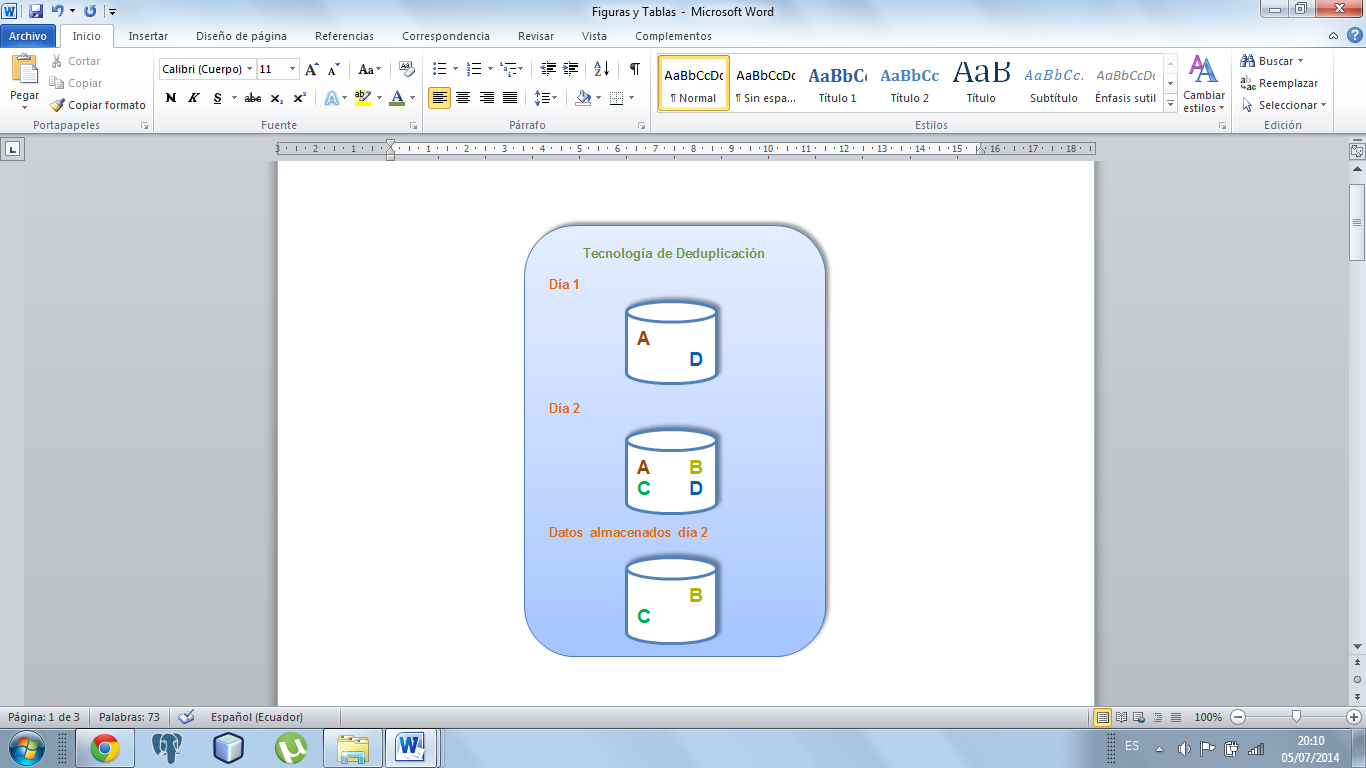
La deduplicación de datos busca la redundancia de secuencias de bytes en ventanas de comparación de gran tamaño. Las secuencias de datos (de más de 8 KB de longitud) se comparan con el historial de otras secuencias semejantes. Se hace referencia a la primera versión almacenada de forma exclusiva de una secuencia, en vez de almacenarla de nuevo. Este proceso queda completamente oculto para los usuarios y las aplicaciones, de modo que todo el archivo es legible después de su escritura. (EMC, 2015, http://mexico.emc.com/corporate/glossary/data-deduplication.htm#).

La deduplicación es una tecnología de reducción de datos orientada a eliminar datos redundantes (duplicados) en un sistema de almacenamiento de datos, almacenando solo una instancia de cada elemento, con el objetivo de reducir el espacio de almacenamiento y el consumo de ancho de banda de red. La tecnología de deduplicación se basa en un índice que referencia la ubicación de cada bloque de datos almacenado en el repositorio. El software que realiza este proceso busca cada nuevo dato que llega al sistema entre todos los datos ya almacenados previamente y almacena este dato solo si no coincide con ningún dato almacenado previamente. (LORTU, 2013, http://www.lortu.es/Technology.aspx).

La deduplicación es una tecnología de reducción de datos cuyo principal objetivo consiste en eliminar datos redundantes (duplicados), para optimizar el espacio de almacenamiento y consumo de ancho de banda de red (LORTU, 2013, http://www.lortu.es/Technology.aspx).

En su forma más general el proceso del sistema de deduplicación consiste almacenar una única copia de los datos almacenados en el disco duro y reemplazar las demás por un indicador a dicha copia. Al momento que un nuevo dato ingresa en el sistema éste es comparado con los datos almacenados previamente y será guardado únicamente si no coincide con los demás datos.

En la **Figura 1-1,** se puede apreciar dicho proceso.



**Figura 1-1.** Proceso general del sistema de

deduplicación.

**Realizado por:** MANCHENO, A. 2015

La deduplicación permite eliminar los datos redundantes almacenados en el disco duro, creando una única copia idéntica de los datos y reemplazando los datos redundantes por punteros a esa única copia (Steffens, 2010, http://liacolombia.com/2010/06/%C2%BFque-es-la-desduplicacion-de-datos-y-como-beneficia-a-las-empresas/).

### Características

A continuación se detallan las principales características de la deduplicación de datos. (Bueno Béjar, 2013, http://blogs.itpro.es/abueno/2013/10/18/data-deduplicacion-en-windows-server-2012-r2-parte-ii-de-ii/).

* **Optimiza la capacidad de almacenamiento.-** Permite almacenar una mayor cantidad de datos en menos espacio físico.
* **Es escalable y eficiente.-** La deduplicación de datos puede ejecutarse paralelamente con grandes volúmenes de datos si afectar su rendimiento del servidor.
* **Garantiza la confiabilidad e integridad de los datos.-** Mantiene la redundancia en los datos referenciados con mayor frecuencia, para garantizar la recuperación de los mismos en caso de corrupción de datos.
* **Mejora el ancho de banda.-** Los tiempos de descargas de archivos son más rápidos y reduce el consumo de ancho de banda.

## Ventajas y desventajas de la deduplicación de datos

En el siguiente punto se enumerarán algunas ventajas que actualmente ofrece la técnica de la deduplicación e datos, así como también se mencionarán sus desventajas.

### Ventajas

* **Optimización del espacio físico en disco.-** Esta es una de las principales ventajas que ofrecen las soluciones de deduplicación y por la que la mayoría de organizaciones hacen uso de ella, gracias a esto se consigue un ahorro del espacio físico de la unidad de almacenamiento, los resultados más óptimos se logran cuando existen grandes volúmenes de información duplicada (HP, 2009, http://h30458.www3.hp.com/es/es/smb/736096.html).
* **Reducción de costos de almacenamiento.-** Es otras de las ventajas que nos ofrece la tecnología de la deduplicación, puesto que aparte de ganar un ahorro de espacio en el disco, se reduce también la frecuente adquisición de nuevos sistemas de almacenamiento.
* **Libera el ancho de banda de red.-** Aparte de reducir el almacenamiento de datos redundantes, la deduplicación también reduce las necesidades de ancho de banda de red, ya que la cantidad de datos de archivos y respaldo que se transfieren contienen menos datos duplicados.
* **Ayuda en la recuperación rápida ante desastres.-** Esto es posible ya que la cantidad de información que debe transmitir contiene menos datos duplicados, haciendo que la recuperación sea más rápida.

### Desventajas

* **No todos los tipos de datos se benefician con la deduplicación.-** Tipos de datos como imágenes, audio y video que a menudo ocupan grandes tamaños en disco, se ven poco favorecidos con la deduplicación debido a que contienen pocos datos redundantes (Dorion, 2009, http://searchdatacenter.techtarget.com/es/consejo/Tecnologia-de-deduplicacion-de-datos-casos-que-justifican-la-deduplicacion).
* **Datos temporales.-** Los datos almacenados temporalmente en el disco duro, no residen el tiempo suficiente para que los algoritmos de deduplicación actúen, esto significa que los ratios de deduplicación serán muy reducidos. Es por eso que la deduplicación actúa mejor con datos almacenados en el disco duro a largo plazo. (Dorion, 2009, http://searchdatacenter.techtarget.com/es/consejo/Tecnologia-de-deduplicacion-de-datos-casos-que-justifican-la-deduplicacion).

## Ratios de deduplicación

A continuación se definirá lo que es un ratio de deduplicación, cómo se calcula y los factores que influyen en el ratio (Jimenez, 2009,

http://oa.upm.es/1803/1/PFC\_FRANCISCO\_JAVIER\_JIMENEZ\_PATRICIO.pdf).

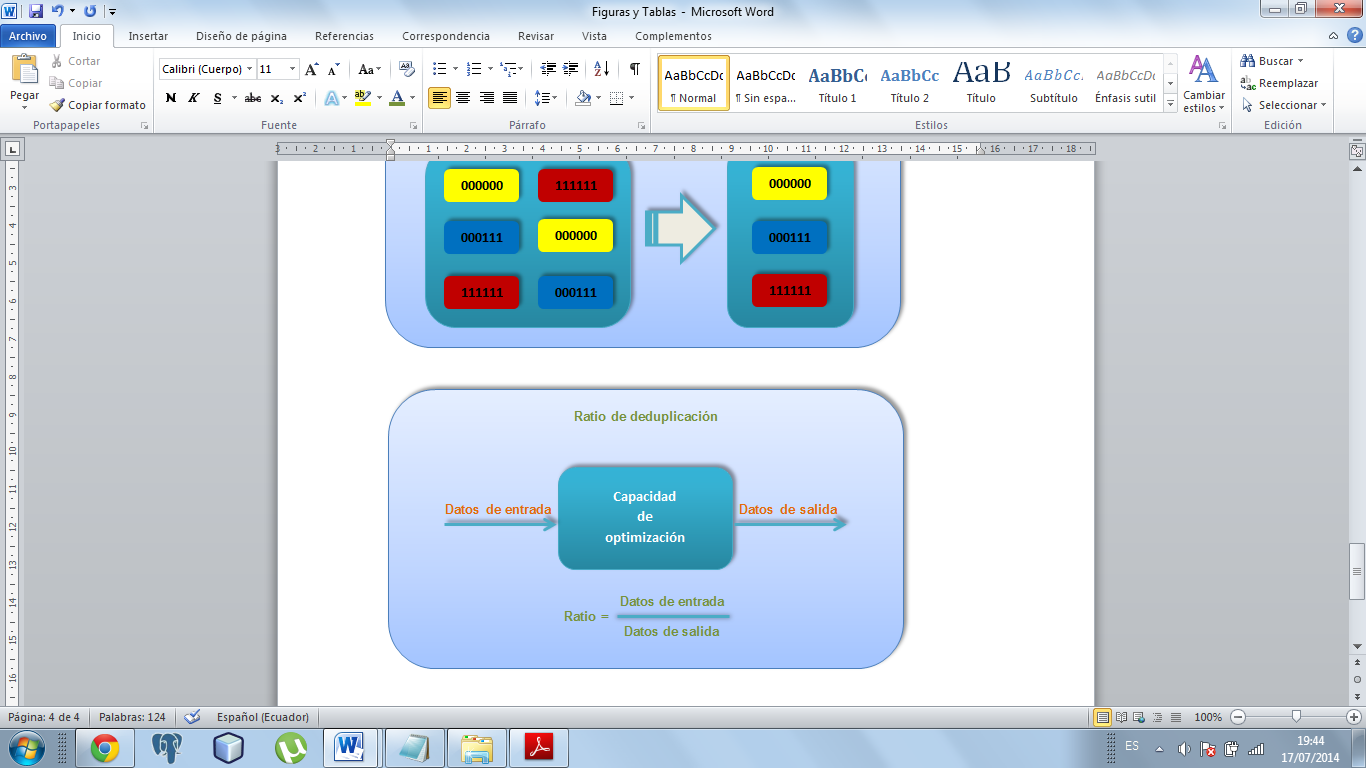
### **Definición**

Un ratio de deduplicación se define como la capacidad de optimización de datos en un sistema de almacenamiento que posee una solución de deduplicación. El ratio expresa la relación entre la cantidad de datos antes del proceso de deduplicación frente a la cantidad de datos después del proceso de deduplicación. (Jimenez, 2009,

http://oa.upm.es/1803/1/PFC\_FRANCISCO\_JAVIER\_JIMENEZ\_PATRICIO.pdf).

### ¿Cómo se calculan?

Se calcula tomando la capacidad total de datos que se examinarán en busca de deduplicación y dividiéndola para la capacidad total de datos deduplicados (Whitehouse, 2009, http://searchdatacenter.techtarget.com/es/consejo/Comprender-las-ratios-de-deduplicacion-de-datos-en-los-sistemas-de-copia-de-seguridad). A continuación en la **Figura 2-1** se muestra el proceso para calcular los ratios de deduplicación:



**Figura 2-1.** Proceso para calcular los ratios de deduplicación.

**Realizado por:** MANCHENO, A. 2015.

Los ratios de reducción generalmente se expresan en términos **n:1**, por ejemplo 10:1, 20:1, etc. En algunas ocasiones la reducción se suele representar también en forma de porcentaje, el cual se obtiene a partir de la siguiente fórmula (Jimenez, 2009,

http://oa.upm.es/1803/1/PFC\_FRANCISCO\_JAVIER\_JIMENEZ\_PATRICIO.pdf):

Si se toma el ejemplo de un ratio 10:1, esto quiere decir que la cantidad de espacio que necesita el total de datos deduplicados es 10 veces menor que la cantidad de espacio requerida para almacenar los mismo datos sin deduplicar (Jimenez, 2009,

http://oa.upm.es/1803/1/PFC\_FRANCISCO\_JAVIER\_JIMENEZ\_PATRICIO.pdf).

Cuando los ratios de deduplicación son más grandes implica que los porcentajes de ahorro de capacidad también crecen, sin embargo este crecimiento empieza a ser lento cuando los ratios crecen a partir de un cierto punto. En la **Tabla 1-1** se puede observar cómo se va dando esta conducta (Jimenez, 2009,

http://oa.upm.es/1803/1/PFC\_FRANCISCO\_JAVIER\_JIMENEZ\_PATRICIO.pdf):

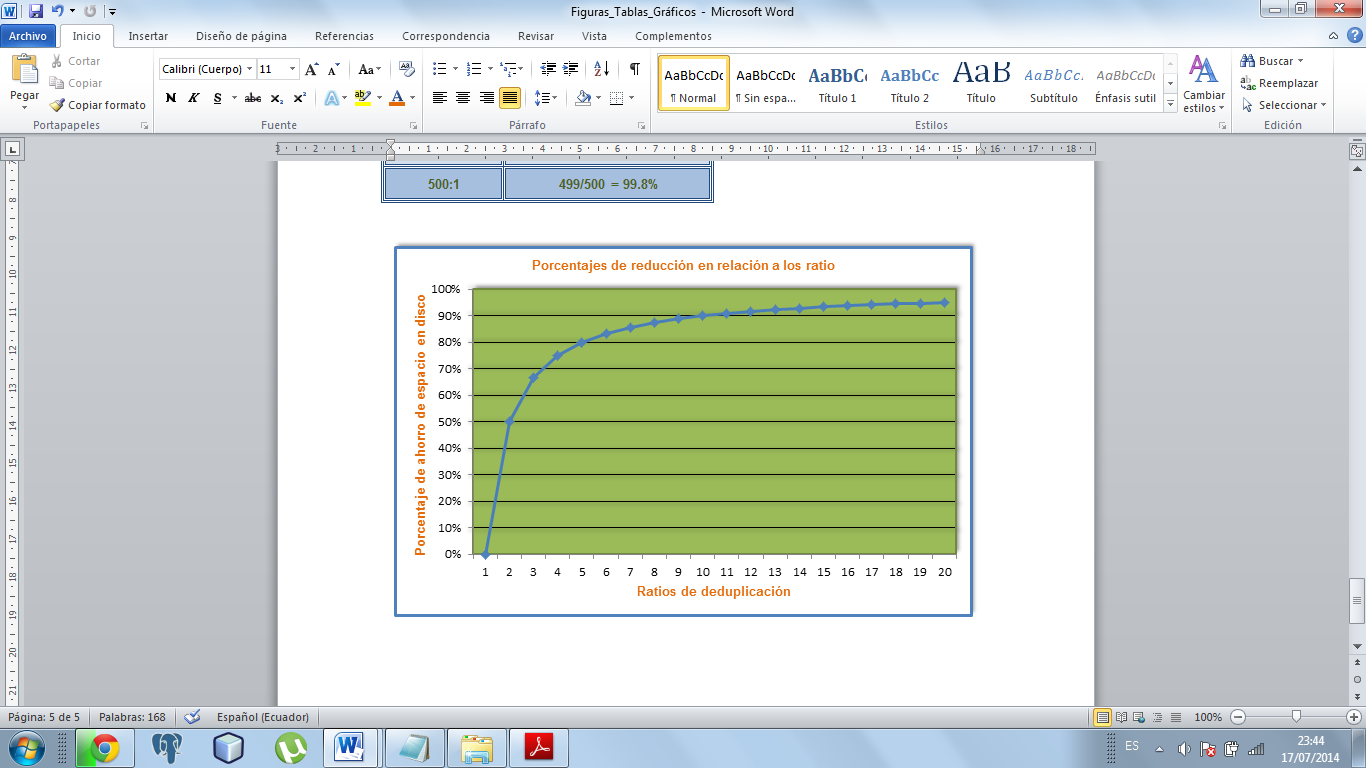
**Tabla 1-1:** Relación de la reducción de ratios y porcentajes.

|  |  |
| --- | --- |
| *ESPACIO DE REDUCCIÓN DE RATIOS Y PORCENTAJES* | |
| *RATIOS* | ***PORCENTAJES = ()*** |
| 2:1 | **1/2 = 50%** |
| 5:1 | **4/5 = 80%** |
| 10:1 | **9/10 = 90%** |
| 20:1 | **19/20 = 95%** |
| 100:1 | **99/100 = 99%** |
| 500:1 | 499/500 = 99.8% |

Fuente: JIMÉNEZ, F. 2009.

Realizado por: MANCHENO, A. 2015.

En la **Tabla 1-1** se puede observar que con un ratio de 10:1 se obtiene un ahorro de espacio del 90% y a partir de ese punto la diferencia entre los porcentajes de reducción empieza a acortarse a medida que el ratio aumenta. En la **Figura 3-1** se puede ver el crecimiento de los porcentajes de reducción en relación a los ratios de deduplicación:



**Figura 3-1.** Porcentajes de reducción en relación a los ratios.

**Fuente:** Jiménez, F. 2009.

Tal como se indicó anteriormente los porcentajes crecen lentamente a medida que los ratios de deduplicación aumentan. En el gráfico se puede comprobar el comportamiento de la curva cada vez que aumentan los ratios, su crecimiento se va dando de forma lenta a partir de ratio mayores a 5 (Jimenez, 2009,

http://oa.upm.es/1803/1/PFC\_FRANCISCO\_JAVIER\_JIMENEZ\_PATRICIO.pdf).

La cantidad total de datos redundantes dentro de un sistema de almacenamiento señala el límite de los ratio de reducción que una solución de deduplicación puede alcanzar (Jimenez, 2009,

http://oa.upm.es/1803/1/PFC\_FRANCISCO\_JAVIER\_JIMENEZ\_PATRICIO.pdf).

### Factores que influyen en los ratios de deduplicación

Los ratios de reducción obtenidos como parte del proceso de búsqueda y comparación de las soluciones de deduplicación pueden variar dependiendo de algunos factores y características del conjunto de datos, como:

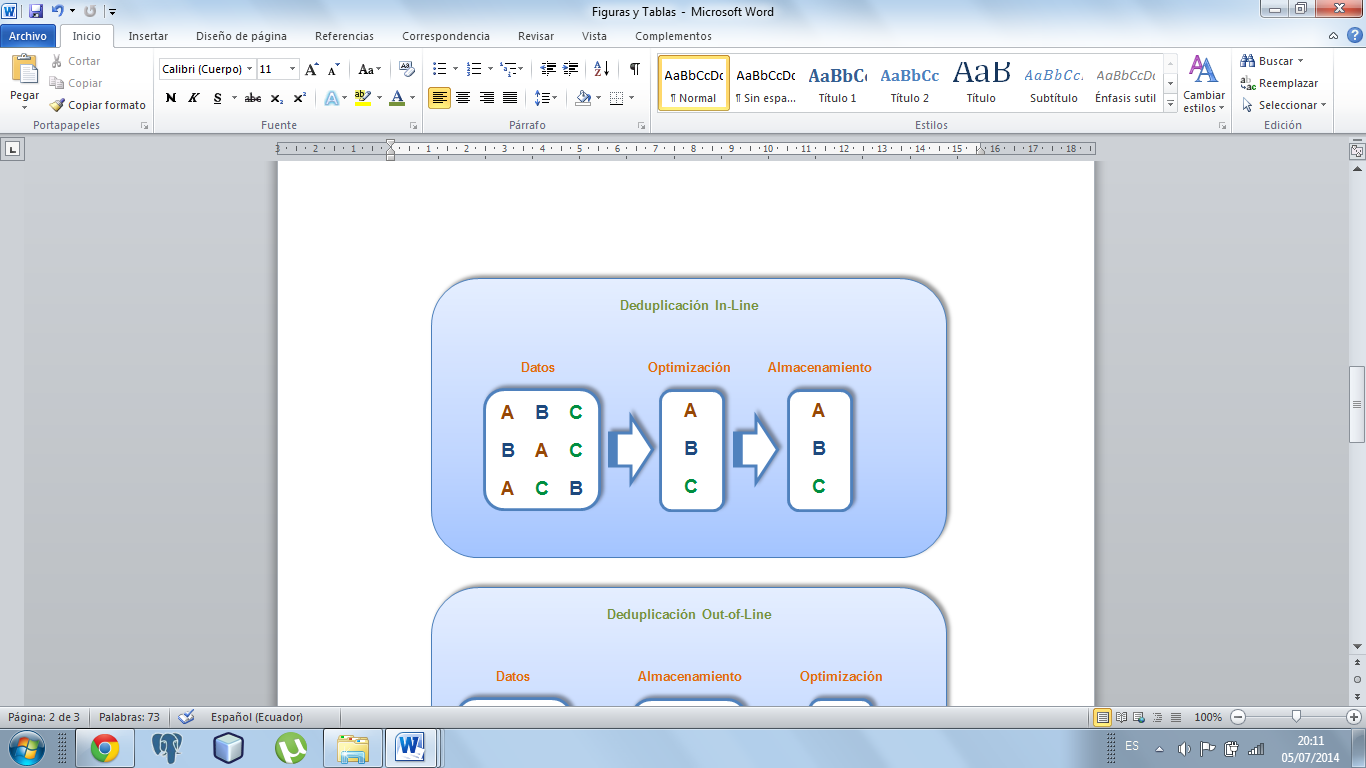
* **Tipo de datos.-** Algunos tipos de datos son más propensos a duplicación (redundancia) que otros.
* **Políticas de backups de datos.-** Mientras mayor sea la frecuencia con la que se realicen backups de los datos, mayor será la probabilidad de encontrar redundancias y por lo tanto se obtendrán ratios de deduplicación mayores. (Hoekstra y Daan, 2012, https://www.os3.nl/\_media/2011-2012/courses/lia/berry\_hoekstra\_daan\_wagenaar\_-\_lia-report-final.pdf).
* **Conservación de datos.-** A mayor tiempo de conservación de datos en el sistema de almacenamiento, mayor será la probabilidad de que el sistema de deduplicación pueda encontrar redundancias en el conjunto de datos. (Hoekstra y Daan, 2012, https://www.os3.nl/\_media/2011-2012/courses/lia/berry\_hoekstra\_daan\_wagenaar\_-\_lia-report-final.pdf).
* **Tasas de variación.-** Mientras menor sea el ratio de variación, existe una mayor posibilidad de encontrar duplicación (redundancia) de datos. (Hoekstra y Daan, 2012, https://www.os3.nl/\_media/2011-2012/courses/lia/berry\_hoekstra\_daan\_wagenaar\_-\_lia-report-final.pdf).
* **Dominios de deduplicación.-** Cuanto más grande sea el espacio del proceso de búsqueda y comparación por parte de la solución de deduplicación, mayor será la posibilidad de detectar datos redundantes. (Hoekstra y Daan, 2012, https://www.os3.nl/\_media/2011-2012/courses/lia/berry\_hoekstra\_daan\_wagenaar\_-\_lia-report-final.pdf).

## Tipos de deduplicación

Dentro de la deduplicación de datos se pueden encontrar dos tipos definidos, la deduplicación en línea o de origen y la deduplicación fuera de línea o destino, conocida también esta última como deduplicación post-process. (Hoekstra y Daan, 2012, https://www.os3.nl/\_media/2011-2012/courses/lia/berry\_hoekstra\_daan\_wagenaar\_-\_lia-report-final.pdf).

### Deduplicación In-line (Origen)

Como se muestra en la **Figura 4-1**, el proceso de optimización se realiza desde el origen, antes de almacenar los datos en el servidor éstos son examinados, si el sistema de deduplicación localiza un bloque que ya haya sido almacenado previamente no almacena el nuevo bloque, el sistema en su lugar crea un puntero al bloque ya existente. La ventaja de este tipo de deduplicación es que requiere menos espacio de almacenamiento, debido a que antes de almacenar los nuevos bloques de datos estos ya son optimizados. La desventaja de esta deduplicación en línea es que el rendimiento del sistema se puede ver afectado si se recibe una gran cantidad de datos por el aumento de los tiempos de cálculo y búsqueda. (Hoekstra y Daan, 2012, https://www.os3.nl/\_media/2011-2012/courses/lia/berry\_hoekstra\_daan\_wagenaar\_-\_lia-report-final.pdf).



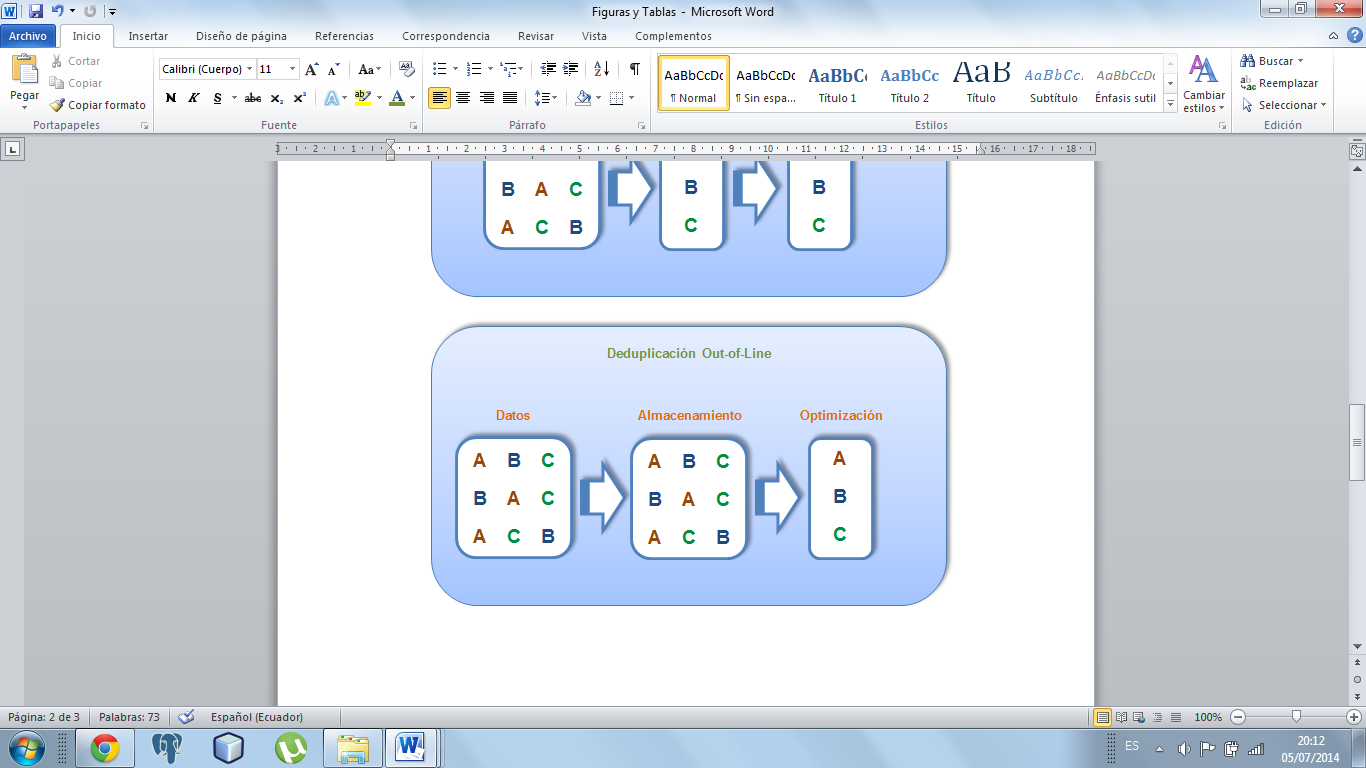
**Figura 4-1.** Deduplicación In-Line.

**Realizado por:** Mancheno, A. 2015.

### Deduplicación Out-of-line (Destino)

Conocida también como deduplicación post-process (posterior al proceso). Los datos son almacenados primeramente en el disco duro, para que en un proceso posterior el sistema de deduplicación los analice en busca de redundancia. Su ventaja radica en que el proceso de deduplicación de los datos antes de su almacenamiento no es realizado en línea, asegurando de esta manera que el rendimiento del sistema no se vea afectado. Por el contrario el principal inconveniente de éste tipo de deduplicación es que requiere de una mayor cantidad de espacio disponible para poder almacenar los datos iniciales antes de que sean deduplicados, lo cual es un problema si el sistema de almacenamiento está al límite de su capacidad máxima (Hoekstra y Daan, 2012, https://www.os3.nl/\_media/2011-2012/courses/lia/berry\_hoekstra\_daan\_wagenaar\_-\_lia-report-final.pdf).

La **Figura 5-1** muestra el funcionamiento de la misma.



**Figura 5-1.** Deduplicación Out-of-Line.

**Realizado por:** MANCHENO, A. 2015.

## 

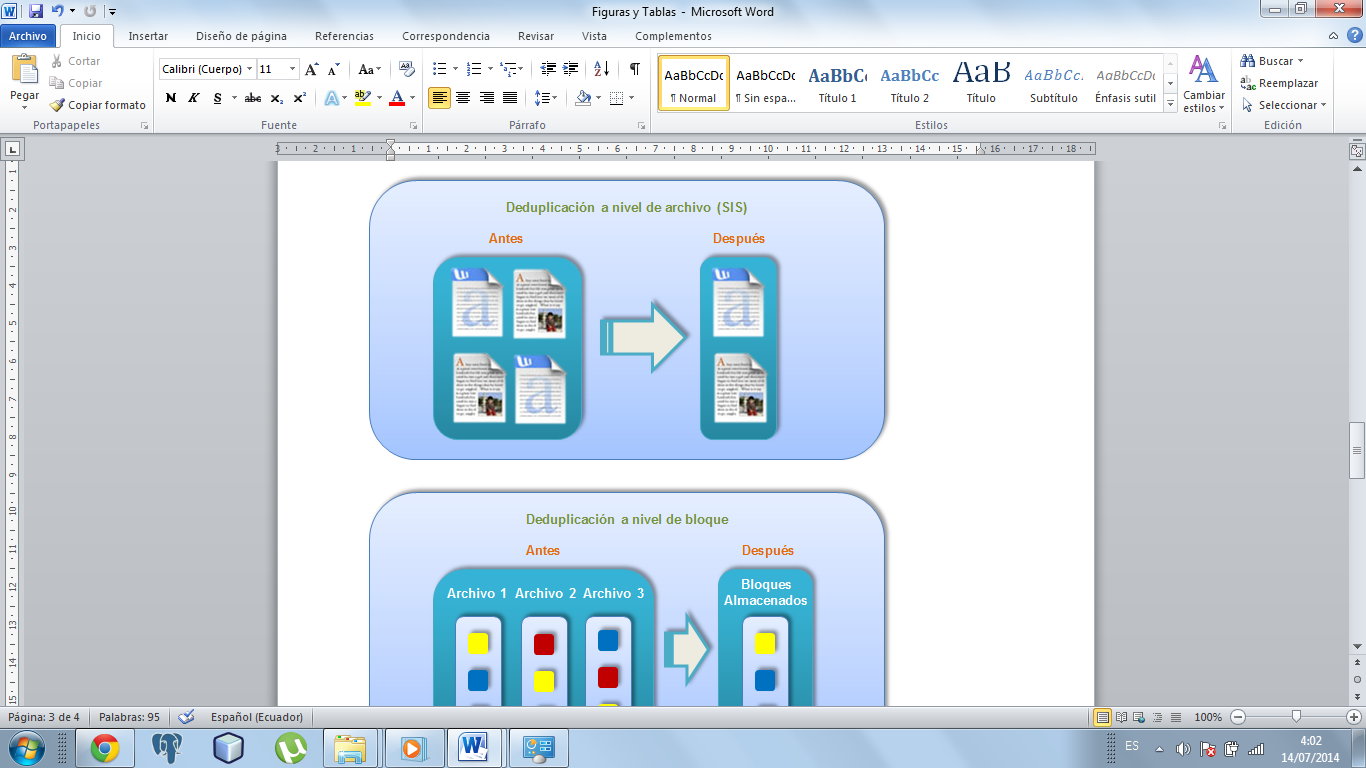
## Niveles de granularidad

El nivel de granularidad es una consideración importante a tener en cuenta, los sistemas de deduplicación proporcionan tres diferentes niveles de granularidad: nivel de archivo, nivel de bloques y nivel de byte. El grado de eficiencia que la solución de deduplicación ofrezca dependerá del nivel de granularidad que emplee, a mayor nivel de granularidad mejor eficiencia en el proceso de deduplicación. (LORTU, 2013, http://www.lortu.es/Technology.aspx).

### Deduplicación a nivel de archivo

La deduplicación a nivel de archivo llamada también “almacenamiento de instancia única” (SIS), fue la primera en utilizar **hash** o **funciones de resumen** para comparar los contenidos entre archivos. La función hash genera un identificador único (Número de Hash) por cada fragmento de datos analizado, éste identificador se almacena en un índice para luego ser comparado con otros identificadores en búsqueda de redundancia. Si los fragmentos analizados de dos archivos tienen el mismo número de hash, se dice que los archivos son iguales. La ventaja de SIS es que requiere menos potencia de procesamiento ya que los números de hash de los ficheros son más fáciles de generar, resultado ser rápido y sencillo. Su principal desventaja radica en que si se cambia un solo byte del archivo, el número de hash también cambia, por lo que ambas versiones del archivo deberán guardarse nuevamente por separado. (StarWind Software, 2009, http://www.starwindsoftware.com/file-level-vs-block-level-vs-byte-level-deduplication).

Los resultados de este tipo de deduplicación se muestran en la **Figura 6-1.**



**Figura 6-1.** Deduplicación a nivel de archivo (SIS).

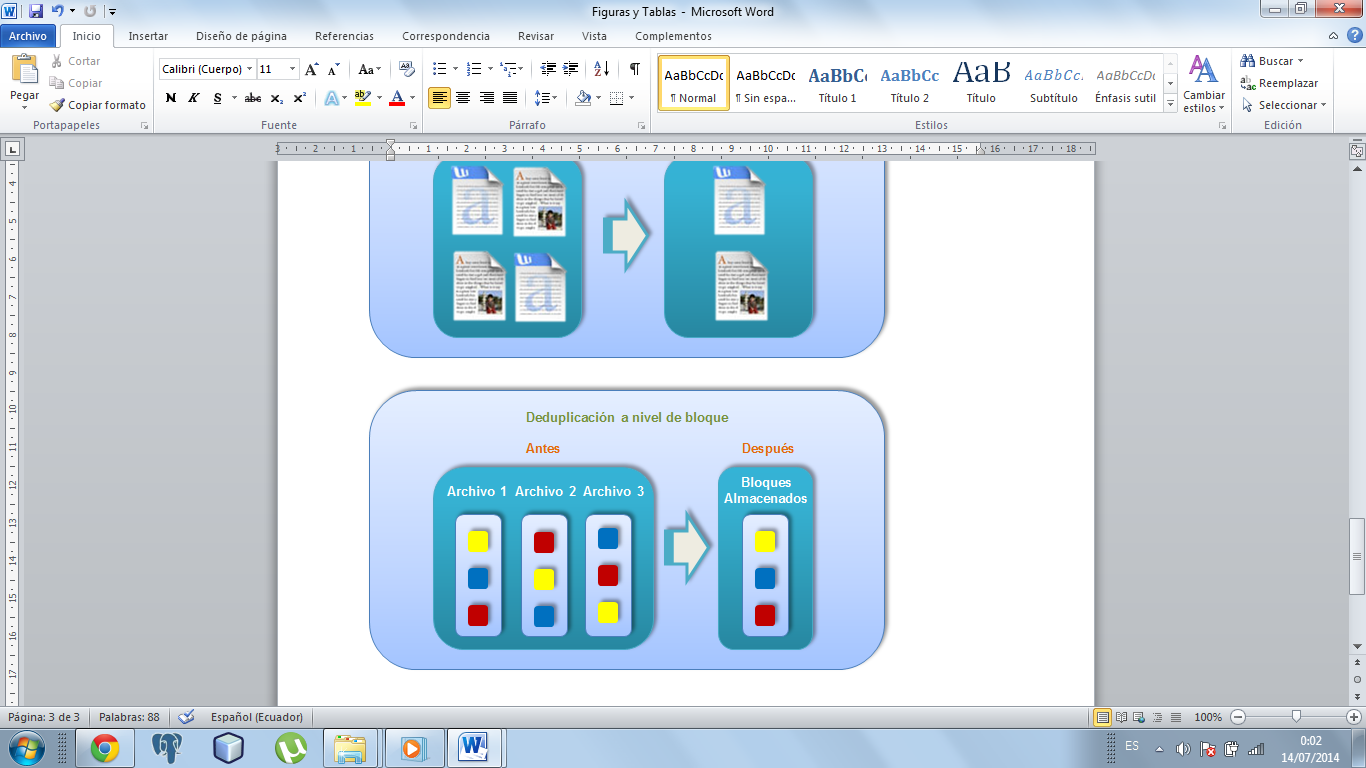
**Realizado por:** MANCHENO, A. 2015.

### Deduplicación a nivel de bloque

Divide los datos en bloques, almacenando una única copia de cada bloque repetido. A cada bloque se le asigna un identificador único, el cual se almacena en un índice para luego ser comparado con los demás identificadores en búsqueda de redundancia. Mientras más pequeños sean los bloques, mayor será el número de comparaciones con el índice, aumentando el potencial para identificar y eliminar redundancias, alcanzando de esta manera mayores tasas de reducción con respecto a la deduplicación a nivel de archivo (SIS) (Whitehouse, 2009, http://searchdatacenter.techtarget.com/es/consejo/Cuando-y-como-utilizar-la-tecnologia-de-deduplicacion-de-datos-en-el-safeguard-de-disco). A diferencia de SIS, en la deduplicación a nivel de bloque, si se hacen pequeños cambios en el contenido de un archivo de gran tamaño no almacena ambas versiones del archivo, por el contrario el sistema sólo almacena los fragmentos modificados. (StarWind Software, 2009, http://www.starwindsoftware.com/file-level-vs-block-level-vs-byte-level-deduplication).

La deduplicación a nivel de bloque requiere de una mayor capacidad de procesamiento que SIS, dado que el número de identificadores que debe procesar es mayor dependiendo del número de bloques en que se han dividido los datos. Esto implicaría que el rendimiento se vea afectado, porque para recuperar los datos es necesario volver a ensamblar nuevamente los bloques, lo cual se traduce en una desventaja. (StarWind Software, 2009, http://www.starwindsoftware.com/file-level-vs-block-level-vs-byte-level-deduplication).

Los resultados de este tipo de deduplicación se muestran en la **Figura 7-1.**



**Figura 7-1.** Deduplicación a nivel de bloque.

**Realizado por:** MANCHENO, A. 2015.

En ocasiones el mismo número de hash (identificador) se puede generar por dos bloques de datos diferentes, dando lugar a una **colisión hash**. Si esto ocurre, el sistema no almacenará los nuevos datos al comprobar que el número de hash ya existe. Esto deriva en otro inconveniente dado que las colisiones hash, a pesar de que su incidencia es reducida, provocan un falso positivo dando lugar a la corrupción de datos. (StarWind Software, 2009, http://www.starwindsoftware.com/file-level-vs-block-level-vs-byte-level-deduplication).

La deduplicación a nivel de bloque puede ser de dos tipos, bloques de **longitud fija** y bloques de **longitud variable.**

#### Bloques de longitud fija

La deduplicación de bloques de longitud fija divide los datos en bloques de tamaños fijo, que van normalmente desde los 4 KB hasta los 64 KB. El principal inconveniente de esta deduplicación es la falta de capacidad en la gestión de archivos cuando los datos son agregados con anterioridad o se insertan en un archivo conocido, y el sistema de deduplicación emplea los mismos bloques fijos para nuevas comparaciones, siendo probable que no detecte bloques redundantes dado que éstos han cambiado o se han desplazado. (Western Digital Technologies, 2013, http://www.wdc.com/wdproducts/library/Flyer/ESN/2579-800022.pdf).

#### Bloques de longitud variable

La deduplicación de bloques de longitud variable emplea secuencias de bytes predeterminadas en un archivo para establecer los límites de los bloques. A diferencia de la deduplicación de bloques de longitud fija, puede detectar cambios en los bloques cuando se modifica un archivo aumentando las posibilidades de encontrar redundancia entre los bloques. Las secuencias de bytes que establecen los límites de los bloques cambian a medida que se modifica un archivo. (Western Digital Technologies, 2013, http://www.wdc.com/wdproducts/library/Flyer/ESN/2579-800022.pdf).

Este enfoque de deduplicación a nivel de bloque requiere de una mayor capacidad de procesamiento que el otro enfoque, debido a que se tiene que analizar el archivo completo, byte a byte, para definir los límites de los bloques. (Western Digital Technologies, 2013, http://www.wdc.com/wdproducts/library/Flyer/ESN/2579-800022.pdf).

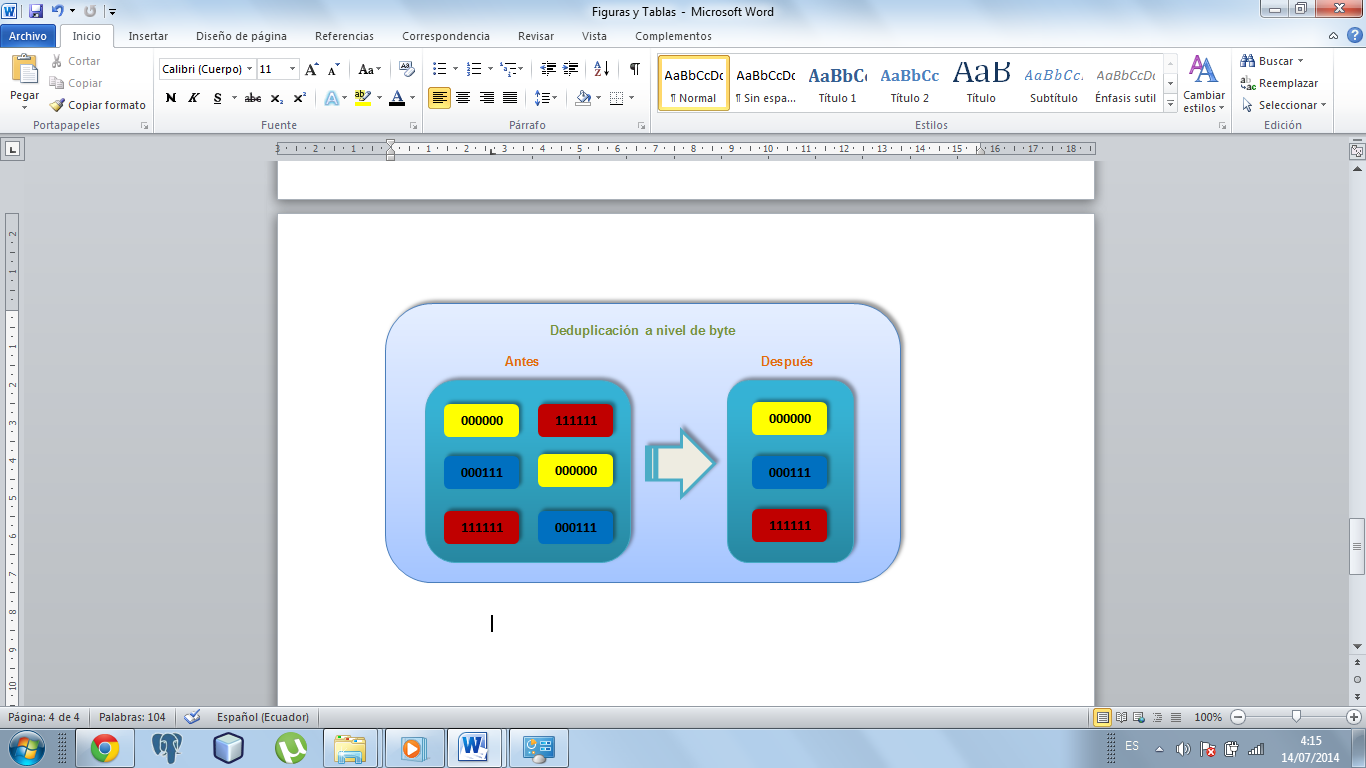
La deduplicación de bloques de longitud variable no permite el control del tamaño de los bloques, dado que la distribución de los bloques es muy variada a pesar que se puede definir un tamaño medio. El tamaño de los bloques varía entre 1 byte y la longitud total del archivo, generando un inconveniente a la hora de almacenar los bloques ya que se requiere de un manejo individual de los bloques de tamaño erróneo. (Western Digital Technologies, 2013, http://www.wdc.com/wdproducts/library/Flyer/ESN/2579-800022.pdf).

Un tamaño de bloque más grande requiere de menos recursos, sin embargo los niveles de comprensión que proporciona son muy bajos. En cambio un tamaño de bloque más pequeño proporciona mejores niveles de comprensión, pero necesita de una mayor cantidad de recursos. (StarWind Software, 2009, http://www.starwindsoftware.com/file-level-vs-block-level-vs-byte-level-deduplication).

### Deduplicación a nivel de byte

La deduplicación a nivel de byte permite un mayor grado de granularidad que los dos niveles anteriores. Los datos nuevos se comparan byte a byte con los datos almacenados previamente. Este método permite obtener un mayor nivel de precisión, casi que garantizando la eliminación total de redundancia (Wendt, 2009, http://searchdatacenter.techtarget.com/es/cronica/Preguntas-frecuentes-sobre-la-deduplicacion-de-datos), pero en contraste su rendimiento se ve afectado, debido a que se aplica a la deduplicación Out-line (posterior al proceso), es decir que los datos son almacenados primeramente, por lo que se requiere de una reserva de caché del disco antes de realizar el proceso de comparación. (Whitehouse, 2009, http://searchdatacenter.techtarget.com/es/consejo/Metodos-de-deduplicacion-de-datos-por-bloques-o-por-bytes).

Los resultados de este tipo de deduplicación se muestran en la **Figura 8-1.**



**Figura 8-1.** Deduplicación a nivel de byte.

**Fuente:** MANCHENO, A. 2015.

## Opendedup (SDFS)

Opendedup (SDFS) es un sistema de ficheros que permite deduplicación en línea. Es un sistema que permite un ahorro de más del 90% del espacio utilizado, elimina archivos redundantes de más de un Petabyte, su rendimiento supera 1Gbps, entre otros. (Cloudadmins.org, 2014, http://www.cloudadmins.org/opendedup-deduplicacion-en-linea/).

Las principales características de Opendedup son:

* 1. Deduplicación Variable y Fija, deduplica datos estructurados y no estructurados.
  2. Reducción de Utilización de almacenamiento - La deduplicación puede reducir la utilización del almacenamiento hasta en un 90% -95%
  3. Escalabilidad - Opendedup puede deduplicar enormes cantidades de datos a través de hasta 126 nodos independientes
  4. Alta disponibilidad - Todos los datos de bloque puede se replica a través de hasta 7 nodos
  5. Replicación - deduplicación eficiente replicación cuenta
  6. Apoyo VMWare - Trabaja con VMS - puede deduplicar en 4k bloque de tamaño. Esto es necesario para deduplicar máquinas virtuales con eficacia
  7. Almacenamiento flexibles - los datos deduplicados pueden ser almacenados localmente, en la red a través de múltiples nodos, o en la nube.
  8. Global Inline deduplicación - comparte datos Opendedup únicas entre volúmenes en tiempo real.
  9. Configuración independiente o agrupada. (Opendedup.org, 2015, opendedup.org).

## ZFS

ZFS es un sistema de archivos creado por Sun Microsystems y actualmente pertenece a Oracle. Una de las características principales de ZFS es que permite deduplicación en línea a nivel de bloques, identificando bloques de datos redundantes y suprimiéndolos, optimizando de esta manera el espacio necesario en disco para almacenar dichos datos. (Mugarza, 2012, http://www.mikelmugarza.com/?p=219).

Como cualquier otra tecnología, la deduplicación con ZFS tiene pros y contras que serán expuestos.

### Ventajas

ZFS descartará cualquier bloque de datos que es idéntico a un bloque ya escrito, mientras se mantiene una referencia para que siempre pueda reproducir el mismo bloque cuando se lee. (Cloudadmins.org, 2014, http://www.cloudadmins.org/opendedup-deduplicacion-en-linea/).

Como buena práctica se recomienda revisar cual será el espacio aproximado que se obtendrá de la implementación del sistema deduplicado:

* Probarlo con algunos datos reales. Esta es la opción más precisa y directa, se debe tomar una cantidad representativa de datos y colocarla en el zpool, luego evaluar visualmente la cantidad de ahorro producida en este proceso. (Cloudadmins.org, 2014, http://www.cloudadmins.org/opendedup-deduplicacion-en-linea/).
* Con el comando *zdb –S* se puede obtener una estimación o simulación, menos precisa que el uso de datos reales, pero bastante aproximada. (Cloudadmins.org, 2014, http://www.cloudadmins.org/opendedup-deduplicacion-en-linea/).
* Usualmente encontrar datos reales de prueba no es un proceso sencillo, por ejemplo si el servidor va orientado al almacenamiento de máquinas virtuales, se deberían evaluar preguntas como ¿Cuántas máquinas son compatibles? ¿Con qué frecuencia están parcheados? ¿Qué tan probable es que las personas apliquen el mismo software / parches / datos a sus máquinas? ¿Cuántos GB de datos, es probable que esto genere? ¿Puede llegar a un caso de prueba representativo, después de todo para hacer la conjetura más apropiada?. (Cloudadmins.org, 2014, http://www.cloudadmins.org/opendedup-deduplicacion-en-linea/).

En cualquier caso, se obtendrá un valor de GB recuperables por cada GB de información, es decir el ahorro que se va a conseguir. Este número puede tener cualquier valor: Algunas personas ven un valor de 1,00 (sin duplicación en absoluto), otros ven algunos ahorros moderados como 1.5 lo que significa que en cada 2GB de espacio podrían almacenarse 3GB de información, y en algunos casos se puede llegar a tener tanto como 20 veces, por ejemplo, un servidor de almacenamiento de virtualización con un perfil de uso muy repetitivo. (Cloudadmins.org, 2014, http://www.cloudadmins.org/opendedup-deduplicacion-en-linea/).

El ahorro de almacenamiento esperado viene dado por la siguiente ecuación.

***Almacenamiento Total - (total de almacenamiento / relación de deduplicación esperado) = Se esperaba Ahorro Almacenamiento***

Como un ejemplo ficticio, se tiene una agrupación de almacenamiento de 10 TB a ser utilizado para el almacenamiento de imágenes de máquinas virtuales en un escenario de escritorio virtual. En una prueba rápida, se crea un pool de 1 TB y se copian algunos datos de VM existentes a la misma, lo que produjo una relación dedup de 2. Esto significa que sólo se necesitan unos 5 TB de capacidad para almacenar los 10 TB de datos gracias a la deduplicación, por lo tanto, se ahorraría 5 TB de almacenamiento en disco. (Cloudadmins.org, 2014, http://www.cloudadmins.org/opendedup-deduplicacion-en-linea/).

Se supone que 1TB de disco tiene un costo de 100 dólares, por lo cual el ahorro por la tecnología de deduplicación es 500 dólares en este caso. (Cloudadmins.org, 2014, http://www.cloudadmins.org/opendedup-deduplicacion-en-linea/).

### ¿Cuál es el Costo?

El ahorro de espacio gracias a la deduplicación no viene de forma gratuita. Hay un costo. En el caso de ZFS, es la memoria: ZFS mantiene una tabla dedup en la que almacena todas las sumas de comprobación de todos los bloques que fueron escritos después que la deduplicación se habilitó. Al escribir nuevos bloques, utiliza esta tabla para determinar si un bloque ha sido o no escrito todavía. (Gonzalez, 2011, http://constantin.glez.de/blog/2011/07/zfs-dedupe-or-not-dedupe).

Con el tiempo, la mesa se hace más grande y más grande, y ya que cada operación de escritura tiene que usarlo, se debe mantener en la memoria principal para evitar la innecesaria lectura desde el disco. Para que quede claro: ZFS pueden trabajar perfectamente bien, incluso si la tabla no está en la memoria. Pero cuanto más grande es la tabla de deduplicación, el rendimiento de escritura será más lento. (Gonzalez, 2011, http://constantin.glez.de/blog/2011/07/zfs-dedupe-or-not-dedupe).

### ¿Cuánta memoria se necesitan para mantener la tabla dedup ZFS?

De acuerdo con la ZFS dedup FAQ, cada entrada en la tabla dedup cuesta unos 320 bytes de memoria por bloque. Para estimar el tamaño de la tabla dedup, se necesita saber cuántos bloques ZFS se necesitarán para almacenar los datos. Esta pregunta puede ser complicado: ZFS utiliza un tamaño de bloque variable entre 512 y 128 K bytes, dependiendo del tamaño de los ficheros que almacena. Así que en realidad no se puede saber de antemano cuántos bloques ZFS utilizará para almacenar los datos. (Gonzalez, 2011, http://constantin.glez.de/blog/2011/07/zfs-dedupe-or-not-dedupe).

Si se almacenan principalmente archivos de gran tamaño (vídeos, fotos, etc.), a continuación, el tamaño medio de bloques estarán más cerca de 128K, si se almacenan archivos pequeños (código fuente, correos electrónicos, otros datos), probablemente se va a estar más cerca de unos pocos K. (Gonzalez, 2011, http://constantin.glez.de/blog/2011/07/zfs-dedupe-or-not-dedupe).

## Sistema operativo Linux

### Origen

Linux hace su aparición a principios de la década de los noventa, era el año 1991 y por aquel entonces un estudiante de informática de la Universidad de Helsinki, llamado *Linus Torvalds* empezó, -como una afición y sin poder imaginar a lo que llegaría este proyecto, a programar las primeras líneas de código de este sistema operativo al que llamaría más tarde Linux. (Martinez, 2014, http://www.linux-es.org/sobre\_linux)

Este comienzo estuvo inspirado en MINIX, un pequeño sistema Unix desarrollado por Andy Tanenbaum. Las primeras discusiones sobre Linux fueron en el grupo de noticias comp.os.minix, en estas discusiones se hablaba sobre todo del desarrollo de un pequeño sistema Unix para usuarios de Minix que querían más. (Martinez, 2014, http://www.linux-es.org/sobre\_linux)

El 3 de julio de 1991, Linus Torvalds mandó el primer mensaje sobre Linux al grupo de noticias comp.os.minix (Martinez, 2014, http://www.linux-es.org/sobre\_linux), mismo que se puede ver en el Anexo 4.

Linus Torvalds nunca anunció la versión 0.01 de Linux (agosto/septiembre 1991), esta versión no era ni siquiera ejecutable, solamente incluía los principios del núcleo del sistema, estaba escrita en lenguaje ensamblador y asumía que uno tenía acceso a un sistema Minix para su compilación. (Martinez, 2014, http://www.linux-es.org/sobre\_linux)

El 5 de octubre de 1991, Linus anuncio la primera versión "Oficial" de Linux, -versión 0.02. Con esta versión Linus pudo ejecutar Bash (GNU Bourne Again Shell) y gcc (El compilador GNU de C) pero no mucho más funcionaba. (Martinez, 2014, http://www.linux-es.org/sobre\_linux)

La versión 1.0.0 llegó el 4 de Enero de 1993 y era ya capaz de ejecutar el sistema X-windows.

El 9 de Mayo 1996, TUX fue propuesto como mascota oficial de Linux. (Martinez, 2014, http://www.linux-es.org/sobre\_linux)

El 9 de junio de 1996 fue lanzada la serie 2.0.x, la 2.2.x no llegó hasta el 25 de enero de 1999 y la 2.4.x hasta el 4 de enero del 2001. (Martinez, 2014, http://www.linux-es.org/sobre\_linux)

Actualmente Linux y sus diferentes distribuciones siguen en constante desarrollo y evolución bajo aportes de la comunidad y una gran cantidad de desarrolladores activos con Linus Torvalds incluido aun en el proyecto. (Martinez, 2014, http://www.linux-es.org/sobre\_linux)

En cuanto a su principal contendiente Windows, los principales puntos a tener en cuenta son La instalación, compatibilidad, software, robustez. (Villanueva, 2007, http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html)

Estos campos son mencionados en el análisis que se hace en rinconsolidario.org que concluye en Tanto Windows como Linux tienen sus ventajas e inconvenientes, aunque desde un punto de vista más técnico Linux sale ganando. (Villanueva, 2007, http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html)

Hoy en día [Linux](http://bitelia.com/tag/linux) está presente en un sin fin de dispositivos a nuestro alrededor, desde *smartphones* hasta consolas de videojuegos, pero está incluso más cerca de lo que pensamos. (Villanueva, 2007, http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html)

Aunque el uso en equipos domésticos y personales no es tan amplio como se espera, todo aquel que accede a alguna página de internet ya está utilizando Linux, dado que es el sistema operativo más utilizado en servidores. (Villanueva, 2007, http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html)

* **Estabilidad**

Nadie duda de la estabilidad de un sistema Linux, más cuando es bien sabido que **una instalación típica puede correr durante años sin presentar fallas**. Linux maneja grandes cantidades de procesos mucho mejor que Windows y no requiere reiniciar ante los cambios de configuración o luego de actualizaciones del sistema. Windows se degrada con el tiempo y de cuando en cuando requerirá intervención del usuario para desfragmentar el disco duro, limpiar el registro o aplicar cualquier otro artilugio que permita devolverle un poco de agilidad al sistema. En Linux, con un buen sistema de archivos (como Ext4 o XFS) jamás oirás la palabra desfragmentar. (Villanueva, 2007, http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html)

* **Seguridad**

Ningún sistema es inmune a los ataques, pero la velocidad de reacción ante un fallo y la cantidad de personas que pueden auditarlo, sin duda marcan la diferencia. **Microsoft reacciona de forma más lenta ante los agujeros de seguridad de Windows** y esto deja a sus sistemas vulnerables durante más tiempo. La comunidad de Linux responde mucho más rápido ante estos eventos y gracias a que el código fuente está disponible, cualquiera con los conocimientos necesarios puede corregir el fallo y ponerlo a disposición de los usuarios de inmediato. No está de más decir que Linux es atacado con mucha menos frecuencia por virus y *malware*. (Villanueva, 2007, http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html)

* **Manejo de aplicaciones**

Las distribuciones Linux usan repositorios oficiales para sus aplicaciones, aplicando controles de calidad y políticas de seguridad que garantizan la integridad y sanidad de los paquetes instalados. Y no sólo se trata de seguridad e integridad, también es un tema de facilidad de uso pues **todo lo que podamos necesitar se encuentra en un sólo lugar**. (Villanueva, 2007, http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html)

* **Flexibilidad**

**Una instalación Linux puede ajustarse tanto como sea necesario**. Es común no instalar interfaz gráfica para evitar el desperdicio de memoria RAM, escoger un sistema de archivos óptimo según las necesidades del servicio o compilar una aplicación si queremos adaptarla a la medida de nuestro sistema. Linux puede reconfigurarse para incluir sólo los servicios esenciales para el tipo de negocio y así optimizar el uso de los recursos. (Villanueva, 2007, http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html)

* **Costos**

Linux es el rey indiscutible en costos de implementación y difícilmente podrá ser derrocado. Incluso las versiones empresariales con soporte corporativo son más económicas que Windows o cualquier otro *software* propietario. Esto se debe a que **la mayoría del *software* que se usa en Linux es gratuito** mientras que el propietario está atado a costosos esquemas de licenciamiento o a ventas de complementos adicionales para agregar nuevas funcionalidades. (Villanueva, 2007, http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html)

* **Comunidad**

**Linux es y siempre ha sido algo comunitario**, esto hace que los desarrolladores siempre estén escuchando que dicen y que necesitan sus usuarios. Este esfuerzo mancomunado permite que la mayoría de las distribuciones puedan publicar actualizaciones cada 6 meses, cosas que ocurre con mucha menos frecuencia del lado de Windows. (Villanueva, 2007, http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html)

* **Libertad**

Quizás la más filosófica de todas las ventajas, pero es que con Linux no tendrás detrás de ti a ninguna compañía limitándote en el uso del sistema o encerrándote bajo cláusulas legales oscuras. **Con Linux eres libre de usar, modificar y combinar lo que desees para satisfacer tus necesidades**.

Entre las distribuciones más usadas para servidores están Debian, Slackware, CentOS y Red Hat Enterprise Linux. (Hipertextual.com, 2014, http://hipertextual.com/archivo/2014/05/linux-servidores/?utm\_content=buffer9884c&utm\_medium=social&utm\_source=twitter.com&utm\_campaign=buffer)

## Métodos de almacenamiento de deduplicación

### Deduplicación de longitud variable

Es un algoritmo de deduplicación de datos que desglosa un sistema de archivos en segmentos de datos de longitud variable de subarchivo con el fin de determinar segmentos exclusivos y repetitivos. Esto reduce dramáticamente el almacenamiento de respaldo durante el proceso de respaldo y recuperación. (EMC, 2015, http://venezuela.emc.com/corporate/glossary/variable-length-deduplication.htm)

La deduplicación de longitud variable es un método avanzado de desglose de un flujo de datos a través de puntos de apoyo sensibles al contexto. Este método de segmentación inteligente de subarchivos proporciona mayor eficiencia de almacenamiento para datos redundantes, independientemente de donde se hayan insertado los nuevos datos. Como el nombre sugiere, la longitud de segmentos varía, con lo que se consiguen mayores proporciones de deduplicación. (EMC, 2015, http://venezuela.emc.com/corporate/glossary/variable-length-deduplication.htm)

#### ¿Quién elige la deduplicación de longitud variable, y por qué?

Organizaciones con un rápido crecimiento de datos, ambientes altamente virtualizados y oficinas remotas se benefician en gran medida con la deduplicación de longitud variable por encima de un enfoque de bloques fijos. La deduplicación de longitud variable reduce el almacenamiento de respaldo y, cuando se realiza en el cliente, reduce también el tráfico de red, lo que lo hace ideal para el respaldo remoto. (EMC, 2015, http://venezuela.emc.com/corporate/glossary/variable-length-deduplication.htm)

#### ¿Cómo funciona la deduplicación de longitud variable?

El software cliente examina el sistema de archivos y aplica un algoritmo hash seguro SHA-1 a segmentos de datos de longitud variable. A cada segmento de datos se le asigna un identificador (ID) único. A continuación, el software cliente determina si este ID único ya se ha almacenado. Si este objeto existe, se hace referencia a un enlace al objeto almacenado en el respaldo. De esta forma, nunca se respalda dos veces el mismo segmento. (EMC, 2015, http://venezuela.emc.com/corporate/glossary/variable-length-deduplication.htm)

#### Beneficios de la deduplicación de longitud variable

Al proporcionar el máximo nivel de deduplicación, la deduplicación de longitud variable reduce el almacenamiento de respaldo, aumenta los tiempos de ejecución de respaldo y disminuye los costos. Cuando se implementa en el cliente, permite a las organizaciones aprovechar el ancho de banda existente, y se reduce, a la vez, la contención de recursos en entornos altamente virtualizados. (EMC, 2014, http://mexico.emc.com/corporate/glossary/variable-length-deduplication.htm).

### Deduplicación de longitud fija

Para determinar segmentos únicos repetitivos, [la deduplicación de longitud fija](http://mexico.emc.com/corporate/glossary/fixed-length-deduplication.htm) es un algoritmo de deduplicación de datos que desglosa un sistema de archivos en segmentos de datos de longitud fija de subarchivo. (EMC, 2015, http://venezuela.emc.com/corporate/glossary/variable-length-deduplication.htm)

#### Limitaciones

La principal limitación de este enfoque es que, cuando los datos en un archivo se cambian de posición, por ejemplo, cuando se agrega una diapositiva a una presentación de Microsoft PowerPoint, todos los bloques siguientes en el archivo se reescriben, y es probable que se consideren distintos a los que estaban en el archivo original. Como resultado, el efecto de compresión es menos significativo. Los bloques más pequeños consiguen una mejor deduplicación que los grandes, pero la deduplicación precisa más procesamiento.

La deduplicación de longitud variable es un enfoque más avanzado, que fija segmentos de longitud variable según sus patrones de datos interiores. Esto resuelve el problema de cambio de posición de los datos del enfoque de bloques de tamaño fijo. (EMC, 2015, http://venezuela.emc.com/corporate/glossary/variable-length-deduplication.htm)

## Disponibilidad de Deduplicación

Actualmente con el auge del uso de internet tenemos dos grandes grupos en los que se ha dividido la información, información online (Inline) e información offline.

De la misma manera, al ser la deduplicación una tecnología orientada al almacenamiento de datos, está sujeta a estas limitaciones y características.

Entendiéndose por deduplicación offline, a ser él mismo capaz de identificar, preparar, deduplicar y manejar los datos existentes, no así a ser utilizado por una aplicación para almacenarlos, pues esta consta como un servidor instalado localmente, por ejemplo Bacula.

## Tablas Hash

Las tablas hash son uno de los mecanismos más utilizados en el desarrollo de aplicaciones. Existen múltiples librerías en casi todos los lenguajes de programación que proporcionan implementaciones muy eficientes de estas tablas (por ejemplo la clase java.util.Hashtable de las librerías del lenguaje Java). (UC3M, 2014,

http://www.it.uc3m.es/labas/course\_notes/dynamic\_data\_structures\_hash\_tables\_what\_are\_they\_es.html)

La implementación de una tabla hash está basada en los siguientes elementos:

* Una tabla de un tamaño razonable para almacenar los pares (clave, valor).
* Una función “hash” que recibe la clave y devuelve un índice para acceder a una posición de la tabla.
* Un procedimiento para tratar los casos en los que la función anterior devuelve el mismo índice para dos claves distintas. Esta situación se conoce con el nombre de colisión.
* Las posibles implementaciones de cada uno de estos tres elementos se traducen en una infinidad de formas de implementar una tabla hash. (UC3M, 2014, http://www.it.uc3m.es/labas/course\_notes/dynamic\_data\_structures\_hash\_tables\_what\_are\_they\_es.html)

## Algoritmo de Compresión LZ4.

LZ4 es un algoritmo que se centra en la compresión y descompresión de alta velocidad. Tiene una relación de compresión ligeramente menor que el algoritmo LZO, que a su vez es menor que algoritmos gzip. Sin embargo, las velocidades de compresión son similares a LZO, y varias veces más mayor que otros algoritmos, mientras que la velocidad de descompresión puede ser hasta tres veces mayor que la de LZO. (UC3M, 2014, http://www.it.uc3m.es/labas/course\_notes/dynamic\_data\_structures\_hash\_tables\_what\_are\_they\_es.html)

Actualmente también esta implementado de forma nativa, a partir de la versión 3.11, en el kernel Linux. La implementación en BSD del sistema de archivos ZFS admite el algoritmo de compresión sobre la marcha LZ4.

Comparado con el algoritmo de compresión por defecto LZJB usado por ZFS, LZ4 es un 50% más rápido cuando comprime datos comprimibles y como tres veces más rápido cuando intenta comprimir datos no comprimibles. El rendimiento en datos no comprimibles es una gran mejora, esto viene de una característica de ‘abortó temprano’, si ZFS detecta que los ahorros de compresión es menos del 12.5% entonces la compresión aborta y los bloques se escriben sin compresión (especialmente útil para archivos grandes multimedia que ya están comprimidos). (FreeBSD, 2013, http://freebsd.mx/freebsd-9-2-caracteristica-sobresaliente-zfs-con-compresion-lz4).

# CAPITULO II

**2. MARCO METODOLÓGICO**

## Introducción

Este capítulo se centrará en el establecimiento de los parámetros básicos que permitirán la elección de un sistema de archivos adecuado que permita la futura implementación y correcto funcionamiento de un servidor de backups optimizando recursos como espacio en disco o uso de datos en internet.

Estos parámetros se establecen en base al estudio de los temas precedentes como son:

* Definición de la arquitectura orientada a la interacción de sistemas de archivos diferentes en el mismo equipo.
* Definición del servidor a utilizar para optimizar el proceso de extracción de backups.
* Definición de los lineamientos propuestos para la implementación de un servidor en Linux.

Se ha establecido para el desarrollo de la presente investigación, pese a no ser la única manera de hacerlo, la implementación de un entorno Linux sobre el sistema de archivos ZFS, dentro del cual se separa un volumen en SDFS, en los cuales se almacenarán los mismos backups automáticamente generados por Bacula para medir sus propiedades.

## Sistemas de archivo deduplicados

Se necesita entonces tomar la mejor decisión sobre qué sistema de archivos se ajusta a las propiedades exigidas para la optimización de recursos en un servidor de backups, de tal manera que la que mejores prestaciones ofrezca pueda ser implementada en un prototipo para su posterior uso.

Basados en la documentación encontrada sobre SDFS y ZFS que superan ampliamente a la de otros sistemas deduplicados, se decide realizar un estudio sobre estos.

Ambos son gratuitos y sostenidos por Comunidades de Desarrolladores y tienen compatibilidad con sistemas operativos Linux basados en Debian.

Estos dos sistemas se someterán al estudio, previamente configurados en un solo equipo virtual con el sistema operativo Linux Mint 17.1, para garantizar así la objetividad de las mediciones.

A continuación se detalla el proceso que se siguió para la configuración previa de los sistemas de deduplicación a evaluar.

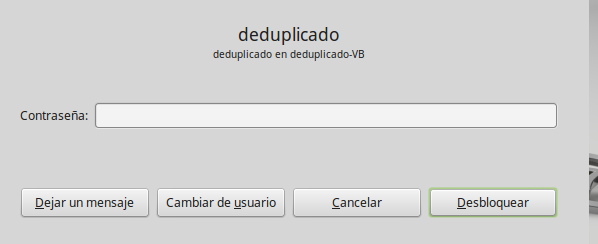
### Instalación y configuración del sistema operativo Linux Mint con el soporte para ZFS y SDFS

Para proceder con la instalación se sugiere el tutorial *How to Install Linux Mint 17.1 KDE Rebecca In Virtualbox In Ubuntu (Host).* (LinuxAndUbuntu, 2015, http://www.linuxandubuntu.com/video-tutorials/how-to-install-linux-mint-17-1-kde-rebecca-in-virtualbox-in-ubuntu-host).

Con la única variante en la versión del escritorio que se utilizará.

Al concluir la instalación se debe iniciar sesión con las credenciales de usuario para comenzar a trabajar.

Para esta instalación se recomienda el tutorial *ZFS-Linux* de: (Hertz, 2013, https://github.com/GlenHertz/ZFS-Linux), tomando en cuenta que el autor tubo una instalación fallida, por lo cual no recomienda su uso, aunque agregándole información obtenida por experiencia personal se logró tener éxito en la instalación corrigiendo los pasos conflictivos.



**Figura 9-2.** Ventana de inicio de sesión.

**Fuente:** Captura de pantalla de inicio de sesión Linux Mint 17.1

#### Instalación del soporte para ZFS desde el repositorio oficial de ZFS nativo para Ubuntu

A continuación se detalla paso a paso el proceso que se siguió para la instalación del sistema de deduplicación ZFS.

## Iniciar sesión como súper usuario

*sudo –i*

## Agregar el repositorio de zfs:native en paquetes estables

*dd-apt-repository --yes ppa:zfs-native/stable*

## Actualizar la base local de software

*apt-get update*

## Instalar las herramientas de compilación *build-essential*

*apt-get install --yes build-essential*

## Agregar el hosti al directorio *etc*

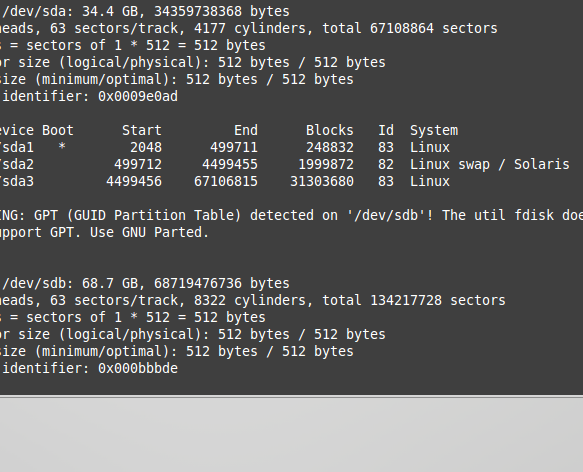
*hostid > /etc/hostid*

## Instalar los módulos necesarios para el soporte ZFS

*apt-get install spl-dkms zfs-dkms ubuntu-zfs mountall zfs-initramfsmodprobe zfsdmesg | grep ZFS:*

*Nota: Los pasos anteriores se repetirán en el futuro, por lo cual nos referiremos a estos en el futuro como instalación del soporte ZFS.*

## Se particionará el disco duro vacío para que quede tal como se puede apreciar en la Figura 10-2.



**Figura 10-2.** Particiones del disco duro.

**Fuente:** Captura de pantalla luego de creadas las particiones.

## Se crearán los volúmenes con ZFS donde irán nuestros archivos en el disco2.

*zpool create -O mountpoint=none rpool raidz1 ata-WDC\_WD30EFRX-1 ata-WDC\_WD30EFRX-2 ata-WDC\_WD30EFRX-3 cache ata-KINGSTON-part2zfs create -o mountpoint=/ rpool/rootzpool set bootfs=rpool/root rpoolzfs set compression=lz4 rpoolzfs create -o mountpoint=/home -o compression=off rpool/homezfs create -o mountpoint=/mnt/pictures -o compression=off rpool/pictureszfs create -o mountpoint=/mnt/music -o compression=off rpool/musiczfs create -o mountpoint=/mnt/mythtv -o compression=off rpool/mythtvzfs create -o mountpoint=/scratch rpool/scratchzpool export rpoolzpool import -R /mnt rpool*

## Con las carpetas montadas en los espacios ZFS se reiniciará la máquina virtual desde el *live cd* nuevamente.

## Instalar nuevamente el soporte para ZFS.

## Se creará una carpeta donde se montará el disco 1

*mkdir /sda3*

## Montar el disco 1

*mount /dev/sda3 /sda3*

## Importar el zpool creado anteriormente

## Copiar todo el contenido del disco 1 en las particiones del disco 2

## Reparar el grub y otros archivos faltantes posibles con el *chroot*

## Agregar el soporte de ZFS al grub para tenerlo al encender el servidor desde el disco duro 2

## Para evitar errores se recargará el *initramfs* y actualizará el grub

## En caso de errores de not found se debe reparar con la ruta absoluta por la fuerza

## Reparar la entrada del grub para que se vea así.

## Instalar el grub por la fuerza en la partición de 256 Mb

## Habilitar el *root login* desde el menú

## Salir de chroot y desmontar todo

## Exportar el pool y reiniciar. En caso de errores relacionados a la instrucción umount ignorar y reiniciar manualmente.

## Probablemente al reiniciar no pase de la pantalla negra, únicamente se debe reiniciar nuevamente el equipo, todo esto ocasionado por la carga forzosa de initramfs desde el grub no actualizado.

## Revisar que al encender el equipo cargue los módulos de soporte de ZFS y revisar la organización jerárquica del pool.

#### Espacio para SDFS

Aquí se detallan los pasos seguidos en el proceso de instalación del espacio para el sistema de deduplicación SDFS.

## Desde la página oficial del proyecto SDFS se debe descargar el paquete correspondiente a la distribución basada en Ubuntu 14.04

<http://www.opendedup.org/quickstart>

## Instalar desde el gestor de paquetes el archivo descargado



## Crear un volumen de 16 GB llamado pool0 en donde se guardarán los backup



## Montar en una carpeta para poder acceder al espacio reservado



Una vez instalados los sistemas de deduplicación se procedió a la instalación, configuración y uso del servidor de backups Bacula para la gestión de los backups y la posterior deduplicación de los mismos con la finalidad de determinar y evaluar el rendimiento de los sistemas de deduplicación ZFS y SDF previamente instalados. Todo el proceso de instalación, configuración y uso de Bacula junto con ZFS y SDFS se detalla en los Anexos A y B.

## Establecimiento de los parámetros para la elección de un sistema de archivos deduplicado para la implementación en un servidor de backups

Las características que se exponen aquí son estándares de la industria propuestos por varios autores expertos en el uso de sistemas de archivos deduplicados:

* **Existencia de una página oficial**

Se refiere a si existe algún sitio web propio del proyecto, el cual proveerá de información o links de descarga y soporte para los software a utilizar.

* **Existencia de documentación en idioma Español**

Se refiere a archivos, enlaces o cualquier información útil compartida con el software o en su lugar de origen, en idioma Español.

* **Existencia de documentación en idioma Inglés**

Se refiere a archivos, enlaces o cualquier información útil compartida con el software o en su lugar de origen, en idioma Inglés.

* **Existencia de un foro o blog para solventar dudas**

Se refiere a si existe o no un espacio de mensajería y comunicación en internet en donde se puedan dejar y contestar preguntas relacionadas al proyecto.

* **Ratio de deduplicación**

Valor porcentual referente a la proporción de tamaño relativa entre el tamaño en disco original y el resultante.

* **Tiempo en generar el backup inicial**

Medida en segundos es generar el primer backup de archivos.

* **Porcentaje de incremento por MB sobre el primer backup incremental**

Incremento de tamaño en MB entre el primer backup y el segundo.

* **Taza de MB/s de escritura en disco durante el proceso de deduplicación**

Velocidad MB/S a la que se escribieron los datos en el disco duro destino.

* **Existencia de paquetes personalizados para un sistema operativo o distribución en particular.**

Paquete de software desarrollado por una empresa o comunidad para su uso en una distribución específica.

* **Método de almacenamiento**

Algoritmo o algoritmos con los que se tratan los datos a almacenar y como se organizan.

* **Flexibilidad de almacenamiento**

Trata acerca del lugar y la manera que permite el almacenamiento de datos una tecnología.

* **Soporte de deduplicación en línea.**

Se refiere a si el proyecto permite el almacenamiento de los datos el localizaciones remotas.

* **Soporte de deduplicación fuera de línea.**

Se refiere a si el proyecto permite el almacenamiento de archivos de manera local.

* **Soporte de Compresión**

Se refiere a si se puede aplicar algún algoritmo de compresión además de la deduplicación para optimizar espacio.

***Nota***

*Otro punto a tomar en cuenta pudo haber sido el algoritmo de deduplicado que aplica cada tecnología, pero hoy en día, las tecnologías de deduplicado comparten algoritmos mutuos, teniendo cerca de 100 algoritmos similares, los cuales se aplican dependiendo el caso.* (CIOSPAIN, 2012)*.*

**CAPÍTULO III**

**3.** **MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS DE RESULTADOS**

## Estudio comparativo de sistemas de archivos para deduplicación

### Estudio a priori

**Tabla 2-3:** Parámetros para la elección a priori de un sistema de archivos deduplicado.

|  |  |  |
| --- | --- | --- |
| **Parámetro de Evaluación** | **Sistemas de Archivos Deduplicados** | |
| **SDFS** | **ZFS** |
| Existencia de una página oficial. | Si | Si |
| Existencia de documentación en idioma Español. | No | No |
| Existencia de documentación en idioma Inglés. | Si | Si |
| Existencia de un foro o blog para solventar dudas. | No | Si |
| Existencia de paquetes personalizados para un sistema operativo o distribución en particular. | No | Si |

**Elaborado por:** MANCHENO, A. 2015.

De acuerdo a la experiencia, se procede a ponderar la importancia de cada uno de los parámetros.

**Tabla 3-3:** Parámetros ponderados.

|  |  |  |
| --- | --- | --- |
| **Parámetro de Evaluación** | **Sistemas de Archivos Deduplicados** | |
| **SDFS** | **ZFS** |
| Existencia de una página oficial. | 3 | 3 |
| Existencia de documentación en idioma Español. | 0 | 0 |
| Existencia de documentación en idioma Inglés. | 3 | 3 |
| Existencia de un foro o blog para solventar dudas. | 0 | 3 |
| Existencia de paquetes personalizados para un sistema operativo o distribución en particular. | 2 | 4 |
| Totales | 8 | 13 |

**Elaborado por:** MANCHENO, A. 2015.

**Medidas de Ponderación**

Se considera como un factor sumamente importante, a la existencia de un paquete personalizado para la distribución que se utilizará.

Su existencia significará un valor de 4, aunque de no existir, se le calificará con un valor de 2, dado que únicamente significaría un incremento no muy elevado de esfuerzo.

La existencia de una página oficial será evaluada con 4 puntos de existir en nuestro idioma y 3 en caso de que esté en inglés, 2 puntos si está en algún otro idioma y 0 en caso de no existir.

Acerca de la documentación, se calificará con 3 puntos como máximo al existir esta en idioma español, dado que es menos importante que los parámetros anteriores.

Acerca de foros o blog, de igual manera obtendrán una máxima calificación de 3 puntos, por las razones antes expuestas.

**Tabla 4-3:** Medidas de ponderación.

|  |  |
| --- | --- |
| **Parámetro de evaluación** | **Calificación** |
| Muy útil y disponible. | 4 |
| Página en Español | 4 |
| Página no en Inglés | 3 |
| Documentación y Foros | 3 |
| Muy útil y no disponible | 2 |
| Página en otros Idiomas | 2 |

**Elaborado por:** MANCHENO, A. 2015.

**Explicación**

De la tabla anterior si todos los sistemas cumplieran todos los parámetros de evaluación se lograría la suma de 20 puntos lo que representaría el 100%.

De tal manera para la representación se empleará una regla de tres simple.

***20 100%***

***8 X = (8\*100)/20 = 40%***

**Resultados**

***SDFS ZFS***

*40% 65%*

**Figura 11-3.** Gráfico análisis de cumplimiento condiciones a priori entre SDFS

y ZFS.

**Elaborado por:** MANCHENO, A. 2015.

ZFS es el sistema de archivos que obtiene el mejor resultado con un 65% debido a la información disponible y sus características.

### Estudio a posteriori

#### Medición en el entorno de funcionamiento

Las siguientes mediciones serán tomadas en función a los resultados obtenidos en el proceso del ANEXO A (Instalación del entorno) y ANEXO B (Configuración del Entorno), para en función a los mismos deducir cual será el que nos brinde óptimas características en nuestro servidor de backups.

A continuación se muestra la medición de características a posteriori entre SDFS y ZFS.

**Tabla 5-3:** Análisis a posteriori de parámetros de funcionamiento.

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **SDFS** | **ZFS** |
| Ratio de deduplicación | 9.7 gb | 4.9gb |
| Tiempo en generar el backup inicial | 90 sec | 21 sec |
| Porcentaje de incremento por MB sobre el primer backup incremental | 881.81% | 197.96% |
| Taza de MB/s de escritura en disco durante el proceso de deduplicación. | 4968.7 | 4968.7 |

**Elaborado por:** MANCHENO, A. 2015.

**Explicación**

Se procede a la ponderación individual de cada fila, tomando en cuenta que los parámetros 1 y 3 representan uso de espacio y el parámetro 2 representa tiempo, recursos que se desea optimizar, por lo cual en la ponderación se los considerará mejores mientras sean más bajos, en cambio el parámetro 4 representa la velocidad a la que se escriben datos en el disco duro del equipo, por lo cual se tiene una relación inversamente proporcional entre este parámetro y el tiempo empleado en una tarea, por lo cual este parámetro es mejor al ser mayor.

Dada la naturaleza de los datos se decide medir la utilización comparada entre ambos, siendo 100 puntos el que tenga mayor uso y su correspondiente en función a este en el sistema de archivos oponente.

*(Optimizar = Utilizar menos recursos)*

**Tabla 6-3:** Ponderación de puntos de uso.

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **SDFS** | **ZFS** |
| Ratio de Deduplicación | 100 | 48.96 |
| Tiempo en generar el backup inicial | 100 | 23.33 |
| Porcentaje de incremento por MB sobre el primer backup incremental | 100 | 22.45 |
| Taza de MB/s de escritura en disco durante el proceso de deduplicación. | 100 | 100 |
| Totales | 400 | 194.74 |

**Elaborado por:** MANCHENO, A. 2015.

**Resultados**

De acuerdo a la tabla anterior se obtuvieron los siguientes resultados

**SDFS** 400 puntos de uso

**ZFS** 194.74 puntos de uso

**Figura12-3.** Gráfico comparación A Posteriori entre SDFS y ZFS.

**Elaborado por:** MANCHENO, A. 2015.

#### Medición de características de funcionamiento SDFS y ZFS

**Tabla 7-3:** Análisis de características de funcionamiento.

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **SDFS** | **ZFS** |
| Método de Almacenamiento. | Bloque de deduplicación variable o estático | Hash de 256 bits |
| Flexibilidad De Almacenamiento. | Local, en red, en nodos múltiples, en la nube. | Local, en red, en nodos múltiples, en la nube. |
| Soporte de Inline Deduplication | Si | Si |
| Soporte de Offline Deduplication | No | Si |
| Soporte de Compresión | No | Si (LZ4) |

**Elaborado por:** MANCHENO, A. 2015.

**Explicación**

Se procede a ponderar con el método de regla de 3, tomando en cuenta que el parámetro 1 se medirá en base a los resultados expuesto en (Besembel, y otros, 2008), donde realizan un estudio comparativo entre diferentes técnicas de almacenamiento de información, luego de lo cual declaran a las tablas hash como el método de mejores prestaciones para el almacenamiento de datos.

**Tabla 8-3:** Ponderación análisis de características de funcionamiento.

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **SDFS** | **ZFS** |
| Método de almacenamiento. | 0 | 1 |
| Flexibilidad de almacenamiento | 1 | 1 |
| Soporte de Inline Deduplication | 1 | 1 |
| Soporte de Offline Deduplication | 0 | 1 |
| Soporte de compresión | 0 | 1 |
| Totales | 2 | 5 |

**Elaborado por:** MANCHENO, A. 2015.

**Resultados**

SDFS 2 puntos

ZFS 5 puntos

**Figura 13-3.** Gráfico de soporte de funciones.

**Elaborado por:** MANCHENO, A. 2015.

**Consideración**

Pese a no ser parte del estudio, también se debe tomar en cuenta los resultados del punto 2.2.1. del capítulo 2, el cual nos indica que es mucho más fácil instalar SDFS en una partición a montar, comparado con ZFS que requiere un proceso de copia forzosa de un sistema operativo a otro disco.

### Resultados obtenidos

Tanto en la medición a priori y a posteriori se denota una mayor prestación de facilidades por parte de ZFS.

Además en cuanto al soporte que presenta también denota una superioridad considerable sobre SDFS.

**Tabla 9-3:** Resumen de resultados.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sistemas de Archivos** | **Comparación A priori (cumplimiento)** | **Comparación a posteriori (costo)** | **Comparación de funcionalidades** |
| SDFS | 40% | 400 puntos | 2 |
| ZFS | 65% | 194.74 puntos | 5 |

**Elaborado por:** MANCHENO, A. 2015.

Finalmente se comprobará la optimización de espacio que se obtuvo entre ZFS que es la mejor opción de deduplicación contra el tamaño real de los archivos involucrados.

Para el presente caso, por la naturaleza de los datos se realiza una nueva tarea de respaldo sobre un total de 6.6 GB de archivos, los cuales son imágenes iso del mismo sistema operativo, haciéndolos idóneos para la deduplicación por las características tan repetitivas de los datos.

En el Anexo D se adjunta la evidencia visual del tamaño.

Luego del proceso de deduplicación, el backup resultante tiene un tamaño de 4.9 GB, siendo considerable el ahorro producido.

**Interpretación**

Se demuestra entonces que una vez realizado el estudio, es el sistema de archivos ZFS el que presenta mejores prestaciones tanto en funcionalidades, en resultados de campo y en prestaciones. Esto nos deja a ZFS como el sistema a implementar en el prototipo, proceso del cual se hablará a profundidad en el capítulo 4.

## Planteamiento del prototipo

A continuación se procede a especificar los pasos a seguir para una correcta instalación y preparación del sistema escogido para la deduplicación.

### Diseño del escenario de pruebas

#### Requerimientos hardware

* Para la presente implementación se propone un escenario de un servidor que contendrá la información a ser respaldada.
* Un servidor que almacenará los backups y un servidor intermedio ejecutando BACULA, el cual se encargará de periódicamente ejecutar los trabajos de respaldo.
* Considerando la reseña de: (Gonzalez, 2011, http://constantin.glez.de/blog/2011/07/zfs-dedupe-or-not-dedupe), se elige la cantidad de RAM que se utilizará, se decide 4gb de RAM.
* Para realizar el proceso de instalación se necesita tener 2 discos duros para en resumen instalar el sistema operativo en uno de ellos y formatear el otro disco en ZFS, luego de lo cual desde un live CD se migrará por la fuerza el sistema operativo instalado en el primero hacia el segundo.

#### Configuración de la red

* Para la configuración en red, se deberá primero establecer una configuración de IP estática en cada uno de los servidores, en el presente desarrollo se sugiere utilizar el segmento de red 192.168.50.0, para lo cual las IP quedarías distribuidas de la siguiente manera.

*Servidor Origen 192.168.50.100*

*Servidor Bacula 192.168.50.101*

*Servidor de Backups 192.168.50.102*

*Todos con máscara 255.255.255.0.*

#### Requerimientos software

**Tabla 10-3:** Requerimientos software para la implementación de ZFS.

|  |  |
| --- | --- |
| **Software** | **Licencia** |
| Virtualbox 4.0 | Libre |
| Linux Mint 17.1 64 bits | Libre |
| Bacula | Libre |
| Paquete de instalación de ZFS | Libre |

**Elaborado por:** MANCHENO, A. 2015.

### Implementación del escenario de pruebas

#### Instalación y configuración de ZFS

* Basados en los resultados del capítulo 3, se decide la implementación del entorno ZFS, para lo cual se utilizará una máquina virtual con 4gb en RAM como lo recomienda la página oficial de ZFS.
* Como se había especificado en los requerimientos software del punto 8.1.2., el virtualizador será Virtualbox en su versión 4, y el sistema operativo será Linux Mint 17.1 64 bits Mate Edition, todo esto basado en la existencia del paquete personalizado de ZFS para Ubuntu en repositorios.
* Siguiendo los pasos especificados en el punto 2.2.1. del capítulo 2, se procede a instalar y mover a ZFS el sistema operativo.
* Instalar desde los repositorios oficiales el soporte para ZFS, luego de lo cual se procede a crear el pool en el que se almacenarán los datos.
* Dentro de este pool se deberán crear los apartados para los datos en donde se almacenarán los archivos propios del sistema operativo a ser migrado.
* Exportar el pool para que se apliquen los cambios y se procede a lanzar nuevamente el sistema operativo live desde el CD de instalación.
* Dentro de este se vuelve a instalar el soporte para ZFS y se crearán las carpetas para mostrar los discos a ser migrados.
* Crear un volumen ZFS en las particiones del segundo disco y con el comando cp se procede a transferir todos los archivos hacia este disco.
* Luego de copiado se debe mover el grub y reparar sus entradas para que inicie en el segundo disco.
* Reparar la entrada del grub para que habilite el soporte ZFS desde el proceso boot.
* Reiniciar el sistema y muy probablemente no se verá más que una pantalla negra, se debe reiniciar nuevamente para que el initramfs cargue los nuevos archivos modificados.
* Con eso se tendrá listo el servidor para almacenar backups con ZFS.

#### Instalación y configuración de Bacula

* Se debe seguir el mismo procedimiento para instalar el sistema operativo en el servidor dedicado para Bacula, pero sin el proceso de migración entre discos, para este se necesitará una máquina virtual con 4gb de RAM y 1 disco duro de 60gb.
* Según la documentación oficial de Bacula, este está disponible en versiones de mysql y postgresql, por experiencia se escoge la versión postgresql.
* Desde los repositorios oficiales se deberá instalar postgresql en su versión 9.2 y se procede a cambiar su acceso y sus credenciales en los archivos pg\_hba.conf y postgres.conf.
* Ya con el sistema operativo listo, se procede a instalar a través del gestor de paquetes apt-get los paquetes bacula, bacula-dir, bacula-sd.
* Con el editor de textos de preferencia, se procede a configurar el archivo /etc/bacula/bacula-dir.conf para poder encontrar las credenciales y rutas del director para bacula (usuario), además se debe escoger que el tipo de backup a obtener será un solo archivo y en donde se almacenará el mismo, es aquí en donde se configurará la dirección hacia la que se dirigirán los backups creados dentro de la red.
* La sintaxis debería seguir el siguiente formato nombre\_de\_usuario@ip\_destino:ruta\_absoluta. Un ejemplo de especificación de dirección en red sería [root@192.168.50.102:/backups](mailto:root@192.168.50.102:/backups), dado que todo el sistema estará hospedado en ZFS es irrelevante la carpeta que se escoja.
* De igual manera se debe editar el archivo /etc/bacula/bacula-sd.conf para que las credenciales coincidan con las del director y la localización de donde se obtendrán los backup. En el presente caso sería [root@192.168.50.100:/](mailto:root@192.168.50.100:/) para obtener un backup completo del disco root, o [root@192.168.50.100:/home/deduplicado](mailto:root@192.168.50.100:/home/deduplicado) para obtener un backup completo de todos los archivos y configuraciones producidos por el usuario “deduplicado”.
* Ya configurado bacula, se procederá a ejecutarlo utilizando la interfaz que brinda bconsole (consola de bacula).
* Desde una terminal del sistema operativo se procede a ejecutar con permisos elevados la consola de bacula con sudo bconsole. Después de la confirmación se lanzará un prompt con un \*. Con el comando run se escogerá el trabajo que se desea realizar, para el presente caso Local Files Backup y se interactuará con el prompt para obtener un backup completo.
* Dentro del archivo /etc/bacula/bacula-dir.conf se puede configurar backups periódicos a una hora determinada y de un tipo determinado.
* Se tienen configuradas las opciones de backup incremental diario, comparativo semanal y completo mensual, por lo tanto se las dejará por defecto, los mismos se llevarán a cabo a las 2 am del día indicado.
* Dentro del mismo prompt \*, para ejecutar un trabajo de restauración se lo hará con el comando restore, de igual manera se interactuarán con las opciones que se ajusten a las necesidades actuales.
* Los pasos detallados se explican en el Anexo A.
* Para el correcto funcionamiento del entorno son necesarios ajustar los puertos 60, 5432, 10000, 20, 21 y 22 en abiertos, esto se realizará en ufw que es el firewall que trae Linux Mint 17.1 por defecto.

##### CONCLUSIONES

* Los sistemas de archivos deduplicados brindan un ahorro de recurso por la eliminación de información duplicada en disco, el backup con bacula en el formato archivo, explota eficientemente las propiedades de los sistemas de archivos deduplicados, dado que este se genera en el servidor, produciendo un elemento mucho más fácil de mover a través de redes.
* El sistema SDFS ofrece capacidades bastante limitadas, aunque muy equilibradas a la facilidad de su instalación, mientras que ZFS ofrece altas prestaciones, también muy equilibradas a la dificultad que presenta realizar una instalación forzosa del sistema operativo sobre el disco preparado.
* Como se menciona en la **Tabla 9-3**, ZFS brinda mejores prestaciones en cuanto al cumplimiento, costo y funcionalidades, dando como resultados el 65%, 194.74 puntos y un total de 5 de las prestaciones medidas respectivamente. Mientras que SDFS dio como resultados un 45%, 400 puntos y un total de 2 de las prestaciones medidas respectivamente, lo cual vuelve a ZFS un sistema de deduplicación más completo y que utiliza menos recursos que SDFS volviéndolo más eficiente, además de lo expuesto en la **Tabla 3-3**, es mayor el soporte e información encontrada para ZFS.
* ZFS tendrá un mejor desempeño en un entorno de producción, dado que su única desventaja evidenciada es su instalación lo cual no sería significativo por ser un proceso que únicamente se da 1 vez.

##### RECOMENDACIONES

* Antes de iniciar con la implementación del prototipo que se propone en este estudio, es importante obtener abundantes conocimientos sobre el manejo de la distribución de Linux a utilizar dado que se emplean instrucciones de nivel intermedio.
* Para una futura investigación se recomienda realizar un estudio comparativo entre la implementación de un servidor de backups bacula con ZFS en un entorno Linux de la familia Debian contra uno de la familia RED HAT, orientado hacia los backups en red dado que Red Hat es una versión empresarial de Linux con ventajas propias de su denominación.
* Para un óptimo funcionamiento se debería implementar el servidor donde se almacenarán los backup con discos de estado sólido, los cuales tienen mejor desempeño que los discos IDE y SATA en cuanto a velocidad de escritura/lectura.

# 

# GLOSARIO

**SDFS** Sistema de Archivos deduplicado sostenido por opendedup

**ZFS** Sistema de Archivos deduplicado sostenido por Oracle

**LINUX** Nombre común que se le da al sistema operativo GNU/LINUX

**DEBIAN** Distribución de Linux con sistema de paquetería apt-get.

**RED HAT** Distribución de Linux con sistema de paquetería yum.

**BACKUP** Copia o respaldo de información seleccionada.

**SERVIDOR** Equipo de alta disponibilidad que provee de servicios a los clientes.

**BACULA** Herramienta de creación y restauración de backups.

**TERMINAL** Interfaz de interacción con el Sistema operativo.

# BIBLIOGRAFÍA

**AIGNEREN, M.** *Diseño Cuantitativos: Análisis e Interpretación de la Infromación.* 2009. http://aprendeenlinea.udea.co/revistas/index.php/ceo/article/viewFile/1651/1303

2014-09-20

**ALVAREZ, CARLOS.** *Conceptos básicos: Deduplicación*. 2011. http://www.netapp.com/es/communities/tech-ontap/es-tot-bb-depublication.aspx

2014-09-20

**BUENO BÉJAR, ANTONIO.** *Data Deduplicación en Windows Server 2012 R2*. 2013. http://blogs.itpro.es/abueno/2013/10/18/data-deduplicacion-en-windows-server-2012-r2-parte-ii-de-ii/.

2014-10-26

**CLOUDADMINS.ORG.** *Opendedup:Deduplicación en línea*. 2014. http://www.cloudadmins.org/opendedup-deduplicacion-en-linea/

2015-01-20

**DIGITAL, WESTERN. 2013.** *Tecnología Progressive Deduplication*. 2013. http://www.wdc.com/wdproducts/library/Flyer/ESN/2579-800022.pdf

2015-01-17

**DOMINGUEZ, MARTIN.** *La deduplicación en el backup.* 2014. http://www.whitebearsolutions.com/la-deduplicacion-en-el-backup/

2015-02-20

**DORION, PIERRE.** *Tecnología de deduplicación de datos: casos que justifican la deduplicación*. 2009.

http://searchdatacenter.techtarget.com/es/consejo/Tecnologia-de-deduplicacion-de-datos-casos-que-justifican-la-deduplicacion

2015-03-09

**EMC.** *Glosario de EMC: Deduplicación de longitud variable*. 2014. http://mexico.emc.com/corporate/glossary/variable-length-deduplication.htm.

2015-01-13

**EMC.** *Glosario de EMC: Deduplicación de longitud variable*. 2014. http://venezuela.emc.com/corporate/glossary/variable-length-deduplication.htm

2015-01-13

**FREEBSD.** *ZFS con compresión LZ4.* 2014.

http://freebsd.mx/freebsd-9-2-caracteristica-sobresaliente-zfs-con-compresion-lz4

2015-05-03

**GONZALEZ, CONSTANTIN.** *ZFS: To Dedupe or not to Dedupe*. 2011. http://constantin.glez.de/blog/2011/07/zfs-dedupe-or-not-dedupe

2015-01-23

**HERTZ, GLEN.** *ZFS on Ubuntu 14.04 Trusty Tahr (Linux Mint 17)*. 2013. https://github.com/GlenHertz/ZFS-Linux

2015-05-18

**HIPERTEXTUAL.COM.** *7 Razones por las que Linux es más utilizado que Windows en servidores*. 2014.

http://hipertextual.com/archivo/2014/05/linux-servidores/?utm\_content=buffer9884c&utm\_medium=social&utm\_source=twitter.com&utm\_campaign=buffer

2015-04-07

**HOEKSTRA, BERRY Y DAAN, WAGENAAR.** *Comparing open source deduplication performance for virtual machines*. 2012.

https://www.os3.nl/\_media/2011-2012/courses/lia/berry\_hoekstra\_daan\_wagenaar\_-\_lia-report-final.pdf

2014-11-22

**HP.***La deduplicación: un almacenamiento más eficaz*. 2009. http://h30458.www3.hp.com/es/es/smb/736096.html

2014-10-23

**JIMENEZ, FRANCISCO.** *Técnicas de deduplicación de datos y aplicación en librerías virtuales de cintas*. 2009. http://oa.upm.es/1803/1/PFC\_FRANCISCO\_JAVIER\_JIMENEZ\_PATRICIO.pdf

2014-12-06

**LINUXANDUBUNTU.** *How to Install Linux Mint 17.1 KDE Rebecca In VirtualBox In Ubuntu (Host)*. 2015.

http://www.linuxandubuntu.com/video-tutorials/how-to-install-linux-mint-17-1-kde-rebecca-in-virtualbox-in-ubuntu-host

2015-06-03

**LORTU.** *¿Qué es la deduplicación?*. 2013. http://www.lortu.es/Technology.aspx

2014-09-20

**MARTINEZ, RAFAEL.** *Sobre Linux.* 2014. http://www.linux-es.org/sobre\_linux

2015-08-07

**MUGARZA, MIKEL.** *Virtualización con software libre: Virtualización del almacenamiento*. 2012. http://www.mikelmugarza.com/?p=219

2014-12-04

**OPENDEDUP.ORG.** *Opendedup*. 2015. opendedup.org

2015-02-13

**ORACLE.COM.** *Propiedad de eliminación de datos duplicados de ZFS*. 2015. http://docs.oracle.com/cd/E37929\_01/html/E36655/gbscy.html#gjhbo

2015-07-05

**ORACLE.COM.** *Propiedades nativas de ZFS configurables*. 2015. http://docs.oracle.com/cd/E56339\_01/html/E53915/gazsd.html#ZFSADMINgjhav

2015-06-14

**PÉREZ DÍAZ, ALONSO.** *Servidor de log syslog*. 2013. http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=641

2014-09-20

**SLIDESHARE.** *Open Source Data Deduplication*. 2011. http://www.slideshare.net/RedWireServices/open-source-data-deduplication-8869930

2015-04-18

**STARWIND SOFTWARE.** *Data Deduplication methods: File-Level vs Block-Level vs Byte-Level deduplication*. 2009.

http://www.starwindsoftware.com/file-level-vs-block-level-vs-byte-level-deduplication

2014-10-17

**STEFFENS, HANS.** *¿Qué es la deduplicación de datos y cómo beneficia a las empresas?.* 2010.

http://liacolombia.com/2010/06/%C2%BFque-es-la-desduplicacion-de-datos-y-como-beneficia-a-las-empresas/

2014-10-19

**UC3M.** *Tablas Hash.* 2014. http://www.it.uc3m.es/labas/course\_notes/dynamic\_data\_structures\_hash\_tables\_what\_are\_they\_es.html

2015-08-19

**VILLANUEVA, PEDRO.** *Windows vs Linux*. 2007.

http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html

2015-06-23

**WHITEHOUSE, LAUREN.** *Cuándo y como utilizar la tecnología de deduplicación de datos en el safeguard de disco*. 2009.

http://searchdatacenter.techtarget.com/es/consejo/Comprender-las-ratios-de-deduplicacion-de-datos-en-los-sistemas-de-copia-de-seguridad

2015-02-12

**WENDT, JEROME.** *Preguntas frecuentes sobre la deduplicación de datos*. 2009. http://searchdatacenter.techtarget.com/es/cronica/Preguntas-frecuentes-sobre-la-deduplicacion-de-datos

2015-02-12

**WHITEHOUSE, LAUREN.** *Métodos de deduplicación de datos por bloques o por bytes*. 2009.

http://searchdatacenter.techtarget.com/es/consejo/Comprender-las-ratios-de-deduplicacion-de-datos-en-los-sistemas-de-copia-de-seguridad

2015-02-12

**WHITEHOUSE, LAUREN.** *Comprender las ratios de deduplicación de datos en los sistemas de copia de seguridad*. 2009.

http://searchdatacenter.techtarget.com/es/consejo/Comprender-las-ratios-de-deduplicacion-de-datos-en-los-sistemas-de-copia-de-seguridad

2015-02-12

# ANEXOS

ANEXO A. INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR BACULA.

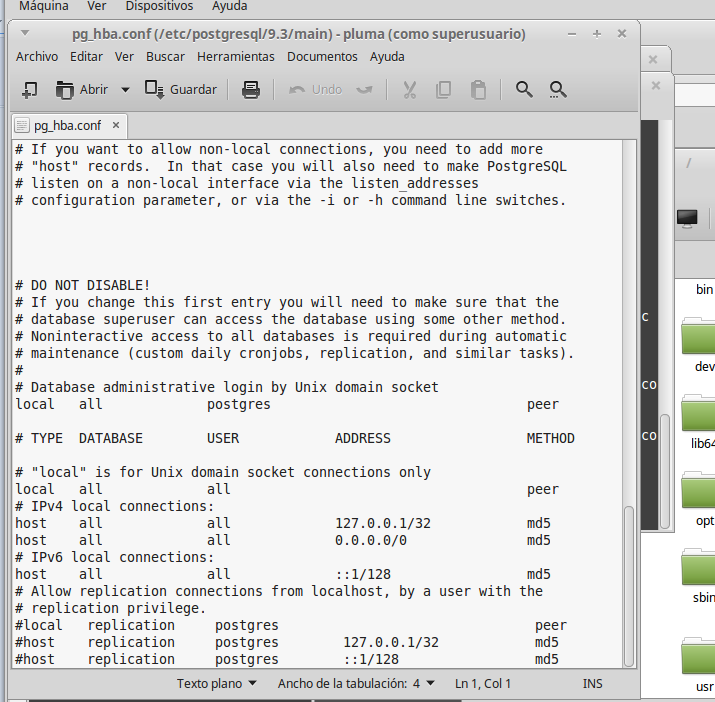
##### Instalación del servidor Bacula para gestionar backups

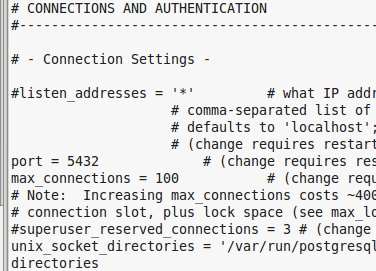
## El servidor bacula provee módulos para postgresql y mysql, por experiencia en la herramienta postgresql y por las 10 razones que se expresa en <http://www.postgresql.org.es/10_razones>, se decide la utilización del módulo con postgresql.

## Instalar postgresql-9.3 desde el repositorio oficial de Linux Mint



## Configurar los archivos *pg\_hba.conf* y *postgresql.conf* para permitir el acceso a los usuarios requeridos, se recomienda la siguiente configuración.





## 

## Reiniciar Postgresql



## Instalar los módulos requeridos para Bacula



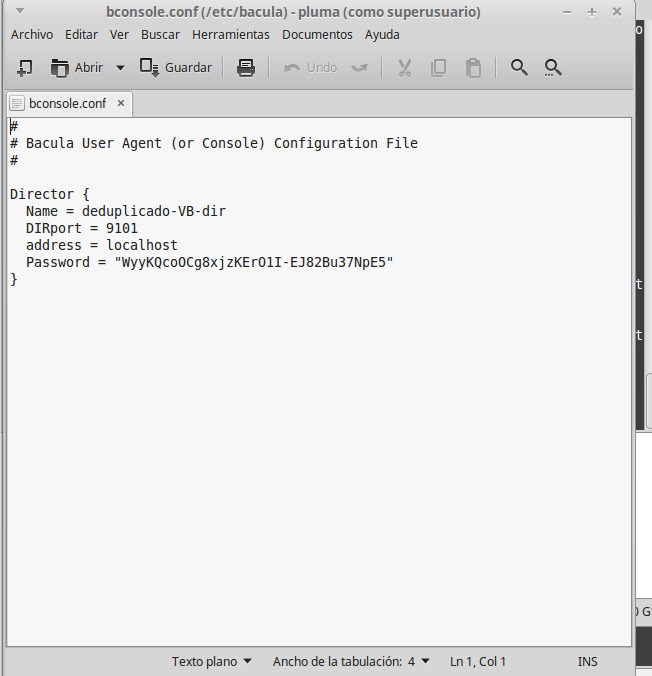
## Configurar el archivo bacula-sd.conf para especificar el directorio donde se guardarán los backup y el formato en el que se guardaran y el archivo bacula-dir.conf para gestionar los parámetros de clientes y rutas de donde se tomaran los backup.

Los documentos mencionados se pueden revisar en el Anexo 1, por ser considerados de importancia pero excesivamente grandes.

## Instalar la herramienta *bacula-console* para la ejecución de instrucciones hacia Bacula

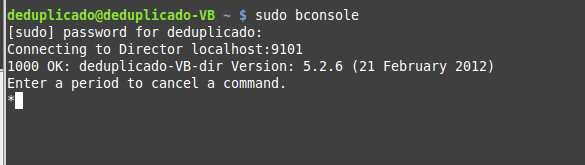


## Hay que revisar que el archivo de configuración *bconsole.conf* tenga las credenciales y las rutas correctas para el director de backups



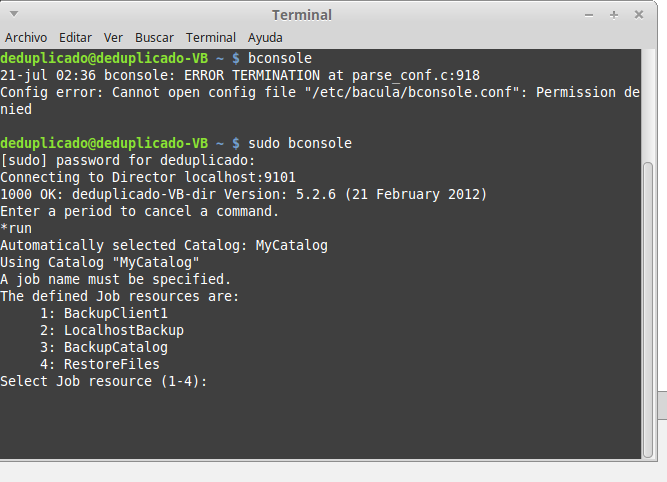
Finalmente se tiene el entorno listo para la ejecución de las pruebas y toma de métricas.

ANEXO B. USO DE BACULA

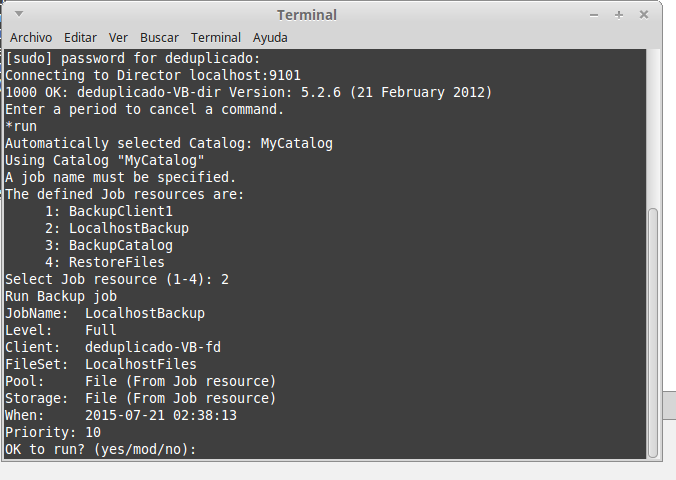


**\* Se necesitará permisos elevados de administrador**

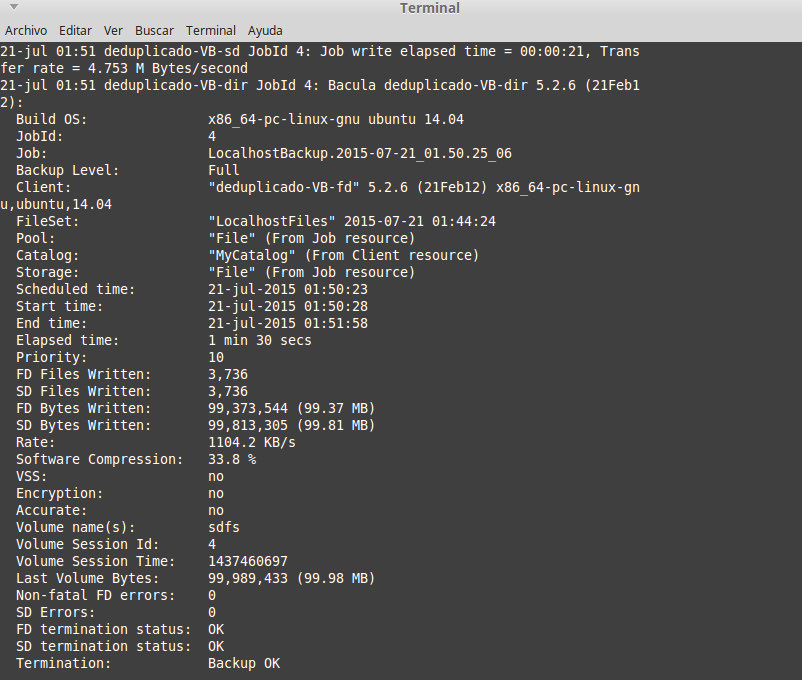
1. Ejecutar el comando *run*



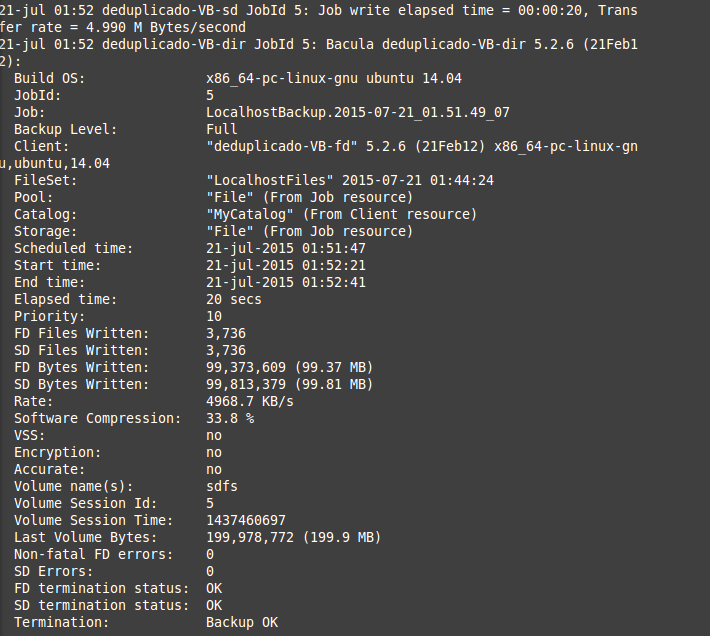
1. Escoger la opción que se creó con LocalhostBackup



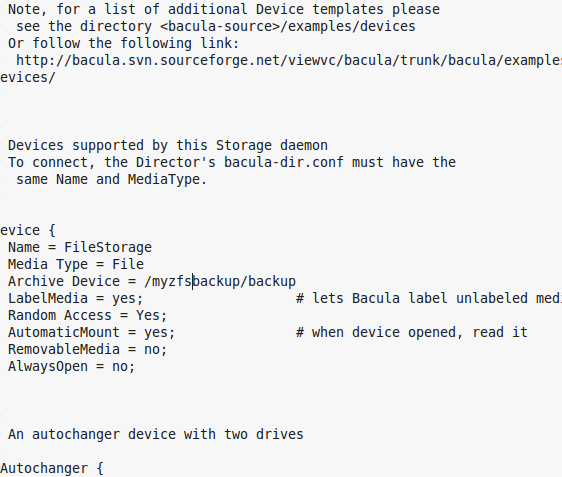
1. Confirmar con la palabra “yes”. Se emitirá un mensaje que para leerlo se puede utilizar el comando messages



1. Observar las características del PC y las características del backup. Al ser el primer backup el tiempo es 1 min 30 secs.
2. Agregar archivos a nuestro directorio a ser respaldado y repetir los pasos para obtener un backup incremental sobre el ya creado.



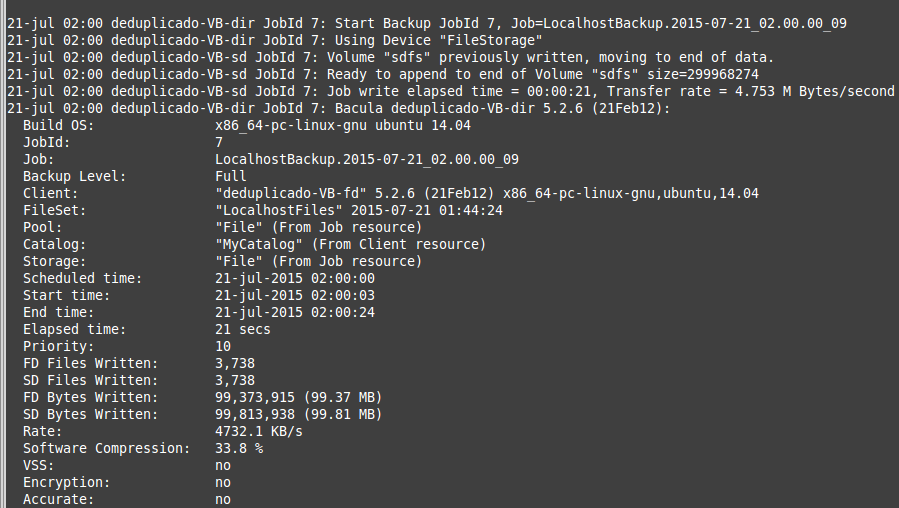
1. Observar que el tiempo es de 20 secs y el tamaño del backup aproximadamente se ha duplicado, todo esto por el algoritmo de deduplicado de SDFS.
2. En cuanto a ZFS marcar como objetivo en el archivo ZFS en el archivo del anexo 1



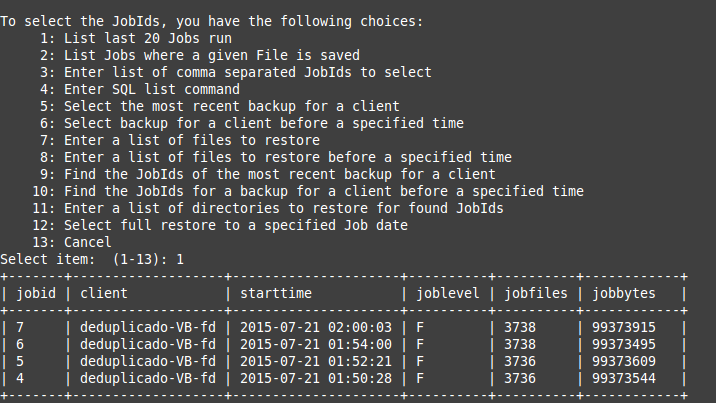
1. Guardar y repetir los pasos anteriores
2. Reiniciar el servidor *bacula-sd*



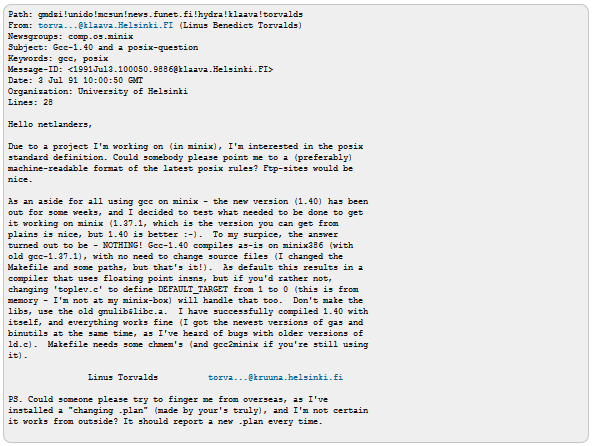
1. Abrir la utilidad *bconsole*
2. Ejecutar un nuevo backup inicial y un nuevo backup incremental



1. Revisar que el tiempo de ZFS pese a ser inicial toma únicamente 21 segundos
2. En cuanto al tamaño del archivo, en la tabla de resumen siguiente se pueden verificar los tamaños de los Job 4 y 5 (SDFS) y 6 y 7 (ZFS).

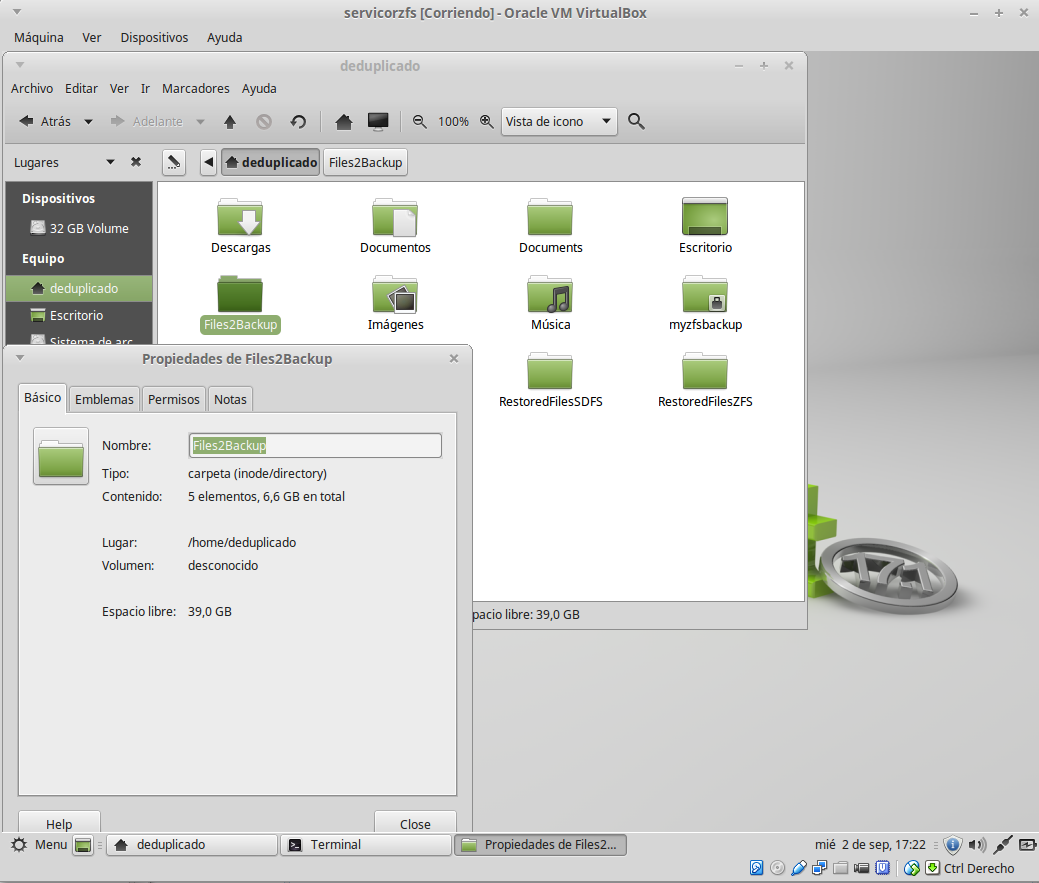


ANEXO C. MENSAJE DE LINUS BENEDICT TORVALDS

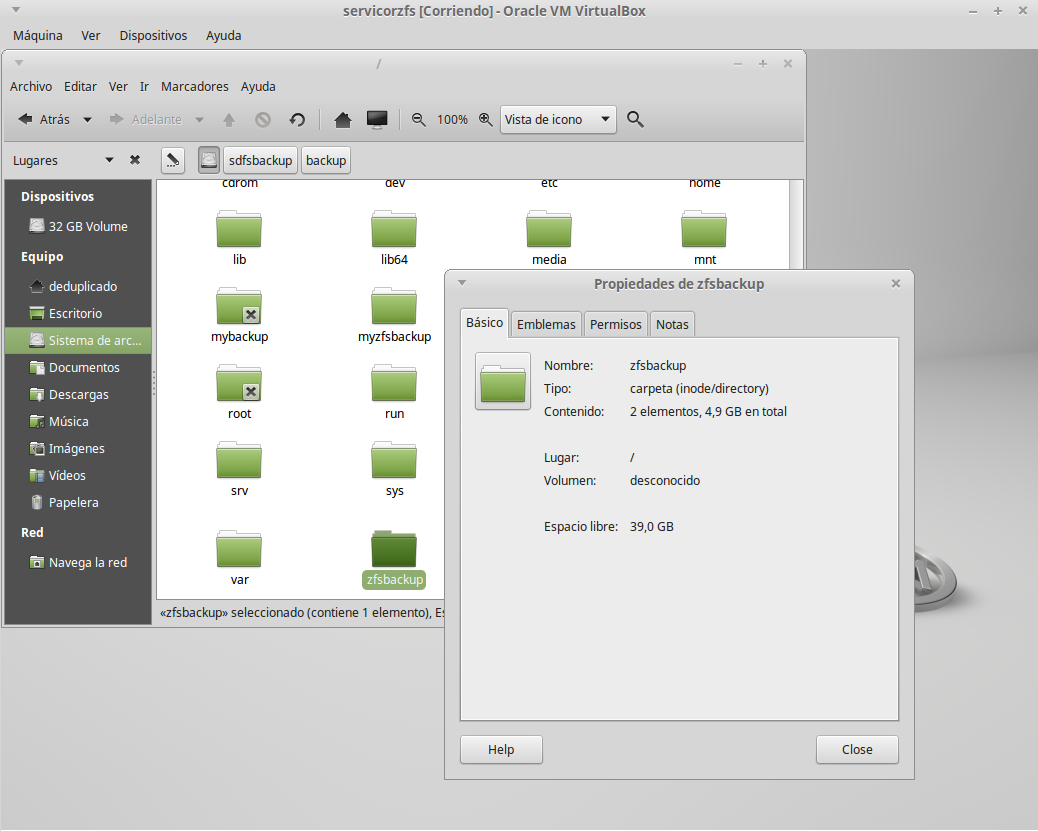


ANEXO D. ANTES Y DESPUÉS DE LA DEDUPLICACIÓN CON ZFS.

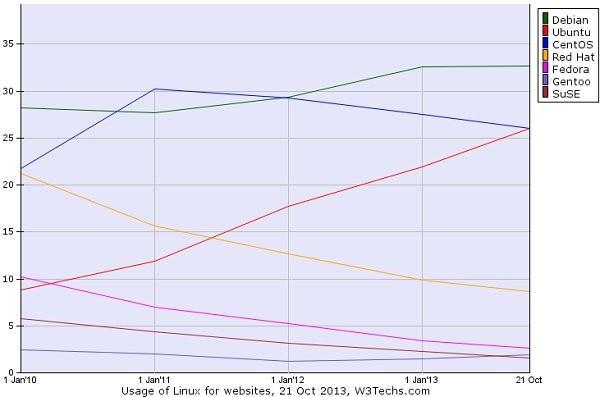
1. Archivos Originales.



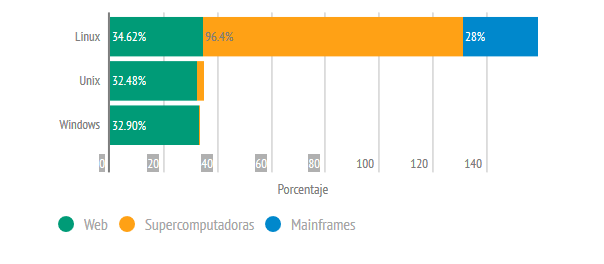
2. Archivos deduplicados con la tecnología ZFS.



**ANEXO E.** GRÁFICO DEL USO DE LINUX EN SERVIDORES ENERO 2011 – OCTUBRE 2013.

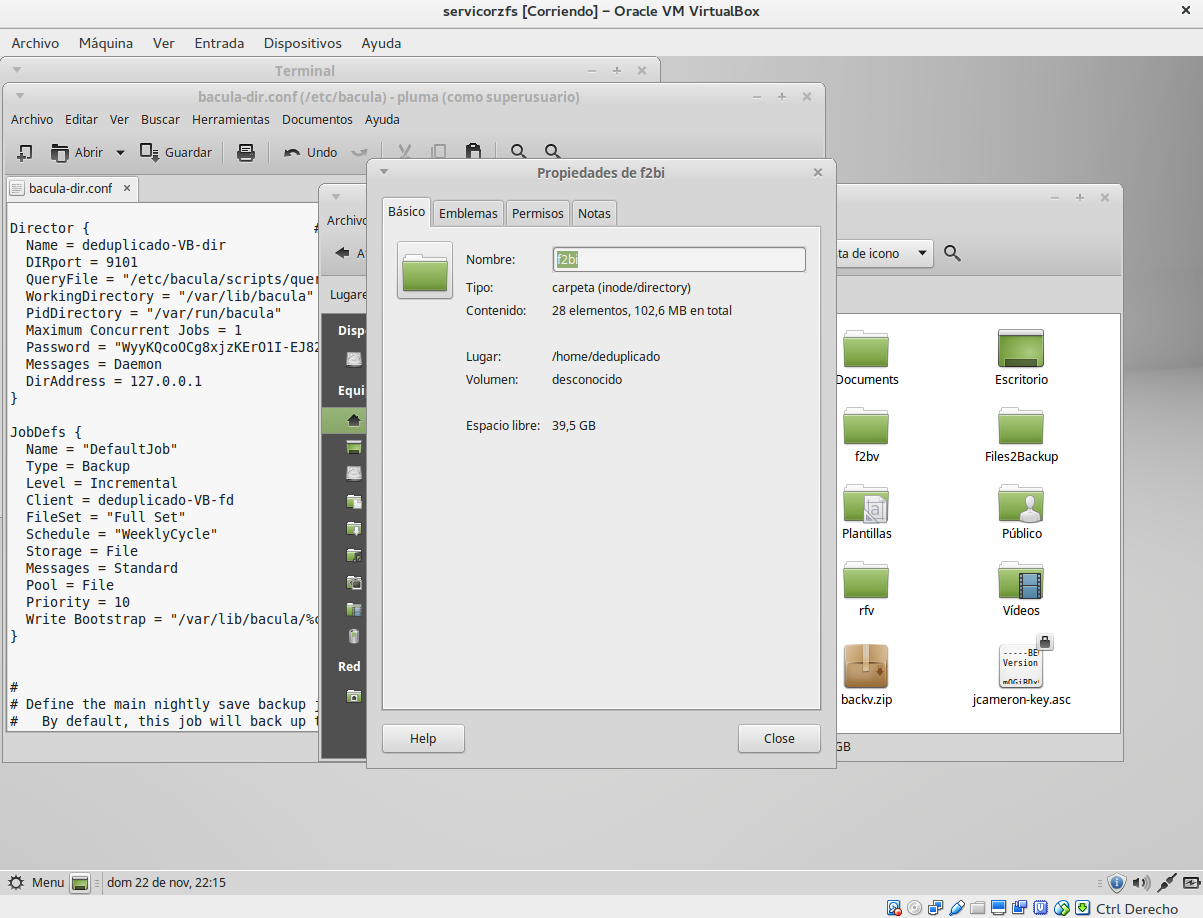


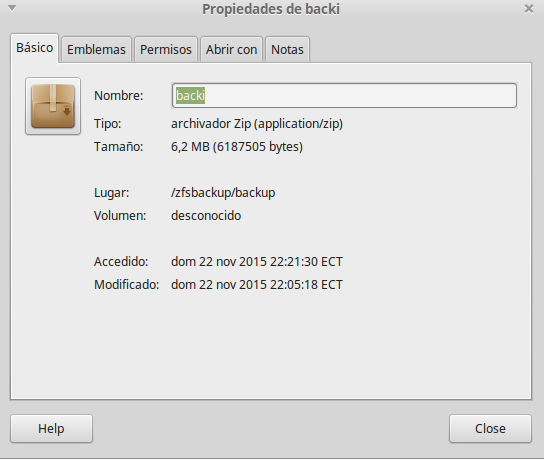
**ANEXO F.** GRÁFICO DEL USO DE LINUX EN DIFERENTES ÁREAS.



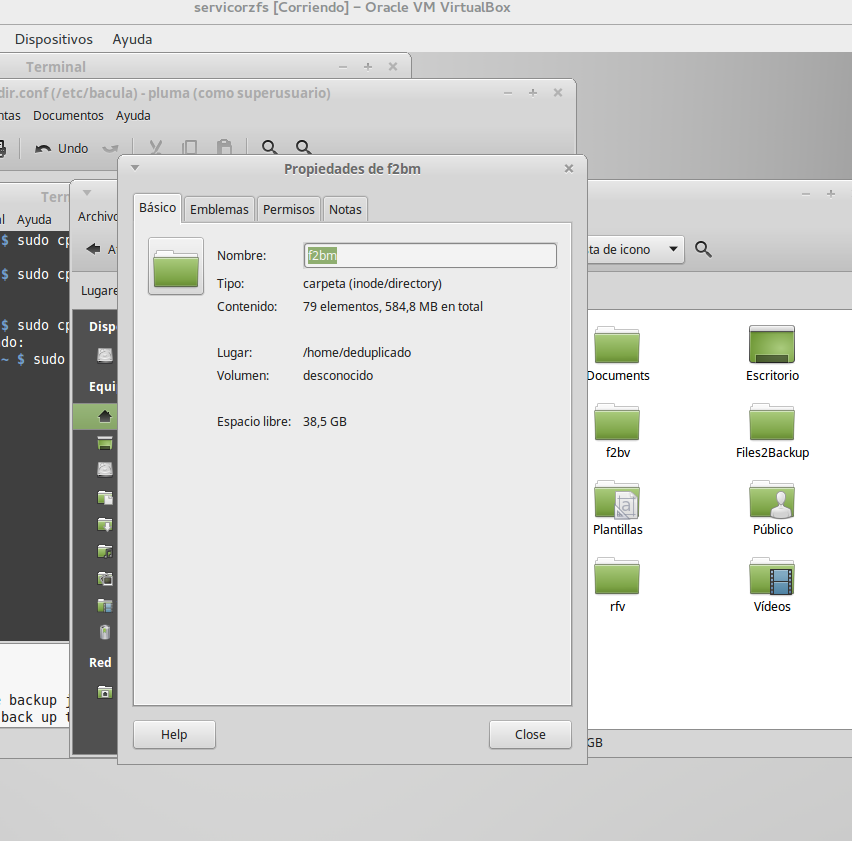
**ANEXO G.** PRUEBAS REALIZADAS DEL ANTES Y DESPUÉS CON LA DEDUPLICACIÓN ZFS EN ARCHIVOS DE IMAGEN, AUDIO Y VIDEO.

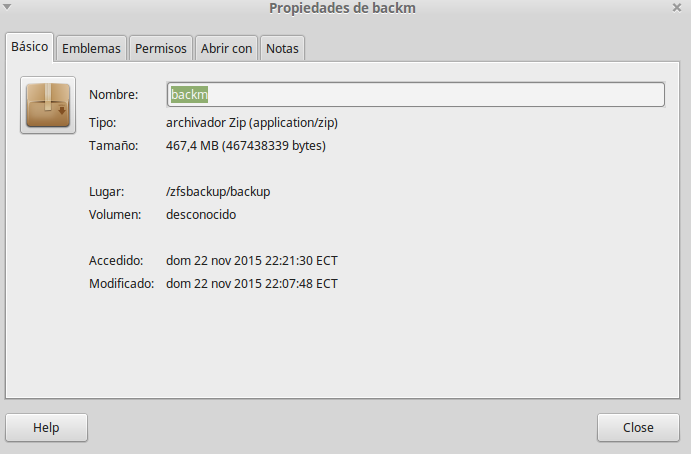
1. Archivos de imagen





1. Archivos de audio





1. Archivos de video

