



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**INFRAESTRUCTURA DE ALMACENAMIENTO TIPO NUBE PARA EL ACCESO  
A UN SISTEMA DE ARCHIVOS DE ALTO DESEMPEÑO DE UN CLUSTER**

**PROYECTO FINAL**

QUE PARA OPTAR POR EL GRADO DE:

**ESPECIALISTA EN CÓMPUTO DE ALTO RENDIMIENTO**

P R E S E N T A:

**ENRIQUE PALACIOS BONETA**

DIRECTOR DEL PROYECTO:

**DR. LUKAS NELLEN FILLA**  
INSTITUTO DE CIENCIAS NUCLEARES, UNAM

CIUDAD UNIVERSITARIA, CD. MX., JULIO 2018.



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



## **Agradecimientos**

Quiero agradecer al Dr. Lukas Nellen Filla por su asesoría en la realización de este proyecto, de igual forma a los miembros del jurado, a la Dra. María Elena Lárraga Ramírez y al Mtro. Juan Luciano Díaz González por su disponibilidad y por enriquecer este trabajo de investigación con su experiencia y comentarios.

A la Universidad Nacional Autónoma de México, por brindarme la oportunidad de seguir desarrollándome académica y profesionalmente, también se agradece a todos los profesores que han compartido su experiencia y que han sido un gran aliciente en mi formación.

Al Instituto de Ciencias Nucleares como a la Secretaría Académica por las recomendaciones y apoyo, así como a la unidad de cómputo y seguridad informática conformada por el Fis. Antonio Ramírez Fernández, Ing. Juan Eduardo Murrieta León y Martín Cruz Villafañe por el soporte y facilidades de infraestructura para realizar este trabajo.

Se agradece al proyecto PAPIIT IN112218.

A mis padres Marcela y Sergio, que gracias a su apoyo, cariño y educación, he podido seguir superándome.

Quiero agradecer de manera infinita a Yazmin Villegas, por su inigualable apoyo y comprensión que han sido fundamentales para el término de este proyecto y para mi desarrollo profesional. Te quiero mucho.

Finalmente, a los amigos que directa e indirectamente contribuyeron para lograr este trabajo.



---

## Resumen

En la actualidad el uso de almacenamiento e intercambio de archivos en línea ha aumentado, pues ofrece múltiples beneficios para los usuarios y grandes corporativos, pero a su vez implica mayores retos técnicos para la infraestructura ya que se requiere garantizar ciertas características para su buen funcionamiento, como son: disponibilidad, sincronía, seguridad, privacidad, velocidad, tolerancia a fallos, escalabilidad, heterogeneidad, entre otros.

El almacenamiento en línea se ha incrementado considerablemente debido a la alta disponibilidad de datos y colaboración en tiempo real, dichos beneficios hacen posible que usuarios, escuelas, empresas y centros de investigación cuenten con un acceso flexible a la información desde cualquier dispositivo y ubicación.

El uso compartido de archivos en línea reúne varios aspectos y diferentes características o aplicaciones, como son los sistemas distribuidos, sistemas orientados al almacenamiento, sistemas de alta disponibilidad y tolerancia a fallos, así como los sitios web de intercambio de archivos, sistemas de archivos compartidos, edición y colaboración, entre otros.

En el presente trabajo se explicarán los aspectos y características del uso compartido de archivos y los requerimientos a considerar para implementar dicha tecnología de forma segura y robusta, enfocándonos en el desempeño para lograr una infraestructura de almacenamiento compartido con sistemas de alto rendimiento, para aprovechar los servicios que en la actualidad se requieren, y que demandan un acceso fácil y amigable a los datos científicos de los cluster y equipos de alto rendimiento para mejorar el análisis, la colaboración y la productividad.

Se analizará la evolución de la nube y sus características, para entender la importancia con la que hoy en día cuenta. También se describirán los sistemas de almacenamiento de alto rendimiento y posteriormente se describirá la forma de uso de los servicios de la nube en estos sistemas, tal como son los servicios de intercambio y sincronía de archivos para el almacenamiento de alto rendimiento y alta disponibilidad.

Finalmente se menciona el caso de éxito académico de gran importancia y escala, basado en una solución de código abierto **ownCloud** llamado CERNbox del CERN, donde se analiza la arquitectura y las características del proyecto.

La recopilación de información en este trabajo de investigación tiene como objetivo implementar -a partir de pruebas- un prototipo amigable y transparente a los almacenamientos de los clusters y repositorios de datos del Instituto de Ciencias Nucleares de la UNAM.



# Índice general

<b>Resumen</b>	<b>V</b>
<b>Índice de contenido</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Importancia de los datos y la información . . . . .	1
1.2. Evolución de la nube . . . . .	3
1.3. Almacenamiento en la nube . . . . .	7
1.4. Modelos de almacenamiento Virtual . . . . .	9
1.4.1. Almacenamiento de instancias . . . . .	10
1.4.2. Almacenamiento de volumen . . . . .	10
1.4.3. Almacenamiento de objetos . . . . .	11
<b>2. Servicios del almacenamiento en la nube</b>	<b>13</b>
2.1. Microsoft OneDrive . . . . .	15
2.2. Dropbox . . . . .	15
2.3. Box . . . . .	15



---

2.4.	ownCloud . . . . .	16
2.5.	Intercambio y sincronización de archivos . . . . .	17
2.6.	La nube en sistemas de alto desempeño . . . . .	18
<b>3.</b>	<b>Sistemas de alto desempeño (HPC)</b>	<b>21</b>
3.1.	Sistemas de archivos para sistemas HPC . . . . .	22
3.1.1.	Interfaz estándar del sistema operativo POSIX . . . . .	23
3.1.2.	Sistema de archivos en espacio de usuario FUSE . . . . .	24
3.2.	Sistema de archivos distribuidos . . . . .	24
3.2.1.	Sistema de archivos distribuidos Lustre . . . . .	27
3.2.2.	Sistema de archivos distribuidos OrangeFS (PVFS2) . . . . .	29
3.2.3.	Sistema de archivos distribuidos EOS . . . . .	29
<b>4.</b>	<b>CERNBox</b>	<b>33</b>
4.1.	Implementación CERNBox con ownCloud . . . . .	35
<b>5.</b>	<b>Prototipo ownCloud ICN</b>	<b>37</b>
5.1.	Antecedentes . . . . .	37
5.2.	Objetivos . . . . .	38
5.3.	Instalación . . . . .	39
5.3.1.	Disco principal ZFS . . . . .	40
5.3.2.	QEMU . . . . .	42
5.3.3.	Red QEMU para las máquinas virtuales . . . . .	43

<i>ÍNDICE GENERAL</i>	<b>IX</b>
5.3.4. Configuración de las máquinas virtuales . . . . .	45
5.3.5. Instalación y configuración del Servidor NFS . . . . .	46
5.3.6. Instalación de la aplicación ownCloud . . . . .	47
<b>6. Conclusiones</b>	<b>51</b>
6.1. Proyectos a futuro . . . . .	52
<b>Bibliografía</b>	<b>54</b>

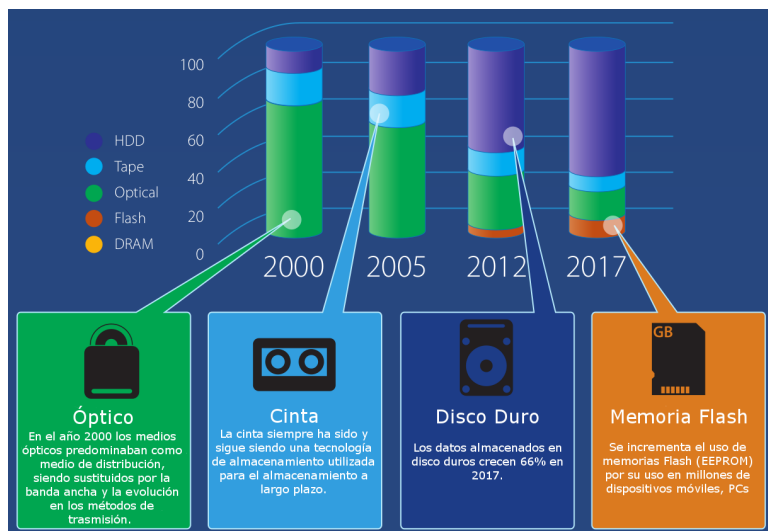


# Capítulo 1

## Introducción

### 1.1. Importancia de los datos y la información

A medida que el mundo digital continúa generando cantidades ilimitadas de datos, la tecnología de almacenamiento tiene el reto de entender qué información se debe de almacenar y en qué medio o tecnología es mejor. Recientemente se han realizado diversas investigaciones para predecir la capacidad de almacenamiento requerida a futuro, por ejemplo, la Corporación Internacional de Datos o International Data Corporation (IDC) publicó una investigación<sup>1</sup> en la cual estimó que la capacidad de almacenamiento de datos sin procesar, es decir, la densidad de bytes (raw storage capacity), se incrementó de 2.596 Exabytes<sup>2</sup> en 2012 a 7.235 EB en 2017 [1].



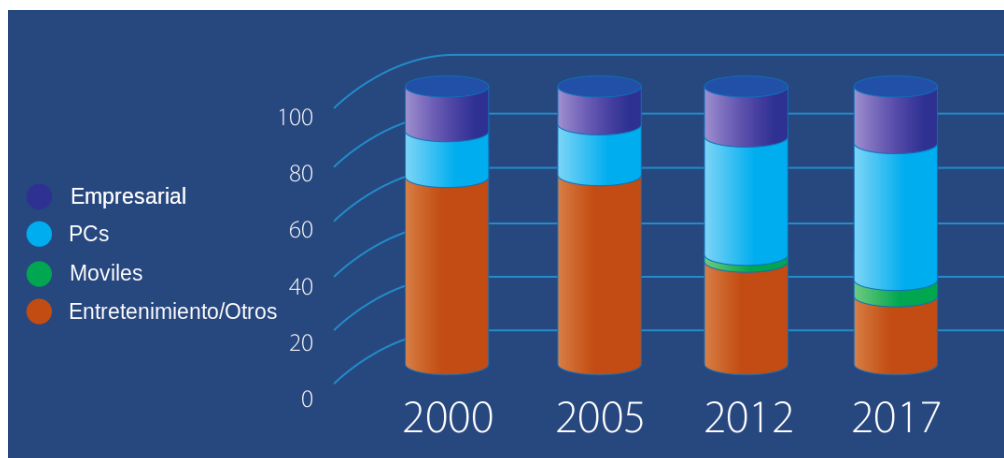
**Figura 1.1** Porcentaje anual de uso de dispositivos para almacenamiento

<sup>1</sup>[https://www.idc.com/downloads/where\\_is\\_storage\\_infographic\\_243338.pdf](https://www.idc.com/downloads/where_is_storage_infographic_243338.pdf)

<sup>2</sup><http://searchstorage.techtarget.com/definition/exabyte>

En una investigación más reciente se concluye que el universo digital se mantiene en constante expansión, al afirmar que del 2005 al 2020, el universo digital crecerá en un factor de 300, es decir, de 130 exabytes a 40,000 exabytes o 40 billones de gigabytes (más de 5,200 gigabytes por cada hombre, mujer y niño en 2020). A partir de hoy y hasta el 2020 se calcula que, el universo digital se duplicará cada dos años.[2] En dicha investigación se evalúan y estiman varios aspectos relacionados con las tecnologías para el almacenamiento de toda esta información y los clasifica en cinco tipos principales de dispositivos, tal como se muestra en la figura 1.1: discos duros, ópticos, de cinta, flash (NAND) y DRAM.

A través de este estudio, IDC también analizó el tipo de dispositivo en el que está almacenada la información, es decir, soluciones empresariales, PC, dispositivos móviles y otras categorías para ver qué áreas tendrán un crecimiento considerable a futuro (ver fig.1.2). Esta información ayuda a las empresas y centros de desarrollo de tecnología a impulsar decisiones altamente críticas basadas en datos en las tendencias.



**Figura 1.2** Dispositivos de almacenamiento

Por otra parte, el incremento de la velocidad en las telecomunicaciones ha facilitado el uso y consumo de formatos digitales (imágenes, documentos y videos), por lo que hoy en día ya es una necesidad tener toda la información a la mano de una forma sencilla y rápida, por tal motivo los fabricantes de dispositivos móviles desarrollan equipos más potentes, pero con capacidades limitadas de almacenamiento.

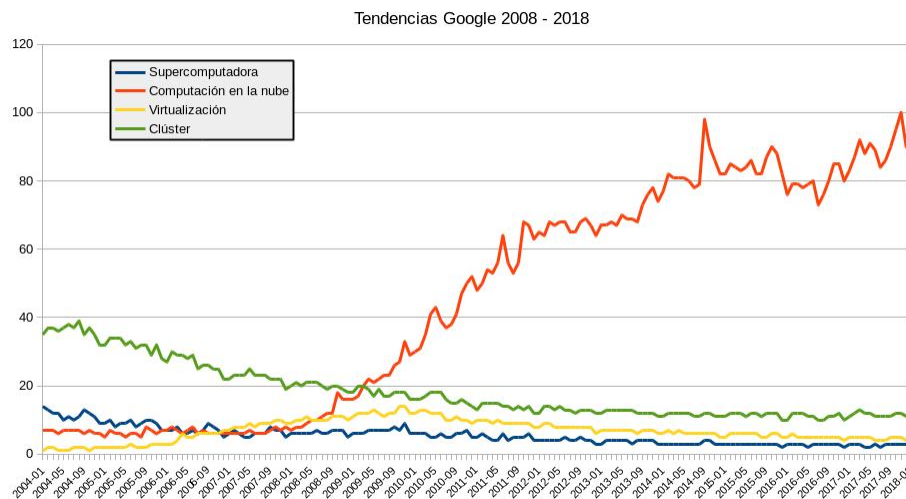
Por tal motivo las formas de almacenar la información digital han tenido que adaptarse para satisfacer las necesidades de los usuarios. Dicha evolución se ha basado en modelos distribuidos, pero que se han usado en computación de forma local (Almacenamiento como Servicio o Storage as Service), pero al contar con mejores comunicaciones, se ha podido incrementar su uso de forma remota, convirtiendo el almacenamiento en un modelo de negocio, por ejemplo, si una empresa provee el servicio por medio de Internet, puede permitirse reducir los costos a sus usuarios u ofrecerlo de forma gratuita pero limitado. Estos servicios se fueron comercializando y formalizando paulatinamente, a tal grado que en la actualidad se le conoce como almacenamiento en la nube [3], concepto originado por el hecho de visualizar el Internet como una nube de servidores, cables y servicios accesibles en todo momento.

## 1.2. Evolución de la nube

El surgimiento de nuevas tecnologías nos hace cuestionarnos sobre ¿cuándo se inventó? ¿cuándo se mencionó por primera vez? e incluso reflexionar sobre ¿qué perspectivas se tienen a futuro?

Cuando pensamos en la computación en la nube, creemos que es un producto e idea que surgió en el siglo XXI (ie. año 2000 en adelante) pero en realidad dicho concepto ha existido casi desde los inicios de la computación, por ejemplo, el concepto “Mainframe” comenzó en la década de 1950, donde múltiples usuarios tenían acceso a una computadora central a través de terminales, idea principalmente originada por el costo y mantenimiento de los equipos en esa época. Después de un tiempo, la tecnología en cómputo evolucionó y por alrededor de los años 1970 se presentaron los primeros conceptos de máquinas virtuales, con lo cual fue posible ejecutar uno o más sistemas operativos al mismo tiempo dentro de un hardware físico.

Con la llegada de los años 1990, las telecomunicaciones empezaron a tener gran importancia al ofrecer conexiones en redes, dejando atrás las conexiones punto a punto, facilitando la comunicación a un costo mucho menor al poder compartir el medio de conexión. El término “computación en la nube” fue ideado por el profesor de la Universidad de Texas, Ramnath Chellappa, quien definió en una conferencia un nuevo paradigma (“A new computing paradigm”) en 1997 [4][5].



**Figura 1.3** Los números representan el interés de búsqueda en relación con el valor máximo del gráfico. El valor 100 indica la popularidad máxima para el término, 50 implica la mitad y cero equivale a una popularidad menor del 1 %.

Por todo lo anterior, existen varias definiciones sobre el cómputo en la nube, por lo que resulta complicado encontrar una definición exacta [6]. En ocasiones se define como una versión fácil del uso del Cómputo Grid [7], o como un medio para tener acceso a recursos o servicios necesarios para realizar tareas con necesidades dinámicas[8] o sistemas donde convergen los términos Cómputo Grid, Cómputo de uso público a demanda (Utility Computing), Virtualización, Clustering, entre otros. Justo esta diversidad

de conceptos tecnológicos son los que dificultan su definición y por ello, se han originado diversos conceptos que dependen de su uso o aplicación. Gran parte de esta confusión en la mezcla de conceptos se debe en gran medida a los cambios de la tecnología y sus implementaciones. Es muy probable que, si se le pregunta a 10 expertos sobre qué es la computación en la nube, es posible que se obtengan 10 respuestas distintas.

Considerando las búsquedas en Internet como indicadores de interés y aceptación de las personas a ciertos temas, palabras o conceptos, este año será un gran año para la computación en la nube, ya que una búsqueda en tendencias de Google revela que a partir del 2009 el volumen de búsquedas de “computación en la nube” comenzó a superar las búsquedas de temas más comunes como Virtualización, Clustes y Supercomputadora, tal como se muestra en la figura 1.3.

El término cómputo en la nube (cloud computing) probablemente proviene (al menos en parte) del uso de una imagen de la nube para representar Internet o algún entorno de red grande. No importando mucho lo que hay en ella o lo que sucede allí, excepto que dependemos de enviar y recibir datos de manera confiable y segura. En tiempos más recientes, se asocia a una forma abstracta donde en lugar de visualizar servidores, switches o procesadores, ahora se visualizan servicios. Aunque aún se tengan las capas internas donde el hardware y el software de las redes existen, ahora hay capas superiores de servicios que se utilizan para crear aplicaciones, y detrás de esos servicios están los datos y los recursos de cómputo. Por lo que a un usuario del servicio no necesariamente le importa cómo se implementa, qué tecnologías se usan o cómo se administra, sólo le importa que haya acceso a él y tener un nivel de confiabilidad necesario para cumplir con los requisitos de la aplicación.

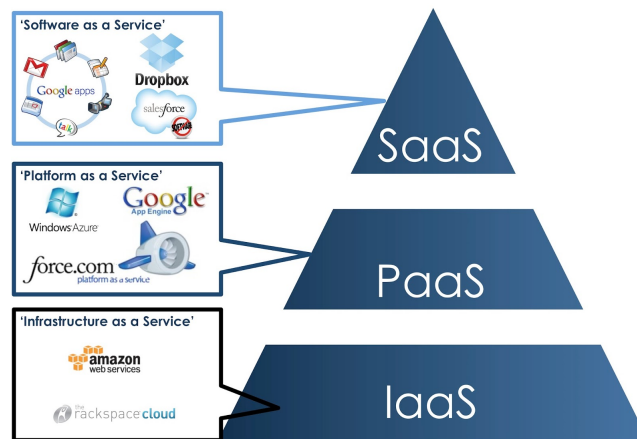


**Figura 1.4** Diseño esquemático tradicional de la nube

Como se mencionó anteriormente, esta idea contempla varios conceptos tecnológicos, por ejemplo, el modelo se puede definir como **computación distribuida**, es decir, una aplicación se crea utilizando el recurso de múltiples servicios desde múltiples ubicaciones, donde sólo se necesita conocer el portal para acceder a los servicios en lugar de que la nube le proporcione recursos disponibles, y que es **Software como Servicio (SaaS)**. Normalmente detrás de la interfaz del servicio suele haber una red de computadoras

(**cómputo en grid**) que proporcionan los recursos y que es distribuida por un proveedor (Fig.1.4). Y una vez que los usuarios paguen por los servicios y los recursos utilizados, eso es cómputo de uso público a demanda (**utility computing**).

El éxito de la nube se debe a la diversidad de servicios y aplicaciones, por lo cual no se puede considerar como un producto específico; ya que ofrece servicios de cómputo dependiendo de las necesidades y la demanda. Actualmente, existen tres modelos de implementación de la nube (Fig.1.5), los cuales son: Infraestructura como servicio (IaaS), Plataforma como servicio (PaaS) y Software como servicio (SaaS), siendo el almacenamiento parte fundamental y característica compartida, por lo que da pie al surgimiento de una posibilidad de un servicio más de la nube, que es el almacenamiento en la nube o Cloud Storage[9].



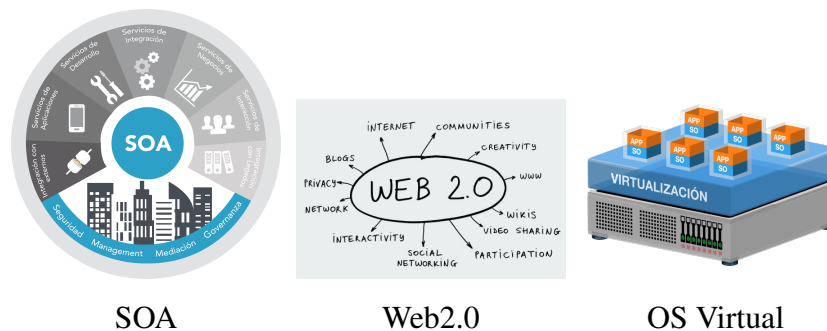
**Figura 1.5** Tecnologías de la nube

- **Software como servicio o SaaS (Software as a Service):** Es el modelo de entrega de servicios más utilizado, que proporciona software o aplicaciones según requiera el usuario de Internet. El beneficio de SaaS es que el usuario final no tiene que comprar el software o pagar la licencia, sólo paga por el servicio que consume, reduciendo los costos e incluso ayuda a los clientes a evitar la instalación del software en sus propias máquinas. Al software y a la aplicación se accede a través de un navegador web o cualquier otra interfaz de cliente ligero, por ejemplo, Google Apps, MS-Office 365, Dropbox, Adobe Creative, Salesforce, entre otros.
- **Plataformas como servicio o PaaS (Platform as a Service):** A menudo se denomina plataforma de desarrollo o de producción para los sistemas SaaS, con la que se facilita el entorno para desarrollar, probar e implementar aplicaciones sin tener ningún control sobre el sistema operativo y la infraestructura de hardware, algunos ejemplos son: Amazon Web Service Elastic Beanstalk, Windows Azure, Heroku, Google App Engine, force.com, Red Hat Open Shift.
- **Infraestructura como servicio o IaaS (Infrastructure as a Service):** Los clientes tienen un mayor control sobre los recursos de hardware. Los proveedores de IaaS proporcionan la infraestructura como recursos de red, servidores y espacio de almacenamiento y/o uso de CPU bajo demanda y en



alquiler o pago por uso. Los usuarios de IaaS tienen control total sobre la plataforma y el software que está desplegado en la infraestructura, por ejemplo, Amazon EC2, Windows Azure, Rackspace, Google Compute Engine.

La evolución de la tecnología y la definición de los estándares ha hecho que la nube tenga un mayor impacto en la infraestructura de los centros de datos, transformando las formas de acceso a los servicios y datos empresariales o científicos, proporcionando acceso simple a través de Internet, y permitiendo a los usuarios pensar en la informática como “ilimitada”, de costo mínimo y confiable, evitando la necesidad de saber cómo se construye, cómo funciona o dónde se encuentra. Dicha evolución se ha logrado por la definición de estándares de tecnologías como virtualización, arquitectura orientada a servicios o Service Oriented Architecture (SOA) y Web 2.0[10] (Fig.??) que han servido de base para que en conjunto la nube y sus servicios sean funcionales. Una arquitectura típica de cómputo en la nube se muestra en la figura 1.4, donde se observa que los usuarios sólo pueden tener interacción con el sistema por medio de interfaces o aplicaciones, por ejemplo, un navegador web donde internamente el sistema se encarga de hacer todo el trabajo.



**Figura 1.6** Estándares que son base tecnológica para la nube

Existen otros tipos de Nubes que no dependen tanto de la parte tecnológica, pues se clasifican de acuerdo a la forma de uso o administración, tal como, la nube privada, pública e híbrida (Figura ??). Una nube privada se basa en un conjunto de recursos compartidos dentro de una organización o centro de datos, cuyo acceso está limitado. Se accede a los recursos por medio de una red privada y segura, donde toda la información es propiedad de la organización.

A diferencia del tipo privado, en una nube pública el acceso es por medio de Internet y menos restringido e incluso se tienen diversas políticas sobre los datos, servicios o aplicaciones compartidas que dependen del proveedor. El modelo híbrido, comparte recursos o servicios de ambos tipos, lo que facilita la adaptación a las diferentes necesidades de la mayoría de las empresa o centros de investigación.

La facilidad de uso y la reducción de costos, es lo que ha generado gran interés en la actualidad, pues permite oportunidades para una colaboración, integración y análisis al tener los servicios y datos en una plataforma compartida.

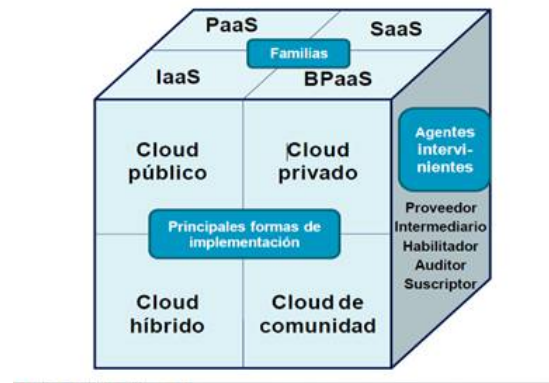


Figura 1.7 Modelos y tipos de la nube

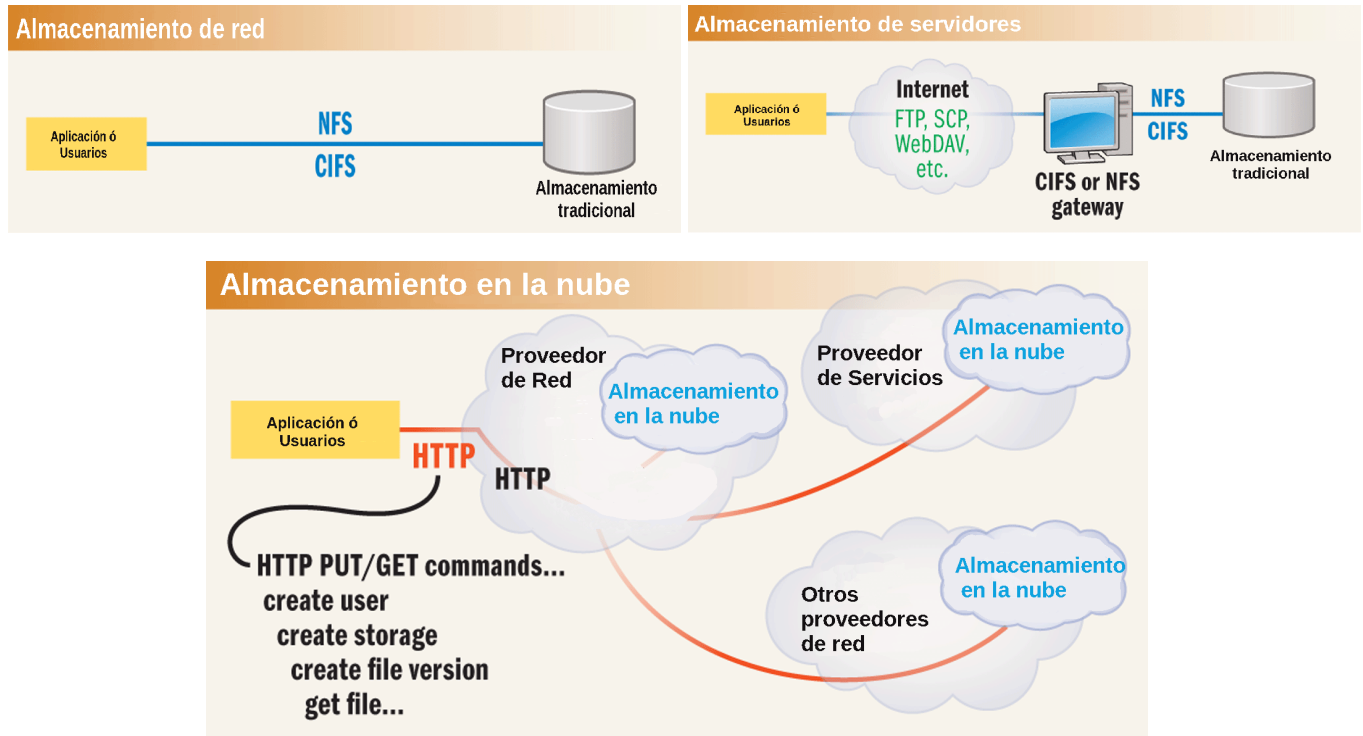
### 1.3. Almacenamiento en la nube

Con la evolución de las telecomunicaciones y su mejoramiento en seguridad (conexiones SSL y TLS) y velocidad (redes de 10Gb y 100Gb, etc.), el concepto de almacenamiento en la nube surge como opción viable para los usuarios, siendo cada vez más común y popular el uso de algunos proveedores comerciales, tal como Dropbox, Box, Microsoft, AWS y Google que en la actualidad se han convertido en herramientas cotidianas para millones de personas. Hoy en día, estos proveedores ofrecen un gran espacio disponible con tres características importantes: acceso a través de API o servicios web, disponibilidad inmediata de grandes cantidades de almacenamiento y pago por lo que se use. Normalmente estos servicios tienen la capacidad de escalar rápidamente dependiendo de las características que necesite el usuario.

Como se ha mencionado, la nube puede relacionarse con infinidad de tecnologías, pero sin la **infraestructura de almacenamiento** ninguna de las otras aplicaciones es posible, por tal importancia y por la gran demanda, actualmente se considera ya como un servicio más de la nube llamado almacenamiento de datos como servicio o “Data Storage as a Service” (DaaS).

El crecimiento de los datos y la necesidad de mantenerlos seguros por más tiempo, ha hecho que los centros de datos se interesen cada vez más en las políticas de uso, así como en su acceso o disponibilidad, por ello, y al igual que otras tecnologías de la nube, ha evolucionado y sufrido cambios. En la figura ?? se muestra el desarrollo del almacenamiento compartido, el cual se realizaba inicialmente por medio de protocolos de red tradicional como NFS (Network File System), FTP (File Transfer Protocol) y CIFS (Common Internet File System), pero rápidamente evolucionó a los almacenamientos conectados a la red (Network attached storage o NAS) donde se perfeccionaron los protocolos, por ejemplo, NFS versión 4 que acepta los protocolos TCP y redes no persistentes por medio de múltiples conexiones, posteriormente evolucionaron a SAN o Storage Area Network en el cual se mejoraron las limitaciones anteriores, pues contaba con más protocolos (iSCSI) que permitían mayor cantidad de accesos, multiusuario, control de los medio de almacenamiento a bajo nivel y la posibilidad de independizar el almacenamiento de cualquier otro sistema. Todo esto abrió nuevas oportunidades en conectividad, escalabilidad, capacidad y políticas de administración.

Como en todas las tecnologías, la evolución y mejoras es constante, por ello, se perfeccionaron las



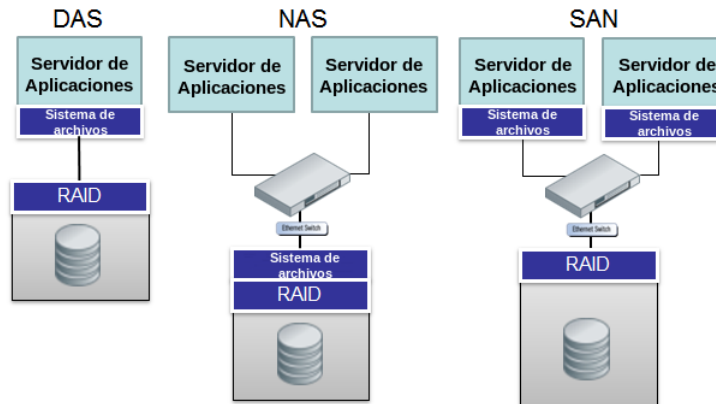
**Figura 1.8** Evolución del almacenamiento

formas de conexión al implementar más protocolos estándar como son iSCSI por bloques y WebDAV, conectados por enlaces tipo Fibre Channel (FC), Ethernet de alta velocidad e Infiniband al dividir el desarrollo en dos caminos, en sistemas de almacenamiento conectado directo o Direct Attached Storage (DAS) y los SAN, que comparten características, pero que a su vez resuelven diferentes tipos de problemas.

Al ser el modelo de almacenamiento más simple, el "Almacenamiento Conectado Directo" (DAS) ha tenido gran aceptación y desarrollo en solucionar los problemas internos que presentan los sistemas de cómputo de alto rendimiento o High Performance Computing (HPC), y a pesar de que se considera un sistema SAN grande y caro, ha sido por muchos años la tecnología más usada y desarrollada. Considerando lo anterior, en los últimos años, se han requerido soluciones más eficientes y de bajo costo, por ello los centros de datos y los desarrolladores de tecnología han fijado su atención en los SAN, pues tiene dispositivos de almacenamiento conectados a la red, basado en un arreglo de discos que proporcionan capacidad, protección y redundancia. En la figura 1.9 se muestran los esquemas de los diferentes sistemas de almacenamiento.

Como se mencionó anteriormente, la necesidad de reducir costos y facilitar el proceso de almacenamiento, forzó la instalación de software para la administración de estos sistemas de forma optimizada.

La tecnología de virtualización del almacenamiento que hace referencia a una nueva capa que separa el dispositivo físico y el volumen de almacenamiento lógico, es una de las mejoras que permitió reducir los costos, migrar volúmenes de almacenamiento en tiempo real o en vivo, movilidad de datos entre discos de diferente tecnología como los discos de estado sólido o SSD (rápidos) a disco estándar o HDD (lentos),



**Figura 1.9** Almacenamiento DAS, NAS, SAN

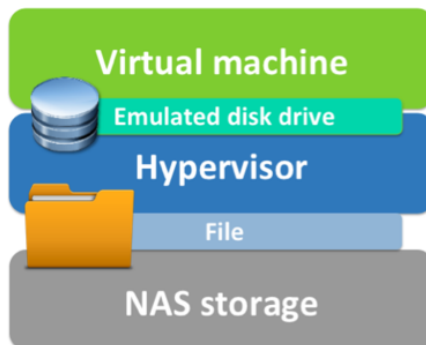
flexibilidad en la capacidad de almacenamiento. De igual forma permitió que la infraestructura de almacenamiento física se utilizara de manera óptima y eficiente, principalmente la posibilidad de abstraer los elementos de todo el sistema de almacenamiento como las unidades lógicas o Logical Unit (LUN) y los sistemas de archivos de máquinas virtuales o VMFS, entre otros, separando el modelo de almacenamiento y el protocolo de almacenamiento.

## 1.4. Modelos de almacenamiento Virtual

En la virtualización, surge el concepto del monitor de máquinas virtuales o hipervisor, el cual se considera como un proceso o software que interviene entre los sistemas huéspedes (máquinas virtuales con diferentes tipos de sistema operativo) y la computadora física y que proporciona un entorno de hardware emulado para cada máquina virtual (cpu, memoria RAM y almacenamiento), como se muestra en la figura 1.10. El hipervisor moderno, emula las unidades de disco físicas locales como una manera de proporcionar almacenamiento para cada máquina virtual, es decir, usa el modelo de unidad de disco local (DAS) para exponer el almacenamiento a las máquinas virtuales. Así como la unidad de almacenamiento en DAS es la máquina física, la unidad en el almacenamiento en disco virtual es la máquina virtual, por consiguiente, los discos virtuales no se exponen como objetos independientes, sino como parte de una máquina virtual en particular. Al igual que con DAS, un disco virtual vive y muere con la propia máquina virtual, por lo tanto, si se elimina la máquina virtual, también se eliminará el disco virtual.

El modelo de almacenamiento virtual con mayor interés es aquel donde el hipervisor simula el disco a la máquina virtual, y donde controla los accesos de entrada y salida (E/S) lo que lleva a una separación entre el modelo de almacenamiento que se expone a la máquina virtual y al administrador.

Estas capas permiten combinar modelos de almacenamiento y protocolos de almacenamiento, incluso se pueden cambiar dinámicamente el protocolo de almacenamiento sin afectar a las máquinas virtuales. La elección de la implementación es completamente transparente para la aplicación, ya que eventualmente todos estos protocolos serán de la misma forma para la máquina virtual, es decir, se detectan como unidades de disco físicas locales conectadas.



**Figura 1.10** Protocolos y modelo de almacenamiento

Toda esta evolución y desarrollos en la tecnología ha forzado a cambiar los centros de datos ya que los sistemas virtualizados son transformados en ambientes de nube, proporcionando modelos adicionales para habilitar varios sistemas de almacenamiento virtualizados (fig. 1.11) para que puedan proporcionar autoservicio y una separación entre la infraestructura y la aplicación.

### 1.4.1. Almacenamiento de instancias

El modelo de almacenamiento en disco virtual es el principal para el almacenamiento virtualizado. En entornos de nube, dicho modelo es uno de tres y por consiguiente, se le asigna un nombre específico en entornos de nube: almacenamiento de instancias.

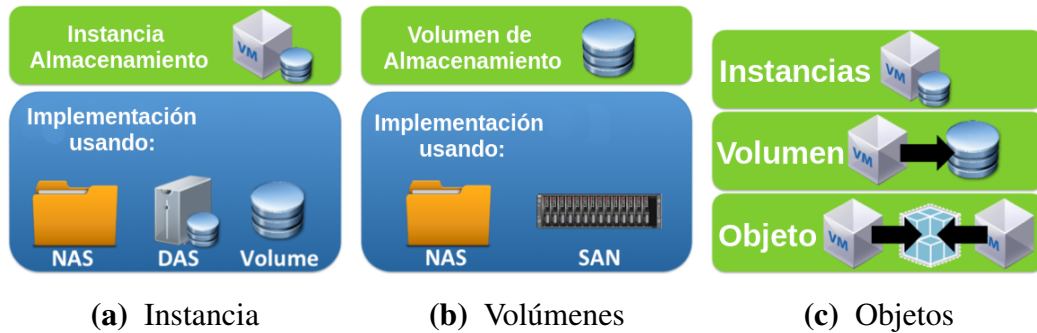
Es importante tener en cuenta que el **almacenamiento de instancias** es un modelo y no un protocolo de almacenamiento, que puede usarse de múltiples maneras, por ejemplo, ocasionalmente el almacenamiento de instancias se implementa utilizando DAS en los nodos de cómputo, llamado almacenamiento efímero debido a que generalmente no es altamente confiable.

El almacenamiento de instancia también se puede implementar como almacenamiento confiable al usar NAS o almacenamiento de volumen, mismo que es un segundo modelo de almacenamiento que se describe a continuación.

### 1.4.2. Almacenamiento de volumen

El almacenamiento de instancia tiene limitantes y su configuración es diferente entre sistema operativo y aplicaciones, por lo que es necesario otro tipo de almacenamiento, tal como el almacenamiento de volumen que pasa a ser un híbrido de almacenamiento de instancias y SAN.

Un volumen se puede desconectar de una máquina virtual y conectarse a otra, a diferencia del almacenamiento de instancias, donde el almacenamiento de volumen generalmente se supone que es altamente confiable y se usa a menudo para los datos del usuario.



**Figura 1.11** Tipos de almacenamiento en la nube

### 1.4.3. Almacenamiento de objetos

Las aplicaciones nativas de la nube también necesitan un espacio para compartir datos entre máquinas virtuales, pero frecuentemente requieren un espacio de nombres que puedan escalar a múltiples centros de datos en todas las regiones geográficas y es precisamente el almacenamiento de objetos quien lo proporciona. El almacenamiento de “objetos” proporciona una abstracción similar a un archivo, pero eventualmente no siempre cuenta con consistencia a todas las solicitudes, esto significa que, si bien todos los clientes obtendrán las mismas respuestas a sus solicitudes, pueden recibir respuestas diferentes temporalmente. Dicha consistencia es similar a la proporcionada por Dropbox entre dos computadoras; los clientes pueden desconectar la sincronía temporalmente, pero eventualmente todo convergerá.

Los almacenamientos de objetos tradicionales también proporcionan un conjunto simplificado de operaciones de datos ajustadas para conexiones de alta latencia:

- Enumeran los objetos en un “depósito”
- Leer un objeto en su totalidad
- Reemplazar los datos en un objeto con datos completamente nuevos.

Este modelo proporciona un conjunto de operaciones más básico que el NAS, pues, permite a las aplicaciones leer y escribir bloques pequeños dentro de un archivo, reducir y mover archivos entre directorios, y así sucesivamente.

Al no ser un modelo estricto permite que el almacenamiento de objetos proporcione espacios de nombres extremadamente grandes en distancias largas con bajo costo y con un buen rendimiento. Muchas aplicaciones diseñadas para la nube están diseñadas para usar almacenamiento de objetos en lugar de NAS, debido a su escala y costo. Por ejemplo, las aplicaciones nativas de la nube a menudo usan almacenamiento de objetos para almacenar imágenes, contenido estático de la WEB y conjuntos de datos analíticos.

Considerando lo anterior, se ha revisado la evolución del almacenamiento, desde la necesidad urgente de almacenar grandes cantidades de datos en la actualidad como la generación de tecnología para soportar dicho crecimiento. Para ello se analizó el desarrollo y la evolución de los principales sistemas de

almacenamiento, también la evolución de los paradigmas de computación como el cómputo en la nube y sus derivados, que aunado a las necesidades obligan a tener modelos, formas de trabajo y pensamiento diferentes basadas en estándares.

Ahora se tiene una imagen completa del surgimiento de los modelos de almacenamiento actuales para la nube y cómo se relacionan con los modelos de almacenamiento utilizados en entornos más convencionales. Del mismo modo que los modelos DAS, SAN y NAS proporcionan un conjunto de herramientas para abordar una variedad de casos de uso físico y virtual. La evolución a los tipos de modelos de instancia, volumen y el almacenamiento de objetos proporcionan un paradigma flexible para la nube. Si bien es posible que todas las instalaciones no utilicen los tres tipos de almacenamiento, también es un hecho que ninguno puede abordar por si solo todos los casos de uso necesario.

A continuación, se analizarán las aplicaciones y propiedades del almacenamiento en la nube como medio de trabajo para sistemas de alto rendimiento del ámbito científico.

## Capítulo 2

# Servicios del almacenamiento en la nube

El almacenamiento en la nube no es tan complicado como mucha gente piensa. De hecho, hoy en día es una de las maneras más seguras de almacenar, realizar copias de seguridad, compartir y acceder a la información que se requiere. Es decir, si existe información en la computadora de casa pero está fuera de ella, puede usar el almacenamiento en la nube y acceder a su información desde cualquier sistema, siempre y cuando fuera subida a la nube. Esto ha facilitado la vida digital actual, reduciendo distancias y eliminando la necesidad de estar en un lugar en específico para poder consultar la información. Todo esto no ha sido fácil ni rápido, como pudimos ver en el capítulo anterior, cada funcionalidad y servicio han sido resultado de varios años y desarrollo tecnológico.

Existen diversas formas de utilizar el almacenamiento en la nube, la primera opción que tiene que elegir en un sistema de almacenamiento en la nube privado, público o híbrido y con base en las características y necesidades del usuario, los servicios van adquiriendo mayor importancia. El principal uso es para copia de seguridad de cualquier información -películas, imágenes, videos, documentos, entre otros- pues facilita la restauración de los datos en los equipos personales de los usuarios.

Recientemente, esta forma de respaldo se ha hecho popular, ya que los usuarios requieren compartir datos con mayor frecuencia, rapidez y seguridad. A continuación, se mencionan algunos de los servicios o características con los que cuentan los principales servicios de almacenamiento en la nube.

- Cliente para sincronía y compatibilidad con sistemas operativos móviles.
- Acceso Web.
- Métodos para compartir (correo, URL, redes sociales, entre otros).
- Uso colaborativo en línea.
- Seguridad.
- Control de versiones.
- Revisión de virus y malware.
- Autenticación centralizada.



Los centros de investigación y empresas prefieren un sistema de almacenamiento en la nube debido al costo, capacidades de seguridad, controles de administración y características de productividad del usuario, así como su capacidad de integrarse fácilmente con otras herramientas y aplicaciones especializadas que ya se utilizan.

- **Costos:** El almacenamiento de datos en la nube es relativamente económico a mediano plazo ya que la inversión o mantenimiento de la infraestructura es alto. También ayuda a hacer copias de seguridad de los datos, ya que no se necesitan centros de datos adicionales, los datos se pueden replicar automáticamente de la nube pública a la privada. En algunos casos, donde el espacio de almacenamiento es grande o tiende a crecer, podrían llegar a igualar el costo por inversión de un almacenamiento local.
- **Seguridad de datos:** La nube puede proteger la información dependiendo de su implementación, siendo ésta una de las características más importante para las empresas, ya que la mayoría de su información es confidencial. En los sistemas actuales los datos están protegidos durante la transmisión con protocolos de conexión segura y existe la posibilidad de que el almacenamiento de igual forma sea cifrado. Otro de los aspectos relacionado a la seguridad de los datos, es la integridad, ya que como se mencionó en el capítulo anterior, los datos se almacenan en servidores que cuentan con sistemas y algoritmos que reducen la probabilidad de tener problemas mecánicos y/o lógicos garantizando la integridad.
- **Escalabilidad:** En esta época digital, todo almacenamiento tiene un costo, por lo que en general, es una característica importante para los usuarios, ya que la tendencia siempre se reducirlos costos e incrementar el espacio disponible. Por lo tanto, estos sistemas tienen implementado la forma de ajustar el espacio a demanda y sólo usar lo necesario. Y en el supuesto caso, si el usuario y/o la aplicación requiere más espacio, cuenta con la capacidad de agregar más. También es posible mejorar varios aspectos, como el rendimiento al realizar las lecturas o escrituras de forma paralela.
- **Accesibilidad:** Con el número cada vez mayor de dispositivos utilizados en la actualidad, se puede acceder al servicio de almacenamiento en la nube desde tabletas, teléfonos inteligentes, netbooks y computadoras de escritorio. El usuario puede acceder a la cuenta desde cualquier conexión a Internet, como el navegador móvil o el sistema de escritorio en el lugar de trabajo.
- **Sincronización:** Esto garantiza la actualización automática de datos y archivos guardados en todos los dispositivos. Por consiguiente, la última versión del archivo guardado en el escritorio estará disponible en el teléfono inteligente.
- **Colaboración:** El servicio de almacenamiento en la nube en línea permite que varias personas editen y trabajen en el mismo archivo o documento pues facilita el trabajo en grupo. Esta es una funcionalidad con la que algunos sistemas no cuentan y que hace la diferencia.
- **Automatización:** El servicio de almacenamiento en la nube en línea simplifica el proceso de copia de seguridad de datos mediante la automatización. El usuario sólo necesita seleccionar qué y cuándo necesita la copia de seguridad para que el servicio haga el resto del trabajo.

Existen varios proveedores de servicios de almacenamiento en la nube, como ownCloud, UbuntuOne, Google Drive, Microsoft OneDrive, Dropbox, entre otros; la mayoría ofrece servicios básicos de forma gratuita, por lo que los usuarios y empresas pueden probar el servicio antes de contratar opciones más especializadas. A continuación, veremos algunas características de los servicios de almacenamiento más usados hoy en día.

## 2.1. Microsoft OneDrive

Almacenamiento de Microsoft. Aquellos que usan Windows 8 y 10 tienen integrado OneDrive en su sistema operativo. Sin embargo, cualquiera puede usarlo en la Web, descargando una aplicación de escritorio para Mac y versiones anteriores de Windows, o las aplicaciones OneDrive para Android, iOS, Windows Phone y Xbox. Puede almacenar cualquier tipo de archivo en el servicio, incluidas fotos, videos y documentos, y luego acceder a ellos desde cualquier PC con Windows o dispositivo móvil. El servicio organiza de forma automática los archivos, por lo que es fácil encontrar lo que necesita. La mayor fortaleza de OneDrive es que funciona con las aplicaciones de Microsoft Office, como Word o PowerPoint, por lo que al ejecutar una de esas aplicaciones, se podrá tener acceso a la lista de documentos recientes guardados en OneDrive. Algo a considerar de éste almacenamiento es que sólo se puede colaborar en tiempo real entre usuarios de Office 365.

## 2.2. Dropbox

Dropbox es uno de los favoritos de los usuarios porque es confiable, fácil de usar y configurar pues, los archivos viven en la nube y se puede acceder a ellos en cualquier momento desde el sitio web de Dropbox, aplicaciones de escritorio para Mac, Windows y Linux (Ubuntu, Debian, Fedora o Gentoo) o iOS y Android.

Esta aplicación se ejecuta en el sistema operativo por medio de una carpeta y comparte los archivos de la computadora a la nube y viceversa. El servicio sincroniza automáticamente tus archivos en todos tus dispositivos para que puedas acceder a todo en cualquier lugar. No hay límite de tamaño en los archivos que carga a Dropbox con el escritorio o las aplicaciones móviles, pero los archivos más grandes pueden tardar varias horas en cargarse, dependiendo de la velocidad de conexión. No cuenta con colaboración en tiempo real.

## 2.3. Box

Le permite compartir archivos con colegas, asignar tareas, dejar comentarios en el trabajo de alguien y recibir notificaciones cuando un archivo cambia. Puede obtener una vista previa de los archivos desde el sitio web de Box e incluso crear documentos de texto básicos. Al igual que otros servicios de almacenamiento en la nube, puede descargar una aplicación de escritorio y sincronizar sus archivos entre el disco duro y la nube. Box también ofrece control sobre la privacidad de los archivos, ya que se puede decidir

quién puede ver y abrir carpetas y archivos específicos, así como quién puede editar y cargar documentos. Incluso se puede proteger con contraseña los archivos individuales y establecer fechas de vencimiento para las carpetas compartidas. Los usuarios empresariales también pueden conectar otras aplicaciones, como Salesforce y NetSuite, para guardar fácilmente documentos en Box e incluso hay complementos para Microsoft Office y Adobe Lightroom que permiten abrir y editar archivos guardados en Box desde esas aplicaciones.

## 2.4. ownCloud

Es una plataforma de intercambio y sincronización de archivos de código abierto que proporciona acceso fácil a los datos a través de una interfaz web e incluso permite la sincronización de clientes. Proporciona una plataforma para ver, compartir y sincronizar datos de manera segura en todos los dispositivos, mientras mantiene todo bajo el control total del usuario. Su interfaz de código abierto hace que la plataforma sea adaptable para aplicaciones y complementos (plugins), lo que permite trabajar con cualquier tipo de almacenamiento. Los usuarios pueden almacenar de forma segura sus datos, incluidos documentos, imágenes, contactos, calendario ya sea en los servidores del proyecto, así como de forma local en servidores existentes. También permite acceder a unidad FTP locales, Dropbox, NAS, entre otros. El cliente de escritorio permite sincronizar archivos en el hogar al mantener las carpetas seleccionadas sincronizadas entre los dispositivos. Los usuarios pueden enviar enlaces públicos protegidos por contraseña a otras personas, lo que les permite cargar archivos, de igual forma, notifica a los usuarios cuando se realiza el intercambio de archivos. Sus funciones de colaboración permiten a los usuarios editar documentos a la vez, o comentar en imágenes y realizar videollamadas.

ownCloud funciona en la mayoría de los sistemas operativos como Windows, Mac, Linux, así como aplicaciones nativas para Android e iOS, lo que lo convierte en una solución ideal para ser utilizado en cualquier dispositivo. Las aplicaciones y los datos se pueden cifrar en el almacenamiento local y en la nube, sin que las claves salgan del propio servidor de Cloud. Sin embargo, las claves también pueden ser administradas por una herramienta externa de administración de claves si es necesario. Todas las transferencias de datos se cifran con HTTPS, según el estándar WebDAV.

Los documentos se pueden editar en tiempo real, mientras que los usuarios también pueden leer archivos PDF, ver videos, ver imágenes y mucho más. La edición colaborativa permite que hasta cinco personas trabajen en archivos de documentos desde el navegador. El software viene con actualizaciones de actividad y notificaciones de actividad relacionada con los datos.

El software del servidor ownCloud es principalmente código PHP diseñado para instalarse en un servidor web. El servidor utiliza una base de datos para la persistencia de metadatos y ofrece al administrador la opción de utilizar una base de datos SQLite para instalaciones más pequeñas y MySQL para instalaciones más grandes. De forma predeterminada, los archivos almacenados en la instancia de ownCloud se almacenan relativamente sin modificaciones en el sistema de archivos del sistema operativo del servidor en un subdirectorío de los archivos de la aplicación ownCloud. Las funciones avanzadas de almacenamiento asociadas con el almacenamiento en la nube, como la agrupación de archivos para redundancia y escalabilidad, no son gestionadas internamente por ownCloud. Dichas características deberían implementarse a nivel del sistema operativo (utilizando un producto como GlusterFS, XtremFS o ZFS).

El software del servidor se puede ampliar instalando / habilitando “aplicaciones” (tanto internas como de terceros), que pueden agregar características de encriptación del lado del servidor, integración con otros servicios / proveedores de almacenamiento en la nube y sistemas de autenticación adicionales como LDAP.

El software de cliente ownCloud consiste en una interfaz web y varias aplicaciones de cliente. Más allá de las capacidades estándar de carga, descarga, eliminación, cambio de nombre, y de archivos, la interfaz web también permite al usuario reproducir archivos multimedia, ver imágenes en una galería y mantener una lista de contactos y un calendario.

Como se pudo analizar, los servicios que actualmente se ofrecen cubren casi las mismas necesidades o comparten algunas características, pero justo esas pequeñas diferencias crean una diversidad de aplicaciones o soluciones a diferentes proyectos o tipos de uso, es decir, donde unos sistemas resuelven todo el problema y sólo el usuario y administrador tienen que concentrarse en su uso, en otros sistemas permiten una adaptabilidad completa para poderse modificar o crear, dependiendo de las necesidades específicas de algunos usuarios. Otra de las características importantes son los costos, ya sea por su uso o por la infraestructura que se requiera en la implementación local.

Uno de los objetivos de este trabajo es reunir información para poder conocer a detalle estos sistemas y poder elegir el mejor. En los siguientes capítulos se describirá la implementación del sistema que cumple con las características que se requieren dependiendo del proyecto o servicio deseado.

## 2.5. Intercambio y sincronización de archivos

El intercambio y sincronización de archivos basada en la nube, permite a los usuarios un acceso útil y rápido a diferencia de los archivos guardados localmente, que sólo están disponibles en la máquina en la que se guardan, por ello, la sincronía mejora la colaboración y el trabajo en grupo.

Existe confusión en estos dos servicios, principalmente originado por los proveedores, ya que “sincronizar y compartir archivos” es un servicio extra que comúnmente en la mayoría de los servicios de almacenamiento se ofrece, y principalmente se refiere a la administración y manipulación de archivos que se almacenan en la nube. Es preciso realizar dicha distinción porque el intercambio de archivos en la nube está limitado a los archivos, mientras que el almacenamiento en la nube no.

La aplicación de sincronización de archivos es una arquitectura cliente/servidor y se encuentra frente al almacenamiento en la nube privada (DAS, SAN, NAS y almacenamiento de objetos) o el almacenamiento en la nube pública.

Como se ha visto, el almacenamiento en la nube no es una aplicación y no requiere que de un servidor que ejecuta alguna aplicación para realizar su función, sólo es un depósito de almacenamiento de datos como cualquier otro de almacenamiento.

En la actualidad, aunque es un servicio ajeno al almacenamiento, es uno de los servicios más útiles en el concepto nube ya que se pueden sincronizar automáticamente los archivos entre ubicaciones y dispositivos. Por ejemplo, digamos que se tiene algunos datos experimentales almacenados en uno de los recursos compartidos de red, pero desea almacenarlos localmente en una computadora portátil para un análisis externo, en la mayoría de los servicios de sincronización, simplemente se da de alta la carpeta de datos de red a la lista de carpetas para sincronizar con la aplicación y se copiarán los datos de la red a la computadora portátil. Posteriormente, si se cambian los archivos o se crean otros nuevos durante el análisis, se copiarán

de nuevo en el recurso compartido de red cuando se vuelva a conectar.

La idea se atribuye a Drew Houston, el fundador de Dropbox, quien dice haberlo inventado después de que se dio cuenta en un viaje en autobús entre Nueva York y Boston de que había dejado atrás su dispositivo USB. [11]

El mecanismo básico detrás de la concepción de Drew Houston y el mecanismo básico utilizado por la mayoría de los servicios de almacenamiento en la nube que ofrecen sincronización hasta el día es una “carpeta de sincronización” que se ve y se comporta como cualquier otra carpeta en su disco duro, excepto que los archivos colocados en esa carpeta están almacenados en su disco y en la nube.

## 2.6. La nube en sistemas de alto desempeño

La nube se está convirtiendo en la piedra angular de la computación moderna, comercial y científica, debido a que agrega funcionalidad al sistema de archivos paralelos a la computación en la nube y surgen más opciones de HPC como servicio en la nube.

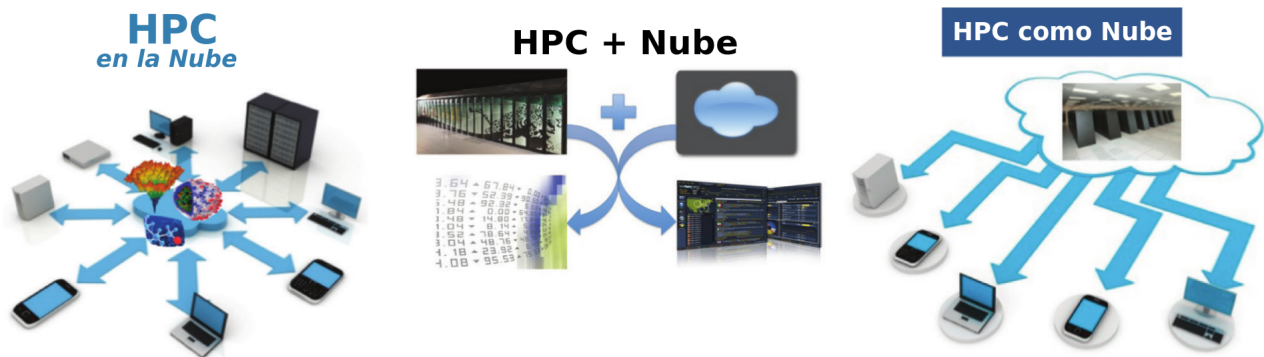
A finales de la década de 2000, la computación en la nube estaba aumentando rápidamente su nivel de madurez y popularidad[12], y por lo mismo, comenzaron a surgir estudios sobre la viabilidad de ejecutar aplicaciones HPC en recursos remotos de la nube. [13] Esta migración no ha sido fácil ya que las aplicaciones HPC tiene características diferente a las tradicionales en la nube, ya que normalmente están enfocadas en tener servicios 24x7 y con poca demanda de recursos o por periodos cortos, a diferencia de las aplicaciones HPC que consumen más recursos y se ejecutan en lotes o conjuntos de trabajos aumentando los tiempos de ejecución, por ejemplo, las aplicaciones paralelas escritas con estándares de paso de mensajes (MPI), hasta los modelos de BigData y aplicaciones de inteligencia artificial, éstas últimas basadas principalmente en el aprendizaje automático.

Como se ha mencionado en capítulos anteriores, la nube surgió como una evolución de una serie de tecnologías, principalmente en visualización y redes de computadoras, que, como analizamos anteriormente, son parte clave de la nube, ya que facilita tanto la administración de la carga de trabajo, como la interacción y asignación de recursos remotos, originando el modelo de negocio donde el usuario paga por los recursos usados o solicitados.

Por todo lo anterior, en la nube considera tres categorías (fig. 2.1) para la ejecución de aplicaciones HPC :

- **HPC en la nube**, se centra en mover las aplicaciones de HPC a entornos de nube.
- **Nube HPC híbrido (HPC+Cloud)**, en la cual los usuarios usan la nube para complementar los recursos de HPC, principalmente usado en sistemas HPC saturados. [14]
- **HPC como servicio de la nube**, se ofrecen los recursos de HPC a través de servicios en la nube. Esta categoría esta relacionadas con cómo se asignan los recursos, es decir, dinámicamente y las abstracciones para simplificar el uso de la nube.

La nube HPC se esta convirtiendo en una plataforma viable para cluster y grids para ejecutar aplicaciones científicas y servicios de análisis empresarial. La mayoría de los esfuerzos de investigación en la



**Figura 2.1** Categorías de HPC

nube de HPC tienen como objetivo comprender el costo-beneficio de mover aplicaciones de uso intensivo de recursos de entornos locales a plataformas de nube pública.

La nube HPC todavía tiene varios problemas abiertos, por ejemplo, la abstracción de la infraestructura de la nube limita la personalización a detalle de las aplicaciones HPC, además, la mayoría de las redes en la nube no son lo suficientemente rápidas para aplicaciones a gran escala, aquellas con alta comunicación entre procesadores, y aunque los grupos científicos pequeños o pequeñas empresas se benefician del rápido acceso a los recursos de las nubes públicas, esto no suele ser cierto para todos los usuarios.

A pesar de que hay avances en los últimos años en los sistemas de la nube, aún queda mucho por hacer en la nube de HPC [13]. Existen diversas publicaciones recientes donde se investiga la posibilidad de adaptar o implementar tecnologías nuevas a los sistemas de la nube, con la finalidad de reducir poco a poco la diferencia de desempeño, como por ejemplo, en algunas de estas investigaciones se analiza el desempeño en el uso de contenedores linux (LXC<sup>1</sup>) virtualizados [15] [16] en los cuales se obtiene un desempeño cercano al de los sistemas físicos. También existen artículos donde analizan el uso de bibliotecas MPI<sup>2</sup> en sistemas virtualizados con contenedores en sistemas HPC basados en la nube[17] para reducir los cuellos de botella al incrementar la carga de trabajo. Las investigaciones han mostrado algunos intentos en la nube HPC, sin embargo, no presentan una visión general.

Si bien, la ejecución de aplicaciones en entornos de nube y las investigaciones de migrar trabajos HPC o a entornos híbridos trae beneficios importantes, existen muchas opciones y muchos enfoques en los que un sistema HPC se puede beneficiar de las opciones que ofrece la Nube. Existen muchas investigaciones y esfuerzos en analizar el remplazo de los sistemas HPC y siempre intentando reducir costos, pero por el momento, aún se está analizando esta opción. Por lo que, en grandes centros de investigación, se han enfocado en sacar provecho de los servicios ya probados de la nube, como son problemas sencillos en los cuales el desempeño no afecta mucho el resultado ni los tiempos, o simplificando problemas complejos para realizar pruebas sencillas.

Otra de las opciones en la que se han enfocado los centros de investigación, va más orientada al uso de los recursos de la nube como herramienta para la colaboración e intercambio de los resultados de las

<sup>1</sup><https://linuxcontainers.org/>

<sup>2</sup><https://www.open-mpi.org/>

investigaciones. Aprovechando las plataformas y servicios ya probados de la nube, han encontrado una herramienta útil para la convivencia entre los sistemas HPC y los servicios de la nube, y sin inventar el hilo negro, de esta forma se ven beneficiados usuarios y se aprovechan todos los avances tecnológicos.

La necesidad de almacenar, compartir y colaborar ha sido una de las herramientas más útiles a nivel comercial, surgiendo muchas opciones en el mercado. Pero en la mayoría de los casos, no están tan enfocadas para los ambientes científicos, ya que comercialmente el porcentaje de usuarios es mucho menor. Aunado a la gran necesidad de colaboración, sincronización e intercambio, la mayoría de los centros de investigación tienen grupos de desarrollo que se han enfocado a la unión de estos dos mundos, el comercial y el científico, originando un desarrollo importante para todos.

El acceso a sistemas de archivos paralelos de alto rendimiento como GPFS (General Parallel File System) y Lustre, así como las interconexiones de baja latencia y alto ancho de banda como InfiniBand dentro de una máquina virtual permitirían que más aplicaciones científicas se beneficien de entornos virtuales sin sacrificar el rendimiento o la facilidad de uso. Los desarrolladores de software del sistema deben explorar métodos para proporcionar capacidades de HPC en entornos virtualizados.

# Capítulo 3

## Sistemas de alto desempeño (HPC)

Desde las décadas de los 70's, el concepto de cómputo de alto desempeño estaba limitado a las supercomputadoras, que eran sistemas integrados muy costosas y que se usaban en empresas como Cray, NEC, IBM, entre otras, y en la mayoría de los casos, usaban procesadores diferentes entre ellos, dificultando la portabilidad de aplicaciones. Con el desarrollo de los procesadores para uso general a principios de los 80's y 90's, la computación comenzó a evolucionar rápidamente, ya que los precios bajaron y se mejoró la compatibilidad.

A finales de los 90's, el surgimiento de los modelos de cluster o conjuntos de computadoras conectadas entre sí fue desplazando a las supercomputadoras, ya que con computadoras estándar se lograban rendimientos competitivos. [18] La computación de alto rendimiento o High Performance Computing (HPC) está siendo dominada por el uso de clusters, permitiendo a los científicos resolver problemas complejos que exigen una gran capacidad de cómputo. [19]

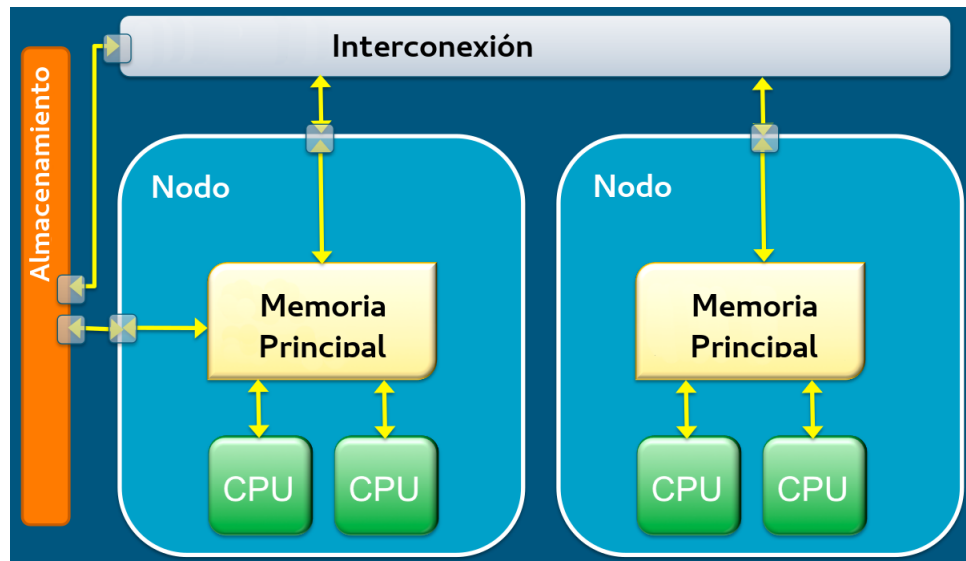
En la actualidad, las aplicaciones que corren en sistemas HPC exigen un alto nivel de rendimiento de red, gran cantidad de almacenamiento a alta velocidad, gran capacidad de memoria RAM y alta capacidad de procesamiento, o incluso todas estas características juntas, por lo consiguiente estos sistemas se han estado especializando para el mejoramiento del rendimiento. Para ellos, y por su complejidad, ha sido necesario dividir los componentes [20][21] para su mejor estudio, investigación y desarrollo como se muestra en la figura 3.1. A continuación, veremos de forma general los componentes de una arquitectura HPC.

Donde cada uno de estos componentes cuenta con toda una línea de investigación para incrementar el rendimiento, ya que, si alguno de estos elementos llegara a tener problemas, todo el sistema presentara problemas.

- **CPUs:** Registros, ALU, multiprocesadores, Pipelining, Caches (L1,L2,L3), etc.
- **Memoria:** Comunicación con el CPU y multiprocesadores, cuellos de botella, etc.
- **Nodos:** Disponibilidad, conectividad de memoria y CPU (NUMA), temperatura, etc.
- **Red entre nodos:** Topologías (Malla, Árbol, Cubo), Ancho de banda, Protocolos de baja latencia (Fibre Channel, InfiniBand, Ethernet).



- **Aplicaciones y Software:** Sistemas operativos, Comunicación entre procesos, Compiladores, etc.
- **Almacenamiento:** Conexión, Sistema de archivo, Ancho de banda, Redundancia, Capacidad, Formas de accesos.



**Figura 3.1** Arquitectura HPC

Considerando la importancia de la infraestructura, en los últimos años y enfocado en la reducción de costos o en la necesidad de ejecutar programas paralelos más grandes, Grid Computing se volvió relevante en el ambiente científico para los usuarios de HPC. La Grid ofrecen a los usuarios acceso a recursos de grandes capacidades administrados por centros de datos especializados [22], lo que ha llamado la atención a un modelo más colaborativo de recursos, tal como la nube.

Recientemente diversas empresas comerciales han permitido el acceso a los centros de investigación para la obtención de resultados mediante la ejecución de HPC, pero en la nube. Lo interesante de ejecutar aplicaciones HPC en la nube es que ya no se trata de un sólo tipo de nodo, es decir, se pueden seleccionar entre una variedad de tipos de nodos, configurados para satisfacer sus necesidades, y puede unirse en configuraciones de clúster que mejoren el rendimiento para su aplicación, abriendo otra línea de investigación.

Considerando la importancia de todos los elementos y lo extenso de cada uno, en este capítulo se abordará el sistema de almacenamiento de un sistema HPC, analizando sus características, limitaciones y nuevos usos y formas de acceso para facilitar la interacción con el usuario.

### 3.1. Sistemas de archivos para sistemas HPC

Conforme la tecnología en procesadores y memoria avanza, en los últimos 20 años la velocidad de ambos ha aumentado en costo/rendimiento, pero el rendimiento de los sistemas de entrada y salida (E/S)

externa a los procesadores ha quedado rezagado, por ejemplo, las unidades de disco aún giran esencialmente a la misma velocidad y los tiempos aún se miden en milisegundos.

La mayoría de los casos de los sistemas HPC requieren alguna forma de almacenamiento de alto rendimiento y aún en la actualidad, uno de los componentes más lentos de la arquitectura HPC por lo general son los discos, este bajo rendimiento en relación con los procesadores degrada fuertemente el rendimiento de los sistemas HPC.

El método más simple es usar el nodo principal como servidor NFS, pero tiene limitaciones al no escalar a un gran número de nodos, incluso esta solución requiere que el nodo principal tenga algún tipo de hardware con el cual se optimiza los accesos de entrada y salida, como los arreglos RAID <sup>1</sup> (Arreglo redundante de discos independientes) y aprovechar la protección de los datos que disponen estos sistemas. Estos sistemas tradicionales de almacenamiento quedan limitados por las características y velocidades del hardware y de la red, ya que los sistemas HPC requieren gran cantidad de acceso simultáneo por todos los nodos. Considerando todas las limitaciones, el desarrollo ha continuado [23], con el cual han surgido varios sistemas de archivos que cubren los requerimientos de los sistemas HPC, llamados **sistemas de archivos distribuidos**.

Este tipo de almacenamiento proporciona amplia escalabilidad, tanto en rendimiento como en capacidad de almacenamiento, facilidad de la disponibilidad de los datos con espacio de nombre global y la capacidad de distribuir archivos muy grandes y posibilidad de concurrencia en el acceso.

Existen varios desarrollos en la actualidad, los cuales están enfocados a distintos objetivos y basados en diferentes modelos, y como se ha mencionado anteriormente, la mayoría están enfocados en mejorar el desempeño al optimizar los accesos de entrada y salida basados en diferentes formas de distribución de la información, pero recientemente una de las características que se empiezan a buscar en la mayoría de los proyectos es la compatibilidad con diferentes sistemas operativos y aplicaciones sin perder las propiedades de seguridad y desempeño, con la finalidad de abarcar más aplicaciones y/o funcionalidades.

Después de mucho desarrollo, las dos formas más comunes de solucionar estos objetivos es por medio del estándar por parte del sistema operativo, es decir, a nivel de kernel o núcleo, como POSIX (Portable Operating System Interface) y la otra forma es a nivel de usuario con la interfaz FUSE (Filesystem in Userspace), con el cual el usuario sin privilegios puede tener un sistema de archivos. Dichas compatibilidades facilitan la conexión entre la mayoría de los sistemas o aplicaciones a los sistema de archivos, independientemente de que estén enfocados en problemas específicos o tengan un modelo muy particular.

### 3.1.1. Interfaz estándar del sistema operativo POSIX

Desde el inicio de los sistemas operativos, se empezó a simplificar el desarrollo de aplicaciones, por lo que una forma de generalizar el uso del sistema de archivos era estandarizar su comportamiento, de tal forma que fueran uniformes e independientes de sus implementaciones. Para garantizar un comportamiento uniforme a través de todos los diferentes sistemas de archivos, se implementó el estándar POSIX<sup>2</sup> por el Instituto de Ingenieros Eléctricos y Electrónicos conocido como IEEE<sup>3</sup>, en donde se define la se-

<sup>1</sup><http://ieeexplore.ieee.org/document/25791/>

<sup>2</sup><http://standards.ieee.org/develop/wg/POSIX.html>

<sup>3</sup><https://ieee.org/>

mántica que los sistemas de archivos deben de tener, garantizando de forma simple que la información sea la correcta (tamaño de archivo, fecha, nombre, etcétera) y funciones básicas (crear, abrir, borrar, escribir, recuperar metadatos, entre otros). A pesar de que este estándar fue diseñado para sistemas de archivos locales, contiene funciones que facilitan el uso remoto o concurrente, como el bloqueo de archivos, por lo que gran parte de los proyectos de sistemas de archivos distribuidos, contemplan la compatibilidad a este estándar, aprovechando estas funciones, pero no ha sido una tarea fácil, ya que los sistemas de archivos modernos y principalmente los sistemas HPC, que requieren funcionalidades más demandantes, enfocándose en los accesos concurrentes, consistencia y coherencia, por medio de protocolos de bloqueos de algunas funciones (lectura y escritura) o la definición de conceptos nuevos como de consistencia eventual, utilizado para consultas de metadatos obsoletos o no actualizados. Todo esto logra una compatibilidad parcial o modificada que por el momento es suficiente.

### 3.1.2. Sistema de archivos en espacio de usuario FUSE

El sistema de archivos en el espacio de usuario (FUSE) es un proyecto de código abierto que busca crear un módulo para el kernel del sistema operativo que permita a los usuarios no privilegiados crear sus propios sistemas de archivos sin tener que escribir a nivel del kernel. Esto se logra ejecutando como módulo el FUSE que opera como un “puente” a la interfaz del kernel únicamente. FUSE es un sistema muy poderoso pues prácticamente todos los recursos de almacenamiento disponibles en la máquina pueden convertirse en un sistema de archivos virtual.

Como se ha mencionado, un sistema de archivos eficiente es importante para los sistemas de computación de alto rendimiento (HPC) y principalmente en aplicaciones científicas a gran escala, por esta razón los sistemas de archivos se implementan como parte del núcleo o kernel del sistema operativo con la finalidad de tener el mejor rendimiento [24] [25]. Sin embargo, a medida que la complejidad de los sistemas de archivos creció, muchos sistemas nuevos comenzaron a desarrollar instrucciones en el espacio de usuario, mejorando la compatibilidad en diversos sistemas operativos, la seguridad al limitar las instrucciones con privilegios en el núcleo o kernel, fácil desarrollo de funciones especializadas y estabilidad al no bloquear el kernel.

En la actualidad esta compatibilidad se ve más presente, por todos los beneficios que se logran, aunque hay un poco de polémica en cuestiones de rendimiento, por eso los desarrollos más recientes se han enfocado en esto, logrando muchas mejoras, razón por la que ha tenido mayor aceptación, aunque el desarrollo continúa enfocado en el rendimiento. Algunos de los sistemas que recientemente han integrado esta compatibilidad, son ZFS, NTFS, GPFS, HDFS, GlusterFS. En la tabla 3.1 se observa la diversidad de sistemas de archivos, sus principales usos y características.

## 3.2. Sistema de archivos distribuidos

A diferencia del almacenamiento centralizado que guarda todos los datos en un solo dispositivo, los sistemas de archivos distribuidos son infraestructuras que proporcionan los mecanismos para poder almacenar los datos de una forma jerárquica en los múltiples servidores de archivos y recursos compartidos, permitiendo que la carga se divida a través de los dispositivos de almacenamiento diferentes conectados

**Tabla 3.1** Usos y características de sistemas de archivos

<b>Campo de uso</b>	<b>Sistema de Archivo</b>	<b>Características</b>
Big Data	- HDFS - QFS - MapR	Integración con MapReduce, Tolerancia a fallas básica en hardware
Supercómputo	- Lustre - GPFS - OrangeFS	Rendimiento de escritura en paralelo, Soporte para interconexiones de alto rendimiento como Infiniband
Discos Compartidos	- GFS2	Alto nivel de POSIX
Propósito General	- NFS - Ceph - GlusterFS	Escalabilidad, Facilidad de administración
Archivos Personales	- AFS - OwnCloud /- Dropbox	Privacidad, Intercambio, Sincronización
Experimentos de Física de altas energías (HEP)	- dCache - XRootD - EOS / Castor - CernVM-FS	Acceso a la cinta, Distribución de software, Tolerancia a fallas básicas en hardware

a la red local, facilitando el acceso de alto rendimiento a través de operaciones de entrada y salida [26] (IOPS<sup>4</sup>) coordinadas y simultáneas entre los clientes y los nodos de almacenamiento. A medida que se aumentan los dispositivos de almacenamiento, el rendimiento se puede duplicar o triplicar fácilmente, aumentando el rendimiento hasta los límites de la capacidad total de la red.

Muchos sistemas de archivos se han desarrollado a lo largo de los años y casi dos décadas de investigación no han tenido éxito al desarrollar sistemas que contengan todas las características necesarias[27][28][29], por lo que en la actualidad se han realizado muchos proyectos que han diseñado e implementado sistemas de archivos distribuidos con una variedad de arquitecturas, funcionalidades y características orientadas a diferentes aplicaciones o usos, por ejemplo, “EOS” junto con el protocolo XRootD está optimizado para acceso de alto rendimiento a conjuntos de datos de gran escala y baja latencia para experimentos científicos como los realizados para física de altas energías en el Organización Europea para la Investigación Nuclear (CERN) [30]. Sistema de archivos “Hadoop (HDFS)” está diseñado como una capa de almacenamiento para el algoritmo MapReduce [31]. Sistema de archivos “CernVM ( CVMFS)” está optimizado para distribuir binarios de software usado para procesamiento de información de experimentos científicos de física de altas energías (High Energy Physics o HEP) por redes internacionales en el protocolo HTTP [32]. Sistema GPFS (General Parallel File System) desarrollado por IBM optimizado para el acceso de entrada y salida dividiendo los archivos individuales en bloques y guardándolos en varios discos. Sistema Lustre, optimizado para el almacenamiento paralelo de alto desempeño, siendo éste último uno de los sistemas de archivo distribuidos más usados en la actualidad por los sistemas HPC, por su crecimiento y amplia adaptabilidad en desempeño.[33].

El reto que tienen estos proyectos es realizar el almacenamiento de datos de forma eficiente, segura y robusta. Aunado a esto, existen diversos detalles y complicaciones, como un ejemplo de los diversos problemas a los que se enfrentan estos sistemas es el de ubicación y disponibilidad de archivos. Una forma de aumentar la disponibilidad de archivos dentro de un sistema de archivos distribuidos es mediante el uso de la replicación o copias de archivos, pues, en su mayoría se pueden dividir en dos categorías principales, como la replicación optimista (donde suponen que las fallas son raras e implementan esquemas de recuperación para manejar inconsistencias) y la replicación pesimista (donde se supone que las fallas son más comunes e intentan asegurar la consistencia de cada acceso)[34].

Un paso importante en la evolución de los sistemas de archivos distribuidos fue el poder considerar que el acceso al archivo remoto podría ser igual al de archivos locales. Esta propiedad, llamada transparencia de red, implica que cualquier operación que se pueda realizar en un archivo local también se puede realizar en un archivo remoto. A continuación, se mencionan las diferentes arquitecturas de los sistemas de archivos distribuidos:

- **Cliente-Servidor:** Protocolo de comunicación que permite a los clientes acceder a los archivos almacenados en un servidor. Proporciona una vista estándar de su sistema de archivos local. Ejemplo: NFS.
- **Clúster:** Consiste en un único servidor maestro junto con múltiples servidores que contienen fragmentos de 64MB permitiendo que un solo servidor controle todos los fragmentos. Las características importantes que se consideran en el diseño son la independencia de los metadatos y los datos, dis-

---

<sup>4</sup>Input/output operations per second

tribución de objetos confiable, y administración de la distribución dinámica de metadatos. Ejemplo: Google File Systems

- **Arquitectura simétrica:** Se basa en la tecnología peer-to-peer que utiliza un sistema de tablas “hash distribuidas”. Los clientes cuentan el administrador de metadatos, por lo que todos los nodos entienden la estructura del disco compartido.
- **Arquitectura asimétrica:** Sistemas donde hay uno o más administradores de metadatos dedicados que mantienen el sistema de archivos y la estructura de disco asociada. Ejemplos: Lustre, Panasas ActiveScale.
- **Paralelos:** Los bloques de datos se fragmentan en paralelo en múltiples dispositivos y servidores de almacenamiento, permitiendo que todos los clientes tengan acceso a los mismos archivos en el mismo tiempo (alta disponibilidad a la concurrencia de lectura y escritura).

Actualmente, se realizan proyectos utilizando varias arquitecturas al mismo tiempo con la finalidad de mejorar en rendimiento o incrementar la funcionalidad y de esta forma poder contar con un sistema integral y flexible que pueda usarse con varios objetivos, a veces llamados sistemas de archivos de propósito general[35].

Alguno de los sistemas de archivos distribuidos más comunes hoy en día son Lustre <sup>5</sup>, OrangeFS <sup>6</sup> y GPFS (General Parallel File System) <sup>7</sup>. Pero existen más desarrollos para sistemas distribuidos para objetivos específicos que comparten algunas de las arquitecturas o tecnologías de los sistemas de propósito general. Uno de los más importantes es el sistema de almacenamiento EOS <sup>8</sup> desarrollado por varios centros de investigación científica, de los cuales el principal es la Organización Europea para la Investigación Nuclear conocido como CERN, con el cual brinda un almacenamiento de alto desempeño optimizado principalmente para los datos generados por el Gran Colisionador de Hadrones o LHC.

Todos estos sistemas se caracterizan por el uso coordinado de varias rutas de E/S y proporcionan un importante beneficio de rendimiento, especialmente cuando se transfieren cargas de trabajo que involucran a una gran cantidad de clientes. La capacidad y el ancho de banda se pueden escalar para almacenar enormes cantidades de datos. Las características de almacenamiento pueden incluir alta disponibilidad, duplicación, replicación e instantáneas. A continuación, se explican las características de los sistemas de archivos mencionado anteriormente.

### 3.2.1. Sistema de archivos distribuidos Lustre

Lustre es un sistema de archivos basado en objetos y se compone de tres servidores principales como se muestra en la figura 3.2: Servidor de metadatos (MDSs), Servidor del almacenamiento de objetos (OSSs) y clientes.

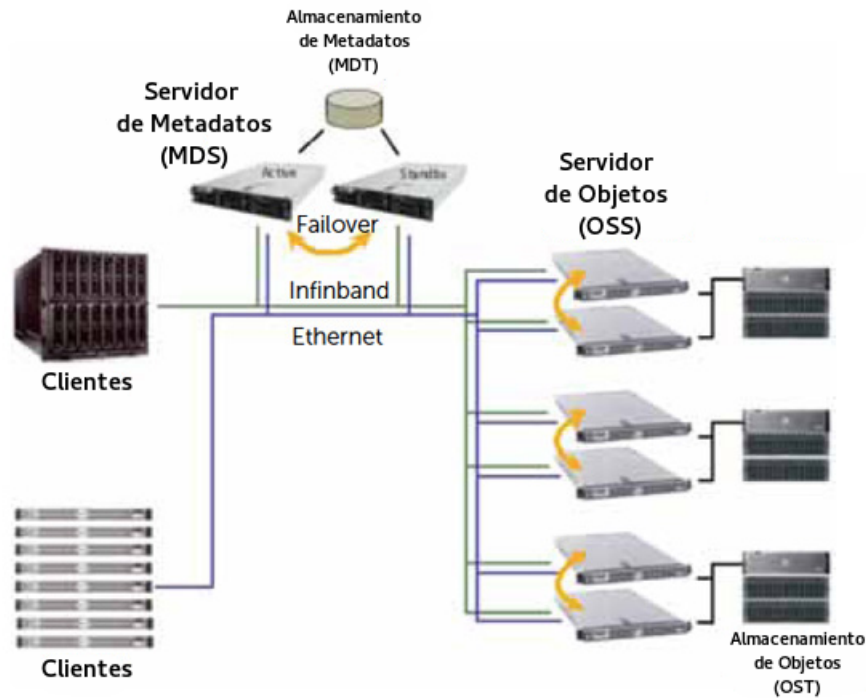
---

<sup>5</sup><http://lustre.org>

<sup>6</sup><http://www.orangefs.org/>

<sup>7</sup><http://https://www.ibm.com/us-en/marketplace/scale-out-file-and-object-storage>

<sup>8</sup><http://eos.web.cern.ch/>



**Figura 3.2** Sistema de archivos Lustre

El servidor de metadatos (MDS) proporciona la información de los archivos a los clientes como el nombre de los archivos, tamaño, ubicación, estructuras de directorio, permisos de acceso, etcétera, y son guardados en uno o más almacenamientos de metadatos llamado MDT. Posteriormente el servidor de objetos (OSS) proporciona los dispositivos de bloques y los datos, guardados en los almacenamientos de objetos (OSTs).

Lustre usa dispositivos de bloques para el almacenamiento de metadatos y datos de los archivos, y cada dispositivo de bloque sólo puede ser gestionado por un servicio de Lustre (como un MDT o un OST). La capacidad total de un sistema de archivos Lustre es la suma de todas las capacidades OST individuales. Los clientes de Lustre acceden y usan datos simultáneamente a través de las llamadas de sistema de E/S POSIX, esta última siendo una de las características que ha facilitado el uso del Lustre ya que brinda compatibilidad con los sistemas operativos más usados y permite garantizar integridad en los archivos en accesos simultáneos.

Este sistema de archivos ha tenido gran aceptación en los ambientes empresariales y en los grandes centros de datos, por lo que su desarrollo ha sido rápido debido a la demanda empresarial pues hoy en día el acceso a los sistemas HPC es más accesible.

La arquitectura se desarrolló como proyecto de investigación en la Universidad de Carnegie Mellon en 1999, posteriormente su desarrollo fue impulsado por el Departamento de Energía de los Estados Unidos junto con HP e Intel. En 2007, Sun Microsystems adquirió su propiedad intelectual [36], incluyéndolo en sus equipos de alto rendimiento. Posteriormente Sun fue adquirido por Oracle en 2010, pero a poco tiempo de estar a cargo del proyecto, anunció que no seguiría el desarrollo del Lustre [37], por lo que surgieron

varias organizaciones para proporcionar soporte y desarrollo como proyecto de código abierto, que incluye Whamcloud, [38] Open Scalable File Systems, Inc. (OpenSFS), EUROPEAN Open File Systems (EOFS) y otros.

A partir del 2013, Intel junto con OpenSFS (Open Scalable File Systems) entre otras, se comenzó a expandir el uso de Lustre más allá del HPC tradicional, como Hadoop, también establecieron un portal de la Comunidad Lustre, donde se publica información y documentación para apoyar a la comunidad de código abierto.

### 3.2.2. Sistema de archivos distribuidos OrangeFS (PVFS2)

En 1993 fue desarrollado el PVFS por la Universidad Clemson como un sistema de archivos para máquinas virtuales paralelas y que fue patrocinado por la NASA para el estudio de los patrones de entrada y salida de los programas paralelos. Con el surgimiento de los Clusters Linux, y al no contar en ese tiempo con sistema de archivos paralelos, el proyecto continuó su desarrollo con el objetivo de tener grandes anchos de banda en la red para las operaciones de lectura y escritura de varios procesadores o hilos. Compatibilidad por medio de interfaces nativas tipo POSIX/UNIX, interfaces orientadas al manejo de mensajes (MPI-IO) [39] y soporte a comandos y utilerías de Linux/UNIX como `ls`, `cp`, `rm`. Después de varios años de desarrollo, la primera versión de PVFS fue abandonada en 2005, iniciando una nueva versión PVFS2, la cual fue desarrollada en el Laboratorio Nacional Argonne y en el Centro de Supercomputación de Ohio, USA[40]. La nueva versión del sistema de archivos se lanzó en 2003, presentando servidores de objetos, metadatos distribuidos, funciones basadas en MPI, soporte para múltiples tipos de redes y una arquitectura de software para una fácil experimentación y crecimiento. En 2007 se inició para IBM, Blue Gene [41] con lo cual se empezaron mejoras basadas en la duplicidad y el desarrollo de extensiones para soportar grandes directorios de pequeños archivos, mejorando la seguridad y capacidad de redundancia. Al entrar en conflictos los desarrollos con IBM, se divide el proyecto en BlueFS (por parte de IBM) y OrangeFS que se convirtió en el proyecto principal a partir del 2011.

El sistema de archivos PVFS2 divide los datos en todos sus discos y consta de 4 componentes principales y un conjunto de programas. Los componentes son PVFS2-server, pvfslib, PVFS-client-core y el módulo para el kernel PVFS. Los programas incluyen la herramienta de administración, tales como `pvfs-ping`, `pvfs-ls`, `pvfs-cp`, entre otros, que operan directamente en el sistema de archivos sin utilizar el módulo kernel. Una característica clave de diseño es el protocolo PVFS con el cual se transmiten los mensajes entre el cliente y el servidor.

### 3.2.3. Sistema de archivos distribuidos EOS

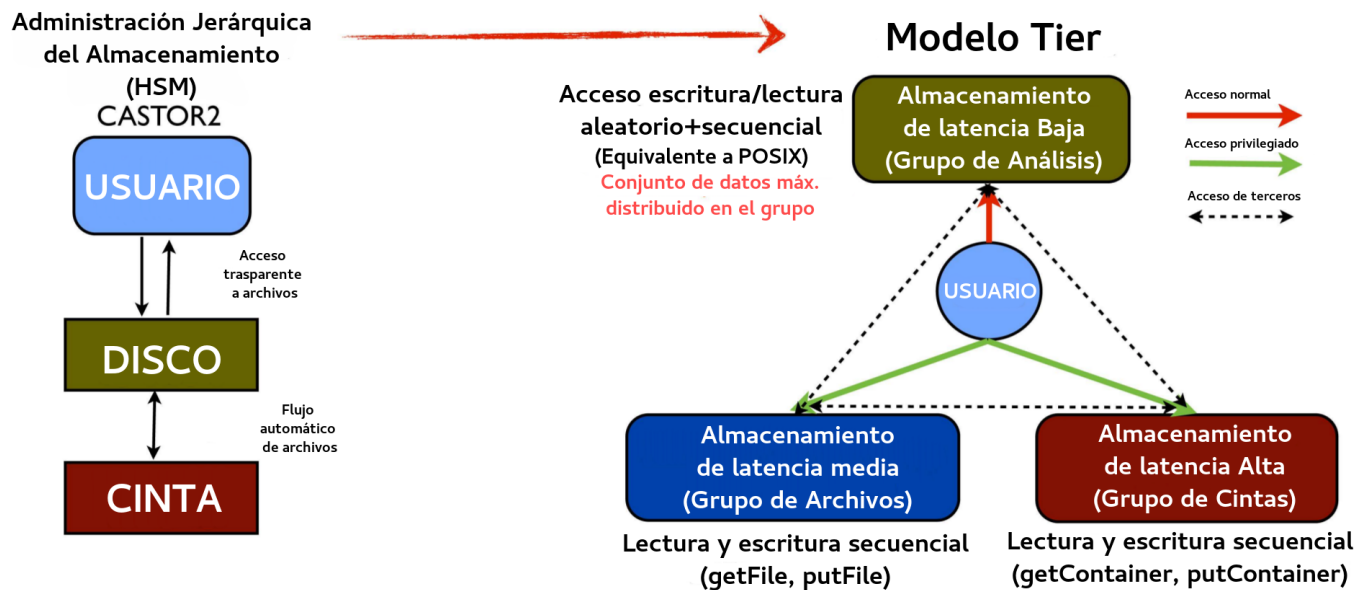
Es un sistema de almacenamiento distribuido de código abierto desarrollado por la Organización Europea para la Investigación Nuclear (CERN) en el 2010 y puesto en producción desde 2011. El objetivo principal del proyecto es proporcionar tecnología de almacenamiento en disco de baja latencia y rápido acceso simultáneo para los requerimientos de los experimentos principales del CERN, como son LHC Alice, Atlas, CMS y Lhcb.

Este proyecto es la continuación de sistema de almacenamiento existente llamado “CASTOR” (ver



fig. 3.3), en el cual requería almacenar 12 millones de archivos por mes de forma jerárquica almacenado en un cache en disco, y posteriormente de forma transparente se traspasan al almacenamiento en cinta en base a políticas definidas. Considerando estas necesidades, se diseñó el nuevo modelo de almacenamiento llamado “Tier” dividiendo el sistema en tres tipos de almacenamiento.

- Almacenamiento de análisis basado en disco de baja latencia que admite acceso de escritura y lectura (“r” y “w”) aleatorio y secuencial (equivalente al sistema POSIX).
- Almacenamiento de archivo basado en disco de latencia media para acceso secuencial de lectura y escritura única.
- Almacenamiento en cinta de alta latencia para el acceso secuencial de lectura y escritura única a los contenedores de datos (datasets).



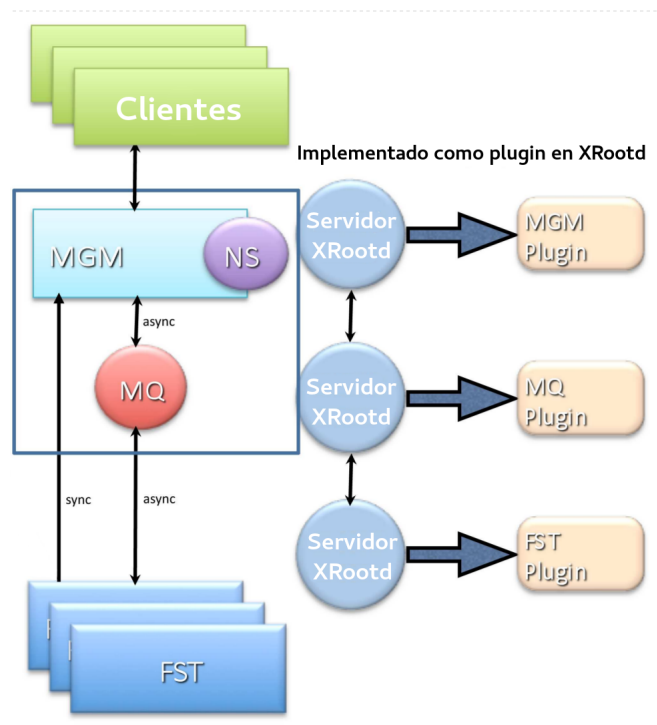
**Figura 3.3** Evolución de EOS: Migración del **Modelo HSM** Administración jerárquica del almacenamiento (Hierarchical Storage Management) al almacenamiento por niveles, que dependen de la velocidad y latencia del dispositivo de almacenamiento para reducir costos (**Modelo Tier**).

El sistema comenzó a funcionar con el experimento ATLAS con 2 PB de tamaño y ha estado creciendo hasta 140 PB de almacenamiento, proporcionados por 44,000 discos duros. El espacio de almacenamiento total actual está segmentado en seis instancias de tolerancia a falla independiente para los cuatro experimentos de LHC Alice, Atlas, CMS y Lhcb, una instancia de experimento compartido más pequeño y una instancia de usuario genérica para todos los usuarios de CERN.

Este sistema de almacenamiento ha aportado al CERN importantes mejoras en comparación con soluciones de almacenamiento anteriores al permitir cambios rápidos, permitiendo que el centro de datos satisfaga rápidamente los requisitos de rendimiento y confiabilidad que cambian entre los experimentos de LHC con un mínimo de movimientos de datos y reconfiguración dinámica, permitiendo el rápido diseño de nuevos flujos de trabajo que se requiere para la migración al almacenamiento de cinta y almacenamiento a largo plazo, que es un requerimiento necesario para el proyecto LHC Run II.

EOS separa la ruta de entrada y salida en el acceso de metadatos a través de un servicio de metadatos “MGM” (componente de administración) proporcionando control en autenticación, administración de permisos y cuotas, control del espacio de nombre y el acceso a los datos a través de los servicios de archivos entrada y salida. Componente del almacenamiento de archivos (“FST”) que corre en cientos de servidores, donde almacena físicamente los archivos (réplicas), mantiene disponibles los archivos para los usuarios y verifica el buen funcionamiento de los discos y la integridad de los datos. (ver fig. 3.4)

Para garantizar la mínima latencia en el acceso a los archivos, todos los metadatos se mantienen en la memoria de los nodos del servidor de metadatos y se mantienen con la tecnología de “Registro Anticipado” (Write Ahead Log). En la mayoría de los casos, los archivos se replican en dos discos tipo JBOD (“Just a Bunch Of Disks”). Las modificaciones de estado y configuración se intercambian entre los servidores de almacenamiento y metadatos utilizando un servicio de cola de mensajes “MQ” ejecutado en el mismo servidor que el “MGM”.



**Figura 3.4** Flujo de EOS

Los tres servicios MGM, FST y MQ se han implementado utilizando la estructura de XRootD (3.4)

<sup>9</sup> basado en un modelo cliente-servidor, proporcionando acceso remoto a través del protocolo XRootD y puede ser montado con autenticación robusta sistema de archivos a través de una implementación FUSE (por ejemplo, LXPLUS, LXBATCH) y cuenta con una semántica POSIX simplificada con el que logra un 99% de los casos de uso típicos, soportando varios protocolos (HTTP, ROOT, GridFTP, SRM, SMB ...).

Un concepto fundamental de EOS es utilizar un conjunto de discos individuales tipo JBOD <sup>10</sup> como medio de almacenamiento sin la necesidad de construir arreglos RAID locales. Todos los nodos de almacenamiento se dividen en grupos y dentro de un grupo, los archivos se almacenan usando algoritmos de RAID de red de nivel de archivo. El algoritmo implementado actualmente es RAID-1 (N) donde “N” es el número de réplicas para cada archivo.

El futuro de la gestión de datos para LHC en CERN trae nuevos requisitos para escalabilidad y un cambio de programación y manejo de datos en comparación con el sistema de almacenamiento masivo HSM en uso en la actualidad. Un pronóstico para el volumen de almacenamiento basado en disco en el CERN es de escala Exabyte (1 Exabyte ó EB =  $10^{18}$  bytes o 1000 Petabyte ó PB) con cientos de millones de archivos.

El reto para el futuro es transformar el servicio de metadatos EOS en una solución de escalamiento horizontal sin perder el acceso a los archivos de baja latencia, que sería fundamental para el análisis científico de la física. La integración de las unidades Ethernet y las tecnologías de almacenamiento de objetos permitirá reducir aún más los costos de almacenamiento e incrementar la confiabilidad y la eficiencia en el futuro. Un objetivo a mediano plazo es fusionar la funcionalidad EOS con las plataformas de almacenamiento de código abierto ampliamente utilizadas para ampliar la comunidad de usuarios de software y su reutilización. Otro proyecto de CERN para el almacenamiento de archivos (CERNBox) se basa en EOS (como sistema de almacenamiento de servicios de fondo), que proporciona instalaciones de sincronización y uso compartido a los usuarios del CERN, y con el que surge toda una línea de investigación al realizar la interacción de arquitecturas y modelos diferentes para tener un almacenamiento más completo.

---

<sup>9</sup><http://xrootd.org>

<sup>10</sup>Just a Bunch of Disks

# Capítulo 4

## CERNBox

La evolución de los sistemas continua, y gran parte de esta evolución es por las necesidades de los usuarios, y por esto, los sistemas de almacenamiento, así como los sistemas HPC se han visto forzados a adaptarse a esta evolución. Recientemente, los usuarios cada vez están más interesados en los servicios de almacenamientos externos en la nube, como Dropbox o Google Drive, para compartir y administrar sus archivos. Estos servicios por su aplicación comercial y desempeño, no se habían considerado como parte de los sistemas de almacenamientos de alto rendimiento, por lo que no se proporcionaban en los modelos de dichos sistemas.

En la Organización Europea para la Investigación Nuclear o CERN<sup>1</sup> (European Organization for Nuclear Research) los administradores de los servicios de almacenamiento de datos CERN-IT-DSS<sup>2</sup> (Data and Storage Services group) realizaron un análisis sobre la necesidad de estos servicios y registraron aproximadamente 4500 equipos de cómputo (computadoras portátiles o de escritorio) con accesos al sistema Dropbox, incluso varios usuarios han considerado la compra de este servicio. [42]

Esta tendencia se presenta en casi todos los centros de investigación, donde los servicios de almacenamiento en la nube para actividades científicas y técnicas permiten a grupos de investigadores, compartir, transferir y sincronizar archivos entre sus computadoras personales, dispositivos móviles y grandes repositorios de datos científicos. Al implementar funcionalidades de sincronización a almacenamiento o repositorios de datos existentes a gran capacidad (del orden de 1 Petabyte) crea una oportunidad para nuevos escenarios para el análisis de datos. Los principales objetivos de estos proyectos es proporcionar una interfaz extremadamente fácil de usar disponible en todas las plataformas de clientes, manteniendo la seguridad e integridad de los datos, sin problemas entre los grupos de trabajo.

Considerando esto, los administradores del centro de datos del CERN, comenzaron a evaluar varios proyectos de código abierto que brindan estas funcionalidades, creando el proyecto CERNBox[43].

Con este sistema se quiere proporcionar una alternativa viable a Dropbox para los usuarios del centro de investigación con la finalidad de que responda las expectativas actuales en el almacenamiento en la nube basado en software libre y en una comunidad abierta para el desarrollo, pero manteniendo la seguridad de los datos. Desde el inicio, este proyecto proporcionó una línea de investigación en interfaces de

---

<sup>1</sup><https://home.cern/>

<sup>2</sup><https://espace.cern.ch/it-dss/>

almacenamiento, enfocándose en mejorar el rendimiento con sistemas de almacenamiento tradicionales como NFS o sistemas locales como RAID o JOBD y en la integración con almacenamiento de datos de alto rendimiento actuales como EOS.<sup>3</sup>

En los inicios del proyecto CERNBox, se analizaron los principales aspectos, características y funcionalidades para poder tener éxito, los cuales, desde el principio se consideraron cubrir los mismos que proporcionan los servicios comerciales, como son: Compatibilidad de los principales sistemas operativos (Windows, Linux, MacOSX, Android e iOS) y configuración pero con acceso sencillo.

Para evitar la dependencia por parte de una empresa o proveedor, la solución tendría que ser basado en software de código abierto, de esta forma se podría tener el control del desarrollo, administración y evolución del servicio a futuro. El análisis para encontrar el proyecto que cumpla con todas las características necesarias no fue fácil ya que el servicio de Dropbox ya ha evolucionado rápidamente. Entre los proyectos que se consideraron para el CERNBox, fueron:

- Syncany ([www.syncany.org](http://www.syncany.org)), que se encuentra en estado experimental.
- SparkleShare ([www.sparkleshare.org](http://www.sparkleshare.org)), usa Git como protocolo, y no es la mejor opción para ciertos tipos de archivos comunes (medios, imágenes, archivos binarios).
- Seafile ([www.seafile.com](http://www.seafile.com)), a pesar de tener poco tiempo de desarrollo, ofrece buenas características, pero la interfaz de usuario no compite con la de Dropbox.
- Pydio 5 (<http://pydio.io>), está desarrollado principalmente como una plataforma basada en la web y los clientes de escritorio aún están en desarrollo.

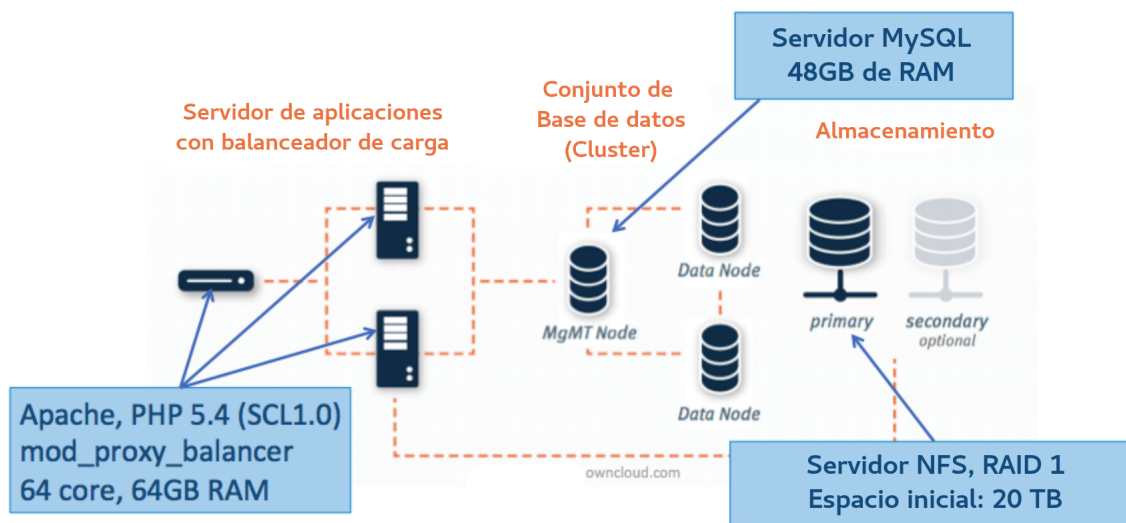
El proyecto de ownCloud, fue uno de los que presentaron mejores servicios y opciones, ya que desde sus inicios se ha desarrollado como una solución de reemplazo Dropbox para entornos corporativos con un amplio conjunto de funciones y una interfaz moderna. Cuenta con un entorno de trabajo web/móvil de escritorio multiplataforma y admite conceptos modernos de integración de escritorio, como arrastrar y soltar, integración de notificaciones del sistema para clientes de sincronización. Se puede tener control de acceso tipo ACLs (Listas de Control de Acceso) para compartir archivos y carpetas del usuario. Los archivos también se pueden compartir de forma anónima con usuarios externos mediante URL hash con fecha de caducidad opcional y protección con contraseña. Por todo lo anterior los miembros del proyecto decidieron considerarlo para el CERNbox. [43]

Aunado a todas los servicios y herramientas con las que cuenta el proyecto, recientemente se han encontrado evaluaciones e implementado en varias universidades e institutos de investigación en Europa (por ejemplo, la Universidad Técnica de Berlín[2], el Instituto Max Planck de Investigación Neurológica). ETH Zurich ejecuta un servicio beta público basado en ownCloud Enterprise Edition y almacenamiento de archivos comerciales. El sistema es utilizado actualmente por 1900 usuarios con el objetivo de ser una plataforma de intercambio para los estudiantes y el personal.

<sup>3</sup>Workshop on Cloud Services for File Synchronisation and Sharing <https://indico.cern.ch/event/336753/>

## 4.1. Implementación CERNBox con ownCloud

Al inicio del proyecto CERNBox, se contaba con la versión 5 de ownCloud, pero ha evolucionado rápidamente. Actualmente, la última versión es la 10.0.9, en la cual han mejorado la integración a sistemas de autenticación OAuth (<https://oauth.net/>). El servidor ownCloud se implementa en lenguaje PHP como parte de una arquitectura. Los clientes de sincronización de escritorio se implementan en C++ con bibliotecas para el ambiente gráfico “Qt” (<http://qt-project.org>) y “libneon” como cliente del protocolo “WebDAV” ([www.webdav.org/neon](http://www.webdav.org/neon)). Uno de los aspectos importantes para la comunidad de código abierto, es que el código fuente del servidor y del cliente de Android, se tiene acceso por medio de la aplicación “github” (<http://github.com/owncloud>) para que cualquier persona modifique o mejore el código por algún problema en particular o para una funcionalidad nueva.



**Figura 4.1** Prototipo CERNBox con NFS

En 2014, en el CERN instaló un servicio de prueba basado en almacenamiento NFS (fig. 4.1), con el cual se probó el sistema y que demostró gran aceptación por parte de los usuarios, pues aumentó su uso constantemente (varias decenas de usuarios nuevos semanalmente). Con un uso amplio, desde la sincronización de archivos de datos (archivos de análisis científicos), archivos de trabajo arbitrarios, hasta documentos de oficina compartidos. Posteriormente al investigar e implementar más protocolos de acceso, como la integración del sistema EOS, la aceptación fue mayor, ya que los usuarios pudieron tener acceso más fácil a todos los datos desde cualquier dispositivo.

El software ownCloud ofrece la posibilidad de utilizar cualquier sistema de archivos (y más recientemente algunos sistemas de almacenamiento de objetos) para almacenar los datos. Los archivos se almacenan en una estructura de directorios directa. Los metadatos (en particular, el que controla la sincronización) se almacenan en una base de datos relacional que, en última instancia, controla el comportamiento de los clientes de sincronización.

La implementación del sistema CERNBox con ownCloud no fue sencilla, ya dicho sistema cuenta

con detalles internos para llevar el control de sincronía, por ejemplo, en el servidor principal ownCloud asume que el sistema de archivos solamente es accedido por el mismo, lo que hace incompatible el acceso a través de comandos de sistema tipo POSIX como “cp”, “mv” y “rm”. A pesar de que ownCloud soporta el acceso a almacenamiento externo, requiere leer constantemente toda la estructura de directorios (el árbol) frecuentemente para poder buscar actualizaciones de los datos. Otro problema al que se enfrentaron fue el escalamiento de la capacidad de archivos, ya que por ejemplo, en el caso del CERN, el sistema EOS cuenta con aproximadamente  $10^8$  archivos, por lo que ownCloud tiene la necesidad de almacenar el registro de los metadatos en una base de datos relacional, pero rápidamente quedaría saturado el sistema. Sin embargo, debido a la naturaleza basada en estado del algoritmo de sincronización de ownCloud y la sencilla organización del sistema de archivos de servidor de ownCloud, fue posible implementar el acceso al sistema EOS y al mismo tiempo cumplir con los protocolos de sincronización de ownCloud.

Una forma de solucionar el problema de la base de datos relacional fue delegando su funcionalidad a EOS, ya que es un sistema de archivos compartido con un espacio de nombres jerárquico. Y la forma para solucionar el problema de sincronización en el sistema EOS, fue por medio de una etiqueta única llamada “ETAG” para que ownCloud pudiera identificar actualizaciones en los archivos y detectar los conflictos en los cambios.

Es importante que el cliente de sincronización no se cambie, para evitar tener problemas con la compatibilidad con los clientes del desarrollo principal por ownCloud. Para el servidor web, desarrollaron un conjunto de interfaces (plugins) para integrarlo con EOS. Actualmente hay 2100 cuentas de usuario en CERNBox. En los congresos más recientes del CERN, los administradores han expuesto la gran aceptación de este proyecto, ya que la integración de EOS como sistema de almacenamiento ha ayudado a incrementar la capacidad de almacenamiento por usuario. Con dicha tendencia y aceptación, los administradores han considerado que en poco tiempo CERNBox podría ser el principal punto de acceso a todos los repositorios de datos del CERN, teniendo todas las ventajas del almacenamiento en la nube y creando una comunidad donde se puedan implementar más funcionalidades, como la sincronía en una sola dirección para experimentos más pequeños, en los cuales actualmente la descarga de archivo se hace de forma manual, entre otras.

Con relación a todo lo analizado y expuesto en capítulos anteriores, en este trabajo se propone la instalación de un prototipo ownCloud, con la finalidad de tener un acceso transparente y amigable a los diferentes repositorios de datos del Instituto de Ciencias Nucleares. Y de esta forma, tener un mejor aprovechamiento de la infraestructura actual y un mejor servicio en el acceso a los datos científicos. A continuación se describirán los detalles del prototipo.

# Capítulo 5

## Prototipo ownCloud ICN

### 5.1. Antecedentes

El Instituto de Ciencias Nucleares de la UNAM realiza investigación teórica, experimental y aplicada en los programas académicos de Estructura de la Materia, Física de Altas Energías, Física de Plasmas, Gravitación y Teoría de Campos, Interacción de la Radiación y Materia, Química de Plasmas, Química de Radiaciones, Radioquímica y Dosimetría, por consiguiente, el uso de herramientas del cómputo científico y sistemas de cómputo de alto rendimiento ha sido parte fundamental para el desarrollo de la investigación del instituto. Siendo parte de una gran cantidad de grupos de investigación nacional e internacional, colabora en proyectos de gran importancia a nivel mundial. Por tal motivo, actualmente el instituto cuenta con 3 clusters para el análisis de diversos proyectos científicos, destacando uno de ellos por el tipo de almacenamiento que posee ya que actualmente cuenta con un repositorio de alto rendimiento Lustre de 4.3 Petabytes, en el cual se almacenan los datos de diversos experimentos y principalmente los datos obtenidos por el Observatorio de Rayos Gamma HAWC (High-Altitude Water Cherenkov Gamma-Ray Observatory) [44]<sup>1</sup>. El instituto colabora en proyectos internacionales, uno de ellos es ALICE (A Large Ion Collider Experiment) del LHC (Large Hadron Collider) en el CERN.<sup>2</sup>

Actualmente en el ICN, el acceso a los datos es por medio de línea de comando o protocolos con interfaces poco amigables (ssh, rsync, sftp), que limitan la interacción y el uso de los datos. En la mayoría de los casos donde el usuario requiere datos locales, realiza la descarga específica a cada uno de los dispositivos donde la requiera, complicando el manejo y la colaboración de los datos científicos. Aunado a esta limitada interacción, el uso del almacenamiento en la nube por parte de los investigadores del instituto es cada vez más necesario, ya que constantemente se requiere tener la mayor cantidad de información disponible para poder ser compartido y con la posibilidad de colaborar con los diferentes grupos de trabajo internacionales, por lo que actualmente han tenido la necesidad de usar los servicios comerciales que también presentan problemas pues, en algunos casos, limitan la capacidad de almacenamiento o la velocidad de acceso y sincronía.

---

<sup>1</sup><https://www.hawc-observatory.org/>

<sup>2</sup>[http://alice.nucleares.unam.mx/experimento\\_alice](http://alice.nucleares.unam.mx/experimento_alice)

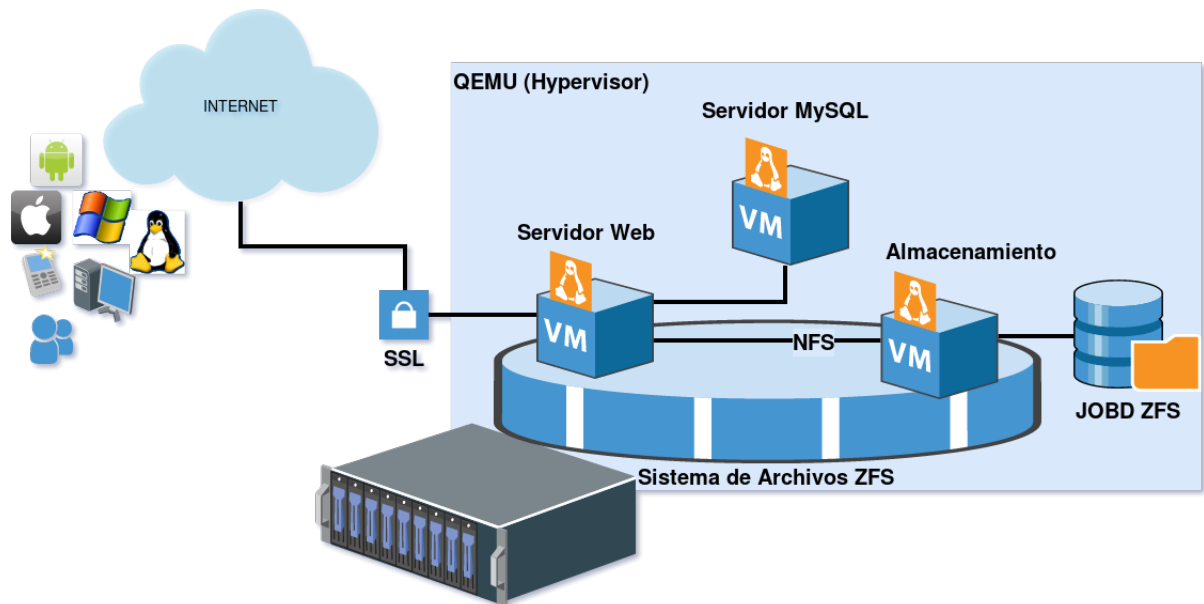


El cluster más grande del ICN, cuenta con un almacenamiento primario Lustre de alto rendimiento de 4.3PB en el cual se tiene un acceso de 20 GB/s en escritura y 12 GB/s en lectura. Consta de 21 nodos OSS que administran 20 OST por cada OSS y 2 MDS en alta disponibilidad, conectados por medio de Infiniband FDR/Ethernet con 42 nodos de procesamiento. También cuenta con un almacenamiento secundario NFS de 24TB en RAID para los usuarios de “Home”. Los otros cluster, cuentan con almacenamiento NFS comunicados por Infiniband a los nodos. Por tal motivo, la interacción de estos almacenamientos con los usuarios es muy importante y el beneficiarse con las herramientas y funcionalidades de la nube, será productivo en el uso de los sistemas de alto rendimiento.

## 5.2. Objetivos

Implementar un prototipo de almacenamiento en la nube con acceso a los clusters, y que cuente con los servicios de sincronización y uso compartido, para resolver los problemas de acceso a la información y promover una nueva forma integrada de acceso a los datos para la investigación científica. El prototipo está basado en una instalación de software libre ownCloud, el cual soporta todas las necesidades que requiere el instituto. Se realizaran pruebas para conectar este servicio a los sistemas de almacenamiento de alto rendimiento de los cluster del ICN.

Con este prototipo se pretende dar solución a los requerimientos de almacenamiento seguro en la nube que necesitan los usuarios e integrar la infraestructura de almacenamiento del ICN, lo cual permitirá definir políticas de acceso específicas a ciertos datos.



**Figura 5.1** Diagrama de la instalación

Tabla 5.1 Especificación de Servidores

MV Servidor Web		MV Servidor MySQL		MV Servidor Almacenamiento	
Linux	Ubuntu 16.04	Linux	Ubuntu 16.04	Linux	CentOS 6.9
CPU Cores	2 Cores	CPU Cores	2 Cores	CPU Cores	16 Cores
RAM	2GB	RAM	32GB	RAM	110 GB
Disco Duro	50GB	Disco Duro	2TB	Almacenamiento	10TB ZFS
Servidor Web	Apache 2.4	Base de Datos	MariaDB V10	Conexión	NFS-Utills
PHP	PHP V7.0				

### 5.3. Instalación

Este prototipo se implementó utilizando un ambiente virtualizado con “QEMU”<sup>3</sup> como hipervisor de código abierto y sin costo, con la finalidad de aprovechar la flexibilidad, rapidez en la instalación y para analizar el desempeño de esta aplicación en estos ambientes, ya que se puede reducir costos de inversión en infraestructura al no tener costo ni licencia por su uso.

Se implementará utilizando 3 servidores virtuales (MV) independientes entre ellos, los cuales van a estar administrados por la aplicación ownCloud de la siguiente forma (ver fig. 5.1):

- **Servidor para la base de datos SQL:** el objetivo de este servidor es ser utilizado como espacio de nombres del sistema de archivos. Cuando el almacenamiento es un sistema de archivos local, la base de datos se utiliza como un caché por medio de un comando de sincronización, pero cuando el almacenamiento es un sistema de archivos basado en objetos, la base de datos es fundamental por la información almacenada en ella, donde esta toda la estructura e información de los datos, por lo que al perder esta información el usuario puede perder sus archivos.
- **Servidor para el acceso web:** es la parte del sistema que procesa y se encarga del uso compartido de los archivos y la sincronización. Cuenta con conexiones directas al almacenamiento como a las bases de datos, por lo que recibe todas las operaciones de datos y metadatos. Además, es donde se pueden definir políticas de acceso a la base de datos para mejorar el uso compartido, como restricción o permisos a dominios completos, accesos restringidos a usuarios en específico, entre otros.
- **Almacenamiento:** es la parte del sistema que guarda los archivos del usuario, es decir, los datos. Los algoritmos de sincronización y los servicios compartidos pueden usar varios almacenamientos diferentes, desde sistemas de archivos locales hasta almacenamientos de objetos. Esto es posible por

<sup>3</sup><https://www.qemu.org/>

medio de sistema de archivos virtual (VFS), los cuales permiten un acceso transparente y estándar a diferentes sistemas de archivos muy particulares, generando una flexibilidad importante para la aplicación y creando una línea de desarrollo muy interesante y con mucho uso a futuro.

- **Servidor para autenticación (opcional):** el objetivo de este servidor es tener la posibilidad de recopilar una base de datos de usuario externa, brindando mayor integración con infraestructuras locales. La conexión más usada es la bases de datos LDAP (Active Directory) y OAuth.

### 5.3.1. Disco principal ZFS

El sistema de archivos “Z” es un administrador de volúmenes lógicos de código libre y abierto construido por Sun Microsystems para su uso en su sistema operativo Solaris[45]. A continuación se mencionan algunas de las características más importantes[46]:

- Alta capacidad de escalabilidad: Al ser un sistema de archivos de 128 bits que es capaz de administrar grandes cantidades de datos (del orden de zettabytes, es decir, mil millones de terabytes).
- Integridad máxima: Dentro de ZFS utiliza una suma de comprobación para garantizar la integridad del archivo y es capaz de realizar reparaciones automáticas (depuración o scrubbing).
- Agrupación de unidades: Cuenta con agrupación de discos que facilita el crecimiento del espacio de almacenamiento.
- RAID: Cuenta con la capacidad de almacenar la información, basado en algoritmos redundantes manteniendo un rendimiento comparable a una controladora de hardware, permitiendo ahorrar en infraestructura al no requerir hardware especializado.
- Compresión de datos en tiempo real: Característica que mejora la velocidad de E/S al utilizar algoritmos de última generación como LZ4.

En el servidor principal se crea un arreglo de discos ZFS, para tener redundancia y alto desempeño. En donde se instalarán las máquinas virtuales y se tendrán los datos de los usuarios del ownCloud. Con esto se reduce la probabilidad de pérdida de datos y del servicio por una falla en algún disco duro.

- Se crean dos arreglos raidz1<sup>4</sup>, permite un disco de paridad por cada arreglo y reparte los bloques en todos los discos reduciendo un cuello de botella (IOPS<sup>5</sup>) al no leer todo el bloque en un solo

---

<sup>4</sup><https://blogs.oracle.com/ahl/what-is-raid-z>

<sup>5</sup><http://www.datacenterjournal.com/raid-and-iops-and-io-observations/>

disco. Con esto se mejora el rendimiento de E/S y la velocidad en el cálculo de paridad. Quedaría un arreglo de 4 discos de 3T y otro de 3 discos de 3T, ambos permiten la misma redundancia que un solo arreglo de 7 discos, pero con mejor desempeño al momento de reconstruir el arreglo en caso de pérdida de algún disco<sup>6</sup>.

```
[root@OwnCloud ~]# zpool create -f storage raidz1 sdb sdc sdd sde raidz1 sdf sdg sdh
```

- Se habilita la función de comprimir con lz4 ya que existen varios análisis donde el rendimiento es mejorado al comprimir con el algoritmo lz4<sup>7 8 9</sup> [47] [48].

```
[root@OwnCloud ~]# zfs set compression=lz4 storage
```

- Se habilita la función “dedup” que evita el almacenamiento de archivos duplicados

```
[root@OwnCloud ~]# zfs set dedup=on storage
```

- Se crea una partición (dataset-name) sobre el arreglo para almacenar los discos de las máquinas virtuales, y se define el espacio de almacenamiento de la partición “servers” a 2.9T.

```
[root@OwnCloud ~]# zfs create storage/servers  
[root@OwnCloud ~]# zfs set quota=2.9T storage/servers
```

- Se crea una partición (dataset-name) sobre el arreglo para almacenar los datos de ownCloud, y se define el espacio de almacenamiento de la partición “data” a 10T.

```
[root@OwnCloud ~]# zfs create storage/data  
[root@OwnCloud ~]# zfs set quota=10T storage/data
```

<sup>6</sup><https://calomel.org/zfs RAID speed capacity.html>

<sup>7</sup><https://ieeexplore.ieee.org/document/6824430/>

<sup>8</sup>[https://link.springer.com/chapter/10.1007/978-1-4842-3306-1\\_1](https://link.springer.com/chapter/10.1007/978-1-4842-3306-1_1)

<sup>9</sup><https://www.servethehome.com/the-case-for-using-zfs-compression/>

- Se despliega la lista de los arreglos.

```
[root@OwnCloud ~]# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
storage             4.85G  12.9T   134K   /storage
storage/data        345M   10.0T   345M   /storage/data
storage/servers     4.45G   2.90T   4.45G   /storage/servers
```

```
[root@OwnCloud ~]# zpool status
pool: storage
state: ONLINE
scan: none requested
config:
   NAME        STATE      READ  WRITE  CKSUM
   storage    ONLINE          0     0     0
     raidz1-0  ONLINE          0     0     0
       sdb     ONLINE          0     0     0
       sdc     ONLINE          0     0     0
       sdd     ONLINE          0     0     0
       sde     ONLINE          0     0     0
     raidz1-1  ONLINE          0     0     0
       sdf     ONLINE          0     0     0
       sdg     ONLINE          0     0     0
       sdh     ONLINE          0     0     0
```

### 5.3.2. QEMU

QEMU es un emulador de un sistema completo, es decir, puede emular diferentes tipos de CPU y dispositivos. Su nombre proviene de abreviatura de “Quick Emulator”, el cual emula los CPU a través de la traducción binaria dinámica por medio de un método que traduce el código nativo de un procesador (código de operación de la computadora invitada o guest), en funciones de software que son compatibles con el CPU que está utilizando en la computadora principal (Anfitrión o Host). Conforme se incrementa el rendimiento de los CPUs y las mejoras en virtualización directa al CPU, la polémica sobre la diferencia en el desempeño a un sistema real es minimizada por la reducción de costos y el desarrollo constante de los hipervisors que mejoran cada vez más el desempeño. QEMU utiliza KVM (Kernel Virtual Machine) como acelerador para la ejecución de código directamente en el CPU, utilizando las extensiones de virtualización Intel VT-x disponibles en la mayoría de las CPU modernas, ejecutando máquinas virtuales a una velocidad casi nativa, lo que le da mejor desempeño que otros hipervisors. (Fig. 5.2) Hoy en día, esta es una de las opciones más usadas en los centros de datos, ya que permite optimizar los recursos y minimizar las pérdidas en el rendimiento. [49]

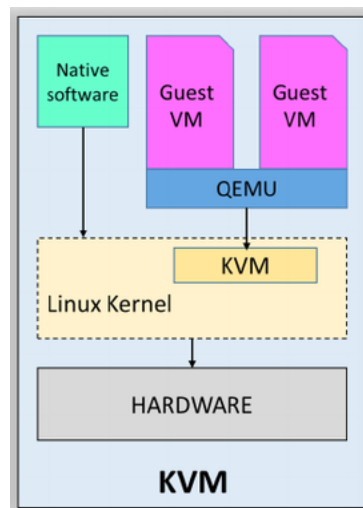


Figura 5.2 QEMU

### 5.3.3. Red QEMU para las máquinas virtuales

Para este prototipo es importante que la red cuente con buen rendimiento, por lo que en ambientes virtualizados se usan puentes virtuales (bridges) que están implementados para ser ejecutados en el núcleo (kernel) del sistema anfitrión y a su vez se crean dispositivos “TAP” para crear interfaces de redes virtuales asignadas a cada máquina virtual. Con estos dispositivos virtuales se puede tener comunicación entre las máquinas virtuales y que a su vez los usuarios tengan acceso desde el Internet (ver fig. 5.3).

Al crear un puente virtual y asignar los dispositivos “TAP” como interfaces de red a cada máquina virtual, se crea una red local con la cual se comunican entre ellas, y por medio de reglas de control de flujo “iptables” se controla la comunicación hacia Internet a través de interfaces físicas del sistema principal (Host) como eth0.

Aunado a esto, para tener un mejor desempeño, se puede asignar a las máquinas virtuales dispositivos virtuales de red controlados por “VirtIO”<sup>10</sup>, los cuales tienen mejor desempeño y baja sobre carga. [50]

<sup>10</sup><https://wiki.libvirt.org/page/Virtio>

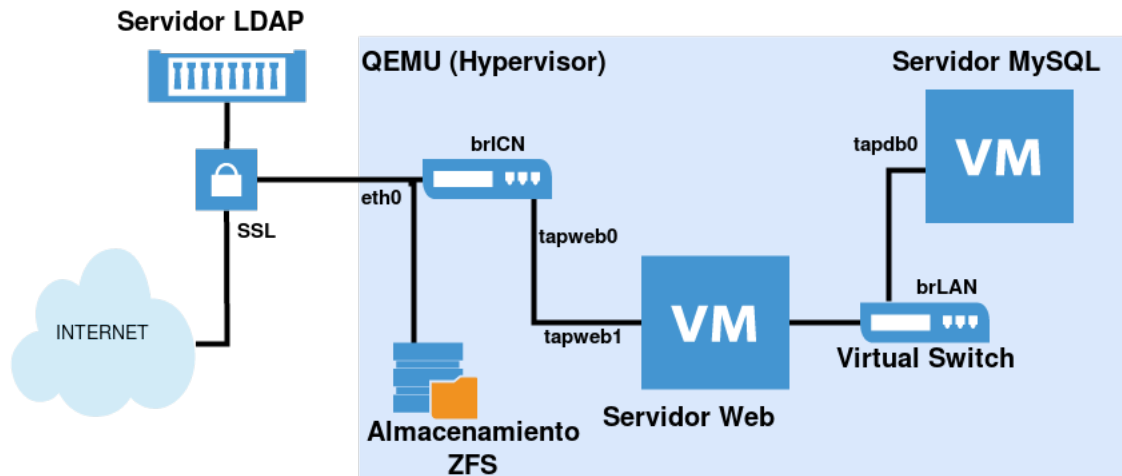


Figura 5.3 Red QEMU

- Se crean los discos duros de los servidores Web y MySQL.

```

#$ qemu-img create -f raw /storage/servers/qemu/webServer.raw 50G
#$ qemu-img create -f raw /storage/servers/qemu/MySQLServer.raw 2T

```

- Instalación de la máquina virtual “Web Server”.

```

#$ qemu-kvm -cpu host -smp 4 -cdrom ubuntu-16.04.4-server-amd64.iso \
-drive file=webServer.raw,if=virtio,index=0,media=disk \
-boot order=dc -m 2048 -k es -net nic,vlan=0,model=virtio \
-net tap,vlan=0,ifname=tapweb0,script=no,downscript=no \
-net nic,vlan=1,model=virtio \
-net tap,vlan=1,ifname=tapweb1,script=no,downscript=no \
-enable-kvm -vnc :177 -usbdevice tablet -name VM_SERVER_WEB

```

- **-cpu host**: Arquitectura a emular. “Host” emula la misma arquitectura que el anfitrión.
- **-smp 2**: Cantidad de núcleos que el invitado puede usar.
- **-cdrom ubuntu-16.04.4-server-amd64.iso**: Unidad de CDROM virtual y monta la imagen.
- **-drive file=webServer.raw,if=virtio,boot=on**: Configuración de un disco duro virtual.
- **-boot order=dc**: Orden de inicio, D=CD, C=Disco Virtual.
- **-m 2048**: Cantidad de Memoria.
- **-k es**: Tipo de teclado.
- **-net nic,vlan=0,model=virtio**: Interface de red virtual a través del controlador VirtIO.

- **-net tap,vlan=0,ifname=tapweb0,script=no,downscript=no**: Switch entre anfitrión y invitado.
  - **-enable-kvm**: Se habilita el soporte de KVM.
  - **-vnc :177**: Configuración del puerto para despliegue VGA de la máquina virtual a VNC.
  - **-usbdevice tablet**: Configuración de dispositivos USB. “Tablet” habilita el ratón.
  - **-daemonize**: Envía el proceso a segundo plano como demonio.
  - **-name VM\_SERVER\_WEB**: Nombre de la máquina virtual.
- Instalación de la máquina virtual “Servidor DB” para metadatos y control de ownCloud.

```
# $ qemu-kvm -cpu host -smp 2 -cdrom ubuntu-16.04.4-server-amd64.iso \
-drive file=MySQLServer.raw,if=virtio,index=0,media=disk \
-boot order=dc -m 32000 -k es -net nic,vlan=0,model=virtio \
-net tap,vlan=0,ifname=tapDB0,script=no,downscript=no \
-net nic,vlan=,model=virtio -enable-kvm -vnc :178 -usbdevice tablet -name VM_SERVER_DB
```

- **-m 32000**: Cantidad de Memoria.
- **-net nic,vlan=0,model=virtio**: Sólo una interface de red virtual a través del controlador VirtIO.
- **-net tap,vlan=0,ifname=tapweb0,script=no,downscript=no**: Switch entre anfitrión y invitado.
- **-enable-kvm**: Se habilita el soporte de KVM.
- **-vnc :178**: Configuración del puerto para despliegue VGA de la máquina virtual a VNC.
- **-usbdevice tablet**: Configuración de dispositivos USB. “Tablet” habilita el ratón.
- **-name VM\_SERVER\_DB**: Nombre de la máquina virtual.

#### 5.3.4. Configuración de las máquinas virtuales

- Red para las máquinas virtuales Servidor Web. Como pudimos ver en el diseño del prototipo (Fig.5.3), en esta máquina virtual “Servidor Web” se requiere que tengan dos interfaces de red. Una conectada a Internet (acceso externo) y otra conectada al puente virtual de red local.



```

#=====
# Servidor Web: /etc/network/interfaces
#==> Red Local
auto ens4
iface ens4 inet static
    address 192.168.0.1
    netmask 255.255.255.0
    broadcast 192.168.0.255
    network 192.168.0.0
    dns-nameserver 132.248.204.1 132.248.10.2
    dns-domain nucleares.unam.mx
    dns-search unam.mx
#=====

```

- Red para las máquinas virtuales Servidor DB. Sólo se requiere acceso a la red local.

```

#=====
# Servidor DB: /etc/network/interfaces
#==> Red Local
auto ens4
iface ens4 inet static
    address 192.168.0.10
    netmask 255.255.255.0
    broadcast 192.168.0.255
    network 192.168.0.0
    gateway 192.168.0.1
    dns-nameserver 132.248.204.1 132.248.10.2
    dns-domain nucleares.unam.mx
    dns-search unam.mx
#=====

```

### 5.3.5. Instalación y configuración del Servidor NFS

Se instala el servidor NFS en el host principal, para simplificar el prototipo y reducir la sobre carga relacionada sobre el sistema invitado (Guest Host).

```

#$ yum install nfs-utils nfs-utils-lib

```

```

#$ yum install nfs-utils nfs-utils-lib

## Configurar el servicio NFS

#$ chkconfig nfs on
#$ chkconfig rpcbind on
#$ service rpcbind start
#$ service nfs start

```

- Configuración del directorio ZFS que se va compartir por NFS.

```
#=====
# Servidor NFS: /etc/exports
#=====
/storage/data 10.10.3.73(rw,no_root_squash,async,no_subtree_check)
```

- Configuración del NFS en el Servidor Web

```
#=====
# Servidor NFS: /etc/fstab
#=====
10.10.3.65:/storage/data /ownCloud-nfs-data nfs defaults,rsize=8192,wsiz=8192,bg,soft,nfsvers=4 0 0
```

### 5.3.6. Instalación de la aplicación ownCloud

- Aplicaciones y software principal requerido para la instalación <sup>11</sup>:
  - Servidor Web: Apache versión 2.4
  - PHP versión 7 con módulos adicionales.
  - Servidor Redis: Manejo en memoria de sesión persistentes para PHP, bloqueo de archivos.
  - Cliente para conexiones NFS.

```
#$ apt-get install -y apache2 mariadb-server libapache2-mod-php7.0 openssl php-imagick php7.0-common \
php7.0-curl php7.0-gd php7.0-imap php7.0-intl php7.0-json php7.0-ldap php7.0-mbstring \
php7.0-mcrypt php7.0-mysql php7.0-pgsql php-smbclient php-ssh2 php7.0-sqlite3 php7.0-xml \
php7.0-zip php-apcu php-redis redis-server php7.0-ldap

#$ apt-get install nfs-common
```

<sup>11</sup>[https://doc.owncloud.org/server/10.0/admin\\_manual/installation/source\\_installation.html](https://doc.owncloud.org/server/10.0/admin_manual/installation/source_installation.html)

## ■ Configuración del Servidor Web Apache

```

#=====
# Servidor Web: /etc/apache2/sites-available/owncloud.conf
#=====

Alias /owncloud "/var/www/owncloud/"

<Directory /var/www/owncloud/>

    Options +FollowSymlinks
    AllowOverride All

    <IfModule mod_dav.c>
        Dav off
    </IfModule>

    SetEnv HOME /var/www/owncloud
    SetEnv HTTP_HOME /var/www/owncloud
</Directory>

```

```
#$ service apache2 restart
```

## ■ Instalación y configuración del Servidor DB.

- Base de datos: MariaDB versión 10.

```

#=====
# Servidor MariaDB: /etc/mysql/mariadb.conf.d/50-server.cnf
#=====

...
user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port           = 3306
basedir         = /usr
datadir         = /var/lib/mysql
tmpdir          = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
...
bind-address    = 0.0.0.0  ## Permisos para conexiones remotas
...

```

- Se crea la base de datos en la máquina virtual “serverDB”:

```
dbServer#$ mysql -p -u root
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 18412
Server version: 10.0.34-MariaDB-0ubuntu0.16.04.1 Ubuntu 16.04
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

DB> use mysql;
DB> CREATE DATABASE ownCloud;
DB> CREATE USER 'owncloudusr'@'localhost' IDENTIFIED BY 'x';
DB> GRANT ALL PRIVILEGES ON ownCloud.* TO 'owncloudusr'@'localhost' IDENTIFIED BY 'x' WITH GRANT OPTION;
DB> GRANT ALL PRIVILEGES ON ownCloud.* TO 'owncloudusr'@'192.168.0.1' IDENTIFIED BY 'x' WITH GRANT OPTION;
DB> FLUSH PRIVILEGES;
```

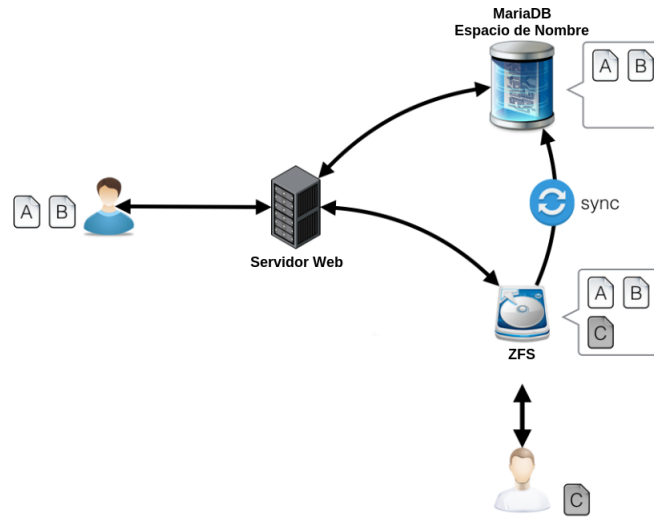
```
#$ wget https://download.owncloud.org/community/owncloud-10.0.8.tar.bz2
#$ tar -xjf owncloud-x.y.z.tar.bz2
#$ cp -r owncloud /var/www
#$ chown -R www-data:www-data /var/www/owncloud/
```

Por medio del navegador se ingresa al instalador del ownCloud para la configuración de los siguientes datos: <http://10.10.3.73/owncloud>

- Directorio para el almacenamiento de datos (/ownCloud-nfs-data)
- Configuración del tipo, usuario y contraseña de la base de datos MariaDB
- Configuración de permisos

La plataforma de sincronización ownCloud cuenta con tres componentes básicos:

- Base de datos SQL: Se utiliza como espacio de nombres del sistema de archivos con el cual proporciona direccionamiento de los nombres. Cuando el almacenamiento es local, la base de datos se utiliza como un caché.
- Servidor web: Es la parte del sistema que controla la sincronización y la interfaz web. Se conecta directamente con el almacenamiento y con la base de datos.
- Almacenamiento: Es donde el sistema guarda los archivos y datos de los usuarios. El sistema acepta una variedad de almacenamientos diferentes que van desde sistemas de archivos locales hasta almacenamientos de objetos. Esta flexibilidad se logra utilizando una interfaz abstracta de sistema de archivo virtual (VFS).



**Figura 5.4** Esquema de sincronización y acceso al almacenamiento

El objetivo del prototipo es crear un acceso amigable a los repositorios científicos y almacenamientos de los clusters del ICN, pero uno de los problemas que presenta la plataforma de sincronización es el uso de una base de datos relacional como espacio de nombre, la cual requiere estar sincronizada con el almacenamiento para tener actualizados los metadatos. Si un usuario tiene acceso externo y modifica archivos, la base de datos del sistema que contiene el espacio de nombre requiere realizar un proceso de sincronía para actualizar dichos cambios (ver fig. 5.4). Una solución al problema anterior es por medio de un proceso de sincronía, que se debe ejecutar frecuentemente, para mantener la base de datos actualizada:

```
# $ sudo -u www-data php /var/www/owncloud/occ files:scan --all
```

# Capítulo 6

## Conclusiones

En este trabajo, se pudo analizar y entender la importancia que hoy en día tiene el almacenamiento de los datos y su crecimiento. También se estudió la evolución de los sistemas de cómputo científico desde sus inicios, hasta la importancia que han tenido los sistemas de alto rendimiento tanto de almacenamiento como de procesamiento, donde toda esta evolución, cada vez se hace más necesaria, por lo tanto, es de gran importancia conocer sus características, para poder implementarlas en la actualidad.

Conforme se avanza en este trabajo, se conoció cronológicamente los orígenes, evolución, características y estructuras de los sistemas en la nube. Y principalmente darnos cuenta de la tendencia que han tenido los sistemas de cómputo de propósito general con relación a los sistemas de alto rendimiento, ya que en los inicios de ambos sistemas, la relación y la dependencia era impensable, pero hoy en día son sistemas que conviven y comparten muchas de las características, incluso en algunas aplicaciones fusionando las características.

Los servicios en la nube, ya son indispensables en la actualidad y la única forma de que se puedan aprovechar es conociendo todas las tecnologías como virtualización y HPC. La computación en la nube representa el siguiente paso evolutivo hacia las Tecnologías de la Información elástica. El surgimiento de la nube transforma la forma en que se diseña y rige la infraestructura a través de servicios para infraestructura, plataforma y aplicaciones.

Las aplicaciones de la nube como ownCloud, son servicios que han crecido en los últimos años, a tal grado que son parte de los servicios básicos para cualquier negocio o institución científica, debido a los beneficios que ofrece. Por esto, la integración de esta interfaz es de gran importancia basándonos en los casos de éxito como el del CERNBox, originando una demanda desconocida de estos servicios en los sistemas de alto desempeño, demostrando que son plataformas innovadoras para el cómputo científico. Siendo éste un segmento de usuarios que normalmente los sistemas o modelos comerciales no cumplen con las especificaciones requeridas, ya sea por las aplicaciones tan especializadas que se cuentan, o por el

enfoque no comercial que en la mayoría de los centros científicos no es su principal objetivo.

Las tendencias y las pruebas realizadas en diversos centros de investigación demuestran que estos servicios tienen potencial de convertirse en el principal acceso a los datos de los usuarios, ya que cuentan con la movilidad requerida, con un acceso multiplataforma y con la capacidad de escalamiento de estos sistemas.

Se instaló un prototipo para el acceso amigable a los repositorios científicos y almacenamientos de clusters, donde se pudieron realizar pruebas. Se encontraron los límites de la implementación inicial y básica, como el problema del acceso directo al almacenamiento y la falta de sincronía en los metadatos. En las pruebas realizadas, esta opción requiere que el sistema de ownCloud ejecutar un proceso de sincronización donde lee todo el almacenamiento externo periódicamente en busca de actualizaciones, limitando el escalamiento en instalaciones más grandes.

También se encontró una implementación modificada que da solución a este problema, donde delegan el manejo de los metadatos a los sistemas de archivos de gran escala como se menciona en el CERNBox [42] [51].

Con este trabajo se crea un prototipo para poder realizar pruebas en el Instituto de Ciencias Nucleares para dotarlo de un acceso a los repositorios locales de una forma amigable y con un sistema más robusto, siendo uno de los servicios que los usuarios solicitan constantemente.

Considero que con este trabajo se cumple con los objetivos propuestos, ya que al tener toda la información, características, análisis, pros y contras de estos sistemas, nos ayuda para la implementación más adecuada para las necesidades del ICN y también se puede aprovechar la infraestructura actual para reducir costo. Aunado a esto, considero que toda la recopilación de artículos de investigación que use para este trabajo servirá como base para la continuación de más proyectos dentro del instituto, pues actualmente cuenta con uno de los almacenamientos más grandes de Latinoamérica y ha sido pionero en infraestructura para almacenamiento dentro de la UNAM.

## 6.1. Proyectos a futuro

Este trabajo plantea la continuidad de la instalación del sistema ownCloud robusto y que cumpla con las necesidades del ICN, y a su vez, probar las soluciones realizadas en el proyecto CERNBox y que a continuación se mencionan:

- Realizar pruebas habilitando la interfaz WebDAV<sup>1</sup> y el protocolo de sincronización de ownCloud

---

<sup>1</sup>WebDAV: Web Distributed Authoring and Versioning

(OSP)<sup>2</sup>, para delegar el control de la sincronización al sistema de archivos de alto rendimiento[52]. Con estas pruebas se podría ampliar el uso del sistema, ya que se podrá facilitar el acceso a más aplicaciones y en algunos casos, mejorar el rendimiento en el acceso a los datos.

- Realizar las nuevas formas de acceso FUSE con las que cuenta el ownCloud, ya que esto mejoraría el acceso a los datos desde las computadoras locales, al eliminar el cliente de sincronización y que requieran despliegue de datos más sofisticados.

---

<sup>2</sup><https://github.com/cernbox/smashbox/blob/master/protocol/protocol.md>





# Bibliografía

- [1] Andrew Burger. *U.S. Tops in Global Data Storage Capacity*. International Data Corporation (IDC), 2013. <http://www.telecompetitor.com/idc-u-s-tops-in-global-data-storage-capacity/>.
- [2] Reinsel D. Gantz J. The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east. *IDC's Digital Universe Study, December 2012, sponsored by EMC.*, 2012.
- [3] Rajnikant Palwe Gurudatt Kulkarni, Rani Waghmare. Cloud storage architecture. *7th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, 2012.
- [4] Ramnath Chellappa. Intermediaries in cloud-computing: A new computing paradigm. *INFORMS*, 1997. <http://cloudcomputing.sys-con.com/node/593313>.
- [5] George Favaloro. *Internet Solutions Division Strategy for Cloud Computing*, November 14, 1996. COMPAQ COMPUTER CORPORATION business plan, CST presentations.
- [6] Discussion forum Slashdot Community. *Multiple Experts Try Defining Cloud Computing*, 2008. <http://tech.slashdot.org/article.pl?sid=08/07/17/2117221>.
- [7] Trevor Doerksen. Cloud computing - the user-friendly version of grid computing. *SYS-CON INDIA*, 2008. <http://cloudcomputing.sys-con.com/node/593313>.
- [8] Kevin Hartig. What is cloud computing? *SYS-CON*, 2008. <http://cloudcomputing.sys-con.com/node/579826>.
- [9] S.Satyanarayana. Cloud computing : Saas. *GESJ: Computer Science and Telecommunications*, 2012.
- [10] S. Shanmugapriya R. Arokia Paul Rajan. Evolution of cloud storage as cloud computing infrastructure service. *IOSR Journal of Computer Engineering (IOSRJCE)*, 1:38–45, 2012. [www.iosrjournals.org](http://www.iosrjournals.org).

- [11] Lisa Marie Segarra and Drew Houston. *The Founder of Dropbox Reveals How He Got the Idea and It Involves a Chinatown Bus*, 2017. <http://fortune.com/2017/06/14/founder-dropbox-got-idea-chinatown-bus/>.
- [12] H. A. Sanjay J. P. Ashwini, C. Divya. Efficient resource selection cluster/grid framework to this reason enable cloud for hpc applications. *4th International Conference on Computer and Communication Technology, IEEE*, 2013. DOI:10.1109/ICCCT.2013.6749599.
- [13] Marco A. S. Netto, Rodrigo N. Calheiros, Eduardo R. Rodrigues, Renato L. F. Cunha, and Rajkumar Buyya. Hpc cloud for scientific and business applications: Taxonomy, vision, and research challenges. *ACM Computing Surveys (CSUR) Volume 51 Issue 1, January 2018 Article No. 8*, 2018. doi: 10.1145/3150224.
- [14] Rajkumar Buyya Marcos Dias de Assuncao, Alexandre di Costanzo. Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters. *HPDC, In Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing*, 2009. doi: 10.1145/1551609.1551635.
- [15] Miguel G. Xavier, Marcelo V. Neves, Fabio D. Rossi, Tiago C. Ferreto, Timoteo Lange, and Cesar A. F. De Rose. Performance evaluation of container-based virtualization for high performance computing environments. *21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, 2013.
- [16] Jacopo Soldani Claus Pahl, Antonio Brogi and Pooyan Jamshidi. Cloud container technologies: a state-of-the-art review. *IEEE Transactions on Cloud Computing*, 2017.
- [17] Xiaoyi Lu Jie Zhang and Dhabaleswar K. High performance mpi library for container-based hpc cloud on infiniband clusters. *45th International Conference on Parallel Processing*, 2016.
- [18] James R. Fischer. The roots of beowulf. *NASA Goddard Space Flight Center*, 2015. <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150001285.pdf>.
- [19] Georg Hager and Gerhard Wellein. *Introduction to High Performance Computing for Scientists and Engineers*. CRC press, 1 edition, 2011.
- [20] David E. Bernholdt, Wael R. Elwasif, James A. Kohl, and Thomas G. W. Epperly. A component architecture for high-performance computing. *Proceedings of the Workshop on Performance Optimization via High-Level Languages and Libraries (POHLL-02)*, 2002.

- [21] Richard Gerber. Introduction to high performance computers. *Proceedings of the National Energy Research Scientific Computing Center*, 2013. <https://www.nersc.gov/assets/Uploads/Gerber-HPC-2.pdf>.
- [22] Ian Foster and Carl Kesselman. *The Grid 2: Blueprint for a new computing infrastructure*. Elsevier, 2 edition, 2003.
- [23] Tran Doan Thanh, Subaji Mohan, Eunmi Choi, SangBum Kim, and Pilsung Kim. A taxonomy and survey on distributed file systems. *Fourth International Conference on Networked Computing and Advanced Information Management, IEEE Computer Society*, 2008. DOI 10.1109/NCM.2008.162.
- [24] Dave Mitchell Michael Condict, Don Bolinger and Eamonn McManus. Microkernel modularity with integrated kernel performance. In *Proceedings of the First Symposium on Operating Systems Design and Implementation (OSDI 1994)*, Nov 1994.
- [25] Hermann Hartig, Michael Hohmuth, Jochen Liedtke, Jean Wolter, and Sebastian Schnberg. The performance of micro-kernel-based systems. *SIGOPS Oper. Syst. Rev.*, 31(5):66–77, October 1997.
- [26] Yanlong Yin Shuibing He, Xian-He Sun. Bps: A performance metric of i/o system. *IEEE 27th International Symposium on Parallel and Distributed Processing Workshops*, 2013. <http://ieeexplore.ieee.org/document/6651099/metrics>.
- [27] Barbara Liskov, Sanjay Ghemawat, Robert Gruber, Paul Johnson, Liuba Shrira, and Michael Williams. Replication in the harp file system. In *SOSP, 91 Proceedings of the thirteenth ACM symposium on Operating systems principles*, 1991.
- [28] John R. Douceur and Roger P. Wattenhofer. Optimizing file availability in a secure serverless distributed file system. *20th IEEE Symposium on Reliable Distributed Systems*, 2001. <http://ieeexplore.ieee.org/document/969727/>.
- [29] Chandramohan A. Thekkath, Timothy Mann, and Edward K. Lee. Frangipani: A scalable distributed file system. In *SOSP, Proceedings of the sixteenth ACM symposium on Operating systems principles*, 1997.
- [30] Derek Weitzel, Brian Bockelman, Dave Dykstra, Jakob Blomer, and Ren Meusel. Accessing data federations with cvmfs. *Journal of Physics: Conference Series*, 898(6):062044, 2017.

- [31] R. Chansler, H. Kuang, S. Radia, and K. Shvachko. The hadoop distributed file system. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST 2010)(MSST)*, volume 00, pages 1–10, 05 2010.
- [32] J Blomer, P Buncic, I Charalampidis, A Harutyunyan, D Larsen, , and R Meusel. Status and future perspectives of cernvm-fs. *Journal of Physics: Conference Series*, 396(5):052013, 2012.
- [33] Phil Schwan. Lustre: Building a file system for 1,000-node clusters. *Proceedings of the Linux Symposium*, pages 380–286, 2003.
- [34] Yasushi Saito and Marc Shapiro. Optimistic replication. *ACM Comput. Surv.*, 37:42–81, 2005.
- [35] Liu Ying. Hadoop on lustre. *Breakthrough Storage Performance, LUG*, 2014. [http://cdn.opensfs.org/wp-content/uploads/2014/10/8-Hadoop\\_on\\_lustre-CLUG2014.pdf](http://cdn.opensfs.org/wp-content/uploads/2014/10/8-Hadoop_on_lustre-CLUG2014.pdf).
- [36] Britta Wulfig. Sun assimilates lustre filesystem. *Linux Magazine Online*, 2007. <http://www.linux-magazine.com/Online/News/Sun-Assimilates-Lustre-Filesystem?category=13402>.
- [37] Rich Brueckner. Oracle has kicked lustre to the curb. *Inside hpc Online*, 2011. <https://insidehpc.com/2011/01/inside-track-oracle-has-kicked-lustre-to-the-curb/>.
- [38] Inside HPC. Whamcloud aims to make sure lustre has a future in hpc. *Inside hpc Online*, 2010. <https://insidehpc.com/2010/08/whamcloud-aims-to-make-sure-lustre-has-a-future-in-hpc/>.
- [39] William Gropp, Ewing Lusk, and Rajeev Thakur. Using mpi-2: Advanced features of the message-passing interface. *MIT Press, Cambridge, MA*, 1999. <https://mitpress.mit.edu/books/using-mpi-2>.
- [40] P. H. Carns, W. B. Ligon III, R. B. Ross, and R. Thakur. Pvfs: A parallel file system for linux clusters. *Extreme Linux Workshop, Atlanta*, pages 317–328, 2000. [https://www.usenix.org/legacy/publications/library/proceedings/als00/2000papers/papers/full\\_papers/carns/carns\\_html/index.html](https://www.usenix.org/legacy/publications/library/proceedings/als00/2000papers/papers/full_papers/carns/carns_html/index.html).
- [41] Samuel Lang, Philip Carns, Robert Latham, Robert Ross, and William Allcock Kevin Harms. I/o performance challenges at leadership scale. *Proceedings of Supercomputing*, 2009. <http://www.mcs.anl.gov/papers/P1648.pdf>.
- [42] L Mascetti, H Gonzalez Labrador, M Lamanna, JT Mościcki, and AJ Peters. Cernbox + eos: end-user storage for science. *Journal of Physics: Conference Series*, 664(6):062037, 2015.
- [43] Jakub T Mościcki and Massimo Lamanna. Prototyping a file sharing and synchronization service with owncloud. *Journal of Physics: Conference Series*, 513(4):042034, 2014.

- 
- [44] Lukas Nellen. Procesar y almacenar los datos del observatorio hawc: un reto para el cómputo de alto rendimiento. *XVI Congreso Nacional de Astronomía — Monterrey*, 2017. [http://cnaastronomia.unam.mx/presentaciones/Plenarias/LUKAS\\_NELLEN\\_CNA2017.pdf](http://cnaastronomia.unam.mx/presentaciones/Plenarias/LUKAS_NELLEN_CNA2017.pdf).
- [45] Oracle Corporation. Solaris zfs administration guide. 2010.
- [46] Bill Moore Jeff Bonwick. The last word in file systems. 2011.
- [47] Sven Schmidt. *Efficient interaction between Lustre and ZFS for compression*. PhD thesis, University of Hamburg, Department of informatics Faculty for mathematics, informatics and natural sciences, 08 2017.
- [48] Niklas Bunge. *Quality of service improvement in ZFS through compression*. PhD thesis, University of Hamburg, Department of informatics Faculty for mathematics, informatics and natural sciences, 05 2017.
- [49] Fabrice Bellard. Qemu, a fast and portable dynamic translator. pages 41–46, 01 2005.
- [50] R. Warnke and T. Ritzau. *qemu-kvm & libvirt: 2010*. Books on Demand, 2010.
- [51] D. Hugo González Labrador. Cernbox: Petabyte-scale cloud synchronisation and sharing platform. *Escuela Superior de Ingeniería de Universidad de Vigo*, 2016.
- [52] Klass Freitag. The file sync algorithm of the owncloud desktop clients. *Workshop on Cloud Services for File Synchronisation and Sharing*, 2014.

# Índice alfabético

Almacenamiento de instancias, 10  
Almacenamiento de objetos, 11  
Almacenamiento de volumen, 10  
Almacenamiento en la nube, 7  
Antecedentes, 37

CERNBox, 33  
Conclusiones, 51

Evolución de la nube, 3

Implementación CERNBox con ownCloud, 35  
Importancia de los datos y su almacenamiento, 1  
Instalación, 39

Modelos de almacenamiento Virtual, 9

Objetivos, 38

Prototipo, 37  
Proyectos a futuro, 52

Sincronización de archivos, 17