UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS

PRUEBAS DE RENDIMIENTO DE UN SISTEMA DE ARCHIVOS PARALELO UTILIZANDO UNIDADES DE ALMACENAMIENTO NVME

T E S I N A

QUE PARA OPTAR POR EL GRADO DE:

ESPECIALISTA EN CÓMPUTO DE ALTO RENDIMIENTO

PRESENTA:

ROBERTO ANGELES MORA

TUTORES:

DR. LUKAS NELLEN FILLA

M. EN I. JUAN LUCIANO DÍAZ GONZÁLEZ

INSTITUTO DE CIENCIAS NUCLEARES

MIEMBROS DEL COMITÉ TUTOR:

M. EN C. JOSÉ LUIS GORDILLO RUÍZ

CENTRO DE CIENCIAS DE LA COMPLEJIDAD

CIUDAD UNIVERSITARIA, NOVIEMBRE 2021

Contenido.

Agradecimientos

A mi familia, a mi padre y a mi madre por siempre apoyarme en todos mis proyectos, a mis hermanas por ser compañeras en esta vida que siempre me animan a seguir adelante.

A la Universidad Nacional Autónoma de México por ser pilar fundamental en el desarrollo de este país, por todo lo que es y lo que nos puede brindar.

Al Posgrado en Ciencia e Ingeniería de la Computación, por haberme dado la oportunidad de cursar este programa de Especialización único en el país.

A mis Tutores, Dr. Lukas Nellen Filla y M. en I. Juan Luciano Díaz González, a mis Profesores, Dr. Ernesto Rubio Acosta, Dra. María Elena Lárraga Ramírez, M. en C. José Luis Gordillo Ruíz e Ing. Adrián Durán Chavesti, por su enorme talento y gran dedicación a la labor científica, por transmitirme tantos conocimientos y dedicar su tiempo a la docencia.

A mis compañeros de generación, por tantas experiencias compartidas, en especial al Fís. Antonio Ramírez Fernández y la Ing. Leticia Rojas Nava, grandes compañeros de trabajo, por todo el apoyo y esas interesantes charlas con café.

Al personal de la Unidad de Cómputo, Redes y Telecomunicaciones del Instituto de Ciencias Nucleares, por el apoyo técnico y las facilidades prestadas para la realización de este proyecto.

Al personal que integra la Coordinación de Supercómputo DGTIC-UNAM, Ing. Leobardo Itehua Rico, M. en I. Lourdes Yolanda Flores Salgado, L.I. Eduardo Iván Ortega Alarcón, Ing. Irving Álvarez Castillo e Ing. Silvia Elizabeth Frausto del Río; quienes compartieron su experiencia y conocimientos del área de Supercómputo.

Al proyecto UNAM PAPIIT IN110621.

A Ali, por tantos momentos compartidos y más, por todos los logros a lo largo de este año tan complicado y lleno de retos.

1. Introducción

El Instituto de Ciencias Nucleares (ICN), como parte del Subsistema de Investigación Científica de la Universidad Nacional Autónoma de México (UNAM), se ha distinguido en su labor académica y científica desde sus orígenes. Colabora activamente en la formación y preparación de personal altamente especializado en diversas áreas de las ciencias, así como en la prestación de servicios científicos y tecnológicos para investigaciones básicas y aplicadas, trabajo que realiza en conjunto con las Facultades de Ingeniería, Medicina, Química y Ciencias.

En el Instituto de Ciencias Nucleares se realizan investigaciones en diversas áreas de la ciencia tales como: Física de Plasmas, Física Nuclear y Molecular, Física de Altas Energías, Gravitación, Teoría de Campos, Química de Radiaciones y Dosimetría, de donde se desprenden los diversos programas académicos que se imparten, buscando ser un referente de excelencia académica realizando investigación teórica, experimental y aplicada, contribuyendo a la formación de profesionistas e investigadores a través de distintos programas de Licenciatura y Posgrado, difundiendo los resultados de los trabajos de investigación realizados en el Instituto, fomentando el desarrollo de las ciencias nucleares y prestando servicios técnicos tanto dentro de la UNAM como a instituciones públicas y privadas en las áreas de su competencia.

En la última década, el Instituto de Ciencias Nucleares ha establecido convenios de colaboración con distintas organizaciones e instituciones en proyectos internacionales, tales como: la NASA, donde hizo contribuciones en el proyecto de la misión espacial *Curiosity* a través de estudios y el diseño de experimentos, el CERN, donde contribuye a través del experimento ALICE y el Observatorio Pierre Auger de rayos cósmicos colaborando en diversos proyectos; entre otros, colabora activamente en el Observatorio HAWC de rayos gamma, dedicado a observar el Universo en las más altas energías; en conjunto con diversas instituciones académicas y de investigación, tanto nacionales como internacionales, principalmente estadounidenses y algunas europeas.

El Instituto de Ciencias Nucleares cuenta actualmente con uno de los Centros de Datos y Cómputo de Alto Rendimiento Académico más importantes a nivel nacional, donde se almacena una gran colección de datos científicos de diversos proyectos de investigación, tanto nacionales como internacionales. Está compuesto por 45 nodos de procesamiento y 52 servidores de almacenamiento, con los que se obtienen una capacidad de alrededor de 8 petabytes, utilizados principalmente en proyectos de investigación en las áreas de física de altas energías y de astrofísica en conjunto con el Instituto de Astronomía (IA) de la UNAM.

Tuvo su origen como el centro de cómputo para apoyar a los colaboradores del Observatorio Pierre Auger (Argentina) y hoy en día se encuentra vinculado a diversos proyectos como ALICE del CERN (Suiza) y el Laboratorio Nacional HAWC del cual almacenan la mayor parte de los datos generados de las observaciones e investigaciones realizadas en dicho proyecto.

Esta constante actividad científica ha sido factor para el crecimiento técnico de este Centro de Datos, el cual ha sido diseñado, desarrollado y puesto en funcionamiento dentro del mismo Instituto con el objetivo de apoyar los proyectos de investigación, aplicar la ciencia e ingeniería en cómputo, generar soluciones de bajo costo adaptándose a las necesidades y demandas de los investigadores, además de crear conocimiento tecnológico para la propia infraestructura de Supercómputo de la UNAM.

Es con estos proyectos y en general en toda su labor de investigación científica donde se requiere de los servicios y aplicaciones del Cómputo de Alto Rendimiento, siendo este Instituto, uno de los principales promotores e implementadores de estas tecnologías para el desarrollo de sus actividades.

En este Centro de Investigación también se entrena a los técnicos en Cómputo de Alto Rendimiento y Almacenamiento y se forman estudiantes en las mismas áreas, con lo que se busca potenciar y hacer crecer ese espacio.

2. Resumen

En este trabajo se busca describir algunas formas sencillas para probar el rendimiento de las unidades de almacenamiento NVMe, así como el procedimiento para la instalación y configuración de un sistema de archivos paralelo Lustre en su forma más básica.

El presente trabajo se compone de tres partes principales, en la primera parte (Marco Teórico) se describen de forma breve, los aspectos teóricos más importantes en torno al tema principal de este trabajo, en la segunda parte (Metodología de Investigación) se explica la forma en que se llevaron a cabo las pruebas en el equipo facilitado, así como los hallazgos encontrados, finalmente en la tercera parte (Resultados) se muestra de forma gráfica, los resultados obtenidos y se ofrece una breve interpretación de estos.

Así mismo, el lector podrá encontrar una serie de Apéndices donde se muestran en forma gráfica los resultados de todas las pruebas realizadas, así como un Glosario con los términos técnicos utilizados en el desarrollo del presente trabajo.

Con toda la experiencia del personal de la Unidad de Cómputo, Redes y Telecomunicaciones del Instituto de Ciencias Nucleares y con la necesidad de ampliar y mejorar las capacidades de la infraestructura con la que cuentan actualmente, es que surge el desarrollo de este proyecto, gracias a la visión del investigador Dr. Lukas Nellen Filla y la asesoría técnica del M. en I. Juan Luciano Díaz González y el Ing. Juan Eduardo Murrieta León.

3. Justificación

El Instituto de Ciencias Nucleares cuenta actualmente con un clúster de Cómputo de Alto Rendimiento y un sistema de archivos paralelo Lustre para el almacenamiento de los datos generados por los diferentes proyectos académicos y de investigación. Es en este último donde se busca hacer cambios sustanciales para mejorar su desempeño, a través del uso de unidades de almacenamiento NVMe para acelerar el análisis de los datos que se usan con mayor frecuencia.

El proyecto consiste en configurar un sistema de almacenamiento paralelo Lustre con equipos facilitados por personal de la Unidad de Cómputo del Instituto, de similares características a los utilizados en el clúster con el que cuenta el Instituto, empleando unidades de almacenamiento NVMe en lugar de discos duros tradicionales.

Con el desarrollo del presente proyecto se pretende observar el comportamiento de las unidades de almacenamiento NVMe y del sistema de archivos paralelo Lustre, mediante diversas pruebas de rendimiento en distintas condiciones, las pruebas de rendimiento medirán lo siguiente:

- Velocidad de lectura y escritura.
- Operaciones de Entrada/Salida por Segundo.
- Tiempo de Ejecución.

4. Marco Teórico

4.1. Cómputo de Alto Rendimiento

"El Cómputo de Alto Rendimiento se refiere generalmente, a la práctica de agregar potencia informática con el objetivo de ofrecer un rendimiento mucho mayor que el que se podría obtener de una computadora de escritorio o estación de trabajo típica para resolver grandes problemas de la ciencia, la ingeniería o los negocios."

Para lograr esos objetivos, se auxilia de tecnologías computacionales como clústeres de computadoras, supercomputadoras o la computación paralela. Por lo que un sistema de Cómputo de Alto Rendimiento es esencialmente un conjunto de nodos interconectados entre sí, los cuales contienen uno o más procesadores, así como su propia memoria.

La necesidad de tener la capacidad para procesar una gran cantidad de información en el menor tiempo posible ha dado lugar al desarrollo de la computación paralela, la cual es una forma de cómputo donde muchas instrucciones se ejecutan simultáneamente, con el principio de que un problema grande se puede dividir en otros más pequeños, los cuales serán resueltos simultáneamente o "en paralelo".

La principal métrica para evaluar el desempeño de los sistemas de Cómputo de Alto Rendimiento son las operaciones de punto flotante por segundo (FLOPS) que pueden procesar, actualmente se ha llegado a la escala de los petaFLOPS, habiendo sido superada previamente la escala de los teraFLOPS y próximamente será alcanzada la escala de los exaFLOPS.

Manejar tales cantidades de información requiere de equipos capaces de procesarla, transmitirla y almacenarla, para estos efectos los grandes fabricantes han desarrollado tecnologías en todas las partes que componen los equipos de cómputo destinados para estas tareas, las cuales mejoran continuamente para lograr el alto rendimiento que requieren las múltiples áreas donde se emplean.

En este contexto encontramos la tecnología de procesadores, que cada día mejoran el rendimiento y velocidad, la tecnología de redes logrando cada vez mayor ancho de banda y latencias muy bajas para poder enviar y recibir grandes cantidades de información, tal es el caso de la tecnología InfiniBand, que es utilizada ampliamente en áreas como la Inteligencia Artificial y el "Big Data", entre otras. Así como las tecnologías de almacenamiento que podemos encontrar en discos duros (HDD) y unidades de estado sólido (SSD), con mayores capacidades y velocidad, haciendo uso de los canales de comunicación existentes como SATA y SAS, aprovechando el caudal que proporciona el bus PCI a través de la tecnología NVMe, las cuales serán descritas más adelante.

4.2. Sistema de Archivos Paralelo

Un sistema de archivos paralelo es un conjunto de software diseñado para almacenar datos en varios servidores en red y para facilitar el acceso de alto rendimiento a través de operaciones de Entrada/Salida (IOPS) simultáneas y coordinadas entre clientes y nodos de almacenamiento.

Un sistema de archivos paralelo divide un conjunto de datos y los distribuye por bloques a múltiples unidades de almacenamiento, que pueden estar ubicados en servidores locales y/o remotos. No es necesario para los usuarios conocer la ubicación física de los bloques de datos para poder recuperar un archivo; dado que el sistema utiliza un espacio de nombres global que facilita el acceso a los datos. Los sistemas de archivos paralelos usan un servidor de metadatos para almacenar información sobre los datos, como el nombre del archivo, la ubicación y el propietario.

Un sistema de archivos paralelo lee y escribe datos en dispositivos de almacenamiento distribuidos utilizando múltiples rutas de E/S al mismo tiempo, como parte de uno o más procesos de un programa informático. El uso coordinado de múltiples rutas de E/S puede proporcionar un beneficio de rendimiento significativo, especialmente cuando se transmiten cargas de trabajo que involucran una gran cantidad de clientes y grandes cantidades de datos.

Dos de los sistemas de archivos paralelos más destacados son: Spectrum Scale de IBM, que fue desarrollado sobre su "*General Parallel File System*" (GPFS), y el sistema de archivos de código abierto Lustre, desarrollado como proyecto de investigación en la Universidad de Carnegie Mellon y utilizado frecuentemente en clústeres de gran escala.

Lustre es un sistema de archivos paralelo, distribuido, escalable, de alto rendimiento y alta disponibilidad, utilizado comúnmente en grandes clústeres de cómputo para soportar aplicaciones que hagan uso intensivo de datos. Su nombre deriva de las palabras "*Linux*" y "*Cluster*"; este sistema de archivos provee alto desempeño para clústeres de cómputo desde tamaño pequeño, grandes clústeres de cómputo hasta sistemas multi sitio. El software del sistema de archivos Lustre está disponible bajo la GNU/GPL.

Desde 2005 este sistema de archivos ha sido utilizado consistentemente por al menos la mitad de las 10 supercomputadoras más rápidas del mundo, así como por más de 60 supercomputadoras que se encuentran dentro de las primeras 100 más rápidas, algunas supercomputadoras que se pueden mencionar y que utilizan este sistema de archivos son, Fugaku (No. 1 - junio 2020) del Centro de Ciencias Computacionales RIKEN en Kobe, Japón; Titan (No. 4 - junio 2017) y Sequoia (No. 1 - junio 2012), dentro de la lista de las 500 supercomputadoras con mayor rendimiento del mundo.

Un sistema de archivos Lustre consiste en una cantidad de equipos de cómputo conectados entre sí y configurados en el servicio del sistema de archivos, para servir las solicitudes de acceso. El propósito primario de un sistema de archivos es permitir al usuario la lectura y escritura de datos persistentes; el sistema de archivos Lustre está diseñado para proveer esta funcionalidad, ser escalable y de alto rendimiento.

Lustre es un sistema de archivos escalable que puede ser parte de múltiples clústeres con decenas de miles de nodos cliente, decenas de petabytes de almacenamiento en cientos de servidores y soportar más del terabyte por segundo (TB/s) de rendimiento agregado en operaciones de entrada y salida (I/O). Esto convierte a Lustre como la solución elegida para grandes centros de datos corporativos o de investigación en áreas como energía, meteorología, simulación, ciencias de la vida, multimedia, finanzas, entre muchas otras.

Lustre utiliza una arquitectura distribuida en donde servidores que almacenan objetos (contenido de archivos) son accedidos por computadoras cliente mediante un protocolo de red eficiente. Existen servidores de metadatos encargados de la asignación del almacenamiento que administran el espacio de nombres del sistema de archivos, responsable de los datos contenidos.

Lustre es un sistema de archivos cliente-servidor, paralelo y distribuido, donde distintos servidores administran la presentación del almacenamiento de datos de los clientes de red, y escriben los datos enviados por los clientes en los objetivos de almacenamiento persistente.

Los servidores de metadatos guardan el espacio de nombres del sistema de archivos, proporcionan el índice al sistema de archivos y lo mantienen. Los servidores de objetos almacenan el contenido de los archivos en objetos binarios distribuidos, cada archivo individual se compone de uno o más objetos, estos objetos son distribuidos a lo largo de los objetivos de almacenamiento disponibles.

Esto permite poder generar archivos muy grandes y acceder a ellos en paralelo mediante procesos distribuidos a través de la infraestructura de red. Los clientes agregan los metadatos al espacio de nombres y los datos del objeto, para presentar un sistema de archivos POSIX coherente a las aplicaciones. Los clientes no acceden directamente al almacenamiento, todas las entradas y salidas son enviadas a través de la red. El cliente de Lustre divide las operaciones de entrada y salida en metadatos y en bloques de datos, comunicando con los servicios apropiados para atender las transacciones de entrada y salida.

Este es el concepto clave del diseño de Lustre, separar el tráfico de metadatos en pequeños bloques aleatorios y con un uso intensivo de IOPS del gran bloque de transmisión.

4.2.1. Componentes de un Sistema de Archivos Lustre

- **Servicios de Metadatos (MetaData Services).**

Se encarga de administrar la información acerca de los archivos y directorios almacenados en el sistema; está compuesto por MDS (MetaData Server, nodo servidor) y MDT (MetaData Target, discos que almacena los metadatos).

- **Servicios de Almacenamiento de Objetos (Object Storage Services).**

Se encarga de obtener y almacenar los archivos de datos en uno o más Objetivos de Almacenamiento de Objetos (OST, discos que almacena los datos) en conjunto con el Servidor de Almacenamiento de Objetos (OSS, nodo servidor).

- **Cliente.**

Es uno o más equipos de cómputo que requieren acceso al almacenamiento de datos, estos pueden ser de cualquier tipo, de cálculo, de visualización o equipos de escritorio, requieren montar el sistema de archivos en la ubicación del MDS.

4.2.2. Capacidades y límites

A continuación, se muestran las principales capacidades y límites para administración y almacenamiento de este sistema de archivos.

Tamaño Mínimo de Volumen	32 MB
Tamaño Máximo de Volumen	300 PB (producción) Más de 16 EB (teóricos)
Tamaño Máximo de Archivos	31.25 PB (LDISKFS) 512 PB (ZFS)
Número Máximo de Archivos	4 billones de archivos por MDT (LDISKFS) 256 trillones de archivos (ZFS)
Número Máximo de MDTs	Hasta 128 MDTs por sistema de archivos
Longitud Máxima Nombres de Archivos	255 bytes
Longitud Máxima Nombres de Ruta	4096 bytes (limitado por Linux VFS)

Tabla 1. Capacidades y límites del Sistema de Archivos Lustre.

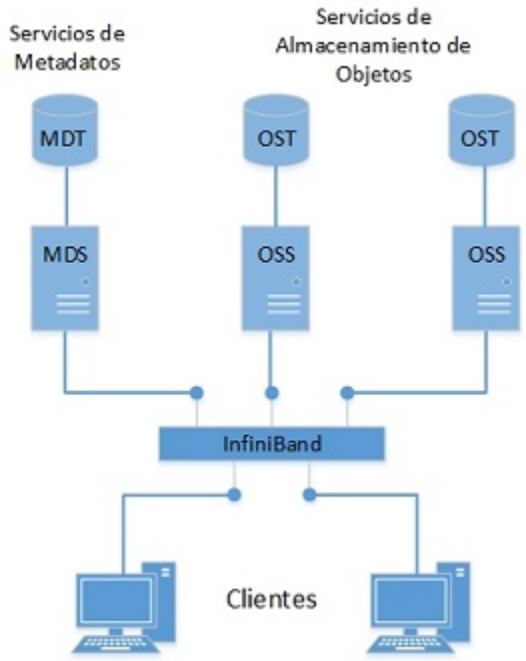

Figura 1. Diagrama de componentes de un Sistema de Archivos Lustre.

4.2.3. Redes y comunicaciones

Lustre utiliza LNET "*Lustre Networking*", que es un protocolo para configurar y comunicar a clientes y servidores a través de la red. LNET es ligero, eficiente y versátil, está diseñado para proveer máximo desempeño sobre diferentes tipos de redes, incluyendo InfiniBand, Ethernet y otros tipos de redes utilizados en clústeres de alto rendimiento tales como Cray y Omni-Path.

LNET está implementado como un módulo del *kernel* de Linux. Para soportar los distintos tipos de redes, LNET tiene un componente de bajo nivel llamado "*Lustre Network Driver*" (LND), implementado como un módulo controlador conectable.

Aunque el controlador *o2ib* LNET utiliza RDMA para las comunicaciones, TCP se usa para establecer la conexión inicial con un par usando el protocolo de nivel superior IP de la estructura. Después de la conexión inicial, el *o2ib* LND utiliza RDMA para todas las comunicaciones posteriores. Por omisión, LNET utiliza el puerto TCP 988 para crear las conexiones y este no debe ser bloqueado por ningún "*firewall*".

4.3. Tecnología InfiniBand

Es un estándar de comunicaciones para redes de cómputo de alto rendimiento que se caracteriza por una velocidad muy alta y una latencia muy baja. Se utiliza para el intercambio de datos entre computadoras; también se utiliza como interconexión directa entre servidores y sistemas de almacenamiento. Su topología de red es de estructura conmutada y diseñada para ser escalable.

La tecnología InfiniBand surge en el año 1999 como la fusión de dos tecnologías de comunicación emergentes; Future I/O desarrollada por COMPAQ, IBM y Hewlett-Packard, con Next Generation I/O, desarrollada por Intel, Microsoft y Sun Microsystems. Esta tecnología fue pensada en principio para evitar los cuellos de botella del bus PCI en las computadoras. Con esta fusión tecnológica se formaría la IBTA, "*InfiniBand Trade Association*", organización que se encarga de mantener y definir las especificaciones del estándar.

Con el protocolo InfiniBand, los datos son transmitidos en paquetes que en conjunto forman un "mensaje"; estos "mensajes" pueden ser leídos o escritos mediante operaciones de Acceso Directo a Memoria Remota (del inglés *Remote Direct Memory Access*), con lo cual el dispositivo accede directamente a la memoria del equipo remoto sin interrumpir el procesamiento del CPU.

Acceso Directo a Memoria Remota (RDMA), es un método de comunicación que permite el acceso directo a la memoria de un sistema remoto sin afectar la carga de procesamiento del sistema operativo, esto reduce la latencia e incrementa el rendimiento lo cual es muy importante principalmente en Centros de Datos y clústeres de Cómputo de Alto Rendimiento.

Una vez establecida la comunicación entre ambos equipos, la aplicación del usuario puede acceder el sistema remoto sin necesidad de hacer llamadas al Sistema Operativo en ninguno de los equipos involucrados. Con esto, se permite el acceso a la memoria del sistema y se reduce la carga en el sistema remoto; la protección de la información está asegurada por las interfaces de hardware y el protocolo de software.

InfiniBand ofrece un rendimiento de flujo de datos, de al menos 2.5 Gb/s; debido a que es escalable, admite Calidad de Servicio (QoS) y Conmutación de Errores (Fail Over), es muy utilizado para la conexión de servidores en entornos de Cómputo de Alto Rendimiento, ya que soporta direccionar hasta 64,000 dispositivos.

4.3.1. Ventajas de RDMA

- Las aplicaciones pueden realizar transferencias de datos sin la intervención de la pila de software de red.
- Los datos se transmiten y reciben directamente en búferes sin que se copien entre las capas de la red.
- Las aplicaciones pueden realizar transferencias de datos sin la intervención del kernel.
- Las aplicaciones pueden acceder a la memoria remota sin consumir tiempo de CPU.

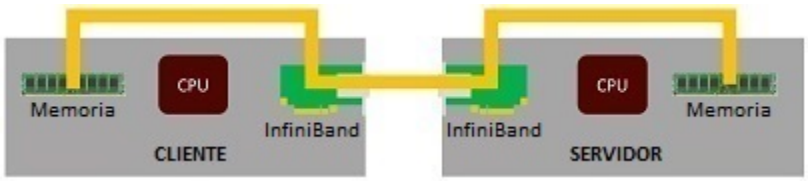

Figura 2. InfiniBand RDMA.

4.3.2. Tipos, Velocidades y Latencia

InfiniBand utiliza flujo de bits en serie para la transferencia de datos, el ancho de enlace (*link width*) depende del número de canales disponibles en un mismo cable, 1, 4, 8, o 12; actualmente se utilizan 4 canales en producción (2 para transmisión TX y 2 para recepción RX). La velocidad de enlace en InfiniBand, se ha incrementado a lo largo de los años con cada nueva generación, como se muestra a continuación.

Tipo	Ancho de Banda	Latencia
Single Data Rate (SDR)	2.5 Gb/s por canal (10 Gb/s 4 canales)	5 μs
Double Data Rate (DDR)	5 Gb/s por canal (20 Gb/s 4 canales)	2.5 μs
Quad Data Rate (QDR)	10 Gb/s por canal (40 Gb/s 4 canales)	1.3 μs
Fourteen Data Rate (FDR)	14 Gb/s por canal (56 Gb/s 4 canales)	0.7 μs
Enhanced Data Rate (EDR)	25 Gb/s por canal (100 Gb/s 4 canales)	0.5 μs
High Data Rate (HDR)	50 Gb/s por canal (200 Gb/s 4 canales)	0.13 μs
Next Data Rate (NDR)	100 Gb/s por canal (400 Gb/s 4 canales)	TBA

Tabla 2. Tipos, velocidades y latencia de la familia InfiniBand.

4.4. Tecnología NVMe

NVMe es un estándar de comunicaciones desarrollado especialmente para Unidades de Estado Sólido (SSD) en todos sus factores de forma (U.2, M.2, AIC, EDSFF) y que define la manera en que el software del equipo anfitrión se comunica con la memoria No Volátil a través del bus PCI Exprés mediante un conjunto de comandos y un conjunto de funciones; es una interfaz mucho más eficiente que provee latencia más baja y es más escalable que las interfaces heredadas como Serial ATA (SATA).

En la especificación, el equipo anfitrión controla la interfaz, donde la arquitectura NVMe ofrece un nuevo mecanismo de alto rendimiento que soporta hasta 65,535 colas para operaciones de entrada y salida (I/O) cada una con hasta 65,535 comandos. Las colas se asignan a los núcleos del procesador, ofreciendo un rendimiento escalable; reduce significativamente la cantidad de comandos asignados en memoria adaptándose a los controladores de dispositivos del sistema operativo, para un mayor rendimiento y una menor latencia.

Linux soporta arreglos RAID de dispositivos NVMe y tiene mejoras para incrementar la confiabilidad, el rendimiento y la escalabilidad, existen soluciones de código abierto (Open Source) para la administración de arreglos de unidades de almacenamiento compatibles con el estándar RAID como *mdadm*, existen otras herramientas para la administración de las unidades NVMe que permiten verificar el estado de los dispositivos, temperatura, resistencia, hacer actualizaciones de firmware, borrar las unidades de forma segura, leer los comandos SMART, entre otras funciones, la herramienta de código abierto en Linux es *nvme-cli*.

La tecnología NVMe fue diseñada para unidades SSD con tecnología flash, los comandos utilizan ciclos de CPU bajos, presenta una latencia de 2.8 microsegundos, se comunica directamente con el CPU, elimina intermediarios al comunicarse directamente con el CPU del sistema, puede generar más de 1 millón de operaciones de entrada y salida (IOPS) y transferir datos a 32,000 MB/s en puertos PCIe Gen. 4 mediante el uso de 16 canales.

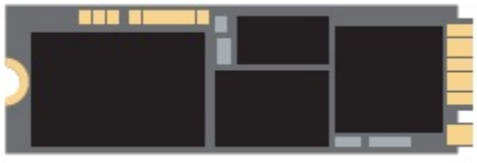

Figura 3. Ilustración de Unidad NVMe con Factor de Forma M.2

4.5. Puntos de Referencia (Benchmarks)

Cuando hablamos de Cómputo de Alto Rendimiento y el desempeño general de los sistemas de cómputo, una parte muy importante de éstos son los medios y sistemas de almacenamiento. El rendimiento que estos ofrecen en cualquier ambiente o entorno de trabajo puede impactar en el óptimo funcionamiento del sistema o aplicación; por lo que realizar pruebas de rendimiento en el sistema de almacenamiento nos podrá indicar qué tan eficiente es para realizar una tarea determinada como, por ejemplo, una búsqueda de información dentro de una base de datos.

Los fabricantes de dispositivos de almacenamiento ofrecen inmensas cantidades de datos de rendimiento sobre sus productos, las cuales no necesariamente reflejan cómo funcionarán dichos dispositivos cuando se utilicen en un ambiente de trabajo específico. Es por esta razón que diversos desarrolladores han puesto a disposición de los usuarios, herramientas para evaluar el rendimiento de los dispositivos de almacenamiento.

Dentro de los ambientes Linux podemos encontrar diversas herramientas que nos podrán ayudar a realizar pruebas de Entrada y Salida (I/O) para evaluar el rendimiento de los discos (HDD / SSD / NVMe); entre las cuales encontramos, Flexible I/O (FIO), IOzone, Bonnie++, entre otros.

Cabe mencionar que las distintas herramientas o utilerías disponibles, utilizan una carga simulada para imitar el tipo de operaciones realizadas, por lo que no hay garantía de que las aplicaciones o los mismos dispositivos de almacenamiento puedan alcanzar el nivel de rendimiento que los fabricantes puedan especificar.

Para los fines de estas pruebas se decidió emplear tres herramientas distintas; dd, IOzone y FIO; a continuación, se describen las principales herramientas y utilerías encontradas o recomendadas para realizar las pruebas de rendimiento.

4.5.1. Bonnie++

Es un programa para probar el rendimiento de discos duros (unidades de almacenamiento) y sistemas de archivos. Existen diferentes tipos de operaciones en los sistemas de archivos que las que distintas aplicaciones utilizan en diferentes grados. Esta herramienta prueba algunos de ellos y para cada prueba da un resultado de la cantidad de trabajo realizado, el porcentaje de carga del CPU, entre otros. Para los resultados de rendimiento los números más altos son los mejores, para el uso del CPU los más bajos son mejores.

En la salida de este comando se muestran dos secciones, la primera muestra el rendimiento de E/S en una forma diseñada para simular algunos tipos de aplicaciones. La segunda sección muestra la creación, lectura y eliminación de archivos pequeños, de forma similar a los patrones de algunas aplicaciones.

4.5.2. dd

Es una utilería en línea de comandos para sistemas operativos UNIX y Linux, cuyo propósito principal es convertir y copiar archivos de datos a bajo nivel, puede ser utilizado para transferir datos específicos, hacer copias de seguridad "en crudo" (*raw data*) y convertir algunas codificaciones soportadas de caracteres predefinidos como ASCII y EBCDIC; sus siglas significan "*Dataset Definition*".

El comando *dd* lee un bloque de entrada, lo procesa y lo escribe en el archivo de salida indicado. Se puede especificar el tamaño del bloque de entrada y salida a utilizar y así obtener una cantidad de datos fija, de esta forma procede a la lectura bloque por bloque del origen y su respectiva escritura en la salida, con esto el tamaño del bloque escrito es idéntico al del bloque leído.

Este comando puede acceder a los dispositivos de hardware como unidades de disco duro y archivos de dispositivos especiales como */dev/null*, */dev/random* y */dev/zero*, ya que en el sistema operativo aparecen como archivos normales; puede leer y escribir en estos archivos siempre y cuando la función esté implementada en los respectivos módulos (controladores).

De forma predeterminada lee desde la entrada estándar (*stdin*) y escribe en la salida estándar (*stdout*), estos parámetros se pueden cambiar utilizando las opciones "*if*" (archivo de entrada por sus siglas en inglés, *input file*) y "*of*" (archivo de salida por sus siglas en inglés, *output file*).

4.5.3. FIO

Es una herramienta de código abierto (Open Source) versátil, capaz de generar cargas de trabajo de entrada y salida (I/O) para medir ciertos parámetros en los sistemas de almacenamiento de sistemas de cómputo mediante cambios en el subsistema de entrada y salida de Linux. Permite configuraciones detalladas de la carga de trabajo y muestra los informes necesarios al finalizar cada prueba.

Pruebas primarias de estado que realiza:

- Rendimiento (lectura y escritura agregada de IOPS)
- Latencia promedio (latencia de lectura y escritura promediada en conjunto)
- Latencia máxima (latencia máxima de lectura o escritura)
- Desviación estándar de latencia (desviación estándar de lectura y escritura promediada en conjunto)

Sus siglas significan "*Flexible IO*" y es una herramienta ampliamente utilizada como referencia estándar de la industria, como prueba de estrés y verificación de entrada y salida.

4.5.4. IOzone

Es una herramienta de código abierto (Open Source) para la evaluación de sistemas de archivos, presenta la ventaja de estar disponible para una amplia variedad de plataformas como Linux, AIX, Solaris, BSD, MacOSX y Windows (mediante Cygwin). IOzone resulta bastante útil ya que genera y mide una gran variedad de operaciones en los sistemas de archivos, midiendo el rendimiento de E/S del archivo de pruebas, básicamente ejecuta una serie de pruebas en un archivo, que se divide en porciones más pequeñas. Entre las pruebas que puede realizar encontramos:

- Lectura
- Escritura
- Lectura Aleatoria
- Escritura Aleatoria
- Re-lectura
- Re-escritura
- Lectura Inversa

5. Metodología de Investigación

Para la realización del presente proyecto, el Tutor responsable de administrar el clúster de Cómputo de Alto Rendimiento del Instituto de Ciencias Nucleares facilitó los equipos para la instalación y configuración del sistema de archivos paralelo, se utilizó un switch Ethernet para la interconexión entre los equipos y su administración, así como para proporcionar conectividad a la red Internet, se utilizó un switch InfiniBand FDR para el intercambio de datos entre los servidores que componen el Sistema de Archivos Paralelo Lustre y el equipo cliente.

Para obtener el máximo rendimiento en las pruebas se eligió utilizar arreglos RAID 0 por software, este tipo de arreglos permite agrupar unidades de almacenamiento (HDD / SSD / NVMe) añadiendo la capacidad de cada unidad simultáneamente para un almacenamiento óptimo de datos, es el modo que ofrece mayor rendimiento, ya que los datos se leen y se escriben en todas las unidades de la matriz en paralelo, los datos se distribuyen sucesivamente en varias unidades de almacenamiento para acelerar su procesamiento. Sin embargo, carece de una función muy importante; la protección de los datos almacenados, ya que si se produce algún error en alguna de las unidades se dejará de tener acceso a los datos.

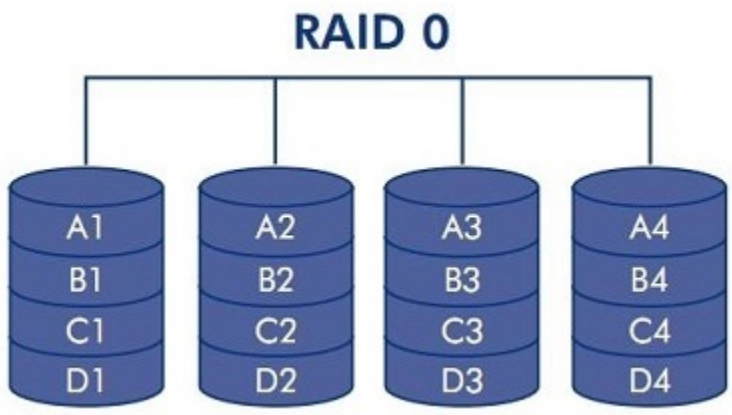

Figura 4. Ilustración de un arreglo RAID 0 con 4 unidades de almacenamiento.

El servidor MDS-MGS cuenta con cinco Discos Duros (HDD) con capacidad de 3 TB cada uno, conectados por puertos SATA-3 y dos Discos Duros más, conectados a través de una tarjeta RAID SAS-2. Para el Servicio de Metadatos, se utilizó un Disco Duro SATA para una capacidad de 3 TB en el MDT-MGT.

El servidor OSS cuenta con cuatro unidades de almacenamiento NVMe de 500 GB cada una, se utilizaron arreglos RAID 0 por software con las cuatro unidades y con ZFS, para tener una capacidad de hasta 2 TB para el Servicio de Almacenamiento de Objetos en el OST.

5.1. Descripción de pruebas

Se ejecutaron distintas pruebas, dependiendo de los parámetros y forma de diagnóstico de cada herramienta, de la siguiente manera:

- **Dispositivos**: Unidades NVMe independientes y arreglo de dos unidades (NVMe1-3).
- **Número de procesos**: Se utilizaron 1, 2, 4 y 8 procesos, para simular la misma cantidad de usuarios simultáneos accediendo a los dispositivos de almacenamiento NVMe.
- **Sistema de archivos**: Se probaron las unidades NVMe por separado y un arreglo de dos unidades, "en crudo" y con ZFS.
- **Tamaño de archivos**: Se utilizaron varios tamaños de archivos, desde 512 MB, 1024 MB, 2048 MB, 4096 MB y hasta 8192 MB, para simular una variedad de cargas de trabajo típicas.
- **Tamaño de bloque**: En pruebas con dd y FIO se utilizó el tamaño del bloque en 512KB, en pruebas automáticas con IOzone se utilizaron bloques desde 4KB hasta 16 MB.
- **Tamaño de la muestra**: Se realizaron un total de 5 repeticiones de cada prueba, excepto para las pruebas automáticas con IOzone donde se ejecutó una sola prueba.

5.2. Descripción de equipos

A continuación, se describen las características generales, nombre de red y función de los equipos que componen el Sistema de Archivos Paralelo Lustre instalado, para la realización de las pruebas de rendimiento, objetivo de este trabajo y de acuerdo con lo descrito en el apartado ***Componentes de un Sistema de Archivos Lustre.***

Nombre del Equipo	ICN01	MGS + MDS + MDT
Sistema Operativo	Linux CentOS 7.8	
Marca	ASUS	
Modelo	RS920A-E6/RS8	
Procesador	AMD Opteron Processor 6212 @ 3.2 GHz (x4 CPU)	
Memoria RAM	64GB totales (16 GB por procesador) DIMM DDR3 Synchronous Unbuffered 800 MHz	
Almacenamiento	Samsung SSD 840 (120 GB) Seagate Barracuda ST3000DM001-1CH1 (3 TB x7) SB7x0/SB8x0/SB9x0 SATA Controller (x5 HDD) ASUS Pike 2108 8-port SAS2 6G RAID Card (x2 HDD)	
Adaptadores de Red	82580 Ethernet Gigabit Network (x4) MT25208 InfiniHost III Ex 10Gbit/s	

Tabla 3. Características generales del nodo ICN-01 (Lustre MGS+MDS+MDT).

Nombre del Equipo	ICN04	CLIENTE
Sistema Operativo	Linux CentOS 7.8	
Marca	DELL	
Modelo	PowerEdge R815	
Procesador	AMD Opteron Processor 6212 @ 3.2 GHz (x4 CPU)	
Memoria RAM	64GB totales (16 GB por procesador) DIMM DDR3 Synchronous Registered 1333 MHz	
Almacenamiento	Crucial M4-CT064M4SSD2 (64 GB) SB7x0/SB8x0/SB9x0 SATA Controller SAS2008 PCI-Express Fusion-MPT SAS-2 (x1 SSD)	
Adaptadores de Red	NetXtreme II BCM5709 Gigabit Ethernet (x4) MT25408A0-FCC-QI ConnectX Dual Port 40Gb/s InfiniBand	

Tabla 4. Características generales del nodo ICN-04 (Lustre Cliente).

Nombre del Equipo	ICN05	OSS + OST
Sistema Operativo	Linux CentOS 7.8	
Marca	SuperMicro	
Modelo	SuperServer SYS-6029P-WTR	
Procesador	Intel Xeon Gold 6132 CPU @ 2.60GHz (x2 CPU)	
Memoria RAM	128 GB (64 GB por procesador) DIMM DDR4 Synchronous 2666 MHz	
Almacenamiento	Crucial M4-CT064M4SSD2 C620 Series Chipset Family SATA Controller (x1 SSD) Non-Volatile Memory controller (NVMe) Kingston Technology SKC2000M8/500G (x4)	
Adaptadores de Red	NetXtremeEthernet Connection X722 for 1GbE (x2) MT27500 Family [ConnectX-3] 40Gbit/s InfiniBand	

Tabla 5. Características generales del nodo ICN-05 (Lustre OSS+OST).

5.3. Diagrama de red

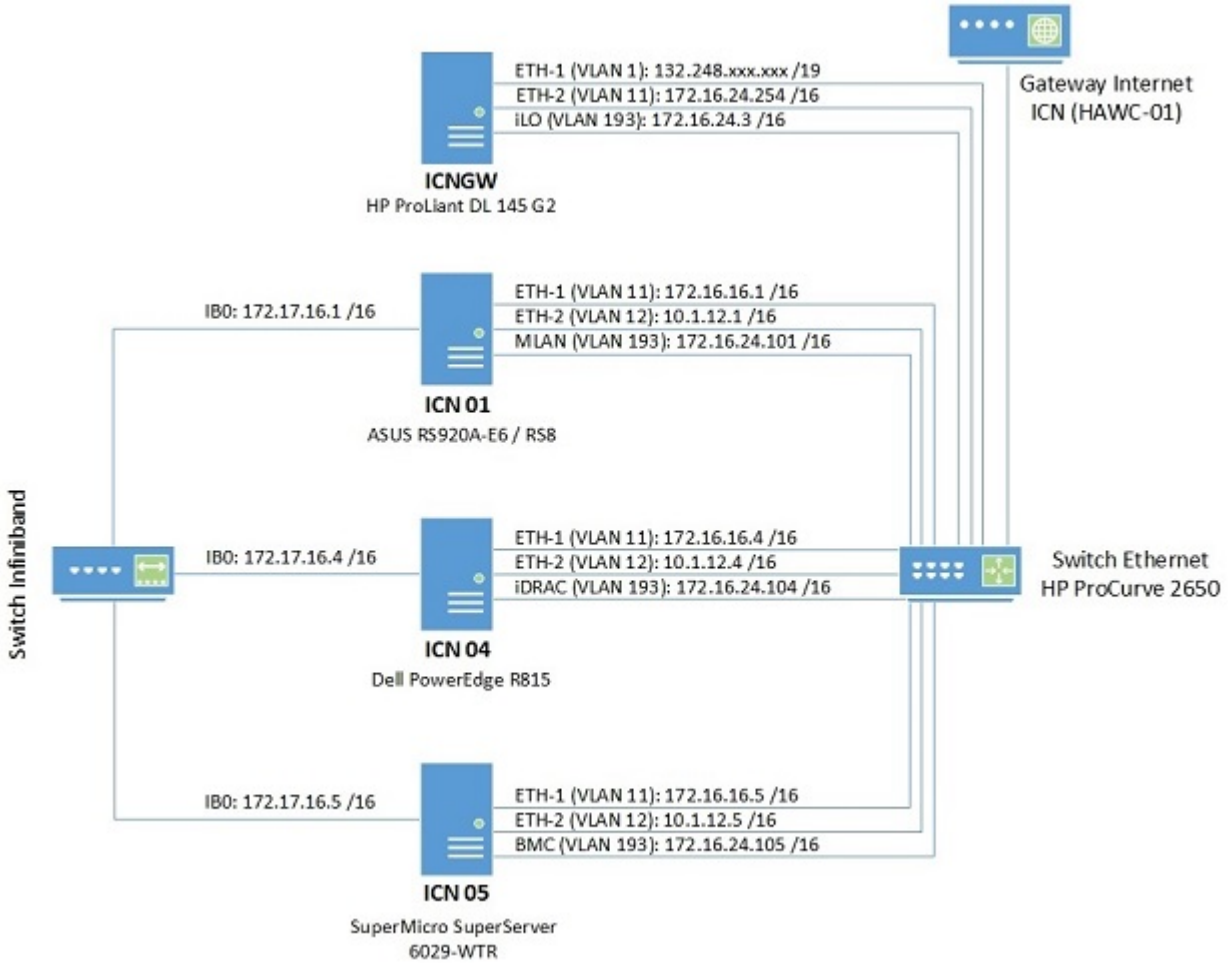

Figura 5. Diagrama de conexiones Ethernet e InfiniBand y direccionamiento IPv4.

ICN01 (Almacenamiento HDD)	Servidor MGS + MDS Lustre FS
ICN04 (Cliente)	Cliente Lustre FS
ICN05 (Almacenamiento NVMe)	Servidor OSS Lustre FS
ICNGW (Servicios)	DHCP, DNS, NAT, NTP, Acceso Remoto

Tabla 6. Lista nodos de red, nombres y funciones.

5.4. Configuración Switch Ethernet

Para la administración y configuración de los equipos incorporados en red que son parte del sistema de archivos paralelo Lustre, se realizó la conexión a través de un switch Ethernet administrable, el cual tiene la siguiente configuración de puertos, VLAN y direccionamiento IPv4.

- **public**: VLAN para el acceso a la red Internet, descarga de paquetes de software y administración remota a través del nodo de servicio.
- **local**: VLAN para el acceso principal de administración y configuración de los equipos en la red local.
- **backup**: VLAN para el acceso secundario de administración y configuración de los equipos en la red local.
- **ipmi**: VLAN para el acceso a los puertos de consola de administración (BMC, iLO, iDRAC, Management LAN) de los equipos en la red local.

A continuación se muestran detalladamente los principales parámetros configurados en el switch Ethernet, para el manejo de la red local.

ID VLAN	Nombre VLAN	Puertos [U]	Puertos [T]	Dirección IP
VLAN 1	public	[1 - 8]	- - -	- - -
VLAN 11	local	[9 - 32, 49]	[41 - 48]	172.16.10.1 /16
VLAN 12	backup	[33 - 40, 50]	[9]	10.1.12.254 /16
VLAN 193	ipmi	[41 - 48]	- - -	- - -

Tabla 7. Configuración de puertos y redes VLAN en Switch Ethernet.

5.5. Configuración Servidores Lustre (MDS / OSS)

Para la configuración de los servidores MGS+MDS y OSS del Sistema de Archivos Paralelo Lustre, se realizó lo siguiente:

- Instalar el sistema operativo CentOS 7 en modo *Servidor de Infraestructura* y seleccionar el *Soporte para InfiniBand*.
- Configurar los adaptadores de red Ethernet e InfiniBand en cada uno de los equipos, siguiendo el diseño establecido.
- Editar el archivo */etc/hosts* en cada uno de los equipos e incluir los nodos que comprenden el sistema de archivos.
- Deshabilitar SELinux.
- Deshabilitar el *firewall* del sistema.
- Actualizar el sistema operativo mediante *yum update*.
- Reiniciar el nodo.
- Instalar las herramientas de desarrollo *'Development Tools'*.
- Instalar el repositorio *epel-release*.
- Instalar las dependencias adicionales.
- Comprobar la versión instalada de CentOS.
- Instalar el repositorio de ZFS para la versión instalada de CentOS.
- Deshabilitar DKMS y habilitar kABI en el archivo de configuración del repositorio de ZFS.
- Instalar ZFS y sus paquetes asociados.
- Reiniciar el nodo.
- Descargar el código fuente de Lustre.

- Configurar para compilar el software para servidores con soporte ZFS.
- Crear e instalar los paquetes RPM.
- Reiniciar el nodo.
- Generar el identificador del anfitrión (*hostid*) persistente en la máquina, en caso de que no exista uno.
- Editar el archivo */etc/modprobe.d/lnet.conf* agregar las opciones del adaptador de red a utilizar en el sistema de archivos paralelo.
- Reiniciar el nodo.
- Cargar los módulos ZFS y Lustre.
- Crear las unidades destino MDT y OST en cada uno de los servidores MDS y OSS, respectivamente.
- Editar el archivo */etc/ldev.conf* en cada uno de los servidores y agregar la unidad creada.
- Iniciar el servicio Lustre.

Para instrucciones detalladas consultar las referencias [7] y [8]

5.6. Configuración Cliente Lustre

Para la configuración del Cliente del Sistema de Archivos Paralelo Lustre, se realizó lo siguiente:

- Instalar el sistema operativo CentOS 7 en modo *Nodo de Cómputo* y seleccionar el *Soporte para InfiniBand*, en el equipo cliente.
- Configurar los adaptadores de red Ethernet e InfiniBand en cada uno de los equipos, siguiendo el diseño establecido.
- Editar el archivo */etc/hosts* e incluir los nodos que comprenden el sistema de archivos.
- Deshabilitar SELinux.
- Deshabilitar el *firewall* del sistema.
- Actualizar el sistema operativo mediante *yum update*.
- Reiniciar el nodo.
- Instalar las herramientas de desarrollo *'Development Tools'*.
- Instalar el repositorio *epel-release*.
- Instalar las dependencias adicionales.
- Descargar el código fuente de Lustre.
- Configurar para compilar el software en modo Cliente.
- Crear e instalar los paquetes RPM.
- Reiniciar el nodo.
- Editar el archivo */etc/modprobe.d/lnet.conf* y agregar las opciones del adaptador de red a utilizar en el sistema de archivos paralelo.
- Reiniciar el nodo.
- Crear el directorio para montaje de la unidad Lustre.

- Montar la unidad Lustre en el directorio creado.
- Editar el archivo */etc/fstab* y agregar la entrada de la unidad Lustre.
- Comprobar el espacio disponible de almacenamiento.

Para instrucciones detalladas consultar las referencias [7] y [8]

5.7. Ejecución de pruebas

dd: Se ejecutaron una serie de pruebas iniciales aprovechando la habilidad de esta utilería para manejar datos en bajo nivel (*raw data*) y acceder a los dispositivos de la misma forma, de esta manera se generaron transferencias de archivos tanto de entrada como de salida para simular la carga de trabajo en operaciones de lectura y escritura.

Esto con el objetivo de medir la velocidad de transferencia (Ancho de banda en MB/s) del dispositivo "en crudo" (*raw device*), es decir directamente al dispositivo de bloques sin punto de montaje ni sistema de archivos. Posteriormente se ejecutaron el mismo tipo de pruebas utilizando los dispositivos con el sistema de archivos ZFS, tanto para las unidades por separado como para un arreglo de dos unidades (NVMe1-3) y comparar el comportamiento.

FIO: Se ejecutaron conjuntos de pruebas para las principales formas de entrada y salida; es decir, Lectura y Escritura Secuenciales, para tamaños de archivo de 512 MB, 1 GB, 2 GB, 4 GB y 8 GB, con las unidades por separado y con un arreglo de dos unidades NVMe compartidas entre los dos procesadores del equipo (CPU-0 y CPU-1), midiendo la velocidad de transferencia, las operaciones de entrada y salida por segundo (IOPS), así como el tiempo de ejecución; con un tamaño de bloque fijo de 512 KB y con 1, 2, 4 y 8 procesos, para simular la carga de trabajo con el mismo número de usuarios simultáneos; tanto "en crudo" como con el sistema de archivos ZFS utilizando un arreglo de dos unidades (NVMe1-3).

IOzone: Se ejecutaron pruebas unitarias con un arreglo de dos unidades (NVMe1-3), con lo que se mide la velocidad de transferencia (MB/s) y las Operaciones de E/S, con tamaños de archivos de 512 MB, 1 GB, 2 GB, 4 GB y 8 GB, utilizando distintos tamaños de bloque para 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192 y 16384 KB, las pruebas incluyen las formas de entrada y salida para Lectura y Escritura Secuenciales, principalmente y comparar los resultados con los obtenidos con dd y FIO; quedando las pruebas con esta herramienta solamente como referencia.

Inicialmente se realizaron pruebas con el sistema de archivos Lustre desde el cliente de red, utilizando la red InfiniBand como medio de transporte de la información entre los nodos del sistema para aprovechar las capacidades antes descritas, las pruebas se realizaron con los programas IOzone y FIO, obteniéndose resultados dudosos, por lo que se decidió probar localmente las unidades y observar el comportamiento de su rendimiento, realizando pruebas básicas simuladas con las herramientas de software antes mencionadas.

Se realizaron pruebas con las unidades por separado de la forma descrita en el apartado anterior, utilizando la herramienta dd para comprobar su rendimiento "en crudo", sin agregar alguna capa de software, como un sistema de archivos (ZFS), un arreglo RAID-0 por software (*mdadm*) o todo lo que involucra el funcionamiento de Lustre; se encontró que la unidad NVMe0 presenta un rendimiento inferior en todas las pruebas, estos resultados se pueden consultar en el Apéndice A; por lo que se procedió a revisar los parámetros de operación en el sistema operativo (Linux CentOS 7) de los dispositivos y puertos utilizados dentro de la placa principal (*Motherboard*), se obtiene lo siguiente:

Consulta de los dispositivos NVMe conectados al bus PCI del sistema.

pci:0 pci@0000:01:00.0 Non-Volatile memory controller nvme0n1 irq:16 memory:9d100000-9d103fff

pci:5 pci@0000:5f:00.0 Non-Volatile memory controller nvme1n1 irq:44 memory:c5e00000-c5e03fff

pci:9 pci@0000:b0:00.0 Non-Volatile memory controller nvme3n1 irq:48 memory:ee500000-ee503fff

pci:10 pci@0000:d8:00.0 Non-Volatile memory controller nvme4n1 irq:50 memory:fbe00000-fbe03fff

Con esta información podemos notar que la unidad NVMe0 se encuentra conectada a través del puerto PCI0 de la placa principal, ocupando el puerto M.2 soportado por el PCH ("*Platform Controller Hub*"), como se puede apreciar en el siguiente diagrama de bloques del sistema.

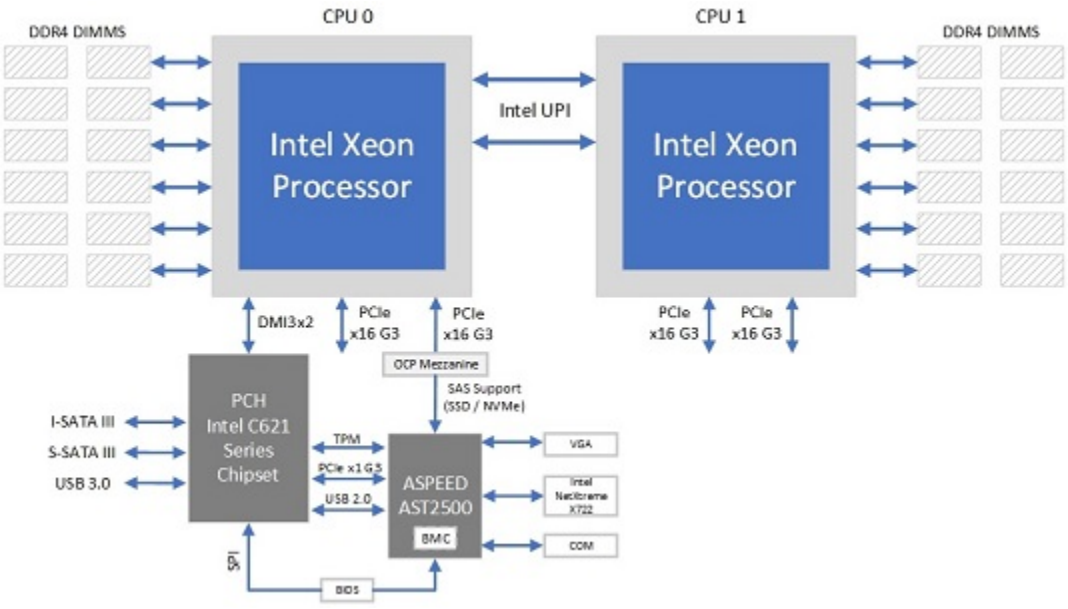

Figura 6. Diagrama de Bloques para una configuración típica de 2 procesadores.

Obteniendo más información sobre los parámetros de operación de los dispositivos NVMe0 y NVMe1 se encuentran las siguientes diferencias:

Para la unidad NVMe0.

01:00.0 Non-Volatile memory controller: Kingston Technology Company, Inc. Device 2262 (rev 03)

DevCtl2: Completion Timeout: 50us to 50ms, TimeoutDis-, **LTR+**, OBFF Disabled - - -

Max snoop latency: **71680ns**

Max no snoop latency: **71680ns**

Para la unidad NVMe1.

5f:00.0 Non-Volatile memory controller: Kingston Technology Company, Inc. Device 2262 (rev 03)

DevCtl2: Completion Timeout: 50us to 50ms, TimeoutDis-, **LTR-**, OBFF Disabled 32c32

Max snoop latency: **0ns**

Max no snoop latency: **0ns**

En este caso, el parámetro **LTR** (*Latency Tolerance Reporting*) se encuentra habilitado (LTR+) en la unidad NVMe0 y para la unidad NVMe1 el mismo parámetro se encuentra deshabilitado (LTR-), por lo que la unidad NVMe0 reporta una tolerancia de latencia hasta 71680ns, por su parte, la unidad NVMe1 reporta una latencia de 0ns, lo que retrasa el envío y recepción de información desde y hacia este dispositivo (NVMe0).

LTR es parte de la especificación del bus PCIe 3.0 y su propósito es mejorar la reserva de tiempo y recursos basado en los requerimientos de desempeño de un punto terminal; esto sirve en general para que el sistema optimice el acceso a los registros de memoria de los periféricos, de tal forma que, si un dispositivo es lento en su acceso, sean atendidas en forma prioritaria, todas aquellas que son rápidas.

Esto es debido a que el puerto PCI0 donde se encuentra conectada la unidad NVMe0, es administrada directamente por el "Concentrador del Controlador de Plataforma" (*Platform Controller Hub*) del conjunto de chips (*Chipset*), dado que la unidad se encuentra instalada en el puerto M.2 de la placa principal. Por lo anterior, la unidad NVMe0 presenta un menor rendimiento con respecto del resto de los dispositivos de almacenamiento NVMe disponibles, también podemos ver que las unidades NVMe1, NVMe3 y NVMe4, se encuentran conectadas directamente a los CPU-0 y CPU-1 mediante los puertos de expansión PCIe x16 G3 ilustrados en el diagrama de bloques del sistema (Figura 5).

6. Resultados

A continuación se muestran los resultados obtenidos para las Unidades NVMe "en crudo" con dd para uno y dos procesos en paralelo.

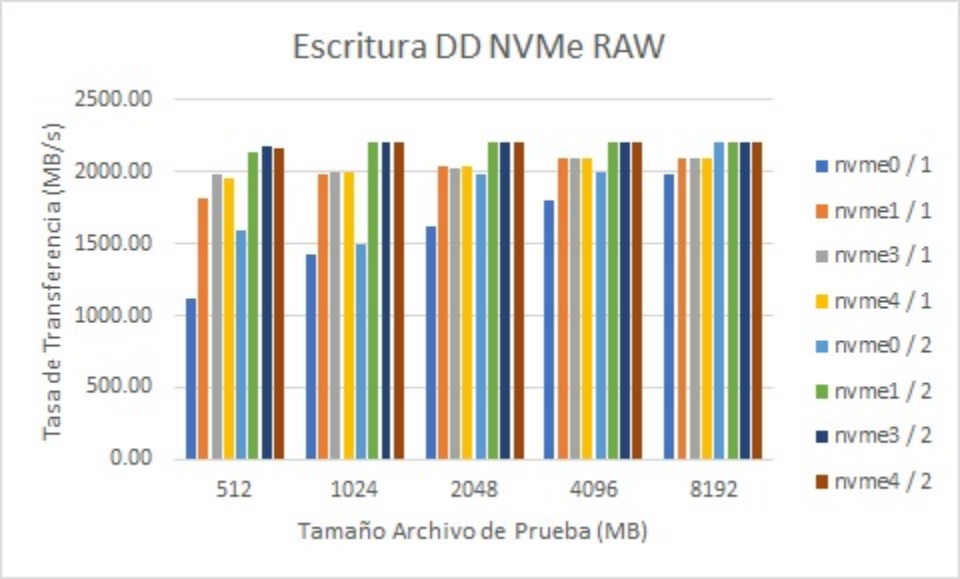

Figura 7. Prueba Rendimiento Escritura dd: Unidades NVMe "en crudo" para 1 y 2 procesos.
*dd if=/dev/zero of=/dev/nvme**X**n1 bs=512K count=1000 oflag=direct*

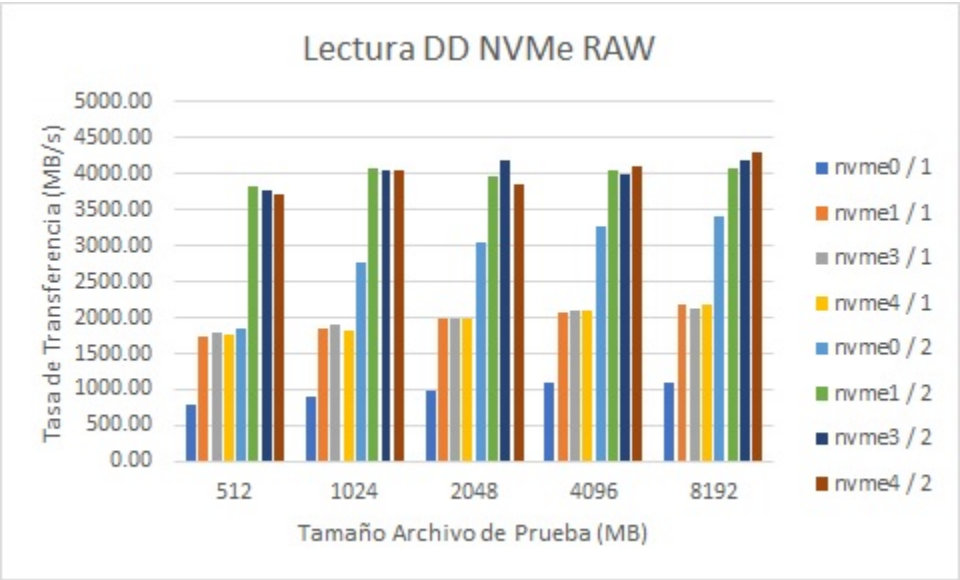

Figura 8. Prueba Rendimiento Lectura dd: Unidades NVMe "en crudo" para 1 y 2 procesos.
*dd if=/dev/nvme**X**n1 of=/dev/null bs=512K count=1000*

Las siguientes gráficas muestran el comportamiento del rendimiento de un arreglo RAID0 y un dispositivo VDEV de ZFS con dd para 1, 2, 4 y 8 procesos.

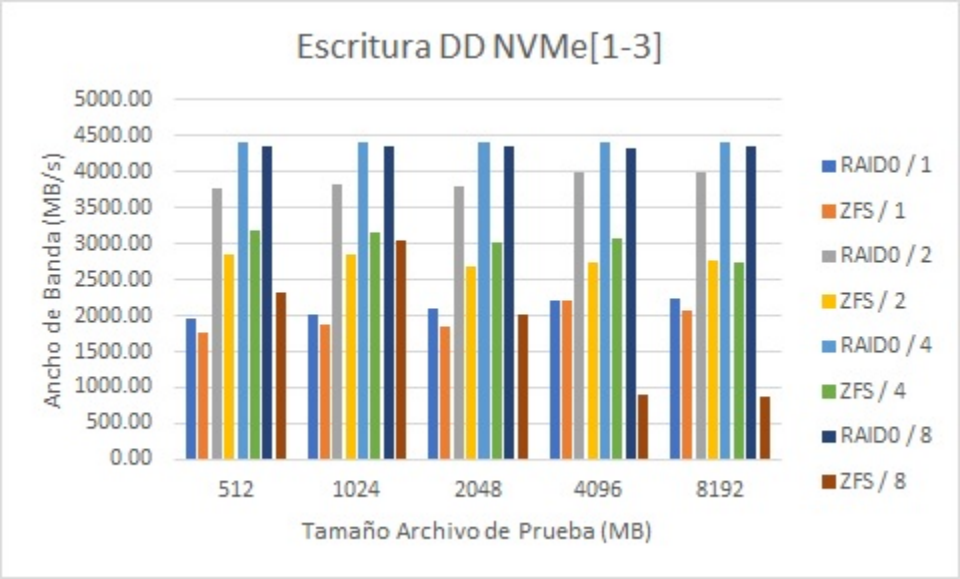

Figura 9. Comparativa Rendimiento Escritura dd: RAID0 vs ZFS para 1, 2, 4 y 8 procesos.

dd if=/dev/zero of=/dev/md13 bs=512K count=1000 oflag=direct

dd if=/dev/zero of=/01.dd bs=512K count=1000 oflag=direct

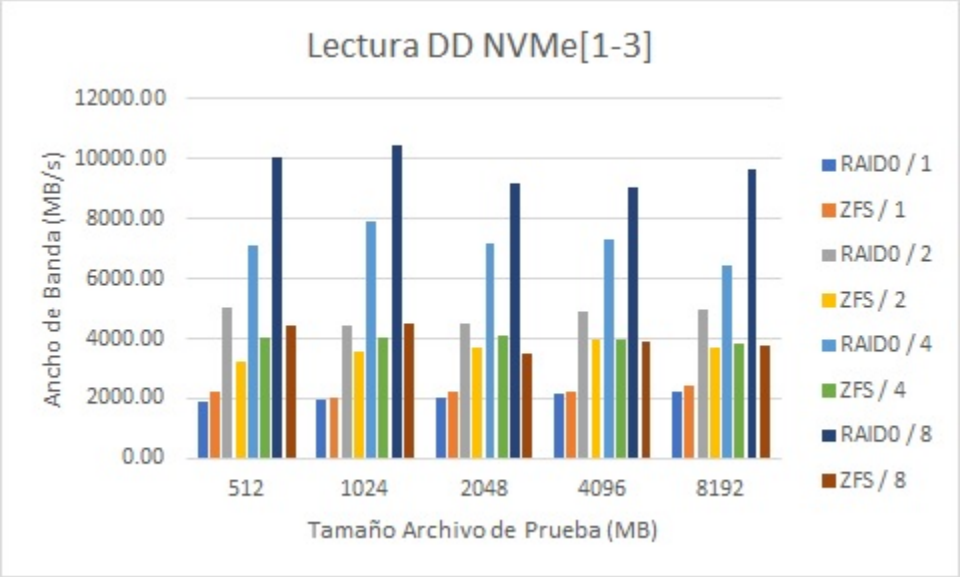

Figura 10. Comparativa Rendimiento Lectura dd: RAID0 vs ZFS para 1, 2, 4 y 8 procesos.

dd if=/dev/md13 of=/dev/null bs=512K count=1000

dd if=/01.dd of=/dev/null bs=512K count=1000

Las siguientes gráficas muestran el comportamiento del rendimiento de un arreglo RAID0 y un dispositivo VDEV de ZFS con FIO para 1, 2, 4 y 8 procesos.

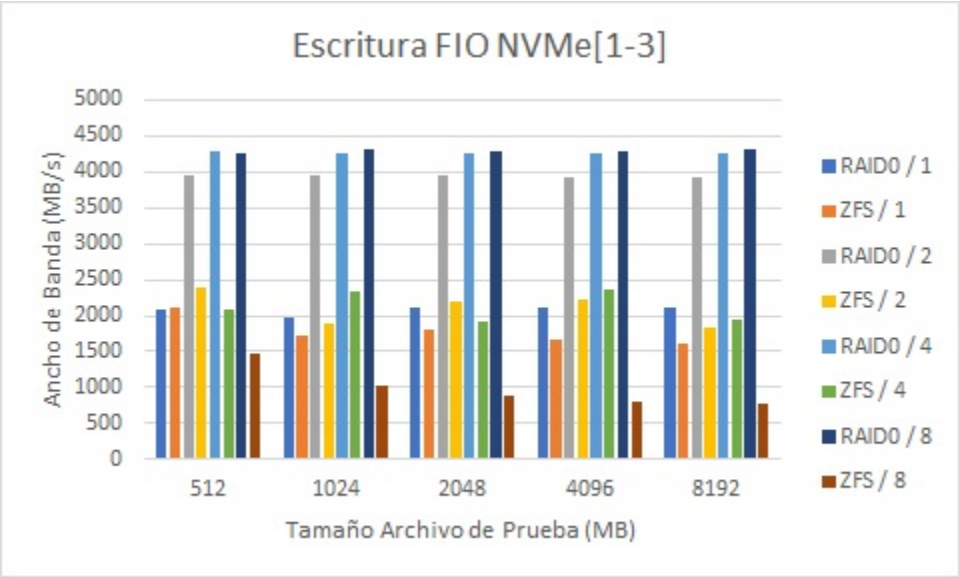

Figura 11. Comparativa Rendimiento Escritura FIO: RAID0 vs ZFS para 1, 2, 4 y 8 procesos.
fio --name=seqwrite --rw=write --direct=1 --ioengine=libaio --bs=512K --zero_buffers=1 --size=512M | 1024M | 2048M | 4096M | 8192M --numjobs=1 | 2 | 4 | 8
RAID0: *--filename=/dev/md13* ZFS: *--directory=/zpool13*

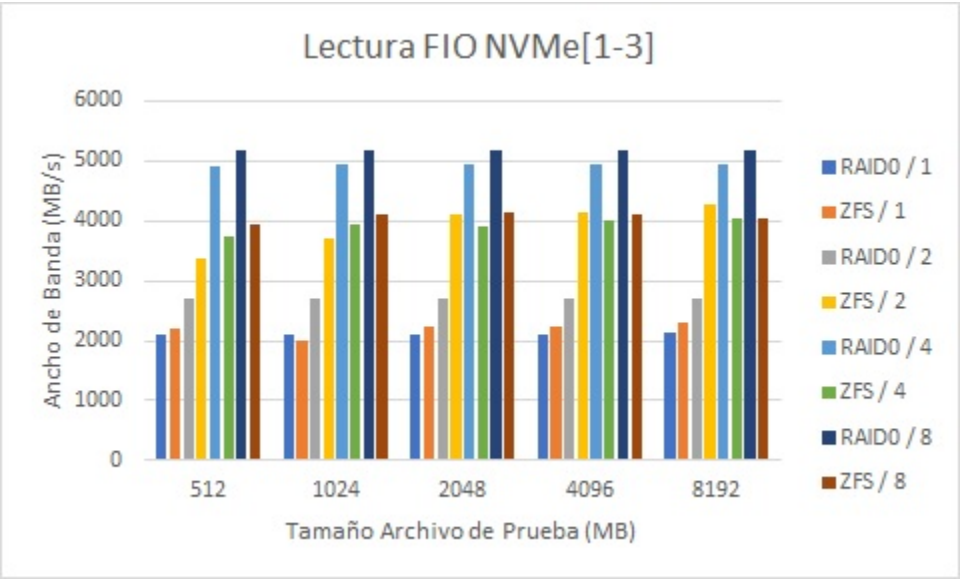

Figura 12. Comparativa Rendimiento Lectura FIO: RAID0 vs ZFS para 1, 2, 4 y 8 procesos.
fio --name=seqread --rw=read --direct=1 --ioengine=libaio --bs=512K --zero_buffers=1 --size=512M | 1024M | 2048M | 4096M | 8192M --numjobs=1 | 2 | 4 | 8
RAID0: *--filename=/dev/md13* ZFS: *--directory=/zpool13*

Operaciones de Entrada/Salida por Segundo generadas con un arreglo RAID0 y un dispositivo VDEV de ZFS con FIO para 1, 2, 4 y 8 procesos.

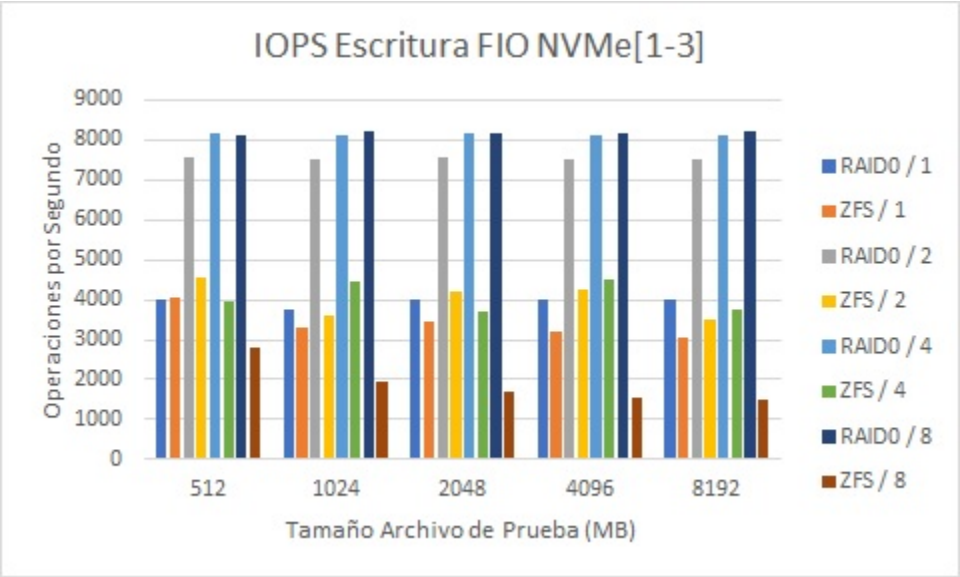

Figura 13. Comparativa IOPS Escritura FIO: RAID0 vs ZFS para 1, 2, 4 y 8 procesos.
fio --name=seqwrite --rw=write --direct=1 --ioengine=libaio --bs=512K --zero_buffers=1 --size=512M | 1024M | 2048M | 4096M | 8192M --numjobs=1 | 2 | 4 | 8
RAID0: *--filename=/dev/md13* ZFS: *--directory=/zpool13*

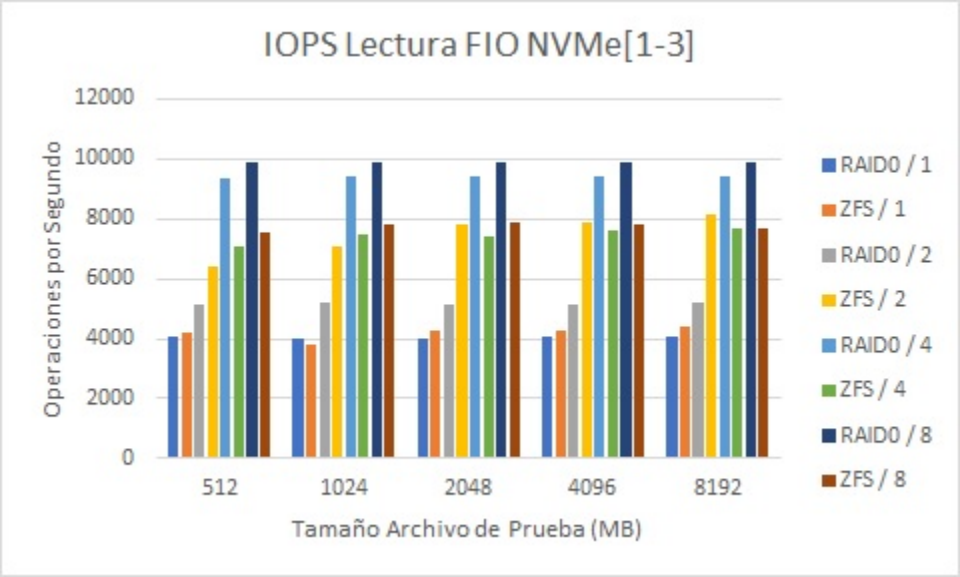

Figura 14. Comparativa IOPS Lectura FIO: RAID0 vs ZFS para 1, 2, 4 y 8 procesos.
fio --name=seqread --rw=read --direct=1 --ioengine=libaio --bs=512K --zero_buffers=1 --size=512M | 1024M | 2048M | 4096M | 8192M --numjobs=1 | 2 | 4 | 8
RAID0: *--filename=/dev/md13* ZFS: *--directory=/zpool13*

En las figuras 7 y 8 se puede apreciar el comportamiento de las diferentes unidades de almacenamiento NVMe disponibles en el equipo, con las cuales se realizaron pruebas de escritura y lectura “en crudo”, con distintas cargas de trabajo para uno y dos procesos simultáneos, con lo que podemos notar que la unidad NVMe0 tuvo un rendimiento inferior en todas las pruebas. Se muestra el identificador de unidad y el número de procesos.

En las figuras 9 y 10, se puede notar el aprovechamiento del acceso en paralelo a las unidades de almacenamiento agrupadas en un arreglo RAID0 ”en crudo” y con un dispositivo VDEV ZFS, lo que nos demuestra que el acceso al dispositivo ”en crudo” es más rápido al no contar con capas adicionales de software, sin embargo, ZFS mostró el rendimiento esperado dadas sus características de operación.

Con esta herramienta de pruebas (FIO) podemos ver ciertos puntos donde ZFS demuestra valores superiores sobre el medio “en crudo”, esto es debido al algoritmo de Caché de Reemplazo Adaptable (Adaptive Replacement Cache) que utiliza ZFS, al tener una carga de trabajo que beneficia significativamente su uso, lo cual se muestra en las figuras 11 y 12.

En las figuras 13 y 14, se muestran los resultados obtenidos de Operaciones Entrada/Salida por Segundo (IOPS), generadas durante las pruebas de rendimiento con la herramienta FIO sobre el arreglo RAID0 “en crudo” y el dispositivo VDEV ZFS.

7. Conclusiones

Al termino de las pruebas y obteniendo los resultados antes mostrados se desprenden las siguientes conclusiones.

Es importante conocer la arquitectura de la placa principal, ya que es la encargada de interconectar todos los componentes, de esta forma obtener el máximo rendimiento de los diferentes componentes, tal es el caso de los resultados obtenidos con la unidad NVMe0, donde un puerto que se conecta a través del Concentrador de Plataforma (PCH), induce una latencia que genera un menor rendimiento en comparación con las demás unidades NVMe que se encuentran conectadas en puertos PCIe independientes de este controlador y que tienen acceso directo a los CPU del sistema, con lo que se obtiene un mayor rendimiento.

En el uso de arreglos, que en este caso se utilizó la configuración del arreglo RAID-0 y su equivalente en ZFS (*VDEV*), donde en esta configuración se agrupan las unidades y se agrega la capacidad de cada unidad simultáneamente, de modo que ofrece el mayor rendimiento dado que los datos se leen y se escriben de forma paralela en todas las unidades de la matriz; en este aspecto al realizar operaciones de acceso a dichos arreglos con dos o más procesos (usuarios), es cuando se aprovecha de mejor manera la forma en la que operan y explotar la habilidad de que las unidades trabajen en paralelo.

Podemos notar que el sistema "en crudo" proporciona un mayor rendimiento, sin embargo, al integrarse una capa adicional de software se obtienen múltiples funcionalidades para crear sistemas de almacenamiento eficientes y con una mejor forma de administración. Dado lo anterior se pudo notar que el sistema de archivos ZFS proporciona las siguientes características para mejorar la eficiencia de los sistemas de almacenamiento:

- ZFS compatible con Lustre también utiliza tecnología avanzada de almacenamiento en *caché* y en unidades de estado sólido para mejorar el rendimiento de lectura, incorporando funciones sólidas de verificación de la integridad de los datos para una mayor disponibilidad y confiabilidad del sistema.
- ZFS evita la sobreescritura accidental de datos existentes y no solo detecta la corrupción de datos, sino que también corrige automáticamente los datos incorrectos. Como beneficio adicional, esta infraestructura de software permitirá que exista una amplia gama de sistemas de archivos alternativos bajo Lustre.
- ZFS convierte de forma eficaz las escrituras aleatorias simultáneas que llegan a un destino de almacenamiento de objetos (OST) en un flujo de escrituras secuenciales más rápidas y que consumen menos recursos, lo que es especialmente importante en sistemas grandes con altas tasas de almacenamiento de datos.

7.1. Trabajo futuro

- Analizar a fondo el sistema de archivos Lustre y los componentes de hardware adecuados para obtener el mayor rendimiento posible.

- Estudiar el sistema de archivos ZFS para el aprovechamiento de las características y ventajas que proporciona.

- Continuar experimentando en laboratorio para optimizar el sistema de archivos Lustre con ZFS y así lograr un sistema moderno y acorde a las necesidades del Instituto de Ciencias Nucleares.

Apéndice A: Resultados de Ancho de Banda

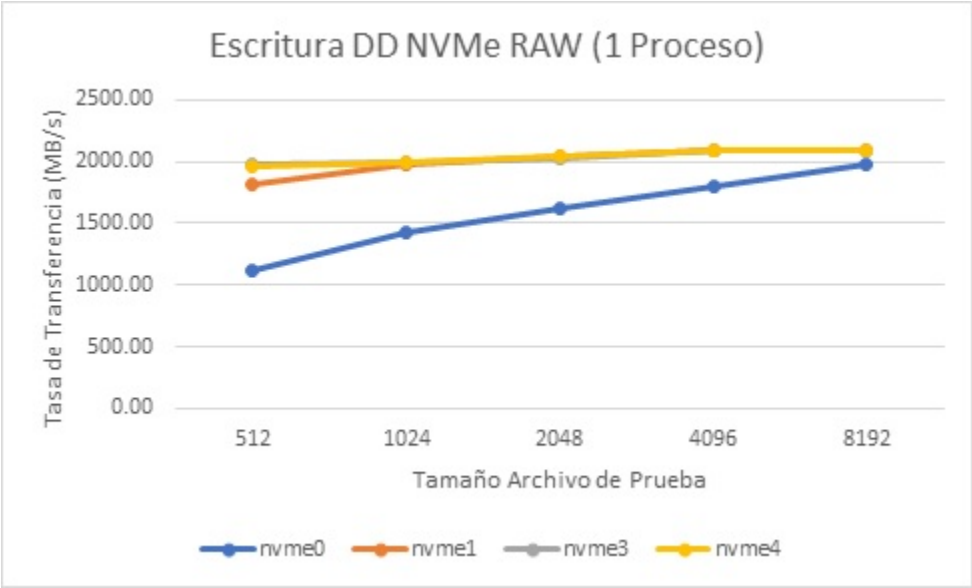

Figura A1. Prueba de Rendimiento Escritura dd: Unidades NVMe "en crudo" con 1 proceso.

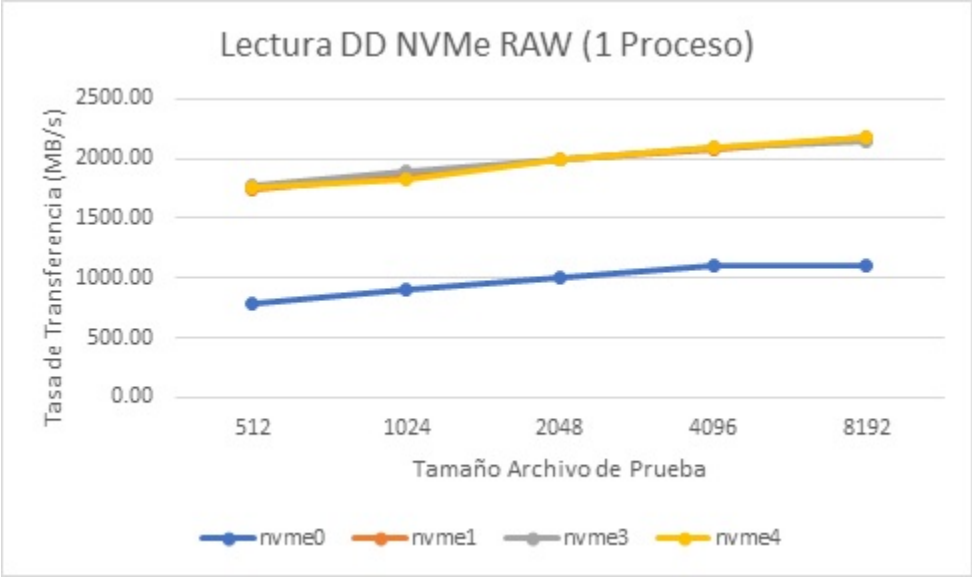

Figura A2. Prueba de Rendimiento Lectura dd: Unidades NVMe "en crudo" con 1 proceso.

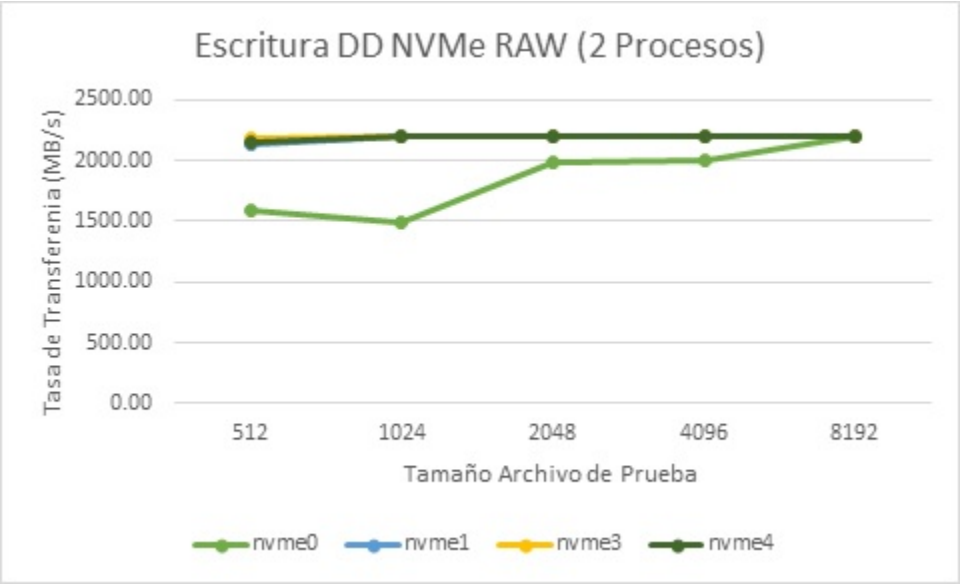

Figura A3. Prueba de Rendimiento Escritura dd: Unidades NVMe "en crudo" con 2 procesos.

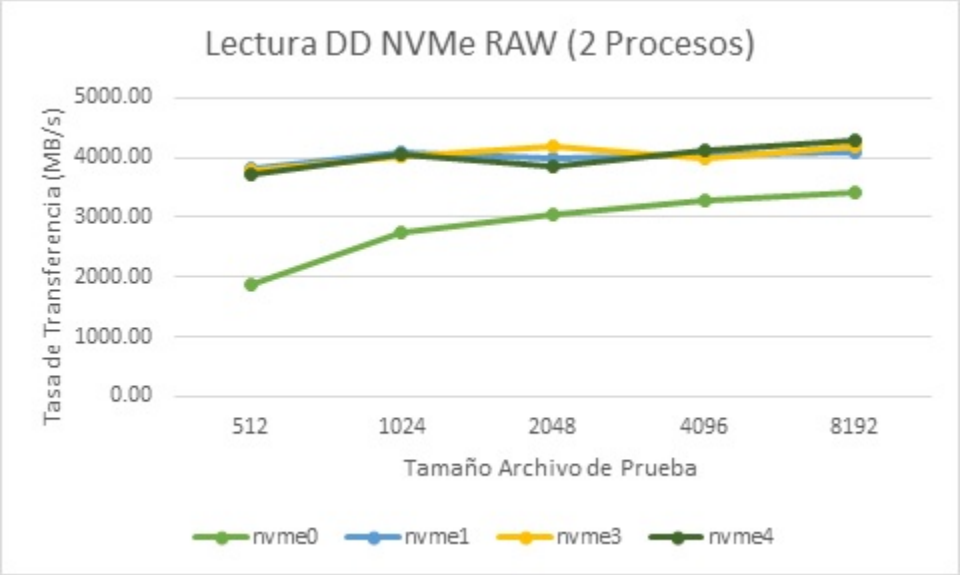

Figura A4. Prueba de Rendimiento Lectura dd: Unidades NVMe "en crudo" con 2 procesos.

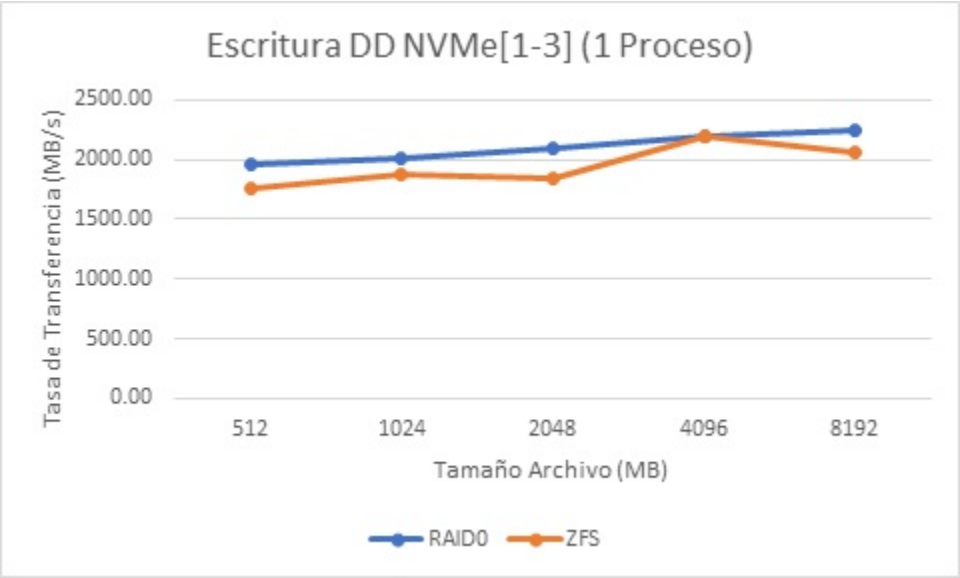

Figura A5. Comparativa de Rendimiento Escritura dd: RAID0 vs ZFS con 1 proceso.

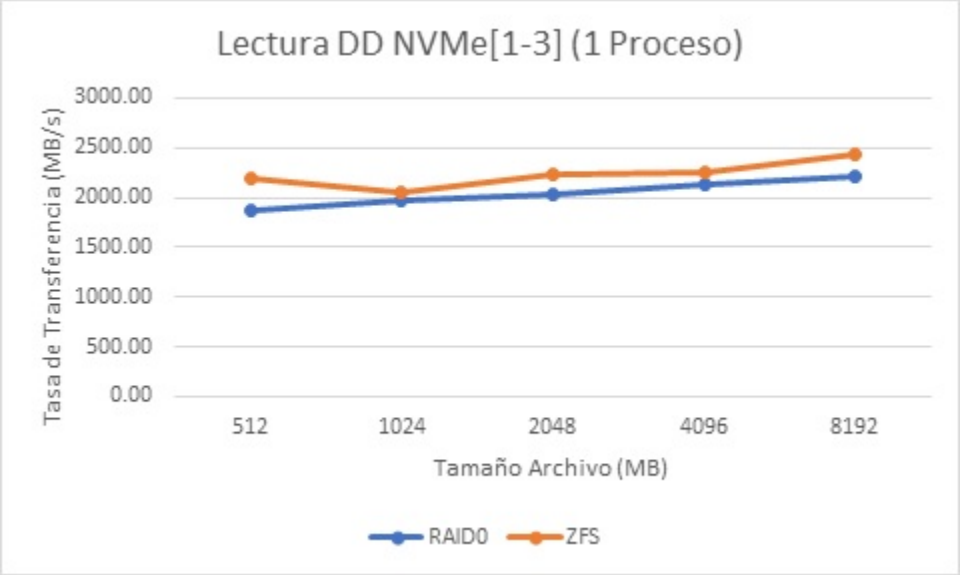

Figura A6. Comparativa de Rendimiento Lectura dd: RAID0 vs ZFS con 1 proceso.

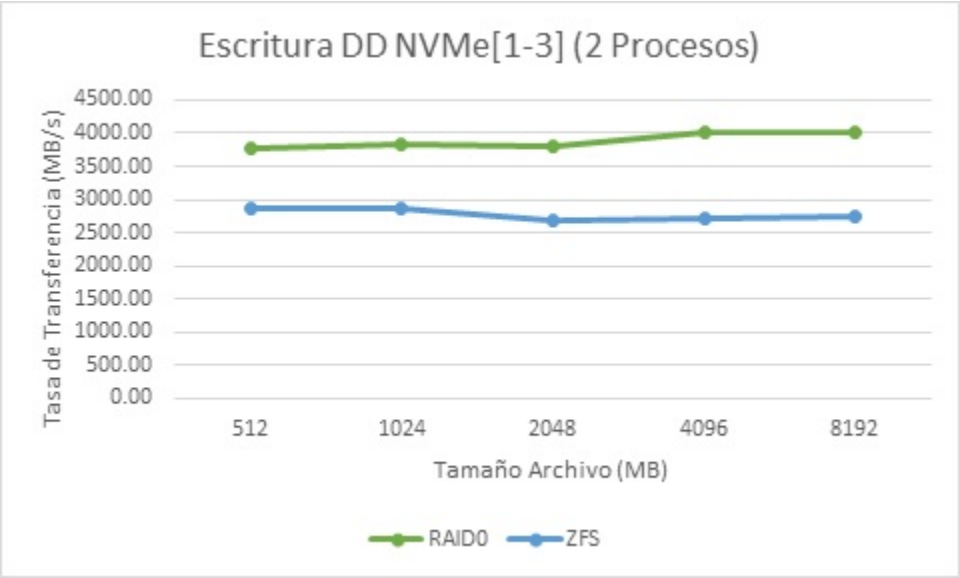

Figura A7. Comparativa de Rendimiento Escritura dd: RAID0 vs ZFS con 2 procesos.

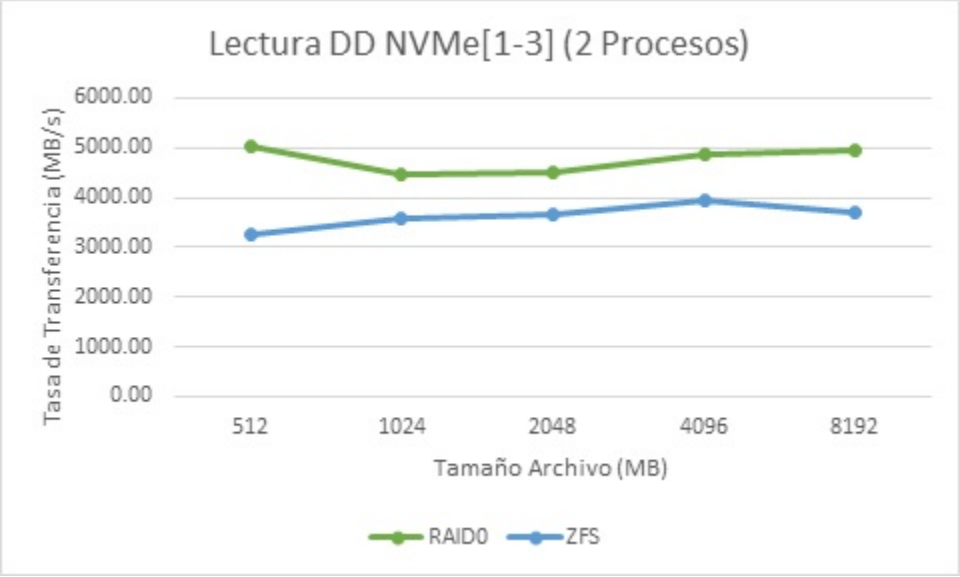

Figura A8. Comparativa de Rendimiento Lectura dd: RAID0 vs ZFS con 2 procesos.

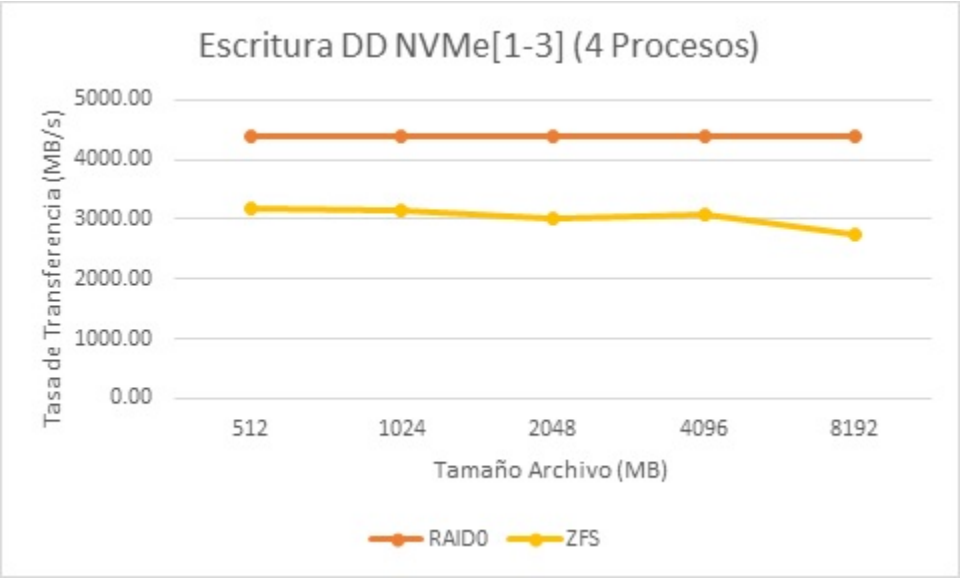

Figura A9. Comparativa de Rendimiento Escritura dd: RAID0 vs ZFS con 4 procesos.

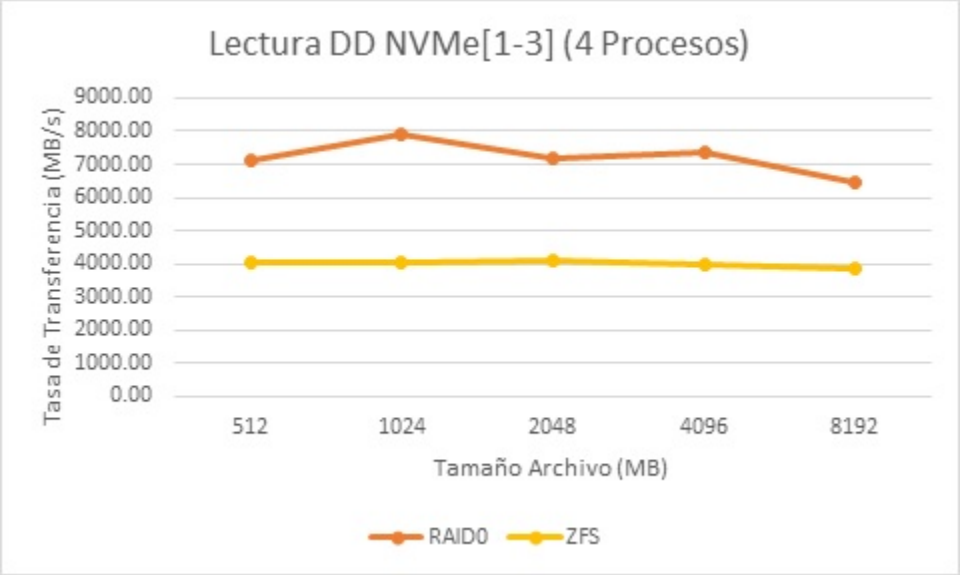

Figura A10. Comparativa de Rendimiento Lectura dd: RAID0 vs ZFS con 4 procesos.

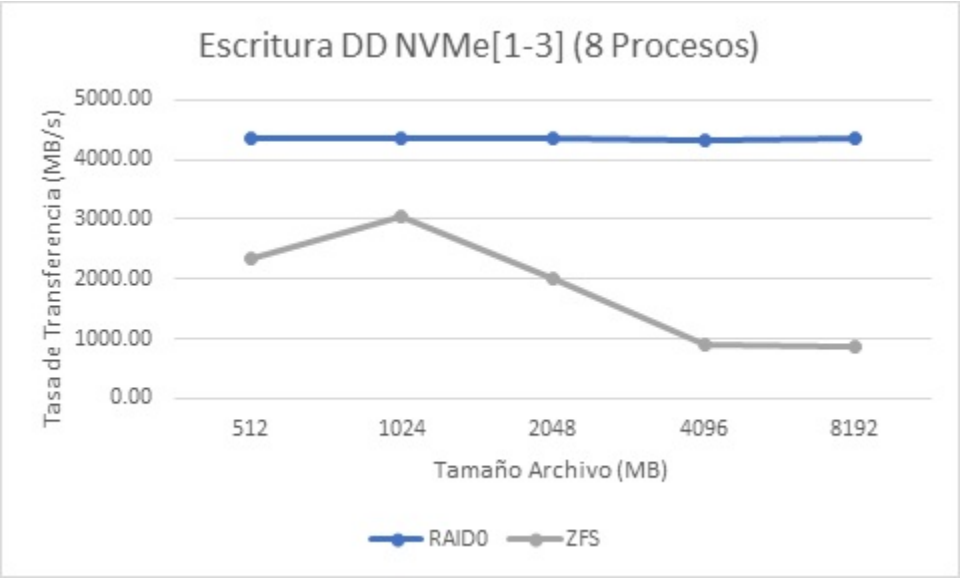

Figura A11. Comparativa de Rendimiento Escritura dd: RAID0 vs ZFS con 8 procesos.

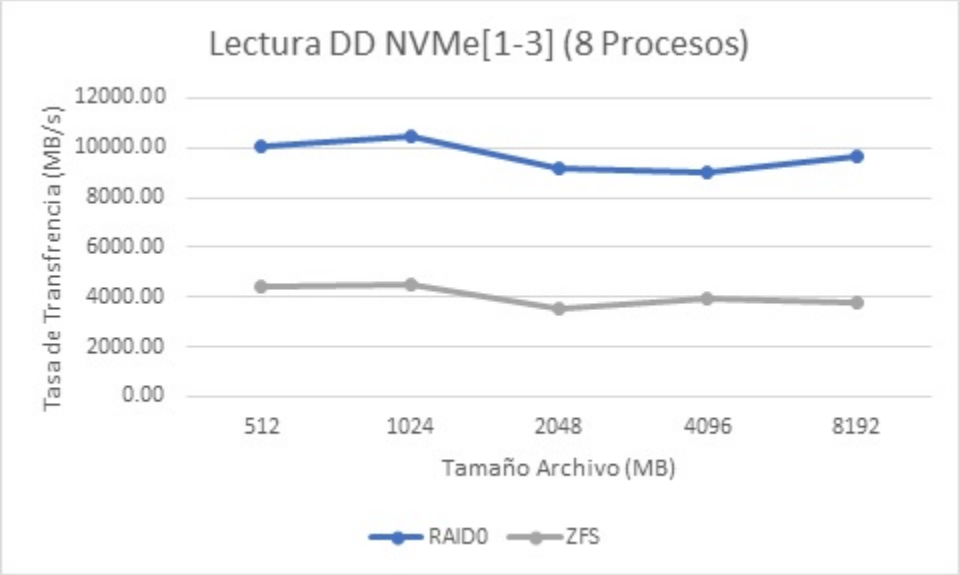

Figura A12. Comparativa de Rendimiento Lectura dd: RAID0 vs ZFS con 8 procesos.

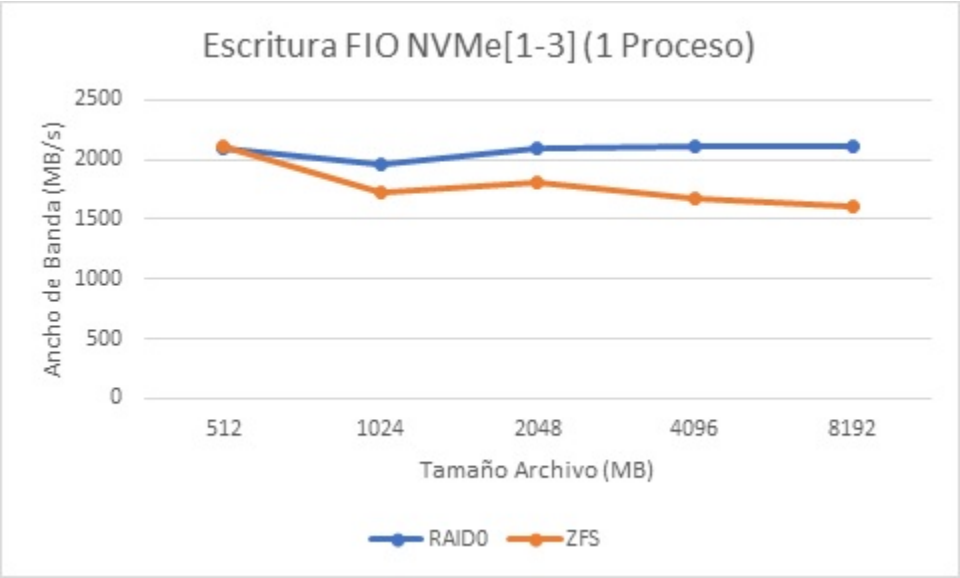

Figura A13. Comparativa de Rendimiento Escritura FIO: RAID0 vs ZFS con 1 proceso.

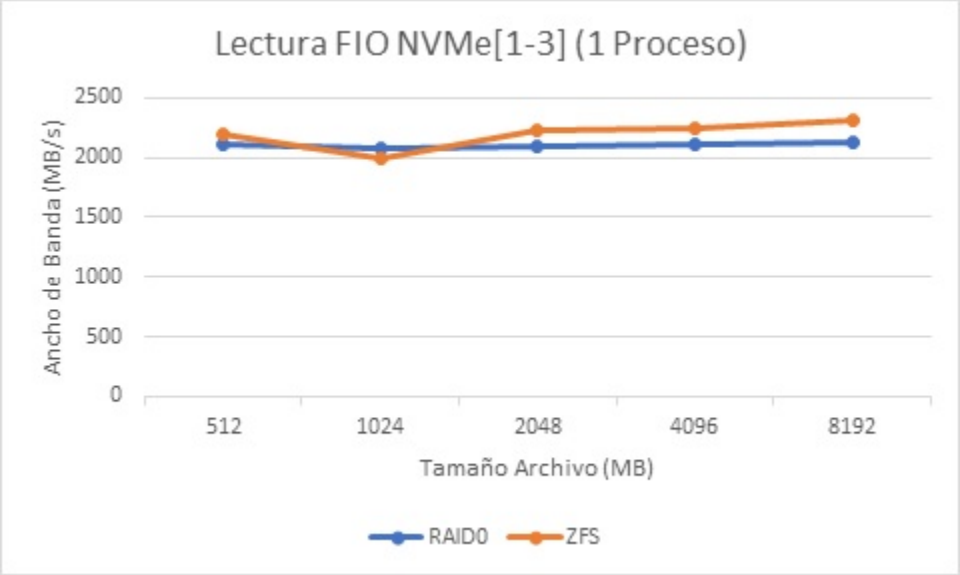

Figura A14. Comparativa de Rendimiento Lectura FIO: RAID0 vs ZFS con 1 proceso.

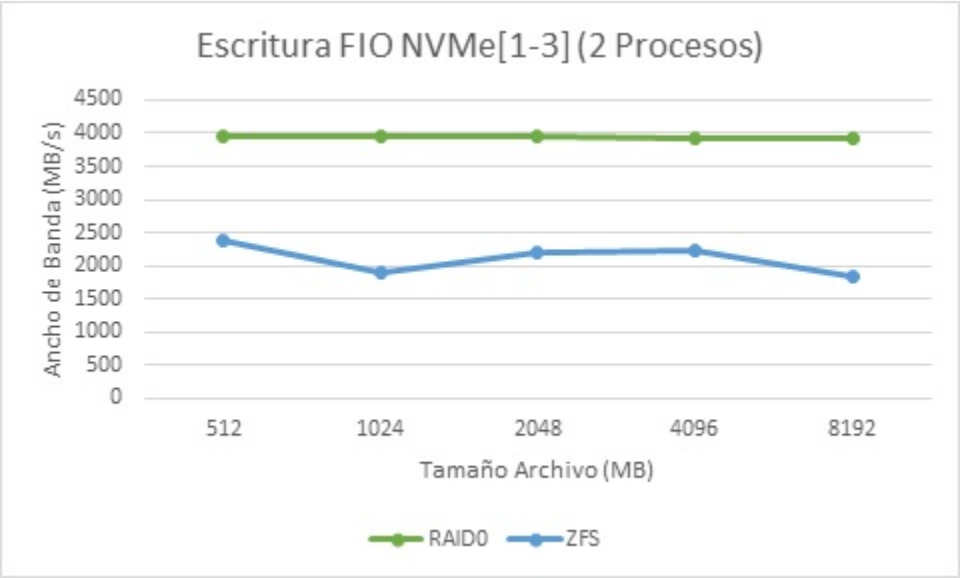

Figura A15. Comparativa de Rendimiento Escritura FIO: RAID0 vs ZFS con 2 procesos.

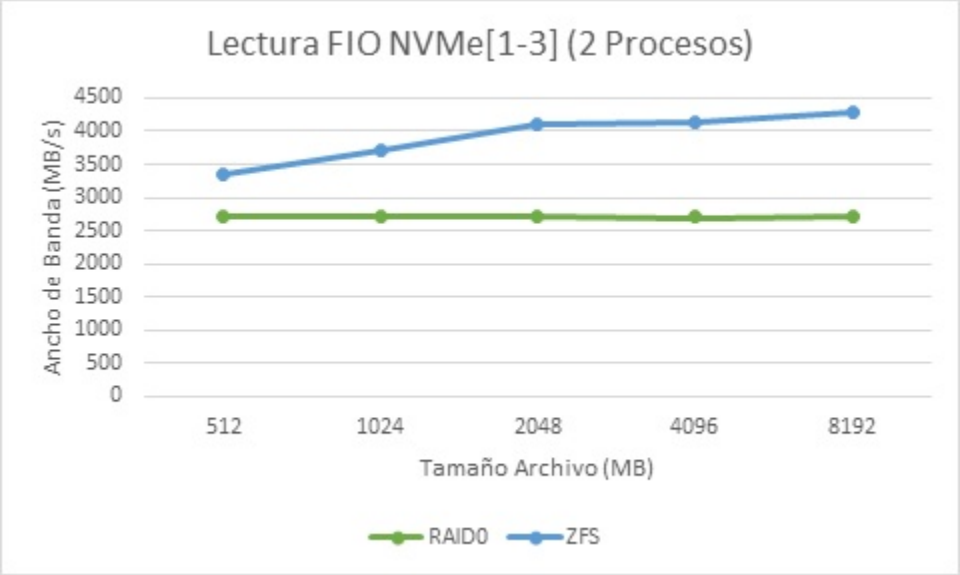

Figura A16. Comparativa de Rendimiento Lectura FIO: RAID0 vs ZFS con 2 procesos.

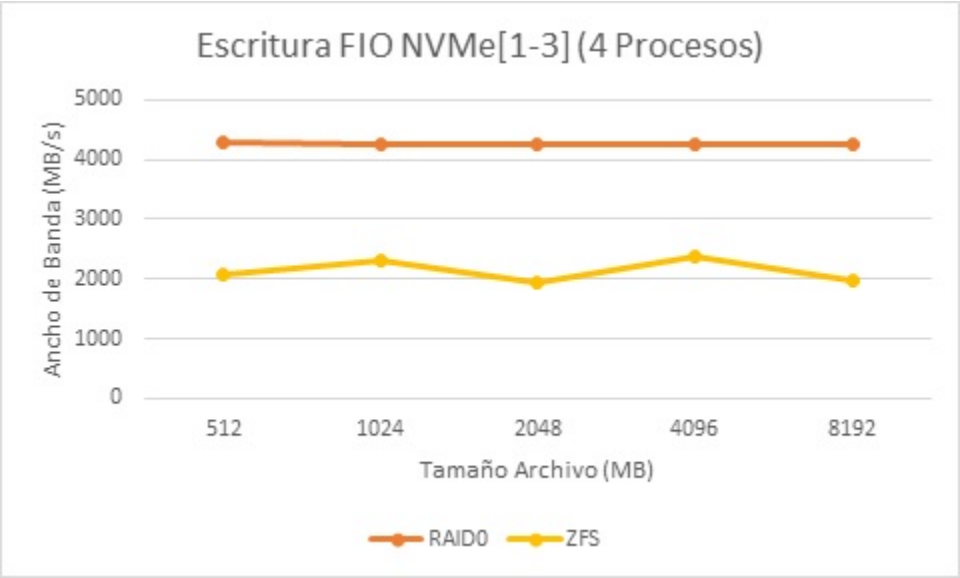

Figura A17. Comparativa de Rendimiento Escritura FIO: RAID0 vs ZFS con 4 procesos.

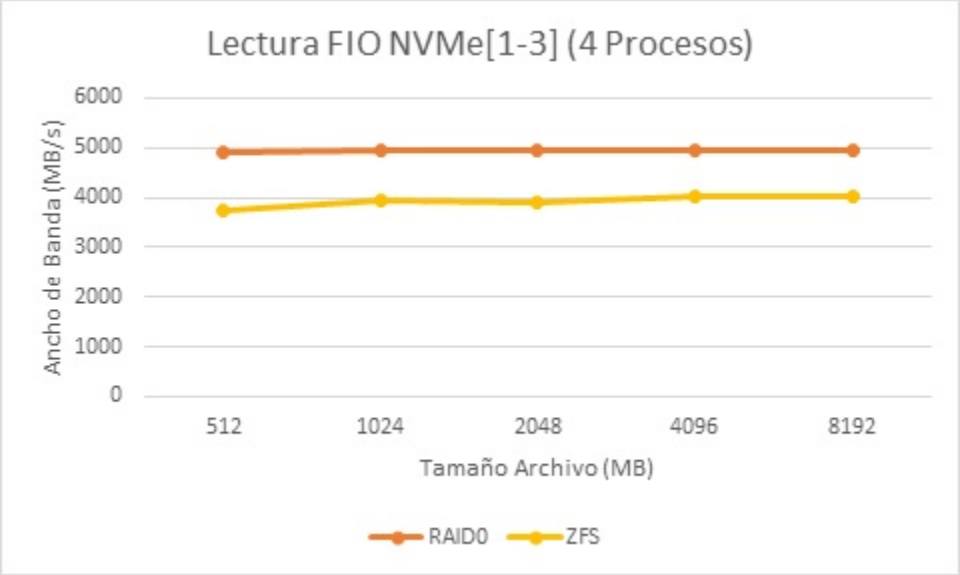

Figura A18. Comparativa de Rendimiento Lectura FIO: RAID0 vs ZFS con 4 procesos.

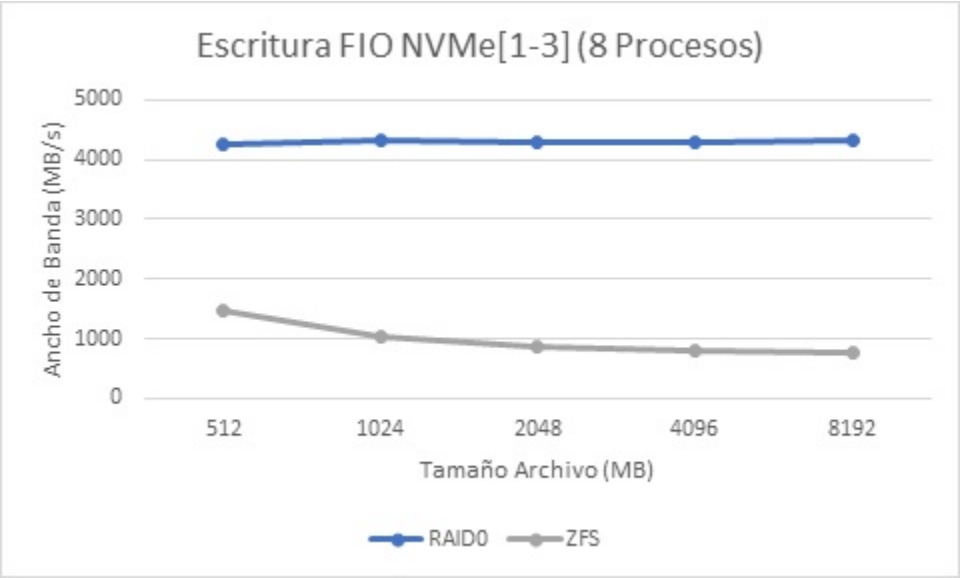

Figura A19. Comparativa de Rendimiento Escritura FIO: RAID0 vs ZFS con 8 procesos.

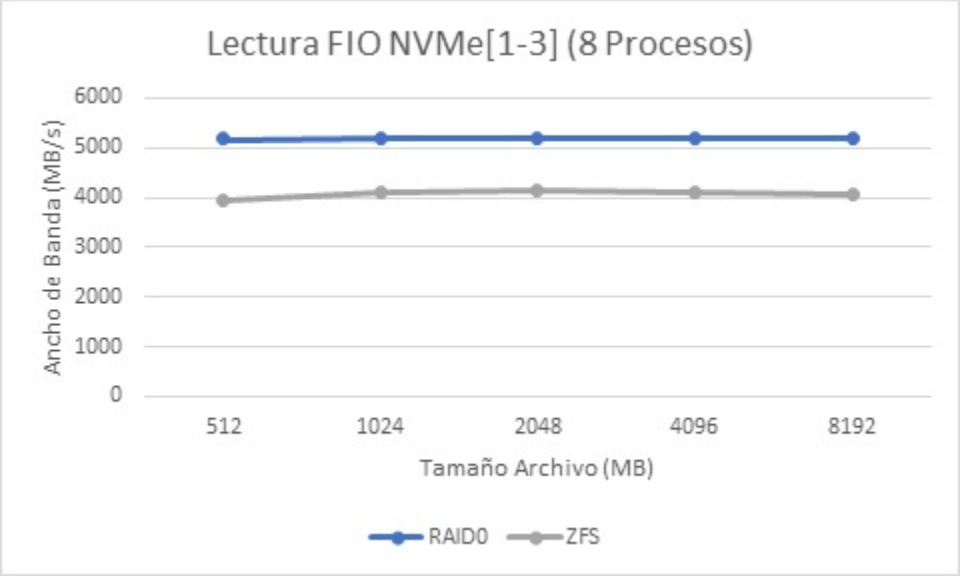

Figura A20. Comparativa de Rendimiento Lectura FIO: RAID0 vs ZFS con 8 procesos.

Apéndice B: Resultados IOPS

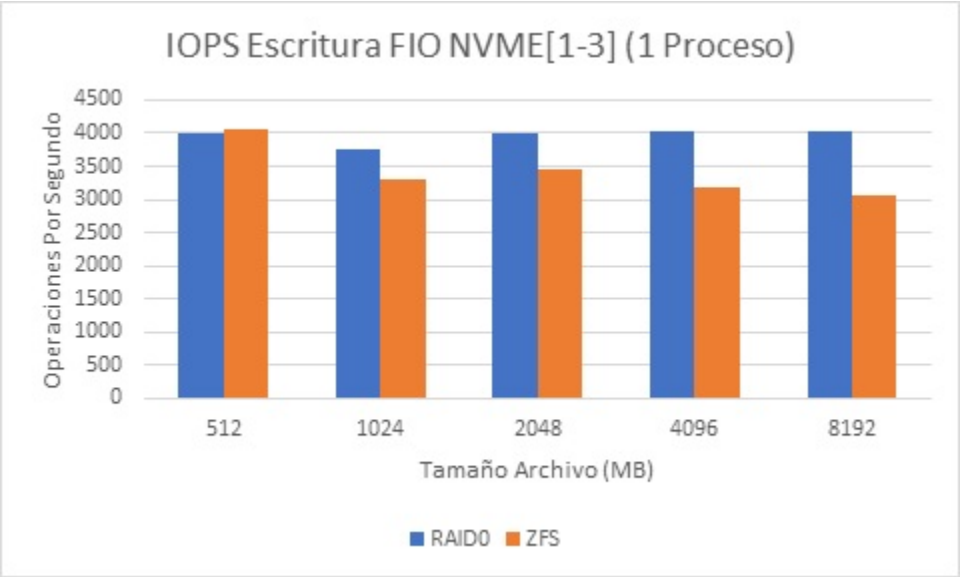

Figura B1. Comparativa de IOPS Escritura FIO: RAID0 vs ZFS con 1 proceso.

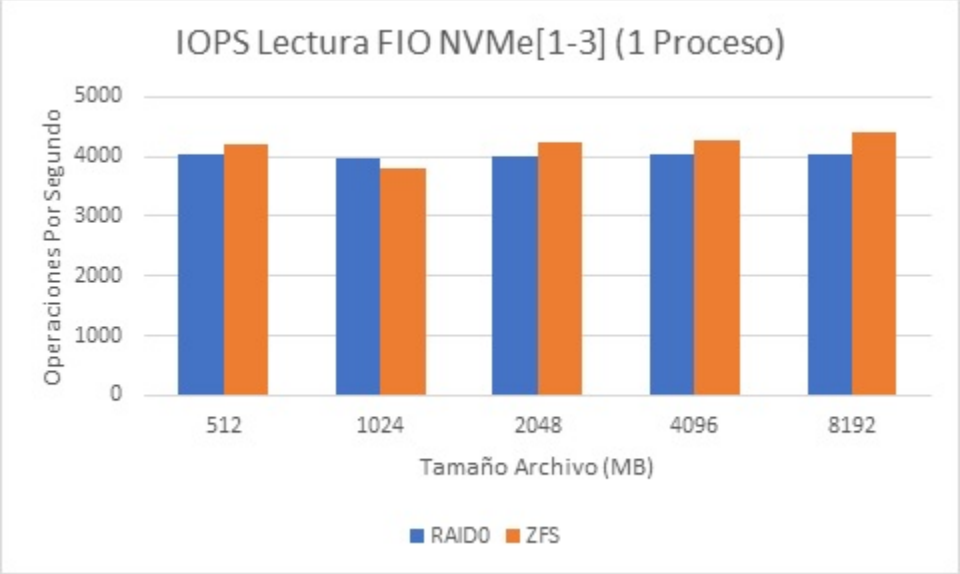

Figura B2. Comparativa de IOPS Lectura FIO: RAID0 vs ZFS con 1 proceso.

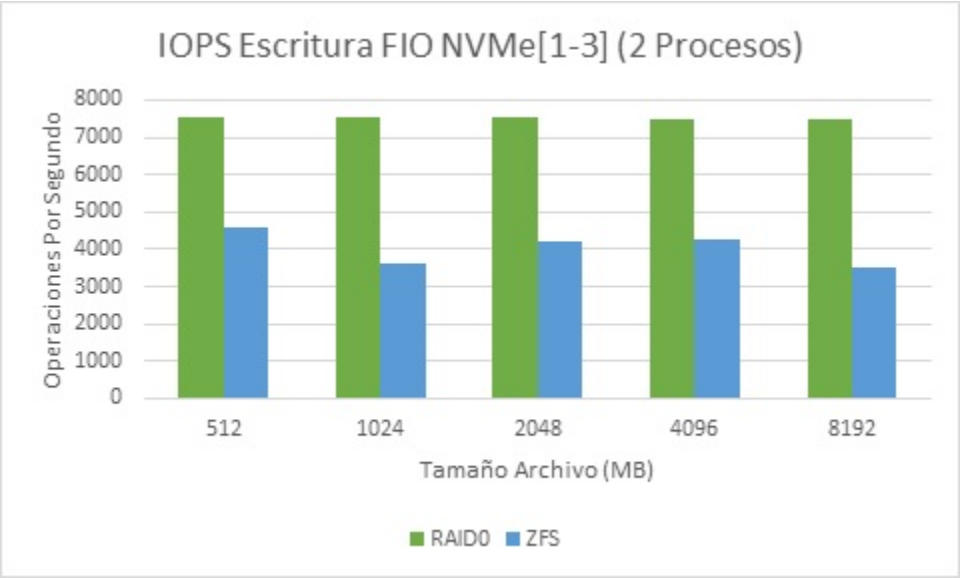

Figura B3. Comparativa de IOPS Escritura FIO: RAID0 vs ZFS con 2 procesos.

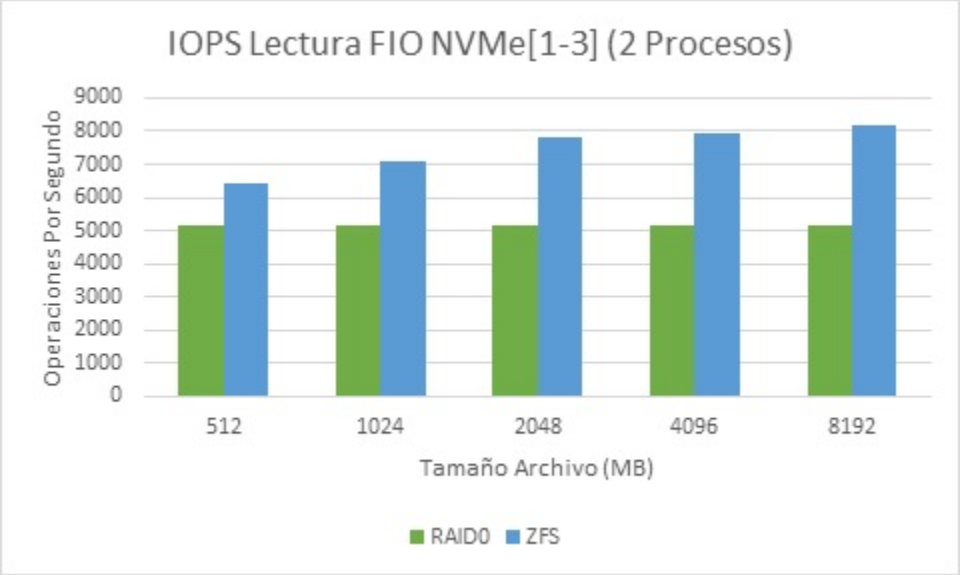

Figura B4. Comparativa de IOPS Lectura FIO: RAID0 vs ZFS con 2 procesos.

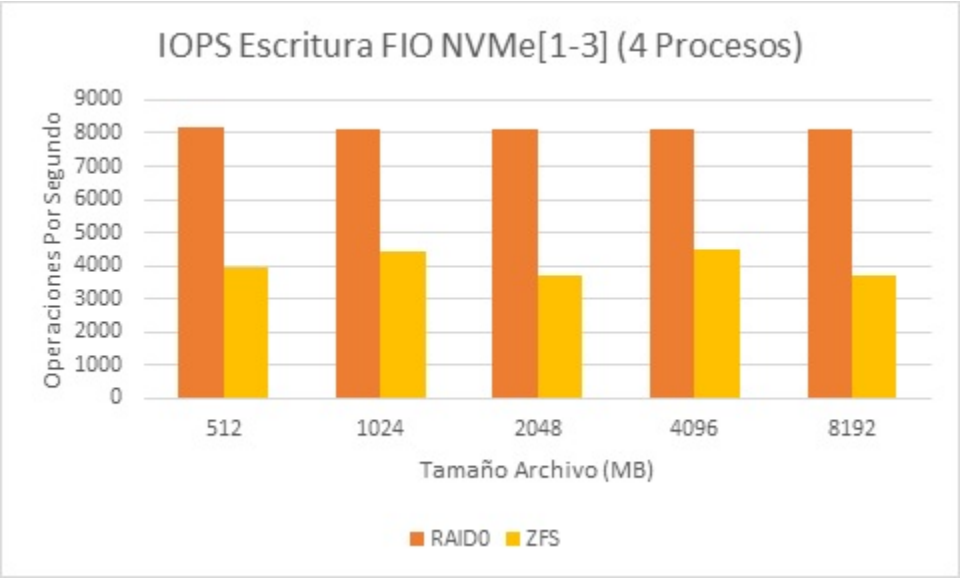

Figura B5. Comparativa de IOPS Escritura FIO: RAID0 vs ZFS con 4 procesos.

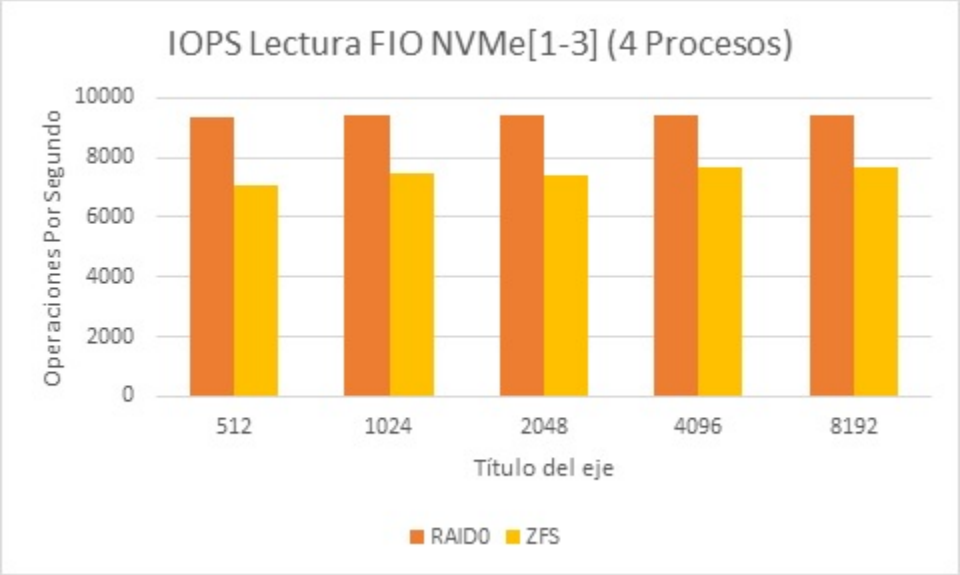

Figura B6. Comparativa de IOPS Lectura FIO: RAID0 vs ZFS con 4 procesos.

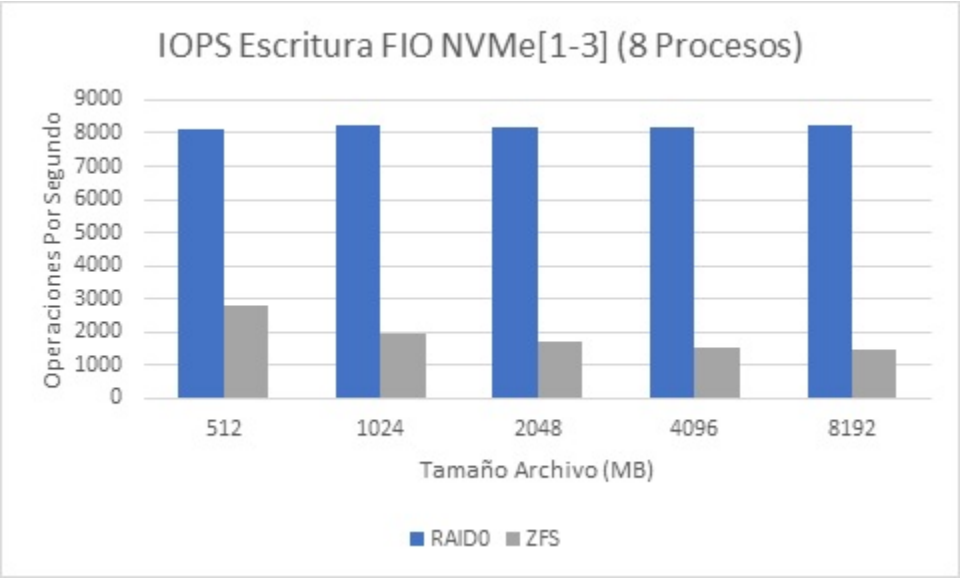

Figura B7. Comparativa de IOPS Escritura FIO: RAID0 vs ZFS con 8 procesos.

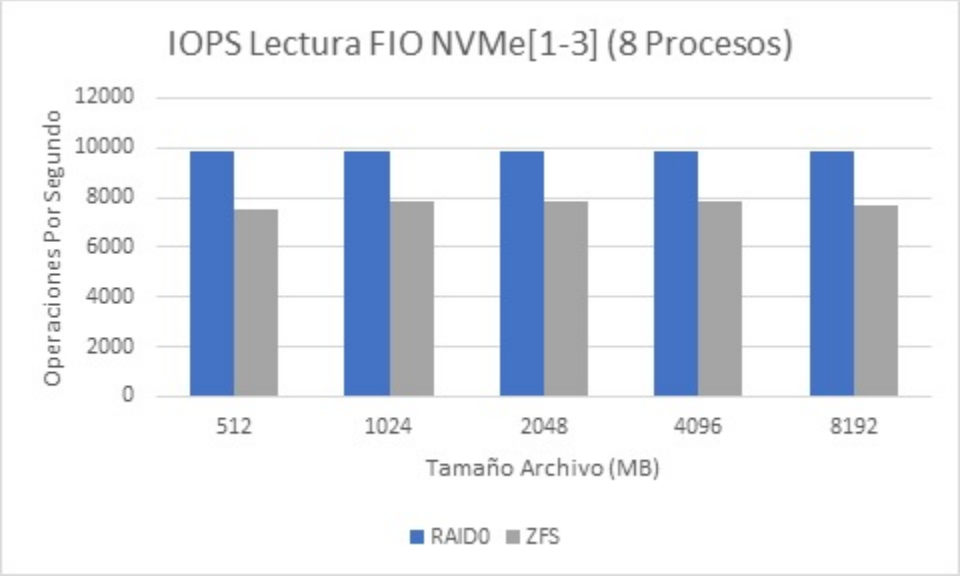

Figura B8. Comparativa de IOPS Lectura FIO: RAID0 vs ZFS con 8 procesos.

Apéndice C: Resultados Tiempo de Ejecución

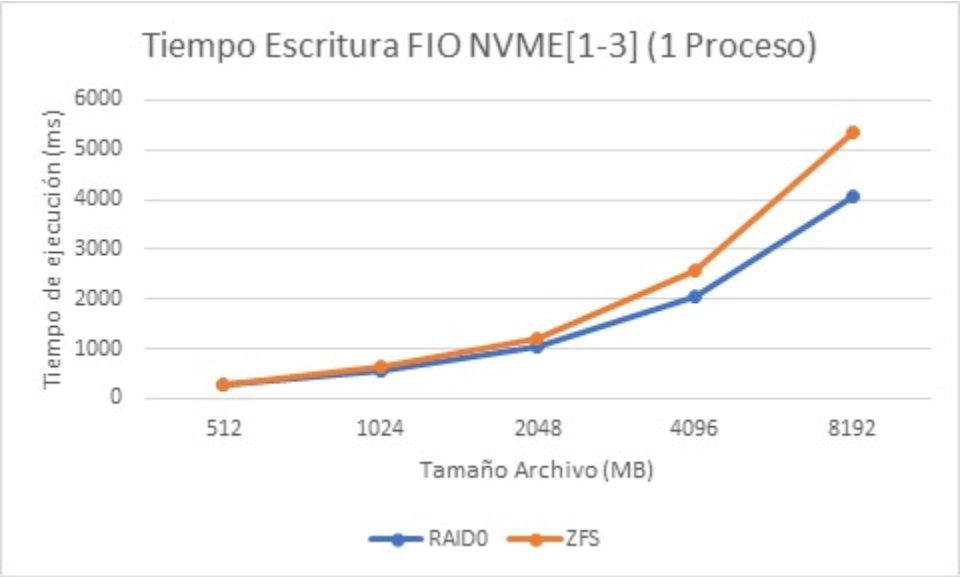

Figura C1. Comparativa de Tiempo Escritura FIO: RAID0 vs ZFS con 1 proceso.

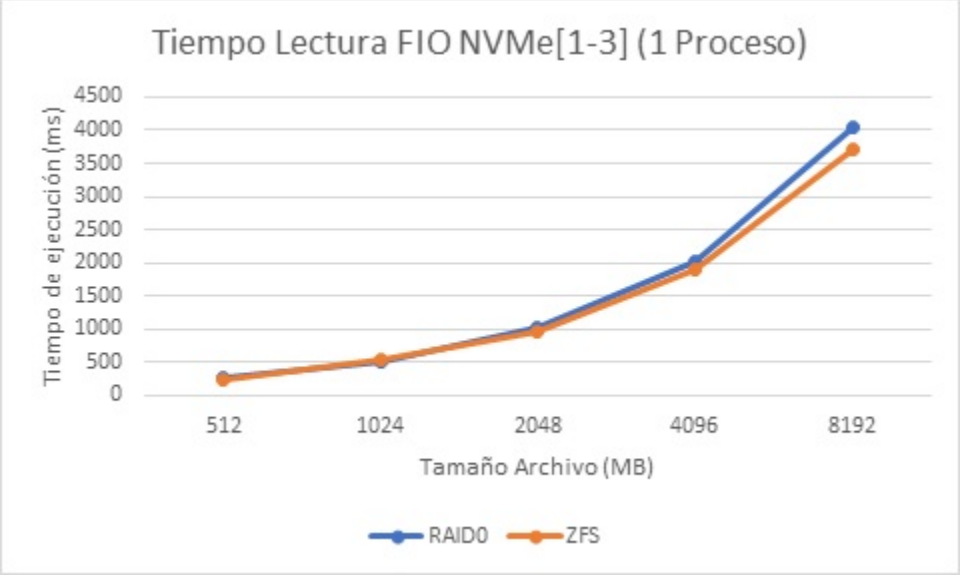

Figura C2. Comparativa de Tiempo Lectura FIO: RAID0 vs ZFS con 1 proceso.

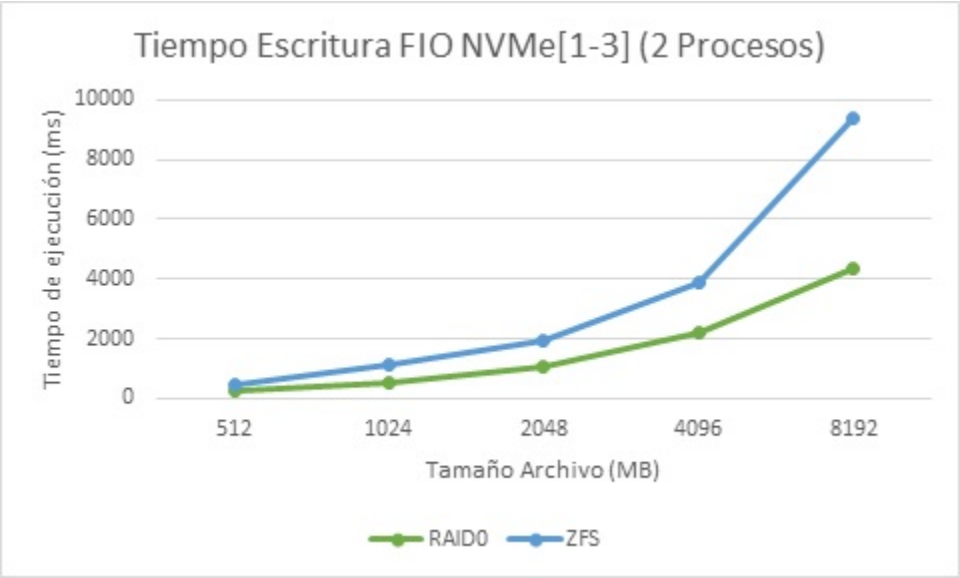

Figura C3. Comparativa de Tiempo Escritura FIO: RAID0 vs ZFS con 2 procesos.

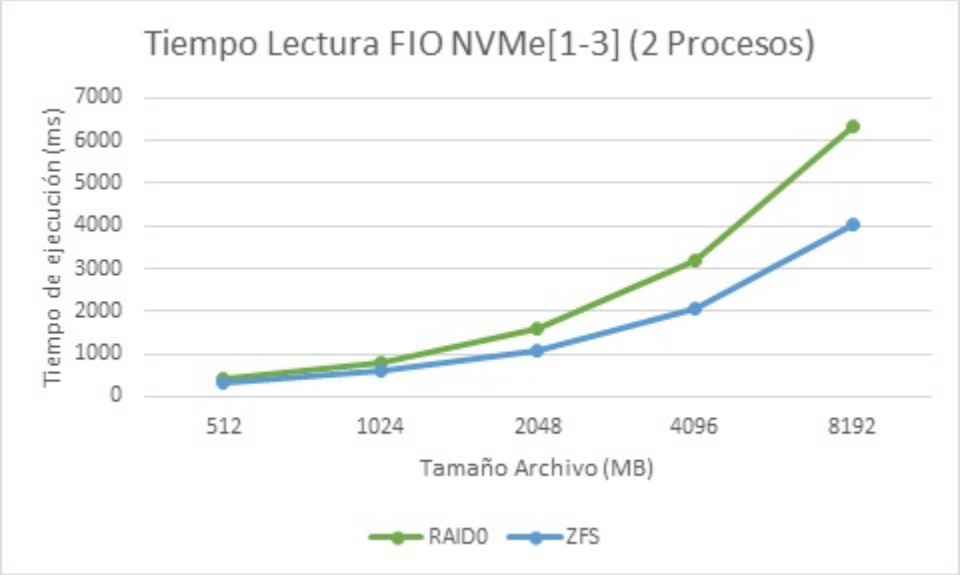

Figura C4. Comparativa de Tiempo Lectura FIO: RAID0 vs ZFS con 2 procesos.

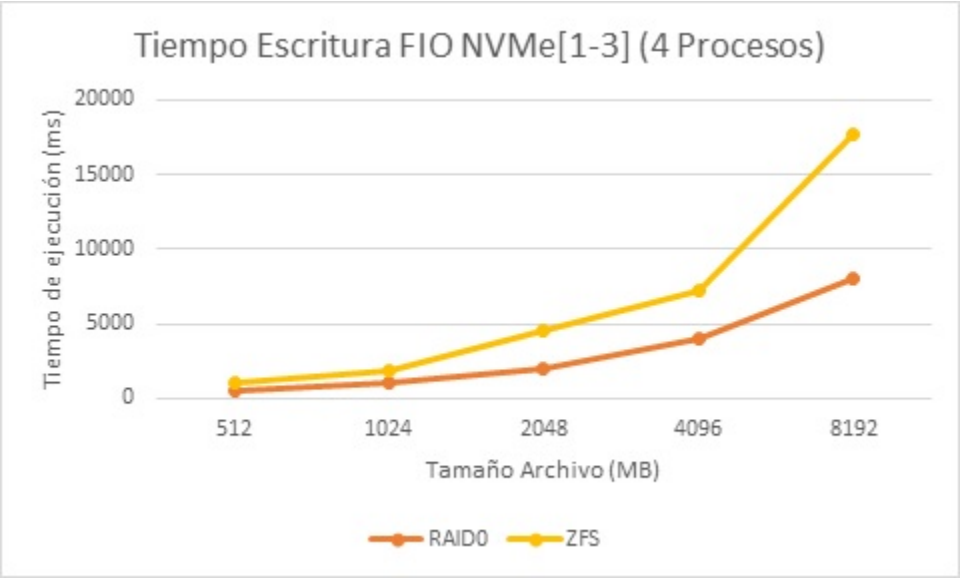

Figura C5. Comparativa de Tiempo Escritura FIO: RAID0 vs ZFS con 4 procesos.

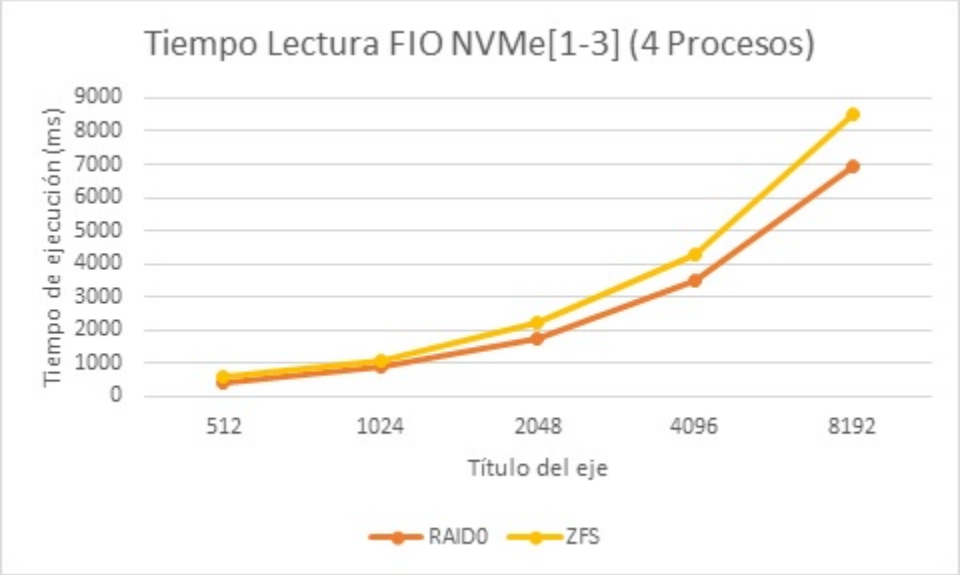

Figura C6. Comparativa de Tiempo Lectura FIO: RAID0 vs ZFS con 4 procesos.

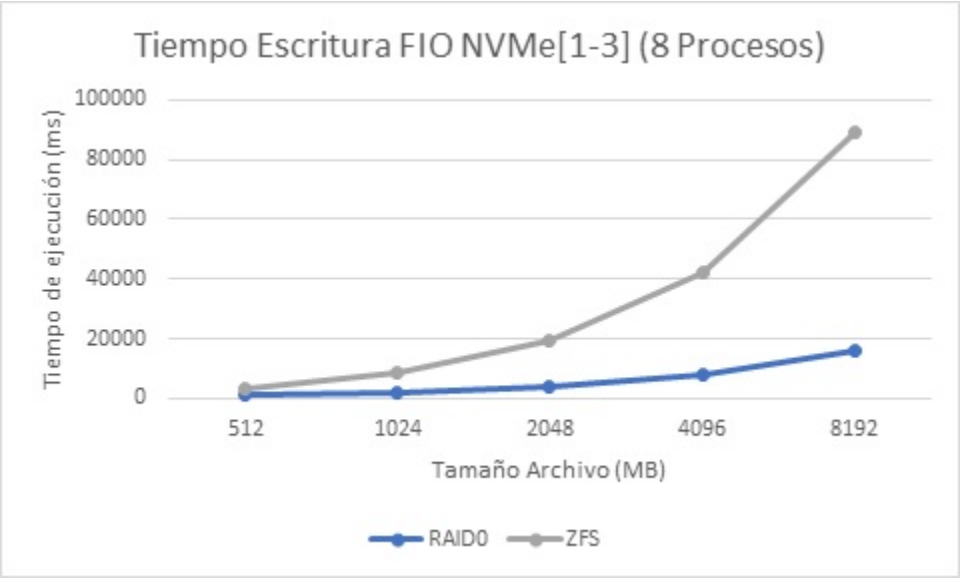

Figura C7. Comparativa de Tiempo Escritura FIO: RAID0 vs ZFS con 8 procesos.

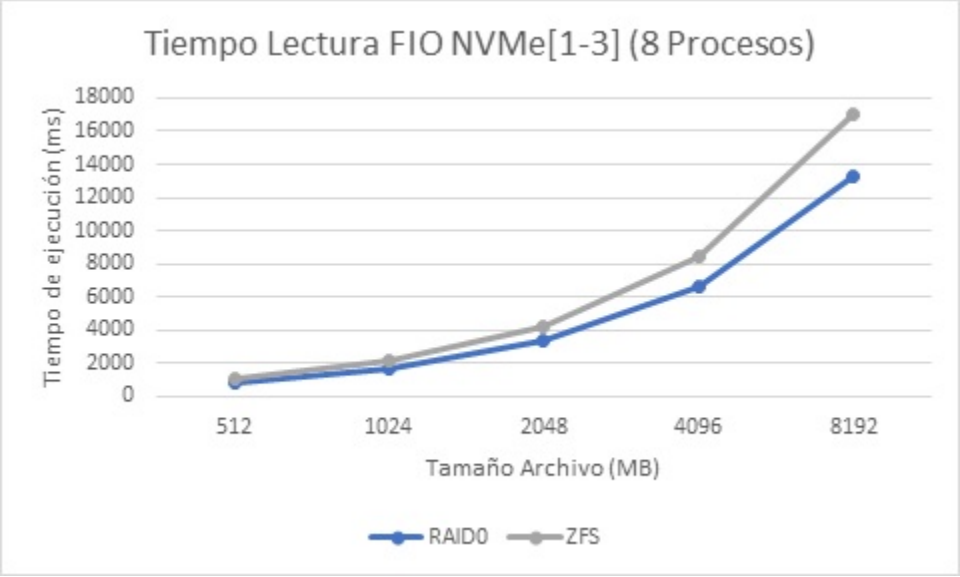

Figura C8. Comparativa de Tiempo Lectura FIO: RAID0 vs ZFS con 8 procesos.

Glosario

ALICE: Acrónimo de "*A Large Ion Collider Experiment*", uno de los ocho experimentos con detectores del Gran Colisionador de Hadrones del CERN.

Ancho de banda: Expresa la cantidad de datos que pueden ser transmitidos en un lapso; en redes de datos se expresa en bps.

ASCII: Código Estándar Americano para el Intercambio de Información por sus siglas en inglés "*American Standard Code for Information Interchange*", es un código de caracteres utilizados en el inglés moderno y basado en el alfabeto latino, utiliza 7 bits para representar letras mayúsculas, minúsculas, números, caracteres especiales y de control, mediante números decimales que van del 0 al 127.

Base de datos: Conjunto de datos organizados de modo tal que resulte fácil acceder a ellos, gestionarlos y actualizarlos.

Bit: Abreviatura de "*binary digit*" (dígito binario). El bit es la unidad más pequeña de almacenamiento en un sistema binario dentro de una computadora.

bps: bits por segundo.

Byte: Unidad de información utilizada por las computadoras. Cada byte está compuesto por ocho bits.

Chipset: Es el conjunto de circuitos integrados que se encarga de controlar determinadas funciones del equipo, como la interacción del microprocesador (CPU) con la memoria, el controlador de puertos PCI, USB, LAN, audio, video, entre otros componentes del equipo.

CERN: Organización Europea para la Investigación Nuclear, organización de investigación europea que opera el laboratorio de física de partículas más grande del mundo.

Cliente: Es una aplicación informática o una computadora que consume un servicio remoto en otra computadora conocida como servidor, normalmente a través de una red de comunicaciones.

Clúster: Es un conjunto de computadoras de hardware comunes, interconectadas entre sí mediante redes de comunicaciones y que se comportan como si fueran una única.

Comando: Instrucción que un usuario da al sistema operativo de la computadora para realizar determinada tarea.

CPU: Unidad Central de Procesamiento (del inglés, "*Central Processing Unit*"), es la parte de hardware en una computadora u otros dispositivos, que su trabajo es interpretar las instrucciones de un programa informático mediante la realización de operaciones aritméticas y lógicas.

EBCDIC: Código de Intercambio Decimal Codificado en Binario Extendido, por sus siglas en inglés "*Extended Binary Coded Decimal Interchange Code*", es un código estándar que representa caracteres alfanuméricos, de control y signos de puntuación, compuestos por 8 bits, con el que se definen un total de 256 caracteres.

Ethernet: Tecnología estándar para redes de área local.

FLOPS: Operaciones de Punto Flotante por Segundo (del inglés, "*Floating Point Operations Per Second*"), una medida del rendimiento de una computadora, principalmente para cálculos científicos.

Gigabit (Gb): Unidad de medida de información que equivale a 10^9 bits = 1,000,000,000 de bits.

Gigabyte (GB): Unidad de medida de información que equivale a 10^9 bytes = 1,000,000,000 de bytes.

GPL: Acrónimo de "*General Public License*" (Licencia Pública General), es una licencia de derecho de autor usada en el software libre y de código abierto, que garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software.

Hardware: Todos los componentes físicos de una computadora y sus periféricos.

HAWC: Acrónimo de "*High Altitude Water Cherenkov*", observatorio diseñado para detectar rayos gamma de origen cósmico a través de la medición de cascadas atmosféricas.

HDD: Unidad de Disco Duro (del inglés, "*Hard Disk Drive*"), dispositivo para el almacenamiento de datos que utiliza un sistema de grabación magnética para almacenar y recuperar archivos digitales.

Host: Anfitrión, se usa en informática para referirse a las computadoras u otros dispositivos conectados que proveen y utilizan servicios de estos.

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos (del inglés "*Institute of Electrical and Electronics Engineers*") , es una asociación mundial dedicada a la normalización y desarrollo en áreas técnicas como la electricidad, electrónica, computación, cibernética, telecomunicaciones, biomedicina, matemáticas, software, etc.

IOPS: Operaciones de Entrada y Salida por Segundo (del inglés, "*Input Output Operations Per Second*"), unidad utilizada para medir el rendimiento de dispositivos de almacenamiento.

IPMI: Interfaz de Administración de Plataforma Inteligente (del inglés, "*Intelligent Platform Management Interface*"), es un conjunto de instrucciones para el subsistema autónomo de la computadora que proporciona capacidades de administración y monitoreo independientemente del CPU, el firmware y el sistema operativo.

Kernel: Es el software que constituye la parte fundamental del sistema operativo, es el principal responsable de permitir a los distintos programas, acceso seguro al hardware de la computadora y gestionar los recursos mediante servicios de llamada al sistema, también llamado Núcleo en español.

Kilobyte (kB): Unidad de medida de información que equivale a 10^3 bytes = 1,000 bytes.

LAN: Red de área local o red de computadoras que abarca un área reducida (del inglés, "*Local Area Network*").

Latencia: Se refiere a la demora de tiempo entre el ingreso y la ejecución de un comando.

Megabyte (MB): Unidad de medida de información que equivale a 10^6 bytes = 1,000,000 de bytes.

Metadatos: Son aquellos datos que describen el contenido de los archivos o la información de estos.

Motherboard: También llamada placa madre, placa base o placa principal, es la tarjeta electrónica en la que se conectan los componentes que constituyen una computadora, ya sea integrados directamente o conectados a través de puertos de expansión.

NASA: Administración Nacional de Aeronáutica y el Espacio (del inglés, "*National Aeronautics and Space Administration*"), agencia del gobierno estadounidense encargada del programa espacial e investigación aeronáutica.

Periférico: Todo dispositivo que se conecta a una computadora.

Petabyte (PB): Unidad de medida de información que equivale a 10^{15} bytes = 1,000,000,000,000,000 de bytes.

POSIX: Norma escrita por la IEEE que define una interfaz estándar del sistema operativo y el entorno (del inglés, "*Portable Operating System Interface for X*"), incluyendo un intérprete de comandos.

RAID: Grupo o matriz redundante de discos independientes (del inglés, "*Redundant Array of Independent Disks*"), el cual hace referencia a un sistema de almacenamiento de datos que utiliza múltiples unidades entre las cuales se distribuyen o replican los datos.

Raw Data: Son también conocidos como datos fuente primarios, son aquellos que se recopilan de la fuente y no han sido procesados para su uso.

Raw Device: Es un dispositivo lógico sin formato, asociado a un archivo de dispositivo de caracteres permitiendo acceder directamente a un dispositivo de almacenamiento, lo que les permite administrar la forma en que se almacenan en caché los datos, en vez de transferir esta tarea al sistema operativo.

Repositorio: Espacio centralizado donde se almacena, organiza, mantiene y difunde información digital, normalmente archivos informáticos que pueden contener trabajos científicos, conjuntos de datos o programas de software.

SAS: Interfaz de transferencia de datos en serie (del inglés, "*Serial Attached SCSI*"), sucesor de la interfaz SCSI (del inglés, "*Small Computer System Interface*").

SATA: Interfaz de bus de computadora (del inglés, "*Serial ATA*") para la transferencia de datos desde un dispositivo de almacenamiento (Unidad de Disco Duro, Unidad de Estado Sólido, Unidad de Disco Óptico).

Servidor: Aplicación en ejecución capaz de atender las peticiones de un cliente y devolverle una respuesta, en términos de hardware, se refiere a las computadoras dedicadas individualmente para este fin.

Software: Término general que designa los diversos tipos de programas usados en computación.

SSD: Unidad de Estado Sólido (del inglés, "*Solid State Drive*"), es un tipo de dispositivo para el almacenamiento de datos que utiliza memoria no volátil y que proporciona mayor velocidad de lectura y escritura de datos.

Terabyte (TB): Unidad de medida de información que equivale a 10^{12} bytes = 1,000,000,000,000 de bytes.

VLAN: Es un método para crear redes lógicas independientes dentro de una misma red física (del inglés, "*Virtual Local Area Network*").

Referencias

[1] Axboe, J. (2017). FIO - Flexible I/O Tester Documentation.
https://fio.readthedocs.io/en/latest/fio_doc.html

[2] Coker, R. (2020). Bonnie++: Test Hard Drive Performance - Linux man page.
https://linux.die.net/man/8/bonnie++

[3] Grun, P. (2010). Introduction to InfiniBand for End Users. Mellanox.
https://www.mellanox.com/pdf/whitepapers/Intro_to_IB_for_End_Users.pdf

[4] Henwood, R. & Mannthey, K. (2013). Components of a Lustre Filesystem - Whamcloud Community Wiki.
https://wiki.whamcloud.com/display/PUB/Components+of+a+Lustre+filesystem

[5] Intel Corp. (2021). Chipset Intel C621 Especificaciones de productos.
https://ark.intel.com/content/www/es/es/ark/products/97338/intel-c621-chipset.htm

[6] Lustre. (2017). Introduction to Lustre Architecture.
http://wiki.lustre.org/images/6/64/LustreArchitecture-v4.pdf

[7] Lustre. (2020). Installing the Lustre Software - Lustre Wiki.
https://wiki.lustre.org/Installing_the_Lustre_Software

[8] Lustre. (2020). Lustre with ZFS Install - Lustre Wiki.
https://wiki.lustre.org/Lustre_with_ZFS_Install

[9] Malpica, J. (2019). El Centro de Datos y Cómputo de Alto Rendimiento de la UNAM, posee la colección de datos científicos más importante de México. Mexico Ambiental.
https://www.mexicoambiental.com/el-centro-de-datos-y-computo-de-alto-rendimiento-de-la-unam-posee-la-coleccion-de-datos-cientificos-mas-importante-de-mexico/

[10] Mellanox. (2008). InfiniBand and TCP in the Data Center.
https://www.mellanox.com/pdf/whitepapers/IB_TCP_in_the_datacenter_WP_110.pdf

[11] NVM Express. (2020). Frequently Asked Questions.
https://nvmexpress.org/education/faqs/

[12] OpenZFS. (2021). OpenZFS Documentation.
https://openzfs.github.io/openzfs-docs/

[13] OpenZFS on Linux. (2021). ZFS on Linux Man Pages.
https://zfsonlinux.org/manpages/0.8.6/index.html

[14] Sterling, T., Anderson, M. & Brodowicz, M. (2018).
High Performance Computing: Modern Systems and Practices. Morgan Kaufmann, an imprint of Elsevier.

[15] TOP500 | Lists. (2021).
https://top500.org/lists/top500/

[16] UNAM ICN. (2020). Especialidad en Cómputo de Alto Rendimiento (2020-1).
https://turing.nucleares.unam.mx/cursos/course/view.php?id=13

[17] UNAM ICN. (2020). Historia del Instituto de Ciencias Nucleares de la UNAM.
http://www.nucleares.unam.mx/historia.php

[18] UNAM ICN. (2020). Instalación de Clusters CentOS 7 UNACH-UNAM 2020.
https://turing.nucleares.unam.mx/cursos/course/view.php?id=14

[19] UNAM ICN. (2020). Instituto de Ciencias Nucleares: Misión y Visión.
http://www.nucleares.unam.mx/mision.php