

Análise comparativa de desempenho dos sistemas de arquivos NFS e SMB/CIFS integrados ao DFS do windows server

*Comparative Performance Analysis of NFS and SMB/CIFS File Systems Integrated
into Windows Server DFS*

*Análisis comparativo del desempeño de los sistemas de archivos NFS y SMB/CIFS
integrados al DFS de Windows Server*

Adilson José da Silva Silvério¹

<https://orcid.org/0009-0004-4305-2449>

Lázaro Emílio Makili²

<https://orcid.org/0000-0002-7371-1407>

RECEBIDO: 17 fevereiro, 2025 | **ACEITE:** 01 setembro, 2025 | **PUBLICADO:** 25 novembro, 2025

Como citar: Silvério, A., Makili, L. (2025). Análise comparativa de desempenho dos sistemas de arquivos NFS e SMB/CIFS integrados ao DFS do windows server. *RAC: Revista Angolana de Ciências*, 7(2), e070209. <https://doi.org/10.54580/R0702.09>

RESUMO

Os sistemas de arquivos distribuídos (DFS) são recursos tecnológicos que permitem a partilha e acesso a informações dentro de uma infra-estrutura de rede de forma segura e eficiente. Na medida em que a adopção destes tipos de sistema é de vital importância para uma instituição e considerando a importância da escolha adequada do protocolo para a confiabilidade e o desempenho da infra-estrutura, o estudo em causa tem como propósito analisar comparativamente o funcionamento e desempenho dos sistemas de arquivos Network File System (NFS) e Server Message Block (SMB)/Common Internet File System (CIFS) integrados ao serviço DFS do Windows Server, analisando variáveis como taxa de transferência, utilização de CPU (Central Processing Unit) e threads sob diferentes cargas de trabalho. O desenho do ambiente de teste teve como base a infra-estrutura de rede da Universidade Katyavala Bwila situada na província de Benguela-Angola, assente na arquitectura cliente-servidor. Com base nos resultados apresentados, permitiu-se conhecer a capacidade de resposta do sistema de acordo às diferentes operações realizadas (escrita, reescrita, leitura, releitura e outras) em arquivos e registos de diferentes tamanhos, onde indicaram um desempenho semelhante entre os protocolos em testes de carga intensiva, com pequenas variações na taxa de transferência em operações específicas. Estas informações fornecem subsídios relevantes para administradores de redes, especialistas e a comunidade científica, na definição de políticas de partilha de arquivos e na escolha do protocolo mais adequado para implantações em ambientes DFS.

¹ Instituto Politécnico da Universidade Katyavala Bwila Departamento de Ensino e Investigação de Informática, Angola. p.eng.ajs.silverio@gmail.com

² Instituto Politécnico da Universidade Katyavala Bwila. Departamento de Ensino e Investigação de Informática, Angola. makili_le@yahoo.com

Palavras-chave: Sistema de arquivo distribuído (DFS); Desempenho; Analise; Windows Server; NFS; SMB/CIFS.

ABSTRACT

Distributed file systems (DFS) are technological resources that enable secure and efficient information sharing and access within a network infrastructure. Since the adoption of these types of systems is vital for an institution, and considering the importance of choosing the right protocol for infrastructure reliability and performance, this study aims to comparatively analyze the operation and performance of Network File System (NFS) and Server Message Block (SMB)/Common Internet File System (CIFS) file systems integrated with the Windows Server DFS service, analyzing variables such as transfer rate, CPU (Central Processing Unit) utilization, and threads under different workloads. The test environment was designed based on the network infrastructure of Katyavala Bwila University, located in Benguela province, Angola, which uses a client-server architecture. Based on the results presented, the system's responsiveness was determined for different operations (write, rewrite, read, reread, and others) on files and records of different sizes. Similar performance was observed between the protocols in intensive load tests, with minor variations in throughput for specific operations. This information provides relevant information for network administrators, experts, and the scientific community in defining file sharing policies and choosing the most appropriate protocol for deployments in DFS environments.

Keywords: Distributed File System (DFS); Performance; Analysis; Windows Server; NFS; SMB/CIFS.

RESUMEN

Los sistemas de archivos distribuidos (DFS) son recursos tecnológicos que permiten compartir y acceder a información dentro de una infraestructura de red de forma segura y eficiente. A medida que la adopción de este tipo de sistemas es de vital importancia para una institución, y considerando la relevancia de elegir el protocolo adecuado para garantizar la confiabilidad y el rendimiento de la infraestructura, el presente estudio tiene como propósito analizar comparativamente el funcionamiento y desempeño de los sistemas de archivos Network File System (NFS) y Server Message Block (SMB)/Common Internet File System (CIFS) integrados al servicio DFS de Windows Server, analizando variables como la tasa de transferencia, la utilización de la CPU (Central Processing Unit) y los threads bajo diferentes cargas de trabajo. El diseño del entorno de prueba se basó en la infraestructura de red de la Universidade Katyavala Bwila, ubicada en la provincia de Benguela (Angola), sustentada en una arquitectura cliente-servidor. Con base en los resultados presentados, fue posible conocer la capacidad de respuesta del sistema de acuerdo con las distintas operaciones realizadas (escritura, reescritura, lectura, relectura y otras) en archivos y registros de diferentes tamaños, donde se indicó un rendimiento similar entre los protocolos en pruebas de carga intensiva, con pequeñas variaciones en la tasa de transferencia en operaciones específicas. Esta información proporciona aportes relevantes para administradores de redes, especialistas y la comunidad científica, en la definición de políticas de compartición de archivos y en la elección del protocolo más adecuado para implementaciones en entornos DFS.

Palabras clave: Sistema de archivos distribuido (DFS); Rendimiento; Análisis; Windows Server; NFS; SMB/CIFS.

INTRODUÇÃO

O acelerar da digitalização e informatização dos serviços faz com que muitas instituições públicas ou privadas recorram à implantação de aplicações (*web*, *móvel*, *desktop*) ou de uma infra-estrutura de rede e serviços que atendam às

suas necessidades. É nesta senda que vem à tona a importância ou necessidade da implantação dos *Distributed File System* (DFS) nas instituições com o objectivo de melhorar o sistema de compartilhamento e armazenamento de arquivos.

A necessidade de compartilhar recursos num sistema computacional surge devido à economia ou à natureza de algumas aplicações. Nesses casos, é necessário facilitar o compartilhamento de dispositivos de longo prazo e seus dados. Isso pode ser feito por meio de sistemas de arquivos distribuídos. “Um sistema de arquivos distribuído, permite que os usuários de computadores fisicamente distribuídos compartilhem dados e recursos de armazenamento, usando um sistema comum de arquivo” (Alangeh, 2021). Para Coulouris et al. (2013) um sistema de arquivos distribuído permite aos programas armazenarem e acederem arquivos remotos exactamente como se fossem locais, possibilitando que os usuários acessem a arquivos a partir de qualquer computador numa rede. O desempenho e a segurança no acesso aos arquivos armazenados num servidor devem ser comparáveis aos arquivos armazenados em discos locais.

Rani et al. (2014) afirmam que o principal objectivo de um sistema de arquivos distribuído é fornecer uma visão comum do sistema de arquivos centralizado, mesmo que tenha uma implementação distribuída, tal como se pode observar na figura 1.

Os DFS podem trazer algumas vantagens como:

- Permitir que vários usuários acessem ou armazenem dados;
- Permitir o compartilhamento remoto de dados;
- Melhorar a capacidade de alterar o tamanho dos dados e também aprimorar a capacidade de trocar dados;
- Mesmo se o servidor ou disco falhar, o sistema de arquivos distribuído pode fornecer transparência de dados;
- Melhorar a disponibilidade dos arquivos, o tempo de acesso e a eficiência da rede.
-

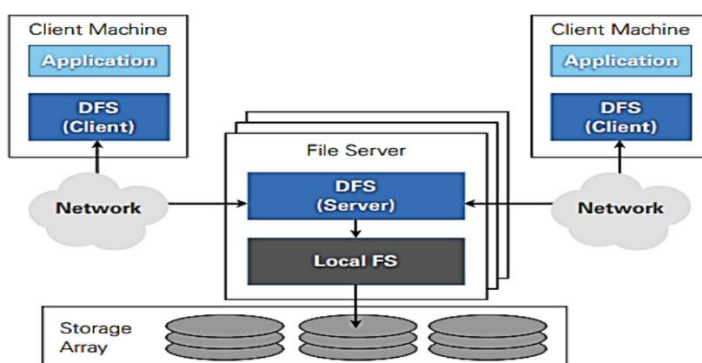


Figura 1 – Arquitectura genérica dos sistemas de arquivos distribuídos.

Fonte: Alangeh (2021).

Sie et al. (2019) afirmam que de acordo com a forma como o armazenamento de arquivos é gerido, existem dois conceitos de sistema de arquivos distribuídos: modelo cliente-servidor e o modelo ponto-a-ponto (P2P).

Existem vários tipos de sistemas de arquivos distribuídos, onde podemos destacar o clássico NFS e outros como SMB/CIFS, *Hadoop Distributed File System* (HDFS), *Andrew File System* (AFS), *Google File System* (GFS), GlusterFS e outros (Bžoch, 2012). Entre os mesmos, alguns são *OpenSource* e outros proprietários e vão permitindo que se implemente sistemas que possam dar um melhor tratamento aos arquivos. Seth (2023) diz que entre os inúmeros protocolos concebidos para este fim, o SMB/CIFS e o NFS são dois dos mais amplamente implementados.

Em particular, a infra-estrutura de rede em estudo utiliza os sistemas operativos da família Linux e Windows para a implementação dos serviços relacionados a gestão de arquivos sob a arquitectura cliente-servidor, neste âmbito o foco da investigação recairá aos sistemas de arquivos NFS e SMB/CIFS por serem os sistemas de partilha de arquivos predominantes nestes sistemas operativos.

Para atender às exigências da reestruturação da infra-estrutura de rede da instituição em estudo, ao crescimento do volume de dados, ao aumento do número de usuários e à necessidade de acesso rápido e confiável aos arquivos, torna-se essencial dispor de informações detalhadas obtidas por meio de testes ou análises de desempenho, informações que permitam responder à questão acerca de como se comportam os sistemas NFS e SMB/NFS quando submetidos a diferentes operações, sob distintas condições relacionadas à carga de trabalho, utilização da CPU e concorrência.

As possíveis respostas permitirão otimizar os recursos da rede e compreender o comportamento dos referidos sistemas, avaliando o seu desempenho diante de diferentes operações (escrita, reescrita, leitura, releitura, escrita e leitura aleatória) sob distintas cargas de trabalho, utilização da CPU (*Central Processing Unit*) e concorrência, tendo como referência a taxa de transferência.

Em função disso, propõe-se realizar uma análise comparativa entre os sistemas de arquivos NFS (versão 4.1) e SMB/CIFS (versão 3.1.1), a fim de auxiliar na definição de políticas de partilha de arquivos e na escolha do protocolo mais adequado para implantações em ambientes DFS.

Trabalhos relacionados

Para Dakic et al. (2024) os sistemas de arquivos assumem um papel crítico na optimização do desempenho, pois determina directamente a velocidade de leitura e escrita, alinhando-se às exigências específicas do ambiente em questão. Neste âmbito, Honwadkar (2011) realça que o uso de tecnologias como *Redundant Array of Independent Disks* (RAID), *Storage Area Networks* (SANs) e *Network Attached Storage* (NAS) podem tornar mais eficiente o processo de partilha, armazenamento e acessos aos arquivos.

O estudo realizado por Sie et al. (2019) apresenta uma implementação prática e análise de desempenho de um sistema de arquivos distribuído (DFS) usando o protocolo NFS em ambiente Windows, com servidor no *Windows Server* 2008 e cliente no *Windows Vista/7*, conectados por rede *ad hoc*. O *benchmarking*

IOzone forneceu dados quantitativos resultantes de um processo avaliativo sobre operações de leitura, escrita, reescrita e releitura em arquivos de 1 MB a 2 GB. O estudo identificou que o desempenho é melhor em arquivos menores, degrada-se com arquivos maiores devido a limitações de *cache* e pode ser otimizado com blocos de transferência de 32 KB. O trabalho apresenta um escopo restrito, apresentando abertura para um estudo que visa comparar o NFS com outros protocolos como SMB/CIFS, como é proposto no caso do presente trabalho. Já Lee et al. (2021) no seu estudo investigam e comparam vários sistemas de arquivos distribuídos, como *Ceph*, *GlusterFS*, *Luster* e *EOS* para ambientes com uso intensivo de dados, tendo como ponto de partida dois layouts: distribuído e RAIN (*Reliable Array of Independent Nodes*). O *layout* distribuído pode utilizar todas as capacidades do disco, mas não fornece cálculos de paridade, o que leva à perda de dados em caso de falha do disco, enquanto o RAIN não pode utilizar totalmente a capacidade do disco, mas os cálculos de paridade fornecem uma camada adicional de protecção.

Embora o *layout* distribuído, baseado em armazenamento horizontal, não forneça mecanismos de paridade, pode-se adoptar a replicação como estratégia complementar para reforçar a confiabilidade e a integridade dos dados. Por outro lado, Kumar (2019) advoga que a optimização de um sistema de arquivos para uma carga específica pode comprometer seu desempenho em cenários distintos, destacando a importância de parâmetros como configurações de rede, disco, número de servidores e clientes, e características da carga de trabalho, devido ao seu impacto crítico no desempenho final.

Zhao e Yuan (2014) propõem um modelo baseado no *Analytic Hierarchy Process* (AHP) para selecção dos factores de desempenho mais relevantes em sistemas de arquivos distribuídos. Entre os factores identificados como mais influentes estão: o número de servidores, o tipo de discos de armazenamento, o número de *threads* e o tipo de conexão de armazenamento. Embora o modelo tenha sido originalmente aplicado ao sistema de arquivos Lustre, os resultados obtidos podem ser adaptados para uma análise comparativa entre outros sistemas, como NFS e SMB/CIFS.

Seth (2023) em sua análise, realça que a escolha entre SMB e NFS para a partilha de ficheiros em ambientes Linux depende de uma série de factores, incluindo o desempenho, a segurança, a compatibilidade e as preferências administrativas. O SMB oferece um ambiente rico em funcionalidades, seguro e compatível com o Windows, que se destaca em redes com sistemas operativos mistos e cenários empresariais. O NFS, com o seu desempenho otimizado e integração nativa com Linux, é mais adequado para implementações Linux homogéneas e aplicações de alto rendimento.

Abordagem Técnica

A avaliação de sistemas como o DFS requer a consideração crítica do esquema de endereçamento e das especificações dos servidores, nomeadamente a memória de acesso aleatório (RAM), o processador e a capacidade de armazenamento (HDD). Estes componentes, cujas funções são descritas a seguir, revelam-se essenciais para a análise de desempenho, conforme demonstrado nas Tabelas I e II:

- Servidor 1 – neste servidor será instalado o Windows Server 2019 e a configuração dos serviços DNS (*Domain Name System*), DFS *Namespace* e o DFS *Replication*. Para a Microsoft Learn (2025a) o *namespaces* é um serviço que permite agrupar pastas compartilhadas localizadas em diferentes servidores em um ou mais *namespaces* estruturados logicamente. Isso torna possível dar aos usuários uma exibição virtual de pastas compartilhadas, onde um único caminho leva a arquivos localizados em vários servidores, juntamente com o serviço de *replication* vai permitir que a replicação aconteça.
- Servidores 2 e 3 – nestes servidores será instalado o *Windows Server* 2019 e configurado serviço DFS *Replication*, funcionarão de forma dedicada à replicação de dados, actuando como suporte para o servidor de *namespace*.
- PC 1, 2 e 3 – estes dispositivos actuarão como plataformas de teste, onde a ferramenta IOZone será configurada e utilizada para a execução dos testes. O IOzone é uma ferramenta de *benchmark* para sistemas de arquivos, permitindo testar o desempenho de E/S de arquivo para as seguintes operações: *read, write, re-read, re-write, read backwards, read strided, fread, fwrite, random read, pread, mmap, aio_read, aio_write*.

Tabela I – Softwares e endereçamento do ambiente de teste.

Dispositivo(serviço)	Softwares	IP	Subnet-mask	DNS
Servidor 1 (DNS e Namespace)	Windows Server 2019	10.10.0.1	255.255.255.0	10.10.0.1
Servidor 2 (Replica 1)	Windows Server 2019	10.10.0.2	255.255.255.0	10.10.0.1
Servidor 3 (Replica 2)	Windows Server 2019	10.10.0.3	255.255.255.0	10.10.0.1
PC 1 (Cliente)	Windows 10+IOZone	10.10.0.10	255.255.255.0	10.10.0.1
PC 2 (Cliente)	Windows 10+IOZone	10.10.0.11	255.255.255.0	10.10.0.1
PC 3 (Cliente)	Windows+IOZone	10.10.0.12	255.255.255.0	10.10.0.1

Tabela II - Características dos computadores que fazem parte do ambiente de testes.

Computador	Memória RAM	Processador	HDD
Servidor <i>Namespace</i>	8G	Core i5; 2.80 GHz; 6 núcleo; 6 processadores lógicos.	1TB
Servidor replica	8G	Core i5; 2.80 GHz; 6 núcleo; 6 processadores lógicos.	1TB
Cliente	8G	Core i5; 2.80 GHz; 6 núcleo; 6 processadores lógicos.	1TB

Para a avaliação do sistema proposto, foi necessário definir uma infra-estrutura de teste, baseada na arquitectura cliente-servidor e no modelo TCP/IP juntamente com os protocolos, técnicas, tipo de replicação e o mecanismo de armazenamento horizontal, isto é, utilizando a capacidade local dos servidores para armazenar os arquivos.

A replicação consiste em dois tipos de soluções, isto é, síncrona e assíncrona. Noor et al. (2019) afirmam que “a solução síncrona ou activa, actualizará duas réplicas ao mesmo tempo e reverterá se uma delas falhar. Alguns dos benefícios desse tipo de solução são alta disponibilidade, *failover* automático e perda mínima de dados” (p. 1299). Já para Dave e Raghuvanshi (2012) na replicação activa, cada solicitação do cliente é processada por todos os servidores. Isso requer que o processo seja determinístico. Determinístico significa que, dado o mesmo estado inicial e uma sequência de solicitação, todos os processos produzirão a mesma sequência de resposta e terminarão no mesmo estado final. Por este motivo adoptou-se para infra-estrutura de teste a replicação síncrona, aumentando assim o nível de tolerância a falhas do sistema.

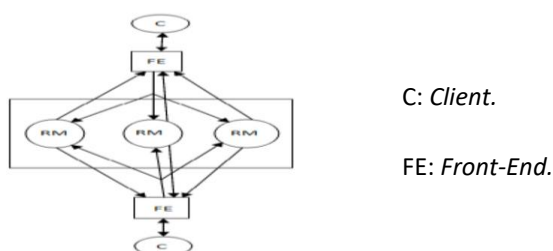


Figura 2 – Replicação activa.

Fonte: Dave & Raghuvanshi (2012, p. 128).

Por outra, os dispositivos envolvidos no processo de replicação, independentemente da forma ou solução adoptada, é necessário que os mesmos se comuniquem. Para que isto aconteça, a adopção de protocolos específicos é crucial. Para a configuração da infra-estrutura de teste foi utilizada o protocolo *Remote Differential Compression* (RDC); segundo a Microsoft Learn (2025b) os sistemas operativos de gestão de servidores da família *Windows* disponibilizam o serviço DFS que permite a criação de um espaço virtual de gestão e replicação de arquivos de forma distribuída como um mecanismo eficiente de mestres múltiplos, possibilitando a sincronização de pastas entre os servidores, utilizando o algoritmo ou protocolo replicação conhecido como RDC.

O RDC detecta alterações nos dados num arquivo e permite que a replicação de sistemas de arquivos distribuídos replique apenas os blocos de arquivos alterados em vez do arquivo inteiro, o que gera eficiência em pastas ou arquivos de tamanhos maiores. Conforme podemos observar na figura 3, que ilustra o modo de funcionamento entre um grupo de replicação, os membros do grupo e as pastas replicadas dentro de um DFS que utiliza o RDC.

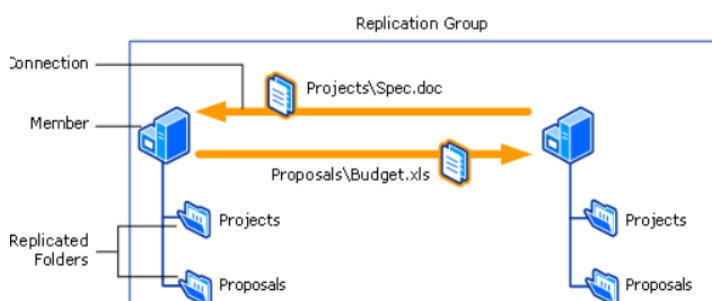


Figura 3 – Relação entre grupo de replicação, membros e pastas replicadas.
 Fonte: Microsoft Learn (2025b).

Conforme Noor et al. (2019), as técnicas de replicação têm sido implementadas com sucesso para sistemas de computação distribuída e permitem que tais sistemas permaneçam distribuídos, ao mesmo tempo em que aumenta sua disponibilidade e desempenho em grande escala, onde o sistema é capaz de operar na presença de falhas sem intervenção do usuário tolerar as falhas que possam ocorrer no ambiente de computação distribuída. O sistema proposto fará uso desta técnica *Two-Replica Distribution Technique* (TRDT) para melhor replicar os dados, permitindo ter dois nós réplicas com as mesmas capacidades do servidor *namespace*. Noor et al. (2019, p. 1301) dizem que para o TRDT funcionar “cada nó deve ter igual capacidade de armazenamento e todos os dados têm duas réplicas. Para N conjuntos de nós ($N = 2n$), onde cada conjunto consiste em dois nós”. Conforme pode-se observar na figura a seguir.

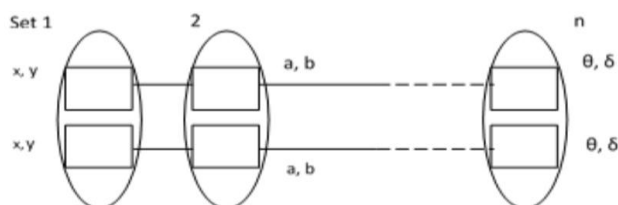


Figura 4 - Técnica de distribuição de réplica de dados quando $N=2n$.
 Fonte: Noor et al. (2019, p. 1301).

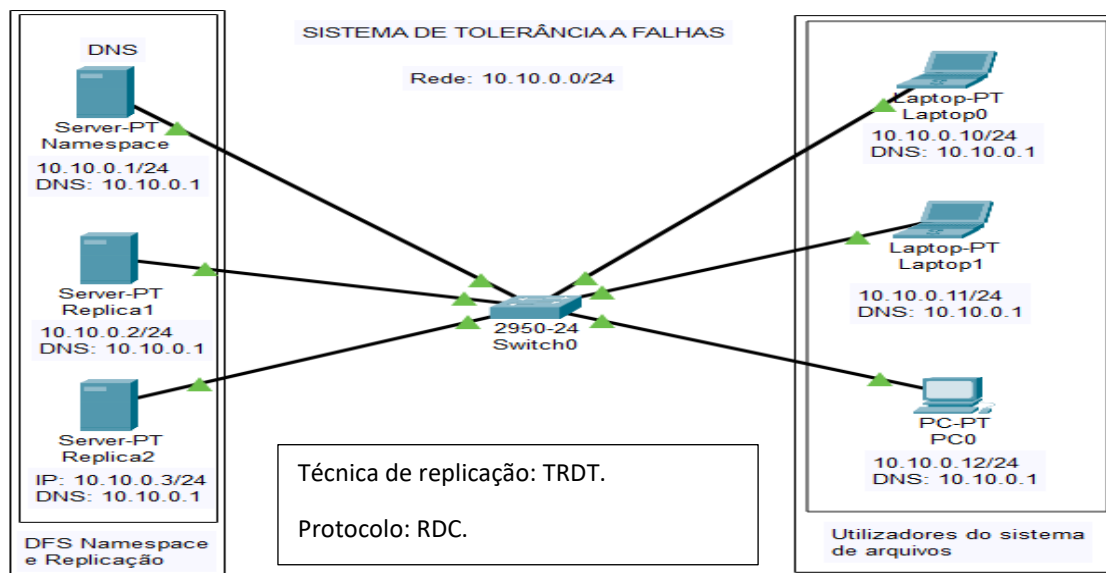


Figura 5 – Desenho do ambiente de teste.

Após a configuração do DFS, a figura a seguir mostra a estrutura criada no servidor *namespace* (`\\ukb.ed\dfs`) e as pastas partilhadas no seu interior (partilha-nfs e partilha-smb). Ainda nesta figura, observa-se também as pastas

de replicação que se encontram no servidor réplica, isto é, para as pastas partilha-nfs e partilha-smb.

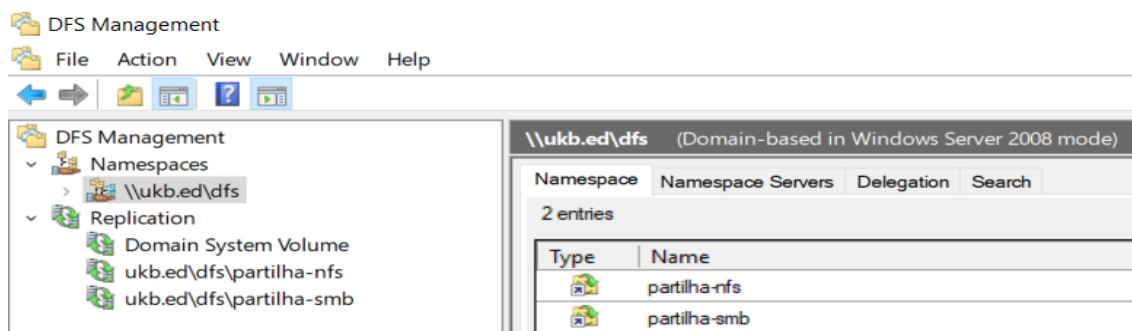


Figura 6 – Estrutura do DFS no servidor namespace.

Na figura 7 pode-se observar a estrutura do DFS no servidor réplica, que tem a função de funcionar como servidor secundário em caso de falha do servidor *master*/principal, contendo as pastas de replicação.

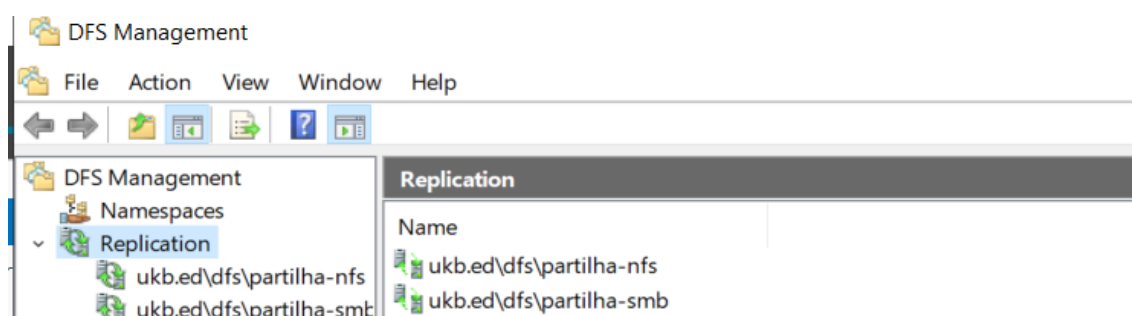


Figura 7 – Estrutura do DFS no servidor replica.

Metodologia de teste

Para analisar as figuras e as tabelas que se seguem como resultado dos testes realizados com o IOzone é importante entender três elementos, conforme Farncomb (2015):

- Tamanho do arquivo: este é simplesmente o tamanho total do arquivo que está sendo gravado ou lido do disco;
- Tamanho do registo: o arquivo é dividido usando-se um determinado tamanho de registos. Um arquivo de 64 MB pode ser composto de 8 registos de 8 MB ou 4 registos de 16 MB. Por isso, alguns dos gráficos são exibidos como completamente zero, alguns dos tamanhos de arquivo testados não são grandes o suficiente para todos os comprimentos de registo testados, por exemplo, não é possível ter um arquivo 512 KB criado a partir de um registo de 8 MB;
- Unidades de transferência de dados³: a velocidade com que um arquivo de determinado tamanho e tamanho de registo foi gravado ou lido a partir de disco, pode ser feita em KBps, MBps ou GBps ou simplesmente taxa de transferência.

O IOZone disponibiliza um conjunto de comandos que podem ser utilizados para medir e analisar o desempenho de diferentes sistemas de arquivos, ocultando

³ 1 KBps = 1.024 bytes (ou 8.192 bits). Um número baixo de KBps ou MBps (taxa de transferência) significa que a operação levará muito mais tempo para a sua efectivação.

alguns resultados tendo em conta o tamanho do arquivo definido durante o teste, mostrando apenas os dados de maior relevância.

Nas tabelas III e IV são apresentados os comandos utilizados durante o processo de testes, suas funções e também os comandos específicos utilizados para medir as variáveis específicas, de acordo aos três tipos de testes realizados, como a carga de trabalho, desempenho por número de *threads* e utilização da CPU, adoptando o KBps como unidade de taxa de transferência.

Os testes de carga e utilização da CPU foram conduzidos de acordo com os seguintes parâmetros: leitura e releitura (*read/re-read*), escrita e reescrita (*write/re-write*), leitura reversa (*read backwards*), regravação de registos (*re-write-record*) e leitura e escrita aleatórias (*random read/write*). Para o caso em estudo, destacou-se a avaliação do desempenho em operações de leitura e escrita que utilizam arquivos temporários de 64KB a 2 GB (eixo X) e, em cada processo de teste dos arquivos, os registos variaram de 4 KB a 16 MB. Para uma melhor análise dos dados classificou-se os arquivos de acordo ao tamanho:

- Arquivos pequenos (4KB – 512 KB);
- Arquivos médios (1MB – 16 MB);
- Arquivos grandes (32 MB – 512 MB);
- Arquivos muito grandes (>=1 GB).

Por outra, os testes de *threads*, foram definidos os seguintes parâmetros de análise: 4 processos, arquivos de 512 KB e registos de 4 KB.

Tabela III – Comandos seleccionados para os testes no IOZone.

Comandos	Descrição
- a	Utilizado para seleccionar o modo totalmente automático. Produzindo uma saída que cobre todas as operações de arquivo, testadas para tamanhos de registo de 4k a 16M para tamanhos de arquivo de 64k a 512M.
- i	Permite definir o tipo de teste a ser realizado, variando de 1 à 12 (0=write/re-write 'escrita/reescrita', 1=read/re-read 'leitura/releitura', 2=random-read/write 'leitura e escrita aleatória', 3=read-backwards 'leitura de trás para frente', 4=re-write-record 'leitura e escrita em um ponto específico', 5=stride-read 'ler um arquivo com um comportamento de acesso acelerado', 6=fwrite/re-fwrite 'utilização de buffer do usuário para escrita e reescrita', 7=fread/re-fread 'utilização de buffer do usuário para leitura e releitura', 8=random_mix 'leitura e gravação de um arquivo com acessos feitos a locais aleatórios dentro do arquivo', 9=pwrite/re-pwrite, 10=pread/re-pread, 11=pwritev/re-pwritev, 12=preadv/re-preadv)
- n	Permite definir o tamanho mínimo do arquivo (em Kbytes) para o modo automático, que vai de 64K a 512M.
- g	Permite definir o tamanho máximo do arquivo (em Kbytes) para o modo automático. Utilizado quando o computador tem mais 512 MB de RAM.
- r	Utilizado para especificar o tamanho do registo (4k a 16M), em Kbytes, para testar. Pode-se também especificar -r #k (tamanho em Kbytes) ou -r #m (tamanho em Mbytes) ou -r #g (tamanho em Gbytes)
- b	Permite criar um arquivo no formato de arquivo binário na saída de resultados compatível com Excel.
- R	Gera relatório Excel. Iozone irá gerar um relatório compatível com Excel para padronizar. Este arquivo pode ser importado com o Microsoft Excel (delimitado por espaço) e usado para criar um gráfico do desempenho do sistema de arquivos.
- f	Usado para especificar o nome do arquivo temporário em teste.
- F	Usado para especificar cada um dos nomes de arquivos temporários a serem usados no teste de taxa de transferência. O número de nomes deve ser igual ao número de processos ou encadeamentos especificados
- t	Permite ao usuário especificar quantos <i>threads</i> ou processos devem estar activos durante a medição.
- z	Usado em conjunto com -a para testar todos os tamanhos de registo possíveis.
- c	Inclui o close() nos cálculos de tempo. É utilizado se o close () do sistema operativo não estiver a funcionar durante o teste.

- e	Permite incluir flush (fsync, fflush) nos cálculos de tempo.
- w	Não desvincula os arquivos temporários quando terminar de usá-los. Deixe-os presentes no sistema de arquivos.
-U	Permite definir o ponto de montagem para desmontar e remontar entre os testes. IOzone desmontará e remontará este ponto de montagem antes de iniciar cada teste. Isso garante que o cache do buffer não contenha nenhum arquivo em teste.
-+ u	Exibirá a utilização da CPU para cada teste realizado.
-+ t	Activa o teste de desempenho de rede.
-+ m	Permite obter as informações de configuração dos clientes para teste de cluster.

Tabela IV - Métrica, sistemas de arquivo e comandos no IOZone.

Métricas	Sistema de arquivo	Dispositivo	Comandos (executados na linha de comandos)
Carga de trabalho	SMB/CIFS	Cliente	iozone -a -i 0 -i 1 -i 2 -i 3 -i 4 -g 2G -R -b resultados-smb.xls \\ukb.ed\dfs\partilha-smb
	NFS	Cliente	iozone -a -i 0 -i 1 -i 2 -i 3 -i 4 -g 2G -R -b resultados-nfs.xls \\ukb.ed\dfs\partilha-nfs
Desempenho, especificando o número de processos ou <i>threads</i>	SMB/CIFS	Cliente	iozone -t 4 -i 0 -i 1 -i 2 -i 3 -i 4 -R -b resultados-desempenho-smb.xls \\ukb.ed\dfs\partilha-smb
	NFS	Cliente	iozone -t 4 -i 0 -i 1 -i 2 -i 3 -i 4 -R -b resultados-desempenho-nfs.xls \\ukb.ed\dfs\partilha-nfs
Utilização da CPU	SMB/CIFS	Cliente	iozone -a -+u -i 0 -i 1 -i 2 -i 3 -i 4 -g 2G -R -b resultados-cpu-smb.xls \\ukb.ed\dfs\partilha-smb
	NFS	Cliente	iozone -a -+u -i 0 -i 1 -i 2 -i 3 -i 4 -g 2G -R -b resultados-cpu-nfs.xls \\ukb.ed\dfs\partilha-nfs

Resultados e discussão

Testes de carga

Conforme pode ser observado na Figura 8, que ilustra a aplicação do comando específico para o teste e a utilização da memória *cache*.

```
Run began: Wed Mar 20 10:35:34 2024
Auto Mode
CPU utilization Resolution = -0.000 seconds.
CPU utilization Excel chart enabled
Using maximum file size of 2097152 kilobytes.
Excel chart generation enabled
Command line used: iozone -a -+u -i 0 -i 1 -i 2 -i 3 -i 4 -g 2G -R -b resultados-cpu-nfs.xls \\ukb.ed\dfs\partilha-nfs
Output is in kBytes/sec
Time Resolution = -0.000000 seconds.
Processor cache size set to 1024 kBytes.
Processor cache line size set to 32 bytes.
File stride size set to 17 * record size.
```

Figura 8 - Parâmetros para os testes de carga e de utilização da CPU no IOZone.

A análise dos dados de desempenho das figuras abaixo (9 e 10) revela diferenças significativas entre os protocolos NFS e SMB/CIFS em operações de escrita, variando conforme o tamanho do arquivo e o tamanho do registo.

Para arquivos pequenos o NFS apresenta um desempenho relativamente estável, mas varia bastante entre tamanhos de registo. Para arquivos de 64 KB e registo de 64 KB, obtém até 477.968 KBps, mas em 256 KB já ultrapassa

1,4.10⁶ KBps, chegando a 2,65.10⁶ KBps em arquivo de 512 KB e registo de 256 KB. O SMB/CIFS apresenta um desempenho mais irregular, com quedas significativas em arquivos de 64 KB e registo de 4 KB, chega a 254.270 KBps, mas em 128 KB há oscilações (máx. ~1,17.10⁶ KBps) em todos os arquivos pequenos, por outra, destacam-se também alguns picos interessantes em registos de 256 KB (1,82.10⁶ KBps) e 512 KB (2,45.10⁶ KBps), mas menos consistentes que o NFS. Para arquivos pequenos, o NFS é mais consistente e escalável, enquanto SMB/CIFS tem variações bruscas, com bons picos, mas menor previsibilidade, isto porque tem *overhead* maior por operação, o que explica alguma instabilidade com registos pequenos, mas em alguns casos supera NFS.

Em arquivos médios o NFS apresenta valores geralmente altos e estáveis, picos próximos a 4,6.10⁶ Kbps (arquivo de 16 MB e registo de 1 MB) e desempenho muito bom em registos de 64 KB a 1 MB. O SMB mostra bom desempenho em registos de 1 a 16 MB, com valores na casa de 3,3 - 4,9.10⁶ KBps (4,8.10⁶ Kbps em arquivo de 16 MB e registo de 1 MB), apesar de haver uma queda em registo de 8 MB (2.476.795 KBps em arquivo de 16 MB e registo de 8 MB vs NFS 4.026.443 KBps). Para arquivos médios, SMB/CIFS tende a superar NFS nos melhores casos, oferecendo maior *throughput* máximo, mas NFS apresenta menor variabilidade e melhor estabilidade.

Em arquivos grandes o NFS apresenta um desempenho sólido, com valores entre 3,3 e 4,6.10⁶ KBps, mantendo uma boa regularidade e não apresenta quedas bruscas. Já o SMB/CIFS atinge patamares semelhantes ou ligeiramente superiores em alguns pontos (4,74.10⁶ KBps em arquivo 512 MB e registo de 2 MB). Neste caso, ambos os protocolos oferecem desempenho próximo, mas SMB/CIFS tem leve vantagem em *throughput* máximo, enquanto NFS é mais estável.

Em arquivos muito grandes ambos sofrem grande degradação, ou seja, uma queda abrupta no desempenho. O NFS sofre maior impacto à medida que os arquivos crescem, mas o SMB/CIFS tende a ser levemente superior, mantendo taxas de escrita consistentemente acima das do NFS, especialmente a partir de 2 GB. Esta queda pode estar associada a limitações de *cache* e *Buffers*, *overhead* de protocolo ou limitação estrutural do sistema de arquivos e da rede.

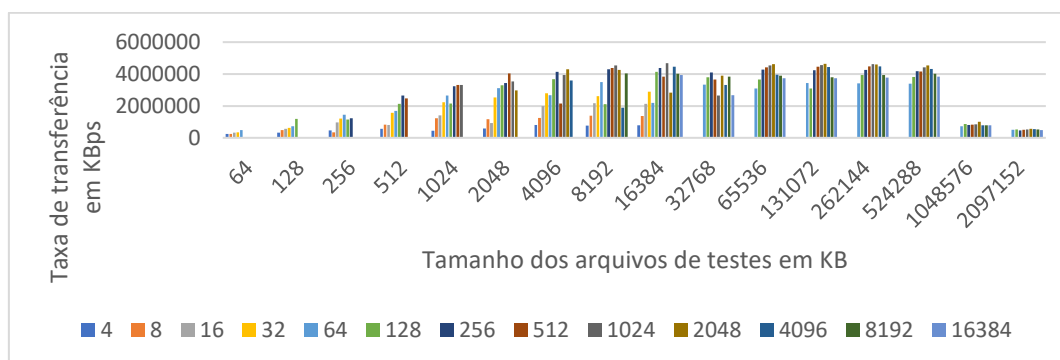


Figura 9 – Relatório da operação de escrita em arquivos NFS.

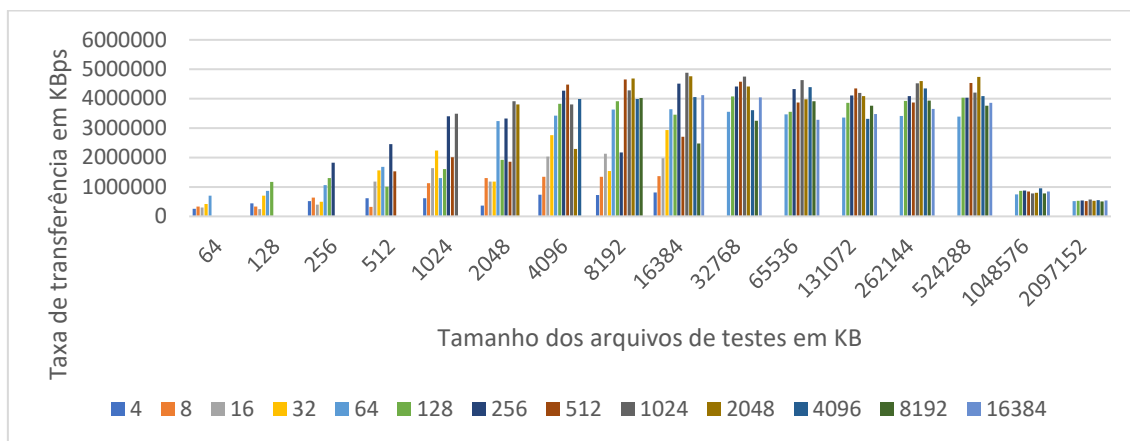


Figura 10 – Relatório da operação de escrita em arquivos SMB/CIFS.

Analisando as figuras abaixo (11 e 12), pode-se referir que para arquivos pequenos, o NFS apresentou desempenho inicial estável e crescente, destacando-se pontualmente em arquivo de 64 KB com registo de 64 KB (7.191.011 KBps), mas com flutuações e quedas acentuadas à medida que os tamanhos de registo aumentam, especialmente em arquivos 256 KB, onde o valor cai para 4.547.069 KBps (em registo de 64 KB). O SMB/CIFS, por sua vez, registou valores mais altos em registos menores (4 KB e 8 KB) e manteve uma curva de desempenho mais linear na faixa de 128 KB a 512 KB, superando o NFS em diversos pontos, como na leitura completa de um arquivo de 128 KB com registo de 128 KB, demonstrando maior consistência. Dessa forma, para cargas de trabalho com arquivos pequenos e acesso sequencial, os dados sugerem que o SMB/CIFS pode ser a escolha mais eficiente. O NFS, por outro lado, mostra vantagem em cenários específicos, como leituras em blocos de 64 KB.

Em arquivos médios o NFS alcançou valores elevados em registos pequenos, como em arquivos de 1 MB com registo de 64 KB (8.172.386 KBps), mas apresenta maior variabilidade à medida que o tamanho do arquivo aumenta, com quedas perceptíveis, por exemplo, em arquivo de 2 MB com registo de 256 KB (4.121.553 KBps). O SMB/CIFS oferece desempenho mais estável e previsível, em arquivo de 2 MB com registo de 128 KB, atingiu 8.774.635 KBps, e sustentou valores consistentes até o arquivo de 16 MB, o que o torna mais adequado em cenários de carga contínua.

Para arquivos grandes o NFS apresenta instabilidade em arquivos de 32 - 64 MB, com desempenho reduzido em arquivo 32 MB com registo de 128 KB (4.848.197 KBps), mas gradualmente estabilizou em volumes maiores, convergindo para valores próximos ao SMB/CIFS. Por outra, o SMB/CIFS teve vantagem em arquivos de 32 MB com registo de 128 KB (5.308.191 KBps) e manteve ligeira superioridade em arquivo de 64 MB. No entanto, entre arquivos maiores (128 MB e 512 MB), ambos protocolos apresentam desempenho equivalente, por exemplo, em arquivo de 128 MB com registo 128 KB, o NFS atingiu 5.308.600 KBps contra 5.466.549 KBps no SMB. O SMB/CIFS oferece melhor equilíbrio entre performance e estabilidade para arquivos grandes, enquanto NFS requer configuração mais cuidadosa, mas pode atingir um desempenho similar.

Para arquivos muito grandes, o desempenho de ambos os protocolos convergem e se torna praticamente equivalente, com pequenas alternâncias de vantagem. Uma análise detalhada mostra que as vantagens são mínimas e alternadas enquanto o SMB/CIFS registra valores ligeiramente superiores em alguns cenários (arquivo de 1 GB com registro de 1 MB), outros casos observam-se em arquivo de 1 GB com registro de 128 KB, o SMB/CIFS registou 5.853.721 KBps, enquanto o NFS obteve 5.769.185 KBps. Já em arquivo de 2 GB com registro de 128 KB, o NFS alcançou 5.865.147 KBps, contra 5.981.038 KBps do SMB. o NFS mantém paridade ou até pequenas vantagens em outros cenários como em arquivos de 1 GB com registro de 4 MB, (NFS obteve 6.448.775 KBps enquanto o SMB/CIFS teve 6.437.136 KBps). Essas diferenças são marginais, confirmando equivalência no manuseio de arquivos muito grandes.

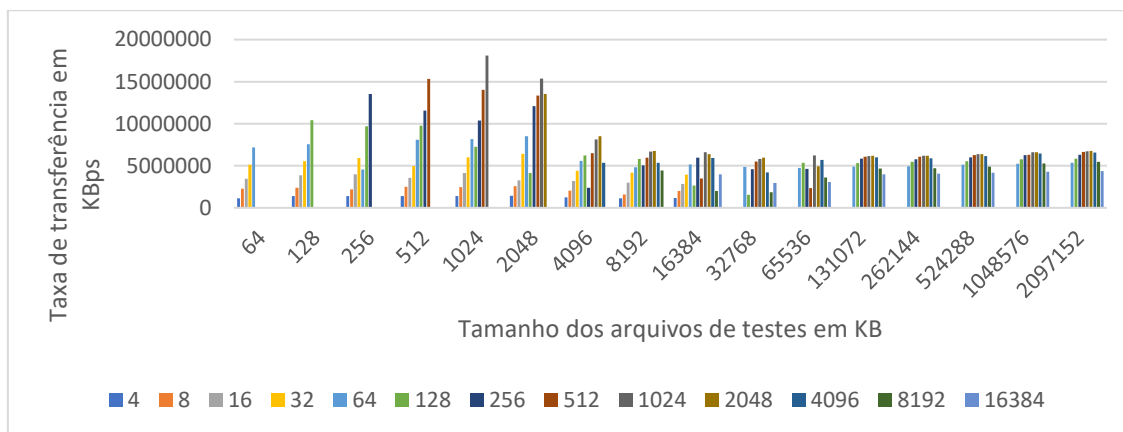


Figura 11 – Relatório da operação de leitura em arquivos NFS.

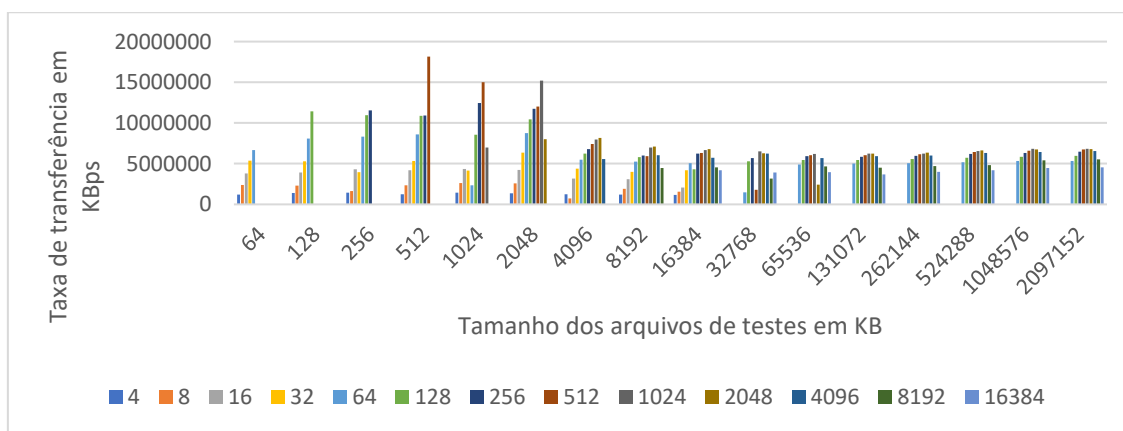


Figura 12 – Relatório da operação de leitura em arquivos SMB/CIFS.

Testes de *threads*

```
Output is in kBytes/sec
Time Resolution = -0.000000 seconds.
Processor cache size set to 1024 kBytes.
Processor cache line size set to 32 bytes.
File stride size set to 17 * record size.
Throughput test with 4 processes
Each process writes a 512 kByte file in 4 kByte records
```

Figura 13 – Parâmetros para os testes de desempenho no IOZone.

A figura 16 é consolidada os dados apresentados nas figuras 14 e 15, permitindo uma análise mais clara, na qual observa-se que o SMB/CIFS tende a ser mais robusto em operações de escrita e acessos aleatórios, o que o torna vantajoso em ambientes de aplicações transacionais (bases de dados, sistemas de arquivos compartilhados com múltiplos clientes escrevendo e lendo blocos aleatórios), já o NFS demonstra maior eficiência em leituras repetidas e acessos sequenciais reversos, o que pode ser vantajoso em cenários de leitura intensiva, como HPC (*High Performance Computing*), onde grandes volumes de dados são lidos em padrões previsíveis ou repetitivos.

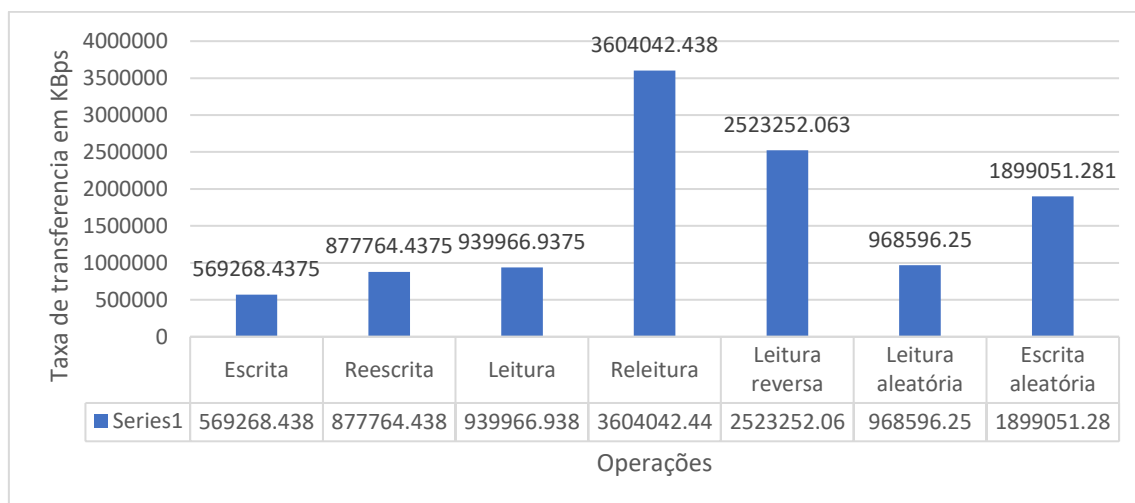


Figura 14 - Relatório dos testes de desempenho em arquivos NFS utilizando 4 processos concorrentes.

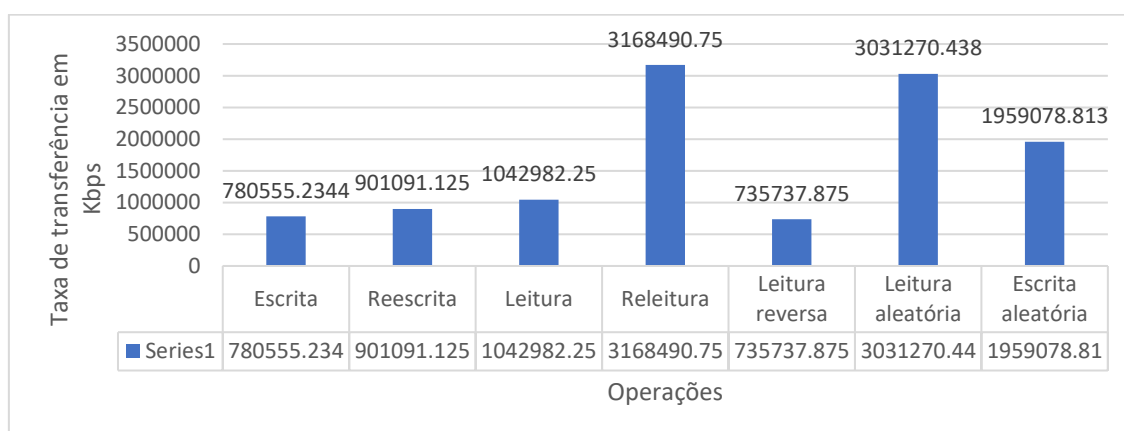


Figura 15 - Relatório dos testes de desempenho em arquivos SMB/CIFS utilizando 4 processos concorrentes.

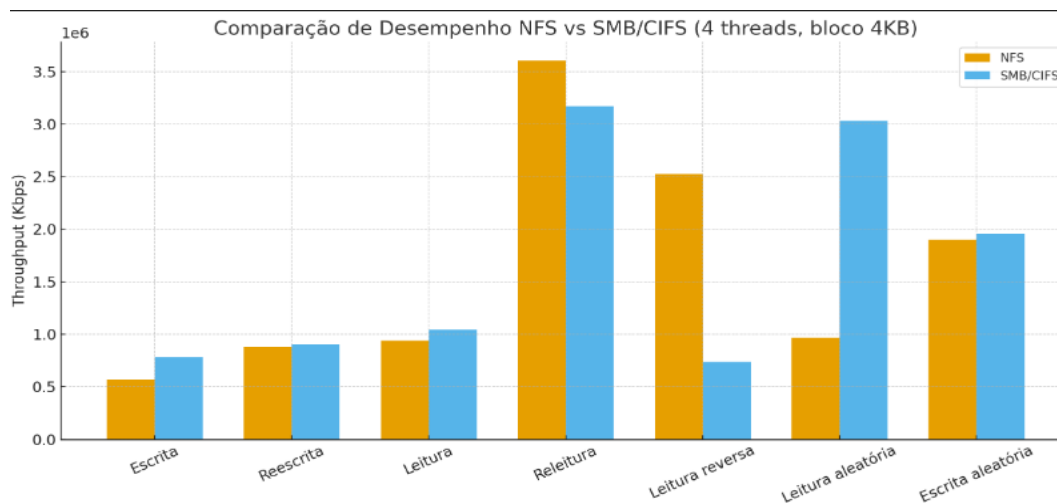


Figura 16 – Comparação entre NFS e SMB/CIFS.

Utilização da CPU

Em operações de escrita, o protocolo SMB/CIFS exibe uma média global de desempenho ligeiramente superior, embora a sobreposição nos desvios padrão sugira eficiência comparável ao NFS. O NFS destaca-se no processamento de arquivos maiores, com menor sobrecarga relativa, enquanto o SMB/CIFS apresenta melhor desempenho com arquivos de tamanho médio.

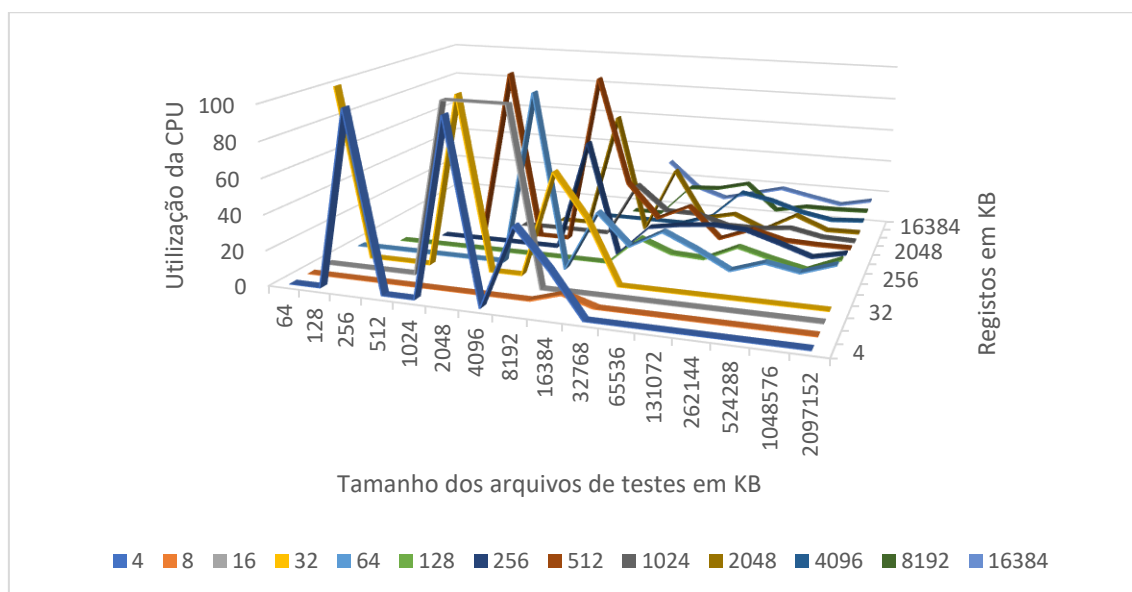


Figura 17 - Relatório de utilização da CPU, operação de escrita em arquivos NFS.

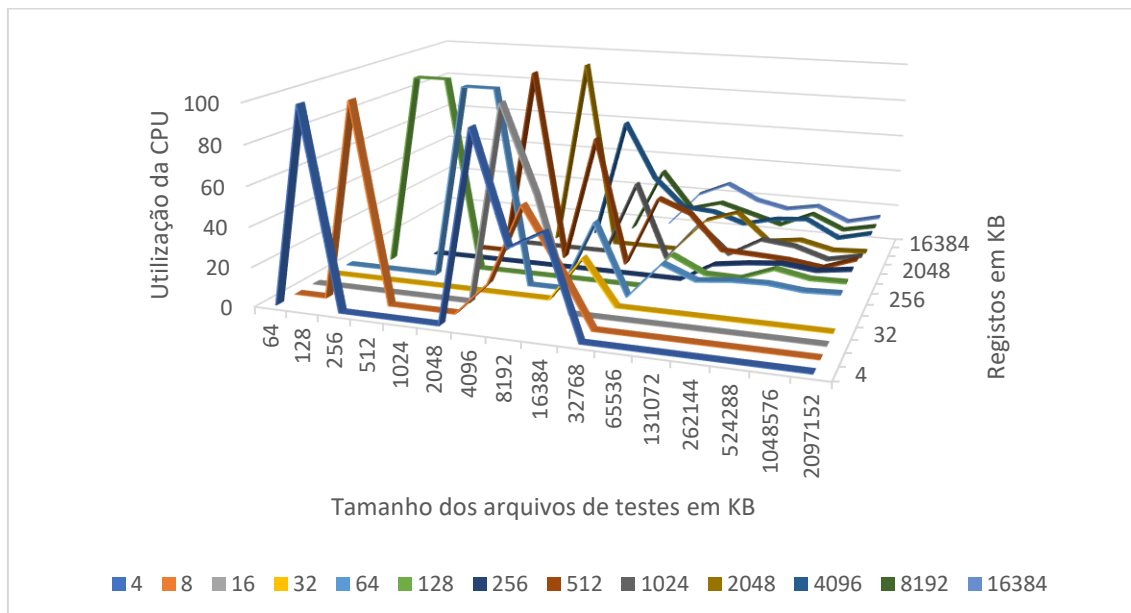


Figura 18 - Relatório de utilização da CPU, operação de escrita em arquivos SMB/CIFS.

Para operações de leitura, o SMB/CIFS apresenta geralmente uma carga média ligeiramente superior ao NFS (91,96% vs. 89,18% no geral), particularmente em ficheiros médios (99,77% vs. 92,27%) e grandes (84,67% vs. 81,96%), o que reflete maior irregularidade e sobrecarga pontual, provavelmente devido à sobrecarga do protocolo. As diferenças são pequenas (<5% na maioria dos casos), e o NFS tem um desempenho comparável ou superior em ficheiros muito grandes e determinados tamanhos de bloco (por exemplo, 64 KB e 512 KB). Ambos os protocolos se aproximam da saturação da CPU (95-100%) para ficheiros >32 MB, indicando um estrangulamento partilhado na taxa de transferência de dados ou no *buffer*.

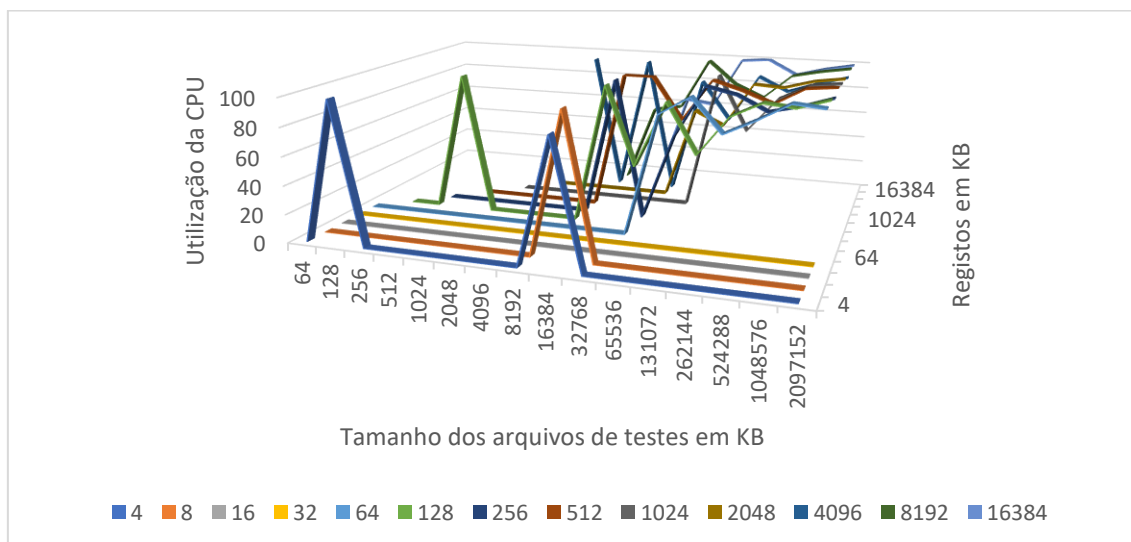


Figura 19 - Relatório de utilização da CPU, operação de leitura em arquivos NFS.

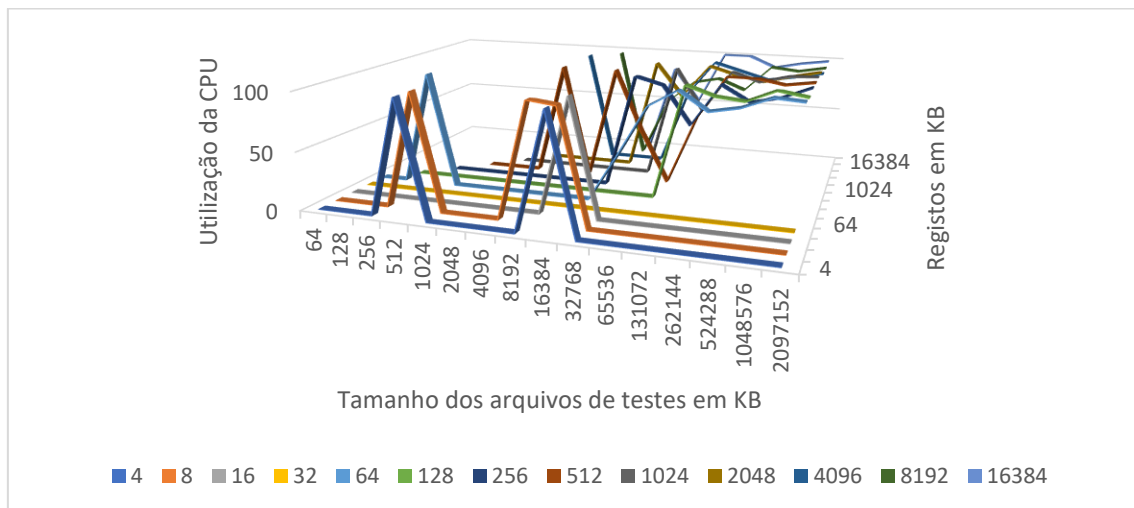


Figura 20 - Relatório de utilização da CPU, operação de leitura em arquivos SMB/CIFS.

Conclusão

Portanto, depois de uma análise exaustiva dos dados de testes, pode-se observar que para o processo de escrita o NFS mostra consistência e eficiência em arquivos pequenos a médios, isso sugere melhor adequação em ambientes de backup incremental, sistemas de *logs* e aplicações de alta fragmentação. O SMB/CIFS apresenta melhor desempenho em cargas médias e grandes, atingindo picos mais altos de *throughput*, essa característica o torna mais apropriado em ambientes voltados para transferência de grandes volumes de dados (multimídia, máquinas virtuais).

Quanto a leitura, os resultados mostram que para aplicações práticas, a escolha deve considerar o padrão de acesso, o tamanho predominante dos arquivos e a infra-estrutura de rede. O SMB/CIFS tende a oferecer maior equilíbrio geral, particularmente mais eficiente para ficheiros grandes e muito grandes e para tamanhos de registo maiores, sendo a escolha preferencial para cenários que envolvam ficheiros ou registos maiores, ideal para partilha arquivos em empresas, indústria da mídia e entretenimento, *big data*. enquanto o NFS pode ser otimizado para obter desempenho competitivo em cenários específicos, especialmente para tamanhos de registo mais pequenos e determinados tamanhos de ficheiro médios, mas é menos consistente no geral, ideal para servidores *web*, hospedagem de aplicações, virtualização.

Em testes de desempenho com 4 processos, o SMB/CIFS mostrou ser mais indicado para cenários com escritas intensivas e acessos aleatórios e o NFS para cenários com leituras repetitivas e padrões não lineares previsíveis, embora a escolha entre NFS e SMB deve, portanto, considerar o perfil da carga de trabalho, como em ambientes de colaboração, servidores de aplicações, virtualização, bases de dados e escritórios (SMB/CIFS) e ambientes científicos, HPC e *big data* orientado a leitura (NFS).

Quanto ao uso da CPU, o NFS e o SMB/CIFS exibem desempenho similar no consumo de CPU para operações de escrita, sem um protocolo claramente superior. Para arquivos pequenos e médios, ambos apresentam alta utilização

de CPU, sendo recomendável otimizar o desempenho com registos de dados maiores ou armazenamento local. Em contrapartida, para arquivos grandes ou muito grandes, ambos se mostram eficientes, com consumo de CPU variando entre 10% e 20%. Já o NFS destaca-se pela eficiência em ambientes de alto volume com arquivos grandes e muito grandes, reduzindo a carga da CPU, enquanto o SMB/CIFS pode ser mais adequado em cenários que priorizam compatibilidade em detrimento da utilização mínima da CPU. Para melhor otimização, recomenda-se ajustar os tamanhos dos registos para ≥ 64 KB, de forma a estabilizar a utilização e reduzir a variabilidade.

Apesar da relevância dos achados para administradores de redes em ambientes Windows, para implantações em larga escala, ambientes heterogêneos (Linux e Windows) ou carga de trabalho críticas e reais (VDI, backup, bancos de dados, virtualização, HPC), recomenda-se realizar novos testes com clientes nativos, configurações otimizadas (NFSv4.2, SMB *Multichannel*, RDMA), armazenamento de alto desempenho (SSD/NVMe, RAID) e métricas complementares (latência, IOPS, consistência).

Este trabalho oferece uma base sólida para tomada de decisão em ambientes DFS do Windows Server, mas reforça a necessidade de avaliações contextualizadas, considerando o perfil específico de uso, a infra-estrutura disponível e os requisitos de desempenho da instituição.

REFERÊNCIAS

- Alangeh, L. N. (2021). *Distributed file systems: A literature review* [Unpublished manuscript]. Department of Computer Engineering, Institute of Natural and Applied Sciences, Gazi University.
<https://www.researchgate.net/publication/355175965>
- Anderson, T. E., Peter, S., Canini, M., Kim, J., Kostić, D., Kwon, Y., Reda, W., Schuh, H. N., & Witchel, E. (2020). Performance and Availability via Client-local NVM in a Distributed File System. *14th USENIX Symposium on Operating Systems Design and Implementation*, 1011–1027.
<https://www.usenix.org/conference/osdi20/presentation/anderson>
- Bžoch, P. (2012). Distributed file systems: The state of the art and concept of Ph.D. thesis (Technical Report No. DCSE/TR-2012-02). University of West Bohemia, Department of Computer Science and Engineering.
<http://www.kiv.zcu.cz/publications/>
- Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2013). *Sistemas Distribuídos: Conceitos e projeto*. Bookman.
- Dave, S., & Raghuvanshi, A. (2012). Fault Tolerance Techniques in Distributed System. *International Journal of Engineering Innovation & Research*, 1(2), 2277–5668.
- Dakic, V., Kovac, M., & Videc, I. (2024). High-Performance Computing Storage Performance and Design Patterns—Btrfs and ZFS Performance for Different Use Cases. *Computers*, 13(6), 139. <https://doi.org/10.3390/computers13060139>
- Farncomb, J. (2015, 22 de Julho). *Windows NFS vs Linux NFS Performance Comparison*. RootUsers. <https://www.rootusers.com/windows-nfs-vs-linux-nfs-performance-comparison/>

- Huang, H. H., & Grimshaw, A. S. (2011). Design, implementation and evaluation of a virtual storage system. *Concurrency and Computation: Practice and Experience*, 23(4), 311–331. <https://doi.org/10.1002/cpe.1644>
- Honwadkar, K. N., & Sontakke, T. R. (2011). Improvement in capacity and efficiency of network storage by configuring hard disk drives on nodes of a LAN. *International Journal of Computer Applications*, 19(3), 15–21. <https://doi.org/10.5120/2338-3048>
- Kumar, S. (2019). *Performance modeling of a distributed file-system*. <https://doi.org/10.48550/arXiv.1908.10036>
- Lee, J. Y., Kim, M. H., Shah, S. A. R., Ahn, S. U., Yoon, H., & Noh, S. Y. (2021). Performance evaluations of distributed file systems for scientific big data in fuse environment. *Electronics (Switzerland)*, 10(12), 1471. <https://doi.org/10.3390/electronics10121471>
- Microsoft Learn. (2025a, 16 de Agosto). *Visão geral de Namespaces DFS*. Learn. <https://learn.microsoft.com/pt-br/windows-server/storage/dfs-namespaces/dfs-overview?tabs=server-manager>
- Microsoft Learn. (2025b, 16 de Agosto). *Visão geral sobre a Replicação do DFS*. Learn. <https://learn.microsoft.com/pt-br/windows-server/storage/dfs-replication/dfs-overview?source=recommendations>
- Noor, A. S. M., Zian, N. F. M., & Bahri, F. N. M. S. (2019). Survey on replication techniques for distributed system. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(2), 1298. <https://doi.org/10.11591/ijece.v9i2.pp1298-1303>
- Sie, S., Kyaw, T., Moh, D., & Khaing, M. (2019). Analysis and optimization of distributed file system performance. *IJCIRAS*, 2(3), 39–44. <http://www.ijciras.com/PublishedPaper/IJCIRAS1317.pdf>
- Seth, V. (2023). Analyzing and comparing the performance of SMB and NFS protocols for efficient file sharing in Linux environments. *International Journal of Scientific Research & Engineering Trends*, 9(1), 1–4. https://ijsret.com/wp-content/uploads/IJSRET_V9_issue1_138.pdf
- Zhao, T., & Yuan, H. (2014). Parameter Analysis Model of Distributed File Based on Improved Analytic Hierarchy Process. *Asian Network for Cientific Information*, 13(11), 1908–1911. <https://doi.org/10.3923/itj.2014.1908.1911>