

# Uma Abordagem do Uso de Sistemas de Arquivos em Computação de Alto Desempenho (HPC)

Ilan Grinspun<sup>1</sup>, Evaldo Costa<sup>1</sup>

<sup>1</sup>Programa de Pós-graduação – Mestrado Profissional em Engenharia de Software (MPES)  
CESAR School – Recife – PE – Brasil

{ig,ebc}@cesar.school

***Resumo.** O uso de soluções de computação de alto desempenho (HPC) tem se expandido para além do contexto tradicional de pesquisa científica e engenharia, passando a sustentar aplicações intensivas em dados em múltiplos domínios. Nesse cenário, o subsistema de armazenamento torna-se decisivo para o desempenho global, a escalabilidade e a disponibilidade dos dados. Este trabalho discute o papel de três arquiteturas de sistemas de arquivos em HPC: Network File System (NFS), Distributed File Systems (DFS) e Parallel File Systems (PFS).*

## 1. Introdução

O uso de soluções de computação de alto desempenho (HPC) vem se tornando cada vez mais utilizada em diferentes áreas de negócios, deixando de estar restrito à pesquisa científica e ao desenvolvimento tecnológico [Silva et al. 2022].

Um ambiente de HPC é composto por nós de processamento, interconexões de alta velocidade, camadas hierárquicas de armazenamento e software de gerenciamento de recursos e dados. Entre esses componentes, o sistema de arquivos ocupa papel central, pois define como os dados são organizados, armazenados, acessados e compartilhados entre múltiplos processos concorrentes. A escolha dessa camada impacta diretamente a latência, a largura de banda, a escalabilidade, a tolerância a falhas e o custo operacional da infraestrutura.

Diante desse contexto, compreender as diferenças entre NFS, DFS e PFS deixou de ser apenas um exercício conceitual e passou a ser requisito para decisões arquiteturais em ambientes acadêmicos e corporativos. Este artigo apresenta uma visão comparativa dessas arquiteturas e propõe um encaminhamento prático para a continuidade da pesquisa.

## 2. Trabalhos Relacionados

A literatura recente mostra que os desafios de armazenamento em HPC não se resumem à capacidade, mas envolvem também padrões de acesso, semântica de consistência, uso intensivo de metadados e mecanismos de tolerância a falhas. O editorial de Antelmi [Antelmi and Carlini 2026] destaca que gestão de dados, sistemas de armazenamento e otimização de workflows permanecem gargalos centrais em ambientes HPC altamente distribuídos, especialmente quando há forte movimentação de dados e pressão de I/O.

No contexto de sistemas distribuídos, Maisto et al. [Maisto et al. 2026] discutem o custo dos mecanismos de redundância em DFS. Os autores observam que a replicação tripla, comum em HDFS, eleva substancialmente o overhead de armazenamento e energia, enquanto as técnicas de Erasure Coding reduzem esse custo, embora acrescentem uma

carga computacional adicional ao cluster. Como resposta, propõem uma arquitetura heterogênea com offloading em FPGA para acelerar o cálculo de erasure codes e preservar a escalabilidade do sistema. Esse resultado é importante porque evidencia que, em DFS, a discussão arquitetural não pode ignorar o custo computacional da resiliência.

Por outro lado, Chen et al. [Chen et al. 2024] investigam o problema sob a ótica dos PFS e mostram que workloads modernos de HPC impõem exigências específicas, com compartilhamento de arquivos entre aplicações, predominância de arquivos pequenos, operações intensivas de metadados e requisitos semânticos frequentemente mais relaxados que os do modelo POSIX completo. Os autores relatam que o PeakFS alcança desempenho de metadados e dados superior a 90% do limite de hardware em seu ambiente experimental, além de ganhos expressivos em operações como criação e consulta de arquivos pequenos. O estudo reforça que o desenho de um PFS deve ser orientado pelo perfil real da carga de trabalho, e não apenas por throughput bruto.

Em conjunto, esses trabalhos mostram um amadurecimento da literatura em duas frentes: otimizações específicas para DFS, com foco em resiliência e eficiência computacional, e otimizações para PFS, com foco em acesso paralelo e metadados. Entretanto, permanece pouco explorada, em textos introdutórios e de apoio à decisão arquitetural, uma comparação sintética entre NFS, DFS e PFS que conecte características de workload, infraestrutura e estratégia experimental. Essa é a lacuna que este trabalho busca preencher: oferecer uma visão comparativa orientada ao uso e, ao mesmo tempo, apontar um caminho prático para a continuidade do estudo.

### **3. Sistemas de Arquivos para HPC**

De forma geral, destacam-se 3 categorias principais:

- Sistema de arquivos de rede (NFS);
- Sistema de arquivos distribuído (DFS);
- Sistema de arquivos paralelo (PFS).

Para definir qual destas diferentes arquiteturas de sistemas de arquivos será utilizada em um ambiente HPC, deve-se levar em consideração quais aplicações serão executadas e os tipos de dados que serão processados.

#### **3.1. Sistema de Arquivos de Rede (NFS)**

O Network File System (NFS) é uma das tecnologias mais tradicionais de compartilhamento de arquivos em rede. Seu modelo cliente-servidor simplifica a administração e o torna adequado para diretórios de usuários, arquivos de configuração e compartilhamento de software em clusters menores [Wang et al. 2021].

Apesar de sua simplicidade, o NFS tende a se tornar gargalo quando submetido a cargas massivamente paralelas, com grande número de clientes acessando simultaneamente os mesmos arquivos ou diretórios. Por isso, sua utilização como camada principal de dados em HPC costuma ser inadequada; em geral, o NFS permanece mais apropriado como solução auxiliar, e não como backend principal de aplicações intensivas em I/O.

### 3.2. Sistema de Arquivos Distribuído (DFS)

Um sistema de arquivos distribuído (DFS) particiona dados em blocos e os distribui por múltiplos servidores, oferecendo escalabilidade horizontal e mecanismos de redundância. Exemplos amplamente conhecidos incluem HDFS, Ceph e GlusterFS.

A principal vantagem do DFS está em sua boa aderência a ecossistemas de Big Data, analytics e processamento distribuído de grandes volumes de informação. Além disso, o modelo favorece disponibilidade e tolerância a falhas. Contudo, essa robustez tem custo: Maisto et al. [Maisto et al. 2026] mostram que mecanismos de redundância, especialmente quando baseados em replicação, podem ampliar o consumo de armazenamento e energia, enquanto alternativas baseadas em erasure coding exigem maior poder computacional. Em consequência, a avaliação de um DFS para HPC deve considerar não apenas capacidade e escalabilidade, mas também o custo de resiliência imposto ao cluster.

### 3.3. Sistema de Arquivos Paralelo (PFS)

Os sistemas de arquivos paralelos (PFS) são projetados para permitir acesso simultâneo a dados por múltiplos nós de computação e múltiplos servidores de armazenamento. Soluções como Lustre, GPFS e BeeGFS são amplamente adotadas em cenários científicos e industriais que demandam altas taxas de transferência e forte paralelismo de I/O.

A literatura recente indica, contudo, que a eficiência de um PFS depende do alinhamento entre arquitetura e workload. Chen et al. [Chen et al. 2024] destacam que padrões como acesso compartilhado a arquivos, grande volume de operações de metadados e presença de arquivos pequenos dificultam o atendimento uniforme de todos os cenários por uma única arquitetura de PFS. Assim, embora os PFS sejam a escolha natural para simulações científicas e aplicações com I/O massivo, seu desenho e sua seleção devem considerar o perfil concreto das aplicações executadas.

### 3.4. Comparativo entre DFS e PFS

As diferenças entre sistemas de arquivos paralelos e distribuídos são importantes a serem consideradas ao avaliar opções de armazenamento em um ambiente de computação de alto desempenho (HPC).

Na Figura 1 temos uma visão geral da arquitetura de sistemas de arquivos utilizados em uma solução de HPC.

Sistemas de arquivos distribuídos são comumente utilizados para o compartilhamento de arquivos, e outros casos de uso que envolvem arquivos menores. Já para sistemas de arquivos paralelos são recomendados para aplicações que possuem arquivos maiores e necessitam de uma alta demanda de desempenho, como pesquisa científica, processamento de imagens e análise financeira.

Em uma arquitetura típica de HPC, é comum o uso combinado dessas soluções, conforme ilustrado por diferentes propostas de sistemas em camadas: uma camada de DFS ou armazenamento distribuído pode ser utilizada como backend de grande capacidade, enquanto um PFS otimizado provê o acesso de alto desempenho às aplicações críticas. A decisão entre DFS, PFS ou uma combinação de ambos deve considerar:

- o perfil de acesso (N–N ou N–1, intensidade de uso de metadados, tamanho dos arquivos);

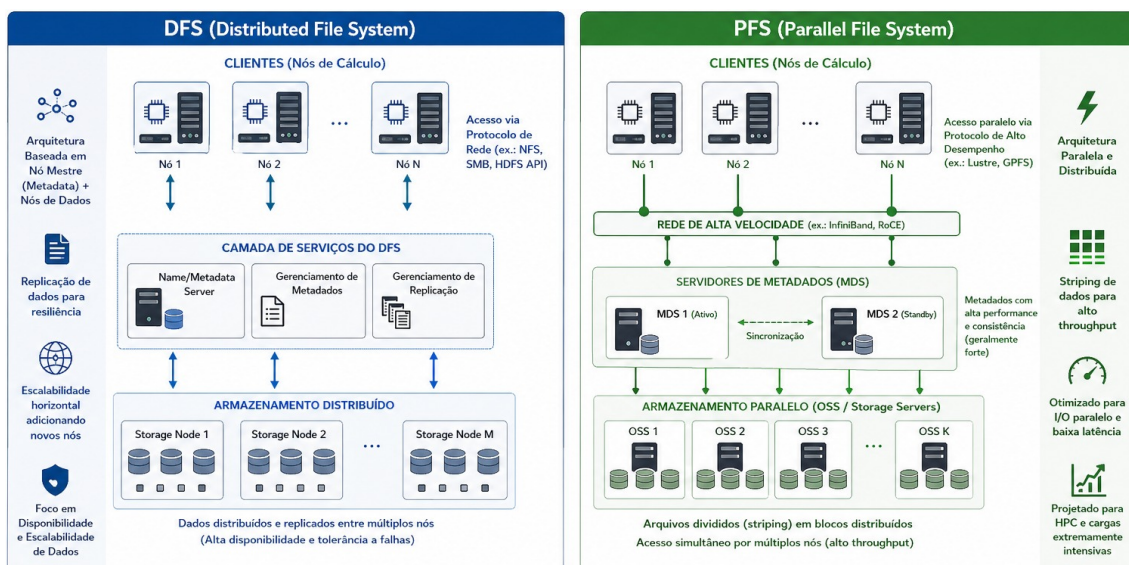


Figura 1. Visão geral da arquitetura de sistemas de arquivos em HPC

- os requisitos de consistência e tolerância a falhas;
- as características do hardware disponível (tipo de rede, SSDs/NVMe, aceleradores);
- as restrições operacionais e de custo do ambiente.

#### 4. Conclusões

Este trabalho apresentou uma visão comparativa sobre NFS, DFS e PFS no contexto de HPC, destacando que a escolha da arquitetura de armazenamento deve ser orientada pelo perfil da aplicação, pelo padrão de acesso aos dados, pelos requisitos de tolerância a falhas e pelas características da infraestrutura disponível.

A revisão da literatura mostrou que pesquisas recentes já avançam em otimizações específicas: de um lado, há esforços para reduzir o custo computacional da resiliência em DFS; de outro, há propostas para ampliar o desempenho de PFS sob workloads complexos e intensivos em metadados. A principal conclusão é que não existe solução universal. Em vez disso, a adequação entre workload e arquitetura deve guiar a decisão.

A compreensão aprofundada das diferenças entre NFS, DFS e PFS é, portanto, fundamental para projetar infraestruturas de HPC capazes de acompanhar o crescimento constante das demandas de armazenamento e processamento de dados.

Como contribuições futuras, o artigo pretende transformar a base teórica aqui apresentada em uma avaliação experimental comparativa, com métricas e cenários definidos, para futuras aplicações em ambientes corporativos e acadêmicos.

#### 5. Trabalhos Futuros

Como encaminhamento prático da pesquisa, propõe-se uma etapa experimental voltada à comparação controlada entre NFS, DFS e PFS em um ambiente de HPC ou cluster de testes com rede de alta velocidade. O objetivo não é apenas reafirmar diferenças conceituais já conhecidas, mas medir como cada arquitetura responde a perfis de carga distintos.

A avaliação futura poderá contemplar, as seguintes métricas: throughput, IOPS, latência, variação com o aumento da concorrência, impacto da rede e custo dos mecanismos de redundância. Também se recomenda separar os testes por perfil de workload, incluindo: (i) arquivos grandes com acesso sequencial; (ii) arquivos pequenos com alta pressão de metadados; (iii) padrão N–N; e (iv) padrão N–1 com compartilhamento intensivo.

Além disso, os resultados de Chen et al. [Chen et al. 2024] e Maisto et al. [Maisto et al. 2026] sugerem dois eixos de aprofundamento relevantes para estudos. O primeiro é a análise de como PFS responde a workloads com forte presença de metadados e compartilhamento de arquivos. O segundo é a investigação do custo computacional da tolerância a falhas em DFS, especialmente quando há uso de erasure coding e eventuais mecanismos de aceleração heterogênea.

Dessa forma, a contribuição prática esperada para a continuidade do estudo é a construção de um protocolo de benchmark orientado por casos de uso, capaz de apoiar decisões arquiteturais em ambientes reais de HPC.

## Referências

- Antelmi, A. and Carlini, E. (2026). Large-scale hpc approaches and applications on highly distributed platforms. *Future Generation Computer Systems*, 179:108365.
- Chen, Y., Yang, H., Lu, K., Huang, W., Wang, J., Wan, J., Zhou, J., Wu, F., and Xie, C. (2024). Peakfs: An ultra-high performance parallel file system via computing-network-storage co-optimization for hpc applications. *IEEE Transactions on Parallel and Distributed Systems*, 35(12):2578–2595.
- Maisto, V., Cilaro, A., Billi, E., and Fader, C. (2026). A hardware/software architecture for multi-threaded offloading of erasure codes in distributed file systems. *Future Generation Computer Systems*, 176:108187.
- Silva, G. P., Bianchini, C. P., and Costa, E. B. (2022). *Programação Paralela e Distribuída com MPI, OpenMP e OpenACC para computação de alto desempenho*. CasaDoCodigo.
- Wang, C., Mohror, K., and Snir, M. (2021). File system semantics requirements of hpc applications. In *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '21, page 19–30, New York, NY, USA. Association for Computing Machinery.