Análise Comparativa de Desempenho entre Lustre e GekkoFS em Cargas de Trabalho Intensivas de Metadados em HPC *

João V. Vargas[1], **Cristiano A. Künas**[1], **Philippe O. A. Navaux**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{jvvoliveira,cakunas,navaux}@inf.ufrgs.br

Resumo. *Sistemas de Computação de Alta Performance (HPC) dependem de sistemas de arquivos paralelos capazes de sustentar altos níveis de throughput e escalabilidade. No entanto, cargas de trabalho modernas — caracterizadas por operações intensivas de metadados e I/O de blocos pequenos — impõem desafios severos a sistemas tradicionais como o Lustre, frequentemente limitados por gargalos em servidores de metadados centralizados. Apresentamos uma análise comparativa entre o Lustre e o GekkoFS, um sistema de arquivos ad-hoc projetado para atenuar tais limitações. Através de benchmarks sintéticos, avaliamos o desempenho de dados e a eficiência em operações de metadados. O objetivo é mapear os cenários de uso em que o GekkoFS oferece vantagens competitivas sobre o Lustre, especialmente em workloads dominados por operações intensivas de metadados e acessos de pequeno porte com alta concorrência.*

1. Introdução

O Lustre é um sistema de arquivos paralelo utilizado em ambientes de Computação de Alta Performance, conhecido por sua alta escalabilidade e capacidade de agregação de throughput. Sua arquitetura separa servidores de metadados (*Metadata Servers* - MDS) e de armazenamento de objetos (*Object Storage Servers* - OSS), permitindo a distribuição de dados por meio de striping entre múltiplos alvos de armazenamento (*Object Storage Targets* - OSTs). Essa abordagem possibilita elevada largura de banda agregada e suporte a milhares de clientes simultâneos, sendo amplamente adotada em supercomputadores e infraestruturas de grande escala que exigem alto desempenho, confiabilidade e eficiência no acesso a grandes volumes de dados [Sun Microsystems, Inc. 2007].

Workloads modernos em HPC têm apresentado padrões de acesso distintos dos tradicionalmente otimizados por sistemas de arquivos paralelos. Aplicações envolvendo análise de dados, treinamento de modelos de aprendizado de máquina e pipelines científicos frequentemente geram grandes quantidades de arquivos pequenos, realizam operações frequentes de criação, abertura e remoção de arquivos e executam acessos de pequeno porte. Esse comportamento pode gerar gargalos significativos em arquiteturas com servidores de metadados centralizados, tornando relevante a investigação de soluções distribuídas [Macedo et al. 2023].

*O presente trabalho foi apoiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Código de Financiamento 001, pelo edital CNPq/MCTI/FNDCT – Universal sob número 408755/2025-3 e pela Petrobras sob número 2020/00182-5. Os experimentos deste trabalho foram realizados na infraestrutura experimental Grid'5000, apoiada por um grupo hospedado pelo Inria e que inclui CNRS, RENATER, diversas universidades e organizações.

Para mitigar tais limitações, surgem sistemas de arquivos efêmeros, entre os quais o GekkoFS destaca-se como uma alternativa promissora, pois é projetado para cenários de *burst buffer* e para *workloads* caracterizados por operações intensivas de metadados. Diferentemente de arquiteturas tradicionais com servidores de metadados centralizados, o `GekkoFS` adota uma abordagem distribuída, armazenando metadados e dados localmente nos nós de computação e utilizando interceptação de chamadas POSIX via `LD_PRELOAD`. Essa estratégia reduz a contenção no gerenciamento de metadados e busca melhorar o desempenho em *workloads* dominados por acessos de pequeno porte e alta concorrência [Vef et al. 2020].

2. Metodologia

As topologias foram configuradas para garantir uma comparação justa. No Lustre, a arquitetura foi composta por um nó dedicado ao servidor de metadados (MDS), dois nós atuando como servidores de armazenamento de objetos (OSS) e três nós configurados como clientes. Já no GekkoFS, foram utilizados três nós como clientes e três como servidores (*daemons*), com metadados e dados distribuídos localmente nos nós de computação.

Os *benchmarks* utilizados foram o MDTest e o IOR. O MDTest é utilizado para medir a escalabilidade de metadados por meio das operações de criação (*create*), consulta (*stat*) e remoção (*remove*) de arquivos. Foram executados quatro processos por nó cliente. Já o IOR foi empregado para avaliar o throughput e IOPS (Operações de Entrada/Saída por Segundo) em operações de leitura e escrita sequenciais. Os testes variaram o tamanho dos blocos entre 64 KB, 1 MB e 4 MB para capturar o comportamento de cada sistema em diferentes granularidades de acesso.

Os experimentos foram conduzidos na infraestrutura experimental Grid'5000, no site de Rennes, utilizando o cluster *paradoxe* [Balouek et al. 2013]. O cluster é composto por 64 nós computacionais, dos quais seis foram alocados para os testes. Cada nó é equipado com dois processadores Intel Xeon Gold 5320 (26 núcleos por CPU, totalizando 52 núcleos físicos por nó), 384 GiB de memória RAM e dois SSDs locais de 1,92 TB. Os nós são interconectados por uma rede de 25 Gbps.

3. Resultados

A Figura 1 apresenta os resultados do MDTest em escala logarítmica para operações de criação (create), consulta (stat) e remoção (remove) de arquivos, variando de 1 a 3 nós com 4 processos por nó.

O GekkoFS supera consistentemente o Lustre nas três operações avaliadas, com a vantagem se acentuando à medida que o número de nós aumenta — comportamento que confirma a hipótese central do trabalho sobre a superioridade da arquitetura distribuída de metadados. Com 1 nó, ambos os sistemas operam em faixas próximas (10^4 ops/s), porém ao escalar para 3 nós, o GekkoFS alcança aproximadamente 10^5 ops/s em todas as operações enquanto o Lustre permanece próximo de 10^4 ops/s — uma diferença de até uma ordem de magnitude.

Este resultado é especialmente relevante para workloads de aprendizado de máquina, nos quais checkpoints frequentes, abertura iterativa de amostras de dados e criação de arquivos temporários geram exatamente o padrão de acesso intensivo a metadados que penaliza arquiteturas com servidor de metadados centralizado.

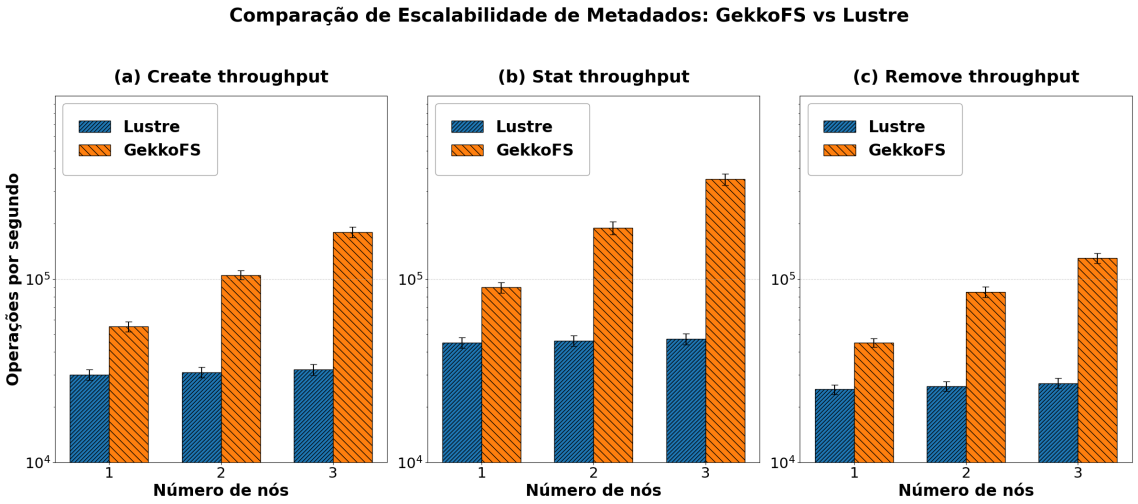

Figura 1. Desempenho do MDTest com quatro processos por nó variando o número de nós.

A Figura 2 revela um cenário distinto. Para operações de escrita sequencial, o Lustre demonstra desempenho competitivo ou superior ao GekkoFS. Com 3 nós, a diferença se reduz, mas o Lustre mantém ligeira vantagem na escrita, resultado esperado dado que o GekkoFS armazena dados localmente via memória/SSD por nós sem as otimizações de escrita em lote presentes no Lustre.

Para operações de leitura sequencial, o Lustre apresenta throughput superior na maioria dos tamanhos de bloco. Esse comportamento reflete sua arquitetura baseada em striping entre OSTs, que explorara o paralelismo durante o acesso aos dados e favorece workloads dominados por leitura de grandes volumes.

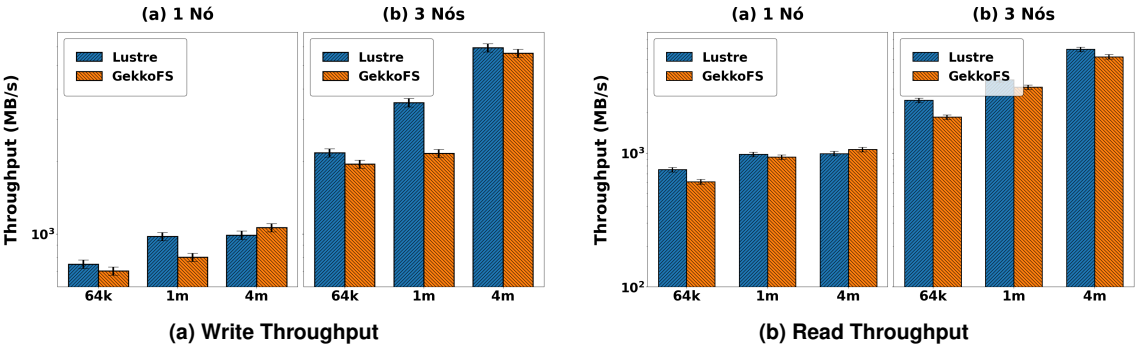

Figura 2. Comparação de Throughput entre Lustre e GekkoFS para operações de escrita e leitura sequenciais.

Note que a escala logarítmica (Fig. 2) suaviza diferenças absolutas. Ambos os sistemas operam na faixa de 10^3 MB/s, indicando que para workloads de dados sequenciais em larga escala — como leitura de datasets de treinamento — a diferença prática entre os dois sistemas é menos crítica do que no caso dos metadados, e a escolha pode depender do padrão predominante (escrita vs. leitura) da aplicação.

Na escrita, o Lustre apresenta IOPS superior ao GekkoFS para blocos pequenos com 1 nó, devido ao uso de cache e agregação de I/O. Contudo, ao utilizar 3 nós, o

desempenho do GekkoFS se aproxima ou supera o Lustre, indicando que o paralelismo distribuído compensa a ausência dessas otimizações.

Na leitura, o Lustre obtém bom desempenho para blocos pequenos. À medida que o tamanho do bloco aumenta, os resultados tendem a convergir, porém o GekkoFS com um IOPS maior.

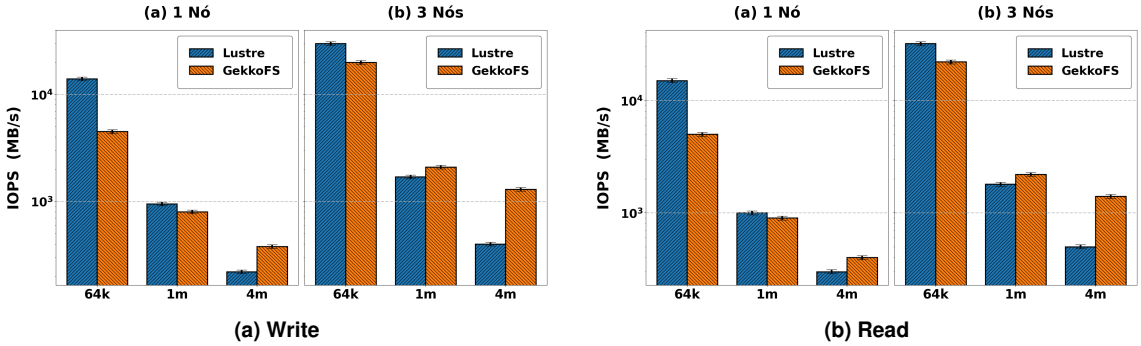

Figura 3. Comparação de IOPS entre Lustre e GekkoFS para 1 e 3 nós.

4. Conclusão

Os experimentos indicaram que o GekkoFS configura-se como uma solução eficaz para *workloads* modernos de HPC, como o treinamento de modelos de aprendizado de máquina, caracterizados por grande quantidade de arquivos pequenos e operações intensivas de metadados. Em contrapartida, o Lustre mantém-se como uma solução robusta para armazenamento persistente e para aplicações que demandam elevada largura de banda agregada no acesso a grandes volumes de dados. Como trabalhos futuros, pretende-se ampliar a avaliação experimental incluindo padrões de acesso com leitura e escrita aleatórias, investigar o impacto do GekkoFS em pipelines científicos reais e analisar o consumo de recursos (CPU e memória) nos nós de computação decorrente da execução dos *daemons* do sistema.

Referências

Balouek, D., Amarie, A., Charrier, G., Desprez, F., Jeannot, E., Jeanvoine, E., Lebre, A., Margery, D., Niclausse, N., Nussbaum, L., Richard, O., Rohr, C., and Sarrazin, H. (2013). Grid'5000: A large scale and highly reconfigurable testbed for experiment-driven research. *Future Generation Computer Systems*, 29(8):2043–2052.

Macedo, R., Miranda, M., Tanimura, Y., Haga, J., Ruhela, A., Harrell, S. L., Evans, R. T., Pereira, J., and Paulo, J. (2023). Taming metadata-intensive hpc jobs through dynamic, application-agnostic qos control. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 47–61. IEEE.

Sun Microsystems, Inc. (2007). Lustre™ file system: High-performance storage architecture and scalable cluster file system. White paper, Sun Microsystems, Inc., Santa Clara, CA, USA.

Vef, M.-A., Moti, N., Süß, T., Tacke, M., Tocci, T., Nou, R., Miranda, A., Cortes, T., and Brinkmann, A. (2020). Gekkofs—a temporary burst buffer file system for hpc applications. *Journal of Computer Science and Technology*, 35:72–91.