Análise do Comportamento de E/S de Aplicações HPC no Supercomputador Intrepid

Valéria S. Girelli[1], Jean Luca Bez[1], Pablo J. Pavan[1], Francieli Z. Boito[2], e 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

[2]Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000 Grenoble, France

{vsgirelli, jlbez, pablo.pavan, navaux}@inf.ufrgs.br

Resumo. *O acesso a dados é um gargalo no desempenho de diversas aplicações de alto desempenho. Portanto, identificar padrões de acesso comuns pode auxiliar na aplicação de otimizações no sistema de armazenamento. Utilizando dados da atividade de E/S de um ano inteiro em um supercomputador, analisamos o comportamento de E/S das aplicações e identificamos padrões ineficientes.*

1. Introdução

O desempenho das operações de E/S em sistemas de computação de alto desempenho (*High Performance Computing – HPC*) é diretamente dependente da forma como as aplicações que executam nesses sistemas realizam essas operações de leitura e escrita [Frings et al. 2009, Boito et al. 2018]. Grandes quantidades de dados são acessadas de forma concorrente, podendo resultar em um gargalo no desempenho de um número crescente de aplicações de HPC. Ocasionalmente, as requisições de leitura e escrita enviadas ao sistema de aramazenamento são muito pequenas, o que dificilmente compensa o custo de acessar os dispositivos de armazenamento [Carns et al. 2009, Boito et al. 2018]. Uma solução para este problema é agregar essas requisições, uma das possíveis otimizações que podem ser aplicadas ao sistema de E/S [Thakur et al. 1999].

Portanto, uma vez que as aplicações de HPC podem apresentar diferentes padrões de acesso, para aplicar otimizações precisa-se antes compreender a forma como essas aplicações realizam operações de E/S. Com o objetivo de compreender o padrão de acesso a dados em um supercomputador, este estudo analisou rastros coletados no supercomputador Intrepid Blue Gene/P, do Argonne National Laboratory (ALCF). Considerando o impacto que o tamanho das requisições pode ter sobre o desempenho das operações de E/S, conduzimos uma análise sobre os tamanhos observados em operações de leitura e escrita com as interfaces POSIX e MPI-IO.

O restante deste artigo está organizado da seguinte forma. Informações a respeito dos dados coletados e da metodologia aplicada estão detalhados na seção 2. A análise e os resultados são apresentados na seção 3. A seção 4 discute trabalhos relacionados. Por fim, a seção 5 conclui este artigo e discute trabalhos futuros.

2. Metodologia

Durante o ano de 2012, dados da execução de uma variedade de aplicações foram coletados pelo Darshan[1], uma ferramenta de caracterização de E/S, no supercomputador

[1]https://www.mcs.anl.gov/research/projects/darshan/

Intrepid Blue Gene/P, em ALCF. O Darshan intercepta chamadas a funções de E/S e gera *logs* com um conjunto de estatísticas para cada arquivo aberto pela aplicação. Os *logs* utilizados foram gerados pelas versões 1.23, 1.24 e 2.0 do Darshan, resultando em 36, 359 e 91.603 execuções, respectivamente. A taxa de aplicações caracterizadas varia entre 20% e 80% de semana para semana, devido a limitações das versões utilizadas. Algumas das informações coletadas foram anonimizadas antes de serem publicadas, como o identificador do *job*, o identificador do usuário e o nome da aplicação [Carns 2013].

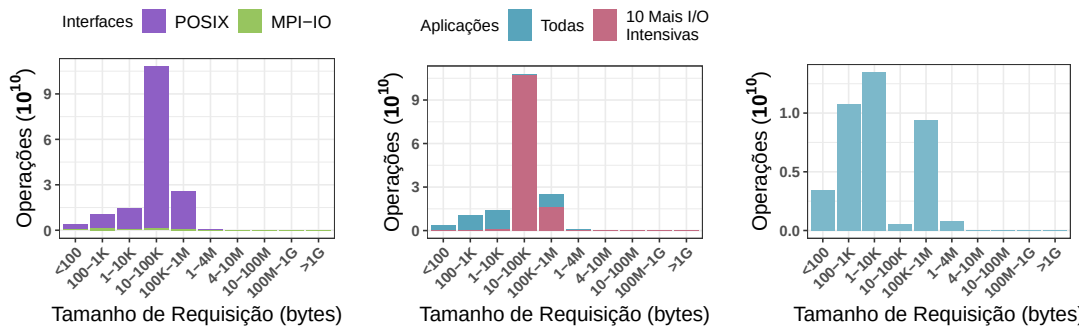
Para extrair e analisar as informações importantes para nosso trabalho, foi utilizada a ferramenta *darshan-parser*, que gera um arquivo texto com informações a respeito da aplicação executada. Além disso, são descritas informações de cada contador capturado pelo Darshan, em um formato tabular. Alguns contadores são separados por tipo de operação (leitura ou escrita) e por interface (MPI-IO ou POSIX). Os tamanhos das operações de leitura e escrita com cada interface são divididos em intervalos.

3. Análise e Resultados

Analisando o número de operações de E/S realizadas com cada interface, observamos que $97,3\%$ das operações de leitura e $99,2\%$ das operações de escrita foram realizadas com POSIX. Trabalhos relacionados mostram que POSIX ainda é mais amplamente utilizada se comparado a MPI-IO [Smirni and Reed 1998, Luu et al. 2015], mas os dados coletados em 2012 realçam ainda mais a diferença entre a utilização de POSIX e de MPI-IO.

A Figura 1a mostra que $64,3\%$ das operações de leitura foram entre 10KB e 100KB. Para operações de escrita, o tamanho mais comum foi de até 100 bytes, como mostra a Figura 2a, um tamanho consideravelmente pequeno e representado por $98,7\%$ das requisições de escrita. Trabalhos relacionados apontaram requisições de leitura e escrita de 16KB, 512KB e 1MB [Kim et al. 2010]. Em [Carns et al. 2011], observou-se requisições de leitura de tamanho entre 100KiB e 1MiB e requisições de escrita de 100 bytes a 1KiB. Pode-se observar, portanto, que as aplicações aqui caracterizadas realizaram operações de tamanho ainda menor do que o observado em trabalhos relacionados. Acessos pequenos podem prejudicar o desempenho das aplicações [Carns et al. 2009, Boito et al. 2018], uma vez que eles determinam o tamanho das transferências de dados entre os nós de processamento e os dispositivos de armazenamento.

Portanto, buscamos analisar se esse comportamento ineficiente representatava o comportamento geral observado no supercomputador Intrepid, ou se ele era resultado das operações de E/S de apenas um grupo de aplicações. Analisando a influência das 10



(a) Total de operações de leitura com ambas as interfaces.

(b) Impacto das 10 aplicações mais E/S intensivas.

(c) Sem as 10 aplicações mais intensivas em E/S.

Figura 1. Análise das requisições de leitura. O eixo y possui diferentes escalas.

aplicações que mais realizaram operações de E/S, observamos pouco impacto na quantidade de operações realizadas com MPI-IO. No entanto, a Figura 1b mostra que essas 10 aplicações foram responsáveis por $99,5\%$ das requisições de leitura de tamanho 10KB a 100KB realizadas com POSIX. Dentre as operações de escrita realizadas com POSIX, as 10 aplicações mais intensivas em E/S foram responsáveis por $99,1\%$ das requisições de até 100 bytes, como mostra a Figura 2b.

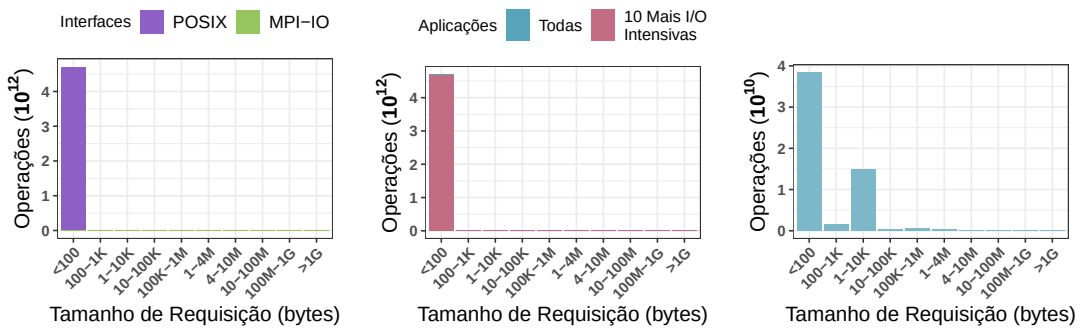
Se desconsiderarmos as 10 aplicações mais intensivas em E/S, observamos que o tamanho de acesso mais comum para operações de leitura passa a ser de 1KB a 10KB, representado por $35,1\%$ das requisições, como mostra a Figura 1c. Para operações de escrita, o tamanho mais utilizado continua sendo de até 100 bytes, porém a Figura 2c mostra uma distribuição mais ampla das requisições.

4. Trabalhos Relacionados

Em um trabalho de caracterização da carga de E/S do supercomputador Spider, em Oak Ridge National Laboratory (ORNL) [Kim et al. 2010], foram observados três principais tamanhos de acesso: até 16KB, 512KB e 1MB. Esses três tamanhos de acesso representaram mais de 95% to total de requisições. Cerca de 50% das operações de escrita e 20% das operações de leitura eram requisições de até 16KB. Foi possível observar que a maior largura de banda era atingida com requisições de 1MB.

Analisou-se o comportamento de 66 aplicações de engenharia e ciências executando no supercomputador Intrepid, em Argonne, durante dois meses de 2010 [Carns et al. 2011]. Demonstrou-se que o tamanho de requisições de leitura mais comum era entre 100KiB (kibibytes) e 1MiB (mebibytes), enquanto que o mais comum para operações de escrita era entre 100 bytes e 1KiB. Constatou-se que algumas poucas aplicações influenciavam os tamanhos de acesso observados. Se essas aplicações fossem desconsideradas na análise, o tamanho de acesso mais frequente para ambas as operações de leitura e escrita passava a ser entre 100KiB e 1MiB. Demonstrou-se também que algumas aplicações aumentavam seu desempenho quando realizavam requisições maiores.

Apesar de o trabalho anterior também ter utilizado dados a respeito da carga de E/S do supercomputador Intrepid, o estudo utilizou um conjunto de dados menor. Além disso, este é o primeiro estudo a investigar o comportamento de E/S do ano inteiro de 2012 neste supercomputador. Nós analisamos os dados de tamanhos de acesso também levando em consideração diferentes interfaces.



(a) Total de operações de escrita com ambas as interfaces.

(b) Impacto das 10 aplicações mais E/S intensivas.

(c) Sem as 10 aplicações mais intensivas em E/S.

Figura 2. Análise das requisições de escrita. O eixo y possui diferentes escalas.

5. Conclusão e Trabalhos Futuros

Este estudo utilizou dados coletados durante um ano inteiro de caracterização de E/S no supercomputador Intrepid utilizando a ferramenta Darshan. Com uma quantidade considerável de dados, observamos que POSIX ainda é surpreendentemente utilizada pelas aplicações que executam no supercomputador. De modo geral, as aplicações realizam operações de escrita utilizando um tamanho consideravelmente pequeno, que não excede os 100 bytes. Isso demonstra que as operações de E/S estão sendo realizadas de maneira ineficiente, provavelmente acessando poucas variáveis por vez. Dessa forma, as aplicações não estão tirando proveito da agregação de requisições realizada pela interface MPI-IO ou por outra interface de alto-nível.

Portanto, pode-se buscar identificar outros comportamentos e padrões de E/S além dos observados que precisam de atenção especial. Desta forma, é possível direcionar esforços no desenvolvimento de otimizações em diferentes níveis do sistema de E/S como, por exemplo, na agregação de requisições pequenas como as observadas neste trabalho.

6. Agradecimentos

Essa pesquisa recebeu apoio de PIBIC CNPq-UFRGS e PROBIC FAPERGS-UFRGS. Também foi apoiada por projeto da Petrobras, concessão n. 2016/00133-9. Foram utilizados recursos de Argonne Leadership Computing Facility, de Argonne National Laboratory, que é apoiado pelo Escritório de Ciência do Departamento de Energia dos Estados Unidos sob contrato DE-AC02-06CH11357.

Referências

Boito, F. Z., Inacio, E. C., Bez, J. L., Navaux, P. O. A., Dantas, M. A. R., and Denneulin, Y. (2018). A checkpoint of research on parallel I/O for high-performance computing. *ACM Comput. Surv.*

Carns, P. (2013). ALCF I/O Data Repository. Technical report, Argonne Leadership Computing Facility.

Carns, P., k. Harms, Allcock, W., Bacon, C., Lang, S., Latham, R., and Ross, R. (2011). Understanding and improving computational science storage access through continuous characterization. *Trans. Storage*, 7(3):8:1–8:26.

Carns, P., Lang, S., Ross, R., Vilayannur, M., Kunkel, J., and Ludwig, T. (2009). Small-file access in parallel file systems. In *2009 IEEE International Symposium on Parallel Distributed Processing*, pages 1–11.

Frings, W., Wolf, F., and Petkov, V. (2009). Scalable massively parallel I/O to task-local files. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, pages 17:1–17:11.

Kim, Y., Gunasekaran, R., Shipman, G. M., Dillow, D. A., Zhang, Z., and Settlemyer, B. W. (2010). Workload characterization of a leadership class storage cluster. pages 1–5.

Luu, H., Winslett, M., Gropp, W., Ross, R., Carns, P., Harms, K., Prabhat, M., Byna, S., and Yao, Y. (2015). A multiplatform study of I/O behavior on petascale supercomputers. In *Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing*.

Smirni, E. and Reed, D. (1998). Lessons from characterizing the input/output behavior of parallel scientific applications. *Performance Evaluation*, 33(1):27 – 44.

Thakur, R., Gropp, W., and Lusk, E. (1999). Data sieving and collective I/O in ROMIO. In *Proceedings. Frontiers '99. Seventh Symposium on the Frontiers of Massively Parallel Computation*.