Simulando o Impacto da Organização Interna de um SSD no seu Desempenho

Vinícius Rodrigues Machado, Francieli Zanon Boito, Rodrigo Virote Kassick, Philippe O. A. Navaux

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)

{vrmachado, fzboito, rvkassick, navaux}@inf.ufrgs.br

Resumo. *Este artigo analisa o desempenho de um SSD ideal com o simulador de discos DiskSim, sob diferentes modelos de acesso. Os sistemas atuais ainda são adaptados para a utilização de HDs. Entretanto, a organização do SSD não é totalmente compatível com essas adaptações, criando a necessidade de analisar a sua performance, a fim de possibilitar que novos sistemas possam obter o melhor desempenho desses dispositivos.*

1. Introdução

O Disco de estado sólido - *Solid State Drive* ou SSD - é uma alternativa recente de armazenamento ao disco rígido (HD). Suas vantagens sobre o HD incluem : resistência a vibrações e quedas, tamanho, dissipação de calor, consumo de energia e produção de ruído. Por outro lado, muitos dos sistemas atuais foram desenvolvidos ainda considerando características de discos rígicos, como acesso sequencial várias vezes mais rápido que acesso aleatório. Por esse motivo, é importante estudar e entender o desempenho de SSDs sob diferentes padrões de acesso, pois isso possibilita a elaboração de otimizações para que novos sistemas possam obter o melhor desempenho no acesso a esses dispositivos. [Chen et al. 2009, Kim et al. 2009]

O comportamento do desempenho observado pode variar significantemente entre diferentes discos de estado sólido. Isso acontece como resultado de diversas opções de projeto que são parte do desenvolvimento desses dispositivos [Rajimwale et al. 2009]. Nesse contexto, o objetivo dessa pesquisa é analisar o impacto dessas opções de projeto no desempenho de SSDs. Uma dificuldade encontrada nesse tipo de estudo é que os fabricantes normalmente não divulgam os parâmetros dos seus dispositivos, os mantendo como segredo industrial. Por esse motivo, foi utilizado um simulador para o presente estudo.

Nesse artigo, o impacto de diversas configurações de *garbage collection* para SSDs é avaliado sob diferentes padrões de acesso. A metodologia desses experimentos é detalhada na próxima seção. A Seção 3 apresenta e discute os resultados obtidos, e a Seção 4 traz considerações finais.

2. Metodologia

A ferramenta de simulação de discos *DiskSim* [Bucy et al. 2008] foi utilizada para avaliar o desempenho de SSDs com diferentes configurações. O simulador original conta somente com a simulação de HDs, mas uma extensão desenvolvida pela *Microsoft Research* adicionou uma modelagem de SSDs ideais [Agrawal et al. 2008]. A ferramenta permite a

especificação de parâmetros da organização de SSDs como tamanho de páginas, tamanho de blocos, número de *chips flash* e política de *garbage collection*.

Os testes foram executados utilizando a interface do *DiskSim* para geração de requisições ao dispositivo simulado. Os acessos foram obtidos de arquivos de rastros sintéticos contendo listas de requisições síncronas. Cada simulação, contida em um arquivo de rastro, corresponde a 300 requisições de um tamanho fixo formando um dos 4 padrões de acesso testados: leitura sequencial, leitura aleatória, escrita sequencial e escrita aleatória. O tamanho das requisições foi variado de 8KB a 4MB, e o tempo necessário para atender cada requisição foi obtido através da média dos 300 acessos realizados durante cada simulação. Esses resultados possuem uma confiança estatística de 95% com erro máximo de 5%.

Foram avaliadas duas organizações de SSD:

- páginas de 8KB, blocos de 256KB (64 páginas), 2048 blocos por *plane*, 8 *planes* por *chip flash* e 8 *chips* ao todo, resultando em um dispositivo de 32GB;
- o número de blocos por *plane* foi reduzido para 256, resultando em um SSD de 4GB.

Duas políticas de *garbage collection* providas pelo simulador foram avaliadas: *greedy* e *wear-aware*. A operação desse mecanismo em *background* foi habilitada através de um parâmetro, a fim de observar o seu impacto nos testes. O percentual mínimo de blocos livres é regulado por outro parâmetro, de forma que o processo de *garbage collection* é disparado em *background* quando a quantidade de blocos livres no dispositivo cai abaixo do limite estipulado. Para esse parâmetro, foram testados os valores 5%, 50% e 90%. A próxima seção apresenta os resultados obtidos com essas configurações.

3. Análise de Desempenho

Os resultados obtidos com os diferentes testes para todas as configurações de SSD descritas na seção anterior estão representados nos gráficos da Figura 1, separados por padrão de acesso. Em cada gráfico, o eixo x representa os diferentes tamanhos de requisição testados, e o eixo y traz o tempo para processar cada requisição (média das 300 requisições de cada simulação). As linhas representam as combinações de parâmetros do modelo de SSDs.

SSDs não permitem a re-escrita de valores em células *flash*, que precisam ser apagadas antes de serem usadas novamente. Por esse motivo, escrever uma página em um disco de estado sólido normalmente requer a escrita em uma outra página do dispositivo, marcando a anterior como inválida para posterior limpeza pelo mecanismo de *garbage collection*. Não esperava-se, portanto, observar diferenças nos testes de leitura ocasionadas por mudanças nas configurações desse mecanismo.

Pode-se observar que os tempos observados para os testes de leitura sequencial (Figura 1(a)) foram significantemente menores que para os testes de leitura aleatória (Figura 1(b)). Os resultados para leitura aleatória são semelhantes aos testes de escrita. Em todos esses houve um pequeno impacto no desempenho ocasionado pelas variações nos parâmetros do modelo. Como trabalho futuro, pretende-se investigar o que pode estar causando esse inesperado efeito nas operações de leitura, que não deveriam ser afetadas.

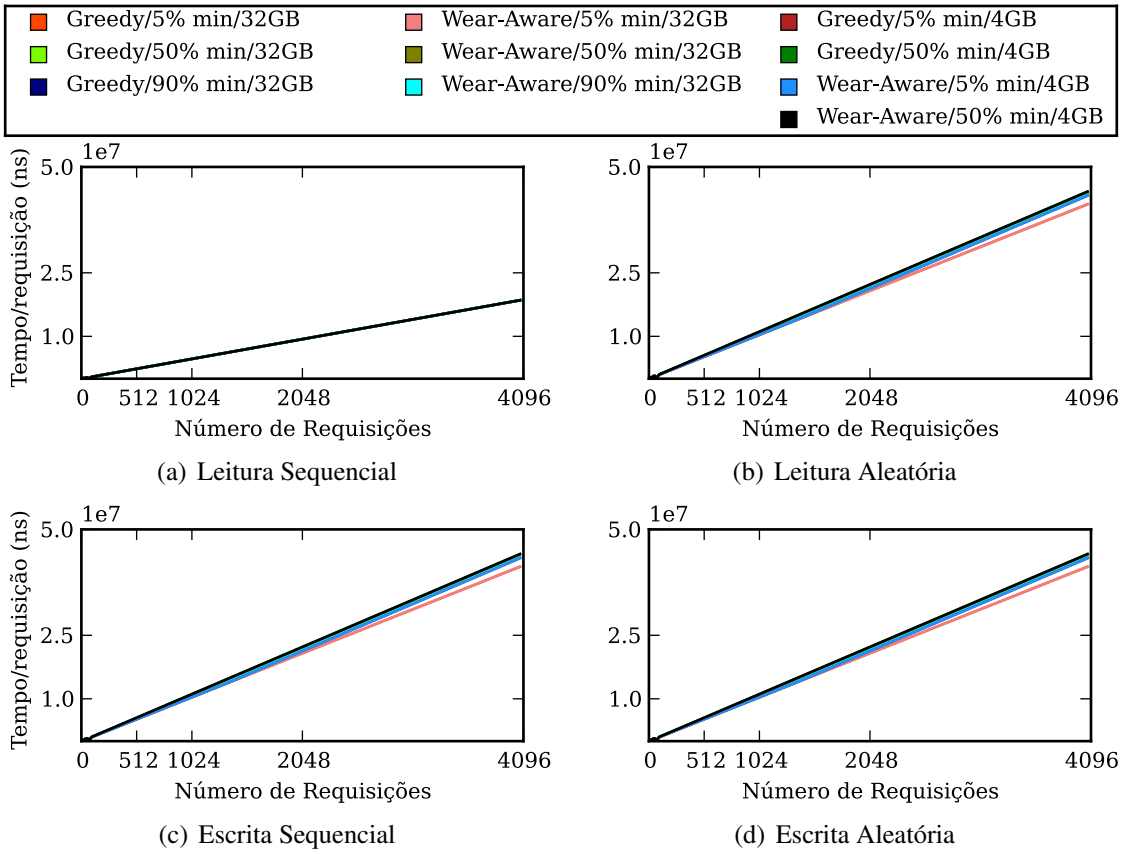

Figura 1. Resultados para todas as configurações testadas.

Não foram observadas diferenças entre as duas políticas de *garbage collection* analisadas - *greedy* e *wear-aware*. Uma possível explicação é que a carga de trabalho gerada nesse estudo não tenha sido o bastante para explorar as diferenças entre as políticas. Pretende-se repetir a avaliação com diferentes padrões de acesso a fim de investigar essas diferenças.

Os testes em que a quantidade de blocos livres foi limitada a apenas 5% no SSD de 32GB obtiveram o melhor resultado de todas as configurações testadas. Nesse caso, o tamanho dos arquivos gerados não foi o bastante para atingir o limite e disparar a execução do *garbage collector*. Nos dispositivos de 32GB, aumentar a quantidade mínima de blocos livres para 50% diminui o desempenho em 5.5% em média, mas aumentar para 90% não continua piorando o desempenho, apenas mantendo o sobrecusto anterior.

O maior tempo por requisição (pior desempenho) foi observado nos testes com mínimo de 50% de blocos livres nos dispositivos de 4GB, pois nesse caso a execução dos testes atinge o limite de blocos livres mais rapidamente porque o dispositivo é menor. Aumentar a quantidade mínima de blocos livres de 5% para 50% nos SSDs de 4GB resulta em um desempenho em média 7% pior (diferença maior do que a observada para os SSDs de 32GB).

4. Conclusão

Esse artigo apresentou uma análise do impacto no desempenho de parâmetros da organização interna de SSDs. Através de simulações, foram avaliados o tamanho do

dispositivo, a política de *garbage collection* empregada e a quantidade mínima de blocos livres a ser mantida. Essa quantidade regula a execução do *garbage collector*, que é disparado quando for necessário liberar blocos.

Foi observado um sobrecusto de 5.5% a 7% induzido pela execução do *garbage collector* em *background*. Nos casos testados, o parâmetro de maior impacto no desempenho foi a quantidade mínima de blocos livres. Não foi observada diferenças significativas entre os resultados para as diferentes políticas.

Como trabalho futuro, pretende-se investigar o resultado inesperado do impacto desses parâmetros nos testes com leitura aleatória. Essa investigação, além do estudo do código do simulador utilizado, poderá incluir o uso de outros simuladores e de dispositivos reais.

Referências

Agrawal, N., Prabhakaran, V., Wobber, T., Davis, J. D., Manasse, M., and Panigrahy, R. (2008). Design tradeoffs for ssd performance. Technical Report 2008.

Bucy, J. S., Schindler, J., Schlosser, S. W., Ganger, G. R., and Contributors (2008). The disksim simulation environment version 4.0 reference manual.

Chen, F., Koufaty, D. A., and Zhang, X. (2009). Understanding intrinsic characteristics and system implications of flash memory based solid state drives. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '09, pages 181–192, New York, NY, USA. ACM.

Kim, J., Oh, Y., Kim, E., Choi, J., Lee, D., and Noh, S. H. (2009). Disk schedulers for solid state drivers. In *Proceedings of the Seventh ACM International Conference on Embedded Software*, EMSOFT '09, pages 295–304, New York, NY, USA. ACM.

Rajimwale, A., Prabhakaran, V., Davis, J. D., of Wisconsin, U., Madison, Research, M., and Valley, S. (2009). Block management in solid-state devices.