

Eficiência Energética e Desempenho de E/S com Arquiteturas de Baixa Potência *

Pablo J. Pavan¹, Ricardo K. Lorenzoni¹, Jean L. Bez², Edson L. Padoin^{1,2},
Francieli Z. Boito³, Philippe O. A. Navaux², Jean-François Méhaut⁴

¹Universidade Reg. do Noroeste do Estado do Rio G. do Sul (UNIJUI) - Ijuí - RS - Brasil

²Universidade Federal do Rio Grande do Sul (UFRGS) - Porto Alegre - RS - Brasil

³Universidade Federal de Santa Catarina (UFSC) - Florianópolis - SC - Brasil

⁴Laboratoire d'Informatique de Grenoble (LIG), Université Grenoble Alpes - France

{pablo.pavan, ricardo.lorenzoni, padoin}@unijui.edu.br

{jean.bez, navaux}@inf.ufrgs.br, francieli.boito@posgrad.ufsc.br

jean-francois.mehaut@imag.fr

Abstract. *This paper presents an I/O performance and energy efficiency analysis of low-power processors when compared to conventional architectures. This study aims at evaluating the viability of using such low-power architectures as servers to file systems in HPC environments. Results have shown that using the MPSoC leads to energy efficiency up to 136 times higher than what is observed with the PC. This advantage is caused by its up to 6.7 times lower power demand. The study concluded that a PC storage server with HDD could be replaced by multiple MPSoC with SSDs to keep a similar performance while decreasing power consumption by up to 85%.*

Resumo. *Este artigo apresenta uma análise de desempenho e eficiência energética de operações de E/S em processadores de baixo consumo quando comparados a arquiteturas convencionais. O objetivo é analisar a viabilidade da utilização destes dispositivos na implementação de sistemas de arquivos para HPC. Os resultados mostraram que o uso do MPSoC levou a uma eficiência energética até 136 vezes maior do que o observado com o PC. Essa vantagem é causada por uma demanda de potência até 6,7 vezes menor. Concluiu-se que um servidor de armazenamento PC com HDD pode ser substituído por múltiplos MPSoC com SSD para manter um desempenho semelhante com uma demanda de potência até 85% menor.*

1. Introdução

O aumento do poder de processamento de arquiteturas para computação de alto desempenho (*High Performance Computing* - HPC) foi acompanhado de um aumento significativo da demanda de potência desses sistemas. Um fenômeno similar ocorre em *data centers*, causado pelo crescimento acelerado da quantidade de dados gerados e utilizados pelas aplicações. Essas grandes demandas energéticas levam a uma situação não ecologicamente correta e

*Trabalho parcialmente apoiado por CNPq, CAPES e FAPERGS. Contou com recursos do projeto Europeu EU H2020 Programme e do MCTI/RNP-Brasil no Projeto HPC4E, sob número 689772. Realizado no contexto do LICIA - Laboratório Internacional Associado entre a UFRGS e o Laboratoire d'Informatique de Grenoble.

pouco sustentável economicamente. Nesse sentido, um relatório do DARPA sugere que futuros sistemas de HPC - dos quais se espera exaflops de desempenho - deveriam obedecer a um limite de 20 MW de demanda de potência [Kogge et al. 2008].

Com essa premissa, pesquisadores têm buscado alternativas para respeitar tais limites. Uma estratégia amplamente abordada é o uso de arquiteturas de baixa potência, substituindo processadores convencionais por processadores *Advanced RISC Machine* (ARM). Apesar do menor desempenho, essas arquiteturas propiciam uma melhor eficiência energética para algumas classes de aplicações científicas [Padoin et al. 2014].

No entanto, o processamento não é o único responsável pelo consumo energético dos sistemas de HPC. Devido à crescente diferença entre as velocidades de processamento e de acesso a dados, é comum que aplicações passem uma porção significativa das suas execuções realizando operações de entrada e saída (E/S). Portanto, é imperativo que o consumo energético dessas operações também seja estudado e aprimorado.

Em arquiteturas de alto desempenho, as operações de E/S são feitas a sistemas de arquivos paralelos, onde máquinas dedicadas atuam como servidores de dados. As principais atividades dos servidores são receber requisições dos nós de processamento e processá-las, acessando dispositivos de armazenamento. Em outras palavras, o poder de processamento dessas máquinas é muitas vezes subutilizado em função do tempo gasto com operações de E/S. Nesse cenário, o uso de arquiteturas de baixa demanda de potência como servidores de armazenamento poderia levar a uma melhor eficiência energética.

A fim de avaliar a viabilidade dessa ideia, este artigo apresenta um estudo comparativo de desempenho e eficiência energética entre um computador equipado com processador tradicional e um *Multi-Processor System-on-Chip* (MPSoC) equipado com um processador ARM. Essa avaliação inclui discos rígidos (HDDs) e dispositivos de estado sólido (SSDs) para o armazenamento. Além disso, o impacto da arquitetura e do dispositivo de armazenamento são avaliados para diferentes padrões de acesso.

O restante do trabalho está organizado da seguinte forma. A Seção 2 discute trabalhos relacionados. A Seção 3 descreve o método experimental e os ambientes utilizados. Resultados são discutidos na Seção 4, seguidos das conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Diversos trabalhos da literatura focam no consumo energético de operações de E/S pois elas correspondem à maior parte do tempo de execução de muitas aplicações. No passado, pesquisadores exploraram a utilização de discos de múltiplas velocidades para servidores de armazenamento [Carrera et al. 2003, Gurumurthi et al. 2003, Zhu et al. 2005]. Estratégias mais recentes empregam principalmente *Dynamic Voltage and Frequency Scaling* (DVFS) para reduzir o consumo do processador durante as operações de E/S, uma vez que estas não costumam requerer um grande poder de processamento.

[Ge et al. 2012] propõem uma estratégia para arquiteturas de HPC que envolve a realização de DVFS nos nós de processamento durante as suas operações de E/S para o sistema de arquivos paralelo. A estratégia apresentada pelos autores leva em consideração as características dos acessos das aplicações para decidir a frequência ótima. Uma técnica semelhante havia anteriormente sido aplicada para aplicações sequenciais por [Shang e Wang 2011]. [Saito et al. 2013] também aplicam DVFS a nós de processamento durante as operações de checkpointing para dispositivos de armazenamento locais.

[Lee 2009] aplica estratégias de DVFS para economizar energia em sistemas multi-core ociosos. [Peraza et al. 2013] e [Lim et al. 2006] reduzem a potência dos processadores em períodos de comunicação por MPI. [Hosseinimotlagh et al. 2014] e [Younge et al. 2010] analisam recursos ociosos em ambientes de cloud para economizar energia. [Ibrahim et al. 2016] estudam desempenho e consumo energético de aplicações *MapReduce* em ambientes de cloud usando diferentes estratégias de DVFS. Diversos outros trabalhos também consideram o consumo energético de aplicações MapReduce em ambientes de cloud. [Cardosa et al. 2012] propõem algoritmos para alocação de máquinas virtuais em máquinas reais visando a eficiência energética.

[Amur et al. 2010] apresentam um sistema de arquivos distribuído em que réplicas primárias dos dados são armazenadas em servidores que estão sempre ligados, enquanto outras réplicas são colocadas em servidores que podem ser ligados ou desligados de acordo com a demanda. [Kim et al. 2011] desenvolvem e avaliam algoritmos que definem o subconjunto de servidores a serem mantidos ligados a fim de manter os dados disponíveis com o menor consumo energético necessário.

[Nijim et al. 2009] combinam dispositivos de armazenamento baseados em flash (SSDs) com discos rígidos para prover armazenamento com menor consumo energético. Isso é alcançado utilizando os dispositivos mais rápidos como uma cache para os discos rígidos. Essa estratégia híbrida de armazenamento é explorada em diversos trabalhos para prover alto desempenho para servidores de E/S [Welch e Noer 2013]. Nesses casos, o SSD é utilizado como uma cache por causa do seu alto custo por byte, que inviabilizaria a substituição total dos discos rígidos.

Independentemente das técnicas discutidas nesta seção, o poder de processamento das máquinas que trabalham como servidores de armazenamento é muitas vezes subutilizado, uma vez que a principal atividade dos servidores é acesso ao dispositivo de armazenamento. Portanto, o uso de arquiteturas mais simples - e que tenham menor consumo energético - nesses servidores poderia constituir uma boa alternativa. Essa alternativa seria complementar às apresentadas nos trabalhos discutidos, e avaliar a sua viabilidade é o objetivo deste trabalho. Apesar da viabilidade do uso dessas arquiteturas para a computação de alto desempenho ter sido objeto de alguns trabalhos, não foram encontradas pesquisas que investiguem o consumo energético de operações de E/S em arquiteturas de baixa potência.

3. Método experimental

Dois ambientes foram utilizados para este trabalho. O primeiro, chamado de PC, é um computador tradicional com um processador Intel Core2Duo modelo E8400. Esse processador é da arquitetura *Wolfdale* e possui pipeline de 14 estágios com execução de até 4 instruções por ciclo. O equipamento possui 6 GB de memória RAM com frequência de 800 MHz.

O segundo ambiente é um MPSoC CubieTruck com um SoC A20 fabricado pela AllWinnerTech e uma dual GPU MALI400 MP2, chamado de MPSoC. O processador é um Dual Core ARM Cortex-A7. Esse processador possui uma arquitetura superescalar, duas unidades de execução parcial, *pipeline* com 8 estágios e execução em ordem. Ele é baseado na arquitetura ARMv7-A e permite escalabilidade e controle sobre o consumo de energia, uma vez que possibilita o desligamento de qualquer um dos cores quando estiverem ociosos. Possui arquitetura e conjunto de instruções idênticos aos processadores Cortex-A15 e Cortex-A17, com a diferença que, na microarquitetura do processador Cortex-A7, o foco é a eficiência energética [Blake et al. 2009]. O equipamento possui 2 GB de memória RAM

com frequência de 480 MHz.

O sistema operacional instalado em ambos equipamentos é GNU/Linux. Na PC utilizou-se Ubuntu com kernel 3.16.0-38 e no MPSoC Debian com kernel 3.4.106. O sistema de arquivos utilizado para os experimentos foi o *ext4*. A Tabela 1 apresenta as principais características dos ambientes experimentais.

Tabela 1. Configuração dos ambientes

	PC	MPSoC
Processador	Intel Core2Duo	ARM Cortex A7
Modelo Processador	E8400	AllWinnerTech SoC A20
Técnica de Fabricação (nm)	45	40
Frequência de Clock	3.0GHz	960MHz
Número de processadores	1	1
Cores/Processador (#)	2	2
TDP do processador (W)	65	0,25
Cache L1/Core (KB)	64 x 2	64
Cache L2/Core (KB)	6144	1024
Memória (GB)	6 (DDR2)	2 (LP DDR3)

Quatro dispositivos de armazenamento foram utilizados, dois SSDs e dois HDDs, a fim de cobrir diferentes características. Eles são apresentados na Tabela 2. Os nomes apresentados na primeira coluna da tabela serão utilizados no restante do texto para referenciá-los. Em todos os experimentos foi utilizada a interface SATA II, suportada pelas duas arquiteturas e por todos os dispositivos. Foram realizados testes com os quatro dispositivos de armazenamento nos dois equipamentos, totalizando oito configurações.

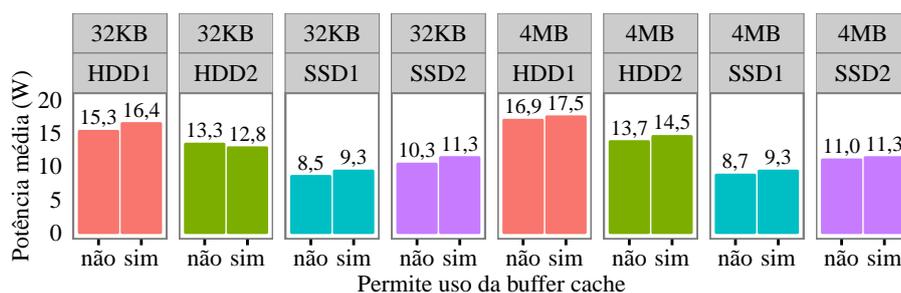
Tabela 2. Dispositivos de armazenamento utilizados

	Tipo	Fabricante	Capacidade (GB)	RPM	Especificações Fabricante	
					Tensão (VDC)	Corrente (A)
HDD1	HDD	Western Digital	160	5400	5	0,55
HDD2	HDD	Seagate	500	7200	5	0,45
SSD1	SSD	Samsung	240	-	5	0,50
SSD2	SSD	Kingston	120	-	5	1,00

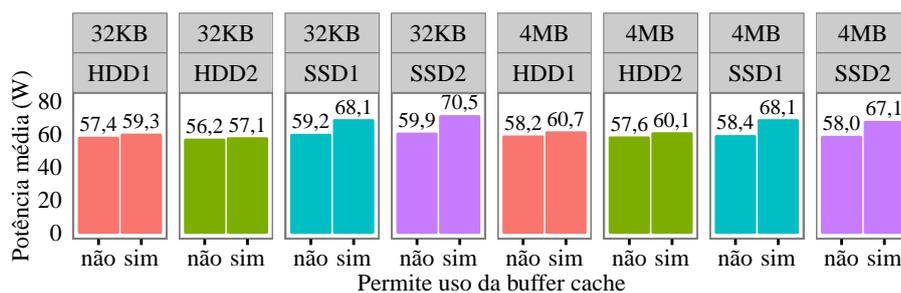
O benchmark utilizado para os testes foi o IOzone¹, escolhido por ser amplamente utilizado e por permitir a descrição de diversos padrões de acesso. Experimentos foram realizados em cada uma das configurações *com* e *sem* o uso da buffer cache. Para os testes sem cache, foi utilizado o parâmetro “-I” para a realização de operações diretamente no dispositivo. Foram testados os seguintes padrões de acesso: escrita sequencial; escrita aleatória; leitura sequencial; e leitura aleatória.

Outro parâmetro avaliado nos testes foi o tamanho das requisições. Neste caso, foram utilizados 32 KB ou 4 MB. Todos os testes acessam 2 GB de dados, ou seja, aumentar o tamanho de cada requisição significa diminuir o número de requisições geradas. Portanto, em cada configuração foram realizados 16 testes, totalizando 128 experimentos. Cada experimento foi repetido dez vezes.

¹Disponível em <http://www.iozone.org/>



(a) Resultados no MPSoC



(b) Resultados no PC

Figura 1. Potência média observada durante os experimentos.

Para medição de potência foi utilizado o equipamento Dranetz PP-4300, que mensura tensão e corrente alternada (CA) de todo o equipamento [Padoin et al. 2013].

4. Resultados

Nesta seção serão apresentados os resultados obtidos. Primeiramente será analisado como o padrão de acesso das aplicações impacta na demanda de potência. Então, um comparativo de eficiência energética será apresentado na Seção 4.2.

4.1. Impacto do padrão de acesso na demanda de potência

Para investigar se a forma como os acessos são feitos - o padrão de acesso - tem algum efeito na demanda de potência, nesta seção são apresentados resultados de potência média. Esse valor é obtido através da média aritmética das medições de potência instantânea providas pelo equipamento de medição durante cada teste. A potência é analisada, uma vez que ela não depende da duração dos testes, ou seja, evidencia-se o efeito do padrão de acesso na demanda da potência sem considerar o seu impacto no tempo de execução.

A Figura 1 apresenta todos os valores de potência obtidos, no MPSoC na Figura 1(a) e no PC na Figura 1(b). Em cada figura, as quatro primeiras caixas apresentam resultados de testes que geram requisições de 32 KB e as outras quatro caixas, resultados com requisições de 4 MB. Cada dispositivo de armazenamento é representado por uma cor diferente. Em cada gráfico, a primeira barra representa o teste sem o uso da cache (acesso direto ao dispositivo de armazenamento), enquanto a segunda barra representa o teste que permite o uso da cache. **O uso da cache aumenta a demanda de potência em até 17,7%, especialmente no PC e com SSDs.** Isso acontece porque usar a cache significa maior uso da memória RAM, o que possui um custo em demanda de potência, especialmente no PC. Os dados apresentados também sugerem que **o tamanho das requisições não tem um impacto significativo na demanda de potência.**

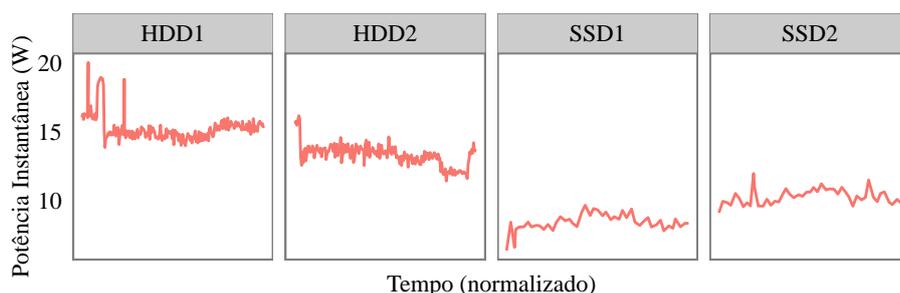


Figura 2. Potência instantânea durante testes no MPSoC com todos os dispositivos de armazenamento testados. Testes sem cache e com requisições de 32 KB.

A Figura 2 apresenta a potência instantânea reportada durante testes no MPSoC com requisições de 32 KB sem cache. Cada caixa representa um dispositivo de armazenamento, e em cada gráfico o eixo x é o tempo daquela execução. Esses tempos foram normalizados para que os quatro gráficos tivessem o mesmo tamanho. Os gráficos dos testes com os HDDs são mais “densos” porque eles apresentam maior tempo de execução e, portanto, as linhas incluem um número maior de pontos. Cada teste inclui escrita, leitura, leitura aleatória e escrita aleatória, executados um logo após o outro, nessa ordem.

Apesar de pequenas variações, não é possível observar um padrão de comportamento ao longo da execução. Apesar de alguns testes (como os apresentados para HDDs) possuírem uma variabilidade maior no início da execução, isso não se confirma como um padrão comparando com os demais resultados. Em outras palavras, o tipo de teste (leitura, escrita, leitura aleatória ou escrita aleatória) não teve efeito na demanda de potência observada nos experimentos deste artigo. Portanto, **o padrão de acesso (tipo de teste e tamanho das requisições) não parece impactar a demanda de potência do sistema.**

É importante notar que não causar diferença na demanda de potência não significa não impactar o consumo energético. O padrão de acesso impacta o tempo de execução e, portanto, a energia total consumida durante a execução, a qual é discutida na próxima seção.

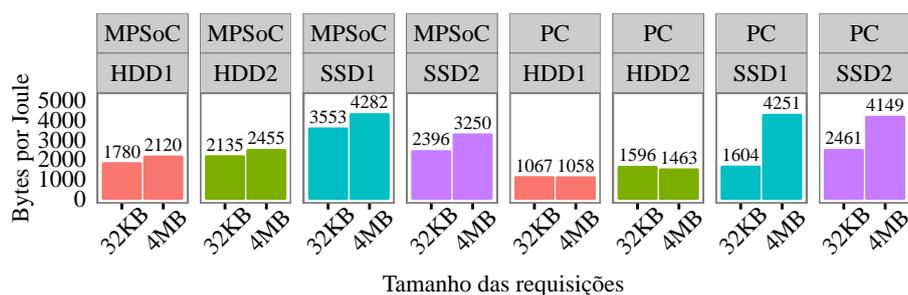
4.2. Eficiência energética

A energia consumida em Joules (J) por um experimento é obtida multiplicando-se a potência média (W) pelo tempo de execução (segundos). Portanto, além do impacto causado pelos padrões de acesso por causa das diferenças nos tempos de execução, espera-se encontrar diferenças no consumo de energia para os mesmos testes entre equipamentos e dispositivos diferentes. Isso acontece porque esses ambientes possuem diferentes demandas de potência.

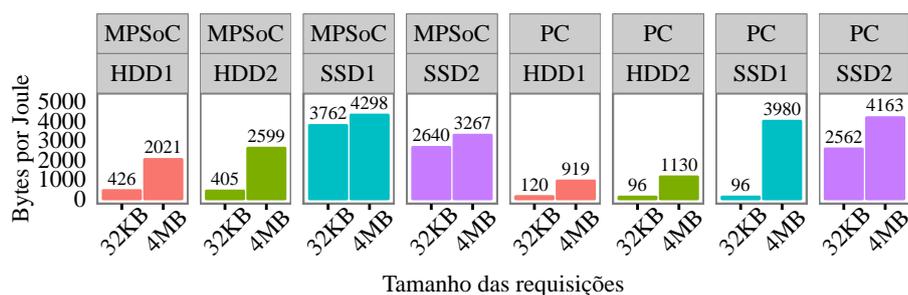
A diferença entre a demanda de potência nos ambientes testados pode ser verificada na Figura 1. A Tabela 3 apresenta valores de potência média obtidos em *idle*, ou seja, sem acessar os dispositivos de armazenamento, e durante os testes (média dos valores obtidos

Tabela 3. Potência média em *idle* e durante testes.

	MPSoC				PC			
	HDD1	HDD2	SSD1	SSD2	HDD1	HDD2	SSD1	SSD2
<i>Idle</i>	13,9	10,4	5,3	6,0	56,4	55,5	54,3	54,8
Experimento	16,1	13,5	8,6	10,6	57,8	56,9	58,8	58,9



(a) Resultados dos testes de **escrita sequencial**



(b) Resultados dos testes de **escrita aleatória**

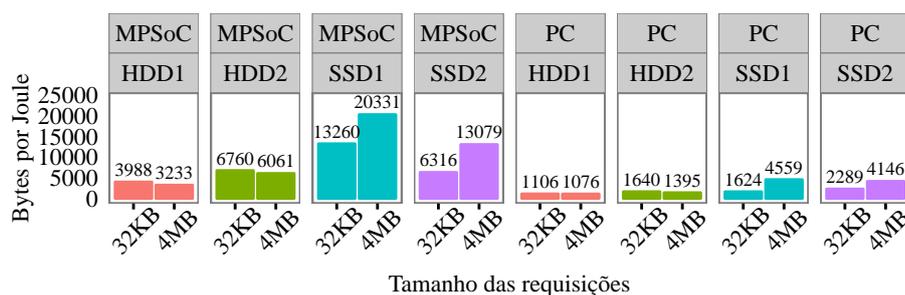
Figura 3. Eficiência energética, em bytes por Joule, observada durante os experimentos de escrita sem o uso da cache.

com requisições de 32 KB e 4 MB). Ao acessar os HDDs e SSDs no PC, a demanda de potência aumenta 3,7% e 6,4% respectivamente. No MPSoC, a potência aumenta 2,6% e 68,4% ao acessar HDDs e SSDs. **Comparado ao PC, o MPSoC apresenta demanda de potência 74,2% menor com HDDs e 83,7% com SSDs.**

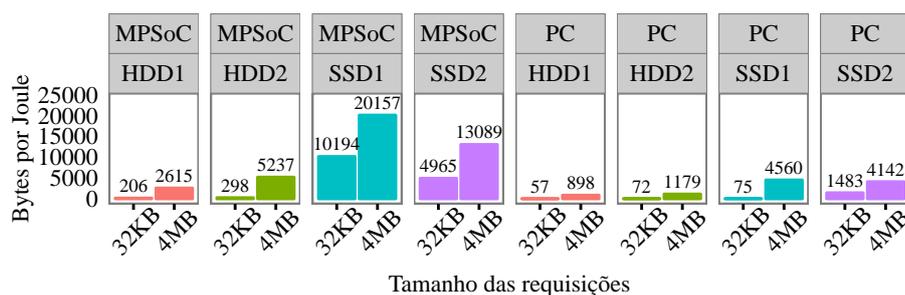
A fim de analisar a **eficiência energética** das diferentes configurações testadas, optou-se por utilizar a métrica **bytes por Joule**, dividindo-se a quantidade de dados escritos ou lidos durante cada teste (2 GB para todos os experimentos apresentados neste artigo) pela energia consumida por esse teste. Essa métrica permite uma comparação das configurações como servidores de armazenamento.

As Figuras 3 e 4 apresentam os resultados de eficiência energética (em bytes por Joule) de todos os experimentos realizados sem o uso da cache para escrita e leitura, respectivamente. Os resultados sem o uso da cache foram considerados os mais apropriados para essa discussão por representarem testes em que os acessos foram feitos diretamente aos dispositivos de armazenamento, ou seja, eles evidenciam o impacto do comportamento desses dispositivos. Cada figura é composta de duas partes, a primeira com os resultados para acesso sequencial aos arquivos, e a segunda para acesso a porções aleatórias. Em cada uma dessas partes, as primeiras quatro caixas apresentam os resultados obtidos no MPSoC com os quatro diferentes dispositivos de armazenamento e as últimas quatro caixas apresentam resultados obtidos no PC. Cada gráfico possui duas barras, a primeira para o resultado do teste com requisições de 32 KB e a segunda para requisições de 4 MB.

O teste de Kolmogorov-Smirnov [Lilliefors 1967] foi aplicado separadamente em cada um dos 128 experimentos para testar a normalidade da amostra de dez repetições das variáveis tempo de execução, banda e bytes por Joule. Os resultados mostraram que essas variáveis nos experimentos executados **não** seguem uma distribuição normal. Por esse mo-



(a) Resultados dos testes de **leitura sequencial**



(b) Resultados dos testes de **leitura aleatória**

Figura 4. Eficiência energética, em bytes por Joule, observada durante os experimentos de leitura sem o uso da cache.

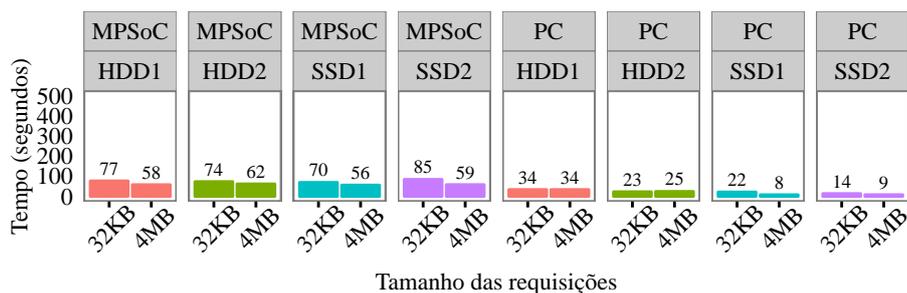
tivo, são apresentados nos gráficos as medianas. Além disso, pelo mesmo motivo, não são apresentadas barras de erro, pois o seu cálculo assume uma distribuição normal. O teste de Dunn [Dunn 1961] foi então utilizado para comparação de medianas.

Os resultados obtidos indicam que o **MPSoC provê maior eficiência energética do que o PC quando discos rígidos são utilizados para armazenamento** - eficiência até 2 vezes maior para escrita sequencial, 4,2 vezes para escrita aleatória, 4,3 vezes para leitura sequencial e 4,4 vezes para leitura aleatória. A diferença foi suportada pelo teste de medianas. Como a velocidade dos HDDs é menor do que dos SSDs, o acesso ao dispositivo é o que limita o tempo de execução e, por isso, o prejuízo no tempo de execução ao trocar o PC pelo MPSoC é pequeno o bastante para ser compensado pela menor demanda de potência. Os tempos de execução de todos os experimentos sem cache são apresentados na Figura 5.

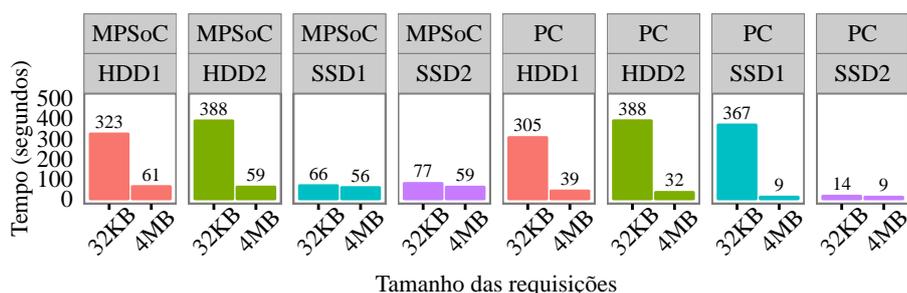
O uso do MPSoC também resultou em maior eficiência energética que o PC nos **testes de leitura nos SSDs** - até 8,4 vezes maior para leitura sequencial e 136 vezes para aleatória. Essa vantagem não pôde ser confirmada pelo teste estatístico apenas para o caso de leituras aleatórias de 32 KB no SSD2. Além disso, maior eficiência energética foi observada com o MPSoC do que com o PC para **escritas de 32 KB no SSD1** - até 39 vezes.

No entanto, **o PC ofereceu melhor eficiência que o MPSoC em testes de escrita com requisições de 4 MB no SSD2** - 1,3 vezes maior. Para os demais casos não discutidos - requisições de 32 KB no SSD2 e de 4 MB no SSD1 - não foi possível concluir através do teste de Dunn que as medianas sejam significativamente diferentes.

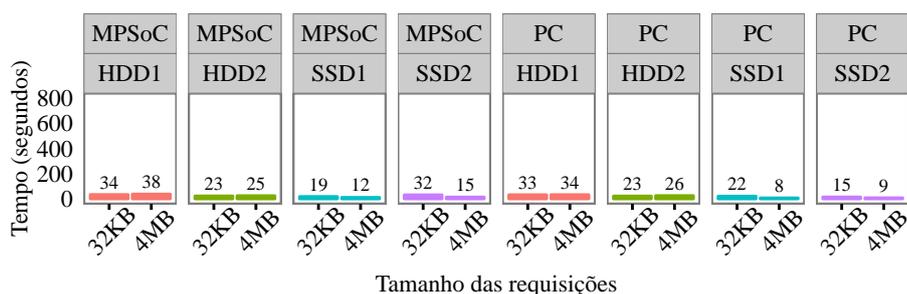
Comparando os resultados obtidos para os dois tamanhos de requisição em cada cenário, todas as diferenças exceto as operações sequenciais (escrita e leitura sequenciais) no PC com o SSD1 podem ser consideradas significativas. **Na maioria dos casos, usar**



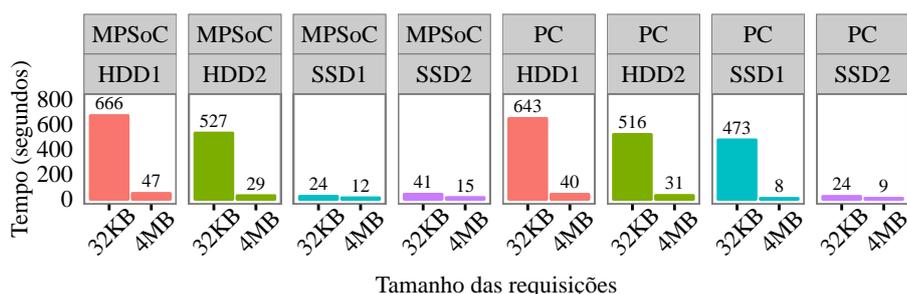
(a) Resultados dos testes de **escrita sequencial**



(b) Resultados dos testes de **escrita aleatória**



(c) Resultados dos testes de **leitura sequencial**



(d) Resultados dos testes de **leitura aleatória**

Figura 5. Tempo de execução dos experimentos sem o uso da cache.

requisições maiores para acessar a mesma quantidade de dados leva a maior eficiência energética - até 2,6 vezes maior para escrita sequencial, 41,5 vezes para escrita aleatória, 2,1 vezes para leitura sequencial e 60,8 vezes para leitura aleatória. Isso acontece porque o tempo de execução é reduzido, uma vez que, conforme discutido na Seção 4.1, não foi observada diferença significativa na demanda de potência desses testes. O tempo de execução reduzido pode ser verificado na Figura 5. O impacto do tamanho da requisição no tempo é maior para

requisições aleatórias porque aumentar o tamanho leva a geração de menos requisições que não são contíguas. Maior eficiência energética foi observada com requisições de 32 KB do que com 4 MB apenas para escrita e leitura sequenciais no PC com HDD2 e para leitura sequencial no MPSoC com HDDs.

Eventuais diferenças entre resultados obtidos acessando o arquivo sequencialmente ou aleatoriamente também são causadas por diferenças nos tempos de execução. Dentre os resultados obtidos, **acessos sequenciais levaram a maior eficiência energética para requisições de 32 KB no HDD2 e no HDD1** (neste exceto para escritas no MPSoC, onde a diferença não é suportada pelo teste estatístico) - até 16,7 vezes maior para escrita e 22,8 vezes para leitura. Tempos de acesso a discos rígidos são tipicamente mais afetados pela sequencialidade dos acessos, e esse impacto é mais importante quando mais requisições são realizadas (portanto para requisições menores). **Para os SSDs, isso ocorreu para escritas no PC com o SSD1 e para leituras no PC com ambos dispositivos.**

Considerando os resultados apresentados nesta seção, a substituição do PC pelo MPSoC como servidor de armazenamento hipotético dependeria das características da carga de trabalho esperada. Por exemplo:

- **Para escritas sequenciais, os resultados da configuração PC + HDD poderiam ser obtidos por três servidores MPSoC + SSD1.** Essa substituição diminuiria a demanda de potência em aproximadamente 55%.
- **Para escritas aleatórias, os resultados da configuração PC + HDD poderiam ser obtidos por dois servidores MPSoC + SSD1.** A diminuição na demanda de potência seria de aproximadamente 70%.
- **Para leituras sequenciais ou aleatórias, um único servidor MPSoC + SSD1** poderia substituir a configuração PC + HDD, demandando 85% menos potência. A configuração PC + SSD seria substituída por dois servidores MPSoC.

Discos rígidos são tipicamente utilizados para servidores de armazenamento por causa do maior custo por byte oferecido pelos SSDs. Isso significa que discos rígidos de preço similar a SSDs possuem maior capacidade de armazenamento. No entanto, ao aumentar o número de servidores substituindo PC por MPSoC, a capacidade de múltiplos SSDs é somada para substituir um HDD. Além disso, acredita-se que o menor custo financeiro de arquiteturas de baixa potência compense a compra de um número maior de dispositivos.

5. Conclusões e Trabalhos Futuros

Este artigo apresentou uma análise de desempenho e eficiência energética de uma arquitetura de baixa demanda de potência - um MPSoC usando um processador ARM - para operações de E/S. Esses resultados foram comparados com um computador tradicional a fim de estudar a viabilidade do uso de arquiteturas de baixa potência como servidores de armazenamento. Esse estudo incluiu diferentes padrões de acesso e dispositivos de armazenamento a fim de cobrir diferentes situações e características.

Os resultados mostraram que a demanda de potência não é afetada pelo padrão de acesso. No entanto, este possui um impacto no tempo de execução, o que afeta o consumo de energia. Foi observado que, utilizando discos rígidos para armazenamento, o uso do MPSoC leva a uma eficiência energética até 4,4 vezes maior do que o observado usando o PC. Para leitura utilizando SSDs, o MPSoC leva a uma eficiência energética até 136 vezes maior. Essa diferença acontece porque, apesar do maior tempo de execução medido nesse ambiente, a demanda de potência é até 6,7 vezes menor.

Concluiu-se que a substituição de um servidor de configuração PC + HDD por múltiplos servidores de baixa potência com SSDs seria viável e manteria um desempenho semelhante. Dependendo da carga de trabalho esperada, essa substituição diminuiria a demanda de potência, e conseqüentemente o consumo de energia, em até 85%.

Como trabalho futuro pretende-se expandir a análise apresentada a outros modelos de equipamentos de dispositivos de armazenamento. Além disso, serão avaliados ambientes distribuídos em que múltiplos servidores de baixa potência oferecem armazenamento.

Agradecimentos

Os autores gostariam de agradecer a **Aishameriane Schmidt**, da Universidade do Estado de Santa Catarina (UDESC), pela valiosa ajuda na análise estatística dos resultados.

Referências

- Amur, H., Cipar, J., e Gupta, V. (2010). Robust and Flexible Power-Proportional Storage. *Proceedings of the 1st ACM symposium on Cloud computing*, páginas 217–228.
- Blake, G., Dreslinski, R., e Mudge, T. (2009). A Survey of Multicore Processors. *IEEE Signal Proc. Magazine*, 26(6):26–37.
- Cardosa, M., Singh, A., Pucha, H., e Chandra, A. (2012). Exploiting spatio-temporal tradeoffs for energy-aware MapReduce in the cloud. *IEEE Transactions on Computers*, 61(12):1737–1751.
- Carrera, E. V., Pinheiro, E., e Bianchini, R. (2003). Conserving disk energy in network servers. *Proceedings of the 17th annual international conference on Supercomputing - ICS '03*, página 86.
- Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64.
- Ge, R., Feng, X., e Sun, X. H. (2012). SERA-IO: Integrating energy consciousness into parallel I/O middleware. Em *Proceedings - 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012*, páginas 204–211.
- Gurumurthi, S., Sivasubramaniam, A., Kandemir, M., e Franke, H. (2003). DRPM: dynamic speed control for power management in server class disks. *30th Annual International Symposium on Computer Architecture, 2003. Proceedings*.
- Hosseinimotlagh, S., Khunjush, F., e Hosseinimotlagh, S. (2014). A cooperative two-tier energy-aware scheduling for real-time tasks in computing clouds. Em *Proceedings...*, páginas 178–182. Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), IEEE.
- Ibrahim, S., Phan, T.-D., Carpen-Amarie, A., Chihoub, H.-E., Moise, D., e Antoniu, G. (2016). Governing energy consumption in hadoop through cpu frequency scaling: An analysis. *Future Generation Computer Systems*, 54:219–232.
- Kim, J., Chou, J., e Rotem, D. (2011). Energy proportionality and performance in data parallel computing clusters. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6809 LNCS:414–431.

- Kogge, P., Bergman, K., Borkar, S., Campbell, D., Carson, W., Dally, W., Denneau, M., Franzon, P., Harrod, W., Hill, K., et al. (2008). Exascale computing study: Technology challenges in achieving exascale systems. páginas 1–297.
- Lee, W. Y. (2009). Energy-saving dvfs scheduling of multiple periodic real-time tasks on multi-core processors. Em *Proceedings...*, páginas 216–223. International Symposium on Distributed Simulation and Real Time Applications, IEEE Computer Society.
- Lilliefors, H. W. (1967). On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402.
- Lim, M. Y., Freeh, V. W., e Lowenthal, D. K. (2006). Adaptive, transparent frequency and voltage scaling of communication phases in MPI programs. Em *Proceedings...*, páginas 14–14. Supercomputers Conference (SC), ACM/IEEE.
- Nijim, M., Manzanares, A., Ruan, X., e Qin, X. (2009). Hybud: An energy-efficient architecture for hybrid parallel disk systems. *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, 0845257(2005).
- Padoin, E. L., Pilla, L. L., Boito, F. Z., Kassick, R. V., Velho, P., e Navaux, P. O. A. (2013). Evaluating application performance and energy consumption on hybrid CPU+GPU architecture. *Cluster Computing*, 16(3):511–525.
- Padoin, E. L., Pilla, L. L., Castro, M., Boito, F. Z., Navaux, P. O. A., e Mehaut, J.-F. (2014). Performance/energy trade-off in scientific computing: The case of ARM big.LITTLE and Intel Sandy Bridge. *IET Computers & Digital Techniques*, 2(3):1–14.
- Peraza, J., Tiwari, A., Laurenzano, M., Carrington, L., e Snavely, A. (2013). PMaC's green queue: a framework for selecting energy optimal DVFS configurations in large scale MPI applications. *Concurrency and Computation: Practice and Experience*, páginas 1–20.
- Saito, T., Sato, K., Sato, H., e Matsuoka, S. (2013). Energy-aware I/O optimization for checkpoint and restart on a NAND flash memory system. *Proceedings of the 3rd Workshop on Fault-tolerance for HPC at extreme scale - FTXS '13*, página 41.
- Shang, P. e Wang, J. (2011). A novel power management for CMP systems in data-intensive environment. *Proceedings - 25th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2011*, páginas 92–103.
- Welch, B. e Noer, G. (2013). Optimizing a hybrid SSD/HDD HPC storage system based on file size distributions. Em *2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST)*, páginas 1–12.
- Younge, A., von Laszewski, G., Wang, L., Lopez-Alarcon, S., e Carithers, W. (2010). Efficient resource management for cloud computing environments. Em *International Conference on Green Computing*, páginas 357–364. IEEE.
- Zhu, Q., Chen, Z., Tan, L., Zhou, Y., Keeton, K., e Wilkes, J. (2005). Hibernator: helping disk arrays sleep through the winter. *Proceedings of the twentieth ACM symposium on Operating systems principles - SOSP '05*, 39:177.