Contribuições para a otimização de uma plataforma de virtualização

José Luís Miranda Gonçalves

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Informática.

Trabalho orientado por:

José Carlos Rufino Amaro

.

Bragança

2022-2023

Contribuições para a otimização de uma plataforma de virtualização

José Luís Miranda Gonçalves

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Informática.

Trabalho orientado por:

José Carlos Rufino Amaro

Bragança

2022-2023

Dedicatória

Dedico este trabalho à minha família, em especial à minha mulher e às minhas filhas pelo apoio que me deram e pela paciência que tiveram comigo quando a realização deste trabalho implicou que atividades familiares fossem postas em segundo plano.

Agradecimentos

Em primeiro lugar agradecer à minha mulher, pelo apoio e incentivo que sempre me deu e pela maneira como "segurou as pontas" do ponto de vista familiar, sempre que a realização deste trabalho implicava a minha ausência.

Agradeço também ao Prof. José Carlos Rufino Amaro, pela orientação, empenho e disponibilidade na realização deste trabalho.

Resumo

Este trabalho visa contribuir com metodologias de análise de desempenho assim como recomendação de boas práticas, a aplicar em plataformas de virtualização. De forma a ilustrar a análise feita é usado o cluster do CeDRI como caso de estudo.

Numa primeira fase foram identificados os pontos que mereceriam mais atenção e onde poderiam estar os maiores ganhos de desempenho. Posteriormente, foi definida uma metodologia de medição de desempenho dos componentes objeto de estudo.

O estudo da topologia de rede, dos seus componentes e das opções de armazenamento mereceram a maior atenção. Observou-se uma discrepância entre os valores nominais dos equipamentos e ou valores de facto medidos dos equipamentos, assim como uma influência significativa do sistema operativo escolhido, na performance de rede.

A utilização de armazenamento local, por contrapartida ao armazenamento partilhado, oferece ganhos de desempenho mais significativos do que inicialmente seria de esperar.

Da análise feita resultou que a opção pela mudança de plataforma de virtualização, permitiria utilizar armazenamento local assim como armazenamento partilhado.

Assim, depois da análise ao cluster do CeDRI, como caso de estudo, foi recomendada a mudança da plataforma de virtualização, pelas razões enumeradas ao longo do trabalho.

Palavras-chave: Virtualização, Armazenamento, Rede, Desempenho.

Abstract

This work aims to contribute with performance analysis methodologies as well as recommendation of good practices, to be applied in virtualization platforms. In order to to illustrate the analysis performed, the CeDRI cluster is used as a case study.

In a first step, the issues that could yield more performance gains were identified. Subsequently, a methodology for measuring the performance of the components under study was defined.

The study of network topology, its components and storage options deserved the most attention. There was a discrepancy between the nominal values of equipment and or the measured values in the analysis made. The software options also had a bigger impact than iniatialy expected.

The use of local storage, as opposed to shared storage, offers more significant performance gains than initially could be anticipated.

From the analysis carried out, resulted that the option to change the virtualization platform, would allow using local storage as well as shared storage.

Thus, after analyzing the CeDRI cluster, as a case study, it was recommended the change of the virtualization platform, for the reasons enumerated throughout the work.

Keywords: Virtualization, Storage, Networking, Performance.

Conteúdo

1 Introdução **1**

1.1 Enquadramento 1

1.2 Objetivos 3

1.3 Estrutura do Documento 3

2 Contexto e Tecnologias **5**

2.1 Preâmbulo 5

2.2 Motivação 5

2.3 Áreas de Intervenção 6

2.4 Plataforma de Virtualização 7

2.4.1 Situação Atual 7

2.4.2 Possíveis Alternativas 7

2.4.2.1 Proxmox 7

2.4.2.2 XCP-ng 8

2.5 Topologia e Tecnologias de Rede 8

2.5.1 Situação Atual 8

2.5.2 Possíveis Alternativas 8

2.6 Tecnologias de Armazenamento 9

2.6.1 Situação Atual 9

2.6.2 Possíveis Alternativas 9

2.6.2.1 Hardware 9

2.6.2.2 Sistema de Ficheiros 10
2.6.2.3 Sistema Operativo 10
2.6.2.4 Armazenamento Remoto vs Armazenameno Local 11
2.6.2.5 NVMe over Fabrics 11
2.6.2.6 Armazenamento Centralizado vs Desagregado vs Hyper-converged 11
2.7 Epílogo 12

3 Otimização da Conexão Edge **13**
3.1 Preâmbulo 13
3.2 Contexto 13
3.3 Ponto de Partida e Objeto de Estudo 15
3.4 Infraestrutura de Testes 16
3.5 Metodologia de Teste 18
3.6 Testes com NAT via pfSense 19
3.6.1 Taxas de Transferência e Cargas de CPU Globais 19
3.6.2 Taxa de Transferência em Função do Paralelismo 22
3.6.3 Cargas de CPU em Função do Paralelismo 23
3.7 Sobrecarga introduzido pelo NAT 25
3.8 Outros Testes 27
3.9 Conclusões 29
3.10 Epílogo 31

4 Otimização do Armazenamento **33**
4.1 Preâmbulo 33
4.2 Introdução 34
4.3 Bancada de Teste 34
4.4 Armazenamento Local RAW 35
4.4.1 IOPS de Escrita e Leitura 36
4.4.2 Latência de Escrita e Leitura 39

4.4.3 Definição do Teste Padrão 41
4.5 Armazenamento Local Formatado 42
4.6 Armazenamento Remoto via NFS 43
4.6.1 Análise Preliminar da Rede 46
4.6.2 Desempenho da Leitura 46
4.6.3 Desempenho da Escrita 48
4.6.4 Comparação EXT4 vs ZFS 49
4.7 Armazenamento Remoto via NVMe-over-Fabrics 51
4.7.1 Leitura RAW 51
4.7.2 Escrita RAW 54
4.8 Outros Testes 54
4.9 Conclusões 55
4.10 Epílogo 58

5 Alterações Preconizadas **59**
5.1 Preâmbulo 59
5.2 Solução preconizada 59
5.2.1 Computação 59
5.2.2 Topologia de Rede 60
5.2.3 Armazenamento 61
5.2.4 Plataforma de Virtualização 61
5.3 Epílogo 62

6 Conclusões **63**
6.1 Topologia de Rede 63
6.2 Armazenamento 65
6.3 Alterações Preconizadas 66
6.4 Trabalho futuro 66

A Otimização da Firewall Edge - Medições em Detalhe **A1**

A.1 Impacto do Multithreading na Firewall Física vs Virtualizada A1

B Código **B1**

B.1 Script para realização de testes de rede B1

Bibliografia **B1**

Lista de Tabelas

3.1 Taxa de Transferência (Gbps) e Cargas de CPU com pfSense nativo. 19
3.2 Taxa de Transferência (Gbps) e Cargas de CPU com pfSense virtual. . . . 20
3.3 Taxa de Transferência (Gbps) com pfSense nativo e virtualizado 22
3.4 Cargas de CPU do cliente iPerf - 1,2,4,8 Threads iPerf. 24
3.5 Cargas de CPU do servidor iPerf - 1,2,4,8 Threads iPerf. 24
3.6 Cargas de CPU da firewall - 1,2,4,8 Threads iPerf. 24
3.7 Medição ponto-a-ponto entre i7Server e i7Client 26
3.8 Medição ponto-a-ponto entre pfSense virtual e i7Client nativo 27
3.9 Medição ponto-a-ponto entre VyOS virtual e i7Client nativo 28
3.10 Medição ponto-a-ponto entre Ubuntu virtual e i7Client nativo 28

4.1 Desempenho da ligação entre cliente e servidor NFS (Gbps) 46
4.2 IOPS (x1000) no acesso remoto 50
4.3 Latência (ms) no acesso remoto 51

A.1 pfSense Virtual sem SMT A1
A.2 pfSense Virtual com SMT A2
A.3 pfSense Físico sem SMT A2
A.4 pfSense Físico com SMT A2

Lista de Figuras

3.1 Arquitetura de Rede do *cluster* do CeDRI. 14

3.2 Topologia da Infraestrutura de Testes da Firewall Edge. 16

4.1 IOPS de escrita em armazenamento local RAW com RAID 0 37

4.2 IOPS de leitura em armazenamento local RAW com RAID 0 38

4.3 Latência de escrita em armazenamento local RAW com RAID 0 39

4.4 Latência de leitura em armazenamento local RAW com RAID 0 40

4.5 Leitura: IOPS e Latência em array de 12 Discos 44

4.6 Escrita: IOPS e Latência em array de 12 Discos 45

4.7 IOPS e latência em leitura de ZFS em acesso remoto 47

4.8 IOPS e latência em leitura de EXT4 em acesso remoto 48

4.9 IOPS e latência em escrita de ZFS em acesso remoto 49

4.10 IOPS e latência em escrita de EXT4 em acesso remoto 50

4.11 NVMe: IOPS e latência em leitura de 1 SSD 52

4.12 NVMe: IOPS e latência em escrita de 1 SSD 53

4.13 Desempenho de Disco Virtual de SAN 56

4.14 Comparação 8 Discos SATA vs 1 NVME - Leitura 56

4.15 Comparação 8 Discos SATA vs 1 NVME - Escrita 56

Capítulo 1

Introdução

A Virtualização de Sistemas [13] é hoje em dia parte das fundações das Tecnologias de Informação e Comunicação, nomeadamente do modelo de Computação em Nuvem [11] [13] (onde implementa a camada IaaS). Todavia, há ainda muitos contextos em que as organizações optam por operar *on-premises* plataformas tradicionais de virtualização, sendo então confrontadas com a necessidade de resolver as questões ligadas à implementação e otimização dessas plataformas. Essas questões envolvem uma série de ponderações quanto aos diferentes componentes da pilha de hardware e software da plataforma de virtualização. Por seu turno, esses componentes ou blocos construtores não são estáticos, evoluindo por via da adoção de novas tecnologias e em resposta a novas demandas dos seus utilizadores.

1.1 Enquadramento

Quando se parte para a implementação de um plataforma de virtualização, isso implica, à partida, que já foi feito todo um trabalho de levantamento de requisitos, necessidades (atuais e futuras), estimativas orçamentais, requisitos em termos de recursos humanos para a sua gestão, entre outros. Encontrar uma solução de virtualização que corresponda à exata medida das necessidades de determinada entidade, não é portanto, tarefa fácil.

Assim, é necessário um compromisso, um exercício de balanceamento entre as diferentes premissas que o levantamento de requisitos fez efervescer ao topo da lista. No processo de escolha de uma plataforma de virtualização, entre todos os fatores a ter em conta e todas as decisões a tomar há uma decisão que se revela de suma importância. Essa decisão tem a ver com a opção por uma plataforma do tipo "chave na mão" fornecida apenas por um único fornecedor ou por uma plataforma personalizada, de acordo com as nossas necessidades específicas, havendo diferentes implicações que se podem associar quer a uma quer a outra opção.

Por exemplo, pode-se apontar a fiabilidade, desempenho e custo elevado como características tipicamente associadas a soluções do tipo "chave na mão", e flexibilidade, capacidade de expansão e menor custo como características associadas a soluções à medida que recorrem a componentes e tecnologias de diferentes fornecedores. Por outro lado, soluções fornecidas por um só vendedor normalmente são devidamente testadas, e o desempenho expectável é bem conhecido, as diferentes métricas de desempenho estão bem identificas e perceber se a solução fornecida corresponde ao que foi contratualizado é tarefa mais tangível, mais linear. Pelo contrario, numa solução personalizada, que eventualmente nunca foi configurada naqueles exatos moldes, com aqueles exatos componentes, implementados/configurados daquela exata maneira, é mais difícil de perceber se tal solução está a atingir o seu máximo potencial.

No caso da adoção de soluções personalizadas, o planeamento e definição da arquitetura da solução será sempre feito com base em informação técnica disponível dos seus componentes individuais, de experiências e casos eventualmente disponíveis na Internet e de uso de outras fontes de informação. A verdade, porém, é de que é impossível, *a priori*, determinar com exatidão o nível de desempenho que determinada solução irá oferecer.

Outro dos fatores que poderá condicionar a eficiência da solução escolhida é o tipo e características da carga de trabalho solicitada à infraestrutura de virtualização. A evolução das cargas de trabalho poderá ditar uma reconfiguração ou revalidação da arquitetura. Dado o carácter mutável deste tipo de plataforma, a avaliação do seu desempenho servirá para potenciar e maximizar o seu aproveitamento.

1.2 Objetivos

Sendo as plataformas de virtualização compostas por um elevando número de componentes de software e hardware, torna-se pouco prático, e sobretudo pouco eficiente, fazer um levantamento e análise de todos os seus componentes de forma igualitária. Assim, torna-se mais objetivo realizar um estudo mais aprofundado sobre áreas que apresentam uma variabilidade maior, quer em termos de desenho da arquitetura, quer na potencial contribuição para o desempenho global da plataforma de virtualização. Assim, é proposto um estudo mais aprofundado do desempenho das diferentes opções em termos de armazenamento e em termos de topologia e tecnologia de rede. A análise a fazer estará certamente condicionada ao hardware disponível para levar a cabo os testes de desempenho necessários. No entanto, essas análises devem fornecer uma perspetiva e uma quantificação dos diferentes vetores de desempenho, de forma a que seja claro quais as implicações para o desempenho global da plataforma.

Esta dissertação pretende contribuir para a melhoria das condições de operação de uma plataforma de virtualização pré-existente. Essa melhoria deverá traduzir-se numa utilização mais eficiente do hardware disponível, num incremento dos níveis de fiabilidade da infra-estrutura e numa melhor experiência de utilização. Para o efeito será feita uma análise à plataforma atual em vertentes consideradas fundamentais, e propostas mudanças que deverão traduzir-se nas melhorias pretendidas.

1.3 Estrutura do Documento

O resto deste documento está organizado da seguinte forma:

Capítulo 2 - Contexto e Motivação: Abordam-se as principais tecnologias envolvidas nas plataformas de virtualização mais representativas na indústria (quais as opções em cima da mesa na hora de definir a arquitetura de virtualização).

Capítulo 3 - Otimização da Conexão Edge: avalia-se a eficiência da firewall assim como outros parâmetros de rede.

Capítulo 4 - Soluções de Armazenamento: avalia-se o desempenho das possíveis alternativas para o sistema de armazenamento.

Capítulo 5 - Alterações Preconizadas: apresenta-se as melhores opções de evolução do *cluster* de acordo com os dados obtidos nos capítulos anteriores.

Capítulo 6 - Conclusões: são apresentadas as conclusões e possíveis vias de trabalho futuro.

Capítulo 2

Contexto e Tecnologias

2.1 Preâmbulo

Ao longo deste capítulo serão introduzidas algumas das condicionantes e motivações para a realização deste trabalho. Serão apresentados alguns dos pontos de partida das questões que se pretende ver respondidas com a realização deste estudo sobre otimização de plataformas de virtualização e a sua aplicação a um caso em particular.

2.2 Motivação

A génese do trabalho descrito neste documento derivou da experiência de utilização e administração do *cluster* do CeDRI (Research Centre in Digitalization and Intelligent Robotics). Apesar de as várias métricas analisadas apontarem para não haver nenhum gargalo significativo nos diferentes componentes do *cluster*, o seu desempenho não correspondia às expectativas inicialmente criadas, face à carga de trabalho alocada e a capacidade do hardware instalado. No entanto, sendo uma solução personalizada, não haveria à partida um ponto de referência do que seria expectável em termos de desempenho.

Foram várias as potenciais razões inicialmente apontadas do que poderia estar na origem da discrepância entre a perspetiva inicial de qual seria o desempenho obtido face ao que na prática e de forma empírica era observado por administradores e utilizadores

do *cluster*. Mas, todas as razões apontadas não passavam de exercícios de especulação e conjeturas, fundamentadas em experiências passadas, e não validadas de forma científica. Por uma lado, não havia uma plataforma minimamente idêntica da qual houvesse dados sobre o desempenho e que se pudessem usar esses dados para comparar métricas de desempenho. Por outro lado, não se sabia onde estaria o limite do que seria possível com o hardware disponível, sendo que o *cluster* até poderia estar de facto configurado na sua forma ótima. Perante estas incertezas tornava-se imperativo obter respostas para as diferentes questões levantadas e fundamentar essas respostas de forma quantitativa.

2.3 Áreas de Intervenção

Conceptualmente, pode-se dividir o *cluster* em três grandes áreas: Rede, Armazenamento e Virtualização. Neste capítulo serão abordadas quais as questões e opções equacionadas e ás quais se tenta dar resposta ao longo dos próximos capítulos.

Em termos das escolhas de software a utilizar, estas prendem-se sobretudo com a questão da plataforma de virtualização em si, e com a stack a utilizar no servidor de armazenamento partilhado. Por uma questão de filosofia académica assim como por questões financeiras a opção por uma solução proprietária ficaria logo à partida posta de parte. Assim, vamos analisar as plataformas *open-source* que cumprem estes critérios e enumerar os principais pontos a favor e contra, sobre o ponto de vista de evolução do *cluster* tanto no que toca a desempenho como no que toca a funcionalidades.

Algumas das questões relativas ao caminho a seguir em termos de evolução do *cluster* em apreço, ou na definição da arquitetura de outros *clusters*, de características similares, são a seguir abordadas de forma a definir o caminho em termos de análise e comparação.

2.4 Plataforma de Virtualização

2.4.1 Situação Atual

O oVirt [10] é a plataforma de virtualização a ser executada neste momento. Como pontos a favor poderia-se apontar a facilidade de gestão de máquinas virtuais, o conhecimento da sua gestão e funcionamento entretanto adquirido e o portal self-service que a plataforma fornece. Como aspetos negativos há a destacar o facto de estar muito dependente do armazenamento partilhado (embora seja possível usar armazenamento local o uso dos dois tipos em simultâneo no mesmo nó é impossível), e a ausência de funcionalidade nativa de backup. Recentemente, a descontinuação do suporte à plataforma, por parte da empresa que a criou (RedHat), veio reforçar a necessidade de equacionar alternativas.

2.4.2 Possíveis Alternativas

2.4.2.1 Proxmox

O Proxmox [12] é uma alternativa possível, apresenta como principal vantagem o facto de se poder executar no mesmo nó máquinas virtuais assentes em armazenamento partilhado e armazenamento local. Acresce a facilidade com que se movem as máquinas virtuais entre os diferentes tipos de armazenamento, facilitando a otimização da distribuição da carga ao nível do armazenamento, à custa de uma redução no grau de automatização de tarefas.

Outro ponto a favor do Proxmox é a possibilidade de se fazer backups automáticos através de um escalonador nativo, ou de forma mais otimizada através da utilização do Proxmox Backup Server.

Como principais pontos negativos, tem-se o facto de não ser uma plataforma familiar, requerer intervenção da parte dos administradores do *cluster* em algumas tarefas e não ter um portal self-service.

2.4.2.2 XCP-ng

O XCP-ng [16] poderia ser uma alternativa ao oVirt, oferece apenas armazenamento partilhado à imagem do oVirt, no entanto oferece um sistema de backups muito robusto (ausente no oVirt). Em termos de desvantagens, apresentar um único ponto de falha devido ao armazenamento partilhado, não suportando armazenamento local.

2.5 Topologia e Tecnologias de Rede

2.5.1 Situação Atual

A rede está assente num switch 10Gbps, estando os nós do *cluster* ligados todos por interfaces 10Gbps. O switch é no entanto partilhado com outros serviços da infraestrutura, havendo diferentes vlans para cada secção da rede. Todas as interfaces de rede estão configuradas com IPv4 e MTU 1500.

2.5.2 Possíveis Alternativas

No que diz respeito às tecnologias de rede, pode-se analisar o impacto que a migração uma rede 10Gbps para uma rede 100Gbps terá, sobretudo no que toca à eficiência do armazenamento partilhado, assim como a introdução de novas tecnologias, nomeadamente a introdução de interfaces de rede com suporte para RDMA [15] e qual a alavancagem que esta tecnologia poderá emprestar ao *cluster*. Nesse sentido, além do estudo da diferença entre uma rede 10Gbps e 100Gbps, de forma a verificar até que ponto há uma correlação entre o valor nominal dos ativos de rede e o desempenho real e mensurável que se pode obter com uma ou outra tecnologia. Outras questões, como qual o grau de impacto que a opção por MTU 1500 ou MTU 9000 tem no *cluster* como um todo, também devem ser analisadas de forma a se perceber qual o impacto que uma ou outra opção, tem no desempenho global do *cluster*.

2.6 Tecnologias de Armazenamento

Este será talvez o ponto onde mais trabalho há a fazer, por duas razões: a primeira, o facto de estar a ser usado armazenamento partilhado implicar que as melhorias passiveis de ser introduzidas neste domínio afetarão transversalmente todas as máquinas virtuais do *cluster*, e a segunda, o facto de o armazenamento, seja de que tipo for, condiciona de forma muito significativa o desempenho de qualquer máquina virtual.

Será então pertinente equacionar as opções disponíveis e que de alguma forma poderiam contribuir para a melhoria do desempenho do *cluster* de uma forma significativa.

2.6.1 Situação Atual

A situação atual contempla o uso de uma única máquina com armazenamento partilhado, com o sistema operativo Truenas Core 12.0-U4[1], um processador Intel Xeon Silver 4112, 128GB de RAM, 12 Discos de 8TB SAS III e um SSD Intel Optane 900P de 280GB a servir de cache para o *array* de 12 discos agregados em raid-z2. O servidor liga a um switch 10Gbps por duas interfaces de 10Gbps a operar em round robin.

2.6.2 Possíveis Alternativas

2.6.2.1 Hardware

Em termos de hardware a adicionar ao *cluster*, poderemos equacionar a hipótese de adicionar interfaces de rede 100Gbps (por exemplo, Mellanox Connect-X 5). Naturalmente uma placa 100Gbps acrescentará desempenho ao sistema, mas como todas as outras opções no que toca a hardware, é necessário quantificar qual seria a melhoria real que se poderá implementar se de facto for essa a opção, uma vez que o custo associado é considerável, pois é preciso contemplar um switch 100Gbps assim como a respetiva cablagem.

O eventual upgrade poderá potenciar de forma mais significativa o desempenho no

[1] Ver `https://www.truenas.com/truenas-core`.

acesso ao servidor de armazenamento partilhado. Na eventualidade de querer usar protocolos baseados em RDMA [15], o uso de interfaces de rede com suporte para esses protocolos é absolutamente necessário.

Outra das opções a equacionar seria a substituição do atual servidor de armazenamento partilhado descrito anteriormente, por um servidor baseado em discos NVMe. Neste caso a migração para um rede 100Gbps seria praticamente obrigatória dado o diferencial de desempenho entre os atuais discos mecânicos e os discos NVMe. Esta assunção deve, no entanto, ser corroborada por medições concretas que confirmem essa assunção.

2.6.2.2 Sistema de Ficheiros

Estando-se a usar atualmente ZFS [9],numa pool com discos mecânicos e rede 10G, parece não haver uma limitação, pelo menos uma limitação óbvia, no que toca ao facto de uma sistema de ficheiros ZFS não ser o ideal para armazenamento partilhado (principalmente se adicionarmos dispositivos de armazenamento mais rápidos como discos SSD ou NVMe). Apesar de toda a funcionalidade em termos de resiliência que um sistema de ficheiros ZFS oferece, em termos de desempenho não parece o sistema de ficheiros com melhor desempenho.

O uso de EXT4 [8] servirá para comparar e validar o potencial de desempenho do ZFS, sendo este um sistema de ficheiros mais simples e com melhor desempenho, pelo menos numa configuração com as opções por omissão.

No caso de o sistema de ficheiros EXT4 ser considerado o mais indicado, implicaria a mudança também de sistema operativo. Sendo o Truenas core, desenhado à volta das funcionalidades e características do ZFS, não seria uma opção viável se a opção por uso de EXT4 for a recomendada.

2.6.2.3 Sistema Operativo

A mudança de Truenas, que é baseado em FreeBSD, poderá ser uma opção em cima da mesa, nomeadamente para um ambiente Linux, com a transição para Linux, ganharíamos mais flexibilidade, poderíamos continuar a usar ZFS mas ficaríamos com a opção de usar

ext4 ou outro sistema de ficheiros. A desvantagem seria o facto perdermos muitas das funcionalidades e ferramentas de gestão em ambiente gráfico que o Truenas nos oferece.

2.6.2.4 Armazenamento Remoto vs Armazenameno Local

A opção por armazenamento remoto ou local está diretamente ligada à escolha da plataforma de virtualização. A única plataforma que oferece de forma transparente as duas opções é o Proxmox. O XCP-ng suporta apenas armazenamento remoto de raíz e o oVirt, embora permita o uso de armazenamento local, as limitações que essa escolha implicam tornam praticamente inviável a escolha do oVirt se quisermos usar armazenamento local.

O armazenamento partilhado poderá fornecer facilidade de gestão, mas o armazenamento local certamente facultará melhor desempenho. O peso a dar a estes dois fatores influenciará certamente a decisão.

2.6.2.5 NVMe over Fabrics

O protocolo NVMe over Fabrics [3], nas suas diversas variantes (ROCE, iWARP, TCP, FC) promete ser uma tecnologia que contribuirá para que o armazenamento partilhado dê um salto de desempenho significativo.

Será assim importante avaliar o que esta tecnologia poderá trazer em termos de desempenho para casos como aquele em apreço. Será necessário fazer uma avaliação da maturidade da tecnologia, quais as implicações da sua adoção, e naturalmente, quais os potencias ganhos de desempenho que se poderão obter.

2.6.2.6 Armazenamento Centralizado vs Desagregado vs Hyperconverged

Em termos de topologia da arquitetura de armazenamento a escolha da plataforma poderá condicionar a escolha da topologia ou vice-versa.

O oVirt está mais vocacionado para o uso de armazenamento centralizado. Suporta armazenamento desagregado, mas a flexibilidade da sua gestão ficaria muito limitada.

O XCP-ng armazenamento centralizado e desagregado, sem grandes perdas de funcionalidade quer num caso quer no outro.

O Proxmox poderá integrar as 3 modalidade de forma complementar, a adoção de uma arquitetura HyperConverged [1] (implementando um cluster ceph [14]) poderá ser o caminho que trará mais vantagens se o Proxmox for a opção escolhida.

2.7 Epílogo

Ficou assim definido o ponto de partida para o estudo realizado. Apesar do mesmo não poder ser alterado, os resultados e as conclusões que se forem sendo tiradas ao longo do trabalho podem suscitar novas questões, que serão devidamente abordadas.

Capítulo 3

Otimização da Conexão Edge

3.1 Preâmbulo

A infraestrutura de rede é um dos componentes críticos numa plataforma de virtualização, tendo uma influência determinante na disponibilidade e desempenho das suas funcionalidades, nomeadamente: gestão da plataforma, acesso dos servidores de virtualização a servidores de armazenamento partilhado, troca de tráfego entre servidores que combinam virtualização e armazenamento em cenários *hyper-converged*, troca de tráfego entre máquinas virtuais e destas com o exterior.

Neste capítulo o foco incide sobre o último destes aspetos, ou seja, a otimização da componente *edge* da plataforma em estudo.

3.2 Contexto

Em termos lógicos, a topologia de rede do *cluster* do CeDRI foi projetada para suportar o conceito de *mini-cluster*: conjunto de máquinas virtuais operando numa rede isolada, satisfazendo necessidades específicas (por exemplo, de um grupo de investigação, de um projeto de I&D, de uma disciplina, de um departamento, etc.). Na figura 3.1 pode-se ver a arquitetura genérica da rede do *cluster*, incluindo *mini-clusters* de máquinas virtuais.

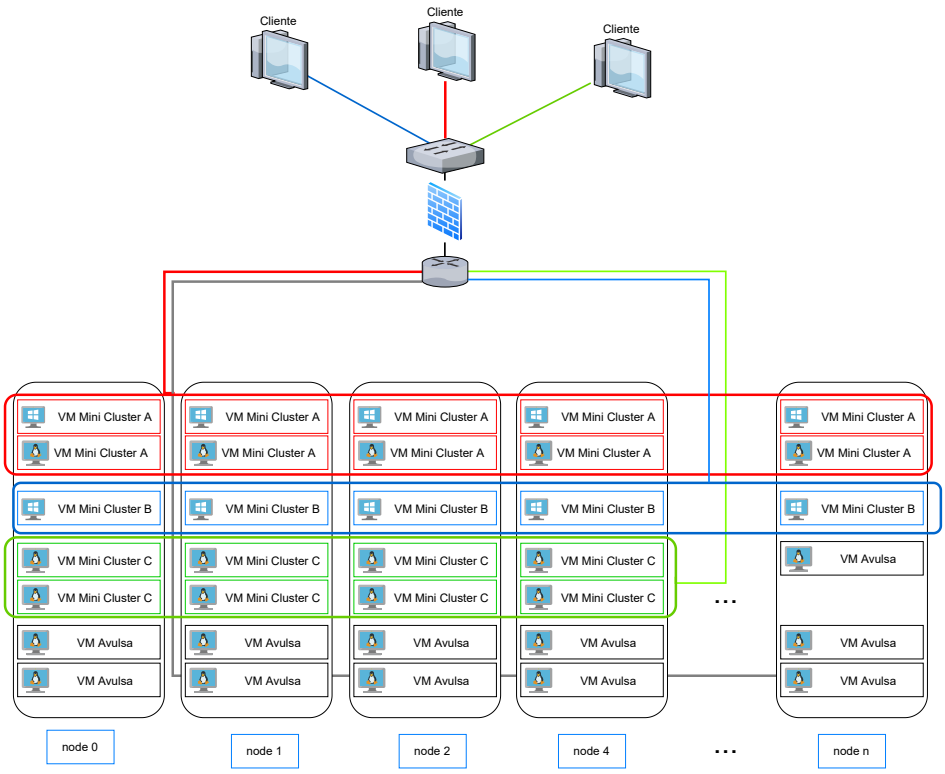


Figura 3.1: Arquitetura de Rede do *cluster* do CeDRI.

Cada *mini-cluster* opera numa *vlan* específica, limitando a visibilidade das suas máquinas virtuais a essa *vlan*. O acesso a estas máquinas, a partir do exterior, faz-se através de uma *firewall edge*, que providencia vários serviços a cada *mini-cluster*: acesso VPN, *leases* DHCP e gestão de nomes DNS. Além de *vlans* específicas de cada *mini-cluster*, existem ainda *vlans* dedicadas a serviços de gestão e de armazenamento do *cluster*, acessíveis apenas pelos servidores de virtualização e armazenamento e por consolas de gestão da plataforma de virtualização.

Num cenário deste género há múltiplas oportunidades de otimização da infraestrutura de rede: a) melhorar o desempenho da comunicação dos servidores de virtualização com os servidores de armazenamento partilhado, b) melhorar o desempenho da comunicação entre os próprios servidores de virtualização (beneficiando o desempenho da comunicação entre máquinas virtuais, ou melhorando as condições para cenários de armazenamento *hyper-converged*), c) melhorar o desempenho e a estabilidade da conexão do *cluster* ao

exterior através da sua *firewall edge*. Neste capítulo o foco incide sobre a vertente c).

De facto, sob o ponto de vista dos utilizadores do *cluster*, a estabilidade e desempenho da sua conexão ao exterior é o primeiro fator que condiciona a experiência da utilização, principalmente nas situações em que há utilização interativa (através de clientes de acesso remoto às máquinas virtuais) ou, mais raramente, quando é necessário transferir grandes quantidades de dados entre as máquinas virtuais e o exterior. Naturalmente, as mesmas questões (estabilidade e desempenho) também se colocam no funcionamento (execução) das máquinas virtuais, mas a qualidade da experiência a este nível é claramente afetada por questões mais ligadas à rede interna, e também com ramificações que ligam ao tipo de armazenamento subjacente, que serão alvo de estudo no Capítulo 4.

3.3 Ponto de Partida e Objeto de Estudo

À partida deste estudo, o serviço de *firewall edge* do *cluster* é providenciado por uma instância pfSense virtualizada, alojada num servidor de virtualização VMWare ESXi, assente numa máquina antiga de linha branca e sem qualquer facilidade de gestão remota de tipo IPMI. Entretanto, foi adquirida uma máquina da linha servidor, para os mesmos fins, colocando-se a questão de encontrar a melhor combinação de um conjunto de parâmetros por forma a otimizar o uso da nova máquina para estas funções. No âmbito deste estudo, os parâmetros considerados foram os seguintes:

P1: firewall física *versus* firewall virtualizada (sob VMWare ESXi [7]);

P2: hyperthreading/SMT ativo *versus* inativo;

P3: MTU 1500 *versus* MTU 9000;

P4: sobrecarga introduzida pelo processamento NAT na firewall;

P5: número de fluxos simultâneos de tráfego NAT.

P6: sistema operativo da firewall (pfSense [4] *versus* VyOS [6] *versus* Ubuntu [5])

A avaliação do impacto causada pela variação destes parâmetros faz-se pela medição da largura de banda utilizada e da carga observada nas CPUs dos sistemas envolvidos, durante a execução da ferramenta de *benchmark* de rede iPerf [2].

3.4 Infraestrutura de Testes

De forma a realizar os testes num ambiente perfeitamente controlado e isolado (não interferindo com a operação normal do *cluster*, nem sofrendo qualquer influência deste), foi montada uma cama de testes independente, com os seguintes sistemas computacionais:

- uma máquina para a firewall: servidor Supermicro AS-1014S-WTRT com um CPU AMD EPYC Zen2 7302P de 16 núcleos a 3.0/3.3GHz, 128GB de RAM DDR4-3200 (8 canais), e interface de rede Broadcom BCM57416 com duas portas 10GBase-T; o CPU desta máquina suporta 2 *hardware threads* por núcleo (total de 32 *hardware threads*), sendo para isso necessário ter a funcionalidade *Simultaneous Multithreading* (SMT) ativa na BIOS; esta funcionalidade é equivalente à tecnologia *Hyperthreading* da Intel, usando-se estas duas designações de forma indistinta neste documento;

- duas máquinas satélites (*end-point*), uma operando na rede interna criada pela firewall e outra na rede externa: máquinas de linha branca com placa mãe Asrock z97 pro4, CPU Intel i7 4790k a 4.0/4.4GHz, 32GB de RAM DDR3-1333, e interface de rede Intel X550 com duas portas 10Gbps.

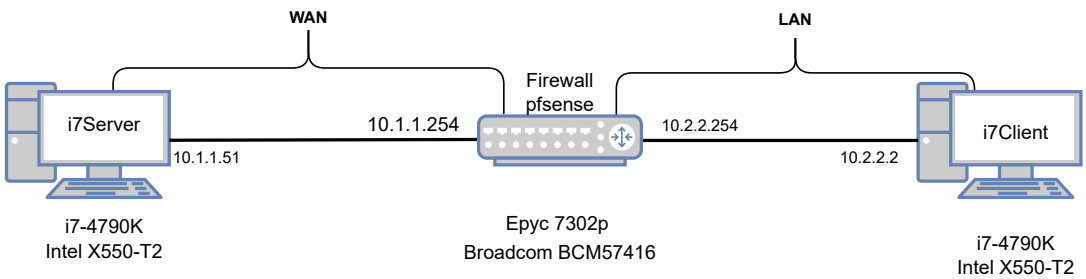


Figura 3.2: Topologia da Infraestrutura de Testes da Firewall Edge.

A figura 3.2 representa a topologia da infraestrutura de testes. Pode-se observar que a ligação das máquinas satélite à *firewall* é direta (sem recurso a um *switch*); este cenário de conexão é assumidamente artificial face ao que ocorre no *cluster* (onde todas conexões de rede passam por *switches*); no entanto, permite usar a capacidade máxima das ligações, o que é compatível com o objetivo mais geral do estudo, que implica perceber os limites máximos do desempenho sob NAT que se consegue extrair da *firewall*.

Como já referido, o servidor foi adquirido expressamente para alojar a *firewall edge* do *cluster*, pretendendo-se otimizar o uso dessa máquina para esses fins. As máquinas de linha branca, não sendo do mesmo tipo que os nós de virtualização atuais do *cluster*, possuem no entanto CPUs com frequências de relógio substancialmente mais elevadas (nalguns casos o dobro); desta forma, não tendo sido possível destacar nós do *cluster* para este cenário de testes, a expetativa inicial (que se veio a confirmar) era a de que usando apenas um par de máquinas com CPU de elevada frequência, isso fosse suficiente para saturar as conexões de rede testadas com a ferramenta iPerf. De referir ainda que as três máquinas tinham a sua BIOS atualizada e foram ativadas as opções garantes do maior desempenho (nomeadamente as de gestão do consumo de energia).

Na dimensão do *software*, foram usadas as seguintes ferramentas e ambientes:

- *firewalls*: pfSense (versão 2.5.1) e VyOS (versão 1.1.8); instaladas na máquina servidora, em ambiente *bare-metal* ou virtualizado, c.f. o cenário de teste;

- iPerf – versão 2.0.14a (em pfSense) e versão 2.0.13 (em Ubuntu); executado nas 3 máquinas, no papel de cliente ou servidor, em função do cenário de teste;

- *hypervisor*: VMWare ESXi 70U2; instalado na máquina servidora para os testes com as *firewalls* virtualizadas; a opção por este *hypervisor* decorre do facto ter sido o usado para alojar a *firewall edge* do *cluster* sendo que, por razões administrativas, pretende-se mantê-lo caso os testes concluam que não limita o desempenho;

- sistema operativo nas máquinas satélite: Ubuntu Server 20.04 LTS.

Antes dos testes paramétricos, despistaram-se eventuais gargalos de desempenho que pudessem afetar as máquinas satélite e condicionassem a saturação de uma ligação 10G direta. Assim, estas duas máquinas foram ligadas diretamente e foram feitos testes com a ferramenta iPerf, que comprovaram a capacidade de saturação da ligação 10G com apenas um par cliente-servidor executando o iPerf com um só fio-de-execução.

3.5 Metodologia de Teste

Para a realização dos testes, foi elaborada uma *script* em *bash* (ver Apêndice D), de forma a automatizar a sua execução em condições perfeitamente controladas. A *script* verifica, de 30s em 30s, a carga de CPU da máquina do servidor iPerf no último minuto[1], e quando esta carga desce a 0.0, despoleta um teste iPerf, que corre durante 60 segundos. Terminado este primeiro teste, o processo repete-se mais 4 vezes, de forma que são feitas 5 medições e obtido um valor médio, que será o considerado na análise que se segue. De referir ainda que estas 5 medições são realizadas para uma combinação particular de parâmetros, repetindo-se consoante o número de combinações paramétricas a testar.

No que toca à comparação da instanciação física com a virtualizada da *firewall*, numa primeira fase foi feita a instalação do pfSense 2.5.1 no servidor Supermicro e realizados os respetivos testes. Depois, foi feito um *backup* da configuração do pfSense, instalou-se o ESXi 7.0U2 no servidor, instanciou-se o pfSense em máquina virtual e importou-se o *backup* da configuração anterior, de forma a replicar, o mais fielmente possível, as condições e configurações da instância física do pfSense, na instância virtualizada.

Depois da realização dos testes e os resultados terem ficando um pouco aquém do esperado, foi também instanciada uma máquina virtual com Ubuntu 20.04 LTS e outra com vyOS 1.1 para tentar identificar potenciais constrangimentos provocados pela uso do pfSense, um sistema baseado em FreeBSD, que eventualmente não terá o mesmo grau de escrutínio em ambientes de virtualização e poder ser mais propenso a incompatibilidades.

[1]Recorde-se que comandos como `uptime`, `w` ou `top` disponibilizam 3 valores para a carga média: no último minuto, nos últimos 5 minutos, e nos últimos 15 minutos (trata-se de médias móveis exponenciais).

3.6 Testes com NAT via pfSense

Esta secção apresenta os resultados dos testes nas seguintes condições: o cliente iPerf executa na máquina i7Client, e troca tráfego com o servidor iPerf que executa na máquina i7Server, sendo que esse tráfego atravessa uma *firewall* pfSense onde sofre NAT.

São feitos testes com o pfSense nativo e virtual. Para cada uma destas variantes estuda-se o efeito da (des)ativação do SMT/Hyperthreading nas 3 máquinas em simultâneo, combinado com a variação do MTU (1500 vs 9000) nas mesmas máquinas, e variando ainda o número de *threads* usados pelo par cliente-servidor do iPerf (1, 2, 4 e 8 *threads*).

3.6.1 Taxas de Transferência e Cargas de CPU Globais

As tabelas 3.1 e 3.2 apresentam resultados globais com o pfSense nativo e virtual, respetivamente. Em cada tabela, os valores das colunas Não e Sim (relativas ao uso do *hyperthreading*) são valores médios dos obtidos com 1, 2, 4 e 8 *threads* em uso pelo iPerf. Por seu turno, a coluna Média corresponde à média aritmética das colunas Não e Sim. Os valores a vermelho e a azul representam os melhores para a *firewall* nativa e virtual, respetivamente; os valores sublinhados representam os melhores em termos globais.

MTU	1500				9000			
Hyperthreading	Não	Sim		Média	Não	Sim		Média
Taxa de Transferência	5,386	5,321	-1%	5,354	9,900	9,900	0%	9,900
Carga CPU Firewall	0,664	0,743	+12%	0,704	0,410	0,374	-9%	0,392
Carga CPU i7Client	0,229$^{\bullet}$	0,148$^{\bullet}$	-35%	0,189‡	0,241*	0,327*	+36%	0,284†
Carga CPU i7Server	0,448$^{\bullet}$	0,363$^{\bullet}$	-19%	0,406‡	0,365*	0,348*	-5%	0,357†

Tabela 3.1: Taxa de Transferência (Gbps) e Cargas de CPU com pfSense nativo.

Da análise destas tabelas podem-se tirar as seguintes ilações em termos gerais:

- taxa de transferência através da *firewall*:
  - com MTU 9000, as taxas de transferência são sempre as máximas possíveis (9,9 Gbps) ou muito próximas (pacotes de maior dimensão aproveitam melhor

MTU	1500				9000			
Hyperthreading	Não	Sim		Média	Não	Sim		Média
Taxa de Transferência	3,229	3,354	+4%	3,292	9,870	9,885	+0,2%	9,878
Carga CPU Firewall	0,897	0,854	-5%	0,876	0,497	0,309	-38%	0,403
Carga CPU i7Client	0,042$^{\bullet}$	0,143$^{\bullet}$	+245%	0,093‡	0,254*	0,308*	+21%	0,281†
Carga CPU i7Server	0,102$^{\bullet}$	0,412$^{\bullet}$	+306%	0,257‡	0,285*	0,327*	+15%	0,306†

Tabela 3.2: Taxa de Transferência (Gbps) e Cargas de CPU com pfSense virtual.

a largura de banda disponível); isto acontece independentemente da *firewall* ser nativa ou virtualizada (neste último caso há um decréscimo, mas insignificante);

- com MTU 1500, as taxas de transferência descem apreciavelmente: em média, para 54% (5,354 / 9,900) e 33% (3,292 / 9,878) dos valores com MTU 9000, consoante a *firewall* é nativa ou virtualizada, respetivamente;
- o factor *hyperthreading* tem um impacto residual na taxa de transferência: na *firewall* nativa, há um decréscimo insignificante (-1%) com MTU 1500 e nenhuma alteração com MTU 9000; na *firewall* virtual, há um benefício marginal (+4%) com MTU 1500 e um benefício irrisório (+0,2%) com MTU 9000;
- **em suma**, para maximizar a taxa de transferência relativa aos fluxos de tráfego que envolvem NAT via pfSense, esses fluxos devem ocorrer com MTU 9000; se isso não for possível (MTU 1500), a *firewall* deve pelo menos ser nativa.

- carga da CPU da *firewall*:
  - o uso de MTU 9000 determina invariavelmente uma carga menor da CPU da *firewall* (com pacotes de maior dimensão, há menos pacotes a processar); na instância nativa, a carga média com MTU 9000 é de ≈ 56% (0,392 / 0,704) da obtida com MTU 1500, e na instância virtualizada é de ≈ 46% (0,403 / 0,876); ou seja, usar MTU 9000 faz cair a carga de CPU da *firewall* para metade;
  - o uso de *hyperthreading* tende a aliviar a carga da *firewall*, com impacto significativo (-38%) na *firewall* virtual com MTU 9000; a única exceção, mas com

pouca relevância (+12%), ocorre na instância nativa com MTU 1500;

- a carga da *firewall* nativa é sempre inferior à observada na versão virtualizada; em média, a carga da instância nativa representa 0,704 / 0,876 ≈ 80% (MTU 1500) ou 0,392 / 0,403 ≈ 97% (MTU 9000), da carga da instância virtualizada;
- a carga foi sempre inferior a 1.0, sinal de que a CPU da firewall nunca esteve sobrecarregada; em particular, a *firewall* demonstrou ser capaz de sustentar taxas máximas de transferência com cargas de CPU entre 0.3 e 0.5;
- **em suma**, para minimizar a carga da CPU da *firewall*, o cenário mais favorável é usar a instância nativa com MTU 9000 e *hypertheading* ativo, seguido da opção pela instância virtual nas mesmas condições de MTU e *hypertheading*.

- carga da CPU nos sistemas finais (cliente e servidor iPerf):
  - com MTU 9000, as cargas médias (†) de CPU nos mesmos sistemas finais são semelhantes, independentemente da *firewall* ser nativa ou virtual;
  - já com MTU 1500, estas cargas (‡) divergem claramente, *grosso modo* duplicando com *firewall* nativa (nesta situação, a *firewall* é mais eficiente no processo NAT e, com menos constrangimentos nesse processo, os sistemas finais podem gerar e receber tráfego a um ritmo maior, aumentando a taxa de transferência);
  - a influência do *hyperthreading* é condicionada pelo valor do MTU; com MTU 1500 ($^{\bullet}$), o *hyperthreading* diminui a carga dos sistemas finais quando a *firewall* é nativa, ao passo que aumenta a carga desses sistemas quando a *firewall* é virtual (curiosamente, o valor da carga com *hyperthreading* acaba por ser similar); com MTU 9000 (*), o uso de *hyperthreading* também aumenta sempre a carga no cliente e no servidor, excepto no servidor quando a *firewall* é nativa (-5%);
  - **em suma**, as cargas observadas são relativamente baixas (<0.5); apesar disso, com MTU 9000 já se atinge o máximo teórica da taxa de transferência, sinal de que não é a CPU dos sistemas finais que limita a exploração da taxa de transferência disponível.

3.6.2 Taxa de Transferência em Função do Paralelismo

Os valores das colunas Não e Sim das tabelas 3.1 e 3.2 representam médias dos valores individuais obtidos com diferente número de *threads* usados pelo sistemas finais durante a execução do iPerf. Nesta secção analisam-se os valores individuais das taxas de transferência, a fim de aferir o impacto do uso de um número diferente de *threads*.

Esta análise é significativa porque fornece pistas sobre o que acontece em cenário real, no *cluster*, onde haverá potencialmente múltiplas máquinas virtuais envolvidas em trocas de tráfego com sistemas externos. No estudo presente apenas se usou um sistema interno (cliente iPerf) e um externo (servidor iPerf), mas utilizando-se um número diferente de *threads* já se torna possível perceber como é que a *firewall* reage ao aumento progressivo do número de fluxos NAT, independentemente da sua origem ser numa só máquina ou em máquinas diferentes (na realidade, em cenário de produção, o número deste fluxos ativos a cada instante é dinâmico, podendo ora aumentar ou diminuir, de forma relativamente imprevisível, consoante a atividade específica de cada máquina virtual do *cluster*).

MTU	1500				9000			
Hyperthreading	Não		Sim		Não		Sim	
Nativa/Virtual	Nat.	Virt.	Nat.	Virt.	Nat.	Virt.	Nat.	Virt.
1 Thread	5,334	3,184	5,282	3,248	9,900	9,870	9,900	9,890
2 Threads	5,368	3,182	5,298	3,322	9,900	9,848	9,900	9,894
4 Threads	5,414	3,264	5,338	3,420	9,900	9,872	9,900	9,892
8 Threads	5,426	3,284	5,366	3,426	9,900	9,890	9,900	9,864
Média	5,386	3,229	5,321	3,354	9,900	9,870	9,900	9,885

Tabela 3.3: Taxa de Transferência (Gbps) com pfSense nativo e virtualizado

Da observação da tabela 3.3 , relativa à taxa de transferência observada nos diferentes cenários, pode-se retirar as seguintes conclusões:

- com MTU 9000, a taxa de transferência tende a manter-se máxima (*firewall* nativa) ou quase máxima (*firewall* virtual), independentemente do número de *threads* do iPerf e do uso ou não de *hyperthreading*; portanto, em cenário de produção, com um

número variável de fluxos NAT, esta capacidade máxima deverá ser mutiplexada proporcionalmente pelos clientes da *firewall* (as máquinas virtuais do *cluster*);

- com MTU 1500, a taxa de transferência aumenta ligeiramente em ambas as *firewalls*, com o aumento do número de *threads* do iPerf (fluxos NAT); embora não se tenham feito testes com um número de *threads* superior a 8, é expectável que a taxa de transferência tenha margem de manobra para acompanhar o aumento do número de fluxos NAT, até se atingir um ponto de saturação que, previsivelmente, deverá ficar aquém do valor máximo obtido com MTU 9000; em qualquer caso, a capacidade NAT disponível da *firewall* será dividida pelos clientes NAT ativos;

Em suma, num cenário de produção, com múltiplos fluxos NAT simultaneos, a capacidade máxima da *firewall* para cada tipo de MTU deverá ser devidamente multiplexada. No fundo, a escalabilidade da conexão *edge* do *cluster* ao exterior estará sempre dependente da escalabilidade da implementação NAT oferecida pela *firewall* e, pelo que se pode observar, essa implementação pode ser escalável.

3.6.3 Cargas de CPU em Função do Paralelismo

Nesta secção repete-se o mesmo tipo de análise da secção anterior, mas desta feita com o foco na evolução das cargas de CPU dos sistemas intervenientes, em função do número de *threads* que participam no *benchmark* iPerf. As tabelas 3.6, 3.4 e 3.5 permitem observar essa evolução para cliente iPerf , servidor iPerf e *firewall* respetivamente.

Da análise das tabelas pode-se concluir o seguinte:

- Tendência geral: nos sistemas finais (cliente e servidor), o aumento do número de *threads* envolvidas no *benchmark* implica, em geral, maior carga de CPU, com exceções pontuais (ver *), na *firewall* a tendência geral é a mesma, mas as exceções (ver *) são em maior número; este comportamento mais irregular prende-se com o facto de que na *firewall* o impacto do aumento do número de *threads* envolvidos no iPerf é indireto, dado que esse aumento só ocorre explicitamente nos sistemas finais;

MTU	1500				9000			
Hyperthreading	Não		Sim		Não		Sim	
Nativa/Virtual	Nat.	Virt.	Nat.	Virt.	Nat.	Virt.	Nat.	Virt.
1 Thread	0,182	0,024	0,048	0,122	0,202	0,152	0,210	0,268
2 Threads	0,234	0,036	0,164	0,176	0,208	0,328	0,296	0,230⋆
4 Threads	0,280	0,042	0,240	0,138	0,276	0,276	0,362	0,398
8 Threads	0,218⋆	0,064	0,138⋆	0,136⋆	0,276⋆	0,258⋆	0,438	0,336
Média	0,229	0,042	0,148	0,143	0,241	0,254	0,327	0,308

Tabela 3.4: Cargas de CPU do cliente iPerf - 1,2,4,8 Threads iPerf.

MTU	1500				9000			
Hyperthreading	Não		Sim		Não		Sim	
Nativa/Virtual	Nat.	Virt.	Nat.	Virt.	Nat.	Virt.	Nat.	Virt.
1 Thread	0,418	0,042	0,254	0,220	0,264	0,222	0,236	0,276
2 Threads	0,438	0,080	0,292	0,310	0,332	0,192⋆	0,234⋆	0,252⋆
4 Threads	0,460	0,124	0,296	0,496	0,444	0,300	0,290	0,364
8 Threads	0,476	0,160	0,610	0,622	0,420	0,426	0,630	0,414
Média	0,448	0,102	0,363	0,412	0,365	0,285	0,348	0,327

Tabela 3.5: Cargas de CPU do servidor iPerf - 1,2,4,8 Threads iPerf.

MTU	1500				9000			
Hyperthreading	Não		Sim		Não		Sim	
Nativa/Virtual	Nat.	Virt.	Nat.	Virt.	Nat.	Virt.	Nat.	Virt.
1 Thread	0,580	0,546	0,692	0,704	0,422	0,558	0,352	0,330
2 Threads	0,438⋆	1,020	0,749	0,838	0,438	0,476⋆	0,374	0,312⋆
4 Threads	0,802	1,020	0,708⋆	0,936	0,398⋆	0,360⋆	0,338⋆	0,214⋆
8 Threads	0,838	1,002⋆	0,822	0,936	0,380⋆	0,592	0,430	0,378
Média	0,664	0,897	0,743	0,854	0,410	0,497	0,374	0,309

Tabela 3.6: Cargas de CPU da firewall - 1,2,4,8 Threads iPerf.

- cliente vs servidor: as cargas do cliente iPerf são manifestamente inferiores às do servidor iPerf, com uma exceção (ver adiante); por seu turno, as cargas destes sistemas finais são tipicamente bastante inferiores às exibidas pela *firewall*, novamente com uma exceção (ver a seguir); a exceção em ambos os casos é a mesma, ocorrendo com MTU 9000 e *hyperthreading* ativo, tanto com *firewall* nativa como virtualizada, situação em que as cargas são similares ou da mesma ordem de grandeza, como indicado alias pelos valores médios, que são semelhantes.

- impacto da virtualização: nos sistemas finais, o impacto nas cargas da natureza nativa versus virtual da firewall segue uma tendência dominante, com uma exceção; assim, as cargas acabam por ser equiparáveis (atente-se nos valores médios) para opções iguais de MTU e *hyperthreading*, a exceção ocorre com MTU 1500 e sem *hyperthreading*, situação em que a carga com *firewall* nativa é notoriamente superior à carga com *firewall* virtual; quanto à própria *firewall*, as cargas da versão virtual são superiores às da versão nativa quando o MTU é 1500, mas são equiparáveis (sendo inferiores) quando o MTU é 9000;

- impacto do MTU 1500: as cargas da *firewall* são claramente superiores às dos sistema finais (embora o máximo alcançado na *firewall*, de ≈1.0, seja um valor que se pode considerar modesto); já com MTU 9000, as cargas da *firewall*, sendo menores que com MTU 1500, ainda tendem a ser superiores às daqueles sistemas, embora com menor distanciamento que com MTU 1500.

De notar que algumas destas conclusões também se podem retirar da análise dos valores médios da tabela 3.3.

3.7 Sobrecarga introduzido pelo NAT

Com base nos resultados dos testes realizados, pode-se concluir que quando se utiliza MTU 9000, obtêm-se taxas de transferência estáveis e consistentes, perto do máximo teórico das

interfaces de rede. Desta forma, torna-se mais interessante estudar comportamento dos sistemas com MTU 1500, para perceber as origens e o impacto das limitações observadas.

Em particular, é importante perceber a dimensão da penalização introduzida pelo mecanismo NAT da *firewall*, face à comunicação direta entre os sistemas finais. Nesse sentido, foi executado o iPerf num cenário com ligação direta entre o cliente e o servidor, e com *hyperthreading* ativo (uma vez que já se tinha concluído que, com MTU 1500, favorecia as taxas de transferência). Os resultados deste teste constam da tabela 3.7.

Tabela 3.7: Medição ponto-a-ponto entre i7Server e i7Client

MTU / Hyperthreading	1500 / Sim				
Threads	1	2	4	8	Média
Taxa de Transferência	9,410	9,416	9,420	9,420	9,417
Carga CPU i7Server	0,146	0,154	0,142	0,194	
Carga CPU i7Client	0,248	0,246	0,378	0,348	

Como se pode ver na tabela 3.7, atingem-se taxas de transferência de cerca de 9,4 Gbps de uma forma consistente, independentemente do número de *threads* utilizado. Comparando estes resultados com os da tabela 3.3, pode-se concluir que o mecanismo NAT do pfSense apenas garante 5,386 / 9,417 $\approx$ 57% (*firewall* nativa) ou 3,229 / 9,417 $\approx$ 34% (*firewall* virtual) da taxa de transferência máxima possível entre os sistemas finais.

Uma vez que o uso de MTU 1500 ainda é a opção mais habitual, e a adotada nas redes que servem o *cluster*, a perda de desempenho face à alternativa (menos viável) de MTU 9000 acaba por ser motivo de reflexão e estimulante da busca de alternativas menos penalizadoras. Neste ponto cabe referir que, com o conhecimento entretanto adquirido, a investigação descrita neste capítulo poderia ter sido orientada no sentido de repetir a avaliação realizada com *firewalls* alternativas ao pfSense, como VyOS ou outras baseadas em Linux. No entanto, no âmbito da exploração do *cluster*, a *firewall edge* não desempenha apenas funções NAT. O conceito de mini-clusters, em torno do qual se organizou a exploração do *cluster*, depende de um conjunto adicional de serviços do pfSense, estreitamente integrados, nomeadamente gestão de utilizadores e certificados, gestão de conexões

VPN, serviço de DHCP, serviço de DNS, etc. A gestão diária desses serviços é também bastante facilitada com base no interface web do pfSense.

3.8 Outros Testes

A assunção do uso do pfSense não obsta, porém, a que se termine este capítulo, apresentando resultados de alguns testes colaterais. Ou seja, nesses testes não se reavaliam outras alternativas no desempenho de funções NAT, mas adquire-se mais algum conhecimento sobre as limitações/potencialidades do pfSense e de outras opções equivalentes, ficando para trabalho futuro a avaliação do desempenho NAT dessas alternativas.

De referir que em todos os testes apresentados nesta secção, a *firewall* interveniente é virtualizada, em linha com o cenário de produção adotado no *cluster*. Adicionalmente, manteve-se o uso de MTU 1500 e o *hypertreading* ativo nos sistemas físicos intervenientes.

Desempenho Ponto-a-Ponto entre pfSense virtual e i7Client nativo No primeiro teste adicional, procurou-se perceber o desempenho potencial da firewall pfSense na troca de pacotes com o cliente, sem execução de NAT. A expectativa era a de que o desempenho dessa troca não deveria ser inferior ao observado nas transações com NAT (rever tabela 3.3). Na realidade, deveria até ser superior, dado não ser executada a funcionalidade NAT. O resultado deste teste pode ser observado na tabela 3.8.

Tabela 3.8: Medição ponto-a-ponto entre pfSense virtual e i7Client nativo

MTU / Hyperthreading	1500 / Sim				
Threads	1	2	4	8	Média
Taxa de Transferência	2,520	2,746	2,476	2,408	2,538
Carga CPU pfSense	1,478	1,768	1,782	1,828	
Carga CPU i7Client	0,1	0,086	0,064	0,14	

Os valores medidos da taxa de transferência contrariam as expectativas, pois acabam por ser inferiores aos da tabela 3.3: em termos médios, representam 2,538 / 3,354 $\approx$ 76%.

Ou seja, ter a *firewall* ainda a realizar NAT em direção a um terceiro interveniente (o sistema i7Server) aumentaria a taxa de transferência medida pelo iPerf em 24%.

Desempenho Ponto-a-Ponto entre VyOS / Ubuntu virtual e i7Client nativo Para tentar despistar a origem do resultado paradoxal obtido no teste ponto-a-ponto entre o pfSense virtual e o cliente nativo, decidiu-se repetir esse teste, mas substituindo o pfSense por outras *firewalls* no papel de servidor iPerf. As tabelas 3.9 e 3.10 mostram os resultados obtidos com VyOS e Ubuntu no desempenho desse papel, respetivamente.

Tabela 3.9: Medição ponto-a-ponto entre VyOS virtual e i7Client nativo

MTU / Hyperthreading	1500 / Sim				
Threads	1	2	4	8	Média
Taxa de Transferência	5,758	9,356	9,332	9,382	9,357
Carga CPU VyOS	0,232	0,334	0,524	0,628	
Carga CPU i7Client	0,150	0,138	0,176	0,190	

Tabela 3.10: Medição ponto-a-ponto entre Ubuntu virtual e i7Client nativo

MTU / Hyperthreading	1500 / Sim				
Threads	1	2	4	8	Média
Taxa de Transferência	9,410	9,416	9,420	9,420	9,417
Carga CPU Ubuntu	0,146	0,154	0,142	0,194	
Carga CPU i7Client	0,248	0,246	0,378	0,348	

Os resultados não deixam margem para dúvidas: as máquinas virtuais VyOS e Ubuntu, executando o mesmo teste no mesmo *hardware* que a máquina virtual pfSense, apresentam taxas de transferência muito superiores, inclusivamente não longe do máximo teórico.

Desta forma, conclui-se que a natureza do sistema operativo no qual assenta a *firewall* pode ter uma influência crucial no seu desempenho (o pfSense assenta em FreeBSD, enquanto que o VyOS assenta em Linux Debian, sendo Ubuntu também uma distribuição Linux). E sendo verdade que nestes testes não se está a realizar NAT, o facto é que estas

diferenças levantam a suspeita, fundamentada, de que também o desempenho com essa funcionalidade poderá ser fortemente influenciado pelo sistema operativo da *firewall*.

3.9 Conclusões

Resumem-se, de seguida, as principais conclusões a retirar dos testes apresentados neste capítulo.

- uso de multithreading
  - os dados recolhidos não apontam para um impacto significativo do uso ou não de multithreading.
  - o uso de multithreading parece apresentar melhores resultados em ambiente virtualizado e a sua desativação produz melhores resultados em ambiente nativo, mas em qualquer dos casos a diferença é marginal.
- grau de paralelismo
  - considerando as taxas de transferência com MTU 9000, os valores obtidos são praticamente idênticos, e correspondem ao máximo nominal, sinal que o pfSense está a partir a carga de forma homogénea; com MTU 1500, a taxa de transferência parte de um patamar inferior e vai subindo com o número de threads.
  - ao contrário do que eventualmente seria de esperar, o aumento do número de threads para execução do teste não resulta num aumento de desempenho significativo.
  - os dados sugerem que o desempenho do CPU não estará a condicionar o desempenho das interfaces de rede.
- implementação nativa ou virtualizada

  - há uma penalização no desempenho (taxa de transferência), quando se passam de uma implementação da firewall em hardware nativo para um ambiente virtualizado e a dimensão dessa penalização depende da MTU.
  - com MTU 9000, essa penalização é insignificante, com MTU 1500, essa penalização é de cerca de 66%.
  - usando MTU 1500, ao passarmos de uma implementação da firewall em hardware nativo, para uma solução virtualizada, temos uma penalização de cerca de 40% na capacidade máxima da firewall.

- sobrecarga introduzida pelo NAT
  - tendo como ponto de partida para a análise os dados constantes da tabela 3.7, podemos concluir que os resultados dos testes realizados com MTU 1500, ficaram bastante aquém do que seria a expectativa inicial.

- Sistema Operativo
  - Se por um lado a instanciação da MV Ubuntu veio provar que o hardware é capaz de atingir o desempenho pretendido, mesmo para MTU 1500, por outro lado não ficou claro o motivo de o pfSense apresentar tão fraco desempenho.

Em suma, mantendo o pfSense como *firewall*, o grande fator diferenciador em termos de maximização do desempenho , é claramente o MTU. Apesar da capacidade dos sistemas, quer em termos de velocidade de relógio (sobretudo os i7Server e i7Client), quer em número de cores (Epyc), parece não ser possível igualar o desempenho para MTU 9000 fazendo a comutação de mais pacotes com MTU 1500.

Uma das questões colocadas inicialmente relativamente à sobrecarga introduzida pelo NAT foi cabalmente respondida, ou seja, muito reduzido ou nenhuma. Mesmo no caso do pfSense, foram obtidos melhores desempenho nos testes em que a firewall fazia apenas o NAT do que nos testes em que fazia o teste da ligação ponto a ponto.

Relativamente à questão das diferença de desempenho entre uma implementação em bare metal da firewall ou uma implementação da mesma virtualizada parece não haver uma diferença significativa.

Dos dados analisados podemos concluir que para um cenário como aquele construído de forma a levar a cabo estes testes de desempenho, ou outros que se assemelhem, a capacidade dos CPUs em causa é mais do que suficiente (e com larga margem) para assegurar que o desempenho global do sistema não fica comprometido pelo desempenho do CPU. Veja-se o exemplo do teste realizado entre a máquina i7 e a máquina virtual ubuntu (Tabela 3.10) em que se consegue atingir débitos a rondar os 9,4 Gbps de forma consistente. A velocidade do CPU deverá no entanto ser a razão, para que em alguns testes, em que apenas se usou uma thread, o desempenho fique um pouco aquém do expectável, com é no caso nos testes com vyOS.

Durante a realização dos testes, foi observado qeu a firewall pfSense apenas usava 2 cores dos 32 disponíveis para fazer NAT mesmo quando se estava a usar 8 threads em simultâneo no cliente e servidor.

Não se tendo medido, porque não seriam medidas fidedignas, foi no entanto observado que quando ao teste referido anteriormente, foi adicionada ao mesmo tempo a maquina virtual com Ubuntu como cliente e o i7Server como servidor, a maquina virtual com o pfSense continuou a usar apenas 2 cores. Nos dois i7, a carga é distribuída pelos 8 *vcores*.

Pelos testes realizados, o uso de uma firewall em ambiente virtualizado, com um hypervisor com menos cores e mais velocidade de relógio e usando MTU 9000 será o caminho mais recomendável.

3.10 Epílogo

O pressuposto com que se abordou a temática desenvolvida ao longo deste capitulo, foi substancialmente revertido, ou seja, a ideia inicial de que seria o hardware usado que condicionaria em grande medida o desempenho global do sistema.

Verificamos assim que o a definição da topologia de rede e o software, nomeadamente o

sistema operativo tem um peso significativo no desempenho global como podemos observar nos diferentes testes.

Todas as maquinas usadas nestes testes, foram instalas usado as definições por omissão fornecidas no momento de instalação, ou seja, não foi feita nenhuma otimização, nenhuma afinação ou alteração desse tipo.

No caso de se querer continuar a usar pfSense, deverá ser feito um estudo mais aprofundado do porquê da discrepância observada e quais as opções para otimização deste sistema operativo. O uso de interfaces de rede 10G expôs estas diferenças apenas no uso de MTU 1500 o mesmo não acontecendo no MTU 9000. Uma vez que o uso de MTU 9000 reduz, grosso modo, o número de pacotes a ser processados em 6 vezes, possivelmente apenas quando se passar para o uso de interfaces de rede de 40G ou superiores começaremos a ver diferenças de capacidade de processamento pelos diferentes sistemas.

Capítulo 4

Otimização do Armazenamento

4.1 Preâmbulo

À imagem do capítulo anterior, no estudo das soluções de armazenamento de suporte ao *cluster*, há também diversas variáveis a considerar. Assim, serão analisadas as correlações entre diferentes variáveis de forma a tentar estabelecer cenários óptimos e relações causa e efeito entre alterações de comportamento/desempenho do sistema como um todo.

O uso de armazenamento partilhado é de certa forma imposto pela plataforma atual (oVirt). Esta plataforma apresenta alguns condicionalismos, que devem ser estudados e avaliados e eventualmente equacionadas alternativas. O uso de armazenamento partilhado implica a dependência de protocolos de acesso a armazenamento remoto: o ponto de partida é o uso atual de NFS, mas a opção por iSCSI poderá também ser equacionada.

A escolha do sistema de ficheiros também poderá condicionar o desempenho do sistema de armazenamento, justificando-se a análise do impacto de diferentes sistemas de ficheiros.

Considerando o fator desempenho como o único a ter em conta, sabe-se que, à partida, os SSDs SATA ou NVMe apresentarão sempre melhor desempenho que os discos mecânicos usados atualmente (rever secção 2.6.1). Todavia, é uma incógnita qual o *speedup* efetivo que de facto se poderá obter com a migração de discos mecânicos para SSDs.

4.2 Introdução

O conhecimento já adquirido com experiências passadas, consulta de literatura e relatos da indústria, faz pensar que uma correta definição da arquitetura de armazenamento será absolutamente crítica para um bom desempenho de um *cluster* de virtualização, sobretudo se este recorrer ao armazenamento partilhado de forma significativa ou até única.

Empiricamente, sabe-se que um *array* de SSDs terá um desempenho superior a um *array* equivalente de discos mecânicos. No entanto, torna-se necessário quantificar de forma sistemática esse ganho e corroborar a assunção inicial.

Em termos de infraestrutura de rede sabe-se também, *a priori*, que uma rede de 100Gbps oferecerá sempre melhor desempenho que uma rede de 10Gbps, mas até que ponto o aumento real no desempenho se aproxima do aumento nominal/esperado ?

Usando armazenamento partilhado, certamente a rede desempenhará um papel fundamental, mas qual será o fator crítico: a largura de banda ou a latência ? Tratando-se de armazenamento partilhado, foram efetuadas medições ao desempenho da rede de forma a verificar que o comportamento da respetiva infraestrutura corresponderia ao expectável.

Dada a miríade de variáveis passíveis de serem avaliadas num teste deste tipo, é pouco prático e pouco produtivo considerá-las a todas. Assim, foi feito um estudo prévio para tentar determinar quais os parâmetros mais relevantes e definir uma metodologia de teste. Para cada parâmetro foram selecionadas as variantes mais representativas das cargas típicas do *cluster* e que poderão apresentar uma correlação mais fidedigna com o desempenho global.

4.3 Bancada de Teste

Como já referido, até agora tem sido utilizado um sistema de armazenamento centralizado baseado em TrueNAS, com discos mecânicos organizados em raid-z2 e com um sistema de ficheiros ZFS. De forma a generalizar os testes e suprimir a limitação do TrueNAS em usar apenas ZFS como sistema de ficheiros, optou-se por usar o Ubuntu Server no

servidor de armazenamento usado nos testes, dado que suporta não só ZFS como outros sistemas de ficheiros relevantes para o estudo, nomeadamente EXT4. A máquina usada como cliente nos vários testes operou também sob a mesma versão do Ubuntu.

Em termos de hardware, as duas máquinas usadas possuem as seguintes características:

- Máquina Servidora (FlashArray)
  - Motherboard Supermicro H12DSU-iN
  - CPU 2 x AMD EPYC Zen 2 7302 de 16 Cores
  - 256 GB RAM ECC de 3200 MT/s
  - 12 x SSD Samsung 1743 de 3.48TB PCIe 4
  - NIC 10G Intel X550-AT2 + NIC 100G Mellanox ConnectX-5
- Máquina Cliente (nó de virtualização)
  - Motherboard Supermicro H11DSU-iN
  - CPU 2 x AMD EPYC Zen 7351 de 16 Cores
  - 256 GB RAM ECC de 2666 MT/s
  - 1 x Intel SSD DC P4600 de 2TB PCIe 3
  - NIC 10G Intel X550-AT2 + NIC 100G Mellanox ConnectX-5

De referir que o *cluster* possui vários tipos de nós, de diferentes gerações, e que a máquina escolhida para cliente pertence ao grupo mais numeroso (logo, mais representativo).

4.4 Armazenamento Local RAW

De forma a se poder analisar onde potencialmente poderão estar as perdas de desempenho introduzidas por diferentes elementos da stack que integram o sistema de armazenamento, numa primeira fase foi feito um estudo do FlashArray da forma mais simples possível,

ou seja, medindo localmente o desempenho dos SSDs em modo RAW (não formatado), permitindo assim observar as suas métricas máximas de desempenho.

O número de SSDs usados em simultâneo (parâmetro *numdisks*) variou, entre 1, 2, 4 e 12 (parâmetro *numdisks*). Para as configurações com 2, 4 e 12 SSDs foram criados *arrays* RAID 0, com esse número de SSDs, e escolhido o tamanho de bloco de 4k como o mais representativo e mais genérico e que abarca de forma mais satisfatória um diferente número de cenários. Os arrays RAID foram criados recorrendo à ferramenta *mdadm* oferecida pelo Linux para gerir arrays RAID baseados em software.

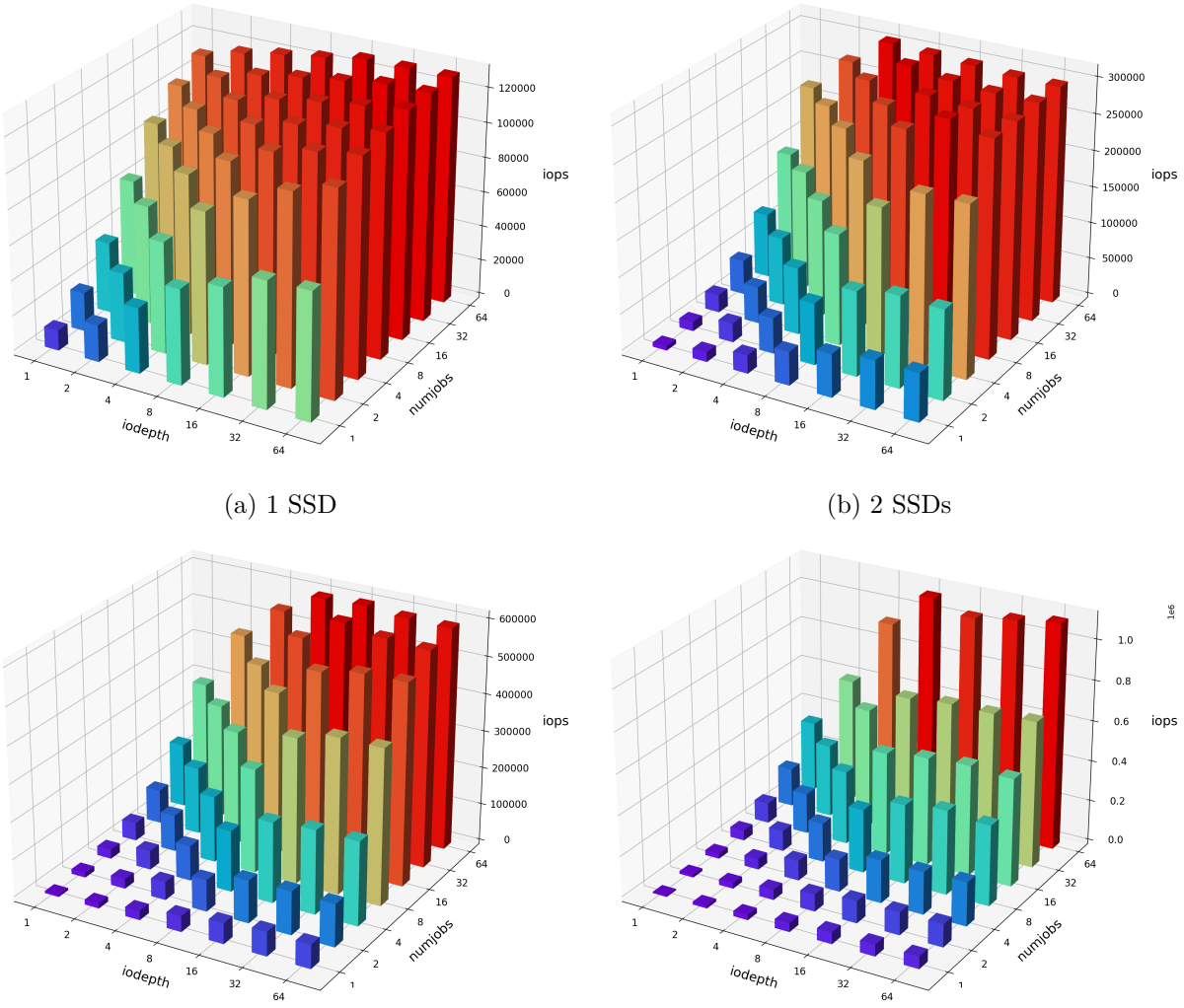
Em termos de variação da carga de trabalho, os parâmetros *numjobs* e *iodepth* permitem simular diferentes cenários de carga. O primeiro corresponde ao número de trabalhos (operações de leitura/escrita) que podem ser executados simultaneamente em determinado dispositivo de armazenamento. O segundo é número de operações de leitura/escrita já solicitadas mas ainda não executadas. A conjugação da análise destes dois parâmetros fornece uma ideia concreta dos pontos críticos dos sistemas de armazenamento assim como da linearidade da capacidade de acomodação da carga de trabalho solicitada.

De forma a realizar as referidas medições, foi usado a ferramenta `fio`[1] e, numa primeira fase, foi feita a medição para diferentes combinações de *numjobs* (1, 2, 4, 8, 16, 32, 64) e *iodepth* (1, 2, 4, 8, 16, 32, 64). Para cada combinação foi feita a análise de desempenho em termos de IOPS[2] de escrita, IOPS de leitura, latência de escrita, e latência de leitura. Quanto ao tipo de teste, foi realizado um teste de leitura e escrita aleatórias com uma distribuição de carga de 75% para leituras e 25% para escritas. Nas secções seguintes são apresentados e discutidos os resultados destes testes preliminares.

4.4.1 IOPS de Escrita e Leitura

Começa-se por analisar o comportamento do sistema servidor em IOPS de escrita (Figura 4.1). Neste vetor, está em linha com o expectável: com mais SSDs, o desempenho cresce de forma proporcional ao número de SSDs no *array*, como podemos verificar através da

[1]Flexible I/O tester - ver `https://fio.readthedocs.io/en/latest/fio_doc.html`.
[2]Input/Output operations per second - ver `https://en.wikipedia.org/wiki/IOPS`.



(a) 1 SSD

(b) 2 SSDs

(c) 4 SSDs

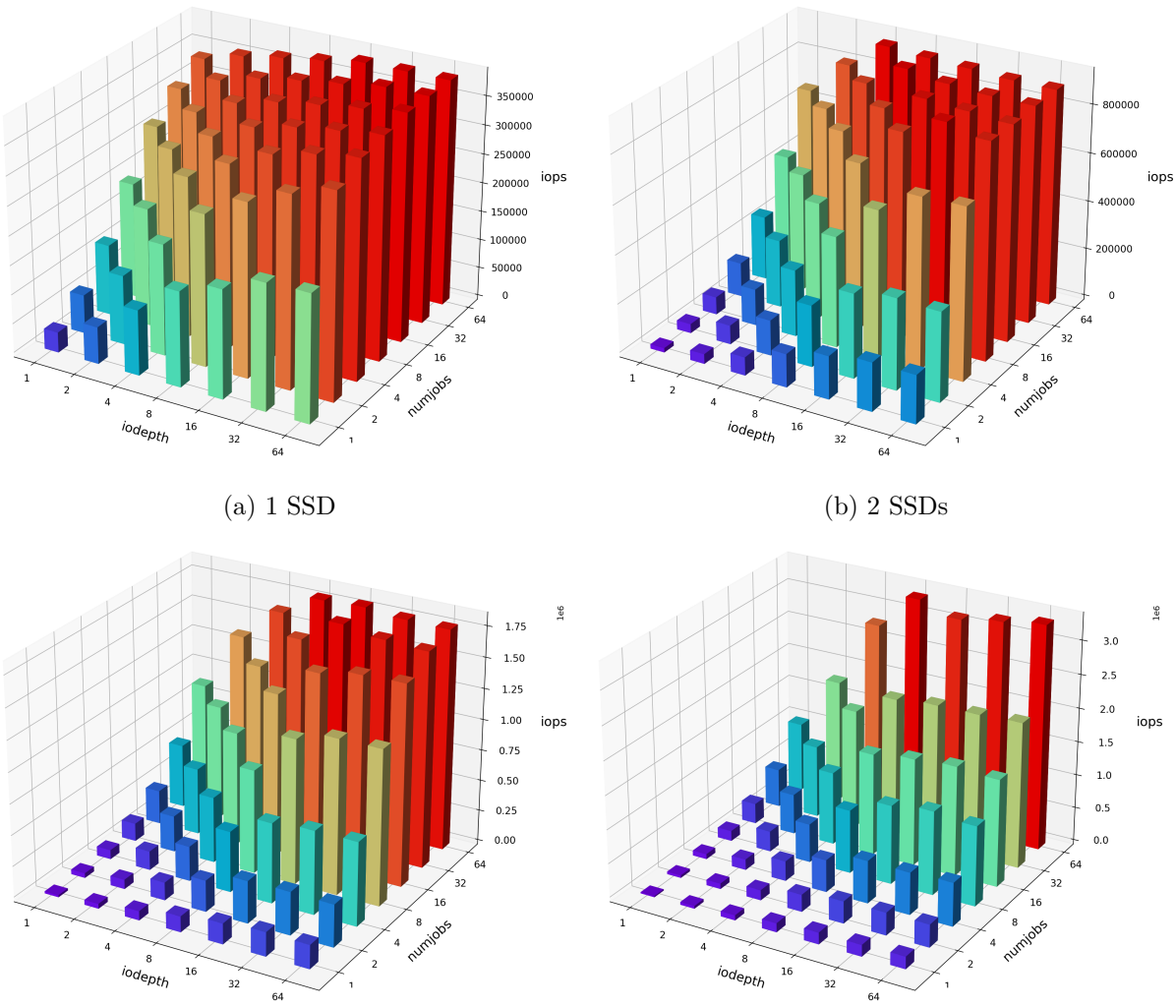
(d) 12 SSDs

Figura 4.1: IOPS de escrita em armazenamento local RAW com RAID 0

análise das figuras 4.1a a 4.1d (nesta última, os IOPS são expressos em milhões).

Embora haja um comportamento consistente nos vários cenários, há *nuances* a relevar: com menos SSDs, os IOPS máximos atingem-se mais rapidamente, com o aumento de *iodepth* ou de *numjobs*; já com mais SSDs, o aumento de *numjobs* torna-se cada vez mais determinante no aumento dos IOPS (em detrimento do aumento de *iodepth*).

Os gráficos das medições para IOPS de leitura constam das figuras 4.2a a 4.2d. Assim como na escrita, a progressão dos resultados obtidos é consistente entre os vários cenários e apresenta correlações semelhantes com o número de dispositivos que constitui o *array*:



(a) 1 SSD

(b) 2 SSDs

(c) 4 SSDs

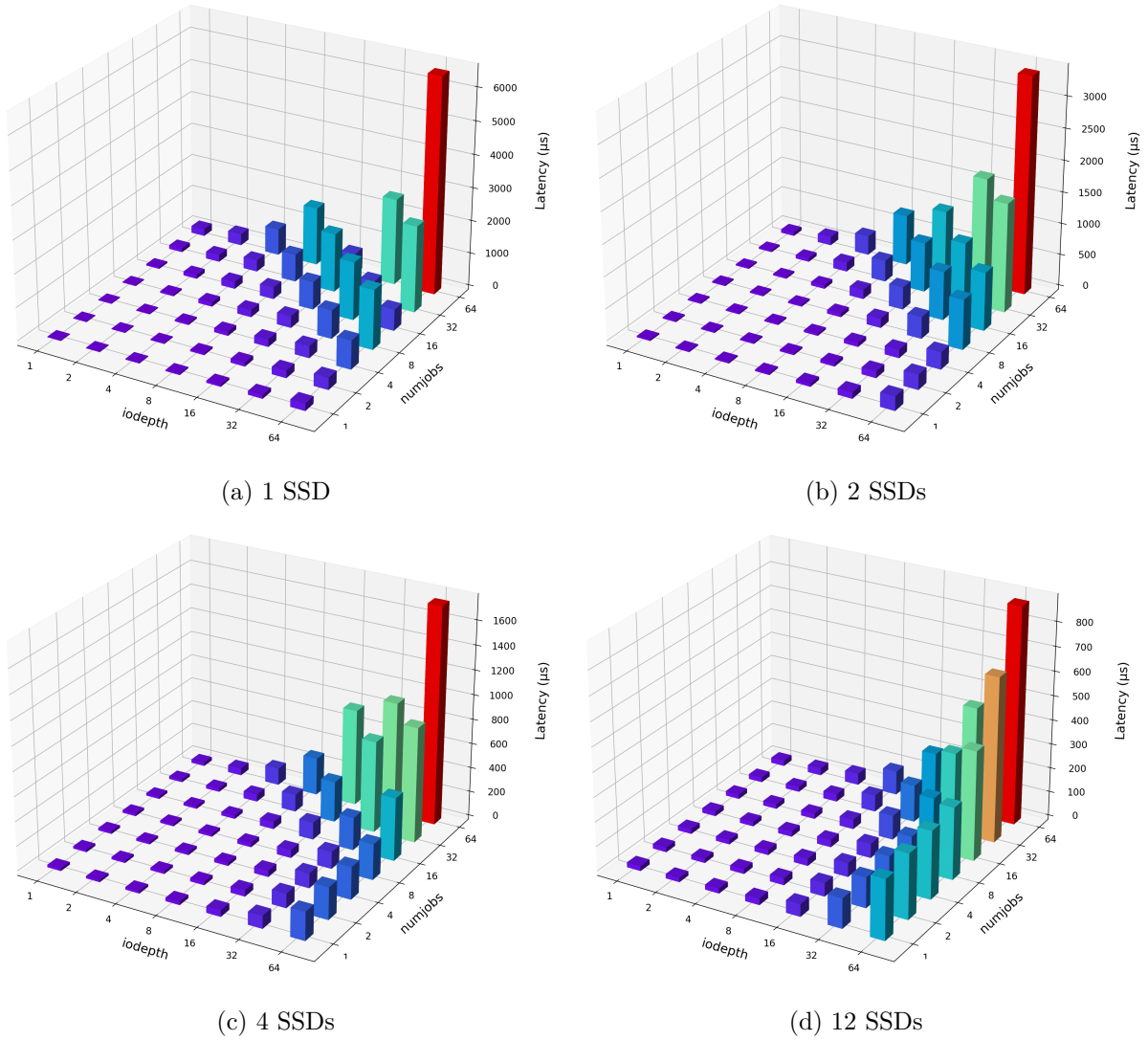
(d) 12 SSDs

Figura 4.2: IOPS de leitura em armazenamento local RAW com RAID 0

quantos mais SSDs, mais necessário se torna aumentar *numjobs* para aumentar os IOPS (e menos influência tem o aumento de *iodepth*), e vice-versa; e também como na escrita, tanto o aumento de *numjobs* como *iodepth* provocam o aumento dos IOPS, mas esse efeito é claramente mais pronunciado quando aumenta o *numjobs*.

De notar que, em termos absolutos, a capacidade de leitura para as mesmas combinações de *iodepth* e *numjobs* é cerca de 3 vezes superior à capacidade de escrita.

4.4.2 Latência de Escrita e Leitura



(a) 1 SSD

(b) 2 SSDs

(c) 4 SSDs

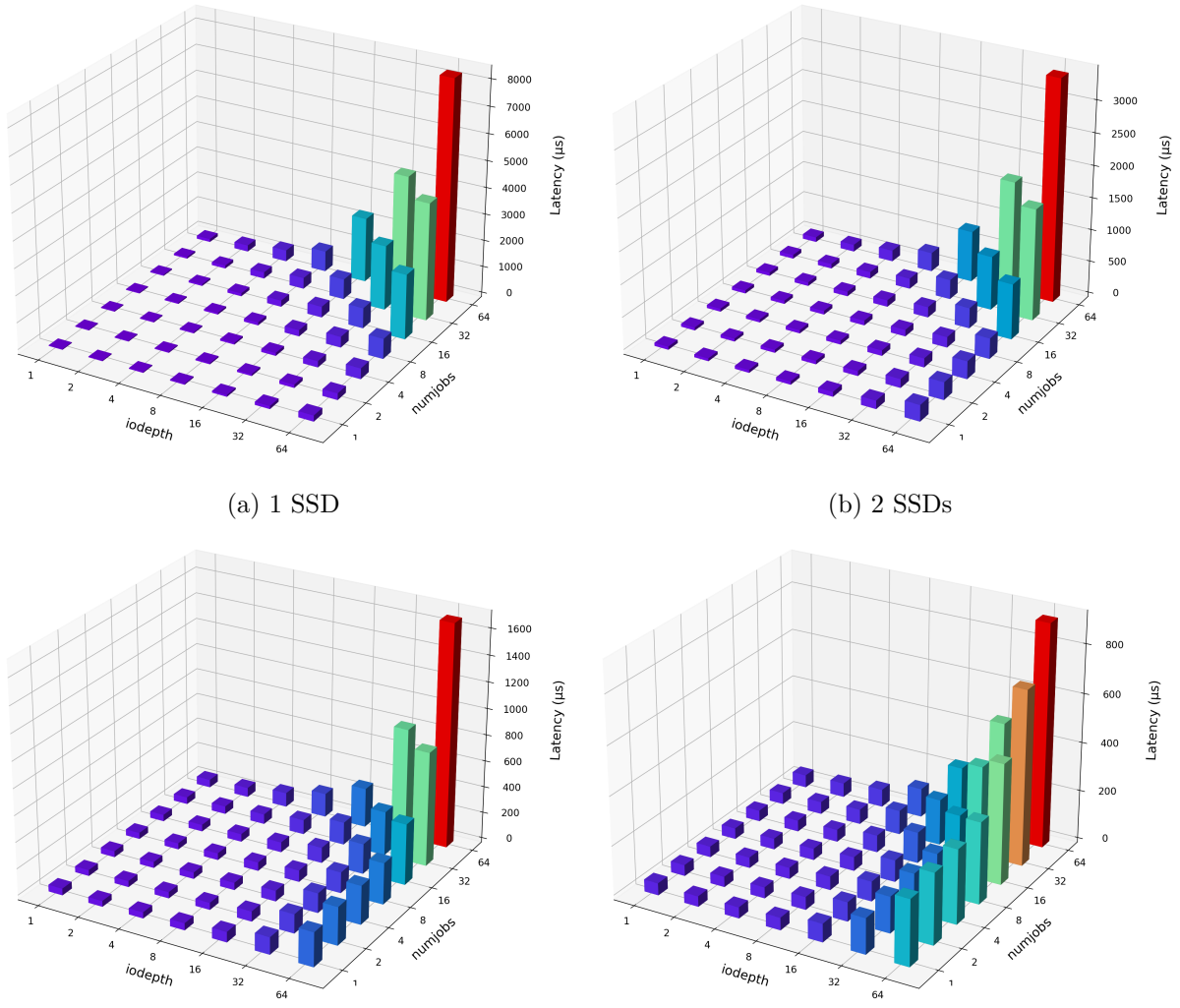
(d) 12 SSDs

Figura 4.3: Latência de escrita em armazenamento local RAW com RAID 0

Observando as figuras 4.3a a 4.3d, relativas à latência na escrita, verifica-se que há uma correlação entre o aumento do número de discos do *array* e a diminuição do tempo de escrita, como seria expectável (dado que há mais discos e trabalhar em paralelo, e cada disco escreve uma quantidade menor de dados). Essa diminuição é de cerca de 50% a cada progressão dos valores analisados.

O padrão das latências representadas nas figuras 4.3c e 4.3d indica que há uma ligeira

alteração do equilíbrio entre a profundidade de fila e o número de trabalhos: para valores de profundidade de fila de 32 e 64, as latências parecem crescer mais acentuadamente, mesmo quando se usa apenas 1 ou 2 trabalhos em simultâneo.



(a) 1 SSD

(b) 2 SSDs

(c) 4 SSDs

(d) 12 SSDs

Figura 4.4: Latência de leitura em armazenamento local RAW com RAID 0

Nas figuras 4.4a a 4.4d, representativas da latência de leitura, verifica-se que o padrão, para as diferentes combinações de número de trabalhos e profundidades de fila se mantém consistente variando o número de SSDs. Esta consistência mantêm-se também no que toca aos valores em si das latências, que caem para metade dobrando o número de SSDs.

As combinações com maior número de trabalhos em simultâneo com filas de profundidade 32 ou 64 resultam no maior acréscimo de latência com qualquer número de SSDs.

Quanto ao padrão observado nos gráficos da latência de escrita, o mesmo não é tão consistente como nos gráficos da latência de leitura, nomeadamente nas dimensões das filas e número de trabalhos mais elevados. Assim, pode-se inferir, a partir desta observação, que as latências de escrita são mais sensíveis a situações limite ou de algum congestionamento.

4.4.3 Definição do Teste Padrão

Dado o número elevado de combinações *numdisks* × *numjobs* × *iodepth* possíveis, torna-se necessário escolher os valores mais relevantes desses parâmetros de forma a viabilizar a análise dos efeitos que outras variáveis, como por exemplo a latência de rede, o sistema de ficheiros ou mesmo outro tipo de configuração do *array* de discos, possam ter no *cluster*.

Ora, com base na observação e comparação entre os diferentes gráficos, pode-se concluir que se por um lado parece haver uma certa linearidade de IOPS em função do número de trabalhos, por outro, em termos de latência das operações, parece ser a partir dos 16 trabalhos simultâneos que se começa a ver a latência crescer mais significativamente.

Outro fator a ter em consideração é o número de máquinas virtuais por nó de computação, que flutuará entre as 10 e as 20. Neste contexto, ao se tentar gerar uma carga representativa, a opção por 8 trabalhos em simultâneo parece ser manifestamente insuficiente e a opção por 32 trabalhos não representa uma carga típica.

Assim, pode-se definir um teste padrão com 16 trabalhos em simultâneo (*numjobs*), com variações de profundidade de fila e um tamanho de bloco de 4k, como o mais representativo e mais abrangente em termos de cargas de trabalho típicas. Isso cria condições para fazer uma análise mais detalhada de outras variáveis, tendo apenas como referência os valores para 16 trabalhos em simultâneo.

Nas colunas azuis das figuras 4.5 e 4.6 estão representados os resultados obtidos desta forma, com um array de 12 SSDs NVMe em RAID0 e acesso local, e que serão usados como referência nas próximas comparações e análises.

4.5 Armazenamento Local Formatado

O estudo preliminar anterior foi realizado sobre armazenamento local em modo RAW e terminou com a fixação de *numjobs*=16. Nesta secção, o armazenamento local volta a estar em foco, mas formatado, introduzindo o sistema de ficheiros como nova variável a ter em consideração. Adicionalmente, utiliza-se a totalidade dos SSDs do FlashArray, organizados em RAID0 (ou nível equivalente), por ser o nível RAID de maior desempenho.

Assim, foram repetidos os testes de medição de IOPS e latência, com o array de 12 SSDs em RAID0, e formatado com os seguintes sistemas de ficheiros: *EXT4*, *EXT 4* com algumas otimizações, *ZFS* em modo assíncrono e *ZFS* em modo síncrono.

A escolha do *EXT4* deriva do facto de ser o sistema de ficheiros de referência do Linux. Neste sentido, testou-se também uma variante montada com base no comando:

```
mount -t ext4 -o defaults,noiversion,auto_da_alloc,noatime,\
errors=remount-ro,commit=20,inode_readahead_blks=32,delalloc, \
barrier=0 /dev/md12 /mnt/array
```

que aplica um conjunto de afinações (definidas com base em experiências realizadas e consulta de literatura) com o objetivo de melhorar o desempenho:

- `noiversion` - não incrementa o registo *i_version* dos inodes;
- `noatime` - não atualiza a data de acesso aos inodes;
- `commit=20` - período (em segundos) de sincronização da cache do sistema de ficheiros em RAM, com o nível de armazenamento secundário;
- `delalloc` - alocação diferida de blocos (agrupa os blocos para escrita adiada).

A escolha de ZFS é óbvia, pois é o sistema de ficheiros usado no servidor de armazenamento do cluster. Neste caso, as variantes síncrona e assíncrona têm o seguinte significado: na variante síncrona, a operação de escrita é só dada como concluída quando os dados são

efetivamente escritos no disco; na variante assíncrona a operação é dada como concluída logo que é registada (com elevada probabilidade de ser concluída posteriormente).

Em relação ao array RAID0, o array formatado com o EXT4 continuou a ser oferecido pelo subsistema *mdadm* do Linux. Para as medições com o sistema de ficheiros ZFS, foi usada uma pool ZFS do tipo *striped*, funcionalmente equivalente a nível RAID0.

Os resultados dos testes com o array formatado são apresentados nas figuras 4.5 e 4.6, lado a lado com a base da comparação fornecida pelo mesmo array em modo RAW.

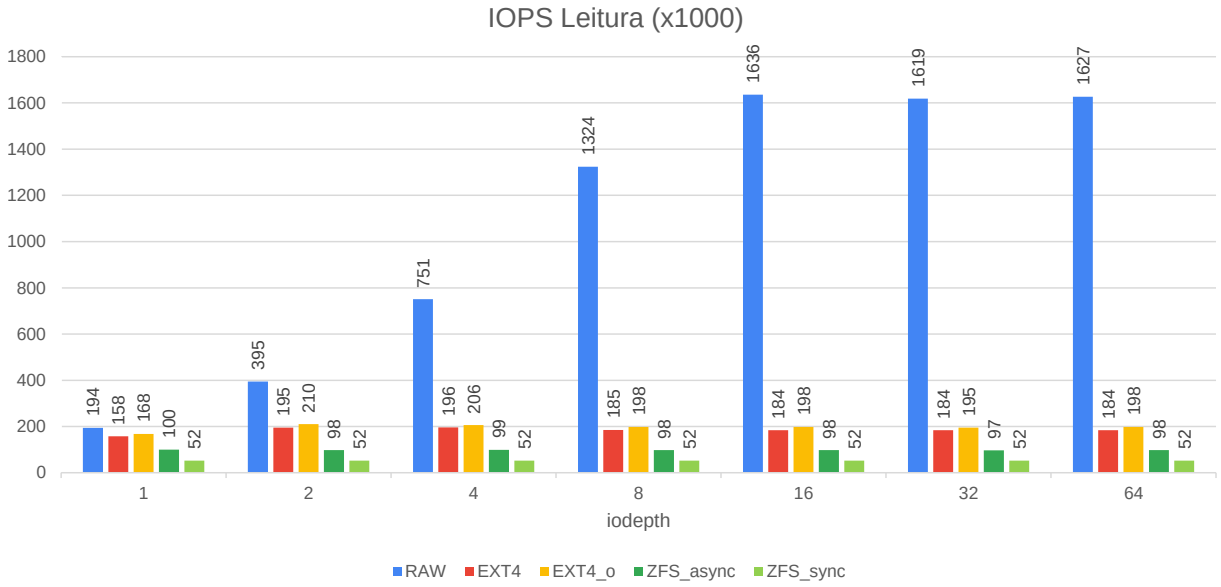
Os resultados vêm confirmar a expectativa inicial de que o ZFS é de facto um sistema de ficheiros mais lento e não vocacionado para funcionar sobre dispositivos NVMe. Os resultados em termos de IOPS, tanto para escrita como leitura, são cerca de 10% do máximo obtido ao se utilizar o mesmo *array* em modo RAW, e são metade ou um quarto dos obtidos com EXT4, c.f. se use ZFS em modo assíncrono ou síncrono.

Em termos de latência, verifica-se um aumento das latências nas diferentes profundidades de fila, mas sobretudo para as escritas. Importa, no entanto, fazer uma ressalva: o uso de mecanismos de cache poderá melhorar drasticamente o desempenho (se bem que a implementação deste tipo de mecanismo no âmbito deste trabalho iria de alguma forma distorcer os resultados e a equitatividade entre comparações realizadas).

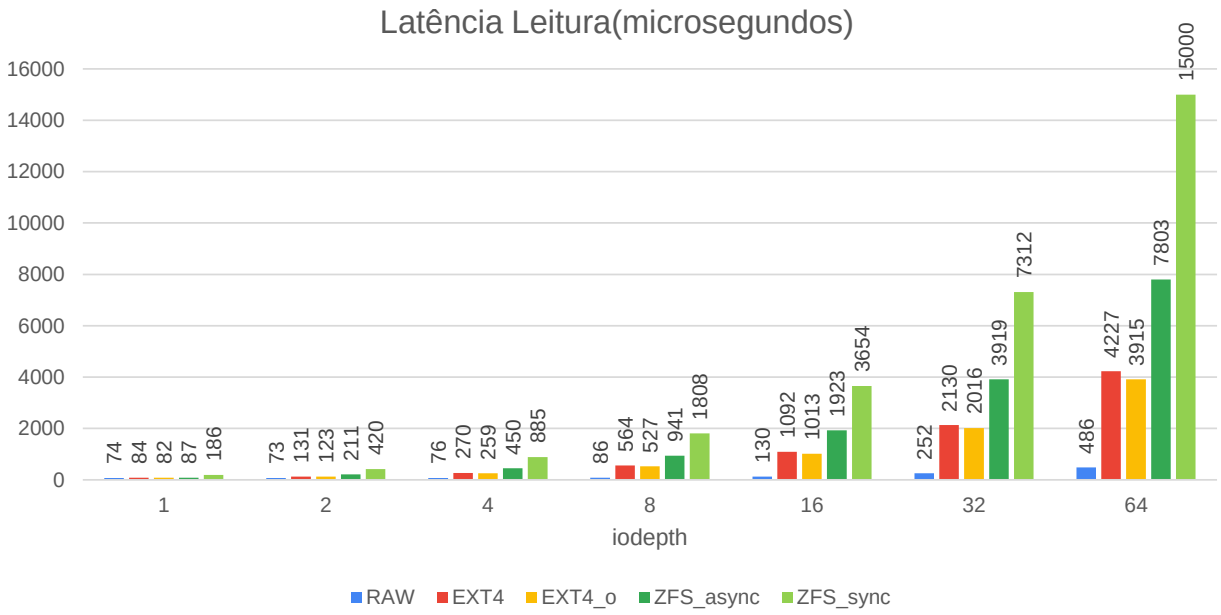
4.6 Armazenamento Remoto via NFS

Tendo uma perceção fundamentada do desempenho e dos valores passíveis de ser obtidos localmente, avança-se para a análise do desempenho do mesmo array RAID0 mas introduzindo o acesso via rede através de NFS,e limitando o estudo apenas aos cenários com o melhor desempenho (array formatado com EXT4 otimizado ou com ZFS assíncrono).

Para as medições no acesso remoto foram utilizadas duas ligações de rede diferentes (10Gbps e 100Gbps) assim como MTUs diferentes (1500 e 9000), para aferir da importância que estes fatores poderiam representar em termos de desempenho. A ligação da rede, quer 10Gbps quer 100Gbps, foi feita ponto a ponto, com um cabo direto entre as duas máquinas (cliente e servidora).

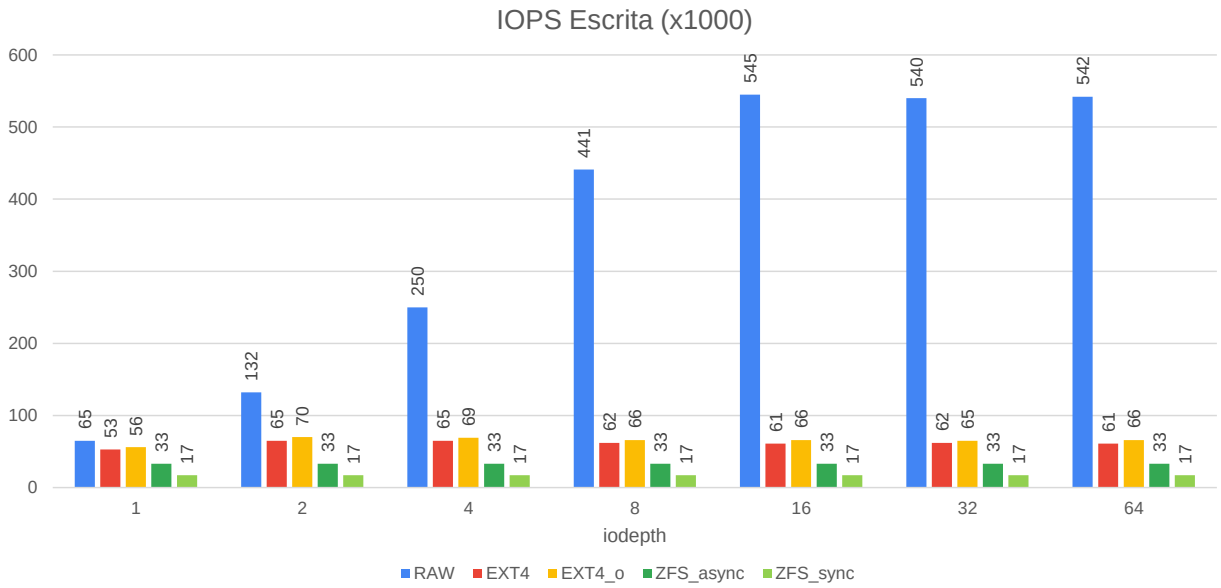


(a) IOPS em leitura

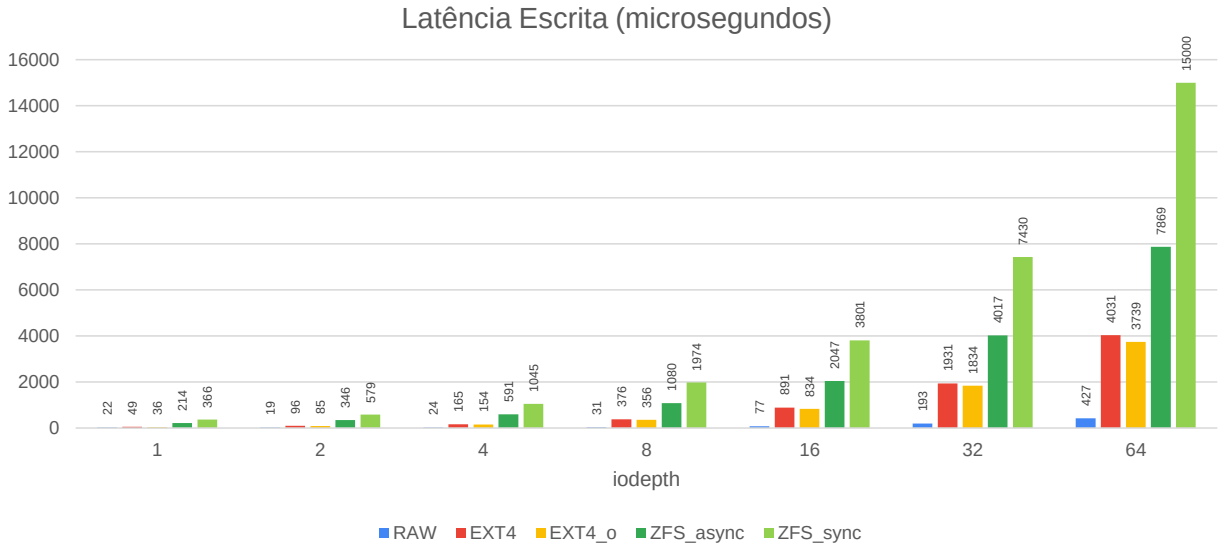


(b) Latência em leitura

Figura 4.5: Leitura: IOPS e Latência em array de 12 Discos



(a) IOPS em escrita



(b) Latência em escrita

Figura 4.6: Escrita: IOPS e Latência em array de 12 Discos

4.6.1 Análise Preliminar da Rede

Antes da realização dos testes com NFS (e outras opções dependentes de rede, como NVMe-oF), foram feitas medições, com o iPerf, entre as máquinas cliente e servidora, para garantir que as ligações de rede 10Gbps e 100Gbps usadas, estariam a funcionar como expectável em termos de desempenho e escalabilidade. Para a realização destas medições adotou-se a metodologia usada no capítulo 3. A tabela 4.1 apresenta os resultados.

	MTU 1500						MTU 9000					
Threads	1	2	4	8	16	32	1	2	4	8	16	32
10G Rate	9.41	9.40	9.42	9.42	9.42	9.42	9.9	9.9	9.9	9.9	9.9	9.9
100G Rate	21.1	40.7	72.8	86.8	86.2	82.4	34	63.3	97.9	98.5	98	98.3

Tabela 4.1: Desempenho da ligação entre cliente e servidor NFS (Gbps)

Pela observação da tabela 4.1 pode-se concluir que na ligação 10Gbps os valores observados são extremamente consistentes, quer com MTU 1500 quer com MTU 9000, para qualquer número de *threads*, com ligeira vantagem para MTU 9000.

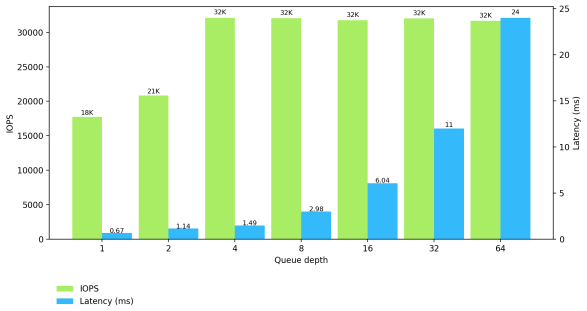
Relativamente à ligação 100Gbps, com MTU 1500 fica longe do máximo nominal, com qualquer número de *threads*, embora com 8 ou mais *threads* se consigam entre 80 e 90Gpbs. Já com MTU 9000, atinge-se um valor muito próximo do máximo com 4 ou mais *threads*.

Tendo como pano de fundo estes valores, apresentam-se de seguida os resultados dos testes de acesso via NFS, ao array RAID0 de 12 SSDs do servidor FlashArray.

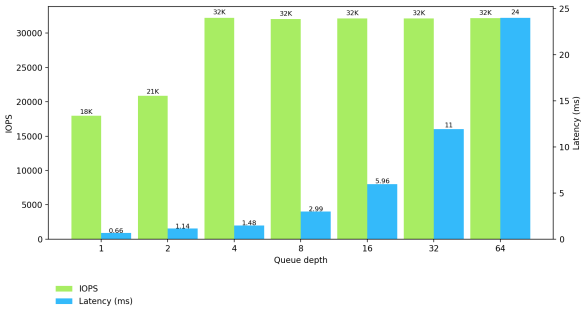
4.6.2 Desempenho da Leitura

As figuras 4.7 e 4.8 apresentam os resultados, em termos de IOPS e latência, da leitura de partilhas NFS formatadas com ZFS e EXT4, respetivamente.

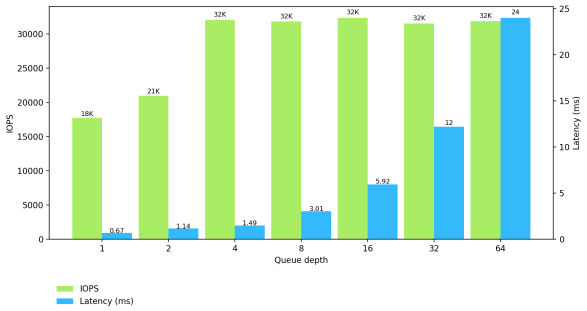
Focando primeiramente a atenção no ZFS, o dado mais relevante resultante da análise das figuras 4.7a e 4.7b, é de que para o uso de MTU 1500 ou MTU 9000, numa ligação 10G, a diferença é praticamente nula, o que aliás é consistente com as medições da tabela 4.1. As figuras 4.7c e 4.7d, correspondentes à situação em que se usa uma ligação 100G,



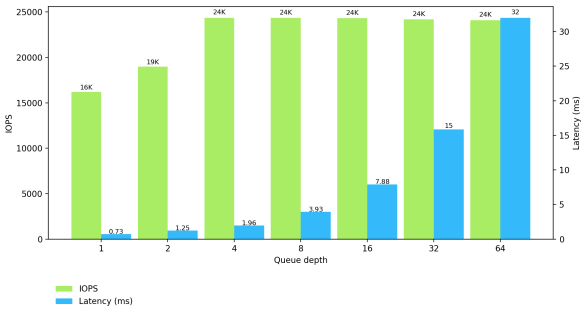
(a) 10G e MTU 1500



(b) 10G e MTU 9000



(c) 100G e MTU 1500



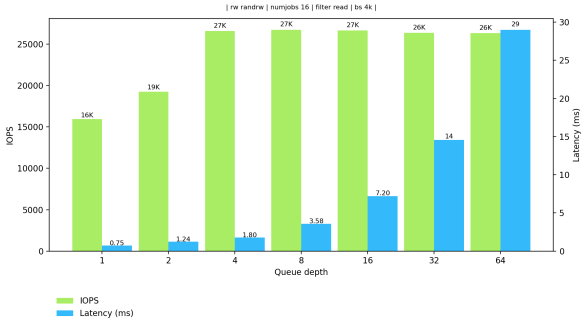
(d) 100G e MTU 9000

Figura 4.7: IOPS e latência em leitura de ZFS em acesso remoto

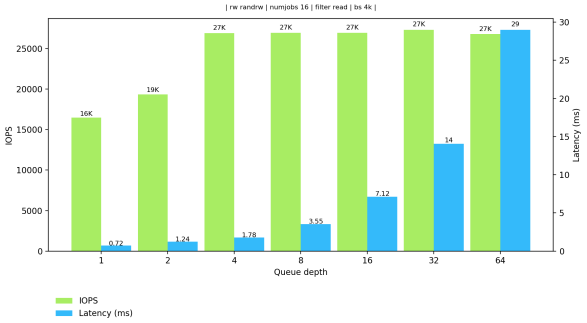
demonstram, algo surpreendentemente, que se obtém melhor desempenho, quer em termos de IOPS quer em termos de latência, com MTU 1500 em vez de 9000.

Para o cenário em que se acede a uma partilha remota formatada com EXT4, as figuras 4.8a e 4.8b, revelam novamente que com uma ligação 10G, não há diferenças significativas, quer em termos de latência quer em termos de IOPS, quando se usam MTUs diferentes. Por outro lado, comparando com as medições análogas feitas para ZFS (figuras 4.7a e 4.7b), verifica-se que a opção por EXT4 acarreta pior número de IOPS e pior latência.

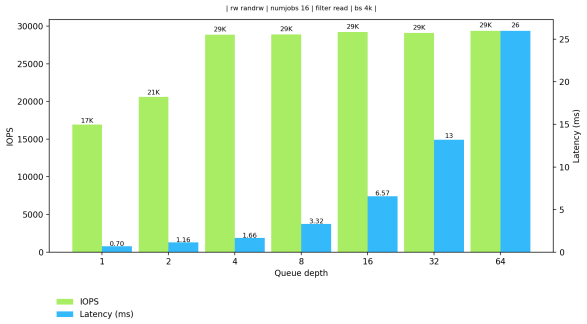
Ainda com EXT4, mudando a ligação de rede de 10G para 100G, não há alterações significativas do desempenho. Verifica-se maior consistência entre as diferentes configurações, embora transpareça a mesma nuance quando se analisa a passagem de 100G e MTU 1500 (figura 4.8c) para 100G e MTU 9000 (4.8c) com uma ligeira diminuição dos máximos obtidos, à semelhança do que aconteceu na leitura de partilha ZFS.



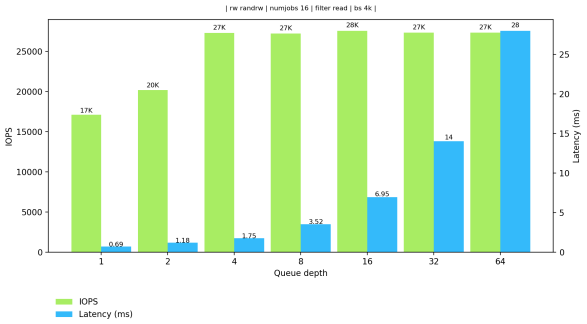
(a) 10G e MTU 1500



(b) 10G e MTU 9000



(c) 100G e MTU 1500



(d) 100G e MTU 9000

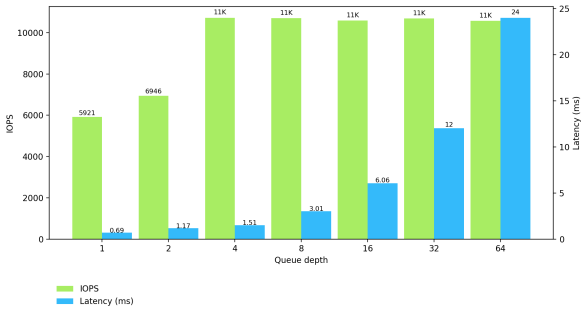
Figura 4.8: IOPS e latência em leitura de EXT4 em acesso remoto

4.6.3 Desempenho da Escrita

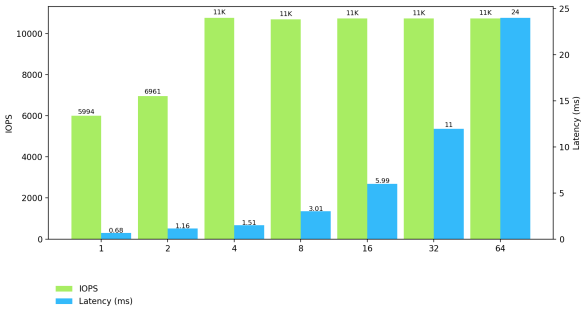
As figuras 4.9 e 4.10 apresentam o desempenho, em termos de IOPS e latência, da escrita em partilhas NFS formatadas com ZFS e EXT4, respetivamente.

Para o uso de ZFS, a análise das figuras 4.9a e 4.9b revela que alternar entre MTU 1500 ou 9000 quando a ligação subjacente é 10G tem um impacto praticamente nula ou insignificante, quer em termos de IOPS quer em termos de latência. Trocando a ligação 10G por uma 100G, verifica-se que o nível de desempenho se mantém para o cenário em que se usa MTU 1500, mas quando se passa para MTU 9000, há uma quebra de desempenho, como já se tinha verificado no caso da leitura (figuras 4.7c e 4.7d).

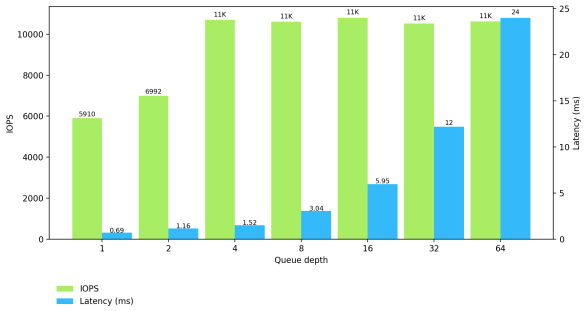
Por fim, no caso da escrita em partilhas formatadas com EXT4, neste caso não há impacto da alteração de MTU e o impacto da mudança de rede (10G vs 100G) é residual. A ligeira diminuição de desempenho verificada quando se passa de MTU 1500 para MTU



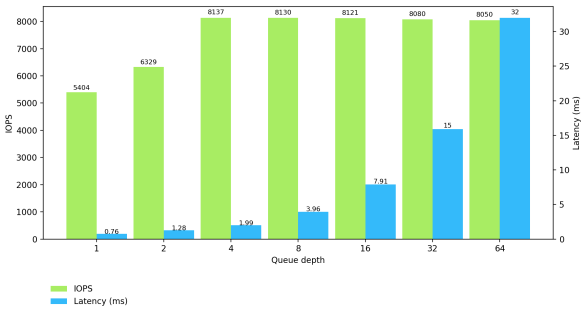
(a) 10G e MTU 1500



(b) 10G e MTU 9000



(c) 100G e MTU 1500



(d) 100G e MTU 9000

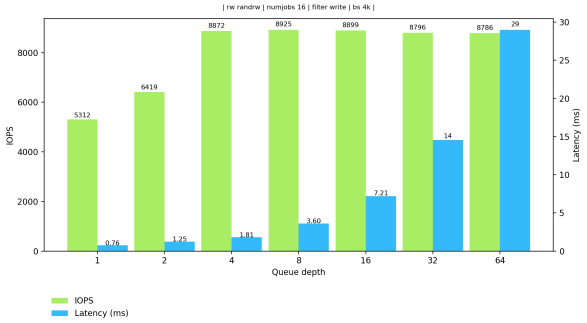
Figura 4.9: IOPS e latência em escrita de ZFS em acesso remoto

9000, é consistente com o padrão já verificado nas figuras 4.7, 4.8 e 4.9.

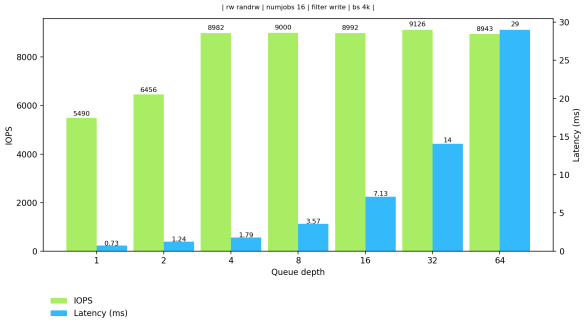
4.6.4 Comparação EXT4 vs ZFS

De forma a se ter uma visão de mais alto nível, os dados constantes nos gráficos das figuras 4.7, 4.8, 4.9 e 4.10, relativos ao acesso a um sistema de ficheiros remoto usando ZFS ou EXT4, foram sintetizados em duas tabelas: a tabela 4.2 corresponde à média de IOPS obtida agregando todas as leituras dos diferentes *iodepth* e a tabela 4.3 representa os valores de latência para os mesmos parâmetros. Estas tabelas mostram uma consistência de comportamento quer no tipo de operação (leitura ou escrita) quer na métrica em análise (IOPS ou latência). As duas tabelas evidenciam também o comportamento contra-intuitivo do sistema para a combinação 100G e MTU 9000, já mencionado anteriormente.

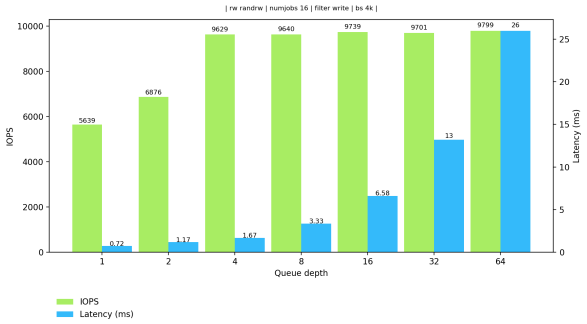
Em suma: para um cenário de acesso via NFS a uma partilha remota, é preferível,



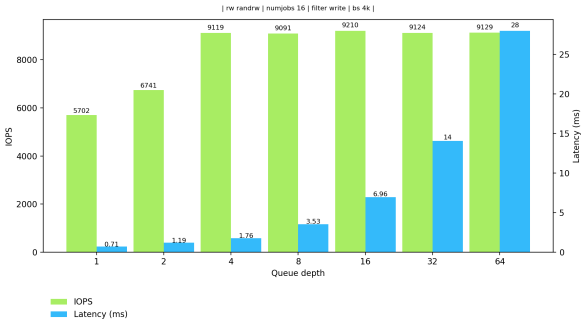
(a) 10G e MTU 1500



(b) 10G e MTU 9000



(c) 100G e MTU 1500



(d) 100G e MTU 9000

Figura 4.10: IOPS e latência em escrita de EXT4 em acesso remoto

em termos de desempenho, que a mesma esteja formatada com ZFS, excepto quando se usa rede 100G e MTU 9000, cenário em que um sistema de ficheiros EXT4 oferece melhor desempenho. Em traços gerais, isto contraria o observado com armazenamento local, em que a adoção de ZFS fornecia, invariavelmente um desempenho inferior ao EXT4.

Tabela 4.2: IOPS (x1000) no acesso remoto

Operação	Leitura				Escrita			
REDE	10G		100G		10G		100G	
MTU	1500	9000	1500	9000	1500	9000	1500	9000
ZFS	**28,43**	**28,43**	**28,43**	22,14	**9,70**	**9,71**	**9,70**	7,46
EXT4	25,43	24,29	26,14	**24,71**	8,00	8,14	8,72	**8,30**

Tabela 4.3: Latência (ms) no acesso remoto

Operação	Leitura				Escrita			
REDE	10G		100G		10G		100G	
MTU	1500	9000	1500	9000	1500	9000	1500	9000
ZFS	**6,76**	**6,75**	**6,89**	8,96	**6,92**	**6,76**	**6,91**	8,99
EXT4	8,22	8,20	7,49	**8,01**	8,23	8,21	7,50	**8,02**

4.7 Armazenamento Remoto via NVMe-over-Fabrics

Sendo a tecnologia NVMe-over-Fabrics umas das tendências da indústria em termos de evolução dos sistemas de armazenamento, quer pelo desempenho, quer pelas capacidades e funcionalidades de gestão, torna-se relevante fazer um estudo das possibilidades que a mesma oferece no âmbito do hardware do cluster: NVMe-over-TCP e NVMe-over-RDMA.

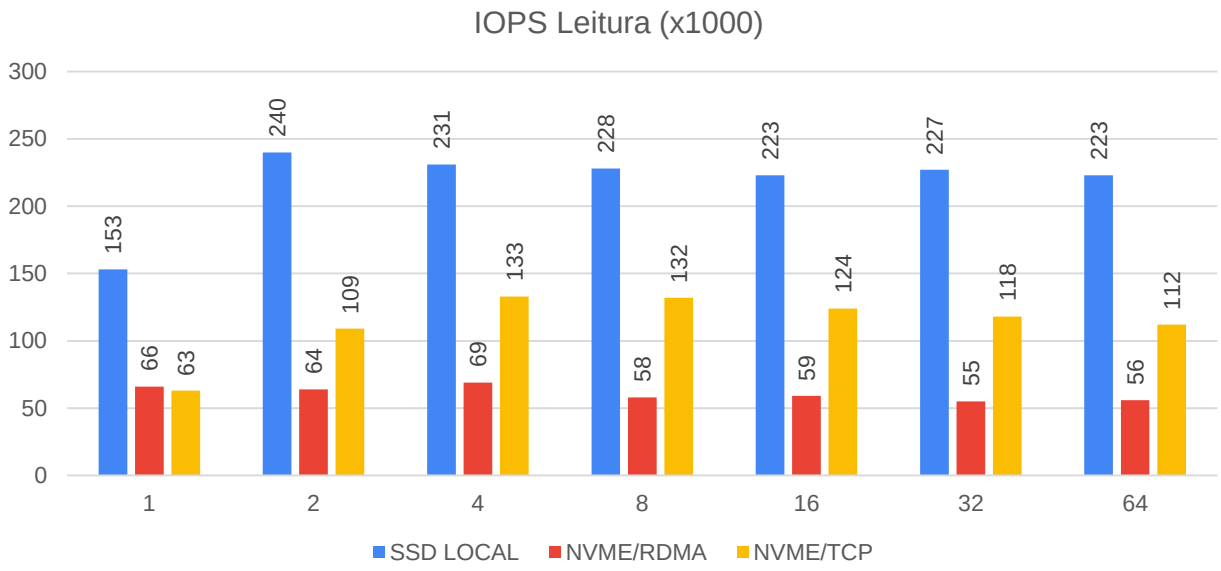
De forma a obter uma perceção da penalização que a rede inevitavelmente introduz no acesso a um SSD remoto (pese embora o baixo nível das variantes NVMe-oF usadas), foi primeiro avaliado o desempenho do SSD NVMe da máquina cliente. De seguida, foi avaliado o desempenho do acesso a um só SSD NVMe remoto (sito na máquina com o FlashArray), através dos protocolos NVMe-over-TCP e NVMe-over-RDMA, usando a ligação 100G e MTU 9000. Esta avaliação restringiu-se ao nível RAW, não se tendo formatado os volumes NVMe local e remotos com qualquer sistema de ficheiros.

Os gráficos das figuras 4.11 e 4.12 permitem comparar os vários cenários testados, apresentando padrões e valores muito distintos daqueles observados em secções anteriores.

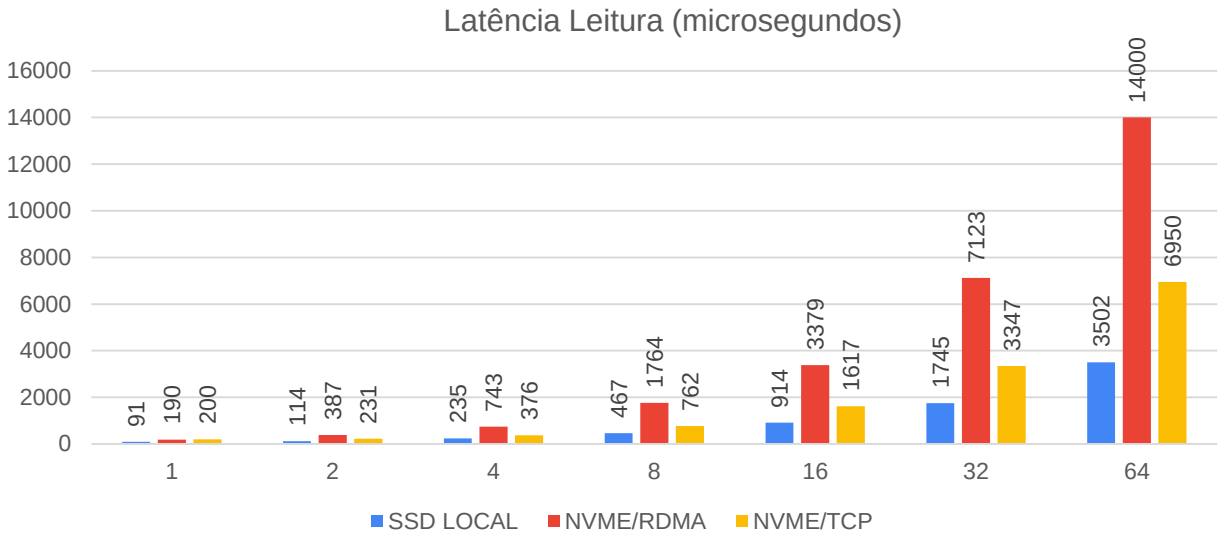
4.7.1 Leitura RAW

Analisando os resultados da leitura (figura 4.11), para o acesso local verifica-se que para uma profundidade de fila de 2, obtém-se logo o rendimento máximo do dispositivo em termos de IOPS. A partir daí há uma muito ligeira diminuição de IOPS, à medida que aumenta a profundidade da fila (e um correspondente aumento da latência).

Para o acesso a um SSD remoto, com NVMe-over-TCP, os IOPS e a latência pioram,

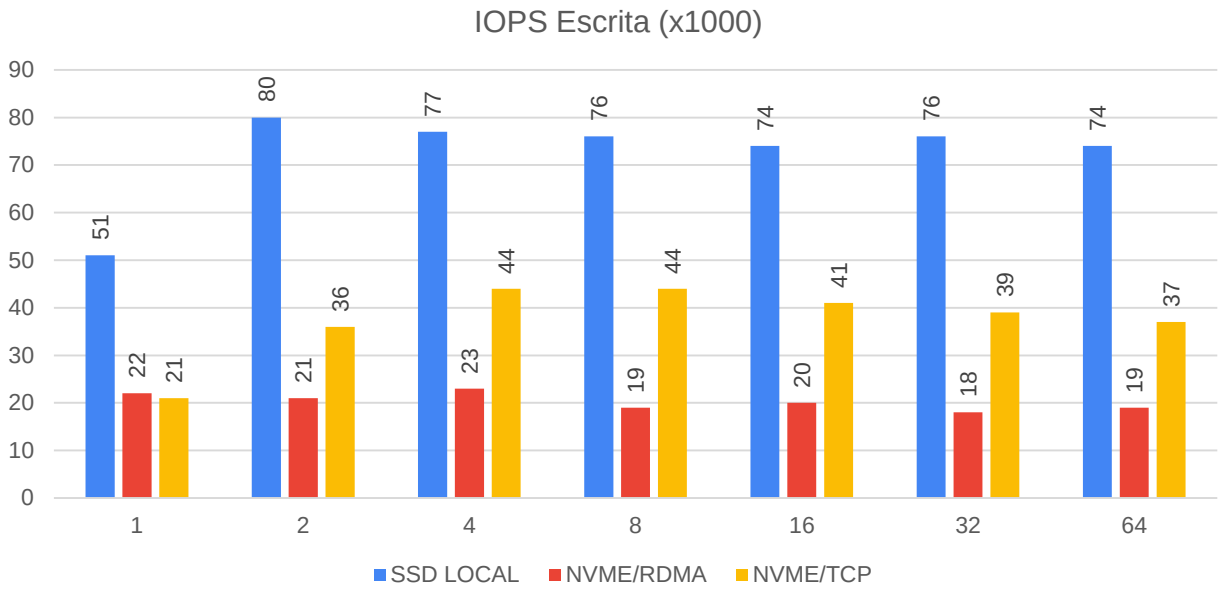


(a) IOPS em leitura

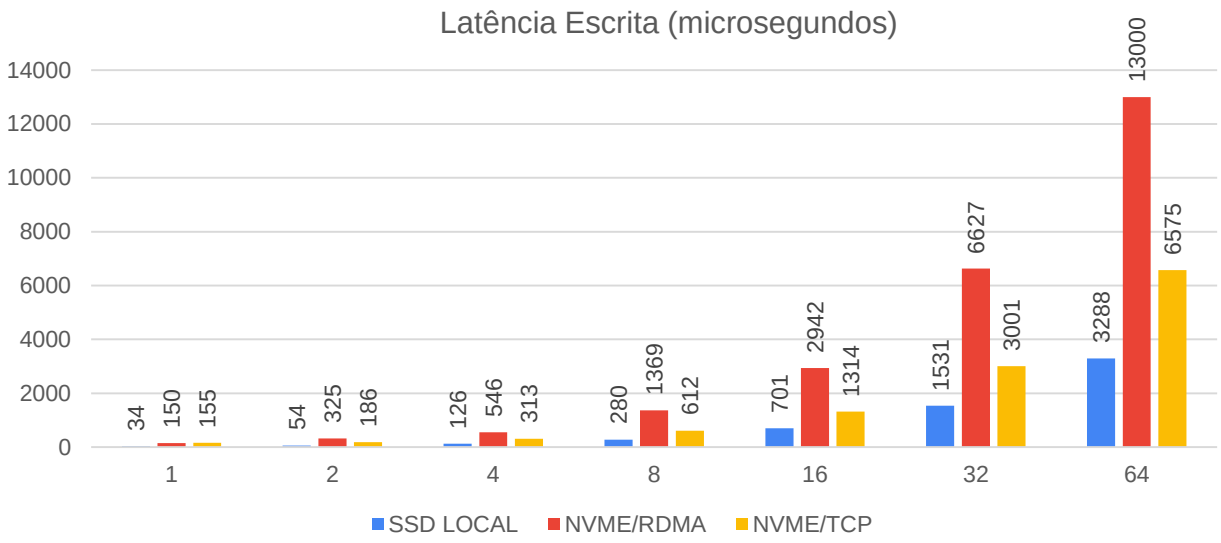


(b) Latência em leitura

Figura 4.11: NVMe: IOPS e latência em leitura de 1 SSD



(a) IOPS em escrita



(b) Latência em escrita

Figura 4.12: NVMe: IOPS e latência em escrita de 1 SSD

como seria de esperar, mas pioram ainda mais com NVMe-over-RDMA, o que é surpreendente, pois o uso de NVMe-over-RDMA deveria proporcionar melhores resultados do que NVMe-over-TCP. Isto sugere que poderá haver afinações a fazer na implementação.

Em traços gerais, face a uma leitura local, a leitura via NVMe-over-TCP impõe uma penalização de cerca 50%, quer em termos de IOPS, quer em termos de latência.

4.7.2 Escrita RAW

Focando agora os resultados da escrita (figura 4.12), o padrão observado é similar ao exibido pela leitura, quer em termos de IOPS, quer em termos de latência, incluindo a prevalência paradoxal do NVME-over-TCP face ao NVME-over-RDMA, e também a penalização de cerca de apenas 50% do NVME-over-TCP face à escrita local.

Em termos de latência, de notar o relativo baixo *overhead* adicional introduzido pela rede. Se tivermos em conta os valores para uma profundidade de fila de 1, a latência local em leitura é de 92 microsegundos e a latência quando se acede ao mesmo dispositivo é de 190 microsegundos, ou seja, o acesso via rede usando RDMA apenas adicionou cerca de 100 microsegundos e até a uma profundidade de fila de 8 e 16 trabalhos, ou seja 128 operações simultâneas, os valores de latência mantém-se abaixo de um milisegundos.

4.8 Outros Testes

Dado o atual sistema de armazenamento do cluster estar em produção e não ser possível realizar medições do seu desempenho sem impactar o bom funcionamento do mesmo, foi necessário arranjar uma forma de colocar em perspetiva os resultados obtidos.

Com o objetivo de contextualizar os resultados anteriormente apresentados foram realizados testes em dois sistemas adicionais muito distintos: i) uma máquina virtual Ubuntu fornecida pelo Centro de Comunicações do IPB, com disco virtual assente num volume importado a partir de uma SAN DELL EMC; ii) uma máquina física semelhante ao servidor de armazenamento de produção do cluster (embora com discos SATA, e não SAS).

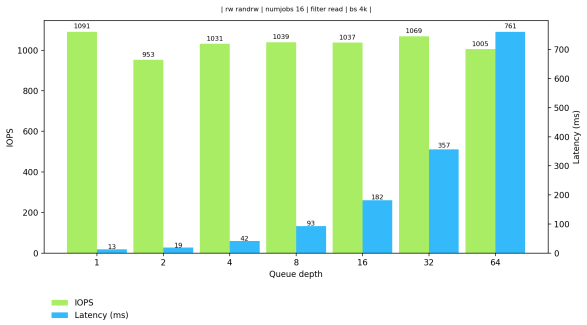
Os resultados dos testes na máquina virtual constam da Figura 4.13. Embora não se trate de um teste isento de ruído, dado o uso simultâneo da SAN por outros sistemas clientes, os seus resultados não deixam de ser interessantes: em termos de desempenho em escrita, os valores registados são significativamente melhores do que os observados na escrita remota via NFS – rever Figuras 4.9 e 4.10; no entanto, em termos de leituras, a diferença de desempenho inverte-se, de forma muito significativa, com a SAN proporcionando valores inferiores, em uma ordem de magnitude, aos observados na leitura remota via NFS – rever Figuras 4.7 e 4.8. Uma possível explicação para esta discrepância poderá ser o facto de uma SAN, sendo um sistema de armazenamento hierárquico (*tiered*), incluir dispositivos / níveis de *caching* que amortecem consideravelmente o impacto da escrita, mesmo de blocos modificados pela primeira vez, ao passo que uma primeira leitura de um bloco não dispensa o acesso ao nível mais baixo da hierarquia.

Nas figuras 4.14 e 4.15 faz-se a contraposição dos resultados dos testes realizados com discos SATA em modo RAW, face aos testes realizados no FlashArray com apenas 1 disco NVMe. Os resultados são também interessantes e até inesperados. Em termos de latência, tanto em leitura como em escrita, o sistema com discos SATA apresenta melhores valores; isto pode-se explicar pelo facto de serem mais dispositivos mas, sobretudo pelo facto de terem sido executadas muito menos operações. Em termos de IOPS, observa-se o inverso: uma vantagem esmagadora do SSD NVMe, independente do *iodepth* ou tipo de operação; esta discrepância pode ser explicada pela natureza do teste, ou seja, leituras e escritas aleatórias nada favoráveis ao modo de operar dos discos rígidos magnéticos.

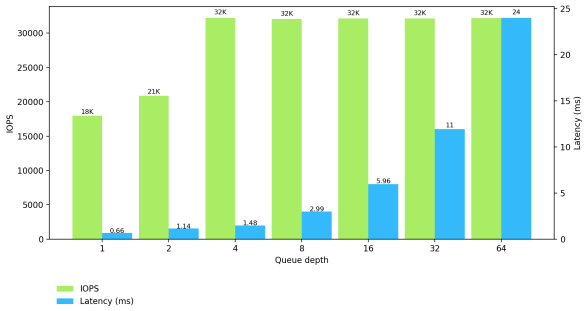
4.9 Conclusões

Ao fazer a análise do comportamento de um sistema de armazenamento, está-se a analisar um sistema composto de várias *stacks*, com vários protocolos e componentes.

Um dos primeiros factos a reter na análise feita neste capítulo, é que a introdução de um sistema de ficheiros reduz significativamente a capacidade nominal (em termos de desempenho) de um sistema de armazenamento. No entanto, essa redução depende muito

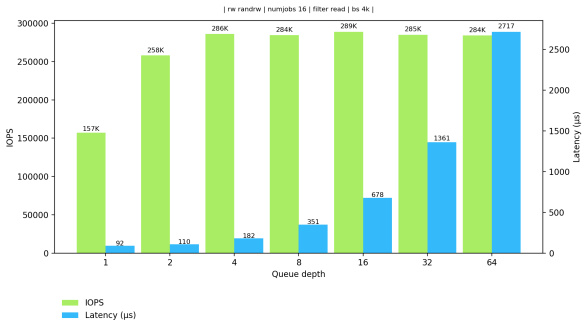


(a) Disco Virtual de SAN - Leitura

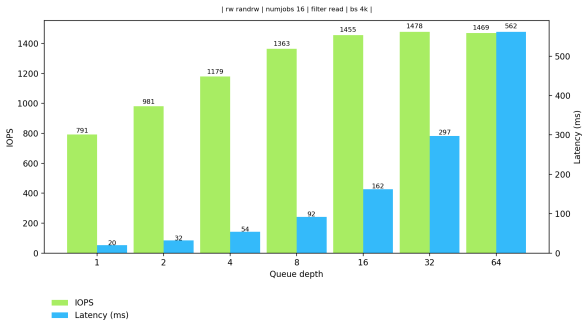


(b) Disco Virtual de SAN - Escrita

Figura 4.13: Desempenho de Disco Virtual de SAN

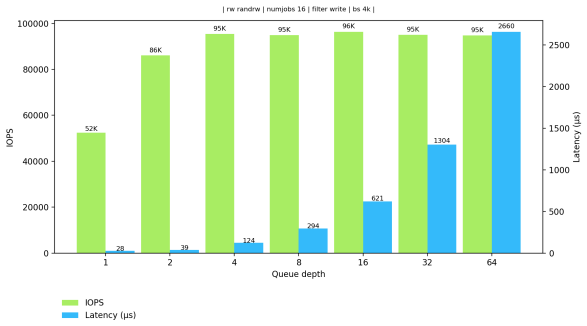


(a) 1 Disco NVME - Leitura RAW

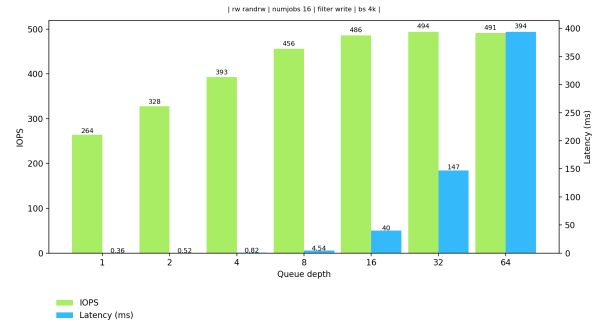


(b) 8 Discos SATA - Leitura RAW

Figura 4.14: Comparação 8 Discos SATA vs 1 NVME - Leitura



(a) 1 Disco NVME - Escrita RAW



(b) 8 Discos SATA - Escrita RAW

Figura 4.15: Comparação 8 Discos SATA vs 1 NVME - Escrita

do sistema de ficheiros utilizado. Neste caso analisaram-se os sistemas de ficheiros EXT4 e ZFS, em ambiente local e posteriormente com acesso via rede. Em acesso local, e para o ambiente de testes montado, o sistema de ficheiros ZFS (em modo assíncrono) apresenta

cerca de 50% do desempenho do sistema de ficheiros EXT4. Se considerarmos o modo síncrono, o ZFS apresenta cerca de 25% do desempenho do EXT4.

Em termos de acesso aos mesmos sistemas, via rede através de uma partilha NFS, a diferença entre sistema de ficheiros é esbatida e até em alguns casos invertida.

De forma a garantir que os dados do acesso via rede são fidedignos, foi realizada uma análise ao seu desempenho de forma isolada. Dessa análise resultou que em termos de ligações 10Gbps, os resultados são extremamente consistentes e lineares, há cerca de uma diferença de 5% de desempenho entre a utilização de MTU 1500 e MTU 9000, com vantagem para MTU 9000. Em termos de ligação 100Gbps, apenas quando usamos 8 threads conseguimos obter o máximo de 86Gbps, ao passo que para MTU 9000 conseguimos obter um máximo de cerca de 98Gpbs usando apenas 4 threads. De facto, o uso de vários trabalhos em simultâneo é fundamental para obtermos um desempenho próximo do máximo teórico de 100Gbps. A utilização de CPUs com velocidades de relógio superior poderá contribuir para melhores desempenhos em 1, 2 e 4 trabalhos simultâneos.

Em termos de análise no acesso aos sistemas de ficheiros via rede, verifica-se que as diferenças de desempenho, constatadas quando se faz medições locais, desaparecem ou são mesmo ligeiramente revertidas no acesso via rede.

Assim, o facto de se usar 10Gbps ou 100Gbps, ou se usar MTU 1500 ou MTU 9000, parece introduzir diferenças mínimas. A combinação 100Gbps com MTU 9000, que à partida seria aquela que apresentaria melhores resultados, revelou-se idêntica as demais combinações e em boa parte dos casos, pior.

Apesar da diferença não ser muito significativa, quando acedido pela rede, o sistema de ficheiros ZFS parece ter um desempenho ligeiramente melhor do que o sistema de ficheiros EXT4, invertendo a hierarquia em relação ao comportamento em ambiente local. Esta tendência tem uma exceção, para a combinação 100Gbps e MTU 9000, o acesso remoto a ZFS é mais penalizado pelo uso desta combinação, e apenas neste caso o EXT4 mostra-se como o sistema de ficheiros com melhor desempenho.

Por fim, analisou-se o acesso via NVMe-over-Fabrics a um só disco no servidor remoto. Primeiro, estabeleceu-se uma base de comparação fazendo uma medição do desempenho

do disco em contexto de acesso local e, posteriormente, analisou-se o seu comportamento usando NVMe-over-RDMA e NVMe-over-TCP. Relativamente aos resultados obtidos, podemos dizer que, grosso modo, o acesso via NVMe-over-TCP é de cerca de 50% do desempenho do dispositivo aquando do acesso local e em termos de NVMe-over-RDMA é de cerca de 50% do NVMe-over-TCP. Ora, seria de esperar que o desempenho via RDMA fosse superior ao desempenho via TCP. Este facto leva a crer que seriam necessárias algumas afinações que não estariam presentes nas definições por omissão.

4.10 Epílogo

Os dados recolhidos ao longo deste capítulo forneceram uma nova perspetiva entre o que são os valores nominais e os valores que acabamos por verificar em sistemas concretos.

As referidas discrepâncias entre o que seria expectável e o que na realidade se verificou, apontam ainda no sentido de ser necessária uma afinação dos parâmetros por omissão das pilhas protocolares envolvidas.

Capítulo 5

Alterações Preconizadas

5.1 Preâmbulo

Depois de feitas todas as medições pertinentes e passíveis de serem feitas, é necessário agora tirar as devidas ilações e fazer uma reflexão sobre qual o melhor caminho a seguir.

Há duas vertentes a considerar: a otimização que poderá ser feita com os componentes existentes e as opções a considerar de futuro na evolução do *cluster*.

5.2 Solução preconizada

5.2.1 Computação

Do estudo realizado pode-se concluir que, sobretudo para aplicações mais *CPU Bond*, a velocidade de relógio desempenha um papel fundamental, por vezes suplantando o número de cores como fator crítico para o desempenho em certas cargas de trabalho.

As medições ao comportamento da *firewall* com o *Multithreading* ativo ou não, são reveladoras de que em ambientes virtualizados até poderá haver um beneficio marginal mas que a mesma implementação em *bare metal* será sempre mais eficiente.

5.2.2 Topologia de Rede

No caso da definição da topologia de rede, no seu sentido mais lato, são várias as conclusões que se pode tirar a partir dos resultados apresentados no capítulo a ela dedicado (Otimização da Conexão Edge) e que podem apontar para uma solução mais equilibrada.

Em termos de análise da pertinência do upgrade de placas 10Gbps para placas 100Gbps, pode-se concluir que os ganhos ficam aquém do expectável. Apesar de se obterem valores muito perto dos máximos nominais (a partir de 4 threads), o uso de placas 100Gbps seria mais recomendável caso o armazenamento partilhado fosse capaz de saturar de forma consistente, ou pelo menos significativa, as ligações 10Gbps.

Outro dos cenários potenciador do uso de placas de 100Gbps, seria a opção por um cenário de infraestrutura HyperConverged, com a opção por um *cluster ceph* como principal opção para o armazenamento. O facto de uma infraestrutura ceph escalar sobretudo horizontalmente e não verticalmente, possibilitaria resolver problemas de resiliência ao aumento do número de cargas que o cluster poderia acomodar mas não no desempenho individual de determinada carga de trabalho.

Naturalmente, a opção por interfaces de rede 100Gbps dá margem para a infraestrutura como um todo evoluir, quer nas tecnologias utilizadas, quer no desempenho e qualidade do serviço oferecido.

Em termos de firewall, os resultados foram surpreendentes, conforme analisado do capítulo dedicado à topologia de rede. Atendendo à atual tipologia da carga de trabalho que *cluster* normalmente recebe, poderá não se justificar a troca de uma firewall pfSense por um outro sistema operativo ou *appliance*, como por exemplo o vyOS, que foi brevemente testado. Apesar do desempenho ser melhor, a troca implicaria um período de aprendizagem significativo.

Um ponto que parece ser consensual e corroborado pelos diferentes testes realizados é que a utilização de MTU 9000 trás ganhos substanciais em todas as vertentes analisadas, quando testamos apenas o desempenho de rede de forma isolada. Já o mesmo não se verifica em termos de armazenamento partilhado.

5.2.3 Armazenamento

O tema do armazenamento foi talvez o mais exaustivamente analisado, pois seria esta a área onde haveria mais ganhos a obter e mais diversidade de opções/configurações poderiam ser consideradas e que eventualmente poderiam catapultar a plataforma para um significativo salto no seu desempenho.

Foram testadas e analisadas as diferentes hipóteses equacionadas, sendo que das diferentes alternativas testadas, não houve nenhuma que realmente se destacasse positivamente das outras ou da solução já existente.

Na verdade, não há uma opção para armazenamento remoto centralizado (open-source ou gratuita), que seja ao mesmo tempo capaz de tirar partido de uma *array* de SSDs NVMe, e que seja compatível com as plataformas de virtualização equacionadas.

Nas conclusões do capítulo dedicado ao armazenamento, são esmiuçadas as diferentes nuances de cada sistema testado, mas querendo de facto maximizar o desempenho, o uso de armazenamento local baseado em SSDs continua a ser melhor opção, embora essa opção tenha outras implicações em termos gestão das máquinas virtuais.

Assim, a opção por um sistema híbrido, com armazenamento local em simultâneo com armazenamento partilhado, será a opção mais flexível, com melhor desempenho à custa de um pouco mais de intervenção dos gestores do *cluster* na criação e manutenção das máquinas virtuais.

O uso de armazenamento local pode ainda facilitar as tarefas de gestão e manutenção do armazenamento partilhado permitindo o *offload* (ainda que temporário) do armazenamento partilhado contribuindo assim para a resiliência e *uptime* do *cluster*.

5.2.4 Plataforma de Virtualização

A latência no acesso ao armazenamento partilhado parece ser o grande calcanhar de Aquiles da atual implementação baseada em oVirt do *cluster* do CeDRI. A introdução de placas de rede de 100Gbps não melhora significativamente os tempos de latência e o desempenho do armazenamento partilhado.

Sendo a carga de trabalho de um significativo conjunto de máquinas virtuais essencialmente leituras e escritas aleatórias de poucos dados em cada operação, facilmente se percebe que a natureza de protocolos como iSCSI ou NFS, em que se tem apenas uma fila de operações por sistema, não será o ideal num sistema tão multiplexado.

O desempenho do NVMe-oF/TCP com resultados encorajadores no acesso via rede a um dispositivo NVMe, fica aquém das expectativas ao não permitir, com software *open-source* disponível, agregar um *array* de dispositivos NVMe.

Das alternativas consideradas, apenas o Proxmox suporta armazenamento partilhado em simultâneo com armazenamento local. Esta característica acaba por ser determinante na escolha do Proxmox como plataforma recomendada para o caso em apreço.

A estratificação que poderá ser implementada, recorrendo a diferentes níveis de desempenho, capacidades e tecnologias do armazenamento, poderá representar um salto significativo no desempenho global.

Em termos de evolução futura, a adoção de uma estratégia *Hyperconverged* poderá acrescer resiliência e desempenho a toda a infraestrutura.

Em termos funcionais a adoção do Proxmox Backup Server como solução de backup e toda a integração que esta solução tem com uma infraestrutura Proxmox, reforça a opção pelo Proxmox como a escolha certa.

5.3 Epílogo

À medida que as análises e medições foram decorrendo, foram identificados potenciais pontos de melhoria. Foram confirmadas algumas das perceções iniciais e desfeitas outras. No entanto, a escolha de uma nova plataforma de virtualização parece ser a decisão certa.

Uma das ideias que também saiu reforçada é a necessidade de uma constante monitorização do sistema e do seu desempenho, ou seja, não assumir que o desempenho nominal dos diferentes componentes que compõem o *cluster* mantém o mesmo à medida que se vão introduzindo novos componentes, sejam eles hardware ou software. A implementação de mecanismos de monitorização automática será portanto uma mais valia a considerar.

Capítulo 6

Conclusões

O estudo de um tema como aquele que este trabalho aborda, implica alguma contenção na latitude e profundidade do trabalho desenvolvido, dada a abrangência do objeto de estudo, das diferentes pilhas de protocolos, do sem número de variáveis envolvidas, das diferentes correlações passíveis de serem estabelecidas. Esta necessidade requereu alguma disciplina, que nem sempre foi fácil de manter, quanto ao objetivo final.

O objetivo final, apresentado nos capítulos introdutórios, era de analisar o grau de correlação entre os valores teóricos e nominais dos diferentes componentes de uma plataforma de virtualização e identificar possíveis estrangulamentos no desempenho. Nas secções seguintes, são apresentadas as conclusões referentes a cada item analisado.

6.1 Topologia de Rede

Sobretudo ao longo do capítulo 3, e brevemente no capítulo 4, foram feitos diversos estudos relacionados com o desempenho da rede. Pode-se encontrar uma análise mais detalhada nas conclusões de cada capítulo, mas importa sublinhar aquelas mais dignas de registo:

- *SMT/Multithreading* - O seu uso parece beneficiar sobretudo ambientes virtualizados. No caso da firewall, a opção por uma implementação em *bare metal* e sem *multithreading* revelou-se a opção de melhor desempenho. No caso de se optar por

ambientes virtualizados, o *multithreading* é uma mais valia e deve estar ativo, uma vez que traz vantagens em termos de desempenho, ainda que marginais.

- Número de *cores* - uma das conclusões mais evidentes, ao longo dos diferentes *benchmarks* realizados, foi que a velocidade de relógio ainda desempenha um papel muito importante, sobretudo quando consideramos apenas o desempenho de uma máquina virtual. Observou-se que mesmo quando a aplicação ou teste tira partido de vários cores, há sempre um ou outro que parece ser o fator limitativo, uma vez que regista sempre ocupações de 100% durante a realização do teste.

- *MTU* - A opção pelo uso de MTU 9000 em detrimento do uso de MTU 1500, parece ser a opção mais acertada. Esta afirmação é suportada pela consistência revelada nas medições realizadas no capítulo 3. No entanto, no capítulo 4, ao analisar o comportamento do sistema de armazenamento remoto, e fazendo a análise combinanda de MTU 1500/9000 com interfaces de rede 10Gbps/100Gbps, verifica-se que a combinação de 100Gbps com MTU 9000 apresenta pior desempenho que as restantes combinações. Ora, este comportamento não seria de todo expectável. Não sendo óbvia a causa, uma das possíveis explicações poderá ser o facto de a combinação dos 100Gbps com MTU 9000 gerar mais dados que os *buffers* da *stack* de armazenamento são capazes de lidar, pelo menos de forma tão eficiente.

- *firewall* - Como referido no início deste capítulo, um dos objetivos deste era o de identificar áreas ou componentes da plataforma que apresentassem oportunidades para melhorar o desempenho. Ora, a firewall parece ser um clara oportunidade de melhoria. Se por um lado se verifica que a realização de NAT não seria fator de preocupação, uma vez que não parece haver impacto significativo no desempenho, mesmo quando se usa várias *streams* de dados, por outro, a inconsistência revelada por algumas das medições realizadas, leva a concluir que haja potencial por explorar.

  A introdução de outros sistemas operativos como o vyOS ou Ubuntu, nas medições e testes realizados (inicialmente não previstos), mostrou que a afinação dos sistemas

operativos, poderá render ganhos de desempenho significativos. Mesmo no caso vyOS e do Ubuntu, sendo sistemas operativos baseados em Debian, mostraram algumas diferenças de desempenho, embora não tão significativas como o pfSense.

6.2 Armazenamento

A análise das diferentes alternativas para o sistema de armazenamento, revelaram dados que à partida não seriam expectáveis. Ao longo do capítulo 4 foram feitas as análises dos diferentes sistemas em pormenor, mas de forma resumida podemos apontar as seguintes conclusões como as mais relevantes:

- Sistema de ficheiros - Em acesso local, de facto o uso de EXT4 parece ter uma vantagem em relação ao ZFS assíncrono e sobretudo em modo síncrono, quer em termos de IOPS quer em termos de latência.

  Em acesso remoto, em qualquer das combinações testadas, o *ZFS* acaba por ter melhor desempenho. A introdução de interfaces de rede de 100Gbps não teve o efeito esperado, e a combinação de 100Gbps com MTU 9000, que potencialmente ofereceria o melhor desempenho, revelou-se mesmo a pior combinação. Esta inversão da ”hierarquia” no desempenho dos sistemas de ficheiros, pela introdução do acesso via rede, carece uma análise mais aprofundada, já fora do escopo deste trabalho.

  Os protocolos NVMe-oF estudados ofereceram de facto uma nova perspetiva em termos de desempenho. O facto de não haver software *Open Source* e/ou gratuito capaz de agrupar vários dispositivos num único *target*, torna esta opção inviável como principal solução de armazenamento do *cluster*, o que não invalida o seu uso discricionário de atribuir dispositivos NVMe a nós de uma forma *ad hoc*.

  A diferença entre o desempenho local do *array* de discos e o mesmo desempenho quando acedido remotamente, faz pesar o prato da balança muito significativamente para uma solução híbrida. Se por um lado um sistema de armazenamento partilhado permite uma melhor otimização de espaço e facilidade de gestão, é igualmente

verdade que constitui um risco adicional derivado da concentração de todas as máquinas virtuais no mesmo sistema.

Como preconizado no capítulo 5, a flexibilidade do Proxmox como plataforma de virtualização permitiria concretizar essa abordagem híbrida que tira partido das vantagens do armazenamento local e remoto.

6.3 Alterações Preconizadas

Ao longo do capitulo 5 foram descritas em mais pormenor as alterações que os dados recolhidos parecem indicar como as relevantes para a melhoria do desempenho da plataforma como um todo.

A principal alteração é a mudança da plataforma de virtualização de forma a ser possível utilizar armazenamento local e partilhado nos diferentes nós de computação. Esta mudança, implica mudar de oVirt para Proxmox. Haverá outras vantagens, mas a possibilidade de usar armazenamento local é mesmo a mais relevante e com mais impacto no desempenho global.

Estas mudanças devem no entanto, ser validadas posteriormente, usando a mesma metodologia de testes para verificar se de facto esta conjetura teórica se verifica na prática.

Os protocolos NVMe-oF apresentam um enorme potencial, mas neste momento a exploração desse potencial implicaria o uso de uma solução comercial.

O uso das placas 100Gbps não representou a mais valia que se julgava teriam em termos de potenciar os diferentes componentes do *cluster*. A sua adoção em substituição das placas 10Gbps será mais uma opção de futuro do que uma solução de presente.

6.4 Trabalho futuro

Em termos de trabalho futuro, o acompanhamento do estado da arte das soluções NVMe-oF, sobretudo no campo *Open Source*, será uma das prioridades.

A otimização do pfSense ou a avaliação de alternativas poderá trazer também ganhos significativos no desempenho da firewall. Os testes realizados revelaram discrepâncias que indicam que haverá potencial para melhorias.

A alteração de plataforma de virtualização proposta, implicará a mudança da distribuição linux de base, ou seja, mudar de CentOS (oVirt) para Debian (Proxmox). Isso implicará diferentes parâmetros por omissão na pilha de software utilizada. Como vimos ao longo deste trabalho, essas opções podem ter um impacto significativo.

A automação dos testes de monitorização de desempenho dos diferentes componentes da plataforma seria o próximo passo na evolução deste trabalho.

No caso específico do cluster do CeDRI, a mudança de oVirt para Proxmox implicará a perda do acesso ao portal self-service, funcionalidade nativa ao oVirt. De forma a colmatar esta perda, é a atrativa a possibilidade de elaborar uma plataforma web equivalente para Proxmox, capaz de receber os pedidos de gestão de máquinas virtuais e traduzir os mesmo em chamadas à API do Proxmox ou a playbooks Ansible, com níveis de privilégio adequados a utilizadores de nível não-administrativo.

A opção por uma solução (comercial ou gratuita, caso seja desenvolvida) capaz de oferecer um array NVMe remoto (e não apenas um SSD), a um nó de virtualização, através de NVMe-OF/TCP poderia representar um salto muito significativo na flexibilidade da gestão dos dispositivos de tipo SSD, evitando a sua dispersão pelos vários nós do cluster, e permitindo uma gestão mais eficaz do espaço de armazenamento de tipo Flash, com percas mínimas de desempenho, e dispensando o recurso a soluções *Hyperconverged*.

Apêndice A

Otimização da Firewall Edge - Medições em Detalhe

A.1 Impacto do Multithreading na Firewall Física vs Virtualizada

Tabela A.1: pfSense Virtual sem SMT

MTU	1500					9000				
threads iperf	1	2	4	8	Média	1	2	4	8	Média
Largura de Banda	3,184	3,182	3,264	3,284	3,229	9,870	9,848	9,872	9,890	9,870
Carga i7Server	0,042	0,080	0,124	0,160	0,102	0,222	0,192	0,300	0,426	0,285
Carga i7Client	0,024	0,036	0,042	0,064	0,042	0,152	0,328	0,276	0,258	0,254
Carga Firewall	0,546	1,020	1,020	1,002	0,897	0,558	0,476	0,360	0,592	0,497

Tabela A.2: pfSense Virtual com SMT

MTU	1500					9000				
N.º Threads	1	2	4	8	Média	1	2	4	8	Média
Largura de Banda	3,248	3,322	3,42	3,426	3,354	9,89	9,894	9,892	9,864	9,885
Carga i7Server	0,22	0,31	0,496	0,622	0,412	0,276	0,252	0,364	0,414	0,3265
Carga i7Client	0,122	0,176	0,138	0,136	0,143	0,268	0,23	0,398	0,336	0,308
Carga Firewall	0,704	0,838	0,936	0,936	0,8535	0,33	0,312	0,214	0,378	0,3085

Tabela A.3: pfSense Físico sem SMT

MTU	1500					9000				
MultiThreading	1	2	4	8	Média	1	2	4	8	Média
Largura de Banda	5,334	5,368	5,414	5,426	5,386	9,9	9,9	9,9	9,9	9,9
Carga i7Server	0,418	0,438	0,46	0,476	0,448	0,264	0,332	0,444	0,42	0,365
Carga i7Client	0,182	0,234	0,28	0,218	0,229	0,202	0,208	0,276	0,276	0,2405
Carga Firewall	0,5798	0,438	0,802	0,838	0,664	0,422	0,438	0,398	0,38	0,4095

Tabela A.4: pfSense Físico com SMT

MTU	1500					9000				
MultiThreading	1	2	4	8	Média	1	2	4	8	Média
Largura de Banda	5,282	5,298	5,338	5,366	5,321	9,9	9,9	9,9	9,9	9,9
Carga i7Server	0,254	0,292	0,296	0,61	0,363	0,236	0,234	0,29	0,63	0,3475
Carga i7Client	0,0482	0,164	0,24	0,138	0,148	0,21	0,296	0,362	0,438	0,3265
Carga Firewall	0,692	0,748	0,708	0,822	0,743	0,352	0,374	0,338	0,43	0,3735

Apêndice B

Código

B.1 Script para realização de testes de rede

```
#!/bin/bash
clear
echo ".....testing started....."

threads=(1 1 1 1 1 2 2 2 2 2 4 4 4 4 4 8 8 8 8 8 0)

data=$(date +%y_%m_%d_%H_%M_%S)

file=$1_$data.txt
logfile=$1_$data.log
echo $data
echo $data > $1_$data.txt

load=‘ssh root@10.2.2.254 top -n 1 | grep load | awk '{print substr($6,
    ↪ 1, length($6)-1)}'‘

for t in ${threads[@]}; do
```

```
load=‘ssh root@10.2.2.254 top -n 1 | grep load | awk ’{print␣substr($6,
    ↪ ␣1,␣length($6)-1)}’‘
while [ $load != "0.00" ]
do
echo -n ">$load"
sleep 15
load=‘ssh root@10.2.2.254 top -n 1 | grep load | awk ’{print␣substr($6,
    ↪ ␣1,␣length($6)-1)}’‘
done

echo "----------------------"
echo "Running␣P=$t"
if [ $t == 1 ]
then
bandwidth=‘iperf -c 10.2.2.254 -P $t -t 60 | awk ’END{print␣$7}’‘
else
bandwidth=‘iperf -c 10.2.2.254 -P $t -t 60 | awk ’END{print␣$6}’‘
fi
cpuserver=‘ssh root@10.2.2.254 top -n 1 | grep load | awk ’{print␣
    ↪ substr($6,␣1,␣length($6)-1)}’‘
cpuclient=‘top -n 1 | grep load | awk ’{print␣substr($12,1,␣length($12)
    ↪ -1)}’‘
echo "Client␣-------->" >> $logfile
echo ‘top -n 1 | grep load‘ >> $logfile

echo "T=$t␣␣BandWidth=$bandwidth␣cpuserver=$cpuserver␣cpuclient=
    ↪ $cpuclient" >> $file
done
```

Bibliografia

[1] Hyper-converged architecture | suse defines. https://www.suse.com/suse-defines/definition/hyper-converged-architecture/. (Accessed on 06/20/2023).

[2] iperf - the tcp, udp and sctp network bandwidth measurement tool. https://iperf.fr/. (Accessed on 06/20/2023).

[3] Non-volatile memory express. https://nvmexpress.org/wp-content/uploads/NVMe-over-Fabrics-1.1a-2021.07.12-Ratified.pdf. (Accessed on 06/20/2023).

[4] pfsense® - world's most trusted open source firewall. https://www.pfsense.org/. (Accessed on 06/20/2023).

[5] Ubuntu server - for scale out workloads | ubuntu. https://ubuntu.com/server.

[6] Vyos – open source router and firewall platform. https://vyos.io/.

[7] What is esxi | bare metal hypervisor | esx | vmware. https://www.vmware.com/products/esxi-and-esx.html.

[8] ext4 filesystem — the linux kernel documentation. https://www.kernel.org/doc/html/v4.19/filesystems/ext4/index.html.

[9] Openzfs documentation — openzfs documentation. https://openzfs.github.io/openzfs-docs/.

[10] ovirt | ovirt is a free open-source virtualization solution for your entire enterprise. https://www.ovirt.org/.

[11] P. K. Paul and M. K. Ghose. Cloud computing: Possibilities, challenges and opportunities with special reference to its emerging need in the academic and working area of information science. *Procedia Engineering*, 38:2222–2227, 2012. INTERNATIONAL CONFERENCE ON MODELLING OPTIMIZATION AND COMPUTING.

[12] Proxmox - powerful open-source server solutions. https://www.proxmox.com/.

[13] M. Singh. Virtualization in cloud computing- a study. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 64–67, 2018.

[14] F. Wang, M. Nelson, S. Oral, S. Atchley, S. Weil, B. W. Settlemyer, B. Caldwell, and J. Hill. Performance and scalability evaluation of the ceph parallel file system. In *Proceedings of the 8th Parallel Data Storage Workshop*, PDSW '13, page 14–19, New York, NY, USA, 2013. Association for Computing Machinery.

[15] X. Wang, H. Song, C.-T. Nguyen, D. Cheng, and T. Jin. Maximizing the benefit of rdma at end hosts. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021.

[16] Xcp-ng - xenserver based, community powered. https://xcp-ng.org/.