



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**MARTA LAÍS DE MACEDO DANTAS**

**ANÁLISE DE DESEMPENHO EM UM AMBIENTE OPENSTACK  
HIPERCONVERGENTE**

**CAMPINA GRANDE - PB**

**2023**

**MARTA LAÍS DE MACEDO DANTAS**

**ANÁLISE DE DESEMPENHO EM UM AMBIENTE OPENSTACK  
HIPERCONVERGENTE**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharela em  
Ciência da Computação.**

**Orientador: Professor Dr. Thiago Emmanuel Pereira da Cunha Silva**

**CAMPINA GRANDE - PB**

**2023**

**MARTA LAÍS DE MACEDO DANTAS**

**ANÁLISE DE DESEMPENHO EM UM AMBIENTE OPENSTACK  
HIPERCONVERGENTE**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharela em  
Ciência da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr. Thiago Emmanuel Pereira da Cunha Silva  
Orientador – UASC/CEEI/UFCG**

**Professora Dra. Joseana Macêdo Fachine Régis de Araújo  
Examinadora – UASC/CEEI/UFCG**

**Professor Tiago Lima Massoni  
Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 14 de Fevereiro de 2023.**

**CAMPINA GRANDE - PB**

## **ABSTRACT**

A cloud computing middleware presents a hyperconverged architecture when its computing, virtualization, storage and network services can be deployed, in a consolidated manner, on the same server. Unlike a classic architecture, in which the services need to be on different servers, the hyperconverged architecture allows optimizing the use of available hardware resources by consolidating several services into a smaller number of resources. However, this consolidation can present challenges as it can create contention for resources. In this work, we intend to perform a performance test of a hyperconverged architecture of the OpenStack cloud. This test will consider the typical load served by a large Brazilian e-commerce company.

# Análise de desempenho em um ambiente *OpenStack* hiperconvergente

Marta Laís de Macedo Dantas\*

marta.dantas@ccc.ufcg.edu.br

Universidade Federal de Campina Grande (UFCG)  
Campina Grande, Paraíba, BR

Thiago Emmanuel Pereira

temmanuel@computacao.ufcg.edu.br

Universidade Federal de Campina Grande (UFCG)  
Campina Grande, Paraíba, BR

## RESUMO

Um *middleware* de computação na nuvem apresenta arquitetura hiperconvergente quando seus serviços de computação, virtualização, armazenamento e rede podem ser implantados, de forma consolidada, em um mesmo servidor. Diferente de uma arquitetura clássica, na qual os serviços precisam estar em servidores diferentes, a arquitetura hiperconvergente permite otimizar o uso dos recursos de *hardware* disponíveis ao consolidar vários serviços em um número menor de recursos. No entanto, esta consolidação pode apresentar desafios, pois pode criar contenção pelos recursos, e por consequência reduzir o desempenho do sistema. Neste trabalho, pretende-se avaliar o desempenho de uma arquitetura hiperconvergente da nuvem *OpenStack*. Este teste considerará a carga típica atendida por uma grande empresa de *e-commerce* brasileira.

### ACM Reference Format:

Marta Laís de Macedo Dantas and Thiago Emmanuel Pereira. 2023. Análise de desempenho em um ambiente *OpenStack* hiperconvergente. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUÇÃO

*Data centers* são instalações físicas onde são armazenados, processados e gerenciados grandes volumes de dados e serviços. Empresas como *Amazon*, *Google* e *Microsoft* têm seus próprios *data centers* onde alocam seus serviços de *clouds* públicas, comercializadas para outras empresas e usuários finais. O uso de *data centers* tem crescido significativamente entre empresas brasileiras nos últimos anos [1]. Isso ocorre devido à crescente demanda por serviços de *cloud*, armazenamento de dados e computação em escala. Em 2022, cerca de \$490,3 bilhões foram gastos mundialmente com serviços de *clouds* públicas. A previsão para 2023 é que esses gastos cheguem a \$591,8 bilhões, com um crescimento de 20,7% [2]. Além de utilizar *clouds* públicas, as empresas também optam por *clouds* privadas e públicas em conjunto, a fim de ter mais controle sobre seus dados, garantir a segurança e disponibilidade de seus serviços e reduzir custos operacionais a longo prazo.

Em infraestruturas de *data centers* tradicionais, máquinas diferentes compõem funções distintas. Em contraponto, infraestruturas

convergentes (CI) mesclam serviços, armazenamento, e equipamentos de rede nas mesmas máquinas. Por sua vez, a infraestrutura hiperconvergente (HCI) evolui este modelo. A hiperconvergência é um desenvolvimento adicional dos conceitos e arquiteturas que surgiram para a convergência. Trata-se de produtos completos em que servidores, armazenamento e funções de rede são acomodados em uma única unidade – uma espécie de solução “tudo em um”.

Uma das principais vantagens da HCI é a possibilidade de fornecer serviços com alta disponibilidade, com menos componentes e com menor custo, viabilizado pela virtualização de recursos de armazenamento com o mesmo hipervisor que fornece máquinas virtuais [3]. Além disso, possibilita a composição de um *data center* com máquinas homogêneas, com a mesma disposição de recursos e de *software*, uma vez que cada máquina irá desempenhar o mesmo papel na infraestrutura. Este ponto é importante tanto para a gerência de *hardware* quanto para a padronização de versões de *software*, dada a afinidade de configuração de máquinas.

Para alcançar as vantagens de um ambiente HCI em uma *cloud* privada, é necessário superar os desafios desta arquitetura, sendo o principal deles a constante disputa de recursos computacionais nas máquinas com múltiplas funções. O estudo apresentado neste trabalho busca realizar e analisar um teste de desempenho de uma *cloud* privada *OpenStack* com arquitetura hiperconvergente, considerando a carga típica de uma empresa de *e-commerce* brasileira.

A relevância deste estudo é considerável por apresentar uma análise sobre um ambiente pouco convencional e de material escasso na comunidade. A contribuição principal deste trabalho é a análise dos resultados de testes de desempenho em um ambiente hiperconvergente de uma *cloud* privada com *OpenStack*, envolvendo também o *cluster* de armazenamento distribuído *Ceph*. A partir dos resultados dos testes do *FIO*, foi possível criar um paralelo de comparação com os resultados fornecidos pelo próprio *Ceph*, considerando as diferenças entre os ambientes. Um aparato de testes funcionais para os componentes *OpenStack* foi montado de maneira secundária, a fim de checar a disponibilidade dos componentes durante os testes. Além disso, o ambiente foi monitorado com objetivo de checar possíveis problemas gerados pelos testes. Com isso, conseguimos resultados objetivos dos testes de desempenho e observabilidade para conclusões gerais sobre o ambiente. Os resultados, de maneira geral, foram satisfatórios e contribuem para a conclusão positiva sobre o uso da hiperconvergência da maneira que foi escolhida.

Nas seções seguintes, vamos contextualizar os principais conceitos necessários para o entendimento do experimento, apresentar a metodologia escolhida e suas fases, reunir e discutir resultados e, por fim, chegar a conclusão da análise do ambiente.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*Conference'17*, July 2017, Washington, DC, USA

© 2023 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 2 CONTEXTO

Nesta seção, abordamos os principais conceitos e tecnologias relacionadas ao experimento e análise proposta. Conceitos básicos serão abordados, como Arquitetura de *Data Centers*, Hiperconvergência, *OpenStack* e *Ceph*.

### 2.1 Arquitetura de *Data Center*

Arquitetura de infraestrutura convergente (CI) é uma abordagem baseada em *hardware* a fim de convergir armazenamento, serviços, processos e equipamentos de rede em um ponto central de gerenciamento. Enquanto a arquitetura hiperconvergente (HCI) é considerada uma evolução da anterior, sendo uma visão mais focada em *software*, que combina toda a pilha de serviços mesclada em um único servidor com uma camada de virtualização. Utilizando o contexto de uma *cloud*, HCI pode ser aplicado, por exemplo, a uma máquina que terá função de *host* (alocação de máquinas virtuais) e também servir como *host* para armazenamento de objetos. Com máquinas sendo utilizadas para múltiplas funções, encontramos então o principal desafio na gerência de uma *cloud* com HCI: gerência de recursos. Ainda assim, as vantagens que essa arquitetura traz são atrativas ao mercado de *cloud*, uma vez que facilita questões de compatibilidade com *hardware*, versão de *software* e facilidade de administração em apenas uma camada.

### 2.2 *OpenStack*

*OpenStack* é um gerenciador *open source* para *clouds* públicas e privadas, uma plataforma que oferece infraestrutura modular de *cloud*. Sua característica modular é alcançada através de suas ferramentas, chamadas pela comunidade de componentes ou projetos, que, em conjunto, lidam com os serviços essenciais de *cloud computing*: recursos de computação, rede, armazenamento, identidade e imagem. A capacidade de personalização, conforme a decisão quanto a pilha de componentes que será utilizada para a *cloud*, é um dos principais atrativos para o mercado. Na Figura 1, é possível visualizar os principais componentes *OpenStack* e suas funções [4].

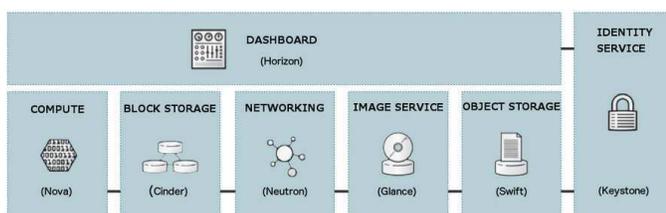


Figura 1: Principais componentes *OpenStack*. Fonte:

### 2.3 *Ceph*

*Ceph* é um sistema *open source* de armazenamento distribuído que fornece alta disponibilidade, escalabilidade e tolerância a falhas. Ele permite que os dados sejam armazenados em uma variedade de dispositivos de armazenamento, incluindo discos rígidos e unidades *flash*, além de ser compatível com diferentes protocolos, como *S3*, *Swift* e *NFS*. O *Ceph* utiliza uma arquitetura baseada em nós, onde cada nó é responsável por gerenciar um conjunto de objetos de armazenamento e seus metadados. Possui também uma camada de

gerenciamento de *cluster* que garante a integridade dos dados e a distribuição de carga. Ele é amplamente utilizado em ambientes de *cloud* e *data center*. *Ceph-OSD* (*Object Storage Daemon*) é um componente chave para a arquitetura do *Ceph*, responsável por gerenciar o armazenamento de objetos em discos locais ou dispositivos de armazenamento externos. Como demonstrado na Figura 2 [5], cada nó no *cluster* do *Ceph* possui um ou mais *OSDs* responsáveis por armazenar fragmentos de objetos e replicá-los para outros nós.

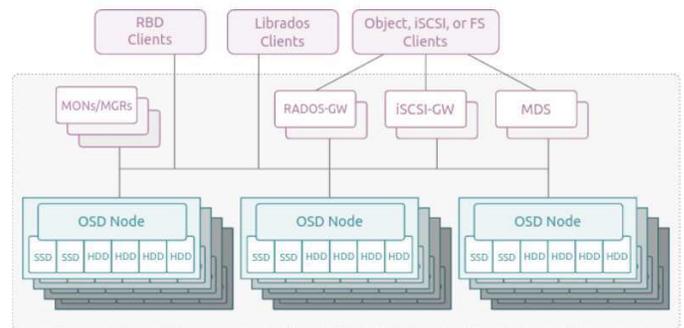


Figura 2: Arquitetura do *Ceph*. Fonte:

## 3 METODOLOGIA

Para guiar a análise dos dados recolhidos, as seguintes questões de pesquisa deverão ser respondidas:

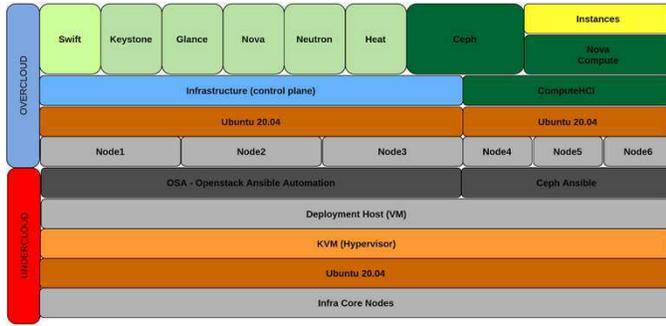
**Q1:** Existe possibilidade de indisponibilidade de componentes motivados pela hiperconvergência?

**Q2:** Dentro do contexto dos testes, existe algum indício que a arquitetura hiperconvergente diminui o desempenho da *cloud*?

O ambiente<sup>1</sup> analisado é composto por 21 *hosts* físicos, sendo 18 deles hiperconvergentes, com *nova-compute* e *ceph-osd* e 3 com funções separadas de infraestrutura. Nestas 18 máquinas HCI, temos os *clients* de todos os componentes *OpenStack* utilizados no ambiente, as máquinas virtuais sendo alocadas de maneira distribuída pelo *nova-compute* e volumes de objetos sendo armazenados pelo *Ceph*. Uma visão geral da arquitetura pode ser vista na Figura 3.

Para responder os questionamentos citados anteriormente, testes de desempenho com diferentes configurações de escrita/leitura foram desenvolvidos para a análise de desempenho, utilizando *FIO* (*Flexible I/O Tester*). O *FIO* é uma ferramenta de *benchmark* que permite a configuração de testes de leitura e escrita personalizados, com suporte para operações de bloqueio, operações randômicas e sequenciais, além de várias configurações de *threads* e processos. Ele também fornece métricas detalhadas de desempenho, como *IOPS*, latência e *bandwidth*, métricas importantes para a análise do ambiente. *Input/Output per Second*, ou *IOPS*, é uma métrica que se refere a quantidade de operações de entrada e saída de dados que um dispositivo de armazenamento consegue realizar por segundo. Na Tabela 1, é possível ver as configurações escolhidas para cada um dos 4 testes, considerando o *IO Depth* e *Block Size*.

<sup>1</sup>Esta pesquisa segue uma abordagem quantitativa descritiva de dados reais de uma empresa de *e-commerce* e busca validar a escolha de arquitetura para uma *cloud* privada interna. Os dados foram coletados por experimentos realizados internamente, que serão explicados posteriormente, e informações confidenciais e sensíveis serão respeitadas.

Figura 3: *Overview* da arquitetura do ambiente.

Test Description	IO Depth	Block Size
Random Read	1,8,16,32	4k,128k,256k,512k
Random Write	1,8,16,32	4k,128k,256k,512k
Mix RW	1,8,16,32	4k,128k,256k,512k

Tabela 1: Definição dos testes do *FIO*.

O comando utilizado para executar o arquivo com definição dos testes é o `fio -name=fio-test -directory=<dir> -ioengine=libaio -direct=1 -numjobs=1 -runtime=300 -ramptime=60 -size=100G bs=Block Size -rw=<format-test> -iodepth=Io Depth`. Para o ambiente de execução desses testes, foram criadas 168 instâncias com 2 *vCPU*, 4GB de memória *RAM*, 40GB de armazenamento local e 40GB para volumes *NVME* e *HDD*. O total de instâncias chegou a 336: metade com volumes *HDD* e outra metade com volumes *NVME*, ambas distribuídas pelo *Ceph* no mesmo ambiente de alocação das instâncias. Todos os testes do *FIO* foram executados nos volumes conectados as instâncias, assim, a carga de escrita e leitura foi automaticamente distribuída pelo *Ceph*. A quantidade de instâncias criadas e suas configurações de recurso, chamadas no *OpenStack* de *flavors*, foram determinadas utilizando dados de outro ambiente produtivo presente na empresa, a fim de simular uma carga mais realista. Ao terminar a execução de cada um dos testes, o *FIO* disponibiliza um *report* com informações de *IOPS*, *bandwidth*, latência e uso de *CPU* da instância, como demonstrado na Figura 4.

Todas as máquinas físicas envolvidas nos testes são monitoradas através das ferramentas *Prometheus* e *Grafana*, permitindo que, durante os testes, sejam analisados os picos de uso de recurso da máquina. Como base de comparação para um ambiente normal, foram utilizadas métricas do *Grafana* de outro ambiente semelhante de produção. Caso algo saísse do normal esperado, logs dos componentes individuais eram analisados para entender se o problema era motivado por alta carga de uso de recursos. Um dos principais objetivos desse monitoramento é reunir indícios comportamentais do ambiente enquanto os testes eram executados, assim, caso exista uso excessivo de *CPU* ou indisponibilidade de algum componente, por exemplo, os alertas ajudam a guiar a análise. Além dos testes de desempenho, como medida auxiliar, também foi executado a cada 30 minutos um teste de *benchmarking* completo utilizando o *Rally*, um componente *OpenStack* que provê análise de desempenho para os componentes. Com isso, além de métricas puras de desempenho

```
Client1: (groupid=0, jobs=1): err= 0: pid=16109: Sat Jun 24 12:07:54 2017
write: IOPS=88, BW=623KiB/s (638kB/s)(39.4MiB/50032msec)
slat (nsec): min=500, max=145500, avg=8318.00, stdev=4781.50
clat (usec): min=170, max=78367, avg=4019.02, stdev=8293.31
lat (usec): min=174, max=78375, avg=4027.34, stdev=8291.79
clat percentiles (usec):
| 1.00th=[ 302], 5.00th=[ 326], 10.00th=[ 343], 20.00th=[ 363],
| 30.00th=[ 392], 40.00th=[ 404], 50.00th=[ 416], 60.00th=[ 445],
| 70.00th=[ 816], 80.00th=[ 6718], 90.00th=[12911], 95.00th=[21627],
| 99.00th=[43779], 99.50th=[51643], 99.90th=[68682], 99.95th=[72877],
| 99.99th=[78119]
bw ( KiB/s): min= 532, max= 686, per=0.10%, avg=622.87, stdev=24.82, samples=
100
iops      : min= 76, max= 98, avg=88.98, stdev= 3.54, samples= 100
lat (usec) : 250=0.04%, 500=64.11%, 750=4.81%, 1000=2.79%
lat (msec) : 2=4.16%, 4=1.84%, 10=4.90%, 20=11.33%, 50=5.37%
lat (msec) : 100=0.65%
cpu       : usr=0.27%, sys=0.18%, ctx=12072, majf=0, minf=21
IO depths : 1=85.0%, 2=13.1%, 4=1.8%, 8=0.1%, 16=0.0%, 32=0.0%, >=64=0.0%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
issued rw: total=0,4450,0, short=0,0,0, dropped=0,0,0
latency   : target=0, window=0, percentile=100.00%, depth=8
```

Figura 4: Exemplo de *report* do *FIO*.

dos testes de desempenho, foco desse trabalho, ainda foi possível observar o comportamento do ambiente por outras óticas de maneira secundária.

## 4 RESULTADOS

Essa seção descreve os resultados obtidos a partir dos testes de desempenho implementados no ambiente hiperconvergente de *cloud* privada *OpenStack*. A análise geral do ambiente e como os testes impactaram o ambiente também serão discutidos nessa seção. As perguntas propostas na seção de metodologia serão respondidas também nessa seção.

A partir dos *reports* gerados pelo *FIO*, os dados foram reunidos e gráficos foram desenvolvidos para as métricas mais importantes. É importante frisar que esse teste de desempenho foi realizado pela primeira vez, logo, não foi possível comparar a dados de outro ambiente produtivo semelhante. Contudo, existem dados de testes da mesma natureza providos pelo *Ceph* [6], que serão utilizados aqui de maneira comparativa, mas considerando as diferenças dos ambientes, como o uso do *Cinder* como componente *OpenStack* responsável pelos volumes.

Na Figura 5, temos os dados de *IOPS* referentes ao teste *Random Write* com bloco de 4k aplicados em 42, 84 e 168 instâncias com volumes *NVME*, com *IO Depth* variando em 1, 8, 16 e 32. Segundo a documentação do *Ceph*, nesse cenário, a média é de cerca 463k *IOPS*. Como é possível observar, a quantidade de *IOPS* cresce junto ao *IO Depth*, atingindo o pico de 463,3k *IOPS* com 84 instâncias.

A Figura 6, mostra os dados de *IOPS* referentes ao teste *Random Read* com bloco de 4k aplicados em 42, 84 e 168 instâncias com volumes *NVME*, com *IO Depth* variando em 1, 8, 16 e 32. Segundo a documentação do *Ceph*, nesse cenário, para *IO Depth* de 16, o valor chega a aproximadamente 1.7M *IOPS*, enquanto o ambiente testado alcança 1.1M com 82 instâncias.

Por último, na Figura 7, temos os dados referentes ao teste de *Random Read-Write Mix (70/30)* com bloco de 4k aplicados em 42, 84 e 168 instâncias com volumes *NVME*, com *IO Depth* variando em 1, 8, 16 e 32. Segundo a documentação do *Ceph*, nesse cenário, para *IO Depth* de 16, o valor chega a aproximadamente 487,9k *IOPS*, enquanto o ambiente testado alcança 305,2K com 82 instâncias.

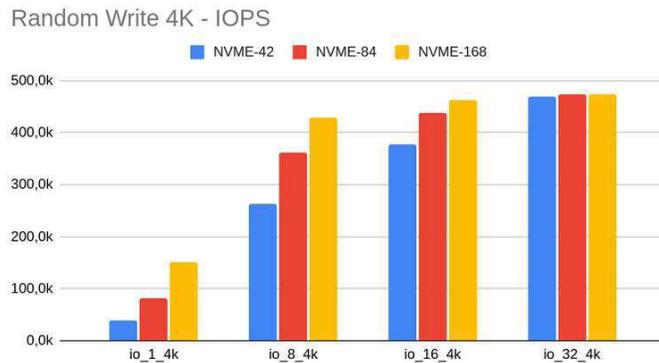


Figura 5: Dados de IOPS para Random Write 4k.

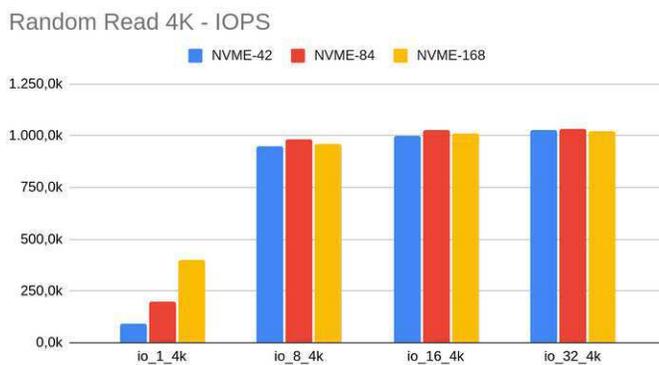


Figura 6: Dados de IOPS para Random Read 4k.

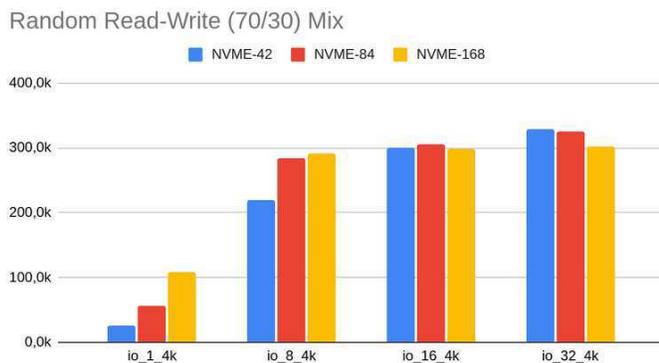


Figura 7: Dados de IOPS para Random Read-Write Mix (70/30) 4k.

Os dados citados anteriormente estão pouco abaixo da média fornecida pelo *Ceph*, como foi previsto, uma vez que o ambiente foco do teste difere do ambiente dos testes do *Ceph*. Assim, as métricas base são utilizadas como indicadores de desempenho. Além dos dados dos testes de desempenho como indicador de bom desempenho do *Ceph*, não foi gerado nenhum alerta relacionado à

indisponibilidade de componentes durante os testes, nem alerta de uso excessivo de *CPU* nas máquinas envolvidas no ambiente HCI.

Com a análise desses resultados, é possível responder aos questionamentos citados na metodologia.

**Q1:** Não pela hiperconvergência, mas sim pelo alto fluxo de requisições ao componente. No caso do objetivo dos testes, o componente mais requisitado é o *Cinder*. O único problema funcional que ocorreu no processo foi “*too many open files*”, relacionado ao *MySQL* que serve de base para o *GaleraDB*, cluster de banco de dados dos componentes *OpenStack*. Esse problema foi resolvido alterando uma variável de limite que estava configurada de maneira errada. Além disso, a ferramenta de *benchmarking* para testes funcionais nos componentes, *Rally*, não gerou nenhum erro de *SLA*.

**Q2:** Não. A hiperconvergência está sendo aplicada nas máquinas que servem como *host* para alocação de instâncias e *Ceph-OSD*. Os *servers* dos componentes *OpenStack* estão separados, sem hiperconvergência, em máquinas dedicadas aos serviços. Logo, não existe concorrência de recurso entre servidores e clientes, e sim entre clientes, o que não gera problemas não-convencionais motivados pela hiperconvergência.

## 5 CONCLUSÃO

Com os dados reunidos, é possível entender que o ambiente não está sendo afetado negativamente pela hiperconvergência, uma vez que os componentes não ficaram indisponíveis em momento algum por motivos de concorrência de recurso. Porém, é importante a atenção para a forma de *deploy* e a divisão de capacidade de cada máquina. No caso do uso de contêineres *LXC* para *deploy* dos componentes *OpenStack*, existe a possibilidade de ajustar os limites *defaults* de uso de recurso, caso exista indício que o *default* seja insuficiente. Além disso, ainda é necessário, para o caso de *nova-compute* e *ceph-osd* na mesma máquina, realizar esses testes de desempenho para entender como cada ambiente funcionará, uma vez que muitos pontos podem influenciar em possíveis erros: gargalos de redes, limites de *CPU*, configuração do *Cinder*, configuração de distribuição de discos do *Ceph*. Assim, é possível aproveitar as vantagens da hiperconvergência em um ambiente *OpenStack* com *Ceph* atendendo-se aos pontos principais alertados acima.

O ideal, seria ter realizado os testes de maneira isolada, com o ambiente sem uso, apenas com o *workload* de testes sendo executados, sem interferência de outros testes e *workloads*. Então, para trabalhos futuros relacionados, recomendo seguir, se possível, com o cenário citado anteriormente, para resultados mais limpos.

## REFERÊNCIAS

- [1] Serviços de data center devem crescer 16% em 2022, 2022. Available in: <https://cndl.org.br/varejosa/servicos-de-data-center-devem-crescer-16-em-2022/>, visited on: 10-15-2022.
- [2] Gartner forecasts worldwide public cloud end-user spending to reach nearly \$600 billion in 2023, 2022. Available in: <https://www.gartner.com/en/newsroom/press-releases/2022-10-31-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023>, visited on: 10-20-2022.
- [3] Andre Oliveira Danilo Oliveira Iure Fé Jean Araujo Rubens Matos Carlos Melo, Jamilson Dantas and Paulo Maciel. Availability models for hyper-converged cloud computing infrastructures. 2018 Annual IEEE International Systems Conference (SysCon), 2018.
- [4] Onur Özkan. Components of openstack. Available in: <https://kartaca.com/en/components-of-openstack/>, visited on: 01-05-2023.
- [5] What is ceph?. Available in: <https://ubuntu.com/ceph/what-is-ceph>, visited on: 01-07-2023.

[6] *Ceph Block Storage Performance on All-Flash Cluster with BlueStore backend*, 2019.  
Available in: <https://ceph.com/community/ceph-block-storage-performance-on->

[all-flash-cluster-with-bluestore-backend/](https://ceph.com/community/ceph-block-storage-performance-on-all-flash-cluster-with-bluestore-backend/), visited on: 09-12-2022.