

DOI: 10.5748/20CONTECSI/PSE/ITM/7232

eLocator: e207232

**ANÁLISE DE FERRAMENTAS DE CLUSTERIZAÇÃO E VIRTUALIZAÇÃO PARA
HARDWARE DE BAIXO CUSTO COM BANCO DE DADOS MARIADB**

Samuel Antonio Vieira – <https://orcid.org/0000-0002-1408-4183>

Faculdade De Tecnologia De Tatuí

ANALYSIS OF CLUSTERING AND VIRTUALIZATION TOOLS FOR LOW-COST HARDWARE WITH MARIADB DATABASE

ABSTRACT

This study investigates the interconnection of servers in clusters as a strategy to enhance the availability, reliability, and performance of systems, with an emphasis on the virtualization of machines and servers. Focusing on low-cost hardware, the work extends a previous research, highlighting the application of these practices in conjunction with the MariaDB database and its associated tools. The research aims to conduct a comparative analysis among three distinct clustering and virtualization tools. The evaluation encompasses various aspects, including performance, scalability, ease of configuration, management resources, support for MariaDB, and flexibility in resource allocation. It focuses on clustering, virtualization, and their specific applications in low-cost hardware, with special attention to the implementation of virtualization in clusters, including hosting virtual machines and integration with MariaDB. It involves the practical implementation of the three selected tools on low-cost hardware. Each tool is assessed in terms of performance, scalability, ease of configuration, management resources, support for MariaDB, and flexibility in resource allocation, with MariaDB used as the database service, adjusting to the specific needs of each cluster.

Keywords: Linux. Cluster. Free Software. MariaDB. Database.

ANÁLISE DE FERRAMENTAS DE CLUSTERIZAÇÃO E VIRTUALIZAÇÃO PARA HARDWARE DE BAIXO CUSTO COM BANCO DE DADOS MARIADB

RESUMO

Este trabalho estuda a interconexão de servidores em *clusters* como uma estratégia para aprimorar a disponibilidade, confiabilidade e desempenho de sistemas, com ênfase na virtualização de máquinas e servidores. Concentrando-se em *hardwares* de baixo custo, o trabalho continua uma pesquisa anterior, destacando a aplicação dessas práticas em conjunto com o banco de dados MariaDB e suas ferramentas associadas. O objetivo é conduzir uma análise comparativa entre três ferramentas distintas de clusterização e virtualização. A avaliação abrange diversos aspectos, incluindo desempenho, escalabilidade, facilidade de configuração, recursos de gerenciamento, suporte ao MariaDB e flexibilidade na alocação de recursos. Ele se concentra em clusterização, virtualização e suas aplicações específicas em *hardwares* de baixo custo, com especial atenção à implementação da virtualização em *clusters*, incluindo a hospedagem de máquinas virtuais e a integração com o MariaDB. Envolve a implementação prática das três ferramentas selecionadas em *hardwares* de baixo custo. Cada ferramenta é avaliada em termos de desempenho, escalabilidade, facilidade de configuração, recursos de gerenciamento, suporte ao MariaDB e flexibilidade na alocação de recursos, com o MariaDB utilizado como serviço de banco de dados, ajustando-se conforme as necessidades específicas de cada *cluster*.

Palavras-chave: Linux. Cluster. SoftwareLivre. MariaDB. Banco de Dados.

INTRODUÇÃO

Servidores em *clusters* referem-se à prática de interconectar vários servidores individuais para trabalharem juntos como uma única unidade. Essa abordagem tem o objetivo de melhorar a disponibilidade, confiabilidade e desempenho do sistema. Cada servidor no *cluster* é chamado de nó e é capaz de executar tarefas independentes ou trabalhar em conjunto para realizar tarefas complexas. A redundância e a distribuição de carga são características comuns em *clusters*, permitindo a continuidade do serviço mesmo se um nó falhar. (Pitanga, 2008)

Eles podem ser implementados para diversos fins, como balanceamento de carga, alta disponibilidade, processamento paralelo e tolerância a falhas. São frequentemente utilizados em ambientes empresariais para garantir a estabilidade e eficiência dos serviços oferecidos.

Neste contexto, Vieira (2023), afirma que a virtualização de máquinas e servidores desempenha um papel importante, proporcionando eficiência nos recursos e facilitando a administração. Em um *cluster*, a implementação da virtualização ocorre de diversas maneiras, pois cada nó do *cluster* pode hospedar uma ou mais máquinas virtuais (VMs). Essas VMs operam de forma isolada, permitindo a execução de diferentes sistemas operacionais e aplicativos em um único nó físico. A alocação dinâmica de recursos permite ajustes conforme necessário, adaptando-se à carga de trabalho em tempo real. Os hipervisores, responsáveis pela gestão e criação de VMs, podem ser executados em cada nó do *cluster*. Esses hipervisores coordenam entre si para equilibrar a carga de trabalho, migrando VMs entre nós para otimizar o desempenho e assegurando alta disponibilidade em caso de falhas.

Este trabalho é uma continuação da pesquisa que visa estudar a ação dos *clusters* e da virtualização em máquinas de baixo custo. O serviço a ser utilizado será o de banco de dados MariaDB e suas ferramentas auxiliares, de acordo com a necessidade e implementação de cada *cluster*.

O objetivo deste trabalho é realizar uma análise comparativa entre três ferramentas de clusterização e virtualização. O escopo da comparação engloba diversos aspectos, incluindo desempenho, escalabilidade, facilidade de configuração, recursos de gerenciamento, suporte a sistemas de banco de dados e flexibilidade na alocação de recursos.

É uma pesquisa sobre clusterização, virtualização, e suas aplicações específicas em *hardwares* de baixo custo. Será dada ênfase à implementação da virtualização em *clusters*, com foco especial na hospedagem de máquinas virtuais e na integração com o banco de dados MariaDB. Envolverá a implementação prática de três ferramentas de clusterização e virtualização em *hardwares* de baixo custo. Cada ferramenta será avaliada quanto ao desempenho, escalabilidade, facilidade de configuração, recursos de gerenciamento, suporte a MariaDB e flexibilidade na alocação de recursos. Será utilizado o MariaDB como serviço de banco de dados, ajustando a configuração conforme a necessidade de cada *cluster*.

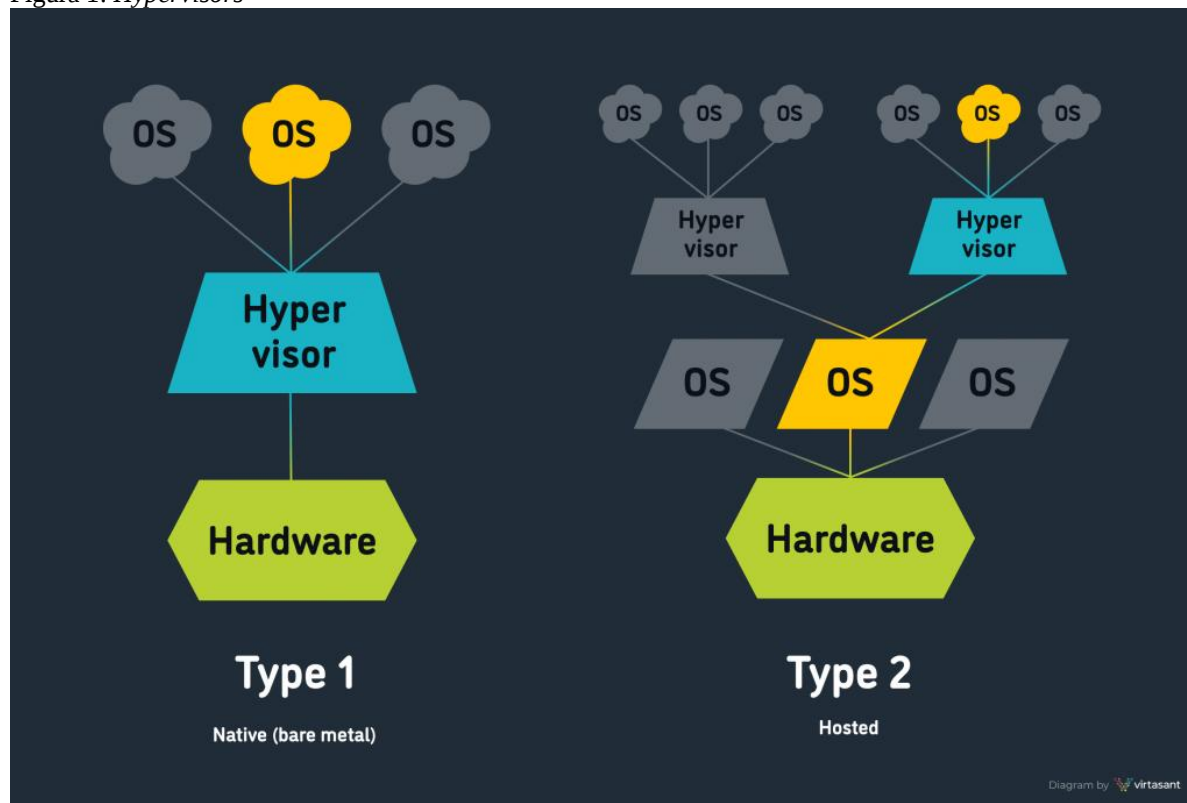
CLUSTER E VIRTUALIZADORES

A virtualização é uma tecnologia que permite a criação de ambientes virtuais ou máquinas virtuais (VMs) em um único *hardware* físico. Esse conceito permite que um servidor hospede múltiplos sistemas operacionais e aplicativos isolados uns dos outros. Cada máquina virtual age como se estivesse em seu próprio ambiente dedicado, mesmo que compartilhe os recursos físicos com outras VMs. Oferece benefícios, como a otimização do uso de recursos, maior flexibilidade na alocação de recursos, facilidade de migração de ambientes entre servidores e a capacidade de isolar aplicativos para melhorar a segurança. Hipervisores, também conhecidos como monitores de máquinas virtuais, são responsáveis

por gerenciar e coordenar as VMs, garantindo que compartilhem eficientemente os recursos do *hardware*.(VMWARE, 2006)

O *hypervisor* cria e gerencia as máquinas virtuais, alocando recursos como CPU, memória RAM e armazenamento para cada uma delas. Ele também fornece a camada de abstração necessária para que as VMs ajam como sistemas independentes, mesmo compartilhando recursos físicos. Além disso, o *hypervisor* controla a comunicação entre as máquinas virtuais e o *hardware*, garantindo um ambiente virtual estável e seguro.

Figura 1: *Hypervisors*



Fonte: Hajdarbegovic (2020)

Veras (2015) afirma que existem duas classes de *Hypervisors*: Tipo 1 (baremetal) e Tipo 2 (hosted) (Fig. 1). No Tipo 1, ele é instalado diretamente no *hardware* físico da máquina, operando sem a necessidade de um sistema operacional hospedeiro. Isso proporciona acesso direto ao *hardware*, resultando geralmente em melhor desempenho e eficiência devido à ausência da sobrecarga de um sistema operacional adicional. No Tipo 2, ele está instalado sobre um sistema operacional hospedeiro existente. Opera em um nível acima do sistema operacional hospedeiro, o que pode levar a alguma perda de desempenho devido à camada adicional. No entanto, é geralmente mais fácil de instalar e configurar, sendo comumente utilizado para testes e desenvolvimento.

Vieira (2020) fez um estudo comparando o uso de dois *hypervisors* do Tipo 1. Os escolhidos foram o *Xen Hypervisor* e *Proxmox Virtual Environment* (Proxmox VE), este focou no uso de *hypervisor* com licença livre e para uso em pequenas e médias empresas. Levantou questões de facilidade de instalação, configuração, uso e manutenção. Sua conclusão foi que ambas ferramentas atendem bem ao seu propósito, levando em consideração o nível de maturidade de seu administrador. O autor destaca que o *Xen Hypervisor* (Fig. 2) é mais robusto e tem a vantagem de trabalhar com a paravirtualização, técnica onde os sistemas operacionais convidados são modificados de forma a melhorar a performance e o compartilhamento de recursos da máquina hospedeira.

Figura 2: Xen Manager em modo texto

```

root@srv-fatec-ta:~# xm list
Name                               ID    Mem  VCPUs    State    Time(s)
Domain-0                           0    2779    16    r----- 883744.7
srv-2008                            7    4096     2    r----- 870405.3
srv-LTSPv2                          5    2048     4    -b----- 34245.2
srv-WEB-V2                          3    1500     2    -b----- 1252635.1
srv-lamp                            6     512     2    -b----- 15769.6
srv-mysql3                          1    1024     2    -b----- 17804.6
srv-samba-adm                       2    4096     8    -b----- 43032.8
root@srv-fatec-ta:~#

```

Fonte: Vieira (2020)

O *Xen Hypervisor* é uma solução de virtualização de código aberto, destaca-se por sua abordagem de paravirtualização. Este *hypervisor*, distribuído sob a GNU General Public License (GPL), oferece uma separação entre os sistemas operacionais convidados e o *hardware*. Ao requerer modificações nos sistemas operacionais convidados, a paravirtualização no Xen possibilita uma interação mais direta e eficaz, resultando em melhor desempenho. Com uma comunidade ativa e ampla adoção em diversos ambientes, o *Xen Hypervisor* continua a ser uma escolha interessante para ambientes de virtualização que buscam eficiência e flexibilidade. (XenProject.Org, 2022)

Figura 3: Interface Web Proxmox VE

Type	Description	Disk usage...	Memory us...	CPU usage	Uptime
qemu	100 (srv-samba)		78.1 %	0.5% of 1C...	76 days 03:11:42
qemu	102 (Win10-testesamba)		87.5 %	6.3% of 4C...	59 days 12:06:43
qemu	101 (Win10-teste)				-
qemu	103 (srv-samba)				-

Start Time	End Time	Node	User name	Description	Status
Nov 15 04:09:03	Nov 15 04:09:05	dell	root@pam	Update package database	Error: command 'apt-get upd...
Nov 14 04:34:54	Nov 14 04:34:56	dell	root@pam	Update package database	Error: command 'apt-get upd...
Nov 13 02:51:54	Nov 13 02:51:56	dell	root@pam	Update package database	Error: command 'apt-get upd...
Nov 12 03:02:45	Nov 12 03:02:48	dell	root@pam	Update package database	Error: command 'apt-get upd...
Nov 11 02:21:54	Nov 11 02:21:56	dell	root@pam	Update package database	Error: command 'apt-get upd...

Fonte: Vieira (2020)

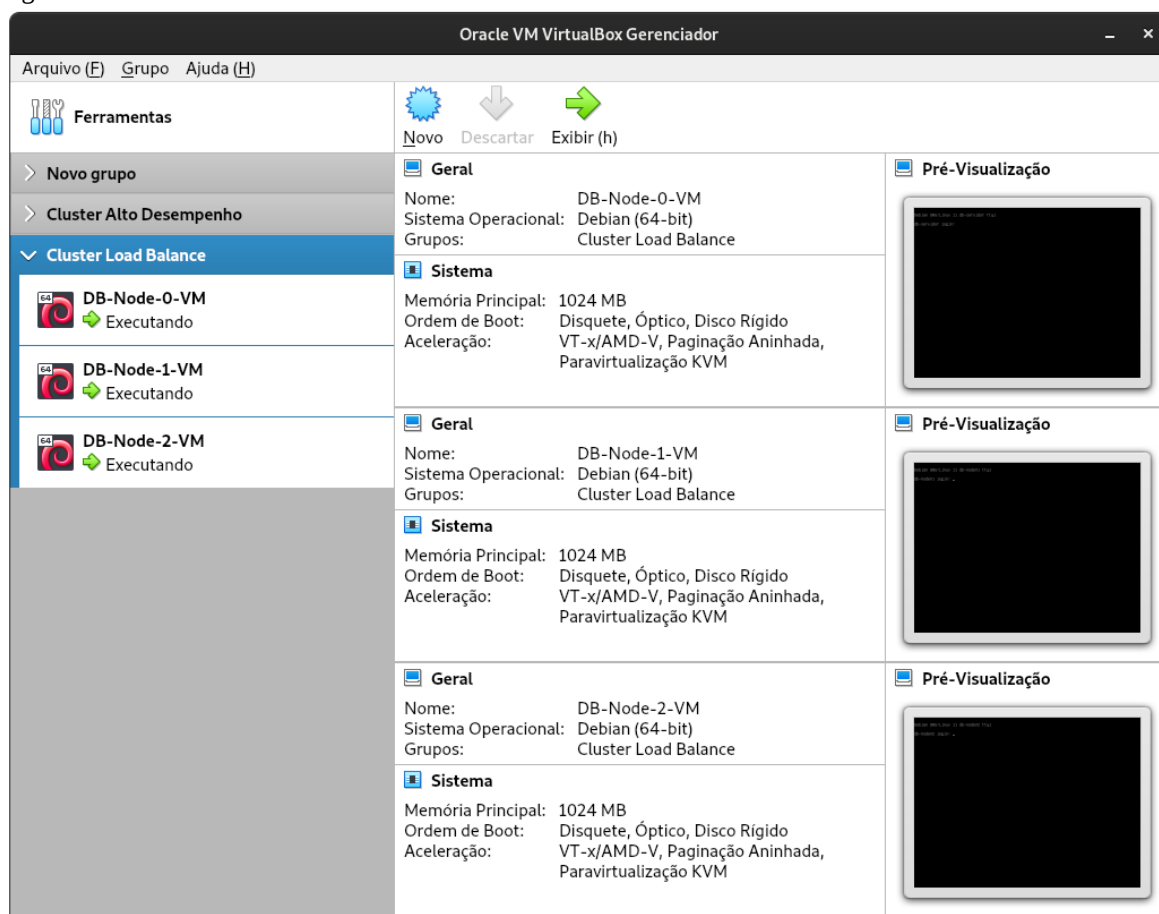
No que diz respeito à curva de aprendizado, facilidade de uso e manutenção, Vieira (2020) afirma que o Proxmox VE (Fig. 3) centraliza melhor os recursos para administração do usuário. Sua ferramenta *web* é uma grande aliada e bem intuitiva. Trabalhar com virtualização total abrange um número maior de convidados que podem ser instalados no sistema. Emerge como uma solução de virtualização aberta que integra virtualização de máquinas e containers, combinando versatilidade com facilidade de gerenciamento. Distribuído sob a licença GNU Affero General Public License (AGPL). Oferece um ambiente de virtualização de código aberto, seu foco na praticidade é visto na interface *web*

intuitiva, simplificando a administração de máquinas virtuais e contêineres. Além disso, ao adotar tecnologias como o KVM (Kernel-based Virtual Machine) e o LXC (Linux Containers), ele proporciona uma abordagem eficiente para ambientes heterogêneos. Com suporte à migração em tempo de execução, backup fácil e monitoramento integrado, o Proxmox VE destaca-se como uma escolha robusta para organizações que buscam uma plataforma de virtualização completa e de alto desempenho. (Proxmox, 2023)

O Oracle VM VirtualBox (Fig. 4), uma solução de virtualização de Tipo 2, se destaca como uma oferta versátil e de código aberto da Oracle Corporation. Distribuído sob a GNU General Public License (GPL), o VirtualBox oferece uma plataforma acessível e amplamente adotada para criação e gerenciamento de máquinas virtuais em diversos ambientes. Com suporte a uma ampla gama de sistemas operacionais como convidados, sua interface gráfica intuitiva e recursos robustos, como snapshots e suporte a dispositivos USB, contribuem para uma experiência de virtualização completa. (Virtualbox.Org, 2022)

Avançando em seus estudos sobre a virtualização, Vieira (2022) propôs um modelo de *cluster* virtualizado, desta vez, o foco era trazer um virtualizador de Tipo 2, de licença aberta, de fácil uso e de licença livre. Para isso, foi utilizado o Oracle VM Virtualbox. Seu estudo utilizou de computadores antigos, com configuração de *hardware* já obsoleto para montar um *cluster* que pudesse ser de fácil manutenção e escalável. Para fazer a parte de *cluster*, foi utilizado o Galera Cluster juntamente com o HAProxy.

Figura 4: Oracle VM VirtualBox



Fonte: Vieira (2022)

Clusters não é um assunto recente, tem suas raízes na década de 60, quando a Industry Business Machine (IBM) resolveu interligar seus mainframes de forma para melhorar a respostas às requisições de processamento que eles recebiam. (Teles, 2018) Sua

evolução, durante as décadas seguintes teve seu ápice no novo milênio, o uso massivo da *internet* criou a necessidade de um *hardware* escalável e com desempenho que atendessem a crescente demanda.

Perez (2021) destaca que um sistema computacional pode ser potencializado quando é dimensionado, com uma base bem definida para o tipo de *cluster* a ser utilizado. Essa abordagem pode trazer benefícios, como maximizar a segurança, otimizar o desempenho e mitigar os custos associados aos sistemas computacionais. *Clusters*, quando baseados em máquinas convencionais, têm a capacidade de oferecer alto poder de processamento sem depender de *hardwares* específicos. Essa flexibilidade é particularmente vantajosa para facilitar a escalabilidade e garantir heterogeneidade quando há a necessidade de aumentar a quantidade de nós para atender a novas demandas.

Além disso, ao longo das últimas décadas, os *clusters* evoluíram para além de simples agrupamentos de servidores, incorporando tecnologias como a virtualização. A virtualização em *clusters* proporciona uma camada adicional de flexibilidade, permitindo a consolidação de recursos e a execução de diferentes sistemas operacionais em ambientes isolados. Essa convergência de tecnologias destaca a importância contínua dos *clusters* em ambientes computacionais modernos, onde a eficiência, escalabilidade e gestão dinâmica de recursos são fundamentais para atender às complexas demandas tecnológicas.

A evolução dos *clusters* não se restringe apenas à esfera tecnológica, mas também impacta a segurança e a resiliência dos sistemas. A distribuição de carga inteligente e estratégias de recuperação rápida de falhas tornam os *clusters* essenciais para garantir a alta disponibilidade e confiabilidade de serviços críticos.

Com a contínua evolução tecnológica, os *clusters* foram adaptados para atender às crescentes exigências de aplicativos modernos e cargas de trabalho mais complexas. A virtualização, em particular, desempenhou um papel importante nesse processo. Ao integrar tecnologias de virtualização em *clusters*, as organizações podem criar ambientes altamente dinâmicos e eficientes. A virtualização de servidores em *clusters* permite a consolidação de recursos, facilitando a gestão e alocação flexível de máquinas virtuais em toda a infraestrutura. Isso não apenas aumenta a eficiência operacional, mas também contribui para a economia de recursos, resultando em custos reduzidos e maior sustentabilidade. A evolução dos *clusters* não se restringe apenas à esfera tecnológica, mas também impacta positivamente a segurança e a resiliência dos sistemas. A distribuição de carga inteligente e estratégias de recuperação rápida de falhas tornam os *clusters* essenciais para garantir a alta disponibilidade e confiabilidade de serviços críticos.

No contexto atual, onde a computação em nuvem desempenha um papel proeminente, os *clusters* continuam a ser uma base essencial para muitas infraestruturas em nuvem. A capacidade de dimensionamento elástico e a resposta dinâmica às demandas variáveis são características fundamentais que os *clusters*, aliados à virtualização, proporcionam aos ambientes de nuvem. À medida que os *clusters* continuam a evoluir, eles desempenham um papel central na conformidade com as crescentes demandas por desempenho, escalabilidade e flexibilidade no cenário tecnológico contemporâneo.

Pitanga (2008) destaca os tipos de *clusters*:

- *Cluster* de Alta Disponibilidade (HA) é projetado para garantir a continuidade operacional, um *cluster* de alta disponibilidade utiliza redundância para minimizar o tempo de inatividade em caso de falha em um nó. Os serviços são transferidos automaticamente para nós saudáveis.
- *Cluster* de Balanceamento de Carga: distribui o tráfego de rede entre vários servidores para otimizar o desempenho e evitar sobrecargas individuais. O objetivo é melhorar a eficiência e a capacidade de resposta do sistema.

- *Cluster* de Computação de Alta Performance (HPC): projetado para realizar computação intensiva, um *cluster* HPC conecta vários computadores ou nós para trabalhar em conjunto em tarefas complexas, como simulações científicas, modelagem climática e processamento de dados.
- *Cluster* de Armazenamento (*Storage Cluster*): focado na gestão eficiente de armazenamento de dados. Ele conecta vários dispositivos de armazenamento para fornecer maior capacidade e desempenho, além de garantir a redundância e a recuperação em caso de falha.
- *Cluster* de Virtualização: projetado para criar ambientes virtuais, um *cluster* de virtualização reúne vários servidores físicos para compartilhar recursos e executar máquinas virtuais. Isso permite uma alocação dinâmica de recursos conforme necessário.
- *Cluster* de Banco de Dados: configurado para fornecer alta disponibilidade e escalabilidade para sistemas de gerenciamento de banco de dados. Ele utiliza replicação e distribuição de carga para otimizar o desempenho e a confiabilidade.

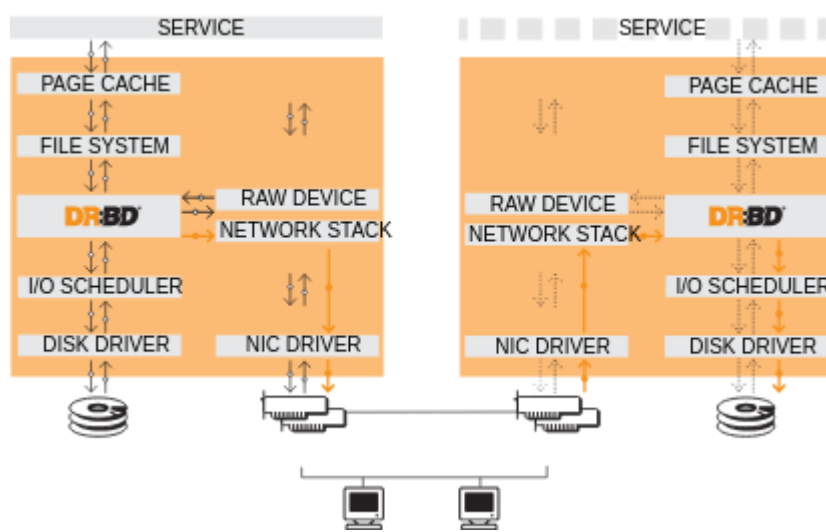
Esses são alguns dos tipos fundamentais, cada um atendendo a necessidades específicas em termos de confiabilidade, desempenho e flexibilidade em diferentes contextos de aplicação.

FERRAMENTAS PARA ALTA DISPONIBILIDADE EM BALANCEAMENTO DE CARGA

Sistemas Linux possuem uma vasta gama de ferramentas para os *clusters*, para trabalhar seus diversos tipos, dentre elas, pode-se destacar o DRBD, o HeartBeat, HAProxy e o Ceph.

O DRBD, ou Distributed Replicated Block Device (Fig. 5), é uma tecnologia de replicação de blocos para sistemas Linux. Ele é projetado para fornecer alta disponibilidade, redundância e recuperação de desastres em ambientes onde a integridade e disponibilidade dos dados são críticas. Opera no nível do bloco de armazenamento e permite a replicação síncrona ou assíncrona de dados entre servidores. Seu modo de trabalho é realizar a replicação de blocos de dados entre dois nós (servidores). Isso significa que, para cada bloco de dados gravado em um nó primário, um bloco correspondente é replicado para um nó secundário. Oferece modos de replicação síncrona e assíncrona. No modo síncrono, as gravações são confirmadas apenas após a replicação ser concluída em ambos os nós, garantindo uma consistência estrita. No modo assíncrono, a replicação ocorre em segundo plano, permitindo maior flexibilidade, mas com uma possível perda de dados em caso de falha. Com *failover* automático, em caso de falha do nó primário, o DRBD facilita a transição automática para o nó secundário, garantindo a continuidade operacional. (Linbit, 2023)

Figura 5: Esquema do DRBD

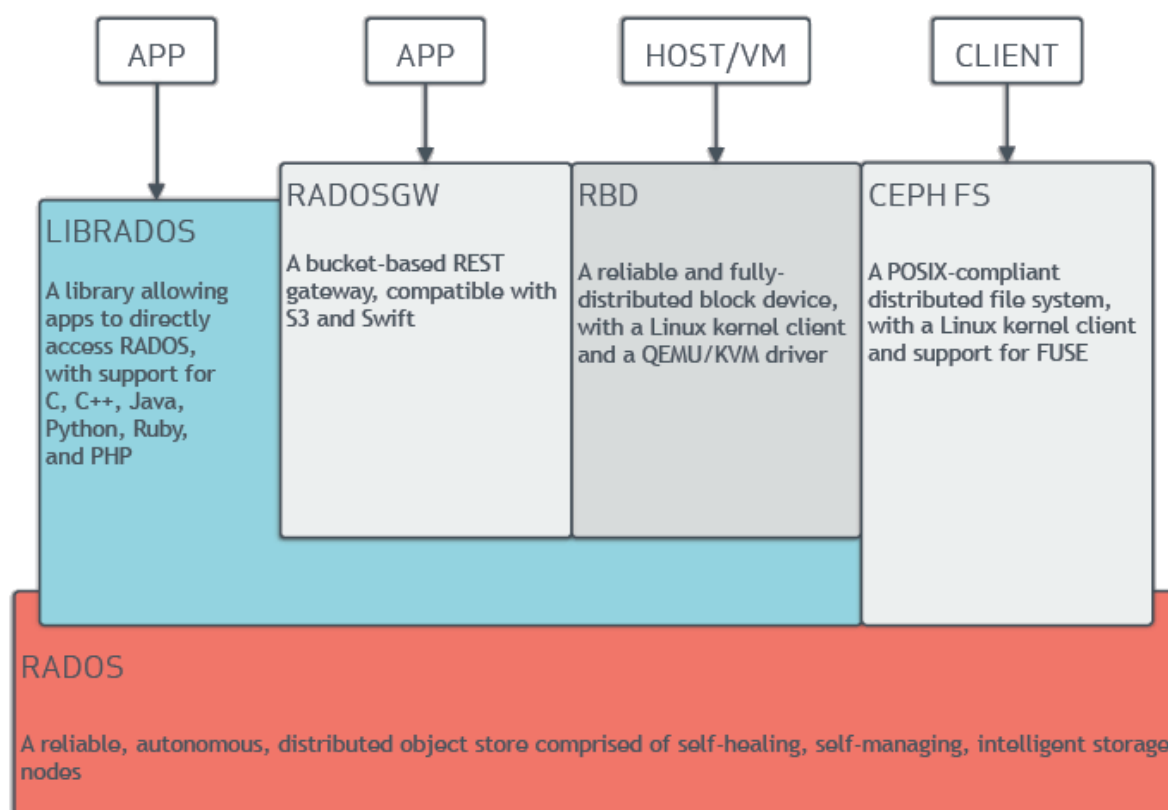


Fonte: LinBit (2023)

O *Heartbeat* é um *software* de código aberto usado para fornecer alta disponibilidade (HA) a sistemas Linux. Ele é projetado para monitorar a disponibilidade de serviços e, em caso de falha, realizar a transferência automatizada desses serviços para um nó de backup, garantindo a continuidade operacional. Utilizado em ambientes que exigem alta disponibilidade para garantir a continuidade operacional de serviços críticos. Sua simplicidade e eficácia o tornam uma escolha popular para implementações que buscam melhorar a confiabilidade de sistemas e serviços. (Li, Yu, Wu, 2009)

O Ceph é uma plataforma de armazenamento distribuído de código aberto projetada para oferecer escalabilidade, alta disponibilidade e desempenho. Desenvolvido pela Ceph Community, o Ceph é frequentemente utilizado em ambientes de nuvem e data centers para fornecer armazenamento distribuído e eficiente. Utiliza o RADOS (Fig. 6) como sistema de armazenamento de objetos ao Ceph. Ele distribui e replica dados em vários servidores chamados OSDs (*Object Storage Devices*), proporcionando escalabilidade e redundância. Tem seu próprio sistema de arquivos, o CephFS. Sistema de arquivos distribuído que permite o compartilhamento de arquivos entre os nós do *cluster*. Ele oferece um espaço de armazenamento comum acessível a todos os servidores, facilitando a colaboração e o compartilhamento de dados. (Ceph..., 2023)

Figura 6: Arquitetura Ceph



Fonte: Ceph Documentation (2023)

O RBD é um serviço que fornece armazenamento de blocos distribuídos. Ele permite criar dispositivos de bloco virtuais em uma infraestrutura distribuída, sendo útil para máquinas virtuais e armazenamento de dados. É escalável e permite o aumento da capacidade de armazenamento e do desempenho ao adicionar novos nós ao *cluster*. (Ceph..., 2023)

O HAProxy, ou *High Availability Proxy. Software* de balanceamento de carga e *proxy* TCP/HTTP de código aberto utilizado para distribuir o tráfego entre servidores para otimizar o desempenho e garantir alta disponibilidade. Sua principal função é equilibrar a carga de solicitações entre vários servidores para otimizar a utilização dos recursos e evitar sobrecargas em servidores individuais. Uma solução adotada em ambientes que exigem balanceamento de carga eficiente, alta disponibilidade e gerenciamento avançado de tráfego. Sua versatilidade o torna uma escolha popular para empresas que buscam otimizar a entrega de serviços online. (HAProxy, 2023)

BANCO DE DADOS MARIADB E SUAS FERRAMENTAS PARA CLUSTER

O MariaDB é um sistema de gerenciamento de banco de dados relacional (SGBDR) de código aberto, originado como um fork do MySQL. Criado com a preocupação sobre o futuro do MySQL após a aquisição pela Oracle, o MariaDB mantém forte compatibilidade com o MySQL, permitindo a execução de aplicações e comandos SQL sem modificações significativas.

Distribuído sob a Licença Pública Geral GNU (GPL), segue os princípios do código aberto, oferecendo liberdade aos usuários para usar, modificar e distribuir o *software*. Destaca-se por suas otimizações de desempenho em relação ao MySQL, incluindo

mecanismos de armazenamento como Aria e InnoDB. A comunidade ativa de desenvolvedores e usuários contribui para a evolução contínua. Além disso, opções de suporte profissional são oferecidas pela MariaDB Corporation e outros provedores. Amplamente adotado em diversos setores, destaca-se como uma boa escolha para empresas que buscam um sistema de gerenciamento de banco de dados robusto e compatível com o MySQL, mantendo os princípios do código aberto. (MariaDB, 2013)

Ele traz a ferramenta Galera *Cluster*, uma ferramenta de banco de dados distribuído projetada para proporcionar alta disponibilidade e escalabilidade ao MariaDB, MySQL e Percona Server. Trata-se de uma implementação de *cluster* de bancos de dados síncronos que utiliza replicação multimestre. Desenvolvido pela Codership, é conhecido por sua integração transparente e eficiência na replicação em ambientes distribuídos. Uma das principais vantagens, é sua capacidade de oferecer alta disponibilidade. Em casos de falhas de nós individuais, a ferramenta realiza automaticamente a promoção de outros nós para manter a continuidade operacional, assegurando que as operações de leitura e gravação sejam distribuídas entre os nós ativos. (MariaDB, 2013)

Além disso, suporta escalabilidade horizontal, permitindo a adição de novos nós conforme necessário para lidar com aumento de carga ou expansão do sistema. A detecção automática de falhas é uma característica importante, facilitando a rápida identificação e recuperação de falhas no *cluster*.

MATERIAIS E MÉTODOS

A metodologia adotada neste estudo é uma pesquisa exploratória com desenvolvimento experimental. A fase exploratória compreende uma revisão sobre clusterização, virtualização e suas aplicações específicas em *hardwares* de baixo custo. A ênfase é dada à implementação da virtualização em *clusters*, com especial atenção à hospedagem de máquinas virtuais e à integração com o banco de dados MariaDB. A fase experimental incluirá a implementação prática de três ferramentas de clusterização e virtualização em servidores de baixo custo. Cada ferramenta será avaliada em termos de desempenho, escalabilidade, facilidade de configuração, recursos de gerenciamento, suporte ao MariaDB e flexibilidade na alocação de recursos. Será utilizado o MariaDB como serviço de banco de dados, ajustando a configuração de acordo com as necessidades de cada *cluster*.

Figura 7: Hardware utilizado



Fonte: Autoria Própria

O *hardware* (Fig. 7) empregado foi composto por três sistemas de computadores que possuíam configurações idênticas, detalhadas a seguir:

- Processador Intel(R) Core(TM) i3 CPU 550 com clock de 3.20GHz;
- 12GB de memória RAM;
- Unidades de armazenamento Solid State Drive (SSD) de 250GB e Hard Disk (HD) de 500GB.

O experimento foi dividido em três etapas, cada uma com um clusterizador específico, em todos os casos foram feitos os seguintes passos:

- Instalação do sistema operacional no hospedeiro;
- Instalação do sistema específico para virtualização (quando necessário);
- Criação das máquinas virtuais;
- Instalação das máquinas convidadas;
- Instalação e configuração do serviço para o banco de dados;
 - Versão: mariadb-server-10.5 (1:10.11.4-1~deb12u1)
- Teste de carga de dados;
- Teste de carga de consultas SQL;
- Teste de *failover*.

Para a primeira etapa foi utilizado a estrutura proposta por Vieira(2022), adaptado para versões mais atualizadas de *software* e o com upgrade do *hardware*:

- Feito o *download* da ISO do Debian 12 - Bookworm, foi criado um pendrive de instalação, este utilizado para todas as máquinas;
- Durante a instalação, o sistema configurou a rede, para o acesso à *internet* e poder fazer o *download* de pacotes adicionais;
- Todo processo foi feito através da interface gráfica do Debian Installer a instalação foi feita em alguns minutos;
- Após o reinício, foram habilitados os pacotes non-free para o sistema poder trabalhar com pacotes de terceiros. (Não foi utilizado o Backports).
- Foi instalado Oracle VM VirtualBox (versão 7.0.12 r159484 (Qt5.15.3));
- Instalado o Oracle VM VirtualBox Extension Pack (7.0.6r155176).

Para a segunda etapa foi utilizado a estrutura proposta por Vieira(2023), adaptado para versões mais atualizadas de *software* e o com upgrade do *hardware*:

- Feito o *download* da ISO do Debian 12 - Bookworm, foi criado um pendrive de instalação, este utilizado para todas as máquinas;
- Durante a instalação, o sistema configurou a rede, para o acesso à *internet* e poder fazer o *download* de pacotes adicionais;
- Todo processo foi feito através da interface gráfica do Debian Installer a instalação foi feita em alguns minutos;
- Após o reinício, foram habilitados os pacotes non-free para o sistema poder trabalhar com pacotes de terceiros. (Não foi utilizado o Backports).
- A instalação do Xen 4.17 foi feita através do comando: *apt install xen-hypervisor-4.17-amd64* e o sistema foi reiniciado;
- Foram feitas configurações adicionais na rede para evitar que houvesse problemas na resolução de nomes durante a replicação de dados nos arquivos */etc/hosts*;
- A instalação DRBD foi feita através do comando: *apt install drbd-utils*;
- Feito as configurações no */etc/drbd.d/global_common.conf*;
- Verificando o sistema com o comando: *cat /proc/drbd*;

- A instalação do *heartbeat* foi feita através do comando: *apt install heartbeat*;
- A configuração foi feita nos arquivos: */etc/ha.d/haresources* e */etc/ha.d/authkeys*;
- O sistema foi testado com os comandos: *cl_status hbstatus* e *cl_status nodestatus srv_cluster01*.

Para a terceira etapa foi utilizado a estrutura proposta por Vieira(2022), adaptado para versões mais atualizadas de *software* e o com upgrade do *hardware*:

- Feito o *download* da ISO do Proxmox 8.0, foi criado um pendrive de instalação, este utilizado para todas as máquinas;
- Durante a instalação, o sistema configurou a rede, para o acesso à *internet* e poder fazer o *download* de pacotes adicionais. Todo sistema ficou na sub-rede 10.132.224.0/255.255.224.0;
- Todo processo foi feito através da interface gráfica do Proxmox a instalação foi feita em alguns minutos;
- A instalação do Ceph foi feita através do comando: *pveceph install -version luminous*. Versão do Ceph disponível no repositório do Proxmox;
- Foram configurados alguns serviços de monitor (*ceph-monitor*), gerência (*ceph-mgr*), OSD (*ceph-ods*) e MDS (*ceph-mds*).
- O sistema foi reiniciado e executado o comando *pveceph init --network 10.132.224.0/19* no nodo 1 do *cluster* para utilizar o *cluster*.

Cada etapa foi feita de maneira isolada, seguindo os passos descritos e observados pelo autor, de forma a garantir a qualidade dos resultados.

CONSIDERAÇÕES FINAIS

Este estudo proporcionou uma visão sobre a interconexão de servidores em *clusters*, destacando-se como estratégia para otimizar a disponibilidade, confiabilidade e desempenho de sistemas. Ao focar hardwares de baixo custo, a pesquisa expandiu investigações anteriores, sublinhando a relevância dessas práticas em conjunto com o banco de dados MariaDB e suas ferramentas correlatas.

O objetivo foi conduzir uma análise comparativa entre três ferramentas distintas de clusterização e virtualização, abordando aspectos como desempenho, escalabilidade, facilidade de configuração, recursos de gerenciamento, suporte ao MariaDB e flexibilidade na alocação de recursos. A pesquisa concentrou-se especificamente em aplicar essas técnicas em hardwares de baixo custo, com uma atenção especial à implementação da virtualização em *clusters*, desde a hospedagem de máquinas virtuais até a integração com o MariaDB.

Alguns pontos podem ser destacados dentre os resultados atingidos. A instalação dos sistemas operacionais no hospedeiros foram, em todos os casos, bem fáceis e intuitivos, mesmo para a ferramenta Proxmox VE, não foi necessário conhecimentos técnicos profundos para deixar o sistema funcionando.

O Proxmox VE também traz a facilidade de instalar o seu virtualizador de forma padrão, característica esperada para um sistema proposto para isso. Em relação aos demais, foi necessário instalação manual. Neste ponto, o mais trabalhoso foi *Xen Hypervisor* que necessitou de um conhecimento mais técnico para ficar pronto para instalação.

Para a criação das VMs e instalação de seus sistemas operacionais, foi bastante distinta a forma de trabalho de cada um. Destaque para o Oracle VirtualBox que traz simplicidade e clareza no processo, há uma experiência bem próxima a de uma máquina física. O Proxmox traz opções para instalação a partir de uma imagem ISO ou através de templates prontos, que podem ser baixados pela *internet*, isso facilita a vida do administrador, principalmente se precisar criar uma máquina teste. Novamente, exigindo um um pouco mais de conhecimento técnico, o Xen tem a vantagem de trabalhar com

paravirtualização, recurso que aproveita melhor os recursos do hospedeiro e pode fazer uma gestão mais eficiente, isso pode ser um diferencial em sistemas que exigem um desempenho mais equilibrado.

Seguindo essa linha, para criar os *clusters* em cada momento distinto, para o Oracle VM VirtualBox foi escolhido trabalhar com o Galera *Cluster* e HAProxy, desta forma todo *cluster* ficou nas máquinas convidadas e todo processo foi gerenciado pelo próprio banco de dados. O Xen foi o sistema que deu mais trabalho para configurar e deixar tudo funcional. Isso também se deve à opção de não usar o Citrix *Hypervisor*. Porém como todo sistema GNU Linux, quando foi colocado em funcionamento, ficou estável e robusto. O destaque ficou novamente com o Proxmox VE, através da interface *web* foi possível configurar o Ceph para clusterizar o armazenamento e habilitar o HA. Neste aspecto estudado, apenas o Oracle VM VirtualBox não apresentou uma ferramenta nativa para clusterização, necessitando ser feita diretamente no banco de dados das máquinas convidadas.

Sobre a escalabilidade, o Proxmox VE tem a opção para a criação de templates com base em suas máquinas, também é possível fazer a migração de uma hospedeira para outra de forma quente (online). A alta disponibilidade é bem feita e o Ceph garante que sempre haverá um sistema íntegro para substituir o defeituoso. O Xen também apresentou bons resultados, apesar de não ter a opção para criação de templates, a paravirtualização ajuda bastante o administrador a gerir isso, mesmo que manual. O *heartbeat* faz uma boa gestão sobre as máquinas interligadas e têm recursos para fazer a gestão de falhas. O DRBD também trabalha de forma eficiente em manter os nós sincronizados. Desta forma o Oracle VM VirtualBox apresentou o resultado menos satisfatório, tudo foi feito nos hospedeiros e o Galera *Cluster*, junto com o HAProxy, trabalharam bem, mas não houve uma gestão centralizada e ela ficou a critério de uma VM, isso pode acarretar problemas em sistemas mais críticos, além de consumir mais recursos.

Na carga de dados e testes de consultas SQL os sistemas tiveram desempenho parecidos, destaque ficou com o Galera *Cluster* com HAProxy, que trabalhou muito bem a gestão de requisições e sempre procurando a máquina que estivesse com maior disponibilidade.

Nos testes de *failover*, o Proxmox trabalhou bem a alta disponibilidade e quando necessário adicionar um novo hardware a sua configuração, não foi necessário fazer muitas configurações. O destaque negativo foi a demora para o sistema entender que havia algum problema e também a demora no ressincronismo. Isso pode ser explicado pelo uso de máquinas de baixo custo e com *hardware* obsoleto. O mesmo se aplica ao Xen *Hypervisor*, novamente, foi necessário um pouco mais de conhecimento técnico para fazer a substituição e ressincronismo das máquinas. Também houve os mesmo problemas de demora para o sistema entender o que estava acontecendo, o que também se refere aos mesmo requisitos de *hardware*. Em ambas etapas, o sistema acabou ficando fora do ar por algum tempo. O destaque deste requisito foi o Galera *Cluster* e HAProxy, apesar de também haver demora no ressincronismo, o que causou um pouco de atraso nas respostas do sistema, ele não ficou fora do ar por nenhum segundo.

A implementação prática das três ferramentas selecionadas em *hardwares* de baixo custo proporcionou uma compreensão de seu desempenho, escalabilidade e configuração prática. Utilizando o MariaDB como serviço de banco de dados, ajustamos sua configuração conforme as necessidades específicas de cada *cluster*, proporcionando uma avaliação abrangente e adaptável.

Os resultados obtidos ao longo deste estudo contribuem para a seleção de ferramentas de clusterização e virtualização em ambientes de baixo custo. Cabe ao gestor de T.I. avaliar os recursos necessários, o grau de criticidade do sistema e complexidade para escolher qual melhor sistema atenderá melhor às necessidades da organização, levando em conta custos e recursos disponíveis.

About VirtualBox. **VirtualBox**, 2022. Disponível em: <<https://www.virtualbox.org/>>. Acesso em: 19 de set. 2022.

CEPH Documentation. 2023. Disponível em: <https://docs.ceph.com/en/latest/start/intro/>. Acesso em: 10 nov. 2023.

Getting Started with MariaDB Galera Cluster. MariaDB, 2013. Disponível em: <https://mariadb.com/kb/en/getting-started-with-mariadb-galera-cluster/> . Acesso em: 03 de abr. 2023.

HAJDARBEGOVIC, Mensur. **Hypervisors: a comprehensive guide**. A Comprehensive Guide. 2020. Disponível em: <https://www.virtasant.com/blog/hypervisors-a-comprehensive-guide>. Acesso em: 30 out. 2023.

HAPROXY Documentation. 2023. Disponível em: <https://docs.haproxy.org/2.8/intro.html>. Acesso em: 19 nov. 2023.

LI, Fei-Fei; YU, Xiang-Zhan; WU, Gang. Design and Implementation of High Availability Distributed System Based on Multi-level Heartbeat Protocol. In: 2009 IITA INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION AND SYSTEMS ENGINEERING (CASE 2009), 2., 2009, Zhangjiajie, China. **2009 IITA International Conference on Control, Automation and Systems Engineering (case 2009)**. Zhangjiajie, China: Ieee, 2009. p. 83-87.

LINBIT. **The DRBD9 User's Guide**. 2023. Disponível em: https://linbit.com/drbd-user-guide/drbd-guide-9_0-en/. Acesso em: 21 nov. 2023.

PEREZ, André. **Sobre a História da Computação Distribuída e Clusters Kubernetes**. 2021. Disponível em: <https://medium.com/team-data-stone/sobre-a-hist%C3%B3ria-da-computa%C3%A7%C3%A3o-distribu%C3%ADa-e-clusters-kubernetes-3d0fe331db7>. Acesso em: 10 out. 2023.

PITANGA, Marcos. **Construindo Supercomputadores com Linux**. 3. ed. Rio de Janeiro: Brasport, 2008.

PROXMOX Virtual Environment. 2023. Disponível em: <https://www.proxmox.com/en/>. Acesso em: 10 out. 2023.

SILVA, Rodrigo Ferreira da. **Virtualização de Sistemas Operacionais**. 2007. 114 f. TCC (Graduação) - Curso de Tecnologia da Informação e Comunicação, Laboratório Nacional de Computação Científica, Petrópolis, 2007.

TELES, F. O que é Cluster e como essa estrutura pode ser benéfica para você. **Desk Manager**, 2018. Disponível em: <https://deskmanager.com.br/blog/cluster/>. Acesso em: 25 de abr. de 2023.

VMWARE Infrastructure Architecture Overview. 2006. Disponível em: https://www.vmware.com/pdf/vi_architecture_wp.pdf. Acesso em: 01 nov. 2023.

VERAS, Manoel. **Virtualização: tecnologia central do datacenter**. 2. ed. Rio de Janeiro: Brasport, 2015.

VIEIRA, S. A. **PROXMOX**: Uma alternativa ao xenserver. In: Mostra De Trabalhos Docentes Em RJ, 2020, São Paulo. **Anais [...]**. São Paulo: Centro Paula Souza, 2020. p. 133-134.

_____. **Utilização de Software Livre Para Clusterização e Virtualização De Servidores** In: Mostra De Trabalhos Docentes Em RJ, 2022, São Paulo. **Anais [...]**. São Paulo: Centro Paula Souza, 2022. p. 396-403.

_____. **Utilização do Software Livre Xen E Proxmox Para Clusterização e Virtualização**. In: Mostra De Trabalhos Docentes Em RJ, 2023, São Paulo. **Anais [...]**. São Paulo: Centro Paula Souza (no prelo)

_____. **Technologies For Server Virtualization: Proxmox And Xen Hypervisor** In: 18th CONTECSI - International Conference On Information Systems And Technology Management Virtual, 2021, São Paulo. **Anais [...]**. São Paulo: USP, 2021

_____. **Cluster de alta disponibilidade com balanceamento de carga em máquinas virtuais**: gerenciando banco de dados MariaDB com galera cluster. In: 2022: Xix Congresso Latino-Americano De Software Livre E Tecnologias Abertas, 2022. **Anais [...]**. Foz do Iguaçu, 2022. p. 86-92.

XENPROJECT.ORG: Documentation. 2022. Disponível em: <https://xenproject.org/help/documentation/>. Acesso em: 20 maio 2023.