



**Faculdade de Tecnologia
Departamento de Engenharia Elétrica**

**Análise de métricas de performance em
computação distribuída: O GNS3 como objeto
de estudo**

Nelson Roberto de Sá Miranda

**TRABALHO DE GRADUAÇÃO
ENGENHARIA DE REDES DE COMUNICAÇÃO**

Brasília
2023

Faculdade de Tecnologia
Departamento de Engenharia Elétrica

**Análise de métricas de performance em
computação distribuída: O GNS3 como objeto
de estudo**

Nelson Roberto de Sá Miranda

Trabalho de Graduação submetido como requi-
sito parcial para obtenção do grau de Bacharel
em Engenharia de Redes de Comunicação

Orientador: Prof. Dr. Georges Daniel Amvame Nze

Brasília
2023

d672a de Sá Miranda, Nelson Roberto.
Análise de métricas de performance em computação distribuída: O GNS3 como objeto de estudo / Nelson Roberto de Sá Miranda; orientador Georges Daniel Amvame Nze. -- Brasília, 2023.
36 p.

Trabalho de Graduação (Engenharia de Redes de Comunicação) -- , 2023.

1. Computação Distribuída. 2. Computação em nuvem. 3. GNS3. 4. Simulação de Redes. I. Amvame Nze, Georges Daniel, orient. II. Título

Faculdade de Tecnologia
Departamento de Engenharia Elétrica

**Análise de métricas de performance em computação
distribuída: O GNS3 como objeto de estudo**

Nelson Roberto de Sá Miranda

Trabalho de Graduação submetido como requi-
sito parcial para obtenção do grau de Bacharel
em Engenharia de Redes de Comunicação

Trabalho aprovado. Brasília, 22 de dezembro de 2023:

Prof. Dr. Georges Daniel Amvamve Nze,
UnB/FT/ENE
Orientador

Prof. Dr. Fábio Lúcio Lopes de Mendonça,
UnB/FT/ENE
Examinador Interno

Esp. Welber Santos de Oliveira, Estácio,
Brasília
Examinador Externo

Brasília
2023

Este trabalho é dedicado a todos que olham para o céu e enxergam mais longe que as estrelas.

Agradecimentos

Aos meus professores, que por suas correções e conhecimentos compartilhados se tornaram parte fundamental em minha formação acadêmica e profissional; ao Seu Nelson e à Dona Márcia, meus pais, por me apoiar na escolha da carreira promissora que estou trilhando e tranquilizar a feitura deste trabalho; ao amor da minha vida, Mariana Araújo, por estar acompanhando todas as etapas deste trabalho com muito carinho, cuidado e dedicação; e a todos os meus colegas que acompanharam o nascimento deste trabalho emanando suas vibrações positivas.

*“Do what you think is interesting,
do something that you think is worthwhile,
because otherwise you won’t do it well anyways.”
(Brian Kernighan)*

Resumo

A computação distribuída é um paradigma em evolução constante desde os anos 60. Por meio desta, é possível executar tarefas dispendiosas em vários computadores, promovendo ao usuário uma visão única de sistema. Os simuladores de rede são uma ferramenta muito útil para o estudo e validação de protocolos e novas tecnologias de telecomunicações, porém, exigem um alto quantitativo de recursos computacionais. O presente trabalho visa utilizar um dos mais adotados simuladores de rede, o GNS3, implementado de forma distribuída, para aferir métricas de uso de recursos computacionais. Até o presente momento, a pesquisa revela que a implementação distribuída do simulador em nuvem privada é vantajosa, uma vez que o usuário é menos impactado pela escassez de recursos. O uso em nuvem pública ainda não pôde ser validado em virtude do modelo de negócio do uso desta tecnologia, mas é proposto como uma perspectiva futura para a implementação da simulação de redes em ambiente acadêmico.

Palavras-chave: Computação Distribuída. Computação em nuvem. GNS3. Simulação de Redes.

Lista de ilustrações

Figura 1 – Linha do tempo evolutiva dos paradigmas de computação	14
Figura 2 – Diferenciação entre tipos de hipervisores - Fonte: Park Place Technologies: What Is a Hypervisor? – Types, Benefits & How Does It Work? . 2023 . .	16
Figura 3 – Esquemático do funcionamento do KVM. Fonte: Kernel-based Virtual Machine Technology (2011), GOTO, Yasunori	17
Figura 4 – Excerto dos appliances disponíveis no GNS3 Marketplace GNS3 Marketplace . 2023	19
Figura 5 – Arquitetura do GNS3 - Fonte: GNS3 Developer Docs v2.2.38	20
Figura 6 – Rede criada pelo virtualizador VMWare entre as máquinas virtuais da bancada de testes. Fonte: Elaboração própria.	22
Figura 7 – Configuração necessária para implementar uma rede virtual entre as máquinas virtuais.	22
Figura 8 – Script de instalação do GNS3 em servidor Linux remoto.	23
Figura 9 – Configurações relacionadas a GNS3 VM. Fonte: elaboração própria . . .	24
Figura 10 – Configurações relacionadas ao servidor de simulação. Fonte: elaboração própria	25
Figura 11 – Configurações relacionadas ao servidor remoto de simulação. Fonte: elaboração própria	26
Figura 12 – Coletando o IP da interface compartilhada do servidor remoto. Fonte: elaboração própria	26
Figura 13 – Topologia de referência para os cenários de análise	28
Figura 14 – Consumo de recursos - vCPU e RAM para a topologia de referência . . .	30
Figura 15 – Consumo de recursos - vCPU e RAM para a topologia de referência - cenário de uso intenso	30
Figura 16 – Teste da presença de processadores com acesso à instrução vmx.	31
Figura 17 – Gráficos de consumo de CPU e RAM para a topologia distribuída. . . .	32
Figura 18 – Gráficos de consumo de CPU e RAM para a topologia distribuída - cenário de uso extremo	33
Figura 19 – Gráficos de largura de banda para a topologia distribuída	33
Figura 20 – Gráficos de consumo de CPU e RAM para a topologia distribuída. . . .	34
Figura 21 – Gráficos de largura de banda para a topologia distribuída.	34

Lista de tabelas

Tabela 1	– Requisitos para uso do GNS3; fonte: Docs GNS3 - Recommended Requirements	13
Tabela 2	– Descrição dos cenários a serem testados experimentalmente para avaliar a performance de um sistema de simulação de redes distribuído.	21
Tabela 3	– Descrição da configuração das máquinas virtuais do ambiente de laboratório. Fonte: Elaboração própria	21
Tabela 4	– Elementos que compõem a topologia de referência, com suas especificações de consumo e tipo de virtualização.	29

Lista de abreviaturas e siglas

BMaaS	<i>Bare Metal as a Service</i>
CPU	<i>Central Processing Unit</i>
CSP	<i>Cloud Service Provider</i>
GB	<i>Gigabyte</i>
GNS3	<i>Graphical Network Simulator 3</i>
HTTP	<i>HyperText Transfer Protocol</i>
IaaS	<i>Infrastructure as a service</i>
JSON	<i>Javascript Object Notation</i>
KVM	<i>Kernel Virtual Machine</i>
PaaS	<i>Plataform as a service</i>
QEMU	<i>Quick Emulator</i>
QT	<i>QuTie</i>
RAM	<i>Random Access Memory</i>
SaaS	<i>Software as a service</i>
VM	<i>Virtual Machine</i>
VPN	<i>Virtual Private Network</i>

Sumário

1	INTRODUÇÃO	12
1.1	Contextualização	12
1.2	Descrição do Problema	12
1.3	Objetivos	13
1.3.1	Objetivo Geral	13
1.3.2	Objetivos Específicos	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Computação Distribuída	14
2.1.1	Computação em nuvem	15
2.2	Virtualização	16
2.2.1	KVM	17
2.3	GNS3	18
2.3.1	Graphical Network Simulator 3	18
2.3.2	A emulação de appliances	18
2.3.3	Arquitetura da ferramenta	19
2.3.4	O uso da ferramenta	20
3	METODOLOGIA	21
3.1	Configuração da base de testes	21
3.2	Instalação e configuração do GNS3	23
3.2.1	Instalação da GNS3 VM	23
3.2.2	Instalação no servidor remoto	23
3.2.3	Configuração do client	24
4	ANÁLISES E RESULTADOS	28
4.1	Topologia de referência - componentes e uso base de recursos	28
4.2	Cenário 1 - Uso local do GNS3	29
4.3	Cenário 2 - Uso do GNS3 em ambiente de nuvem computacional	31
4.4	Cenário 3 - Uso do GNS3 com um servidor remoto	32
4.4.1	Configurando os ativos de rede no servidor remoto	32
4.4.2	Configurando os appliances no servidor remoto	33
5	CONCLUSÕES	35
6	REFERÊNCIAS BIBLIOGRÁFICAS	36

1 Introdução

1.1 Contextualização

Desde a origem da computação moderna em meados de 1945, por meio do trabalho de Alan Turing, até a década de 80, os computadores eram demasiadamente caros e extensos[1]. Além disso, não havia forma conhecida publicamente até o momento de interconectar tais máquinas, até o advento das redes locais e da Internet.

Com a disseminação das redes locais, foi possível verificar uma crescente tendência tecnológica: o uso de vários computadores geograficamente espalhados, trabalhando num objetivo comum. Percebia-se, então, que além da possibilidade de comunicação máquina-máquina e do *downsizing*, a computação distribuída seria um dos grandes impactos na forma de se realizar o processamento de tarefas dispendiosas.

Nesse contexto, a capacidade de processamento individual de cada nó de um sistema distribuído passa a ter menos importância para o usuário final, a despeito do conjunto total de recursos que lhe é oferecido. Nessa tônica, o presente trabalho visa utilizar uma aplicação comum a engenheiros e estudantes como base de testes, a fim de avaliar a melhoria na experiência do usuário ao se utilizar a computação distribuída.

1.2 Descrição do Problema

A simulação de redes é uma valiosa ferramenta para estudantes, pesquisadores e profissionais na área de Tecnologia da Informação e Redes de Computadores. Sua grande utilidade na demonstração de conceitos, teste de protocolos em várias situações, promovem a robustez e eficiência dos avanços da área[2].

Para isto, existem no mercado uma série de simuladores e emuladores de rede, cada um com suas características e adequações para os diversos cenários em que podem atuar. Neste trabalho, será analisado o Graphical Network Simulator 3 (GNS3), um simulador baseado em virtualização de *appliances*, para ilustrar o uso da computação distribuída como solucionadora de problemas de falta de recursos computacionais.

O GNS3, como simulador, exige um quantitativo elevado de recursos computacionais, frequentemente exaurindo recursos de um computador pessoal comum. A Tabela 1, coletada da documentação oficial do simulador, indica os requisitos mínimos de uso da ferramenta[3].

Tabela 1 – Requisitos para uso do GNS3; fonte: [Docs GNS3 - Recommended Requirements](#)

	Sistema Operacional	CPU	RAM	Armazenamento
Mínimos	Windows 7 ou superior	2 ou mais núcleos lógicos	4 GB	1 GB
Recomendados	Windows 7 ou superior	4 ou mais núcleos habilitados para virtualização	16 GB	35 GB
Otimizados	Windows 7 ou superior	8 ou mais núcleos habilitados para virtualização	32 GB	80 GB

Uma vez que é comum, durante o uso da ferramenta de simulação, que sejam esgotados os recursos computacionais locais, será feita a avaliação do uso da computação distribuída para mitigar este contratempo comum aos usuários.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral deste trabalho é demonstrar como a computação distribuída pode auxiliar a execução de tarefas onerosas, computacionalmente falando, por meio de tecnologias como *pooling* de recursos, visão única de sistema e balanceamento de carga.

1.3.2 Objetivos Específicos

Os objetivos específicos deste trabalho emergem da plataforma escolhida para a análise das métricas de uso da computação distribuída. São estes:

- Demonstrar como a implementação distribuída do GNS3 reduz o impacto e a necessidade de recursos no computador pessoal do usuário final;
- Demonstrar de que maneira deve ser realizada a implementação da infraestrutura a fim de permitir o uso do simulador da melhor maneira possível;
- Debater a possibilidade futura de implementação do GNS3 em nuvem computacional.

2 Fundamentação Teórica

2.1 Computação Distribuída

A computação distribuída, ou um sistema distribuído, é definido por Coulouris como um sistema em que elementos de hardware ou software localizados em uma rede de computadores se comunicam e coordenam suas ações por meio de troca de mensagens[5]. O objetivo central da implementação dos sistemas distribuídos repousa no fato de que a informação e a computação são cada vez mais demandistas no que se refere ao uso de recursos computacionais, como memória RAM, unidades de processamento e armazenamento.

A história da computação distribuída e das redes de computadores estão intimamente relacionadas. A possibilidade de comunicação entre elementos de computação é a força motriz que impulsiona o desenvolvimento de aplicações que podem ser computadas colaborativamente. Ao observar a linha do tempo evolutiva destas duas tecnologias, é possível verificar a mudança do paradigma da computação centralizada para o paradigma da computação distribuída.

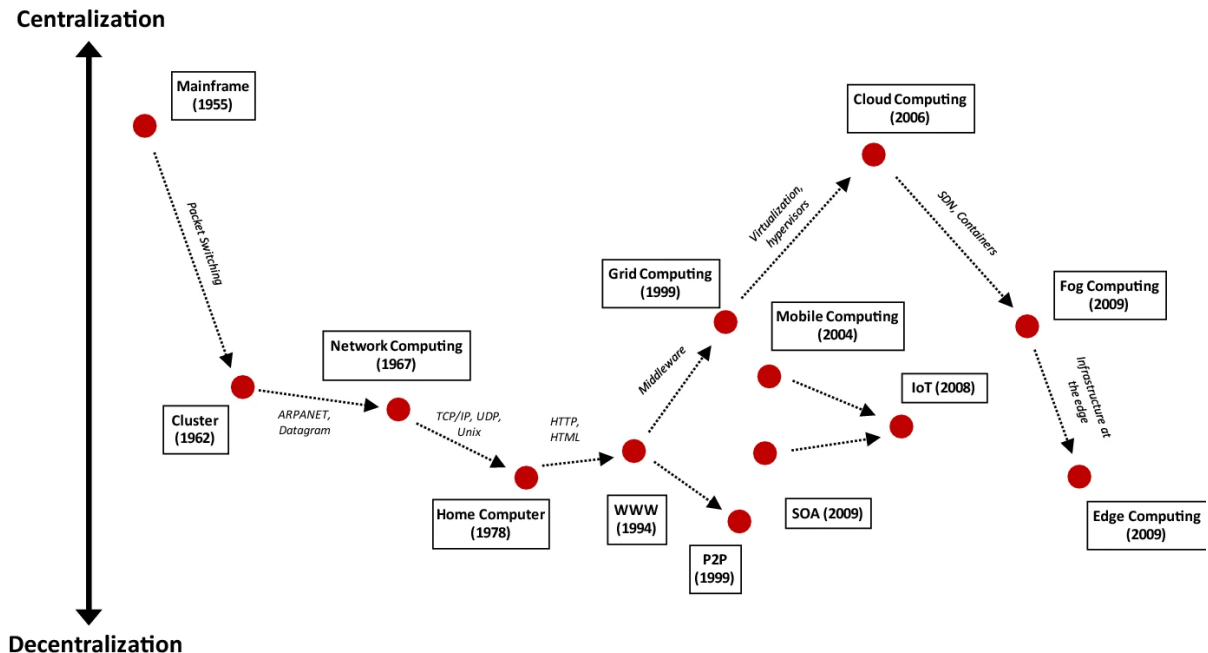


Figura 1 – Linha do tempo evolutiva dos paradigmas de computação. Fonte: *The evolution of distributed computing systems: from fundamental to new frontiers* (2021)

Como ilustra a Figura 1, a computação centralizada nos *mainframes* entre as décadas de 50 e 60 foi gradativamente migrando para sistemas distribuídos por meio da evolução das redes de computadores, iniciando com a rede militar ARPANET, passando pelo embrião da Internet, até culminarmos hoje com a computação em nuvem, em névoa e na borda.

Similarmente, os computadores evoluíram a ponto de agregar em si mais recursos computacionais, a despeito da tendência de *downsizing*. Computadores centrais que ocupavam salas inteiras a algumas décadas eram menos poderosos que dispositivos móveis atualmente.

Neste trabalho, será abordada a influência da computação distribuída em computadores virtualizados, uma vez que esta associação tecnológica é a base do funcionamento do simulador de redes contido na presente pesquisa.

2.1.1 Computação em nuvem

A computação em nuvem é um modelo de computação baseado em serviços, em que altera a maneira tradicional de uso de infraestrutura de tecnologia da informação. Consumidores que anteriormente investiram em infraestrutura física, gerenciando todos os equipamentos como servidores, armazenamento, rede, energia e afins; passam a investir no serviço de tecnologia por meio da Internet. Assim, o custo da implementação e manutenção de serviços passa a ser responsabilidade de um provedor externo, geralmente conhecido como *Cloud Service Provider*, ou CSP.

É bastante comum que os provedores de nuvem ofereçam seus serviços de tecnologia no modelo de assinatura, onde os serviços consumidos são pagos conforme o uso, gerando assim uma mudança também no modelo de negócios dos consumidores. A infraestrutura física que antes gerava *despesas de capital*, passa a gerar *despesas operacionais*.

Algumas características específicas da computação em nuvem são as seguintes:

- Serviço autônomo sob demanda;
- Agrupamento de recursos;
- Amplo acesso via *Internet*;
- Elasticidade;
- Serviços quantificados.

Podemos verificar a referência do [NIST, SP 800-145, 2011](#)[10] para mais detalhes quanto às características citadas. A referência [NIST, SP 800-146, 2011](#)[11] traz recomendações relacionadas à boas práticas de implementação e segurança dos sistemas em nuvem.

É notável a ascensão da computação em nuvem no contexto da computação distribuída, uma vez que esta incorpora seus principais objetivos. Tarefas onerosas podem ser delegadas à instâncias de computação bastante poderosas localizadas dentro de *datacenters*, sob responsabilidade dos provedores de serviço. O amplo acesso via Internet, bem como a

visão única de sistema oferecida pela computação em nuvem representam o que Tanenbaum e Coulouris definem como computação distribuída, associada a uma visão mercadológica acerca da tecnologia da informação.

Neste contexto, a virtualização e o *pooling* de recursos tomam um papel fundamental tanto para o modelo de negócios da computação em nuvem bem como para a entrega da tecnologia de forma quantificável para o usuário final.

2.2 Virtualização

A tecnologia da virtualização permite que um servidor que execute uma única função possa abrigar vários servidores virtuais, cada um executando seu próprio conjunto independente de tarefas.[8]

Para isso, é utilizada uma camada de software chamada *hypervisor*, ou hipervisor, a fim de permitir que os recursos físicos sejam virtualizados, separando logicamente os mesmos entre os servidores virtuais.

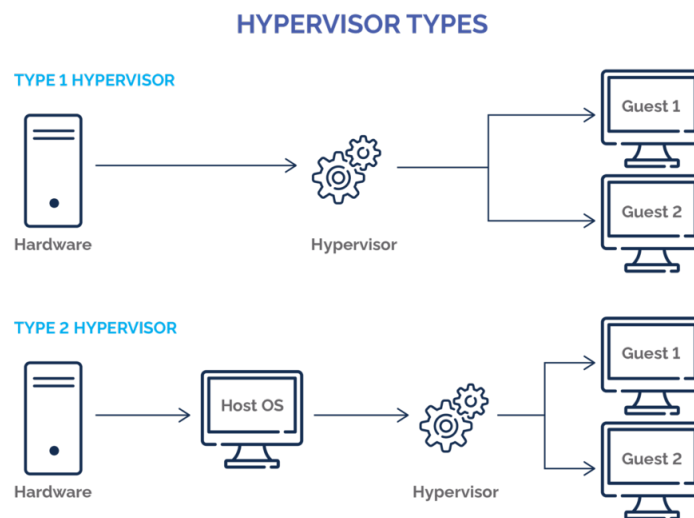


Figura 2 – Diferenciação entre tipos de hipervisores - Fonte: [Park Place Technologies: What Is a Hypervisor? – Types, Benefits & How Does It Work?. 2023](#)

A Figura 2 mostra as diferenças entre os dois tipos de hipervisores existentes. Os hipervisores de tipo 1 são aqueles que são executados diretamente no *hardware*, sem a necessidade de um sistema operacional intermediário entre os recursos computacionais físicos e virtuais. Este tipo de hipervisor é mais comum em *server farms* e *datacenters*, já que seu objetivo é abstrair completamente o conjunto de recursos disponível. Exemplos deste tipo de hipervisor são [VMWare ESXi](#), [Microsoft Hyper-V](#) e [Citrix Hypervisor](#).

Os hipervisores de tipo 2, por sua vez, requerem um sistema operacional na máquina

física a fim de executarem o processo de virtualização. Este tipo de hipervisor depende do gerenciamento de hardware provido pelo sistema operacional preexistente, processo este que introduz latência intrínseca ao processo de virtualização. Dito isso, este tipo de hipervisor é voltado para usuários comuns, onde a performance não é um fator limitante para o uso. Exemplos deste tipo de hipervisor são [QEMU](#), [Oracle Virtualbox](#) e [VMWare Workstation](#).

Neste trabalho, um dos principais objetivos está em como a virtualização pode ser utilizada para se realizar a simulação de redes por meio do GNS3.

2.2.1 KVM

A tecnologia KVM, *Kernel-based Virtual Machine*, é uma maneira que permite executar processos de virtualização a partir do sistema operacional Linux. De forma específica, o KVM utiliza o Linux como um hipervisor de tipo 1, sendo apoiado pelo conjunto específico de instruções do processador (Intel VT-x ou AMD-V).

De acordo com Yasunori Goto[7], o mecanismo de funcionamento do KVM envolve dois componentes de importância fundamental: o emulador de hardware QEMU e as instruções Intel VT-x. Esse conjunto de softwares permite extrair da virtualização a melhor performance possível, gerenciar os recursos de forma mais eficaz e promover uma segurança mais endurecida, uma vez que podemos integrar recursos nativos de segurança do Linux (SELinux). A Figura 3 ilustra o fluxo das instruções de processador.

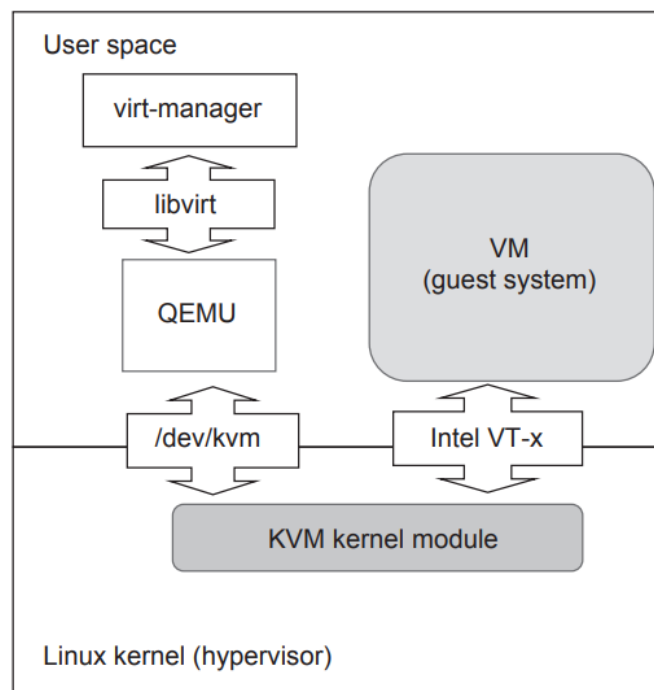


Figura 3 – Esquemático do funcionamento do KVM. Fonte: Kernel-based Virtual Machine Technology (2011), GOTO, Yasunori

Como é possível observar, o KVM promove uma ponte entre o espaço do usuário

e o espaço do kernel. No espaço de usuário, são invocadas as bibliotecas contendo as funções necessárias para a execução do QEMU, que por sua vez, define quais funções serão necessárias para servir a máquina virtual. O QEMU então solicita ao espaço do kernel as instruções de processador a fim de permitir a execução das funções que a máquina virtual necessita, executando, portanto, um sistema operacional inteiro como um processo da máquina hospedeira.

Considerados estes pontos, a influência deste tipo de virtualização para o GNS3 se torna clara, uma vez que os servidores GNS3 podem utilizar do KVM associado ao QEMU para executar uma miríade de *appliances* distintos[3]. Logo, a presença de um processador capaz de executar instruções de virtualização é fundamental e preponderante para o funcionamento pleno do GNS3 em seu máximo potencial.

2.3 GNS3

2.3.1 Graphical Network Simulator 3

O GNS3 é uma ferramenta de simulação de redes utilizada por engenheiros de rede e estudantes de tecnologia da informação. [De acordo com a documentação oficial](#)[3], com ele é possível emular, configurar, testar e verificar redes reais e virtuais.

A documentação base, no site oficial, é bastante esparsa, trazendo apenas os tópicos básicos de instalação, uso, requisitos mínimos e alguns tutoriais. Porém, existe a documentação para desenvolvedores, com mais detalhes técnicos acerca do funcionamento, modelagem e arquitetura da solução. Além disso, o conteúdo comunitário presente no [GitHub](#)[9] e nos [fóruns oficiais](#) é rico em informações, então detalhes mais específicos serão referenciados também por essas fontes.

2.3.2 A emulação de appliances

O GNS3 pode ser integrado a alguns hipervisores para expandir sua capacidade de simulação de topologias de rede mais realistas. Por padrão, vêm instalados poucos componentes, permitindo apenas a verificação de conceitos básicos de interconexão de computadores, mas esta funcionalidade pode ser expandida por meio da instalação de componentes externos. Há uma variedade de *appliances* presentes na página oficial do [GNS3 Marketplace](#), onde os usuários da ferramenta podem escolher o componente que mais se adequa às necessidades e importá-lo para dentro do simulador. A grande vantagem desta funcionalidade do GNS3 é trazer a versatilidade para seu uso, dado que as possibilidades de cenários simulados estarão restritos apenas à disponibilidade de recursos computacionais à disposição do usuário.

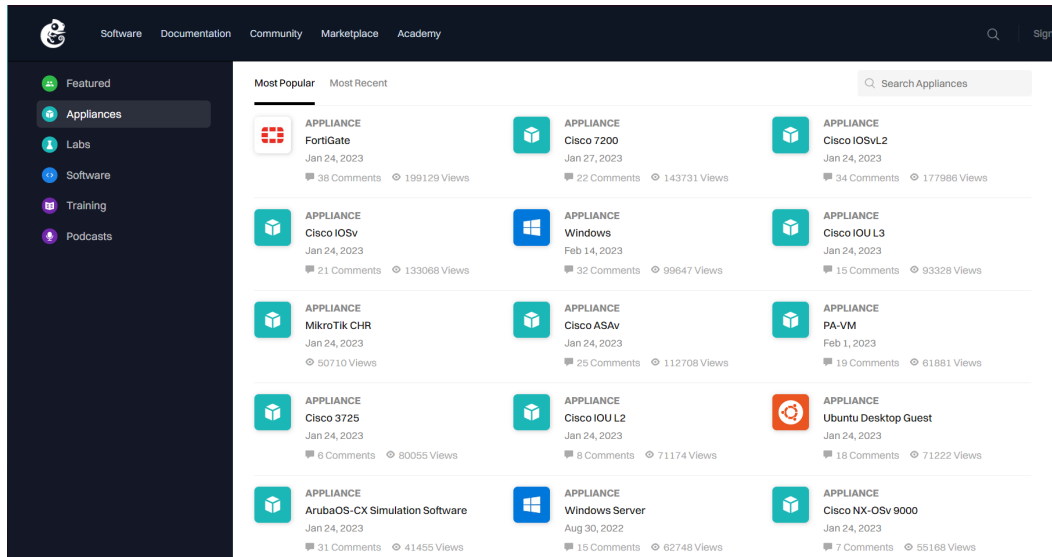


Figura 4 – Excerto dos appliances disponíveis no GNS3 Marketplace [GNS3 Marketplace](#). 2023

A Figura 4 mostra uma captura de tela do *marketplace* oficial do GNS3. Além de componentes externos, podemos também adquirir topologias de rede já pré-configuradas. Na Figura 4, podemos notar que a variedade de componentes é vasta, incluindo ativos de rede como roteadores e *switches* de vários fabricantes, bem como sistemas operacionais diversos, além de appliances utilitários.

Novamente, a virtualização tem um papel chave na solução, já que uma das principais vantagens do GNS3 é possibilitar o uso de quase qualquer *appliance* compatível com virtualização.

2.3.3 Arquitetura da ferramenta

Os componentes do GNS3 são:

- As interfaces de visualização;
- As unidades de execução;
- A controladora;
- Os emuladores.

As interfaces de visualização englobam a interface gráfica baseada em QT *gns3-gui* e a interface *web gns3-web*. As interfaces de execução são os executáveis *gns3-server* ou uma GNS3 VM. A GNS3 VM por sua vez é uma imagem de máquina virtual que executa o sistema operacional Linux com a distribuição Ubuntu, com o *gns3-server* e o conjunto de aplicações para virtualização de *appliances*. Na [página da documentação para desenvolvedores](#), há um

diagrama simples de como os componentes do *software* se relacionam. A Figura 5 é uma recriação do diagrama presente na documentação.

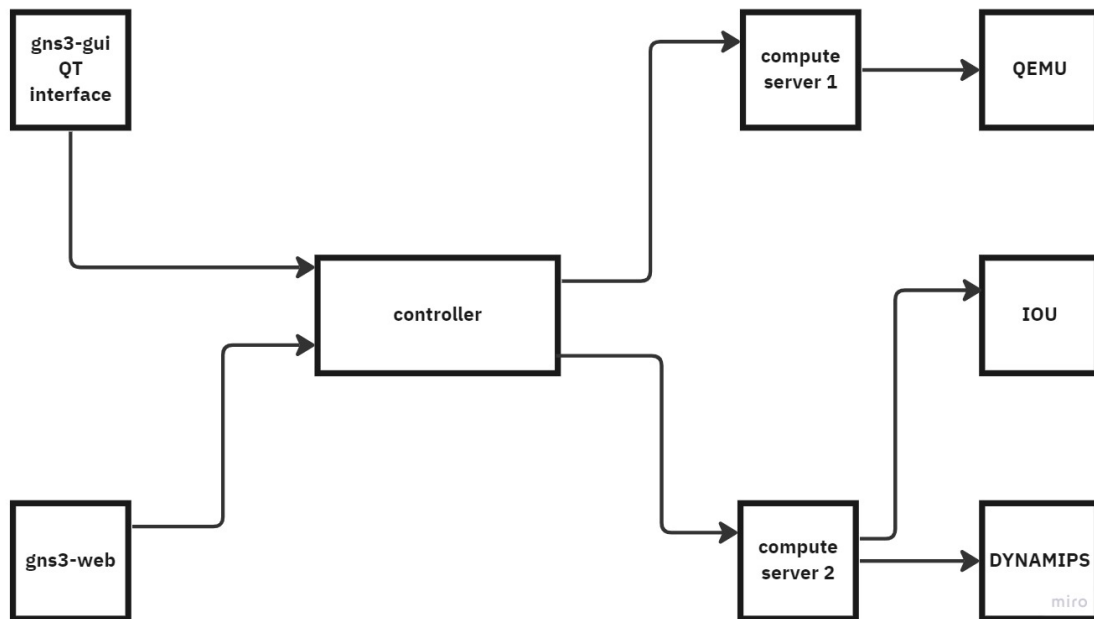


Figura 5 – Arquitetura do GNS3 - Fonte: [GNS3 Developer Docs v2.2.38](#)

Dessa forma, os componentes interagem entre si de acordo com a topologia da Figura 5, por meio de API REST que trafegará dados em JSON. A comunicação é mediada por meio da controladora, que é manifestada na forma de um projeto. O projeto abriga os metadados que serão fornecidos para a API, como os *appliances* utilizados, como estão conectados e que tipo de emuladores estão utilizando. Por sua vez, os emuladores executarão as imagens instaladas dos *appliances*.

2.3.4 O uso da ferramenta

Existem várias formas de se utilizar o GNS3. Podemos instalá-lo nos sistemas operacionais Windows, Linux, Mac OS, cada um destes podendo utilizar a interface QT ou Web. Ainda, podemos manifestar a unidade de execução tanto pelo executável local quanto pelo uso da GNS3 VM. Além disso, é possível configurar o uso de mais de uma unidade de execução remota para realizar balanceamento de carga. Esta funcionalidade será bastante explorada, visto que na implementação final teremos unidades de execução presentes em ambiente remoto.

Por meio da interface QT, o gerenciamento e configuração são mais amigáveis aos usuários. [A documentação oficial](#) possui guias de como proceder a fim de realizar a instalação básica, porém, após extensa série de experimentos, foi obtida a configuração otimizada para o melhor resultado.

3 Metodologia

Neste trabalho serão demonstrados os resultados das análises laboratoriais da implementação do simulador GNS3 em três cenários: primeiramente o uso tradicional, apenas em um computador, para que seja obtida uma linha de base para o consumo dos recursos computacionais. Em seguida, será analisada a implementação usando uma instância remota do simulador, a fim de aferir a redução do uso dos recursos na máquina principal. Além disso, busca-se a implementação em uso distribuído do GNS3 num ambiente de nuvem computacional. Para tal fim, será implementada uma topologia de redes como referência base, e distribuir-se-á o processamento desta simulação em diferentes estilos, a fim de compilar os resultados. Por fim, será apresentada uma maneira de replicar a configuração do simulador para uso distribuído.

Na Tabela 2, é apresentado um resumo de como os cenários para a realização do experimento irão se desenrolar ao longo da pesquisa.

Tabela 2 – Descrição dos cenários a serem testados experimentalmente para avaliar a performance de um sistema de simulação de redes distribuído.

Cenário 01	Cenário 02	Cenário 03
Implementação Local	Implementação em nuvem pública	Implementação distribuída virtualmente
Há apenas uma instância do GNS3, representada por uma única máquina virtual GNS3 VM, que executará toda a topologia de referência.	Haverão duas instâncias do GNS3, onde na bancada local teremos uma GNS3 VM, e em nuvem teremos um servidor Linux, podendo ser escalonado conforme a necessidade de recursos computacionais.	Haverá duas instâncias do GNS3, uma representada pela GNS3 VM, outra representada por uma máquina Linux, conectadas por uma interface de rede virtual.

3.1 Configuração da base de testes

Para este estudo, será implementado um laboratório envolvendo duas máquinas virtuais. Uma delas é a máquina virtual Linux rodando o simulador GNS3, a outra, é um servidor Linux, também rodando o GNS3. As máquinas virtuais estão hospedadas no virtualizador VMWare Workstation 17. A Tabela 3 mostra onde serão executadas as duas instâncias do simulador, apontando os recursos computacionais disponíveis no agregado.

Tabela 3 – Descrição da configuração das máquinas virtuais do ambiente de laboratório. Fonte: Elaboração própria

	GNS3 VM	LINUX SERVER
VERSÃO DO SISTEMA OPERACIONAL	UBUNTU 20.04	UBUNTU 20.04
VERSÃO DO SIMULADOR	v2.2.41	v2.2.41
CPU ALOCADA	4 vCPUs	8 vCPUs
MEMÓRIA RAM ALOCADA	16 GB	12 GB

Como indicado, há um total de 28 Gigabytes de memória RAM e 12 vCPUs. A comunicação entre estas máquinas se dá por meio da interface de rede virtual criada pelo

virtualizador, que por sua vez, atua como um switch de camada 2, conforme ilustrado na figura 6.

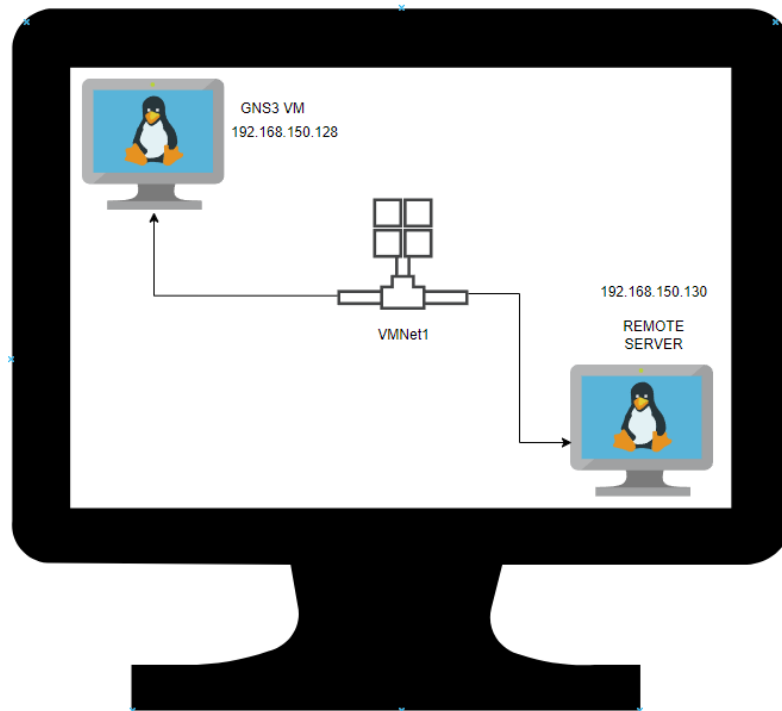


Figura 6 – Rede criada pelo virtualizador VMWare entre as máquinas virtuais da bancada de testes.
Fonte: Elaboração própria.

A fim de permitir que as máquinas virtuais se comuniquem por meio da interface de rede virtual, deve-se configurar no virtualizador o seguinte adaptador de rede:

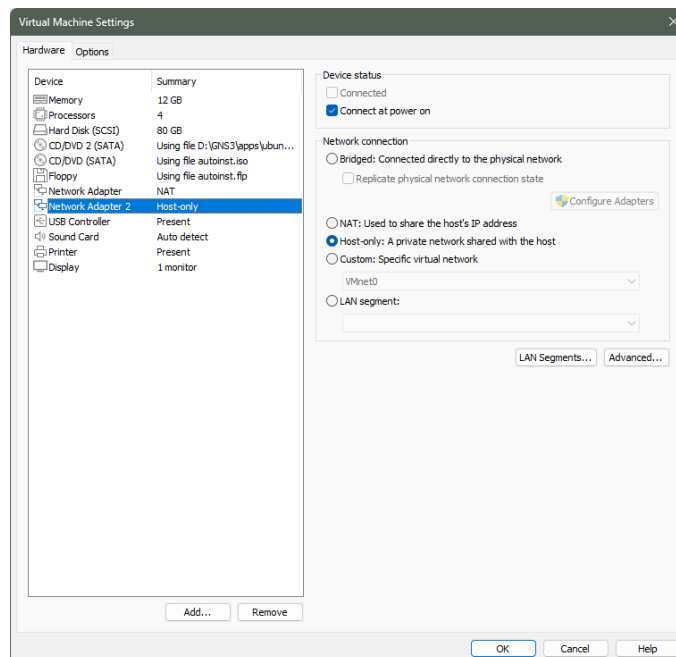


Figura 7 – Configuração necessária para implementar uma rede virtual entre as máquinas virtuais.

Repetindo a configuração da figura 7 para todas as máquinas, permite-se a comunicação entre as mesmas.

Com as máquinas já devidamente configuradas e com os sistemas operacionais instalados, deve-se instalar e configurar o simulador. Este passo possui três etapas: instalação da GNS3 VM, instalação do GNS3 no servidor remoto, e a instalação do cliente local para o uso do simulador.

3.2 Instalação e configuração do GNS3

3.2.1 Instalação da GNS3 VM

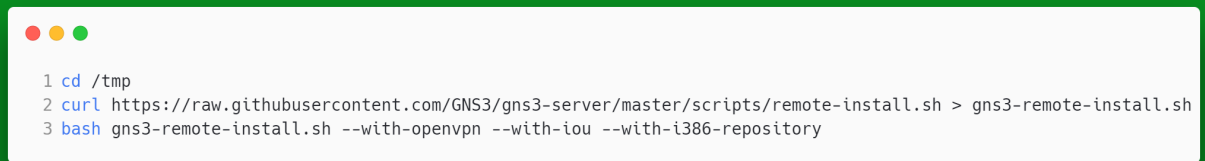
Considerando as instruções presentes na documentação oficial do simulador[3], a instalação da GNS3 VM se dá de maneira direta por meio da importação de uma imagem de disco pré-configurada no virtualizador de preferência. No presente caso, foi escolhido o virtualizador VMWare Workstation Player, logo, foi instalada a versão para esta solução específica.

O passo a passo para esta instalação é direto:

- Acessar a página do GitHub contendo as versões da GNS3 VM e obter o arquivo de imagem de disco apropriado;
- Criar uma máquina virtual a partir da imagem de disco apropriada;
- Iniciar a máquina virtual e tomar nota do endereço de IP entregue pela interface de rede virtual.

3.2.2 Instalação no servidor remoto

Para o servidor remoto, os passos são ainda mais simples. A equipe de desenvolvimento do GNS3 fornece um script de instalação que pode ser executado em qualquer máquina Linux. Quando o script for executado, este instalará as ferramentas necessárias para o simulador, bem como apresentar a opção de instalar um servidor de VPN para prover a conexão segura entre o cliente e o servidor.



```
1 cd /tmp
2 curl https://raw.githubusercontent.com/GNS3/gns3-server/master/scripts/remote-install.sh > gns3-remote-install.sh
3 bash gns3-remote-install.sh --with-openvpn --with-iou --with-i386-repository
```

Figura 8 – Script de instalação do GNS3 em servidor Linux remoto.

O script presente na Figura 8 possui algumas flags auxiliares, como *-with-openvpn*, que instala o servidor de VPN, ou *-with-iou*, para uso de emulação de equipamentos Cisco.

O servidor remoto, após a instalação, pode ser usado tanto como servidor principal como um servidor auxiliar, que é o caso deste trabalho.

3.2.3 Configuração do client

A instalação do client do GNS3, que abriga a interface gráfica, é simples e direta, bastando executar o *wizard* de instalação. Este instalador, assim como a GNS3 VM, pode ser obtido na página do GitHub. Contudo, a configuração inicial não executa o simulador da maneira mais otimizada para o uso comum, tampouco para as análises presentes neste trabalho. Segue, então, o passo a passo para a configuração otimizada:

Após a instalação do client, acessar as preferências do simulador por meio do atalho `Ctrl + Shift + P` e encontrar a seção GNS3 VM, como na figura 9.

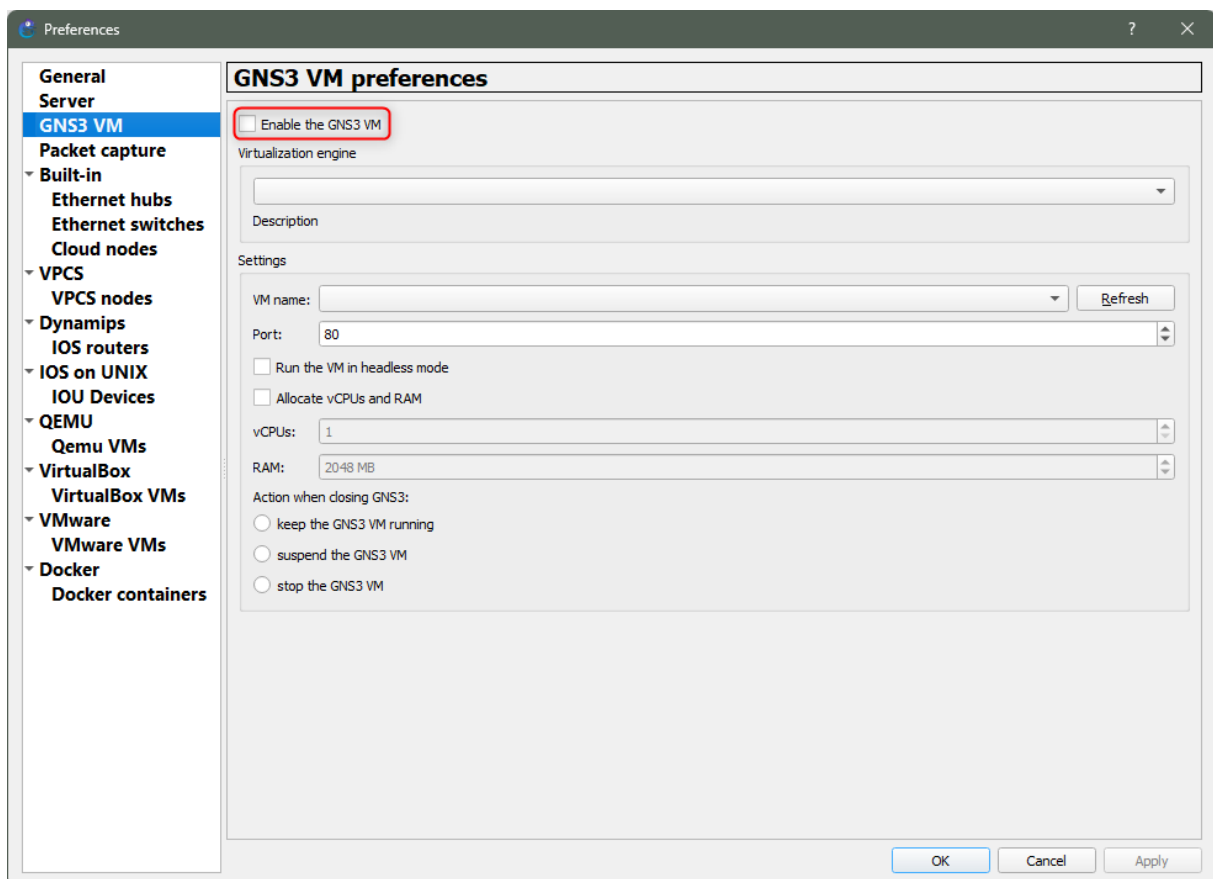


Figura 9 – Configurações relacionadas a GNS3 VM. Fonte: elaboração própria

Como ilustrado, deve-se desmarcar a opção *Enable the GNS3 VM*. Em seguida, na seção Server, aplicar as configurações como indicado na figura 10:

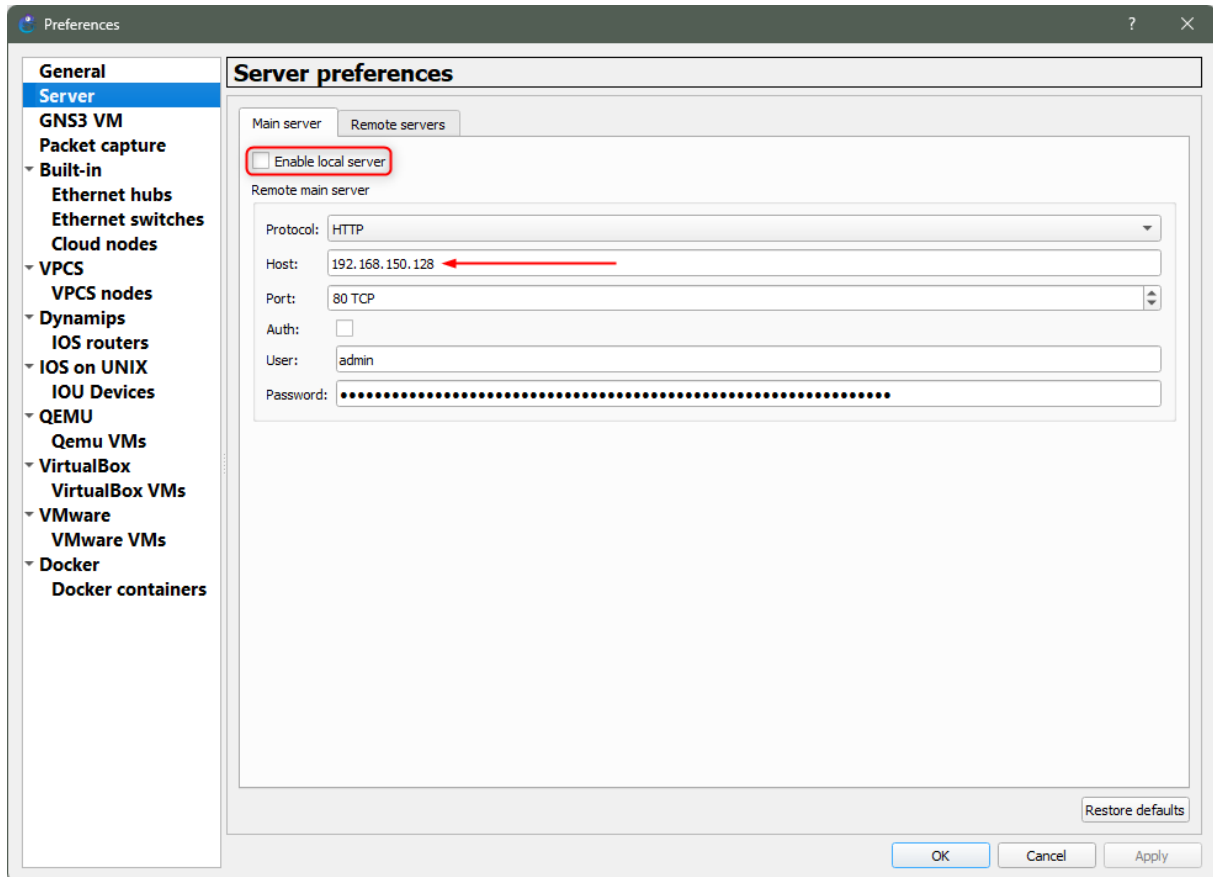


Figura 10 – Configurações relacionadas ao servidor de simulação. Fonte: elaboração própria

Deve-se desativar o servidor local e acionar o uso de um servidor principal remoto. O servidor principal remoto a ser configurado será a GNS3 VM, instalada como na seção 3.2.1. O endereço IP coletado da VM naquele passo então deve ser inserido no campo *Host*, bem como a porta, no campo *Port*.

Finalmente, com o servidor remoto também implementado, deve-se acessar a aba *Remote Servers* como na Figura 11 e inserir o endereço IP coletado do servidor remoto. A fim de coletar o endereço IP do servidor remoto, executa-se o comando `ip -c a`, para exibir as configurações das interfaces de rede do sistema, e toma-se a nota do endereço IP pertencente à rede provida pela interface de rede compartilhada do virtualizador. A Figura 12 indica a coleta do IP do servidor remoto.

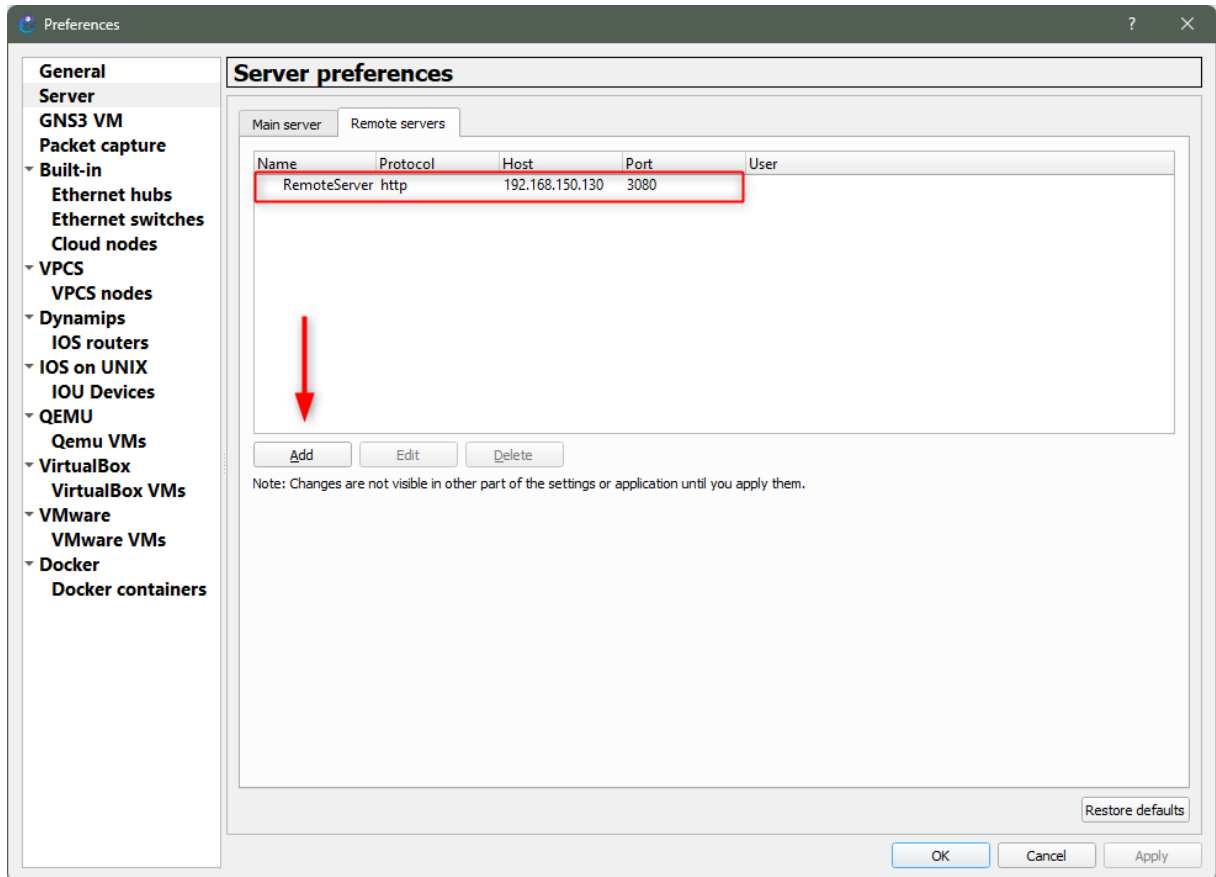


Figura 11 – Configurações relacionadas ao servidor remoto de simulação. Fonte: elaboração própria

```

sysadmin@lab-server:~$ ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 00:0c:29:1c:a8:10 brd ff:ff:ff:ff:ff:ff
   inet 192.168.242.132/24 brd 192.168.242.255 scope global dynamic ens33
       valid_lft 911sec preferred_lft 911sec
   inet6 fe80::20c:29ff:fe1c:a810/64 scope link
       valid_lft forever preferred_lft forever
3: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 00:0c:29:1c:a8:1a brd ff:ff:ff:ff:ff:ff
   inet 192.168.150.130/24 brd 192.168.150.255 scope global dynamic ens34
       valid_lft 910sec preferred_lft 910sec
   inet6 fe80::20c:29ff:fe1c:a81a/64 scope link
       valid_lft forever preferred_lft forever
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen
1000
   link/ether 52:54:00:9a:1a:e8 brd ff:ff:ff:ff:ff:ff
   inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default
qlen 1000
   link/ether 52:54:00:9a:1a:e8 brd ff:ff:ff:ff:ff:ff
6: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
   link/ether 02:42:aa:d4:db:dc brd ff:ff:ff:ff:ff:ff
   inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
sysadmin@lab-server:~$ _

```

Figura 12 – Coletando o IP da interface compartilhada do servidor remoto. Fonte: elaboração própria

Após a aplicação destas configurações, a bancada de testes estará pronta para uso e análise. Os mesmos passos apresentados nesta seção podem ser utilizados para o uso do GNS3 em infraestrutura distribuída, cabendo apenas a alteração dos endereços IP e, se necessário, a implementação de infraestrutura de redes associada a fim de alocar a GNS3 VM e o servidor remoto na mesma rede.

4 Análises e Resultados

4.1 Topologia de referência - componentes e uso base de recursos

Nas seguintes análises, utilizaremos a topologia de referência da Figura 13:

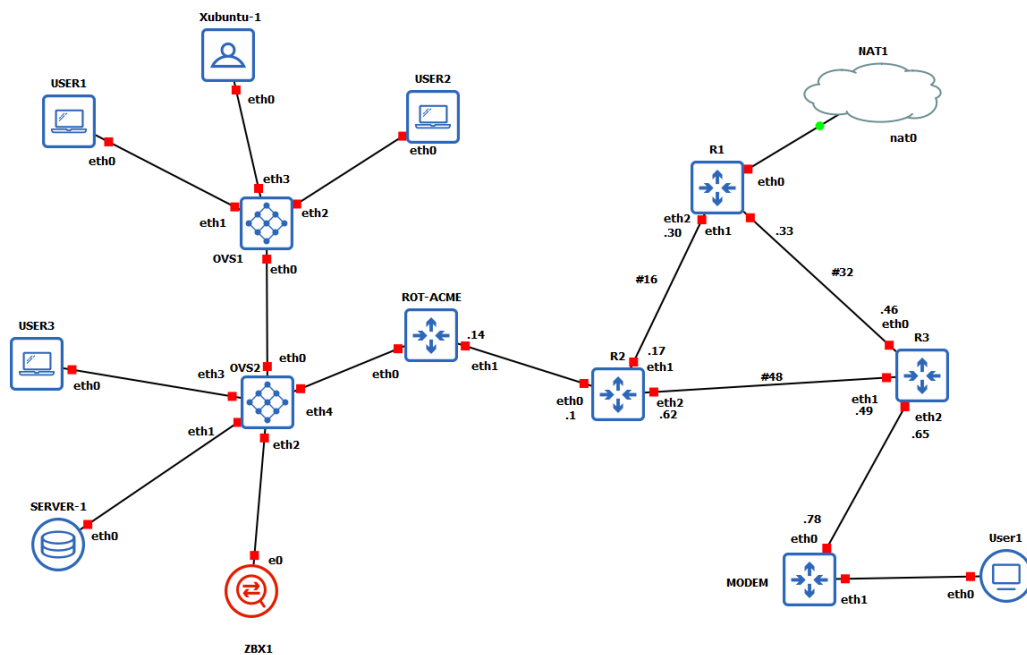


Figura 13 – Topologia de referência para os cenários de análise

Na topologia, há uma heterogeneidade em relação ao método que o simulador utiliza para invocar os equipamentos. Enquanto alguns são virtualizados por meio da tecnologia KVM, usando o QEMU para a interação, outros são *containers* Docker.

A hipótese que este trabalho visa explorar repousa no uso da computação distribuída para a redução da carga computacional em instâncias isoladas de computação. Ou seja, as métricas de *baseline* calculadas devem diminuir, ao alterarmos onde os *appliances* virtualizados forem executados.

A Tabela 4 traz o demonstrativo dos componentes da topologia e seus parâmetros de consumo máximo. No agregado, o consumo de memória RAM da topologia e de vCPUs é de 16,5 GB e 19, respectivamente.

Tabela 4 – Elementos que compõem a topologia de referência, com suas especificações de consumo e tipo de virtualização.

Appliance	RAM	vCPU	Interfaces de Rede	Virtualização
R1	512 MB	1	3	QEMU
R2	512 MB	1	3	QEMU
R3	512 MB	1	3	QEMU
ROT-ACME	512 MB	1	3	QEMU
MODEM	512 MB	1	3	QEMU
OVS1	-	-	16	Docker
OVS2	-	-	16	Docker
SERVER-1	2048 MB	2	1	QEMU
ZBX-1	2048 MB	2	1	QEMU
USER1	2048 MB	2	1	QEMU
User1	2048 MB	2	1	QEMU
USER2	2048 MB	2	1	QEMU
USER3	2048 MB	2	1	QEMU
Xubuntu-1	2048 MB	2	1	QEMU
NAT	-	-	1	-

4.2 Cenário 1 - Uso local do GNS3

Neste cenário, iremos implementar uma topologia no GNS3, executando os componentes da topologia somente na GNS3 VM. Este cenário é o mais comum no ambiente acadêmico e profissional, e exige um quantitativo grande de recursos mesmo em topologias de complexidade não tão intensa.

Dessa forma, foi implementada a topologia da Figura 13. Os *appliances* foram configurados devidamente, e foi possível traçar o seguinte perfil de uso. Há um comportamento peculiar no consumo de recursos pelo simulador próximo ao início da simulação, quando o simulador precisa executar quase simultaneamente todos os processos de boot nas máquinas virtuais. Após este período inicial, o comportamento se torna aproximadamente constante. A Figura 14 traz os gráficos relativos à este cenário.

Como é possível observar, a memória RAM disponível decai com um perfil quase linear até atingir o patamar de aproximadamente 5,8 GB, o que indica um consumo de aproximados 6 GB, em condição normal de uso. A carga computacional nos cores do processador virtual apresentou o comportamento esperado também, com um pico de 100% de carga durante o boot dos *appliances*, apresentando uma queda abrupta para o patamar dos 3% a 5%, com picos ocasionais de 40 a 60%.

Considerou-se também o cenário de estresse da topologia, com todos os *appliances* sendo executados em potencial máximo.

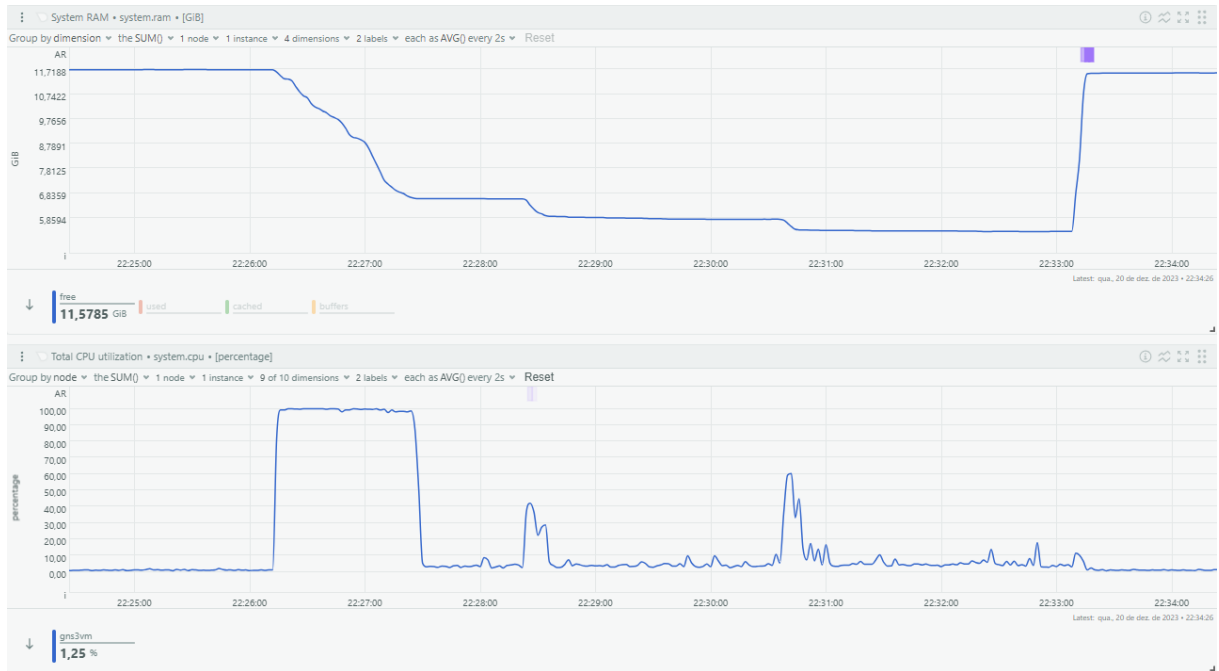


Figura 14 – Consumo de recursos - vCPU e RAM para a topologia de referência

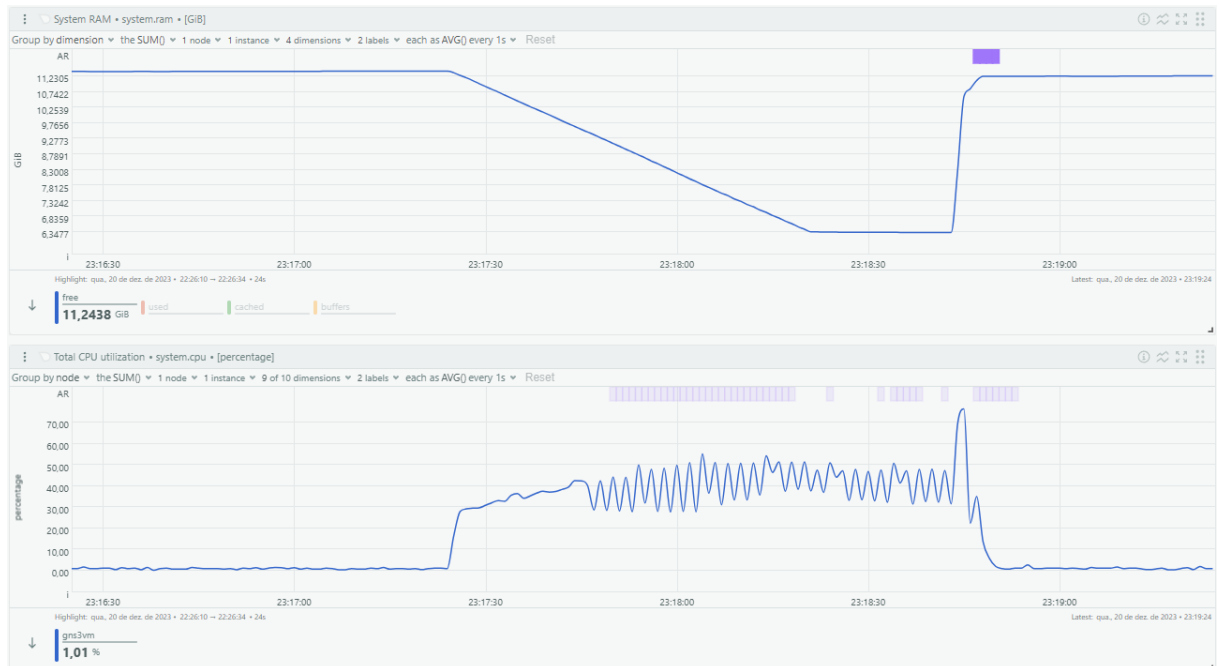


Figura 15 – Consumo de recursos - vCPU e RAM para a topologia de referência - cenário de uso intenso

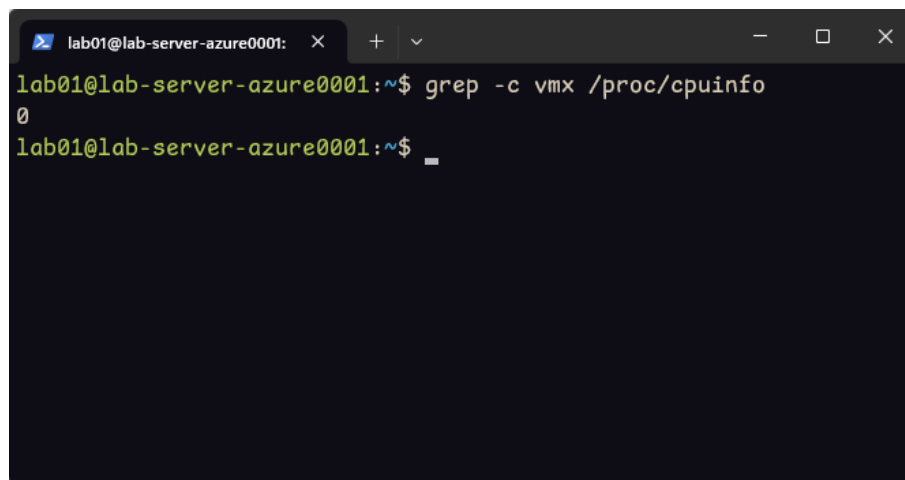
Ao utilizar a topologia com todas as funcionalidades simultaneamente, nota-se a elevação na carga experimentada pelo processador. Enquanto no cenário simples de pós configuração os picos eram menos frequentes, neste caso a carga foi bem mais intensa, tendo uma média de 40%, chegando a um máximo local em 70%.

4.3 Cenário 2 - Uso do GNS3 em ambiente de nuvem computacional

Conforme discorreu a seção 2.1.1, a nuvem computacional é, na perspectiva do usuário final, um *pool* praticamente infinito de recursos, limitado somente ao custo de utilização. Dessa forma, a pesquisa caminhou visando executar um servidor remoto em nuvem, para que o servidor pudesse ser ajustado de forma mais simplificada à medida em que fosse necessário.

Contudo, como discutido no presente estudo, uma das tecnologias mais importantes (senão imperativas) para o uso total do GNS3 é a da virtualização com base no KVM. Executando a instalação como na seção 3.2.2, foi feito um teste para aferir a possibilidade de execução do KVM. A seção 2.2.1 cita que é necessário o acesso à instruções do processador para a virtualização desta forma, e no caso do kernel Linux sobre um processador Intel, as instruções se chamam simplesmente *vmx*. O teste consiste em acessar os metadados do processador da máquina virtual em nuvem e buscar na relação de instruções, a instrução *vmx*. Com o comando `grep vmx /proc/cpuinfo`, é possível aferir a contagem de ocorrências da instrução requerida nas informações de sistema.

Ao executar o comando em diversas máquinas, em diversos provedores de nuvem, o resultado foi consistentemente nulo, como na Figura 16.



```
lab01@lab-server-azure0001: ~$ grep -c vmx /proc/cpuinfo
0
lab01@lab-server-azure0001: ~$
```

Figura 16 – Teste da presença de processadores com acesso à instrução *vmx*.

Após mais pesquisas, foi constatado que este comportamento se dá devido ao modelo de negócio dos provedores de nuvem, de forma geral. As máquinas virtuais são pagas conforme o consumo, mas especialmente, pela unidade. A precificação varia entre os provedores[12], mas o modelo é o mesmo. Assim, é bastante coerente que a instrução *vmx* não seja disponibilizada nas máquinas virtuais em nuvem, uma vez que tal prática traria a possibilidade dos usuários subverterem a lógica comercial deste produto de tecnologia da informação.

Considerados estes pontos, foi então considerada a implementação em um servidor remoto implementado numa infraestrutura com um nível de acesso mais profundo aos recursos computacionais.

4.4 Cenário 3 - Uso do GNS3 com um servidor remoto

Este cenário envolve o teste do simulador em questão numa infraestrutura distribuída. Para esta etapa do experimento, utiliza-se duas máquinas virtuais: a GNS3 VM e um servidor Ubuntu 20.04, ambas sendo executadas no mesmo virtualizador.

Utilizando um servidor remoto para dividir o uso da carga, foi configurada a mesma topologia em diferentes modos. Primeiramente, os ativos de rede foram configurados no servidor remoto. Posteriormente, os endpoints foram configurados no servidor remoto.

4.4.1 Configurando os ativos de rede no servidor remoto

Neste cenário, configurou-se a topologia normalmente, e assim como no caso de uso local, foram realizados os testes de iniciação e de estresse. A Figura 17 traz o comportamento do teste de iniciação da topologia.

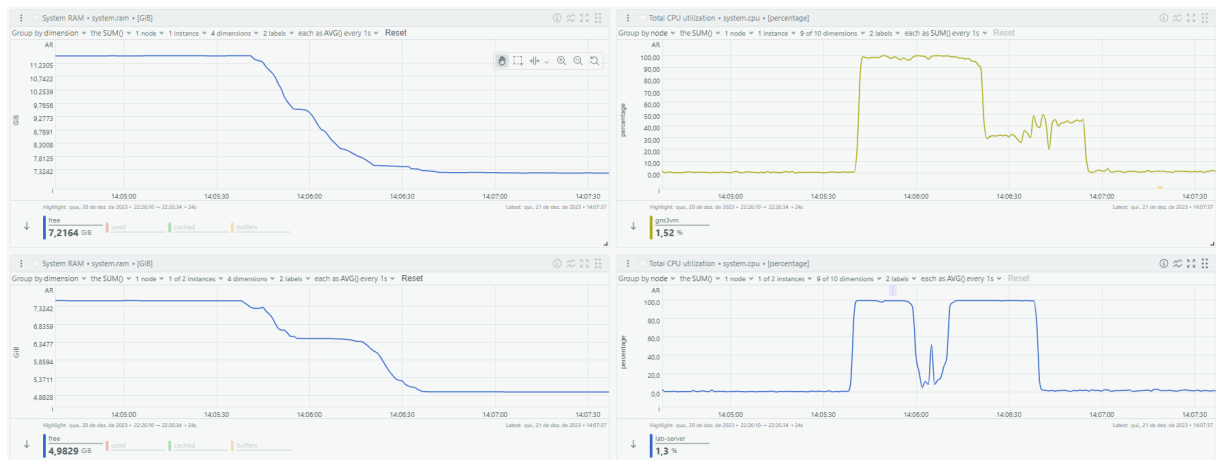


Figura 17 – Gráficos de consumo de CPU e RAM para a topologia distribuída.

Novamente, como na Figura 17, percebe-se o pico em 100% na inicialização dos equipamentos, e depois um limiar entre 3 e 5%. Porém, os picos de consumo no sistema local chegam no máximo a 15%, o que indica grande redução da carga posta sobre o processador virtual. Com a memória RAM, o cenário se repete. O consumo passa a ser na faixa de 3,5 GB, o que representa uma redução de 2,3 GB no sistema local.

Para o teste de estresse, o consumo de CPU local teve um aumento significativo, chegando novamente aos picos de 100%. Isto se deve ao fato de que, na topologia distribuída, além do processamento local, o sistema deve informar ao servidor remoto o status de processamento da tarefa atual. Este aspecto adicional promove uma carga extra no processador,

que para este caso extremo se mostrou relevante. O gráfico presente na Figura 18 permite a análise deste fenômeno.



Figura 18 – Gráficos de consumo de CPU e RAM para a topologia distribuída - cenário de uso extremo

É possível verificar que os picos de 40 a 60% no servidor remoto aconteceram da mesma maneira que no servidor local.

Também é possível aferir a largura de banda no link entre as instâncias do simulador.

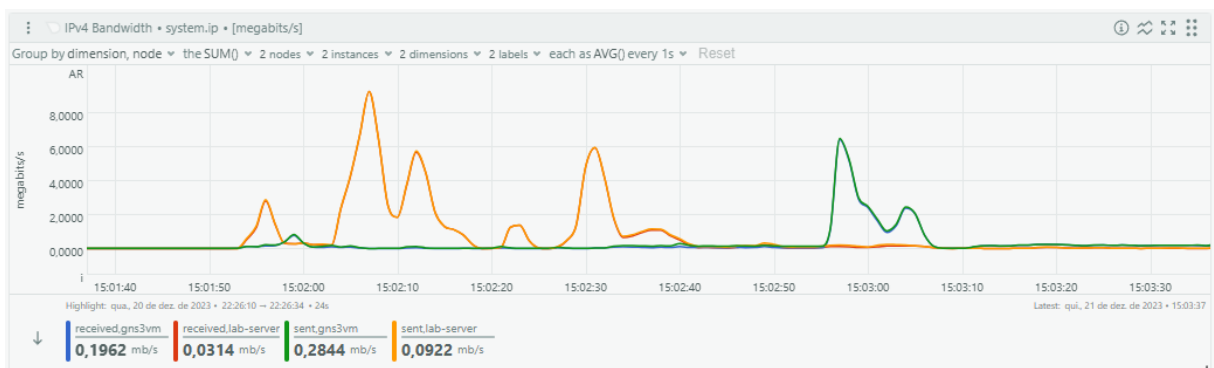


Figura 19 – Gráficos de largura de banda para a topologia distribuída

Como visto na Figura 19, há uma troca constante de mensagens entre as instâncias do simulador, para sincronização do estado da simulação. É possível verificar a dominância inicial do quantitativo de mensagens oriundas do servidor remoto, uma vez que o fluxo de informação nos ativos de rede é superior ao dos appliances no servidor local.

4.4.2 Configurando os appliances no servidor remoto

Neste outro cenário, os appliances serão executados no servidor remoto, e os testes procederão como no cenário anterior.

Iniciando com o teste da inicialização da topologia, analisa-se o gráfico da Figura 20.



Figura 20 – Gráficos de consumo de CPU e RAM para a topologia distribuída.

É clara a redução no consumo de RAM pela máquina local. O patamar inferior se encontra na casa dos 7,5 GB, o que indica um consumo de apenas 4,5 GB, enquanto o servidor remoto consumiu a memória quase que totalmente. A carga no processador apresentou o comportamento esperado, com o platô em 100% presente, mas após este transiente de boot, não houveram picos de carga, com o consumo na faixa dos 2%. O servidor remoto, por sua vez, consumiu menos dos processadores virtuais, mas teve seu transiente com duração mais curta.

Em relação a largura de banda, o comportamento também foi como o esperado, conforme o gráfico na Figura 21.

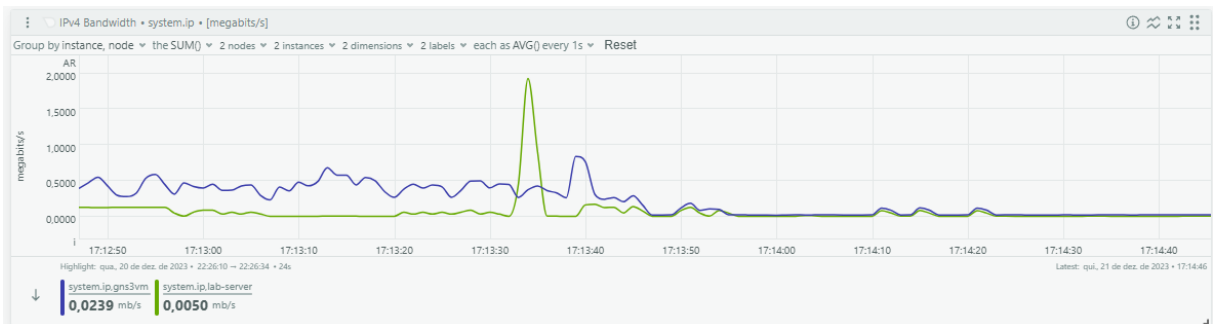


Figura 21 – Gráficos de largura de banda para a topologia distribuída.

Aqui, houve uma troca de mensagens mais discreta, com um pico ocorrido após a estabilização da topologia.

5 Conclusões

Conforme foi verificado nos experimentos, o uso de uma instância remota para a execução remota do simulador GNS3 é benéfica ao usuário, pois a carga é reduzida no computador pessoal. Isto comprova a eficácia da computação distribuída na tarefa de dividir o custo computacional de tarefas altamente dispendiosas. Há algumas ressalvas neste quesito, pois o resultado destes experimentos não é completamente conclusivo, uma vez que ainda existem casos de 100% de utilização de recursos como CPU e memória RAM. De fato, a bancada de testes envolvida neste experimento não possuía um agregado de recursos a fim de permitir a implementação num cenário de uso extremo com valores absolutos de consumo mais altos, o que certamente mudaria a porcentagem de alívio no computador pessoal do usuário final.

Para trabalhos futuros, faz-se necessária a repetição do experimento aqui discutido. Aumentar a quantidade de máquinas executando as instâncias do GNS3 certamente aumentará a largura de banda necessária para a configuração de um sistema de simulação de redes, mas esta hipótese deve ser comprovada experimentalmente. A adição de mais instâncias pode reduzir ainda mais o consumo de cada sistema individual, porém, o percentual deve ser também explorando a fundo.

Eis a importância de estudos futuros relacionados à execução deste experimento em nuvem computacional, devido à alta disponibilidade de recursos, bem como a escalabilidade e elasticidade associadas a este paradigma de computação. Outro fator a ser estudado futuramente é o Bare-Metal-as-a-Service, ou BMaaS, que traz dentro das nuvem públicas a possibilidade de configuração da torre de virtualização em maior profundidade a fim de entregar a tecnologia do KVM para os usuários, permitindo assim o uso do GNS3 em nuvem.

Além disso, também pode ser estudada futuramente a possibilidade de implementação de um compêndio de imagens de appliances feitas para uso com a tecnologia Docker, que remove a necessidade de um hipervisor dos servidores. Tal fato também possibilitaria o uso do simulador em nuvem pública.

Em suma, até o presente momento, não é possível executar o simulador GNS3 em nuvem computacional devido às limitações na infraestrutura dos provedores de nuvem pública, muito associadas ao modelo de negócios da nuvem. Para um primeiro momento, é fato que a computação distribuída pode permitir o uso da ferramenta de forma mais otimizada, reduzindo o consumo de recursos, mas mais experimentos se fazem necessários a fim de extrair conclusões mais definitivas acerca do tema.

6 Referências bibliográficas

1. Van Steen, M., Tanenbaum, A. S. (2017). Distributed systems (p. 20). Leiden, The Netherlands: Maarten van Steen.
2. Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., ... & Yu, H. (2000). Advances in network simulation. *Computer*, 33(5), 59-67.
3. GNS3 Documentation. (2023). Disponível em <https://docs.gns3.com/>
4. Park Place Technologies: What Is a Hypervisor? – Types, Benefits & How Does It Work? (2023) Disponível em <https://www.parkplacetechnologies.com/blog/what-is-hypervisor-types-benefits/>
5. Coulouris, G., Dollimore, J., & Kindberg, T. (2001). Distributed Systems: Concepts and Design.
6. Lindsay, D., Gill, S.S., Smirnova, D. *et al.* The evolution of distributed computing systems: from fundamental to new frontiers. *Computing* **103**, 1859–1878 (2021).
7. Kernel-based Virtual Machine Technology (2011), GOTO, Yasunori
8. Rodríguez-Haro, F., Freitag, F., Navarro, L., Hernández-sánchez, E., Farías-Mendoza, N., Guerrero-Ibáñez, J. A., & González-Potes, A. (2012). A summary of virtualization techniques. *Procedia Technology*, 3, 267-272.
9. GNS3 GitHub - Releases (2023) Disponível em <https://github.com/GNS3/gns3-gui/releases>
10. National Institute of Standards and Technologies, NIST, SP 800-145, (2011)
11. National Institute of Standards and Technologies, NIST, SP 800-146, (2011)
12. [Cloud Pricing Comparison 2023: AWS vs Azure vs Google Cloud \(2021\)](#)