

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES**

ANDERSON FINSTERBUSCH

**ESTUDO DE VIABILIDADE DE USO DE CUBIETRUCK COMO
SERVIDOR PARA PEQUENAS EMPRESAS**

MONOGRAFIA DE ESPECIALIZAÇÃO

**PATO BRANCO
2015**

ANDERSON FINSTERBUSCH

**ESTUDO DE VIABILIDADE DE USO DE CUBIETRUCK COMO
SERVIDOR PARA PEQUENAS EMPRESAS**

Trabalho de Conclusão de Curso, apresentado ao II Curso de Especialização em Redes de Computadores – Configuração e Gerenciamento de Servidores e Equipamentos de Redes, da Universidade Tecnológica Federal do Paraná, câmpus Pato Branco, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. MSc. Adriano Serckumecka.

**PATO BRANCO
2015**

TERMO DE APROVAÇÃO


Estudo de Viabilidade de Uso de Cubietruck como Servidor para Pequenas Empresas

por

Anderson Finsterbusch

Esta monografia foi apresentada às 10h00min do dia 27 de outubro de 2015, como requisito parcial para obtenção do título de ESPECIALISTA, no II Curso de Especialização em Redes de Computadores – Configuração e Gerenciamento de Servidores e Equipamentos de Redes, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho **aprovado**.

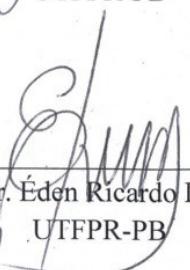
Banca Examinadora



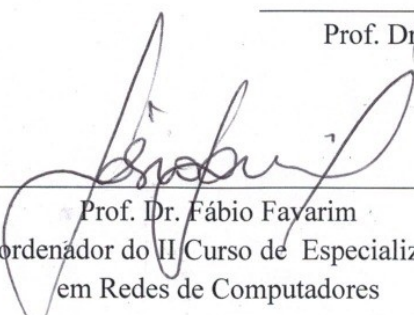
Prof. M.Sc. Adriano Serckumecka
Orientador / UTFPR-PB



Prof. Dr. Fábio Favarim
UTFPR-PB



Prof. Dr. Eden Ricardo Dosciatti
UTFPR-PB



Prof. Dr. Fábio Favarim
Coordenador do II Curso de Especialização
em Redes de Computadores

*Dedico este trabalho a todos
que me apoiaram e acreditaram no
meu potencial, em especial, meus
familiares.*

AGRADECIMENTOS

Eu, Anderson Finsterbusch, primeiramente, agradeço a Deus por ter me dado a oportunidade de estar no mundo, me dando forças para me dedicar todos os dias aos estudos. Ao avô (opa) Sr. Gerhard Finsterbusch, *in memoriam*, que nos momentos que passamos juntos sempre me ensinou os princípios básicos de tecnologia, me incentivando a seguir a carreira que estou. Aos meus pais, Helmuth Finsterbusch e Ana Lourdes R. Finsterbusch, que conseguiram juntos me passar uma ótima educação, sendo honesto, disciplinado, ético e respeitoso com todos. À minha família, agradeço todo o amor, carinho, compreensão e respeito. À minha esposa Vanessa e filho Augusto Gerhard, que conseguiram me auxiliar nessa reta final, me fazendo ter mais ânimo, e são as pessoas mais importantes na minha vida.

Aos meus colegas de curso, que sempre estiveram ajudando, proporcionando vários momentos de alegria e descontração e criando um laço de grande amizade. A Universidade Tecnológica Federal do Paraná, que nos proporcionou essa oportunidade de ensino, com um curso de qualidade, contribuindo para a minha carreira. E por fim, a todos meus professores, seria injusto citar nomes, pois todos que passaram por mim, contribuíram muito para chegar até onde estou. Obrigado.

“A quem me pergunta se sou pessimista ou otimista, respondo que o meu conhecimento é de pessimista, mas a minha vontade e a minha esperança são de otimista.”

Albert Schweitzer

RESUMO

FINSTERBUSCH, Anderson. Estudo de Viabilidade de Uso de Cubietruck como Servidor para Pequenas Empresas. 2015. 74 f. Monografia de Trabalho de Conclusão de Curso (II Curso de Especialização em Redes de Computadores), Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

Hoje, a informatização atinge as mais diversas empresas, e até mesmo empresas de pequeno porte necessitam de computadores para realizar suas atividades diárias, conseqüentemente para melhor segurança e desempenho no trabalho estes computadores precisam ser gerenciados por servidores. Este trabalho é uma análise comparativa entre dois servidores de baixo custo voltado para pequenas empresas, instalando serviços básicos em dois hardwares: Servidor Cubietruck e servidor desktop, com o objetivo de verificar qual o melhor custo/benefício para essas empresas, testando o desempenho principalmente no servidor Cubietruck, podendo assim as empresas usufruir deste recurso, em que muitas ainda não possuem. Estes servidores disponibilizarão serviços de Internet, intranet, banco de dados, backup e segurança. A análise verificará qual o melhor desempenho em hardware, software e consumo de energia.

Palavras-chave: Servidor Cubietruck. Servidor desktop. Desempenho em hardware. Consumo de energia. Custo/benefício.

ABSTRACT

FINSTERBUSCH, Anderson. Use Feasibility Study Cubietruck as Small Business Server. 2015. 74 f. Monografia de Trabalho de Conclusão de Curso (II Curso de Especialização em Redes de Computadores), Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

Today, the computerization reaches the most diverse companies, and even small businesses require computers to perform their daily activities accordingly for better security and performance at work these computers need to be managed by servers. This work is a comparative analysis between two low cost facing small business servers, installing basic services in two hardware: Cubietruck server and desktop server, in order to find what the best cost / benefit for these companies, testing the performance especially in Cubietruck server, so businesses can make use of this feature, in which many still do not have. These servers make available Internet service, intranet, database, backup and security. The analysis will verify which the best performance in hardware, software and power consumption.

Key words: Cubietruck server. Desktop server. Performance hardware. Energy consumption. Cost/benefit.

LISTA DE SIGLAS

ARM	<i>Acorn Risc Machine</i> (Processador com conjunto reduzido de instruções)
CD	<i>Compact Disc</i> (Disco compacto)
CPU	<i>Central Processing Unit</i> (Unidade central de processamento)
DHCP	<i>Dynamic Host Configuration Protocol</i> (Protocolo de configuração dinâmica de host)
DVD	<i>Digital Versatile Disc</i> (Disco digital versátil)
FTP	<i>File Transfer Protocols</i> (Protocolo de transferência de arquivos)
GPU	<i>Graphics Processing Unit</i> (Unidade de processamento gráfico)
HD	<i>Hard Disk</i> (Disco rígido)
HDMI	<i>High-Definition Multimedia Interface</i> (Interface multimídia de alta definição)
HTML	<i>HyperText Markup Language</i> (Linguagem de marcação de hipertexto)
I/O	<i>Input/Output</i> (Entrada/saída)
NTP	<i>Network Time Protocol</i> (Protocolo de sincronização de relógio na rede)
PHP	<i>Personal Home Page</i> (Linguagem de programação de páginas para Internet)
SDCARD	<i>Secure Digital Card</i> (Cartão de memória)
SPDIF	<i>Sony/Philips Digital Interface Format</i> (Conector de áudio)
SSH	<i>Secure Shell</i> (Conexão segura com outro computador)
SSL	<i>Secure Sockets Layer</i> (Protocolo de camada de segurança)
TLS	<i>Transport Layer Security</i> (Protocolo de segurança de telecomunicações)
UFW	<i>Uncomplicated Firewall</i> (Aplicativo de bloqueio de acessos na rede de computadores)
UTP	<i>Unshielded Twisted Pair</i> (Par trançado sem blindagem)
VGA	<i>Video Graphics Array</i> (Padrão de disposição gráfica para vídeo)

LISTA DE FIGURAS

Figura 1. Placa Cubietruck com case em acrílico.....	19
Figura 2. Placa Cubietruck com detalhamento dos componentes.....	19
Figura 3. Roteador/Switch Mikrotik RB2011UAS-RM visão horizontal.....	24
Figura 4. Multímetro Digital com Alicata Amperímetro Foxlux.....	25
Figura 5. Diagrama de rede para os testes.....	28
Figura 6. Fotografia rede de computadores utilizado para testes.....	29
Figura 7. Exemplo acesso SSH no servidor Cubietruck.....	30
Figura 8. Comando Nmap no servidor Cubietruck.....	31
Figura 9. Comando Nmap no servidor desktop.....	31
Figura 10. Sysbench - Processador - Servidor Cubietruck.....	32
Figura 11. Sysbench - Processador - Servidor desktop.....	33
Figura 12. Resultado teste Sysbench – Processador.....	33
Figura 13. Sysbench – Teste de memória - Servidor Cubietruck.....	34
Figura 14. Sysbench – Teste de memória - Servidor desktop.....	35
Figura 15. Resultado Sysbench – Teste de memória.....	35
Figura 16. Sysbench – Gravação de arquivos - Servidor Cubietruck.....	36
Figura 17. Sysbench – Gravação de arquivos - Servidor desktop.....	37
Figura 18. Sysbench – Leitura de arquivos - Servidor Cubietruck.....	37
Figura 19. Sysbench – Leitura de arquivos - Servidor desktop.....	38
Figura 20. Resultado teste Sysbench – Gravação e leitura de arquivos em disco....	39
Figura 21. Sysbench – Teste banco de dados – Servidor Cubietruck.....	40
Figura 22. Sysbench – Teste banco de dados – Servidor desktop.....	40
Figura 23. Resultado teste Sysbench - Banco de dados MySQL (Transações/segundo).....	41
Figura 24. Iperf – Desempenho da rede com 06 clientes - Servidor Cubietruck.....	42
Figura 25. Iperf – Desempenho da rede com 06 clientes - Servidor desktop.....	42
Figura 26. Resultado teste Iperf – Desempenho da rede com 6 clientes.....	43
Figura 27. Iperf – Desempenho da Rede com 06 clientes - Servidor Cubietruck.....	43
Figura 28. Iperf – Desempenho da Rede com 06 clientes - Servidor desktop.....	44
Figura 29. Resultado teste Iperf – Desempenho da rede ponto a ponto.....	44
Figura 30. Rsync – Transferência de arquivos – Cliente x Servidor - Cubietruck.....	45
Figura 31. Rsync – Transferência de arquivos – Cliente x Servidor - Desktop.....	46
Figura 32. Resultado teste Rsync – Transferência de arquivos cliente x servidor....	46
Figura 33. Rsync – Transferência de arquivos – Servidor x Cliente - Cubietruck.....	47
Figura 34. Rsync – Transferência de arquivos – Servidor x Cliente – Desktop.....	47
Figura 35. Resultado teste Rsync – Transferência de arquivos - Servidor x Cliente.	48
Figura 36. Wireshark – Desempenho <i>proxy</i> – sem <i>proxy</i> – Servidor Cubietruck.....	49
Figura 37. Wireshark – Desempenho <i>proxy</i> – com <i>proxy</i> – Servidor Cubietruck.....	50
Figura 38. Wireshark – Desempenho <i>proxy</i> – Servidor Cubietruck.....	50
Figura 39. Consumo de energia – Operação normal.....	51
Figura 40. Consumo de energia – Processamento máximo.....	52

LISTA DE TABELAS

Tabela 1. Consumo de energia – Cálculo em watts – Servidor desktop.....	52
Tabela 2. Consumo de energia – Cálculo em reais – Servidor desktop.....	53
Tabela 3. Consumo de energia – Cálculo em reais – Servidor Cubietruck.....	53
Tabela 4. Custo para aquisição dos servidores.....	54

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 CONSIDERAÇÕES INICIAIS.....	14
1.2 OBJETIVOS.....	15
1.2.1 Objetivo Geral.....	15
1.2.2 Objetivos Específicos.....	15
1.3 JUSTIFICATIVA.....	15
1.4 ESTRUTURA DO TRABALHO.....	16
2 REFERENCIAL TEÓRICO.....	17
2.1 CUBIETRUCK.....	17
2.2 SISTEMA OPERACIONAL LINUX DEBIAN.....	20
2.3 SERVIÇOS INSTALADOS.....	20
2.3.1 SSH – Acesso Remoto.....	21
2.3.2 NTP – Relógio do sistema.....	21
2.3.3 DHCP.....	21
2.3.4 Proxy – Squid.....	21
2.3.5 Banco de dados – MySQL.....	22
2.3.6 Compartilhamento de arquivos – Samba.....	22
2.3.7 Firewall – UFW.....	22
3 MATERIAIS E MÉTODO.....	23
3.1 MATERIAIS.....	23
3.1.1 – Switch.....	24
3.1.2 – Multímetro alicate amperímetro.....	24
3.1.3 – Software Sysbench.....	25
3.1.4 – Software Rsync.....	26
3.1.5 – Software Iperf.....	26
3.1.6 – Software Wireshark.....	26
3.2 Método.....	27
4 RESULTADOS E DISCUSSÃO.....	28
4.1 CENÁRIO DOS TESTES.....	28
4.2 ACESSO REMOTO VIA SSH.....	29
4.3 SERVIÇO DHCP.....	30
4.4 SYSBENCH.....	32
4.4.1 Sysbench – Teste de Processador.....	32
4.4.2 Sysbench - Teste de Memória.....	34
4.4.3 Sysbench - Teste de gravação e leitura de arquivos.....	36
4.4.4 Sysbench - Teste de Banco de dados MySQL (Transações por segundo).....	39
4.5 IPERF.....	41
4.5.1 Iperf – Desempenho da Rede – 06 clientes.....	41
4.5.2 Iperf – Desempenho da Rede – Conexão ponto a ponto.....	43
4.6 RSYNC.....	45
4.6.1 Rsync – Transferência de arquivos – Cliente X Servidor.....	45
4.6.2 Rsync – Transferência de arquivos – Servidor X Cliente.....	47
4.7 WIRESHARK.....	48
4.7.1 Wireshark – Desempenho <i>proxy</i> – Servidor Cubietruck.....	48
4.8 CONSUMO DE ENERGIA.....	51
4.8.1 Consumo de energia – Servidor desktop.....	51
4.9 CUSTO DE EQUIPAMENTOS.....	54
5 CONCLUSÃO.....	55
REFERÊNCIAS.....	56

ANEXOS.....	58
Anexo 1 – Instalação sistema operacional Debian no dispositivo Cubietruck.....	58
Anexo 2 – Instalação sistema operacional Debian no computador desktop.....	58
Anexo 3 – Instalação e configuração de serviços: SSH – Acesso Remoto.....	59
Anexo 4 – Instalação e configuração de serviços: NTP – Relógio do sistema.....	60
Anexo 5 – Instalação e configuração de serviços: DHCP.....	62
Anexo 6 – Instalação e configuração de serviços: Proxy - Squid.....	64
Anexo 7 – Instalação e configuração de serviços: Banco de dados - MySQL.....	67
Anexo 8 – Instalação e configuração de serviços: Compartilhamento de arquivos – Samba.....	68
Anexo 9 – Instalação e configuração de serviços: UFW - Firewall.....	73

1 INTRODUÇÃO

Neste capítulo são apresentados as considerações iniciais, e as características e necessidades das micro e pequenas empresas, bem como levantados alguns benefícios da utilização de um servidor dedicado de baixo custo.

1.1 CONSIDERAÇÕES INICIAIS

Atualmente diversas microempresas e empresas de pequeno porte com no máximo 10 computadores, trabalham utilizando um computador desktop de um usuário como sendo o servidor, onde fica instalado o banco de dados de seus sistemas e seus aplicativos. Este mesmo terminal não fica restrito apenas para esta função, já que o usuário utiliza-o para as demais funções da empresa, como emissão de notas fiscais, gerenciamento do sistema de controle de compra, venda, financeiro, além de utilizar a Internet para acesso a bancos e e-mails. Com isso, pode ocorrer falhas na segurança dos dados.

Essas empresas trabalham ainda desta maneira, pois tem a imagem de um servidor como uma grande sala onde fica vários computadores e imaginam que o custo para implantar será muito elevado. Mas os tempos mudaram, e os servidores já não são grandes máquinas e o custo já está bem acessível para esse porte de empresas. Hoje um servidor pode ser montado a partir de um desktop padrão, ou em um dispositivo microcomputador, que tem baixo custo e consumo reduzido, além do pequeno espaço ocupado.

Como no mercado já existem equipamentos de baixo custo, é possível instalar os serviços essenciais, montando um multiservidor que atenda essas empresas. No decorrer deste trabalho, é apresentado a instalação destes serviços em um desktop padrão e em um dispositivo microcomputador, analisando seu custo/benefício.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Elaborar um projeto para um multiservidor de baixo custo que atenda as necessidades relacionadas a serviços de rede para micro e pequenas empresas, e compará-lo com um servidor desktop de hardware equivalente.

1.2.2 Objetivos Específicos

- Instalação do sistema operacional e testes de funcionamento;
- Instalação dos serviços de rede;
- Execução de testes de desempenho com os serviços instalados;
- Análise do consumo de energia;
- Coleta e análise dos dados gerados, elegendo o melhor custo/benefício.

1.3 JUSTIFICATIVA

Com base na experiência profissional como analista de implantação de sistemas, muitas pequenas empresas não possuem um servidor dedicado em seus estabelecimentos, em que estão sujeitas a diversos problemas, como: segurança na rede, acesso indevido a Internet, falta de segurança com banco de dados, entre outros.

Com um servidor dedicado, essas empresas terão benefícios, como:

- Centralizar dados no servidor, compartilhando arquivos, impressoras, o sistema gerenciador da empresa, banco de dados. Com isso minimiza a sobrecarga do computador terminal, liberando mais recursos da máquina e assim a empresa pode manter computadores antigos trabalhando normalmente;

- Segurança na rede, com a implantação de *firewall*, *proxy*, bloqueando acessos externos e controlando o acesso da rede na empresa;
- Compartilhar arquivos pela Internet, implantando um servidor FTP, as empresas podem compartilhar arquivos seguros através da rede com seus funcionários que estão fora da empresa;
- Política de backups, implantando uma ferramenta dedicada gerenciando os backups da empresa automaticamente.

Como essas empresas possuem pouco recursos para investirem em servidores dedicados, esse trabalho surgiu como uma forma de auxiliar essas empresas a utilizarem servidores com baixo custo, mostrando uma análise comparativa dos custos destes servidores e seus respectivos desempenhos.

Com o intuito voltado para o baixo custo, esse trabalho analisará se o servidor Cubietruck atende as necessidades de uma empresa de pequeno porte, comparando seu desempenho e consumo energético com o de um servidor desktop com capacidade de processamento semelhante.

Com este trabalho pretende-se analisar o melhor custo/benefício para esse porte de empresas, permitindo maior controle dos recursos de rede, com redução de gastos com equipamentos e energia elétrica.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado como segue:

- No Capítulo 2 descreve-se o referencial teórico, apresentando o dispositivo Cubietruck, juntamente o sistema operacional e os serviços instalados para realizar a análise;
- No Capítulo 3 apresenta-se os materiais e método utilizados para a execução deste trabalho;
- No Capítulo 4 demonstra-se o resultado obtido com a análise comparativa, descrevendo qual servidor tem o melhor custo/benefício;
- Por fim no Capítulo 5 são apresentados as conclusões do trabalho.

2 REFERENCIAL TEÓRICO

Neste capítulo é apresentado o referencial teórico utilizado para a análise e comparação do servidor Cubietruck com o servidor desktop.

2.1 CUBIETRUCK

Cubietruck (CUBIEBOARD, 2015) também conhecido como Cubieboard 3, é um dispositivo embarcado, ou seja, um microcomputador em uma placa única, que suporta vários sistemas Linux.

Atualmente estão sendo desenvolvidos vários projetos utilizando esse dispositivo, sendo um deles interligar a placa Cubietruck a um televisor com entrada HDMI, para se obter acesso à Internet através do televisor. Ou também transformá-lo num roteador sem fio, para compartilhar Internet em uma casa ou empresa.

No mercado existem outros dispositivos embarcados, como o Raspberry, Panda board, Beagle bone, Sharks Cove, Arduíno e também outras versões do Cubietruck, o Cubieboard, o Cubieboard2, e o Cubieboard4.

O Cubieboard foi lançado em 2012 pela empresa CubieTeam, com as seguintes características:

- Processador: ARM cortex-A8, NEON, VFPv3, 256KB L2 *cache*;
- Vídeo: Mali400, OpenGL ES GPU;
- Memória: 1GB DDR3 @480MHz;
- Conexão vídeo: HDMI 1080p *Output*;
- Rede: 10/100M *Ethernet*;
- Memória interna: 4Gb *Nand Flash*;
- Conexões: 2 USB Host, 1 micro SD slot, 1 SATA, 1 ir;
- 96 Pinos Extensão: I2C, SPI, RGB/LVDS, CSI/TS, FM-IN, ADC, CVBS, SPDIF-OUT, R-TP;
- Dimensões: 11cm comprimento x 8cm largura x 1.4cm altura;
- Compatível com Android, Ubuntu, Debian, ArchLinux, OpenELEC, Fedora entre outras distribuições Linux.

Em 2013 foi lançado o Cubieboard 2, em que foi inserido um novo processador, modelo ARM Cortex-A7 *Dual-Core*, 1MB L2 *cache*.

Já o Cubietruck ou Cubieboard3, foi lançado no final de 2013, possuindo o mesmo processador do Cubieboard2, tendo como diferencial o dobro de memória flash com 8GB, e o dobro de memória RAM 2GB. Além disso possui saída VGA, conector SPDIF (Áudio), e comunicação sem fio integrado na placa com *Wireless* e *Bluetooth*. Abaixo os detalhes da configuração:

- Processador: ARM® Cortex™-A7 *Dual-Core*;
- Vídeo: ARM® Mali400 MP2 *Complies with OpenGL ES 2.0/1.1*;
- Memória: 1GB/2GB DDR3@480MHz;
- Conexão: HDMI e VGA 1080P;
- Placa rede: RTL8211E 10M/100M/1G *Ethernet*;
- *Wireless*: Wifi+BT BCM AP6210;
- Conexão SATA 2.0 suporte 2.5' HDD, (para 3.5' HDD, necessário fonte energia 12V)
- Armazenamento: 8GB NAND memória *flash* + slot MicroSD;
- Conexões: 2x USB, 1x OTG, 1x SPDIF, 1x IR, 1 *Headphone*;
- Alimentação: 5V – 2.5A;
- 54 pinos de extensão: I2S, I2C, SPI, CVBS, LRADC x2,UART, PS2, PWMx2, TS/CSI, IRDA, LINEIN&FMIN&MICIN, TVINx4;
- Pré-instalado sistema operacional: Android 4.2.2.

Na Figura 1 é mostrado a placa Cubietruck instalada no case em acrílico. Já na Figura 2 é mostrado a placa Cubietruck com o detalhamento dos componentes.

A versão atual do Cubieboard é o Cubieboard4, com diferencial de possuir o processador *Octa-Core* A15/A7 com GPU de 64 núcleos, *wifi dual band* e conexão de entrada para microfone. O restante segue as mesmas configurações da Cubietruck.

Para este trabalho, será utilizado o Cubietruck, em que será instalado o sistema operacional Linux Debian 7.8, com os seguintes serviços: NTP, SSH, DHCP, *proxy* autenticado, certificados SSL, banco de dados, compartilhamento de arquivos, *firewall*.

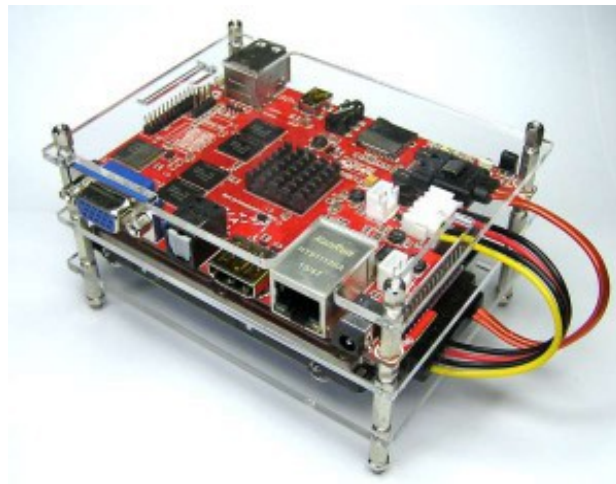


Figura 1. Placa Cubietruck com case em acrílico

Fonte: CUBIETRUCK, 2015.

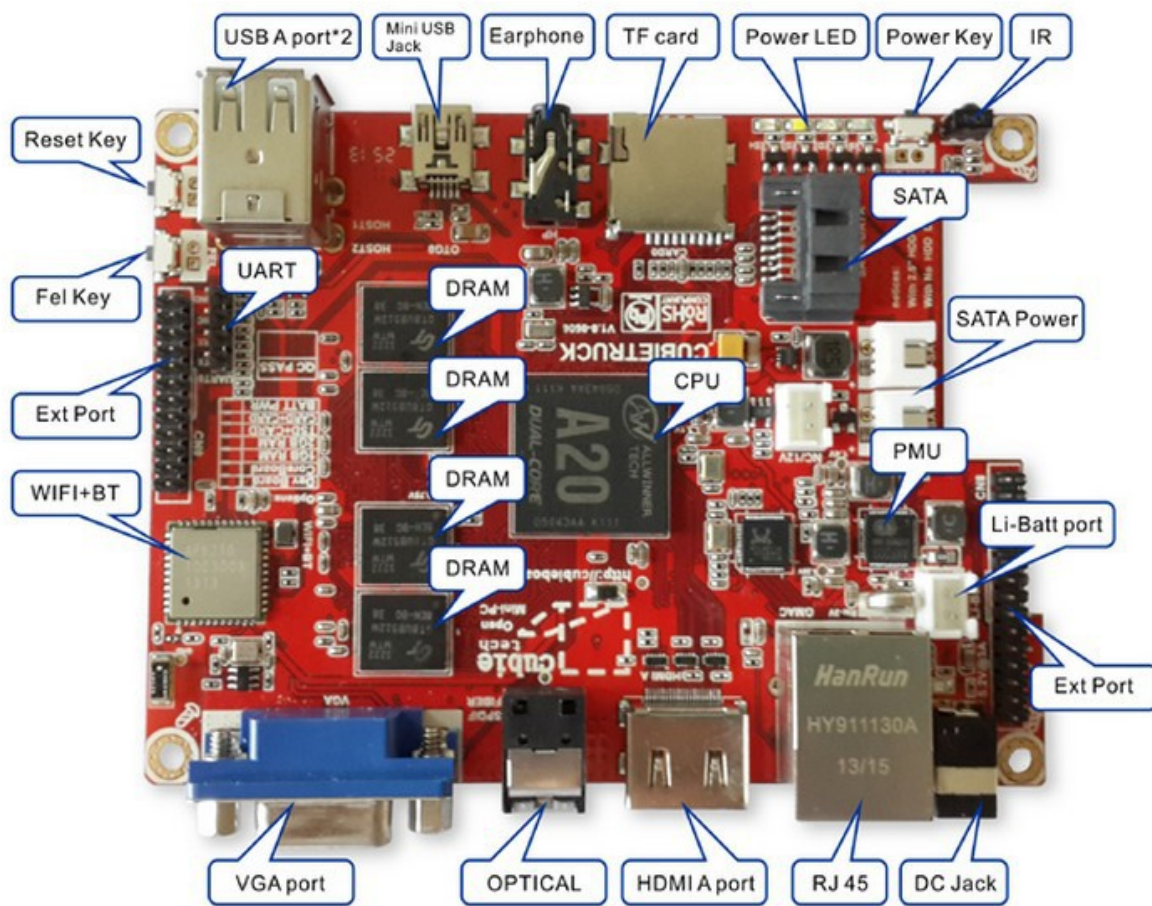


Figura 2. Placa Cubietruck com detalhamento dos componentes

Fonte: CUBIETRUCK, 2015.

2.2 SISTEMA OPERACIONAL LINUX DEBIAN

Lançado em 16 de agosto de 1993 por Ian Murdock, Debian é um sistema operacional Linux livre, ou seja, gratuito, mantido por uma associação de indivíduos ao redor do mundo, através do projeto GNU/LINUX e por doações de organizações sem fins lucrativos. Utilizado tanto para desktop como para servidor com suporte a várias arquiteturas, como: Amd64, arm64, armel, armhf, i386, mips, mipsel, powerpc, ppc64el e s390x. Diversas distribuições Linux surgiram e surgem através do Debian, entre elas Ubuntu, Mint, Knoppix, Kurumin, Lubuntu.

A atual versão do Debian é a 8 denominada “*Jessie*”, porém para esse trabalho, será utilizado a versão 7.8 denominada “*Wheezy*”.

No Cubietruck a versão do Debian instalado é a 7.8 ARM debian-server SD CARD em modo VGA, em que foi customizada pela própria Cubieboard para ser instalado em cartão de memória com suporte a vídeo VGA.

No servidor desktop a versão do Debian instalado é a 7.8 i386-netinst.iso, que é uma versão mínima, com recursos básicos.

Nas duas versões o sistema executa sem ambiente gráfico, para poder aproveitar melhor os recursos de hardware de cada servidor. Anexo a este trabalho encontra-se os procedimentos de instalação do sistema operacional Debian em cada equipamento.

2.3 SERVIÇOS INSTALADOS

Para realizar a comparação dos servidores, foram instalados nos dois servidores alguns serviços de administração, armazenamento, compartilhamento e segurança em uma rede de computadores. Este trabalho não tem como foco detalhar o funcionamento de cada serviço, e sim verificar se ao executar estes serviços, os servidores comportam a carga de trabalho. Abaixo apresenta-se os serviços instalados, e em anexo a este trabalho, encontra-se os procedimentos de instalação de cada serviço.

2.3.1 SSH – Acesso Remoto

Para poder administrar o servidor, será instalado o serviço SSH, que permite acessar remotamente o servidor via terminal por linha de comando. O protocolo SSH é um protocolo de comunicação que criptografa os dados entre o cliente e o servidor, tornando segura sua comunicação. Para o servidor remoto, foi utilizado o aplicativo *Openssh* versão 6.0, que é uma versão gratuita do protocolo SSH suportando as versões 1.3, 1.5 e 2 do protocolo. O *Openssh* criptografa as comunicações inclusive senhas, aumentando a segurança na rede.

2.3.2 NTP – Relógio do sistema

Em uma empresa que utiliza notas fiscais eletrônicas, é indispensável que o relógio do sistema esteja sempre sincronizado, para evitar transmissões incorretas de arquivos para a receita estadual. Por esse motivo, é instalado o serviço NTP no servidor, que mantém o relógio do sistema sincronizado com outros servidores via Internet. A versão do NTP utilizada foi a 4.2.6.

2.3.3 DHCP

O DHCP é um protocolo de configuração dinâmica de rede, possibilitando que todos os dispositivos de uma rede, adquiram suas configurações automaticamente pelo servidor, em vez de configurar manualmente em cada dispositivo. Para este trabalho foi utilizado o DHCP ISC *server* versão 4.2.2.

2.3.4 Proxy – Squid

Um servidor *proxy* tem como função melhorar o desempenho de acesso das páginas de Internet, por meio de um *cache* de páginas, em que armazena as páginas acessadas e quando solicitadas novamente, estas páginas serão automaticamente processadas. Outra função é bloquear o acesso em determinadas

páginas, evitando acessos indevidos pelos funcionários em sites de jogos, pornográficos e redes sociais em horário de trabalho, onde nesses sites existe maior incidência de vírus. Para este trabalho foi utilizado o software Squid versão 3.4.8, que além de ser gratuito é de fácil instalação/configuração.

2.3.5 Banco de dados – MySQL

Hoje em dia toda empresa precisa ter um sistema de automação, em que estes necessitam de um gerenciador de banco de dados. Um gerenciador popularmente conhecido é o MySQL, que é desenvolvido em código aberto, utiliza poucos recursos de memória, tem um bom desempenho e é estável de fácil administração. Para este trabalho foi utilizado a versão 5.5.43 do MySQL.

2.3.6 Compartilhamento de arquivos – Samba

Uma das funções de um servidor é possuir a viabilidade de armazenar arquivos, permitindo o seu compartilhamento e acesso de modo seguro. Como a maioria dos sistemas operacionais desktops das empresas são da plataforma Windows e os servidores neste trabalho são Linux, necessitam de um software para realizar esse compartilhamento, neste caso o software Samba. O Samba é um software livre que compartilha recursos de impressoras e arquivos, seguro, estável e rápido para clientes Windows e Linux. Neste trabalho foi utilizado a versão 3.6.6 do Samba.

2.3.7 Firewall – UFW

Para proteger uma rede contra invasões externas e acessos não autorizados, se faz necessário implementar um *firewall*, que realiza esses bloqueios filtrando e impedindo os acessos. No Linux, a ferramenta *iptables* é utilizado para essas regras, mas para esse trabalho foi utilizado a ferramenta UFW versão 0.33-2, uma versão simplificada do *iptables*, de fácil utilização e gratuito.

3 MATERIAIS E MÉTODO

Neste capítulo é apresentado como é realizado a instalação do sistema operacional Linux Debian e os serviços mencionados no capítulo anterior, nos dois servidores.

3.1 MATERIAIS

Os materiais utilizados para a realização deste trabalho foram:

- Dispositivo Cubietruck (Servidor);
- Computador desktop (marca: Positivo, processador: Pentium dual-core E5300 2.6Ghz 64 Bits, memória: DDR2 2GB, disco rígido: HD SATA 160 GB, placa de rede Atheros Attansic L1 1Gbit/s, fonte de alimentação: Seasonic SFX SS-300SFE 300W);
- Sistema operacional Linux Debian 7.8 versão ARM debian-server para o Cubietruck e versão i386-netinst para o computador desktop;
- Dispositivo Cubietruck (06 estações com sistema operacional Linux Debian 7 versão ARM);
- Notebook HP (Modelo: dv-5 2115br AMD Turion(tm) II P520 *Dual-Core*, com sistema operacional *dual boot*: Linux Ubuntu 14.04 e Windows 7 Home Premium);
- Roteador/Switch (marca: Mikrotik, modelo: RB2011UAS-RS, portas: 5 *Fast Ethernet + 5 Gigabit Ethernet*);
- Multímetro alicate amperímetro (marca: Foxlux, modelo: FX-AA);
- Software Sysbench (Testes: CPU/Memória/Disco/Banco de dados);
- Software Rsync (Testes: transferência de arquivos);
- Software Iperf (Testes: desempenho da rede);
- Software Wireshark (Testes: desempenho do proxy).

3.1.1 – Switch

O *Switch* tem como função ligar vários dispositivos de uma rede, estabelecendo uma ligação direta entre o dispositivo de origem com o de destino. Neste trabalho foi utilizado um roteador Mikrotik modelo RB2011UAS-RS, mostrado na Figura 3, com 05 portas *Fast Ethernet* e 05 *Gigabit Ethernet*. Porém para os testes, as portas *gigabits* foram configuradas para trabalhar em *Fast Ethernet*, para poder comparar o desempenho igualmente com todas as portas. Também foi configurado o roteador apenas para trabalhar no modo *switch*.



Figura 3. Roteador/Switch Mikrotik RB2011UAS-RM visão horizontal

Fonte: MIKROTIK, 2015.

3.1.2 – Multímetro alicate amperímetro

Multímetro é um instrumento de medição elétrica, capaz de realizar a medição da corrente elétrica na função amperímetro, tensão elétrica na função voltímetro e resistência elétrica na função ohmímetro. Dependendo do modelo além das funções citadas, podem também realizar a medição de capacitância e frequência de sinais alternados.

Neste trabalho foi utilizado o multímetro digital Foxlux modelo FX-AA, mostrado na Figura 4, com alicate amperímetro, onde será medido a corrente elétrica na função amperímetro, e a tensão no momento da medição com o voltímetro (FOX LUX, 2015).



Figura 4. Multímetro Digital com Alicates Amperímetro Foxlux

Fonte: FOXLUX, 2015.

3.1.3 – Software Sysbench

Sysbench projeto mantido por Alexey Kopytov, é um aplicativo multitarefa de software livre para ambientes Linux, que permite avaliar o desempenho do hardware testando a performance dos seguintes itens: processador, memória, arquivos de leitura e gravação no disco e desempenho do gerenciador do banco de dados. (SYSBENCH, 2015):

- **Processador:** testa o desempenho de CPU calculando os números primos até um valor máximo especificado pelo usuário, no teste é executado para determinar quanto tempo leva para que o sistema calcule estes números.
- **Memória:** neste teste a memória é lida ou escrita de forma sequencial onde é especificado pelo usuário o tamanho total do arquivo utilizado no teste, exibindo o tempo que utiliza para executar esta tarefa e a velocidade de transferência.
- **I/O de arquivos:** realiza vários testes de operações de leitura e gravação de arquivos sendo elas sequenciais ou randômicas.
- **Banco de dados:** executa operações básicas para medir a performance do gerenciador de banco de dados MySQL, com as seguintes operações: (criação de tabelas, remoção de tabelas, inserção de dados e outras).

3.1.4 – Software Rsync

Rsync desenvolvido por Andrew Tridgell, o mesmo autor do SAMBA, é um aplicativo de código aberto que permite cópia/transferência rápida de arquivo, sendo um grande aliado para realizar backups. O software herdou as propriedades de criptografia do protocolo SSH, tornando sua transmissão de dados mais segura que o FTP. Permite sincronizar o conteúdo de duas pastas, transferindo apenas as modificações. Pode ser utilizado para realizar cópias de arquivos entre computadores na rede. O *rsync* roda em sistemas Linux, porém é possível realizar cópias de arquivos Windows na rede, informando o seu caminho. (RSYNC, 2015).

3.1.5 – Software Iperf

O Iperf desenvolvido pelo *National Laboratory for Applied Network Research* (NLNR), é um software livre que permite medir o rendimento da largura de banda de uma rede de computadores, onde pode ser identificado o mau desempenho de uma rede, até mesmo a eficiência de uma porta de um *switch*. Também permite simular conexões simultâneas tanto para o cliente quanto para o servidor. O aplicativo roda em diversas plataformas, entre elas Linux, Windows e MacOS X, onde normalmente realiza-se os testes entre: Linux-Linux, Linux-Windows e Windows-Windows (IPERF, 2015).

3.1.6 – Software Wireshark

Wireshark é um aplicativo para análise de pacotes da rede, desenvolvido por Gerald Combs, onde faz a captura de pacotes na rede e mostra de forma detalhada através de diversos filtros disponíveis na ferramenta. Software livre disponível em diversas plataformas (Linux, Windows, Mac, BSD) (WIRESHARK, 2015). Para este trabalho será utilizado a ferramenta para obter o desempenho do *proxy* na rede, comparando a velocidade que proporcionou o acesso a um site especificado.

3.2 MÉTODO

O método apresenta as etapas e procedimentos seguidos para o desenvolvimento deste trabalho. Os métodos adotados são descritos a seguir:

A primeira etapa para o desenvolvimento deste trabalho foi o levantamento bibliográfico sobre o dispositivo Cubietruck. Porém as referências sobre este dispositivo encontra-se apenas no site do fabricante, onde foi possível coletar todas as informações necessárias.

A segunda etapa foi a instalação e configuração do sistema operacional Debian no dispositivo Cubietruck e no servidor desktop.

Na terceira etapa foi realizado a montagem do cenário de testes, montando e configurando uma rede de computadores em laboratório, com os servidores, o *switch* e as estações.

Na quarta e última etapa foi realizado as instalações dos aplicativos utilizados para os testes, executando e coletando os resultados obtidos.

4 RESULTADOS E DISCUSSÃO

4.1 CENÁRIO DOS TESTES

Para simular uma rede de computadores se baseado no porte das pequenas empresas, foi utilizado nos testes um dispositivo Cubietruck e um computador *desktop* sendo os servidores de comparação. Juntamente foi interligado seis dispositivos Cubietrucks e um notebook, simulando os *hosts* da rede através do roteador/switch Mikrotik RB2011 UAS-RM, conectados por cabos UTP categoria 5. Na Figura 5 mostra-se a imagem do diagrama de rede e na Figura 6, mostra-se a imagem da rede montada utilizado para os testes.

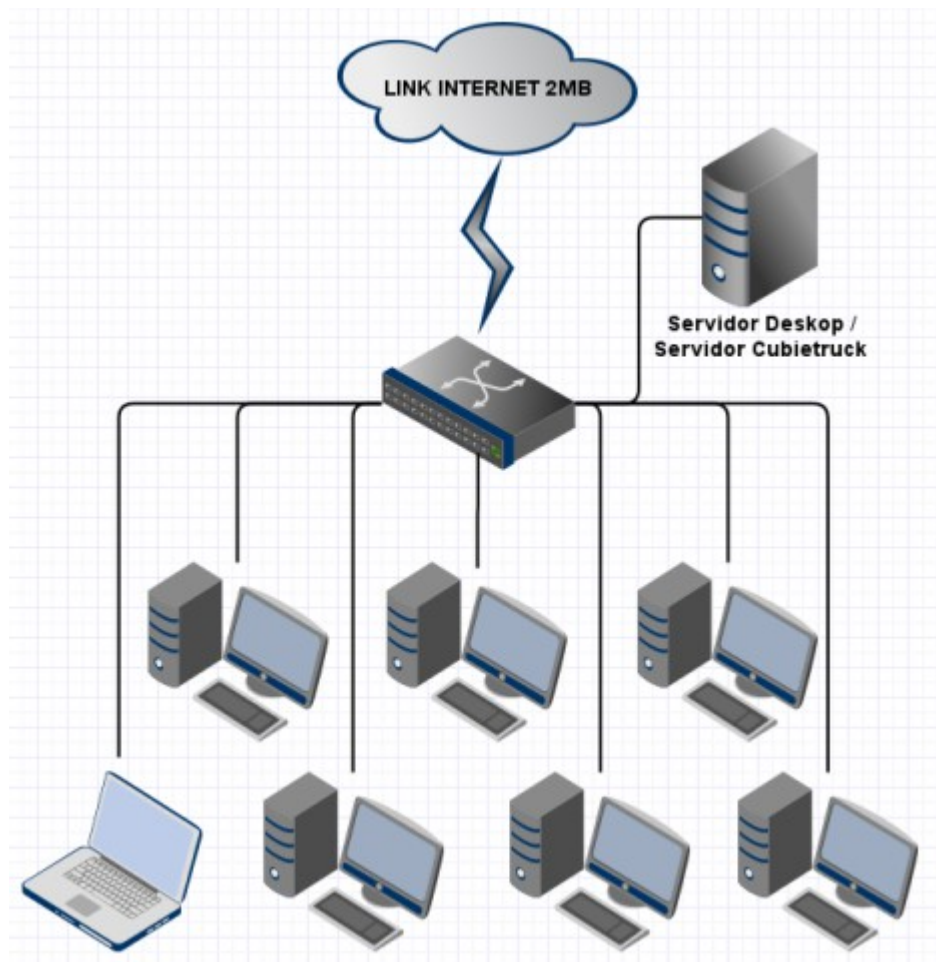


Figura 5. Diagrama de rede para os testes

Fonte: Autoria própria.

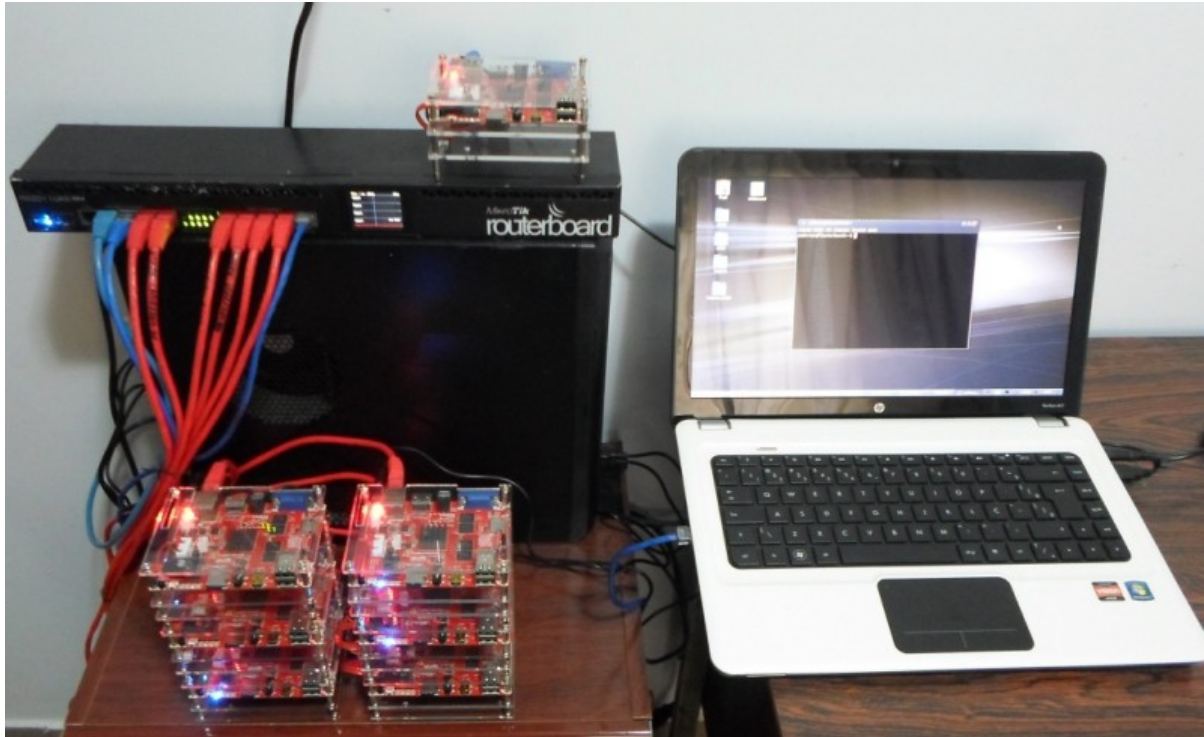
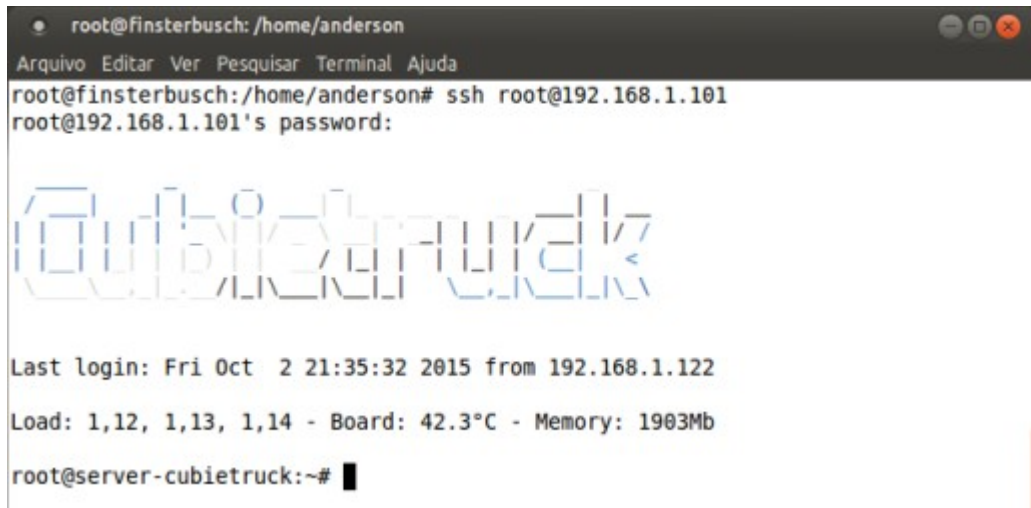


Figura 6. Fotografia rede de computadores utilizado para testes

Fonte: Autoria própria.

4.2 ACESSO REMOTO VIA SSH

Para acessar remotamente os servidores foi utilizado o serviço SSH, conectando via terminal o notebook com cada servidor. No servidor Cubietruck, foi inserido o seguinte comando: `# ssh root@192.168.1.101`, inserindo a senha do usuário `root`, exemplo na Figura 7. Já no servidor desktop, o comando foi o seguinte: `# ssh root@192.168.1.201`, também inserindo a senha do usuário `root`. Em ambos os servidores o serviço funcionou corretamente. Nos anexos a este trabalho encontra-se os procedimentos para instalação e configuração do serviço SSH.

A terminal window titled 'root@finsterbusch: /home/anderson' showing an SSH session. The user enters 'ssh root@192.168.1.101' and provides the password. The terminal displays a large ASCII art logo for 'Cubietruck' in blue. Below the logo, it shows the last login time as 'Fri Oct 2 21:35:32 2015 from 192.168.1.122' and system statistics: 'Load: 1,12, 1,13, 1,14 - Board: 42.3°C - Memory: 1903Mb'. The prompt changes to 'root@server-cubietruck:~#'.

```
root@finsterbusch: /home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@finsterbusch:/home/anderson# ssh root@192.168.1.101
root@192.168.1.101's password:

Cubietruck

Last login: Fri Oct 2 21:35:32 2015 from 192.168.1.122
Load: 1,12, 1,13, 1,14 - Board: 42.3°C - Memory: 1903Mb
root@server-cubietruck:~#
```

Figura 7. Exemplo acesso SSH no servidor Cubietruck

Fonte: Autoria própria.

4.3 SERVIÇO DHCP

Em ambos servidores foi instalado o serviço DHCP, para distribuir endereços de rede para os computadores. Na configuração do DHCP no servidor Cubietruck, foi definido o endereço IP no servidor sendo o 192.168.1.101, e a faixa de distribuição dos endereços ficou entre o endereço 192.168.1.111 e 192.168.1.142. Para verificar se o servidor atribuiu os endereços IP corretamente dentro da faixa, foi executado o seguinte comando:

```
root@server-cubietruck:~# nmap -sP 192.168.1.0/24
```

Já na configuração do DHCP no servidor desktop, foi definido o endereço IP no servidor sendo o 192.168.1.201, e a faixa de distribuição dos endereços ficou entre o endereço 192.168.1.211 e 192.168.1.242. O comando para verificar se o servidor atribuiu os endereços IP corretamente dentro da faixa foi:

```
root@debian-server:~# nmap -sP 192.168.1.0/24
```

As Figuras 8 e 9, mostram os endereços de rede que cada computador recebeu do servidor Cubietruck e também do servidor desktop.

```

root@finsterbusch: /home/janderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@server-cubietruck:/etc# nmap -sP 192.168.1.0/24

Starting Nmap 6.47 ( http://nmap.org ) at 2015-09-27 16:58 BRT
Nmap scan report for 192.168.1.1
Host is up (0.0022s latency).
MAC Address: 00:21:80:25:8B:80 (AP Router Ind. Eletronica Ltda)
Nmap scan report for 192.168.1.116
Host is up (-0.10s latency).
MAC Address: 02:42:03:C1:40:4D (Unknown)
Nmap scan report for 192.168.1.117
Host is up (0.00036s latency).
MAC Address: 02:48:09:41:D1:51 (Unknown)
Nmap scan report for 192.168.1.118
Host is up (0.00044s latency).
MAC Address: 02:C8:08:01:E1:1B (Unknown)
Nmap scan report for 192.168.1.119
Host is up (-0.10s latency).
MAC Address: 02:02:03:C2:3D:E8 (Unknown)
Nmap scan report for 192.168.1.120
Host is up (0.00024s latency).
MAC Address: 02:82:03:02:93:8E (Unknown)
Nmap scan report for 192.168.1.121
Host is up (0.00018s latency).
MAC Address: 02:96:03:81:4E:FA (Unknown)
Nmap scan report for 192.168.1.122
Host is up (-0.10s latency).
MAC Address: D8:D3:85:3F:20:F1 (Hewlett-Packard Company)
Nmap scan report for server-cubietruck (192.168.1.101)
Host is up.
Nmap done: 256 IP addresses (9 hosts up) scanned in 1.97 seconds
root@server-cubietruck:/etc#

```

Figura 8. Comando Nmap no servidor Cubietruck

Fonte: Autoria própria.

```

root@finsterbusch: /
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@debian-server:~# nmap -sP 192.168.1.0/24

Starting Nmap 6.00 ( http://nmap.org ) at 2015-09-27 16:02 BRT
Nmap scan report for 192.168.1.1
Host is up (0.0023s latency).
MAC Address: 00:21:80:25:8B:80 (AP Router Ind. Eletronica Ltda)
Nmap scan report for debian-server (192.168.1.201)
Host is up.
Nmap scan report for 192.168.1.213
Host is up (0.0016s latency).
MAC Address: 02:02:03:C2:3D:E8 (Unknown)
Nmap scan report for 192.168.1.214
Host is up (0.0013s latency).
MAC Address: 02:42:03:C1:40:4D (Unknown)
Nmap scan report for 192.168.1.215
Host is up (0.0016s latency).
MAC Address: 02:82:03:02:93:8E (Unknown)
Nmap scan report for 192.168.1.216
Host is up (0.0013s latency).
MAC Address: 02:C8:08:01:E1:1B (Unknown)
Nmap scan report for 192.168.1.217
Host is up (0.00063s latency).
MAC Address: 02:96:03:81:4E:FA (Unknown)
Nmap scan report for 192.168.1.218
Host is up (0.00062s latency).
MAC Address: 02:48:09:41:D1:51 (Unknown)
Nmap scan report for 192.168.1.220
Host is up (0.00061s latency).
MAC Address: D8:D3:85:3F:20:F1 (Hewlett-Packard Company)
Nmap done: 256 IP addresses (9 hosts up) scanned in 7.98 seconds
root@debian-server:~#

```

Figura 9. Comando Nmap no servidor desktop

Fonte: Autoria própria.

4.4 SYSBENCH

Nos testes de processador, memória, leitura e gravação de arquivos em disco e banco de dados, será utilizado a ferramenta *Sysbench*, já descrito na seção 3.1.3. A ferramenta foi instalada nos servidores com o seguinte comando: `# apt-get install sysbench`.

4.4.1 Sysbench – Teste de Processador

Para comparar o desempenho do processador, o comando utilizado em cada servidor foi: `# sysbench --test=cpu --num-threads=2 run`. Neste teste, verifica-se qual o tempo total que o processador obteve para executar a operação. As Figuras 10 e 11 mostram o teste nos dois servidores.

```

root@server-cubietruck:~# sysbench --test=cpu --num-threads=2 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 2

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 10000

Test execution summary:
  total time:                142.1381s
  total number of events:    10000
  total time taken by event execution: 284.2491
  per-request statistics:
    min:                      27.66ms
    avg:                      28.42ms
    max:                      68.55ms
    approx. 95 percentile:    32.44ms

Threads fairness:
  events (avg/stddev):       5000.0000/120.00
  execution time (avg/stddev): 142.1245/0.01

root@server-cubietruck:~# █

```

Figura 10. Sysbench - Processador - Servidor Cubietruck

Fonte: Autoria própria.


```

root@debian-server:~# sysbench --test=cpu --num-threads=2 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 2

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 10000

Test execution summary:
total time: 5.4742s
total number of events: 10000
total time taken by event execution: 10.9428
per-request statistics:
  min: 1.09ms
  avg: 1.09ms
  max: 9.20ms
  approx. 95 percentile: 1.09ms

Threads fairness:
  events (avg/stddev): 5000.0000/14.00
  execution time (avg/stddev): 5.4714/0.00

root@debian-server:~# █

```

Figura 11. Sysbench - Processador - Servidor desktop

Fonte: Autoria própria.

O resultado mostra que o servidor Cubietruk obteve o tempo total de 142,1381 segundos para executar a tarefa, já o servidor desktop obteve o tempo total de 5,4742 segundos, mostrando um desempenho em processamento superior ao servidor Cubietruk, conforme ilustrado no gráfico apresentado na Figura 12.

Sysbench - Teste Processador

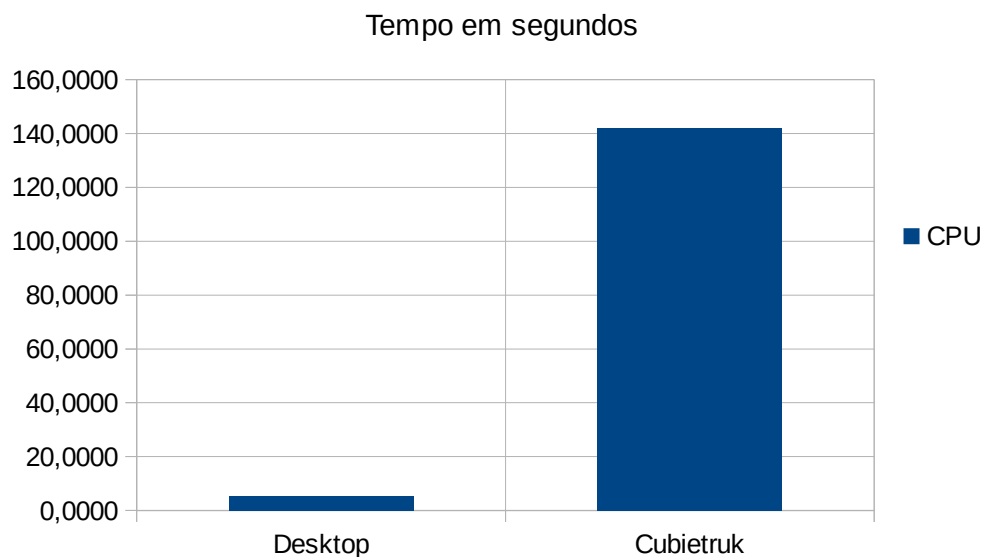


Figura 12. Resultado teste Sysbench – Processador

Fonte: Autoria própria.

4.4.2 Sysbench - Teste de Memória

Para comparar o desempenho da memória, o comando utilizado em cada servidor foi: `# sysbench --test=memory --memory-block-size=1M --memory-total-size=10G run`. Neste teste, verifica-se qual o tempo total que a memória obteve para executar a operação. As Figuras 13 e 14 mostram o teste nos dois servidores.

O resultado mostra que o servidor Cubietruck obteve o tempo total de 1,7874 segundos para executar a tarefa, já o servidor desktop obteve o tempo total de 0,2732 segundos, conforme ilustrado na Figura 15, mostrando como resultado o desempenho superior do servidor desktop comparado com o servidor Cubietruck.

```

root@finsterbusch: /home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@server-cubietruck:~# sysbench --test=memory --memory-block-size=1M --memory-total-size=10G run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Doing memory operations speed test
Memory block size: 1024K

Memory transfer size: 2048M

Memory operations type: write
Memory scope type: global
Threads started!
Done.

Operations performed: 2048 ( 1147.70 ops/sec)

2048.00 MB transferred (1147.70 MB/sec)

Test execution summary:
total time: 1.7844s
total number of events: 2048
total time taken by event execution: 1.7817
per-request statistics:
  min: 0.81ms
  avg: 0.87ms
  max: 10.03ms
  approx. 95 percentile: 0.88ms

Threads fairness:
  events (avg/stddev): 2048.0000/0.00
  execution time (avg/stddev): 1.7817/0.00

root@server-cubietruck:~# █

```

Figura 13. Sysbench – Teste de memória - Servidor Cubietruck

Fonte: Autoria própria.

```

root@finsterbusch: /home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@debian-server:~# sysbench --test=memory --memory-block-size=1M --memory-total-size=10G run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Doing memory operations speed test
Memory block size: 1024K

Memory transfer size: 2048M

Memory operations type: write
Memory scope type: global
Threads started!
Done.

Operations performed: 2048 ( 7496.88 ops/sec)

2048.00 MB transferred (7496.88 MB/sec)

Test execution summary:
total time: 0.2732s
total number of events: 2048
total time taken by event execution: 0.2722
per-request statistics:
  min: 0.13ms
  avg: 0.13ms
  max: 0.42ms
  approx. 95 percentile: 0.13ms

Threads fairness:
  events (avg/stddev): 2048.0000/0.00
  execution time (avg/stddev): 0.2722/0.00

root@debian-server:~# █

```

Figura 14. Sysbench – Teste de memória - Servidor desktop

Fonte: Autoria própria.

Sysbench - Teste Memória

Tempo em segundos

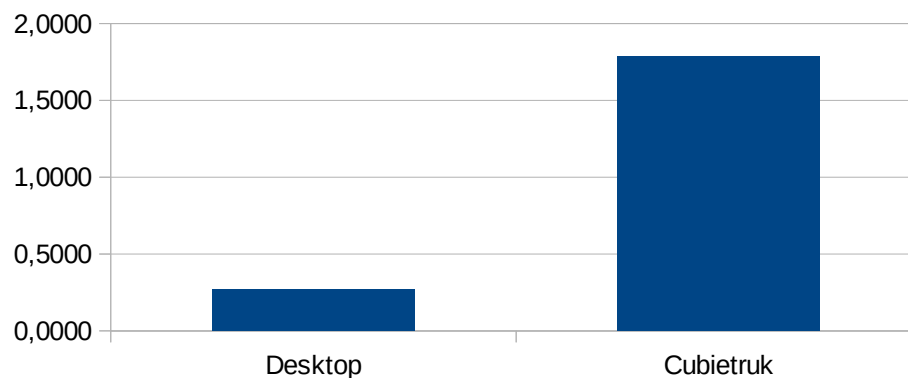


Figura 15. Resultado Sysbench – Teste de memória

Fonte: Autoria própria.

4.4.3 Sysbench - Teste de gravação e leitura de arquivos

Neste teste é analisado a taxa de transferência, verificando quantos *Megabits* por segundo cada servidor obtêm para realizar a gravação e leitura de um arquivo de 100 *megabytes* no disco de armazenamento, em um período de tempo de 240 segundos. Primeiro foi criado um arquivo de teste com o seguinte comando: `# sysbench --test=fileio --file-test-mode=rndwr --file-total-size=100MB prepare`. Após, foi realizado o teste de gravação de arquivo aleatório com o seguinte comando em cada servidor: `# sysbench --test=fileio --file-total-size=100MB --file-test-mode=rndrw --max-time=240 --max-requests=0 run`. Em seguida foi realizado o teste de leitura de arquivo aleatório com o seguinte comando: `# sysbench --test=fileio --file-total-size=100MB --file-test-mode=rndrd --max-time=240 --max-requests=0 run`. As Figuras 16, 17, 18 e 19 mostram estes testes nos dois servidores.

```

root@finsterbusch: /home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@server-cubietruck:~/Testes# sysbench --test=fileio --file-total-size=100MB --file
-test-mode=rndrw --max-time=240 --max-requests=0 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Extra file open flags: 0
128 files, 800Kb each
100Mb total file size
Block size 16Kb
Number of random requests for random IO: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!
Time limit exceeded, exiting...
Done.

Operations performed: 600 Read, 400 Write, 1232 Other = 2232 Total
Read 9.375Mb Written 6.25Mb Total transferred 15.625Mb (66.496Kb/sec)
4.16 Requests/sec executed

Test execution summary:
total time: 240.6170s
total number of events: 1000
total time taken by event execution: 2.0806
per-request statistics:
  min: 0.04ms
  avg: 2.08ms
  max: 1339.48ms
  approx. 95 percentile: 0.11ms

Threads fairness:
  events (avg/stddev): 1000.0000/0.00
  execution time (avg/stddev): 2.0806/0.00
root@server-cubietruck:~/Testes#

```

Figura 16. Sysbench – Gravação de arquivos - Servidor Cubietruck

Fonte: Autoria própria.

```

root@finsterbusch:/home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@debian-server:~/Testes# sysbench --test=fileio --file-total-size=100MB --file-test-mode=rndrw --max-time=240 --max-requests=0 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Extra file open flags: 0
128 files, 800Kb each
100Mb total file size
Block size 16Kb
Number of random requests for random IO: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!
Time limit exceeded, exiting...
Done.

Operations performed: 8580 Read, 5720 Write, 18188 Other = 32488 Total
Read 134.06Mb Written 89.375Mb Total transferred 223.44Mb (953.28Kb/sec)
59.58 Requests/sec executed

Test execution summary:
total time: 240.0146s
total number of events: 14300
total time taken by event execution: 0.2390
per-request statistics:
  min: 0.01ms
  avg: 0.02ms
  max: 1.26ms
  approx. 95 percentile: 0.02ms

Threads fairness:
  events (avg/stddev): 14300.0000/0.00
  execution time (avg/stddev): 0.2390/0.00

root@debian-server:~/Testes#

```

Figura 17. Sysbench – Gravação de arquivos - Servidor desktop

Fonte: Autoria própria.

```

root@finsterbusch:/home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@server-cubietruck:~/Testes# sysbench --test=fileio --file-total-size=100MB --file-test-mode=rndrd --max-time=240 --max-requests=0 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Extra file open flags: 0
128 files, 800Kb each
100Mb total file size
Block size 16Kb
Number of random requests for random IO: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Threads started!
Time limit exceeded, exiting...
Done.

Operations performed: 5344642 Read, 0 Write, 0 Other = 5344642 Total
Read 81.553Gb Written 0b Total transferred 81.553Gb (347.96Mb/sec)
22269.28 Requests/sec executed

Test execution summary:
total time: 240.0006s
total number of events: 5344642
total time taken by event execution: 221.4968
per-request statistics:
  min: 0.03ms
  avg: 0.04ms
  max: 14.91ms
  approx. 95 percentile: 0.04ms

Threads fairness:
  events (avg/stddev): 5344642.0000/0.00
  execution time (avg/stddev): 221.4968/0.00

root@server-cubietruck:~/Testes#

```

Figura 18. Sysbench – Leitura de arquivos - Servidor Cubietruck

Fonte: Autoria própria.


```

root@finsterbusch: /home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@debian-server:~/Testes# sysbench --test=fileio --file-total-size=100MB --file-test-mode=rndrd --max-time=240 --max-requests=0 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Extra file open flags: 0
128 files, 800Kb each
100Mb total file size
Block size 16Kb
Number of random requests for random IO: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Threads started!
Time limit exceeded, exiting...
Done.

Operations performed: 32310311 Read, 0 Write, 0 Other = 32310311 Total
Read 493.02Gb Written 0b Total transferred 493.02Gb 2.0542Gb/sec
134626.21 Requests/sec executed

Test execution summary:
total time:                240.0002s
total number of events:    32310311
total time taken by event execution: 213.2804
per-request statistics:
  min:                    0.00ms
  avg:                    0.01ms
  max:                    0.35ms
  approx. 95 percentile: 0.01ms

Threads fairness:
  events (avg/stddev):    32310311.0000/0.00
  execution time (avg/stddev): 213.2804/0.00

root@debian-server:~/Testes# █

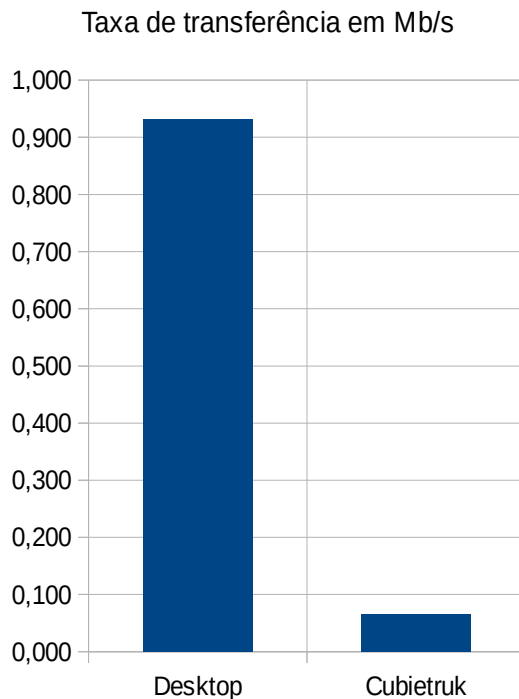
```

Figura 19. Sysbench – Leitura de arquivos - Servidor desktop

Fonte: Autoria própria.

Alguns resultados foram convertidos em *megabits* para se ter uma comparação. O resultado mostra que o servidor Cubietruck realizou a gravação do arquivo em uma taxa de transferência no valor de 0,065 Mb/s e na leitura sua taxa de transferência foi de 347,96 Mb/s. Já o servidor desktop realizou a gravação do arquivo com uma taxa de transferência no valor de 0,931 Mb/s, e para a leitura a taxa de transferência foi de 2.103,501 Mb/s. Em ambos os resultados o desempenho de gravação e leitura de arquivos do servidor desktop foi superior ao servidor Cubietruck, devido neste trabalho o servidor desktop utilizar como armazenamento de arquivos um disco rígido modelo SATA II (velocidade de até 300MB/s), e o servidor Cubietruck utilizar um cartão de memória micro SD de classe 4 (taxa mínima de transferência de dados de 4MB/s). Na Figura 20, é ilustrado estes resultados.

Sysbench - Teste gravação de arquivos



Sysbench - Teste leitura de arquivos

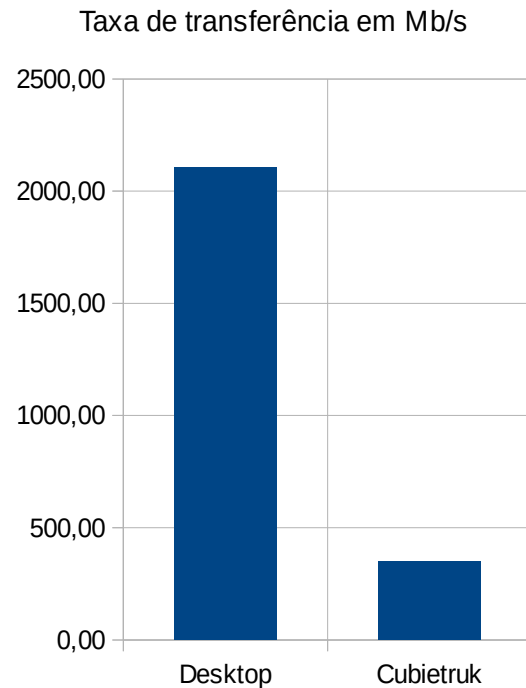


Figura 20. Resultado teste Sysbench – Gravação e leitura de arquivos em disco

Fonte: Autoria própria.

4.4.4 Sysbench - Teste de Banco de dados MySQL (Transações por segundo)

Este teste simula várias operações no banco de dados MySQL, onde no final é analisado quantas transações por segundo obteve. Para medir o desempenho, primeiramente foi criado uma tabela em cada servidor com 1.000.000 de linhas de dados. O comando utilizado foi `# time sysbench --test=oltp --oltp-table-size=1000000 --mysql-db=teste --mysql-user=root --mysql-password=senhaservidor prepare`. Após, o comando para executar o teste foi: `# sysbench --test=oltp --oltp-table-size=1000000 --mysql-db=Teste --mysql-user=root --mysql-password=senhaservidor --max-time=60 --oltp-read-only=on --max-requests=0 --num-threads=8 run`. As Figuras 21 e 22 mostram a execução dos testes nos servidores.

```

Threads started!
Time limit exceeded, exiting...
(last message repeated 7 times)
Done.

OLTP test statistics:
  queries performed:
    read:                86380
    write:                0
    other:               12340
    total:               98720
  transactions:         6170 (102.77 per sec.)
  deadlocks:            0 (0.00 per sec.)
  read/write requests: 86380 (1438.73 per sec.)
  other operations:     12340 (205.53 per sec.)

Test execution summary:
  total time:            60.0391s
  total number of events: 6170
  total time taken by event execution: 479.8862
  per-request statistics:
    min:                 32.55ms
    avg:                 77.78ms
    max:                 2305.78ms
    approx. 95 percentile: 97.17ms

Threads fairness:
  events (avg/stddev):  771.2500/4.55
  execution time (avg/stddev): 59.9858/0.01

root@server-cubietruck:~# █

```

Figura 21. Sysbench – Teste banco de dados – Servidor Cubietruck

Fonte: Autoria própria.

```

Threads started!
Time limit exceeded, exiting...
(last message repeated 7 times)
Done.

OLTP test statistics:
  queries performed:
    read:                720594
    write:                0
    other:               102942
    total:               823536
  transactions:         51471 (857.78 per sec.)
  deadlocks:            0 (0.00 per sec.)
  read/write requests: 720594 (12008.92 per sec.)
  other operations:     102942 (1715.56 per sec.)

Test execution summary:
  total time:            60.0049s
  total number of events: 51471
  total time taken by event execution: 479.6150
  per-request statistics:
    min:                 2.29ms
    avg:                 9.32ms
    max:                 35.89ms
    approx. 95 percentile: 11.78ms

Threads fairness:
  events (avg/stddev):  6433.8750/31.97
  execution time (avg/stddev): 59.9519/0.01

root@debian-server:~/Testes# █

```

Figura 22. Sysbench – Teste banco de dados – Servidor desktop

Fonte: Autoria própria.

O resultado mostra que o servidor Cubietruck realizou 102,77 transações por segundo, já o servidor desktop realizou 857,87 transações por segundo, ambos no tempo de 240 segundos, mostrando novamente o desempenho superior do servidor desktop comparado ao servidor Cubietruck. Na Figura 23, é ilustrado estes resultados:

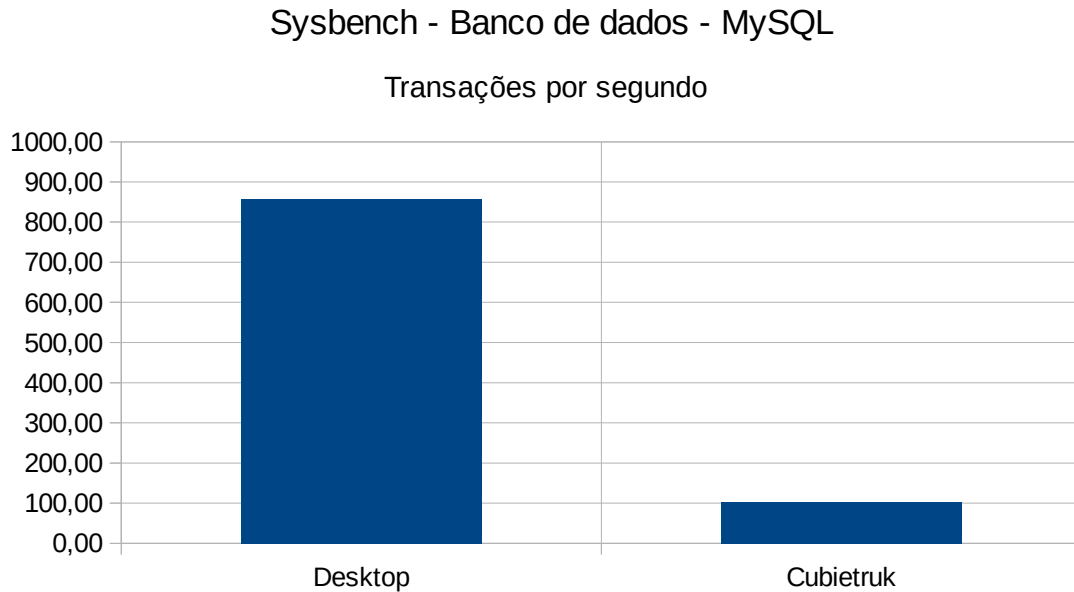


Figura 23. Resultado teste Sysbench - Banco de dados MySQL (Transações/segundo)

Fonte: Autoria própria.

4.5 IPERF

Para o teste de desempenho da rede, é utilizado a ferramenta *Iperf*, já descrito na seção 3.1.5. A ferramenta foi instalada nos servidores e nas estações de teste com o seguinte comando: `# apt-get install iperf`.

4.5.1 Iperf – Desempenho da Rede – 06 clientes

Este teste simula a geração de tráfego na rede com 06 clientes por 60 segundos. Nos servidores foi inserido o seguinte comando: `# iperf -s -f M`, para a ferramenta ficar aguardando a conexão com a estação.

Na estação foi inserido o seguinte comando: `iperf -c ipsevidor -P 6 -f M -t 60`. Nas Figuras 24 e 25 mostram os testes sendo executados com a conexão nos dois servidores.

```

root@finsterbusch: /home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@finsterbusch: /home/anderson# iperf -c 192.168.1.101 -P 6 -f M -t 60
-----
Client connecting to 192.168.1.101, TCP port 5001
TCP window size: 0.08 MByte (default)
-----
[ 8] local 192.168.1.115 port 48636 connected with 192.168.1.101 port 5001
[ 3] local 192.168.1.115 port 48631 connected with 192.168.1.101 port 5001
[ 4] local 192.168.1.115 port 48632 connected with 192.168.1.101 port 5001
[ 5] local 192.168.1.115 port 48633 connected with 192.168.1.101 port 5001
[ 6] local 192.168.1.115 port 48634 connected with 192.168.1.101 port 5001
[ 7] local 192.168.1.115 port 48635 connected with 192.168.1.101 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 5] 0.0-60.1 sec  112 MBytes   1.87 MBytes/sec
[ 6] 0.0-60.1 sec  112 MBytes   1.87 MBytes/sec
[ 3] 0.0-60.1 sec  112 MBytes   1.87 MBytes/sec
[ 4] 0.0-60.1 sec  112 MBytes   1.87 MBytes/sec
[ 7] 0.0-60.1 sec  112 MBytes   1.87 MBytes/sec
[ 8] 0.0-60.1 sec  113 MBytes   1.87 MBytes/sec
[SUM] 0.0-60.1 sec  675 MBytes  11.2 MBytes/sec
root@finsterbusch: /home/anderson#

```

Figura 24. Iperf – Desempenho da rede com 06 clientes - Servidor Cubietruck

Fonte: Autoria própria.

```

root@finsterbusch: /home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@cubietruck06: ~# iperf -c 192.168.1.201 -P 6 -f M -t 60
-----
Client connecting to 192.168.1.201, TCP port 5001
TCP window size: 0.02 MByte (default)
-----
[ 7] local 192.168.1.216 port 52073 connected with 192.168.1.201 port 5001
[ 4] local 192.168.1.216 port 52069 connected with 192.168.1.201 port 5001
[ 5] local 192.168.1.216 port 52070 connected with 192.168.1.201 port 5001
[ 6] local 192.168.1.216 port 52071 connected with 192.168.1.201 port 5001
[ 3] local 192.168.1.216 port 52068 connected with 192.168.1.201 port 5001
[ 8] local 192.168.1.216 port 52072 connected with 192.168.1.201 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 7] 0.0-60.0 sec  642 MBytes   10.7 MBytes/sec
[ 3] 0.0-60.0 sec  655 MBytes   10.9 MBytes/sec
[ 8] 0.0-60.1 sec  546 MBytes   9.09 MBytes/sec
[ 5] 0.0-60.1 sec  610 MBytes   10.2 MBytes/sec
[ 6] 0.0-60.1 sec  760 MBytes   12.6 MBytes/sec
[ 4] 0.0-60.3 sec  643 MBytes   10.7 MBytes/sec
[SUM] 0.0-60.3 sec  3856 MBytes  63.9 MBytes/sec
root@cubietruck06: ~#

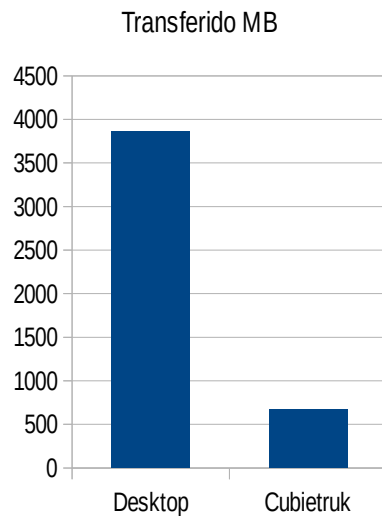
```

Figura 25. Iperf – Desempenho da rede com 06 clientes - Servidor desktop

Fonte: Autoria própria.

O resultado indica que no servidor Cubietruck houve 675MB transferidos com uma taxa de transferência de 11,2MB/s. Já no servidor desktop houve 3.856MB transferidos com uma taxa de transferência de 63,9MB/s, mostrando que tanto em quantidade de arquivos transferidos e largura de banda o servidor desktop é superior ao servidor Cubietruck. Na Figura 26 é ilustrado estes resultados.

Iperf - Desempenho da Rede - 6 clientes



Iperf - Desempenho da Rede - 6 clientes

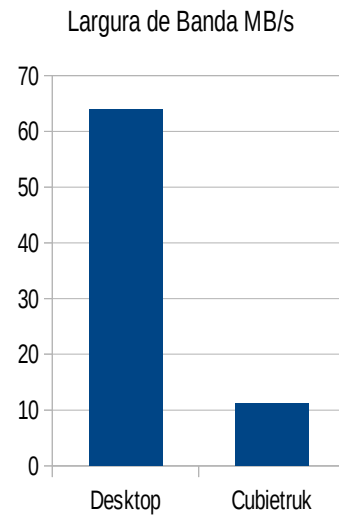


Figura 26. Resultado teste Iperf – Desempenho da rede com 6 clientes

Fonte: Autoria própria.

4.5.2 Iperf – Desempenho da Rede – Conexão ponto a ponto

Este teste simula a geração de tráfego na rede com a conexão ponto a ponto, ou seja, cliente com servidor, durante 60 segundos. Nos servidores foi inserido o seguinte comando: `# iperf -s -f M`, para a ferramenta ficar aguardando a conexão com a estação. Na estação foi inserido o seguinte comando: `iperf -c ipservidor -P 1 -f M -t 60`. Nas Figuras 27 e 28 mostram os testes sendo executados com a conexão nos dois servidores.

```

root@finsterbusch: /home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@finsterbusch:/home/anderson# iperf -c 192.168.1.101 -P 1 -f M -t 60
-----
Client connecting to 192.168.1.101, TCP port 5001
TCP window size: 0.08 MByte (default)
-----
[ 3] local 192.168.1.115 port 48629 connected with 192.168.1.101 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-60.0 sec  2670 MBytes 44.5 MBytes/sec
root@finsterbusch:/home/anderson#

```

Figura 27. Iperf – Desempenho da Rede com 06 clientes - Servidor Cubietruck

Fonte: Autoria própria.

```

root@finsterbusch: /home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@finsterbusch:/home/anderson# iperf -c 192.168.1.201 -P 1 -f M -t 60
-----
Client connecting to 192.168.1.201, TCP port 5001
TCP window size: 0.08 MByte (default)
-----
[ 3] local 192.168.1.220 port 53251 connected with 192.168.1.201 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-60.0 sec  6731 MBytes  112 MBytes/sec
root@finsterbusch:/home/anderson#

```

Figura 28. Iperf – Desempenho da Rede com 06 clientes - Servidor desktop

Fonte: Autoria própria.

O resultado é visualizado no formato MB (Megabytes), em que no servidor Cubietruck houve 2.670MB transferidos com uma taxa de transferência de 44,5MB/s. Já no servidor desktop houve 6.731MB transferidos com uma taxa de transferência de 112MB/s. No teste o servidor desktop é superior em quantidade de arquivo transferido e largura de banda comparado ao servidor Cubietruck, levando em consideração os testes de desempenho do processador, memória e disco anteriormente efetuados no que influência nos resultados. Estes números são representados pela Figura 29.

Iperf - Desempenho da Rede - Ponto a ponto Iperf - Desempenho da Rede - Ponto a ponto

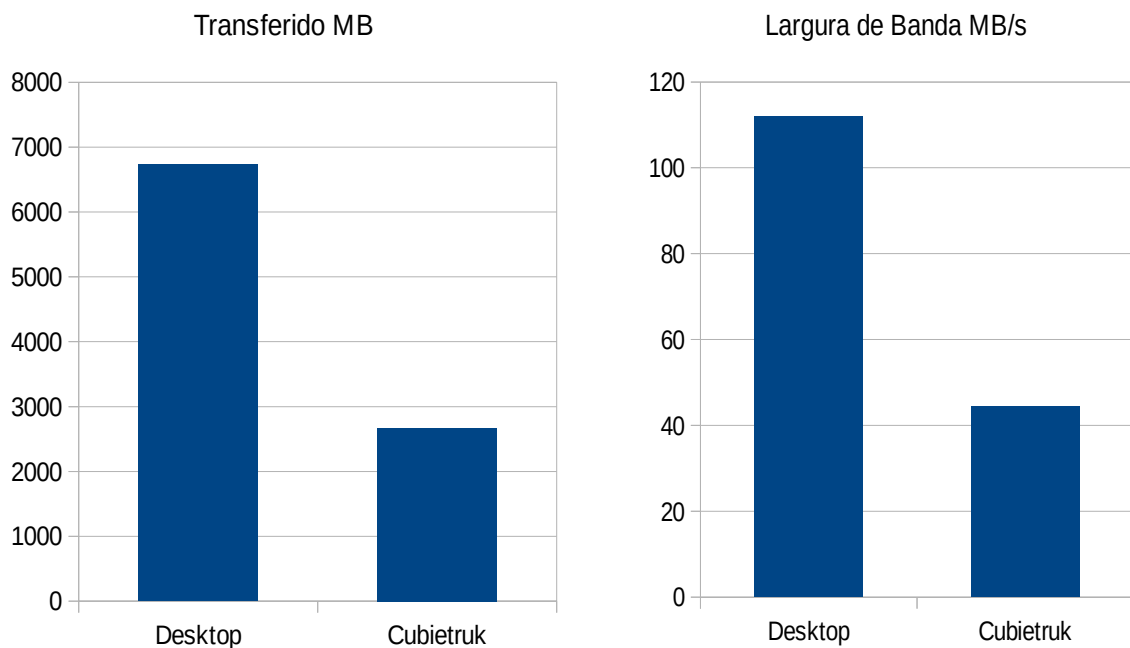


Figura 29. Resultado teste Iperf – Desempenho da rede ponto a ponto

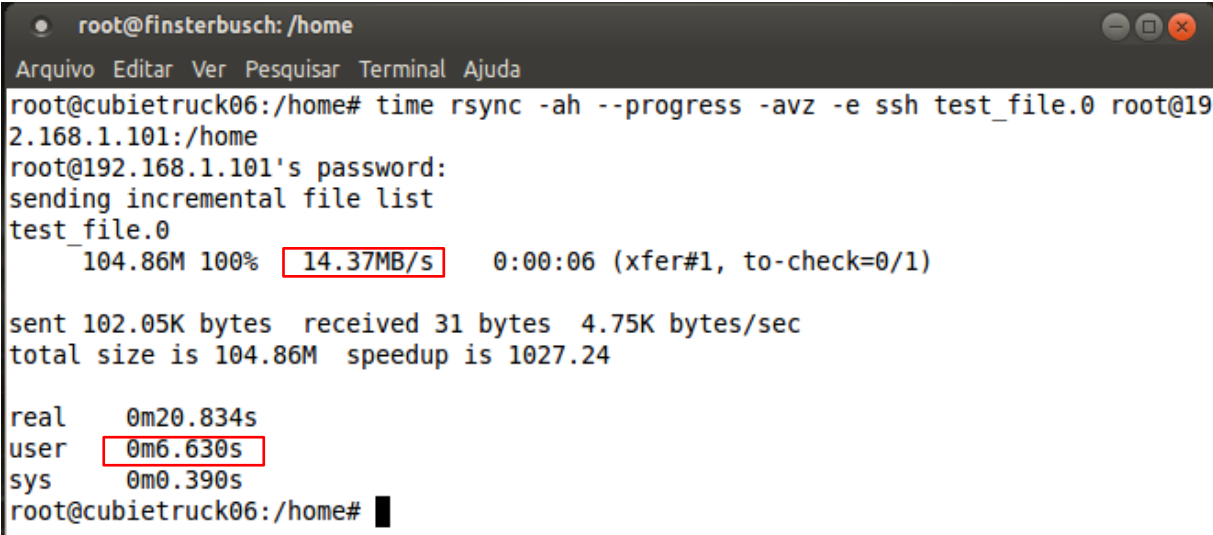
Fonte: Autoria própria.

4.6 RSYNC

Para o teste de desempenho da rede, foi utilizado a ferramenta *Rsync*, já descrito na seção 3.1.4. A ferramenta foi instalada nos servidores e nas estações de teste com o seguinte comando: `# apt-get install rsync`.

4.6.1 Rsync – Transferência de arquivos – Cliente X Servidor

Neste teste foi copiado um arquivo de 100MB de uma estação sendo cliente para o servidor, simulando transferência de arquivos ou backup. Na estação é inserido o seguinte comando: `# rsync -ah --progress -avz -e ssh "arquivoorigem" root@caminhodestino`. Será solicitado a senha do usuário para estabelecer a conexão. Nas Figuras 30 e 31 mostram os testes sendo executados com a conexão nos dois servidores.



```

root@finsterbusch: /home
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@cubietruck06:/home# time rsync -ah --progress -avz -e ssh test_file.0 root@192.168.1.101:/home
root@192.168.1.101's password:
sending incremental file list
test_file.0
  104.86M 100% 14.37MB/s   0:00:06 (xfer#1, to-check=0/1)

sent 102.05K bytes  received 31 bytes  4.75K bytes/sec
total size is 104.86M  speedup is 1027.24

real    0m20.834s
user    0m6.630s
sys     0m0.390s
root@cubietruck06:/home# █

```

Figura 30. Rsync – Transferência de arquivos – Cliente x Servidor - Cubietruck

Fonte: Autoria própria.

```

root@finsterbusch: /home
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@cubietruck06:/home# time rsync -ah --progress -avz -e ssh test_file.0 root@19
2.168.1.201:/home
root@192.168.1.201's password:
sending incremental file list
test_file.0
  104.86M 100% 14.38MB/s  0:00:06 (xfer#1, to-check=0/1)

sent 102.05K bytes  received 31 bytes  10.74K bytes/sec
total size is 104.86M  speedup is 1027.24

real    0m9.790s
user    0m6.680s
sys     0m0.270s
root@cubietruck06:/home# █

```

Figura 31. Rsync – Transferência de arquivos – Cliente x Servidor - Desktop

Fonte: Autoria própria.

O resultado indica que a transferência do arquivo para o servidor Cubietruck obteve uma taxa de transferência no valor de 14,37MB/s em um tempo de 6,63 segundos. Já para o servidor desktop a taxa de transferência foi de 14,38MB/s com um tempo de 6,68 segundos, como podemos observar na Figura 32, tanto o servidor Cubietruck como o servidor desktop obtiveram taxa de transferência e o tempo semelhantes como resultado.

Rsync - Transferência de arquivos

Conexão Cliente - Servidor

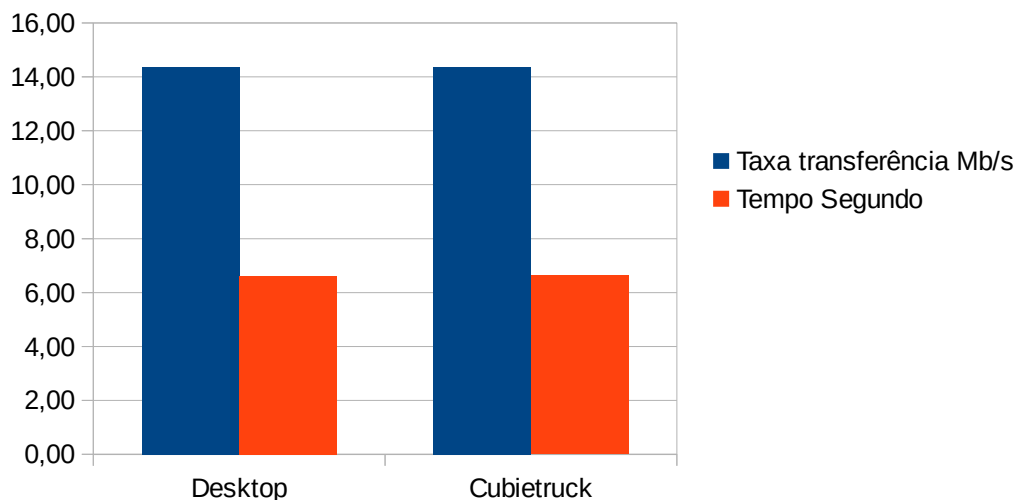
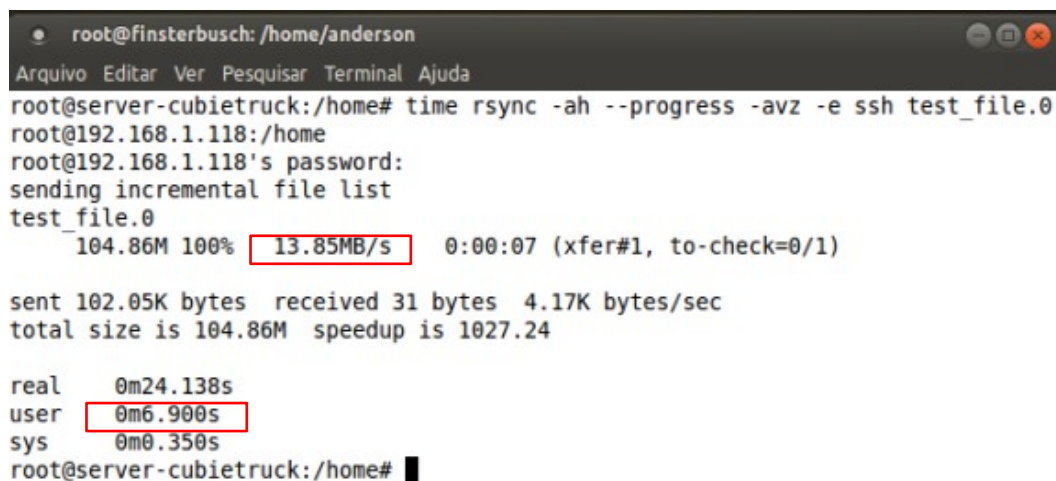


Figura 32. Resultado teste Rsync – Transferência de arquivos cliente x servidor

Fonte: Autoria própria.

4.6.2 Rsync – Transferência de arquivos – Servidor X Cliente

Neste teste foi copiado um arquivo de 100MB do servidor para uma estação cliente, simulando transferência de arquivos ou backup. No servidor é inserido o seguinte comando: `# rsync -ah --progress -avz -e ssh "arquivoorigem" root@caminhodestino`. Será solicitado a senha do usuário para estabelecer a conexão. Nas Figuras 33 e 34 mostram os testes sendo executados com a conexão nos dois servidores.



```

root@finsterbusch: /home/anderson
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@server-cubietruck:/home# time rsync -ah --progress -avz -e ssh test_file.0
root@192.168.1.118:/home
root@192.168.1.118's password:
sending incremental file list
test_file.0
  104.86M 100% 13.85MB/s  0:00:07 (xfer#1, to-check=0/1)

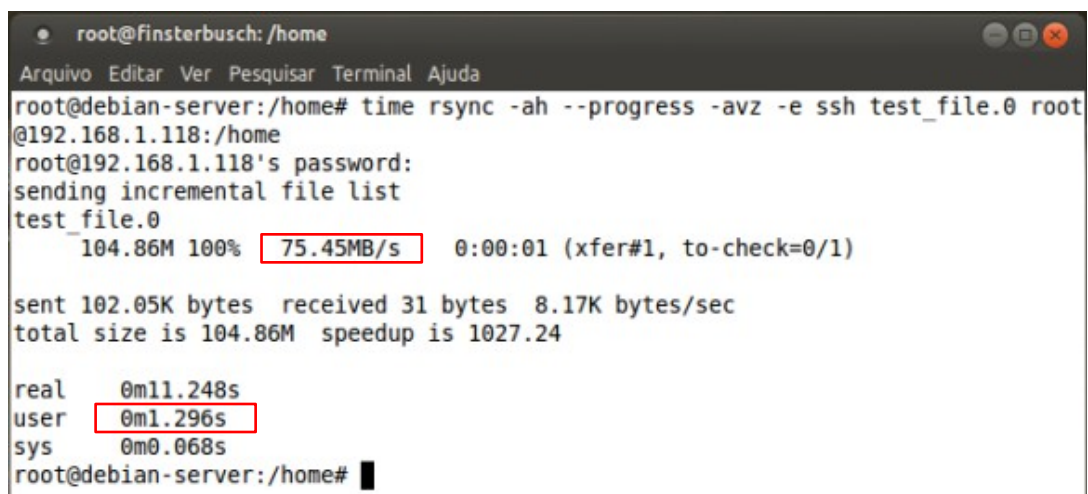
sent 102.05K bytes received 31 bytes  4.17K bytes/sec
total size is 104.86M speedup is 1027.24

real    0m24.138s
user    0m6.900s
sys     0m0.350s
root@server-cubietruck:/home#

```

Figura 33. Rsync – Transferência de arquivos – Servidor x Cliente - Cubietruck

Fonte: Autoria própria.



```

root@finsterbusch: /home
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@debian-server:/home# time rsync -ah --progress -avz -e ssh test_file.0 root
@192.168.1.118:/home
root@192.168.1.118's password:
sending incremental file list
test_file.0
  104.86M 100% 75.45MB/s  0:00:01 (xfer#1, to-check=0/1)

sent 102.05K bytes received 31 bytes  8.17K bytes/sec
total size is 104.86M speedup is 1027.24

real    0m11.248s
user    0m1.296s
sys     0m0.068s
root@debian-server:/home#

```

Figura 34. Rsync – Transferência de arquivos – Servidor x Cliente – Desktop

Fonte: Autoria própria.

O resultado indica que a transferência do arquivo do servidor Cubietruck para a estação, obteve uma taxa de transferência no valor de 13,85MB/s em um tempo de 6,90 segundos. Já do servidor desktop para a estação, a taxa de transferência foi de 75,45MB/s com um tempo de 1,29 segundos, mostrando que o servidor desktop foi superior neste teste. Na Figura 35, é ilustrado estes resultados:

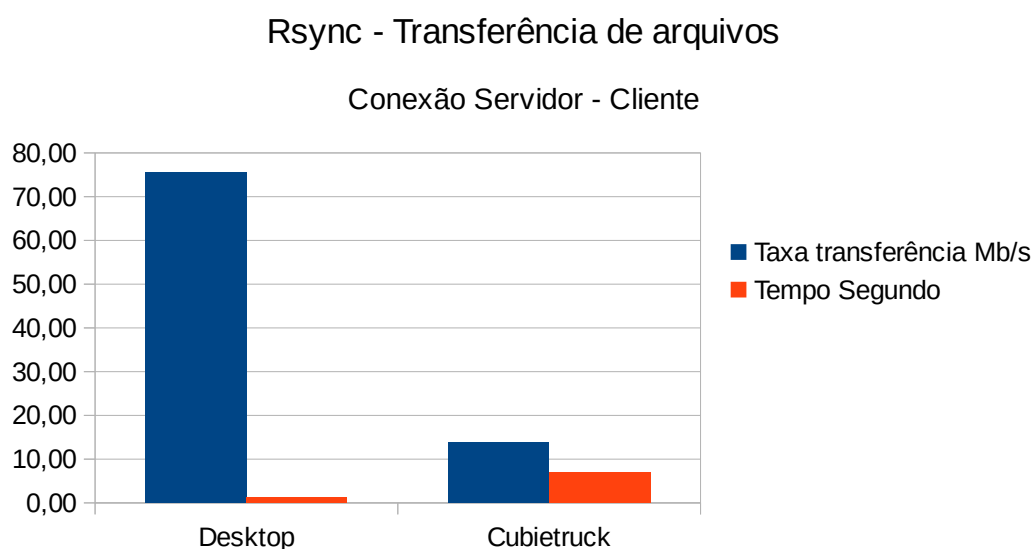


Figura 35. Resultado teste Rsync – Transferência de arquivos - Servidor x Cliente

Fonte: Autoria própria.

4.7 WIRESHARK

Para o teste de desempenho do *proxy* foi utilizado a ferramenta *Wireshark*, já descrito na seção 3.1.6. A ferramenta foi instalada em uma estação com o seguinte comando: `# apt-get install wireshark`.

4.7.1 Wireshark – Desempenho *proxy* – Servidor Cubietruck

Para realizar este teste, no navegador da estação não estava configurado o *proxy*, para podermos antes realizar o teste de acesso a um site sem o *proxy*. Iniciado a ferramenta *Wireshark*, foi definido a captura de pacotes da interface `eth0` e iniciado a captura no site: `www.infoexame.com.br`. A Figura 36 mostra o tempo que obteve o acesso ao site sem a configuração do *proxy* no navegador.

No.	Time	Source	Destination	Proto
9536	29.492997000	177.43.198.28	192.168.1.122	HTTP
9724	30.029318000	177.43.198.28	192.168.1.122	HTTP
9786	30.201017000	177.43.198.28	192.168.1.122	HTTP
9953	30.652082000	177.43.198.28	192.168.1.122	HTTP
10203	31.980701000	177.43.198.28	192.168.1.122	HTTP
10232	32.064435000	177.43.198.28	192.168.1.122	HTTP
10281	32.260672000	177.43.198.28	192.168.1.122	HTTP
10334	42.096424000	177.43.198.28	192.168.1.122	HTTP
10743	45.702151000	177.43.198.28	192.168.1.122	HTTP
10757	47.014455000	177.43.198.28	192.168.1.122	HTTP
10763	47.657589000	177.43.198.28	192.168.1.122	HTTP

▾ Frame 10757: 480 bytes on wire (3840 bits), 480 bytes captured (3840 bits) on
 Interface id: 0
 Encapsulation type: Ethernet (1)
 Arrival Time: Oct 1, 2015 22:24:27.905107000 BRT
 [Time shift for this packet: 0.000000000 seconds]
 Epoch Time: 1443749067.905107000 seconds
 [Time delta from previous captured frame: 0.038133000 seconds]
 [Time delta from previous displayed frame: 1.312304000 seconds]
 [Time since reference or first frame: 47.014455000 seconds]
 Frame Number: 10757
 Frame Length: 480 bytes (3840 bits)
 Capture Length: 480 bytes (3840 bits)
 [Frame is marked: False]
 [Frame is ignored: False]

Figura 36. Wireshark – Desempenho proxy – sem proxy – Servidor Cubietruck

Fonte: Autoria própria.

Após, foi inserido o endereço do servidor *proxy* no navegador e repetido o teste, acessando o mesmo site www.infoexame.com.br. Iniciado a captura do site pela ferramenta *Wireshark*, onde obtive o resultado conforme a Figura 37.

O resultado mostra o desempenho do servidor *proxy* no Cubietruck onde, sem a configuração do *proxy* no navegador, o acesso ao site levou 47,01 segundos. Já com a configuração do *proxy* no navegador, o acesso levou 5,83 segundos. Na Figura 38, é ilustrado estes resultados:

No.	Time	Source	Destination	Proto
6573	3.987428000	192.168.1.101	192.168.1.122	HTTP
6587	4.074464000	192.168.1.101	192.168.1.122	HTTP
6639	4.140330000	192.168.1.101	192.168.1.122	HTTP
6644	4.196344000	192.168.1.101	192.168.1.122	HTTP
6930	5.297726000	192.168.1.101	192.168.1.122	HTTP
6945	5.299048000	192.168.1.101	192.168.1.122	HTTP
7121	5.313453000	192.168.1.101	192.168.1.122	HTTP
7128	5.415114000	192.168.1.101	192.168.1.122	HTTP
7149	5.663227000	192.168.1.101	192.168.1.122	HTTP
7155	5.832423000	192.168.1.101	192.168.1.122	HTTP
7159	5.833849000	192.168.1.101	192.168.1.122	HTTP

▼ Frame 7159: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on in
 Interface id: 0
 Encapsulation type: Ethernet (1)
 Arrival Time: Oct 1, 2015 22:32:48.285284000 BRT
 [Time shift for this packet: 0.000000000 seconds]
 Epoch Time: 1443749568.285284000 seconds
 [Time delta from previous captured frame: 0.000009000 seconds]
 [Time delta from previous displayed frame: 0.001426000 seconds]
 [Time since reference or first frame: 5.833849000 seconds]
 Frame Number: 7159
 Frame Length: 101 bytes (808 bits)
 Capture Length: 101 bytes (808 bits)
 [Frame is marked: False]
 [Frame is ignored: False]

Figura 37. Wireshark – Desempenho proxy – com proxy – Servidor Cubietruck
 Fonte: Autoria própria.

Wireshark - Desempenho Proxy

Tempo em segundos

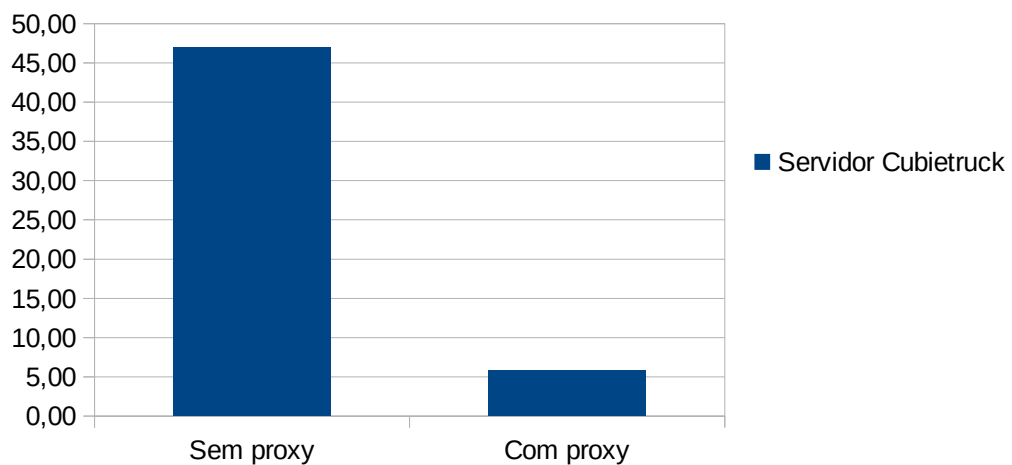


Figura 38. Wireshark – Desempenho proxy – Servidor Cubietruck
 Fonte: Autoria própria.

4.8 CONSUMO DE ENERGIA

Para realizar o teste de consumo de energia, foi utilizado a ferramenta Multímetro Alicate Amperímetro, descrita na seção 3.1.2. A ferramenta foi utilizada apenas no servidor desktop, pois no servidor Cubietruck os dados do consumo de energia foram utilizados no máximo especificado pelo fabricante.

4.8.1 Consumo de energia – Servidor desktop

O primeiro teste realizado no servidor desktop, foi o consumo de energia em operação normal, sem exigir processamento do computador. A fonte de energia do servidor é ligado em uma extensão com uma bobina de fio de cobre com 10 voltas, para que ao medir a intensidade da corrente elétrica com a alicate do multímetro, o resultado é amplificado e obtido na medida em *ampere*. Na Figura 39, é exibido esta medição.



Figura 39. Consumo de energia – Operação normal

Fonte: Autoria própria.

O segundo teste realizado no servidor desktop, foi exigido o consumo máximo do processador, utilizando o comando `#sysbench --test=cpu --num-threads=2 run`, o processador operou em 100%. Com isso se obtêm uma estimativa de carga máxima de consumo no servidor. Na Figura 40, é exibido esta medição.



Figura 40. Consumo de energia – Processamento máximo
Fonte: Autoria própria.

O resultado mostra que a intensidade da corrente elétrica medida no servidor desktop em operação normal foi de 0,43 *ampere*, pois é preciso dividir o valor obtido por 10, onde foi utilizado uma bobina para amplificar o sinal. Na medição com o processador operando no máximo, o resultado obtido foi de 0,65 *ampere*.

O consumo de um computador é calculado em *watts*, ou seja, é preciso transformar a medida obtida de *ampere* para *watts*. O cálculo é realizado com a seguinte fórmula: $Watts = Volt \times Ampere$. No momento do teste, a tensão (volt) medida foi de 121v. Na Tabela 1, mostra-se o consumo obtido em *watts*:

Servidor Desktop	Volts	Ampere	Watts
Consumo normal	121	0,43	52,03
Consumo máximo	121	0,65	78,65

Tabela 1. Consumo de energia – Cálculo em watts – Servidor desktop
Fonte: Autoria própria.

Estimando que este servidor fique ligado 24 horas/dia, baseando em uma média de 8 horas de trabalhos diários, e as outras 16 horas em consumo normal, um total de 30 dias trabalhados no mês, e que o custo do quilowatts/hora é de R\$ 0,74592 (COPEL, 2015), o custo de energia para este servidor será de aproximadamente R\$ 37,14. Na Tabela 2, mostra-se o cálculo do consumo mensal total em quilowatts e o valor em reais.

	Média Watts	Total Watts
Média (8hs)	65,34	522,72
22 dias		11.499,84
Restante (16hs)	52,03	832,48
22 dias		18.314,56
Fim de semana (48h)	52,03	2.497,44
8 dias		19.979,52
	TOTAL	49.793,92
	DIVIDE 1000	
	Total Kwatts/h	49,79392
Custo R\$	0,74592	R\$ 37,14

Tabela 2. Consumo de energia – Cálculo em reais – Servidor desktop

Fonte: Autoria própria.

No servidor Cubietruck, o seu consumo máximo é de 5w com uma tensão de 5v. Na Tabela 3 mostra-se o cálculo de consumo mensal em reais, com o servidor trabalhando em máximo consumo, onde teve como resultado o valor de aproximadamente R\$ 3,40 mensais.

	Média Watts	Total Watts
Média (8hs)	5	40,00
22 dias		880,00
Restante (16hs)	5	80,00
22 dias		1.760,00
Fim de semana (48h)	5	240,00
8 dias		1.920,00
	TOTAL	4.560,00
	DIVIDE 1000	
	Total Kwatts/h	4,56
Custo R\$	0,74592	R\$ 3,40

Tabela 3. Consumo de energia – Cálculo em reais – Servidor Cubietruck

Fonte: Autoria própria.

4.9 CUSTO DE EQUIPAMENTOS

Realizado o levantamento do custo do dispositivo Cubietruck e de um modelo de computador desktop semelhante do trabalho, como resultado da pesquisa, obtêm-se a Tabela 4 com os valores:

Cubietruck			Desktop		
Loja	Descrição	Valor	Loja	Descrição	Valor
Loja Mundi	01 CubieTruck		Ponto Frio	Marca: Daten	
www.lojamundi.com.br	01 Dissipador de calor		www.pontofrio.com.br	Processador: Dual Core Celeron	
	01 Cabo USB "B" OTG			Memória: 2 Gb	
	01 Cabo USB de energia			Disco Rígido: 500Gb	
	01 Cabo para HD SATA 2.5				R\$ 940,90
	01 Cabo extensor USB "B" para USB Fêmea				R\$ 70,30
	01 Case acrílico	R\$ 486,80			R\$ 1.011,20
	01 Fonte	R\$ 38,03			
	FRETE	R\$ 20,00			
		R\$ 544,83			
	Preço avaliado dia 17/09/2015				

Tabela 4. Custo para aquisição dos servidores

Fonte: Autoria própria.

5 CONCLUSÃO

Este trabalho teve como objetivo comparar do servidor Cubietruck com um servidor desktop, para verificar se este equipamento atende as necessidades de empresas de pequeno porte, com o intuito de reduzir custos.

Ao término deste trabalho levando em consideração os testes realizados, ficou evidente que o servidor Cubietruck tem um desempenho menor comparado com o servidor desktop, principalmente em processamento de dados, que influencia nos resultados. Outro fator que influenciou nos resultados foi a utilização de um cartão de memória micro SD de classe 4 no servidor Cubietruck, em que tem um desempenho inferior ao disco rígido utilizado no servidor desktop. Porém se a empresa não precisar do servidor operando como gerenciador de banco de dados, ou aplicações que necessitam de processamento, a empresa pode optar pelo Cubietruck com o cartão de memória, pois nele é possível rodar serviços como: *firewall*, *proxy*, sem exigir que o equipamento trabalhe no limite de sua capacidade, permitindo ainda algum crescimento futuro na rede.

Podemos concluir que o Cubietruck tem um custo/benefício, baseando no levantamento do equipamento no mercado e também em seu consumo energético reduzido, onde há possibilidade de serviços funcionarem por mais tempo com o uso de um *no-break*, o que é limitado com um servidor desktop. Além de que o seu tamanho não ocupa grande espaço físico, podendo até ser implantado em pequenos escritórios.

O objetivo deste trabalho foi alcançado, onde foi possível mesmo em laboratório, simular uma rede e rodar os principais serviços em um servidor, realizando assim a comparação entre os servidores.

Para trabalhos futuros pretende-se realizar o mesmo teste, porém utilizando o dispositivo Cubieboard4, que possui um processador *Octa-Core*, estimando-se assim que o desempenho de processamento seja superior ao do Cubietruck.

REFERÊNCIAS

COPEL. **Copel – Taxas e Tarifas**. Disponível em: <www.copel.com/hpcopel/tarifas/> Acessado em 04/10/2015.

CUBIEBOARD. **Cubieboard3(Cubietruck)** Disponível em: <<http://cubieboard.org/model/cb3/>> Acessado em 19/04/2015.

CUBIETRUCK. **Cubietruck.com**. Disponível em: <<http://www.cubietruck.com/products/cubietruck-cubieboard3-cortex-a7-dual-core-2gb-ram-8gb-flash-with-wifi-bt-1>> Acessado em 18/04/2015.

DL.CUBIEBOARD. **DL.cubieboard.org**. Disponível em: <<http://dl.cubieboard.org/model/cubietruck/Image/Debian-server/debian-server-v1.1/debian-server-ct-card0-vga-v1.1.img.7z>> Acessado em 26/04/2015.

DEBIAN. **Arquivo de imagens versões Debian** Disponível em: <<http://cdimage.debian.org/cdimage/archive/7.8.0/i386/list-cd/debian-7.8.0-i386-netinst.iso>> Acessado em 25/04/2015.

FOXLUX. **Multímetro digital com alicate amperímetro**. Disponível em: <<http://www.foxlux.com.br/site/produtos/materiais-eletricos/multimetro-digital-com-alicate-amperimetro/>> Acessado em 01/10/2015.

IPERF. **The network bandwidth measurement tool**. Disponível em: <<https://iperf.fr/>> Acessado em 03/10/2015.

MIKROTIK. **Roteadores Mikrotik**. Disponível em: <<http://routerboard.com/RB2011UiAS-RM>> Acessado em 01/10/2015.

RSYNC. **Rsync.samba.org**. Disponível em: <<https://rsync.samba.org/>> acessado em 01/10/2015.

SAMBA. **Samba.org**. Disponível em: <<https://www.samba.org>> Acessado em 16/08/2015.

SSH. **Openssh.com**. Disponível em: <<http://www.openssh.com>> Acessado em 15/08/2015.

SQUID. **Squid-cache.org**. Disponível em: <<http://www.squid-cache.org>> Acessado em 15/08/2015.

SYSBENCH. **Github.com**. Disponível em: <<https://github.com/akopytov/sysbench>> Acessado em 01/10/2015.

WIRESHARK. **Wireshark.org**. Disponível em: <<https://www.wireshark.org/>> Acessado em 03/10/2015.

ANEXOS

Anexo 1 – Instalação sistema operacional Debian no dispositivo Cubietruck

Para instalação do sistema operacional Debian no Cubietruck, gravar a imagem do sistema no cartão de memória. A imagem está disponível no seguinte endereço:

<http://dl.cubieboard.org/model/cubietruck/Image/Debian-server/debian-server-v1.1/debian-server-ct-card0-vga-v1.1.img.7z>, já na versão para arquitetura ARM, e configurada para cartão de memória no modo de vídeo VGA.

Para gravar a imagem no cartão de memória, é necessário o programa *Win32DiskImager*, onde está disponível no endereço:

<http://dl.cubieboard.org/software/tools/windows/win32diskimager-v0.7-binary.zip>.

Este software é desenvolvido para Windows, e basta apenas carregar a imagem e copiar para a unidade do cartão de memória.

Após gravar a imagem no cartão de memória, inserir o cartão no Cubietruck e aguardar o sistema operacional ser carregado. Não é necessário fazer instalação, pois na própria imagem já está realizado a instalação. Por padrão a senha do *root* é cubieboard.

Anexo 2 – Instalação sistema operacional Debian no computador desktop

Para a instalação do sistema operacional Debian no computador desktop, baixar a imagem do sistema no seguinte endereço:

<http://cdimage.debian.org/cdimage/archive/7.8.0/i386/iso-cd/debian-7.8.0-i386-netinst.iso>.

Após gravar a imagem em um CD e iniciar o computador desktop com o CD, seguir as configurações padrões de instalação que é informada no passo a passo do CD.

Anexo 3 – Instalação e configuração de serviços: SSH – Acesso Remoto

Instalação

No terminal com acesso *root*, digitar o seguinte comando:

```
root@server-cubietruck:~# apt-get install openssh-server openssh-client
```

Configuração

As configurações do servidor SSH estão no arquivo */etc/ssh/sshd_config*.

Informar os endereços onde o serviço estará ativo. Neste caso serão liberados conexões no endereço do servidor *192.168.1.101*, através da porta *22* e utilizando a versão *2* do protocolo SSH:

[/etc/ssh/sshd_config](#)

```
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will
bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
ListenAddress 192.168.1.101
Protocol 2

#[...]
```

Por segurança, o servidor SSH apenas permite o acesso remoto do usuário *root* sem senhas. Deste modo, para adquirir privilégios de *root*, deverá ser efetuado o *login* com direitos não privilegiado e em seguida adquirir privilégios de *root*. Desta maneira a senha de *root* estará protegida de uma possível invasão.

[/etc/ssh/sshd_config](#)

```
#[...]  
  
# Authentication:  
LoginGraceTime 120  
PermitRootLogin without-password  
StrictModes yes  
  
#[...]
```

Ajustar também para não permitir *logins* com senhas vazias:

[/etc/ssh/sshd_config](#)

```
#[...]  
  
# To enable empty passwords, change to yes (NOT RECOMMENDED)  
PermitEmptyPasswords no  
  
#[...]
```

Reiniciar o serviço:

```
root@server-cubietruck:~# service ssh restart
```

Anexo 4 – Instalação e configuração de serviços: NTP – Relógio do sistema

Instalação

```
root@server-cubietruck:~# apt-get install ntp ntp-doc
```

As informações de fuso horário encontra-se no arquivo `/etc/timezone`. Com o seguinte comando verificamos qual fuso horário o servidor está atuando no momento.

```
root@server-cubietruck:~# cat /etc/timezone  
America/Sao_Paulo
```

Configuração

Caso precise alterar a zona do fuso horário, utilizar seguinte o comando:

```
root@server-cubietruck:~# dpkg-reconfigure tzdata
```

Nas opções, escolher a região geográfica e em seguida a região do país. Será exibido um resultado da seleção.

```
Current default time zone: 'America/Sao_Paulo'  
Local time is now:      Sun Aug 23 10:09:38 BRT 2015.  
Universal Time is now: Sun Aug 23 13:09:38 UTC 2015.
```

O comando `date` exibe a data e hora atual do sistema.

```
root@server-cubietruck:~# date  
Dom Ago 23 10:11:09 BRT 2015
```

O mesmo comando `date` também permite alterar a hora do sistema manualmente, utilizando a sintaxe `date <MMDDhhmm>`:

```
root@server-cubietruck:~# date 08221200  
Sat Ago 22 12:00:00 WEST 2015
```

Verificação

Para fazer a verificação inserir o comando `ntpq -p`, onde informará quais servidores estão conectados.

```
root@server-cubietruck:~# ntpq -p
      remote           refid      st t when poll reach   delay   offset  jitter
=====
+four2.fairy.mat 200.98.196.212  2 u  36  64  373 183.086    8.714   4.118
*ns2.pads.ufrj.br .GPS.           1 u  37  64  377   61.350   12.551   3.563
-vel.itat.io     131.188.3.223  2 u  38  64  377 279.202   26.664   2.229
+alpha.rueckgr.a 36.224.68.195  2 u  24  64  377 280.639   24.708   2.936
```

Anexo 5 – Instalação e configuração de serviços: DHCP

Instalação

```
root@server-cubietruck:~# apt-get install isc-dhcp-server
```

Configuração

Para configurar as interfaces, deve-se alterar o arquivo `/etc/default/isc-dhcp-server`.

[/etc/default/isc-dhcp-server](#)

```
# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#
# [...]
```

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP
requests?
#     Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth0"
```

A configuração principal fica no arquivo `/etc/dhcp/dhcpd.conf`. Neste arquivo é informado o nome de domínio (*option domain-name "home.lan";*), os endereços dos servidores (*option domain-name-servers 192.168.1.100, 192.168.1.1;*). Também pode ser definido o período e máximo que o endereço IP é atribuído (*default-lease-time 600 e max-lease-time 7200;*).

[/etc/dhcp/dhcpd.conf](#)

```
#
# Sample configuration file for ISC dhcpd for Debian
#
#
# The ddns-updates-style parameter controls whether or not the server
will
# attempt to do a DNS update when a lease is confirmed. We default to
the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks...
option domain-name "tcc.posredes";
option domain-name-servers 192.168.1.100, 192.168.1.1;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# [...]
```

Para a rede `192.168.1.0`, recomenda-se utilizar um bloco de endereços entre (*range 192.168.1.32 192.168.1.63;*), no *router* utilizar (*option routers 192.168.1.1;*) e para endereço de *broadcast* (*option broadcast-address 192.168.1.255;*)

[/etc/dhcp/dhcpd.conf](#)

```
# [...]  
  
## SubNet home.lan  
subnet 192.168.1.0 netmask 255.255.255.0 {  
    range 192.168.1.32 192.168.1.63;  
    option routers 192.168.1.1;  
    option broadcast-address 192.168.1.255;  
}
```

Reiniciar o serviço *dhcp*:

```
root@server-cubietruck:~# service isc-dhcp-server restart
```

Anexo 6 – Instalação e configuração de serviços: Proxy - Squid

Instalação

```
root@server-cubietruck:~# apt-get install squid3
```

Configuração

A configuração do *squid3* é realizado no arquivo */etc/squid3/squid.conf*. Por padrão a porta de conexão é a 3128, mas pode ser alterada para 8080 que é utilizada para serviço de cache da internet.

[/etc/squid3/squid.conf](#)

```
#[...]  
  
# Squid normally listens to port 3128  
http_port 3128  
  
#[...]
```


Por segurança o *squid3* só responderá a solicitações proveniente da rede local ou no próprio servidor. Esta restrição é definido por uma lista de controle de acesso ou (*Access Control List*) (*acl home.lan src 192.168.1.0/24*) e autorizado o acesso apenas aos sistemas incluídos nessa lista (*http_access allow home.lan*):

[/etc/squid3/squid.conf](#)

```
# [...]

# TAG: acl
#     Defining an Access List

# [...]

#Default:
# ACLs all, manager, localhost, and to_localhost are predefined.
#
#
# Recommended minimum configuration:
#

# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
#acl localnet src 10.0.0.0/8      # RFC1918 possible internal network
#acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
#acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
#acl localnet src fc00::/7      # RFC 4193 local private network range
#acl localnet src fe80::/10     # RFC 4291 link-local (directly
# plugged) machines

acl home.lan src 192.168.1.0/24

# [...]
```

As definições de permissão de acesso, estão na seção *http_access*:

[/etc/squid3/squid.conf](#)

```
# [...]

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
#http_access allow localnet
http_access allow localhost
```

```

http_access allow home.lan

# And finally deny all other access to this proxy
http_access deny all

# [...]

```

O tamanho total de armazenamento em *cache* do *squid3* pode ser ajustado por qualquer valor, como por exemplo, 1024Mb:

[/etc/squid3/squid.conf](#)

```

# [...]

# Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/spool/squid3 100 16 256
cache_dir ufs /var/spool/squid3 1024 16 256

# [...]

```

O nome do servidor *proxy* pode também ser definido:

[/etc/squid3/squid.conf](#)

```

# [...]

# TAG: visible_hostname
#     If you want to present a special hostname in error messages,
#     etc,
#     define this.  Otherwise, the return value of gethostname()
#     will be used.  If you have multiple caches in a cluster and
#     get errors about IP-forwarding you must set them to have
#     individual
#     names with this setting.
#Default:
# visible_hostname localhost
visible_hostname proxy.home.lan

# [...]

```

Para testar o arquivo de configuração, utilizar o comando:

```

root@server-cubietruck:~# squid3 -k parse

```

Para reiniciar o serviço *squid3*:

```
root@server-cubietruck:~# service squid3 restart
```

Anexo 7 – Instalação e configuração de serviços: Banco de dados - MySQL

Instalação

```
root@server-cubietruck:~# apt-get install mysql-server mysql-client
```

Durante o processo de instalação é solicitado a senha de *root* do servidor MySQL.

Configuração

Por segurança a configuração predefinida do MySQL aceitará apenas conexões locais (endereço *127.0.0.1*). Caso seja necessário alterar o endereço do servidor, essa alteração é realizada no *bind-address* que fica no arquivo de configuração */etc/mysql/my.cnf*.

[/etc/mysql/my.cnf](#)

```
# [...]
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address          = 127.0.0.1
# [...]
```

Verificação

```
root@server-cubietruck:~# mysql -u root -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 42

Server version: 5.5.30-1.1 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> show databases;
```

```
+-----+
```

```
| Database          |
```

```
+-----+
```

```
| information_schema |
```

```
| bacula             |
```

```
| mysql              |
```

```
| performance_schema |
```

```
+-----+
```

```
4 rows in set (0.02 sec)
```

```
mysql> quit
```

Bye

```
root@server-cubietruck:~#
```

Anexo 8 – Instalação e configuração de serviços: Compartilhamento de arquivos – Samba

Instalação

```
root@server-cubietruck:~# apt-get install samba
```

Configuração

Toda configuração do Samba é realizado no arquivo/ *etc/samba/smb.conf*.

Configuração global

[/etc/samba/smb.conf](#)

```
# [...]

#===== Global Settings =====

[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will
part of
    workgroup = cubietruck

# server string is the equivalent of the NT Description field
    server string = %h server

# [...]
```

Por segurança, define-se o acesso ao servidor samba a apenas alguns blocos da rede. Deste modo, o serviço *samba* só aceitará ligações do próprio servidor através da interface *lo* (127.0.0.1/8) ou através da interface de rede *eth0* (192.168.1.0/24).

[/etc/samba/smb.conf](#)

```
# [...]

##### Networking #####

# The specific set of interfaces / networks to bind to
# This can be either the interface name or an IP address/netmask;
# interface names are normally preferred
    interfaces = 127.0.0.0/8 192.168.1.0/24

# Only bind to the named interfaces and/or networks; you must use the
# 'interfaces' option above to use this.
# It is recommended that you enable this feature if your Samba machine
is
# not protected by a firewall or is a firewall itself. However, this
# option cannot handle dynamic or non-broadcast interfaces correctly.
    bind interfaces only = yes

# [...]
```

Compartilhamento por usuários

Por padrão do samba compartilha o diretório */home* de cada usuário. Porém podemos deixar restrito esse acesso a apenas um diretório específico de cada usuário:

```
root@server-cubietruck:~# mkdir /etc/skel/Documents; chmod 0700
/etc/skel/Documents
```

Criar o diretório de arquivos para cada usuário existente:

```
root@server-cubietruck:~# su - anderson -c 'mkdir ~/Documents; chmod
0700 ~/Documents'
```

Em seguida são definidos os parâmetros de compartilhamento de cada usuário, indicando o caminho do diretório e permitindo o acesso em escrita:

[/etc/samba/smb.conf](#)

```
# [...]

#===== Share Definitions =====

[homes]
  comment = Home Directories
  path = /home/%S/Documents
  browseable = no

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
  read only = no

# [...]
```

Criação de usuários

```
root@server:~# smbpasswd -a anderson
New SMB password:
```

Retype new SMB password:

Compartilhamento comum

É possível definir também um compartilhamento comum a todos os usuários, onde é criado um diretório de armazenamento de arquivos, inserindo os direitos de acesso. Também criar um grupo chamado *nogroup*:

```
root@server:~# mkdir -p /home/common/public
root@server:~# chgrp nogroup /home/common/public
root@server:~# chmod 0770 /home/common/public
```

Em seguida, deverá ser definido o compartilhamento no arquivo de configuração do samba, em que é definido o caminho para o diretório, e que a máscara de criação de arquivos e diretórios permitirá o acesso aos próprios usuários e ao grupo *nogroup*. Todas as alterações são feitas no ficheiro */etc/samba/smb.conf*:

[/etc/samba/smb.conf](#)

```
# [...]

### public share ###

[public]
    comment = Public Share
    path = /home/common/public
    public = yes
    browseable = yes
    writable = yes
    create mask = 0770
    directory mask = 0770
    force group = nogroup

# [...]
```

Lixeira

Para ativar a “Lixeira” no compartilhamento do *home* de cada usuário, editar o arquivo `/etc/samba/smb.conf`. Nesta configuração, quando o usuário apagar um arquivo, estes serão movidos para a pasta `@RecycleBin`, podendo ser recuperados.

[/etc/samba/smb.conf](#)

```
# [...]

#===== Share Definitions =====

[homes]
  comment = Home Directories
  path = /home/%S/Documents
  browseable = no

  # Support for Shared Recycle Bin
  vfs object = recycle
    recycle:repository = @RecycleBin
    recycle:keeptree = yes
    recycle:versions = yes
    recycle:exclude = *.tmp, *.bak

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
  read only = no

# [...]
```

Validação da configuração

Para validar toda configuração realizada no samba, executar o seguinte comando:

```
root@server-cubietruck:~# testparm

Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit
(16384)
Processing section "[homes]"
Processing section "[public]"
Processing section "[printers]"
Processing section "[print$]"
```



```
Loaded services file OK.  
Server role: ROLE_STANDALONE  
Press enter to see a dump of your service definitions
```

Finalmente, reiniciar os serviços:

```
root@server:~# service smbd restart  
ieroot@server:~# service nmbd restart
```

Anexo 9 – Instalação e configuração de serviços: UFW - Firewall

Instalação

```
root@server-cubietruck:~# apt-get install ufw
```

Configuração

A configuração do *ufw* é feita através da linha de comandos.

Por padrão o *ufw* autoriza o servidor a iniciar comunicações externas (*DEFAULT_OUTPUT_POLICY="ACCEPT"*), mas ignora qualquer tentativa de acesso externo (*DEFAULT_INPUT_POLICY="DROP"*). Esta configuração bloqueia, por exemplo, o acesso remoto a um servidor.

Primeiramente liberar o acesso remoto ao servidor via *ssh*:

```
root@server-cubietruck:~# ufw allow ssh/tcp
```

Agora é ativado o registro de eventos:

```
root@server-cubietruck:~# ufw logging on
```

E finalmente ativamos o firewall:

```
root@server-cubietruck:~# ufw enable
```

```
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

Neste momento só será possível acessar remotamente o servidor usando o protocolo *SSH*, que utiliza a porta 22. Demais serviços será bloqueado o acesso remoto.

Para verificar o estado do *firewall*, usamos o seguinte comando:

```
root@server-cubietruck:~# ufw status verbose
```

```
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing)
New profiles: skip
```

To	Action	From
--	-----	----
22/tcp	ALLOW IN	Anywhere
22/tcp	ALLOW IN	Anywhere (v6)

Conforme a necessidade pode-se ir liberando o acesso as portas, para pacotes tcp e udp, ou separadamente:

```
root@server-cubietruck:~# ufw allow 53/tcp
```

```
root@server-cubietruck:~# ufw allow 53/udp
```

```
root@server-cubietruck:~# ufw allow 53
```