

UNIVERSIDADE DO SAGRADO CORAÇÃO

RODRIGO CESAR RODRIGUES

**ANÁLISE DE DESEMPENHO DE UM CLUSTER DO
TIPO BEOWULF**

BAURU
2015

RODRIGO CESAR RODRIGUES

**ANÁLISE DE DESEMPENHO DE UM CLUSTER DO
TIPO BEOWULF**

Trabalho de conclusão de curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração, como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob a orientação do Prof. Esp. André Luiz Ferraz Castro.

BAURU
2015

Rodrigues, Rodrigo Cesar

R6962a

Análise de Desempenho de um Cluster do tipo Beowulf / Rodrigo Cesar Rodrigues. -- 2015.
96f. : il.

Orientador: Prof. Esp. André Luiz Ferraz Castro.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração – Bauru – SP.

1. Benchmarks. 2. Cluster Beowulf. 3. Linux. I. Castro, André Luiz Ferraz. II. Título.

RODRIGO CESAR RODRIGUES

ANÁLISE DE DESEMPENHO DE UM CLUSTER DO TIPO BEOWULF

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, sob a orientação do Prof. Esp. André Luiz Ferraz Castro.

Banca examinadora:

Prof. Esp. André Luiz Ferraz Castro
Universidade do Sagrado Coração

Prof. Me. Henrique Pachioni Martins
Universidade do Sagrado Coração

Prof. Me. Patrick Pedreira Silva
Universidade do Sagrado Coração

Bauru, 07 de dezembro de 2015.

Dedico este trabalho aos meus pais: Paulo Rodrigues e Roseli de Oliveira Rodrigues e a minha irmã: Danielle Cristina Rodrigues, por tudo que fizeram por mim, e por todos os valores éticos passados.

AGRADECIMENTOS

Agradeço a todos que me ajudaram na produção desse trabalho.

Agradeço primeiramente a meu orientador Esp. André Luiz Ferraz Castro, pelo seu conhecimento passado a mim, e por todo o seu tempo em me ajudar a compor este trabalho.

Agradeço Professor Dr. Elvio Gilberto da Silva por todos os seus conselhos, cobranças e indicações.

Agradeço os professores que me auxiliaram neste trabalho com suas críticas e sugestões para aprimora-lo –

Me. Henrique Pachioni Martins, Me. Patrick Pedreira Silva e Esp. Alex Setolin Beirigo.

Agradeço as bibliotecárias Laudeceia Machado e Valquíria Tavares pelos auxílios referentes à composição estrutural desta obra.

“Sabedoria não depende do intelecto e memória, mas da maturidade, pureza e perfeição da personalidade do indivíduo.”
(BARDON, 2011, p. 51).

RESUMO

A tecnologia está em constante evolução, diminuindo drasticamente o preço de um produto tecnológico, recentemente disponibilizado ao mercado consumidor. Diante desta rápida mudança de valores, o custo necessário para manter equipamentos eletrônicos periodicamente atualizados é alto, por este motivo, as empresas realizam compras de novos produtos em um prazo médio de cinco anos. Com o intuito de amenizar os gastos em adquirir computadores potentes para alcançar um alto desempenho, este trabalho realiza a montagem de um cluster do tipo beowulf, a partir de computadores considerados ultrapassados, com seu custo inferior a novos, com o objetivo de que eles possam trabalhar em conjunto para verificar, perante testes, se eles obtêm resultados superiores ou equivalentes a um único equipamento operando sozinho. Foram utilizados três computadores iguais para, verificar se houve um ganho considerável de processamento, interligados via cabo de rede, conectados a um roteador para haver comunicação constante, operando paralelamente através do protocolo de troca de mensagens MPI, utilizando o sistema operacional Linux Mint. Para medir o processamento foram realizados testes de cinco maneiras diferentes em cada programa utilizado no trabalho (bzip2, zip, rarcrack e povray) gerando tabelas e gráficos comparativos. Perante os resultados obtidos a montagem de um cluster do tipo beowulf, depende da aplicação a ser realizada, onde houve uma grande oscilação de processamento em cada tipo de teste, havendo uma melhora pouco significativa em renderizar imagens, utilizando o programa povray.

Palavras-chave: Benchmarks. Cluster Beowulf. Linux.

Abstract

The technology is constantly evolving, reducing drastically the price of a technological product, recently made available to the consumer market. In the face of this rapid change, the cost required to maintain periodically updated electronic equipment is high, for this reason, companies carry out purchases of new products for an average term of five years. In order to ease the expense in acquiring powerful computers to achieve high performance, this job performs the Assembly of a cluster of type beowulf, from computers considered outdated, with its cost less than new, with the intention that they can work together to check, before tests, if they get superior results or equivalent to a single equipment operating alone. We used three computers equal to verify whether there has been a considerable gain of processing, connected via network cable, connected to a router to be constant communication, operating in parallel via MPI message exchange protocol, using the operating system Linux Mint. To measure the five tests were carried out processing different ways in each program used at work (bzip2, zip, rarcrack and povray) generating comparative tables and charts. Before the results of the Assembly of a cluster of beowulf, depends on the application type to be performed, where there was a large variation in processing each type of test, with a negligible improvement in rendering images, using the program povray.

Keywords: Benchmarks. Cluster Beowulf. Linux.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de Funcionamento do Processador	12
Figura 2 - Tabela de valores e equivalências dos bits.....	13
Figura 3 - Pirâmide de hierarquia das memórias.....	15
Figura 4 - Hierarquia de memórias mostrando capacidade, latência e custo.....	17
Figura 5 - Ligação da memória cache com a CPU e MP.....	17
Figura 6 – Funcionamento da Memória virtual	19
Figura 7 - Classes de rede IP.....	21
Figura 8 - Tipos de cabos de redes	24
Figura 9 - Tabela de versões do SO Linux Mint e seus codinomes	31
Figura 10 - Esquema de sistemas fortemente acoplado	32
Figura 11 - Tabela com diferenças entre sistemas fortemente acoplados	33
Figura 12 - Esquema do projeto	39
Figura 13 – Tabela Compactando Usando Bzip2.....	43
Figura 14 – Gráfico Compactando Usando Bzip2	44
Figura 15 – Tabela Compactando Usando Zip.....	46
Figura 16 – Gráfico Compactando Usando Zip	46
Figura 17 – Tabela Descompactar Usando BZip2.....	48
Figura 18 - Gráfico Descompactar Usando BZip2.....	49
Figura 19 - Tabela Descompactar Zip	50
Figura 20 - Gráfico Descompactar Usando Zip	51
Figura 21 - Tabela Copiar Arquivo	53
Figura 22 - Gráfico Copiar Arquivo.....	53
Figura 23 - Tabela Rarcrack.....	55
Figura 24 - Gráfico Rarcrack	56
Figura 25 - Imagem Renderizada	58
Figura 26 - Tabela Povray	59
Figura 27 - Gráfico Povray	59
Figura 28 - Tela de boot do Linux Mint.....	65
Figura 29 - Tela inicial do Linux Mint.....	66
Figura 30 - Requisitos necessários para instalação do Linux Mint.....	67
Figura 31 - Tipos de instalação do Linux Mint.....	68
Figura 32 - Particionamento do disco rígido.....	69

Figura 33 - Seleção de fuso horário	70
Figura 34 - Escolha do layout do teclado	71
Figura 35 - Identificação do computador	72
Figura 36 - Instalação do sistema operacional	73
Figura 37 - Atualização de programas.....	74
Figura 38 - Término da instalação.....	75
Figura 39 - Carregamento Linux Mint.....	76
Figura 40 - Sistema iniciado pela primeira vez.....	77
Figura 41 - Abrindo o navegador.....	78
Figura 42 - Abrindo o terminal	79

LISTA DE ABREVIATURAS E SIGLAS

AT&T - American Telephone and Telegraph

Bit - Binary digit

Byte - 8 Bits

CD - Compact Disk

CPU - Central Processing Unit

DVD - Digital Versatile Disc

E/S - Entrada e Saída de informações no computador

FSF - Free Software Foundation

GPL - General Public License

HD - Hard Disk

Hertz / Hz - 1 Bit por segundo

IP - Internet Protocol

ISA - Instruction Set Architecture

L - Level

LAN - Local-area networks (Rede Local)

MB - Megabytes

Megahertz / MHz - Milhões de pulsos por Segundo

GB - Gigabytes

Gigahertz / Ghz - Bilhões de pulsos por segundo

MIMD - Multiple Instruction Multiple Data

MP - Memória Principal

MPI - Message Passing Interface

NASA - National Aeronautics and SpAce

NFS – Network File System

RAM - Random Access Memory

ROM - Read-Only Memory

SO - Sistema Operacional

STP - Shielded Twisted-Pair

TCP - Transmission Control Protocol

UAL - Unidade Aritmética e Lógica

UTP - Unshielded Twisted-Pair

UC - Unidade de Controle

SUMÁRIO

1	INTRODUÇÃO	8
2	OBJETIVOS	10
2.1	OBJETIVO GERAL	10
2.2	OBJETIVOS ESPECÍFICOS	10
3	HARDWARE	11
3.1	PROCESSADOR.....	11
3.2	TIPOS DE MEMÓRIA.....	14
3.2.1	Registrador.....	15
3.2.2	Memória Cache	16
3.2.3	Memória Secundária.....	18
3.2.4	Memória Virtual.....	18
3.3	SOFTWARE	20
3.4	REDE LOCAL	20
3.4.1	Ethernet	22
3.4.2	Roteador	24
3.4.3	TCP/IP	24
3.5	SISTEMA OPERACIONAL	25
3.5.1	Kernel	26
3.6	ARQUITETURA DE COMPUTADORES.....	26
3.7	UNIX	27
3.7.1	Projeto GNU	28
3.7.2	Linux	29
3.7.3	Código fonte.....	30
3.7.4	Linux Mint.....	30
3.8	SISTEMAS FORTEMENTE E FRACAMENTE ACOPLADOS.....	31
3.9	MPI	33
3.10	CLUSTER.....	34
3.10.1	Cluster Beowulf	34
3.12	FERRAMENTAS	35
3.12.1	Benchmark	35
3.12.2	Rarcrack	36
3.12.3	SSH.....	36

3.12.4	NFS	36
3.12.5	Povray	37
3.12.6	Compactadores	37
4	TRABALHOS CORRELATOS	38
5	METODOLOGIA.....	39
6	ANÁLISE BENCHMARKS	42
6.1	COMPACTANDO COM BZIP2.....	42
6.2	COMPACTANDO COM ZIP	44
6.3	DESCOMPACTANDO COM BZIP2	47
6.4	DESCOMPACTANDO COM ZIP	49
6.5	COPIAR ARQUIVOS.....	51
6.6	QUEBRA DE SENHAS USANDO RARCRACK	54
6.7	RENDERIZAR IMAGEM USANDO POVRAY	56
7	CONSIDERAÇÕES FINAIS	60
	REFERÊNCIAS.....	61
	APÊNDICE A – INSTALAÇÃO DO SISTEMA OPERACIONAL	65
	APÊNDICE B – CONFIGURAÇÃO DO NÓ SERVIDOR	78
	APÊNDICE C – CONFIGURAÇÃO DOS NÓS ESCRAVOS	85

1 INTRODUÇÃO

A tecnologia está evoluindo de maneira acelerada realizando constantemente o aprimoramento dos produtos, esta mudança é perceptível, ao comprar um item eletrônico qualquer, lançado há poucos dias no mercado, em questão de um curto período de tempo, este objeto adquirido tem seu valor degradado, sendo esse o preço pago a esta evolução tecnológica diária. (CHINARELLI, [c2012]).

Antigamente as tecnologias de ponta estavam restritas a serem utilizadas apenas em âmbito corporativo. Atualmente a população almeja fazer a adoção destas ferramentas para uso pessoal, equiparando assim a tecnologia empresarial e caseira, causando o fenômeno deste desenvolvimento acelerado, a onde os equipamentos que estão surgindo são voltados, primeiramente para os consumidores. Os custos para periodicamente obter um equipamento de última geração, são elevados, impossibilitando fazer a renovação de todos os componentes, para se manter atualizado, em meio a esta situação, as empresas fazem a compra de novos produtos, em um prazo estipulado na média de cinco anos. (RAPIDA..., [c2011]).

Devido aos custos elevados em possuir computadores potentes, para se alcançar um alto desempenho, é interessante verificar se é possível obter, resultados equivalentes ou superiores optando por fazer os computadores trabalharem em conjunto, realizando a utilização de clusters, a partir de máquinas ultrapassadas. (ALECRIM, [c2013]).

O processo de evolução tecnológica faz com que computadores antigos, percam sua função, diminuindo bruscamente seu desempenho que, um dia foi suficiente para realizarem suas ações. Os computadores novos possuem preços altos, tornando-se um peso no fator custo versus desempenho. Apesar de satisfazer o desempenho, seu custo é elevado, sendo um fator decisivo no momento da compra. Dependendo do objetivo a ser satisfeito, reduzir os gastos, adquirindo várias CPUs, para funcionarem como uma única, alcançando um desempenho maior, trabalhando em conjunto, pode se tornar uma alternativa.

Com base neste contexto este projeto propõe a criação de clusters do tipo beowulf, para empregar um valor a computadores considerados fracos, a fim de demonstrar por meio de testes, se obterão um desempenho superior, comparado a um único computador trabalhando sozinho, podendo ser uma alternativa relevante

comprar vários computadores de custo inferior, alcançando um alto desempenho, do que um computador novo, com maior custo, para obter o desempenho desejado.

2 OBJETIVOS

Apresenta-se nos tópicos a seguir o objetivo geral e os objetivos específicos da pesquisa.

2.1 OBJETIVO GERAL

Montar um cluster do tipo beowulf, realizando testes para demonstrar o desempenho obtido.

2.2 OBJETIVOS ESPECÍFICOS

- a) instalar o sistema operacional GNU/Linux Mint em três microcomputadores;
- b) configurar os computadores para operarem em um cluster do tipo beowulf, sendo um destes computadores, o servidor e os demais escravos;
- c) interligar os computadores em rede;
- d) executar softwares Benchmarks, para analisar o processamento individual e concorrente dos processadores;
- e) fazer uma comparação, demonstrando por tabelas e gráficos os resultados realizados, através de testes de benchmark.

3 HARDWARE

A origem da palavra hard, vem do idioma inglês, identificando-se por toda e qualquer ferramenta manual, é em si um elemento de difícil modificação, denominado ser qualquer material físico. (MONTEIRO, 2007).

Segundo Aguilar (2008), o usuário ao interagir com o computador, proporciona uma entrada de informações, então o computador processará estas informações, e retornará uma saída (resultado), retornando a resposta, estes dispositivos que fornecem uma entrada ou recebem uma saída também são difundidos pela sigla E/S, estes poderão possuir as seguintes funcionalidades:

- a) armazenar a informação para processá-la;
- b) executar uma sequência de operações para receber a entrada ou retornar uma saída;
- c) controlar, monitorar e certificar que as operações serão feitas em determinada sequência;
- d) apresentar o resultado em uma saída referente às informações das entradas. (AGUILAR, 2008).

Segundo Monteiro (2007) resumindo de maneira simplificada, todos os componentes que estiverem conectados ao computador, seja interno ou externamente, são hardwares, capazes de prover uma entrada ou saída de informações.

3.1 PROCESSADOR

O processador, normalmente é chamado de unidade central de processamento ou mais comum, a sigla do inglês CPU é um dispositivo que fica localizado na parte interna do computador, responsável por executar as instruções do programa, do termo inglês run muitas vezes traduzido por rodar, executar e funcionar. (AGUILAR, 2008).

Para entender a organização da CPU, devemos considerar as ações que ela deve executar:

- **Busca de instrução:** a CPU lê uma instrução da memória.
- **Interpretação de instrução:** é decodificada para determinar a ação requerida.
- **Busca de dados:** a execução de uma instrução pode requerer leitura de dados da memória ou de um módulo de E/S.
- **Processamento de dados:** a execução de uma instrução pode requerer efetuar uma operação aritmética ou lógica sobre os dados.
- **Escrita de dados:** os resultados da execução podem requerer escrever dados na memória ou em um módulo de E/S. (STALLINGS, 2002, p. 429, grifo do autor).

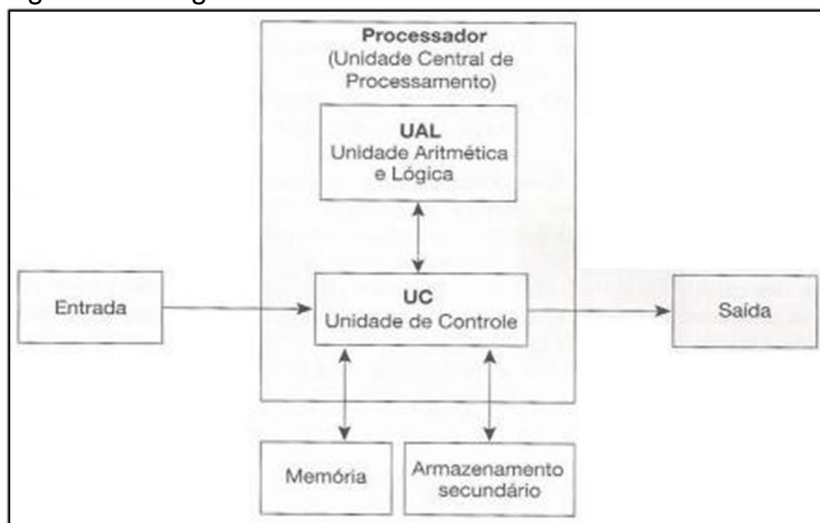
Segundo Aguilar (2008), a CPU controla o processamento de informação, seja armazenando na memória ou recuperando dela, podendo guardar estes resultados realizados na memória para uso posterior.

O processador é a união de dois componentes menores:

- a) unidade de Controle, é encarregado de realizar a sincronização de todo o processamento da unidade central de processamento, coordenando a ordem de execução das instruções;
- b) unidade Aritmética e Lógica faz operações aritméticas e lógicas, realizando comparações, além de efetuar operações matemáticas básicas: soma, divisão, multiplicação e subtração, alterando e armazenando dados na memória. (AGUILAR, 2008).

O diagrama da Figura 1 demonstra o funcionamento interno do processador:

Figura 1 – Diagrama de Funcionamento do Processador



Fonte: Aguilar (2008, p. 5).

O computador é um equipamento eletrônico que realiza funções internamente sob a forma eletrônica, se comunicando com os componentes interligados a ele em meio a voltagem ou corrente elétrica. Para essa máquina representar eletricamente os símbolos da linguagem humana, foi desenvolvida sua própria estrutura de informação para se comunicar e agir. (MONTEIRO, 2007).

A unidade de informação de um sistema digital em um circuito de computador, é denominado dígito binário abreviado do inglês bit, este pode estar em um estado de ligado = 1 ou desligado = 0. Em 1964 os projetistas de computador da empresa IBM, estabeleceram o uso de grupos de 8 bits (2^8), como a unidade de memória do computador, assim 8 bits = 1 byte. (NULL; LOBUR, 2010).

A Figura 2 exemplifica estes valores:

Figura 2 - Tabela de valores e equivalências dos bits

Byte	Byte (B)	<i>equivale a</i>	8 bits
Kilobyte	Kbyte (KB)	<i>equivale a</i>	1.024 bytes (2^{10})
Megabyte	Mbyte (MB)	<i>equivale a</i>	1.024 Kbytes (2^{20})
Gigabyte	Gbyte (GB)	<i>equivale a</i>	1.024 Mbytes (2^{30})
Terabyte	Tbyte (TB)	<i>equivale a</i>	1.024 Gbytes (2^{40})
Petabyte	Pbyte (PB)	<i>equivale a</i>	1.024 Tbytes (2^{50})
Exabyte	Ebyte (EB)	<i>equivale a</i>	1.024 Pbyte (2^{60})
Zetabyte	Zbyte (ZB)	<i>equivale a</i>	1.024 Ebyte (2^{70})
Yota	Ybyte (YB)	<i>equivale a</i>	1.024 Ybytes (2^{80})
1 Tb = 1.024Gb Gb = 1.024Mb Mb = 1.048.576b Kb = 1.073.741.824 b			

Fonte: Aguilar (2008, p. 10).

O computador organiza em grupos ordenados de bits para possuírem um significado, estes representam uma informação, sendo um carácter para a máquina, para nós humanos é equivalente a uma linguagem de comunicação. Todo carácter é armazenado e convertido em um conjunto de bits, a palavra carácter é utilizada em fins comerciais, enquanto byte é usado na linguagem técnica. (MONTEIRO, 2007).

O processador é o cérebro do computador, pois ele quem executa as instruções do programa. Cada sistema de computação contém um relógio que mantém o sistema sincronizado, o relógio envia pulsos elétricos a todos os componentes. O número de pulsos emitidos a cada segundo pelo relógio se denomina frequência. As frequências são medidas em ciclos por segundo chamados de hertz. (NULL; LOBUR, 2010).

A velocidade do relógio é a quantidade de bits, que serão transferidos, tratando-se de um objeto físico, é medido em Hertz, que são as ondulações eletromagnéticas dos sinais, assemelha-se $1 \text{ Hz} = 1 \text{ bit por segundo}$. Nos processadores é utilizada a inicial do nome da unidade equivalente, complementando com Hz para demonstrarmos a velocidade do processador, o termo bytes é utilizado para referenciar a memória. (MONTEIRO, 2007).

Assim, gerando milhões de pulsos por segundo, dizemos que operam em intervalos de Megahertz, já operando em bilhões de pulsos por segundo dizemos que opera em Gigahertz. (NULL; LOBUR, 2010).

Um exemplo para melhor compreensão é dizer que um processador tem 100 Mhz, este será equivalente a (2^{20}) ou seja 1.048.576 de pulsos por segundo. (MONTEIRO, 2007).

3.2 TIPOS DE MEMÓRIA

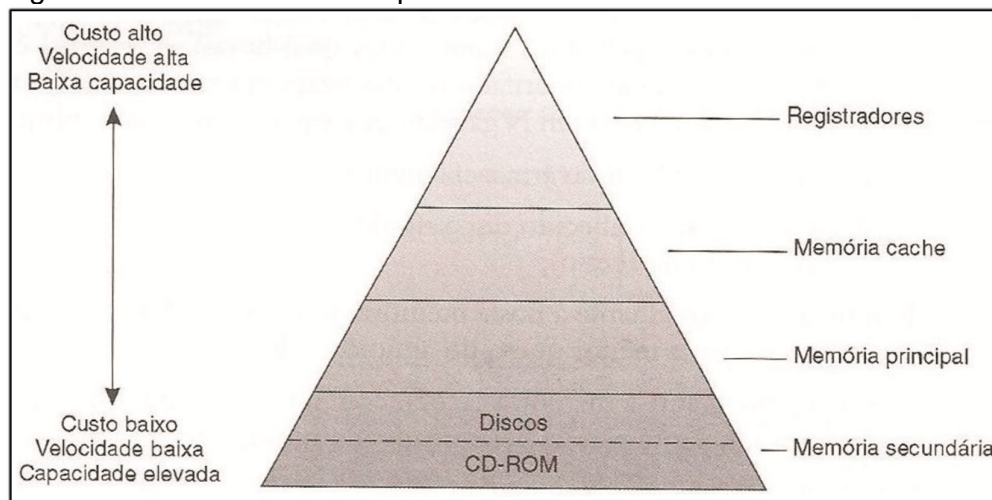
Conforme Aguilar (2008), a memória de um computador pode possuir milhões de bytes, estes bytes são a unidade elementar de armazenamento, é utilizado múltiplos termos para definir o seu tamanho (Figura 2).

Uma das características da memória é ser ou não volátil. A memória volátil não armazena qualquer informação quando, há ausência de energia, enquanto a memória não volátil guarda seus valores, para utilização futura. Dentre suas características se destaca a velocidade de acesso, quantidade de informações que podem ser armazenadas e seu custo. (ENGLANDER, 2011).

Este componente é dividido em vários dispositivos, com velocidades e localizações diferentes, servindo ao mesmo propósito, armazenar e recuperar informações. (MONTEIRO, 2007).

Na Figura 3 está uma pirâmide, apresentando os tipos de memórias hierarquicamente, começando pela memória mais rápida para a mais lenta.

Figura 3 - Pirâmide de hierarquia das memórias



Fonte: Monteiro (2007, p. 86).

A finalidade de existir diferentes tipos de memórias é manter uma velocidade compatível com a UC, são divididas em duas classes, RAM, a qual as especificações do computador se referem e ROM. (NULL; LOBUR, 2010).

As memórias do tipo ROM, são apenas para leitura, seu conteúdo não pode ser alterado, nela estão armazenados programas estáticos, como os sistemas para o hardware funcionar. (SILBERSCHATZ; GALVIN; GAGNE, 2010).

A memória principal, também chamada de memória primária ou RAM, trabalha em constante conjunto com o processador, com o objetivo de armazenar as instruções do programa que está sendo executado, o processador vai procurando os dados na memória para processar as informações. (MONTEIRO, 2007).

3.2.1 Registrador

Registro ou registrador é o local de armazenamento único e permanente dentro da CPU, armazena temporariamente uma informação para, ser manipulada diretamente pela UC durante a execução de instruções, é utilizado de maneiras distintas pelo computador, podendo conter dados que estão sendo processados, guardando uma instrução que está sendo executada e até códigos binários. (ENGLANDER, 2011).

Por ser construído com a mesma tecnologia do processador, necessita de energia. Sendo uma memória volátil, sua velocidade é superior às demais memórias,

com um tamanho pequeno suficiente para armazenar um único dado ou instrução, variando de 8 a 128 bits normalmente. (MONTEIRO, 2007).

3.2.2 Memória Cache

Segundo Parhami (2008), para acompanhar a velocidade do processador, armazenando instruções e dados em tempo equivalente, não é possível guardar muita informação, devido a velocidade do processador, ficando a cargo de memórias com tempo de acesso inferior, como a MP, armazenar o restante do programa, enquanto esta memória de acesso rápido chamada de cache apenas armazena os dados que são utilizados com maior frequência, ou que serão usados na próxima instrução.

Conforme Parhami (2008), a origem de sua palavra vem do francês *cache* que significa esconder, a ideia básica desta memória para a CPU é analisar em primeiro lugar, se a informação que procura está armazenada se não estiver, irá para a memória principal, a vantagem é o grande tempo de acesso, porque a memória principal, não acompanha a CPU, para o processador acessá-la sofre um grande retardo.

O motivo das memórias não serem inteiramente caches, acompanhando o processador, havendo apenas um retardo imperceptível e aumentando muito o desempenho da CPU em geral, é seu alto custo, este fator limita a construção dos componentes, sendo obrigados a dividir em memórias mais lentas. A figura 4, mostra a divisão das memórias, separando hierarquicamente por velocidade de acesso, considerando a constante evolução tecnológica foi incrementada uma memória terciária, que terá capacidade de armazenamento muito maior que a secundária, porém mais lenta. (PARHAMI, 2008).

Figura 4 - Hierarquia de memórias mostrando capacidade, latência e custo

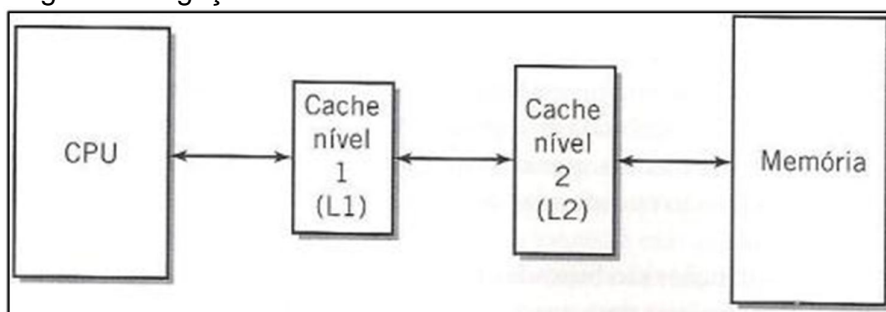


Fonte: Parhami (2008, p. 330).

Com o intuito de diminuir ainda mais o atraso no acesso as memórias, foram projetados níveis para a cache, com a introdução de várias caches, aumentará sua efetividade, separando maior quantidade de instruções, diminuindo o acesso a MP. Como foi criado na linguagem inglesa, a referência dos níveis da cache se dá a sua inicial L do inglês level (nível). (ENGLANDER, 2011).

A Figura 5, mostra como é feita a ligação entre a memória principal e o processador em meio as caches.

Figura 5 - Ligação da memória cache com a CPU e MP



Fonte: Englander (2011, p. 21).

Antigamente era um componente de uma placa, hoje está situado dentro do processador, existindo mais de dois níveis de cache, atualmente CPUs comportam caches de nível três, para antecipar cada vez mais requisições futuras na memória principal. (ENGLANDER, 2011).

3.2.3 Memória Secundária

A grande quantidade de dados armazenados na MP, dificulta a utilização, para solucionar esta questão, as informações usadas com menor frequência ou com necessidade de serem guardadas para uso posterior, são alocadas em dispositivos distintos da memória principal, podendo ser fitas, discos, CD's ou DVD's, aguardando qualquer solicitação de dados para ser usados novamente. (WEBER, 2012).

Segundo Monteiro (2007), esta memória também pode ser chamada de memória auxiliar ou memória de massa, sua maior característica é ser um dispositivo não volátil, entre todos os tipos de memória, possui a maior quantidade de armazenamento, seu tempo de acesso é inferior comparado as outras, porque este dispositivo de armazenamento é uma peça e não um circuito.

Existem dois tipos de dispositivos de armazenamento, uns estão ligados diretamente ao sistema, para acesso imediato as informações e outros são conectados quando houver a necessidade de acesso. Dentre os discos para utilização posterior estão o CD-ROM e DVD, para uso imediato está o disco rígido, muitas vezes chamado pelo seu nome em inglês HD, este é normalmente particionado (dividido) em um formato para, utilização do SO e arquivos do usuário. (AGUILAR, 2008).

A causa do HD ser magnético é sua característica de manter armazenado os dados magneticamente no disco, quando não houver energia, assim alocando as informações permanentemente. (MONTEIRO, 2007).

3.2.4 Memória Virtual

Todos os programas necessitam de memória para funcionarem, o problema, se encontra na grande quantidade de softwares abertos em estado ocioso, isto sobrecarrega a memória principal. Para resolver este problema, todo e qualquer programa que não esteja em uso, será deslocado para a memória virtual. (HENNESSY; PATTERSON, 2008).

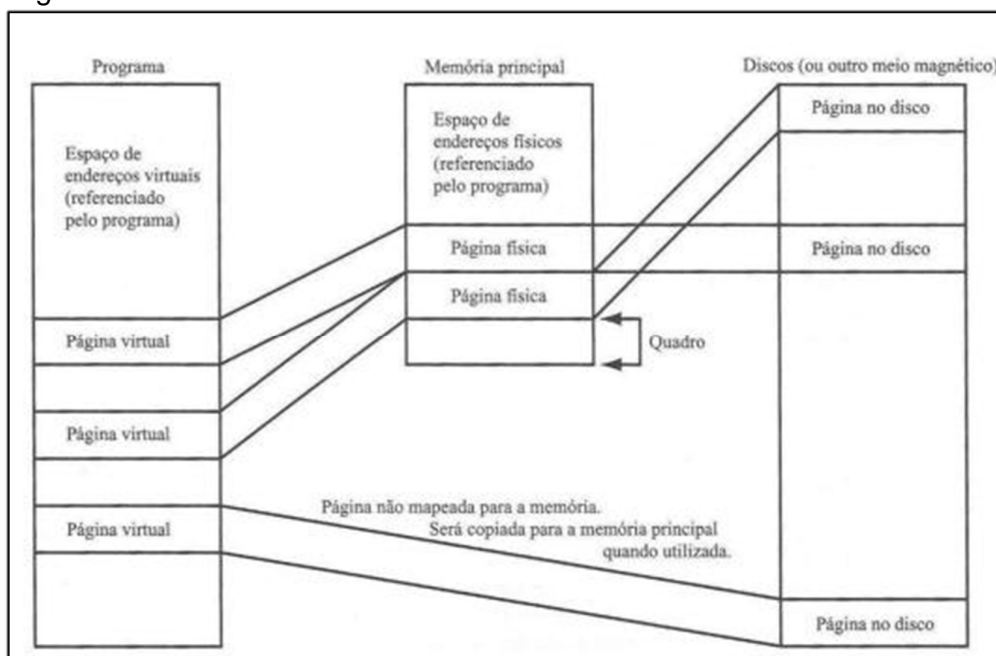
Segundo Carter (2003), a memória virtual consiste em armazenar informações, alocando um espaço exclusivo na memória secundária, para uso

posterior destes dados, normalmente é utilizado os discos rígidos. A uma exceção dos programas não poderem ficar inteiramente executados na memória virtual, devido a lentidão provocada pelo acesso, pois a memória principal é mais rápida que um dispositivo de memória secundária.

A execução do programa é separada em endereços, quando o software estiver em execução será aberta uma página para referenciar este endereço, na memória principal, se ocorrer do software ficar ocioso é aberto uma página na memória virtual, e quando for exigido novamente a informação ela se conectará com o programa, assim a memória virtual se torna uma extensão da memória principal, agilizando seu uso. (CARTER, 2003).

A Figura 6 demonstra o processo descrito acima.

Figura 6 – Funcionamento da Memória virtual



Fonte: Carter (2003, p. 182).

Em sistemas operacionais GNU/Linux, é reservado um espaço do disco rígido para a memória virtual chamado de swap, está área de troca entre a HD e a memória, aumenta o desempenho do sistema, por reservar em uma partição, ao invés de ficar incluso ao sistema operacional, como outros sistemas, diminuindo o tempo de acesso. (VOLKERDING; JOHNSON; REICHARD, 1998).

3.3 SOFTWARE

Segundo Tanenbaum (2010), o Software é basicamente um programa que consiste em algoritmos, que são instruções detalhadas de como realizar determinada ação, estas podem ser guardadas no computador, a definição se limita em uma lista de procedimentos a cumprir e não o local de seu armazenamento.

Monteiro (2007), explica que o software, mostra a “inteligência” do computador, manipulando a máquina (hardware) a seguir uma sequência de instruções (programa). Para determinar uma função para o computador exercer é considerado fácil, derivado do termo do inglês soft, em oposição ao hardware que, para se acrescentar um componente, ao computador é preciso fabricá-lo. Cada maneira distinta de ordenar os componentes a agirem, significa ser um programa diferente que está sendo executado, podendo ou não utilizar o mesmo hardware.

3.4 REDE LOCAL

É um grupo de dispositivos ou normalmente computadores interconectados, por um canal de comunicação, podendo compartilhar arquivos, recursos e usuários, denominado LAN. (STALLINGS, 2005).

As LANs são limitadas, em uma área geográfica, um exemplo é serem comumente utilizadas em campus universitários e edifícios inteiros, havendo a necessidade de grandes conexões locais, necessitará de várias LANs, como é o caso de grandes empresas, dentre as redes locais, a mais popular é a ethernet. (FOROUZAN, 2008).

Segundo Tanenbaum (2003), o conjunto de redes conectadas de dois ou mais computadores, é chamada de inter-rede ou internet (com a letra i em minúsculo), o motivo é que a “Internet”, não se compara com uma simples rede que conecta poucos computadores, mas um vasto conjunto de redes interligadas, é um sistema que não pode ser planejado nem controlado por ninguém, é o meio que interliga todo o mundo por uma conexão.

O meio que cada dispositivo é conectado a internet é chamado de interface de rede, para cada interface a um endereço de IP para codificar o número na rede e o número do host (computador ligado a rede), a rede é controlada por uma empresa e o host pelo usuário. Todos os endereços de IP possuem 32 bits, os IPs fazem

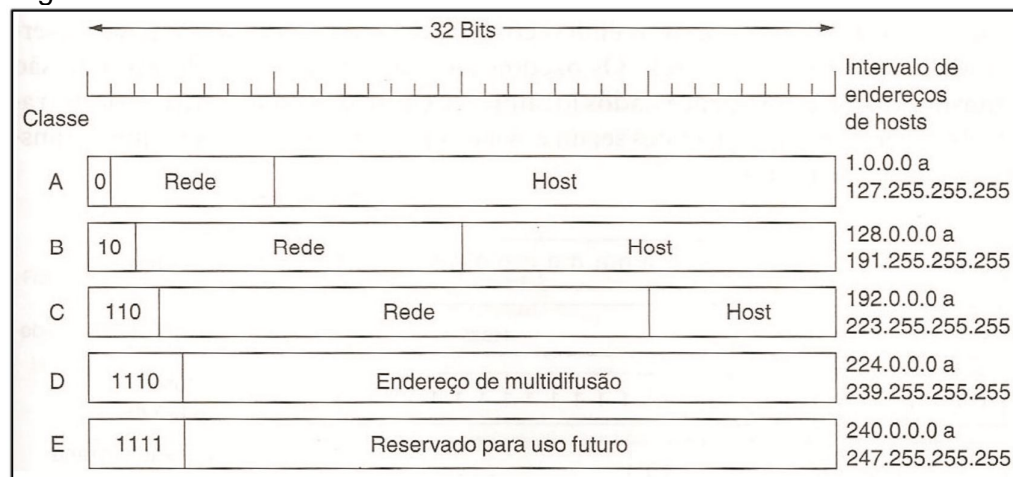
referência a interface de rede e não ao host, assim se um usuário estiver em duas redes, precisaria de dois endereços de IP. (TANENBAUM, 2003).

Segundo Tanenbaum (2003), estes endereços foram divididos em cinco categorias, esta divisão é chamada de endereçamento de classe completo, o formato destas classes é a seguinte:

- a) **classe A** - permite 128 redes e cada rede com até 16 milhões de hosts;
- b) **classe B** - permite 16 mil redes e cada rede com até 65536 hosts;
- c) **classe C** - permite 2 milhões de redes e cada rede com 256 hosts;
- d) **classe D** – admite a multidifusão na qual um datagrama (conexão não confiável) é direcionado a vários hosts;
- e) **classe E** – Está reservada para uso futuro. (TANENBAUM, 2003).

A Figura 7 mostra as classes de redes e suas respectivas informações de IP.

Figura 7 - Classes de rede IP



Fonte: Tanenbaum (2003, p. 465).

Conforme Tanenbaum (2003), os endereços de IP são números de 32 bits escritos em notação decimal com pontos, cada um dos 4 bytes, é escrito de 0 a 255, o endereço mais baixo é 0.0.0.0 e o mais alto é 255.255.255.255. Os números 0 e 1 possuem funções específicas, 0 refere-se a rede atual, fazendo os dispositivos conectados com este número referenciar a sua própria rede, sem saber ao certo seu IP, basta saber a que pertence classe de rede para, informar a quantidade de zeros

que será incluso. Já o número 1 permite a difusão na rede local, normalmente é uma LAN, os endereços com apenas 1 no campo de host permite que as máquinas façam difusão em LANs distantes.

Para os dispositivos se comunicarem devem estar com o mesmo número de rede, na medida que as redes crescem podem gerar problemas, pois a um limite de números de IPs estes são facilmente alcançados pela grande quantidade de hosts, para solucionar este problema, será preciso que a rede seja dividida, em diversas partes para seu uso, continuando a funcionar como uma única rede. Basicamente, um único endereço de IP qualquer, poderá criar várias sub-redes, para implementar a divisão, será preciso uma máscara de rede, indicando que a rede foi dividida, estas também são escritas em notação decimal com pontos, a divisão de sub-redes não é visível, aos usuários. (TANENBAUM, 2003).

3.4.1 Ethernet

Segundo Oliveira e Andrade (2012), a tecnologia Ethernet é a interconexão da LAN, sua característica principal é baseada na transmissão de pacotes, decompondo vários arquivos de diversos tamanhos, em frações, para serem enviados, através desta conexão.

A placa de rede é um periférico do computador, responsável por conectar os computadores a rede, através do cabo apropriado. (VASCONCELOS, 2007).

A tecnologia foi desenvolvida pela empresa Xerox em 1976, para interligar os periféricos, são utilizados três tipos de cabos; Coaxiais, Fibra ótica e Par trançado. (DYSON; COLEMAN; GILBERT, 1999).

Par trançado consiste em dois fios de cobre isolados, organizados em espiral, um par de fios age como um único enlace de comunicação, diversos pares de enlaces estão em um cabo, em meio a uma manta protetora. Dentre os demais tipos de cabos, este é o mais fácil de trabalhar em relação ao baixo custo, porém possui sérios problemas relacionados a interferência, é o meio de comunicação inferior a todos os demais, tanto em velocidade quanto em distância, atualmente é o mais utilizado. (STALLINGS, 2005).

Conforme Barbosa (c2011), existem dois tipos de cabo de par trançado, um pode ser revestido com uma camada de cobre, protegendo o fio, chamado por sua

sigla STP e o segundo é um simples cabo envolto de uma camada plástica também conhecido por sua sigla, UTP.

Segundo Barbosa (c2011), o cabo de par trançado é dividido nas seguintes categorias:

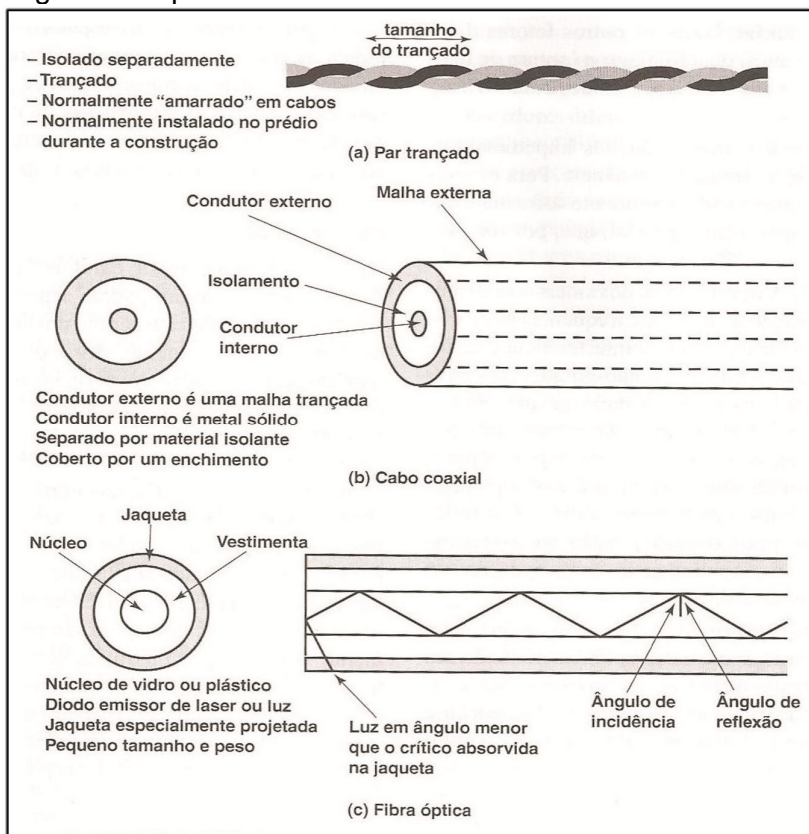
- a) **categoria 1:** cabos telefônicos que apenas transmitem voz, padrão utilizado em telefonia anterior a 1983;
- b) **categoria 2:** cabo com velocidade de 4 Mbps, contendo quatro pares trançados;
- c) **categoria 3:** cabo com velocidade de 10 Mbps, contendo quatro pares trançados com nove torções por metro;
- d) **categoria 4:** cabo com velocidade de 16 Mbps, contendo quatro pares trançados;
- e) **categoria 5:** cabo com velocidade de 100 Mbps, contendo quatro pares trançados de fio de cobre;
- f) **categoria 5e e 6:** cabo com velocidade de Gigabit, contendo quatro pares trançados de fio de cobre.

Segundo Stallings (2005), o cabo Coaxial consiste em um condutor cilíndrico envolvendo um único fio, esta camada que protege o fio é revestida por uma malha, protegendo de interferências. Podendo ser usado em distâncias maior que o par trançado, era o meio tradicional de conectar as LANs, caiu em desuso devido a seu custo alto, mesmo possuindo maior capacidade.

A fibra ótica consiste em diversos vidros ou plásticos que criam uma fibra ótica, em meio a um fio cilíndrico, dividido em três camadas concêntricas, as duas internas possuem diferentes índices de refração, o centro se chama núcleo, estas camadas são cobertas por uma jaqueta protetora que absorve a luz, as fibras óticas são agrupadas em cabos óticos, é o meio de comunicação que possui maior velocidade de transmissão. (STALLINGS, 2005).

A Figura 8 demonstra os tipos de cabos de rede, comentados anteriormente.

Figura 8 - Tipos de cabos de redes



Fonte: Stallings (2005, p. 190).

3.4.2 Roteador

O roteador é um dispositivo utilizado para conectar redes independentes, criando a interligação das redes ou uma grande rede, tendo a função de direcionar as informações a seus destinos. (FOROUZAN, 2008).

Segundo Stallings (2005), o roteador deve fornecer as informações sem qualquer alteração, independentemente do equipamento conectado, além de certificar que os dados foram transmitidos na mesma ordem de envio, para garantir sua confiabilidade, o protocolo mais utilizado para essa funcionalidade é o TCP.

3.4.3 TCP/IP

O TCP verifica se todas as informações enviadas foram recebidas integralmente e na ordem correta, se ocorrer alguma perda, o software do TCP se certificará que seja retransmitido, até chegar a o seu receptor. (TANENBAUM, 2010).

O Internet Protocol é um protocolo sem conexão e não confiável, a característica desse serviço é enviar pelo melhor caminho, o problema é que, não assume a confiabilidade, ausentando-se de verificar e monitorar quaisquer erros, não garantindo que as informações cheguem ao destino. Para assegurar a confiança, foi combinado com o TCP. Uma analogia dessa união para, fácil entendimento é supor que o protocolo IP seja uma agência de correios, está fará o possível para entregar a correspondência da melhor forma possível, mas nem sempre conseguirá. Se um pedido não for registrado, será perdido, ficando por conta do remetente no nosso caso o TCP, descobrir a perda e resolver o problema, para que chegue ao seu destino. (FOROUZAN, 2008).

3.5 SISTEMA OPERACIONAL

Segundo Shay (1996), um sistema operacional é um software responsável por receber instruções fornecidas pelo usuário, instruindo o hardware a executar as tarefas solicitadas. Este programa controla outros softwares residentes na memória, supervisionando os demais executados no computador para, obter um desempenho eficiente, a partir da utilização dos recursos do hardware.

Segundo Weber (2012), o SO exerce ações imperceptíveis ao usuário, realizando a todo instante, três funções básicas:

- a) executa programas, organizando a sequência de operações sobre os serviços, controlando os programas de acordo com sua prioridade e seu tempo de execução;
- b) administra os recursos do computador, alocando corretamente na memória os programas que estiverem em execução, disponibiliza a utilização do hardware, em demanda a necessidade dos programas para, não haver conflitos em seu uso;
- c) gerência os dados, controlando a utilização dos meios de armazenamento dos softwares e dados do usuário;

3.5.1 Kernel

Kernel é o núcleo do sistema operacional, o programa ou uma coleção de programas mais importante dentro de um computador, porém não é visto nem tampouco utilizado diretamente pelos usuários em sua maioria. (SOARES; FERNANDES, 2013, p. 28).

Já Morimoto (2009, p. 48) explica esta afirmação: “O kernel é a peça fundamental do sistema, responsável por prover a infraestrutura básica necessária para que os programas funcionem, além de ser o responsável por dar suporte aos mais diferentes periféricos [...]”.

Além disso, Soares e Fernandes (2013) complementam dizendo que o objetivo do núcleo é criar uma conexão entre o hardware e o software, conectando os programas quanto os periféricos, conforme as instruções forem requeridas, pois o SO nada realiza sem os programas, estes tem a função de interagir direto no núcleo, para o sistema operacional reconhecer todo o hardware e iniciar o programa, esperando o software solicitá-lo novamente.

3.6 ARQUITETURA DE COMPUTADORES

Arquitetura de computadores faz referência aos atributos visíveis de um sistema, como o conjunto de instruções a ser seguido, número de bits usados para representar um dado, mecanismos de entrada e saída, e endereçamento de memória. Todos estes atributos tem um impacto na execução lógica do programa. (STALLINGS, 2002).

A arquitetura de computadores, é a combinação do hardware com o software a ser executado, é chamada de ISA, são definidas por diferentes fatores, como o espaço que o software precisa, a complexidade do programa, em relação a sua dificuldade de decodificação para executar uma instrução, assim como as tarefas realizadas e o número total de instruções. (NULL; LOBUR, 2010).

3.7 UNIX

A origem deste SO remonta ao fim década de 1960 no laboratório da companhia American Telephone and Telegraph em Nova Jersey, trabalhando com o Massachusetts Institute of Technology, os laboratórios Bell da AT&T, estavam desenvolvendo um sistema operacional denominado Multics com Ken Thompson, Dennis Ritchie, Brian Kernighan dentre outras pessoas do Computer Science Research Group. (TAYLOR; ARMSTRONG, 1998).

Segundo Taylor e Armstrong (1998), no ano de 1969 a equipe de desenvolvimento estava cada vez mais desmotivada com o Multics, por ser um sistema lento e caro para computadores de grande porte, tornando-se facilmente obsoleto.

Thompson e Ritchie se motivaram em criar o sistema, devido a um jogo desenvolvido para o Multics, incompatível com o PDP-7 (computador), então, criaram a partir de um código base esse jogo para o PDP-7, originando posteriormente, o UNIX. Realizando esta reutilização de código ganharam grande popularidade, pois até o momento, os softwares eram projetados para aplicações específicas, desde então os programas passaram a ser desenvolvidos independentemente do computador, com esta portabilidade, os fabricantes de softwares alteraram seus programas para, serem compatíveis com o UNIX. (SOARES; FERNANDES, 2013).

O UNIX evoluiu durante a década de 1970, passando a ser usado em cada vez mais equipamentos e ganhando mais recursos. Quase sempre ele era usado em aplicações "sérias", incluindo instalações militares, bancos e outras áreas onde não existe margem para falhas. Devido a tudo isso, o sistema se tornou muito robusto e estável. (MORIMOTO, 2009, p. 14).

A filosofia deste sistema é a modularidade, sua utilização será para diversas finalidades, não somente a uma explícita. Dos primórdios até hoje, os melhores sistemas operacionais são os baseados em UNIX, denominados UNIX-like, são os mais antigos, difundidos além de serem confiáveis. O surgimento de sistemas UNIX-like vieram em decorrência das documentações do UNIX que eram distribuídas e explicadas, passo a passo seu código, estas documentações foram disponibilizadas até a versão 6, a partir desta seu código-fonte foi fechado, não disponibilizando abertamente sua documentação. Por não ser um sistema gratuito foi um grande

incentivo a desenvolvedores criarem um SO parecido com as mesmas finalidades, com base nas documentações anteriores dos UNIXs. (SOARES; FERNANDES, 2013).

3.7.1 Projeto GNU

Segundo Soares e Fernandes (2013), apesar de ser um animal, este projeto não tem qualquer ligação com o bicho, apenas uma ironia para dizer que “GNU não é UNIX”. A cabeça deste ser, se tornou o símbolo do projeto para simbolizar a liberdade, está, é presente em todos os animais selvagens, tomando suas próprias decisões sem qualquer limitação, assim todos os programas gratuitos são distribuídos com seu código-fonte aberto, são livres como um animal, ao contrario do UNIX que possui fins lucrativos, sendo limitado em alguns aspectos. Este projeto tem como finalidade o desenvolvimento de um SO semelhante ao UNIX porem, totalmente gratuito, podendo ser utilizado em diversas arquiteturas, contando com softwares de código fonte aberto e podendo ser compartilhado por qualquer usuário.

Criado por Richard M. Stallman sem quaisquer fins lucrativos, com o ideal de utilizar um sistema operacional sem custos, semelhante ao UNIX e disponível para qualquer individuo. Richard também criou a Free Software Foundation, uma organização para promover a liberdade dos usuários de computadores a possuírem softwares livres. Para poder disponibilizar todos os programas livremente, era preciso um documento formal, então Richard, criou a licença GNU GPL, fornecendo quatro liberdades. (SOARES; FERNANDES, 2013).

A licença GPL, tão comentada, mas ao mesmo tempo tão mal-compreendida, pode ser resumida em 4 direitos básicos e uma obrigação. (MORIMOTO, 2009, p. 15).

- a) os softwares podem ser utilizados por qualquer individuo com a intenção e a finalidade que desejar, sem restrição alguma;
 - b) autorização de produzir cópias dos programas e distribuí-las comercialmente;
 - c) direito de fazer alterações e redistribuí-las;
 - d) dever e obrigação de compartilhar as, modificações produzidas, para atrair outras pessoas que, tenham interesse em aprimorar a aplicação;
- (MORIMOTO, 2009).

3.7.2 Linux

O sistema operacional Linux foi desenvolvido por Linus Torvalds, em decorrência de uma decepção com o sistema Minix que era baseado no UNIX, por possuir diversas falhas, Linus decidiu, desenvolver um novo sistema operacional. (OLIVEIRA; CARISSIMI; TOSCANI, 2010).

Quando começou, enviou uma mensagem para reunir pessoas interessadas: "Olá para todos que estão usando o Minix – estou fazendo um sistema (livre) operacional (como um passatempo, pois não será grande e profissional como o GNU) " [...]. (BONAN, 2003, p. 3).

Na época Linus não imaginava que este pequeno projeto ganharia proporção e se tornaria o SO padrão do projeto GNU, por ser disponibilizado pela licença da FSF, ganhou grande popularidade, na medida em que foi crescendo, seu desenvolvimento acatou todo o mundo, em um ciclo interminável, um bom exemplo disto é, uma pessoa situado em uma determinada cidade na Rússia começa a desenvolver uma correção de falhas sobre, uma nova placa incompatível com o sistema, outro indivíduo no Brasil na cidade de Curitiba desenvolve uma aplicação que tenha necessidade. Desta maneira, milhares de pessoas vão contribuindo pouco a pouco constantemente, com auxílio de grandes empresas como NASA, HP, SUN dentre outros. (BONAN, 2003).

Esse sistema operacional é a união de softwares e programas que compõem o projeto GNU, (devido ao projeto são, inteiramente gratuitos) mais o núcleo Linux, o conjunto se denomina distribuição. Todas as distribuições possuem softwares base criados dentro do projeto GNU sobre a licença da FSF, possuindo distinções entre si, como os programas, procedimentos de instalação e manutenção do sistema operacional. Há também distribuições comerciais, destas o valor cobrado referente aos serviços prestados ou aplicações desenvolvidas propriamente da distribuição. (OLIVEIRA; CARISSIMI; TOSCANI, 2010).

3.7.3 Código fonte

O código-fonte é a estrutura de todos os códigos usados por um programador para o software funcionar, existem programas que o código é disponibilizado e outros que são fechados, aonde o usuário não terá acesso a sua codificação. (BONAN, 2003).

A vantagem de se possuir o código fonte aberto é compreender o funcionamento do software, assim realizando alterações, adicionando ou removendo recursos, corrigindo falhas ou adaptando conforme desejado. (ENTENDA..., [2001]).

3.7.4 Linux Mint

De acordo com o Guia Oficial do Usuário Linux Mint Cinamon Edition o Linux Mint é um sistema operacional que começou a ser desenvolvido em 2006, sendo construído sobre a base do kernel do Linux. Seu principal objetivo é distribuir um SO gratuito e livre de custos, podendo ser usado por empresas ou usuários domésticos. A partir da versão 5, o Linux Mint respeita um período de lançamento, a cada seis meses uma nova versão é desenvolvida. Os codinomes utilizados, sempre serão nomes femininos com terminação na letra “a”, seguindo uma ordem alfabética. (GUIA..., [2015]).

A Figura 9 demonstra os codinomes utilizados, com suas respectivas versões:

Figura 9 - Tabela de versões do SO Linux Mint e seus codinomes

Versão	Codinome
1.0	Ada
2.0	Barbara
2.1	Bea
2.2	Bianca
3.0	Cassandra
3.1	Celena
4.0	Daryna
5	Elyssa
6	Felicia
7	Gloria
8	Helena
9	Isadora
10	Julia
11	Katya
12	Lisa
13	Maya
14	Nadia
15	Olivia
16	Petra
17	Qiana
17.1	Rebecca

Fonte: Guia... (c2015, p. 6).

Cada edição, é feita personalizada com a finalidade de suprir certas necessidades, abaixo segue as edições populares:

- MATE Edition (utiliza o ambiente gráfico MATE)
- CinnamonEdition (utiliza o ambiente gráfico Cinnamon)
- KDE Edition (utiliza o ambiente gráfico KDE)
- Xfce Edition (utiliza o ambiente gráfico Xfce). (GUIA..., [2015], p. 7).

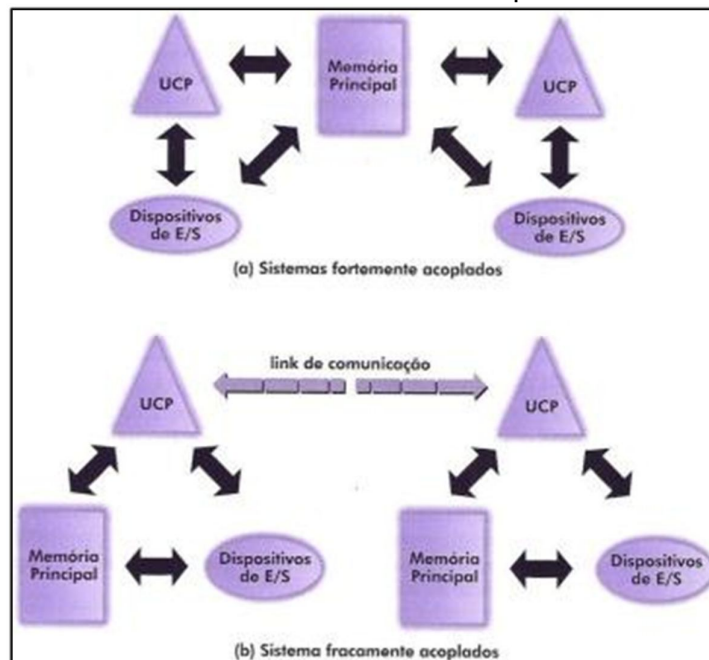
3.8 SISTEMAS FORTEMENTE E FRACAMENTE ACOPLADOS

As arquiteturas responsáveis por realizarem múltiplas sequências de instruções são denominadas MIMD, de acordo com os fatores de compartilhamento da memória principal, distância física entre processadores, tempo de acesso a MP, mecanismo de comunicação, sincronização e velocidade de comunicação entre processadores, é possível determinar o grau de acoplamento de um sistema, classificando em sistemas fortemente acoplados ou fracamente acoplados. Nos sistemas fortemente acoplados, os processadores compartilham uma única memória principal, controlados por um único SO. Sistemas fracamente são hardwares com

seus próprios processadores, MP e sistema operacional agindo independentes, conectados por uma rede e se comunicando. (MACHADO; MAIA, 2010).

A Figura 10 demonstra estas diferenças:

Figura 10 - Esquema de sistemas fortemente acoplado e sistemas fracamente acoplado



Fonte: Machado e Maia (2010, p. 224).

Os sistemas fortemente acoplados dividem a carga de processamento, aonde cada nó consiste em compartilhar a memória e os recursos de entrada e saída, enquanto os sistemas fracamente acoplados são vários computadores completos interligados, dividindo todas as funções. (ENGLANDER, 2011).

A Figura 11 mostra a diferença entre os sistemas:

Figura 11 - Tabela com diferenças entre sistemas fortemente acoplados e sistemas fracamente acoplados

Características	Sistemas fortemente acoplados	Sistemas fracamente acoplados
Espaço de endereçamento	Centralizado	Distribuído
Comunicação entre os processadores	Muito rápida	Lenta
Arquitetura de hardware	Complexa	Mais simples
Sistema operacional	Homogêneo	Geralmente heterogêneo
Cópias do sistema operacional	Existe apenas uma cópia	Existem várias cópias, uma para cada sistema
Programação paralela	Mais fácil	Mais difícil
Número de processadores	Centenas	Não existe limite
Escalabilidade	Baixa	Alta
Disponibilidade	Baixa/Média	Média/Alta
Administração	Simple	Complexa
Custo de software	Menor	Maior
Intercomunicação	Proprietária	Padronizada
Segurança	Centralizada	Distribuída

Fonte: Machado e Maia (2010, p. 225).

3.9 MPI

Os protocolos simples de troca de mensagem possuem apenas as funções de mandar e receber informações, não sendo eficientes para os clusters, primeiro devido a sua limitação de comunicação, segundo porque não podem manipular uma sincronização de informações. A partir deste problema, foi desenvolvido um novo padrão de troca de mensagens chamado Interface de passagem de mensagens Message Passing Interface. A MPI é projetada para aplicações paralelas, fazendo utilização da rede para conectar os computadores, e identificando cada um pela informação do groupID e processID, assim saberá exatamente a fonte da mensagem e seu grupo correspondente, desta forma vários grupos estarão interligados, comunicando-se ao mesmo tempo. (TANENBAUM; STEEN, 2008).

[...] O MPI inclui funções para comunicação ponto-a-ponto (do tipo envio-recebimento), bem como várias formas de comunicação coletiva:

Propagação (Broadcasting): um nó envia uma mensagem para todos os nós

Distribuição (Scattering): um nó envia mensagens distintas para cada nó

Coleta (Gathering): um nó recebe uma mensagem de todos os nós

Troca completa: todos os nós realizam distribuição e coleta [...]. (PARHAMI, 2008, p. 531).

3.10 CLUSTER

Um cluster é conceituado como uma coleção de sistemas independentes conectados entre si para executar um conjunto de aplicações, fornecendo a “ilusão” de ser um único sistema. (OLIVEIRA; CARISSIMI; TOSCANI, 2010, p. 292).

Segundo Silberschatz, Galvin e Gagne (2010), podemos explicar que cada computador acoplado é denominado nó, os clusters são conectados entre si a partir de uma rede local, compartilhando sua memória. Existem vários tipos de cluster, dentre os principais estão:

Cluster de alta disponibilidade: Neste tipo de cluster, todos os nós executam uma mesma aplicação, se um membro do cluster falhar, os demais nós assumem seu local de onde o computador falhou, sempre mantendo disponível o serviço. Cluster de alto desempenho: Responsável por fornecer um poder de processamento maior do que um único processador, sendo capazes de executar determinada função simultaneamente, aumentando o desempenho, desenvolvido especificamente para combinar seu hardware para obter uma solução em menor tempo. (SILBERSCHATZ; GALVIN; GAGNE, 2010).

3.10.1 Cluster Beowulf

Seu nome faz referência a um manuscrito encontrado no século XI, é um poema que descreve um herói que possui uma força descomunal e enfrenta uma perigosa criatura para salvar seu reino, esta história serviu de inspiração aos pesquisadores Thomas Sterling e Donald Becker da agência NASA em 1994 a denominarem este padrão de cluster. (ALECRIM, [c2013]).

Esse Cluster de alto desempenho é construído com computadores disponíveis no comércio como os microcomputadores pessoais, é um hábito criar este cluster com hardware descartado, sendo uma grande estratégia, levando em consideração, a questão do custo comparado ao fator desempenho para, realizar cálculos de alta demanda. São conectados em uma simples rede local, o interessante é não necessitar de programas e componentes específicos, por isso é muito utilizado. Normalmente é composto de softwares com código-fonte aberto, executando o sistema operacional Linux. (SILBERSCHATZ; GALVIN; GAGNE, 2010).

Null e Lobur (2010) afirmam que este aglomerado de beowulf com estas características, pode-se transformar qualquer conjunto de computadores empoeirados a vários anos, em um novo e extremamente poderoso computador, bastando conectá-los via ethernet.

3.12 FERRAMENTAS

A seguir serão apresentados, todas as ferramentas que serão utilizadas na montagem do cluster do tipo beowulf.

3.12.1 Benchmark

Benchmark são programas desenvolvidos para medir o desempenho em tempo real do computador, são divididos em duas classes de programas para computadores desktops. Um analisará o uso intensivo do processador, enquanto o outro verificará a potência da CPU em processamento gráfico. (HENNESSY; PATTERSON, 2008).

Segundo Parhami (2008), são sempre utilizados por todos os fabricantes de computador, para realizar testes, anulando qualquer projeto extremamente falho e aprimorando com ajustes para aumentar seu desempenho ou corrigindo possíveis falhas. Desta maneira antes de lançarem o produto no mercado, disponibilizaram a avaliação feita pelo benchmark e esses resultados, serão um fator de comparação entre diferentes plataformas e classes de computadores.

Aos interessados no hardware, serão de grande auxílio as informações das análises para avaliarem a eficácia do produto, estabelecendo a relação entre custo-desempenho, comparando entre os concorrentes, qual será o melhor para sua determinada utilização. Para analisar o desempenho realizado, será preciso listar todos os testes efetuados, que sejam relevantes, de acordo com a aplicação desejada do computador, comparando o desempenho em meio a gráficos e informações detalhadas. (PARHAMI, 2008).

3.12.2 Rarcrack

O Rarcrack é um programa que encontra senhas em arquivos de formatos 7z, Rar e Zip, utilizando o método de força-bruta, este método consiste em fazer tentativas e erros até encontrar a senha correta. (VIEIRA, [c2012]).

Este software através do método de força bruta faz a combinação de caracteres através do próprio alfabeto: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ, ao chegar na ultima letra do alfabeto, a letra z maiúscula, automaticamente será adicionado outro carácter, adicionando quantos forem necessários até encontrar a senha correta. (KEDVES, [c2007]).

3.12.3 SSH

O nome SSH faz referência a empresa finlandesa que o desenvolveu, é uma ferramenta utilizada para implementação de cliente e servidor, podendo ser administrada remotamente ou seja ter acesso a outro computador sem estar presente. Este software tem como diferencial uma grande quantia de informações disponíveis, perante os outros programas que realizam acesso remoto. (BONAN, 2003).

Existem diversos tipos de SSH, dentre os gratuitos se destaca o OpenSSH, desenvolvido por Tatu Ylonen o pacote deste programa traz tanto as configurações do cliente como do servidor, para conectar os computadores, basta incluir o nome da maquina ou seu endereço de IP, feito isso o SSH tentara procurar os hosts e fará sua comunicação. (CASTELIANO, [2001]).

3.12.4 NFS

O sistema de arquivos de rede abreviado do inglês NFS fornece acesso a arquivos distribuídos pela rede, permitindo que um cliente remoto acesse em meio a rede os arquivos, é o sistema utilizado como padrão para está finalidade em sistemas operacionais UNIXs. O NFS, segue este modelo de cliente/ servidor, aonde o servidor implementa os arquivos e o armazenamento, os clientes se conectaram e acessaram o sistema de arquivos compartilhado, através das informações

compartilhadas ao determinado cliente, poderá ser compartilhado simultaneamente vários sistemas de arquivos. (JONES, [c2010]).

A melhor maneira de compartilhar arquivos com outros computadores que possuam o sistema operacional Linux, é utilizar o NFS, além de ser, fácil trabalhar com este software. Para realizar o compartilhamento dos arquivos, após configurar o servidor com seu respectivo IP e nome de host, será preciso criar os usuários e sua configuração de IP e nome de host igual do servidor, feito isso, basta compartilhar na rede os arquivos desejados no servidor, podendo distribuir vários arquivos, bastando apenas criar os compartilhamentos desejados aos usuários. (MORIMOTO, [c2006]).

3.12.5 Povray

O Povray é um programa que cria imagens em 3D, através de linhas de comandos, podendo manipular: texturas, cores, dimensões, ângulos e demais componentes de uma imagem. (POV-RAY [c2008]).

A sigla Povray origina-se do inglês Persistence of Vision Ray-Tracer , o Ray-Tracer simula na cena a iluminação dos raios de luz, que percorreram os objetos da imagem que será gerada, tornando o mais próximo do real. Este software gera apenas imagens, em formato .pov, fazendo referencia especifica ao programa. (INTRODUCTION... [c2008]).

3.12.6 Compactadores

O software Tar faz o agrupamento de arquivos, unindo vários em um único, no formato .tar, possui dois compactadores / descompactadores que reduzem o tamanho, dos arquivos agrupados, são o gzip no formato .gz e o bzip2 no formato .bz2. (COMPACTANDO... [c2010])

O programa Zip, é um compactador / descompactador de arquivos, sendo um dos mais antigos, a realizar esta tarefa, gera um arquivo no formato .zip. (CISNEIROS, [c2008]).

4 TRABALHOS CORRELATOS

O ambiente de computação paralela para alto desempenho é bem explorada, tanto por grandes empresas como, usuários que desejam obter um desempenho maior em suas atividades, com custos reduzidos, proporcionando um incentivo para realizarem a montagem de um cluster do tipo beowulf. É de fácil acesso encontrar estas informações de como e qual maneira realizar testes e qual a maneira de usar os computadores paralelamente, na tentativa de alcançar uma potência maior, contando com uma boa quantidade de informações e resultados para fazer a montagem do ambiente e executando as ações desejadas.

Dentre os trabalhos pode-se citar:

“Análise de desempenho de um algoritmo paralelo implementado em um cluster beowulf” de autoria de Robson Costa. Neste trabalho o autor monta um cluster do tipo beowulf a partir de 12 computadores iguais: Processador Pentium 3 550 Mhz, 64 Mb Ram e 10 Gb de HD, coordenados por um servidor de mesmas características porém com 384 Mb de Ram, para multiplicar matrizes em tempo reduzido. Segundos seus resultados, oito computadores trabalhando em conjunto multiplicaram uma matriz de ordem 512x512 em 128 segundos, resultando mais demorado que um único computador trabalhando sozinho, que (107 segundos), a partir do décimo computador interligado, ao cluster, o tempo reduziu para 46 segundos. (COSTA, 2002).

“Manual de Montagem de um Cluster Beowulf sob a plataforma GNU/LINUX” de autoria de Danny Augusto Vieira Tonidandel. Foi desenvolvido a união de 3 computadores diferentes, para realizar o calculo de pi, comprovando que houve progresso em calcular os próximos 10^8 números nodais de pi, com um ganho de 1 segundo comparando a um único computador executando o calculo de pi, e os três trabalhando em paralelo. (TONIDANDEL, 2008).

5 METODOLOGIA

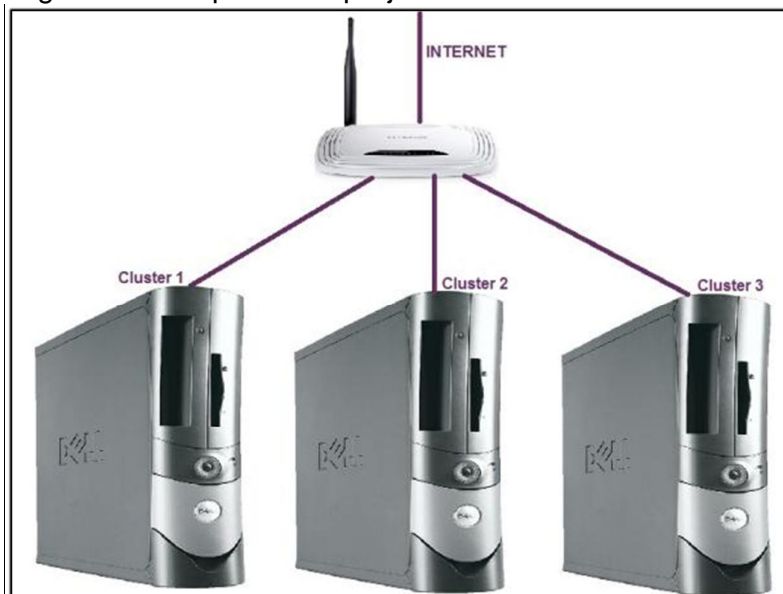
Este trabalho possui caráter experimental como modelo de pesquisa, sendo interdisciplinar focado em ciência da computação, aplicado de forma direta com participação do pesquisador. (PÁDUA, 2007).

Foi realizados neste trabalho, a montagem de um cluster do tipo Beowulf utilizando três computadores da marca DELL, modelo Optiplex GX60, Processador Celeron 2.0 GHz, 128 Kb Cache L2, Memória RAM 512 Mb, HD 80 Gb.

Todos os computadores foram formatados, instalando o sistema operacional Linux Mint 17.1 XFCE Edition 32 bits, disponibilizado gratuitamente no site¹.O particionamento do disco rígido foi 1 Gb reservado para swap e o restante 79 Gb para o Sistema Operacional, no formato ext4.

Atribuindo-se um nome para computador e um endereço de IP falso (IP de Sub-rede), com classificação Classe C, com IP de uma rede 192.168.0.0, e para cada computador, ficando respectivamente, seu nome e IP:cluster1: 192.168.0.11; cluster2: 192.168.0.12; cluster3: 192.168.0.13, todos foram configurados na máscara de rede: 255.255.255.0 . Estes foram conectados por meio de um cabo de rede de par trançado categoria 5, ao roteador da marca tp-link modelo tl-wan470n, possuindo 4 portas, trabalhando na velocidade de 150 Mbps, conforme na Figura 12.

Figura 12 - Esquema do projeto



Fonte: Elaborada pelo Autor.

¹ torrents.linuxmint.com/torrents/linuxmint-17.1-xfce-32bit.iso.torrent

O computador de nome cluster 1 (servidor) se encarregou de administrar todo o cluster, a partir de suas ações, gerenciou todos os recursos disponíveis de hardware. Enquanto o servidor comanda as ações, os demais computadores aguardaram receber instruções para executarem.

Foram instalados em todos os computadores, os seguintes programas:

- a) OpenSSH para ter acesso a todos os nós ligados;
- b) MPICH2, que será o responsável por sincronizar todas as máquinas, a realizarem o processamento paralelo, utilizando a troca de mensagens MPI, em meio a suas ações;
- c) NFS que compartilhará todos os arquivos na LAN, para todo o cluster, necessário para o funcionamento;
- d) Pov-ray;
- e) Rarcrack;

Todos os programas são distribuídos gratuitamente e atualizados pelo repositório de softwares desta distribuição.

Para medir o desempenho dos computadores foram utilizados softwares com propósitos de benchmark. Sendo realizado o download destes softwares gratuitamente da internet, instalando nos computadores, para analisar o comportamento individual e paralelo. Estes testes incluem:

- a) compactar arquivos ISO de tamanhos 250 Mb, 500 Mb, 1 Gb, 2 Gb e 4Gb, usando o programa Bzip2;
- b) compactar arquivos ISO de tamanhos 250 Mb, 500 Mb, 1 Gb, 2 Gb e 4Gb, usando o programa Zip;
- c) descompactar arquivos ISO de tamanhos 250 Mb, 500 Mb, 1 Gb, 2 Gb e 4Gb, usando o programa Bzip2;
- d) compactar arquivos ISO de tamanhos 250 Mb, 500 Mb, 1 Gb, 2 Gb e 4Gb, usando o programa Zip;
- e) fazer a cópia de arquivos ISO de tamanhos 250 Mb, 500 Mb, 1 Gb, 2 Gb e 4Gb.
- f) utilizar o programa Rarcrack para encontrar senhas de arquivos compactados com a seguinte codificação: jjj, tt, DDD, NNN e WWW no formato Zip;

- g) renderizar uma imagem com o programa Povray, nas dimensões: 180x120, 320x240, 640x480, 800x600 e 1024x768;

Os resultados destas análises foram mostrados em gráficos e tabelas que apresentaram o desempenho alcançado entre o cluster por completo, com dois nós em funcionamento e um único nó trabalhando sozinho. O detalhamento destas ocorrências indicam se houve um aumento significativo, no desempenho.

6 ANÁLISE BENCHMARKS

Após realizar a instalação do sistema operacional (Apêndice A), configurar o nó servidor (Apêndice B) e os nós escravos (Apêndice C), será descrito, como foram feitos todos os testes e seus respectivos resultados, apresentando todos os testes realizados em tabelas e os gráficos demonstraram a média com um computador representado pela cor azul, dois computadores representados pela cor vermelha e três computadores trabalhando em conjunto representado pela cor verde.

Para fazer os testes, foi preciso estar com o terminal aberto no cluster1, logado como administrador, estando localizado na pasta testes. Todos os testes precisaram ser comandados, utilizando o programa MPI, seguido do comando de quais computadores realizaram a tarefa, usando o programa time, da própria distribuição que retornará o tempo de execução e o tipo de análise a ser feita.

6.1 COMPACTANDO COM BZIP2

Para realizar os testes de compactação com o programa Bzip2, foram dados os comandos individualmente, até terminar o procedimento que retornou o tempo decorrido do procedimento. Os testes são respectivamente a compactação dos arquivos de 250 Mb (a.iso), 500 Mb (b.iso), 1 Gb (c.iso), 2 Gb (d.iso) e 4 Gb (e.iso), que irão gerar o arquivo compactado no format Bzip2, que será salvo na pasta bzip2 respectivo ao seu nome sendo: a.bz2, b.bz2, c.bz2, d.bz2 e e.bz2. Foram processados, estes comandos com um nó, dois nós e três nós trabalhando em conjunto. Foram utilizados os comandos listados a seguir em negrito:

```
mpirun -n 1 -host=cluster1 time tar -cjf bzip2/a.bz2 a.iso
```

```
mpirun -n 2 -host=cluster1,cluster2 time tar -cjf bzip2/a.bz2 a.iso
```

```
mpirun -n 3 -host=cluster1,cluster2,cluster3 time tar -cjf bzip2/a.bz2 a.iso
```

```
mpirun -n 1 -host=cluster1 time tar -cjf bzip2/b.bz2 b.iso
```

```
mpirun -n 2 -host=cluster1,cluster2 time tar -cjf bzip2/b.bz2 b.iso
```

```
mpirun -n 3 -host=cluster1,cluster2,cluster3 time tar -cjf bzip2/b.bz2 b.iso
```

```

mpirun -n 1 -host=cluster1 time tar -cjf bzip2/c.bz2 c.iso
mpirun -n 2 -host=cluster1,cluster2 time tar -cjf bzip2/c.bz2 c.iso
mpirun -n 3 -host=cluster1,cluster2,cluster3 time tar -cjf bzip2/c.bz2 c.iso

```

```

mpirun -n 1 -host=cluster1 time tar -cjf bzip2/d.bz2 d.iso
mpirun -n 2 -host=cluster1,cluster2 time tar -cjf bzip2/d.bz2 d.iso
mpirun -n 3 -host=cluster1,cluster2,cluster3 time tar -cjf bzip2/d.bz2 d.iso

```

```

mpirun -n 1 -host=cluster1 time tar -cjf bzip2/e.bz2 e.iso
mpirun -n 2 -host=cluster1,cluster2 time tar -cjf bzip2/e.bz2 e.iso
mpirun -n 3 -host=cluster1,cluster2,cluster3 time tar -cjf bzip2/e.bz2 e.iso

```

A Figura 13 é uma tabela que demonstra todos os resultados obtidos, a partir dos comandos realizados anteriormente.

A primeira linha mostra o tamanho dos arquivos compactados.

A linha Cluster1 mostra os resultados de um único computador realizando estes testes.

As linhas Cluster1, Cluster2 e Média mostram os resultados de dois computadores trabalhando em conjunto e sua respectiva média.

As linhas Cluster1, Cluster2, Cluster3 e Média mostram os resultados dos três computadores trabalhando em conjunto e sua respectiva média.

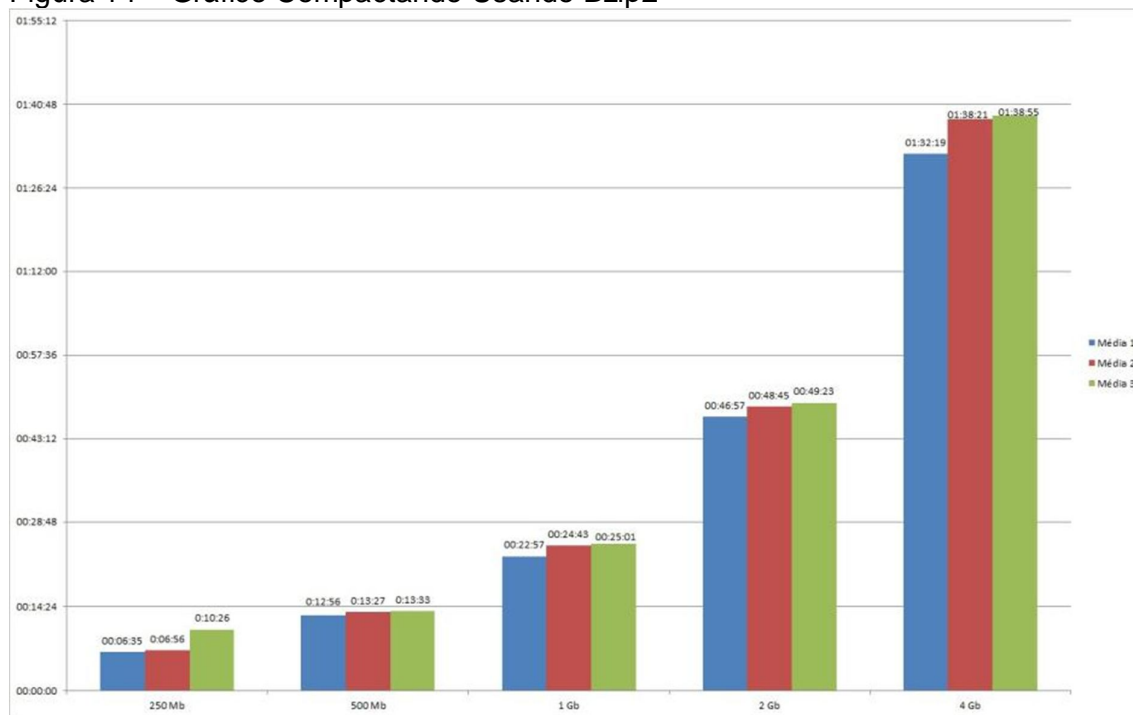
Figura 13 – Tabela Compactando Usando Bzip2

Tamanho	250 Mb	500 Mb	1 Gb	2 Gb	4 Gb
Cluster 1	00:06:35	0:12:56	00:22:57	00:46:57	01:32:19
Cluster 1	0:06:43	0:13:14	00:24:17	00:48:33	01:36:49
Cluster 2	0:07:08	0:13:41	00:25:09	00:48:56	01:39:53
Média	0:06:56	0:13:27	00:24:43	00:48:45	01:38:21
Cluster 1	0:06:49	0:13:27	00:24:42	00:48:30	01:37:32
Cluster 2	0:12:02	0:13:32	00:25:05	00:49:10	01:38:00
Cluster 3	0:12:28	0:13:40	00:25:16	00:50:30	01:41:12
Média	0:10:26	0:13:33	00:25:01	00:49:23	01:38:55

Fonte: Elaborada pelo autor.

Na Figura 14, observa-se que em todos os testes ao trabalhar em conjunto o desempenho piora um pouco, com exceção do primeiro teste que, tem uma grande diferença dos três computadores trabalhando juntos.

Figura 14 – Gráfico Compactando Usando Bzip2



Fonte: Elaborada pelo Autor.

6.2 COMPACTANDO COM ZIP

Para realizar os testes de compactação com o programa Bzip2, foram utilizados os comandos individualmente, até terminar o procedimento que retornou o tempo decorrido do procedimento. Os testes são respectivamente a compactação dos arquivos de 250 Mb (a.iso), 500 Mb (b.iso), 1 Gb (c.iso), 2 Gb (d.iso) e 4 Gb (e.iso), que irão gerar o arquivo compactado no format zip, que será salvo na pasta zip, respectivo ao seu nome sendo: a.zip, b.zip, c.zip, d.zip e e.zip. Foram processados, estes comandos com um nó, dois nós e três nós trabalhando em conjunto. Foram utilizados os comandos listados abaixo em negrito:

mpirun -n 1 -host=cluster1 time zip -r zip/a.zip a.iso

mpirun -n 2 -host=cluster1,cluster2 time zip -r zip/a.zip a.iso

mpirun -n 3 -host=cluster1,cluster2,cluster3 time zip -r zip/a.zip a.iso

```
mpirun -n 1 -host=cluster1 time zip -r zip/b.zip b.iso
mpirun -n 2 -host=cluster1,cluster2 time zip -r zip/b.zip b.iso
mpirun -n 3 -host=cluster1,cluster2,cluster3 time zip -r zip/b.zip b.iso

mpirun -n 1 -host=cluster1 time zip -r zip/c.zip c.iso
mpirun -n 2 -host=cluster1,cluster2 time zip -r zip/c.zip c.iso
mpirun -n 3 -host=cluster1,cluster2,cluster3 time zip -r zip/c.zip c.iso

mpirun -n 1 -host=cluster1 time zip -r zip/d.zip d.iso
mpirun -n 2 -host=cluster1,cluster2 time zip -r zip/d.zip d.iso
mpirun -n 3 -host=cluster1,cluster2,cluster3 time zip -r zip/d.zip d.iso

mpirun -n 1 -host=cluster1 time zip -r zip/e.zip e.iso
mpirun -n 2 -host=cluster1,cluster2 time zip -r zip/e.zip e.iso
mpirun -n 3 -host=cluster1,cluster2,cluster3 time zip -r zip/e.zip e.iso
```

A Figura 15 é uma tabela que demonstra todos os resultados obtidos, a partir dos comandos realizados anteriormente.

A primeira linha mostra o tamanho dos arquivos compactados.

A linha Cluster1 mostra os resultados de um único computador realizando estes testes.

As linhas Cluster1, Cluster2 e Média mostram os resultados de dois computadores trabalhando em conjunto e sua respectiva média.

As linhas Cluster1, Cluster2, Cluster3 e Média mostram os resultados dos três computadores trabalhando em conjunto e sua respectiva média.

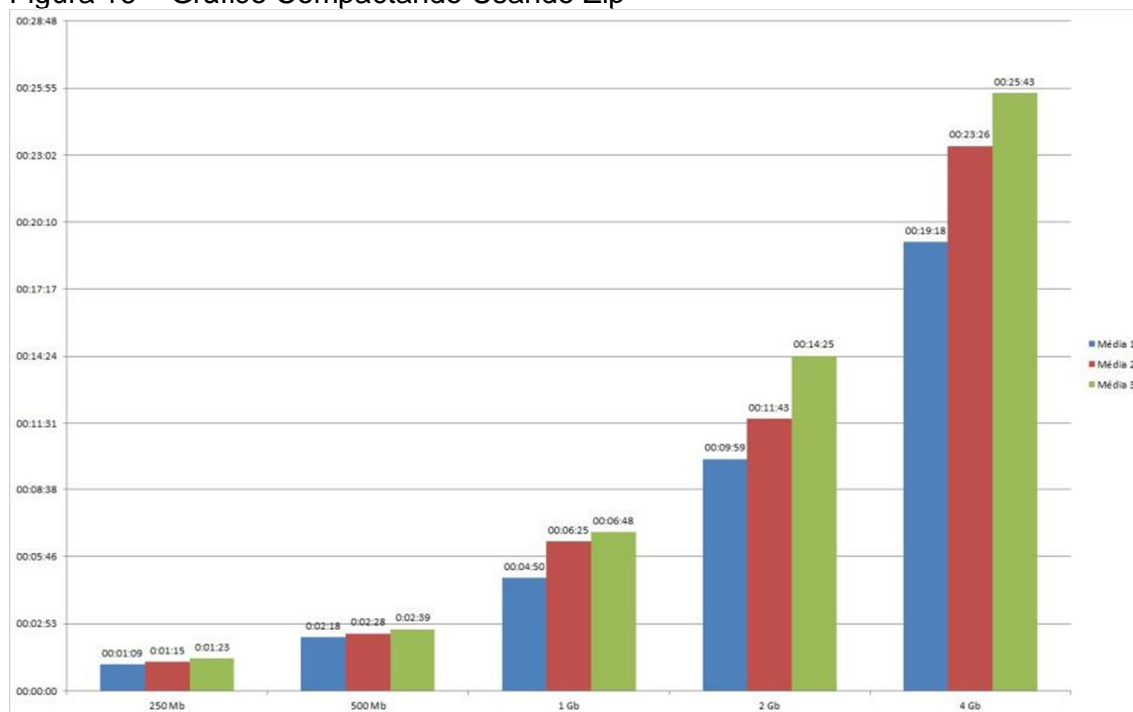
Figura 15 – Tabela Compactando Usando Zip

Tamanho	250 Mb	500 Mb	1 Gb	2 Gb	4 Gb
Cluster 1	00:01:09	0:02:18	00:04:50	00:09:59	00:19:18
Cluster 1	0:01:14	0:02:26	00:06:17	00:11:20	00:22:47
Cluster 2	0:01:16	0:02:30	00:06:33	00:12:05	00:24:05
Média	0:01:15	0:02:28	00:06:25	00:11:43	00:23:26
Cluster 1	0:01:20	0:02:28	00:06:00	00:11:59	00:21:58
Cluster 2	0:01:22	0:02:40	00:06:20	00:13:37	00:22:46
Cluster 3	0:01:28	0:02:48	00:08:05	00:17:39	00:32:24
Média	0:01:23	0:02:39	00:06:48	00:14:25	00:25:43

Fonte: Elaborada pelo Autor.

Na Figura 16 observa-se que, em todos os testes, ao trabalhar em conjunto o desempenho piora gradativamente, sem qualquer progresso satisfatório.

Figura 16 – Gráfico Compactando Usando Zip



Fonte: Elaborada pelo Autor.

6.3 DESCOMPACTANDO COM BZIP2

Para realizar os testes de descompactação com o programa Bzip2, é necessário estar na pasta bzip2, com o comando **cd bzip2**, para entrar na pasta. Foram utilizados os comandos individualmente, até terminar o procedimento que retornou o tempo decorrido do procedimento. Os testes são respectivamente a descompactação dos arquivos a.bz2, b.bz2, c.bz2, d.bz2 e e.bz2, que irão gerar o arquivo descompactado no format iso, respectivo ao seu nome sendo: a.iso, b.iso, c.iso, d.iso e e.iso. Foram processados, estes comandos com um nó, dois nós e três nós trabalhando em conjunto. Foram utilizados os comandos listados abaixo em negrito:

```
mpirun -n 1 -host=cluster1 time tar -xjf a.bz2
```

```
mpirun -n 2 -host=cluster1,cluster2 time tar -xjf a.bz2
```

```
mpirun -n 3 -host=cluster1,cluster2,cluster3 time tar -xjf a.bz2
```

```
mpirun -n 1 -host=cluster1 time tar -xjf b.bz2
```

```
mpirun -n 2 -host=cluster1,cluster2 time tar -xjf b.bz2
```

```
mpirun -n 3 -host=cluster1,cluster2,cluster3 time tar -xjf b.bz2
```

```
mpirun -n 1 -host=cluster1 time tar -xjf c.bz2
```

```
mpirun -n 2 -host=cluster1,cluster2 time tar -xjf c.bz2
```

```
mpirun -n 3 -host=cluster1,cluster2,cluster3 time tar -xjf c.bz2
```

```
mpirun -n 1 -host=cluster1 time tar -xjf d.bz2
```

```
mpirun -n 2 -host=cluster1,cluster2 time tar -xjf d.bz2
```

```
mpirun -n 3 -host=cluster1,cluster2,cluster3 time tar -xjf d.bz2
```

```
mpirun -n 1 -host=cluster1 time tar -xjf e.bz2
```

```
mpirun -n 2 -host=cluster1,cluster2 time tar -xjf e.bz2
```

```
mpirun -n 3 -host=cluster1,cluster2,cluster3 time tar -xjf e.bz2
```

A Figura 17 é uma tabela que demonstra todos os resultados obtidos, a partir dos comandos realizados anteriormente.

A primeira linha mostra o tamanho dos arquivos descompactados.

A linha Cluster1 mostra os resultados de um único computador realizando estes testes.

As linhas Cluster1, Cluster2 e Média mostram os resultados de dois computadores trabalhando em conjunto e sua respectiva média.

As linhas Cluster1, Cluster2, Cluster3 e Média mostram os resultados dos três computadores trabalhando em conjunto e sua respectiva média.

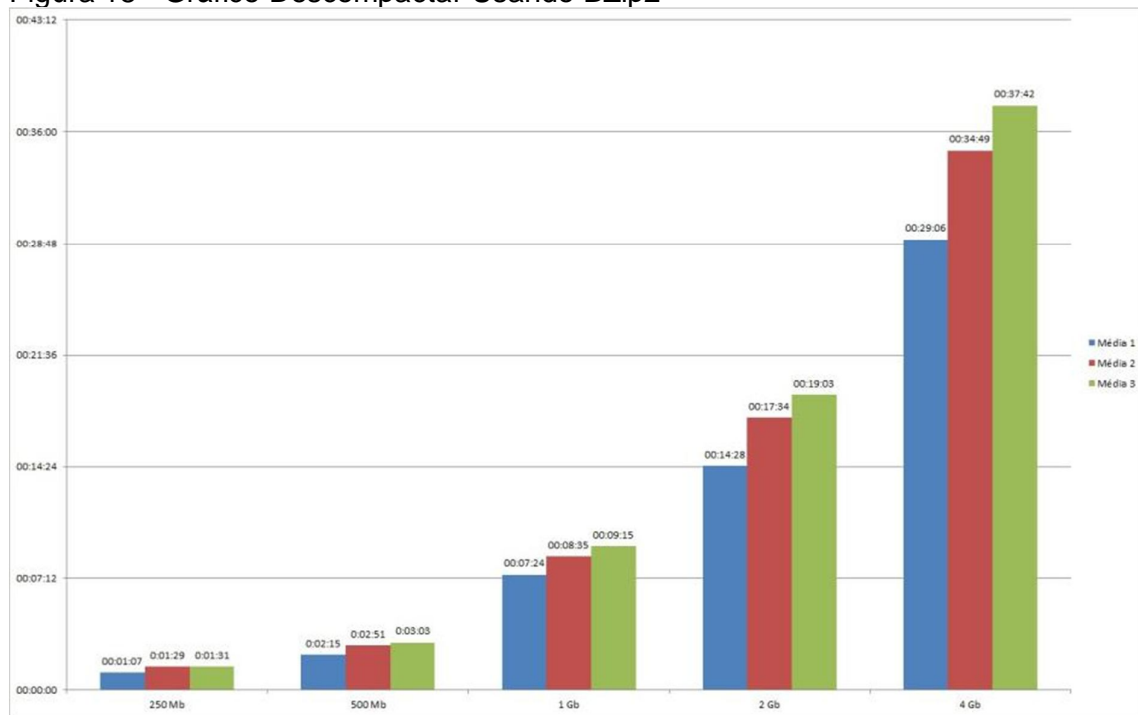
Figura 17 – Tabela Descompactar Usando BZip2

Tamanho	250 Mb	500 Mb	1 Gb	2 Gb	4 Gb
Cluster 1	00:01:07	0:02:15	00:07:24	00:14:28	00:29:06
Cluster 1	0:01:22	0:02:48	00:08:34	00:17:33	00:34:48
Cluster 2	0:01:35	0:02:53	00:08:37	00:17:35	00:34:50
Média	0:01:29	0:02:51	00:08:35	00:17:34	00:34:49
Cluster 1	0:01:25	0:02:51	00:08:51	00:17:43	00:35:18
Cluster 2	0:01:27	0:02:55	00:08:53	00:17:56	00:36:14
Cluster 3	0:01:40	0:03:23	00:10:01	00:21:30	00:41:34
Média	0:01:31	0:03:03	00:09:15	00:19:03	00:37:42

Fonte: Elaborada pelo Autor.

Na Figura 18 observa-se que em todos os testes ao trabalhar em conjunto o desempenho piora gradativamente, sem qualquer progresso satisfatório.

Figura 18 - Gráfico Descompactar Usando BZip2



Fonte: Elaborada pelo Autor.

6.4 DESCOMPACTANDO COM ZIP

Para realizar os testes de descompactação com o programa zip, é necessário estar na pasta zip, com o comando **cd zip**, para entrar na pasta. Foram utilizados os comandos individualmente, até terminar o procedimento que retornou o tempo decorrido do procedimento. Os testes são respectivamente a descompactação dos arquivos a.zip, b.zip, c.zip, d.zip e e.zip, que irão gerar o arquivo descompactado no format iso, respectivo ao seu nome sendo: a.iso, b.iso, c.iso, d.iso e e.iso. Foram processados, estes comandos com um nó, dois nós e três nós trabalhando em conjunto. Foram utilizados os comandos listados abaixo em negrito:

```
mpirun -n 1 -host=cluster1 time unzip -o a.zip
```

```
mpirun -n 2 -host=cluster1,cluster2 time unzip -o a.zip
```

```
mpirun -n 3 -host=cluster1,cluster2,cluster3 time unzip -o a.zip
```

```
mpirun -n 1 -host=cluster1 time unzip -o b.zip
```

```
mpirun -n 2 -host=cluster1,cluster2 time unzip -o b.zip
```

```
mpirun -n 3 -host=cluster1,cluster2,cluster3 time unzip -o b.zip
```



```

mpirun -n 1 -host=cluster1 time unzip -o c.zip
mpirun -n 2 -host=cluster1,cluster2 time unzip -o c.zip
mpirun -n 3 -host=cluster1,cluster2,cluster3 time unzip -o c.zip

```

```

mpirun -n 1 -host=cluster1 time unzip -o d.zip
mpirun -n 2 -host=cluster1,cluster2 time unzip -o d.zip
mpirun -n 3 -host=cluster1,cluster2,cluster3 time unzip -o d.zip

```

```

mpirun -n 1 -host=cluster1 time unzip -o e.zip
mpirun -n 2 -host=cluster1,cluster2 time unzip -o e.zip
mpirun -n 3 -host=cluster1,cluster2,cluster3 time unzip -o e.zip

```

A Figura 19 é uma tabela que demonstra todos os resultados obtidos, a partir dos comandos realizados anteriormente.

A primeira linha mostra o tamanho dos arquivos descompactados.

A linha Cluster1 mostra os resultados de um único computador realizando estes testes.

As linhas Cluster1, Cluster2 e Média mostram os resultados de dois computadores trabalhando em conjunto e sua respectiva média.

As linhas Cluster1, Cluster2, Cluster3 e Média mostram os resultados dos três computadores trabalhando em conjunto e sua respectiva média.

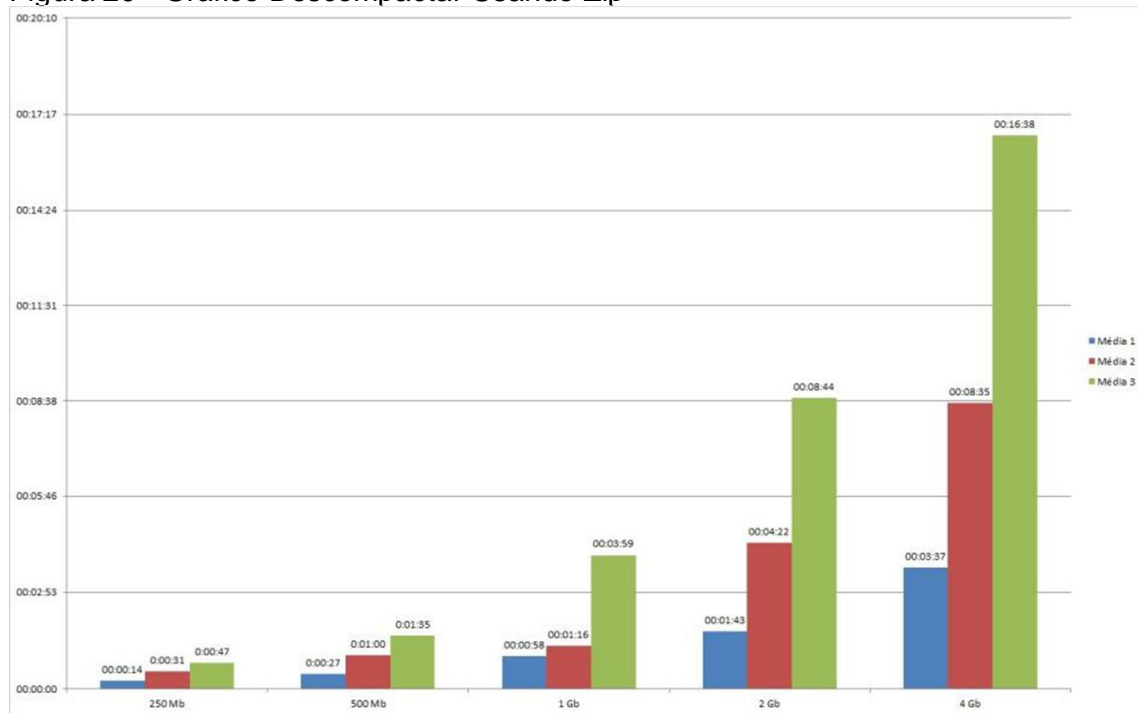
Figura 19 - Tabela Descompactar Zip

Tamanho	250 Mb	500 Mb	1 Gb	2 Gb	4 Gb
Cluster 1	00:00:14	0:00:27	00:00:58	00:01:43	00:03:37
Cluster 1	0:00:29	0:00:59	00:01:18	00:04:20	00:08:33
Cluster 2	0:00:33	0:01:01	00:01:15	00:04:23	00:08:36
Média	0:00:31	0:01:00	00:01:16	00:04:22	00:08:35
Cluster 1	0:00:25	0:00:55	00:02:17	00:05:58	00:10:05
Cluster 2	0:00:58	0:01:52	00:04:49	00:09:49	00:19:54
Cluster 3	0:00:57	0:01:58	00:04:52	00:10:25	00:19:56
Média	0:00:47	0:01:35	00:03:59	00:08:44	00:16:38

Fonte: Elaborada pelo Autor.

Na Figura 20, observa-se que em todos os testes ao trabalhar em conjunto o desempenho piora bruscamente, sem qualquer progresso satisfatório ao trabalhar com mais de um computador.

Figura 20 - Gráfico Descompactar Usando Zip



Fonte: Elaborada pelo Autor.

6.5 COPIAR ARQUIVOS

Para realizar os testes de copia de arquivo, foram utilizados os comandos individualmente, até terminar o procedimento que retornou o tempo decorrido do procedimento. Os testes são respectivamente a copia dos arquivos a.iso, b.iso, c.iso, d.iso e e.iso, que irão gerar o arquivo copiado, que será salvo na pasta copia, com um caracter a mais para identificar a copia, respectivamente: aa.iso, bb.iso, cc.iso, dd.iso e ee.iso. Foram processados, estes comandos com um nó, dois nós e três nós trabalhando em conjunto. Foram utilizados os comandos listados abaixo em negrito:

mpirun -n 1 -host=cluster1 time cp -p a.iso copia/aa.iso

mpirun -n 2 -host=cluster1,cluster2 time cp -p a.iso copia/aa.iso

mpirun -n 3 -host=cluster1,cluster2,cluster3 time cp -p a.iso copia/aa.iso

```
mpirun -n 1 -host=cluster1 time cp -p b.iso copia/bb.iso  
mpirun -n 2 -host=cluster1,cluster2 time cp -p b.iso copia/bb.iso  
mpirun -n 3 -host=cluster1,cluster2,cluster3 time cp -p b.iso copia/bb.iso
```

```
mpirun -n 1 -host=cluster1 time cp -p c.iso copia/cc.iso  
mpirun -n 2 -host=cluster1,cluster2 time cp -p c.iso copia/cc.iso  
mpirun -n 3 -host=cluster1,cluster2,cluster3 time cp -p c.iso copia/cc.iso
```

```
mpirun -n 1 -host=cluster1 time cp -p d.iso copia/dd.iso  
mpirun -n 2 -host=cluster1,cluster2 time cp -p d.iso copia/dd.iso  
mpirun -n 3 -host=cluster1,cluster2,cluster3 time cp -p d.iso copia/dd.iso
```

```
mpirun -n 1 -host=cluster1 time cp -p e.iso copia/ee.iso  
mpirun -n 2 -host=cluster1,cluster2 time cp -p e.iso copia/ee.iso  
mpirun -n 3 -host=cluster1,cluster2,cluster3 time cp -p e.iso copia/ee.iso
```

A Figura 21 é uma tabela que demonstra todos os resultados obtidos, a partir dos comandos realizados anteriormente.

A primeira linha mostra o tamanho dos arquivos copiados.

A linha Cluster1 mostra os resultados de um único computador realizando estes testes.

As linhas Cluster1, Cluster2 e Média mostram os resultados de dois computadores trabalhando em conjunto e sua respectiva média.

As linhas Cluster1, Cluster2, Cluster3 e Média mostram os resultados dos três computadores trabalhando em conjunto e sua respectiva média.

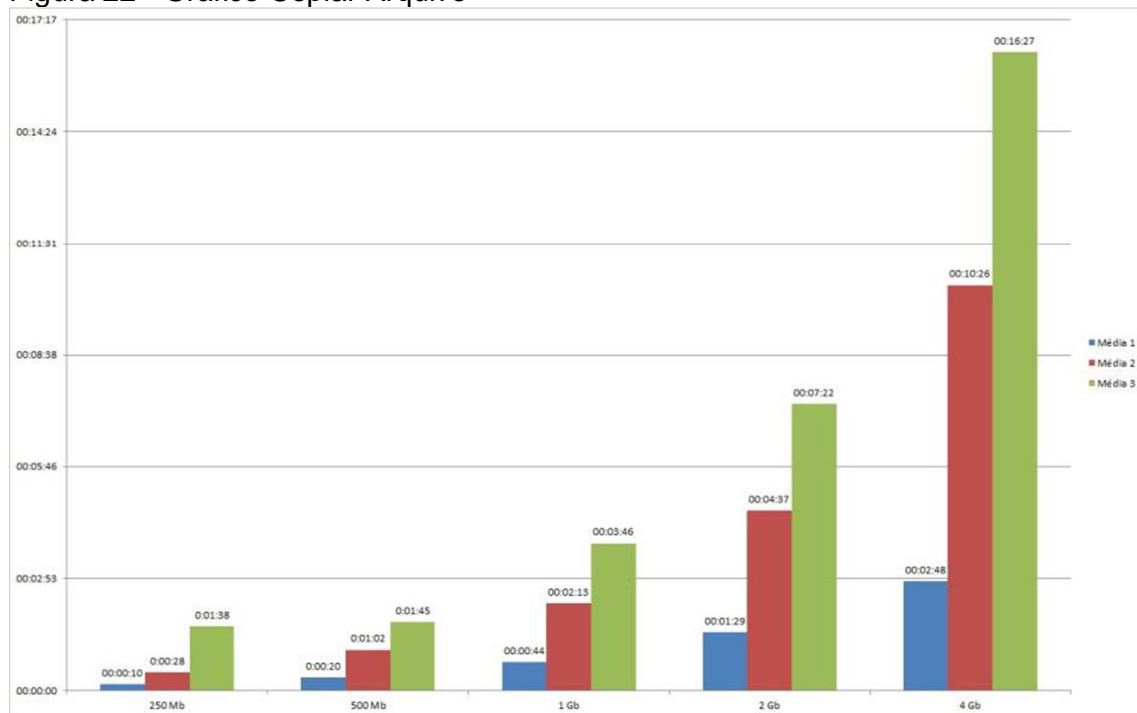
Figura 21 - Tabela Copiar Arquivo

Tamanho	250 Mb	500 Mb	1 Gb	2 Gb	4 Gb
Cluster 1	00:00:10	0:00:20	00:00:44	00:01:29	00:02:48
Cluster 1	0:00:24	0:00:57	00:02:06	00:04:27	00:10:27
Cluster 2	0:00:32	0:01:08	00:02:21	00:04:46	00:10:26
Média	0:00:28	0:01:02	00:02:13	00:04:37	00:10:26
Cluster 1	0:00:33	0:00:38	00:01:30	00:03:01	00:08:49
Cluster 2	0:02:10	0:02:18	00:04:52	00:09:31	00:19:41
Cluster 3	0:02:12	0:02:19	00:04:56	00:09:35	00:20:50
Média	0:01:38	0:01:45	00:03:46	00:07:22	00:16:27

Fonte: Elaborada pelo Autor.

Na Figura 22, observa-se que em todos os testes ao trabalhar em conjunto o desempenho piora bruscamente, sem qualquer progresso satisfatório ao trabalhar com mais de um computador.

Figura 22 - Gráfico Copiar Arquivo



Fonte: Elaborada pelo Autor.

6.6 QUEBRA DE SENHAS USANDO RARCRACK

Para realizar os testes para encontrar senhas, utilizando o programa Rarcrack, é preciso entrar na pasta senha, com o commando **cd senha**. Foram utilizados os comandos individualmente, até terminar o procedimento que retornou o tempo decorrido do procedimento. Os testes são respectivamente a quebra de senha dos arquivos **jjj.zip**, **ttt.zip**, **DDD.zip**, **NNN.zip** e **WWW.zip**, que irão mostrar a senha, correta quando terminar o procedimento. Foram processados, estes comandos com um nó, dois nós e três nós trabalhando em conjunto. Foram utilizados os comandos listados abaixo em negrito:

```
mpirun -n1 -host=cluster1 time rarcrack --threads 1 --type zip jjj.zip  
mpirun -n2 -host=cluster1,cluster2 time rarcrack --threads 2 --type zip jjj.zip  
mpirun -n3 -host=cluster1,cluster2,cluster3 time rarcrack --threads 3 --type zip  
jjj.zip
```

```
mpirun -n1 -host=cluster1 time rarcrack --threads 1 --type zip ttt.zip  
mpirun -n2 -host=cluster1,cluster2 time rarcrack --threads 2 --type zip ttt.zip  
mpirun -n3 -host=cluster1,cluster2,cluster3 time rarcrack --threads 3 --type zip  
ttt.zip
```

```
mpirun -n1 -host=cluster1 time rarcrack --threads 1 --type zip DDD.zip  
mpirun -n2 -host=cluster1,cluster2 time rarcrack --threads 2 --type zip DDD.zip  
mpirun -n3 -host=cluster1,cluster2,cluster3 time rarcrack --threads 3 --type zip  
DDD.zip
```

```
mpirun -n1 -host=cluster1 time rarcrack --threads 1 --type zip NNN.zip  
mpirun -n2 -host=cluster1,cluster2 time rarcrack --threads 2 --type zip NNN.zip  
mpirun -n3 -host=cluster1,cluster2,cluster3 time rarcrack --threads 3 --type zip  
NNN.zip
```

```
mpirun -n1 -host=cluster1 time rarcrack --threads 1 --type zip WWW.zip
```

```
mpirun -n2 -host=cluster1,cluster2 time rarcrack --threads 2 --type zip WWW.zip
```

```
mpirun -n3 -host=cluster1,cluster2,cluster3 time rarcrack --threads 3 --type zip WWW.zip
```

A Figura 23 é uma tabela que demonstra todos os resultados obtidos, a partir dos comandos realizados anteriormente.

A primeira linha mostra a senha dos arquivos codificados.

A linha Cluster1 mostra os resultados de um único computador realizando estes testes.

As linhas Cluster1, Cluster2 e Média mostram os resultados de dois computadores trabalhando em conjunto e sua respectiva média.

As linhas Cluster1, Cluster2, Cluster3 e Média mostram os resultados dos três computadores trabalhando em conjunto e sua respectiva média.

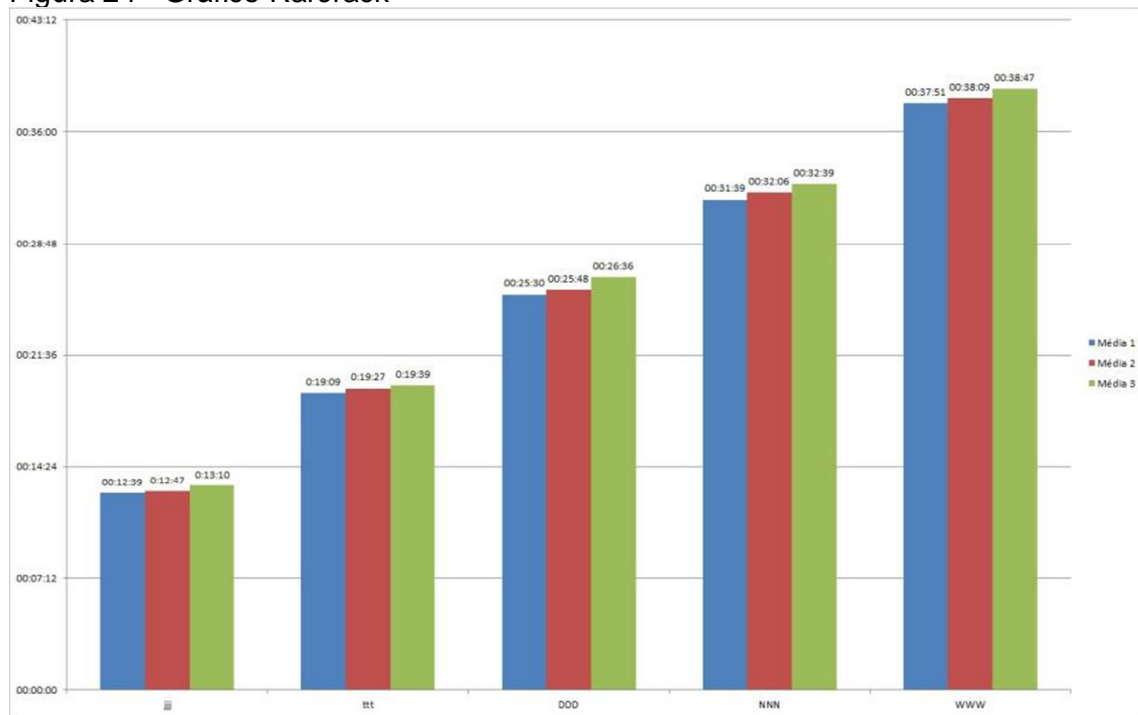
Figura 23 - Tabela Rarcrack

Senha	jjj	ttt	DDD	NNN	WWW
Cluster 1	00:12:39	0:19:09	00:25:30	00:31:39	00:37:51
Cluster 1	0:12:47	0:19:27	00:25:48	00:32:06	00:38:09
Cluster 2	0:12:47	0:19:27	00:25:48	00:32:06	00:38:09
Média	0:12:47	0:19:27	00:25:48	00:32:06	00:38:09
Cluster 1	0:13:10	0:19:39	00:26:36	00:32:39	00:38:47
Cluster 2	0:13:10	0:19:39	00:26:36	00:32:39	00:38:47
Cluster 3	0:13:10	0:19:39	00:26:36	00:32:39	00:38:47
Média	0:13:10	0:19:39	00:26:36	00:32:39	00:38:47

Fonte: Elaborada pelo Autor.

Na Figura 24 observa-se que em todos os testes ao trabalhar em conjunto o desempenho piora bem pouco, sem qualquer progresso satisfatório ao trabalhar com mais de um computador.

Figura 24 - Gráfico Rarcrack



Fonte: Elaborada pelo Autor.

6.7 RENDERIZAR IMAGEM USANDO POVRAY

Para realizar os testes de renderização de imagem, foi preciso entrar na pasta da imagem que sera renderizada, **cd plants_demo**. Foram utilizados os comandos individualmente, até terminar o procedimento que, retornou a imagem renderizada como mostra a Figura 25. Os testes são respectivamente renderizar a imagem nos tamanhos: 180x120, 320x240, 640x480, 800x600 e 1024x768. Foram processados, estes comandos com um nó, dois nós e três nós trabalhando em conjunto. Foram utilizados os comandos listados abaixo em negrito:

```
mpirun -n 1 -host=cluster1 time povray plants_demo.pov -w180 -h120
mpirun -n 2 -host=cluster1,cluster2 time povray plants_demo.pov -w180 -
h120
mpirun -n 3 -host=cluster1,cluster2,cluster3 time povray plants_demo.pov -
w180 -h120
```

```
mpirun -n 1 -host=cluster1 time povray plants_demo.pov -w320 -h240
mpirun -n 2 -host=cluster1,cluster2 time povray plants_demo.pov -w320 -
h240
mpirun -n 3 -host=cluster1,cluster2,cluster3 time povray plants_demo.pov -
w320 -h240
```

```
mpirun -n 1 -host=cluster1 time povray plants_demo.pov -w640 -h480
mpirun -n 2 -host=cluster1,cluster2 time povray plants_demo.pov -w640 -
h480
mpirun -n 3 -host=cluster1,cluster2,cluster3 time povray plants_demo.pov -
w640 -h480
```

```
mpirun -n 1 -host=cluster1 time povray plants_demo.pov -w800 -h600
mpirun -n 2 -host=cluster1,cluster2 time povray plants_demo.pov -w800 -
h600
mpirun -n 3 -host=cluster1,cluster2,cluster3 time povray plants_demo.pov -
w800 -h600
```

```
mpirun -n 1 -host=cluster1 time povray plants_demo.pov -w1024 -h768
mpirun -n 2 -host=cluster1,cluster2 time povray plants_demo.pov -w1024 -
h768
mpirun -n 3 -host=cluster1,cluster2,cluster3 time povray plants_demo.pov -
w1024 -h768
```


Figura 25 - Imagem Renderizada



Fonte: Tran (c2002).

A Figura 26 é uma tabela que demonstra todos os resultados obtidos, a partir dos comandos realizados anteriormente.

A primeira linha mostra a dimensão das imagens renderizadas.

A linha Cluster1 mostra os resultados de um único computador realizando estes testes.

As linhas Cluster1, Cluster2 e Média mostram os resultados de dois computadores trabalhando em conjunto e sua respectiva média.

As linhas Cluster1, Cluster2, Cluster3 e Média mostram os resultados dos três computadores trabalhando em conjunto e sua respectiva média.

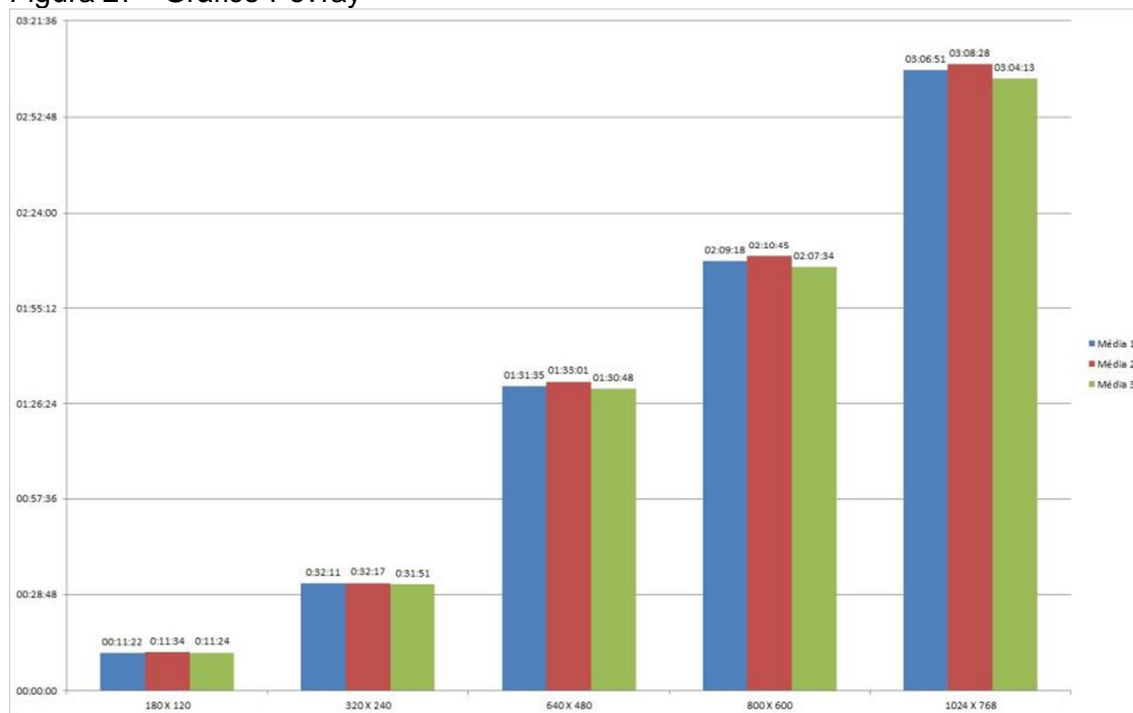
Figura 26 - Tabela Povray

Tamanho	180 X 120	320 X 240	640 X 480	800 X 600	1024 X 768
Cluster 1	00:11:22	0:32:11	01:31:35	02:09:18	03:06:51
Cluster 1	0:11:12	0:32:12	01:31:11	02:08:49	03:05:27
Cluster 2	0:11:56	0:32:22	01:34:52	02:12:42	03:11:29
Média	0:11:34	0:32:17	01:33:01	02:10:45	03:08:28
Cluster 1	0:10:56	0:30:59	01:28:26	02:05:59	03:01:54
Cluster 2	0:11:11	0:31:12	01:30:13	02:06:03	03:03:53
Cluster 3	0:12:05	0:33:22	01:33:45	02:10:40	03:06:51
Média	0:11:24	0:31:51	01:30:48	02:07:34	03:04:13

Fonte: Elaborada pelo Autor.

Na Figura 27 observa-se que em todos os testes ao trabalhar em conjunto o desempenho melhora bem pouco, com excessão do primeiro teste que não tem qualquer progresso significativo.

Figura 27 - Gráfico Povray



Fonte: Elaborada pelo Autor.

7 CONSIDERAÇÕES FINAIS

Baseando-se nas informações dos trabalhos correlatos de Danny Augusto Vieira Tonidandel que realizou o calculo de pi, unindo três computadores distintos, tendo um progresso de apenas um segundo e Robson Costa, que montou treze computadores iguais (excessão do servidor), para montarem uma determinada matriz, conseguindo um ganho de desempenho, a partir do décimo computador. Pode-se concluir que o funcionamento de um cluster do tipo beowulf depende, de dois fatores, os tipos de testes que serão realizados e a quantidade de computadores.

Perante aos resultados apresentados nesse trabalho, a união de vários computadores iguais, trabalhando em uma determinada função, demonstrou que dependendo da aplicação, o desempenho oscila. O único, teste que teve um ganho de processamento, foi renderizar uma imagem utilizando o programa povray, levando em consideração o pouco progresso que teve, não é viável construir um cluster com poucos computadores em relação ao custo beneficio, pois se aumenta-se o número de computadores. Pelas análises realizadas, com certeza haveria um progresso maior em renderizar imagens.

Levando todas as informações contidas nesse trabalho, tudo dependerá da aplicação do computador, assim, talvez compense adquirir um único computador potente, do que vários inferiores trabalhando em conjunto, mas em outras situações, compensará mais realizar a montagem de um cluster do tipo beowulf, do que adquirir um único computador.

REFERÊNCIAS

- AGUILAR L. J. **Fundamentos de programação**: algoritmos, estruturas de dados e objetos. 3. ed. São Paulo: McGraw-Hill Interamericana do Brasil, 2008.
- ALECRIM. E. Cluster: conceito e características. **InfoWester**, [c2013]. Disponível em: <<http://www.infowester.com/cluster.php>>. Acesso em: 6 maio. 2015.
- BARBOSA M. A. da. Cabeamento de rede **ICEI PUC Minas**, [c2011] Disponível em: <<http://www.icei.pucminas.br/professores/marco/cabeamento-de-rede/>>. Acesso em: 1 Jun. 2015.
- BARDON F. **Magia Prática**: 10 passos para a iniciação. 1. ed. Rio de Janeiro: Linhas Tortas, 2011.
- BONAN, A. R. **Configurando e usando o sistema operacional Linux**. 2. ed. São Paulo: Futura, 2003.
- CARTER, N. **Teoria e problemas de arquitetura de computadores**. Porto Alegre: Bookman, 2003.
- CASTELIANO, A. Introdução ao SSH. **O Linux**, [2001]. Disponível em: <http://olinux.uol.com.br/artigos/372/print_preview.html>. Acesso em: 1 Jun. 2015.
- CHINARELLI, M. A. A Evolução da Tecnologia através dos tempos. **TI Especialistas**, [c2012] Disponível em: <<http://www.tiespecialistas.com.br/2012/09/a-evolucao-da-tecnologia-atraves-dos-tempos/>>. Acesso em: 2 Jun. 2015.
- CISNEIROS, H. **Devin**. [c2008]. Disponível em: <<http://www.devin.com.br/descompactar-compactar-linux/>> Acesso em: 25 nov. 2015.
- COSTA. R. **Análise de Desempenho de um Algoritmo Paralelo Implementado em um Cluster Beowulf**. [2002]. 70 f. Monografia (Graduação em Bacharel em Informática) – Universidade do Planalto Catarinense. Disponível em: <<http://user.das.ufsc.br/~robson/arquivos/tcc.pdf>>. Acesso em: 1 Jun. 2015.
- COMPACTANDO e descompactando no Linux nos formatos: tar, tar.gz, tar.bz2, zip e rar. **Melhor Web**. [c2010]. Disponível em: < <http://www.melhorweb.com.br/artigo/27-Compactando-e-descompactando-no-Linux-nos-formatos--tar--tar-gz--tar-bz2--zip-e-rar.htm>> Acesso em: 25 nov. 2015.
- DYSON, P.; COLEMAN, P.; GILBERT, L. **O ABC da Intranet**. São Paulo: Makron Books do Brasil, 1999.
- ENGLANDER. I. **A arquitetura de hardware computacional, software de sistema e comunicação em rede**: uma abordagem da tecnologia da informação. 4. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2011.

ENTENDA o que é o código-fonte de um programa. **Folha de São Paulo**, São Paulo, 20 ago. 2001. Disponível em: <<http://www1.folha.uol.com.br/folha/informatica/ult124u7618.shtml>>. Acesso em: 1 Jun. 2015.

FOROUZAN B. A. **Protocolo TCP/IP**. 3. ed. São Paulo: McGraw-Hill Interamericana do Brasil, 2008.

GUIA Oficial do Usuário Linux Mint Cinnamon Edition. **Linux Mint**, [2015]. Disponível em: <http://www.linuxmint.com/documentation/user-guide/Cinnamon/portuguese_brazil_17.1.pdf>. Acesso em: 20 jun. 2015.

HENNESSY, J. L.; PATTERSON, D. A. **Arquitetura de computadores**: uma abordagem quantitativa. 4. ed. Rio de Janeiro: Elsevier, 2008.

INTRODUCTION to POV-RAY. **POVRAY** [c2008]. Disponível em: <<http://www.povray.org/documentation/view/3.6.1/1/>> Acesso em: 25 nov. 2015.

JONES, T. Sistemas de arquivos de Rede em Linux: NFS: Útil como sempre e ainda em evolução, **IBM**, [c2010] Disponível em: <<http://www.ibm.com/developerworks/br/library/l-network-file-systems/>>. Acesso em: 1 Jun. 2015.

KEDVES, D. Z. Using RarCrack. **RarCrack!**, [c2007]. Disponível em: <<http://rarcrack.sourceforge.net/>> Acesso em: 22 Nov. 2015.

MACHADO, F. B.; MAIA, L. P. **Arquitetura de sistemas operacionais**. 5. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2013.

MONTEIRO, M. A. **Introdução à organização de computadores**. 5. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2007.

MORIMOTO, E. C. **Linux Guia Prático**. Porto Alegre: Meridional, 2009.

_____. Compartilhamentos em NFS. **Hardware**, [c2006]. Disponível em: <<http://www.hardware.com.br/livros/entendendo-linux/compartilhamentos-nfs.html>>. Acesso em: 1 Jun. 2015.

NULL, L.; LOBUR, J. **Princípios básicos de arquitetura e organização de computadores**. 2. ed. Porto Alegre: Bookman, 2010.

OLIVEIRA, A. S.; ANDRADE, F. S. **Sistemas embarcados**: hardware e firmware na prática. 2. ed. São Paulo: Érica, 2012.

OLIVEIRA, R. D. S.; CARISSIMI, A. D. S.; TOSCANI, S. S. **Sistema Operacionais**. 4. ed. Porto Alegre: Bookman, 2010.

PÁDUA, E, M, M. de, **Metodologia da pesquisa**: Abordagem teórico-prática. 13º ed. Campinas: Papyrus Editora, 2007.

PARHAMI, B. **Arquitetura de Computadores**: de microprocessadores a supercomputadores. São Paulo: McGraw-Hill Interamericana do Brasil, 2008.

POV-RAY. **Baixaki**. [c2008] Disponível em:
<<http://www.baixaki.com.br/download/pov-ray.htm>> Acesso em: 25 Nov. 2015.

RAPIDA evolução da tecnologia pessoal é desafio para setores de TI, **IDGNOW**, [c2011], Disponível em: < <http://idgnow.com.br/ti-corporativa/2011/03/21/rapida-evolucao-da-tecnologia-pessoal-e-desafio-para-setores-de-ti/#&panel1-1>>. Acesso em: 2 Jun. 2015.

SHAY, W. A. **Sistemas Operacionais**. São Paulo: Makron Books do Brasil, 1996.

SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE G. **Fundamentos de Sistemas Operacionais**. 8. ed. Teresópolis: Livros Técnicos e Científicos, 2010.

SOARES, W.; FERNANDES, G. **Linux Fundamentos**. São Paulo: Érica, 2013.

STALLINGS, W. **Arquitetura e organização de computadores**: projeto para o desempenho. 5. ed. São Paulo: Pearson Education do Brasil, 2002.

_____. **Redes e Sistemas de Comunicação de Dados**. 5. ed. Rio de Janeiro: Elsevier, 2005.

TANENBAUM A, S. **Organização Estruturada de Computadores**. 5. ed. Limão: Pearson Education do Brasil, 2010.

_____. **Redes de Computadores**. 4. ed. Rio de Janeiro: Elsevier, 2003.

TANENBAUM A, S.; STEEN, V. M. **Sistemas Distribuídos**: princípios e paradigmas. 2. ed. São Paulo: Pearson Education do Brasil, 2008.

TAYLOR, D.; ARMSTRONG, J. C. Jr. **Aprenda em 24 Horas Unix**. Rio de Janeiro: Campos, 1998.

TONIDANDEL. D. A. V, **Manual de Montagem de um Cluster Beowulf sob a Plataforma GNU/LINUX**, [2008]. 90 f. Monografia (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Ouro Preto Disponível em: <<http://www.em.ufop.br/cecau/monografias/2008/DANNY%20TONIDANTEL.pdf>>. Acesso em: 1 Jun. 2015.

TRAN G. Plant Demo. **Oyonale**, [c2002]. Disponível em:
<<http://www.oyonale.com/image.php?code=556&mode=info§ion=2002&lang=en>>. Acesso em: 5 Set. de 2015.

VASCONCELOS, L. **Hardware na Prática**. 2. ed. Rio de Janeiro: Laércio Vasconcelos Computação, 2007.

VIEIRA, V. Quebre senhas de arquivos .ZIP, .RAR ou .7z com a ferramenta RarCrack, **Seja Livre**, [c2012] Disponível em:< <http://sejalivre.org/quebre-senhas-de-arquivos-zip-rar-ou-7z-com-a-ferramenta-rarcrack/>> Acesso em: 22 Nov. 2015.

VOLKERDING, P.; JOHNSON, E. F.; REICHARD, K. **Programando para Linux**. São Paulo: Markon Books do Brasil, 1998.

WEBER, R. F. **Fundamentos de arquitetura de computadores**. 4. ed. Porto Alegre: Bookman, 2012.

APÊNDICE A – INSTALAÇÃO DO SISTEMA OPERACIONAL

A seguir será mostrado o passo a passo do procedimento de instalação do sistema operacional Linux Mint 32 Bits, que será feito em todos os computadores.

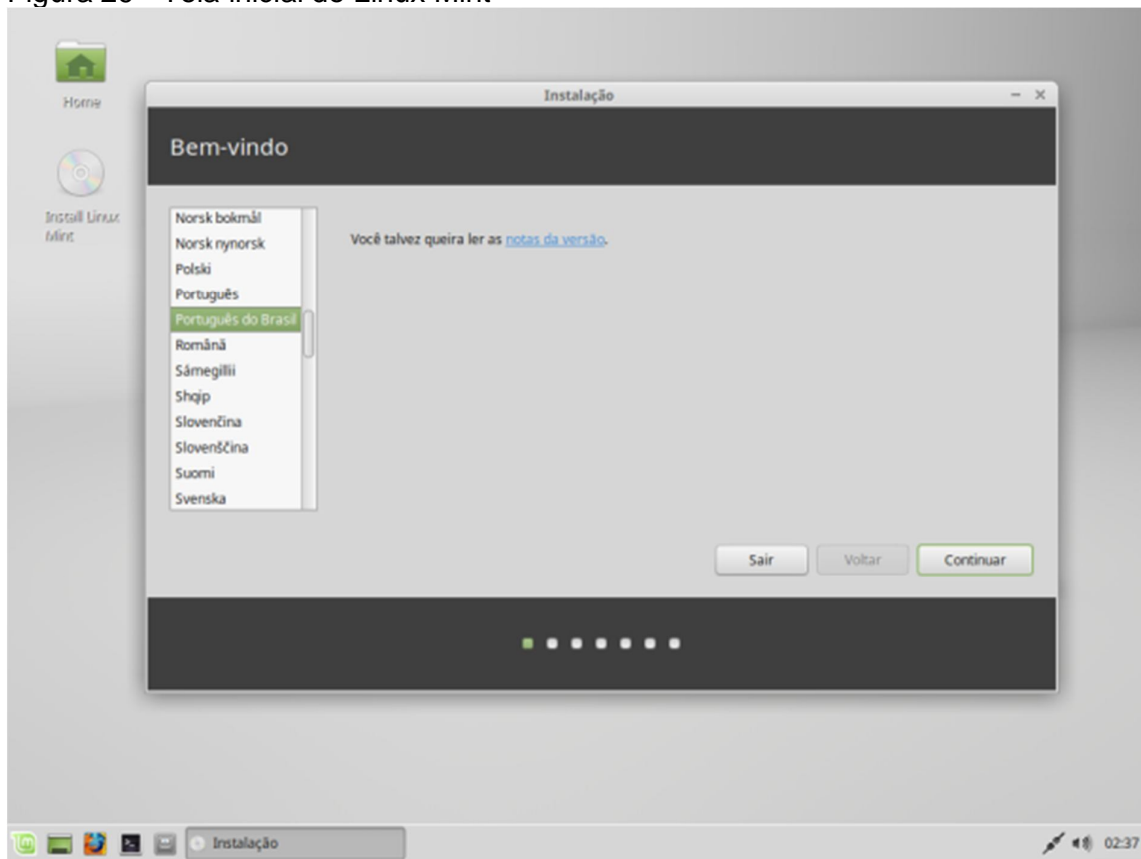
Figura 28 - Tela de boot do Linux Mint



Fonte: Elaborada pelo Autor.

Após gravar o sistema operacional em uma mídia de DVD, será inserida no cd-rom o disco de DVD, para ser executado, a o termino de seu carregamento, irá deparar-se com a tela da Figura 28. Nesta imagem há várias opções, será selecionado a primeira, para realizar a instalação padrão.

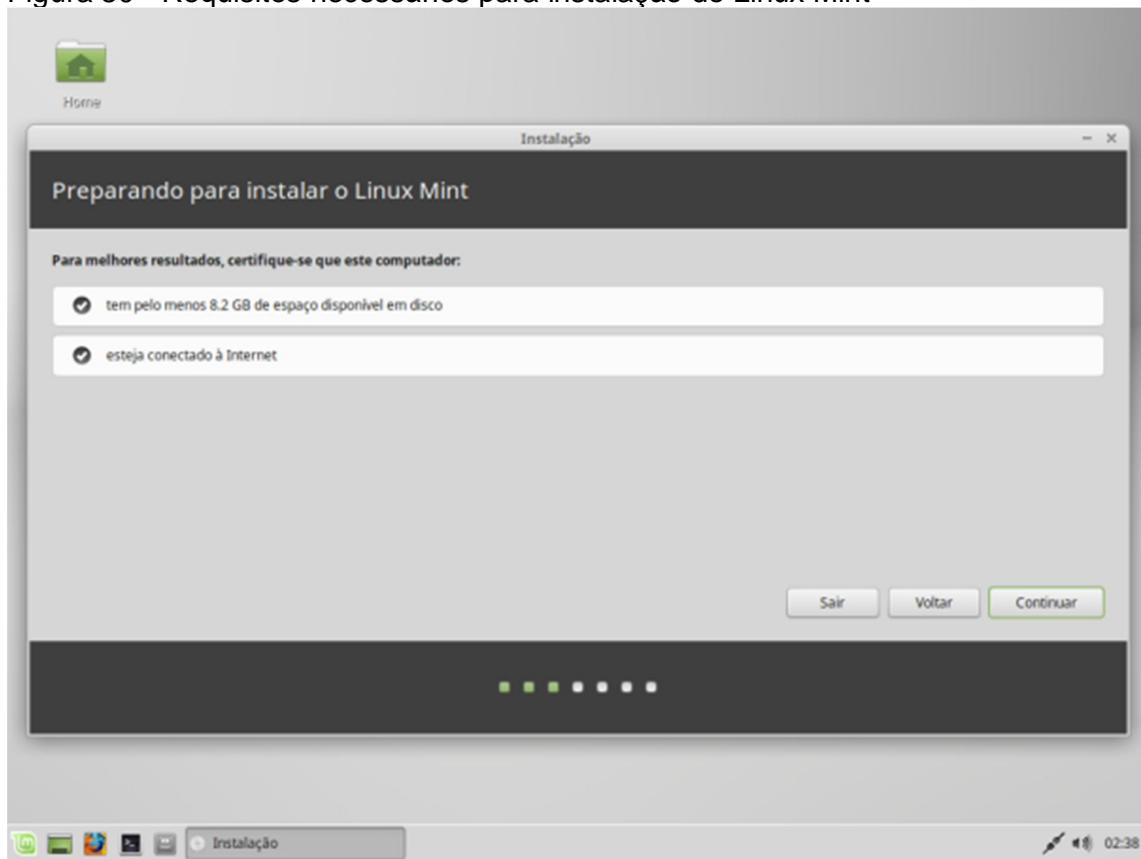
Figura 29 - Tela inicial do Linux Mint



Fonte: Elaborada pelo Autor.

A Figura 29 mostra o sistema inicializado, a partir disto será feita a instalação, abrindo o ícone que está na tela com o símbolo de um cd escrito "Install Linux Mint", assim que clicado com o mouse será aberta a janela para, escolher o idioma do sistema, será utilizado o "Português do Brasil", continuará a instalação pressionado o botão "Continuar".

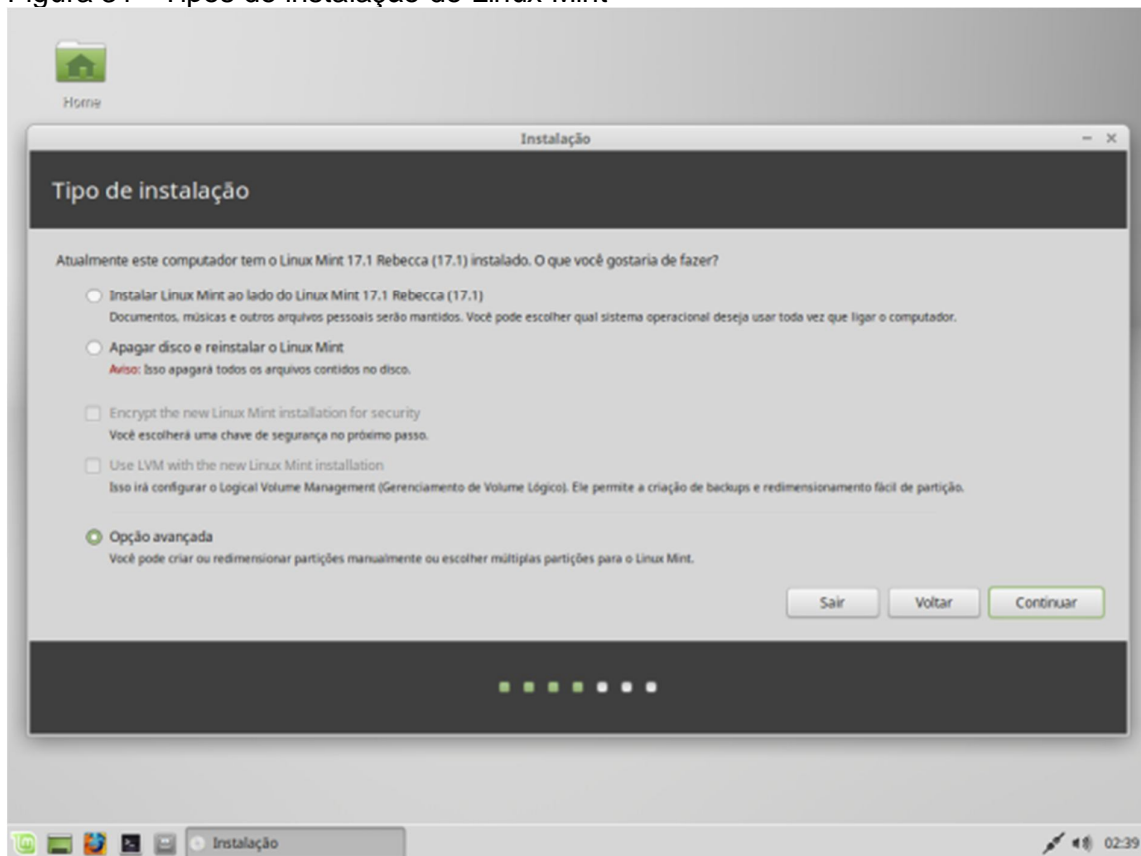
Figura 30 - Requisitos necessários para instalação do Linux Mint



Fonte: Elaborada pelo Autor.

A Figura 30 mostra que o computador será checado com relação aos requisitos mínimos de instalação, que são, espaço de 8.2 Gb livres em uma unidade de armazenamento, e requisito opcional a internet estar conectada, para atualizar os pacotes do sistema operacional, como programas, correções de erros do sistema. Continuará a instalação pressionado o botão "Continuar".

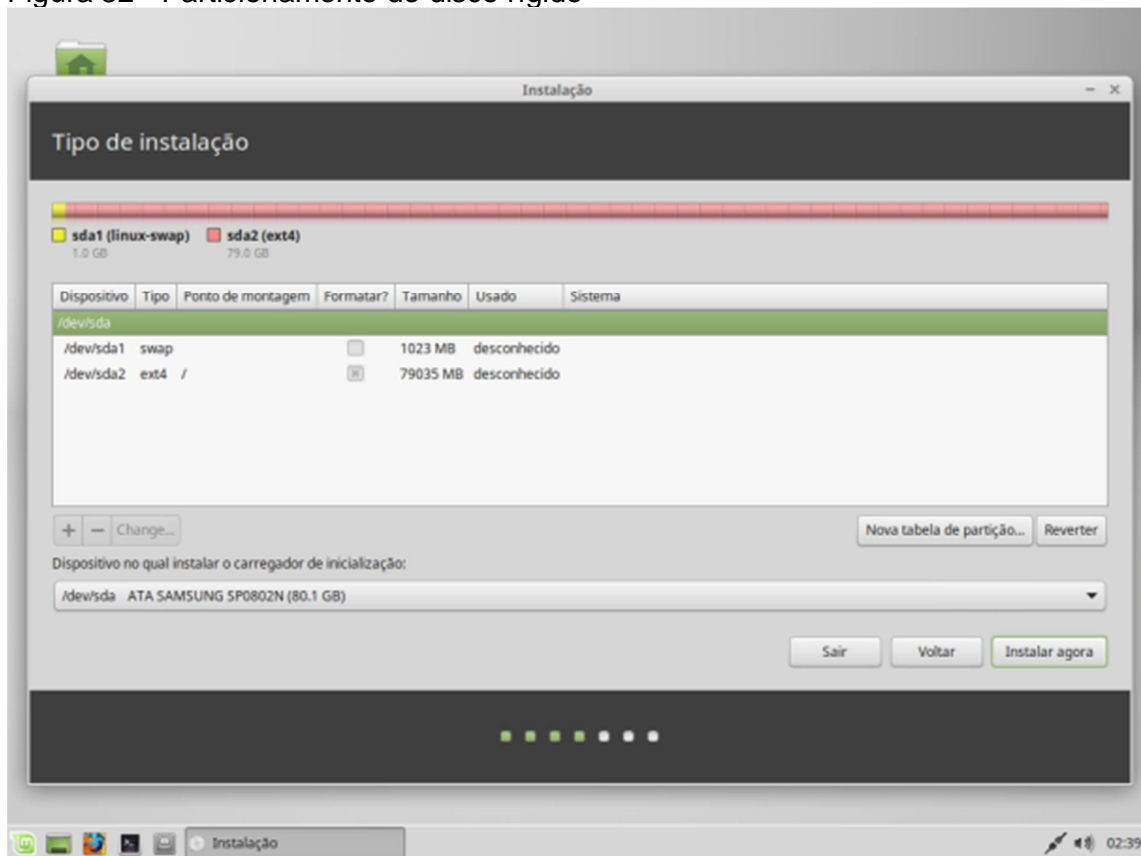
Figura 31 - Tipos de instalação do Linux Mint



Fonte: Elaborada pelo Autor.

A Figura 31 demonstra os tipos de instalação que podem ser feitos, será selecionado a opção “Opção Avançada”, para poder particionar a HD manualmente, continuará instalação pressionado o botão “Continuar”.

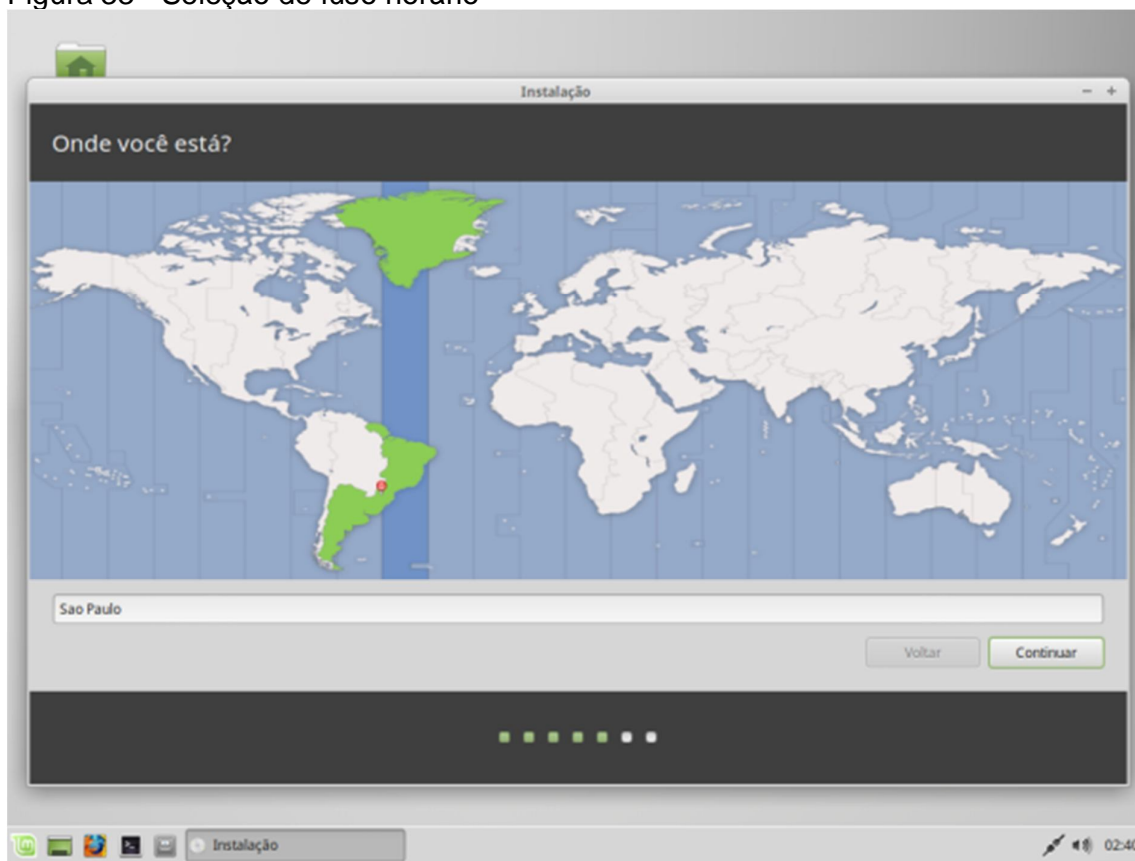
Figura 32 - Particionamento do disco rígido



Fonte: Elaborada pelo Autor.

Nesta etapa será particionado o disco rígido, como mostrado na Figura 32, separando 1 Gb para a partição swap e os 79 Gb restantes serão para o sistema operacional, escolhendo o tipo de partição ext4, feito o particionamento, a instalação do sistema operacional dará início, continuará instalação pressionado o botão “Instalar agora”.

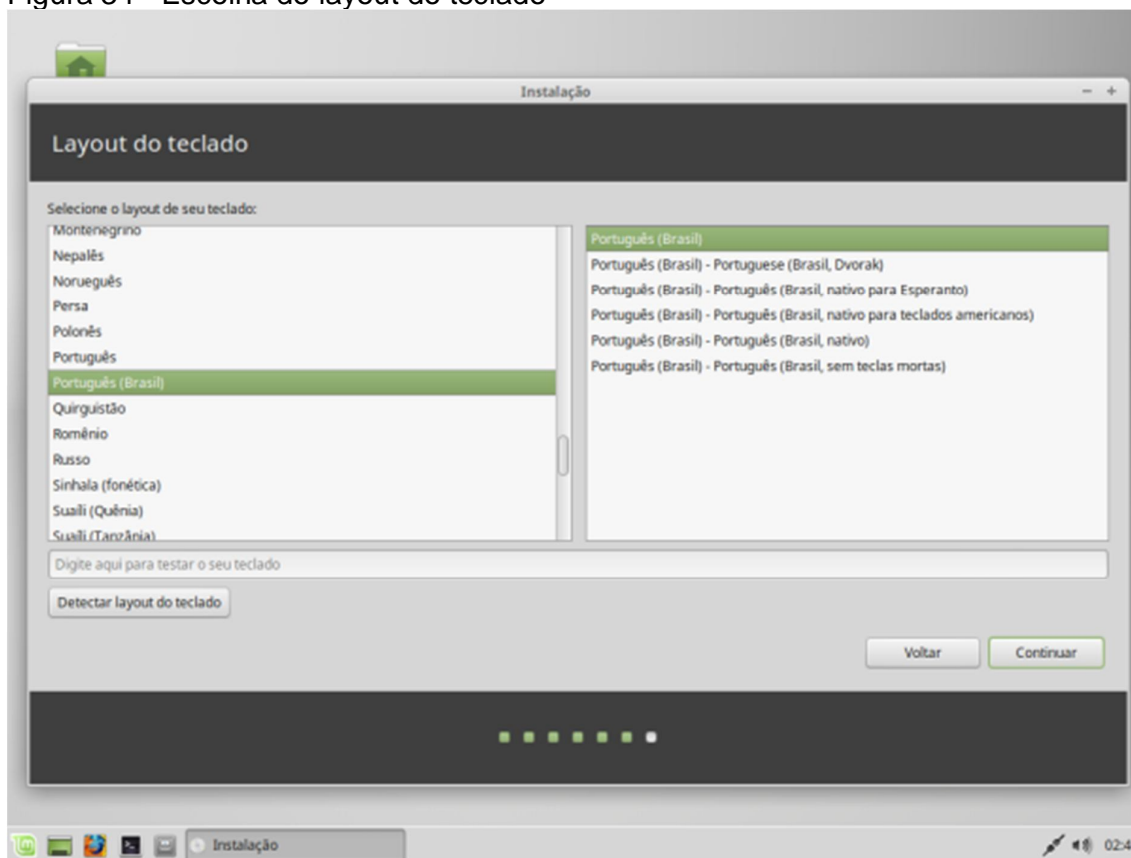
Figura 33 - Seleção de fuso horário



Fonte: Elaborada pelo Autor.

Neste momento da instalação é pedido que informe o fuso horário do relógio, a Figura 33 mostra que será escolhido o fuso horário da cidade de São Paulo, continuará a instalação pressionado o botão "Continuar".

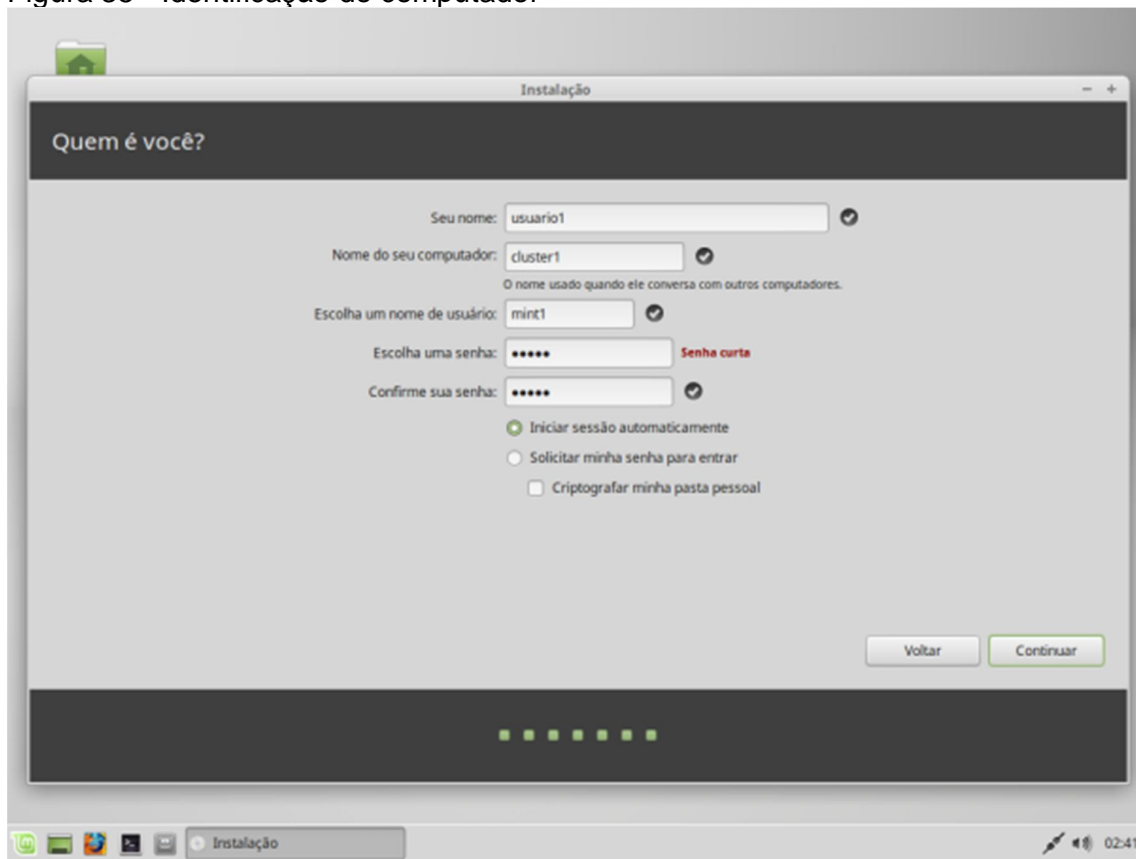
Figura 34 - Escolha do layout do teclado



Fonte: Elaborada pelo Autor.

Neste momento é solicitado, a configuração do teclado, para que todas as teclas funcione corretamente, foi escolhido a configuração Português (Brasil), como mostrado na Figura 34, continuará a instalação pressionado o botão “Continuar”.

Figura 35 - Identificação do computador



Fonte: Elaborada pelo Autor.

A Figura 35 mostra que são requeridos os dados referentes à identificação do computador, cada nó do cluster será configurado de uma maneira, como mostrado abaixo:

No1:

Nome: usuario1

Nome do computador: cluster1

Nome do usuário: mint1

Senha: linux

No2:

Nome: usuario2

Nome do computador: cluster2

Nome do usuário: mint2

Senha: linux

No3:

Nome: usuario3

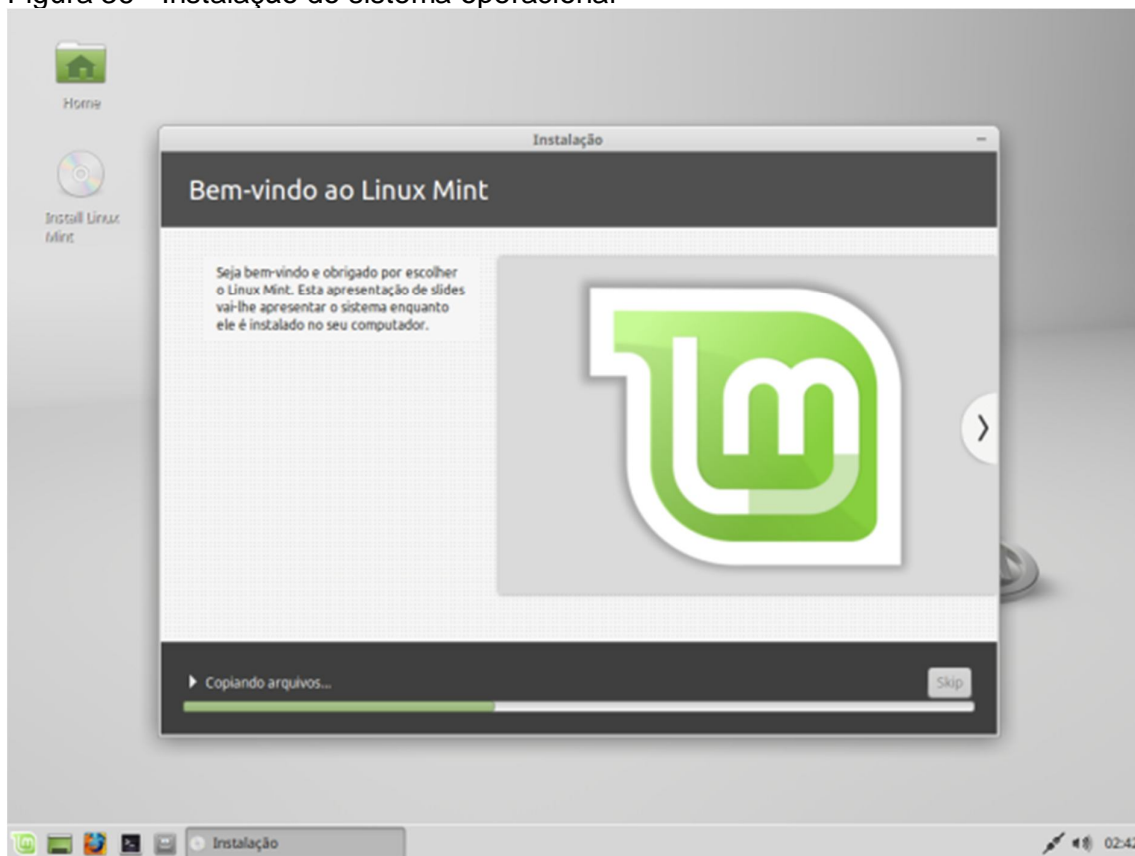
Nome do computador: cluster3

Nome do usuário: mint3

Senha: linux

Continuará a instalação pressionado o botão “Continuar”

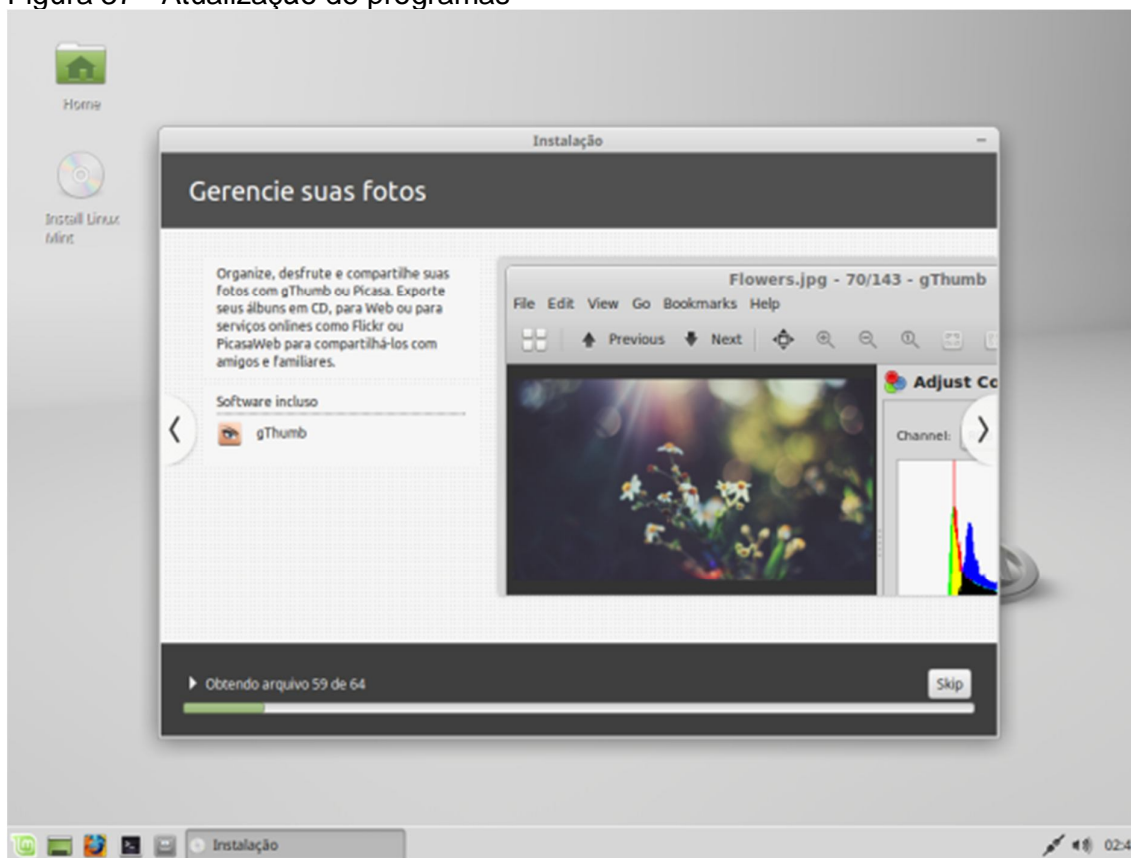
Figura 36 - Instalação do sistema operacional



Fonte: Elaborada pelo Autor.

Nesta etapa, aparecerão diversas informações do sistema operacional Linux Mint, enquanto ele vai sendo instalado, a Figura 36 demonstra esse processo.

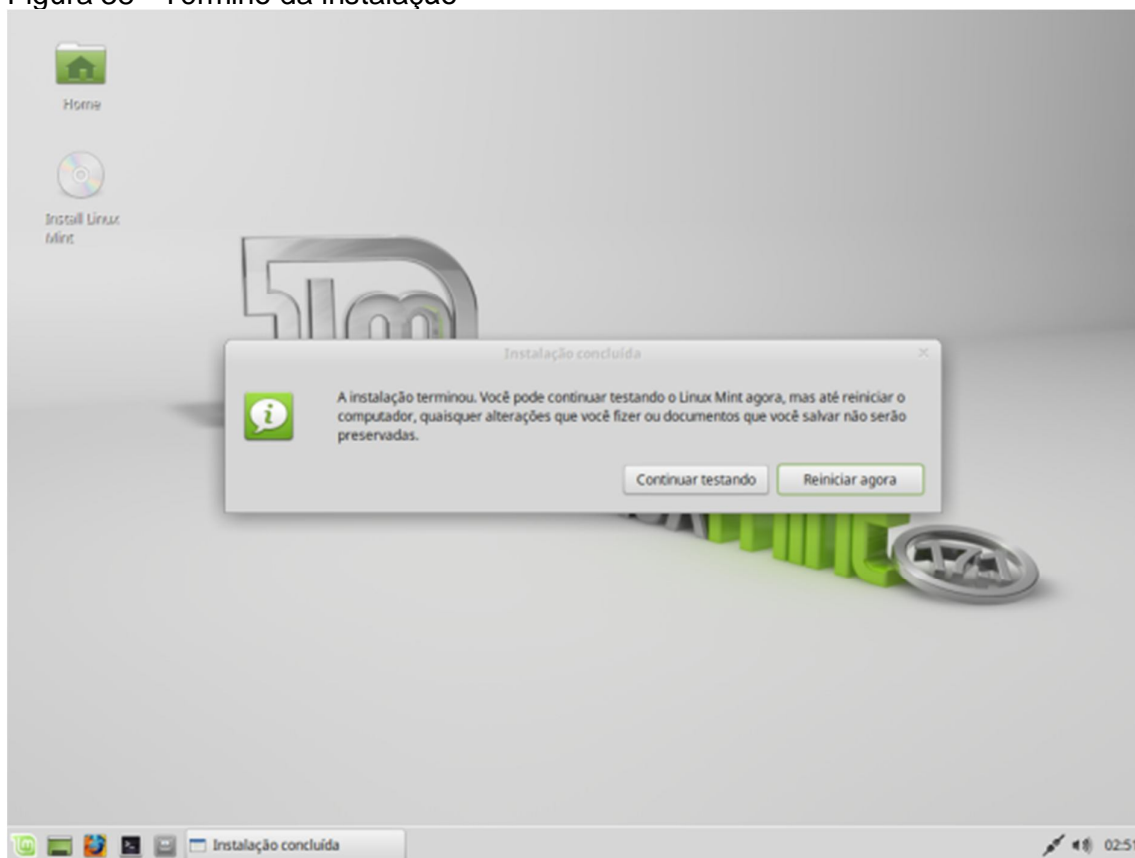
Figura 37 - Atualização de programas



Fonte: Elaborada pelo Autor.

Após instalar o sistema operacional, serão buscados na internet atualizações a o sistema inteiro, tanto correções de falhas, melhorias e atualizações dos programas que são instalados por padrão, a Figura 37 mostra a obtenção de atualizações.

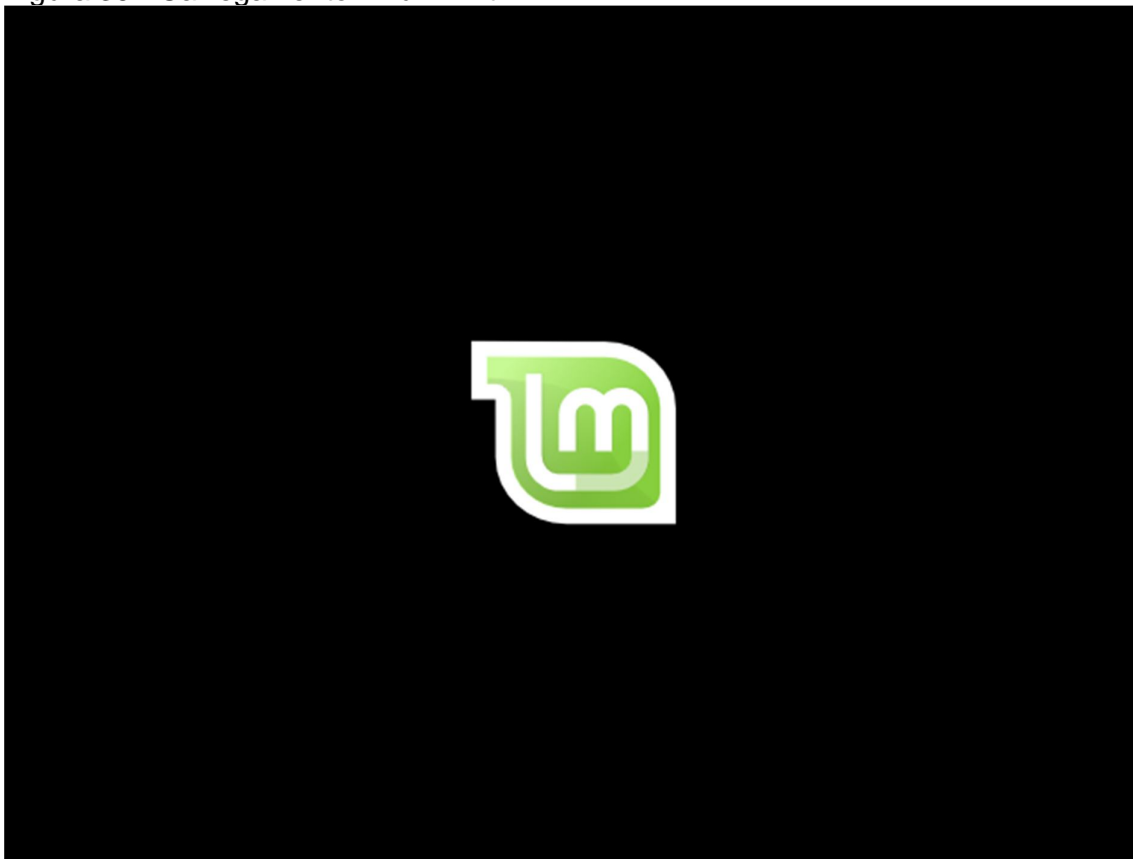
Figura 38 - Término da instalação



Fonte: Elaborada pelo Autor.

A Figura 38 mostra o término da instalação aparecerá a tela de “Instalação Concluída”, será reiniciado o sistema operacional, clicando no botão “Reiniciar agora”, para que possa ser acessado o sistema instalado.

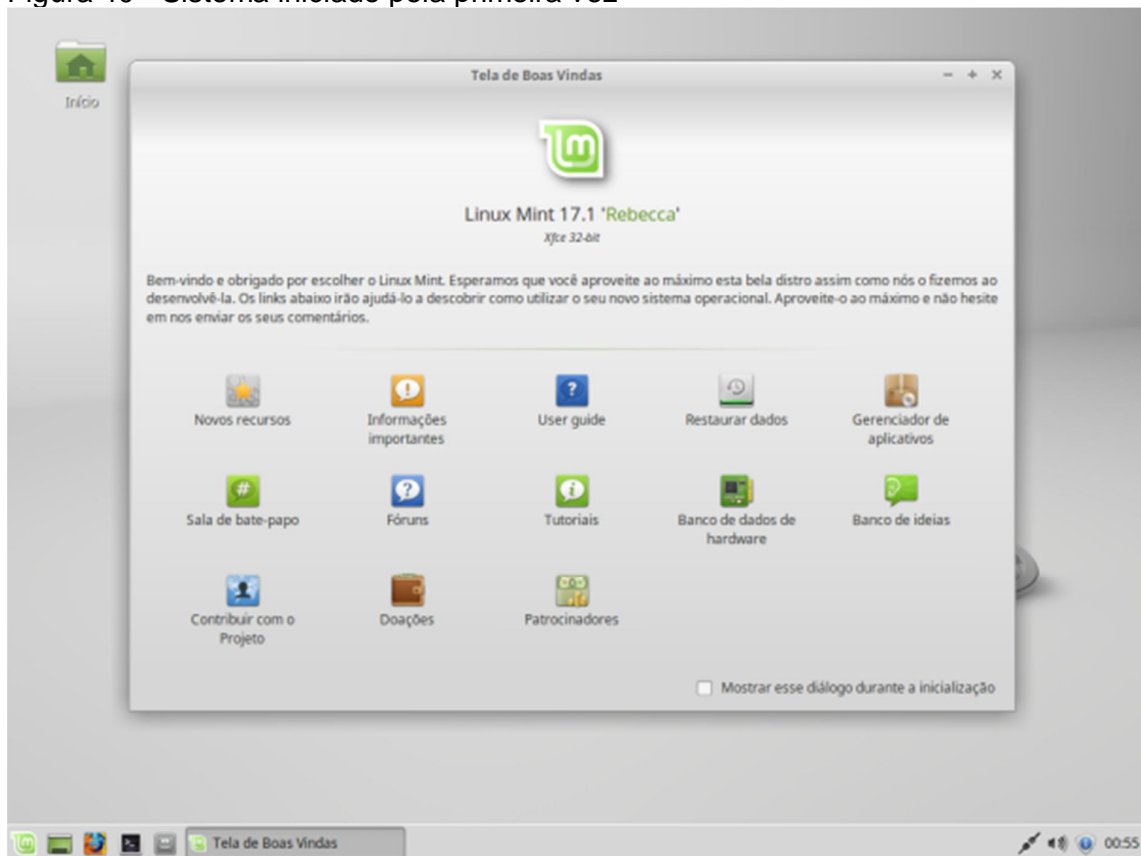
Figura 39 - Carregamento Linux Mint



Fonte: Elaborada pelo Autor.

Toda vez que o sistema operacional Linux Mint iniciar, aparecerá a Figura 39, demonstrando seu carregamento.

Figura 40 - Sistema iniciado pela primeira vez



Fonte: Elaborada pelo Autor.

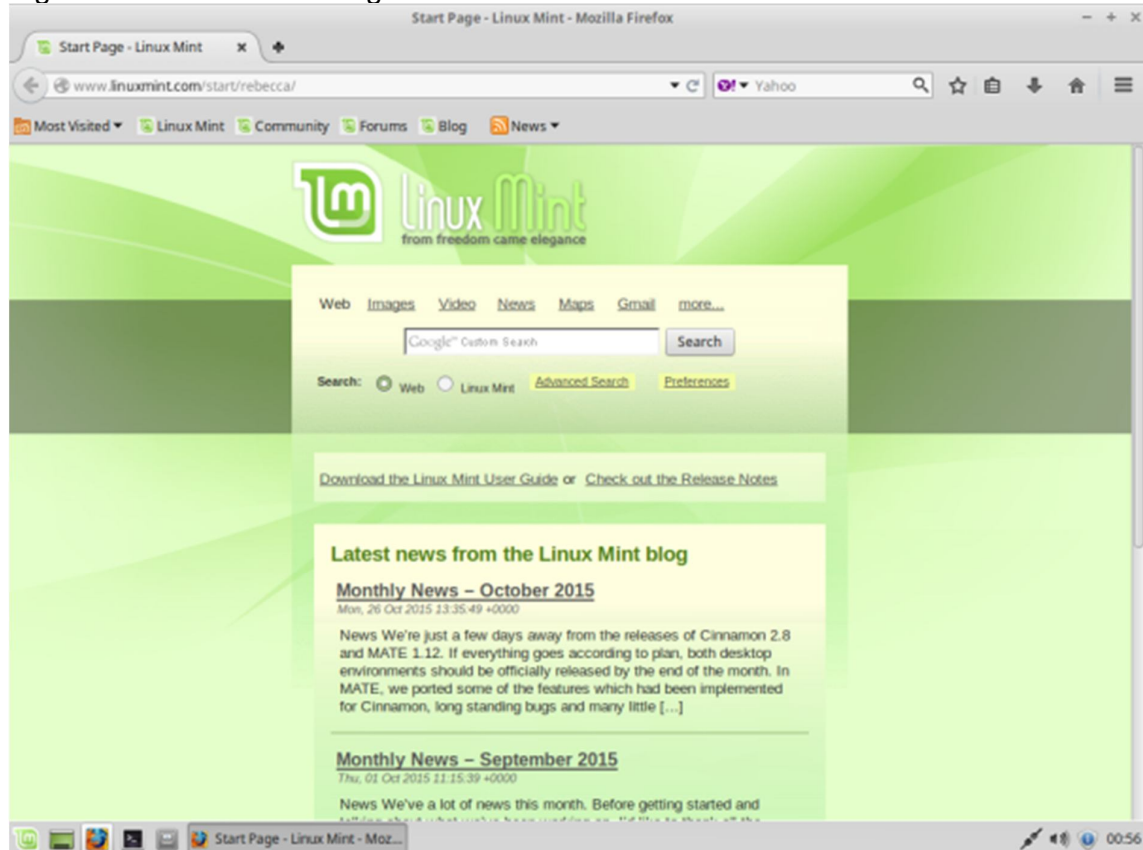
Quando o sistema for iniciado pela primeira vez aparecerá a “Tela de Boas Vindas”, será removida a opção “Mostrar esse dialogo durante a inicialização” para que, não apareça o mesmo toda vez que o sistema for iniciado e fechar a janela no “X” no canto direito superior da janela, a Figura 40 mostra a imagem.

Ao fim desta etapa, todos os computadores serão configurados de acordo com seu tipo, sendo servidor ou escravo.

APÊNDICE B – CONFIGURAÇÃO DO NÓ SERVIDOR

Será descrita todas as configurações do nó servidor.

Figura 41 - Abrindo o navegador



Fonte: Elaborada pelo Autor.

O Linux é case-sensitive, por isso os comandos precisam ser sempre digitados literalmente, respeitando as letras maiúsculas e minúsculas. (MORIMOTO, 2009, p. 11).

Será preciso abrir o navegador, clicando no ícone vermelho e azul na barra inferior da tela, como mostrado na Figura 41, para fazer o download dos programas: MPI², Povray³ e a imagem que será renderizada pelo povray⁴, serão salvos na área de trabalho. Abaixo estão seus respectivos links de download:

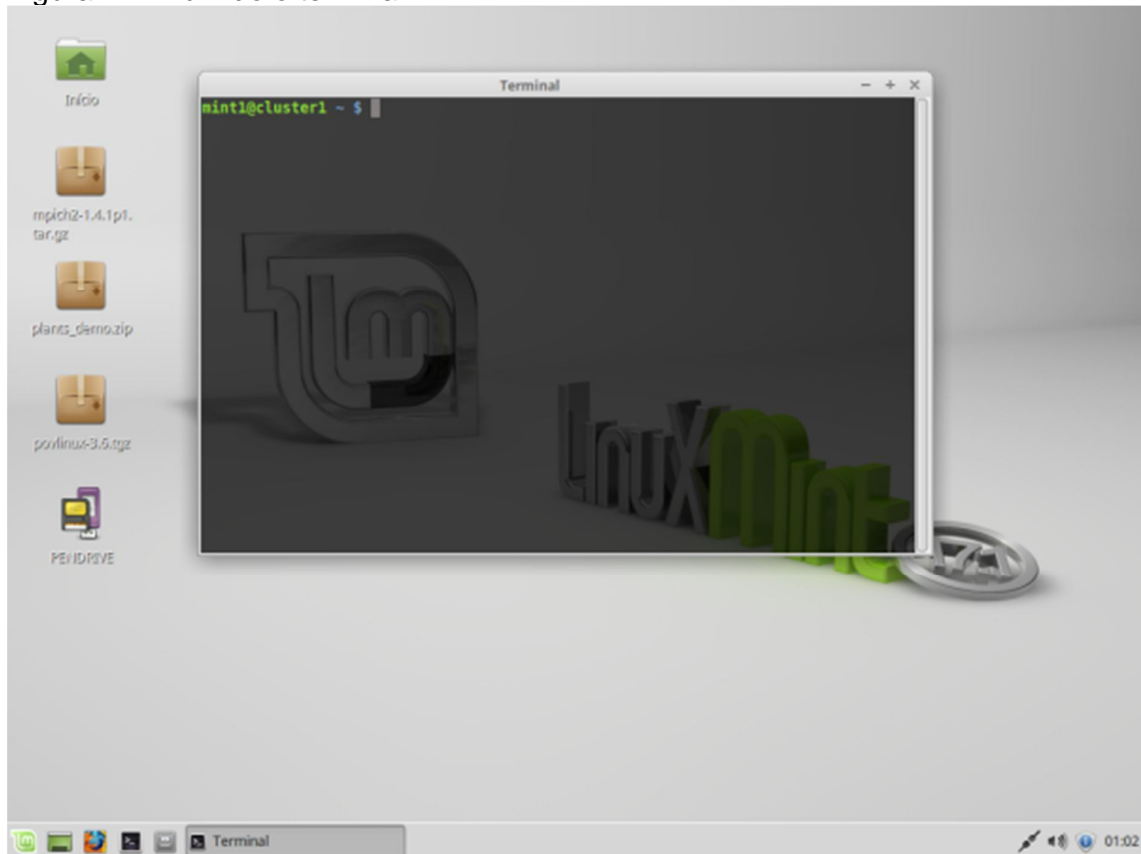
Terminando os downloads, será fechado o navegador, clicando no “X” no canto direito superior.

² <http://www.mpich.org/static/downloads/1.4.1p1/hydra-1.4.1p1.tar.gz>

³ <http://www.povray.org/redirect/www.povray.org/ftp/pub/povray/Old-Versions/Official-3.62/Linux/povlinux-3.6.tgz>

⁴ http://www.oyonale.com/downloads/plants_demo.zip

Figura 42 - Abrindo o terminal



Fonte: Elaborada pelo Autor.

Após plugar um pendrive no computador, será aberto o terminal, clicando no ícone de fundo preto, na barra inferior da tela, como mostrado na Figura 42, todos os procedimentos realizados a seguir no terminal, será feito um comentário, explicando o que será desenvolvido, em seguida, será dado um comando que estará em negrito, teclando enter após digita-lo.

Entrar como administrador do sistema, este pedirá a senha, que é linux:

su

(digitar senha linux)

O comando apt-get é o instalador de programas deste sistema, fazendo download dos programas e suas dependências automaticamente, para certificar que todos os programas estejam atualizados, será dado o comando de atualização.

apt-get update

Será realizado o download dos programas: OpenSSH, NFS, Rarcrack e dependências do MPI que é o Gfortran, que serão utilizados no cluster. Após ordenar o comando de instalação abaixo, será preciso confirmar a instalação dos mesmo, digitando “yes”, quando solicitar.

apt-get install build-essential gfortran nfs-kernel-server ssh rarcrack

Criar uma pasta comum para todos os computadores terem acesso a os arquivos que serão manipulados, então será preciso ir na raiz do sistema.

cd /

Criar a pasta testes.

mkdir testes

Mover para está pasta, os arquivos que foram feito download. Acessando a área de trabalho, aonde estes se encontram.

cd /home/mint1/Área\ de\ Trabalho

Mover os arquivos para a pasta testes.

mv -v mpich2-1.4.1p1.tar.gz povlinux-3.6.tgz plants_demo.zip /testes

Mover para esta pasta, as isos que estão no pendrive.

Acessando o local do pendrive.

cd /media/mint1/pendrive

Mover os arquivos para a pasta testes.

mv -v a.iso b.iso c.iso d.iso e.iso /testes

Após mover os arquivos do pendrive, o mesmo será removido do computador.

Compartilhar a pasta comum com os demais computadores.

echo '/testes *(rw,sync,no_root_squash) >> /etc/exports

Para os nós escravos poderem executar os comandos que o servidor ordena é preciso que tenham permissão de executar os programas, será criada uma chave de autorização.

Será preciso ir à pasta do administrador do sistema.

cd /root

Gerar a chave sem senha, do tipo rsa, teclando enter três vezes seguidas, quando solicitado.

ssh-keygen -b 1024 -t rsa

Ao gerar a chave, automaticamente cria-se uma pasta denominada .ssh, será preciso “entrar” nela.

cd .ssh

Gerar um arquivo, autorizando a chave pública criada.

cat id_rsa.pub >> authorized_keys

Copiar a chave autorizada para a pasta de testes, para todos os nós aceitar esta chave.

cat authorized_keys >> /testes/authorized_keys

Configurar o OpenSSH, para aceitar a chave, gerada, permitindo a senha em branco como foi criada.

Para isso o comentário que é representado pelo símbolo “#”, será removido das seguintes linhas:

```
#AuthorizedKeysFile      %h/.ssh/authorized_keys  
#PasswordAuthentication yes
```

Também será preciso mudar linha abaixo para “yes”, permitindo a senha em branco:

```
PermitEmptyPasswords no
```

nano /etc/sshd_config

```
AuthorizedKeysFile %h/.ssh/authorized_keys  
PermitEmptyPasswords yes  
PasswordAuthentication yes
```


Salvar e sair do nano, teclando primeiro CTRL + X, depois S, para aceitar as alterações.

Será preciso alterar outro arquivo para que aceite a senha do tipo RSA, como foi gerada, bastando apenas remover o comentário " #" da linha:

```
# RSAAuthentication yes
```

```
nano /etc/ssh_config
```

```
RSAAuthentication yes
```

Salvar e sair do nano, teclando primeiro CTRL + X, depois S, para aceitar as alterações.

Agora será instalado o MPI.

Entrar na pasta de testes.

```
cd /testes
```

Criar uma pasta, para instalar o MPI, esta pasta se chamará "mpi".

```
mkdir mpi
```

Descompactar o programa MPI.

```
tar -zxf mpich2-1.4.1p1.tar.gz
```

Entrar nessa pasta que foi descompactada.

```
cd mpich2-1.4.1p1
```

Configurar o MPI para instalar na pasta mpi.

```
./configure --prefix=/testes/mpi --with-pm=hydra
```

```
make
```

```
make install
```

Configurar o MPI, para executar sua função assim que comandada.

```
echo ' /testes Export PATH=/testes/mpi/bin:$PATH
```

```
LD_LIBRARY_PATH=/testes/mpi/lib:$LD_LIBRARY_PATH '
```

```
>> /testes/.bashrc
```

Para não precisar executar toda vez o comando source, para poder utilizar o MPI, será copiado para a pasta do administrador, sendo executado automaticamente.

Ir até a pasta testes

```
cd ..
```

Copiar o arquivos para a pasta do administrador.

```
cp -p .bashrc /root
```

Para os demais computadores poderem trabalhar com o MPI será preciso alterar a permissão da pasta mpi.

```
chmod 777 mpi
```

Agora será configurado a rede para interligar os nós, a o termino a internet não funcionará mais, devido a está configuração.

Alterar o arquivo interfaces, ficando da seguinte maneira:

```
nano /etc/network/interfaces
```

```
auto eth0
iface eth0 inet static
address 192.168.0.11
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1
```

Salvar e sair do nano, teclando primeiro CTRL + X, depois S, para aceitar as alterações.

Configurar para que todos os nós sejam reconhecidos na rede, comentando com # a linha: 127.0.0.1 cluster1.

```
nano /etc/hosts
```

```
#127.0.0.1 cluster1
192.168.0.11 cluster1
192.168.0.12 cluster2
192.168.0.13 cluster3
```

Salvar e sair do nano, teclando primeiro CTRL + X, depois S, para aceitar as alterações.

Gerar a partir do arquivo `authorized_keys`, os arquivos compactados em formato zip com senha, para poder fazer testes futuramente, com o programa Rarcrack, os próprios nomes dos arquivos farão referência a sua própria senha, `jjj.zip` terá a senha `jjj`, assim sucessivamente.

```
zip -re jjj.zip authorized_keys
zip -re ttt.zip authorized_keys
zip -re DDD.zip authorized_keys
zip -re NNN.zip authorized_keys
zip -re WWW.zip authorized_keys
```

Criar várias pastas para armazenar os arquivos que serão manipulados futuramente:

pasta de nome: `senha` para guardar os arquivos compactados em zip com senha.

pasta de nome: `bzip` para guardar os arquivos compactados em bzip2.

pasta de nome: `zip` para guardar os arquivos compactados em zip.

pasta de nome: `copia` para guardar os arquivos copiados.

pasta de nome: `imagem` para guardar a imagem que será renderizada.

pasta de nome: `iso` para guardar as isos.

```
mkdir senha bzip zip copia imagem iso
```

Mover para a pasta `senha` os arquivos compactados com senha, no formato zip.

```
mv -v jjj.zip ttt.zip DDD.zip NNN.zip WWW.zip
```

Mover para a pasta `iso` os arquivos em formato iso.

```
mv -v a.iso b.iso c.iso d.iso e.iso iso/
```

Mover o arquivo para a pasta imagem, o arquivo da imagem que foi feito o download.

```
mv -v plants_demo.zip
```

Entrar na pasta imagem.

```
cd imagem
```

Descompactar a imagem.

```
unzip -o plants_demo.zip
```

Voltar a pasta de testes.

```
cd ..
```

Descompactar o Povray.

```
tar -xzf povlinux-3.6.tgz
```

Entrar na pasta do povray.

```
cd povray-3.6
```

Instalar o povray.

```
./install
```

Voltar a pasta de testes.

```
cd ..
```

Dar permissão de acesso a todos os nós.

```
chmod 777 *
```

Terminando estas configurações, será preciso reiniciar o sistema, e configurar os nós escravos.

APÊNDICE C – CONFIGURAÇÃO DOS NÓS ESCRAVOS

Será descrita todas as configurações dos nós escravos.

Entrar como administrador do sistema, este pedirá a senha, que é linux:

su

(digitar senha linux)

Atualizar todos os programas existentes no sistema.

apt-get update

Fazer o download dos programas: OpenSSH, NFS, Rarcrack e dependências do MPI que é o Gfortran, que serão utilizados no cluster.

apt-get install build-essential gfortran nfs-kernel-server ssh rarcrack

Configurar a pasta testes criado no servidor aonde estão os arquivos para rodar o cluster.

echo 'cluster1:/testes /testes nfs defaults 0 0 ' >> /etc/fstab

Configurar a rede para interligar os nós, a o termino, a internet não funcionará mais, devido a está configuração, o nó2 será configurado com as informações do cluster2 assim o nó3 com suas respectivas informações do cluster3.

Alterar o arquivo interfaces:

nano /etc/network/interfaces

cluster 2

auto eth0

iface eth0 inet static

address 192.168.0.12

netmask 255.255.255.0

network 192.168.0.0

broadcast 192.168.0.255

gateway 192.168.0.1

cluster 3

auto eth0

iface eth0 inet static

```
address 192.168.0.13
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1
```

Salvar e sair do nano, teclando primeiro CTRL + X, depois S, para aceitar as alterações.

Configurar para que todos os nós sejam reconhecidos na rede, comentando com # a linha: 127.0.0.1 cluster2 se for o nó2 e 127.0.0.1 cluster3 se for o nó3.

nano /etc/hosts

```
#127.0.0.1 cluster2 ou #127.0.0.1 cluster3
192.168.0.10 cluster1
192.168.0.11 cluster2
192.168.0.12 cluster3
```

Salvar e sair do nano, teclando primeiro CTRL + X, depois S, para aceitar as alterações.

Montar a pasta de testes.

mount -a

Para certificar que a pasta de testes tenha todos os arquivos, será reiniciado o sistema.

shutdown -r now

O carregamento do sistema irá demorar um pouco até que seja sincronizado com o cluster1, ao termino do carregamento, o terminal será aberto novamente.

Para dar continuidade às configurações, será preciso entrar como administrador novamente.

su

(digitar senha linux)

Entrar na pasta do administrador do sistema.

cd /root

Criar a pasta .ssh para armazenar a chave autorizada do nó servidor.

mkdir .ssh

Copiar a chave autorizada para dentro da pasta .ssh.

cp -p /testes/authorized_keys /root/.ssh

Configurar o OpenSSH, para aceitar a chave, gerada, permitindo a senha em branco como foi criada.

Para isso o comentário que é representado pelo símbolo “#”, será removido das seguintes linhas:

```
#AuthorizedKeysFile      %h/.ssh/authorized_keys  
#PasswordAuthentication yes
```

Tambem será preciso mudar linha abaixo para “yes”, permitindo a senha em branco:

```
PermitEmptyPasswords no
```

nano /etc/sshd_config

```
AuthorizedKeysFile %h/.ssh/authorized_keys  
PermitEmptyPasswords yes  
PasswordAuthentication yes
```

Salvar e sair do nano, teclando primeiro CTRL + X, depois S, para aceitar as alterações.

Será preciso alterar outro arquivo para que aceite a senha do tipo RSA, como foi gerada, bastando apenas remover o comentário “#” da linha:

```
# RSAAuthentication yes
```

nano /etc/ssh_config

```
RSAAuthentication yes
```

Salvar e sair do nano, teclando primeiro CTRL + X, depois S, para aceitar as alterações.

Entrar na pasta do povray.

```
cd povray-3.6
```

Instalar o povray.

```
./install
```

Terminando essas configurações os computadores estarão interligados, em pleno funcionamento de cluster, assim que o programa MPI for acionado.