



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

FERNANDO LUIZ DE SIQUEIRA CARDOSO
TIAGO ANTERO DE SOUSA ALVES

**IMPLEMENTAÇÃO DE UM CLUSTER DE ALTO DESEMPENHO PARA
APLICAÇÕES EM ENGENHARIA NAVAL**

Belém
2018

FERNANDO LUIZ DE SIQUEIRA CARDOSO
TIAGO ANTERO DE SOUSA ALVES

**IMPLEMENTAÇÃO DE UM CLUSTER DE ALTO DESEMPENHO PARA
APLICAÇÕES EM ENGENHARIA NAVAL**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação da Faculdade de Computação do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará como pré-requisito para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Raimundo Viégas Junior.

Belém

2018

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

FERNANDO LUIZ DE SIQUEIRA CARDOSO
TIAGO ANTERO DE SOUSA ALVES

**IMPLEMENTAÇÃO DE UM CLUSTER DE ALTO DESEMPENHO PARA
APLICAÇÕES EM ENGENHARIA NAVAL**

Data da Defesa:

21/12/2018 Conceito: _____

Banca Examinadora

Prof. MSc. André Vinícius da Costa Araújo

Prof. Dr. Josivaldo de Souza Araújo

Prof. Dr. Raimundo Viégas Junior

Faculdade de Computação/UFPA – Orientador

Belém

2018

AGRADECIMENTOS

Agradeço à minha família, pelo apoio e compreensão a todo esforço aplicado a este trabalho.

Aos meus amigos Pilantras, que sempre estavam lá me cobrando e não me deixando pensar em desistir.

Meus amigos da Confraria da Cruz da Caravaca, que sempre estavam lá para me fazer rir em meio as crises e dividir todo o ódio acumulado do trabalho.

Ao Professor Viégas, que não desistiu de mim e do meu tema, mesmo após todo o tempo de desenvolvimento deste trabalho.

Ao Professor André, por me incentivar a continuar na elaboração dos testes, fornecendo todo o *hardware*, *software* e acesso ao espaço do laboratório de engenharia naval, mesmo após eu me desligar e começar a trabalhar fora da instituição.

À todos meus amigos do curso de Sistemas de Informações, que sempre me ajudaram fornecendo material de pesquisa, tirando dúvidas, ou simplesmente criando momentos de confraternização, para renovar as forças.

Aos alunos de Engenharia Naval, que forneceram material para os testes presentes nesse TCC, compartilhando da empolgação da implementação do cluster para o laboratório.

Em especial, aos dois melhores alunos de Engenharia Naval, que receberam o título honorável de “Mais Legais da Naval”, Eduardo e Zilmar, respectivamente.

Ao Tiago, minha dupla de TCC, que mesmo em trabalhos diferentes, arrumávamos tempos de continuarmos.

Fernando Luiz De Siqueira Cardoso

“Ergamos um brinde aos amigos ausentes, amores perdidos, velhos deuses e à Estação das Brumas, e que cada um de nós sempre conceda ao diabo aquilo o que lhe é merecido”.

Neil Gaiman

AGRADECIMENTOS

A Deus que é o senhor das nossas vidas e que nunca nos abandonou, mesmo nas grandes tempestades.

A esta Universidade, docentes e demais servidores por nos dar a oportunidade de expandir nossos horizontes através de um ensino de qualidade.

Ao Professor Raimundo Viégas Junior, nosso orientador, pela dedicação, paciência e por ter aceitado este desafio.

Ao Fernando, minha dupla de TCC, que mesmo com todas as dificuldades conseguimos chegar nesse momento.

Ao meus amigos Wanderson(guru) e Adriana que acreditaram em mim em momentos que nem eu acreditava mais.

A minha namorada Thais pela paciência, incentivo e dedicação para vencer mais este desafio em minha vida.

A minha família, meus irmão e principalmente a minha mãe Euza que além de ser mãe foi um pai criando quatro filhos sozinha.

Em memória do meu pai Valdevino, pelos ensinamentos e por saber que ela estaria extremamente feliz com essa conquista.

Tiago Antero De Sousa Alves

*“Algo é só impossível
até que alguém duvide e acabe
provando o contrário”
Albert Einstein.*

RESUMO

O curso de Engenharia Naval da UFPA foi fundado no ano de 2005, através da resolução provisória do magnífico reitor da UFPA, sendo que em 2007 com a resolução nº 3601 foi aprovado pelo Conselho Superior (IES) criando o respectivo curso, tendo em sua grade curricular, matérias onde ensinam conceitos de dimensionamento de cascos de embarcações, plataformas oceânicas, e demais áreas de atuação de Engenharia Naval, e engenharias como um todo, utilizando-se *softwares* que requerem alto poder de processamento. Com base nisso, foi realizado neste trabalho um estudo sobre os conceitos fundamentais da computação paralela, apresentando arquiteturas e modelos computacionais de tipos de *clusters*, através de um breve estudo sobre os supercomputadores na história, vendo sua importância em trabalhos científicos e estudos acadêmicos, com isso, elaborou-se uma configuração de *cluster* de alto desempenho, para ser aplicados em *softwares* que executam diversos cálculos de resistências de materiais.

Palavras-chave: *cluster*, computação paralela, alto desempenho.

ABSTRACT

The UFPA Naval Engineering course was founded in 2005, through the provisional resolution of the magnificent rector of UFPA, and in 2007 with resolution 3601 was approved by the Superior Council (IES) creating the respective course, having in its curricula, materials where they teach concepts of dimensioning of ship hulls, ocean platforms, and other fields of Naval Engineering, and engineering as a whole, using software that requires high processing power. a study on the fundamental concepts of parallel computing, presenting architectures and computational models of clusters types, through a brief study on supercomputers in history, seeing their importance in scientific works and academic studies, with this, a configuration is elaborated of high-performance clustering, to be applied in software that performs various resistance calculations s of materials.

Keywords: *cluster*, parallel computing, high-performance.

LISTA DE ILUSTRAÇÕES

Figura 1: Imagem caputura do site do projeto Folding@home.....	21
Figura 2: Cluster montado com desktops.....	25
Figura 3: Exemplo de cluster configurado.....	26
Figura 4: Um dos cluster atuais da NASA.....	28
Figura 5: Estrutura básica de um cluster.....	29
Figura 6: Cluster simétrico.....	30
Figura 7: Cluster assimétrico.....	31
Figura 8: Cluster estendido.....	31
Figura 09: Arquitetura do cluster Beowulf.....	32
Figura 10: Exemplo de cluster Beowulf de estações de trabalho.....	33
Figura 11: Estrutura típica de um Windows Compute Cluster Server.....	34
Figura 12: Primeiro cluster do Google.....	40
Figura 13: Cluster avançado construído com equipamentos específicos.....	43
Figura 14: Cluster de Alta Disponibilidade.....	45
Figura 15: Cluster de Balanceamento de Carga.....	47
Figura 16: Cluster de Alto Desempenho.....	48
Figura 17: Layout da rede de computadores, Headnode e Switch.....	50
Figura 18: Planta Baixa do Pavimento Superior do Laboratório de Engenharia Naval, apresentando a localização física do servidor de domínio, headnode, switch e nós computacionais.....	51
Figura 19: Etapa inicial de processo de renderização.....	53
Figura 20: Adicionando filtros e efeitos de iluminação à imagem.....	53
Figura 21: Imagem finalizada.....	54
Figura 22: Tela de resultados apresentado por um dos módulos do programa.....	54
Figura 23: SHIPFLOW, é compatível com cluster de Alto Desempenho.....	55
Figura 24: Análise estrutural de uma partição pequena do casco de navio.....	56
Figura 25: Análise estrutural de uma segunda partição maior do casco de navio.....	57

Figura 26: Análise estrutural das duas partições anteriores em simultâneo.....	56
Figura 27: Instalação e configuração dos computadores, aqui dispostos em 4 fileiras com 3 equipamentos em cada. (setembro/2013).....	58
Figura 28: Outro ângulo apresentando as demais fileiras e o rack com o switch ao fundo.(setembro/2013).....	59
Figura 29: Tela do software Lizard Linpak.....	61
Figura 30: Resultado em GFLOPS após todos os cálculos.....	62
Figura 31: Gráfico comparativo entre o desempenho em GFLOPS X Nós do cluster, e a estimativa de valores ideais.....	62
Figura 32: Função Get Results da ferramenta Solver, é utilizada para obter os cálculos computacionais da rede do cluster.....	63
Figura 33: Tela de resultado de tempo utilizado durante o cálculo computacional.....	64
Figura 34: Resultado apresentando a deformação da estrutura.....	65
Figura 35: Relação Tempo x PCs do cluster TESTE A.....	66
Figura 36: Relação Tempo x PCs TESTE B.....	66
Figura 37: Relação Tempo x PCs TESTE AB.....	67
Figura 38: Relação TEMPO x TESTES.....	68

LISTA DE TABELAS

Tabela 1: Número de transistores a cada 10 anos.....	15
Tabela 2: Breve apresentação de modelos de paralelismo.....	23
Tabela 3: Comparação TESTE X Tempo.....	58
Tabela 4: Comparativo Tempo de execução X Nós no <i>cluster</i>	65
Tabela 5: Comparativo da percentagem de ganho de desempenho a incluir mais nós computacionais aos testes.	68

LISTA DE ABREVIATURAS E SIGLAS

CPU – *Central Processing Unit* – Unidade Central de Processamento
GB – Gigabytes
GFlops – Gigaflops
HA – *High Availability*
HPC – *High Performance Computing*
LAN – *Local Area Network*
MB – Megabytes
MFlops – Megaflops
MIMD – *Multiple Instruction Multiple Data*
MISD – *Multiple Instruction Single Data*
MPI – *Message Passage Interface*
MPP – Processadores Paralelos Massivos
NASA – *National Airspace and Space Agency*
NFS – *Network File System*
PC – *Personal computer*
RAM – *Random Access Memory* – Memória de acesso aleatório
TB – Terabytes
TCP/IP – Acrônimo de *Transmission Control Protocol/Internet Protocol*
TFlops – Teraflops

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Contexto do Trabalho	14
1.2 Aplicação De Cluster Beowulf Em Instituições De Ensino	17
1.3 Construção de um Cluster HPC Para Simulações CFD	18
1.4 Computação genérica de alto desempenho com GPUS	18
1.5 Tão poderoso quando um Supercomputador	19
1.6 Projeto	20
1.7 Motivação	21
1.8 Objetivo Geral	22
1.9 Objetivos Específicos	22
1.10 Contribuições do Trabalho	22
1.11 Organização do Trabalho	22
2 FUNDAMENTOS DE CLUSTERS	23
2.1 Cluster	24
2.2 História do Cluster	26
2.3 Estrutura básica de Clusters	28
2.3.1 <i>clusters</i> simétricos	29
2.3.2 Clusters assimétricos	30
2.3.3 <i>Cluster</i> estendido	31
2.4 Cluster Para Sistemas Operacionais Linux E Windows	32
2.5 Uso do cluster	35
2.5.1 vantagens	35
2.5.2 desvantagens	36
2.6 Desenvolvimento da Computação em Cluster	36
2.7 Tarefas onde são aplicadas a Computação em <i>Cluster</i>	39
2.8 Funcionamento Básico Dos <i>Clusters</i>	40
2.9 Principais pontos positivos do uso do <i>cluster</i>	42
2.10 Modelos de <i>Cluster</i>	43
2.10.1 cluster para alta disponibilidade - high availability	44
2.10.2 <i>Cluster</i> para balanceamento de carga - <i>load balancing</i>	45
2.10.3 <i>Cluster</i> para alto desempenho	47
3 PROPOSTA	49

3.1 Implementação do cluster de alto desempenho.....	49
3.2 Gerenciamento do <i>Cluster</i>	52
3.3 Aplicações	52
3.3.1 <i>Autodesk</i>	53
3.3.2 DNV (<i>det norske veritas</i>).....	54
3.3.3 <i>Shipflow</i>	55
3.3.4 Ansys inc.	55
3.4 Laboratório de Engenharia Naval	58
4 ANÁLISE DA PROPOSTA	60
4.1. Testes de Desempenho	60
4.2 Analise do <i>Cluster</i> de Alto Desempenho	62
5 CONCLUSÃO	69
REFERÊNCIAS.....	70
APÊNDICE.....	72
APÊNDICE A – Especificação de <i>hardware</i> do <i>cluster</i> laboratório de engenharia naval	73
APÊNDICE B – Instalação e configuração de um <i>cluster</i> na plataforma <i>windowserver hpc</i>	74

1 INTRODUÇÃO

Atualmente, pode-se observar uma necessidade maior de poder computacional para aplicações diárias. Antes, *softwares* que necessitavam pouco uso de memória RAM, hoje em dia exigem uma grande quantidade delas através de suas atualizações com novos recursos disponíveis.

1.1 Contexto do Trabalho

No início dos anos 2000, era comum computadores pessoais possuírem 128MB de memória RAM, porém, observa-se que atualmente os mesmos computadores pessoais utilizando 8GB de memória RAM, de acordo com dados da plataforma de *softwares* e jogos, *Steam*, que em setembro de 2014 possuía uma base de dados de mais de 100 milhões de usuários, apresentou em janeiro de 2015 uma pesquisa de *hardware* e *software*, onde apenas 0,01% utilizam menos de 512MB e 29,39% utilizam 8GB.

Outro exemplo desta crescente necessidade são os requisitos de utilização exigidos para o próprio Sistema Operacional.

Em outubro de 2001 era lançado o *MicrosoftWindows XP*, que exigia os seguintes requisitos de *hardware*:

- Processador Intel Pentium/Celerom ou AMD K6/Athlon/Duron de 300 MHz (mínimo de 233MHz);
- 128MB de RAM (mínimo de 64MB);
- 1,5GB de espaço disponível no disco rígido.

Em janeiro de 2007 era lançado o *Microsoft Windows Vista*, que apresentava mais de uma versão, tendo a versão *Ultimate* exigindo os seguintes requisitos de *hardware*:

- Processador 32 bits (x86) de 1GHz ou 64 bits (x64) de 1GHz;
- 1GB de RAM
- 128MB de memória gráfica, mínimo de 64MB;
- 15GB de espaço disponível no disco rígido.

Em outubro de 2009, era lançado o *MicrosoftWindows Seven*, que também apresentava mais de uma versão, tendo a versão *Ultimate* exigindo os seguintes requisitos de *hardware*:

- Processador 32 bits (x86) de 1GHz ou 64 bits (x64) de 1GHz;
- 1GB de RAM para processadores x86 e 2GB de RAM para processadores x64;
- 128MB de memória gráfica;
- 16GB (x86) e 20GB (x64) de espaço disponível no disco rígido.

A partir disso, não houve muita alteração na exigência do uso de *hardware* para utilização do sistema, apenas os recursos presentes na placa gráfica disponível que se atualizaram, o Sistema Operacional manteve-se com a mesma quantidade de *hardware* necessário, enquanto *softwares* de terceiros necessitavam de mais processamento, memória RAM e qualidade gráfica.

Mas por que houve esse aumento de requisitos de *hardware*, segundo Gordon Moore:

A complexidade para componentes com custos mínimos tem aumentado em uma taxa de aproximadamente um fator de dois por ano [...]. Certamente em um curto prazo pode-se esperar que esta taxa se mantenha, se não aumentar. A longo prazo, a taxa de aumento é um pouco mais incerta, embora não haja razões para se acreditar que ela não se manterá quase constante por pelo menos 10 anos. Isso significa que em torno de 1975, o número de componentes por circuito integrado para um custo mínimo será 65.000. Eu acredito que circuitos grandes como este poderão ser construídos em um único componente (SEVCENKO, 2004).

Constatou-se, com isso, que a miniaturização vinha permitindo dobrar o número de transistores em circuitos integrados a cada ano enquanto o custo permaneceria constante, uma tendência que deveria se manter por pelo menos mais dez anos. Em 1975, ele atualizou o artigo e constatou que o número passaria a dobrar a cada vinte e quatro meses, dando origem à Lei de Moore. A Tabela a seguir apresenta a evolução do número de transistores ao longo do tempo.

Tabela 1: Número de transistores a cada 10 anos.

8088 (1979):	29.000 transistores
486 (1989):	1.200.000 transistores
Pentium III (1999):	21.000.000 transistores
Core i7 (2009):	731.000.000 transistores

Fonte: Adaptado MOORE

A realidade é que fabricantes de chips como a Intel precisam lançar novos processadores regularmente para alimentar o ciclo dos upgrades e manter assim as vendas em alta. Cada nova geração precisa ser consideravelmente mais potente que a anterior, o que leva à busca por novos processos de fabricação.

Originalmente, a Lei de Moore não falava nada sobre o desempenho, mas apenas sobre o número de transistores em processadores, módulos de memória e outros circuitos. Entretanto, a sofisticação dos circuitos acompanha o desempenho, já que mais transistores significam mais unidades de processamento, mais *core*, mais memória cache e assim por diante. Além disso, novas técnicas de fabricação permitem também aumentar o *clock*, o que resulta em mais instruções por ciclo e mais ciclos por segundo.

Sendo assim, quando se necessita de um grande poder de processamento acima do estipulado pelo prazo da Lei de Moore, recorre-se a computação de alto desempenho. É uma necessidade em certos ramos da pesquisa científica, governo e empresas, através de cálculos de problemas matemáticos complexos, previsão de clima, simulações geotérmicas, simulações financeiras, renderização de efeitos especiais em filmes e imagens 3D de *softwares* específicos, et, devido a essa alta complexidade de processamento, surge a necessidade de supercomputadores.

Eles já existem desde 1970. São os mais poderosos, mais rápidos e de maior custo, e exatamente pelo alto custo, se tornam inviáveis para o uso comum. Entretanto, havia uma ideia da IBM existente desde a década de 60, como uma maneira de interligar os *mainframes* com o objetivo de obter uma solução comercialmente viável de paralelismo.

Existem vários tipos de cluster implementados e em funcionamento, abaixo vou descrever três trabalhos desenvolvidos em universidades do país. Estas Instituições de Ensino, como ambiente propício para pesquisas e estudos precisam acompanhar o desenvolvimento oferecendo aos estudantes e pesquisadores oportunidades para realizarem projetos e soluções que necessitem de cálculos complexos, que somente são alcançados com recursos computacionais com alto poder de processamento, e um projeto que apesar de não ser um cluster 'tradicional', e uma forma de computação distribuída.

1.2 Aplicação de Cluster Beowulf em instituições de ensino

Monografia apresentada por Claudio Prado e João da Silva à Faculdade de Tecnologia de Guaratinguetá, para graduação no Curso Superior de Tecnologia em Informática, ênfase Redes de Computadores, apresenta um estudo sobre as necessidades de instituições de ensino para o uso de Cluster de Alto Desempenho, através de um questionário enviado as instituições, questionando a utilização de um cluster oferecendo alto desempenho e baixo custo, como centro de processamento.

Garantindo assim a instituição de ensino uma ferramenta viável sem a necessidade de grandes investimentos, correlacionando a necessidade de um laboratório para aplicações de alto desempenho, e a utilização dos alunos para aplicação prática de conhecimento adquirido, não sendo necessariamente de áreas comuns à informática, podendo outras faculdades utilizarem o espaço para o desenvolvimento que visa o pleno desenvolvimento da pessoa, por exemplo, os alunos da faculdade de biologia, podem utilizar o cluster para sequenciamento de genomas, e demais coisas inerentes a cada curso (PADRO; SILVA, 2010).

Todas as pesquisas e estudos em que envolvem cálculos complexos, simulações e modelagem computacional se beneficiam do processamento de alto desempenho, como as grandes áreas da educação e as suas subáreas, engenharia, matemática, física, química, meteorologia (ou ciências atmosféricas), ciências da computação, bioquímica, biologia, genética, medicina, nanotecnologia, geociências e etc.

Um cluster de alto desempenho passa a exercer uma grande influência no meio acadêmico quando aplicado e incentivado a utilização desse tipo de estrutura, pois aos alunos passam a conhecer uma nova vertente de processos computacionais, assim, podem investir mais em alta necessidade de computação, por exemplo, ao utilizar uma aplicação que levaria muito tempo para ser finalizada em apenas um computador, o aluno reduz o trabalho para evitar perda de tempo, ou falhas durante o processo, um grande poder computacional, expande o horizonte, utilizando muito mais e em menor tempo do que normalmente utilizaria (PADRO; SILVA, 2010).

1.3 Construção de um Cluster HPC Para Simulações CFD

Monografia apresentada Ernani Kopp para obtenção do grau de especialista em Teleinformática e Redes de Computadores, do Departamento Acadêmico de Eletrônica da Universidade Tecnológica Federal do Paraná.

Abrangendo o nosso tema de estudo, pois utilizam do mesmo software em modo paralelo para elaboração de um trabalho envolvendo fluídos hidrodinâmicos, como as engenharias demandam um grande poder computacional, e o custo das máquinas utilizadas nessas áreas são sempre elevadas, investiu-se em um laboratório de alto desempenho, sendo a opção mais viável economicamente, necessitando apenas de treinamento para sua utilização, e pessoal qualificado para implementar e orientar os alunos, esse investimento pode ser aplicado tanto por instituições de ensino, interessadas em pesquisa científica e tecnológica como de indústrias de grande porte.

No trabalho, é citado o Laboratório de Ciências Térmicas – LACIT, localizado na UTFPR – campus Curitiba, é um grupo de pesquisa criado em 1999 com atividades na área de Mecânica dos Fluidos, Transferência de Calor e Termodinâmica. O grupo trabalha em diversas áreas de Engenharia Térmica e seus estudos envolvem não só investigação básica, mas também aplicada (KOPP, 2012).

NO LACIT são feitas diversas simulações de escoamentos monofásicos, bifásicos, comportamento de bombas entre outras atividades referentes à área de mecânica dos fluidos. As simulações são realizadas através do software CFX® um programa comercial de CFD (do inglês *Computational Fluid Dynamics* que integram o *software* Ansys Inc.

1.4 Computação genérica de alto desempenho com GPUS

Normalmente ao se falar de cluster, definia-se apenas o processador sendo utilizado para computação paralela, porém, nos últimos anos houve uma grande diferença no poder computacional entre CPU e GPU, sendo estas últimas bastante superior ao poder computacional alcançado pelo processador, uma única GPU de alto nível, como a NVIDIA GTX 970, pode chegar a incríveis 4.466 GFLOPS (para comparação, nosso cluster presente nesse estudo chegou a 282 GFLOPS com 8 computadores em cluster), porém, a diferença na arquitetura e utilização entre esses dois modelos explicam essa grande diferença entre eles.

As CPU's foram desenhadas com objetivo principal de otimizar um único fio de execução sequencial, sendo essa a sua especialidade. Tecnologias como o *multi-threading* e a inclusão de múltiplos núcleos de execução num único circuito, a adição de mais núcleos, não alteram significativamente este panorama, sendo que todos os núcleos são semelhantes.

As GPU foram desenhadas para resolver rapidamente problemas relacionados com a computação gráfica que são paralelizáveis, mesmo no caso de algoritmos serem sequenciais, o domínio dos dados é paralelo, como exemplo os cálculos em três dimensões paralelamente para apresentar corretamente os resultados em tempo real na tela.

O fato de GPU's se concentrarem nas tarefas paralelas destinadas essencialmente ao processamento gráfico, permite descartar as responsabilidades de qualquer processador em termos de gestão de *hardware* e de *software* de modo geral, isso explica a diferença entre performance entre essas unidades, e a necessidade de utilização desse poder extra.

Os estudos sobre a utilização de GPU'S em computação paralela, ainda são relativamente novos no campo de vista computacional, já há diversos estudos para se maximizar o desempenho alcançado nesse tipo de cluster, e como GPU's de alto nível tem um custo elevado, acaba perdendo o atrativo econômico ao se utilizar de computadores já existentes em laboratórios, sendo esse um investimento que deverá ser analisado.

1.5 Tão poderoso quando um Supercomputador

Um grupo de pesquisadores do departamento de Engenharia Elétrica e de Computação da Universidade Federal do Pará, construiu um cluster, interligando em rede doze computadores no Laboratório de Análise Numérica em Eletromagnetismo – LANE, no Centro Tecnológico da UFPA.

O projeto foi desenvolvido em parceria com empresas de telefonia celular, para estudo da propagação de ondas eletromagnéticas em ambientes abertos e fechados, em especial com florestas, típicas da Amazônia. “Nesse projeto busca-se contribuir ao planejamento de redes de comunicação sem fio em geral, e em especial de sistemas celulares, interesse dos financiadores. Esse estudo envolve problemas complexos, com soluções que demandam por sistemas computacionais

com grande capacidade de processamento e que só foi possível realizá-lo com o cluster montado”, conta Leônidas Souza, professor que desenvolveu o cluster no LANE, após utilizar um sistema de cluster no seu pós-doutorado em Londres, e sentindo a necessidade de continuar os estudos quando regressou.

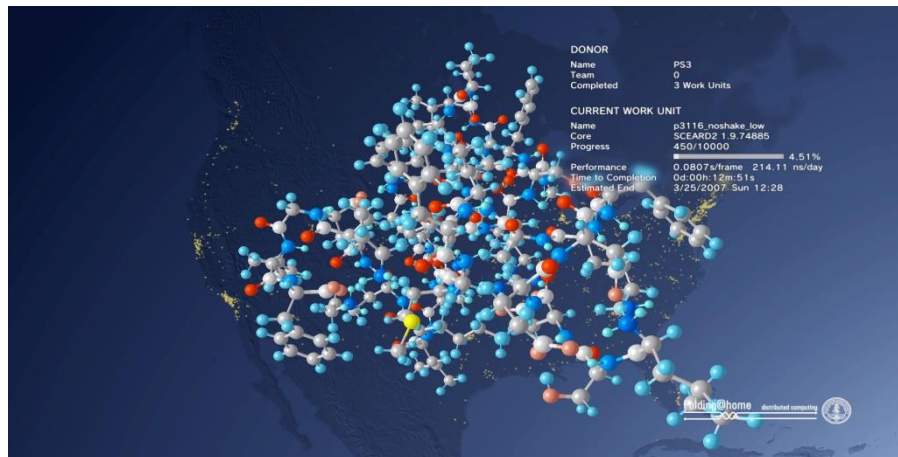
Os pesquisadores desenvolveram em 2015, um projeto em parceria com a Eletronorte, de análise de sistema de aterramento para subestação de energia elétrica. E estavam negociando com a Albrás um projeto de otimização do processo de obtenção do alumínio. Existe um sério problema correlacionado à alta intensidade do campo magnético do forno da empresa. “Como se sabe, toda corrente elétrica gera um campo magnético. Na fábrica da Albrás, trabalha-se com corrente elétrica de 170 mil ampères. Esta corrente alta gera um enorme campo magnético que, por sua vez, interfere na obtenção do alumínio, criando uma espécie de resistência no processo”, relata o pesquisador Leônidas Souza.

A proposta dos pesquisadores da UFPA é a elaboração de um projeto de modificação do layout das cubas. “Este é um problema de grande importância porque a melhora de 1% em eficiência, por exemplo, pode resultar em lucro de milhões de dólares à empresa durante o período de um ano. O sistema da Albrás é muito complexo e envolve muitos detalhes. Sem o nosso cluster, não teríamos condições computacionais para fazer a análise”, afirma Leônidas Souza.

1.6 Projeto

Em 2000, foi lançado o *software Folding@home*, que é um projeto de computação distribuída, foi desenvolvido pela Sony em parceria com um grupo de cientistas da Universidade Stanford (Califórnia, EUA) que visa ajudar a entender o desenvolvimento e a encontrar a cura para várias das doenças mais comuns e também para o mal de Alzheimer, doenças infecciosas como o Ebola e vários tipos de câncer. Eles estão estudando o enovelamento de proteínas e precisam realizar milhões de cálculos complicados para que suas pesquisas avancem.

Figura 1: Imagem captura do site do projeto Folding@home



Fonte: <https://foldingathome.org/>

O objetivo do projeto é resolver problemas computacionais maciços relacionados com a investigação de enovelamento de proteínas, esses problemas exigem tantos cálculos, que desenvolveram o *software* para ser instalado em computadores pessoais ao redor do mundo, para doar um pouco do poder de processamento, disponível para *Windows*, *Linux* e OS X, configurando-se automaticamente, necessitando informar qual horário e qual porcentagem poderá doar para o projeto, contam com uma página com as informações de pesquisas que estão sendo desenvolvidas, seu doadores, e estão atualmente com uma campanha para alcançar a marca de um milhão de processadores, no momento o projeto está com cerca de 110 mil colaboradores.

1.7 Motivação

Após participar de diversos projetos de Engenharia Naval no decorrer do curso de Sistemas de Informações, constatou a grande necessidade de poder de processamento que diversos segmentos da área de Engenharia Naval, e demais engenharias, utilizavam. Sendo assim, foi desenvolvido o estudo e aplicação de um *cluster* de alto desempenho para ser utilizados pelos alunos e professores do curso de Engenharia Naval, podendo essa implementação posteriormente ser aplicada e aperfeiçoada para as necessidades dos demais cursos.

1.8 Objetivo Geral

Implementação de um *cluster* de altodesempenho num ambiente educacional para gerar um grande poder de processamento para aplicações da Faculdade de Engenharia Naval (FENAV) da Universidade Federal do Pará (UFPA).

1.9 Objetivos Específicos

De forma a atingir o objetivo geral deste trabalho, foram traçados os seguintes objetivos específicos:

- i. Implementação de um cluster de alto desempenho;
- ii. Instalação e configuração do cluster de alto desempenho;
- iii. Realizar testes que comprovem o ganho de desempenho do cluster
- iv. Analisar os resultados
- v.

1.10 Contribuições do Trabalho

A principal contribuição deste trabalho está na implementação de um *cluster* de alto desempenho que vai diminuir o tempo necessário para realização de cálculos computacionais utilizados pelos usuários.

1.11 Organização do Trabalho

Além da Introdução, este trabalho está organizado nos seguintes capítulos:

- O capítulo 2 aborda uma visão geral sobre os fundamentos de cluster, apresentando conceitos e diversos tipos e aplicabilidades de cluster, bem como a necessidade de combiná-los, caso seja necessário.
- O capítulo 3 trata dos trabalhos relacionados, apresentando outros estudos e projetos realizados envolvendo cluster, tanto em meio acadêmico quanto em indústrias.
- O capítulo 4 apresenta a proposta de implementação de um cluster de alto desempenho, bem como os softwares utilizados na área de engenharia naval, e suas opções quanto ao processamento paralelo.
- O capítulo 5 fornece uma análise dos resultados obtidos na simulação de testes, utilizando grande poder computacional, demonstrando a diminuição do tempo de execução dos testes no cluster.

- O capítulo 6 apresenta a conclusão do trabalho, destacando suas principais contribuições e sugestões para trabalhos futuros.

2 FUNDAMENTOS DE CLUSTERS

Os avanços tecnológicos existem para facilitar nossa vida a cada dia. Dentre todas as inovações na área de Tecnologia da Informação (TI) que surgiram no decorrer das últimas décadas, a computação é uma das revoluções mais populares, promissoras e bem-sucedidas, e sua evolução tem sido cada vez mais rápida. É uma aliada das pessoas, encurtando as distâncias e principalmente no que diz respeito a rapidez em que simplifica processos, na execução de tarefas que levariam horas para serem executadas (TANNEBAUM, 2006).

O sucesso e o poder das inovações no âmbito de Tecnologia da Informação, são tão grandes, que por conta de sua praticidade em processar várias tarefas simultaneamente, tarefas de alta complexidade que exigem o máximo dos processadores para a sua execução, as empresas, indústrias e instituições que compõe pesquisas, também apostaram nessa inovação, especificamente em Computação Paralela ou Distribuída, fracamente acoplada, utilizando as técnicas de *clustering*, bem como suas tendências e suas principais aplicações atualmente.

Tabela 2: Breve apresentação de modelos de paralelismo.

MODELO	DESCRIÇÃO
Processamento de múltiplas CPU's	Fortemente acopladas: Sistemas <i>Multicore</i> , placas-mães com mais de um soquete de processador.
Paralelismo por CHIP	<i>Multithreading</i> : Capacidade que o sistema operacional possui de executar várias tarefas (<i>threads</i>) sem que uma interfira na outra, sendo executadas de forma independente.
	<i>Multicore</i> : Um processador que possui mais de um núcleo, que podem executar várias instruções ao mesmo tempo.
Pipeline (Instruções)	Técnica de <i>hardware</i> que permite que a CPU realize a busca de uma ou mais instruções além da próxima a ser executada, sendo criado uma fila na memória dentro do processador, que é muito mais rápida que a memória RAM do sistema.

Fonte: Adaptado TANNEBAUM, (2006)

A tecnologia de *clustering* surgiu há pouco tempo como uma resposta ao aumento súbito da necessidade de poder computacional em áreas como: ciências, engenharia, astronomia, medicina e outras, onde a realização de testes, experiências e estudos muitas vezes requerem poder de processamento muito além do que poderia ser obtido com um único equipamento, uma vez que, apesar do aumento das velocidades de processamento dos computadores atuais, os mesmos ainda são, individualmente, insuficientes para os anseios dessas áreas.

Com o crescimento das necessidades de trabalho, surgiram os chamados *clusters*, que é uma estrutura de computação que pode proporcionar melhor desempenho, confiabilidade e agilidade para a execução de processos de alta complexidade (TANNEBAUM, 2006).

2.1 Cluster

Quando o assunto é computação de alto desempenho, não é difícil pensar em servidores sofisticados e caros respondendo por este trabalho. No entanto, é possível obter resultados tão bons quanto ou superiores a partir de alguma solução de *cluster* - uma tecnologia capaz de fazer computadores mais simples trabalharem em conjunto, como se formassem uma máquina só.

Há muito tempo vem se enfrentado muitos problemas relacionados ao custo da aquisição de supercomputadores. Estes ainda continuam com o preço muito elevado, apesar da evolução tecnológica. Os computadores pessoais, por outro lado, vêm-se barateando a cada lançamento de novas tecnologias, tornando cada vez mais viável a criação de sistemas paralelos através da criação de sistemas distribuídos fracamente acoplados (definidos como "sistemas distribuídos onde os processadores não compartilham os mesmos barramentos") através da criação de *clusters*. (TANNEBAUM, 2006, p.32).

Cluster é um termo em inglês que significa "aglomerar" ou "aglomeração" e pode ser aplicado em vários contextos. No caso da computação, ele define uma arquitetura de sistema capaz de combinar vários computadores para trabalharem em conjunto ou, então, pode denominar o grupo em si de computadores combinados. Ou seja, um *cluster* consiste em computadores vagamente ou fortemente ligados que trabalham em conjunto para que, em muitos aspectos, eles possam ser vistos como um único sistema e ao contrário da computação em grade, um *cluster* tem,

para executar a mesma tarefa, cada conjunto de “nós” é controlado e programado por *software*. (TANNEBAUM,2006)

Figura 2: *Cluster* montado com *desktops*



Fonte: Laboratório Chemnitzer *Linux Cluster* (CLIC)

É importante ressaltar que é, mas, recomendável que todos os computadores sejam iguais, ou seja, possuam a mesma configuração, mas é ideal que eles tenham o mesmo sistema operacional, para facilitar a configuração do *software*, já que não será necessário fazer correções na comunicação entre um sistema e outro, garantindo assim que não haja perdas no desempenho.

Os computadores que compõem os nós do *cluster* podem ser simples máquinas domésticas, que juntas podem oferecer desempenho suficiente para agilizar o tempo para se obter resultados em determinadas áreas, aplicações que levariam mais de um dia para serem executados num único computador mediano, podem ser executados em poucas horas em computadores trabalhando juntos.

Figura 3: Exemplo de *cluster* configurado



Fonte: MORAIS, Dhiogo (2012 – VII CONNEPI)

O objetivo da criação de um *cluster* é sempre aumentar a eficiência da fusão, ou seja, otimizar o uso pleno dos recursos de todas as estações e evoluir na dinamicidade do circuito. Por tanto, há tipos de supercomputadores que são focados em diferentes benefícios para a fusão e, conseqüentemente, são mais adequados a determinadas tarefas e mercados (MACHADO; MAIA, 2007).

2.2 História do Cluster

Os Sistemas Paralelos ou Distribuídos são temas relativamente atuais vem sendo usados inicialmente por grandes empresas do setor de Tecnologia, como já vimos anteriormente.

O uso desses sistemas iniciou-se com a IBM na década de 60, interligando grandes *mainframes*. O objetivo era obter uma solução comercialmente viável de paralelismo. Na mesma época já havia um sistema que fazia a distribuição das

tarefas nos *mainframes* interligados: o HASP (*Houston Automated Spooling Program*), que em seguida teve como o seu sucessor JES (*Job Entry System*). Apesar de essa tecnologia ter sido implantada há décadas, até hoje a IBM a utiliza, só que agora com PSS (*Parallel Sysplex System*).

Esse sistema consiste em manter os velhos *mainframes* úteis, proporcionando melhor desempenho e redução de custo, já que irão ficar em funcionamento por muito mais tempo (TANEMBAUM, 2006, p.31).

Na década de 1980, três tendências se convergiram: microcomputadores de alto desempenho, redes de alta velocidade, e ferramentas padronizadas para computação distribuída de alto desempenho. Essas convergências foram fundamentais para o aprofundamento nas técnicas de sistemas paralelos.

Já que, sempre houve a necessidade de alto poder de processamento, seja para aplicações científicas ou para outras aplicações que exijam tal poder de processamento. A implementação de tais aplicações, só era possível com a aquisição dos supercomputadores. Estes ainda continuam com preços elevados. Com a tecnologia de cluster obtém-se alto poder de processamento com baixo custo (TANEMBAUM, 2006, p.32).

Uma quarta tendência possível é a crescente necessidade de poder de processamento para aplicações científicas e comerciais unidas ao alto custo e a baixa acessibilidade dos tradicionais supercomputadores.

No final de 1993, dois cientistas da NASA - Donald Becker e Thomas Sterling - começaram a planejar um sistema de processamento distribuído com computadores Intel *Desktop*, com o objetivo de fazer com que esses computadores conectados em rede se assemelhassem ao processamento de um supercomputador, e assim tivessem um menor custo (JAQUIE, 2010).

No início de 1994, já trabalhando no CESDIS – *Center of Excellence in Space Data and Information Sciences*, Centro de Excelência em Dados Espaciais e Ciência da Informação-, criaram o projeto *Beowulf*. Com 16 computadores Intel 486 DX4 conectados em uma rede *Ethernet* com sistema operacional *Linux* fizeram esse primeiro *cluster* chegar a um desempenho de 60 Mflops. A partir daí o projeto *Beowulf* tornou-se de grandes interesses pelos meios acadêmicos, pela NASA entre outros (TANEMBAUM, 2006).

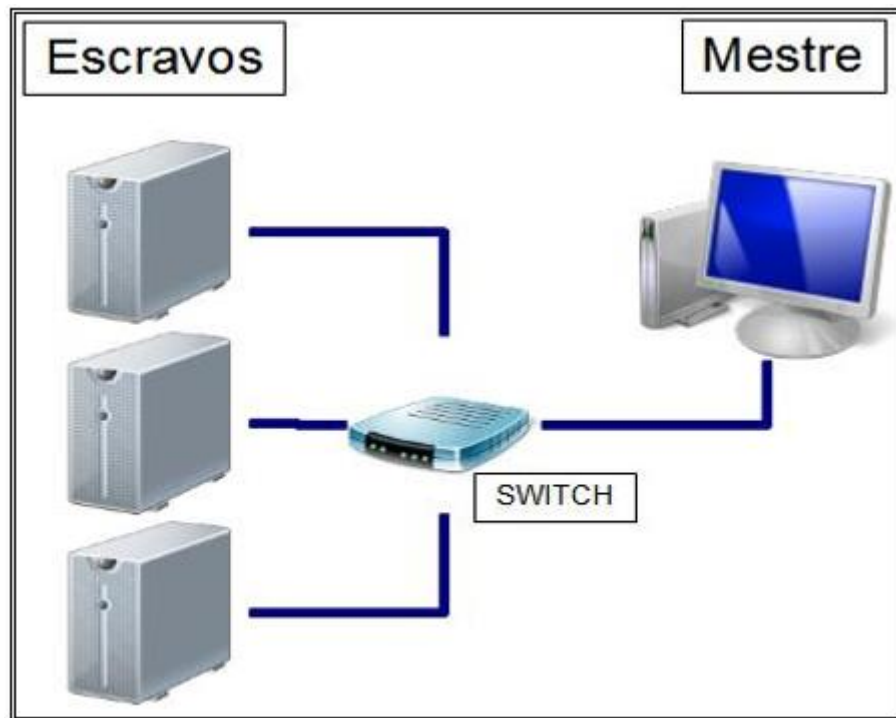
Figura 4: Um dos *cluster* atuais da NASA



Fonte: Supercomputador Plêiades

2.3 Estrutura básica de Clusters

É necessário pensar um pouco sobre a estrutura interna do *cluster* e sua topologia: isso implicará decidir quais papéis as máquinas individuais irão exercer e qual o tipo de interconexão é a mais adequada, a seguir temos uma figura exemplificando a estrutura básica de um cluster.

Figura 5: Estrutura básica de um *cluster*

Fonte: Adaptado SLOAN (2004).

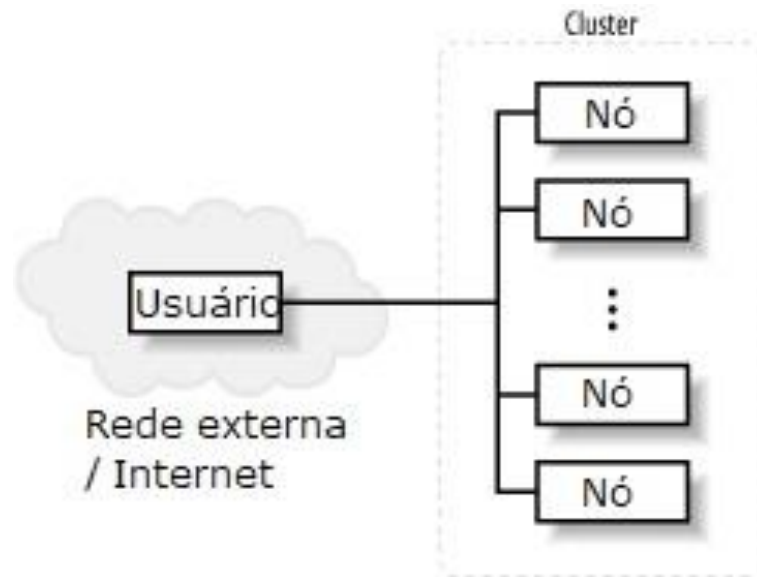
Um conceito básico são os elementos que compõem um *cluster*, que são seguintes:

- Nó mestre computador responsável pela administração dos nós computacionais exerce também a função de autenticação e segurança do *cluster*;
- Nó escravo: computador dedicado a processar os dados do *cluster*.

Genericamente, cada nó pode ser uma máquina uni processador, no entanto, usualmente, os nós são PCs, A figura 5 apresenta a arquitetura de um *cluster*.

2.3.1 *clusters* simétricos

A abordagem mais simples é um conjunto simétrico, cada nó pode funcionar como um computador individual. Cria-se uma sub rede com diversas máquinas e pode-se adicionar nós que são gerenciados por *softwares* de *cluster* específico. O problema desse tipo de configuração é a falta de segurança e gerenciamento, visto que não existe um controle centralizado. Um exemplo de *cluster* simétrico são os *clusters* que usam de estações de trabalho.

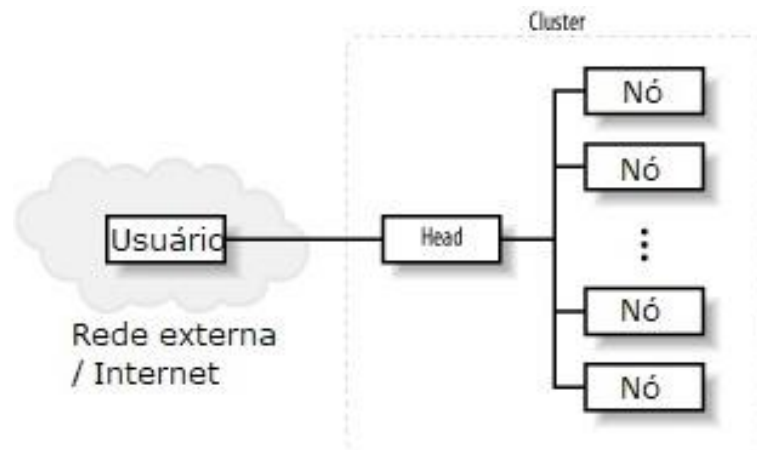
Figura 6: *Cluster* simétrico

Fonte: Adaptado SLOAN (2004).

2.3.2 Clusters assimétricos

Para o *cluster* desempenhar tarefas mais específicas e complexas, uma arquitetura assimétrica é mais adequada. Com *clusters* assimétricos um computador é o nó principal ou *headnode*. Ele serve como um gateway entre os nós restantes e os usuários. Os nós computacionais geralmente utilizam configuração mínima de *hardware*. Uma vez que todo o tráfego deve passar através do *head node*, os *clusters* assimétricos tendem a fornecer um nível de segurança mais elevado (BACELLAR, 2010).

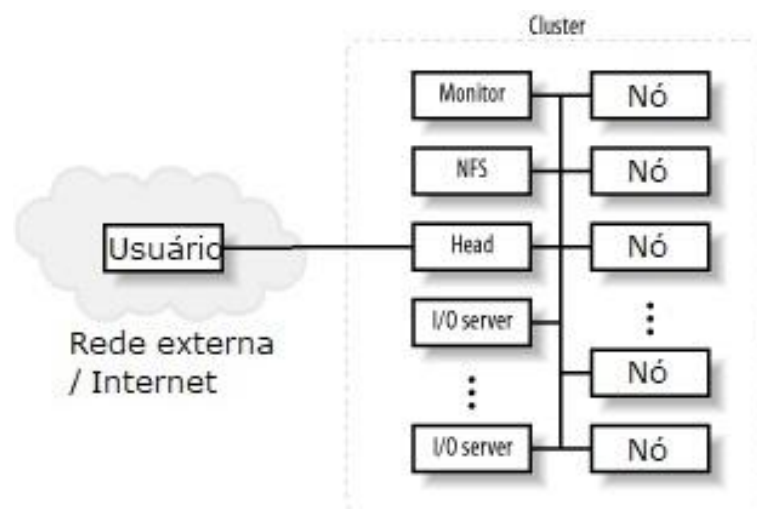
O *head node* muitas vezes age como um servidor primário o que possibilita gerenciamento centralizado dos nós computacionais possibilitando serviços como instalação remota e autenticação de usuários (BACELLAR, 2010).

Figura 7: *Cluster* assimétrico

Fonte: Adaptado SLOAN, 2004

2.3.3 *Cluster* estendido

Para um melhor intercâmbio entre o *cluster* e as fontes de dados é necessário “expandir” a topologia do *cluster*, a fim de melhorar o acesso deste pelos usuários. Para isso usa-se redes heterogêneas: cenário ao qual o *cluster* é dinamicamente ligado a outro tipo de rede, por exemplo, uma LAN, de um laboratório de pesquisa que pode continuamente estabelecer contato com o ele para processamento de dados (BARCELLOS; GASPARY, 2006).

Figura 8: *Cluster* estendido

Fonte: Adaptado SLOAN, 2004.

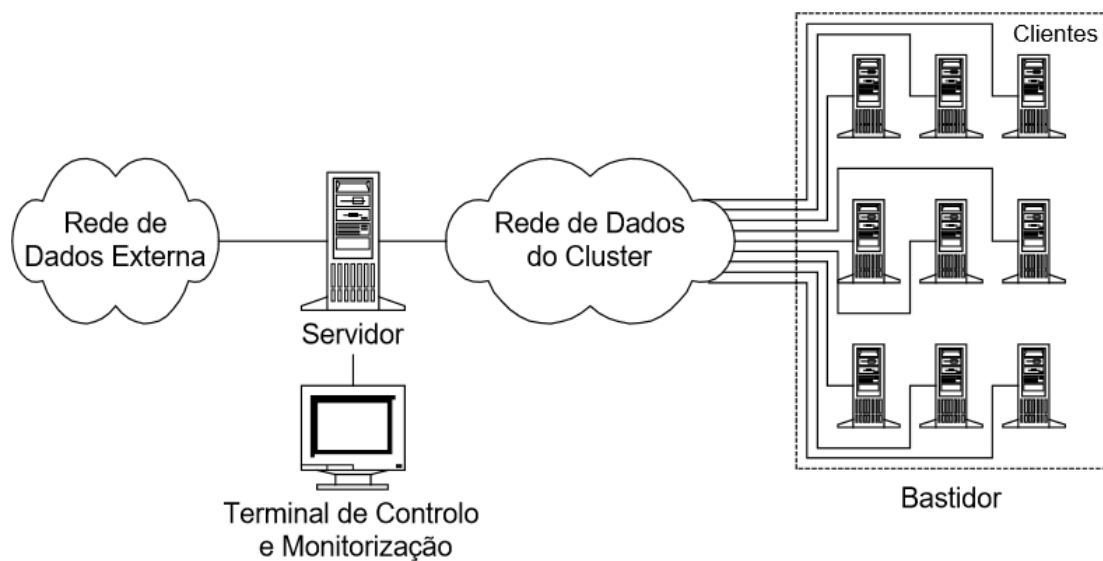
O usuário pode se conectar ao *cluster* através de uma estação de trabalho e diretamente da mesma, gerenciar tarefas rotineiras tais como verificar o status de uma simulação, ou processar rapidamente uma quantidade de dados que demoraria mais tempo em uma estação de trabalho.

Outra questão crítica é o projeto de rede. Em pequenos *clusters*, alguns *switches* podem ser suficientes; em *clusters* maiores, uma rede totalmente conectada pode ser proibitivamente cara. Em situações mais complexas de processamento, o intercâmbio de dados entre os nós pode ser prejudicado, pois a troca de mensagens e sincronização entre eles exige interfaces de comunicações mais sofisticadas; de baixa latência de comutação, caso contrário o desempenho geral cai e *cluster* perde muito de sua viabilidade (COLOMBET; DESBAT, 2003).

2.4 Cluster Para Sistemas Operacionais Linux E Windows

O projeto criado por Donald Becker, é conhecido como *cluster Beowulf* 'cluster de alto desempenho' para o sistema *Linux*, e hoje são usados em todo mundo, principalmente para processamento de dados com finalidade científica, uma área em que são muito utilizados é na renderização de filmes. As principais vantagens do *cluster Beowulf*:

Figura 09: Arquitetura do *cluster Beowulf*



Fonte: Adaptado de STERLING (2002)

- Sistema escalável, sendo possível pôr em rede e coordenar um grande número de nós, não existindo um limite definido para o tamanho do *cluster*.
- Flexibilidade em relação ao *hardware*: Os equipamentos utilizados são facilmente comercializados, não necessitando de um equipamento específico para a criação do *cluster*.
- No caso de um nó defeituoso, a substituição é tão simples quanto mudar um PC. Desta forma, é possível gerenciar as falhas de maneira eficiente, baseando-se na fácil substituição de equipamentos.
- Baixo custo

Figura 10: Exemplo de *cluster Beowulf* de estações de trabalho



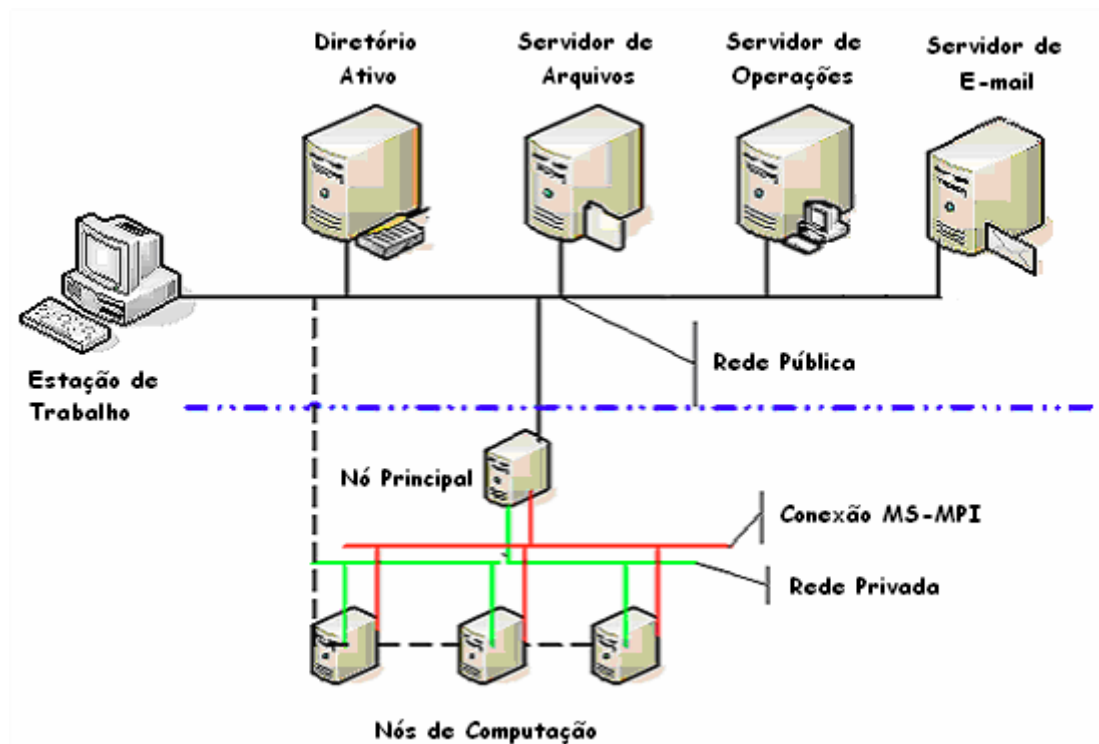
Fonte: *Cluster Beowulf (Ecgtheow)* do MTU (*Michigan Tech University – Universidade Tecnológica de Michigan*)

A Figura 11 apresenta a estrutura desse tipo de cluster, o *Windows Compute Cluster Server* oferece um ambiente mais amigável sendo baseado principalmente em interfaces gráficas, o que facilita a sua utilização. Porém, por ser um sistema fechado, não oferece tantas alternativas de ferramentas quanto o *GNU/Linux*.

Existem outras opções no mercado que são os *clusters* comerciais que geralmente usam computadores e *softwares* proprietários. Por ser um pacote “fechado” entre *software* e *hardware* a estabilidade e desempenho do sistema são bem combinados; aliado a esse benefício, existe ainda opção de suporte e garantias bem abrangentes disponíveis ao cliente, entretanto o custo é muito mais alto se comparado a *clusters* que utilizam computadores pessoais.

Outro recurso interessante é o uso de estações de trabalho como “auxiliares” de *clusters*, por exemplo, um *cluster* pode se conectar a uma rede LAN com várias estações de trabalho ociosas e usar os recursos computacionais disponíveis de cada estação para processar dados. Esse tipo de estrutura recebe o nome de nós não dedicados, como vemos a seguir.

Figura 11: Estrutura típica de um *Windows Compute Cluster Server*



Fonte: RUSSEL (2005)

Devido à amplitude e importância de diversas aplicações abordadas pela computação de alto desempenho e à necessidade da evolução das tecnologias de programas voltadas a esta arquitetura, a *Microsoft* lançou o *Windows Compute Cluster Server*, que é um sistema operacional voltado à implantação, configuração e gerenciamento de *clusters*. A *Microsoft* tem compartilhado seu interesse com outras

indústrias de *software* e comunidade acadêmica para construção de tecnologias de suporte a aplicações de alto desempenho sobre arquiteturas paralelas (RUSSEL, 2005).

Em um mercado tradicionalmente dominado por várias versões do sistema operacional *Linux*, o *Windows Compute Cluster Server* tem sido proposto pela *Microsoft* como uma alternativa baseada na versão servidor do sistema operacional *Windows* para simplificar a implantação e gerenciamento de *cluster*.

Os *clusters* do *Windows* fornecem três tecnologias de *cluster* diferentes, porém, complementares. As tecnologias de *cluster*, que acompanham produtos distintos, podem ser utilizadas separadamente ou de forma combinada para fornecer serviços escalonáveis e de alta disponibilidade.

O *Windows Compute Cluster Server* gerencia um grupo de computadores que inclui um único “nó” principal e um ou mais “nós” de computação, ou *compute nodes*.

Com a finalidade de explorar o desempenho do *cluster* com tecnologia *Windows*, foi desenvolvido o MS-MPI, que é uma implantação do MPI adaptada às características do *Windows Compute Cluster Server* com a colaboração do Laboratório Nacional de Argonne (ANL) nos EUA (RUSSEL, 2005).

2.5 Uso do cluster

Foi apresentando anteriormente que a computação em *cluster* se mostra muitas vezes como uma solução viável porque os nós podem até mesmo ser compostos por computadores simples, como PCs de desempenho mediano. Juntos, eles configuram um sistema de processamento com capacidade suficiente para dar conta de determinadas aplicações que, se fossem atendidas por supercomputadores ou servidores sofisticados, exigiriam investimentos muito maiores.

2.5.1 vantagens

Podemos citar quando abordamos o uso de *cluster* são:

- Pode-se obter resultados tão bons quanto ou até superiores que um servidor sofisticado a partir de máquinas mais simples e mais baratas (ótima relação custo-benefício);
- Não é necessário depender de um único fornecedor ou prestador de serviço para reposição de componentes;

- A configuração de um *cluster* não costuma ser trivial, mas fazer um supercomputador funcionar poder ser muito mais trabalhoso e exigir pessoal especializado;
- É possível aumentar a capacidade de um *cluster* com a adição de nós ou remover máquinas para reparos sem interromper a aplicação;
- Há opções de *softwares* para *cluster* disponíveis livremente, o que facilita o uso de uma solução do tipo em universidades, por exemplo;
- Relativa facilidade de customização para o perfeito atendimento da aplicação;
- Um *cluster* pode ser implementado tanto para uma aplicação sofisticada quanto para um sistema doméstico criado para fins de estudos, por exemplo.

2.5.2 Desvantagens

Mas, mesmo com todos esses benefícios, os *clusters* não são a solução perfeita para todo e qualquer problema computacional, devido.

- A facilidade de expansão do *cluster* pode ser tornar um problema, a quantidade de máquinas pode aumentar tanto que a manutenção se torna mais trabalhosa, o espaço físico pode ficar impróprio, etc.;
- A tecnologia de comunicação utilizada pode não oferecer a velocidade de transferência de dados ou o tempo de resposta necessário, dependendo da aplicação;

Por estes aspectos, fica evidente que as necessidades e os requisitos de uma aplicação devem ser bem avaliados para que se possa decidir entre a implementação de um *cluster* ou outra tecnologia. Se o *clustering* for a opção escolhida, deve-se seguir com a avaliação, desta vez para se decidir sobre as soluções e recursos disponíveis.

2.6 Desenvolvimento da Computação em Cluster

A Computação em *cluster* é basicamente uma situação onde mais de um computador está conectado por uma placa de rede trabalhando. A principal vantagem desta técnica reside na possibilidade de reutilização de equipamentos legados de baixa capacidade de processamento computacional para implementação de pequenos supercomputadores a fim de obter um aumento considerável da

capacidade computacional com custo relativamente baixo. Outros problemas, como a situação do "lixo tecnológico", também pode ser beneficiada pela criação de *clusters*, uma vez que serão menos computadores nos lixões.

A cada dia novos computadores são lançados no mercado, devido ao rápido avanço tecnológico e evolução de aplicações de *software* que exigem cada vez mais capacidade dos computadores. Não só os cientistas precisam de computadores mais velozes, mas também várias empresas que trabalham com processamento crítico e necessitam de tais capacidades dos *hardwares*. Seguindo essa lógica, dificilmente um computador irá atender a todas as necessidades de processamento, uma vez que, mesmo que a velocidade continue a subir, ela não pode aumentar indefinidamente seu ciclo, devido ao efeito joule (transformação de energia elétrica em energia térmica) que resultará em problemas como: travamento, diminuição do desempenho, aumento no custo, etc. (PINHO; MAZZA; ROQUE, 2006).

Para poder aumentar o poder de processamento, uma saída é a utilização de computadores paralelos ou um aglomerado de computadores. Ou seja, multiprocessadores e multicomputadores. A principal diferença entre eles está na presença ou ausência de memória compartilhada, o que interfere no modo como são projetados, construídos e programados além de interferir diretamente nos seus preços.

Para qualquer sistema de computação paralelo deve existir um meio de comunicação entre eles para trocar informações. Um multiprocessador compartilha uma memória comum entre todas as CPU's, podendo assim compartilhar um único espaço de endereço virtual mapeado na memória. Os processos são lidos e escritos respectivamente pela instrução *LOAD* e *STORE*. Esse modelo de comunicação foi bem aceito pelos programadores, além deles serem aplicados em diversos problemas.

Todas as CPU veem a mesma imagem e utilizam um único sistema operacional. Em consequência, há somente um mapa de página e uma tabela de processos. Esse sistema de uma única imagem que distingue um multiprocessador de um multicomputador, no qual cada computador tem sua própria cópia do sistema operacional.

Quando todos os processadores têm acesso igualitário nos módulos de memória e a todos os dispositivos de entrada e saída operado pelo sistema operacional temos um sistema denominado SMP (*Symmetric MultiProcessor* –

Multiprocessador Simétrico). Nos multicomputadores cada CPU tem sua própria memória privada, a qual pode acessá-la apenas através das instruções *LOAD* e *STORE*. Cada computador têm um único espaço de endereço físico para cada CPU. Para que haja comunicação entre os demais CPU's, é utilizada uma rede de interconexão. A ausência de memória compartilhada em *hardware* em um multicomputador traz sérias implicações para a estrutura do *software*. Não tem como ter um único espaço de endereço virtual para que todos os processos possam ser lidos e escritos apenas executando instruções *LOAD* e *STORE*.

A comunicação entre processos em um multicomputador costuma usar primitivas de *software* tais como *sende receive*. Este tipo de comunicação deixa essa estrutura bem mais complicada. Uma vez que, subdividir os dados corretamente e posicioná-los em localizações ótimas é fundamental em um multicomputador, diferente do multiprocessador. Conclui-se, então, que programar um multicomputador é muito mais difícil do que programar um multiprocessador. Porém, há suas vantagens em construir um multicomputador: é mais fácil de construir, o custo é bem menor, pode aumentar o poder de processamento adicionando outros computadores (PINHO; MAZZA; ROQUE, 2006).

Desvantagens em construir um multiprocessador são: mais difícil de construir, em compensação, mais fácil de programar. Com as vantagens e desvantagens desses sistemas gerou-se um esforço em construir sistemas híbridos, os quais são fáceis de construir e relativamente fáceis de programar. As pesquisas em computação paralela se convergiram entre as duas arquiteturas, combinando as forças de cada uma.

Um dos grandes problemas é como adicionar novas CPU mantendo o sistema escalável. Uma solução é usar *hardware* de multicomputadores fazendo com que o sistema operacional simule memória compartilhada obtendo assim um único espaço de endereço virtual de compartilhamento entre as páginas em todo o sistema, o que é conhecido com DSM (*Distributed Shared Memory* – Memória Compartilhada Distribuída). O sistema operacional apenas atende as falhas (TANEMBAUM 2006, p.33).

Outra forma de supercomputador é a computação em *cluster*, que pode conter milhares de computadores conectados. Pode ser fortemente acoplado – quando duas ou mais CPU estão perto uma da outra, tem alta largura de banda, ou pode ser fracamente acoplada quando as CPU ficam distantes entre si, tem baixa largura de banda. A troca de mensagens desse supercomputador se dá através da interconexão de redes de alta velocidade. Os tipos de computadores utilizados

podem ser homogêneos ou heterogêneos. Eles não precisam ter periféricos como: monitores, mouse, teclados, logo o custo de um *cluster* se torna bastante viável.

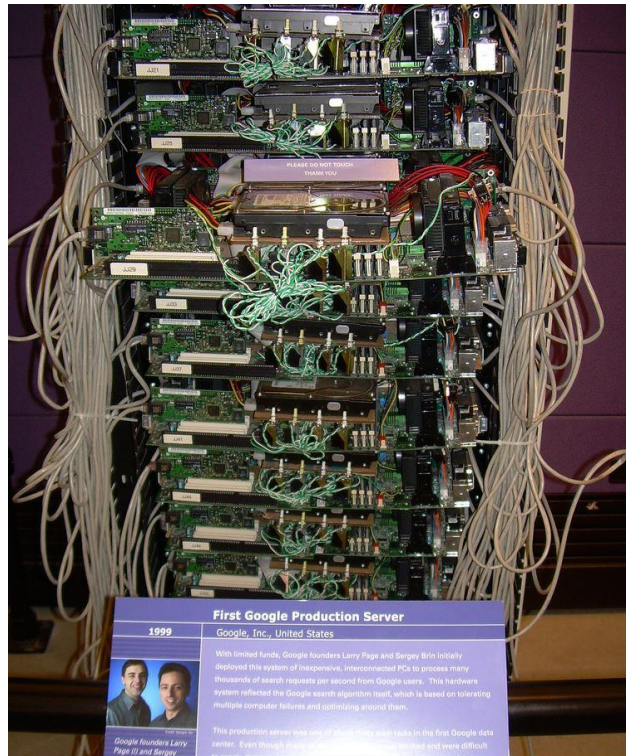
2.7 Tarefas onde são aplicadas a Computação em *Cluster*

Um *cluster* pode ser aplicado em diversas tarefas, dentre elas as mais complexas, ou seja, as que exijam alto poder de processamento são realizadas com mais eficiência e rapidez. Podemos citar como exemplos de tarefas que exigem alto poder de processamento:

- Servidores WEB;
- Engenharia Genética;
- Cálculos Científicos;
- Banco de Dados;
- Processamento de imagens;
- Distribuição de Cargas, etc.

Podemos apresentar ainda, como exemplo, o grande número de empresas, principalmente aquelas que trabalham com tecnologias ligadas a web como o *Google*, *Yahoo*, *LocaWeb*, *KingHost*, etc., que aproveitam todo o poder computacional de *clusters* para garantir a agilidade e a disponibilidade de seus serviços, na imagem a seguir o primeiro cluster desenvolvido pela Google.

Figura 12: Primeiro *cluster* do Google



Fonte: Computer History Museum, São Francisco

As empresas citadas como exemplo, tem áreas que requerem um grande poder de processamento devido às realizações de testes, experiências e estudos mais aprofundados. Essas necessidades tornaram o *cluster* mais difundido devido à facilidade de implantação, ao baixo custo e ao alto poder de processamento que se pode chegar.

2.8 Funcionamento Básico Dos *Clusters*

Para que um *cluster* seja constituído, é necessário fazer uso de alguns elementos básicos. O primeiro são os equipamentos a serem utilizados como nós. Pode-se usar máquinas apenas para funcionar como nós. Neste caso, os computadores teriam apenas dispositivos de *hardware* (CPU) sem a necessidade de ter monitor, teclado, mouse ou outros periféricos.

Cluster avançado construído com equipamentos específicos, mas também é possível utilizar computadores "convencionais", como *desktops* para fins domésticos ou para uso em escritório. Assim, uma universidade ou uma empresa, por exemplo, pode utilizar máquinas que foram substituídas por modelos mais recentes para criar

um *cluster* e, eventualmente, economizar com a aquisição de servidores. Os nós podem ainda ser não dedicados ou dedicados. No primeiro caso, cada computador que faz parte do *cluster* não trabalha exclusivamente nele. No segundo, o nó é utilizado somente para este fim, fazendo com que dispositivos como teclados e monitores sejam dispensáveis - se, por algum motivo, for necessário acessar uma máquina em particular, pode-se fazê-lo via terminal, a partir do nó principal, por exemplo.

Outro elemento importante é o sistema operacional. Como já informado, os nós escravos não precisam ser exatamente iguais no que diz respeito ao *hardware*, mas é recomendado que todos possuam configurações idênticas, e o mesmo sistema operacional. Esta homogeneidade é importante para diminuir a complexidade de configuração e manutenção do sistema, e garantir que os procedimentos rotineiros ao *cluster*, como monitorização, distribuição de tarefas e controle de recursos sejam executados de maneira uniforme. Para reforçar estes aspectos, pode-se até mesmo adotar sistemas operacionais preparados especialmente para *clustering* (KIRNER, 2001).

Do ponto de vista do *software*, o *cluster* conta ainda com o elemento que faz o papel de *middleware*: trata-se de um sistema que permite o controle do *cluster* em si e, portanto, está intimamente ligado ao sistema operacional. É o *middleware* que lida, por exemplo, com as bibliotecas que fazem toda a comunicação do *cluster* - uma delas é o padrão MPI (*Message Passing Interface*). Além de trabalhar com o gerenciamento do *cluster*, o *middleware* oferece uma interface para que um administrador possa configurar o *cluster*, ferramentas para manutenção e otimização, recursos de monitoramento e assim por diante (KIRNER, 2001).

Por padrão, o *middleware* é instalado em uma máquina chamada de nó mestre. O nome deixa claro: trata-se do já mencionado nó principal, que efetivamente controla o *cluster* a partir da distribuição de tarefas, do monitoramento e de procedimentos relacionados. A comunicação entre os nós - que é onde está a delimitação do que constitui o *cluster* em si - é feita a partir de uma tecnologia de rede local. Os padrões *Ethernet* (*GigabitEthernet*, *FastEthernet*, etc.) são bastante utilizados justamente por serem mais comuns e, portanto, melhor suportados e menos custosos. Mas há outras opções viáveis, entre elas, o *Myrinet* e o *InfiniBand*, ambos com características bastante apropriadas para *clustering* (MAZZA, et al., 2004).

2.9 Principais pontos positivos do uso do *cluster*

Os principais pontos positivos na implementação de um *cluster* por uma determinada entidade:

- Redução de custos com aquisição de computadores de alta capacidade;
- Aumento da produtividade (realizar mais tarefas em menos tempo);
- Prevenção de perda ou inutilização de equipamentos legados;
- Aumento da disponibilidade de serviços (com técnicas de *failover* automático);
- Diminuição de lixo tecnológico;

Embora os *clusters*, atualmente, já sejam bastante conhecidos e utilizados, a maior parte de suas aplicações são em empresas que lidam com buscas na web ou com análises que envolvam muitos cálculos simultâneos (normalmente, pesquisas científicas), não sendo, portanto, muito comuns no ambiente corporativo de empresas de médio e grande porte que não sejam voltadas à TI, que ainda preferem investir seus recursos na compra de equipamentos de grande porte de qualidade reconhecida, de marcas como Dell, HP, IBM entre outras.

Porém, com o advento da virtualização, é provável que esse panorama, aos poucos, comece a mudar, uma vez que o *cluster* se torna extensível, não sendo mais necessário um *cluster* para cada serviço. Um único *cluster* pode, dessa forma, hospedar vários (ou até mesmo todos) serviços necessários à entidade, em ambientes operacionais diferentes, porém com todos eles se valendo do aumento de processamento e disponibilidade da tecnologia.

Um outro fator positivo para o aumento na utilização dos *clusters* é o custo para as empresas, uma vez que os equipamentos voltados à “computação pessoal” estão cada dia mais baratos, enquanto que os voltados à “computação *corporate*” mantêm os seus preços, o que pode servir para fomentar a compra de vários equipamentos mais baratos ao invés de investir pesado em um único equipamento que, muitas vezes, será subutilizado (SCHAEFER, et al., 2002).

Por outro lado, a computação distribuída ainda é vista com certo receio por muitos profissionais de TI, por não conhecerem os seus benefícios, ou ainda, por não possuir embasamento teórico necessário para perceber suas vantagens. Um

outro ponto negativo é o trabalho inicial para se montar um ambiente de *clustering* para uma determinada entidade. É muito mais simples para o profissional encomendar um servidor de grande porte de boa marca, com todas as redundâncias possíveis, e deixá-lo funcionando do que lidar com vários equipamentos ao mesmo tempo, realizando todos os ajustes finos necessários para atender da melhor maneira possível à necessidade do cliente, uma vez que a segunda alternativa, além de tempo, demanda um conhecimento teórico maior.

Figura 13: *Cluster* avançado construído com equipamentos específicos



Fonte: <http://www.infowester.com/cluster.php>

Portanto, embora a tecnologia de *clustering* seja bastante benéfica em vários pontos para praticamente qualquer corporação, o seu crescimento tende a ser lento, seja pela credibilidade que algumas marcas de equipamentos conseguiram (com justiça) obter junto aos seus clientes com relação aos seus equipamentos de grande porte, seja pela falta de conhecimento e/ou tempo da equipe de TI da entidade para propor (ou montar) um *cluster* ou ainda, pela falta de informação da entidade como um todo sobre esta tecnologia (SOUZA; ROQUE, 2004).

2.10 Modelos de *Cluster*

Há uma enormidade de aplicações que só podem ser atendidas satisfatoriamente com computação de alto desempenho. Com o advento da computação em nuvem, este cenário se torna se, ainda mais amplo: pode-se ter uma infraestrutura tecnológica respondendo a vários clientes simultaneamente de maneira remota, por exemplo. Em todos estes casos e em qualquer outro tipo de aplicação crítica - que não pode parar de funcionar ou não pode perder dados (os sistemas bancários, por exemplo) -, o *cluster* pode se mostrar como uma solução viável, desde que o tipo mais adequado seja escolhido.

Para que o *cluster* atenda as mais diversas áreas, foi necessário classificá-lo em tipos, os principais são: *cluster* de alto desempenho, *cluster* de alta disponibilidade e *cluster* de balanceamento de carga, cada um com uma finalidade específica.

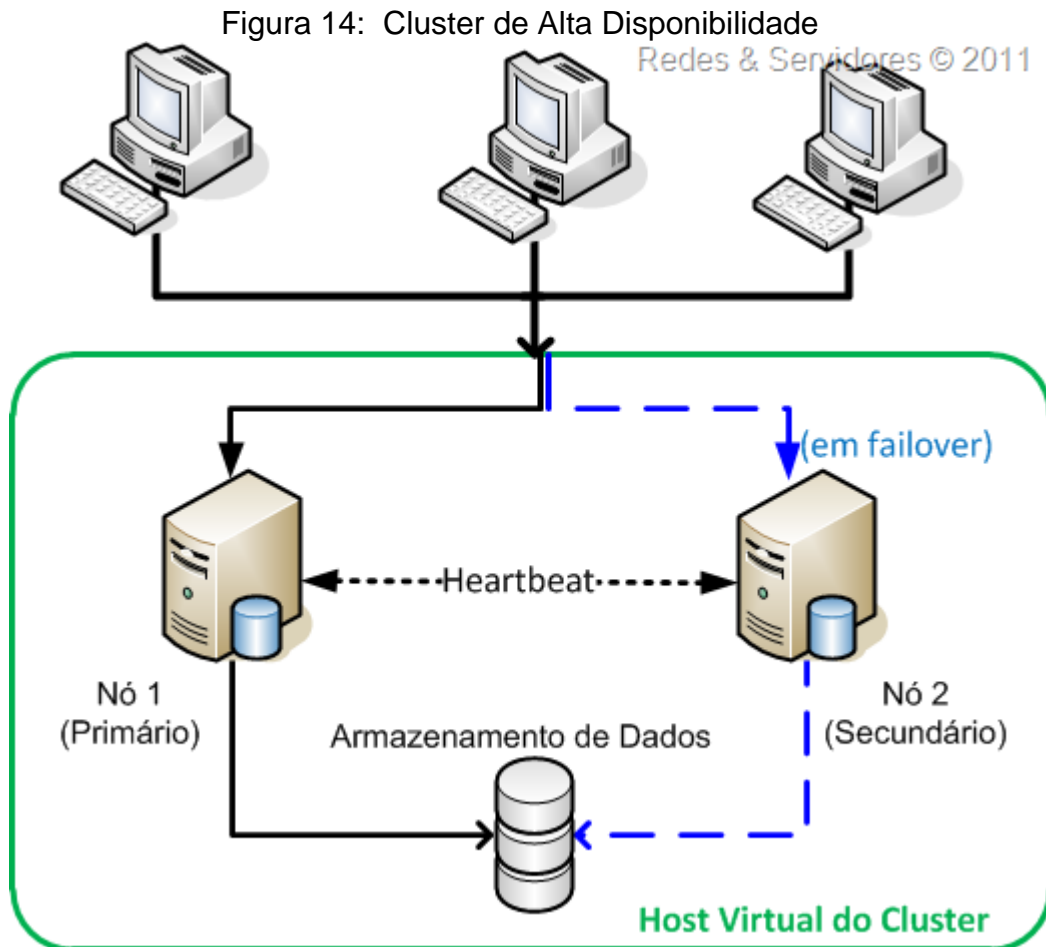
2.10.1 cluster para alta disponibilidade - high availability

Nos *clusters* de alta disponibilidade, o foco está em sempre manter a aplicação em pleno funcionamento: não é aceitável que o sistema pare de funcionar, mas se isso acontecer, a paralisação deve ser a menor possível, como é o caso de soluções de missão crítica que exigem disponibilidade de, pelo menos, 99,999% do tempo a cada ano, por exemplo.

Para atender a esta exigência, os *clusters* de alta disponibilidade podem contar com diversos recursos: ferramentas de monitoramento que identificam nós defeituosos ou falhas na conexão, replicação (redundância) de sistemas e computadores para substituição imediata de máquinas com problemas, uso de geradores para garantir o funcionamento em caso de queda de energia, entre outros.

Em determinadas circunstâncias, é tolerável que o sistema apresente algum grau de perda de desempenho, especialmente quando esta situação é consequência de algum esforço para manter a aplicação em atividade. Utilizado para controlar a distribuição de carga entre os demais computadores, mantendo-as equilibradas. Precisa de um monitoramento constante na sua comunicação para que em caso de falha não haja uma interrupção dos serviços disponibilizados pelo *cluster*. (Pitanga 2004).

Exemplo: Utilização de dois computadores como servidores de páginas de Internet, onde um fica ligado normalmente e o outro fica em alerta, se acaso o que está em funcionamento falhar, o que está em alerta assume imediatamente a função. Isso manterá a página no ar sem que haja interrupções, como ilustrado na figura abaixo.



Fonte: <https://redes-e-servidores.blogspot.com/2011/04/failover-clustering-i.html>

2.10.2 Cluster para balanceamento de carga - *load balancing*

Em *clusters* de balanceamento de carga, as tarefas de processamento são distribuídas o mais uniformemente possível entre os nós. O foco aqui é fazer com que cada computador receba e atenda a uma requisição e não, necessariamente, que distribua uma tarefa com outras máquinas. Por exemplo, um grande site na internet recebe por volta de mil visitas por segundo e que um *cluster* formado por 20 nós tenha sido desenvolvido para atender a esta demanda.

Como se trata de uma solução de balanceamento de carga, estas requisições são distribuídas igualmente entre as 20 máquinas, de forma que cada uma receba e realize, em média, 50 atendimentos a cada segundo. Não basta ao *cluster* de balanceamento de carga ter um mecanismo meramente capaz de distribuir as requisições - é necessário que este procedimento seja executado de forma a garantir um "equilíbrio" na aplicação.

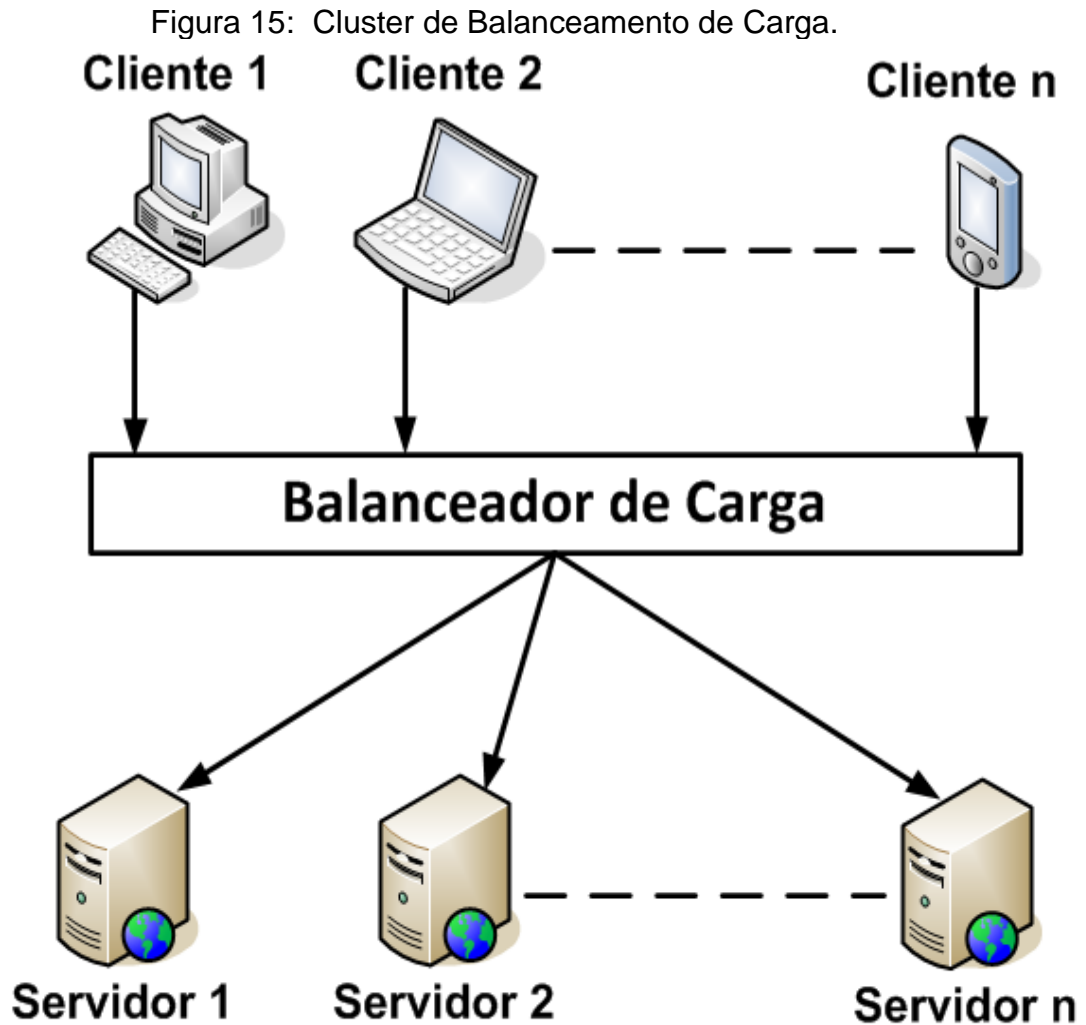
Para tanto, o mecanismo pode monitorar os nós constantemente para verificar, por exemplo, qual máquina está lidando com a menor quantidade de tarefas e direcionar uma nova requisição para esta. O balanceamento de carga pode ser utilizado em vários tipos de aplicações, mas o seu uso é bastante comum na internet, já que soluções do tipo têm maior tolerância ao aumento instantâneo do número de requisições, justamente por causa do equilíbrio oriundo da distribuição de tarefas.

Utilizados para melhorar o desempenho das tarefas computacionais. Fraciona-se o processamento total entre os computadores buscando-se um melhor desempenho (PITANGA, 2004).

Os *clusters* de balanceamento de carga fornecem escalabilidade e alta disponibilidade a aplicativos e serviços baseados em TCP e UDP pois combinam até 32 servidores executando o *Windows Server 2003, Web Edition*, o *Windows Server 2003, Standard Edition*, o *Windows Server 2003, Enterprise Edition* ou o *Windows Server 2003, Data center Edition* em um único *cluster*. Com a utilização do Balanceamento de carga de rede para criar um grupo de computadores de *cluster* clonados, ou idênticos, é possível aumentar a disponibilidade desses servidores: Servidores Web e de FTP (*File Transfer Protocol*), servidores ISA (para serviços de servidores proxy e firewall), servidores de rede virtual privada (VPN), servidores *Windows Media*, Serviços de terminal através de rede de longa distância corporativa (PITANGA, 2004).

Você pode instalar *clusters* de balanceamento de carga de rede através de Conexões de rede ou usando o Gerenciador de balanceamento de carga de rede. Para obter mais informações sobre *clusters* de balanceamento de carga de rede.

Exemplo: Servidores de páginas de Internet. Se houver 1000 visitas por minuto, estas serão divididas pela quantidade de computadores. Portanto, no caso de dez computadores cada um terá que lidar com 100 visitas por minuto.



Numa situação de balanceamento de carga no servidor, todos os pedidos são tratados e distribuídos por um balanceador de carga central.

Fonte: <https://redes-e-servidores.blogspot.com/2011/04/balanceamento-de-carga-v.html>

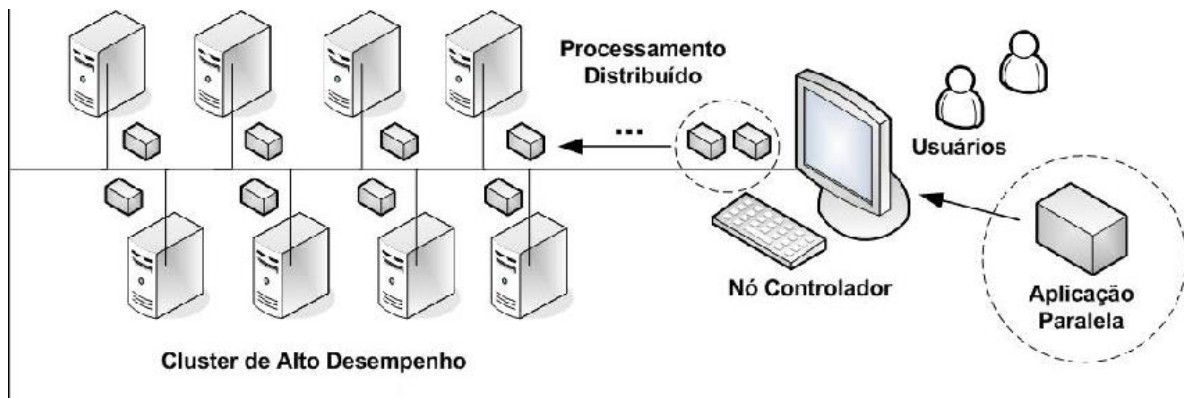
2.10.3 Cluster para alto desempenho

Clusters de alto desempenho são direcionados a aplicações bastante exigentes no que diz respeito ao processamento. Sistemas utilizados em pesquisas científicas, por exemplo, podem se beneficiar deste tipo de *cluster* por necessitarem analisar uma grande variedade de dados rapidamente e realizar cálculos bastante complexos. O foco deste tipo é o de permitir que o processamento direcionado à aplicação forneça resultados satisfatórios em tempo hábil, mesmo que haja centenas de milhares de *Gigaflops* envolvidos com a tarefa (1 GFLOP corresponde a 1 bilhão de instruções de ponto flutuante executadas por segundo).

O *cluster* de alto desempenho tem a finalidade de aumentar a disponibilidade dos serviços que estão sendo executados em conjunto. Está relacionado ao conceito de redundância, ou seja, a mesma tarefa sendo executada por outros computadores (Pitanga 2004).

Exemplo: Para cálculos científicos voltados a previsões meteorológicas e financeiras. Se forem utilizados 100 computadores o tempo será de até 100 vezes mais rápido do que se tivesse sido feito em um único computador.

Figura 16: Cluster de Alto Desempenho



Fonte: https://www.researchgate.net/figure/Figura-03-Cluster-de-Alto-Desempenho_fig2_228555383

3 PROPOSTA

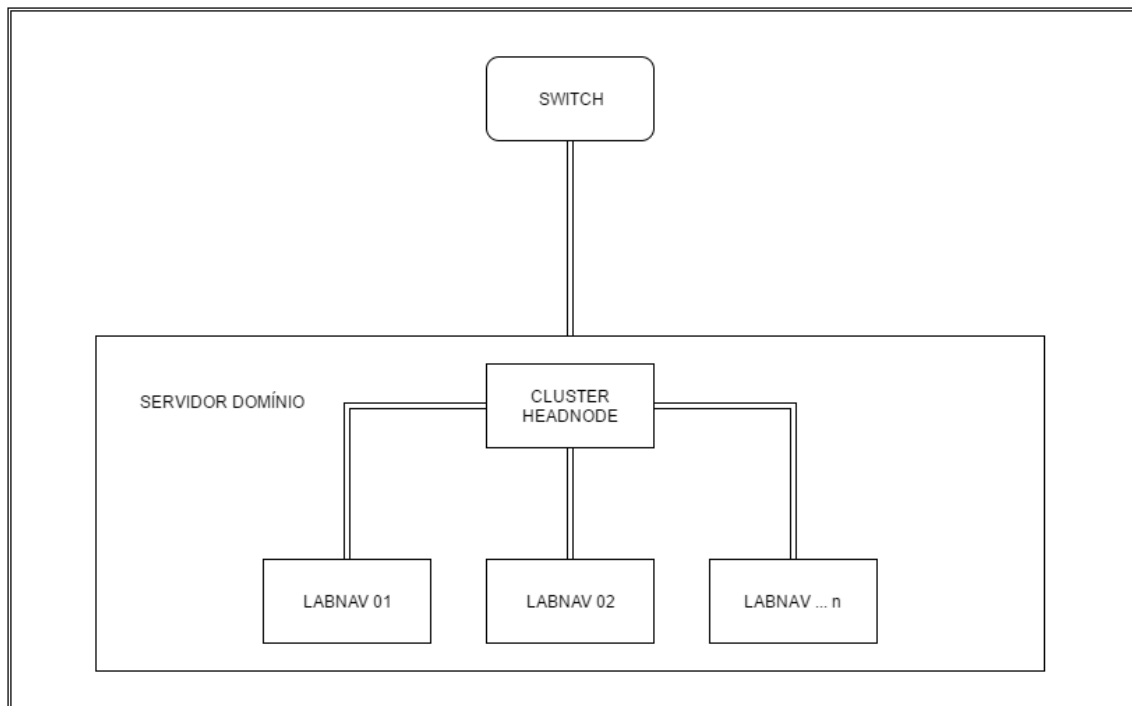
Há atualmente um grande número de aplicações da FENAV que possuem alta complexidade de cálculos e necessitariam de supercomputadores de grande capacidade de processamento, porém, com custos elevadíssimos, torna a aquisição a nível educacional praticamente inviável, sendo assim, um *cluster* apresenta a solução mais rápida e viável de se obter resultado, sendo de baixo custo comparado a um supercomputador.

Utilizando-se dos computadores do próprio laboratório de engenharia naval, que são dedicados ao ensino dos alunos do curso, apresentando os mais diversos *softwares* de grande porte empresarial e educacional, como *ShipFlow*, DNV (*Det Norske Veritas*), *Ansys*, *AutoCAD*, *REVIT*, entre outros.

3.1 Implementação do cluster de alto desempenho

O modelo de *cluster* escolhido foi o de alto desempenho, pois como já foi citado anteriormente, é necessário resolver cálculos complexos. Os computadores estão interligados através de um domínio denominado *labnav.cluster.br*, onde podem se comunicar e realizar as operações em paralelo, para isso, foi selecionado uma máquina para funcionar como servidor de domínio utilizando-se do *WindowsServer 2003*, o gerenciamento do *cluster* é feito a partir de uma outra máquina utilizando-se do *WindowsServer 2008 HPC Pack R2*, sendo instalado a aplicação *Microsoft HPC Pack 2008 R2* nesse servidor, que será chamado *HEADNODE*, sendo também instalado no restante das máquinas, que serão chamadas de *LABNAV01*, *LABNAV02*, ..., *LABNAVn*, que são os nodes computacionais do *cluster*, interligados através dessa plataforma ao nó principal (*HEADNODE*), todas as máquinas estão interligadas pelo servidor de domínio compartilhando o mesmo usuário, assim, não é necessário reconfigurar, bastando apenas alternar os usuários.

Figura 17: Layout da rede de computadores, Headnode e *Switch*



Fonte: Adaptado de Russel (2005)

- *Hardware*: Todos os computadores pertencentes ao *cluster* possuem o mesmo *hardware*, que está mais detalhado no Apêndice A – Especificação *Hardware* do *Cluster*.
- Padrões de Rede: O laboratório possui um *switch* 2952-SFP PLUS que conecta todo o prédio do laboratório de engenharia naval. Todos os nós possuem o mesmo adaptador de rede NVIDIA nForce 10/100/1000 Mbps *Ethernet*, com os drivers mais recentes instalados.
- Sistema Operacional: Por necessidade de adaptação ao funcionamento do laboratório dedicado aos alunos de engenharia naval, foi escolhido a plataforma *Windows* como base para implementação do *cluster*, pois não seria necessário alterar o sistema operacional e reconfigurar as máquinas dos usuários, o *cluster* possui:
 - 1(um) *WindowsServer* 2003 como controlador de domínio;
 - 1(um) *WindowsServer* 2008 HPC Pack R2 como gerenciador do
 - *Cluster*;
 - 24(vinte e quatro) *Windows 7 Ultimate* SP1 com a aplicação *Microsoft HPC Pack* 2008

3.2 Gerenciamento do *Cluster*

Contrariando o grande número de servidores *cluster* na plataforma *Linux*, a *Microsoft* lançou a versão HPC a partir do *WindowsServer 2003*, na versão *WindowsServer 2008*, ofereceu um pacote próprio, o *Microsoft HPC Pack 2008*, que fornece a interface e as ferramentas necessárias para o gerenciamento do *cluster*, tendo integração com outros produtos *Microsoft*, facilitando a adição de novos nós ao *cluster*, pois, basta ser executado em um *Windows 7* e configurado para aumentar a estrutura. Entre os destaques são prover segurança, recursos de gerenciamento escalável para o *cluster*, organizador de tarefas e interface de passagem de mensagem (MPI- *Message Passing Interface*) para programação paralela.

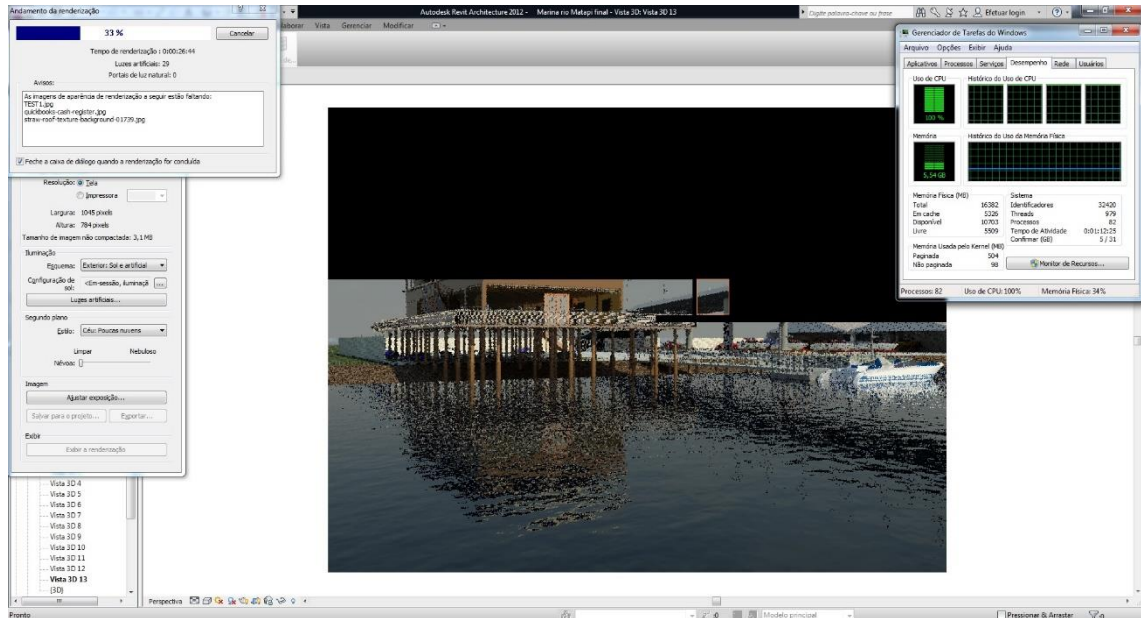
Com acesso ao *headnode*, usuários podem lançar trabalhos (*Job*) no *cluster*, cada *Job* é uma requisição de recursos submetida ao escalonador de trabalhos (*Job Scheduler*), que contém ou conterá uma ou mais tarefas (*tasks*). Uma tarefa não pode existir sem estar associada a um trabalho. Um trabalho pode conter uma única tarefa. O escalonador de trabalho é o serviço responsável por colocar trabalhos e tarefas em filas, alocar recursos, despachar tarefas para nós de computação, e monitorar o estado de trabalhos, tarefas e nós.

3.3 Aplicações

Necessita-se que o *software* seja compatível com processamento paralelo para funcionar corretamente dentro do *cluster*, há várias aplicações de engenharias que necessitam de grande poder computacional, e por isso, são completamente compatíveis com o processamento paralelo, em engenharia naval, temos alguns *softwares* que abrangem essa estrutura computacional, abaixo alguns desses *softwares*.

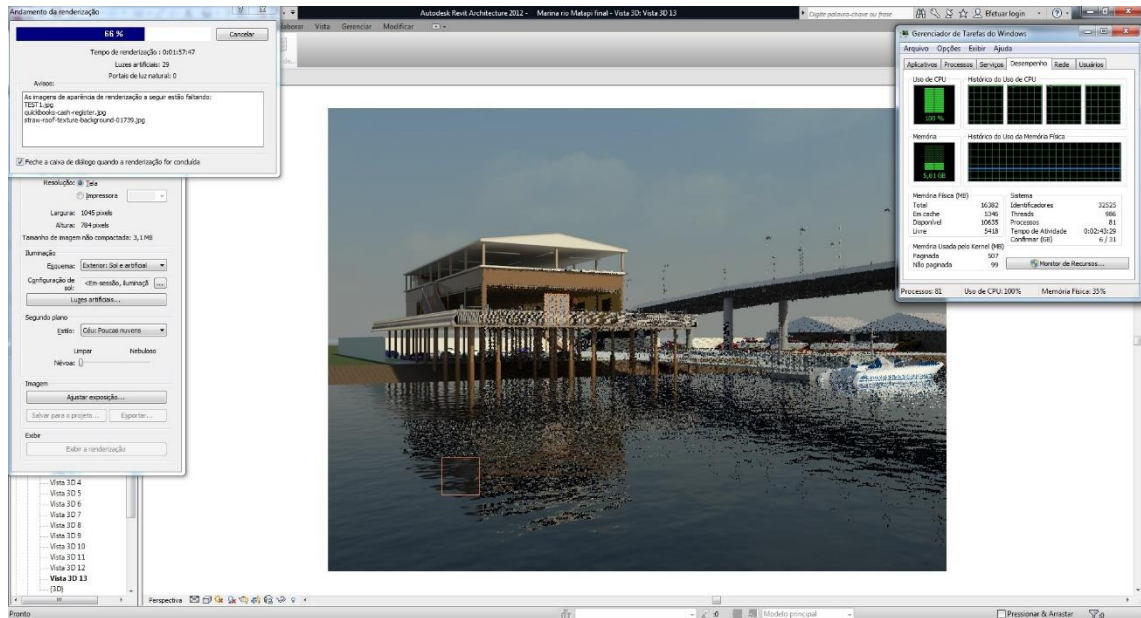
3.3.1 Autodesk

Figura 19: Etapa inicial de processo de renderização.



Fonte: Projeto de uma Marina realizado por LOUREIRO, Emanuel (2014)

Figura 20: Adicionando filtros e efeitos de iluminação à imagem



Fonte: Projeto de uma Marina realizado por LOUREIRO, Emanuel (2014)

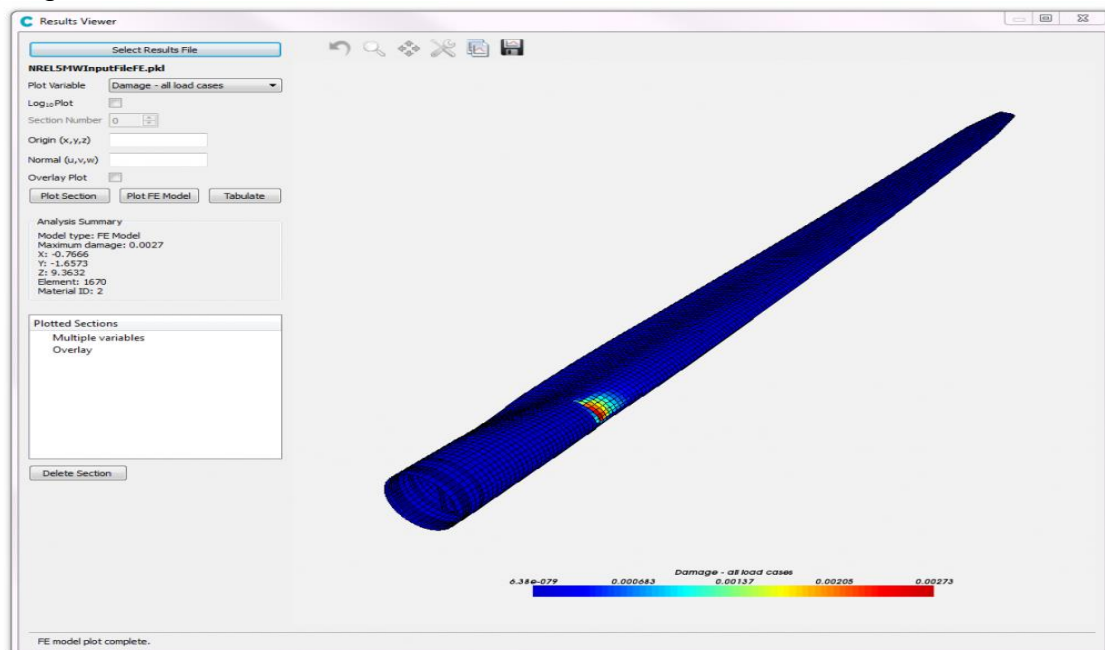
Figura 21: Imagem finalizada



Fonte: LOUREIRO; EMANNUEL (2014)

3.3.2 DNV (*det norske veritas*)

Figura 22: Tela de resultados apresentado por um dos módulos do programa.



Fonte: Exemplos encontrados no site do produto DNV.

3.3.3 Shipflow

Figura 23: SHIPFLOW, é compatível com cluster de Alto Desempenho



Fonte: Imagem encontrada no site do produto Shipflow

Na última versão, possui grande apoio para computação paralela, oferecendo suporte através da licença em rede computacional.

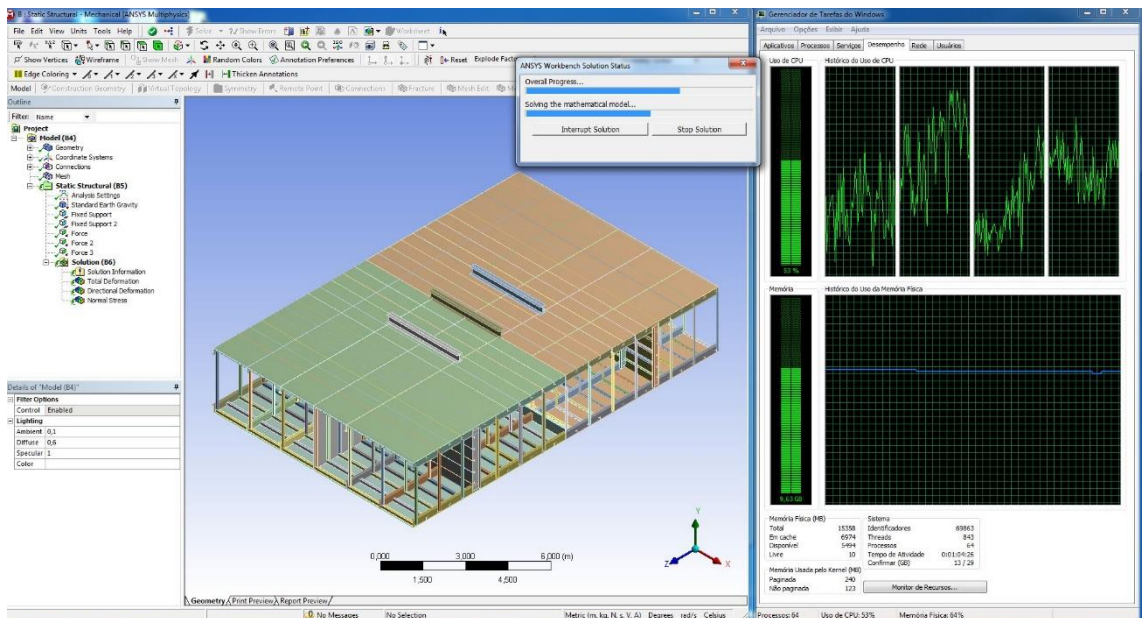
3.3.4 Ansys inc.

Foi o software escolhido para a realização dos testes dentro do cluster, é um desenvolvedor de programas de simulação para engenharia assistida por computador, possuindo várias vertentes, como *Ansys CFD (Computational Fluid Dynamics – Dinâmica de Fluidos Computacional)*, permite aos engenheiros testar sistemas através da simulação de escoamentos de fluidos em um ambiente virtual, como por exemplo, a dinâmica dos fluidos em cascos de navios, turbinas a gás (incluindo os compressores, câmara de combustão, turbinas e pós-combustão), aerodinâmica de aeronaves, bombas, ventiladores, HVAC (*Heating, Ventilating and Air Conditioning – AVAC – Aquecimento, Ventilação e Ar Condicionado*), navios de mistura, hidro ciclones, aspiradores, tecnologia de simulação em: Mecânica Estrutural, *Multiphysics*, Eletromagnetismo, Hidrodinâmica, *Workflow, Workbench* e diversos outros segmentos.

Apresentando várias áreas de interesse de engenharia naval, sendo compatível com processamento paralelo, servirá como estudo de caso para a análise da proposta.

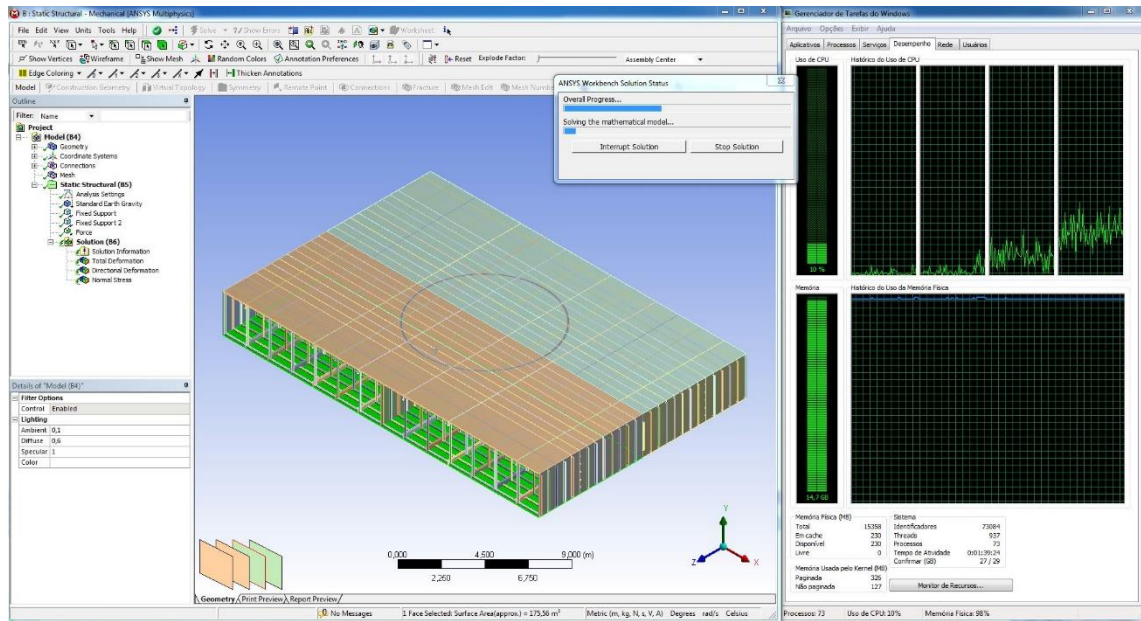
A seguir, algumas telas do funcionamento do programa com o uso e consumo de processamento e memória do sistema. Na Figura 20 temos a análise estrutural de uma partição de um casco de navio, são feitos vários testes através de elementos finitos onde se calcula a resistência dos materiais selecionados para a construção do navio. A Figura 21 apresenta uma partição bem maior que a primeira selecionada e a Figura 22 é o cálculo das duas partições em simultâneo.

Figura 24: Análise estrutural de uma partição pequena do casco de navio.



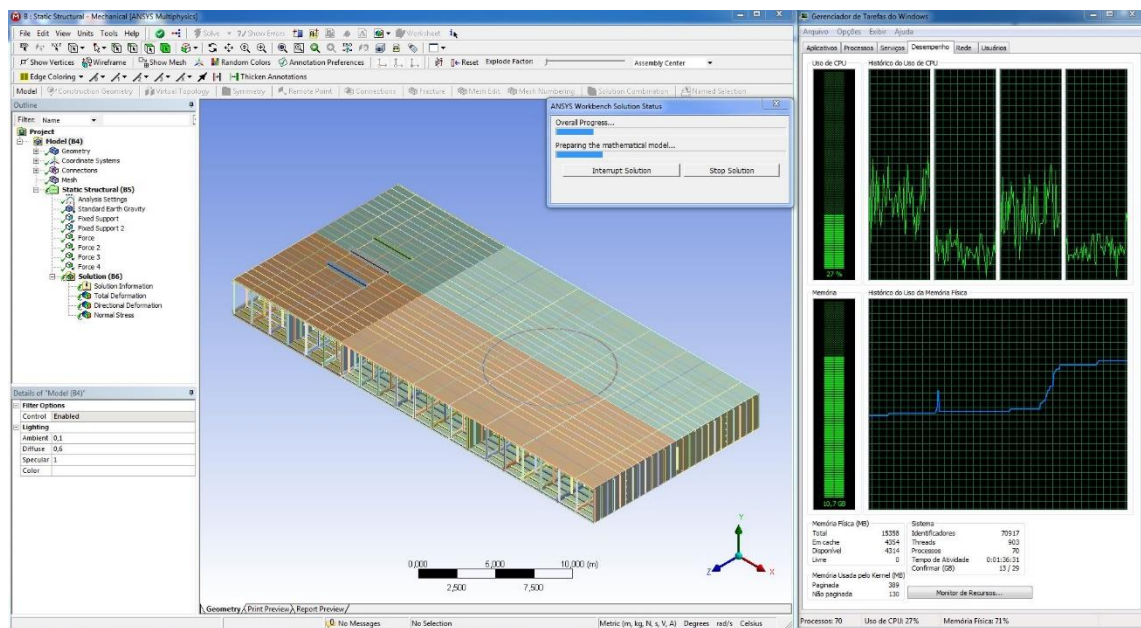
Fonte: Autoria própria

Figura 25: Análise estrutural de uma segunda partição maior do casco de navio.



Fonte: Autoria própria

Figura 26: Análise estrutural das duas partições anteriores em simultâneo.



Fonte: Autoria própria

A tabela a seguir apresenta a comparação do tempo de processamento em cada etapa:

Tabela 3: Comparação TESTE X Tempo

TESTE A – PARTIÇÃO MENOR	3050 SEGUNDOS ~ 50 MINUTOS
TESTE B – PARTIÇÃO MAIOR	25762 SEGUNDOS ~ 429 MINUTOS
TESTE AB – AS DUAS PARTIÇÕES	36935 SEGUNDOS ~ 615 MINUTOS

Fonte: Autoria própria

3.4 Laboratório de Engenharia Naval

Localizado no campus do Guamá da Universidade Federal do Pará, o laboratório integra a primeira etapa do Complexo de Ensino e Pesquisa em Engenharia Naval e Hidroviária da UFPA. O seu laboratório de informática oferece aos alunos 24 PC's idênticos, todos instalados e configurados para dar total apoio ao ensino de engenharia naval, o curso conta com cinco turmas de 20 alunos, uma por ano, e funciona como sala de aula durante os semestres do curso, matérias como Introdução à Ciência da Computação, Introdução ao CAD, entre outras, há ainda a possibilidade de minicursos serem aplicados.

Figura 27: Instalação e configuração dos computadores, aqui dispostos em 4 fileiras com 3 equipamentos em cada. (setembro/2013)



Fonte: Autoria própria

Figura 28: Outro ângulo apresentando as demais fileiras e o hack com o *switch* ao fundo.(setembro/2013)



Fonte: Autoria própria

Os alunos podem também usar livremente o espaço para estudos fora dos horários das disciplinas, o acesso é livre. Pensando nisso, a implementação do *cluster* deve obedecer essa estrutura sem alterar o acesso ou o uso dos alunos, que são os grandes beneficiados. Para isso, será adicionada uma conta de usuário veiculada ao controlador de domínio (utilizando o mesmo login (*CLUSTER\Administrador* e senha) e somente serão usados quando algum trabalho for executado dentro do *cluster*, o controlador de domínio e o gerenciador do *cluster* ficarão em outra sala, interligados via *switch* ao sistema, ou seja, não ficarão presentes na mesma sala e poderão ser usados livremente para eventuais testes e correções.

4 ANÁLISE DA PROPOSTA

Como foi apresentado nos capítulos anteriores, houve um crescimento de programas que oferecem suporte para computação paralela de alto desempenho, laboratórios científicos e de engenharia em instituições acadêmicas e indústrias estão sendo beneficiados por estas aplicações.

Esta proposta visa a configuração de um *cluster* de alto desempenho na plataforma *Windows* para os estudantes e corpo docente do curso de engenharia naval.

A implementação do *cluster* HPC na plataforma *Windows* deve obedecer algumas etapas.

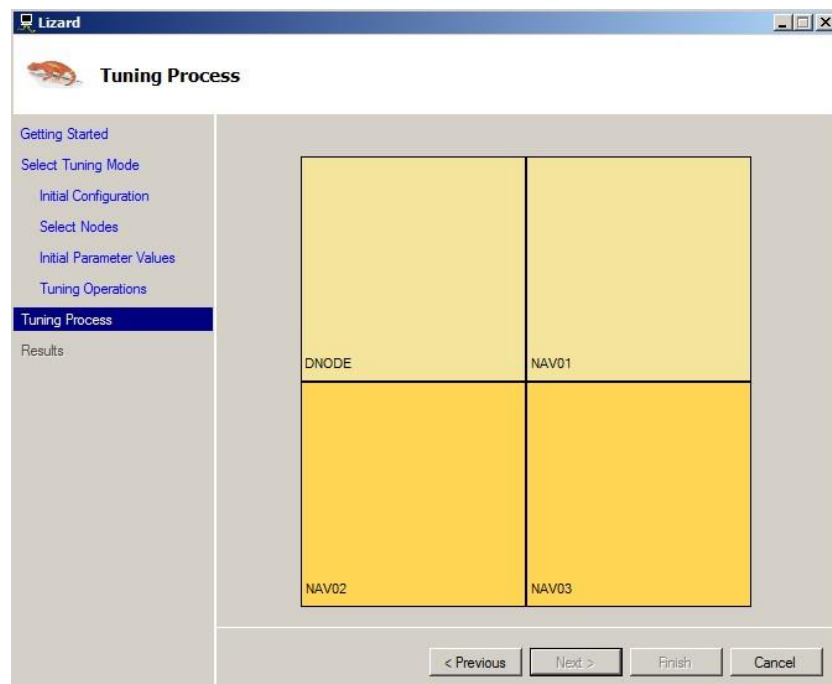
- Criar um servidor de domínio;
- Criar o nó principal HPC (HEADNODE) e vincular ao domínio;
- Adicionar nós WORKSTATION ao nó principal e ao domínio
- Executar testes de desempenho.

Mas para isso, é necessário definir o que é um servidor de domínio e sua função. O domínio é uma estrutura lógica que compartilha uma central de serviços e diretórios. A base de dados contém contas de usuários e segurança da informação do domínio. Com uma única conta, os usuários podem validar-se no domínio a partir de qualquer máquina, a partir daí, usufruir dos recursos da rede para os quais o administrador do domínio lhes der permissões. Assim, pode compartilhar o mesmo usuário para os nós do *cluster*, compartilhando sua estrutura de rede não precisando reconfigurar manualmente cada computador participante do domínio.

Um breve detalhamento do processo de instalação e configuração está descrito no Apêndice B.

4.1. Testes de Desempenho

Foi utilizado o *software Lizard LinPack* fornecido pela *Microsoft* para determinar o desempenho e a eficiência computacional que podem ser obtidos pelo *cluster* baseado no *HPC Pack 2008*, e *Windows HPC Server 2008*. Ele calcula e reporta um valor máximo de desempenho em bilhões de operações de ponto flutuante por segundo (GFLOPS) e um valor percentual para a eficiência alcançada no desempenho máximo, se utilizando de uma fórmula matemática para alcançar o máximo desempenho dos processadores dos nós do *cluster*.

Figura 29: Tela do *software* Lizard Linpak.

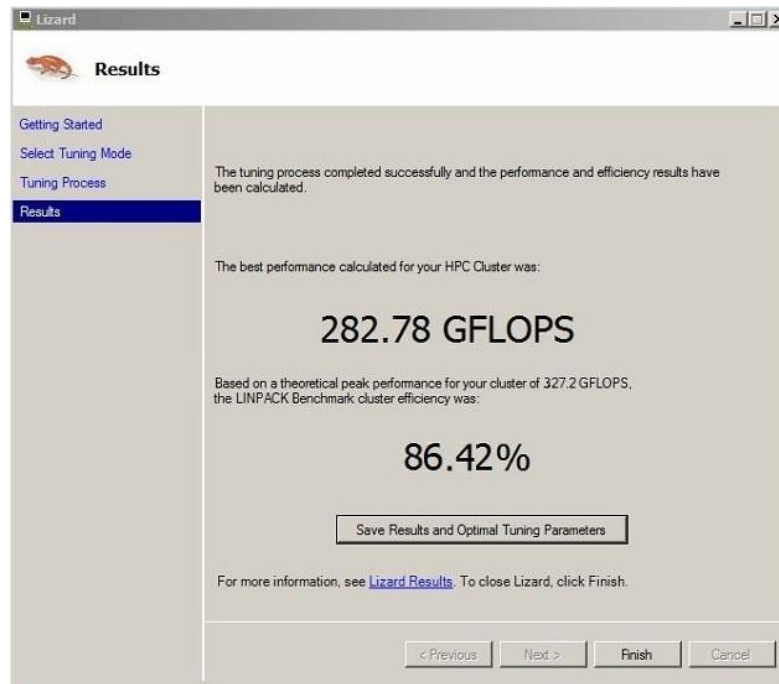
Fonte: Autoria própria

O *software* realiza diversos testes de conectividade da rede, e demais requisitos para o correto funcionamento do *cluster*, sendo ideal para avaliar se a configuração está correta. Após os testes, o *software* retorna os valores em GFLOPS dos cálculos efetuados, fornecendo um valor numérico do poder computacional que foi alcançado com o *cluster* configurado.

Como apresentado na Figura 30 conseguimos um valor de 282,78 GFLOPS utilizando 8 nós do *cluster*.

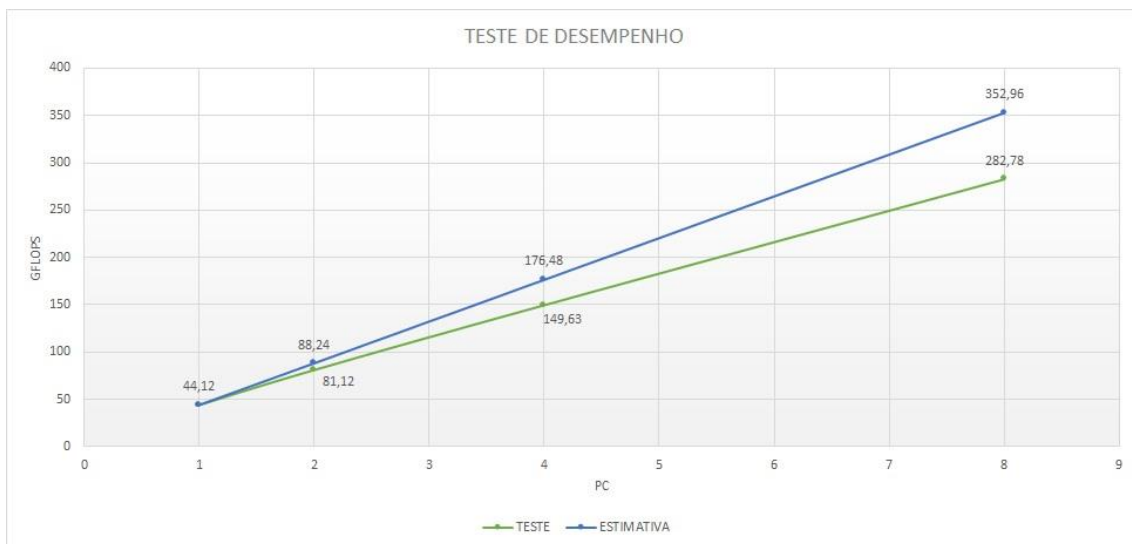
Na Figura 32, o gráfico comparativo dos testes realizados vezes a estimativa, sendo esta, o valor esperado ao dobrar os recursos computacionais obtidos com o primeiro resultado de 44,12 GFLOPS para um computador, nota-se a perda de desempenho entre cada nó adicionado ao *cluster*, pois há perda de velocidade no ambiente real pelo tempo de resposta da rede, com base nessa informação, estima-se um valor total de 1 TFLOPS com 24 PCs interligados, que é o mesmo poder computacional que o último colocado no TOP500 (lista de supercomputadores criada em 1993 que é atualizada em junho e novembro com os 500 computadores mais rápido do mundo) obteve em 2005.

Figura 30: Resultado em GFLOPS após todos os cálculos.



Fonte: Autoria própria

Figura 31: Gráfico comparativo entre o desempenho em GFLOPS X Nós do *cluster*, e a estimativa de valores ideais.



Fonte: Autoria própria

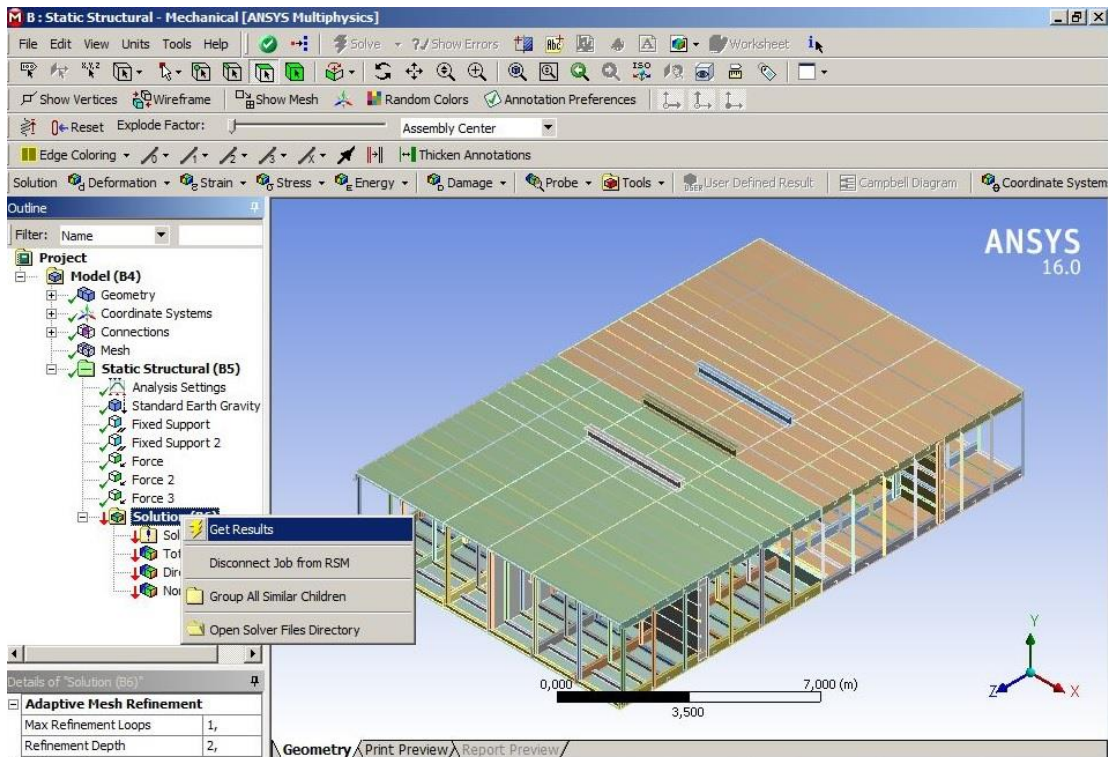
4.2 Análise do *Cluster* de Alto Desempenho

Para a análise de desempenho do *cluster* foi utilizado o *software* ANSYS INC, o teste consiste em analisar o material (aço, alumínio, etc.), utilizado na construção

de um navio, com isso é possível atestar se o projeto e material utilizado é viável ou não, o TESTEA, o TESTEB, TESTEAB.

O *software* fornece suporte a várias opções de computação paralela em diversos sistemas operacionais, e disponibiliza a ferramenta solver, para resolução de cálculos matemáticos da aplicação que fora projetada, presente na aplicação *ANSYS MULTIPHYSICS*, que separa o cálculo das deformações em vários pedaços iguais e envia-os para os nós do *cluster*, após cada nó finalizar o cálculo, a ferramenta junta-os novamente através da função *GetResults*, como visto na Figura 32.

Figura 32: Função *Get Results* da ferramenta Solver, é utilizada para obter os cálculos computacionais da rede do *cluster*.

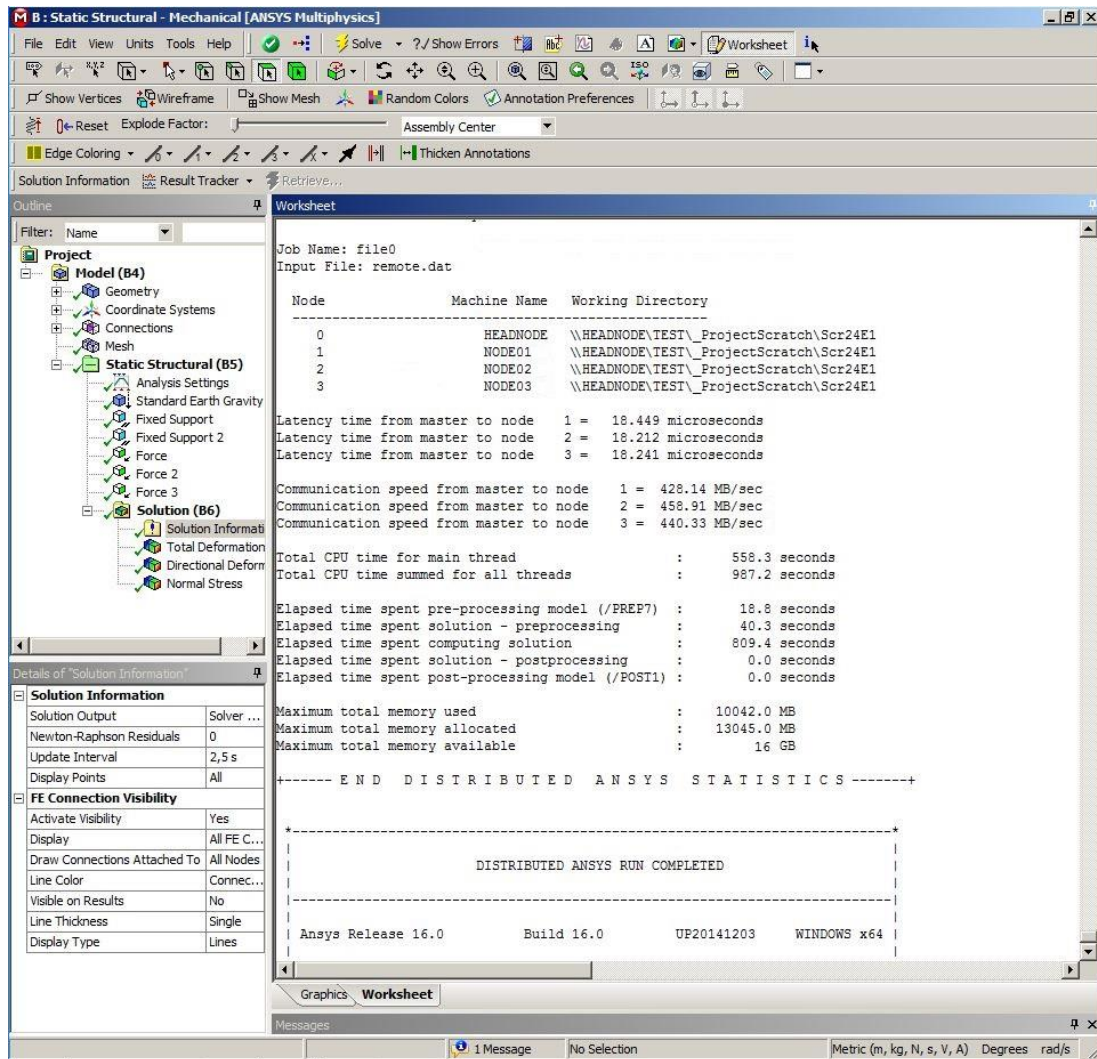


Fonte: Autoria própria

Este cálculo dura bastante tempo sendo executado em apenas um *hardware*, e dependendo dessa configuração, é comum demorar dias para executar um planejamento mais complexo, com a distribuição do cálculo, foi possível reduzir o tempo de cada teste dentro de nosso *cluster* em aproximadamente 70% utilizando 8 computadores interligados.

O software retorna os valores computacionais em tempo de respostas e velocidades de transferência entre os nós e o computador principal. A imagem a seguir apresenta essa tela de resultado.

Figura 33: Tela de resultado de tempo utilizado durante o cálculo computacional



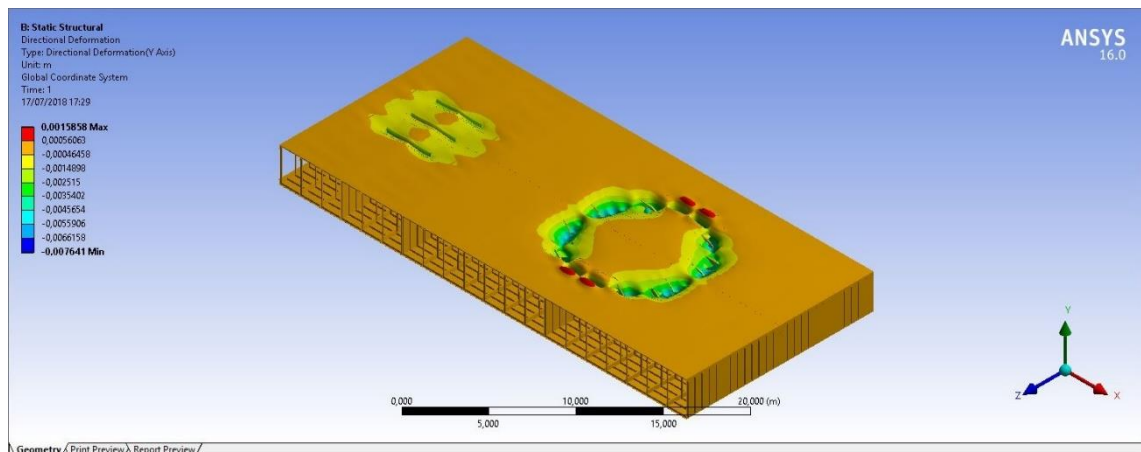
Fonte: Autoria própria

O desempenho foi medido por três testes similares com complexidades diferentes. O TESTE A representa uma intercessão pequena de um casco de navio cinco metros quadrado sendo de baixa complexidade; o TESTE B representa uma intercessão maior do casco de navio vinte, metros quadrados, sendo de alta complexidade; o TESTE AB, como a nomenclatura sugere, e junção das duas partes, mas o ponto de intercessão entre as placas, sendo de extrema complexidade.

Os testes foram executados em cinco interações de cada etapa de n nós adicionado ao *cluster* (sendo n a fórmula de 2^n , $n > 1$, tendo como resultado de 2, 4, 8, 16, 32, ..., nós adicionados ao *cluster*, mas nos mantivemos no máximo de 8 nós ao *cluster*, devido aos resultados o problema ocorre por causa do gargalo na rede, o fluxo e muito grande de dados com isso o tempo de resposta acaba sendo cada vez maior, foi retirado uma média arredondada dos valores.

O resultado retorna os valores de deformação da estrutura, direção e tensão mecânica dos materiais que foram utilizados na elaboração do projeto, podendo alterar o tipo de material, sua quantidade, antes de ir para a construção final.

Figura 34: Resultado apresentando a deformação da estrutura.



Fonte: Autoria própria

Nos testes, obtivemos os resultados a seguir:

Tabela 4: Comparativo Tempo de execução X Nós no *cluster*.

	1PC	2PC	4PC	8PC
TESTE A	50 min	28 min	18 min	14 min
TESTE B	429 min	247 min	161 min	128 min
TESTE AB	615 min	387 min	258 min	213 min

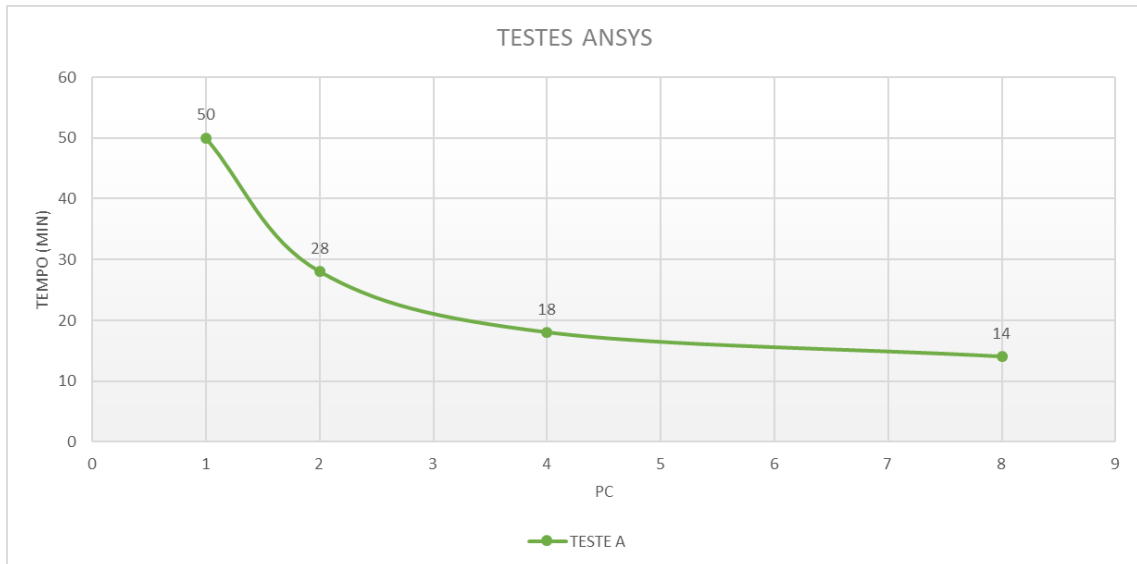
Fonte: Autoria própria

Percebe-se que ao incluir um segundo computador para distribuir a computação, o percentual de tempo reduzido é muito maior do que ao incluir mais

nós ao *cluster*, pois o tempo da comunicação pela rede também se torna bastante complexo e demanda um maior tempo para transmitir as informações.

A seguir os gráficos de relação tempo em minutos versus o número de nós do *cluster*.

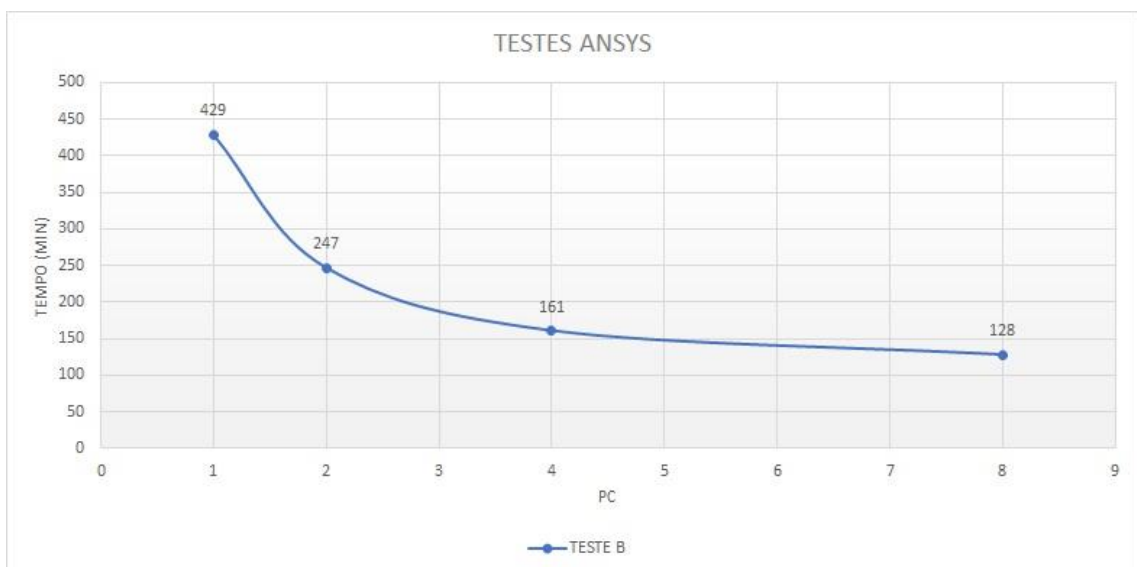
Figura 35: Relação Tempo x PCs do *cluster* TESTE A



Fonte: Autoria própria

Para o TESTE A, percebe-se um significativo ganho de desempenho, pois o teste, apesar de ainda complexo, requer bem menos cálculos computacionais em comparação com os demais.

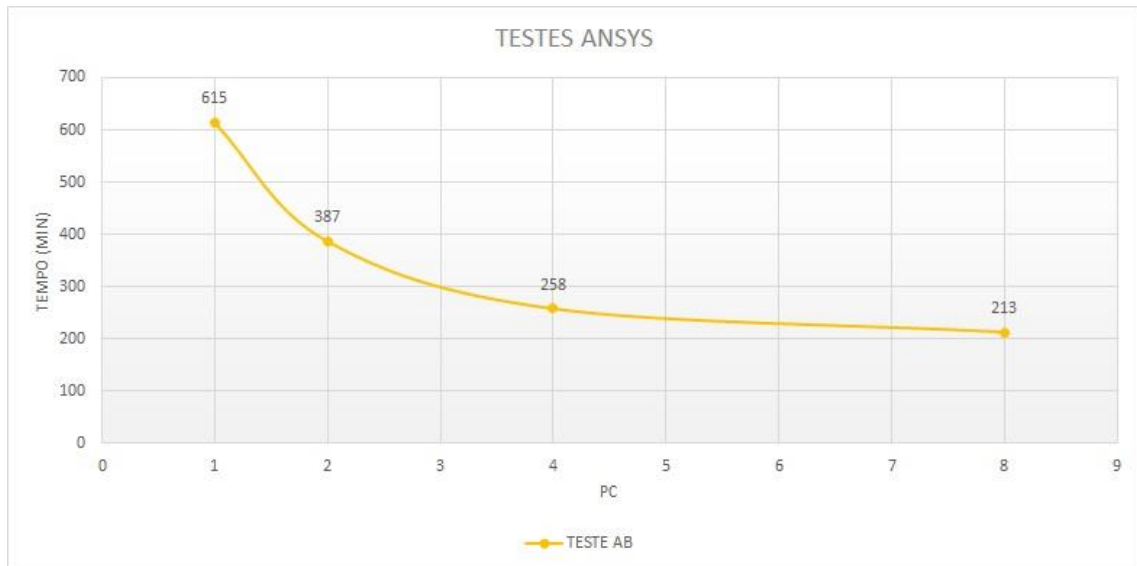
Figura 36: Relação Tempo x PCs TESTE B



Fonte: Autoria própria

Para o TESTE B, percebe-se um menor ganho de desempenho a partir do 4 nó computacional, pois como é um teste bem mais complexo que o teste A, o volume de dados e cálculos é muito maior.

Figura 37: Relação Tempo x PCs TESTE AB

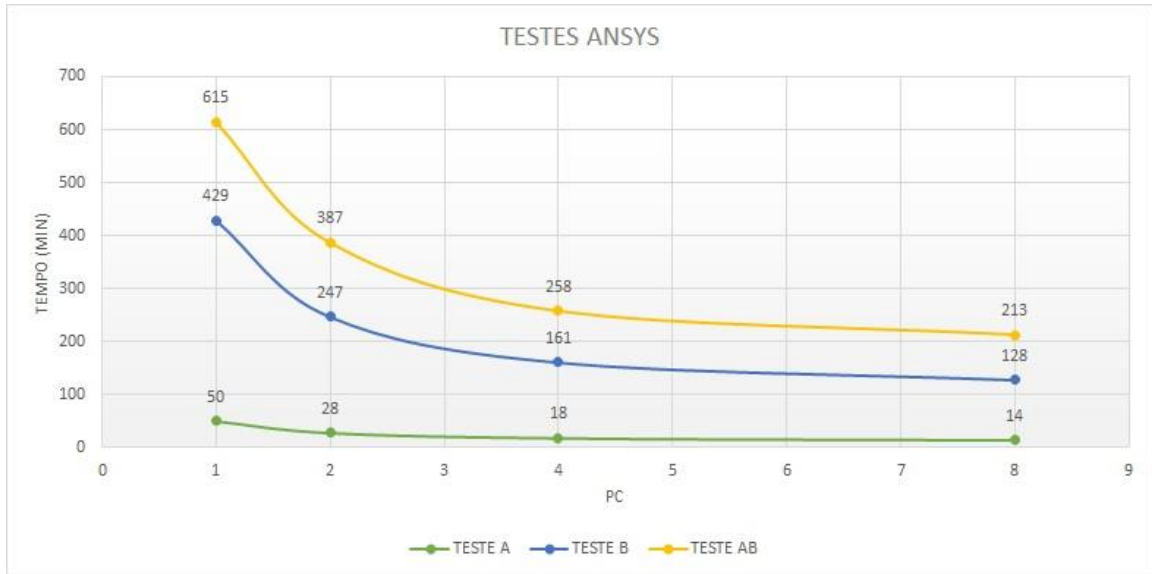


Fonte: Autoria própria

O TESTE AB, apresenta uma complexidade muito maior aos dois primeiros, pois ao juntar os dois cálculos em um só, aumenta-se bem mais a complexidade.

O gráfico a seguir apresenta a junção dos três testes no mesmo gráfico, nota-se o pouco ganho de performance a partir do quarto nó computacional nos três casos.

Figura 38: Relação TEMPO x TESTES



Fonte: Autoria própria

A percentagem do ganho de desempenho, variou de acordo com a dificuldade dos testes executados, sendo demonstrada na tabela a seguir:

Tabela 5: Comparativo da percentagem de ganho de desempenho a incluir mais nós computacionais aos testes.

	2PC	4PC	8PC
TESTE A	44%	20%	12%
TESTE B	42%	20%	8%
TESTE AB	37%	21%	7%

Fonte: Autoria própria

Nota-se a significativa perda de desempenho ao incluir mais nós ao cluster, pois o volume de dados se intensifica ao distribuir mais a aplicação, sendo necessário investir em uma melhor infraestrutura de rede, para que não ocasione esse gargalo, vale ressaltar também que, o TESTE A é bem mais simples que os demais testes, sendo assim, de pequena quantidade de dados sendo transmitido na rede.

5 CONCLUSÃO

Neste trabalho foi realizado um estudo de comparação de ganho de desempenho em arquiteturas paralelas, com foco na plataforma *Windows HPC Server*, de *cluster* de estações de trabalho.

Nos três testes foi obtido um aumento de desempenho à medida que se aumentava o número de nós de processamento, sendo o maior ganho, ao se acrescentar o segundo nó, pois à medida que vai se incluindo mais nós, a quantidade de interações na rede tornou-se complexa demais, e demandava um tempo maior de execução, mas ainda assim havia ganho no tempo de desempenho.

Dos 24 computadores disponíveis no Laboratório de Informática de Engenharia Naval, utilizamos apenas 8 para os nossos testes. Pois ao incluir mais nós ao *cluster*, não houve um significativo ganho de desempenho, devido a um gargalo na rede com um fluxo muito grande de dados, sendo assim a partir do oitavo nó, o cluster acaba não sendo mais vantajoso.

Por se tratar de um trabalho inicial, houveram algumas falhas nas execuções dos testes, e com a falta de conhecimento abrangente sobre os *softwares* de Engenharia Naval disponíveis, se tornaram ainda mais difíceis, já que era necessário ter alguém que dominava a ferramenta para auxiliar, para o futuro, aconselha-se a utilização de minis cluster pode se desenvolver mais de um cluster com até quatro computadores onde o ganho de desempenho é grande, uma melhor estruturação da rede para evitar o gargalo e melhorar o fluxo de dados, mudança da estrutura do *cluster*, utilizando de sistemas operacionais livres (Linux), e um estudo mais amplo para incluir placas gráficas para um maior processamento.

REFERÊNCIAS

ANSYS Inc. Disponível em <https://www.ansys.com/>. Acesso em: 25 out. 2016.

BARCELLAR, Hilário Viana. **Cluster: Computação de Alto Desempenho**. Disponível em: << <http://www.ic.unicamp.br/~ducatte/mo401/1s2010/T2/107077-t2.pdf>>>. Acesso em: 10 out. 2018.

BARCELLOS, M. P.; GASPARY, L. P. Tecnologias de Rede para Processamento de Alto Desempenho. **Anais** da 3ª Escola Regional de Alto Desempenho (ERAD). Santa Maria, RS: UFRGS, 2006.

CARVALHO, F. H. **Computação de Alto Desempenho em Plataforma Windows**. 2007. (Curso de curta duração ministrado). Departamento de Computação – Universidade Federal do Ceará (UFC) Campus Universitário do Pici.

COLOMBET, L; DESBAT, L. Speedup and efficiency of large size applications on heterogeneous networks. **Theoretical Computer Science**, 196: 31-44, 1998.

JAQUIE, Kalinka Regina Lucas. **Extensão da Ferramenta de Apoio à Programação Paralela (F.A.P.P.) para ambientes paralelos virtuais**. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde08022001-095456/>>. Acesso em: 5 set. 2018.

KIRNER, C., Arquiteturas de sistemas avançados de computação, **Anais...** da Jornada EPUSP/IEEE em Sistemas de Computação de Alto Desempenho, pp. 307-353, 2001.

KOPP, E. M. **Construção de um cluster HPC para simulações de CFD**. 2012. 60p. Monografia (Especialização em Teleinformática e redes de computadores) – Programa de Pós-Graduação em Tecnologia, Universidade Tecnológica Federal do Paraná. Curitiba, 2012.

MACHADO, Francis Berenger. MAIA, Luiz Paulo. **Arquitetura de Sistemas Operacionais**. 4ª ed. Editora LTC, São Paulo, 2007.

MAZZA, M., et al. A dynamical model of fast cortical reorganization. **Journal of Computational Neuroscience**, 16:177-201, 2004

OLIVEIRA, V. **Computação Genérica de Alto Desempenho com GPUs**, MAP-i Doctoral Program in Computer Science, 2007.

PADRO, Claudio Lima do; SILVA, João Messias Alves. **Aplicação de cluster beowulf em instituições de ensino**. 2010. 119 f. Monografia (Curso Superior de Tecnologia em Informática, ênfase Redes de Computadores) - Faculdade de Tecnologia de Guaratinguetá, São Paulo, 2010.

PINHO, M., MAZZA, M.; ROQUE A. C., A computational model of the primary auditory cortex exhibiting plasticity in the frequency representation. **Neurocomputing**, 70: 3-8, 2006.

PITANGA, M. **Construindo Supercomputadores com Linux**. Rio de Janeiro: Brasport Livros e Multimídia Ltda, 2004.

PRADO, C. L.; SILVA, J. M. A. **Aplicação de Cluster Beowulf em Instituições de Ensino**. Guaratinguetá, 2010. 117p. Monografia, Faculdade de Tecnologia de Guaratinguetá.

RUSSEL, C. **Visão Geral do Microsoft Windows Compute Cluster Server 2003**, *Microsoft Windows Server 2003 Administrator's Companion*, MSPress, 2005.

SCHAEFER, M., Et al. Simulation of spiking neural networks: architectures and implementations. **Neurocomputing**, 48: 647-679, 2002.

SLOAN, J.D. **High Performance Linux Clusters with OSCAR**, Rocks, OpenMosix, and MPI, O'Reilly, 2004.

STERLING, T. **Beowulf Cluster Computing with Linux**, Estados Unidos: Mit Press, 2002.

SOUZA, S. M.; ROQUE A. C., Self-sustained waves in a computational model of the olfactory epithelium with gap junctions. **Neurocomputing**, 58-60: 1033-1039, 2004.

TANENBAUM, A. S. **Sistemas Operacionais Modernos**. 2. ed. São Paulo: Pearson Prentice Hall, 2006.

FOLDING at Home. Disponível em: <<https://foldingathome.org/>>. Acesso em: 10 out. 2016.

TÃO poderoso quanto um supercomputador. [S.l.: s.n.], 2015. Disponível em: <<https://sdufal.wordpress.com/2009/12/14/taopoderosoquantoumsupercomputador/>>. Acesso em: 06 out. 2018.

APÊNDICE

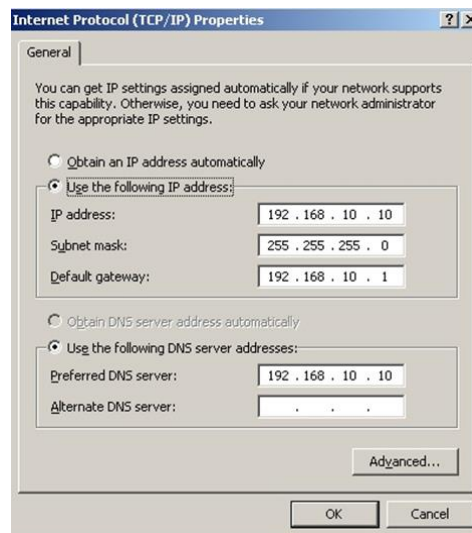
APÊNCIDE A – Especificação de *hardware* do *cluster* laboratório de engenharia naval

- PROCESSADOR: AMD Phenom™ II X4 955 Black Edition 3.4 GHz
Soquete: AM3;
Processo de fabricação: 45nm;
Quantidade de transistores: 758 milhões;
Número de núcleos: 4.
- MEMÓRIA: 2 MARKVISION 8GB DDR3 1600 MHz
8 bancos internos independentes;
8 bitpre-fetch.
- HDD 3,5: SEAGATE
Interface: SATA 6 GB/s;
Cache: 64 MB;
Armazenamento: 1 TB;
Rotação: 7200 RPM.
- PLACA MÃE: ASUS AM3 M4N68T
Formato: MicroATX;
Dimensões: 24,4 cm x 20,8 cm;
Processadores compatíveis: Soquete AM3, AMD Phenom II;
Slots de memória: 2 Slot DDR3 de 240 pinos;
Capacidade máxima: 8 GB (por slot), Dual-Channel.
- NVIDIA nForce 10/100/1000 Mbps *Ethernet*
Integrado a placa mãe, padrão de mercado.

APÊNDICE B – Instalação e configuração de um *cluster* na plataforma *windowsserver hpc*

Para criarmos o domínio precisaremos instalar o *Active Directory* (AD) – serviço de diretórios, que é um conjunto de atributos sobre recursos e serviços existentes na rede, uma maneira mais fácil para simplificar o acesso aos recursos da rede centralizando-os num único computador, que será dominado Servidor de Domínio. O AD é criado quando estamos instalando o Domínio, pois não é possível criar um domínio antes e depois instalar o AD, esse processo ocorre junto.

Configuração de IP para domínio.



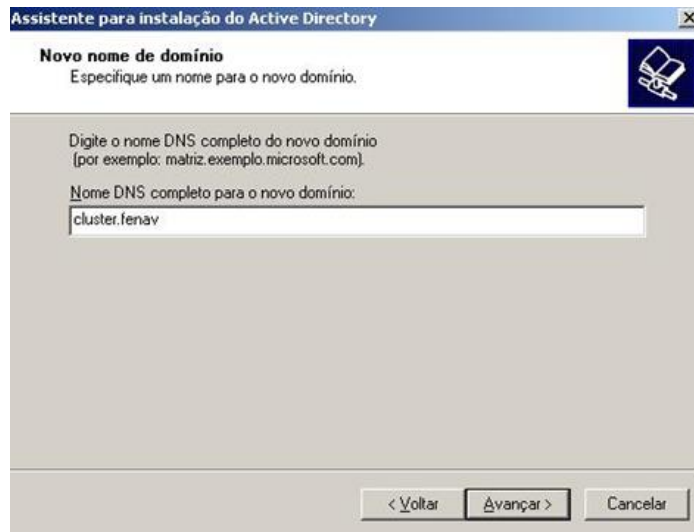
Utilizando do *WindowsServer* 2003 x64 bits para implementação do servidor de domínio, devemos forçar um IP, como na Figura A-1, deve-se definir também o servidor DNS - *Domain Name System* - Sistema de Nomes de Domínios, é um sistema de gerenciamento de nomes hierárquicos e distribuído para computadores. Indica-se como DNS o IP escolhido para o servidor de domínio.

Tela inicial do Assistente para instalação do Active Directory, o servidor de domínio.



Para que os participantes do domínio acessem à internet e outras configurações através do servidor de domínio. Após isso, configurar o *Active Directory* através do comando no menu Iniciar > Run > dcpromo.exe, onde se abrirá a janela com a configuração do Assistente para Instalação do Active Directory, durante o assistente, deve-se especificar como será configurado o domínio, no nosso caso, iremos adicionar um controlador de domínio para um novo domínio e criaremos uma nova floresta para esse domínio criado, uma floresta, são os nós computacionais que pertencem a esse domínio, como os computadores de uma empresa, ou um *cluster*, escolheremos um nome a ser definido para o domínio que no nosso caso será *cluster.fenav*, avançando será pedido para definir o nome NetBIOS, que serve para o reconhecimento do domínio na rede, por padrão é deixado o primeiro nome do domínio escolhido. *CLUSTER*.

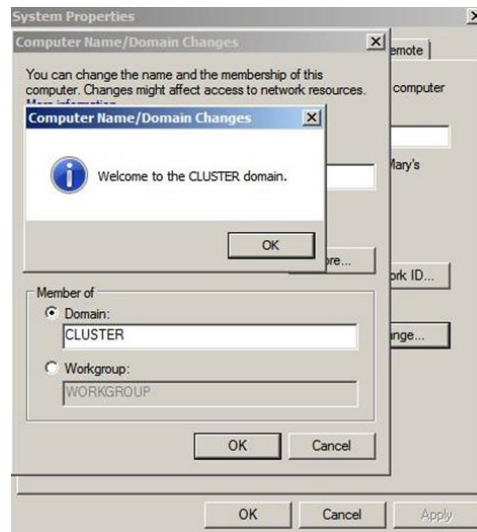
Nome definido para o servidor de domínio.



Para criar o controlador do *cluster*, o headnode, usaremos uma máquina com o *Windows HPC Server 2008 R2* de 64 bits, como todo o restante dos nós, apresenta versão de 64 bits. O computador principal de onde partirá todos as tarefas para o *cluster*, precisamos vinculá-lo ao domínio criado anteriormente, configurando a faixa de IP como a do servidor de domínio, e definindo a opção *Preferred DNS Server* igual ao IP do servidor de domínio, a imagem a seguir apresenta essa configuração.

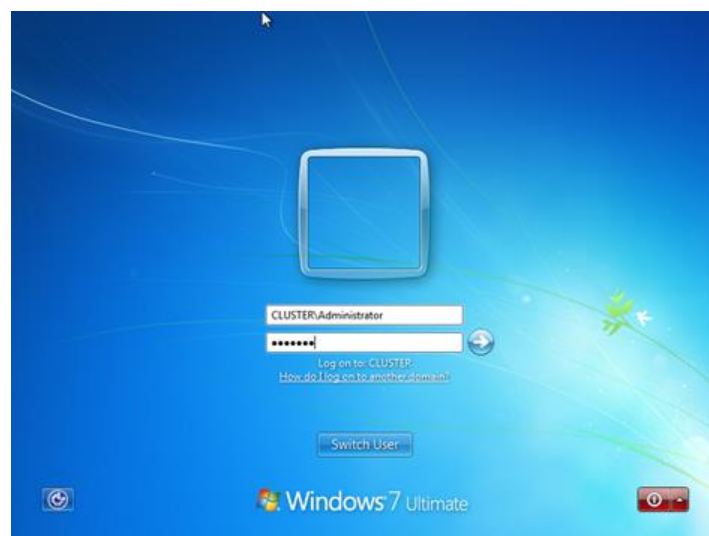
Em seguida, deve-se adentrar o domínio, configurando através da janela *Computer Name/Domain Changes*, digitando em *Member of* o domínio *CLUSTER* criado a pouco, aproveitando para renomear o computador para seguir o padrão que será adotado para diferenciação dos nomes da rede. Será pedido para digitar usuário e senha para ingressar no domínio, o mesmo que foi configurado a pouco, se tudo estiver correto, entraremos no domínio.

Tela de ingresso ao servidor de domínio.



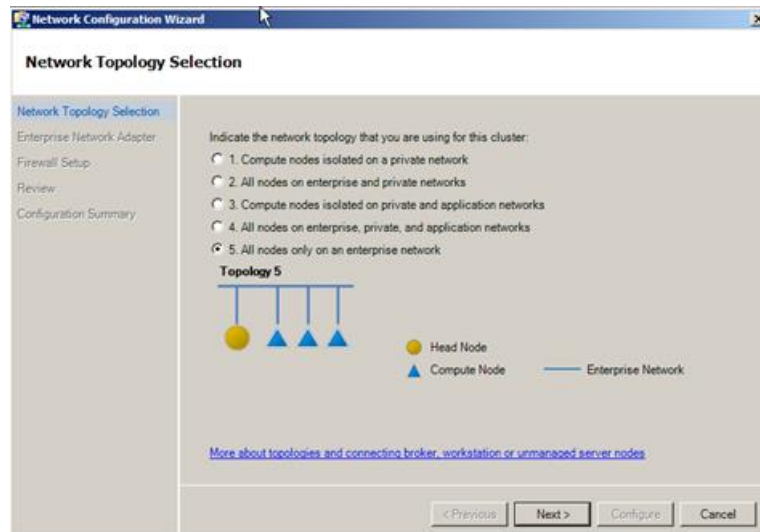
Para que as configurações entre em vigor será preciso reiniciar o *Windows*, e ao retornar, é importante que se faça o login com o usuário do Servidor de Domínio. Para continuar a instalação do *cluster*, deve-se realizar o download do pacote *HPC Pack Express 2008 R2 with SP4* e instalar o *HPC 2008 R2 Cluster Manager*, o programa apresenta uma interface onde pode-se enviar os trabalhos para o *cluster*.

Entrada no sistema como usuário pertencente ao servidor de domínio.



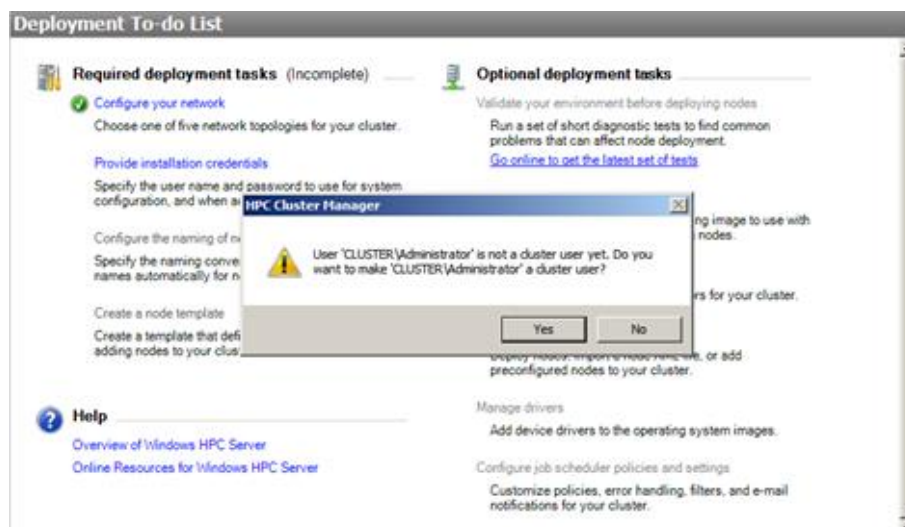
Por ser o headnode, durante a instalação selecionaremos a opção *Create a new HPC Cluster* e seguir as opções padrões do instalador.

Tela de configuração de topologia de rede.



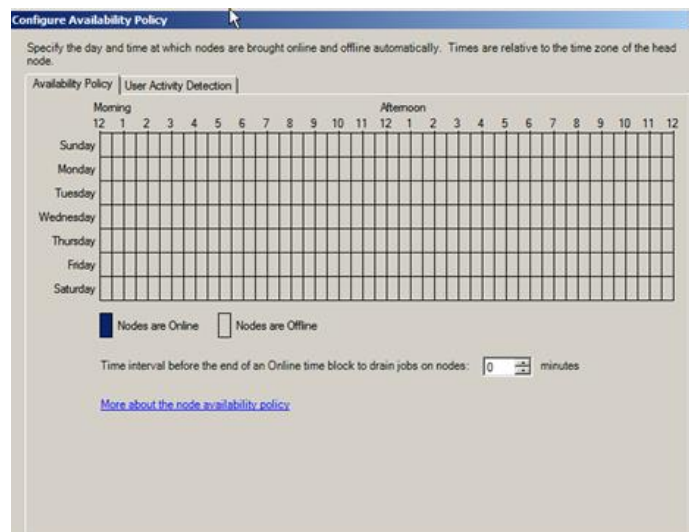
A primeira tela do programa, nos dá opção de configurar a topologia de rede, para o nosso uso, é escolhido a topologia onde todos os nós computacionais pertencem a uma rede corporativa, pois é o mesmo modelo que o laboratório está distribuído, dentro da interface, deve-se conceder o acesso ao servidor de domínio clicando em *Provide installation credential*, abrirá a janela para informar o usuário e senha, deve-se fornecer o login e senha do servidor de domínio, após isso será apresentado a tela confirmando que o usuário está se tornando um usuário de *cluster*.

Tela de confirmação de credenciais do domínio.



Em *Choose Node Template Style*, indicaremos quais nós computacionais serão adicionados ao *headnode*, no nosso padrão escolhido, serão workstation, - estações de trabalho comuns dos usuários do laboratório, será preciso definir qual modelo de utilização desses nós será utilizado escolhendo em *Availability Policy*, - Política de Disponibilidade - , pode-se definir para os nós ingressarem ao *cluster* quando estiverem com pouco uso de CPU, se tem ou não teclado e mouse conectado ao computador, ou em determinados horários dos dias, ou adicioná-los manualmente, podendo começar um *job* no *cluster* durante a semana e os nós serão adicionados ou excluídos do *cluster* de acordo com o horário estabelecido, aumentando o ganho de performance automaticamente, sem precisar estar presente ou interromper o uso dos alunos durante os horários de funcionamento do laboratório de informática.

Tela de configuração da Política de Disponibilidade.



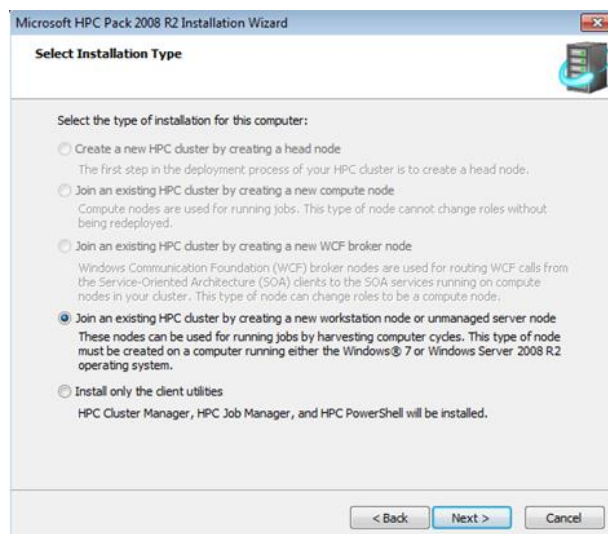
Tendo definido todas as opções, será pedido para informar um nome para o *Node Template*, - Modelo de Nó, que será exportado para os futuros nós computacionais que ingressarão no *cluster*. Essa configuração é importante pois, através dela será possível definir o modelo que será implementando para o *cluster*, sendo necessário apenas importar na configuração posteriormente.

Tela de login nos computadores do laboratório.



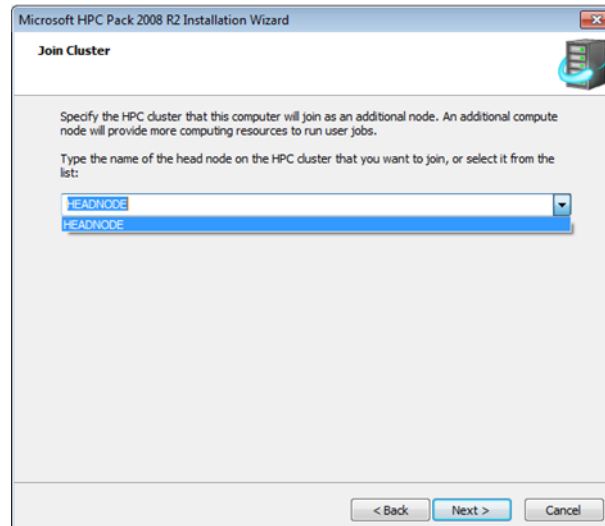
O próximo passo é incluir os nós workstation que definimos anteriormente ao nó principal, para isso devemos incluir no domínio criado os computadores que serão utilizados no *cluster*, sendo todos instalados e configurados com o sistema operacional *Windows 7 Ultimate Edition* de 64 bits, o processo para adicionar ao domínio é exatamente o mesmo feito anteriormente, só mudando o final do IP em cada máquina, mas agora teremos duas contas no sistema, uma para quando estivermos utilizando o *cluster*, e outra para uso do laboratório de informática.

Tela de configuração de entrada ao cluster criado anteriormente.



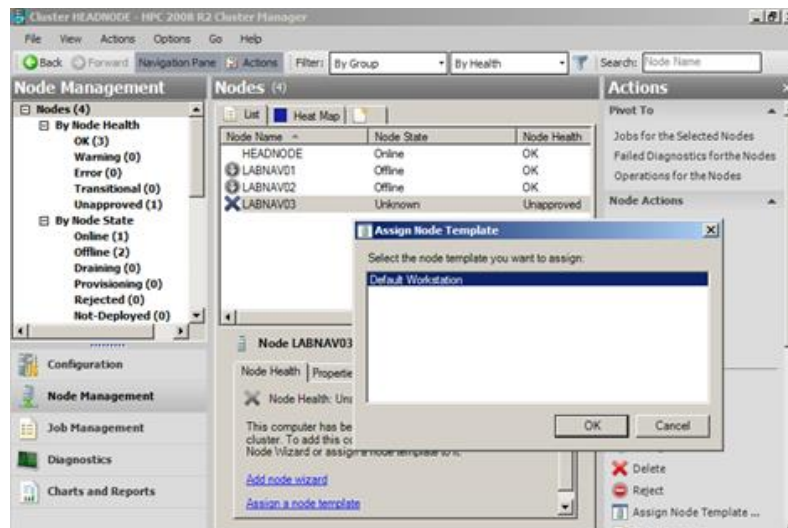
Entrando na conta vinculada ao domínio, instalaremos o mesmo *HPC 2008 R2 Cluster Manager* através do *HPC Pack Express 2008 R2 with SP4*, e adicionaremos os nós workstation ao *cluster* escolhendo a opção *Join an existing HPC cluster*, deverá aparecer o nome do headnode criado anteriormente.

Selecionando o nome do nó gerenciador do cluster.



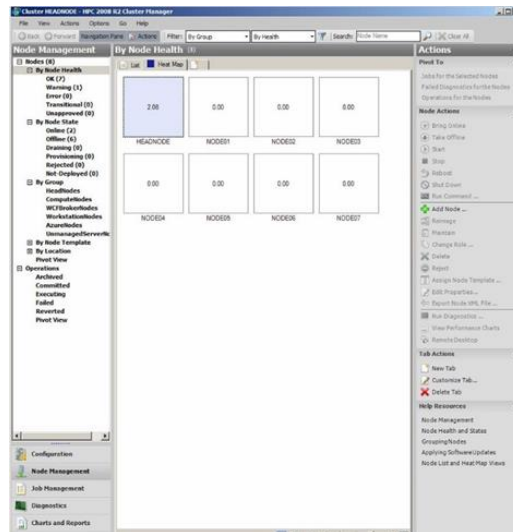
Após o processo de instalação, poderá configurar o *cluster* através da janela *Node Management* do *HPC 2008 R2 Cluster Manager*, deverá aparecer os nodes que adicionamos ao *cluster*, juntamente com o *headnode*, porém, ainda falta atribuir o modelo de *workstation* que criamos previamente, feito isso, todas as configurações estarão prontas, precisando apenas repetir o processo a cada nó inserido ao *cluster*.

Definindo o template previamente configurado para o novo nó adicionado ao cluster.



O processo de instalação se repetirá para cada computador que irá ingressar no *cluster*, a cada um, deve-se confirmar a integridade da rede, estando tudo OK, nosso *cluster* estará pronto para uso.

Visão geral do cluster com oito nós devidamente configurados.



Repetimos a configuração com 8 computadores disponíveis do laboratório para testes, podendo ser incluídos todos os 24 PCs do laboratório posteriormente.