

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**SISTEMA WEB DE GERENCIAMENTO DE *CLUSTER*  
*BEOWULF* PARA COMPUTAÇÃO FLUIDODINÂMICA**

**JAVIER OMAR AUGUSTO**

**BLUMENAU, JUNHO**  
**2007**

**2007/I-10**

**JAVIER OMAR AUGUSTO**

**SISTEMA WEB DE GERENCIAMENTO DE *CLUSTER*  
*BEOWULF* PARA COMPUTAÇÃO FLUIDODINÂMICA**

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Sistemas de Informação – Bacharelado.

Prof. Paulo Fernando da Silva – Orientador

**BLUMENAU, JUNHO  
2007**

**2007/I-10**

# **SISTEMA WEB DE GERENCIAMENTO DE *CLUSTER* *BEOWULF* PARA COMPUTAÇÃO FLUIDODINÂMICA**

Por

**JAVIER OMAR AUGUSTO**

Trabalho aprovado para obtenção dos créditos  
na disciplina de Trabalho de Conclusão de  
Curso II, pela banca examinadora formada por:

---

Presidente: Prof. Paulo Fernando da Silva – Orientador, FURB

---

Membro: Prof. Antônio Carlos Tavares – FURB

---

Membro: Prof. Miguel Alexandre Wisintainer – FURB

**BLUMENAU, JUNHO  
2007**

Dedico este trabalho aos meus pais  
Ricardo e Silvia Margarita Augusto.  
A Deus, por tudo.

What do I think of Western civilization? I think it would be a very good idea.

Mohandas Karamchand Gandhi.

## AGRADECIMENTOS

Gostaria de agradecer a todos que de uma forma ou outra contribuíram com esse trabalho.

Agradeço a Deus, pelo dom da feliz vida que ele me presenteou, além de permitir que eu tenha pessoas imensamente importantes ao meu lado.

Não sendo o bastante dedicar, preciso agradecer a minha família pelo amor, paciência e enorme ajuda. Sempre estiveram presentes em todos os meus momentos.

A Leide, por iluminar a caminhada de minha vida e por ter me dado todo o apoio e amor imensurável nesses últimos anos.

A minha segunda família, “EXN Crew” que desde cada canto desse mundo me brindaram apoio incondicional e acreditaram em mim.

Aos amigos Jean Ricardo Otto, Marlos Sedrez, Fábio Bruns, Jionei Dolzan, Israel Campos, o “Cata”, Eduardo Krieck e Adriano “BBB..” Rezena, pela imensa amizade e principalmente pela companhia.

Ao Prof. Paulo Fernando, por ter acreditado na conclusão deste trabalho.

À família Filippi-Mendonça, pelo amor, companhia, e imensa ajuda.

À equipe do Laboratório de Fluidodinâmica Computacional do Prof. Dr. Henry França Meier, Dirceu Noriler e Vinicyus Wiggers, por terem acreditado em mim, pela ajuda com o ANSYS CFX, CFD e  $\LaTeX$ , a paciência, amizade e fraterno apoio, sem eles este trabalho não teria acontecido.

A Sir Clive Sinclair, o culpado de tudo, pois seria injustiça não mencioná-lo.

## RESUMO

Este trabalho apresenta a especificação, desenvolvimento e implantação de um sistema orientado à *web* para gerenciamento, submissão e simulação de aplicações baseadas no *software* de computação fluidodinâmica ANSYS CFX, executadas sobre um *cluster beowulf*. A solução possibilita simular, de maneira eficiente, os domínios da fluidodinâmica de médio e grande porte utilizados pelo Laboratório de Fluidodinâmica Computacional da FURB. Os resultados obtidos nesse trabalho mostram a importância e eficiência da computação paralela utilizando *clusters beowulf* para o processamento e resolução de problemas da fluidodinâmica computacional

**Palavras Chave:** *Clusters Beowulf*. Sistemas distribuídos. Computação paralela. *High performance computing* (HPC). Computação Fluidodinâmica. CFD. ANSYS CFX.

## **ABSTRACT**

The following work presents the specification, developing and deployment of a web oriented system for managing, submission and simulation of ANSYS CFX software computational fluid dynamics based applications. The solution enables the simulation, in an easy way, of computational fluid dynamics middle to big-sized domains used by the Computing Fluid Dynamic Laboratory of FURB. The results gathered in this work shows the importance and efficiency of parallel computing using beowulf clusters for process and resolution of computational fluid dynamics problems.

**Key-Words:** Beowulf Clusters. Distributed Systems. Parallel Computing. High performance computing (HPC). Computational Fluid Dynamic. CFD. ANSYS CFX.



## LISTA DE ILUSTRAÇÕES

Figura 2.1 – Exemplo de topologia de <i>cluster</i> para processamento paralelo . . . . .	19
Figura 2.2 – Cluster NASA: Um <i>cluster</i> feito com 64 PCs . . . . .	21
Figura 2.3 – Solução de Alta-Disponibilidade ou <i>High Availability Clustering</i> . . . . .	22
Figura 2.4 – Solução de <i>Load-Balance</i> da Oracle para servidores Linux usando o <i>Global Single Instance</i> . . . . .	24
Figura 2.5 – Resultados de vários modelos de CFD obtidos com o CFX. . . . .	29
Figura 2.6 – Arquitetura do ANSYS CFX. . . . .	30
Figura 2.7 – Exemplo do resultado de uma simulação típica de modelos automotivos. . . . .	31
Quadro 3.1 – Lista de programas a serem instalados no master e nos nós através do comando “ <i>apt-get</i> ” . . . . .	35
Quadro 3.2 – Conteúdo do arquivo <i>/etc/hosts</i> . . . . .	36
Quadro 3.3 – Chave compartilhada para acesso seguro usando o OpenSSH . . . . .	37
Quadro 3.4 – Lista de programas a serem instalados somente no <i>master</i> . . . . .	39
Quadro 3.5 – Conteúdo do arquivo <i>/etc/apache2/sites-enabled/cluster-cfd</i> . . . . .	40
Quadro 3.6 – Conteúdo do arquivo <i>/etc/apache2/sites-enabled/000-default</i> . . . . .	41
Quadro 3.7 – Requisitos Funcionais . . . . .	42
Quadro 3.8 – Requisitos Não Funcionais . . . . .	43
Figura 3.1 – Visão geral do sistema . . . . .	43
Figura 3.2 – Diagrama de Casos de Uso para Usuário . . . . .	44
Figura 3.3 – Diagrama de Casos de Uso para Administrador . . . . .	46
Figura 3.4 – Diagrama MER . . . . .	47
Figura 3.5 – Diagrama de Atividade do processo de submissão de <i>job</i> via <i>web</i> . . . . .	49
Figura 3.6 – Diagrama de Atividade do processo de agendamento via <i>web</i> . . . . .	49
Figura 3.7 – Funcionamento do CGI . . . . .	51
Figura 3.8 – Código exemplo demonstrando o uso das funções HTML em conjunto com o Perl . . . . .	54
Figura 3.9 – Rotina de consulta da senha criptografada no banco . . . . .	55

Figura 3.10 – Código fonte do <i>script</i> Perl <i>rodar_job.pl</i> . . . . .	56
Quadro 3.9 – Entrada “ <i>cron</i> ” gerada pelo processo de agendamento do sistema . . . . .	56
Figura 3.11 – Código fonte do <i>script</i> BASH para atualizar os gráficos do estado do CPU . . . . .	57
Figura 3.12 – Simulação “teste” a ser executada dentro do <i>cluster beowulf</i> . . . . .	58
Figura 3.13 – Tela de acesso ao sistema . . . . .	58
Figura 3.14 – Tela principal do sistema <i>web</i> . . . . .	59
Figura 3.15 – Tela de alteração de dados pessoais, senha e <i>e-mail</i> . . . . .	60
Figura 3.16 – Tela de submissão de <i>jobs</i> . . . . .	60
Figura 3.17 – Tela de alocação de recursos . . . . .	61
Figura 3.18 – Tela do estado de processamento do <i>cluster</i> . . . . .	62
Figura 3.19 – Acompanhamento da evolução da solução a partir dos resultados obtidos no <i>cluster</i> . . . . .	63

## LISTA DE ALGORITMOS

3.1 Modificação do <i>script</i> de inicialização do PVM . . . . .	35
3.2 Alterações nos arquivos <i>/etc/profile</i> e <i>.bashrc</i> . . . . .	38

## LISTA DE TABELAS

Tabela 3.1 – Tabela de particionamento dos discos no <i>cluster</i> . . . . .	34
Tabela 3.2 – Tabela de tempo de simulação obtidos no <i>cluster</i> . . . . .	63

## LISTA DE SIGLAS

**API** *Application Programming Interface*

**CESDIS** *Center of Excellence in Space Data and Information Sciences*

**CFD** *Computational Fluid Dynamics*

**CGI** *Common Gateway Interface*

**CPU** *Central Processing Unit*

**DHCP** *Dynamic Host Configuration Protocol*

**DNS** *Domain Name Service*

**FHS** *Filesystem Hierarchy Standard*

**FURB** *Fundação Universidade Regional de Blumenau*

**HA** *High Availability*

**IP** *Internet Protocol*

**LFC** *Laboratório de Fluidodinâmica Computacional*

**LDPS** *Laboratório de Desenvolvimento de Processos de Separação*

**MP** *Message Passing*

**MPI** *Message Passing Interface*

**MPICH** *Message Passing Interface Portable Implementation*

**MPP** *MultiProcessing Parallel Processor*

**NIS** *Network Information Service*

**NFS** *Network File System*

**NUMA** *Non-Uniform Memory Access*

**OO** *Orientação a Objetos*

**PBS** *Portable Batch System*

**PVM** *Parallel Virtual Machine*

**RAID** *Redundant Array of Inexpensive Disks*

**SSH** *Secure Shell*

**SMP** *Symmetric Multi-Processing*

**SO** *Sistema Operacional*

**SQL** *Structured Query Language*

**TCP** *Transmission Control Protocol*

**UDP** *User Datagram Protocol*

**UML** *Unified Modeling Language*

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	ESCOPO E PROBLEMA . . . . .	16
1.2	OBJETIVOS DO TRABALHO . . . . .	16
1.3	ESTRUTURA DO TRABALHO . . . . .	17
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>18</b>
2.1	CLUSTERS . . . . .	18
2.1.1	CLUSTER BEOWULF . . . . .	19
2.1.2	ALTA DISPONIBILIDADE ( <i>High Availability</i> ) . . . . .	22
2.1.3	BALANCEAMENTO DE CARGA ( <i>Load Balancing</i> ) . . . . .	23
2.2	PROCESSAMENTO EM PARALELO . . . . .	24
2.2.1	PARALLEL VIRTUAL MACHINE (PVM) . . . . .	25
2.2.2	MESSAGE PASSING INTERFACE (MPI) . . . . .	27
2.3	FLUIDODINÂMICA COMPUTACIONAL . . . . .	27
2.4	ANSYS CFX . . . . .	29
2.5	TRABALHOS CORRELATOS . . . . .	32
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>33</b>
3.1	INSTALAÇÃO E CONFIGURAÇÃO . . . . .	33
3.1.1	Instalação do SO, principais bibliotecas e aplicativos . . . . .	34
3.1.2	Relacionamento de confiança entre as máquinas através de chaves RSA compartilhadas; . . . . .	36
3.1.3	Instalação do ANSYS CFX-10 . . . . .	37
3.1.4	Instalação dos serviços necessários para executar o sistema <i>web</i> no <i>master</i> . . . . .	39
3.2	REQUISITOS PRINCIPAIS . . . . .	41
3.3	VISÃO GERAL DA SOLUÇÃO PROPOSTA . . . . .	42
3.4	ESPECIFICAÇÃO . . . . .	44

3.4.1	Casos de Uso . . . . .	44
3.4.2	Modelo Conceitual . . . . .	47
3.4.3	Diagrama de Atividades . . . . .	48
3.5	IMPLEMENTAÇÃO . . . . .	50
3.5.1	Técnicas e ferramentas utilizadas . . . . .	50
3.5.2	Operacionalidade da implementação . . . . .	53
3.6	UTILIZAÇÃO DO SISTEMA . . . . .	57
3.7	RESULTADOS E DISCUSSÃO . . . . .	62
<b>4</b>	<b>CONCLUSÃO</b>	<b>65</b>
4.1	LIMITAÇÕES DO SISTEMA . . . . .	66
4.2	TRABALHOS FUTUROS . . . . .	66
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>67</b>
	<b>APÊNDICE</b>	<b>70</b>
	APÊNDICE A - SCRIPT PARA CRIAÇÃO DAS TABELAS NO MYSQL . . . . .	70
	APÊNDICE B - SCRIPT PARA CRIAÇÃO DO USUÁRIO ADMIN NO BANCO MYSQL	71
	<b>ANEXO</b>	<b>72</b>
	ANEXO A - LICENÇAS DE DISTRIBUIÇÃO DE SOFTWARE COM CÓDIGO FONTE ABERTO . . . . .	72

# 1 INTRODUÇÃO

Nos últimos 40 anos, a modelagem e simulação dos fenômenos associados à dinâmica dos fluidos ganharam um grande impulso com o avanço da computação, o desenvolvimento dos recursos computacionais em *software* e *hardware* e sua capacidade computacional de processamento. A modelagem numérica, conhecida como *Computational Fluid Dynamics* (CFD) ou em português fluidodinâmica computacional, tem sido largamente empregada, tendo em vista a facilidade de aplicação e o baixo custo. O sucesso da fluidodinâmica computacional reside no fato de resolver numericamente, mas de uma forma rigorosa as equações provenientes dos balanços de massa, momento e energia.

*Clusters* de computadores são geralmente utilizados para se obter alto desempenho na execução de aplicações paralelas. Sua utilização tem aumentado significativamente ao longo dos anos e resulta hoje em uma presença de quase 60% entre as 500 máquinas mais rápidas do mundo (MEURER, 2004). A arquitetura de um *cluster* consiste basicamente em um conjunto de computadores independentes e interconectados que juntos representam um único recurso computacional integrado (BUYA; BAKER, 2001).

Em busca de soluções viáveis, empresas de menor porte e universidades com poucos recursos financeiros, podem recorrer a uma alternativa para obtenção de processamento de alto desempenho a custos razoáveis, aproveitando o hardware existente. Essa alternativa pode ser concretizada através do uso de *clusters* de computadores, que se tornou possível através do desenvolvimento e barateamento da tecnologia (PITANGA, 2004, p.26).

O Laboratório de Fluidodinâmica Computacional (LFC) e o Laboratório de Desenvolvimento de Processos de Separação (LDPS) da Fundação Universidade Regional de Blumenau (FURB) vêm desde 1998 envidando esforços para a consolidação de uma estrutura numérica e experimental, capaz de desenvolver modelos e métodos da fluidodinâmica computacional. O conjunto de atividades dos referidos laboratórios nos últimos 3 anos produziu um alto índice de publicações científicas em congressos e revistas nacionais e internacionais, além de várias dissertações de mestrado e teses de doutorado. Neste período, diversos projetos multi-institucionais e sólidas parcerias Universidade/Indústria, com fomento público e privado, garantiram uma infra-estrutura mínima (MEIER, 2004, p.1).

Para MEIER (2004, p.15), a grande vantagem da solução computacional do tipo *cluster beowulf* é a possibilidade do processamento paralelo em consonância com a computação de alto

desempenho, aplicado aos estudos de verificação e validação em fluidodinâmica computacional (V&V in CFD).

Tornando relevante a utilização da tecnologia de *clusters* como substrato para a aceleração do tempo de processamento e simulação dos projetos feitos dentro do LFC, o presente trabalho tem como objetivo o desenvolvimento de um sistema *web* que facilite o gerenciamento dos trabalhos submetidos no *cluster* a serem processados através do uso do *software* ANSYS CFX, como assim também a própria infra-estrutura operacional de processamento, viabilizando os estudos em fluidodinâmica computacional desenvolvidos pelo LFC do Departamento de Engenharia Química da FURB.

## 1.1 ESCOPO E PROBLEMA

A área de aplicação dos *clusters* é muito diversificada. Em qualquer local onde tiver um grande problema computacional em que o processamento paralelo seja considerado uma vantagem, pode ser indicado para utilização em um *cluster*. São geralmente as aplicações sequenciais consideradas grandes em demasia. Em geral, elas resolvem problemas fundamentais que possuem grandes impactos econômicos ou científicos. São tidas como impossíveis de resolver sem o uso de modernos computadores paralelos, por causa do tamanho de suas necessidades em matéria de tempo de processamento, memória e de comunicação (PITANGA, 2004, p.28).

Conforme SCHNORR (2003, p.17), embora a utilização de *clusters* seja bastante difundida, a tarefa de monitoramento de recursos dessas máquinas é considerada complexa. Essa complexidade advém do fato de existirem diferentes configurações de software e hardware que podem ser caracterizadas como *cluster*.

Desta forma, este trabalho se torna relevante por procurar a utilização da tecnologia de *clusters* como substrato para a aceleração do tempo de processamento e simulação dos projetos feitos dentro do LFC da FURB.

## 1.2 OBJETIVOS DO TRABALHO

O objetivo do presente trabalho é desenvolver um sistema de gerenciamento orientado à *web* para suprir a demanda da gerência de aplicações CFX executadas sobre um *cluster beowulf*, visando facilitar a administração, submissão e resolução dos diversos cálculos fluidodinâmicos realizados no LFC.



Os objetivos específicos do trabalho são:

- a) instalar e configurar uma infra-estrutura de *cluster beowulf*, fazendo uso de 4 PCs rodando Debian GNU/Linux no LFC da FURB, com implementações de troca de mensagens, que viabilizem a solução de problemas numéricos de médio e grande porte;
- b) instalar e configurar o *software CFX* para executar neste *cluster*;
- c) desenvolver uma aplicação *web* que possibilite aos usuários do LFC o acesso ao *software* principal para solução de problemas da dinâmica dos fluídos baseado no conjunto de ferramentas da ANSYS, permitindo o gerenciamento dos trabalhos submetidos, o controle de acesso ao sistema, controle do balanceamento de carga das aplicações do *software CFX*, escalonamento das aplicações entre os nós do *cluster beowulf* e o acompanhamento do desempenho das aplicações neste contexto.

### 1.3 ESTRUTURA DO TRABALHO

Este trabalho está dividido em quatro capítulos que são referidos a seguir:

No primeiro capítulo, é apresentada, objetivamente, uma introdução ao trabalho, suas motivações, seus objetivos e sua estrutura.

No segundo capítulo, é fornecida uma breve explanação sobre alguns fundamentos que servem de base para este trabalho como *clusters*, processamento em paralelo, CFD e o *software* de fluidodinâmica computacional CFX, com foco no seu funcionamento independente da utilização neste trabalho.

No terceiro capítulo, é tratado o desenvolvimento do trabalho, mostrando sua especificação com diagramas de caso de uso, diagramas de classe e de sequência. É também mostrado um caso de uso da aplicação desenvolvida. O quarto capítulo, apresenta as conclusões do trabalho, suas limitações e possíveis extensões para o mesmo.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda, de forma resumida, conceitos técnicos sobre as principais áreas e tecnologias envolvidas no desenvolvimento deste trabalho como *clusters*, processamento em paralelo abordando as bibliotecas PVM e MPI, CFD e o *software* ANSYS CFX, tecnologia base utilizadas pelo sistema *web*.

### 2.1 CLUSTERS

De acordo com PITANGA (2004, p.26), quando se fala em alto desempenho computacional, imagina-se um supercomputador dedicado, com custo elevadíssimo e de difícil operação. Os *clusters* possibilitam a realização de processamento que antes eram destinados unicamente a computadores de alta performance, tendo capacidade de completá-los com o mesmo desempenho. Isso permite que o sistema substitua esses computadores ou que seja apenas acrescentado mais computador ao *cluster* para aumentar a disponibilidade de processamento. As tecnologias de *cluster* possibilitam a solução de diversas tarefas que necessitam grande volume de processamento e disponibilidade às aplicações, podendo ser desde o simples melhoramento no desempenho de bancos de dados até o processamento de pesquisas científicas complexas.

Pode-se garantir que *cluster* é o que se chama de um sistema estruturado com vários computadores, com objetivo de que sejam vistos como um único computador, que trabalham de maneira contígua, para proceder à realização de processamentos pesados, balanceamentos de carga e alta disponibilidade. Em outras palavras, os computadores dividem as tarefas de processamento e trabalham como se fosse um único computador de alta capacidade (KOWAL-TOWSKI, 1995).

Por não haver regras para todos os tipos de *cluster*, estes podem ser definidos segundo o tipo de Sistema Operacional (SO) que utilizam (heterogêneo ou homogêneo), ou do kernel<sup>1</sup> do SO, ou até mesmo conforme os padrões de comunicação agregada (memória compartilhada tipo *Non-Uniform Memory Access* (NUMA)<sup>2</sup> ou troca de mensagens) e também conforme a tecnologia de rede e os protocolos utilizados:

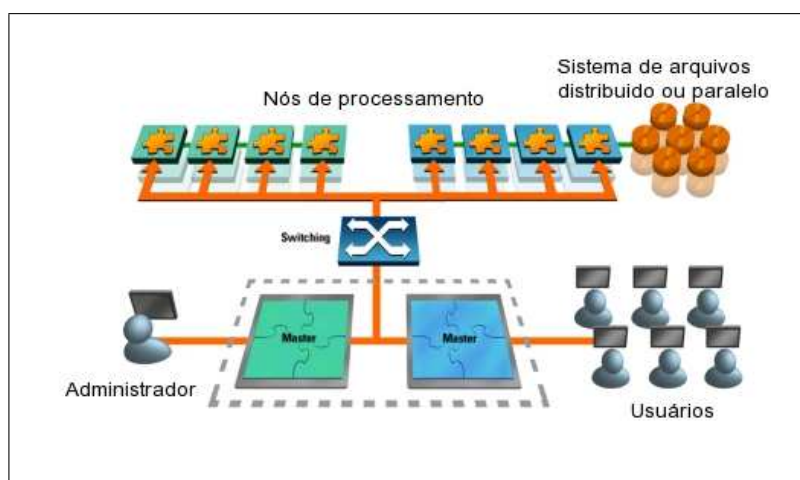
---

<sup>1</sup>Kernel - é o “coração” do SO, representa a camada mais baixa de interface com o *hardware*, sendo responsável por gerenciar os recursos do sistema computacional como um todo.

<sup>2</sup>NUMA - é uma memória de computador projetada para uso em sistemas multiprocessados.

- a) heterogêneo: dois ou mais tipos diferentes de sistemas operacionais formam parte do *cluster*, por exemplo GNU/Linux e Sun Solaris;
- b) homogêneo: somente um tipo de sistema operacional dentro do *cluster*.

Todo computador de um *cluster* pode ser chamado de nó<sup>3</sup> ou nodo. Os computadores devem ser ligados a uma rede, podendo ser formada por qualquer topologia, dependendo da técnica de *cluster* usada. Independente da topologia, o sistema deve ser criado de forma que possibilite a inclusão ou a retirada de um nodo “em situações de problemas”, sem obstruir o funcionamento do *cluster*, conforme pode apreciar-se na fig. 2.1.



Adaptado de: (PENGUINCOMPUTING, 2007)

**Figura 2.1** – Exemplo de topologia de *cluster* para processamento paralelo

### 2.1.1 CLUSTER BEOWULF

Para MERKLEY (2004), a busca por uma solução mais viável financeiramente levou aos pesquisadores Thomas Sterling e Donald J. Becker a interligarem 16 computadores pessoais, cada um com um microprocessador 486, usando o sistema operacional GNU/Linux e uma rede Ethernet<sup>4</sup>, visando alcançar o desempenho desejado com um custo menor.

Em 1994, junto com o *Center of Excellence in Space Data and Information Sciences* (CESDIS), uma divisão da NASA, tinha-se a necessidade de obter uma máquina com poder de processamento em torno de um gigaflop a fim de resolver problemas relacionados a grandes cálculos seqüenciais de dados em aplicações destinadas ao estudo das ciências da terra e do espaço, que resultaria em cálculos de um bilhão de operações em ponto flutuante por segundo. Uma máquina *MultiProcessing Parallel Processor* (MPP) com esse poder computacional na

<sup>3</sup>Nó - Máquinas de um *cluster*. Mesmo que nodo ou *node*

<sup>4</sup>Ethernet, tecnologia de interconexão para redes locais - Local Area Networks (LAN) - baseada no envio de pacotes.

época girava em torno de um milhão de dólares, o que era considerado algo muito caro para um grupo restrito de pesquisadores. Foi então montado o projeto de um *cluster* com 16 PCs 486 DX4, sistema operacional GNU/Linux, e um fato importante foi o uso de Ethernet 10Mbit/s. Os processadores eram relativamente rápidos para um único Ethernet, e os switches do Ethernet eram ainda muito caros. Para balancear o sistema, Donald Becker reescreveu seus *drivers* do Ethernet para Linux e construiu um Ethernet ligado a uma espécie de “*channel bonded*”<sup>5</sup>, em que o tráfego da rede fosse realizado por meio de dois ou mais *Ethernets*. Na época o *Fast-Ethernet* e os *Switches* eram inviáveis em função do custo. Com isso conseguiu-se uma grande capacidade de processamento a um custo relativamente baixo comparado com os computadores de alta performance. Esse cluster atingiu a marca de 70 megaflops<sup>6</sup>, com um custo final do projeto de US\$ 40.000,00, o que representou 10% do preço de uma máquina MPP com processamento equivalente em 1994. O sucesso do projeto se proliferou rapidamente no meio científico e acadêmico (STERLING, 2002).

Muitos problemas encontrados por computadores de alta performance podem facilmente ser divididos em peças menores, a serem trabalhadas simultaneamente. Os computadores de alta performance tradicionais podem frequentemente ser substituídos por um *cluster beowulf*, que são as máquinas individuais programadas para agir como um único computador.

Conforme SANTOS (2005, p.19), a opção pelo *cluster beowulf* é uma boa alternativa já que é um conjunto de computadores, podendo ser estações de trabalho, interligados através de uma rede para solucionar um determinado problema que requer alto desempenho (fig. 2.2). A essência de funcionamento desse tipo de *cluster* é baseada no uso de *softwares* livres, que reduzem ainda mais os custos.

De acordo com PITANGA (2004, p.27), algumas das características comuns de um *cluster beowulf* são a independência de fornecedores de *hardware* e *software*, periféricos escaláveis, *software* livre de código aberto, uso de ferramentas de computação distribuída disponível livremente com alterações mínimas, retorno à comunidade do projeto e melhorias, nenhum componente feito sob encomenda (em relação à dependência de fornecedor por soluções de *hardware* proprietárias).

A combinação de *hardware*, *software*, experiência e expectativa, desde o ambiente de desenvolvimento do *cluster beowulf*, gera certos eventos evolucionários naturais, que certamente são importantes, como as melhorias do desempenho dos microprocessadores e da tecnologia

---

<sup>5</sup>Channel Bonding - Topologia que permite aumentar a velocidade ou ter uma redundância na conexão usando duas ou mais placas de rede.

<sup>6</sup>Megaflop - FLOPS (ou *flops*) é um acrônimo que significa *Floating point Operations Per Second* que, em português, quer dizer operações de ponto flutuante por segundo. Isto é usado para determinar a performance de um computador, especificamente no campo de cálculos científicos, que fazem grande uso de cálculos com ponto flutuante



Fonte: (CSERI; NASA, 2007)

**Figura 2.2** – Cluster NASA: Um *cluster* feito com 64 PCs

de rede (gerando ganhos em termos do custo total de aquisição), tornando ele cada dia mais forte. Para a comunidade acadêmica, isso conduz a uma pesquisa diferenciada e à exploração de idéias novas. Com isso a utilização de soluções de MPPs proprietárias correlacionou o ciclo de vida de tais máquinas ao das carreiras profissionais daqueles que trabalham nelas. Explorar esses recursos requer ao programador adotar um modelo de programação específico e um estado de amarração com a tecnologia (MERKLEY, 2004).

Os *cluster beowulf* tem como característica ainda no seu funcionamento montar a arquitetura para executar uma única imagem ao usuário, tendo muitos computadores para processar vários recursos conforme a demanda. A tarefa é separada em partes independentes, distribuídas nos vários nós que estão na estrutura do *cluster*, nos quais as informações são processadas, realizadas pela máquina que é designada como *master* ou “mestre” do sistema.

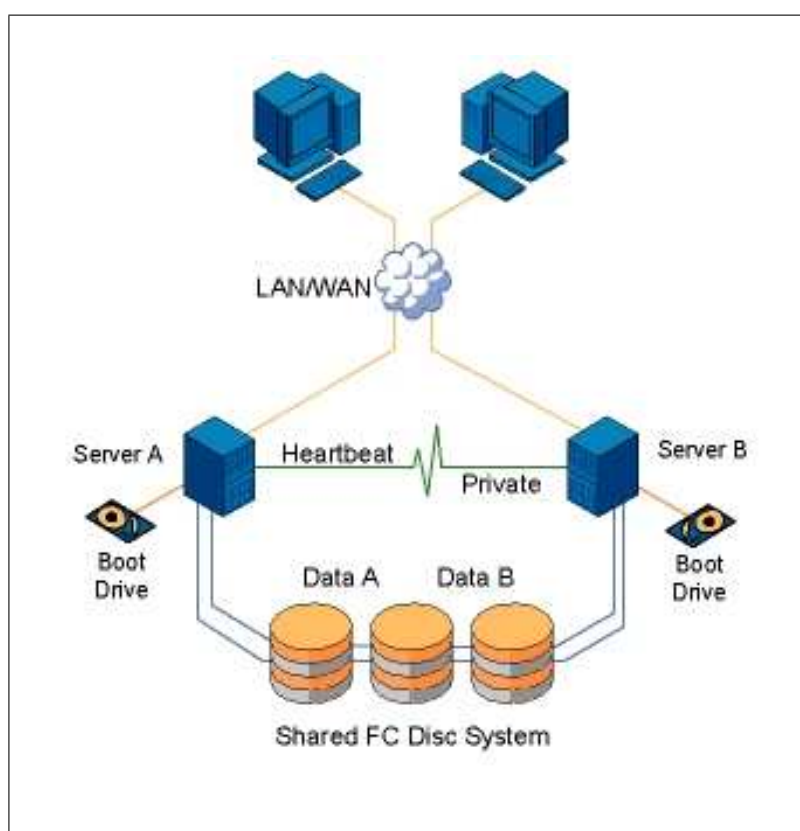
Por ser um *cluster* de alto desempenho, baixo custo, e com índice alto de aproveitamento, sua área de aplicação se expande rapidamente, sendo aplicada inclusive à produção de filmes. Um exemplo é o filme Titanic, para o qual foi montado um *cluster* com 105 máquinas comuns e Sistema Operacional Linux, utilizado para transformar os cálculos matemáticos da simulação em imagens, isso a um custo 40% menor do que poderia apresentar uma solução de um computador de alta performance para realizar o mesmo serviço (MERKLEY, 2004).

Apesar dos custos com o trabalho do ser humano (mão-de-obra) e do sistema de refrigeração e infra-estrutura física ser maior, ainda assim os *clusters beowulf* ficam, em média, dez vezes mais baratos do que os sistemas proprietários. Em um espaço limitado e não refrigerado, construir um *cluster beowulf* com quatro, oito, ou mesmo dezesseis nodos é à pri-

meira vista mais barato, pois não há custo com sistema operacional, os recursos requeridos são mínimos e os benefícios são maiores do que os custos (MERKLEY, 2004).

### 2.1.2 ALTA DISPONIBILIDADE (*High Availability*)

São *clusters* desenvolvidos para gerar uma alta disponibilidade de serviços e recursos por intermédio da redundância. De modo geral, se um nó do cluster tiver algum problema e falhar, os serviços e as aplicações transparentemente sem percepção do usuário estarão disponíveis para ele em outro nó, conforme demonstra-se na fig. 2.3.



Fonte: DELL (2007)

**Figura 2.3** – Solução de Alta-Disponibilidade ou *High Availability Clustering*

Esse sistema de *cluster* de *High Availability* (HA) mantém a disponibilidade dos recursos e serviços prestados em um sistema computacional, utilizando uma replicação dos recursos em outro servidor, por uma redundância e configurações de *software* para realizar essa tarefa. Na idéia principal de um *cluster* várias máquinas interagem como uma única máquina, tendo a função de monitorar uns aos outros. Em uma eventual situação esses backups acabam assumindo seus serviços caso algum deles venha a falhar. A grande complexidade está no *software*, que tem a função de monitorar os recursos dos computadores do *cluster*, verificando se os serviços estão sempre *on-line*, devidamente em cada servidor e definitivamente saber como

realizar a substituição do serviço se houver algum tipo de problema (PITANGA, 2004, p.27).

Os *clusters* HA têm, em sua maioria, o objetivo de não deixar o sistema parar, mas em eventuais quedas de performance de algum serviço e de capacidade de processamento, ainda que o sistema não pare completamente, pode executar a troca do nodo para retomar o melhor nível de performance, conforme tolerância configurada no *software* que monitora o sistema. Periodicamente esse *software*, conhecido como *keep alive*<sup>7</sup>, envia uma mensagem de um nodo para o outro. Se o nodo primário responder ao secundário, e vice-versa, significa que os nós do *cluster* estão em perfeito estado de funcionamento.

Sabe-se da importância de ter um sistema de tolerância a falhas, e, para isso, é possível disponibilizar alguns hardwares existentes, como placas, software de RAID<sup>8</sup>, fontes de energia e placas redundantes, sendo esses sistemas completamente interligados no monitoramento, havendo a possível substituição do recurso parado.

Normalmente, um servidor de qualidade mantém uma disponibilidade de 99,5%, enquanto uma solução através de clusters de computadores mantém 99,99% de disponibilidade (PITANGA, 2004, p.27).

### 2.1.3 BALANCEAMENTO DE CARGA (*Load Balancing*)

O tipo de *cluster* baseado em balanceamento de carga, tem a função de distribuir o tráfego e requisições de máquinas que estão no sistema. Isso pode ocorrer de duas formas:

- a) O sistema é exclusivo para realização de balanceamento de carga, ou seja, os nós não são utilizados para processar aplicativos individuais. Nessa situação há um nodo responsável por manter e controlar as solicitações e pedidos;
- b) O sistema realiza o balanceamento de carga e roda os aplicativos dos usuários, desse modo os nodos também são extremamente responsáveis em manter e controlar as solicitações e pedidos.

Sua grande especialidade é resolver problemas de serviços que tenham inúmeras solicitações em tempo real, como os serviços de comércio eletrônico, empresas que tenham seu sistema *on-line*, provedores de Internet e para solucionar diferentes números de carga que

---

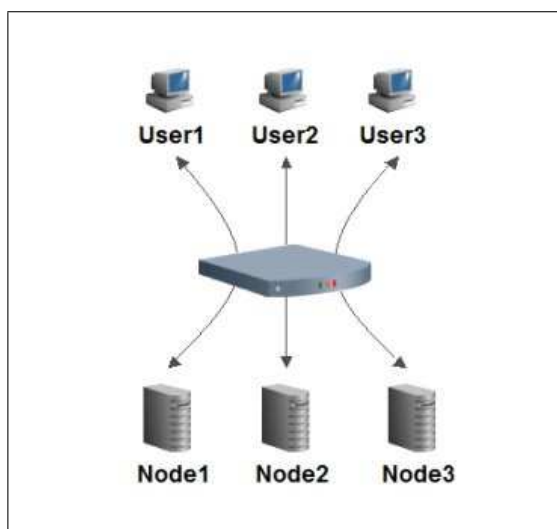
<sup>7</sup>*Keep-alive* este parâmetro controla a frequência que o *Transmission Control Protocol* (TCP) verifica se uma conexão ociosa está intacta, enviando um pacote *keep-alive*. Se o sistema remoto estiver alcançável e operacional, ele vai responder ao pacote *keep-alive*. Pacotes *keep-alive* não são enviados por padrão. Esse recurso pode ser habilitado em uma conexão por uma aplicação.

<sup>8</sup>RAID - *Redundant Array of Inexpensive Disks* (conjunto redundante de discos baratos) vários HDs são combinados para aumentar a performance. E, a um nível mais complexo, o RAID pode ser usado também para melhorar a confiabilidade do equipamento por meio de espelhamento ou paridade.

possa ter num exato momento. Para se ter um *cluster* de característica escalável, é preciso garantir que cada servidor tenha sua utilização de uma forma completa. Por tanto, não tendo um serviço e fazendo o balanceamento de carga entre máquinas com a mesma capacidade de retorno para um usuário, podemos, inicialmente, ter problemas como mais de uma máquina que possa responder à solicitação feita, em que a comunicação da rede fica mais congestionada e prejudicada (PITANGA, 2004).

Nesta estrutura de *cluster* são interligados os nós para suprir todas as solicitações de recursos e as requisições enviadas de clientes, sendo distribuídas de uma forma rápida e equilibrada entre os nodos. Esse tipo de *cluster* não trabalha ao mesmo tempo em um único processo, fazendo uma espécie de redirecionamento das solicitações de um jeito uniforme, definido em um algoritmo específico para escalonamento.

Os *clusters* com balanceamento de carga tornam-se indispensáveis em projetos de portais de comércio eletrônico, entre outros. São preferidos pelas empresas de *web* pela sua eficácia e seus benefícios e são aplicáveis a redes privadas de empresas de outros segmentos, permitindo resolver problemas dos usuários internos, fig. 2.4.



Fonte: (CHAN, 2006)

**Figura 2.4** – Solução de *Load-Balance* da Oracle para servidores Linux usando o *Global Single Instance*

## 2.2 PROCESSAMENTO EM PARALELO

De acordo com TANENBAUM (1995, p.245), processamento paralelo é o modo de execução em que várias instruções são executadas ao mesmo tempo e, normalmente, sobre vários dados.

A computação paralela está subentendida no conceito de computação distribuída, porém,



dependendo da aplicação o paralelismo não ocorre. Por isso, a importância de se definir a computação paralela, sendo que ela ocorre quando, num dado instante, existe mais de um processo trabalhando na resolução de um mesmo problema (SANTOS, 2005, p.21).

O processamento em paralelo é o método de quebrar problemas grandes em pequenos componentes, tarefas ou cálculos menores que são resolvidos em paralelo. O processamento em paralelo surgiu como uma tecnologia chave na computação moderna. Nos últimos anos, tem sido testemunhado uma enorme aceitação e adoção do processamento em paralelo, tanto para computação científica de alta-*performance* quanto para aplicações de finalidade “mais gerais”. Este foi o resultado de uma demanda para um desempenho mais elevado, um custo mais baixo, e uma produtividade sustentada. A aceitação de processar em paralelo foi facilitada por dois desenvolvimentos principais: processadores massivamente paralelos (MPPs) e o uso difundido da computação distribuída (MORRISON, 2003, p.12, tradução nossa).

Para PITANGA (2004, p.8), o paralelismo pode ser definido como uma técnica utilizada em grandes tarefas complexas para obter resultados na forma mais rápida possível, dividindo-se em tarefas pequenas que serão distribuídas em vários processadores para serem executadas simultaneamente.

A principal estrutura de um *cluster beowulf* é a utilização do sistema operacional GNU/Linux, com bibliotecas PVM ou MPI para fazer as trocas de mensagens, que facilitam a utilização e implementação para as aplicações paralelas. Estas duas bibliotecas possibilitam o desenvolvimento de um *software* portátil baseado na *Application Programming Interface* (API)<sup>9</sup> ou Interface de Programação de Aplicativos (MORRISON, 2003, p.56, tradução nossa).

### 2.2.1 PARALLEL VIRTUAL MACHINE (PVM)

TANENBAUM (1995, p.245) define PVM como: [...] um sistema distribuído que roda em um conjunto de máquinas sem memória compartilhada, máquinas estas que mesmo assim aparecem como um único computador para seu usuário.

O PVM fornece um ambiente paralelo utilizando o modelo de *Message Passing* (MP) (Passagem de Mensagem), que é um método de comunicação entre vários computadores com memória própria. Esses computadores se comunicam por troca de mensagens através de uma rede de computadores, respeitando as regras dos protocolos de comunicação (MORETTI; BITTENCOURT, 2004).

---

<sup>9</sup>API - é um conjunto de rotinas e padrões estabelecidos por um *software* para utilização de suas funcionalidades por programas aplicativos – isto é: programas que não querem envolver-se em detalhes da implementação do *software*, mas apenas usar seus serviços.

Segundo LOURENÇO (2002), existem vários mecanismos para troca de informações que podem ser utilizados na construção de *clusters*, alguns deles de distribuição livre como o PVM, que é um pacote de *software* que apresenta um conjunto de máquinas trabalhando como um único recurso computacional paralelo.

O PVM é estritamente o nome da biblioteca que permite escrever programas paralelos, que funcionam em um *cluster* baseados em Fortran e C. Do ponto de vista histórico, o PVM foi a primeira solução para funcionar em redes de computadores, criando assim uma máquina virtual paralela. Desde esse começo, foi adaptado para trabalhar em conjunto com supercomputadores paralelos (fazendo uso de recursos de memória distribuída e compartilhada) (MORRISON, 2003, p.56, tradução nossa).

Na utilização do PVM o administrador precisa configurar o *cluster*, visto que a melhor condição é manter as máquinas e a sua estrutura mais parecidas possíveis, para posteriormente ajudar na manutenção. Mas para isso é preciso promover uma forma de relação de confiança entre elas, e a forma mais simples é a utilização no GNU/Linux do *rhosts* e *rsh*, pois o administrador executa o gerenciador do PVM, podendo então adicionar novos nodos ao beowulf, depois é simplesmente rodado o programa desenvolvido para PVM.

A inicialização e a execução dos programas paralelos são feitas da seguinte forma. Antes da chamada de qualquer função, deve-se inicializar os “*daemons*”<sup>10</sup> nas diversas máquinas que formam a máquina virtual. Os “*daemons*” (pvmd3) ficam aguardando o recebimento de uma mensagem enquanto estiverem ativos. Uma vez rodando o programa mestre e os programas escravos, as funções de envio e recebimento das mensagens estabelecem a comunicação entre os diversos programas. Portanto, podemos “empacotar” um conjunto de informações, que possui um “nome” e enviar este pacote para um outro programa, que o estará aguardando, até que chegue uma determinada mensagem identificada por este “nome”. As informações passadas ao programa escravo normalmente são os dados necessários para sua execução. Ao terminar a execução do programa escravo os seus resultados são devolvidos ao programa mestre que reúne informações de vários escravos que foram executados em paralelo.

Para PITANGA (2004, p.114), o PVM foi um dos primeiros sistemas de *software* a possibilitar que programadores utilizem uma rede de sistemas heterogêneos para desenvolver aplicações paralelas.

---

<sup>10</sup>Daemon - é um programa de computador que roda em *background*, ao invés de ser controlado diretamente por um usuário. Tipicamente, os *daemons* têm nomes que terminam com a letra “d”; por exemplo, syslogd é o *daemon* que gerencia o *log* do sistema.

### 2.2.2 MESSAGE PASSING INTERFACE (MPI)

Conforme PITANGA (2004, p.122), o MPI é uma biblioteca com funções para troca de mensagens, responsável pela comunicação e sincronização de processos em um *cluster* paralelo. O principal objetivo do MPI é disponibilizar uma interface que seja largamente utilizada no desenvolvimento de programas que utilizem troca de mensagens.

O MPI define um conjunto de rotinas para facilitar a comunicação (troca de dados e sincronização) entre processos em memória, ele é portátil para qualquer arquitetura, tem aproximadamente 129 funções para programação e ferramentas para análise de *performance* (PITANGA, 2004, p.123).

O MPI consiste em um padrão com inúmeras implementações desenvolvidas normalmente pelas universidades, sendo elas suportadas comercialmente por empresas de máquinas MPP. O seu funcionamento baseia-se nas idéias bases do PVM (envio e recebimento de mensagens), porém disponibiliza mais recursos que este. Ele tem opções mais aprimoradas de como mandar as mensagens *broadcast*<sup>11</sup> para todos os nodos e *multicast*<sup>12</sup>, basicamente para um grupo específico de nós, facilitando assim o melhor controle sobre o tipo de tratamento que as mensagens terão ao serem recebidas por cada nó do *cluster beowulf*.

A configuração do MPI depende principalmente da estrutura onde está amarrada a implementação. Certas configurações chegam a instalar *front-ends*<sup>13</sup> para os compiladores C e Fortran, mas de uma maneira generalizada a sua aplicação é parecida. Com a utilização dessas bibliotecas, existem algumas alternativas para que as aplicações existentes sejam configuradas de uma maneira mais adequada em um ambiente distribuído, tornando novas aplicações mais práticas de gerenciar e programar. O valor dessa facilidade vem ajustado ao valor da performance (MORRISON, 2003, p.56, tradução nossa).

## 2.3 FLUIDODINÂMICA COMPUTACIONAL

Estudos de fluidodinâmica através da solução numérica das equações do movimento acopladas com as equações de conservação de massa e de energia, em um referencial microscópico, é denominado de estudo de fluidodinâmica computacional ou estudo de CFD (MA-

<sup>11</sup>Broadcast - em uma rede de computadores um sinal de *broadcast* é um aviso enviado simultaneamente para todos os micros da rede.

<sup>12</sup>Multicast - é um método ou uma técnica de transmissão de um pacote de dados para múltiplos destinos ao mesmo tempo.

<sup>13</sup>Front-end - em *cluster* é o nome dado à máquina mestre e à cabeça da arquitetura responsável pela distribuição dos processos e da criação da imagem virtual do sistema.

LISKA, 2004). Compreende estudos de modelagem matemática, geração de malhas numéricas em geometrias complexas, métodos numéricos e métodos de visualização científica. Aplicações industriais desta técnica de análise e simulação de processos iniciaram-se a partir de 1995 em situações denominadas de “gargalos” de processos, principalmente na indústria petroquímica (HAMILL, 1996).

A aplicação de técnicas numéricas computacionais para a resolução de problemas complexos da engenharia é atualmente uma das ferramentas mais utilizadas. Tudo isso foi possível graças ao grande desenvolvimento da computação nestas últimas décadas, o que permitiu o desenvolvimento da simulação numérica de forma acentuada (SILVA; AMARAL, 2006).

Os métodos numéricos e analíticos são designados como métodos teóricos, uma vez que tem por objetivo a solução de equações diferenciais que formam o modelo matemático. Devido à complexidade dos problemas de engenharia os métodos analíticos só podem ser aplicados se forem adotadas várias idealizações, o que gera um resultado que na maioria das vezes não representa de forma satisfatória o fenômeno físico real estudado (MALISKA, 2004).

No entanto os métodos numéricos podem ser empregados, levando-se em consideração todos os parâmetros físicos, matemáticos e geométricos que regem um problema qualquer.

Tendo em vista que a grande maioria dos fenômenos físicos já possui seu modelo matemático conhecido e validado, torna-se desnecessária a experimentação laboratorial, uma vez que os processos de simulação numérica podem resolver tais problemas de forma mais rápida e mais barata. Os métodos numéricos mais aplicados na atualidade são os métodos dos elementos finitos e o método dos volumes finitos. Entretanto, há uma tendência mundial de se utilizar mais amplamente o método dos volumes finitos (SILVA; AMARAL, 2006).

A CFD começou no final da década de 1970 e início dos anos 1980 e estudavam-se principalmente escoamentos com interesses bélicos, em especial o deslocamento de ar ao redor de aviões. Entretanto, visto que as mesmas equações que regem os fenômenos de fluxo nestes equipamentos são as equações que se aplicam aos escoamentos de um modo geral, esta ciência rapidamente se desenvolveu para a aplicação em outros campos da engenharia (MALISKA, 2004).

A partir da segunda metade do século passado, a fluidodinâmica computacional, CFD, começou a ser utilizada como uma ferramenta efetiva para muitos pesquisadores, que antes apenas poderiam contar com experimentos para compreender os fenômenos associados com os escoamentos de fluidos, no sentido de concluir seus trabalhos mais rapidamente e com menor custo (ORSZAG; STAROSELKY, 2000). Os autores também creditam a larga aplicação da ferramenta de CFD ao imenso avanço no setor computacional, que nos últimos cinquenta anos se desenvolveu a uma taxa de 2 vezes a cada ano e meio ou dois anos. Neste sentido, afirmam

que o desenvolvimento de tecnologias numéricas e matemáticas para solução de problemas surgiu na mesma velocidade e projetam um desenvolvimento ainda maior dos modelos e métodos computacionais.

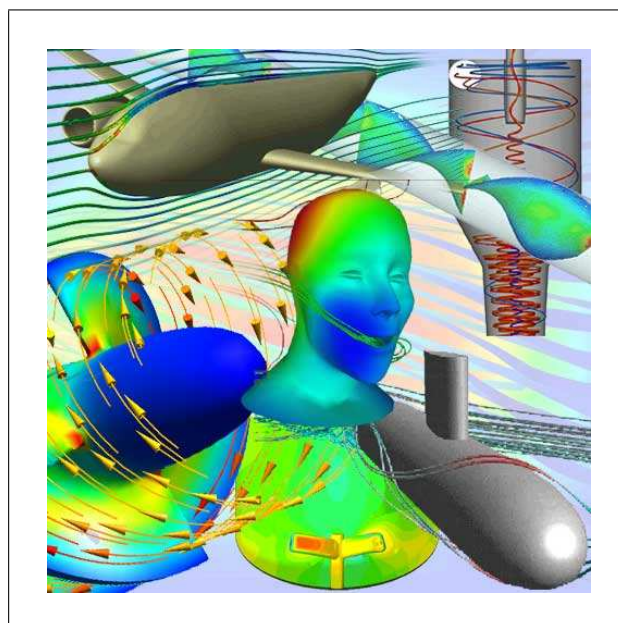
A fluidodinâmica computacional é uma tecnologia essencial a engenheiros para prever e manipular a dinâmica dos fluidos nos processos e equipamentos, a fim de minimizar os custos de construção e operação através de menor consumo energético e de matéria-prima, além da capacidade cada vez menor da geração de resíduos e contaminação ambiental (JOSHI; RENADE, 2003).

## 2.4 ANSYS CFX

O CFX da ANSYS é um *software* baseado no método dos volumes finitos por elementos para solução de problemas da dinâmica dos fluidos computacional.

Por muitos anos o CFX tem sido o software de CFD mais popular do mundo para predição de escoamentos complexos encontrados nas indústrias de processo e químicas. Agrega valor aos engenheiros responsáveis pelo desenvolvimento ou aprimoramento de produtos e processos que envolvam o escoamento de fluidos, transferência de calor e/ou reações químicas (ANSYS Company, 2003, tradução nossa).

Conforme ANSYS Company (2003), [...] o software atende especificamente às áreas: automotiva, biomédica, metalurgia, petróleo e gás, geração de energia, processos químicos, turbo máquinas, incêndio e segurança, aeroespacial, entre outros. fig. 2.5.

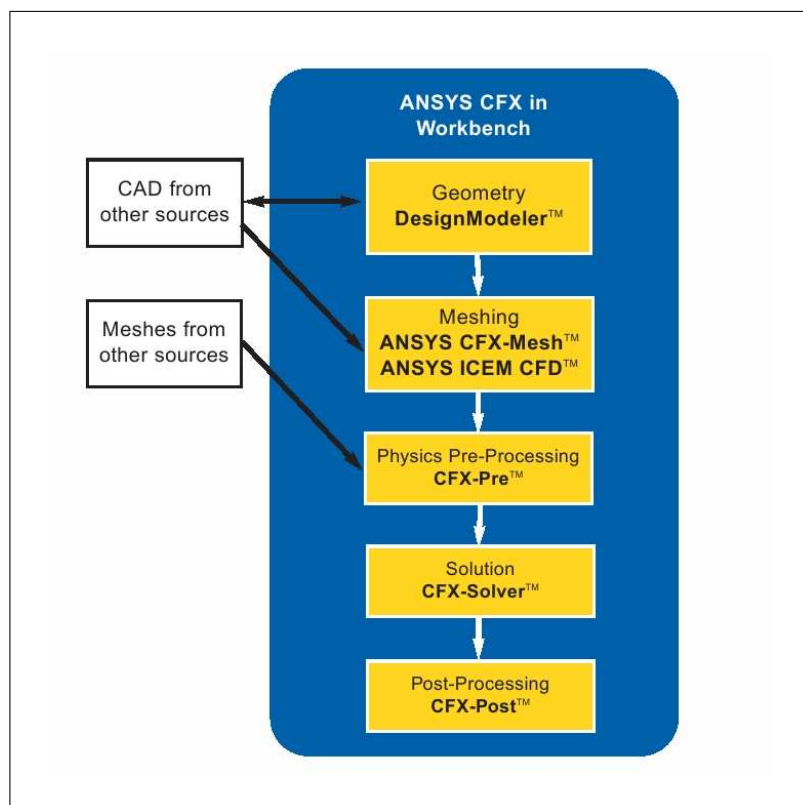


Fonte: (AEA Technology, 2000)

**Figura 2.5** – Resultados de vários modelos de CFD obtidos com o CFX.

O CFD tem se tornado uma parte integral da engenharia e da análise estrutural e ambiental de empresas, instituições de pesquisa e grandes corporações que requerem da habilidade de “antecipar” a performance de novos desenhos ou processos industriais antes de serem manufaturados ou implementados. A solução CFX da ANSYS vem acompanhando a fluidodinâmica computacional e contribuindo com esse sucesso por mais de 20 anos no mundo todo (ANSYS Company, 2003, tradução nossa).

A arquitetura do software CFX é dividida em três módulos principais, o CFX-Pre (Pre-Processor - Physics Preprocessor), o CFX-Solver e o CFX-Post conforme a fig. 2.6. O CFX-Pre define que para um problema de CFD existem três passos distintos - criação ou importação geométrica, criação de malha e organização física do problema. A flexibilidade do sistema CFX pode adaptar-se facilmente à geometria preferida como também para soluções de *meshing* (subdivisão de problemas em “n” pequenos problemas). O CFX-Solver, também conhecido como o coração do ANSYS CFX, é o motor algébrico que resolve os problemas numéricos matemáticos usando soluções multidimensionais e de *multigrid* (domínios múltiplos), produzindo cálculos exatos às equações lineares com uma rápida convergência. A tecnologia CFX está disponível na *suite* ANSYS Workbench. Criações geométricas, *meshing*, definição de problemas da física, solução e pós-processamento para CFD usam-se em um ambiente único de simulação totalmente integrado.



Fonte: (AEA Technology, 2000)

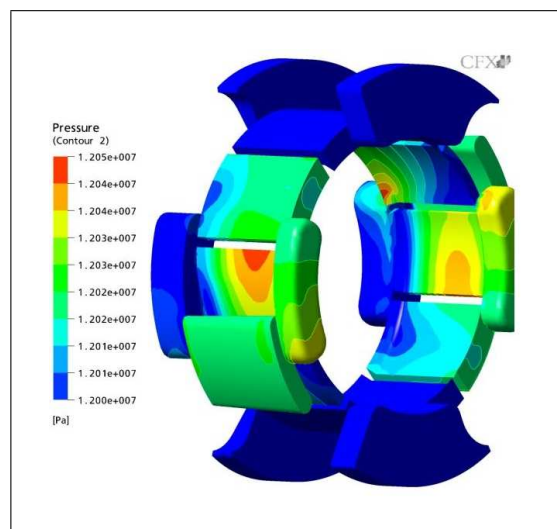
**Figura 2.6** – Arquitetura do ANSYS CFX.

O módulo CFX-Pre é uma interface moderna, consistente e intuitiva para a definição de

problemas físicos complexos requeridos pela análise em CFD. A ferramenta também lê de uma ou mais fontes para resolver o *meshing* e, desta maneira, fornecer várias opções para resultados em domínios diferentes.

O “*solver*” é totalmente escalável para processamento em paralelo, alcançando praticamente um escalonamento linear em termos de tempo por CPU vezes o tamanho do problema, isto é, fornece um crescimento constante conforme o problema a ser resolvido pela infra-estrutura computacional. Torna-se transparente resolver problemas de maneira serial ou em paralelo, representando verdadeiramente à física tradicional (ANSYS Company, 2003, tradução nossa).

Por último, o CFX-Post é uma ferramenta quantitativa, gráfica e poderosa de pós-processamento que permite rapidamente extrair informação útil dos resultados CFX. A fig. 2.7 apresenta um resultado gerado pelo pós-processador a partir dos dados obtidos numa simulação paralela do CFX-Solver.



Fonte: (AEA Technology, 2000)

**Figura 2.7** – Exemplo do resultado de uma simulação típica de modelos automotivos.

Uma vez que o modelo foi “resolvido”, os resultados gerados podem ser analisados tanto gráfica quanto numericamente. O *software* CFX fornece uma série de soluções para pós-visualização que variam de simples gráficos em 2D até representações avançadas em 3D de partículas, vetorização ou animações simuladas (ANSYS Company, 2003).

Combinando recursos de memória RAM com CPU ou múltiplos processadores em *Symmetric Multi-Processing* (SMP)<sup>14</sup>, o uso do CFX em ambiente paralelo fornece um método poderoso para resolver problemas à larga escala de CFD, permitindo ao usuário reduzir o tempo de cálculo computacional (ANSYS Company, 2003).

<sup>14</sup>SMP - Multiprocessamento simétrico é uma tecnologia que permite a um determinado sistema operacional distribuir tarefas entre dois ou mais processadores. Esse método, mais especificamente, permite que vários processadores compartilhem o processamento de instruções requisitadas pelo sistema.

De acordo com ANSYS Company (2003), o *software* CFX roda nas plataformas Hewlett-Packard PA-8000 ou IA-64, Itanium/Itanium2 rodando HP-UX 11i, Sun UltraSparc rodando Solaris 8,9 ou 10, SGI rodando IRIX 6.5.19, IBM Power rodando AIX 5.1, HP-Compaq rodando Tru64Unix 5.1, plataformas x86 usando os sistemas Windows 2000, Windows XP ou Redhat/SuSE Linux inclusive com processadores AMD64 Opteron e Intel EM64T. Suporte a métodos de “*double precision*” e suporte a endereçamentos de memória de 64-bit disponível para a maioria das plataformas.

## 2.5 TRABALHOS CORRELATOS

Em SCHNORR (2003), é descrito um protótipo de modelo de integração de sistemas de monitoramento de *clusters*, utilizando o SO Linux, as ferramentas *Ganglia* e *Performance Co-Pilot* em conjunto com as bibliotecas de rastreamento DECK e MPI. Neste trabalho o autor demonstra o modelo complexo do monitoramento de *clusters* como também a depuração das aplicações DECK e MPI executadas nessa infra-estrutura.

SCHNORR (2003) menciona em seu trabalho a importância da execução de aplicações paralelas em ambientes de *clusters* e *grids*, com a necessidade da tarefa de monitoramento. Segundo SCHNORR (2003), uma situação que demonstra a complexidade da tarefa de monitoramento de *clusters* acontece quando o desenvolvedor de aplicações em paralelo necessita de dados e informações relativas ao ambiente de execução da sua aplicação para entender melhor o seu comportamento. Com isso surgiu a idéia de desenvolver uma aplicação *web* que permitiria aos usuários dos recursos do *cluster* o monitoramento dos recursos, basicamente através da coleta de informações, e dessa maneira obter dados a respeito das aplicações executadas na infra-estrutura distribuída.

Em MINETTO (2005), foi desenvolvido um trabalho de pesquisa sobre Portais de *Grids* computacionais, onde são apresentadas as teorias e tecnologias envolvidas com os *Grids* computacionais, abordando as características, principais vantagens e ferramentas de desenvolvimento mais utilizadas em implementações distribuídas. Neste trabalho também são descritos portais que facilitam o acesso por parte dos usuários a todas as características e vantagens da computação em paralelo. Conforme MINETTO (2005), usando-se os portais de informação encontrados na Internet e os *Grids* computacionais, os usuários podem facilmente interagir com o ambiente computacional, submetendo procedimentos e acompanhando seu processamento.



### 3 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento, utilização e resultados deste trabalho da seguinte forma:

- a) instalações necessárias - contempla a instalação e configuração do *cluster beowulf*, do ANSYS CFX-10 no *cluster*, do serviço de *web* e o banco de dados;
- b) especificação da aplicação;
- c) implementação;
- d) resultados e discussão.

#### 3.1 INSTALAÇÃO E CONFIGURAÇÃO

Para o desenvolvimento da arquitetura do *cluster* foram utilizados computadores comuns, até um pouco defasados, com plataforma Debian GNU/Linux interligados pela rede acadêmica *Fast-Ethernet* do IPT da FURB. Formam parte desse *cluster* quatro computadores, um mestre e três nós, onde o mestre assume o papel de controlador e a função de executar o sistema *web*.

Algumas etapas no seguinte roteiro:

- a) instalação do sistema operacional e das principais bibliotecas e aplicativos para permitir a execução do *cluster*;
- b) configuração dos endereços lógicos IP para o controlador mestre e nós computacionais;
- c) criar o relacionamento de confiança entre as máquinas através de chaves RSA compartilhadas;
- d) instalação do ANSYS CFX-10;
- e) instalação dos serviços necessários para executar o sistema *web* no *master*.

Existem mais tópicos a serem acrescentados, porém o foco do trabalho é ter uma arquitetura básica para computação de alta performance em ambiente *cluster* GNU/Linux.

### 3.1.1 Instalação do SO, principais bibliotecas e aplicativos

A escolha pela distribuição Debian GNU/Linux foi feita primeiramente por causa da utilização do sistema “APT” para o gerenciamento de pacotes. Esta ferramenta inclui mais de 6500 novos pacotes na versão “*Etch*” (última versão estável), somando um total de mais de 18700 pacotes de *softwares* pré-compilados de código aberto, e uma rígida implementação do *Filesystem Hierarchy Standard* (FHS) ou Padrão de Hierarquia do Sistema de Arquivos. Além disso, o Debian é a primeira distribuição Linux a ter um contrato social formalizado e registrado. Tecnicamente é o estatuto de uma entidade não governamental formalizada (DEBIAN PROJECT, 2007).

Para a instalação do SO nas máquinas, foi utilizada a versão de instalação mínima via rede do sistema Debian GNU/Linux conhecida como “*netinstall*”, de simplesmente 140MB em formato ISO. Após feita a gravação em CD virgem não-regravável, se procede a inicializar o sistema através da unidade de CD-ROM.

Em todas as máquinas que compõem o *cluster* foram instalados discos de 80GBytes para suportar o sistema operacional, os *softwares* necessários para executar o *cluster* e o CFX. Na tabela 3.1 temos o particionamento de disco atribuído a cada um dos membros do *cluster*.

**Tabela 3.1** – Tabela de particionamento dos discos no *cluster*.

partição	tamanho	ponto de montagem	tipo
/dev/hda1	100MB	/boot	ext3
/dev/hda2	2GB	-	swap
/dev/hda3	72GB	/	ext3

Após feito o particionamento e escolhendo a opção “*Standard System*”, tem-se um sistema Debian básico após a sua instalação. É neste momento que, previamente reiniciando, instala-se o que realmente irá ser necessário. É importante salientar que desta maneira não temos nenhum programa desnecessário ocupando espaço nem serviços “*daemons*”.

Embora existam métodos como “clonagem” de discos ou de instalação através de uma rede interna visando facilitar a instalação em cada um dos nós de maneira rápida e fácil, o *cluster beowulf* desenvolvido para este trabalho é de pequeno porte e a infra-estrutura dedicada a este serviço é bastante básica.

Após a instalação básica do sistema, adiciona-se o usuário encarregado de submeter os trabalhos para serem processados no *cluster*. O comando a ser executado é “*useradd -g www-data -s /bin/bash -d /var/cluster/cfd -m lfc*” e “*passwd lfc*” para especificar uma senha. Este usuário receberá o *job* submetido através do sistema *web* e, mediante uso de um *script*,

encaminhará através do PVM a distribuição do trabalho entre os membros do *cluster*.

Nesse ponto, o sistema Debian já está apto para ser usado. No quadro 3.1 tem-se a lista de *softwares* necessários a serem instalados em cada membro do *cluster beowulf*.

```
# apt-get install mc iptables psad pvm pvm-dev \
libpvm3 gzip unzip bzip2 sudo strace psutils \
psmiscs dnsutils net-tools openssh binutils rsync \
perl-modules perl-base perl iproute rrdtool \
rdcollect
```

**Quadro 3.1** – Lista de programas a serem instalados no master e nos nós através do comando “*apt-get*”

Em seguida, deve ser alterado o *script* de execução do PVM “*/usr/bin/pvm*” em cada um dos membros do *cluster* de acordo com o algoritmo 1. Esta modificação permite ter um maior nível de compatibilidade entre a versão local do PVM e a fornecida pela ANSYS.

```
1 #!/bin/sh
2 # Modificado por Javier 02/06/06 2:39PM - Aponta pro ambiente CFX
3 PVM_ROOT=/opt/CFX/Shared/pvm3.4.4_11-1
4 if [ "$PVM_RSH" = ]; then
5     export PVM_RSH="/usr/bin/ssh"
6 fi
7 export PVM_ROOT
8 exec /opt/CFX/Shared/pvm3.4.4_11-1/lib/LINUX/pvm "$@"
```

**Algoritmo 3.1** – Modificação do *script* de inicialização do PVM

A identificação das máquinas que formam o *cluster* deve ser feita pelo nome e não pelo endereço IP, facilitando a administração de serviços como *Secure Shell* (SSH)<sup>1</sup> e PVM. Para isto, foi necessário associar os nomes com os endereços IP modificando o arquivo *hosts*. Este arquivo tem a função de associar o nome das máquina ao endereço IP correspondente. Uma cópia desse arquivo deve ser mantida em cada nó. Para a configuração do endereço da rede, foi usada uma configuração estática pelo fato do *cluster* possuir poucos nós e fazer parte da rede acadêmica. O conteúdo do arquivo *hosts* é apresentado no quadro 3.2 a seguir.

O controlador mestre usará o domínio totalmente qualificado denominado *master.lfc.furb.br*, assim como os nós computacionais receberão os nomes *no1.lfc.furb.br*, *no2.lfc.furb.br*, respectivamente. Outras possíveis configurações de endereçamento lógico empregam o uso de resolução de nome através do *Domain Name Service* (DNS) ou nomes de domínio, ou através do serviço de informação de rede *Network Information Service* (NIS).

<sup>1</sup>SSH - é um programa de computador e um protocolo de rede que permite a conexão com outro computador na rede, de forma a executar comandos de uma unidade remota. Possui as mesmas funcionalidades do TELNET, com a vantagem da conexão entre o cliente e o servidor ser criptografada.

```
# /etc/hosts: Local Host Database
201.54.199.2    master  master.lfc.furb.br
201.54.199.72   no1    no1.lfc.furb.br
201.54.199.71   no2    no2.lfc.furb.br
201.54.199.70   no3    no3.lfc.furb.br
```

**Quadro 3.2** – Conteúdo do arquivo */etc/hosts*

O serviço de *Network File System* (NFS) tem a função de permitir que diretórios e arquivos sejam compartilhados através da rede. Usuários e programas acessam arquivos localizados em computadores remotos como se fossem locais. O acesso é transparente baseado em uma arquitetura cliente/servidor, centralizando assim a administração de discos, pois é possível ter os diretórios em uma única máquina e compartilhar a diversos clientes conectados por uma rede local. O servidor NFS exporta diretórios para disponibilizá-los a clientes remotos, que no caso serão os nós computacionais (PITANGA, 2004, p.79).

Devido ao fato das máquinas estarem expostas à Internet através da rede acadêmica do Instituto de Pesquisas Tecnológicas da FURB com endereços IPs roteáveis, dispensou-se do uso da tecnologia de compartilhamento de arquivos NFS por questão de segurança. Caso seja construído um *cluster* com uma infra-estrutura dedicada e uma grande quantidade de nós, seria ideal a construção de uma rede privada para realizar a atribuição dinâmica, via protocolo *Dynamic Host Configuration Protocol* (DHCP), evitando a necessidade de atribuir manualmente endereços IP. Esta rede privada estará numa faixa de endereços inválidos na Internet, ou seja, as máquinas dessa intranet não são acessíveis pela Internet.

### 3.1.2 Relacionamento de confiança entre as máquinas através de chaves RSA compartilhadas;

Toda comunicação entre *master* e nós será feita através do OpenSSH, implementação de código aberto do SSH. O OpenSSH possibilita conexões remotas através de canais criptografados, deixando atrás os antigos programas de administração remota do UNIX *telnetd*, *rlogind*, etc. Pelo fato do OpenSSH encriptar todos os dados transmitidos, será configurado no ambiente o uso de autenticação por chaves sobre o protocolo SSHv2, dispensando a autenticação por senhas via teclado. Esta configuração aumenta o nível de segurança e confiabilidade entre as máquinas do *cluster*, permitindo a automação de *scripts* para a submissão dos trabalhos. No quadro 3.3 encontra-se o procedimento para a criação das chaves RSA.

Uma vez criada a chave que irá ser compartilhada, copia-se o arquivo “*~/.ssh/id\_rsa.pub*” para o diretório “*~/.ssh*” dos nós remotos, alterando o nome para

“*authorized.keys*”. Se houver mais de uma chave a ser compartilhada dentro do *cluster* (ex. administração remota), executa-se o comando “*cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized.keys*”. Por último, altera-se as permissões do diretório e do arquivo, como o comando “*chmod -R 700 ~/.ssh*”.

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key
(/home/lfc/.ssh/id_rsa):
Created directory ‘‘/home/lfc/.ssh’’.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/lfc/.ssh/id_rsa.
Your public key has been saved in
/home/lfc/.ssh/id_rsa.pub.
The key fingerprint is:
.....
```

**Quadro 3.3** – Chave compartilhada para acesso seguro usando o OpenSSH

Para testar se o procedimento anteriormente descrito deu certo, é necessário acessar o servidor *master* desde qualquer outro nó através do comando “*ssh lfc@no1*”.

### 3.1.3 Instalação do ANSYS CFX-10

Para a instalação da ferramenta ANSYS CFX versão 10 será necessário dispor de uma licença de uso por máquina que faz parte do *cluster*, ou seja um total de quatro licenças de uso. Cada licença deverá ter suporte pelo menos para processamento local, serial e paralelo. O uso dos programas da *suite* CFX Workbench e CFX ICED também requerem licenças porém não serão necessários.

O arquivo de instalação do CFX encontra-se no CD fornecido pela ANSYS conforme a plataforma e arquitetura a ser utilizada, por exemplo, para sistemas LINUX existem dois tipos de versões, a IA32 compatível com processadores de 32 bits e a AMD64/EMT64 para processadores com suporte a 64 bits. Nos CDs também é incluído o guia de instalação “*Quick Installation Guide*”, fornecendo maiores detalhes caso haja necessidade de realizar a instalação do ANSYS CFX em mais de uma plataforma (ex. Sun Solaris, IRIX, etc.).

Para começar a instalação, cria-se o local onde o CFX irá ser instalado através do comando “*mkdir /opt/CFX*”. Logo, montamos o CD de instalação conforme a arquitetura usada

com o comando “`mount /dev/hdc /mnt/cdrom`” (com o leitor de CDs configurado como “escravo” do primeiro canal IDE) ou diretamente “`mount /mnt/cdrom`”, e inicia-se a instalação através do comando “`/mnt/cdrom/install`”.

Após a execução do *script* de instalação, deve-se especificar quais irão ser as ferramentas a serem instaladas. A não ser em casos específicos, normalmente o padrão da instalação já é suficiente, caso for feito a instalação de uma ferramenta da qual não se possua uma licença válida, o programa não irá executá-la e acusará um erro ao usuário. O instalador do CFX automaticamente criará a subpasta “CFX-10.0”, este local também é referenciado através da variável *CFXROOT*.

A seguinte modificação inicializa o serviço de licenciamento a cada *boot* do SO. Para isto, acrescenta-se a linha “`su lfc -c /opt/CFX/shared_files/licensing/linem64t/lmgrd -c /opt/CFX/shared_files/licensing/license.dat &`” dentro do arquivo “`/etc/rc.local`”. Isto fará com que o serviço de licenciamento do CFX leia automaticamente nossa licença, a qual é guardada na pasta “`/opt/CFX/shared_files/licensing/`” como o arquivo de nome “`license.dat`”, processe a chave e gere a configuração necessária para ativar o licenciamento do master e dos nós conforme for necessário.

Nesse ponto, restaria fazer algumas alterações na configuração das variáveis do ambiente *shell*. O arquivo “`/etc/profile`” contém comandos que são executados para todos os usuários do sistema no momento do *login*, e é lido antes do arquivo de configuração pessoal de cada usuário (no nosso caso o usuário “`lfc`”). Conforme indicado no manual da ANSYS, terão de acrescentar-se algumas variáveis próprias do CFX, mudanças no caminho de execução (*PATH*), comandos aliases e modificações de variáveis do PVM. Estas mudanças são representadas no algoritmo 3.2.

```
1 alias rsh='ssh'
2 PATH=/opt/CFX/CFX-10.0/bin:$PATH
3 CFX5RSH=/usr/bin/ssh
4 PVM_RSH=/usr/bin/ssh
5 PVM_ROOT=/opt/CFX/Shared/pvm3.4.4_11-1
6 export PATH CFX5RSH PVM_ROOT PVM_RSH
```

**Algoritmo 3.2** – Alterações nos arquivos `/etc/profile` e `.bashrc`

Para ganhar tempo na instalação do CFX nos nós do *cluster*, basta realizar a cópia da pasta `/opt/CFX/*` para cada um dos nós e aplicar as alterações nas variáveis do ambiente. Como o CFX usa o PVM como sistema distribuído, e ao mesmo tempo o PVM usa o serviço de RSH para estabelecer a comunicação entre os nós, é especificado um “*alias*” para forçar o uso do comando “`ssh`” a cada vez que for chamado o comando “`rsh`”, e mediante o uso de variáveis, especifica-se o uso do comando “`rsh`” usado pelo PVM para também ser o “`ssh`”. Como já tinha

tido estabelecido o relacionamento de confiança através de chaves RSA entre os nós, o padrão de comunicação fica cifrado e seguro.

Uma vez instalado, é possível verificar as variáveis atribuídas no sistema para o CFX através do comando “*cfx5info*”, ou selecionando no menu principal do “*Launcher*” a opção *Show* → *Installation*.

#### 3.1.4 Instalação dos serviços necessários para executar o sistema *web* no *master*

Conforme foi explicado anteriormente, no nó mestre será instalado o sistema de gerenciamento *web*. Para esta função, será necessário instalar o servidor de páginas *web* Apache, a linguagem de programação interpretada Perl em conjunto com as respectivas bibliotecas e o banco de dados MySQL. A lista dos pacotes a serem instalados usando a ferramenta “*apt-get*” é apresentado no quadro 3.4.

```
# apt-get install apache2-doc apache2.2-common \  
apache2-utils apache2-mpm-prefork libapache2-mod-perl2 \  
ca-certificate libcgi-session-perl ssl-cert libplrpc-perl \  
libcgi-session-expireessions-perl libhtml-table-perl \  
libdbi-perl libnet-daemon-perl libcrypt-passwdmd5-perl \  
libclass-dbi-mysql-perl libclass-dbi-perl mysql-common \  
libdbd-mysql-perl libcgi-perl mysql-server-5.0 \  
libcrypt-unixcrypt-perl libcrypt-des-ede3-perl \  
libdbd-mysql-perl libcgi-perl mysql-server-5.0 \  
mysql-client-5.0 libdatetime-perl \  
libset-crontab-perl
```

**Quadro 3.4** – Lista de programas a serem instalados somente no *master*

Para especificar uma senha de acesso ao “Administrador do banco” (usuário “*root*”), digita-se o comando “*mysqladmin -u root password 'minhanovasenha'*”.

O *script* necessário para realizar a criação das tabelas dentro do banco MySQL encontra-se no “Apêndice A”. Para inserir o usuário “*admin*” que terá as permissões necessárias para começar a criar as contas no sistema, será necessário executar o *script* descrito no “Apêndice B”.

O sistema *web* usa a pasta “*/var/clusterofd*” e a subpasta “*storage*” no *master* como local

referência à pasta que contém arquivos de configurações e o armazenamento dos *jobs* submetidos ao *cluster*. Desta maneira, é necessário dar à subpasta de armazenamento “*storage*” as permissões de sistema “770” para o usuário “*lfc.www-data*”.

A pasta “*/srv/projeto-cluster*” é o diretório principal do servidor de páginas *web* Apache. Dentro dessa pasta coloca-se o código Perl, as imagens e os códigos CSS e HTML estáticos. Para completar a instalação do servidor *web*, adicionam-se os seguintes arquivos na pasta de configuração do apache no *master*, conforme apresentado no quadro 3.5 e no quadro 3.6.

O arquivo de configuração apresentado no quadro 3.5 prepara o servidor para aceitar conexões criptografadas via *web* através do uso do SSL.

```
NameVirtualHost *:443
<VirtualHost *:443>
    ErrorLog /var/log/apache2/cluster-error.log
    LogLevel debug
    ServerAdmin javier.augusto@gmx.net
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/apache.pem
    DocumentRoot /srv/projeto-cluster/httpd
    <Directory />
        Options None
        AllowOverride None
    </Directory>
    <Directory ‘‘/srv/projeto-cluster/httpd/cgi-bin’’>
        AllowOverride None
        Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
    ServerSignature Off
    HostnameLookups Off
    AddHandler cgi-script .cgi
    AddHandler cgi-script .pac
    AddHandler cgi-script .pa </VirtualHost>
```

**Quadro 3.5** – Conteúdo do arquivo */etc/apache2/sites-enabled/cluster-cfd*

Um detalhe importante a ser mencionado é o uso do arquivo de configuração



```
NameVirtualHost *:80
<VirtualHost *:80>
  ServerAdmin webmaster@lfc.furb.br
  DocumentRoot /srv/projeto-cluster/web-default
  <Directory />
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>
</VirtualHost>
```

**Quadro 3.6** – Conteúdo do arquivo */etc/apache2/sites-enabled/000-default*

“*/var/clusterbfd/config.pl*”. Este arquivo centraliza as configurações das variáveis a respeito do servidor de banco de dados, usuário e senha usados para acessar o banco, etc. Este arquivo terá de ser alterado conforme a infra-estrutura a ser usada.

Após essas configurações, o ambiente de *cluster* está pronto para a implementação e execução de aplicações paralelas. Se for necessária a inserção de novos nós escravos, basta repetir as configurações realizadas nos nós escravos que já estão em funcionamento. Além disso, a única alteração nos demais nós, incluindo o mestre, é inclusão do endereço IP e do nome da nova máquina no arquivo *hosts*.

Da forma que o *cluster* está configurado a sua arquitetura é flexível, no sentido de que não há uma limitação para a quantidade de nós escravos, sendo possível a inserção, remoção ou, ainda, a substituição dos mesmos de maneira simples, conforme for necessário.

## 3.2 REQUISITOS PRINCIPAIS

O objetivo da atividade de levantamento dos requisitos é de definir qual programa é preciso desenvolver. No quadro 3.7 apresentam-se os requisitos funcionais previstos para o sistema, identificando os requisitos que deverão ser implementados.

Os requisitos não funcionais põem restrições sobre como o programa deve funcionar (ANQUETIL, 2002). O quadro 3.8 lista os requisitos não funcionais previstos para o sistema.

Requisitos Funcionais	Caso de Uso
RF01: O sistema deverá permitir aos Usuários a alteração da senha de acesso.	UC01
RF02: O sistema deverá possibilitar o registro de agendamentos de trabalhos dentro do <i>cluster</i> pelo Usuário e Administrador.	UC02
RF03: O sistema deverá permitir a escolha dos recursos computacionais dentro do <i>cluster</i> para a alocação de trabalhos pelo Usuário e Administrador.	UC03
RF04: O sistema deverá permitir a escolha da metodologia de processamento pelo CFX-Solver para o processamento de trabalhos dentro do <i>cluster</i> pelo Usuário e Administrador.	UC04
RF05: O sistema deverá permitir a emissão de relatório dos trabalhos que foram agendados no <i>cluster</i> pelo Usuário e Administrador.	UC05
RF06: O sistema deverá permitir a geração de relatório dos trabalhos executados no <i>cluster</i> pelo Usuário e Administrador.	UC06
RF07: O sistema deverá permitir o cadastro de Usuários pelo Administrador.	UC07
RF08: O sistema deverá acompanhar o estado de execução em termos de processamento do <i>cluster</i> pelo Usuário e Administrador.	UC08
RF09: O sistema deverá possibilitar ao Administrador adicionar recursos computacionais para o <i>software</i> CFX escalar dentro do <i>cluster</i> conforme for necessitado pelo desempenho dos trabalhos.	UC09
RF10: O sistema deverá possibilitar ao Administrador a geração de relatórios de uso do <i>cluster</i> pelos Usuários.	UC10

**Quadro 3.7** – Requisitos Funcionais

### 3.3 VISÃO GERAL DA SOLUÇÃO PROPOSTA

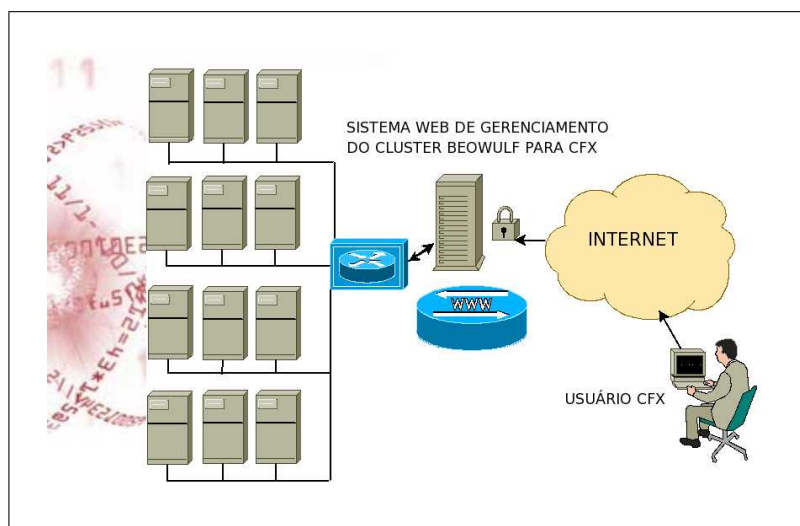
Cada usuário é responsável pelo desenvolvimento do caso a ser executado como também dos recursos alocados dentro do *cluster*. Se o usuário não reservar recursos computacionais para a execução do trabalho, o sistema automaticamente alocará o máximo disponível em quantidades de nós para realizar a distribuição de carga e, desta maneira, acelerar ao máximo o nível de processamento computacional. Para cada *job* submetido, seja agendado ou executado no momento da submissão, o sistema criará uma pasta no servidor *master* com o nome do arquivo principal CFX e a data atual. Dentro dessa pasta irão ficar os arquivos necessários para o processamento das aplicações CFX, os *logs* e *dumps* numéricos gerados pelo CFX-Solver conforme os cálculos numéricos forem resolvidos. Também é gerado no banco de dados uma referência

Requisitos Não Funcionais
RNF01: O sistema deverá utilizar os protocolos TCP/IP para comunicação.
RNF02: O banco de dados utilizado deve ser MySQL versão 4.0 ou superior.
RNF03: O servidor Web deve ser Apache versão 1.3.x ou 2.x com suporte à linguagem Perl em modo <i>Common Gateway Interface</i> (CGI).
RNF04: A linguagem Perl instalada no servidor de páginas deverá possuir suporte à implementação DBI de acesso ao banco de dados MySQL (Perl-DBD-MySQL).
RNF05: O sistema deverá possuir controle de acesso por usuário e senha.
RNF06: O sistema deverá possuir uma versão do ANSYS CFX instalada e licenciada.
RNF07: A versão do pacote PVM deverá ser o PVM3 ou superior.
RNF08: A versão do pacote MPI deverá ser compatível com o <i>software</i> CFX.
RNF09: O sistema deverá possuir um serviço de SSH ( <i>Secure Shell</i> ) habilitado e configurado.
RNF10: O sistema deve ter tempo de resposta em consultas inferior a 5 segundos em uma condição de rede normal.
RNF11: O sistema deverá permitir a geração de relatório de acesso ( <i>log</i> ).
RNF12: O sistema deverá gerar código HTML compatível com os navegadores Mozilla, Internet Explorer, Opera ou Netscape.

**Quadro 3.8** – Requisitos Não Funcionais

a cada *job*, usuário que submeteu o trabalho, nome de arquivo, data e estado do processo, de forma a controlar e emitir posteriormente os relatórios das submissões ao *cluster*. O estado do *job* no *cluster* pode ser “processando”, “erro”, “agendado” ou “finalizado”.

Uma visão geral do sistema pode ser visto na fig. 3.1.



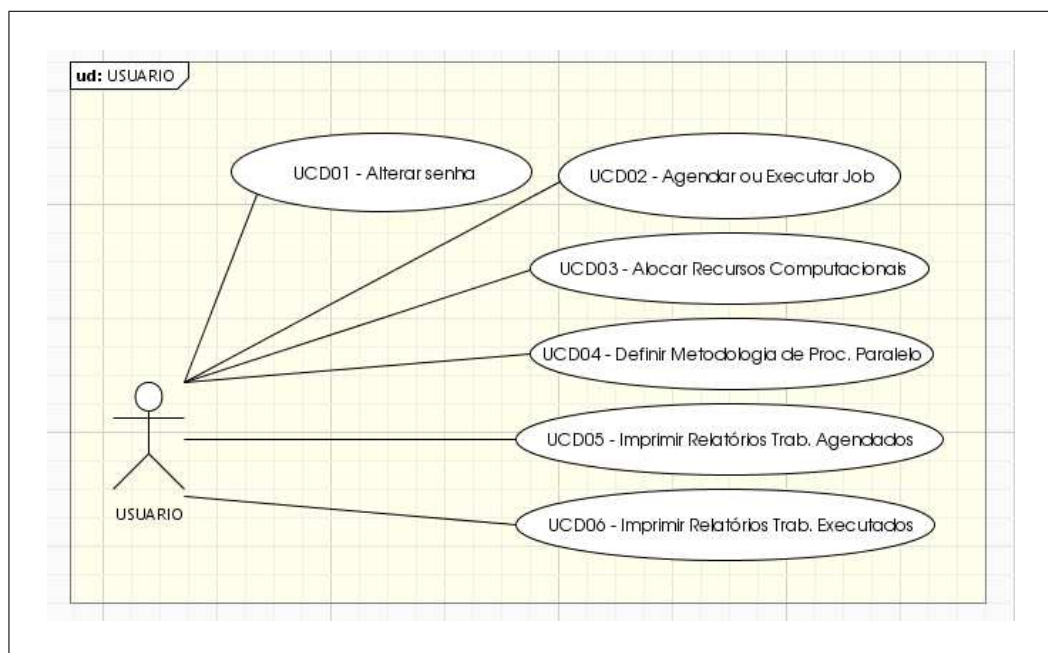
**Figura 3.1** – Visão geral do sistema

### 3.4 ESPECIFICAÇÃO

A especificação do sistema foi feita seguindo a linguagem de modelagem *Unified Modeling Language* (UML). É uma linguagem para especificar, construir, visualizar e documentar um sistema de software (MACORATTI, 2006). Os diagramas foram especificados utilizando a ferramenta Poseidon UML 6, edição *Community Edition* (GENTLEWARE, 2007).

#### 3.4.1 Casos de Uso

Os diagramas de casos de uso estão divididos em duas partes e representam o caso de uso para o “usuário” e o caso de uso para o “administrador”. A fig. 3.2 e fig. 3.3 especificam os diagramas de caso de uso do sistema, onde, como atores, tem-se o usuário e o administrador. A seguir são detalhados os casos de uso do aplicativo desenvolvido.



**Figura 3.2** – Diagrama de Casos de Uso para Usuário

##### 3.4.1.1 Caso de uso: Usuário - Alterar senha

Ao selecionar esta opção, o sistema deverá exibir uma tela solicitando ao Usuário informar uma nova senha, a alteração do nome do usuário ou do e-mail cadastrado. Informando quaisquer dessas opções, o sistema deverá exibir uma tela indicando se foi ou não realizada essa

mudança.

#### 3.4.1.2 Caso de uso: Usuário - Agendar ou Executar *job*

O Usuário deve selecionar uma data e horário para realizar o agendamento de execução do trabalho CFX dentro do *cluster*. O sistema exibirá então uma tela indicando cada um dos itens a serem informados, como dia, hora, minutos, arquivo CFX com extensão “.def” e, opcionalmente, o arquivo de dados “.ref” ou “.bak”. Após ter selecionado data, horário, e o(s) arquivo(s) CFX, o sistema deverá exibir uma janela indicando se foi ou não agendado corretamente o trabalho dentro do *cluster*. Caso o usuário não queira agendar o trabalho mas somente executá-lo, deve usar a opção de execução de trabalhos presente na mesma tela.

#### 3.4.1.3 Caso de uso: Usuário - Alocar Recursos Computacionais

Permite que o Usuário escolha quais irão ser os recursos usados para realizar a resolução do problema dentro do *cluster*.

#### 3.4.1.4 Caso de uso: Usuário - Definir Metodologia de Processamento em Paralelo

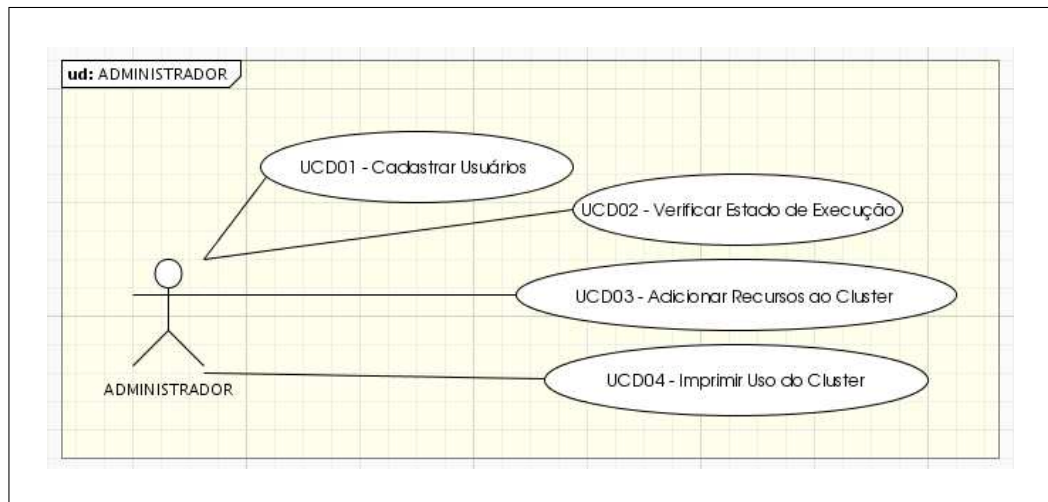
Embora o sistema *web* permita escolher entre PVM e MPI, o CFX somente irá usar o PVM pelo fato do uso de uma plataforma única com GNU/Linux. Mais informações na seção “Limitações do sistema”.

#### 3.4.1.5 Caso de uso: Usuário - Imprimir Relatórios de Trabalhos Agendados

Permite ao Usuário verificar quais trabalhos estão agendados no *cluster beowulf* e quais recursos estarão sendo usados.

#### 3.4.1.6 Caso de uso: Usuário - Imprimir Relatórios Trabalhos Executados

Permite ao Usuário verificar quais trabalhos foram executados no *cluster beowulf*, e apagar ou salvar os resultados obtidos conforme for desejado.



**Figura 3.3** – Diagrama de Casos de Uso para Administrador

#### 3.4.1.7 Caso de uso: Administrador - Cadastrar Usuários

O Administrador deverá realizar o cadastro do Usuário para permitir a utilização do sistema *web*.

#### 3.4.1.8 Caso de uso: Administrador - Verificar Estado de Execução

Permite ao Administrador ter um estado de cada um dos trabalhos executados no *Cluster* pelos Usuários.

#### 3.4.1.9 Caso de uso: Administrador - Adicionar Recursos ao *Cluster*

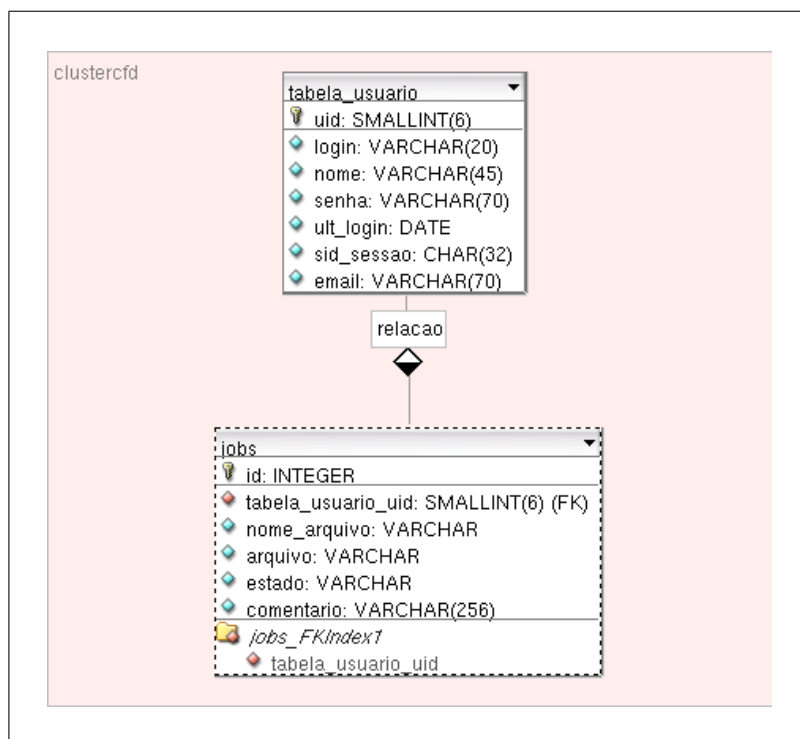
Permite ao Administrador o cadastro de recursos computacionais para serem disponibilizados para posterior uso por parte dos Usuários.

3.4.1.10 Caso de uso: Administrador - Imprimir Relatórios Uso do *Cluster*

Possibilita ao Administrador obter um relatório estatístico sobre a quantidade total de trabalhos executados pelos Usuários como também quais Usuários usaram mais o *cluster*.

## 3.4.2 Modelo Conceitual

Na fig. 3.4, pode ser visualizado o modelo conceitual (MER), através do qual foram geradas as tabelas utilizadas no aplicativo. O modelo MER foi construído utilizando a ferramenta DBDesigner 4.



**Figura 3.4 – Diagrama MER**

A tabela “tabela\_usuario” guarda os dados a respeito dos usuários que utilizam o sistema para submissão dos trabalhos CFX no *cluster*. O campo “uid” é gerado automaticamente pelo banco e é a chave primária. O campo “login” fornece o nome a ser utilizado para acessar o sistema. Usa-se o campo “nome” para indicar o nome completo do usuário. O campo da senha é criptografado e a função será comentada posteriormente. O campo “sid\_sessao” é atualizado com a sessão gerada na hora de acessar o sistema. Para informar um e-mail de contato, usa-se o campo “e-mail”. Por último o campo “ult\_login” é atualizado a cada vez que o usuário acessa o sistema e serve para que o “admin” tenha um controle de quando foi realizado um

acesso. A tabela “jobs” guarda informações a respeito dos trabalhos que são executados ou agendados no sistema e é relacionada à tabela “tabela\_usuario”. O campo “nome\_arquivo” é usado para informar o nome gerado pelo sistema *web*, este nome é gerado usando o nome do arquivo submetido mais a data e o horário. Já o campo “arquivo” informa o verdadeiro nome do trabalho a ser processado. Por último, o campo “estado” indica qual o estado do processo dentro do sistema. O campo “comentario” é reservado e não foi utilizado.

### 3.4.3 Diagrama de Atividades

Um diagrama de atividades representa os estados de uma atividade e são orientados a fluxos de controle. Os dois fluxos principais podem ser acompanhados nas fig. 3.5 e fig. 3.6 a seguir.

O diagrama da fig. 3.5 representa o processo realizado pelo usuário quando executa um *job* para ser processado através do sistema *web*. Ao efetuar a submissão, o sistema *web* realiza uma série de verificações antes de repassar o *job* ao subprocesso de submissão do *cluster*. Primeiramente, o sistema verifica se foi reservado algum recurso computacional, caso o usuário tenha reservado algum recurso, as opções ficarão salvas no arquivo “*settings*” dentro da pasta pessoal do usuário no *master*. Se o usuário não tinha reservado recursos para a sessão em uso, o sistema também utilizará o arquivo “*settings*” porém com a configuração original, que usa todos os recursos do *cluster*. A escolha do método de processamento em paralelo é feita verificando se existe ou não o arquivo “MPI” dentro da pasta pessoal do usuário, se existir é porque o usuário indicou o uso do MPI, caso contrário o sistema automaticamente irá utilizar o PVM. Depois, o sistema atualiza o banco de dados com as informações do *job* para o estado de “processando” e repassa ao subsistema de processamento do CFX a execução do problema. Por último, é verificado o resultado da simulação através dos sinais que retorna o CFX-Solver, se forem diferentes a “0” significa que houve um erro na resolução do trabalho. O código fonte do *script* em Perl implementado encontra-se na pág. 55.

Na fig. 3.6, o diagrama representa o processo realizado pelo sistema *web* sempre que um usuário agendar um *job*. Basicamente, o sistema implementa a mesma rotina que no diagrama anterior com a única diferença que ao invés de executar o *job*, será criada uma entrada do tipo “*crontab*” para posteriormente executar a simulação dentro do *cluster* no horário agendado.



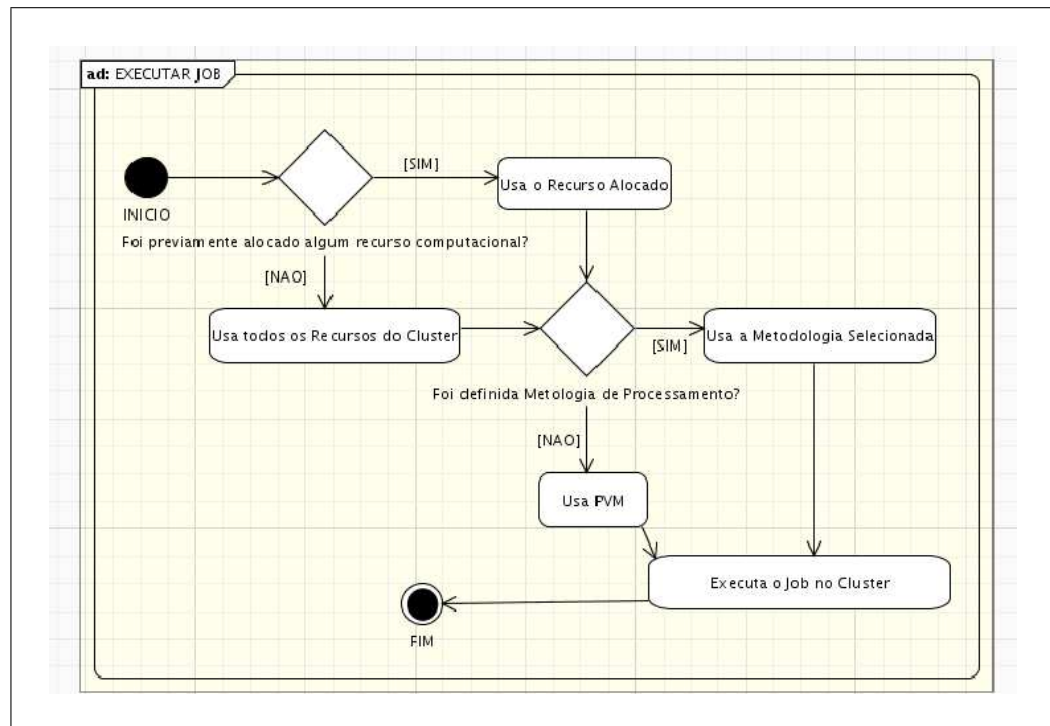


Figura 3.5 – Diagrama de Atividade do processo de submissão de *job* via *web*

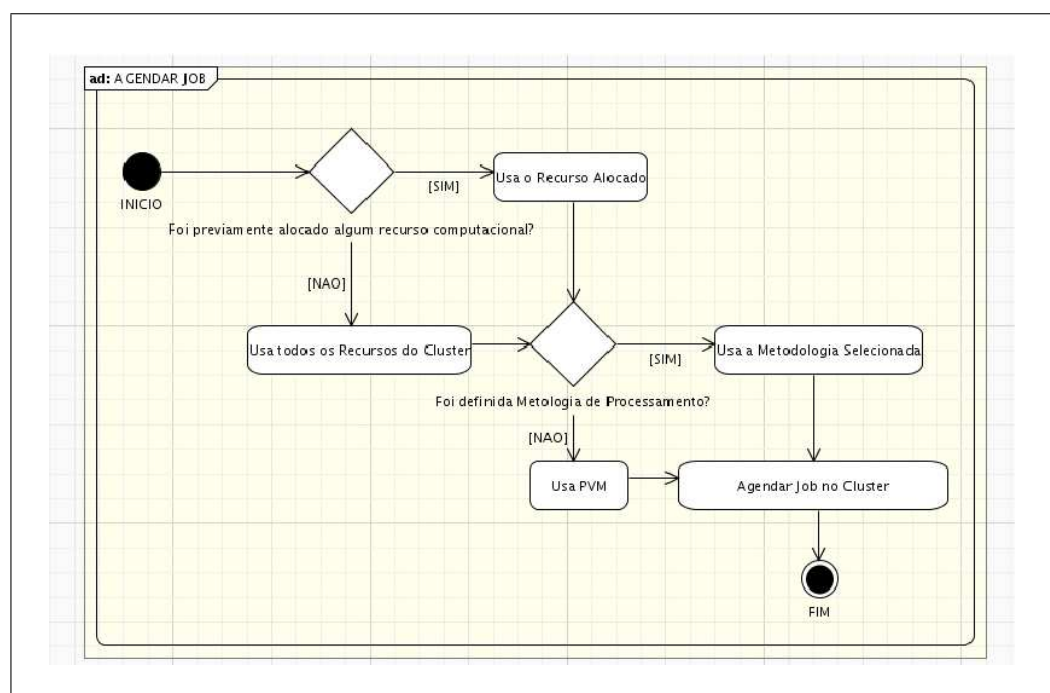


Figura 3.6 – Diagrama de Atividade do processo de agendamento via *web*

### 3.5 IMPLEMENTAÇÃO

Para o desenvolvimento da aplicação foi utilizado o ambiente ActiveState Perl Dev-Kit, onde foi utilizada a linguagem Perl através do CGI no servidor *web*, *scripts* em Perl para o subsistema de agendamento e execução dos *jobs* do CFX em paralelo, e o interpretador de comandos (*shell*) BASH para implementar as rotinas de atualização do estado do *cluster* e nós.

#### 3.5.1 Técnicas e ferramentas utilizadas

A seguir serão apresentadas as técnicas e ferramentas utilizadas no desenvolvimento deste trabalho.

##### 3.5.1.1 Perl

O Perl foi criado por Larry Wall em 1987 e é uma linguagem de programação estável e multiplataforma, usada em aplicações de missão crítica em todos os setores, sendo destacado o seu uso no desenvolvimento de aplicações *web* de todos os tipos. A origem do Perl remonta ao *shell scripting*, Awk e linguagem C, estando disponível para praticamente todos os sistemas operacionais, embora seja usado mais comumente em sistemas UNIX e compatíveis (WIKIPEDIA, 2007a).

É uma linguagem de programação interpretada, assim como muitas outras linguagens da Internet como Javascript ou ASP. Isto quer dizer que o código dos *scripts* em Perl não se compila e sim, que cada vez que se quer executar, se lê o código e se coloca em funcionamento o que há escrito. Além disso, é extensível a partir de outras linguagens, já que desde Perl podem ser feitas chamadas a subprogramas escritos em outras linguagens. Também desde outras linguagens é possível executar o código Perl (ALVAREZ, 2007).

Perl é uma das linguagens preferidas por administradores de sistema UNIX e autores de aplicações para a *web*. É especialmente versátil no processamento de cadeias (*strings*), manipulação de texto e no *pattern matching* implementado através de expressões regulares, além de permitir tempos de desenvolvimento curtos (WIKIPEDIA, 2007a).

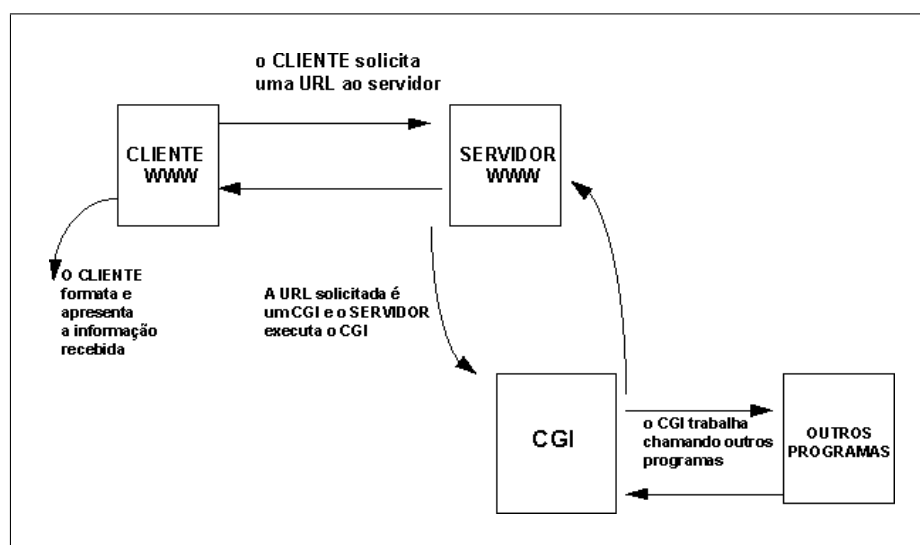
Os pontos fortes do Perl são (CGICLUBE.NET, 2007):

- a) declaração de variáveis implícita. O primeiro carácter distingue se a variável é um escalar, um *array* ou um *array* associativo;

- b) *strings* e *arrays* não necessitam de definição de tamanho. Dinamicamente eles são criados pequenos e crescem conforme a necessidade;
- c) todas variáveis são inicializadas com um valor *default*;
- d) um conjunto muito rico de operações de busca “padrões” em textos;
- e) um conjunto completo de funções aritméticas;
- f) um conjunto de funções internas à linguagem com diversas funcionalidades;
- g) uma sintaxe “exótica”, semelhante em alguns aspectos à “C”, mas bem diferente em outros;
- h) um filosofia de fazer o trabalho rápido e não de forma elegante;
- i) o uso de expressões regulares;
- j) fácil acesso a qualquer banco de dados.

### 3.5.1.2 CGI

*Common Gateway Interface* (CGI) é o método usado para permitir a interação entre o servidor WWW e outros programas executados no sistema. Foi desenvolvido originalmente para servidores WWW CERN e NCSA, para a plataforma UNIX (OTSUKA, 2007). Consiste numa importante tecnologia que permite gerar páginas dinâmicas, permitindo a um navegador passar parâmetros para um programa alojado num servidor *web*. Assim, designam-se por *scripts* CGI os pequenos programas que interpretam esses parâmetros e geram a página depois de os processar (WIKIPEDIA, 2007c). Pode apreciar-se o funcionamento conceitual na fig. 3.7.



Fonte: (OTSUKA, 2007)

**Figura 3.7** – Funcionamento do CGI

O CGI foi concebido como o culminar de discussões por especialistas durante os

primórdios da Internet, nomeadamente entre Rob McCool, John Franks, Ari Luotonen, George Phillips e Tony Sanders (WIKIPEDIA, 2007c).

Embora a linguagem tipicamente associada aos CGI seja o PERL, o CGI foi concebido de forma a ser independente da linguagem utilizada. Atualmente tecnologias como ASP.NET ou PHP continuam a utilizar a especificação (WIKIPEDIA, 2007c).

### 3.5.1.3 BASH

O BASH é um interpretador de comandos, uma espécie de tradutor entre o SO e o usuário, normalmente conhecido como *shell*. Permite a execução de seqüências de comandos direto no *prompt* do sistema ou escritas em arquivos de texto, conhecidos como *shell scripts* (WIKIPEDIA, 2007b).

O BASH é o *shell* desenvolvido para o projeto GNU, da *Free Software Foundation*, que se tornou padrão nas várias distribuições Linux. Pode ser usado também com outros sistemas operacionais, como o Unix. É compatível com o *Bourne shell* (sh), incorporando os melhores recursos do *C shell* (csh) e do *Korn Shell* (ksh) (WIKIPEDIA, 2007b).

### 3.5.1.4 RRD Tool

RRD é a sigla para *Round Robin Database* (base de dados round-robin). O RRD é um sistema para armazenar e mostrar dados em série obtidos em um determinado período de tempo (cpu, banda de rede, temperatura da máquina, etc). Os dados são armazenados de maneira bastante compacta e não aumentam com o tempo (por isso que o banco é dito "circular"). O RRDTOOL também é capaz de gerar gráficos a partir desses dados (POP-PR, 2004).

### 3.5.1.5 MySQL

Atualmente existem vários SGBD no mercado, sejam eles proprietários ou *opensource*. Dentre muitos, o MySQL tem se destacado e se disseminado entre a comunidade de desenvolvedores, principalmente para aplicações em ambiente *web*.

#### 3.5.1.6 Poseidon

Esta é uma ferramenta para modelagem UML desenvolvida pela Gentleware AG, para a modelagem de projetos de *software*. Possui uma interface intuitiva com características de *zoom*, navegação rápida, impressão, arrastar e soltar, pré-visualização e edição de códigos, geração de documentação e exportação de diagramas. Para este trabalho, somente o módulo que permite o desenvolvimento dos casos de uso foi utilizado. A ferramenta permite também realizar a engenharia reversa para Java, geração automática de documentação, e total controle de “Desfazer” e “Refazer” (GENTLEWARE, 2007).

#### 3.5.1.7 DBDesigner

O DBDesigner versão 4 é um *software* de desenho de bancos de dados que integra a modelagem do banco, desenho, criação e manutenção dentro de um sistema só. Encontra-se disponível para as plataformas Microsoft Windows 2k/XP e GNU/Linux para KDE ou Gnome. É gratuito, disponível para ser baixado da Internet sem custo algum e acompanha o código fonte através da licença GNU GPL (TOOLS, 2007).

#### 3.5.1.8 ActiveState Perl Dev-Kit

O ActiveState Perl Dev-Kit era uma solução gratuita de ambiente IDE para programação em Perl fornecida pela ActiveState Software. Atualmente, o ActiveState Perl Dev-Kit tornou-se o ActivePerl Pro Studio em versão comercial e inclui o ambiente IDE “Komodo” em conjunto com o PDK Pro Pack, o entorno de desenvolvimento ActiveState Perl Development Kit. O ambiente “Komodo” oferece um avançado suporte para não somente Perl, mas também para PHP, Python, Ruby e Tcl no Linux, Mac OS X, Solaris e Windows (ACTIVESTATE, 2007).

### 3.5.2 Operacionalidade da implementação

A fim de organizar a coexistência entre o código Perl usado pelo sistema e o HTML apresentado pelo CGI ao *browser*, foram criadas funções que imprimem em blocos as diferentes partes da página *web*. Isto permite uma organização mais limpa do código fonte,

separando de maneira mais prolixa o que é impresso em termos de *layout web* ao *browser*, do que tem a ver com o próprio sistema. As funções criadas são *show\_httpheaders()*, *show\_htmlheaders()*, *show\_menu()*, *fechar\_html()*, *fechar\_footer()*. Estas funções se encontram no arquivo “*/var/clusterbfd/config.pl*”, fornecendo os recursos necessários para a impressão de código HTML, senhas, dados sobre a conexão e rotinas de consultas no banco de dados. Também engloba os componentes necessários para o sistema. Cada código Perl-CGI usa esse arquivo a fim de centralizar as informações usadas no sistema *web*. A fig. 3.8 mostra um exemplo de como o uso dessas funções facilita o desenvolvimento do código em Perl sem ficar “poluído” pelos *tags* HTML.

```
#!/usr/bin/perl
require '/var/clusterbfd/config.pl'; # arquivo onde ficam as funcoes e as variaveis

# [...]
#
# -----
# sessao e cabecalhos html
# -----
    &show_httpheaders();
    &show_htmlheaders();
# -----
# gerar menu usuario
# -----
    &show_menu();
# -----
# CONTEUDO ESPECIFICO PARA ESTE MODULO
# -----
if ($errormessage || $success) {
    my $message="";
    # [...]
}
# -----
    &fechar_footer();
    &fechar_html();
```

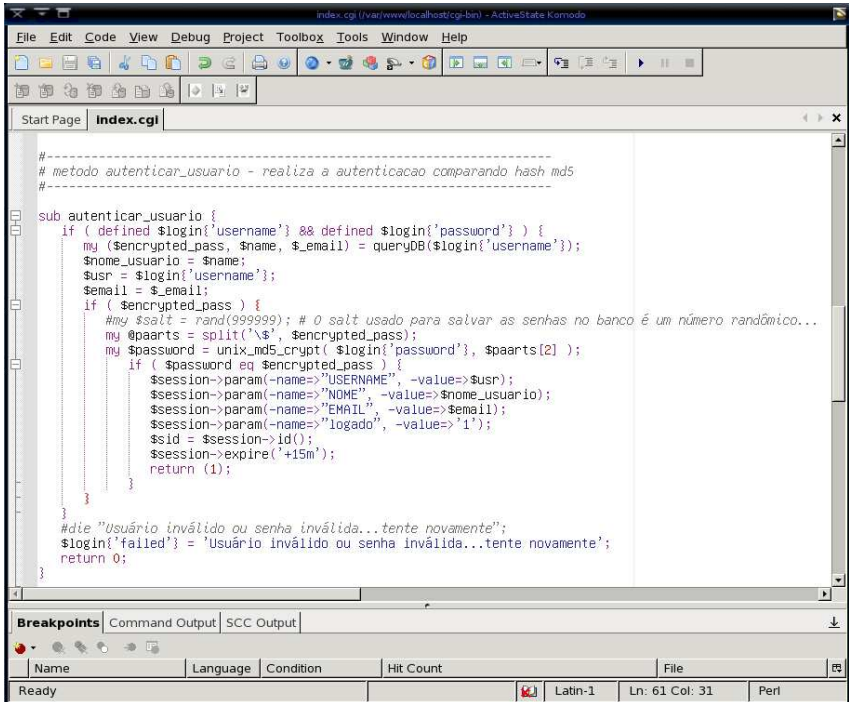
**Figura 3.8** – Código exemplo demonstrando o uso das funções HTML em conjunto com o Perl

Seguindo o exemplo da fig. 3.8, a primeira função a ser utilizada é *show\_httpheaders()*, onde são impressos através da camada de aplicação os cabeçalhos necessários para iniciar uma conexão cliente-servidor. É a primeira resposta que o *browser* obtém do CGI assim que iniciar uma conexão usando o protocolo HTTPS, indicando que irá ser fornecido conteúdo HTML. São encaminhados através da função *show\_htmlheaders()* os tags HTML a respeito do documento (HTML 4.01) como título da página *web*, conteúdo do CSS, autor do documento e codificação. A função *show\_menu()* complementa a página imprimindo o *layout* do menu e a barra de opções. A seguir, se implementa o código e as rotinas necessárias para o funcionamento do sistema. Para fechar o *layout* usa-se a função *fechar\_footer()*. Por último, a função *fechar\_html()* encerra o código HTML.

Quando a página principal do sistema é aberta é exibida a tela de *login*. Aqui é fornecido usuário e a senha para acessar o sistema. A rotina responsável usa os recursos do método orientado a objetos “CGI::Session” para registrar uma sessão válida caso o usuário informe corretamente a senha. Um dado importante a salientar é que as senhas informadas pelo “admin”

na hora de realizar o cadastro dos usuários são salvas no banco de dados totalmente criptografadas. Essas senhas são geradas através de um valor randômico, que combinado com a senha escolhida, gerará um *hash* criptográfico MD5 que será salvo no banco para posterior consulta. Usando a função “Crypt::PasswdMD5” e comparando o conteúdo do “*salt*” armazenado no banco entre \$ (por ex. \$1\$salt\$5JH9p/tWH61oI13vRDseE/), a rotina gera uma nova senha temporária usando a senha informada pelo usuário com o *hash* que foi utilizado no momento da criação. Isto se deve ao fato que o MD5 é um *hash* de uma via só, ou seja, não existe maneira de ser decifrado. Por tanto, se a senha temporária for idêntica à senha salva na terceira coluna do *hash*, o usuário terá informado corretamente sua senha e lhe será permitido o acesso ao sistema.

A rotina que estabelece a consulta pode ser apreciada na fig. 3.9. Após ter a senha validada, é gerada a sessão que o usuário irá utilizar acompanhado dos dados da conta no servidor e do tempo máximo de 30 minutos antes de ter a sessão expirada.



```

index.cgi (/var/www/localhost/cgi-bin) - ActiveState Komodo
File Edit Code View Debug Project Toolbox Tools Window Help
Start Page index.cgi
#-----
# metodo autenticar_usuario - realiza a autenticacao comparando hash md5
#-----
sub autenticar_usuario {
    if ( defined $login{username} && defined $login{password} ) {
        my ($encrypted_pass, $name, $email) = queryDB($login{username});
        $nome_usuario = $name;
        $usr = $login{username};
        $email = $email;
        if ( $encrypted_pass ) {
            my $salt = rand(999999); # 0 salt usado para salvar as senhas no banco é um número randômico...
            my @paarts = split('\$', $encrypted_pass);
            my $password = unix_md5_crypt( $login{password}, $paarts[2] );
            if ( $password eq $encrypted_pass ) {
                $session->param(-name=>"USERNAME", -value=>$usr);
                $session->param(-name=>"NOME", -value=>$nome_usuario);
                $session->param(-name=>"EMAIL", -value=>$email);
                $session->param(-name=>"logado", -value=>'1');
                $sid = $session->id();
                $session->expire('+15m');
                return (1);
            }
        }
        #die "usuário inválido ou senha inválida...tente novamente";
        $login{failed} = 'Usuário inválido ou senha inválida...tente novamente';
        return 0;
    }
}
Breakpoints Command Output SCC Output
Name Language Condition Hit Count File
Ready Latin-1 Ln. 61 Col. 31 Perl

```

**Figura 3.9** – Rotina de consulta da senha criptografada no banco

Sempre que for submetido um processo ao *cluster*, o sistema *web* poderá usar dois *scripts* diferentes para repassá-lo ao CFX-Solver. O primeiro *script* Perl usa-se caso o *job* a ser simulado utilize um arquivo opcional de resolução tipo “.res” ou “.bak”, esse *script* é o “/var/cluster/cfd/rodar\_job\_res.pl”, caso contrário o *script* a ser usado vai ser o “/var/cluster/cfd/rodar\_job.pl”. Para o primeiro *script*, o sistema *web* fornece quatro “argumentos” (valores) que são representados como \$ARGV[0], \$ARGV[1], \$ARGV[2] e \$ARGV[3], atribuídos às variáveis \$pasta, para informar qual a pasta onde irá ser executado o processo de

resolução, *\$arquivo* para identificar o arquivo CFX, *\$arquivo\_res* para identificar o arquivo com resultados numéricos e *\$codigo\_banco* para gerenciar as informações salvas no banco na tabela *jobs*. Já para o segundo *script*, são repassados somente os três primeiros valores *\$ARGV[0]*, *\$ARGV[1]*, *\$ARGV[2]*, atribuídos às variáveis *\$pasta*, *\$arquivo* e *\$codigo\_banco*. Os *scripts* são utilizados tanto quando os *jobs* são agendados quanto executados. A fig. 3.10 mostra o código fonte do *script* Perl “*rodar\_job.pl*”.

```
my $parallel = "$pasta/settings";
open(FILE, "< $parallel") or die "Impossível abrir $parallel : $!";
@linhas_arq = <FILE>;
close(FILE);

#delete @ENV{qw(IFS CDPATH ENV BASH_ENV)};
$ENV{'PWD'} = '$pasta';
$ENV{'HOME'} = '$pasta';
$ENV{'SHELL'} = "/bin/bash";

$exec_command = "cd $pasta && sudo su ifc -c '/opt/CFX/CFX-10.0/bin/cfx5solve -def $arquivo -par-dist @linhas_arq' ";
system("$exec_command");

$exit_value = $? >> 0;
if ($exit_value) {
    $error_msg = "Error (exit value: $exit_value)!!";
    $erro = "cd $pasta && touch ERRO && echo $arquivo >> ERRO && echo $error_msg >> ERRO";
    system("$erro");
    notifica_banco($codigo_banco, "erro");
} else {
    $ok = "cd $pasta && touch OK";
    system("$ok");
    notifica_banco($codigo_banco, "finalizado");
}

exit 0;
```

**Figura 3.10** – Código fonte do *script* Perl *rodar\_job.pl*

No caso do agendamento do *job*, o sistema utiliza o componente “Config::Crontab”, que providencia uma interface orientada a objetos para tratar arquivos do tipo “crontab”. O arquivo “crontab” é usado pelo serviço “cron” do Linux. Trata-se de uma ferramenta que permite programar a execução de comandos e processos de maneira repetitiva, ou apenas uma única vez como no nosso caso. No quadro 3.9 tem-se um exemplo de uma entrada tipo “cron” gerada pelo sistema *web* após o agendamento de um trabalho chamado “Benchmark” a ser executado na data 4/06 às 17:30hs.

```
PWD=/var/clusterbfd/storage/javito/Benchmark-4-06_17-30
30 17 4 06 * perl /var/clusterbfd/rodar_job.pl Benchmark-4-06_17-30
Benchmark-4-06_17-30/Benchmark.def Benchmark-4-06_17-30
```

**Quadro 3.9** – Entrada “cron” gerada pelo processo de agendamento do sistema

Seguindo o exemplo anterior, o *script* que executa o trabalho agendado é “*rodar\_job.pl*”, e sempre verifica na pasta do trabalho submetido o arquivo “*settings*”, onde se encontram os dados sobre os recursos a serem usados (quantidade de nós).

A atualização dos dados sobre o nível de processamento usado pelo *cluster* é feita a cada minuto em cada CPU através do *script* em BASH e a opção “*/var/clusterbfd/gerar\_update.sh update*”. Este *script* encontra-se instalado no *master* e em cada nó.



A geração dos gráfico do estado de processamento de cada nó do *cluster*, usando os dados previamente coletados através da ferramenta “RRD Tool”, são feitos em intervalos de 5 minutos. Os gráficos são atualizados através do *script* em BASH “*atualiza\_graficos.sh*”. Parte do *script* pode ser visto na fig. 3.11.

```
#!/bin/bash
# Script para atualizar os graficos sobre processamento do cluster e nós.
# -----
# Obviamente que o primeiro vai ser o master... entao nem vou usar o ssh ;-)

/var/clusterbfd/gerar_graficos.sh update
sleep 5
# gera o grafico...
/var/clusterbfd/gerar_graficos.sh grafico
sleep 5

#verifica a quantidade de nós no cluster...
for i in $(/bin/cat /var/clusterbfd/settings | sed -e 's/master/ /g' | sed -e 's/\,/ /g' ); do
# atualiza os dados do sistema para a base RRD e aguarda 5 segundos...
ssh $i '/var/clusterbfd/gerar_graficos.sh update'
sleep 5
# gera o grafico...
ssh $i '/var/clusterbfd/gerar_graficos.sh grafico'
sleep 5
# copia o grafico gerado para a servidor master na pasta dos graficos...
scp $i:/var/clusterbfd/graficos/ $i-load.png /srv/projeto-cluster/htdocs/graficos/
scp $i:/var/clusterbfd/graficos/ $i-cpu.png /srv/projeto-cluster/htdocs/graficos/
;
done
```

**Figura 3.11** – Código fonte do *script* BASH para atualizar os gráficos do estado do CPU

As primeiras linhas do *script* atualizam os dados e os gráficos para o nó *master*. O *loop* do “for” conta cada máquina cadastrada no sistema, tira as vírgulas e o cadastro do nó *master*, e atualiza usando o relacionamento de chaves criptografadas, os dados em cada nó e a criação dos gráficos. Por último, copia os resultados dentro da pasta do sistema *web* para ser posteriormente visualizado.

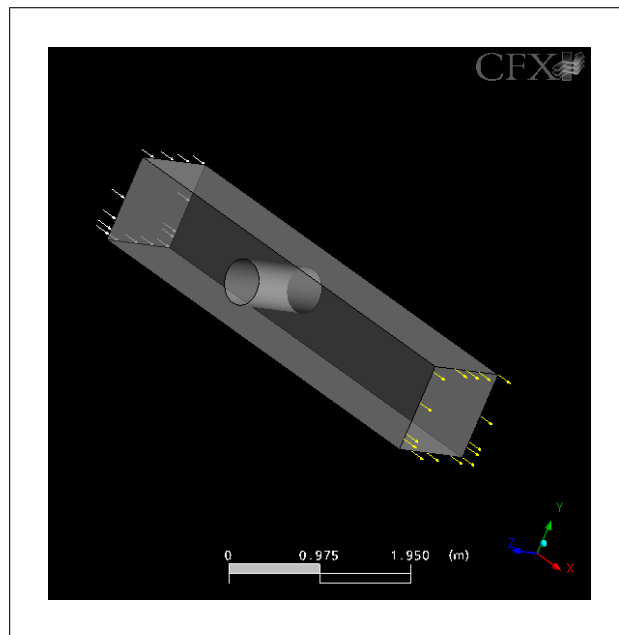
A opção “*update*” repassada ao *script* BASH atualiza o banco “RRD” com os dados estatísticos do uso do CPU na máquina onde for executado. A opção “*grafico*” faz com que o “RRDTool” use esses dados estatísticos para gerar um gráfico conforme for passando o tempo.

### 3.6 UTILIZAÇÃO DO SISTEMA

Esta seção aborda a utilização do sistema através de um projeto de cálculo CFX. Envolve:

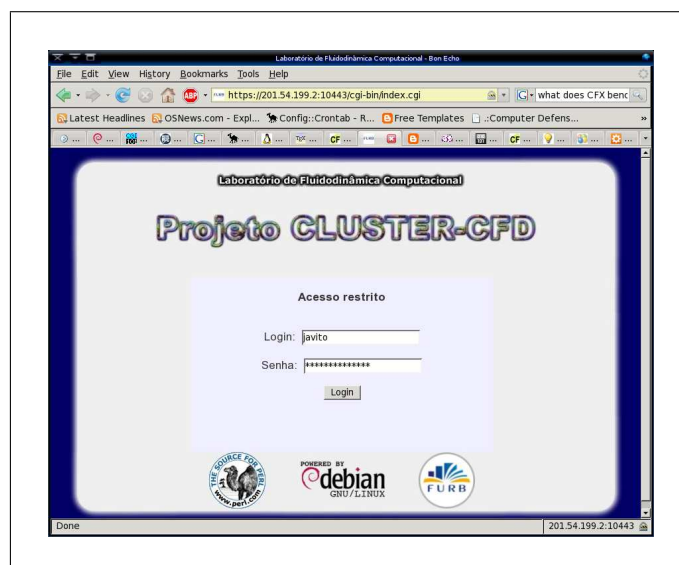
- acesso ao sistema através da interface *web* via HTTPS;
- alteração da senha e o nome do usuário;
- agendar um processamento no *cluster*;
- realizar uma alocação de recursos usando somente dois nós e fazer a submissão do *job* para ser processado no *cluster*;
- salvar os resultados obtidos.

Como projeto teste de simulação, é utilizado um escoamento através de um cilindro, considerado um experimento clássico na mecânica dos fluidos. Ele consiste de uma tubulação de seção quadrada com um obstáculo de seção cilíndrica posicionado a 1,7 m da entrada. O cilindro tem 0,5 mts e o comprimento total do duto é de 5 m. O fluido a ser usado foi “ar” tipo isotérmico, entrando no sistema com um velocidade de entrada de 5 m/s. A figura fig. 3.12 mostra o domínio físico usado para a simulação dentro do *cluster beowulf* construída com o ANSYS ICEM, parte da *suite* de ferramentas para engenharia mecânica da ANSYS.



**Figura 3.12** – Simulação “teste” a ser executada dentro do *cluster beowulf*

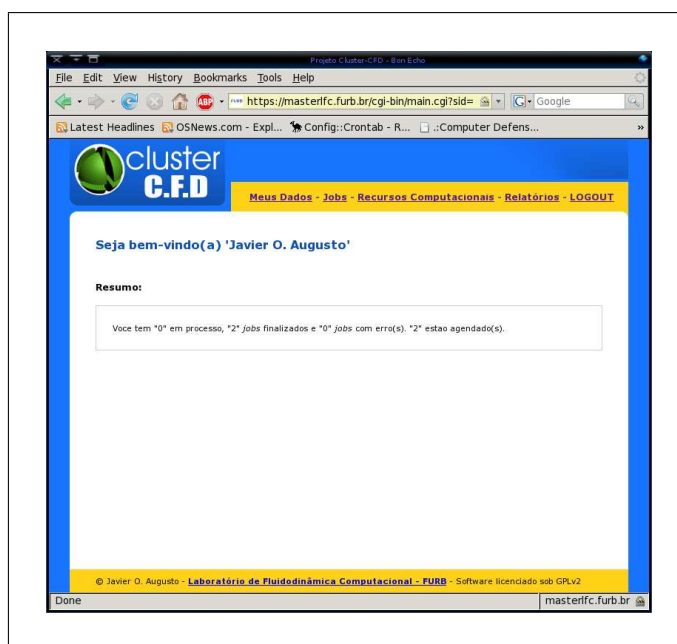
Ao acessar o sistema *web* através do serviço de HTTPS, a tela de login representada na fig. 3.13 será exibida.



**Figura 3.13** – Tela de acesso ao sistema

Para realizar o acesso ao sistema, o administrador deve inicialmente ter cadastrado o usuário, contendo como informações essenciais “nome do login” e “senha de acesso”. Este cadastramento é possível somente se o usuário que está acessando o sistema é o “admin”. Se for, será exibido um menu específico onde é possível cadastrar um usuário definindo nome completo, senha, e-mail de contato e o *login* a ser utilizado para acessar o sistema.

Em seguida, o usuário acessa ao menu principal “*main.cgi*” e recebe um resumo dos *jobs* dentro do *cluster* conforme apresentado na fig. 3.14.



**Figura 3.14** – Tela principal do sistema *web*

Na barra superior temos as opções de “Meus dados” para alteração dos dados do usuário, “Jobs” para o agendamento e submissão dos *jobs*, “Recursos Computacionais” para a alocação dos recursos e visualização do estado de execução do *cluster*, “Relatórios” para apresentar relatórios sobre os trabalhos submetidos no *cluster* e “Logout” para sair do sistema.

Na fig. 3.15 aprecia-se a tela de alteração de dados pessoais, senha e *e-mail*. Existem algumas limitações impostas pelo sistema, por exemplo na hora de informar uma nova senha o comprimento mínimo terá de ser de 6 caracteres e o *e-mail* fornecido terá de respeitar o formato “usuario@domínio”. Caso retorne algum erro, o sistema exibe uma nota informando qual o erro apresentado, caso contrário apresenta uma mensagem indicando que as mudanças foram salvas.

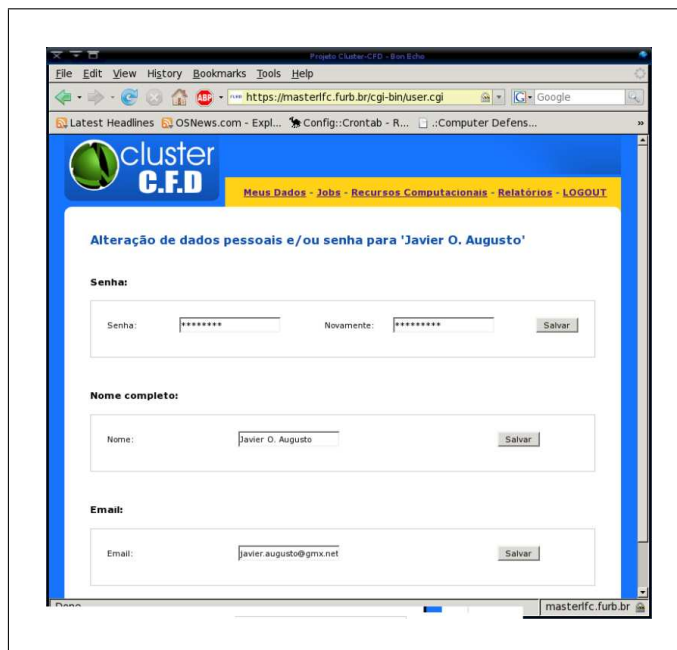
The screenshot shows a web browser window with the URL <https://masterfc.furb.br/cgi-bin/user.cgi>. The page title is "Alteração de dados pessoais e/ou senha para 'Javier O. Augusto'". The interface includes a navigation bar with "Meus Dados", "Jobs", "Recursos Computacionais", "Relatórios", and "LOGOUT". The main content area has three sections: "Senha:" with fields for "Senha:" (containing "\*\*\*\*\*") and "Novamente:" (containing "\*\*\*\*\*"), a "Salvar" button, and a "Nome completo:" section with a "Nome:" field containing "Javier O. Augusto" and a "Salvar" button. The "Email:" section has an "Email:" field containing "javier.augusto@gmx.net" and a "Salvar" button.

Figura 3.15 – Tela de alteração de dados pessoais, senha e e-mail

Ao clicar na opção “Jobs”, visto na fig. 3.16, o sistema exibe a tela de submissão de trabalhos a serem processados no *cluster beowulf*. Nela deverá ser informado o arquivo CFX com extensão “.def” e opcionalmente um arquivo com dados numéricos pré-processados pelo *cluster* ou pelo CFX-Solver, normalmente com as extensões “.ref” ou “.bak”. Para realizar o agendamento no *cluster* é necessário fornecer uma data válida. A data não pode ser inferior à hora atual, e será agendada sempre dentro do mês em transcurso.

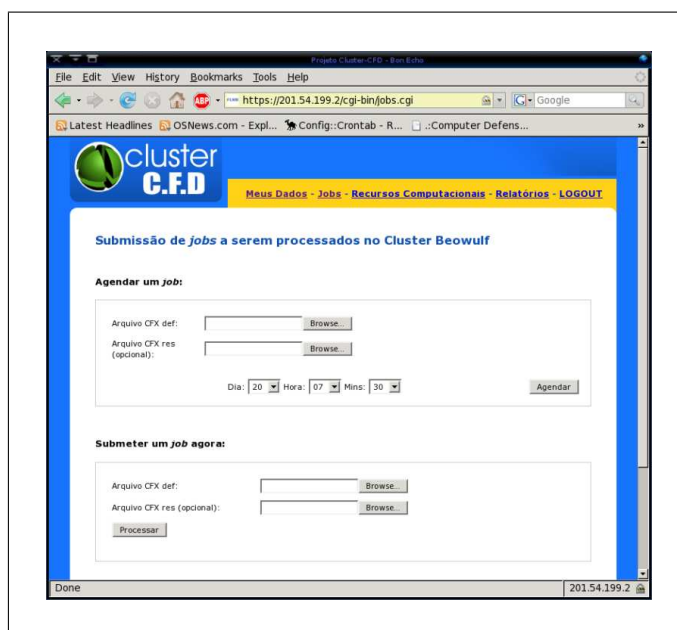
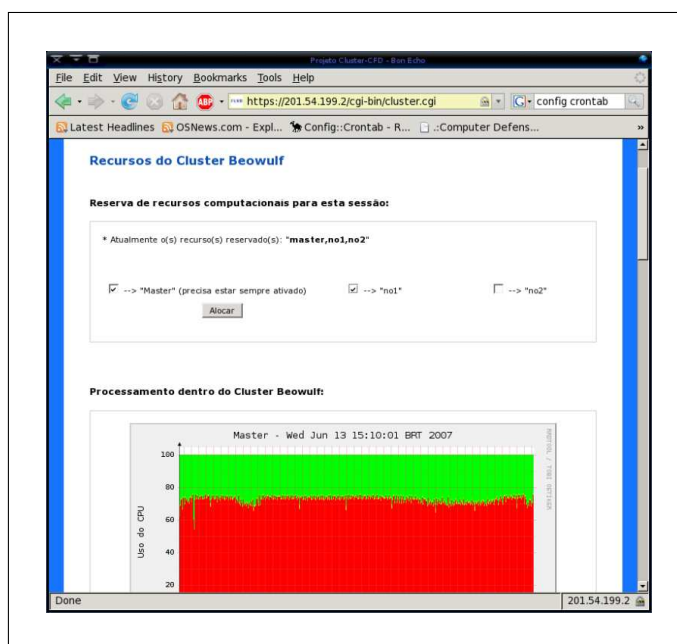
The screenshot shows a web browser window with the URL <https://201.54.199.2/cgi-bin/jobs.cgi>. The page title is "Submissão de jobs a serem processados no Cluster Beowulf". The interface includes a navigation bar with "Meus Dados", "Jobs", "Recursos Computacionais", "Relatórios", and "LOGOUT". The main content area has two sections: "Agendar um job:" with fields for "Arquivo CFX def:" and "Arquivo CFX res (opcional):", each with a "Browse..." button, and a date selection section with "Dia:" (20), "Hora:" (07), and "Mins:" (30) dropdowns, and an "Agendar" button. The "Submeter um job agora:" section has fields for "Arquivo CFX def:" and "Arquivo CFX res (opcional):", each with a "Browse..." button, and a "Processar" button.

Figura 3.16 – Tela de submissão de jobs

Após informar-se o arquivo CFX e uma data válida para o agendamento do processo, o usuário pressiona o botão “Agendar” e o sistema criará uma entrada do tipo “*crontab*” no nó “*master*”.

Cada vez que é feito o acesso ao sistema, automaticamente usa-se toda a capacidade do *cluster* a não ser que previamente tenha sido especificado algum recurso na opção “Recursos Computacionais”. No exemplo anterior, o agendamento usará todos os recursos do *cluster* pois não tinham sido informado quais eram os nós a serem usados.

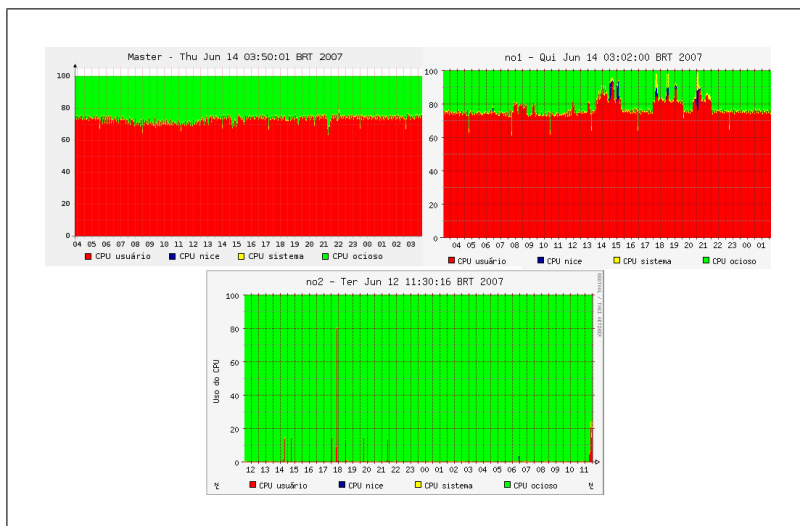
Ao clicar na opção de “Recursos Computacionais”, visto na fig. 3.17, o sistema exibe uma tela informando quais recursos estão sendo alocados para a sessão em uso e o estado de processamento dos nós que fazem parte do *cluster*. De acordo com o estado de processamento dos nós do *cluster* o usuário poderá alocar os nós ociosos que não estão sendo usados. Após realizar a seleção dos nós conforme o estado de execução do *cluster*, o usuário pressiona o botão “Alocar” e os recursos serão alocados para essa sessão.



**Figura 3.17** – Tela de alocação de recursos

Logo, voltando para a opção de “Jobs”, deverá ser informado em “Submeter um *job* agora” o arquivo CFX que será executado nesse preciso momento dentro do *cluster* com os recursos que anteriormente tinham sido reservados. Após indicar o arquivo, o usuário pressiona o botão “Processar” e o trabalho será processado no *cluster* usando os recursos previamente alocados para essa sessão. O estado da execução pode ser acompanhado pelo gráfico estatístico de uso do CPU dentro do *cluster* na opção de “Recursos Computacionais”. Voltando ao menu principal clicando no gráfico superior “Cluster CFD”, também se tem um resumo atualizado do estado dos processos submetidos no *cluster*.

Na fig. 3.18, aprecia-se um gráfico que mostra o processamento distribuído pelo sistema usando somente os recursos “master” e “no1”. Pode-se apreciar que o “no2” está em estado ocioso, pois não foi alocado para executar nossa simulação.



**Figura 3.18** – Tela do estado de processamento do *cluster*

Uma vez finalizado o processamento da simulação, procede-se a salvar os resultados obtidos pelo *cluster*. Este processo é feito acessando a opção “Relatórios”. A partir dela é possível ter uma lista com todos os trabalhos que foram salvos no sistema. O usuário poderá apagar estas pastas clicando no botão representado por um “lixeiro” ou gerar um arquivo compactado “.tar.gz” e baixá-lo no seu computador para posterior análise clicando no “disquete”.

### 3.7 RESULTADOS E DISCUSSÃO

Nesta seção são apresentados os resultados obtidos com a realização deste trabalho.

A quantidade de recursos computacionais necessários às simulações dos métodos numéricos aplicados a CFD, no geral, é grande. Por esse motivo, a exploração de técnicas aplicadas à fluidodinâmica computacional tem crescido exponencialmente nos últimos 20 anos. Em geral, as novas técnicas visam aperfeiçoar as soluções que demandam grande poder de processamento, reduzindo o tempo despendido pelas tarefas. Estes cálculos dependem diretamente do uso do processamento massivamente paralelo através dos *clusters beowulf*,

A tabela 3.2 mostra os resultados obtidos com os testes práticos, onde se pôde verificar o ganho no desempenho de processamento do *cluster*. Como a estrutura usada no *cluster* não é uniforme, ou seja, não possui uma única configuração de CPU e RAM, a resolução do problema não escala linearmente.

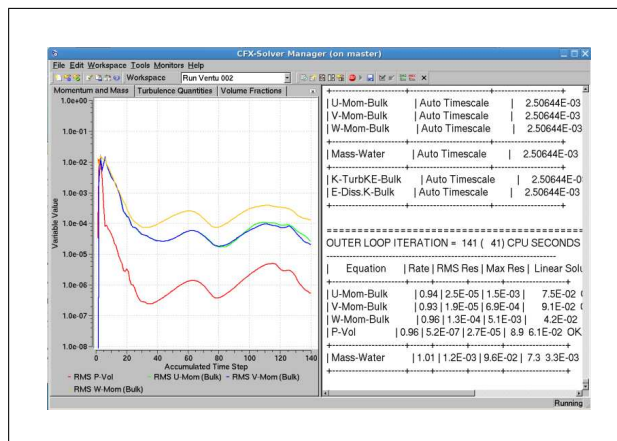
**Tabela 3.2** – Tabela de tempo de simulação obtidos no *cluster*

quantidade de nós	tempo total de simulação
1	4 mins 33 segs
2	3 mins 18 segs
3	2 mins 28 segs

Analisando os resultados apresentados na tabela 3.2, confirma-se mais uma vez a utilização de *clusters* como uma alternativa, que realmente proporciona alto poder computacional e baixo custo, pois, como no primeiro teste, houve uma melhora de desempenho conforme foram adicionados novos nós. Por exemplo, o tempo gasto para o simulação CFX, paralelamente, utilizando três nós foi aproximadamente 50% menor que tempo gasto para a execução do processamento de forma serial usando somente os recursos do nó *master*.

A solução numérica das equações que descrevem o comportamento dos sistemas físicos não são soluções exatas, e sim soluções aproximadas. A obtenção dessas soluções aproximadas requer o uso de estratégias numéricas para solução de sistemas de equações (algébricas) não-lineares. No entanto, sabe-se da dificuldade de obter-se uma solução desse tipo de problema. Assim, se o problema não estiver bem estruturado há risco de “divergência” da solução numérica. Portanto, o fato do uso de uma estrutura computacional de alto desempenho não necessariamente vai garantir o resultado esperado. Desta maneira, é essencial que o usuário tenha a capacidade de interpretar esses resultados e assim, aprimorá-los para obter as simulações esperadas. Também, dependendo da simulação a ser processada no *cluster*, pode demorar semanas ou até meses para chegar ao resultado, mesmo assim nunca se tem certeza se o resultado irá ser o esperado ou não.

A fig. 3.19 mostra uma imagem da tela de acompanhamento da solução conforme é processada no *cluster* pela ferramenta CFX-Post, onde o usuário pode acompanhar a evolução da solução a partir dos resíduos nas aproximações numéricas que são apresentadas no gráfico.

**Figura 3.19** – Acompanhamento da evolução da solução a partir dos resultados obtidos no *cluster*

---

No endereço “<http://masterlfc.furb.br/works/>” encontram-se alguns dos resultados das simulações feitas utilizando o sistema web de gerenciamento de aplicações CFX pela equipe do LFC/LDP/LVV da FURB. Estas animações foram geradas a partir do processamento obtido no *cluster* e posteriormente interpretados pela ferramenta ANSYS CFX-Post.



## 4 CONCLUSÃO

A partir deste trabalho foi possível verificar que a utilização da fluidodinâmica computacional em conjunto com a computação de alto-desempenho já é uma realidade. Recentemente a equipe do LFC/LDP/LVV da FURB receberam um importante prêmio em nível nacional na área de pesquisa e desenvolvimento sobre biocombustíveis. Os resultados apresentados nessa pesquisa foram obtidos com a estrutura computacional em paralelo montada para o desenvolvimento deste trabalho. A adoção de *clusters* como recursos computacionais para a obtenção de processamento de alto desempenho têm contribuído nas pesquisas e desenvolvimento de novos trabalhos em CFD.

Apesar de o LFC ter uma infra-estrutura precária comparada com as grandes universidades, como Unicamp ou PUC-Rio, um dos pontos fortes da plataforma montada é a grande capacidade de escalabilidade, interoperabilidade e baixo custo, o que torna ideal para continuar acrescentando recursos computacionais com baixos valores de investimentos. Os testes efetuados demonstraram a total viabilidade de implementação deste projeto, mas existe uma enorme dificuldade para encontrar recursos financeiros para acompanhar a demanda computacional requerida pelo laboratório. Nos dias atuais, as indústrias estão cada vez mais competitivas e, para se manter em um mercado como este, são necessários vários investimentos em pesquisa e desenvolvimento de novas tecnologias.

Além da obtenção de um ambiente de processamento paralelo, um dos objetivos desse trabalho foi desenvolver uma ferramenta que permitisse ao LFC o gerenciamento das aplicações CFX executadas no *cluster beowulf* através de uma interface amigável via *web*.

Toda a pesquisa e desenvolvimento realizados durante a implementação deste trabalho trouxeram bastante conhecimento não só sobre as tecnologias de *clusters*, mas também sobre o funcionamento da fluidodinâmica computacional em conjunto com as ferramentas da ANSYS, conceituadas a nível mundial. A adoção de novas tecnologias por parte do LFC, apresenta uma oportunidade de atuação a nível nacional em resolução de problemas em CFD, trazendo avanços à região.

A realização deste trabalho abre margem para diversos outros na área de computação paralela e de alto desempenho, possibilitando a implementação de novas aplicações paralelas, bem como a utilização e estudo de aplicações existentes que exigem um ambiente de alto desempenho.

## 4.1 LIMITAÇÕES DO SISTEMA

O sistema desenvolvido apresenta uma limitação no processamento em paralelo usando o MPI (ou MPICH). O ANSYS CFX depois da versão 10.0, sempre que for executado em ambiente de *cluster* usando o sistema GNU/Linux produz erros de segmentação causado pela queda de sincronismo de páginas da biblioteca MPI, limitação esta imposta pelo *software*. Isto encontra-se documentado. A ANSYS recomenda o uso da biblioteca MPI (ou MPICH) somente em plataformas certificadas com recursos de processamento centralizado tipo NUMA, por exemplo equipamentos SGI, ou quando for usado com a plataforma Microsoft Windows.

## 4.2 TRABALHOS FUTUROS

Neste item são apresentadas algumas idéias e sugestões que podem ser utilizadas em trabalhos futuros, que seguindo a linha deste trabalho têm o seu foco voltado para a utilização de soluções de *clusters beowulf* no processo de auxílio para a CFD.

Apesar dos bons resultados de desempenho já mostrados, ainda existem possibilidades de melhorias no desenvolvimento de códigos como a investigação e implementação de técnicas de otimização dos algoritmos usados no sistema.

Como sugestão para trabalhos futuros pode-se citar estudos mais aprofundados em como estender o sistema *web* para outras aplicações além das simulações CFX em *cluster*, tentando identificar outras soluções em CFD para processamento em HPC. Também, é interessante avaliar o uso de escalonadores tipo o “BProc” ou sistemas do tipo *Portable Batch System* (PBS), afim de criar uma única visão dos processos que estão rodando no *cluster*, oferecendo um mecanismo para gerência dos processos existentes em todos os nós através de uma única interface.

Outra sugestão seria a execução de testes de *benchmarks* para a avaliação da performance do conjunto com variação na quantidade de nós utilizado.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ACTIVESTATE. **ActivePerl Pro Studio**. 2007. Paginação Irregular. Acessado em 25 de abril de 2007. Disponível em: <<http://www.activestate.com/Products/ActivePerlProStudio/>>.
- AEA Technology. **CFX-TASCflow Theory documentation, Version 2.11.1, Chapter 4: Turbulence Closure Models**. 2000. Paginação Irregular. Acessado em 13 de março de 2007. Disponível em: <<http://www.software.aeat.com/cfx/>>.
- ALVAREZ, Rubén. **O que é Perl**. 2007. Paginação Irregular. Acessado em 19 de abril de 2007. Disponível em: <<http://www.criarweb.com/artigos/222.php?manual=11>>.
- ANQUETIL, Nicolas. **Desenvolvimento de Software Orientado a Objetos**. 2002. Paginação Irregular. Acessado em 16 de maio de 2007. Disponível em: <<http://www.ucb.br/ucbtic/mgcti/paginapessoalprof/Nicolas/Disciplinas/UML/index.html>>.
- ANSYS Company. **CFX-5 Solver Theory Manual CFX Ltd. Version 5.7**. Oxfordshire, UK: ANSYS Company/CFX Ltd., 2003.
- BUYYA, Rajkumar; BAKER, Mark. Emerging technologies for multicluster/grid computing. In: **2001 IEEE International Conference on Cluster Computing (CLUSTER 2001), 8-11 October 2001, Newport Beach, CA, USA**. Newport Beach, California: IEEE Computer Society, 2001. p. 457–458.
- CGICLUBE.NET. **Perl e CGI**. 2007. Paginação Irregular. Acessado em 21 de abril de 2007. Disponível em: <<http://cgiclube.cidadeinternet.com.br/?area=operl>>.
- CHAN, Steven. **In-Depth: Load-Balancing E-Business Suite Environments**. 2006. Paginação Irregular. Acessado em 6 de dezembro de 2006. Disponível em: <<http://blogs-oracle.com/schan/newsItems/departments/optimizingPerformance/2006/06/16>>.
- CSERI; NASA. **Solving complex science and engineering problems using high performance computers**. 2007. Paginação Irregular. Acessado em 2 de abril de 2007. Disponível em: <<http://www.cs.mtu.edu/~steve/CSERI/>>.
- DEBIAN PROJECT. **Sobre o Debian**. 2007. Paginação Irregular. Acessado em 18 de abril de 2007. Disponível em: <<http://www.debian.org/index.pt.html>>.
- GENTLEWARE. **Poseidon for UML Community Edition**. 2007. Paginação Irregular. Acessado em 18 de abril de 2007. Disponível em: <<http://www.gentleware.com/uml-software-community-edition.html>>.
- HAMILL, Nathalie. **CFD comes of age in the CPI**. Pune, India: Chem Engng, 1996. 65-72 p.
- JOSHI, J. B.; RENADE, V. V. **Computational fluid dynamics for designing process equipment: expectations, current status, and path forward**. UK: Industrial & Engineering Chemistry Research, Vol.42, No.6, 2003. 1115-1128 p.

- KOWALTOWSKI, Tomasz. **John von Neumann: Suas Contribuições à Computação**. Campinas: Departamento de Ciências da Computação, Universidade de Campinas, Dez 1995. 18 p.
- LOURENÇO, João. **Introdução ao PVM**. 2002. Paginação Irregular. Acessado em 15 de abril de 2006. Disponível em: <[http://www-asc.di.fct.unl.pt/~jml/LEI/SCPD/Docs-/Introducao\\_ao\\_PVM.pdf](http://www-asc.di.fct.unl.pt/~jml/LEI/SCPD/Docs-/Introducao_ao_PVM.pdf)>.
- MACORATTI, José Carlos. **UML - Unified Modeling Language e Visual Modeler**. 2006. Paginação Irregular. Acessado em 28 de abril de 2007. Disponível em: <[http://www.macoratti-net/uml\\_vb.htm](http://www.macoratti-net/uml_vb.htm)>.
- MALISKA, Clóvis R. **Transferência de Calor e Mecânica dos Fluidos Computacional**. Florianópolis: Editorial LTC, 2004. 472 p. ISBN 85-216-1396-2.
- MEIER, Henry (Cood). **Computação de alto desempenho para estudos de validação e verificação em fluidodinâmica computacional (CAD – CFD)**. Blumenau: Laboratório de Fluidodinâmica computacional, Universidade Regional de Blumenau, fev 2004. 28 p.
- MERKLEY, Phil. **Beowulf history**. 2004. Paginação Irregular. Acessado em 10 de abril de 2006. Disponível em: <<http://www.beowulf.org/overview/history.html>>.
- MEURER, H. **Top500 SuperComputers**. 2004. Paginação Irregular. Acessado em 15 de abril de 2006. Disponível em: <<http://www.top500.org>>.
- MINETTO, Elton Luís. **Portais de Grids Computacionais**. Florianópolis, Santa Catarina, Brasil, Fev 2005. 55 p.
- MORETTI, Célio Oda; BITTENCOURT, Tulio Nogueira. **Aspectos Gerais da Computação Paralela e do Sistema PVM**,. 2004. Paginação Irregular. Acessado em 15 de abril de 2006. Disponível em: <[http://www.lmc.ep.usp.br/people/tbitten/gmec/Boletins\\_Tecnicos/](http://www.lmc.ep.usp.br/people/tbitten/gmec/Boletins_Tecnicos/)>.
- MORRISON, Richard S. **Cluster computing, architectures, operating systems, parallel processing & programming languages**. Boston: GNU Press, 2003. 292 p. Versão 2.4.
- ORSZAG, Steven A.; STAROSELSKY, I. CFD: Progress and problems. **Computer Physics Communications**, v. 127, p. 165–171, maio 2000.
- OTSUKA, Joice Lee. **Tutorial - Como Criar Documentos Interativos no WWW**. 2007. Paginação Irregular. Acessado em 23 de abril de 2007. Disponível em: <<http://penta.ufrgs.br/edu/forms/cgi.html>>.
- PENGUINCOMPUTING. **Scyld Beowulf Platform**. 2007. Paginação Irregular. Acessado em 12 de abril de 2007. Disponível em: <[http://www.penguincomputing.com/index.php?option=com\\_content&task=view&id=91&Itemid=496](http://www.penguincomputing.com/index.php?option=com_content&task=view&id=91&Itemid=496)>.
- PITANGA, Marcos. **Construindo Supercomputadores com Linux**. Rio de Janeiro: Brasport, 2004. 292 p. ISBN 85-7452-163-9.
- POP-PR. **Medidas de Tráfego RRD Tool e CACTI**. 2004. Paginação Irregular. Acessado em 23 de abril de 2007. Disponível em: <[http://www.pop-pr.rnp.br/tiki-index.php?page=Medidas de trafego RRDTOOL e CACTI](http://www.pop-pr.rnp.br/tiki-index.php?page=Medidas_de_trafego_RRDTOOL_e_CACTI)>.

SANTOS, Carlos H. **Computação paralela aplicada a problemas eletromagnéticos utilizando o método FDTD**. 2005. 95 p. Tese (Dissertação Mestrado) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas.

SCHNORR, Lucas M. **Visualização simultânea e multi-nível de informações de monitoramento de Cluster**. 2003. 113 p. Tese (Dissertação Mestrado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul, Porto Alegre.

SILVA, Cláudio Gomes; AMARAL, Henrique Mariano C. **Simulação de Fluxos de Fluidos Governados pelas Equações de Euler**. 2006. Paginação Irregular. Acessado em 19 de março de 2007. Disponível em: <[www.cct.uema.br/Relatorios\\_Tecnicos/RTindex.asp](http://www.cct.uema.br/Relatorios_Tecnicos/RTindex.asp)>.

STERLING, Thomas. **Beowulf cluster computing with Linux**. Cambridge, MA, USA: MIT Press, 2002. (Scientific and engineering computation). ISBN 0-262-69274-0.

TANENBAUM, Andrew S. **Sistemas operacionais modernos**. Rio de Janeiro: Editora Prentice-Hall do Brasil Ltda, 1995. ISBN 85-879-18575-0.

TOOLS, Fabulous Force Database. **Online Dokumentation**. 2007. Paginação Irregular. Acessado em 19 de abril de 2007. Disponível em: <<http://www.fabforce.net>>.

WIKIPEDIA. **Perl**. 2007a. Paginação Irregular. Acessado em 19 de abril de 2007. Disponível em: <<http://pt.wikipedia.org/wiki/Perl>>.

WIKIPEDIA. **BASH**. 2007b. Paginação Irregular. Acessado em 21 de abril de 2007. Disponível em: <<http://pt.wikipedia.org/wiki/Bash>>.

WIKIPEDIA. **CGI**. 2007c. Paginação Irregular. Acessado em 24 de abril de 2007. Disponível em: <<http://pt.wikipedia.org/wiki/CGI>>.

## APÊNDICE

### APÊNDICE A - SCRIPT PARA CRIAÇÃO DAS TABELAS NO MYSQL

```
CREATE DATABASE IF NOT EXISTS 'cluster CFD';
USE 'cluster CFD';
CREATE TABLE 'tabela_usuario' (
  'uid' smallint(6) UNSIGNED NOT NULL auto_increment,
  'login' varchar(20) NOT NULL,
  'nome' varchar(45) NOT NULL,
  'senha' varchar(70) NOT NULL,
  'email' varchar(70) NOT NULL,
  'sid_sessao' char(32) default NULL,
  'ult_login' datetime NOT NULL,
  PRIMARY KEY ('uid'),
  KEY 'login' ('login')
) ENGINE=MyISAM AUTO_INCREMENT=1 ;
CREATE TABLE jobs (
  'id' INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  'tabela_usuario_uid' SMALLINT(6) UNSIGNED NOT NULL,
  'nome_arquivo' VARCHAR(128) NULL,
  'arquivo' VARCHAR(256) NULL,
  'estado' VARCHAR(64) NOT NULL,
  'comentario' TEXT NULL,
  PRIMARY KEY(id),
  INDEX jobs_FKIndex1(tabela_usuario_uid),
  FOREIGN KEY(tabela_usuario_uid)
  REFERENCES tabela_usuario(uid)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
) ENGINE=MyISAM AUTO_INCREMENT=1 ;
```

## APÊNDICE B - SCRIPT PARA CRIAÇÃO DO USUÁRIO ADMIN NO BANCO MYSQL

```
INSERT INTO 'tabela_usuario' (  
    'uid', 'login', 'nome', 'senha', 'email', 'sid_sessao', 'ult_login'  
) VALUES (  
1, 'admin', 'Administrador', '$1$579794.1$U9kgyF3fseLXpbaX1QrD1',  
'admin@masterlfc.furb.br', '', '0000-00-00 00:00:00');
```

## ANEXO

### ANEXO A - LICENÇAS DE DISTRIBUIÇÃO DE SOFTWARE COM CÓDIGO FONTE ABERTO

#### GNU GENERAL PUBLIC LICENSE

Versão 2, junho de 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

This is an unofficial translation of the GNU General Public License into Brazilian Portuguese.

It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL – only the original English text of the GNU GPL does that. However, we hope that this translation will help Brazilian Portuguese speakers understand the GNU GPL better.

Esta é uma tradução não-oficial da Licença Pública Geral GNU (“GPL GNU”) para o português do Brasil. Ela não foi publicada pela Free Software Foundation, e legalmente não afirma os termos de distribuição de software que utiliza a GPL GNU – apenas o texto original da GPL GNU, em inglês, faz isso. Contudo, esperamos que esta tradução ajude aos que utilizam o português do Brasil a entender melhor a GPL GNU.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

A qualquer pessoa é permitido copiar e distribuir cópias desse documento de licença, desde que sem qualquer alteração.



## LICENÇA PÚBLICA GERAL GNU

### Introdução

As licenças de muitos software são desenvolvidas para restringir sua liberdade de compar-tilhá-lo e mudá-lo. Contrária a isso, a Licença Pública Geral GNU pretende garantir sua liberdade de compartilhar e alterar software livres – garantindo que o software será livre e gratuito para os seus usuários. Esta Licença Pública Geral aplica-se à maioria dos software da Free Software Foundation e a qualquer outro programa cujo autor decida aplicá-la. (Alguns outros software da FSF são cobertos pela Licença Pública Geral de Bibliotecas, no entanto.) Você pode aplicá-la também aos seus programas.

Quando nos referimos a software livre, estamos nos referindo a liberdade e não a preço. Nossa Licença Pública Geral foi desenvolvida para garantir que você tenha a liberdade de distribuir cópias de software livre (e cobrar por isso, se quiser); que você receba o código-fonte ou tenha acesso a ele, se quiser; que você possa mudar o software ou utilizar partes dele em novos programas livres e gratuitos; e que você saiba que pode fazer tudo isso.

Para proteger seus direitos, precisamos fazer restrições que impeçam a qualquer um negar estes direitos ou solicitar que você deles abdique. Estas restrições traduzem-se em certas responsabilidades para você, se você for distribuir cópias do software ou modificá-lo.

Por exemplo, se você distribuir cópias de um programa, gratuitamente ou por alguma quantia, você tem que fornecer aos recebedores todos os direitos que você possui. Você tem que garantir que eles também recebam ou possam obter o código-fonte. E você tem que mostrar-lhes estes termos para que eles possam conhecer seus direitos.

Nós protegemos seus direitos em dois passos: (1) com copyright do software e (2) com a oferta desta licença, que lhe dá permissão legal para copiar, distribuir e/ou modificar o software.

Além disso, tanto para a proteção do autor quanto a nossa, gostaríamos de certificar-nos que todos entendam que não há qualquer garantia nestes software livres. Se o software é modificado por alguém mais e passado adiante, queremos que seus recebedores saibam que o que eles obtiveram não é original, de forma que qualquer problema introduzido por terceiros não interfira na reputação do autor original.

Finalmente, qualquer programa é ameaçado constantemente por patentes de software. Queremos evitar o perigo de que distribuidores de software livre obtenham patentes individuais, o que tem o efeito de tornar o programa proprietário. Para prevenir isso, deixamos claro

que qualquer patente tem que ser licenciada para uso livre e gratuito por qualquer pessoa, ou então que nem necessite ser licenciada.

Os termos e condições precisas para cópia, distribuição e modificação se encontram abaixo:

## LICENÇA PÚBLICA GERAL GNU

### TERMOS E CONDIÇÕES PARA CÓPIA, DISTRIBUIÇÃO E MODIFICAÇÃO

0. Esta licença se aplica a qualquer programa ou outro trabalho que contenha um aviso colocado pelo detentor dos direitos autorais informando que aquele pode ser distribuído sob as condições desta Licença Pública Geral. O "Programa" abaixo refere-se a qualquer programa ou trabalho, e "trabalho baseado no Programa" significa tanto o Programa em si como quaisquer trabalhos derivados, de acordo com a lei de direitos autorais: isto quer dizer um trabalho que contenha o Programa ou parte dele, tanto originalmente ou com modificações, e/ou tradução para outros idiomas. (Doravante o processo de tradução está incluído sem limites no termo "modificação".) Cada licenciado é mencionado como "você".

Atividades outras que a cópia, a distribuição e modificação não estão cobertas por esta Licença; elas estão fora de seu escopo. O ato de executar o Programa não é restringido e o resultado do Programa é coberto apenas se seu conteúdo contenha trabalhos baseados no Programa (independentemente de terem sido gerados pela execução do Programa). Se isso é verdadeiro depende do que o programa faz.

1. Você pode copiar e distribuir cópias fiéis do código-fonte do Programa da mesma forma que você o recebeu, usando qualquer meio, deste que você conspícua e apropriadamente publique em cada cópia um aviso de direitos autorais e uma declaração de inexistência de garantias; mantenha intactas todos os avisos que se referem a esta Licença e à ausência total de garantias; e forneça a outros recebedores do Programa uma cópia desta Licença, junto com o Programa.

Você pode cobrar pelo ato físico de transferir uma cópia e pode, opcionalmente, oferecer garantia em troca de pagamento.

2. Você pode modificar sua cópia ou cópias do Programa, ou qualquer parte dele, assim gerando um trabalho baseado no Programa, e copiar e distribuir essas modificações ou trabalhos sob os termos da seção 1 acima, desde que você também se enquadre em todas estas condições:

- (a) Você tem que fazer com que os arquivos modificados levem avisos proeminentes afirmando que você alterou os arquivos, incluindo a data de qualquer alteração.
- (b) Você tem que fazer com que quaisquer trabalhos que você distribua ou publique, e que integralmente ou em partes contenham ou sejam derivados do Programa ou de suas partes, sejam licenciados, integralmente e sem custo algum para quaisquer terceiros, sob os termos desta Licença.
- (c) Se qualquer programa modificado normalmente lê comandos interativamente quando executados, você tem que fazer com que, quando iniciado tal uso interativo da forma mais simples, seja impresso ou mostrado um anúncio de que não há qualquer garantia (ou então que você fornece a garantia) e que os usuários podem redistribuir o programa sob estas condições, ainda informando os usuários como consultar uma cópia desta Licença. (Exceção: se o Programa em si é interativo mas normalmente não imprime estes tipos de anúncios, seu trabalho baseado no Programa não precisa imprimir um anúncio.)

Estas exigências aplicam-se ao trabalho modificado como um todo. Se seções identificáveis de tal trabalho não são derivadas do Programa, e podem ser razoavelmente consideradas trabalhos independentes e separados por si só, então esta Licença, e seus termos, não se aplicam a estas seções quando você distribui-las como trabalhos em separado. Mas quando você distribuir as mesmas seções como parte de um todo que é trabalho baseado no Programa, a distribuição como um todo tem que se enquadrar nos termos desta Licença, cujas permissões para outros licenciados se estendem ao todo, portanto também para cada e toda parte independente de quem a escreveu.

Desta forma, esta seção não tem a intenção de reclamar direitos ou contestar seus direitos sobre o trabalho escrito completamente por você; ao invés disso, a intenção é a de exercer o direito de controlar a distribuição de trabalhos, derivados ou coletivos, baseados no Programa.

Adicionalmente, a mera adição ao Programa de outro trabalho não baseado no Programa (ou de trabalho baseado no Programa) em um volume de armazenamento ou meio de distribuição não faz o outro trabalho parte do escopo desta Licença.

3. Você pode copiar e distribuir o Programa (ou trabalho baseado nele, conforme descrito na Seção 2) em código-objeto ou em forma executável sob os termos das Seções 1 e 2 acima, desde que você faça um dos seguintes:

- (a)O acompanhe com o código-fonte completo e em forma acessível por máquinas, que tem que ser distribuído sob os termos das Seções 1 e 2 acima e em meio normalmente utilizado para o intercâmbio de software; ou,
- (b)O acompanhe com uma oferta escrita, válida por pelo menos três anos, de fornecer a qualquer um, com um custo não superior ao custo de distribuição física do material, uma cópia do código-fonte completo e em forma acessível por máquinas, que tem que ser distribuído sob os termos das Seções 1 e 2 acima e em meio normalmente utilizado para o intercâmbio de software; ou,
- (c)O acompanhe com a informação que você recebeu em relação à oferta de distribuição do código-fonte correspondente. (Esta alternativa é permitida somente em distribuição não comerciais, e apenas se você recebeu o programa em forma de código-objeto ou executável, com oferta de acordo com a Subseção b acima.)

O código-fonte de um trabalho corresponde à forma de trabalho preferida para se fazer modificações. Para um trabalho em forma executável, o código-fonte completo significa todo o código-fonte de todos os módulos que ele contém, mais quaisquer arquivos de definição de "interface", mais os "scripts" utilizados para se controlar a compilação e a instalação do executável. Contudo, como exceção especial, o código-fonte distribuído não precisa incluir qualquer componente normalmente distribuído (tanto em forma original quanto binária) com os maiores componentes (o compilador, o "kernel" etc.) do sistema operacional sob o qual o executável funciona, a menos que o componente em si acompanhe o executável.

Se a distribuição do executável ou código-objeto é feita através da oferta de acesso a cópias de algum lugar, então ofertar o acesso equivalente a cópia, do mesmo lugar, do código-fonte equivale à distribuição do código-fonte, mesmo que terceiros não sejam compelidos a copiar o código-fonte com o código-objeto.

4. Você não pode copiar, modificar, sub-licenciar ou distribuir o Programa, exceto de acordo com as condições expressas nesta Licença. Qualquer outra tentativa de cópia, modificação, sub-licenciamento ou distribuição do Programa não é válida, e cancelará automaticamente os direitos que lhe foram fornecidos por esta Licença. No entanto, terceiros que de você receberam cópias ou direitos, fornecidos sob os termos desta Licença, não terão suas licenças terminadas, desde que permaneçam em total concordância com ela.
5. Você não é obrigado a aceitar esta Licença já que não a assinou. No entanto, nada mais o dará permissão para modificar ou distribuir o Programa ou trabalhos derivados deste.

Estas ações são proibidas por lei, caso você não aceite esta Licença. Desta forma, ao modificar ou distribuir o Programa (ou qualquer trabalho derivado do Programa), você estará indicando sua total aceitação desta Licença para fazê-los, e todos os seus termos e condições para copiar, distribuir ou modificar o Programa, ou trabalhos baseados nele.

6. Cada vez que você redistribuir o Programa (ou qualquer trabalho baseado nele), os recebedores adquirirão automaticamente do licenciador original uma licença para copiar, distribuir ou modificar o Programa, sujeitos a estes termos e condições. Você não poderá impor aos recebedores qualquer outra restrição ao exercício dos direitos então adquiridos. Você não é responsável em garantir a concordância de terceiros a esta Licença.

7. Se, em consequência de decisões judiciais ou alegações de infringimento de patentes ou quaisquer outras razões (não limitadas a assuntos relacionados a patentes), condições forem impostas a você (por ordem judicial, acordos ou outras formas) e que contradigam as condições desta Licença, elas não o livram das condições desta Licença. Se você não puder distribuir de forma a satisfazer simultaneamente suas obrigações para com esta Licença e para com as outras obrigações pertinentes, então como consequência você não poderá distribuir o Programa. Por exemplo, se uma licença de patente não permitirá a redistribuição, livre de "royalties", do Programa, por todos aqueles que receberem cópias direta ou indiretamente de você, então a única forma de você satisfazer a ela e a esta Licença seria a de desistir completamente de distribuir o Programa.

Se qualquer parte desta seção for considerada inválida ou não aplicável em qualquer circunstância particular, o restante da seção se aplica, e a seção como um todo se aplica em outras circunstâncias.

O propósito desta seção não é o de induzi-lo a infringir quaisquer patentes ou reivindicação de direitos de propriedade outros, ou a contestar a validade de quaisquer dessas reivindicações; esta seção tem como único propósito proteger a integridade dos sistemas de distribuição de software livres, o que é implementado pela prática de licenças públicas. Várias pessoas têm contribuído generosamente e em grande escala para os software distribuídos usando este sistema, na certeza de que sua aplicação é feita de forma consistente; fica a critério do autor/doador decidir se ele ou ela está disposto a distribuir software utilizando outro sistema, e um licenciado não pode impor qualquer escolha.

Esta seção destina-se a tornar bastante claro o que se acredita ser consequência do restante desta Licença.

8. Se a distribuição e/ou uso do Programa são restringidos em certos países por patentes ou direitos autorais, o detentor dos direitos autorais original, e que colocou o Programa sob

esta Licença, pode incluir uma limitação geográfica de distribuição, excluindo aqueles países de forma a tornar a distribuição permitida apenas naqueles ou entre aqueles países então não excluídos. Nestes casos, esta Licença incorpora a limitação como se a mesma constasse escrita nesta Licença.

- 9.A Free Software Foundation pode publicar versões revisadas e/ou novas da Licença Pública Geral de tempos em tempos. Estas novas versões serão similares em espírito à versão atual, mas podem diferir em detalhes que resolvem novos problemas ou situações. A cada versão é dada um número distinto. Se o Programa especifica um número de versão específico desta Licença que se aplica a ele e a "qualquer nova versão", você tem a opção de aceitar os termos e condições daquela versão ou de qualquer outra versão publicada pela Free Software Foundation. Se o programa não especifica um número de versão desta Licença, você pode escolher qualquer versão já publicada pela Free Software Foundation.
- 10.Se você pretende incorporar partes do Programa em outros programas livres cujas condições de distribuição são diferentes, escreva ao autor e solicite permissão. Para o software que a Free Software Foundation detém direitos autorais, escreva à Free Software Foundation; às vezes nós permitimos exceções a este caso. Nossa decisão será guiada pelos dois objetivos de preservar a condição de liberdade de todas as derivações do nosso software livre, e de promover o compartilhamento e reutilização de software em aspectos gerais.

#### AUSÊNCIA DE GARANTIAS

- 11.UMA VEZ QUE O PROGRAMA É LICENCIADO SEM ÔNUS, NÃO HÁ QUALQUER GARANTIA PARA O PROGRAMA, NA EXTENSÃO PERMITIDA PELAS LEIS APLICÁVEIS. EXCETO QUANDO EXPRESSADO DE FORMA ESCRITA, OS DETENTORES DOS DIREITOS AUTORAIS E/OU TERCEIROS DISPONIBILIZAM O PROGRAMA "NO ESTADO", SEM QUALQUER TIPO DE GARANTIAS, EXPRESSAS OU IMPLÍCITAS, INCLUINDO, MAS NÃO LIMITADO A, AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E AS DE ADEQUAÇÃO A QUALQUER PROPÓSITO. O RISCO TOTAL COM A QUALIDADE E DESEMPENHO DO PROGRAMA É SEU. SE O PROGRAMA SE MOSTRAR DEFEITUOSO, VOCÊ ASSUME OS CUSTOS DE TODAS AS MANUTENÇÕES, REPAROS E CORREÇÕES.
- 12.EM NENHUMA OCASIÃO, A MENOS QUE EXIGIDO PELAS LEIS APLICÁVEIS OU ACORDO ESCRITO, OS DETENTORES DOS DIREITOS AUTORAIS, OU QUALQUER OUTRA PARTE QUE POSSA MODIFICAR E/OU REDISTRIBUIR O

PROGRAMA CONFORME PERMITIDO ACIMA, SERÃO RESPONSABILIZADOS POR VOCÊ POR DANOS, INCLUINDO QUALQUER DANO EM GERAL, ESPECIAL, ACIDENTAL OU CONSEQÜENTE, RESULTANTES DO USO OU INCAPACIDADE DE USO DO PROGRAMA (INCLUINDO, MAS NÃO LIMITADO A, A PERDA DE DADOS OU DADOS TORNADOS INCORRETOS, OU PERDAS SOFRIDAS POR VOCÊ OU POR OUTRAS PARTES, OU FALHAS DO PROGRAMA AO OPERAR COM QUALQUER OUTRO PROGRAMA), MESMO QUE TAL DETENTOR OU PARTE TENHAM SIDO AVISADOS DA POSSIBILIDADE DE TAIS DANOS.

### FIM DOS TERMOS E CONDIÇÕES

#### Como Aplicar Estes Termos aos Seus Novos Programas

Se você desenvolver um novo programa, e quer que ele seja utilizado amplamente pelo público, a melhor forma de alcançar este objetivo é torná-lo software livre que qualquer um pode redistribuir e alterar, sob estes termos.

Para isso, anexe os seguintes avisos ao programa. É mais seguro anexá-los logo no início de cada arquivo-fonte para reforçarem mais efetivamente a inexistência de garantias; e cada arquivo deve possuir pelo menos a linha de “copyright” e uma indicação de onde o texto completo se encontra.

<uma linha que forneça o nome do programa e uma idéia do que ele faz.>

Copyright (C) <ano> <nome do autor>

Este programa é software livre; você pode redistribuí-lo e/ou modificá-lo sob os termos da Licença Pública Geral GNU, conforme publicada pela Free Software Foundation; tanto a versão 2 da Licença como (a seu critério) qualquer versão mais nova.

Este programa é distribuído na expectativa de ser útil, mas SEM QUALQUER GARANTIA; sem mesmo a garantia implícita de COMERCIALIZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Consulte a Licença Pública Geral GNU para obter mais detalhes.

Você deve ter recebido uma cópia da Licença Pública Geral GNU junto com este programa; se não, escreva para a Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

Inclua também informações sobre como contactá-lo eletronicamente e por carta.

Se o programa é interativo, faça-o mostrar um aviso breve como este, ao iniciar um modo interativo:

Gnomovision versão 69, Copyright (C) ano nome do autor

O Gnomovision não possui QUALQUER GARANTIA; para obter mais detalhes digite 'show w'. Ele é software livre e você está convidado a redistribuí-lo sob certas condições; digite 'show c' para obter detalhes.

Gnomovision versão 69, Copyright (C) <ano> <nome do autor>

O Gnomovision não possui QUALQUER GARANTIA; para obter mais detalhes digite 'show w'. Ele é software livre e você está convidado a redistribuí-lo sob certas condições; digite 'show c' para obter detalhes.

Os comandos hipotéticos 'show w' e 'show c' devem mostrar as partes apropriadas da Licença Pública Geral. Claro, os comandos que você usar podem ser ativados de outra forma que 'show w' e 'show c'; eles podem até ser cliques do mouse ou itens de um menu – o que melhor se adequar ao programa.

Você também deve obter do seu empregador (se você trabalha como programador) ou escola, se houver, uma “declaração de ausência de direitos autorais” sobre o programa, se necessário. Aqui está um exemplo; altere os nomes:

Yoyodyne, Inc., aqui declara a ausência de quaisquer direitos autorais sobre o programa 'Gnomovision' (que executa interpretações em compiladores) escrito por James Hacker.

<assinatura de Ty Coon>, 1o. de abril de 1989

Ty Con, Vice-presidente

Esta Licença Pública Geral não permite incorporar seu programa em programas proprietários. Se seu programa é uma biblioteca de sub-rotinas, você deve considerar mais útil permitir ligar aplicações proprietárias com a biblioteca. Se isto é o que você deseja, use a Licença Pública Geral de Bibliotecas GNU, ao invés desta Licença.