UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANA
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ESPECIALIZAÇÃO EM TELEINFORMÁTICA E REDES DE COMPUTADORES

ERNANI MAIESKI KOPP

CONSTRUÇÃO DE UM CLUSTER HPC PARA SIMULAÇÕES DE CFD

MONOGRAFIA

CURITIBA
2012

ERNANI MAIESKI KOPP

CONSTRUÇÃO DE UM CLUSTER HPC PARA SIMULAÇÕES DE CFD

Monografia apresentada como requisito parcial para obtenção do grau de especialista em Teleinformática e Redes de Computadores, do Departamento Acadêmico de Eletrônica da Universidade Tecnológica Federal do Paraná.
Orientador: Prof. Dr. Kleber Kendy Nabas.

CURITIBA
2012

RESUMO

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

O propósito desta monografia é realizar um estudo sobre clusters de computadores e implementar um cluster de alto desempenho usando a ferramenta Windows® Compute Cluster Server. A pesquisa apresenta conceitos teóricos de sistemas de computação paralela, fundamentos sobre clusters e ênfase nas principais características da ferramenta Windows® Compute Cluster Server® para testar a aplicabilidade dos conceitos teóricos na construção de um simples cluster HPC de alto desempenho e sua viabilidade para resolução de simulações numéricas específicas de programas de CFD.

Palavras-chave: cluster, computação paralela, HPC, Compute Cluster Server.

ABSTRACT

KOPP, M.Ernani. **Construction of an HPC cluster for CFD simulations**. 2012. 60 f. Monograph (Specialization in Teleinformática and computer networks) - Graduate Program in Technology, Federal Technological University of Paraná. Curitiba, 2012.

The purpose of this monograph is a study on clusters of computers and implements a high-performance cluster using the Windows ® Compute Cluster Server. The research presents theoretical concepts of parallel computing systems, fundamentals of clusters and focusing on core tool features Windows ® Compute Cluster Server ® to test the applicability of theoretical concepts in the construction of a simple high-performance cluster HPC and its viability for solving specific numerical simulations of CFD programs.

Keywords: cluster, parallel computer, HPC, Compute Cluster Server

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria de Pesquisa e Pós-Graduação do *Campus* Curitiba
Departamento Acadêmico de Eletrônica
Especialização em Teleinformática e Redes de Computadores

TERMO DE APROVAÇÃO

Título da Monografia (Construção de um cluster HPC para simulações de CFD)

por

Ernani Maieski Kopp

Esta monografia foi apresentada às 14:30 do dia 31 de Maio de 2012 como requisito parcial para a obtenção do título de ESPECIALISTA EM TELEINFORMÁTICA E REDES DE COMPUTADORES, Universidade Tecnológica Federal do Paraná. O candidato foi argüido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado com a nota 8,5 (OITO INTEIROS E CINCO DÉCIMOS)

Prof. Dr. Kleber Hendy Nabas
(UTFPR)

Prof. Dr. Walter Godoy Júnior
(UTFPR)

Visto da Coordenação

Prof. Dr. Walter Godoy Júnior
Coordenador do Curso

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

HPC	*High Performance Computing*
CFD	*Computational Fluid Dynamics*
MPI	*Message Passing Interface*
FLOPS	*Floating-point Operations Per Second*
UMA	*Uniform Memory Access*
NUMA	*Non-Uniform Memory Access*
FCFS	*First-Come First-Served*

SUMÁRIO

1 INTRODUÇÃO

1.1 TEMA

Na última década, as redes de computadores se tornaram mais rápidas e mais confiáveis, o que possibilitou a interligação dos computadores pessoais de maneira eficiente formando sistemas distribuídos, tais como o cluster. Os clusters são construídos com o intuito de oferecer uma imagem de um sistema único (apesar da distribuição de seus componentes), de forma que o usuário, em determinados casos, não perceba que está trabalhando com vários computadores ao mesmo tempo.

Clusters de alto desempenho também conhecidos como *clusters HPC* (do inglês *High Performance Computing*), estão se tornando cada vez mais populares em centros de pesquisa e indústrias de pequeno a médio porte que necessitem de um poder computacional mais poderoso que simples computadores pessoais possam oferecer; um dos fatores são preços relativamente mais acessíveis e uma gama maior de opções disponíveis no mercado. Existem soluções de baixo custo que usam software livre e configurações de hardwares mais simples e outras de custo mais elevado que utilizam softwares proprietários e hardwares sofisticados. A escolha da solução ideal dependerá de uma análise minuciosa das exigências necessárias, tanto de hardware e software, para assim conseguir um custo/beneficio satisfatório.

Reconhecendo então a amplitude e importância do nicho de aplicações abordadas pela Computação de alto desempenho, e aproveitando-se da necessidade iminente da evolução das tecnologias de software voltadas a estas arquiteturas, a Microsoft tem lançado nos últimos anos um programa específico para esta área, a qual culminou no lançamento do Windows® Compute Cluster Server (WCCS), sistema operacional voltado a simples implantação, configuração e gerenciamento de clusters. O intuito do presente trabalho é usar as ferramentas Windows Compute Cluster Server para montar um cluster HPC de baixo custo que possibilite um desempenho superior a qualquer estação de trabalho disponível.

1.2 OBJETIVOS

A seguir, será apresentado o objetivo geral e específico, que se pretende atingir com este projeto de pesquisa.

1.2.1 Objetivos Gerais

Realizar um estudo sobre computação paralela e clusters para adquiri conceitos básicos sobre o tema e depois implantar uma solução de cluster HPC de usando o Windows Compute Cluster Server.

1.2.2 Objetivos específicos

- Aprender conceitos de programação paralela e cluster;
- Descrever os recursos do Windows Compute Cluster Server;
- Montar e configurar HPC usando o Windows Compute Cluster Server;
- Realizar um teste básico do cluster HPC com o CFX.

1.3 PROBLEMA

Simulações numéricas exigem alto desempenho computacional, pode-se citar, por exemplo, simulações de fluidos em tubulações. Existem programas de CFD (do inglês: *Computational Fluid Dynamics*) que conseguem simular variados tipos de situação, porém a custo de muito esforço computacional. Empresas de pequeno e médio porte não possuem condições financeiras suficientes para comprar supercomputadores, mas necessitam de um sistema com poder computacional mais elevado que simples estações de trabalho nesse panorama os clusters que surgem cada vez mais como alternativas com abrangente custo/benefício.

1.4 JUSTIFICATIVA

Ao final deste projeto de pesquisa onde seus objetivos sejam alcançados, o mesmo poderá servir de orientação inicial para usuários que queiram conhecer os recursos e utilidades de um cluster HPC de relativo baixo custo, pois mesmo usando softwares proprietárias como o Windows é possível usufruir do período de teste, 6

meses, e estudar o programa com o objetivo de conseguir um conhecimento mais abrangente sobre clusters HPC.

1.5 PROCEDIMENTOS METODOLÓGICOS

Para o desenvolvimento deste projeto, utilizar-se-á referencias bibliográficas sobre o assunto proposto, equipamentos de redes e matérias didáticos de apoio. O estudo contará com uma abordagem básica sobre paralelismo estritamente ligado ao objetivo do trabalho para depois partir para uma abordagem mais prática com implantação de alguns dos conceitos teóricos pré-estudados, com o objetivo de montar um cluster HPC de baixo custo e possíveis considerações futuras de como melhora-lo.

1.6 ESTRUTURA DO TRABALHO

Os capítulos 2, 3 contêm conhecimentos teóricos necessários para um melhor desenvolvimento do tema. O capitulo 2 mostra uma prévia sobre paralelismo: são discutidos arquitetura de sistemas paralelos de alto desempenho e uma abordagem básica sobre MPI(do inglês *Message Passing Interface*); o capitulo 3 apresenta a uma explicação básica de clusters e seus tipos de topologias, o capitulo 4 é dedicado exclusivamente a explicar o funcionamento do Windows® Compute Cluster Server , o capitulo 5 mostra a aplicação dos conceitos apresentados nos capítulos anteriores com ênfase no capitulo 4, finalmente o capitulo 6 mostra a conclusão da monografia e considerações futuras sobre o cluster HPC construido.

2 COMPUTAÇÃO DE ALTO DESEMPENHO

2.1 INTRODUÇÃO

A aplicação que uma determinada tecnologia pode exercer pode revelar a seu poder e importância para aperfeiçoar as pesquisas. Tradicionalmente, estas aplicações emergem nas áreas de ciências computacionais e engenharia, de interesse tanto de instituições acadêmicas interessadas em pesquisa científica e tecnológica como de indústrias de grande porte (JUNIOR,2004). Podem-se enumerar algumas delas:

- Previsão climática,
- Previsão e simulação dos efeitos de catástrofes (erupções vulcânicas, terremotos, *tsunamis*,tornados, furacões, etc.),
- Fisiologia dos seres vivos,
- Modelagem de reservatórios de petróleo,
- Dinâmica dos fluidos,
- Descoberta de novos fármacos,
- Genômica e bioinformática,
- Engenharia financeira, econofísica e finanças quantitativas
- Mineração de dados,

A lista acima revela o escopo de utilidades que o sistema de alto desempenho pode exercer nas áreas cientificas. Basicamente os sistemas de alto desempenho devem realizar complexos cálculos matemáticos oriundos de simulações de fenômenos físicos. Esses cálculos requerem grande velocidade de processamento de variáveis de ponto flutuantes o que envolve também o suporte adequado no nível de linguagens de programação; esses fatores geram uma das principais características de sistemas de alto desempenho: o parâmetro FLOPS (do inglês, *Floating-point Operations Per Second*, ou operações de ponto flutuante por segundo), portanto um sistema é mais interessante quando consegue processar mais FLOPS por segundo. Atualmente os supercomputadores mais rápidos trabalham na casa do Petaflops (10^{15}) operações por segundo.

2.2 CONCEITOS SOBRE PROCESSAMENTO PARALELO

O processamento paralelo implica na divisão do problema a ser resolvido em tarefas, de forma que estas possam ser executadas por vários processadores simultamente (JUNIOR, 2004). Os processadores deverão cooperar entre si buscando maior eficiência. Dentre os vários fatores que explicam a necessidade de processamento paralelo, está a busca por melhor desempenho dos algoritmos utilizados, obtendo resultados satisfatórios em um tempo reduzido, quando comparado à execução seqüencial.

As diversas áreas nas quais a computação se aplica, seja na pesquisa básica ou na aplicação tecnológica, requerem cada vez mais recursos computacionais, em virtude dos algoritmos complexos que são utilizados e do tamanho do conjunto de dado a ser processado.

Concorrência/Paralelismo: A concorrência consiste em diversas tarefas ou processos sendo executado aparentemente de forma simultânea, o que não implica na utilização de mais de um processador (SNOW, 1992).

Afirmar que processos estão sendo executados em paralelo implica na existência de mais de um processador, ou seja, paralelismo ocorre quando há mais de um processo sendo executado no mesmo intervalo de tempo. Na Figura 1, é mostrado graficamente o conceito de paralelismo.

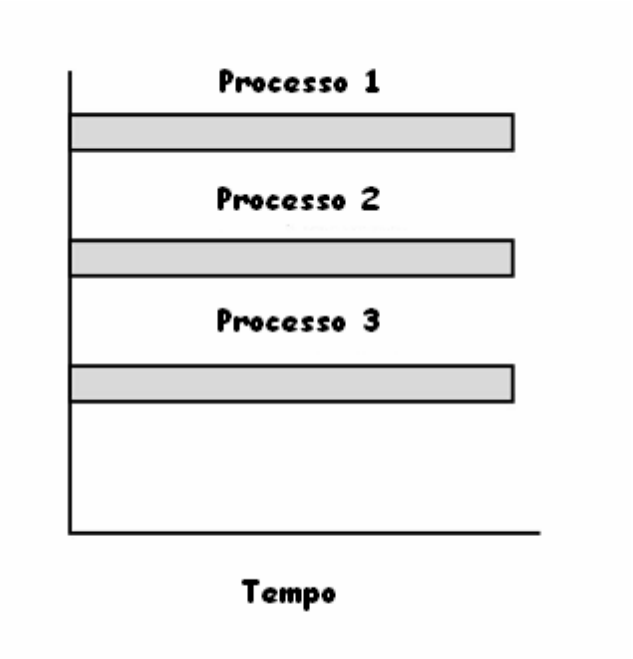

Fig. 1– Paralelismo Real (Adaptado de SNOW,1992)

Observa-se que num instante de tempo qualquer, tem-se três processos executados simultaneamente.

Quando vários processos são executados em um único processador, sendo que somente um deles é executado a cada vez, tem-se um pseudo-paralelismo. O usuário tem a falsa impressão de que suas tarefas são executadas em paralelo. Na realidade, o processador é compartilhado pelos processos. Na Figura 2, observa-se que, em um determinado instante, somente um processo é executado, enquanto que os outros foram iniciados, mas aguardam a liberação do processador para continuarem sua execução.

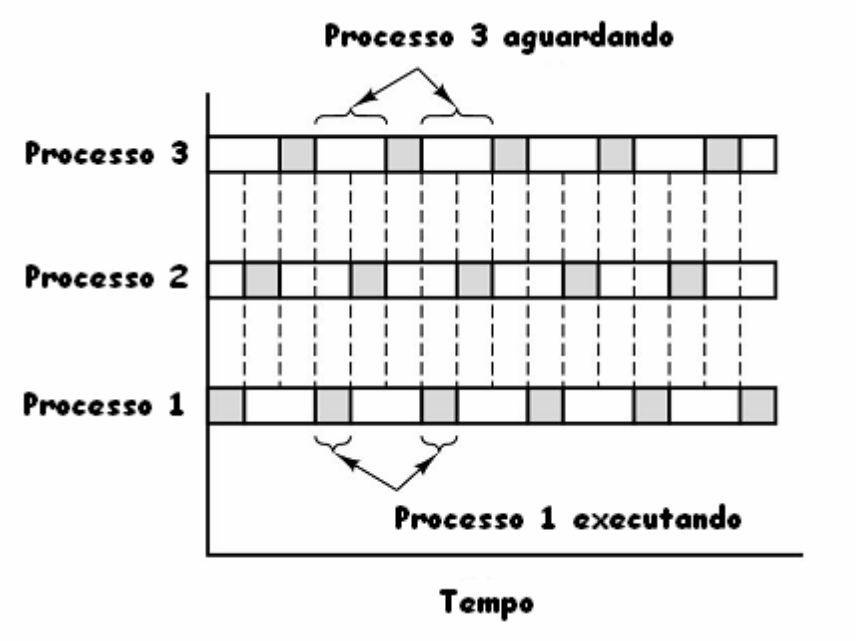

Fig. 2 – Pseudo Paralelismo (Adaptado de SNOW,1992)

Segundo as definições apresentadas acima, podem-se definir três tipos deformas de programação:

- Programação Seqüencial: quando várias tarefas são executadas uma após a outra;
- Programação Concorrente: quando várias tarefas competem pelo uso do processador;
- Programação Paralela: quando várias tarefas são executadas em paralelo

No desenvolvimento de uma arquitetura paralela, após a especificação do problema a ser resolvido, deve-se decidir em quantas tarefas o problema será dividido e como essas tarefas irão interagir. Essas decisões são diretamente afetadas pela natureza do problema e pelas características do hardware disponível.

Para a construção de uma arquitetura paralela eficiente devem ser levadas em contas três questões essenciais. A primeira é com relação à distribuição de dados pelos processadores. O programador deve estabelecer uma distribuição adequada dos volumes de rotinas de cálculo e de comunicação. A segunda questão a ser analisada é com relação à topologia de interconexão entre os processadores, para obter uma rede estável

para a troca de mensagens entre eles. A terceira questão a ser observada é com relação à distribuição de controle entre os processadores. As tarefas devem ser alocadas aos processadores respeitando o sincronismo entre as suas interações.

2.3 *MESSAGE PASSING INTERFACE* (MPI)

Para a comunicação entre os processadores é necessário um ambiente de troca de mensagens para gerenciá-los. Um ambiente de troca de mensagens consiste em uma camada de serviço que atende as solicitações de envio e recebimento de mensagens, segundo um protocolo bem definido. Define ainda bibliotecas para uso com linguagens de programação (como C e Fortran), permitindo o desenvolvimento de aplicações paralelas.

O MPI surgiu com a finalidade de criar um padrão de troca de mensagens melhorando a portabilidade das aplicações para diferentes máquinas. Sua primeira versão foi publicada em 1994 e atualizada em junho de 1995. Atualmente está sendo discutida uma nova extensão chamada MPI 2. Este é um produto resultante de um Fórum aberto constituído de pesquisadores, empresas e usuários que definiram a sintaxe, a semântica e o conjunto de rotinas padronizadas para a passagem de mensagens (IGNÁCIO, 2002).

Há duas características que podem ser apresentadas como limitações do MPI. A primeira delas é deixar o programador como responsável direto pela paralelização. A segunda característica é com relação ao custo. Em alguns ambientes específicos, o custo de comunicação pode tornar-se extremamente proibitivo pela quantidade de transmissão de mensagens necessárias à aplicação.

Como ponto forte do MPI pode-se citar as seguintes características (PACHECO,1995):

- Eficiência: o MPI foi cuidadosamente projetado para executar eficientemente em máquinas de diferentes configurações;
- Facilidade: é baseado em rotinas comuns de paralelismo e de outras bibliotecas de trocas de mensagens;
- Portabilidade: é compatível com sistemas de memória distribuída, memória compartilhada e outras arquiteturas especiais;

- Transparência: permite que um programa seja executado em sistemas heterogêneos sem mudanças significativas;
- Segurança: provê uma interface de comunicação confiável, eximindo o usuário de possíveis preocupações com falhas;
- Escalabilidade: suporta comunicação coletiva com a criação de subgrupos de processos, melhorando ainda mais a troca de mensagens.
- Na biblioteca MPI, assim como em qualquer biblioteca de troca de mensagens, existem componentes comuns. Esses componentes são as rotinas de gerência de processos, de comunicação de grupos e de comunicação ponto a ponto.
- As rotinas de gerência de processos são as funções de inicializar, finalizar, determinar o número e identificar os processos. As rotinas de comunicação de grupos são as funções de broadcast e sincronização de processos, dentre outras. As rotinas de comunicação ponto a ponto realizam a comunicação entre dois processadores.

2.4 ESTRUTURA FÍSICA DE SISTEMAS COMPUTACIONAIS DE ALTO DESEMPENHO

Em sistemas computacionais existem três condições para melhorar o desempenho: usar um algoritmo melhor, um computador mais rápido ou usar de paralelismo, ou seja, dividir os dados em múltiplos computadores.

Basicamente, a primeira condição é satisfeita se o algoritmo for amplamente testado e revisado para descobrir cálculos redundantes, loops desnecessários e erros de lógica; a segunda condição é apropriada às aplicações que não exijam computação de alto desempenho. Atualmente o uso de processadores *multi-core* permite o uso mais abrangente de aplicações que usam complexos cálculos matemáticos e a terceira condição é o paralelismo, que basicamente remete ao uso de redes de computadores para dividir os cálculos necessários entre diferentes computadores que trabalham ao mesmo tempo para resolvê-los.

2.4.1 Arquitetura com um processador

Este tipo de sistema utiliza-se da arquitetura de Newman que consiste em um processador conectado a memória por um barramento; instruções ou dados são “movidos” entre o processador e a memória pelo barramento. O desempenho desse tipo de arquitetura depende da velocidade de processamento do processador e da latência deste com o barramento e conseqüentemente da memória. O problema da arquitetura Newman é que o processador trabalha numa velocidade bem superior à memória, portanto o processador tem que “esperar” a memória processar seus dados para então iniciar uma nova tarefa computacional. Esse tipo de problema é conhecido como *Neumann bottleneck*; uma forma de amenizar esse problema é a inserção de memórias de alto desempenho conhecidas como memória cachê: estas possuem baixíssima latência e são encapsulados junto ao chip do processador onde fazem um intermédio entre as memórias de latência mais elevadas (RAM) possibilitando um melhor aproveitamento dos ciclos de *clock* do processador. Além da memória cachê existem outras tecnologias que ajudam a melhorar o desempenho desse tipo de arquitetura:

- *pipelines*: permitem a execução de mais de uma instrução simultaneamente (no mesmo ciclo de clock). Isto é obtido através da implementação de múltiplas unidades funcionais, que são unidades onde as instruções são executadas.
- Multi-núcleo (múltiplos núcleos, do inglês multi-core) consiste em colocar dois ou mais núcleos de processamento (cores) no interior de um único chip. Estes dois ou mais núcleos são responsáveis por dividir as tarefas entre si, ou seja, permitem trabalhar em um ambiente multitarefa.

2.4.2 Sistemas de múltiplos Processadores

Sistemas com múltiplos processadores são arquiteturas que possuem dois ou mais processadores interligados e que funcionam em conjunto na execução de tarefas independentes ou no processamento simultâneo de uma mesma tarefa. Inicialmente, os computadores eram vistos como máquinas seqüenciais, em que o processador executava as instruções de um programa uma de cada vez. (SLOAN, 2004) Com a implantação de sistemas com múltiplos processadores, o conceito de paralelismo pode ser expandido a um nível mais amplo.

2.4.2.1-UMA (Uniform Memory Access)

Neste tipo de máquina, o tempo para o acesso aos dados na memória é o mesmo para todos os processadores a para todas as posições da memória. Essas arquiteturas também são chamadas de SMP (*Symmetric MultiProcessor*). A forma de interconexão mais comum neste tipo de máquina é o barramento e a memória geralmente é desenvolvida com um único módulo. O principal problema com este tipo de arranjo é que o barramento e a memória tornam-se gargalos para o sistema, que fica limitado a uma única transferência por vez. A figura 3 apresenta uma arquitetura UMA.

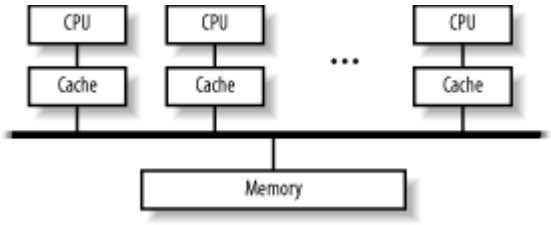

Fig. 3 – Arquitetura UMA (Fonte: SLOAN, 2004)

2.4.2.2 NUMA (Non-Uniform Memory Access)

Neste tipo de multiprocessadores, a memória geralmente é distribuída e, portanto desenvolvida com múltiplos módulos. Cada processador está associado a um módulo, mas o acesso aos módulos ligados a outro processador é possível. O espaço de endereçamento é comum a todos os processadores e a latência para ler ou escrever na memória pertencente a outro processador é maior que a latência para o acesso à memória local. A figura 4 apresenta a arquitetura NUMA

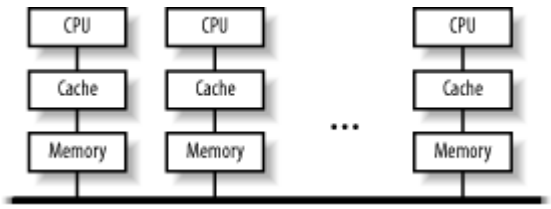

Fig. 4 – Arquitetura NUMA (Fonte: SLOAN, 2004)

2.5 AGLOMERADOS DE COMPUTADORES (CLUSTERS)

Um aglomerado de computadores, ou cluster, é um grupo de computadores que trabalham juntos para realizar determinada tarefa computacional. Um cluster possui três elementos básicos: uma coleção de computadores individuais, uma rede entre esses computadores e software que gerencie a comunicação entre os computadores.

O sistema operacional mais utilizado em clusters atualmente é o GNU/Linux (JUNIOR, 2004), representando mais de 75 % da lista do TOP500 (lista das 500 arquiteturas de super-computação com maior desempenho), Porém, há também soluções proprietárias como o Windows® Compute Cluster Server ®, que é a alternativa da Microsoft® para computação de alto desempenho. Ele utiliza a família Windows Server® e combina os serviços do *Active Directory* com a facilidade de uso do Windows® para obter uma plataforma eficiente de alto desempenho

O sistema de cluster GNU/Linux mais conhecido é o *Beowulf*, este faz uso de computadores pessoais, não especializados e, portanto mais baratos. O projeto foi criado por Donald Becker da NASA, e hoje são usados em todo mundo, principalmente para processamento de dados com finalidade científica, uma área em que são muito utilizados é na renderizacão de filmes. As principais vantagens do cluster *Beowulf*:

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

A figura 5 Apresenta um exemplo de cluster *Beowulf,* de estações de trabalho.

Fig. 5 – Cluster *Beowulf (*Fonte: http://www.dailyvillain.com/gamingr)

O Windows® Compute Cluster Server® oferece um ambiente mais amigável sendo baseado principalmente em interfaces gráficas, o que facilita a sua utilização. Porém, por ser um sistema fechado, não oferece tantas alternativas de ferramentas quanto o GNU/Linux. Mais detalhes sobre o Windows® Compute Cluster Server serão vistos no capitulo 4.

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

Outro recurso interessante é o uso de estações de trabalho como "auxiliares" de clusters, por exemplo, um *cluster* pode se conectar a uma rede LAN com várias estações de trabalho ociosas e usar os recursos computacionais disponíveis de cada estação para processar dados. Esse tipo de estrutura recebe o nome de *cluster* COW *do inglês, cluster of workstations*).

3 ESTRUTURA BÁSICA DE CLUSTERS

É necessário pensar um pouco sobre a estrutura interna do cluster e sua topologia: isso implicará decidir quais papéis as máquinas individuais irão exercer e qual o tipo de interconexão é a mais adequada. Um conceito básico são os elementos que compõem um cluster, que são seguintes:

- Nó principal (também conhecido por *head-node*): computador responsável pela administração dos nós computacionais exerce também a função de autenticação e segurança do *cluster*;
- Nó computacional: computador dedicado a processar os dados do cluster.

3.1 *CLUSTERS* SIMÉTRICOS

A abordagem mais simples é um conjunto simétrico; cada nó pode funcionar como um computador individual. Cria-se uma sub-rede com diversas máquinas e pode-se adicionar nós que são gerenciados por software de cluster específico. O problema desse tipo de configuração é a falta de segurança e gerenciamento, visto que não existe um controle centralizado, como mostra a Figura 6. Um exemplo de cluster simétrico são os clusters que usam de estações de trabalho

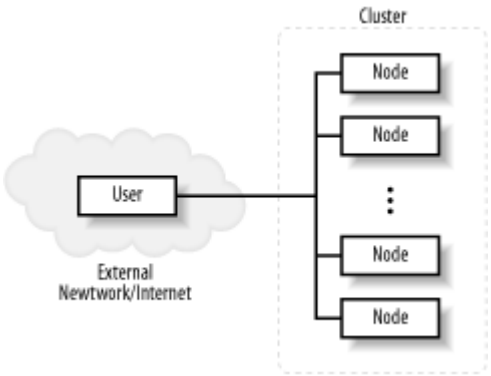

Fig. 6 – Cluster simétrico (Fonte: SLOAN, 2004)

3.2 *CLUSTERS* ASSIMÉTRICOS

Para o cluster desempenhar tarefas mais especifica e complexas, uma arquitetura assimétrica é mais adequada, a figura 7 mostra um esquema desse tipo de cluster. Com clusters assimétricos um computador é o nó principal ou *head-node*. Ele serve como um *gateway* entre os nós restantes e os usuários. Os nós computacionais geralmente utilizam configuração mínima de hardware. Uma vez que todo o tráfego deve passar através do *head node*, os clusters assimétricos tendem a fornecer um nível de segurança mais elevado.

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

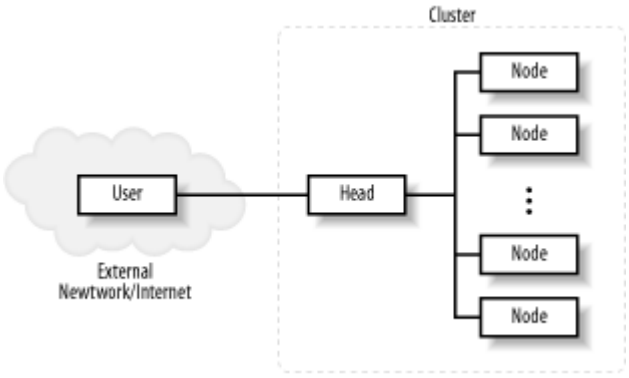

Fig. 7 – Cluster Assimétrico (Fonte: SLOAN, 2004)

Para um melhor intercâmbio entre o cluster e as fontes de dados é necessário "expandir" a topologia do cluster, detalhes na figura 8 a fim de melhorar o acesso deste pelos usuários. Para isso usa-se a redes heterogêneas: cenário ao qual *cluster* é dinamicamente ligado a outro tipo de rede, por exemplo, uma LAN, de um laboratório de pesquisa que pode continuamente estabelecer contato com o ele para processamento de dados.

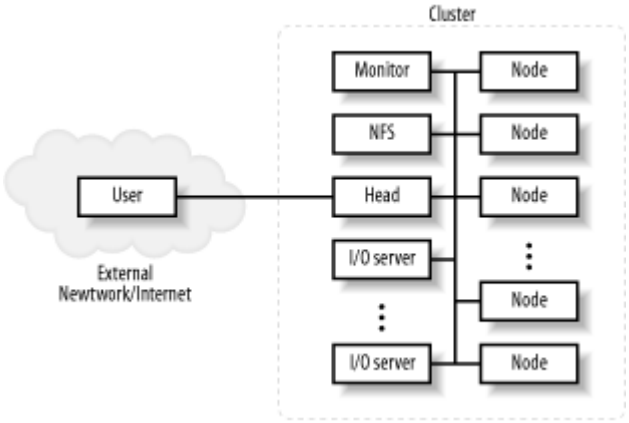

Fig. 8 – Cluster “Estendido” (Fonte: SLOAN, 2004)

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

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

3.3 MODELOS DE CLUSTERS

Um aglomerado de computadores, *cluster*, pode ser definido conforme a função especifica em um determinado cenário, portanto pode-se classificá-los como: de alta disponibilidade, de balanceamento de carga e de alto desempenho.

3.3.1 Alta disponibilidade:

Clusters de alta disponibilidade, também chamados de clusters de *failover*, são freqüentemente utilizados em aplicações de missão crítica onde determinado serviço não pode parar, por exemplo, um site de compras na Internet. A chave para a disponibilidade elevada é a redundância. Um cluster de alta disponibilidade é composto por várias máquinas, um subconjunto dos quais pode fornecer o serviço

apropriado. Apenas uma máquina ou servidor está diretamente disponível e todas as outras máquinas estarão em modo de “espera”. Elas irão controlar o servidor primário para assegurar que ele permaneça operacional. Se o servidor primário falhar, um servidor secundário assume o seu lugar no sistema, enquanto o servidor primário estiver indisponível.

3.3.2 Balanceamento de carga:

A idéia por trás de um cluster de balanceamento de carga é proporcionar melhor desempenho dividindo o trabalho entre vários computadores. Por exemplo, quando um servidor web é construído usando o cluster de balanceamento, as diferentes consultas para o servidor são distribuídas entre os computadores nos clusters. Isto pode ser conseguido utilizando um algoritmo de rodízio simples. Por exemplo, *Round-Robin* DNS pode ser usado para mapear as respostas para consultas DNS para os endereços IP diferentes. Ou seja, quando uma consulta DNS é feita, o servidor DNS local retorna os endereços da máquina próxima do cluster, visitando as máquinas em um *round-robin*. No entanto, essa abordagem pode levar a desequilíbrios de cargas. Algoritmos mais sofisticados usam o *feedback* das máquinas individuais para determinar qual máquina pode lidar melhor com a próxima tarefa de processamento.

3.3.3 Alto desempenho

Também conhecido como HPC (do inglês *High-performance computing*) Este tipo de cluster é destinado à grandes tarefas computacionais. Uma complexa tarefa computacional pode ser dividida em pequenas tarefas que são distribuídas ao redor dos nós computacionais, como se fosse um supercomputador massivamente paralelo. Estes *clusters* são usados para computação cientifica ou análises financeiras, tarefas típicas para exigência de alto poder de processamento.

4 WINDOWS® COMPUTE CLUSTER SERVER®

Devido à amplitude e importância de diversas aplicações abordadas pela computação de alto desempenho e à necessidade da evolução das tecnologias de programas voltadas a esta arquitetura, a Microsoft® lançou o Windows® Compute cluster Server®, que é um sistema operacional voltado à implantação, configuração e gerenciamento de clusters.

A Microsoft® tem compartilhado seu interesse com outras indústrias de software e comunidade acadêmica para construção de tecnologias de suporte a aplicações de alto desempenho sobre arquiteturas paralelas (RUSSEL, 2005). Em um mercado tradicionalmente dominado por várias versões do sistema operacional Linux, o Windows® Compute Cluster Server® tem sido proposto pela Microsoft® como uma alternativa baseada na versão servidor do sistema operacional Windows® para simplificar a implantação e gerenciamento de *cluster*.

4.1 ARQUITETURA

O Windows® Compute Cluster Server® gerencia um grupo de computadores que inclui um único “nó” principal (do inglês, “*head node*”) e um ou mais “nós” de computação (do inglês, “*compute nodes*”). A Figura 9 apresenta um diagrama de uma das possíveis arquiteturas para o cluster. Neste caso, pode-se observar o “nó” principal e os “nós” de computação. O “nó” principal controla e serve como mediador para todos os acessos aos recursos do grupo (*Active Directory*,Servidor de Arquivos, Servidor de Operações e o Servidor de E-mail) e é o único ponto de gerenciamento, implantação e agendamento de serviços para o cluster.

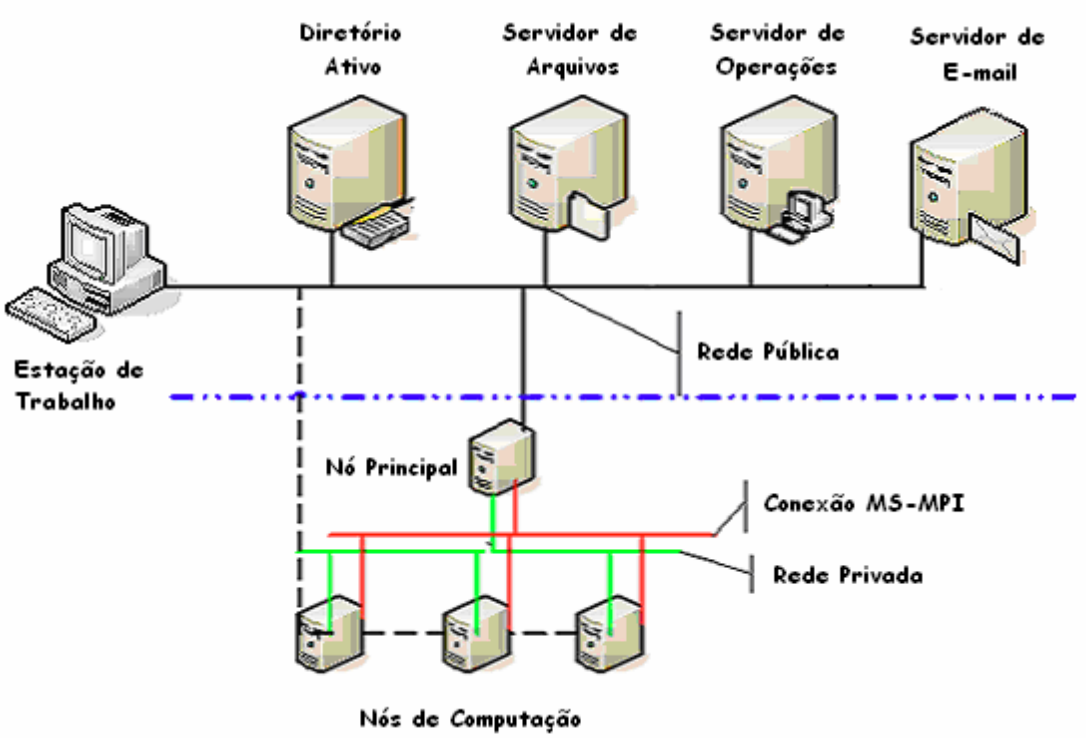

Fig. 9 – Estrutura típica de um Windows® Compute Cluster Server®(Adaptado de RUSSEL, 2005)

4.2 INTERFACE DE PASSAGEM DE MENSAGENS DA MICROSOFT® (MS-MPI)

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

4.3. IMPLANTAÇÃO DO CLUSTER

Para a implantação do Windows® Compute Cluster Server ® são necessários alguns requisitos de hardware, dentre eles:

- Processador: Intel ou AMD com suporte à arquitetura de 64 bits;
- Memória RAM: mínimo 512 MB;
- Máximo de 4 processadores por nó;
- Espaço mínimo de disco para instalação: 4GB.

Uma vez satisfeitos estes requisitos, é essencial o planejamento da implantação, inicialmente consistindo na escolha de uma topologia de rede.

O Windows® Compute Cluster Server® distingue cinco topologias de rede, classificadas em três tipos diferentes: rede pública, rede privada e rede MPI. A rede pública corresponde à rede corporativa, conectada ao “nó” principal e, opcionalmente aos “nós” de computação. É através da rede pública que os usuários se conectam ao cluster para submeter seus trabalhos, inclusive remotamente. Todo tráfego no cluster relacionado ao gerenciamento e implantação deve ser realizado sobre a rede pública também conhecida como *Enterprise*, caso não exista uma rede privada. A rede privada é dedicada ao cluster, sem acesso externo, encarregando-se do tráfego relacionado ao gerenciamento, implantação e tráfego MPI, caso não esteja disponível uma rede MPI. A rede MPI, quando disponível, responsabiliza-se por todo tráfego MPI, em geral sobre uma interconexão de alto desempenho. A existência da rede MPI é essencial em aplicações de computação intensiva, onde a cooperação entre as tarefas exige a troca de grandes massas de dados.

As cinco topologias de rede que podem ser utilizadas pelo Windows® Compute Cluster Server ® são:Cenário 1: O “nó” principal possui duas NICs (do inglês *Network Interface Card*). Na Figura 10, observa-se que uma NIC está conectada à rede pública ou corporativa existente e a outra está conectada ao cluster que interliga o “nó” principal aos “nós” de computação. Na rede privada trafegam todas as informações entre o “nó” principal e os outros “nós”, inclusive implantação, gerenciamento e tráfego MPI.

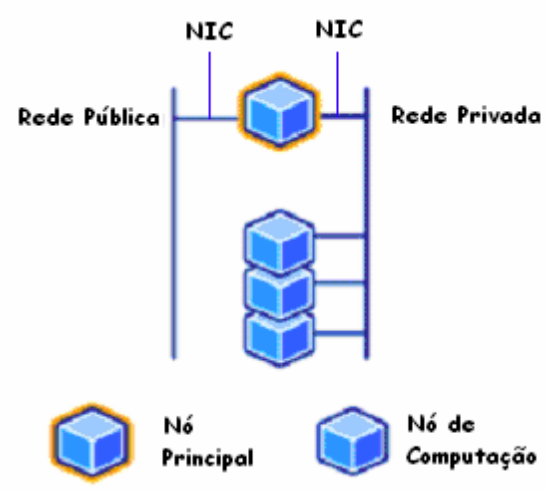

Fig.10 – Topologia 1 (Fonte Adaptado de RUSSEL, 2005)

- Cenário 2: Na Figura 11 é apresentado o Cenário 2. Pode-se observar que tanto o “nó” principal como os ”nós” de computação possuem duas NICs, uma

conectada à rede pública e outra à rede privada. Como no Cenário 1, as comunicações entre os “nós” – inclusive implantação, gerenciamento e tráfego MPI – são todas transmitidas na rede privada. Entretanto, neste caso, a rede pública está ligada a cada computador. Nesta topologia, toda a comunicação entre os “nós” de computação e a rede pública, pode ser feita diretamente, sem a necessidade de passar pelo “nó” principal como ocorre no Cenário 1.

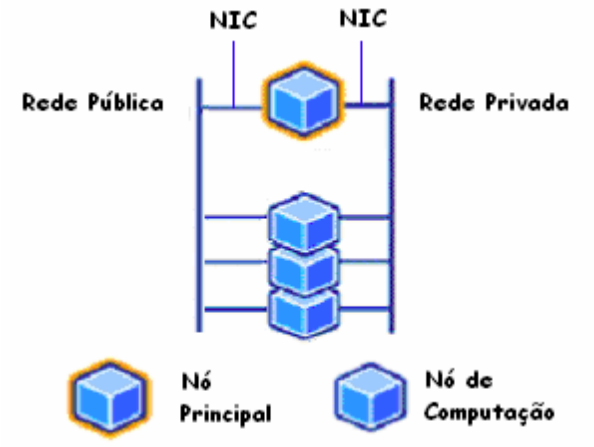

Fig. 11 – Topologia 2 (Fonte Adaptado de RUSSEL, 2005)

- Cenário 3: O terceiro cenário é similar ao primeiro no sentido de que a rede pública é ligada somente ao “nó” principal. A principal diferença entre os Cenários 1 e 3 é que no Cenário 3, o cluster é instalado com uma rede MPI que conecta todos os “nós”. Como mostrado na Figura 12, o “nó” principal agora possui uma NIC adicional, uma placa de alta velocidade conectada à rede MPI. Além disso, cada computador “nó” possui uma segunda NIC, uma para a rede privada e outra para a rede MPI. A rede MPI é usada para isolar o tráfego MPI, melhorando o desempenho da troca de informações entre os “nós”.

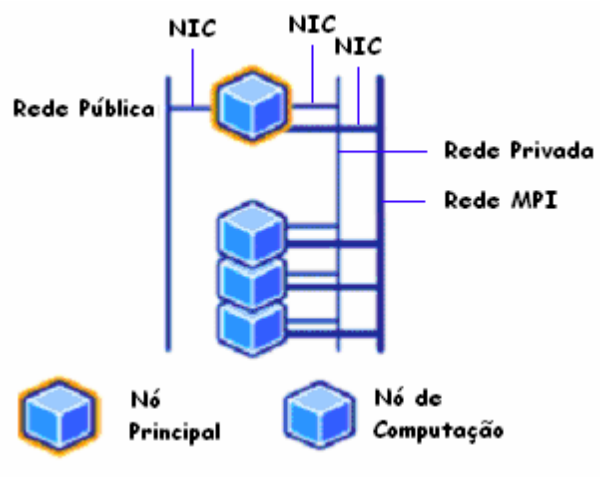

Fig. 12 – Topologia 3 (Fonte Adaptado de RUSSEL, 2005)

- Cenário 4: Na Figura 13 é apresentada a quarta topologia de rede. Esta é igual à segunda topologia, exceto porque inclui uma rede MPI de alta velocidade. A rede privada transmite apenas tráfego de implantação e gerenciamento. A rede MPI existe, como no Cenário 3, para isolar o tráfego MPI.

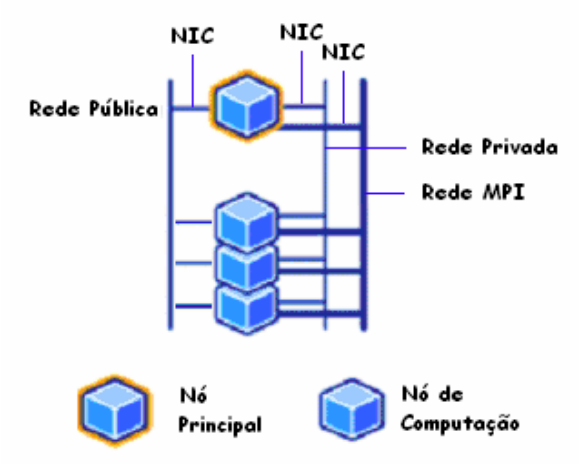

Fig. 13 – Topologia 4(Fonte Adaptado de RUSSEL, 2005)

- Cenário 5: Pode-se observar, na Figura 14, que o Cenário 5 não possui nenhuma rede privada intra-cluster. Todo o tráfego, inclusive intra-cluster, MPI e público, é transmitido através da rede pública. Isso maximiza a acessibilidade, mas à custa de uma redução no desempenho da rede.

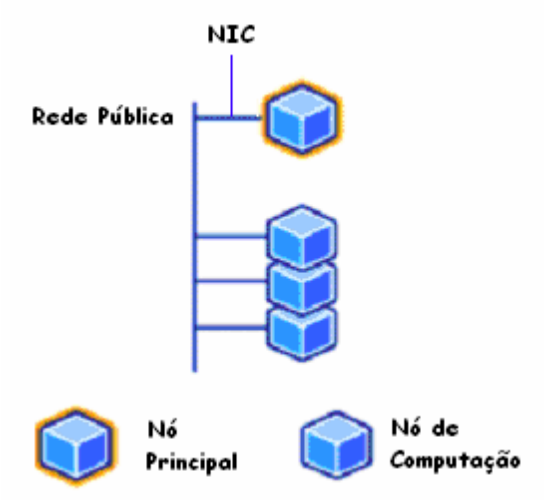

Fig. 14 – Topologia 5 (Fonte Adaptado de RUSSEL, 2005)

As topologias apresentadas diferenciam-se pelo suporte ou não às redes privadas e MPI e pela forma de acesso à rede pública pelos “nós” de computação, para que estes tenham acesso aos controladores de domínio e serviços de rede pública. Em resumo, os diferentes cenários servem para confinar os diferentes tipos de tráfego em redes especializadas, reduzindo o tempo de comunicação.

4.4 GERENCIADOR DE TRABALHOS E TAREFAS

Os usuários podem submeter trabalhos para serem executados no cluster através do Gerenciador de Trabalhos e Tarefas. Um trabalho (*job*) é uma requisição de recursos submetida ao escalonador de trabalhos (*Job Scheduler*), que contém ou conterá uma ou mais tarefas (*task*). Uma tarefa não pode existir sem estar associada a um trabalho. Um trabalho pode conter uma única tarefa. O escalonador de trabalho é o serviço responsável por colocar trabalhos e tarefas em filas, alocar recursos, despachar tarefas para nós de computação, e monitorar o estado de trabalhos, tarefas, e nós. A ordem dos trabalhos na fila obedece a uma política de escalonamento. No Windows® Compute Cluster Server® são suportados três tipos de tarefas: tarefa paralela, varredura paramétrica e fluxo de tarefas, detalhadas a seguir:

- Tarefa Paralela (*Parallel Task*): o trabalho é formado por um conjunto de tarefas que executam simultaneamente trocando mensagens a fim de cooperar para realizar alguma computação. Trata-se de um programa MPI característico de cálculo científico e de engenharia;

- Varredura Paramétrica (*Parametric Sweep*): o trabalho é formado por várias tarefas independentes que não se comunicam, sendo dispensável o uso da biblioteca MPI. Normalmente, cada tarefa possui seu próprio conjunto de dados de entrada e fornece sua saída em arquivos separados.
- Fluxo de Tarefas (*Task Flow*): o trabalho é formado por um conjunto de tarefas, geralmente distintas e independentes, que devem ser executadas em uma ordem determinada devido à algum tipo de dependência de processamento. O programador da aplicação é que determina o tipo de trabalho a ser utilizado, segundo a estrutura de programação paralela desenvolvida.

A designação de uma tarefa para os "nós de" computação é feita pelo escalonador de trabalho, o qual envia para um dos "nós" dentre aqueles designados para o trabalho. A menos que dependências entre as tarefas sejam definidas, estas são servidas segundo a política FCFS (*First-Come First-Served*) (Russel, 2005)

No caso de várias tarefas que não trocam informações entre si (varredura paralela e fluxo de tarefas), estas são alocadas "nó" a "nó", até que os processadores de cada "nó" estejam todos ocupados (RUSSEL, 2005). Assim, se o "nó" designado pelo escalonador de trabalho dispõe de quatro processadores, o máximo suportado pela versão atual do Windows® Compute Cluster Server®, as quatro primeiras tarefas serão alocadas nestes quatro. As quatro seguintes, nos quatro processadores do próximo "nó" designado, até que todos os processadores, em todos os "nós", estejam ocupados. Caso haja mais tarefas que processadores, estas devem aguardar a liberação de um processador.

5 APLICAÇÃO DO CLUSTER

Localizado na UTFPR – campus Curitiba, O Laboratório de Ciências Térmicas - LACIT é um grupo de pesquisa criado em 1999 com atividades na área de Mecânica dos Fluidos, Transferência de Calor e Termodinâmica. O grupo trabalha em diversas áreas de Engenharia Térmica e seus estudos envolvem não só investigação básica, mas também aplicada.

NO LACIT são feitas diversas simulações de escoamentos monofásicos, bifásicos, comportamento de bombas entre outras atividades referentes à área de mecânica dos fluidos. As simulações são realizadas através do software CFX® um programa comercial de CFD (do inglês *Computational Fluid Dynamics*). A figura 16 mostra um exemplo de simulação no CFX.

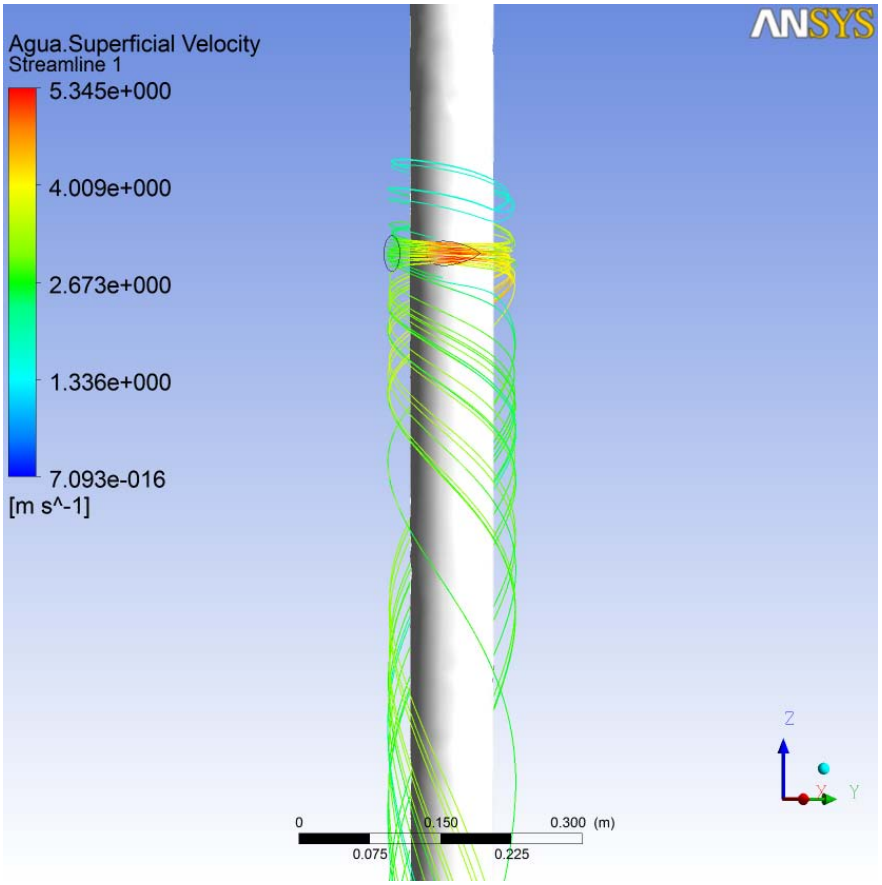

Fig.16 – Tela do CFX (Fonte autoria Própria)

Essas simulações requerem alto poder computacional, pois o computador terá que executar complexos e extensos cálculos matemáticos que exigem alta quantidade de

tempo computacional para serem resolvidos, por exemplo, uma simulação rotineira de um escoamento bifásico, água e gás, leva em uma estação de trabalho com 4GB de memória e processador de 4 núcleos de velocidade de 2.8Ghz cerca de 15 a 20 horas para ser completada; dependendo do refinamento dos cálculos esse tempo pode passar facilmente de 20 para 30 a 40 horas, sendo portanto um inconveniente para o usuário esperar tanto tempo até analisar os resultados da simulação.

Há, portanto, necessidade de um sistema com um poder computacional mais poderoso que permita executar essas simulações em um intervalo de tempo menor, é neste contexto que o cluster HPC entra como solução. Mais detalhe do cluster é mostrado nos capítulos seguintes.

5.1 O CLUSTER HPC

O cluster é composto de um rack com 42 U de altura e dois servidores que podem ser incorporados ao rack como módulos, A figura 17 apresenta detalhes do cluster HPC. Basicamente esses servidores possuem as seguintes características:

- 2 Processadores Intel® Quad Core Intel X5560 Xeon , 2.8GHz, 8M Cache,
- 32 GB de memória DDR-3 Registered DIMM, 1333 MHz (8 x 4 GB)
- 03 discos rígidos de 146GB SAS 3.5 de 15.000 rpm(raid 5)
- 4 interfaces de rede 10/100/1000 UTP integradas
- Fonte Redundante de Alta Potência Energy Smart (870W), Ventiladores redundantes e Hot-swap

Algumas vantagens desse tipo de configuração em relação a clusters HPC de baixo custo como o *Beowulf* que utilizam computadores pessoais são as fontes redundantes e o sistema de refrigeração mais sofisticado além do hardware bem mais robusto. Esses fatores garantem maior confiabilidade que o cluster irá estar funcional por um período maior de horas. Mais detalhes sobre a configuração do *hardware* do cluster se encontram no apêndice B

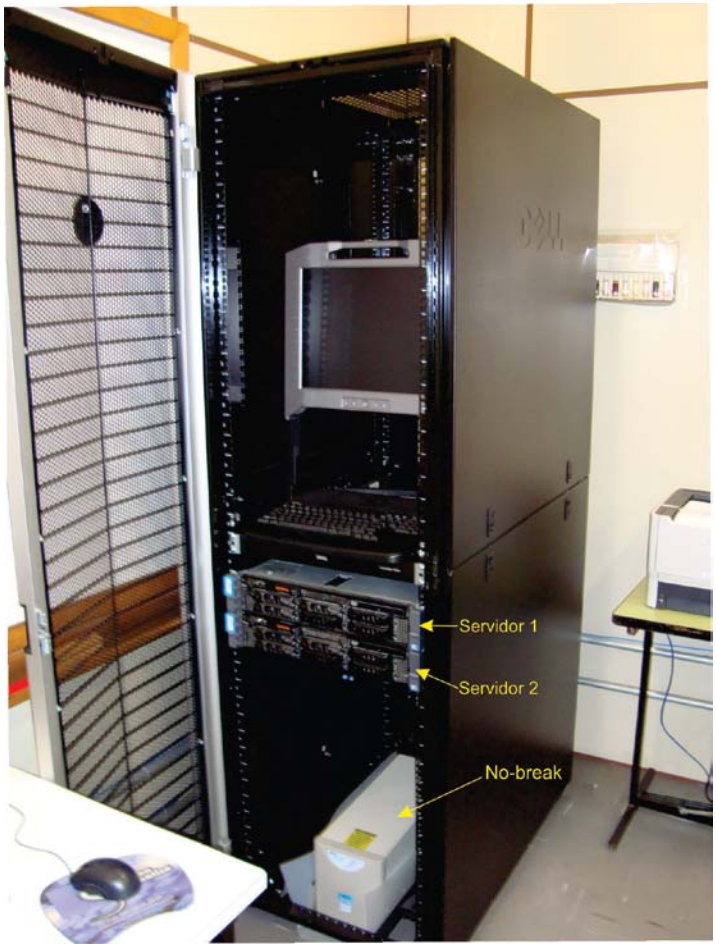

Fig. 17 – Cluster HPC LACIT (Fonte Autoria Própria)

Como os processadores possuem a tecnologia *Hyper-Threading* ou hiper-processamento, tecnologia usada em processadores que o faz simular dois processadores tornando o sistema mais rápido quando se usa vários programas ao mesmo tempo; como cada servidor possui 2 processadores Quad, 4 núcleos, então tem-se em cada processador, 4 núcleos físicos e 4 virtuais portanto 8 processadores físicos e 8 virtuais em cada servidor num total de 16 processadores por servidor, portanto o cluster possui 32 processadores.

5.2 WINDOWS HPC SERVER 2008 R2

O sistema operacional escolhido foi o Windows HPC Server 2008 R2, atualmente é o sistema HPC mais moderno da Microsoft. Pode-se fazer *download* gratuitamente por um período de 6 meses no seguinte site: http://www.microsoft.com/hpc/en/us/trial/what-is-hpc.aspx.

Os detalhes do processo de instalação serão descartados, pois se foge do escopo do trabalho, informações mais detalhadas da instalação podem ser obtidas no link acima.

Depois de instalado o Windows® deve-se realizar os seguintes passos:

- Instalar o Windows® HPC Pack 2008 R2 SP 2 , o qual contem as ferramentas necessárias para utilizar o cluster,
- Configurar um domínio para o nó principal e ingressar os nós computacionais neste domínio, detalhes de como configurar um domínio podem ser vistos no apêndice B;
- Configurar uma topologia mais apropriada ao cluster, detalhes dos tipos de topologias podem ser vistos no capitulo 4; através do assistente de instalação do Windows® HPC Pack 2008, detalhes do procedimento de instalação se encontra no apêndice C;

A topologia escolhida foi a de tipo 3 pois foi a que melhor se adequou as necessidades do LACIT. A figura 18 esclarece melhor o contexto

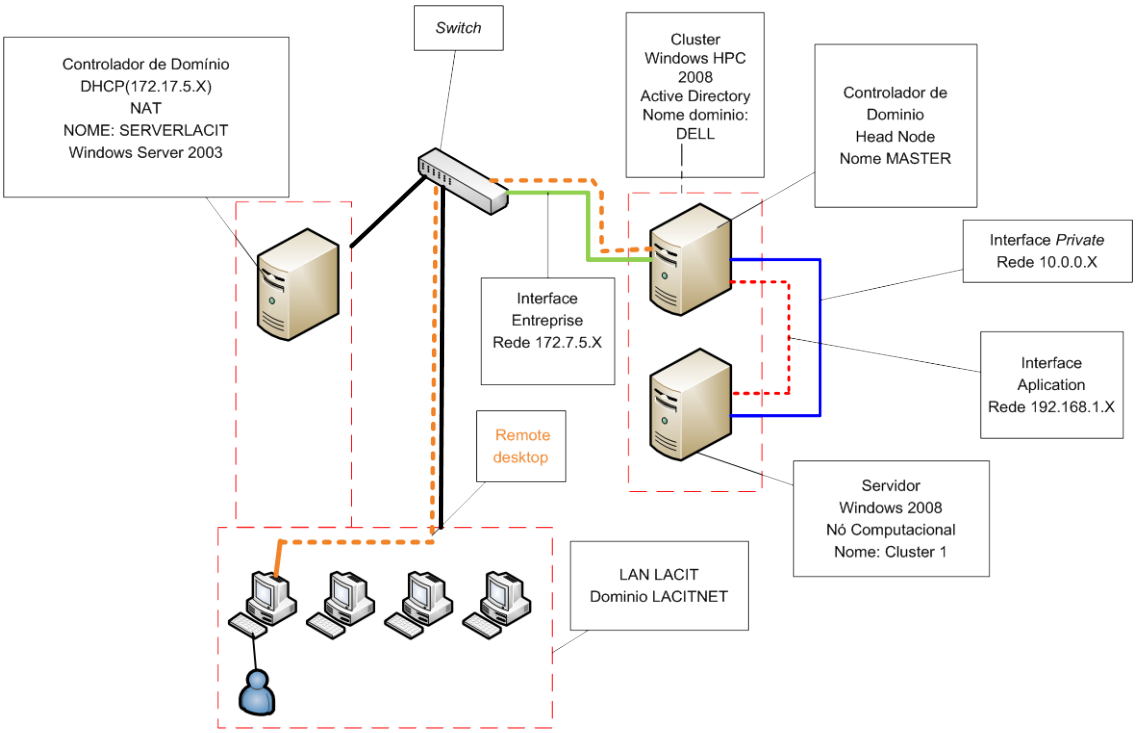

Fig. 18 – Topologia Rede Cluster na rede LACITNET (Fonte Autoria Própria)

Alguns detalhes da figura 5-1 devem ser mais bem descritos:

O servidor SERVERLACIT é um controlador de domínio (LACITNET) baseado na plataforma Windows® Server 2003 *standard*, também realiza funções de DHCP (do inglês, *Dynamic Host Configuration Protocol*), servidor de arquivos e rote amento da LAN LACIT com a Internet através de NAT; possui as seguintes características de hardware:

- Processador Intel Pentium D 3.2GHZ
- 2GB de memória DDR2 600
- Dois discos de 250GB em Raid1

Os computadores que formam a LAN (do inglês *Local Area Network*) do LACIT usam o sistema Windows® 7 Professional; possuem modernos processadores de 4 núcleos e mínimo de 4GB de memória RAM. Essas configurações são exigidas visto que mesmo com o cluster funcional ainda são realizadas simulações em estações de trabalho.

O cluster esta em seu próprio domínio, DELL, e o controlador do domínio é chamado MASTER este exerce também a função de *head node* do cluster. No domínio DELL esta incorporado o servidor chamado CLUSTER 1 com a função de nó computacional.

Existe, portanto dois domínios na mesma rede, porém não há relações de confiança entre eles, ou seja, se um usuário da rede LAN LACIT, domínio LACITNET, quiser acessar o cluster tal usuário deverá estar cadastrado no domínio DELL. Isto não é necessariamente um problema visto que existem poucos usuários que utilizam o cluster

O *remote Desktop Protocol* (ou somente RDP), como mostra afigura 19 é um protocolo multicanal que permite que um usuário conecte a um computador rodando o Microsoft® *Terminal Services*. Existem clientes para a maioria das versões do Windows, e outros sistemas operacionais como o Linux. O servidor escuta por padrão a porta TCP 3389. Por razões de segurança o suporte a *remote desktop* vem desabilitado na família Windows , deve portanto ativá-lo no servidor MASTER para conseguir usar esse protocolo corretamente.

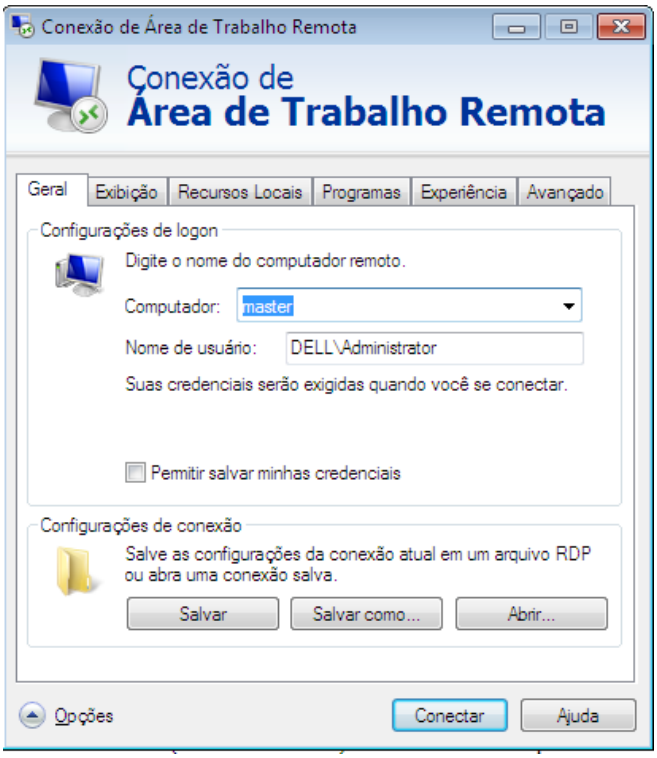

Fig. 19 – Área de trabalho Remota (Fonte: Autoria Própria)

Para acessar o cluster o usuário deve executar o *Remote Desktop*. O nome de usuário e senha são aqueles que estão cadastrados em MASTER, que é o controlador de domínio do cluster. Após a autenticação o usuário tem acesso ao desktop do *head node* e pode executar as simulações através do programa CFX® e demais aplicações necessárias que devem já estar previamente instaladas e configuradas. Um detalhe importante é que o independente do número de nós computacionais disponíveis o numero de nós que o programa CFX pode distribuir é limitado pela sua licença de us;. no LACIT a licença adquirida esta restrita a 12 processadores, portanto, mesmo que o cluster tivesse mais de 12 processadores disponíveis licença impediria de usa-los.

5.3 GERENCIAMENTO BÁSICO DO CLUSTER

Para Gerenciar o cluster usa-se o programa HPC 2008 Cluster Manager, nele é possível obter informações sobre a topologia de rede usada, estado dos nós do cluster: *online* ou *offline*, e nível de uso do processador em cada nó. Basicamente essas informações foram as mais usadas, porém o programa possui uma gama maior de informações disponíveis para um completo gerenciamento do cluster, as figuras 20 e 21 mostram algumas características desse gerenciador.

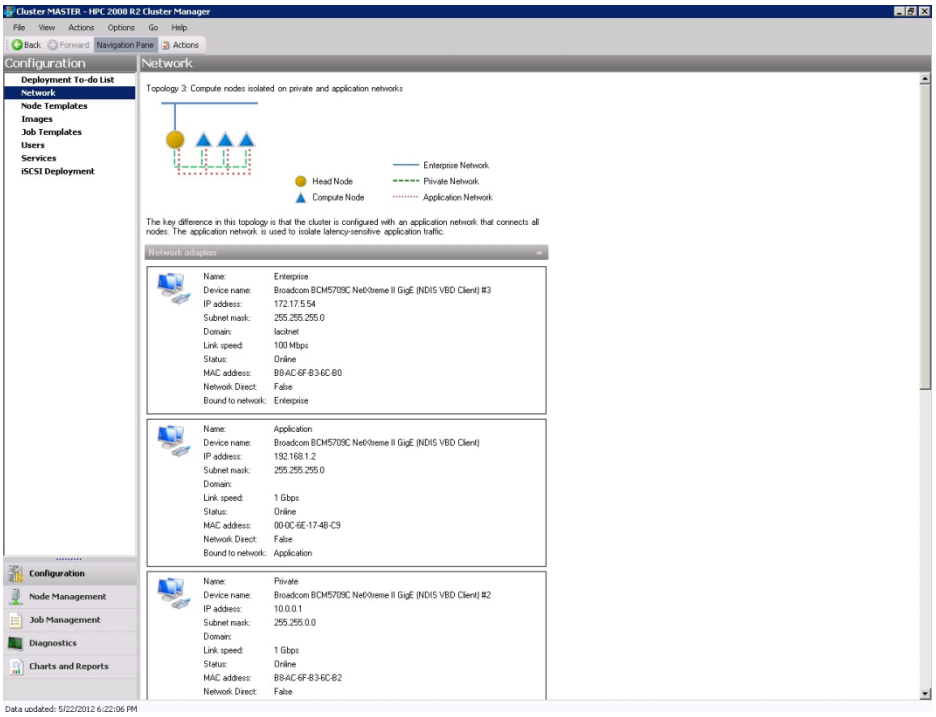

Fig. 20 – Topologia do cluster HPC apresentada no *HPC 2008 Cluster Manager* (Fonte Autoria Própria)

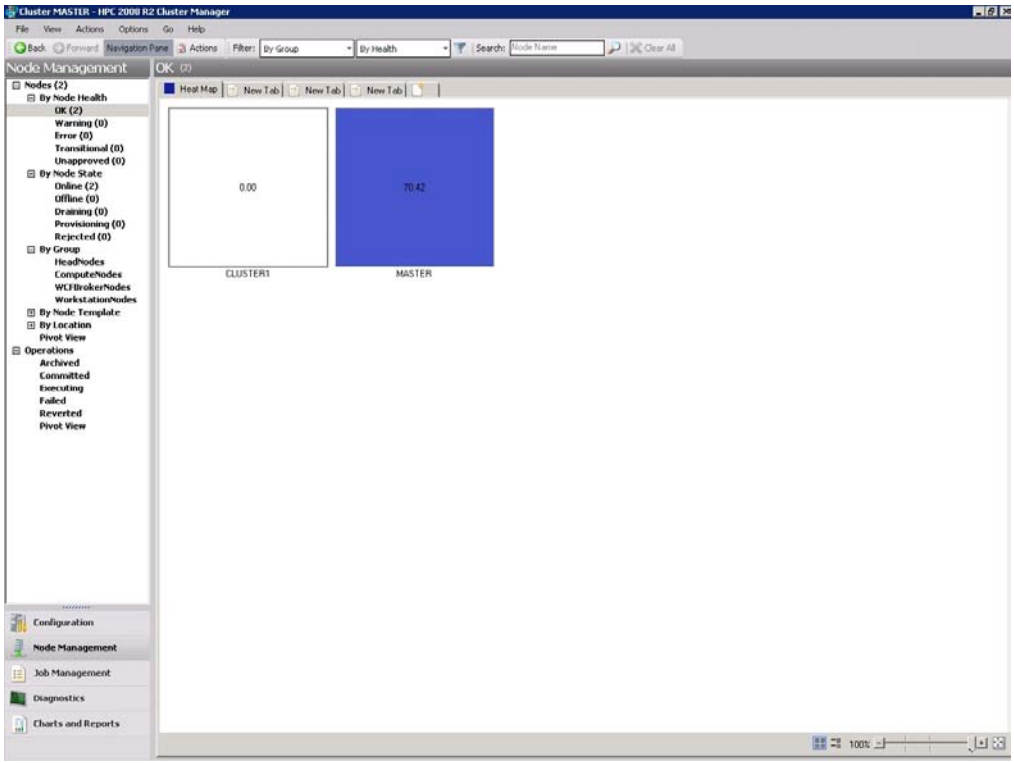

Fig. 21 – Gerenciamento de nós do cluster (Fonte Autoria Própria)

No *heat Map* da figura 21 é possível ver o quanto de utilização de processamento cada nó computacional esta exercendo no momento e seu estado operacional.

5.4 UTILIZANDO O CLUSTER COM O CFX

O CFX possui suporte a MPI, que permite a distribuição das tarefas computacionais entre os nós do cluster como mostrado na figura 22.

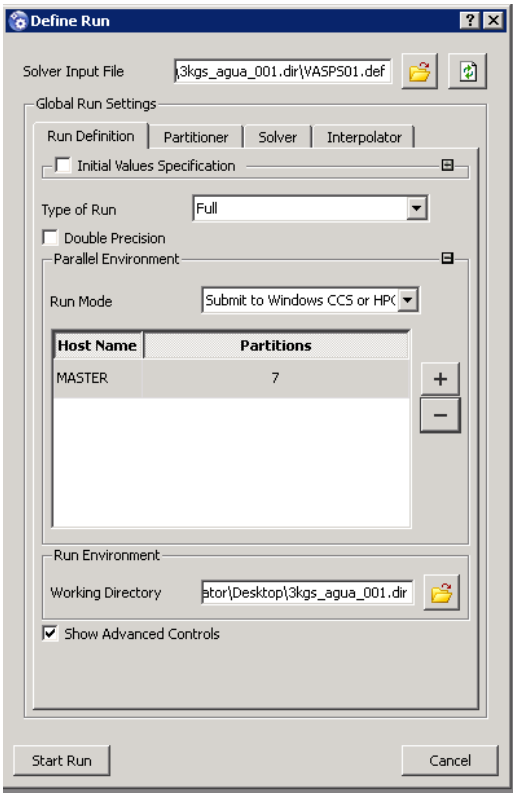

Fig. 22 – Tela MPI CFX (Fonte Autoria Própria).

Após abrir a simulação desejada clica-se em *define run* onde é possivel definir o número de processadores que serão usados para resolver as tarefas computacionais: em *type of run* escolhe-se full em *run mode* coloca-se *Submit* to Windows® CSS

Na figura 23 pode-se ver a distribuição de tarefas entre os nós

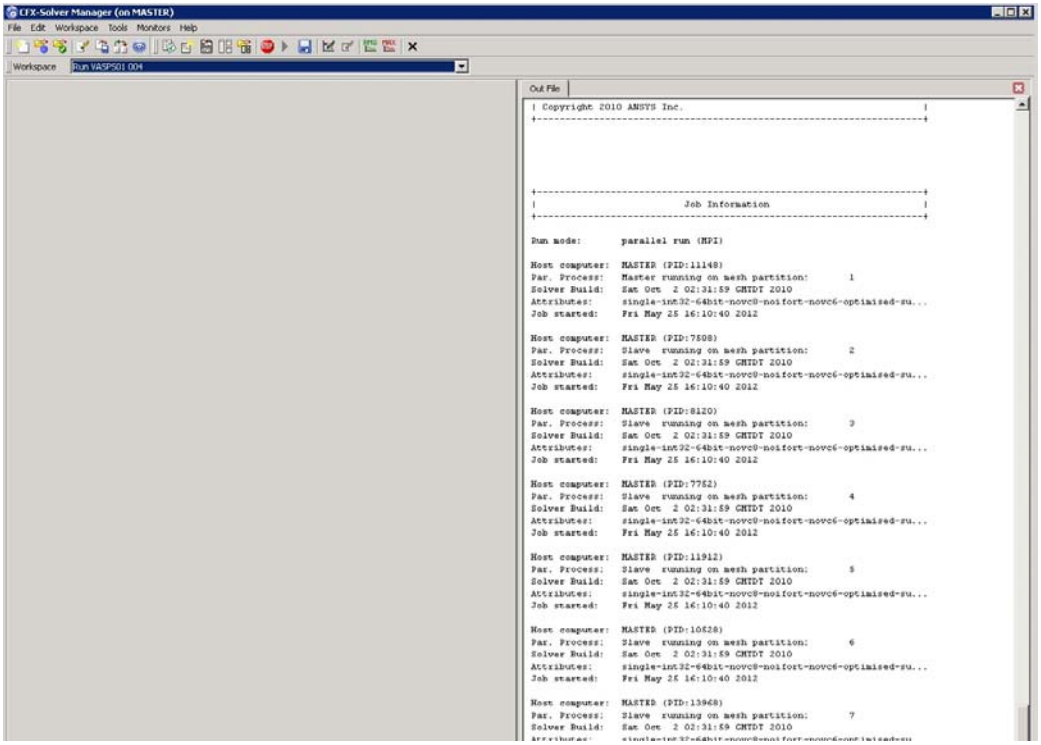

Fig. 23 – Distribuição dos Jobs entre os nós (Fonte Autoria Própria).

5.5 DESEMPENHO DO CLUSTER HPC

Para uma breve análise de desempenho do cluster foram feitas simulações em um computador pessoal com uma configuração de hardware superior trata-se de um computador com:

- Processador intel® Core I7 de 2,8GHz
- 12 GB Memoria DDR 3 de 1600MHz

Foram realizadas três simulações, um escoamento monofásico, água, um bifásico, água e gás e uma simulação do comportamento do escoamento nas pás de uma bomba centrípeta.

Essas simulações foram realizadas no computador pessoal citado acima e no cluster com o intuito de análise de desempenho. A tabela 1 baixo mostra os resultados.

Simulação	Tempo computador pessoal	Tempo no cluster
Escoamento monofásico	3 horas	1 hora e 40 minutos
Escoamento bifásico	10 horas	6 horas e 50 minutos
Escoamento bomba	18 horas	15 horas

Tabela 1 - Desempenho do Cluster (fonte: Autoria Própria)

Pode-se ver o ganho de desempenho do cluster, um dos fatores que limitam um desempenho melhor é a licença do CFX que só permite 12 processadores e outro é um estudo mais detalhado de como o CFX distribuiu os *Jobs* entre os clusters.

6 CONCLUSÃO

Dependendo da aplicação, simular numericamente requer alto esforço computacional. O desenvolvimento científico e tecnológico na área de informática levou ao surgimento de supercomputadores. Porém, sua utilização é inviável devido ao alto custo associado a esta tecnologia.

Com o intuito de encontrar uma solução alternativa viável financeiramente, estudos sobre computação de alto desempenho – HPC (do inglês, *High Performance Computing*) foram realizados. A partir destes estudos, foi proposta uma nova arquitetura com alto poder computacional, o cluster. O cluster é um sistema que compreende um "aglomerado" de computadores pessoais ou servidores dedicados que trabalham em conjunto para executar aplicações ou realizar outras tarefas, mas, que é visto como um recurso único pelo usuário. Devido a estas características estes sistemas têm sido objetos de diversos estudos.

Como resultado de um destes estudos, a Microsoft® lançou em o Windows® *Compute Cluster Server* que é um sistema operacional voltado à implantação, configuração e gerenciamento de cluster. Windows® Compute Cluster Server possui várias ferramentas que facilitam o gerenciamento do cluster. Apesar de se tratar de um programa proprietário o uso do Windows® Compute Cluster Server permite o usuário novato nos fundamentos de cluster entender o funcionamento destes à luz das tecnologias da Microsoft®, porém os conceitos intrínsecos dos clusters como nós, *head-nodes*, e escalonamento de tarefas são pertinentes também à plataforma GNU/Linux, portanto, no futuro tal usuário poderá implantar clusters baseados em sistemas Linux com um conhecimento mais sólido sobre *clusters*. Em relação ao cluster HPC construído neste trabalho há algumas considerações importantes a serem mencionadas:

Por se tratar de um trabalho inicial pode-se caracterizar como sendo um primeiro passo as etapas realizadas neste trabalho, da teoria a pratica, pois se conseguiu montar a estrutura básica de um cluster HPC, o passo seguinte prevê algumas melhorias a serem feitas para atingir um nível mais elevado de aprimoramento, são elas:

- Compra de mais servidores para adiciona-los como nós computacionais ao cluster;

- Compra de uma licença HPC de uso do CFX® de pelo menos 80 processadores;
- Estabelecer uma relação de confiança entre o domínio DELL e o domínio LACITNET para facilitar a administração de novos usuários e melhorar a gerencia entre eles;
- Estudo mais aprimorado do *Job management* para programas do CFX e futuramente de programas de linguagem Fortran relevantes às simulações de CFD.
- Implantar o uso de estações de trabalho ociosas como nós computacionais para melhorar o desempenho do cluster

REFERÊNCIAS

JUNIOR, F.H.C. **Computação de Alto Desempenho em Plataforma Windows,** Artigo, Universidade Federal do Ceará (UFC), 2004.

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

SNOW, C.R. **Concurrent Programming**, Cambridge University Press, 1992.

IGNACIO, A.A.V. **MPI: Uma ferramenta para implementação paralela**, Pesquisa Operacional, Rio de Janeiro, v. 22, n. 1, 2002.

PACHECO, P. S. **A user guide to MPI**. Technical Report, San Francisco, CA,USA, 1995.

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

APÊNDICE A – ESPECIFICAÇÃO HARDWARE DO CLUSTER

SERVIDOR DELL POWEREDGE R710 (LOTE 03 / ITEM 01 - Dell PowerEdge R710)

• 2 Processadores Quad Core Intel X5560 Xeon , 2.8GHz, 8M Cache, 6.40 GT/s QuickPath Interconnect,

Tecnologia Turbo Hyper-Threading

• 32 GB de memória DDR-3 Registered DIMM, 1333 MHz (8 x 4 GB), 2R para

• 2 processadores, Advanced ECC

• 03 discos rígidos de 146GB SAS 3.5" de 15.000 rpm

• Configuração dos Discos em RAID 5

• Placa SD card Vflash 1GB para controladora iDRAC6 Enterprise

• 2 Hbas Qlogic 2460 *Single Port, Fibre Channel* 4Gbps, PCI-e

• Placa controladora de array interna PERC6i com 256MB de cache e bateria (Raid 0, 1, 5, 6, 10, 50, 60)

• Placa de gerenciamento remoto iDRAC6 Enterprise

• 4 interfaces de rede 10/100/1000 UTP integradas

• Riser com 1 slot PCIe x16 e 2 slots PCIe x4

• Cabos de força C13-C14, 12 A, 4 metros

• Software de gerenciamento Dell *OpenManage* (DVD e documentação)

• Console de Gerenciamento Dell

• Fonte Redundante de Alta Potência *Energy Smart* (870W), Ventiladores redundantes e Hot-swap

• Unidade de DVD ROM de 16x, SATA

• Sem sistema operacional

• Não inclui teclado, mouse, monitor

• Instalação on-site não inclusa

• Trilhos deslizantes para rack padrão 19” com braço de gerenciamento de cabos

• 5 anos de garantia

VALOR UNITÁRIO – R$ 24.154,53

VALOR TOTAL – R$ 48.309,06

RACK DELL

• Rack com 42U de altura

• Dimensões do equipamento - Largura: 60,5 cm; Profundidade: 107 cm;

• Altura: 200 cm; Peso: 225 Kg.

• 4 Réguas de energia 12 amperes 110/220V com 7 conectores IEC C13 cada

• Gaveta de 1U com monitor LCD 17", teclado US e touchpad, conexões USB

• Estabilizadores laterais para Rack

• SwitchBox de teclado/monitor/mouse de 8 portas UTP

• Gabinete de 1U com trilhos para rack padrão 19”

• 8 cabos CAT5 2,1m

• 8 módulos de conversão KVM USB-UTP para *switch Box*

• 5 anos de garantia

VALOR UNITÁRIO – R$ 18.900,00

APÊNDICE B – INSTALAÇÃO DO ACTIVE DIRECTORY

O *Active Directory* é uma implementação de serviço de diretório no protocolo LDAP que armazena informações sobre objetos em rede de computadores e disponibiliza essas informações a usuários e administradores desta rede. É um *software* da Microsoft utilizado em ambientes Windows.

O *Active Directory*, AD, a exemplo do NIS, surgiu da necessidade de se ter um único diretório, ou seja, ao invés do usuário ter uma senha para acessar o sistema principal da empresa, uma senha para ler seus e-mails, uma senha para se logar no computador, e várias outras senhas, com a utilização do AD, os usuários poderão ter apenas uma senha para acessar todos os recursos disponíveis na rede. Pode-se definir um diretório como sendo um banco de dados que armazena as informações dos usuários.

O AD surgiu juntamente com o Windows 2000 Server. Objetos como usuários, grupos, membros dos grupos, senhas, contas de computadores, relações de confiança, informações sobre o domínio, unidades organizacionais, etc, ficam armazenados no banco de dados do AD. Além de armazenar vários objetos em seu banco de dados, o AD disponibiliza vários serviços, como: autenticação dos usuários, replicação do seu banco de dados, pesquisa dos objetos disponíveis na rede, administração centralizada da segurança utilizando GPO, entre outros serviços. Esses recursos tornam a administração do AD bem mais fácil, sendo possível administrar todos os recursos disponíveis na rede centralizadamente.

Etapa 1 - Executar o DCPROMO

a) Clique no menu Iniciar, escolha a opção "Executar..." / Digite: dcpromo / Clique no botão "OK"
b) A janela do "Assistente para instalação do Active Directory" irá aparecer. Clique no botão "Avançar".
c) Na janela de "Compatibilidade de sistema operacional" leia os requisitos mínimos dos clientes do AD. A seguir, clique no botão "Avançar".
d) Na janela de "Tipo de controlador de domínio", selecione a opção "Controlador de domínio para um novo domínio" e clique no botão "Avançar".
e) Na janela de "Criar novo domínio", selecione a opção "Domínio em uma nova floresta" e clique no botão "Avançar".

f) A janela de "Novo nome de domínio" é a opção mais importante na criação do AD. Como todo o sistema do AD é baseado no DNS, a criação do nome de domínio irá afetar toda a operação da rede.
g) Entre com o nome DNS completo do domínio, por exemplo: meudominio.com.br

Clique no botão "Avançar".

h) Este parte poderá demorar alguns minutos, pois o sistema irá procurar pelo servidor DNS e verificar se o nome já existe.
i) Na janela de "Nome do domínio NetBIOS", aceite a opção padrão (que é o primeiro nome do domínio DNS) e clique no botão "Avançar".
j) Na janela de "Pastas do banco de dados e log", lembre-se que a partição deverá ser NTFS e você somente deverá alterar os caminhos padrões por motivos de desempenho.

O caminho "\Windows\NTDS" é o local onde serão armazenados os dados do AD.

k) Aceite as opções padrões e clique no botão "Avançar".
l) Na janela de "Volume de sistema compartilhado", a partição também deverá ser NTFS e somente deverá ser alterado caso haja problemas de desempenho.
m) O caminho "\Windows\SYSVOL" é o local onde serão armazenados as GPOs e scripts do AD e esta pasta é replicada para todos os outros DC.
n) Aceite a opção padrão e clique no botão "Avançar".

Passo2: Lembre-se que o servidor DNS requerido pelo AD deve aceitar registro SRVs e atualizações dinâmicas.

Portanto, o mais recomendável é utilizar o servidor DNS do Windows Server 2003 e deixar que o assistente faça a instalação e configuração do mesmo.

a) Selecione a opção "Instalar e configurar o servidor DNS neste computador e definir este computador para usar o servidor DNS como seu servidor DNS preferencial" e clique no botão "Avançar".
b) Na janela de "Permissões", selecione a opção "Permissões compatíveis somente com os sistemas operacionais de servidor Windows 2000 ou Windows Server 2003" e clique no botão "Avançar".

Esta opção somente deverá ser alterada caso você tenha DCs rodando em plataforma Windows NT, o que não é o caso do presente trabalho.

c) Na janela de senha, digite e confirme a senha de administrador do modo de restauração; clique no botão "Avançar".
d) Esta senha é importante, pois ela não é a mesma senha do administrador do DC e deve ser usada quando houver problemas no DC ou quando o DC for removido do computador.
e) Na janela de "Resumo", verifique as opções selecionadas. Caso as opções estejam corretas, clique no botão "Avançar".
f) Você irá acompanhar o assistente executando as tarefas solicitadas.
g) Caso ocorra algum erro, aguarde o assistente finalizar e depois execute-o novamente para desfazer as alterações.
h) Clique no botão "Concluir".

Você precisará reiniciar o computador para iniciar o AD. Clique no botão "Reiniciar agora".

APÊNDICE C - CONFIGURAÇÃO BÁSICA DO CLUSTER HPC

Instalação das interfaces de rede ao cluster

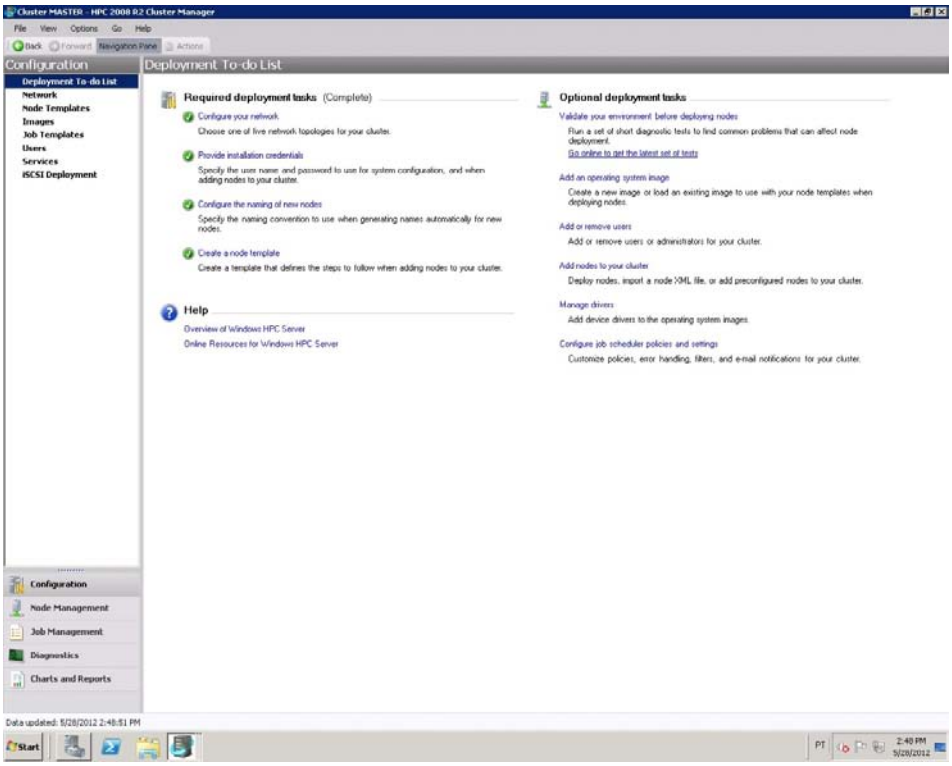

Fig. A1-Tela do assistente de configuração do cluster

Passo 1- no Windows HPC cluster manager, clica-se em configure *your network*, link localizado em *deployment to list* como aparece em destaque na figura A1;

Passo 2- A primeira tela é a de configuração da topologia, conforme visto na figura A2;

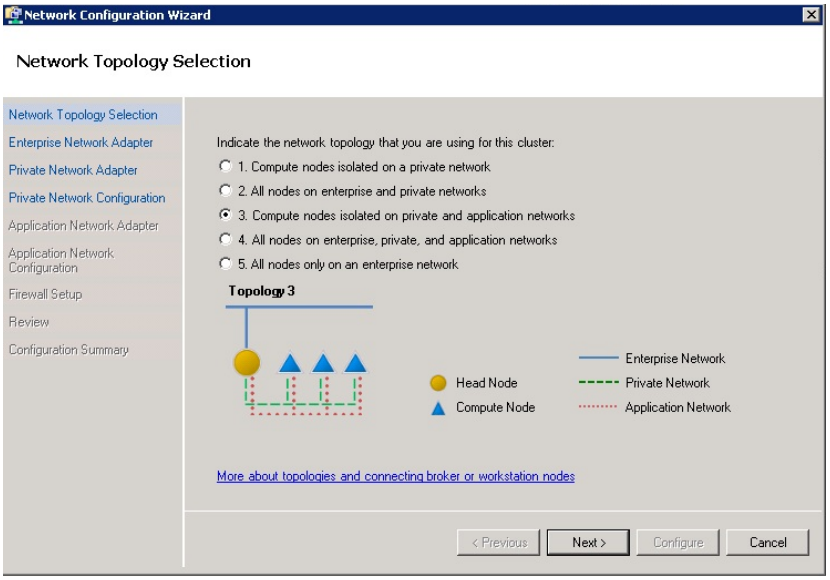

Fig. A2 - Tela de configuração de topologia

Passo 3: Após escolhido a topologia configura-se as interfaces de rede como visto nas figuras A3, A4 e A5. Como a interface Enterprise está configurada para obter um endereço IP automaticamente, as configurações de IP que aparecem na Figura .A3 são referentes ao servidor DHCP do servidor Serverlacit

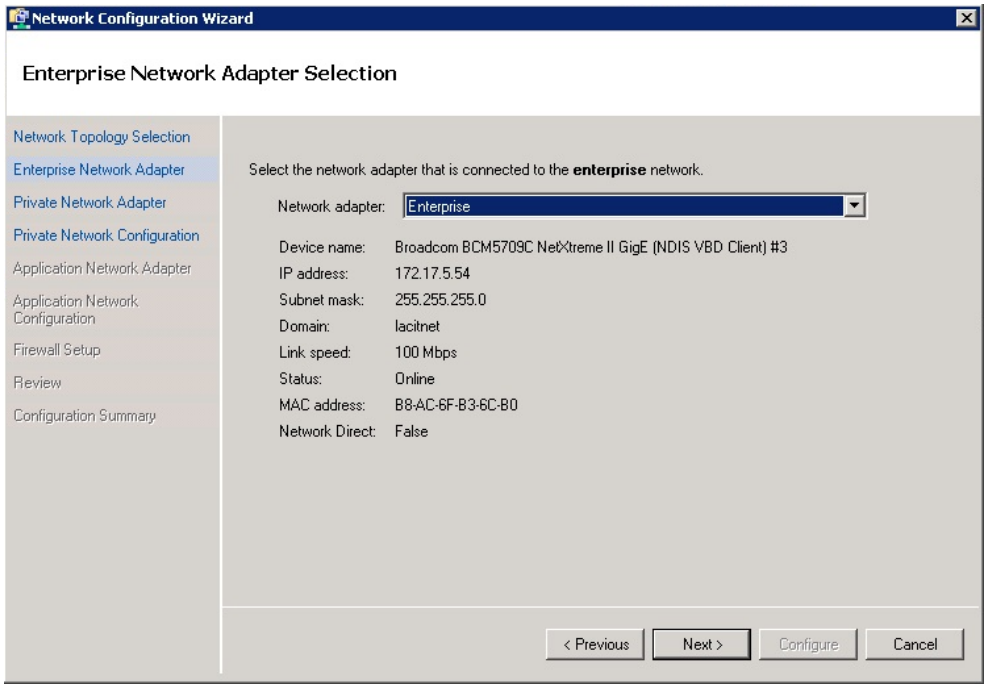

Fig. A3 – Status da interface *Enterprise*

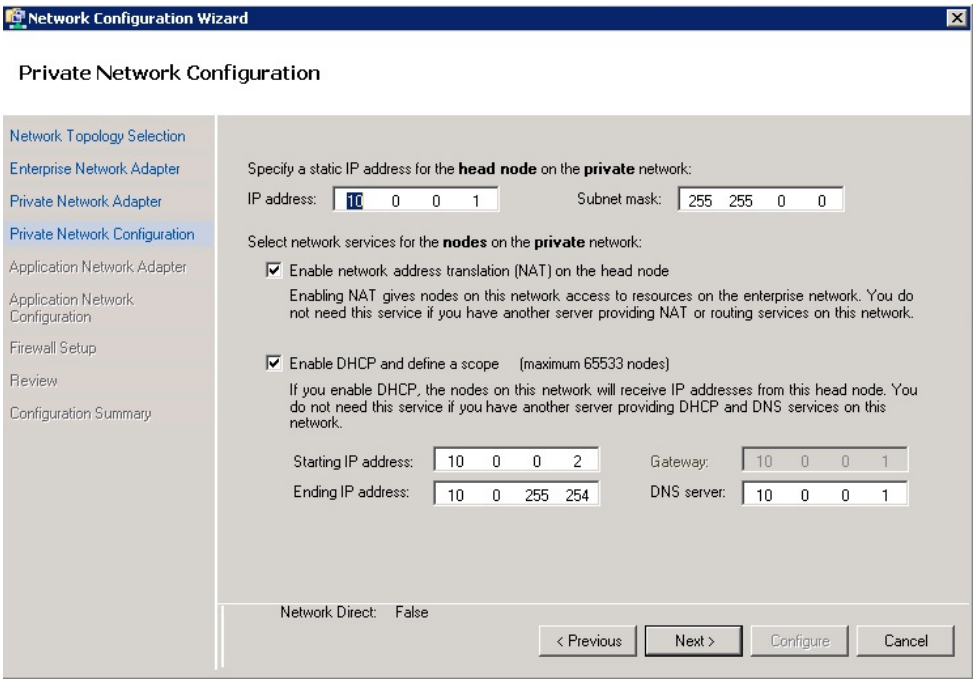

Fig. A4 – Configurações da Interface *Private*

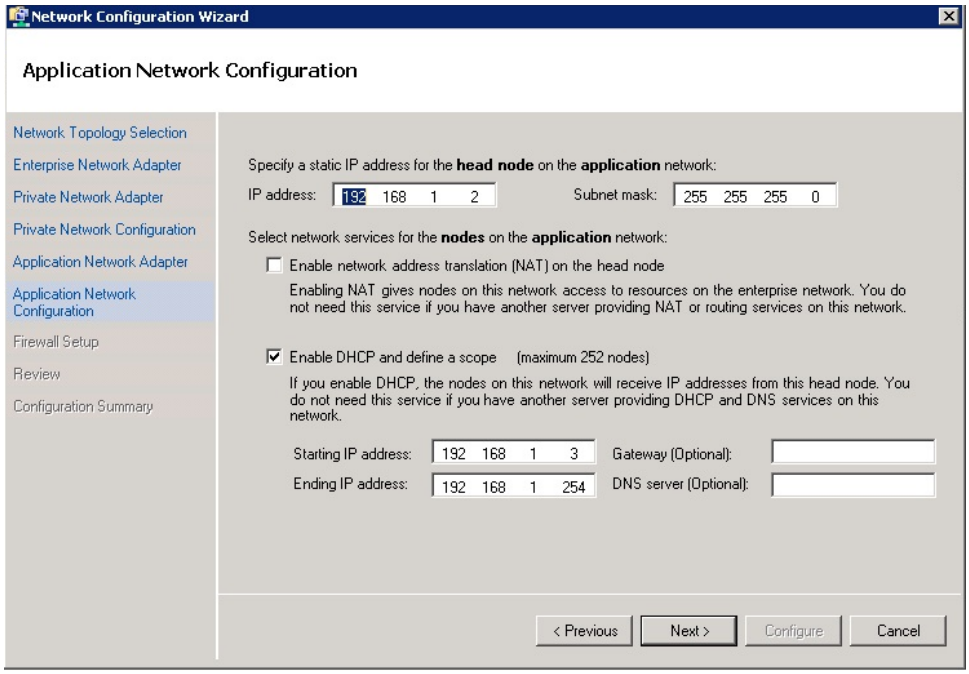

Fig. A5-Configuração da interface *Application*

Passo 4 : Depois se configura o firewall como visto na figura A6;

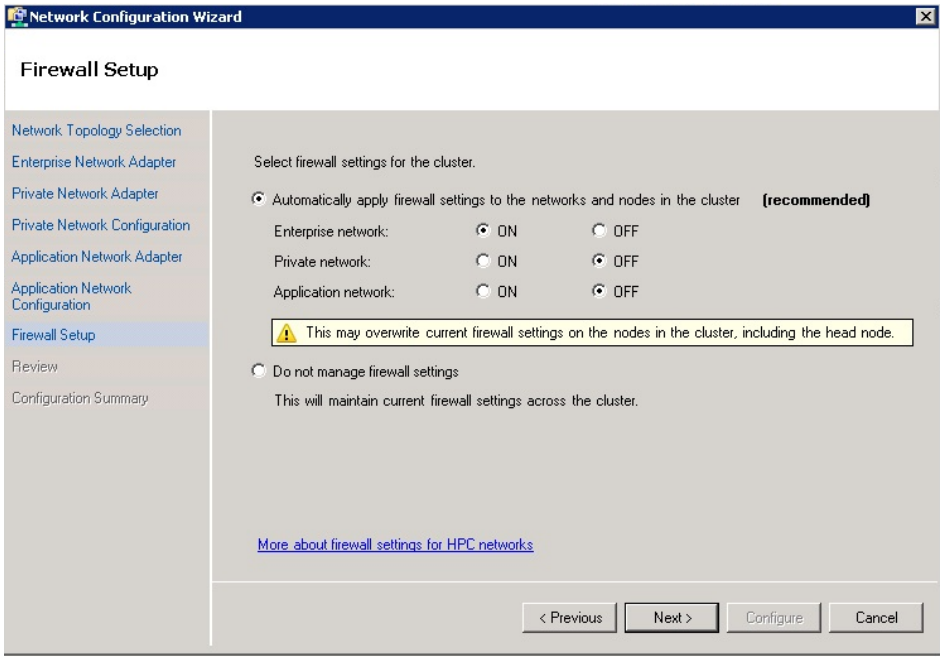

Fig. A6 – Configuração *Firewall*

Passo 5: Após realizadas as configurações aparecerá a janela de *configuration Summary* como visto na figura A7;

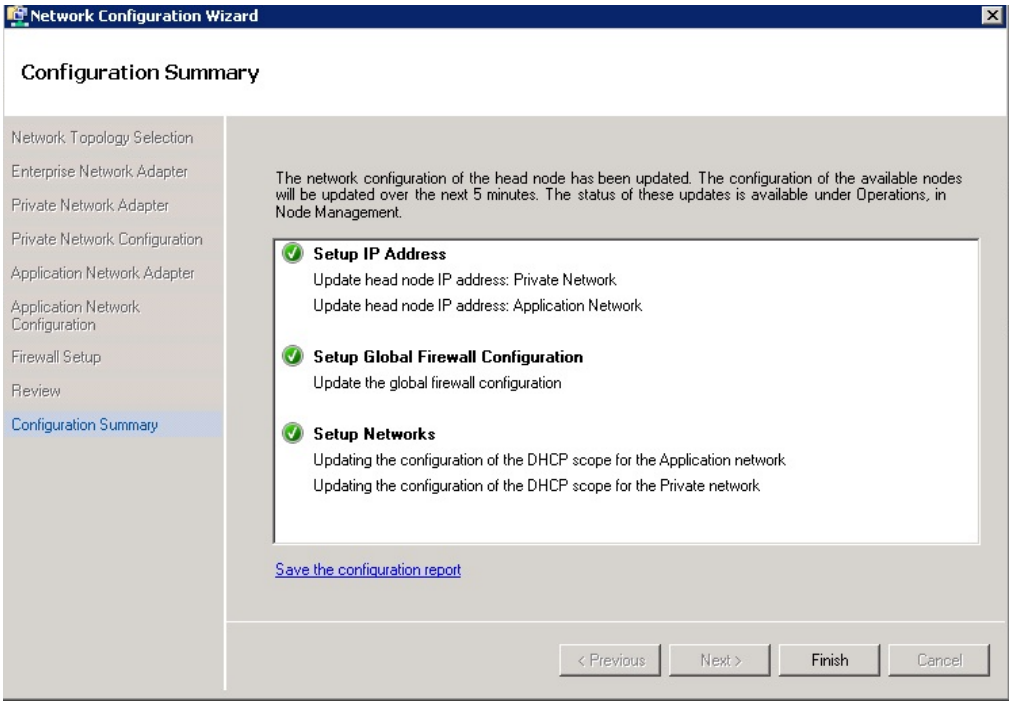

Fig. A7 - Status das configurações das interfaces

Procedimentos para Adicionar nó ao cluster

Passo 1: Clique em *Add Node* conforme figura A8. É importante salientar que o controlador de domínio será o *Head Node* e cada nó computacional deverá ter instalado o HPC PACK. Ao instalar o HPC PACK no controlador de domínio este será automaticamente configurado como *head node*.

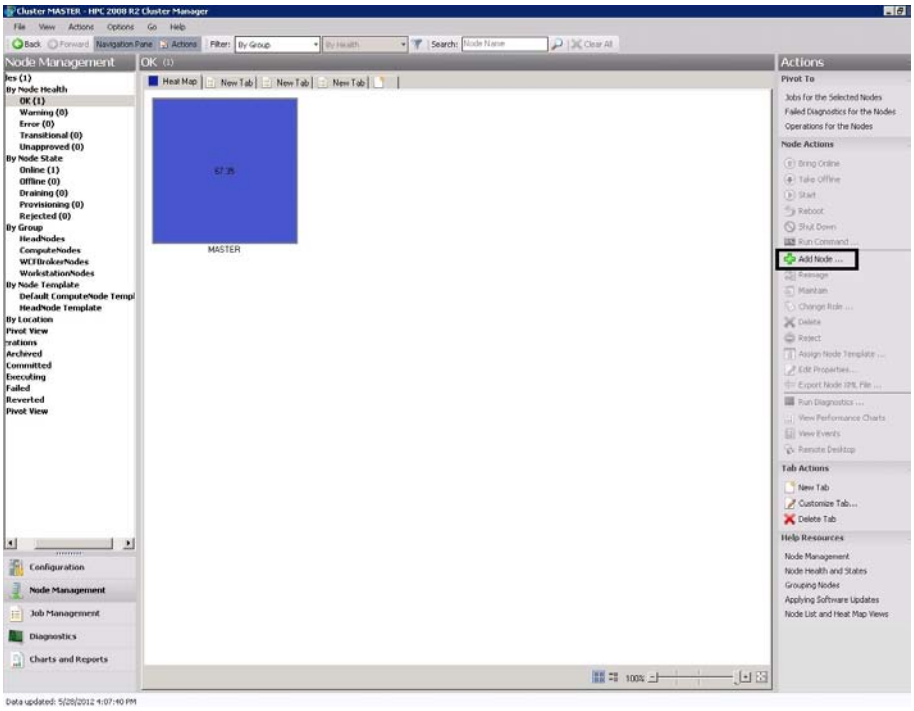

Fig. A8 – Tela inicial do *Node Management*

Passo 2: Aparecerá a tela da figura 2, marque a opção *Add compute nodes or brokes nodes that already been configured;*

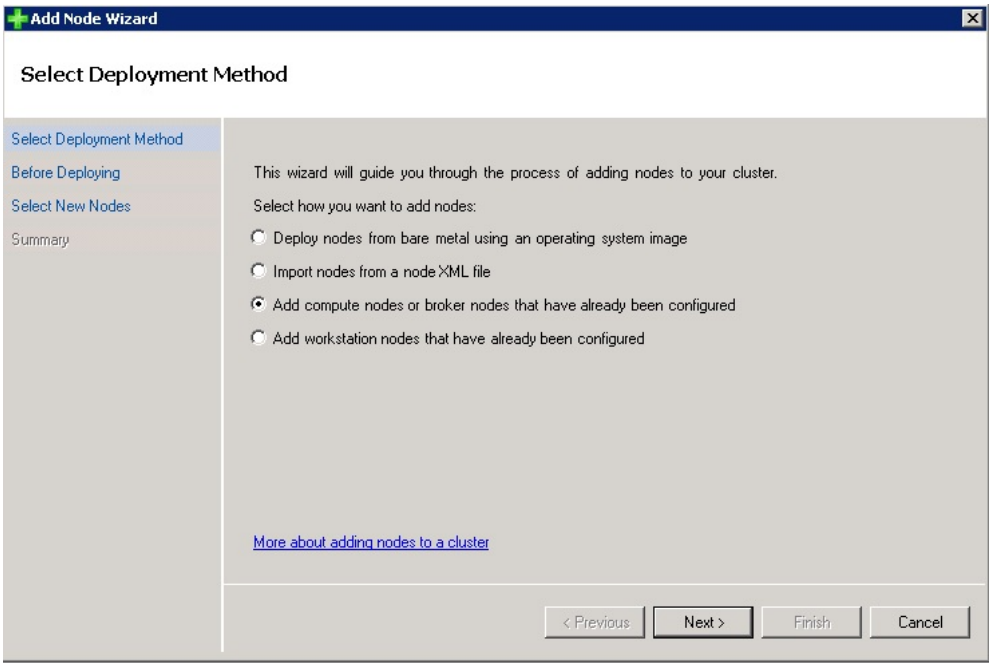

Fig. A9 – Tela para selecionar os tipos de nós a serem adicionados

Passo 3: A figura A10 mostra as condições para se adicionar nós pré-configurados, clique em *next*;

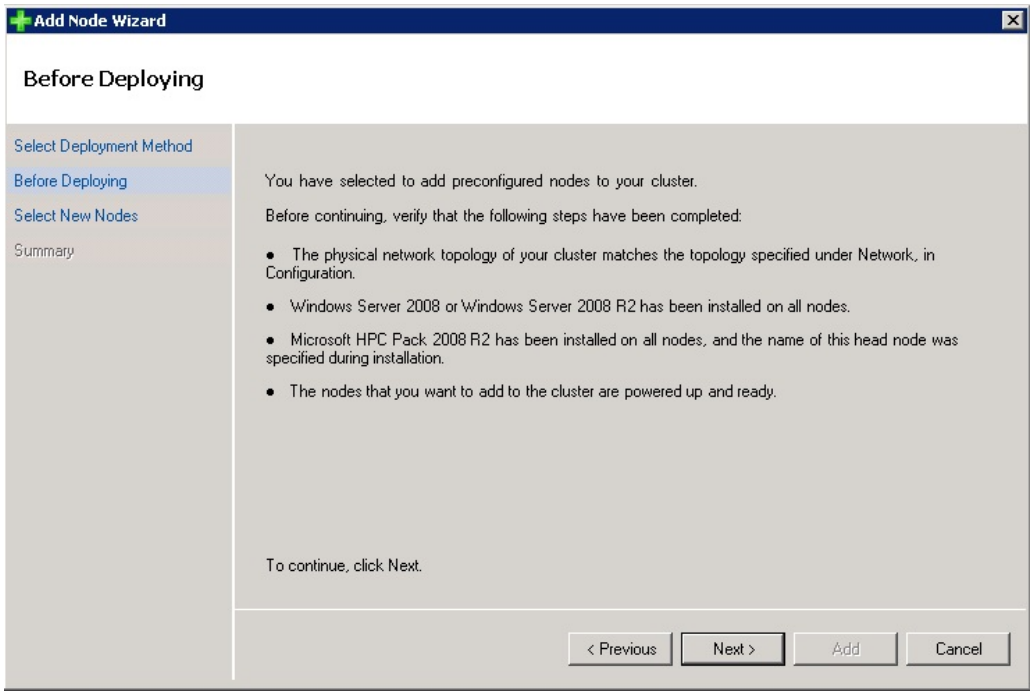

Fig. A10 – Condições necessários para se adicionar nós pré-configurados

Passo 4: Se tudo estiver corretamente configurado aparecerá a lista de nós computacionais disponíveis como aparece na Figura A11;

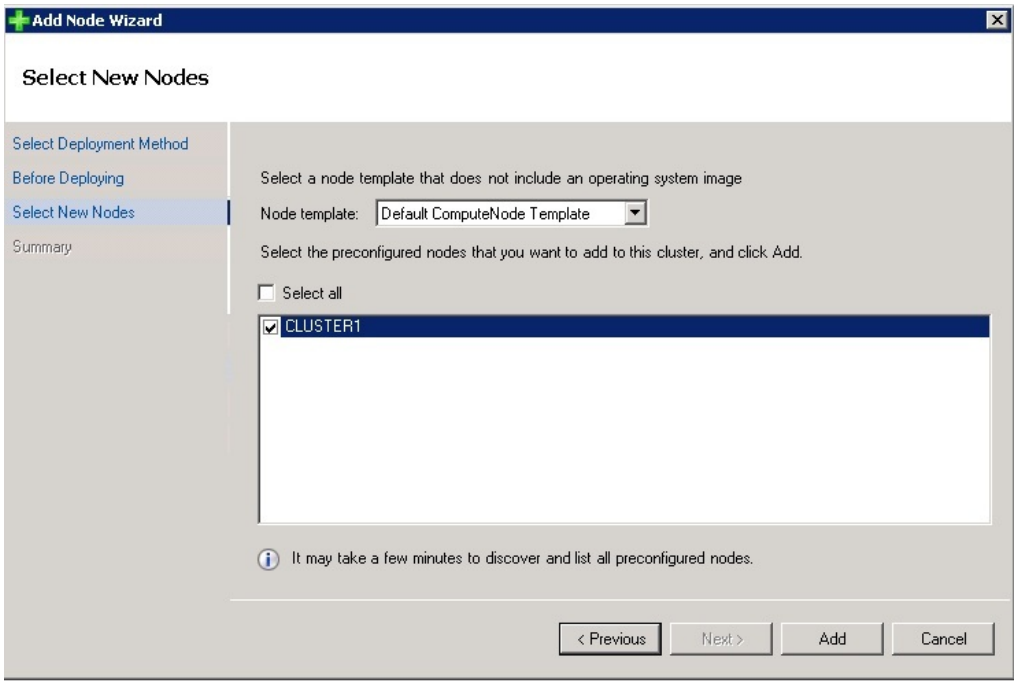

Fig. A11 – Adição dos nós pré-configurados

Passo 5: Após todas as configurações realizadas clique *finish* na tela que aparece na figura A12; a figura A13 mostra o nó computacional já adicionado ao cluster.

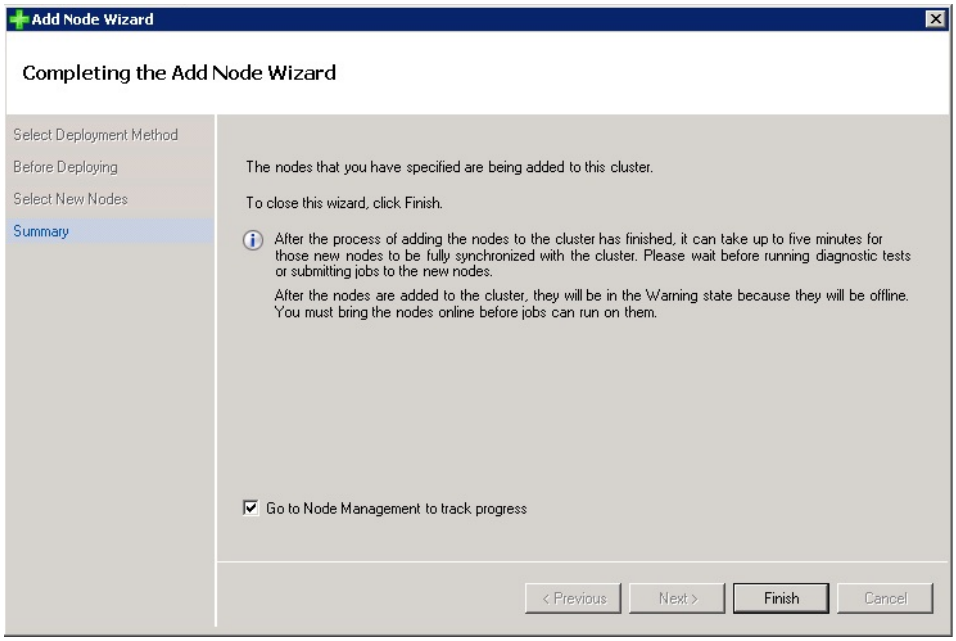

Fig. A12 – Completando a adição do nó ao cluster

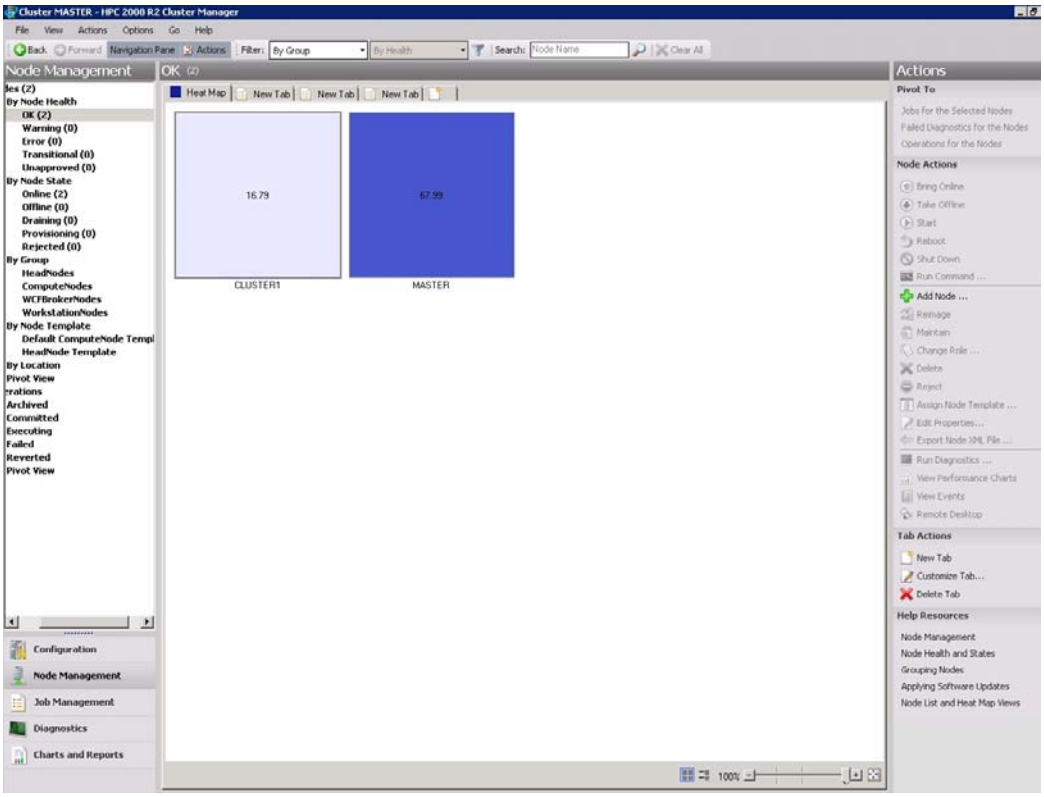

Fig. A13 – Nó computacional adicionado ao cluster