

**ESCOLA POLITÉCNICA
DA
UNIVERSIDADE DE SÃO PAULO**

**DEPARTAMENTO DE ENGENHARIA DE ENERGIA E
AUTOMAÇÃO ELÉTRICAS**



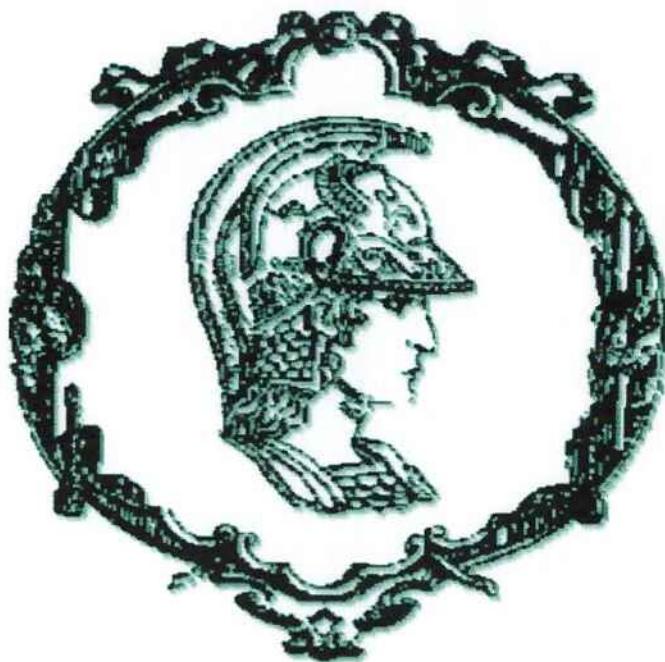
**Processamento Distribuído no Cálculo
Eficiente de Perdas Técnicas em Sistemas
de Distribuição de Energia Elétrica**

Edgar Sartori Lisboa

São Paulo
2006

**ESCOLA POLITÉCNICA
DA
UNIVERSIDADE DE SÃO PAULO**

**DEPARTAMENTO DE ENGENHARIA DE ENERGIA E
AUTOMAÇÃO ELÉTRICAS**



PROJETO DE FORMATURA

**Processamento Distribuído no Cálculo Eficiente
de Perdas Técnicas em Sistemas de Distribuição
de Energia Elétrica**

AUTOR: Edgar Sartori Lisboa

ORIENTADOR: Prof. Dr. Nelson Kagan

COORDENADOR: Prof. Dr. Luiz Cláudio Ribeiro Galvão

*À minha mãe,
minha família*

e

*Nelson Luiz,
por me indicar a direção certa.*

AGRADECIMENTOS

Ao Prof. Dr. Nelson Kagan, meu orientador, pelos ensinamentos, pela dedicação e pelas contribuições que foram fundamentais para o desenvolvimento deste trabalho;

Ao Prof. Dr. Luiz Cláudio Ribeiro Galvão pelo apoio e inesgotável paciência;

Aos amigos da Daimon-Interplan pela grande confiança e infinita compreensão;

Aos amigos André, Daniel e Otávio;

À minha revisora pelas palavras de incentivo;

A todos os colegas do curso que, direta ou indiretamente, contribuíram para o desenvolvimento deste trabalho.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

ABSTRACT

<u>1 - INTRODUÇÃO</u>	<u>10</u>
1.1 MOTIVAÇÃO	10
1.2 OBJETIVO	13
1.3 ORGANIZAÇÃO DO TRABALHO	14
<u>2 - REVISÃO DA LITERATURA</u>	<u>16</u>
2.1 ESTUDO DO FLUXO DE POTÊNCIA EM REDES ELÉTRICAS DE DISTRIBUIÇÃO	16
2.2 PERDAS TÉCNICAS EM REDES ELÉTRICAS DE DISTRIBUIÇÃO	17
2.2.1 PERDAS NO SISTEMA ELÉTRICO	17
2.2.2 CLASSIFICAÇÃO DAS PERDAS DE ENERGIA ELÉTRICA	17
2.2.3 CÁLCULO DAS PERDAS TÉCNICAS EM SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA	
ELÉTRICA	18
2.2.4 METODOLOGIA PARA O CÁLCULO DE PERDAS TÉCNICAS POR SEGMENTO	19
2.3 COMPUTAÇÃO DE ALTO DESEMPENHO	22
2.3.1 COMPUTAÇÃO DISTRIBUÍDA	23
2.3.2 COMPUTAÇÃO PONTO-A- PONTO (<i>P2P</i>)	30
2.3.3 PLATAFORMAS DE DESENVOLVIMENTO	36
2.4 A PLATAFORMA GPU	41
2.5 AVALIAÇÃO DO DESEMPENHO DE APLICAÇÕES DISTRIBUÍDAS	43

2.5.1 A LEI DE AMDAHL	44
2.5.2 PARALELIZAÇÃO	45
<u>3 - ESPECIFICAÇÃO DO APLICATIVO</u>	<u>46</u>
3.1 INTRODUÇÃO	46
3.1.1 ESCOLHA DA PLATAFORMA DE DESENVOLVIMENTO	46
3.1.2 OBJETIVOS DO APLICATIVO	47
3.1.3 ESCOPO DO APLICATIVO	47
3.2 REQUISITOS GERAIS	48
3.2.1 SOFTWARE E SISTEMA OPERACIONAL	48
3.2.2 HARDWARE	49
3.3 INTERFACE	49
3.4 ARQUITETURA	50
3.5 FUNCIONAMENTO	51
<u>4 - AVALIAÇÃO DO APLICATIVO</u>	<u>52</u>
4.1 METODOLOGIA	52
4.2 REDE ELÉTRICA EM ESTUDO	52
4.3 ESTRUTURA FÍSICA E TESTES REALIZADOS	52
4.4 RESULTADOS OBTIDOS	53
4.4.1 CASO-BASE	53
4.4.2 PRIMEIRO-TESTE	54
4.4.3 SEGUNDO-TESTE	55
<u>5 - CONCLUSÕES</u>	<u>58</u>
5.1 TÓPICO PARA FUTURO DESENVOLVIMENTO	58

<u>6 - ABREVIATURAS</u>	<u>60</u>
<u>7 - REFERÊNCIAS BIBLIOGRÁFICAS</u>	<u>61</u>
<u>8 - ANEXOS</u>	<u>64</u>
8.1 FIGURAS ADICIONAIS	65
8.2 TABELAS DOS TEMPOS DE CÁLCULO	67
8.2.1 CASO-BASE	67
8.2.2 PRIMEIRO –TESTE	69
8.2.3 SEGUNDO-TESTE	72

LISTA DE FIGURAS

<i>Figura 1 - Segmentos do sistema de distribuição</i>	20
<i>Figura 2 - Interface do Módulo GPU</i>	42
<i>Figura 3 - Módulo de geração de terrenos no GPU</i>	43
<i>Figura 4 - Ganho no desempenho em função do número de processadores</i>	44
<i>Figura 5- Interface do Módulo de Gerenciamento de Cálculo Distribuído</i>	49
<i>Figura 6 - Arquitetura e Componentes do Módulo de Computação Distribuída</i>	51
<i>Figura 7 - Módulo de gerenciamento executando três cálculos simultaneamente</i>	65
<i>Figura 8 - Módulo GPU exibindo os processos em execução simultânea</i>	65
<i>Figura 9 – Carga de trabalho de dois nós distintos, com flutuação devido à troca de processo</i>	66

LISTA DE TABELAS

<i>Tabela 1 - Resultados do Caso-base</i>	53
<i>Tabela 2 - Estatísticas do primeiro teste</i>	54
<i>Tabela 3 - Tempos estimados e reais para o primeiro-teste</i>	54
<i>Tabela 4 - Tempos de processamento segundo-teste</i>	55
<i>Tabela 5 - Latências de processamento segundo-teste</i>	56
<i>Tabela 6 - Tempos estimados e reais para o segundo-teste</i>	56

RESUMO

Para uma estimativa precisa das perdas técnicas em redes de distribuição elétricas, utiliza-se o procedimento de análise destas redes através do cálculo do fluxo de potência em diversos patamares diários das curvas de carga. Este procedimento torna a estimativa das perdas um processo extremamente moroso quando realizado em redes com dimensões reais. Uma alternativa para a redução deste tempo de análise é o uso de técnicas computacionais de alto desempenho.

Este trabalho avalia a aplicação da computação de alto desempenho, mais especificamente do modelo ponto-a-ponto, na eficientização do processo de cálculo.

Desenvolveu-se um sistema computacional, para uso em microcomputador, com o objetivo de atestar o modelo ponto-a-ponto no cálculo eficiente. Para o cálculo das perdas técnicas foi adotado um modelo completo de análise em oito segmentos aplicado a uma rede-modelo de dimensões reais. Ao final, o cálculo eficiente é comparado com o método tradicional de cálculo seqüencial.

ABSTRACT

The precise technical losses calculation on distribution electrical networks is achieved thru a power flow analysis of that network using an eight-segment model.

This calculation is so computing-intensive that takes much time to execute on real size electrical networks. The using of High Performance Computing (HPC) techniques must brighten up this problem.

This work evaluates High Performance Computing, more precisely the peer-to-peer computing model, on technical losses calculation speeding-up.

To evaluate the peer-to-peer model, a computational system was developed using the eight-segment model on losses calculation, applied to a real-size test electrical network. This work ends with a comparison between the traditional sequential method of calculation and the parallel method.

1 - INTRODUÇÃO

1.1 Motivação

O montante de energia elétrica consumido por consumidores livres e cativos no país registrou o valor de 29.639GWh em março de 2006, significando um aumento de 4,7% sobre o mesmo período do ano anterior [1]. As concessionárias devem atender este mercado crescente ao mesmo tempo em que garantem níveis mínimos de qualidade.

O alto custo da construção de novas unidades geradoras e da construção de novas linhas de transmissão e distribuição, a competitividade do setor e o alto valor das perdas impõem a necessidade do uso racional da energia produzida e da redução destas perdas.

Neste contexto, a análise das perdas técnicas é de extrema importância para um melhor aproveitamento do Sistema de Distribuição de Energia Elétrica, aumentando assim os níveis de qualidade e reduzindo os custos para atender a crescente demanda.

Para a estimativa das perdas técnicas utiliza-se o procedimento de análise das redes elétricas com o cálculo do fluxo de potência em diversos patamares diários das curvas de carga.

O cálculo do fluxo de potência é um procedimento básico, comumente realizado no estudo de redes elétricas. Está associado ainda a diversas análises como a análise do nível de carregamento dos equipamentos da rede elétrica, o planejamento da expansão do sistema elétrico, a previsão do carregamento dos equipamentos da rede após manobra de chaves quando em operação, entre outros.

Embora seja uma etapa de elevada importância na resolução de diversos problemas relacionados ao estudo de redes elétricas, o cálculo do fluxo de potência pode

ser um procedimento muito dispendioso computacionalmente. Por exemplo, um cálculo realizado em toda a rede de uma concessionária de distribuição, nos níveis de alta, média e baixa tensão em diversos patamares de carga pode consumir diversas horas de processamento, mesmo em computadores de grande porte. Por isso, quaisquer meios que venham reduzir o tempo de execução desta etapa de cálculo produzirão efeitos positivos ao longo de toda a análise da rede elétrica.

Para que se consiga uma diminuição no tempo de processamento pode-se atuar em duas frentes distintas:

Na primeira delas procura-se alterar o programa (*software*) em execução, tornando-o mais eficiente ou fazendo uso de novos algoritmos mais eficientes.

Na segunda, buscamos equipamentos (*hardware*) mais potentes para realizar as mesmas tarefas computacionais. Pode-se destacar ainda o uso de técnicas de computação de alto-desempenho como os processamentos paralelos, através do qual, vários processadores realizam parte de uma mesma atividade, ou da computação distribuída onde centros de processamento dividem a carga de trabalho, proporcionando a economia de tempo desejada.

Todas as técnicas citadas acima, naturalmente, possuem vantagens e desvantagens em relação umas às outras, e podem ser utilizadas separadamente ou em conjunto para melhorar o desempenho do sistema computacional utilizado na análise de redes elétricas.

As técnicas citadas primeiramente implicam na alteração e melhoria do algoritmo utilizado sendo que muitas vezes isto não é possível, pois em geral são algoritmos já bem estudados, de domínio amplo e, por isso, oferecendo poucas possibilidades de ganho de desempenho.

As outras técnicas, citadas posteriormente, necessitam obrigatoriamente um investimento adicional em equipamentos, quer seja na aquisição de máquinas mais poderosas ou com processadores adicionais, quer seja na aquisição de mais máquinas para associação destas.

No caso da utilização da computação distribuída ou do processamento paralelo, pode ainda haver necessidade de se desenvolver um trabalho adicional no programa para adequá-lo à arquitetura do novo sistema. No entanto, estas são as técnicas que melhor trazem resultado: se um problema é bem codificado, pelo menos em teoria, quanto maior o número de máquinas trabalhando para realizá-lo, menor será o tempo de execução, limitado apenas pela porção não paralelizável do código em execução, segundo o enunciado da lei de Amdahl[19][20]. Outras peculiaridades dos sistemas de processamento paralelo como a latência e o congestionamento dos canais de transmissão também influenciam o máximo ganho de desempenho, mas ainda assim, a abordagem baseada em hardware fornece uma perspectiva bem superior em comparação àquela baseada em software.

Sobre as técnicas de aumento do desempenho baseadas em *hardware* são feitas ainda algumas considerações:

- O uso de computadores mais poderosos é uma alternativa freqüentemente utilizada, porém esta se limita ao uso dos sistemas disponíveis comercialmente e, por isso, os ganhos tendem a ser limitados e marginalmente pequenos.
- A utilização de máquinas capazes de realizar processamento paralelo representa uma solução superior a anterior, pois é escalável, ou seja, podemos aumentar o número de processadores utilizados se precisarmos de um melhor desempenho. No entanto, estes sistemas são muito caros e

também oferecem limites rígidos para o número máximo de unidades de processamento.

- Os sistemas de computação distribuída apresentam diversas vantagens, aliando as qualidades presentes nos dois anteriormente citados. São altamente escaláveis, pois permitem que mais máquinas sejam acrescentadas ao sistema até que as necessidades de processamento sejam supridas; são mais baratos que sistemas multiprocessados. Podemos utilizar um número maior de máquinas mais lentas, porém mais baratas e amplamente disponíveis comercialmente. Dessa forma, consegue-se um custo menor para a mesma capacidade de processamento. Apesar das vantagens descritas, um sistema baseado em computação distribuída, apresenta as desvantagens do aumento da latência de processamento mediante ao aumento do número de máquinas executando um processamento e da dificuldade adicional na adaptação do *software* utilizado.

1.2 Objetivo

Pelo exposto anteriormente, o objetivo a ser atingido por este projeto é o desenvolvimento de uma aplicação ilustrando a análise das perdas técnicas de redes elétricas de distribuição através do uso da computação distribuída como ferramenta computacional. Este problema foi escolhido devido a sua já citada importância e às reais possibilidades de ganho no desempenho. A computação distribuída foi adotada como meio de prover este ganho pelas características já descritas, tratando-se de um melhor modelo para a resolução do problema.

A aplicação desenvolvida consiste em um módulo que executará o cálculo simultâneo das perdas técnicas de uma rede elétrica complexa. O aplicativo realiza o processamento num sistema composto por pelo menos dois computadores interligados apropriadamente que dividem a carga de processamento. O software apresenta uma interface com o usuário que fornece um mínimo de informações necessárias para a averiguação e comparação dos resultados obtidos. Os resultados almejados são os ganhos de desempenho no processamento de uma mesma rede elétrica comparando-se duas situações distintas: uma análise realizada num sistema computacional isolado e, posteriormente, a mesma análise realizada num sistema computacional distribuído.

1.3 Organização do Trabalho

Neste primeiro capítulo, apresenta-se o trabalho, suas motivações, objetivos almejados e a organização do presente documento.

No capítulo 2, encontra-se uma revisão da literatura. Encontramos aqui os principais conceitos utilizados no desenvolvimento do trabalho.

No Item 2.1 apresentamos a conceituação do estudo de fluxo de potência em redes elétricas para em seguida, discorrer sobre perdas técnicas em sistemas de distribuição no item 2.2. Neste item são citadas metodologias convencionais de cálculo para, em seguida, apresentarmos a metodologia adotada neste trabalho. Procura-se descrever, ainda que de forma sucinta, as características mais importantes que levaram a escolha da metodologia adotada.

No item 2.3 é feita uma rápida conceituação sobre a computação de alto desempenho e sua utilização na resolução de problemas que apresentam altas demandas computacionais. No subitem 2.3.1 são descritas as características, vantagens e desvantagens na escolha da computação distribuída como método de computação de

alto desempenho. Ainda no item 2.3 introduz-se a computação ponto-a-ponto (p2p), técnica utilizada no presente trabalho. Neste item são descritas as suas características, vantagens, desvantagens e finalmente é dada uma visão geral sobre as plataformas até então disponíveis para criação de aplicações p2p.

Nos itens seguintes, 2.4 e 2.5 respectivamente, é apresentada a plataforma de desenvolvimento escolhida, para posteriormente descrever um meio para prever o tempo de execução de aplicativos em sistemas distribuídos, a lei de Amdahl. É importante calcular o tempo teórico de processamento, pois de outra forma só seria possível obter o desempenho comparativo da aplicação, não tendo noção exata do quanto ela está otimizada.

O aplicativo é descrito e sua modelagem é comentada no item 3 -, para em seguida, no item 4 -, apresentar e comentar os testes realizados.

Finalmente, no item 5 - encontramos a conclusão do trabalho, que contempla também, um tópico com sugestões para futuro desenvolvimento.

Nos anexos ao final do documento, encontramos as tabelas contendo os dados coletados e sua análise, telas de exemplo da execução do aplicativo e seu código-fonte.

2 - REVISÃO DA LITERATURA

2.1 Estudo do Fluxo de Potência em Redes Elétricas de Distribuição

Por “estudo de fluxo de potência da rede elétrica” entende-se a resolução matemática do modelo do circuito elétrico que representa a rede, para o qual se dispõe da topologia, com as constantes elétricas dos seus elementos, das demandas das cargas e das tensões dos geradores que o excitam. [5]

Através do estudo de fluxo de potência, são realizadas as simulações de operação da rede com base em modelos. Este estudo é utilizado para obter na rede, as tensões em suas barras, o valor das correntes circulantes em seus trechos, a determinação dos desequilíbrios de corrente e tensão ou até mesmo calcular as perdas elétricas nos equipamentos da rede que, se comparadas com a demanda, fornecem critérios de eficiência operativa para a criação de estudos visando um melhor desempenho técnico e econômico. Portanto, o estudo do fluxo de potências é imprescindível para as análises que serão executadas posteriormente.

No estudo do fluxo de potência, após a modelagem das linhas, equipamentos e das cargas que compõem a rede em estudo, parte-se para a resolução das equações da rede modelada.

O processo de resolução matemática da rede é computacionalmente dispendioso e este custo computacional é crescente com o aumento do número de elementos de rede existentes.

Na resolução de redes em malha, são utilizadas matrizes cuja dimensão está associada ao número de elementos da rede representados. Para a resolução do problema, são geralmente utilizados o método de Gauss não simplificado ou o método de Newton-

Raphson [5]. O processo de resolução de redes radiais se aproveita da estrutura topológica da rede evitando a necessidade de cálculos matriciais sendo, portanto, um pouco mais simples. Em ambos os casos, como dito anteriormente, o uso de recursos computacionais é intensivo.

2.2 Perdas Técnicas em Redes Elétricas de Distribuição

2.2.1 Perdas no Sistema Elétrico

Segundo a definição da Empresa de Pesquisa Energética (EPE), as perdas constituem o montante da energia produzida que não é faturada pelos agentes vendedores. No Boletim de Mercado[1] de março de 2006 da própria EPE, com referência nos últimos 12 meses, o nível de perdas no Brasil, considerando o sistema interligado e os sistemas isolados, encontra-se em 17,2%. Deve-se destacar que na região nordeste o índice chega a 18,9% e nos sistemas isolados o índice atinge patamares de até 35,7%. Desse total, 65% das perdas de energia dizem respeito aos sistemas de distribuição.

2.2.2 Classificação das Perdas de Energia Elétrica

As perdas de energia podem ser classificadas segundo sua natureza (energia ou demanda), origem (técnica ou comercial), localização (global, transmissão ou distribuição) e componentes (condutores da rede primária, transformadores, condutores da rede secundária, ramais de ligação, medidores, equipamentos e diversos) [2].

A perda de demanda é a diferença entre a demanda de entrada e a demanda de saída num componente do sistema elétrico num dado instante. A perda de energia é a diferença entre a energia de entrada e a energia de saída num componente do sistema elétrico num dado intervalo de tempo.

A perda técnica é aquela que ocorre nos componentes da rede elétrica, por exemplo, as perdas no cobre ocorridas em transformadores de potência. A perda comercial é a energia distribuída, porém não faturada pelo agente distribuidor.

As perdas globais são as perdas nos sistemas de geração, transmissão e distribuição. As perdas na transmissão são as perdas nos sistemas de geração e de transmissão. As perdas na distribuição são as perdas que ocorrem somente no sistema de distribuição.

2.2.3 Cálculo das Perdas Técnicas em Sistemas de Distribuição de Energia Elétrica

Conforme apresentado por PENTEADO e HASHIMOTO [3], as perdas técnicas não podem ser medidas diretamente; logo, são avaliadas indiretamente através da caracterização matemática dos componentes da rede como, por exemplo, pela estimação das perdas no ferro e no cobre em transformadores de potência. Para tal, se faz necessário o estabelecimento de uma metodologia de cálculo.

O cálculo das perdas é, em geral, realizado em segmentos através de metodologias distintas que variam de autor para autor. De acordo com a extensa pesquisa realizada por MÉFFE [4], em linhas gerais, as diversas metodologias se assemelham quanto ao cálculo das perdas de energia. Este cálculo é realizado de forma indireta, ou seja, primeiro calcula-se a perda de demanda no horário de ponta do sistema para em seguida calcular a perda de energia utilizando o fator de perdas, que deve ser estimado.

Com a metodologia proposta por MÉFFE em [4], e adotada neste trabalho, o cálculo das perdas de energia é feito utilizando as curvas típicas de carga determinadas

para o sistema da concessionária, dispensando o cálculo das perdas técnicas de energia a partir de estimativa do fator de perdas.

2.2.4 Metodologia para o Cálculo de Perdas Técnicas por Segmento

A metodologia de MÉFFE adotada para o cálculo das perdas divide o sistema de distribuição em oito segmentos, a saber: sistema de alta tensão, subestação de distribuição, rede primária, transformador de distribuição, rede secundária, ramal de ligação, medidor de energia e outros (. Este último segmento agrupa as perdas referentes a equipamentos como reguladores de tensão, capacitores, correntes de fuga de pára-raios e perdas em conexões entre outros.

Nas diversas metodologias analisadas por MÉFFE, as perdas de energia nos principais segmentos do sistema, incluindo linhas e transformadores, são calculadas indiretamente a partir do cálculo das perdas máximas de demanda e de estimativas do fator de perdas. Em contrapartida, MÉFFE propõe a utilização de metodologia baseada no uso de curvas de carga típicas por classe de consumidor e por tipo de atividade desenvolvida.

Para aplicar essa metodologia, é necessário que a concessionária tenha uma base de dados onde estão cadastrados os dados de todas suas redes primárias, redes secundárias, transformadores de distribuição, subestações de distribuição, redes de alta tensão e também os dados de todos os seus consumidores.

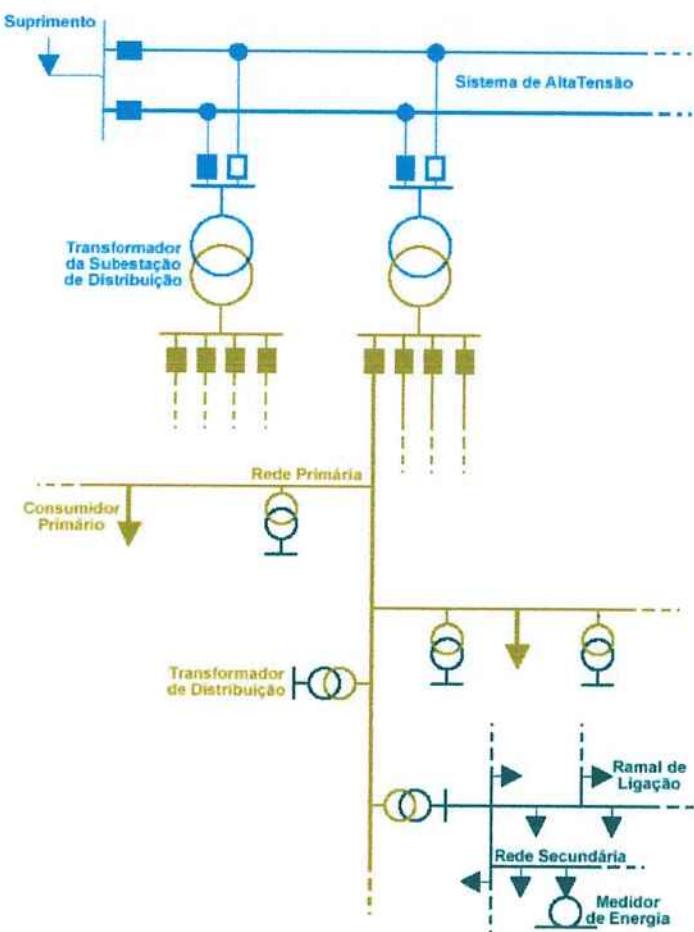


Figura 1 - Segmentos do sistema de distribuição

2.2.4.1 Procedimento Geral de Cálculo

Na metodologia de MÉFFE, o cálculo das perdas é feito no sentido a montante, partindo-se dos consumidores secundários. A seguinte sequência é dotada:

- Inicialmente, para cada medidor de energia é atribuído um valor fixo de perdas;
- Para o cálculo das perdas nos ramais de ligação são assumidos valores típicos de comprimento e resistência ôhmica de condutores padronizados e utilizadas curvas de carga típicas dos consumidores;

- c) O valor das perdas da rede secundária ligada a um transformador de distribuição (ET – estação transformadora) é obtido através do cálculo elétrico da rede, por fase, utilizando a curva de carga;
- d) Em seguida, o valor das perdas no ferro e no cobre é calculado na ET com base nos valores nominais e nas curvas de carga diárias por fase. Estas curvas são obtidas pela agregação das curvas de carga de todos os seus consumidores;
- e) A perda num alimentador primário é calculada, por fase, utilizando o valor concentrado em cada ET e adicionando-se os consumidores primários, avaliados segundo as suas curvas de carga diárias. Desta forma obtém-se a curva de carga diária do alimentador;
- f) Após o cálculo de todos os alimentadores é possível agora contabilizar as perdas nos transformadores da subestação (SE), agregando as perdas no ferro e no cobre. Pode-se então obter a curva de carga diária da subestação por fase;
- g) Ao término do cálculo de todas as SEs obtêm-se as perdas do sistema de alta-tensão através do cálculo elétrico por fase, através das curvas de carga obtidas;
- h) A diferença entre a energia medida nos pontos de suprimento e a soma da energia faturada com as parcelas de perdas calculadas nos passos anteriores é igual à soma das perdas não técnicas e das perdas técnicas classificadas no segmento “outros”.

2.2.4.2 Considerações Gerais

Na metodologia proposta as perdas são calculadas de forma detalhada, através de cálculo elétrico específico para cada componente de um determinado segmento do sistema, eliminando as simplificações existentes nos modelos encontrados na bibliografia pesquisada por MÉFFE.

A principal vantagem da metodologia é que esta não depende das simplificações usualmente adotadas como, por exemplo, a estimativa do fator de perdas a partir do fator de carga, que pode influir de forma significativa nos resultados do cálculo. O cálculo das perdas de energia é efetuado diretamente a partir da curva de carga.

Outro ponto favorável à metodologia apresentada é a representação por fase de cada trecho da rede e a representação por fase dos transformadores. Os desequilíbrios de correntes são considerados ao realizar o cálculo por fase e, portanto, não há a necessidade de estimar um fator de desequilíbrio, tornando os resultados dos cálculos mais precisos.

O inconveniente trazido por esta metodologia é a demanda computacional gerada devido à necessidade da execução do cálculo do fluxo de potências para cada um dos 96 patamares diários de carga para cada um dos segmentos em análise. Dessa forma, a análise de redes de dimensões reais pode ser extremamente lenta.

2.3 Computação de Alto Desempenho

A computação de alto desempenho ou HPC, do inglês *High Performance Computing*, é um ramo da ciência da computação que se concentra no desenvolvimento de supercomputadores e do software que são operados neste. A computação de alto desempenho faz uso de uma classe de sistemas computacionais projetados para

processamento intensivo de grandes volumes de dados. A grande capacidade de processamento desses sistemas é obtida basicamente por dois meios: No primeiro é feito uso de processadores mais potente, no segundo é feito uso de muitos processadores trabalhando na resolução de um mesmo problema. Em geral, quando o uso de processadores mais potentes se torna tecnologicamente ou financeiramente inviável, resta como alternativa a segunda. O uso de múltiplos núcleos de processamento pode ser feito numa mesma máquina, chamado de multiprocessamento ou processamento paralelo, ou ainda em máquinas distintas, chamado de processamento distribuído. Com a evolução das tecnologias de comunicação e pela flexibilidade da arquitetura, a computação distribuída tem sido a escolhida para a resolução de problemas que apresentam altas demandas computacionais, como é o caso da execução do cálculo do fluxo de potências para cada um dos 96 patamares diários de carga.

2.3.1 Computação Distribuída

A computação distribuída faz uso dos sistemas distribuídos na resolução de tarefas computacionais. É um processo de agregação do poder computacional. Segundo definição do IEEE[7], um sistema distribuído trata-se de um sistema computacional no qual diversos computadores interconectados compartilham as tarefas computacionais atribuídas ao sistema como um todo.

De acordo com ATTIYA[8], esta definição é generalista e encobre toda uma gama de sistemas computacionais mais modernos que vão desde sistemas *VLSI* integrados em um único componente ou até mesmo a *Internet*.

Nos dias atuais, os sistemas distribuídos não estão restritos aos ambientes de pesquisa, mas já fazem parte de soluções utilizadas por empresas e até mesmo pelo

governo. Tipicamente, estes sistemas fornecem a possibilidade de compartilhamento de recursos, serviços e informações.

A computação distribuída é conhecida pela sua grande quantidade de modelos, que não se traduzem exatamente em arquiteturas reais. Por isso, ATTIYA[8] e outros autores organizam o tema de acordo com a taxonomia de FLYNN[9][10] na qual os modelos são classificados de acordo com o número concorrente de instruções e de fluxos de dados disponíveis na arquitetura. Mais detalhes sobre a diferenciação dos modelos pode ser encontrado nas referências indicadas.

2.3.1.1 Caracterização de Sistemas Distribuídos

A caracterização dos sistemas distribuídos pode ser realizada com base nos modelos, como citado anteriormente. Porém, é comum classificar os sistemas de computação distribuída pelo modo os quais seus componentes estão conectados (topologia física do sistema) ou ainda pelo fluxo de informação no sistema (topologia lógica).

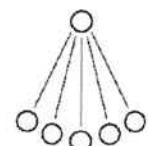
As principais topologias são descritas a seguir:

Topologias Básicas

- **Centralizada**

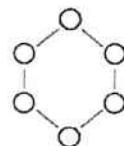
Esta é a topologia mais comumente encontrada, está presente nas aplicações cliente-servidor como em servidores da *Internet*.

Todas as funcionalidades e informações estão concentradas num único servidor com muitos clientes se conectando a ele para enviar e receber informações.



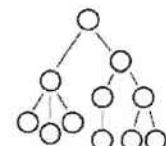
- **Anel**

Quando apenas um servidor não é capaz de atender uma demanda muito grande, utiliza-se um conjunto de máquinas dispostas em anel e trabalhando como um servidor distribuído. As máquinas do anel se comunicam com o objetivo de promover um compartilhamento comum do estado de funcionamento.



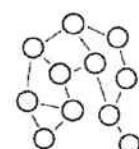
- **Hierárquica**

A informação é enviada por um nó central e percorre todos os outros nós da rede até o seu destino, através de uma estrutura em forma de árvore.



- **Descentralizada**

Nesta topologia, todos os nós estão interconectados e desempenham o mesmo papel na rede. É a topologia adotada na computação ponto-a-ponto e também é a escolhida para o desenvolvimento do projeto. As razões serão detalhadas adiante.

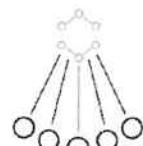


Topologias Híbridas

Topologias híbridas, compostas por uma combinação de duas ou mais daquelas descritas acima, podem ainda ser utilizadas:

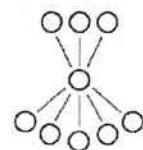
- **Centralizada +Anel**

Trata-se de uma rede centralizada onde o papel do servidor é realizado por um anel de servidores.



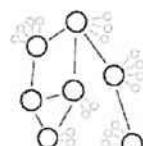
- **Centralizada + Centralizada**

Nesta topologia um servidor também é cliente de um ou mais outros servidores.



- **Centralizada + Descentralizada**

Nesta organização, redes centralizadas operam como nós da rede descentralizada.



2.3.1.2 Aplicações

Existem muitos tipos diferentes de sistemas baseados em computação distribuída. O objetivo principal de um sistema desse tipo é interligar usuários e recursos de uma forma transparente, aberta e *escalável*. Idealmente, este arranjo é incomparavelmente mais tolerante a falhas e mais poderoso que outros baseados em servidores isolados.

Diversos fabricantes desenvolveram e comercializaram sistemas de computação distribuída durante anos, criando numerosas iniciativas e arquiteturas que permitem o processamento de dados através de sistemas conectados em rede.

No mundo corporativo, computação distribuída geralmente significa dividir diversos passos de um processo, para que estes sejam realizados mais eficientemente. Tipicamente, é utilizada a arquitetura cliente-servidor. Empresas de computação gráfica utilizam a computação distribuída para realizar a *renderização* de imagens em animações em 3D.

No mundo acadêmico e científico, a computação distribuída é utilizada para reduzir o tempo de solução de problemas matemáticos complexos. Estes são divididos

em parcelas menores, resolvidas por muitos computadores e em seguidas combinadas novamente para fornecer a solução do problema original. Cientistas a utilizam, entre outras coisas, na pesquisa e criação de novas drogas, realização de previsões climáticas e simulações do mundo físico.

No âmbito científico a otimização é tão profunda que vai até ao código-fonte. Freqüentemente, programas são modificados para que trechos deste, executando tarefas semelhantes, sejam processados em paralelo. No âmbito corporativo esta prática não é tão comum, pois a modificação do código-fonte de uma aplicação demanda alto conhecimento técnico e muito tempo de trabalho sendo, muitas vezes, inviável economicamente. Nestes casos, normalmente as aplicações é que são distribuídas e, desta forma, a distribuição das tarefas é limitada à distribuição dos diferentes processos existentes.

O mais famoso e um dos mais engenhosos projetos a utilizar a computação distribuída é o *SETI@Home*[23][24]. Criado em 1984, o instituto SETI da Universidade da Califórnia, em Berkeley, conduz pesquisas sobre o espaço e possíveis formas de vida. No início do projeto, a maior parte das pesquisas era feita utilizando-se telescópios convencionais. O programa *SETI@Home* foi criado onze anos depois, quando os pesquisadores se deram conta que para a magnitude do universo pesquisado seria necessário muito poder computacional. O conceito por detrás do *SETI@Home* é obter auxílio dos usuários da Internet no processamento, de forma barata e rápida, dos dados obtidos pelos radiotelescópios. Então, uma equipe formada por David Gedye e Craig Kasnoff apresentou uma proposta do projeto que foi largamente apoiada no 5º Congresso Internacional de Bioastronomia[25]. O projeto foi lançado oficialmente em maio de 1999 e hoje conta com aproximadamente dois milhões de unidades de

processamento executando o aplicativo. Os radiotelescópios do programa, localizados em Arrecibo, Porto Rico, obtêm diariamente cerca de 40 gigabytes de dados.

2.3.1.3 Vantagens

Existem quatro vantagens básicas de se projetar uma aplicação para um sistema distribuído, a saber:

- Diminuição do tempo de execução da aplicação;
- Aumento do grau de confiabilidade e disponibilidade da aplicação;
- O uso de partes do sistema para fornecer especialização funcional;
- Escalabilidade.

A diminuição do tempo de execução da aplicação pode ser obtida através do uso do paralelismo existente em um sistema distribuído. Alguns programas terão seus tempos de execução diminuídos se partes desses programas forem executadas em processadores diferentes ao mesmo tempo.

Sistemas distribuídos são potencialmente mais confiáveis, pois desde que os processadores são autônomos, uma falha em uma unidade não afeta o funcionamento correto das demais. Portanto, a confiabilidade do sistema pode ser aumentada ao se replicar funções e/ou dados da aplicação nos vários processadores. Desse modo, se alguns elementos processadores falharem, os demais poderão continuar o serviço. Exemplos clássicos de aplicações tolerantes falhas é o controle de aeronaves e de fábricas automatizadas.

Qualquer aplicação implementada como um conjunto de serviços especializados amolda-se no ambiente de um sistema distribuído. Cada um desses serviços pode usar um ou mais servidores dedicados, garantindo bom desempenho e alta confiabilidade. Os

servidores podem trocar requisições entre si através da rede. É fácil adicionar novos servidores para atender a novos serviços ou para aumentar a capacidade computacional de serviços já existentes. Recursos especiais do sistema são passíveis de serem compartilhados, pois os servidores se comunicam através da rede.

Uma outra vantagem do uso de sistemas distribuídos é a escalabilidade¹. A escalabilidade é a capacidade que um sistema ou um processo tem de suportar uma carga de trabalho crescente ou de ser prontamente expandido. Assim, em sistemas distribuídos, o poder de processamento total disponível depende intimamente do número de computadores colaborando com o esforço de cálculo.

2.3.1.4 Inconvenientes e Desvantagens

As principais desvantagens associadas ao uso da computação distribuída são inerentes a sua filosofia de funcionamento e sua organização.

O uso de redes de comunicação introduz uma camada adicional responsável por uma maior latência na execução das tarefas, se comparadas às tarefas executadas em uma única máquina. Portanto, num caso extremo, tarefas que necessitam de um grande volume de dados para execução podem tornar-se até mais vagarosas quando executadas num ambiente distribuído. Os programas devem ser cuidadosamente estruturados e projetados para seja possível usufruir das vantagens da distribuição das tarefas.

Além disso, dependendo da arquitetura utilizada, um computador central ainda é necessário para coordenar a divisão do trabalho.

¹ Uma definição possível para escalabilidade encontra-se no tópico 2.3.2.1.2

2.3.2 Computação ponto-a-ponto (p2p)

O termo ponto-a-ponto, do inglês “*peer-to-peer*”, refere-se a uma classe de sistemas e aplicações que empregam recursos distribuídos na execução de tarefas críticas de maneira descentralizada.

A computação ponto-a-ponto (p2p) é ainda uma categoria de computação distribuída em desenvolvimento e pode ser utilizada para muitas finalidades diferentes. Programas de compartilhamento de arquivos tais como o *Napster* e o *Gnutella* são os dois exemplos mais conhecidos do p2p. A idéia por detrás do p2p é que todos os nós conectados em uma rede computacional, mais geralmente a *Internet*, têm acesso a todos os nós restantes nessa rede. A tecnologia p2p lida com a transferência de informação de uma forma que pode ser específica de acordo com a necessidade dos usuários, utilizando conexões diretas para acelerar os tempos de transferência de informação.

Apesar disso, a tecnologia p2p não possui verdadeiramente uma definição concisa e simples. Diversos autores tentam caracterizar aplicações p2p baseadas no seu grau de centralização, mas esta é uma simplificação drástica e que fere as bases da computação p2p. De forma ampla, p2p é o paradigma computacional que torna as redes simétricas. Neste paradigma podem existir serviços centralizados, mas o cliente (ou nó) é o foco principal da implementação ou aplicação.

Em seu trabalho, SCHOLLMEIER[11] busca uma diferenciação entre a computação p2p e o modelo cliente-servidor, bem próximos e freqüentemente confundidos. Ele conclui que a diferença se encontra no papel dos nós na rede computacional.

Os serviços disponíveis numa rede computacional necessitam, em geral, de um nó responsável pela sua manutenção, e outros nós na rede utilizam esse recurso através do nó responsável. Num paradigma cliente-servidor esses papéis são fixos ou pelo

menos se alteram muito raramente. Esta é a principal diferença para o p2p, que não possui um cliente ou um servidor de recursos previamente definido e qualquer nó da rede pode ora prover, ora utilizar recursos computacionais. Muitas vezes isso ocorre de forma simultânea.

Os sistemas p2p podem ainda ser subclassificados utilizando critérios como a centralização dos serviços (pureza do sistema), dando origem a uma taxonomia própria. Mais detalhes está disponível nas referências citadas.

2.3.2.1 Características

Esta seção descreve sucintamente os tópicos que mais influenciam as aplicações e sistemas p2p.

2.3.2.1.1 Descentralização

O modelo p2p questiona a necessidade do armazenamento e processamento apenas em um ponto centralizado, ao contrário do que ocorre no modelo cliente-servidor. Uma das idéias mais fortes na descentralização é a ênfase dada ao usuário no controle dos dados e recursos.

Em sistemas totalmente descentralizados, todos os nós são participantes igualitários da rede computacional. Na prática, isso torna complicada a implementação de modelos p2p, pois não há um servidor com uma visão geral de todos os nós da rede. Esta é a razão pela qual muitos sistemas p2p são construídos com base numa abordagem híbrida, utilizando a topologia centralizada.

Em sistemas totalmente descentralizados, a simples ação de verificar a existência e localizar a rede computacional já é uma tarefa muito difícil. No caso do *Gnutella*, sistema escolhido para o desenvolvimento do presente trabalho, novos nós precisam

conhecer o endereço de outro nó ou usar uma lista de endereços conhecidos para que possam se incorporar à rede computacional.

A classificação de sistemas p2p baseados em seu grau de centralização ou autonomia, dá origem a categorias de sistemas p2p puros e sistemas p2p híbridos. Esta categorização reflete diretamente a auto-organização e escalabilidade do sistema.

2.3.2.1.2 Escalabilidade

Um benefício imediato à descentralização é o aumento da escalabilidade. A escalabilidade é limitada por fatores como a quantidade de operações centralizadas que serão realizadas, o grau de paralelismo presente na aplicação e o modelo de uso no desenvolvimento desta. A escalabilidade depende também da demanda de comunicação entre nós necessária para realização de uma tarefa; quanto menor esta demanda, mais escalável será a aplicação ou o sistema.

2.3.2.1.3 Auto-organização

A auto-organização é definida como “um processo onde a organização de um sistema aumenta espontaneamente, isto é, sem que este aumento seja controlado por fatores ambientais ou qualquer outro sistema externo”. [12]

Nos sistemas p2p a auto-organização é necessária para dar suporte à escalabilidade e a tolerância à falhas num ambiente onde a conexão com os nós é intermitente. Desta forma, a responsabilidade do gerenciamento das mudanças ocorridas no sistema é distribuído por todos os componentes da rede.

2.3.2.1.4 Desempenho

O desempenho é uma preocupação significante nos sistemas p2p. Os sistemas p2p buscam o aumento de desempenho através da agregação da capacidade de

processamento dispersa pelos nós do sistema. Devido à natureza descentralizada desses modelos, o desempenho é afetado principalmente por três tipos de recursos: armazenamento, processamento e comunicação. Em particular, atrasos na comunicação podem ser significativos em redes abrangendo grandes áreas geográficas. Por outro lado, a comunicação limitada pela largura da banda disponível, torna-se a preocupação mais significativa quando se trata de um sistema com um grande volume de mensagens propagadas.

Em sistemas como o *Gnutella*, não há um responsável pela sua coordenação. Ao invés disto, a comunicação é coordenada por cada um dos nós individualmente. Tipicamente, é utilizado um mecanismo de encaminhamento de mensagens na busca de informação e dados. O problema com este tipo de sistema é que se acaba tendo um volume muito grande de mensagens propagadas de um nó a outro. Cada encaminhamento de mensagem contribui para o crescimento da demanda pela banda de comunicação. Nesses sistemas, a largura de banda necessária é proporcional ao número de mensagens trocadas, que, por sua vez, é proporcional ao número de nós conectados.

Há três mecanismos básicos para otimização do desempenho: replicação, *caching* e roteamento inteligente.

A primeira técnica faz a duplicação das informações mais utilizadas em diversos nós da rede, diminuindo assim a distância média para se alcançar determinado recurso. No *caching*, a duplicação de recursos é temporária, e só é feita mediante a utilização deste, ou seja, somente os recursos recém utilizados permanecem duplicados. Já o roteamento inteligente cuida para que apenas os melhores caminhos entre dois nós sejam utilizados durante o uso de recursos e troca de mensagens.

2.3.2.1.5 Segurança

Os sistemas p2p utilizam a maioria das técnicas de segurança utilizadas nos sistemas distribuídos: cadeias confiáveis entre nós e recursos compartilhados, trocas de chaves de sessão, encriptação e assinaturas. Mais detalhes nas referências [13][14][15][16].

2.3.2.1.6 Transparência

Em sistemas distribuídos, a transparência foi associada com a capacidade de conectar estes sistemas a sistemas locais de forma não perceptível. Segundo COULOURIS[17], a principal forma de transparência é a de localização, mas outras incluem transparência no acesso, na replicação, falhas, mobilidade, escalonamento, etc.

2.3.2.1.7 Tolerância a Falhas

Um dos principais objetivos no projeto de sistemas p2p é evitar um ponto central de falhas. Embora isto esteja contemplado intrinsecamente no modelo p2p, outros tipos de falhas são ocasionadas justamente pelo fato dos recursos estarem distribuídos pelo sistema. A indisponibilidade de conexão de um nó pode acarretar na indisponibilidade de determinado recurso. Isto pode acontecer tanto porque o recurso está indisponível devido a uma falha no sistema de comunicação quanto porque o nó responsável pelo recurso teve problemas.

Um aspecto interessante do modelo p2p é que a responsabilidade de manutenção do sistema está distribuída por todos os nós, garantindo assim a sua disponibilidade. Isto é bem diferente do modelo cliente-servidor, onde a responsabilidade de manutenção do sistema é toda do nó servidor. Assim, nos sistemas p2p, mesmo que um nó esteja indisponível, o sistema se mantém ativo.

2.3.2.1.8 Interoperabilidade

Ainda não há interoperabilidade entre os diversos sistemas p2p disponíveis. Isto se deve a falta de um padrão estabelecido, embora esforços como o JUXTA[18] sejam feitos nesse sentido.

2.3.2.2 Vantagens

A grande vantagem já citada dos sistemas p2p é a distribuição da responsabilidade de fornecer os serviços por todos os nós da rede. Este fato elimina a indisponibilidade do serviço no caso de um único ponto de falha na rede computacional. Além do mais, redes p2p exploram a disponibilidade da largura de banda usando vários canais de comunicação.

A tecnologia p2p tem a capacidade de disponibilizar recursos com maior robustez, a um custo reduzido e maximizando os recursos de cada dispositivo ligado à rede computacional. Companhias como a Intel já utilizam a tecnologia p2p para reduzir o custo de distribuir documentos e arquivos por toda a empresa.

2.3.2.3 Desvantagens

Infelizmente a tecnologia p2p tem algumas desvantagens dadas pela natureza redundante da rede p2p. A atribuição dos canais de comunicação em redes p2p é não-determinística por natureza. Por exemplo, clientes ao efetuar exatamente a mesma requisição podem conectar-se a máquinas completamente diferentes, por vias de comunicação diferentes e com resultados diferentes. Pedidos enviados a uma rede p2p podem não resultar numa resposta imediata ou mesmo nunca devolver resposta alguma.

Além disso, o sistema não é estático e os recursos da rede p2p podem desaparecer da mesma forma que os nós que os possuem se conectam ou desconectam da rede.

No entanto, a tecnologia p2p pode ultrapassar todas estas limitações. Embora os recursos possam desaparecer às vezes devido à indisponibilidade do nó que o fornece, uma aplicação p2p pode utilizar um mecanismo de replicação dos recursos mais utilizados em múltiplos computadores. Quanto maior o número de recursos interligados, menor a possibilidade de uma requisição por determinado recurso não ser atendida. Ou seja, a própria estrutura da rede p2p que causa o problema pode ser utilizada para resolvê-lo.

2.3.3 Plataformas de Desenvolvimento

O envio e recebimento de mensagens é ponto fundamental para o funcionamento de aplicações distribuídas. A passagem de mensagens de um membro a outro do sistema distribuído tem como principal finalidade a sincronização dos processos executados paralelamente. Sem essa sincronização ocorrem erros como o disparo de uma etapa de processamento antes que o resultado da etapa anterior seja recebido.

É necessário um aplicativo para tornar usável toda a estrutura física de uma rede de computadores. Antigamente utilizavam-se sistemas específicos para viabilizar esta comunicação, porém hoje em dia estes componentes já vêm integrados na maioria dos sistemas operacionais.

Apesar disso, os componentes necessários para a troca de mensagens de processamento e gerenciamento dos processos em sistemas distribuídos ainda não se encontram integrados aos sistemas operacionais e por isso, precisamos utilizar componentes adicionais de software. Estes componentes adicionais geralmente se apresentam na forma de plataformas de desenvolvimento (*framework*).

Plataforma de desenvolvimento é uma infra-estrutura criada para a construção, distribuição e execução de aplicações e serviços que utilizam uma determinada tecnologia.

Os *frameworks* oferecem facilidades para o desenvolvimento de novas aplicações, assim como as ferramentas necessárias para a customização e integração de aplicações pré-desenvolvidas. Não se tratam de sistemas p2p, mas sim do suporte aos componentes básicos para desenvolvê-los, por exemplo, ferramentas para busca e identificação de nós, comunicação, segurança e outros recursos agregados, para o desenvolvimento de sistemas p2p.

A seguir encontramos uma descrição das principais plataformas pesquisadas para o desenvolvimento da aplicação p2p deste projeto.

2.3.3.1 Groove

Lançado em 2001 por um desenvolvedor da *Lotus Notes*[21], o Groove[27] é um sistema p2p colaborativo; entretanto, pode também ser considerado uma plataforma.

Groove foi concebido para proporcionar comunicação, compartilhamento de conteúdo e ferramentas para atividades em grupo. Por exemplo, todas as ferramentas herdam automaticamente o sistema segurança sem que seja necessário ao desenvolvedor da aplicação se preocupar com isso.

O objetivo principal da plataforma é permitir que os usuários se comuniquem sem que seja necessário o uso de um servidor. Outros objetivos incluíam a segurança, um ambiente assíncrono que permitisse desconexões, livre de administração e flexível.

O Groove foi desenvolvido para comunicação, colaboração e compartilhamento de conteúdo.

2.3.3.2 Magi

Magi[28] evoluiu de um projeto de pesquisa liderado por Greg Bolcer da Universidade da Califórnia. A primeira versão da plataforma foi comercialmente lançada em 2000.

Trata-se de um *framework* para a construção de aplicações colaborativas, multiplataforma e seguras, utilizando tecnologias da Internet como HTTP, WebDAV e XML.

O objetivo do Magi é proporcionar o compartilhamento de informações a qualquer dispositivo que utilize padrões baseados na Internet, procurando facilitar o desenvolvimento de aplicações distribuídas baseadas em Java que utilizam XML.

2.3.3.3 FreeNet

FreeNet[29][30] é um sistema de compartilhamento de arquivos baseado num projeto original de Ian Clarke. O principal objetivo da FreeNet era tornar o uso do sistema anônimo. Isto é, após entrar no sistema, um usuário deve ser capaz de realizar requisições a arquivos sem que alguém seja capaz de identificar quem originou estas requisições.

Da mesma forma, se um usuário guarda um arquivo no sistema, deve ser impossível de identificar a origem de tal arquivo. Finalmente, o operador de um nó FreeNet deve ser incapaz de conhecer quais arquivos estão armazenados em seu computador. Isto é conseguido através de um sistema de encriptação dos dados armazenados.

O grau de anonimato presente neste sistema dificulta a análise de seu comportamento e descoberta de erros. Por esta razão, apenas o resultados de simulações

estão disponíveis, não foram feitas tentativas de se avaliar o sistema operando realmente.

2.3.3.4 JXTA

JUTXA[18] é um projeto que visa desenvolver uma plataforma aberta, colaborativa e que suporte uma grande variedade de aplicações de computação distribuída, possibilitando a execução destas em qualquer dispositivo conectado em rede.

O projeto JUTXA foi divulgado pela Sun Microsystems em abril de 2001 e logo se tornou popular. Seu objetivo principal é prover um *framework* de uso geral para programação da rede e computação. Outros objetivos são: a interoperabilidade, permitindo a fácil identificação de nós interconectados e a independência de plataformas, ou seja, independência da linguagem de programação, do sistema operacional e da plataforma de rede utilizada.

2.3.3.5 Windows Peer-to-Peer Networking

O Windows Peer-to-Peer Networking é um componente do sistema operacional que permite a criação de novas aplicações p2p para computadores executando sistema operacional Windows.

Esta plataforma era recém-lançada no momento de realização deste trabalho. Por isso, pouca informação estava disponível. Ainda assim, ela apresenta-se muito promissora, principalmente pela possibilidade de profunda integração com o sistema operacional e ferramentas comerciais de desenvolvimento. [31]

2.3.3.6 Gnutella

Gnutella[22] é um protocolo de compartilhamento de arquivos. As aplicações que implementam o Gnutella permitem a seus usuários a busca e obtenção de arquivos de outros usuários conectados à *Internet*. A tecnologia de compartilhamento de arquivos Gnutella foi introduzida em março de 2000 por dois empregados da AOL como um programa de código aberto e com funcionalidades semelhantes as do *Napster*.

O objetivo do Gnutella é fornecer uma solução puramente distribuída onde os usuários podem executar programas que implementam o protocolo Gnutella para buscar e obter arquivos compartilhados.

Gnutella não se trata de um sistema, mas sim de um protocolo de comunicação. Para ingressar na rede, um nó deve primeiro conhecer o endereço de outro nó. Quando um usuário faz uma requisição para um serviço, ele submete o pedido a todos os nós conhecidos por ele. Estes nós, podendo atender ou não ao pedido inicial, repassam a requisição adiante para os nós que eles conhecem (*flooding*). A mensagem é encaminhada até que seja atingido seu tempo máximo de vida (TTL), do inglês, *Time-To-Live*.

Embora o modelo Gnutella seja bem-sucedido, em teoria, sua escalabilidade não é boa. O número de mensagens e de respostas potenciais aumenta exponencialmente a cada requisição. Por exemplo, se cada nó estiver conectado a apenas outros dois e o TTL de uma requisição for igual a sete (o número padrão), então o número de requisições será igual a 128, e o número de respostas será substancialmente superior.

O protocolo Gnutella não provê um mecanismo de tolerância a falhas. Espera-se que nós suficientes estejam conectados na rede de forma que as requisições serão propagadas tão longe que sempre encontrarão uma resposta. Entretanto, a natureza

distribuída do protocolo não garante este comportamento. Na verdade, estudos [32] demonstram que apenas uma pequena fração de usuários permanece conectada tempo suficiente para responder a requisições de outros usuários.

Gnutella desenvolveu muitas técnicas inovadoras de busca e armazenamento em p2p e apesar das deficiências, foi amplamente adotado pelos usuários e é um dos sistemas mais utilizados. Hoje em dia, Gnutella é muito mais um paradigma que uma aplicação. Algumas empresas fazem uso do protocolo Gnutella em seus aplicativos comerciais como aplicativos para gerenciamento de projetos e para ensino a distância.

2.4 A Plataforma GPU

A GPU (Global Processing Unit)[33][34] é uma plataforma de desenvolvimento para aplicações que pretendem utilizar a computação ponto-a-ponto. Ela foi criada aproveitando o protocolo de compartilhamento de arquivos Gnutella, descrito anteriormente. Esta foi a plataforma escolhida como base para o desenvolvimento do presente trabalho. Detalhes sobre o protocolo Gnutella são encontrados em [22].

A plataforma de desenvolvimento foi codificada em *Borland Delphi*, uma linguagem derivada do Pascal e disponível para sistemas *Windows* e *Linux*. A GPU fornece as funções básicas de comunicação para que seja possível incorporar, no software que faz uso desta plataforma, recursos para compartilhamento da capacidade de processamento, ferramentas de distribuição de tarefas, monitoração da fila de processos, entre outras.

GPU se apresenta na forma de um aplicativo cliente Gnutella que permite aos usuários um compartilhamento temporal da capacidade de processamento de seus computadores. O compartilhamento não reconhece privilégios entre os usuários. Cada

nó participante concorda em ceder recursos de processamento na medida do necessário, e em contrapartida, pode utilizar o poder de processamento de outros nós da rede.

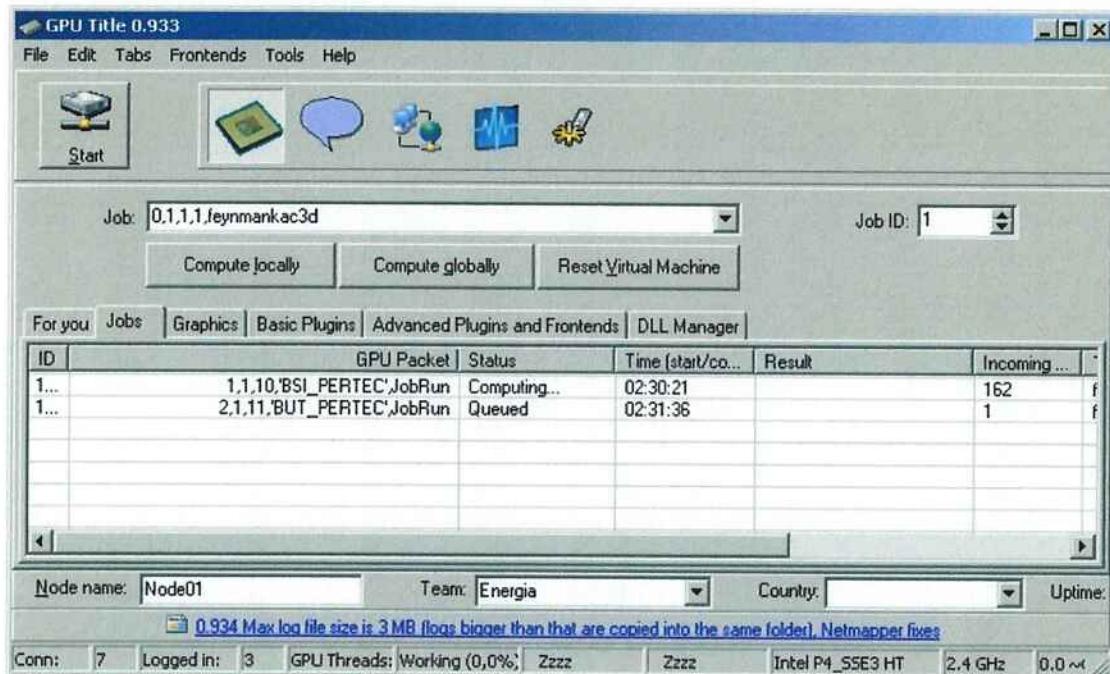


Figura 2 - Interface do Módulo GPU

Os serviços oferecidos pelos nós são estendidos por meio de *Plug-Ins*, ou seja, módulos independentes que são agregados à plataforma básica buscando o oferecimento de um novo serviço.

Como foi baseado no protocolo Gnutella, é possível utilizar a GPU tanto em computadores numa rede local quanto em computadores fisicamente dispersos, porém conectados pela Internet.

O uso da GPU oferece a desvantagem de tratar-se ainda de uma plataforma em desenvolvimento e, por isso, há a possibilidade do surgimento de dificuldades não previstas em sua utilização.

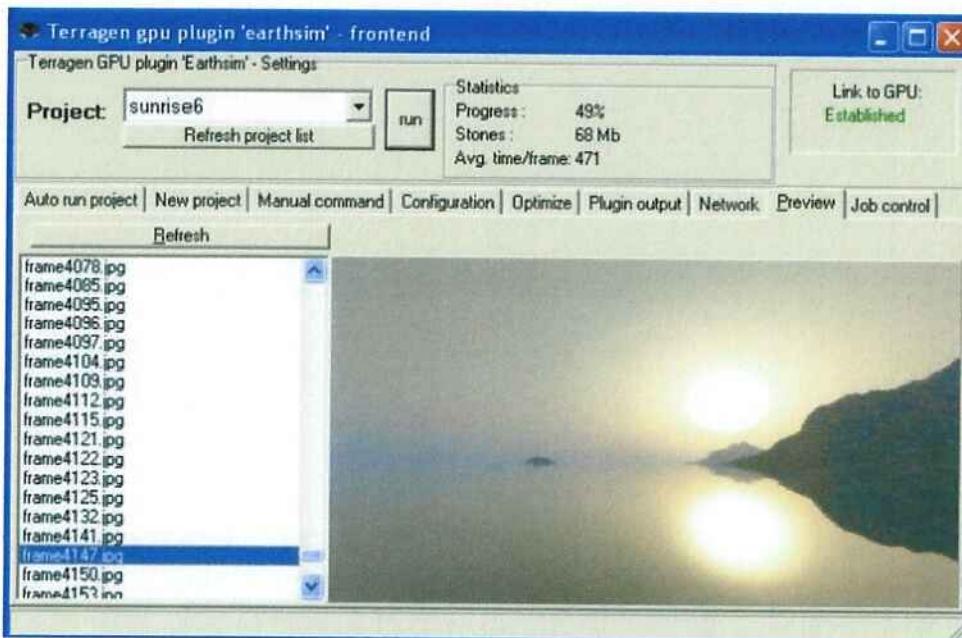


Figura 3 - Módulo de geração de terrenos no GPU

A Figura 3 demonstra uma aplicação gráfica de geração de terrenos em três dimensões que utiliza a plataforma GPU. Esta aplicação faz uso de vários computadores na rede para diminuir o tempo necessário para a construção de imagens de terrenos em três dimensões baseadas em modelos.

2.5 Avaliação do Desempenho de Aplicações Distribuídas

A avaliação de desempenho de aplicações distribuídas pode ser realizada de forma qualitativa meramente através da comparação do tempo de execução serial de uma tarefa computacional e o tempo de execução da mesma tarefa num sistema composto por um agrupamento de máquinas(*cluster*) operando em paralelo.

Para a realização de uma análise quantitativa, mais profunda, pode-se utilizar como parâmetro a lei de Amdahl.

2.5.1 A Lei de Amdahl

A lei de Amdahl recebe este nome devido a Gene Amdahl, um arquiteto de computadores e funcionário da IBM[35]. Em [36], AMDAHL demonstra o ganho de desempenho esperado em um sistema quando apenas parte deste sistema é melhorado. Seu trabalho é freqüentemente utilizado em computação paralela para prever o máximo ganho teórico de desempenho em sistemas paralelos.

A lei de Amdahl pode ser interpretada de forma mais técnica, mas em linhas gerais, diz que é o algoritmo que determina o ganho máximo de desempenho e não o número máximo de unidades de processamento, comportamento ilustrado na Figura 4. O limite é atingido quando uma maior paralelização do algoritmo é impossível.

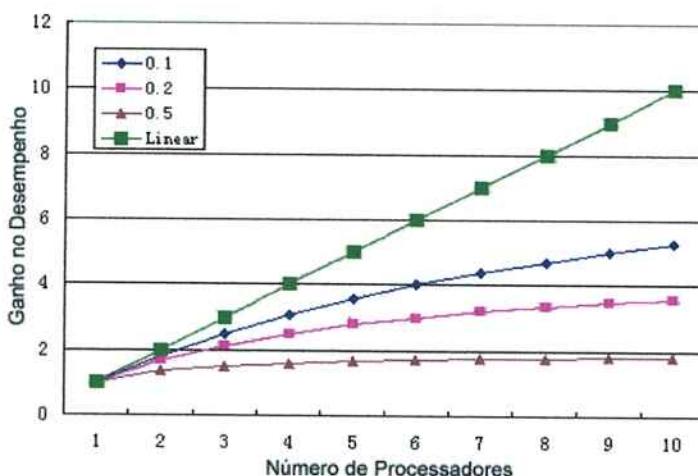


Figura 4 - Ganho no desempenho em função do número de processadores

A lei de Amdahl é uma aplicação computacional da lei econômica dos rendimentos decrescentes que em sua forma mais simples é dada por:

$$\frac{1}{(1-P) + \frac{P}{S}} \quad (1)$$

Onde:

- P – É um percentual indicando a representatividade da parcela do processo computacional passível de otimização;

- S – É a otimização prevista para a parcela P.

Em casos onde há a possibilidade de otimização de mais de uma porção da aplicação, a fórmula geral fica:

$$\frac{1}{\frac{P_1}{S_1} + \frac{P_2}{S_2} + \frac{P_3}{S_3} + \frac{P_4}{S_4} + \dots} \quad (2)$$

Onde P_n , assim como no caso anterior, é o valor percentual da representatividade da tarefa computacional perante o tempo total de execução do algoritmo. E S_n é o ganho obtido com a otimização de cada parcela P_n .

2.5.2 Paralelização

No caso específico da paralelização, a lei de Amdahl impõe que se F é a fração da aplicação que é seqüencial, isto é, que não pode se beneficiar da paralelização, e que $(1-F)$ é a fração que pode ser paralelizada, então, o máximo ganho de desempenho obtido com o uso de N centros de processamento é dado por:

$$\frac{1}{F + \frac{(1-F)}{N}} \quad (3)$$

No limite, com N tendendo ao infinito, o máximo ganho de desempenho teórico alcançável é de:

$$\frac{1}{F} \quad (4)$$

Por isso, a paralelização de uma aplicação é tanto melhor quanto menor for o índice F , ou seja, quanto menor for a fração seqüencial.

3 - ESPECIFICAÇÃO DO APLICATIVO

3.1 Introdução

Existe uma extensa gama de fatores que influenciam a execução de aplicativos em ambientes distribuídos. Alguns desses fatores foram abordados na revisão da literatura, mas não é possível tratar de todos no decorrer do projeto.

Algumas áreas como, por exemplo, os algoritmos que fazem a distribuição da carga de processamento na rede, são tão complexas que dispõem de pesquisas próprias. Portanto, buscaram-se soluções que propiciem uma estrutura mínima para a criação da aplicação ponto-a-ponto, de forma que esta tenha apenas um ganho de desempenho, mesmo que ele não seja ótimo.

3.1.1 Escolha da Plataforma de Desenvolvimento

As plataformas de desenvolvimento descritas no item 2.3.3 foram analisadas com o objetivo de determinar a mais aplicável para a resolução do problema do cálculo simultâneo de perdas técnicas em redes elétricas.

A maioria das plataformas p2p foram desenvolvidas visando um dos dois objetivos: ou o compartilhamento de arquivos, ou a criação de um ambiente colaborativo. Sob este ponto de vista, nenhuma das plataformas se adequa precisamente ao problema.

No processo de seleção, algumas plataformas foram descartadas devido à linguagem na qual foram codificadas. Além disso, outro parâmetro para escolha da plataforma é a facilidade de integração com o módulo de cálculo de perdas técnicas desenvolvido por MÉFFE.

Devido a essas características, a plataforma escolhida foi a GPU, baseada no protocolo Gnutella.

3.1.2 Objetivos do Aplicativo

O objetivo do aplicativo é atestar o uso da computação distribuída, mais especificamente do modelo de computação p2p, como uma ferramenta útil na otimização do processo de análise das perdas técnicas em redes elétricas.

Como o objetivo principal da aplicação não é obter o máximo ganho possível, optou-se por paralelizar o algoritmo do processo de cálculo e não o código fonte do módulo de MÉFFE.

3.1.3 Escopo do Aplicativo

3.1.3.1 Nome e componentes principais

O aplicativo é denominado simplesmente de Módulo de Cálculo Distribuído. Este módulo opera em conjunto com os outros dois módulos já descritos: o módulo de comunicação fornecido pela plataforma GPU e o módulo de cálculo implementado por MÉFFE.

Estes três componentes, devidamente adaptados para operarem em conjunto, integram o aplicativo de Cálculo Distribuído, cujo módulo principal será responsável pela coordenação e execução das tarefas.

3.1.3.2 Limites de Operação

Os ganhos de desempenho esperados são limitados pela capacidade de divisão do problema em partes que possam ser enviadas aos nós para processamento, conforme o enunciado da lei de Amdahl.

O cálculo de perdas técnicas em redes elétricas de distribuição apresenta o problema adicional da agregação das perdas elétricas no decorrer do cálculo, isto é, as perdas de um segmento são freqüentemente utilizadas como dados de entrada no cálculo dos segmentos seguintes. Isto significa que os nós da rede, calculando segmentos relacionados, devem interagir diversas vezes para que apenas uma etapa de cálculo seja concluída. Desta forma, a fração não paralelizável do processo de cálculo foi identificada como sendo a Subestação. Por isso, optou-se pela divisão espacial da rede agregando-se o cálculo das perdas por subestação.

3.2 Requisitos Gerais

3.2.1 Software e Sistema Operacional

O programa foi desenvolvido em linguagem de programação C++, aproveitando-se os recursos da orientação a objetos. Para o desenvolvimento foi escolhido o ambiente *Borland, C++ Builder* versão 6.0.

Esta escolha foi pautada na necessidade de integração com o módulo de cálculo já desenvolvido nesta mesma linguagem de programação. Além disso, o módulo de comunicação da plataforma GPU foi desenvolvido em *Delphi*, um tipo de Pascal também orientado a objetos e de fácil integração.

Logo, a versão final do programa somente estará disponível para o sistema Microsoft Windows.

3.2.2 Hardware

Os requisitos mínimos de hardware para a execução do software são os mesmos para a execução do sistema operacional.

Para a operação como ambiente distribuído faz-se necessário, também, a presença de uma rede de comunicação.

Desta forma, se um computador executa o sistema operacional e está conectado a outros por uma rede, ele pode utilizar o módulo de computação distribuída.

3.3 Interface

A interface com o usuário é bem limitada, visto que o objetivo do aplicativo não é a interação com o usuário, como se pode verificar na Figura 5.

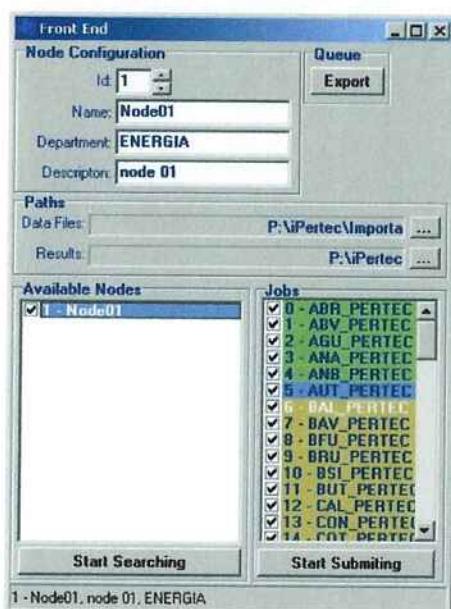


Figura 5- Interface do Módulo de Gerenciamento de Cálculo Distribuído

Através da interface apresentada, o usuário pode pesquisar a lista de nós de processamento disponíveis para uso, verificar seu estado de utilização e ainda selecionar quais nós se deseja utilizar no cálculo.

A lista de processos de cálculo disponíveis também é apresentada, e da mesma forma que na lista de nós, podemos visualizar o estado dos processos e selecionar quais devem ser executados.

Também está disponível na interface, um conjunto de parâmetros, editáveis pelo usuário, para identificação do nó, localização do módulo de cálculo e dos arquivos de entrada de dados e de saída dos resultados.

Têm-se ainda a função de exportação do arquivo com os tempos de processamento.

3.4 Arquitetura

O módulo de gerenciamento do cálculo distribuído é implementado na forma de uma biblioteca de ligação dinâmica (dll) que possui três componentes principais:

- *Plugin GPU* - Se comunica com o módulo GPU sendo responsável pelo envio e recebimento das requisições para outro nó e pela comunicação com o módulo de cálculo;
- *Front End* – Responsável pela interface com o usuário, gerenciamento dos parâmetros de configuração e de identidade dos nós. Além disso, é responsável pelo criação da lista de processos para execução e exibição da lista de nós disponíveis;
- Módulo de Controle – Este é o módulo responsável pela atribuição dos processos selecionados aos nós escolhidos. A lógica de atribuição pode ser alterada para se obter um comportamento diferente do cluster.

O diagrama da Figura 6 ilustra a arquitetura, a iteração entre os componentes e o fluxo dos dados no módulo de computação distribuída.

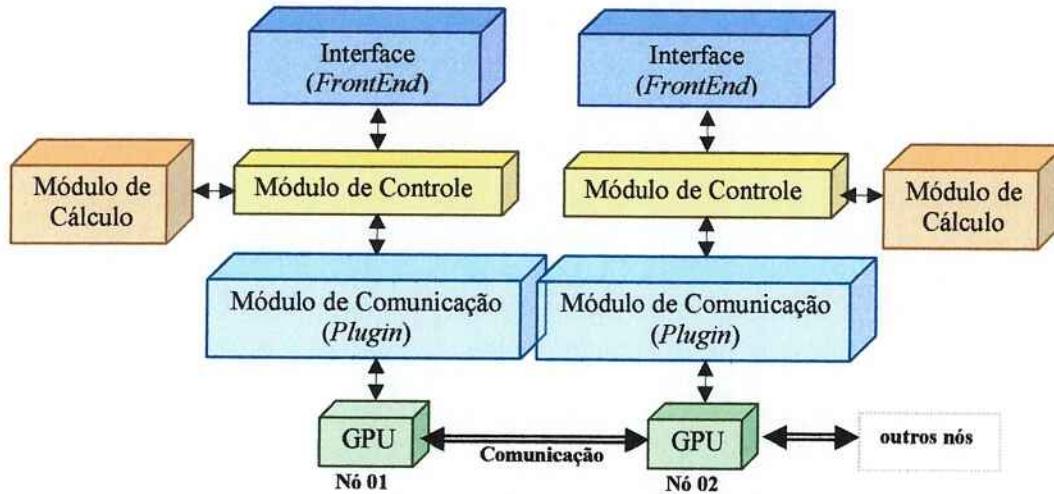


Figura 6 - Arquitetura e Componentes do Módulo de Computação Distribuída

3.5 Funcionamento

O funcionamento do aplicativo se dá segundo a seqüência a seguir:

- Inicialmente é carregada a plataforma GPU em todos os computadores que irão colaborar com o processamento;
- A plataforma GPU se encarrega automaticamente da carga do módulo de gerenciamento do cálculo distribuído em cada um;
- O módulo de gerenciamento por sua vez verifica num diretório previamente configurado a existência de arquivos de subestações não processados e a existência do módulo de cálculo de redes elétricas.
- É realizada a pesquisa por outros nós na rede p2p;
- O módulo está pronto para receber ou requerer processos de cálculo.

Após o início do processamento, o módulo de gerenciamento atribui o cálculo de determinadas subestações a cada um dos nós disponíveis na rede que as executam, retornando como resultado arquivos contendo um resumo das perdas elétricas.

4 - AVALIAÇÃO DO APLICATIVO

4.1 Metodologia

A metodologia adotada para a avaliação do aplicativo se baseia na análise dos tempos de processamento de uma rede elétrica padrão. Os tempos de processamento são comparados e observa-se o ganho ou não no desempenho do processamento com múltiplos nós se comparado a um caso-base. Adicionalmente é feita uma comparação com os tempos previstos pela lei de Amdahl.

4.2 Rede Elétrica em Estudo

A rede elétrica utilizada no estudo é complexa e real. Ela é composta por 70 subestações de distribuição, totalizando 733 circuitos de distribuição primários e cerca de 7000 circuitos de distribuição secundários alimentando alguns milhares de transformadores de distribuição. Esta rede foi escolhida devido à sua representatividade e ao longo tempo necessário para a sua análise.

4.3 Estrutura física e Testes Realizados

Para efeito de comparação utilizaram-se computadores idênticos como nós da rede p2p, com a configuração, a saber:

- Processador: Intel Pentium IV – 2,4Ghz;
- Memória física: 1Gb;
- Disco Rígido: 40Gb;
- Velocidade da rede: 10Mbps.

Foram realizados três tipos de testes:

- O primeiro, chamado de caso-base, foi realizado utilizando-se apenas um único nó do *cluster*. Este teste servirá como base para a análise comparativa dos outros dois casos;
- O segundo caso, chamado de primeiro-teste, faz uso de dois nós de processamento iguais. Estes nós são idênticos ao primeiro. Da análise deste caso já é possível tirar conclusões referentes sobre a viabilidade de utilização do modelo p2p;
- O terceiro caso, chamado de segundo-teste, é realizado com três nós iguais de processamento. Novamente, estes nós são idênticos ao utilizado na execução do caso-base. Deste teste, comparando aos dois anteriores, podemos ter uma noção da evolução do desempenho ao aumentarem-se novos nós na rede p2p.

4.4 Resultados Obtidos

4.4.1 Caso-base

Na Tabela 1 encontramos um resumo do resultado obtido na execução do caso-base. Verificamos que o tempo total de processamento da rede elétrica é da ordem de 6 horas.

	Hora Início	Hora Final	Tempo Gasto	Tempo de Proc.	Latência	Tempo Nô 01	Latência Nô 01
Totais	4:00:15	10:09:42	6:09:27	6:05:09	0:04:18	6:05:09	0:04:18
Médias				0:05:09	0:00:04	0:05:09	0:00:04

Tabela 1 - Resultados do Caso-base

Os tempos foram calculados com base nos arquivos de registro gerados pelo Módulo de Gerenciamento. Neste arquivo está especificado o tempo de início e término de cada tarefa, do recebimento e envio de mensagens. Por isso, foi possível extrair além

do tempo total e médios de cálculo, a latência existente entre as tarefas de processamento. Os valores foram obtidos para cada tarefa e depois as médias dos tempos foram calculadas.

4.4.2 Primeiro-Teste

Na Tabela 1 vê-se o resultado do primeiro-teste. Pode-se imediatamente notar, em comparação com o caso anterior, uma sensível redução no tempo gasto de processamento. A tabela completa com todos os dados encontra-se em anexo ao final do trabalho.

	Hora Início	Hora Final	Tempo Gasto	Tempo de Proc.	Latência Cluster	Tempo Nô 01	Tempo Nô 02	Latência Nô 01	Latência Nô 02
Totais	2:14:20	5:44:15	3:29:55	6:19:57	0:20:36	3:24:13	2:55:44	0:01:38	0:18:58
Médias				0:05:21	0:00:18	0:07:34	0:04:00	0:00:04	0:00:26

Tabela 2 - Estatísticas do primeiro teste

O tempo anterior de processamento que era da ordem de 6 horas foi reduzido para um valor de aproximadamente 3 horas e 30 minutos. Foi uma redução da ordem de 43%.

Cálculo do Tempo Previsto pela lei de Amdahl		
	Sem Latência	Com Latência
Tempo da Aplicação Serial:	6:09:27	
Tempo Seqüencial:	0:05:21	0:05:39
Fração Seqüencial (F):	0,01448	0,01529
Nós Utilizados:		2
Ganho Estimado:	1,97144	1,96987
Tempo Estimado:	3:07:24	3:07:33
Tempo Real Gasto:	3:29:55	

Tabela 3 - Tempos estimados e reais para o primeiro-teste

Na Tabela 3 é mostrado o tempo teórico calculado pela equação (3) da lei de Amdahl. O valor do parâmetro N é igual ao número de unidades de processamento, ou seja, igual a dois. Como valor para o parâmetro F da fração seqüencial da aplicação foi

utilizado o tempo médio de processamento de uma subestação, pois se considerou este tempo como não paralelizável, segundo exposto no item 3.1.3.2.

O tempo real de processamento na verdade é superior ao registrado devido à latência de comunicação, e execução de ações adjacentes ao processamento da rede, por isso, tentando obter dados mais precisos, considerou-se a latência como um acréscimo no tempo seqüencial da aplicação e seu valor médio foi agregado ao valor utilizado anteriormente. Desta forma calcularam-se os dados que constam da coluna identificada como “com latência”.

Podemos notar que embora o tempo de processamento não seja otimizado o suficiente para atingir o valor teórico de Amdahl, ainda sim é muito próximo, apresentando um erro da ordem de apenas 12%. Esta discrepância deve-se às aproximações realizadas no cálculo do tempo seqüencial da aplicação e na adoção de um tempo médio de processamento para representar todas as subestações. Mesmo assim, se considerarmos o tempo total de processamento, o erro é de apenas 5%.

4.4.3 Segundo-Teste

Nas Tabela 4 e Tabela 5 vêem-se o resultado do segundo-teste. Pode-se novamente notar, em comparação com ambos os casos anteriores, uma nova redução no tempo gasto no processamento. A tabela completa com todos os dados encontra-se em anexo ao final do trabalho.

	Hora Início	Hora Final	Tempo Gasto	Tempo de Proc.	Tempo Nº 01	Tempo Nº 02	Tempo Nº 03
Totais	6:47:36	9:28:07	2:40:31	6:39:44	1:44:59	2:21:38	2:33:07
Médias				0:05:38	0:04:34	0:05:54	0:06:23

Tabela 4 - Tempos de processamento segundo-teste

	Tempo de Proc.	Latência Cluster	Latência Nô 01	Latência Nô 02	Latência Nô 03
Totais	6:39:44	0:39:54	0:31:11	0:01:34	0:07:09
Médias	0:05:38	0:00:35	0:01:21	0:00:04	0:00:18

Tabela 5 - Latências de processamento segundo-teste

O tempo anterior de processamento que era da ordem de 3 horas e 30 minutos foi reduzido para um valor de aproximadamente 2 horas e 40 minutos. Foi uma redução da ordem de 23%. Essa redução é da ordem de 57% se comparado com o caso-base.

Cálculo do Tempo Previsto pela lei de Amdahl

	Sem Latência	Com Latência
Tempo da Aplicação Serial:	6:09:27	
Tempo Seqüencial:	0:05:38	0:06:13
Fração Seqüencial (F):	0,01524	0,01683
Nós Utilizados:		3
Ganho Estimado:	2,91127	2,90232
Tempo Estimado:	2:06:54	2:07:18
Tempo Real Gasto:		2:40:31

Tabela 6 - Tempos estimados e reais para o segundo-teste

Na Tabela 6 é mostrado novamente o tempo teórico calculado pela equação (3) da lei de Amdahl. O valor do parâmetro N é igual ao número de unidades de processamento, ou seja, igual a três. Como valor para o parâmetro F da fração seqüencial da aplicação foi utilizado o tempo médio de processamento de uma subestação, pois se considerou este tempo como não paralelizável, segundo exposto no item 3.1.3.2.

De forma análoga ao descrito no item anterior, o tempo real de processamento na verdade é superior ao registrado devido à latência de comunicação, e execução de ações adjacentes ao processamento da rede, por isso, tentando obter dados mais precisos, considerou-se a latência como um acréscimo no tempo seqüencial da aplicação

e seu valor médio foi agregado ao valor utilizado anteriormente. Desta forma calcularam-se os dados que constam da coluna identificada como “com latência”.

Podemos notar que embora o tempo de processamento não seja otimizado o suficiente para atingir o valor teórico de Amdahl, ainda sim é muito próximo. Esta discrepância deve-se às aproximações realizadas no cálculo do tempo seqüencial da aplicação e na adoção de um tempo médio de processamento para representar todas as subestações. Mesmo assim, se considerarmos o tempo total de processamento, o erro é de apenas 8%.

5 - CONCLUSÕES

No decorrer do presente trabalho procurou-se apresentar a computação ponto-a-ponto como uma ferramenta computacional auxiliar na resolução de problemas com alta demanda computacional na área de sistemas elétricos de potência. Em especial, a aplicação utilizada foi voltada para o cálculo das perdas técnicas em redes elétricas de distribuição.

Mostrou-se que é possível ganhos de desempenho absolutos na análise de redes reais utilizando-se o paradigma p2p, cujo modelo foi implementado através da plataforma de desenvolvimento GPU. Foi possível observar, com base nos três casos apresentados, que o tempo de processamento foi realmente reduzido com o uso de tal ferramenta.

Além disso, a comparação realizada com previsões teóricas do tempo de processamento indica que embora o aplicativo não tenha sido refinado para o desempenho ótimo, este já se encontra num bom patamar. E com isto, considera-se atingido o objetivo proposto pelo presente trabalho.

5.1 Tópico para futuro desenvolvimento

Conforme citado, devido à abrangência do tema, alguns tópicos não foram cobertos por este trabalho, porém devido a sua relevância estes são referidos para futuros desenvolvimentos.

O canal de comunicação interfere diretamente sobre a latência do sistema de processamento, por isso, um estudo detalhado sobre o comportamento da rede de comunicação serviria como base para uma melhor estimativa da latência de processamento imposta pelo canal de comunicação, assim como seu limite máximo.

Outro fator importante é a dinâmica utilizada na atribuição dos processos aos nós que irão executá-los. Este tópico está intimamente ligado a temas como a teoria de filas. No aplicativo atual, a atribuição dos processos é feita sem qualquer otimização, sendo atribuído o primeiro processo da fila de espera para a primeira máquina disponível. A aplicação de métodos de teoria das filas e a criação de critérios de atribuição favoreceriam uma redução ainda maior no tempo de processamento de redes elétricas grandes, principalmente em sistemas p2p onde os nós não possuam capacidades iguais de processamento e por isso mereçam um tratamento diferenciado na atribuição dos processos.

O último tópico fica por conta da plataforma utilizada no desenvolvimento. Durante a elaboração deste trabalho, surgiram outras plataformas para computação p2p que não puderam e devem ser mais bem analisadas e possivelmente utilizadas como base nos novos desenvolvimentos.

6 - ABREVIATURAS

- ET – Estação Transformadora
- SE – Subestação
- EPE – Empresa de Pesquisa Energética
- API – Application Programming interface
- GPU – Global Processing Unit
- p2p – peer-to-peer
- HPC – High Performance Computing
- 3D – Três dimensões
- DLL – Dynamic Linking Library

7 - REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Empresa de Pesquisa Energética – EPE. **Boletim de Mercado**. Maio 2006. Rio de Janeiro. Ministério de Minas e Energia, Maio 2006. 27 p.
- [2] Associação Brasileira dos Distribuidores de Energia Elétrica – ABRADEE. Método Para a Definição dos Investimentos em Função dos Níveis de Qualidade - Horizonte de Médio e de Longo Prazo. Relatório CODI-19.33.
- [3] PENTEADO JUNIOR, Aderbal de Arruda; HASHIMOTO, Kleber. **Cálculo das Perdas Técnicas Regulares**, Brasília – DF, setembro 2005. Apresentação.
- [4] MÉFFE, André. **Metodologia Para Cálculo de Perdas Técnicas Por Segmento do Sistema de Distribuição**, 2001, 152 p. Dissertação (Mestrado em Engenharia) – Escola Politécnica da Universidade de São Paulo, Universidade de São Paulo, São Paulo. 2001.
- [5] KAGAN, Nelson. **Introdução aos Sistemas de Distribuição de Energia Elétrica**. Editora Edgard Blücher, São Paulo, 2005
- [6] BURGHART, Ted. **Distributed Computing Overview**. 1998. Disponível em: <<http://library.thinkquest.org/C007645/english/1-basic.htm>> Acesso em: julho 2006
- [7] Institute of Electrical and Electronics Engineers. **IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries**. New York, NY: 1990.
- [8] Distributed Computing: Fundamentals, Simulations, and Advanced Topics By Hagit Attiya, Jennifer Welch.
- [9] Flynn, M., **Some Computer Organizations and Their Effectiveness**, IEEE Trans. Comput., Vol. C-21, pp. 948, 1972.
- [10] Duncan, Ralph, **A Survey of Parallel Computer Architectures**, IEEE Computer. February 1990, pp. 5-16
- [11] Rüdiger Schollmeier, **A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications**, IEEE Proceedings of the First International Conference on Peer-to-Peer Computing, Institute of Communication Networks, Technische Universität München, 2002.
- [12] Heylighen, F. 1997. **Principia Cybernetica Web**. <<http://pespmc1.vub.ac.be/SELFORG.html>>, visitado em julho 2006.
- [13] Shamir, A. 1979. **How to share a secret**. Communications of the ACM, vol. 22, n. 11, pp. 612-613, Nov. 1979.

- [14] Castro, M. and Liskov, B. 1999. **Practical Byzantine Fault Tolerance** Proc. Usenix OSDI, Berkeley, Califórnia.
- [15] Necula, G. 1997. **Proof-Carrying Code**. Proceedings of the 24th Annual ACM Symposium on Principles of Programming Languages, pp 106-119. Paris France, January 1997.
- [16] Ball, T. and Rajamani, S. K. (2001) **Automatically validating temporal safety properties of interfaces**. In the Proceedings of the International SPINWorkshop on Model Checking of Software, pp. 333-344, May 2001.
- [17] Coulouris, G., Dollimore, J., and Kindberg, T., **Distributed Systems. Concepts and Design**. Addison Wesley, 2001.
- [18] **The JXTA home page**. <<http://www.jxta.org>>, Página do projeto,visitado em julho de 2006.
- [19] Lewis, T. G. and El-Rewini, H., **Introduction to Parallel Computing**, Prentice-Hall, 1992.
- [20] Shi, Yan. **Reevaluating Amdahl's Law and Gustafson's Law**, Temple University (MS:38-24) , Philadelphia, PA, 1996.
- [21] **Lotus Notes**. <<http://www-142.ibm.com/software/sw-lotus/products/product4.nsf/wdocs/noteshomepage>>, visitado em julho de 2006
- [22] Ivkovic, Igor. **Improving Gnutella Protocol: Protocol Analysis And Research Proposals**, University of Waterloo, Canada.
- [23] **SETI@home**. <<http://setiathome.berkeley.edu>>, Página do projeto, visitado em julho de 2006.
- [24] **setibr**. <<http://www.setiathome.com.br>>, Página brasileira de apoio ao projeto Seti@Home, visitado em julho de 2006.
- [25] Gedye, David; Anderson, David, **A new major SETI project based on Project SERENDIP data and 100,000 personal computers**, Conference Paper, Astronomical and Biochemical Origins and the Search for Life in the Universe, IAU Colloquium 161, Publisher: Bologna, Italy, 1997.
- [26] **Intel**, <<http://www.intel.com>>, Página da empresa, visitado em julho de 2006.
- [27] **Groove Networks**,
<<http://www.groove.net/index.cfm/pagename/VirtualOffice>> Página do Produto, visitado em julho de 2006.
- [28] Bolcer, G. 2001, **Magi: An Architecture for Mobile and Disconnected Workflow**, Endeavors Technology Inc., 2000.
- [29] Mike, G. **The FreeNetwork Project**, Electronic Frontier Foundation, 2001.

- [30] **The FreeNetwork Project**, <<http://www.freenetproject.org>>, visitado em julho de 2006.
- [31] **Microsoft peer-to-peer Networking**,
<<http://www.microsoft.com/technet/itsolutions/network/p2p/default.mspx>>, Página do aplicativo, visitado em julho de 2006
- [32] Milojicic, Dejan S., **Peer-to-Peer Computing**, HP Laboratories, Palo Alto, 2003.
- [33] Mengotti, Tiziano. **GPU: A framework for distributed computing over Gnutella**, Dissertação (Mestrado em Ciência da Computação) – ETH Zürich, Switzerland. 2004.
- [34] **Global Processing Unit**, <http://gpu.sourceforge.net/docdev.php>, Página do projeto, visitado em julho de 2006.
- [35] **IBM – International Business Machines**, <<http://www.ibm.com>>, Página da Empresa, visitado em julho de 2006.
- [36] Gene Amdahl, **Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities**, AFIPS Conference Proceedings, (30), pp. 483-485, 1967.
- [37] Case, Karl E., Fair, Ray C. **Principles of Economics**, Prentice-Hall.

8 - ANEXOS

8.1 Figuras Adicionais

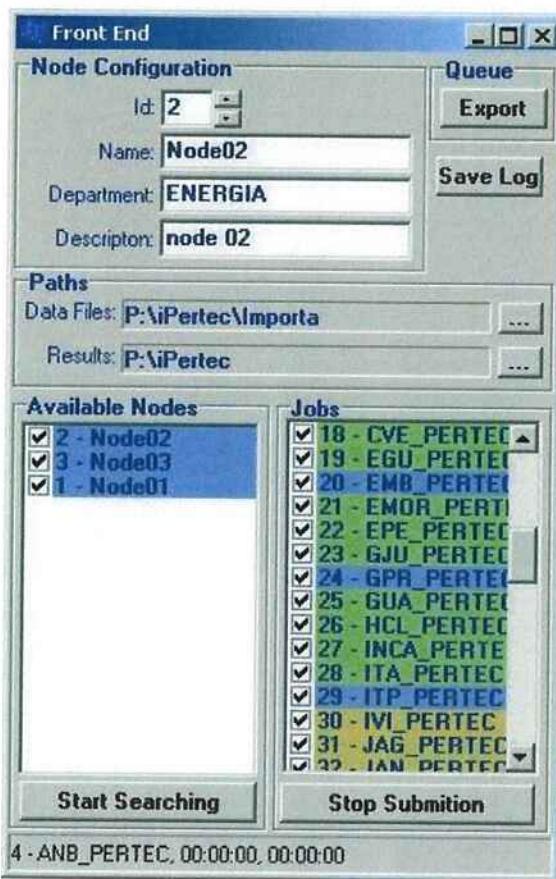


Figura 7 - Módulo de gerenciamento executando três cálculos simultaneamente

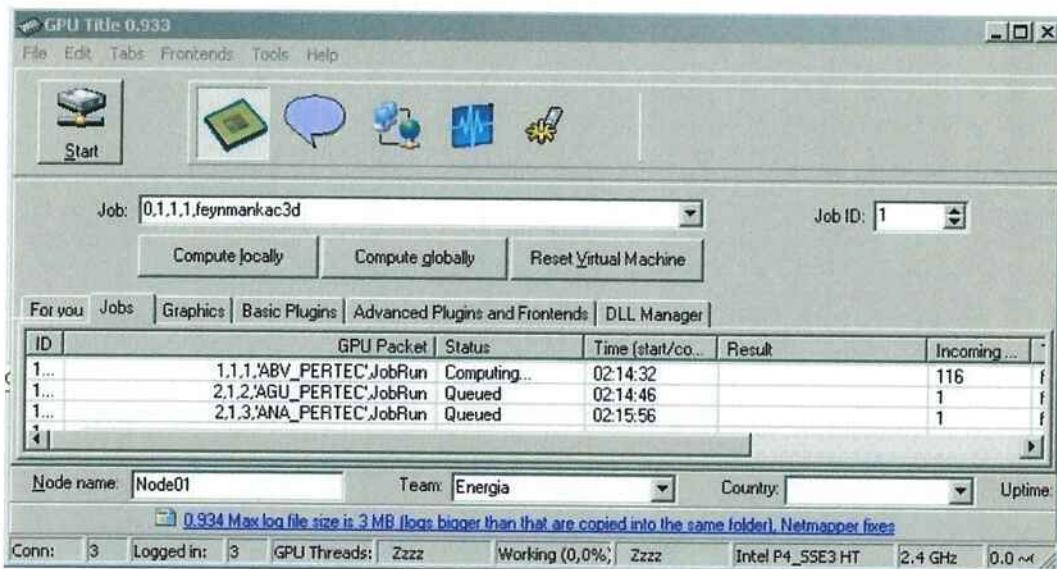


Figura 8 - Módulo GPU exibindo os processos em execução simultânea

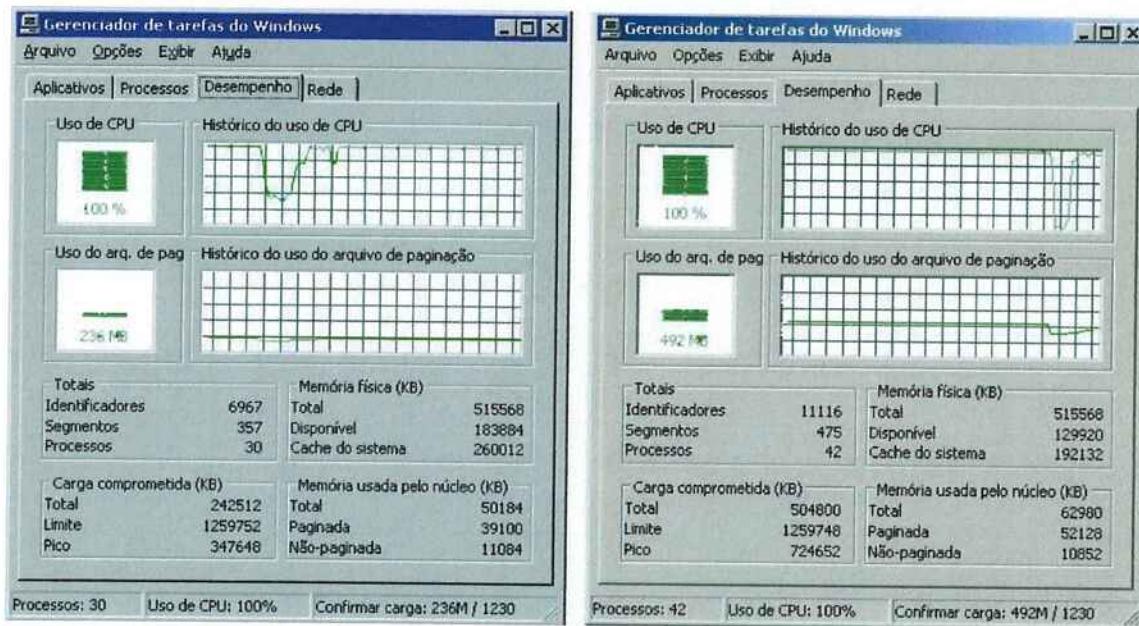


Figura 9 – Carga de trabalho de dois nós distintos, com flutuação devido à troca de processo

8.2 Tabelas dos Tempos de Cálculo

8.2.1 Caso-base

Processamento Singular							
Cód da SE	Nó Utilizado	Início	Final	Tempo de Proc.	Latência	Tempo Nô 01	Latência Nô 01
0	1	04:00:15	04:00:37	00:00:22	00:00:03	0:00:22	0:00:03
1	1	04:00:40	04:03:08	00:02:28	00:00:02	0:02:28	0:00:02
2	1	04:03:10	04:04:01	00:00:51	00:00:04	0:00:51	0:00:04
3	1	04:04:05	04:04:35	00:00:30	00:00:05	0:00:30	0:00:05
4	1	04:04:40	04:08:32	00:03:52	00:00:03	0:03:52	0:00:03
5	1	04:08:35	04:16:21	00:07:46	00:00:04	0:07:46	0:00:04
6	1	04:16:25	04:19:50	00:03:25	00:00:05	0:03:25	0:00:05
7	1	04:19:55	04:22:59	00:03:04	00:00:01	0:03:04	0:00:01
8	1	04:23:00	04:26:15	00:03:15	00:00:06	0:03:15	0:00:06
9	1	04:26:21	04:26:22	00:00:01	00:00:04	0:00:01	0:00:04
10	1	04:26:26	04:38:36	00:12:10	00:00:05	0:12:10	0:00:05
11	1	04:38:41	04:42:43	00:04:02	00:00:03	0:04:02	0:00:03
12	1	04:42:46	04:44:40	00:01:54	00:00:06	0:01:54	0:00:06
13	1	04:44:46	04:45:51	00:01:05	00:00:05	0:01:05	0:00:05
14	1	04:45:56	05:03:40	00:17:44	00:00:06	0:17:44	0:00:06
15	1	05:03:46	05:15:25	00:11:39	00:00:06	0:11:39	0:00:06
16	1	05:15:31	05:16:05	00:00:34	00:00:06	0:00:34	0:00:06
17	1	05:16:11	05:22:55	00:06:44	00:00:01	0:06:44	0:00:01
18	1	05:22:56	05:25:52	00:02:56	00:00:04	0:02:56	0:00:04
19	1	05:25:56	05:27:36	00:01:40	00:00:05	0:01:40	0:00:05
20	1	05:27:41	05:41:24	00:13:43	00:00:02	0:13:43	0:00:02
21	1	05:41:26	05:41:30	00:00:04	00:00:06	0:00:04	0:00:06
22	1	05:41:36	05:44:08	00:02:32	00:00:03	0:02:32	0:00:03
23	1	05:44:11	05:45:50	00:01:39	00:00:01	0:01:39	0:00:01
24	1	05:45:51	06:53:10	01:07:19	00:00:06	1:07:19	0:00:06
25	1	06:53:16	06:56:45	00:03:29	00:00:06	0:03:29	0:00:06
26	1	06:56:51	06:56:57	00:00:06	00:00:04	0:00:06	0:00:04
27	1	06:57:01	06:57:05	00:00:04	00:00:06	0:00:04	0:00:06
28	1	06:57:11	06:57:13	00:00:02	00:00:03	0:00:02	0:00:03
29	1	06:57:16	07:07:17	00:10:01	00:00:05	0:10:01	0:00:05
30	1	07:07:22	07:17:28	00:10:06	00:00:04	0:10:06	0:00:04
31	1	07:17:32	07:18:08	00:00:36	00:00:04	0:00:36	0:00:04
32	1	07:18:12	07:29:59	00:11:47	00:00:03	0:11:47	0:00:03
33	1	07:30:02	07:32:41	00:02:39	00:00:06	0:02:39	0:00:06
34	1	07:32:47	07:35:14	00:02:27	00:00:03	0:02:27	0:00:03
35	1	07:35:17	07:36:07	00:00:50	00:00:06	0:00:50	0:00:06
36	1	07:36:13	07:36:15	00:00:02	00:00:03	0:00:02	0:00:03
37	1	07:36:18	07:38:50	00:02:32	00:00:03	0:02:32	0:00:03
38	1	07:38:53	07:49:00	00:10:07	00:00:03	0:10:07	0:00:03
39	1	07:49:03	07:58:43	00:09:40	00:00:05	0:09:40	0:00:05
40	1	07:58:48	08:10:30	00:11:42	00:00:03	0:11:42	0:00:03
41	1	08:10:33	08:18:51	00:08:18	00:00:02	0:08:18	0:00:02

42	1	08:18:53	08:18:56	00:00:03	00:00:02	0:00:03	0:00:02
43	1	08:18:58	08:20:40	00:01:42	00:00:03	0:01:42	0:00:03
44	1	08:20:43	08:22:51	00:02:08	00:00:02	0:02:08	0:00:02
45	1	08:22:53	08:24:20	00:01:27	00:00:03	0:01:27	0:00:03
46	1	08:24:23	08:35:25	00:11:02	00:00:03	0:11:02	0:00:03
47	1	08:35:28	08:43:11	00:07:43	00:00:02	0:07:43	0:00:02
48	1	08:43:13	08:44:05	00:00:52	00:00:03	0:00:52	0:00:03
49	1	08:44:08	08:44:09	00:00:01	00:00:04	0:00:01	0:00:04
50	1	08:44:13	08:51:30	00:07:17	00:00:03	0:07:17	0:00:03
51	1	08:51:33	09:01:19	00:09:46	00:00:04	0:09:46	0:00:04
52	1	09:01:23	09:06:26	00:05:03	00:00:02	0:05:03	0:00:02
53	1	09:06:28	09:06:34	00:00:06	00:00:04	0:00:06	0:00:04
54	1	09:06:38	09:11:42	00:05:04	00:00:06	0:05:04	0:00:06
55	1	09:11:48	09:11:57	00:00:09	00:00:01	0:00:09	0:00:01
56	1	09:11:58	09:18:01	00:06:03	00:00:02	0:06:03	0:00:02
57	1	09:18:03	09:20:30	00:02:27	00:00:03	0:02:27	0:00:03
58	1	09:20:33	09:20:37	00:00:04	00:00:01	0:00:04	0:00:01
59	1	09:20:38	09:20:45	00:00:07	00:00:03	0:00:07	0:00:03
60	1	09:20:48	09:22:24	00:01:36	00:00:04	0:01:36	0:00:04
61	1	09:22:28	09:25:14	00:02:46	00:00:04	0:02:46	0:00:04
62	1	09:25:18	09:40:41	00:15:23	00:00:02	0:15:23	0:00:02
63	1	09:40:43	09:45:18	00:04:35	00:00:05	0:04:35	0:00:05
64	1	09:45:23	09:45:27	00:00:04	00:00:01	0:00:04	0:00:01
65	1	09:45:28	09:47:00	00:01:32	00:00:03	0:01:32	0:00:03
66	1	09:47:03	10:02:23	00:15:20	00:00:05	0:15:20	0:00:05
67	1	10:02:28	10:05:00	00:02:32	00:00:03	0:02:32	0:00:03
68	1	10:05:03	10:06:48	00:01:45	00:00:06	0:01:45	0:00:06
69	1	10:06:54	10:09:36	00:02:42	00:00:03	0:02:42	0:00:03
70	1	10:09:39	10:09:42	00:00:03		0:00:03	
Totais			6:05:09	0:04:18	6:05:09	0:04:18	
Médias			0:05:09	0:00:04	0:05:09	0:00:04	
				Hora Início	Hora Final	Tempo Gasto	
				4:00:15	10:09:42	6:09:27	

8.2.2 Primeiro -teste

Agrupamento de dois nós									
Cód da SE	Nó Utilizado	Início	Final	Tempo de Proc.	Latência	Tempo N° 01	Tempo N° 02	Latência N° 01	Latência N° 02
0	2	02:14:20	02:14:26	00:00:06	00:00:09		0:00:06		0:00:09
1	1	02:14:32	02:17:38	00:03:06	00:00:03	0:03:06		0:00:03	
2	2	02:14:35	02:15:34	00:00:59	00:00:11		0:00:59		0:00:11
3	2	02:15:45	02:16:25	00:00:40	00:00:10		0:00:40		0:00:10
4	2	02:16:35	02:20:58	00:04:23	00:00:07		0:04:23		0:00:07
5	1	02:17:41	02:25:58	00:08:17	00:00:03	0:08:17		0:00:03	
6	2	02:21:05	02:25:21	00:04:16	00:00:09		0:04:16		0:00:09
7	2	02:25:30	02:31:14	00:05:44	00:00:11		0:05:44		0:00:11
8	1	02:26:01	02:30:07	00:04:06	00:00:04	0:04:06		0:00:04	
9	1	02:30:11	02:30:19	00:00:08	00:00:02	0:00:08		0:00:02	
10	1	02:30:21	02:43:31	00:13:10	00:00:04	0:13:10		0:00:04	
11	2	02:31:25	02:38:46	00:07:21	00:00:08		0:07:21		0:00:08
12	2	02:38:54	02:42:03	00:03:09	00:00:11		0:03:09		0:00:11
13	2	02:42:14	02:44:24	00:02:10	00:04:06		0:02:10		0:04:06
14	1	02:43:35	03:03:34	00:19:59	00:00:04	0:19:59		0:00:04	
15	2	02:48:30	03:02:07	00:13:37	00:00:10		0:13:37		0:00:10
16	2	03:02:17	03:02:58	00:00:41	00:00:09		0:00:41		0:00:09
17	2	03:03:07	03:12:15	00:09:08	00:00:08		0:09:08		0:00:08
18	1	03:03:38	03:06:48	00:03:10	00:00:05	0:03:10		0:00:05	
19	1	03:06:53	03:08:37	00:01:44	00:00:08	0:01:44		0:00:08	
20	1	03:08:45	03:25:40	00:16:55	00:00:04	0:16:55		0:00:04	
21	2	03:12:23	03:12:34	00:00:11	00:00:18		0:00:11		0:00:18
22	2	03:12:52	03:15:51	00:02:59	00:00:06		0:02:59		0:00:06
23	2	03:15:57	03:17:46	00:01:49	00:00:06		0:01:49		0:00:06

24	2	03:17:52	03:23:03	00:05:11	00:00:10	0:05:11	0:00:10
25	2	03:23:13	03:28:16	00:05:03	00:00:01	0:05:03	0:00:01
26	2	03:28:17	03:28:25	00:00:08	00:00:00	0:00:08	0:00:00
27	1	03:25:44	03:25:52	00:00:08	0:00:02	0:00:08	0:00:02
28	1	03:25:54	03:25:56	00:00:02	00:00:03	0:00:02	0:00:03
29	1	03:26:59	03:38:17	00:12:18	00:00:02	0:12:18	0:00:02
30	2	03:28:25	03:44:57	00:16:32	00:00:00	0:16:32	0:00:00
31	2	03:44:57	03:45:04	00:00:07	00:00:00	0:00:07	0:00:00
32	1	03:38:19	03:58:21	00:20:02	00:00:05	0:20:02	0:00:05
33	2	03:45:04	03:49:48	00:04:44	00:00:11	0:04:44	0:00:11
34	2	03:49:59	03:54:25	00:04:26	00:00:09	0:04:26	0:00:09
35	2	03:54:34	03:56:09	00:01:35	00:00:11	0:01:35	0:00:11
36	2	03:56:20	03:56:28	00:00:08	00:00:12	0:00:08	0:00:12
37	2	03:56:40	03:59:46	00:03:06	00:00:08	0:03:06	0:00:08
38	1	03:58:26	04:10:31	00:12:05	00:00:02	0:12:05	0:00:02
39	2	03:59:54	04:11:07	00:11:13	00:00:10	0:11:13	0:00:10
40	1	04:10:33	04:23:41	00:13:08	00:00:04	0:13:08	0:00:04
41	2	04:11:17	04:20:51	00:09:34	00:00:09	0:09:34	0:00:09
42	2	04:21:00	04:21:06	00:00:06	00:00:09	0:00:06	0:00:09
43	2	04:21:15	04:23:11	00:01:56	00:00:09	0:01:56	0:00:09
44	2	04:23:20	04:25:41	00:02:21	00:00:04	0:02:21	0:00:04
45	1	04:23:45	04:25:36	00:01:51	00:00:04	0:01:51	0:00:04
46	1	04:25:40	04:37:34	00:11:54	00:00:06	0:11:54	0:00:06
47	2	04:25:45	04:33:47	00:08:02	00:00:16	0:08:02	0:00:16
48	2	04:34:03	04:35:01	00:00:58	00:00:07	0:00:58	0:00:07
49	2	04:35:08	04:35:12	00:00:04	00:00:11	0:00:04	0:00:11
50	2	04:35:23	04:43:07	00:07:44	00:02:02	0:07:44	0:02:02
51	1	04:37:40	04:50:33	00:12:53	00:00:02	0:12:53	0:00:02
52	2	04:45:09	04:54:20	00:09:11	00:06:29	0:09:11	0:06:29
53	1	04:50:35	04:50:49	00:00:14	00:00:02	0:00:14	0:00:02
54	1	04:50:51	04:59:25	00:08:34	00:00:01	0:08:34	0:00:01
55	1	04:59:26	04:59:41	00:00:15	00:00:06	0:00:15	0:00:06

56	1	04:59:47	05:07:06	00:07:19	00:00:05	0:07:19	0:00:05
57	2	05:00:49	05:03:32	00:02:43	00:00:07	0:02:43	0:00:07
58	2	05:03:39	05:03:45	00:00:06	00:00:09	0:00:06	0:00:09
59	2	05:03:54	05:04:03	00:00:09	00:00:21	0:00:09	0:00:21
60	2	05:04:24	05:06:07	00:01:43	00:00:38	0:01:43	0:00:38
61	2	05:06:45	05:10:29	00:03:44	00:00:10	0:03:44	0:00:10
62	1	05:07:11	05:29:02	00:21:51	00:00:05	0:21:51	0:00:05
63	2	05:10:39	05:18:59	00:08:20	00:00:07	0:08:20	0:00:07
64	2	05:19:06	05:19:17	00:00:11	00:00:09	0:00:11	0:00:09
65	2	05:19:26	05:22:11	00:02:45	00:00:10	0:02:45	0:00:10
66	2	05:22:21	05:44:15	00:06:41		0:06:41	0:00:00
67	1	05:29:07	05:32:51	00:03:44	00:00:06	0:03:44	0:00:06
68	1	05:32:57	05:35:59	00:03:02	00:00:03	0:03:02	0:00:03
69	1	05:36:02	05:40:14	00:04:12	00:00:03	0:04:12	0:00:03
70	1	05:40:17	05:40:23	00:00:06		0:00:06	0:00:00

Estatísticas do Cluster

	Hora Início	Hora Final	Tempo Gasto	Tempo de Proc.	Latência	Tempo Nô 01	Tempo Nô 02	Latência Nô 01	Latência Nô 02
Totais	2:14:20	5:44:15	3:29:55	6:19:57	0:20:36	3:24:13	2:55:44	0:01:38	0:18:58
Médias				0:05:21	0:00:18	0:07:34	0:04:00	0:00:04	0:00:26

Cálculo do Tempo Previsto
pela lei de Amdahl

Tempo da Aplicação Serial: 6:09:27

Tempo Sequencial: 0:05:21 0:05:39
Fração Seqüencial (F): 0,01448 0,01529

Nós Utilizados: 2

Ganho Estimado: 1,97144 1,96987

Tempo Estimado: 3:07:24 3:07:33

Tempo Real Gasto: 3:29:55

8.2.3 Segundo-teste

Agrupamento de três nós											
Cód da SE	Nó Utilizado	Ínicio	Final	Tempo de Proc.	Latência	Tempo Nó 01	Tempo Nó 02	Tempo Nó 03	Latência Nó 01	Latência Nº 02	Latência Nº 03
0	2	06:47:36	06:47:51	00:00:15	00:00:05		0:00:15			0:00:05	
1	3	06:47:51	06:51:55	00:04:04	00:03:02		0:04:04			0:03:02	
2	1	06:47:48	06:49:00	00:01:12	00:00:02	0:01:12			0:00:02		
3	2	06:47:56	06:48:36	00:00:40	00:00:05		0:00:40			0:00:05	
4	2	06:48:41	06:52:28	00:03:47	00:00:04		0:03:47			0:00:04	
5	1	06:49:02	06:57:32	00:08:30	00:00:09	0:08:30			0:00:09		
6	2	06:52:32	06:56:34	00:04:02	00:00:01		0:04:02			0:00:01	
7	3	06:54:57	06:59:13	00:04:16	00:00:13		0:04:16			0:00:13	
8	2	06:56:35	06:59:57	00:03:22	00:00:05		0:03:22			0:00:05	
9	1	06:57:41	06:57:42	00:00:01	00:00:09	0:00:01			0:00:09		
10	1	06:57:51	07:10:36	00:12:45	00:00:11	0:12:45			0:00:11		
11	3	06:59:26	07:05:45	00:06:19	00:00:14		0:06:19			0:00:14	
12	2	07:00:02	07:01:56	00:01:54	00:00:01		0:01:54			0:00:01	
13	2	07:01:57	07:03:11	00:01:14	00:00:06		0:01:14			0:00:06	
14	2	07:03:17	07:22:23	00:19:06	00:00:06		0:19:06			0:00:06	
15	3	07:05:59	07:22:47	00:16:48	00:00:02		0:16:48			0:00:02	
16	1	07:10:47	07:11:23	00:00:36	00:00:09	0:00:36			0:00:09		
17	1	07:11:32	07:18:29	00:06:57	00:00:15	0:06:57			0:00:15		
18	1	07:18:44	07:21:57	00:03:13	00:00:11	0:03:13			0:00:11		
19	1	07:22:08	07:24:00	00:01:52	00:00:10	0:01:52			0:00:10		
20	2	07:22:29	07:38:45	00:16:16	00:00:06		0:16:16			0:00:06	
21	3	07:22:49	07:23:00	00:00:11	00:00:04		0:00:11			0:00:04	
22	3	07:23:04	07:26:08	00:03:04	00:00:11		0:03:04			0:00:11	
23	1	07:24:10	07:25:54	00:01:44	00:00:11		0:01:44			0:00:11	
24	1	07:26:05	07:43:58	00:17:53	00:00:09	0:17:53			0:00:09		
25	3	07:26:19	07:30:22	00:04:03	00:00:15		0:04:03			0:00:15	

26	3	07:30:37	07:30:47	00:00:10	00:00:10	0:00:10	0:00:10
27	3	07:30:57	07:31:04	00:00:07	00:00:08	0:00:07	0:00:08
28	3	07:31:12	07:31:16	00:00:04	00:00:06	0:00:04	0:00:06
29	3	07:31:22	07:45:32	00:14:10	00:00:00	0:14:10	0:00:00
30	2	07:38:51	07:52:56	00:14:05	00:00:03	0:14:05	0:00:03
31	1	07:44:07	07:44:55	00:00:48	00:00:01	0:00:48	0:00:01
32	1	07:44:56	08:00:18	00:15:22	00:00:11	0:15:22	0:00:11
33	3	07:45:32	07:45:44	00:00:12	00:00:00	0:00:12	0:00:00
34	3	07:45:44	07:49:16	00:03:32	00:00:08	0:03:32	0:00:08
35	3	07:49:24	07:50:43	00:01:19	00:00:11	0:01:19	0:00:11
36	3	07:50:54	07:51:04	00:00:10	00:00:10	0:00:10	0:00:10
37	3	07:51:14	07:55:38	00:04:24	00:00:09	0:04:24	0:00:09
38	2	07:52:59	08:04:42	00:11:43	00:00:04	0:11:43	0:00:04
39	3	07:55:47	08:10:29	00:14:42	00:01:12	0:14:42	0:01:12
40	1	08:00:29	08:13:08	00:12:39	00:00:07	0:12:39	0:00:07
41	2	08:04:46	08:13:35	00:08:49	00:00:04	0:08:49	0:00:04
42	3	08:11:41	08:11:49	00:00:08	00:00:07	0:00:08	0:00:07
43	3	08:11:56	08:14:22	00:02:26	00:00:09	0:02:26	0:00:09
44	1	08:13:15	08:16:08	00:02:53	00:00:08	0:02:53	0:00:08
45	2	08:13:39	08:15:14	00:01:35	00:00:01	0:01:35	0:00:01
46	3	08:14:31	08:29:33	00:15:02	00:00:07	0:15:02	0:00:07
47	2	08:15:15	08:23:22	00:08:07	00:00:06	0:08:07	0:00:06
48	1	08:16:16	08:17:16	00:01:00	00:00:11	0:01:00	0:00:11
49	1	08:17:27	08:17:28	00:00:01	00:00:09	0:00:01	0:00:09
50	1	08:17:37	08:25:17	00:07:40	00:21:34	0:07:40	0:21:34
51	2	08:23:28	08:33:12	00:09:44	00:00:06	0:09:44	0:00:06
52	3	08:29:40	08:36:52	00:07:12	00:00:09	0:07:12	0:00:09
53	2	08:33:18	08:33:28	00:00:10	00:00:06	0:00:10	0:00:06
54	2	08:33:34	08:38:48	00:05:14	00:00:06	0:05:14	0:00:06
55	3	08:37:01	08:37:15	00:00:14	00:00:11	0:00:14	0:00:11
56	3	08:37:26	08:46:26	00:09:00	00:00:11	0:09:00	0:00:11
57	2	08:38:54	08:41:27	00:02:33	00:00:02	0:02:33	0:00:02

58	2	08:41:29	08:41:35	00:00:06	00:00:04	0:00:06		0:00:04
59	2	08:41:39	08:41:48	00:00:09	00:00:01	0:00:09		0:00:01
60	2	08:41:49	08:43:31	00:01:42	00:00:03	0:01:42		0:00:03
61	2	08:43:34	08:46:37	00:03:03	00:00:03	0:03:03		0:00:03
62	3	08:46:37	09:28:07	00:41:30		0:41:30		0:00:00
63	2	08:46:40	08:52:14	00:05:34	00:00:06	0:05:34		0:00:06
64	1	08:46:51	08:46:59	00:00:08	00:00:07	0:00:08		0:00:07
65	1	08:47:06	08:48:48	00:01:42	00:06:40	0:01:42		0:06:40
66	2	08:52:20	09:10:48	00:18:28		0:18:28		0:00:00
67	1	08:55:28	08:58:25	00:02:57	00:00:08	0:02:57		0:00:08
68	1	08:58:33	09:00:41	00:02:08	00:00:07	0:02:08		0:00:07
69	1	09:00:48	09:03:41	00:02:53	00:00:12	0:02:53		0:00:12
70	1	09:03:53	09:03:58	00:00:05		0:00:05		0:00:00

Estatísticas do Cluster

	Hora Início	Hora Final	Tempo Gasto	Tempo de Proc.	Latência	Tempo Nó 01	Tempo Nº 02	Tempo Nº 03	Latência Nº 01	Latência Nº 02	Latência Nº 03
Totais	6:47:36	9:28:07	2:40:31	6:39:44	0:39:54	1:44:59	2:21:38	2:33:07	0:31:11	0:01:34	0:07:09
Médias			0:05:38	0:00:35	0:04:34	0:05:54	0:06:23	0:01:21	0:00:04	0:00:18	

Cálculo do Tempo Previsto pela lei de Amdahl	Sem Latência	Com Latência
Tempo da Aplicação Serial:		6:09:27
Tempo Sequencial:	0:05:38	0:06:13
Fração Seqüencial (F):	0,01524	0,01683
Nós Utilizados:	3	
Ganho Estimado:	2,91127	2,90232
Tempo Estimado:	2:06:54	2:07:18
Tempo Real Gasto:	2:40:31	