4. Avaliação de Desempenho de Índices de Carga e Resultados

Neste capítulo é feita uma avaliação de desempenho dos índices de carga mais utilizados na literatura. Para tanto, é adotado o ambiente de escalonamento AMIGO (dyn**AM**ical flex**I**ble schedulin**G** envir**O**nment) (Souza, 2000).

4.1 Considerações Iniciais

O escalonamento pode alterar o desempenho das aplicações executadas em plataformas computacionais distribuídas, tanto positivamente quanto negativamente e, como já discutido anteriormente, um dos fatores com forte influência no sucesso de um escalonamento é o índice de carga.

Considerando os conceitos envolvidos na atividade de escalonar processos e os objetivos propostos para este escalonamento (neste caso o balanceamento de carga), faz-se necessário avaliar o desempenho obtido por diversos tipos de aplicações escalonadas utilizando-se distintos índices de carga.

Com este objetivo, far-se-á uso do ambiente AMIGO (Souza, 2000) e de suas políticas de escalonamento (DPWP (Araujo, 1999), I/O Best (Voorsluys & Souza, 2002), MimMax (Pereira & Souza, 2002), entre outras) para avaliar o desempenho de diversos índices de carga existentes na literatura.

O objetivo deste capítulo é testar e validar a eficiência dos índices de carga estabelecidos, demonstrando a possibilidade de se obter ganhos de desempenho quando um ambiente de escalonamento é utilizado fazendo uso de índices de carga apropriados.

Considerando que a qualidade do escalonamento é determinada pela política utilizada sob o AMIGO (principalmente no tocante aos índices de carga utilizados por essas políticas), e não pelo ambiente, os resultados descritos aqui são dependentes da política de escalonamento adotada.

As avaliações expostas neste capítulo demonstram o comportamento de aplicações paralelas executadas com e sem a utilização do AMIGO, citando as melhorias de desempenho obtidas de acordo com os índices de carga utilizados. É

analisado o impacto de fatores que afetam o escalonamento e a escolha dos índices de carga, tais como diferentes classes de aplicação, carga de trabalho da plataforma, entre outros, além da simples verificação de melhorias no desempenho.

4.2 Plataforma Utilizada para Realização dos Experimentos

As execuções foram realizadas no Laboratório de Sistemas Distribuídos e Programação Concorrente (LASDPC) do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC/USP). Os experimentos foram realizados em uma rede de computadores pessoais conectados por uma rede padrão *ethernet 100 Mbits* interligada por um *switch*, em que todos utilizavam o sistema operacional Linux (kernel 2.x.x) e o sistema de arquivos NFS. Essa organização de 10 máquinas possibilita a formação de uma arquitetura MIMD com memória distribuída, tornando factível a execução de aplicações paralelas desenvolvidas sobre ambientes que aceitam esse tipo de plataforma de hardware.

Com relação aos softwares utilizados nos experimentos, optou-se por verificar o desempenho de aplicações PVM (versão 3.3.11), uma vez que as aplicações e o ambiente não eram focos de avaliação, e sim os índices de carga implementados.

Durante a avaliação de desempenho impediu-se o acesso externo à plataforma computacional através da sua rede de comunicação, eliminando assim a presença de multiusuários.

4.3 Experimentos Realizados

Devido ao considerável número de fatores envolvidos em uma avaliação de desempenho, houve a necessidade de serem definidos experimentos visando contemplar alguns objetivos. Nesses experimentos, verificou-se o desempenho das aplicações[4] quando o escalonamento utiliza a política *round-robin* (que não leva em conta nenhum índice de carga para distribuir as tarefas aos processadores) ou quando se adota o AMIGO fazendo uso de políticas que fazem uso de diferentes índices de carga.

[4] Neste trabalho considera-se o desempenho de uma aplicação como sendo o seu tempo de execução.

Considerando que a avaliação a ser feita é dos índices de carga e que o intuito é validar e testar a eficiência desses índices, optou-se por cinco tipos de aplicações distintas: uma CPU-Bound, uma Memory-Bound, uma Disk-Bound, uma Network-Bound, e uma aplicação CPU-Bound/Network-Bound. A escolha por essas classes de aplicações permite que o desempenho obtido com o escalonamento seja avaliado em função da aplicação que o utiliza, e a influência do índice de carga de acordo com o tipo de aplicação que é submetido ao sistema.

Como **CPU-Bound** foi utilizada a aplicação "Método dos Trapézios Compostos" como definida em (Cortés, 1999). O método dos trapézios compostos é um algoritmo numérico típico que necessita de muito processamento e de pouca comunicação (vide Apêndice A). Na aplicação que utiliza o método do trapézio composto para solucionar integrais definidas, o algoritmo utilizou uma propriedade das integrais, onde dada uma função genérica $f(x)$ que possa ser integrada nos pontos de a até b, ela pode ser dividida no somatório de n outras integrais. Assim o cálculo das integrais é dividido entre os processos da aplicação e resolvido em paralelo. A comunicação acontece quando um processo mestre envia para cada escravo a faixa de integrais a ser calculada e quando os escravos enviam para o mestre as integrais já calculadas.

A Figura 4.1 apresenta o diagrama da paralelização do algoritmo do trapézio composto.

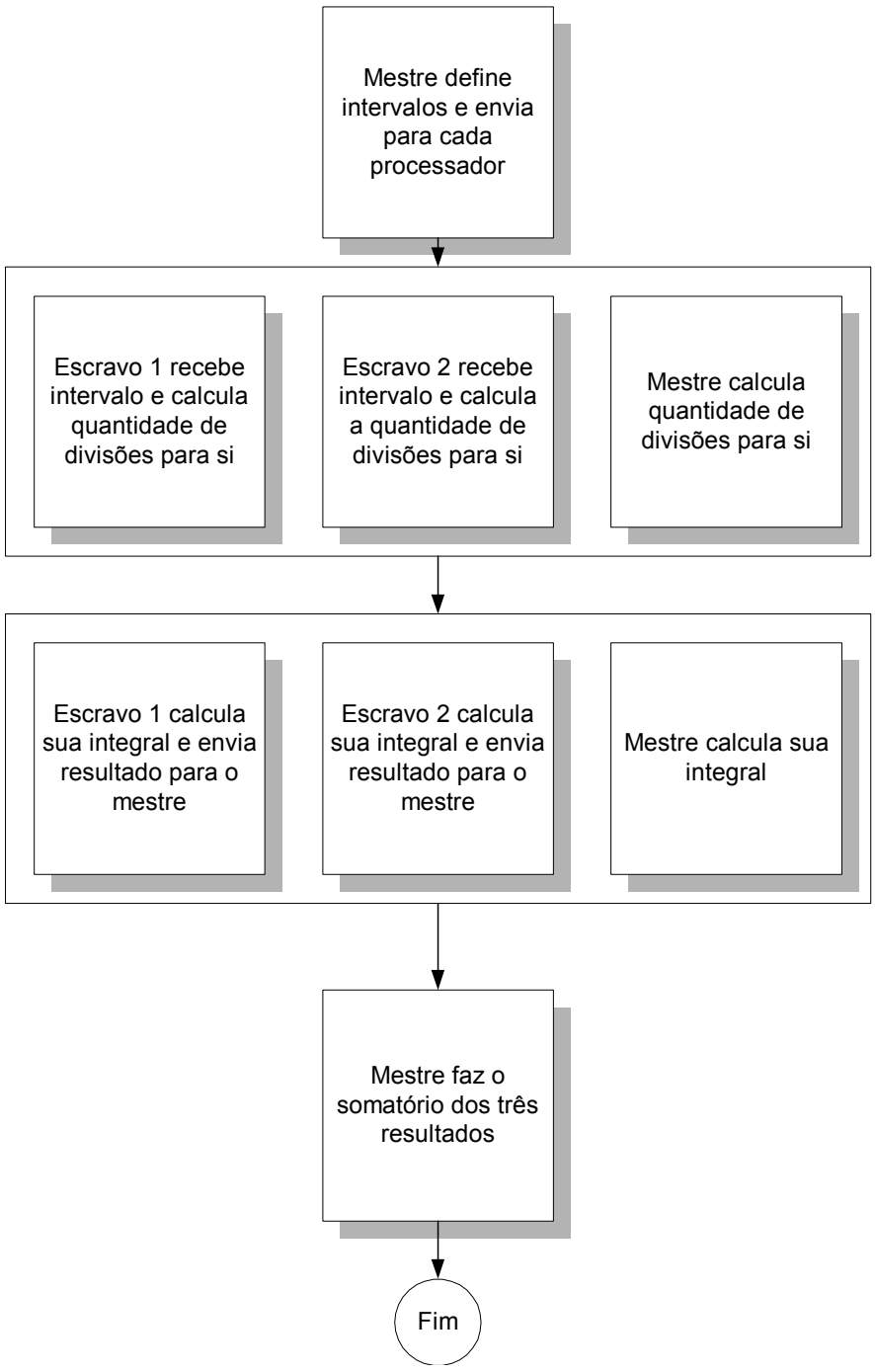

Figura 4.1 - Paralelização do algoritmo do trapézio composto.

Como **Memory-Bound** foi utilizada uma aplicação de multiplicação paralela de matrizes. O algoritmo é simples, onde a matriz resultante da multiplicação é obtida através do algoritmo a seguir:

```
Definir tamanho das matrizes quadradas -> TAM;
Inteiros i,j,k;
Reais A[TAM][TAM],B[TAM][TAM],C[TAM][TAM];
Criar matriz A e B;
Inicializar matriz C;
Para k=0 ate k< TAM faça
   Para i=0 até i < TAM faça
      Para j=0 até j < TAM faça
         C[k][i] ‹- C[k][i] + (A[k][j] * B[j][i]);
      Fim-para-j
   Fim-para-i
Fim-para-k
```

A paralelização foi feita da seguinte maneira: A geração das matrizes A e B é feita criando-se duas matrizes quadradas. O processo mestre gera a matriz A, e ao mesmo tempo os processos escravos geram a mesma matriz B. Ao terminarem o processo, o mestre manda a sua matriz gerada A para os processos escravos, e um dos processos escravos envia a matriz B para o mestre. De posse dos dados o mestre envia aos escravos a posição da linha onde cada processo iniciará a sua parte da multiplicação. A diferença é que a matriz não é efetivamente calculada, ou seja, os valores são simplesmente alocados na memória e retornados para o processo mestre. Posteriormente os escravos mandam os respectivos pedaços para o mestre. A Figura 4.2 mostra o diagrama da paralelização.

A título de exemplificação (uma vez que nos testes realizados as matrizes possuem tamanho 1000x1000), se as matrizes geradas tiverem o tamanho de 90x90 ocorrerá o seguinte processamento:

- Mestre gera uma matriz A 90x90 e escravos geram a matriz B 90x90 ao mesmo tempo;
- Mestre envia matriz A para escravos;
- Mestre e escravos calculam a quantidade de linhas que cada um irá trabalhar;
- Escravo 1 envia a matriz B para o mestre;
- Mestre envia a posição zero para escravo 1;
- Mestre envia a posição 30 para escravo 2;
- Escravo 1 realiza a multiplicação da linha 0 até a linha 29;
- Escravo 2 realiza a multiplicação da linha 30 até a linha 59;
- Mestre realiza a multiplicação da linha 60 até a linha 89;
- Escravo 1 envia da linha 0 a linha 29 da matriz C;
- Escravo 2 envia da linha 30 a linha 59 da matriz C;
- Mestre recebe as respectivas linhas e as junta a seu resultado.

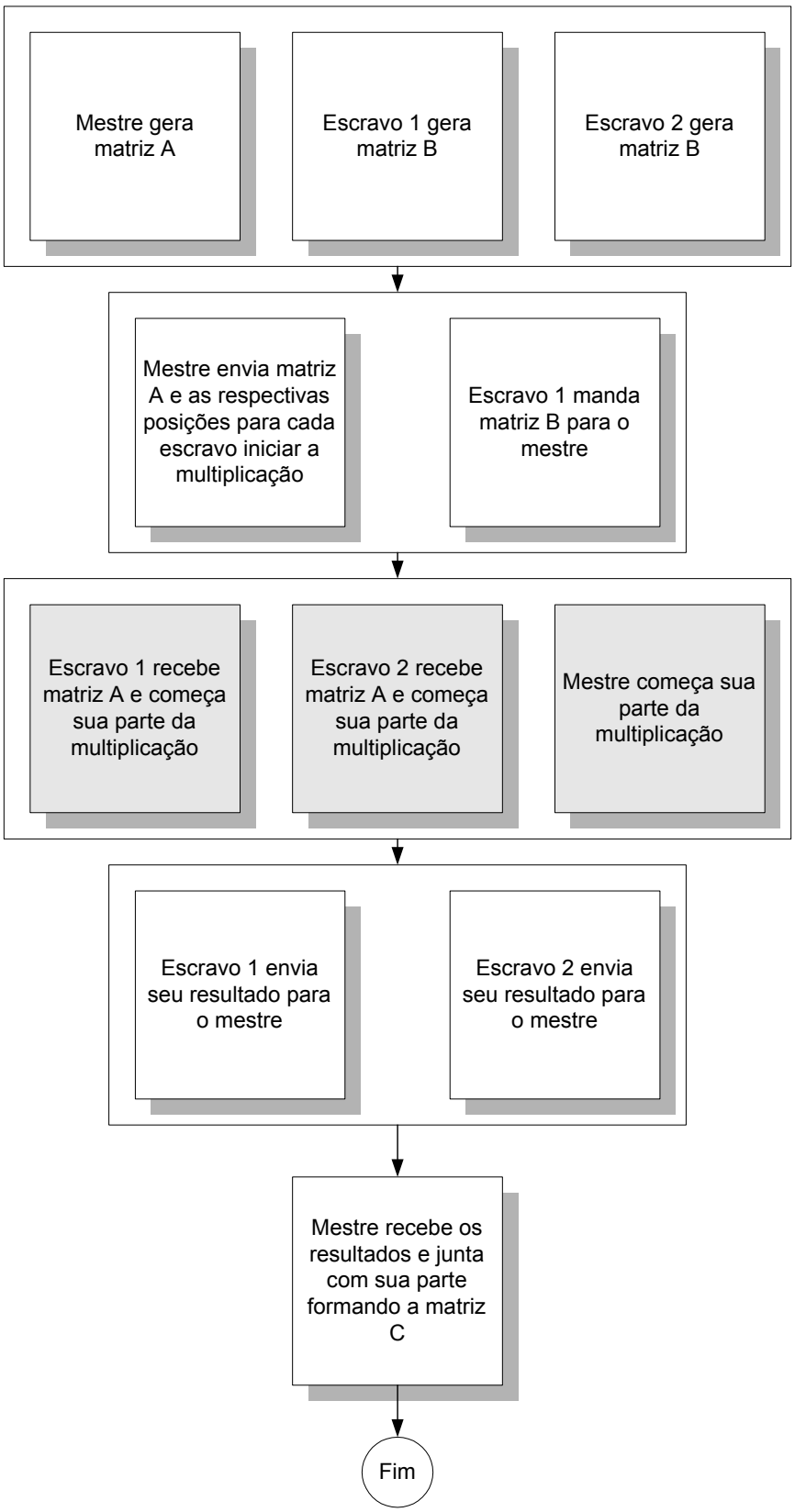

Figura 4.2 - Paralelização do algoritmo de multiplicação de matrizes.

Como aplicação **Disk-Bound** foi elaborada uma aplicação “sintética” que efetua escritas e leituras em disco (vide Apêndice A). Nessa aplicação foi possível variar o percentual do tempo total de acessos para leitura e para escrita no disco.

A aplicação **Network-Bound** foi representada por uma versão paralela do algoritmo de ordenação quicksort, desenvolvido em (Branco, 1999). Nessa versão paralela do quicksort, um vetor com **n** elementos é dividido em **p** processos escravos que ordenam em paralelo partes desse vetor (vide Apêndice A). Depois de ordenados, os subvetores são enviados para o processo mestre que os une novamente, gerando um único vetor ordenado. Em função de alguns fatores como: tamanho do vetor utilizado (306.000 elementos inteiros), rede de conexão utilizada (ethernet de 100 Mbits) e o próprio algoritmo de ordenação, o tempo utilizado para o envio de cada subvetor é maior que o tempo gasto pelos escravos para ordenar cada um desses subvetores, o que o torna uma aplicação mais voltada para comunicação do que para processamento.

Na avaliação feita com as aplicações Método dos Trapézios Compostos, da Multiplicação de Matrizes, do Quicksort Paralelo e das aplicações "sintéticas" para Memória e Disco, comparou-se não só o escalonamento feito pelo PVM e pelo ambiente de escalonamento AMIGO, mas o escalonamento feito pelo ambiente AMIGO fazendo uso de diversos índices de carga.

Essas execuções estabeleceram nas aplicações o escalonamento a ser realizado (dinamicamente, portanto), com o intuito de verificar a qualidade do escalonamento sugerido pelo AMIGO, mediante a apresentação dos índices de carga.

4.4 Análise Estatística Considerada

O objetivo de realizar uma análise estatística neste trabalho é verificar se as diferenças de desempenho das aplicações, considerando os diversos índices de carga descritos na literatura, são estatisticamente significativas.

Desse modo, para os experimentos apresentados neste capítulo, utiliza-se técnica estatística de Teste de Hipóteses (Francisco, 1995; Achcar & Rodrigues, 1995).

Essa técnica é desenvolvida a partir de duas hipóteses, H_0 ou hipótese de nulidade e H_1 ou hipótese alternativa, formuladas sobre valores que deseja-se comparar.

O primeiro passo para a utilização do Teste de Hipóteses consiste na definição dessas duas hipóteses que, após realizar o cálculo dos testes, uma delas será

aceita e a outra rejeitada. A hipótese H_0 ou hipótese de nulidade é geralmente formulada procurando-se "discordar" dos valores obtidos com o experimento. A hipótese H_1 ou hipótese alternativa é aquela que geralmente "concorda" com os valores obtidos no experimento quando comparados "*in-loco*". Assim, a hipótese H_0 é a negação da hipótese H_1.

A hipótese H_0 utilizada nas comparações pode ser exemplificada da seguinte maneira: “o desempenho da aplicação usando o AMIGO é melhor que o desempenho da mesma aplicação usando somente o PVM”. Essa hipótese H_0 é utilizada quando os valores obtidos com o AMIGO são menores que os obtidos com o PVM.

Para se realizar a análise estatística dos tempos coletados fazem-se, portanto, os seguintes testes de hipóteses:

- para amostras onde o tempo do AMIGO < tempo do PVM:

H_0: μAMIGO $\geq$ μPVM

H_1: μAMIGO < μPVM

- para amostras onde o tempo do AMIGO > tempo do PVM:

H_0: μAMIGO $\leq$ μPVM

H_1: μAMIGO > μPVM

onde: μAMIGO e μPVM são as médias dos tempos de execução com o ambiente de escalonamento AMIGO fazendo uso dos diversos índices de carga e com o PVM fazendo uso da política round robin respectivamente.

Considerando-se que os 30 tempos obtidos para cada média comparada possuem uma distribuição normal, a estatística dos testes de hipóteses acima é dada por:

$$Z = \frac{\overline{X}_{AMIGO} - \overline{X}_{PVM}}{\sqrt{\frac{S^2_{AMIGO}}{n_{AMIGO}} + \frac{S^2_{PVM}}{n_{PVM}}}}$$

Equação 4.1

onde: $\overline{X}_{AMIGO}$ e $\overline{X}_{PVM}$ são as médias amostrais dos tempos obtidos; S^2_{AMIGO} e S^2_{PVM} representam o desvio padrão amostral; e n_{AMIGO} e n_{PVM} representam o tamanho das amostras (neste caso 30).

Para um nível de significância (α) igual a 0.01 (probabilidade de estar correto 99% das vezes que a análise estatística for feita), rejeita-se a hipótese nulidade quando Z ultrapassar o limite fornecido por $Z_{0,01}$, o qual é 2.57. O valor de $Z_{0,01}$ = 2.57 é fornecido pela Tabela de Distribuição Normalizada (Achcar & Rodrigues 1995). Rejeita-se a hipótese nulidade H_0 caso $Z \leq -2.57$, ou então, $Z \geq 2.57$.

4.5 Avaliação de Heterogeneidade

O sistema de computação utilizado para os testes é composto por um conjunto de máquinas heterogêneas formando uma arquitetura MIMD, que será aqui referida como cluster "C".

Toda informação sobre o desempenho de cada máquina proporciona conhecimentos essenciais para uma melhor tomada de decisão. Uma dessas informações é a representação da heterogeneidade desse sistema.

A preocupação em contemplar a heterogeneidade existe, uma vez que se almeja efetuar um balanceamento de cargas mais “justo”, pois não se deve distribuir a mesma carga para todas as máquinas do sistema se existem diferenças entre elas.

Uma avaliação de desempenho foi então realizada para classificar as máquinas de acordo com suas capacidades. Essa avaliação dos recursos foi feita através de *benchmarks*.

Dentro do próprio ambiente de escalonamento AMIGO, utilizado para executar os testes, existe um arquivo de apoio à tomada de decisão onde, para cada tipo específico de aplicação são adotados valores fornecidos pelo *benchmark* correspondente e caracterizado mediante análise dos recursos mais utilizados pela aplicação. A Figura 4.3 apresenta a estrutura do arquivo de configuração dos *benchmarks* presentes no ambiente AMIGO.

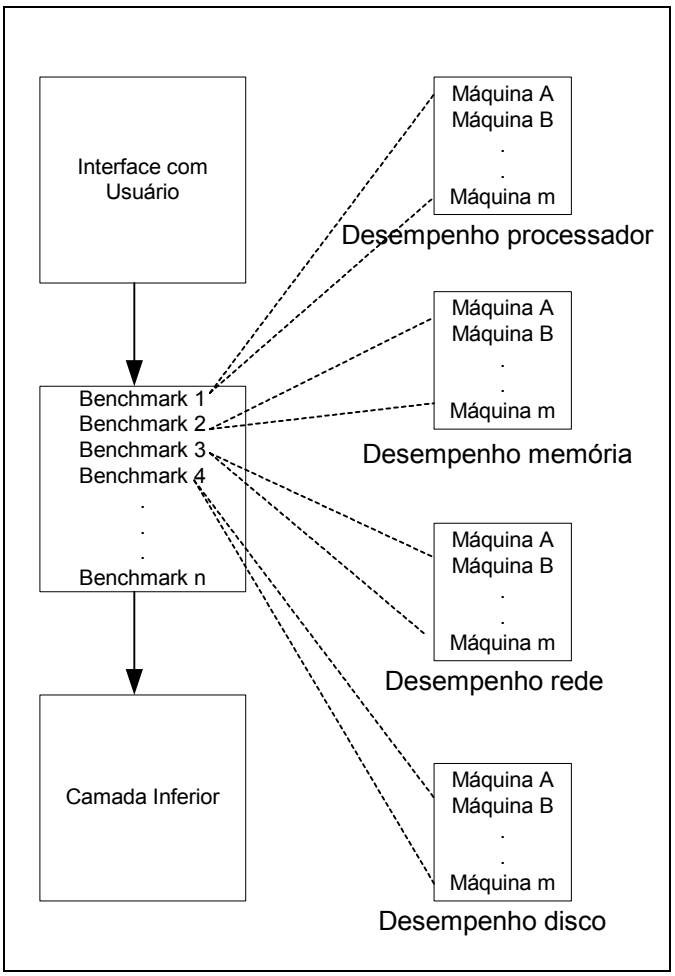

Figura 4.3 – Arquivo de configuração dos *benchmarks* do ambiente AMIGO

Uma vez que esse valor obtido pelo *benchmark* é normalizado, pode-se comparar as máquinas do sistema de igual para igual, sem a preocupação de estar sendo "injusto" com determinadas máquinas e alocando para essas, cargas maiores do que elas possam suportar. Assim, os *benchmarks* foram executados em cada uma das máquinas que compõem o sistema computacional utilizado para os testes apresentados nesta tese.

Os *benchmarks* para os índices foram estipulados como segue:

- a potência computacional de cada uma da máquina $m_i \in C$ é s_i. A quantidade de recurso disponível na máquina m_i é h_i, e a_i é a medida de velocidade de acesso ao recurso. A medida final é dada então com relação a dois parâmetros (a_i e h_i). Nos estudos aqui apresentados os pesos dados aos dois parâmetros foram iguais;

$$s_i = \frac{h_i}{\max\{h_i\}} wh_i + \frac{a_i}{\max\{a_i\}} wa_i \qquad \textbf{Equação 4.2}$$

Tomando por base a avaliação feita pelo *benchmark* normalizado, foi possível colocar as máquinas segundo uma classificação, com valores variando em 0 e 1 e sendo a melhor máquina do sistema classificada com o melhor valor (1), e as outras máquinas com valores proporcionalmente menores.

4.6 Análise dos Resultados Obtidos

A métrica utilizada nas avaliações de desempenho é a redução do tempo de execução da aplicação e a unidade de medida utilizada é o segundo. Cada valor expresso nos gráficos e tabelas desta seção representa a média de 30 execuções.

Para todas as avaliações feitas, a ordem de inclusão dos processadores na máquina paralela virtual foi do processador com maior potência computacional para o processador com menor potência.

Considerando que as aplicações executadas possuem organização mestre x escravo, os processos “mestre” de cada aplicação sempre foram executados no processador de maior potência, enquanto que os escravos (nove para cada aplicação) distribuídos pelos processadores segundo as políticas estabelecidas pelo PVM e pelo ambiente de escalonamento AMIGO.

4.6.1 Desempenho dos Diversos Tipos de Aplicações com o AMIGO e a Variação dos Índices de Carga

Esta seção apresenta a avaliação de alguns índices de carga largamente empregados em escalonadores, visando embasar a composição de um índice mais geral que será apresentado no capítulo 6.

Esta seção apresenta os gráficos e as conclusões obtidas a partir dos experimentos realizados. Os resultados apresentados nos gráficos correspondem a valores médios obtidos após 30 execuções de cada uma das aplicações, sob cada uma das formas de escalonamento e cada um dos diferentes índices de carga considerados.

4.6.1.1 Índices de Carga de Memória

Dentre os índices de carga de memória, três foram selecionados por apresentarem melhores resultados segundo a literatura (Kaplan & Nelson, 1994; Bricker et al, 1991; Duke et al, 1994; Suplick, 1994; LSF, 2001), além de serem bastante utilizados: (1) memória livre; (2) memória "propriamente" livre; (3) memória "propriamente" livre + swap.

(1) por memória livre têm-se (do arquivo /proc/meminfo) os valores de **memória livre** (*memfree*) fornecidos pelo próprio sistema operacional. Entretanto, observa-se que essa memória dita livre não é a memória que realmente está livre, mas sim parte dela, fazendo com que os valores dos índices não espelhem a realidade de cada máquina. Esse fato pode ser melhor observado quando da apresentação dos resultados dos testes realizados com esses índices, em comparação com os demais adotados;

(2) desse modo, um novo índice foi adotado, uma vez que se pode observar que a quantidade de memória livre (sugerida como índice de carga) não pode ser simplesmente obtida do sistema através da variável *memfree*, uma vez que não espelha a realidade (de acordo com testes realizados neste trabalho). A **memória "propriamente" livre** é dada pela somatória das variáveis *memfree*, *Inact_dirty*, *Inact_clean*, *Inact_target* obtidas também a partir do arquivo /proc/meminfo. Essa memória "propriamente" livre é de fácil obtenção a partir do momento em que se sabe quais são os parâmetros ou quais são as variáveis que devem ser associadas a cada um dos diversos sistemas operacionais existentes;

(3) e por último, agregou-se à **memória "propriamente" livre** a quantidade de **swap** (quantidade de memória alocada para troca de páginas) disponível na máquina, também obtido a partir do arquivo /proc/meminfo.

Com o intuito de obter um índice de carga que seja significativo, e que reflita a carga atual da máquina em termos de memória, foi definida na política de informação a obtenção dos resultados em uma base regular (intervalos de 1 segundo) e uma média exponencial após as N medições, de acordo com a equação 4.3, onde v_i é a média exponencial, l_i é a média da carga corrente, $p.v_i$ é a média do período anterior (LSF, 2001; Suplick, 1994).

$$v_i = (l_i - pv_i)\frac{2}{1+N} + pv_i \qquad \textbf{Equação 4.3}$$

A avaliação de um índice de carga depende também do estabelecimento de um limite que define um nível, a partir do qual uma máquina é considerada sobrecarregada.

A política utilizada permite a definição de um limite para cada máquina. Esse valor é também utilizado para tornar o índice de carga computável como parte de uma composição do índice formado por mais de uma media de carga. Isso faz com que todos os índices se tornem compatíveis em uma determinada escala.

O índice normalizado é dado por:

$$n_i = \frac{v_i}{th_i} \qquad \textbf{Equação 4.4}$$

onde th_i é o limite definido e v_i é a média exponencial. O valor do índice normalizado (n_i) será maior que 1 se a máquina estiver livre ou moderada, e terá seu valor decrementado à medida que se torna sobrecarregada.

Para cada índice de carga específico, um limite deverá ser definido. Dessa forma, dividindo-se cada valor de carga pelo limite correspondente, torna-se possível a obtenção, por parte da política de informação, da potência computacional de cada dispositivo (Ferrari & Zhou, 1987).

A heurística criada para se efetuar as decisões de escalonamento é baseada na estimativa obtida da avaliação de desempenho da máquina (*benchmark*) e do índice de carga (l_i). O índice de carga é multiplicado pelo valor do *benchmark* ($s_i - Eq.4.2$), então a máquina tem seu potencial diminuído ($n_i - Eq.4.4$). O resultado representa a capacidade disponível ($c_i - Eq.4.5$) de cada máquina individualizada na plataforma.

$$c_i = n_i . s_i \qquad \textbf{Equação 4.5}$$

Dessa forma, somente um valor é transmitido (c_i), uma vez que esse valor encapsula conhecimento suficiente para uma decisão de alta qualidade.

Essa carga é coletada periodicamente e colocada no cache. Essa informação é então disseminada quando o último índice de carga calculado sugere que o estado da máquina local foi alterado em um certo grau. A freqüência de atualização pode ser feita conforme a necessidade. Assim, quanto mais freqüente a carga for atualizada mais precisa será a decisão, entretanto, maior será a sobrecarga de comunicação.

Foi assumido nestes testes que a quantidade de memória livre disponível é o total de memória existente na máquina excetuando-se 20MB, quantidade essa normalmente utilizada pelo sistema operacional. O limite foi então definido como 20% do total de memória principal.

Os gráficos apresentados nas figuras que seguem apresentam os valores obtidos e as tabelas apresentam os resultados dos testes de hipótese realizados.Os tempos de execução são dados em segundos.

No primeiro experimento utilizou-se como índice de carga a quantidade de memória livre, enquanto que no segundo experimento utilizou-se a quantidade de memória propriamente livre e por fim a associação da memória propriamente livre em conjunto com a quantidade de memória para que se pudesse efetuar troca de páginas disponível.

De modo a organizar as Figuras e Tabelas de acordo com os objetivos considerados em cada experimento e, com isso, apresentar conclusões mais coesas, as Figuras 4.4 a 4.7 e as Tabelas 4.1 a 4.4 foram organizadas como pertencentes ao experimento 1, enquanto que as Figuras de 4.8 a 4.11 e as Tabelas de 4.5 a 4.8 relacionadas ao experimento 2 e as Figuras 4.12 a 4.15 e as Tabelas 4.9 a 4.12 ao experimento 3. A discussão sobre cada um dos experimentos executados encontra-se na tabela 4.14.

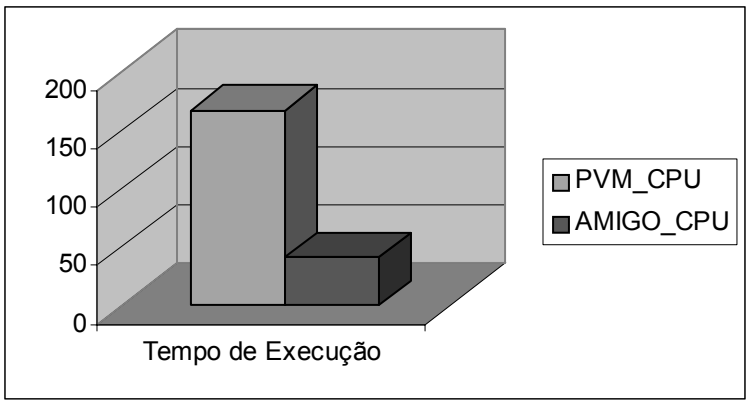

Figura 4.4 - Gráfico da execução de uma aplicação CPU-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória livre.

Tabela 4.1 - Valores calculados para o desempenho obtido com a execução de uma aplicação CPU-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	167,639	42,189
Desvio Padrão	0,486	0,057
Variância	0,236	0,003
Hipótese α=0,01		**-1404,499**

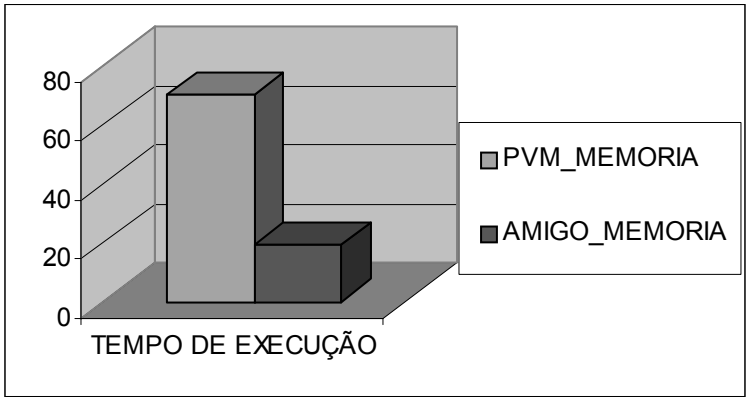

Figura 4.5 - Gráfico da execução de uma aplicação Memory-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória livre.

Tabela 4.2 - Valores calculados para o desempenho obtido com a execução de uma aplicação Memory-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	70,438	19,362
Desvio Padrão	4,658	2,002
Variância	21,694	4,009
Hipótese α= 0,01		**-55,179**

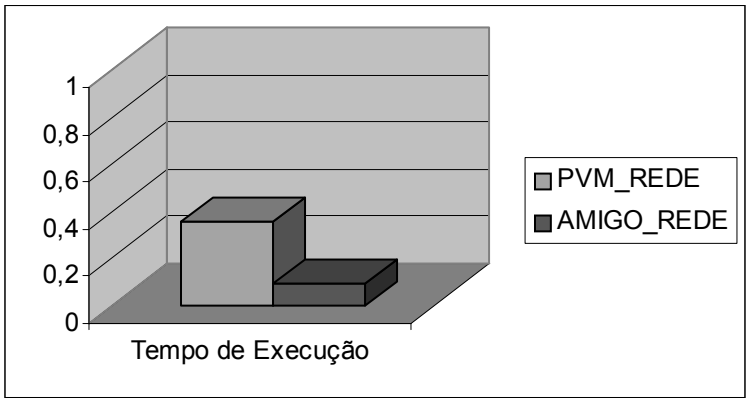

Figura 4.6 - Gráfico da execução de uma aplicação Network-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória livre.

Tabela 4.3 - Valores calculados para o desempenho obtido com a execução de uma aplicação Disk-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	38,807	39,628
Desvio Padrão	2,101	1,586
Variância	4,414	2,516
Hipótese α= 0,01		**1,708**

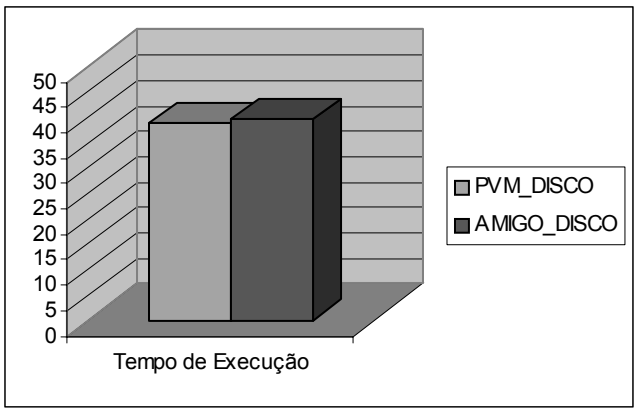

Figura 4.7 - Gráfico da execução de uma aplicação Disk-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória livre.

Tabela 4.4 - Valores calculados para o desempenho obtido com a execução de uma aplicação Network-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	0,355	0,089
Desvio Padrão	0,153	0,064
Variância	0,023	0,004
Hipótese α= 0,01		**-8,803**

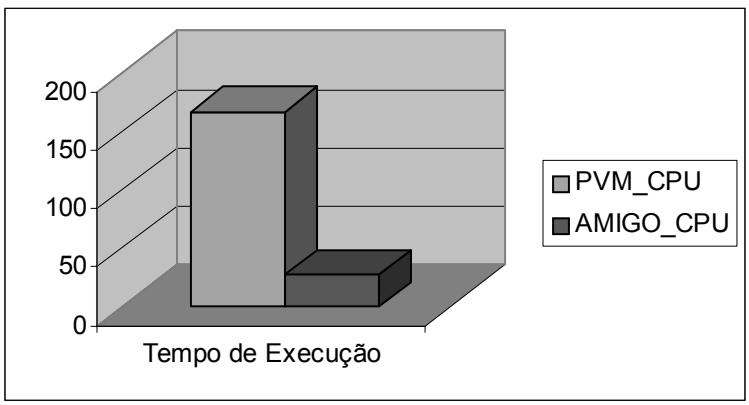

Figura 4.8 - Gráfico da execução de uma aplicação CPU-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória propriamente livre.

Tabela 4.5 - Valores calculados para o desempenho obtido com a execução de uma aplicação CPU-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	167,639	27,292
Desvio Padrão	0,486	0,052
Variância	0,236	0,003
Hipótese α= 0,01		**-1573,2944**

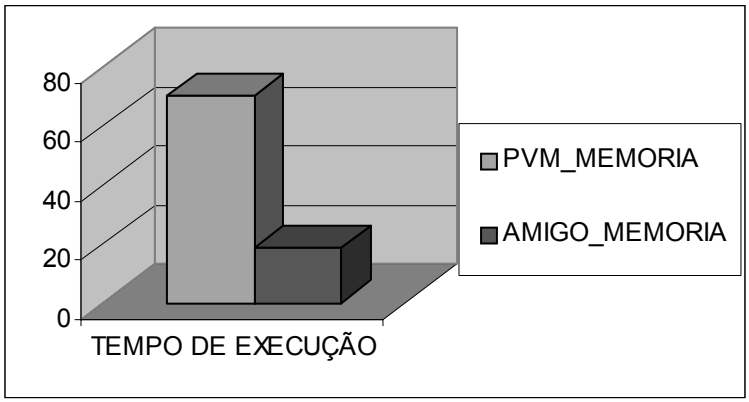

Figura 4.9 - Gráfico da execução de uma aplicação Memory-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória propriamente livre.

Tabela 4.6 - Valores calculados para o desempenho obtido com a execução de uma aplicação Memory-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	70,438	18,697
Desvio Padrão	4,658	1,889
Variância	21,694	3,160
Hipótese α= 0,01		**-56,384**

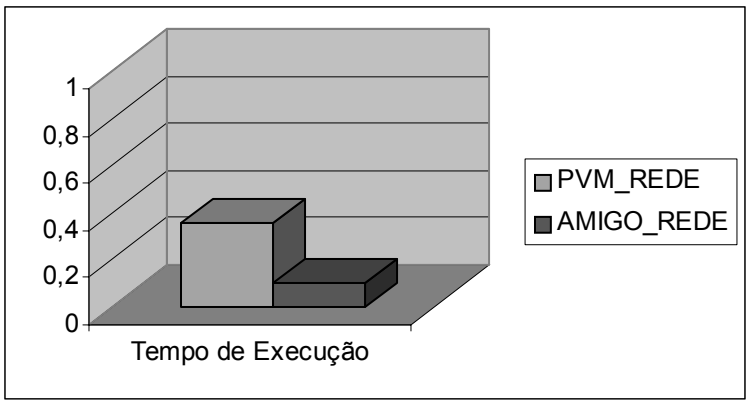

Figura 4.10 - Gráfico da execução de uma aplicação Network-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória propriamente livre.

Tabela 4.7 - Valores calculados para o desempenho obtido com a execução de uma aplicação Network-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	0,355	0,100
Desvio Padrão	0,153	0,106
Variância	0,023	0,011
Hipótese α= 0,01		**-7,517**

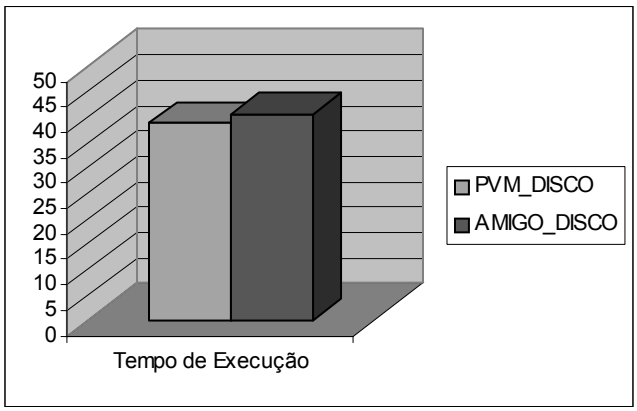

Figura 4.11 - Gráfico da execução de uma aplicação Disk-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória propriamente livre.

Tabela 4.8 - Valores calculados para o desempenho obtido com a execução de uma aplicação Disk-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	38,807	40,661
Desvio Padrão	2,101	5,961
Variância	4,414	35,537
Hipótese α= 0,01		**1,607**

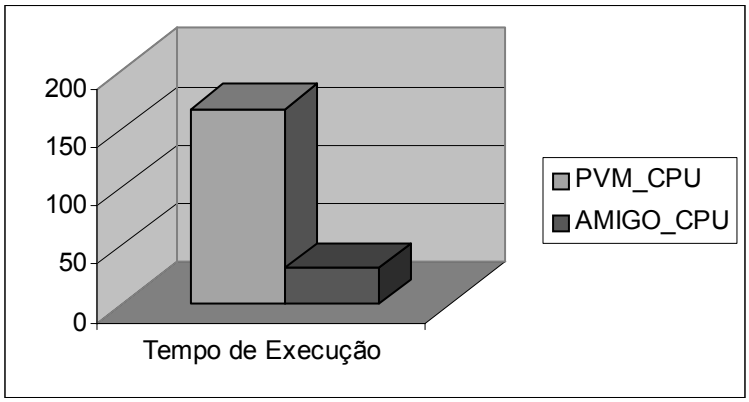

Figura 4.12- Gráfico da execução de uma aplicação CPU-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória propriamente livre associado ao swap.

Tabela 4.9- Valores calculados para o desempenho obtido com a execução de uma aplicação CPU-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	167,639	30,949
Desvio Padrão	0,486	0,015
Variância	0,236	0,0002
Hipótese α= 0,01		**-1540,232**

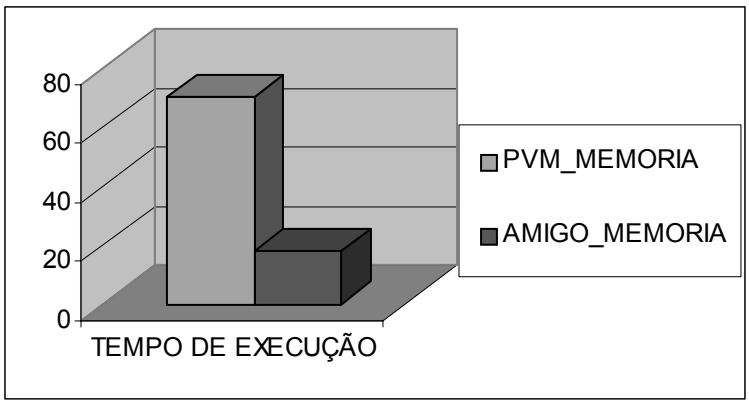

Figura 4.13- Gráfico da execução de uma aplicação Memory-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória propriamente livre associado ao swap.

Tabela 4.10- Valores calculados para o desempenho obtido com a execução de uma aplicação Memory-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	70,438	17,939
Desvio Padrão	4,658	1,460
Variância	21,694	1,779
Hipótese α= 0,01		**-58,911**

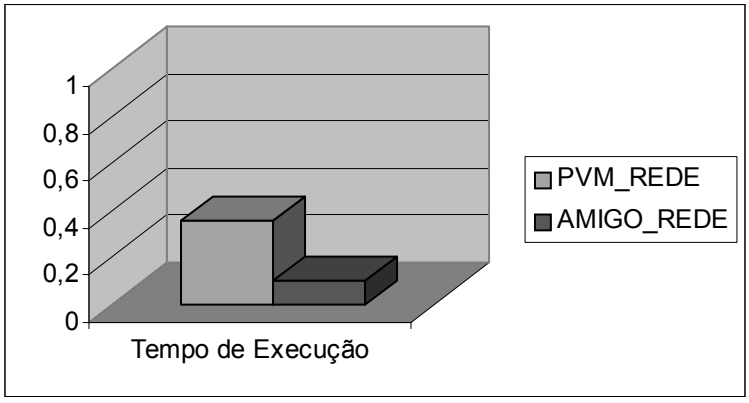

Figura 4.14- Gráfico da execução de uma aplicação Network-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória propriamente livre associado ao swap.

Tabela 4.11- Valores calculados para o desempenho obtido com a execução de uma aplicação Network-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	0,355	0,098
Desvio Padrão	0,157	0,101
Variância	0,023	0,010
Hipótese α= 0.01		**-7,690**

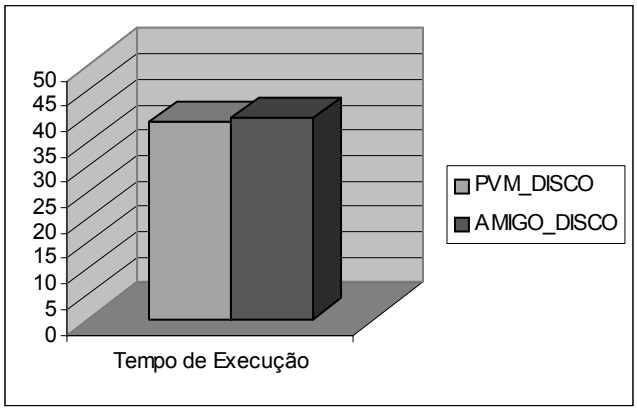

Figura 4.15- Gráfico da execução de uma aplicação Disk-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando um índice de carga de memória propriamente livre associado ao swap.

Tabela 4.12- Valores calculados para o desempenho obtido com a execução de uma aplicação Disk-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	38,807	39,605
Desvio Padrão	2,101	1,420
Variância	4,414	2,018
Hipótese α= 0,01		**1,724**

Tabela 4.13- Teste de hipótese dos valores obtidos por classe de aplicações executadas com relação aos diferentes índices de carga de memória implementados.

Hipótese α= 0,01	Índices	Classe de Aplicação
-1,323	Memória propriamente livre - Memória livre	Memória
-3,146	Memória propriamente livre e swap - Memória livre	
-1,740	Memória propriamente livre e swap - Memória propriamente livre	

Hipótese α= 0,01	Índices	Classe de Aplicação
-1055,659	Memória propriamente livre - Memória livre	CPU
-1037,632	Memória propriamente livre e swap - Memória livre	
371,469	Memória propriamente livre e swap - Memória propriamente livre	

Hipótese α= 0,01	Índices	Classe de Aplicação
0,918	Memória propriamente livre - Memória livre	DISCO
-0,058	Memória propriamente livre e swap - Memória livre	
-0,944	Memória propriamente livre e swap - Memória propriamente livre	

Hipótese α= 0,01	Índices	Classe de Aplicação
0,482	Memória propriamente livre - Memória livre	REDE
0,409	Memória propriamente livre e swap - Memória livre	
-0,073	Memória propriamente livre e swap - Memória propriamente livre	

A tabela 4.13 apresenta os valores dos testes de hipótese quando comparados os diferentes índices de carga de memória analisados. Mediante os valores apresentados, pode-se notar que para as classes de aplicações de rede e disco é indiferente o uso de um dos três índices como sendo índice de carga. Entretanto, para aplicações de memória e CPU o uso do índice de memória com valor mais significativo em questão de melhora de desempenho é o índice que considera a memória propriamente livre associada ao swap.

Os resultados apresentados nesses experimentos confirmam a idéia de que sob determinadas condições, utilizar técnicas apropriadas de escalonamento (diferentes do escalonamento round-robin) pode não surtir efeitos positivos no desempenho final da aplicação. Os valores obtidos nesses experimentos foram fortemente influenciados pelas características das aplicações (ambas utilizaram todas as máquinas disponíveis na máquina paralela) e pelo conhecimento prévio que o desenvolvedor possuía sobre o desempenho das máquinas utilizadas, possibilitando a escolha adequada do índice de carga utilizado.

Em situações onde nem todas as máquinas são utilizadas e o desenvolvedor não possui conhecimento sobre a configuração dessas máquinas, é possível que a utilização de ambientes de escalonamento e técnicas mais aprimoradas possam trazer melhorias de desempenho.

Apesar das diferenças entre as execuções, pode-se constatar através de teste de hipóteses que apenas os resultados obtidos com as aplicações Memory-Bound, CPU-Bound e Network-Bound eram realmente significativos, adotando-se 0,01 como nível de significância. Além da significância de diferença entre os resultados obtidos a partir das características específicas das aplicações, pode-se observar a significância entre os diferentes índices de memória utilizados. Pode-se observar que os índices que obtiveram nível de significância foram a memória propriamente livre e a associação dessa ao swap.

Por esses experimentos pode-se concluir que para aplicações de memória os índices de memória mais significativos e que devem ser utilizados para compor o índice de desempenho proposto neste trabalho são os índices de memória propriamente livre e o índice que associa o *swap*.

O melhor desempenho obtido pelo AMIGO ocorre na situação onde há pior desempenho com o round robin. Nesse caso, o AMIGO escalona os processos apenas para as melhores máquinas uma vez que elas estão ociosas, enquanto o round robin continua distribuindo processos para todas as máquinas de maneira igualitária.

4.6.1.2 Índices de Carga de CPU

Dentre os índices de carga de CPU, três foram selecionados por apresentarem melhores resultados segundo a literatura (Kaplan & Nelson, 1994; Bricker et al, 1991; Duke et al, 1994; Suplick, 1994; Wolffe, Hosseini & Vairavan, 1997; LSF, 2001) e por serem bastante utilizados: (1) porcentagem de CPU livre; (2) número de processos prontos na fila de CPU e (3) capacidade de carga.

(1) para obter a porcentagem de CPU livre utilizaram-se os valores obtidos a partir do programa *top* usado pelo Linux para fornecer dados estatísticos sobre o estado da máquina local;

(2) a utilização do número de processos prontos na fila de CPU deu-se pelo fato de ser este um dos índices mais considerados na literatura. Esse índice é obtido, de modo semelhante ao anterior, a partir da função *getindexload()* implementada a partir de alterações feitas no código fonte do

programa *top* (Maxwells, 2000; Ferreira, 2003).O valor obtido para esse índice de carga foi o tamanho médio da fila de processos na CPU nos últimos 20 (vinte) segundos, média esta calculada através da técnica da "janela deslizante" (Araújo et al, 1999);

(3) e a capacidade de carga, índice proposto por (Wolffe, Hosseini & Vairavan, 1997) que leva em conta a utilização de CPU e a velocidade relativa da CPU.

Assim como para os índices de memória, com o intuito de obter um índice de carga que seja significativo e que reflita a carga atual da máquina em termos de CPU, foi definida na política de informação a obtenção dos resultados em uma base regular (intervalos de 1 segundo) e uma média exponencial após as N medições (Suplick, 1994; LSF, 2001).

A utilização dos *benchmarks* e a definição dos limites para os índices de CPU são análogos aos implementados para os índices de memória.

Os gráficos apresentados nas figuras que seguem apresentam os valores obtidos e as tabelas apresentam os resultados dos testes de hipótese realizados.

No primeiro experimento utilizou-se como índice de carga a porcentagem de CPU livre, no segundo experimento foi utilizado o número de processos prontos na fila de CPU, e no terceiro e último exemplo foi utilizada a capacidade de carga.

As Figuras e Tabelas são organizadas de acordo com os objetivos considerados em cada experimento e, com isso, apresentam conclusões mais coesas. As Figuras 4.16 a 4.19 e as Tabelas 4.14 a 4.17 foram organizadas como pertencentes ao experimento 1. As Figuras 4.20 a 4.23 e as Tabelas 4.18 a 4.21 foram organizadas como pertencentes ao segundo experimento. As Figuras 4.24 a 4.27 e as Tabelas 4.22 a 4.25 como pertencentes ao terceiro experimento.

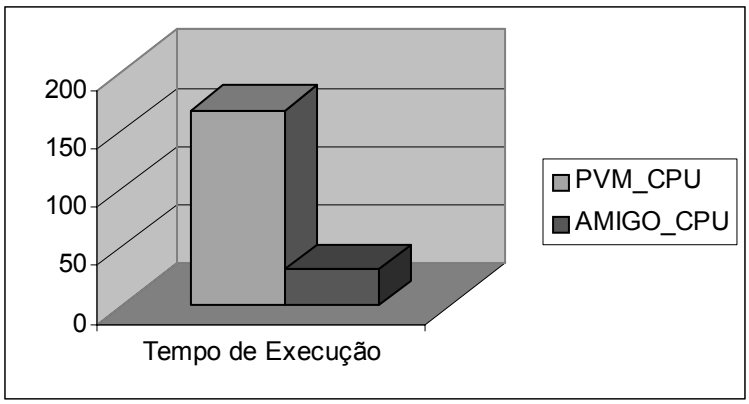

Figura 4.16 - Gráfico da execução de uma aplicação CPU-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga a porcentagem de CPU livre.

Tabela 4.14 - Valores calculados para o desempenho obtido com a execução de uma aplicação CPU-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	167,639	31,604
Desvio Padrão	0,486	4,510
Variância	0,236	20,344
Hipótese α= 0,01		**-164,244**

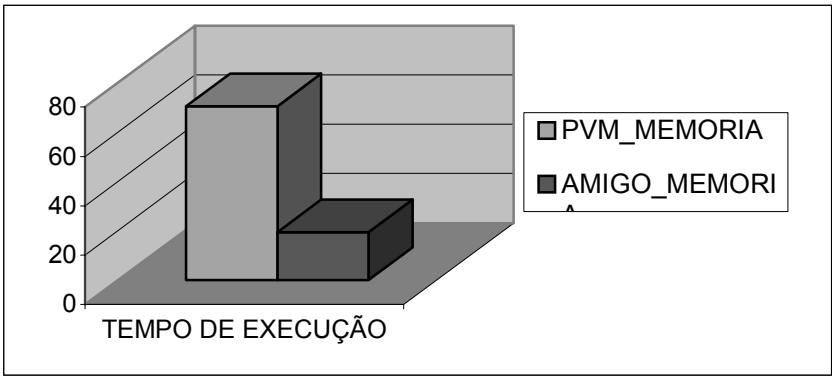

Figura 4.17 - Gráfico da execução de uma aplicação Memory-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga a porcentagem de CPU livre.

Tabela 4.15 - Valores calculados para o desempenho obtido com a execução de uma aplicação Memory-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	70,438	19,499
Desvio Padrão	4,658	1,857
Variância	21,694	3,449
Hipótese α= 0,01		**-55,641**

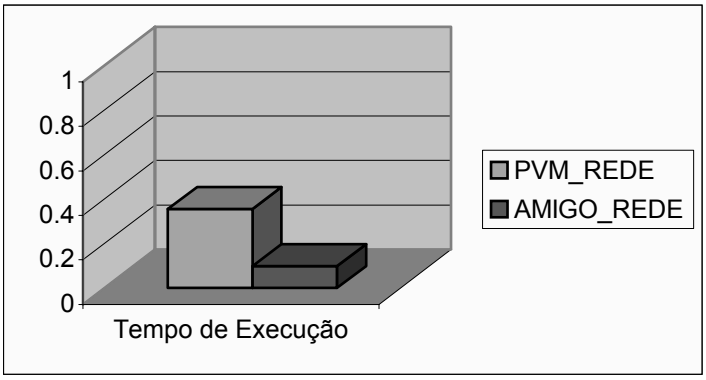

Figura 4.18 - Gráfico da execução de uma aplicação Network-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga a porcentagem de CPU livre.

Tabela 4.16 - Valores calculados para o desempenho obtido com a execução de uma aplicação Network-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	0,355	0,097
Desvio Padrão	0,153	0,110
Variância	0,023	0,012
Hipótese α= 0,01		**-7,511**

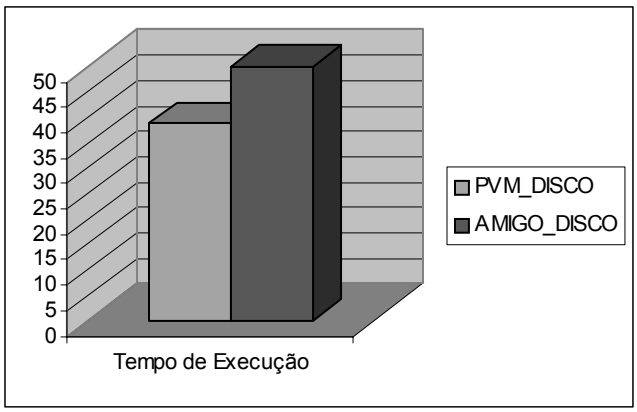

Figura 4.19 - Gráfico da execução de uma aplicação Disk-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga a porcentagem de CPU livre.

Tabela 4.17 - Valores calculados para o desempenho obtido com a execução de uma aplicação Disk-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	38,807	58,208
Desvio Padrão	2,101	3,769
Variância	4,414	14,204
Hipótese α= 0,01		**24,628**

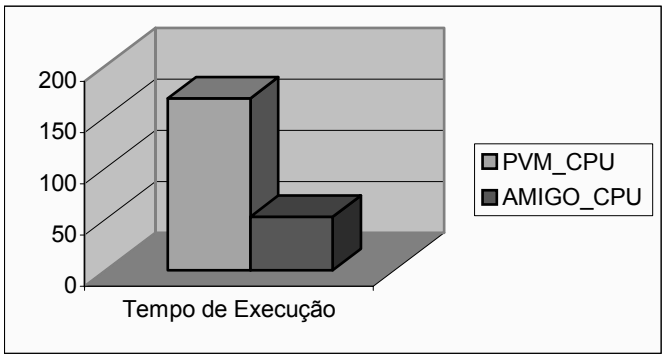

Figura 4.20 - Gráfico da execução de uma aplicação CPU-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de processos prontos na fila de CPU.

Tabela 4.18 - Valores calculados para o desempenho obtido com a execução de uma aplicação CPU-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	167,639	51,904
Desvio Padrão	0,486	2,851
Variância	0,236	8,131
Hipótese α= 0,01		**-219,152**

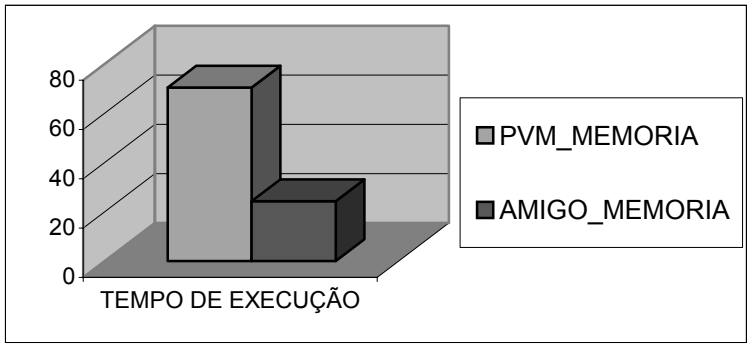

Figura 4.21 - Gráfico da execução de uma aplicação Memory-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de processos prontos na fila de CPU.

Tabela 4.19 - Valores calculados para o desempenho obtido com a execução de uma aplicação Memory-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	70,438	24,299
Desvio Padrão	4,658	1,191
Variância	21,694	1,419
Hipótese α= 0,01		**-52,565**

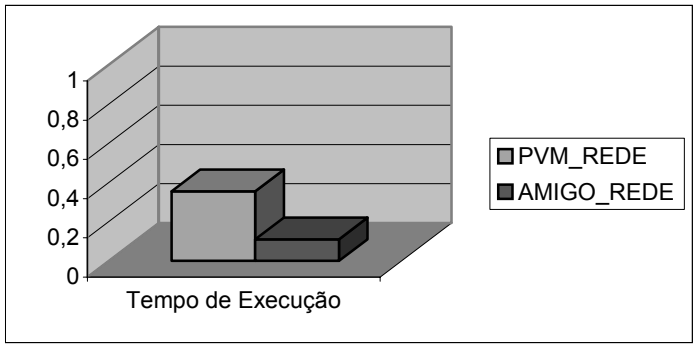

Figura 4.22 - Gráfico da execução de uma aplicação Network-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de processos prontos na fila de CPU.

Tabela 4.20 - Valores calculados para o desempenho obtido com a execução de uma aplicação Network-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	0,355	0,109
Desvio Padrão	0,153	0,108
Variância	0,023	0,012
Hipótese α= 0,01		**-7,203**

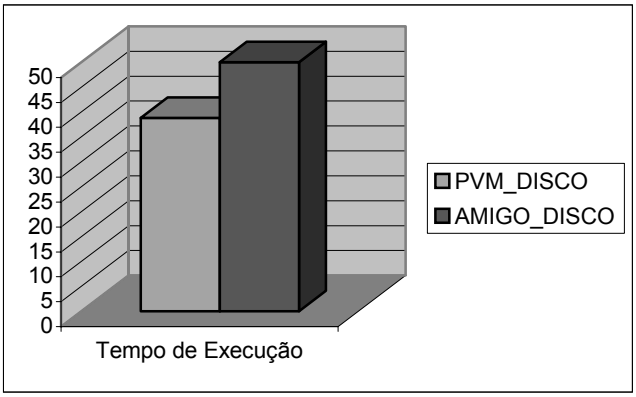

Figura 4.23 - Gráfico da execução de uma aplicação Disk-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de processos prontos na fila de CPU.

Tabela 4.21 - Valores calculados para o desempenho obtido com a execução de uma aplicação Disk-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	38,807	62,371
Desvio Padrão	2,101	4,446
Variância	4,414	19,769
Hipótese α= 0,01		**26,245**

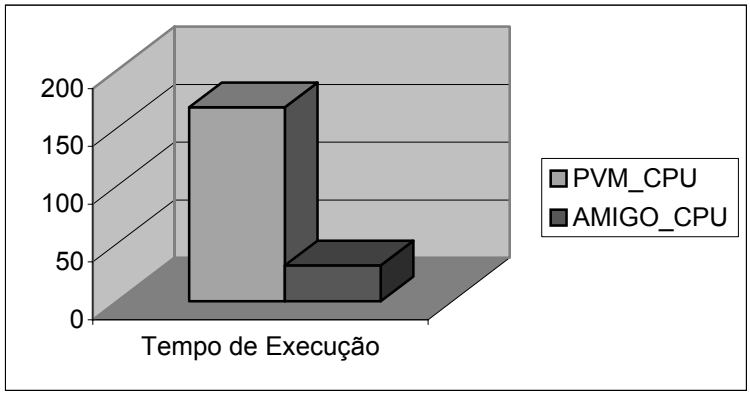

Figura 4.24 - Gráfico da execução de uma aplicação CPU-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga a capacidade de carga.

Tabela 4.22 - Valores calculados para o desempenho obtido com a execução de uma aplicação CPU-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	167,639	30,949
Desvio Padrão	0,486	0,015
Variância	0,236	0,0002
Hipótese α= 0,01		**-1540,232**

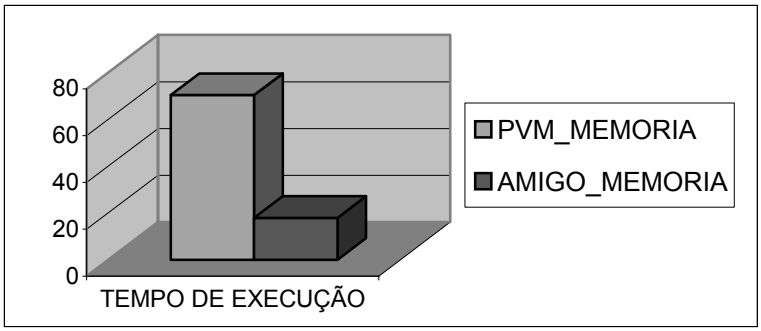

Figura 4.25 - Gráfico da execução de uma aplicação Memory-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga a capacidade de carga.

Tabela 4.23 - Valores calculados para o desempenho obtido com a execução de uma aplicação Memory-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	70,438	17,939
Desvio Padrão	4,658	1,460
Variância	21,694	1,779
Hipótese α= 0,01		**-58,911**

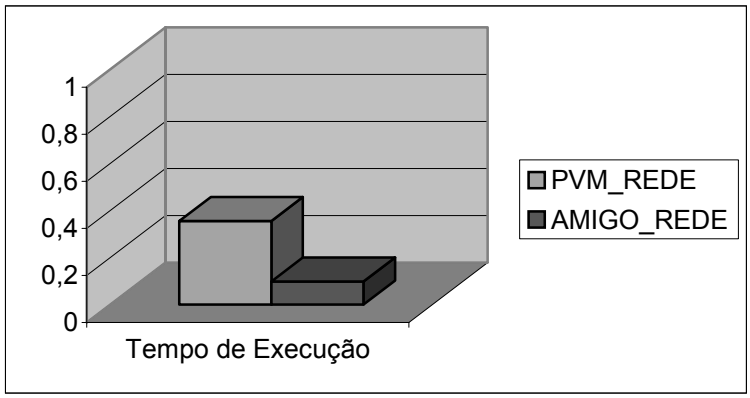

Figura 4.26 - Gráfico da execução de uma aplicação Network-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga a capacidade de carga.

Tabela 4.24 - Valores calculados para o desempenho obtido com a execução de uma aplicação Network-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	0,355	0,098
Desvio Padrão	0,153	0,012
Variância	0,023	0,010
Hipótese α= 0,01		**-7,690**

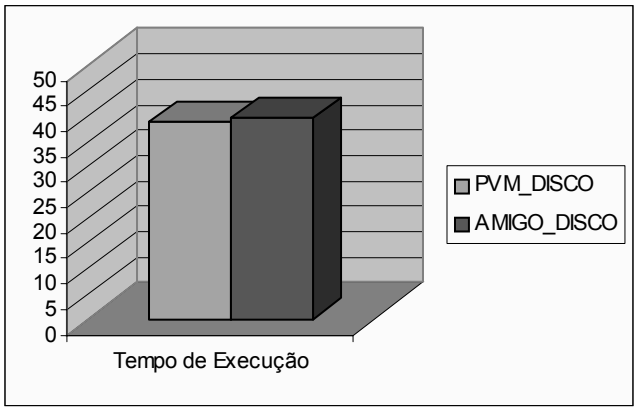

Figura 4.27 - Gráfico da execução de uma aplicação Disk-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga a capacidade de carga.

Tabela 4.25 - Valores calculados para o desempenho obtido com a execução de uma aplicação Disk-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	38,806	39,605
Desvio Padrão	2,101	1,420
Variância	4,414	2,018
Hipótese α= 0,01		**1,724**

Tabela 4.26- Teste de hipótese dos valores obtidos por classe de aplicações executadas com relação aos diferentes índices de carga de CPU implementados.

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
11,914	Processos prontos na fila - % CPU livre	Memória
-3,618	Capacidade de Carga - % CPU livre	
-18,488	Capacidade de Carga – Processos prontos na fila	

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
0,000	Processos prontos na fila - % CPU livre	CPU
-0,000	Capacidade de Carga - % CPU livre	
-40,249	Capacidade de Carga – Processos prontos na fila	

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
3,911	Processos prontos na fila - % CPU livre	Disco
-25,298	Capacidade de Carga - % CPU livre	
-26,713	Capacidade de Carga – Processos prontos na fila	

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
0,044	Processos prontos na fila - % CPU livre	Rede
0,046	Capacidade de Carga - % CPU livre	
-0,579	Capacidade de Carga – Processos prontos na fila	

A tabela 4.26 apresenta os valores dos testes de hipótese quando comparados os diferentes índices de carga de CPU analisados.

Por esses experimentos pode-se concluir que para aplicações do tipo CPU-BOUND e MEMORY-BOUND os índices de CPU são significativos. Entretanto, essa conclusão pode ser errônea, uma vez que em sua grande maioria, as máquinas que possuem melhor CPU são as máquinas que possuem maior quantidade de memória.

Esses e outros experimentos levaram à definição de um modelo em redes de fila que permite a simulação dos mesmos índices de carga com variações na configuração do ambiente de teste. Este modelo será apresentado e discutido no capítulo 6.

4.6.1.3 Índices de Carga de Rede

Uma medida candidata para avaliar a carga na interface de rede de uma máquina é o número de pacotes que chegam e que saem. Assim, dentre os índices de rede, dois foram selecionados por apresentarem melhores resultados segundo a literatura (Kaplan & Nelson, 1994; Bricker et al, 1991; Duke et al, 1994; Suplick, 1994; LSF, 2001) e por serem bastante utilizados: (1) número de pacotes que entram; (2) número de pacotes que saem.

(1) o número de pacotes que entram em cada interface de rede é obtido através do comando *netstat* que permite que a rede seja monitorada;

(2) de modo análogo, o número de pacotes que saem é obtido através do comando *netstat* que permite que a rede seja supervisionada.

De modo análogo aos demais índices apresentados, foi definida na política de informação a obtenção dos resultados em uma base regular (intervalos de 1 segundo) e uma média exponencial após N medições.

A utilização dos *benchmarks* e a definição dos limites para os índices de rede são análogas à implementada para os índices de CPU.

Os gráficos apresentados nas figuras que seguem apresentam os valores obtidos, bem como as tabelas apresentam os resultados dos testes de hipótese realizados.

No primeiro experimento utilizou-se como índice de carga o número de leituras, e no segundo experimento o número de escritas.

As Figuras e Tabelas são organizadas de acordo com os objetivos considerados em cada experimento e, com isso, apresentam conclusões mais coesas. As Figuras 4.28 a 4.31 e as Tabelas 4.27 a 4.30 foram organizadas como pertencentes ao experimento 1. As Figuras 4.32 a 4.35 e as Tabelas 4.31 a 4.34 foram organizadas como pertencentes ao segundo experimento.

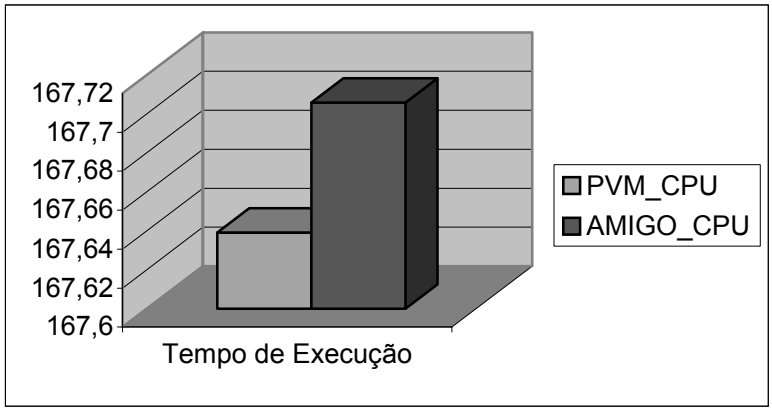

Figura 4.28 - Gráfico da execução de uma aplicação CPU-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de pacotes que entram.

Tabela 4.27 - Valores calculados para o desempenho obtido com a execução de uma aplicação CPU-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	167,639	167,706
Desvio Padrão	0,486	0.517
Variância	0,236	0,267
Hipótese α= 0,01		**0,515**

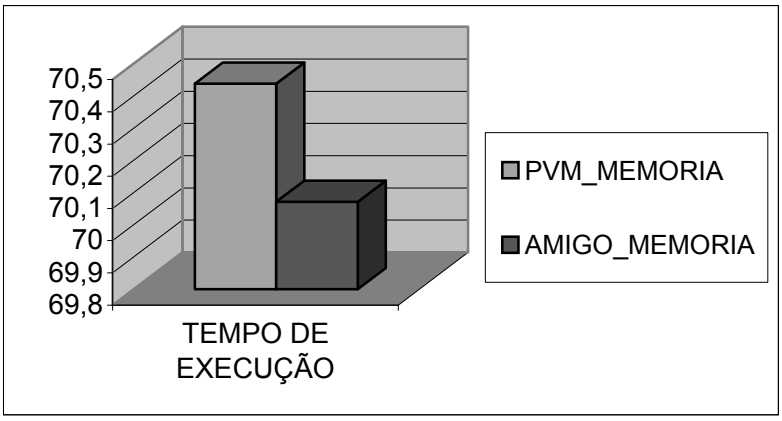

Figura 4.29 - Gráfico da execução de uma aplicação Memory-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de pacotes que entram.

Tabela 4.28 - Valores calculados para o desempenho obtido com a execução de uma aplicação Memory-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	70,438	70,072
Desvio Padrão	4,658	4,485
Variância	21,694	20,113
Hipótese α= 0,01		**-0,311**

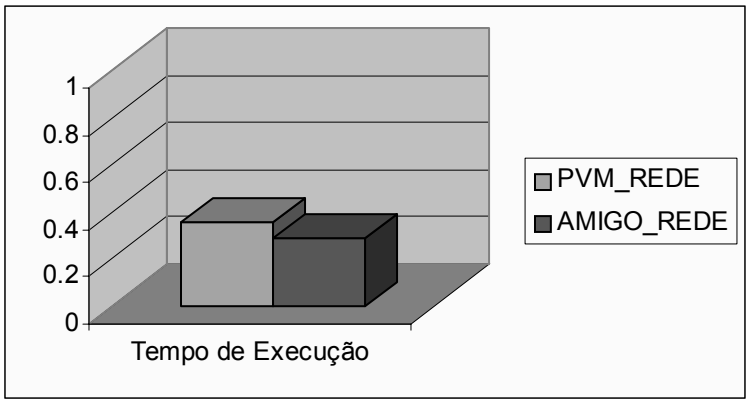

Figura 4.30 - Gráfico da execução de uma aplicação Network-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de pacotes que entram.

Tabela 4.29 - Valores calculados para o desempenho obtido com a execução de uma aplicação Network-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	0,355	0,288
Desvio Padrão	0,153	0,075
Variância	0,023	0,006
Hipótese α= 0,01		**-2,170**

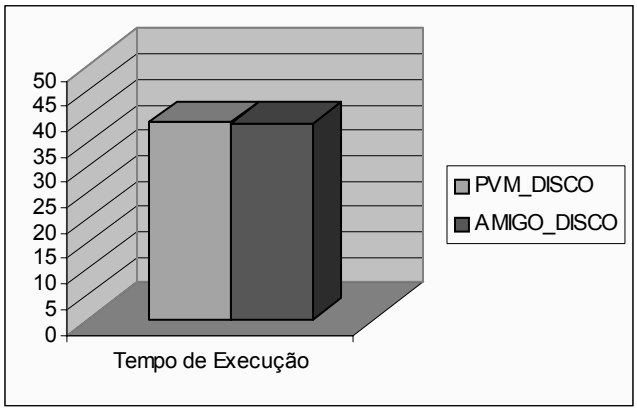

Figura 4.31 - Gráfico da execução de uma aplicação Disk-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de pacotes que entram.

Tabela 4.30 - Valores calculados para o desempenho obtido com a execução de uma aplicação Disk-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	38,807	38,540
Desvio Padrão	2,101	2,254
Variância	4,414	5,073
Hipótese α= 0,01		**-0,474**

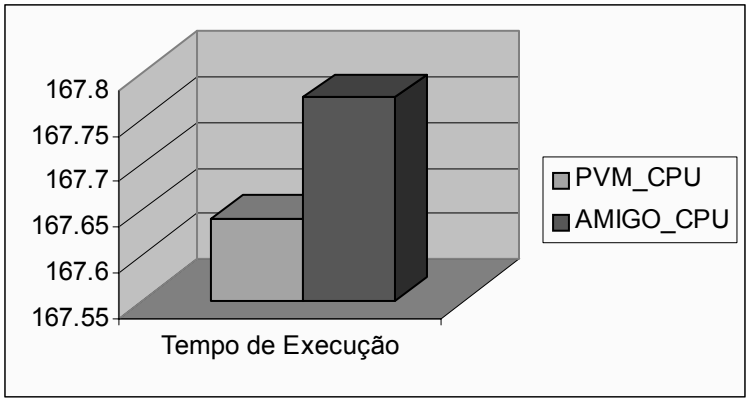

Figura 4.32 - Gráfico da execução de uma aplicação CPU-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de pacotes que saem.

Tabela 4.31 - Valores calculados para o desempenho obtido com a execução de uma aplicação CPU-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	167,639	167,773
Desvio Padrão	0,486	0,7249
Variância	0,236	0,525
Hipótese α= 0,01		**0,837**

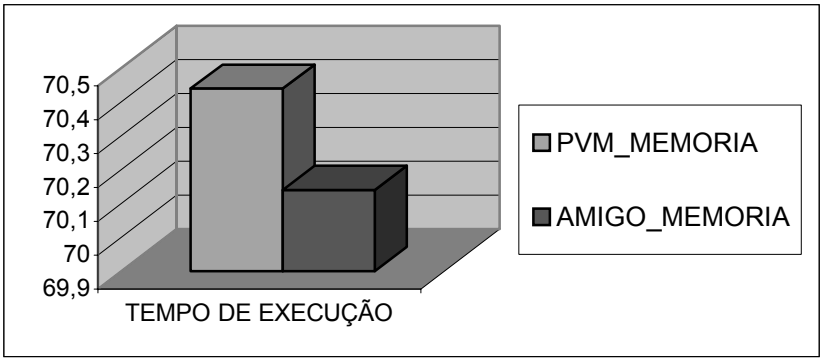

Figura 4.33 - Gráfico da execução de uma aplicação Memory-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de pacotes que saem.

Tabela 4.32 - Valores calculados para o desempenho obtido com a execução de uma aplicação Memory-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	70,438	70,139
Desvio Padrão	4,658	4,334
Variância	21,694	18,779
Hipótese α= 0,01		**-0,257**

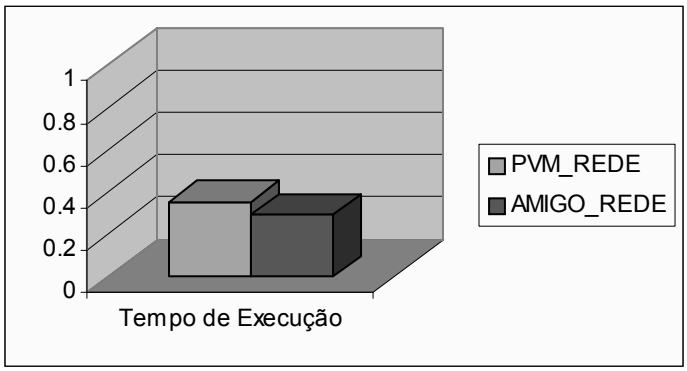

Figura 4.34 - Gráfico da execução de uma aplicação Network-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de pacotes que saem.

Tabela 4.33 - Valores calculados para o desempenho obtido com a execução de uma aplicação Network-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	0,355	0,293
Desvio Padrão	01526	0,073
Variância	0,023	0,005
Hipótese α= 0,01		**-2,011**

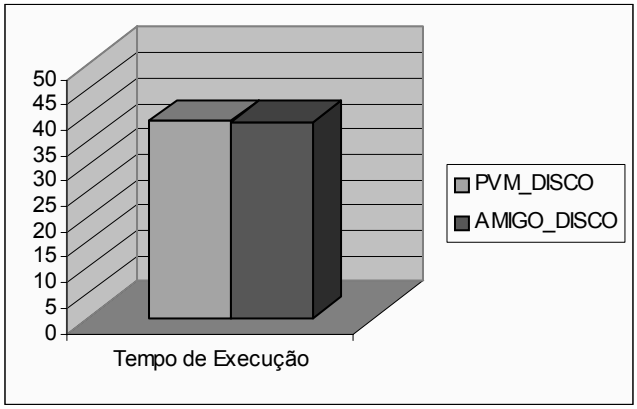

Figura 4.35 - Gráfico da execução de uma aplicação Disk-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de pacotes que saem.

Tabela 4.34 - Valores calculados para o desempenho obtido com a execução de uma aplicação Disk-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	38,807	38,607
Desvio Padrão	2,101	2,367
Variância	4,414	5,599
Hipótese α= 0,01		**-0,346**

Tabela 4.35- Teste de hipótese dos valores obtidos por classe de aplicações executadas com relação aos diferentes índices de carga de CPU implementados.

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
0,059	Número de pacotes que saem – Número de pacotes que entram	Memória

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
0,41	Número de pacotes que saem – Número de pacotes que entram	CPU

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
0,112	Número de pacotes que saem – Número de pacotes que entram	Disco

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
0,272	Número de pacotes que saem – Número de pacotes que entram	Rede

Os valores dos testes de hipótese, quando comparados diferentes índices de rede, são apresentados na tabela 4.35.

Sendo a rede um fator de grande importância em sistemas distribuídos, esse índice de carga deve ser melhor estudado e avaliado. Desse modo, como não é objetivo deste trabalho explicar profundamente índices de carga de rede, um outro trabalho está sendo desenvolvido com o objetivo de se aprofundar no estudo de índices de carga de rede e de se tirar um melhor proveito desses índices e da forma com que as aplicações são dispostas no sistema distribuído visto como um todo (Ishii, 2003).

4.6.1.4 Índices de Carga de Disco

Dentre os índices de carga de disco, têm-se: número de leituras por segundo e número de escritas por segundo.

- por número de leituras adotam-se valores obtidos através do comando *iostat -D*, que proporciona separadamente o número de leituras por segundo realizada por cada disco;
- de modo análogo, o número de escritas também pode ser obtido a partir do arquivo gerado pela execução da linha de comando *iostat -D*, que proporciona separadamente o número de escritas por segundo.

Assim como nos índices de carga analisados anteriormente, com o intuito de que o índice de carga seja significativo, e que reflita a carga atual da máquina em

termos tanto de leitura como escrita, foi definida na política de informação a obtenção dos resultados em uma base regular (intervalos de 1 segundo) e uma média exponencial após N medições.

A utilização dos *benchmarks* e a definição dos limites para os índices de disco são análogas às implementadas para os índices de memória, CPU e rede.

Os gráficos apresentados nas figuras que seguem apresentam os valores obtidos, bem como as tabelas apresentam os resultados dos testes de hipótese realizados.

No primeiro experimento utilizou-se como índice de carga o número de leituras, e no segundo experimento o número de escritas.

As Figuras 4.36 a 4.39 e as Tabelas 4.36 a 4.39 foram organizadas como pertencentes ao experimento 1. As Figuras 4.40 a 4.43 e as Tabelas 4.40 a 4.43 foram organizadas como pertencentes ao segundo experimento.

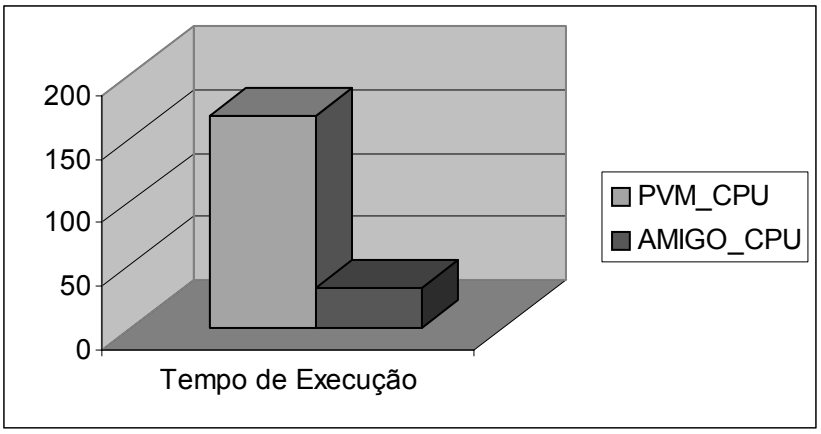

Figura 4.36 - Gráfico da execução de uma aplicação CPU-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de escritas.

Tabela 4.36 - Valores calculados para o desempenho obtido com a execução de uma aplicação CPU-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	167,639	31,637
Desvio Padrão	0,486	4,250
Variância	0,236	18,063
Hipótese α= 0,01		**-174,136**

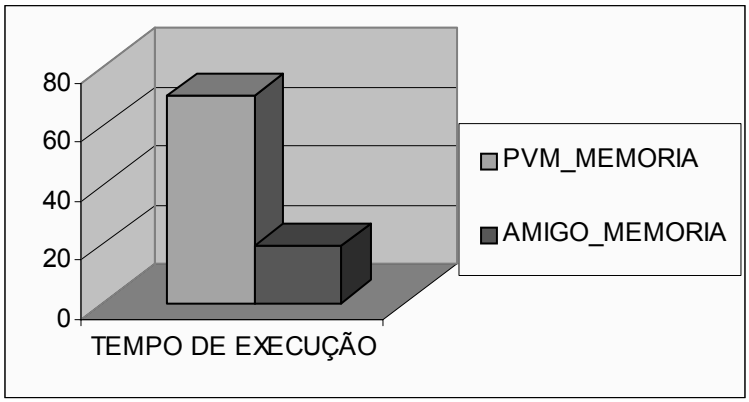

Figura 4.37 - Gráfico da execução de uma aplicação Memory-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de escritas.

Tabela 4.37 - Valores calculados para o desempenho obtido com a execução de uma aplicação Memory-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	70,4384	19,0662
Desvio Padrão	4,658	0,992
Variância	21,694	0,985
Hipótese α= 0,01		**-59,085**

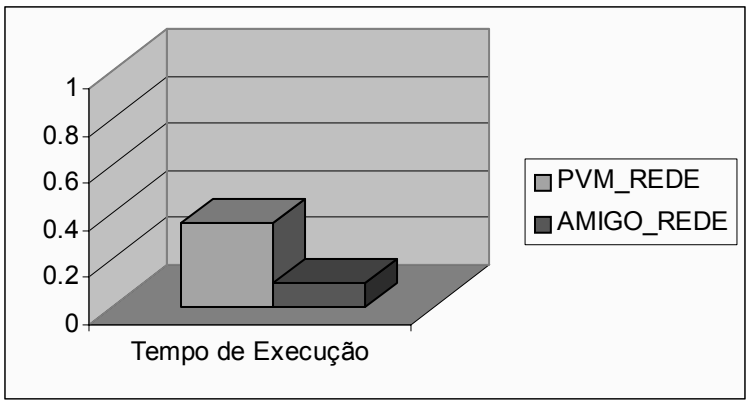

Figura 4.38 - Gráfico da execução de uma aplicação Network-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de escritas.

Tabela 4.38 - Valores calculados para o desempenho obtido com a execução de uma aplicação Network-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	0,355	0,098
Desvio Padrão	0,152	0,110
Variância	0,023	0,012
Hipótese α= 0,01		**-7,495**

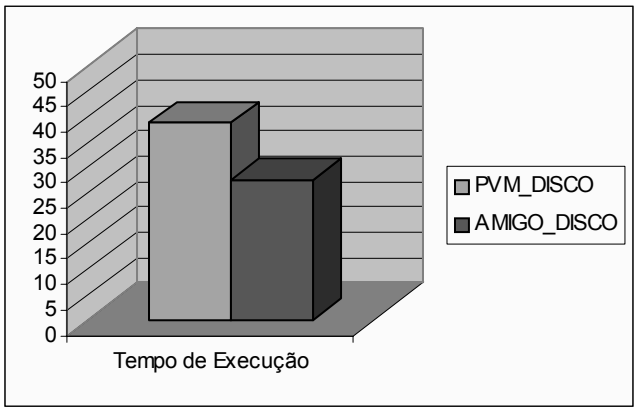

Figura 4.39 - Gráfico da execução de uma aplicação Disk-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de escritas.

Tabela 4.39 - Valores calculados para o desempenho obtido com a execução de uma aplicação Disk-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	38,807	27,575
Desvio Padrão	2,101	1,912
Variância	4,414	3,654
Hipótese α= 0,01		**-21,658**

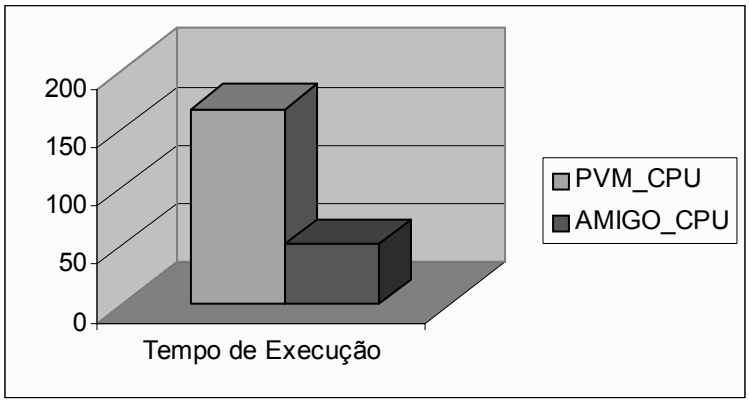

Figura 4.40 - Gráfico da execução de uma aplicação CPU-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de leituras.

Tabela 4.40 - Valores calculados para o desempenho obtido com a execução de uma aplicação CPU-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	167,639	51,933
Desvio Padrão	0,486	2,870
Variância	0,236	8,238
Hipótese α= 0,01		**-217,699**

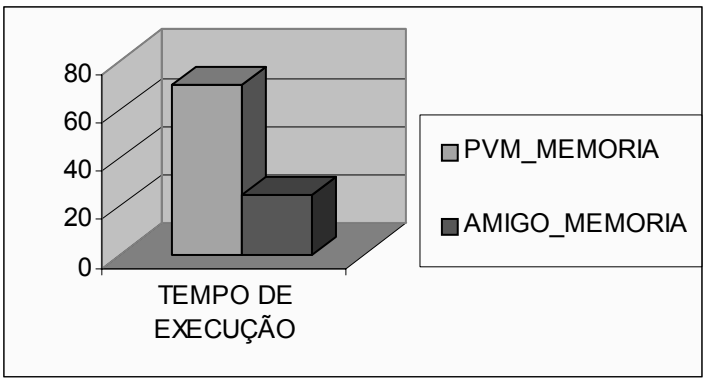

Figura 4.41 - Gráfico da execução de uma aplicação Memory-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de leituras.

Tabela 4.41 - Valores calculados para o desempenho obtido com a execução de uma aplicação Memory-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	70,438	24,268
Desvio Padrão	4,658	1,179
Variância	21,694	1,389
Hipótese α= 0,01		**-52,635**

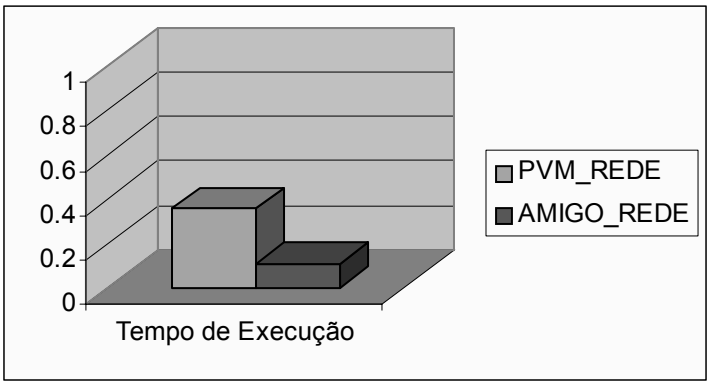

Figura 4.42 - Gráfico da execução de uma aplicação Network-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de leituras.

Tabela 4.42 - Valores calculados para o desempenho obtido com a execução de uma aplicação Network-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	0,355	0,109
Desvio Padrão	0153	0,108
Variância	0,023	0,012
Hipótese α= 0,01		**-7,211**

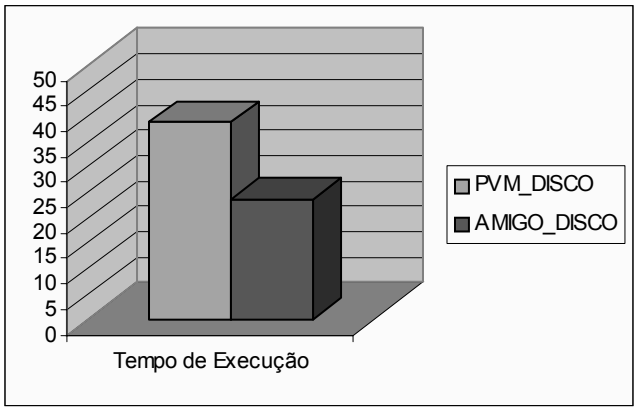

Figura 4.43 - Gráfico da execução de uma aplicação Disk-Bound no PVM comparado com a mesma aplicação fazendo uso do ambiente AMIGO executando como índice de carga o número de leituras.

Tabela 4.43 - Valores calculados para o desempenho obtido com a execução de uma aplicação Disk-Bound executada no PVM e no AMIGO.

	PVM	AMIGO
Média	38,806	23,537
Desvio Padrão	2,101	2,175
Variância	4,414	4,731
Hipótese α= 0,01		**-27,657**

Tabela 4.44- Teste de hipótese dos valores obtidos por classe de aplicações executadas com relação aos diferentes índices de carga de CPU implementados.

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
18,488	Número de escritas – Número de leituras	Memória

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
21,676	Número de escritas – Número de leituras	CPU

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
-7,637	Número de escritas – Número de leituras	Disco

Hipótese $\propto$ = 0,01	Índices	Classe de Aplicação
0,412	Número de escritas – Número de leituras	Rede

Como pode ser observado através dos testes realizados, a melhor utilização desses índices deverá ser feita através do agrupamento dos dois, caracterizando o número de transferências por segundo, respeitando os respectivos tempos necessários para efetuar as leituras e escritas (Tabela 4.44).

4.7 Considerações Finais

Os valores descritos aqui demonstram que o uso de um ambiente de escalonamento (neste caso o AMIGO) em associação com os diversos índices de carga permite ganhos reais de desempenho, com a mesma aplicação e plataforma, quando comparados a um escalonamento round robin.

Exemplificou-se também o impacto no desempenho com a utilização de diferentes classes de aplicação sob o mesmo escalonamento. Cada tipo de aplicação apresentou melhoras significativas quando a elas foram impostos escalonamentos que fazem uso de índices de carga apropriados para cada tipo, o que permitiu demonstrar a real necessidade de tipos específicos de índices para as diferentes classes de aplicações, além de deixar claro que um melhor escalonamento pode ser fruto do fato de se conhecer a priori a classe de aplicação a ser escalonada.

Através desse exemplo percebe-se que em alguns casos o escalonador não apresenta as características necessárias para realizar o escalonamento de processos apropriadamente, deixando claro que uma única política de escalonamento não consegue proporcionar bons desempenhos para quaisquer tipos de aplicação. Isso direciona a busca de um índice de carga que considere as características das aplicações de forma mais geral.

Levando-se em conta alguns dos parâmetros analisados dentre os diversos modelos matemáticos utilizados para efetuar o cálculo de índices de carga, constatou-se que um dos parâmetros necessários para a obtenção de um índice mais geral e que contemple mais fielmente ambientes heterogêneos é o grau de heterogeneidade do sistema computacional distribuído considerado.

Assim, efetuou-se um estudo detalhado das métricas existentes na literatura que permitem a medição do grau de heterogeneidade de um sistema, e que será apresentado no próximo capítulo.