6. Índice de Desempenho - Um Modelo em Redes de Fila

Este capítulo tem por objetivo apresentar o índice de desempenho proposto neste trabalho, bem como as especificações e parametrizações do modelo em redes de filas utilizado.

6.1 Considerações Iniciais

O interesse por pesquisas voltadas à obtenção de índices de carga e desempenho que possam satisfazer às necessidades de aplicações tanto acadêmicas quanto comerciais, tem aumentado consideravelmente ao longo dos últimos anos (Ferrari & Zhou, 1987; Mehra, 1993; Feitelson et al, 1997; Wolffe, Hosseini & Vairavan, 1997; Mello, 2003).

O problema em encontrar métricas que representem a situação real de um dado elemento do sistema (de um elemento de processamento, ou ainda de um subsistema completo - uma estação de trabalho, um servidor, entre outros) é ainda mais complexo, dada a natureza não determinística dos diversos domínios de aplicação que fazem uso da computação paralela/distribuída.

Desse modo, conforme observado nos diversos artigos publicados ao longo das últimas décadas e na análise crítica apresentada no capítulo 3, nenhum dos índices de carga presentes na literatura disponível foi proposto visando uma situação mais abrangente, que considere e represente mais do que um único recurso do sistema e que agregue a isso a heterogeneidade desses recursos.

Simulações prévias, tais como as utilizadas em (Ferrari & Zhou, 1987; Kunz, 1991; Mehra, 1993; Xu & Lau, 1997; Mello, 2003) entre outras, representam somente o recurso CPU, enquanto que para situações reais, como demonstram os estudos de caso apresentados no capítulo 4, fazem-se necessárias modelagens de outros recursos para a obtenção de um índice mais confiável e flexível.

Objetivando obter não somente informações sobre a carga de trabalho, mas a situação de operação de cada um dos elementos do sistema envolvido no processo,

o índice de desempenho[9] proposto neste trabalho, pode fornecer uma informação que leve em consideração não somente as heterogeneidades configuracional e arquitetural, mas também a heterogeneidade temporal das máquinas existentes no sistema, suprindo, assim, uma lacuna existente.

6.2 Visão Macroscópica

Os parâmetros necessários para a obtenção dos índices de desempenho foram levantados tendo como ponto de partida os estudo apresentados no capítulo 3. Uma visão macroscópica e *top-down* do projeto do índice de desempenho é apresentada na Figura 6.1.

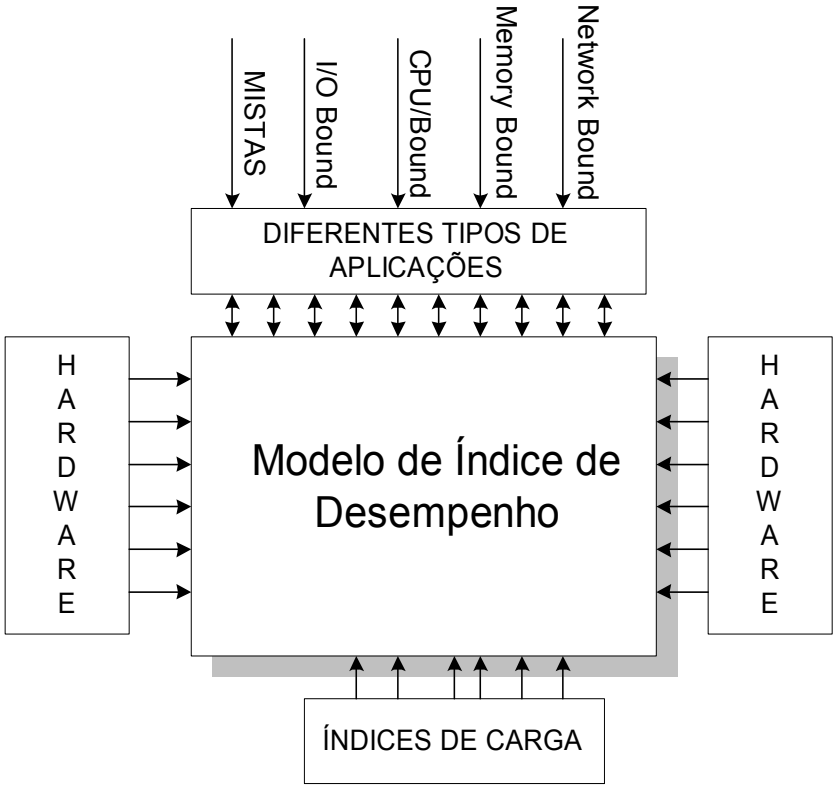

Figura 6.1 – Visão macroscópica do projeto

Para obtenção do ***M***od***E***lo ***D***e ***Í***ndice de ***D***esempenho em ***A***mbientes ***h***eterogêneos (MEDIDA*h*) faz-se necessário o conhecimento sobre os diferentes

[9] Apesar de índice de carga e índice de desempenho serem considerados, e muitas vezes utilizados como sinônimos, neste trabalho serão tratados como métricas distintas, apesar de diretamente correlacionadas

tipos de aplicações existentes, sobre o hardware (sistema computacional distribuído, neste caso específico) e sobre os índices de carga existentes atualmente.

Como visto previamente (capítulos 2 e 3), as aplicações subdividem-se basicamente em quatro grupos: CPU-Bound, Network-Bound, Memory-Bound e IO-Bound. O hardware a ser considerado é composto de quatro recursos básicos: CPU, Memória, Disco e Rede, sendo que esses podem ser (e normalmente são) heterogêneos. Quanto aos índices de carga, verificou-se que em sua grande maioria, baseiam-se em ambientes de domínio específicos, homogêneos ou, quando muito, configuracionalmente heterogêneos, mas arquiteturalmente homogêneos.

A figura 6.2 aponta as lacunas existentes na literatura no que diz respeito aos índices de carga e de desempenho, quando levados em consideração os níveis arquiteturais e configuracionais.

Como pode ser observado, existem na literatura métricas consagradas tanto para ambientes configuracional e arquiteturalmente homogêneos, como para ambientes configuracionalmente heterogêneos e arquiteturalmente homogêneos. Para ambientes configuracionalmente heterogêneos e arquiteturalmente homogêneos essas métricas não fazem sentido, e desse modo referências não são observadas na literatura.

O MEDIDA*h*, dessa maneira, vem ao encontro das necessidades citadas na literatura, onde, a existência de índices de carga e desempenho que sejam tanto configuracional quanto arquiteturalmente heterogêneos não é observada.

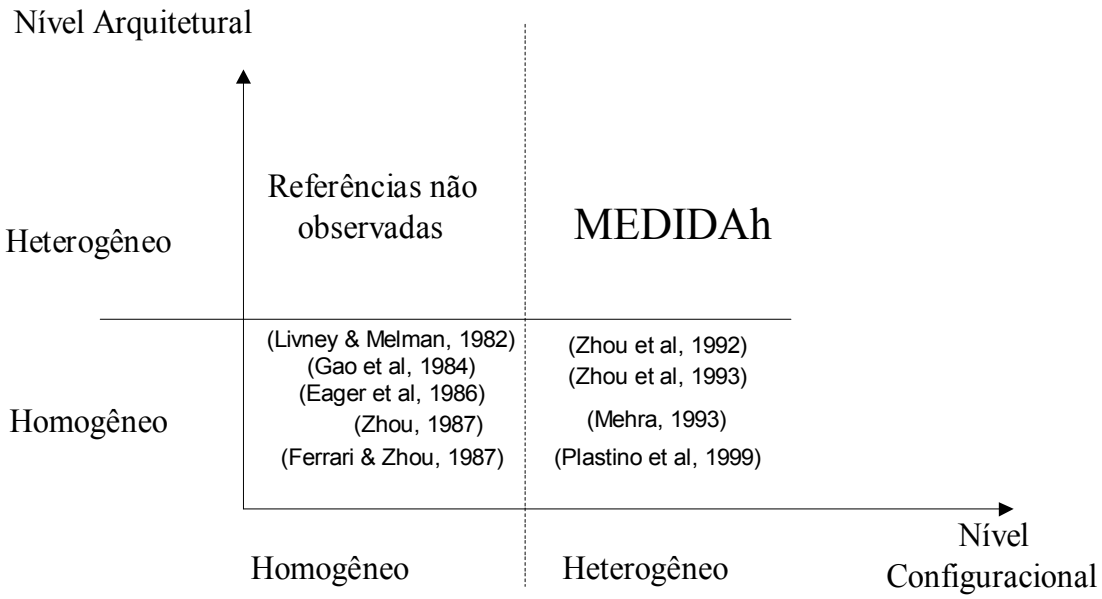

Figura 6.2 – Lacunas existentes na literatura quando levado em consideração os níveis arquiteturais e configuracionais

6.3 Gerenciamento de Recursos

Todas as políticas de distribuição de carga consideram um subconjunto de recursos do sistema como base para a distribuição da carga. Existem algumas características principais que dependem da configuração física do sistema distribuído, destacando-se:

- Sistemas que são homogêneos tanto em capacidade quanto em compatibilidade, tendem a concentrar a obtenção da carga em um único recurso, que de acordo com a literatura é considerado suficiente para a obtenção de um índice de carga (Kunz, 1991) (Harchol-Balter & Downey, 1997);
- Sistemas que são homogêneos em termos de compatibilidade de recursos, mas heterogêneos em termos da capacidade dos recursos podem ainda fazer uso somente de valores obtidos de um único recurso para representar a carga, fazendo a normalização apropriada dos valores (por exemplo, a DPWP (Araújo, 1999)), o que não quer dizer que a utilização de índices que levem em conta mais de um recurso esteja incorreto, ou que não apresente valores melhores que o uso de um único recurso;

- Sistemas puramente heterogêneos, que diferem tanto em capacidade quanto em compatibilidade, tendem a considerar a capacidade de uma máquina através da obtenção de índices que considerem diversos recursos e dessa forma poder-se efetuar a distribuição apropriada da carga.

Essas considerações implicam que não somente os requisitos dos processos a serem mapeados para sistema heterogêneos, mas sistemas homogêneos também podem se beneficiar da obtenção de índices de vários recursos e pelo controle dos tipos de aplicações (mistura de CPU, I/O e memória em qualquer máquina).

Em particular, quando observa-se um processo em execução em uma máquina, o que se tem, na realidade, é que o processo consome parte dos recursos disponíveis (como por exemplo CPU, memória, entre outros), deixando uma outra parte que pode ser utilizada por novos processos. Dessa forma, mesmo em sistemas homogêneos, pode-se observar momentos em que a capacidade do sistema disponível para a execução de novos processos indique, claramente, uma situação de heterogeneidade.

Por outro lado, pode ocorrer, de um sistema heterogêneo se apresentar como homogêneo em um dado instante, embora isso seja uma situação rara a menos que um critério bastante flexível seja usado para caracterizar a homogeneidade (Branco et al, 2003a; Branco et al, 2003c).

Além disso, deve ser observado que ao se falar em homogeneidade e heterogeneidade deve-se considerar os diferentes recursos existentes nas máquinas de um sistema. Isto é, não se pode alocar processos às máquinas considerando-se apenas a capacidade de processamento se atividades de I/O não são desprezíveis. Isso torna ainda mais complexa a visão global do sistema em termos de homogeneidade/heterogeneidade, o que deve ser refletido nos índices de desempenho adotados para o escalonamento de processos.

Portanto, observa-se que considerar a homogeneidade/heterogeneidade de um sistema só faz sentido se for definido em relação a que conjunto de recursos (processador, memória, disco, entre outros) é feita a análise e em que momento essa avaliação ocorre.

O primeiro enfoque tem sido tratado em alguns níveis nos trabalhos publicados (Maheswaran, Braun & Siegel, 1999; Siegel & Ali, 1999; Scott & Potter, 1994; Siegel, Antonio & Metzger, 1996; Siegel, Dietz & Antonio, 1997; Singh & Youssef, 1996). O conceito de homogeneidade/heterogeneidade temporal, contudo, não tem sido relatado na literatura, embora constitua um fator que não pode, de modo geral, ser desprezado.

Portanto observa-se a existência de uma heterogeneidade Temporal ou Dinâmica, o que significa que em um determinado instante, mesmo em sistemas arquiteturalmente e configuracionalmente homogêneos as diversas máquinas podem ser vistas como temporalmente heterogêneas, levando a um sistema heterogêneo.

6.3.1 CPU

A maioria dos sistemas operacionais permite o acesso a medidas da utilização de CPU fazendo uso de variáveis que representam o tempo gasto nos estados de usuário, sistema e ocioso (Maxwells, 2000; Bovet & Cesati, 2000; Ferreira, 2003). Esses valores podem, então, ser usados para compor métricas que indiquem a capacidade disponível para a execução de novos processos.

Algumas medidas disponíveis nos sistemas operacionais que podem ser utilizadas são:

- número de chamadas ao sistema;
- número de interrupções;
- tempo de CPU idle;
- tempo de CPU em funções de SO;
- tempo de CPU para programas de usuário;
- comprimento de fila de CPU;
- média do comprimento de fila de CPU;
- número de usuários ativos (logados ou ativos);
- Capacidade de carga – utilização efetiva da CPU **{(1 - CPU utilization) * relative CPU Speed}**
- Carga média do uso de CPU

A métrica mais utilizada nos trabalhos da área é o comprimento médio da fila de processos prontos para a execução. Essa métrica provê informações do número médio de processos esperando pela CPU em intervalos de tempo de 1, 5 e 15 minutos (Maxwells, 2000; Bovet & Cesati, 2000; Ferreira, 2003). Essas informações, podem ser usadas como uma medida da utilização da CPU.

6.3.2 Memória[10]

O sistema de memória torna-se um fator limitante para o desempenho quando os programa em execução necessitam de mais memória do que a disponível fisicamente na máquina. A memória é um recurso importante que deve ser cuidadosamente gerenciado, constituindo um dos pontos críticos do sistema.

Dessa forma, dentre os índices de carga já publicados e investigados na literatura para a obtenção da carga do recurso memória têm-se :

- quantidade de memória utilizada;
- tamanho da área de *swap*;
- quantidade de paginação;
- número de trocas de contexto (*switches*);
- número de processos esperando por memória livre;
- páginas de memória usadas por todos os processos;
- páginas de memória usadas pelos processos ativos;
- número de páginas livres de memória;
- Memória virtual disponível.

Quando se pensa em memória de modo geral, sabe-se que os sistemas operacionais normalmente fazem uso de regiões de disco para armazenamento temporário. Essas regiões, denominadas partições de *swap*, são usadas como áreas de armazenamento para a memória virtual quando o sistema não tem memória física suficiente para manipular os processos correntes.

[10] Como decisão de projeto, optou-se por trabalhar somente com máquinas MIMD de memória distribuída, a fim de delimitar um contexto de atuação.

Considerando que o acesso a disco é várias vezes mais lento que o acesso à memória física, o tamanho da área de *swap* e o número de vezes que o processo de paginação[11] é executado constituem fortes parâmetros a serem analisados e avaliados quando se pensa em desempenho do recurso memória, sendo que esse desempenho deteriora quando o sistema começa a realizar uma quantidade muito grande de paginação.

O desempenho da memória degrada quando o sistema de memória não atende adequadamente à demanda de páginas; há problemas de falta de área de *swap;* processos utilizam grandes quantidades de memória, entre outros.

6.3.3 Disco

Para o recurso disco, as medidas disponíveis nos sistemas operacionais que podem ser utilizadas são:

- quantidade de pedidos de entrada/saída;
- número de processos esperando na fila por disco;
- taxa de dados transferida em cada disco;
- porcentagem de utilização de cada disco;
- leituras por segundo;
- escritas por segundo.

O número de transferências no disco, utilizado como uma métrica, é um contador acumulativo e requer no mínimo duas amostras para medir o desempenho do número de transferências do disco em um certo período de tempo.

Uma vantagem dessa métrica com relação às demais é que ela contém o *swapping*, porque cada troca de página de memória induz transferências de disco. Isto deve ser levado em consideração quando combinar as métricas para formar uma função de carga para prever a utilização da memória a partir daí.

[11] corresponde à troca de uma página de memória física para memória virtual e/ou vice-versa.

6.3.4 Rede

A comunicação através de uma rede de interconexão é um outro fator importante que pode degradar o desempenho global de algumas aplicações. Considerar o efeito dessa comunicação em um índice de desempenho não é trivial (Ishii, 2003).

Vários fatores influenciam o desempenho de aplicações que requerem altas taxas de comunicação. A tecnologia da rede de interconexão tem papel fundamental, uma vez que a capacidade de comunicação está diretamente ligada aos detalhes da tecnologia.

Por exemplo, em redes onde ocorrem colisões (por exemplo uma Ethernet 100 Mb interligada por um *switch*), o volume de colisões pode ser usado como uma métrica de utilização da rede, indicando se a capacidade de comunicação está próxima ou não da saturação.

Algumas possíveis métricas para avaliar a carga na interface de rede de uma máquina compreendem:

- Número de pacotes entrando/saindo de cada interface de rede;
- Número de colisões;
- Taxa de transferência dos dados;
- Taxa de erros.

O número de pacotes com erros pode também indicar altas taxas de falha no meio de transferência, levando a queda de desempenho. Essas métricas não são boas para medir a carga gerada na rede por uma máquina, porém são apropriadas para medir o desempenho global da rede que pode ser usado em decisões de escalonamento de processos que requerem comunicação.

6.4 Índices de Carga Visando ao Índice de Desempenho.

O índice de desempenho apresentado em Santana (Santana & Zaluska, 1988; Santana, 1990), em uma versão preliminar que considerava o balanceamento de cargas em um conjunto de servidores de arquivos, leva em consideração heurísticas e um mecanismo para normalizar o índice construído. Esse mecanismo considera a heterogeneidade do ambiente através de uma experimentação prática com cada um

dos servidores de arquivos existentes no sistema. A partir dessa experimentação, curvas que representam o padrão de desempenho de cada servidor são estabelecidas e o índice de desempenho é construído considerando-se esse padrão.

Essa abordagem proposta para os servidores de arquivos será tomada como ponto de partida para se estudar o problema mais genericamente, buscando-se estabelecer modelos que possam representar o padrão necessário para o estabelecimento dos índices de desempenho.

Entende-se por índice de desempenho a métrica que seja capaz de fornecer uma imagem da capacidade de trabalho, ou melhor, que constitui uma grandeza que ilustra claramente o que pode ser esperado, em termos de desempenho, do elemento em análise.

Nesse sentido, a heterogeneidade do ambiente deve ser considerada para se ter uma visão real da potência computacional disponível. Sendo assim, esse índice deve levar em consideração também a heterogeneidade das aplicações, de modo que uma medida real da potência computacional seja obtida.

Um bom índice de desempenho, assim como os índices de carga, deve possuir meios de estimar o futuro através de valores atuais e fatores do passado e, sendo assim, para que se possa obter um bom índice de desempenho suas bases devem estar fundadas nos índices de carga.

Como tem sido observado, os índices de carga são por deveras voláteis, deixando aparente a instabilidade das métricas consideradas. Essa instabilidade ocorre devido às flutuações das cargas de trabalho. Não existindo determinismo tem-se a necessidade de elaborar modelos que reflitam essa característica, e que considerem as flutuações da carga em um intervalo de tempo.

O desafio do não-determinismo está na estratégia de aprendizado para descobrir regras heurísticas que permitam a escolha da “melhor” alternativa, sem que exista necessidade de explorar todas elas.

Um índice de desempenho, nesse contexto, tem por objetivo a generalização dos procedimentos, além de permitir que as abordagens já desenvolvidas possam ser testadas e consideradas, constituindo assim uma nova abordagem que contemple a heterogeneidade do ambiente. Pode, ainda, ser considerado como uma

nova estratégia para se medir o comportamento de um dado sistema. A figura 6.3 apresenta a estratégia para obtenção do índice de desempenho.

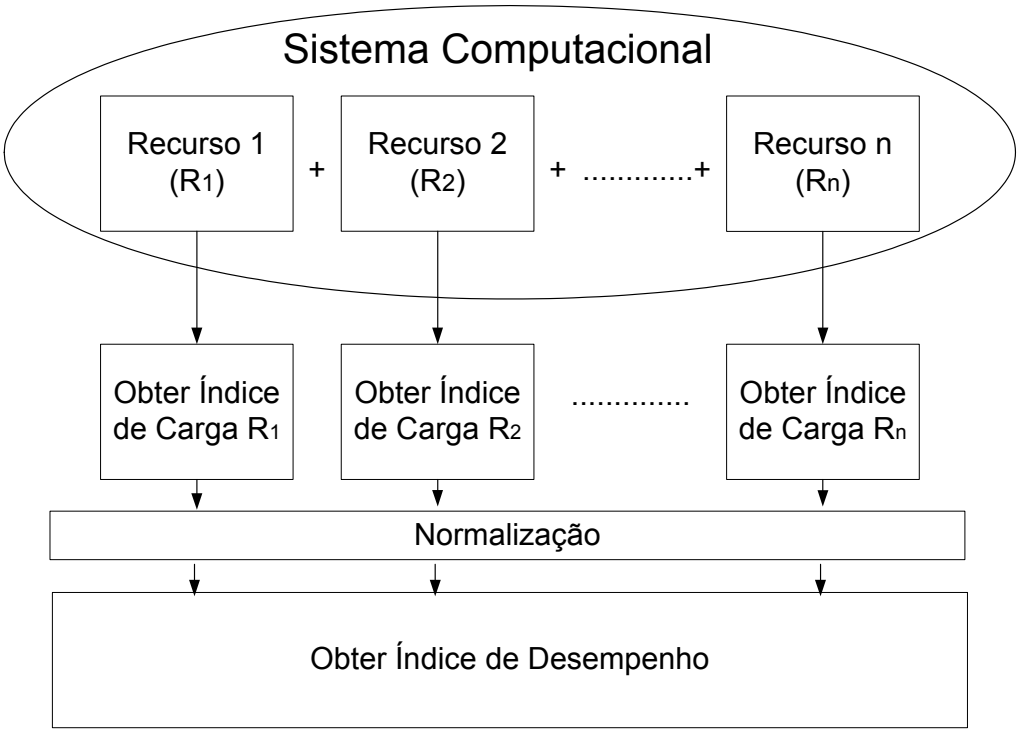

Figura 6.3 - Estratégia para Obtenção do Índice de Desempenho

Tendo-se em vista os quatro recursos básicos a serem analisados em uma máquina, pode-se então, partindo da estratégia apresentada anteriormente, obter a função apresentada na equação 6.1:

$$ID = f(I_{CPU}, I_{Memoria}, I_{Disco}, I_{Rede})$$

Equação 6.1

A função apresentada na equação 6.1 pode, ainda, fazer uso de pesos para cada um dos índices específicos dos recursos:

$$ID = f(W_1(I_{CPU}), W_2(I_{Memoria}), W_3(I_{Disco}), W_4(I_{Rede}))$$

onde ID é o índice de desempenho que leva em consideração os quatro recursos básicos. W_1, W_2, W_3 e W_4 são os pesos que serão dados aos índices de acordo com a característica da aplicação a ser escalonada.

I_{CPU} é uma combinação dos índices de CPU que mais se adequam a aplicações estritamente CPU-Bound. $I_{Memória}$ é a combinação dos índices de Memória que mais se adequam a aplicações estritamente Memory-Bound, I_{disco} é a

combinação dos índices de Disco que mais se adequam a aplicações estritamente Disk-Bound e I_{rede} é a combinação dos índices de rede que mais se adequam a aplicações estritamente Network-Bound.

Cada índice de carga será calculado independentemente e levará em consideração um *benchmark* específico. O problema das medidas é que elas são apresentadas com valores que não podem ser diretamente combinados e comparados a partir das múltiplas máquinas, sem que ocorra a normalização. Cada medida opera em uma escala aberta, o que implica que o valor mínimo é igual a zero, entretanto o valor máximo não pode ser determinado porque depende da utilização e capacidade de cada máquina.

Desse modo, cada medida é normalizada separadamente de modo que cada índice específico dos recursos CPU, Disco, Memória e Rede tenha seu valor apresentado entre 0 e 1 (os índices I_{CPU}, I_{Disco}, $I_{Memória}$ e I_{Rede}, podem ser elaborados a partir de uma média ponderada $\sum_{i=1}^{n} \frac{I_{recurso}}{n}$ de vários índices de carga, contemplando várias visões do uso do recurso).

Com base nos resultados obtidos no capítulo 4, foram escolhidos os índices *capacidade de carga*, *memória propriamente livre + swap*, *número de escritas e leituras*, e o *número de pacotes que entram e saem* na interface de rede, para compor o índice de desempenho que será utilizado no estudo de caso apresentado nesta tese (esses índices não são necessariamente os únicos a serem utilizados, outros podem ser incluídos ou até mesmo substituídos).

Uma vez que cada medida é normalizada segundo os *benchmarks* relativos (permitindo a comparação das máquinas de igual para igual), e que as máquinas podem ser dispostas segundo uma classificação com valores variando de 0 a 1, os valores de cada uma das medidas dos diferentes recursos podem, simplesmente, ser somados e ponderados.

Após gerada a função do índice de desempenho, torna-se necessária, para finalização, a escolha dos pesos apropriados.

Esses pesos serão definidos com base nas características das aplicações. Um trabalho de doutorado em desenvolvimento no grupo de Sistemas Distribuídos e Computação Paralela do ICMC-USP visa exatamente à obtenção dessas

características e desses pesos com relação aos recursos utilizados por essas aplicações, através da utilização de técnicas de redes neurais (Senger, 2001).

Dessa maneira, cada aplicação será averiguada quando de sua chegada, e suas características serão repassadas para o escalonador, permitindo-se determinar qual índice terá mais peso no cálculo final do índice de desempenho.

6.4.1 Uma Variante do Índice de Desempenho.

Uma característica interessante da função apresentada na equação 6.1 é a possível utilização desta como índice de carga específico para cada recurso, CPU, Disco, Rede, Memória, podendo cada uma delas ser vista como um vetor base. Assim, considerando na representação do vetor a ordem <CPU, Disco, Rede, Memória>, o vetor base <1,0,0,0> representa uma aplicação 100% CPU. Da mesma forma, pode-se ter <0,1,0,0> para aplicação 100% Disco, <0,0,1,0> para aplicação 100% rede e <0,0,0,1> para aplicação 100% memória.

Os *n* recursos que uma máquina pode prover podem ser considerados para formar um espaço n dimensional. Se uma máquina provê os recursos CPU, Rede, Disco e Memória então ela forma um espaço quadri-dimensional, no qual um ponto localiza o estado atual desta máquina.

Uma vez que a faixa de valores desses recursos concentram-se entre 0 e 1, e que esses recursos são tratados de modo vetorial, então uma máquina ociosa localiza-se na origem <0,0,0,0> e uma máquina completamente sobrecarregada localiza-se no vértice oposto <1,1,1,1>.

A título de exemplificação, a figura 6.4 apresenta um espaço tridimensional, uma vez que espaços quadri-dimensionais são complexos em termos de representações gráficas e a representação de um hipercubo tornaria a visualização muito complexa.

Na figura 6.4, podem ser observados três pontos particulares de carga em uma máquina. A situação A representa uma máquina com grande utilização de CPU, que não utiliza memória e com uma utilização média de rede. De modo análogo, pode-se observar que a situação B faz uso médio de memória e médio de rede enquanto que a situação C faz grande uso de CPU, de memória e de rede.

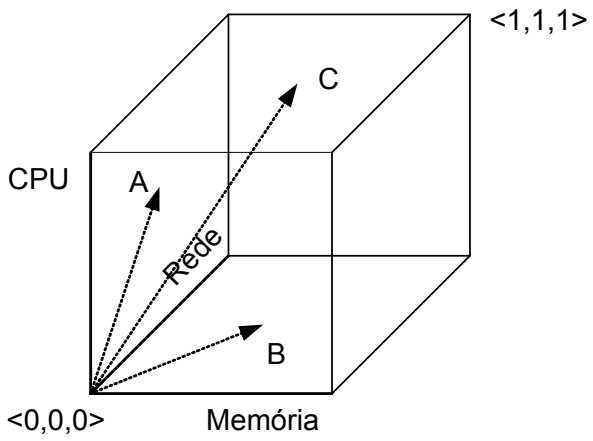

Figura 6.4 - Espaço tri dimensional usado para descrever a carga atual de uma máquina e os três pontos indicando cargas potenciais da máquina.

Através dessa representação dois tipos de informação podem ser obtidos. O ângulo entre o vetor da utilização da máquina e o eixo x, mostra a porcentagem relativa de utilização cada recurso. O comprimento do vetor representa quanto de cada recurso é utilizado.

Considerando dois recursos (1 e 2), o balanceamento das cargas pode ser observado através do ângulo θ, sendo que para $\theta \approx 45º$, ambos recursos estão igualmente carregados; $\theta >> 45º$ indica que o recurso 1 é predominante e $\theta << 45º$ indica que o recurso 2 é o predominante.

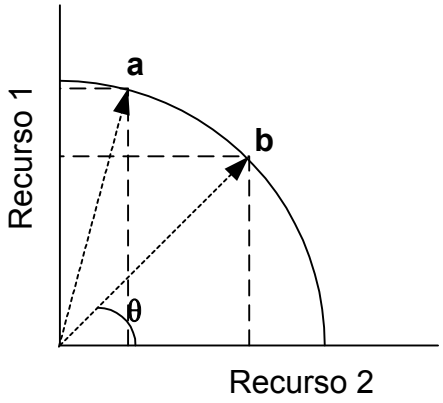

Figura 6.5 - Áreas diferentes para dois vetores com mesmo comprimento (ângulos distintos)

Uma vez que o comprimento dos vetores é o mesmo, apesar da máquina **b** estar balanceada ($|45º - \theta| = 0$) com relação aos recursos 1 e 2, ela é classificada igualmente à máquina **a** que está menos sobrecarregada que a **b** com relação ao recurso 2.

Para que sejam notadas as condições de sobrecarga, deve-se verificar simultaneamente o ângulo θ e o comprimento do vetor. Se $\theta \approx 0$ e comprimento tende a 1, então o recurso 2 está próximo da saturação; da mesma forma, se $\theta \approx 90$ e comprimento tende a 1 então o recurso 1 está próximo da saturação.

A figura 6.6 ilustra a chegada de um processo P, que utiliza apenas o recurso 1 (R_1 bound) em duas máquinas distintas mas que estão igualmente carregadas.

O recurso 1 está mais carregado na máquina M_1 enquanto que o recurso 2 está mais carregado na máquina M_2 em termos de utilização. O processo P (que é recurso 1 bound) pode ser alocado em M_1 e M_2. Deve-se determinar em qual situação obtêm-se um melhor resultado. Uma métrica que pode ser adotada, neste caso, é a distância Euclidiana entre o ponto e a origem, isto é, o comprimento do vetor da origem ao ponto. Portanto, a equação 6.1 pode ser reescrita como:

$$\text{ID} = \sqrt{I_{Cpu}^2 + I_{Disco}^2 + I_{Memória}^2 + I_{\text{Re}de}^2} \ .$$

Para o exemplo apresentado na figura 6.6, obtêm-se os vetores C_1 e C_2. O resultado do comprimento C_2 é menor que o comprimento C_1 e dessa forma o processo P é alocado em M_2.

Isso demonstra que apesar das máquinas estarem igualmente carregadas, a distinção de carga quanto aos recursos que estão sendo utilizados e o tipo de tarefa que será alocada permite uma melhor alocação da tarefa. Essa identificação de carga por recurso é provida pela métrica aqui proposta.

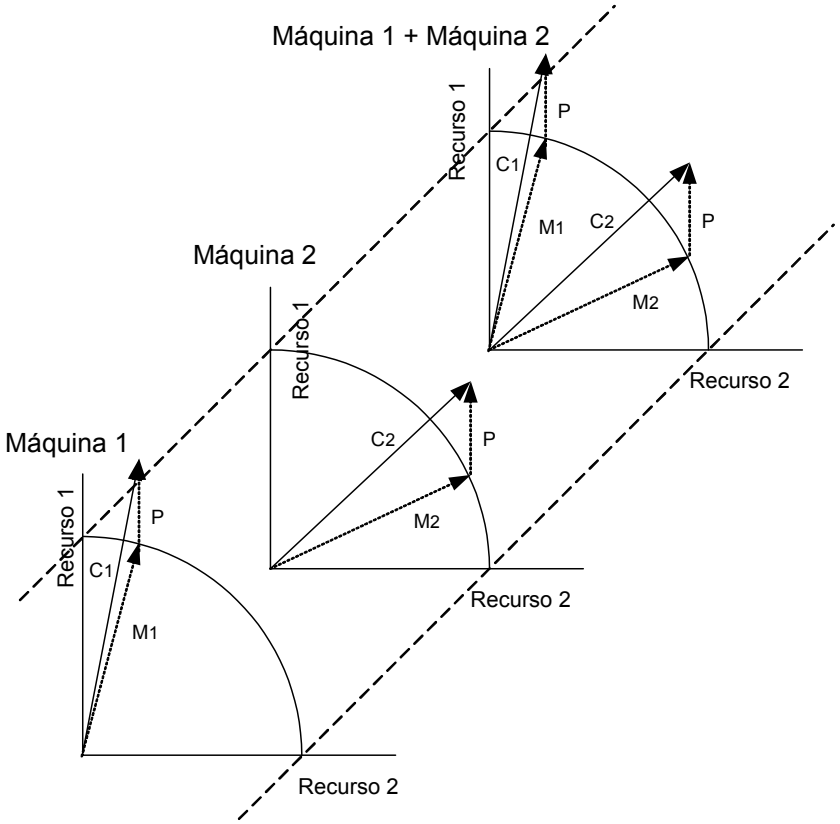

Figura 6.6 - Espaço bidimensional formado pelos recursos 1 e 2, e duas máquinas com cargas iguais (processo limitado por um recurso).

O exemplo apresentado na Figura 6.6 ilustra um exemplo simples, pois o processo utiliza apenas um recurso. Um exemplo mais complexo pode ser dado por uma aplicação que utilize mais de um recurso, como ilustrado na Figura 6.7.

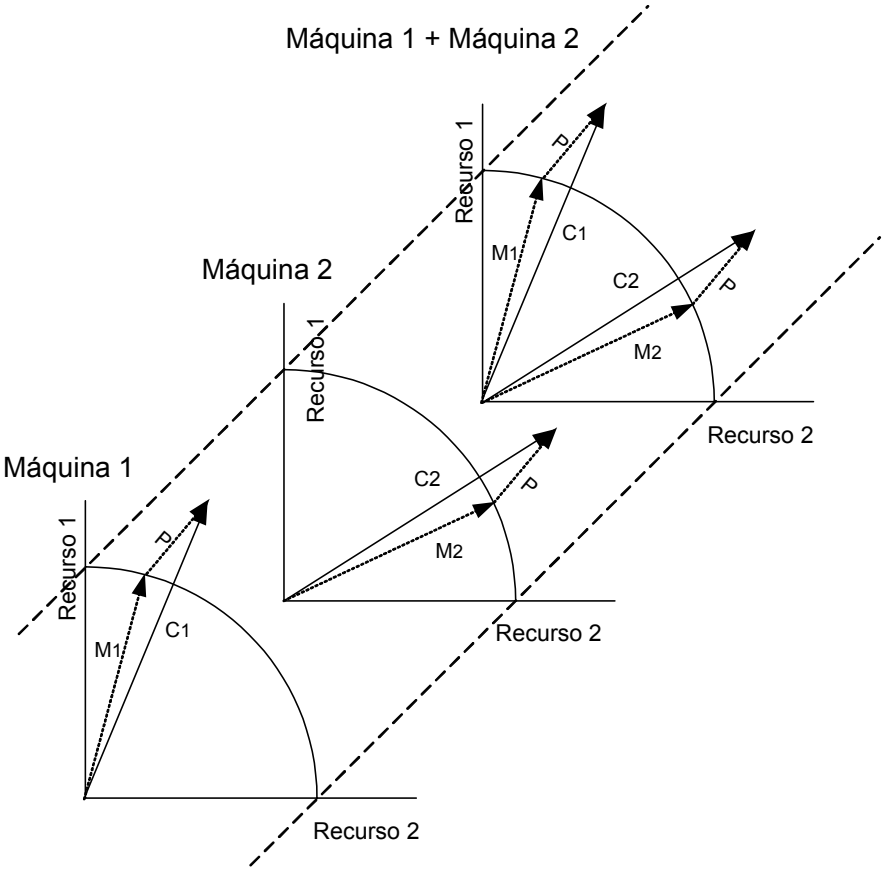

Figura 6.7 - Espaço bidimensional formado pelos recursos 1 e 2, e duas máquinas com cargas iguais (processo limitado por dois recursos).

Nesta figura, o processo P utiliza mais de um recurso em duas máquinas distintas mas que estão, novamente, igualmente carregadas.

O recurso 1 está mais carregado na máquina M_1 enquanto que o recurso 2 está mais carregado na máquina M_2 em termos de utilização. O processo P pode ser alocado em M_1 e M_2. Deve-se determinar, entretanto, em qual máquina obtêm-se um melhor resultado, e para isso faz-se uso do vetor resultante. O vetor resultante que apresentar menor comprimento indica para qual máquina o processo P deverá ser alocado. Neste caso, observa-se que o processo deve ser alocado na máquina M_2 para que se possa obter um melhor desempenho.

Partindo das análises tecidas pode-se notar que o proposto neste trabalho não considera valores particulares de cada recurso, mas sim a relação existente entre os diferentes recursos que compõem uma máquina, permitindo que a alocação dos processos seja efetuada e uma maneira mais equilibrada.

Desta forma, o índice de desempenho proposto nesta tese de doutorado baseia-se na distância euclidiana entre o ponto origem (onde a máquina está ociosa)

e o ponto resultante entre os vetores de carga da máquina antes de receber uma determinada aplicação mais o vetor da carga imposta por essa aplicação. A máquina mais adequada para receber a aplicação é aquela onde se obtém a menor distância euclidiana. Esse índice, baseados nos vetores de carga, será referenciado no restante deste trabalho por *VIP (Vector for Index of Performance)*.

O VIP considera que os pesos, definidos na equação 6.1, serão iguais para todos os recursos. Outras variantes do *VIP* podem ser estabelecidas com pesos diferentes obtendo-se o PVIP (*Ponderated Vector for Index of Performance*). O PVIP pode apresentar melhores resultados para os casos em que se tem algum conhecimento do tipo de aplicação a ser considerada ou quando for possível a utilização de um índice adaptativo.

6.5 Avaliação do MEDIDAh

No capítulo 4 foram apresentados testes de execução de índices de carga fazendo uso do ambiente AMIGO.

Em virtude da quantidade de testes, da necessidade de se efetuar testes que levariam muito tempo, optou-se pelo desenvolvimento de um modelo em redes de fila e pelo uso de simulação para avaliação da nova métrica proposta.

A representação do sistema através de um modelo que representa os índices de carga e desempenho e a solução desse modelo através da simulação torna-se atrativa principalmente quando se deseja inserir modificações e obter resultados. A análise de índices de carga e de desempenho é complexa, principalmente do ponto de vista da configuração das máquinas e do ambiente de escalonamento do qual se fará uso.

O processo de desenvolvimento de uma simulação envolve diversas etapas. Em primeiro lugar, faz-se necessário especificar o modelo, abstraindo as características mais importantes do sistema. Tendo o sistema sido modelado, é necessário transformar o modelo em um programa de simulação. Em simulações estocásticas, devido à aleatoriedade dos dados de entrada, o programa deve ser executado diversas vezes a fim de garantir que a influência da aleatoriedade nos resultados finais seja minimizada.

6.5.1 Metodologia Para o Desenvolvimento da Avaliação de Desempenho Através de Técnicas de Modelagem

A metodologia do estudo de desempenho através das técnicas de modelagem é composta de vários passos que englobam o desenvolvimento do modelo, os testes para garantir que este está correto e a obtenção dos resultados através da experimentação do modelo. O primeiro passo em um estudo de avaliação de desempenho, independente da técnica utilizada para resolver o modelo, consiste em identificar o problema que gerou a necessidade de avaliação de desempenho. Feito isso, deve-se analisar o sistema que vai ser avaliado no estudo e estabelecer os objetivos desejados. A seção 6.5.2.1 descreve esses passos, aplicados ao estudo de índices de carga e desempenho.

Uma vez estando de posse dos objetivos da análise de desempenho, o sistema é cuidadosamente estudado para que se possam abstrair as características fundamentais para a construção de um modelo representativo. Uma tarefa relativamente difícil nesta fase consiste na tomada de decisão sobre quais elementos do sistema devem ser incluídos no modelo e de que forma incluí-los. O nível de detalhamento deve basear-se no propósito para o qual o mesmo está sendo construído. A seção 6.5.2.2 detalha esses passos.

O passo de formulação do modelo gera os requisitos para os dados de entrada que servirão como parâmetros do mesmo. Quando a modelagem é efetuada sobre um sistema existente, os parâmetros do modelo podem ser medidos, caso contrário, devem ser estimados (seção 6.5.2.3).

Deve-se, então, escolher a técnica para solução do modelo, conforme apresentado na seção anterior. A escolha da simulação envolve o desenvolvimento de uma representação computacional do modelo (seção 6.5.2.4), o qual deve ser conferido para garantir que está livre de erros de programação e de lógica. A verificação compara a representação computacional com o modelo visando garantir que o modelo foi fielmente representado. Não existem regras específicas para efetuar essa tarefa, mas sim algumas abordagens que podem ser seguidas, como, por exemplo, inspeção, comparando o programa de simulação e o modelo (Higginbotton, 1998). Nesse caso especificamente, os testes executados em plataforma real permitiram a obtenção dos parâmetros e estão sendo de grande valia na validação do modelo.

O último passo antes do início da experimentação do modelo para a obtenção dos resultados envolve a validação, que é utilizada para mostrar que o modelo representa o sistema em estudo, ou seja, que reproduz seu comportamento. Quando o sistema em estudo existe e pode ser utilizado para medições, a validação pode se basear na comparação dos resultados do modelo com aqueles obtidos nas medições efetuadas no sistema real. Se o sistema não existe é preciso utilizar alguma forma de validação do modelo conceitual.

Verificado e validado o modelo, pode-se experimentá-lo para obter os resultados desejados, selecionando as medidas que serão utilizadas para avaliar o desempenho. Deve-se tomar cuidado na obtenção e análise dos resultados da simulação por se tratar de uma simulação estocástica. Para isso, utilizam-se técnicas de análise de saída que auxiliam na obtenção de uma estimativa precisa das medidas de desempenho.

6.5.2 Desenvolvimento da Avaliação de Desempenho

As seções que seguem descrevem o estudo de desempenho através de um modelo representativo do comportamento de um escalonador de processos, segundo os passos de uma simulação. Esse escalonador, por sua vez, faz uso de uma única política, entretanto, avalia diferentes índices de carga e desempenho. O primeiro passo foi iniciado em capítulos anteriores que descreveram o comportamento de um escalonador de processos.

6.5.2.1 Identificando Problemas e Objetivos

Neste trabalho o sistema em estudo consiste na avaliação do comportamento dos diversos índices de carga em comparação com o índice de desempenho proposto, visando não somente apresentar o comportamento dos mesmos, bem como determinar a sobrecarga causada pelo uso impróprio desses índices de acordo com tipos específicos de classes de aplicações que a ele são submetidos.

Uma vez que a maioria dos escalonadores de processos se comporta de maneira semelhante, e o objetivo não é estudar as diferenças entre escalonadores, o objetivo do estudo de desempenho é construir um método que permita a avaliação dos diversos índices, oferecendo ao usuário da computação distribuída uma maneira objetiva de escolher o índice e a(s) classe(s) de aplicação(ões) que melhor se adaptem.

6.5.2.2 Características Importantes para a Construção do Modelo

Definida a meta do estudo de desempenho, o passo seguinte corresponde à identificação das características do sistema que apresentam impacto no desempenho do mesmo e que, portanto, devem ser incluídas em um modelo representativo. Essa tarefa exige familiaridade com os índices e com o funcionamento do escalonador de processos, bem como das políticas e do comportamento das diferentes classes de aplicações, além de um conhecimento mais profundo do que a descrição elementar encontrada na literatura.

Dessa forma, foi necessário identificar e definir todas as tarefas executadas. A implementação dos índices e a execução dos mesmos, realizada na etapa anterior (capítulo 4), serviram de alicerce e permitiram a obtenção desse conhecimento.

Uma das dificuldades em se desenvolver um modelo – uma abstração – do comportamento básico dos índices reside no fato de que não existem descrições detalhadas dos mesmos na literatura. A maioria dos trabalhos efetuados e encontrados analisa estudos de caso específicos e constituem implementações reais, fazendo com que as definições dos parâmetros e do modelo se tornassem tarefas mais difíceis.

Para que os índices pudessem ser avaliados optou-se por utilizar um escalonador (responsável unicamente pela atividade de escalonar os processos). A esse escalonador estarão associados tempos de escalonamento e tempo de atualização da tabela dos índices nas demais máquinas, dependendo do tipo de escalonamento a ser realizado.

Cada máquina é subdividida em elemento processador, rede e disco, sendo a memória considerada como um elemento quantitativo e não como um centro de serviço propriamente. Cada máquina pertencente ao sistema é ainda responsável por receber as tarefas (neste caso as classes de aplicações que serão inseridas para serem escalonadas) e atualizar a tabela de índices. (Esses tempos de atualização da tabela foram obtidos a partir dos experimentos realizados fazendo uso do ambiente paralelo virtual PVM e do ambiente de escalonamento AMIGO).

Os elementos de processamento têm seu tempo de serviço relativo uns aos outros, medidos de acordo com a capacidade de cada um. Esse valor é obtido a partir do desempenho das aplicações baseado nos *GHZxIPC* (GHz vezes as

Instruções por Clock) (White Paper AMD, 2002). Da mesma forma, o elemento disco será definido por *t= tempo de seek + o tamanho do arquivo/bandwidth* (White Paper AMD, 2002).

Para os elementos de rede será utilizado o tamanho da mensagem/80Mb/s (considerando uma rede interligada por um *switch* que garante essa capacidade de transmissão (Kant & Mohapatra, 2000)).

Com essas especificações podem-se realizar experimentações com diferentes tipos de configurações de máquinas, classes de aplicações e índices. Além disso, será possível avaliar o impacto da heterogeneidade das máquinas e a polaridade (positiva ou negativa) do grau de heterogeneidade variando-se as configurações e os parâmetros utilizados.

6.5.2.3 Definição dos Parâmetros do Modelo

Para ser utilizado e produzir os resultados de desempenho esperados, o modelo de simulação deve receber os parâmetros adequados. Como o objetivo do estudo é a avaliação dos índices de carga e desempenho proposto, são necessários parâmetros relativos não somente à composição desses índices, mas também referentes ao escalonador e às classes de aplicações.

Esses parâmetros dizem respeito à configuração física das máquinas (por exemplo: quantidade de memória, capacidade do processador, especificações de disco, tempo de quantum) e referentes aos índices de carga e desempenho propriamente ditos aliados aos parâmetros do escalonador (por exemplo, tempo necessário para que a atualização dos índices possa ser feita, entre outros).

A métrica chave utilizada é o tempo médio de resposta observado, que mede o tempo consumido por um processo no sistema do início ao fim de sua execução (Ferrari & Zhou, 1987; Shivaratri et. al., 1992). Esse tempo médio de resposta influencia no desempenho final do sistema, sendo que quanto menor o tempo de resposta, maior o desempenho.

Este projeto de doutorado, assim como os principais trabalhos da área de alocação de carga (Ferrari & Zhou, 1987; Shivaratri et. al., 1992; Feitelson & Rudolph, 1995; Feitelson & Rudolph, 1996; Feitelson & Rudolph, 1998; Mello, 2003), considera como resultado da simulação o tempo médio de resposta dos processos.

6.5.2.4 Desenvolvimento do Modelo em Redes de fila

Existem várias técnicas para representar um modelo, como por exemplo Redes de Fila (MacDougall, 1987), Redes de Petri, *Statechart* (Petri, 1966)*, Estelle* (Budkowski & Dembinski, 1987). As Redes de Fila são mais adequadas em situações onde existam os conceitos de clientes sendo atendidos por um prestador de serviços, como é o caso do modelo a ser implementado.

Como ferramenta para implementação do modelo foi utilizado o ambiente de simulação ASIA (Bruschi, 1997) desenvolvido no ICMC-USP. Esse ambiente faz uso da linguagem SMPL (MacDougall, 1987) que permite que especificação do modelo de uma maneira gráfica e interativa.

- Modelo em Redes de Fila

Tomando como base os experimentos realizados com os índices de carga e a partir da experiência adquirida com relação a esses índices e ao comportamento do ambiente de escalonamento utilizado (AMIGO), foi possível obter-se o embasamento necessário para a construção do modelo.

Os quatro recursos fundamentais são modelados: CPU, Rede, Disco e Memória. O modelo de índice de carga é projetado para ser utilizado em ambientes heterogêneos.

O fluxo de cada aplicação (processo ou tarefa) parte do recurso escalonador para a fila do processador. Neste recurso o processo fica o tempo necessário para processamento até que um outro recurso seja requisitado ou que o quantum expire. Os processos se movem pelos recursos do sistema e retornam para o fim da fila do recurso processador até que seja finalizado.

- Escalonador - para prover uma alocação realística e poder fazer uso de uma política e dos índices de carga e desempenho, um sistema de escalonamento é necessário. O escalonador é um centro de serviço que tem como responsabilidade efetuar a distribuição dos processos para os processadores existentes no sistema. Essa distribuição é feita tomando-se como base uma tabela de carga que classifica os elementos do sistema de acordo com o índice de carga utilizado. Desse modo, dois são os tempos de serviço adotados para o escalonador, um quando a alocação do processo é feita somente

através da consulta à tabela e outra quando se faz necessária não só a consulta desta, mas também sua atualização.

- Processador - o processador é a parte principal do modelo, uma vez que todos os processos devem passar por ele antes que decida visitar qualquer outro elemento do sistema. Cada processador controla seu quantum e está em um dos dois estados: ocioso ou executando. No estado executando executa o tempo do quantum e inicia o próximo evento. No estado ocioso não existe processo a ser executado. A taxa de serviço do processador é dada por *GHZxIPC.* O tempo total de processamento é dividido em n+1, onde n é o número total de visitas que o processo necessita fazer para ser finalizado. Cada visita ao processador consome o tempo do processador e depois decide se deseja visitar outro recurso ou continuar no elemento processador. Caso o tempo do processo expire o quantum, este retorna ao processador, e o tempo total é recalculado para que represente a nova visita. Um processo termina quando o tempo de processamento é totalmente satisfeito. Para minimizar a possibilidade de sobra de recursos no final da execução dos processos o tempo gasto pelos processos é ajustado durante cada visita, permitindo uma visita a mais ao processador.

- Memória - a memória é freqüentemente um recurso negligenciado nos modelos de simulação de distribuição de carga, e conseqüentemente nos modelos que verificam métricas de carga presentes na literatura. Em sua maioria esses simuladores consideram a memória como sendo um elemento infinito (Eager et al, 1986; Harchol-Balter & Downey, 1997). Isso é aceitável se os algoritmos de distribuição concentram-se no elemento processador como base para a distribuição. Entretanto, como esse é um recurso indispensável para os testes dos índices de carga, uma modelagem cuidadosa considerando esse recurso é necessária. Neste trabalho a memória não é tratada como um recurso, mas sim como uma variável. Cada aplicação (processo ou tarefa) gasta uma certa quantidade de memória que é reduzida da variável Q_m do respectivo recurso escalonado (20% da memória é reservado para o

sistema operacional). O sistema de memória deve interagir com os processos de modo inteligente e simples. O modelo adotado é de alocação estática e não dinâmica, pois a quantidade de memória é alocada ao processo por todo o seu tempo de vida. O espaço de *swap* de memória será considerado infinito, com objetivo de simplificar a modelagem.

- Disco - para o disco são assumidos: arquivos locais e remotos, uma vez que se trata de um sistema distribuído. Entretanto, para os testes efetuados neste trabalho os arquivos são tratados como locais. *Cache* de arquivo também não é considerado neste modelo e assume-se que os tempos de leitura e escrita são os mesmos (Martin, 1994). O desempenho da escrita e leitura no disco também foi medido usando um programa que escreve grande quantidade de informações aleatórias no disco local. O desempenho de leitura em disco foi então testado levando-se em consideração cerca de cinco minutos para ler o arquivo e produzindo cerca de 300 amostras por execução. O *cache* foi embutido entre as fases de leitura e escrita para garantir que o tempo real de transferência foi medido[12].

- Rede - a rede também é mapeada como um recurso que será alocado para a transmissão e recepção de mensagens. O tempo da rede é dado pelo tamanho da mensagem dividido por 80Mb/s, garantida pela interligação dessa rede através de um *switch*. Uma vez que o ambiente considerado é um ambiente distribuído, as aplicações concorrem pela utilização dos recursos de rede. Essa disputa é caracterizada pela realização de leituras e escrita na rede de comunicação. O recurso rede também não é considerado nos diversos modelos de simulação apresentados na literatura, entretanto esse é sem dúvidas um dos recursos mais importantes em um sistema computacional distribuído.

A fila dos recursos opera baseada no primeiro processo a chegar como sendo o primeiro processo a ser atendido (FIFO - First In First Out). Detalhamento dos

[12] Para assegurar a veracidade dos tempos de serviço do disco o desempenho foi medido fazendo-se uso do comando Iostat do Unix (comando que reporta as atividades de disco).

parâmetros e da complexidade do modelo podem ser melhor observados no código da simulação apresentado no apêndice A.

A figura 6.8 apresenta uma visão macro do modelo implementado em redes de fila.

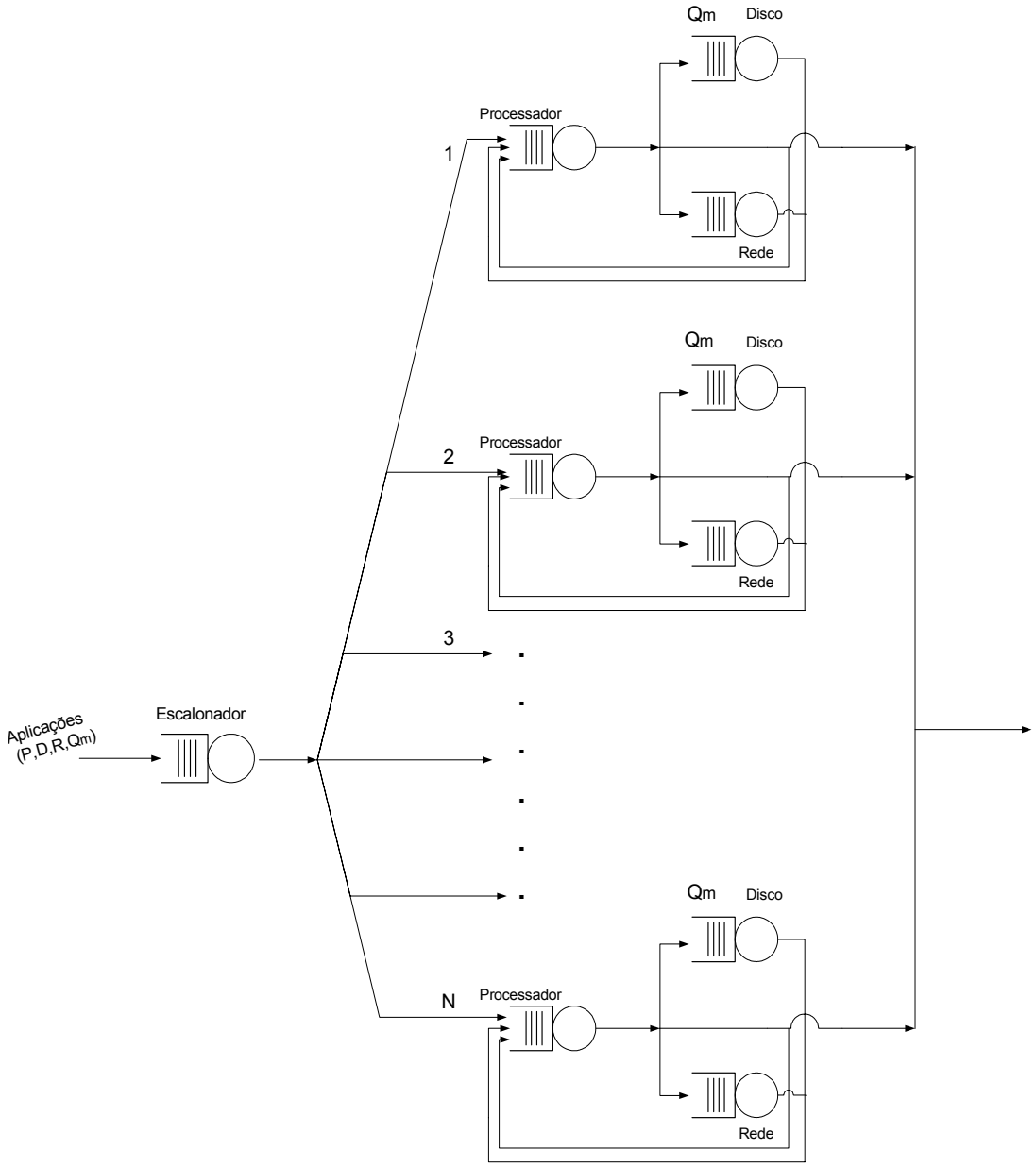

Figura 6.8 - Modelo em Redes de Fila do Simulador de Índices de Carga e Desempenho.

*** Neste diagrama Qm representa a quantidade de memória presente no recurso.**

- Verificação e Validação do Modelo

A fase de verificação consiste em observar se o programa de simulação é uma implementação válida do modelo (MacDougall, 1987). Esta fase responde à seguinte pergunta: “O modelo operacional (programa) representa o modelo conceitual (modelo)?”.

Para pequenos modelos, uma inspeção bem realizada no programa de simulação final pode ser suficiente.

Para grandes modelos é sugerido que a verificação seja contínua, não esperando que todo o programa esteja pronto para ser iniciado. Os meios pelos quais a verificação pode ser realizada são (Banks, 1998): seguir os princípios da programação estruturada (*top-down* ou programação modular); documentar muito bem o programa; mais de uma pessoa analisar o programa; verificar se os valores de entrada estão sendo usados apropriadamente; utilizar um depurador.

Já na fase de validação deve-se demonstrar que o modelo proposto é uma representação do sistema a ser simulado, respondendo a seguinte pergunta: “O modelo conceitual pode substituir o sistema real?” (Banks, 1998). A validação pode ser considerada em dois casos: quando o sistema simulado existe e pode ser medido e quando o sistema modelado não existe, tendo-se apenas o projeto do sistema a ser simulado. No primeiro caso, o objetivo da análise é avaliar uma mudança proposta no sistema e a validação é baseada na comparação dos resultados do modelo com as medidas reais do sistema. No segundo caso, o objetivo da análise é estimar o desempenho do projeto ou mesmo avaliar projetos alternativos. Neste caso, o modelo é validado baseando-se nos resultados esperados no projeto do sistema e cada suposição e abstração são justificadas. Outra possibilidade para avaliação é através de comparações com outros modelos do projeto já validados, como por exemplo, modelos analíticos (MacDougall, 1987), ou através de situações em que os resultados já são conhecidos.

Nesta tese, tanto para a fase de verificação como para a fase de validação foram utilizados os conceitos propostos por Sargent (Sargent, 1999).

Segundo Sargent (Sargent, 1999) a verificação do programa de simulação não é tarefa fácil de ser executada e uma validação completa do modelo em todo o domínio da aplicação é, muitas vezes, impraticável. Como uma alternativa prática,

uma série de testes pode ser executada para demonstrar que os resultados específicos obtidos a partir do modelo são compatíveis com os dados obtidos a partir de um sistema real, levando a um grau razoável de confiança no modelo e simulação.

Os resultados do modelo de escalonamento de processos proposto nesta tese foram validados, através da utilização de diversas formas e diversas técnicas, com ênfase particular nos seguintes parâmetros e saídas:

- Tempo gasto para escalonar uma aplicação;
- Tamanho de fila nos recursos processador, disco e rede;
- Comprimento da fila nos recursos escalonador, processador, disco e rede;
- Tempo final de resposta do modelo;
- Número de acessos ao processador.

As técnicas utilizadas para validação do modelo foram: *1.Face Validity*, *2.Fixed Value* e *3.Internal Validity* e são detalhadas a seguir:

1. *Face Validity*

Esta técnica é utilizada para demonstrar que o relacionamento entre as entradas e saídas é razoável. No modelo de simulação do escalonamento apresentado várias características foram verificadas, destacando-se entre elas:

- Influência da carga de trabalho: quando o número de aplicações aumenta e o número de máquinas permanece constante as filas dos recursos escalonador e processar aumentam e, conseqüentemente, o tempo final de processamento aumenta;
- Potência computacional: quando um processador mais potente é utilizado, o tamanho da fila, conseqüentemente, diminui. O mesmo pode ser verificado para os recursos disco e rede;

Todas as características comparadas levaram a resultados coerentes.

2. *Fixed Value*

Nesta técnica, algumas variáveis são configuradas para causar resultados conhecidos ou de cálculo possível. A título de exemplificação, pode-se efetuar a execução de uma simulação para uma única aplicação, que é utilizada para verificar

a ocorrência de filas vazias. Esta característica foi observada em todos os exemplos simulados.

3. Internal Validity

Várias execuções da simulação com números aleatórios independentes foram efetuadas para se obter cada valor do programa de simulação (uso de 15 sementes diferentes). Esta técnica (MacDougall, 1987) reduz a variância e permite a construção de um intervalo de confiança para medir a precisão obtida. Os resultados obtidos a partir dessas simulações podem também ser usados para demonstrar que a variabilidade é aceitável. A tabela 6.1 apresenta resultados típicos obtidos a partir de um exemplo particular, para diferentes saídas.

Tabela 6.1 – Variabilidade nas saídas da simulação

Saída	Valor máximo	Valor mínimo	Média	Desvio Padrão
Tempo de resposta em uma aplicação CPU-Bound (índice CPU). Plataforma Homogênea.	165,07	156,09	160,22	2,87
Tempo de resposta em uma aplicação CPU-Bound (índice Disco). Plataforma homogênea.	1530,11	1472,86	1493,99	18,19
Tempo de resposta em uma aplicação CPU-Bound (Round-Robin). Plataforma homogênea.	166,62	154,99	161,20	3,51
Tempo de resposta em uma aplicação CPU-Bound (índice CPU). Plataforma parcialmente homogênea.	275,09	253,78	266,85	6,49
Tempo de resposta em uma aplicação CPU-Bound (índice Disco). Plataforma parcialmente homogênea.	1522,27	1447,28	1491,28	20,55
Tempo de resposta em uma aplicação CPU-Bound (Round-Robin). Plataforma heterogênea.	331,53	299,97	314,71	9,24
Tempo de resposta em uma aplicação CPU-Bound (índice CPU). Plataforma heterogênea.	251,77	236,66	242,26	4,59
Tempo de resposta em uma aplicação CPU-Bound (índice Disco). Plataforma heterogênea.	2166,02	2030,42	2089,56	33,30
Tempo de resposta em uma aplicação CPU-Bound (Round-Robin). Plataforma heterogênea.	323,66	256,25	289,72	16,55

Os resultados da tabela 6.1 mostram valores pequenos para o desvio padrão das saídas obtidas com a simulação, o qual demonstra uma pequena dispersão desses dados e uma boa precisão interna, uma vez que grandes dispersões nos resultados da simulação significam que o modelo é inapropriado ou o número de execuções insuficiente.

Os resultados e as conclusões obtidas são apresentados detalhadamente no próximo capítulo.

6.6 Considerações Finais

A utilização de sistemas distribuídos, apesar de apresentar uma solução viável para o problema de custo da computação paralela, traz consigo diversos problemas, destacando-se a heterogeneidade. Assim, quando se deseja explorar esse tipo de sistemas, vários níveis e tipos de heterogeneidade devem ser considerados.

Foi apresentado um índice de desempenho, original, baseado em uma métrica Euclidiana e que busca a obtenção da disponibilidade dos recursos através da relação vetorial existente entre carga e recurso. Adicionalmente, neste capítulo, foram apresentados o programa de simulação de índices de carga e desempenho e considerações sobre o desenvolvimento, verificação e validação de um modelo de escalonamento de processos. Foram utilizadas as Redes de Fila para representar o modelo, com solução através de simulação.

A motivação para a construção do modelo está no fato de que existe uma carência de ferramentas práticas de avaliação desses índices e a correspondente avaliação de desempenho.

As técnicas de modelagem são indicadas para efetuar essa análise exatamente pela não necessidade da presença física do objeto de estudo. A vantagem da simulação sobre as técnicas analíticas reside no fato de que as alterações, que por ventura são impostas ao modelo, podem ser mais facilmente refletidas. Dessa forma, optou-se por realizar o estudo dos índices utilizando técnicas de modelagem e a resolução do modelo por simulação.

A validação do modelo foi efetuada através das técnicas propostas por Sargent (Sargent, 1999).

As grandes vantagens da simulação em relação às outras técnicas de avaliação são a possibilidade de representar a execução do modelo em diferentes plataformas e a capacidade de, através de pequenas alterações no comportamento do modelo, representar diferentes índices de carga. Desse modo, o próximo capítulo apresenta os resultados e a análise estatística dos resultados obtidos a partir do modelo proposto neste capítulo.