

VIP e PVIP: Índices de Desempenho para Avaliação de Sistemas Heterogêneos

Kalinka Regina Lucas Jaquie Castelo Branco
Programa de Pós-Graduação em Ciência da Computação
Centro Universitário Eurípides de Marília
UNIVEM – Marília
Caixa Postal 2041, 17525-901
{kalinka}@fundanet.br

Marcos José Santana
Regina Helena Carlucci Santana
Sarita Mazzini Bruschi
Departamento de Ciências Matemáticas e de Computação
ICMC/USP – São Carlos
Caixa Postal 668, 13560-970
{mjs, rcs, sarita}@icmc.usp.br

and

Célia Leiko Ogawa Kawabata
UNICEP – Centro Universitário Central Paulista
Rua Miguel Petroni, 5111 São Carlos, SP, Brasil
{celiak}@gmail.com

Abstract

An evaluation of two new performance indices (PIV - Performance Index Vector and WPIV – Weighted Performance Index Vector) that are based on a Euclidian metric is presented in this paper. Aiming to maximize the use of the machines, the proposed indices are a combination of several usual indices and the results of their evaluation through a simulator show an appropriate behavior for different kinds of applications.

Keywords: Heterogeneous Computing, Load Index, Distributed Systems, Performance Index.

Resumo

Uma avaliação de dois novos índices de desempenho (VIP – *Vector for Index of Performance* e PVIP - *Ponderated Vector for Index of Performance*) tendo sua base em uma métrica Euclidiana é apresentada neste artigo. Esses índices têm como objetivo maximizar a utilização das máquinas e do sistema como um todo. Os índices propostos são uma combinação de diversos índices de carga encontrados na literatura e os resultados de sua avaliação através de um programa de simulação mostram um comportamento adequado para diferentes tipos de aplicações.

Palavras Chaves: Computação Heterogênea, Índice de Carga, Sistemas Distribuídos, Índice de Desempenho.

1 INTRODUÇÃO

Os sistemas computacionais distribuídos mostraram ao longo dos anos suas vantagens sobre os sistemas centralizados, conquistando em pouco tempo lugar de destaque no cenário computacional. Essa classe de sistemas tem sido cada vez mais aperfeiçoada a fim de prover melhor desempenho a um custo relativamente baixo.

Várias pesquisas vêm sendo desenvolvidas levando-se em consideração os problemas existentes na utilização da computação paralela distribuída. Embora as vantagens obtidas com a computação paralela distribuída sejam evidentes, diversos novos problemas emergem dessa nova abordagem. A maioria dessas pesquisas aborda problemas com os meios de interconexão, com a portabilidade das aplicações, com os protocolos de comunicação, com o escalonamento de processos e suas implicações no desenvolvimento final do sistema, entre outros. O escalonamento de processos constitui um tema de grande importância, principalmente por influenciar diretamente o desempenho do sistema.

Uma nova maneira de se tratar a heterogeneidade dos sistemas computacionais distribuídos é apresentada neste artigo, e leva ao estabelecimento de novos índices de desempenho que podem ser utilizados de forma eficiente em sistemas heterogêneos [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]. Os índices de desempenho aqui propostos baseiam-se em uma métrica Euclidiana e fazem uso de vetores para obtenção de características do estado de carga das máquinas e do sistema. Uma vez que esses índices combinam os diferentes índices de carga presentes na literatura, os resultados por eles apresentados são por deveras melhores que os resultados apresentados por cada índice de carga individualmente.

A finalidade é contribuir com o desenvolvimento das pesquisas tanto em escalonamento de processos quanto em computação heterogênea, fazendo uso dos índices de carga já consolidados e das características individuais da cada máquina ou sistema a ser avaliado.

Desse modo, este artigo está estruturado da seguinte forma: a seção 2 faz uma análise dos trabalhos correlatos, efetuando a definição de índices de carga para ambientes homogêneos e apresentando os diversos modelos de índices de carga presentes na literatura. A seção 3 descreve os índices propostos neste artigo, os quais são comparados com os índices de carga encontrados na literatura. A seção 4 apresenta a ferramenta desenvolvida para análise de índices de carga e dos índices propostos. Na seção 5 encontram-se os teste e resultados obtidos com a simulação e a seção 6 conclui o artigo.

2 TRABALHOS RELACIONADOS

A referência até hoje utilizada nesta área é [1]. Embora muitas pesquisas são realizadas visando o escalonamento de processos e fazendo uso de índices de carga para melhor obtenção de desempenho, pouco tem sido efetivamente realizado no âmbito de escalonamento de processos em sistemas arquiteturais e configuracionalmente heterogêneos.

Os autores propõem a utilização de um índice de carga obtido através da combinação linear do tempo de serviço (s_j) requerido por uma tarefa para sua execução em um determinado recurso (r_j), sendo que o comprimento da fila do recurso (r_j) é dado por (q_j), de tal modo que o índice de carga (li = load index) é obtido através de:

$$li = \sum_{j=1}^N s_j \times q_j \quad (\text{Equação 1})$$

onde N é o número total de recursos que possuem filas. Esse modelo contempla ambientes compostos por máquinas configuracionalmente e arquiteturalmente homogêneas. O trabalho de [13] propõe a utilização da Capacidade de Carga como sendo um índice de carga para ambientes heterogêneos. Por capacidade de carga entende-se a utilização efetiva do processador. É considerado um índice de carga para ambientes heterogêneos por efetuar a normalização da velocidade de cada CPU em relação às demais.

$$(1 - \text{Utilização_de_CPU}) \times \text{Velocidade_Relativa_da_CPU}$$

Essa métrica representa a capacidade de processamento efetivamente restante no recurso processador, entretanto não leva em conta os demais recursos que estão envolvidos no processamento como um todo, como, por exemplo, memória, disco, rede, o que o torna um índice muito específico e pouco flexível. Em adição às considerações tecidas com relação a esse índice, deve ser salientado que os experimentos com os índices de carga por eles efetuados (a execução de aplicações para obtenção do tempo final de resposta) foram executados em máquinas arquiteturalmente homogêneas.

Por outro lado, [14] propõem a obtenção da carga como sendo o número de itens de dados existente na fila do processador. A função de carga é denotada por w . A carga total do sistema é dada por W , sendo esta obtida através de:

$$W = \sum_{i=1}^{P-1} w[i] \quad (\text{Equação 2})$$

Considerando um conjunto de P processadores, sendo esses completamente homogêneos e considerando as tarefas a serem executadas nesses processadores também homogêneas, tem-se que a média total nesses sistemas é dada por W/P e denotada por w . De forma geral, constata-se que as pesquisas na área ainda se encontram em andamento, especialmente nos quesitos de obtenção de índices de carga e modelos de simulação para averiguação de desempenho.

Os modelos de simulação utilizados na maioria dos casos, onde se efetua o estudo da implicação dos índices de carga no resultado final das aplicações, constituem modelos simplificados e que não levam em conta, até o presente momento, características de heterogeneidade da plataforma [2] [13] [14], contemplando grande maioria de parâmetros relativos

somente a máquina e aplicações homogêneas. Dessa maneira, índices que contemplem a heterogeneidade tanto configuracional quanto arquitetural do sistema fazem-se necessários.

3 UMA NOVA MÉTRICA PARA ESCALONAMENTO DE PROCESSOS EM AMBIENTES HETEROGÊNEOS

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 sobre um intervalo de tempo. Os modelos para obtenção de carga, em sua grande maioria, levam em conta características específicas de alguns recursos.

Um bom índice de desempenho, assim como nos 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.

Dessa maneira, levando-se em consideração os quatro recursos básicos de uma máquina, pode-se então obter a função apresentada na equação 3:

$$ID = f(I_{CPU}, I_{Memoria}, I_{Disco}, I_{Rede}) \text{ (Equação 3)}$$

A equação 3 pode fazer uso de pesos para cada um dos índices específicos dos recursos:

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 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. $I_{Memoria}$ 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.

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

Cada índice de carga é calculado independentemente e leva em consideração um benchmark específico. Uma vez que as medidas são apresentadas com valores que não podem ser diretamente combinados e comparados uma normalização deve ser utilizada. 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 de vários índices de carga, contemplando várias visões do uso do recurso).

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 [20].

3.1 VIP - Vector for Index of Performance

Uma característica da função apresentada na equação 3 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 uma 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 1 apresenta um espaço tridimensional, onde podem ainda 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.

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 $\theta \approx 45^\circ$, ambos recursos estão igualmente carregados; $\theta \gg 45^\circ$ indica que o recurso 1 é predominante e $\theta \ll 45^\circ$ indica que o recurso 2 é o predominante.

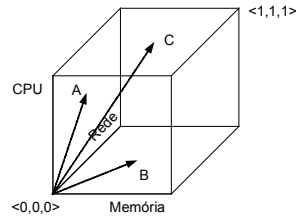


Figura 1 - Espaço tri dimensional usado para descrever a carga atual de uma máquina e os três pontos indicando cargas potenciais da máquina.

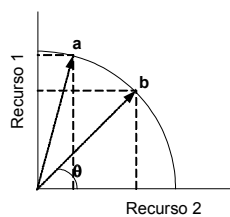


Figura 2 - Áreas diferentes para dois vetores com mesmo comprimento (ângulos distintos)

Uma vez que o comprimento dos vetores é o mesmo, apesar da máquina b estar balanceada ($|45^\circ - \theta| = 0$) 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 2 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 utiliza mais de um recurso em duas máquinas distintas mas que estão pode ser alocado em M_1 e M_2 que estão igualmente carregadas. 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 3 pode ser reescrita como:

$$ID = \sqrt{I_{Cpu}^2 + I_{Disco}^2 + I_{Memória}^2 + I_{Rede}^2}$$

Para o exemplo apresentado na figura 3, 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.

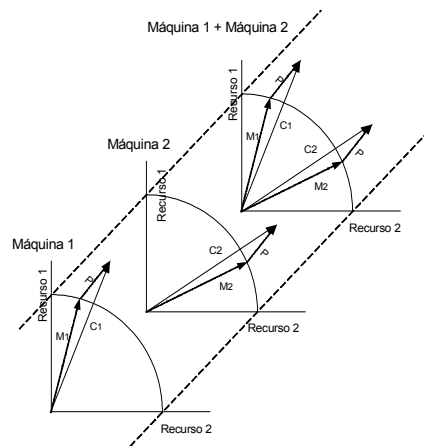


Figura 3 - Espaço bidimensional formado pelos recursos 1 e 2, e duas máquinas com cargas iguais (processo limitado por dois recursos).

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 de uma maneira mais equilibrada.

Desta forma, o índice de desempenho apresentado neste artigo 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, baseado 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 3 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.

4 METODOLOGIA PARA O DESENVOLVIMENTO DA AVALIAÇÃO DE DESEMPENHO - TÉCNICAS DE MODELAGEM

Visando a avaliação do comportamento dos diversos índices de carga encontrados na literatura em comparação com os índices propostos VIP e PVIP foi desenvolvido um programa de simulação que representará o comportamento de um sistema distribuído heterogêneo onde aplicações paralelas são executadas. O programa também avalia a sobrecarga causada pelo uso impróprio desses índices quando da submissão de tipos específicos de classes de aplicações.

4.1 Definição do modelo

Um primeiro passo para o desenvolvimento da simulação é a abstração das principais características do sistema a ser simulado em um modelo. Para a modelagem do sistema em estudo, foi utilizada a técnica de Redes de Filas [15]. O modelo é composto por um centro de serviço (servidor + fila) que representa um escalonador (responsável unicamente pela atividade de escalonar os processos) e vários conjuntos de 3 centros de serviço (processador, disco e rede) representando as máquinas onde as aplicações são executadas (figura 4). Essas máquinas, em conjunto serão consideradas como uma máquina paralela virtual.

A memória não foi modelada como uma fornecedora de serviços, sendo somente uma variável quantitativa (Q_m) definida em cada máquina. Ao escalonador estão associados tempos de escalonamento e tempo de atualização da tabela dos índices nas demais máquinas, dependendo do tipo de escalonamento a ser realizado. Os tempos de atualização da tabela foram obtidos a partir dos experimentos realizados fazendo uso do ambiente paralelo virtual PVM [16] e do ambiente de escalonamento AMIGO (*dynAMical flexIble schedulinG environment*) [17].

As aplicações que chegam ao escalonador pertencem a uma determinada classe, a qual define as aplicações em termos de porcentagem de utilização de processamento, memória, disco e rede. A função do escalonador é escolher a máquina onde a aplicação será executada de acordo com algum índice de carga. A máquina escolhida para a execução recebe a aplicação no elemento processador e atualiza sua tabela de índices de acordo com a classe que a aplicação pertence.

No elemento processador, a aplicação fica o tempo necessário para processamento até que um outro recurso seja requisitado (rede ou disco) ou até que o quantum expire. Os processos se movem pelos recursos do sistema e retornam para o fim da fila do processador até que seja finalizado.

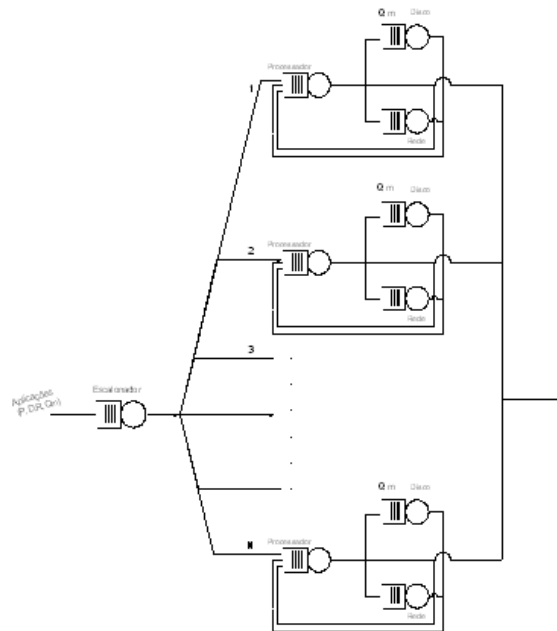


Figura 4 Modelo em redes de filas de um simulador de índice de carga e desempenho.

4.2 Parâmetros do modelo

Para finalizar a definição do modelo é necessário definir os parâmetros a serem utilizados nos centros de serviço. Os parâmetros e seus significados podem ser observados a seguir:

1. Número de máquinas: Número de máquinas a serem utilizadas no cluster.
2. Taxa de chegada: Média da distribuição exponencial que determina a frequência com que as aplicações chegam para serem executadas.
3. Taxa de serviço do escalonador: Média da distribuição exponencial que determina o tempo que o escalonador irá demorar a decidir sobre a melhor máquina para executar a aplicação.
4. Tempo de cálculo dos índices: Média da distribuição exponencial que determina o tempo necessário para atualização dos índices de desempenho.
5. Tempo de atualização dos índices: Periodicidade de atualização da tabela de índices.
6. Tempo de atualização do índice proposto: Média da distribuição exponencial que determina o tempo necessário para que os vetores utilizados no índice de desempenho sejam atualizados.
7. Tempo de serviço dos processadores: Média da distribuição exponencial que determina o tempo que cada aplicação utilizará em cada processador do cluster.
8. Tempo de serviço dos discos: Média da distribuição exponencial que determina o tempo que cada aplicação utilizará no período de acesso a disco, em cada disco do cluster.
9. Tempo de serviço das interfaces de rede: Média da distribuição exponencial que determina o tempo que cada aplicação utilizará no período de utilização da rede, em cada interface de rede do cluster.
10. Quantidade de Memória: Quantidade de memória (Mbytes) em cada máquina do cluster.
11. Quantum: Período que cada aplicação pode ficar sendo executada nos processadores, em cada máquina do cluster.
12. Utilização processamento: Característica da aplicação em termos de porcentagem de utilização de processamento.
13. Utilização disco: Característica da aplicação em termos de porcentagem de utilização de disco.
14. Utilização rede: Característica da aplicação em termos de porcentagem de utilização de rede.
15. Utilização memória: Característica da aplicação em termos de porcentagem de utilização de memória.
16. Índice: Escolha do índice a ser analisado.

Os elementos de processamento possuem tempos de serviço relativos uns aos outros, medidos de acordo com a capacidade de cada um. O elemento disco será definido por $t = \text{tempo de seek} + \text{o tamanho do arquivo/bandwidth}$ [14]. 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 [15]).

O resultado fornecido pelo programa de simulação é o tempo que a aplicação levará para ser executada na máquina paralela virtual.

Com essas especificações pode-se realizar experimentações com diferentes tipos de configurações do cluster (alterando-se os parâmetros 1, 3, 7, 8, 9 e 10), das aplicações (alterando-se os parâmetros 2, 12, 13, 14 e 15) e índices (alterando-se os parâmetros 4, 5, 6 e 16).

O código do programa de simulação foi desenvolvido utilizando a biblioteca SMPL [16].

4.3 Validação do modelo

Tanto a validação quanto a verificação do modelo foi efetuada através das técnicas propostas por Sargent [18].

Segundo Sargent [18] a verificação do programa de simulação não é uma tarefa fácil de ser executada e a 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 neste artigo foram validados, através da utilização de diversas formas e 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.

Os resultados apresentados na tabela 5 apresentam valores do desvio padrão baseado nas saídas obtidas com a simulação, demonstrando a baixa dispersão dos dados e a boa precisão, uma vez que resultados de dispersão altos indicam que o modelo de simulação é inadequado ou que o número de execuções é insuficiente.

Tabela 5 – Variabilidade das saídas da simulação

Output	Maximum value	Minimum value	Mean value	Standard deviation
Response time in a CPU-Bound application (CPU index). Homogeneous platform.	165.07	156.09	160.22	2.87
Response time in a CPU-Bound application (Disk index). Homogeneous platform.	1530.11	1472.86	1493.99	18.19
Response time in a CPU-Bound application (Round-Robin). Homogeneous platform.	166.62	154.99	161.20	3.51
Response time in a CPU-Bound application (CPU index). Partially homogeneous platform.	275.09	253.78	266.85	6.49
Response time in a CPU-Bound application (Disk index). Partially homogeneous platform.	1522.27	1447.28	1491.28	20.55
Response time in a CPU-Bound application (Round-Robin). Heterogeneous platform.	331.53	299.97	314.71	9.24
Response time in a CPU-Bound application (CPU index). Heterogeneous platform.	251.77	236.66	242.26	4.59
Response time in a CPU-Bound application (Disk index). Heterogeneous platform.	2166.02	2030.42	2089.56	33.30
Response time in a CPU-Bound application (Round-Robin). Heterogeneous platform.	323.66	256.25	289.72	16.55

5 RESULTADOS OBTIDOS

Os resultados apresentados nas tabelas demonstram a viabilidade do uso do índice de desempenho proposto neste artigo, uma vez que os tempos médios de resposta, quando do seu uso, nos três tipos de plataformas avaliados, é sempre melhor quando comparado aos índices tradicionais, excetuando-se os índices específicos para cada aplicação.

A seguir é apresentado o gráfico (figuras 5), para facilitar a visualização do comportamento dos vários tipos de aplicações quando submetidas ao escalonamento fazendo uso de diversos índices de carga.

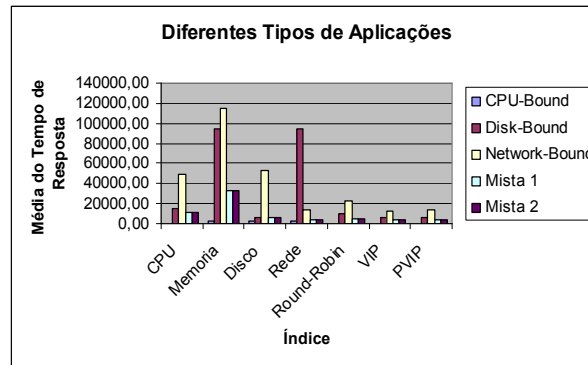


Figura 5 – Resumo do comportamento de diferentes tipos de aplicações quando submetida aos vários índices de carga tradicionais e aos índices propostos neste artigo.

Pode-se notar, através da observação dos resultados apresentados no gráfico, que em todos os casos o índice de desempenho fornece resultados melhores que os apresentados pelos outros índices individualmente, excetuando-se o índice particular apropriado para o tipo de aplicação. Entretanto, quando submetidas aplicações mistas, que exploram diversos recursos, o comportamento do índice desempenho é visivelmente melhor que os demais índices individuais.

Os resultados analisados e apresentados vêm também ao encontro dos resultados encontrados na literatura que indicam que os índices de carga genéricos, além de apresentarem uma tendência à sobrecarga maior, não devem apresentar a mesma qualidade de representação de carga de trabalho, quando comparados com os índices específicos utilizados corretamente [1] [19]. Entretanto, o índice de desempenho proposto, apesar de genérico, apresenta características flexíveis, o que o torna muito próximo muito dos índices específicos de cada aplicação, além de apresentar resultados muito bons quando submetido a aplicações mistas.

Os gráficos apresentados referem-se aos da configuração onde as máquinas são heterogêneas, entretanto, os resultados apresentados para essa configuração são extensíveis às demais configurações, e só não são aqui reproduzidos para não sobrecarregar o texto.

Os resultados apresentados anteriormente refletem as médias das execuções do programa de simulação com diferentes sementes de números aleatórios. A simulação foi desenvolvida para executar 5000 aplicações, independentemente do tipo de aplicação considerada. Dessa forma, aplicações do tipo CPU-Bound vão finalizar em um tempo real de simulação muito menor que aplicações Disk-Bound ou Network-Bound.

Essa abordagem foi adotada por proporcionar resultados mais adequados, uma vez que sempre será considerado o mesmo número de aplicações.

Por outro lado, como os tipos de aplicações são distintos, os tempos finais são diferentes e não podem ser diretamente comparados. Desta forma, para viabilizar a comparação, optou-se por normalizar os valores obtidos, baseando-se no índice que gerou o menor tempo de resposta.

A avaliação dos resultados obtidos está dividida em duas fases:

I – Comparação entre os índices tradicionais;

II – Comparação entre os índices propostos neste trabalho e os índices tradicionais.

As tabelas 6 e 7 apresentam os resultados normalizados para os índices tradicionais. Observa-se na tabela 6, por exemplo, que em sistemas homogêneos, para uma aplicação CPU-Bound, o melhor índice é o de CPU. Se um índice de memória, rede ou disco for utilizado, tem-se um tempo de resposta 9,32 vezes maior que o obtido para o índice de cpu.

As aplicações envolvidas no processo são uniformemente distribuídas entre os recursos.

Tabela 6 – Tabela normalizada pelo melhor índice de carga por tipo de aplicação em plataforma homogênea

Aplicações						
Índices	CPU-Bound	Disk-Bound	Network-Bound	Mista 1	Mista 2	Média
CPU	1,00	1,03	1,01	1,01	1,02	1,01
Memória	9,32	9,42	9,38	9,48	9,31	9,38
Disco	9,32	1,00	1,04	1,02	1,00	2,68
Rede	9,32	9,42	1,00	1,00	1,01	4,35
Round-Robin	1,01	1,01	1,01	1,00	1,01	1,01

Analisando-se os resultados das tabelas 6 e 7, observa-se que o melhor índice é sempre o específico da aplicação. O problema encontrado é saber a priori o tipo da aplicação e que para sistemas homogêneos, quando não se têm informações sobre a aplicação pode-se utilizar tanto o índice de cpu quanto o *round-robin* que os resultados serão praticamente iguais.

Tabela 7 – Tabela normalizada pelo melhor índice de carga em plataforma heterogênea

Aplicações						
Índices	CPU-Bound	Disk-Bound	Network-Bound	Mista 1	Mista 2	Média
CPU	1,00	2,32	3,68	2,98	2,94	2,59
Memória	8,63	14,28	8,60	9,04	8,93	9,90
Disco	8,63	1,00	3,99	1,90	1,87	3,48
Rede	8,63	14,28	1,00	1,00	1,00	5,18
Round-Robin	1,20	1,48	1,71	1,44	1,40	1,45

Para sistemas heterogêneos, a melhor opção, quando não se conhece o tipo de aplicação é a utilização dos índices de CPU e escalonamento *Round-Robin*. No entanto, nesses casos, existem variações que dependem do grau de heterogeneidade do sistema e do tipo de aplicação. Observa-se que no caso do round-robin obtêm-se valores mais estáveis.

Na grande maioria dos trabalhos apresentados na literatura, o índice de CPU é utilizado para qualquer tipo de aplicação. Considerando-se os resultados apresentados pode-se comparar a utilização do índice de CPU com a abordagem Round-robin.

Assim, considerando-se que em um conjunto de aplicações tem-se:

x aplicações CPU-Bound

(1-x) aplicações não CPU-Bound

e que os tempos para executar as aplicações são: t_1 = tempo para executar aplicações CPU-Bound com índice de carga de CPU; t_2 = tempo para executar aplicações não CPU-Bound com índice de carga de CPU; t_{RR} = tempo médio para executar qualquer tipo de aplicação utilizando round-robin

tem-se que: o tempo para executar x aplicações CPU = $x*t_1$; o tempo para executar (1-x) aplicações não CPU = $(x-1) * t_2$; TRR = tempo para executar todas as aplicações com round-robin = t_{rr} ; $TCPU$ = tempo para executar todas as aplicações com índice de cpu = $x*t_1 + (1-x)*t_2$

assim, quando $TCPU < TRR$ deve-se utilizar o índice de CPU e quando $TCPU > TRR$ deve-se utilizar o round-robin.

Portanto, $x t_1 + (1 - x) t_2 < T_{RR}$ Equação 4

$$x < \frac{T_{RR} - T_2}{T_1 - T_2}$$

resolvendo a equação tem-se: de modo que o índice de CPU é mais adequado que o round-robin.

Para os casos homogêneos e parcialmente heterogêneos, o índice de CPU é claramente mais adequado.

Para o caso heterogêneo, aplicando-se a equação 4, tem-se que: $x > 0,77$

isto é, se mais de 77% das aplicações forem do tipo CPU-Bound a utilização do índice de CPU é mais apropriada para que não ocorra perda de desempenho, caso contrário o round-robin pode ser utilizado sem que ocorra prejuízo.

Por outro lado, resultados diferentes podem ser observados quando se faz uso do índice proposto - VIP.

5.1 Comparação entre os índices de desempenho e os índices tradicionais

As tabelas 8 e 9 apresentam os resultados normalizados para os índices tradicionais. Observa-se na tabela 5.6, por exemplo, que em sistemas homogêneos, para uma aplicação CPU-Bound, o melhor índice, novamente, é o de CPU. Se um índice de memória, rede ou disco for utilizado, tem-se um tempo de resposta 9,37 vezes maior que o obtido para o índice de cpu. As aplicações envolvidas no processo são uniformemente distribuídas entre os recursos.

Tabela 8 – Tabela normalizada pelo melhor índice de carga por tipo de aplicação em plataforma homogênea

Aplicações						
Índices	CPU-Bound	Disk-Bound	Network-Bound	Mista 1	Mista 2	Média
CPU	1,00	1,04	1,02	1,02	1,02	1,02
Memória	9,37	9,52	9,44	9,51	9,36	9,44
Disco	9,37	1,01	1,05	1,02	1,00	2,69
Rede	9,37	9,52	1,01	1,00	1,01	4,38
Round-Robin	1,01	1,02	1,02	1,00	1,01	1,01
VIP	1,00	1,00	1,00	1,00	1,00	1,00

Tabela 9 – Tabela normalizada pelo melhor índice de carga por tipo de aplicação em plataforma heterogênea

Aplicações						
Índices	CPU-Bound	Disk-Bound	Network-Bound	Mista 1	Mista 2	Média
CPU	1,14	2,32	3,73	3,00	2,98	2,64
Memória	9,82	14,28	8,72	9,11	9,07	10,20
Disco	9,82	1,00	4,04	1,92	1,90	3,74
Rede	9,82	14,28	1,01	1,01	1,02	5,43
Round-Robin	1,36	1,48	1,74	1,46	1,42	1,49
VIP	1,00	1,00	1,00	1,00	1,00	1,00

De modo análogo, analisando-se os resultados das tabelas 8 e 9, observa-se que para sistemas homogêneos, parcialmente heterogêneos e heterogêneos, a melhor opção, quando não se conhece o tipo de aplicação é a utilização do índice proposto VIP. Principalmente quando o tipo de aplicação é misto, o uso do VIP torna-se ainda mais atraente em plataformas heterogêneas.

Para o caso heterogêneo, aplicando-se a equação 4, tem-se que $x > 1$, isto é, em 100% dos casos deve-se fazer uso do índice VPI para que não ocorra perda de desempenho.

A partir dessa análise, pode-se observar que, excetuando-se a plataforma homogênea, o índice de desempenho apresenta-se muito melhor em termos de uso quando do não conhecimento da plataforma utilizada, bem como do desconhecimento do tipo de aplicação submetida ao escalonamento.

O gráfico apresentado na figura 6 ilustra a interação existente entre os fatores.

Pode-se observar no gráfico que para os índices de carga 5 e 6 o comportamento das aplicações é mais estável do que quando utilizados os demais índices de carga.

Isso demonstra que para esses índices praticamente não existe influência no uso desses para os vários tipos de aplicações, de modo que podem ser utilizados indiscriminadamente.

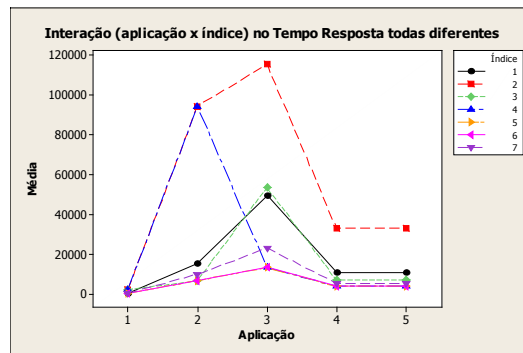


Figura 6 – Relação entre os fatores envolvidos na análise de variância

Na legenda apresentada, os índices correspondem aos: 1- cpu, 2- memória, 3- disco, 4- rede, 5 VIP, 6- PVIP e 7-Round-robin.

Tabela 10 – Análise de variância

Análise de Variância (ANOVA)					
Fontes de Variação (Variável Independente)	SS	Graus de Liberdade	MS	Estatística F	Nível de Significância p
Aplicação	1,21158E+11	4	3,02894E+10	64801,56	0,000
Índice	1,37804E+11	6	2,29673E+10	49136,60	0,000
Aplicação X Índice	1,64489E+11	24	6,85371E+09	14662,93	0,000

Pode-se então, a partir dos resultados apresentados na tabela 10, responder a Hipótese 1, a qual aceita a hipótese (uma vez que $P < 0,05$) que os vários índices de carga e os diferentes tipos de aplicações influenciam o tempo de resposta final do sistema.

6 CONCLUSÃO

O novo índice de desempenho proposto foi utilizado com diversos tipos de aplicações, várias comparações foram efetuadas apresentando um aumento de desempenho, o que demonstrou que a escolha do índice de carga influencia na qualidade das operações de escalonamento de processos, mais especificamente no balanceamento de cargas.

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. As 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.

Partindo da confecção do modelo em redes de fila para escalonamento de processos apresentado no capítulo 6, pode-se testar não somente o índice de desempenho, mas também os demais índices de carga existentes na literatura, demonstrando a usabilidade e flexibilidade desse novo índice principalmente em plataformas heterogêneas.

Os resultados aqui obtidos demonstram também a necessidade de se levar em consideração todos os recursos envolvidos no processo de escalonamento de processos para que decisões mais acertadas possam ser tomadas, independentemente do objetivo que se queira alcançar com o escalonamento, principalmente quando o tipo de aplicação faz uso dos diversos tipos de recursos ou quando não se sabe explicitamente o tipo de aplicação que será escalonada.

A avaliação realizada, embora parcial, demonstra o potencial existente do novo índice proposto. Empregado de forma correta o índice de desempenho pode melhorar o desempenho de um escalonador de maneira significativa.

Como trabalhos futuros estima-se efetuar a avaliação desses índices em ambientes corporativos e de realidade aumentada.

7 AGRADecIMENTOS

Os autores gostariam de agradecer às agências de financiamento CAPES, CNPq e FAPESP pelo apoio dado aos projetos do Laboratório de Sistemas Distribuídos e Programação Concorrente (LaSDPC) do ICMC-USP.

8 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Ferrari, D.; Zhou, S. (1987). An Empirical Investigation of Load Indices for Load Balancing Applications. In Proceedings of Performance'87, the 12th Int'l Symposium on Computer Performance Modeling, Measurement, and Evaluation, p.515-528.
- [2] Kunz, T. (1991). The Influence of Different Workload Descriptions on a Heuristic Load Balancing Scheme. IEEE Transactions on Software Engineering, v.17, n.7, p.725-730, July.
- [3] Ambrosius, S. L.; Freund, R. F.; Scott, S. L.; Siegel, H. J. (1996). Work-Based Performance Measurement and Analysis of Virtual Heterogeneous Machines. In the 5th Heterogeneous Computing Workshop (HCW'96). April.
- [4] Ekemeccic, I; Tartalja, I.; Milutinovic, V. (1996). A Survey of Heterogeneous Computing: Concepts and Systems. Proceedings of IEEE, 84: 1127-1144.
- [5] Eshaghian, M.M.(1996). Heterogeneous Computing. Artech House, Norwood, MA.
- [6] El-Rewini, H.; Ali, H. H.; Lewis, T. (1995). Task Scheduling in Multiprocessing Systems. IEEE Computer, p. 27-37, december.
- [7] Freund, R.F. (1989). Optimal selection theory for superconcurrency. Proceedings of Supercomputing '89, November, pp. 699-703.
- [8] Freund, R.; Conwell, D. (1990). Superconcurrency: A Form of Distributed Heterogeneous Supercomputing. Supercomputing Review, October, pp. 47-50.
- [9] Geist, A.; Beguelin, A.; Dongarra, J.; Jiang, W.; Manchek, R.; Sunderam, V. (1994). PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing, The MIT Press, Londres.
- [10] Kafil, M.; Ahmad, I. (1997). Optimal Task Assignment in Heterogeneous Computing Systems. 6th Heterogeneous Computing Workshop (HCW'97). April, pp. 135-146.
- [11] Khokhar, A. A.; Prasanna, V.K.; Shaaban, M.E.; Wang, C.L. (1993). Heterogeneous Computing: Challenges and Opportunities. IEEE Computer, 26(6): 18-27, June.
- [12] Braun, T. D; Siegel, H. J.; Beck, N.; Boloni, L.; Maheswaran, M. ; Reuther, A. I.; Robertson, J. P.; Theys, M. D.; Yao, B (1998). A Taxonomy for Describing Matching and Scheduling Heuristics for Mixed-Machine Heterogeneous Computing Systems. IEEE Workshop on Advances in Parallel and Distributed Systems. October, pp. 330-335.
- [13] Wolffe, G. S.; Hosseini, S. H.; Vairavan, K. (1997). An Experimental Study of Workload Indices for Non-dedicated, Heterogeneous Systems. In the proceedings of PDPTA'97, v.1, p. 470-478.
- [14] Fontlupt, C.; Marquet, P.; Dekeyser, J. (1998). Data Parallel Load Balancing Strategies. Parallel Computing, 24 p. 1665-1684.
- [15] Jain, R. (1991). The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. John Wiley & Sons.
- [16] Sunderam, V. S, Geist, A., Dongarra, J., Monchek, R. (1994). The PVM Concurrent Computing System: Evolution, Experiences and Trends, Parallel Computing, v. 20, p. 531-545.
- [17] Souza, P. S. L. (2000). AMIGO: Uma Contribuição para a Convergência na Área de Escalonamento de Processos. Phd Thesis (Doctorate in "Applied Physics – option: Computational Physics" Sciences) - São Carlos Institute of Physics, USP – University of São Paulo.
- [18] Sargent, R. G. (1999) Validation and Verification of Simulation Models. In Proceedings of the 1999 Winter Simulation Conference, p. 39 – 48.
- [19] Mehra, P.; Wah, B.W. (1993). Automated Learning of Load-Balancing Strategies for a Distributed Computer System. University of Illinois at Urbana-Champaign.
- [20] Branco, K. R. L.J. C. (2004). Índices de Carga e Desempenho em Ambientes Paralelos/Distribuídos – Modelagem e Métricas. Phd Thesis (Doctorate in Computer Sciences and Computational Mathematics) – Institute of Mathematical and Computational Sciences –São Carlos campus. USP – University of São Paulo.