

# Avaliação de Desempenho dos Componentes de Políticas de Escalonamento de Processos em Ambientes Distribuídos

Paulo Ricardo Paz Vital e Ricardo Ribeiro dos Santos  
Universidade Católica Dom Bosco – Departamento de Engenharia de Computação  
Av. Tamandaré 6000, CEP 790117-900, Campo Grande, MS, Brasil  
{pvital, rics}@ec.ucdb.br

## Resumo

*Este artigo apresenta os resultados de uma avaliação de desempenho de políticas de escalonamento de processos adotadas no ambiente de escalonamento AMIGO. Nesta avaliação de desempenho foram utilizadas três políticas que adotam diferentes métricas de escalonamento. O tempo de execução dos componentes existentes em cada política foi considerado para avaliar o desempenho.*

## 1. Introdução

Para que processos sejam alocados entre os computadores de um sistema distribuído, deve ser adotado algum algoritmo para a distribuição. A aplicação destes algoritmos é comumente conhecida como **escalonamento de processos**. Essa atividade possui grande importância para sistemas distribuídos e paralelos, uma vez que a otimização do sistema e um melhor desempenho estão totalmente relacionadas com o uso adequado da capacidade de processamento instalada.

Assim, este artigo aborda a problemática da atividade de escalonar processos através dos resultados obtidos de uma avaliação de desempenho dos componentes de políticas de escalonamento utilizadas no ambiente de escalonamento AMIGO (*dynAMical flexIble scheduling environment*) desenvolvido por [5]. Os resultados obtidos neste trabalho permitem indicar quais das políticas inseridas no ambiente possuem o melhor desempenho e os fatores que influenciam na sobrecarga dos componentes das políticas de escalonamento.

## 2. Escalonamento de Processos

O escalonamento de processos é a atividade responsável pela distribuição de processos aos elemen-

tos de processamento<sup>1</sup> para que sejam executados. Ele pode ser de dois tipos, o escalonamento realizado utilizando somente um computador (escalonamento local) e o realizado utilizando diversos computadores (escalonamento global).

O desenvolvimento de um algoritmo de escalonamento de processos, também chamado de política de escalonamento, pode ser realizado de diversas maneiras. Uma delas consiste em dividi-lo em partes menores chamadas de **componentes**, que nada mais são que módulos responsáveis por uma fatia da atividade de “escalonar processos”. Segundo [4], um algoritmo de escalonamento global possui quatro componentes, sendo eles:

- **Componente de Transferência:** é o componente responsável por determinar se um nó está em um estado adequado para participar das transferências de tarefas, ou como emissor, ou como receptor, indicando-o como tal;
- **Componente de Seleção:** é responsável por selecionar qual tarefa vai ser transferida após a política de transferência ter determinado qual nó será o emissor;
- **Componente de Localização:** é responsável por achar um “parceiro de transferência” adequado para um nó;
- **Componente de Informação:** é quem decide quando as informações sobre os estados referentes aos outros nós do sistema podem ser coletados, de onde podem ser coletadas, e qual informação é coletada.

O escalonamento de processos necessita de certas políticas de escalonamento para realizar a tarefa de

---

<sup>1</sup> Elemento de processamento, estação de trabalho, máquina e computador, são sinônimos que indicam um *hardware* composto por, no mínimo, um processador com acesso à memória

distribuir os processos entre os elementos de processamento. Essas políticas têm por objetivo manter o sistema computacional com volume de trabalho proporcional a capacidade.

### 3. O Ambiente AMIGO

O AMIGO [5] é um ambiente de escalonamento que reúne diferentes políticas de escalonamento de processos onde o usuário tem a possibilidade de escolher qual destas políticas é mais apropriada para suas necessidades.

Com o objetivo de flexibilizar não só a política de escalonamento, mas tudo que pode influenciá-lo, o AMIGO permite que o usuário selecione de acordo com suas necessidades: as classes de *software*, a plataforma de hardware e outras configurações que devem ser conhecidas para que o escalonamento de uma aplicação seja possível.

O fornecimento de transparência para o usuário é a maior preocupação do AMIGO pois, se um ambiente de escalonamento que se diz flexível não possuir transparência, implica em o próprio usuário ter que escolher qual a melhor política de escalonamento a ser utilizada para determinada aplicação. Além de ser flexível, o AMIGO é dinâmico. O dinamismo do ambiente possibilita a mudança de política e/ou mecanismos de escalonamento em tempo de execução.

Inicialmente o AMIGO foi desenvolvido para aplicações paralelas em plataformas distribuídas, hete-rôgeneas ou não, com sistema operacional LINUX (ou baseados em sistemas UNIX) e gerenciadas pelo ambientes de passagem de mensagem PVM (*Parallel Virtual Machine*) [2] ou MPI (*Message Passing Interface*) [3].

A estrutura do AMIGO é organizada em duas camadas, uma superior e outra inferior. Essa divisão permite que a atividade de escalonar processos, que ocorre na camada inferior, seja totalmente independente da configuração e do monitoramento, ocorridos na camada superior. Além disso, a separação proporciona maior modularidade ao ambiente e facilita a portabilidade para outras plataformas.

As políticas de escalonamento compõem a camada inferior do ambiente AMIGO. De maneira mais específica, essa camada é a responsável por:

1. receber as requisições de escalonamento a partir de um ambiente de passagem de mensagens;
2. repassar essas requisições para uma política de escalonamento ativa;
3. repassar as informações retornadas pela política para o ambiente de passagem de mensagens.

### 4. Avaliação de Desempenho e Resultados Obtidos

Os experimentos para a avaliação de desempenho foram realizados em um ambiente distribuído composto por oito máquinas, todas elas com o sistema operacional RedHat Linux 7.2, o ambiente de escalonamento AMIGO e uma versão do ambiente de passagem de mensagens PVM adaptado para o funcionamento com o AMIGO.

Em um momento anterior ao início dos experimentos, foi realizada a instrumentação das políticas de escalonamento DPWP [1], MinMax [6] e IOBest [7], sendo localizadas as partes referentes aos componentes e adicionados a eles a função *gettimeofday*, uma vez que tanto as políticas de escalonamento quanto o AMIGO foram desenvolvidos em linguagem de programação C.

Tais políticas foram escolhidas por serem dinâmicas e tratarem o escalonamento sob diferentes métricas. A política DPWP utiliza métricas para escalonar aplicações *CPU-BOUND*. A MinMax utiliza um conjunto de informações (capacidade da CPU, memória RAM, etc.) a fim de determinar as máquinas mais aptas para participar de um escalonamento, através do conceito de que mínimos recursos determinam meios de execução com o máximo de desempenho. Por fim, a IOBEST adota métricas para escalonar aplicações *IO-BOUND*.

A Tabela 1 apresenta as funções que implementam os componentes de informação e localização das políticas de escalonamento que foram instrumentadas. Nesta Tabela é possível notar a ausência de funções referentes aos componentes de transferência e de seleção, nas três políticas de escalonamento. Isso acontece pois as políticas consideram como nó emissor aquele que cria novos processos, e no componente de seleção é o ambiente de passagem de mensagem quem seleciona os processos a serem escalonados, e não a política.

Os experimentos foram realizados executando 30 (trinta) vezes com nove processos a serem escalonados, as aplicações listadas a seguir:

- ordenação de vetores através do método *Quick-Sort*;
- resolução de Integrais;
- problema de entrada e saída na fila de CPU (IO-CPU).

Essas aplicações foram escolhidas devido a algumas características que apresentam, das quais podemos citar a utilização de muita comunicação, muito processamento e muita entrada/saída, respectivamente.

Os resultados obtidos correspondem ao tempo de execução dos componentes que compõem as políticas

Políticas	Componente de Informação	Componente de Localização
DPWP	<i>get_message</i> <i>Put_Table</i> <i>getloadlocal</i> <i>getloadremote</i> <i>getindexload</i> <i>getpowerlocal</i> <i>get_averageload</i>	<i>get_receivers</i>
MinMax	<i>getlocalindex</i> <i>readlocalload</i> <i>updatehost</i> <i>Put_Table</i> <i>putloadremote</i> <i>getloadlocal</i>	<i>select_receivers</i> <i>send_receivers</i>
IOBest	<i>update_local_load</i> <i>request_loads</i> <i>manage_loads</i> <i>get_benchers</i>	<i>choose_receivers</i>

**Tabela 1. Funções instrumentadas das políticas de escalonamento**

de escalonamento, e baseados neles, pode-se observar o comportamento de cada um dos componentes nas diferentes políticas.

Nas três políticas utilizadas, foram observadas que algumas das funções instrumentadas não apresentaram os tempos de suas execuções como previsto teoricamente, 30 resultados para cada uma.

Na política DPWP, as funções *getindexload* e *get\_averageload* geram mais resultados que a função *get\_message*, quando analisado o componente de informação. Isto acontece, pois a própria política realiza o cálculo de carga da máquina local (*getindexload*) em intervalos de um segundo, e também realiza um cálculo da média das cargas relativos aos últimos 20 segundos (*get\_averageload*), enquanto a função *get\_message*, que é a responsável por desempacotar uma mensagem enviada do AMIGO para a política DPWP e selecionar qual função deve tratá-la, só é executada uma vez a cada escalonamento.

Nos resultados da política MinMax, ao analisarmos os resultados do componente de informação, nota-se que as funções *readlocalload* e *updatehost* apresentam mais resultados em relação às outras funções. Essas duas funções fazem parte de blocos de instruções que capturam e atualizam, respectivamente, as cargas das máquinas em intervalos de um segundo.

Já na política IOBest, a função *update\_local\_load* é quem apresenta a característica descrita. Isto se deve pois essa função tem a finalidade de receber a carga lo-

cal do módulo de gerenciamento de cargas em intervalos de um segundo.

De posse de todos os dados adquiridos, foram realizados os cálculos de média, desvio padrão, variância e somatório de todas as funções, de modo a obtermos os valores médios (em microsegundos) a serem analisados.

Assim, as Tabelas 2, 3 e 4, apresentam os valores do tempo médio de execução referentes a análise da sobrecarga de cada componente durante a execução das três aplicações para cada uma das políticas de escalonamento, respectivamente.

Componente	QuickSort
Localização	904,67933
Informação	759066,10535
Componente	Integral
Localização	772,23783
Informação	4590295,55794
Componente	IO-CPU
Localização	939,96528
Informação	227885,24643

**Tabela 2. Valores do tempo médio de execução de cada componente quando utilizada a política DPWP**

Componente	QuickSort
Localização	1307722,56850
Informação	11134,50517
Componente	Integral
Localização	1999677,41965
Informação	260831,11969
Componente	IO-CPU
Localização	1345794,08171
Informação	7520,07955

**Tabela 3. Valores do tempo médio de execução de cada componente quando utilizada a política MinMax**

Já as Tabelas 5 e 6 comparam a sobrecarga dos componentes de localização e informação em cada uma das políticas analisadas. Essa comparação foi feita utilizando os valores correspondentes à média dos tempos de execuções dos componentes quando executada somente a aplicação de resolução de Integrais.

Componente	QuickSort
Localização	6603,00251
Informação	2356,17124
Componente	Integral
Localização	5385,99492
Informação	11725,78078
Componente	IO-CPU
Localização	6958,60388
Informação	1012,68246

**Tabela 4. Valores do tempo médio de execução de cada componente quando utilizada a política IOBest**

Política de Escalonamento	Tempo de Execução
DPWP	772,23783
IOBest	5385,99492
MinMax	1999677,41965

**Tabela 5. Valores médio do tempo de execução das políticas de escalonamento sobre o componente de localização**

## 5. Conclusões

Analisando os componentes de localização e informação, a política de escalonamento DPWP possui um desempenho superior em relação às outras. Isto pode ser comprovado quando comparamos os tempos médio de execução do componente de localização das três políticas, que dentre todos os outros, é o mais importante pelo fato de ser o responsável por selecionar e indicar quais serão as máquinas receptoras do ambiente mais aptas para escalonar determinada aplicação.

Dentre as aplicações, as classificadas como CPU-Bound, são as que apresentam melhor desempenho em todas as políticas de escalonamento quando anali-

sado o componente de localização, exceto na MinMax. Isto acontece pelo fato da MinMax “abrir” os arquivos de configurações das máquinas do ambiente e das aplicações a serem escalonadas, para analisar os recursos mínimos e indicar quais máquinas podem participar do escalonamento. Já quando analisado o componente de informação as aplicações do tipo I/O-Bound são as que apresentam um melhor desempenho nas três políticas de escalonamento.

Algumas sugestões para trabalhos baseados neste, podem consistir em avaliar o desempenho das políticas de escalonamento utilizando técnicas de simulação e modelagem analítica das políticas e avaliar o desempenho dos demais módulos do ambiente AMIGO a fim de determinar possíveis sobrecargas no processamento desses módulos.

## Referências

- [1] A. P. F. Araújo, P. S. L. Souza, M. J. Santana, and R. C. H. Santana. Dpwp - a new load balancing algorithm. *5th International Conference on Information Systems, Analysis and Synthesis: ISAS'99*, Julho 1999.
- [2] A. Beguelin, A. Geist, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine: A User's guide and Tutorial for Networked Parallel Computing*. The MIT Press, Massachusetts, 1994.
- [3] MPI. Mpi: Message passing interface. <http://www-unix.mcs.anl.gov/mpi/> Último acesso em 15/07/2004.
- [4] N. G. Shivaratri, P. Kreuger, and M. Singhal. Load distribution for locally distributed systems. *IEEE Computer*, 25:33–44, 1992.
- [5] P. S. L. Souza. *AMIGO: Um Ambiente de Escalonamento Flexível e Dinâmico*. PhD thesis, Instituto de Física de São Carlos - USP, São Carlos - SP, 2000.
- [6] P. S. L. Souza, D. C. Pereira, W. Voorsluys, and S. R. S. Souza. An approach to deal with heterogeneity and multi-users on cluters using minimal resources. *IASTED'2003*, February 2003.
- [7] W. Voorsluys, P. S. L. Souza, M. J. Santana, R. H. C. Santana, K. R. C. Branco, D. C. Pereira, and S. R. S. Souza. Iobest - a new dynamic scheduling policy for i/o specific demand. *IPDPS'2003*, Julho 2003.

Política de Escalonamento	Tempo de Execução
DPWP	4590295,55794
IOBest	11725,78078
MinMax	260831,11969

**Tabela 6. Valores médio do tempo de execução das políticas de escalonamento sobre o componente de informação**