

# Avaliação de Desempenho de um Reconhecedor de Imagens Concorrente\*

Eduardo Moschetta<sup>†</sup>, Fernando S. Osório,  
Gerson Geraldo H. Cavalheiro

Programa Interdisciplinar de Pós-Graduação em Computação Aplicada  
Centro de Ciências Exatas e Tecnológicas  
Universidade do Vale do Rio dos Sinos  
{eduardom, osorio, gersonc}@exatas.unisinos.br

## Introdução

Ao longo dos anos, informações multimídia vêm sendo largamente utilizadas. Com o aumento do uso dessas informações, em especial imagens e vídeos, novas aplicações foram desenvolvidas a fim de permitir o processamento dessas sobre diferentes contextos.

Nesse cenário, um possível tratamento seria, a partir de uma coleção de imagens-exemplo, buscar uma determinada imagem. Tais aplicações caracterizam-se como sistemas de recuperação de imagens, sendo na literatura mais conhecidas como “Content Based Image Retrieval” (CBIR) ou “query by example” (QBE) [DAS 97].

Basicamente, o método de busca de um fragmento de imagem em um banco de imagens consiste em percorrer todas as imagens do banco e, para cada uma, criar uma cabeça de leitura. Essa cabeça de leitura desloca o fragmento pixel a pixel pela imagem, sendo que para cada deslocamento é aplicado um algoritmo de *matching*, o qual verifica a similaridade entre o fragmento e a região da imagem sobreposta pelo mesmo.

Em geral, algoritmos de *matching* necessitam percorrer todos pixels do fragmento e realizar diversos cálculos, tornando visível a grande carga computacional gerada por esses algoritmos. Uma solução para esse problema pode ser buscada na utilização de processamento paralelo [FAN 2000].

Nesse trabalho é apresentada uma avaliação de desempenho para um reconhecedor de imagens implementado de forma concorrente [MOS 2002] [MOS 2002a], que utiliza técnicas de paralelismo híbrido em arquiteturas do tipo SMP (*Symmetric Multi-Processors*) e agregados de computadores.

## Implementação Concorrente

A implementação realizada explora dois níveis de concorrência. O primeiro nível, de granulosidade mais grossa, considera que imagens são dados independentes entre si,

---

\*Projeto Anahy. CNPq/Protem-CC (68.0120/01-7), FAPERGS (00/2740-8) e UNISINOS (37.00.001/00-0).

<sup>†</sup>PIBIC/CNPq

podendo várias buscas serem efetivadas ao mesmo tempo. Sendo assim, as imagens (grãos grossos) podem ser distribuídas, de forma igual, entre os diferentes nodos da arquitetura, sendo que cada nó fica responsável pela busca no seu subconjunto de imagens.

O segundo nível, mais fino e interno ao nó, explora o fato de que diferentes regiões da imagem podem ser comparadas ao mesmo tempo, o que leva à possibilidade de criar-se múltiplas cabeças de leitura. Na implementação realizada, cria-se uma lista de tarefas e um *pool* de execução em cada nó, composto de várias cabeças de leitura (*threads*) que podem percorrer diferentes regiões de diferentes imagens de forma concorrente. As tarefas implementam a computação de uma operação de *matching* sobre uma região da imagem. Na ferramenta desenvolvida esta tarefa é programada em uma função, sendo que diferentes funções podem ser adicionadas de forma a implementar diferentes algoritmos.

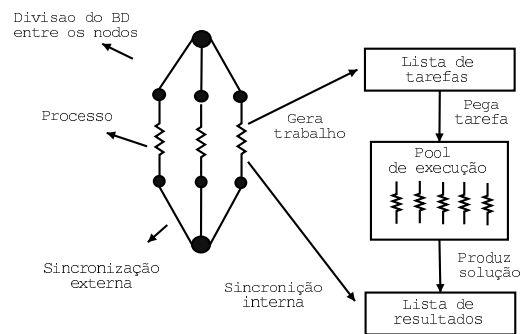


Figura 1: Modelo de Concorrência

A Figura 1 mostra o grafo que representa a execução da aplicação, para uma arquitetura composta por  $n$  nodos. As sincronizações internas obtêm todas soluções encontradas pelas *threads* em diferentes regiões de imagens, enquanto que a sincronização externa, necessária apenas uma vez, permite que nodos retornem ao mestre os resultados obtidos em seus subconjuntos de imagens.

## Balanceamento de Carga

Com vistas a minimizar o impacto da diferença de velocidades dos nodos, foi introduzido um mecanismo de balanceamento de carga. Este mecanismo gera a distribuição de carga computacional através da migração de imagens (migração lógica, pois fisicamente as imagens são acessíveis a todos os nodos).

Nessa implementação, o balanceamento de carga é realizado através de um nó mestre, designado apenas para isso. Do outro lado, os nodos escravos ficam responsáveis pela busca propriamente dita. Ao processar uma nova imagem, o nó escravo envia uma mensagem ao mestre, fazendo com que esse o identifique. O mestre atualiza o estado da execução global, podendo, entre outras situações, detectar que esse nó estará ocioso na sequência. Com base no estado atualizado, o mestre gera uma ação a ser enviada de volta ao escravo. Nesse modelo, observa-se que o mestre tem condições de controlar tanto a evolução da execução quanto a distribuição do trabalho da aplicação.

É importante observar que o uso de *threads*, além de aumentar o *throughput* do nó, permitiu que se sobreposse parte do custo gerado pelas comunicações com cálculo efetivo [CAV 2001].

## Resultados

Alguns testes de execução foram realizados com a aplicação, a fim de validar a proposta de paralelismo híbrido documentada. Os experimentos foram realizados em uma máquina SMP (2 x PIII 1GHz) e em um agregado com 6 processadores (3 nodos bi-processados, sendo 1 x PIII 600 MHz e 2 x PIII 1 GHz). O banco de imagens utilizado nos testes consistia de 510 imagens. A diferença de tamanho das imagens e a heterogeneidade do agregado (quanto à capacidade de processamento) permitiram avaliar e validar o balanceamento de carga proposto.

Os testes consistiram na busca de três fragmentos de dimensões diferentes (pequeno, médio e grande) para os dois algoritmos de *matching* implementados (ponto a ponto e histogramas) nos sistemas de cores RGB e HSV (esse último apenas para o algoritmo de comparação de histogramas). A título comparativo, foram realizadas execuções sequenciais dessas buscas, conforme a Tabela 1.

Tabela 1: Execução sequencial

Algoritmo	Ponto a Ponto	Histogramas	
Fragmento	RGB	RGB	HSV
Pequeno	257.150 s	3.324 s	11.067 s
Médio	541.827 s	2.674 s	9.871 s
Grande	16.758 s	53 s	203 s

A Tabela 2 documenta os tempos de execução em uma máquina bi-processada (SMP), bem como o ganho obtido com essas execuções em relação às execuções sequenciais equivalentes. As buscas foram configuradas com 20 cabeças de leitura (*threads*). Nota-se que o ganho obtido foi próximo a 2 (número de processadores), independente da carga inicial de trabalho, o que valida o escalonamento local de tarefas.

Tabela 2: Execução em uma máquina SMP

Algoritmo	Ponto a Ponto		Histogramas			
Cores	RGB		RGB		HSV	
Fragmento	Tempo	Ganho	Tempo	Ganho	Tempo	Ganho
Pequeno	129.275 s	1,99	1.675 s	1,98	5.675 s	1,95
Médio	282.025 s	1,92	1.392 s	1,92	5.228 s	1,89
Grande	8.891 s	1,88	34 s	1,56	134 s	1,51

Os tempos e ganhos das execuções no agregado de computadores podem ser visualizados na Tabela 3, para a mesma configuração efetuada na máquina SMP. Os *speed-ups*

obtidos foram próximos ao número de processadores do agregado (6 processadores), validando o escalonamento global da aplicação, em especial o balanceamento de carga.

Tabela 3: Execução em um agregado de computadores

Algoritmo	Ponto a Ponto		Histogramas			
Cores	RGB		RGB		HSV	
Fragmento	Tempo	Ganho	Tempo	Ganho	Tempo	Ganho
Pequeno	48.672 s	5,28	627 s	5,30	2.118 s	5,23
Médio	104.399 s	5,18	517 s	5,17	1.911 s	5,17
Grande	3.721 s	4,50	22 s	2,41	59 s	3,44

## Conclusão

A implementação concorrente proposta neste trabalho viabilizou a redução do tempo necessário para recuperação de imagens em grandes bases das mesmas.

Um novo algoritmo de *matching* é incluído através da adição de uma nova função que o implementa. A exploração dos recursos de programação do hardware é possível através do núcleo de escalonamento provido pela ferramenta.

Os resultados documentados nesse artigo têm sido aplicados na modelagem de Anahy – ambiente de programação para o processamento de alto desempenho –, em especial na validação do ambiente e definição da interface de programação do mesmo.

## Referências

- [CAV 2001] CAVALHEIRO, G. G. H. Introdução à programação paralela e distribuída. In: ESCOLA REGIONAL DE ALTO DESEMPENHO, 1., 2001, Gramado. **Anais...** [S.l.: s.n.], 2001.
- [DAS 97] DAS, M.; RISEMAN, E.; DRAPER, B. Focus: searching for multi-colored objects in a diverse image database. In: IEEE CONF. ON COMP. VIS. AND PATTERN RECOGNITION, 1997, Porto Rico. **Anais...** [S.l.: s.n.], 1997.
- [FAN 2000] FAN, J.; YIU, S.; PO, L. Image retrieval using parallel algorithm on the beowulf class supercomputer. In: INTERNATIONAL WORKSHOP ON MULTIMEDIA DATA STORAGE, RETRIEVAL, INTEGRATION AND APPLICATIONS, 2000. **Anais...** [S.l.: s.n.], 2000. p.188–191.
- [MOS 2002] MOSCHETTA, E.; OSÓRIO, F. S.; CAVALHEIRO, G. G. H. Reconhecedor de imagens usando técnicas de alto desempenho. In: ESCOLA REGIONAL DE ALTO DESEMPENHO, 2., 2002, São Leopoldo, RS. **Anais...** [S.l.: s.n.], 2002.
- [MOS 2002a] MOSCHETTA, E.; OSÓRIO, F. S.; CAVALHEIRO, G. G. H. Reconhecimento de imagens em aplicações críticas. In: III WORKSHOP EM SISTEMAS COMPUTACIONAIS DE ALTO DESEMPENHO, 2002, Vitória, ES. **Anais...** [S.l.: s.n.], 2002.