

***UMA ANÁLISE DO DESEMPENHO DE DIFERENTES
PROCESSADORES NA EXECUÇÃO DOS ALGORITMOS MAXMIN***

*An analysis of the performance of different processors in the
implementation of the Algorithms Maxmin*

Walteno Martins Parreira Júnior, Marcio Oliveira Costa, Danilo Humberto D. Santos,
Cássio S. Matsumoto

RESUMO

O trabalho busca comparar o desempenho de um conjunto de algoritmos denominados de MaxMin em processadores com frequências (Hz) diferentes. Assim foram realizados os testes e obtidos os dados de desempenho, ou seja, tempo de execução de cada um e em seguida comparados com a teoria disponível na literatura e desenvolvidas as considerações sobre os resultados.

Palavras-chave: Tempo de execução. Algoritmos MaxMin. Performance de processadores.

ABSTRACT

The work searches to compare the one performance conjunto of called algorithms of MaxMin in processors with frequencies (Hz) different. As soon as the tests had been carried through and gotten the performance data, that is, time of execution of each one and after that comparing them with the theory available in literature and developed the consideration on the results.

Keywords: Time of execution. MaxMin Algorithms. Performance of processors.

INTRODUÇÃO

O trabalho aborda o desempenho de três processadores cujo ano de fabricação, modelo e velocidade de processamento (Clock), entre outras características que são diferentes, e comparando o desempenho no final das execuções de um conjunto de algoritmos com tempos de execução conhecidos.

Um algoritmo corresponde a uma descrição de um padrão de comportamento, expresso em termos de um conjunto finito de ações. Ao executarmos a operação $a + b$ percebemos um mesmo padrão de comportamento, mesmo que a operação seja realizada para valores diferentes de a e b (ZIVIANI, 2002, p.1).

O uso do tempo de execução é uma das formas de comparar a diferença de desempenho entre algoritmos utilizando parâmetros conhecidos. Há várias formas de comparar diferenças de desempenho entre computadores e entre programas computacionais. No estudo de desempenho de programas computacionais, a complexidade é uma destas formas.

O custo de utilização de um algoritmo pode ser medido de várias maneiras. Uma delas é através da execução do programa de computador real, sendo o tempo de execução medido diretamente (ZIVIANI, 2002, p.4).

E complementa que os dados desta forma não podem ser generalizados, pois alguns fatores devem ser considerados, tais como: os resultados encontrados são dependentes do compilador da linguagem escolhida para codificar o algoritmo; e também que os resultados são dependentes da configuração do hardware utilizado. A utilização e a quantidade de memória RAM influenciam nos resultados.

Segundo Drozdek (2002, p.48), a complexidade computacional indica o esforço necessário para executar um algoritmo. Este custo pode ser medido de várias maneiras e o fator tempo é um dos mais importantes.

Segundo Cormen e outros (2002, p.7), os “algoritmos criados para resolver o mesmo problema muitas vezes diferem de forma drástica em sua eficiência. Essas diferenças podem ser muito mais significativas que as

diferenças relativas a hardware e software”. E este pode ser observado nos objetivos dos diversos algoritmos MaxMin.

Usando três algoritmos diferentes denominados de MaxMin1, MaxMin2, MaxMin3, onde todos retornam o maior e o menor valor de uma lista com uma quantidade n de elementos, deve-se observar três situações distintas em função da posição do elemento procurado, durante a execução (ZIVIANI, 2002, p.6). Na situação de **melhor caso**, este ocorre quando os elementos da lista estão em ordem crescente, portanto $(N-1)$. Comparações são necessárias. Na situação de **pior caso**, quando os elementos da lista estão em ordem decrescente, portanto $2(N-1)$ comparações são necessárias. E para terminar, na situação onde ocorre o **caso médio**, pois os elementos da lista são maiores que o maior valor da lista, aproximadamente na metade das vezes e são menos aproximadamente na metade das vezes, portanto, $(3N/2 - 3/2)$ comparações são necessárias.

Segundo Parreira Júnior (2006, p.16-17), os algoritmos MaxMin2 e MaxMin3 são superiores, em performance, ao MaxMin de forma geral. O algoritmo MaxMin3 é superior ao MaxMin2 com relação ao pior caso, e é bastante próximo quanto ao caso médio.

Segundo Pradela e outros (2008, p.3), a complexidade dos algoritmos MaxMin é de ordem $O(n)$ para todos os casos, e a diferença entre os algoritmos são os números de comparações para cada caso.

Segundo Ziviani (2002, p.3), “o projeto de algoritmos é fortemente influenciado pelo estudo de seus comportamentos”. E continua afirmando que o projetista deve observar as opções de algoritmo que podem ser utilizadas e considerar os aspectos relacionados ao tempo de execução e de espaço ocupado como fatores importantes neste estágio. Neste estudo, será considerado o fator tempo de execução para o desenvolvimento das observações.

Os algoritmos foram codificados em linguagem C, usando o compilador Turbo C disponível nos laboratórios da Instituição.

O Turbo C e o C++ são compiladores em [linguagem] C, muito usados em faculdades e para o aprendizado da linguagem de programação C e C++. Eles foram originariamente criados pela Borland, mesma empresa que fabrica o Delphi. Todos os Turbo são gratuitos, porque são programas obsoletos, sendo criados em 1990, e com interface no DOS (INFOMAROTO, 2008).

Uma observação importante na execução dos algoritmos é que nas três implementações dos algoritmos foram utilizados a função “delay (100)” para retardar a execução em décimos de segundos e sem essa função não seria possível comparar valores, pois, o tempo de execução seria de aproximadamente 0 (Zero) devido à velocidade de processamento dos processadores atuais.

O objetivo do presente trabalho é a análise dos algoritmos e a implementação de algoritmos de busca de menor e maior elemento de um vetor, conhecidos como MaxMin, testando sua eficiência em máquinas distintas. A ideia principal foi realizar a comparação entre os tempos encontrados na execução de algoritmos em processadores diferentes que estão à disposição nos laboratórios de pesquisa da Instituição.

MATERIAL E MÉTODOS

Um algoritmo trivial para calcular o máximo e o mínimo de L tem a seguinte configuração (onde se deve considerar M1 como sendo o máximo e o mínimo temporário): se o máximo temporário é menor que o M2, considerar então M2 como o novo máximo temporário; se o mínimo temporário é maior do que M2, considerar então M2 como sendo o mínimo temporário; repetir o processo para M3 e assim sucessivamente até Mn. Após a comparação com Mn, tem-se que o máximo e o mínimo temporários são os valores desejados. Foram realizadas $2(n-1)$ comparações do máximo e mínimo temporários com os elementos da lista (PARREIRA JUNIOR, 2006, p.16).

O Processador Pentium III 933 MHz é um microprocessador de sexta geração, fabricado pela Intel entre os anos de 1999 – 2003. Dentre os

processadores testados, este é o de menor Clock, de menor Cache L2 e também de velocidade de barramento. Atualmente o processador Pentium III é encontrado apenas em computadores antigos, pouco aproveitados, tanto em atividade empresarial, quanto para uso doméstico.

O quadro 1 apresenta um comparativo dos tempos de execução nos 3 algoritmos para o processador Pentium III.

Algoritmo	Tempo (em segundo)		
	Melhor caso	Pior caso	Caso médio
Maxmin1	0.93	0.93	0.93
Maxmin2	0.94	0.94	0.88
Maxmin3	0.41	0.41	0.41

Quadro 1- Mostra o tempo de execução para o processador PIII (em segundos)

Outro processador usado nos testes foi um Processador Pentium IV 2.8GHz também disponível no Laboratório de Pesquisa da Instituição.

O Pentium 4 foi lançado pela Intel em novembro de 2000, trazendo uma arquitetura completamente redesenhada, baseada na idéia do uso de um longo pipeline para permitir que o processador fosse capaz de atingir frequências de clock elevadas e no uso de um cache L1 muito rápido e um barramento de dados capaz de realizar 4 transferências por ciclo para mantê-lo alimentado com o volume necessário de dados e instruções (MORIMOTO, 2005).

No quadro 2 é apresentado um comparativo dos tempos de execução considerando a execução dos programas que implementam os 3 algoritmos com a utilização do processador Pentium IV.

Algoritmo	Tempo (em segundo)		
	Melhor caso	Pior caso	Caso médio
Maxmin1	0.93	0.93	0.93
Maxmin2	0.94	0.93	0.93
Maxmin3	0.38	0.39	0.39

Quadro 2 - Mostra o tempo de execução para o processador PIV (em segundos)

O terceiro processador usado na pesquisa é o Processador Core 2 Quad 2.67GHz e este modelo está sendo fabricado até os dias atuais e é bastante utilizado empresarialmente e também em computadores de alta performance.

Esta nova arquitetura é baseada na arquitetura do Pentium M, trazendo algumas novas características. [...] O Pentium M é um processador voltado para o mercado de notebooks e é baseado na arquitetura Intel de 6ª geração, também conhecida como P6, que é a mesma arquitetura usada pelos processadores Pentium Pro, Pentium II, Pentium III e os primeiros modelos de Celeron. [...] A arquitetura Core foi criada com o conceito da tecnologia de múltiplos núcleos em mente, ou seja, mais de um núcleo de processamento dentro de um único processador físico (TORRES; LIMA, 2006).

[...] é o processador Quad-Core da Intel, vendido sob a marca "Core 2 Quad". [...] é na verdade um processador dual-chip, onde temos dois processadores dual-core independentes, colocados dentro do mesmo encapsulamento e interligados através do front-side bus (MORIMOTO, 2007).

O Quadro 3 apresenta os tempos conseguidos com a execução dos programas codificados a partir dos três algoritmos.

Algoritmo	Tempo (em segundo)		
	Melhor caso	Pior caso	Caso médio
Maxmin1	0.92	0.92	0.92
Maxmin2	0.95	0.94	0.94
Maxmin3	0.39	0.39	0.39

Quadro 3 - Mostra o tempo de execução para o processador Core 2 Quad (em segundos)

Cada programa codificado foi executado três vezes em cada um dos computadores e depois foi calculada a média destas execuções para a confecção dos quadros com os tempos de execução.

CONSIDERAÇÕES FINAIS

Como pôde ser observado na apresentação dos resultados, foram testadas as mesmas codificações dos algoritmos em três máquinas diferentes,

com processadores diferentes e mesmo assim os resultados foram próximos. Observando as comparações do quadro três, nota-se que a diferença de tempo ao final da execução dos algoritmos foi mínima, inferiores a 10%, logo, pode-se concluir que os processadores que possuem uma arquitetura constituída por múltiplos núcleos são irrelevantes nestas situações, pois os algoritmos apresentados neste artigo – MaxMin – foram desenvolvidos na linguagem C, cujo compilador foi desenvolvido para trabalhar em máquinas com processadores de núcleo simples. Conclui-se, então, que a relevância a ser considerada neste teste são os algoritmos utilizados.

Destacando-se entre os três, o MaxMin3 sendo, em um valor aproximado, 60% mais rápido que os demais. Quanto ao MaxMin1 e MaxMin2 nota-se uma grande proximidade entre os valores encontrados no tempo de execução, observando que os códigos fonte de ambos foram escritos de forma semelhante, apresentando apenas uma pequena melhora no MaxMin2, em relação ao MaxMin1.

REFERÊNCIAS

CORMEN, Thomas H. et al. **Algoritmos: teoria e prática**. Rio de Janeiro: Elsevier, 2002, 10ª reimpressão.

DROZDEK, Adam. **Estrutura de dados e algoritmos em C++**. São Paulo: Pioneira Thomson Learning, 2002.

INFOMAROTO (Blog). **Download de todas as versões do turbo c e c++**. 24 ago. 2008. Disponível em: <http://www.infomaroto.com/blog/download-de-todas-as-versoes-do-turbo-c-e-c/>. Acesso em: 21 out. 2010.

MORIMOTO, Carlos E. **Kentsfield**. 8 ago. 2007. Disponível em <http://www.hardware.com.br/termos/kentsfield>. Acesso em: 20 out. 2010.

MORIMOTO, Carlos E. **Pentium 4**. 26 jun. 2005. Disponível em <http://www.hardware.com.br/termos/pentium-4>. Acesso em: 20 out. 2010.

PARREIRA JÚNIOR, Walteno M.. **Análise de algoritmos** (Apostila). Ituiutaba: FEIT-UEMG, 2006.

PRADELA, Izaura P. et al. Comparação do tempo de execução de algoritmos que retornam o maior e menor elemento de um vetor. In: Seminário de Iniciação Científica e extensão da UEMG, 10, 2008, Divinópolis. **Anais...** Divinópolis: UEMG e FUNED. 2008. CD-ROM. Disponível em: www.waltenomartins.com.br/artigos. Acesso em: 22 out. 2010.

TORRES, Gabriel; LIMA, Cássio. **Por dentro da microarquitetura Intel core.** 25 abr. 2006. Disponível em: <http://www.clubedohardware.com.br/printpage/Por-Dentro-da-Microarquitetura-Intel-Core/1203>. Acesso em: 22 out. 2010.

ZIVIANI, Nivio. **Projeto de algoritmos: com implementação em Pascal e C.** São Paulo: Pioneira – Thomson Learning, 2002.

AUTORES

Walteno Martins Parreira Júnior, professor dos cursos de Engenharia da Computação, Engenharia Elétrica e Sistemas de Informação da Fundação Educacional de Ituiutaba – FEIT, associada à Universidade do Estado de Minas Gerais – UEMG, Campus de Ituiutaba-MG. Especialista de Design Instrucional para EaD e Informática Aplicada à Educação. Mestrando em Educação no PPGED-UFU.
waltenomartins@yahoo.com

Marcio Oliveira Costa, professor dos cursos de Engenharia da Computação e Sistemas de Informação da Fundação Educacional de Ituiutaba – FEIT, associada à Universidade do Estado de Minas Gerais – UEMG, Campus de Ituiutaba-MG.
costafilo@yahoo.com.br

Danilo Humberto D. Santos, acadêmico do curso de Engenharia da Computação da Fundação Educacional de Ituiutaba – FEIT, associada à Universidade do Estado de Minas Gerais – UEMG, Campus de Ituiutaba-MG.
danilo1berto@hotmail.com

Cássio S. Matsumoto, acadêmico do curso de Engenharia da Computação da Fundação Educacional de Ituiutaba – FEIT, associada à Universidade do Estado de Minas Gerais – UEMG, Campus de Ituiutaba-MG.
cassiomatsumoto@hotmail.com