

***OBSERVANDO AS DIFERENÇAS DE TEMPO NAS EXECUÇÕES
DE ALGORITMOS EM PROCESSADORES DIFERENTES***

***Noting the Time Differences in the Execution of Algorithms
on Different Processors***

Walteno Martins Parreira Júnior, Marcio Oliveira Costa, Dione Mateus Parreira,
Thassio Souza e Silva

RESUMO

O trabalho aborda as diferenças de desempenho quando é executada a codificação de algoritmos em processadores com diferentes frequências (Hz) e quantidade de núcleos distintos. Tendo em nota os dados obtidos dos algoritmos, comparando-os com a teoria apresentada nos referenciais e obtendo as informações necessárias para uma conclusão do assunto.

Palavras-chave. Algoritmos Maxmin. Comparações entre Processadores. Algoritmos.

ABSTRACT

The work approaches the performance differences when the codification of algorithms in processors with different frequencies (Hz) and amount of distinct nuclei are executed. Having in note the gotten data of the algorithms, comparing them with the theory presented in the referenciais and getting the necessary information for the one conclusion of the subject.

Keywords: Maxmin Algorithms. Comparisons Between Processors. Algorithms.

INTRODUÇÃO

Esse trabalho aborda a comparação da execução da codificação dos Algoritmos Maxmin1, Maxmin2 e Maxmin3 e a consequente diferença observada entre as execuções destes programas em três processadores distintos. Segundo Capron e Johnson (2004, p.309), um processador é a unidade central de processamento (CPU) de um computador.

Para esta pesquisa foram considerados três computadores equipados com os seguintes processadores: Pentium D 3.2 GHz, Core 2 Quad 2.83 GHz (com um núcleo de processamento) e Core 2 Quad 2.83 GHz (com dois núcleos de processamento). E considerando que as quantidades de memória principal dos três computadores são do mesmo tamanho.

Para efeito de comparação, os algoritmos foram compilados no compilador Borland C. Segundo Capron e Johnson (2004, p.293), um compilador é um “tradutor que converte as declarações simbólicas de uma linguagem de alto nível em linguagem de máquina executável por computador”.

Será que há diferença significativa de desempenho entre os programas executados nos diferentes processadores, considerando que foram todos gerados no mesmo compilador?

Os Algoritmos Maxmin

Segundo Salvetti e Barbosa (1998, p.5), um algoritmo é uma sequência finita de instruções ou de operações básicas cuja execução resolve um problema computacional em um tempo finito.

Ainda segundo Salvetti e Barbosa (1998, p.129), o objetivo do algoritmo MaxMin é o de determinar o maior e o menor valor de uma lista com n números.

Considerando o problema de achar o maior valor e o menor valor em uma lista L com n elementos em um conjunto C , onde $L = (m_1, m_2, \dots, m_n)$.

Segundo PARREIRA JÚNIOR (2006, p. 16), um algoritmo trivial para calcular o máximo e o mínimo de L seria: considerar M_1 como sendo o máximo e o mínimo temporário; se o máximo temporário é menor que o que M_2 , considerar então M_2 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,..., 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.

Podem ocorrer três situações quando da execução do programa. A primeira situação é a do **melhor caso**, que neste algoritmo ocorre quando os elementos do vetor estão em ordem crescente. A segunda situação é relativo ao pior caso, que ocorre quando os elementos do vetor estão em ordem decrescente. E por último, o **caso médio**, onde o valor na posição I do vetor é maior do que o índice max a metade das vezes.

Existem vários aspectos a considerar na determinação da complexidade de tempo de um algoritmo. Um critério pode ser medir o tempo de execução requerido por diferentes algoritmos para a solução de um problema particular em um computador específico (TOSCANI; VELOSO, 2001, p. 10).

Como pode ser observado no Quadro 1, o tempo médio calculado para o processamento dos programas MaxMin1 em segundos é apresentado.

Processadores	Tempo (em Segundo)		
	Melhor caso	Pior caso	Caso médio
Pentium D 3.2 GHz	1.92	1.92	1.92
Core 2 Quad 2.83 GHz (com um núcleo de processamento)	1,92	1.92	1.92
Core 2 Quad 2.83 GHz (com dois núcleo de processamento)	1.76	1.76	1.76

Quadro 1 – Tempo de comparação do MaxMin1 (em segundos).

Segundo PARREIRA JÚNIOR (2006, p. 16-17), o algoritmo MaxMin2 é uma versão melhorada do MaxMin1, onde a comparação $A[i] < \min$ só é necessária quando o resultado da comparação $A[i] > \max$ é falsa, considerando

i como o índice da posição do vetor e faz com que possa ser considerado as situações descritas a seguir.

Em primeiro lugar, o melhor caso ocorre quando os elementos do vetor estão em ordem crescente na leitura da lista. Que o pior caso ocorre quando os elementos do vetor estão em ordem decrescente. E que no caso médio, o vetor [i] é maior do que o índice max a metade das vezes em que executado o processo.

Pode ser observado no quadro 2 os tempos coletados nos processadores com a utilização do MaxMin2.

Processadores	Tempo (em Segundo)		
	Melhor caso	Pior caso	Caso médio
Pentium D 3.2 GHz	1.87	1.93	1.92
Core 2 Quad 2.83 GHz (com um núcleo de processamento)	1.87	1.93	1.92
Core 2 Quad 2.83 GHz (com dois núcleo de processamento)	1.70	1.81	1.76

Quadro 2 – Comparação de tempo do Maxmin2 (em segundos).

Para aperfeiçoar o algoritmo MaxMin3, o algoritmo faz com que os elementos do vetor são comparados de dois em dois e os elementos maiores são comparados com max e os menores com min. Quando o número de elementos do vetor (n) é ímpar, o elemento que está na posição vetor[n] é duplicado na posição vetor [n+1] para evitar um tratamento de exceção, pois não teria com quem fazer a comparação.

Independente dos casos, o tamanho da entrada do vetor nos três algoritmos foi constante.

No quadro 3, os tempos encontrados no programa que implementa o algoritmo MaxMin3 para os três processadores foram:

Processadores	Tempo (em Segundo)		
	Melhor caso	Pior caso	Caso médio
Pentium D 3.2 GHz	0.88"	0.88"	0.88"
Core 2 Quad 2.83 GHz (com um núcleo de processamento)	0.88"	0.88"	0.88"
Core 2 Quad 2.83 GHz (com dois núcleo de processamento)	0.77"	0.77"	0.77"

Quadro 3 – Comparação de tempo do MaxMin3 (em segundos).

Número de Comparações em Cada Algoritmo

O Quadro 4 apresenta uma tabela com o número de comparações que são realizadas pelo algoritmo em cada situação que pode encontrar no momento da execução. Assim, pode-se observar que para os algoritmos MaxMin1 e MaxMin3 não apresentam diferença entre as situações de melhor caso, caso médio ou pior caso, pois sempre fazem o mesmo número de comparações. Somente o MaxMin2 é que há uma diferenciação no número de comparações para os casos que podem aparecer durante a execução.

Segundo Ziviani (2002, p. 6), o melhor caso corresponde ao menor tempo de execução, independentemente do tamanho da entrada, enquanto o caso médio corresponde a média dos tempos de todas as entradas de um tamanho n .

Processadores	Tempo (em Segundo)		
	Melhor caso	Pior caso	Caso médio
Maxmin1	$2(n-1)$	$2(n-1)$	$2(n-1)$
Maxmin2	$(n-1)$	$2(n-1)$	$3n/2-3/2$
Maxmin3	$3n/2-2$	$3n/2-2$	$3n/2-2$

Quadro 4 - Tabela de Complexidades. Fonte: PARREIRA JÚNIOR (2006, p.18).

Conforme as complexidades descritas por PARREIRA JÚNIOR (2006, p. 18) e utilizando vetores com vinte elementos, podem-se calcular os seguintes valores e que são apresentados no Quadro 5.

Processadores	Tempo (em Segundo)		
	Melhor caso	Pior caso	Caso médio
Maxmin1	38	38	38
Maxmin2	19	38	29
Maxmin3	28	28	28

Quadro 5 – Dados esperados para um vetor de 20 posições.

RESULTADOS E DISCUSSÃO

Comparando os resultados encontrados com cada um dos algoritmos nos diferentes processadores, nota-se que os resultados são muito próximos para o MaxMin1 e MaxMin2, ou seja, quase não houve mudanças nos tempos resultantes das execuções, com exceção do MaxMin3 quando usado o processador Core 2 Quad 2.83 GHz, que apresentou um resultado de aproximadamente 15% melhor no tempo, enquanto nos dois primeiros algoritmos esta diferença é de aproximadamente 10%.

Esta diferença entre o MaxMin3 os outros dois algoritmos, era esperada pela literatura, pois neste algoritmo há a comparação de dois a dois elementos, ele é o que resolve o algoritmo de forma mais otimizada.

O que não acompanhou a literatura é o MaxMin2 na situação de melhor caso, que deveria apresentar o menor valor de tempo dentre todos os encontrados e isto não ocorreu para nenhum dos processadores utilizados, apresentando valores próximos aos encontrados para as outras situações em todos os processadores utilizados.

O processador tem o seu papel fundamental no processamento do algoritmo, considerando que todos os atributos das máquinas, ou eram iguais, ou quase os mesmos. De cada processador, o que mais era esperado, é que quando o processador fosse um tanto quanto melhor que o outro, os resultados apresentados teriam diferenças um tanto significativo, o que não ocorreu.

CONSIDERAÇÕES FINAIS

Atualmente os processadores trabalham com um desempenho elevadíssimo, exemplo deixado pelos dados tirados das comparações, onde não há diferença ou quando existe, é mínima. Lembrando que esta diferença foi possível de se realizar, pois utilizamos uma função “delay” para parar a execução em poucos micros segundos em cada comparação, caso contrário os dados seriam muito próximo de zero.

O que ficou a desejar foi eficiência do processador com dois núcleos, pois teoricamente ele deveria executar praticamente com o dobro de rapidez dos demais. Isso não acontece por que os algoritmos não foram feitos para terem partes de seus códigos executados em núcleos diferentes.

Novas pesquisas e testes serão realizados para observar estes dados que não foram suficientemente explicados pela literatura, de modo a detectar as causas de suas ocorrências.

REFERÊNCIAS

CAPRON, H. L.; JOHNSON, J. A. **Introdução à informática**. São Paulo - SP: Pearson Prentice Hall, 2004.

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

SALVETTI, Dirceu Douglas; BARBOSA, Lisbete Madsen. **Algoritmos**. São Paulo - SP: Makron Books, 1998.

TOSCANI, Laira V.; VELOSO, Paulo A. S. **Complexidade de algoritmos: análise, projeto e métodos**. Porto Alegre: Instituto de Informática da UFRGS, Sagra Luzzatto. 2001.

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

AUTORES

Walteno Martins Parreira Júnior é professor dos cursos de Engenharia da Computação, Engenharia Elétrica e Sistemas de Informação da Fundação Educacional de Ituiutaba, associada à Universidade do Estado de Minas Gerais, 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 é professor dos cursos de Engenharia da Computação e Sistemas de Informação da Fundação Educacional de Ituiutaba, associada à Universidade do Estado de Minas Gerais, Campus de Ituiutaba-MG. Especialista em História da Filosofia: Tópicos Especiais e Mestrando em Psicanálise, Educação e Sociedade.
marcioyz@yahoo.com.br

Dione Mateus Parreira é discente do curso de Engenharia da Computação da Fundação Educacional de Ituiutaba, associada à Universidade do Estado de Minas Gerais, Campus de Ituiutaba-MG.
dione.engcomp@hotmail.com

Thassio Souza e Silva é discente do curso de Engenharia da Computação da Fundação Educacional de Ituiutaba, associada à Universidade do Estado de Minas Gerais, Campus de Ituiutaba-MG.
thassiovolley@hotmail.com