
 <p>Universidade do Minho</p>	Módulo 10 Otimização de desempenho e processamento vetorial	
--	--	---

Objectivos:

Com esta sessão pretende-se melhorar a compreensão das técnicas que permitem melhorar o desempenho de programas, incluindo técnicas de processamento vetorial.

Exercícios:

Utilize o PAPI para contar o número de instruções de salto e a quantidade de saltos corretamente previstos, alterando o módulo `papi_inst.cpp`. Utilize os seguintes eventos:

PAPI_BR_INS – Instruções de salto

PAPI_BR_MSP – Saltos erradamente previstos

PAPI_TOT_IIS – Instruções iniciadas

- a) Recolha dados para a execução da rotina `convolve3x1`, utilizando uma imagem de 256x256 (utilize como grau de optimização `-O3`).

```
./convolve AC_images/abe_natsumi256.pgm result.ppm 1
```

Explique a diferença entre o número de instruções iniciadas (contador `PAPI_TOT_IIS`) e o total de instruções efetivamente executadas. Explique também o valor obtido para o número de saltos erradamente previstos.

- b) Efetue um desdobramento do ciclo, efectuado duas chamadas à rotina `kernel` por cada iteração do ciclo (assuma que o número de iterações é sempre múltiplo de 2). Obtenha novamente as métricas recolhidas em a). Que conclusão pode retirar? (nota: compare o número de #I e #CC e calcule o CPE e #I por Elemento de cada uma das versões).
- c) Compile agora a rotina `convolve3x1.cpp` com a opção `-msse4` para gerar código com operações vectoriais. Para que o compilador gere operações vectoriais deve trocar a ordem dos ciclos X e Y¹. Explique porque é que sem esta troca o compilador não gera instruções vectoriais. Verifique que o *assembly* gerado contém instruções vectoriais, localizando instruções que utilizam o registo vectorial `xmm0`. Utilize o seguinte comando para gerar o *assembly* da rotina `convolve3x1.cpp`:

```
gcc -O3 -msse4 -S convolve3x1.cpp
```

- d) Analisando o *assembly* gerado, verifica-se que também é gerada uma versão do código sem instruções vectoriais e que um conjunto de testes aos parâmetros da rotina permite optar por uma ou outra versão. Explique porquê esta necessidade.
- e) Compare #I, #CC, o número de saltos (`PAPI_BR_INS`) e o número de instruções vectoriais executadas (`PAPI_VEC_INS`) com e sem a opção `-msse4` e explique as diferenças obtidas.

¹ Devido a limitações do compilador gcc utilizado no laboratório, deve também iniciar o ciclo X em 2 e remover a divisão por 3, caso contrário o compilador não irá gerar instruções vectoriais.