

# Desenvolvimento de Código Otimizado

Alyson Matheus Maruyama Nascimento - 8532269

Atividade 5

## **Vetorização (Aula 7)**



**Universidade de São Paulo - São Carlos**

# Programa Escolhido

O programa utilizado nessa atividade encontra-se no arquivo *matmul.c* e é responsável por realizar uma multiplicação de matrizes. Para isso, o programa recebe por argumento (*argv*) um único valor inteiro (quantidade de linhas e colunas das matrizes), popula cada uma das matrizes com valores aleatórios sorteados, e realiza a multiplicação. Vale notar que ambas as matrizes possuem mesma dimensão e são quadradas.

## Compilação e Execução

O programa foi compilado e executado diversas vezes, considerando diferentes *flags* de compilação e mantendo a mesma arquitetura. Para cada vez que foi compilado, o programa foi executado com o mesmo parâmetro através do *perf* e as estatísticas retornadas foram salvas na pasta *results/*, entregue juntamente com o arquivo comprimido dessa atividade.

Todos os comandos de compilação e execução utilizados encontram-se dentro do arquivo *compile\_exec.sh*, que corresponde ao script criado para essa atividade e que também foi entregue no arquivo comprimido junto à esse documento.

## Flags

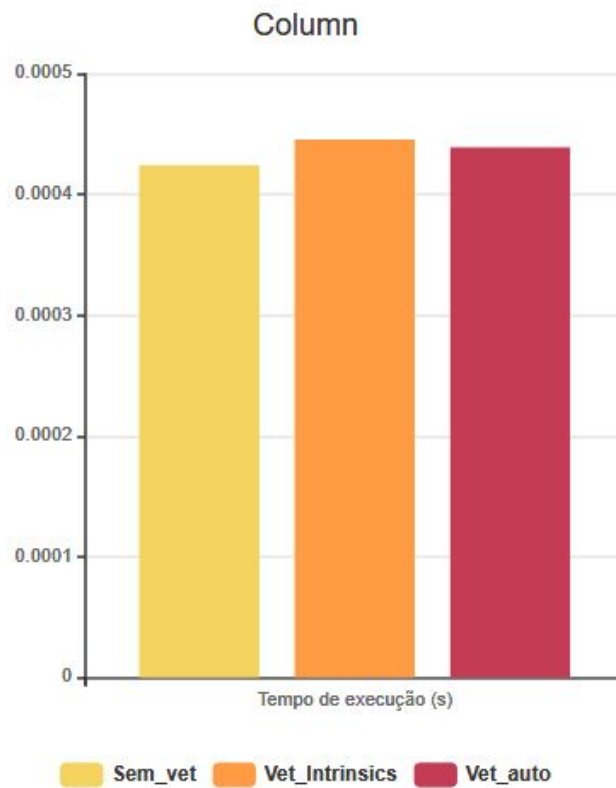
Além da flag de otimização *O3* utilizada em todas as compilações, temos as seguintes flags de vetorização:

- Nenhuma (Sem vetorização)
- *mavx2* (vetorização *intrinsics*)
- *ftree-vectorize*

## Resultados

Como citado anteriormente, as estatísticas retornadas pelo comando *perf stat* encontram-se na pasta *results/*. Os gráficos abaixo resumem as informações

coletadas e demonstra as diferenças de tempo de execução, acessos à cache e *cache-misses* entre cada compilação.



*Figura 1: diferenças de tempo de execução*

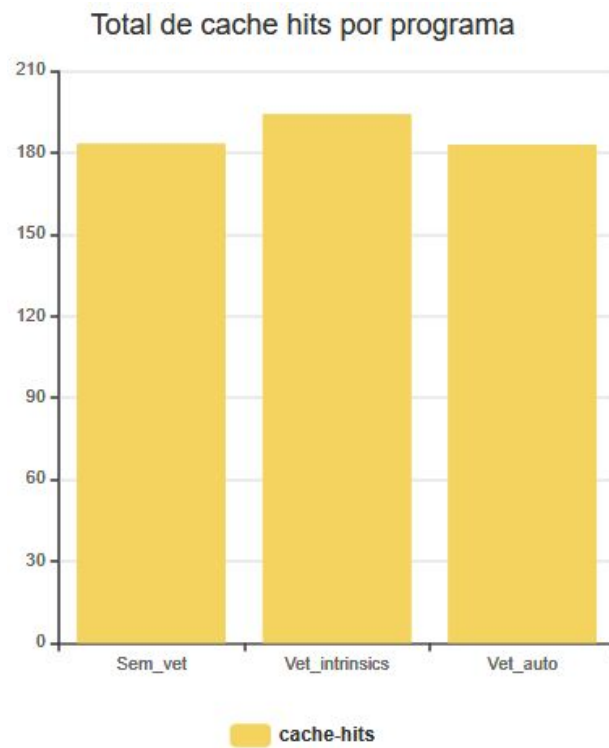


Figura 2: total de cache hits por programa

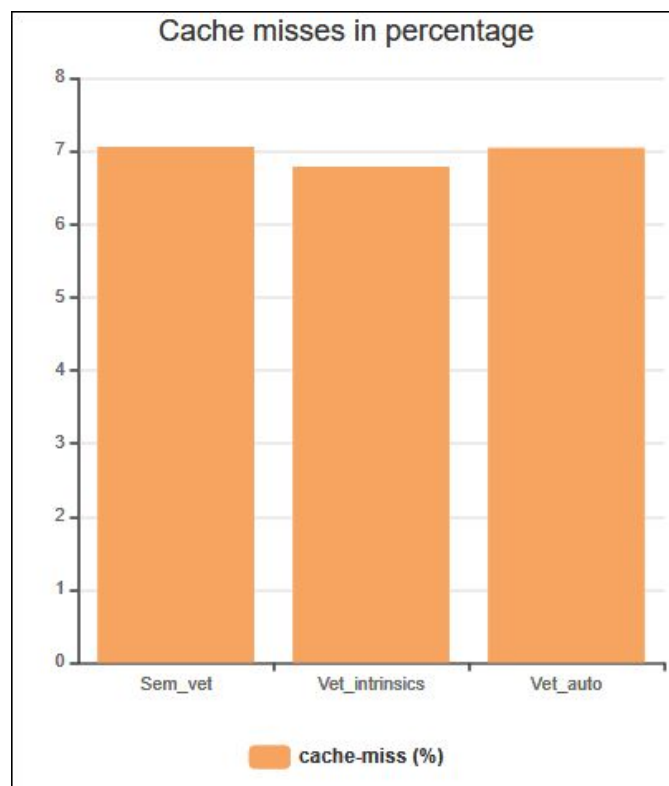


Figura 3: porcentagem de cache misses quando comparada ao total de hits

Dentre os dados coletados, podemos destacar que a vetorização através das instruções *intrinsic* obteve maior quantidade de *cache hits*, e consequentemente menor quantidade de *misses* quando comparada com as outras. Entretanto, apesar desse ganho, ele não foi refletido no tempo de execução do programa