

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO

Vetorização do algoritmo de Decomposição LU

Discentes: Flávio Henrique Lopes Barbosa, Jaysa Keylla Siqueira Barbosa, José Augusto Agripino de Oliveira.

1. Introdução

Entendemos a vetorização como o uso de código otimizado, pré-compilado e escrito em uma linguagem de nível baixo, que é usado para performar operações matemáticas em uma determinada sequência de dados. Para utilizar vetorização em C, a priori, será utilizada uma ferramenta do compilador GCC, que é a auto-vetorização de códigos. Em posse dele, poderá ser feita a comparação com e sem a auto-vetorização do código.

2. Metodologia

Esse recurso de vetorização pode ser explorado por meio de um processo de vetorização automática provido por compiladores como GCC (GNU Compiler Collection) ou ICC (Intel C/C++ Compiler). Quando a opção de vetorização automática é ativada, o compilador procura por estruturas de código que possam ser vetorizadas, ou seja, por laços internos que não apresentem, por exemplo, controle de fluxo ou dependência de dados entre os elementos dos vetores que estão sendo processados. Portanto, a vetorização automática é transparente e não requer qualquer esforço por parte do programador.

1. `$ gcc -pg -O3 -fno-free-vectorize -fopt-info-vec-missed -fopt-info-vec-optimized decomposicaoLU.c -o decomposicaoLU`
2. `$ gprof decomposicaoLU > ordem_n.txt`
3. `$ gprof decomposicaoLU | gprof2dot > out_n.dot`

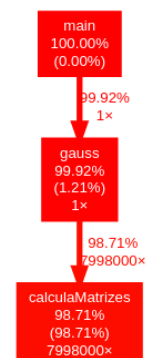
3. Resultados

3.1 Saída do programa para matriz quadrada de ordem 4000

Antes da vetorização:

Each sample counts as 0.01 seconds.

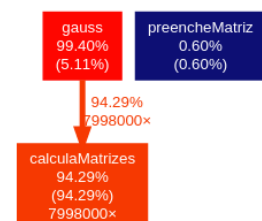
%	cumulative	self		self	total	
time	seconds	seconds	calls	s/call	s/call	name
98.71	72.73	72.73	7998000	0.00	0.00	calculaMatrizes
1.21	73.62	0.89	1	0.89	73.62	gauss
0.08	73.68	0.06	1	0.06	0.06	preencheMatriz



Depois da vetorização:

Each sample counts as 0.01 seconds.

%	cumulative	self		self	total	
time	seconds	seconds	calls	us/call	us/call	name
94.97	9.41	9.41	7998000	1.18	1.18	calculaMatrizes
5.19	9.93	0.51				gauss
0.61	9.99	0.06				preencheMatriz

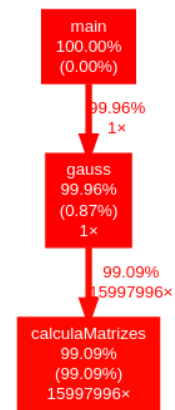


3.2 Saída do programa para matriz quadrada de ordem 5657

Antes da vetorização:

Each sample counts as 0.01 seconds.

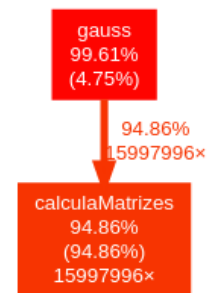
%	cumulative	self		self	total	
time	seconds	seconds	calls	s/call	s/call	name
99.09	213.03	213.03	15997996	0.00	0.00	calculaMatrizes
0.87	214.90	1.87	1	1.87	214.90	gauss
0.03	214.97	0.07	1	0.07	0.07	preencheMatriz
0.00	214.98	0.01				_init



Depois da vetorização:

Each sample counts as 0.01 seconds.

%	cumulative	self		self	total	
time	seconds	seconds	calls	us/call	us/call	name
95.57	21.96	21.96	15997996	1.37	1.37	calculaMatrizes
4.78	23.06	1.10				gauss
0.39	23.15	0.09				preencheMatriz

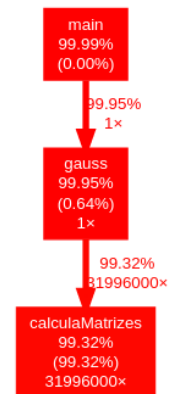


3.3 Saída do programa para matriz quadrada de ordem 8000

Antes da vetorização:

Each sample counts as 0.01 seconds.

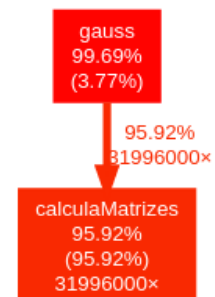
%	cumulative	self		self	total	
time	seconds	seconds	calls	s/call	s/call	name
99.32	593.13	593.13	31996000	0.00	0.00	calculaMatrizes
0.64	596.93	3.80	1	3.80	596.93	gauss
0.04	597.17	0.24	1	0.24	0.24	preencheMatriz
0.01	597.20	0.03				_init



Depois da vetorização:

Each sample counts as 0.01 seconds.

%	cumulative	self		self	total	
time	seconds	seconds	calls	us/call	us/call	name
96.65	67.89	67.89	31996000	2.12	2.12	calculaMatrizes
3.80	70.56	2.67				gauss
0.32	70.79	0.22				preencheMatriz

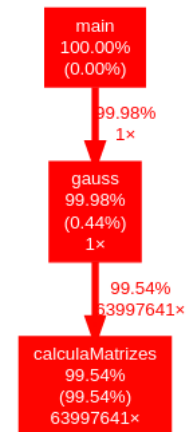


3.4 Saída do programa para matriz quadrada de ordem 11314

Antes da vetorização:

Each sample counts as 0.01 seconds.

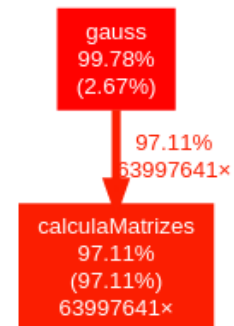
%	cumulative	self		self	total	
time	seconds	seconds	calls	Ks/call	Ks/call	name
99.54	1679.62	1679.62	63997641	0.00	0.00	calculaMatrizes
0.44	1687.05	7.43	1	0.01	1.69	gauss
0.02	1687.33	0.28	1	0.00	0.00	preencheMatriz
0.00	1687.37	0.04				_init



Depois da vetorização:

Each sample counts as 0.01 seconds.

%	cumulative	self		self	total	
time	seconds	seconds	calls	us/call	us/call	name
97.83	192.50	192.50	63997641	3.01	3.01	calculaMatrizes
2.69	197.79	5.29				gauss
0.22	198.21	0.42				preencheMatriz
0.01	198.22	0.01				__libc_csu_init

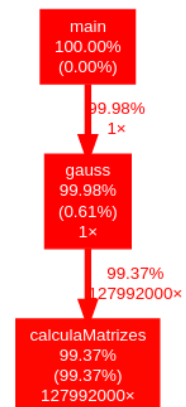


3.5 Saída do programa para matriz quadrada de ordem 16000

Antes da vetorização:

Each sample counts as 0.01 seconds.

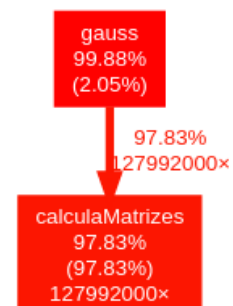
%	cumulative	self		self	total	
time	seconds	seconds	calls	Ks/call	Ks/call	name
99.37	2706.73	2706.73	127992000	0.00	0.00	calculaMatrizes
0.61	2723.35	16.62	1	0.02	2.72	gauss
0.02	2723.78	0.43	1	0.00	0.00	preencheMatriz
0.00	2723.84	0.06				_init



Depois da vetorização:

Each sample counts as 0.01 seconds.

%	cumulative	self		self	total	
time	seconds	seconds	calls	us/call	us/call	name
98.58	560.87	560.87	127992000	4.38	4.38	calculaMatrizes
2.06	572.60	11.73				gauss
0.12	573.31	0.71				preencheMatriz



4. Conclusões

Os resultados obtidos com a auto-vetorização foram muito melhores que os resultados sem a auto-vetorização. O gargalo, que é a função *calculaMatrizes*, teve um tempo muito menor de execução com a auto-vetorização. Conforme a mensagem explicitada pelo GCC após as flags de auto vetorização, faltaram laços de repetição para serem vetorizados.

-Laços das linhas 20 e 21;

```
19 void preencheMatriz(float *A) {
20     for(int i=0; i<n; i++) {
21         for(int j=0; j<n; j++) {
22             A[(i*n)+j] = rand()%100000;
23         }
24     }
25 }
```

-Laços das linhas 48 e 50;

```
46 //Calculando os multiplicadores da Gauss
47 //Movimenta a coluna
48 for(int j=0; j<(n-1); j++){
49     //Movimenta a linha
50     for(int i=j+1; i<n; i++){
51         ( aux[(i*n)+j] != 0 ) ? calculaMatrizes(i, j, aux, M) : NULL;
52     } //Todos os multiplicadores da coluna j estão definidos
53 }
```

Sendo assim, é necessário modificar o código para que seja possível vetorizar o que restou. Os laços 20 e 21 não podem ser vetorizados por existir um `rand()` no escopo. O GCC retorna que o loop da linha 48 não pode ser vetorizado por existir vários loops aninhados. Por fim, o laço da linha 50 possui controle de fluxo.

5. Referências

LOSS, Flavio. **Otimizando códigos de Data Science em Python com Vetorização**. 2020. Disponível em: <https://medium.com/geleia/otimizando-c%C3%B3digos-de-data-science-em-python-com-vetoriza%C3%A7%C3%A3o-b5cd9188530c>. Acesso em: 24 out. 2022.