



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

## **AVALIAÇÃO DA CAPACIDADE DE VETORIZAÇÃO DE TRÊS COMPILADORES C DISPONÍVEIS NO SUPERCOMPUTADOR SANTOS DUMONT**

Leonardo Sattler Cassará

Projeto Final apresentado aos Professores Celso L. Mendes e Stephan Stephany como parte da avaliação do curso CAP-399.

Repositório do projeto:

[<github/CAP-399>](https://github.com/leonardocassar/CAP-399)

INPE

São José dos Campos

18 de dezembro de 2020

## RESUMO

No presente trabalho, os compiladores Intel v.19.1, PGI v.19.4 e GNU v.8.3 são avaliados quanto à capacidade de vetorização. Foi utilizada uma versão ligeiramente modificada da coleção de testes TSVC (Test Suite for Vectorizing Compilers) descrita em [Maleki et al. \(2011\)](#). Este manuscrito está dividido em três seções: na Seção I, os resultados da bateria de testes são apresentados, caracterizando os ganhos e desempenhos obtidos pela vetorização de cada compilador; na Seção II, as flags de compilação são alteradas de modo a buscar *Speedups* superiores aos obtido durante os testes da primeira seção; as conclusões são apresentadas na Seção III.

## LISTA DE TABELAS

	<u>Pág.</u>
I.1 Tempos de execução em segundos obtidos em dez dos 151 loops da bateria de testes. . . . .	1
I.2 <i>Speedups</i> de dez dos 151 loops da bateria de testes. . . . .	2
I.3 Médias dos 151 <i>Speedups</i> resultantes dos testes dos três compiladores. . .	3
I.4 Somas dos tempos obtidos nos 151 loops das seis execuções. . . . .	3
I.5 Número de loops vetorizados e não-vetorizados (dentre os 151 loops de teste) por cada compilador. . . . .	3
I.6 Nomes dos loops que não foram vetorizados por nenhum compilador. O total é igual a 45. . . . .	4
II.1 Flags de compilação utilizadas nas Seções I e II. . . . .	5
II.2 <i>Speedups</i> médios após as novas flags de compilação e seus valores de $\delta$ com relação aos presentes na Tabela I.3. . . . .	6
II.3 Tempos totais com as novas flags de compilação e seus respectivos $\delta$ 's em termos dos valores da Tabela I.4. . . . .	6
II.4 Número de loops vetorizados e não-vetorizados (diferença com relação aos números da Tabela I.5 entre parênteses). . . . .	6

## LISTA DE FIGURAS

	<u>Pág.</u>
I.1 Histogramas dos resultados obtidos nas execuções sem vetorização (barras de cor laranja) e com vetorização (barras de cor azul) dos três compiladores testados. O número de bins é igual a 25. As estatísticas de cada distribuição estão indicadas, incluindo a média ( <b>mean</b> ), mediana ( <b>median</b> ), o desvio padrão ( <b>std</b> ), o coeficiente de assimetria ( <b>skew</b> ) e o coeficiente de curtose ( <b>kurt</b> ). . . . .	2
I.2 Diagrama de Venn do número de loops vetorizados e não-vetorizados pelos compiladores ICC, PGI e GCC ao longo dos testes realizados. . . . .	4
II.1 Histogramas e estatísticas dos resultados obtidos com as novas flags de compilação. O número de bins é igual a 25. . . . .	6

## SUMÁRIO

	<u>Pág.</u>
<b>PREFÁCIO</b> . . . . .	<b>iii</b>
<b>I APRESENTAÇÃO DOS RESULTADOS</b> . . . . .	<b>1</b>
a) Tempos de cada execução . . . . .	1
b) <i>Speedups</i> de cada execução . . . . .	2
c) <i>Speedups</i> médios . . . . .	3
d) Tempos totais . . . . .	3
e) Número de loops vetorizados e não-vetorizados . . . . .	3
f) Loops não-vetorizados . . . . .	4
<b>II BUSCANDO MAIORES <i>SPEEDUPS</i></b> . . . . .	<b>5</b>
<b>III CONCLUSÕES</b> . . . . .	<b>7</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> . . . . .	<b>8</b>

## PREFÁCIO

Os resultados deste trabalho foram gerados no supercomputador Santos Dumont sob usuário *leonardo.cassara*. Todos os arquivos referentes à geração dos resultados se encontram divididos em seis diretórios: três referentes à Seção I deste projeto, GCC\_1, ICC\_1, PGI\_1 (e referentes aos compiladores GNU v.8.3, Intel v.19.1 e PGI v.19.4, respectivamente), e três que dizem respeito à Seção II deste projeto, GCC\_2, ICC\_2, PGI\_2 (e referentes aos mesmos compiladores). Cada diretório contém seus respectivos scripts de submissão, outputs, makefiles, os programas de teste e outros arquivos necessários para a geração dos resultados. Tais diretórios se encontram no seguinte local do Santos Dumont:

```
/scratch/padinpe/leonardo.cassara/PSMP_scratch/Projeto/
```

A análise dos resultados foi realizada localmente (em um computador pessoal) através da linguagem Python, com uso das bibliotecas *numpy* e *pandas* para geração das tabelas e *matplotlib* para criação das figuras. Todos os arquivos referentes à análise dos resultados estão no repositório deste projeto. O conteúdo do repositório é descrito abaixo:

- pasta **sdumont**: cópia dos seis diretórios presentes no Santos Dumont, disponibilizando os resultados para o público sem acesso ao supercomputador.
- pasta **Relatorio**: contém os arquivos  $\text{\LaTeX}$  utilizados para a geração deste manuscrito.
- *data\_handler.py*: script de análise dos resultados, que gera as tabelas e as figuras presentes neste manuscrito, bem como uma planilha com os resultados.
- *data\_comp.py*: script para comparação dos resultados, que gera tabelas com a alteração relativa entre as quantidades contrastadas.
- *stats\_tools.py*: script para cálculo das estatística dos resultados, utilizado pelos scripts *data\_handler.py* e *data\_comp.py*.
- *results1.xlsx*: planilha contendo os resultados apresentados nas Tabelas I.1 e I.2 deste manuscrito.
- *results2.xlsx*: planilha contendo os resultados da Seção II.

## I APRESENTAÇÃO DOS RESULTADOS

A bateria de testes realizada neste projeto consistiu de seis execuções: duas para cada um dos três compiladores, sendo a primeira com e a segunda sem vetorização ativada durante a compilação (ver Tabela II.1 para detalhes das flags de compilação utilizadas em cada caso). Os resultados obtidos na bateria de testes são apresentados a seguir para os três compiladores testados: ICC (versão 19.1), PGI (versão 19.1) e GCC (versão 8.3).

### a) Tempos de cada execução

A Tabela I.1 exibe os tempos obtidos em dez dos 151 loops de teste (cinco primeiros e cinco últimos) durante as seis execuções da bateria de testes.

Tabela I.1 - Tempos de execução em segundos obtidos em dez dos 151 loops da bateria de testes.

Loop	Tempos de execução [s]					
	ICC		PGI		GCC	
	Não-vet.	Vet.	Não-vet.	Vet.	Não-vet.	Vet.
S000	0.414	0.169	0.360	0.338	0.540	0.170
S111	0.274	0.269	0.269	0.290	0.286	0.252
S1111	0.566	0.371	0.581	0.580	0.680	0.339
S112	0.628	0.631	0.295	0.334	0.811	0.436
S1112	0.616	0.243	0.232	0.267	0.808	0.329
⋮	⋮	⋮	⋮	⋮	⋮	⋮
vpvpv	1.150	0.627	0.664	0.628	1.257	0.684
vtvtv	1.156	0.630	0.673	0.630	1.268	0.685
vsumr	2.005	0.418	4.006	0.502	4.009	4.007
vdotr	2.055	0.904	4.013	0.890	4.013	4.010
vbor	0.133	0.089	0.358	0.045	0.358	0.090

A Figura I.1 exibe os histogramas dos tempos de execução obtidos com os três compiladores testados, para compilações sem vetorização (barras de cor laranja) e com vetorização (barras de cor azul). Os histogramas também apresentam as estatísticas de cada caso.

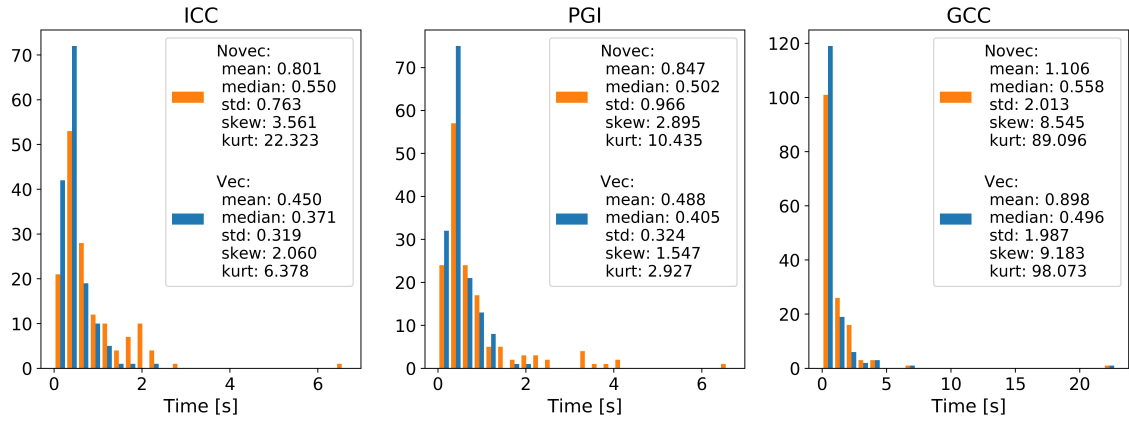


Figura I.1 - Histogramas dos resultados obtidos nas execuções sem vetorização (barras de cor laranja) e com vetorização (barras de cor azul) dos três compiladores testados. O número de bins é igual a 25. As estatísticas de cada distribuição estão indicadas, incluindo a média (**mean**), mediana (**median**), o desvio padrão (**std**), o coeficiente de assimetria (**skew**) e o coeficiente de curtose (**kurt**).

## b) *Speedups* de cada execução

A Tabela I.2 apresenta os *Speedups* de dez dos 151 loops para os três compiladores testados, obtidos através da Tabela I.1.

Tabela I.2 - *Speedups* de dez dos 151 loops da bateria de testes.

Loop	<i>Speedups</i>		
	ICC	PGI	GCC
S000	2.45	1.07	3.18
S111	1.02	0.93	1.13
S1111	1.53	1.00	2.01
S112	1.00	0.88	1.86
S1112	2.53	0.87	2.46
⋮	⋮	⋮	⋮
vpvpv	1.83	1.06	1.84
vtvtv	1.83	1.07	1.85
vsumr	4.80	7.98	1.00
vdotr	2.27	4.51	1.00
vbor	1.49	7.96	3.98

### c) *Speedups* médios

A partir da Tabela I.2, foi calculado o *Speedup* médio de cada compilador:

Tabela I.3 - Médias dos 151 *Speedups* resultantes dos testes dos três compiladores.

	ICC	PGI	GCC
<i>Speedup</i> médio	1.79	1.96	1.41

### d) Tempos totais

A partir da Tabela I.1, foi calculado o tempo total de cada execução da bateria de testes:

Tabela I.4 - Somas dos tempos obtidos nos 151 loops das seis execuções.

	ICC		PGI		GCC	
	Não-vet.	Vet.	Não-vet.	Vet.	Não-vet.	Vet.
Tempo total [s]	120.891	67.916	127.894	73.688	166.968	135.628

Com os valores da Tabela I.4, os *Speedups* dos compiladores ICC, PGI e GCC são, respectivamente, 1.78, 1.74 e 1.23 (comparar com os valores da Tabela I.3).

### e) Número de loops vetorizados e não-vetorizados

Através da Tabela I.2, e utilizando o limiar de 1.5 adotado em Maleki et al. (2011), foi obtido o número de loops vetorizados e não-vetorizados pelos três compiladores testados:

Tabela I.5 - Número de loops vetorizados e não-vetorizados (dentro os 151 loops de teste) por cada compilador.

ICC		PGI		GCC	
Não-vet.	Vet.	Não-vet.	Vet.	Não-vet.	Vet.
57	94	101	50	108	43

#### f) Loops não-vetorizados

Considerando o mesmo limiar de 1.5, foi determinado o número de loops não-vetorizados por nenhum dos três compiladores testados. Ao todo, 45 loops nunca foram vetorizados durante a bateria de testes. Tais loops são mostrados a seguir:

Tabela I.6 - Nomes dos loops que não foram vetorizados por nenhum compilador. O total é igual a 45.

S111	S1113	S114	S1115	S118	S123	S126	S141	S161
S162	S171	S172	S232	S1232	S252	S255	S256	S257
S258	S277	S281	S293	S2101	S2111	S31111	S3110	S3112
S321	S322	S323	S332	S341	S342	S343	S353	S481
S482	S491	S4112	S4113	S4114	S4115	S4116	vag	vas

A Figura I.2 resume a distribuição de loops vetorizados e não-vetorizados pelos três compiladores ao longo dos testes.

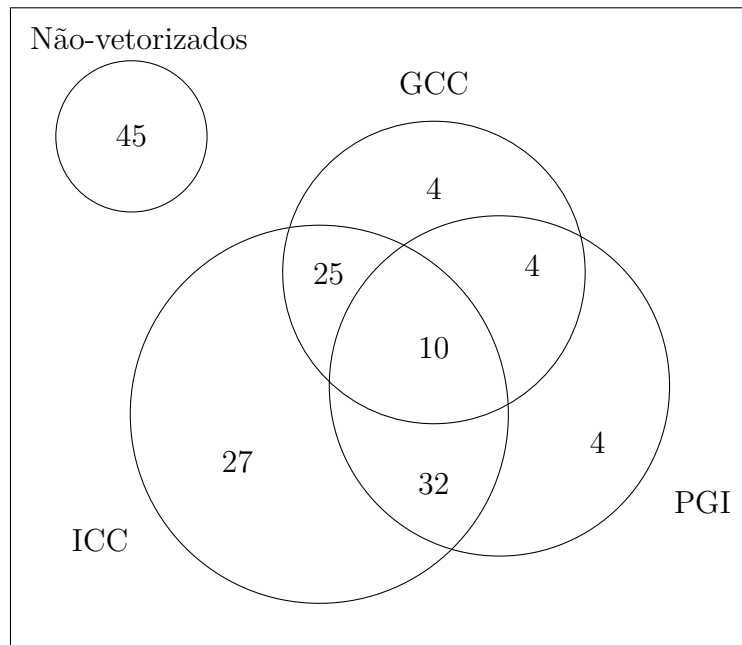


Figura I.2 - Diagrama de Venn do número de loops vetorizados e não-vetorizados pelos compiladores ICC, PGI e GCC ao longo dos testes realizados.



## II BUSCANDO MAIORES *SPEEDUPS*

A Tabela II.1 detalha as flags de compilação utilizadas para gerar os resultados das Seções I e II. A Figura II.1 ilustra os resultados das execuções com as novas flags. Os novos *Speedups* médios calculados estão na Tabela II.2; os novos tempos totais estão na Tabela II.3; a Tabela II.4 indica o número de loops vetorizados e não-vetorizados por cada compilador, com a informação de quantos loops a mais ou a menos foram vetorizados. Nas Tabelas II.2 e II.3 está indicada a alteração relativa  $\delta$  das quantidades apresentadas com relação aos resultados da Seção I, definida por:

$$\delta = \frac{x_{II} - x_I}{x_I}, \quad (\text{II.1})$$

sendo  $x_I$  e  $x_{II}$  os valores das quantidades pertinentes às Seções I e II, respectivamente. Ou seja, os valores de  $x_I$  são adotados como referência para quantificar a performance das novas otimizações.

Tabela II.1 - Flags de compilação utilizadas nas Seções I e II.

Flags		
ICC	Seção I	Seção II
Otimização base	<code>-std=c99 -O3</code>	
Vetorização	(ativada via <code>-O3</code> da otim. base)	<code>-xSSE4.2</code>
Sem vetorização	<code>-no-vec</code>	
Relatório de vet.	<code>-qopt-report=2 -qopt-report-phase=vec</code>	
PGI	Seção I	Seção II
Otimização base	<code>-c99 -O3</code>	<code>-c99 -O3 -Mcache_align</code>
Vetorização	<code>-Mvect=sse</code>	<code>-Mvect=sse -Mvect=prefetch</code>
Sem vetorização	<code>-Mnovect</code>	
Relatório de vet.	<code>-Minfo</code>	
GCC	Seção I	Seção II
Otimização base	<code>-O3</code>	<code>-O3 -ffast-math</code>
Vetorização	<code>-ftree-vectorize</code>	
Sem vetorização	<code>-fno-tree-vectorize</code>	
Relatório de vet.	<code>-fdump-tree-vect-blocks=report.dump</code>	

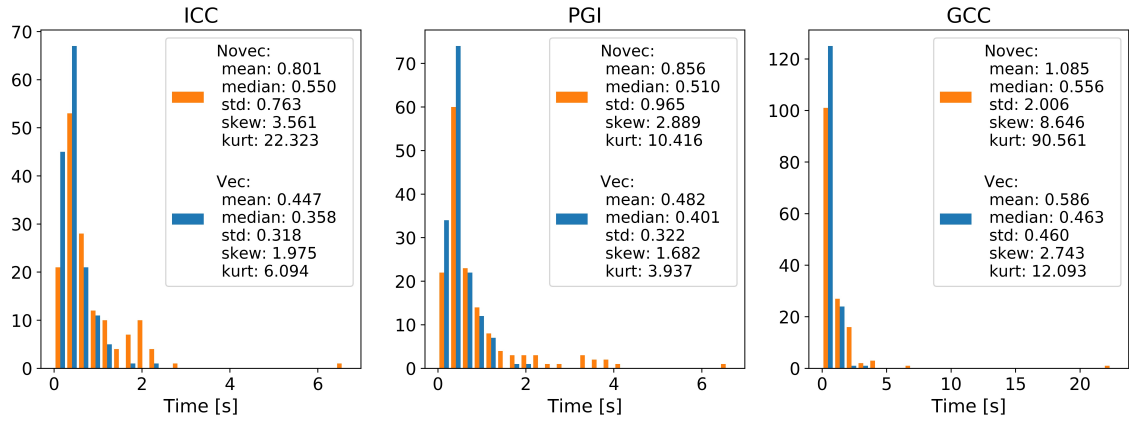


Figura II.1 - Histogramas e estatísticas dos resultados obtidos com as novas flags de compilação. O número de bins é igual a 25.

Tabela II.2 - *Speedups* médios após as novas flags de compilação e seus valores de  $\delta$  com relação aos presentes na Tabela I.3.

	ICC	PGI	GCC
<i>Speedup</i> médio	1.83	2.00	1.971
Alteração relativa ( $\delta$ )	+1.1%	+2.1%	+39.7%

Tabela II.3 - Tempos totais com as novas flags de compilação e seus respectivos  $\delta$ 's em termos dos valores da Tabela I.4.

	ICC		PGI		GCC	
	Não-vet.	Vet.	Não-vet.	Vet.	Não-vet.	Vet.
Tempo total [s]	120.891	67.476	129.241	72.834	163.855	88.499
Alteração relativa ( $\delta$ )	0%	-0.6%	+1%	-1.2%	-1.9%	-34.7%

Tabela II.4 - Número de loops vetorizados e não-vetorizados (diferença com relação aos números da Tabela I.5 entre parênteses).

ICC		PGI		GCC	
Não-vet.	Vet.	Não-vet.	Vet.	Não-vet.	Vet.
53	98 (+4)	99	52 (+2)	92	59 (+16)

### III CONCLUSÕES

#### Conclusões da Seção I

A partir dos resultados apresentados na Seção I, conclui-se sobre a performance individual dos compiladores:

- Os tempos médios (Figura I.1) e totais (Tabela I.4) das versões vetorizada e não-vetorizada foram menores para o compilador ICC. Apesar dessa performance superior no geral, ele obteve um *Speedup* médio intermediário (Tabela I.3); este foi o compilador que vetorizou o maior número de loops (Tabela I.5).
- O *Speedup* médio do compilador PGI foi o maior dentre os compiladores testados; enquanto seus tempos totais se aproximam dos valores do compilador ICC (Tabela I.4), seu número de loops vetorizados e não-vetorizados é comparável ao do GCC (Tabela I.5).
- Conforme indicado pela alta assimetria e curtose dos tempos obtidos com o GCC, a performance geral deste compilador foi penalizada pela presença de um loop que durou acima de 20 segundos (Figura I.1, plot da direita); este foi o compilador com menor *Speedup* médio e piores tempos totais.

Sobre a capacidade de vetorização dos três compiladores, o diagrama de Venn da Figura I.2 revela que:

- ICC foi o compilador que exclusivamente vetorizou o maior número de loops: 27 (17.9% do total), contra 4 (2.6% do total) tanto do PGI quanto do GCC.
- Somente 10 loops foram vetorizados por todos os compiladores, ou seja, 6.6% dos 151 loops de teste;
- 45 loops não foram vetorizados por nenhum compilador, representando 29.8% dos loops de teste.

#### Conclusões da Seção II

Com relação aos resultados obtidos na Seção II, destaca-se:

- ICC teve somente sua flag de vetorização alterada; PGI teve tanto sua flag de otimização base quanto a de vetorização alteradas; GCC teve somente sua flag de otimização base alterada (ver Tabela II.1).
- Todos os compiladores aumentaram seus *Speedups* médios; dito isto, GCC foi o compilador que apresentou a maior alteração nos resultados; de fato, seu *Speedup* médio superou o do ICC (Tabela II.2) e seu número de loops vetorizados superou o do PGI (Tabela II.4).
- A nova flag de otimização base utilizada para o compilador GCC melhorou a performance da versão não-vetorizada (redução de 1.9% do tempo total), enquanto ofereceu um ganho ainda maior para a versão vetorizada (redução de 34.7% do tempo total), conforme mostra a Tabela II.3; a Figura II.1, plot da direita, indica que a assimetria e a curtose diminuíram significativamente com a eliminação da amostra com mais de 20 segundos na versão vetorizada.
- PGI foi o compilador que obteve o menor número de novos loops vetorizados: +2, contra +4 do ICC e +16 do GCC (Tabela II.4).
- Mesmo o ICC apresentando a melhora menos significativa de *Speedup* médio, obtendo +1.1% contra +2.1% do PGI e +39.7% do GCC (Tabela II.2), ele permaneceu como o compilador com menores tempos tanto na versão vetorizada quanto na versão não-vetorizada (Figura II.1 e Tabela II.3).
- O número de loops vetorizados por todos os compiladores subiu de 10 para 24, que representa 15.8% do total.
- O número de loops não vetorizados por nenhum compilador caiu de 45 para 40, representando agora 26.8% do total. A interseção destes dois conjuntos é igual a 39, ou seja, os compiladores não foram capazes de vetorizar 39 loops em nenhum dos casos.

## REFERÊNCIAS BIBLIOGRÁFICAS

MALEKI, S.; GAO, Y.; GARZAR, M. J.; WONG, T.; PADUA, D. A. et al. An evaluation of vectorizing compilers. In: IEEE. **2011 International Conference on Parallel Architectures and Compilation Techniques**. [S.l.], 2011. p. 372–382. i, 3