

Comparação do Desempenho entre GCC e Clang

Acadêmico: Douglas Rorie Tanno¹
RA: 104156

Disciplina de Inteligência Artificial
Prof. Anderson Faustino da Silva

¹ra104156@uem.br

1. Introdução

É comum encontrar discussões em matérias, fóruns e artigos sobre qual é o melhor compilador, principalmente em relação ao GCC e o Clang. Entretanto, é difícil chegar a uma conclusão, pois ambos os compiladores possuem seus prós e contras, dependendo do seu uso.

Este trabalho tem como objetivo apresentar e comparar o desempenho de execução do GCC e do Clang para cinco algoritmos e níveis de otimização diferentes.

2. Experimentos

Para a execução dos experimentos, foram selecionados cinco algoritmos implementados em linguagem C, disponíveis no link¹. No site são apresentados diversos algoritmos que solucionam problemas comuns. Foram selecionados os algoritmos *Fannkuchredux*, *Fasta*, *Mandelbrot*, *Simple* e *Pidigits*. Os algoritmos foram executados com base no exemplo de entrada disponível no site:

- **Fannkuchredux**: entrada 12.
- **Fasta**: entrada 25000000.
- **Mandelbrot**: entrada 16000.
- **Simple**: entrada 16000.
- **Pidigits**: entrada 10000. Para a execução, foi necessário utilizar a biblioteca GMP, utilizando o comando `-lgmp`.

Os algoritmos foram executados com diferentes níveis de otimização: `-O0`, `-O1`, `-O2` e `-O3`. A otimização na compilação pode aumentar o desempenho da execução do algoritmo. O nível `-O0` não realiza nenhuma otimização, ocorrendo a compilação padrão de quando nenhum nível de otimização é especificado. A otimização aumenta conforme o nível aumenta, sendo o nível `-O1` equivalente a uma otimização mínima, `-O2` de otimização maior, e `-O3` de otimização maior ainda. Níveis superiores geram otimização equivalente a de `-O3` [Linux.die.net 2009].

O resultado de cada execução dos algoritmos foi redirecionado o arquivo `/dev/null`. Este arquivo tem a propriedade de sempre estar vazio, ou seja, quaisquer dados enviados para `/dev/null` são descartados. Isso foi feito pois o processo de imprimir o resultado no prompt de comando demanda mais tempo de execução, sendo que gravá-lo em um arquivo é um processo mais rápido. O arquivo vazio foi utilizado para evitar a geração de arquivos de texto desnecessários na máquina.

¹<https://benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/c.html>

Além disso, para obter o tempo de execução de cada algoritmo, foi utilizado o comando `time`, que retorna o tempo de execução real, `user` e `sys` em minutos e segundos. O tempo utilizado para o trabalho foi o real.

Cada algoritmo foi compilado pelo GCC e pelo Clang 10 vezes para cada nível de otimização. Após o término dos experimentos, foi calculada a média aritmética simples dessas 10 execuções.

2.1. Plataforma

Para a execução do trabalho, foi utilizado um desktop com as seguintes características:

Processador Intel(R) Core(TM) i5 CPU 760 2.80 GHz; Memória RAM 4GB; Disco rígido de 1000GB; Sistema operacional Windows 10 Pro 64 bits.

Durante a compilação e execução dos algoritmos, foi utilizada a máquina virtual Oracle VM VirtualBox 6.0, emulando a imagem do Ubuntu versão 20.04.3.

Foram instalados e utilizados o GCC na versão 9.4.0 e Clang na versão 10.0.0-4ubuntu1.

3. Resultados

Os resultados do tempo das execuções, em segundos, foram organizados de acordo com cada algoritmo utilizado durante os experimentos. Além disso, foi calculada a média aritmética e o desvio padrão (DP).

3.1. Fannkuchredux

Os resultados de todos experimentos com o algoritmo *Fannkuchredux* estão apresentados na Tabela 1. Já as médias dos resultados dos experimentos estão representadas na Figura 1.

Tabela 1. Resultados dos experimentos com o algoritmo Fannkuchredux

	GCC				Clang			
	-O0	-O1	-O2	-O3	-O0	-O1	-O2	-O3
1	126.705	49.223	47.429	25.495	112.316	42.654	41.528	42.602
2	126.271	50.259	48.240	25.582	109.241	42.354	41.346	47.154
3	133.526	46.422	46.539	26.593	113.068	41.165	41.405	42.567
4	126.000	46.531	46.471	24.636	108.708	41.295	41.930	41.528
5	126.018	46.358	47.117	24.392	108.884	41.297	42.500	41.786
6	130.253	47.273	47.570	24.434	108.731	41.385	41.516	41.891
7	129.612	51.739	45.343	24.693	108.747	41.807	41.221	41.861
8	126.521	46.065	44.647	24.366	108.961	41.620	45.042	45.616
9	127.560	46.167	44.969	24.788	108.819	45.140	41.375	41.563
10	127.335	45.849	44.912	24.674	108.705	42.104	41.138	41.452
DP	2.316	1.962	1.214	0.677	1.554	1.123	1.113	1.862

Pode-se verificar na Tabela 1 que os tempos de execução que mais variaram, respectivamente, dentre as 10 execuções realizadas, foram os de nível -O0 e -O1 do GCC, seguidos pelo nível -O3 do Clang.

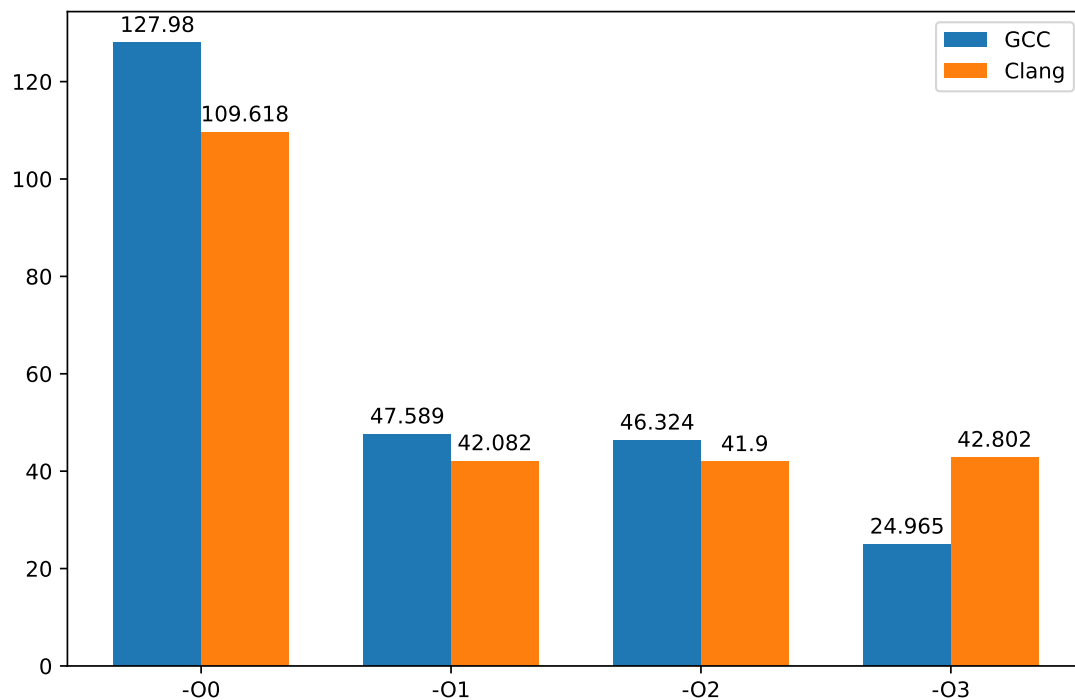


Figura 1. Média do tempo de execução, em segundos, do algoritmo Fannkuchredux, para cada nível de otimização de compilação do GCC e Clang.

De acordo com a Figura 1, pode-se verificar que para o nível -O0, as execuções compiladas com GCC teve, em média, tempo de execução 16.75% superior às com Clang. O tempo de execução com o GCC também se manteve superior ao com o Clang para os níveis -O1 e -O2, de 13.08% e 10.56%, respectivamente. Contudo, para o nível -O3, o GCC teve maior desempenho que o Clang, possuindo tempo de execução, em média, 41.67% mais rápido.

3.2. Fasta

Os resultados de todos experimentos com o algoritmo *Fasta* estão apresentados na Tabela 2. Já as médias dos resultados dos experimentos estão representadas na Figura 2.

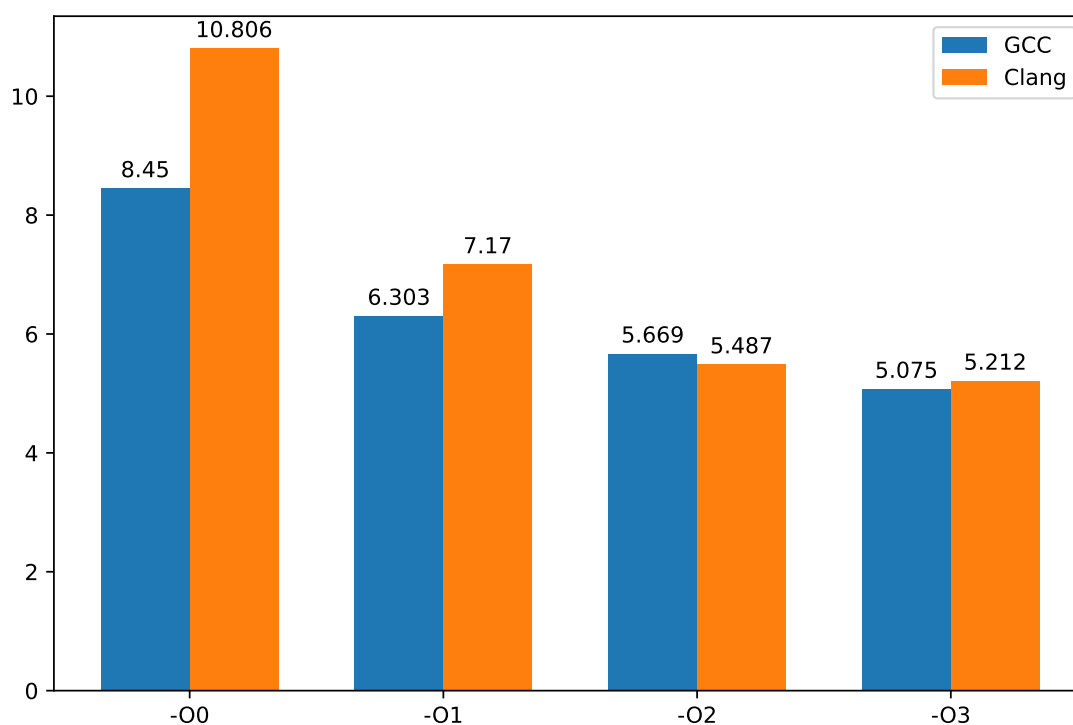
Com base na Tabela 2, pode-se verificar que não houve tanta variação no tempo de execução entre os experimentos, sendo as variações mais acentuadas, mesmo que pequenas, para os níveis -O1, -O0 e -O3 do GCC, respectivamente, seguidas pelo nível -O0 do Clang.

De acordo com a Figura 2, pode-se identificar que as execuções através da compilação com o GCC tiveram tempo menor do que as com o Clang para os níveis -O0, -O1 e -O3, sendo, respectivamente, 27.88%, 13.76% e 2.7% mais rápidas. Para o nível -O3, o GCC obteve tempo de execução médio de 3.32% superior ao com o Clang. Contudo, mesmo havendo diferenças mínimas, não foi verificada variação de desempenho significativa para ambos compiladores, em níveis de otimização -O2 e -O3, para o algoritmo *Fasta*.

Tabela 2. Resultados dos experimentos com o algoritmo Fasta.

	GCC				Clang			
	-O0	-O1	-O2	-O3	-O0	-O1	-O2	-O3
1	8.606	6.702	5.410	5.048	11.242	6.542	5.706	5.643
2	8.456	6.552	5.618	5.088	10.782	6.551	5.488	5.238
3	8.432	6.435	5.399	5.037	10.745	9.035	5.420	5.207
4	8.394	6.686	7.644	5.060	10.813	10.521	5.433	5.175
5	8.411	6.160	5.421	5.230	10.739	6.973	5.700	5.163
6	8.397	6.111	5.531	5.085	10.758	6.538	5.768	5.132
7	8.511	6.125	5.376	5.027	10.695	6.468	5.375	5.137
8	8.379	6.183	5.416	5.069	10.772	6.317	5.347	5.140
9	8.493	6.068	5.411	5.043	10.697	6.375	5.314	5.151
10	8.418	6.006	5.462	5.061	10.817	6.375	5.319	5.135
DP	0.476	0.670	0.162	0.344	0.220	0.094	0.178	0.136

Figura 2. Média do tempo de execução, em segundos, do algoritmo Fasta, para cada nível de otimização de compilação do GCC e Clang.



3.3. Mandelbrot

Os resultados de todos experimentos de execução do algoritmo *Mandelbrot* estão apresentados na Tabela 3. Já as médias dos resultados dos experimentos estão representadas na Figura 3.

Tabela 3. Resultados dos experimentos com o algoritmo Mandelbrot.

	GCC				Clang			
	-O0	-O1	-O2	-O3	-O0	-O1	-O2	-O3
1	29.263	17.125	13.846	14.227	26.606	14.906	15.116	14.887
2	28.808	15.194	14.030	13.843	26.532	14.768	14.799	14.812
3	28.017	15.504	13.962	14.338	26.524	14.696	14.858	15.059
4	28.212	15.009	13.769	14.579	26.511	14.772	14.788	14.815
5	28.541	14.828	13.616	14.946	26.618	14.741	14.780	14.755
6	27.804	15.598	13.659	13.983	26.503	14.758	14.743	14.750
7	27.874	14.851	13.563	14.580	26.460	14.756	14.834	14.798
8	27.968	14.872	13.557	14.656	26.662	15.027	14.888	14.698
9	27.773	14.955	13.590	14.337	27.221	14.762	15.347	14.725
10	27.856	14.844	13.630	13.880	26.381	14.727	14.841	15.129
DP	0.014	0.058	0.179	0.021	0.014	0.012	0.010	0.016

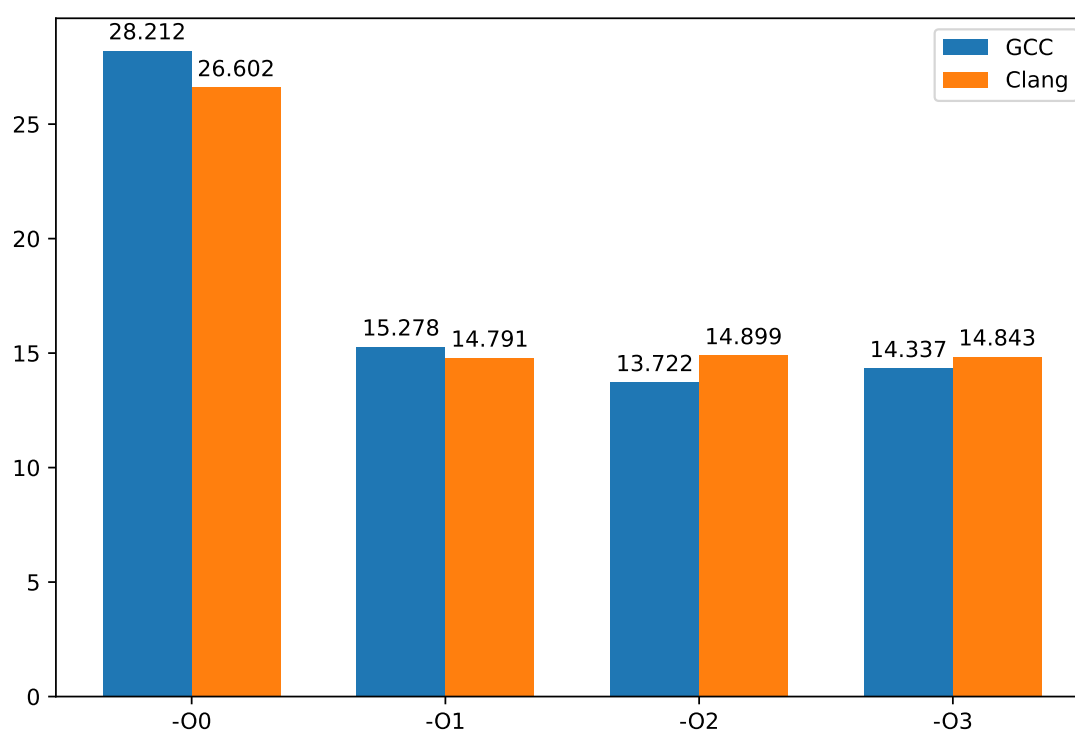


Figura 3. Média do tempo de execução, em segundos, do algoritmo Mandelbrot, para cada nível de otimização de compilação do GCC e Clang.

Baseando-se nos resultados apresentados na Tabela 3, pode-se verificar que não houveram tantas diferenças no tempo de execução entre os experimentos, onde as mais acentuadas ocorreram, respectivamente, para os níveis -O2 e -O1 utilizando o GCC.

De acordo com a Figura 3, também pode-se conferir que as diferenças no tempo de execução entre os compiladores não foram tão grandes. Os experimentos com GCC obtiveram tempo de execução superiores aos com Clang para os níveis -O0 e -O1 em 6.05% e 3.29%, respectivamente. Porém, para os níveis -O2 e -O3, os experimentos com

GCC tiveram tempo de execução inferiores aos com Clang, equivalente a 8.58% e 3.53%, respectivamente.

3.4. Simple

Os resultados de todos experimentos de execução do algoritmo *Simple* estão apresentados na Tabela 4. Já as médias dos resultados dos experimentos estão representadas na Figura 4.

Tabela 4. Resultados dos experimentos com o algoritmo Simple.

	GCC				Clang			
	-O0	-O1	-O2	-O3	-O0	-O1	-O2	-O3
1	55.652	31.935	29.338	32.946	62.284	27.082	27.390	27.371
2	55.553	33.739	28.985	28.900	57.885	27.001	27.219	31.099
3	55.160	32.601	28.770	28.881	57.651	27.066	27.442	26.979
4	57.748	34.964	28.916	28.698	57.614	27.085	27.858	27.057
5	59.274	32.873	28.854	28.564	57.418	29.369	27.352	27.055
6	57.109	32.860	28.807	28.646	57.453	27.187	27.197	27.646
7	56.439	32.687	28.705	28.517	57.222	27.194	27.307	30.858
8	54.756	32.694	28.841	28.829	57.670	27.350	27.120	27.039
9	54.781	32.960	29.039	28.762	57.801	27.611	27.125	27.270
10	54.962	32.890	28.996	28.910	57.906	27.190	27.227	27.139
DP	0.066	0.251	0.662	0.055	0.151	1.357	0.164	0.147

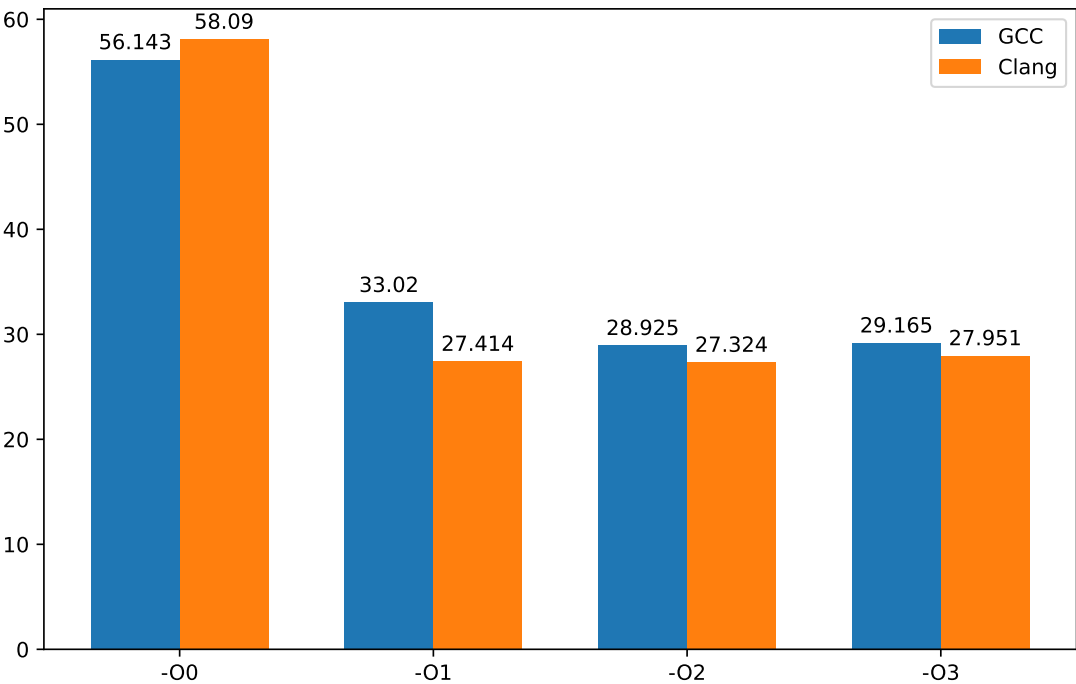


Figura 4. Média do tempo de execução, em segundos, do algoritmo Simple, para cada nível de otimização de compilação do GCC e Clang.

Pode-se verificar na Tabela 4 que os tempos de execução que mais variaram, dentre as 10 execuções realizadas, foram, respectivamente, os de nível -O1 com o Clang, e nível -O2 e -O1 com o GCC.

De acordo com a Figura 4, pode-se verificar que para o nível -O0, as execuções compiladas com GCC teve, em média, tempo de execução 3.47% inferior às compiladas com Clang. Contudo, as execuções com GCC tiveram tempo de execução superiores às com o Clang para os níveis -O1, -O2 e -O3 em 20.45%.

3.5. Pidigits

Os resultados de todos experimentos com o algoritmo *Pidigits* estão apresentados na Tabela 5. Já as médias dos resultados dos experimentos estão representadas na Figura 5.

Pode-se verificar na Tabela 5 que os tempos de execução que mais variaram, dentre as 10 execuções realizadas, foram, respectivamente, os de nível -O3 com o Clang, -O0 com o GCC, -O0 com o Clang, e -O3 com o GCC.

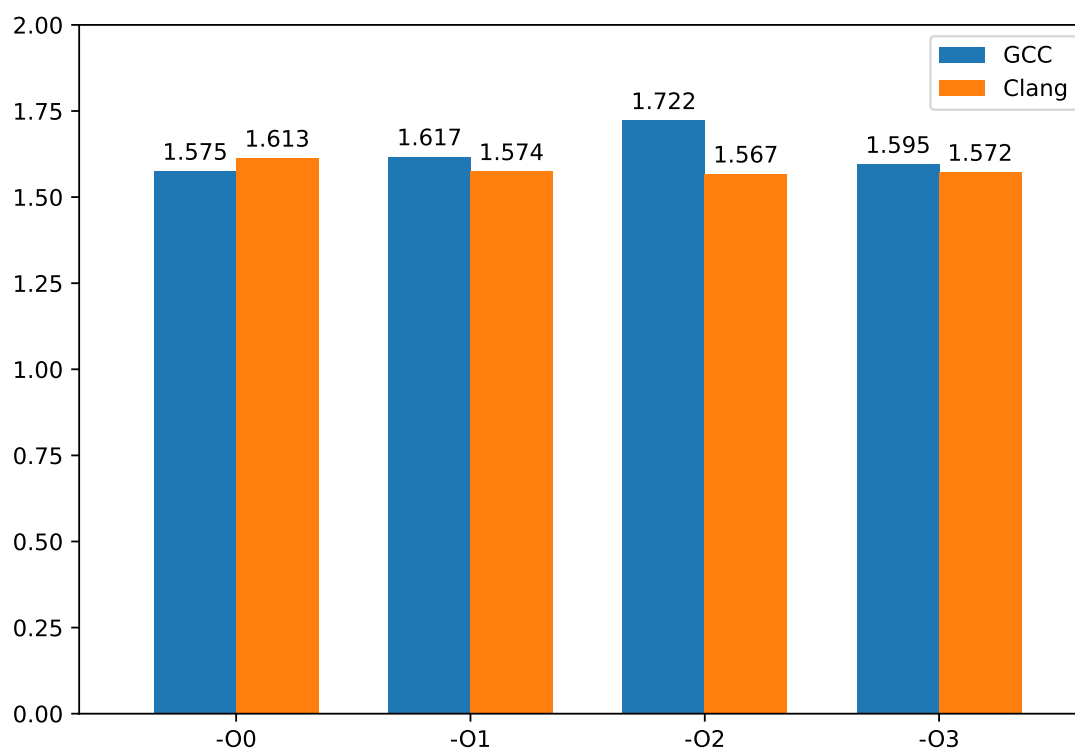


Figura 5. Média do tempo de execução, em segundos, do algoritmo Pidigits, para cada nível de otimização de compilação do GCC e Clang.

De acordo com a Figura 5, pode-se verificar que para o nível -O0, as execuções compiladas com GCC teve, em média, tempo de execução 2.41% inferior às com Clang. Entretanto, o tempo de execução com o GCC se manteve superior ao com o Clang para os níveis -O1, -O2 e -O3 em 2.73%, 9.89% e 1.46%, respectivamente.

Tabela 5. Resultados dos experimentos com o algoritmo Pidigits.

	GCC				Clang			
	-O0	-O1	-O2	-O3	-O0	-O1	-O2	-O3
1	1.602	1.676	1.991	1.634	1.637	1.598	1.592	1.614
2	1.581	1.717	1.554	1.572	1.614	1.579	1.558	1.573
3	1.562	1.563	2.065	1.619	1.591	1.564	1.562	1.564
4	1.558	1.580	1.911	1.584	1.621	1.556	1.564	1.564
5	1.585	1.586	1.605	1.558	1.590	1.566	1.557	1.560
6	1.554	1.565	1.625	1.585	1.622	1.591	1.555	1.559
7	1.582	1.713	1.602	1.597	1.608	1.569	1.571	1.589
8	1.584	1.609	1.617	1.607	1.619	1.574	1.565	1.562
9	1.565	1.571	1.635	1.600	1.606	1.577	1.569	1.566
10	1.578	1.591	1.614	1.595	1.623	1.562	1.575	1.567
DP	1.419	0.770	0.171	1.267	1.413	0.673	0.205	1.526

4. Considerações Finais

Com base nos resultados obtidos, pode-se concluir que, no geral, o desempenho em tempo de execução de algoritmos compilados com GCC ou Clang pode variar de acordo com o algoritmo utilizado. A diferença de tempo entre os dois compiladores mais acentuada ocorreu para o algoritmo *Fannkuchredux*, onde, no nível de otimização -O3, as execuções compiladas com o Clang demoraram 41.67% a mais que as com o GCC.

Em relação aos níveis de otimização utilizados na compilação, foi possível verificar que a diminuição no tempo de execução dos algoritmos também podem variar com o tipo de algoritmo utilizado. Para os algoritmos *Fannkuchredux*, *Fasta*, *Mandelbrot* e *Simple*, houve uma melhora significativa no tempo de execução, quando comparados os diversos níveis de otimização com o nível -O0, sem otimização. Além disso, foi possível analisar que a principal diferença no tempo de execução se dá principalmente no nível de otimização -O1, havendo poucas melhorias para os níveis subsequentes, como visto para o algoritmo *Mandelbrot*. Entretanto, a compilação com os diversos níveis de otimização não demonstrou efeitos positivos para o algoritmo *Pidigits*. Para o GCC, todos os níveis de otimização resultaram em maior tempo de execução que o nível -O0, principalmente para o -O2. Já para o Clang, houve pequenas melhorias para os níveis -O1 e -O2, porém o nível -O3 resultou em tempo de execução maior que utilizando o nível -O2.

Para trabalhos futuros, seria importante explorar uma gama maior de algoritmos, estudando suas características de implementação e o seu uso, com objetivo de mapear os cenários mais favoráveis para a compilação com o GCC ou o Clang, além de verificar se o uso de níveis de otimização na compilação se mostra eficiente.

Referências

Linux.die.net (2009). gcc(1) - linux man page. web. disponível em: <https://linux.die.net/man/1/gcc>.