Desenvolvimento de Código Otimizado

Alyson Matheus Maruyama Nascimento - 8532269

Atividade 4

Otimização do compilador

Universidade de São Paulo - São Carlos

Programa Escolhido

A atividade propõe escolher um dos programas do CLBG disponível em https://benchmarksgame-team.pages.debian.net/benchmarksgame/. Sendo assim, o programa escolhido foi o *fasta-gcc-2.c*, que pode ser encontrado no seguinte link:

https://benchmarksgame-team.pages.debian.net/benchmarksgame/program/fasta-gcc-2.html

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.

O programa principal corresponde ao arquivo *fasta.gcc-2.c*, contendo o código retirado do CLBG

Flags

As flags de compilação utilizadas foram:

Padrões:

- Nenhuma
- O1
- O2
- O3

Retiradas do *paper* "Finding Best Compiler Options for Critical Software Using Parallel Algorithms":

- fdce
- fif-conversion
- finline
- fpeephole2
- ftree-ch

Além disso, a flag *-march=core2* foi utilizada em todas as compilações (ler mais a respeito: https://gcc.gnu.org/onlinedocs/gcc/x86-Options.html)

Resultados

Como citado anteriormente, as estatísticas retornadas pelo comando *perf stat* encontram-se na pasta *results/*. O gráfico abaixo demonstra as diferenças de tempo de execução (em segundos) para cada flag utilizada.

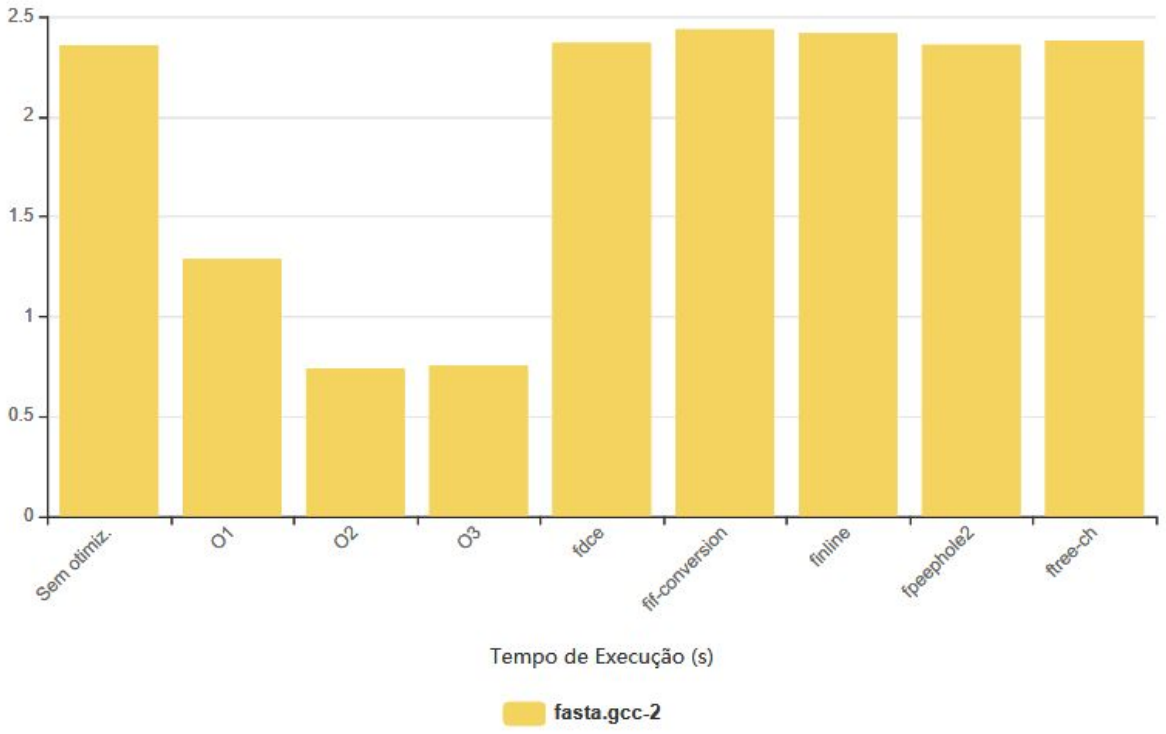

Figura 1: diferença de tempo de execução para cada flag

Conclusão

Ao analisar as saídas do *perf*, podemos perceber que o uso das flags do artigo *"Finding Best Compiler Options for Critical Software Using Parallel Algorithms"* teve praticamente nenhum ou pouco impacto em relação ao tempo

de execução do programa quando comparado com a compilação sem nenhuma flag de otimização.

Por outro lado, as flags O1, O2 e O3 alcançaram melhor desempenho em tempo, alcançando valores abaixo da metade do tempo de execução do programa original.