Universidade de São Paulo

SSC0951 – Desenvolvimento de Código Otimizado

Atividade 3

Nome	**Nº USP**
João Pedro Rodrigues Freitas	11316552
Rafael Kuhn Takano	11200459
Vinícius Santos Monteiro	11932463

Novembro
2022

1 Introdução

O objetivo dessa atividade é fazer a análise de desempenho do código abaixo e compilá-lo utilizando diferentes flags de otimização (sem otimização, O1, O2, O3) com a flag de arquitetura. Ainda, compilá-lo utilizando as flags encontradas no artigo "Finding Best Compiler Options for Critical Software Using Parallel Algorithms"

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

2 Metodologia

Para análise do experimento, utilizamos o gprof para obtermos os resultados de cada teste.

Os comandos utilizados para compilarmos com as flags (sem otimização, O1, O2, O3), utilizando a flag de arquitetura foram respectivamente:

```
gcc -o main main.c -pg -lm -march=native
gcc -o main01 main.c -pg -lm -O1 -march=native
gcc -o main02 main.c -pg -lm -O2 -march=native
gcc -o main03 main.c -pg -lm -O3 -march=native
```

As flags encontradas no artigo "Finding Best Compiler Options for Critical Software Using Parallel Algorithms" utilizadas para compilar o código foram:

- -fdce
- -fstrict-aliasing
- -ftree-pre
- -ftree-vrp
- -freorder-functions
- -fsignaling-nans
- -ffp-contract=on
- -fipa-profile
- -fsched-spec
- -funroll-all-loops

Para facilitar a execução, criamos um Makefile. Portanto, para executar o código, basta utilizar o comando:

```
make test ARGS=50000000
```

3 Resultados

Os resultados obtidos foram:

%Time	Cumulative seconds	self seconds	calls	self ms/calls	total ms/calls	name
99.11	8.91	8.91	50000000	0.00	0.00	advance
0.44	8.95	0.04				_init
0.33	8.98	0.03				main
0.11	8.99	0.01	2	5.00	5.00	energy
0.00	8.99	0.00	2	0.00	0.00	scale_bodies
0.00	8.99	0.00	1	0.00	0.00	offset_momentum

Tabela 1: Resultados sem otimização

%Time	Cumulative seconds	self seconds	calls	self ms/calls	total ms/calls	name
99.26	4.03	4.03	50000000	0.00	0.00	advance
0.49	4.05	0.02				main
0.25	4.06	0.01	2	5.00	5.00	energy
0.00	4.06	0.00	2	0.00	0.00	scale_bodies
0.00	4.06	0.00	1	0.00	0.00	offset_momentum

Tabela 2: Resultados com otimização O1

%Time	Cumulative seconds	self seconds	calls	self ns/calls	total ns/calls	name
100.00	2.86	2.86	50000000	57.20	57.20	advance
0.00	2.86	0.00	2	0.00	0.00	energy
0.00	2.86	0.00	2	0.00	0.00	scale_bodies
0.00	2.86	0.00	1	0.00	0.00	offset_momentum

Tabela 3: Resultados com otimização O2

%Time	Cumulative seconds	self seconds	calls	self Ts/calls	total Ts/calls	name
100.00	4.27	4.27				main
0.00	4.27	0.00	2	0.00	0.00	energy
0.00	4.27	0.00	1	0.00	0.00	offset_momentum

Tabela 4: Resultados com otimização O3

%Time	Cumulative seconds	self seconds	calls	self ms/calls	total ms/calls	name
99.67	9.01	9.01	50000000	0.00	0.00	advance
0.22	9.03	0.02				main
0.11	9.04	0.01	2	5.00	5.00	energy
0.00	9.04	0.00	2	0.00	0.00	scale_bodies
0.00	9.04	0.00	1	0.00	0.00	offset_momentum

Tabela 5: Resultados com as flags do artigo

Desse modo, pode-se concluir que o programa escolhido performou melhor utilizando a flag -O2, com tempo de execução de 2.86 segundos.

Ainda, a performance do programa compilado sem flags de otimização e com as flags do artigo são semelhantes (próximo aos 9 segundos); e a performance utilizando -O1 e -O3 são, também, semelhantes (próximo aos 4 segundos).