CS6241: Project 1 - Part 1

Zhen Li   zhenli.craig@gatech.edu   2024-03-23

Contents

1 Baseline

The llvm bytecode size are measured by python script `os.path.get_size()`.

The time are measured by a `time` based profiling tool `hyperfine`, which runs the program 2 times for warmup and then 10 times to take the average. (macOS seems do not have equivalent to `perf` on Linux.)

Bench		Bytecode size (byte)	User (ms)	System (ms)	Mean Total (ms)
is	original	22784	16.154	0.477	17.005
	baseline	24800	20.670	0.541	21.650
	ratio	**108.85%**	**127.95%**	**113.56%**	**127.32%**
bfs	original	14384	0.230	0.308	0.747
	baseline	15728	0.301	0.366	0.848
	ratio	**109.34%**	**131.05%**	**118.59%**	**113.40%**
dither	original	31984	39.616	2.145	42.690
	baseline	34992	71.908	2.448	75.778
	ratio	**109.40%**	**181.51%**	**114.15%**	**177.51%**
jacobi-1d	original	5840	1247.679	0.969	1252.289
	baseline	6624	9595.287	1.897	9627.112
	ratio	**113.42%**	**769.05%**	**195.79%**	**768.76%**
check_elimination	original	3728	0.210	0.337	0.749
	baseline	4160	0.270	0.348	0.797
	ratio	**111.59%**	**128.30%**	**103.27%**	**106.48%**
check_modification	original	3744	0.208	0.333	0.743
	baseline	4176	0.278	0.355	0.859
	ratio	**111.54%**	**134.12%**	**106.62%**	**115.63%**

2 Gupta's methods

2.1 Compile time check stats

2.1.1 check_elimination

Function	stage	Lower	Upper	Total
main	After Insertion	3	3	6
main	After Modification	4	4	8
main	After Elimination	1	2	3

main	After Loop Propagation	1	2	3

2.2 Runtime check stats

2.3 Static spill code generated in each commenting on causes of performance degradations