Denisse Blum
Koushik Chowdhury
Github: https://github.com/chykoushik/hpc-toolsA1
ssh: git@github.com:chykoushik/hpc-toolsA1.git

dgesv benchmark	**Fill out with: - Optimization + platform specific compilation flags applied - Execution times for my_dgesv (your implementation)**	
	icc (18.0.5)	**gcc (8.3.0)**

No opt.	Options: -O0
	Exec time (small): 2.67s
	Exec time (medium):
	Exec time (large):
		Options: -O0
		Exec time (small): 2.42s
		Exec time (medium): 19.79s
		Exec time (large): 158.06s

Opt level O1	Options: -O1
	Exec time (small): 1.49s
	Exec time (medium):
	Exec time (large):
		Options: -O1
		Exec time (small): 1.41s
		Exec time (medium): 11.55s
		Exec time (large): 92.91s

Opt level O2	Options: -O2
	Exec time (small): 0.66s
	Exec time (medium):
	Exec time (large):
		Options: -O2
		Exec time (small): 1.41s
		Exec time (medium): 11.75s
		Exec time (large): 95.84s

Opt level O3	Options:
	Exec time (small): 0.19s
	Exec time (medium):
	Exec time (large):
		Options: -O3
		Exec time (small): 1.42s
		Exec time (medium): 11.51s
		Exec time (large): 93.33s

Opt level Ofast	Options:
	Exec time (small): 0.18s
	Exec time (medium):
	Exec time (large):
		Options: -Ofast
		Exec time (small): 1.42s
		Exec time (medium): 11.71s
		Exec time (large): 97.97s

IPO	Options:
	Exec time (small): 0.18s
	Exec time (medium):
	Exec time (large):
		Options: -Ofast -fipa-pta -march=native
		Exec time (small): 1.42s
		Exec time (medium): 11.75s
		Exec time (large): 98.99s

PGO	Options:
	Exec time (small): 1.49s
	Exec time (medium):
	Exec time (large):
		Options: -Ofast -fipa-pta -fprofile-generate -march=native
		Exec time (small): 1.41s
		Exec time (medium): 11.71s
		Exec time (large): 95.25s

autovectorization	Options:
	Exec time (small): 0.17s
	Exec time (medium):
	Exec time (large):
		Options: -lm -ftree-loop-vectorize -O3 -fopt-info-vec
		Exec time (small): 1.41s
		Exec time (medium): 11.47s
		Exec time (large): 95.36s

We took the following matrix sizes for both gcc and icc

small 1024 × 1024
medium 2048 × 2048
large 4096 × 4096

If you check the execution time for gcc, we can observed that for “small”, except for “O0”, all execution times are range in 1.41s-1.42s but for “OO” it received 2.42s. Also same for the “medium” and “large” execution, except “O0”, all other benchmark provided almost close execution time to each other.

If we carefully check the “icc”, got “small”, execution time for “O3”, “Ofast”, “ipo” and “atovectorization” are close to each other where “O1” provides the highest execution time among all of them.

In most benchmarks for dgesv, icc compilation is faster than gcc, and in both cases, "O0" takes more execution time than others. According to the execution time for icc and gcc, "autovectorization" is the most efficient, and for gcc, along with “autovectorization”, “O1” provides the comparatively best result.

We were not able to run our code with icc (2048 and 4096), it gave a segmentation fault, but our implementation is fine with icc (1024).