HPC Tools

Assignment N°2

Haytam EL MERABETI
HPC Master student
elmerabeti@cy-tech.fr

Faculty of Computer Science

October 25, 2022

Task 2

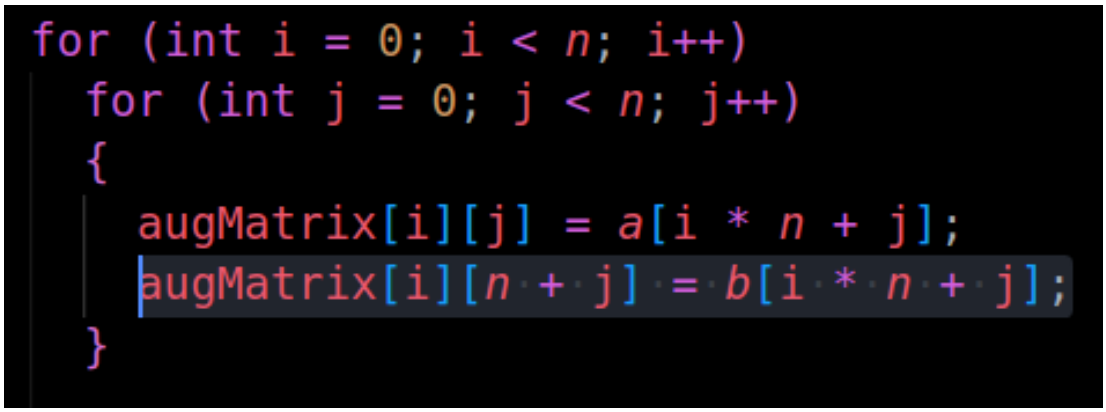
Compiler’s optimizations :

	icc	gcc
No Opt	Exec time (small) : 7.77s Exec time (medium) : 63.00s Exec time (large) : 502.63s	Exec time (small) : 8.04s Exec time (medium) : 65.30s Exec time (large) : 523.02s
Opt level O1	Exec time (small) : 1.22s Exec time (medium) : 13.52s Exec time (large) : 112.42s	Exec time (small) : 1.24s Exec time (medium) : 13.70s Exec time (large) : 111.80s
Opt level O2 (autovec with icc only)	Exec time (small) : 0.67s Exec time (medium) : 10.16s Exec time (large) : 87.81s	Exec time (small) : 1.22s Exec time (medium) : 13.52s Exec time (large) : 110.44s
Opt level O3 (autovec)	Exec time (small) : 0.67s Exec time (medium) : 10.17s Exec time (large) : 87.80s	Exec time (small) : 0.86s Exec time (medium) : 10.55s Exec time (large) : 90.00s
Opt level Ofast (autovec)	Exec time (small) : 0.67s Exec time (medium) : 10.13s Exec time (large) : 87.72s	Exec time (small) : 0.86s Exec time (medium) : 10.55s Exec time (large) : 90.01s

Other optimizations :

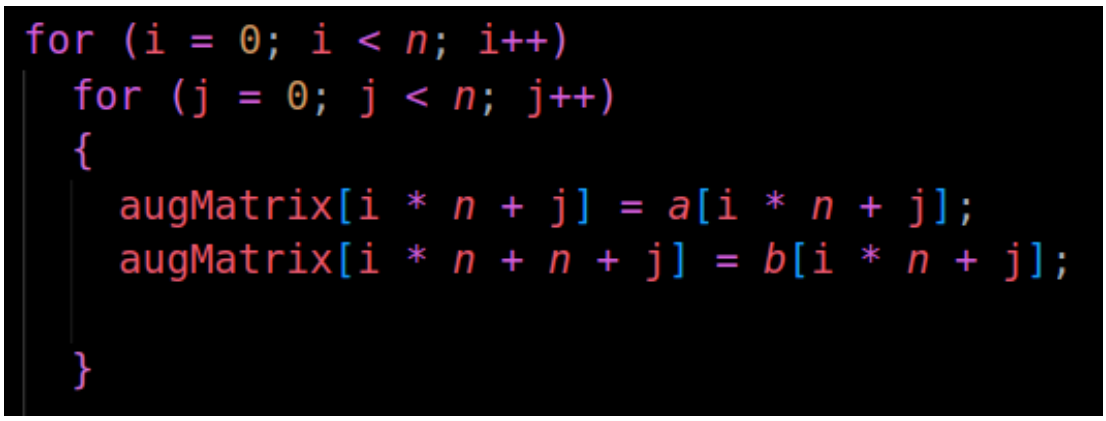
The following are other optimizations methods applied on the code or the compiler’s flags (Testing the impact of the following modifications in the code with -O0 flag) :

- Loop optimizations :



	icc	gcc
No Opt + Loop fusion 1	Exec time (small) : 7.75s Exec time (medium) : 62.97s Exec time (large) : 502.10s	Exec time (small) : 8.00s Exec time (medium) : 65.08s Exec time (large) : 520.24s

- Improving augmatrix instantiation to exploit cache locality :

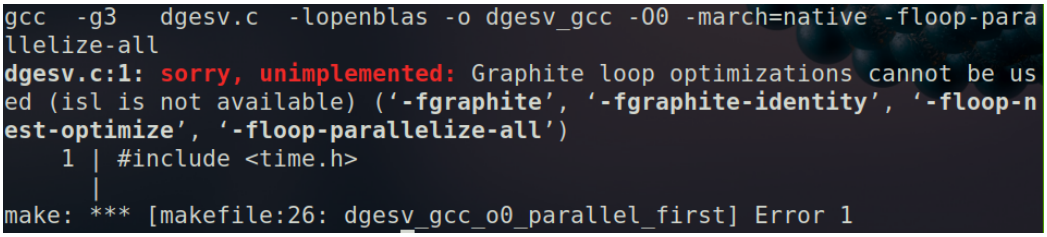


	icc	gcc
No Opt + new augmatrix	Exec time (small) : 7.28s Exec time (medium) : 58.70s Exec time (large) : 472.21s	Exec time (small) : 7.98s Exec time (medium) : 64.33s Exec time (large) : 516.74s

- PGO :

	icc	gcc
No Opt + PGO :	Exec time (small) : 6.03s Exec time (medium) : 48.55s Exec time (large) : 323.25s	Exec time (small) : 6.55s Exec time (medium) : 53.00s Exec time (large) : 427.81s

- Autoparallelization :
  - With gcc, I got the following error that I couldn't fix ; I guess it has something to do with gcc's configuration itself :



I found this post in this forum but I didn't try the solution provided (mainly because I didn't understand what was the problem for the original poster): https://www.myroms.org/forum/viewtopic.php?t=5506)

	icc	gcc
No Opt + Autoparallelization :	Exec time (small) : 8.69s Exec time (medium) : 60.18s Exec time (large) : 474.86s	Exec time (small) : ??s Exec time (medium) : ??s Exec time (large) : ??s

- All improvements + Add restrict to augmatrix :

	icc	gcc
O3 + All improvements (for PGO the first executable executed one time on each size):	Exec time (small) : 1.78s Exec time (medium) : 6.21s Exec time (large) : 56.68s	Exec time (small) : 0.65s Exec time (medium) : 6.65s Exec time (large) : 68.64s
Ofast + All improvements (for PGO the first executable executed one time on each size):	Exec time (small) : 1.81s Exec time (medium) : 6.22s Exec time (large) : 57.09s	Exec time (small) : 0.65s Exec time (medium) : 6.68s Exec time (large) : 69.08s

Conclusion :

Its clear that the aggressive optimizations applied with the -Ofast flag impact negatively on the performance of our code, and the autoparallelization wasn’t successful. But the autovectorization included in the O3 flag had a huge positive impact, alongside the Profile-Guided Optimizations.

⇒ The optimal configation that we reach so far is the -O3 flag with PGO and all previous code modifications (I will focus more on cache misses and memory issues on the final task):

```
dgesv_gcc_o3_pfprof_gen: dgesv.c
  gcc  -g3   dgesv.c  -lopenblas -o dgesv_gcc -O3 -march=native -fprofile-generate

dgesv_gcc_o3_pfprof_use: dgesv.c
  gcc  -g3   dgesv.c  -lopenblas -o dgesv_gcc -O3 -march=native -fprofile-use

dgesv_icc_o3_prof_gen: dgesv.c
  icc  -g3   dgesv.c  -mkl -o dgesv_icc -O3 -xHost -prof-gen

dgesv_icc_o3_prof_use: dgesv.c
  icc  -g3   dgesv.c  -mkl -o dgesv_icc -O3 -xHost -prof-use
```