Course 02614: Assignment 1 — Evaluation January 8, 2023

Group #3: Gustav Gamst Larsen s180820
Lukas Leindals s183920
Simon Majgaard s184012

Comments:

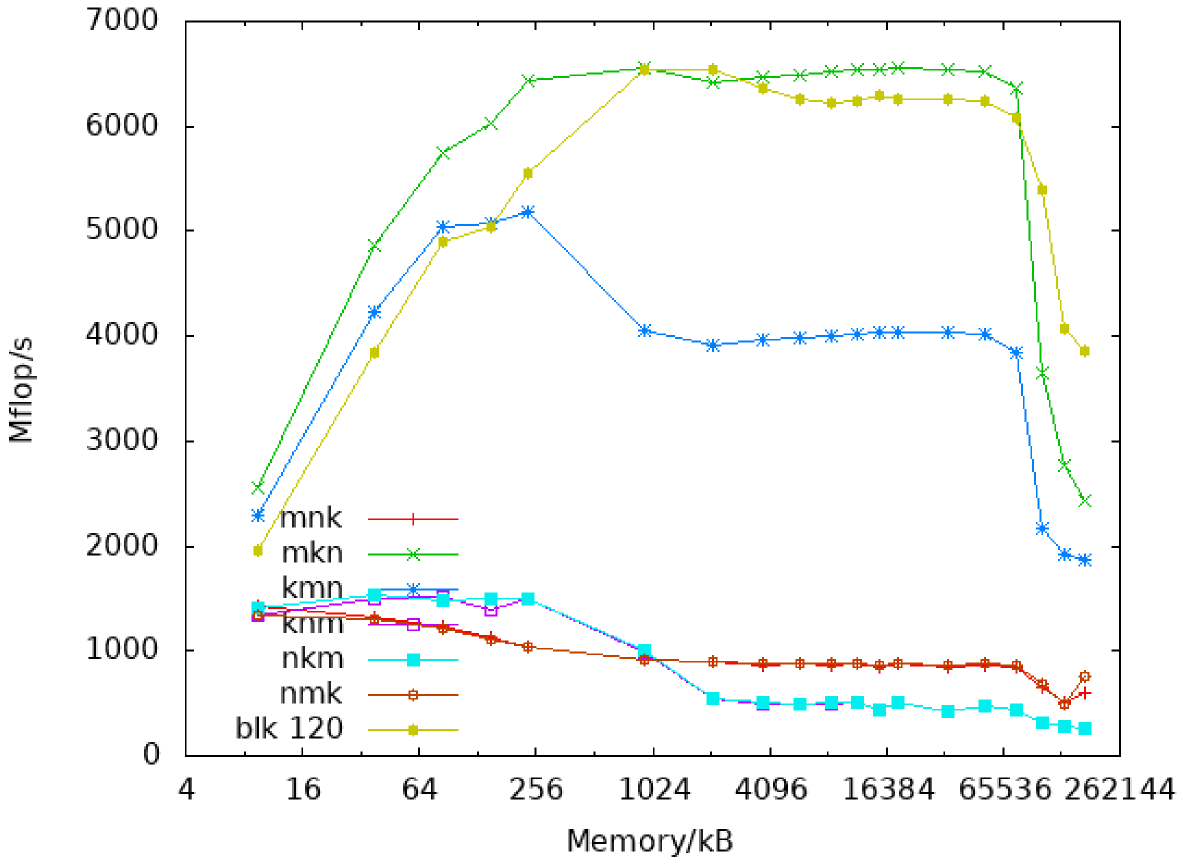
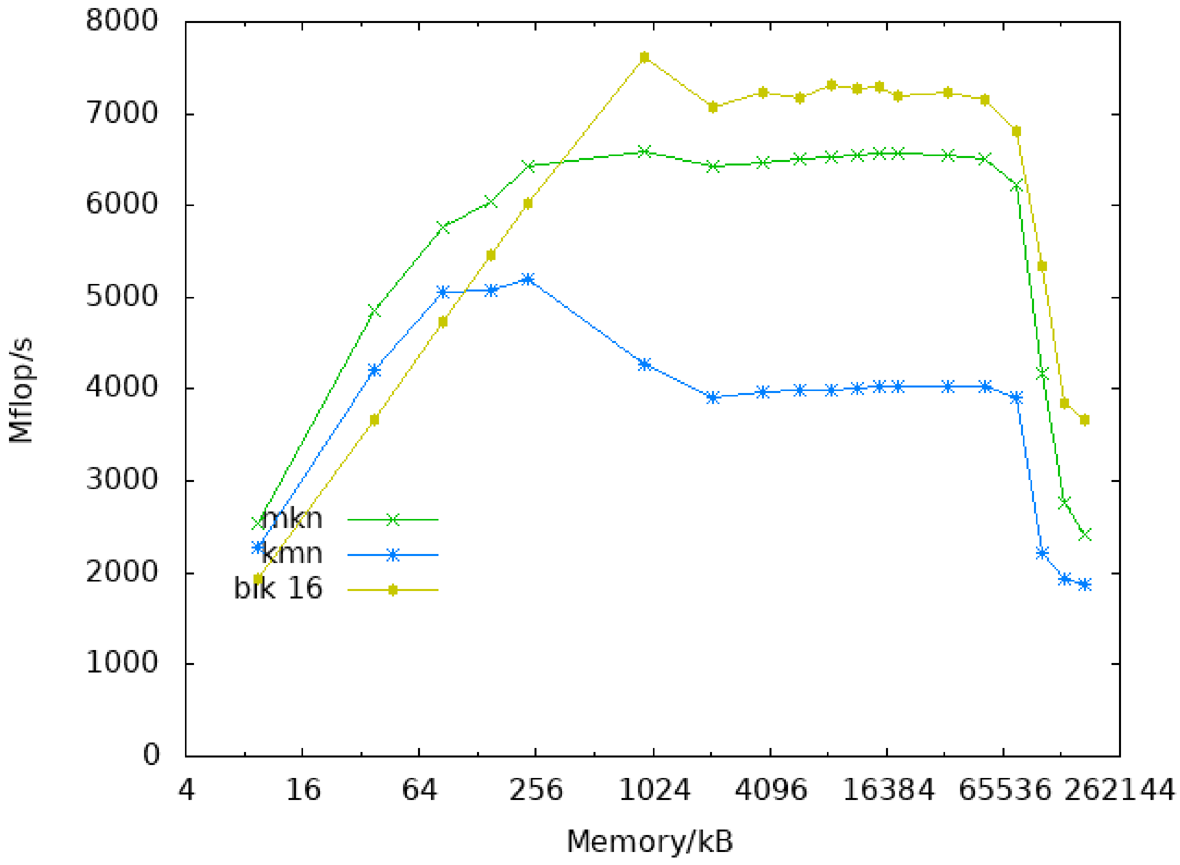
General evaluation:
Please use lines and symbols for your plots! Which compiler and version did you use? I found GCC 6.3 from your uploaded library, but it's not mentioned in the report.

- quite extensive analysis of the six permutations, when it comes to the compiler options
- you could have saved quite some time, by thinking about the usefulness of your option choices, e.g. ”Why don't we need LTO here?”, or ”Is there anything in the code, where the ”fast math” options could make a difference?”.
- you can see that, too, in your histograms! BTW, a plot like the upper one on the next page, has all the information needed.
- when you used the Performance Analyzer, did you fix the number of iterations, as mentioned, and described in the README file? By looking at Table 3, I am afraid you did not - which is indicated by the CPU time used to be always about 3 seconds (for the smaller problems). 3 seconds is the min. CPU time set by the driver!
- The above means, that in principle only the lines for 1200x1200 in your table are useful data, which you need to interpret! From those line you can see, that like the Mflop/s values for the different permutations, also the cache hits/misses fall into 3 groups (depending on the last index)!

Loop blocking:

- you blocked the fastest permutation, but why did you block two loops, only?
- when finding the optimal bs, why did you use powers of 2? You might jump over ”interesting” values, that way!
- your implementation gives a performance boost for bs=16, and problems between the L2 and L3 sizes, but it drops performance when going beyond the L3 cache (like the other versions) - see lower figure next page.
- a better choice would be blocking in all 3 loops - as the inner loop gets larger, data can no longer stay in cache!

Poor	Not quite adequate	Adequate	Good	Very good
😠	🙁	😐	🙂	😄
			✓	


7000
6000
5000
4000
3000
2000
1000
0
Mflop/s
4
16
64
256
1024
4096
16384
65536
262144
Memory/kB
mnk
mkn
kmn
knm
nkm
nmk
blk 120

8000
7000
6000
5000
4000
3000
2000
1000
0
Mflop/s
4
16
64
256
1024
4096
16384
65536
262144
Memory/kB
mkn
kmn
blk 16