1 terops / sec

c += b\*d = 2 ops and 4 mem access and 2 bytes = 2 ops / 8 bytes = operational intensity

same regardless of problem size and mapping.

OSP = 5/(6\*num bytes) = 5 ops and (N^3)/6 -> ((5\*N^3)/6)/time

Vectorization multiplies the numerator

(should expect a factor of 8 right away from vectorization) (1 instruction in assemble for 8 instructions)

Why was it vectorized and why a factor of 20 not 8?

Fused multiply add unit makes a mult and an adds 1 op. – only get 1 or 2 for your code. Must mult first then add, if not, FMA will not be used.

Study vectorization and accession patterns and why j innermost is fastest – the elements in memory have to be contiguous – it happens when j is the innermost.

Issue, execution, write back – can do 8x of this at a time in vectorization

Intel advisor toolkit – intel’s provided roofline toolkit – learn how to use this

Plot flops/sec instead of execution time.

Report only on OSP problem. Introduce the problem, what we solved and the approach, all results, etc. Write the report in terms of how you would explain to someone. You start with poor performance then how to improve it. Talk about legality and permutation options. Explain why the fastest thoroughly.