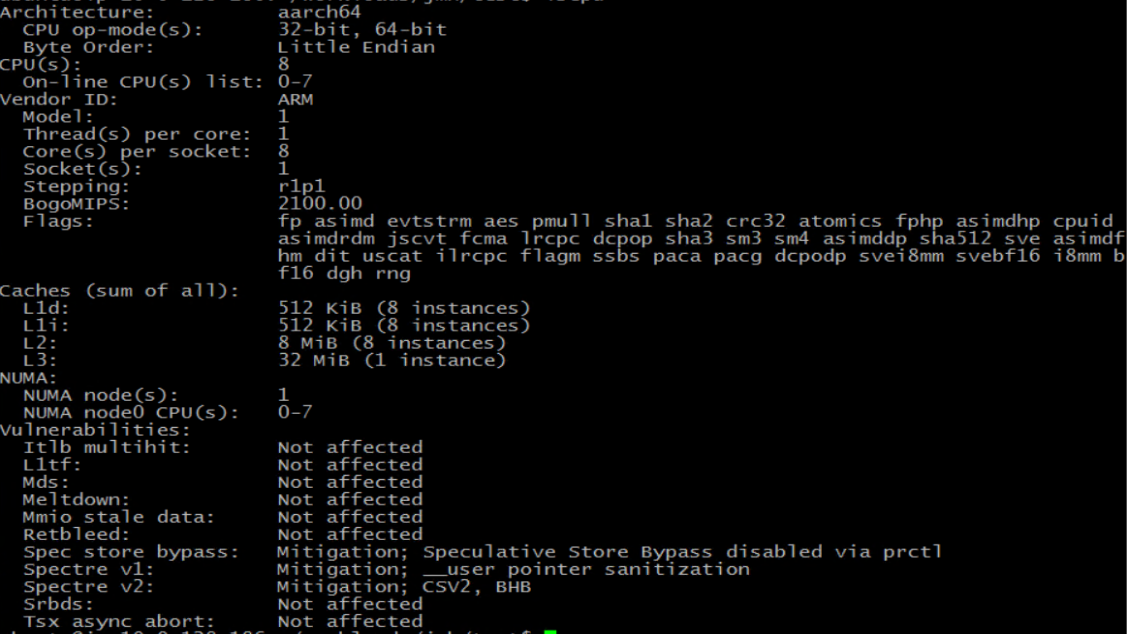
**GVT3 VS ICELAKE VECTOR API PERF**

Since Vector API benchmarking kernels are single threaded hence what matters most is CORE and RAM.

Thus, we are selecting instances with similar RAM and storage configurations.

GVT3: (C7G.2xlarge: 8 VCPU, 16 GB RAM, 32GB SSD]



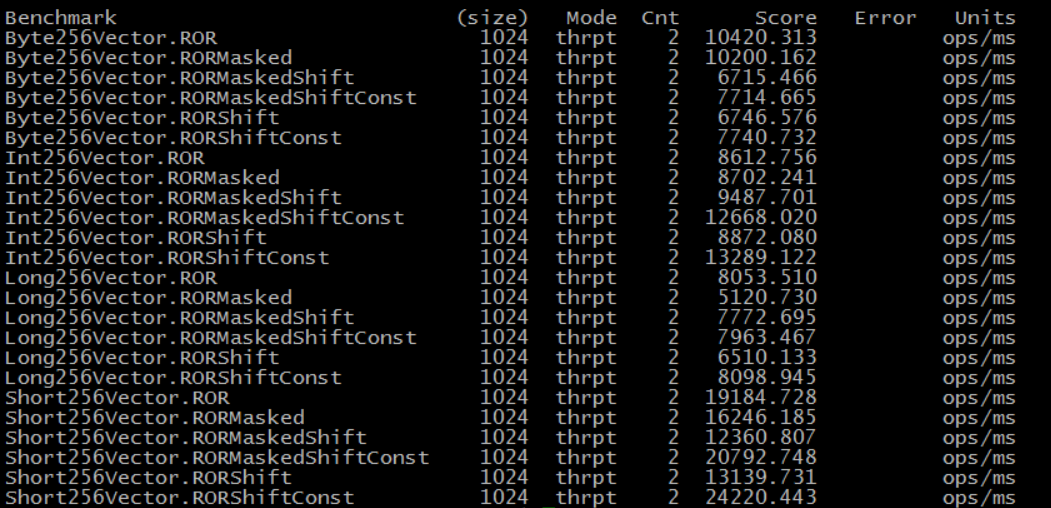
ICELAKE: (M7i.xlarge: 4 VCPU, 16 GB RAM, 32 GB SSD)



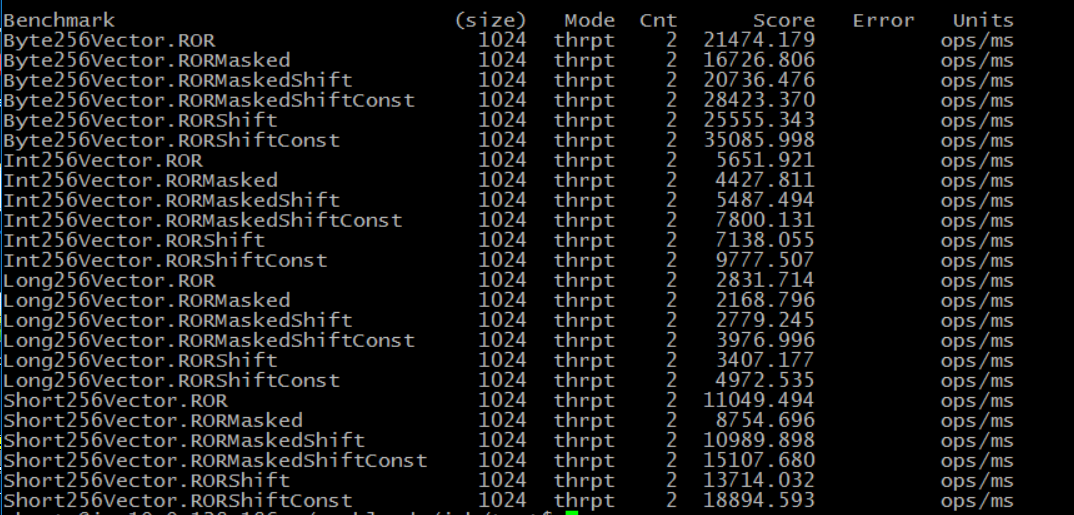
ROTATE PERFORMANCE:

Vector Size 256 bits:

ICX:



GVT3:



Observations:

* Rotate is an integral operation, we do not perform rotation over floating point vectors.
* Apart from Byte level rotation, X86 scores above GVT3 for all other types.
* Unlike X86, ARM does not have a direct rotate instruction for integer and long types.
* ARM always degenerates rotates into Shift left, right and Or operations. X86 does so for sub-word types.
* ARM has single instructions of shift left/right for all the types. ARM vector instruction encoding explicitly encodes lane size, thus same operations can be performed over vectors of varying length and lane sizes.
* X86 does not have a direct shift instruction for byte vector. Thus, we promote each byte lane to word and then perform shift left/right operations. Unpacking-packing b/w byte and words adds to the path length and hence reduces overall throughput.

