Hanif Khan

Project 5
CS575

Oregon State University

1. What machine you ran this on

   I ran this on Flip server with info:

   Architecture: x86_64

   CPU op-mode(s): 32-bit, 64-bit

   Byte Order: Little Endian

   CPU(s): 24

   On-line CPU(s) list: 0-23

   Thread(s) per core: 2

   Core(s) per socket: 6

   Socket(s): 2

   NUMA node(s): 2

   Vendor ID: GenuineIntel

   CPU family: 6

   Model: 44

   Model name: Intel(R) Xeon(R) CPU X5650 @ 2.67GHz

   Stepping: 2

   CPU MHz: 2660.016

   BogoMIPS: 5320.03

Virtualization: VT-x

L1d cache: 32K

L1i cache: 32K

L2 cache: 256K

L3 cache: 12288K

NUMA node0 CPU(s): 0,2,4,6,8,10,12,14,16,18,20,22

NUMA node1 CPU(s): 1,3,5,7,9,11,13,15,17,19,21,23

Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 popcnt aes lahf_lm tpr_shadow vnmi flexpriority ept vpid dtherm ida arat

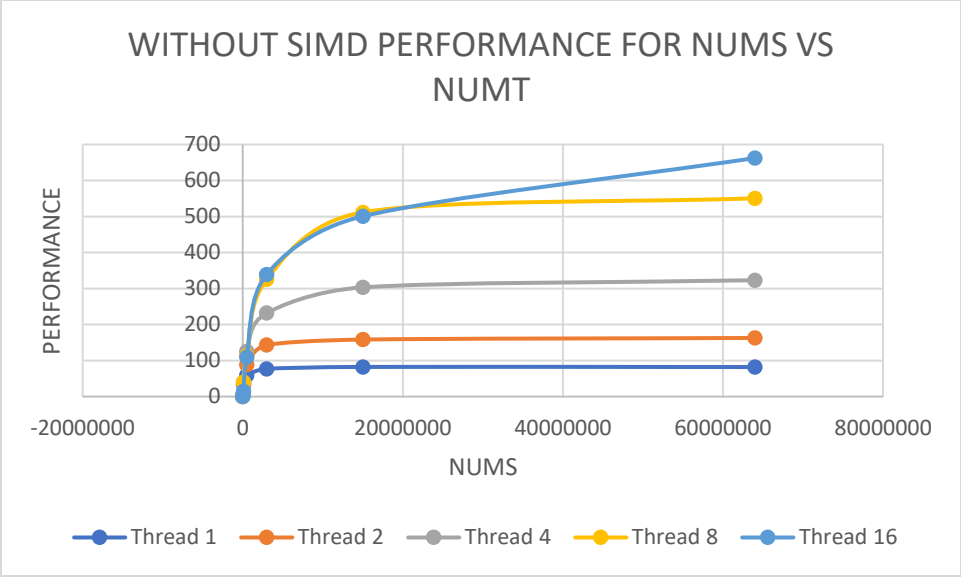
2. Show the tables and graphs

Tables:

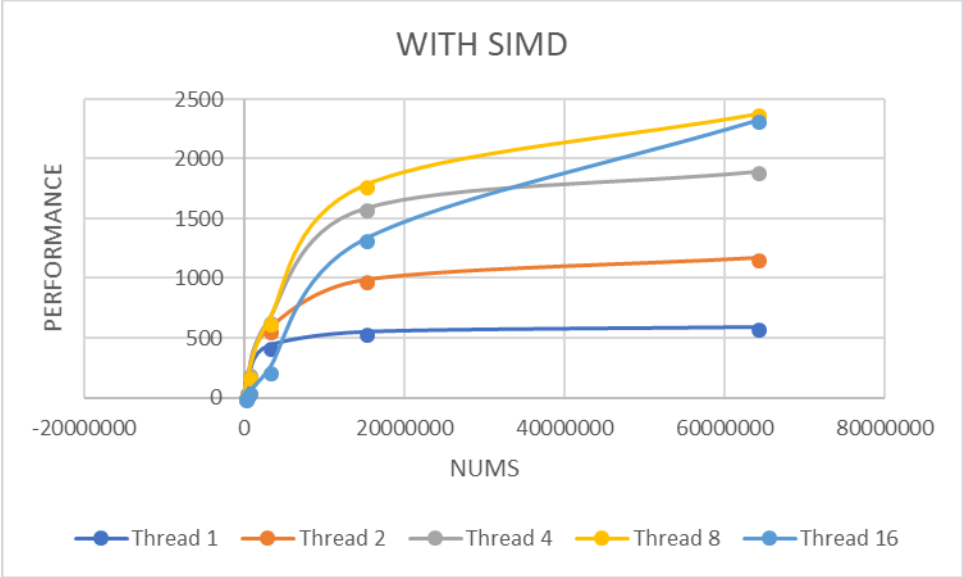
| | | 1000 | 10000 | 100000 | 500000 | 3000000 | 15000000 | 64000000 |
|---|---|---|---|---|---|---|---|---|
| | Thread 1 | 0.56 | 5.17 | 32.45 | 58.85 | 77.03 | 82.38 | 81.97 |
| | Thread 2 | 0.54 | 4.52 | 36.67 | 88.25 | 143.04 | 158.32 | 162.73 |
| | Thread 4 | 0.46 | 4.46 | 39.87 | 125.16 | 231.84 | 303.23 | 323.12 |
| | Thread 8 | 0.38 | 4.09 | 39.08 | 112.76 | 325.1 | 511.61 | 550.69 |
| Without SIMD | Thread 16 | 0.25 | 1.49 | 14.48 | 109.76 | 339.24 | 500.63 | 662.33 |
| | Thread 1 | 0.6 | 5.85 | 45.91 | 156.49 | 429.86 | 547.53 | 587.11 |
| | Thread 2 | 0.5 | 5.12 | 42.34 | 173.13 | 568.62 | 982.9 | 1168.31 |
| | Thread 4 | 0.47 | 4.43 | 39.86 | 197.08 | 644.19 | 1584.95 | 1895.8 |
| | Thread 8 | 0.43 | 3.79 | 27.72 | 181.29 | 633.84 | 1778.91 | 2377.68 |
| WITH SIMD | Thread 16 | 0.11 | 1.63 | 13.66 | 47.18 | 225.54 | 1327.09 | 2324.14 |

Graphs:

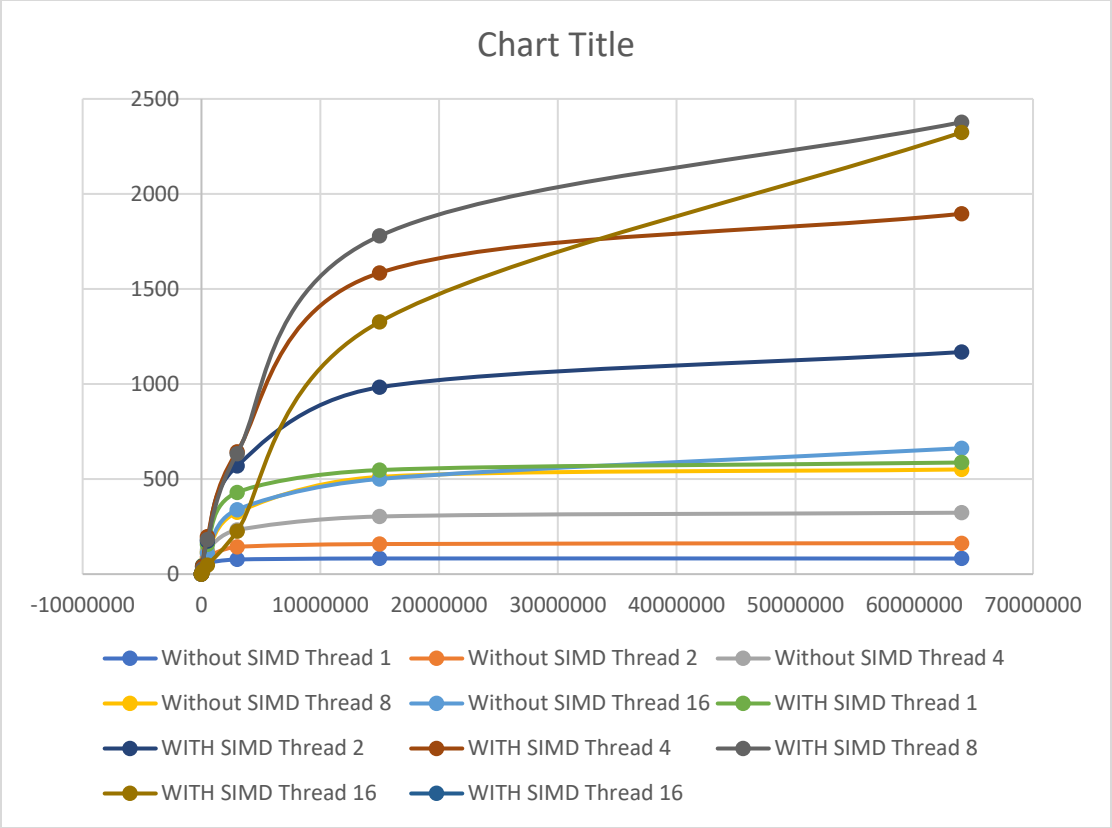
For Without SIMD:




With SIMD:




SIMD vs NON SIMD




3. What patterns are you seeing in the performance curves? (Pay special attention to what the curves are doing at the far end of NUMS.)
   I can Observe that the performance of the SIMD is much more better than the NON SIMD. Also, it can be seen that the curves after particular number it doesn't increases the performance.

4. Why do you think the patterns look this way?

   In the SIMD computing performs data level parallelism:

   - ➔ Same operation performed on many data elemenets
   - ➔ Improve performance, No intra-vector dependencies

5. What does that mean for the proper use of vectorized parallel computing?

   As definition SIMD processing exploits data-level parallelism. Data-level parallelism means that the operations required to transform a set of vector elements can be performed

on all elements of the vector at the same time. That is, a single instruction can be applied to multiple data elements in parallel.