1. Analysis

How is the memory subsystem structured on your machine. On Linux system likwid-topology (developed by Google) is a good tool to know how the memory system is architectured.
Question: How much DRAM memory? Memory speed? Memory Technology? (Not all is exposed unless you are root.)
-> DRAM memory : 65314.3 (Domain 0) + 65536 MB (Domain 1)
-> Memory Speed : DDR4 speed 1600/1866/2133
(https://ark.intel.com/products/83361/Intel-Xeon-Processor-E5-2667-v3-20M-Cache-3_20-GHz)
-> Memory Technology : DDR4

Question: How much bandwidth can your processor draw from the bus? You will need to discover width of the memory bus and clock speed of the memory bus. Full duplex/half duplex? (On intel processor, ARK often indicates something.)
-> Bandwidth processor can draw from bus : 9.6 GT/s (GigaTransfer per second) QPI
(https://ark.intel.com/products/83361/Intel-Xeon-Processor-E5-2667-v3-20M-Cache-3_20-GHz)
-> Memory Bandwidth mentioned on arch.intel.com site : 68 GB/s.
-> Xeon Processor E5 2667 v3 is full duplex.

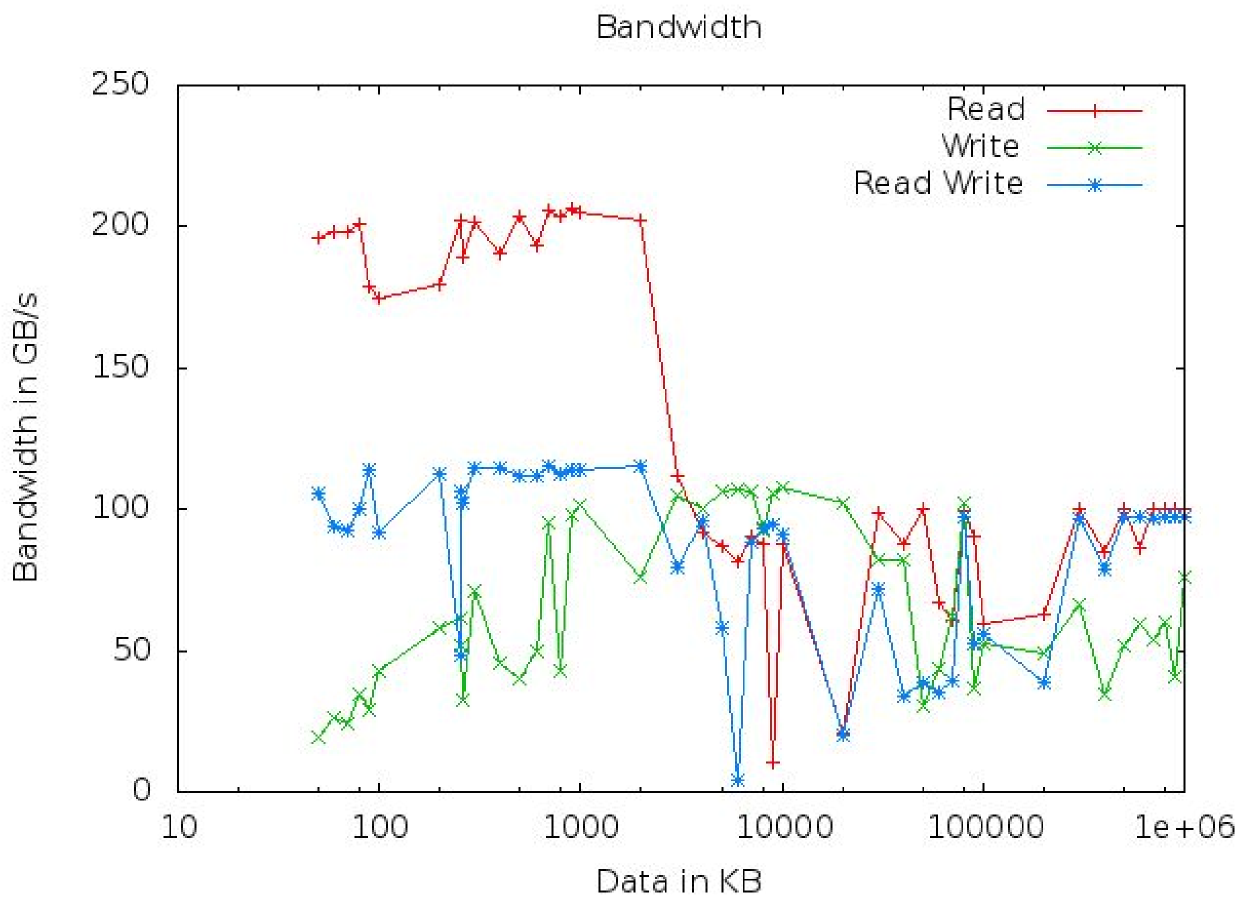
Question: What is the highest level of data cache? How big is it? How is it shared across core? Same question for all levels of data cache.
-> Highest level of cache is L3 cache. It's size is 20MB and even processors share one L3 cache and odd number processors share another L3 cache.
-> L2 cache is of size 256KB and following groups share the cache.

- (0 2)
- (4 6)
- (8 10)
- (12 14)
- (1 3)
- (5 7)
- (9 11)
- (13 15)

-> L1 cache is of size 32KB and is share in same group as L2 cache is shared.

2. Bandwidth

Question: Write a code to measure read bandwidth
-> Attached
Question: Write a code to measure write bandwidth
-> Attached
Question: Write a code to measure read/write bandwidth
-> Attached
Question: Measure read, write, read/write bandwidth at different size of data. (From 1KB to 200MB)



Data points in Bandwidth.txt file

3. Latency

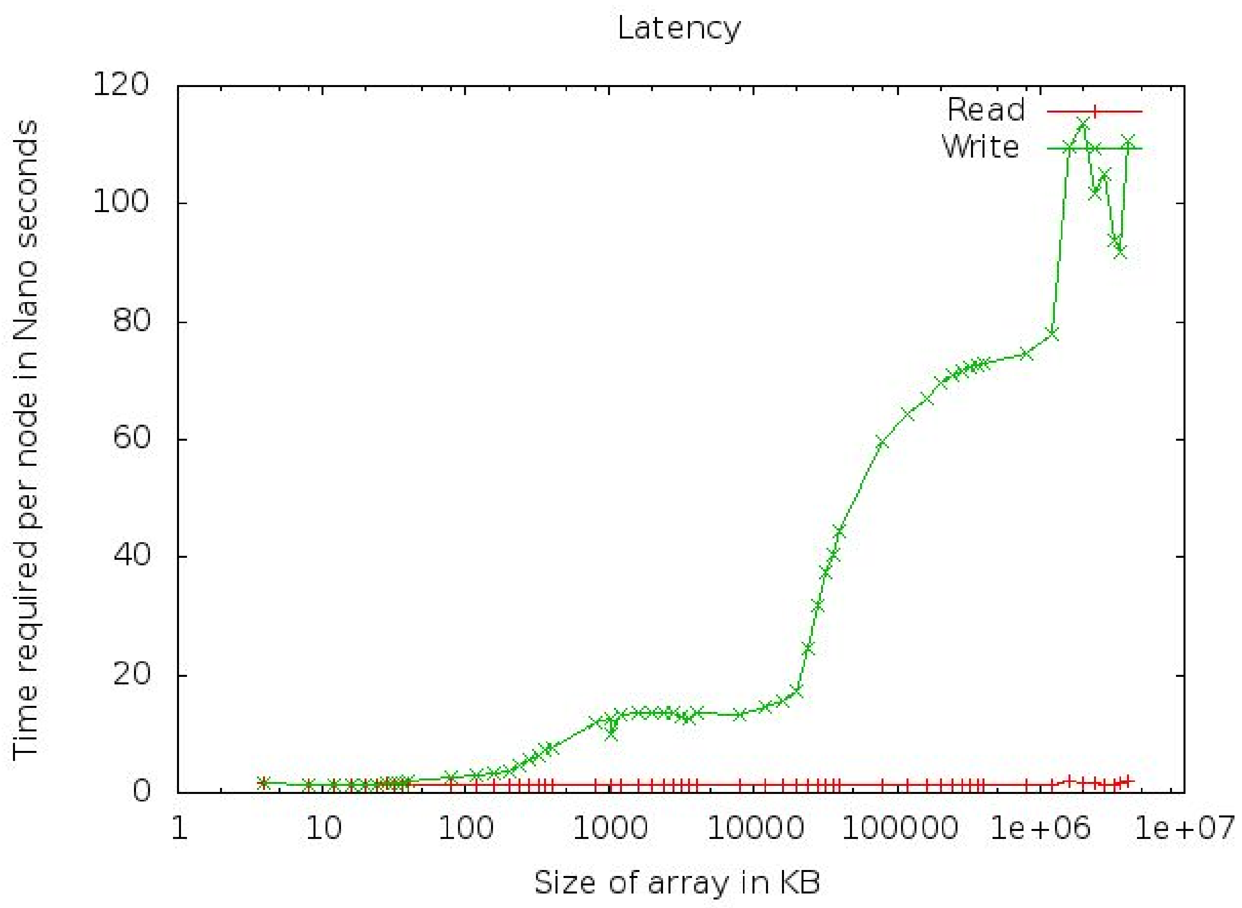
Question: Write a code that traverses some element of a linked list as described above.
-> Attached

Question: Write function to create a linked list which structure is aggreable to the core structure.
-> Wrote code such that, the i th element of list holds the address of i+1 element

Question: What is the latency in that case?
-> 1 nano second, as hardware prefetching occurs

Question: Write a function to create a linked list that will jump around the memory subsystem while avoiding to build short cycles.
-> Scrambled the array index, using another array.

Question: For different size (From 1kB to 200MB), what is the latency of the system?
->



Red line is Sequential
Green Line is Scattered
Data points in latency.txt