**Intel Memory Latency Checker**

(Intel MLC-3.9a)

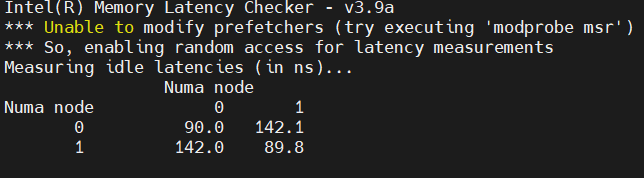
**Introduction:**

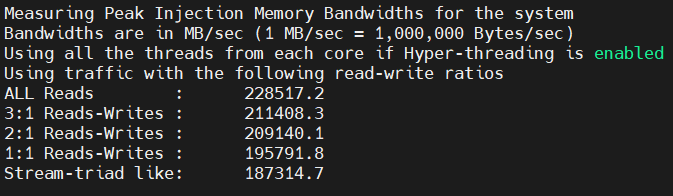
For any workload, memory configuration coupled with the behavior of the memory module on the system can be a key indicator for achieving peak performance. The more involved the memory module setup, the more arduous it will prove to be to manually investigate and apply optimal configurations. It is no shrouded forethought as to the relationship between memory bandwidth and overall peak performance of any workload. Additionally, when NUMA nodes come into the picture, there is not just an increase in complexity but also in the uncertainty of the efficiency/behavior of the memory module. Since information could be distributed across multiple, universally accessible (by every CPU) nodes, bandwidth and latency highly depend on the distance from the cell containing the CPU/IO bus (making the memory access request) to the cell containing the target memory.

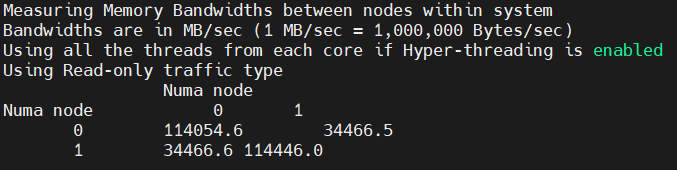
Hence, seeing as to how pertinent yet manually taxing it is to identify memory bottlenecks and progressively introduce optimizations, Intel MLC (Memory Latency Checker) provides you with precisely this and more. By analyzing memory latencies and bandwidths for varying loads to consolidate insightful interpretations on even systems with the most intricately configured memory modules (including caches and specific sets of cores), Intel MLC can not only be used to accelerate the process of optimizing your workload but also allow you to conclude how deviant your system performance is with respect to how it should be performing( for example, is your NUMA I/O, throughput and latency as it should be?) ; thereby aiding you in setting your system configuration in the most optimal way.

**Tool Execution:**

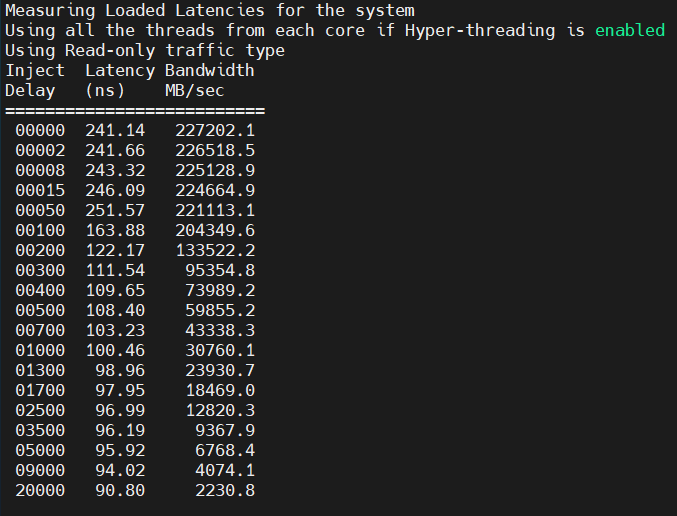
1. When the tool is launched without command-line arguments, it automatically identifies and measures the following topology information.
2. Idle memory latencies between local and cross NUMA nodes.

****

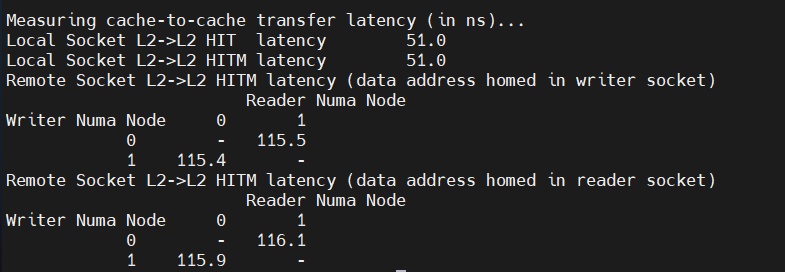
1. Peak memory bandwidth for requests varying with various ratios of read to writes. 
2. Bandwidth values for requests between local and cross NUMA nodes.(MB/sec).



1. Latencies at different Bandwidth points for the system with read-only traffic.

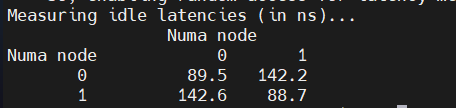


1. Cache-to-cache HIT and HIT MISS latencies(ns).



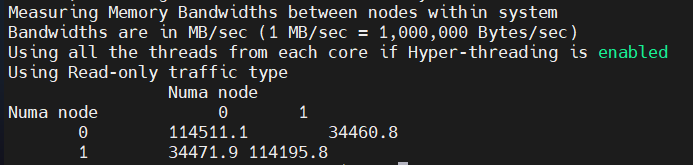
1. With command line arguments.
2. **mlc --latency\_matrix**

Local and cross-NUMA node memory latency.



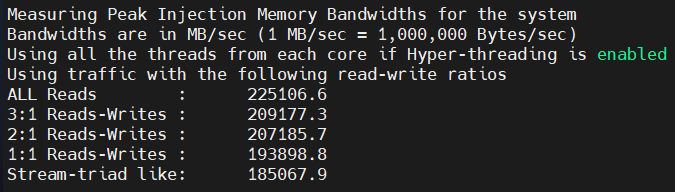
1. **mlc --bandwidth\_matrix**

Local and cross NUMA node memory b/w.



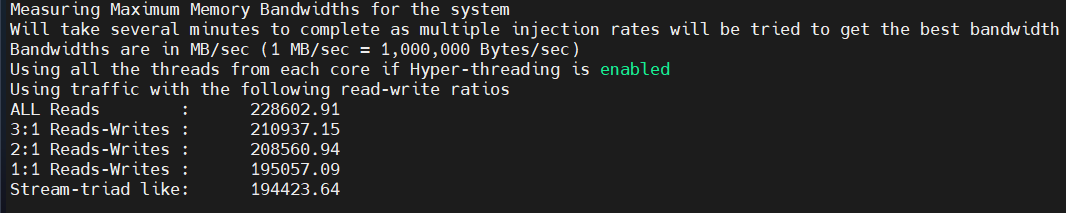
1. **mlc --peak\_injection\_bandwidth**

Peak memory b/w for various read-write ratios with all local accesses.



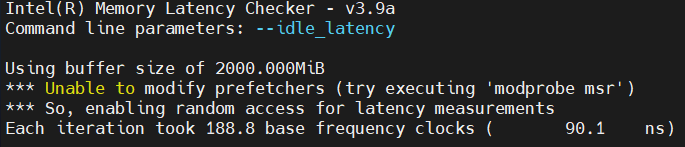
1. **mlc --max\_bandwidth**

Maximum memory b/w for various read-write ratios and load injection rates

****

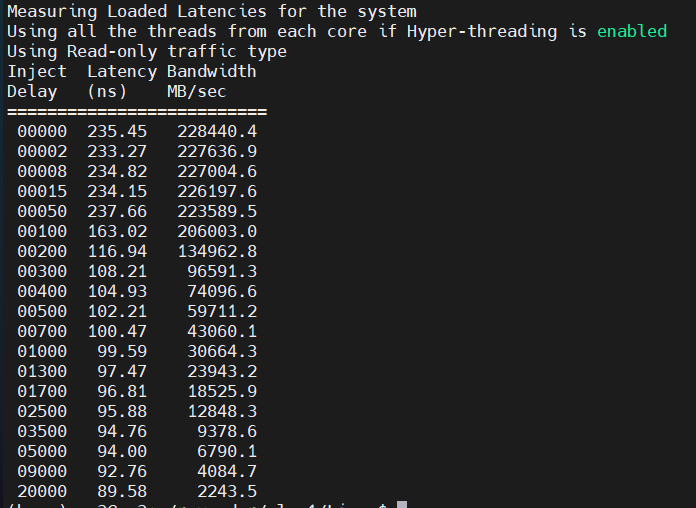
1. **mlc --idle\_latency**

Idle memory latency of the platform.



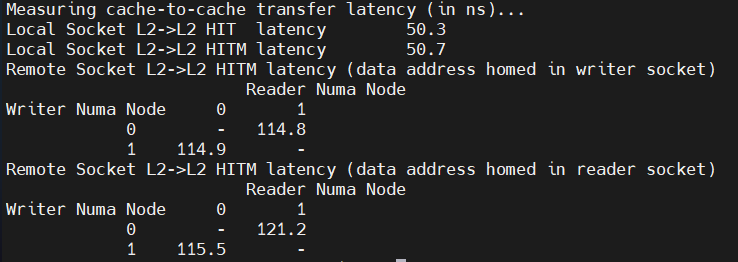
1. **mlc --loaded\_latency**

Loaded memory for different bandwidths (MB/sec).



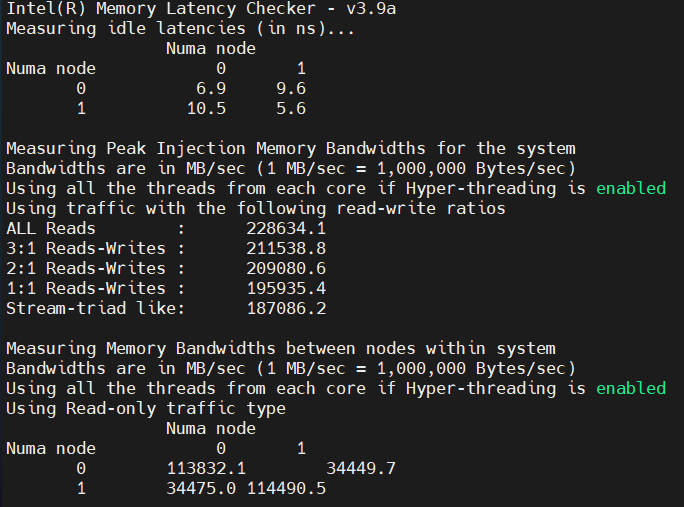
1. **mlc --c2c\_latency**

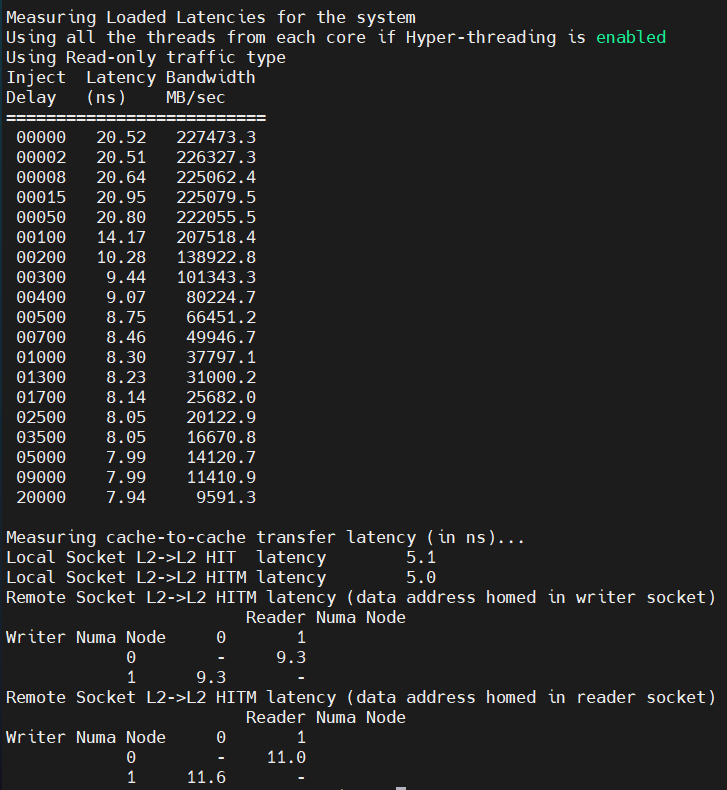
Itprints the cache-to-cache HIT and HIT MISS latencies of the platform

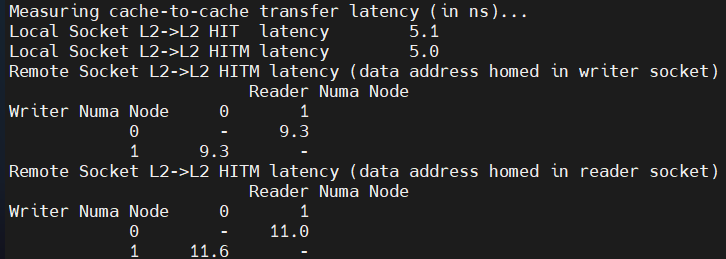


1. **mlc -e**

Similar to ./mlc command function but does not modify prefetcher settings.

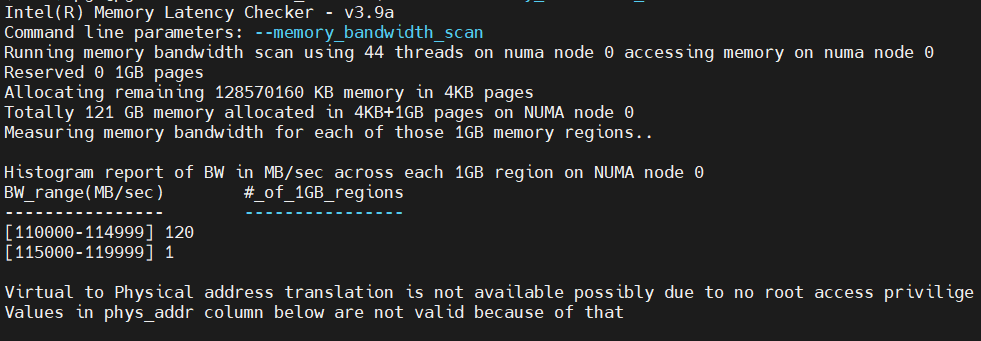


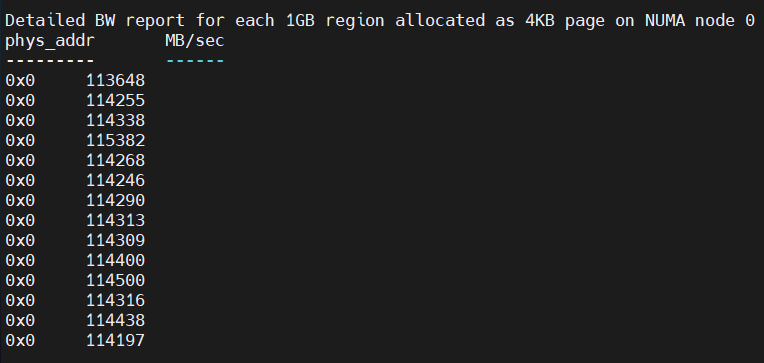




1. **mlc --memory\_bandwidth\_scan**

Memory bandwidth and bandwidth histogram report(NUMA node 0) across entire memory for each 1 GB address range

****



…