Intel® oneAPI VTune™ Profiler 2021.1.1 Gold

Elapsed Time: 0.047s

> Application execution time is too short. Metrics data may be unreliable. Consider reducing the sampling interval or increasing your application execution time.

Clockticks:	44,100,000
Instructions Retired:	73,440,000
CPI Rate:	0.600
MUX Reliability:	0.857
Retiring:	39.8% of Pipeline Slots
Light Operations:	41.6% of Pipeline Slots
FP Arithmetic:	0.0% of uOps
FP x87:	0.0% of uOps
FP Scalar:	0.0% of uOps
FP Vector:	0.0% of uOps
Other:	100.0% of uOps
Heavy Operations:	0.0% of Pipeline Slots
Microcode Sequencer:	2.7% of Pipeline Slots
Assists:	0.0% of Pipeline Slots
Front-End Bound:	21.4% of Pipeline Slots

> Issue: A significant portion of Pipeline Slots is remaining empty due to issues in the Front-End.
>
> Tips: Make sure the code working size is not too large, the code layout does not require too many memory accesses per cycle to get enough instructions for filling four pipeline slots, or check for microcode assists.

Front-End Latency: 12.2% of Pipeline Slots
ICache Misses: 0.0% of Clockticks
ITLB Overhead: 0.0% of Clockticks
Branch Resteers: 0.0% of Clockticks
Mispredicts Resteers: 0.0% of Clockticks
Clears Resteers: 0.0% of Clockticks
Unknown Branches: 0.0% of Clockticks
DSB Switches: 0.0% of Clockticks
Length Changing Prefixes: 0.0% of Clockticks
MS Switches: 0.0% of Clockticks
Front-End Bandwidth: 9.2% of Pipeline Slots
Front-End Bandwidth MITE: 24.5% of Clockticks
Front-End Bandwidth DSB: 0.0% of Clockticks
(Info) DSB Coverage: 26.3%
Bad Speculation: 6.1% of Pipeline Slots
Branch Mispredict: 0.0% of Pipeline Slots
Machine Clears: 6.1% of Pipeline Slots
Back-End Bound: 32.7% of Pipeline Slots

> A significant portion of pipeline slots are remaining empty. When operations take too long in the back-end, they introduce bubbles in the pipeline that ultimately cause fewer pipeline slots containing useful work to be retired per cycle than the machine is capable to support. This opportunity cost results in slower execution. Long-latency operations like divides and memory operations can cause this, as can too many operations being directed to a single execution port (for example, more multiply operations arriving in the back-end per cycle than the execution unit can support).

Memory Bound: 23.4% of Pipeline Slots

> The metric value is high. This can indicate that the significant fraction of execution pipeline slots could be stalled due to demand memory load and stores. Use Memory Access analysis to have the metric breakdown by memory hierarchy, memory bandwidth information, correlation by memory objects.

L1 Bound: 12.2% of Clockticks

> This metric shows how often machine was stalled without missing the L1 data cache. The L1 cache typically has the shortest latency. However, in certain cases like loads blocked on older stores, a load might suffer a high latency even though it is being satisfied by the L1. Note that this metric

value may be highlighted due to DTLB Overhead or Cycles of 1 Port Utilized issues.

DTLB Overhead: 0.6% of Clockticks
Load STLB Hit: 0.0% of Clockticks
Load STLB Miss: 0.6% of Clockticks
Loads Blocked by Store Forwarding: 0.0% of Clockticks
Lock Latency: 0.0% of Clockticks

A significant fraction of CPU cycles spent handling cache misses due to lock operations. Due to the microarchitecture handling of locks, they are classified as L1 Bound regardless of what memory source satisfied them. Note that this metric value may be highlighted due to Store Latency issue.

Split Loads: 0.0% of Clockticks
4K Aliasing: 1.2% of Clockticks
FB Full: 0.0% of Clockticks

This metric does a rough estimation of how often L1D Fill Buffer unavailability limited additional L1D miss memory access requests to proceed. The higher the metric value, the deeper the memory hierarchy level the misses are satisfied from. Often it hints on approaching bandwidth limits (to L2 cache, L3 cache or external memory). Avoid adding software prefetches if indeed memory BW limited.

- **L2 Bound:** 0.0% of Clockticks
- **L3 Bound:** 0.0% of Clockticks
  - **Contested Accesses:** 0.0% of Clockticks
  - **Data Sharing:** 0.0% of Clockticks
  - **L3 Latency:** 0.0% of Clockticks
  - **SQ Full:** 0.0% of Clockticks
- **DRAM Bound:** 0.0% of Clockticks
  - **Memory Bandwidth:** 0.0% of Clockticks
  - **Memory Latency:** 12.2% of Clockticks
- **Store Bound:** 0.0% of Clockticks
  - **Store Latency:** 12.2% of Clockticks
  - **False Sharing:** 0.0% of Clockticks
  - **Split Stores:** 0.0% of Clockticks
  - **DTLB Store Overhead:** 0.6% of Clockticks
    - **Store STLB Hit:** 0.0% of Clockticks
    - **Store STLB Hit:** 0.6% of Clockticks
- **Core Bound:** 9.3% of Pipeline Slots
  - **Divider:** 0.0% of Clockticks
  - **Port Utilization:** 4.9% of Clockticks
    - **Cycles of 0 Ports Utilized:** 12.2% of Clockticks
      - **Serializing Operations:** 12.2% of Clockticks
      - **Mixing Vectors:** 0.0% of uOps
    - **Cycles of 1 Port Utilized:** 6.1% of Clockticks
    - **Cycles of 2 Ports Utilized:** 6.1% of Clockticks
    - **Cycles of 3+ Ports Utilized:** 12.2% of Clockticks
      - **ALU Operation Utilization:** 15.3% of Clockticks
        - **Port 0:** 12.2% of Clockticks
        - **Port 1:** 12.2% of Clockticks
        - **Port 5:** 12.2% of Clockticks
        - **Port 6:** 24.5% of Clockticks
      - **Load Operation Utilization:** 6.1% of Clockticks
        - **Port 2:** 12.2% of Clockticks
        - **Port 3:** 12.2% of Clockticks
      - **Store Operation Utilization:** 12.2% of Clockticks
        - **Port 4:** 12.2% of Clockticks
        - **Port 7:** 0.0% of Clockticks
    - **Vector Capacity Usage (FPU):** 0.0%
- **Average CPU Frequency:** 1.043 GHz
- **Total Thread Count:** 1
- **Paused Time:** 0s

Effective Physical Core Utilization: 22.4% (0.895 out of 4)

> The metric value is low, which may signal a poor physical CPU cores utilization caused by:
>   - load imbalance
>   - threading runtime overhead
>   - contended synchronization
>   - thread/process underutilization
>   - incorrect affinity that utilizes logical cores instead of physical cores

> Explore sub-metrics to estimate the efficiency of MPI and OpenMP parallelism or run the Locks and Waits analysis to identify parallel bottlenecks for other parallel runtimes.

Effective Logical Core Utilization: 11.2% (0.895 out of 8)

> The metric value is low, which may signal a poor logical CPU cores utilization. Consider improving physical core utilization as the first step and then look at opportunities to utilize logical cores, which in some cases can improve processor throughput and overall performance of multi-threaded applications.

Collection and Platform Info:

Application Command Line: ./codecs/hm/decoder/TAppDecoderStatic "-b" "./bin/hm/encoder_intra_main.cfg/CLASS_B/BasketballPass_416x240_50_QP_37_hm.bin"

User Name: root

Operating System: 5.4.0-65-generic DISTRIB_ID=Ubuntu DISTRIB_RELEASE=18.04 DISTRIB_CODENAME=bionic DISTRIB_DESCRIPTION="Ubuntu 18.04.5 LTS"

Computer Name: eimon

Result Size: 9.3 MB

Collection start time: 09:37:43 10/02/2021 UTC

Collection stop time: 09:37:43 10/02/2021 UTC

Collector Type: Event-based sampling driver

CPU:

Name: Intel(R) Processor code named Kabylake ULX

Frequency: 1.992 GHz

Logical CPU Count: 8

Cache Allocation Technology:

Level 2 capability: not detected

Level 3 capability: not detected