

**Elapsed Time:** 0.045s

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

<b>Clockticks:</b>	44,820,000
<b>Instructions Retired:</b>	69,480,000
<b>CPI Rate:</b>	0.645
<b>MUX Reliability:</b>	0.964
<b>Retiring:</b>	36.1% of Pipeline Slots
<b>Light Operations:</b>	38.8% of Pipeline Slots
<b>FP Arithmetic:</b>	0.0% of uOps
<b>FP x87:</b>	0.0% of uOps
<b>FP Scalar:</b>	0.0% of uOps
<b>FP Vector:</b>	0.0% of uOps
<b>Other:</b>	100.0% of uOps
<b>Heavy Operations:</b>	0.0% of Pipeline Slots
<b>Microcode Sequencer:</b>	2.6% of Pipeline Slots
<b>Assists:</b>	0.0% of Pipeline Slots
<b>Front-End Bound:</b>	18.1% of Pipeline Slots
<b>Front-End Latency:</b>	24.1% of Pipeline Slots
<b>ICache Misses:</b>	0.0% of Clockticks
<b>ITLB Overhead:</b>	1.2% of Clockticks
<b>Branch Resteers:</b>	0.0% of Clockticks
<b>Mispredicts Resteers:</b>	0.0% of Clockticks
<b>Clears Resteers:</b>	0.0% of Clockticks
<b>Unknown Branches:</b>	0.0% of Clockticks
<b>DSB Switches:</b>	0.0% of Clockticks
<b>Length Changing Prefixes:</b>	0.0% of Clockticks
<b>MS Switches:</b>	0.0% of Clockticks
<b>Front-End Bandwidth:</b>	0.0% of Pipeline Slots
<b>Front-End Bandwidth MITE:</b>	36.1% of Clockticks
<b>Front-End Bandwidth DSB:</b>	0.0% of Clockticks
<b>(Info) DSB Coverage:</b>	34.8%
<b>Bad Speculation:</b>	6.0% of Pipeline Slots
<b>Branch Mispredict:</b>	0.0% of Pipeline Slots
<b>Machine Clears:</b>	6.0% of Pipeline Slots
<b>Back-End Bound:</b>	39.8% 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).

<b>Memory Bound:</b>	16.8% of Pipeline Slots
<b>L1 Bound:</b>	0.0% of Clockticks
<b>DTLB Overhead:</b>	1.8% of Clockticks
<b>Load STLB Hit:</b>	0.0% of Clockticks
<b>Load STLB Miss:</b>	1.8% of Clockticks
<b>Loads Blocked by Store Forwarding:</b>	0.0% of Clockticks
<b>Lock Latency:</b>	0.0% of Clockticks
<b>Split Loads:</b>	0.0% of Clockticks
<b>4K Aliasing:</b>	0.0% of Clockticks
<b>FB Full:</b>	0.0% of Clockticks
<b>L2 Bound:</b>	0.0% of Clockticks
<b>L3 Bound:</b>	12.0% of Clockticks
<b>Contested Accesses:</b>	0.0% of Clockticks
<b>Data Sharing:</b>	0.0% of Clockticks
<b>L3 Latency:</b>	0.0% of Clockticks
<b>SQ Full:</b>	0.0% of Clockticks
<b>DRAM Bound:</b>	12.0% of Clockticks
<b>Memory Bandwidth:</b>	0.0% of Clockticks
<b>Memory Latency:</b>	12.0% of Clockticks
<b>Store Bound:</b>	0.0% of Clockticks
<b>Store Latency:</b>	0.0% of Clockticks
<b>False Sharing:</b>	0.0% of Clockticks
<b>Split Stores:</b>	0.0% of Clockticks
<b>DTLB Store Overhead:</b>	0.0% of Clockticks
<b>Store STLB Hit:</b>	0.0% of Clockticks
<b>Store STLB Hit:</b>	0.0% of Clockticks
<b>Core Bound:</b>	22.9% of Pipeline Slots

This metric represents how much Core non-memory issues were of a bottleneck. Shortage in hardware compute resources, or dependencies software's instructions are both categorized under Core Bound. Hence it may indicate the machine ran out of an 000 resources, certain execution units are overloaded or dependencies in program's data- or instruction- flow are limiting the performance (e.g. FP-chained long-latency arithmetic operations).

<b>Divider:</b>	0.0% of Clockticks
<b>Port Utilization:</b>	16.4% of Clockticks
<b>Cycles of 0 Ports Utilized:</b>	18.1% of Clockticks
<b>Serializing Operations:</b>	0.0% of Clockticks
<b>Mixing Vectors:</b>	0.0% of uOps
<b>Cycles of 1 Port Utilized:</b>	0.0% of Clockticks
<b>Cycles of 2 Ports Utilized:</b>	12.0% of Clockticks
<b>Cycles of 3+ Ports Utilized:</b>	18.1% of Clockticks
<b>ALU Operation Utilization:</b>	27.1% of Clockticks
<b>Port 0:</b>	24.1% of Clockticks
<b>Port 1:</b>	24.1% of Clockticks
<b>Port 5:</b>	24.1% of Clockticks
<b>Port 6:</b>	36.1% of Clockticks
<b>Load Operation Utilization:</b>	24.1% of Clockticks
<b>Port 2:</b>	36.1% of Clockticks
<b>Port 3:</b>	24.1% of Clockticks
<b>Store Operation Utilization:</b>	24.1% of Clockticks
<b>Port 4:</b>	24.1% of Clockticks
<b>Port 7:</b>	12.0% of Clockticks
<b>Vector Capacity Usage (FPU):</b>	0.0%
<b>Average CPU Frequency:</b>	1.117 GHz
<b>Total Thread Count:</b>	1
<b>Paused Time:</b>	0s

**Effective Physical Core Utilization:** 22.4% (0.896 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% (0.896 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\_randomaccess\_main.cfg/CLASS\_B/BasketballPass\_416x240\_50\_QP\_27\_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:** 9.4 MB

**Collection start time:** 10:02:42 10/02/2021 UTC

**Collection stop time:** 10:02:42 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