

Recommendations:

Hotspots: Start with Hotspots analysis to understand the efficiency of your algorithm.

Use Hotspots analysis to identify the most time consuming functions. Drill down to see the time spent on every line of code.

Microarchitecture Exploration: There is low microarchitecture usage (55.6%) of available hardware resources. of Pipeline Slots

Run Microarchitecture Exploration analysis to analyze CPU microarchitecture bottlenecks that can affect application performance.

Threading: There is poor utilization of logical CPU cores (13.8%) in your application.

Use Threading to explore more opportunities to increase parallelism in your application.

Elapsed Time: 4.861s

CPU:

IPC: 2.091

SP GFLOPS: 0.000

DP GFLOPS: 0.007

x87 GFLOPS: 0.001

Average CPU Frequency: 2.222 GHz

GPU:

Time: 0.0% (0s) of Elapsed time

GPU utilization is low. Consider offloading more work to the GPU to increase overall application performance.

IPC Rate: 1.000

Effective Logical Core Utilization: 13.8% (1.103 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.

Effective Physical Core Utilization: 26.9% (1.075 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.

Microarchitecture Usage: 55.6% of Pipeline Slots

Retiring: 55.6% of Pipeline Slots

Front-End Bound: 21.2% 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.

Back-End Bound: 16.8% of Pipeline Slots

Memory Bound: 5.7% of Pipeline Slots

Core Bound: 11.1% of Pipeline Slots

Bad Speculation: 6.4% of Pipeline Slots

Memory Bound: 5.7% of Pipeline Slots

L1 Bound: 6.2% of Clockticks

L2 Bound: 0.5% of Clockticks

L3 Bound: 3.3% of Clockticks

DRAM Bound: 1.2% of Clockticks

DRAM Bandwidth Bound: 0.0% of Elapsed Time

Store Bound: 1.3% of Clockticks

Vectorization: 0.2% of Packed FP Operations

A significant fraction of floating point arithmetic instructions are scalar. Use Intel Advisor to see possible reasons why the code was not vectorized.

Instruction Mix:**SP FLOPs:**

0.0% of uOps

Packed:

19.1% from SP FP

128-bit:

11.8% from SP FP

A significant fraction of floating point arithmetic vector instructions is executed with a partial vector load. Make sure you compile the code with the latest instruction set or use Intel Advisor for vectorization help.

256-bit:

7.4% from SP FP

Scalar:

80.9% from SP FP

A significant fraction of floating point arithmetic instructions are scalar. Use Intel Advisor to see possible reasons why the code was not vectorized.

DP FLOPs:

0.1% of uOps

Packed:

0.2% from DP FP

128-bit:

0.2% from DP FP

256-bit:

0.0% from DP FP

Scalar:

99.8% from DP FP

A significant fraction of floating point arithmetic instructions are scalar. Use Intel Advisor to see possible reasons why the code was not vectorized.

x87 FLOPs:

0.0% of uOps

Non-FP:

99.8% of uOps

FP Arith/Mem Rd Instr. Ratio: 0.007

The metric value is low. This can be a result of unaligned access to data for vector operations. Use Intel Advisor to find possible data access inefficiencies for vector operations.

FP Arith/Mem Wr Instr. Ratio: 0.016

The metric value is low. This can be a result of unaligned access to data for vector operations. Use Intel Advisor to

find possible data access inefficiencies for vector operations.

GPU Active Time: 0.0%

GPU utilization is low. Consider offloading more work to the GPU to increase overall application performance.

Collection and Platform Info:

Application Command Line: ./codecs/hhi-vvc-codec/encoder/vvencFFapp "-c" "./configs/hhi-vvc-codec/randomaccess_faster.cfg" "-i" "./sequences/CLASS_A/Kimono_1920x1080_24.yuv" "-wdt" "1920" "-hgt" "1080" "-b" "./bin/hhi-vvc-codec/randomaccess_faster.cfg/CLASS_A/Kimono_1920x1080_24_QP_37_hhi-vvc-codec.bin" "-o" "./rec_yuv/hhi-vvc-codec/randomaccess_faster.cfg/CLASS_A/Kimono_1920x1080_24_QP_37_hhi-vvc-codec.yuv" "-fr" "24" "-fs" "0" "-f" "2" "-q" "37"

Operating System: 5.4.0-62-generic DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04 DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.5 LTS"

Computer Name: eimon

Result Size: 3.9 MB

Collection start time: 05:02:14 02/02/2021 UTC

Collection stop time: 05:02:19 02/02/2021 UTC

Collector Type: Event-based sampling driver,Event-based counting driver

CPU:

Name: Intel(R) Processor code named Kabylake
ULX

Frequency: 1.992 GHz

Logical CPU Count: 8

Max DRAM Single-Package Bandwidth: 10.000 GB/s

Cache Allocation Technology:

Level 2 capability: not detected

Level 3 capability: not detected

GPU:

Name:

Display controller: Intel Corporation Device 22807

Vendor:

Intel Corporation

EU Count:

24

Max EU Thread Count:

7

Max Core Frequency:

1.150 GHz