

Intel® QuickSync Video and FFmpeg*

Linux* Transcode Performance

What is Intel® Quick Sync Video?

Intel® Quick Sync Video refers to the dedicated media processing capabilities of Intel® Processor Graphics Technology. The searchable processor specification list at <http://ark.intel.com> is the authoritative site for Intel processor capabilities. Look for Processor Graphics and Intel® Quick Sync Video (available as a technology filter). The [Intel® Media Server Studio website](#) has more info on hardware support including OEMS/ODMs.

Introduction

FFmpeg* 2.8 and forward includes Intel® Quick Sync Video accelerated h264_qsv, mpeg2_qsv, and hevc_qsv codecs. These provide a fast time-to-market path to server transcode solutions based on Intel® Xeon® processors with processor graphics.

Installation is covered in a separate white paper: Intel® Quick Sync Video and FFmpeg: Install & Validation. For a quick overview please see <https://software.intel.com/en-us/articles/accessing-intel-media-server-studio-for-linux-codecs-with-ffmpeg>.

The relatively simple task of installing opens access to Intel Media Server Studio codecs in FFmpeg, providing a new set of performance/quality options to build solutions for today and tomorrow. **The diagram below shows the number of concurrent 1920x1080p30 FFmpeg transcodes on a single Intel® Xeon® processor E3-1285L v4.** The rest of the paper gives more context for how these results were obtained with details to quick-start your implementations.

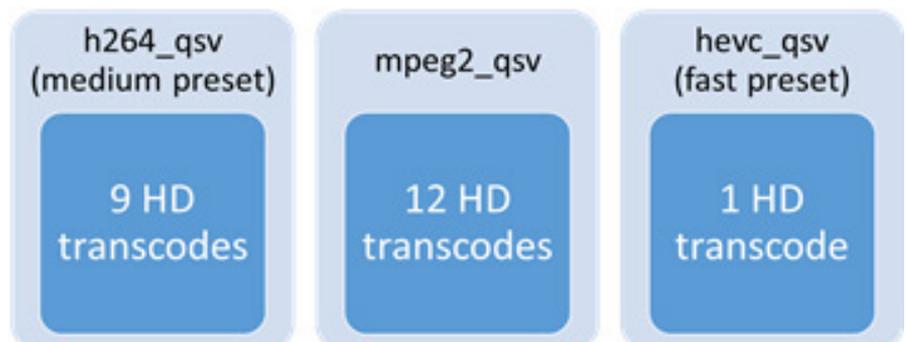


Table of Contents

Introduction 1

Methodology..... 2

 System Configuration..... 2

 Performance comparison notes 3

 Quality comparison notes.... 3

Implementation details:

h264_qsv 3

 H264 multiple concurrent transcode output performance..... 3

 Encode quality/target usage presets 4

 Concurrent perf characteristics 6

Implementation details:

mpeg2_qsv 7

 Concurrent perf characteristics 8

Implementation details:

hevc_qsv 9

 Performance 9

 Preset behavior 10

 Command lines..... 11

Conclusion..... 12

Appendix 12

 Additional information on Intel® Media Server Studio 12

 Test file preparation 12

 Known issues 13

 System tuning..... 13

Methodology

This paper provides some basic descriptions of codec behavior and performance/quality tradeoffs for the FFmpeg/libavcodec *_qsv codecs. The intent is to provide a foundation for further evaluation based on your inputs and scenarios. For simplicity, results are based on a single hardware configuration with performance tests on a single “mosaic” input.

Test Input Details

TEST INPUTS



[crowd_run](#)
1920x1080 y4m



[ducks_take_off](#)
1920x1080 y4m



[park_joy](#)
1920x1080 y4m



[blue_sky](#)
1920x1080 y4m



Mosaic of above
(2000 frames, resized to 1920x1080. See preparation details in appendix)

These inputs were chosen because they are industry standard encoder tests, commonly used for validation and appearing in many publications. They align with other results from Intel to make comparison easier. The additional mosaic is used for h264_qsv and mpeg2_qsv performance tests. Increasing input sequence length to 2000 minimizes the impact of session initialization and termination.

System Configuration

Software:

- Operating System: CentOS* 7.1
- [Intel® Media Server Studio 2015 R6](#)
- [FFmpeg 2.8.1](#)

Hardware:

- Platform: [Intel® Server Board S1200V3RPM](#) (NOTE: only M version of S1200V3RPx board family supports integrated graphics)
- Processor: Intel® Xeon® processor E3-1285L v4 @ 3.4 GHz (more info at ark.intel.com)

Note: On the tested configuration default GPU frequency and performance governor settings were sufficiently optimal. On some systems larger system tuning effects may be seen -- especially with the hybrid GPU accelerated HEVC codec (hevc_qsv). See the appendix for more information.

Performance comparison notes

Performance numbers in this paper are based on the commonly used scenario of one decode feeding multiple encodes, sometimes called 1:N. This allows a single input to be processed and encoded at multiple resolutions, bitrates, etc. For simplicity this paper uses multiple encodes at same resolution.

Quality comparison notes

This paper does not attempt to compare quality vs. other codecs. The goal is to show basic quality/performance tradeoffs for the *_qsv codecs to help you start your own comparisons.

Tools to consider:

- Intel Media Server Studio provides the [Video Quality Caliper](#) and [Video Pro Analyzer](#) tools with a rich set of features to examine encoder output and to make comparisons easier.
- The [precompiled psnr utility](#) from the libyuv project can be used for simple quality metric automation. This paper uses global PSNR-Y from this tool.

Implementation details: h264_qsv

Realtime 1080p h264 transcode density for "mosaic" input, realtime=30FPS

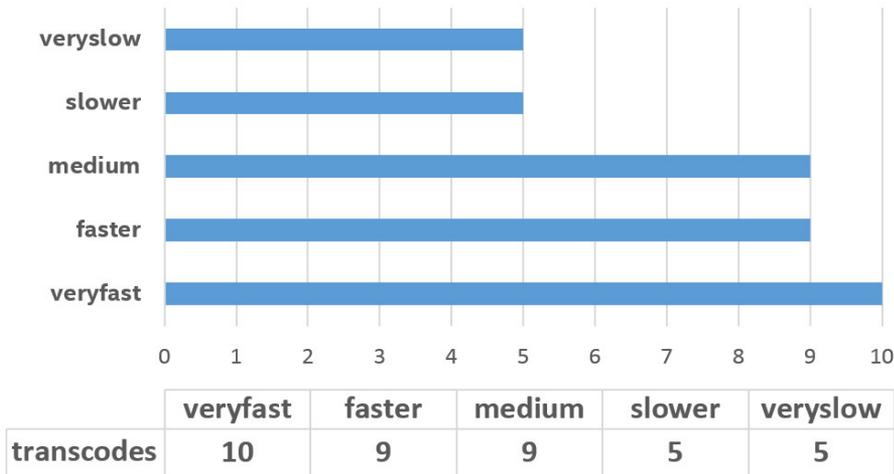
Intel hardware provides fast decode, encode, and transcode for h264. Many of the benefits of Intel acceleration are available using the FFmpeg codec h264_qsv.

See also: [Intel® Quick Sync Video Technology on Intel® Iris™ Graphics and Intel® HD Graphics family – Flexible Transcode Performance and Quality](#)

H264 Multiple Concurrent Transcode Output Performance

Traditional performance/quality tradeoffs of software codecs are disrupted by high speed h264 hardware acceleration. In many cases hardware acceleration provides a much higher FPS boost than would be possible with a software implementation.

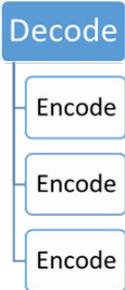
Concurrent 1920x1080p30 "mosaic" transcodes by preset



What is transcode density?
 This metric is the number of streams which can be transcoded concurrently while maintaining a specified frames per second (FPS) throughput. The FPS must be greater than playback frame rate to be considered "realtime". For this paper realtime=30 FPS.

EXAMPLE H264_QSV COMMAND LINE

h264 tests are 1:N: This specifies one input (-i) and several parameter sequences ending with an output. Usually other steps like resize would be included, but, for simplicity, the performance tests for this paper specify multiple outputs at the same bitrate, and with the same resolution as the input.



```

ffmpeg -y -i infile \
-vframes 2000 -b:v 8000K -vcodec h264_qsv -preset veryfast out00.h264 \
-vframes 2000 -b:v 8000K -vcodec h264_qsv -preset veryfast out01.h264 \
-vframes 2000 -b:v 8000K -vcodec h264_qsv -preset veryfast out02.h264
    
```

Encode quality/target usage presets

PRESET	TU	NOTES
veryfast	7	Fastest setting, biggest quality drop
faster	6	Close to TU4/5 quality, faster performance
fast	5	Between 6 and 4
medium	4	Balanced performance and quality
slow	3	Between 4 and 2
slower	2	Close to TU1 quality, faster performance
veryslow	1	Best quality

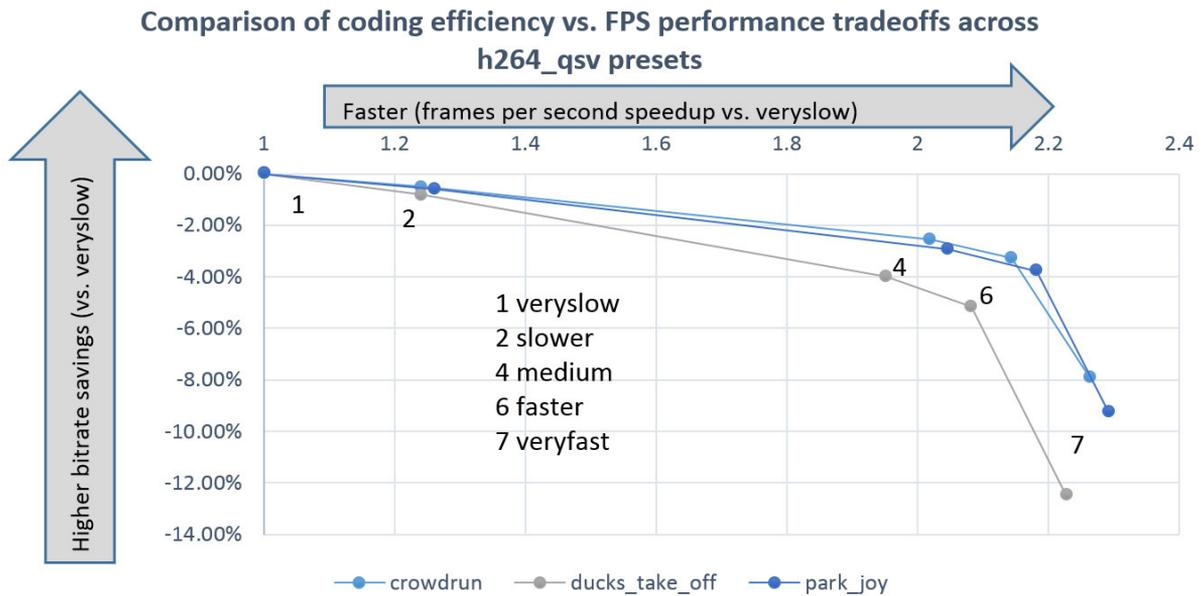
Preset behavior is highly input dependent. As a general summary, TUs 1-6 (veryslow to faster) provide a relatively smooth range of performance/quality tradeoffs. AVC TU7 is a special case. It is tied to quality from earlier hardware generations to simplify describing cross-generation performance improvements. TU7/veryfast quality can be significantly lower than the rest of the TU/preset spectrum.

The graph below illustrates the quality/performance tradeoff patterns for the crowd_run, ducks, and park_joy test videos. The TU1/veryslow and TU2/slow presets group together as high quality options. TU4/medium is a good balance, relatively small quality loss for ~2x speedup. TU6/faster has a similar relationship to TU4/medium as TU2 to TU1: moderate speedup with relatively small quality loss. The quality loss for TU7/veryfast is large relative to the speed improvement vs. TU6/faster.

While objective metrics like PSNR are limited, the encode fidelity differences are more than mathematical. Though the quality range for h264_qsv is not wide, presets can make a visible difference. Below is a subset of frame four of the crowd_run sequence, encoded at 6Mbps with TU7/veryfast and TU1/veryslow using lookahead (LA) bitrate control (the default in h264_qsv). This is at the low end of usable bitrates for this sequence at HD resolution. The subjective quality of TU1/veryslow vs. TU7/veryfast corresponds to the large difference in objective metrics.

What is coding efficiency?

The coding efficiency metric measures the relative bitrate required to achieve a specified quality level. The goal of most codecs is to reduce the number of bits required to retain quality – or maximize quality at a given bitrate. Here h264_qsv veryslow is the standard, and the graph shows that veryfast needs to use a ~10 percent higher bitrate to achieve a similar PSNR score.



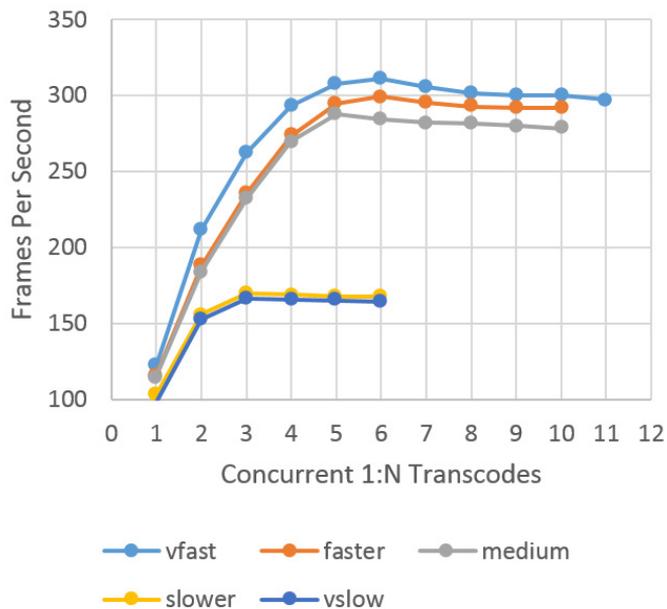
CODING EFFICIENCY LOSS (BITRATE INCREASE REQUIRED TO REGAIN SAME QUALITY) AND FPS SPEEDUP BY H264_QSV PRESET, COMPARED TO H264_QSV VERYSLOW.

	TU2/SLOW		TU4/MEDIUM		TU6/FASTER		TU7/VERYFAST	
	speedup	BD %	speedup	BD %	speedup	BD %	speedup	BD %
crowdrun	1.2X	-0.5%	2.0X	-2.5%	2.1X	-3.3%	2.3X	-7.9%
ducks	1.2X	-0.8%	2.0X	-4.0%	2.1X	-5.2%	2.3X	-12.5%
parkjoy	1.3X	-0.6%	2.0X	-2.9%	2.2X	-3.8%	2.3X	-9.2%
average	1.2X	-0.6%	2.0X	-3.1%	2.1X	-4.1%	2.3X	-9.9%



Concurrent performance characteristics

h264_qsv total "mosaic" frames per second by preset



Transcode capacity/density is not fully represented by single transcode FPS. Experiments based on 4-6 concurrent transcodes give a better picture of FPS capacity, with relatively minor (a few percent) additional overhead as number of streams increases. Optimal settings depend on inputs, preset used, and other factors like async depth.

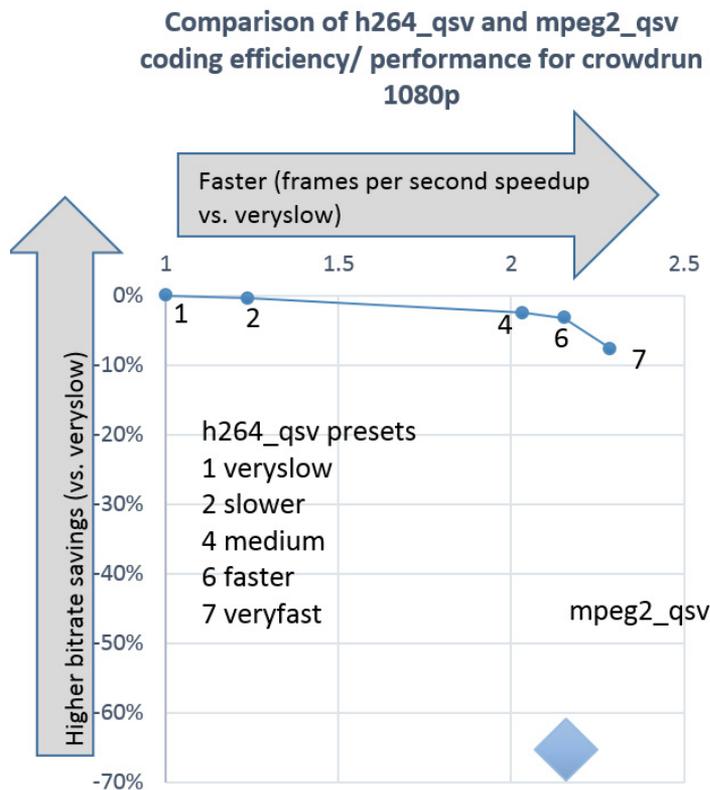
TOTAL FRAMES PER SECOND BY PRESET AS NUMBER OF CONCURRENT TRANSCODES INCREASES:

N	VFAST	FASTER	MEDIUM	SLOWER	VSLOW
1	122.1	115.4	114.8	102.8	98.1
2	211.8	187.8	184	155.8	152.8
3	262.2	235.8	232.5	169.5	166.2
4	293.2	273.6	269.6	168.8	166
5	307.5	294.5	287.5	168	165
6	310.8	299.4	284.4	168	164.4
7	305.2	295.4	282.1		
8	301.6	292.8	281.6		
9	299.7	291.6	279.9		
10	300	292	278		
11	297				

Implementation details: mpeg2_qsv

Realtime 1080p transcode density for "mosaic" input, realtime=30FPS

Unlike h264_qsv, which has a range of preset options to consider, all mpeg2_qsv presets execute as TU1/slow. As expected with mpeg2, large bitrate increases are required to match even the lowest range of h264 quality. The coding efficiency graph below shows that mpeg2_qsv bitrate must be increased by ~60 percent to match h264_qsv TU7/veryfast.



FRAMES PER SECOND AND SPEEDUP/CODING EFFICIENCY LOSS VS. H264_VERYLOW FOR EACH PRESET AND MPEG2_QSV.

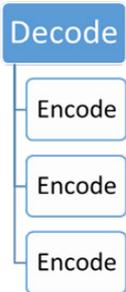
	AVG FPS	SPEED-UP	H264	MPEG2
h264 veryslow	130.4	1.0	0.00%	
h264 slower	161.7	1.2	0.41%	
h264 medium	265.2	2.0	2.52%	
h264 faster	280.6	2.2	3.24%	
h264 veryfast	298.4	2.3	7.62%	
mpeg2	281.3	2.2		65.33%

As with h264_qsv, the performance data collected is for a 1:N pipeline.

EXAMPLE MPEG2_QSV COMMAND LINE:

Mpeg2 tests are 1:N.

This specifies one input (-i) and several parameter sequences ending with an output. Usually other steps like resize would be included, but, for simplicity, the performance tests for this paper specify multiple outputs at the same bitrate, and with the same resolution as the input.

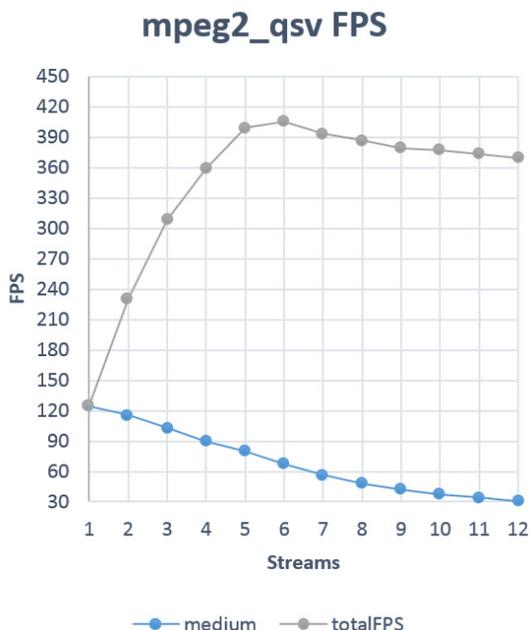


```

ffmpeg -y -i infile \
-vframes 2000 -b:v 8000K -vcodec mpeg2_qsv -preset veryfast out00.h264 \
-vframes 2000 -b:v 8000K -vcodec mpeg2_qsv -preset veryfast out01.h264 \
-vframes 2000 -b:v 8000K -vcodec mpeg2_qsv -preset veryfast out02.h264
  
```

Concurrent performance characteristics

Frames per second per stream and total frames per second for the “mosaic” test input by number of concurrent mpeg2 transcodes:



N	MPEG2 FPS	TOTALFPS
1	124.8	124.8
2	115.3	230.6
3	102.9	308.7
4	89.7	358.8
5	79.8	399
6	67.5	405
7	56.2	393.4
8	48.3	386.4
9	42.1	378.9
10	37.7	377
11	33.9	372.9
12	30.8	369.6

Optimal settings for mpeg2_qsv also depend on inputs, preset, async depth, etc. In addition, because the mpeg2_qsv implementation is more heavily dependent on EUs, this adds additional overhead vs. h264_qsv. Thus the relative performance drop as load increases can be larger than for h264_qsv.

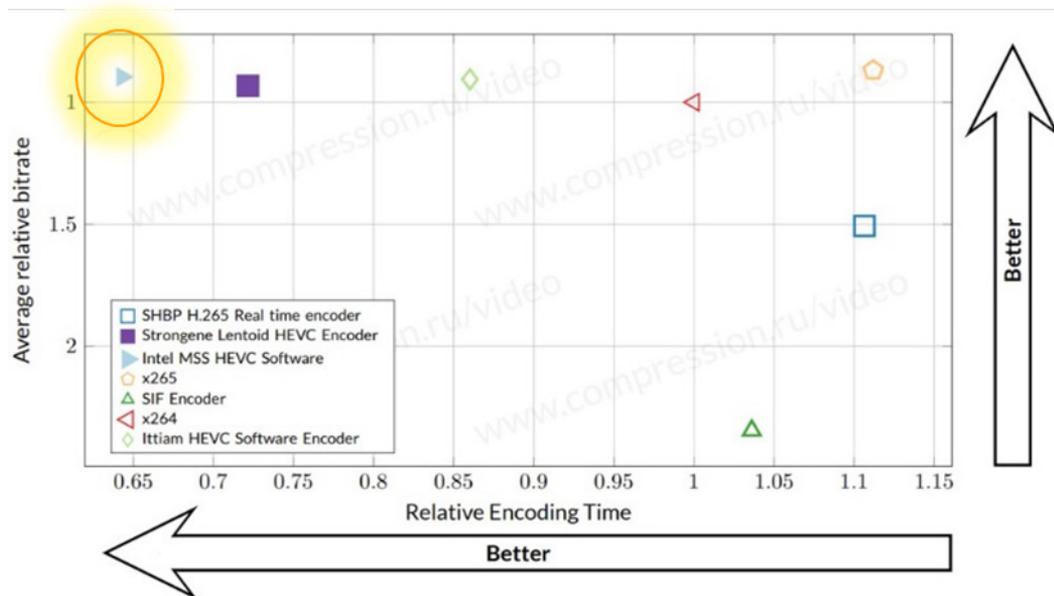
Implementation details: hevc_qsv

Realtime 1080p HEVC transcode density for "mosaic" input, realtime=30FPS

HEVC is gathering momentum as the successor to h264. Several flavors of Intel's HEVC codecs are available via the hevc_qsv interface.

- Intel® Media Server Studio Professional Edition offers software and GPU accelerated HEVC implementations as plugins.
- While not available yet for Linux, future hardware generations will have a higher performance GPU-only HEVC implementation as well.

The [2015 Moscow State University comparison of HEVC implementations](#) shows that Intel's HEVC has a great combination of performance and quality

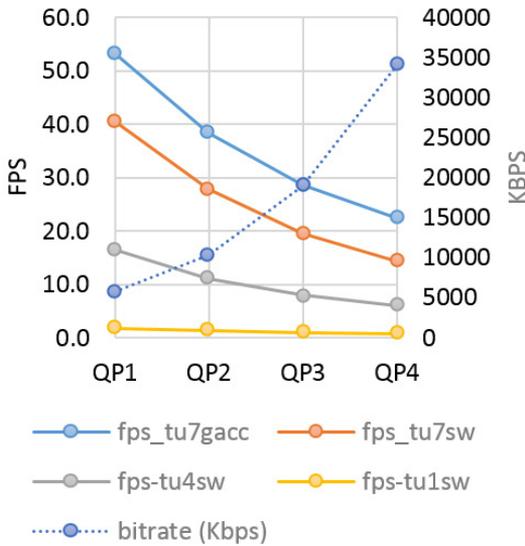


QP settings from that paper are used here instead of bitrate to simplify cross-referencing. Since HEVC has a lower frame rate than H264 or mpeg2, FFmpeg copy overhead is relatively smaller and performance is closer to using the Intel Media Server Studio SDK directly.

Performance

On the system tested, only one 1920x1080 encode at ≥ 30 FPS is possible. The results below are an average of frames per second results for crowd_run, ducks_take_off and parkjoy at the QP values specified by the [Intel HEVC white paper](#). Frames per second performance varies significantly by QP/bitrate for the HEVC software and Gacc/hybrid plugins.

crowdrun 1080P HEVC FPS relation to preset and bitrate

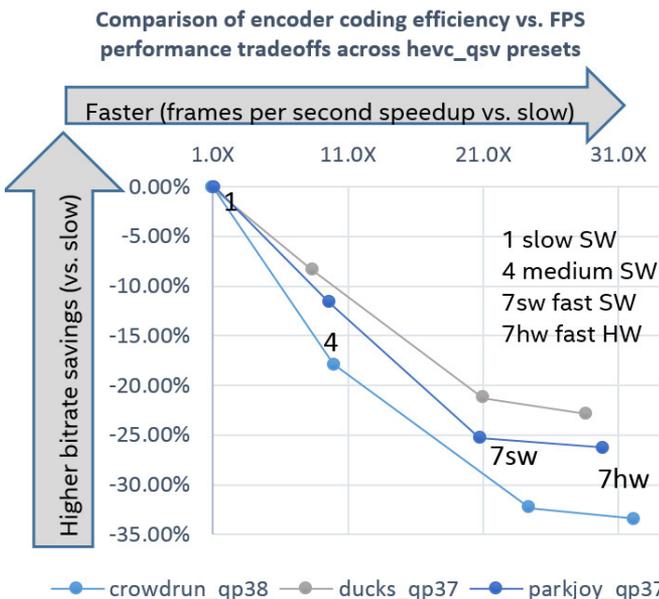


FRAMES PER SECOND FOR EACH QP IN THE HEVC PAPER FOR CROWD_RUN, BY PRESET. AVERAGE BITRATE FOR ALL PRESETS FOR EACH QP INTENDED TO SHOW FRAMERATE RANGE CHANGES ACROSS BITRATES.

	QP	FPS_TU7 GACC	FPS_TU7 SW	FPS-TU4 SW	FPS-TU1 SW	BITRATE (KBPS)
1	38	53.3	40.5	16.4	1.7	5727
2	34	38.3	27.8	11.1	1.3	10291
3	30	28.6	19.5	7.9	1.0	18907
4	26	22.4	14.3	5.9	0.8	34167

Preset behavior

For hevc_qsv there are effectively four presets: software slow, software medium, software fast, and GPU accelerated fast. These provide a wider range of performance/quality tradeoffs than the other *_qsv codecs. The GPU accelerated version of TU7/fast provides similar quality as the software HEVC “fast” preset with a significant speed boost.



SPEEDUP AND CODING EFFICIENCY LOSS (BITRATE INCREASE REQUIRED TO REACH SAME QUALITY) RELATIVE TO HEVC_QSV SLOW PRESET

CROWDRUN

preset	speedup	crowdrun_qp38
med	9.9X	-17.89%
fastSW	24.5X	-32.30%
fastHW	32.2X	-33.39%

DUCKS

preset	speedup	ducks_qp37
med	8.4X	-8.39%
fastSW	21.0X	-21.25%
fastHW	28.7X	-22.85%

PARKJOY

preset	speedup	parkjoy_qp37
med	9.6X	-11.58%
fastSW	20.7X	-25.25%
fastHW	29.8X	-26.22%

Comparing h264_qsv and hevc_qsv (both medium preset) in CQP mode. HEVC software and GPU accelerated implementations provide noticeable quality improvements vs. h264 at the same bitrate.



Command Lines

There are some command line differences vs. h264_qsv and mpeg2_qsv. Intel HEVC implementations are available to the Intel Media Server Studio SDK via the professional edition as plugins. Since QP is used by the HEVC white paper, this must be specified for QPI, QPP, and QPP instead of bitrate.

hevc_qsv tests are
encode only

Raw YUV

Encode

Example command line:

Note that constant QP is used instead of specifying a target bitrate. The FFmpeg -q parameter sets QPP. The HEVC paper specifies QPI values, with QPP=QPB=QPI+1.

```
ffmpeg -y -framerate 50 -s 1920x1080 -i infile.yuv \
-vframes 500 -vcodec hevc_qsv -preset fast \
-load_plugins (Plugin GUID) -q (QPI+1) \
-i_qfactor 1.0 -i_qoffset -1.0 -b_qfactor 1.0 -b_qoffset 0.0 out00.hevc
```

HEVC plugin GUIDs can change between releases. Check `/opt/intel/mediasdk/plugins/plugins.cfg` for the correct GUID.

For Intel Media Server Studio 2015 R6 the HEVC encoder GUIDs are:

- GPU accelerated/hardware implementation: e5400a06c74d41f5b12d430bbaa23d0b
- Software implementation: 2fca99749fdb49aeb121a5b63ef568f7

Conclusion

Intel® Xeon® processors with Intel® Quick Sync Video provide compelling solutions for today's and tomorrow's rapidly increasing media workloads. Today's server media solutions are complex and growing to meet a wide variety of new requirements. It isn't necessary to re-architect them to start seeing the benefits of Intel's hardware and software engineering. Simply installing the Media Server Studio components (including HEVC plugins from the Professional edition if needed) and recompiling FFmpeg makes them available for your applications. This paper assumes successful install and focuses on documenting some basic patterns of expected quality/performance behavior for each codec to save time so you can focus on accelerating your scenarios and your solutions.

For more complete information about compiler optimizations, see our [Optimization Notice](#).

Appendix

Additional information on Intel® Media Server Studio

Learn More:

- [Intel® Media Server Studio](#) (available in three editions)
- [Read](#) User Reviews
- Frost & Sullivan Awards Intel with [2015 Global Video Encoding and Transcoding Technology Innovation Leadership](#)

MSU 2015 HEVC/H.265 Video Codecs Comparison: (compression.ru/video/codec_comparison/hevc_2015).

The Intel HEVC White Paper: https://software.intel.com/sites/default/files/managed/d7/07/Intel_HEVCWhitepaper_v1%2050_R6_24Jun2015.pdf.

More information on Intel® Quick Sync Video h264: [Intel® Quick Sync Video Technology on Intel® Iris™ Graphics and Intel® HD Graphics family – Flexible Transcode Performance and Quality](#).

Test file preparation

Download 1920x1080 inputs from Xiph.org:

```
crowd_run_1080p50.y4m
ducks_take_off_1080p50.y4m
park_joy_1080p50.y4m
blue_sky_1080p25.y4m
```

Convert to YUV

```
$ ffmpeg -i in.y4m out.yuv
```

Make blue_sky as long as the other inputs

```
$ cat blue_sky_1080p25.yuv blue_sky_1080p25.yuv blue_sky_1080p25.yuv > blue_sky3.yuv
```

Create mosaic

```
ffmpeg -y -s 1920x1080 -i ducks_take_off_1080p50.yuv -s 1920x1080 -i crowd_run_1080p50.yuv -s 1920x1080 -i park_joy_1080p50.yuv -s 1920x1080 -i blue_sky3.yuv -filter_complex "nullsrc=size=1920x1080 [base]; [0:v] setpts=PTS-STARTPTS, scale=960x540 [upperleft]; [1:v] setpts=PTS-STARTPTS, scale=960x540 [upperright]; [2:v] setpts=PTS-STARTPTS, scale=960x540 [lowerleft]; [3:v] setpts=PTS-STARTPTS, scale=960x540 [lowerright]; [base][upperleft] overlay=shortest=1 [tmp1]; [tmp1][upperright] overlay=shortest=1:x=960 [tmp2]; [tmp2][lowerleft] overlay=shortest=1:y=540 [tmp3]; [tmp3][lowerright] overlay=shortest=1:x=960:y=540" -c:v libx264 -crf 10 out_tmp.h264
```

create list.txt file with 4 entries, 4*500=2000 frames

```
$cat list.txt  
file 'out_tmp.h264'  
file 'out_tmp.h264'  
file 'out_tmp.h264'  
file 'out_tmp.h264'
```

Run ffmpeg with concat filter to create a longer test sequence

```
$ffmpeg -y -f concat -i list.txt -c copy test_sequence.h264
```

Known issues

1. The FFmpeg *_qsv codecs do not always drain all frames from hardware buffers. Output streams may omit the last few frames. Effects on performance measurements are minimal. Quality metric calculations are based on input frame count -4. This will be fixed in future updates to FFmpeg.
2. The Intel *_qsv codecs present compelling performance, quality, and TCO opportunities and will continue to improve. However, especially for h264_qsv and mpeg2_qsv, significant additional performance is possible via:
 - Use of video instead of system memory
 - Implementing gpu-only pipelines with multiple decode, frame processing (i.e. resize), and encode executing asynchronously
 - Removing redundant synchronization
3. CPU utilization with FFmpeg *_qsv codecs is usually lower than with pure software alternatives. However, CPU use can still be >90 percent. This is primarily due to extra copies between CPU and GPU memory.

System tuning

Note: these were not necessary on the system tested, and are included only as a reference to help performance investigation on your system.

Governor:

(as root)

```
#cpupower frequency-set -g performance
```

Pin GPU frequency:

(as root)

```
#cp /sys/class/drm/card0/gt_max_freq_mhz /sys/class/drm/card0/gt_max_freq_mhz
```

Check GPU frequency:

```
$cat /sys/class/drm/card0/gt_cur_freq_mhz
```



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order. Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's Web site at www.intel.com.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.

Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit <http://www.intel.com/performance>.

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804

Intel, the Intel logo, Xeon, Core, and Iris are trademarks of Intel Corporation in the U.S. and/or other countries.

Copyright © 2016 Intel Corporation. All rights reserved. Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.