

Tvheadend - Feature #5783

HW acceleration decoding on input stream (NVIDIA)

2019-11-21 23:33 - Ronny M.

Status:	Fixed	Start date:	2019-11-21
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Transcoding	Estimated time:	0.00 hour
Target version:	4.4		

Description

Hello @all,

I observed a strange issue when using NVIDIA Quattro P5000 to decode MPEGTS, H265 or H264 input streams.

```
nvidia-smi pmon
```

# gpu	pid	type	sm	mem	enc	dec	command
# Idx	#	C/G	%	%	%	%	name
0	31069	C	0	0	4	0	tvheadend
0	31069	C	0	0	3	0	tvheadend
0	31069	C	0	0	3	0	tvheadend
0	31069	C	0	0	4	0	tvheadend
0	31069	C	0	0	4	0	tvheadend
0	31069	C	0	0	4	0	tvheadend
0	31069	C	0	0	4	0	tvheadend
0	31069	C	0	0	3	0	tvheadend

As you can see decoding is 0. I would expect some output for the GPU, when hwaccel is active.

I was not able to find something on the logs indicating any information on hwaccel.

```
Nov 21 23:24:26 TVHeadend tvheadend[31069]: idnode: find node (null) class profile
Nov 21 23:24:26 TVHeadend tvheadend[31069]: idnode: find node 0dba7d9c601d823d54bd345d95b806d0
class profile
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7: creating subscription for rbb
Berlin HD weight 0 using profile h265
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7: find service for rbb Berlin HD
weight 100
Nov 21 23:24:26 TVHeadend tvheadend[31069]: service: esfilter:
"Astra 19.2/10891.25H/rbb Berlin HD" AUDIO 005 002 05316 AC3 ger EXCLUSIVE
Nov 21 23:24:26 TVHeadend tvheadend[31069]: service: esfilter:
"Astra 19.2/10891.25H/rbb Berlin HD" TELETEXT 004 001 05314 TELETEXT und IGNORE
Nov 21 23:24:26 TVHeadend tvheadend[31069]: mpegts: 10891.25H in Astra 19.2 - tuning on DVB-S #06
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7: linking sub 0x7f48f407da20 to svc
0x55762897f610 type 0
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7:
"10.X.X.X [ kodi | Kodi Media Center ]" subscribing on channel "rbb Berlin HD", weight: 100,
adapter: "DVB-S #06", network: "Astra 19.2", mux: "10891.25H", provider: "ARD", service:
"rbb Berlin HD", profile="h265", hostname="10.X.X.X, username="kodi", client="Kodi Media Center"
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7: chain 01: input
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7: chain 02: parser input
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7: chain 03: profile input
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7: chain 04: tsfix input
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7: chain 05: transcoder input
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7: chain 06: profile sharer input
Nov 21 23:24:26 TVHeadend tvheadend[31069]: subscription: 01A7: chain 07: wtimeshift input
Nov 21 23:24:26 TVHeadend tvheadend[31069]: TS: hbbtv: pid AA tableid 74 extraid 0000000000000010
len 433
Nov 21 23:24:26 TVHeadend tvheadend[31069]: TS: hbbtv: section 0 last 0 ver 0 (ver 0 st 2 incomp
0 comp 1)
Nov 21 23:24:26 TVHeadend tvheadend[31069]: TS: hbbtv: skip, already complete (2)
```

```
Nov 21 23:24:26 TVHeadend tvheadend[31069]: TS: hbbtv: pid AA tableid 74 extraid 0000000000000010 len 433
Nov 21 23:24:26 TVHeadend tvheadend[31069]: TS: hbbtv: section 0 last 0 ver 0 (ver 0 st 2 incomp 0 comp 1)
Nov 21 23:24:26 TVHeadend tvheadend[31069]: TS: hbbtv: skip, already complete (2)
Nov 21 23:24:26 TVHeadend tvheadend[31069]: transcode: 0012: 01:H264: ==> Using profile h265_nvenc
Nov 21 23:24:26 TVHeadend tvheadend[31069]: transcode: 0012: 05:AC3: ==> Copy
Nov 21 23:24:26 TVHeadend tvheadend[31069]: transcode: 0012: 06:DVBSUB: ==> Filtered out
Nov 21 23:24:26 TVHeadend tvheadend[31069]: idnode: find node 0e0a7b3d28c1674083858d21bd63290a class service
Nov 21 23:24:26 TVHeadend tvheadend[31069]: idnode: find node 0e0a7b3d28c1674083858d21bd63290a class service
Nov 21 23:24:26 TVHeadend tvheadend[31069]: idnode: find node 0e0a7b3d28c1674083858d21bd63290a class service
Nov 21 23:24:26 TVHeadend tvheadend[31069]: idnode: find node 0e0a7b3d28c1674083858d21bd63290a class service
Nov 21 23:24:26 TVHeadend tvheadend[31069]: idnode: find node 0e0a7b3d28c1674083858d21bd63290a class service
Nov 21 23:24:26 TVHeadend tvheadend[31069]: idnode: find node 0e0a7b3d28c1674083858d21bd63290a class service
Nov 21 23:24:26 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: opts: bf=0,gpu=0,rc=constqp,quality=25,tvh_filter_deint=1,width=1280,height=720,pix_fmt=0,tvh_require_mta=1
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters: source args: 'video_size=1280x720:pix_fmt=yuv420p:time_base=1/100:pixel_aspect=1/1'
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters: filters: 'yadif'
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: +-----+
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: | in |default--[1280x720 1:1 yuv420p]--Parsed_yadif_0:default
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: | (buffer) |
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: +-----+
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: +-----+
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: Parsed_yadif_0:default--[1280x720 1:1 yuv420p]--default| out |
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: | (buffersink) |
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: +-----+
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: +-----+
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: in:default--[1280x720 1:1 yuv420p]--default| Parsed_yadif_0 |default--[1280x720 1:1 yuv420p]--out:default
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: | (yadif) |
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: filters dump: +-----+
Nov 21 23:24:27 TVHeadend tvheadend[31069]: transcode: 0012: 01:HEVC: [h264 => hevc_nvenc]: opts: quality=25
Nov 21 23:24:27 TVHeadend tvheadend[31069]: TS: hbbtv: pid AA tableid 74 extraid 0000000000000010 len 433
Nov 21 23:24:27 TVHeadend tvheadend[31069]: TS: hbbtv: section 0 last 0 ver 0 (ver 0 st 2 incomp 0 comp 1)
Nov 21 23:24:27 TVHeadend tvheadend[31069]: TS: hbbtv: skip, already complete (2)
Nov 21 23:24:27 TVHeadend tvheadend[31069]: TS: hbbtv: pid AA tableid 74 extraid 0000000000000010 len 433
Nov 21 23:24:27 TVHeadend tvheadend[31069]: TS: hbbtv: section 0 last 0 ver 0 (ver 0 st 2 incomp 0 comp 1)
Nov 21 23:24:27 TVHeadend tvheadend[31069]: TS: hbbtv: skip, already complete (2)
Nov 21 23:24:27 TVHeadend tvheadend[31069]: idnode: find node 0e0a7b3d28c1674083858d21bd63290a class service
```

So I presume a small bug in the hwaccel module.

If you need more information I'm willing to help.

BR,
Ronny

History

#1 - 2019-11-22 03:28 - Flole Systems

Looks like nvenc is just doing encoding and not decoding, I observed this aswell a while ago. I would like to get this changed aswell but unfortunately I haven't really looked into transcoding and for me this is not really causing problems at the moment unless I try to use it for a h265 encoded 4k channel.

#2 - 2019-11-25 03:03 - Andreas Fornberg

Fole Systems wrote:

Looks like nvenc is just doing encoding and not decoding, I observed this aswell a while ago. I would like to get this changed aswell but unfortunately I haven't really looked into transcoding and for me this is not really causing problems at the moment unless I try to use it for a h265 encoded 4k channel.

For decoding you need nvdec. nvenc is only for encoding.

#3 - 2019-11-25 06:53 - Flole Systems

Yes I'm aware of that, but I thought that selecting the nvenc encoder automatically selects the nvdec decoder, which is not the case. I've looked at the ffmpeg makefile that's part of tvheadend and it doesn't contain the required flags to compile for nvdec. I guess that's the first thing we need to fix, and then we can start to look at how to tell tvheadend to actually use it.

#4 - 2019-11-25 17:56 - Ronny M.

Hallo,
As far as I had come, it seems only encoding makes.
The decoding is not addressed at all, or is not in the compile of ffmpeg.
At vaapi I also saw some parts in the source, but not at nvidia Decoding.
Who can take care of this? Because I can unfortunately no programing C.

maybe even the current form is wrong,
because with a SPAWN profile it looks right ...
ffmpeg [...] -hwaccel nvdec [...] -i pipe:0 -c:v hevc_nvenc [...] -f mpegts pipe:1

BR
Ronny

#5 - 2019-11-26 02:54 - Flole Systems

I've done some investigation regarding this topic and came up with this patch to enable some nvidia related hardware acceleration functions (nvdec and cuvid) in the ffmpeg build. However, this is not enough, we probably need a copy of src/transcoding/transcode/hwaccels/vaapi.c and a few additions in src/transcoding/transcode/hwaccels/hwaccels.c for nvdec/cuvid.

```
--- Makefile.ffmpeg      2019-03-13 17:49:41.330644268 +0100
+++ Makefile.ffmpeg      2019-11-25 23:47:07.594583854 +0100
@@ -558,8 +558,10 @@
 
 ifeq (yes,$(CONFIG_NVENC))
 
-EXTLIBS += nvenc
+EXTLIBS += nvenc nvdec cuvid
+ENCODERS += h264_nvenc hevc_nvenc
+DECODERS += h264_nvdec hevc_nvdec vp8_nvdec mjpeg_nvdec h264_cuvid hevc_cuvid vp8_cuvid mjpeg_cuvid mpeg1_cuv
id mpeg2_cuvid mpeg4_cuvid
+HWACCELS += h264_nvdec hevc_nvdec vp8_nvdec mjpeg_nvdec h264_cuvid hevc_cuvid vp8_cuvid mjpeg_cuvid mpeg1_cuv
id mpeg2_cuvid mpeg4_cuvid
 
 $(LIB_ROOT)/$(FFNVCODEC)/.tvh_download:
     $(call DOWNLOAD,$(FFNVCODEC_URL),$(LIB_ROOT)/$(FFNVCODEC_TB),$(FFNVCODEC_SHA1))
 
--- configure            2019-11-26 01:47:33.113533125 +0100
+++ configure            2019-11-26 01:47:27.133390956 +0100
@@ -600,6 +600,7 @@
     #check_cc_header nvEncodeAPI || \
     # die "NVENC library (https://developer.nvidia.com/nvidia-video-codec-sdk) not found"
```

```

    enable nvenc
+   enable hwaccels
    else
        disable nvenc
    fi

```

#6 - 2019-11-26 09:51 - Joe User

A good description is here: <https://devblogs.nvidia.com/nvidia-ffmpeg-transcoding-guide/>

#7 - 2019-11-26 15:46 - Flole Systems

The description does not cover the c implementation though. Still for someone who doesn't know how this works it could be helpful, but it's not helping with the implementation unfortunately.

#8 - 2019-11-26 17:34 - Ronny M.

Hallo,
 so i have checked this code, and i have a little bit change this ;-)
 now ready for usage, but "NVDEC" is not working in TVHeadend only all ffmpeg HWACCELS option is now ready for TVHaedend.
 And you must install NVIDIA CUDA for ext Libs. (see in coments)

Now you can see is the HWACCELS section filled.

BR
 Ronny

```

index 8768de8b1..e6792e9c2 100644
--- a/Makefile.ffmpeg
+++ b/Makefile.ffmpeg
@@ -164,10 +164,12 @@ endif
    ECFLAGS    := -I$(EPREFIX)/include $(CFLAGS_PI)
    ELIBS      := -L$(EPREFIX)/lib -ldl

+
    CONFIGURE := FFMPEG_PREFIX=$(EPREFIX) \
                PKG_CONFIG=$(ROOTDIR)/support/pkg-config.ffmpeg \
                ./configure --prefix=/ffmpeg --enable-static --disable-shared
    CONFIGURE_PI := CC="$(COMPILER) $(CFLAGS_PI)" $(CONFIGURE)
+CONFIGURE_FFMPEG := $(CONFIGURE)

# #####
# NASM
@@ -558,8 +560,14 @@ endif

ifeq (yes,$(CONFIG_NVENC))

-EXTLIBS += nvenc
-ENCODERS += h264_nvenc hevc_nvenc
+## YOU MUST INSTALL CUDA from NVIDIA for "libnpp"
+EXTLIBS += nvenc cuvid nonfree libnpp
+ENCODERS += h264_nvenc hevc_nvenc
+HWACCELS += h264_nvdec hevc_nvdec vp8_nvdec mjpeg_nvdec mpeg1_nvdec mpeg2_nvdec mpeg4_nvdec
+ECFLAGS += -I/usr/local/cuda/include
+
+## NVIDIA LIB for "libnpp"
+CONFIGURE_FFMPEG += --extra-ldflags=-L/usr/local/cuda/lib64

$(LIB_ROOT)/$(FFNVCODEC)/.tvh_download:
    $(call DOWNLOAD,$(FFNVCODEC_URL),$(LIB_ROOT)/$(FFNVCODEC_TB),$(FFNVCODEC_SHA1))
@@ -675,7 +683,7 @@ $(LIB_ROOT)/$(FFMPEG)/.tvh_build: \
    $(LIB_ROOT)/$(LIBOPUS)/.tvh_build \
    $(LIB_ROOT)/$(FFNVCODEC)/.tvh_build \
    $(LIB_ROOT)/$(FFMPEG)/.tvh_download
-    cd $(LIB_ROOT)/$(FFMPEG) && $(CONFIGURE) \
+    cd $(LIB_ROOT)/$(FFMPEG) && $(CONFIGURE_FFMPEG) \
        --disable-all \
        --enable-gpl \
        --enable-nonfree \
diff --git a/configure b/configure
index e76760479..f07e7a73e 100755
--- a/configure
+++ b/configure
@@ -600,6 +600,7 @@ if enabled ffmpeg_static; then
    #check_cc_header nvEncodeAPI || \

```

```
# die "NVENC library (https://developer.nvidia.com/nvidia-video-codec-sdk) not found"
enable nvenc
+ enable hwaccels
else
disable nvenc
fi
```

BEFORE (ORIGINAL)

```
cd /usr/src/tvheadend/build.linux/ffmpeg/ffmpeg-4.1.1 && FFMPEG_PREFIX=
/usr/src/tvheadend/build.linux/ffmpeg/build/ffmpeg PKG_CONFIG=
/usr/src/tvheadend/support/pkg-config.ffmpeg ./configure --prefix=/ffmpeg --enable-static --disable-shared \
--disable-all \
--enable-gpl \
--enable-nonfree \
--extra-cflags="-I/usr/src/tvheadend/build.linux/ffmpeg/build/ffmpeg/include -fPIE" \
--extra-libs="-L/usr/src/tvheadend/build.linux/ffmpeg/build/ffmpeg/lib -ldl" \
--pkg-config="/usr/src/tvheadend/support/pkg-config.ffmpeg" \
--enable-openssl \
--enable-avutil --enable-avcodec --enable-avformat --enable-swscale --enable-avresample
--enable-swresample --enable-avfilter \
--enable-libx264 --enable-libx265 --enable-libvpx --enable-libtheora --enable-libvorbis
--enable-libopus --enable-nvenc --enable-nonfree \
--enable-protocol=file --enable-protocol=http --enable-protocol=https --enable-protocol=hls
--enable-protocol=mmsh --enable-protocol=mmst --enable-protocol=rtmp --enable-protocol=rtmpe --enable-protocol
=rtmps --enable-protocol=rtmpt --enable-protocol=rtmpte --enable-protocol=rtmpts --enable-protocol=
ffrtmpcrypt --enable-protocol=ffrtmphttp --enable-protocol=rtp --enable-protocol=srtmp --enable-protocol=tcp
--enable-protocol=udp --enable-protocol=udplite --enable-protocol=unix --enable-protocol=crypto \
--enable-decoder=mpeg2video --enable-decoder=mp2 --enable-decoder=aac --enable-decoder=vorbis
--enable-decoder=ac3 --enable-decoder=eac3 --enable-decoder=aac_latm --enable-decoder=opus --enable-decoder=
h264 --enable-decoder=hevc --enable-decoder=theora --enable-decoder=flac --enable-decoder=libvorbis \
--enable-encoder=mpeg2video --enable-encoder=mp2 --enable-encoder=aac --enable-encoder=vorbis
--enable-encoder=flac --enable-encoder=libx264 --enable-encoder=libx265 --enable-encoder=libvpx_vp8
--enable-encoder=libvpx_vp9 --enable-encoder=libtheora --enable-encoder=libvorbis --enable-encoder=libopus
--enable-encoder=h264_nvenc --enable-encoder=hevc_nvenc \
--enable-demuxer=mpegts --enable-demuxer=matroska --enable-demuxer=hls --enable-demuxer=flv
--enable-demuxer=live_flv \
--enable-muxer=mpegts --enable-muxer=matroska --enable-muxer=mp4 --enable-muxer=ogg \
--enable-bsf=h264_mp4toannexb --enable-bsf=hevc_mp4toannexb \
--enable-filter=yadif --enable-filter=format --enable-filter=hwupload --enable-filter=hwdownload
--enable-filter=scale --enable-filter=null --enable-filter=aresample --enable-filter=anull \
\
--disable-programs \
--disable-doc \
--disable-htmldpages \
--disable-manpages \
--disable-podpages \
--disable-txtpages
```

External libraries:

iconv	libtheora	libvorbis	libvpx	libx26
4	libx265	libxcb	openssl	zlib
libopus				

External libraries providing hardware acceleration:

cuda	cuvid	ffnvcodec	nvdec	nvenc
	v4l2_m2m	vaapi		

Libraries:

avcodec	avfilter	avformat	avresample	avutil
	swresample	swscale		

Programs:

Enabled decoders:

aac	ac3	flac	hevc	mp2
	opus	theora	vorbis	vp3
aac_latm	eac3	h264	libvorbis	mpeg2v
ideo				

Enabled encoders:

aac	h264_nvenc	libopus	libvorbis	libvpx
_vp9	libx265	mp2	mpeg2video	vorbis
flac	hevc_nvenc	libtheora	libvpx_vp8	libx26

Enabled hwaccels:

Enabled parsers:

aac_latm ac3

Enabled demuxers:

flv hls live_flv matroska mpegts

Enabled muxers:

adts latm matroska mov mp4
mpegts ogg

Enabled protocols:

crypto file https rtmp rtmpt
rtmpts srtp tls udplite
ffrtmpecrypt hls mmsh rtmpe rtmpte
rtp tcp udp unix
ffrtmpehttp http mmst rtmps

Enabled filters:

anull aresample format hwdownload hwuplo
ad null scale yadif

Enabled bsfs:

h264_mp4toannexb hevc_mp4toannexb null

Enabled indevs:

Enabled outdevs:

License: nonfree and unredistributable

WARNING: Building with deprecated library libavresample

NOW (with this patch)

```
cd /usr/src/tvheadend/build.linux/ffmpeg/ffmpeg-4.1.1 && FFMPEG_PREFIX=
/usr/src/tvheadend/build.linux/ffmpeg/build/ffmpeg PKG_CONFIG=
/usr/src/tvheadend/support/pkg-config.ffmpeg ./configure --prefix=/ffmpeg --enable-static --disable-shared
--extra-ldflags=-L/usr/local/cuda/lib64 \
  --disable-all \
  --enable-gpl \
  --enable-nonfree \
  --extra-cflags=
"-I/usr/src/tvheadend/build.linux/ffmpeg/build/ffmpeg/include -fPIE -I/usr/local/cuda/include" \
  --extra-libs="-L/usr/src/tvheadend/build.linux/ffmpeg/build/ffmpeg/lib -ldl" \
  --pkg-config="/usr/src/tvheadend/support/pkg-config.ffmpeg" \
  --enable-openssl \
  --enable-avutil --enable-avcodec --enable-avformat --enable-swscale --enable-avresample
--enable-swresample --enable-avfilter \
  --enable-libx264 --enable-libx265 --enable-libvpx --enable-libtheora --enable-libvorbis
--enable-libopus --enable-nvenc --enable-cuvid --enable-nonfree --enable-libnpp --enable-nonfree \
  --enable-protocol=file --enable-protocol=http --enable-protocol=https --enable-protocol=hls
--enable-protocol=mmsh --enable-protocol=mmst --enable-protocol=rtmp --enable-protocol=rtmpe --enable-protocol
=rtmps --enable-protocol=rtmpt --enable-protocol=rtmpte --enable-protocol=rtmpts --enable-protocol=
ffrtmpecrypt --enable-protocol=ffrtmpehttp --enable-protocol=rtp --enable-protocol=srtp --enable-protocol=tcp
--enable-protocol=udp --enable-protocol=udplite --enable-protocol=unix --enable-protocol=crypto \
  --enable-decoder=mpeg2video --enable-decoder=mp2 --enable-decoder=aac --enable-decoder=vorbis
--enable-decoder=ac3 --enable-decoder=eac3 --enable-decoder=aac_latm --enable-decoder=opus --enable-decoder=
h264 --enable-decoder=hevc --enable-decoder=theora --enable-decoder=flac --enable-decoder=libvorbis \
  --enable-encoder=mpeg2video --enable-encoder=mp2 --enable-encoder=aac --enable-encoder=vorbis
--enable-encoder=flac --enable-encoder=libx264 --enable-encoder=libx265 --enable-encoder=libvpx_vp8
--enable-encoder=libvpx_vp9 --enable-encoder=libtheora --enable-encoder=libvorbis --enable-encoder=libopus
--enable-encoder=h264_nvenc --enable-encoder=hevc_nvenc \
  --enable-demuxer=mpegts --enable-demuxer=matroska --enable-demuxer=hls --enable-demuxer=flv
--enable-demuxer=live_flv \
  --enable-muxer=mpegts --enable-muxer=matroska --enable-muxer=mp4 --enable-muxer=ogg \
  --enable-bsf=h264_mp4toannexb --enable-bsf=hevc_mp4toannexb \
  --enable-filter=yadif --enable-filter=format --enable-filter=hwupload --enable-filter=hwdownload
--enable-filter=scale --enable-filter=null --enable-filter=aresample --enable-filter=anull \
  --enable-hwaccel=h264_nvdec --enable-hwaccel=hevc_nvdec --enable-hwaccel=vp8_nvdec --enable-hwaccel=
mjpeg_nvdec --enable-hwaccel=mpeg1_nvdec --enable-hwaccel=mpeg2_nvdec --enable-hwaccel=mpeg4_nvdec \
```

```
--disable-programs \  
--disable-doc \  
--disable-htmlpages \  
--disable-manpages \  
--disable-podpages \  
--disable-txtpages
```

External libraries:

```
iconv          libtheora          libvorbis          libvpx          libx264  
4              libx265          libxcb             openssl        zlib  
libopus
```

External libraries providing hardware acceleration:

```
cuda          cuvid          ffnvcodec          libnpp          nvdec  
              nvenc          v4l2_m2m          vaapi
```

Libraries:

```
avcodec          avfilter          avformat          avresample      avutil  
                swresample          swscale
```

Programs:

Enabled decoders:

```
aac              eac3              h263              hevc              mjpeg  
                mpeg1video          mpeg4              theora            vp3  
aac_latm         flac              h264              libvorbis         mp2  
                mpeg2video          opus              vorbis            vp8  
ac3
```

Enabled encoders:

```
aac              h264_nvenc        libopus           libvorbis         libvpx  
_vp9            libx265           mp2              mpeg2video        vorbis  
flac            hevc_nvenc        libtheora         libvpx_vp8        libx264  
4
```

Enabled hwaccels:

```
h264_nvdec      hevc_nvdec        mjpeg_nvdec      mpeg1_nvdec      mpeg2_  
nvdec           mpeg4_nvdec       vp8_nvdec
```

Enabled parsers:

```
aac_latm        ac3                h263              mpeg4video
```

Enabled demuxers:

```
flv             hls                live_flv          matroska          mpegts
```

Enabled muxers:

```
adts            latm              matroska          mov                mp4  
                mpegts           ogg
```

Enabled protocols:

```
crypto          file              https             rtmp              rtmpt  
                rtmps            srtplib          tls                udplite  
ffrtmptcrypt   hls              mmsh             rtmpe            rtmpte  
                rtp              tcp              udp                unix  
ffrtmpthttp    http             mmst             rtmps
```

Enabled filters:

```
anull           aresample        format            hwdownload        hwuplo  
ad              null              scale            yadif
```

Enabled bsfs:

```
h264_mp4toannexb  hevc_mp4toannexb  null
```

Enabled indevs:

Enabled outdevs:

License: nonfree and unredistributable

WARNING: Building with deprecated library libavresample

#9 - 2019-11-27 10:16 - saen acro

Ronny M. wrote:

And you must install NVIDIA CUDA for ext Libs. (see in coments)

Most any transcoding software recommend installing of [nVidia cuda-toolkit](#) package with, include video drivers also, Nvenc is part of CUDA

#10 - 2019-11-29 23:32 - Flole Systems

If I see that correctly most of the code needed is already part of ffmpeg's libavcodec/nvdec.c file, all that's left is integrating this file to tvheadend. So if someone wants to work on this, this might be a good starting point.

#11 - 2019-11-30 16:04 - saen acro

Flole Systems wrote:

If I see that correctly most of the code needed is already part of ffmpeg's libavcodec/nvdec.c file, all that's left is integrating this file to tvheadend. So if someone wants to work on this, this might be a good starting point.

not exactly need some of <https://github.com/FFmpeg/nv-codec-headers>

#12 - 2020-10-31 02:22 - Flole Systems

- Tracker changed from Bug to Feature
- Assignee deleted (John Törnblom)
- Found in version deleted (4.3)

#13 - 2020-11-02 07:45 - Christian Zippel

- File 01ffmpeg-static.patch added

Flole Systems wrote:

I've looked at the ffmpeg makefile that's part of tvheadend and it doesn't contain the required flags to compile for nvdec. I guess that's the first thing we need to fix, and then we can start to look at how to tell tvheadend to actually use it.

I dont think so.

With two changes in the makefile i become also the ffmpeg-programs.

With them i had a look in it.

Here the included encoders:

```
root@774f3eff9c20:/# ffmpeg -encoders | grep nvenc
ffmpeg version 4.3.1 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 10 (Ubuntu 10.2.0-5ubuntu1~20.04)
  configuration: --prefix=/ffmpeg --enable-static --disable-shared --enable-gpl --enable-nonfree --extra-cflags='-I/tmp/tvheadend/build.linux/ffmpeg/build/ffmpeg/include -fPIE' --extra-libs='-L/tmp/tvheadend/build.linux/ffmpeg/build/ffmpeg/lib -ldl' --pkg-config=/tmp/tvheadend/support/pkg-config.ffmpeg --enable-openssl --enable-avutil --enable-avcodec --enable-avformat --enable-swscale --enable-avresample --enable-swresample --enable-avfilter --enable-libx264 --enable-libx265 --enable-libvpx --enable-libtheora --enable-libvorbis --enable-libfdk-aac --enable-libopus --enable-nvenc --enable-vaapi --enable-nonfree --enable-protocol=file --enable-protocol=http --enable-protocol=https --enable-protocol=hls --enable-protocol=mms --enable-protocol=rtmp --enable-protocol=rtmpe --enable-protocol=rtmps --enable-protocol=rtmpt --enable-protocol=rtmpte --enable-protocol=rtmpts --enable-protocol=ffrtmpcrypt --enable-protocol=ffrtmphttp --enable-protocol=rtp --enable-protocol=srt --enable-protocol=tcp --enable-protocol=udp --enable-protocol=udplite --enable-protocol=unix --enable-protocol=crypto --enable-decoder=mpeg2video --enable-decoder=mp2 --enable-decoder=aac --enable-decoder=vorbis --enable-decoder=ac3 --enable-decoder=eac3 --enable-decoder=aac_latm --enable-decoder=opus --enable-decoder=h264 --enable-decoder=hevc --enable-decoder=theora --enable-decoder=flac --enable-decoder=libvorbis --enable-encoder=mpeg2video --enable-encoder=mp2 --enable-encoder=aac --enable-encoder=vorbis --enable-encoder=flac --enable-encoder=libx264 --enable-encoder=libx265 --enable-encoder=libvpx_vp8 --enable-encoder=libvpx_vp9 --enable-encoder=libtheora --enable-encoder=libvorbis --enable-encoder=libfdk_aac --enable-encoder=libopus --enable-encoder=h264_nvenc --enable-encoder=hevc_nvenc --enable-encoder=h264_vaapi --enable-encoder=hevc_vaapi --enable-encoder=vp8_vaapi --enable-encoder=vp9_vaapi --enable-demuxer=mpegts --enable-demuxer=matroska --enable-demuxer=hls --enable-demuxer=flv --enable-demuxer=live_flv --enable-muxer=mpegts --enable-muxer=matroska --enable-muxer=mp4 --enable-muxer=ogg --enable-bsf=h264_mp4toannexb --enable-bsf=hevc_mp4toannexb --enable-filter=yadif --enable-filter=format --enable-filter=hwupload --enable-filter=hwdownload --enable-filter=scale --enable-filter=null --enable-filter=aresample --enable-filter=anull --enable-filter=deinterlace_vaapi --enable-filter=scale_vaapi --enable-hwaccel=mpeg2_vaapi --enable-hwaccel=h264_vaapi --enable-hwaccel=hevc_vaapi --enable-hwaccel=vp9_vaapi --disable-doc --disable-htmlpages --disable-manpages --disable-podpages --disable-txtpages
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
```



```

libavdevice      58. 10.100 / 58. 10.100
libavfilter      7. 85.100 / 7. 85.100
libavresample    4.  0.  0 / 4.  0.  0
libswscale       5.  7.100 / 5.  7.100
libswresample    3.  7.100 / 3.  7.100
libpostproc     55.  7.100 / 55.  7.100
V..... h264_nvenc      NVIDIA NVENC H.264 encoder (codec h264)
V..... nvenc          NVIDIA NVENC H.264 encoder (codec h264)
V..... nvenc_h264     NVIDIA NVENC H.264 encoder (codec h264)
V..... nvenc_hevc     NVIDIA NVENC hevc encoder (codec hevc)
V..... hevc_nvenc     NVIDIA NVENC hevc encoder (codec hevc)

```

Here are the Decoders:

```

root@774f3eff9c20:/# ffmpeg -decoders | grep cuvid
ffmpeg version 4.3.1 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 10 (Ubuntu 10.2.0-5ubuntu1~20.04)
  configuration: --prefix=/ffmpeg --enable-static --disable-shared --enable-gpl --enable-nonfree --extra-cflags='-I/tmp/tvheadend/build.linux/ffmpeg/build/ffmpeg/include -fPIE' --extra-libs='-L/tmp/tvheadend/build.linux/ffmpeg/build/ffmpeg/lib -ldl' --pkg-config=/tmp/tvheadend/support/pkg-config.ffmpeg --enable-openssl --enable-avutil --enable-avcodec --enable-avformat --enable-swscale --enable-avresample --enable-swresample --enable-avfilter --enable-libx264 --enable-libx265 --enable-libvpx --enable-libtheora --enable-libvorbis --enable-libfdk-aac --enable-libopus --enable-nvenc --enable-vaapi --enable-nonfree --enable-protocol=file --enable-protocol=http --enable-protocol=https --enable-protocol=hls --enable-protocol=mms --enable-protocol=mmst --enable-protocol=rtmp --enable-protocol=rtmpe --enable-protocol=rtmps --enable-protocol=rtmpt --enable-protocol=rtmpte --enable-protocol=rtmpts --enable-protocol=ffrtmpcrypt --enable-protocol=ffrtmphttp --enable-protocol=rtp --enable-protocol=srt --enable-protocol=tcp --enable-protocol=udp --enable-protocol=udplite --enable-protocol=unix --enable-protocol=crypto --enable-decoder=mpeg2video --enable-decoder=mp2 --enable-decoder=aac --enable-decoder=vorbis --enable-decoder=ac3 --enable-decoder=eac3 --enable-decoder=aac_latm --enable-decoder=opus --enable-decoder=h264 --enable-decoder=hevc --enable-decoder=theora --enable-decoder=flac --enable-decoder=libvorbis --enable-encoder=mpeg2video --enable-encoder=mp2 --enable-encoder=aac --enable-encoder=vorbis --enable-encoder=flac --enable-encoder=libx264 --enable-encoder=libx265 --enable-encoder=libvpx_vp8 --enable-encoder=libvpx_vp9 --enable-encoder=libtheora --enable-encoder=libvorbis --enable-encoder=libfdk_aac --enable-encoder=libopus --enable-encoder=h264_nvenc --enable-encoder=hevc_nvenc --enable-encoder=h264_vaapi --enable-encoder=hevc_vaapi --enable-encoder=vp8_vaapi --enable-encoder=vp9_vaapi --enable-demuxer=mpegts --enable-demuxer=matroska --enable-demuxer=hls --enable-demuxer=flv --enable-demuxer=live_flv --enable-muxer=mpegts --enable-muxer=matroska --enable-muxer=mp4 --enable-muxer=ogg --enable-bsf=h264_mp4toannexb --enable-bsf=hevc_mp4toannexb --enable-filter=yadif --enable-filter=format --enable-filter=hwupload --enable-filter=hwdownload --enable-filter=scale --enable-filter=null --enable-filter=aresample --enable-filter=anull --enable-filter=deinterlace_vaapi --enable-filter=scale_vaapi --enable-hwaccel=mpeg2_vaapi --enable-hwaccel=h264_vaapi --enable-hwaccel=hevc_vaapi --enable-hwaccel=vp9_vaapi --disable-doc --disable-htmllpages --disable-manpages --disable-podpages --disable-txtpages
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter    7. 85.100 / 7. 85.100
  libavresample  4.  0.  0 / 4.  0.  0
  libswscale     5.  7.100 / 5.  7.100
  libswresample  3.  7.100 / 3.  7.100
  libpostproc   55.  7.100 / 55.  7.100
V..... h264_cuvid      Nvidia CUVID H264 decoder (codec h264)
V..... hevc_cuvid      Nvidia CUVID HEVC decoder (codec hevc)
V..... mjpeg_cuvid     Nvidia CUVID MJPEG decoder (codec mjpeg)
V..... mpeg1_cuvid     Nvidia CUVID MPEG1VIDEO decoder (codec mpeg1video)
V..... mpeg2_cuvid     Nvidia CUVID MPEG2VIDEO decoder (codec mpeg2video)
V..... mpeg4_cuvid     Nvidia CUVID MPEG4 decoder (codec mpeg4)
V..... vc1_cuvid       Nvidia CUVID VC1 decoder (codec vc1)
V..... vp8_cuvid       Nvidia CUVID VP8 decoder (codec vp8)
V..... vp9_cuvid       Nvidia CUVID VP9 decoder (codec vp9)

```

As you see, all decoders are included.

And here:

```

root@774f3eff9c20:/# ffmpeg -hwaccels
ffmpeg version 4.3.1 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 10 (Ubuntu 10.2.0-5ubuntu1~20.04)
  configuration: --prefix=/ffmpeg --enable-static --disable-shared --enable-gpl --enable-nonfree --extra-cflags='-I/tmp/tvheadend/build.linux/ffmpeg/build/ffmpeg/include -fPIE' --extra-libs='-L/tmp/tvheadend/build.linux/ffmpeg/build/ffmpeg/lib -ldl' --pkg-config=/tmp/tvheadend/support/pkg-config.ffmpeg --enable-openssl --enable-avutil --enable-avcodec --enable-avformat --enable-swscale --enable-avresample --enable-swresample --enable-avfilter --enable-libx264 --enable-libx265 --enable-libvpx --enable-libtheora --enable-libvorbis --enable-libfdk-aac --enable-libopus --enable-nvenc --enable-vaapi --enable-nonfree --enable-protocol=file --enable-protocol=

```

```

http --enable-protocol=https --enable-protocol=hls --enable-protocol=mmsh --enable-protocol=mmst --enable-prot
ocol=rtmp --enable-protocol=rtmpe --enable-protocol=rtmps --enable-protocol=rtmpt --enable-protocol=rtmpte --e
nable-protocol=rtmpts --enable-protocol=ffrtmpcrypt --enable-protocol=ffrtmphttp --enable-protocol=rtp --enabl
e-protocol=srtsp --enable-protocol=tcp --enable-protocol=udp --enable-protocol=udplite --enable-protocol=unix -
--enable-protocol=crypto --enable-decoder=mpeg2video --enable-decoder=mp2 --enable-decoder=aac --enable-decoder
=vorbis --enable-decoder=ac3 --enable-decoder=eac3 --enable-decoder=aac_latm --enable-decoder=opus --enable-de
coder=h264 --enable-decoder=hevc --enable-decoder=theora --enable-decoder=flac --enable-decoder=libvorbis --en
able-encoder=mpeg2video --enable-encoder=mp2 --enable-encoder=aac --enable-encoder=vorbis --enable-encoder=fla
c --enable-encoder=libx264 --enable-encoder=libx265 --enable-encoder=libvpx_vp8 --enable-encoder=libvpx_vp9 --
enable-encoder=libtheora --enable-encoder=libvorbis --enable-encoder=libfdk_aac --enable-encoder=libopus --ena
ble-encoder=h264_nvenc --enable-encoder=hevc_nvenc --enable-encoder=h264_vaapi --enable-encoder=hevc_vaapi --e
nable-encoder=vp8_vaapi --enable-encoder=vp9_vaapi --enable-demuxer=mpegts --enable-demuxer=matroska --enable-
demuxer=hls --enable-demuxer=flv --enable-demuxer=live_flv --enable-muxer=mpegts --enable-muxer=matroska --ena
ble-muxer=mp4 --enable-muxer=ogg --enable-bsf=h264_mp4toannexb --enable-bsf=hevc_mp4toannexb --enable-filter=y
adif --enable-filter=format --enable-filter=hwupload --enable-filter=hwdownload --enable-filter=scale --enable
-filter=null --enable-filter=aresample --enable-filter=anull --enable-filter=deinterlace_vaapi --enable-filter
=scale_vaapi --enable-hwaccel=mpeg2_vaapi --enable-hwaccel=h264_vaapi --enable-hwaccel=hevc_vaapi --enable-hwa
ccel=vp9_vaapi --disable-doc --disable-htmllpages --disable-manpages --disable-podpages --disable-txtpages
libavutil      56. 51.100 / 56. 51.100
libavcodec     58. 91.100 / 58. 91.100
libavformat    58. 45.100 / 58. 45.100
libavdevice    58. 10.100 / 58. 10.100
libavfilter     7. 85.100 /  7. 85.100
libavresample  4.  0.  0 /  4.  0.  0
libswscale     5.  7.100 /  5.  7.100
libswresample  3.  7.100 /  3.  7.100
libpostproc   55.  7.100 / 55.  7.100

```

Hardware acceleration methods:

```

cuda
vaapi

```

So HW Accel is also in for cuda.

```

root@774f3eff9c20:/# ffmpeg -filters | grep cuda
ffmpeg version 4.3.1 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 10 (Ubuntu 10.2.0-5ubuntu1~20.04)
  configuration: --prefix=/ffmpeg --enable-static --disable-shared --enable-gpl --enable-nonfree --extra-cflag
s='-I/tmp/tvheadend/build.linux/ffmpeg/build/ffmpeg/include -fPIE' --extra-libs='-L/tmp/tvheadend/build.linux/
ffmpeg/build/ffmpeg/lib -ldl' --pkg-config=/tmp/tvheadend/support/pkg-config.ffmpeg --enable-openssl --enable-
avutil --enable-avcodec --enable-avformat --enable-swscale --enable-avresample --enable-swresample --enable-av
filter --enable-libx264 --enable-libx265 --enable-libvpx --enable-libtheora --enable-libvorbis --enable-libfdk
-aac --enable-libopus --enable-nvenc --enable-vaapi --enable-nonfree --enable-protocol=file --enable-protocol=
http --enable-protocol=https --enable-protocol=hls --enable-protocol=mmsh --enable-protocol=mmst --enable-prot
ocol=rtmp --enable-protocol=rtmpe --enable-protocol=rtmps --enable-protocol=rtmpt --enable-protocol=rtmpte --e
nable-protocol=rtmpts --enable-protocol=ffrtmpcrypt --enable-protocol=ffrtmphttp --enable-protocol=rtp --enabl
e-protocol=srtsp --enable-protocol=tcp --enable-protocol=udp --enable-protocol=udplite --enable-protocol=unix -
--enable-protocol=crypto --enable-decoder=mpeg2video --enable-decoder=mp2 --enable-decoder=aac --enable-decoder
=vorbis --enable-decoder=ac3 --enable-decoder=eac3 --enable-decoder=aac_latm --enable-decoder=opus --enable-de
coder=h264 --enable-decoder=hevc --enable-decoder=theora --enable-decoder=flac --enable-decoder=libvorbis --en
able-encoder=mpeg2video --enable-encoder=mp2 --enable-encoder=aac --enable-encoder=vorbis --enable-encoder=fla
c --enable-encoder=libx264 --enable-encoder=libx265 --enable-encoder=libvpx_vp8 --enable-encoder=libvpx_vp9 --
enable-encoder=libtheora --enable-encoder=libvorbis --enable-encoder=libfdk_aac --enable-encoder=libopus --ena
ble-encoder=h264_nvenc --enable-encoder=hevc_nvenc --enable-encoder=h264_vaapi --enable-encoder=hevc_vaapi --e
nable-encoder=vp8_vaapi --enable-encoder=vp9_vaapi --enable-demuxer=mpegts --enable-demuxer=matroska --enable-
demuxer=hls --enable-demuxer=flv --enable-demuxer=live_flv --enable-muxer=mpegts --enable-muxer=matroska --ena
ble-muxer=mp4 --enable-muxer=ogg --enable-bsf=h264_mp4toannexb --enable-bsf=hevc_mp4toannexb --enable-filter=y
adif --enable-filter=format --enable-filter=hwupload --enable-filter=hwdownload --enable-filter=scale --enable
-filter=null --enable-filter=aresample --enable-filter=anull --enable-filter=deinterlace_vaapi --enable-filter
=scale_vaapi --enable-hwaccel=mpeg2_vaapi --enable-hwaccel=h264_vaapi --enable-hwaccel=hevc_vaapi --enable-hwa
ccel=vp9_vaapi --disable-doc --disable-htmllpages --disable-manpages --disable-podpages --disable-txtpages
libavutil      56. 51.100 / 56. 51.100
libavcodec     58. 91.100 / 58. 91.100
libavformat    58. 45.100 / 58. 45.100
libavdevice    58. 10.100 / 58. 10.100
libavfilter     7. 85.100 /  7. 85.100
libavresample  4.  0.  0 /  4.  0.  0
libswscale     5.  7.100 /  5.  7.100
libswresample  3.  7.100 /  3.  7.100
libpostproc   55.  7.100 / 55.  7.100
... hwupload_cuda      V->V      Upload a system memory frame to a CUDA device.
... overlay_cuda      VV->VV     Overlay one video on top of another using CUDA
... scale_cuda        V->V      GPU accelerated video resizer
... thumbnail_cuda    V->V      Select the most representative frame in a given sequence of consecutive fram
es.
T.. yadif_cuda        V->V      Deinterlace CUDA frames

```

Filters are also in - so no need for extra scaler.

So we need "only" the nvdec part in tvheadend. ;-)

I include here my patch for the ffmpeg BINs for testing.

#14 - 2020-11-02 12:02 - Flole Systems

There's no point in including the encoders if they are never used. The patch above also includes them.

#15 - 2021-01-10 21:02 - Ronny M.

Fole Systems wrote:

There's no point in including the encoders if they are never used. The patch above also includes them.

Hi Flole,
nice now it is at least not a bug anymore, but a feature.... ;-)

That's right and exactly my problem, I understand everything else that is still necessary here, but unfortunately I don't have programming experience here, but if I can help you / or others with testing and reports, I'm happy to be on board!


because knowing always only harms those who don't have it. ;-)

#16 - 2021-01-11 00:04 - Flole Systems

Interestingly enough I am having issues transcoding using nvenc for some reason, so I need to figure out why that isn't working for me at the moment Otherwise even if I wanted and had time I couldn't start working on that. Seems like I'm the only one with that issue though for some reason....

#17 - 2021-01-30 06:13 - Ronny M.


Fole Systems wrote:

Interestingly enough I am having issues transcoding using nvenc for some reason, so I need to figure out why that isn't working for me at the moment  Otherwise even if I wanted and had time I couldn't start working on that. Seems like I'm the only one with that issue though for some reason....

[Fole Systems](#)

Please describe your Problem...
with NVENC have there are always many times many problems....

Come you from Germany?
Then we can in a simple chat or call via Telephone initiate ;-) and all the Problems inspect and to discuss..!?!?

and not one question or comment and 20 Days Later a Answer.... 

#18 - 2021-02-20 22:43 - Flole Systems

- Status changed from New to Accepted
- Target version set to 4.4

I figured out the issue with nvenc (and also have a nvdec working for h264 and hevc). I just need to find a fix for that nvenc issue.

#19 - 2021-05-05 16:49 - Flole Systems

- Status changed from Accepted to Need feedback

PR is opened, please test and see if it works.

#20 - 2021-05-16 14:33 - Ronny M.

Previous tests look good so far, but the CPU load is still high despite the GPU now doing the decoding. My guess is that this is due to the missing HW upload / download of the filter.

#21 - 2021-07-07 15:58 - Flole Systems

- Status changed from Need feedback to Fixed

Files

01ffmpeg-static.patch	1.03 KB	2020-11-02	Christian Zippel
-----------------------	---------	------------	------------------