Direct SDI GPU Communication

Stefan DESSENS

June 6, 2013 — VidiGo

Abstract

In dit document wordt een korte samenvatting gegeven van het nut van AMD SDI-Link en NVIDIA GPUDirect. Hiervoor is een plugin voor DAVE ontwikkeld op basis van de VideoMasterHD SDK van DeltaCast, de testresultaten met deze plugin worden eveneens besproken.

1 oude situatie

In de oude situtatie wordt de data als volgt overgezet:

1. De data komt binnen via de SDI kaart.
2. De data wordt overgezet van de SDI kaart naar het systeemgeheugen.
3. De Data ondergaat een sysmem-naar-sysmem copy om deze naar een de address space van DAVE te verplaatsen (DaveMemoryPool).
4. De data ondergaat een sysmem-naar-sysmem copy om deze te verplaatsen naar de OpenGL driver.
5. De data wordt overgezet van het systeemgeheugen naar de GPU.

Bij SDI output gebeurt hetzelfde, maar dan in omgekeerde volgorde.

2 AMD SDI-Link

AMD SDI-Link is een techniek waarbij beelden worden overgezet van de SDI kaart naar een AMD GPU. Dat kan op 2 manieren gebeuren:

1. Via P2P transfers (direct over de PCI bus).
2. Via het systeemgeheugen, hierbij wordt de sysmem-naar-sysmem copy overgeslagen.

DeltaCast ondersteunt alleen de 2e optie. De fabrikanten DataPath, Bluefish en AJA ondersteunen ook P2P transfers. Momenteel is de enige videokaart die AMD SDI-Link ondersteunt de V7900.

3 NVIDIA GPUDirect

NVIDIA GPUDirect is een techniek waarbij, onder andere, videobeelden kunnen worden overgezet tussen SDI kaarten en NVIDIA GPU's. Dat kan op dezelfde 2 manieren als bij AMD gebeuren.

1. Via P2P transfers (direct over de PCI bus).
2. Via het systeemgeheugen, hierbij wordt de sysmem-naar-sysmem copy overgeslagen.

P2P transfers worden alleen gesupport door de SDI kaarten van NVIDIA zelf. Deze zijn significant duurder dan third-party kaarten.

De videokaarten Quadro 4000, 5000, K4000, K5000 plus enkele oudere kaarten worden ondersteund door NVIDIA GPUDirect.

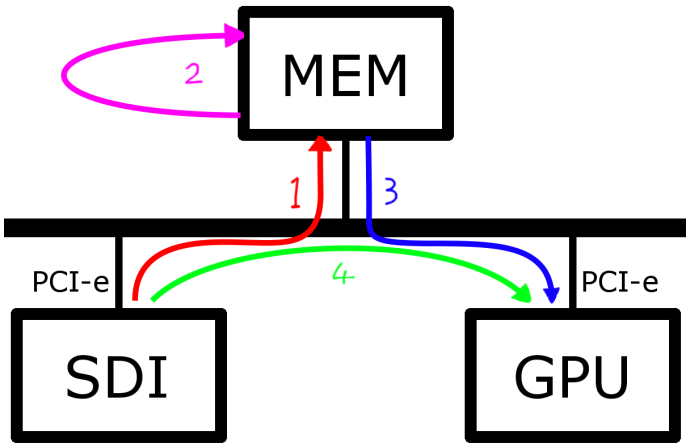

Figure 1: Deze afbeelding weergeeft de paden welke worden afgelegd door de videodata.

Bij de oude situatie legt de data hetvolgende pad af	$1 \rightarrow 2 \rightarrow 2 \rightarrow 3$
bij SDI-Link of GPUDirect zonder P2P	$1 \rightarrow 3$
Bij P2P	4

Stap 1 en 3 worden gelimiteerd door de PCI-e bandwidth ($6 - 12GB/s$) én de memory bandwidth ($20 - 40GB/s$).
Stap 2 wordt alleeen gelimiteerd door de memory bandwidth.
Stap 4 wordt alleen gelimiteerd door de PCI-e bandwidth

4 Testresultaten

De volgende hardware is in deze test gebruikt:

Processor	i7-3960X
RAM	8GB DDR3-1666 (2x4GB)
Videokaart	AMD FirePro V7900 (PCI-e 2.0 x16) HP NVIDIA Quadro 4000 (PCI-e 2.0 x16) Gainward NVIDIA GTX 680 (PCI-e 3.0 x16)

Voor een opsomming van de specificaties van de grafische kaarten, zie appendix A.

4.1 Input

Er worden verschillende hoeveelheden input streams getest. Het formaat is altijd 1080i50. De tabel geeft aan of de test wel of niet is geslaagd, dat wordt gekwantificeerd aan de hand van de hoeveelheid dropped frames. Geen dropped frames betekent een geslaagde test. Bij sommige testen kon er niet met zekerheid worden vastgesteld of de hoeveelheid streams vloeiend kan worden aangestuud vanwege een fluctuerende framerate, dit wordt geindiceerd met 'soms'.

	AMD V7900			Quadro 4000			GTX 680		
aantal streams	oude plugin	nieuwe plugin	nieuwe plugin[1]	oude plugin	nieuwe plugin	nieuwe plugin[1]	oude plugin	nieuwe plugin[2]	nieuwe plugin[2]
1	✓	✓	✓	✓	✓	✓	✓	N/A	N/A
2	✓	✓	✓	✓	✓	✓	✓	N/A	N/A
3	✓	✓	✓	✓	✓	✓	✓	N/A	N/A
4	✓	✓	✓	✓	✓	✓	✓	N/A	N/A
5	✓	✓	✓	✓	✓	✓	✓	N/A	N/A
6	✓	soms	✓	✓	✓	✓	✓	N/A	N/A
7	✓	soms	✓	✓	✓	✓	✓	N/A	N/A
8	✓	×	×	✓	×	×	✓	N/A	N/A

De maximale hoeveelheid streams neemt bij deze videokaarten iets af wanneer er gebruik wordt gemaakt van de nieuwe plugin. De afname is bij AMD erger dan bij NVIDIA, alhoewel de gemiddelde framerate bij AMD hoger ligt. De framerate fluceert erg bij de V7900.

Zowel de V7900 als de Quadro 4000 zijn vermoeddelijk niet in staat om meer dan 8 streams aan te sturen. Binnen de testopstelling kon de limiet van de GTX 680

[1] Zonder color-space conversie (RGB → YUV)
[2] Wordt niet ondersteunt

niet worden bereikt, deze gaf met 8 input streams een framerate van 190 fps[3] en verbruikte 8 GB/s aan bandwidth. Er dient hierbij wel vermeld te worden dat de framerate niet lineair schaalt. De oplossingen met AMD SDI-Link en NVIDIA GPUDirect gebruikten ongeveer 2.2 GB/s bandwidth.

4.2 output

Verschillende SDI-outputs zijn op dezelfde manier gemeten, met verschillende hoeveelheden output streams. Het formaat was telkens 1080i50.

	AMD V7900			Quadro 4000			GTX 680		
aantal streams	oude plugin	nieuwe plugin	nieuwe plugin[4]	oude plugin	nieuwe plugin	nieuwe plugin[4]	oude plugin	nieuwe plugin[5]	nieuwe plugin[5]
1	✓	✓	✓	✓	✓	✓	✓	N/A	N/A
2	✓	✓	✓	✓	✓	✓	✓	N/A	N/A
3	✓	✓	✓	✓	✓	✓	✓	N/A	N/A
4	×	✓	✓	✓	✓	✓	✓	N/A	N/A
5	×	✓	✓	×	✓	✓	✓	N/A	N/A
6	×	✓	✓	×	✓	✓	✓	N/A	N/A
7	×	✓	✓	×	×	✓	✓	N/A	N/A
8	×	×	×	×	×	✓	✓	N/A	N/A

Bij output streams biedt SDI-Link / GPUDirect wel voordeel. Bij AMD kunnen 133% meer streams worden aangestuurd ten opzichte van dezelfde kaart zonder SDI-Link, bij NVIDIA 50 tot 100%. Dat de framerate verder stijgt bij NVIDIA bij het uitschakelen van de color space conversie betekent vermoeddelijk dat de grafische kaart te zwak is. Bij 8 output streams behaalde de GTX 680 een framerate van 40 fps (onder de 25 treden er dropped frames op).

[3] Aldus GDEBugger
[4] Zonder color-space conversie (RGB → YUV)
[5] Wordt niet ondersteunt

5 Bandwidth

Bij 8 streams tegelijk werd een memory bandwidth gemeten van ongeveer $2.2GB/s$ met de nieuwe plugin, en $8GB/s$ met de oude plugin. Dit wijkt erg af van de theoretische bandwidth. De theoretische bandwidth wordt als volgt berekend:

$$Bytes/s = x * y * (\frac{bitsperpixel}{8}) * frames/sec \ * copy's$$

Merk op dat bij interlaced video de data slechts met de helft van de frequentie wordt versuurd.

Met GPUDirect of SDI-Link zijn 2 buffer-copy's benodigd, in alle andere gevallen zijn dat er 4. Dat levert de volgende theoretische en praktische data op:

	1 stream (theorie)	8 streams (theorie)	8 streams (praktijk)
oud	$0.41GB/s$	$3.32GB/s$	$8GB/s$
nieuw	$0.21GB/s$	$1.66GB/s$	$2.2GB/s$

Met de oude plugin bestaat er een flink verschil tussen theorie en praktijk. Het is niet duidelijk wat hier de oorzaak van is.

Het Intel X79 platform (Sandy-Brige E) is momenteel het platform met de hoogste memory bandwidth beschikbaar, en heeft een praktische memory bandwidth van $37GB/s$ in quad-channel configuratie[6]. In het testsysteem zijn met de GTX 680 tot 16 1080i50 streams getest in met een dual-channel DDR3-1600 memory configuratie, dat leverde geen problemen op. Quad-channel memory levert ongeveer 80% meer bandwidth op dan dual-channel[7]. Theoretisch is het dus mogelijk om minimaal $16 + 80\% \approx 29$ streams aan te sturen voordat de memory bandiwdth een bottlenck wordt.

Een andere mogelijke bandwidth bottleneck is de PCI-e bus. De PCI-e bus wordt gebruikt om data van en naar de grafische en SDI kaart te sturen. Elk beeld moet tweemaal over de PCI-e bus verstuurd worden, om de bytes van de SDI kaart naar de GPU te verplaatsen of andersom. PCI-e 2.0 heeft een praktische bandwidth van $6GB/s$. Dit is voldoende voor $\frac{6GB/s}{0.21GB/s} = 28$ streams. PCI-e 3.0 verdubbelt de bandwidth van PCI-e 2.0, en is vanaf midden 2013 breed beschikbaar. Het is niet mogelijk om de daadwerkelijk gebruikte PCI-e bandwith te meten.

[6] http://www.anandtech.com/show/5091/intel-core-i7-3960x-sandy-bridge-e-review-keeping-the-high-end-alive/4

[7] http://www.legitreviews.com/article/1779/3/

6 latency

De latency is tijdens dit project niet getest, maar op basis van het pad wat de data doorloopt kan een inschatting gemaakt worden van de verwachte latency winst. Bij de berekeningen wordt gebruik gemaakt van de volgende gegevens:

soort transfer	bandwidth	tijd bij één 1080i50 frame
PCI-e 2.0 16x $\leftrightarrow$ mem	$6GB/s$	$0.70ms$
mem $\leftrightarrow$ mem (4 thread)	$37GB/s$	$0.11ms$
mem $\leftrightarrow$ mem (1 threads)	$10GB/s$	$0.40ms$
mem $\leftrightarrow$ PCI-e 3.0 16x	$12GB/s$	$0.35ms$

De memory bandwidth moet gedeeld worden als er meer dan een transfer tegelijk plaats vind. Bij de volgende rekenvoorbeelden wordt er uit gegaan van een situatie met 4 input streams en 4 output streams, tevens wordt er uit gegaan dat er altijd optimale transfers en chunking plaats vind. Chunking houdt in dat er er al met een read wordt begonnen als de data slechts gedeeltelijk geschreven is, dit wordt echter niet overal gebruikt.

Uit testen blijkt dat de NVIDIA OpenGL implementatie slechts één thread gebruikt bij het kopieren van data, zoals wanneer de data wordt doorgegeven vanuit de pool naar de OpenGL driver. Maar dat er in die situatie wel chunking wordt gebruikt.

Oude situatie, input. Bij output gebeurt hetzelfde in omgekeerde volgorde.

SDI	$\rightarrow$	DeltaCast drv.	$4 * 0.70ms$	
DeltaCast drv.	$\rightarrow$	pool (1 thread)	$1 * 0.40ms$	overlapt met voorgaande
pool	$\rightarrow$	OpenGL drv.	$4 * 0.40ms$	
OpenGL drv.	$\rightarrow$	GPU	$0 * 0.35ms$	chunking
		totaal	$4.80ms$	

De totale latency bij in $\rightarrow$ output is tweemaal de input latency, ofwel $9.60ms$.

Nieuwe situatie, input. Bij output gebeurt hetzelfde in omgekeerde volgorde.

SDI	$\rightarrow$	DeltaCast drv.	$4 * 0.70ms$	
DeltaCast drv.	$\rightarrow$	GPU	$0 * 0.35ms$	chunking
		totaal	$2.80ms$	

De output naar de GPU neemt geen tijd in beslag omdat de transfer kan overlappen met input vanuit de SDI kaart. De genoemde bandwidth voor PCI-e betreft een per-device limiet. De PCI-e bus bevat genoeg bandbreedte om meerdere kaarten tegelijk met de maximale snelheid aan te sturen.

De totale latency bij de nieuwe situatie in $\rightarrow$ output is dus $5.60ms$. Het verschil tussen de oude en de nieuwe methode is $9.60 - 5.60 = 4.00ms$.
De bovenstaande berekening is echter niet volledig correct, zo worden er vaak SDI kaarten gebruikt met PCI-e 8x in plaats van 16x, en is het onreallistisch dat er geen tijd verloren gaat door o.a. latency en chunking. Ten slotte gaat er

ook een aanzienlijke tijd 'verloren' vanwege de bewerkingen en de latency naar de GPU.
Deze factoren zijn echter voor zowel de oude situatie als SDI-Link / GPUDirect hetzelfde. Het is daarom niet mogelijk om te concluderen wat de *totale* latency is. Het verschil tussen de oude en de nieuwe methode is echter wel zinvolle informatie.
Wanneer er meer streams worden gebruikt neemt het verschil in latency ongeveer rechtevenredig toe.

7 conclusie

De volgende conclusies kunnen worden getrokken:

1. Zowel AMD SDI-Link als NVIDIA GPUDirect laten een lichte achteruitgang zien in de hoeveelheid aanstuurbare input streams. Het zou geen probleem moeten zijn om hiervoor de oude plugin te blijven gebruiken.
2. Bij AMD stijgt het aantal aanstuurbare output streams met 133%, bij NVIDIA tussen de 50 en 100% indien gebruik wordt gemaakt van respectievelijk SDI-Link of GPUDirect.
3. Het is zinloos om de AMD FirePro V7900 of Quadro 4000 in te zetten, deze kaarten kunnen niet concurreren met de GTX 680 op prijs noch prestaties. Ondanks dat deze geen GPUDirect ondersteunt.
4. De memory bandwidth is bij reallistische hoeveelheden 1080i50 streams nooit de bottleneck. Het gebruik van GPUDirect of SDI-Link verlaagt de bandwidth requirements met een factor 3 a 4. Mogelijk is deze winst wel interessant bij 4K video.
5. SDI-Link of GPUDirect leveren een verwachte latency winst op van $4.00ms$ bij 8 streams.

De V7900 is de enige videokaart die AMD aanbied met SDI-Link ondersteuning. Al eerder is de conclusie getrokken dat de AMD FirePro V7900 te traag is, met het huidige aanbod kaarten is AMD SDI-link dus zinloos.

NVIDIA bied onder ander de K5000 aan. Deze is gebaseerd op dezelfde architectuur als de GTX 680, maar bevat ongeveer 30% lagere kloksnelheden en is derhalve *tot* 30% trager. De verwachting is dat met deze kaart een stuk meer output streams aangestuurd kan worden dan met de GTX 680, hoeveel dat precies is is zonder de hardware in huis te hebben lastig te zeggen. Ik verwacht dat de K5000 ongeveer 50% meer output streams kan draaien dan de GTX 680.

8 Voor de devvers

In de implementatie zelf zitten nog wat bugjes. Aan de kant van DeltaCast zijn aanwezig:

1. De implementatie van AMD werkt alleen waneer er geen OpenGL context wordt geladen. (DaveContext())
2. In of output format switchen lijkt niet te werken met resoluties $< 720p/i$.
3. Input format switchen levert soms een crash op binnen een bepaalde timing window. Vaak als je meerdere malen, heel snel achter elkaar, de SDI kabel er in- of uitsteekt.
4. Na switchen van output format geeft de BlackMagic Smartview HD geen beeld meer totdat de stream compleet opnieuw wordt opgestart.
    - De BlackMagic Smartview HD is een 17.3 monitor met een SDI in- en output. VidiGo gebruikt een aantal van deze schermen voor testdoeleinden.
5. Als de computer in sleep gaat terwijl er streams draaien, werkt de stream vaak niet meer tot de volgende reboot.

Aan de kant van de plugin zelf werken de volgende dingen nog niet naar behoren of zijn incorrect geimplementeerd:

1. Het geluid werkt nog niet. Dat zou opgelost moeten kunnen worden door een 2e stream te openen specifiek voor het geluid. (Vergeet niet beide streams te stoppen bij het wisselen van format of pauzeren).
2. Er gaat iets fout bij de color space conversie wat ervoor zorgt dat de kleuren rood en blauw worden omgewisseld (RGB $\rightarrow$ BGR).

A Specificaties

	AMD V7900[8]	Quadro 4000[9]	GTX 680[10]
TDP	$<$ 150W	142 W	195 W
Memory bandwidth	160 GB/s	89.6 GB/s	192.2 GB/s
pixel fill rate	23.2 Gpixels/s	15.2 Gpixels/s	33.86 Gpixels/s
Texture fill rate	58 Gtexels/s	15.2 Gtexels/s	135.42 Gtexels/s
Single precision	1856 FLOPS	486.4 GFLOPS	3250.18 GFLOPS
3Dmark11 GPU[11]	3380	2080	10650

[8] `http://www.gpuzoo.com/GPU-AMD/FirePro_3D_V7900.html`
[9] `http://www.gpuzoo.com/GPU-NVIDIA/Quadro_4000.html`
[10] `http://www.gpuzoo.com/GPU-Gainward/GeForce_GTX_680.html`
[11] `http://community.futuremark.com/hardware/gpu`