Intel I7/Nvidia vs AMD Ryzen/Radeon: Selected Workloads

This executive summary compares the performance of selected workloads on AMD's Ryzen 7 1700 and Radeon RX560 with Intel's I7-7700 and Nvidia GTX1050.

Profiling GPU/CPU Performance in Windows, RHEL, OSX, iOS & Android - http://davidjyoung.com/cmg/nvidiagpuperf.pdf

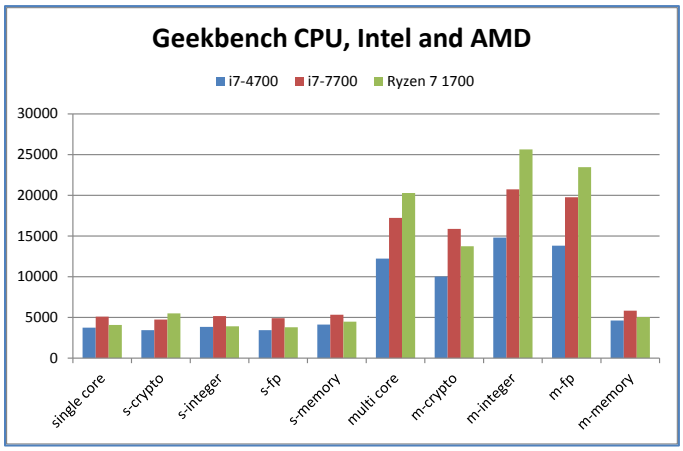

Figure 1 – Geekbench CPU Comparisons

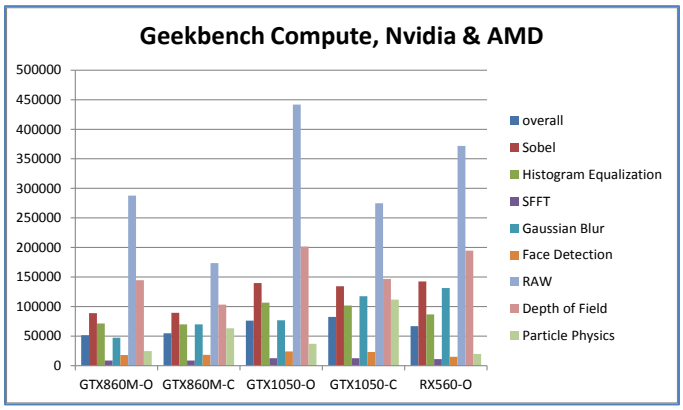

Figure 2 – Geekbench Compute Comparisons

platform	ipad	droid turbo	mac mini	asus luggable	asus luggable	asus luggable	asus desktop	asus desktop	asus desktop	lenovo desktop
CPU model	A7	snapdragon 805	I7-3615QM	I7-4700	I7-4700	I7-4700	I7-7700	I7-7700	I7-7700	Ryzen 7 1700
OS	IOS 10.3.2	Android 6.0.1	OSX 10.12.5	Win 8.1	Win 8.1	Win 8.1	Win 10	Win 10	Win 10	Win 10
CPU single core	1297	1061	3368	3620	3620	3620	5091	5091	5091	4069
CPU multiple core	2198	2959	10632	11722	11722	11722	17224	17224	17224	20281
GPU vendor	apple	qualcomm	intel	intel	nvidia	nvidia	intel	nvidia	nvidia	AMD
GPU model	a7gpu	adreno 420	4000	4600	GTX860M	GTX860M	HD630	GTX1050	GTX1050	Radeon RX 560
geekbench workload	OpenCL	Opencl	OpenCL	OpenCL	OpenCL	CUDA	OpenCL	OpenCL	CUDA	OpenCL
overall result	543	3860	6167	13809	45547	46903	22639	76263	82509	66849
Sobel	317	3412	3056	17219	92620	85611	29143	139765	134272	142531
Gpixels/sec	0.014	0.15	0.13	0.75	4.08	3.77	1.28	6.16	5.92	6.82
Histogram Equalization	326	4901	2409	13699	68766	54343	26091	106619	101866	86690
Gpixels/sec	0.01	0.153	0.075	0.43	2.15	1.7	0.82	3.33	3.18	2.71
SFFT	36	743	2324	2520	8658	8689	3664	12589	12763	11158
Gflops	0.09	1.85	5.8	6.28	21.6	21.7	9.13	31.4	31.8	27.8
Gaussian Blur	249	13391	4271	15721	42874	64346	35318	76858	117422	131492
Gpixels/sec	0.004	0.234	0.074	0.275	0.751	1.13	618.7	1.35	2.06	2.3
Face Detection	649	2820	2247	8020	10988	13937	13888	24030	23248	15165
Msubwindows/sec	0.189	0.823	0.656	2.34	3.21	4.07	4.06	7.02	6.79	4.43
RAW	1726	4463	26458	48823	268237	160936	90633	441642	274802	371827
Gpixels/sec	0.017	0.043	0.256	0.47	2.6	1.56	0.877	4.27	2.66	3.6
Depth of Field	2172	11453	24064	24861	121874	99608	36113	201635	146756	194802
Mpixels/sec	6.31	33.3	69.9	72.2	354	289	104.9	586	426	565.9
Particle Physics	3391	2059	20017	14539	21810	40308	15248	37103	111741	20029
FPS	536	326	3164	2298	3448	6372	2439	5865	17664	3166.4

Figure 3 – Detailed Geekbench Compute Statistics

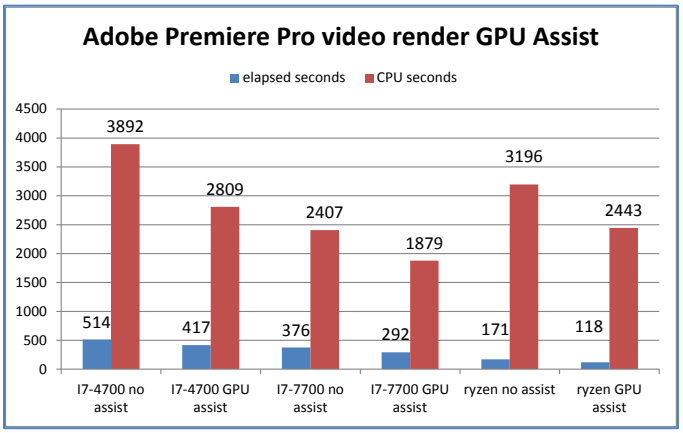

Figure 4 – Adobe Premiere Pro Video Render GPU Assist

The elapsed render times are significantly lower in the Ryzen 16 processor environment, even lower with the Radeon 560x GPU assist and Mercury OpenCL engine. Figures 5 and 6 show the top CPU consumers as reported by Vtune Locks and Waits in CPU/GPU and CPU only rendering environments.

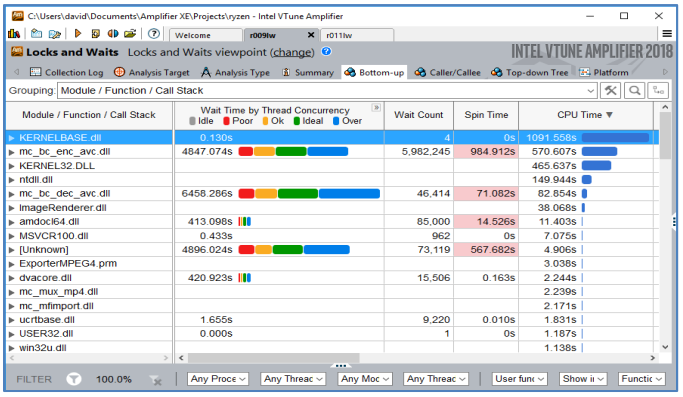

Figure 5 – Premiere Pro Render with GPU Assist

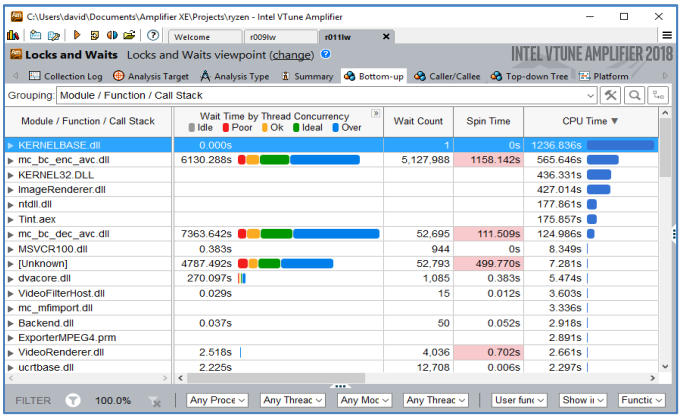

Figure 6 – Premiere Pro Render CPU Only (no GPU assist)

HDFS/Spark/SSD Server Sizing Kit: Real, VMWare, GCP Dataproc,TensorFlowOnSpark - http://davidjyoung.com/cmg/hdfssparksizing.pdf

The Ryzen 7 1700 performs significantly more vioperf seeks than the I7-7700, almost twice as many, giving you a lot more bang for your SSD buck (see Figure 7 following).

	vioperf write mb/s	vioperf read and mb/s	vioperf write mb/s	vioperf read mb/s	vioperf seeks /second	fio read iops	fio read and iops	fio write iops
homessd	501	216	216	480	10,730	5,123	4,784	4,786
usbssd	391	186	186	422	11,047	3,310	3,023	3,024
usbrbs	116	47	47	109	865	170	128	128
i7-4700 vm total	**1,008**	**449**	**449**	**1,011**	**22,642**	**8,603**	**7,935**	**7,938**
homessd	499	211	211	510	37,268	10,912	10,489	10,473
usbssd	415	195	195	436	22,414	6,166	5,558	5,549
usbrbs	119	47	47	115	691	180	129	129
i7-7700 total	**1,033**	**453**	**453**	**1,061**	**60,373**	**17,258**	**16,176**	**16,151**
pers-ssd	185	199	199	162	4,763	1,643	1,651	1,650
pers-rbs	55	33	33	80	219	220	168	167
GCP dataproc total	**240**	**233**	**233**	**242**	**4,982**	**1,863**	**1,819**	**1,817**
homessd	493	226	226	520	69,338	10,200	8,085	8,084
usbssd	389	183	183	401	35,089	5,308	4,688	4,688
usbrbs	105	44	44	110	978	320	169	169
ryzen total	**987**	**453**	**453**	**1,031**	**105,405**	**15,828**	**12,942**	**12,941**

Figure 7 – vioperf, fio statistics

The I7 tests used the T3 usb SSD as input data to the tables residing on the Internal SSD. The configuration was changed for the Ryzen 7 1700 tests: the T3 usb SSD was used as input data to the tables which resided on both the T3 usb SSD and the internal SSD. This change slowed the data load process, but doubled the I/O throughput during query processing, two SSD's are better than one.

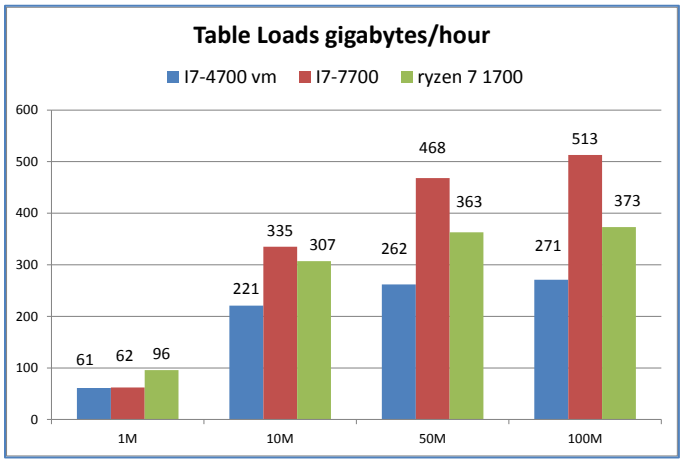

Figure 8 – HDFS textfile, orc table load statistics

Figure 9 shows response times for all 71 Setquery SQLs, for the ten measured tables. The two SSD's used in the Ryzen 7 1700 environment come into play in the 100 million row textfile queries. The table is scanned constantly at approximately 900MB/second for the duration of the workload.

	gcp2 spark1	gcp1 spark1	i77700 spark1	i74700 spark1	Ryzen 16G spark1	Ryzen 16G spark2	Ryzen 32G spark1	Ryzen 32G spark2
1kt	9	41	14	35	29	27	25	29
1ko	27	56	15	22	25	30	24	28
1mt	367	150	58	21	19	85	88	83
1mo	62	53	22	40	20	43	38	39
10mt	509	527	200	310	397	229	168	169
10mo	80	123	85	127	85	85	69	71
50mt	1063	1641	849	2110	2119	2160	638	630
50mo	232	528	225	341	244	220	220	198
100mt	1570	3323	4552	4339	3115	3117	3027	3066
100mo	368	617	394	615	495	388	438	340
total	**4287**	**7059**	**6414**	**7960**	**6548**	**6384**	**4710**	**4653**

Figure 9 – Spark SQL response times

	spark 2 16G task	spark2-16G gc	spark2-32G task	spark2-32G gc	spark2-32G input	spark2-32G shuffle write
1kt	8s	91ms	7s	.2s	22.5m	14.7k
1ko	8s	.3s	7s	.2s	1.7m	14.7k
1mt	2.5m	3s	2.2m	3s	22.7g	267m
1mo	58s	7s	56s	1s	299m	118k
10mt	40m	2.7m	28m	1.9m	228g	266m
10mo	6.7m	1.2m	7.3m	1.6m	2.9g	265m
50mt	6.3h	25m	2.5h	12m	1.1t	1.36g
50mo	37m	6.9m	33m	5.3m	14.4g	1.3g
100mt	13.1h	58m	12.9h	50m	2.3t	2.7g
100mo	1.3h	18m	1.1h	13m	288g	2.8g

Figure 10a – Spark task time, gc time, input, shuffle write

	GC Type	# Collections	Collection Elapsed	Job Elapsed	Queries/ Second	Namenode CPU
I7-4700 VM	CMS	467	3.437	1984	30.24	714
I7-4700 VM	G1	367	3.812	1920	31.25	727
I7-7700	CMS	469	1.712	1400	42.85	322
I7-7700	G1	379	1.796	1400	42.85	330
GCP2				1713	35.03	
GCP3				1412	42.49	
Ryzen 7 1700				1875	32.00	

Figure 10b – JDBC Burners

Figure 11 shows elapsed, CPU, and power (I7-7700/GTX1050) for native tensorflow mnist_deep processing.

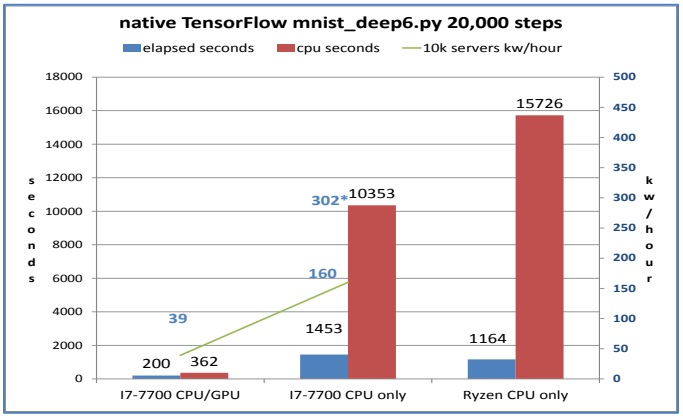

Figure 11 – Native Tensorflow elapsed, CPU, Power

Figure 12 shows the top CPU consumers for the mnist_deep.py process as reported by Vtune's Basic Hotspots.

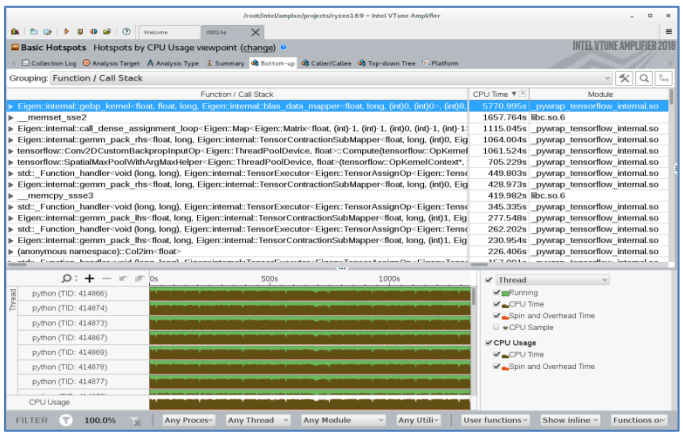

Figure 12 – Native Tensorflow CPU Only Ryzen 7 1700

I7-7700 vs I7-4700: Setquery on Vertica, Couchbase, Mongodb - http://davidjyoung.com/cmg/i77700perf.pdf

Figure 13 following summarizes Vertica KPI for three real machines, two AWS setups, and two GCP setups.

	I7-4700HQ	I7-7700	ryzen 7 1700	E5-2676-1	E5-2676-2	E5-2676-3	Xeon-1	Xeon-2
	pc ssd	pc ssd	pc ssd*2	aws rbs	aws rbs	aws rbs	gcp rbs	gcp ssd
	8p32g	8p32g	16p32g	8p32g	16p64g	40p160g	8p30g	8p30g
create100m	164	111	171	233	234	233	214	206
load100m	117	78	122	269	237	230	314	174
dbd	210	150	236	241	215	242	265	264
load400m	799	561	764	1322	756	680	1542	1026
vsql71q	5.39	4.53	5.33	7.405	5.509	6.244	8.309	7.147
jdbc50	84	68	62	112	99	216	129	128
jdbc50	83	66	60	113	96	220	127	122
backup	24	47	37	163	140	149	180	59
restore	18	36	36	137	132	138	178	53
qph	47,421	56,424	47,955	34,517	46,397	40,935	30,762	35,763
loadmbph	450,563	641,711	471,204	272,315	476,190	529,412	233,463	350,877
jdbcqph	123,614	155,455	171,000	90,796	106,875	46,636	80,787	84,098
backupgbph	2004	1023	1300	295	344	323	267	815
restoregbph	2672	1336	1336	351	364	349	270	907

Figure 13 – Vertica KPI

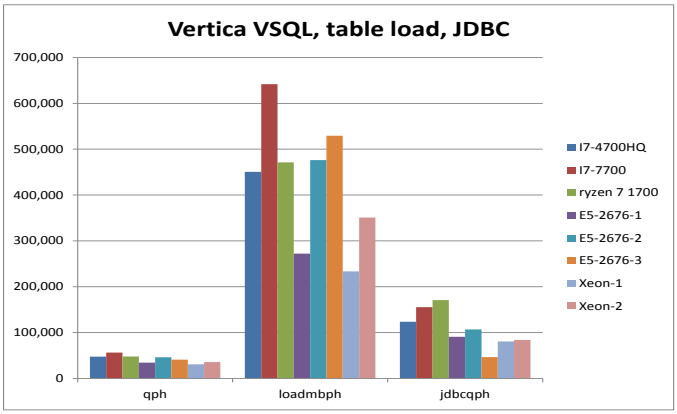

Figure 14 – Vertica VSQL, table loads, JDBC 50 thread

[Futuremark – 3DMark and PCMark10](#)

Three suites are executed under Windows 10: 3DMark Firestrike, 3DMark Timespy, and PCMark10. The Ryzen and I7-7700 boot on 7200 RPM rbs disks, the I7-4700 boots on the Samsung 850 Pro 2TB SSD. The results are summarized in Figures 15-17 following.

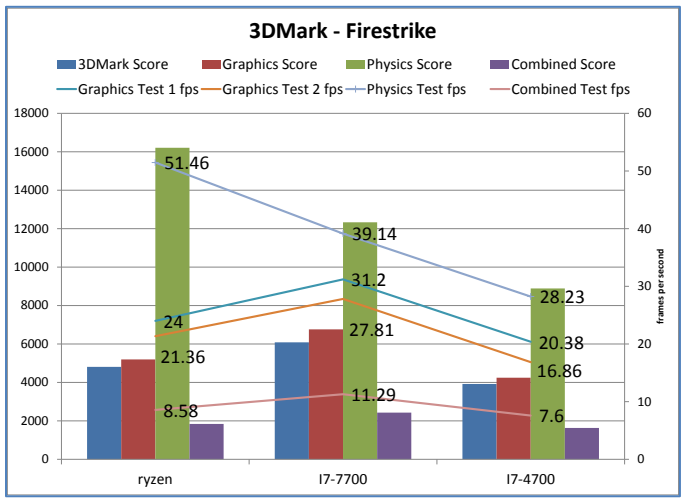

Figure 15 – 3DMark Firestrike Results

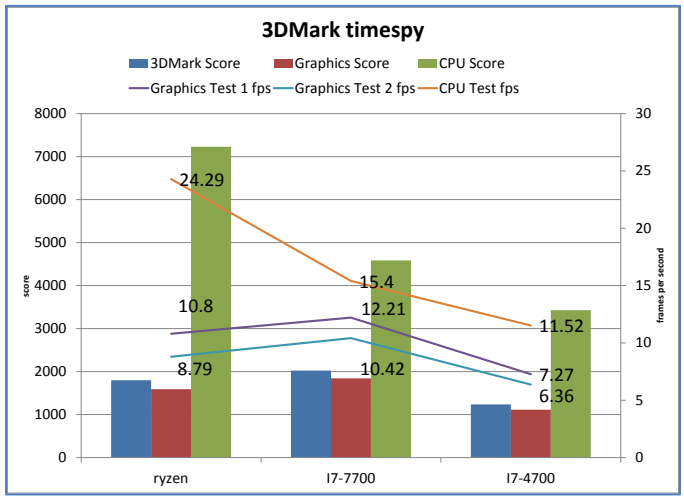

Figure 16 – 3DMark Timespy Results

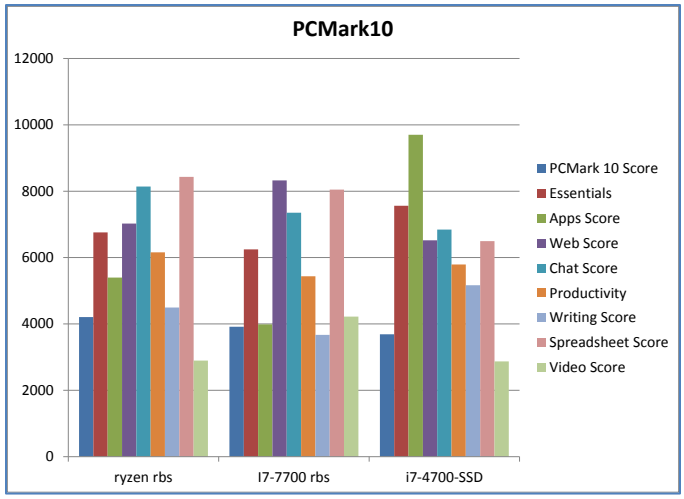

Figure 17 – PCMark10 Results

Figure 18 lists detailed PCMark10 results.

	ryzen rbs	i7-7700 rbs	i7-4700 SSD
PCMark 10 Score	4207	3916	3688
Essentials	6758	6248	7563
Apps Score	5399	3985	9701
Firefox cold start s	5.05462	9.83523	1.99024
GIMP cold start s	8.71706	10.66498	6.09758
Writer cold start s	13.58077	25.02541	4.92462
Chromium cold start s	4.50771	7.94657	0.99416
Firefox warm start s	1.29226	1.11085	0.85319
GIMP warm start s	2.66164	3.15501	2.60996
Writer warm start s	1.81285	2.43854	1.5194
Chromium warm start s	0.32325	0.34625	0.24913
Web Score	7026	8326	6519
Video 2160p VP9 fps	29.64	29.88	30
Video 1080p VP9 fps	30	30	29.88
Video 2160p H.264 fps	30	30	30
Video 1080p H.264 fps	30	30	30
Map infographics update s	0.1657	0.10094	0.17032
Map zooming s	0.04328	0.03642	0.04288
Shop animate 3D object fps	300	300	95.80905
Shop load 3D object s	2.0012	1.39056	1.83416
Shop view image fps	60	60	60
Social media feed update s	0.17918	0.1233	0.1687
Social media page load s	0.18977	0.12071	0.15867
Chat Score	8140	7353	6843
Playback group OCL fps	29.9987	29.98778	29.88333
Face detect group OCL fps	49.09342	11.91051	9.50095
Encode group OCL fps	20.27355	21.45423	19.67811
Playback group CPU fps	29.91708	29.86241	29.8763
Face detect group CPU fps	14.84105	11.93612	9.66277
Playback private OCL fps	29.86889	29.97167	29.99278
Face detect private OCL fps	124.5097	68.76965	55.1727
Encode private OCL fps	26.41868	27.79618	28.47991
Playback private CPU fps	29.99556	29.86056	29.97944
Face detect private CPU fps	57.95088	68.96954	53.88921
Productivity	6157	5435	5793
Writing Score	4495	3670	5167
Add pictures to document s	0.65907	0.58782	0.85144
Cut and paste s	0.3	0.33043	0.3
Save document s	1.88466	1.84118	1.55063
Copy and paste s	0.12	0.21894	0.12
Load document s	2.91471	5.314	1.57094
Spreadsheet Score	8434	8049	6497
Recalculate Energy market OCL s	0.74118	2.39747	3.54437
Recalculate Monte Carlo OCL s	1.9865	3.79838	5.62099
Recalculate Stock history CPU s	1.21478	1.02114	1.39267
Recalculate Building design CPU	0.68403	0.57906	0.81036
Save document s	2.36276	1.73011	2.07096
Edit cells s	0.97635	0.81159	0.97647
Copy data and compute 2 s	3.31618	2.71174	3.38668
Copy formulas s	0.78828	0.9399	0.84573
Copy plain data s	2.53422	2.41384	2.56122
Copy data and compute s	1.93483	1.80334	1.96468
Open document s	1.67066	1.60863	1.53206
Digital Content Creation	4858	4801	3108
Photo Score	6288	4903	3119
Thumbnail loading s	0.15036	0.37087	0.47005
Batch transformation s	20.71258	15.51563	34.24139
Save PNG s	18.73505	16.1905	18.54015
Save JPEG s	1.79325	2.27429	1.91543
Wavelet denoise s	1.29371	2.09182	2.62361
Local contrast s	9.23553	9.31695	25.83339
Unsharp mask 2 s	1.8205	4.99407	7.9317
Gaussian blur s	0.7603	1.9822	3.42433
Noise adding s	0.62194	0.51032	0.77582
Unsharp mask 1 s	5.01298	4.28024	5.65924
Color adjusting s	5.45029	4.95007	7.97814
Video Score	2895	4221	2870
Deshaking OCL fps	65	84	54
Deshaking CPU fps	13	16	14
Downscaling OCL fps	166	173	105
Downscaling CPU fps	36	45	33
On the go fps	13.43471	28.91733	18.17317

Figure 18 – PCMark10 Results

AMD APP SDK – A Complete Development Platform

From the product documentation:

"AMD OpenCL™ Accelerated Parallel Processing (APP) technology is a set of advanced hardware and software technologies that enable AMD graphics processing cores (GPU), working in concert with the system's x86 cores (CPU), to execute heterogeneously to accelerate many applications beyond just graphics. This enables better balanced platforms capable of running demanding computing tasks faster than ever, and sets software developers on the path to optimize for AMD Accelerated Processing Units (APUs). The AMD APP Software Development Kit (SDK) is a complete development platform created by AMD to allow you to quickly and easily develop applications accelerated by AMD APP technology. The SDK provides samples, documentation, and other materials to quickly get you started leveraging accelerated compute using OpenCL™ or C++ AMP in your C/C++ application. (Check the AMD OpenCL™ Zone for these and other tools and libraries.)"

And since this is OpenCL™, everybody is invited to the party: GPUs (AMD and Nvidia), Integrated adapters (Intel) and CPUs (AMD and Intel). This section investigates the OpenCL examples under Windows 10. The Lenovo IdeaCentre equipped with AMD hardware has integrated GPU counter support in both the Task Manager and the Windows Performance Monitor.

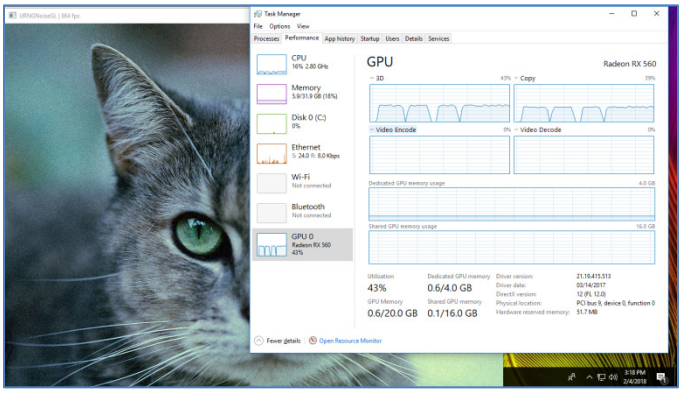

Figure 19 – Task Manager GPU Performance

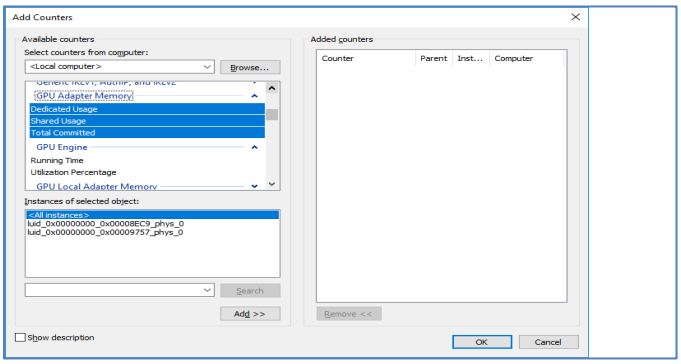

Figure 20 – GPU Performance Counters

The AMD APP SDK for Windows is packaged as Visual Studio Solutions, similar to the Nvidia CUDA BlackScholes_nvrtc sample shown below in Figures 21 – 23.

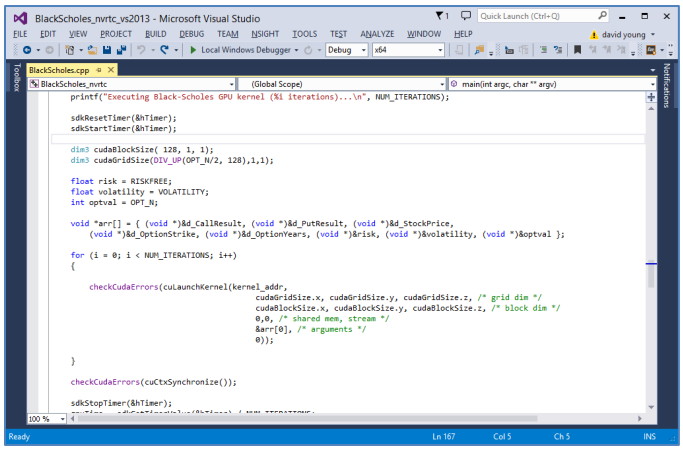

Figure 21 - CUDA BlackScholes_nvrtc GPU code

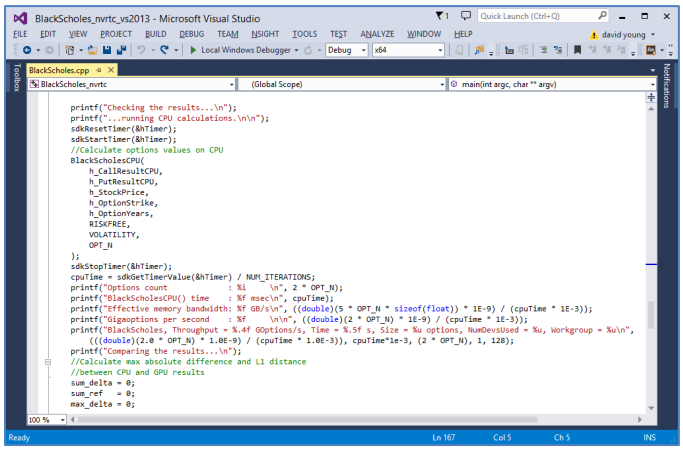

Figure 22 – CUDA BlackScholes_nvrtc CPU code

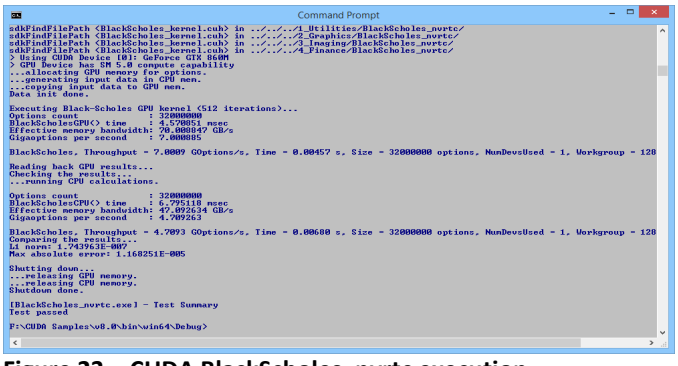

Figure 23 – CUDA BlackScholes_nvrtc execution

Figure 24 shows code from the MemoryOptimizations sample, Figure 25 uses "help" to show the sample's execution parameters. Figure 26 shows the bat file used to test selected samples. The –p parameter identifies the platform, the –device parameter identifies the device, either cpu or gpu.

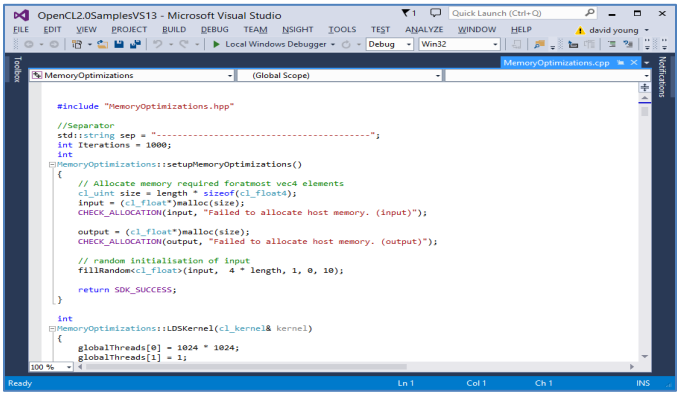

Figure 24 – APP OpenCL 2.0 MemoryOptimizations code

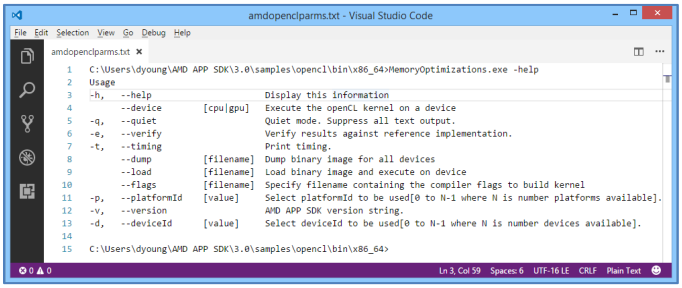

Figure 25 – MemoryOptimizations execution parameters

```
MemoryOptimizations.exe -p 0
MemoryOptimizations.exe -p 1
MemoryOptimizations.exe -p 2
MemoryOptimizations.exe --device cpu
ConstantBandwidth.exe -p 0
ConstantBandwidth.exe -p 1
ConstantBandwidth.exe -p 2
ConstantBandwidth.exe --device cpu
BufferBandwidth.exe -p 0
BufferBandwidth.exe -p 1
BufferBandwidth.exe -p 2
BufferBandwidth.exe --device cpu
GlobalMemoryBandwidth.exe -p 0
GlobalMemoryBandwidth.exe -p 1
GlobalMemoryBandwidth.exe -p 2
GlobalMemoryBandwidth.exe --device cpu
ImageBandwidth.exe -p 0
ImageBandwidth.exe -p 1
ImageBandwidth.exe -p 2
ImageBandwidth.exe --device cpu
LDSBandwidth.exe -p 0
LDSBandwidth.exe -p 1
LDSBandwidth.exe -p 2
LDSBandwidth.exe --device cpu
SoAversusAoS.exe -p 0
SoAversusAoS.exe -p 1
SoAversusAoS.exe -p 2
SoAversusAoS.exe --device cpu
TransferOverlap.exe -p 0
TransferOverlap.exe -p 1
TransferOverlap.exe -p 2
TransferOverlap.exe --device cpu
```

Figure 26 – APP OpenCL 2.0 bat file

Figures 27 - 31 show the speeds of copying various data setups via the MemoryOptimizations sample for the measured GPUs, integrated adapters, and CPUs.

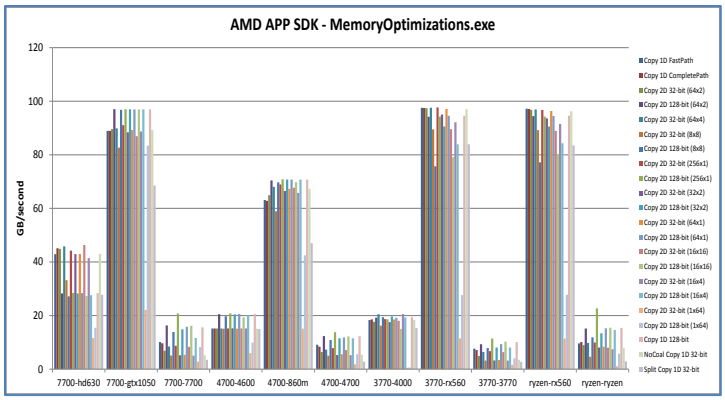

Figure 27 – MemoryOptimizations Results

	7700-hd63(	7700-gtx1050	7700-7700	4700-4600	4700-860m	4700-4700	3770-4000	3770-rx560	3770-3770	ryzen-rx560	ryzen-ryzen
Copy 1D FastPath	42.9	88.94	10.11	15.16	63.13	9.09	18.32	97.56	7.61	97.21	9.61
Copy 1D CompletePath	45.11	88.96	9.64	15.17	62.82	8.38	18.56	97.46	7.13	97.10	10.10
Copy 2D 32-bit (64x2)	44.87	89.61	6.85	15.15	64.91	6.40	17.66	97.41	4.93	96.76	9.00
Copy 2D 128-bit (64x2)	28.24	97.04	16.32	20.52	70.45	12.34	19.23	94.19	9.30	94.44	15.17
Copy 2D 32-bit (64x4)	45.84	89.87	8.43	15.14	68.05	7.29	20.46	97.57	6.38	96.93	9.78
Copy 2D 32-bit (8x8)	33.18	82.67	5.07	15.08	58.91	5.05	16.23	89.50	3.21	89.23	4.61
Copy 2D 128-bit (8x8)	27.14	96.81	13.86	19.72	69.73	10.88	19.39	75.72	7.87	77.21	11.90
Copy 2D 32-bit (256x1)	44.23	91.06	8.73	15.21	68.95	7.83	18.68	97.68	6.69	96.73	9.95
Copy 2D 128-bit (256x1)	28.43	97.03	20.79	20.82	70.85	13.80	18.56	94.28	11.42	94.29	22.66
Copy 2D 32-bit (32x2)	42.95	88.42	5.16	15.21	66.49	5.27	17.54	95.04	3.24	93.61	8.07
Copy 2D 128-bit (32x2)	28.24	97	14.86	20.44	70.76	11.50	19.67	90.54	8.08	90.55	13.39
Copy 2D 32-bit (64x1)	42.96	89.27	5.32	15.11	67.27	5.56	18.52	97.10	3.39	96.36	8.39
Copy 2D 128-bit (64x1)	28.38	96.93	15.82	20.63	70.75	11.81	19.15	94.58	9.33	94.52	15.21
Copy 2D 32-bit (16x16)	46.32	86.93	8.31	15.18	67.71	7.10	18.02	89.58	6.31	88.91	8.01
Copy 2D 128-bit (16x16)	27.25	96.9	16.19	19.31	69.77	12.22	15.02	79.13	10.23	80.32	15.48
Copy 2D 32-bit (16x4)	41.45	88.71	5.02	15.20	65.79	5.19	20.59	92.15	3.20	91.53	7.41
Copy 2D 128-bit (16x4)	27.61	96.968	11.63	20.25	70.75	11.50	19.30	83.89	8.08	84.32	14.60
Copy 2D 32-bit (1x64)	11.62	22.12	2.77	5.96	15.08	1.78	0.49	11.43	1.64	11.42	1.08
Copy 2D 128-bit (1x64)	15.42	83.44	8.23	9.86	42.51	5.51	0.64	27.65	4.05	27.77	5.76
Copy 1D 128-bit	28.39	97	15.59	20.61	70.74	12.28	19.43	94.59	10.08	94.62	15.39
NoCoal Copy 1D 32-bit	43.09	89.31	5.22	15.11	67.34	5.42	18.41	97.07	3.46	96.26	7.93
Split Copy 1D 32-bit	27.73	68.52	3.48	14.98	47.03	2.78	15.34	83.91	2.71	83.49	2.89

Figure 28 - MemoryOptimizations Results

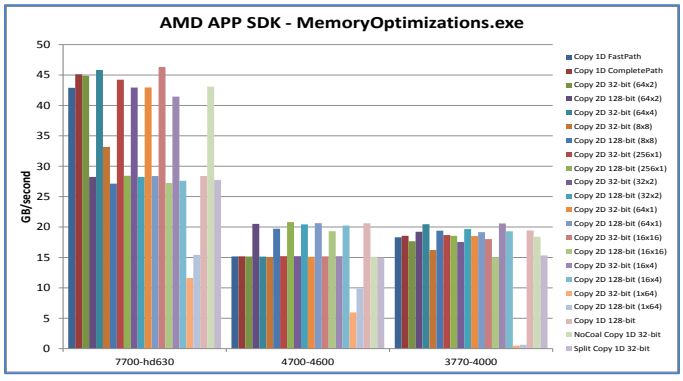

Figure 29 – MemoryOptimizations Integrated Adapters

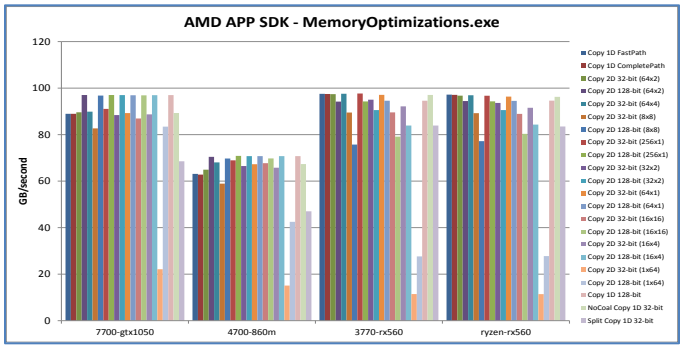

Figure 30 - MemoryOptimizations GPUs

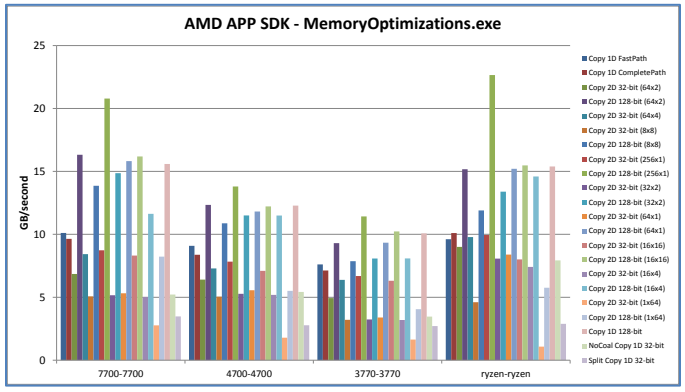

Figure 31 - MemoryOptimizations CPUs

Figure 32 compares speeds for the ConstantBandwidth sample.

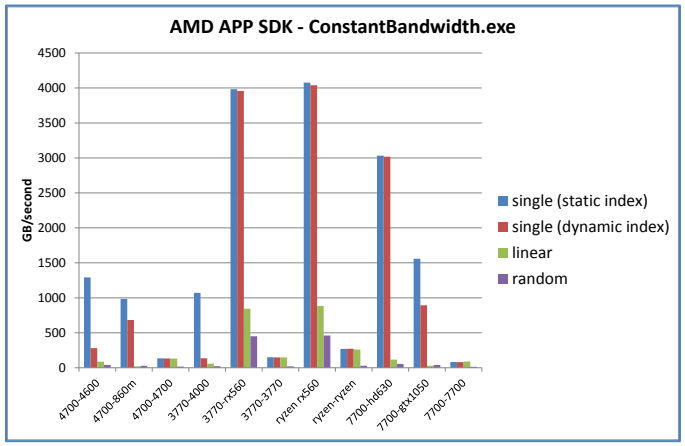

Figure 32 – ConstantBandwidth Results

Figure 33 compares speeds for the GlobalMemoryBandwidth sample.

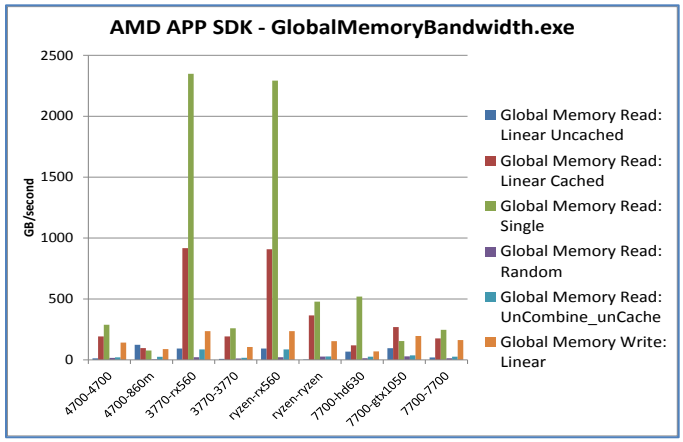

Figure 33 – GlobalMemoryBandwidth Results

Software/Hardware Configuration

The Vmware Player 6.0.6 machine is an ASUS G750JM (i7-4700HQ @2.40GHZ, gbcpu 3620/11722, iopen, nopen, ncuda) with 4 cores, 8 processors, an Nvidia GTX860M, 32GB of RAM (DDR3@1.600GHZ), running Windows 8.1. The VM is a 24GB Centos73 image using 500GB on the Samsung 2TB SSD, all of the usb T3, all of the usb 2TB rbs disk and out of the box HDP 2.6. The Futuremark tests were run under Windows 10, which boots off of the Samsung 2TB SSD. Refer to the following videos: centos7vminstall hdp26install

An ASUS G11CD-K I7-7700 @ 3.60GHZ was also measured (gbcpu 5091/17224, iopen, opencl, cuda), with 4 cores, 8 processors, an Nvidia GTX1050, a 2TB Samsung 850 PRO SSD, the T3 SSD, the usb 2TB rbs, and 32 GB of RAM (DDR4@2.133GHZ). RHEL73 was installed on the 2TB Internal SSD (Server with GUI), no LVM, and out of the box HDP 2.6.

The I7-7700 was re-imaged for the Spark sizing experiments: Centos 7.3, CUDA 8.0.61 (base plus patch) with NVidia driver 375.26-2, CUDnn 6.0 (April 27, 2017 for Cuda 8), tensorflow (1.3.0), tensorflow-gpu (1.3.0), tensorflow-tensorboard (0.1.8), tensorflowonspark (1.0.7), Python 2.7, and HDP 2.6.3. The Nvidia installs were seamless on Centos, just like on Windows.

An IdeaCentre 720 Performance PC (AMD) Ryzen 7 1700 @ 3.00GHZ was also measured (gbcpu 4069/20281, opencl) with 8 cores, 16 processors, a Radeon 560X, a 500GB Samsung 850 PRO SSD, the T3 SSD, the usb 2TB rbs, a 1TB internal rbs for Windows 10, and 32 GB of RAM (DDR4 @ 2.400GHZ). Centos7.3 was installed on the internal 500GB SSD, no LVM, and out of the box HDP 2.6.3.

An ASUS-CM6870 (gbcpu 3740/12811, opencl) was added to the AMD APP SDK Windows 10 measurements, 32GB of RAM, I7-3770 @ 3.40GHZ, two Samsung SSD 850 Pros, and an AMD Radeon 560X.

Three HDP 2.6.3 parms were changed for all machines. The maximum Yarn container size was lowered to 4GB, the Hadoop trash facility was disabled (fs.trash.interval=0), and the Replication factor was set to 1.