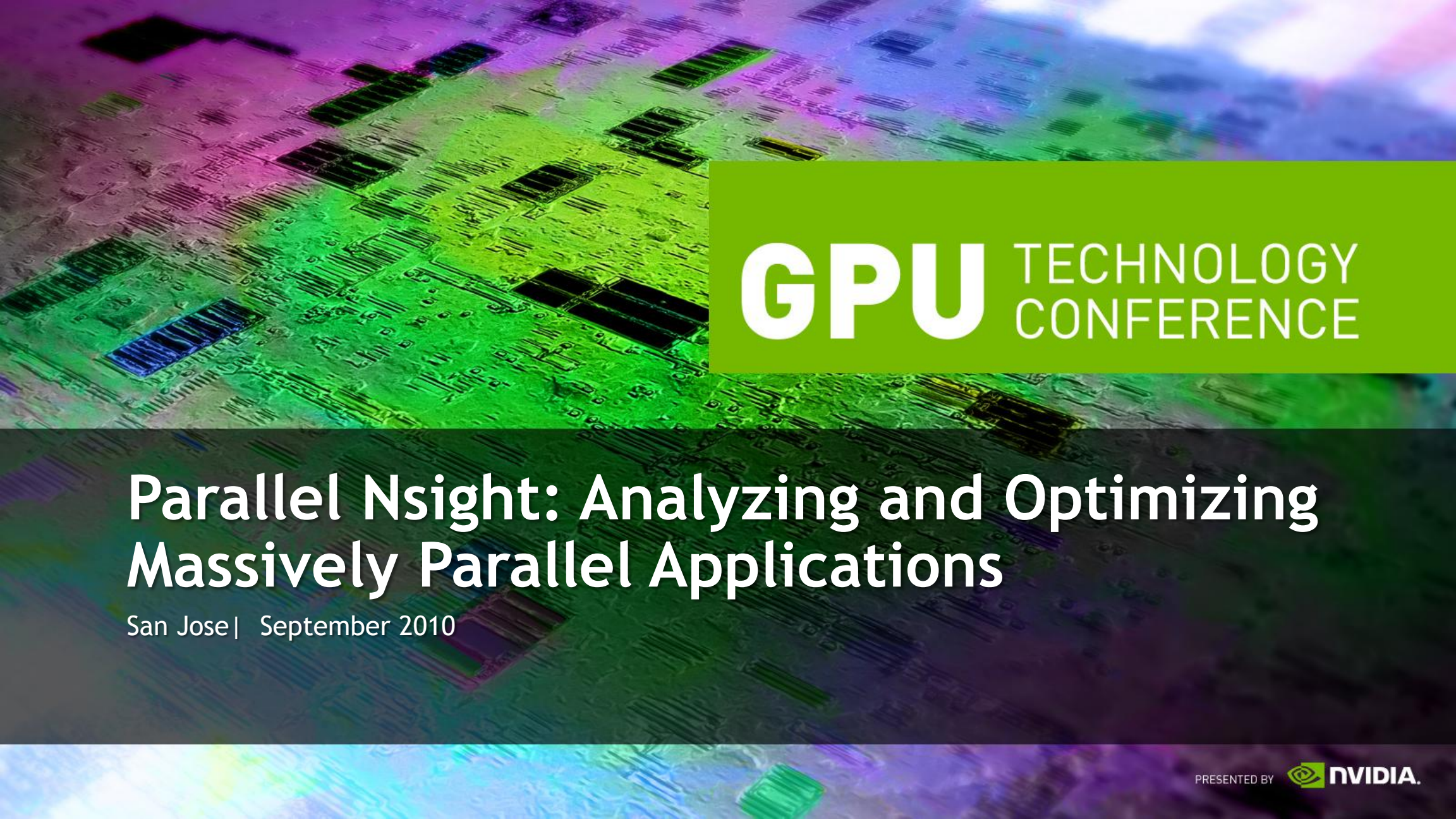




GPU TECHNOLOGY CONFERENCE

Parallel Nsight: Analyzing and Optimizing Massively Parallel Applications

San Jose | September 2010

NVIDIA Parallel Nsight™

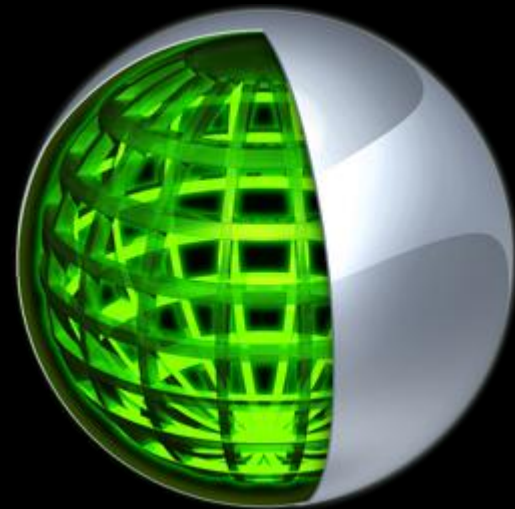
Application Development Environment for Heterogeneous Platforms

- Build GPU code
- Debug GPU APIs and source code
- Profile GPU APIs and source code
- Trace and analyze GPU+CPU activities



New Analysis Features in Parallel Nsight 1.5

- Visual Studio 2010 support
- Trace and profiling activities in the same session
- CUDA profiler improvements
- NVIDIA tools extension improvements
- OpenGL/Direct3D frame and workloads trace



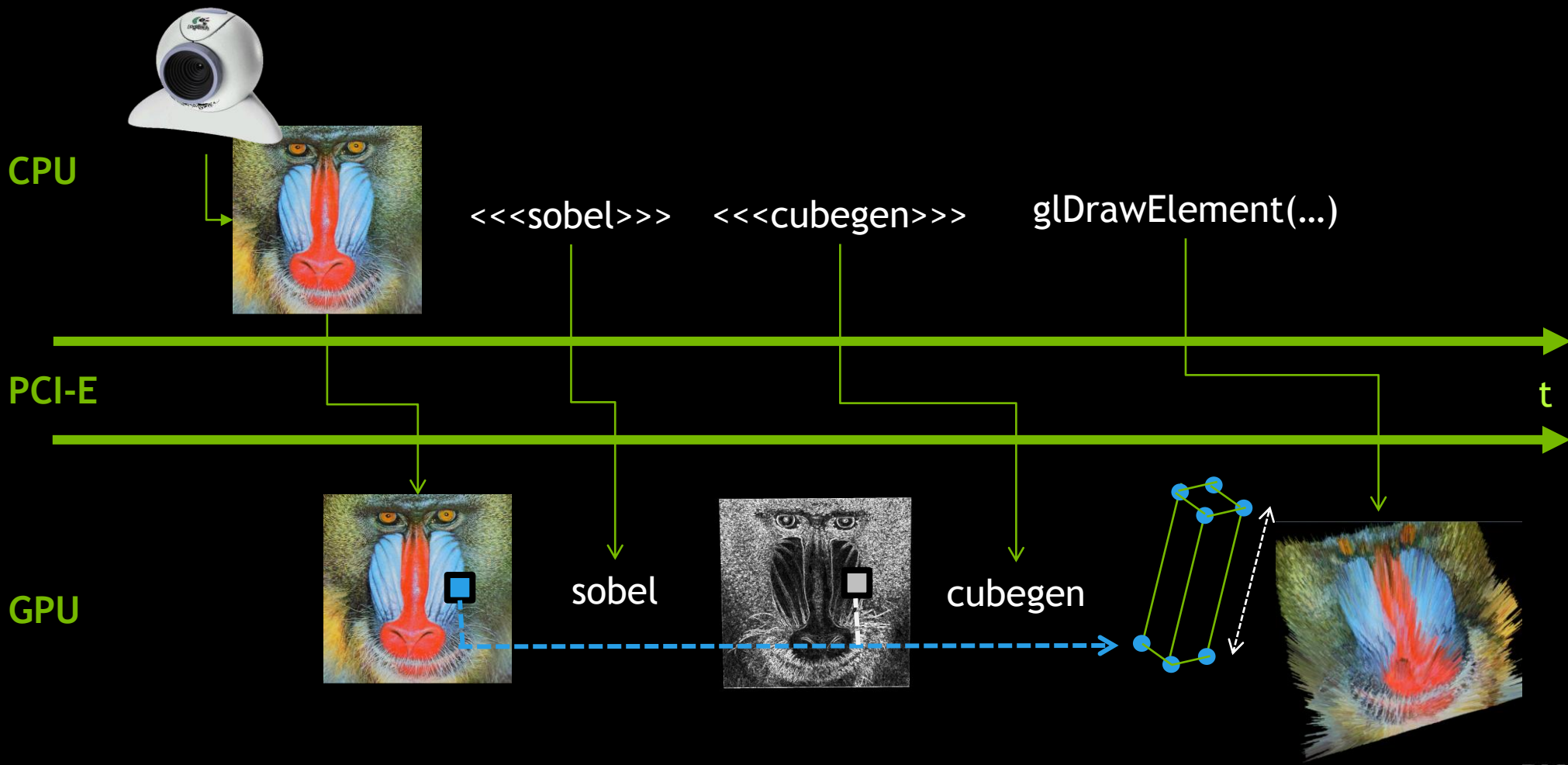
Analysis and Optimization Strategies

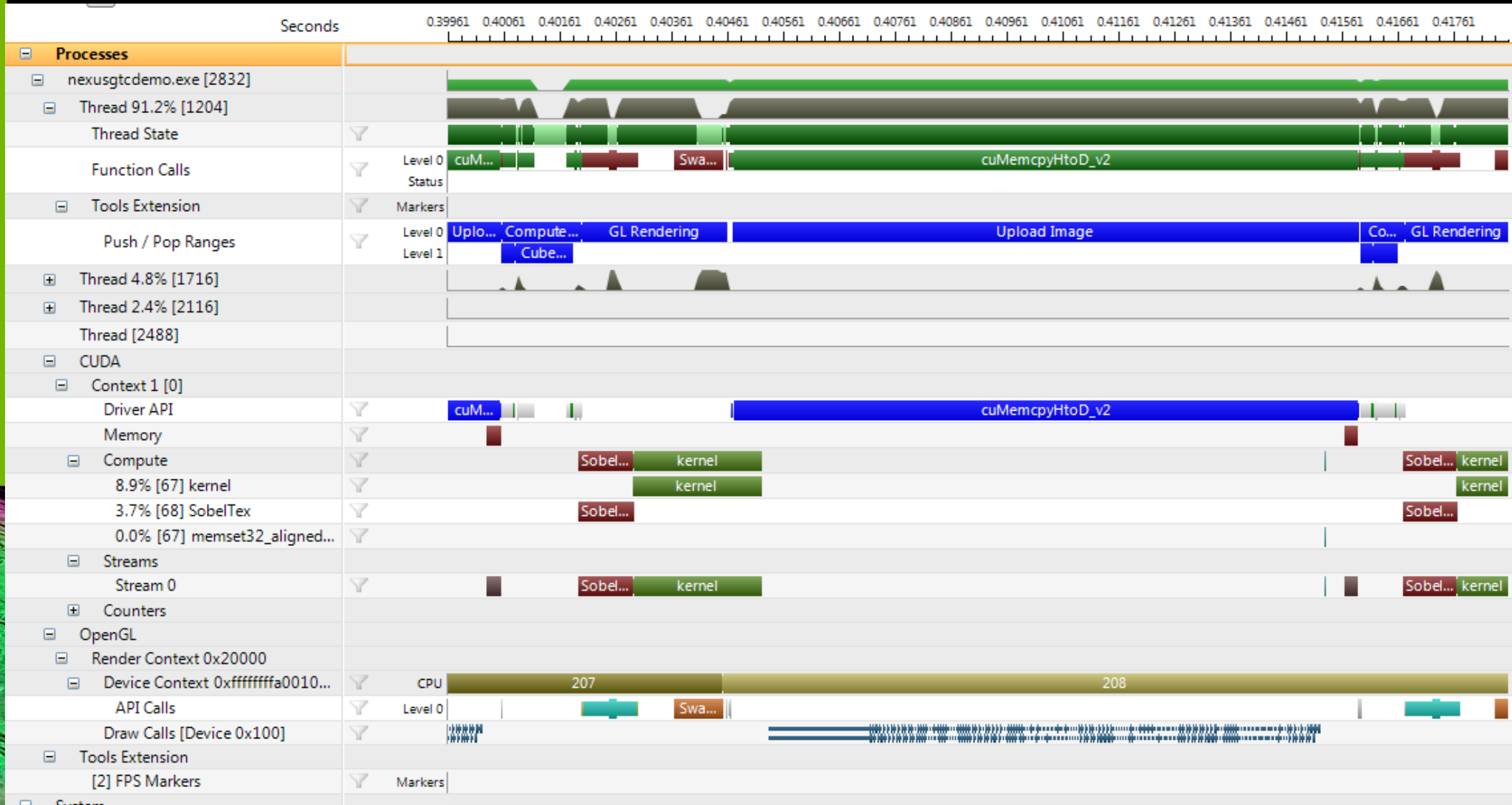
- Understand what the application is really doing
- Untangle false dependencies
 - consumer/producer/interop
 - multi-core/multi-GPU
- Find out where the cycles are spent
- Profile the big offenders

What is my application really doing?

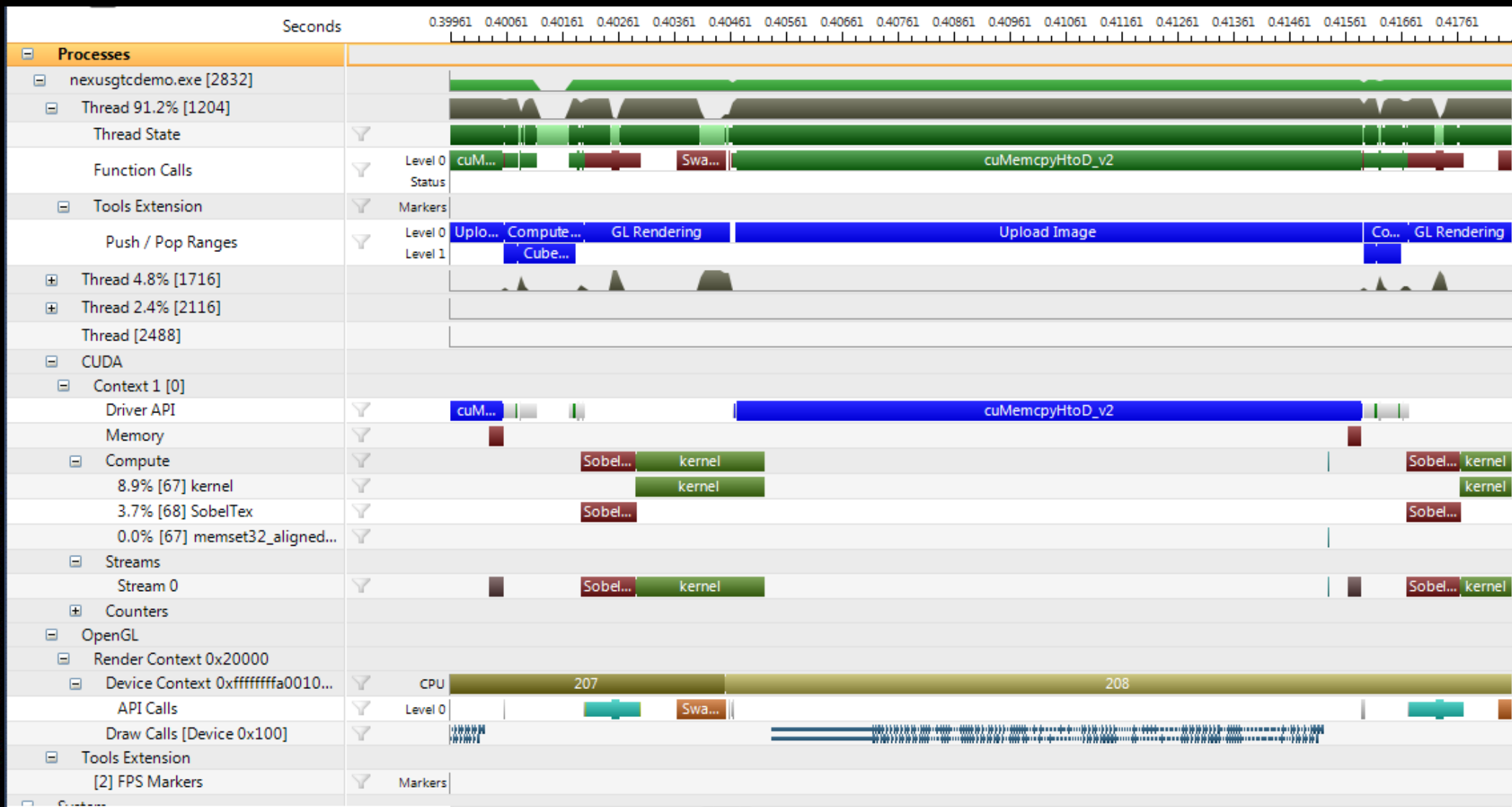
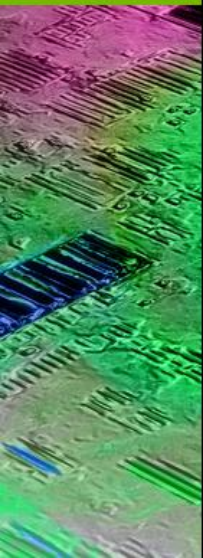
- Trace
 - Run your application as is in Parallel Nsight
 - Source code decoration
 - Performance monitors
- CPU threads
- GPU Workloads
- Memory transfers
- Dependencies

Nsight GTC 2010 Demo Introduction





Event ID	Level	Indent	Message	Event Type	Time (μs)	Duration (μs)	Process ID	Thread ID	Category	Color	Payload
5		L	Cube generation	RangePop	133344.8	489.6	2832	1204	[0]	□	
3		L	Compute Work	RangePop	133528.0	898.0	2832	1204	[0]	□	
6		r	GL Rendering	RangePush	133529.2	1886.0	2832	1204	[0]	□	
6		L	GL Rendering	RangePop	135415.2	1886.0	2832	1204	[0]	□	
7		r	Upload Image	RangePush	136701.6	10779.2	2832	1204	[0]	□	
7		L	Upload Image	RangePop	147480.8	10779.2	2832	1204	[0]	□	
8		r	Compute Work	RangePush	147523.6	934.0	2832	1204	[0]	□	
9		r	Sobel	RangePush	147524.8	235.2	2832	1204	[0]	□	
9		L	Sobel	RangePop	147760.0	235.2	2832	1204	[0]	□	
10		r	Cube generation	RangePush	147760.8	508.4	2832	1204	[0]	□	
10		L	Cube generation	RangePop	148269.2	508.4	2832	1204	[0]	□	
8		L	Compute Work	RangePop	148457.6	934.0	2832	1204	[0]	□	
11		r	GL Rendering	RangePush	148458.4	1911.2	2832	1204	[0]	□	
11		L	GL Rendering	RangePop	150369.6	1911.2	2832	1204	[0]	□	
12			GTC 2010 Nsight Demo [Sobel] 37.0 fps (linear,no stream)	Marker	150700.0		2832	1204	[2] FPS Markers	■	37.0478744506836
13		r	Upload Image	RangePush	151837.6	10493.6	2832	1204	[0]	□	
13		L	Upload Image	RangePop	162331.2	10493.6	2832	1204	[0]	□	
14		r	Compute Work	RangePush	162586.0	4926.8	2832	1204	[0]	□	
15		r	Sobel	RangePush	162588.4	236.0	2832	1204	[0]	□	
15		L	Sobel	RangePop	162824.4	236.0	2832	1204	[0]	□	
16		r	Cube generation	RangePush	162825.2	4506.4	2832	1204	[0]	□	
16		L	Cube generation	RangePop	167331.6	4506.4	2832	1204	[0]	□	
14		L	Compute Work	RangePop	167512.8	4926.8	2832	1204	[0]	□	
17		r	GL Rendering	RangePush	167514.0	1900.0	2832	1204	[0]	□	
17		L	GL Rendering	RangePop	169414.0	1900.0	2832	1204	[0]	□	
18		r	Upload Image	RangePush	170258.4	10387.6	2832	1204	[0]	□	
18		L	Upload Image	RangePop	180646.0	10387.6	2832	1204	[0]	□	
19		r	Compute Work	RangePush	180804.4	965.6	2832	1204	[0]	□	
20		r	Sobel	RangePush	180805.2	230.8	2832	1204	[0]	□	
20		L	Sobel	RangePop	181036.0	230.8	2832	1204	[0]	□	
21		r	Cube generation	RangePush	181036.8	535.6	2832	1204	[0]	□	



Activity Type

☐ Application Trace

Collects events from the target application. The analysis session and data collection are stopped when the launched application exits.

☐ System Trace

Collects events from the target application and all native child processes of the target application. The analysis session and data collection are not stopped when the launched application exits. The session and data collection must be stopped manually.

☒ Profile CUDA Kernels

Collects performance counters for each CUDA kernel launch.

Profile CUDA Settings

Counter Set

GeForce GTX 480

Single Multiprocessor

- | | | |
|--|--|---|
| ☐ Branches Taken | ☐ Instructions Issued | ☐ Active Warps |
| ☐ Divergent Branches | ☐ Instructions Executed | ☒ Shared Load |
| ☐ Thread Blocks Executed (Multiprocessor) | ☐ Warps Launched | ☐ Shared Store |
| ☒ Global Load Requests | ☐ Threads Launched | |
| ☐ Global Store Requests | ☐ Active Cycles | |

Texture Processing Cluster (TPC)

- ☒ Local Loads
- ☐ Local Stores

Connection Status



Application Control



Launch

Kill

Capture Control



Start

Stop

Cancel

☒ Open Report on Stop

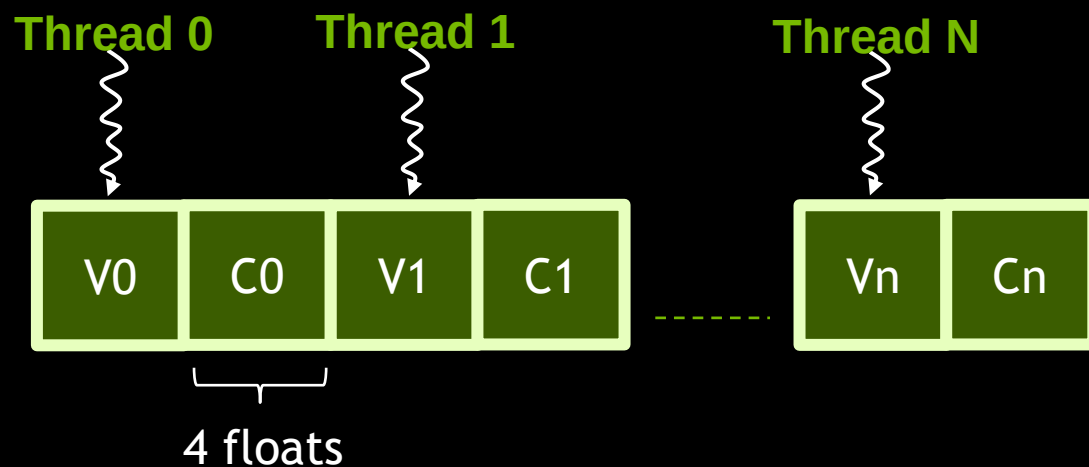
Summary Page

Kernel with Interleaved memory layout

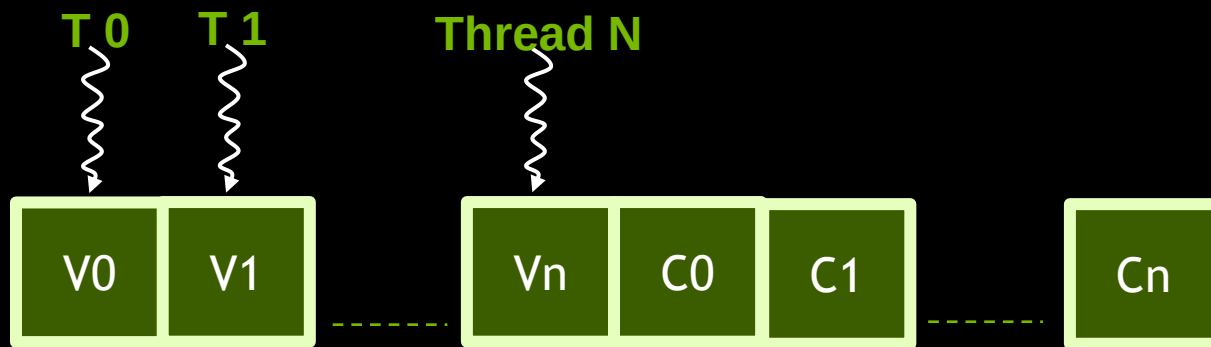
Function Name	Duration (μs)	Global Store Requests	Shared Store	Local Stores	FB Subpartition 0 Read Sectors	FB Subpartition 0 Write Sectors	FB Read Throughput [approx.] (MB/s)	FB Write Throughput [approx.] (MB/s)
kernel	22597.6	72000	0	115835	519143	13057018	1,402.2	35,266.5
kernel	22483.6	74304	0	119511	523513	13046715	1,421.2	35,417.3
kernel	22462.8	74304	0	119434	519920	13048891	1,412.7	35,456.0
kernel	22418.8	75456	0	121267	520243	13048259	1,416.4	35,523.9
kernel	22451.2	75456	0	121332	526749	13053570	1,432.0	35,487.0
kernel	22529.6	74880	0	120370	525019	13049615	1,422.3	35,352.8
kernel	22436.0	73728	0	118520	523504	13049836	1,424.1	35,500.9
kernel	22524.8	74880	0	120422	522644	13049448	1,416.2	35,359.9
kernel	22484.8	74304	0	119440	519003	13049089	1,408.8	35,421.8
kernel	22397.2	75456	0	121361	519608	13046184	1,416.0	35,552.5
kernel	22564.4	74880	0	120311	525885	13048560	1,422.5	35,295.5
kernel	22532.0	75456	0	121276	519704	13049653	1,407.8	35,349.2
kernel	22476.4	75456	0	121236	518500	13047390	1,408.0	35,430.5
kernel	22442.0	74304	0	119497	517915	13047078	1,408.6	35,483.9
kernel	22459.6	73728	0	118532	522938	13044791	1,421.1	35,449.9
kernel	22492.8	74304	0	119490	524786	13046592	1,424.0	35,402.5
kernel	22509.2	75456	0	121287	525883	13047034	1,426.0	35,377.9
kernel	22440.8	76032	0	122298	524518	13049641	1,426.6	35,492.8

Vertex attribute memory layout

- Interleaved



- Linear



Kernel with Linear memory layout

Function Name	Duration (μs)	Global Store Requests	Shared Store	Local Stores	FB Subpartition 0 Read Sectors	FB Subpartition 0 Write Sectors	FB Read Throughput [approx.] (MB/s)	FB Write Throughput [approx.] (MB/s)
kernel	5146.4	84672	0	135811	159908	4915773	1,896.5	58,300.0
kernel	5183.6	84096	0	134929	160983	4916064	1,895.5	57,885.0
kernel	5177.6	85248	0	136737	156580	4970127	1,845.8	58,589.4
kernel	5142.8	85248	0	136811	152981	4959130	1,815.6	58,855.3
kernel	5148.4	84672	0	135928	155305	4956968	1,841.2	58,765.7
kernel	5143.2	84672	0	135881	149961	4942619	1,779.6	58,654.8
kernel	5148.4	84672	0	135862	150574	4947850	1,785.1	58,657.6
kernel	5162.0	85248	0	136783	149982	4959093	1,773.4	58,636.0
kernel	5156.4	84096	0	134947	158388	4943268	1,874.8	58,512.4
kernel	5160.4	84672	0	135803	155813	4956547	1,842.9	58,624.1
kernel	5167.2	84672	0	135892	157720	4925025	1,863.0	58,174.6
kernel	5230.8	85824	0	137642	161716	4992312	1,887.0	58,252.4
kernel	5180.0	84096	0	134966	160375	4920583	1,889.7	57,978.5
kernel	5142.4	84672	0	135868	149925	4957396	1,779.5	58,839.3
kernel	5156.0	84096	0	134936	153404	4946788	1,815.9	58,558.6
kernel	5165.2	84672	0	135860	150422	4957690	1,777.5	58,583.1
kernel	5153.6	84096	0	134927	153944	4944586	1,823.2	58,559.8
kernel	5194.4	84672	0	135879	165617	4944656	1,946.0	58,100.6

Cache configuration

Function Name	Duration (μs)	Cache Configuration Executed	Global Store Requests	Shared Store	Local Stores	FB Subpartition 0 Read Sectors	FB Subpartition 0 Write Sectors	FB Read Throughput [approx.] (MB/s)	FB Write Throughput [approx.] (MB/s)
kernel	4940.0	PREFER_L1	85824	0	137894	103971	4537912	1,284.6	56,067.2
kernel	4928.4	PREFER_L1	85824	0	137846	103876	4517671	1,286.4	55,948.5
kernel	4939.2	PREFER_L1	85248	0	136990	100606	4514695	1,243.2	55,789.4
kernel	4944.4	PREFER_L1	85248	0	136931	103198	4529098	1,273.9	55,908.5
kernel	4938.0	PREFER_L1	85824	0	137850	106134	4530734	1,311.8	56,001.2
kernel	4928.4	PREFER_L1	85824	0	137812	104256	4523948	1,291.1	56,026.3
kernel	4953.2	PREFER_L1	85824	0	137839	107154	4526062	1,320.4	55,771.8
kernel	4942.8	PREFER_L1	84672	0	135961	103479	4533991	1,277.8	55,987.1
kernel	4941.6	PREFER_L1	85824	0	137888	107113	4528358	1,323.0	55,931.1
kernel	4928.8	PREFER_L1	85824	0	137826	101552	4549040	1,257.6	56,332.4
kernel	4927.2	PREFER_L1	85248	0	136903	106318	4497306	1,317.0	55,709.9
kernel	4942.0	PREFER_L1	85824	0	137816	100655	4515093	1,243.1	55,762.7
kernel	4949.6	PREFER_L1	85824	0	137883	104401	4527039	1,287.4	55,824.4
kernel	4922.4	PREFER_L1	85824	0	137821	98951	4527374	1,226.9	56,137.0
kernel	4934.8	PREFER_L1	85248	0	137015	100311	4527038	1,240.7	55,991.8
kernel	4940.0	PREFER_L1	85824	0	137941	104699	4534311	1,293.6	56,022.7
kernel	4949.2	PREFER_L1	85824	0	137867	106090	4537820	1,308.3	55,961.9
kernel	4948.8	PREFER_L1	85824	0	137896	105505	4527010	1,301.2	55,833.1

Function Name	Start Time (μs)	Duration (μs)	Occupancy	Registers per Thread	Dynamic Shared Memory per Block (bytes)	Local Memory per Thread (bytes)	Local Memory (bytes)	Cache Configuration Executed	Grid Dimensions	Block Dimensions	API Call ID	Stream ID	Process ID	Local Loads	Global Load Requests	Shared Load
kernel	699108.0	10208.4	0.67	28	0	0	48758784	PREFER_SHARED	{32, 32, 1}	{16, 16, 1}	229972	1	2972	71281	1984	0
kernel	2612819.6	10202.4	0.67	28	0	0	48758784	PREFER_SHARED	{32, 32, 1}	{16, 16, 1}	245494	1	2972	71289	1984	0
kernel	863008.0	10206.8	0.67	28	0	0	48758784	PREFER_SHARED	{32, 32, 1}	{16, 16, 1}	231365	2	2972	72471	2016	0
kernel	581553.6	10206.0	0.67	28	0	0	48758784	PREFER_SHARED	{32, 32, 1}	{16, 16, 1}	229375	2	2972	71328	1984	0
kernel	556753.6	10209.6	0.67	28	0	0	48758784	PREFER_SHARED	{32, 32, 1}	{16, 16, 1}	229176	1	2972	72512	2016	0
kernel	2708806.4	10191.2	0.67	28	0	0	48758784	PREFER_SHARED	{32, 32, 1}	{16, 16, 1}	246290	1	2972	71312	1984	0
kernel	839493.2	10210.8	0.67	28	0	0	48758784	PREFER_SHARED	{32, 32, 1}	{16, 16, 1}	231166	1	2972	71422	1984	0
kernel	2589292.8	10196.0	0.67	28	0	0	48758784	PREFER_SHARED	{32, 32, 1}	{16, 16, 1}	245295	2	2972	71311	1984	0
SobelShared	2082487.6	771.2	0.33	28	2304	0	48758784	PREFER_SHARED	{2, 128, 1}	{16, 4, 1}	241428	1	2972	12272	3888	3240
SobelShared	2500223.2	774.4	0.33	28	2304	0	48758784	PREFER_SHARED	{2, 128, 1}	{16, 4, 1}	244612	1	2972	12192	3888	3240
SobelShared	2635255.6	774.8	0.33	28	2304	0	48758784	PREFER_SHARED	{2, 128, 1}	{16, 4, 1}	245607	2	2972	12224	3888	3240
SobelShared	2935280.4	770.4	0.33	28	2304	0	48758784	PREFER_SHARED	{2, 128, 1}	{16, 4, 1}	247796	1	2972	12176	3888	3240
SobelShared	2106368.0	774.4	0.33	28	2304	0	48758784	PREFER_SHARED	{2, 128, 1}	{16, 4, 1}	241627	2	2972	11504	3672	3060
SobelShared	2743342.0	772.8	0.33	28	2304	0	48758784	PREFER_SHARED	{2, 128, 1}	{16, 4, 1}	246403	2	2972	12224	3888	3240
SobelShared	2523619.2	774.0	0.33	28	2304	0	48758784	PREFER_SHARED	{2, 128, 1}	{16, 4, 1}	244811	2	2972	11536	3672	3060
SobelShared	2587812.8	774.8	0.33	28	2304	0	48758784	PREFER_SHARED	{2, 128, 1}	{16, 4, 1}	245209	2	2972	11552	3672	3060
SobelShared	2034866.8	773.2	0.33	28	2304	0	48758784	PREFER_SHARED	{2, 128, 1}	{16, 4, 1}	241030	1	2972	11536	3672	3060

← → ↻ 🖥️ 🏠 🌐 ▶ CUDA Launches

▼ Filter

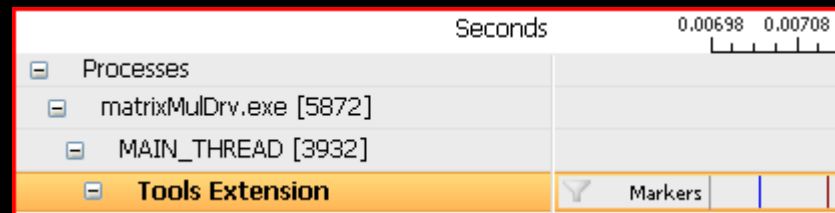
Function Name	Start Time (μs)	Duration (μs)	Occupancy	Registers per Thread	Dynamic Shared Memory per Block (bytes)	Cache Configuration Executed	Grid Dimensions	Block Dimensions	API Call ID	Context ID	Stream ID	Process ID	Global Load Requests	Shared Load	Local Loads
SobelTex	2469225.8	1439.5	0.75	26	0	PREFER_SHARED	{512, 1, 1}	{384, 1, 1}	11378	1	1	5684	15840	0	47520
SobelTex	2499741.5	1445.3	0.75	26	0	PREFER_SHARED	{512, 1, 1}	{384, 1, 1}	11556	1	2	5684	16272	0	48816
SobelTex	2530331.4	1445.0	0.75	26	0	PREFER_SHARED	{512, 1, 1}	{384, 1, 1}	11734	1	1	5684	16272	0	48816
SobelTex	2561090.5	1447.3	0.75	26	0	PREFER_SHARED	{512, 1, 1}	{384, 1, 1}	11912	1	2	5684	16272	0	48816
SobelTex	2591415.9	1448.1	0.75	26	0	PREFER_SHARED	{512, 1, 1}	{384, 1, 1}	12090	1	1	5684	16272	0	48816
SobelTex	2621833.7	1447.5	0.75	26	0	PREFER_SHARED	{512, 1, 1}	{384, 1, 1}	12268	1	2	5684	16272	0	48816
SobelTex	2652105.3	1438.6	0.75	26	0	PREFER_SHARED	{512, 1, 1}	{384, 1, 1}	12446	1	1	5684	15840	0	47520
SobelShared	2683873.4	889.6	0.75	28	5376	PREFER_SHARED	{2, 43, 1}	{16, 12, 1}	12638	1	2	5684	3024	3240	9744
SobelShared	2714936.2	886.9	0.75	28	5376	PREFER_SHARED	{2, 43, 1}	{16, 12, 1}	12830	1	1	5684	3024	3240	9744
SobelShared	2746238.4	885.3	0.75	28	5376	PREFER_SHARED	{2, 43, 1}	{16, 12, 1}	13022	1	2	5684	3024	3240	9744
SobelShared	2777040.5	888.1	0.75	28	5376	PREFER_SHARED	{2, 43, 1}	{16, 12, 1}	13214	1	1	5684	3024	3240	9744
SobelShared	2807614.7	887.5	0.75	28	5376	PREFER_SHARED	{2, 43, 1}	{16, 12, 1}	13406	1	2	5684	3024	3240	9744
SobelShared	2838130.9	888.6	0.75	28	5376	PREFER_SHARED	{2, 43, 1}	{16, 12, 1}	13598	1	1	5684	3024	3240	9744
SobelShared	2868770.2	889.8	0.75	28	5376	PREFER_SHARED	{2, 43, 1}	{16, 12, 1}	13790	1	2	5684	3024	3240	9744
SobelShared	2899693.3	888.4	0.75	28	5376	PREFER_SHARED	{2, 43, 1}	{16, 12, 1}	13982	1	1	5684	3024	3240	9744

NVIDIA Tools eXtensions - NVTX

- Source code instrumentation and decoration:
 - Markers
 - Ranges
 - Resource Naming
- `nvToolsExt.h`

Markers

```
// NEW - message only  
nvtx::Mark("video transfer")
```



Event ID	Level Indent	Message	Event Type	Time (µs)	Duration (µs)	Process ID	Thread ID	Category	Color	Payload
1		Entering main()	Marker	7035.7		5872	3932		[0]	
2		Entering runTest()	Marker	7115.6		5872	3932		[0]	12345678

Attributes

- Color ARGB
- Payload/Type 64-bit - unsigned int64, int64, double
- Message Text
- Category ID for grouping

```
// NEW - w/ attributes
```

```
nvtx::Mark(nvtx::Attributes().color(0xFFFF0000).message("video transfer"));
```

Attributes - Source Code Sample

```
nvtxEventAttributes_t eventAttrib = {0};  
eventAttrib.version = NVTX_VERSION;  
eventAttrib.size = NVTX_EVENT_ATTRIB_STRUCT_SIZE;  
eventAttrib.category = CATEGORY_VIDEO;  
eventAttrib.colorType = NVTX_COLOR_ARGB;  
eventAttrib.color = COLOR_RED;  
eventAttrib.messageType = NVTX_MESSAGE_TYPE_ASCII;  
eventAttrib.message.ascii = "video transfer";  
eventAttrib.payloadType = NVTX_PAYLOAD_TYPE_INT64;  
eventAttrib.payload.llValue = nBytes;  
  
nvtxMarkEx(&eventAttrib);  
  
// NVTX HELPER  
  
nvtx::Mark(nvtx::Attributes().category(CATEGORY_VIDEO).color(COLOR_RED).message("video  
transfer").payload(nBytes));
```


Ranges and Scoped Ranges

```
nvtxRangePushA("Sobel");
```

```
...
```

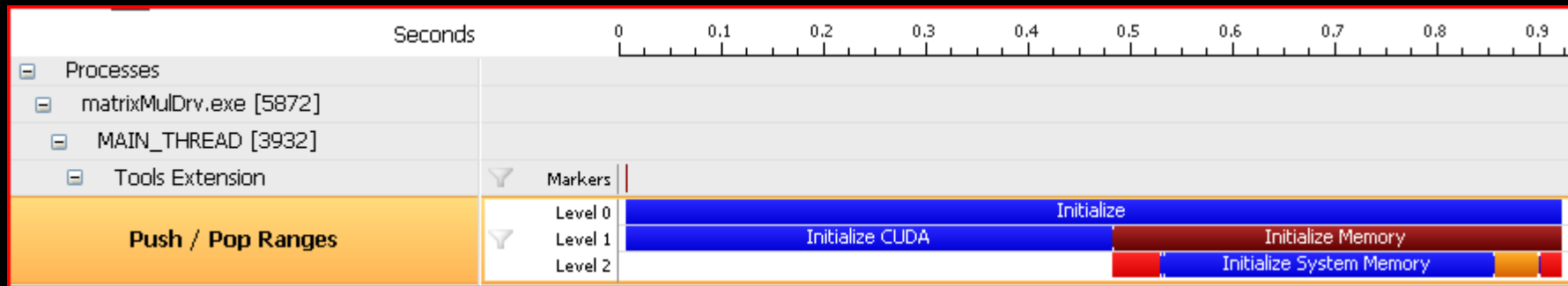
```
nvtxRangePop();
```

```
{
```

```
    NvtxUtil::ScopedRange SR("Update Video Frame");
```

```
...
```

```
}
```



Event ID	Level Indent	Message	Event Type	Time (µs)	Duration (µs)	Process ID	Thread ID	Category	Color	Payload
6		Initialize Memory	RangePush	483099.9	437531.1	5872	3932	[0]	■	
7		Allocate System Memory (Input)	RangePush	483109.8	46029.1	5872	3932	[1] Memory Allocation	■	33554432
7		Allocate System Memory (Input)	RangePop	529138.9	46029.1	5872	3932	[1] Memory Allocation	■	33554432
8		Memory Object 2	RangePush	529151.8	3662.1	5872	3932	[0]	□	
9		Memory Object 2	RangePop	529165.5	581677.7	5872	3932	[2] Memory Object Lifetime	■	

Nested Ranges

```
nvtxRangePushA("Compute Work");
```

```
    nvtxRangePushA("Sobel");
```

```
    ...
```

```
    nvtxRangePop();
```

```
    nvtxRangePushA("CubeGen");
```

```
    ...
```

```
    nvtxRangePop();
```

```
nvtxRangePop();
```

Ranges and Object Lifetime

```
nvtxEventAttributes_t lifeTimeAttrib = {0};  
lifeTimeAttrib.version = NVTX_VERSION;  
lifeTimeAttrib.size = NVTX_EVENT_ATTRIB_STRUCT_SIZE;  
lifeTimeAttrib.category = 3;  
lifeTimeAttrib.color = 0xFF0088FF;  
lifeTimeAttrib.colorType = NVTX_COLOR_ARGB;  
lifeTimeAttrib.messageType = NVTX_MESSAGE_TYPE_ASCII;  
lifeTimeAttrib.message.ascii = "Memory Object A";  
lifeTime_D_A = nvtxRangeStartEx(&lifeTimeAttrib);  
cutilDrvSafeCallNoSync(cuMemAlloc( &d_A, mem_size_A ));  
...  
cutilDrvSafeCallNoSync(cuMemFree(d_A));  
nvtxRangeEnd(lifeTime_D_A);
```

Ranges and Object Lifetime

```
// nvtxHelper
nvtxRangeId_t rid_A = nvtx::RangeStart(nvtx::Attributes()
    .category(CATEGORY_CUDA_MEMORY)
    .color(COLOR_RED).message("A"));

cuMemAlloc(&d_A, mem_size_A);

...

cuMemFree(d_A);
nvtx::RangeEnd(rid_A);
```


Event ID	Level Indent	Message	Event Type	Time (μs)	Duration (μs)	Process ID	Thread ID	Category	Color	Payload
9		Memory Object A	RangeStart	529165.5	591677.7	5872	3932	[3] Memory Object Lifetime	Blue	
10		Memory Object B	RangeStart	531033.5	590164.6	5872	3932	[3] Memory Object Lifetime	Blue	
8	L		RangePop	532813.9	3662.1	5872	3932	[0]	White	
11	L	Initialize System Memory	RangePush	532825.8	322879.2	5872	3932	[0]	White	
11	L	Initialize System Memory	RangePop	855705.0	322879.2	5872	3932	[0]	White	
12	L	Copy System Memory to Device	RangePush	855718.1	43005.9	5872	3932	[2] Memory Transfer	Orange	
12	L	Copy System Memory to Device	RangePop	898724.0	43005.9	5872	3932	[2] Memory Transfer	Orange	
13	L		RangePush	898735.2	1599.0	5872	3932	[0]	White	
14		Memory Object C	RangeStart	898747.5	222902.4	5872	3932	[3] Memory Object Lifetime	Blue	
13	L		RangePop	900334.2	1599.0	5872	3932	[0]	White	
15	L	Allocate System Memory (Output)	RangePush	900344.7	20272.8	5872	3932	[1] Memory Allocation	Red	16777216
15	L	Allocate System Memory (Output)	RangePop	920617.5	20272.8	5872	3932	[1] Memory Allocation	Red	16777216
6	L	Initialize Memory	RangePop	920631.0	437531.1	5872	3932	[0]	Dark Red	
3	L	Initialize	RangePop	920641.9	912401.5	5872	3932	[0]	White	
9		Memory Object A	RangeEnd	1120843.2	591677.7	5872	3932	[3] Memory Object Lifetime	Blue	
10		Memory Object B	RangeEnd	1121198.1	590164.6	5872	3932	[3] Memory Object Lifetime	Blue	
14		Memory Object C	RangeEnd	1121649.9	222902.4	5872	3932	[3] Memory Object Lifetime	Blue	

[-] Processes	
[-] matrixMulDrv.exe [5872]	
[-] MAIN_THREAD [3932]	
[-] Tools Extension	
[0]	Markers
[3] Memory Object Lifetime	Ranges 0 Ranges 1 Ranges 2 Memory Object A Memory Object B Memory Object C

Grouping

```
nvtxNameCategory( uint32_t category, const char* name );
```

Category	
[2]	Memory Transfer
[2]	Memory Transfer
	[0]
[3]	Memory Object Lifetime
	[0]
[1]	Memory Allocation
[1]	Memory Allocation
	[0]
	[0]

[-] Processes	
[-] matrixMulDrv.exe [5872]	
+ MAIN_THREAD [3932]	
[-] Tools Extension	
[0]	
[3] Memory Object Lifetime	

OS Thread Naming

```
nvtxNameOsThread(GetCurrentThreadId(), "MAIN_THREAD");
```

Without...

[-] matrixMulDrv.exe [2852]	
[-] Thread [1660]	
Function Calls	

Process ID	Thread Name	Thread ID
2852		1660
2852		1660
2852		1660
2852		1660
2852		1660
2852		1660

With...

[-] matrixMulDrv.exe [5872]	
[-] MAIN_THREAD [3932]	
[+] Tools Extension	

Process ID	Thread Name	Thread ID
4876	MAIN_THREAD	5212
4876	MAIN_THREAD	5212
4876	MAIN_THREAD	5212
4876	MAIN_THREAD	5212
4876	MAIN_THREAD	5212

CUDA Resource Naming

```
void nvtxNameCuDeviceA( CUdevice  device, const char * name );
```

```
void nvtxNameCuContextA( CUcontext  context, const char * name );
```

```
void nvtxNameCuStreamA( CUstream  stream, const char * name );
```


CUDA Resource Naming

```
CUresult status = cuCtxCreate( &cuContext, 0, cuDevice );
if (CUDA_SUCCESS != status) goto Error;
nvtxNameCuContext( cuContext, "CTX_NAME" );
```

CUDA Device ID	Device	Contexts	Device%	H to D (bytes)	D to H (bytes)
[0] DEVICE_NAME	GPU 0 - GeForce GTX 470	1	13.0	33,554,432	16,777,216

CUDA Contexts

	Total	No Context	1 CTX_NAME
--	--------------	-------------------	--------------------------

Cuda Device ID [Address]	[0] DEVICE_NAME [0x0]
Device	GPU 0 - GeForce GTX 470
Process	matrixMulDrv.exe [4876]
MAX_THREADS_PER_BLOCK	1024
MAX_BLOCK_DIM_X	1024
MAX_BLOCK_DIM_Y	1024

Function Name	Module ID	Function ID	Count	Device %	Device Time (μs)	Min (μs)	Avg (μs)	Max (μs)	Context Name	Context ID	Signature	Process ID
matrixMul	4	1	1	13.03	110261.2	110261.2	110261.2	110261.2	CTX_NAME	1	matrixMul	4876

[-] CUDA	
[-] Context 0	
Driver API	⌵
[-] Context 1 CTX_NAME [0] DEVICE_NAME	
Driver API	⌵
Memory	⌵

OpenCL Resource Naming

- Device
- Context
- Command Queues
- Memobjects
- Programs
- Samplers
- Events

OpenCL	
API	
Context [1] <<CONTEXT>> [[57700176] <<DEVICE>>]	
API	
Queue [1] <<QUEUE>>[57700176] <<DEVICE>>	
Kernel Commands [468]	
Memory Commands [468]	

Program Name	Program ID	Actions	Creation Time (μs)	Lifetime (μs)	# Devices	Device IDs	# Builds	# Kernels	Kernel IDs	Context Name	Context ID	Process Name	Process ID
<<PROGRAM>>	1		153912.2	10896708.0	1	0x3706F50	1	1	1	<<CONTEXT>>	1	oclSimpleGL.exe	5128

...Beyond Nsight 1.5

- TCC Profiling support
- More complex experiments and analysis
- Source code correlation for certain performance bottlenecks
- Additional Resource Naming - OpenCL, CUDA-C, OpenGL...

Conclusion

- Macro and micro level performance analysis tools
- Nvtx to help understand what is the application really doing
- Nvtx for using Nsight more efficiently
- CUDA Profiling live with dynamic experimentation

NVIDIA Parallel Nsight™ at GTC 2010

Parallel Nsight Lounge by Microsoft (Ballroom Concourse)

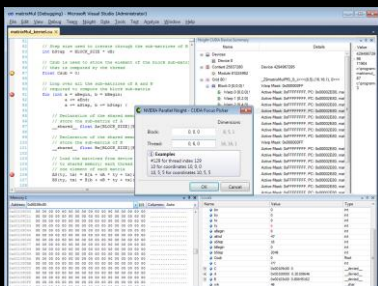
From 10am-8pm each day, give-a-ways daily at 3pm

All Parallel Nsight Sessions at GTC are in Room B

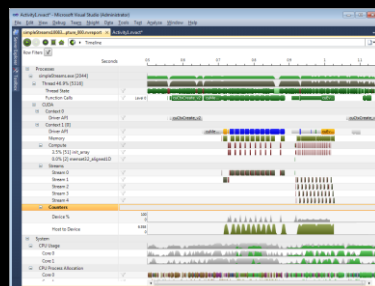
Tues, 5-5:50pm: Parallel Nsight for Accelerated DirectX 11 Development



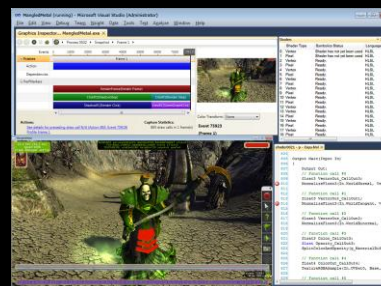
www.nvidia.com/ParallelNsight



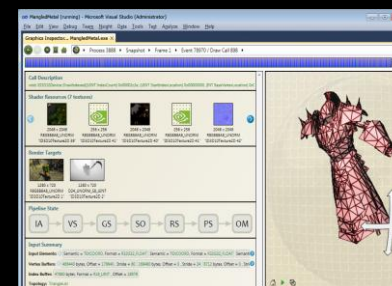
Compute Debugger



System Analysis



Graphics Debugger



Graphics Inspector