HPC Programming

Requirements

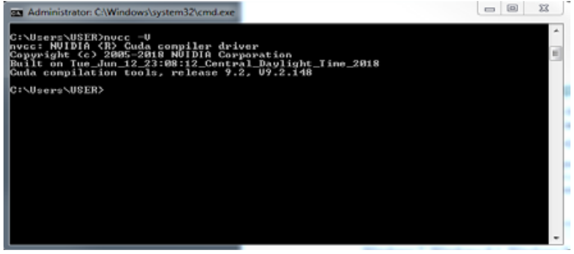
1). NVIDIA QUADRO Graphic Card , Graphic card used in this document is Quadro K5000
2). Download the Cuda toolkit from : https://developer.nvidia.com/cuda-toolkit-70
3). Visual Studio IDE

Introduction

CUDA is a parallel computing platform and programming model invented by NVIDIA. It enables dramatic increases in computing performance by harnessing the power of the graphics processing unit (GPU).

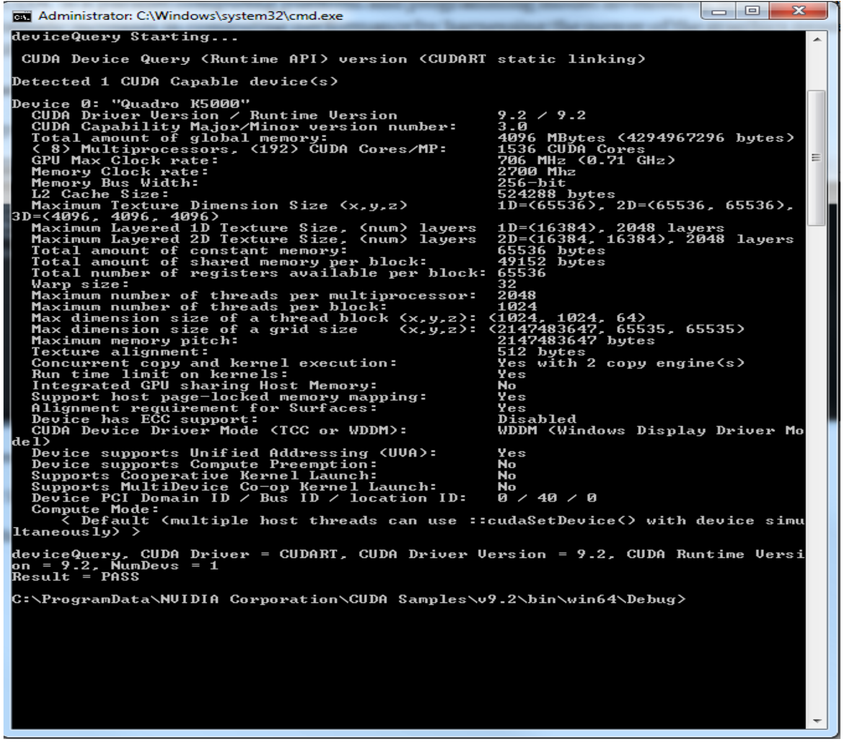
Test installation

Open command prompt and type nvcc -V should give the version that has been installed.



```
C:\Users\USER>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2018 NVIDIA Corporation
Built on Tue_Jun_12_23:08:12_Central_Daylight_Time_2018
Cuda compilation tools, release 9.2, V9.2.148

C:\Users\USER>
```

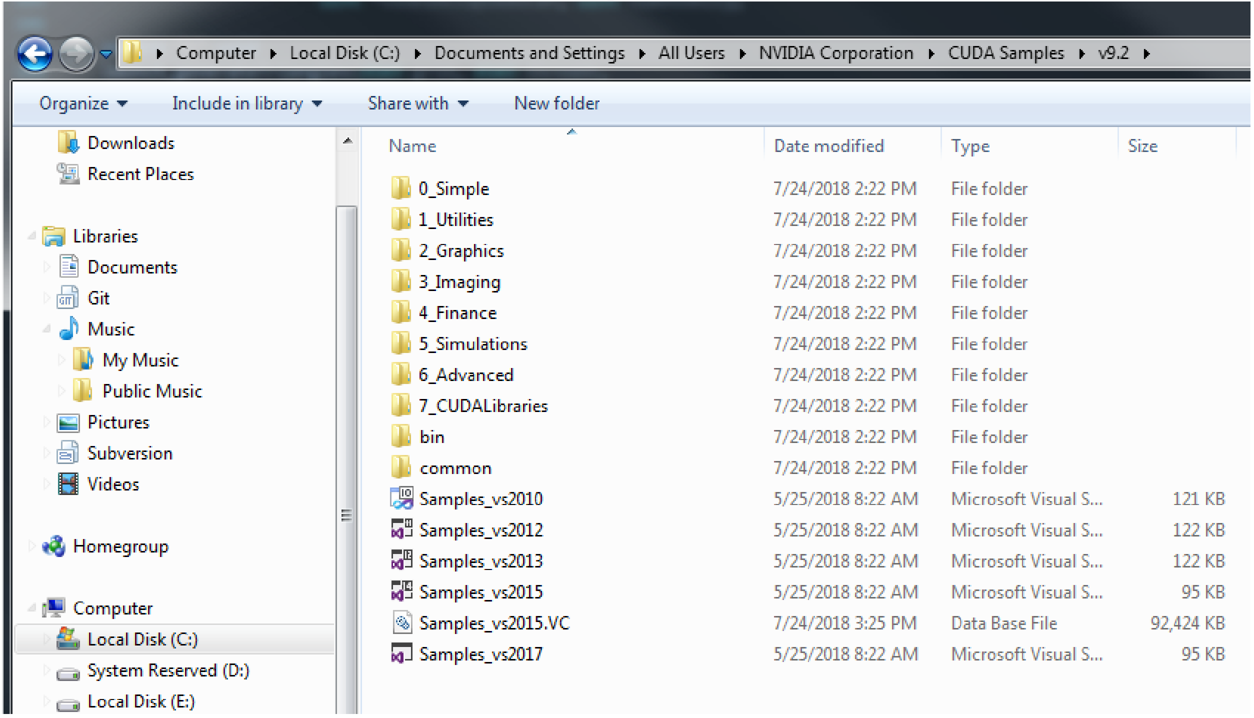
Test correct installation by navigating to :
C:\ProgramData\NVIDIA Corporation\CUDA Samples\v9.2\bin\win64\Debug
Execute deviceQuery
Which should give the output :



```
deviceQuery Starting...

 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: "Quadro K5000"
  CUDA Driver Version / Runtime Version          9.2 / 9.2
  CUDA Capability Major/Minor version number:    3.0
  Total amount of global memory:                 4096 MBytes (4294967296 bytes)
  ( 8) Multiprocessors, (192) CUDA Cores/MP:     1536 CUDA Cores
  GPU Max Clock rate:                            706 MHz (0.71 GHz)
  Memory Clock rate:                             2700 Mhz
  Memory Bus Width:                              256-bit
  L2 Cache Size:                                 524288 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(65536), 2D=(65536, 65536),
3D=(4096, 4096, 4096)
  Maximum Layered 1D Texture Size, (num) layers  1D=(16384), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(16384, 16384), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  2048
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 2 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  CUDA Device Driver Mode (TCC or WDDM):         WDDM (Windows Display Driver Mo
del)
  Device supports Unified Addressing (UVA):      Yes
  Device supports Compute Preemption:            No
  Supports Cooperative Kernel Launch:            No
  Supports MultiDevice Co-op Kernel Launch:      No
  Device PCI Domain ID / Bus ID / location ID:   0 / 40 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simu
ltaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 9.2, CUDA Runtime Versi
on = 9.2, NumDevs = 1
Result = PASS

C:\ProgramData\NVIDIA Corporation\CUDA Samples\v9.2\bin\win64\Debug>
```

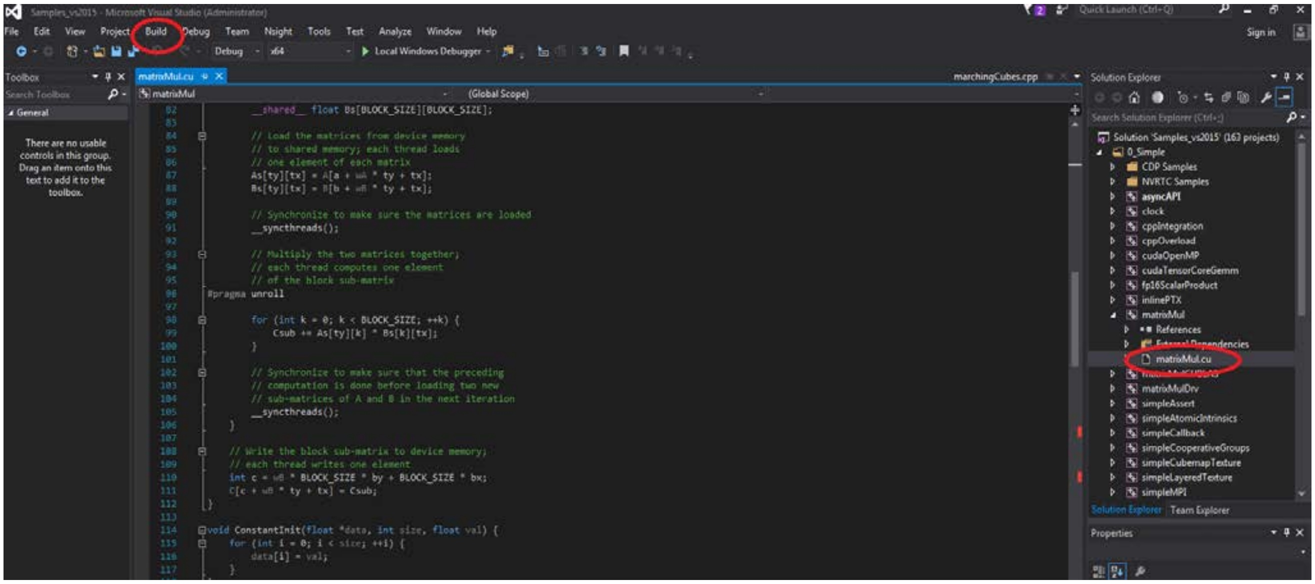
Developing using CUDA toolkit

1). Open the Visual studio IDE (I am using Visual Studio 2015)
Under the CUDA Samples are the solutions for samples that are installed with the toolkit:



Name	Date modified	Type	Size
0_Simple	7/24/2018 2:22 PM	File folder	
1_Utilities	7/24/2018 2:22 PM	File folder	
2_Graphics	7/24/2018 2:22 PM	File folder	
3_Imaging	7/24/2018 2:22 PM	File folder	
4_Finance	7/24/2018 2:22 PM	File folder	
5_Simulations	7/24/2018 2:22 PM	File folder	
6_Advanced	7/24/2018 2:22 PM	File folder	
7_CUDALibraries	7/24/2018 2:22 PM	File folder	
bin	7/24/2018 2:22 PM	File folder	
common	7/24/2018 2:22 PM	File folder	
Samples_vs2010	5/25/2018 8:22 AM	Microsoft Visual S...	121 KB
Samples_vs2012	5/25/2018 8:22 AM	Microsoft Visual S...	122 KB
Samples_vs2013	5/25/2018 8:22 AM	Microsoft Visual S...	122 KB
Samples_vs2015	5/25/2018 8:22 AM	Microsoft Visual S...	95 KB
Samples_vs2015.VC	7/24/2018 3:25 PM	Data Base File	92,424 KB
Samples_vs2017	5/25/2018 8:22 AM	Microsoft Visual S...	95 KB

2). Open the solution as per the Visual Studio IDE installed on your workstation (Samples_vs2015.VC)
3). Use the build to compile the solution as shown.



```
        __shared__ float Bs[BLOCK_SIZE][BLOCK_SIZE];

        // Load the matrices from device memory
        // to shared memory; each thread loads
        // one element of each matrix
        As[ty][tx] = A[a + wA * ty + tx];
        Bs[ty][tx] = B[b + wB * ty + tx];

        // Synchronize to make sure the matrices are loaded
        __syncthreads();

        // Multiply the two matrices together;
        // each thread computes one element
        // of the block sub-matrix
#pragma unroll

        for (int k = 0; k < BLOCK_SIZE; ++k) {
            Csub += As[ty][k] * Bs[k][tx];
        }

        // Synchronize to make sure that the preceding
        // computation is done before loading two new
        // sub-matrices of A and B in the next iteration
        __syncthreads();
    }

    // Write the block sub-matrix to device memory;
    // each thread writes one element
    int c = wB * BLOCK_SIZE * by + BLOCK_SIZE * bx;
    C[c + wB * ty + tx] = Csub;
}

void ConstantInit(float *data, int size, float val) {
    for (int i = 0; i < size; ++i) {
        data[i] = val;
    }
```

4). After compiling

Go to the debug directory of the solution and run i.e

C:\Documents and Settings\All Users\NVIDIA Corporation\CUDA Samples\v9.2\0_Simple\matrixMul\x64\Debug

(I am trying the sample on matrix multiplication)

5). Use the command prompt (Startmenu > CMD)to cd to file matrixMul.cu on the Debug directory

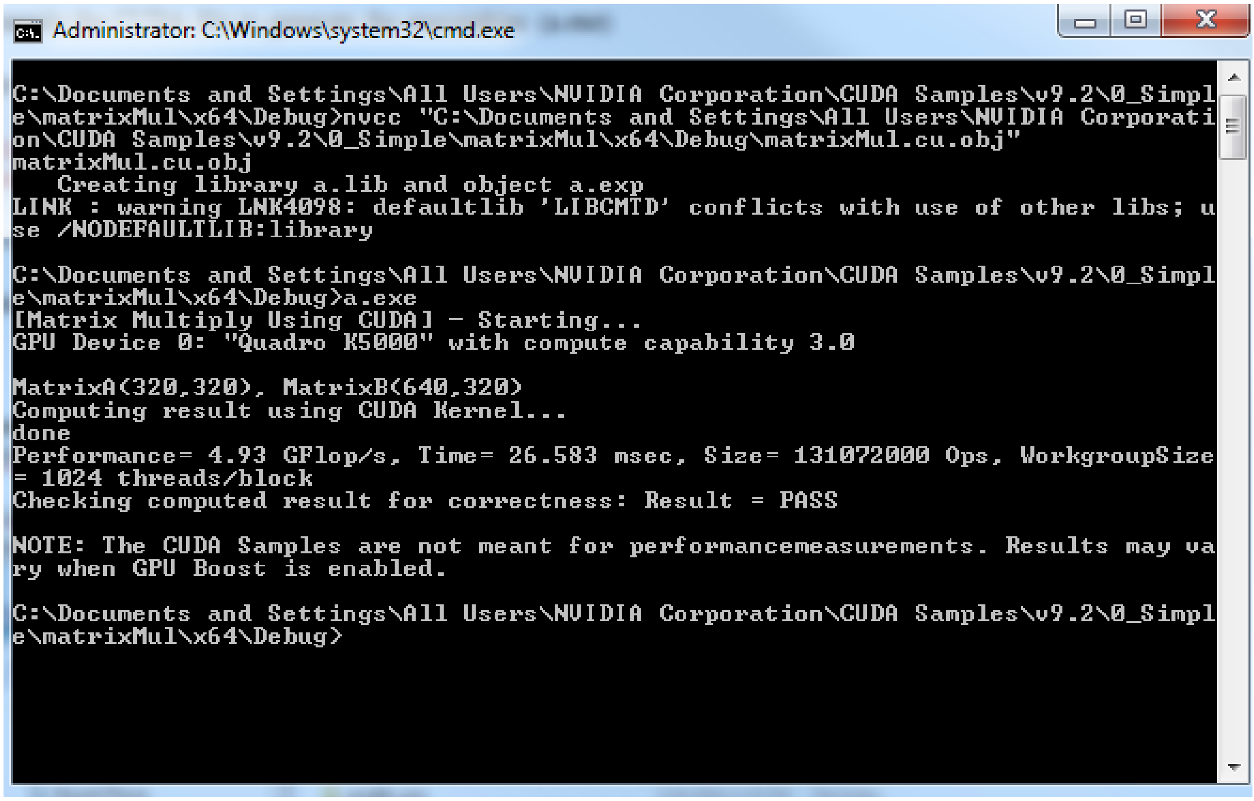
6). Compile the CUDA file to generate the executables (a.exe)

The syntax to compile is :

nvcc " C:\Documents and Settings\All Users\NVIDIA Corporation\CUDA
Samples\v9.2\0_Simple\matrixMul\x64\Debug\ matrixMul.cu"

7). It will create the filed a.exe

Run a.exe to give you the output as shown:



```
C:\Documents and Settings\All Users\NVIDIA Corporation\CUDA Samples\v9.2\0_Simpl
e\matrixMul\x64\Debug>nvcc "C:\Documents and Settings\All Users\NVIDIA Corporati
on\CUDA Samples\v9.2\0_Simple\matrixMul\x64\Debug\matrixMul.cu.obj"
matrixMul.cu.obj
   Creating library a.lib and object a.exp
LINK : warning LNK4098: defaultlib 'LIBCMTD' conflicts with use of other libs; u
se /NODEFAULTLIB:library

C:\Documents and Settings\All Users\NVIDIA Corporation\CUDA Samples\v9.2\0_Simpl
e\matrixMul\x64\Debug>a.exe
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Quadro K5000" with compute capability 3.0

MatrixA(320,320), MatrixB(640,320)
Computing result using CUDA Kernel...
done
Performance= 4.93 GFlop/s, Time= 26.583 msec, Size= 131072000 Ops, WorkgroupSize
= 1024 threads/block
Checking computed result for correctness: Result = PASS

NOTE: The CUDA Samples are not meant for performancemeasurements. Results may va
ry when GPU Boost is enabled.

C:\Documents and Settings\All Users\NVIDIA Corporation\CUDA Samples\v9.2\0_Simpl
e\matrixMul\x64\Debug>
```

8.) In this case it is showing my Graphic card **Quadro K5000** , and the computation done by the GPU.

Task One - Create a CUDA code to read computation power:

Steps

1). Open visual studio > New Project > Cuda Project > Enter Name of the project

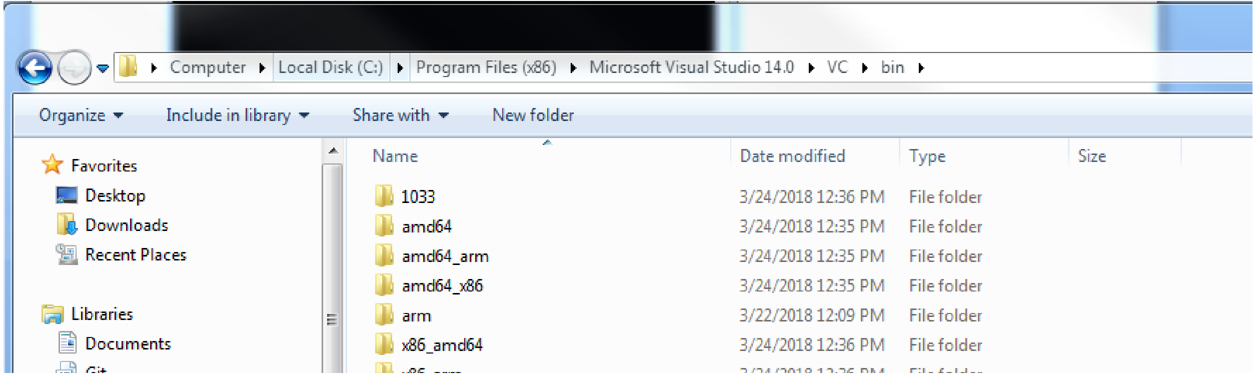
2). Type the code on the .cu file as below:

Errors encountered:

```
> nvcc -o a.out matrixMul.cu
```

1). nvcc fatal : Cannot find compiler 'cl.exe' in PATH

solution add the path:



[redacted]

Computer ▸ Local Disk (C:) ▸ Program Files (x86) ▸ Microsoft Visual Studio 14.0 ▸ VC ▸ bin ▸

Name	Date modified	Type	Size
1033	3/24/2018 12:36 PM	File folder	
amd64	3/24/2018 12:35 PM	File folder	
amd64_arm	3/24/2018 12:35 PM	File folder	
amd64_x86	3/24/2018 12:35 PM	File folder	
arm	3/22/2018 12:09 PM	File folder	
x86_amd64	3/24/2018 12:36 PM	File folder	
x86_arm	3/24/2018 12:36 PM	File folder	

to windows path environment.

cl is the c/c++ compiler as located in your linux or windows workstation

Programming using CUDA toolkit

1).Start Visual basic 2015.
2).New project
3). Select CUDA 9.2
4).This will create a file " .cu " where you write GPU specific code

Quick start on a simple hello world to execute on GPU

```
// includes, system
#include <stdio.h>
// includes CUDA Runtime
#include <cuda_runtime.h>
__global__
void kernel(void) {
}
int main(void) {
       kernel << <1, 1 >> >();
       printf("Hello, World!\n");
       getchar();
}
```

CUDA C keyword __global__ indicates that a function.
nvcc splits source file into host and device components — NVIDIA’s compiler handles device functions like kernel() — Standard host compiler handles host functions like main()

Memory management on CUDA programs

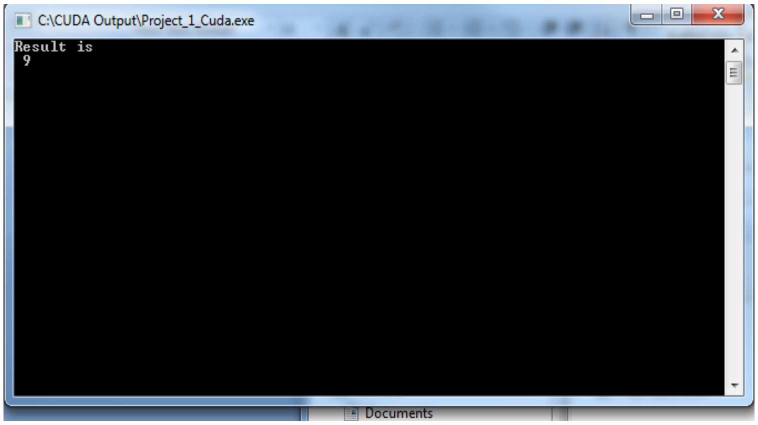
Host and device memory are distinct entities, where
- Device pointers point to GPU memory which may be passed to and from host code and may not be dereferenced from host code.
- Host pointers point to CPU memory , may be passed to and from device code and may not be dereferenced from device code.

CUDA API for dealing with device memory :
cudaMalloc(), cudaFree(), cudaMemcpy()

I will illustrate with the simple addition code below:

```
// includes, system
#include <stdio.h>
// includes CUDA Runtime
#include <cuda_runtime.h>
__global__
void add(int *a, int *b, int *c) {
       *c = *a + *b;
}
int main(void) {
       //host copies of a, b, c
       int a, b, c;
    //device copies of a, b, c
       int *dev_a, *dev_b, *dev_c;
    //we need space for an integer
       int size = sizeof(int);
    //allocate device copies of a, b, c
       cudaMalloc((void**)&dev_a, size);
       cudaMalloc((void**)&dev_b, size);
       cudaMalloc((void**)&dev_c, size);
       a = 2;
       b = 7;
       //copy inputs to device
       cudaMemcpy(dev_a, &a, size, cudaMemcpyHostToDevice);
       cudaMemcpy(dev_b, &b, size, cudaMemcpyHostToDevice);
       //launch add() kernel on GPU, passing parameters
       add <<< 1, 1 >>>(dev_a, dev_b, dev_c);
       //copy device result back to host copy of c
       cudaMemcpy(&c, dev_c, size, cudaMemcpyDeviceToHost);
       cudaFree(dev_a);
       cudaFree(dev_b);
       cudaFree(dev_c);
       printf("Result is \n %d " , c);
       getchar();
       return 0;
}
```

Output above will be as below :



```
Result is
 9
```

Parallel programming on CUDA programs

To be done

3d Max Rendering

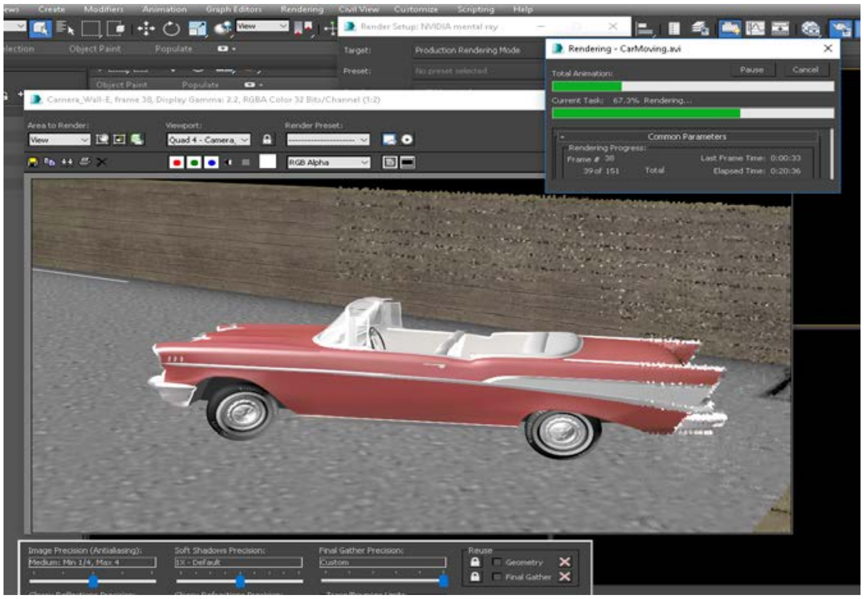
Introduction

This is aimed to make use of the GPU in rendering animation. NVIDIA Tesla K20Xm will be used.

Test 1.

Rendering a car animation of 151 frames on dx360 M4 Server (20CPUs,65536MB RAM)on NVIDIA Tesla K20Xm.

a). Normal 3d max render time is as shown below :



Render Setup: NVIDIA mental ray

Target: Production Rendering Mode

Preset: No preset selected

Rendering - CarMoving.avi

Total Animation:

Current Task: 67.3% Rendering...

Rendering Progress:

Frame # 38

39 of 151 Total

Last Frame Time: 0:00:33

Elapsed Time: 0:20:36

Camera_Wall-E, frame 38, Display Gamma: 2.2, RGBA Color 32 Bits/Channel (1:2)

Area to Render: View

Viewport: Quad 4 - Camera

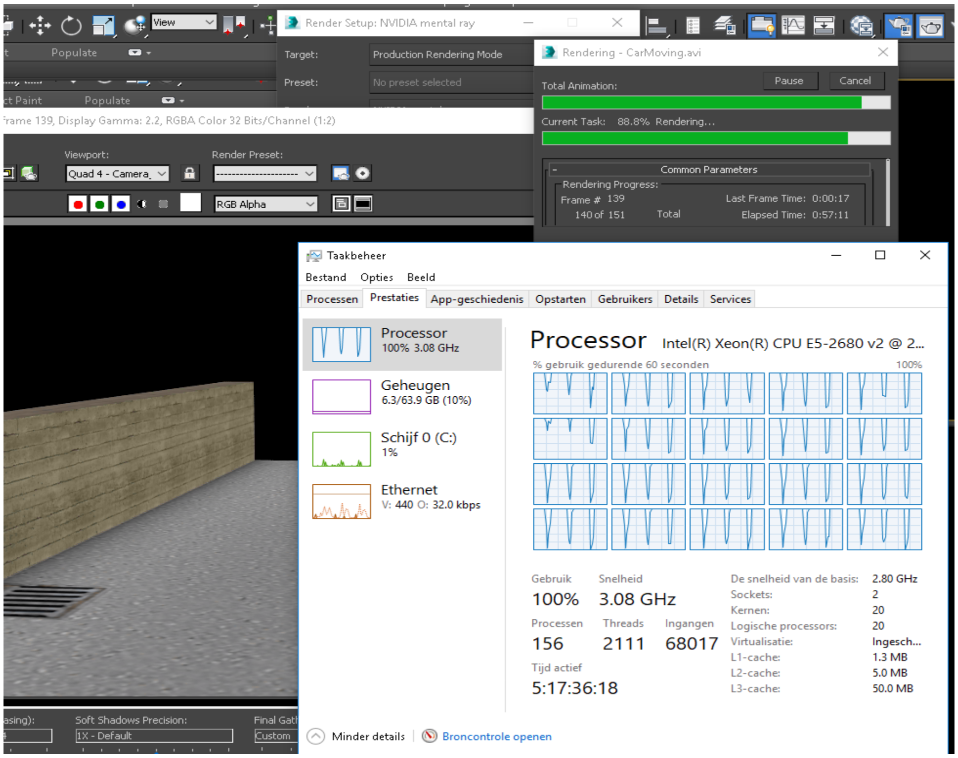
RGB Alpha

Image Precision (Antialiasing): Medium: Min 1/4, Max 4

Soft Shadows Precision: 1X - Default

Final Gather Precision: Custom

The processor statistics of one of the dx360 M4 Server node as shown below (rendering via 3dmax):



Render Setup: NVIDIA mental ray

Target: Production Rendering Mode

Preset: No preset selected

Frame 139, Display Gamma: 2.2, RGBA Color 32 Bits/Channel (1:2)

Viewport: Quad 4 - Camera_

Render Preset: --------------------

RGB Alpha

Soft Shadows Precision: 1X - Default

Final Gath: Custom

Rendering - CarMoving.avi

Total Animation:

Current Task: 88.8% Rendering...

Rendering Progress:

Frame # 139 Last Frame Time: 0:00:17

140 of 151 Total Elapsed Time: 0:57:11

Taakbeheer

Processor 100% 3.08 GHz

Geheugen 6.3/63.9 GB (10%)

Schijf 0 (C:) 1%

Ethernet V: 440 O: 32.0 kbps

Processor Intel(R) Xeon(R) CPU E5-2680 v2 @ 2...

% gebruik gedurende 60 seconden 100%

Gebruik 100%

Snelheid 3.08 GHz

Processen 156

Threads 2111

Ingangen 68017

Tijd actief 5:17:36:18

De snelheid van de basis: 2.80 GHz

Sockets: 2

Kernen: 20

Logische processors: 20

Virtualisatie: Ingesch...

L1-cache: 1.3 MB

L2-cache: 5.0 MB

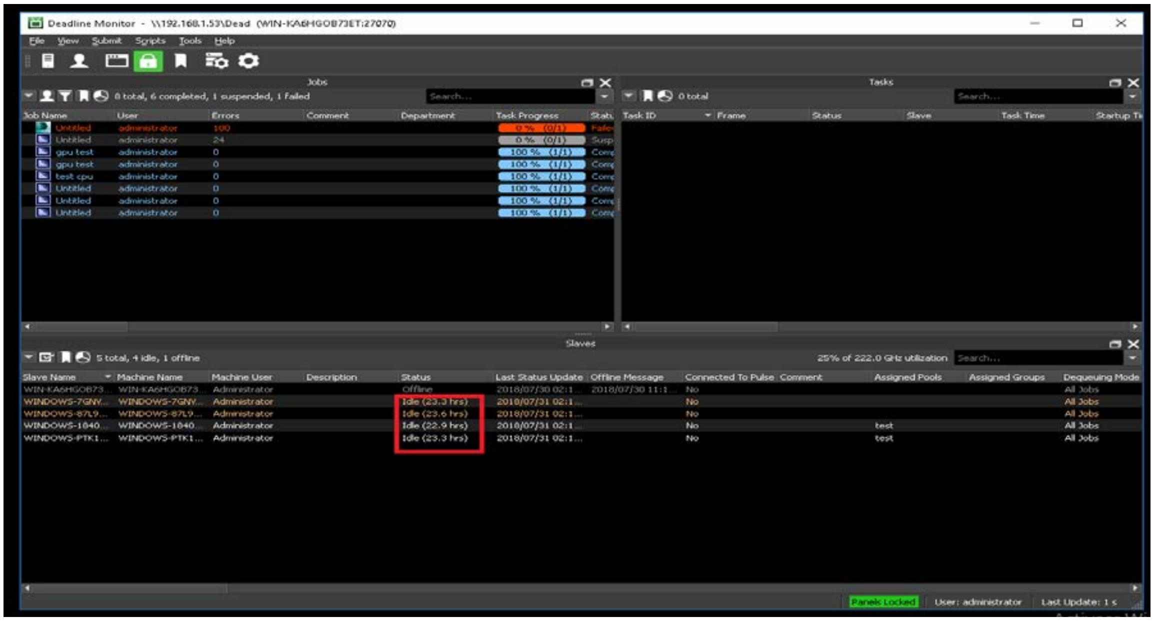
L3-cache: 50.0 MB

The time taken to render above is :

58minutes , the resulting file a **17.3 MB** video file.

b). Deadline rendering of same animation.

The same animation will now be submitted to the render farm with the four slaves.
All the slaves are as shown on the picture below:



Deadline Monitor - \\192.168.1.53\Dead (WIN-KA6HGOB73ET:27070)

Jobs — 8 total, 6 completed, 1 suspended, 1 failed

Job Name	User	Errors	Comment	Department	Task Progress	Statu
Untitled	administrator	100			0 % (0/1)	Faile
Untitled	administrator	24			0 % (0/1)	Susp
gpu test	administrator	0			100 % (1/1)	Comp
gpu test	administrator	0			100 % (1/1)	Comp
test cpu	administrator	0			100 % (1/1)	Comp
Untitled	administrator	0			100 % (1/1)	Comp
Untitled	administrator	0			100 % (1/1)	Comp
Untitled	administrator	0			100 % (1/1)	Comp

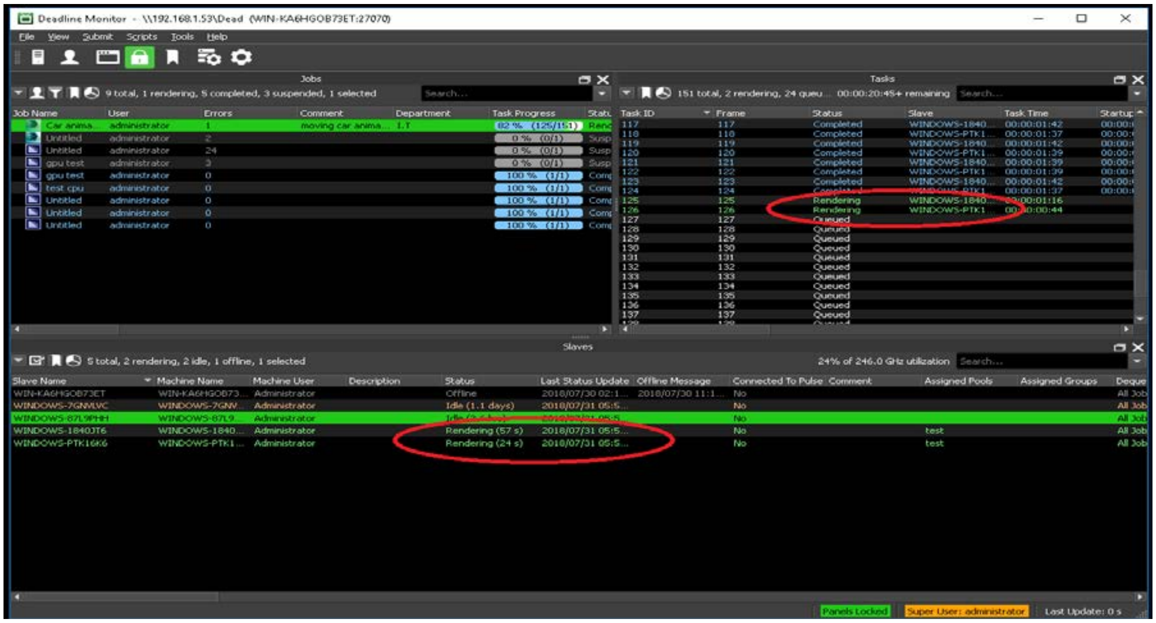
Tasks — 0 total

Task ID	Frame	Status	Slave	Task Time	Startup Ti

Slaves — 5 total, 4 idle, 1 offline — 25% of 222.0 GHz utilization

Slave Name	Machine Name	Machine User	Description	Status	Last Status Update	Offline Message	Connected To Pulse	Comment	Assigned Pools	Assigned Groups	Dequeuing Mode
WIN-KA6HGOB73...	WIN-KA6HGOB73...	Administrator		Offline	2018/07/30 02:1...	2018/07/30 11:1...	No				All Jobs
WINDOWS-7GNV...	WINDOWS-7GNV...	Administrator		Idle (23.3 hrs)	2018/07/31 02:1...		No				All Jobs
WINDOWS-87L9...	WINDOWS-87L9...	Administrator		Idle (23.6 hrs)	2018/07/31 02:1...		No				All Jobs
WINDOWS-1840...	WINDOWS-1840...	Administrator		Idle (22.9 hrs)	2018/07/31 02:1...		No		test		All Jobs
WINDOWS-PTK1...	WINDOWS-PTK1...	Administrator		Idle (23.3 hrs)	2018/07/31 02:1...		No		test		All Jobs

Submitted 3dmax job is as shown:



Deadline Monitor - \\192.168.1.53\Dead (WIN-KA6HGOB73ET:27070)

Jobs — 9 total, 1 rendering, 5 completed, 3 suspended, 1 selected

Job Name	User	Errors	Comment	Department	Task Progress	Statu
Car anima...	administrator	1	moving car anima...	I.T	82 % (125/151)	Renc
Untitled	administrator	2			0 % (0/1)	Susp
Untitled	administrator	24			0 % (0/1)	Susp
gpu test	administrator	3			0 % (0/1)	Susp
gpu test	administrator	0			100 % (1/1)	Comp
test cpu	administrator	0			100 % (1/1)	Comp
Untitled	administrator	0			100 % (1/1)	Comp
Untitled	administrator	0			100 % (1/1)	Comp
Untitled	administrator	0			100 % (1/1)	Comp

Tasks — 151 total, 2 rendering, 24 queu... 00:00:20:45+ remaining

Task ID	Frame	Status	Slave	Task Time	Startup
117	117	Completed	WINDOWS-1840...	00:00:01:42	00:00:
118	118	Completed	WINDOWS-PTK1...	00:00:01:37	00:00:
119	119	Completed	WINDOWS-1840...	00:00:01:42	00:00:
120	120	Completed	WINDOWS-PTK1...	00:00:01:39	00:00:
121	121	Completed	WINDOWS-1840...	00:00:01:39	00:00:
122	122	Completed	WINDOWS-PTK1...	00:00:01:39	00:00:
123	123	Completed	WINDOWS-1840...	00:00:01:42	00:00:
124	124	Completed	WINDOWS-PTK1...	00:00:01:37	00:00:
125	125	Rendering	WINDOWS-1840...	00:00:01:16	
126	126	Rendering	WINDOWS-PTK1...	00:00:00:44	
127	127	Queued			
128	128	Queued			
129	129	Queued			
130	130	Queued			
131	131	Queued			
132	132	Queued			
133	133	Queued			
134	134	Queued			
135	135	Queued			
136	136	Queued			
137	137	Queued			

Slaves — 5 total, 2 rendering, 2 idle, 1 offline, 1 selected — 24% of 246.0 GHz utilization

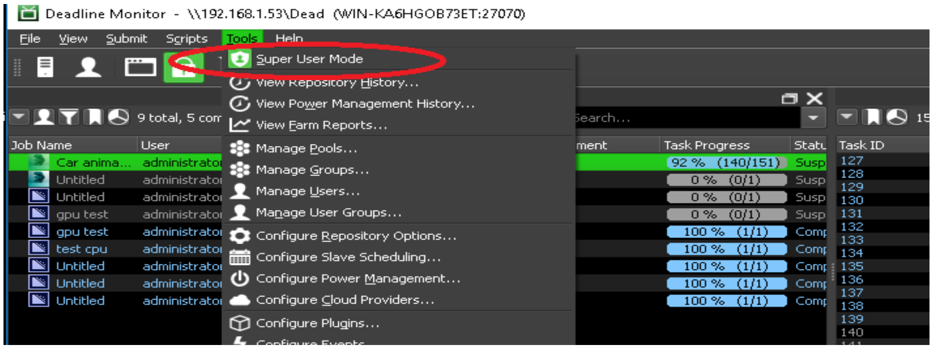
Slave Name	Machine Name	Machine User	Description	Status	Last Status Update	Offline Message	Connected To Pulse	Comment	Assigned Pools	Assigned Groups	Deque
WIN-KA6HGOB73ET	WIN-KA6HGOB73...	Administrator		Offline	2018/07/30 02:1...	2018/07/30 11:1...	No				All Job
WINDOWS-7GNMLVC	WINDOWS-7GNV...	Administrator		Idle (1.1 days)	2018/07/31 05:5...		No				All Job
WINDOWS-87L9PHH	WINDOWS-87L9...	Administrator		Idle (2.6 hrs)	2018/07/31 05:5...		No				All Job
WINDOWS-1840JT6	WINDOWS-1840...	Administrator		Rendering (57 s)	2018/07/31 05:5...		No		test		All Job
WINDOWS-PTK16K6	WINDOWS-PTK1...	Administrator		Rendering (24 s)	2018/07/31 05:5...		No		test		All Job

NB: Two nodes are the only being scheduled due to license

Configuring render farm to use the NVIDIA Tesla K20Xm

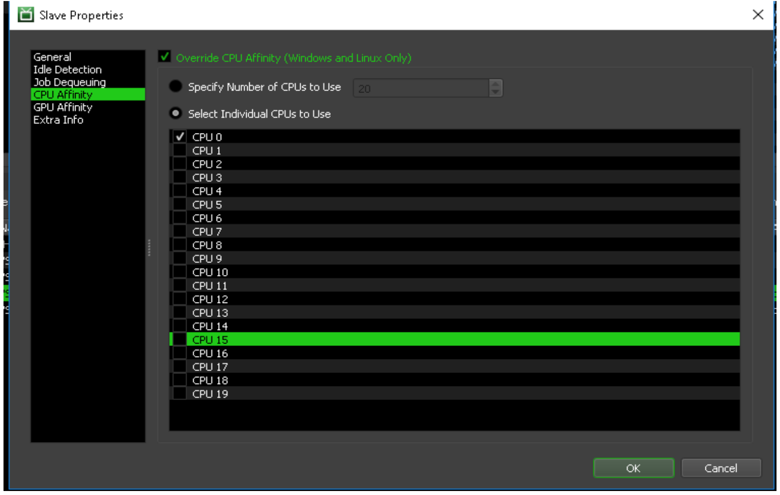
Below are the screenshots of the configuration:

1). Ensure you are on super user mode.



Job Name	User	ment	Task Progress	Statu
Car anima...	administrato		92 % (140/151)	Susp
Untitled	administrato		0 % (0/1)	Susp
Untitled	administrato		0 % (0/1)	Susp
gpu test	administrato		0 % (0/1)	Susp
gpu test	administrato		100 % (1/1)	Comp
test cpu	administrato		100 % (1/1)	Comp
Untitled	administrato		100 % (1/1)	Comp
Untitled	administrato		100 % (1/1)	Comp
Untitled	administrato		100 % (1/1)	Comp

2).Right click on slave properties > then go to CPU Affinity, unselect all cpus



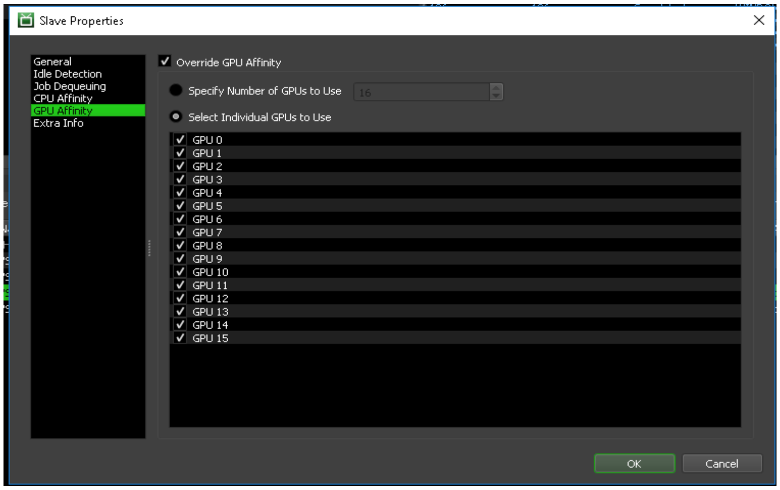
☑ Override CPU Affinity (Windows and Linux Only)

○ Specify Number of CPUs to Use 20

● Select Individual CPUs to Use

☑ CPU 0
□ CPU 1
□ CPU 2
□ CPU 3
□ CPU 4
□ CPU 5
□ CPU 6
□ CPU 7
□ CPU 8
□ CPU 9
□ CPU 10
□ CPU 11
□ CPU 12
□ CPU 13
□ CPU 14
□ CPU 15
□ CPU 16
□ CPU 17
□ CPU 18
□ CPU 19

3).Go to GPU Affinity and select the GPU's of that slave that you want to use on the slave render as shown below:



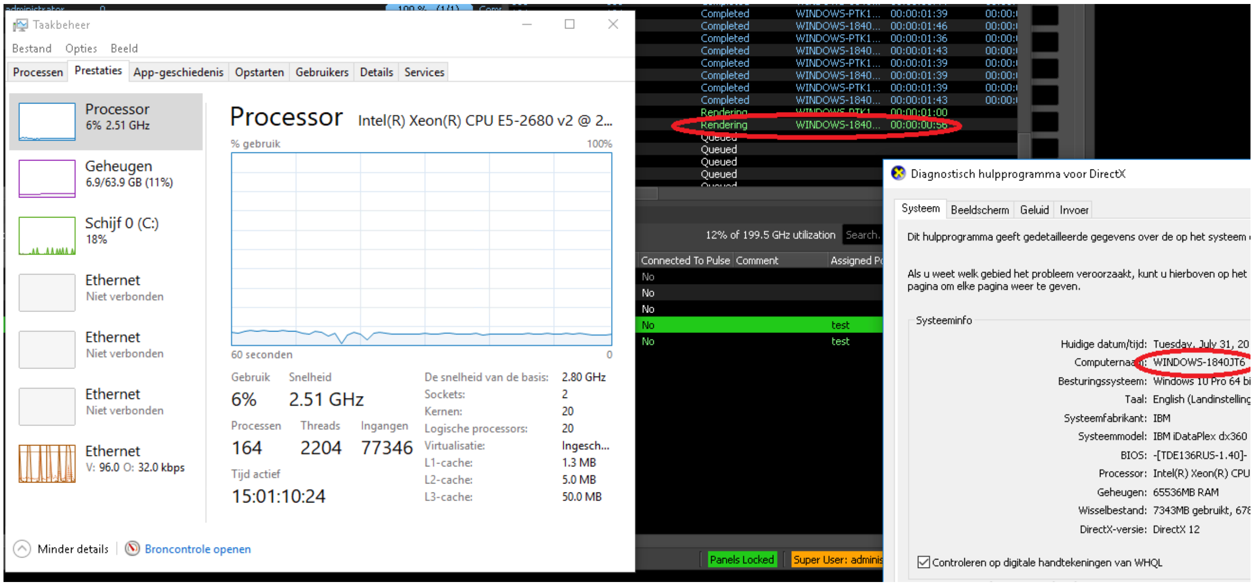
☑ Override GPU Affinity

○ Specify Number of GPUs to Use 16

● Select Individual GPUs to Use

☑ GPU 0
☑ GPU 1
☑ GPU 2
☑ GPU 3
☑ GPU 4
☑ GPU 5
☑ GPU 6
☑ GPU 7
☑ GPU 8
☑ GPU 9
☑ GPU 10
☑ GPU 11
☑ GPU 12
☑ GPU 13
☑ GPU 14
☑ GPU 15

Comparing with the normal CPU render , below is the screenshot of the processor, on one of the m4server nodes.



Processor 6% 2.51 GHz

Geheugen 6.9/63.9 GB (11%)

Schijf 0 (C:) 18%

Ethernet Niet verbonden

Ethernet Niet verbonden

Ethernet Niet verbonden

Ethernet V: 96.0 O: 32.0 kbps

Processor Intel(R) Xeon(R) CPU E5-2680 v2 @ 2...

Gebruik	Snelheid	
6%	2.51 GHz	
Processen	Threads	Ingangen
164	2204	77346

Tijd actief 15:01:10:24

De snelheid van de basis:	2.80 GHz
Sockets:	2
Kernen:	20
Logische processors:	20
Virtualisatie:	Ingesch...
L1-cache:	1.3 MB
L2-cache:	5.0 MB
L3-cache:	50.0 MB

Completed	WINDOWS-PTK1...	00:00:01:39	00:00:
Completed	WINDOWS-1840...	00:00:01:46	00:00:
Completed	WINDOWS-PTK1...	00:00:01:36	00:00:
Completed	WINDOWS-1840...	00:00:01:43	00:00:
Completed	WINDOWS-PTK1...	00:00:01:39	00:00:
Completed	WINDOWS-1840...	00:00:01:39	00:00:
Completed	WINDOWS-PTK1...	00:00:01:39	00:00:
Completed	WINDOWS-1840...	00:00:01:43	00:00:
Rendering	WINDOWS-PTK1...	00:00:01:00	
Rendering	WINDOWS-1840...	00:00:00:56	
Queued			
Queued			
Queued			
Queued			

12% of 199.5 GHz utilization

Connected To Pulse	Comment	Assigned P
No		
No		
No		
No		test
No		test

Diagnostisch hulpprogramma voor DirectX

Dit hulpprogramma geeft gedetailleerde gegevens over de op het systeem

Als u weet welk gebied het probleem veroorzaakt, kunt u hierboven op het pagina om elke pagina weer te geven.

Systeeminfo

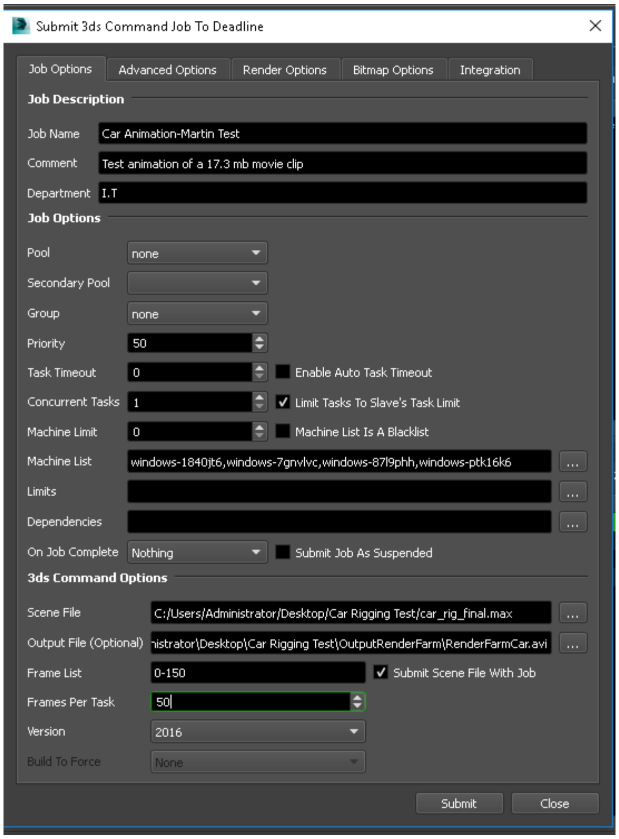
Huidige datum/tijd:	Tuesday, July 31, 20
Computernaam:	WINDOWS-1840JT6
Besturingssysteem:	Windows 10 Pro 64 bi
Taal:	English (Landinstelling
Systeemfabrikant:	IBM
Systeemmodel:	IBM iDataPlex dx360
BIOS:	-[TDE136RUS-1.40]-
Processor:	Intel(R) Xeon(R) CPU
Geheugen:	65536MB RAM
Wisselbestand:	7343MB gebruikt, 678
DirectX-versie:	DirectX 12

☑ Controleren op digitale handtekeningen van WHQL

Time taken to render the script is as shown below:

To be done.

Settings to submit the 3dmax file as shown below:



Job Description

Job Name: Car Animation-Martin Test

Comment: Test animation of a 17.3 mb movie clip

Department: I.T

Job Options

Pool: none

Secondary Pool:

Group: none

Priority: 50

Task Timeout: 0 □ Enable Auto Task Timeout

Concurrent Tasks: 1 ☑ Limit Tasks To Slave's Task Limit

Machine Limit: 0 □ Machine List Is A Blacklist

Machine List: windows-1840jt6,windows-7gnvlvc,windows-87l9phh,windows-ptk16k6

Limits:

Dependencies:

On Job Complete: Nothing □ Submit Job As Suspended

3ds Command Options

Scene File: C:/Users/Administrator/Desktop/Car Rigging Test/car_rig_final.max

Output File (Optional): nistrator\Desktop\Car Rigging Test\OutputRenderFarm\RenderFarmCar.avi

Frame List: 0-150 ☑ Submit Scene File With Job

Frames Per Task: 50

Version: 2016

Build To Force: None

c). Application of CUDA Programming on render farm

On progress

Reference:

https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html