**Assignment 7**

**Performance Modeling for Convolutions on GPU and Results**

**All were run on a single GPU**

Q1) How many Flops needs to be done to compute a convolution of dimension k on a image of size

m\*n?

1. Our Image size is m\*n and our kernel size which would be used for convolution is k\*k
2. In convolution, for a single pixel, we will do 2 times k\*k operations ie with our kernel we will do multiplication first and then addition.
3. So for an image with a size of m\*n, we will do a total of (m\*n\*k\*k\*2) flops.

Q2) ) How much memory needs to be moved to compute a convolution of dimension k on an image of

size n\*m?

* We will bring the image of size m\*n from memory also the kernel of size k\*k.
* So we will need to fetch ((m\*n)+ (k\*k))\*4 bytes from Global memory.
* Also, we will write back (m\*n)\*4 bytes to memory.
* As I am using **const float\* \_\_restrict\_\_ kernel,** the kernel would be cached and we need to bring it only once.

Q3) Assuming the performance numbers you measured in assignment 5 , how long should

computing a convolution of dimension 3 on an image of 1024x768 take?

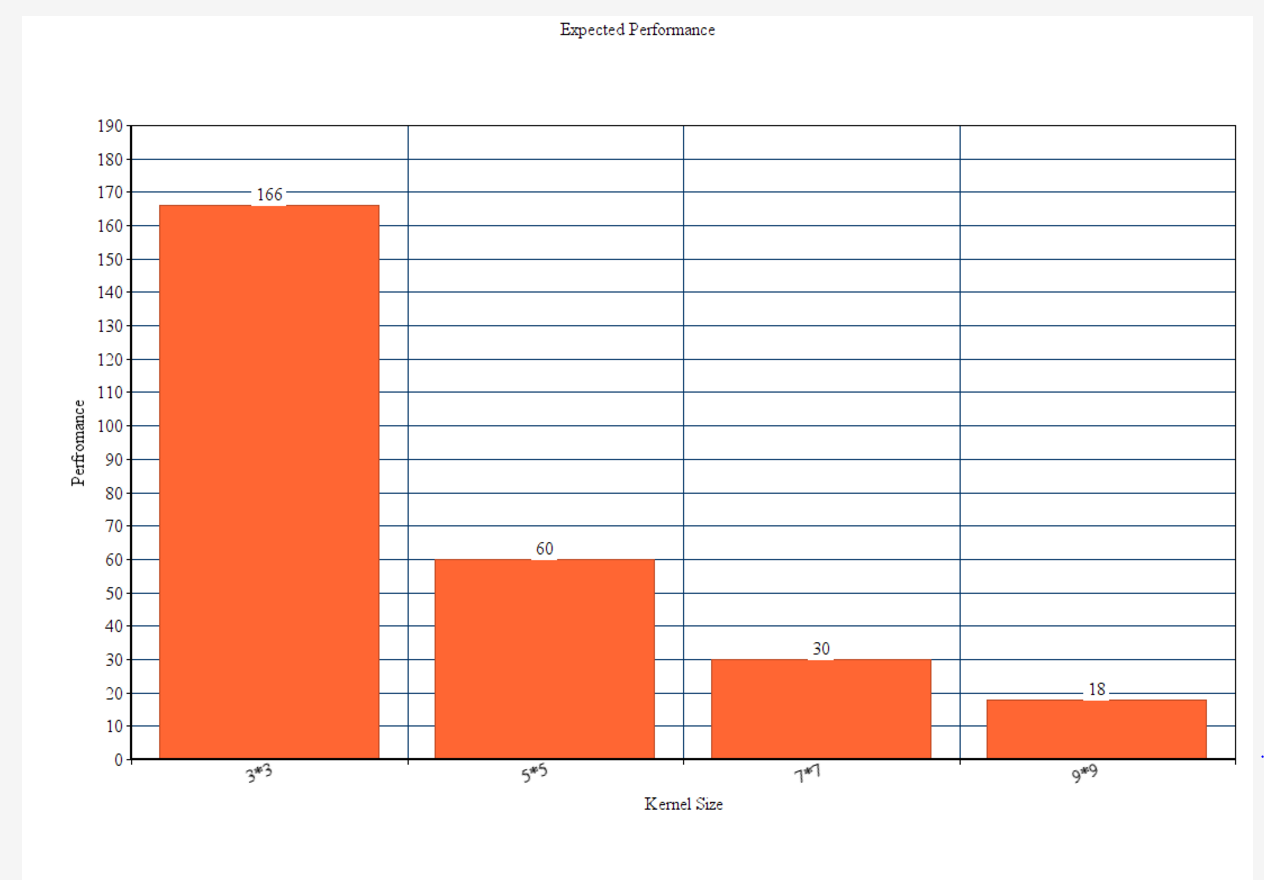
* Considering m\*n= 1024 \* 768 and k\*k=3\*3.
* From Assignment 5, we got our Flop rating as 3TB and Memory Bandwidth as 155GB/sec.
* Flop time for calculation= ((m\*n\*k\*k\*2)/ 3\*10^12))= 4.71 usec.
* Memory time calculation= (2(m\*n)+ (k\*k))\*4)/155\*10^9 = 40.5 usec.

**Performance Model**

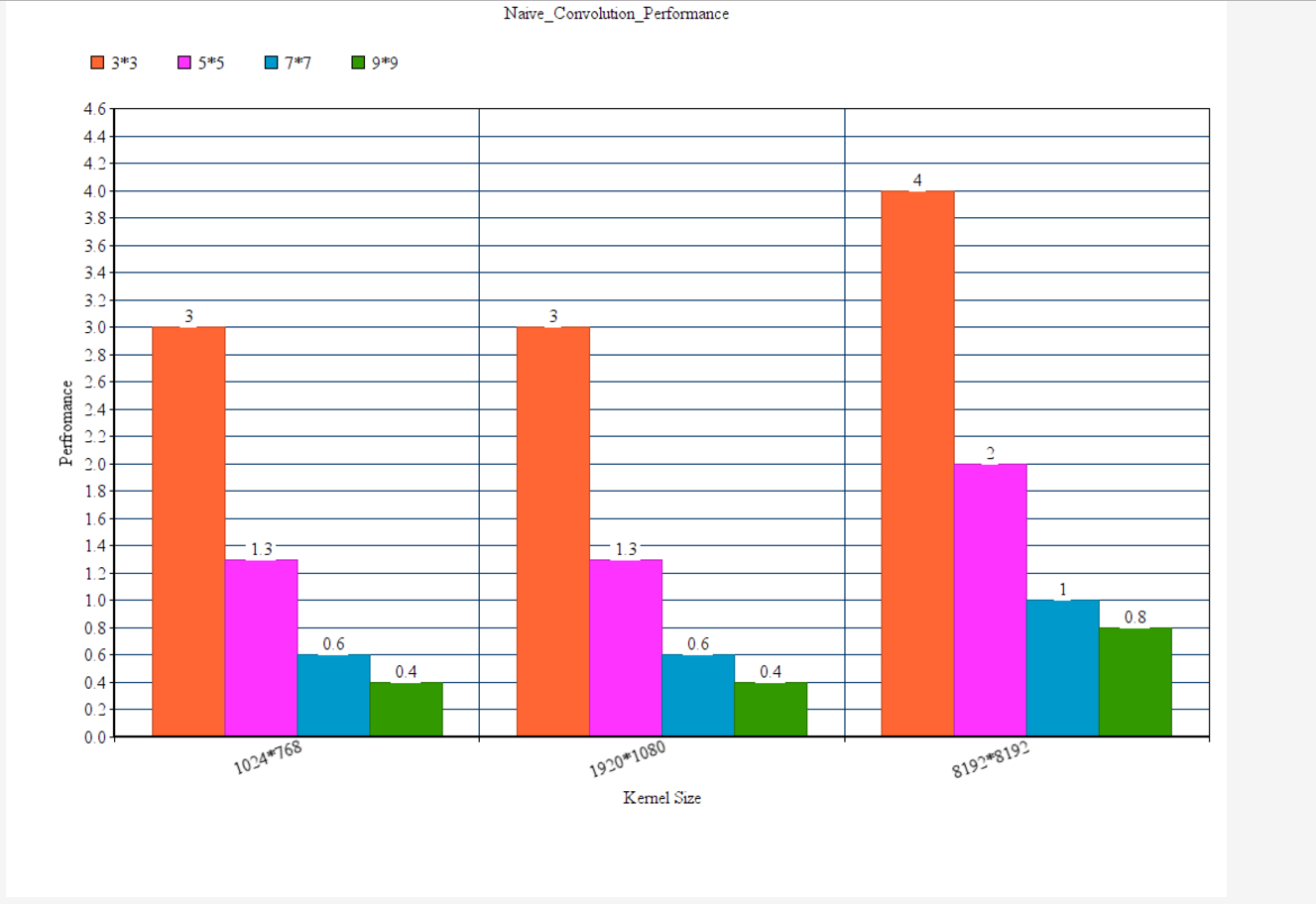
Peak flop= ((1024 \*768)/ 4.71 \* 10^-6) = 166 Gp/sec.

Memory flops= ((1024 \*768)/ 40.5 \* 10^-6) = 19Gp/sec.

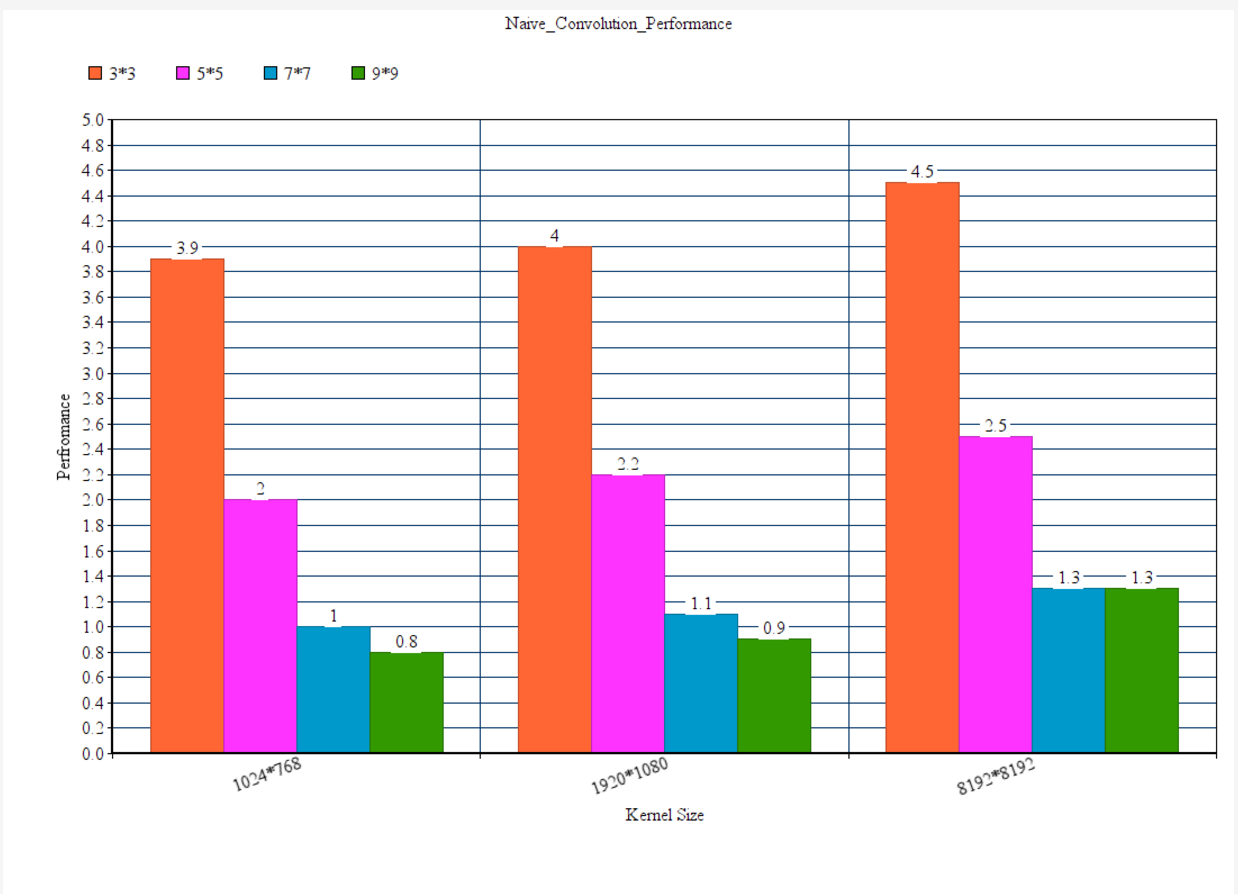
**Expected Performance Graph**



**Naïve \_Convolution Performance**



**Performance With using Shared Memory**



**We are able to get almost double the output using shared memory**