

## Performance of GTX Titan X GPUs and Code Optimization

---

**Hwancheol Jeong, Sangbaek Lee, Weonjong Lee, Jeonghwan Pak\***

*Lattice Gauge Theory Research Center, CTP, and FPRD,  
Department of Physics and Astronomy,  
Seoul National University, Seoul, 151-747, South Korea  
E-mail: [wlee@snu.ac.kr](mailto:wlee@snu.ac.kr)*

**Jangho Kim**

*National Institute of Supercomputing and Networking, Korea Institute of Science and Technology  
Information, Daejeon, 34141, South Korea*

**Juhyun Chung**

*Hankuk academy of foreign studies, Yongin, 449-854, South Korea*

Recently Nvidia has released a new GPU model: GTX Titan X (TX) in a lineage of the Maxwell architecture. We use our conjugate gradient code and non-perturbative renormalization code to measure the performance of TX. The results are compared with those of GTX Titan Black (TB) in a lineage of the Kepler architecture. We observe a significant gain in the single and double precision calculations much greater than the theoretical expectation.

*The 33rd International Symposium on Lattice Field Theory  
14-18 July, 2015  
Kobe International Conference Center*

---

\*Speaker.

## 1. Introduction

In 2015, NVIDIA released a new GPU of the Maxwell architecture: GTX Titan X (TX). TX GPUs have 20% higher computing power in the single precision (SP) floating point calculation than GTX Titan Black (TB) GPUs of the Kepler architecture. However, the former GPUs have significantly less computing power in the double precision (DP) floating point calculation than the latter GPUs by a factor of 1/7, if we compare their peak performance. Here, we would like to compare the actual performance of TX GPUs with that of TB GPUs. For this purpose, we use our conjugate gradient (CG) inverter code to check the SP performance of TX GPUs, and our non-perturbative renormalization (NPR) code to probe the DP performance. The CG code adopts the mixed precision algorithm. Hence, the SP calculation is dominant in this code. The NPR code calculates the matching factors in DP without much network communication. Hence, this code is good for testing the DP performance.

## 2. GTX Titan X

Architecture	Fermi	Kepler		Maxwell
GPU Model	GTX 580	TITAN BLACK	K40	TITAN X
SP TFLOPs	1.58	5.00	4.29	6.00
DP TFLOPs	0.20	1.30	1.43	0.19
Memory Size (GB)	1.5	6	12	12
L1 cache <sup>1</sup> (KB)	64	64	64	96
L2 cache (KB)	768	1536	1536	3072
Memory Bandwidth (GB/sec)	192.4	336	288	336.5

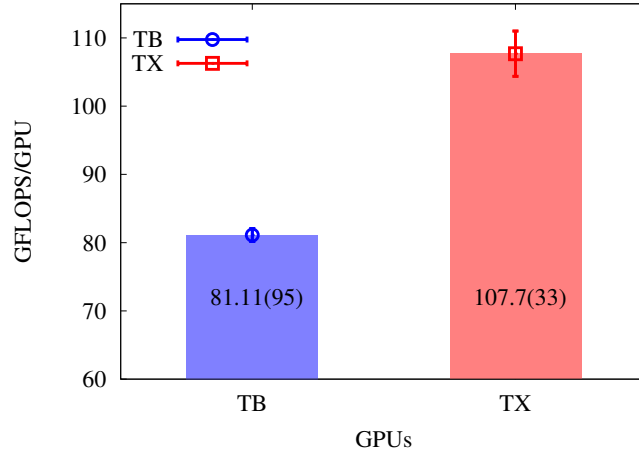
**Table 1:** Chip specification of recent NVIDIA GPUs. For more details, please refer to Refs. [1, 2, 3].

In Table 1, we list chip specification for recent NVIDIA GPU models which we have been using for our numerical study. In the table, there are three generations of GPU chip architecture: Fermi (2010), Kepler (2012), and Maxwell (2014). TX GPUs are designed based on the Maxwell architecture, which have twice more memory, twice more L2 cache and 50% more L1 cache<sup>1</sup> than TB GPUs, while the memory bandwidth is about the same. One merit for TX is that its SP peak performance is about 20% higher than that of TB. One superficial caveat is that the peak speed of the DP calculation in TX GPUs is 1/7 of that of TB GPUs. We will address this issue later in Section 4.

## 3. CG Performance

The conjugate gradient (CG) code uses the mixed precision algorithm for HYP-smearred staggered quarks. Hence, the SP calculation in this code is absolutely dominant compared with the

<sup>1</sup>Here, the nominal L1 cache means the memory shared by the L1 cache and the shared memory.



**Figure 1:** CG performance on a single GPU in units of GFLOPS. The results are measured by running the CG inverter code 10 times on the  $20^3 \times 64$  MILC asqtad lattice.

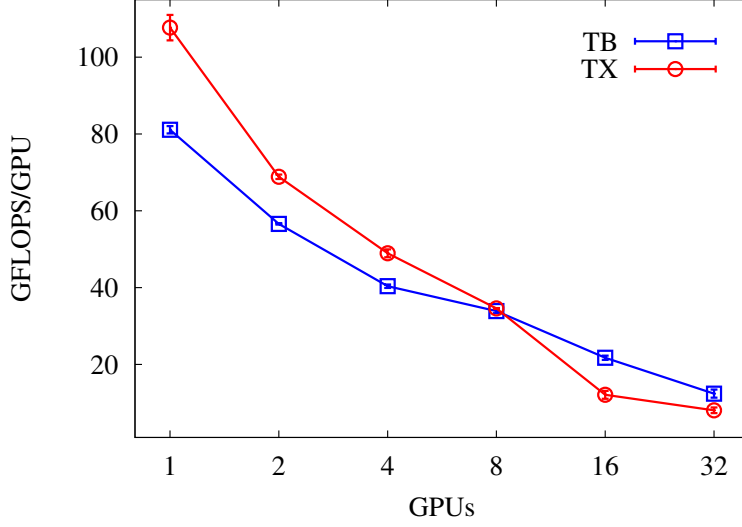
DP part. This code performs about 15000 iterations in our production job for  $B_K$  on the  $20^3 \times 64$  MILC asqtad coarse lattices. For each iteration, it uses network communication through infiniband to transfer the data vector to the nearest neighbor nodes with 2 GPUs per node.

In Fig. 1, we present the SP performance of the CG code on a single TB or TX GPU obtained using the same production job for  $B_K$ . Here, since we use only a single GPU for this test job, there is no network communication involved through the infiniband. Theoretically, the ratio of the SP peak performance between TB and TX GPUs is 1.2 as one can see in Table 1. Hence, we expect that the SP performance will increase by 20% for TX. In practice, we obtain about 33(4)% gain in the SP performance, which is significantly higher than the theoretical expectation.

In Fig. 2, we repeat the same kind of performance tests using the same production job on multiple GPUs of TB or TX type. Here, we observe TX GPUs perform significantly better than TB GPUs when we use the small number of GPUs ( $\leq 8$ ). However, for more than 8 GPUs, TB GPUs outperform TX GPUs. This indicates that we need to feed enough SP calculations above a certain threshold (= 20 million SP floating point calculations per GPU per iteration in CG) in order to obtain a significant gain using TX GPUs compared with TB GPUs. In Table 2, we present numerical values of the data points in Fig. 2.

# of nodes	TB	TX	TX/TB ratio
1	81.11(95)	107.7(33)	1.328(44)
2	55.86(57)	68.86(58)	1.233(16)
4	40.38(44)	48.9(10)	1.212(29)
8	33.92(55)	34.614(68)	1.020(17)
16	21.74(55)	12.1(10)	0.566(50)
32	12.4(11)	8.07(73)	0.650(81)

**Table 2:** Data for CG performance on multiple GPUs. They are measured by running the CG code 10 times on  $20^3 \times 64$  MILC asqtad lattice.



**Figure 2:** Scalability of CG performances in the unit of GFLOPS with multiple GPUs. The results are obtained by running the CG code 10 times on the  $20^3 \times 64$  MILC asqtad lattice.

Therefore, in order to maximize the computing performance per GPU, it is better to use smaller number of GPUs for a given production job. However, there are some caveats in this business: first, the memory of TX GPUs are limited, and second, we need to make the running time of the production job less than 2 hours in order to avoid hardware failure as much as possible. Hence, it is necessary to run the production job on multiple GPUs which are usually greater than the smallest number of GPUs. For example, when we run the production job on the  $28^3 \times 96$  MILC asqtad fine lattices, it is ideal to use 4 TX GPUs in practice.

One ambiguity in this test is that the CPU configuration for TX GPUs is different from that for TB GPUs. For TX GPUs, we use i7-5930K CPUs with 32 giga byte DDR4 RAM. For TB GPUs, we use i7-4820K CPUs with 32 giga byte DDR3 RAM. DDR4 is slightly faster than DDR3. In addition, i7-5930K CPU is slightly faster than i7-4820K CPU. Hence, this could make a tiny difference in the above data shown in Figs. 1 and 2.

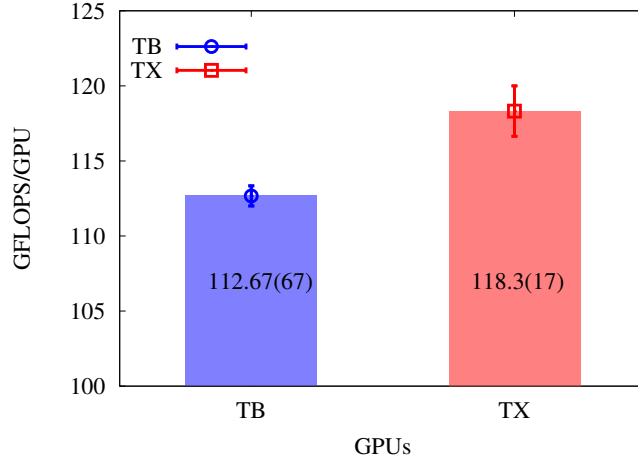
#### 4. NPR performance

Here we present the performance of the non-perturbative renormalization (NPR) code in the RI-MOM scheme. The NPR code use only the DP floating point calculation and does not have any network communication with nearest neighbors during the NPR calculation. Hence, the NPR code is very adequate to measure the DP performance of GPUs.

In the NPR code, the subroutine to calculate the one color trace four-fermion operator contraction occupies 97% of the total running time of the NPR code [4].

$$\begin{aligned}
 O_{i,l}^{f_1 f_2 f_3 f_4}(z) = & [\bar{\chi}_{i;c_1}^{f_1}(z_A) (\gamma_{S_1} \otimes \xi_{F_1})_{AB} \chi_{i;c_2}^{f_2}(z_B)] [\bar{\chi}_{i;c_3}^{f_3}(z_C) (\gamma_{S_2} \otimes \xi_{F_2})_{CD} \chi_{i;c_4}^{f_4}(z_D)] \\
 & \times [U_{i;AD}]_{c_1 c_4}(z) [U_{i;CB}]_{c_3 c_2}(z)
 \end{aligned} \quad (4.1)$$

We present a general form of one-color trace four-fermion operators in Eq. (4.1). Here,  $\chi$  and  $\bar{\chi}$  represent staggered quark fields, and  $U_{i;AD}(z)$  represents a gauge link field. The subscripts  $S_1$  and



**Figure 3:** NPR performance, actually DP performance, in the unit of GFLOPS. The results are measured by running the NPR code 10 times on the  $20^3 \times 64$  MILC asqtad lattice at  $a \approx 0.12fm$ .

$S_2$  represent the spin structure and the subscripts  $F_1$  and  $F_2$  represent the taste structure. In order to measure the DP performance of GPUs, we use this subroutine which calculate the one color trace four-fermion contraction.

As one can see in Table 1, the peak DP performance for TX is 1/7 of that for TB theoretically. Hence, theoretically we expect that the NPR code runs 7 times faster on a TB GPU. In Fig. 3, we present the DP performance of the NPR code on TB and TX GPUs. To our big surprise, it turns out that the TX GPU outperform the TB GPU by 5.0(6)%.

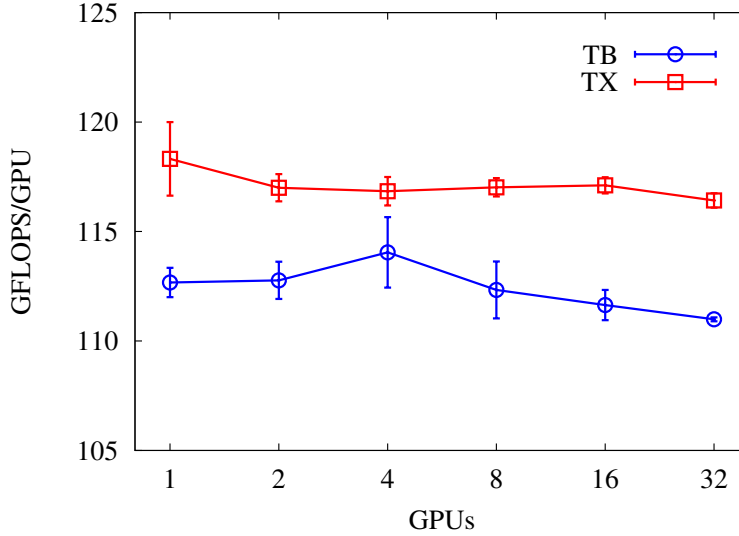
What is the reason for this unexpected surprise? The answer is that our NPR code is dominated by the data transfer between the GPU global memory and the CUDA cores. Hence, even though the TB GPU has 7 times more DP calculation power than TX, the TB GPU cannot feed the data into the CUDA cores as fast. The bottle neck is on the data transfer in the case of the NPR code.

Let us address this issue on the bottle neck in data transfer more systematically. Basically we want to introduce a concept of the CGMA ratio (Compute to Global Memory Access) [5]. The CGMA ratio is the number of DP floating point calculations per memory access. For example, let us consider the calculation of  $C = A + B$ . There are three memory accesses to read data from  $A$  and  $B$  and to save data into  $C$ . There is one DP floating point calculation for  $+$ . Hence, the CGMA ratio for this calculation is  $1/3$ .

GPU Model	Memory bandwidth (GB/sec)	Peak performance (GFLOPS)	Memory bandwidth (GB/sec) for CGMA=1	CGMA for peak performance
GTX 580	192.4	204.8	1638.4	8.5
TB	336	1331.2	10649.6	31.7
K40	288	1464.3	11714.6	40.7
TX	336.5	194.6	1556.5	4.6

**Table 3:** CGMA Ratio for various GPUs in units of [GFLOPS/(GB/sec)\*Byte]

In Table 3, we present data for the CGMA ratio on various GPUs. Here, we present the memory bandwidth and the peak DP performance in the second and third columns, respectively. In the fourth column, we present memory bandwidth required to achieve the CGMA ratio equal to



**Figure 4:** Scalability of the DP performance on multiple GPUs. We use the NPR code to measure the DP performance on TX or TB GPUs, on the  $20^3 \times 64$  MILC asqtad lattices.

1. In other words, with this bandwidth we can achieve one to one ratio between the DP floating point calculation and the data transfer. In the fifth column, we show the CGMA ratio for the peak performance on each GPU.

The CGMA ratio of our NPR code is 2.96 [GFLOPS/(GB/sec)\*Byte]. Let us assume that we use the full memory bandwidth of each GPU. Then, we can achieve 124 GFLOPS for TB and 125 GFLOPS for TX at CGMA = 2.96. Hence, we observe that this actual DP performance is much lower than the theoretical peak. This indicates that the bottle neck is on the data transfer.

Then at CGMA = 2.96, we expect that TX GPUs will outperform TB GPUs only by 0.15%. However, in practice we find a gain of 5.0(6)% in Fig. 3, which is much larger than the theoretical expectation by an order of magnitude. At present, we do not understand this enormous amount of gain in TX. This needs further investigation in the future.

In Fig. 4, we present the DP performance per GPU when we run the NPR code on multiple GPUs. In Table 4, we summarize the actual values of the data points in Fig. 4. Since the NPR code does not have data transfer to the nearest neighbors, we expect that the performance will be constant regardless of the number of GPUs used for the measurement. The data in Fig. 4 is consistent with this expectation. In this test jobs, TX GPUs outperform TB GPUs by about 5% regardless of the number of GPUs. This large gain remains as a conundrum, which needs further investigation in the future.

In our production job on NPR, we use 28 GPUs of TX or TB type on  $28^3 \times 96$  MILC asqtad lattices. A single production job takes about 6.5 hours.

## 5. Conclusion

Recently NVIDIA has released a new GPU model, TX. Here, we present the results of our performance test on TX GPUs and compare them with those on TB GPUs.

# of nodes	TB	TX	TX/TB ratio
1	112.67(67)	118.3(17)	1.050(6)
2	112.23(79)	117.04(57)	1.038(9)
4	114.1(16)	116.84(65)	1.024(6)
8	112.3(13)	117.02(42)	1.042(4)
16	111.64(69)	117.11(37)	1.049(7)
32	110.99(90)	116.42(33)	1.049(3)

**Table 4:** DP performance per GPU in units of GFLOPS. The data describes the data points in Fig. 4.

In the SP performance test, TX outperforms TB by 33(4)%, which is significantly higher than the theoretical expectation. This test is done using the CG code.

In the DP performance test with the NPR code, TX outperforms TB by 5.0(6)%, which is dramatically larger than the theoretical expectation of 0.15% gain. At present, we do not understand this gain very well. This needs further investigation in the future. The DP performance of TX with the NPR code is controlled by the data transfer, which is the main bottle neck. Hence, in order to increase the DP performance, we need to optimize the NPR code further for higher CGMA ratio.

## Acknowledgments

The research of W. Lee is supported by the Creative Research Initiatives program (No. 2015001776) of the NRF grant funded by the Korean government (MSIP). This work was supported by SNU Undergraduate Research Program. W. Lee would like to acknowledge support from the KISTI supercomputing center through the strategic support program (No.KSC-2014-G3-002) with much gratitude. J. Kim was supported by a Young Scientists Fellowship through the National Research Council of Science & Technology (NST) of Korea. Computations were carried out on the DAVID GPU clusters at Seoul National University.

## References

- [1] *GeForce GTX TITAN X Specifications*. <http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x/specifications>.
- [2] *GeForce GTX TITAN X Specifications*. <http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-black/specifications>.
- [3] SWME Collaboration, Y.-C. Jang, H. Jeong, J. Kim, W. Lee, J. Pak, and Y. Chung, *Code Optimization on Kepler GPUs and Xeon Phi*, *PoS LATTICE2014* (2014) 035, [1411.2223].
- [4] SWME Collaboration, H. Jeong, J. Kim, J. Kim, W. Lee, J. Pak, and S. Park, *Non-perturbative Renormalization of Four-Fermion Operators Relevant to  $B_K$  with Staggered Quarks*, *PoS LATTICE2014* (2015) 286, [1410.6607].
- [5] H. Jeong, W. Lee, J. Pak, K.-j. Choi, S.-H. Park, J.-s. Yoo, J. H. Kim, J. Lee, and Y. W. Lee, *Performance of Kepler GTX Titan GPUs and Xeon Phi System*, *PoS LATTICE2013* (2014) 423, [1311.0590].