mühendislikdergisi
Dicle Üniversitesi Mühendislik Fakültesi

Dicle University Journal of Engineering (DUJE)

web: http://dergipark.gov.tr/dumf

Research Article

A Comprehensive Performance Comparison of Dedicated and Embedded GPU Systems

Adnan Ozsoy [1,*]

[1] Department of Computer Engineering, Hacettepe University, adnan.ozsoy@hacettepe.edu.tr
(ORCID: https://orcid.org/0000-0002-0302-3721)

ARTICLE INFO

Article history:

Received 26 May 2020
Received in revised form 29 June 2020
Accepted 9 July 2020
Available online 30 September 2020

Keywords: *NVIDIA Jetson, Embedded GPGPU, CUDA*

ABSTRACT

General purpose usage of graphics processing units (GPGPU) is becoming increasingly important as graphics processing units (GPUs) get more powerful and their widespread usage in performance-oriented computing. GPGPUs are mainstream performance hardware in workstation and cluster environments and their behavior in such setups are highly analyzed. Recently, NVIDIA, the leader hardware and software vendor in GPGPU computing, started to produce more energy efficient embedded GPGPU systems, Jetson series GPUs, to make GPGPU computing more applicable in domains where energy and space are limited. Although, the architecture of the GPUs in Jetson systems is the same as the traditional dedicated desktop graphic cards, the interaction between the GPU and the other components of the system such as main memory, central processing unit (CPU), and hard disk, is a lot different than traditional desktop solutions. To fully understand the capabilities of the Jetson series embedded solutions, in this paper we run several applications from many different domains and compare the performance characteristics of these applications on both embedded and dedicated desktop GPUs. After analyzing the collected data, we have identified certain application domains and program behaviors that Jetson series can deliver performance comparable to dedicated GPU performance.

Doi: 10.24012/dumf.742732

* Corresponding author
Adnan Ozsoy
✉ adnan.ozsoy@hacettepe.edu.tr

Please cite this article in press as A. Ozsoy, “A Comprehensive Performance Comparison of Dedicated and Embedded GPU Systems”, DUJE, vol. 11, no.3, pp. 1011-1020, September 2020.

Introduction

The use of graphics processing units (GPUs) become more prevalent for accelerating non-graphics computations as their computation power leave central processing units (CPUs) behind [1]. Especially in fields where large data sets need to be processed and the process is parallelizable. There is a huge and expanding gap between CPU performance and GPU performance due to graphical processing units' architecture and massive core count [2]. As a result, general purpose graphics processing units (GPGPU) is gaining importance and popularity as a powerhouse system.

General purpose GPU computing (GPGPU) is practiced in many fields such as image classification, speech recognition, language processing, sentiment analysis, video captioning, video surveillance, face detection, satellite imagery, etc. [3, 4]. Other than those fields, it is used in scientific computing, especially in bioinformatics for purposes like cancer cell detection, diabetic grading and drug discovery [5]. Moreover, GPUs are widely used in researches on the topic of autonomous vehicles as well [6]. Even though those fields seem to be unrelated, techniques used for solving their problems share a lot in common. For instance, pedestrian detection, image classification and speech recognition all use deep learning and convolutional neural networks. Both require high performance accelerator to solve compute intensive problems. As a consequence, GPGPU plays a big role for many fields that are different from each other but all require to solve compute intensive problems.

Computational needs for the systems on the field and mobile platforms in the recent years forced an additional product line of hardware and software solutions. This shift from more capable and complete systems of desktop/workstation machines to mobile hardware shows its existence in several areas. Self-driving cars, drones, field processing units, computer vision, machine learning applications and also IoT applications are examples for this growing area of mobile and embedded computing. To satisfy the mobile computing needs in these areas, NVIDIA released a series of embedded devices, called Jetson series [7]. Jetson embedded platform provides a rapid development environment and deploy process for compute-hungry applications with more efficiency in terms of energy, cost and space. Jetson platforms are the first of its kind as a mobile supercomputer where it both facilitates ARM CPUs and a very capable GPU. This system-on-chip structure and the accompanying host module with necessary I/O gates act as a complete system.

Although Jetson systems are energy, cost and space efficient, their computation capabilities are falling short compared to dedicated off-the-shelf desktop grade GPUs which have higher core counts and bigger memories. However, there are studies that experiment Jetson cards on different applications and report that the slowdown of Jetson performance compared to dedicated GPUs is not always as big as the theoretical core ratio [8]. When choosing a hardware for a specific task, it is hard to make a decision on which hardware will suit the needs for that specific task. There is a lack of a comprehensive comparison of embedded and dedicated GPUs in the literature and the capabilities of Jetson series are not truly investigated and compared against the dedicated GPUs.

In this study, we provide a comprehensive comparison of embedded and dedicated GPUs which is missing in the literature. In our tests we use an NVIDIA GeForce Titan X and a Jetson TX2. Through extensive benchmarking and comparing these devices for different applications, it will be possible to investigate strengths and weaknesses of the embedded GPGPU solutions against dedicated desktop solutions. To truly understand their capabilities and exploit these cards to several scenarios of application use, we tested these two cards on two main benchmarks. As novel contributions of this paper, we provide a performance guideline of mainstream application domains for NVIDIA embedded Jetson Series Cards, give a comparison between Desktop level GPUs (Titan X) and embedded Jetson series (Jetson TX2), identify application domains and behavior that suit better on Jetson series in terms of space, cost and energy efficiency.

The rest of the paper is formed as follows. Section 2 lists related work in the field. Section 3 provides background information for NVIDIA GPGPU Framework and difference in dedicated and embedded GPUs. Section 4 introduces benchmark suites used for tests. Section 5 gives results of tests and a thorough analysis of results. Lastly, Section 6 concludes the paper.

Related Work

In literature, there are many publications that utilize embedded GPUs for certain tasks. Jetson TK1, the first model of Jetson series, appeared in various papers after NVIDIA launched it in 2014. To list a few of them; [9] used TK1 for real-time object detection with Caffe framework, in [10] Jetson TK1 is compared with another system-on-chip device, named FireFly, and The Astro Cluster, which consists of 46 Jetson TK1 boards, is built and tested for distributed GPU tasks. After the release of Jetson TX1, the second generation of Jetson cards, it also gained popularity among researchers. [11] uses both Jetson TK1 and Jetson TX1 for comparing the performance and energy-efficiency of five different heterogeneous computing platforms for bio-molecular and cellular simulation workloads. [12] evaluates the effectiveness of Jetson TX1 in real-time computer vision workloads. Jetson TX2, the successor of Jetson TX1, is launched in March 2017. [13] experimented with Jetson TX2 on keyframe and feature-based monocular simultaneous localization and mapping (ORB-SLAM) for an unmanned aerial vehicle (UAV). [14] Facilitates TX2 for surveillance systems and face recognition. In [15] another vehicle localization for autonomous navigation uses TX2. Recently, deep-learning and vision applications also make use of the NVIDIA Jetson series cards [29,30]. In all of those publications, the Jetson series modules are tested and the performance results are reported for that specific application only.

In literature there is a wide range of publications that aim to compare accelerator hardware performance. The vast majority of these studies are between dedicated GPUs and other hardware such as field-programmable gate array (FPGAs), application-specific integrated circuit (ASICs) or multi core CPUs [16-19]. However, there is a huge demand in using the embedded GPUs because of their advantages in power, cost and size but there is no comprehensive study to identify the performance of these hardware compared to dedicated solutions. The main downside of embedded solutions is having less cores and as a result delivering less compute power. However, in a wide range of applications, the Jetson cards capabilities may differ. In the aforementioned publications, the test results for different applications all differ in terms of slowdown a Jetson card face compared to a dedicated GPU. That indicates that Jetson cards performance needs to be analyzed comprehensively.

In this paper, we provide performance results of two commonly used and known benchmarks, compare results with the results obtained from a dedicated high-end desktop GPU vs Jetson TX2 and analyze the performance results. Different than the papers in the literature where their main focus is on a specific application performance, we focus on the general group of applications and application behavior that fit very well and very badly on Jetson series.

Background

General Purpose Graphic Processing Unit: GPGPU is the highest trend in high-performance computing which uses graphics processing units for not just graphic related tasks but also any type of general purpose computational needs. The massively parallel architecture of the GPU that is traditionally designed for only computer graphics, now used as a powerhouse for any compute intensive applications. Algorithms to be used for GPGPU must have two main properties. First, they must be data parallel. Second, they must be throughput intensive. Data parallel means that an execution of a program requires to run the same operation on many different data points. An algorithm is throughput intensive if it processes lots of data where a huge potential for parallel processing exist. With the help of the simple but many processing units on GPUs, it is possible to achieve extreme performance on data parallel HPC algorithms.

Compute Unified Device Architecture (CUDA): CUDA is a general purpose parallel computing platform introduced by NVIDIA in 2006 [20]. It enables programmers and scientists to develop applications that are aimed to work on GPU. Its application programming interface is mainly based on C but there are also bindings for various programming languages including Python and FORTRAN. The variety on language bindings provide a large number of options and easy to port existing legacy code to GPUs.

NVIDIA Jetson TX2: Jetson TX2 is a system-on-chip (SoC) product that designed for demanding embedded applications. It is a credit-card sized module running under 7.5 watts with a high processing capability. It has 8 GB of LPDDR4 memory, 32 GB eMMC storage supported with 256 cores Pascal GPU, Quad core ARM and Dual core Denver processor.

By using CUDA and Jetson TX2 developers can cope with the problems that requires high computational needs under certain space and cost limitations. Jetson TX2 is also suitable for applications which have limited access to an energy resource since its low-energy usage.

Benchmarks: In the testing phase, two different benchmark suites, Paralution and SHOC, are used to test Jetson TX2 and GeForce Titan X cards. Both benchmark tests include several applications from very different domains that provide a good coverage of capabilities.

*Paralution:*Benchmark suite Paralution is a library for sparse iterative methods with CUDA support [21]. Paralution contains 25 applications which include parallel solvers and preconditioners which can run on GPU. Those applications are mainly based on C/C++ but it has also plug-ins for some other platforms including FORTRAN, MATLAB and OpenFOAM. It also supports many sparse matrix representations (e.g. COO, CSR, DIA etc.). In addition to iterative solvers and preconditioners, PARALUTION includes a benchmark application that applies several matrix operations on the matrices given as input and gives total execution time as output.

SHOC: The Scalable Heterogeneous Computing Benchmark Suite [22], SHOC, is a group of testing programs to analyze characteristics and behavior of GPU and multicore processors. Inputs used in the benchmark are provided in the package itself. This benchmark suite includes 3 levels of programs from Low to High level operations:

- Level 0 has bus speed, memory latency and bandwidth and peak flops tests for single and double precision.
- Level 1 has Fast Fourier Transform, reduction, matrix multiplications, scan, sort tests.
- Level 2 has high level programs like Quality Threshold Clustering algorithm and chemical applications.

Results and Analysis

We ran benchmarks on two distinct systems; embedded development platform Jetson TX2 and GeForce Titan X GPU mainly used in desktop machines. Since we only interested in GPU performance, the rest of the specifications of the systems are not related in the scope of this paper. Jetson TX2 technical specifications are discussed in background section. Table-1 shows an overall comparison between Jetson TX2 and GeForce Titan X [23, 24].

Table-1 shows that GeForce Titan X has 12 times more cores, 1.5 times more memory than Jetson TX2. The individual core speed of Jetson TX2 is 1.3 times faster than cores in Titan X. Thus, the theoretical computational slowdown ratio of porting an application to a Jetson TX2 rather than Titan X is 9.23 times (12/1.3).

Table 1. Jetson TX2 and Titan X Technical Specifications

	Jetson TX2	GeForce Titan X
GPU	NVIDIA Pascal 256 cores @1300 MHz	NVIDIA Maxwell 3072 cores @1000 MHz
CPU	Quad ARM A57 + Dual Denver2	N/A
Memory	8 GB	12 GB
Storage	32GB	N/A
Power	7.5W	250W
Price	$400	$1200

Downside of Titan X is that it consumes 33 times more power than TX2 respectively, and the total power consumption is even worse in practice since it must be a component of a system where the system will also consume energy. Other than these technical features, their costs should be considered too. As of March 2020, EVGA GeForce Titan X [25] costs $1,199.99 whereas NVIDIA Jetson TX2 costs $399.99 [26]. So, GeForce Titan X is far more expensive (3x more than TX2) than Jetson cards. Also, buying a GeForce Titan X will not be sufficient and in order to build a complete system, developers should spend approximately $2000-$5000 more to have a complete system and that is even more costly than Jetson development kits. When analyzing the results, we consider these advantages and disadvantages of GeForce Titan X and Jetson TX series.

***Paralution Tests*: **Paralution does not come with its own dataset so we collected matrix sets from Matrix Market [27] and The SuiteSparse Matrix Collection [28] to conduct Paralution benchmarks. There are different categories of matrices, thus we tried to pick matrices from divergent domains. After running benchmark on matrices collected initially, we collected a second set of matrices according to our observations from the obtained results. For each matrix in these sets, we ran benchmarks on Jetson TX2 and GeForce Titan X. Then, we compared the results and observed where the gap between these three devices expands and shrinks.

Benchmark application outputs 16 different execution times for different operations. There are two types of benchmarks; Stand-alone and combined benchmarks. In stand-alone benchmarks, the application executes certain operation for N times without applying another operation. In combined benchmarks, however, it executes all operations sequentially and repeats that block for N times.

Stand-alone and combined benchmark values are prefixed with SA_ and C_ on figures, respectively. On the X-axis, matrix names are present and Y-axis shows slowdown which is equal to Equation 1 for each matrix and operation.

$$\text{Slowdown}= \frac{(Execution\ time\ on\ Jetson\ TX2)}{(Execution\ time\ on\ GeForce\ Titan\ X)} \quad (1)$$

Benchmark's results for matrix set-1 can be seen on Figure 1. The benchmark results show a variety of slow down rates among different operations and different matrix kinds. The slowdowns are expected since the computational power is 9.23x times less than Titan X as explained at the beginning of this section. However, not all slowdowns are around expected range, some of them are more, some are less that requires a deeper look into the results.

The operation with highest slowdown is SA_DIA SpMV (14.74x on average) and the performance of systems is at its closest point for SA_COO SpMV (3.14x on average) operation. After observing these results above, we decided to concentrate on SpMV (Sparse Matrix Vector Multiplication) operations which use different sparse matrix formats since the values recorded for those operations contain extreme cases; the highest and lowest slowdowns. So, we eliminate other operations and decide to prepare a new input set for SpMV operations. For that purpose, we collect matrices which have different structures for our second matrix set. We choose 31 new matrices for 3 types of structures: banded, block triangular and diagonal. Results for matrix set-2 are in Figure 2, 3 and 4.

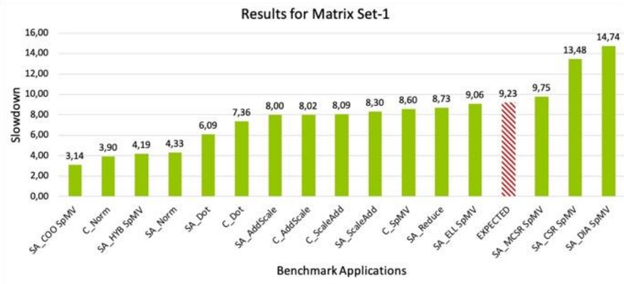


Figure 1. Benchmark results for matrix set 1. Matrices ordered by descending nonzero elements vs Slowdown.

Analyzing results taken from matrix set-1 and set-2, we conclude that representation of the matrices is crucially important for devices' performance. For applications which use banded matrices that are less sparse, the Jetson TX2 shows more slowdown than more sparse matrices. In some cases, more than theoretical slowdown has been witnessed (e.g. matrix bcsstk21 and MCSR format).

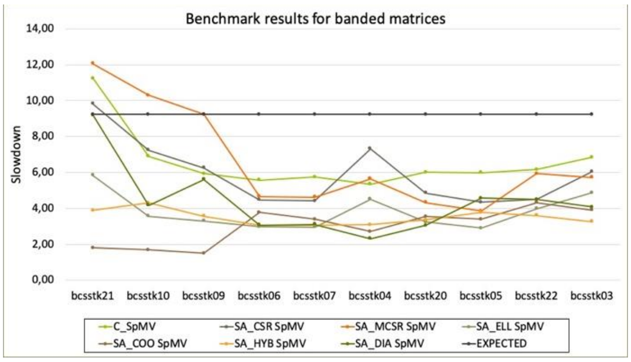


Figure 2 – Benchmark results for banded matrices. Matrix names, ordered by descending nonzero elements vs slowdown

In terms format wise performance comparison of Jetson TX2 and GeForce Titan X, COO format has the lowest average slowdown rate than expected slowdown (GeForce Titan X is faster 3.01 times on average, expected is 9.23 times). MCSR format has the highest average slowdown rate (GeForce Titan X is faster 6.63 times on average, expected is 9.23 times).

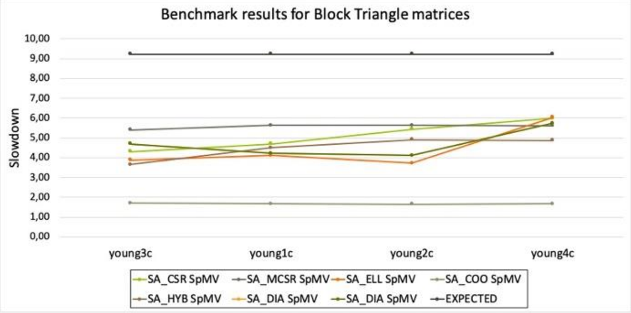


Figure 3 – Benchmark results for block triangular matrices. Matrix names, ordered by descending nonzero elements vs slowdown

For block triangle matrices, the performance of COO format matrices on Jetson TX2 gets even better with a slowdown of 1.68 times on average, expected is 9.23 times (Figure 3). Again, MCSR format on average is the worst among these formats with a slowdown of 5.57 times. However, none of the matrices in this category achieved less than the expected slowdown, making Jetson TX2 a more efficient solution than GeForce Titan X in terms of computation power efficiency. A similar trend shows in diagonal matrices as shown in Figure 4 where the Jetson TX2 shows more slowdown on less sparse matrices compared to more sparse matrices.

Considering the difference between the number of CUDA cores and core speeds (which we have discussed at the beginning of Section 4), we expected slowdown to be around 9.23 and in general tested SpMV formats performed better than expected for more sparse matrices. At the same time, GeForce Titan X has additional disadvantages in terms of power consumption and cost. Adding all together, Jetson TX2 is more efficient than GeForce Titan X for SpMV operations on those formats.

Values on Figure 2, 3 and 4 show us that performance difference between operations do not fluctuate as in Figure 1 for matrices which have the same structure. So, the structure of the matrix plays a big role for our comparison. Also, observing Figure 4, it can be said that if the structure of the matrix is appropriate for a format, performance difference between GeForce Titan X and Jetson TX2 remains less than 5x in most cases. For example, slowdown value of SA_DIA SpMV operation changes between 3.72 and 4.24 for matrices other than bcsstm25. If the program needs to convert the format to a different one than given, then the Jetson TX2 gets negatively affected more than Titan X.

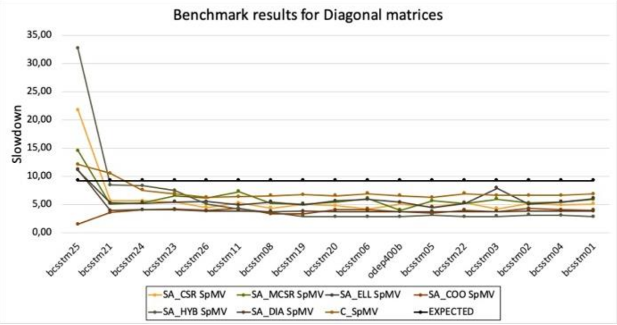


Figure 4 – Benchmark results for diagonal matrices. Matrix names, ordered by descending nnz vs Slowdown

Considering their costs, sizes and power consumptions, we concluded that Jetson TX2 is more efficient than GeForce Titan X in sparse matrix-vector multiplication operations for matching formats. This is a key finding since the use of Jetson cards are commonly thought for only mobile and limited setups, however the performance comparisons show that Jetson cards can also deliver matching performance with dedicated desktop GPUs.

SHOC Tests**:** SHOC benchmark is also tested both on Jetson TX2 and GeForce Titan X with the input that is provided by the package itself. The results below are obtained from input size category 3 which is suitable for Discrete GPUs. The figures have slowdown in the Y-axis and different applications on X-axis.

The slowdown rate of each test is compared to theoretical slowdown between Jetson and Titan X which is approximately 9.23x. From the results of Figure 5 we can recognize three interesting applications where the slowdown ratios are very different than expected. These three applications are global memory read and write (gmem_*), bus speed (bspeed_*), and reduction/scan/sort operations with/without PCI Express included results.

Global Memory Read And Write: Global Memory Read and write tests applied on Jetson TX2 and Titan X with same inputs. Results are shown for aligned access and strided (uncoalesced) access. As seen in the Figure 6, when the program switches from aligned accesses to strided (uncoalesced) accesses, GeForce Titan X experience 5 times slower bandwidth for read operations from global memory and 20 times slower bandwidth for write operations. These values for Jetson TX2 are 3x and 2x, respectively.

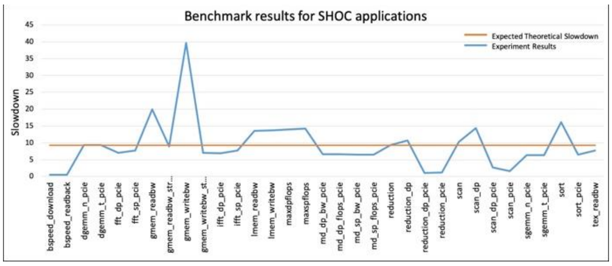
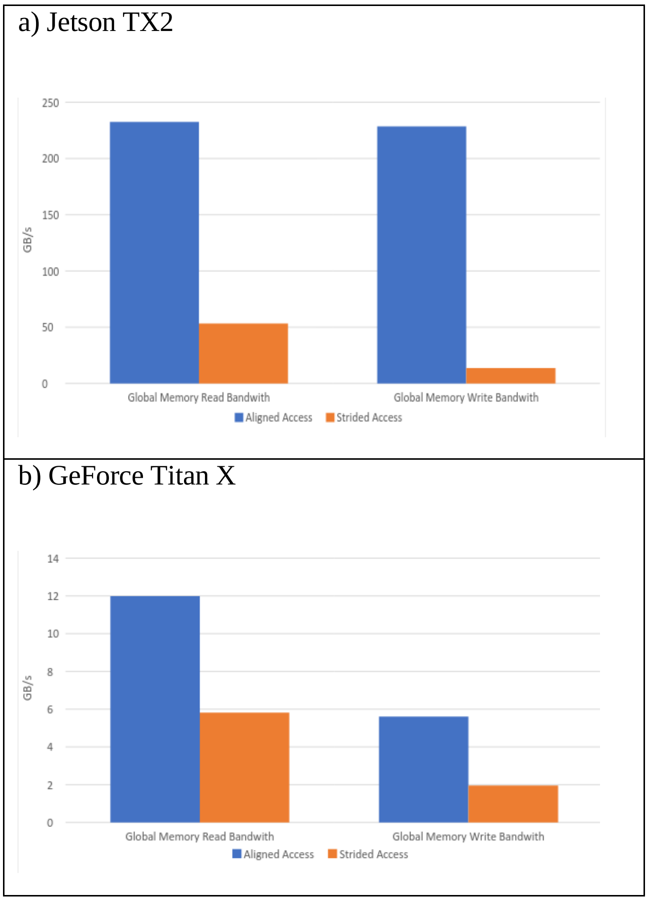


Figure 5. Benchmark results for SHOC applications slowdown rates. Applications vs Slowdown.

Global Memory Read And Write: Global Memory Read and write tests applied on Jetson TX2 and Titan X with same inputs. Results are shown for aligned access and strided (uncoalesced) access. As seen in the Figure 6, when the program switches from aligned accesses to strided (uncoalesced) accesses, GeForce Titan X experience 5 times slower bandwidth for read operations from global memory and 20 times slower bandwidth for write operations. These values for Jetson TX2 are 3x and 2x, respectively.

It is clear that both of the cards lost bandwidth performance changing from regular to irregular accesses, however, the drop in the Jetson TX2 is not as dramatic as it is on Titan X. Thus, we can say that programs that has to show irregular memory access pattern using variable access patterns to global memory can be run more effectively on Jetson TX2.

Bus Speed: Rather than a standard PCI Express connection, Jetson has a memory shared among CPU and GPU cores on the same chip. This feature gives it a competent performance with the GeForce Titan X. The Figure 7 results show that, Jetson has 2 times faster bus speed than GeForce Titan X. Programs with high memory copy tendency can work on Jetson TX2 more efficiently.


a) Jetson TX2

b) GeForce Titan X

Figure 6. Global Memory tests for aligned vs strided access

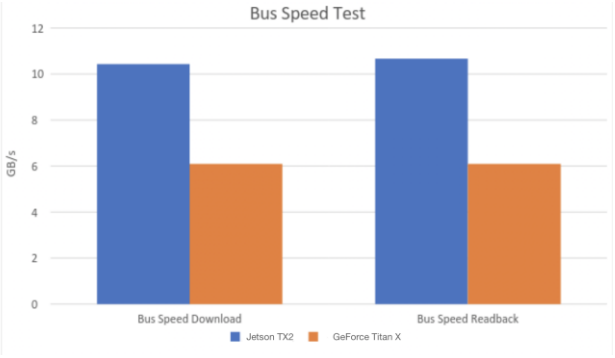


Figure 7. Bus Speed Download and Read back Tests

Reduction: Reduction test measures the performance of the reduction operation with different precision floating point data. It is applied for both single and double precision floating point numbers. For the single precision operations Jetson has a better performance than GeForce Titan X as shown Figure 8. These results include PCI Express operations and Jetson TX2 benefits from the memory that is among CPU and GPU as mentioned in bus speed section above.

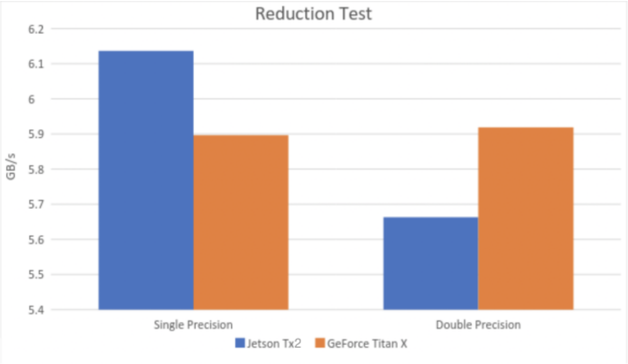


Figure 8. Reduction Operation Test Results

On double precision operations, GeForce Titan X performed better. There is an architectural difference between Jetson TX2 and GeForce Titan X where TX2 uses Pascal architecture and Titan X facilitates Maxwell architecture, thus we can say that the main reason of the difference between single and double precisions can be from behavioral reasons about architecture. However, GeForce Titan X is a high-end hardware with more Streaming Multiprocessors (SM). In each SM there is a number of specialized hardware for double precision operations, thus more SMs lead to better performance naturally. The main reason for the Jetson's better single precision performance compared to Titan X is the included PCI Express data movement cost. From the total benchmark results, we can see that version without the PCI Express is running faster on Titan.

Scan: Scan test measures the performance parallel prefix sum operation with floating point data. It is applied for both single and double precision floating point numbers. Results for this test given in Figure 9 show that, Jetson performed 2x slowdown compared to Titan X, where way better than expected slowdown rate. Slowdown rate increases more on double precision test for similar reasons described in the reduction operation but it is still an efficient option compared to the theoretical rate. One open question is why Scan with PCI Express included is not better in Jetson. In Reduction, because of the PCI Express cost, reduction performed better on Jetson. Because of the time limitations, we leave this open problem for future work.

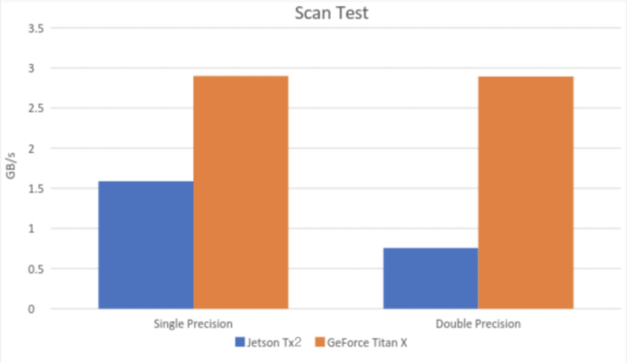


Figure 9. Scan Operation Test Results

To conclude, Jetson TX2 performed better than the expected theoretical slowdown rate in bus speed, reduction and scan tests. As a result, we can conclude, Jetson TX2 works more efficiently on program domains with high memory access needs. Also scan, sort and reduction operations especially with single precision data, work more efficiently considering the energy consumption and cost of these two hardware. Jetson's unique memory system plays a big role on this performance. Furthermore, on global memory access test, Jetson TX2 works more stable on transition between aligned and strided access. On the same test, we have observed 5-6 times more slowdown on performance with the strided global memory access. Concluding that, applications with irregular access patterns can work on Jetson TX2 more efficiently than GeForce Titan X.

Conclusion

GPGPUs are powerhouse solution for compute intensive applications where performance is the main concern. However, in recent years, the use of GPUs finds new fields. The need for high performance solutions in limited resource scenarios is always a problem and there are solutions that uses embedded hardware where the power and space is the main resource that is scarce. However, the programming and re-use of these systems is a tedious job that requires additional skills than general programming knowledge. The recent developments in machine learning and AI created a demand in high performance hardware solutions on the field as well. Thus, NVIDIA introduced embedded solutions for resource limited scenarios, called Jetson series. Although the GPUs on the Jetson series cards have the same architecture with desktop GPUs, their use in development kits and interaction with other components of the system is different.

Consequently, in this paper we experiment with a set of applications from different domains to fully understand the capabilities of the Jetson series embedded solutions. We compared Jetson TX2 results with a high-end desktop GPU, Titan X, and analyze the results. We conclude that although the Jetson TX2 is more than an order of magnitude less equipped device than Titan X in terms of the number of cores, memory, and speed, Jetson TX2 can achieve comparable performance to Titan X for certain applications and program behavior. Our main contribution in this paper is identifying the strong side of Jetson TX2 device and provide a guide for researchers. Our tests showed us that Jetson series cards are not only devices for limited resource scenarios, but also capable to deliver good performance in certain applications.

For future work, we would like to investigate more on the energy usage side and make experiments and analyze results for energy efficiency of different applications. Because of the space limitation and scope of in this paper, we couldn't touch base on more thorough analysis of different applications and their behavior on Jetsons. Another interesting direction for future work will be multi Jetson performance through message passing protocols. Last but not least, the Jetson development kit comes with many different interfaces for other hardware. The investigation of hardware that can be connected to Jetson and their performance is an interesting open research area.

Acknowledgement

This research was supported by TUBITAK (The Scientific and Technological Research Council of Turkey) grant number 117E142 and NVIDIA GPU Teaching Center.

References

1. Reese, J. and Zaranek, S., Gpu programming in matlab. MathWorks News&Notes. Natick, MA: The MathWorks Inc, pp.22-5. 2012.
2. Kirk, D., NVIDIA CUDA software and GPU parallel computing architecture. In ISMM (Vol. 7, pp. 103-104). 2007, October.
3. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks, 25th Int. Conf. on Neural Information Processing Systems, p.1097-1105. 2012.
4. CUDA Spotlight GPU Applications Showcase. https://devblogs.nvidia.com/parallelforall/cuda-spotlight-gpu-accelerated-speech-recognition/ (Accessed at 22.05.2020)
5. GPU Technology Conference, Tutorials. http://on-demand.gputechconf.com/gtc/2015/webinar/deep-learning-course/intro-to-deep-learning.pdf (Accessed: 22.05.2020)
6. GPU Technology Conference, Tutorials. http://on-demand.gputechconf.com/gtc/2014/presentations/S4621-deep-neural-networks-automotive-safety.pdf (Accessed: 22.05.2020)
7. NVIDIA Embedded Platform. https://developer.nvidia.com/embedded/jetson-embedded-platform (Accessed : 22.05.2020)
8. B. Baumann. "Jetson TK1", Institut Für Technische Informatik, Advanced Seminar Computer Engineering, Seminar Winter Term 2014/2015. 2015.
9. C. Alicea-Nieves. Caffe Framework on the Jetson TK1: Using Deep Learning for Real Time Object Detection. SUNFEST at PENN. (https://sunfest.seas.upenn.edu/) 2018.
10. R. J. Abbasi. HPCG benchmark for characterising performance of SoC devices, (Unpublished Master Thesis). The Australian National University. 2015.
11. Stone JE, Hallock MJ, Phillips JC, Peterson JR, Luthey-Schulten Z, Schulten K. Evaluation of emerging energy-efficient heterogeneous computing platforms for biomolecular and cellular simulation workloads. IEEE 30th Int. Parallel and Distr. Processing Symposium Workshops, IPDPSW. IEEE Computer Society. p. 89-100. 2016.
12. Nathan Otterness, Ming Yang, Sarah Rust, Eunbyung Park, James H. Anderson, F. Donelson Smith, Alexander C. Berg, Shige Wang. An Evaluation of the NVIDIA TX1 for Supporting Real-Time Computer-Vision Workloads. RTAS 2017: 353-364. 2017.
13. D. Bourque, CUDA-Accelerated Visual SLAM For UAVs, (Unpublished Master Thesis). Worcester Polytechnic Institute. 2017.
14. Jose, E., Greeshma, M., TP, M.H. and Supriya, M.H., March. Face recognition based surveillance system

using facenet and mtcnn on jetson tx2. 5th Int. Conf. on Advanced Computing & Communication Systems (ICACCS) (pp. 608-613). IEEE. 2019.

15. Giubilato, R., Chiodini, S., Pertile, M. and D., S., An evaluation of ROS-compatible stereo visual SLAM methods on a nVidia Jetson TX2. Measurement, 140, pp.161-170. 2019.

16. Van Essen, B., Macaraeg, C., Gokhale, M. and Prenger, R., Accelerating a random forest classifier: Multi-core, GP-GPU, or FPGA. 20th International Symposium on Field-Programmable Custom Computing Machines (pp. 232-239). 2012.

17. Jones, D.H., Powell, A., Bouganis, C.S. and Cheung, P.Y., GPU versus FPGA for high productivity computing. International Conference on Field Programmable Logic and Applications (pp. 119-124). IEEE. 2010, August.

18. Nurvitadhi, E., Venkatesh, G., Sim, J., Marr, D., Huang, R., Ong Gee Hock, J., Liew, Y.T., Srivatsan, K., Moss, D., Subhaschandra, S. and Boudoukh, G., Can FPGAs beat GPUs in accelerating next-generation deep neural networks?. In Proceedings of the 2017 ACM/SIGDA Int. Symposium on Field-Programmable Gate Arrays (pp. 5-14). 2017, February.

19. Nurvitadhi, E., Sim, J., Sheffield, D., Mishra, A., Krishnan, S. and Marr, D., Accelerating recurrent neural networks in analytics servers: Comparison of FPGA, CPU, GPU, and ASIC. 26th International Conference on Field Programmable Logic and Applications (FPL) (pp. 1-4). IEEE. 2016, August.

20. CUDA C Programming Guide, http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html (Accessed : 22.05.2020)

21. Paralution Benchmark Suite. https://developer.nvidia.com/paralution, (Accessed: 22.05.2020)

22. Danalis, A., Marin, G., McCurdy, C., Meredith, J.S., Roth, P.C., Spafford, K., Tipparaju, V. and Vetter, J.S., March. SHOC benchmark suite. 3rd Workshop on GPGPU (pp. 63-74). 2010.

23. GeForce Titan X Specifications, http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x/specifications (Acessed : 22.05.2020)

24. Jetson TX2 Module Data Sheet. https://developer.nvidia.com/embedded/jetson-tx2 (Acessed : 22.05.2020)

25. EVGA GeForce GTX TITAN X(12G-P4-2990-KR) on Amazon.com , https://www.amazon.com/dp/B07MK6CWLR/ref=dp_cr_wdg_tit_rfb (Accessed : 22.05.2020)

26. NVIDIA Jetson TX2 Development Kit on Amazon.com,https://www.amazon.com/B06XPFH939 (Accessed : 22.05.2020)

27. Matrix Market, (Accessed: 22.05.2020) http://math.nist.gov/MatrixMarket/

28. The SuiteSparse Matrix Collection, https://www.cise.ufl.edu/research/sparse/matrices/ (Accessed : 22.05.2020)

29. Mittal, Sparsh. "A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform." Journal of Systems Architecture 97 (2019): 428-442.

30. Cui, Han, and Naim Dahnoun. "Real-Time Stereo Vision Implementation on Nvidia Jetson TX2." In 2019 8th Mediterranean Conference on Embedded Computing (MECO), pp. 1-5. IEEE, 2019