Parallel Computing: Technology Trends
I. Foster et al. (Eds.)

doi:10.3233/APC200091

Performance, Power Consumption and Thermal Behavioral Evaluation of the DGX-2 Platform

Matej SPETKO [a,1], Lubomir RIHA [a] and Branislav JANSIK [a]
[a] *IT4Innovations National Supercomputing Center, VŠB – Technical University of Ostrava, Ostrava, Czech Republic*

Abstract. In this paper, we evaluate the performance, power consumption and its variation and also thermal behavior of the DGX-2 server from Nvidia. We present a development of specialized synthetic benchmarks to measure raw performance of GPUs for single, double, half precision and also Tensor Core units. With these benchmarks, we were able to reach peak performance and verify the specification provided by Nvidia. We achieved 130.79 TFLOPS peak performance in half-precision on Tensor Cores. We also measured the thermal stability of the DGX-2 system. It can hold its peak performance when all 16 GPUs are fully loaded except Tensor Core workload, when thermal throttling occurred with with up to 1 % performance penalty. During single-precision workload we observed 23 % variation of the power consummation of individual GPUs installed in the system. Finally, we have evaluated the behavior of the Tesla V100-SXM3 chip under the DVFS tuning. Running at optimal frequency, the compute bound workload can save up to 39% energy while the run-time increases by 51 %. More importantly, memory bound workload can save up to 31 % with 2 % throughput penalty and during the communication over NVLink one can save up to 26 % energy with no penalty.

Keywords. DGX-2, tensor core, performance analysis, energy efficiency, dynamic voltage and frequency scaling (DVFS)

1. Introduction

In this paper, we evaluate the performance of the Nvidia DGX-2 system using a new synthetic benchmark, designed to achieve and measure the peak performance of both CPUs as well as Nvidia GPUs. For this paper, we have developed a new version of this benchmark with support for Tensor Cores [1]. With our benchmark, we were able to match V100-SXM3's peak performance stated by Nvidia. In adition, we measured GPU memory and NVLink throughput.

Research in related work is focused on different aspects of DGX-2 system. For instance, *Dissecting the NVIDIA Volta GPU Architecture via Microbenchmarking* [2] is more focused on the V100 GPU architecture. This work explores deeply the whole V100 memory hierarchy, including throughput and latency measurements. It also inspects native

[1]Corresponding Author: IT4Innovations National Supercomputing Center, VŠB – Technical University of Ostrava, 17. listopadu 15/2172, 708 33 Ostrava – Poruba, Czech Republic; E-mail: matej.spetko@vsb.cz.

Volta instructions with issue latency measurements. Furthermore, *Evaluating Modern GPU Interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect* [3] focuses on GPU communication technologies. It analyses aspects like throughput, latency and topology of different GPU interconnects that are used on several GPU servers, including the DGX-2.

The goal of this paper is to evaluate the thermal stability and GPU power consumption. Moreover we performed dynamic frequency and voltage scaling (DVFS) for compute bound, memory bound and communication workloads and stated the most efficient configuration for these workload types. In the end we also evaluate power consumption of the whole node.

1.1. DGX-2 platform description

The main focus of the DGX-2 server is to accelerate tasks in artificial intelligence. However, it is well-suited to run any GPU or multi-GPU application. It contains 16 Tesla V100-SXM3 GPUs interconnected with high speed NVLink interconnect [4]. It also features a pair of Intel Xeon Platinum 8168 CPUs, 1.5 TB of memory and 30 TB of fast NVMe SSD storage. The server can be equipped with either eight EDR Infiniband or 100 Gb Ethernet network cards. [5] The GPUs are spread across two trays, each containing 8 GPUs in two rows. Cooling fans are located at the start of the tray as shown in Figure 1.

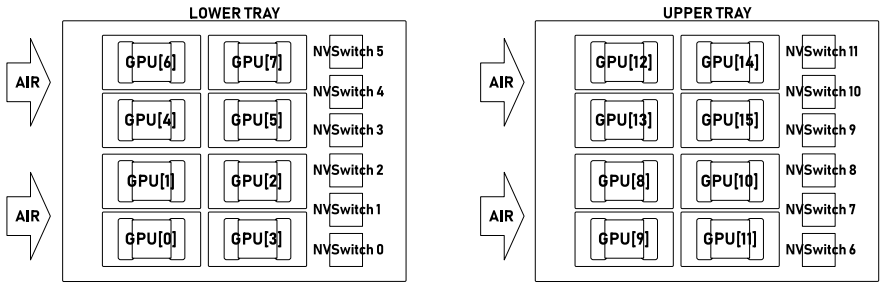

Figure 1. Physical GPU layout of the DGX-2 server.

The V100-SXM3 GPU is equipped with 80 streaming multiprocessors (SMs) and 32 GB of HBM2 memory. Each SM consists of these processing units: 64 FP32 (float), 64 INT32 (32 bit integer), 32 FP64 (double precision) and 8 Tensor Cores (16 bit floating point – half precision). [4]. The basic operation with 16 bit floating point data type – half is performed on floating point units. It can also perform half2 vector operations and reach double the performance of float.

GPUs on DGX-2 system are interconnected with hi-speed bus called NVLink in version 2. Single NVLink-V2 link can provide 25 GBps throughput in single direction and 50 GBps in both directions. The system is also equipped with 12 NVSwitches. Each GPU is connected to six of these switches with single NVLink-V2 link, providing 300 GBps bidirectional peer-to-peer (P2P) throughput. [3]

The Volta architecture introduce Tensor Cores – processing units designed to perform fused multiply-add operation with 16×16 half precision matrices. The result accumulation can be done either in half-precision or in single-precision. The programmer can access `mma_sync()` function which performs a warp level operation: every thread of warp is participating in the matrix multiplication. [1]

2. Measurement Methodology Description

2.1. Benchmarks

The Mandelbrot benchmark is designed to measure pure floating point performance of the processor at very high arithmetic intensity. It executes the Mandelbrot iterations $z_{k+1,i} = z_{ki}^2 + c_i$ where $z_{0i} = 0$ and the constants c_i are from the Mandelbrot set of complex numbers. The Mandelbrot iterations may be repeated indefinitely and remain bounded. For simplicity and efficiency, we select the constants c_i only from the numbers on the real axis. The benchmark is implemented in CUDA PTX assembly code [6]. Each thread on the GPU device is initialized with eight unique constants c_i. We use 32 threads per block and 12 blocks per streaming multiprocessor. After the initialization, all computation runs in the registers only, avoiding any references to memory. The loop over k updates all values of z_{ki} using FMA instructions. Further, the loop is unrolled 100 times, counting 800 consecutive fused mulitply-add instructions, in order to out-weight the loop overhead of three instructions. The loop counter runs one million times to vastly outrun the clock granularity and provide reliable performance measurements. The measurement may be repeated number of times. The arithmetic intensity of the benchmark is 12.5×10^6 FLOP per byte in double precision and up to quadruple of that in single and half precision.

The Mandelbrot benchmark may be naturally extended to matrix domain. In matrix form, the square matrix Z is updated as $Z_{k+1} = Z_k * Z_k + C$, where the $*$ refers to matrix-matrix multiplication, the matrix $Z_0 = \mathbf{0}$ and the square matrix C has eigenvalues from the Mandelbrot set. Such matrix iterations may be repeated indefinitely and the matrix Z will remain bounded. It would be natural to use the matrix Mandelbrot iterations as a load to benchmark the Tensor Cores. However the WMMA interface does not allow to insert the output of the WMMA instruction as an input into the next WMMA instruction directly due to the fact that the matrix fragments held by individual thread registers are not identical for input and output matrices. Reusing the output registers as input registers into the WMMA instruction introduces permutations into the matrix, in addition some matrix elements are repeated and some are lost. Nevertheless, the l2 norm of the matrices created in this way remains approximately correct. Recognizing that the reuse of the output registers as input registers to WMMA instructions approximately conserves the l2 norm and using the property of sub-additivity and sub-multiplicativity of the l2 norm, we are able to select the C as real valued, random matrices, taken such that their eigenvalues lie well within the bounds of the Mandelbrot set and the matrix iterations remain bounded indefinitely. Utilizing this result, we have implemented the matrix Mandelbrot benchmark for the Tensor Cores, using the PTX WMMA instructions API [6]. The data are kept in the registers only, the Z and C being 16 bit floating point 16×16 matrices. Each block is initialized to unique C matrix. The C matrices are pre-computed off the benchmark code, by shifting and scaling a randomly generated square matrices. The block count, loop unrolling and loop count remains the same as for the scalar version. The arithmetic intensity exceeds millions of floating point operations per byte. The arithmetic intensity of the matrix benchmark for the Tensor Cores is 1.6×10^9 FLOP per byte. [7]

The throughput of the memory subsystem was measured by STREAM [8] benchmark, modified for GPUs, also available at the GIT repository [7]. All functions of the STREAM benchmark were measured: copy, scale, add, triadd. The throughput of NVLink interconnect was measured by performing peer-to-peer (P2P) data transfer between two

GPUs with `cudaMemcpyPeerAsync()` call. [9] The throughput was measured in single direction as well as bidirectionally.

2.2. Frequency Scaling and Energy Measurement

To simulate compute bound workload, we took our Mandelbrot benchmark. On the other hand, memory bound workload is represented by the STREAM benchmark. Furthermore, measurement of P2P data transfer over NVLink was also performed. Energy measurement and frequency scaling was performed using tools provided by Nvidia. To measure energy consumption, the Nvidia Management Library – `NVML` was used. For the frequency scaling and taking samples with power, temperature and frequency the `nvidia-smi` utility was used. In this paper, we use the same methodology to measure energy efficiency as described in the Green500 tutorial [10] with the exception that we use our Mandelbrot benchmark to determine peak performance instead of Linpack benchmark.

`NVML` provides C-based programmatic interface for monitoring and managing Nvidia GPUs. It is intended to be a platform for building third party applications. During the experiments, `NVML` was used to access a total energy consumption counter for the GPU. This counter can be accessed with `nvmlDeviceGetTotalEnergyConsumption()` function call [11]. To measure energy consumed by certain workload the value of this counter was read two times: right before launch and right after it finishes. Subtracting these values yield energy consumed by the workload in mJ. Application initialization and cleanup is not included in this measurement, only the main loop with the measured workload.

The Nvidia System Management Interface: `nvidia-smi` was used to collect power, temperature and frequency samples to analyze power and thermal properties. These samples were captured at approximately 100 Hz sampling rate. Furthermore this utility was used to change frequency of GPUs. The frequency was decreased from 1597 MHz to 675 MHz in approximately 7 MHz predefined steps. HBM2 memory frequency cannot be tuned, thus staying at 958 MHz even when the card is idle.

3. Results

3.1. Performance

We have not found any peak performance numbers published for V100-SXM3 GPU used in DGX-2 server. However, we were able to retrieve these numbers from Nvidia Profiler. The performance of Tensor Cores is not stated for this version of V100 GPU. V100-SXM2 revision has Tensor Core peak performance of 125 TFLOPS in the half-precision, running at 1530 MHz [4]. If we scale up this number to match SXM3's 1597 MHz, we should be getting 130.484 TFLOPS in half-precision. Global memory bandwidth is according to Nvidia Profiler 980.992 GBps. NVLink's unidirectional P2P throughput is 150 GBps and 300 GBps in both directions. Results of Mandelbrot benchmark, STREAM benchmark and NVLink P2P transfer benchmark are shown in Table 1.

3.2. Power and Thermal Properties

The physical layout of the DGX-2 server causes that cold air is not distributed equally among all the GPUs. The GPUs are placed in two trays, where each tray contains 8 GPUs

Mandelbrot benchmark		
	Specification [TFLOPS]	Measurement [TFLOPS]
double	8.177	8.1765
float	16.353	16.3530
half2	32.707	32.7038
tensor	130.484	130.7928

STREAM benchmark	
	Throughput [GBps]
copy	825.473
scale	826.518
add	873.631
triadd	872.368

NVLink P2P transfer	
latency	2.45 us
unidirectional	145.16 GBps
bidirectional	266.46 GBps

Table 1. Table on the left compares performance measured by Mandelbrot benchmark to the performance specified by Nvidia. Table in the middle shows memory throughput measured by STREAM benchmark. Table on the right shows the result of P2P data transfer over NVLink interconnect.

in two rows. High-RPM cooling fans are located at the beginning of these trays. GPUs placed in the first row are facing these fans directly and receive cold air, while GPUs in the second row receive air that has been already heated by the GPUs in the first row.

In general, this causes that GPUs in the second row run at higher temperature than the ones in the first row. This also means that they can reach their TDP of 350 W when they are under the full load and thermal throttling must be performed, which results in performance imbalance among the GPUs. Figure 2 shows how GPUs in the first row influence GPUs in the second row by running Mandelbrot benchmark on Tensor Cores for 4 minutes on all 16 GPUs one by one.

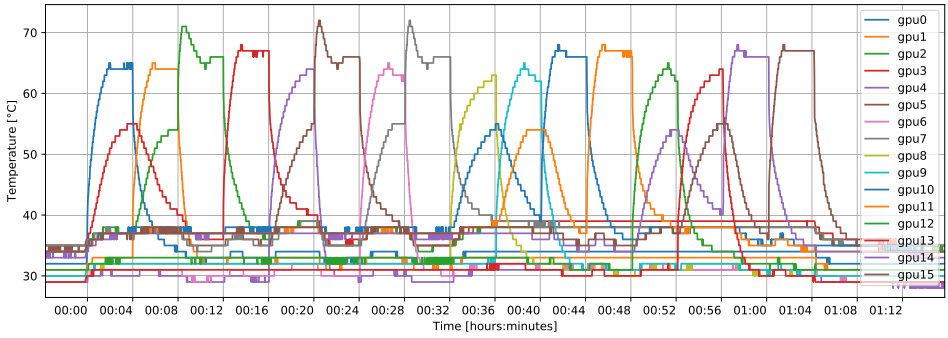

Figure 2. Temperature of all GPUs when fully loaded with Tensor core benchmark one by one. Each GPU was loaded for approximately 4 minutes. GPU in the first row (0, 1, 4, 6, 8, 9, 12, 13) increases also temperature of the GPU located directly behind it in the second row (2, 3, 5, 7, 10, 11, 14, 15).

When running Tensor Core Mandelbrot benchmark on all 16 GPUs at once, GPUs in the front row reach a maximum temperature of 57 °C while GPUs in the second row peak is 72 °C. During this benchmark, certain GPUs from the second row tend to throttle down their frequencies to as low as 1575 MHz (from the maximum 1597 MHz) causing approximately 1 % performance loss, see Figure 3. It shows that running the same Mandelbrot benchmark on all the GPUs results in significant variation in power consumption of individual GPUs, reaching up to 23 % for single precision version. This is caused by both their location in the server as well as their manufacturing variations. We can also observe that for single precision, double precision and Tensor Core version, when some GPUs reach the TDP, they under-clock their frequencies.

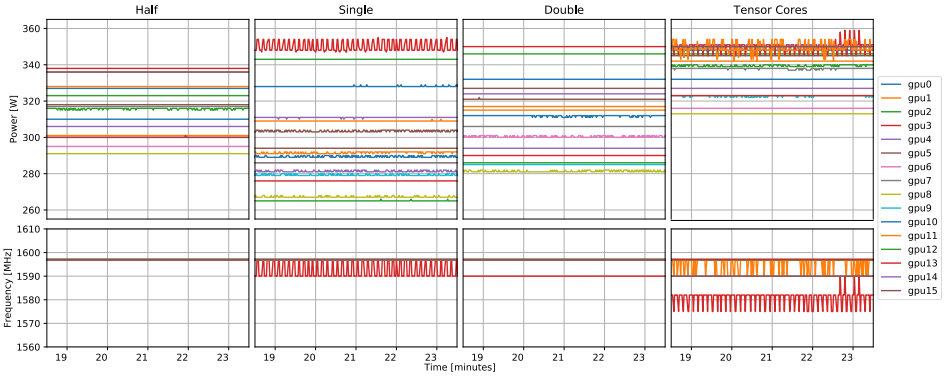

Figure 3. Power consumption variation of all the GPUs in the DGX-2 when under full load using compute bound workload with four different data-types. The variation for the single-precision workload is up to 23%.

3.3. Frequency Scaling

To determine whether we can get better energy efficiency of the Tesla V100-SXM3, we have performed the DVFS tuning test for compute bound, memory bound and NVLink workloads. The first frequency scaling test was performed for all the data types of the Mandelbrot benchmark (float, double, half2, tensor). Benchmark in each data type performed the same amount of the floating-point operations: 81920×10^9. The frequency was scaled from 1597 MHz to 675 MHz in approximately 7 MHz steps. Each frequency step was measured 6 times and average value is reported. Heat up runs were performed before the actual measurement.

Figure 4 shows the result of the frequency scaling. The two bottom plots display the same data in logarithmic scale as the top two plots in linear scale. Table 2 compares runs at base frequency 1597 MHz with the runs at the most energy efficient frequency for each workload type.

In general, the most efficient frequency for Mandelbrot benchmark is 1057 MHz. Running at this frequency we can save up to 39 % of the energy while the run-time will increase by 51 %. Interesting number to point out is the energy efficiency of double data type. Running at base frequency it achieves 24.8 GFLOPS/W whereas running at 1050 MHz the efficiency reaches 40.67 GFLOPS/W. This efficiency number is getting close to 50 GFLOPS/W, which is the limit for building exascale system with 20 MW power consumption [12]. On the other hand, the peak performance at this frequency is only 5.37 TFLOPS which is 66 % of the original 8.17 TFLOPS.

The second frequency scaling test was done using the STREAM benchmark. During this experiment each workload transferred the same amount of data: 7.924 TB. Each frequency step was measured 6 times and average value is reported. Before the measurement started heat up runs were performed.

The results of the frequency scaling of the STREAM benchmark are shown in the Figure 5. The peak throughput achieved during this experiment is lower than in the subsection 3.1, because we were measuring the average throughput and not the best case like the original STREAM does. Furthermore, the Figure 5 also shows a staircase shape when the frequency is lower than 1 GHz. This is probably caused by the GPU having certain memory operation modes. These modes do not match the granularity of which the

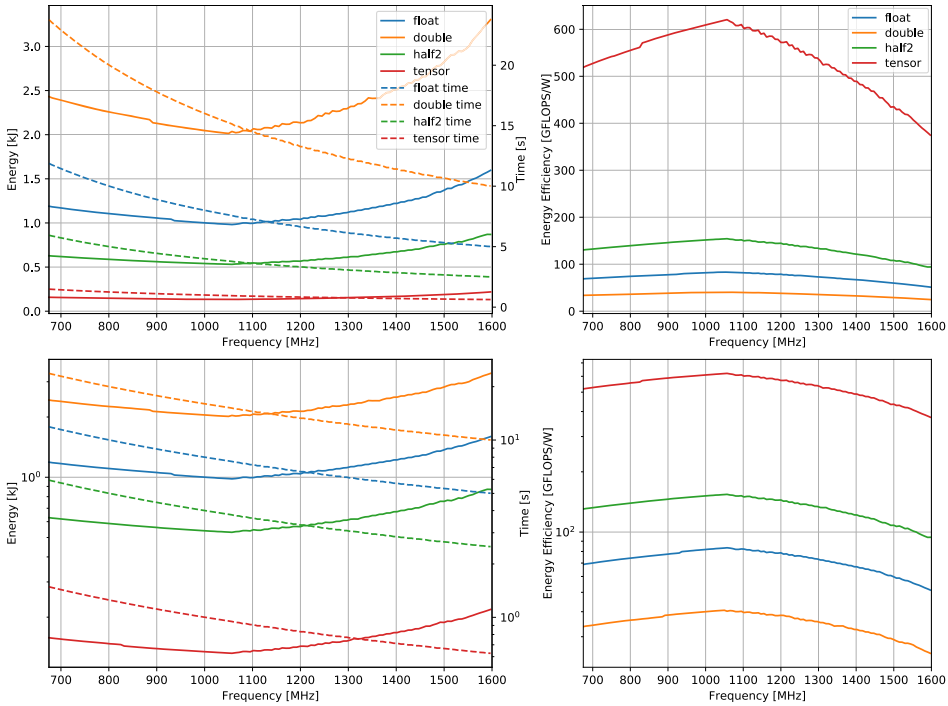

Figure 4. Frequency scaling of Mandelbrot benchmark. Plot in the top left corner shows consumed energy and run-time of workload. Plot in the top right corner shows energy efficiency. The two plots at the bottom display the same data in logarithmic scale.

	Frequency [MHz]	Time [s]	Time difference	Energy [J]	Energy savings	Performance [TFLOPS]	Energy efficiency [GFLOPS/W]
double	1597	10.02		3303		8.17	24.80
	1050	15.25	152.16%	2015	39.01%	5.37	40.67
float	1597	5.01		1596		16.34	51.33
	1057	7.57	150.99%	982	38.50%	10.82	83.46
half2	1597	2.51		870		32.69	94.18
	1057	3.78	151.05%	531	38.97%	21.64	154.30
tensor	1597	0.63		219		130.65	374.90
	1057	0.95	151.04%	132	39.58%	86.50	620.51

Table 2. Mandelbrot benchmark running at base frequency compared to the most efficient frequency for each workload.

streaming multiprocessor can change its frequency. The result of the energy consumption for base frequency and the most efficient frequency is shown in Table 3. On average, up to 31 % of the energy can be saved by scaling down to 1005 MHz. By doing that, the transfer time increased by 2 % which is almost identical in compare to the data transfer at the base frequency.

The last frequency scaling experiment was done for unidirectional P2P data transfer over NVLink. The amount of transferred data was 859 GB. One frequency step was

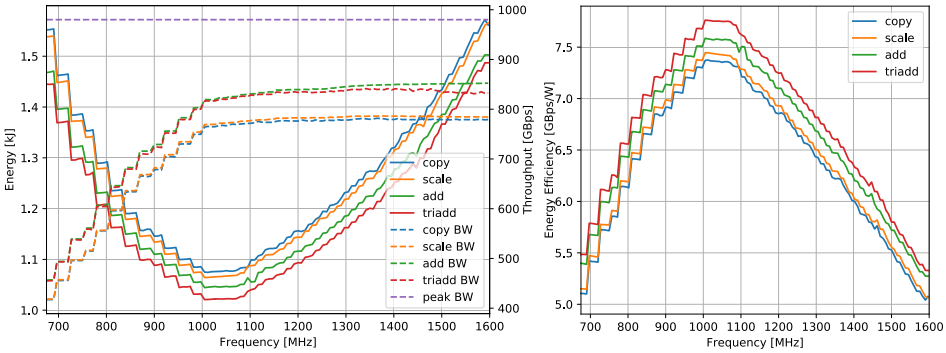

Figure 5. Frequency scaling of the STREAM benchmark. Plot on the left shows consumed energy and throughput. Plot on the right shows the energy efficiency.

	Frequency [MHz]	Time [s]	Time difference	Energy [J]	Energy savings	Throughput [GBps]	Energy efficiency [GBps/W]
copy	1597	10.17		1561		779.12	5.08
	1012	10.35	101.78%	1074	31.18%	765.47	7.38
scale	1597	10.10		1566		784.43	5.06
	1005	10.30	101.99%	1064	32.02%	769.14	7.45
add	1597	9.30		1503		852.27	5.27
	1005	9.67	104.02%	1044	30.49%	819.31	7.59
triadd	1597	9.52		1487		832.54	5.33
	1005	9.71	101.98%	1021	31.36%	816.40	7.76

Table 3. STREAM benchmark running at base frequency compared to the most efficient frequency for each workload.

measured 10 times. Figure 6 shows the result of this experiment. Running at 1140 MHz can save up to 26 % energy without any throughput penalty. The throughput starts to drop when the frequency decreases from 1140 MHz. In addition, the staircase shape similar to Figure 5 can be seen. Table 4 shows the most efficient frequency for source and receive device and compares it to the base frequency.

	Frequency [MHz]	Time [s]	Time difference	Energy [J]	Energy savings	Throughput [GBps]	Energy efficiency [GBps/W]
SRC DEV	1597	5.93		563		144.90	1.51
	1110	6.19	104.34%	421	25.22%	138.88	2.04
RCV DEV	1597	5.93		569		144.90	1.53
	1140	5.94	100.15%	417	26.71%	144.69	2.08

Table 4. NVLink P2P transfer benchmark running at base frequency compared to the most efficient frequency for source device and receive device.

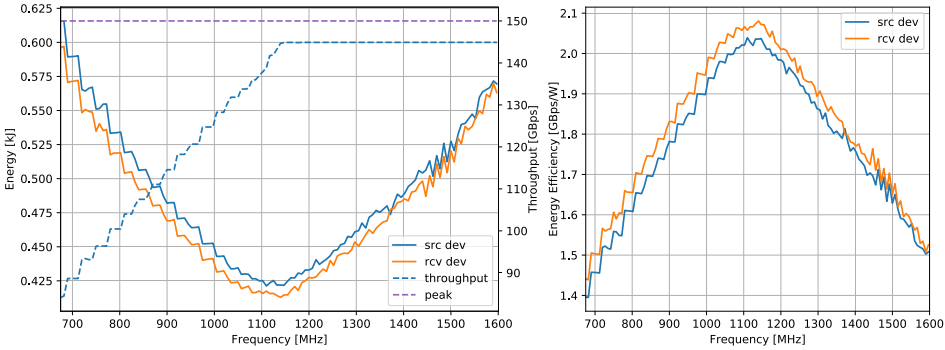

Figure 6. Frequency scaling of the NVLink P2P transfer benchmark. Plot on the left shows consumed energy and throughput. Plot on the right shows the energy efficiency.

3.4. Overall power consumption of DGX-2 server

To supplement the overall picture of DGX-2 efficiency, we also need to look at the energy consumption of the server as a whole. Unfortunately, we measured these numbers only with one power sample because the administrative privileges are needed to retrieve them. Nevertheless, they can give some idea about the efficiency and power consumption of the whole node including all peripherals and cooling. These power consumption numbers were retrieved using `ipmitool` utility.

When idle, the DGX-2 consumes 2340 W, GPUs altogether consumes 768 W. When loaded with Tensor Core Mandelbrot benchmark at 1597 MHz, the consumption rises to 7254 W whereas GPUs alone consume 5340 W. When running the same benchmark at 1057 MHz, the whole node consumption drops to 4056 W and GPUs consume 2248 W.

When running double precision Mandelbrot benchmark at base frequency GPUs consume 4936 W and the whole node 6708 W. At this frequency the server reaches 130.8 TFLOPS, meaning the performance per watt reaches 19.52 GFLOPS/W. When we scale down the frequency to 1057 MHz, GPUs alone consume 2116 W. Consumption of the whole node drops to 3666 W. As a result, the DGX-2 can achieve efficiency of 23.60 GFLOPS/W at this frequency but the performance drops to 86.4 TFLOPS.

4. Conclusion

We have developed a set of benchmarks to determine the raw performance of GPUs in the Nvidia DGX-2 server. We verified that performance numbers of V100-SXM3 GPU are according to specification. We were able to reach 130.79 TFLOPS in half precision using Tensor Cores on a single GPU. When running full load on all 16 GPUs at the same time, some of the GPUs may thermal throttle by 1 % due to an uneven cooling solution and manufacturing variations. We observed 23 % variation in power consumption fo GPUs when running float Mandelbrot benchmark. To get the best performance per watt out of V100-SXM3 GPU, it makes sense to scale down the frequency. For compute bound workload the most efficient frequency is 1057 MHz making 39 % energy savings while run-time is increased by 51 %. The energy efficiency achievable for double precision workload is 40.67 GFLOPS/W whereas running at base frequency is

only 24.8 GFLOPS/W. Memory bound workload has its sweet spot at 1005 MHz. At this frequency, the throughput penalty is only 2 % while energy savings can reach 31 %. Peer-to-peer transfer achieves the best energy efficiency at 1140 MHz frequency, being able to save 26 % energy without any throughput penalty. The whole DGX-2 node in the idle mode consumes 2.3 kW of power. When all 16 GPUs are loaded with double precision workload, the consumption increases to 6.7 kW with 19.52 GFLOPS/W energy efficiency. However, running the same workload at 1057 MHz it consumes 3.6 kW, having the energy efficiency 23.60 GFLOPS/W.

5. Acknowledgments

This work was supported by The Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project IT4Innovations National Supercomputing Center LM2015070.

This work was also partially supported by the SGC grant No. SP2019/59 "Infrastructure research and development of HPC libraries and tools", VŠB – Technical University of Ostrava, Czech Republic.

References

[1] M. A. Raihan, N. Goli, and T. M. Aamodt, "Modeling deep learning accelerator enabled gpus," in *2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 79–92, March 2019.

[2] Z. Jia, M. Maggioni, B. Staiger, and D. P. Scarpazza, "Dissecting the nvidia volta gpu architecture via microbenchmarking," *ArXiv*, vol. abs/1804.06826, 2018.

[3] A. Li, S. Song, J. Chen, J. Li, X. Liu, N. R. Tallent, and K. J. Barker, "Evaluating modern gpu interconnect: Pcie, nvlink, nv-sli, nvswitch and gpudirect," *ArXiv*, vol. abs/1903.04611, 2019.

[4] NVIDIA Corp., "NVIDIA TESLA V100 GPU ARCHITECTURE." http://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf, August 2017. "[Online; accessed 2019-07-12]".

[5] NVIDIA Corp., "DGX-2/2H SYSTEM User Guide." https://docs.nvidia.com/dgx/pdf/dgx2-user-guide.pdf, May 2019. "[Online; accessed 2019-07-15]".

[6] NVIDIA Corp., "PARALLEL THREAD EXECUTION ISA." https://docs.nvidia.com/cuda/pdf/ptx_isa_6.4.pdf, May 2019. "[Online; accessed 2019-07-12]".

[7] IT4Innovations, "Mandelbrot CPU benchmark." https://code.it4i.cz/jansik/mandelbrot. "[Online; accessed 2019-07-12]".

[8] J. D. McCalpin, "Memory bandwidth and machine balance in current high performance computers," *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pp. 19–25, Dec. 1995.

[9] NVIDIA Corp., "CUDA RUNTIME API." docs.nvidia.com/pdf/CUDA_Runtime_API.pdf, July 2019. "[Online; accessed 2019-09-24]".

[10] R. Ge, X. Feng, H. Pyla, K. Cameron, and W. Feng, "Power measurement tutorial for the green500 list." https://www.top500.org/files/green500/tutorial.pdf, June 2007. "[Online; accessed 2019-09-24]".

[11] NVIDIA Corp., "NVML Reference Manual." https://docs.nvidia.com/pdf/NVML_API_Reference_Guide.pdf, August 2019. "[Online; accessed 2019-09-16]".

[12] K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Karp, S. Keckler, D. Klein, R. Lucas, M. Richards, A. Scarpelli, S. Scott, A. Snavely, T. Sterling, R. S. Williams, K. Yelick, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Keckler, D. Klein, P. Kogge, R. S. Williams, and K. Yelick, "Exascale computing study: Technology challenges in achieving exascale systems," 2008.