PL-3

Evolution of the GPU Device widely used in AI and Massive Parallel Processing

Toru Baji

NVIDIA *(Japan)*

ATT New Tower 13F, 2-11-7 Akasaka, Minato-ku, Tokyo 107-0052, Japan

tbaji@nvidia.com

Abstract

While the CPU performance cannot benefits anymore from Moore's law, GPU (Graphic Processing Unit) still continue to increase its performance 1.5times/year. From this reason, GPU is now widely used not only for computer graphics but also for massive parallel processing and AI (Artificial Intelligence). In this paper, the details of this continuous performance growth, the constant evolution in transistors count and die size, and the scalable GPU architecture will be described.

(Keywords: GPU, Die Size, Transistor Count, Moore's Law, AI, GPU Computing, SoC)

Introduction

Two types of programmable processors exist in the market. One is the conventional CPU and the other is GPU. Typical CPU is composed of 1 to 8 cores while GPU has thousands of cores. CPU is good for sequential processing, while GPU is good to accelerate software with massive parallel executions.

GPU was initially dedicated for 3D graphics. However from 2006, when GPU started to apply general-purpose cores, GPU architecture became capable to be used as a general purpose massive-parallel processor. NVIDIA developed a software framework CUDA (Compute Unified Device Architecture) that make it possible to easily program the GPU for these application.

Recently two key technologies are highlighted in the industry. AI and Autonomous Driving Cars. AI requires a massive parallel operation to train many-layers of neural networks for big training data. Some of them have more than 100 layers. With CPU alone, it was impossible to finish the training in a practical time. The latest multi-GPU system with GV100 makes it possible to finish the training and inference of complex AI in 1/10 ~ 1/100 time of CPU.

For the autonomous driving cars, 10-100 TOPS of performance is required to implement perception, localization, path planning processing and again SoC with integrated GPU will play a key role there.

In this paper, the evolution of the GPU which is one of the biggest commercial devices requiring state-of-the-art fabrication technology will be introduced.

CPU performance growth saturation and multi-CPU core limitation

Thanks to the Moore's Law that predicted the continuous evolution of the integration level (number of transistor will double every 1.5years), the performance of the CPU was increasing constantly according to that law. Fig. 1 shows that initially the CPU performance grew almost 1.5 times every year. However around the year 2010, this performance increase has been reduced to 1.1 times per year when the clock frequency came to a GHz order. The main reasons are the increasing leakage current that prohibited further transistor speed increase, and the logic complexity by applying more sophisticated computer architecture.

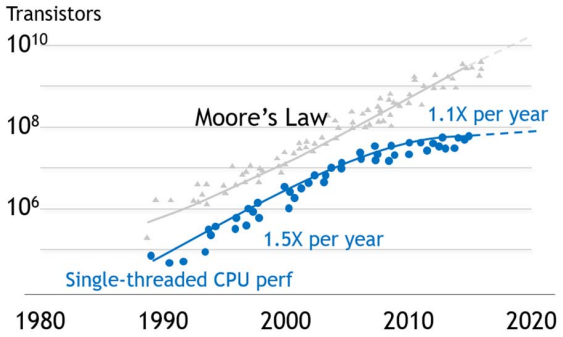


Fig. 1: 40 Years of CPU Performance Trend Data [1]

Then the multi-CPU architecture started to be introduced. Although each CPU clock frequency are limited to a few GHz, the total performance can be increased if multiple threads can be executed in parallel. This works very well when you have few threads to run in parallel like the PC OS and some applications. However the numbers of threads to be executed in popular pc systems are limited, and even if you further increase the number of CPU cores, those cores are not effectively used. Fig.2 shows this limitation. Amdahl's Law is describing the efficiency of N-Core system depending on the ratio of the sequential processing (R) in the application program. If you have just 20% of the application program that should be executed sequentially (and not parallel), then even if you have 8 CPU cores, the efficiency will be

978-1-5386-3712-8/18/$31.00 ©2018 IEEE

limited to 3.3. For this reason, the popular commercial available CPUs are mainly up to 4-8 cores even today.

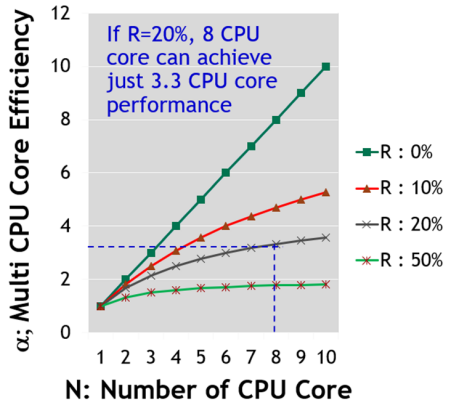


Fig.2 Amdahl's Law limits multi-CPU Efficiency

$$Tb = Ta * (R + (1 - R)/N)$$
$$\boldsymbol{a = Ta/Tb = 1 / (R + (1 - R)/N)}$$

N: Number of CPU Cores
R: Ratio of Sequential Processing
1- R: Ratio of Parallel Processing
a: Multi CPU Core Efficiency
(a =1 is equivalent to a single CPU)
Ta: Single CPU Execution Time
Tb: N CPU Core Execution Time

GPU for Massive-Parallel Computation

GPU was originally used for computer graphics. These applications are inherently parallel. Each processed vertices, polygons and pixels are data independent and can be executed in parallel. The processing speed is almost linearly proportional to the number of processing cores. To increase the quality of the 3D graphics, which is almost in a photo realistic level today, a big revolution occurred in the graphics architecture in 2006. Until then, the programmable GPU cores were dedicated for vertices and pixels. From 2006 those became a general-purpose processing cores and various algorithms can be executed on the same core.

Besides graphics application, there are many applications that required massive parallel processing like those shown in Fig.3. Actually those applications are executed by GPU today. As the latest news, GPU computing also made a great contribution in two Nobel prizes in 2017, Chemistry prize: Cryogenic Electron Microscopy and Physics prize: Detection of Gravitational Waves.

Fig.3 Many Application requires Massive-Parallel Computation

NVIDIA noticed the potential of this revolutionary GPU architecture. In parallel with the unified-shader GPU hardware development, NVIDIA developed the massive-parallel programming foundation "Compute Unified Device Architecture (CUDA)" [1]. Scientists and programmers can now use the full power of GPU using some simple extensions to their familiar programming language C, C++,

GPU dramatically increased its numbers of cores especially from the introduction of the unified-shader. The latest Volta architecture GPU V100 comes with 5,120 Single-Precision Floating Point FP32 Cores and 2,560 Double-Precision Floating Point FP64 Cores. This is a significant advantage to the CPU which has at most 64-72 cores in the Xeon Phi high-end family. Fig.4 shows the GPU performance advantage to the CPU. For the detail GPU architecture to achieve this level of performance, please refer the famous computer architecture book in the reference [2] .
GPU's applications are basically parallel and these are represented by R=0 (no sequential processing) in Fig.2 Amdah's law curve. That means that the multi core GPU efficiency is linear proportional to the number of GPU cores. Thanks to this fact, GPU can benefits 100% from the Moore's law or constant increase in the integration density. GPU performance growth is still keeping the pace of 1.5 times per year as shown in Fig. 4. Today, the performance gain against CPU is 10-100 times depending on the application. By the year 2025, this

is expected to be close to 1,000 times.

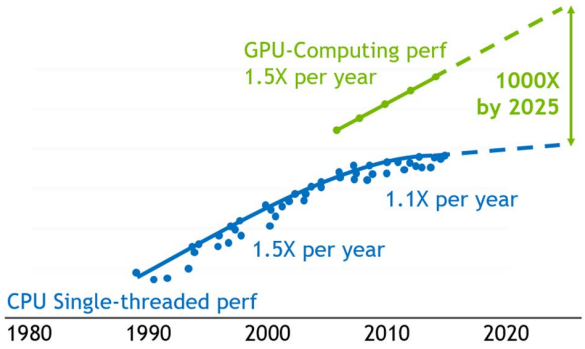


Fig. 4: GPU/CPU Performance Trend Data [1]

GPU Evolution towards the Highest-Performance and Biggest Die-Size Processor

NVIDIA flagship GPUs used to be one of the biggest die size chip for each generations of manufacturing technologies. It should use the big area to accommodate the necessary number of cores to achieve the high parallel computing performance requirements in the market.

Fig. 5 shows the increase in Transistor counts for the flagship GPUs in each generations. Especially from the graphics dedicated G70 to graphics/GPU computing capable G80 generation in 2006, the number of cores has been increased drastically from 32 cores to 128 cores, but the number of transistors just doubled. Now the highest level of integration is implemented by the Volta GPU GV100. It has 5,120 FP32 cores and 2,560 FP64 cores, 21.1-Billion transistors and the die size is 815mm^2 using 12nm Fin-FET technology.

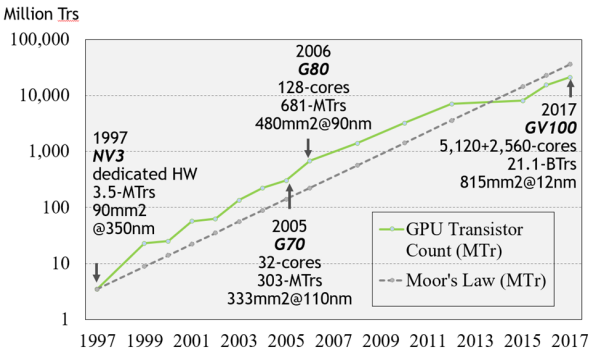


Fig.5 Evolution of GPU Cores, Tr and Chip Size

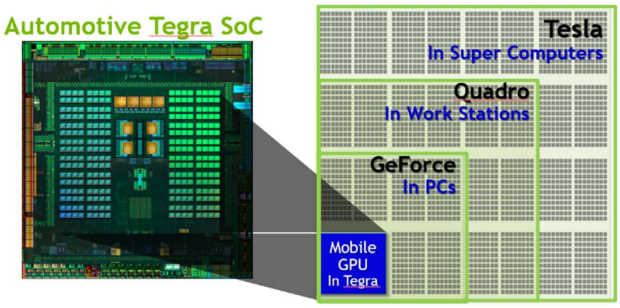
Moore's Law based Transistor count increase starting from the 1st generation GPU NV3 is also plotted by the dashed line. GPU Transistor counts is almost following the Moore's Law.

In the era of EUV Lithography that is considered to be applied from 5-7nm technology, the maximum chip size allowed for the lithography process might be reduced. Facing that problem, NVIDIA is making an advanced research of MCM (Multi-Chip-Module) GPU. [3]

One Scalable GPU Architecture

GPU is now widely used from supercomputers, workstations, consumer applications and autonomous driving. NVIDIA is using the same GPU architecture across all these products. Basically the number of cores are scaled. Therefore all the application software and programming environments can be shared. The latest autonomous driving SoC (System on a Chip) Xavier comes with 512 core Volta generation GPU and eight 64-bit CPU cores. It is quite a powerful chip with 7-Billion Transistors.

Fig.6 One Scalable GPU Architecture



Conclusion

GPU is now used not only for computer graphics but also for massive parallel processing and AI. To meet the strong demand for performance, GPU has been constantly increased its number of cores to the limit of the available maximum die size. This GPU architecture is scalable serving various applications from Supercomputers to autonomous driving.

References

[1] Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten. New plot and data collected for 2010-2015 by K. Rupp.

[2] Hennessy J, Patterson D., [Computer Architecture, A Quantitative Approach 5th Edition], Morgan Kaufmann Publishers, US, Chapter 4.4 (2012)

[3] Akhil Arunkumar et al, "MCM-GPU: Multi-Chip-Module GPUs for Continued Performance Scalability", ISCA 2017, June 24-28, 2017