Principle of GPU and Exploration of Multi-Cores GPUs System Performance

Meng Cao
School of Engineering& Applied Science
George Washington University
Washington DC, US
meng_cao@gwu.edu
GID: G26710740
Data: 03/13/2015

Contents

***Abstract*—General Purpose Graphics Processing Unit has already been widely used in high performance computers and workstations to accelerate those applications that require high graphic performance, such as the large 3D games and 3D design software. High performance GPU bring users the extremely shocked visual experience with smooth graphs and detailed textures. At the same time, users are constantly chasing a better GPU performance not only for the sense of achievement in hardware but also for the more verisimilar and visual pictures. Currently, GPUs play a significant role in the computer hardware, however during the process of chasing higher performance, we cannot solely rely on traditional strategies if considering the power of GPU and the cost performance. To deal with this issue, an alternative strategies that take the usage of parallelism thought, which is called multi-cores GPUs system, is proposed in recent years. To study the framework of multi-cores GPUs, we should have a comprehensive understanding of how does single GPU work. So this paper describes and explore the principle of single GPU, the basement of multi-core GPUs, the framework that used to connect and cooperate the multi-cores GPUs, and the performance test of it.**

Key words: single GPU, multi-core GPUs, parallel, performance

I. INTRODUCTION

In the early 1980s, ubiquitous interactive 3D graphics was still the stuff of science fiction and at that time the most popular GPU, MDA and MGA that produced by IBM (actually it is not an independently modular hardware as the GPU we mention today but it has the function of GPU) can only display very simple black and white bitmap graphics that were used to combine characters and quite simple rough pictures. At that time the main task of computers was dealing with logic and mathematic computation, and few application had the requirement of complicated graph processing service.

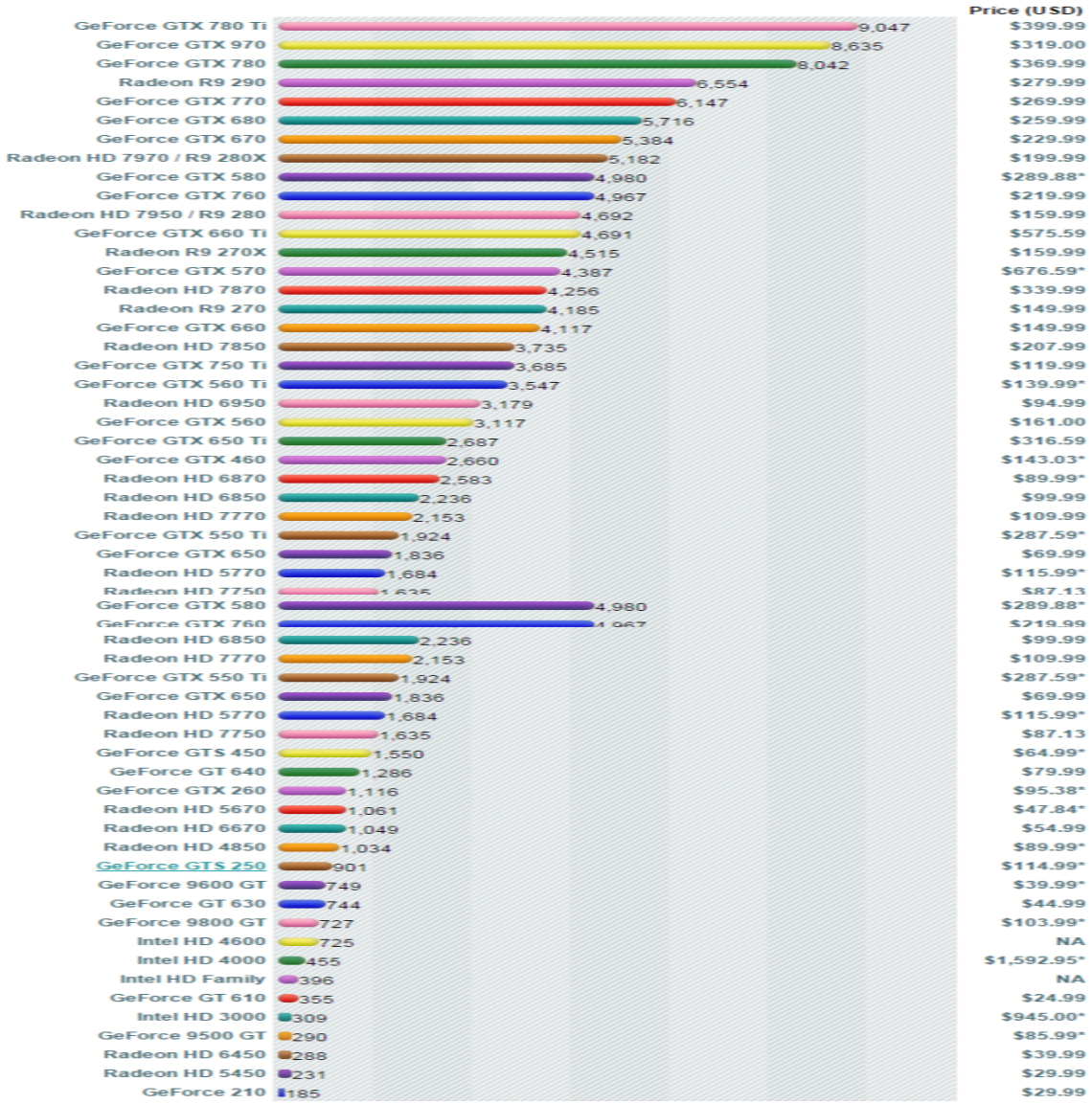


Fig. 1. This picture shows the rank of the performance and the price of single GPUs during the past few years. The performance is represented by the points that come from 3DMark software. And the price only shows the expensive and cheap relationship between cards.

However by the end of the 2000s, just after the rising of color monitors and graphic applications, nearly every new computer contained a graphics processing unit (GPU) which was dedicated to providing a high-performance of rich texture and perfect 3D experience. From that time, the high-speed development of computer hardware and software contributed to each other and made the performance of computer to a higher level constantly. The relation between GPUs and graphic-related applications is the best example. Advanced GPU makes computers possible to generate and display smooth, texture detailed graphs. On the other hand, these kind of graphs that generated from graphic related software such as those huge 3D games, bring people

into a new virtual visual world. From then, people start to chase more detailed and vivid graphs, which push and accelerate the development of hardware with an amazing high speed.

An evidence of this relation is that during the last ten years, performance of GPU increased so many times. The Fig.1 shows the rank of most of the popular GPUs in recently years. All of these GPUs were introduced to the market during 2009 to 2014. The points each GPU get came from the 3D mark software, a software special for evaluating the performance of GPU. From the rank, if we simply use the points to stand for the performance of each GPU, we can see that during the 6 years, the performance increase nearly 50 times.

To meet the need of those graphic applications, two tendency can be figured out easily during the development of GPUs in the high graphic performance required computers field. (1) One tendency is that GPU has already become the most expensive and significant part, and at the same time the bottleneck of performance in computers. Owing to the huge push of the floating point arithmetic and the graphic requirement from huge 3D games, a rapid growing of GPU performance happened in the recent 15 years and exactly at the same time, higher requirements is proposed. There seems no upper boundary to the needs of graphic processing power; even though the rapid increase in power, the demand is considerably more than what is available. And also, to display a slight more vivid, texture detailed graph, multiples of pixel is requisite, which means a slight improvement in frame quality requires a huge increase of performance. Therefore with the widely used multi-core central processors, which usually cause the imbalanced assignment and the waste of the resource, GPU has become the bottleneck of the performance rather than CPUs. （2） Another tendency is that, the integrated graphics play a much smaller role, while the more powerful discrete graphics occupied the GPU domain which is marked by the drastic competition between NVIDIA and AMD just after AMD merging the ATI, a traditional GPU company, after which AMD can develop its own GPU technology. Now even the newest and most powerful integrated graphic Intel HD5300 cannot compete with the discrete graphic that was produced several years ago. Since integrated graphics is welded on the computer main board, people have to control its size and power, which are two crucial factors to the performance, to

meet the space and cooling requirement. Modern computer engineers just let the integrated graphic deal with those basic graphic tasks and leave those professional and abundant graphic processing to discrete graphics. This specialized modularized hardware strategy is necessary in high performance required computers. A modern high-performance graphics workstation suitable for solid modelling must incorporate a number of features to provide high speed rendering of objects while at the same time remaining affordable[9].

If overview the development history of GPUs, we can find we can double the performance of GPU every year in the past several years by means of using larger video memory and more powerful processor with larger power. However, just the same as CPU, we cannot always apply for a larger power, because the heat, the power and the processor efficient can influence the enhancement of performance. (1) Overheating is always dangerous to processor because it can reduce the processor's life. So when the chip environment is too heat and the cooling system cannot deal with this heat, computer has to slow down the processor by decrease the clock frequency subjectively to reduce the production rate of heat. Obviously this can reduce the performance of GPU. (2) A powerful processor means a larger die, but it is hard to distribute the current to the whole large die with an approximate perfect pattern. During this process, it has to waste a lot of energy. A large die also means more chips, and that means a computer has to use more energy to charge them. To elimination the heat generated from processor, a computer has to have a powerful radiator. All of these led to an abundant power usage. (3) It is true that more chips always means more processing power, but we have to take the production and operation cost into consideration. Based on the three points above, even though people can frequently update our GPU to the newest generation to acquire a better performance, but once people take the power of GPU, the cost performance and the enhancement of graphic performance into consideration, people might only get an limited improvement. So the memory space of a single GPU device is not enough to contain very large networks now [8].One example is that, as Fig. 2 shows, one of the newest NVIDIA flagship desktop GUP GTX 980 may cast $600, which has nearly double the price of the last generation, but it only comes with a quite low improvement in performance. According to the Tencent website, GTX 980 brings only less than 25% score than GTX 970. Though in some other computer

components such as CPU, memory or hard disk, such an improvement is extremely huge, but in GPU field, just as what we said above, such improvement is quite slight that we can hardly figure out an obvious improvement in the frame quality.

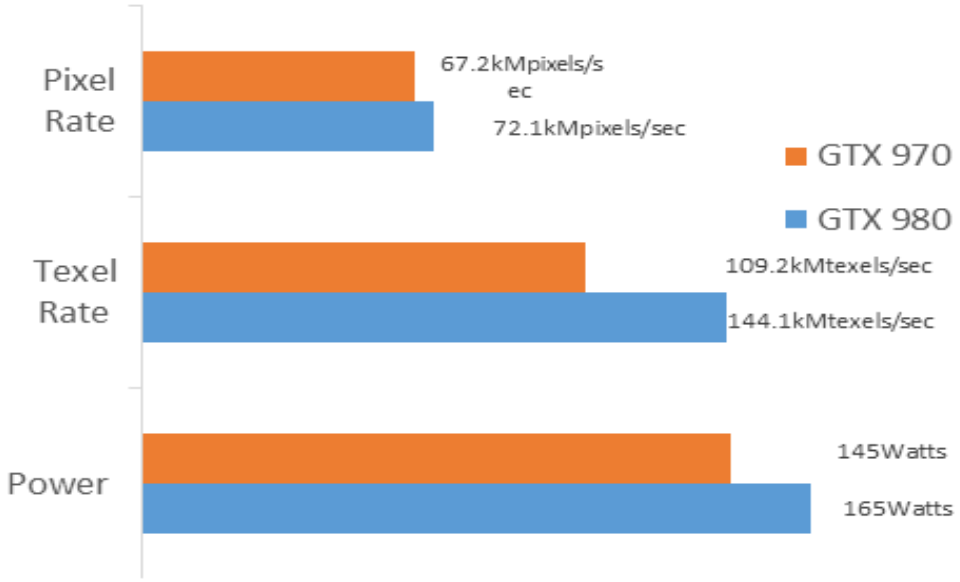


Fig. 2. Comparison between GeForce GTX 970 and GeForce GTX 980. The figure shows three items: pixel rate, Texel fill rate and the power. These data come from the professional GPU testing software 3DMark.

To break the bottleneck mentioned above, a new method to improve the performance of graphic display has been proposed. Instead of chasing the maximal performance of a single GPU, connecting several of it and (as Fig.3 shows) letting them work together to enhance the whole performance is a more efficient and wiser way. As the performance of both multicore CPUs and GPUs continues to scale at a Moore's law rate, it is becoming more common to use heterogeneous multi-core and multi-GPU architectures to attain the highest performance possible from a single compute node [7].

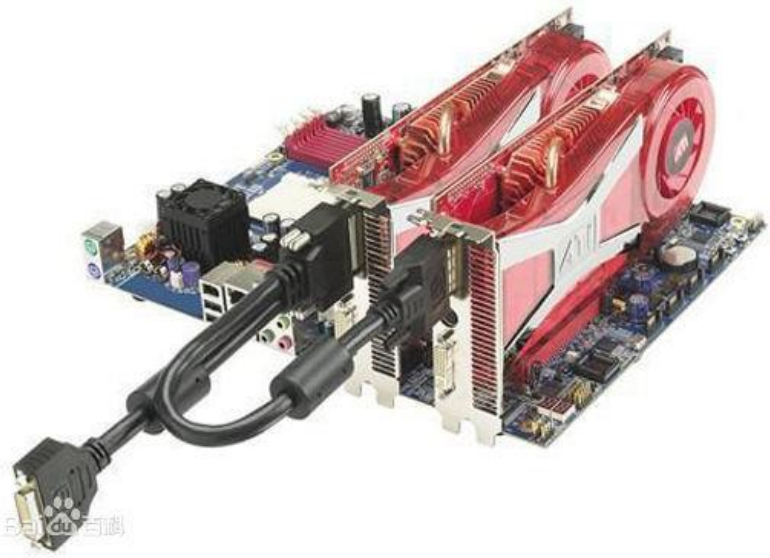


Fig. 3. This figure shows the structure of the multi-core GPUs system

Very similar as the multi-core CPUs system, this kind of structure can instantiated an order of cores and divide the input task into several parts and then distribute them to different cores.

The following are the key contributions of this paper:

1. The background information of single GPU.
2. The framework and connection of multi-GPUs.
3. The evaluation of multi-GPUs.
4. The conclusions.

II. BACKGROUND INFORMATION

A. Basic terms and performance parameter of GPU

All of the images you see on your monitor are consist with numeral small light spots called pixels. Now the most common resolution of monitor is 1920 multiply 1024 which means there are nearly 2 million pixels on the monitor. To display an image, GPU has to decide what to do with every pixel——judging the color of each pixels according to the binary data that comes from video memory or the system bus of CPU. So what should GPU do is to translate the binary data into pixel so that we can see pictures combined with these pixels. Actually GPU's job is much more complicated than CPU but its principles and components

are easy to understand, according to Jeff Tyson and Tracy V.Wilson [1]. In order to explore the principle of GPU, some related terms and background information are needed.

The terms and performance related parameters and parts are as follows:

1. Video memory

The first part we need to know is video memory (as Fig.4 shows). As mentioned above, GPU receive the data from CPU and then process these data. (1)During processing, the data should be putted in the video memory and then used by the processor of GPU. (2) And during the processing, some temporary calculate result should be stored. (3) In some advanced GPU, GPU will predict the frames that may be used in the next few seconds, so GPU should preprocess some frame and store them.

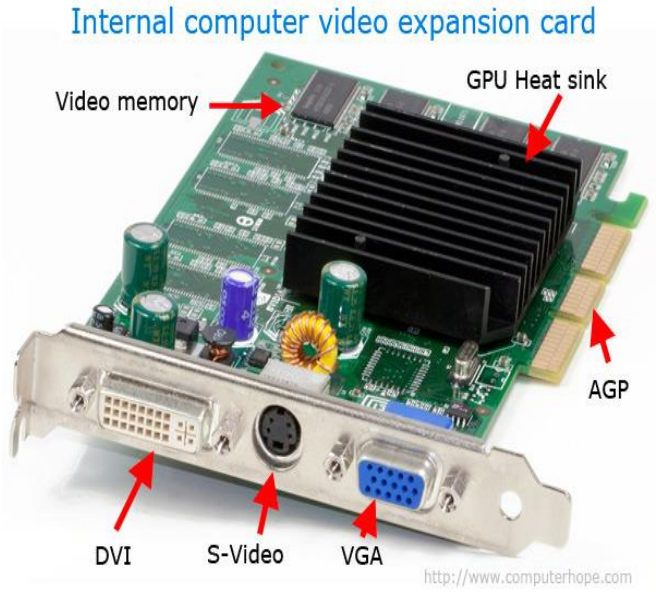


Fig. 4. This figure is used to show the video memory and the structure of GPU

If, in the next few seconds, what the graphs that has already been preprocessed is going to be used, GPU processor just them load from the video memory. If not, or the video memory is running out, GPU will empty these preprocessed frames, release memory and store new temporary calculation results or new preprocessed frames. The advantage of this strategy is that the GPU can constantly output stable, smooth graphs in a full load condition even though doing this may the decrease power efficiency. In these three processes, the storage GPU use is video memory. Theoretically, the GPU performance has a linear relationship between the

capacities of video memory within the section that video memory is the bottleneck of the whole output performance of GPU, but in the actual situation, many other factors can affect the performance of GPU, such as the speed of GPU chips, the reading and loading speed of video memory and the speed limitation of the interface between GPU and monitors.

2. Clock cycles and operation frequency

The second term is clock cycles or operation frequency. Both of them are used to describe the speed of GPU processor or video memory. They refer to the time used to accomplish one data processing, so a shorter clock cycles, or a larger operation frequency means in a given time, GPU can deal with more tasks, thus improve the performance. In general the clock cycles is represented in nanosecond and the operation frequency is represented in MHz. The most common clock cycles are 3.6ns, 3.3ns, and 2.8ns. For DDE SDRAM video memory, it is better to describe the operation frequency with a modified frequency——equivalent operation frequency, considering the character that this kind of memory can send and receive data during either the rising edge or falling edge.

3. Stream Processing Units

In the past few years, Stream Processing Units became popular in GPU fields especially for those high performance required GPUs. Stream processing units is a versatile new shade units using the stream processing strategy, a computer programming paradigm that allows some applications to exploit a limited form of parallel graphic processing more easily. It is a new technical indicators for the GPUs compared to the former Pixel Pipelines and Vertex Pipelines. Basically a GPU has several different shaders that can be used as small processors to process parts of the image. Typically we can imagine them as several different processor that only take in charge of their own task——a part of the image the GPU is processing.

SPU is a new GPU architecture and it combined the previous vertex shader and pixel shader together, which means a stream processor is a generic shader that can be turned into a specific shader on demand. It can either complete the VS (Vertex Shader, vertex shader device) operation to be completed PS (Pixel Shader, pixel shaders) operations and any VS/PS ratio can be composed according to need, thereby to play a broader

developer space. This is essential to graphic design software and 3D game developers since this function leave them a huge open field within which they can achieve more complicated special effects and generate more smooth and texture detailed pictures. Stream processing unit is composed of rendering pipeline part of the rendering pipeline, including a complete stream processors and texture mapping processors. Adopt a unified shader architecture can effectively and fully utilize graphics idle resources, no longer need to rely on the previous generation of graphics rendering pipeline execution command queuing and individually executed.

4. Bit wide

The third crucial term is bit wide. This parameters is crucial to both GPU and video memory [8]. We can image this as the width of a door, and the wider the door, the more people can go through in a certain time. So a large bit wide means the memory can exchange more data with chips or interface. Their computational formulas are as follows:

$$V_{memory\ capacity} = v_{particles\ capacity} \times n_{paritcles\ \ number} \tag{1}$$

$$B_{memory\ bit\ wide} = b_{particles\ bit\ wide} \times n_{particles\ number} \tag{2}$$

$$B_{memory\ bit\ wide} = b_{particles\ bit\ wide} \times n_{particles\ number} \tag{3}$$

After knowing the bit wide and speed (operation frequency), we can calculate the bandwidth.

$$B_{bandwidth} = F \times B \div 8 \tag{4}$$

The reason why we use 8 to divide here is that, 1 byte equals 8 bit. Bandwidth is the measure of video memory speed.

B. Graphics technique

1. Pixel fill rate

Pixel fill rate is the number of pixels the GPU can render to the screen in a certain time. In this case, fill rates are given in megapixels per second or in gig pixels per second (in the case of newer cards), and they are obtained by multiplying the number of raster operations (ROPs) by the clock frequency of the graphics

processor unit (GPU) of a video card [2]. Before the pixel shader processing became the more limiting factor, this was the most accurate measure of performance (along with Pixel fill rate). The computational formula of Pixel Fill Rate is as follows:

$$R = (\#\ \text{of ROPs}) \times (\text{Core Clock})$$

In this formula, ROPs refers to Raster Operator. It is the last stage of the graphics pipeline which writes the textured/shaded pixels to the frame buffer. Raster Operators (ROPs) handle several chores near the end of the of the pixel pipeline. ROPs handle anti-aliasing, Z and color compression, and the actual writing of the pixel to the output buffer.

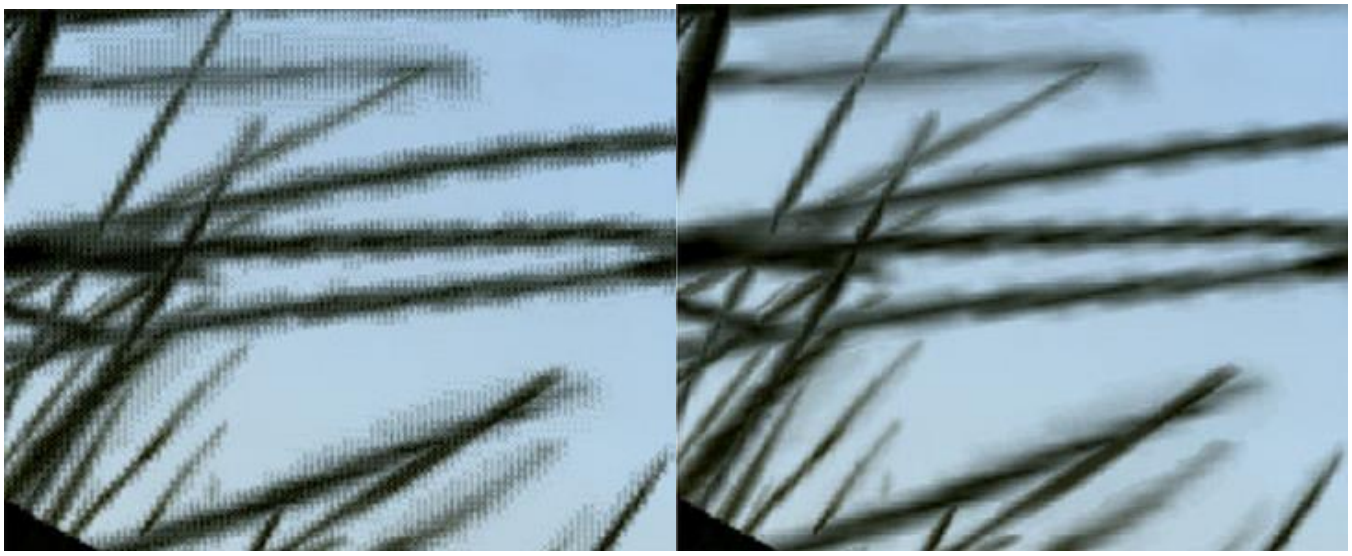


Fig. 5. This figure shows the effect of pixel fill rate

The Core Clock is the clock speed of the GPU. Each GPU has two clock speeds of interest, the memory clock, and the core clock. The core clock is the speed at which the graphics processor——the processor of GPU, on the GPU main board operators. The clock speed of a chip, combined with the number or configuration of the pipelines within the chip, give a pretty accurate picture of what the performance of the chip will be.

In many cases, the difference between two GPUs that belong to the same generation from the same company will differ only in core and memory clock speeds. They have the same framework, the same physical structure and other related parameters [8]. The ability to set these clock speeds to whatever they

want gives the manufacturer the ability to market multiple GPU from the same chip.

As Fig. 5 shows, the two pictures show the same object but the first one seems blurry. This is because the GPU that process the first picture is not powerful enough to calculate all of the pixel in a certain time. The computation speed is quite low, so before generating the next frame, GPU could not complete the task of this frame.

One thing strange is that maybe you'll notice that retail card clock speeds often differ from the reference card specification (especially with NVIDIA GPUs) [9] [12]. This is because retail card manufacturers are given some room to adjust clock speeds in order to differentiate their card from every other card on the market based on the same reference design. Though it is much likely to occupy the market instead of provide numeral kinds of products that are suitable for every users, said NVIDIA, we can use this character to improve the performance of GPU ourselves by manually changing the core clock speed on most GPUs using third party utilities, or, in some cases, the driver provided by the manufacturer. Most cards don't gain much performance from over clocking, because a slight over clocking is not enough to speed up the processor to generate a distinguishable performance enhancement, but sometimes GPUs are released with a tremendous amount of over-clocking headroom and serious performance gains can be had simply by cranking up a dial in your driver control panel [4].

However, typically faster clocked cards will use more power, and thus produce more heat, so over clocking or faster clock always bring a huge challenge to the cooling system. At the same time, it can also cut down the processor life.

Also, note that some advanced GPU can dynamically adjust their core clock to different speeds according to the amount of the task, and that most chips run at a slower clock frequency when in 2D mode (regular programs in windows, not 3D games).

2. Z-Buffering

In computer graphics and GPU field, z-buffering, also known as depth buffering, is the management of image depth coordinates in three-dimensional (3-D) graphics, usually done in hardware, sometimes in

software [3]. Z-buffering is a kind of tool that help GPU resolve the problem of visibility. In the process of generate a 3D picture, it is not wise to calculate and display each frame independently [10]. To promote the efficiency and the whole output performance, advanced GPUs usually use another strategy. First they built the whole 3D structure with modeling algorithm [7]. After this step, a virtual structure is saved into the video memory. Then to display the picture from a certain angle, the GPUs have to simulate like our eyes to look in an angle of view and get a perspective picture.

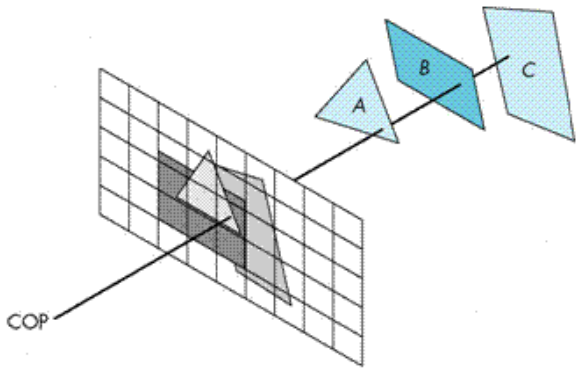


Fig. 6. This figure is used to show the function of Z-Buffering

In Fig. 6, all of the key structure points which decide the shape of the virtual structure can been seen from the perspective picture and their position relationship in the 2D plate is located perfectly according to the angle of the view. Then the Z-Buffering start to work.

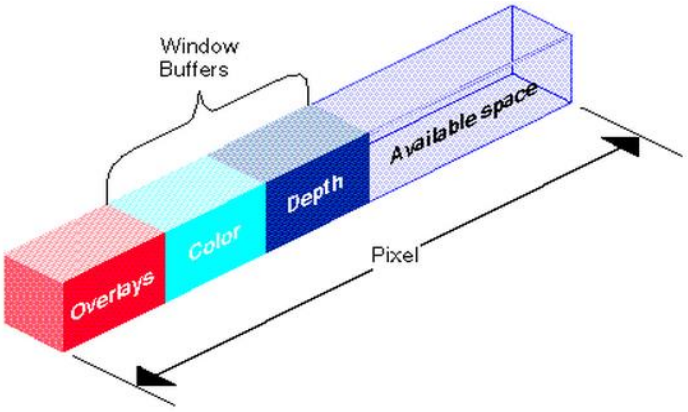


Fig. 7. This figure is used to show the process of record the depth

As Fig.7 shows, Z-Buffering can record the depth of every pixel and decide which one should be colored

and which one should not. Only those pixel that have the smallest depth in that pixel position will be colored and thus become visible [9]. By doing this, GPU can transfer the perspective picture, which only have the key position points, into a picture that only those structure should be seen and do not covered by other structures can be displayed in the final output graph as Fig.8 shows. In a word, Z-Buffering is a storage that can help to decide which pixels are visible, and which are hidden.

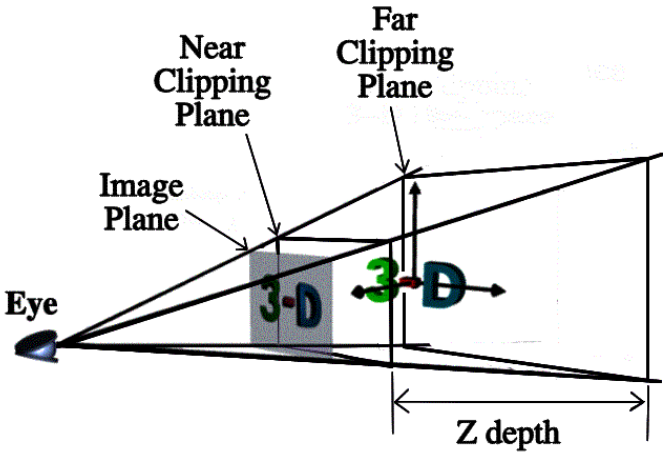


Fig. 8. This figure is used to show the function of Z-Buffering

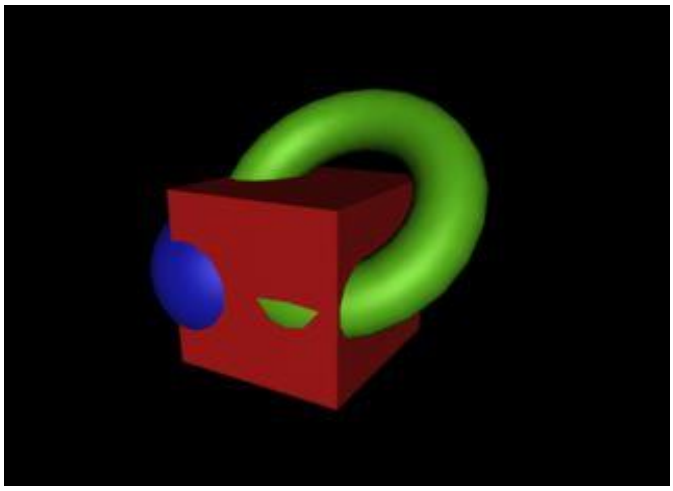


Fig. 9. This figure is used to show the effect of Z-Buffering

After knowing the basic function of Z-buffering, let us see some complicated cases. In Fig.9 we can see three 3D objects. Before they are displayed, GPU will generate the perspective structure of them three independently [13]. Then according to the angle we want to look, Z-Buffering will record the depth of every overlapped points. Only those do not overlap with others or overlap but stands near to the screen can be seen

on the final frame. Finally, we see this picture.

Apart from Z-Buffering, another method, the coloring algorithm is another common solution which, though less efficient, can also handle non-opaque scene elements [5].

During the process of generate a single frame, GPU send every key points (which we may call it object) into the processor to calculate their depth. When an object is rendered, the depth of a generated pixel (z coordinate) is stored in a buffer (the z-buffer or depth buffer) [6]. To help remember the 2D location of the pixel well and use the storage more efficiently, this buffer is usually arranged as a two-dimensional array (x-y) with one element for each screen pixel. If another object of the scene is rendered in the same pixel, the method can calculate the new depth and compare it with the former depth in this pixel and decide whether overrides the current buffer if the object is closer to the observer. The chosen depth is then saved to the z-buffer, replacing the old one [15]. When a new object is rendered, this process will be operated repeatedly. The z-buffer will allow the program to correctly reproduce the usual depth perception: a closer object shelters a deeper one. This process is called z-culling. And after rendering all of the objects in this pixel, only the object which the depth of it is saved in this location's buffer can be colored and seen in the final graph.

Considering the precision of the depth, the granularity of a z-buffer has a great influence on the scene quality. For example, a z-buffer that has a capacity of 16 bits can result in artifacts (called "z-fighting") when two objects are very close to each other. Due to the limitation of the buffer capacity, two closed enough may have the same depth, thus GPU processor cannot judge which one is closer and which one is further. For this reason a 24-bit or 32-bit z-buffer behaves much better, although the problem cannot be entirely eliminated without additional algorithms (actually we can never resolute this kind of problems with digital information instead of analog information because the digitization must bring rounding off). And for the same reason an 8-bit z-buffer is almost never used since it has too little precision [4] [11].

Similar as z-buffering, a variation on z-buffering which results in more evenly distributed precision is called w-buffering. In another word, w-buffering is a z-buffering that can distinguish more close depth. When distinguishing to close depth, it is used, if necessary, accompany with z-buffering in the way that z-buffering

record the high-order of the depth and w-buffering record the low-order. That means in such case we use w-buffering to enlarge the bit of z-buffering so that it can distinguish more precise numbers. In general, this is desirable, but sometimes it will cause artifacts to appear as objects become more distant.

From the last decade, the Z-buffer technology has become widely used in all kinds of computers that require a graphic processing function. And in the high performance required field such as 3D design and huge 3D games, it is the most common and basic part in the GPUs. The Z-buffer is implemented as hardware in the silicon ICs (integrated circuits) within these computers. The Z-buffer is also used (implemented as software as opposed to hardware) for producing computer-generated special effects for films [3].

Now with the tendency that people chasing a more real and verisimilar frames, not only the shape or the texture of the picture is necessary, but also the feel of tridimensional and the shadow of objects. So another function that z-buffering can provide is the generation of shadows. As we know z-buffer can record the depth information of the key points of the 3D structure, so we can calculate the shadow position of a certain point according to the view angle and a virtual light source. This technique that generate the shadow for 3D objects is called shadow mapping. According to Wikipedia, this concept was introduced by Lance Willianms in 1978, in a paper entitled "Casting curved shadows on curved surfaces". Since then, it has been used both in pre-rendered scenes and real time scenes in many console and PC games [].

At the start of a new scene, the z-buffer must be cleared to a defined value, usually 1.0, because this value is the upper limit (on a scale of 0 to 1) of depth, meaning that no object is present at this point through the viewing frustum [2,3].

On recent PC GPUs (1999–2005), z-buffering management uses a significant chunk of the available memory bandwidth since it is corresponding to the different pixel and nearly every pixel need a memory space to storage the depth information. Various methods have been proposed to reduce the performance cost of z-buffering and save memory space. One strategy is lossless compression (computer resources to compress/decompress are cheaper than bandwidth) and ultra-fast hardware z-clear that makes obsolete the "one frame positive, one frame negative" trick (skipping inter-frame clear altogether using signed numbers

to cleverly check depths) [3].

According to the mood, z-culling is the earliest pixel elimination method that is based on depth, which can provides an increased performance when the cost of rendering of hidden surfaces increases and also a direct result of z- buffering. The depth of each pixel candidate is compared to the existing geometry where it might be hidden.

When a z-buffering is used, a pixel can be removed as soon as the depth is known. In this way, it is possible to get rid of the process of lighting and texturing a potentially invisible pixel. And time-consuming pixel shaders will not be operated for the removed pixels, which will make -culling a good optimization one when the fill rate, lighting, texturing or pixel shaders become the main bottlenecks [12].

3. Anti-Aliasing

In the GPU field, anti-aliasing is a technique that can eliminate the jagged edge of frames on the output image of monitors. It is also called edge softness. See as Fig.10.

Fig. 10. This figure is used to illustrate what is aliasing

The rugged aliasing is generated in the condition that a high resolution is displayed on a low resolution monitor or the GPU cannot calculate the precise location of the key points coordinate. Since processor could not calculate the precise coordinate of a point, a small area is used to represent the point. When another point that is close enough to the former is also represented by a small area, these two areas may

overlap. If a few overlap occur in a small zone, then it seems as the jagged edge of a boundary. The anti-aliasing technology can fix these kind of problems property and it is usually used in the digital signal processing, digital photography, and digital sound and computer graphics fields [13].

The resource signal that comes from a high resolution consist more pixels. An analog signal contains infinitely-great information. During the sampling procession, huge amount of information are represented by little information, which in the GPU field means more pixels transfer to few pixels, so some data or information should have to be ignored, leading to the damage and lose of the original signal. In the 3D graph field, every frame is combined with so many pixels. The pixels of the screen is limited so when we want to show the location of polygon, we cannot use the absolute coordinate. The only method we can use is relative coordinates. Since we do not have enough samples to stand for everything in the virtual 3D world, in the output frame, something like wave, circle, spot, zigzag or flicker may be found in the overlap region. Of course this can seriously affect the frame quality.

Anti-aliasing technique can be divided into following six categories.

1) SSAA

SSAA means super-sampling anti-aliasing. This is the earliest anti-aliasing technique. It may take up much system resources but the principle is simple and the effect is direct. First, it map frame into memory and zoom in the frame. Then use the super-sampling technique to sample the zoomed in frames. During this process, generally GPU processor will select two or four adjacent pixels and combine them together to get a final pixels that have the characters of adjacent pixels. This final pixel is the one that we can see on the monitor. Before combination, GPU processor will record every character of the former pixels and generated a similar color to get the transitional effect. By doing this, the boundary can seems smoother with a related color. After this, GPU processor will load the frame and zoom it out into the right size and then save the frame into video memory again to replace the original frame. This frame is ready to display. In a word this technique can zoom in a frame and optimize it and zoom out and then output the frame.

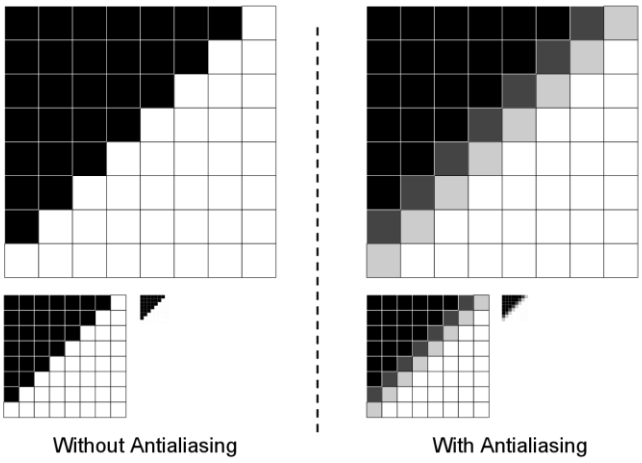


Fig. 11. This figure is used to illustrate the process of anti-aliasing

The times we zoom in is a crucial index of this technique and is used to represent the effect of anti-aliasing. Generally, newest GPU can achieve ×8, ×16, or ×32 anti-aliasing. The effect of anti-aliasing is shown as Fig. 12.

The two sampling method are as follows:

Ordered Grid Super-Sampling, which is OGSS, this select two adjacent pixels.

Rotated Grid Super-Sampling, which is RGSS, this select four adjacent pixels.

Fig. 12. This figure is used to illustrate the process of anti-aliasing

2) MSAA

MSAA means multi sampling anti-aliasing. It is a special kind of super sampling anti-aliasing. It comes from OpenGL and it only process the data that stored in Z-Buffer and Stencil Buffer with the method of SSAA. This can be as simple as only process the pixels that located in the boundary of polygons. Compared with SSAA that process all of the data, MSAA can save much more system resource and the effect of MSAA is only a little inferior than SSAA.

3) CSAA

CSAA means Coverage Sampling anti-aliasing. This technique was introduced by NVIDIA with its first practical AA technology. And also it is a private technology special for NVIDIA. Based on MSAA, it can save more energy and system resource. When processing the coordinate of the points that located in the polygon boundary, GPU processor will overlap the sub-pixel's coordinate that need to be sampled and set the pixel coordinate to the location that is calculated by the driver and GPU processor. Compared to stander sampling MSAA, CSAA can operate boundary sampling in the highest efficiency and at the same time improve the whole performance. For the same GPU, sampling performance of 16×CSAA is only slightly inferior to 4×MSAA, but the effect is just the same as 8×CSAA. 8×MSAA has the same performance with 4×MSAA, but it take up the same power as 2×MSAA.

4) CFAA

CFAA means custom filter anti-aliasing. This technology came from the R600 of AMD. CFAA is a special MSAA that have a larger sampling area. Different from the MSAA that only sample pixels that near the polygon boundary, CFAA can select the pixels that have a significant influence to the anti-aliasing effect smartly by the control of driver and then zoom in this area to process. This technology can get a better effect with a slight sacrifice of performance. It also takes up fewer system resources.

5) FXAA

FXAA means fast approximate anti-aliasing. It is based on the traditional MSAA but is better than MSAA. Essentially, it is a kind of single pixel shader. Similar with MLAA, it aims at the latter stages to

render frames. MLAA means morphological antialiasing, which is AMD's special technique for 3D games. MLAA contains three steps. First, find the discontinuous pixels within a certain area. Then decide which render mood should be used. And at last with the selected mood, mix the pixel colors near boundary. The biggest advantage of MLAA is that it can release the pressure of system when render several object at the same time. Do not like MLAA that use direct compute, FXAA is only a later stage single pixel shader, and it doesn't rely on any GPU computation. So FXAA is suitable for any GPU and for both NVIDIA and AMD. FXAA and MSAA is the most common technique in GPU field.

6) TXAA

TXAA means temporal anti-aliasing. According to NVIDIA website, it can creates a smoother, clearer image than any other anti-aliasing solution by combining high-quality MSAA multi sample anti-aliasing, post processes, and NVIDIA-designed temporal filters. TXAA is a new film–style anti–aliasing technique designed specifically to reduce temporal aliasing (crawling and flickering seen in motion when playing games). This technology is a mix of a temporal filter, hardware anti–aliasing, and custom CG film–style anti–aliasing resolves. To filter any given pixel on the screen, TXAA uses a contribution of samples, both inside and outside of the pixel, in conjunction with samples from prior frames, to offer the highest quality filtering possible. TXAA has improved spatial filtering over standard 2xMSAA and 4xMSAA. For example, on fences or foliage and in motion, TXAA starts to approach and sometimes exceeds the quality of other high–end professional anti–aliasing algorithms.

TXAA combines the raw power of MSAA with sophisticated resolve filters similar to those employed in CG films to produce a smoother image far in advance of any comparable technique. Depending on the type of shading implemented in a given game, the performance impact of TXAA can be slightly different. In contrast to methods like FXAA that attempt to maximize performance at the expense of quality, TXAA attempts to maximize quality at the expense of performance. TXAA is a superior option for those looking for the highest–quality anti–aliasing with the most efficient performance possible to make games such as Assassin's Creed IV Black Flag and Batman Arkham Origins look their absolute best.

Fig. 13. This figure is used to illustrate the comparison of TXAA

C. Principle of GPU

To “translate” binary data from CPU into pictures, five steps must be involved.

1. CPU send binary data to GPU through system bus.
2. Binary data is sent to video memory and wait there to be processed.
3. Load data from video memory and send them into Digital Analog Converter to transfer 0 and 1 into

images.

4. Send the images to monitor

The GPU only take in charge the last two steps. During the two steps, the three techniques mentioned above will be used.

III.MULTI-CORE GPUS

A. Connection of multi-core GPUs system

Too break the bottleneck of the single GPU performance and to get maximum efficiency, both NVIDIA and AMD launched multi-core GPU technologies based on their own parallel method. Multi-core, as this name conveys, is the technology that connect two GPUs together and let they cooperate to process one frame, getting a higher performance. It can cut down the task of one single and promote the quality of the output frame. AMD calls it Crossfire, and NVIDIA calls it Scalable Link Interface (SLI). In essence, both companies are using a similar concept to distribute the workload presented by your favorite game across two, three, or four graphics cards.

NVIDIA's revolutionary combination of multiple professional GPUs can increase the visualization and compute, gendering ability of computer system. Whether you design cars, render scenes with millions of polygons, or work with large models for seismic interpretation, or play large 3D games, a multi-cores GPU system will significantly increase the performance you have available. Not only the performance of the output graphs quality, this system can also speed up the system and reduce these processing and computing times into a fraction of the number of secondary GPUs, the number how many would be accomplished with CPUs or even single GPUs [14].

What a multi-core GPUs system can provides is more than just performance gains. It also gives you the freedom to run your applications with full features and effects enabled under the max monitor resolutions. Nowadays, even the newest and the most expensive single GPU cannot afford the highest effects such as the 8 times anti—aliasing or the high quality texture. This freedom includes real-time ray tracing and global

illumination for full realism of your models and scenes. Now let us use NVIDIA GPU and the SLI technology as an example to explore how to achieve the connection of the multi-core GPUs system. NVIDIA and AMD have different connection operations but the reason why we choose NVIDIA is that the SLI technology is much more complicated and strict. So long as we know how to set up SLI system, the Crossfire is just a piece of cake.

To connect GPUs together, the following steps are needed:

1) Driver installation

The installation of multiple video cards is a piece of cake under Windows Vista and Windows 7, even if the cards or GPUs are from different manufacturers. This means they have the same what we call framework. GPU cards from AMD and NVIDIA cooperate splendidly in multi-screen scenarios. If you want to increase 3D performance, however, Crossfire and SLI modes always require at least two graphics cards from the same chip manufacturer, both the GPU processor and the bios chip. Not only the chip but also the principle of parallel mechanism that make the increase of 3D performance by mixing AMD and NVIDIA cards together is not possible.

In order to divide the frame task and distribute 3D rendering to two different cards in SLI mode, two identical GPUs and similar frame buffers is necessarily needed. Different GPUs do not work, nor do cards based on the same GPU but with different parameters such as the memory capacities, the core clock or the bandwidth, although the current drivers don't actually provide any error messages if cards with different chips are connected via the SLI Bridge——a cable that connect two GPUs special for SLI.

Even though the bridge is silently accepted, physically connected and power charged, there is no performance increase, as the driver cannot activate SLI mode, leading to just one GPU works and another in salience. The SLI driver software’s interface will only show the information whether the installed graphics cards are able to cooperate.

For example, when we are using two or more NVIDIA GPUs, no matter whether the GPUs are matched with each other, one GPU can be dedicated to physics acceleration (PhysX). This connecting can be done

when two GPUs are connected and working together in an SLI mood and another third GPU is added. According to a test that was done by **, two identity GeForce GTX 280 GPUs and a different one, GeForce GTX 275 are connected with a SLI-mode PCB bridge cable that has three expanding interface []. The driver can recognizes the pair of GeForce GTX 280 GPUs with an SLI configuration, and offers to reserve one of the GeForce GTX 280 cards or the additional GTX 275 for physics calculations. In this testing, the drivers have no problem handling the three SLI-bridged graphics cards. They three were truly connected physically and none error information was reported. But actually due to the GPU generation version and parameter difference, the mismatching third card is simply ignored for SLI bonding and of course if the systems is working, the third GPU was useless and cannot contribute to the performance.

For SLI mood, the driver software can be easily found from NVIDIA official website. Just follow the instruction and install. After the installation, all necessary drivers programs are installed automatically and assigned to the respective GPUs. According to the official website instruction, if the installation is fresh, the driver announces an SLI configuration, and all you have to do is to activate it by checking a box []. If everything goes well and other error information reported, the configuration then works and the GPU processor of the main GPU will divide the 3D frame rendering workload as soon as a huge 3D game is started with an SLI-profile assigned to it in NVIDIA's driver.

2) Connectors

The connection of the Crossfire and SLI technique is quite different according to their principle and mechanism. Each technique has a max number of GPUs that can connected and work together. This max number is determined by both the limitation of the interface and the parallel capacity ability of GPUs. The Crossfire and SLI solutions are pretty much identical in this aspect.

As we know when using a SLI multi-core mood, we should connect the GPUs via both the PCI-E interface on the main board and the SLI bridge cable.

Actually this system can still works if the bridge cable is removed. While this will bring some additional

stress to the PCI-E bus. Two reasons can be used to account this. First is that the bandwidth of PCI-E bus and the PCI-E interface may not enough to transfer the inter-GPUs data stream and the CPU to GPU data. The second is that bus system use an interrupt mode, which may undermine the efficiency of the data transmit. So bridge cable is necessary. If the GPU have one additional secondary GPU, two cards can be connected. If have two or three or at most seven GPUs, they can be connected with a multi-way bridge cable that has many interface in one cable.

Owing to the different structure, the advantage and necessity of connection GPUs with bridges cables should be considered separately for the NVIDIA and AMD. According to &&&, you could easily leave the bridge connector off when using mainstream graphics cards, since the PCI Express bus is quick enough for the data traveling over it. This is only an option for certain AMD-based configurations, from a pair of Radeon HD 4650s down to two Radeon HD 4350s [6]. Whereas, dipping down into the Radeon HD 3000-series or up to Radeon HD 4670s, a bridge cable is also necessary.

If GPU has two bridge connections using the Crossfire mood, it doesn't matter if you use the left or the right one since this technique do not distinguish the main and secondary GPUs. Nor are there any advantages to using multiple bridge connectors between GPUs just because you can. Only the PCI-E interface and PCI-E bus is enough to synchronize the inter-GPU data stream and make the driver recognize the Crossfire configuration and achieve the system.

3) Mixed-GPU setup

If you want to run an SLI configuration, you have to use GPU with the same GPU model that have the same generation, version, identity main parameters such as the bandwidth, core and memory frequency. Different graphics card manufacturers aren't a problem, but one thing is crucial that they must have the same bios chips. The chips should be produced by the same manufacture and have the same version. And also the clock rates may be an important factor. The first main GPU (the one closest to the CPU or just selected by the driver software arbitrarily) is usually the "master" in the configuration. If this card is overclocked, then there's a risk that the secondary GPU will try to match the same speeds, too. So if you suffer from frequent

crashes or the overheating, you should try to adjust the clock rates of the master card to match the second card (or even slower) and see if this can make the configuration more stable [6]. Secondary GPUs always fellow the main, and this means you can only adjust the system by adjusting the main GPU. Plus, both GPUs should have the same frame buffer, too.

AMD mood allows two different GPUs in the multi-core system, but they have to stay within the same generation, as indicated in this chart since staying within the same generation means they have the same main parameters. According to a test, they tried pairing a Radeon HD 5800 card with a 5700 board, and of course it didn't work. Pairing a Radeon HD 5870 with an HD 5850 or an HD 5830 works fine, though [6]. Whether this test make sense is debatable, given that a Crossfire configuration is at best only twice as fast as its weakest card. But this can be a good example to explain the flexibility of the mixing mood. This mood allow you buy a cheap at first, and, once prices fall or the current performance is not good enough for you, just upgrade the computer with adding only one better GPU and you do not need to but two identity GPUs.

The different combinations worked splendidly in this tests. With AMD GPUs, the driver software usually select the fastest one as the main GPU. The performance increase by doing this is very small, but it can be seen in all the benchmarks. Again, running a differently-clocked Crossfire combination might cause some issues due to higher clock rates of the master card [6]. The main GPU can process in a high speed but the secondary one could not finished its task within the desirable time. We have to change the core clock rates and change the card sequence to let them match each other. Again let us use the former test as an example. In that test we have there GPUs, they are the Radeon HD 5830, HD 5850, and HD 5870. The Radeon HD 5870 was selected as the master card automatically. Only when we used it as the third card in the configuration did it work properly since we have to slow down its frequency to match the other two lower performance GPUs if it is the main GPU. This adjusting strategy doesn't really have an impact on performance or the complement of the system, since all three cards will slow down to match the speed of the Radeon HD 5830. However it is a matter of performance waste.

4) CPU Scaling on GPU

When using NVIDIA GPUs, the results are slightly different. For the most part, the mainstream high 1960×1024 or higher resolutions and demanding anti-aliasing put most of the load on the graphics cards. Some differences can be seen when using Battlefield: Bad Company 2, where increasing the CPU performance stops having any effect beyond 3.80 GHz. The older Source engine is a different matter. While AMD performance scales with increased CPU speed, NVIDIA only show small increases [6].

B. Introduction of SLI

1) Introduction

The full name of SLI is Scalable Link Interface. It is a technique that connect two pieces of PCI Express GPUs together and improve the whole performance of the system by the cooperation of two GPUs.

The operation principle is much complicated than the name. According to the official report, at the time that PCI became the most common and popular socket, the famous 3D graphic chips company 3DFX invented a SLI technology that support the 3D GPU called Voodoo 2 which came from its sub company. Its principle and mechanism is quite the same as the SLI and Crossfire we use today. But at that time, this technique can only be used in some special field during to the high cost. During the next few years, with the development of related techniques and the reduction of the manufacturing and research and development costs, AGP socket was updated to PCI-E, which eliminated one of the limitation of the SLI technology (but at that time the new socket structure was not common), and more powerful mainboard, GPU and processor became much common than before. It is exactly under this background that NVIDIA decide to explore the huge SLI market once more time. So 3DFX called together the former engineers from 3DFX Company and invent a new SLI technique based on the old 3D acceleration of the Voodoo 2. Owing to the support of other hardware, this new SLI can connect more powerful GPUs and can expand the 3D effect to an extremely level. But at that time, this new acceleration technique did not attract the attention. Later after NVIDIA merging 3DFX, NVIDIA did not continue invest on this technique since at that time the AGP socket detained the market since only the new PCI socket can support the new SLI), leaving SLI a history for a long period. But history didn’t buried SLI from then. Later with the popularity of PCI-E, it was time for NVIDIA and its SLI to

domain the market

2) Principle

SLI is NVIDIA's multi-core GPUs technology. Insert two GPU that carry NVIDIA GPU processor to the socket of PCI-EX16 main board, and then use a SLI cable to connect them together. This cable can transport data between the two GPUs, and the frame data that has already been processed separately will be collected with this cable. After collecting the frames, they will be integrated and output as an entire graph that will be display on monitors. To make sure the division of task and cooperation with each other, NVIDIA integrate the SLI control unit inside the GPU chips. This chip control the connection, task division and cooperation of two GPUs. When a frame is rendered, the chip divide it into several parts. One of the division mechanism is dividing the frame into two parts——upper and lower parts， and let them to be processed by two GPUs. After rendering, the chip will combine the two frame together and output it to the monitor. During the combination process, another algorithm will be used to make the process precise.

Even though SLI require two identical GPUs work together, they are not even during the process. One is chosen to be the main GPU and another is secondary. The secondary GPU only receive task from main GPU and send the result to it. Both of them are connected with main board with PCI-E interface. And actually the cable used to connect the two GPU is a PCB card. That means, the row data is send from the CPU via PCI-E of the main board to the main GPU. And then the chip of main GPU divide the task and send a part of the task to the secondary GPU via the cable——PCB card.

As we know for a single GPU system, GPU is inserted to the main board, and main board send the image or rendered task to GPU. The GPU complete the task independently. But now in multi-core GPUs, SLI connect two processor together and obviously the advantage is that the connected system can break the traditional single ship method and enhance performance by the parallel mechanism.

According to the official report, SLI can support at most 8 GPUs. Although it is useless in the PC field, since none application will require such a high requirement for GPUs and the cost of such a system is difficult to afford. But in the workstations, a field that hardware cost is much less important than

performance and working efficiency, it has a significant meaning. 8 GPUs working together means an extremely high performance in the rendered process.

C. Introduction of Crossfire

1) Introduction

Crossfire technique, or the CF， is the multi-core GPU technology comes from ATI. The reason why we do not distinguish ATI and AMD is that, though AMD merged ATI, but actually it is ATI that take in charge of the GPU and main board business. Quite the same as SLI, it also allow several GPUs work together to generate a high performance output. This technology was first introduced to the world on June 1st, 2005, just one year later than SLI. From that time, a huge revise was introduced to that technique.

According to their official website, Crossfire support both Intel and AMD chipset, the ATI Radeon Xpress Chipset, general Radeon graphic processor and Radeon Crossfire graphic card.

Compare to SLI, the biggest advantage of Crossfire is its compatibility. The special designed ATI multi graphic processor mode allow crossfire be suitable for almost every application. And for the application, few refinement or optimizing is need to implement the crossfire technique.

2) Principle

Crossfire can operate in the following four mode:

a. Alternate Frame Rendering

Divide frames to different GPU processors. For example VGA 1 process the（1,3,5,7,9）frame，and VGA 2 process（2,4,6,8,10）frame. In the testing mood, a mood that the requirement of performance is really high that each index of GPU is the bottleneck , since one processor take in charge of an entire frame, the time each framed used is not shorten, so this method cannot improve the quality of graphs, but it can make graphs more smooth and increase the FPS.

b. Scissor (Split Frame Rendering)

Divide the frame into two parts for example the upper and lower part, or the left and right part. Then let each GPU processor process each part separately. After finishing the gendering the two pictures, the main GPU will combine them together and display the ultimate graph. In this mood, the time used to render one graph is double, so the quality of the graph will be promoted. But in the real working environment, this mood may bring a problem that, the task amount of each part may be strongly uneven and one GPU is in full-load situation but another may be low load, leading to the waste of performance.

c. Super Tiling

To deal with the issue above, developers invent another mood for crossfire. The Scissor mod divide a frame into two parts. But if divide frames into so many parts and distribute them to two GPU processors, statistically two GPU processor will receive almost equal task, thus avoiding the waste of performance and power. This mood can output the best performance but not all framework such as OpenGL support this.

d. Crossfire Super AA

This mood can increase the quality of pictures. Let two graphics cores running the AA operation, then combine the results. For example, both the two cores run the 4 times AA operation, and in the end an 8 times AA frame will be generated. The increase of this mood is limited since even if a single core can work in this way and it cannot improve the smooth or the FPS.

D. Comparison between SLI and Crossfire

The difference of these two mood can be summarized into the following aspects.

1) Connection type

Crossfire use a dedicated main GPU method. To achieve the crossfire, a main GPU with a general common GPU connected to it with a special cable is need. SLI use a completely symmetrical parallel mechanism. Several same GPU should be inserted to a special designed main board. The signal transmission relay on the integrated circuit on the main board. Relatively, crossfire is more convenient since users can add any kind of GPUs depending on their demand and for the hardware only the main GPU is special designed. However for the SLI, users have to use a special main board that has several GPU interface and have to use several same

GPUs. But the advantages of SLI that it can bring stable and high compatibility performance. To connect them, main board support is needed. Only those main board that have more than 2 GPU interface can support multi-cores. And for Crossfire, the AIT Radeon Xpress 200 main board, which is produced by the former GPU company ATI, is strongly recommended. For SLI, nForce4 SLI, a main board special for NVIDIA GPU is recommended.

2) Selection of GPUs

For Crossfire technique, it does not require two identical GPUs in the brand, physical shape, or basic parameters such as core clock, frequency, video memory capacity. All the users should do is selecting one as main GPU and connect others with it. However for the SLI technique, two identical GPUs are need, even the bios chip and the alterable core frequency.

3) Compatibility

Crossfire technique support more huge 3D games including some old ones. This character is due to the flexibility of the mechanism. Not too much improvement or optimizing is need to adapt this technique. Compared to this, SLI support few games and only the newest, special optimized for it can adapt it. This character comes from the highly parallel symmetric structure, which makes the optimizing much more difficult.

Now the ATI Radon Xpress 200 chipset is compatible with both Intel and AMD platform, so users from Intel or AMD can use Crossfire technique. SLI can also support the two platform.

4) Rendering mode

Crossfire has three rendering modes whereas SLI has only two. Crossfire has a special mode called “tiles coloring technology”. It is high efficient but with low compatibility.

5) Image quality

Crossfire has a “Super AA” mode that support 14 times anti-aliasing. This technique allows GPU output the highest level image quality under the mode of multi-cores. SLI doesn’t have similar techniques.

IV. TESTING ON MULTI-CORE GPUS

This is a test I did four months ago. In the test, we choose four GTX 980 GPUs. They came from four different companies: InnoVISION, GALAXY Technology, GIGABYTE and Colorful. The testing result of GTX 980 is shown as Fig.14. Just as we mentioned before, NVIDIA's SLI only distinguish GPUs that are identity but different manufactures. The only difference of the four GPUs is their package. In this section, we will first introduce the theoretic performance of SLI. And then we will give the application testing results, which include the graphic performance and power performance.

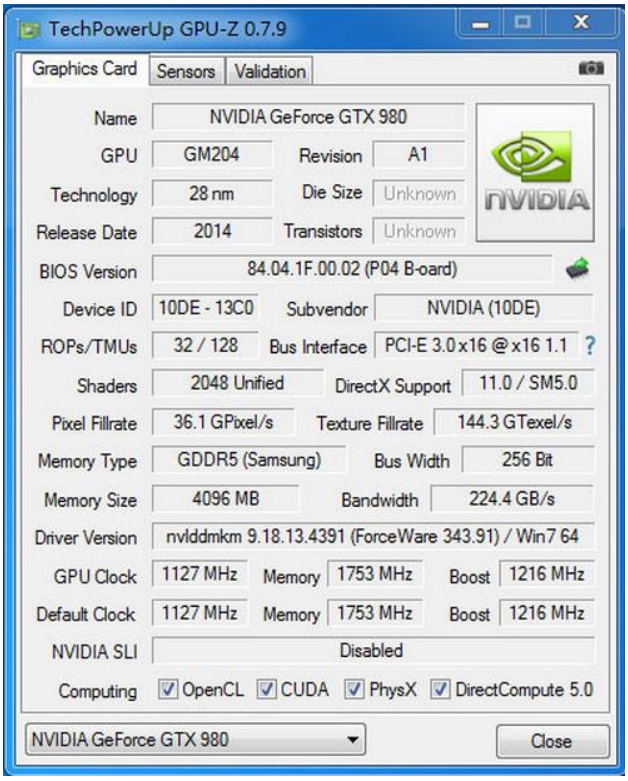


Fig. 14. This figure shows the parameter of the testing GPU GeForce GTX 980.

The testing environment we use is a desktop with a quad core i7-4900MQ processors. It has four processors and eight thread, the main frequency is 3.2MHz and the cache size is 20MB. The main board we use is GALAXY Technology X99 GAMING 5. The memory is Samsung DDR4-3000-16GB. The power we use is Antec 1200U HCP Platinum. The hard disk is Seagate 1T-7200 r/min. The driver software version is 344.07, the newest at that time.

A. Theoretical test

Before we operate applications, we use 3DMark to calculate the theoretical performance of the single GPU and multi-core system. During the process, this software will use GPU to generate a dramatic 3D graphs. The points can reflect both the computing ability and the grinder performance.

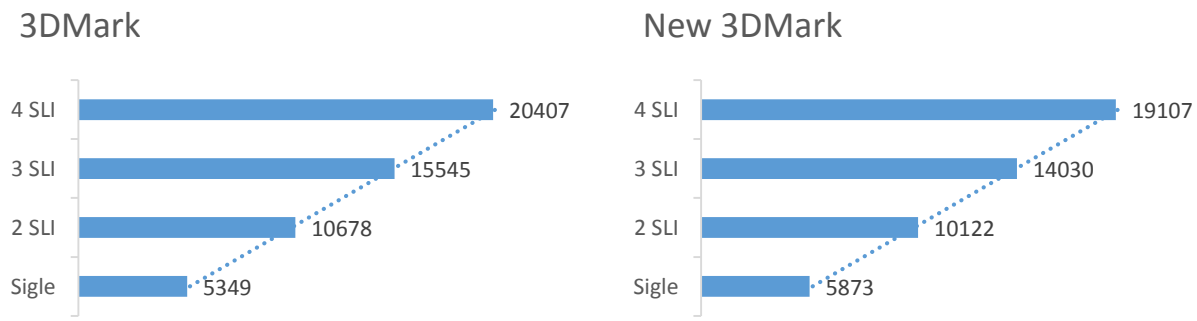


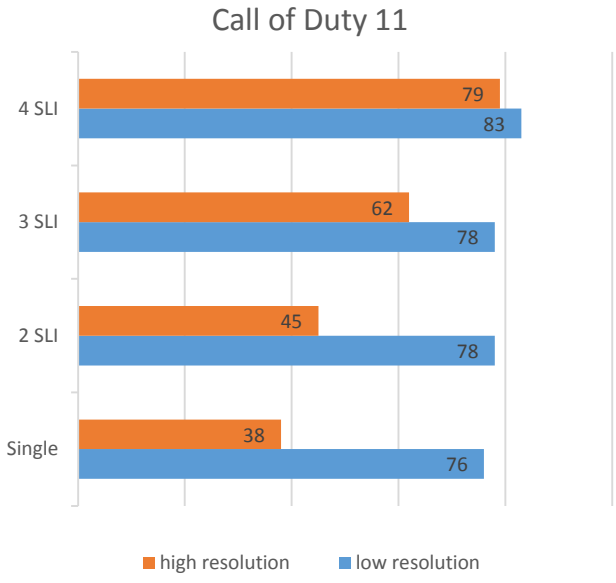
Fig. 15. This figure shows the parameter of the testing GPU GeForce GTX 980.

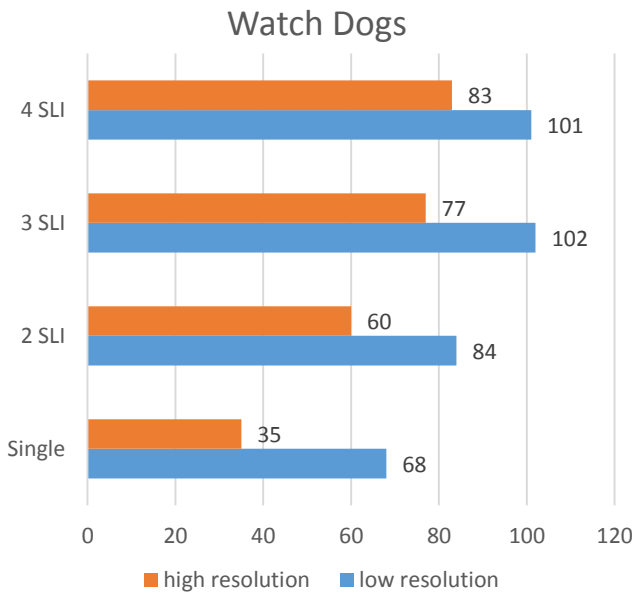
From the theoretical result of the four mood, we can easily found the liner relationship between the points and the number of GPUs. Compared with single, 2 SLI GPUs get nearly double points. And when one additional GPU is added, almost 5000 points is added to the total result. The points can in some degree stand for the GPU performance.

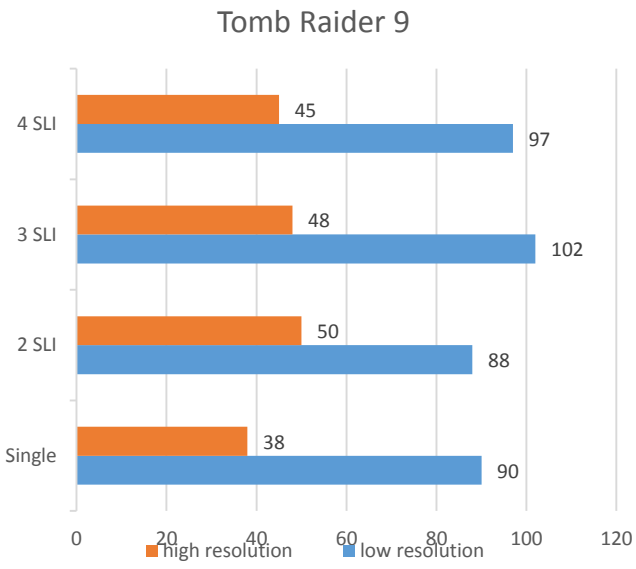
B. Application test

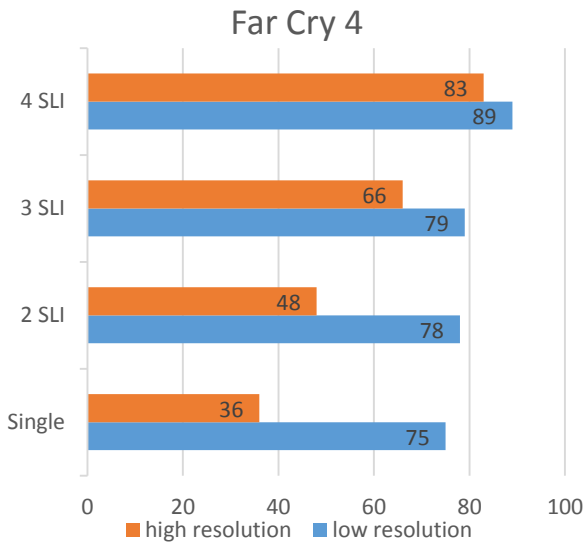
The theoretical result comes from the certain 3D graphs that is integrated within the software. So to explore in a more casual condition, how multi-core system will attribute the whole performance, we need to test the system in applications. Now we choose six huge 3D games as testing applications. They are Call of Duty 11, Watch Dogs, Tomb Raider 9, Far Cry 4, Splinter Cell 6, and Mass EFFECT. The reason we choose these six 3D games is that all of them require an extremely high performance of GPUs. Generally, we cannot compared the points of each games with each other. We can only compare the points with different mood of GPU that is used to test the same game. This is because different games have different 3D action engine and support different graph quality. So only the inter game comparison is meaningful. For each games, we have

two setting. In the first test, we set the resolution as 1920×1080 and other effect to be highest——the best image quality, the best effects and the max anti-aliasing. In the second test, we turn the resolution into 3840×2160 and leave other effect setting remain the same. If the game doesn't support such a resolution, we can set the GPU and supplement missing pixels.









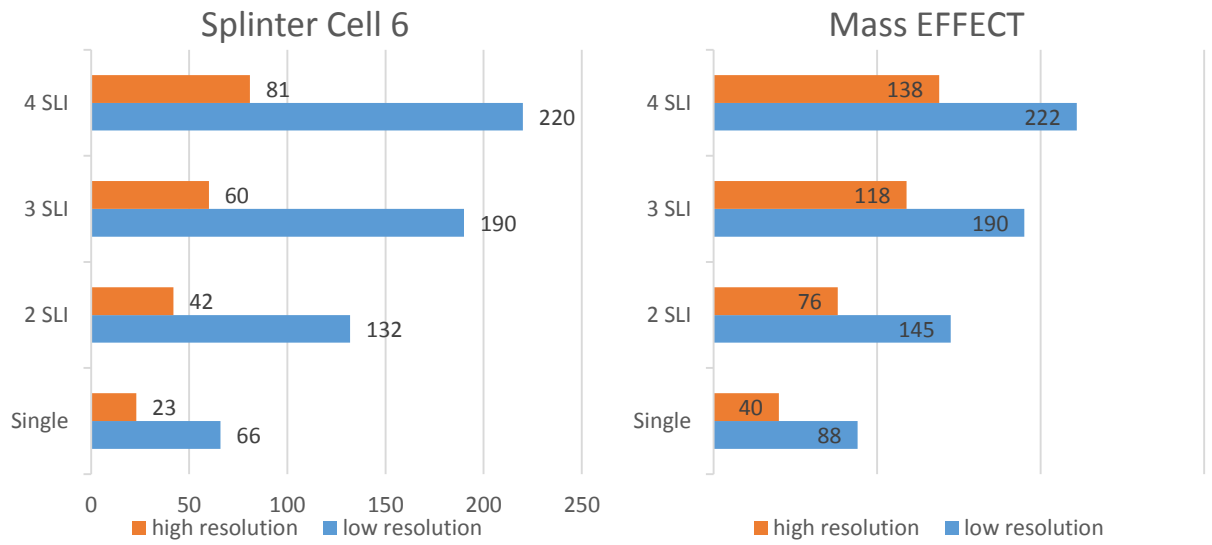


Fig. 16. This figure shows the testing result of GeForce GTX 980 in several applications.

From the result, we can easily find out that for some applications such as Call of Duty 11, Tomb Raider 9, Far Cry 4, when testing the low resolution mood, the four SLI mood has a tiny difference. The cause of this result may be the low requirement of these applications when output in a low resolution. Only one GPU is enough to handle the graphic task and the bottleneck of performance is not GPU. So when adding additional GPUs, no improvement is observed. But when transfer to a high resolution, the requirement is extremely high, so the performance will be one of the bottleneck of image quality. In this condition, we can see that single GPU did badly. When added additional GPUs, the points increased with the number of GPUs. This liner relationship can explain two points: (1) Even in the four GPU system, GPU performance was still a bottleneck. (2) The graph performance will truly be increased by the additional GPUs. In the Tomb Raider 9, only one GPU is enough for both low and high resolution, so when transfer to SLI mood, no noticeable improvement is funded.

Because the load of GPU system is strongly depend on the 3D graphs in the game, which means different movements or action frames may send different amount of task to GPU, so during the test, we only test performance for the same scene. But maybe due to the temperature of desktop, CPU or other factors, some unusual result was get such as the decrease of the Mass EFFECT in the low resolution mood. Another thing we can see is that the performance of two GPUs is not two times of a single. This is easy to understand since

the main GPU should have to divide and combine the frames.

V. CONCLUSIONS

A. Results

In a sum, multi-core GPU system indeed can improve the performance of GPUs and the effect of this strategy is obviously high. For some application, it can multiply the performance, which brings a better application experience.

B. Future works

With the development of GPUs and the requirement of GPU performances, multi-core GPUs system must become much more common. Predictably, in the next few years, the focus of work for GPU engineers should be the cooling system and the software framework that can farthest explore the potential of multi-GPUs. I used to be a GPU enthusiasts, I collected GPUs and gave evaluation reports for some websites. Before this paper, I cared more about the performance parameter and output graphs quality. But now from this paper, I get a comprehensive understanding of GPU principles and the rules of science paper.

VI. REFERENCE

[1] J. Tyson and T. V. Wilson, "How stuff works," [Online]. Available: http://computer.howstuffworks.com/graphics-card.htm. [Accessed 01 03 2014].

[2] "Wikipedia," [Online]. Available: http://en.wikipedia.org/wiki/Z-buffering. [Accessed 02 03 2015].

[3] "Wikipedia," [Online]. Available: http://en.wikipedia.org/wiki/Fillrate. [Accessed 19 02 2015].

[4] "GPUReview," [Online]. Available: serious performance gains can be had simply by cranking up a dial in your driver control panel. [Accessed 04 03 2015].

[5] T. Kreiss, "tom's HARDWARE," 5 08 2010. [Online]. Available: http://www.tomshardware.com/reviews/amd-crossfire-nvidia-sli-multi-gpu,2678-4.html.

[6] "emutalk.net," [Online]. Available: http://www.emutalk.net/threads/42527-Nintendo-64-Consoles-Specs?s=c42adc26aee07b113cf1fd3172114e8e. [Accessed 01 03 2015].

[7] F. Song, S. Tomov and J. Dongarra, "Enabling and Scaling Matrix Computations on," in *Proceedings of the 26th ACM international conference on Supercomputing*, 2012, July.

[8] J. Jin, S. J. Turner, B.-S. Lee, J. Zhong and B. He, "Simulation of Information Propagation over Complex Networks: Performance Studies on Multi-GPU," in *Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*, 2013, Oct.

[9] R. W. Fredrickson and A. C. Goris, Graphics frame buffer with strip Z buffering and programmable Z buffer location, Hewlett Packard Company, Oct 2, 1990.

[10] R. Ausavarungnirun, K. K.-W. Chang, L. Subramanian, G. H. Loh and O. Mutlu, "Staged memory scheduling: achieving high performance and scalability in heterogeneous systems," in *Proceedings of the 39th Annual International Symposium on Computer Architecture*, June 2012.

[11] L. Solano-Quinde, B. Bode and A. K. Somani, "Techniques for the parallelization of unstructured grid applications on multi-GPU systems," in *Proceedings of the 2012 International Workshop on Programming Models and Applications for Multicores and Manycores*, February 2012.

[12] M. Becchi, S. Byna, S. Cadambi and S. Chakradhar, "Data-aware scheduling of legacy kernels on heterogeneous platforms with distributed memory," in *Proceedings of the twenty-second annual ACM symposium on Parallelism in algorithms and architectures*, June 2010.

[13] V. Saravanan, S. Kaushik, P. S. Krishna and D. P. Kothari, "Performance analysis of multi-threaded multi-core CPUs," in *Proceedings of the First International Workshop on Many-core Embedded Systems*, June 2013.

[14] G. Kim, M. Lee, J. Jeong and J. Kim, "Multi-GPU System Design with Memory Networks," in *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, December 2014.

[15] T. Lutz, C. Fensch and M. Cole, "PARTANS: An autotuning framework for stencil computation on multi-GPU systems," in *Transactions on Architecture and Code Optimization (TACO)*, ACM, January 2013, pp. 59.4-59.8.