Um olhar sobre o estado da arte de arquiteturas de GPU: NVIDIA e AMD

Levy G. da S. Galvão[1] e Thiago M. Souto[2]

***Abstract*— As GPUs evoluíram a partir da necessidade de processamento gráfico tridimensional, atualmente possuindo aplicações de propósito geral em várias áreas: bioinformática, inteligência artificial, jogos, *data centers*, etc. Com o desenvolvimento, surge a necessidade de acompanhar as mudanças e melhorias realizadas em arquiteturas de GPU. Diante disso este trabalho pretende apresentar o básico de arquiteturas de GPU a partir do estado da arte de duas marcas diferentes que dominam o mercado: NVIDIA e AMD, de forma a explicitar os elementos principais de uma GPU e como eles diferem entre marcas. Também buscando comparar a tecnologia atual com a geração predecessora.**

I. INTRODUÇÃO

Uma unidade de processamento gráfico (GPU) é um processador desenvolvido especificamente para otimizar tarefas de renderização gráfica tridimensional e processamento de vídeo. Atualmente estão presentes em vários dispositivos e podem estar dispostos em duas formas: integrados no mesmo *chip* que a unidade de processamento central (CPU); ou com um circuito integrado totalmente dedicado em uma placa de coprocessador separado, geralmente baseadas em PCIe. Quando integradas, possuem uma baixa quantidade de *cores* (unidades) e se encontram em *tablets* e *smartphones*, por exemplo. As GPUs dedicadas possuem uma maior quantidade de *cores* (de centenas a milhares de unidades) de processamento paralelo e se encontram em estações de trabalhos, sistemas de jogos e supercomputadores. (STALLINGS, 2017)

Devido a sua semelhança com a CPU, faz-se necessário deixar claro suas diferenças. Elas são desenvolvidas e otimizadas para dois tipos diferentes de aplicações com arquiteturas que diferem significativamente. Uma grande área da CPU é reservada para memória *cache* e lógica de controle, na intenção de processar um código sequencial o mais rápido possível. Enquanto que a GPU usa uma arquiteutra de instrução única e dados múltiplos (SIMD) massivamente paralela para executar operações matemáticas, altamente previsíveis e que carecem de desvios.(STALLINGS, 2017)

Nos últimos anos a GPU deixou de trabalhar apenas com atividades com finalidade gráfica para encontrar seu caminho para as mais variadas aplicações, como: processamento de sinais, modelagem estatística, bioinformática, etc. Assim derivando o termo computação de uso geral usando uma GPU (GPGPU). Essa migração foi permitida graças às linguagens de GPGPU amigáveis ao programador e modificações na arquitetura para permitir a computação de uso geral. (STALLINGS, 2017)

O início da arquitetura da GPU, entre o início da década 1980 e o final da década de 1990, foi marcado por uma GPU com estágios especializados de processamento não programável e fixo. No fim da década de 1990, o desenvolvimento tecnológico permitiu a diminuição do tamanho e custos dos sistemas gráficos, permitindo trazer os processadores gráficos para PC. Em meados de 2000 houve uma mudança na arquitetura da GPU, de uma pipeline de hardware especializado e fixo para um processador totalmente programável. Em 2006 a NVIDIA realização uma alteração para permitir o uso de uma nova linguagem de GPGPU, CUDA, capaz de tornar um coprocessador SIMD altamente paralelizado e acessível a fim de acelerar o tempo de execução de programas sem finalidade gráfica. (STALLINGS, 2017)

Diante do rápido desenvolvimento na tecnologia de GPUs, este trabalho pretende mostrar o estado da arte da arquitetura de duas marcas diferentes que dominam o mercado: NVIDIA e AMD, de forma a explicitar os elementos principais de uma GPU e como eles diferem entre marcas. Também pretende comparar a geração atual de cada marca com sua predecessora.

II. ARQUITETURAS NVIDIA E AMD

Desde 2006, a NVIDIA progrediu em várias gerações de tecnologias de GPU, apresentando mudanças em suas microarquiteturas, mas mantendo a mesma ISA (CUDA).

- Tesla (Q3 2006);
- Fermi (Q1 2010);
- Kepler (Q1 2012);
- Maxwell (Q1 2014);
- Pascal (Q1 2016);
- Volta (Q3 2017);
- Turing (Q3 2018);
- Ampere (lançamento previsto para Q3 2020)
- Hopper (futuramente)

Enquanto isso, a AMD evoluiu na alteração de suas arquiteturas, porém seguindo um padrão de gerações.

- Tera Scale 1 (Q2 2007);
- Tera Scale 2 (Q3 2013);
- Tera Scale 3 (Q3 2010);
- Graphics Next Core Gen 1 (Q1 2012);
- Graphics Next Core Gen 2 (Q3 2013);
- Graphics Next Core Gen 3 (Q2 2015);
- Graphics Next Core Gen 4 (Q2 2016);
- Graphics Next Core Gen 5 (Q2 2017);

*This work was not supported by any organization

[1] Graduando em Engenharia Elétrica, Universidade Federal do Rio Grande do Norte, Brasil.

[2] Graduando em Engenharia Elétrica, Universidade Federal do Rio Grande do Norte, Brasil.

- RDNA 1 (Q2 2019);
- RDNA 2 (lançamento previsto para 2020);

As microarquiteturas também são uma figura que representam o desenvolvimento. Muitas vezes estas chegam até ser confundidas com as arquiteturas em si, sendo tratadas como sinônimos, como no caso da RDNA 1 e Navi. Mas vale destacar que enquanto que a microarquitetura trata de como o conjunto de instruções serão implementados, a arquitetura em si engloba ambos os conceitos.

- Southern Islands (2012);
- Sea Islands (2013);
- Volcanic Islands (2014);
- Polaris (Q2 2016);
- Vega (2017);
- Navi (2018);

III. Estado da arte da arquitetura de GPUs: RDNA (AMD) e Turing (NVIDIA)

Apesar da existência de diferentes fabricantes de GPU, suas arquiteturas possuem diversas semelhanças. Estas podem ser traduzidas em nomes diferentes para unidades básicas que possuem a mesma função, quantidade e posição diferente de blocos, etc.

Assim, após apresentar uma arquitetura básica e já ultrapassada, vale a pena comparar duas arquiteturas estado da arte de diferentes fabricantes líderes de mercado para entender as mudanças e melhorias.

A. Turing (NVIDIA) - chip processador TU102

A GPU do TU102 possui 6 blocos Graphics Processing Clusters (GPCs) que possuem uma Raster Engine (mecanismo de varredura) vinculada a uma série de TPCs; cada bloco GPC contém 6 blocos Texture Processing Clusters (TPCs); e cada bloco TPC contém 2 blocos Streaming Multiprocessor (SM).

O bloco GigaThread Engine é responsável por receber todas as *threads* e agendá-las e distribui-las. Elas são organizadas de tal forma que aquelas que possuem as mesmas instruções estejam agrupadas em grupos de 32. Essa coleção é chamada de *warp* e são executadas independentemente, não necessitando que outras acabem para serem executadas. Cada Streaming Multiprocessor trabalha com 4 *warps* (cada TPC fica com 2 *warps*). As unidades de execução da NVIDIA (CUDA *cores*) são escalares.

Aprofundando na SM do TU102, observa-se 4 CUDA cores, contendo:

- 1 unidade de agendamento de instrução e despacho;
- 16 ALUs escalares FP32 (IEEE754);
- 16 ALUs escalares INT32;
- 2 Tensor cores;
- 4 SFUs;
- 4 unidade de armazenar/carregar (lida com cache R/W);
- 2 FP64 (omitidas no diagrama de blocos);
- 4 unidades de textura;
- 1 Ray Tracing (RT) core.

A unidade FP32 (FP64) trabalha com números de ponto flutuante de 32-bits (64-bits) e a INT32 trabalha com números inteiros de 32-bits, e ambas trabalham concorrentemente. As SFUs apresentam operações como seno, cosseno, recíproca e raiz quadrada que são executadas em um ciclo de clock. Cada unidade de textura possui sistemas de endereçamento e filtragem de textura.

Os Tensor Cores são ALUs especializadas em operações matriciais e lidam com dados FP16, INT8 ou INT4, de forma que em um ciclo de clock ocorram até 64 operações de multiplicar-depois-somar. Isso permite que operações matriciais sejam feitas enquanto os CUDA cores realizam outras operações.

O RT Core é uma unidade especial e única da arquitetura Turing e que executa algoritmos matemáticos específicos que são utilizados para a tecnologia de ray tracing da NVIDIA.

A hierarquia de memória da arquitetura Turing trabalha com múltiplos níveis de cache para alcançar uma baixa latência e potência e alta largura de banda. Cada CUDA core possui um register file de 64 kB (256 kB por SM). Em seguida, há uma memória compartilhada de 96 kB para cada SM (64 kB de cache L1 e 32 kB de cache par textura ou espaço para registradores). Por fim, cada GPC tem acesso à memória cache L2 de 6144 kB.

As unidades de saída de renderização (ROP, render output unit) são componentes presentes na SM que atuam nos passos finais do processo de renderização. Elas atuam processando pixeis e elementos de textura por operações algébricas em pixeis finais (rasterização), controlando o antialiasing.

No caso da microarquitetura Turing, as unidades de textura podem endereçar e buscar 4 elementos de textura, filtram eles bilinearmente em um elemento e escrevem em cache em um ciclo de clock. Cada GPC contém dois ROP, cada qual emitindo 8 pixeis por clock. Ao todo, o chip TU102 emite 96 piexeis por clock.

B. RDNA (AMD) - chip processador Navi

A GPU do Navi possui 2 blocos chamados Shader Engines (SEs); estes podem ser divididos em outros 2 blocos chamados Asynchronous Compute Engines (ACEs); cada ACE comporta 5 blocos chamados Workgroup Processors (WGPs), que consiste de 2 Compute Units (CUs).

Observa-se que a hierarquia geral da arquitetura do processador Navi é semelhante a da arquitetura Turing, permitindo traçar paralelos entre elas.

Como por exemplo: o Graphics Command Processor do Navi possui a mesma função que a GigaThread Engine do TU102; os *warps* são chamados de *waves*, mas ainda mantém suas independências. Porém, cada Compute Unit pode lidar apenas com 2 *waves*. As unidades de execução da AMD (Stream Processors) trabalham com vetores, possuindo uma unidade dedicada para operações escalares.

Ao aprofundar na CU do Navi, observa-se um par de:

- 32 Stream Processors (SPs) que trabalham com ALUs vetoriais FP32 e INT32 (IEEE754);
- 1 Special Function Units (SFU);
- 1 ALU escalar INT32;
- 4 texture units.

Vale destacar que cada um dos SPs possuem sua própria instruction unit, permitindo que uma *wave32* possa ser resolvida em um ciclo de clock para cada conjunto de SPs. Essa arquitetura permite que as vector units lidem com *waves* de 16 *threads* e *waves* de 64 *threads*, porém, com uma taxa duas vezes mais rápida e duas vezes mais lenta, respectivamente. Contrastando com o Tensor Core da TU102, a Navi também realiza operações matriciais, mas requer um maior número de SPs.

A hierarquia de memória do Navi é melhor detalhada que a do Turing. Começando pelo nível mais baixo, o de register file, cada SP possui 256 kB de registradores vetoriais de propósito geral. O register file de cada unidade escalar tem capacidade para 32 kB.

Cada par de compute unit compartilha um cache LO de instruções de 32 kB e um cache de dado escalares de 16 kB. Mas cada CU possui um próprio cache L0 vetorial de 32 kB. Conectando toda essa memória às ALUs existe um compartilhamento de memória local de 128 kB. Em termos de cache L1, uma vez que 2 CU formam um WGP, e 5 desses formam um ACE, tem-se que cada ACE tem acesso a sua própria cache L1 de 128 kB. No geral, toda a GPU é suportada por uma cache L2 de 4 MB, que está conectado com as caches L1 e outras sessões do processador. Para maximizar a largura de banda da memória, o Navi implementa compressão de cor sem perdas entre L1, L2 e a memória local GDDR6.

Em relação às ROPs não há tanta diferença da AMD para a NVIDIA. Assim como em Turing, na RDNA a capacidade de endereçamento e busca são iguais. Porém o arranjo delas são diferentes. No chip da AMD possui 4 render backends (RBs) por ACE e cada um pode emitir 4 pixeis misturados por ciclo de clock, totalizando 64 para o chip da Navi.

IV. EVOLUÇÃO DAS ARQUITETURAS

Essa seção pretende fazer uma análise entre cada uma das arquiteturas apresentadas em tópicos anteriores e comparar quais recursos adicionais cada uma possui em relação à sua predecessora, para cada marca.

A. Da microarquitetura Pascal para a Turing

Apesar da Turing ser uma sucessora da Pascal, a NVIDIA pretende manter as duas co-existindo e não competindo, apesar de Turing representar um salto em melhoria de desempenho (melhor que a Pascal), ela possui um preço alto, deixando-a em uma categoria ao lado da Pascal. Um dos recursos que a encarecem é a tecnologia de Ray Tracing (RT), que por muitos usuários é criticada, pois atualmente o mercado possui poucos jogos que implementam essa tecnologia. Aliando o RT e definição 4K, torna-se um desafio aos desenvolvedores de jogos para mantar uma boa taxa de quadros por segundos, prejudicando a experiência dos jogos. Dessa forma muitos usuários buscam refúgio em chips da família Pascal.

Um detalhe a relevante é que a microarquitetura predecessora cronologicamente da Turing é a Volta. Porém ela não será levada em consideração aqui, pois sua aplicação é voltada para data centers, enquanto que a Pascal e Turing são voltadas para estações de trabalho.

A arquitetura Turing se propõe como uma arquitetura de GPU de propósito múltiplo que pode executar três coisas ao mesmo tempo: processamentos de pixeis, inteligência artificial (IA) e Ray Tracing em tempo real. Enquanto que Pascal envelhece por não possuir recursos avançados para IA e ray tracing.

Outro ponto a ser destacado é a comparação entre as tecnologias de fabricação da Pascal e Turing, que houve uma redução de 16nm (GeForce GTX 1080) para 12nm (GeForce RTX 2080); em seguida o aumento da quantidade de transistores e o tamanho da matriz, de 7.2 bilhões e 314 mm^2 para 13.6 bilhões para 545 mm^2.

O número de SMs aumentou, enquanto o número de CUDA cores em cada SM reduziu. Em alguns chips isso fez manter o mesmo número médio de CUDA cores na GPU, porém essa fragmentação permite um uso mais completo da GPU, com múltiplos grupos de instruções independentes sendo processados paralelamente.

B. Da arquitetura GCN para a RDNA

Iniciando pela CU, esta possui mudanças, porém elas são na ordem de como os componentes foram organizados. Na RDNA, cada conjunto de 32 SPs possui sua própria unidade de instrução, enquanto que a GCN tinha apenas um agendador para 4 conjuntos de 16 SPs. Significa dizer que uma wave32 pode ser resolvida em um ciclo de clock. A arquitetura RDNA também permite que as unidades vetoriais lidem com wave16 (16 threads) ou wave64 (64 threads), uma vez que no GCN só são possível waves64.

A quantidade de SFUs foi reduzida, mas elas podem operar com conjunto de dados com o dobro de tamanho.

Na arquitetura GCN, as CU formam as entidades básicas de *shade*, contendo ALUs, carregar/armazenar e acesso a memória. Na RDNA o WGP forma a unidade básica e cada WGP contém 2 CU, permitindo maior poder computacional a largura de banda de memória pode ser redirecionada para um único WGP.

V. PERSPECTIVAS FUTURAS

A arquitetura RDNA 2 (7nm), sucessora da RDNA e a arquitetura Ampere (7nm), sucessora da Turing, estão previstas para serem lançadas em 2020.

Ainda existe pouca documentação sobre a RDNA 2, mas a AMD promete uma performance extrema com eficiência de energia; rigidez para a resolução 4K para jogos; e hardware com suporte a Ray Tracing e Variable Rate Shading.

Para a Ampere, se espera uma GPU GA100 poderosa dentro dos cards A100. Possuirá 108 SMs, contando com 64 FP32, 32 FP64 e 64 INT32 núcleos dentro de cada SM, e daí em diante.

Uma proposta de trabalho futuro pode ser avaliar o desempenho da nova arquitetura Ampere em comparação com a antiga arquitetura Volta.

VI. CONCLUSÕES

O mercado de GPUs possuem uma gama de variedades. Neste trabalho foi levado em conta apenas um recorte de duas arquiteturas de duas empresas, com chips voltados para o uso pessoal.

Atualmente com a capacidade de as GPUs trabalharem com propósitos gerais, pode-se encontrar microarquiteturas NVIDIA, como a Volta e Ampere que são voltadas para sistemas mais robustos como *data centers*.

O profissional da área deve se manter atualizado com as novas arquiteturas, porém conhecer as gerações anteriores é essencial para a tomada de decisões. Isso pois, uma nova e mais robusta arquitetura não significa que será economicamente viável para o projeto. Recursos adicionais como no caso do Ray Tracing da NVIDIA pode sair caro, caso não haja a intenção de usá-lo.

REFERENCES

[1] Stallings, William. Computer organization and architecture: designing for performance. Pearson Education India, 2017.

[2] NVIDIA. NVIDIA Turing GPU architecture. Graphics reinvented. WP-09183-001v01. 2018

[3] AMD. Introducing RDNA architecture. The all new Radeon TM gaming architecture powering “Navi”. 2019

[4] Patterson, David A., and John L. Hennessy. Computer Organization and Design RISC-V Edition: The Hardware Software Interface. Morgan kaufmann, 2018.

[5] Taylor, Paul. "Nvidia confirms Pascal and Turing GPUs will co-exist, but not compete." TechSpot, 10 Sep. 2018, https://www.techspot.com/news/76347-nvidia-confirms-pascal-turing-gpus-co-exist-but.html

[6] Evanson, Nick. “Navi vs. Turing: An Architecture Comparison.” TechSpot, 19 Dec. 2019, www.techspot.com/article/1874-amd-navi-vs-nvidia-turing-architecture/.

[7] Lambert, Matthew. "Nvidia’s Turing Architecture Explained." Bit-tech, 14 Sep. 2018, https://bit-tech.net/features/tech/graphics/nvidias-turing-architecture-explained/2/.

[8] Verma, Akshat. "RDNA vs Navi vs GCN: What is the Difference What they Mean?." Graphics Card Hub, 24 Jan. 2020, https://graphicscardhub.com/rdna-vs-navi-vs-gcn/.

[9] Gulati, Abheek. "An Overview of AMD’s GPU Architectures." Medium, 11 Nov. 2019, https://medium.com/high-tech-accessible/an-overview-of-amds-gpu-architectures-884432a717a6.