Nghiên cứu và tìm hiểu kiến trúc GPU Nvidia

# Tổng quan về điện toán GPU (GPU computing)

## GPU là gì?

GPU viết tắt cho Graphics Processing Unit được gọi là Đơn vị xử lý đồ họa ( đôi khi được gọi VPU – Visual Processing Unit) là một bộ vi xử lý chuyên dụng nhận nhiệm vụ tăng tốc, xử lý đồ họa cho bộ vi xử lý trung tâm CPU.

Các GPU hiện đại có năng suất rất cao trong xử lý đồ họa máy tính. Với cấu trúc mang tính xử lý song song mạnh mẽ của mình, GPU cho thấy nó hiệu quả hơn CPU rất nhiều trong nhiều thuật toán phức tạp.

GPU được sử dụng trong các hệ thống nhúng, điện thoại di động, máy tính cá nhân, máy trạm, máy chơi game v…v.. Trong máy tính cá nhân, một GPU có thể xuất hiện ở card đồ họa, hoặc nó cũng có thể được gắn trên mainboard.

## Lịch sử phát triển của điện toán GPU (GPU computing)

GPU được đưa ra và giới thiệu lần đầu tiên vào 31-8-1999 bởi NVIDIA và được xem như bộ xử lý song song phổ biến nhất cho đến nay. Dựa trên những mong muốn chưa được thỏa mãn về khả năng đồ họa như thật, thời gian thực, GPU đã phát triển trở thành bộ xử lý có thể biểu diễn các phép tính dấu chấm động (floating-point) cũng như khả năng lập trình. Những GPU ngày nay đã thật sự vượt qua các CPU trong khả năng tính toán số học và băng thông bộ nhớ, khiến chúng trở thành bộ xử lý lý tưởng để tăng tốc nhiều ứng dụng xử lý dữ liệu song song khác nhau.

**GPGPU**

Các nỗ lực để khai thác GPU cho các ứng dụng không mang tính đồ họa đã được tiến hành từ năm 2003. Đặc biệt trong khoa học máy tính, cùng với các nhà nghiên cứu trong các lĩnh vực như hình ảnh y khoa, điện từ… Họ đã bắt đầu sử dụng GPU để chạy các ứng dụng điện toán đa dụng (general purpose). Họ đã nhận thấy sự xuất sắc trong khả năng biểu diễn các phép tính dấu chấm động của các GPU, dẫn đến tăng hiệu suất rất lớn cho một loạt các ứng dụng khoa học. Từ đó dẫn đến sự ra đời của GPGPU – General Purpose computing (Điện toán đa dụng).

**GPU computing**

Tuy mô hình GPGPU đã chứng minh khả năng tăng tốc tuyệt vời của mình, nhưng nó vẫn phải đối mặt với một số vấn đề. Thứ nhất, nó đòi hỏi các lập trình viên phải có kiến thức sâu sắc về các giao diện lập trình ứng dụng đồ họa (graphics APIs) và kiến trúc của GPU. Thứ hai, các vấn đề phải được thể hiện trong các thuật ngữ của các chương trình xác định tọa độ đỉnh (**vertex coordinate)**,kết cấu ( **textures)** và đổ bóng (**shader),** qua đó tăng sự phức tạp lên rất nhiều. Thứ ba, các tính năng lập trình cơ bản như truy xuất đọc, viết ngẫu nhiên vào bộ nhớ đã không được hỗ trợ, giới hạn phần nào mô hình lập trình. Cuối cùng, việc thiếu hỗ trợ độ chính xác kép (**double precision**) khiến cho một số ứng dụng khoa học không thể chạy trên GPU.

Để khắc phục những vấn đề này, NVIDIA đã giới thiệu 2 kỹ thuật then chốt:

* Đồ họa hợp nhất và kiến trúc tính toán G80 (được giới thiệu trong các GPU GeForce 8800®, Quadro FX 5600® và Tesla C870®)
* CUDA: một kiến trúc phần mềm và phần cứng cho phép lập trình trên GPU bằng nhiều ngôn ngữ lập trình cấp cao (C, C++, Fortran…)

Cùng với nhau, 2 kỹ thuật này được xem như là một cách tiếp cận mới trong việc sử dụng GPU. Thay cho các đơn vị lập trình đồ họa chuyên dụng với các giao diện lập trình ứng dụng đồ họa (graphic API), giờ đây lập trình viên có thể viết các chương trình C với sự mở rộng của kiến trúc CUDA và sử dụng vào các mục đích tổng quan hơn (không chỉ dành cho lập trình đồ họa) với bộ xử lý song song mạnh mẽ. Và NVIDIA gọi đây là cách lập trình mới trên GPU – Điện toán GPU (**GPU computing**) – hỗ trợ nhiều ứng dụng hơn, nhiều ngôn ngữ lập trình hơn, và phân chia một cách rõ ràng so với mô hình lập trình “GPGPU” trước đây.

## Những lĩnh vực ứng dụng của GPU

# Kiến trúc GPU

## Kiến trúc G80

## Kiến trúc GT200

## Tổng quan về kiến trúc Fermi

Là kiến trúc Fermi đầu tiên dựa trên nền tảng GPU, được bổ sung 3 triệu transistor và có đến 512 nhân CUDA. Một nhân CUDA thực thi 1 chỉ thị dấu chấm động hoặc số nguyên trên 1 clock cho 1 thread. 512 nhân CUDA được tổ chức vào 16 SM với mỗi SM chứa 32 nhân. GPU có 6 phân vùng bộ nhớ 64-bit, for a 384-bit memory interface, hỗ trợ lên đến tổng cộng 6 GB bộ nhớ GDDR5 DRAM. Một host interface kết nối GPU đến CPU thông qua cổng PCI-Express. Bộ lập lịch trình toàn cục Giga Thread phân phối các khối thread đến các bộ lập lịch trình của riêng từng SM.

### Third Generation Streaming Multiprocessor

Thế hệ SM thứ 3 giới thiệu nhiều kiến trúc mang tính đột phá khiến nó không chỉ là SM mạnh mẽ nhất từng được xây dựng, về cả khả năng lập trình và tính hiệu quả.

**512 nhân CUDA hiệu suất cao**

Mỗi SM có 32 bộ xử lý CUDA – một sự cải thiện gấp 4 lần so với các thiết kế SM trước đây. Mỗi bộ xử lý CUDA có đầy đủ các pipeline đơn vị luận lý toán học số nguyên(ALU) và đơn vị dấu chấm động (FPU). Các GPU trước đây sử dụng các bộ số học dấu chấm động IEE 754-1985. Kiến trúc Fermi bổ sung chuẩn floating-point mới là IEEE 754-2008, cung cấp chỉ thị fused multiply-add (FMA) cho cả các phép tính chính xác đơn và chính xác kép. FMA được cải thiện hơn chỉ thị multiply-add (MAD) bằng cách thực hiện các phép nhân và phép cộng chỉ với 1 bước final rounding đơn, không mất sự chính xác trong phép cộng. FMA chính xác hơn thực hiện các tính toán riêng rẽ. GT200 đã bổ sung bộ FMA chính xác kép.

Trong GT200, bộ luận lý toán học số nguyên bị giới hạn độ chính xác chỉ 24-bit cho các thao tác nhận; và kết quả là các đa chỉ thị cạnh emulation sequences were require for integer arithmetic. Trong kiến trúc Fermi, bộ luận lý toán học số nguyên mới hỗ trợ đầy đủ độ chính xác 32-bit cho tất cả các chỉ thị, ổn định đối với các đòi hỏi của các ngôn ngữ lập trình chuẩn. Bộ luận lý toán học số nguyên ALU cũng đồng thời tối ưu để hỗ trợ hiệu quả các thao tác chính xác 64-bit và chính xác mở rộng. Đạ dạng các chỉ thị được hỗ trợ, bao gồm Boolean, shift, move, compare, convert, bit-field extract, bit-reverse insert và population count.

**16 đơn vị Load/Store**

Mỗi SM có 16 đơn vị load/store, cho phép địa chỉ nguồn và địch có thể được tính trong 16 thread mỗi clock. Hỗ trợ các đơn vị tải và lưu trự dữ liệu ở mỗi địa chỉ để lưu trữ trên cache hoặc DRAM.

**4 đơn vị chức năng đặc biệt**

Các đơn vị chức năng đặc biệt (SFUs) thực thi chỉ thị siêu việt như là sin, cosine, phản đề(reciprocal) và căn bậc 2. Mỗi SFU thực thi 1 chỉ thị trên 1 thread mỗi clock; 1 warp thực thi 8 clock. Pipeline SFU được tách riêng ra từ đơn vị điều phối (dispatch), cho phép đơn vị điều phối hỗ trợ những đơn vị thực thi khác trong khi SFU đang bị chiếm.

**Thiết kế cho độ chính xác động**

Độ chính xác kép số học là trái tim của những ứng dụng HPC như linear algebra, numerical simulation và quantum chemistry. Kiến trúc Fermi đã được thiết kế đặc biệt để cung cấp sự thi hành các tính toán chính xác kép chưa được biết trước, lên đến 16 double precision fused multiply-add operation và có thể được thi hành trên mỗi SM, mỗi clock, một sự cải thiện gây ấn tượng sấu sắc so vượt trên kiến trúc GT200.

**Dual Warp Scheduler**

SM sắp xếp các thread vào 1 nhóm 32 thread song song được gọi là Warp. Mỗi SM có 2 bộ lập lịch trình warp và 2 đơn vị phân phối chỉ thị, cho phép 2 warp được đưa ra và thực thi đồng thời. Bộ lập lịch trình kênh đôi warp chọn 2 warp, và đưa ra 1 chỉ thị từ mỗi warp đến 1 nhóm 16 nhân, 16 đơn vị load/store hoặc 4 SFU. Vì warp thực thi độc lập, bộ lập lịch trình của Fermi không cần kiểm tra các phụ thuộc từ trong các luồng chỉ thị.

Hầu hết các chỉ thị có thể được đưa ra theo đôi, 2 chỉ thị số nguyên, 2 chỉ thị floating hoặc một sự pha trộn giữa các chỉ thị interger, floating point, load, store và SFU có thể được đưa ra đồng thời. Các chỉ thị chính xác kép không hỗ trợ phân phối kênh đôi với bất kỳ thao tác nào khác.

**Bộ nhớ dùng chung có thể cấu hình 64 KB(64KB Configurable Shared Memory) và L1 Cache**

Một trong những kiến trúc đột phá chính đã cải thiện khả năng có thể lập trình và sự thực thi của các ứng dụng GPU đó là bộ nhớ dùng chung được gắn trên chip. Bộ nhớ dùng chung cho phép các thread trong cùng 1 thread block hợp tác, làm dễ dàng hơn sự mở rộng khả năng tái sử dụng dữ liệu trên chip và giảm thiểu off-chip traffic. Bộ nhớ dùng chung chính là chìa khóa cho nhiều ứng dụng CUDA hiệu suất cao.

G80 và GT200 có 16 KB dung lượng của bộ nhớ chung trên mỗi SM. Trong kiến trúc Fermi, mỗi SM có 16 KB bộ nhớ trên chip, nhưng có thể được cấu hình thành 48 KB bộ nhớ dùng chung và 16 Kb bộ nhớ đệm L1 hoặc ngược lại 16 KB bộ nhớ dùng chung và 48 KB bộ nhớ đệm L1.

Đối với những ứng dụng có mở rộng việc sử dụng bộ nhớ dùng chung, việc nhân 3 dung lượng bộ nhớ dùng chung (từ 16 KB lên 48 KB) mang lại sự cải thiện đáng kể về hiệu suất, đặc biệt cho như những vấn đề bị ràng buộc về băng thông bộ nhớ. Còn đối với những ứng dụng sử dụng bộ nhớ dùng chung như là phần mềm quản lý vùng nhớ đệm, code có thể được sắp xếp hợp lý để sử dụng ưu điểm của hệ thống phần cứng vùng nhớ đệm, trong khi vẫn có thể truy cập ít nhất 16 KB bộ nhớ dùng chung cho sự phối hợp giữa các thread tường minh.

## Thế hệ Parallel Thread Execution ISA thứ 2

Fermi là kiến trúc đầu tiên hỗ trợ các tập chỉ thị mới Parallel Thread eXecution (PTX) 2.0. PTX là 1 máy ảo mức thấp và ISA được thiết kế để hỗ trợ các thao tác của bộ xử lý thread song song. Ở thời điểm cài đặt chương trình, các chỉ thị PTX được chuyển đổi sang chỉ thị máy bởi GPU driver.

Các mục tiêu chính của PTX là:

Cung cấp 1 ISA vững chắc trải rộng trên nhiều thế hệ GPU

Đạt được hoàn toàn hiệu suất của GPU trong các ứng dụng đã được biên dịch

Cung cấp 1 ISA độc lập cho C, C++, Fortran và các trình biên dịch khác

Cung cấp 1 bộ phân bố code ISA cho các lập trình viên ứng dụng và middleware

**….**

PTX 2.0 giới thiệu nhiều tính năng có thể cải thiện đáng kể khả năng lập trình, sự chính xác và hiệu suất của GPU. Những tính năng này bao gồm: đầy đủ độ chính xác dấu chấm động IEEE 32-bit, không gian địa chỉ hợp nhất cho tất cả các biến và con trỏ, địa chỉ 64-bit và các chỉ thị mới cho OpenCL và DirectCompute. Nhưng quan trọng nhất, PTX 2.0 được thiết kế đặc biệt để cung cấp sự hỗ trợ đầy đủ cho ngôn ngữ lập trình C++

**Không gian địa chỉ hợp nhất cho phép hỗ trợ đầy đủ C++**

Fermi và PTX 2.0 ISA bổ sung không gian địa chỉ hợp nhất nhằm hợp nhất những không gian địa chỉ riêng biệt (địa chỉ cục bộ của thread, của block dùng chung và toàn cục) cho các thao tác tải và lưu trữ. Ở PTX 1.0, các chỉ thị load/store được chỉ ra cụ thể cho một trong 3 không gian địa chỉ này; các chương trình có thể tải hoặc lưu trữ các giá trị vào các không gian địa chỉ được chỉ định vào lúc biên dịch chương trình. Rất khó để có thể bổ sung đầy đủ con trỏ của C và C++ vì không gian địa chỉ của con trỏ có thể không được biết trong lúc biên dịch, và có thể chỉ được xác định trong lúc chạy chương trình.

Với PTX 2.0, một không gian địa chỉ hợp nhất sẽ hợp nhất tất cả 3 không gian địa chỉ vào 1, tạo thành 1 không gian địa chỉ liên tiếp nhau. Một tập đơn các chỉ thị load/store hợp nhất thao tác trên các không gian địa chỉ này, làm tăng thêm những tập chỉ thị load/store của 3 vùng nhớ riêng biệt cục bộ, dùng chung và toàn cục. The 40-bit unified…

Sự bổ sung của không gian địa chỉ hợp nhất cho phép Fermi hỗ trợ các chương trình viết bằng ngôn ngữ C++ thật sự. Trong C++, tất cả các biến và hàm đặt trong các đối tượng và sẽ được truyền đi theo con trỏ. PTX 2.0 khiến điều đó là có thể để sử dụng các con trỏ hợp nhất để truyền caca1 đối tượng trong bất kỳ không gian bộ nhớ nào, và các đơn vị phần cứng chuyển đổi địa chỉ của Fermi tự động anh xạ các tham chiếu của con trỏ đến đúng không gian bộ nhớ.

Fermi và PTX 2.0 ISA cũng hỗ trợ thêm cho C++ hàm ảo, con trỏ hàm và các toán tử ‘new’ và ‘delete’ để cấp phát và thu hồi động các đối tượng. Các thao tác quản lý exception của C++ như ‘try’ và ‘catch’ cũng được hỗ trợ.

**Tối ưu hóa cho OpenCL và DirectCompute**

OpenCL và DirectCompute có liên quan mật thiết với mô hình lập trình CUDA, chia sẽ những ý niệm chính của thread, thread block, grid, barrier synchronization, bộ nhớ dùng chung của mỗi block, bộ nhớ toàn cục và các thao tác atomic. Fermi, một kiến trúc CUDA thế hệ thứ 3, tự nhiên nó đã tối ưu hóa rất tốt cho những API này. Ngoài ra, Fermi còn cung cấp phần cứng hỗ trợ cho OpenCL và DirectCompute các chỉ thị chuyển đổi định dạng, cho phép đồ họa và các chương trình tính toán dễ dàng thao tác trên cùng dữ liệu. PTX 2.0 ISA cũng hỗ trợ thêm cho DirectionCompute các chỉ thị population count, append và bit-reverse.

**IEEE 32-bit Floating Point Precision**

Các chỉ thị dấu chấm động chính xác đơn bây giờ hỗ trợ mặc định các con số subnormal trong phần cứng cũng như 4 chế độ làm tròn IEEE 754-2008 (neares, zero, positive infinity và negative infinity)

Những con số subnormal là những con số nhỏ nằm giữa 0 và số con số bình thường hóa nhỏ nhất của một hệ thống số dấu chấm động. Các thế hệ GPU trước đây thi hành các tính toán subnormal trong các phần mềm có quản lý exception, sử dụng cả ngàn vòng. Đơn vị dấu chấm động của Fermi subnormal các con số trong phần cứng, cho phép các giá trị dần dần trở về 0…??

Một chuỗi các thao tác thường được sử dụng trong đồ họa máy tính, linear algebra và các ứng dụng khoa học là nhấn 2 số, cộng kết quả với số thứ 3. Ví dụ: D = A x B + C. Các thế hệ GPU trước đây tăng tốc hàm này với chỉ thị multiply-add (MAD) cho phép cả 2 thao tác có thể được thực hiện chỉ trong 1 clock. Chỉ thị MAD thi hành một phép nhân sử dụng phương pháp cắt ngắn và làm tròn số. Fermi bổ sung chỉ thị mới fused multiply-add (FMA) cho cả các con số dấu chấm động độ chính xác đơn 32-bit và độ chính xác kép 64-bit (GT200 đã hỗ trợ FMA chỉ với độ chính xác kép), qua đó đã cải thiện multiply-add bằng cách giữ lại đầy đủ độ chính xác trong từng giai đoạn trung gian. Việc tăng độ chính xác giúp ích rất nhiều cho các thuật toán số học, như vẽ hình học giao nhau, chính xác hơn trong các vòng lặp phép tính toán học, làm tròn chính xác và nhanh hơn trong các thao tác phép chia và căn bậc 2.

**Improved Conditional Performance through Predication**

## Những đổi mới của hệ thống bộ nhớ con

**NVIDIA Parallel DataCache with Configurable L1 and Unified L2 Cache**

Làm việc với hàng trăm ứng dụng điện toán GPU từ nhiều lĩnh vực, Nvidia nhận ra rằng mặc dùng bộ nhớ dùng chung Shared memory mang lại lợi ích cho nhiều vấn đề, nhưng lại không thích hợp cho tất cả vấn đề. Một vài thuật toán ánh xạ một cách tự nhiên vào Shared memory, những thuật toán khác lại đòi hỏi bộ nhớ đệm, trong khi một số khác thì lại đòi hỏi sự kết hợp của cả 2. Hệ thống phân cấp tối ưu hóa bộ nhớ nên cung cấp ích lợi từ cả Shared memory và cache và cho phép lập trình viên một sự lựa chọn dựa trên những phân hoạch của nó. Hệ thống phân cấp bộ nhớ của Fermi đáp ứng cả 2 điều này.

Việc thêm vào một hệ thống bộ nhớ đệm phân cấp cho các thao tác load / stored đã giải quyết những vấn đề thật sự có ý nghĩa. Những kiến trúc GPU truyền thống hỗ trợ đường dẫn “load” read-only cho các thao tác texture và đường dẫn “export” write-only cho xuất dữ liệu pixel. Tuy nhiên, cách tiếp cận này khó thích hợp để thực thi các chương trình có sử dụng thread C hoặc C++ cho nhiều mục đích mà đòi hỏi read và write phải có thứ tự. Như 1 ví dụ sau: spilling…

**GPU đầu tiên hỗ trợ bộ nhớ ECC**

Fermi là GPU đầu tiên hỗ trợ Error Correcting code (ECC) dựa trên sự bảo vệ của dữ liệu trên bộ nhớ. Người dùng điện toán GPU đòi hỏi phải có ECC để nâng cao tính toàn vẹn dữ liệu trong những môi trường điện toán hiệu suất cao. EEC là 1 tính năng được mong đợi rất nhiều trong những lĩnh vực như medical imaging và large-scale cluster computing.

Những bức xạ xuất hiện tự nhiên có thể khiến 1 bit lưu trữ trên bộ nhớ bị thay đổi, dẫn đến một lỗi mềm. Công nghệ ECC phát hiện và sửa những lỗi mềm bit-đơn này trước khi chúng ảnh hưởng đến hệ thống. Bởi vì xác suất của những bức xạ như vậy gây ra các lỗi tăng tuyến tính cùng với số hệ thống được cài đặt, cho nên ECC là một yêu cầu thiết yếu trong những large cluster installation.

Fermi hỗ trợ các mã Single-Error Correct Double-Error Detact (SECDED) ECC có thể sửa bất kỳ lỗi bit đơn nào trong phần cứng cũng như dữ liệu khi được truy cập. Ngoài ra , SECDED ECC bảo đảm tất cả các lỗi bit kép và các lỗi đa bit cũng sẽ được phát hiện và báo cáo, do đó chương trình có thể được chạy lại chứ không phải tiếp tục chạy với những dữ liệu sai.

Những file register của Fermi, những bộ nhớ dùng chung, bộ đệm L1, L2 và bộ nhớ DRAM là những thành phần được ECC bảo vệ, điều này khiến cho ECC không chỉ là một phần quan trọng nhất của GPU trong các ứng dụng HPC mà còn là thành phần đáng tin cậy nhất. Ngoài ra, Fermi hỗ trợ các chuẩn công nghiệp như kiểm tra dữ liệu trong suốt quá trình chuyển đổi từ chip sang chip. Tất cả các GPU của NVIDIA hỗ trợ luôn cả chuẩn PCI Express cho CRC kiểm tra cùng với thử lại ở tầng liên kết dữ liệu. Fermi cũng hỗ trợ những chuẩn tương tự GDDR5 cho CRC kiểm tra cùng với thử lại (aka “EDC”) trong suốt quá trình chuyển đổi dữ liệu thông qua memory bus.

**Fast Atomic Memory Operations**

Những thao tác bộ atomic memory là quan trong torng lập trình song song, cho phép đồng thời nhiều thread biểu diễn đúng các đắn thao tác đọc-sửa-viết trên những cấu trúc dữ liệu dùng chung. Các thao tác nguyên tử như add, min, max và compare-swap là những thao tác nguyên tử trong ý nghĩa của các thao tác đọc, sửa và viết, được thi hành mà không có sự ngắt quãng bởi những thread khác. Nhugn74 thao tác nguyên tử của bộ nhớ được sử dụng rộng rãi cho sắp xếp song song, giảm thiểu caca1 thao tác và xây dựng cấu trức dự liệu song song mà không khóa sự thực thi của những serialize thread.

Với sự kết hợp của nhiều đơn vị nguyên tử trong phần cứng và thêm cả bộ đệm L2, các thao tác nguyên tử của Fermi biểu diễn nhanh gấp 20x lần khi so sánh với thế hệ GT200.

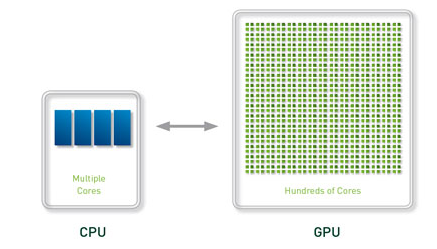
**Bộ lập lịch trình Thread GigaThread**

### Second Generation Parallel Thread Execution ISA

### Memory Subsystem Innovations

### GigaThread Thread Scheduler

# GPU và CPU



# Các thuật toán song song áp dụng trên GPU