LẬP TRÌNH SONG SONG VỚI GPU

1. Giới thiệu về lập trình song song và CUDA

1.1 Lập trình/Tính toán song song là gì? What is Parallel Programming/Computing?

1.1.1 What is Parallel Programming/Computing?

1.1.1.1 Serial Programming/Computing

Trước khi đến với khái niệm về Parallel Programming/Computing, chúng ta hãy xem lại về Serial Programming/Computing (Lập trình/Tính toán tuần tự)

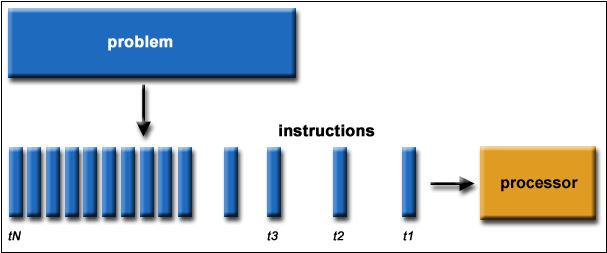
Traditionally, software has been written for **serial computation**:

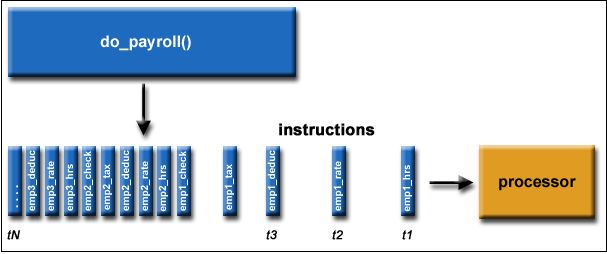
+ A problem is broken into a discrete series of instructions

+ Instructions are executed sequentially one after another

+ Executed on a single processor

+ Only one instruction may execute at any moment in time





+ 1 vấn đề lớn sẽ được chia làm chuỗi những câu lệnh thực thi.

+ Những câu lệnh thực thi được thực hiện tuần từ, lệnh này đến lệnh khác.

+ Quá trình thực thi trên 1 bộ xử lý đơn.

+ Tại mỗi thời điểm chỉ có 1 instruction được thực hiện.

1.1.1.2 Parallel Programming/Computing

In the simplest sense, parallel computing is the simultaneous use of multiple compute resources to solve a computational problem.

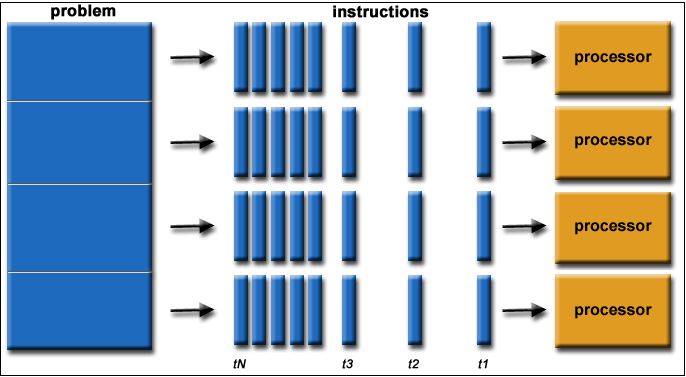
Hiểu đơn giản, tính toán song song là việc sử dụng đồng thời nhiều tài nguyên tính toán để giải quyết một vấn đề.

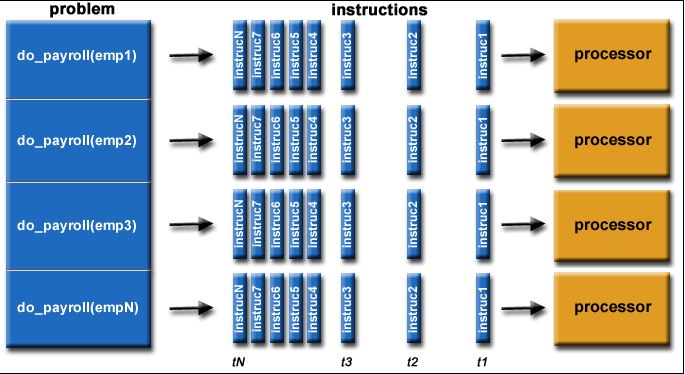
+ A problem is broken into discrete parts that can be solved concurrently

+ Each part is further broken down to a series of instructions

+ Instructions from each part execute simultaneously on different processors

+ An overall control/coordination mechanism is employed





+ Một vấn đề được chia thành các phần rời rạc có thể được giải quyết đồng thời  
+ Mỗi phần lại được chia nhỏ thành một loạt các hướng dẫn  
+ Instructions(các câu lệnh) từ mỗi phần thực hiện đồng thời trên các bộ xử lý khác nhau  
+ Có một cơ chế để kiểm soát / điều phối tổng thể.

1.1.2. Parallel Programming Models

There are several parallel programming models in common use:

Shared Memory (without threads) Bộ nhớ dùng chung

**Threads**

Distributed Memory / Message Passing

Data Parallel

Hybrid

Single Program Multiple Data (SPMD)

Multiple Program Multiple Data (MPMD)

**Threads Model**

This programming model is a type of shared memory programming. Nó là 1 dạng chia sẻ bộ nhớ.

In the threads model of parallel programming, a single "heavy weight" (chính/quan trọng) process can have multiple "light weight"(nhỏ, phụ), concurrent (đồng thời) execution paths.

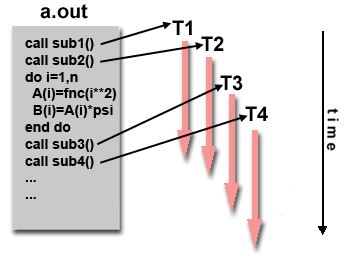
Explain:

- The main program a.out is scheduled to run by the native operating system. a.out loads and acquires all of the necessary system and user resources to run. This is the "heavy weight" process.

Chương trình chính a.out được lên lịch để chạy bởi hệ điều hành gốc. a.out load và giành lại tất cả các tài nguyên hệ thống và tài nguyên người dùng cần thiết để chạy. Đây là quá trình "chính/quan trọng".

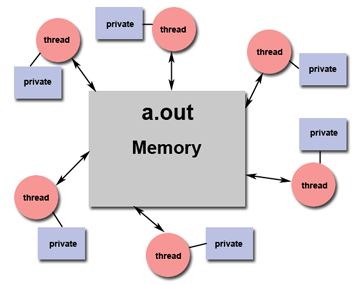
- a.out performs some serial work, and then creates a number of tasks (threads) that can be scheduled and run by the operating system concurrently.

a.out thực hiện một số công việc tuần tự, và sau đó tạo ra một số nhiệm vụ (luồng) có thể được lập lịch và chạy bởi hệ điều hành 1 cách đồng thời.



- Each thread has local data, but also, shares the entire resources of a.out. This saves the overhead associated with replicating a program's resources for each thread ("light weight"). Each thread also benefits from a global memory view because it shares the memory space of a.out.

Mỗi luồng có dữ liệu local, nhưng cũng chia sẻ toàn bộ tài nguyên của a.out. Điều này tiết kiệm chi phí liên quan đến việc sao chép tài nguyên của một chương trình cho mỗi luồng ("trọng lượng nhẹ"). Mỗi luồng cũng được hưởng lợi từ chế độ xem bộ nhớ chung vì nó chia sẻ không gian bộ nhớ của a.out.



A thread's work may best be described as a subroutine within the main program. Any thread can execute any subroutine at the same time as other threads.

Một công việc của thread có thể được mô tả như là một chương trình con trong chương trình chính. Bất kỳ luồng nào cũng có thể thực thi bất kỳ chương trình con nào cùng lúc với các luồng khác.

Threads communicate with each other through global memory (updating address locations). This requires synchronization constructs to ensure that more than one thread is not updating the same global address at any time.

Các luồng giao tiếp với nhau thông qua bộ nhớ toàn cục (cập nhật vị trí địa chỉ). Điều này đòi hỏi các cấu trúc đồng bộ hóa để đảm bảo rằng không có nhiều hơn một luồng cập nhật cùng một địa chỉ toàn cục bất cứ lúc nào.

Threads can come and go, but a.out remains present to provide the necessary shared resources until the application has completed.

Các threads có thể được gọi hoặc kết thúc, nhưng a.out vẫn tồn tại để cung cấp các tài nguyên được chia sẻ cần thiết cho đến khi ứng dụng hoàn tất.

1.2 GPU là gì?

1.2.1 CPU? Central Processing Unit

Trước khi tìm hiểu GPU chúng ta hãy xem lại về CPU.

The central processing unit (CPU) is the unit which performs most of the processing inside a computer. To control instructions and data flow to and from other parts of the computer, the CPU relies heavily on a chipset, which is a group of microchips located on the motherboard.

Bộ xử lý trung tâm (CPU) là đơn vị thực hiện hầu hết quá trình xử lý bên trong máy tính. Để kiểm soát instruction và data flow đến và đi từ các phần khác của máy tính, CPU dựa chủ yếu vào chipset, là một nhóm vi mạch nằm trên bo mạch chủ.

The CPU has two components:

Control Unit: extracts instructions from memory and decodes and executes them

Trích xuất các instructions từ bộ nhớ, giải mã và thực thi.

Arithmetic Logic Unit (ALU): handles arithmetic and logical operations

For 30 years, one of the important methods for the improving the performance of consumer computing devices has been to increase the speed at which the processor’s clock operated.

Trong vòng 30 năm từ 1980s đến khoảng 2004, thì phương thức tối ưu để tăng tốc độ máy tính là việc tăng tần số. Vì như ta biết “thời gian chạy 1 chương trình bằng tổng số câu lệnh nhân với thời giant rung bình của 1 câu lệnh”. => nguyên nhân chính là ở thời điểm này hầu hết PC vẫn là single processor. Điểm yếu của việc tăng tần số là sẽ làm tăng năng lượng.

In 2005, CPU manufacturers began offering processors with two computing cores instead of one.

Over the following years, they followed this development with the release of three-, four-, six-, and eight-core central processor units. Sometimes referred to as the multicore revolution, this trend has marked a huge shift in the evolution of the consumer computing market.

Năm 2005, máy tính 2 lõi ra đời, và nhiều năm sau đó lần lượt 3,4,6,8 lõi được sản xuất. Đánh dấu 1 bước đột phá lớn của máy tính.

Leading CPU manufacturers have already announced plans for 12- and 16-core CPUs, further confiming that parallel computing has arrived for good.

1.2.2 GPU? Graphics processing unit.

1.2.3 CPU vs GPU? Hay GPU hơn gì CPU?

1.3 Lập trình song song với GPU là như thế nào?

1.4 CUDA là gì?

1.5 Mối liên quan giữa lập trình song song và CUDA

2. Tổ chức thread trong CUDA

Refer:

<https://techblog.vn/xu-li-song-song>

<https://computing.llnl.gov/tutorials/parallel_comp/>