Parallel Programming Principle and Practice

Lecture 4-1 —Computer cluster/

multi-computers with distributed memory

Outline

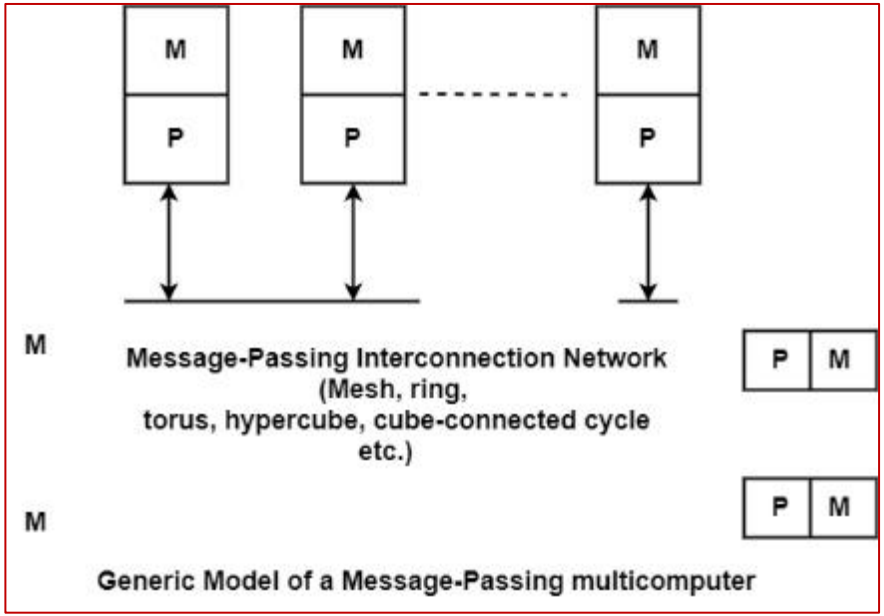
- Generic distributed-memory multi-computer system architecture
- Interconnection network for distributed-memory multi-computers
- Real modern distributed-memory multi-computer system architecture

Computer cluster/multi-computers with distributed memory

Generic distributed-memory multi-computer system architecture

Generic distributed-memory multi-computer system architecture

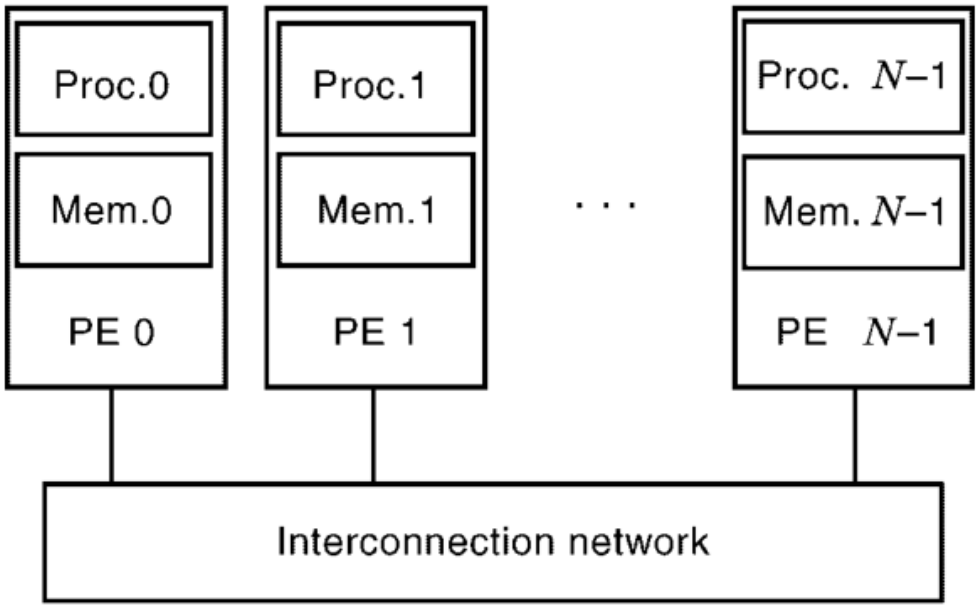
- Inclunodesdes multiple computers known as, related by a message-passing network
  - Each node is an independent computer including a processor, local memory, and sometimes connected disks or I/O peripherals
  - The message-passing network supports point-to-point static connections among the nodes
  - All local memories are private and are applicable only by local processors



Generic Model of a Message-Passing multicomputer

Generic distributed-memory multi-computer system architecture

- A processor and a memory module form a processor–memory pair that is called processing element (PE)
- All N PEs are interconnected via an interconnection network
- The distributed PE memory modules act as a single shared address space
  - processor can access any memory cell



Generic distributed-memory multi-computer system architecture

Multicomputer Generations

- Each multicomputer uses routers (路由) and channels (通道) in its interconnection network, and heterogeneous systems may involve mixed node types and uniform data representation and communication protocols
  - First generation: hypercube architecture, software-controlled message switching, processor boards
  - Second generation: mesh-connected architecture, hardware message switching, software for medium-grain distributed computing
  - Third generation: fine-grained distributed computing (each VLSI chip containing the processor and communication resources)

Computer cluster/multi-computers with distributed memory

Interconnection network for distributed-memory multi-computers

Interconnection network for distributed-memory multi-computers

- An interconnection network is a system of links that connects one or more devices to each other for the purpose of inter-device communication
- interconnection network is used primarily to connect processors to processors, or to allow multiple processors to access one or more shared memory modules
- The way that these entities are connected to each other has a significant effect on the cost, applicability, scalability, reliability, and performance of a parallel computer
- In general, the entities that are connected to each other, whether they are processors or memories, will be called nodes

Interconnection network for distributed-memory multi-computers

- An interconnection network may be classified as shared or switched
  - A shared network can have at most one message on it at any time
  - A switched network allows point-to-point messages among pairs of nodes and therefore supports the transfer of multiple concurrent messages

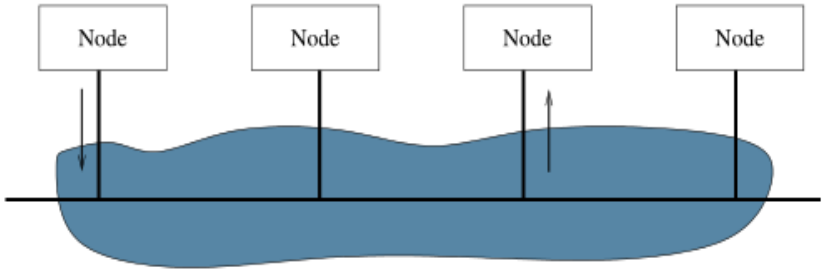


Figure 2.11: A shared network connecting 4 nodes.

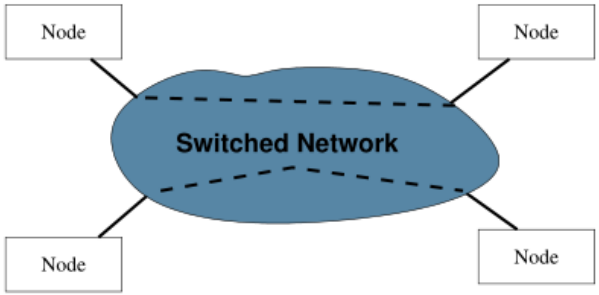
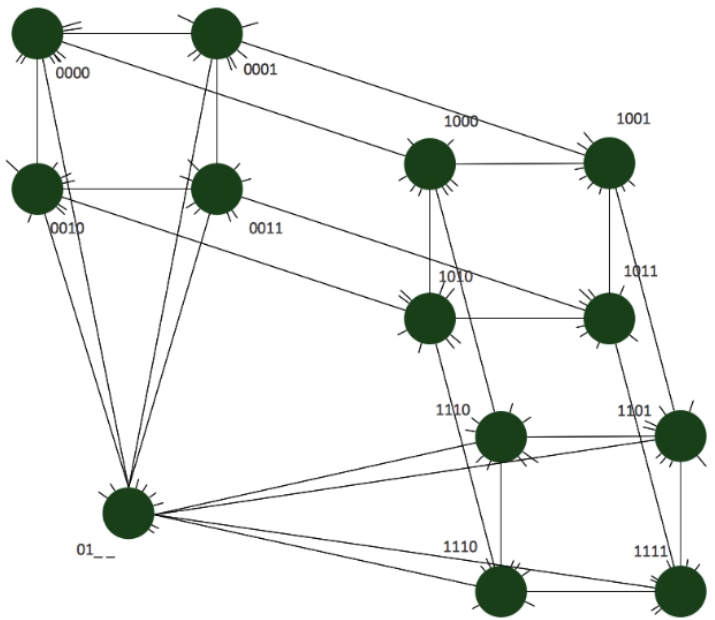


Figure 2.12: A switched network connecting 4 nodes.

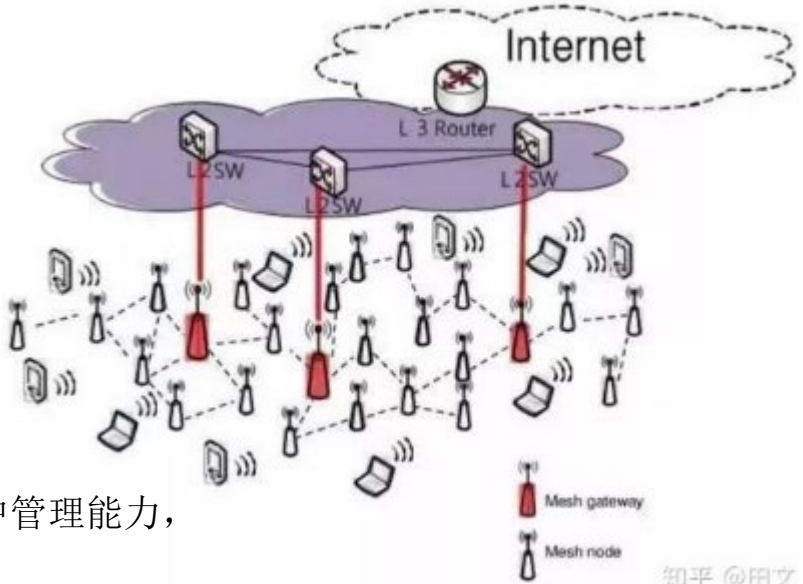
Example : hypercube architecture（超立方体网络拓扑）

- **超立方体网络拓扑**是一种环形网络，用于连接多个处理器和内存模块
- 由2m个节点组成，这些节点构成正方形的顶点，形成一个互连网络连接
- 环面是一个具有n维网格网络的拓扑结构，节点之间呈环形连接
- 本质上是一个每个维度有两个节点的多维网格网络。由于拓扑之间的相似性，通常将这些拓扑划分为k-ary *d*-dimensional网格拓扑族，其中d表示维数，k表示每个维中的节点数



Example : mesh-connected architecture

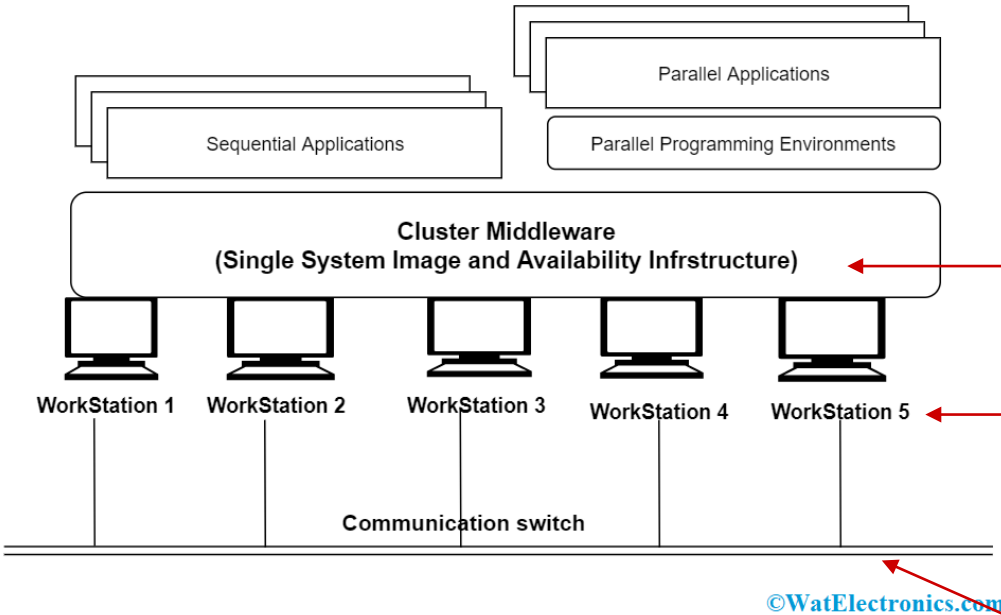
- 网络中所有的节点都互相连接，并且每一个节点至少连接其他两个节点，所有的节点之间形成一个整体的网络
  - 有线时代，由于网线的存在，要实现Mesh布局的网络显得非常困难
  - 在无线时代，由于脱离了网线的羁绊，再通过各种全新技术的应用，无线Mesh网络的布局就显得相对容易了
- 网络拓扑结构来看，无线Mesh网络具有如下特点
  - 结构灵活，方便部署
  - 无线Mesh网中每个节点都是AP节点，具备自动配置和集中管理能力，简化网络的管理维护
  - 无线Mesh网络拥有至少一条备用路径，因此在数据传输的可靠性上非常高
  - 无线Mesh网络大幅提升信号覆盖范围
  - 任意节点之间互相连通，将传统WLAN中的无线“热点”扩展为真正大面积覆盖的无线“热区”



Computer cluster/multi-computers with distributed memory

Real modern distributed-memory multi-computer system architecture

Cluster Computer Architecture



- A kind of parallel/distributed processing network
- Designed with an array of interconnected individual computers
- Collectively as a single standalone system.
- A node – Either a single or a multiprocessor network having memory, input and output functions and an operating system
- Two or more nodes are connected on a single line or every node might be connected individually through a LAN connection

Example Clusters : Berkeley NOW

- 100 Sun UltraSparcs (处理器)
  - 200 disks
- Myrinet SAN (网络)
  - 160 MB/s
- Fast comm. (通讯协议)
  - AM, MPI, ...
- Ether/ATM switched external net (以太/ATM交换外网)
- Global OS
- Self Config

Parallel Programming Principle and Practice

Lecture 4-2 —Accelerator (加速器) / GPU

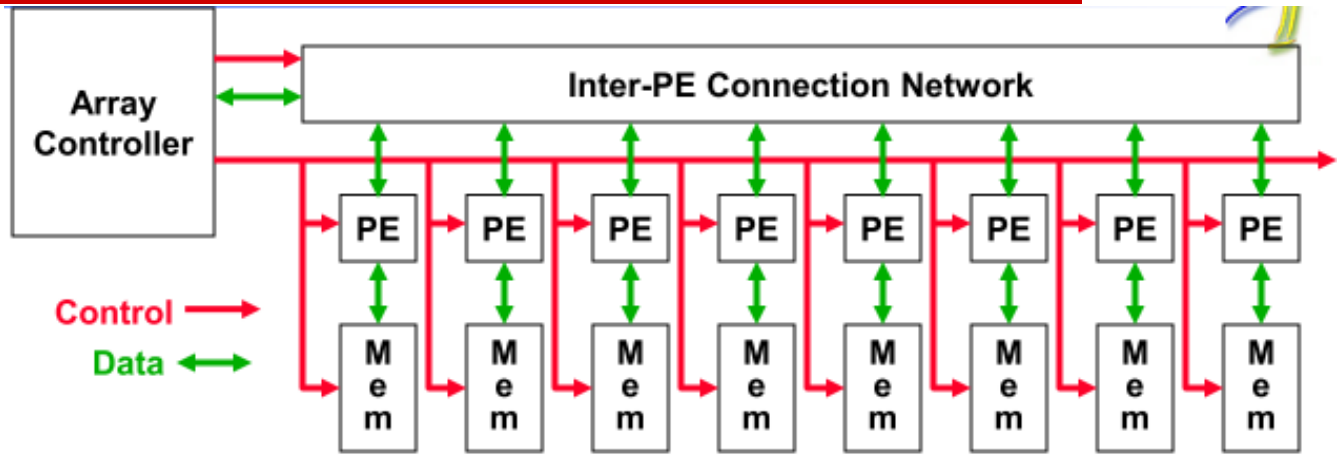
Outline

- SIMD system architecture
- Real modern GPU architecture

Accelerator / GPU

SIMD system architecture

SIMD Architecture

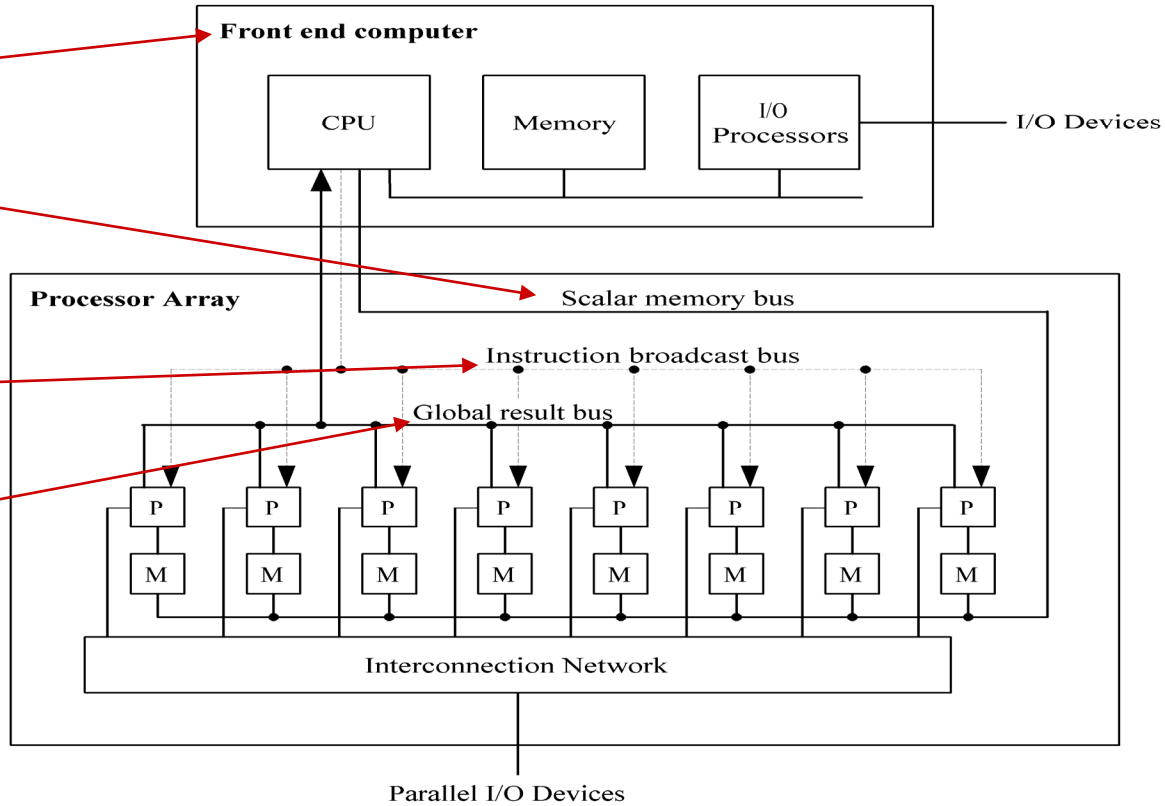


Array Controller:
阵列控制器
PE： 处理单元

- GPU (Graphics Processing Units) have SIMD properties
  - Central controller broadcasts instructions to multiple Processing Elements (PEs)
    - ✓ Only requires one controller for whole array
    - ✓ Only requires storage for one copy of program
    - ✓ All computations are fully synchronized

Possible Architecture for a Generic SIMD

- Front end computer（前端计算机）作为阵列控制器
  - 前端计算机通过存储总线，与处理单元的局部存储器进行数据交换
  - 通过指令广播总线将指令广播到每一个处理单元上
  - 处理结果通过全局结果总线返回前端机



SIMD Architecture

- SIMD计算机的操作模型可用五元组表示

$$M=(N，C，I，M，R)$$

式中：

（1）N为机器的处理单元（PE）数

（2）C为由控制部件（CU）直接执行的指令集，包括标量和程序流控制指令

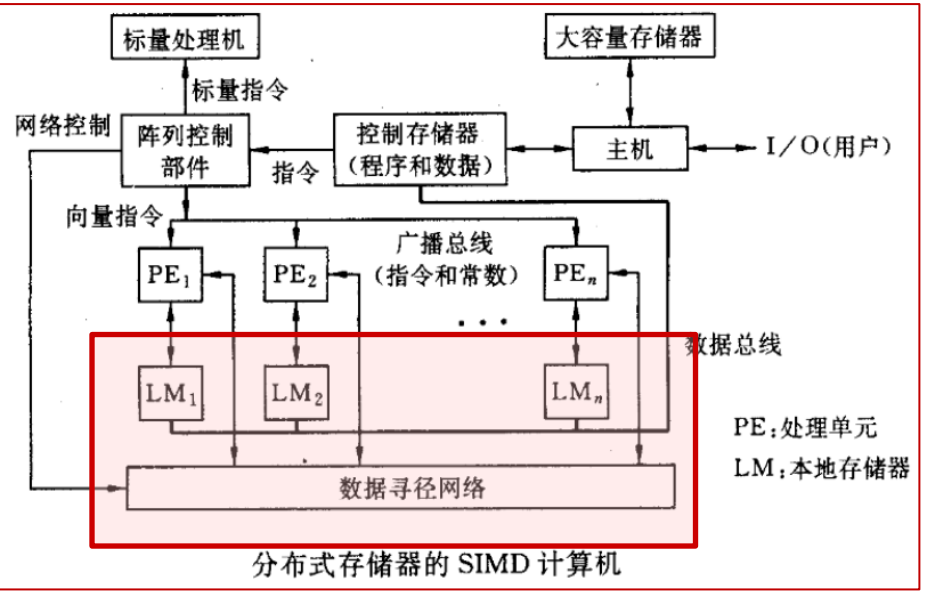
（3）I为由CU广播至所有PE进行并行执行的指令集，它包括算术运算、逻辑运算、数据寻径、屏蔽以及其他由每个活动的PE对它的数据所执行的局部操作

（4）M为屏蔽方案集，其中每种屏蔽将PE集划分为允许操作和禁止操作两种子集

（5）R是数据寻径功能集，说明互连网络中PE间通信所需要的各种设置模式

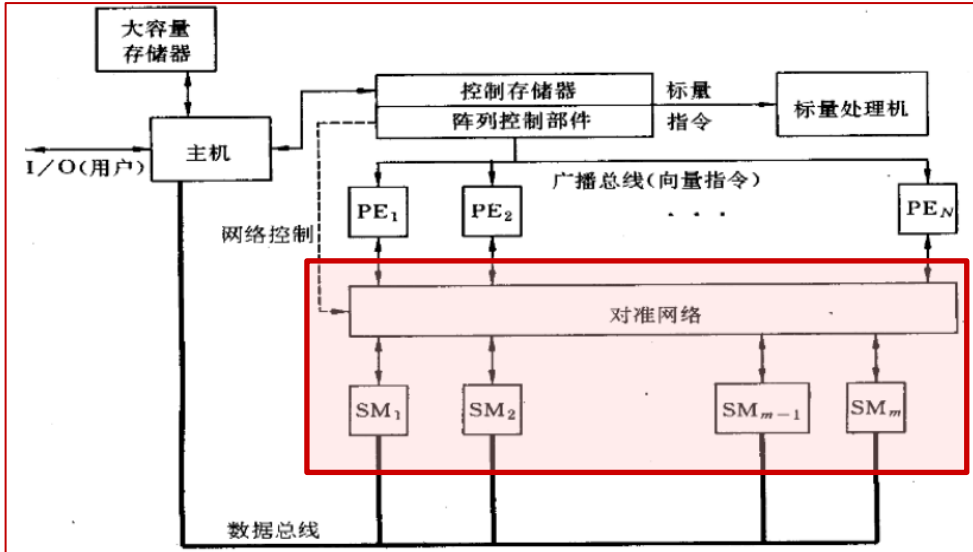
SIMD Architecture

- 分布存储器结构



分布式存储器的 SIMD 计算机

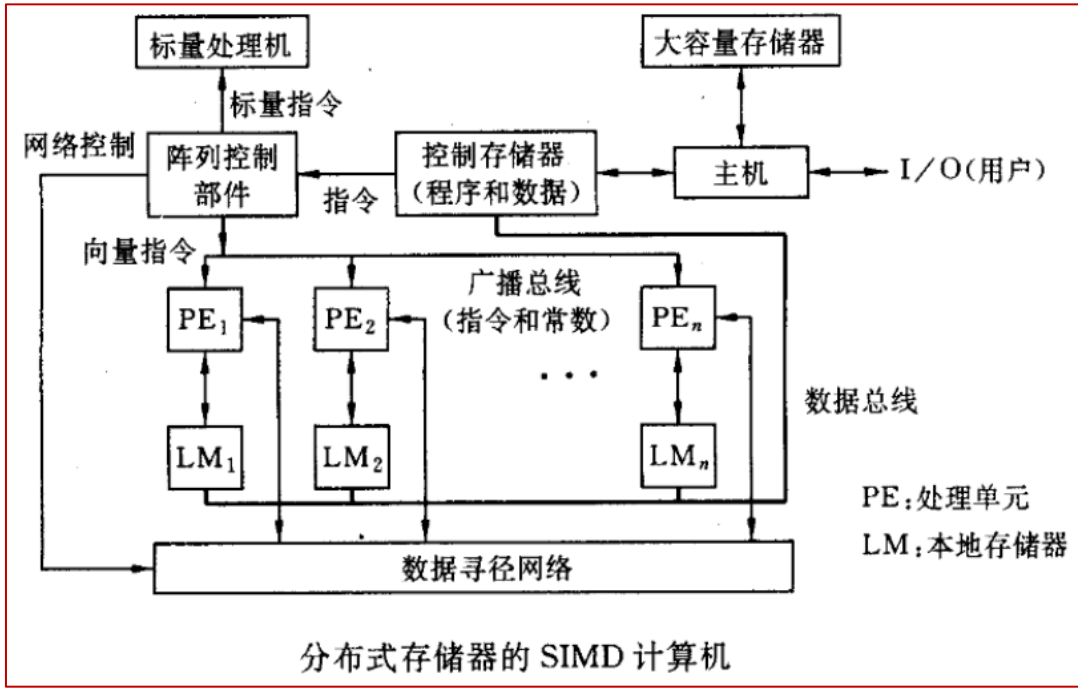
- 共享存储器结构



共享存储器的 SIMD 计算机

SIMD Architecture

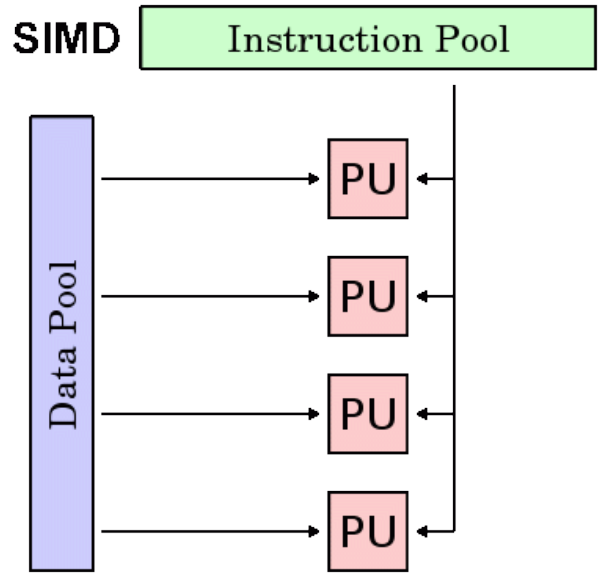
- 一般工作原理
  - 程序和数据存放在控制存储器中
  - 控制器(阵列控制器）从控制存储器取出指令进行译码
  - 如果是标量指令，则送标量处理机执行
  - 如果是向量指令，则将指令广播到每一个PE上执行，并将划分后的数据通过数据总线分布到所有PE的局部存储器LM中
  - 如果是通讯指令，则控制器控制数据寻径网络进行PE间的数据交换
  - 所有的PE同一时钟周期执行同一条指令



分布式存储器的 SIMD 计算机

SIMD: Single Instruction, Multiple Data

- Single instruction operates on multiple data elements
  - Array processor
  - Vector processor
- A type of parallel computer
  - Best suited for specialized problems characterized by a high degree of regularity, such as graphics/image processing
  - Synchronous (同步) (lockstep) and deterministic execution
  - Two varieties: Processor Arrays and Vector Pipelines



Data Parallelism (A strength for SIMD)

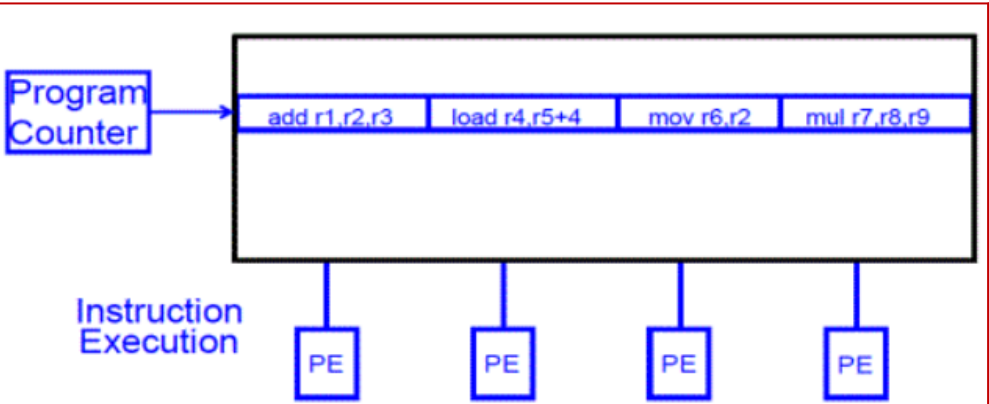
- All tasks (or processors) apply the same set of operations to different data
- Example:

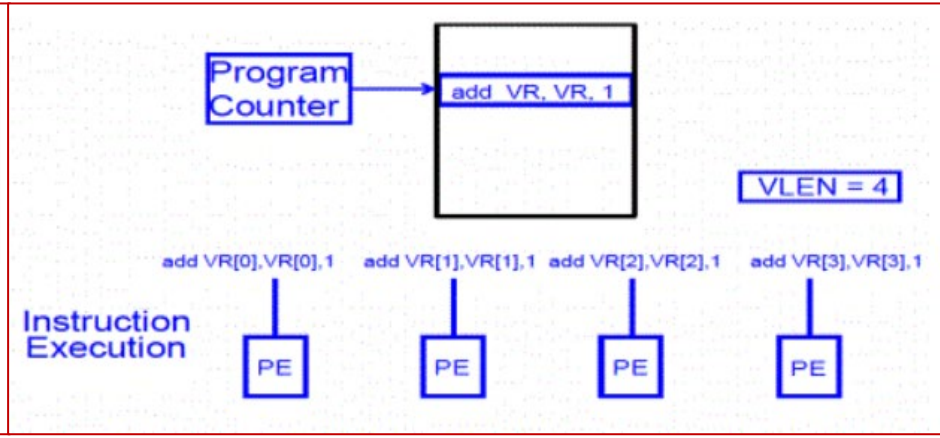
```
for i ← 0 to 99 do
   a[i] ← b[i] + c[i]
endfor
```

- Accomplished on SIMDs by having all active processors execute the operations synchronously

VLIW VS Array processor

- VLIW
  - 多个独立的操作通过编译器包装在一起
- 阵列处理器
  - 多个（不同的）数据元素的单操作





Data Parallelism

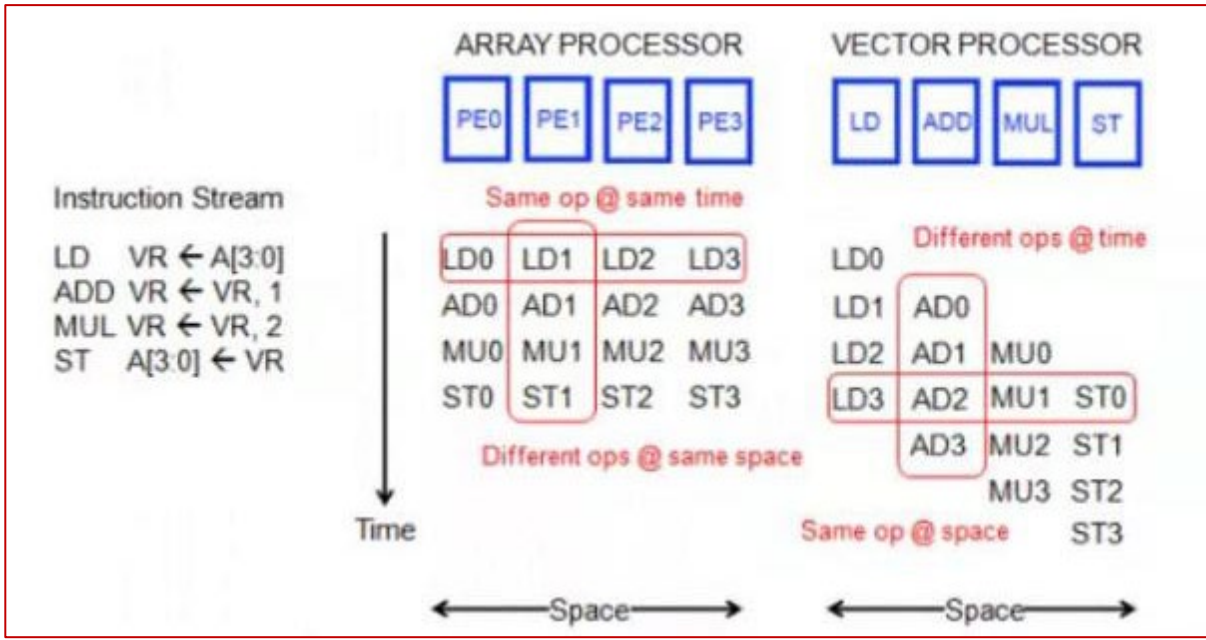
- Concurrency arises from performing the same operations on different pieces of data
  - Single instruction multiple data (SIMD)
  - E.g., dot product of two vectors
- Contrast with data flow
  - Concurrency arises from executing different operations in parallel (in a data driven manner)
- Contrast with thread ("control") parallelism
  - Concurrency arises from executing different threads of control in parallel
- SIMD exploits instruction-level parallelism
  - Multiple instructions concurrent: instructions happen to be the same

SIMD Processing

- Single instruction operates on multiple data elements
  - In time or in space
- Multiple processing elements
- Time-space duality
  - Array processor: Instruction operates on multiple data elements at the same time
  - Vector processor: Instruction operates on multiple data elements in consecutive time steps

SIMD Processing

- Array processor *VS* Vector processor
  - Array processor
    - ✓ Same op at same time
    - ✓ Different ops at same space
  - Vector processor
    - ✓ Different ops at time
    - ✓ Same op at space

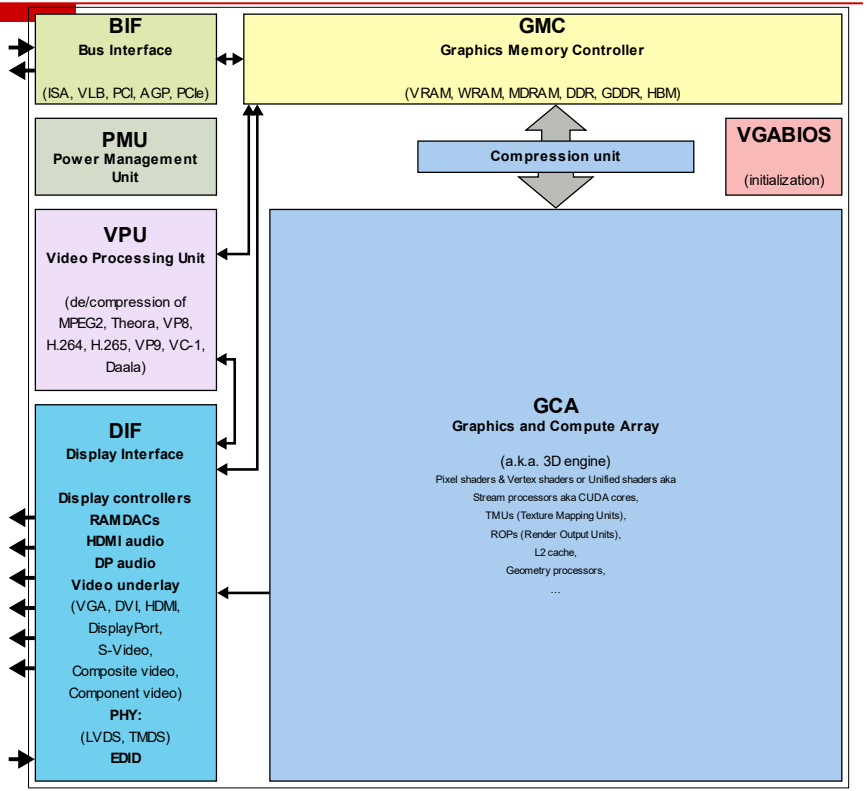


Accelerator / GPU

Real modern GPU architecture

Graphics processing unit (GPU)(图形处理单元)

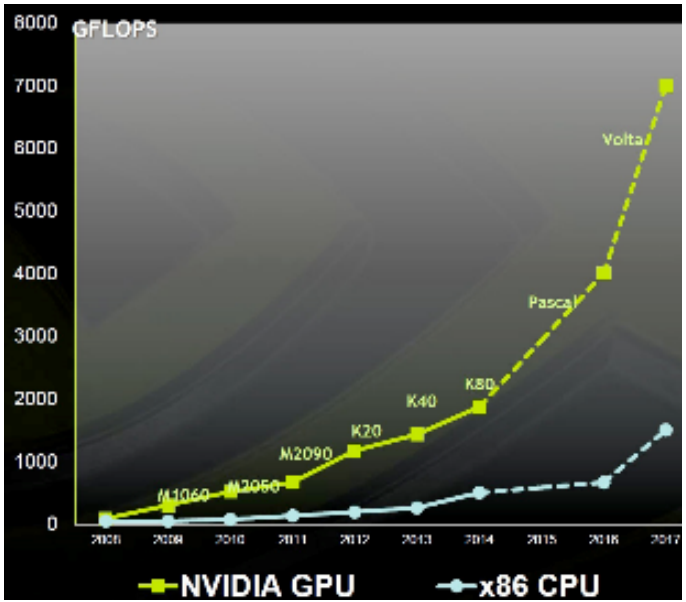
- A specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device
- Used in embedded systems, mobile phones, personal computers, workstations, and game consoles
- More efficient than general-purpose central processing units (CPUs) for algorithms that process large blocks of data in parallel



Components of a GPU

Graphics processing unit (GPU)(图形处理单元)

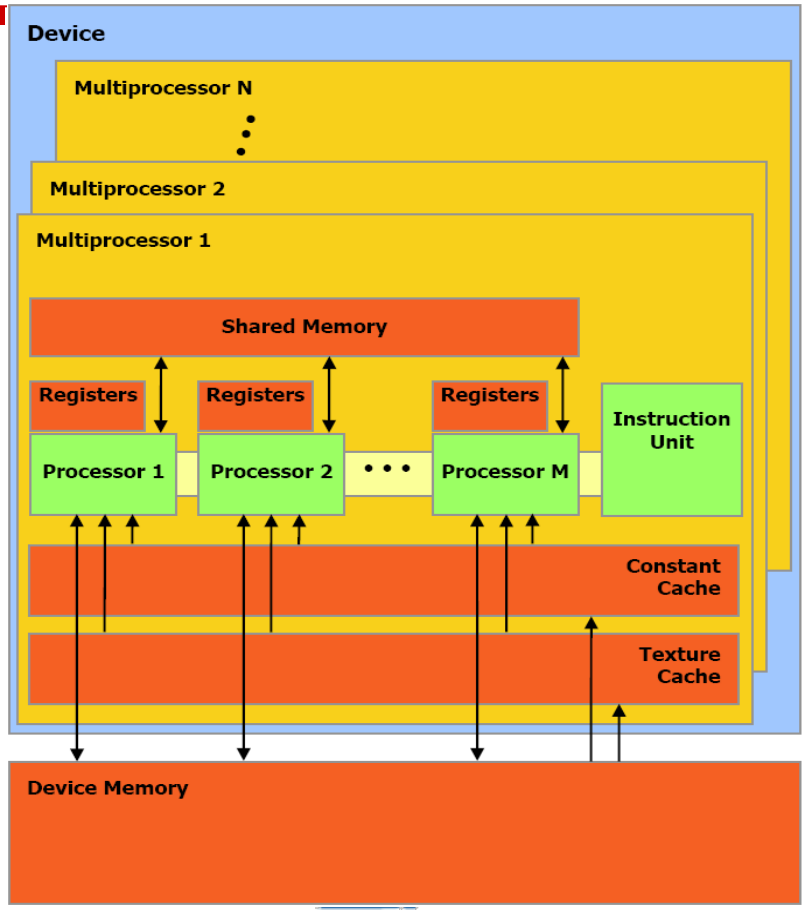
- Graphics Processing Unit (GPU)
  - Do the same numerical operation in parallel to many pixels stored in high-bandwidth graphic memory
  - The same approach fits well for many other parallel numerical problems, e.g. training a neural network



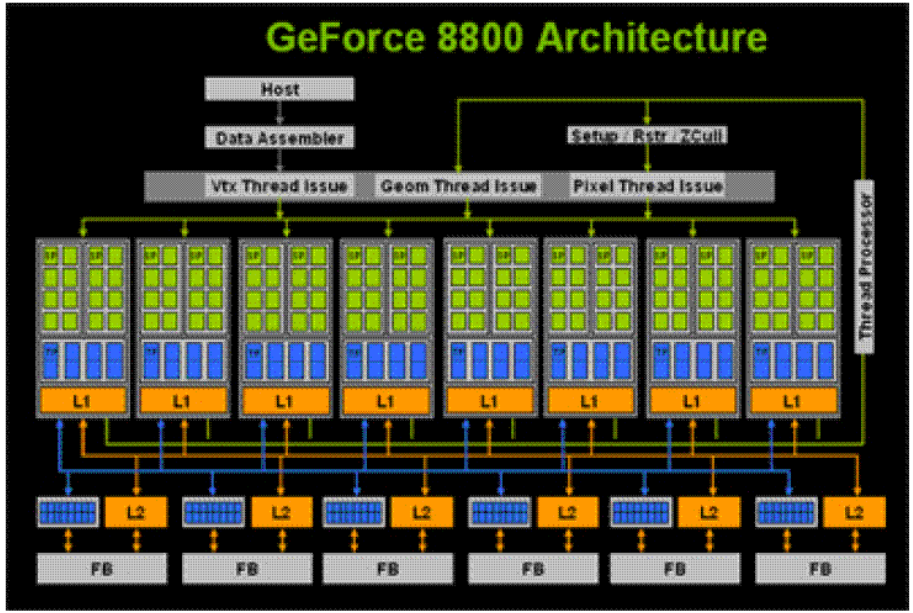
Peak Floating Point Operations Per Second (FLOPS)

Nvidia G80 GPU Architecture Overview

- 16 Multiprocessors Blocks
- Each MP Block has
  - 8 Streaming Processors (IEEE 754 spfp compliant)
  - 16K Shared Memory
  - 64K Constant Cache
  - 8K Texture Cache
- Each processor can access all of the memory at 86Gb/s, but with different latencies
  - Shared – 2 cycle latency
  - Device – 300 cycle latency



8800 GPU Architecture



Relation to 8800 GPU

- Key to the GeForce 8800 architecture is the use of numerous scalar stream processors (SPs)
- Stream processors are highly efficient computing engines that perform calculations on an input stream and produces an output stream that can be used by other stream processors
- Stream processors can be grouped in close proximity, and in large numbers, to provide immense parallel processing power

Parallel Programming Principle and Practice

*Lecture 4-3 —New development

Outline

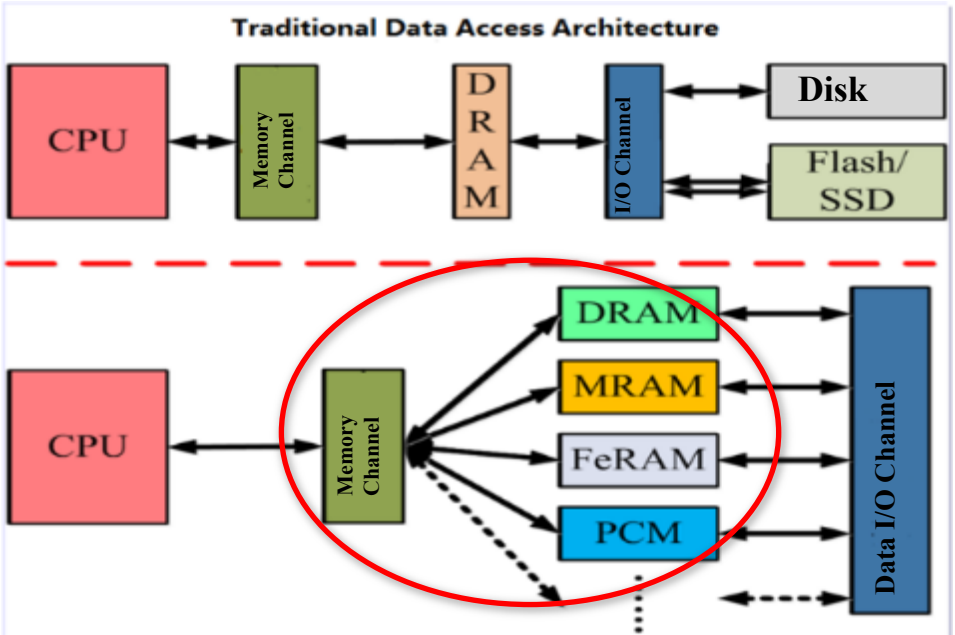
- In-memory computing
- Non-volatile memory (NVM) /Storage class memory (SCM)
- Tensor processing unit (TPU)

*New development

In-memory computing

In-memory computing(存内计算)

- Stores data in RAM rather than in databases hosted on disks.
- Eliminate the I/O and ACID transaction requirements of OLTP applications and exponentially speeds data access



Why in-memory computing?

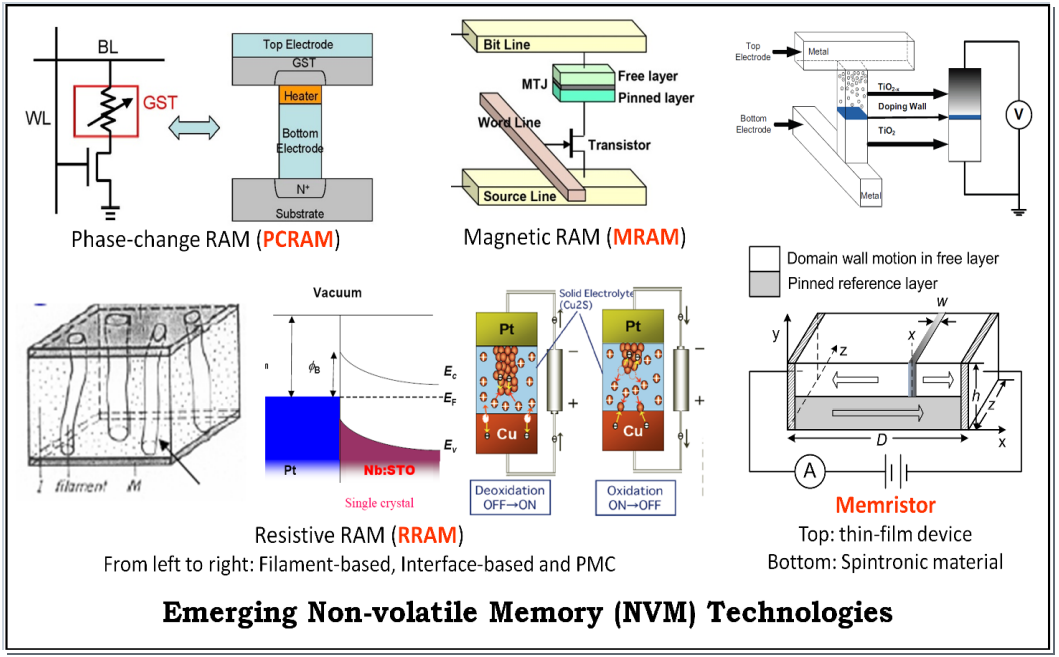
- Enables extremely fast processing
- Quickly analyze massive volumes of data in real time at very high speeds
- Increase performance

*New development

Non-volatile memory (NVM)

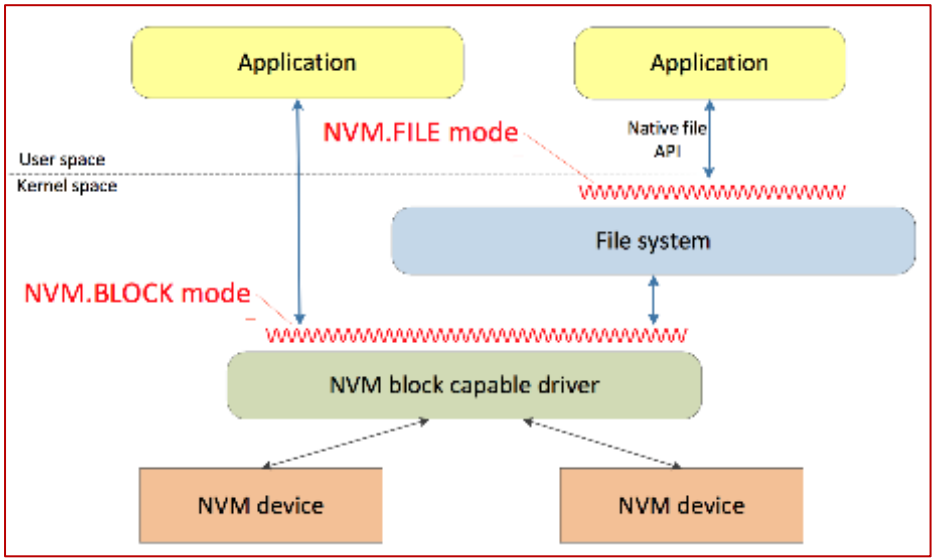
Non-volatile memory (NVM)(非易失性存储器)

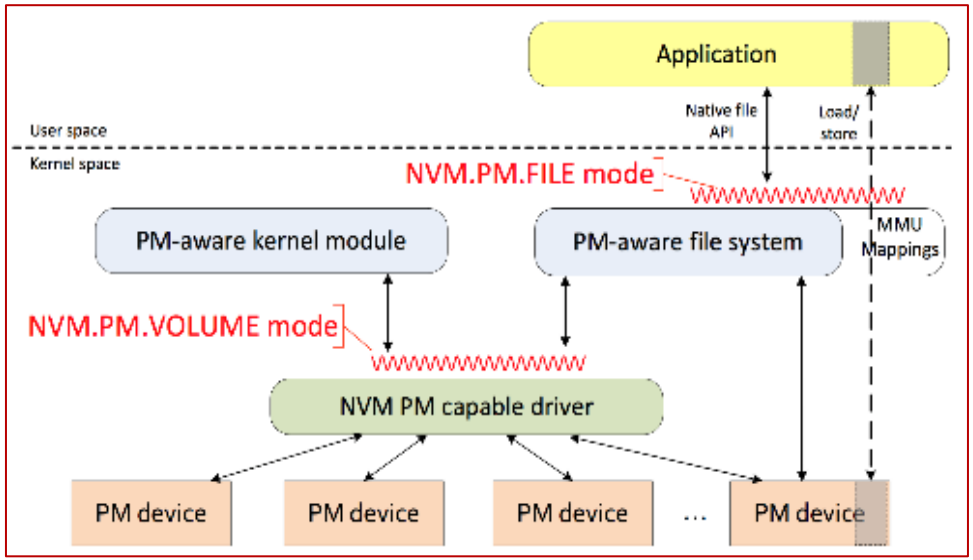
- A type of computer memory that can retain stored information even after power is removed
  - In contrast, volatile memory needs constant power in order to retain data
  - Store data in floating-gate memory cells consisting of floating-gate MOSFETs



Two NVM Programming Modes Overview

- Non-Volatile Memory (NVM) programming model in development





*New development

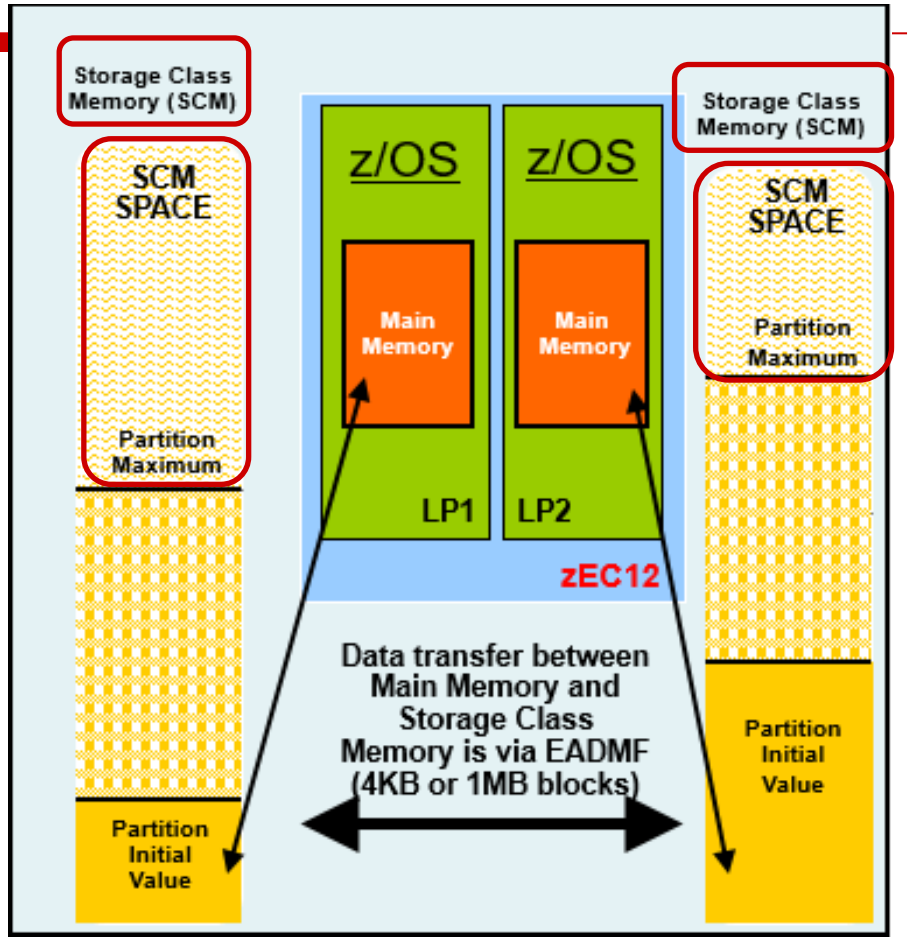
Storage class memory (SCM)

Storage class memory (SCM)(存储级内存)

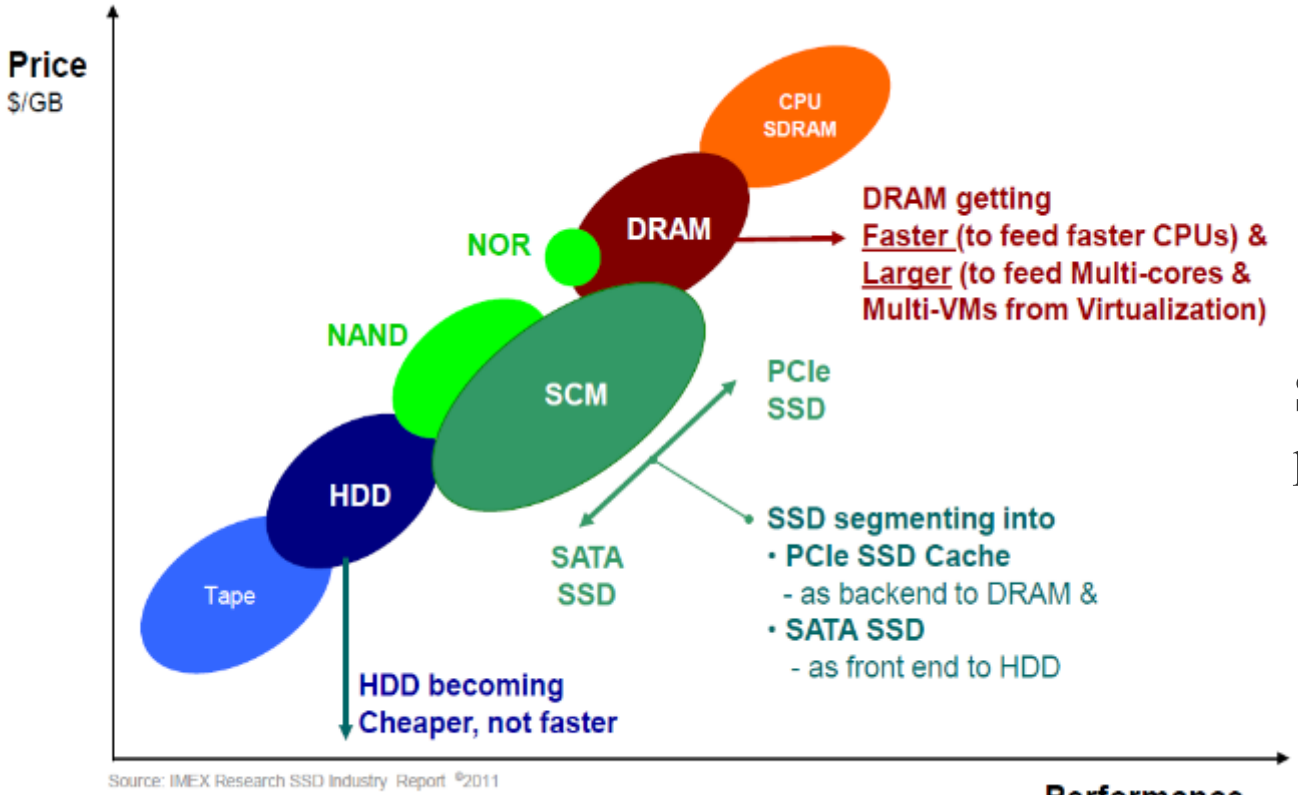
- A new memory and storage technology that offers features in both dynamic random-access memory (DRAM) and traditional flash storage
- Extend the performance advantage of DRAM to stateful persistent storage
- Enable the use of SCM in either memory or SSD storage use cases
- The advantages of SCM technology include
  - Faster performance and lower application latency than NVMe flash
  - Accelerate write-intensive online transaction processing (OLTP) workloads
  - Accelerate high-performance computing
  - Higher write durability than offered by NVMe SSDs

Storage class memory (SCM)(存储级内存)

- SCM's importance in data centers
  - High-performance storage
  - Persistent storage
  - Block-level data access
  - Less processing, more throughput



New Advances on Memory and Storage — Storage Class Memory



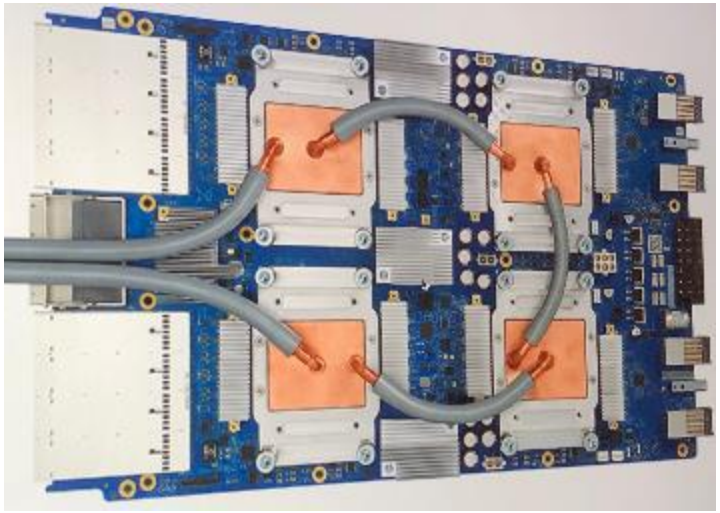
SCM is middling in both performance and price

*New development

Tensor processing unit (TPU)

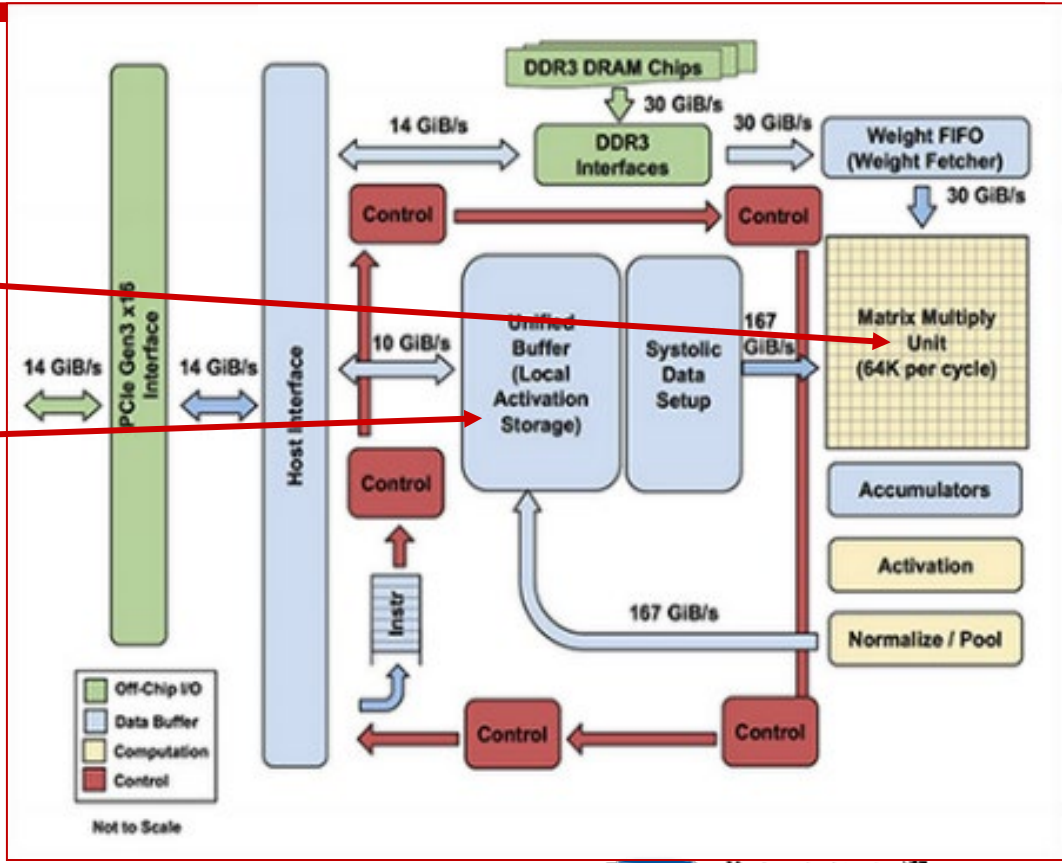
Tensor processing unit (TPU) (张量处理单元)

- Referred to as a TensorFlow processing unit—is a special-purpose accelerator for machine learning
- An AI accelerator application-specific integrated circuit (ASIC) developed by Google specifically for neural network machine learning
- Using for accelerating specific machine learning workloads using processing elements—small DSPs with local memory



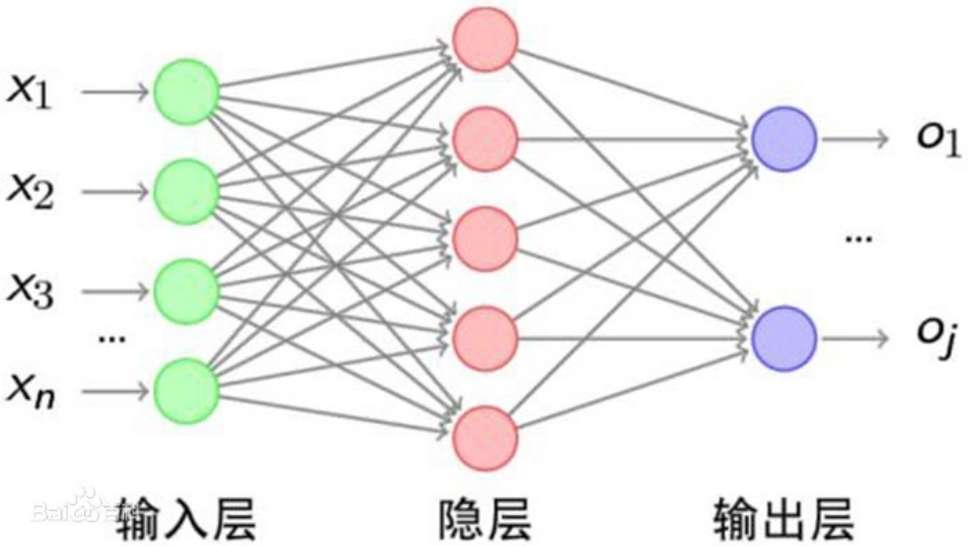
Tensor processing unit (TPU) (张量处理单元)

- Google's TPU has
  - a systolic matrix multiply unit, systolic arrays, the heart of TPU
  - a unified buffer (24-MB register file)
  - hardwired activation unit

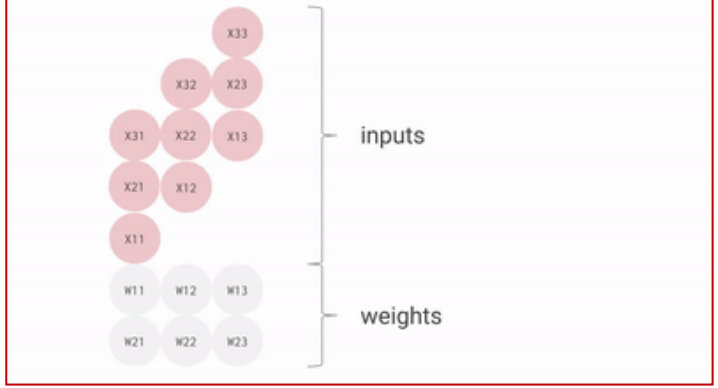


An Example Relationship between TPU computing and neural network

TPU calculates vectors to assist in linear regression



Multiplying
an input matrix by
a weight matrix



Conclusion

- **Computer cluster/multi-computers with distributed memory**
  - Generic distributed-memory multi-computer system architecture
  - Interconnection network for distributed-memory multi-computers
  - Real modern distributed-memory multi-computer system architecture
- **Accelerator/GPU**
  - SIMD system architecture
  - Real modern GPU architecture
- ***New development**
  - In-memory computing
  - Non-volatile memory (NVM) /Storage class memory (SCM)
  - Tensor processing unit (TPU)