Advanced
Institute for
Artificial
Intelligence

HPC

Advanced Institute for Artificial Intelligence – AI2

https://advancedinstitute.ai

Agenda

- Computação de Alto Desempenho (HPC)
- Arquitetura Computacional
- Vetorização
- GPU
- Bibliotecas para abstração de desempenho

Introdução a HPC

O que é Computação de alto Desempenho ou HPC (High Performance Computing)?

- É uma área de pesquisa que lida com desafios relacionados com execução de aplicações com alto custo computacional
  - Normalmente, tais aplicações são utilizadas com muita frequência, e qualquer melhoria de desempenho provocam impacto
- Avanço no desenvolvimento de arquiteturas computacionais com maior poder computacional
  - Uma arquitetura computacional mais moderna provê recursos para melhorar o desempenho, porém a utilização de tais recursos nem sempre é trivial

Introdução a HPC

O que é uma aplicação que demanda HPC?

- Restrições de tempo impactam no uso do software
  - Aplicação A demora algumas horas em um PC comum. Executada uma vez ao ano.
  - Aplicação B demora 10 minutos. Executada diversas vezes ao longo do dia
- Otimizar a aplicação A apresenta impacto muito alto, mesmo que seja reduzir em poucos segundos a execução
- É importante avaliar de modo mais amplo a otimização, impacto na operação, impacto para outras aplicações e impacto quanto ao uso do hardware

Introdução a HPC

Otimização de desempenho de aplicações

- Caracterização das demandas computacionais das aplicações, por exemplo, demanda por espaço de armazenamento, processamento, memória RAM, etc
- Segmentação da aplicação em parte menores para que sejam executadas simultaneamente em diferentes recursos de uma arquitetura (paralelismo)
- Desenvolvimento de camadas de software otimizadas para diversas e diferentes arquiteturas computacionais
  - Tais camadas abstraem a complexidade do trabalho de otimização e permitem uso eficiente da arquitetura

Introdução a HPC

Aplicações de Aprendizagem Profunda (Deep Learning) normalmente são computacionalmente intensivas

- Tarefas computacionalmente intensivas em Deep Learning:
  - Busca por hiperparâmetros
  - Treino do modelo com novos conjuntos de dados
  - Prototipação de modelos
- Quanto mais rápido uma aplicação de aprendizagem profunda é executada, ainda que com um ganho não tão expressivo, apresenta impacto alto no trabalho dos especialistas desse domínio

Modelo de desempenho em processadores atuais

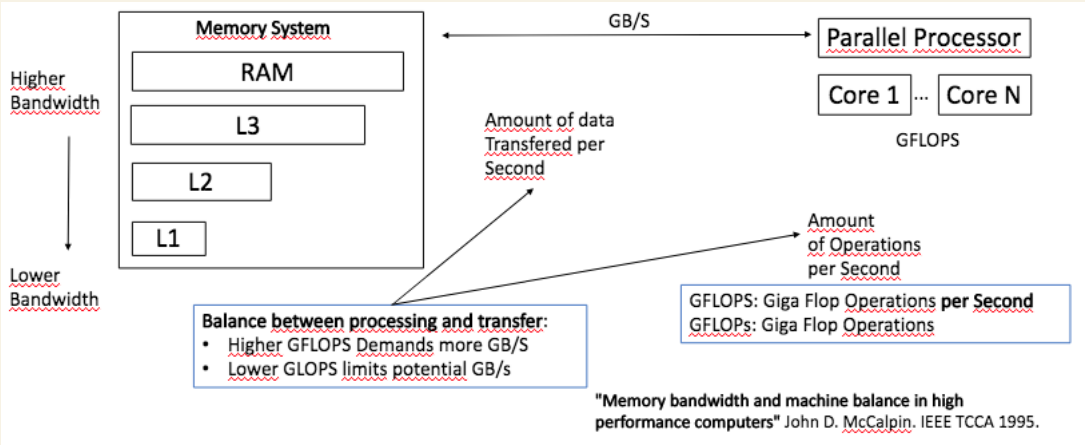
- Processador e sistema de memória trabalham de forma independente
- O desempenho da arquitetura computacional é medido de forma complementar
  - O desempenho do processador depende da eficiencia do sistema de memória entregar os dados para os registradores do processador
    - Se os dados são entregues de modo ineficiente, o processoador pode ficar ocioso em muitos momentos
  - O sistema de memória trabalha de modo eficiente se a vazão do processador é alta
    - Se o processador não executa as operações de modo eficiente, o sistema de memória pode ficar ocioso em muitos momentos

Introdução a HPC

Técnicas para otimização de Desempenho

- Processador
  - Vetorização permite executar uma mesma intrução para conjuntos diferentes de dados
- Sistema de memória
  - Múltiplos níveis de memória permitem antecipar o envio de dados para o processador, aumentando a eficiência da execução das instruções

Introdução a HPC

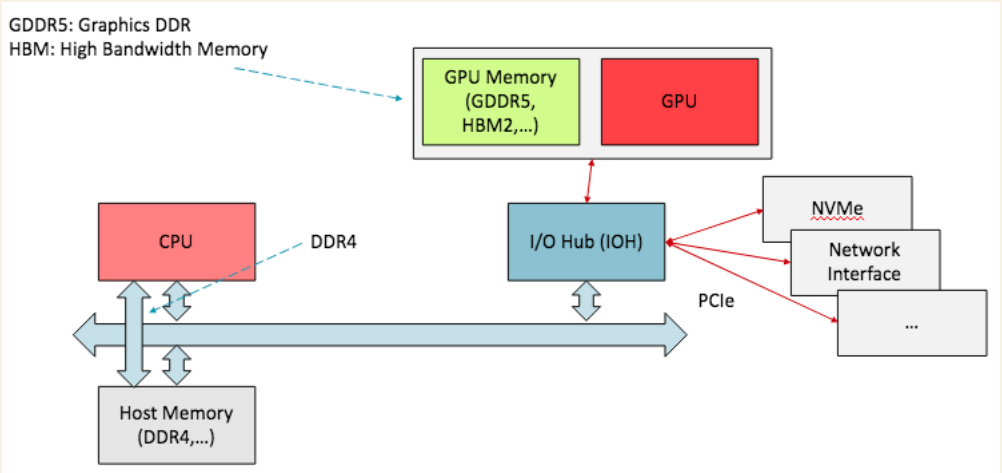


GPU

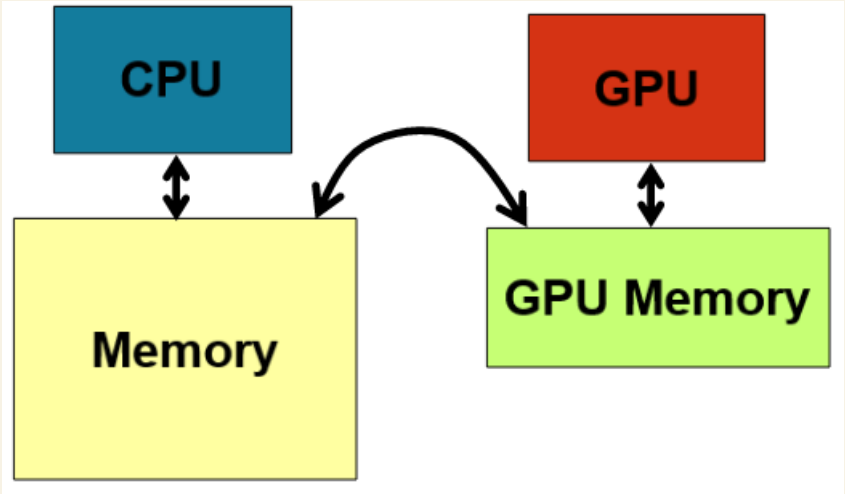
GPU

- É um multiprocessador paralelo / multithread otimizado para processamento de imagem.
  - O processamento gráfico é uma aplicação massivamente paralela
- GPGPU
  - Computação de uso geral usando GPU
- A GPU serve como um processador gráfico programável e como uma plataforma de computação paralela escalável.
  - Os sistemas podem combinar CPU + GPU para executar aplicativos

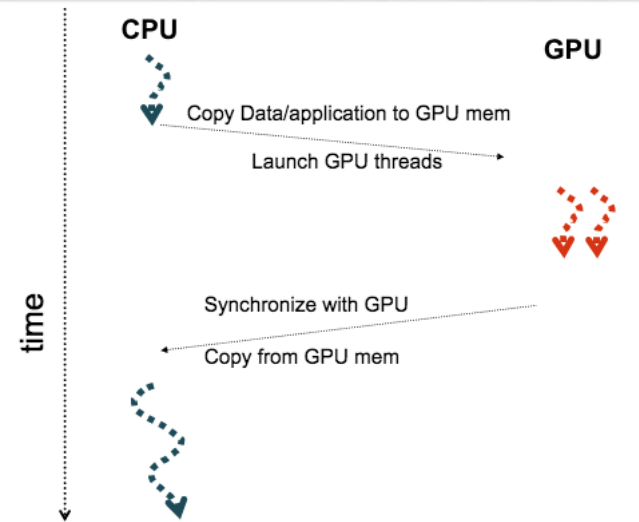
GPU


GDDR5: Graphics DDR
HBM: High Bandwidth Memory
GPU Memory (GDDR5, HBM2,...)
GPU
CPU
DDR4
I/O Hub (IOH)
NVMe
Network Interface
...
PCIe
Host Memory (DDR4,...)

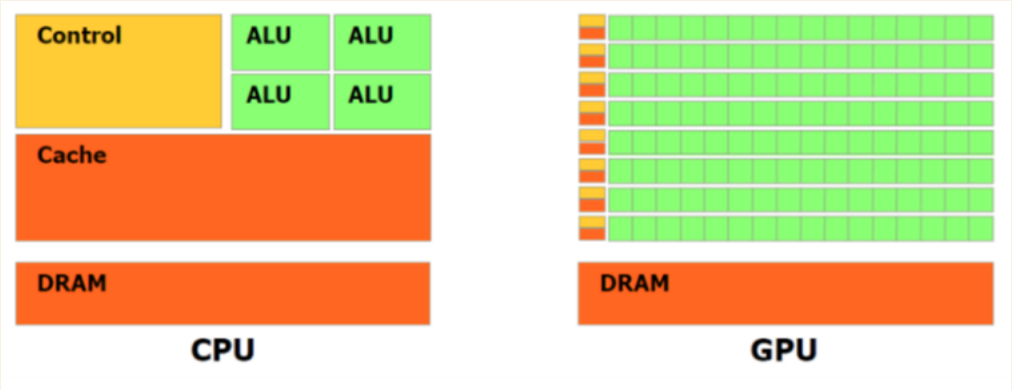
GPU



GPU


CPU
GPU
Copy Data/application to GPU mem
Launch GPU threads
Synchronize with GPU
Copy from GPU mem
time

GPU



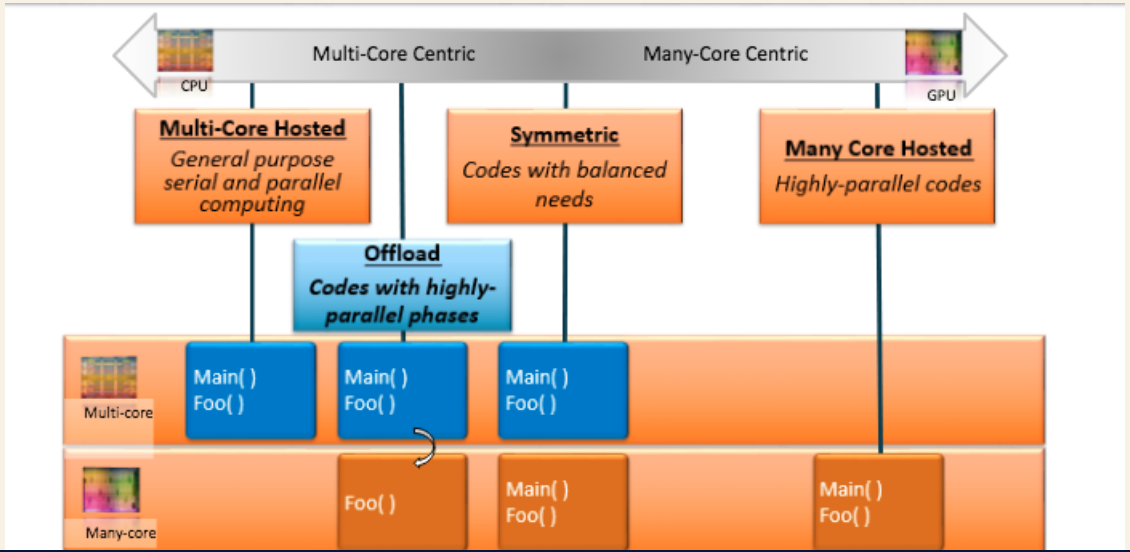
GPU

- lógica simplificada (sem execução fora de ordem, sem previsão de ramificação) significa que muito mais do chip é dedicado à computação de ponto flutuante
  - Núcleo da CPU Core x Núcleo da GPU
- Organizados como várias unidades, com cada unidade sendo efetivamente uma unidade vetorial, todos os núcleos fazendo a mesma coisa ao mesmo tempo
  - Kernel: uma rotina paralela para executar no hardware paralelo
- Maior largura de banda de memória que a CPU

GPU

- Objetivo não geral
  - Aplicações massivamente paralelas
    - Processamento gráfico
  - Aplicativos que exploram a localidade da memória
    - Cada unidade paralela realiza acesso ao seu próprio subconjunto de dados
    - Os algoritmos de paralelo de dados utilizam atributos da GPU
- Grandes matrizes de dados, taxa de transferência de streaming
- Cálculo de ponto flutuante de baixa latência (FP)

GPU


Multi-Core Centric
Many-Core Centric
CPU
GPU
Multi-Core Hosted
General purpose serial and parallel computing
Symmetric
Codes with balanced needs
Many Core Hosted
Highly-parallel codes
Offload
Codes with highly-parallel phases
Multi-core
Many-core
Main( )
Foo( )
Main( )
Foo( )
Main( )
Foo( )
Foo( )
Main( )
Foo( )
Main( )
Foo( )

GPU

- Como usar os recursos da GPU
  - Bibliotecas
    - Cublas
    - Tensorflow
  - Extensões da linguagem (diretivas)
    - OpenMP, OpenACC, OpenCL
  - fácil de otimizar código
  - Flexibilidade mínima
- Linguagem de Programação
  - API Cuda
  - Flexibilidade máxima
  - Acesso de baixo nível

Basic Linear Algebra Subroutines (BLAS)

- Escrito por especialistas para prover suporte a operações de diversas de algebra linear
  - multiplicação de matrizes
  - Operações com matrizes esparsas
- Com otimizações para diferentes arquiteturas computacionais
  - Math Kernel Library (Intel)
  - CuBlas (NVidia)
  - ATLAS ou LAPACK (Projetos abertos)

Introdução a HPC

- Muitos softwares são baseados em operações com algebra linear
- Tais bibliotecas facilitam obter o melhor desempenho de modo automatico
  - Explora os recursos da aquitetura de modo eficiente
  - Permite paralelizar em diversos processoadores e/ou coprocessadores
- Outras bibliotecas de mais alto nível (como Tensorflow ou Theano), facilitam o uso de bibliotecas Blas.
  - Esse grande nível de abstração possui impacto positivo no desempenho