

# Introduzione al Many/Multi-core Computing

Sistemi Operativi e reti

Flavio Vella

Università degli Studi di Perugia - Corso di Laurea Magistrale in Informatica

6 giugno 2011



# Outline del corso

- ▶ Introduzione
  - ▶ Definizioni
  - ▶ Motivazioni
  - ▶ Storia
- ▶ Architettura
- ▶ Framework
- ▶ Algoritmica

# Parte I

## Introduzione

# Definizioni

*GPU computing or GPGPU is the use of a GPU (graphics processing unit) to do general purpose scientific and engineering computing. [Nvidia].*

*Heterogeneous Computing is the use transparently of **all computational devices** to do general purpose scientific and engineering computing.*

## Motivazioni in teoria

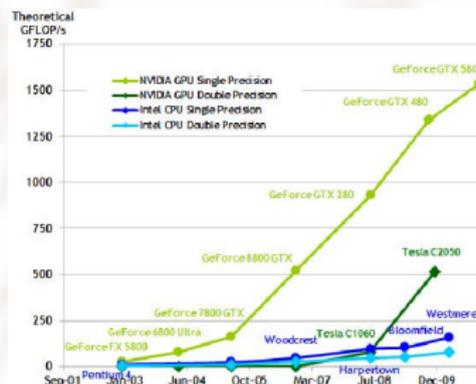
- ▶ Legge empirica di Moore. *"Le prestazioni dei processori, e il numero di transistor ad esso relativo, raddoppiano ogni 18 mesi..."*.
- ▶ Raggiungimento dei limiti fisici nella costruzione dei transistor  $10nm$  nel 2017.

**Necessità di parallelizzare per incrementare le performance!!!**

# Motivazioni in pratica

- ▶ Le GPU sono dispositivi paralleli a basso costo già disponibili.
- ▶ Performance.
- ▶ L'architettura SIMD si adatta bene a molte tipologie di problemi.

# Motivazioni in pratica



Tipologia	Modello	GFLOPS	Anno di produzione
CPU	Opteron 6 series	120	2011-Q1
GPU	Nvidia GTX 580	1500	2010-Q4
CPU	Intel i7 9x	50	2009-Q4
GPU	FireStream 9270	1000	2009-Q1

# Applicazioni

- ▶ Algebra lineare
- ▶ Crittografia
- ▶ Data mining
- ▶ Scienze della terra
- ▶ Scientific computing (in generale)
- ▶ Teoria dei segnali
- ▶ Video processing

and more...

# Introduzione alla Grafica Computazionale

## Definizione

Insieme di tecniche per la produzione di immagini bitmap a partire da dati o acquisiti o costruiti da modelli.

## Fasi

- ▶ Definizione degli oggetti che compongono l'immagine.
- ▶ Produzione dell'immagine (rendering).

# Introduzione alla Grafica Computazionale

## **Pipeline grafica**

insieme delle operazioni caratteristiche della fase di rendering.

# Introduzione alla Grafica Computazionale

- ▶ Trasferimento della descrizione della scena: l'insieme di vertici che definiscono gli oggetti, i dati di illuminazione, le texture e il punto di osservazione della scena.
- ▶ Trasformazioni di vertici: rotazioni, scaling e traslazioni degli oggetti.
- ▶ Clipping: eliminazione di oggetti o parte di essi che non sono visibili per la prospettiva data.
- ▶ Lighting and shading: calcolo dell'illuminazione ed ombreggiatura degli oggetti a partire dalla fonte luminosa della scena.
- ▶ Rasterization: generazione dell'immagine bitmap. Le coordinate 3D vengono trasformate in coordinate 2D. In questa fase vengono applicate le texture agli oggetti ed altri effetti grafici.

# Evoluzione delle GPU

La Graphics Processing Unit è la componente deputata ad eseguire le operazioni di rendering nelle moderne schede video.

**1980** I primi chip grafici: funzioni limitate in particolare assenza di funzioni per il disegno delle scene bidimensionali.

**1985** I chip grafici erano vere e proprie CPU opportunamente modificate (design e ISA). Soluzione costosa spinta dalla diffusione di applicazioni CAD.

**1990** Chip grafici integrati, dedicati e programmabili (Riduzione dei costi).

# Evoluzione delle GPU

## 1995

- ▶ Diffusione della grafica 3D su spinta del mercato dei videogiochi.
- ▶ Sviluppo di chip integrati per l'accelerazione 3D.
- ▶ Affermazione delle specifiche OpenGL e DirectX.  
Nascosta la complessità di programmazione di acceleratori 3D differenti.
- ▶ Pipeline grafiche implementate all'interno delle GPU.

# Evoluzione delle GPU

**2000** Introduzione nelle GPU dello shading.

## Definizione

Capacità di eseguire shader ovvero un insieme di istruzioni che implementano la pipeline grafica.

## Tipologie di shader

- ▶ shader di vertici: trasforma e gestisce la posizione dei vertici di un oggetto.
- ▶ shader di pixel o di frammento: gestisce i singoli pixel di una immagine consentendo l'applicazione di texture.
- ▶ shader geometrici: a partire da una serie di vertici di un oggetto costruisce oggetti più complessi (esempio la tassellatura)

# Evoluzione delle GPU

Vengono prodotte GPU con capacità di shading programmabili.

Ogni shader è eseguito su unità di calcolo dedicate.

Le GPU diventano "flessibili" quasi come le CPU.

Nascono i primi progetti GPGPU (General Purpose computing on GPU) che sfruttano le unità di shading sui vertici.

Primi esempi utilizzavano le API OpenGL per definire shader sui vertici tali da mappare l'applicazione parallela general purpose.

# Evoluzione delle GPU

## 2005

Problema di distribuzione del carico delle unità di shader specializzate (scarsa efficienza).

Nascita del Unified Shader Model. Ovvero il modello che utilizza lo stesso insieme di istruzioni per la definizione delle varie tipologie di shader.

Le unità di calcolo sono tutte identiche.

## 2007 "full" GPGPU

Nvidia rilascia Compute Unified Device Architecture (CUDA).

AMD rilascia Brook+.

Tali framework consentono di utilizzare l'elevato numero di unità di calcolo della GPU senza accervi tramite API grafiche.

# Multicore era

Diffusione dei processori multicore nel mercato consumer.

**2007** Intel Yonah 2 core per processore.

**2010** AMD Opteron Magny-Cours 12 core per processore.

Diffusione delle applicazioni concepite per architetture parallele.

Diffusione della applicazioni GPGPU.

Effetto collaterale: problemi di portabilità!

Necessità di programmare i dispositivi di calcolo a disposizione in modo trasparente (Heterogeneous Computing).

**2008** Khronos Compute Working Group rilascia Open Computing Language (OpenCL).

*OpenCL is the first open, royalty-free standard for cross-platform, parallel programming of modern processors found in personal computers, servers and handheld/embedded devices. [Khronos].*

