Il contributo

Storia e applicazioni delle GPU

Hello World!

Tradizionale messaggio informatico

Andrea D'Urbano, Alessandro Fasiello *Scuola superiore ISUFI, Università del Salento*

L'articolo è strutturato in due macro sezioni. Nella prima sarà presentata la rapidissima evoluzione delle GPU: una tecnologia che si colloca come supporto hardware per nuovi paradigmi di calcolo come, ad esempio, *deep learning* nell'universo dei *Big Data*. Nella seconda parte si andranno ad analizzare le applicazioni di queste tecnologie in vari ambiti come ad esempio medico o economico.

Introduzione

Comunemente usate per processare le immagini ludiche dei videogiochi al computer, le GPU (Graphics Processing Unit) risultano essere, in realtà, un'importante risorsa per numerosi campi di ricerca, offrendo una potenza di calcolo altrimenti irraggiungibile. Processori ottimizzati per accelerare i calcoli grafici, si presentano, infatti, con la peculiare caratteristica di essere fortemente parallelizzati, offrendo così una capacità di calcolo parallelo ordini di grandezza superiore alle CPU.

Storia ed evoluzione delle GPU

Spinte dalla ricerca di un *rendering real-time* di immagini 3D ad alta risoluzione sempre più veloce e complesso, proprie del mondo del gaming e delle sue simulazioni, le GPU si sono fatte strada nel mercato elettronico trasformandosi da facoltativo dispositivo anche per i calcolatori più potenti, a necessità indiscussa per ogni configurazione. Ma quali sono le ragioni del loro successo?

Effettuare *rendering* di immagini risulta essere un'operazione caratterizzata da parallelismo intrinseco, poiché l'operazione di elaborazione di ogni pixel è analoga ma sostanzialmente indipendente da quella degli altri. È per questo motivo che una soluzione provvista di un'inferiore velocità di calcolo seriale ma con una potenza di calcolo parallelo decisamente maggiore si rivela necessaria. Si preferisce così l'utilizzo delle GPU alle CPU (Central Processing Unit) i processori primari del computer, caratterizzate da elevatissima velocità seriale (con frequenze nell'ordine dei GHz) ma con limitato parallelismo (dai 2 o 4 core delle classiche configurazioni da casa/ufficio, ai rarissimi 36 core propri di configurazioni professionali orientate a workstations e servers).

Gli albori

Le origini delle Schede Video possono essere viste nei semplici frame buffer integrati dei primi anni '80, chip TTL che sfruttando la CPU erano in grado di elaborare modelli wireframe solo su display raster [1].

Pioniere in questi anni fu la *IBM*, produttrice nel 1984 della scheda *PGA* (*Professional Graphics Controller*) che, prima a montare un microprocessore *Intel 8088* dedicato e 320 kB di VRAM, era in grado di accelerare l'elaborazione grafica 2D e 3D permettendo la rotazione del modello e il clipping delle immagini. Non pensata per il mercato di massa, ma per la sola applicazione professionale, con il suo prezzo di listino di circa 5000 dollari, offriva una valida e competitiva alternativa alle Workstation dedicate al CAD, i cui prezzi raggiungevano e superavano i 50000 dollari.

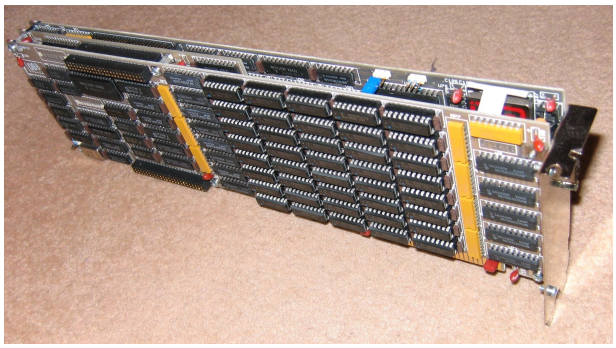

Figura 1: *IBM Professional Graphics Controller*

Negli anni immediatamente successivi gli obbiettivi raggiunti furono l'aggiunta di funzioni quali la rasterizzazione di poligoni, ombreggiatura e illuminazione di solidi e vertici, z-buffer e fusione dei colori.

Di notevole importanza, nel 1989, il lancio della libreria *OpenGL* (tuttora standard per la grafica 3D in ambiente Unix) da parte dell'azienda *Silicon Graphics*, emersa in questi anni come leader nel campo della Computer Grafica ad alte prestazioni.

Gli anni '90

Nel '93 la *Silicon Graphics* lancia il suo *RealityEngine*, dove vi erano per la prima volta schede e chip logici distinti per i diversi livelli della pipeline

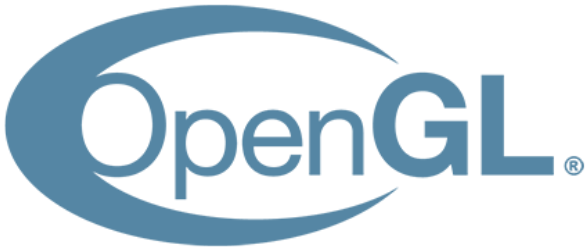

Figura 2: *Logo della libreria OpenGL*

grafica, anche se era ancora presente una forte dipendenza, per la prima parte, dalla CPU.

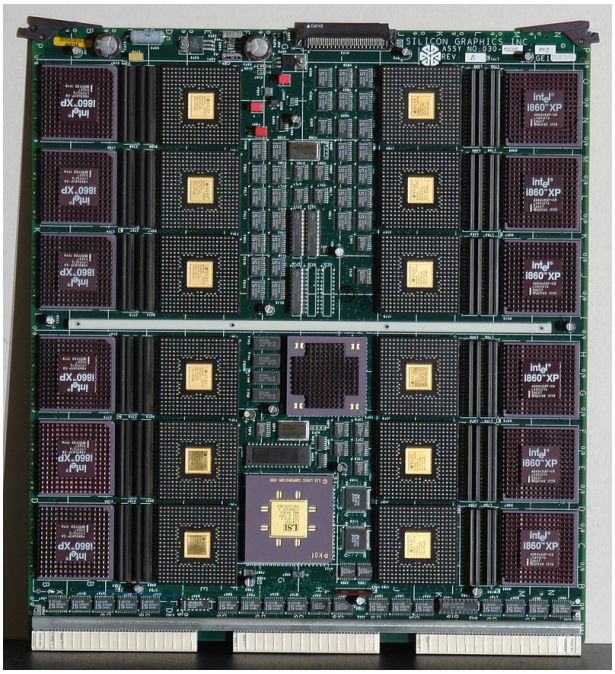

Figura 3: *RealityEngine*

Se le schede della *SGI* erano mirate al pubblico di professionisti e montate su Workstation, negli stessi anni iniziavano a farsi strada nel market domestico aziende come il colosso *NVIDIA*, *3DFX*, *ATI* e *Matrox*.

Schede dedicate al gaming come la *3DFX Voodoo* (1996) e la *NVIDIA Riva TNT* (1998) vedono la luce in questi anni, spinte dalla richiesta di elaborazione di immagini 3D da parte di pietre miliari videoludiche quali *Quake*, *Doom* o *Wolfenstein 3D*.

Caratterizzate da pipeline limitate dalla capacità di generare un solo pixel per ciclo di Clock, la necessità di maggiore velocità di elaborazione ha aperto la strada alla loro parallelizzazione, determinando, così l'aumento del numero di Core al fine di aumentare il numero di pixel elaborati per ciclo di Clock [1].

Un notevole passo avanti, che segna l'inizio di una nuova era e determina la nascita del termine "GPU", è stato compiuto nel 1999 con il lancio della *NVIDIA GeForce256*, prima scheda in grado di implementare sul suo processore dedicato l'intera pipeline con il supporto hardware per *Transform and Lightning*. Il salto tecnologico fu tale da surclassare completamente le altre case produttrici (*3DFX* fu inglobata dalla stessa *NVIDIA* poco dopo), lasciando come unica rivale la linea di chip grafici *Radeon* dell'azienda *ATI* (successivamente acquistata da *AMD*, unico attuale concorrente di *NVIDIA* nella produzione di GPU), la quale era riuscita a difendersi grazie al lancio (nell'aprile del 2000) della *ATI Radeon 7500*, dalle caratteristiche simili alla diretta concorrente [2].

Figura 4: *NVIDIA GeForce256, la prima GPU al mondo*

L'avvento delle pipeline programmabili

Se le rivoluzionarie schede del decennio precedente risultavano, tuttavia, ancora rigide e ingessate nel loro unico obiettivo di fornire la sola accelerazione grafica, questa limitazione è stata superata nel 2001, quando la *NVIDIA* rilascia *GeForce 3*, la prima GPU ad avere parte della pipeline programmabile attraverso programmi denominati *shaders*, piccoli kernels scritti in un linguaggio basato sull'assembly. Nello stesso anno la *Microsoft* lancia la *X-Box* munita di processori grafici *NVIDIA*, prima console a sfruttare le librerie grafiche *DirectX*, le quali rappresentano tuttora il principale standard grafico nei sistemi *Windows*[2].

L'anno successivo è segnato dall'avvento nel mercato della prima scheda completamente programmabile, sempre di casa *NVIDIA*: la *GeForce FX*, provvista di 80 milioni di transistor, 128 MB di DRAM DDR a 128 bit e frequenza di clock di

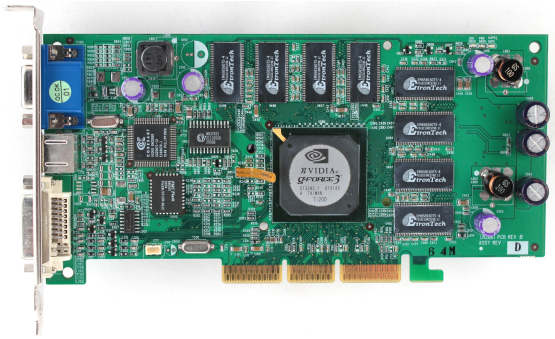

Figura 5: *NVIDIA GeForce 3*

400 MHz, seguita dalla *ATI Radeon 9700*. Queste schede permettevano operazioni con vertici e shaders programmabili, consentendo limitate operazione di mapping di input-output specificate dall'utente.

Da qui fu breve il passo che portò, finalmente, nel 2003 al GPU computing per operazioni non grafiche, con la comparsa del completo supporto per i floating point e un avanzata elaborazione delle *texture*, grazie all'avvento della libreria *DirectX 9* [1].

L'esplosione nello sviluppo

Il 2004 vede una forte accelerazione nello sviluppo tecnologico delle GPU.

Dal lato hardware, vengono rilasciate, dalle ormai uniche concorrenti in campo, la *GeForce 6* e la *Radeon X800*, prime schede ad usare il bus PCI-express, a supportare multi-rendering buffers e calcoli a 64-bit con double; dal lato software, vedono la luce i primi linguaggi ad alto livello dedicati alle GPU: *Brook* e *Sh*.

Il frutto di tale innovazione matura nel 2006 in cui si assiste ad una vera e propria esplosione nell'utilizzo delle GPU come processori di massivo calcolo parallelo grazie sopratutto all'avvento della *NVIDIA GeForce 8 series*, la cui 8800 fu la prima GPU con un processore unificato completamente programmabile chiamato *Streaming Multiprocessor*, o SM, che gestiva il calcolo di vertici, pixel e geometria. Supportando le librerie DirectX 10, sono le prime schede a sfruttarne lo *shader model 4.0* che ne determina una maggiore programmabilità [1].

A coronare il processo verso le GPGPU (General Purpose GPU), nasce, per la *NVIDIA G80*,

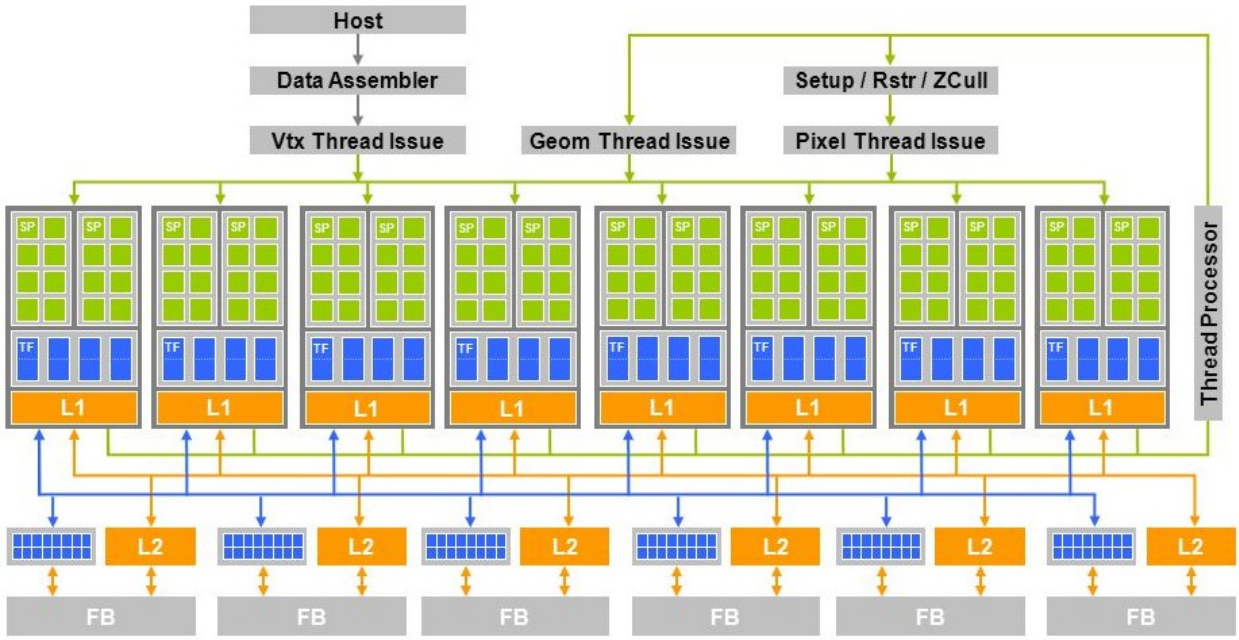

Figura 6: *L'architettura della GeForce 8 series è costruita intorno all'idea di un processore programmabile*

il linguaggio *CUDA*, imitato poco dopo da *ATI Stream* per le schede ATI [3].

Partendo proprio dalla rivoluzionaria *G80* e dall'analoga *Quadro FX5600*, nel 2007 *NVIDIA* lancia le GPU *TESLA*. Grazie ad esse "La potenza di calcolo precedentemente disponibile solo nei supercomputer viene resa accessibile a un numero molto più grande di ricercatori in campi quali la ricerca farmacologica, l'imaging medicale e la modellizzazione meteorologica." [2]

L'avvento dell'architettura Fermi

Ormai orientata al GPGPU, nel 2009 *NVIDIA* annuncia la sua nuova architettura: *Fermi*, che arriverà sul mercato agli inizi del 2010.

Si tratta di un'architettura disegnata appositamente per il GPU computing e dotata, quindi, di gerarchia cache HW, ECC, spazio dei memory adress unico, esecuzione del kernel simultanea, migliori prestazioni in precisione double e dual warp schedulers. La potenza di calcolo parallelo risulta ormai impressionante grazie ad un totale di 480 CUDA core già al lancio della prima scheda Fermi, la *GTX480 Fermi*, provvista di 3 miliardi di transistor, 1.5 GB di DRAM GDDR5 a 384-bit e frequenza di funzionamento di 700MHz [1].

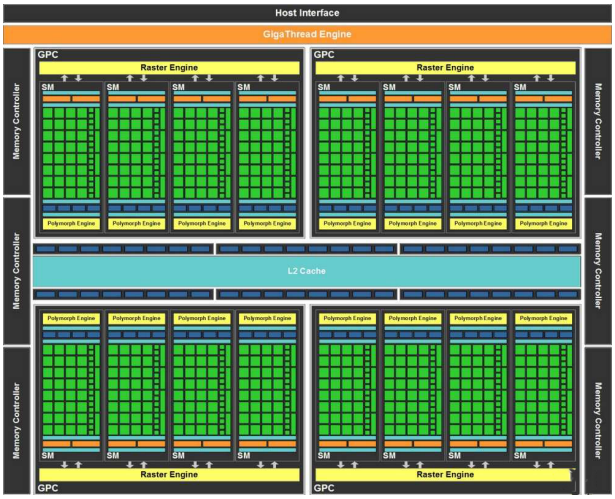

Figura 7: *Architettura Fermi*

L'architettura Kepler, alla ricerca dell'efficienza energetica

Nel 2012 *NVIDIA* continua ad innovare presentando la *GeForce GTX serie 600*, "le GPU di gioco più veloci al mondo" (cosí definite dall'azienda al loro lancio), prime a sfruttare l'architettura Kepler.

Questa nuova architettura basa le sue fondamenta sulle soluzioni Fermi focalizzandosi, però, sull'efficienza energetica raggiunta attraverso l'uso di un clock GPU unificato, una programmazione statica semplificata delle istruzioni e una maggiore enfasi sulle prestazioni per Watt. Ab-

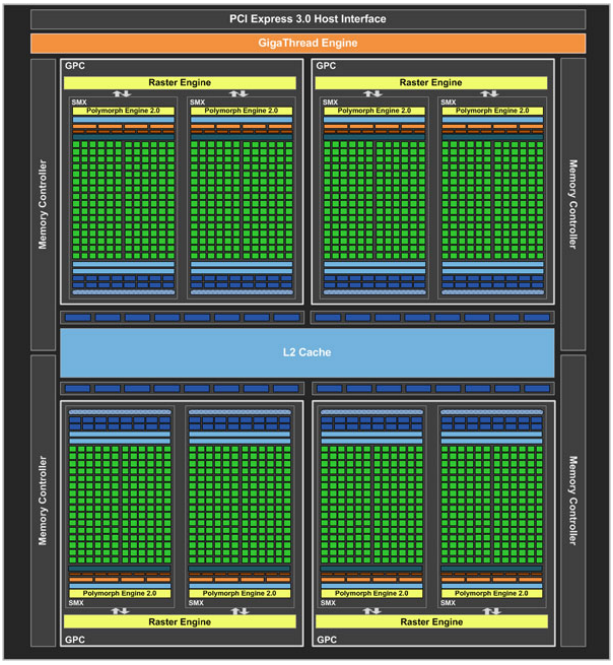

Figura 8: *Architettura Kepler*

bandonare l'uso dello shader clock visto nelle precedenti architetture ha infatti aumentato l'efficienza nonostante il conseguente aumento di CUDA core necessari a raggiungere superiori livelli di performance, complici core meno dispendiosi di energia (due core Kepler consumano il 90% di energia di un core Fermi) e il passaggio ad un clock unificato in grado di offrire un risparmio energetico del 50% [4].

Maxwell: prestazioni a basso consumo

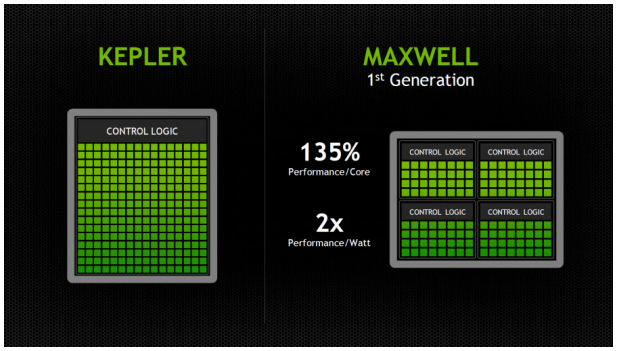

Figura 9: *Confronto strutture Kepler e Maxwell*

Nel 2014 viene presentata Maxwell, l'architettura *NVIDIA* di decima generazione, che presenta innovazioni nelle prestazioni, nella grafica e nell'efficienza delle GPU *GeForce GTX*: la casa produttrice dichiara, infatti, operando un confronto tra la *GeForce GTX 550 Ti*, dotata di architettura Fermi, e la nuova *GeForce GTX 750 Ti*, un raddoppio delle prestazioni a fronte di un dimezzamento dei consumi, ridotti, invece, addirittura ad un quarto se confrontata con la *GeForce GTX 480* (la testa di serie dell'architettura Fermi in quanto a prestazioni) di cui raggiunge le performace [5].

Figura 10: *Architettura Maxwell*

Nel settembre dello stesso anno vengono presentate le top gamma della serie 900: le *GeForce GTX 970* e *980*. Tali schede fanno sfoggio di nuove tecnologie quali: la VXGI (Voxel Global Illumination), che consente per la prima volta a GPU destinate al *gaming* di offrire un'illuminazione globale dinamica in tempo reale; MFAA (Multi-Frame Sampled Anti-Aliasing), che modifica gli "anti-aliasing sample pattern" sia nei singoli fotogrammi che tra fotogrammi multipli per produrre la miglior qualità d'immagine con la stessa rapidità dell'anti-aliasing convenzionale; il DSR (Dinamic Super Resolution), che utilizza un 13-tap Gaussian filter per renderizzare l'immagine in 4K e scalarla alla dimensione nativa del display in modo da ottenere una qualità migliore rispetto al rendering diretto in 1080p; il VR Direct, che punta all'ottimizzazione della realtà virtuale (sopratutto se utilizzate in configurazione doppia tramite SLI grazie al quale ciascuna GPU puó essere assegnata al rendering

delle immagini destinate ad un singolo occhio) [6].

2015: NVIDIA si lancia nel *Deep Learning*

Il 2015 segna un anno di svolta nell'azienda ormai leader del settore che si tuffa nell'ambito del *Deep Learning* sviluppando diversi prodotti ad esso mirati: *NVIDIA Tegra X1*, un chip mobile a 256 core in grado di offrire un Teraflop di potenza di elaborazione alle applicazioni di deep learning e Computer Vision; *Jetson TX1*, un supercomputer integrato su un modulo che abilita una nuova generazione di macchine intelligenti e autonome; *NVIDIA Drive*, il quale apre la via verso lo sviluppo di auto a guida autonoma e, prima tra le novità, la GPU *GeForce GTX TITAN X*, che con i suoi 6 Teraflops di potenza di calcolo rappresentava il processore più potente mai realizzato fino ad allora per l'addestramento delle reti neurali profonde [2].

Pascal e Polaris: il nuovo metodo di produzione FinFET

Il 2016 porta con se nuove architetture su entrambi i principali fronti di sviluppo GPU: *NVIDIA* e *AMD* lanciano le architetture Pascal e Polaris, entrambe basate sulla nuova tecnologia FinFET rispettivamente a 16 e 14 nm che permette di produrre nuovi transistor non più planari, ma tridimensionali.

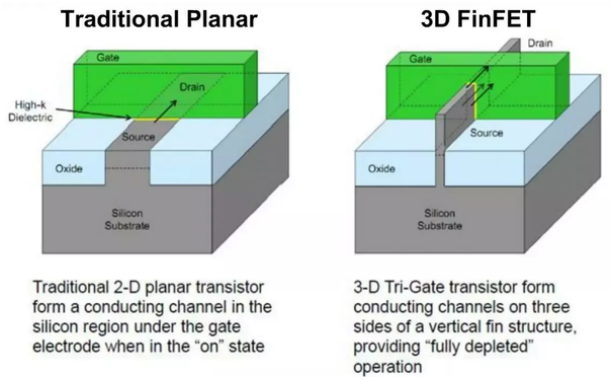

Figura 11: *Il nuovo metodo di produzione FinFET*

La prima di queste viene lanciata da *NVIDIA* ad Aprile, presentata sulla scheda *Tesla P100*, in grado di offrire 5.3 Teraflops di performance in double-precision (3 volte più veloce rispetto alla precedente generazione di *Tesla K40*) e 10.6 Teraflops in single-precision [7, 8]. Il mese successivo invece, tale architettura viene impiegata nella serie 10, determinando la nascita della *GeForce GTX 1080* che supera le prestazioni della precedente top gamma, la *Titan X*, riducendone i consumi di quasi 1.5 volte. Si presenta con 2560 Cuda Cores, un clock di base a 1607 MHz che arriva in boost a 1733 MHz, 8 GB di memoria GDDR5X a 256-bit a 10 Gbps [9, 10].

Il cavallo di bataglia *NVIDIA* della serie 10, la *GeForce 1080 Ti* giungerà nel mercato nei primi mesi dell'anno successivo, presentando un notevole incremento delle prestazioni (ben 35% di potenza in più rispetto alla *1080*) , con 11.3 Teraflops, 3584 Cuda Cores, Clock boost a 1582 MHz, memoria di 11 GB GDDR5X a 352-bit dotata di una velocità di 11 Gbps [11].

AMD risponde a Giugno con la sua linea *Radeon 400 series*, equipaggiata della sopracitata architettura Polaris in grado di sfruttare al massimo la tecnologia FinFET per ottenere transistor di 14 nm. Tale architettura migliora il processamento dei comandi con due nuove tecniche di Quality-of-Service (QoS) progettate per aumentare la velocità di risposta e le prestazioni del sistema. La prima è la Quick Response Queue che consente agli sviluppatori di designare una compute tasks queue come prioritaria tramite le API, in modo da far destinare dagli ACE dei workgroups all'attività ad alta priorità prima delle attività standard [12].

Figura 12: *Architettura Polaris*

Questo schema di definizione delle priorità garantisce che le attività ad alta priorità utilizzino più risorse e vengano completate per prime. Ad esempio, questa tecnica viene utilizzata nell'*AMD LiquidVR SDK* per stabilire le priorità

di "distorsione temporale", un'attività sensibile alla latenza e al jitter e garantire che la distorsione temporale avvenga immediatamente prima della sincronizzazione verticale [13].

SPECIFICHE DEL SISTEMA	
GPU	**8 Tesla V100**
Prestazioni (precisione mista)	**1 petaFLOPS**
Memoria della GPU	**Totale sistema 256 GB**
CPU	**Doppio Intel Xeon 20 core E5-2698 v4 2,2 GHz**
Core NVIDIA CUDA®	**40.960**
NVIDIA Tensor Core (su sistemi basati su V100)	**5.120**
Requisiti di alimentazione	**3.500 W**
Memoria di sistema	**512 GB di memoria RDIMM DDR4 a 2.133 MHz**
Spazio di archiviazione	**4 SSD RAID 0 da 1,92 TB**
Rete	**Doppia 10 GbE, 4 IB EDR**
Sistema operativo	**Ubuntu di Canonical Red Hat Enterprise Linux**
Peso del sistema	**61 kg**
Dimensioni del sistema	**P 866 x L 444 x H 131 (mm)**
Dimensioni dell'imballo	**P 1.180 x L 730 x 284 H (mm)**
Temperatura di funzionamento	**10–35° C**

Figura 13: *Nello stesso 2016 NVIDIA lancia DGX-1, il primo supercomputer compatto dedicato al deep learning [14]*

2017: Vega e Volta

Nel 2017 sia *AMD* che *NVIDIA* creano nuove architetture, tuttavia destinate a due mercati differenti. *AMD* dedica *Vega* principalmente al mercato del *gaming*, andando a concorrere ancora una volta contro l'architettura *Pascal NVIDIA* (di cui contestualmente viene prodotta la punta di diamante: la sopradescritta *1080 Ti*); nasce così *AMD RX Vega series*, che porta con sè alcune novità, come il supporto per le memorie HBM2 (più veloci rispetto alle GDDR5), il Primitive Shader per una migliore elaborazione della geometria che sostituisce la precedente pipeline *Vertex Shader + Geometry Shader* garantendo maggiore efficienza e produttività e l'NCU per l'elaborazione nativa di operazioni a 8, 16, 32 o 64 bit in ciascun ciclo di clock, fornendo anche supporto per Rapid Packed Math che permette di elaborare 2 operazioni in 16-bit alla stessa velocità di un operazione in 32-bit [15, 16].

Il lavoro di *NVIDIA* è, invece, ancora una volta dedicato ai supercomputer per l'IA: si tratta di *Volta*, l'architettura "progettata per potare l'intelligenza artificiale in tutti i settori" [17]. Applicata prima su *NVIDIA Tesla V100* e successivamente su *Quadro GV100* e *Titan V*, presenta importanti innovazioni tecnologiche: 640 Tensor Core in grado di accelerare operazioni matriciali, cuore dell'IA, ed eseguire moltiplicazioni di matrici a precisione mista accumulando calcoli in una singola operazione, superando i 100 Teraflops di prestazioni in deep learning; una nuova architettura in grado di combinare Cuda e Tensor Core, con oltre 21 miliardi di transistor; il processo di produzione FinFET a 12 nm; una nuova generazione NVLink per una migliore scalabilità e connessione in parallelo [18]. Queste innovazioni raccolgono come frutto la triplicazione della velocità di deep learning training rispetto alla precedente generazione, e un aumento della velocità fino a 40 volte rispetto al training operato su CPU (NVIDIA Tesla T4 GPU vs Xeon Gold 6140 CPU) [19, 20, 21].

Turing RTX: GPU sotto una nuova luce

Nel 2018*NVIDIA* ottiene quello che essi stessi definiscono come il più importante risultato dall'invenzione della GPU NVIDIA CUDA nel 2006. La nuova architettura Turing riesce, infatti, a combinare il ray-tracing in tempo reale, l'IA, la simulazione e la rasterizzazione ottenendo una vera e propria rivoluzione della grafica [22].

È dotata di processori dedicati per il ray-tracing chiamati RT Core che accelerano l'elaborazione di luce e suono negli ambienti 3D, arrivando anche a 10 Giga Ray al secondo. Turing accelera il ray-tracing in tempo reale di 25 volte rispetto all'architettura *NVIDIA Pascal* di precedente generazione e può eseguire il rende-

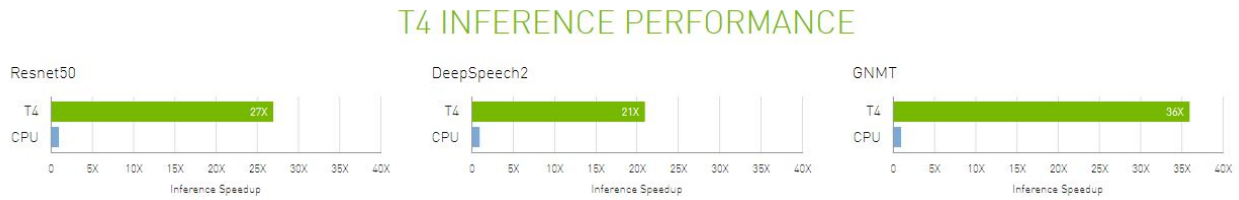

Figura 14: *Tempo di training Tesla Volta vs Tesla Pascal*

ring di frame finali per effetti cinematografici con una velocità 30 volte superiore alle CPU; migliora in modo rilevante le prestazioni raster con una pipeline grafica avanzata e nuove tecnologie di shading programmabili [23]. Queste tecnologie includono shading a tasso variabile, shading texture-space e rendering a più viste, che forniscono un'interattività più fluida con modelli e scene di grandi dimensioni e una migliore esperienza in realtà virtuale. Turing è dotata anche di Tensor Core, offrendo fino a 500 bilioni di operazioni di deep learning al secondo. Questo livello di prestazioni accelera in modo rilevante le funzionalità IA, ad esempio denoising, scaling della risoluzione e re-timing di video. Queste nuove GPU dispongono di una nuova architettura di multiprocessore streaming (SM) che supporta oltre 16 bilioni di operazioni a virgola mobile insieme a 16 bilioni di operazioni a cifra intera al secondo. Gli sviluppatori possono sfruttare fino a 4.608 core CUDA con gli SDK NVIDIA CUDA 10, FleX e PhysX per creare simulazioni complesse, ad esempio particelle o dinamiche dei fluidi per visualizzazioni scientifiche, ambienti virtuali ed effetti speciali [24].

Tale nuova architettura vede la luce nelle *NVIDIA Quadro RTX*, nella *GeForce RTX serie 20*, in schede quali le *RTX 2080, 2070, 2060*, le corrispettive versioni potenziate denominate "Ti", i loro più recenti upgrade denominati "Super" e nella più potente *NVIDIA Titan RTX*, con cui viene raggiunta la vetta dei 130 Tensor Teraflops [25, 26].

Figura 15: *Esempio di Ray-Tracing in tempo reale*

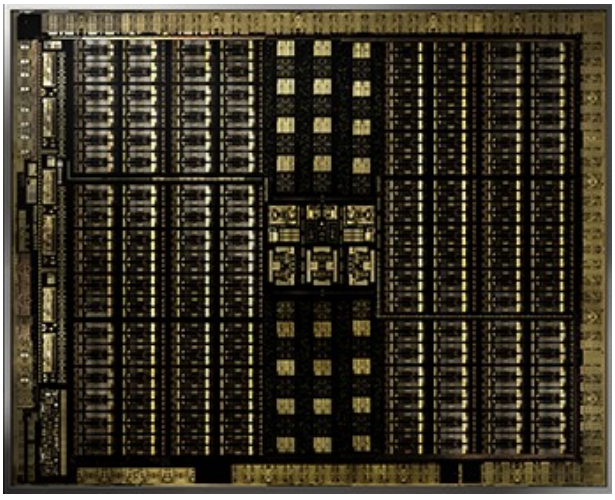

Figura 16: *Architettura Turing*

Applicazioni

La grande potenza di calcolo parallelo offerta dalle GPU permette un'applicazione diretta in svariati campi nei quali l'analisi di una massiva quantità di dati è fondamentale. Questa straordinaria capacità di elaborazione, unita alla possibilità di sfruttare le GPU come ideale supporto hardware per AI (Artificial Intelligence), giustifica la crescente importanza di questa tecnologia in un sempre più vasto spettro di campi [27].

Biologia

Nel caso della ricerca biologia, esistono diverse modellizzazioni usate per simulazioni di evoluzione temporale di sistemi biologici. Ad esempio una modellizzata usata è il sistema *species-based*, nel quale un'entità biologica è suddivisa in classi composte da elementi indistinguibili. Le molecole possono essere rappresentate da funzioni di concentrazione variabili nel tempo e le loro in-

terazioni descritte da relazioni differenziali. Per risolvere le relative equazioni differenziali, quasi sempre analiticamente intrattabili, si sfruttano metodi numerici nei quali è necessario compiere operazioni di algebra lineare. Sfruttando apposite librerie di algebra lineare per architettura GPU si possono velocizzare di molto i calcoli necessari. Le variabili possono anche essere considerate discrete, descrivendo le interazioni tramite processi stocastici. Gli algoritmi però che servono per simulare questi processi sono intrinsecamente sequenziali e quindi difficili da parallelizzare; la possibilità di calcolo parallelo offerta dalle GPU può essere però sfruttata nel generare la sequenza di numeri random o si possono cercare di parallelizzare i calcoli per le singole reazioni.

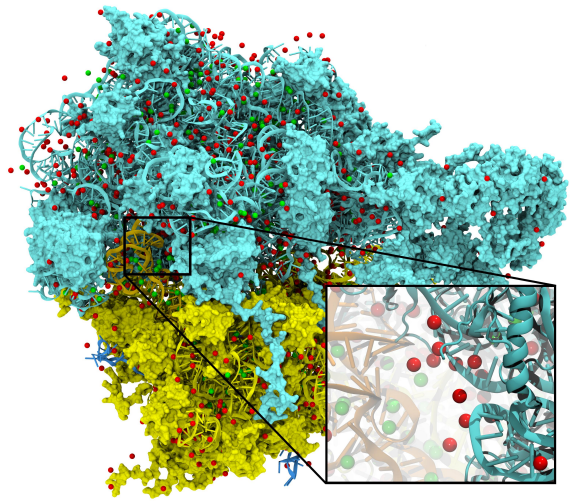

Figura 17: *Modellizzazione di un ribosoma e accelerazione tramite GPU dei processi di interazione con ioni, riprodotto da [28]*

I modelli presentati possono essere migliorati considerando dei compartimenti tra i quali modellizzare il passaggio di sostanze: vengono in pratica simulate così le barriere biologiche. Se si decide di procedere implicitamente, la variabile (discreta o continua) rappresentante un'entità biologica che si potrebbe trovare da entrambi i lati della barriera, viene divisa in due differenti variabili dando vita ad un modello più pesante. Procedendo esplicitamente invece, esistono varie rappresentazioni dei compartimenti. Un esempio per il quale esistono molte implementazioni GPU sono i *P systems*: modelli computazionali che mimano la struttura di una cellula. Questi *P systems* sono costituiti da un insieme di membrane (che può contenere altre membrane), un insieme di sostanze in ogni membrana e delle regole di evoluzione. Questi sistemi esibiscono due livelli di parallelismo, tra membrane e tra sostanze chimiche, che possono essere implementati brillantemente su CUDA (che presenta una struttura logica simile). Ciò supporta l'efficacia di implementazione di questi modelli su CUDA.

L'ultimo esempio analizzato in questa sezione è quello dei modelli *Agent-based*, una generalizzazione dei modelli ad automa cellulare. A partire da tante unità autonome che interagiscono tra di loro solo localmente, i modelli *Agent-based* riescono a simulare fenomeni emergenti: effetti globali riconducibili non alle singolo entità costituenti il sistema bensì all'interazione tra le stesse. Per questo motivo questo metodo è usato per simulare processi infiammatori, crescita tumorale, processi intracellulari e così via. Lo sviluppo di questa tipologia di simulazioni usando calcolo parallelo, è stata dimostrata essere circa tre ordini di grandezza più veloce rispetto all'approccio sequenziale [29].

Ambito medico

Un interessante esempio di applicazione di questo nuovo paradigma di calcolo si può osservare in ambito medico. Con il progredire delle tecnologie utilizzate in fisica medica è diventato necessario cambiare approccio all'analisi della crescente mole di dati, sia per analizzare efficientemente le informazioni provenienti da migliaia di pazienti che per gestire i processi di imaging resi estremamente pesanti dal costante aumentare della risoluzione degli apparati disponibili. Sono molteplici i fattori che rendono limitante il tempo di elaborazione necessario per l'imaging o per la pianificazione di trattamenti; la maggiore sensibilità spaziale e temporale dei macchinari utilizzati, il rendering di risultati animati (4D invece di 3D), l'utilizzo di geometrie coniche per i fasci di raggi-x, le sequenze di impulsi sempre più sofisticate nella risonanza magnetica e la complessità crescente degli algoritmi di pianificazione dei trattamenti.

Per quanto riguarda la ricostruzione delle immagini, la velocità impiegata nel processo è di estrema importanza. Ad esempio con applicazioni real-time è possibile calibrare, durante l'acqui-

sizione stessa dei dati, vari parametri di ricostruzione e ottimizzare il rapporto segnale-rumore. Prima dell'avvento delle GPU la tecnologia che permetteva una ricostruzione real-time si basava su un design specifico di hardware come ad esempio *field programmable gate array* (FPGA) e *application-specific integrated circuits* (ASICs). Il grosso del lavoro che una GPU deve eseguire utilizzando gli algoritmi abitualmente adoperati nella ricostruzione delle immagini si traducono in operazioni di moltiplicazione tra un vettore di dati (rappresentante la discretizzazione del segnale in ingresso) e una matrice che descrive la risposta del sistema di imaging. Questa modellizzazione è giustificata da un approccio che considera il sistema di imaging assimilabile ad una trasformazione lineare. Quindi il core del processo di ricostruzione si basa sulla discretizzazione e inversione delle trasformazioni lineari in esame (spesso di Fourier o di Radon). La moltiplicazione tra matrice della trasformazione e vettore dei dati non viene eseguita esplicitamente usando la definizione della moltiplicazione di righe per colonne poichè questo approccio sarebbe computazionalmente inefficiente. Viene sfruttata invece la forma della matrice in esame per ottimizzare il processo di calcolo tramite appositi algoritmi (spesso si tratta di matrici sparse, ovvero con molti elementi nulli).

Nel caso del calcolo di dosi e in terapie radiologiche la situazione risulta meno favorevole. Questi trattamenti utilizzano radiazioni ionizzanti per distruggere le cellule tumorali del paziente cercando di minimizzare i danni ai tessuti sani. Per calcolare quindi le dosi di radiazione e la loro distribuzione angolare in funzione del tessuto da attaccare, della forma tridimensionale della zona interessata, dello stadio della malattia e di altri parametri, è necessario risolvere un problema di ottimizzazione. Al momento i piani di trattamento in IMRT (Intensity-Modulated RadioTherapy), data la complessità del processo, sono calcolati per "prove ed errori": vengono generate e simulate molteplici possibili soluzioni fino a quando non si raggiunge un buon rapporto tra la dose di radiazione assorbita dal target e il rischio di danni ai tessuti sani. Diminuire il tempo di calcolo necessario per questi processi di ottimizzazione sarebbe utile anche per sviluppare un nuovo paradigma proposto per migliorare

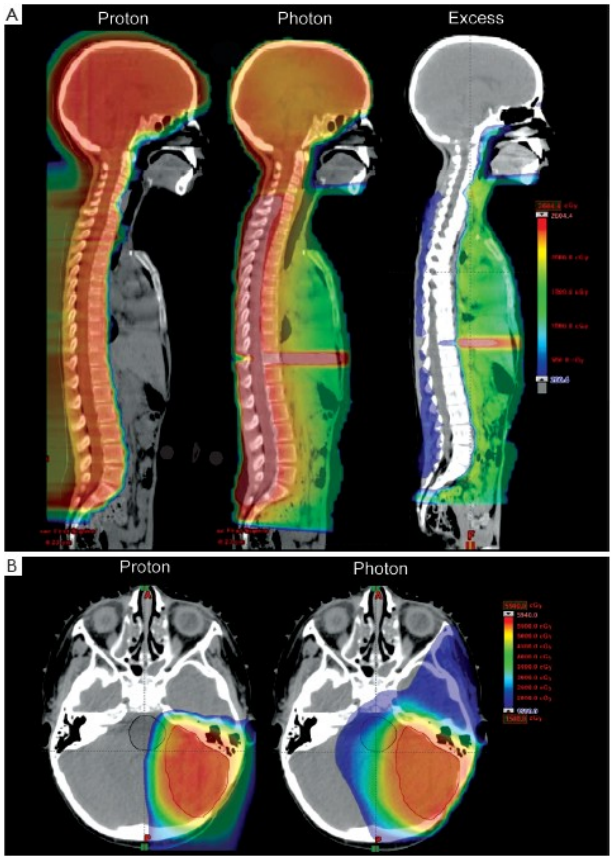

Figura 18: *Paragone dosimetrico tra protoni e fotoni (la terza colonna si riferisce alla dose in eccesso depositata dai fotoni). In basso, la simulazione della dose assorbita dal cervello. [30]*

la qualità dei trattamenti radiologici: la radioterapia adattiva. Questa metodologia prevede l'utilizzo di un modello tridimensionale del paziente ed una ricalibrazione in tempo reale dei parametri dei fasci di radiazioni, basandosi sulle dosi precedentemente somministrate. Tradizionalmente il calcolo delle dosi è ottenuto con tecniche analitiche (pencil beam, PB, convolution-superposition, CS) e tecniche di simulazione statistica Monte-Carlo (MC). Queste ultime sono le più accurate ma al contempo più lente rispetto alle tecniche analitiche. Le tecniche MC si prestano naturalmente ad un calcolo parallelo, poichè si basano sulla sovrapposizione dell'interazione con il bersaglio di miliardi di particelle simulate indipendentemente. Tuttavia il guadagno in tempo dovuto all'implementazione di queste simulazioni su archiettura GPU è modesto a causa di processi fisici. Infatti, in particolare per le simulazioni ad energie più elevate, si deve tenere conto del presentarsi di altri effetti fisici quali assorbimenti, scattering e produzione di coppie

di particelle-antiparticelle. Vengono così create varie particelle secondarie che impediscono una parallelizzazione di tante particelle simultaneamente. Sono stati proposti varie soluzioni, tra cui l'utilizzo di una coda globale per organizzare il processamento delle varie particelle e il calcolo di fotoni ed elettroni a parte, sequenzialmente invece che concorrenzialmente.

In fisica medica, una volta reppresentati i dati acquisiti in immagini, è spesso necessario combinare informazioni codificate in immagini differenti prese ad esempio in momenti diversi o unire i dati provenienti da più strumenti. Un'altra tecnica fondamentale nella manipolazione di immagini consiste nel raggruppare pixel (o più in generale voxel, dei pixel tridimensionali) con proprietà assimilabili. Questa tecnica permette ad esempio di definire, tramite un processo automatizzato, i contorni dei vari tessuti del paziente. Questi procedimenti appena descritti sono computazionalmente onerosi ma utilizzando le GPU possono essere migliorati di diversi ordini di grandezza [31].

Ambito economico

Sempre più nel settore del commercio, in particolare per quanto riguarda grandi aziende e multinazionali, viene sfruttata la tecnologia delle GPU per gestire più efficacemente l'enorme mole di dati a disposizione. Un esempio può essere quello della collaborazione tra le celebri multinazionali Walmart e Hewlett Packard (HP), che hanno sfruttato i dati sugli acquisti dei clienti intorno al globo. Sfruttando tecniche di machine learning sono riusciti a migliorare efficacemente le loro strategie di decisione dei prezzi e nelle campagne pubblicitarie [32].

Tecniche simili sono sfruttate dagli algoritmi di famosi social network come Facebook ed Instagram per gestire le liste di amici suggeriti o di post e pubblicità proposti all'utente. In generale moltissime compagnie che si basano su contenuti on-line fronteggiano la necessità di gestire una grande quantità di dati spesso anche prodotta ad un tasso elevato. In questi casi le soluzioni hardware offerte dalla tecnologia delle GPU si rivelano vincenti.

Pubblica amministrazione

Anche l'amministrazione pubblica, data la grande quantità di dati che deve gestire, necessita sempre più di tecnologie performanti su analisi in parallelo. Infatti la gestione di documenti virtuali è facilmente parallelizzabile per ogni persona. La grande mole di dati deriva prima di tutto dall'elevato numero di persone che vivono in uno stato ma in secondo luogo anche dalla varietà di servizi che il singolo cittadino riceve. Ad esempio il livello di dati sanitari prodotti dalla fascia più anziana di popolazione è ben diversa da quella generata da trentenni, o le informazioni scolastiche e universitarie relative a bambini e ragazzi sono estremamente diverse da quelle prodotte dalle altre fasce di età.

Per migliorare l'efficacia della pubblica amministrazione sta avvenendo, con vari gradi di successo, una rivoluzione digitale in modo tale da ridurre drasticamente l'utilizzo di carta, ottenendo così un impatto positivo sull'ambiente, e aumentando la velocità di erogazione dei servizi diminuendone di molto i costi. Secondo un report di McKinsey [27], l'utilizzo di tecniche proprie del mondo dei *Big Data* nel settore pubblico dell'Europa porterebbe, potenzialmente, ridurre le spese di amministrazione del 15-20 %. Questa stima fornisce un'idea dell'impatto che queste tecnologie possono avere sulla società.

Conclusioni

È stata fornita una visione dello sviluppo della tecnologia delle GPU e di una minima parte delle sue applicazioni. Sfruttando, oltre alla potenza di calcolo parallelo offerto dalle GPU, anche tecnologie e tecniche come cloud computing, computazione quantistica, bio-inspired computing, intelligenza artificiale e machine learning, sarà possibile ottenere traguardi prima ritenuti irraggiungibili. Il futuro si presenta carico di sfide ed opportunità in questi settori innovativi.

[1] Chris McClanahan; History and Evoluzion of GPU Architecture
https://pdfs.semanticscholar.org/2479/80e834f1c8f684d85067402f950930e6af91.pdf

[2] CRONOLOGIA DI NVIDIA Una storia di innovazione
https://www.nvidia.com/it-it/about-nvidia/corporate-timeline/

[3] Tariq, S. (2011). An introduction to GPU computing and CUDA architecture. NVIDIA Corporation, 6(5)

[4] NVIDIA GeForce GTX 680: arriva Kepler
https://www.hwupgrade.it/articoli/skvideo/3181/nvidia-geforce-gtx-680-arriva-kepler_2.html

[5] NVIDIA è leader nella rivoluzione "Performance Per Watt" con l'architettura grafica "Maxwell"
https://www.nvidia.it/object/geforce-gtx-750-feb18-2014-it.html

[6] NVIDIA svela la nuova architettura Maxwell
http://www.digitalnewschannel.com/comunicati-stampa/nvidia-svela-la-nuova-architettura-maxwell-16448

[7] ARCHITETTURA NVIDIA PASCAL
Infinite capacità di elaborazione per infinite opportunità
https://www.nvidia.com/it-it/data-center/pascal-gpu-architecture/

[8] NVIDIA TESLA P100 The World's First AI Supercomputing Data Center GPU
https://www.nvidia.com/en-us/data-center/tesla-p100/

[9] NVIDIA GeForce GTX 1080
https://www.techpowerup.com/gpu-specs/geforce-gtx-1080.c2839

[10] GEFORCE GTX 1080
10: GAMING PERFECTED
https://www.nvidia.com/it-it/geforce/products/10series/geforce-gtx-1080/

[11] GEFORCE GTX 1080 Ti
10: GAMING PERFECTED
https://www.nvidia.com/it-it/geforce/products/10series/geforce-gtx-1080-ti/?nvid=nv-int-geo-it-1080-ti#buyfrompartners

[12] RADEON Dissecting the Polaris architecture
https://www.amd.com/system/files/documents/polaris-whitepaper.pdf

[13] AMD POLARIS
I DETTAGLI SULLA NUOVA ARCHITETTURA
https://tech.everyeye.it/articoli/speciale-amd-polaris-dettagli-sulla-nuova-\architettura-30047.html

[14] NVIDIA DGX-1
STRUMENTO DI RICERCA ESSENZIALE PER L'IA
https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-1/dgx-1-rhel-centos-datasheet-update-r2_Updates_NV_web_it_IT.pdf

[15] Radeon RX Vega 64 Graphics
https://www.amd.com/en/products/graphics/radeon-rx-vega-64

[16] The curtain comes up on AMD's Vega architecture
https://techreport.com/review/31224/the-curtain-comes-up-on-amds-vega-architecture/

[17] NVIDIA VOLTA
Architettura GPU Tensor Core, progettata per portare l'intelligenza artificiale in tutti i settori.
https://www.nvidia.com/it-it/data-center/volta-gpu-architecture/

[18] TENSOR CORES IN NVIDIA VOLTA
La nuova generazione del deep learning
https://www.nvidia.com/it-it/data-center/tensorcore/

[19] ACCELERATED COMPUTING AND THE DEMOCRATIZATION OF SUPERCOMPUTING
https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-product-literature/sc18-tesla-democratization-tech-overview-r4-web.pdf

[20] NVIDIA TESLA V100 GPU ARCHITECTURE
THE WORLD'S MOST ADVANCED DATA CENTER GPU
http://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf

[21] TENSOR CORES
https://developer.nvidia.com/tensor-cores

[22] RTX. IT'S ON.
NVIDIA TURING
https://www.nvidia.com/it-it/geforce/turing/

[23] NVIDIA GeForce RTX: analisi dell'architettura delle prime GPU con Ray Tracing
https://www.hwupgrade.it/articoli/skvideo/5254/nvidia-geforce-rtx-analisi-dell-architettura\-delle-prime-gpu-con-ray-tracing_index.html

[24] NVIDIA TURING
La rivoluzione della grafica
https://www.nvidia.com/it-it/design-visualization/technologies/turing-architecture/

[25] NVIDIA QUADRO RTX
La prima GPU al mondo con ray-tracing
https://www.nvidia.com/it-it/design-visualization/quadro-desktop-gpus/

[26] NVIDIA TITAN RTX
https://www.nvidia.com/it-it/titan/titan-rtx/

[27] Manyika, J. (2011). Big data: The next frontier for innovation, competition, and productivity.
http://www.mckinsey.com/Insights/MGI/Research/Technology_and_Innovation/Big_data_The_next_frontier_for_innovation.

[28] Trabuco L. G., Falck E., Villa E., Schulten K. (2007, January). Modeling and simulations of a bacterial ribosome. In BIOPHYSICAL JOURNAL (pp. 571A-571A). 9650 ROCKVILLE PIKE, BETHESDA, MD 20814-3998 USA: BIOPHYSICAL SOCIETY

[29] L. Dematté, D. Prandi: "GPU computing for systems biology", *Briefings in bioinformatics* **11** (2010) 323.

[30] J. Q. Dinh, A. Mahajan, M. B. Palmer, D. R. Grosshans: "Particle therapy for central nervous system tumors in pediatric and adult patients", *Translational Cancer Research* **1** (2012) 137.

[31] G. Pratx, L. Xing: "GPU computing in medical physics: A review", *Medical physics* **38** (2011) 2685.

[32] C.L. Philip Chen, C.-Y. Zhang, *Data-intensive applications, challenges, techniques and technologies: A survey on Big Data*, Inform. Sci. (2014), http://dx.doi.org/10.1016/j.ins.2014.01.015

Andrea D'Urbano: Studente del corso di Laurea in Fisica

Alessandro Fasiello: Studente del corso di Laurea in Ingegneria dell'informazione