



UNIVERSITÀ DI PISA

Laurea Specialistica in Ingegneria Aerospaziale

ANALISI CFD DI UNA VETTURA DA  
COMPETIZIONE CON ARCHITETTURA DI  
ELABORAZIONE IN PARALLELO BASATA SU  
PROCESSORI GRAFICI

Relatori:

PROF. GIOVANNI LOMBARDI

ING. MARCO MAGANZI

DOTT. ALBERTO CIAMPA – INFN PISA

Tesi di laurea di:

**MARCO LEDDA**

Anno Accademico 2013 – 2014

Questa tesi è stata scritta con L<sup>A</sup>T<sub>E</sub>X usando ArsClassica, una rielaborazione dello stile ClassicThesis di André Miede realizzata da Lorenzo Pantieri.

WEBSITE:

<http://www.lorenzopantieri.net/>

*To the biggest stone...*

"Rational arguments don't usually work on religious people.  
Otherwise there would be no religious people."

— Gregory House.

# SOMMARIO

Il presente lavoro di tesi si focalizza sui possibili vantaggi che si possono ottenere nell'utilizzo dei processori grafici nella fluidodinamica computazionale; questo risulta possibile grazie al CUDA, l'architettura di elaborazione in parallelo creata da NVIDIA, che permette di sfruttare la potenza di calcolo delle GPU per scopi differenti dalle normali elaborazioni grafiche.

Punto di partenza della tesi è lo studio effettuato in lavori passati sulla Ferrari 458 Italia GT2. Viene descritto il processo di generazione della *mesh* di superficie, di quella di volume e le loro caratteristiche; successivamente viene illustrato il *setting* del *software* Fluent per l'analisi CFD. Viene fatta una prima campagna di prove in cui si effettua l'analisi CFD del modello della vettura con diverse configurazioni del *cluster*, per poi confrontare i risultati con quelli che si ottengono in una seconda campagna di prove in cui si effettua l'analisi CFD con l'ausilio del CUDA.

# INDICE

## I Parte introduttiva 1

1	INTRODUZIONE	2
1.1	Punto di partenza	2
1.2	Ferrari 458 Italia GT2	3
1.3	Lavoro da svolgere	4
2	CFD	6
2.1	Introduzione alla CFD	6
2.2	Modelli di calcolo	7
2.3	Evoluzione della CFD	8
3	CUDA	10
3.1	CUDA e GPGPU	10
3.1.1	GPU	10
3.1.2	Confronto tra CPU e GPU	13
3.1.3	Evoluzione delle GPU	15
3.1.4	Linguaggi di programmazione	16
3.2	Architettura del CUDA	17
3.2.1	Terminologia del CUDA	17
3.2.2	Hardware	18
3.3	Esempio pratico	19

## II Caso da studiare 21

4	GENERAZIONE DELLA MESH	22
4.1	Mesh di superficie	22
4.2	Mesh di volume	24
5	ANALISI CFD	26
5.1	Coefficienti aerodinamici	26
5.1.1	Efficienza aerodinamica	28
5.1.2	Bilanciamento della vettura	28
5.1.3	Sottogruppi del modello	29
5.2	Setting di Fluent	30
5.3	Condizioni al contorno	34
5.4	Soluzione	34

### III Risultati e conclusioni 36

6	RISULTATI	37
6.1	Confronto fra le architetture	37
6.2	Grandezze da valutare	38
6.3	Cluster	39
6.3.1	48 processori AMD	39
6.3.2	128 processori AMD	41
6.3.3	256 processori AMD	43
6.3.4	512 processori AMD	45
6.4	CUDA	47
6.4.1	48 processori Intel	47
6.4.2	48 processori e 8 GPU	49
7	CONCLUSIONI	52

### Appendice 55

A	REPORT FLUENT	55
A.1	Cluster	55
A.2	CUDA	59
A.3	Setup Fluent	60
	BIBLIOGRAFIA	63

## ELENCO DELLE FIGURE

Figura 1	Ferrari 458 Italia GT2	3
Figura 2	Simulazione CFD Hennessey Venom GT	6
Figura 3	Scheda video nVIDIA Tesla	11
Figura 4	Architettura della GPU della GeForce GTX 280	11
Figura 5	Confronto fra i <i>core</i> delle CPU e delle GPU	13
Figura 6	Confronto fra l'architettura delle CPU e quella delle GPU	14
Figura 7	Confronto fra i FLOPS delle CPU e delle GPU	14
Figura 8	Larghezza di banda della memoria nelle CPU e nelle GPU	15
Figura 9	Linguaggi di programmazione supportati dal CUDA	16
Figura 10	Architettura del CUDA	18
Figura 11	Architettura nVIDIA GeForce 8800 GTX	19
Figura 12	Rappresentazione CAD del modello della vettura	22
Figura 13	<i>Mesh</i> del cofano della vettura	22
Figura 14	Rappresentazione CAD del dominio di calcolo	23
Figura 15	<i>Mesh</i> del dominio di calcolo	24
Figura 16	<i>Mesh</i> in prossimità del modello	24
Figura 17	Sistema di riferimento per la valutazione delle forze	27
Figura 18	Storia di convergenza del coefficiente di resistenza	35
Figura 19	Storia di convergenza del coefficiente di portanza	35
Figura 20	Convergenza $c_x$ – 48 processori AMD	40
Figura 21	Convergenza $c_z$ – 48 processori AMD	40
Figura 22	Convergenza $c_m$ – 48 processori AMD	40
Figura 23	Convergenza $c_x$ – 128 processori AMD	41
Figura 24	Convergenza $c_z$ – 128 processori AMD	42
Figura 25	Convergenza $c_m$ – 128 processori AMD	42
Figura 26	Convergenza $c_x$ – 256 processori AMD	43
Figura 27	Convergenza $c_z$ – 256 processori AMD	44



Figura 28	Convergenza $c_m$ – 256 processori AMD	44
Figura 29	Convergenza $c_x$ – 512 processori AMD	45
Figura 30	Convergenza $c_z$ – 512 processori AMD	46
Figura 31	Convergenza $c_m$ – 512 processori AMD	46
Figura 32	Convergenza $c_x$ – 48 processori Intel	48
Figura 33	Convergenza $c_z$ – 48 processori Intel	48
Figura 34	Convergenza $c_m$ – 48 processori Intel	48
Figura 35	Prestazioni di ANSYS Fluent 15.0	49

## ELENCO DELLE TABELLE

Tabella 1	Scheda tecnica Ferrari 458 Italia GT2	4
Tabella 2	Specifica tecnica di una scheda video nVIDIA Tesla	12
Tabella 3	Confronto fra le specifiche tecniche	13
Tabella 4	Elaborazione di un'immagine di $2048 \times 2048$ <i>pixel</i>	20
Tabella 5	Dimensioni del dominio di calcolo	23
Tabella 6	Proprietà della <i>mesh</i> di volume	25
Tabella 7	Valori di riferimento delle grandezze	27
Tabella 8	Condizioni di assetto del modello	30
Tabella 9	Models	32
Tabella 10	Equations	32
Tabella 11	Numerics	32
Tabella 12	Relaxation	32
Tabella 13	Linear Solver	33
Tabella 14	Pressure–Velocity Coupling	33
Tabella 15	Discretization Scheme	33
Tabella 16	Solution Limits	33
Tabella 17	Risultati delle simulazioni	51

# **Parte I**

## **Parte introduttiva**

# 1 | INTRODUZIONE

## 1.1 PUNTO DI PARTENZA

Il presente lavoro si inserisce nell'ambito di una serie di studi commissionati da Ferrari S.p.A per la valutazione, al variare dell'assetto della vettura e della velocità di prova, degli effetti di interferenza in galleria del vento mediante l'ausilio della CFD<sup>1</sup>; questi studi sono documentati nella tesi *Effetti di interferenza in wind tunnel su una vettura da competizione al variare delle condizioni di prova* [4], punto di partenza di questo lavoro.

La valutazione dei termini correttivi da applicare alle misurazioni effettuate in galleria del vento si basa sul confronto tra i risultati ottenuti in quattro diverse simulazioni:

**WIND TUNNEL (WT)** analisi CFD in galleria del vento con modello in scala 2 : 5

**NO STING (NS)** analisi CFD in galleria del vento privata dei supporti con modello in scala 2 : 5

**FREE AIR (FA)** analisi CFD in aria libera con modello in scala 2 : 5

**FREE AIR, REAL SCALE (FA-RS)** analisi CFD in aria libera con modello in scala 1 : 1

Ciascuna simulazione è caratterizzata dalle seguenti fasi:

1. modifica dell'assetto della vettura e dell'incidenza dell'ala e creazione della *mesh* di superficie tramite il *software* CAD<sup>2</sup> ANSA
2. creazione della *mesh* di volume tramite il *software* TGrid
3. analisi CFD tramite il *software* Fluent con *cluster* a 48 processori

<sup>1</sup> Computational Fluid Dynamics: branca dell'aerodinamica che simula al calcolatore ciò che sperimentalmente viene testato in una galleria del vento.

<sup>2</sup> Computer-Aided Design: famiglia di *software* grafici per supportare l'attività di progettazione e di disegno tecnico.

## 1.2 FERRARI 458 ITALIA GT2



Figura 1: Ferrari 458 Italia GT2

La vettura oggetto dello studio è la Ferrari 458 Italia GT2 [3], riportata in Figura 1, evoluzione *racing* della 458 Italia per le competizioni della classe Gran Turismo.

Dovendo rispettare i regolamenti di ACO<sup>3</sup> e FIA<sup>4</sup>, la potenza massima della vettura è limitata a 465 CV, inferiore di circa 100 CV rispetto a quella della 458 Italia stradale. Altra particolarità, il cambio doppia frizione sette marce di serie viene sostituito sulla GT2 da un sequenziale Hewland a sei rapporti; inoltre, l'utilizzo di ABS<sup>5</sup> e *Traction Control* non è permesso in gran parte delle serie alle quali l'auto partecipa. In Tabella 1 è riportata la scheda tecnica della vettura.

Nonostante le pesanti limitazioni regolamentari su propulsore, elettronica e cambio, i regolamenti consentono ampia libertà di modifiche a sospensioni e aerodinamica. Il risultato è una vettura estremamente precisa e maneggevole, molto sensibile alle regolazioni e con un *grip* eccellente. Infine, anche se dotata di potenza inferiore e caratterizzata da un peso superiore, la 458 Italia GT2 registra tempi sul giro inferiori delle controparti di classe GT3.

3 Automobile Club de l'Ouest: associazione motoristica francese organizzatrice della 24 ore di Le Mans.

4 Fédération Internationale de l'Automobile: federazione di circa 150 Automobile Club nazionali di oltre 100 Paesi del mondo, rappresentanti quasi 100 milioni di automobilisti.

5 Antilock Braking System: sistema di sicurezza che evita il bloccaggio delle ruote dei veicoli garantendone la guidabilità durante le frenate.

Tabella 1: Scheda tecnica Ferrari 458 Italia GT2

DIMENSIONI E PESI		
Lunghezza	(mm)	4518
Larghezza	(mm)	2036
Altezza	(mm)	1160
Passo	(mm)	2650
Carreggiata anteriore	(mm)	1720
Carreggiata posteriore	(mm)	1688
Peso a secco	(kg)	1245
PNEUMATICI		
Anteriori	Pirelli	325/650/18"
Posteriori	Pirelli	325/705/18"
FRENI		
Anteriori	(mm)	380 × 35
Posteriori	(mm)	332 × 32
MOTORE		
Tipo		V8 90°
Alesaggio a corsa	(mm)	94 × 81
Cilindrata totale	(cm <sup>3</sup> )	4498
Potenza massima		465 CV a 6250 rpm
Coppia massima		520 N m a 5750 rpm

### 1.3 LAVORO DA SVOLGERE

Nello sviluppo della tesi si lavorerà esclusivamente con la simulazione *free air, real scale*; il lavoro sarà caratterizzato dalle seguenti fasi:

1. creazione della *mesh* di superficie tramite il *software* ANSA
2. creazione della *mesh* di volume tramite il *software* TGrid
3. prima campagna di prove; analisi CFD tramite il *software* Fluent con le seguenti configurazioni del *cluster*:
  - a) 128 processori
  - b) 256 processori
  - c) 512 processori

in cui verranno valutati gli effetti dell'aumento del numero di processori sul valore dei vari coefficienti e sul tempo necessario per effettuare l'elaborazione

4. seconda campagna di prove; analisi CFD tramite il *software* Fluent con l'ausilio del CUDA<sup>6</sup>, con le seguenti configurazioni di calcolo:
  - a) 48 processori
  - b) 48 processori e 8 GPU

Mentre il *cluster* utilizza macchine con architettura AMD, il CUDA sfrutterà due macchine di architettura Intel; la prima simulazione con 48 processori su queste macchine verrà effettuata per mettere a confronto le due architetture. Successivamente, sfruttando anche le GPU nella simulazione, si verificherà di quanto, grazie alla loro potenza di calcolo, possano effettivamente diminuire i tempi di elaborazione nell'analisi CFD.

In tutte e due le campagne di prove verranno valutati i seguenti parametri:

- coefficiente di resistenza
- coefficiente di portanza
- coefficiente di momento
- efficienza della vettura
- bilanciamento della vettura
- tempo necessario per effettuare l'elaborazione

Alla fine verranno confrontati i risultati ottenuti verificando in questo modo se è effettivamente possibile e conveniente implementare l'utilizzo del CUDA nella fluidodinamica computazionale come avviene in modo soddisfacente in altri ambiti, come illustrato nel [Capitolo 3](#) a pagina 11.

Tutte le simulazioni previste, sia quelle sulle macchine AMD che quelle sulle macchine Intel, verranno effettuate sulle macchine ospitate e mantenute dall'[INFN](#)<sup>7</sup> Pisa nel proprio centro di calcolo.

<sup>6</sup> Compute Unified Device Architecture: architettura per l'elaborazione in parallelo creata da nVIDIA.

<sup>7</sup> Istituto Nazionale di Fisica Nucleare.

# 2 | CFD

## 2.1 INTRODUZIONE ALLA CFD

La CFD è quella branca della fluidodinamica in cui vengono effettuate simulazioni al calcolatore parallelamente alla sperimentazione in galleria del vento; mediante metodi numerici estremamente avanzati vengono risolte le complesse equazioni che governano il moto di un fluido attorno a un corpo.

I principali vantaggi della CFD sono:

- possibilità di modificare facilmente la geometria del problema e le grandezze fluidodinamiche che lo caratterizzano
- possibilità di ottenere in *output* il valore di tutte le variabili fluidodinamiche
- possibilità di confrontare rapidamente i risultati ottenuti nelle varie simulazioni

In [Figura 2](#) si riporta a titolo di esempio la simulazione del flusso d'aria attorno all'autovettura sportiva Hennessey Venom GT.

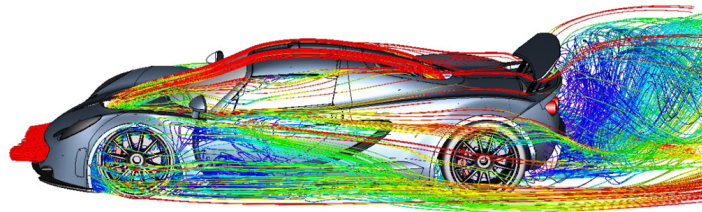


Figura 2: Simulazione CFD Hennessey Venom GT

Per l'analisi CFD nei vari ambiti di studio esistono numerosi *software*, sia commerciali, come:

- CFX
- Fluent
- KIVA

- NUMECA
- STAR-CCM+

sia *open source*, come:

- OpenFOAM
- Code Saturne

## 2.2 MODELLI DI CALCOLO

Il principale compito della CFD è la risoluzione delle equazioni di NAVIER - STOKES in quanto, essendo questa possibile in via analitica solamente in casi molto semplici<sup>1</sup>, per casi reali, caratterizzati dalla presenza di flussi turbolenti, risulta necessario l'approccio numerico. Esistono diversi modelli:

**DIRECT NUMERICAL SOLUTION (DNS)** in cui si discretizzano lo spazio e il tempo con griglie della dimensione voluta e si eseguono i calcoli su di esse; questo modello garantisce i risultati più accurati ma richiede risorse di calcolo elevatissime e quindi non utilizzabile in applicazioni industriali

**REYNOLDS AVERAGED NAVIER-STOKES (RANS)** il moto turbolento viene considerato come somma di una componente media e una fluttuante; le grandezze delle equazioni di partenza vengono mediate in un certo intervallo di tempo riducendo in questo modo i tempi di calcolo in quanto le scale del moto medio risultano essere notevolmente maggiori di quelle del moto turbolento

**LARGE EDDY SIMULATION (LES)** si determina numericamente il comportamento delle scale turbolente più grandi e si modella opportunamente quello delle scale più piccole; è un modello in forte sviluppo in quanto fornisce risultati più accurati delle RANS e ha un costo computazionale notevolmente inferiore a quello della DNS

---

<sup>1</sup> Geometrie semplici con flussi laminari.



## 2.3 EVOLUZIONE DELLA CFD

La CFD comincia la sua storia negli anni '70 grazie all'aumento delle capacità di calcolo dei calcolatori; le prime simulazioni permettevano la risoluzione del flusso transonico attorno a un profilo bidimensionale. Con l'evoluzione della metodologie numeriche e il continuo aumento della potenza di calcolo dei computer, all'inizio degli anni '80 si passò alla risoluzione di problemi supersonici tridimensionali e, alla fine della stessa decade, alla risoluzione di problemi che richiedevano modelli di gas reali.

Negli ultimi anni la CFD ha fatto passi da gigante tanto da assumere un ruolo fondamentale nella fase di progettazione e sviluppo aerodinamico delle vetture a elevate prestazioni, in particolare da competizione, divenendo strumento complementare e per molti versi insostituibile rispetto alla galleria del vento. Comprovata ormai la validità di questo strumento estremamente efficiente, la CFD permette di testare varie configurazioni con tempi e costi inferiori rispetto a quelli che devono essere sostenuti in galleria del vento a seguito della realizzazione e installazione del modello, misurazione delle grandezze e così via. Inoltre la CFD consente di simulare condizioni fisiche difficilmente riproducibili in galleria del vento come:

- analisi termica
- raffreddamento freni alla termica del sottocofano di vetture stradali
- aerodinamica acustica
- analisi di comfort su vetture di serie

Il miglioramento delle prestazioni e della qualità dei risultati offerti dalla CFD è indissolubilmente legato all'evoluzione nel campo sia *hardware* che *software* dei computer, che comporta investimenti costanti per l'aggiornamento del sistema di calcolo e dei programmi utilizzati nelle analisi.

A causa della continua richiesta di aumento della complessità e fedeltà delle simulazioni del fluido, i metodi di generazione delle griglie sono diventati sempre più complessi e sempre maggiori sono le richieste in termini di risorse necessarie per effettuare le elaborazioni. Inoltre, essendo il tempo di elaborazione una delle variabili fondamentali nel campo della CFD, la

diminuzione di questo consente la limitazione dei costi; per tutti questi motivi, per i programmi menzionati precedentemente, risulta fondamentale la possibilità di sfruttare la potenza di calcolo delle GPU<sup>2</sup> delle schede video, operazione che diventa possibile grazie all'architettura di elaborazione in parallelo CUDA, che verrà ampiamente approfondita nel [Capitolo 3](#).

---

<sup>2</sup> Graphics Processing Unit: non è altro che il microprocessore della scheda video.

# 3 | CUDA

## 3.1 CUDA E GPGPU

Il CUDA è un'architettura di elaborazione in parallelo creata da nVIDIA nel 2006 e introdotta con le schede GeForce serie 8, grazie alla quale, sfruttando la potenza di calcolo delle schede video, è possibile aumentare notevolmente le capacità di calcolo dei computer. Tramite il CUDA le GPU possono essere usate non solo in ambito grafico, come avveniva in passato, ma anche in elaborazioni del tutto generali ed estremamente esigenti in termini di potenza di calcolo, per le quali le tradizionali architetture di CPU<sup>1</sup> non hanno una capacità di elaborazione sufficiente; questo approccio prende il nome di GPGPU<sup>2</sup>.

### 3.1.1 GPU

La prima scheda video che ha permesso di sfruttare il CUDA è stata la nVIDIA GeForce 8800 GTX, lanciata nel novembre del 2006; le principali schede video nVIDIA, disponibili ora, compatibili con questa architettura, suddivise in base al settore di utilizzo, sono:

- INTRATTENIMENTO
  - nVIDIA GeForce
  - nVIDIA GeForce Mobile
- DESIGN E PROGETTAZIONE
  - nVIDIA Quadro
  - nVIDIA Quadro Mobile
- ANALISI COMPUTAZIONALI AD ALTE PRESTAZIONI
  - nVIDIA Tesla

---




<sup>1</sup> Central Processing Unit: unità che coordina in maniera centralizzata tutte le altre unità di elaborazione presenti nel computer.

<sup>2</sup> General-Purpose Computing on Graphics Processing Units: settore della ricerca informatica che ha come scopo l'utilizzo della GPU per scopi diversi dalla tradizionale creazione di un'immagine tridimensionale.



Figura 3: Scheda video nVIDIA Tesla

In [Figura 3](#) è riportata una scheda nVIDIA della serie Tesla mentre in [Tabella 2](#) è riportato un esempio di specifica tecnica. La [Figura 4](#) illustra l'architettura della GPU della scheda nVIDIA GeForce GTX 280, in cui si individuano:

-  240 scalar cores
-  On-chip memory
-  Texture units

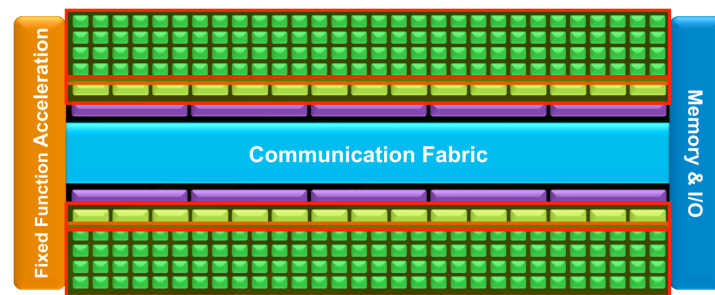


Figura 4: Architettura della GPU della GeForce GTX 280

Grazie alla notevole quantità di GPU compatibili, il CUDA viene utilizzato in numerosi settori quali:

- elaborazione di immagini e video
- videogiochi
- biologia e chimica computazionale

Tabella 2: Specifica tecnica di una scheda video nVIDIA Tesla

CARATTERISTICA	VALORE
Processore	1 × Tesla T10
Numero di core	240
Clock del core	1,296 GHz
FLOPS	933 GFLOPS Single Precision 78 GFLOPS Double Precision
Memoria	4,0 GB
Banda della memoria	picco di 102 GB /s
Memory I/O	512 Bit, 800 MHz GDDR3
Form Factor	Full ATX: 4,736" × 10,5" Dual Slot Wide
System I/O	PCIe ×16 Gen2
Potenza	160 W

- ricostruzione di immagini CT<sup>3</sup>
- analisi sismica
- *Ray Tracing*<sup>4</sup>

L'utilizzo delle GPU per scopi differenti dall'ambito grafico comporta notevoli vantaggi:

**PRESTAZIONI** tali soluzioni offrono un'elevata potenza teorica, e di conseguenza un ridotto tempo di elaborazione, se confrontate con analoghe elaborazioni svolte dalle CPU

**COSTI** il costo di una qualunque GPU è, a parità di fascia di mercato, paragonabile a quello di una CPU ma, offrendo prestazioni teoriche superiori, è possibile grazie ad esse limitare i costi

**AGGIORNAMENTI** la creazione di nuove architetture di GPU è molto più rapida rispetto alle evoluzioni nel campo delle CPU; stessa cosa per quanto riguarda l'aumento della capacità di elaborazione da una generazione all'altra, che risulta superiore per le GPU rispetto alle CPU

<sup>3</sup> La tomografia computerizzata, in radiologia, è una metodica diagnostica per immagini, che sfrutta radiazioni ionizzanti e consente di riprodurre sezioni o strati corporei del paziente ed effettuare elaborazioni tridimensionali.

<sup>4</sup> Tecnica generale di geometria ottica che si basa sul calcolo del percorso fatto dalla luce seguendone i raggi attraverso l'interazione con le superfici. È usata nella modellazione di sistemi ottici come lenti per fotocamere, microscopi, telescopi e binocoli.

**CONSUMO/PRESTAZIONI** nonostante il consumo energetico sia superiore nelle GPU rispetto alle CPU, l'enorme potenza teorica di elaborazione permette l'ottenimento di bassi valori del rapporto consumo/prestazioni

3.1.2 Confronto tra CPU e GPU

Un modo semplice per capire la differenza tra una CPU e una GPU consiste nel mettere a confronto il modo in cui effettuano l'elaborazione dei dati. Come evidenziato in [Figura 5](#), una CPU è costituita da diversi *core* ottimizzati per l'elaborazione seriale sequenziale mentre una GPU è costituita da centinaia di *core* più piccoli e di maggiore efficienza progettati per la gestione di più operazioni in modo simultaneo. In [Figura 6](#) è riportato il confronto fra l'architettura della CPU e quella della GPU, mentre in [Tabella 3](#) è riportato il confronto fra le specifiche tecniche di un processore e di una scheda video.

Tabella 3: Confronto fra le specifiche tecniche

SPECIFICATIONS	Core i7 960	GTX285
Processing Elements	4 cores, 4 way SIMD @3,2 GHz	30 cores, 8 way SIMD @1,5 GHz
Resident Threads (max)	4 cores, 2 threads 4 width SIMD 32 strands	30 cores, 32 SIMD 32 width SIMD 30 720 strands
SP GLOPS	102	1080
Memory Bandwidth	25,6 GB /s	159 GB /s
Register File	—	1,875 MB
Local Store	—	480 kB

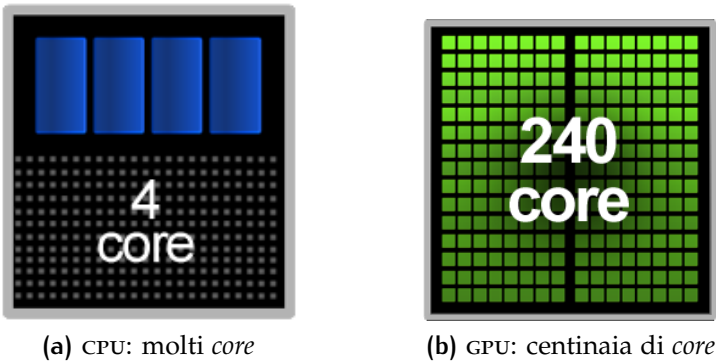


Figura 5: Confronto fra i *core* delle CPU e delle GPU

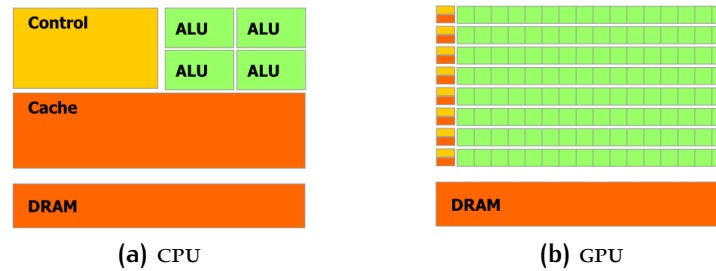


Figura 6: Confronto fra l'architettura delle CPU e quella delle GPU

Il CUDA sfrutta contemporaneamente la potenza di calcolo sia della CPU che della GPU ottenendo in questo modo un modello di elaborazione eterogeneo: la parte sequenziale dell'applicazione viene eseguita sulla CPU mentre la parte a maggiore densità di calcolo viene eseguita sulla GPU. Grazie alla presenza di centinaia di *core*, le GPU permettono la gestione di migliaia di processi contemporaneamente e quindi sono estremamente adatte nel caso di problemi che possono essere espressamente elaborati in parallelo: lo stesso programma viene eseguito da più *data elements* parallelamente, con i *core* che condividono la memoria e un *file* di registro. Dal punto di vista dell'utente l'esecuzione dell'applicazione risulta più rapida perché si evita in questo modo la creazione di grosse *cache* di memoria e l'elevata velocità della GPU viene sfruttata per accelerare notevolmente i tempi di elaborazione dei dati.

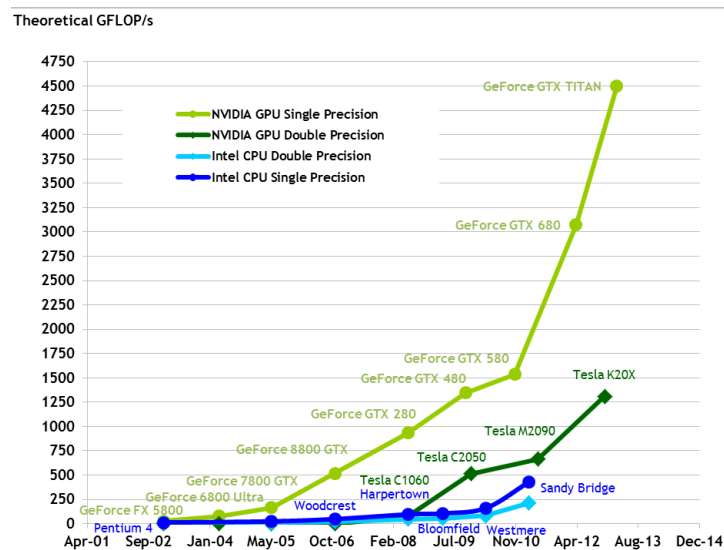


Figura 7: Confronto fra i FLOPS delle CPU e delle GPU

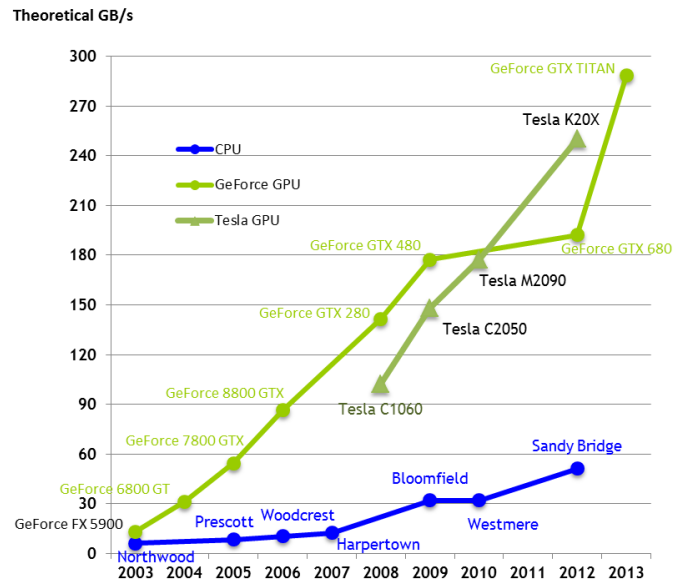


Figura 8: Larghezza di banda della memoria nelle CPU e nelle GPU

### 3.1.3 Evoluzione delle GPU

In Figura 7 è riportato il confronto negli anni fra i FLOPS<sup>5</sup> delle CPU e delle GPU, mentre in Figura 8 è riportato il confronto negli anni fra la larghezza di banda della memoria.

Le nuove GPU con il supporto per il CUDA hanno permesso a chiunque di disporre di strumenti validi per effettuare elaborazioni di dati in parallelo; nel 2007, secondo dati di mercato forniti da nVIDIA, risultavano vendute circa 46 milioni di schede video con GPU compatibili con il CUDA, con il costo dei *developer kit* che si aggirava attorno ai 200\$ per 500 GFLOPS. Il CUDA ha quindi permesso di usufruire di potenze di calcolo presenti in passato esclusivamente nei supercomputer ma a costi accessibili per le masse.

Attualmente l'ultima versione stabile del CUDA è la 6 che permette di accelerare istantaneamente le applicazioni sino a otto volte con la semplice sostituzione delle librerie *CPU-based* esistenti. Le principali caratteristiche di questa versione sono:

**UNIFIED MEMORY** semplifica la programmazione consentendo alle applicazioni di accedere alla memoria della CPU e della GPU senza la necessità di copiare manualmente i dati dall'una all'altra; inoltre, rende più facile implementa-

<sup>5</sup> Floating point Operations Per Second: numero di operazioni in virgola mobile eseguite in un secondo dalla CPU o dalla GPU.



re il supporto per l'accelerazione GPU in una più ampia gamma di linguaggi di programmazione

**DROP-IN LIBRARIES** permettono di accelerare automaticamente i calcoli BLAS e FFTW delle applicazioni fino a otto volte semplicemente sostituendo le librerie per CPU esistenti con le equivalenti GPU-*accelerated*

**MULTI-GPU SCALING** librerie BLAS e FFTW per GPU ridisegnate che scalano automaticamente le prestazioni su un massimo di otto GPU per singolo nodo e offrono oltre 9 TFLOPS di prestazioni in doppia precisione per nodo, sostenendo maggiori carichi di lavoro rispetto al passato (fino a 512 GB)

### 3.1.4 Linguaggi di programmazione supportati

Il CUDA supporta tutti i più importanti linguaggi di programmazione, come evidenziato in [Figura 9](#). Si riportano due esempi di codice: il [Codice 1](#) è un esempio di codice in C standard mentre il [Codice 2](#) è un esempio di codice in C parallelo.




GPU Computing Applications						
Libraries and Middleware						
CUFFT CUBLAS CURAND CUSPARSE	CULA MAGMA	Thrust NPP	VSIP SVM OpenCL	PhysX OptiX	iray	MATLAB Mathematica
Programming Languages						
C	C++	Fortran	Java Python Wrappers	DirectCompute	Directives (e.g. OpenACC)	
CUDA-Enabled NVIDIA GPUs						
Kepler Architecture (compute capabilities 3.x)	GeForce 600 Series	Quadro Kepler Series	Tesla K20 Tesla K10			
Fermi Architecture (compute capabilities 2.x)	GeForce 500 Series GeForce 400 Series	Quadro Fermi Series	Tesla 20 Series			
Tesla Architecture (compute capabilities 1.x)	GeForce 200 Series GeForce 9 Series GeForce 8 Series	Quadro FX Series Quadro Plex Series Quadro NVS Series	Tesla 10 Series			
		 Entertainment	 Professional Graphics	 High Performance Computing		

Figura 9: Linguaggi di programmazione supportati dal CUDA

Oltre a numerosi videogiochi e ai programmi della Creative suite di Adobe, come Adobe Photoshop e Adobe Premier, altri *software* che utilizzano l'architettura CUDA sono:

- **badaboom** - programma di transcodifica video che converte i *file* video in formati eseguibili su dispositivi por-

**Codice 1:** Esempio di codice in C standard

```

void saxpy_serial(int n, float a, float *x, float *y)
{
    for (int i = 0; i < n; ++i)
        y[i] = a*x[i] + y[i];
}
//Perform SAXPY on 1M elements
saxpy_serial(4096*256, 2.0, x, y);

```

**Codice 2:** Esempio di codice in C parallelo

```

__global__
void saxpy_parallel(int n, float a, float *x, float *y)
{
    int i = blockIdx.x*blockDim.x + threadIdx.x;
    if (i < n) y[i] = a*x[i] + y[i];
}
//Perform SAXPY on 1M elements
saxpy_parallel<<<4096, 256>>>(n, 2.0, x, y);

```

tatili; grazie al CUDA è possibile accelerare il processo di conversione rendendolo sino a 18 volte più rapido rispetto ai metodi tradizionali

- **TMPGEnc** - altro *software* di conversione video
- **muveeReveal** - *software* per la creazione di filmati professionali
- **CyberLink PowerDVD** - *software* di *editing* video e DVD *suite*; uno dei primi a supportare completamente la tecnologia CUDA

## 3.2 ARCHITETTURA DEL CUDA

### 3.2.1 Terminologia del CUDA

Nella documentazione nVIDIA viene utilizzata una terminologia speciale di cui si riporta una breve descrizione:

**THREAD** elemento composto dai dati base che devono essere elaborati; questi risultano essere notevolmente più leggeri dei *thread* delle CPU

**WARP** gruppo di 32 *thread*, minima dimensione dei dati elaborati da un multiprocessore CUDA

**BLOCK** blocco che può contenere dai 64 ai 512 *thread*

**GRID** unione di più blocchi

**HOST** termine con cui si fa riferimento alla CPU

**DEVICE** termine con cui si fa riferimento alla GPU

Gli elementi appena descritti sono illustrati in [Figura 10](#).

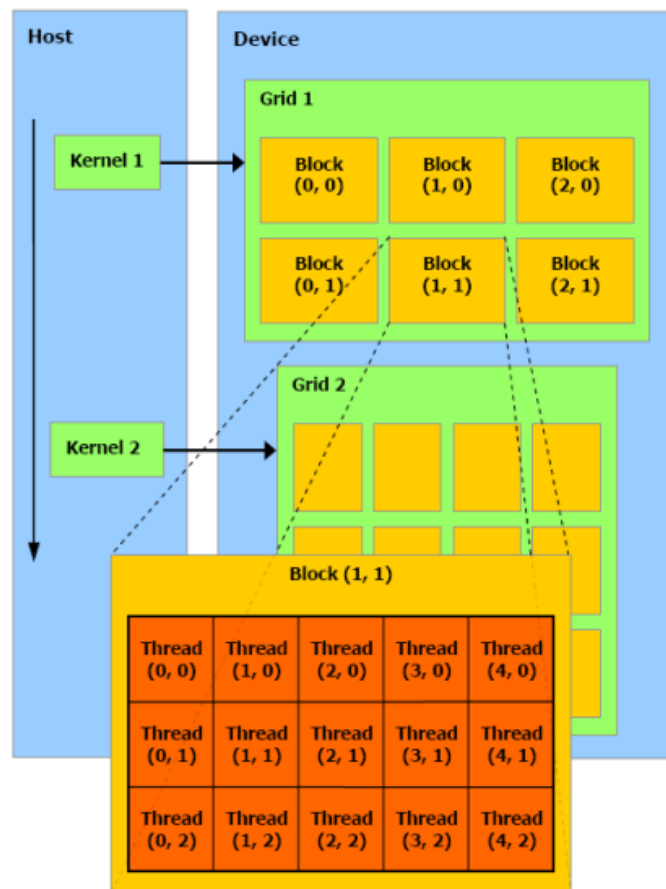


Figura 10: Architettura del CUDA

### 3.2.2 Hardware

Facendo riferimento alla [Figura 11](#), il cuore della scheda video è costituito da diversi *cluster*, in questo caso otto, denominati TPC<sup>6</sup>; ogni *cluster* è a sua volta costituito da un'unità *texture*,

<sup>6</sup> Texture Processor Clusters.

TEX, e da due multiprocessori *streaming*, SM. Questi processori sono costituiti da un *front end* che lancia le istruzioni, e da un *back end* (il quale opera a frequenza doppia rispetto al *front end*), costituito da un gruppo di otto unità di calcolo, SP, e due SFU<sup>7</sup>, che eseguono l'istruzione in modalità SIMT<sup>8</sup>.

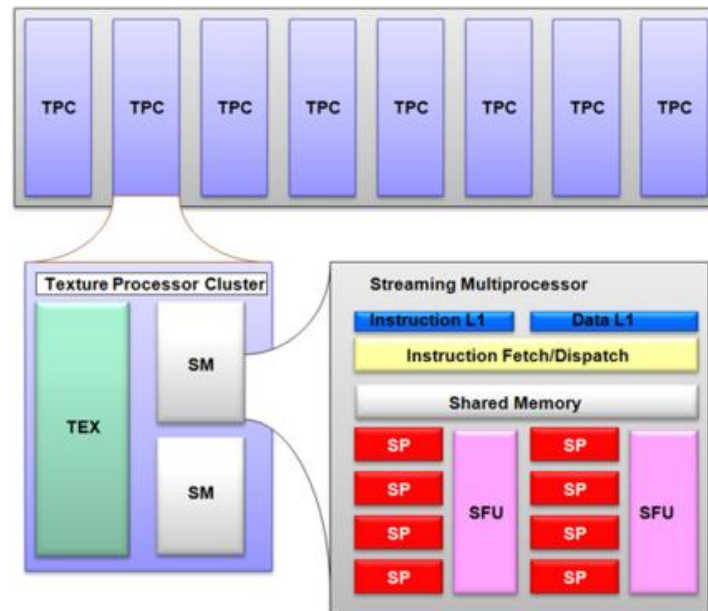


Figura 11: Architettura NVIDIA GeForce 8800 GTX

### 3.3 ESEMPIO PRATICO

Si riportano i risultati ottenuti dal sito [tom's HARDWARE](#) nel confronto fra le prestazioni di una CPU e di una GPU, in cui si utilizza un programma che prende una *heightmap* e ne calcola la corrispondente *normal map*<sup>9</sup>. L'esempio in questione è molto semplice ma permette di avere informazioni per effettuare un confronto fra le prestazioni delle CPU e quelle delle GPU, valutando il tempo necessario per l'elaborazione del processo.

Il notebook utilizzato per effettuare il confronto delle prestazioni ha le seguenti caratteristiche:

- processore Core 2 Duo T5450

<sup>7</sup> Super Function Units.

<sup>8</sup> Single Instruction Multiple Threads.

<sup>9</sup> Per ogni *pixel* dell'immagine iniziale viene applicata una matrice che determinerà il colore del *pixel* risultante dall'immagine generata dai *pixel* adiacenti.

- scheda video nVIDIA GeForce 8600M GT
- sistema operativo Windows Vista

La scelta di un notebook di fascia bassa non è casuale ma viene fatta per dimostrare che si hanno dei vantaggi anche con computer economici e non solo con sistemi equipaggiati di GPU potentissime. In [Tabella 4](#) sono riportati i risultati ottenuti processando un'immagine di  $2048 \times 2048$  *pixel*.

**Tabella 4:** Elaborazione di un'immagine di  $2048 \times 2048$  *pixel*

DEVICE	TEMPO (ms)
CPU 1 thread	1419
CPU 2 threads	749
CPU 4 threads	593
GPU 8600M GT (blocks of 256 pixels)	109
GPU 8600M GT (blocks of 128 pixels)	94
GPU 8600M GTX (blocks of 128/256 pixels)	31

Dai risultati si deduce che la GPU è sempre più veloce; anche una GPU di scarsa qualità risulta essere circa sei volte più veloce di una CPU di buon livello.

Quello precedente è chiaramente un esempio molto semplice in cui risulta agevole l'implementazione del CUDA. L'obiettivo del presente lavoro è quello di capire se è possibile aggiungere definitivamente nell'elenco dei settori che utilizzano il CUDA con profitto, riportato a pagina [11](#), anche la fluidodinamica computazionale o, se ciò non fosse possibile, tentare di capire quali siano i limiti dell'implementazione e come possano essere superati.

# **Parte II**

## **Caso da studiare**

# 4

## GENERAZIONE DELLA MESH

### 4.1 MESH DI SUPERFICIE

Il primo passo è quello della generazione della *mesh* di superficie, operazione che viene eseguita tramite il *software* ANSA. Data la simmetria del problema, non dovendo analizzare configurazioni con l'auto imbardata, si effettua la simulazione su metà dominio di calcolo. In [Figura 12](#) è riportata la rappresentazione CAD del modello della Ferrari 458 Italia GT2.

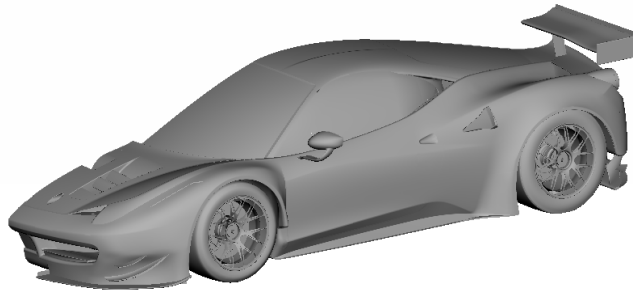


Figura 12: Rappresentazione CAD del modello della vettura

In [Figura 13](#) è riportato un particolare della *mesh* di superficie del cofano del modello della vettura; la *mesh* è realizzata sia con elementi triangolari che con elementi tetragonali.

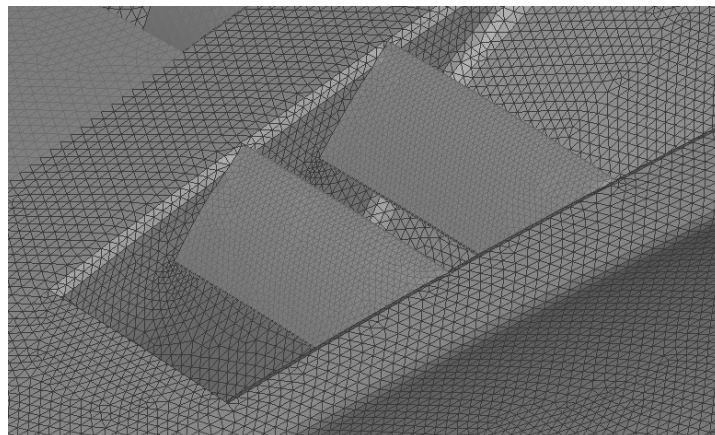


Figura 13: Mesh del cofano della vettura

La simulazione *free air, real scale* ha lo scopo di riprodurre il caso del modello in scala 1 : 1 immerso in un flusso non confinato; tuttavia risulta necessaria la definizione di un dominio chiuso in cui effettuare il calcolo computazionale. Questo dominio è un parallelepipedo di dimensioni tali da non influenzare il campo attorno al modello, in modo tale che per esso il dominio risulti effettivamente infinito. Data la simmetria del problema, così come per la vettura, se ne studia solo la metà.

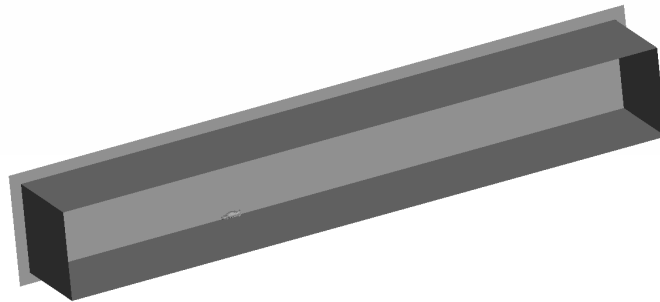
Per la determinazione delle dimensioni del dominio si prende in considerazione il passo  $p$  del modello come lunghezza di riferimento. Il passo della vettura reale, come riportato in [Tabella 1](#) a pagina 4, è pari a:

$$p = 2,65 \text{ m}$$

**Tabella 5:** Dimensioni del dominio di calcolo

GRANDEZZA	VALORE
Lunghezza <sup>1</sup>	$26p = 68,9 \text{ m}$
Larghezza	$10p = 26,5 \text{ m}$
Altezza	$10p = 26,5 \text{ m}$

In [Figura 14](#) è rappresentato il dominio con all'interno il modello della vettura, mentre in [Figura 15](#) è rappresentata la *mesh* di superficie del dominio di calcolo.



**Figura 14:** Rappresentazione CAD del dominio di calcolo

In [Figura 16](#) è riportato il particolare della *mesh* in prossimità del modello, che risulta essere più fitta per meglio rappresentare i gradienti delle varie grandezze in questa zona.

<sup>1</sup> Di cui 10p sono davanti la vettura, 15p dietro.



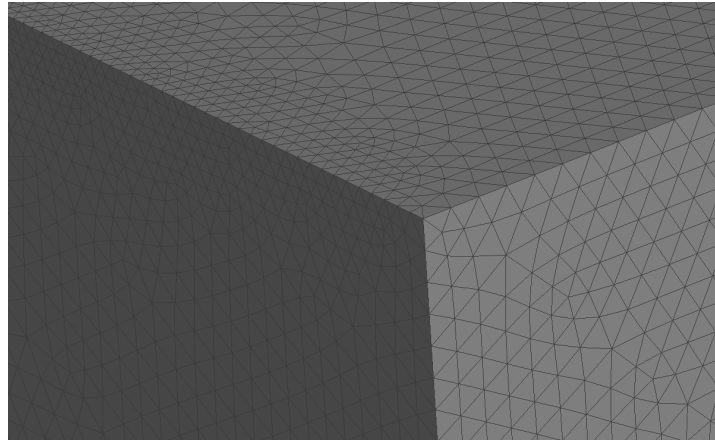


Figura 15: *Mesh* del dominio di calcolo

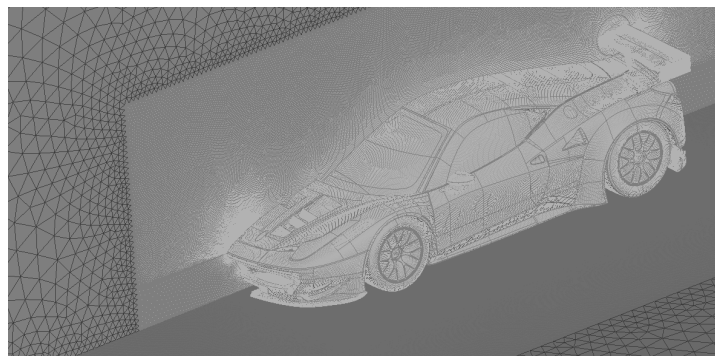


Figura 16: *Mesh* in prossimità del modello

## 4.2 MESH DI VOLUME

Si passa successivamente alla generazione della *mesh* di volume tramite il *software* TGrid. Si utilizza una *mesh* tetraedrica non strutturata ad eccezione del vano racchiuso dai radiatori nel quale si utilizza una griglia prismatica; questa è già presente nel modello di partenza inizialmente fornito, il quale già contiene anche la griglia all'interno dei cerchi delle ruote e in corrispondenza dei dischi dei freni. Le parti del modello già dotate di celle di volume vengono direttamente importate in TGrid e mantenute inalterate durante la generazione della *mesh* tetraedrica.

La *mesh* tetraedrica viene generata aumentando le dimensioni delle celle a partire da quelle di bordo, rispettando i vincoli di dimensione massima imposti e seguendo una progressione geometrica pari a:

$$gf = 1,20$$

dove *gf* è il *growth factor*, valore scelto sulla base di studi di

sensibilità alla griglia effettuati in [4, p.109]. Successivamente si effettua l'operazione di rifinitura in cui si sfrutta un algoritmo basato sulla diminuzione della *Skewness*  $Sk$ , che rappresenta un indice della qualità della griglia<sup>2</sup>, definita come:

$$Sk = \frac{d_i - d}{d}$$

dove:

$d_i$  dimensioni della cella ideale<sup>3</sup>

$d$  dimensioni della cella

Valori di  $Sk$  prossimi a 0 caratterizzano celle di ottima qualità mentre con  $Sk$  che tende a 1 diminuisce la regolarità della cella che tende a diventare un tetraedro degenere con nodi complanari. Una buona qualità della griglia permette di ridurre notevolmente i tempi di calcolo in quanto permette di raggiungere la convergenza della soluzione più rapidamente. In [Tabella 6](#) sono riportate le proprietà della griglia ottenuta, composta da circa 59 milioni di celle.

**Tabella 6:** Proprietà della *mesh* di volume

PARAMETRO	VALORE
Growth Factor	1,20
Numero di celle	~ 59 milioni
Skewness massima	0,899
Skewness media	0,240

<sup>2</sup> Dà informazioni su quanto le celle si allontanano dall'essere tetraedri regolari.

<sup>3</sup> Volume di un tetraedro regolare avente per lato il raggio del cerchio circoscritto al tetraedro in esame.

# 5 | ANALISI CFD

## 5.1 COEFFICIENTI AERODINAMICI

Il passo successivo è quello dell'analisi CFD tramite il *software* Fluent; i coefficienti aerodinamici adimensionali che vengono valutati sono:

- coefficiente di resistenza  $c_x$
- coefficiente di portanza  $c_z$
- coefficiente di momento  $c_m$
- efficienza aerodinamica  $E$
- bilanciamento  $B$

Viene valutato anche il tempo necessario per effettuare ciascuna simulazione, nodo cruciale dell'analisi che si sta effettuando.

Il sistema di riferimento che viene utilizzato per la valutazione dei carichi è riportato in [Figura 17](#), con l'origine nell'intersezione fra l'asse passante per i mozzi delle ruote anteriori e il piano di simmetria del modello; l'asse  $x$  è diretto dall'avantreno al retrotreno, l'asse  $z$  verso l'alto e l'asse  $y$  in modo tale da rendere la terna levogira.

I coefficienti di forza vengono valutati tramite l'espressione generale:

$$c_f = \frac{F}{1/2\rho V^2 S}$$

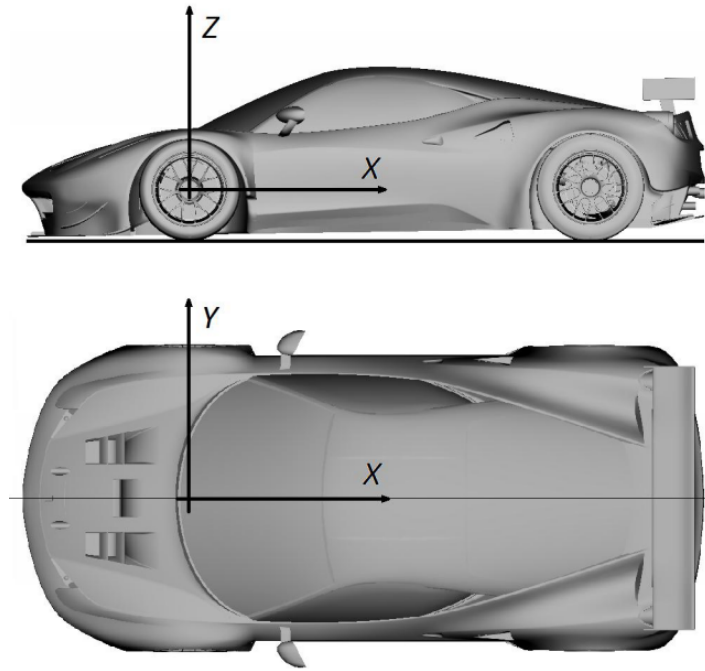
dove:

- $F$  generica componente di forza
- $\rho$  densità di riferimento
- $V$  velocità asintotica
- $S$  sezione frontale della vettura

mentre il coefficiente di momento come:

$$c_m = \frac{M_y}{1/2\rho V^2 S p}$$

dove:



**Figura 17:** Sistema di riferimento per la valutazione delle forze

$M_y$  momento attorno all'asse  $y$   
 $p$  passo della vettura

In [Tabella 7](#) sono riportati i valori di riferimento delle grandezze necessarie per la valutazione dei coefficienti adimensionali.

**Tabella 7:** Valori di riferimento delle grandezze

VARIABILE		VALORE
Area <sup>1</sup>	(m <sup>2</sup> )	1,025
Densità	(kg/m <sup>3</sup> )	1,225
Lunghezza <sup>2</sup>	(m)	2,65
Pressione <sup>3</sup>	(Pa)	0
Temperatura	(K)	288,16
Velocità	(m/s)	45
Viscosità	(kg/m s)	$1,789 \times 10^{-5}$

<sup>1</sup> Superficie frontale del modello della vettura.

<sup>2</sup> Il passo  $p$  del modello.

<sup>3</sup> Il valore 0 indica quanto la pressione si discosta dalla *operating pressure*, posta pari a 101 325 Pa.

### 5.1.1 Efficienza aerodinamica

Ricavati i valori dei vari coefficienti è possibile calcolare l'efficienza aerodinamica come:

$$E = -\frac{c_z}{c_x}$$

che rappresenta il rendimento aerodinamico della vettura; nella formula si introduce il segno meno in modo tale che una vettura con un carico deportante maggiore, a parità di resistenza, abbia un'efficienza maggiore.

### 5.1.2 Bilanciamento della vettura

Per ultimo viene calcolato il bilanciamento, valutabile come:

$$B = \frac{c_{z_a}}{c_z}$$

indice del buon comportamento di guida della vettura e definito come il rapporto tra il coefficiente di portanza sull'anteriore e il coefficiente di portanza globale. Per ricavare il bilanciamento si segue la seguente procedura: il carico aerodinamico totale agente sulla vettura lungo l'asse  $z$ ,  $F_z$ , viene bilanciato dalla reazione del suolo in corrispondenza delle ruote; introducendo il momento di trasporto  $M_y$ , ovvero il momento totale sviluppato dalle forze aerodinamiche lungo l'asse  $y$  calcolato rispetto all'asse anteriore della vettura, si ottiene il seguente sistema:

$$\begin{cases} F_{z_a} + F_{z_p} = F_z \\ F_{z_p} p = -M_y \end{cases}$$

dove:

$F_{z_a}$  reazione vincolare sulle ruote anteriori

$F_{z_p}$  reazione vincolare sulle ruote posteriori

Sviluppando l'espressione delle forze e dei momenti si ottiene:

$$\begin{cases} \frac{1}{2}\rho V^2 S c_{z_a} + \frac{1}{2}\rho V^2 S c_{z_p} = \frac{1}{2}\rho V^2 S c_z \\ \frac{1}{2}\rho V^2 S c_{z_p} p = -\frac{1}{2}\rho V^2 S p c_m \end{cases}$$

e quindi:

$$\begin{cases} c_{z_a} + c_{z_p} = c_z \\ c_{z_p} = -c_m \end{cases}$$

da cui è possibile determinare il coefficiente di portanza sull'anteriore come:

$$c_{z_a} = c_z + c_m$$

### 5.1.3 Sottogruppi del modello

È possibile suddividere il modello in più sottogruppi, ciascuno dei quali contribuisce ai coefficienti aerodinamici globali; i sottogruppi che si possono individuare sono:

**BODY** comprende gli elementi costituenti la parte superiore dell'esterno della vettura, quali:

- cofano
- fiancata
- fiancata posteriore
- tetto
- specchietto
- lunotto posteriore
- parabrezza
- paraurti
- passaruota posteriore
- passaruota anteriore
- base
- sospensione anteriore
- sospensione posteriore

**UNDERBODY** comprende gli elementi costituenti la parte inferiore dell'esterno della vettura, quali:

- fondo anteriore
- fondo posteriore
- fondo intermedio
- diffusore

**WING** comprende gli elementi:

- ala
- sostegno
- paratia laterale

**WHEEL** comprende gli elementi:

- ruota anteriore
- ruota posteriore

- freni anteriori
- freni posteriori

Il programma provvede alla valutazione del contributo ai vari coefficienti di ogni elemento di ciascun sottogruppo costituente il modello della vettura; successivamente si effettua il calcolo dei coefficienti globali semplicemente sommando i vari contributi dei singoli componenti. Questi coefficienti globali non tengono conto del contributo dei radiatori, operazione che in [4] viene eseguita separatamente, non essendo valutato il loro contributo direttamente dal programma, per poi sommare i due valori. In questo lavoro non è importante tanto il calcolo preciso dei coefficienti aerodinamici quanto il confronto dei risultati che si ottengono nelle diverse configurazioni di calcolo, prima senza e poi con l'apporto del CUDA, sia da un punto di vista numerico che da un punto di vista temporale. Per questo motivo si tralascia il calcolo del contributo ai coefficienti dei radiatori.

Negli studi precedenti venivano effettuate delle simulazioni con il modello della vettura in differenti configurazioni di assetto; nel presente lavoro si studia un'unica configurazione, quella che in [4] viene denominata "assetto CFD", caratterizzata dai parametri di assetto riportati in [Tabella 8](#).

**Tabella 8:** Condizioni di assetto del modello

PARAMETRO	VALORE
Altezza della vettura all'anteriore (mm)	23,40
Altezza della vettura al posteriore (mm)	24,60
Incidenza dell'ala (°)	9,38

## 5.2 SETTING DI FLUENT

I parametri da fissare per caratterizzare il problema vengono scelti in base all'esperienza di studi effettuati in passato e alle informazioni ricevute dall'azienda. Vengono fatte le seguenti scelte:

- modello RANS<sup>4</sup> per la risoluzione delle equazioni del moto utilizzando come metodo di chiusura del sistema il  $k - \epsilon$  *realizable*

<sup>4</sup> Introdotto nel [Capitolo 2](#) a pagina 6.

- studio di tipo *steady-state*, date le non stazionarietà presenti nel campo e le oscillazioni da queste causate sulle grandezze di interesse; permette di semplificare e velocizzare l'analisi
- metodo numerico di tipo *pressure-based* data la natura sostanzialmente incompressibile del flusso
  - algoritmo *segregated*, meno esigente dal punto di vista delle risorse rispetto al *coupling*, i quali differiscono per come vengono trattate le equazioni di conservazione della quantità di moto e di continuità
  - valori di *default* per gli *under-relaxation factors*, parametri per il controllo dei residui nel raggiungimento della convergenza della soluzione
  - schema SIMPLEC per l'accoppiamento  $p - v$ ; garantisce maggiore velocità di convergenza rispetto ad altri schemi
- metodo *node-based* di GREEN - GAUSS per la stima dei gradienti, il quale sfrutta i valori nodali assunti da ciascuna variabile
- *non-equilibrium-wall-functions* per risolvere la *k equation*, da preferire data la complessità del problema; permette di prevedere con maggiore accuratezza l'evoluzione del flusso e gli effetti del gradiente di pressione
- schemi di tipo *Second Order Upwind* per la discretizzazione delle derivate delle variabili, in modo da garantire un'accuratezza del second'ordine, evitando inoltre problemi di instabilità numerica dovuti alle elevate variazioni di velocità localizzate nello strato limite. Per la pressione invece si fa uso di schemi standard

Nelle Tabelle dalla 9 alla 16 si riporta il *setting* completo del programma per la simulazione.



Tabella 9: Models

MODEL	SETTINGS
Space	3D
Time	Steady
Viscous	Realizable $k - \varepsilon$ turb. model
Wall Treatment	Non-Equilibrium Wall Fun.
Heat Transfer	Disabled
Solidification and Melting	Disabled
Species	Disabled
Coupled Dispersed Phase	Disabled
NOx Pollutants	Disabled
SOx Pollutants	Disabled
Soot	Disabled
Mercury Pollutants	Disabled

Tabella 10: Equations

EQUATION	SOLVED
Flow	yes
Turbulence	yes

Tabella 11: Numerics

NUMERIC	ENABLED
Absolute Velocity Formulation	yes

Tabella 12: Relaxation

VARIABLE	RELAXATION FACTOR
Pressure	0,3
Density	1,0
Body Forces	1,0
Momentum	0,7
Turbulent Kinetic Energy	0,8
Turbulent Dissipation Rate	0,8
Turbulent Viscosity	1,0

Tabella 13: Linear Solver

VARIABLE	SOLVER TYPE	TERMINATION CRITERION	RESIDUAL REDUCTION TOLERANCE
Pressure	V-Cycle	0,1	—
X-Momentum	Flexible	0,1	0,7
Y-Momentum	Flexible	0,1	0,7
Z-Momentum	Flexible	0,1	0,7
Turb. Kinetic Energy	Flexible	0,1	0,7
Turb. Dissipation Rate	Flexible	0,1	0,7

Tabella 14: Pressure–Velocity Coupling

PARAMETER	VALUE
Type	SIMPLEC
Skewness Correction	0

Tabella 15: Discretization Scheme

VARIABLE	SCHEME
Pressure	Standard
Momentum	Second Order Upwind
Turbulent Kinetic Energy	Second Order Upwind
Turbulent Dissipation Rate	Second Order Upwind

Tabella 16: Solution Limits

QUANTITY	LIMIT
Minimum Absolute Pressure	1
Maximum Absolute Pressure	$5 \times 10^{10}$
Minimum Temperature	1
Maximum Temperature	5000
Minimum Turb. Kinetic Energy	$1 \times 10^{-14}$
Minimum Turb. Dissipation Rate	$1 \times 10^{-20}$
Maximum Turb. Viscosity Ratio	100 000

## 5.3 CONDIZIONI AL CONTORNO

Le condizioni al contorno fissano lo stato del flusso sul bordo del dominio di calcolo; per le varie superfici si sono usate le seguenti condizioni:

**INTERNAL** per le superfici perfettamente permeabili che non offrono quindi nessuna resistenza al passaggio del fluido; vengono trattate in questo modo tutte le superfici che, non viste dal flusso, servono esclusivamente per l'identificazione di volumi chiusi all'interno del modello

**SYMMETRY** per le superfici in corrispondenza di piani di simmetria e per la modellazione dei *zero-shear slip walls* in flussi viscosi

**VELOCITY-INLET** per le superfici attraversate da un flusso, in ingresso o in uscita, con valori della velocità e direzione noti; oltre al valore della velocità e alla sua direzione è necessario definire i parametri di turbolenza

**PRESSURE-OUTLET** per le superfici attraversate da un flusso in uscita con valore della pressione noto; è necessario definire i parametri di turbolenza e il modo in cui trattare l'eventuale flusso reverso, assente a convergenza avvenuta ma presente nelle prime iterazioni

**MOVING-WALLS** per le superfici in movimento rispetto al resto del dominio di calcolo

## 5.4 SOLUZIONE

In base agli studi condotti precedentemente, grazie alla stabilità del risultato che si ottiene, è possibile limitare il numero di iterazioni a 5000 dato che, come riportato sia in [Figura 18](#) che in [Figura 19](#), in cui sono riportate a titolo di esempio le storie di convergenza per i coefficienti di resistenza e di portanza, la convergenza della soluzione la si raggiunge attorno alle 3500 iterazioni. Inoltre, per limitare le incertezze, i valori medi finali delle grandezze vengono valutati considerando esclusivamente le ultime 1000 iterazioni.

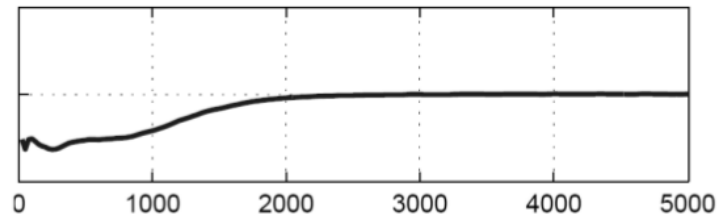


Figura 18: Storia di convergenza del coefficiente di resistenza

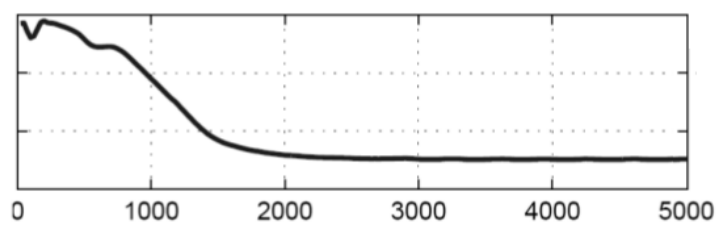


Figura 19: Storia di convergenza del coefficiente di portanza

## **Parte III**

### **Risultati e conclusioni**

# 6 | RISULTATI

## 6.1 CONFRONTO FRA LE ARCHITETTURE

La prima campagna di prove prevede tre simulazioni caratterizzate dalle seguenti configurazioni del *cluster*:

1. 128 processori
2. 256 processori
3. 512 processori

in cui si analizzano gli effetti dell'aumento del numero di processori sul valore dei vari coefficienti adimensionali e sul tempo di elaborazione. I risultati ottenuti vengono confrontati con quelli ottenuti in [4], punto di partenza del lavoro svolto, in cui si effettuava l'analisi CFD con 48 processori. L'architettura *hardware* di queste macchine si basa su tecnologia AMD a 24 *core* e 48 GB di RAM, e le simulazioni vengono fatte utilizzando la versione 14.5 di Fluent.

La seconda campagna di prove prevede due simulazioni con le seguenti configurazioni di calcolo:

1. 48 processori
2. 48 processori e 8 GPU

su due macchine con architettura *hardware* basata su tecnologia Intel a 24 *core* e 32 GB di RAM. La prima simulazione viene effettuata per confrontare le due architetture, AMD e Intel, a parità di numero di processori, 48 e di versione di Fluent, 14.5; le simulazioni, per tutte e due le architetture, girano su due nodi di calcolo interconnessi tramite Infiniband.

Essendo le macchine Intel dotate di un totale di otto schede nVIDIA Tesla K20, la seconda simulazione viene effettuata per verificare i vantaggi che si possono ottenere sfruttando anche la potenza di calcolo delle GPU nell'analisi CFD. Per questo motivo è stata installata la versione 15.0 di Fluent, supportata dal CUDA, a differenza della 14.5 in cui è supportato esclusivamente ANSYS Mechanical.

La versione 15.0 supporta le configurazioni multi-GPU e offre una maggiore produttività nelle simulazioni CFD. Questo miglioramento delle prestazioni è il risultato degli innovativi solutori accelerati sviluppati da nVIDIA, denominati AmgX, in collaborazione con ANSYS. L'attuale implementazione in Fluent 15.0 accelera i solutori di flussi accoppiati basati sulla pressione, accelerando la porzione dedicata al flusso delle simulazioni CFD; per poter sfruttare questo vantaggio basta aggiungere le GPU Tesla K20, Tesla K20X o Tesla K40 a qualsiasi infrastruttura HPC esistente. Questa soluzione è progettata per l'esecuzione su più nodi con più GPU in configurazione *cluster*, proprio come i sistemi basati su CPU.

## 6.2 GRANDEZZE DA VALUTARE

Per ciascuna simulazione vengono valutati:

- coefficiente di resistenza
- coefficiente di portanza
- coefficiente di momento
- efficienza
- bilanciamento
- tempo necessario per effettuare la simulazione

Come anticipato nel Paragrafo 5.4 a pagina 34, per limitare le incertezze, i valori medi finali delle grandezze vengono valutati considerando esclusivamente le ultime 1000 iterazioni.

## 6.3 CLUSTER

### 6.3.1 48 processori AMD

Si riportano i risultati ottenuti in [4]: la simulazione è caratterizzata dalla stessa configurazione sia per le *mesh* di superficie e di volume che per il *setting* di Fluent ma vengono utilizzati 48 processori delle macchine con architettura AMD per la risoluzione delle equazioni del moto. La simulazione viene completata in un tempo pari a:

$$t_{48} = 20 \text{ h } 32 \text{ min } 30 \text{ s}$$

Per il coefficiente di resistenza si ottiene:

$$c_x = 0,522$$

per il coefficiente di portanza si ottiene:

$$c_z = -1,413$$

per il coefficiente di momento si ottiene:

$$c_m = 0,833$$

per l'efficienza si ottiene:

$$E = 2,707$$

mentre per il bilanciamento si ottiene:

$$B = 0,410$$

Nelle Figure dalla 20 alla 22 sono riportate le storie di convergenza dei tre coefficienti aerodinamici.



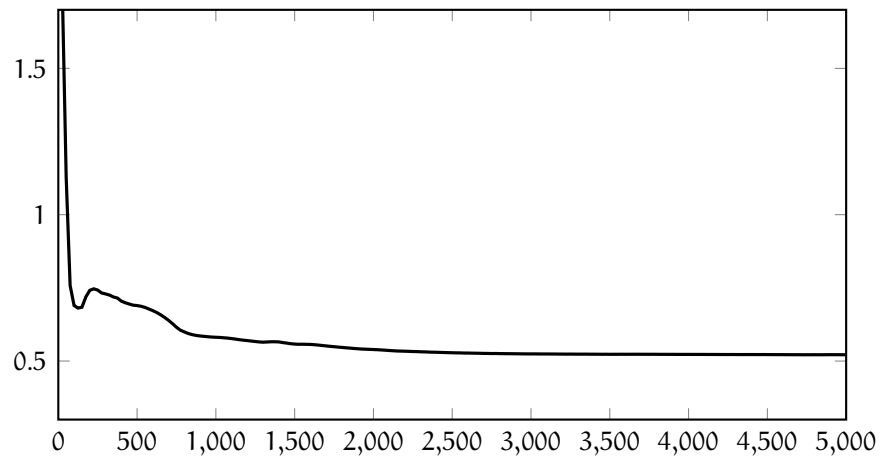


Figura 20: Convergenza  $c_x$  – 48 processori AMD

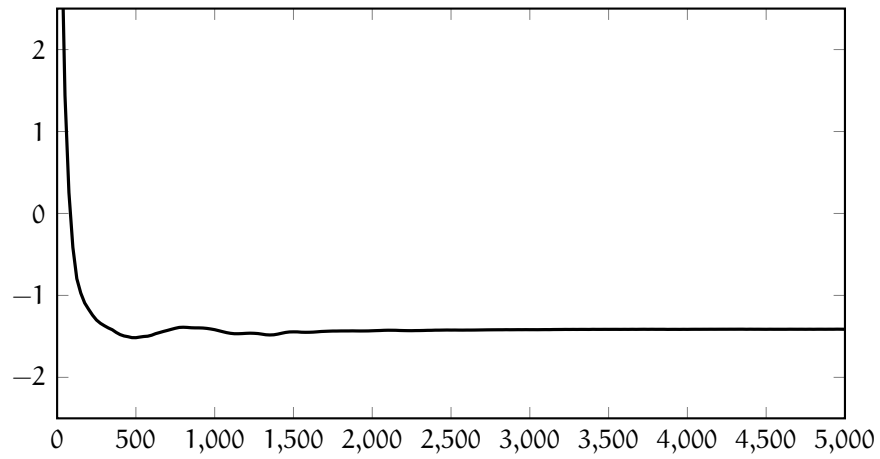


Figura 21: Convergenza  $c_z$  – 48 processori AMD

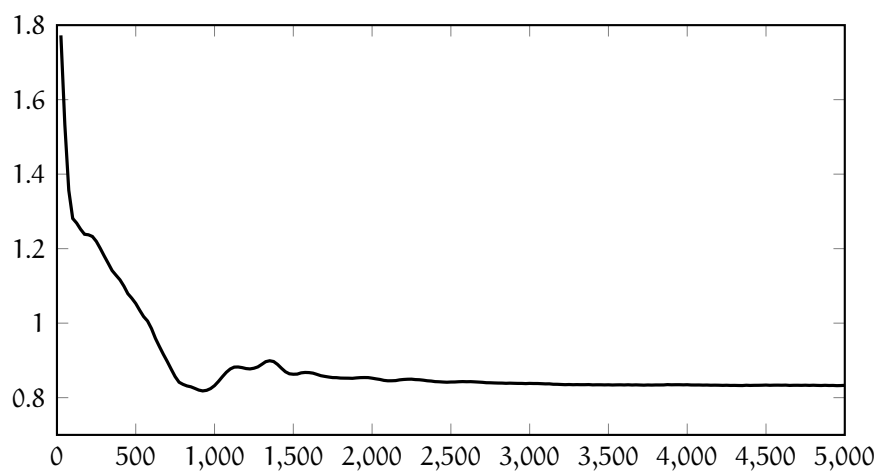


Figura 22: Convergenza  $c_m$  – 48 processori AMD

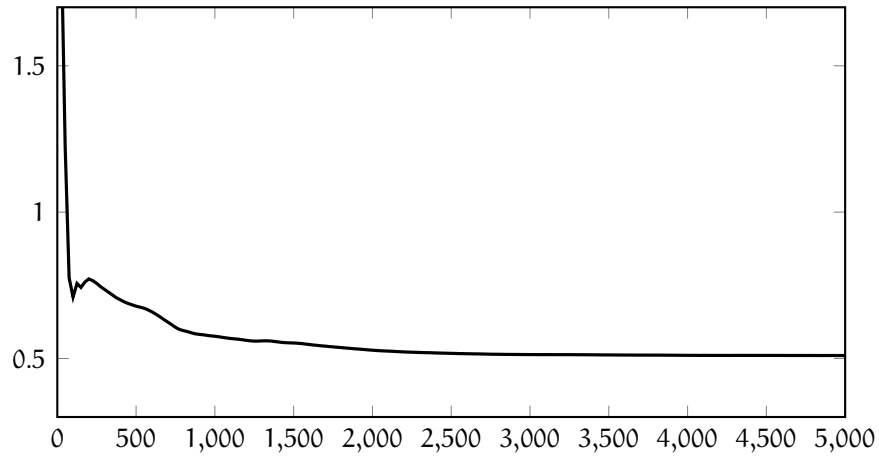


Figura 23: Convergenza  $c_x$  – 128 processori AMD

### 6.3.2 128 processori AMD

La simulazione con 128 processori sulle macchine con architettura AMD viene completata in un tempo pari a:

$$t_{128} = 14 \text{ h } 29 \text{ min } 12 \text{ s}$$

Per il coefficiente di resistenza si ottiene:

$$c_x = 0,510$$

per il coefficiente di portanza si ottiene:

$$c_z = -1,465$$

per il coefficiente di momento si ottiene:

$$c_m = 0,862$$

per l'efficienza si ottiene:

$$E = 2,872$$

mentre per il bilanciamento si ottiene:

$$B = 0,411$$

Nelle Figure dalla 23 alla 25 sono riportate le storie di convergenza dei tre coefficienti aerodinamici.

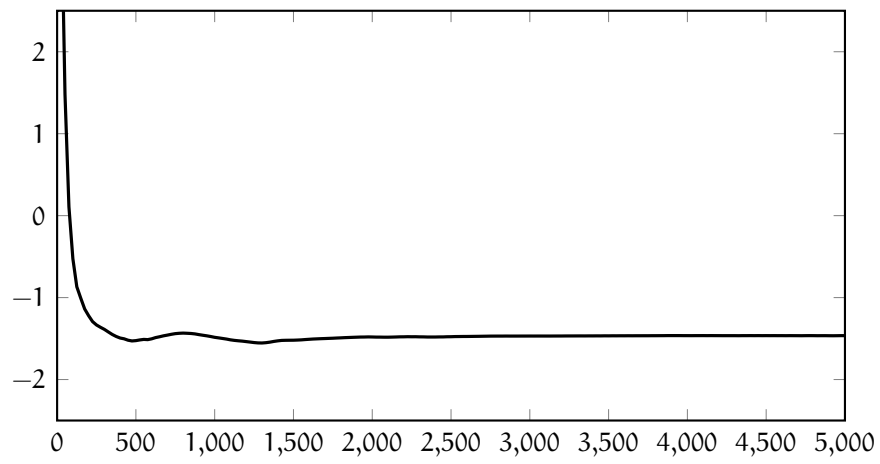


Figura 24: Convergenza  $c_z$  – 128 processori AMD

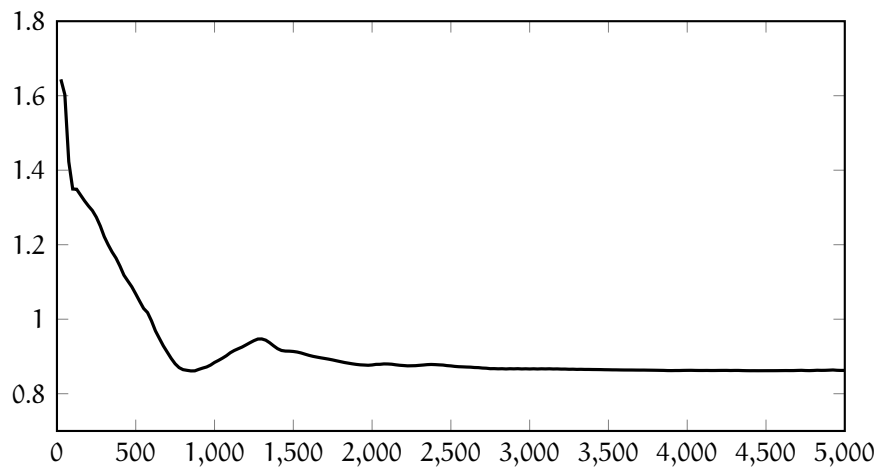


Figura 25: Convergenza  $c_m$  – 128 processori AMD

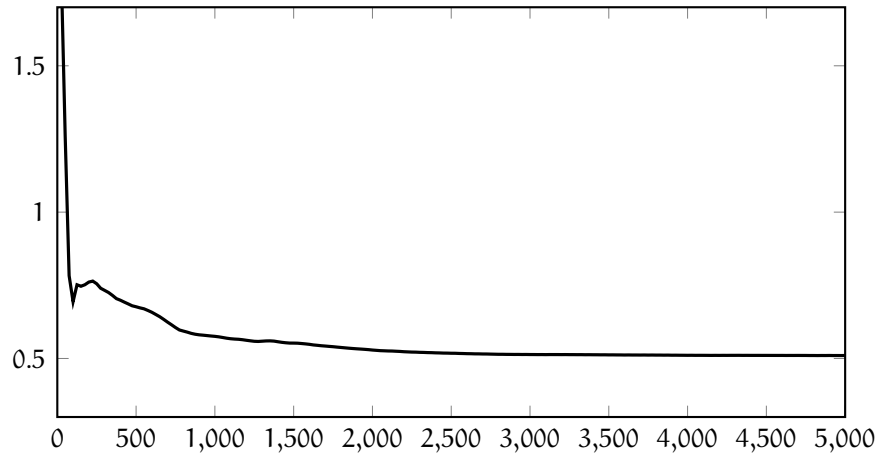


Figura 26: Convergenza  $c_x$  – 256 processori AMD

### 6.3.3 256 processori AMD

La simulazione con 256 processori sulle macchine con architettura AMD viene completata in un tempo pari a:

$$t_{256} = 6 \text{ h } 9 \text{ min } 33 \text{ s}$$

Per il coefficiente di resistenza si ottiene:

$$c_x = 0,510$$

per il coefficiente di portanza si ottiene:

$$c_z = -1,464$$

per il coefficiente di momento si ottiene:

$$c_m = 0,862$$

per l'efficienza si ottiene:

$$E = 2,871$$

mentre per il bilanciamento si ottiene:

$$B = 0,411$$

Nelle Figure dalla 26 alla 28 sono riportate le storie di convergenza dei tre coefficienti aerodinamici.

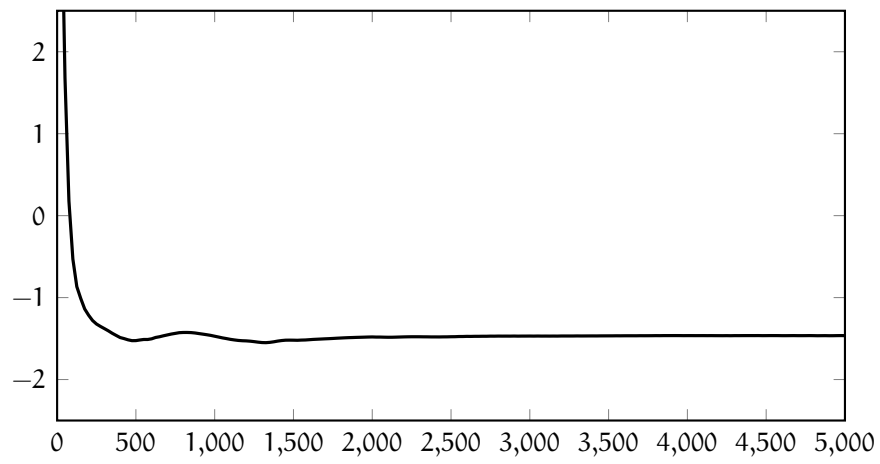


Figura 27: Convergenza  $c_z$  – 256 processori AMD

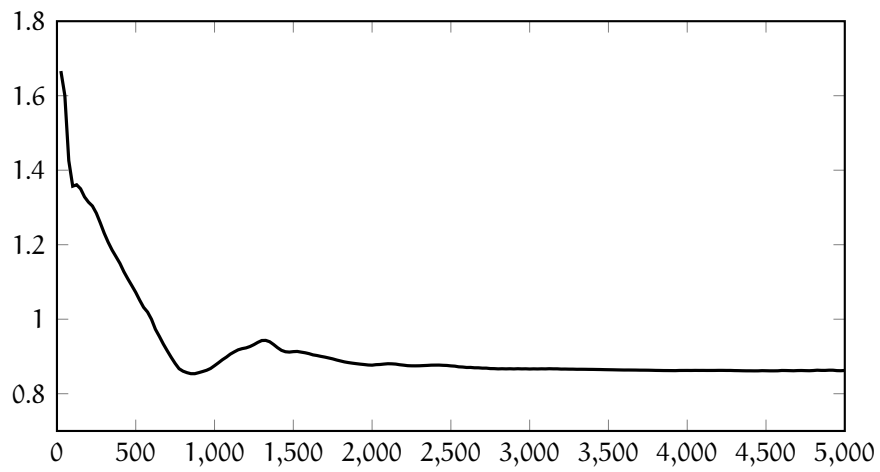


Figura 28: Convergenza  $c_m$  – 256 processori AMD

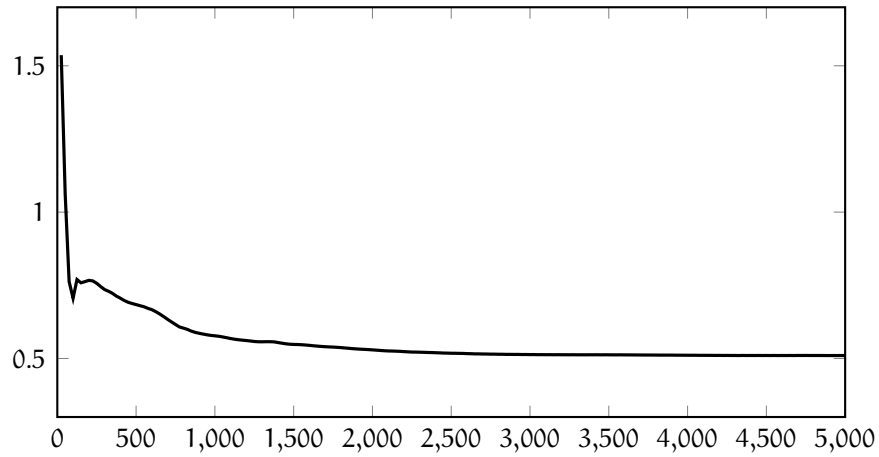


Figura 29: Convergenza  $c_x$  – 512 processori AMD

#### 6.3.4 512 processori AMD

La simulazione con 512 processori sulle macchine con architettura AMD viene completata in un tempo pari a:

$$t_{512} = 3 \text{ h } 59 \text{ min } 12 \text{ s}$$

Per il coefficiente di resistenza si ottiene:

$$c_x = 0,510$$

per il coefficiente di portanza si ottiene:

$$c_z = -1,467$$

per il coefficiente di momento si ottiene:

$$c_m = 0,865$$

per l'efficienza si ottiene:

$$E = 2,876$$

mentre per il bilanciamento si ottiene:

$$B = 0,410$$

Nelle Figure dalla 29 alla 31 sono riportate le storie di convergenza dei tre coefficienti aerodinamici.

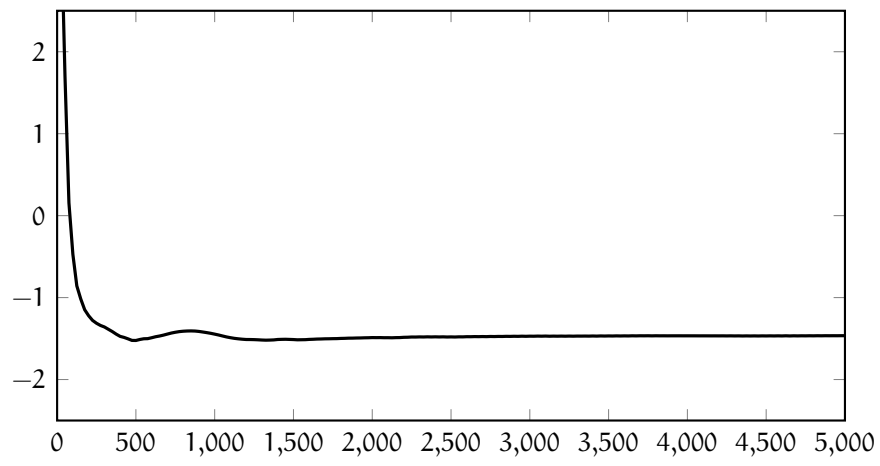


Figura 30: Convergenza  $c_z$  – 512 processori AMD

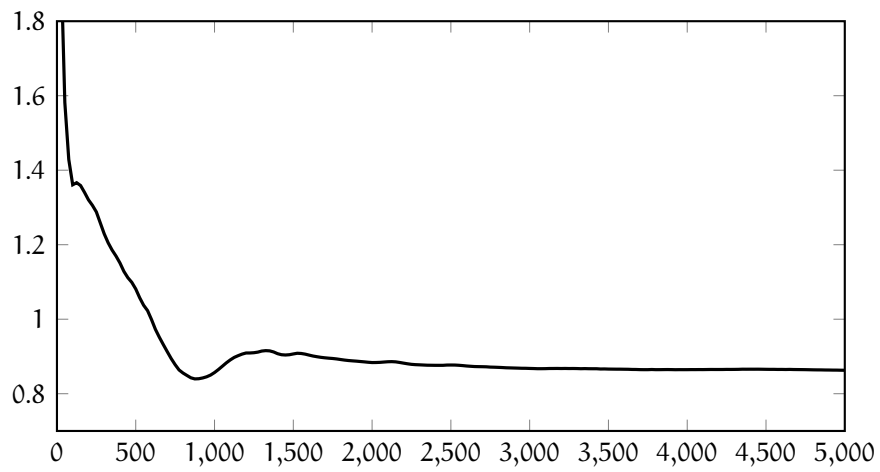


Figura 31: Convergenza  $c_m$  – 512 processori AMD

## 6.4 CUDA

### 6.4.1 48 processori Intel

La simulazione con 48 processori sulle macchine con architettura Intel viene completata in un tempo pari a:

$$t'_{48} = 24 \text{ h } 3 \text{ min } 3 \text{ s}$$

superiore di oltre 3 h al tempo necessario a effettuare la simulazione con lo stesso numero di processori sulle macchine con architettura AMD, come evidenziato nel Paragrafo 6.3.1 a pagina 39. Questo fatto è dovuto principalmente alla notevole differenza in termini di memoria RAM disponibile nelle due architetture, che passa dai 48 GB nelle macchine AMD ai 32 GB nelle macchine Intel.

Per il coefficiente di resistenza si ottiene:

$$c_x = 0,510$$

per il coefficiente di portanza si ottiene:

$$c_z = -1,463$$

per il coefficiente di momento si ottiene:

$$c_m = 0,862$$

per l'efficienza si ottiene:

$$E = 2,867$$

mentre per il bilanciamento si ottiene:

$$B = 0,411$$

Nelle Figure dalla 32 alla 34 sono riportate le storie di convergenza dei tre coefficienti aerodinamici.



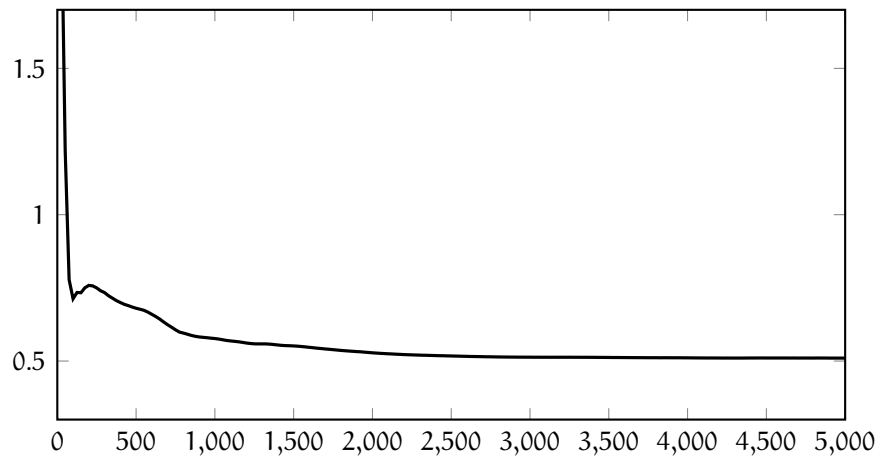


Figura 32: Convergenza  $c_x$  – 48 processori Intel

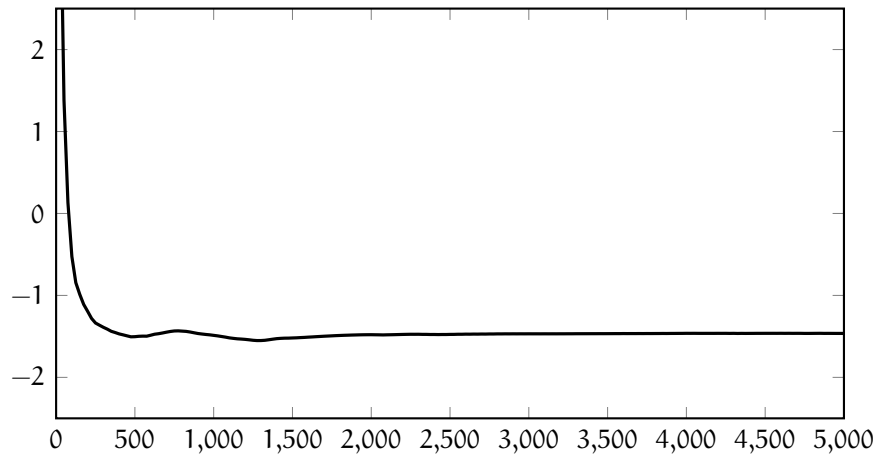


Figura 33: Convergenza  $c_z$  – 48 processori Intel

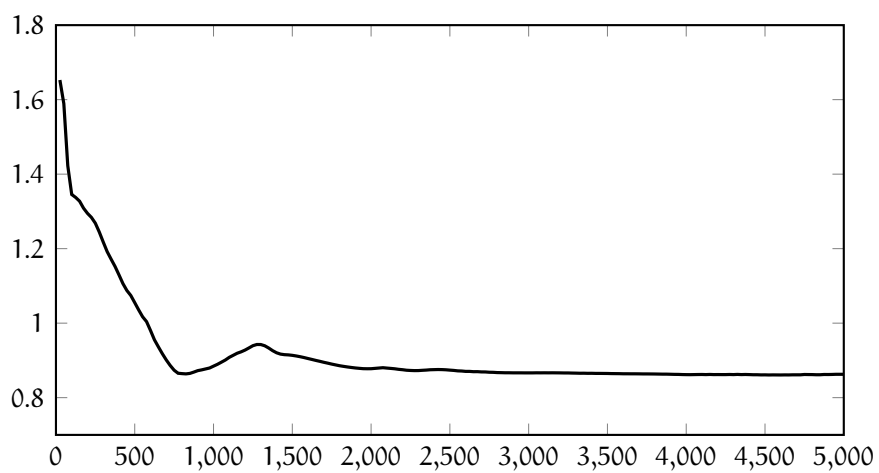


Figura 34: Convergenza  $c_m$  – 48 processori Intel

### 6.4.2 48 processori e 8 GPU

Come anticipato nel Paragrafo 6.1 a pagina 37, essendo le macchine Intel dotate di otto schede nVIDIA Tesla K20, questa simulazione permette di verificare i reali vantaggi che si possono ottenere dall'utilizzo delle GPU insieme alle CPU nelle analisi CFD. Come già detto, questo test ha richiesto l'installazione della versione 15.0 di Fluent, versione supportata dal CUDA.

L'ultima simulazione, a causa di problemi logistici, non è stata possibile effettuarla; essendo l'uso delle GPU richiesto da vari gruppi di studio e lavoro, è possibile usarle in maniera intensiva esclusivamente la notte in una finestra temporale di 10 h, al termine della quale tutti i processi attivi vengono ammazzati, o durante il fine settimana. A causa di notevoli ritardi nella consegna delle licenze di Fluent 15.0, inoltre, si è potuto lavorare con esso solamente per pochissimo tempo e solo il fine settimana, in quanto non sufficiente la finestra temporale presente ogni notte per effettuare una simulazione completa. Questo fatto ha impedito di risolvere i vari problemi che ogni volta sono sorti in seguito al lancio della simulazione, dovuti al non ancora corretto richiamo delle GPU, parallelamente ai processori, per l'analisi CFD. Per questo motivo non è stato possibile effettuare una vera e propria simulazione sfruttando appieno il contributo delle GPU.

Quello che ci si aspettava da questa simulazione era, secondo dati forniti da nVIDIA, un incremento delle prestazioni di un fattore di circa 2 e quindi un tempo necessario per effettuare la simulazione di circa 12/13 h, come riportato in Figura 35, dove vengono confrontate le prestazioni delle GPU Tesla K20 con le CPU nella risoluzione di un problema di aerodinamica con Fluent 15.0.

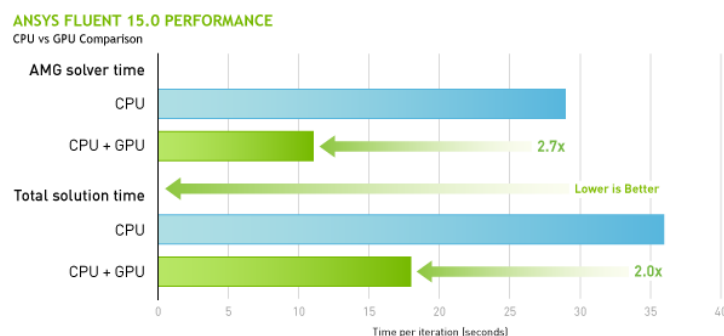


Figura 35: Prestazioni di ANSYS Fluent 15.0

In [Tabella 17](#) sono riassunti i risultati ottenuti nelle varie simulazioni.

Tabella 17: Risultati delle simulazioni

SIMULAZIONE	$C_x$	$C_z$	$C_m$	E	B	TEMPO
Cluster - 48 processori AMD	0,522	-1,413	0,833	2,707	0,410	20 h 32 min 30 s
Cluster - 128 processori AMD	0,510	-1,465	0,862	2,872	0,411	14 h 29 min 12 s
Cluster - 256 processori AMD	0,510	-1,464	0,862	2,871	0,411	6 h 9 min 33 s
Cluster - 512 processori AMD	0,510	-1,467	0,865	2,876	0,410	3 h 59 min 12 s
CUDA - 48 processori Intel	0,510	-1,463	0,862	2,867	0,411	24 h 3 min 3 s
CUDA - 48 processori Intel & 8 GPU	-	-	-	-	-	-

## 7 | CONCLUSIONI

Nel presente lavoro, che si inseriva nell'ambito di una serie di studi commissionati da Ferrari S.p.A, sono stati valutati i possibili vantaggi dell'implementazione del CUDA, l'architettura di elaborazione in parallelo creata da nVIDIA, nell'analisi CFD.

Il punto di partenza è stato il caso *free air, real scale*, cioè analisi CFD in aria libera con modello della vettura in scala 1 : 1, studiato in lavori precedenti e di cui si disponevano i risultati dell'analisi CFD con *cluster* a 48 processori, con l'architettura *hardware* di queste macchine che si basa su tecnologia AMD. Ripartendo dalla creazione della *mesh* di superficie e di quella di volume, si è proceduto a una prima campagna di prove in cui si sono effettuate le analisi CFD tramite il *software* Fluent con le seguenti configurazioni del *cluster*:

1. 128 processori
2. 256 processori
3. 512 processori

verificando gli effetti dell'aumento del numero di processori sui valori dei coefficienti aerodinamici e sul tempo necessario per l'elaborazione. Partendo dalle oltre 20 h necessarie per completare la l'analisi CFD a 48 processori, le tre simulazioni hanno evidenziato una diminuzione dei tempi di elaborazione, passando dalle circa 14 h con 128 processori, alle quasi 4 h con 512 processori; per quanto riguarda i coefficienti invece si è evidenziata una variazione di circa il 2-3%, a seconda del coefficiente, rispetto al caso a 48 processori ma una sostanziale costanza di essi nelle tre simulazioni effettuate.

Successivamente si è proceduto a una seconda campagna di prove in cui si sono effettuate due analisi CFD nelle macchine con architettura *hardware* basata su tecnologia Intel e dotate inoltre di otto schede video nVIDIA Tesla K20, con le seguenti configurazioni di calcolo:

1. 48 processori
2. 48 processori e 8 GPU

la prima per mettere a confronto, a parità di numero di processori, le due architetture AMD e Intel, la seconda per verificare i vantaggi in termini di tempo necessario per l'elaborazione dell'utilizzo delle GPU, parallelamente alle CPU, nell'analisi CFD.

La prima simulazione ha permesso di verificare che la simulazione con le macchine AMD risulta essere di oltre 3 h più veloce, passando dalle 20 h e 30 min con le macchine AMD alle oltre 24 h con le macchine Intel, fatto imputabile alla diversa quantità di memoria RAM installata, 48 GB nelle macchine AMD, 32 GB nelle macchine Intel.

La seconda simulazione, obiettivo principale della tesi, per problemi legati alle licenze e all'utilizzo delle GPU, non è stata possibile completarla; questo fatto ha impedito il confronto dei risultati con quelli ottenuti nelle simulazioni con le macchine AMD, in cui si è aumentata la potenza di calcolo aumentando il numero di CPU, e con quelli ottenuti nella simulazione con le macchine Intel, in cui, sfruttando anche le 8 schede video a supporto dei 48 processori, ci si aspettava una diminuzione del tempo di elaborazione di un fattore di circa 2, secondo dati forniti da nVIDIA, e quindi un tempo di elaborazione di circa 12/13 h.

# Appendice

# A | REPORT FLUENT

## A.1 CLUSTER

48 processori AMD

Started at Fri Jan 3 19:41:09 2014

Results reported at Sat Jan 4 16:13:39 2014

Your job looked like:

```
-----  
# LSBATCH: User input  
/afs/pi.infn.it/pisw/Fluent/14.5/v145/fluent/bin/fluent 3d -  
t48 -ssh -pmyrinet -g -i FA_RS_launcher.jou -lsf  
-----
```

Successfully completed.

Resource usage summary:

CPU time :	294568.00 sec.
Max Memory :	7353 MB
Average Memory :	5269.52 MB
Total Requested Memory :	-
Delta Memory :	-
(Delta: the difference between total requested memory and actual max usage.)	
Max Swap :	9278 MB
Max Processes :	28
Max Threads :	48



## 128 processori AMD

---

Started at Tue Apr 15 20:55:56 2014

Results reported at Wed Apr 16 11:25:08 2014

Your job looked like:

```
-----
# LSBATCH: User input
/afs/pi.infn.it/pisw/Fluent/14.5/v145/fluent/bin/fluent 3d -
  t128 -ssh -pinfiniband -g -i FA_launcher.jou -lsf
-----
```

Successfully completed.

Resource usage summary:

```

CPU time :                415201.00 sec.
Max Memory :              6597 MB
Average Memory :          5269.52 MB
Total Requested Memory : -
Delta Memory :            -
(Delta: the difference between total requested memory
and actual max usage.)
Max Swap :                9892 MB

Max Processes :           37
Max Threads :             53

```

---

---

 256 processori AMD
 

---

Started at Thu Apr 24 09:15:05 2014

Results reported at Thu Apr 24 15:24:38 2014

Your job looked like:

```
-----
# LSBATCH: User input
/afs/pi.infn.it/pisw/Fluent/14.5/v145/fluent/bin/fluent 3d -
t256 -ssh -pinfiniband -g -i FA_launcher.jou -lsf
-----
```

Successfully completed.

Resource usage summary:

```

CPU time :                175614.00 sec.
Max Memory :              4914 MB
Average Memory :          3627.65 MB
Total Requested Memory : -
Delta Memory :            -
(Delta: the difference between total requested memory
and actual max usage.)
Max Swap :                9518 MB

Max Processes :           55
Max Threads :             71

```

---

## 512 processori AMD

---

Started at Thu Apr 24 16:44:17 2014

Results reported at Thu Apr 24 20:43:29 2014

Your job looked like:

```
-----
# LSBATCH: User input
/afs/pi.infn.it/pisw/Fluent/14.5/v145/fluent/bin/fluent 3d -
t512 -ssh -pinfiniband -g -i FA_launcher.jou -lsf
-----
```

Successfully completed.

Resource usage summary:

```

CPU time :                112837.00 sec.
Max Memory :              5509 MB
Average Memory :          4209.14 MB
Total Requested Memory : -
Delta Memory :            -
(Delta: the difference between total requested memory
and actual max usage.)
Max Swap :                10958 MB

Max Processes :           64
Max Threads :             80

```

---

## A.2 CUDA

48 processori Intel

Started at Wed Jun 4 19:02:32 2014

Results reported at Fri Jun 6 05:05:35 2014

Your job looked like:

-----  
# LSBATCH: User input

/afs/pi.infn.it/pisw/Fluent/14.5/v145/fluent/bin/fluent 3d -  
t48 -ssh -pmyrinet -g -i FA\_RS\_launcher.jou -lsf  
-----

Successfully completed.

Resource usage summary:

CPU time :	2046627.00 sec.
Max Memory :	30804 MB
Average Memory :	29662.56 MB
Total Requested Memory :	-
Delta Memory :	-
(Delta: the difference between total requested memory and actual max usage.)	
Max Swap :	43182 MB
Max Processes :	40
Max Threads :	72

## A.3 SETUP FLUENT

### Fluent Settings

Fluent

Version: 3d, pbns, rke (3d, pressure-based, realizable k-epsilon)

Release: 14.5.7

Title:

Models

-----

Model	Settings
-----	
Space	3D
Time	Steady
Viscous	Realiz. k-eps. turb. model
Wall Treatment	Non-Equilib. Wall Fun.
Heat Transfer	Disabled
Solidif. and Melting	Disabled
Species	Disabled
Coupled Disp. Phase	Disabled
NOx Pollutants	Disabled
SOx Pollutants	Disabled
Soot	Disabled
Mercury Pollutants	Disabled

Material Properties

-----

Material: air (fluid)

Property	Units	Method	Value(s)
-----			
Density	kg/m3	constant	1.225
Cp	j/kg-k	constant	1006.43
Thermal Conduct.	w/m-k	constant	0.0242
Viscosity	kg/m-s	constant	1.7894e-05
Molecular Weight	kg/kgmol	constant	28.966
Thermal Exp. Coeff.	1/k	constant	0
Speed of Sound	m/s	none	#f

Material: aluminum (solid)

Property	Units	Method	Value(s)
-----			
Density	kg/m <sup>3</sup>	constant	2719
Cp (Specific Heat)	J/kg-K	constant	871
Thermal Conductivity	W/m-K	constant	202.4

### Solver Settings

#### Equations

Equation	Solved
-----	
Flow	yes
Turbulence	yes

#### Numerics

Numeric	Enabled
-----	
Absolute Velocity Formulation	yes

#### Relaxation

Variable	Relaxation Factor
-----	
Pressure	0.3
Density	1
Body Forces	1
Momentum	0.7
Turbulent Kinetic Energy	0.8
Turbulent Dissipation Rate	0.8
Turbulent Viscosity	1

#### Linear Solver

Variable	Solver Type	Termin. Criterion	Residual Red. Tolerance
-----			
Pressure	V-Cycle	0.1	
X-Momentum	Flexible	0.1	0.7
Y-Momentum	Flexible	0.1	0.7
Z-Momentum	Flexible	0.1	0.7
Turb. Kin. Energy	Flexible	0.1	0.7
Turb. Diss. Rate	Flexible	0.1	0.7

## Pressure-Velocity Coupling

Parameter	Value
-----	
Type	SIMPLEC
Skewness Correction	0

## Discretization Scheme

Variable	Scheme
-----	
Pressure	Standard
Momentum	Second Order Upwind
Turbulent Kinetic Energy	Second Order Upwind
Turbulent Dissipation Rate	Second Order Upwind

## Solution Limits

Quantity	Limit
-----	
Minimum Absolute Pressure	1
Maximum Absolute Pressure	5e+10
Minimum Temperature	1
Maximum Temperature	5000
Minimum Turb. Kinetic Energy	1e-14
Minimum Turb. Dissipation Rate	1e-20
Maximum Turb. Viscosity Ratio	100000

## BIBLIOGRAFIA

- [1] Inc. ANSYS. *Fluent 14.0 User's Guide*. 2011.
- [2] Inc. ANSYS. *TGrid 5.0 User's Guide*. 2008.
- [3] 458 Italia GT2. 2013. URL: [http://www.ferrari.com/Italian/GT\\_Sport\\_Cars/Sport-Prototipi/Pages/Scheda\\_Tecnica\\_458\\_GT2.aspx](http://www.ferrari.com/Italian/GT_Sport_Cars/Sport-Prototipi/Pages/Scheda_Tecnica_458_GT2.aspx).
- [4] B Baldacchini M. and Magurno. *Effetti di interferenza in wind tunnel su una vettura da competizione al variare delle condizioni di prova*. Pisa, 2013.
- [5] G. Buresti. *Elements of Fluid Dynamics*. London: Imperial College Press, 2012.
- [6] G. Lombardi. *Appunti del corso di Aerodinamica degli Aeromobili*. Pisa, 2011.
- [7] BETA CAE Systems S.A. *Ansa 15.0 User's Guide*. 2013.



# RINGRAZIAMENTI

P.S. I ringraziamenti sono scritti in bianco.