

PARALLEL HIGH PERFORMANCE COMPUTING
CdS magistrale in informatica

1 – INTRODUZIONE
Dipartimento di Matematica e Applicazioni
Universita' degli Studi di Napoli Federico II

wpage.unina.it/lapegna

1

Marco Lapegna
Parallel High Performance Computing
1 – Introduzione

metodi dell'indagine scientifica

Metodi tradizionali

- Teoria (risoluzione carta e penna di equazioni)
- Sperimentazione (costruzione di prototipi)

Limitazioni:

- Troppo difficile (es: costruire grandi gallerie del vento)
- Troppo costoso (es: costruzione di prototipi usa e getta)
- Troppo lento (es: studio di evoluzione di galassie o clima)
- Troppo pericoloso (es: armi, virus, esperimenti climatici)

Terza metodologia: Simulazione (uso di calcolatori)

2

Marco Lapegna
Parallel High Performance Computing
1 – Introduzione

computational science

Studio dei fenomeni scientifici attraverso
la simulazione computazionale

3 distinte competenze
interdisciplinari

Fisica,
Chimica,
Biologia ...

risultati

Descrizione
del problema

matematica

Scienza
computazionale

informatica

Risoluzione
del problema

3

Marco Lapegna
Parallel High Performance Computing
1 – Introduzione

Esempio: previsioni meteorologiche

calcolo delle variabili (temperatura,
vento, umidità, ...) in una **griglia di
punti** mediante sofisticato modelli
matematici

- **Superficie** = 100 milioni di Km²
- **Altezza** s.l.m. = 10 Km
- **Discretizzazione** (distanza tra i
punti) = 100m (1Km³ = 10³ punti)

$$100 \times 10^6 \text{ Km}^2 \times 10 \text{ Km} = 10^9 \text{ Km}^3$$

$$= 10^9 \times 10^3 = 10^{12} \text{ punti}$$


4

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

esempio (cont.)

previsioni per i prossimi 2 giorni (48 ore)

ogni ora di simulazione in ogni cubo necessarie
 10^4 operazioni floating point (flop)



Intera superficie = $10^{12} \times 10^4 \times 48 =$
 $\sim 5 \times 10^{17}$ operazioni

Calcolatore da
1 Tflops (1×10^{12} op/sec)
 ~ 500000 sec (circa 150 ore)

Calcolatore da
100 Tflops (100×10^{12} op/sec)
 ~ 5000 sec (circa 1.5 ore)

5

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

esempio (cont.)

Con un calcolatore da 200 Tflops e' possibile

Impiegare 2500 sec

↓

Ridurre i tempi

Previsioni su 4 giorni

↓

Aumentare la dimensione del problema

**Obiettivi della computational science:
Risolvere problemi piu' complessi
in minor tempo**

6

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

Computational Science

campi di applicazione

Scienza

- Modelli climatici
- Modelli astrofisici
- Biologia: proteine, nuovi farmaci
- Nuovi materiali

Ingegneria

- Crash test
- Terremoti e modelli strutturali
- Disegno di aeroplani
- Disegno di motori

Economia

- Modelli finanziari
- Motori di ricerca
- Data mining

Altro

- Simulazione di test nucleari
- Crittografia


Per tali problemi sono richieste
potenze di calcolo di almeno
1 Pflops (10^{15} op/sec)

7

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

i supercalcolatori

- Un supercalcolatore e' un **unico sistema** (hardware e software) capace di fornire una potenza di calcolo paragonabile a quella massima che e' possibile ottenere con la **corrente tecnologia**
- E' una definizione che **dipende dal tempo** (Un supercomputer di oggi e' un computer ordinario di domani)
- Nell'accezione comune un calcolatore e' un supercomputer se rientra nella **Top500 list**.
 - Lista aggiornata 2 volte all'anno
 - Performance misurata con risoluzione di sistemi di equazioni lineari (Linpack benchmark)
 - www.top500.org



8

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

top 500 list: novembre 2022

1	Frontier - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	LUMI - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	Leonardo - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,463,616	174.70	255.75	5,610
5	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096

9

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

ambiti di utilizzo

Segments System Share

Segments	Count
1 Industry	274
2 Research	103
3 Academic	66
4 Government	34
5 Others	14
6 Vendor	9

Oltre la meta' dei supercomputers nella Top500 list e' **usato nell'industria** !!

10

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

come e' fatto un supercomputer (1)

Architecture System Share

Architecture	Count
1 Cluster	492
2 MPP	37

Oltre il 90% dei supercomputers nella Top500 list sono **cluster**!!

11

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

i clusters

Un cluster e' un gruppo di **computer connessi da una rete dedicata o switch** ad alta velocita', che lavorano allo stesso problema, dando l'impressione di una unica risorsa di calcolo

Ogni nodo ha una sua **propria memoria** e comunica con gli altri nodi attraverso **espliciti scambi di messaggi**

Esistono cluster con migliaia di nodi

E' un esempio di calcolatore parallelo MIMD a memoria **distribuita**

12

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

problemi dei cluster

Comunicazione tra i nodi:

- E' necessario trasferire dati in maniera esplicita con apposite librerie (MPI, PVM,..)

Comunicazione

- Rischio di nodi inattivi in attesa di messaggi

Rete o switch:

- i nodi sono molto piu' veloci delle reti:
- Esempio Core I7 5950 (Broadwell, 2015)
 - A 3 GHz i 4 core possono eseguire **48 op.f.p. in un nanosecondo**
 - Una rete Infiniband a 120 Gbit/sec puo' trasferire **4 dati reali tra i nodi in un nanosecondo**

13

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

come e' fatto un supercomputer (2)

Cores per Socket System Share

Cores per Socket	Count
1	20
2	24
3	16
4	18
5	12
6	14
7	64
8	8
9	10
10	28
11	48
12	68
13	32
14	22
15	260
16	4
17	6
Others	1

tot. = 500

Tutti i supercomputers nella Top500 list usano CPU **multicore!!**

14

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

CPU multicore

Una CPU multicore e' un insieme di unita' processanti autonome in un unico chip, che condividono risorse (cache, bus, e memoria centrale)

I core possono condividere la cache o avere cache proprie

Oggi si hanno al piu' 16-18 core

E' un esempio di calcolatore parallelo MIMD a memoria **condivisa**

Intel Broadwell (2015)

15

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

problemi delle CPU Multicore

Comunicazione tra i core:

- Non c'e' comunicazione diretta ma solo attraverso le cache o la memoria centrale (risorse condivise)

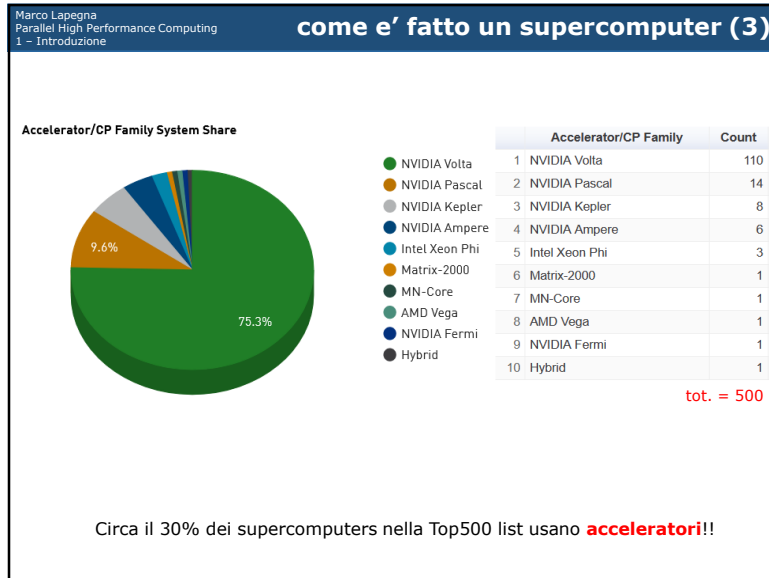
Sincronizzazione

- Rischio di core inattivi

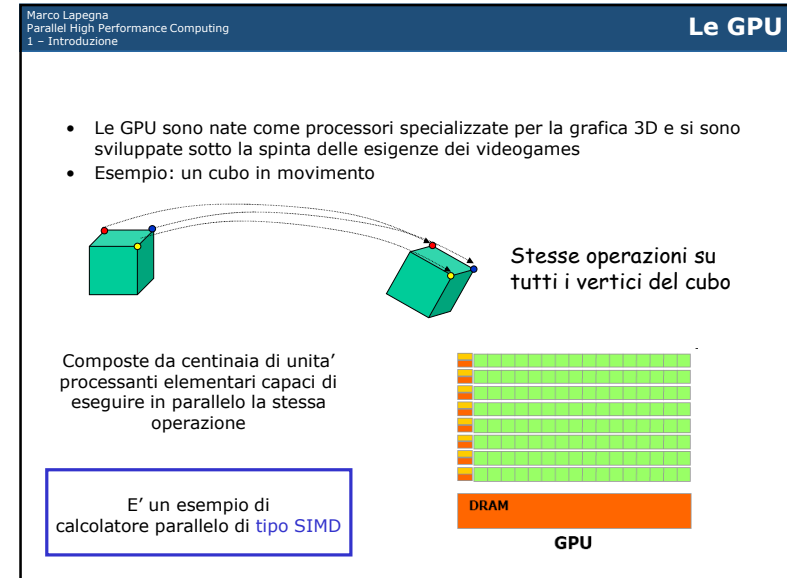
Bus verso la memoria:

- i core sono molto piu' veloci della memoria:
- Esempio Core I7 5950 (Broadwell, 2015)
 - A 3 GHz i 4 core possono eseguire **48 op.f.p. in un nanosecondo**
 - A 10.5 GB/sec il bus puo' trasferire solo **2.5 dati reali alla CPU in un nanosecondo**
 - Essenziale un uso efficiente delle cache

16



17



18

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

problemi delle GPU

Efficiente in casi particolari dove e' necessario eseguire le stesse operazioni su molti dati

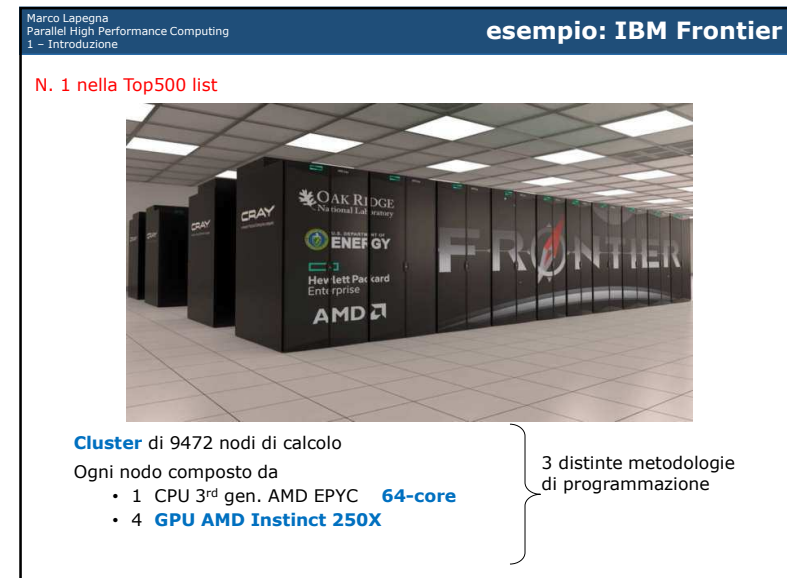
- Applicazioni data parallel
- ad es. Algebra lineare numerica

Interfacciamento con la memoria

- Es. Nvidia Tesla C1060
 - Peak performance 933 Gflop/sec s.p.
 - Memory bandwidth 102 GB/sec (25x10⁹ dati reali/sec)

Necessari ambienti software che tengono conto del particolare hardware

19




20

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

non solo supercalcolatori

calcolo parallelo
calcolo distribuito



- reti veloci
- risorse limitate
- risorse dedicate e omogenee
- applicazione gestisce le risorse
- costo hardware notevole
- overhead sw sistema < 5%
- presenza di vincoli temporali
- es. Tianhe 2A
 - 10 milioni di core
 - 33 Pflops
 - N.2 Top 500 (2017)

- reti lente
- risorse potenzialmente illimitate
- risorse condivise e disomogenee
- ambiente sw.gestisce le risorse
- costo hardware trascurabile
- overhead sw sistema > 20%
- assenza di vincoli temporali
- es. SETI@home
 - 5 milioni CPU
 - 769 Tflops
 - si posizionerebbe al N.130 della Top 500 (2017)

21

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

Progetto SETI@home (1999)

Progetto per la ricerca di intelligenze extraterrestri

Ricerca di sequenze regolari di frequenze all'interno dei segnali radio raccolti dal radiotelescopio di Arecibo



Progetto a base volontaria dove ognuno puo' mettere a disposizione il proprio computer quando non viene utilizzato, facendo partire un programma al posto dello screensaver

Migliaia di nastri da 35 GB. ogni nastro da 35 Gbyte e' diviso in oltre 150000 workunit di circa 350 Kb

Completato nel marzo 2020. Ha permesso lo studio della radiazione cosmica di fondo e la sperimentazione di tecnologie software per il calcolo distribuito

22

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

le dimensioni del progetto

ogni workunit contiene i segnali raccolti in circa 100 sec di osservazione in una data frequenza

- i dati sono ospitati su server presso l'Univ. della California
- Le workunit vengono spedite ai client partecipanti
- i client eseguono FFT con diversi campionamenti ed analisi statistiche per circa 3×10^{12} flops
- i risultati vengono rispediti ai server a Berkley

Numerosi progetti analoghi

- genome@home
- folding@home (oltre 100 Pflops nel 2016 !!)
- QMC@home
- BOINC

23

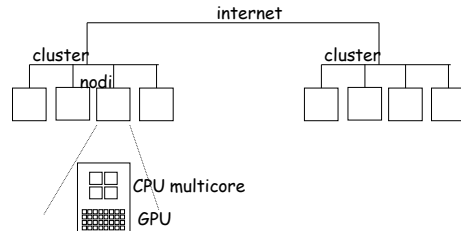
Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

calcolo parallelo e distribuito

- GPU computing
- Multicore computing (c.p. a memoria condivisa)
- Cluster computing (c.p. a memoria distribuita)
- Distributed computing

forme diverse di calcolo che prevedono **differenti metodologie per lo sviluppo di algoritmi**

sempre piu' spesso coesistono in un unico sistema



24

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

principale problema

spostamento dei dati tra le memoria

The diagram illustrates the memory hierarchy from top to bottom: CPU, Cache L1 (C. L1), Cache L2, Mem. centrale, Dischi e mem. secondarie, and Memorie di sistemi remoti. The CPU is at the top, connected to Cache L1, which is connected to Cache L2, which is connected to Mem. centrale, which is connected to Dischi e mem. secondarie, which is connected to Memorie di sistemi remoti.

Unico modello

$$t_{mem} = \alpha + n \beta$$

α = latenza
 β = 1/bandwidth
 n = numero bytes

$t_{mem} \gg t_{calc}$

obiettivo comune:
ridurre gli spostamenti dei dati tra i livelli di memoria

cache \leftrightarrow RAM \leftrightarrow mem. remote

cicli di clock

- 2-3: Piccole (MB), veloci e costose
- ~ 30 : Mem. centrale
- > 100: Dischi e mem. secondarie
- > 100: Memorie di sistemi remoti

Grandi (TB), lente e economiche

25

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

obiettivi del corso

Esplorare varie forme di parallelismo ed acquisire metodologie per lo sviluppo di algoritmi **ad alte prestazioni** in differenti ambienti computazionali

Comprendere in maniera critica

- Quando il calcolo parallelo e' utile
- Le caratteristiche degli ambienti (hw e sw) di programmazione parallela.
- Algoritmi che tengano conto dell'hardware e dei modelli
- Alcune questioni legate al calcolo ad alte prestazioni (HPC)

Un interessante caso di studio

- prodotto di matrici

26

Marco Lapegna
Parallel High Performance Computing
1 - Introduzione

un caso di studio

The diagram shows the calculation of the dot product of two vectors, C and B , resulting in a scalar value $C(i,j)$. The calculation is shown as $C(i,j) = C(i,j) + \sum_{k=1}^n A(i,k)B(k,j)$. The diagram also shows the matrices A and B with the row i of A and the column j of B highlighted in red.

$C(i,j) = C(i,j) + \sum_{k=1}^n A(i,k)B(k,j)$
 $i, j = 1, \dots, n$

Un importante nucleo computazionale in molti problemi

- Utilizzato in tutti i problemi di algebra lineare (fattorizzazioni, autovalori, metodi iterativi,...)
- E' possibile applicare le idee sviluppate in altri contesti
- Favorevole rapporto tra dati $O(n^2)$ e operazioni $O(n^3)$
- Uno degli algoritmi piu' studiati in assoluto

27