



ISTITUTO DI ISTRUZIONE SUPERIORE



ALDO MORO

Liceo Scientifico
Istituto tecnico industriale

Via Gallo Pecca 4/6 10086
RIVAROLO CANAVESE

ANNO SCOLASTICO 2013-2014

ESAME DI STATO

Filippo Valle

Analisi di raggi cosmici mediante supercomputer

Classe VBs - P.N.I.

Sezione Scientifica



Figura 1: Supercomputer Castor

Sommario

In questo documento descriverò un'esperienza che ho realizzato, durante la quale ho studiato come arrivano sulla Terra i raggi cosmici, particelle che vengono prodotte durante le reazioni nucleari nelle stelle. Per fare questo ho utilizzato un supercomputer come laboratorio virtuale per simulare le direzioni dei raggi.

Dopo aver presentato brevemente i concetti di supercomputer e di raggio cosmico, descriverò e commenterò le varie analisi che ho fatto con riferimenti ai grafici realizzati durante l'esperienza e infine mostrerò i risultati che ho ottenuto.

La mia esperienza può essere divisa in due fasi: una di raccolta dati, che ha avuto luogo a Torino ed una seconda di analisi e rielaborazione degli stessi, per realizzare la quale sono stato a Lugano (Svizzera) presso il Centro Svizzero di Calcolo Scientifico.

A Torino, presso la scuola di Fisica dell'Università degli Studi, ho potuto, nell'ambito di un progetto rivolto agli studenti del liceo, utilizzare una speciale strumentazione composta da sensori posti sul tetto e da apposite apparecchiature in grado di convertire i dati provenienti dai sensori in numeri analizzabili su di un computer. In questo modo ho raccolto informazioni su quando arrivano a Terra alcuni tipi di raggi cosmici e in che modo. Dopo tre giorni di raccolta dati avevo abbastanza materiale per poter trarre alcune conclusioni e per poter studiare il fenomeno.

A Lugano ho proseguito la mia esperienza facendo ulteriori analisi. Questa volta ho potuto utilizzare un supercomputer per fare delle simulazioni da confrontare poi con i dati reali. In questa seconda fase ho scoperto come si debba procedere quando si ha a disposizione un supercomputer, diverso da un computer standard. Per esempio, dovendo scrivere un programma da fare funzionare su un supercomputer bisogna utilizzare particolari metodi, che permettono di sfruttare al massimo la potenza della macchina che si sta utilizzando. Ho inoltre imparato il concetto di esecuzione in parallelo e ho scritto ed ottimizzato un mio programma tenendo conto dei diversi modi di procedere per avere i maggiori benefici possibili da macchine parallele. Una volta finito di comprendere tutto ciò che mi serviva ho cominciato a generare simulazioni che prevedevano l'uso di diverse variabili.

Finita l'esperienza ho scoperto come può essere descritta in maniera verosimile la distribuzione dei raggi cosmici nelle varie parti dell'atmosfera.

Abstract

In this document I am going to describe an experience I have made in order to study how cosmic rays come to Earth. Cosmic rays are particles that are produced, for instance, in stars during combustion. While doing this I used a supercomputer as a virtual laboratory where I was able to generate different simulations of virtual rays.

At the beginning I am going to explain what a supercomputer and cosmic rays are, after that I will describe and comment analysis I made, moreover in order to explain the results of my experience I will show some graphics.

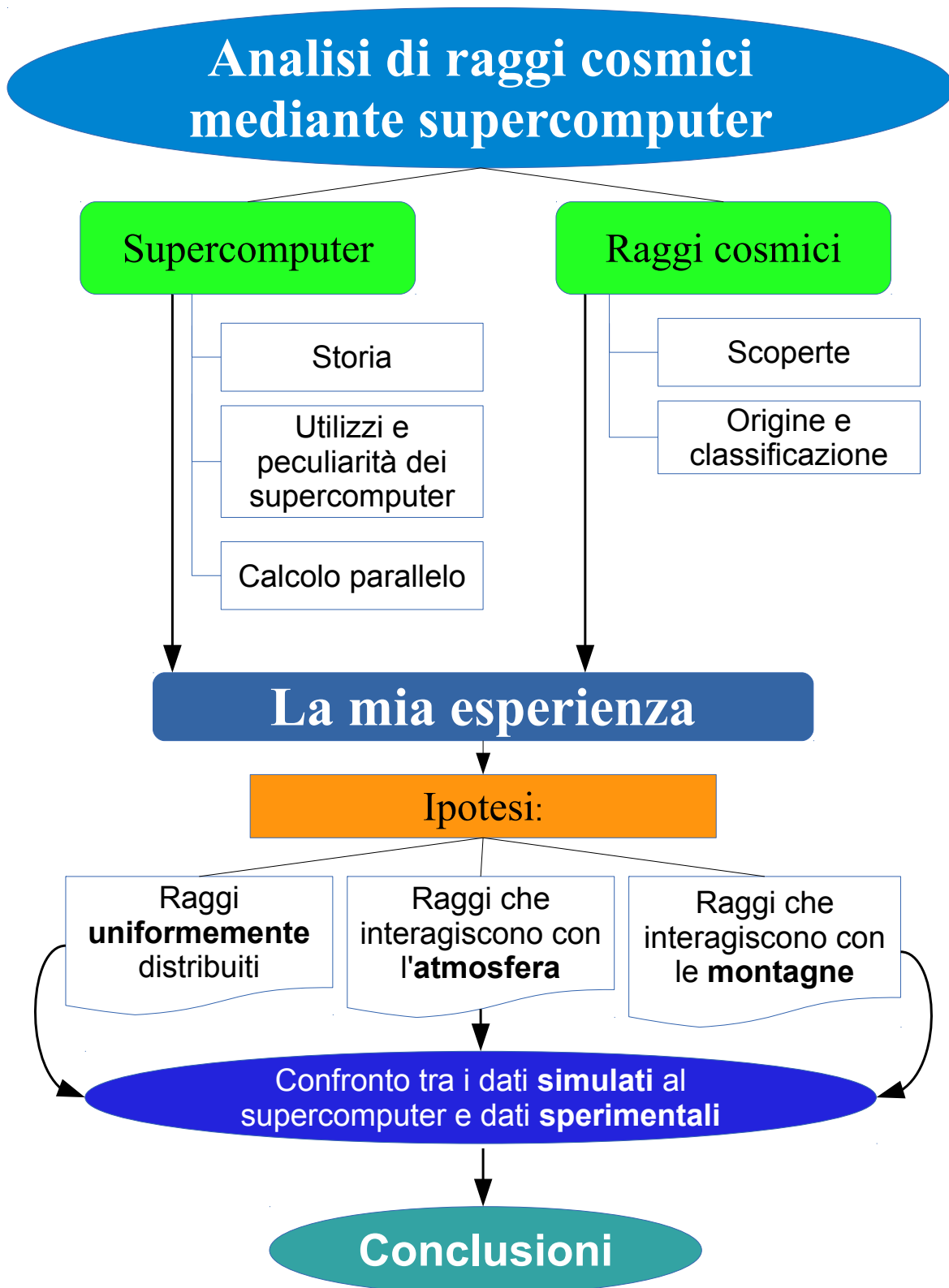
My experience could be divided into two parts: the first one, which took place in Turin, where I collected data; the second one, which took place at Swiss National Supercomputing Centre in Lugano (Switzerland) where I made further analysis of the data using supercomputer.

In Turin, at University of Turin's Physics facility, I was able to use particular instrumentation composed by both sensors placed on the roof of the building either by special equipment capable of converting the data from the sensors into numbers visualizable on a computer. By the way I have collected information about when and how some types of cosmic rays arrive. After three days of data collection I had enough material in order to be able to draw some conclusions and to study the phenomenon.

In Lugano I continued my experience. Here I was able to use a supercomputer to do simulations, which I compared with real data. In this second phase I discovered how to proceed when you have a supercomputer available. For instance, when you write a program to be run on a supercomputer you must use special methods that allow you to maximize the power of the machine you are using. I also learned the concept of parallel execution, typical in this sort of machines. Once understood what I needed I started doing simulations of various kinds and with different variables.

Finally when the experience was completed I found as it could be described the distribution of cosmic rays in different parts of the atmosphere.

Rappresentazione del percorso cognitivo



Indice

1	Introduzione	1
2	Supercomputer	1
2.1	Definizione	1
2.2	Perché <i>Super</i> ?	2
2.3	Storia del calcolo parallelo	3
3	Calcolo parallelo	5
3.1	Hardware parallelo	5
3.1.1	Castor	5
3.2	Software parallelo	6
3.2.1	Test di un algoritmo parallelo	7
4	Metodi di MonteCarlo	8
4.1	Un esempio: calcolare pigreco	8
4.2	Un altro esempio: l'andamento della Borsa	9
4.3	MonteCarlo e i raggi cosmici	10
5	Raggi cosmici	11
5.1	Cosa sono e chi li ha scoperti	11
5.2	Origine e classificazione	12
5.2.1	Composizione ed energie	12
6	La mia esperienza	13
6.1	Raccolta dei dati	13
6.2	Simulazioni e analisi dei dati	14
6.2.1	Arrivo dei raggi uniforme	16
6.2.2	Presenza di atmosfera	16
6.2.3	Presenza di montagne	17
6.2.4	Presenza sia di montagne sia di atmosfera	18
7	Conclusioni	18
8	Codice	21
	Riferimenti Bibliografici	25

1 Introduzione

Ogni giorno siamo colpiti da miliardi di particelle provenienti dallo spazio senza neanche rendercene conto: queste particelle sono i cosiddetti raggi cosmici. Questi raggi sono prodotti dalle reazioni nucleari che avvengono nelle stelle, come il Sole, o nelle Supernovae. Dopo essere state prodotte, queste particelle viaggiano indisturbate per miliardi di chilometri fino a quando non incontrano un ostacolo, quale può essere un pianeta.

In questo documento descriverò come ho **studiato le direzioni di arrivo dei raggi cosmici**, che giungono sul nostro Pianeta (per una completa definizione di raggio cosmico si veda la sezione 5 a pagina 11).

Innanzitutto la mia esperienza può essere suddivisa in **due momenti** principali, che corrispondono ad altrettante fasi che ho realizzato in due sedi distinte:

la raccolta dei dati: tramite alcuni particolari rilevatori posti sul tetto della scuola di Fisica dell'Università degli Studi di Torino ho raccolto i dati di arrivo di raggi cosmici durante tre giornate consecutive;

la creazione di simulazioni e l'analisi dei dati: grazie ai supercalcolatori presenti al Centro Svizzero di Calcolo Scientifico di Lugano, ho potuto simulare il comportamento di raggi cosmici virtuali, che ho poi confrontato con i dati raccolti.

Durante la prima fase, nell'ambito di un progetto rivolto agli studenti del liceo che prevedeva la misurazione degli angoli di arrivo dei raggi cosmici a Terra, ho potuto utilizzare delle particolari apparecchiature, che mi hanno permesso di avere delle informazioni riguardo a raggi cosmici **realmente giunti** sul nostro Pianeta. Queste informazioni riguardavano in particolare l'orario e la direzione di arrivo delle particelle. Dopo aver raccolto i primi dati ho potuto fare alcune considerazioni, ma per analizzare e studiare in maniera più approfondita la questione ho proseguito con la fase successiva.

Nella seconda fase, in cui ho lavorato in maniera più attiva, ho potuto utilizzare un supercomputer per fare ulteriori analisi. Questa volta, però, non ho lavorato su dati ricavati sperimentalmente, bensì su **dati virtuali frutto di elaborazioni al supercomputer**. I supercomputer che ho utilizzato sono molto più potenti dei computer che si utilizzano quotidianamente e proprio per questo utilizzandoli ho potuto analizzare una grande quantità di dati, altrimenti difficile da gestire. Per rendere l'idea di cosa sia un supercomputer lo si può immaginare come un armadio al cui interno sono presenti migliaia di computer "normali", che lavorano insieme ai compiti che sono stati loro assegnati; poiché le varie parti di un supercalcolatore lavorano in maniera abbastanza indipendente queste macchine vengono anche chiamate **computer paralleli**.

2 Supercomputer

2.1 Definizione

Cercando su di un vocabolario [9] il vocabolo *supercalcolatore* si legge che: il termine, entrato nell'uso sul finire degli anni Sessanta, indica i più potenti e sofisticati elaboratori elettronici, caratterizzati, rispetto ai calcolatori ordinari, da una maggiore velocità di operazioni; attualmente, il termine indica specificatamente i calcolatori paralleli, che raggiungono le volute alte velocità facendo lavorare parecchie unità di calcolo identiche (processori) in parallelo fra loro.

Un supercomputer è dunque un computer in grado di eseguire i calcoli necessari ad un determinato programma in maniera molto rapida sfruttando i suoi numerosi processori utilizzati in parallelo (si dice che due processori sono in parallelo se riescono ad eseguire operazioni diverse in contemporanea e in maniera indipendente gli uni dagli altri). Per essere efficaci è necessario che anche il software, ossia il programma da eseguire, sia il più possibile parallelo in modo da sfruttare al massimo l'architettura di questi speciali calcolatori.

2.2 Perché *Super*?

Questi potentissimi supercalcolatori sono realizzati con componenti simili a quelle presenti nei classici computer in commercio utilizzati comunemente nelle abitazioni o negli uffici come:

la CPU: è la parte **centrale di elaborazione** dei dati, serve ad eseguire i calcoli richiesti dalle varie parti del programma durante l'esecuzione. Una CPU può contenere uno o più cores;

i cores: sono i **nuclei elaborativi** delle CPU, sono loro che eseguono effettivamente i calcoli del programma;

la GPU è l'unità di **elaborazione grafica**, funge da coprocessore e si contraddistingue per essere specializzata nel rendering di immagini grafiche o comunque in operazioni specifiche. Serve ad aumentare le capacità della CPU;

la memoria (RAM): serve a contenere i file del programma e a **salvare i dati** durante la sua esecuzione;

i nodi: sono **schede** che possono contenere: una o più CPU, la memoria, eventualmente degli acceleratori (GPU) e i collegamenti verso l'esterno e verso altri nodi;

lo spazio di salvataggio dei dati (scratch): serve a **contenere i risultati** dopo l'esecuzione del programma in modo da poterli analizzare anche a distanza di tempo;

le cassette magnetiche: servono a contenere i dati per **periodi molto lunghi**. Benché sembri una tecnologia antiquata queste cassette sono molto affidabili e riescono a contenere una quantità di dati maggiore rispetto ad altri supporti;

i threads: sono le varie parti in cui è diviso il **programma** (detto software) in modo che possa essere eseguito in contemporanea da più unità di calcolo. Se impostato correttamente il supercomputer esegue un thread per ogni core;

Eppure, benché siano composti con **elementi simili**, un supercomputer differisce da un computer ordinario in molti aspetti, primo fra tutti l'altissima velocità di elaborazione, ottenuta mediante più CPU che, opportunamente controllate e collegate, operano in parallelo tra loro: a volte vengono collegate diverse decine di migliaia di unità, che riescono a lavorare contemporaneamente.

Come vedremo in seguito, operando in parallelo si riesce a **ridurre** drasticamente **il tempo di esecuzione** di un programma, ottenendo così i risultati in tempi più rapidi o riuscendo ad elaborare, a parità di tempo di esecuzione, una maggiore quantità di dati rispetto ad un computer classico.

Queste qualità vengono sfruttate in molteplici ambiti sia militari (ad esempio nel calcolo di **trattorie di vettori aerospaziali** o di satelliti) che civili. Tra questi ultimi ha particolare rilievo il campo delle **previsioni meteorologiche**, la qualità delle quali dipende in gran parte dalla possibilità di disporre tempestivamente dei risultati di complesse elaborazioni. In questo caso i supercomputer permettono di rappresentare in anticipo l'evoluzione dell'atmosfera e di prevedere i fenomeni atmosferici con capacità di gran lunga superiori a quelle ottenibili con altri mezzi. Inoltre il calcolo di una previsione del tempo è facile da rendere parallelo: ogni processore è in grado di calcolare una regione di spazio diversa senza dover dipendere da nient'altro.

Analogamente a quanto avviene in meteorologia, anche nella cosiddetta modellazione della realtà questi potenti calcolatori consentono di simulare eventi del genere più vario, purché si sappia esprimere la loro evoluzione in forma numerica, attraverso sistemi di equazioni. In tal modo si ha a disposizione un vero e proprio **laboratorio virtuale**, in cui si possono simulare situazioni irraggiungibili sperimentalmente come per esempio, negli studi geofisici, il centro della Terra o, in astronomia, l'evoluzione di una galassia; si immaginino le dimensioni, in anni luce, dell'edificio che dovrebbe contenere un laboratorio in cui studiare una galassia!! Nel mio caso ho potuto simulare il percorso di raggi cosmici all'interno dell'atmosfera, anche questo praticamente impossibile da realizzare in un laboratorio reale.

Agli inizi del Novecento il **poeta futurista Marinetti** scrisse: "*Ricostruiamo futuristicamente l'universo*" inneggiando all'uso delle macchine nel suo prossimo futuro [6]; è interessante pensare

come lo studio e soprattutto la ricostruzione del nostro universo tramite una macchina (in questo caso un supercomputer) fosse già stato immaginato e profetizzato nel secolo scorso; in effetti quello che anch'io ho fatto nella mia esperienza è stato ricreare un ambiente dell'universo tramite una macchina.

Tuttavia fin dai primi supercomputer emersero alcuni limiti: si immagini, ad esempio, di avere 52 carte di uno stesso mazzo distribuite in disordine e di volerle mettere in ordine. Se ci fossero quattro persone a volersi dividere le attività secondo i colori (cuori, quadri, fiori, picche) si farebbe certamente prima che se si dovesse svolgere la stessa attività da soli (in linea teorica se si raddoppiano gli aiutanti si dimezza il tempo impiegato). Ma se si fosse in 52, tutti intorno allo stesso tavolo, a voler ordinare il mazzo si farebbe probabilmente solo una gran confusione. Quest'esempio può far comprendere il **limite** al di là del quale non conviene insistere sul parallelismo del programma ma, piuttosto, sulla velocità della singola unità di calcolo.

Un'altro problema emerse quando **Gene Amdahl**, uno dei maggiori architetti della recente storia dei computer, enunciò all'inizio degli anni Settanta un'ipotesi che venne poi denominata Legge di Amdahl: "*Un perfetto parallelismo nelle attività dei computer non è possibile perché esisteranno sempre delle sequenze ineliminabili di software che richiede necessariamente l'esecuzione in serie*". Si pensi, per esempio, a una serie di numeri di **Fibonacci** in cui ogni numero è la somma dei due precedenti e i cui primi fattori sono 1, 1, 2, 3, 5, 8, 13, 21..., come definito dalla successione $f(n+2) = f(n+1) + f(n)$; non è possibile calcolare con un supercomputer che esegue le operazioni in parallelo il fattore n -esimo della successione poiché ogni successiva operazione dipende dalle due precedenti [7].

Nella mia esperienza ho dovuto sospendere il parallelismo (si notino nel programma a pagina 21 le istruzioni **!\$OMP SINGLE**, per esempio alla riga 104) per esempio nel momento del **salvataggio** dei file con i dati simulati; ho infatti sperimentato che se troppi processori scrivono contemporaneamente nello stesso file, la memoria si sovraccarica e il programma si blocca.

2.3 Storia del calcolo parallelo

All'inizio degli **anni Sessanta** si comprese che, per alcuni tipi di problemi, occorre computer che si discostassero dall'originale architettura di Von Neumann, che prevede, come caratteristica più importante, la sequenzialità delle operazioni. Si scoprì anche che a parità di tecnologia all'interno di un calcolatore si potevano ottenere prestazioni più elevate se venivano utilizzate, contemporaneamente, molteplici unità di elaborazione in grado di eseguire, in **parallelo**, gruppi differenti, e opportunamente selezionati, di istruzioni. Nacquero così i supercomputer. L'obiettivo di tali supercomputer era, in generale, quello di risolvere problemi caratterizzati dalla grande quantità di dati numerici sui quali occorreva eseguire ripetutamente sequenze di istruzioni abbastanza simili, piuttosto che eseguire molte istruzioni, ma necessariamente in sequenza. Ciò che distingueva dunque i supercomputer erano essenzialmente tre elementi: l'utilizzo delle più avanzate tecnologie circuitali, l'uso di molteplici unità di elaborazione e la progettazione di opportuni algoritmi che prevedevano l'esecuzione di più operazioni simili da parte di diverse unità di calcolo.

Secondo quanto riporta lo scienziato Richard Feynman già durante la Seconda Guerra Mondiale ai tempi della ricerca sulla bomba nucleare, nell'ambito del progetto Manhattan, a Los Alamos si era cercato di rendere **più efficienti** i sistemi di calcolo che all'epoca si avevano a disposizione. Allora si usarono delle schede colorate (ad ogni colore corrispondeva un diverso problema) che venivano mischiate in modo da eseguire contemporaneamente i calcoli per problemi differenti [4]. In questa maniera gli scienziati riuscirono ad eseguire i calcoli di cui avevano bisogno nelle loro ricerche in maniera più efficace poiché il computer risolveva nello stesso tempo più problemi diversi.

Tra gli **anni Sessanta** e la metà degli anni Settanta la società **Control Data Corporation (CDC)** con i suoi supercomputer fu l'azienda che ebbe un maggior impatto nella produzione di questo nuovo strumento di calcolo. I tecnici della CDC per ottenere elevate prestazioni di calcolo svilupparono diverse soluzioni tecnologiche come l'utilizzo di processori specializzati per i diversi compiti, l'utilizzo di pipeline (unità di calcolo divise in sezioni altamente specializzate) e l'utilizzo di processori vettoriali progettati appositamente per eseguire le operazioni in parallelo. Alcune scelte della CDC rischiarono di far fallire la società e alcuni tecnici insoddisfatti dei dirigenti della società

abbandonarono la CDC per tentare nuove strade. Tra questi il più famoso fu sicuramente **Seymour Cray** che con il suo supercomputer Cray-1 segnò una nuova strada sia per le soluzioni tecnologiche utilizzate sia per l'elevata potenza di calcolo sviluppata.

Dalla metà degli **anni Settanta** fino alla fine degli anni Ottanta la Cray Research rappresentò il leader nel settore del supercalcolo. Cray estese le innovazioni utilizzate nelle macchine della CDC portandole al loro estremo ed utilizzando soluzioni innovative come il raffreddamento per mezzo di liquidi o la creazione di strutture a torre ove alloggiare le schede con le unità di calcolo in modo da ridurre la lunghezza media delle connessioni (se due processori sono fisicamente più vicini i segnali in viaggio tra i due impiegano meno tempo).

Durante gli **anni Novanta** i supercomputer divennero macchine a parallelismo massiccio basate su centinaia se non migliaia di processori elementari. Inizialmente questi processori erano sviluppati esplicitamente per il supercalcolo, ma verso la fine degli anni Novanta oramai si era passati definitivamente all'utilizzo di processori generici che fornivano elevate prestazioni e costi ridotti per via della produzione in serie attuata da ditte come la Intel [17].



Figura 2: Supercomputer Piz Daint al Centro Svizzero di Calcolo Scientifico. A novembre 2013 era il sesto supercomputer più potente al mondo.

Con il passare degli anni i supercomputer si sono sempre più perfezionati fino a raggiungere la potenza di alcuni **Petaflop/s**¹ (un computer con la potenza di 1 petaflop/s riesce ad eseguire 1 000 000 000 000 000 di operazioni con la virgola in un secondo).

Ogni sei mesi circa viene stilata una classifica la **Top500** in cui vengono elencati i **cinquecento supercomputer più potenti al mondo**, inoltre viene anche costruita una speciale classifica, la GREEN500, con le macchine più ecologiche: infatti uno dei problemi dei supercomputer sono proprio i consumi di energia necessaria al loro raffreddamento. Ad oggi (la quarantaduesima ed ultima rilevazione è stata fatta a Novembre 2013) il supercomputer più potente al mondo è il cinese **Thiane-2 (Milky Way II)** sviluppato dalla China's National University of Defense Technology e utilizzato dal National Super Computer Center in Guangzhou. Questa macchina è in grado di erogare una potenza di 33,86 petaflops [16]. Per fare un paragone un computer per uso scolastico eroga circa 10 gigaflop/s (un milionesimo rispetto al supercomputer cinese) ed un moderno smartphone circa 0,11 gigaflop/s. Inoltre si può immaginare la velocità di sviluppo dell'informatica pensando che nella prima edizione della Top500 nel 1993 risultò primo il supercomputer CM-5 sviluppato presso i LosAlamos National Laboratories in America ed aveva una potenza di calcolo di 59.7 gigaflop/s, il computer top di gamma di Apple oggi ha una potenza di 81,6 gigaflop/s!

¹flop/s è l'abbreviazione di FLoating Operation Per Second: ossia operazioni con numeri con la virgola che il computer è in grado di eseguire in un secondo.

La crescente potenza di calcolo è descritta dalla legge (empirica) di Moore, che afferma: “*La complessità dei dispositivi (numero di transistor per pollice quadrato nei microprocessori) raddoppia ogni 18 mesi*” (nel futuro vale applicato al numero di cores per processore) [12]. Inoltre la sfida per la produzione di supercomputer sempre più potenti non è ancora finita: negli ultimi anni si è cominciato a parlare [8] di **calcolo in exa-scale**: ossia di una potenza di calcolo mille volte superiore al più potente supercomputer oggi costruito; non è difficile che nei prossimi anni si arrivi ad avere supercomputer con una potenza di calcolo nell’ordine dell’**Exaflop**.

3 Calcolo parallelo

Durante la mia esperienza ho potuto utilizzare un supercalcolatore e per farlo funzionare ho scritto un programma che generava, utilizzando un **Metodo di MonteCarlo** (descritto nella sezione 4, pagina 8), alcuni raggi cosmici virtuali che giungevano sulla Terra.

Il mio programma prevedeva, in una prima fase, di generare dei **raggi cosmici virtuali** provenienti da **tutte le direzioni** e senza particolari caratteristiche. In una fase successiva, ho introdotto nella simulazione **alcune variabili** come la presenza dell’atmosfera e di montagne (come potrebbero essere le Alpi a Torino) nei dintorni del luogo dell’esperimento; queste variabili portano però ad **escludere alcuni raggi** (se, per esempio, un raggio proveniente dallo spazio rimane bloccato in una montagna non viene considerato), per ottenere una discreta quantità di risultati utili ed analizzabili si rende dunque necessaria la generazione di una quantità maggiore di dati. All’aumentare del numero di raggi da simulare aumenta, in modo direttamente proporzionale, anche il tempo di elaborazione necessario, come si può notare dal grafico 3, che mostra il tempo necessario a calcolare n eventi.

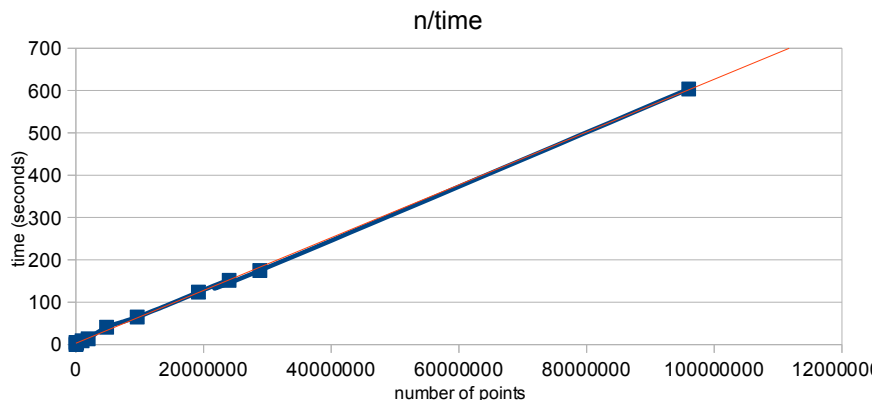


Figura 3: Grafico del tempo necessario al calcolatore per calcolare un numero n di eventi.

Considerata la grande quantità di dati da analizzare, ho scritto il programma in modo da renderlo parallelo e sfruttare così al massimo le potenzialità del supercomputer che avevo a disposizione.

3.1 Hardware parallelo

3.1.1 Castor

Il supercomputer (foto in figura 1) che ho utilizzato durante l’esperienza si trova presso il Centro Svizzero di Calcolo Scientifico dove sono presenti diversi supercomputer a disposizione dei ricercatori di tutta Europa. In questo centro ad ogni macchina viene dato il nome di una **montagna alpina**; quello che ho utilizzato io si chiama *Castor* dal nome di una montagna del massiccio del Monte Rosa, nelle Alpi Pennine a sua volta chiamata così in onore del dioscuro della mitologia greca.

La macchina è prodotta dalla International Business Machines Corporation (IBM), un’azienda statunitense, tra le maggiori al mondo nel settore informatico. Il nome tecnico della macchina è *IBM iDataPlex Research & Development Cluster CASTOR* ed è formata da **32 nodi** di calcolo: in

ogni nodo sono presenti **due CPU** Intel Xeon 5650 da $2,6GHz$ con all'interno **sei cores** ognuna, **2 GPUs** NVIDIA Tesla e **24 GB di memoria RAM**. In totale vi sono dunque 6 cores per CPU e due CPU per nodo, cioè 12 cores per nodo che moltiplicati per i 32 nodi danno 384 cores (si immagini che un computer scolastico ne ha al massimo 4). Per quanto riguarda la RAM avevo a disposizione 768 GB di memoria comune. I risultati li ho invece salvati in uno scratch da 30 Terabyte.

In linea teorica questo supercomputer è in grado di erogare 665 Gigaflop/s di potenza (circa seicento volte più potente di un computer classico).

Si noti come la potenza del supercomputer sia data non dal singolo elemento (un processore di un normale computer funziona a circa $2,4 - 3,2GHz$, in linea con il singolo processore del supercalcolatore), ma dalla presenza di **numerosi elementi di calcolo ben organizzati**: i singoli nodi riescono a comunicare in maniera efficace e ad ottimizzare al massimo le risorse durante l'esecuzione in parallelo di un programma. In aggiunta anche la presenza di una grande quantità di memoria rende possibile la gestione di molti dati, difficoltosa in computer domestici in cui se ne hanno a disposizione solo pochi GigaByte.

Sempre riguardo alla memoria dei computer si immagini che la stima iniziale fornita dalla NASA della memoria necessaria all'Apollo Guidance Computer (il computer che negli anni Sessanta controllava le rotte delle navicelle Apollo in **orbita verso la Luna**) fu di soli 4 kilobyte² di memoria riscrivibile (RAM) e 4 Kilobyte di memoria in sola lettura (ROM). Nella sua versione finale, l'AGC arrivò ad avere 2 kilobyte di RAM e 36 kilobyte di ROM [1]!

3.2 Software parallelo

Per poter sfruttare al meglio tutta la potenza di calcolo è però necessario scrivere un programma in grado di far eseguire al supercalcolatore più operazioni in parallelo, altrimenti se tutte le operazioni venissero eseguite in serie da una sola unità, la potenza utilizzata sarebbe uguale a quella di un computer da ufficio.

Per scrivere il programma (stampato a pagina 21) che generasse le simulazioni, ho utilizzato il **linguaggio Fortran 95**. Il nome Fortran deriva da FORMula TRANslator, letteralmente traduttore di formule matematiche in algoritmi; proprio per la facilità di gestire il linguaggio matematico è, insieme al C, il linguaggio di programmazione più utilizzato in ambito scientifico. Il Fortran venne realizzato tra il 1954 e il 1957 da un gruppo di esperti operanti presso la IBM, da allora ne sono state presentate varie versioni l'ultima delle quali, quella che ho utilizzato, risale al 1995.

Le istruzioni di Fortran non sono tuttavia sufficienti per ordinare al computer di eseguire il codice in parallelo. Per questo ho utilizzato **OpenMP** (MP è la sigla di Multi Processore): una libreria che permette di inserire nel codice alcune istruzioni (a pagina 21 tutte quelle che iniziano con **!\$OMP**), con cui è possibile inviare al supercomputer istruzioni su quali parti del programma eseguire in parallelo. Inoltre grazie a OpenMP è anche possibile scegliere quanti threads devono eseguire il programma: in questo modo si possono fare dei **test** volti a verificare l'efficienza del codice parallelo e il tempo che si guadagna eseguendo lo stesso programma su un diverso numero di threads.

Quando si scrive un **software in parallelo** bisogna tenere di alcuni aspetti e potenziali problemi inesistenti in programmi non paralleli. Per esempio, durante la scrittura del programma ho dovuto tenere conto del fatto che per generare numeri casuali i computer partono dall'ora del singolo processore. Avendo una singola unità di calcolo ciò è ininfluente, ma con più unità bisogna tenerne conto. Chiedendo l'ora di un processore, Fortran restituisce su tutti i processori sempre lo stesso valore, che corrisponde alla media delle ore di tutti i processori (gli orologi delle CPU partono alla loro accensione, perciò, non accendendosi mai tutti simultaneamente, non sono mai sincronizzati). A causa di questo in un primo momento i numeri casuali generati da diversi processori erano tutti identici! Per ovviare a questo problema ho moltiplicato l'ora per il numero del processore e ho imposto al programma di utilizzare quello come punto di partenza per la generazione dei numeri casuali, facendo in modo che fossero tutti diversi.

²Un Kilobyte equivale a 0,001 Gigabyte. Perciò la memoria del computer che ha portato l'uomo sulla Luna è circa un millesimo di quella dei nostri computer!

3.2.1 Test di un algoritmo parallelo

Dopo aver scritto il programma, ho voluto verificare l'utilità dell'utilizzo del supercomputer. Per fare questo ho eseguito il seguente test: ho fatto girare su Castor una versione semplificata del programma e, utilizzando i comandi di OpenMP, ho imposto al calcolatore di eseguire i calcoli prima su un solo thread poi crescendo fino ad otto. Ho anche chiesto al programma di **misurare il tempo** necessario all'esecuzione del programma ogni volta. Infine ho realizzato il grafico 4, che mostra il tempo necessario ad eseguire lo stesso programma su n threads.

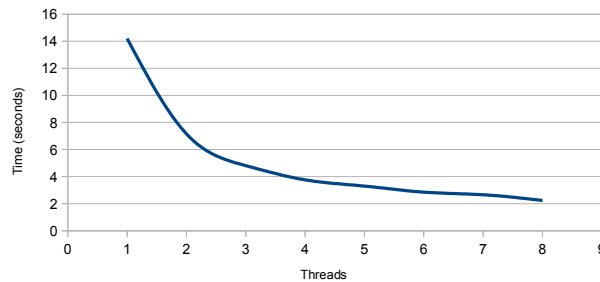


Figura 4: Tempo di esecuzione di un algoritmo parallelo su un numero diverso di threads.

Si può notare che aumentando il numero dei threads, il tempo necessario ad eseguire il codice si riduce notevolmente: il **programma parallelo è efficace!**

Per fare ulteriori analisi, si può studiare lo **speedup**. Lo speedup è il rapporto tra il tempo impiegato dall'algoritmo non parallelo e dallo stesso algoritmo parallelo eseguito su n threads; in pratica se si ha uno speedup di 2 allora, in quel caso, il programma parallelo ha impiegato la metà del tempo rispetto allo stesso programma non parallelo.

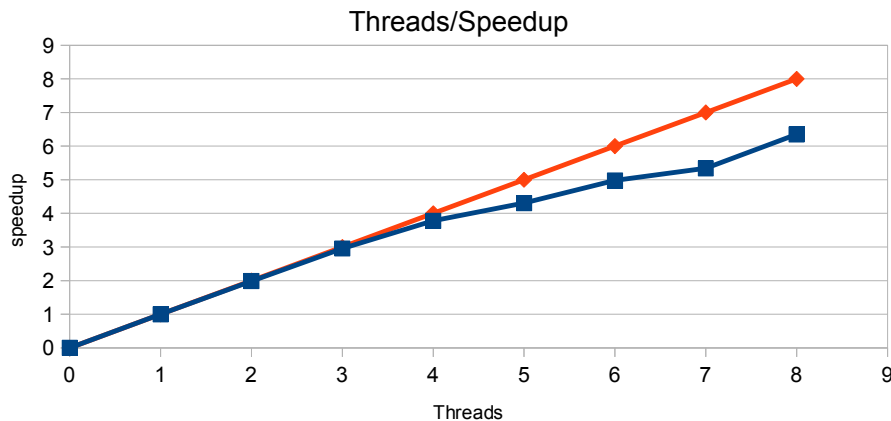


Figura 5: Speedup durante l'esecuzione del software parallelo.

Osservando il grafico 5 si può facilmente notare che all'aumentare del numero dei threads aumenta in maniera direttamente proporzionale lo **speedup**. Si può inoltre notare che la curva dello speedup si allontana dalla retta (rossa) $y = x$: questo perché è **impossibile raggiungere lo speedup ideale** rappresentato da tale retta. Questo avviene perché, come enunciato dalla già citata legge di Amdahl, non si riesce a scrivere tutto il programma in parallelo, ma si devono lasciare alcune parti in serie: per esempio la scrittura di uno stesso file non può essere eseguita contemporaneamente da più threads.

Inoltre non si raggiungono i valori ideali a causa del tempo perso nel momento in cui due o più unità devono comunicare tra di loro. I vari cores sono posizionati ad una certa **distanza** e il segnale elettrico che percorre lo spazio tra le varie unità, viaggiando a velocità prossime a quella della luce, impiega un determinato tempo. Questo tempo è piccolissimo (la luce ha una velocità di circa $300\,000\,Km/s$, quindi per percorrere pochi centimetri impiega circa $0,000000000000003$ secondi),

ma se moltiplicato per il numero di comunicazioni che mediamente avvengono durante l'esecuzione di un programma, diventa una significativa perdita.

Per eseguire questo test ho utilizzato due processori con quattro cores ognuno, questo contribuisce a giustificare il fatto che lo speedup è molto vicino al valore ideale solo fino a quando i threads sono meno dei quattro di una singola CPU. I cores all'interno della stessa CPU comunicano in maniera molto veloce soprattutto a causa della loro ridottissima distanza, invece dovendo comunicare due cores in due CPU diverse il tempo perso aumenta.

Utilizzando un algoritmo ed un programma parallelo si possono dunque ottenere le prestazioni necessarie a simulare una quantità di raggi cosmici sufficiente per ottenere una buona simulazione.

4 Metodi di MonteCarlo

Durante la mia esperienza ho utilizzato un algoritmo che sfruttava un metodo di MonteCarlo.

MonteCarlo è il nome di una serie di **metodi statistici** basati sui **numeri casuali**. Essi sono utili per superare i problemi legati ai test esatti (ad esempio i metodi basati sul calcolo combinatorio). Il metodo è usato per generare stime attraverso simulazioni e fu utilizzato per la prima volta nell'ambito del, già citato, Progetto Manhattan per simulare la diffusione di neutroni nella fusione nucleare. I normalizzatori del metodo furono, in particolare, Enrico Fermi e John Von Neumann. Esso è relativamente facile da utilizzare con algoritmi paralleli ed è proprio per questo motivo che spesso vengono utilizzati su macchine che prevedono l'uso di questi algoritmi come i supercomputer. Inoltre è facile calcolare l'errore che il metodo stesso ha prodotto e quindi ci si rende subito conto se l'analisi effettuata può o meno essere utile. Il nome *MonteCarlo* fu, invece, inventato in seguito da Nicholas Constantine Metropolis in riferimento alla nota tradizione nei giochi d'azzardo del piccolo **Stato omonimo nel sud della Francia**.

4.1 Un esempio: calcolare pigreco

Un esempio di utilizzo di un metodo MonteCarlo in ambito prettamente matematico può essere il seguente: dati un quadrato di lato 1 e il quarto di cerchio con lo stesso lato come in figura 6, si inizia generando dei **punti casuali** all'interno del quadrato; se si generassero infiniti punti si coprirebbe tutta l'area del quadrato e del cerchio, generandone un numero molto elevato si ottiene comunque una buona stima delle aree e il rapporto tra i punti caduti all'interno del cerchio e il totale dei punti generati è pari al rapporto tra l'area del cerchio e quella del quadrato ossia $\frac{\pi}{4}$ (l'area del quadrato vale $l^2 = 1^2 = 1$ e quella del quarto di cerchio $\frac{A_{cerchio}}{4} = \frac{\pi r^2}{4} = \frac{\pi 1^2}{4} = \frac{\pi}{4}$). Da qui, moltiplicando per 4, si può facilmente stabilire il valore di **pigreco** $\simeq 3,141592\dots$. In ogni caso, quando si utilizza un metodo di Monte Carlo, la precisione del nostro risultato aumenta con l'aumentare del numero dei punti casuali generati, perciò per ottenere una stima più precisa si rende necessario fare in modo di ottenere la maggior quantità possibile di dati.

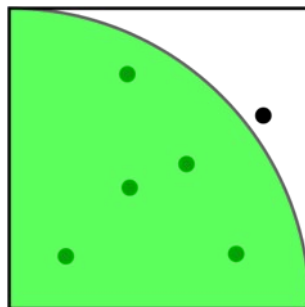


Figura 6: Esempio di utilizzo di un metodo di MonteCarlo.

4.2 Un altro esempio: l'andamento della Borsa

I metodi di MonteCarlo possono anche essere utilizzati per fare analisi statistiche in **ambito finanziario**. Se si è a conoscenza di alcuni fattori che possono influenzare l'andamento di una determinata azione, di un derivato, di un tasso di cambio o di un qualsiasi prodotto finanziario si è facilmente in grado di creare una simulazione MonteCarlo adeguata a mostrare possibili scenari riguardo al futuro del prodotto preso in considerazione.

In pratica è sufficiente scrivere un'equazione o un sistema di equazioni che descriva, in funzione del tempo, l'andamento del prodotto finanziario da analizzare, dopo averne verificato la validità e l'affidabilità per ciò che è successo nel passato se ne studia l'andamento nel futuro.

Analizziamo un semplice **esempio**: se si ipotizza di possedere un'azione che abbia in un tempo zero (t_0) un valore $S_0 = S(t_0)$ e che, in caso di guadagno, si investirà ciò che si è guadagnato in azioni identiche, si giunge alla conclusione che dopo un tempo T il valore dell'azione sarà aumentato di un fattore ΔS che dipende dall'investimento iniziale S e dall'andamento del valore dell'azione μ ($\Delta S = \mu S_0 \Delta t$). Se tutto procede perfettamente e senza influenze di nessun tipo (cosa praticamente impossibile in un mercato finanziario) si ottiene che al tempo T possediamo un'azione che vale $S(T)$ con $S(T) = S_0 e^{\mu(T-t_0)}$ [13].

Con una simulazione MonteCarlo si può stimare cosa succederà all'azione nel futuro. Prendendo due parametri: il primo (μ) che indichi l'**andamento generale del titolo** in un dato periodo di tempo e un altro (σ) che dà indicazioni sull'**ampiezza delle fluttuazioni**³ cui è soggetto il titolo nello stesso periodo ovvero sul numero di azioni in circolazione e sulla fiducia degli investitori, si può scrivere che:

$$\Delta S = \mu S_0 \Delta t + \sigma S_0 \Delta W$$

con

$$\Delta W = Z \sqrt{\Delta t}$$

dove: ΔS è il guadagno o, se negativo, la perdita; μ il parametro che definisce l'andamento del mercato; $\Delta t = (T - t_0)$ il periodo preso in considerazione; σ il parametro che indica le fluttuazioni del mercato e Z un numero casuale ($Z \in [-1, 1]$).

A differenza del calcolo di pigreco, in questo caso i parametri che ho inserito nella simulazione non sono realmente fissi nel tempo e variano in base a fattori non sempre noti o prevedibili ed è pertanto necessario aggiornare i parametri (μ e σ) molto spesso seguendo l'evolversi del mercato. Se facciamo una simulazione con $\mu = 0,45$ e $\sigma = 0,1$: dati che si riscontrarono in borsa nell'**anno 1928** in cui il valore delle azioni cresceva a dismisura (μ è relativamente grande) senza particolari variazioni e con grande fiducia nel mercato da parte degli investitori (σ è relativamente piccolo), si ottiene il grafico rappresentato in figura 7.

Dal grafico si deduce che negli anni prima della crisi si poteva stimare una continua crescita, come dimostrato dal fatto che a sinistra (anni prima della crisi) le varie curve sono tutte molto vicine e relativamente simili; tuttavia con il passare del tempo l'andamento del mercato diventa sempre più imprevedibile: si nota infatti che vi è grande discrepanza tra le varie curve del grafico dopo un certo periodo di tempo.

Come ci racconta la Storia, il **24 ottobre del 1929** scoppiò la bolla speculativa che si creò negli anni precedenti e i titoli della borsa di Wall Street persero quasi la metà del loro valore nel giro di poche ore e nessuno fu in grado di prevedere come e quando si sarebbe proposta una situazione del genere. Utilizzando una simulazione come quella descritta sopra si possono avere informazioni riguardo un possibile realizzarsi di situazioni come la crisi degli anni Trenta (la curva azzurra del grafico rappresenta un crollo dell'azione a metà del 1929). La simulazione che ho realizzato rende l'idea di come avere un metodo di stima può aiutare a vedere realizzate ipotesi e a prendere in considerazione tutti gli eventi possibili prima di realizzare un'operazione finanziaria o di investire in un determinato titolo. Dopo aver visionato una simulazione come questa si può sapere che, nelle condizioni in cui è realizzata, si ha una grande probabilità di crescita nel breve periodo e che con il passare del tempo diventa probabile anche una decrescita netta (curve azzurra e rossa) [2].

³le fluttuazioni sono le variazioni del valore dell'indice dell'azione, sono tanto più grandi tanto è maggiore la differenza tra il massimo e il minimo valore assunti dal prodotto in un determinato periodo.

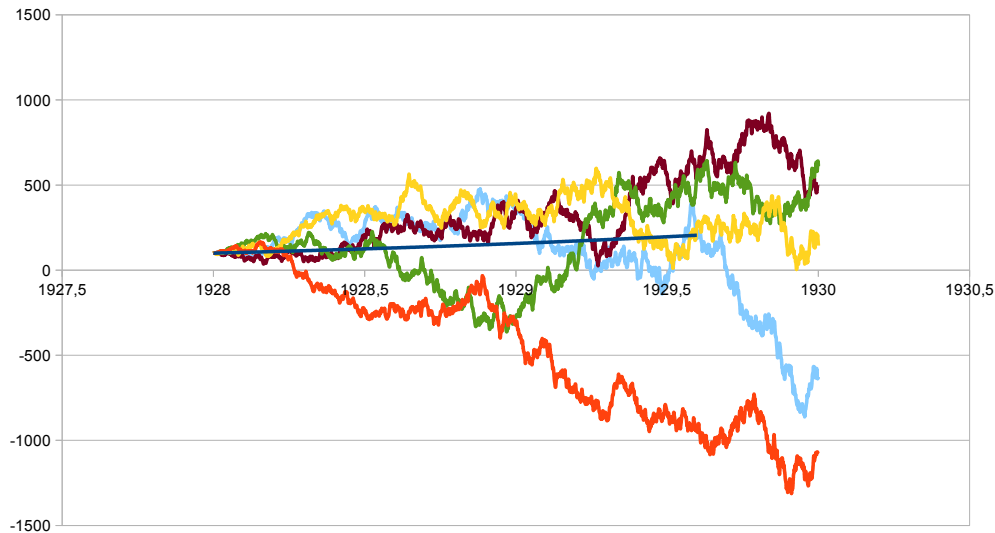


Figura 7: Simulazioni MonteCarlo per l'andamento della Borsa nel biennio 1928-1930.

Per avere una simulazione migliore è necessario, per quanto possibile, far variare i parametri in maniera da tenere in conto anche fattori come: il costo del denaro, i cambi delle varie valute in gioco, i movimenti speculativi, il costo di oro e petrolio; inoltre resta comunque impossibile inserire nella simulazione eventi imprevedibili come uragani, terremoti o guerre che hanno il loro peso sull'andamento del mercato. In **tempi moderni** le simulazioni degli economisti utilizzano equazioni sempre più complesse che, per quanto possibile, tengono in considerazione, sempre con variabili basate sul caso, anche questi fattori.

4.3 MonteCarlo e i raggi cosmici

Anche nella mia esperienza ho utilizzato un Metodo di MonteCarlo. Nel mio caso il metodo è stato utilizzato per simulare l'arrivo dei raggi cosmici: ho generato su di una immaginaria **semisfera** con il centro sui rilevatori due angoli, che descrivevano il percorso seguito dalla particella nel suo viaggio attraverso l'atmosfera. In particolare sulla semisfera ho generato alcuni punti, definiti tramite due angoli: il primo, theta (θ), rispetto alla verticale (zenit) e il secondo, phi (ϕ), rispetto all'asse nord-sud. Questi due angoli sono sufficienti a definire la direzione del raggio.

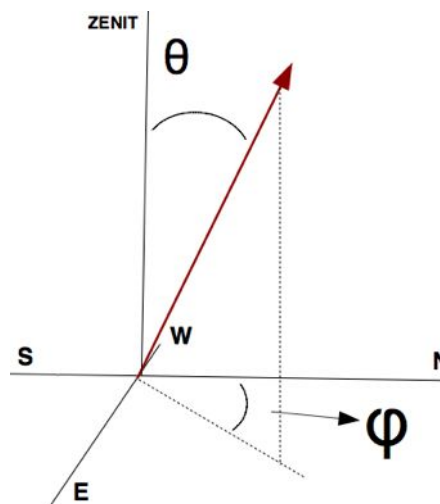


Figura 8: Schema degli angoli θ e ϕ .

Considerato che non si possono avere informazioni a priori su come un raggio possa giungere al rilevatore, ho generato i punti in maniera casuale e uniforme, in modo da coprire tutte le direzioni possibili. Per essere sicuro di coprire tutte le direzioni gli angoli sono, come tipico dei metodi di MonteCarlo, generati a partire da **numeri casuali** uniformemente distribuiti tra 0 e 1. In un secondo momento, per verificare altre ipotesi, ho fatto in modo di escludere alcuni punti e di accettare solo quelli compatibili con l'ipotesi in questione.

L'esclusione di alcuni punti causa però la perdita di dati e quindi una diminuzione della precisione del modello, per questo è necessario, come nel caso del calcolo di pigreco, generare un numero maggiore di dati per ottenere una buona simulazione. Maggior numero di dati, però significa anche maggior consumo di risorse e tempo di calcolo: per questo può essere necessario eseguire le varie operazioni in parallelo attraverso un supercomputer.

5 Raggi cosmici

5.1 Cosa sono e chi li ha scoperti

All'inizio del ventesimo secolo durante alcuni esperimenti sui **raggi gamma** (una forma di radiazione elettromagnetica) era stata rilevata radiazione anche in presenza di schermi sufficientemente spessi, che avrebbero dovuto bloccare tale radiazione. Si pensò quindi all'esistenza di qualcosa di più penetrante dei raggi gamma. L'ipotesi più accreditata fu quella che questo tipo di radiazione provenisse dal suolo.

Nel 1907 Padre Theodore Wulf perfeziona l'**elettrometro**, uno strumento resistente agli urti e alle vibrazioni utilizzato per misurare la radiazione naturale terrestre. Dopo che si scoprì che il segnale dovuto alla radioattività decresce in modo esponenziale quando ci si allontana dalla sorgente di radiazione nel 1910 Padre Wulf prova la sua strumentazione a Parigi sulla Tour Eiffel, a 300 metri di altezza dal suolo: a dispetto di quanto atteso, trova che il segnale diminuisce molto meno del previsto. La discrepanza tra ciò che si attendeva e ciò che verifica poteva essere data dal fatto che forse l'assorbimento in aria della radiazione naturale terrestre è più piccolo di quanto ci si potesse aspettare oppure esiste un'ulteriore sorgente di radiazione nella parte alta atmosfera.

Victor Hess, nel 1911, decise allora di rilevare la **radiazione naturale** a diverse altitudini servendosi di una mongolfiera. Il risultato fu che con l'aumentare della quota la radiazione misurata non diminuiva (come dovrebbe accadere se la radiazione arrivasse dal suolo) bensì si faceva sempre più consistente; dopo vari voli in mongolfiera arrivò a scrivere: "*I risultati delle presenti osservazioni potrebbero essere spiegati assumendo che una radiazione di potere altamente penetrante entra nella nostra atmosfera dall'esterno, ed ancora produce, nei livelli più bassi, parte della ionizzazione osservata in laboratorio*" [10]. La scoperta della presenza di una qualche radiazione proveniente dallo spazio gli valse il Premio Nobel per la fisica nel 1936.

Il termine *raggi cosmici* fu ideato, però, da Robert Andrews Millikan nel 1925 per definire questa nuova radiazione che sembrava provenire indistintamente da tutta la volta celeste. Millikan pensava che i **raggi cosmici** fossero per natura onde elettromagnetiche (raggi γ); altri sostenevano che i raggi cosmici fossero costituiti invece di particelle dotate di massa e non essendo all'epoca conosciute particelle neutre con una considerevole forza penetrante si ipotizzò che queste particelle dovessero essere cariche.

In seguito, intorno al 1930, Compton eseguì un gran numero di misure registrando l'intensità dei raggi cosmici e rilevando una deviazione verso i poli magnetici. Il fenomeno fu definito come effetto latitudine e dimostrò che i raggi cosmici sono composti, per la maggior parte, da **particelle cariche**. La presenza di particelle cariche in prossimità dei poli è dimostrata dalle caratteristiche aurore polari, effetti ottici che si verificano quando una particella carica interagisce con le particelle già presenti nell'atmosfera.

Successivamente si studiò anche l'effetto noto come bottiglia magnetica: una particella carica che arriva, per esempio, dal Sole viene intrappolata dal campo magnetico terrestre in particolare nelle zone dette **fasce di Van Allen**. Quando arriva in una di queste zone la particella, essendo una particella carica in moto in un campo magnetico variabile, è sottoposta a una forza di Lorentz che le fa compiere un particolare moto elicoidale verso uno dei due poli; quando arriva in corrispondenza di

un polo magnetico l'energia cinetica accumulata le permette di rimbalzare e tornare indietro verso l'altro. Questo avviene sia a causa della conformazione del campo magnetico terrestre, più intenso ai poli, sia perché le particelle provenienti dallo spazio sono cariche [15].

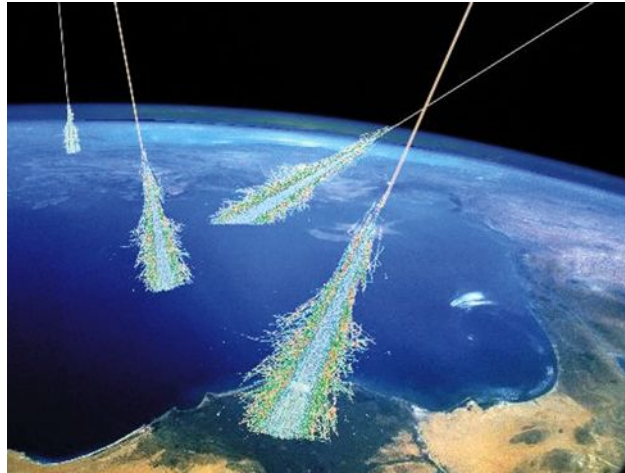


Figura 9: Sciami di diversi raggi cosmici primari arrivano a Terra dopo aver attraversato l'atmosfera.

5.2 Origine e classificazione

I raggi cosmici possono avere un'energia fino a dieci milioni di volte superiore a quella delle particelle prodotte nei laboratori costruiti dall'uomo. Le particelle rivelate dagli strumenti sulla superficie terrestre non sono i raggi cosmici provenienti direttamente dallo spazio (detti raggi cosmici primari). Questi infatti urtano e interagiscono con le **molecole di azoto** presenti nell'alta atmosfera ed avendo una grande energia sono in grado di generare durante questi urti altre particelle che possono a loro volta generarne altre ancora, in modo da formare una cascata di particelle a minore energia detta sciame (figura 9). In questo modo i prodotti dell'interazione di un unico raggio cosmico di alta energia possono arrivare a terra su un'area di diversi chilometri quadrati.

I raggi cosmici, contrariamente a quanto avviene con la luce e le altre radiazioni che viaggiano nel cosmo in **linea retta** e di cui è possibile vedere la provenienza e studiare le caratteristiche della sorgente, vengono **continuamente deviati** nello spazio dalle forze magnetiche che si esercitano su di essi. Pertanto lo studio dell'origine dei raggi cosmici è estremamente complesso: per rendere l'idea sarebbe simile lo studio del Sole e delle stelle da parte di ipotetici abitanti di un pianeta completamente ricoperto di nubi. Essi constatarebbero la presenza di una luce diffusa, ma non saprebbero da quale direzione tale luce proviene, non potendo vedere né il Sole né le singole stelle.

Si ritiene che gran parte dei raggi cosmici abbiano origine da esplosioni (Supernovae) alla fine della vita delle stelle e che vengano accelerati fino alla velocità della luce in seguito a urti fra nuvole di gas negli spazi interstellari. Per quelli più energetici, si è addirittura ipotizzato che essi possano essere prodotti nel **nucleo di lontane galassie**, forse in prossimità di grandi buchi neri centrali. Lo studio dei raggi cosmici costituisce tutt'ora uno dei capitoli più interessanti e misteriosi della fisica e dell'astrofisica [11].

5.2.1 Composizione ed energie

I raggi cosmici sono composti prevalentemente da particelle **cariche positivamente**: in particolare sono costituiti per l'85% da protoni, per il 12% da nuclei di particelle alfa (una particella α equivale ad un nucleo di elio ${}^4\text{He}$, ossia due protoni e due neutroni), per il 2% da elettroni e per l'1% da nuclei pesanti, fotoni e neutrini [14].

I raggi cosmici composti da particelle pesanti (protoni e neutroni) **interagiscono con la materia**, in particolare con l'atmosfera; quelli composti da particelle ad alta energia (fotoni e neutrini) non interagiscono con la materia e per questo possono essere rilevati anche in laboratori sotterranei, come quello presente sotto al Gran Sasso. Peraltro il prendere in considerazione l'effetto delle

montagne nello studio del cosmo non è recente, bensì ci sono tracce di persone che studiano il cielo dalle profondità delle montagne (probabilmente in miniere a cielo aperto) già in autori latini come **Plinio il vecchio** il quale, nella *Naturalis historia*, scrisse: “*Ci sono dei pazzi che si infilano nel fondo delle miniere per osservare le stelle del cielo*” [5].

I raggi cosmici, a seconda della loro provenienza, possono essere divisi in **primari**, rilevabili dallo spazio, che hanno un'energia pari o superiore a $10^{15}eV$ e **secondari**, ossia prodotti dall'interazione di un raggio primario con l'atmosfera, hanno un'energia compresa tra 10^7eV e $10^{15}eV$.

Tanto più un raggio ha un'energia alta tanto più sarà difficile rilevarlo, in quanto eventi con raggi cosmici di alta energia sono molto più **rari** rispetto a eventi a energie minori. Per esempio si rilevano migliaia di **raggi cosmici secondari** a bassa energia per metro quadrato ogni secondo, tuttavia per raggi cosmici ad energie più alte è necessario aspettare anche diversi anni per rilevare una particella su una superficie di un chilometro quadrato (particelle con energia di $10^{20}eV$ vengono rilevate su aree di un chilometro quadrato una volta ogni secolo).

Nella mia esperienza ho rilevato la parte finale di uno sciame, ossia particelle a bassa energia che, in quanto tali, sono molto frequenti e bloccate da ostacoli.

6 La mia esperienza

6.1 Raccolta dei dati

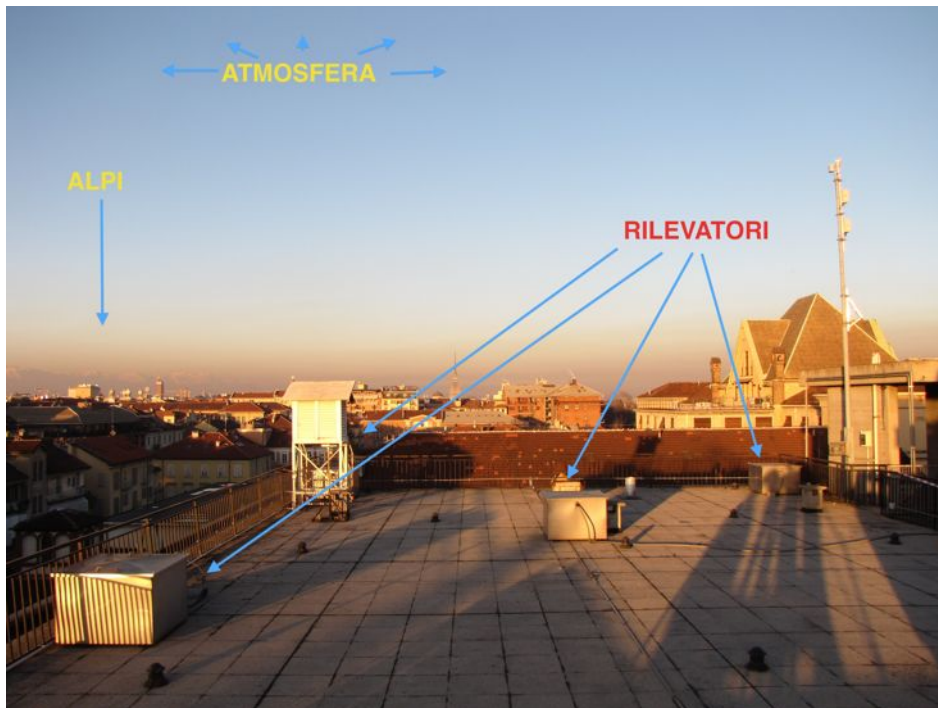


Figura 10: Luogo dell'esperimento (Torino, Italia).

Nella prima fase dell'esperienza per misurare l'angolo di arrivo della particella primaria ho utilizzato cinque particolari **rilevatori**. Questi rilevatori inviano un segnale ad un calcolatore ogni volta che rilevano particelle di uno stesso sciame.

Il rilevatore utilizzato nella mia esperienza è composto da varie componenti tra cui:

uno scintillatore: consiste in una piastra di un particolare materiale **trasparente** in grado di emettere **luce** (sottoforma di uno o più fotoni) quando una particella carica (come quelle che compongono i raggi cosmici) vi passa attraverso. Viene posto in una scatola posizionata sul tetto di un edificio, in modo che i raggi in arrivo non siano ostacolati dalle murature in cemento delle costruzioni;

un fotomoltiplicatore: è un tubo di plastica con una serie di piastre di metallo tra cui viene applicata una differenza di potenziale. Posizionato sopra lo scintillatore è in grado di trasformare la **luce** prodotta dallo scintillatore in un impulso elettrico. Successivamente l'impulso elettrico prodotto viene inviato a due apparecchiature dette ADC e TDC;

Analog to Digital Converter (ADC): riceve l'impulso dal fotomoltiplicatore e misura l'energia della particella rilevata. Invia poi il dato ad un calcolatore;

Time to Digital Converter (TDC): analogamente all'ADC riceve i segnali dai fotomoltiplicatori e calcola i tempi di arrivo delle particelle su ognuno dei cinque rilevatori;

un calcolatore: serve a ricevere i segnali in entrata da ADC e TDC. Tramite appositi programmi riesce a calcolare le direzioni di arrivo sfruttando la differenza nei tempi di ogni singolo rilevatore.

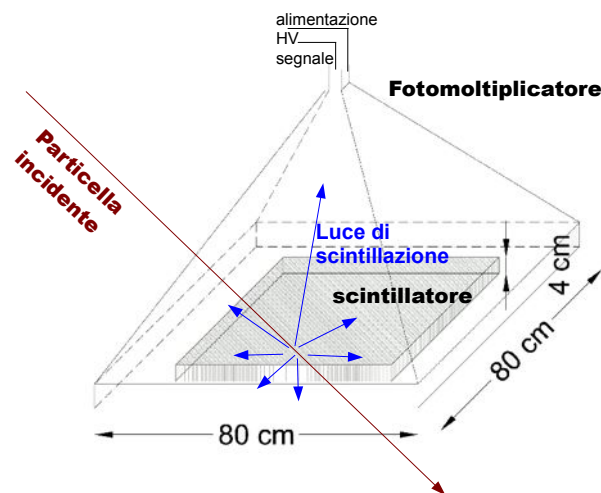


Figura 11: Schema del rivelatore.

Dopo aver raccolto i dati reali ho cominciato a simulare l'arrivo di raggi al supercalcolatore.

6.2 Simulazioni e analisi dei dati

Nella seconda parte dell'esperienza ho utilizzato il supercalcolatore per generare dei raggi cosmici virtuali in diverse maniere per poi studiarli.

Come già detto, per descrivere il **percorso** del raggio cosmico ho immaginato una semisfera virtuale e dal suo centro ho generato, tramite un metodo di Monte Carlo, i vettori che rappresentano la direzione da cui la particella è arrivata.

Durante la generazione di questi punti sulla sfera è necessario tenere in considerazione il fatto che se si generassero i due **angoli** indipendentemente non si otterrebbe un risultato uniforme e ben distribuito, bensì si avrebbe un risultato con pochi punti all'equatore⁴ e molti punti ai poli come in figura 12. Questo avviene perché i punti generati all'equatore nelle zone limitrofe vengono distribuiti lungo una linea molto lunga, al contrario i punti generati ai poli vengono distribuiti su una superficie decisamente più piccola.

Per rimediare a questi inconvenienti si possono utilizzare per calcolare gli angoli le seguenti formule:

$$\theta = |\arccos(u)|$$

$$\phi = 2 * \pi * v$$

Si sono generati due **numeri casuali** (u e v) compresi tra zero ed uno e partendo da questi gli angoli: l'angolo rispetto alla verticale sarà dunque definito come l'arcocoseno in valore assoluto

⁴Nota bene: i termini equatore e poli sono riferiti alla semisfera con il centro sui rilevatori e non alla Terra!

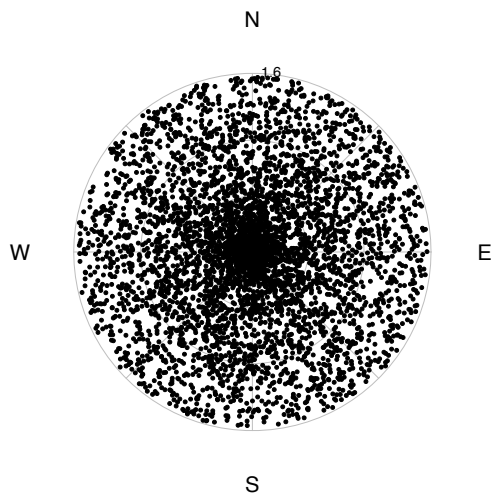


Figura 12: Esempio di distribuzione di punti generati non uniformemente.

(essendo una semisfera si accettano solo i valori positivi) del primo numero generato, il secondo invece è generato uniformemente tra zero e 360 gradi (o $2\pi rad$). In questa maniera si otterrà una distribuzione corretta e **uniforme** (come in figura 14).

Obiettivo della mia esperienza sarà trovare un modo per simulare nella maniera più precisa possibile i dati raccolti scrivendo una simulazione che generi un grafico il più vicino possibile al grafico ottenuto con i dati **rilevati sperimentalmente** (figura 13).

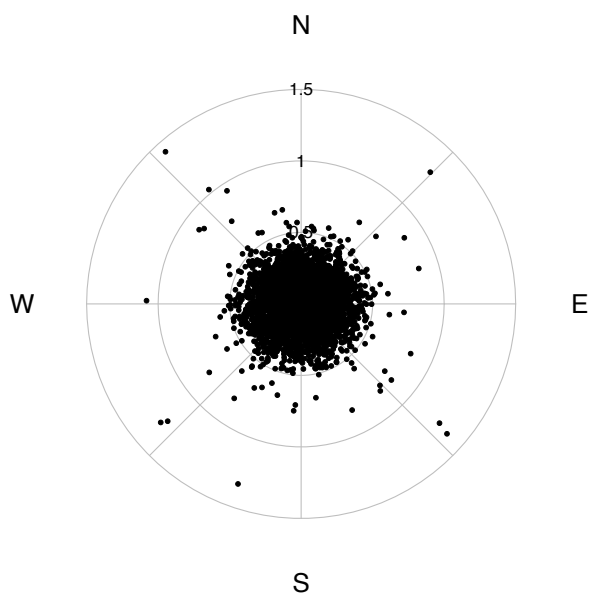


Figura 13: Dati sperimentali.

Per poter trovare la migliore approssimazione ho generato vari modelli: partendo dall'ipotesi più semplice ho introdotto variabili in modo da avere modelli sempre più precisi. In particolare ho realizzato simulazioni:

1. con i raggi provenienti in **modo uniforme** da tutte le direzioni;
2. tenendo in considerazione la presenza dell'**atmosfera**;
3. escludendo i punti che sarebbero arrivati da dietro le **montagne**;
4. **considerando entrambe** le precedenti ipotesi.

6.2.1 Arrivo dei raggi uniforme

Nella prima simulazione ho ipotizzato che tutti i raggi arrivassero indifferentemente da **tutte le direzioni** e senza ostacoli di alcun tipo e ho perciò generato i punti distribuiti uniformemente. Con il supercomputer si può generare un numero differente di punti; un numero di raggi maggiore equivale ad una maggiore precisione, ma necessita di un tempo di calcolo maggiore. Nella figura 14 si nota la differenza tra una simulazione con 100 raggi ed una con 500 decisamente più completa; nella simulazione completa ho generato oltre settecento milioni di raggi!

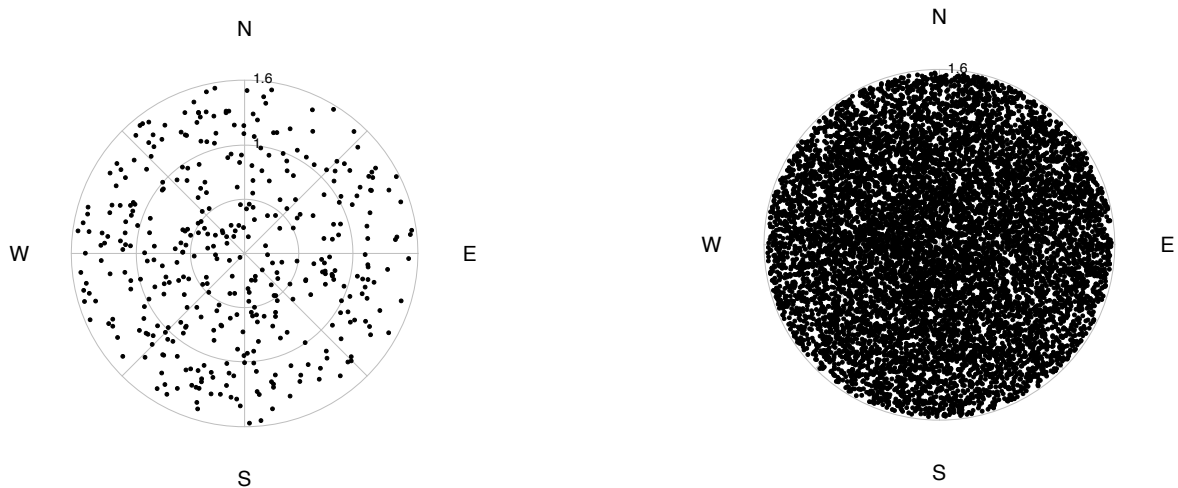


Figura 14: Distribuzioni uniformi con rispettivamente 100 e 500 punti.

In queste due simulazioni il risultato virtuale differisce notevolmente da ciò che si è ottenuto sperimentalmente e mostrato dalla figura 13. Per questo motivo è necessario scartare l'ipotesi di un arrivo dei raggi da ogni direzione in maniera uniforme. Da adesso ogni simulazione prevederà di **escludere** alcuni dati.

6.2.2 Presenza di atmosfera

Per migliorare la simulazione precedente rendendola più precisa e realistica si può introdurre la variabile data dall'**atmosfera**. Come già detto, una particella quando arriva sulla Terra interagisce con le molecole di azoto presente nell'atmosfera: questi urti hanno come conseguenza non solo la creazione di ulteriori particelle, ma anche il rallentamento della particella stessa. Questo significa che con l'aumentare dello spessore dell'atmosfera decresce la **possibilità che la particella stessa arrivi** sulla superficie terrestre e possa essere così rilevata.

Facendo variare il fattore che determina se una particella raggiunge o non raggiunge il rilevatore si ottengono grafici in cui i punti sono concentrati in maniera diversa vicino allo zenit. Durante questa operazione sorge un nuovo problema: se si prova con un fattore basso si ottiene un risultato troppo simile alla distribuzione uniforme e quindi inutile; diversamente tentando con un fattore più alto si ottiene un risultato più preciso. Tuttavia, nel secondo caso, si escludono troppi punti e si rischia di escludere tutti i raggi. Per ottenere un risultato ottimale bisogna generare ulteriori eventi con un dispendio di risorse e di tempo (confronta grafico 3).

Dopo varie ipotesi e tentativi l'**equazione che meglio approssimava gli eventi sperimentali** l'ho identificata in $y = e^{-1,2*x}$, dove y è la probabilità che la particella arrivi sul rilevatore e x lo spessore di atmosfera (in chilometri) attraversato dalla particella; si noti il già citato fattore, il cui valore ideale l'ho riscontrato in 1,2.

In questo caso il risultato è molto simile agli eventi reali (si veda la figura 13), ma non tiene in considerazione l'ipotesi che considera la presenza di montagne nei dintorni del luogo in cui è stato realizzato l'esperimento.

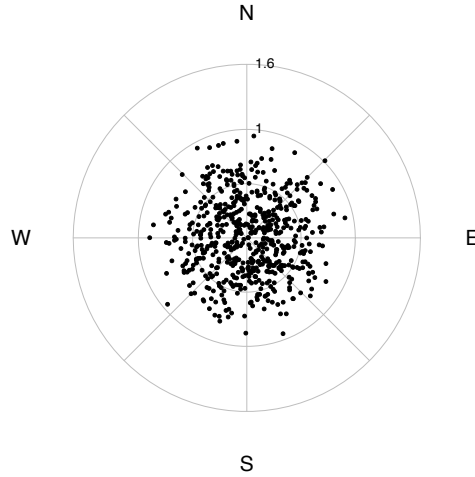


Figura 15: Distribuzione con atmosfera.

La figura 16 mostra invece come si può calcolare lo spessore dell'atmosfera dato l'angolo rispetto alla verticale. In pratica è sufficiente disegnare dei triangoli rettangoli e tramite alcune funzioni trigonometriche e con il teorema di Pitagora ricavare il lato di nostro interesse.

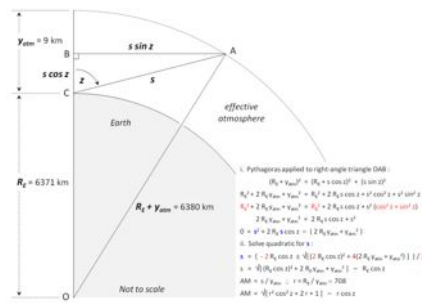


Figura 16: Formule per calcolare lo spessore dell'atmosfera.

6.2.3 Presenza di montagne

Prendendo in considerazione il fatto i dati sperimentali sono stati raccolti a Torino, città molto vicina alle **Alpi** si può pensare che i raggi studiati siano ostacolati da esse e perciò, anche nella simulazione, si possono escludere alcuni dei raggi generati ed ottenere così un risultato più realistico. In questa simulazione ho dunque tenuto in considerazione il fattore delle montagne: se, per esempio, vi è una **montagna** alta 1700m a 10Km si devono escludere dalla simulazione tutti gli angoli theta maggiori di 80,25 gradi $80,25 = \left(\left(\frac{\pi}{2} - \arctan \left(\frac{1,7}{10} \right) \right) * \frac{360}{2\pi} \right)$ in un determinato intervallo dell'angolo phi per l'orizzontale.

La **città di Torino** è relativamente vicina alle Alpi (10 – 20Km in linea d'aria) che possono bloccare le particelle a bassa energia in arrivo da Nord-NordOvest; sempre Torino è altresì presente la collina che, anche se molto bassa, blocca una piccola porzione di raggi a Est come si vede in figura 18.

Come si può notare dalla figura ho completamente escluso i punti che sarebbero giunti dalle direzioni che, nella situazione sperimentale, erano coperte dall'ostacolo delle montagne.

In questa pericolosa situazione però non si è presa in considerazione l'ipotesi della presenza dell'atmosfera e pertanto il risultato, ancora una volta, non coincide con i dati raccolti sperimentalmente (figura 13).

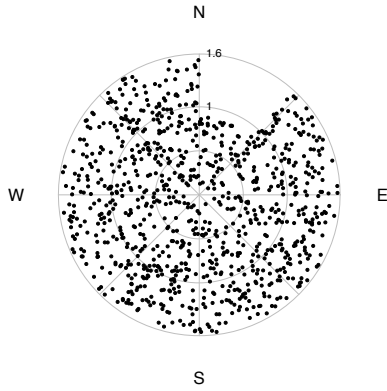


Figura 17: Simulazione con una montagna.

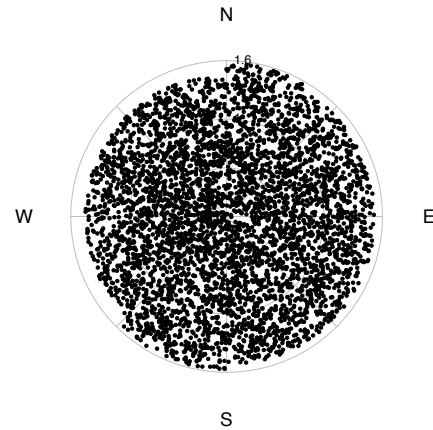


Figura 18: Simulazione con le montagne torinesi.

6.2.4 Presenza sia di montagne sia di atmosfera

Prendendo in considerazione sia le interazioni del raggio con l'atmosfera sia la presenza di montagne nella zona dell'esperimento si può generare una ulteriore simulazione, che produce il risultato in figura 19.

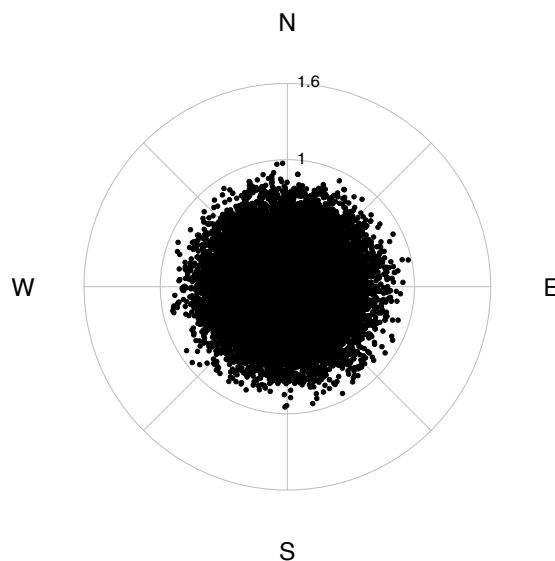


Figura 19: Distribuzione con montagne e atmosfera.

In questo caso i dati ottenuti sono molto simili a ciò che ci si aspettava per eguagliare i dati sperimentali, perciò si può dedurre che sia l'**atmosfera** sia la presenza di **montagne** influiscono sull'arrivo dei raggi cosmici sulla Terra.

7 Conclusioni

In definitiva con quest'esperienza ho verificato che una particella generata dall'interazione di un raggio cosmico con le molecole dell'atmosfera giunge a Terra con maggiore probabilità se lo spessore di atmosfera attraversato è minimo (per esempio lungo la direzione verticale), inoltre le particelle vengono bloccate se incontrano degli ostacoli come le montagne.

Per fare questo ho raccolto i dati di arrivo di alcuni raggi cosmici con gli appositi rilevatori della Facoltà di Fisica a Torino. In un secondo momento ho invece generato alcune simulazioni per cercare di comprendere meglio i dati raccolti sperimentalmente. Considerato l'elevato numero

di punti e di dati da produrre per ottenere un risultato accettabile, non ho utilizzato un normale computer, bensì una macchina del Centro Svizzero di Calcolo Scientifico (mostrata in figura 1), che mi ha permesso di ottenere tutti i dati di cui avevo bisogno in un tempo ragionevolmente breve. Per utilizzare tale macchina ho però dovuto scrivere un programma che facesse eseguire al calcolatore più operazioni in **parallelo**, in questo modo sono stato in grado di sfruttare le potenzialità di una macchina con un numero elevato di nuclei di calcolo.

Nel scrivere il programma ho imparato come gestire un **supercomputer** con i comandi di OpenMP e quando non conviene utilizzare un algoritmo in parallelo. Ho fatto alcuni test per verificare che effettivamente l'esecuzione in parallelo era vantaggiosa. Dopo aver compreso il concetto di simulazione di MonteCarlo e di come applicarla al calcolo parallelo sono passato alle simulazioni vere e proprie.

In un primo momento ho generato una distribuzione uniforme, facendo in modo che a parità di area ci fosse lo stesso numero di punti evitando così di avere un alta densità di eventi ai poli. In seguito ho iniziato a creare i diversi modelli. In particolare ho realizzato più simulazioni ognuna partendo da una differente ipotesi: in questo modo sono riuscito ad avere più grafici (figure: 14, 15, 18, 19) da confrontare con quello creato con i dati sperimentali (figura 13).

Successivamente ho simulato le **interazioni della particella con l'atmosfera**: per fare questo ho scartato alcuni punti in base al risultato di un'equazione. Questa equazione conteneva un fattore e dopo vari tentativi il risultato migliore l'ho trovato in 1,2. In esperienze successive si potrebbe approfondire questo dato: per esempio si potrebbe cercare un legame sperimentale tra questo numero e il numero di decadimenti avvenuti durante la formazione dello sciame. Inoltre durante la mia esperienza ho studiato particelle a **bassa energia** (10^{12} elettronvolt) in quanto hanno un'alta frequenza di arrivo (arriva circa una particella al secondo per metro quadrato); probabilmente particelle a più alta energia avrebbero un fattore più basso in quanto, ipoteticamente, risentirebbero di meno degli urti con le molecole dell'atmosfera; anche questo fatto potrebbe essere approfondito in ricerche successive.

Considerato che le particelle che compongono i raggi cosmici, in particolare quelli a bassa energia da me studiati, non sono in grado di **attraversare la materia**, ho costruito un modello che prevedeva la presenza di ostacoli. Gli ostacoli presenti sperimentalmente erano rappresentati dalle Alpi; per questo ho tolto dalla simulazione tutti i punti in arrivo da dietro le montagne. Tuttavia in questo modo ho escluso solo una piccola quantità di punti, considerato che l'angolo coperto dai monti misura solo pochi gradi.

Un'ulteriore simulazione ha invece tenuto conto di **entrambe le due precedenti ipotesi**: ossia della presenza sia dell'atmosfera sia delle montagne. Quest'ultima simulazione ha dato un risultato molto simile ai dati sperimentali. Come si nota dalla figura 22, in entrambi i casi vi è la presenza di un maggior numero di punti al centro, dove lo spessore dell'atmosfera è minimo, e vi sono meno dati per gli angoli molto alti in presenza delle catene montuose a est e ad ovest.

Si può notare che anche le distribuzioni dell'angolo θ rispetto alla verticale sono simili: hanno un **massimo** in 0 e **decregono** fino ad avere un numero piccolissimo di punti oltre 0,5 radianti, ciò è una diretta conseguenza del fatto che quando lo spessore dell'atmosfera aumenta i raggi non sono in grado di passare. Con ulteriori analisi si potrebbe arrivare ad un parametro più preciso e con più cifre significative in modo da rendere la simulazione ancora più verosimile e corretta: infatti, come si deduce dal grafico, con il parametro attuale la distribuzione simulata finisce di decrescere intorno a 0,7.

In particolare, con riferimento alla figura 16, con la seguente formula:

$$spessore = \sqrt{(R_e * \cos(\theta_{i,j}))^2 + 2 * R_e * Y_{atm} + Y_{atm}^2} - R_e * \cos(\theta_{i,j})$$

(in cui R_e rappresenta il raggio della Terra $\simeq 6400Km$, Y_{atm} lo spessore dell'atmosfera sulla verticale $\simeq 9Km$ e $\theta_{i,j}$ l'angolo preso in considerazione); si può misurare lo spessore dell'atmosfera quando θ vale 0,5. Facendo il calcolo si trova: $10,2534Km$.

Da ciò si può dedurre che quando una particella si trova ad attraversare uno spazio maggiore di poco più di **dieci chilometri** le possibilità che arrivi oltre sono praticamente nulle; anche in questo caso sarebbe interessante indagare per capire da quale fenomeno fisico causi proprio questo valore.

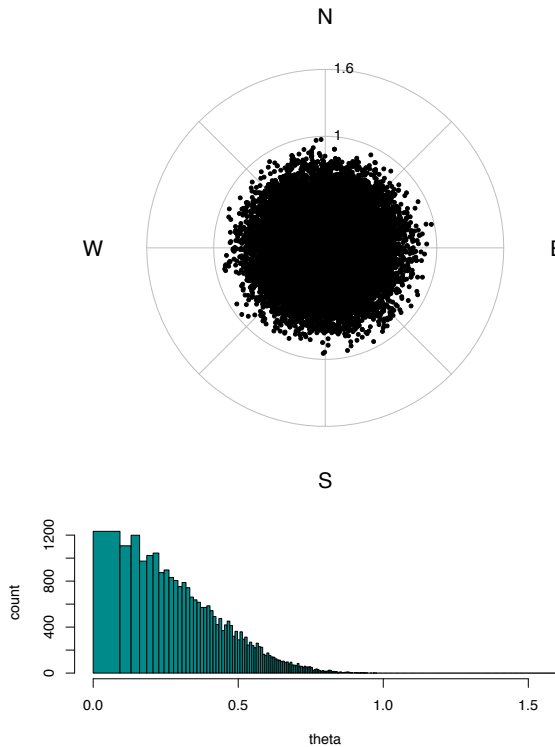


Figura 20: Simulazione

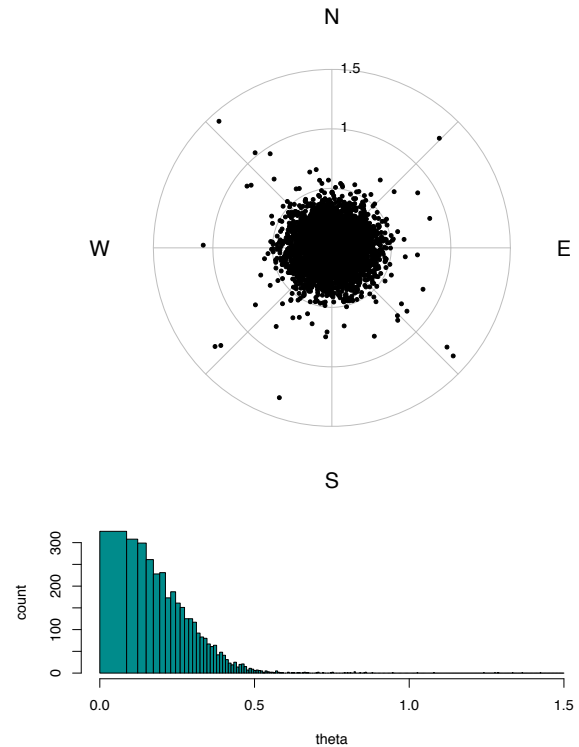


Figura 21: Dati sperimentali

Figura 22: Confronto tra i dati della simulazione e i dati sperimentali.

Un'ulteriore considerazione può essere fatta in merito al fatto che le particelle in questione sono **cariche e dotate di massa**, pertanto sono influenzate sia dal campo gravitazionale e sia dal campo magnetico della Terra. In particolare una particella che arriva a Terra perpendicolarmente alla superficie non subirà deviazioni dovute alla gravità, come accade ad un qualsiasi oggetto in caduta libera; invece le particelle in arrivo con un certo angolo verranno in parte deviate, anche se di un piccolo angolo se considerata la loro microscopica massa, e compiranno una traiettoria parabolica, che le porterà ad effettuare un percorso più lungo e quindi ad avere meno probabilità di giungere a destinazione.

Infine benché lo studio dei raggi cosmici non abbia implicazione dirette, conoscendoli si possono avere maggiori informazioni sulla composizione e sulle dinamiche presenti nel **nostro Universo**. Inoltre studiando alcuni tipi di particelle, può capitare di trovarne di nuove: è il caso del **positrone**, scoperto studiando i raggi cosmici negli anni Novanta e utilizzato oggi anche in alcune analisi mediche.

In conclusione dopo varie ipotesi le simulazioni hanno dato risultati utili e **in grado di spiegare i risultati ricavati sperimentalmente**, dunque l'esperienza può dirsi riuscita.

8 Codice

Di seguito il listato del programma, scritto nel linguaggio Fortran 95 e reso parallelo grazie all'uso di OpenMP, utilizzato per le simulazioni:

```

program raggi

!*****
! Simulation of cosmic rays directions *
5 !* using MonteCarlo's methods          *
!* Results compared with                *
!* real data collected in Turin         *
!* on December 2012.                   *
!* Program written on September 2013.  *
10 !*****

USE omp_lib !Librerias
USE par_zig_mod

15 implicit none
real :: u, v !random numbers
integer :: n, h, i, j, seedi
real :: seedr, start, finish
real :: s, probabilita, z !Private
20 integer, parameter :: m = 960, hit = 750000 !Points
real, allocatable, dimension(:, :) :: theta, phi !Angles
real, parameter :: pi = 4*atan(1.), e = 2.71828182846, Re = 6371.0, Yatm = 9
!constant
Logical, allocatable, Dimension (:, :) :: mont, get
25 character*(1000) :: fileposition, filename, string2, string3, ext
integer :: len2, len3, status2, status3
character*(1000) :: filename

!variables for random
30 integer, parameter :: grainsize = 32
real(4) :: r
real :: time
integer :: len, status, kpar, npar
integer, allocatable :: seed(:)
35 character*(8) :: str

!example of command line ./raggi 8 Files 1 program threads folder filename

n = hit !initalize n
40
!To submit, comment next two lines
WRITE(*,*) "How many points?"
READ(*,fmt = '(I15)') n

45 allocate (theta(m, n)) !allocate array
allocate (phi(m, n))
allocate (mont(m, n))
allocate (get(m, n))

50 !$ start = omp_get_wtime() !Start time

kpar = 0 !Inizialize kpar for non-parallell machines

!Random
55 call random_seed()
call get_command_argument(1, str, len, status)
read(str, '(i2)') npar
allocate(seed(npar+1))
!$ call omp_set_num_threads(npar)
60 call cpu_time(time)
seedr = time*10000
seedi = INT(seedr)
call srand(seedi)
DO h = 1, npar+1
65 r = rand()
seed(h) = r*123+1
ENDDO
call par_zigset(npar+1, seed, grainsize)
!end Random
70
call get_command_argument(2, string2, len2, status2) !Read from command line

```

```

!file position without / (ex. for /Files/file.csv write Files)
read(string2,'(A)') fileposition
call get_command_argument(3, string3, len3, status3) !read file number (ex.
75 !uniform1.csv uniform2.csv
read(string3,'(A)') filename

ext = '.csv' !Set file extension

80 filename = fileposition(:len2)//'/uniform-'//filename(:len3)//ext !File opening
open(unit = 1, FILE = filename, STATUS = 'REPLACE')
filename = fileposition(:len2)//'/atmosphere-'//filename(:len3)//ext
open(unit = 2, FILE = filename, STATUS = 'REPLACE')
filename = fileposition(:len2)//'/mountain-'//filename(:len3)//ext
85 open(unit = 3, FILE = filename, STATUS = 'REPLACE')
filename = fileposition(:len2)//'/real-'//filename(:len3)//ext
open(unit = 4, FILE = filename, STATUS = 'REPLACE')

WRITE(*,*) "Ready"

90 !$OMP PARALLEL
!$OMP DO PRIVATE(kpar, u, v, j) !Uniform
DO i = 1,m
!$ kpar = omp_get_thread_num()+1
95 DO j = 1,n
u = par_uni(kpar)
v = par_uni(kpar)
theta(i, j) = ABS(acos(v)) !generate theta, abs is for semisphere
phi(i, j) = 2*pi*u
100 END DO
END DO
!$OMP END DO

!$OMP SINGLE !Write data
105 DO i = 1,m
DO j = 1,n
WRITE(1,fmt = '(2F10.3,I2)') theta(i, j), phi(i, j)
get(i,j) = .false.
END DO
110 END DO
!$OMP END SINGLE

!$OMP SINGLE
WRITE(*,*) "Ended uniform"
115 !$OMP END SINGLE

!$OMP DO PRIVATE(kpar, j, z, s, probabilita) !With atmosphere
DO i = 1,m
!$ kpar = omp_get_thread_num()+1
120 DO j = 1,n
z = par_uni(kpar)
s = SQRT((Re*cos(theta(i, j)))*2+2*Re*Yatm+Yatm**2)-Re*cos(theta(i, j)) !Atmosphere's
width
probabilita = Prob(s)
IF (z > probabilita) get(i, j) = .true.
125 END DO
END DO
!$OMP END DO

!$OMP SINGLE !Write data
130 DO i = 1,m
DO j = 1,n
IF (get(i, j)) WRITE(2,fmt = '(2F10.3)') theta(i, j), phi(i, j)
get(i,j) = .false.
END DO
135 END DO
!$OMP END SINGLE

!$OMP SINGLE
WRITE(*,*) "Ended atmosphere"
140 !$OMP END SINGLE

!$OMP DO PRIVATE(kpar, j) !With mountain
DO i = 1,m
!$ kpar = omp_get_thread_num()+1
145 DO j = 1,n

```

```

    mont(i, j) = (((theta>1.45).AND.((phi>3.92).AND.(phi<6.28))).OR.((theta>1.23).AND.((phi
    >0.52).AND.(phi<3.14)))) get(i, j) = .NOT.mont(i, j)
    END DO
END DO
!$OMP END DO

150 !$OMP SINGLE      !Write data
DO i = 1,m
    DO j = 1,n
        IF (get(i, j)) WRITE(3,fmt = '(2F10.3)') theta(i, j), phi(i, j)
155     get(i,j) = .false.
    END DO
END DO
!$OMP END SINGLE

160 !$OMP SINGLE
WRITE(*,*) "Ended mountain"
!$OMP END SINGLE

!$OMP DO PRIVATE(kpar, j, z, s, probabilita) !With mountain and atmosphere
165 DO i = 1,m
    !$ kpar = omp_get_thread_num()+1
    DO j = 1,n
        z = par_uni(kpar)
        s = SQRT((Re*cos(theta(i, j))**2+2*Re*Yatm+Yatm**2)-Re*cos(theta(i, j)))      !
        Atmosphere's width
170     probabilita = Prob(s)
        IF (z > probabilita) get(i, j) = .true.
        IF (mont(i,j)) get(i,j) = .false.
    END DO
END DO
175 !$OMP END DO

!$OMP SINGLE !Write data
DO i = 1,m
    DO j = 1,n
180     IF (get(i, j)) WRITE(4, fmt = '(2F10.3)') theta(i, j), phi(i, j)
    END DO
END DO
!$OMP END SINGLE

185 !$OMP END PARALLEL

close(unit = 1)      !File closing
close(unit = 2)
close(unit = 3)
190 close(unit = 4)

!$ finish = omp_get_wtime() !Finish time

Write(*,fmt = '(A8, F10.2, A8)') "Ended: ", abs(finish-start), " seconds"
195 CONTAINS

real function Prob(esse)
real:: esse
200 real, parameter:: a = 1.2 !Factor to try
    Prob = 1 - e**(-a*esse)
end function

end program raggi

205
!*****
!* Copyright (C) 2012 Filippo *
!* *
!* This program is free software: you can redistribute it and/or modify *
210 !* it under the terms of the GNU General Public License as published by *
!* the Free Software Foundation, either version 3 of the License, or *
!* (at your option) any later version. *
!* *
!* This program is distributed in the hope that it will be useful, *
215 !* but WITHOUT ANY WARRANTY; without even the implied warranty of *
!* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *
!* GNU General Public License for more details. *
!* *
!* You should have received a copy of the GNU General Public License *

```

220 | *!* along with this program. If not, see <<http://www.gnu.org/licenses/>>. **
| *!******

Riferimenti bibliografici

- [1] R. AJOLFI, *The fourth crewmember*, Tr@nquillity base, Natale 2013 ed., 2013.
- [2] V. CASTRONOVO, *Un mondo al plurale*, vol. 3a, La nuova Italia, 2012. Manuale di storia.
- [3] CSCS, *Castor*. http://user.cscs.ch/computing_resources/castor/index.html. Webpage di Castor.
- [4] R. FEYNMAN, *Il piacere di scoprire*, Adelphi, Milano, 2002.
- [5] A. GIAMPAOLI, *Gran sasso, cercatori di stelle nel cuore della montagna*. <http://assergiracconta.altervista.org/archivioNews.php?page=1&id=4899>, Maggio 2013.
- [6] R. GUERRA, *Marinetti e il duemila*, Divenire, 3 (2009). Paragrafo: Futurismo e informatica.
- [7] E. HOFMANN, *Evoluzione e prospettive nell'high performance computing*, Mondo Digitale, 2 (2003), pp. 51–65.
- [8] V. IA, *Supercomputing's exaflop target*, Communications of the ACM, 54 (2011), pp. 16–18.
- [9] ISTITUTO DELL'ENCICLOPEDIA ITALIANA, *Treccani.it - vocabolario treccani on line*. <http://www.treccani.it>, 2011.
- [10] M. C. MACCARONE, *Alla ricerca dei raggi cosmici di più alta energia*. http://www.pa.iasf.cnr.it/IASF/outreach/varie/Maccarone_RaggiCosmici.pdf, Ottobre 2004.
- [11] OSSERVATORIO ASTROFISICO DI ARCETRI, *I raggi cosmici*. <http://www.arcetri.astro.it/aaaa/wp-content/uploads/2011/04/raggicosmiciecamera1.pdf>, 2002.
- [12] S. PARISOTTO, *Introduzione al calcolo parallelo per il metodo degli elementi finiti*. http://profs.sci.univr.it/~caliari/aa1213/calcolo_scientifico/seminario_calcoloparallelo.pdf, Aprile 2013.
- [13] A. PORTALE, *Metodo montecarlo ed analisi finanziaria*. [http://web.dmi.unict.it/Public/Uploads/links/Metodo%20Montecarlo%20ed%20Analisi%20Finanziaria\(Portale\).pdf](http://web.dmi.unict.it/Public/Uploads/links/Metodo%20Montecarlo%20ed%20Analisi%20Finanziaria(Portale).pdf).
- [14] A. SANTORO, *Studio delle direzioni d'arrivo dei raggi cosmici di altissima energia con l'osservatorio auger*, Tesi di laurea triennale, (2007).
- [15] P. TIPLER, *Invito alla fisica*, Zanichelli, Bologna, 1991. Manuale di fisica.
- [16] TOP500, *Top 500 supercomputer sites*. <http://www.top500.org/>, 2010.
- [17] WIKIBOOKS, *Supercomputer — wikibooks, pensa liberamente, impara liberamente*, 2013. [Online; visitato il 2-marzo-2014].