

# Supercomputer

Edizione 1.0 19/06/2006

Nota: L'attuale versione del libro è reperibile all'indirizzo:

<http://it.wikibooks.org/wiki/Supercomputer>

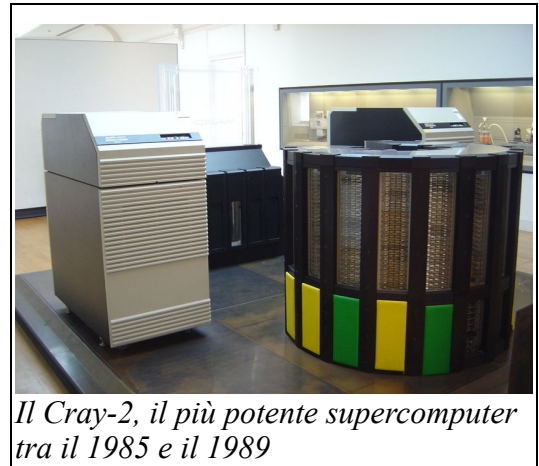
## Indice dei contenuti

Supercomputer.....	1
Introduzione.....	4
Definizione.....	5
Filosofia.....	5
Storia.....	6
Linea del tempo dei supercomputer.....	7
Macchina analitica.....	9
Progetto.....	9
Parziale costruzione.....	9
Zuse Z1.....	10
Zuse Z2.....	10
Zuse Z3.....	10
Funzionamento rispetto a una macchina di Turing universale.....	10
Atanasoff-Berry Computer.....	10
Colossus.....	11
ENIAC.....	12
NORC.....	12
TX-0.....	13
SAGE.....	14
LARC.....	14
IBM 7030.....	15
CDC 6600.....	16
CDC 7600.....	16
CDC STAR-100.....	16
ILLIAC IV.....	17
Cray-1.....	19
Storia.....	19
CDC Cyber.....	20
Cray X-MP.....	20
Descrizione.....	20
Configurazioni.....	21
Cray-2.....	21
ETA-10.....	22
Storia.....	22
Connection Machine.....	22
ASCI Red.....	23
ASCI White.....	24
Earth Simulator.....	24
Blue Gene.....	25
Blue Gene/L.....	25
Storia.....	25
Architettura.....	26
Tassonomia di Flynn.....	27
SISD.....	28
SIMD.....	28
Processori vettoriali.....	28
Array processor.....	28
Array sistolici.....	28
MISD.....	29
MIMD.....	29

Sistemi a memoria distribuita.....	29
Massively Parallel Processing.....	29
Cluster Of Workstations.....	29
Sistemi a memoria condivisa.....	29
Uniform Memory Access.....	29
Cache Only Memory Access.....	30
Macchine dataflow.....	30
Macchine a riduzione.....	30
Calcolo parallelo.....	31
Topologia.....	31
Processori.....	31
Parallelismo massivo.....	31
Calcolo distribuito.....	31
Grid.....	32
Hardware.....	33
Processori.....	33
Memoria.....	33
Altro.....	33
Pipeline.....	34
Processore vettoriale.....	34
Cache.....	35
Uniform Memory Access.....	36
Non-Uniform Memory Access.....	36
Concetti basilari.....	36
Corenza delle cache e NUMA (ccNUMA).....	37
NUMA e cluster.....	37
Raffreddamento a liquido.....	38
RAID.....	38
Software.....	40
Linguaggi di programmazione.....	40
Primitive di sistema.....	40
Programmi.....	40
Legge di Amdahl.....	42
Occam.....	43
Panoramica del linguaggio.....	43
Canali.....	43
Strutture di controllo.....	43
Occam 2.....	44
Occam 2.1.....	44
MPI.....	45
PVM.....	45
openMosix.....	46
Misura delle prestazioni.....	47
Utilizzi.....	48
Bibliografie.....	49
Licenza.....	50

## Introduzione

Questo è un libro prodotto dal sito [it.wikibooks.org](http://it.wikibooks.org) con il lavoro collaborativo degli utenti del sito. La versione aggiornata del libro è disponibile sul sito dove è anche possibile visionare l'elenco completo degli autori accedendo alla cronologia delle singole sezioni. Il libro tratterà la storia, le caratteristiche tecniche e i principali supercomputer del pianeta. Il libro affronterà l'argomento con un approccio non specialistico sebbene per comprendere a pieno le informazioni sono necessarie conoscenze basilari dell'architettura dei computer di programmazione e di architetture dei processori. Il primo capitolo tratta la definizione di cosa è un supercomputer. Nel secondo si descrive perché i supercomputer sono diversi dagli usuali computer. Il terzo capitolo descrive brevemente la storia. Dopo si trova la linea del tempo che raccoglie la lista dei più potenti supercomputer del pianeta nel corso degli anni. Dopo si trovano una serie di brevi schede su alcuni di questi sistemi. Il capitolo successivo introduce la teoria partendo dalla classificazione dei vari sistemi di calcolo. Questa classificazione è molto importante dato che la maggior parte delle architetture indicate nella classificazione sono state sviluppate per il supercalcolo. Nel capitolo successivo si parla brevemente del calcolo parallelo dato che tutti i supercomputer moderni fanno uso di sistemi di elaborazione parallela. Il successivo capitolo parla dell'hardware dei supercomputer infatti vengono trattati solamente alcuni sistemi particolarmente importanti per il supercalcolo. Nel successivo capitolo si parla del software utilizzato dai supercomputer, con particolare riferimento alle infrastrutture software peculiari dei supercomputer. Il penultimo capitolo descrive i sistemi utilizzati per misurare le prestazioni dei supercomputer mentre l'ultimo capitolo parla delle applicazioni del supercalcolo.



## Definizione

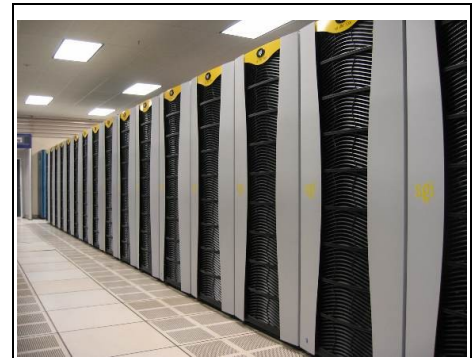
I supercomputer sono dei sistemi di elaborazione progettati per ottenere potenze di calcolo estremamente elevate, dedicati ad eseguire calcoli particolarmente onerosi. Ovviamente sono strumenti costosi, e normalmente sono di proprietà di società o enti di ricerca che condividono il loro utilizzo tra molti dipendenti/ricercatori: e vista la velocità attuale dello sviluppo tecnologico nel campo dell'informatica e dei microprocessori, di solito perdono l'aggettivo "super" dopo pochi anni dalla loro nascita, superati da macchine ancora più potenti.

In passato, i supercomputer erano dotati di architetture più sofisticate e componentistica migliore degli usuali computer al fine di poter svolgere con maggior efficienza le elaborazioni assegnate. Tuttavia in tempi recenti (a partire dagli anni '90) il divario fra la potenza delle CPU per supercomputer e i normali microprocessori commerciali è andato assottigliandosi progressivamente, e l'architettura dei supercomputer ha visto crescere progressivamente il numero di CPU impiegate. Attualmente (2006) la maggior parte dei nuovi supercomputer in progetto e di quelli in funzione sono basati su cluster di migliaia o addirittura centinaia di migliaia di unità di calcolo non molto più potenti di un buon personal computer, connesse da reti locali ad alta velocità e coordinate da estensioni apposite del sistema operativo adottato, generalmente una versione di Unix.

Il parallelismo sempre più massiccio dei supercomputer e in generale delle macchine nate per garantire elevate capacità di calcolo ha stimolato molte ricerche in campo software per esplicitare il parallelismo intrinseco nelle operazioni e nei programmi: queste ricerche hanno portato alla definizione di alcuni linguaggi di programmazione paralleli, come l'Occam, che è senz'altro il più diffuso del gruppo.

## Filosofia

Ciò che differenzia un supercomputer da classico computer non è solo la maggior potenza di calcolo ma anche le architetture utilizzate per sviluppare queste macchine. I classici computer sono progettati secondo l'architettura di Von Neumann dato che le operazioni che l'utente esegue usualmente con la macchina possono essere svolte in modo efficiente da questa architettura. I supercomputer invece spesso analizzano molti dati senza una vera interazione dell'utente dato che usualmente devono eseguire un numero ridotto di operazioni su un elevato numero di dati. Quindi i supercomputer utilizzano architetture alternative che sono in grado di assolvere questi compiti con maggiore efficienza rispetto all'architettura di Von Neumann. Negli ultimi anni questa distinzione comunque si è affievolita dato che il diffondersi dei giochi tridimensionali ha costretto i computer classici e le loro schede grafiche ad adottare soluzioni sviluppate per i supercomputer. Contestualmente l'aumento della potenza di calcolo dei microprocessori per computer e problemi di carattere tecnico legati alla difficoltà di elevare le frequenze di funzionamento delle CPU hanno spinto il settore del supercalcolo verso l'utilizzo di soluzioni *classiche* in modo da abbattere i costi senza penalizzare le prestazioni. Infatti l'attuale supercomputer più potente del pianeta il Blue Gene/L utilizza moltissimi componenti standard e le sue CPU sono dei derivati delle CPU IBM PowerPC 440 con l'aggiunta di un'unità in virgola mobile addizionale per aumentarne le prestazioni.



*Supercomputer Columbia  
assemblato dalla Silicon Graphics*

## Storia

Usualmente si ritiene che i supercomputer siano i più potenti computer del pianeta, questa classificazione comprende tutti i primo computer a partire dalla macchina analitica di Babbage passando per i vari modelli di Zuse (i primi computer funzionanti) fino ad arrivare ai supercomputer dei giorni nostri.

Molti esperti dell'informatica invece fanno risalire i supercomputer agli anni cinquanta e in particolare al NORC prodotto da IBM per via delle soluzioni particolari adottate per ottenere una macchina con le migliori prestazioni e quindi con soluzioni tecniche diverse da quelle utilizzate per i modelli precedenti.

Tra gli anni 60 e la metà degli anni 70 la società CDC con i suoi supercomputer fu l'azienda leader del settore. I tecnici della CDC per ottenere elevate

prestazioni di calcolo svilupparono diverse soluzioni tecnologiche come l'utilizzo di processori specializzati per i diversi compiti (CDC 6600) l'utilizzo di pipeline (CDC 7600) e l'utilizzo di processori vettoriali (CDC STAR-100). Scelte strategiche 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 Seymour Cray che con il Cray-1 segnò una nuova strada per le soluzioni tecnologiche utilizzate e per l'elevata potenza di calcolo sviluppata.

Dalla metà degli anni 70 fino alla fine degli anni 80 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 a liquido o delle strutture a torre ove alloggiare le schede con le unità di calcolo in modo da ridurre la lunghezza media delle connessioni.

Negli anni 90 i supercomputer divennero macchine a parallelismo massivo basate su centinaia se non migliaia di processori elementari. Inizialmente questi processori erano ancora dei processori sviluppati esplicitamente per il supercalcolo come quelli utilizzati dal CM-5/1024 ma verso la fine degli anni novanta ormai si era passati definitivamente a processori generici che fornivano elevate prestazioni e costi ridotti per via della produzione in serie attuata da ditte come Intel o Advanced Micro Devices.

Negli ultimi anni i supercomputer sono macchine sempre più parallele e dotate di un numero sempre maggiore di processori elementari. Esempio eclatante è il Blue Gene/L che con i suoi 131.072 processori è la più potente macchina del pianeta. Questa macchina è dotata di processori PowerPC 440 una variante della famiglia di processori PowerPC G5. Difatti i moderni processori in realtà sono dei complessi programmi che vengono compilati da appositi programmi che generano gli schemi utilizzati dalle fabbriche per produrre i processori. Quindi per adattare un processore generico basta modificare il programma e ricompilarlo.

È da notare che l'Europa dopo un inizio promettente non ha mai trainato la ricerca dell'ambito dei supercomputer e in generale dei computer. Dalla seconda guerra mondiale fino agli anni 90 gli Stati Uniti d'America hanno praticamente posseduto sempre i più potenti computer del pianeta. Ma dagli anni 90 in poi il Giappone ha iniziato ad imporsi come competitore credibile all'egemonia Statunitense grazie a cospicui finanziamenti pubblici a progetti di supercalcolo. Negli ultimi anni il progetto Blue Gene (finanziato dal Dipartimento della Difesa Statunitense) ha rinsaldato la posizione Statunitense ed al momento non sono noti progetti in grado di intaccare la posizione Statunitense nel settore del supercalcolo.



*Il Cray-1 una macchina rivoluzionaria per l'epoca (1976)*

## Linea del tempo dei supercomputer

Periodo	Supercomputer	Velocità di picco	Posizione
1936	Macchina analitica di Babbage	0.3 OPS	RW Munro, Woodford Green, Essex, Regno Unito
1938	Zuse Z1	0.9 FLOPS	Konrad Zuse appartamento dei genitori, Methfesselstraße, Berlino, Germania
1939	Zuse Z2	0.9 OPS	Konrad Zuse appartamento dei genitori, Methfesselstraße, Berlino, Germania
1941	Zuse Z3	1.4 FLOPS	German Aerodynamics Research Institute (Deutsche Versuchsanstalt für Luftfahrt) (DVL), Berlino, Germania
1942	Atanasoff Berry Computer (ABC)	30 OPS	Iowa State University, Ames, Iowa, USA
1942	Telecommunications Research Establishment Heath Robinson	200 OPS	Bletchley Park, Regno Unito
1943	Tommy Flowers Colossus	5 kOPS	Bletchley Park, Regno Unito
1946–1948	University of Pennsylvania ENIAC	50 kOPS	Aberdeen Proving Ground, Maryland, USA
1954	IBM NORC	67 kOPS	Naval Surface Warfare Center Dahlgren Division, Dahlgren, Virginia, USA
1956	Massachusetts Institute of Technology TX-0	83 kOPS	Massachusetts Institute of Technology, Lexington, Massachusetts, USA
1958	IBM Semi Automatic Ground Environment	400 kOPS	23 basi United States Air Force nel Nord America
1960	UNIVAC LARC	500 kFLOPS	Lawrence Livermore National Laboratory, California, USA
1961	IBM 7030 "Stretch"	1.2 MFLOPS	Los Alamos National Laboratory, New Mexico, USA
1964	CDC 6600	3 MFLOPS	Lawrence Livermore National Laboratory, California, USA
1969	CDC 7600	36 MFLOPS	Lawrence Livermore National Laboratory, California, USA
1974	CDC STAR-100	100 MFLOPS	Lawrence Livermore National Laboratory, California, USA
1975	Burroughs ILLIAC IV	150 MFLOPS	NASA Ames Research Center, California, USA
1976	Cray-1	250 MFLOPS	Los Alamos National Laboratory, New Mexico, USA (più di 80 nel pianeta)
1981	CDC Cyber 205	400 MFLOPS	(molti siti nel pianeta)
1983	Cray X-MP/4	941 MFLOPS	Los Alamos National Laboratory Lawrence Livermore National Laboratory Battelle; Boeing
1984	M-13	2.4 GFLOPS	Scientific Research Institute of Computer Complexes, Mosca, Unione Sovietica
1985	Cray-2/8	3.9 GFLOPS	Lawrence Livermore National Laboratory, California, USA
1989	ETA10-G/8	10.3 GFLOPS	Florida State University, Florida, USA
1990	NEC SX-3/44R	23.2 GFLOPS	NEC Fuchu Plant, Fuchu, Giappone
1991	APE100	100 GFLOPS	INFN, Roma, Italia
1993	Thinking Machines CM-5/1024	59.70 GFLOPS	Los Alamos National Laboratory; National Security Agency
1993	Fujitsu Numerical Wind Tunnel	124.50 GFLOPS	National Aerospace Lab, Giappone
1993	Intel XP/S140	143.40 GFLOPS	Sandia National Laboratories, USA

1994	Fujitsu Numerical Wind Tunnel	170.40 GFLOPS	National Aerospace Lab, Giappone
1996	Hitachi SR2201/1024	220.4 GFLOPS	University of Tokyo, Giappone
1996	Hitachi/Tsukuba CP-PACS/2048	368.2 GFLOPS	Center for Computational Physics, University of Tsukuba, Tsukuba, Giappone
1997	Intel ASCI Red/9152	1.338 TFLOPS	Sandia National Laboratories, Albuquerque, USA
1999	Intel ASCI Red/9632	2.3796 TFLOPS	Sandia National Laboratories, Albuquerque, USA
2000	IBM ASCI White	7.226 TFLOPS	Lawrence Livermore National Laboratory, California, USA
2002	NEC Corporation Earth Simulator	35.86 TFLOPS	Earth Simulator Center, Giappone
2004	IBM Blue Gene/L (32,768)	70.72 TFLOPS	United States Department of Energy/IBM, USA
2005	IBM Blue Gene/L (65,536)	136.8 TFLOPS	United States Department of Energy/United States National Nuclear Security Administration/Lawrence Livermore National Laboratory, USA
2005	IBM Blue Gene/L (131,072)	280.6 TFLOPS	United States Department of Energy/United States National Nuclear Security Administration/Lawrence Livermore National Laboratory, USA



## **Macchina analitica**

La macchina analitica (analytical engine in lingua inglese) è un'importante passo della storia dell'informatica. La macchina analitica è stato il primo progetto di un computer meccanico sviluppato per eseguire compiti generici. Il progetto venne sviluppato dal professore di matematica Charles Babbage che cercò anche di realizzarla praticamente. La prima descrizione del progetto si ebbe nel 1837 sebbene Babbage continuò a sviluppare il progetto fino alla sua morte avvenuta nel 1871. Per motivi politici e finanziari la macchina non venne mai sviluppata, ma i moderni personal computer anno notevoli analogie con la macchina analitica pur essendo stati sviluppati quasi cento anni dopo.

Alcuni storici ritengono che con le competenze tecniche dell'epoca la macchina non avrebbe potuto funzionare mentre altri storici ribattono che un supporto politico e finanziario maggiore avrebbe permesso di superare i problemi tecnici e quindi avrebbe permesso la costruzione della macchina.

## **Progetto**

Inizialmente Charles Babbage progetto lo sviluppo della macchina differenziale una macchina meccanica ideata per realizzare dei lunghi tabulati dei logaritmi e delle funzioni trigonometrici. Durante lo sviluppo del progetto Babbage si rese conto di poter progettare una macchina generica molto più potente e interessante dal punto di vista pratico.

La macchina analitica doveva essere alimentata da un motore a vapore e doveva essere lunga più di 30 metri per 10 metri di profondità. I dati d'ingresso e il programma sarebbero stati inseriti tramite schede perforate, un metodo già utilizzati per *programmare* i telai meccanici dell'epoca e che verrà utilizzato anche dai primi computer. I dati di uscita sarebbero stati prodotti da uno stampatore e da un arco in grado di tracciare curve. La macchina sarebbe stata in grado di perforare delle schede per memorizzare dei dati da utilizzare successivamente. La macchina utilizzava una aritmetica in base 10 a virgola fissa. la memoria interna sarebbe stata in grado di contenere 1000 numeri di 50 cifre. L'unità aritmetica sarebbe stata in grado di svolgere le quattro operazioni aritmetiche base.

Il linguaggio di programmazione utilizzato dalla macchina era un linguaggio molto simile al moderno assembler. Cicli e salti condizionati erano previsti nel linguaggio rendendo il linguaggio (e quindi anche la macchina) Turing completo. Tre differenti tipi di schede perforate erano previste. Una tipologia di schede perforate erano riservate alle operazioni matematiche, un altro tipo era previsto per le costanti matematiche e il terzo tipo serviva ad caricare e salvare i dati.

Nel 1842 il matematico italiano Luigi Federico Menabrea a un congresso ascolta Babbage descrive il progetto della macchina analitica. Menabrea realizza una descrizione in francese della macchina. Il lavoro di Menabrea viene tradotto in inglese e significativamente ampliato da Ada Lovelace. Lovelace negli anni successivi realizzare molti lavori teorici per la macchina analitica e in onore del suo lavoro un moderno linguaggio di programmazione è stato chiamato Ada.

## **Parziale costruzione**

Nel 1878 il comitato della British Association for the Advancement of Science si pronuncio contro la costruzione della macchina analitica e questo vanifico la speranze di Babbage di poter accedere a sovvenzioni governative.

Nel 1910 il figlio di Babbage, Henry P. Babbage affermo che una parte della macchina analitica era stata effettivamente costruita e era stata utilizzata per calcolare una lista (errata) di multipli del pi-greco. La parte costruita era una piccola parte del progetto completo, non era programmabile e non era dotata di unità di memorizzazione.

La macchina analitica viene considerato il primo computer mai costruito anche se questa

definizione viene contestata da diversi storici dato che la macchina non è mai stata realmente completata.

## **Zuse Z1**

Lo Zuse Z1 era un computer meccanico creato da Konrad Zuse nel 1937. Era un calcolatore meccanico gestito elettricamente in grado di gestire una limitata programmazione. Il programma era inserito tramite un nastro perforato.



*Riproduzione dello Z1*

## **Zuse Z2**

Lo Z2 era un computer meccanico con relè creato da Konrad Zuse nel 1939. Il computer era un miglioramento del precedente Zuse Z1 infatti utilizzava lo stesso meccanismo a memoria meccanica ma rimpiazzava la parte aritmetica e di controllo con dei relé elettrici.

## **Zuse Z3**

Lo Z3 è il primo calcolatore totalmente programmabile e totalmente automatico, quindi viene spesso indicato come il primo computer della storia. Il suo creatore è Konrad Zuse.

Lo Z3 è composto da 2.200 relè funzionanti a una frequenza compresa tra i 5 e i 10 hertz. Il sistema utilizzava parole lunghe 22 bit. Le operazioni venivano eseguite da un'unità aritmetica in virgola mobile. La macchina venne completata nel 1941 (il 12 maggio dello stesso anno fu presentata a un pubblico di scienziati a Berlino). Il computer originale venne distrutto dagli alleati nel 44 durante un bombardamento alleato di Berlino. Una copia perfettamente funzionante venne realizzata negli anni 60 dalla Zuse KG e è in esposizione permanente al Deutsches Museum. Nel 1998 è stato dimostrato che lo Z3 è Turing completo.

### **Funzionamento rispetto a una macchina di Turing universale**

Lo Z3 permetteva di realizzare loop ma non supportava istruzioni di salto condizionato (anche se non sarebbe stato difficile inserirne una). Tuttavia esiste una possibilità di implementare una macchina di Turing universale con lo Z3 (assumendo un nastro di lunghezza infinita e nessuna possibilità di errore), questa possibilità è stata dimostrata nel 1998. Il metodo utilizzato è scomodo ma la macchina di Turing è un dispositivo scomodo da utilizzare. È nata per essere universale e semplice da utilizzare per dimostrare teoremi non per essere utilizzata con programmi reali.

Da un punto di vista pratico lo Z3 forniva delle istruzioni utili a risolvere problemi di ingegneria del 1940 e serviva per facilitare la vita agli ingegneri non va paragonata con i moderni computer destinati a poter assolvere qualsiasi compito.

## **Atanasoff-Berry Computer**

L'Atanasoff-Berry Computer (spesso chiamato ABC) è stato il primo computer elettronico digitale è uno dei maggiori passi avanti della storia dei calcolatori. Il computer è stato progettato e costruito dal Dr. John Vincent Atanasoff e Clifford E. Berry all'Iowa State University nel periodo 1937-42. L'Atanasoff-Berry Computer introdusse molte innovazioni nel campo dei calcolatori. Introdusse i numeri binari in un computer digitale (lo Zuse Z3 utilizzava i numeri binari ma era un calcolatore elettromeccanico) e la loro gestione, il calcolo parallelo le memorie rigenerative e una separazione tra dati e istruzioni. Per le innovazioni introdotte dall'ABC John Vincent Atanasoff ha ricevuto la National Medal of Technology dal presidente degli Stati Uniti d'America George H. W. Bush in una cerimonia alla Casa bianca il 13 novembre 1990.

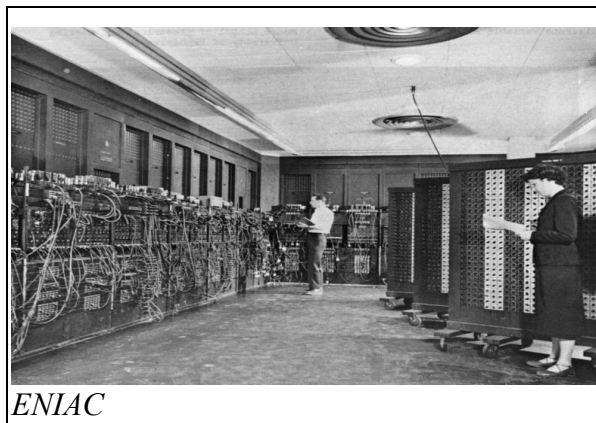


costruiti prima della fine della guerra.

Il Colossus fu sviluppato per forzare i messaggi codificati con il Lorenz SZ 40/42. La macchina comparava due flussi di dati effettuando delle comparazioni basate su operazioni booleane. Un flusso è il messaggio da decifrare mentre il secondo è un flusso generato dalla macchina che cerca di individuare la chiave di codifica effettuando delle prove e confrontando i risultati. Se una simulazione dimostra un grado di accuratezza superiore a una specificata soglia il risultato veniva stampato tramite una macchina da scrivere elettrica. Si stima che sviluppasse circa 5000 operazioni al secondo e quindi tra il 1943 e il 1946 fu il più potente computer del pianeta. Nel 1946 fu presentato l'ENIAC.

## **ENIAC**

L'ENIAC (Electronic Numerical Integrator And Computer) è considerato il primo computer elettronico della storia, progettato presso l'University of Pennsylvania da J. Presper Eckert e John Mauchly e completato due anni prima dell'invenzione del transistor, nel 1946. Tra il 1946 e il 1954 fu il più potente computer del pianeta fino a quando l'IBM non presentò il NORC.



Come tutti i primi computer occupava una stanza di grandi dimensioni, ed era costituito da diversi componenti, una sorta di "armadietti" che contenevano circa 18.000 valvole termoioniche. Tali valvole si bruciavano frequentemente durante

il funzionamento della macchina e dovevano essere costantemente sostituite da un addetto che girava con un carrello pieno di valvole e controllava periodicamente il loro funzionamento.

La programmazione avveniva tramite cavi di collegamento sui pannelli, che dovevano essere ogni volta scollegati e ricollegati, ed ogni configurazione consentiva al computer di risolvere un diverso problema.

L'attività dell'ENIAC era finanziata inizialmente dalla Marina degli Stati Uniti, interessata al calcolo delle traiettorie balistiche che il computer poteva effettuare in una grande varietà di situazioni diverse, trovando soluzioni numeriche di alcune equazioni differenziali (da qui deriva il nome di integratore numerico). Successivamente l'ENIAC venne anche utilizzato per scopi civili, come la classificazione dei dati dei censimenti.

L'ENIAC venne utilizzato anche per applicazioni scientifiche. Per esempio venne utilizzato da John von Neumann per realizzare la prima previsione del tempo al computer del pianeta. In quel particolare esperimento processò 250.000 operazioni in virgola mobile in circa 24 ore e realizzò una previsione di 24 ore dai dati di ingresso.

## **NORC**

Il NORC (Naval Ordnance Research Calculator) è un supercomputer elettronico della prima generazione basato su valvole termoioniche prodotto dalla IBM per la United States Navy Bureau of Ordnance. Il sistema entrò in servizio nel dicembre del 1954 e a quei tempi era il più potente computer del pianeta. Fu superato nel 1956 dal TX-0.

La macchina utilizzava delle valvole CRT o valvola di Williams per memorizzare e gestire le 3600 parole della memoria con un tempo di accesso di 8 millisecondi. Ogni parola consiste in 16 cifre decimali utilizzando quattro bit per ogni cifra più due per la verifica degli errori. Ogni parola può memorizzare 13 cifre con due indici o una istruzione. NORC utilizza un insieme di 66 valvole in

parallelo come memoria. Ogni valvola memorizza un bit dei 66 memorizzati per ognuna delle 900 parole in modo che ognuno dei quattro insiemi di 66 tubi memorizzi 900 parole.

La velocità del NORC era di circa 15.000 operazioni per secondo. Una addizione richiedeva 15 microsecondi mentre una moltiplicazione richiedeva 31 microsecondi e una divisione 227 microsecondi senza contare il tempo necessario per accedere alla memoria. Il sistema era capace di eseguire operazioni in doppia precisione sebbene fossero usate raramente.

Il sistema principale era formato da 1982 componenti, ognuno composto da uno o più valvole e della relativa elettronica. Il sistema era composto da 62 tipi diversi di valvole sebbene metà dei circuiti utilizzassero solo un tipo di valvole e l'80% dei circuiti utilizzassero solo 6 tipi di valvole. In totale il sistema era formato da 9.800 valvole e 10.000 diodi.

Il NORC aveva otto nastri magnetici che potevano leggere e scrivere fino a 71.500 caratteri al secondo. Il sistema era dotato di due stampanti in grado di stampare 150 linee per minuto, anche se poteva essere utilizzata solo una stampante alla volta. Aveva anche un lettore di card che poteva leggere 100 card al minuto, con quattro parole in ogni card.

Il computer fu presentato alla Marina statunitense il 2 dicembre 1954. Alla cerimonia di presentazione, calcolò le prime 3089 cifre di pi greco, il che era un record per quel tempo. Il calcolatore funzionò solo per 13 minuti. Nel 1955 il NORC venne spostato alla Naval Proving Ground nel Dahlgren Virginia. Fu il migliore computer disponibile fino al 1958 quando vennero acquistati computer più moderni. Venne utilizzato fino al 1968. Il suo progetto influenzò l'IBM 701 e la serie IBM 700.

## **TX-0**

Il TX-0 (acronimo di Transistorized eXperimental computer ZERO spesso chiamato tixo) era un supercomputer totalmente a transistor con una memoria di sistema della notevole capacità (per i tempi) di 64.000 parole di 18 bit. Il TX-0 venne attivato nel 1956 e rimase operativo per tutti gli anni 60. Tra il 1956 e il 1958 fu il più veloce supercomputer del pianeta fino a quando fu superato dal sistema SAGE.

Sviluppato dai Lincoln Laboratory del Massachusetts Institute of Technology utilizzava largamente i transistor e era in effetti il primo esperimento su vasta scala di un computer basato su transistor con una tale memoria di sistema. Il sistema in effetti era una versione a transistor del computer Whirlwind, un altro progetto dei Lincoln Labs. Ma mentre il Whirlwind riempiva il piano di un edificio il TX-0 occupava una singola stanza ed era veloce. Come il Whirlwind il TX-0 era equipaggiato con uno schermo inserito in un contenitore da oscilloscopio di 12 pollici. L'uscita era uno schermo di 512 x 512 pixel in uno schermo di 7 x 7 pollici.

Il TX-0 non è stato progettato per essere utilizzato come sistema a se stante. La macchina disponeva di una memoria di 64.000 parole e quindi utilizzava un indirizzamento a 16 bit ma per ridurre i costi si decise di utilizzare parole di soli 18 bit. Venivano utilizzati solo 2 bit per le istruzioni infatti la macchina disponeva solo di quattro istruzioni. Queste istruzioni consentivano di memorizzare un dato, sommare dei dati e effettuare un salto. La quarta istruzione consentiva di accedere a una seconda serie di istruzioni che potevano essere utilizzate singolarmente o collegate. Queste istruzioni addizionali richiedevano ulteriori 10 microsecondi per essere eseguite.

Con il termine del TX-0 i lavori del laboratorio si concentrarono sul progetto TX-1. Tuttavia il progetto dovette affrontare molte difficoltà per via della complessità e in seguito venne riconvertito ad un progetto più modesto il TX-2 del 1958. Il nucleo della memoria per il tempo era molto grande e buona parte deriva da quello del TX-0 che fu cannibalizzato per realizzare il TX-2. In seguito il TX-0 non è stato considerato molto interessante e prestato semipermanentemente al laboratorio di elettronica del MIT dal giugno del 1958 dove è diventato il centro di quello che in seguito è diventato il MIT Artificial Intelligence Lab.

Trasportato al dal lincoln Labs con solo 4 K di memoria il computer non era più costretto a utilizzare 16 bit per indirizzare la memoria e in seguito si espanse il numero di istruzioni portandole da 4 a 16 e un registro indice venne aggiunto. Questo rese la programmazione della macchina molto più semplice ma consentiva l'espansione della memoria a 8 K. L'espansione della memoria consentì lo sviluppo di molte grandi innovazioni dell'informatica come il riconoscimento della voce, il riconoscimento della scrittura e vennero sviluppati molti progetti come programmi di scrittura e debugger.

## **SAGE**

SAGE, è l'acronimo di Semi Automatic Ground Environment è un sistema automatico di rilevamento, inseguimento e intercettazione di aerei nemici del North American Aerospace Defense Command utilizzato tra gli anni 50 e gli anni 80. Nel momento che il sistema fu completamente operativo la minaccia degli aerei sovietici era ormai totalmente sostituita dalla minaccia dei missili internazionali sovietici.

Tuttavia il SAGE fu un sistema molto innovativo, era un sistema online basato su computer, gestito in tempo reale che comunicava tramite modem. Usualmente il sistema viene considerato uno dei più avanzati e estesi sistemi informatici mai sviluppati.

Il lavoro di IBM per il progetto SAGE (il progetto e la realizzazione dei computer AN/FSQ-7, le valvole termoioniche i nuclei di ferrite delle memorie alla base del computer Whirlwind II) fu un importante fattore di successo della società e sicuramente rafforzò il ruolo di leader di IBM nel settore dell'informatica. I computer del SAGE rappresentarono i più veloci sistemi di calcolo del pianeta tra il 1958 e il 1960 quando venne presentato il LARC.



*SAGE Sector Control Room. Lo schermo mostra la mappa degli Stati Uniti nel settore nord contrale. Le immagini sono fornite su concessione MITRE*

## **LARC**

L'UNIVAC LARC (Livermore Advanced Research Computer) fu il primo tentativo del Remington Rand di costruire un supercomputer. Fu progettato per il multiprocesso con 2 CPU (chiamati Computer) e un processore dedicato all'input/output (chiamato processor).

Solo due LARC sono stati costruiti.

Il primo fu costruito per Lawrence Livermore National Laboratory nel giugno 1960.

Il secondo fu costruito per il Navy's David Taylor Model Basin.

Il NARC è un computer con un'architettura con parole di 48 bit. Il sistema utilizzava una notazione bi-quinaria a codifica decimale con 4 bit per cifra con parole di 11 cifre (negative e positive). Il set di istruzioni era di 48 bit, ogni cifra della macchina è dotata di un bit di parità per la verifica degli errori. In sostanza ogni parola occupava 60 bit (48 per i dati e 12 per la parità). La configurazione base prevede 26 registri generici che potevano essere espansi fino a 99. Il sistema accedeva ai registri in un microsecondo.

La configurazione base era dotata di un solo computer che poteva essere espanso con un secondo computer.

Il processor era una CPU indipendente (con un diverso set di istruzioni rispetto ai computer) che provvedeva a gestire 12 o 24 unità a tamburo magnetico, tra quattro e quaranta unità a nastro UNISERVO II, due registratori elettronici, una stampante ad alta velocità e un lettore di carte ad alta velocità.

Il nucleo di memoria del LARC era formato da 2.500 parole per banco e alloggiava quattro banche per ogni armadio di memoria. La configurazione base era composta da 8 banche (due armadi), 20.000 parole. La memoria poteva essere espansa fino a un massimo di 39 banchi (10 armadi con un banco libero) 97.500 parole al massimo. Il sistema implementava un bit di parità per ogni cifra con parole formate in totale da 60 bit. la memoria aveva un tempo di accesso di 8 microsecondi e un tempo di ciclo di 4 microsecondi. Ogni banco operava in modo indipendente degli altri e impiegava 4 microsecondi ad accedere a un dato se non era occupato. Utilizzando un accesso sequenziale ai banchi di memoria si poteva ottenere un tempo di accesso di 4 microsecondi (per esempio ponendo i dati in un banco e le istruzioni in un altro banco).

Il data transfer bus collegava i due computer e il processor alla memoria tramite una connessione con multiplexer per massimizzare il trasferimento. Ogni quattro microsecondi il bus era diviso in otto slot da 500 nanosecondi:

\*Processor - istruzioni e dati

\*Computer 1 - istruzioni

\*Computer 2 - dati

I\*/O DMA Synchronizer - dati

\*Non usato

\*Computer 2 - istruzioni

\*Computer 1 - dati

I\*/O DMA Synchronizer - dati

Il nucleo della memoria utilizzava un sistema di gestione a priorità per impedire un accesso simultaneo allo stesso banco di memoria da parte dei due computer o da parte di un computer e del processor per impedire conflitti o deadlock. I banchi di memoria sono non disponibili per 4 microsecondi dopo un accesso a una cella di memoria del banco. Se un'altra sezione del banco deve essere letta o scritta questa rimane bloccata fino a quando l'operazione non può essere svolta. Per prevenire deadlock e timeout dell'I/O il sistema era dotato di priorità come indicato:

\*I/O DMA Synchronizer - alta priorità

\*Processor media priorità

\*Computers - bassa priorità

Quando un'accesso ad alta priorità è in esecuzione il nucleo di memoria è bloccato per 4 microsecondi, nel secondo ciclo le operazioni a bassa priorità sono comunque non eseguite se esistono operazioni a priorità elevate. I cicli a bassa densità sono eseguiti solo se non vi sono cicli ad alta priorità.

Il LARC venne sviluppato con una tecnologia basata su transistor già obsoleta quando la prima macchina venne presentata. Comunque il LARC era un computer molto veloce per i tempi. Eseguiva addizioni in 4 microsecondi, moltiplicazioni in 8 microsecondi e divisioni in 28 microsecondi. Fu il più veloce supercomputer del pianeta tra il 1960 e il 1961 fino a quando IBM presentò l'IBM 7030.

## **IBM 7030**

L'IBM 7030, conosciuto anche come *Stretch*, fu il primo supercomputer prodotto da IBM. Il primo 7030 venne installato a Los Alamos nel 1961.

Inizialmente venduto a 13.5 milioni di dollari statunitensi in seguito il prezzo venne ridotto a 7.78 milioni di dollari per via delle prestazioni non all'altezza delle aspettative e il prezzo con i clienti

che avevano già acquistato la macchina fu rinegoziato. Nonostante il 7030 fosse molto più lento di quanto ci si aspettasse comunque il supercomputer fu la più veloce macchina del pianeta tra il 1961 e il 1964 quando venne superato dal CDC 6600.

### **CDC 6600**

Il CDC 6600 fu il primo supercomputer costruito dalla Control Data Corporation nel 1963. In generale viene considerato il primo supercomputer con un ampio successo commerciale. Divenne il più veloce supercomputer della sua era essendo tre volte più rapido del precedente IBM 7030. Rimase il più potente computer del pianeta tra il 1964 e il 1969 quando venne sorpassato dal successore CDC 7600.

### **CDC 7600**

Il CDC 7600 era un supercomputer sviluppato dal Seymour Cray come successore della CDC 6600 e prodotto dalla Control Data Corporation. Il sistema permise alla CDC di rafforzare la propria dominazione nel settore dei supercomputer per buona parte degli anni 70. Era circa 10 volte più rapido del precedente CDC 6600 e era in grado di sviluppare fino a 36 Megaflops con codice compilato appositamente per la macchina. Sebbene il 7600 condividesse con il 6600 molte caratteristiche come la dimensione delle istruzioni da 60 bit per parola il codice non era compatibile con il CDC 6600. In aggiunta il codice sorgente assembler COMPASS non era compatibile, alcune istruzioni del 7600 non erano presenti sul CDC 6600 e vice-versa. Tra il 1969 e il 1974 fu il più potente supercomputer del pianeta fino a quando fu superato dal CDC STAR-100 sempre prodotto dalla Control Data Corporation.

### **CDC STAR-100**

Lo STAR-100 era un supercomputer prodotto dalla Control Data Corporation e fu la prima macchina a utilizzare processori vettoriali per migliorare le prestazioni nel calcolo matematico. Purtroppo alcune caratteristiche della macchina influirono negativamente sulle prestazioni della macchina e infatti le prestazioni reali della macchina furono molto inferiori rispetto a quanto preventivato. Le prime installazioni commerciali furono effettuate nel 1974, la CDC fu spinto a immettere la macchina per mantenere la sua supremazia nel settore dei supercomputer anche perché dopo due anni sarebbe stato presentato il Cray-1.

Lo STAR era la prima macchina CDC ad architettura a 64 bit. La sua architettura non era simile al CDC 6600 o al CDC 7600. Era una macchina con una CPU tipo-RISC con il supporto di diversi processori secondari che si occupino di effettuare compiti secondari gravando il processori da questi compiti e consentendogli di effettuare unicamente operazioni aritmetiche. Nella STAR sia la CPU che i processori periferici erano deliberatamente semplificati per abbattere il costo e la complessità della macchina.

Nuove e complesse istruzioni furono rese disponibili attraverso l'APL programming language e lo stesso linguaggio permetteva l'elaborazione e gestione di ampi vettori di dati. La memoria fisica aveva parole di 512 bit chiamate SWORD (Superword) e era divisa in 32 banchi indipendenti. La CPU fu progettata per utilizzare le istruzioni supplementari e l'hardware aggiuntivo della macchina per ridurre al minimo i tempi di accesso alla memoria. Per esempio un programma che somma 400 numeri può essere composto da una singola istruzione e dal vettore composto dai quattrocento dati (che possono essere al massimo 65.535). La cpu decodificherà la singola istruzione e poi inizierà ad accedere alla memoria per effettuare le quattrocento somme senza dover più decodificare altre istruzioni. Come nel caso di sistema a pipeline le prestazioni del sistema migliorano se la cpu conosce in anticipo le istruzioni da eseguire e se queste istruzioni sono tra loro condizionate.

Per rendere le prestazioni dello STAR migliori del CDC 7600 le pipeline dello STAR furono rese



più profonde. Mentre il 7600 manteneva fino a 8 istruzioni contemporaneamente lo STAR era in grado di mantenere fino a 25 istruzioni in modo da poter utilizzare tutte le unità di calcolo. Per poter ottenere elevate prestazioni le pipeline vettoriali dovevano essere costantemente piene altrimenti le pipeline rallentavano il sistema invece di velocizzarlo. Per lo STAR il punto di guadagno era di circa 50 dati altrimenti l'utilizzo delle pipeline vettoriali non era conveniente dal punto di vista delle prestazioni.

Quando nel 1974 la macchina fu presentata era chiaro che le prestazioni reali della macchina non erano quelle che la maggior parte delle persone si aspettava. Realizzare programmi vettoriali era difficile dato che spesso i dati da elaborare e le stesse istruzioni dipendevano dai risultati delle operazioni precedenti e quindi non potevano essere efficacemente vettorializzati. Le prestazioni dichiarate potevano essere ottenute solo in rari casi. Inoltre le prestazioni delle operazioni non vettoriali erano state sacrificate per velocizzare le operazioni vettoriali e quindi quando i programmi non utilizzavano queste istruzioni le prestazioni decadevano drammaticamente.

Lo STAR-100 fu una delusione per molte persone a lo stesso capo progetto Jim Thornton abbandonò la CDC per andare alla Network System Corporation. Una versione successiva e migliorata venne in seguito rilasciata come CDC Cyber 203 e in seguito venne ulteriormente migliorata come CDC Cyber 205, ma in quel periodo già esisteva il Cray-1 una macchina ad alte prestazioni notevolmente migliore del CDC Cyber.

## **ILLIAC IV**

L'ILLIAC IV è uno dei più famosi e malfamati supercomputer della storia. Fu l'ultimo di una serie di computer di ricerca dell'University of Illinois. Il progetto dell'ILLIAC IV è un progetto basato su un parallelismo massivo con 256 processori sviluppato per trattare grandi quantità di dati, un'idea che è alla base dei processori vettoriali. La macchina fu alla fine presentata nel 1976 dopo una decade di sviluppo ed era molto in ritardo, molto costosa e con prestazioni molto inferiori a macchine come il Cray-1.

Verso i primi anni 60 lo sviluppo dei computer si indirizzava verso una diminuzione della complessità in modo da renderli più veloci potendo implementare direttamente in hardware le istruzioni. Infatti aggiungere funzioni poteva rendere il computer più lento dato che il clock della macchina dipende dalla velocità delle operazioni e il clock massimo è fissato dall'istruzione più lenta della macchina. Lo stato dell'arte del disegno dell'hardware di allora prevedeva il progetto dei singoli transistor che realizzavano le funzioni logiche della cpu. L'aggiunta di funzioni logiche aumenta il numero di transistor e quindi rallenta il sistema. Per raggiungere la massima velocità i progettisti di cpu dovevano cercare un compromesso ottimo.

Molte soluzioni vennero esplorate alla fine degli anni 60. Una di queste era l'utilizzo di una struttura chiamata pipeline. Normalmente una cpu preleva un'istruzione, la decodifica l'esegue e salva il risultato. Tutte queste operazioni vengono eseguite da un'unità polifunzionale che può eseguire una singola istruzione alla volta. In una struttura a pipeline esistono diverse unità specializzate che eseguono questi compiti in modo parallelo, come in una catena di montaggio. La struttura a pipeline fu una delle maggiori innovazioni di Seymour Cray che rese il CDC 6600 fino a dieci volte più veloce dei suoi concorrenti.

Un'altra soluzione al problema era il calcolo parallelo. Progettare computer con un certo numero di cpu generiche. Un computer progettato in questo modo fornisce buone prestazioni solo quando il problema da affrontare può essere agevolmente suddiviso in un sottoinsieme di problemi più piccoli che ogni cpu affronta in modo indipendente. Non tutti i problemi possono essere parallelizzati difatti questo è tuttora un grande problema per l'informatica. L'uso di cpu generiche può essere molto costoso ma in un disegno massivamente parallelo si può utilizzare anche delle cpu più semplici.

La Westinghouse Electric Corporation esplorò questa soluzione con il progetto Solomon. Loro

notarono che i computer erano utilizzati principalmente per eseguire calcolo matematici per gli ingegneri e gli scienziati. Loro svilupparono una cpu che si sarebbe occupata di controllare diverse unità di calcolo (che oggi chiamiamo ALU) che si occuperanno di svolgere i calcoli in parallelo. Ogni unità di calcolo eseguiva lo stesso programma ma con dati diversi un concetto conosciuto come SIMD. Con un contratto dell'US Air Force RADC research arm svilupparono un prototipo di scheda madre nel 1964 ma il contratto con il RADC terminò e Westinghouse decise di non continuare il progetto.

Uno dei principali sviluppatori del progetto Solomon Daniel Slotnick si trasferì all'University of Illinois e convinse l'università a continuare la ricerca. Nel 1964 l'università firmò un contratto con il DARPA per finanziare il progetto che in seguito sarebbe diventato l'ILLIAC IV seguendo la linea di ricerca già iniziata da altri progetti. Burroughs era un collaboratore primario e doveva fornire gli hard disk ad alta velocità oltre a provvedere allo sviluppo del Burroughs B6500 un computer utilizzato per gestirli. Texas Instruments aveva un contratto per la realizzazione di diversi circuiti integrati Emitter coupled logic (ECL). ILLIAC è stata la prima macchina a utilizzare gli ECL. Lo sviluppo venne avviato nel 1965 e il primo progetto fu terminato nel 1966.

Il progetto puntava ad ottenere un computer in grado di processare un miliardo di istruzioni per secondo (un Gigaflops). Per realizzare questo si era previsto di utilizzare 256 unità elementari di calcolo a 13 Mhz gestite da quattro CPU. L'ILLIAC era progettato con un'architettura a 64 bit, ogni unità elementare aveva una memoria di 2048 parole per memorizzare i risultati. Le CPU erano in grado di accedere a tutta la memoria della macchina a differenza delle unità di calcolo, infatti queste per questioni di semplicità potevano accedere solo alla loro zona di memoria. Ogni unità di calcolo era dotata di 6 registri per uso generico e poteva spostare i dati dai registri verso un registro di una delle otto unità di calcolo circostanti.

Inizialmente si pensava di alloggiare tutte le 256 unità di calcolo in un solo computer ma in breve si comprese che il progetto andava modificato e semplificato. Si decise di dividere le unità in blocchi da 64 da alloggiare in armadi separati. Visti i ritardi di progetto e costruzione ci si rese conto che realisticamente si poteva realizzare un solo armadio e quindi le prestazioni iniziali da 1 Gigaflops venivano ridotte a 200 Megaflops.

Il lavoro all'università si concentrò primariamente sulla realizzazione di trasferimento efficiente di dati verso le unità di calcolo. I problemi di questa tipologia di computer è che se non eseguono software parallelo non sono più veloci di altri sistemi di calcolo. Per permettere l'esecuzione di codice parallelo furono sviluppati molti linguaggi di programmazione come l'IVTRAN e il TRANQUIL, delle versioni parallele del Fortran e il Glypnir una conversione parallela dell'Algol. Normalmente questi linguaggi fornivano un supporto diretto degli array di dati da processare in modo da sfruttare le unità di calcolo parallele.

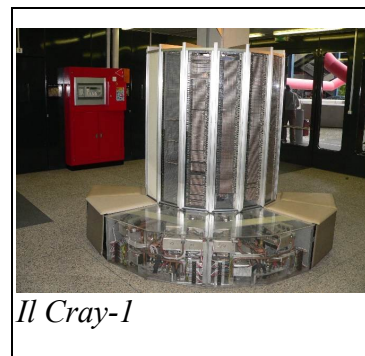
Mentre il computer era sotto sviluppo alla fine degli anni 60 si svolsero diverse proteste all'università dato che il progetto era realizzato con sovvenzioni militari e molti ritenevano che l'università fosse in combutta con il ministero della difesa. Le proteste raggiunsero l'apice il 9 maggio 1970 il giorno delle Illazioni. Dopo l'attentato del 24 agosto all'edificio di matematica dell'University of Wisconsin l'Università decise di abbandonare il progetto e il progetto passò in mano alla NASA che per via del Progetto Apollo aveva un'enorme disponibilità economica e era interessata ad ogni sviluppo tecnologico che potesse essere utilizzato in ambito spaziale. La NASA creò la Advanced Computing division, e la macchina venne spostata a Moffett Field California, casa dell'Ames Research Center.

Lo spostamento rallentò lo sviluppo della macchina infatti questa fu completata solo nel 1972. Dagli iniziali 8 milioni di dollari del progetto del 1966 il costo finale fu di 31 milioni di dollari e le prestazioni si erano ridotte dagli iniziali 1 Gigabyte a 100 Megaflops o 150 Megaflops di picco. Nonostante i problemi la macchina se doveva trattare problemi parallelizzabili era dalle 2 alle 6 volte più veloce di un CDC 7600. Per la NASA quella era una macchina perfetta dato che i problemi di fluidodinamica computazionale sono molto parallelizzabili.

Sebbene l'ILLIAC fu terminato nel 1972 la macchina non era affidabile. Continuava ad aver problemi di affidabilità che gli ingegneri tentarono di risolvere fino a quando nel 1974 fu in grado di eseguire il primo programma completo e divenne operativa. Non stante la macchina fosse dichiarata operativa funzionava solo dal lunedì al venerdì e era in grado di richiedere fino a 40 ore di manutenzione alla settimana. La prima applicazione completa fu pronta per il 1976, lo stesso anno che venne presentato il Cray-1 un supercomputer con le stesse prestazioni. La macchina fu quindi il più potente computer del pianeta anche se solo per pochi mesi dato che il Cray-1 era in grado di fornire l'80% di prestazioni in più a costi inferiori e con affidabilità maggiore. Tuttavia la macchina venne utilizzata negli anni seguenti e anzi a Ames svilupparono la loro versione di Fortran. Infine nel 1982 la macchina venne dismessa e l'advanced computing division venne disciolto.

## **Cray-1**

Il Cray-1 è un supercomputer sviluppato da un team di progettisti guidati da Seymour Cray per la Cray Research. Il primo Cray-1 venne installato ai Los Alamos National Laboratory nel 1976, e divenne famoso come uno dei più famosi e meglio riusciti supercomputer della storia. Fu il più potente supercomputer del pianeta tra il 1976 e il 1981 quando venne presentato il CDC Cyber 205.



*Il Cray-1*

## **Storia**

All'inizio degli anni 70 Cray stava lavorando per la Control Data Corporation a una nuova macchina chiamata CDC 8600, il successore logico del CDC 6600 e CDC 7600. L'8600 era fondamentalmente quattro 7600 inseriti in una singola scatola con una modalità speciale di funzionamento che gli consentiva di lavorare all'unisono con una modalità tipo SIMD.

Jim Thornton, inizialmente collega di Cray durante il progetto aveva lanciato un progetto più radicale, il CDC STAR-100. A differenza dell'approccio a forza bruta dell'8600 lo STAR seguiva una strada differente. Il processore principale dello STAR era meno potente di quello del 7600 ma lo STAR era dotato di hardware aggiuntivo che velocizzava alcune operazioni molto comuni nei supercomputer.



*Particolare della torre*

Nel 1972 l'8600 era giunto alla fine. La macchina era incredibilmente complessa e era diventata ingestibile. Bastava che un singolo componente si rompesse per bloccare la macchina. Cray andò a parlare con William Norris, il Control Data e CEO della società, affermando che era necessaria una riprogettazione completa. In quel periodo la società stava subendo una crisi finanziaria e lo stesso STAR era stato un insuccesso. Norris quindi non aveva i soldi per finanziare una riprogettazione della macchina.

Cray abbandonò la società e ne fondò una nuova con sede a poche centinaia di metri dalla CDC che lui e altri ex dipendenti della CDC avevano acquistato a Chippewa Falls WI. Il gruppo iniziò a cercare un'idea; inizialmente l'idea di realizzare un supercomputer sembrava impossibile, poiché se anche la CDC non aveva avuto abbastanza soldi per realizzare il progetto come avrebbe potuto fare una piccola compagnia appena nata e praticamente senza fondi. Ma il CTO della società fece degli incontri con alcuni agenti di Wall Street e trovò investitori disposti a finanziare la realizzazione di un nuovo supercomputer. Quindi alla società di Cray mancava oramai solo un nuovo progetto.

Nel 1975 il Cray-1 venne annunciato. L'eccitazione dovuta alla nuova macchina creò una guerra tra i centri LLNL e Los Alamos su chi dovesse essere il primo cliente. Los Alamos vinse il primo esemplare e nel 1976 venne installato il Cray-1 con matricola 001 per un periodo di prova di alcuni mesi. Il National Center for Atmospheric Research (NCAR) divenne cliente ufficiale della Cray Research nel luglio del 1977 pagando 8.86 milioni di dollari (7.9 milioni di dollari e 1 milione di dollari per i dischi). La macchina del NCAR venne dismessa nel gennaio del 1989. La compagnia si aspettava di vendere una dozzina di macchine ma in realtà vendette più di ottanta macchine a un prezzo compreso tra i 5 milioni e gli 8 milioni di dollari. Le sue macchine resero Cray una celebrità nel suo ambiente e la sua compagnia un successo fino all'inizio degli anni 90 quando il mercato dei supercomputer si sgonfiò.



*Particolare del sistema di raffreddamento*

## **CDC Cyber**

I supercomputer della famiglia Cyber sono dei mainframe sviluppati dalla Control Data Corporation durante gli anni 70 e gli anni 80. La linea Cyber include modelli molto diversi di calcolatori. La serie 170 fu basata sui CDC 6600 e CDC 7600, la linea 180 fu sviluppata in Canada, la linea 200 fu basata sul CDC STAR-100 e la linea CYBERPLUS o Advanced Flexible Processor (AFP). Questi sistemi furono sviluppati per svolgere le usuali operazioni di un supercomputer sebbene diversi modelli inclusero dei [processori vettoriali per incrementare le prestazioni delle macchine in alcuni compiti specifici. Tra il 1981 e il 1983 il CDC Cyber 205 fu il più potente supercomputer del pianeta fino all'introduzione del Cray X-MP/4.

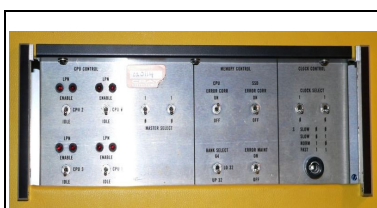
## **Cray X-MP**

Il Cray X-MP era un supercomputer sviluppato, costruito e venduto dalla Cray Research. Questa fu la prima macchina della società a essere basata sui processori vettoriali e è la quarta generazione di supercomputer. Fu presentata nel 1982 ed era il successore del Cray-1 (1976), fu il più veloce supercomputer del pianeta nel 1983 ma nel 1984 venne superato dal supercomputer M-13. Il principale sviluppatore fu Steve Chen.

## **Descrizione**

La macchina venne sviluppata con una configurazione a ferro di cavallo nella parte inferiore. I processori funzionavano a 10 nanosecondi (il precedente Cray-1A eseguiva un ciclo in 12.5 nanosecondi) sviluppando una velocità teorica di 200 megaflops per processore o di 800 megaflops per quattro processori installati su una macchina. I processori supportavano la matematica parallela, le pipeline e la memoria condivisa tra le varie pipeline del processore.

Inizialmente il sistema era fornito con il sistema operativo proprietario Cray Operation System (COS) e con UniCOS (un derivato del System V). Dal 1984 l'UniCOS divenne il primo sistema operativo della macchina.



*Pannello di controllo del CRAY-XMP48*



*CRAY-XMP48 all'École Polytechnique Fédérale de Lausanne in Svizzera*

## Configurazioni

Il X-MP era venduto con uno, due o quattro processori e da uno a sedici megaword (da 8 a 128 Megabyte) di memoria di sistema RAM. Inizialmente il sistema era in grado di indirizzare fino a 16 megaword per via dell'indirizzamento a 24 bit ma in seguito il XMP/EA estese l'indirizzamento fino a 2 gigaword teorici. In pratica il massimo realizzato fu un computer con 64 megaword. Il XMP/EA aveva un clock di 8.5 nanosecondi e un picco teorico di 942 megaflops. Nel 1982 X-MP/48 veniva venduto a 15 milioni di dollari più il costo degli hard disk.



*Sistema di raffreddamento  
CRAY-XMP48*

## Cray-2



Il Cray-2 è un supercomputer vettoriale sviluppato dalla Cray Research nel 1985. Quando venne presentato era la macchina più veloce del pianeta e rimpiazzava

il precedente Cray X-MP. Il computer venne superato dall'ETA10-G nel 1990.

Dopo la presentazione del fortunato Cray-1 Seymour Cray iniziò immediatamente a sviluppare il suo successore. Nel 1979 Cray decise di abbandonare il suo precedente lavoro alla CDC dato lo scarso supporto che questa forniva al suo lavoro. Cray fondò con altri componenti della sua squadra la Cray Research. Inizialmente i laboratori erano posizionati a Chippewa Falls, vicino agli uffici della CDC ma in seguito si spostarono nel nuovo quartiere generale a Boulder. Nei nuovi laboratori iniziarono lo sviluppo della generazione successiva di macchine. I laboratori vennero chiusi una decina di anni dopo quando venne inaugurata la nuova sede a Colorado Springs.

Cray per incrementare la velocità della macchina lavorò su più soluzioni in contemporanea. Più unità funzionali nello stesso sistema per ottenere un elevato parallelismo, contenitori di ridotte dimensioni per ridurre le latenze di collegamento e componenti veloci per innalzare la velocità di clock. Un esempio classico è il CDC 8600 che include quattro macchine tipo CDC 7600 in una sola macchina cilindrica di 1 x 1 metro funzionante a 125 Megahertz (8 nanosecondi). Sfortunatamente una simile densità e un ciclo di clock così elevato porta anche degli inconvenienti. Un singolo malfunzionamento comprometteva l'intera macchina

Una soluzione di questo problema era l'integrazione dei singoli circuiti in circuiti integrati. Ogni circuito integrato contiene una serie di moduli predisegnati che vengono opportunamente programmati per funzionare. Questo permette una produzione in serie che abbassa i costi e ne innalza l'affidabilità infatti i chip vengono testati singolarmente prima della vendita. L'8600 era progettato con l'utilizzo di tecnologia basata su MOSFET, una tecnologia semplice ma che non forniva la velocità richiesta da Cray. Innovazioni avvenute nella seconda metà degli anni 70 permisero a Cray di utilizzare una nuova tecnologia costruttiva dei circuiti integrati che innalzò la velocità fino a 80 Megahertz (12.5 nanosecondo) del Cray-1. Infatti il Cray-1 era molto



*Il Cray-2. Dentro il cilindro  
si trovano le CPU e il  
liquido di raffreddamento.  
A lato gli hard disk*



*Interno del Cray-2*



*Dettaglio del lato superiore*

più veloce del 8600 dato che integrava molta più logica funzionante a frequenza maggiore. Sebbene lo sviluppo tecnologico continuava a incrementare integrazione dei chip la loro dimensione fisica era invece limitata da problemi di natura meccanica. Comunque i rapidi incrementi dell'elettronica consentì un rapido incremento della complessità dei microprocessori. Questi miglioramenti tecnologici furono molto apprezzati da Cray dato che questo puntava a migliorare di 10 volte le prestazioni della macchina rispetto al precedente Cray-1. Nel Cray-2 quindi si decise di seguire un approccio simile al CDC 8600 infatti la macchina fu progettata con una velocità doppia del precedente e con un elevato numero di unità elementari.

## **ETA-10**

L'ETA10 era una linea di supercomputer sviluppati dalla ETA System (una divisione della Control Data Corporation) negli anni 80. La linea di supercomputer utilizzava lo stesso set di istruzioni del CDC Cyber 205.

## **Storia**

Dopo la divisione dalla CDC nel settembre del 1983 ETA si concentrò sullo sviluppo di computer con un ciclo di clock inferiore ai 10 nanosecondi. Questo obiettivo spinse ETA a sviluppare molte innovazioni. Tra queste ETA sviluppo sistemi di raffreddamento a liquido e utilizzo circuiti CMOS per i microprocessori.

L'ETA10 raggiunse gli obiettivi della società infatti sviluppava 10 gigaflops e molti modelli avevano un ciclo di clock di 7 nanosecondi, un ciclo molto ridotto per gli standard della metà degli anni 80. Nel 1989 l'ETA10-G/8 divenne il più potente computer del pianeta. Fu superato l'anno successivo dal NEC SX-3/44R.

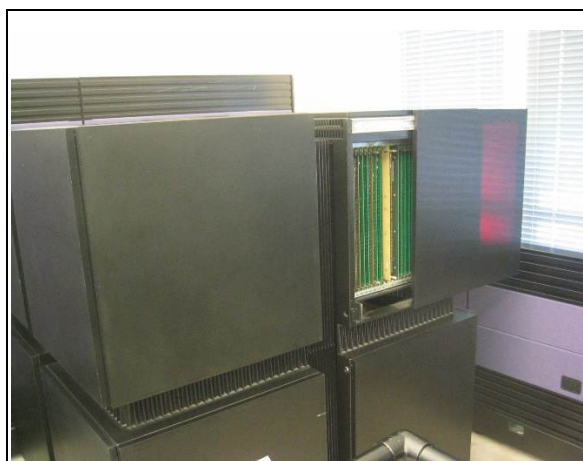
Si presuppone che in seguito la società si fosse concentrata sul successore dell'ETA10 e cioè l'ETA30 in sistema in grado di sviluppare 30 gigaflops.

ETA venne reincorporata nella CDC e cesso di esistere come società il 17 aprile 1989.

## **Connection Machine**

Connection Machine sono una serie di supercomputer sviluppato da Danny Hillis all'inizio degli anni 80 al Massachusetts Institute of Technology in contrapposizione alla tradizionale architettura di Von Neumann. Il CM-1 inizialmente concepito al MIT era un computer massivamente parallelo basato su una topologia ad ipercubo che collegava i vari nodi. Ogni nodo era un processore molto semplice dotato di una propria memoria che eseguivano codice SIMD. Le Connection Machine inizialmente erano progettate per applicazioni di intelligenza artificiale e di calcolo simbolico, sebbene ebbero successo nelle scienze applicate che richiedevano elevate potenza di calcolo.

Hillis e Sheryl Handler fondarono la Thinking Machines a Waltham nel Massachusetts (in seguito si spostarono a Cambridge Massachusetts) nel 1983 e insieme al loro team svilupparono il CM-1 e il CM-2 che in alcune configurazioni poteva arrivare ad avere fino a 65.536 processori. I singoli processori erano molto semplici, processavano un bit alla volta. Il CM-2 presentato nel 1987



*Thinking Machines CM-2 esposto al Computing Museum in San Jose. Un pannello è stato parzialmente rimosso per mostrare l'elettronica interna*

aggiunse 3132 coprocessori in virgola mobile Weitek. Ogni coprocessore è condiviso da 32 processori. Due varianti del CM-2 vennero prodotte, il piccolo CM-2a con 4096 o 8192 processori e il veloce CM-200.

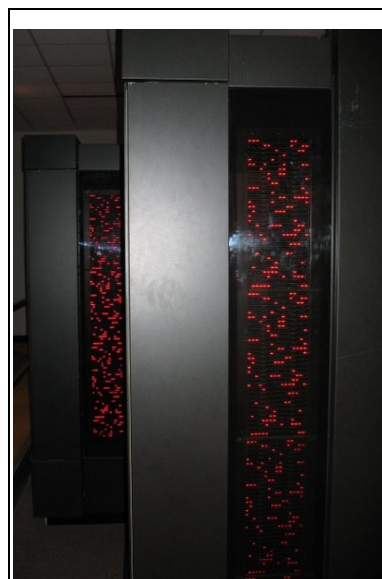
Il pannello luminoso del FROSTBURG, il CM-5, esposto al National Cryptologic Museum. Il pannello viene utilizzato per sorvegliare il funzionamento dei singoli nodi e per la diagnostica

Data la loro iniziale designazione al campo simbolico e all'intelligenza artificiale i processori a singolo bit del CM-1 e il CM-2 sono stati influenzati da linguaggio di programmazione lisp e in particolare da una variante del Common Lisp \*Lisp (pronunciato Star-Lisp) il primo linguaggio di programmazione implementato nel CM-1. Molti programmi per CM-1/2 sono scritti in \*Lisp.

Con l'arrivo nel 1991 del CM-5 la Thinking Machines abbandonò l'architettura a ipercubo del CM-2 per una nuova architettura MIMD basata su una rete Fat tree che gestisce una serie di processori RISC SPARC. L'ultimo CM-5E rimpiazzò i processori SPARC con i veloci SuperSPARC e nella configurazione a 1024 processori divenne il più potente supercomputer del pianeta nel 1993, posizione che mantenne per pochi mesi prima che il Numerical Wind Tunnel della Fujitsu non lo superasse.

L'elenco completo delle macchine Connection Machine in ordine cronologico è :CM-1, CM-2, CM-200, CM-5 e CM-5E.

Le macchine Connection Machines sono famose per il loro design coreografico. Il CM-2 era un Cubo grigio con diversi LED rossi sulla superficie. Il CM-5 come visibile a lato è una torre con un ampio pannello pieno di diodi LED. Durante il funzionamento i LED continuano a lampeggiare per indicare il funzionamento dei nodi e infatti la macchina è stata utilizzata dal film Jurassic Park nella sala di controllo dell'isola.



*Il pannello luminoso del FROSTBURG, il CM-5, esposto al National Cryptologic Museum. Il pannello viene utilizzato per sorvegliare il funzionamento dei singoli nodi e per la diagnostica*

La tesi originale di Danny Hillis da cui sono nate le macchine Connection Machine si intitolava: The Connection Machine (MIT Press Series in Artificial Intelligence) (ISBN 0262081571) . Il libro è in stampa dal 2005 e descrive in modo accurato la filosofia della macchina, il software e la sua architettura inclusi i collegamenti tra le CPU, le memorie e la loro programmazione in lisp.

## **ASCII Red**

ASCII Red o ASCII Option Red, è un supercomputer installato ai Sandia National Laboratories a Albuquerque New Mexico.

Il progetto nasce da una collaborazione tra Intel e i laboratori Sandia, una parte del progetto governativo statunitense Accelerated Strategic Computing Initiative (ASCI).

Il sistema è stato costruito come fase uno dell'Accelerated Strategic Computing Initiative (ASCI) del United States Department of Energy e del National Nuclear Security Administration per sviluppare un simulatore in grado di rimpiazzare i test nucleari sotterranei che una moratoria firmata dal presidente statunitense George H. W. Bush nel 1992 e estesa da Bill Clinton nel 1993. ASCII Red ha iniziato ad operare nel 1997.

È formato da una rete a maglia (38 X 32 X 2) MIMD massively-parallel processing consistente in 4.510 nodi di calcolo, 1212 Gigabyte di memoria distribuita e 12.5 Gigabyte di memoria su disco. Inizialmente la macchina utilizzava processori Pentium Pro alla frequenza di 200 Mhz ma in seguito

venne aggiornata con l'utilizzo di processori Pentium II Overdrive. Attualmente il sistema è formato da un totale di 9298 Pentium II Overdrive a 333 Mhz. Il sistema è composto da 104 armadi che occupano una superficie di 230 metri quadrati. Il sistema è stato progettato prevedendo l'utilizzo di componenti non dedicati e prevedendo un'architettura molto scalabile.

Il primo ASCI Red è stato il primo supercomputer del pianeta a superare la 1 Teraflops secondo il test Linpack (1996) e quindi è diventato il più potente computer del pianeta secondo la TOP500. Dopo l'aggiornamento ai Pentium II Overdrive il sistema raggiunto la potenza di calcolo di 2 Teraflops. Il sistema venne superato nel 2000 dal ASCI White prodotto da IBM.

Differenti parti del sistema utilizzano differenti sistemi operativi. Per la programmazione viene utilizzata una normale macchina uni] con una versione distribuita del sistema operativo Unix creata da Intel, il Paragon XP/S Supercomputer. La gestione dei nodi di calcolo viene affidata a una versione molto leggera del sistema operativo Cougar un derivato del sistema operativo SUBMON per sistemi Paragon.

## **ASCI White**

ASCI White è il nome di un supercomputer dei Lawrence Livermore National Laboratory in California.

Il sistema è composto da un cluster di computer IBM RS/6000 SP. Il sistema è formato da 512 macchine collegate tra loro, ogni macchina ha 16 processori e quindi il totale la macchina è dotata di 8.192 processori, 6 Terabyte di memoria RAM e 160 Terabyte di memoria su disco. Nonostante l'elevato numero di processori questa è una macchina relativamente lenta (per gli standard del 2006) dato che ogni processore funziona a una frequenza di soli 375 Mhz. Di conseguenza ogni programma in esecuzione normalmente occupa decine se non centinaia di processori. Il sistema pesa 106 tonnellate e consuma 3 MegaWatt di corrente elettrica e un numero analogo per raffreddare il sistema. Il sistema è in grado teoricamente di sviluppare 7.226 Gigaflops e utilizza il sistema operativo AIX dell'IBM.

ASCI White è formato da tre sistemi separati, White con 512 nodi, Ice don 28 nodi e Frost con 68 nodi.

Il computer è stato assemblato a Poughkeepsie New York. Terminato nel giugno del 2000 p stato trasportato in un apposito centro in California dove è stato reso operativo il 15 agosto 2001. Le prestazioni teoriche sono di 12.300 Gigaflops anche se il test Linpack ha misurato prestazioni reali inferiori del 40%. Il sistema è costato 110 milioni di dollari. Tra il 2000 e il 2002 è stato il più potente supercomputer del pianeta fino a quando il supercomputer Earth Simulator non lo superò.

Il sistema è stato costruito come terza fase del Accelerated Strategic Computing Initiative (ASCI) avviato dall'United States Department of Energy e dal National Nuclear Security Administration per costruire un simulatore in grado di rimpiazzare i test nucleari proibiti dalla moratoria firmata dal presidente statunitense George H. W. Bush nel 1992 ed ampliata da Bill Clinton nel 1993.

## **Earth Simulator**

L'Earth Simulator (ES) è stato il più potente supercomputer del pianeta tra il 2002 e il 2004. Il sistema è posizionato nell'Earth Simulator Center (ESC) a Kanazawa-ku (ward), Yokohama-shi, Giappone. Il computer è in grado di sviluppare 35.86 Teraflops di operazioni in virgola mobile al secondo. Il sistema è stato sviluppato dal National Space Development Agency of Japan, Japan Atomic Energy Research Institute, e dal Japan Marine Science and Technology Center nel 1997 per studiare le evoluzioni del clima su scala globale. La costruzione inizio nell'ottobre del 1999 e venne completata nel febbraio del 2002. Il sistema divenne operativo ufficialmente l'11 marzo 2002. Il costo del progetto è stato di 7.2 miliardi di yen. L'Earth Simulator venne superato dal prototipo Blue Gene/L il 29 settembre 2004.



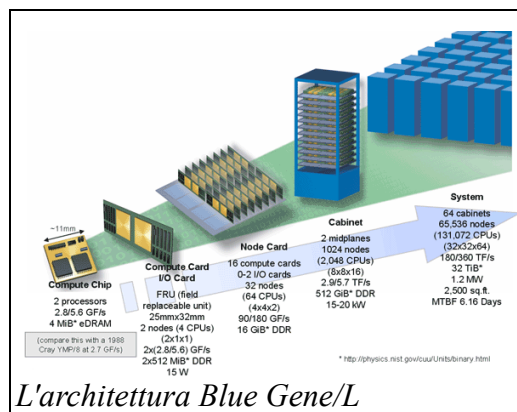
Costruito dalla NEC l'ES è basato sull'architettura NEC SX-6. Il sistema è formato da 640 nodi con 8 processori vettoriali e 16 Gigabyte di memoria RAM per ogni nodo. Il sistema nel complesso ha 5120 processori e 10 Terabyte di memoria RAM. Due nodi occupano 1 metro x 1.4 metri x 2 metri di cabinet. Ogni cabinet consuma 20 KVA di potenza. Il sistema ha 700 terabyte di hard disk (450 per il sistema e 250 per gli utenti) e 1.6 petabyte di spazio su nastri. L'ES al momento della sua presentazione era cinque volte più veloce dell'IBM ASCI White il precedente più potente computer del pianeta.

Il sistema è in grado di effettuare simulazioni dell'intera atmosfera planetaria e degli oceani con una precisione di 10 chilometri.

L'ESC è stato accuratamente progettato per proteggere il computer dai disastri naturali. Dei file percorrono l'esterno dell'edificio per poter intercettare e scaricare a terra eventuali fulmini o scariche ad alta tensione. L'illuminazione è fornita da lampade alogene installate esternamente alla sala dei computer per prevenire eventuali interferenze magnetiche. L'edificio è costruito su speciali supporti isolanti che oscillando e comprimendosi proteggono l'edificio dalle scosse sismiche.

## Blue Gene

Blue Gene è il nome di un'architettura progettata per realizzare la nuova generazione di supercomputer a parallelismo massivo sviluppati per lavorare con potenze di calcolo che vanno dalle decine di Teraflops per arrivare fino al Petaflops. Il più potente computer BlueGene attualmente sviluppa 280 Teraflops. Il progetto è una cooperazione del United States Department of Energy (che parzialmente finanzia il progetto), industrie (IBM in particolare) e Università. Vi sono cinque progetti Blue Gene, tra i quali il Blue Gene/L, il Blue Gene/C e il Blue Gene/P.



## Blue Gene/L

Il primo sistema Blue Gene, il **Blue Gene/L** è stato sviluppato in collaborazione con il Lawrence Livermore National Laboratory (LLNL). Il sistema sviluppa una potenza di picco teorica di 360 Teraflops e secondo il test Linpack genera 280 Teraflops.

## Storia

Il 29 settembre 2004 IBM annunciò che il prototipo Blue Gene/L nei suoi laboratori a Rochester nel Minnesota aveva superato le prestazioni del precedente supercomputer più veloce del pianeta l'Earth Simulator. Il Blue Gene/L sviluppava 36.01 Teraflops rispetto ai 35.86 Teraflops dell'Earth Simulator. Il sistema era formato da 8 torri con 1024 nodi per ogni torre. Quando la configurazione venne raddoppiata il sistema raggiunse i 70.72 Teraflops.

Il 24 marzo 2005 il dipartimento statunitense per l'energia annunciò che il Blue Gene/L installato al LLNL aveva superato i precedenti record di velocità raggiungendo i 135.5 Teraflops con 32 torri.

Seconda la lista Top500 di giugno 2005, i sistemi basati su Blue Gene/L hanno conquistato 5 delle prime dieci posizioni e 16 delle prime 64 posizioni.

Il 27 ottobre 2005 LLNL e IBM annunciarono che Blue Gene/L aveva nuovamente superato il suo precedente record raggiungendo i 280.6 Teraflops con una configurazione formata da 65.536 nodi di calcolo distribuiti in 32 torri e con 1024 addizionali nodi dedicati alle comunicazioni tra le unità.

Blue Gene/L è stato il primo supercomputer a funzionare a più di 100 Teraflops con un'applicazione

reale. Nella fattispecie l'applicazione che si chiamava three-dimensional molecular dynamics code (ddcMD) nato per simulare la solidificazione (processi di crescita e fissazione) dei metalli fusi ad alte pressioni ed alte temperature. Il progetto ha vinto il Gordon Bell Prize del 2005.

## Architettura

Ogni nodo di calcolo o di comunicazione è un singolo ASIC con associata una memoria DRAM. L'ASIC integra al suo interno due PowerPC 440 a 700 Mhz costruito con una tecnologia a 130 nanometri. Ogni processore è dotato di due unità di calcolo in virgola mobile a doppia precisione e un sottoinsieme che gestisce la cache e la memoria DRAM oltre a gestire la comunicazione interprocesso. Le due unità FPU forniscono a ogni nodo Blue Gene una potenza teorica di 5.6 Gigaflops. I singoli nodi non implementano una gestione coerente della cache.

Integrando tutti i sottosistemi in un unico chip consente di abbattere la potenza dissipata infatti ogni chip consuma solo 17 Watt, tenendo conto anche della DRAM. Questo consente di assemblare moltissimi chip in uno spazio ridotto e quindi di inserire in una singola torre 1024 nodi di calcolo con gli allegati elementi di comunicazione mantenendo ragionevoli i consumi e le potenze da dissipare.

Se si analizza il sistema confrontando i Flops per Watt o in Flops per Watt per metro<sup>2</sup> si nota che il sistema Blue Gene è un progetto estremamente efficiente.

Ogni nodo Blue Gene è collegato a tre reti di comunicazioni parallele che generano una rete con topologia di un toroide tridimensionale per realizzare una veloce rete peer-to-peer tra i vari nodi e una rete collettiva per comunicazioni collettive. I nodi di comunicazione (I/O) utilizzano Linux come sistema operativo e comunicano con il mondo esterno tramite una rete ethernet. In fine una rete ethernet separata e privata provvede alle operazioni di configurazioni dei singoli nodi di diagnostica e di avvio del sistema.

I nodi Blue Gene utilizzano una versione minimalista di sistema operativo che supporta un singolo programma. Per poter utilizzare il Blue Gene/L con più programmi contemporaneamente il sistema può essere suddiviso in più sottosistemi isolati elettricamente dai vari nodi. Il numero di nodi che possono formare una partizione deve essere una potenza di 2. Per esempio 2<sup>5</sup> corrisponde a 32 nodi che possono formare una partizione. La massima partizione ovviamente comprende tutti i nodi di calcolo. Prima di eseguire un programma bisogna riservare una partizione, dopo di che si avvia il programma, i nodi coinvolti rimarranno inaccessibili agli altri nodi del sistema e alla fine del programma i nodi saranno liberati per poter essere nuovamente utilizzati.

Data la presenza di un tale numero di elementi qualche nodo potrebbe guastarsi e quindi ogni nodo può essere disconnesso elettricamente dal sistema per non compromettere l'affidabilità del sistema.

# Tassonomia di Flynn

Nel 1966 Michael J. Flynn classificò i sistemi di calcolo a seconda della molteplicità del flusso di istruzioni e del flusso dei dati che possono gestire; in seguito questa classificazione è stata estesa con una sottoclassificazione per considerare anche il tipo di architettura della memoria. In base a questa classificazione ogni sistema di calcolo rientra in una di queste categorie:

**SISD** (Single Instruction Single Data)

**SIMD** (Single Instruction Multiple Data)

Processori vettoriali

Array processor

Array sistolici

**MISD** (Multiple Instruction Single Data)

**MIMD** (Multiple Instruction Multiple Data)

Sistemi a memoria distribuita

**MPP** Massively Parallel Processing

**COW** Cluster Of Workstations

Sistemi a memoria condivisa

**UMA** Uniform Memory Access

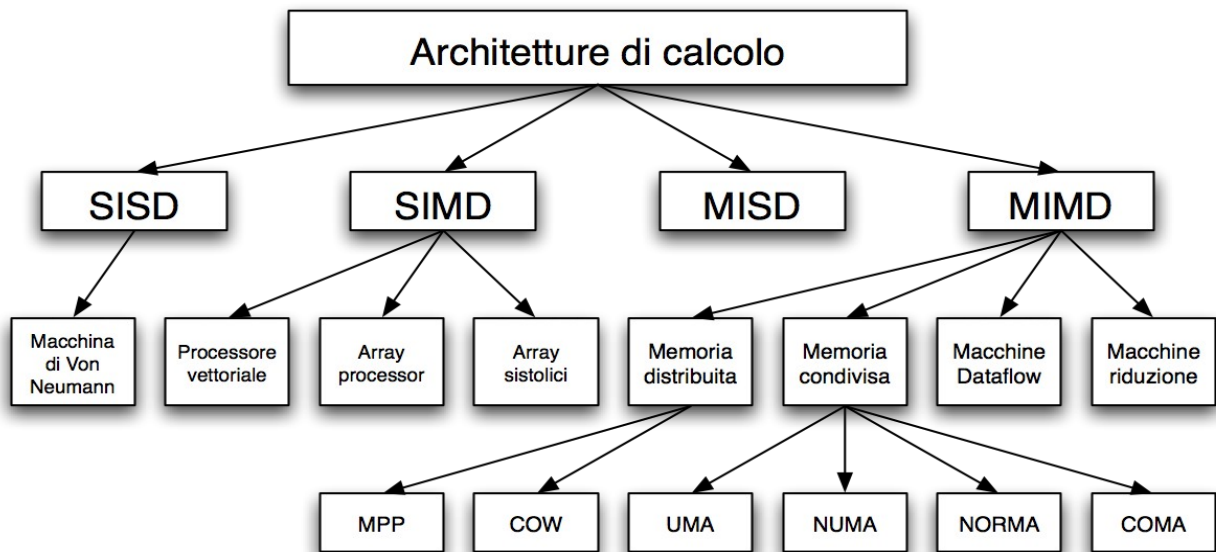
**NUMA** NonUniform Memory Access

**NORMA** NO remote Memory Access

**COMA** Cache only memory Access

Macchine dataflow

Macchine a riduzione



## **SISD**

Nessun parallelismo: le operazioni vengono eseguite sequenzialmente, su un dato alla volta. È la classica struttura della macchina di von Neumann, quasi tutti i computer in commercio sono basati su questa filosofia. I supercomputer hanno abbandonato quasi subito questa architettura per via delle ridotte prestazioni con elevati flussi di dati.

## **SIMD**

La stessa istruzione coinvolge più dati contemporaneamente.

## **Processori vettoriali**

Questo tipo di processori, oltre ai normali registri e istruzioni scalari, contiene degli speciali tipi di registri (registri vettoriali) che possono contenere N valori contemporaneamente, ed ogni operazione che coinvolga uno di questi registri viene eseguita su tutti i valori in esso memorizzati. Affinché questo meccanismo sia efficiente è necessario che il collegamento da e verso la memoria sia molto veloce, cioè abbia una banda passante molto elevata: in questo tipo di macchine anche la memoria è organizzata in modo vettoriale, vale a dire strutturata in modo che sia possibile leggere o scrivere esattamente N valori contemporaneamente. Inoltre in genere è possibile specificare un altro registro vettoriale come destinazione dell'operazione vettoriale corrente, dove il risultato verrà ulteriormente manipolato. Queste macchine sono programmabili con facilità (il parallelismo è gestito in maniera del tutto trasparente al programmatore), ma danno buone prestazioni solo nel caso di algoritmi con molte istruzioni vettoriali: sono particolarmente adatte per applicazioni di calcolo scientifico. Sono anche dette macchine a *parallelismo temporale*

## **Array processor**

Un array processor invece non ha affatto istruzioni scalari, ma solo vettoriali; è costituito da una unità di controllo (UC) che gestisce un array di processori (PE, Processor Element): i collegamenti fra PE e PE, e fra PE e memoria, sono di tipo matriciale, vale a dire che ogni PE comunica con i suoi quattro vicini, con la UC e con la memoria. La UC legge le istruzioni, se sono scalari le esegue lei stessa, se sono vettoriali le invia a ogni PE che si occupa di un singolo dato dell'array, in parallelo: quando tutti i PE hanno terminato la UC passa all'istruzione successiva. Per questo un array processor viene considerato una macchina a *parallelismo spaziale*.

Le prestazioni di un array processor sono ancora più legate al tipo di operazione: è molto veloce solo quando opera su array e vettori.

Una evoluzione dell'array processor è la Connection Machine, che al posto dei normali PE introduce delle *celle* costituite da un PE e una memoria locale, connesse con una topologia ipercubica.

## **Array sistolici**

Gli array sistolici sono delle architetture che elaborano un flusso di dati che si muove in modo prevedibile e ritmico lungo uno specifico percorso durante la sua elaborazione. Sono utilizzati

spesso nell'elaborazione dei segnali dato che i dati sono campionati con delle frequenze conosciute e devono subire delle elaborazioni predefinite che interessano tutti i dati. In questi array ogni elemento esegue una specifica elaborazione che dipende solamente dai dati di ingresso e dal suo stato interno. I dati elaborati sono posti in uscita dove un altro elemento provvederà a riceverli ed elaborarli. Le operazioni sono sincronizzate tramite un clock globale. Gli algoritmi eseguiti su questi array sono detti sistolici in analogia con il flusso sanguigno che provvede ad *impulsi* tramite percorsi predefiniti.

## **MISD**

Attualmente non esistono macchine MISD. Sono stati sviluppati alcuni progetti di ricerca ma non esistono processori commerciali che ricadono in questa categoria.

## **MIMD**

Più istruzioni vengono eseguite contemporaneamente su più dati diversi. Sotto questa classificazione ricadono i cluster di computer, che negli ultimi anni stanno avendo una notevole diffusione.

## **Sistemi a memoria distribuita**

In questa categoria ricadono le macchine che assegnano ad ogni nucleo di calcolo (nodo) una propria memoria riservata. Se un nodo deve accedere ai dati memorizzati in altro nodo deve farne richiesta attraverso uno scambio di messaggi tra i nodi o tecniche analoghe.

### ***Massively Parallel Processing***

Le macchine MPP sono composte da centinaia di processori (che possono diventare anche centinaia di migliaia in alcune macchine) collegati da una rete di comunicazione. Le macchine più veloci del pianeta sono basate su queste architetture.

### ***Cluster Of Workstations***

Le architetture COW sono sistemi di elaborazione basati su classici computer collegati da reti di comunicazione. I cluster di calcolo ricadono in questa classificazione.

## **Sistemi a memoria condivisa**

In questa categoria ricadono le macchine dove più unità di calcolo pur eseguendo programmi differenti accedono alla stessa memoria. I sistemi SMP ricadono in questa categoria.

### ***Uniform Memory Access***

Questi sistemi di elaborazione sono dotati di una memoria centralizzata che i processori utilizzano tramite un sistema a bus usualmente. La caratteristica fondamentale di questo sistema è il tempo di accesso alla memoria che è costante per ogni processore e per qualsiasi zona di memoria. Questo sistema è relativamente semplice da implementare ma non è molto scalabile. Questi sistemi al massimo gestiscono una dozzina di processori.

### **Non Uniform Memory Access**

Questi sistemi di elaborazione suddividono la memoria in una zona ad alta velocità assegnata

singolarmente ad ogni processore ed una eventuale zona comune per lo scambio dei dati. Questi sistemi sono molto scalabili ma più complessi da sviluppare.

### **Cache Only Memory Access**

Questa tipologia di elaboratori sono dotati solamente di memorie cache. Analizzando le architetture NUMA si è notato che queste mantenevano delle copie locali dei dati nelle cache e che questi dati erano memorizzati come doppioni anche nella memoria principale. Questa architettura elimina i doppioni mantenendo solo le memorie cache.

### **Macchine dataflow**

Le macchine a dataflow utilizzano un approccio innovativo nella programmazione. Il programma non è composto da una serie di istruzioni da eseguire sequenzialmente ma da un approccio data-driven. In sostanza le operazioni vengono eseguite solamente quando i dati per le elaborazioni sono disponibili. Non esiste un program counter che tiene traccia dello stato del programma.

### **Macchine a riduzione**

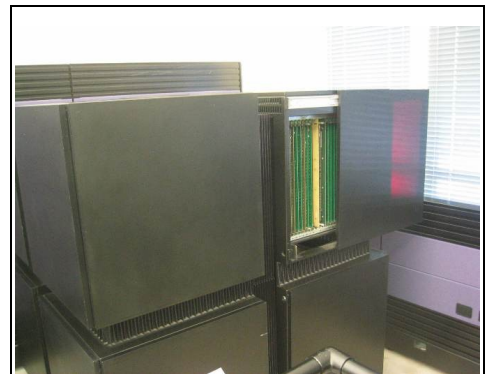
Le macchine a riduzione utilizzano un approccio simile a quello delle macchine dataflow ma utilizzano un punto di vista diverso. Invece di utilizzare un approccio data-driven utilizzano un approccio demand-driven. Questo approccio prevede che le computazioni vengano eseguite solamente quando vi è una richiesta dei risultati da elaborare. Anche in questa tipologia di macchina non vi è un program counter.

## Calcolo parallelo

Per incrementare in maniera significativa le prestazioni dei supercomputer fin dagli anni sessanta si decise di far ricorso a più unità di calcolo che lavorando in parallelo permettano di moltiplicare le prestazioni dei supercomputer. Nel corso degli anni vennero sviluppati varie tipologie di macchine parallele suddivise per il numero di processori, per topologia di collegamento e per tipologia di processore.

### Topologia

Per ottimizzare i tempi di accesso alla memoria si svilupparono topologie di collegamento che permettessero un accesso rapido alla memoria senza incrementare in maniera eccessiva i collegamenti. Le più diffuse topologie di collegamento sono quelle a matrice, toroide, cubo e varianti come il doppio toroide o l'ipercubo (un cubo a quattro dimensioni). L'aumento del numero di connessioni incrementa il numero di memorie collegate direttamente e quindi riduce i tempi di accesso ma nel contempo aumenta i costi di realizzazione e le difficoltà di gestione dato che una topologia più complessa implica un'algoritmo di accesso alla memoria più complesso.



*Computer basato su parallelismo massivo Thinking Machines CM-2*

### Processori

Alcune macchine parallele sono dotate di un numero limitato di processori generici che svolgono il compito di supervisionare a controllare un più elevato numero di processori semplici deputati allo svolgimento delle operazioni aritmetiche. Questo permette di contenere i costi dato che si possono utilizzare pochi processori costosi per gestire molti processori semplici ma pone problemi dal lato della programmazione dato che i programmi devono poter essere adeguatamente parallelizzati per poter funzionare su queste macchine particolari.

### Parallelismo massivo

Questa tipologia di supercomputer è dotata di migliaia o più processori che svolgono un programma in parallelo. Le più potenti macchine del pianeta sono di questo tipo. Ovviamente queste macchine offrono buone prestazioni solo con problemi che possono essere scomposti in migliaia di problemi più semplici e possibilmente indipendenti, dato che in queste macchine il tempo necessario ad un processore per contattare un altro processore può essere molto elevato.

### Calcolo distribuito

Questa tipologia di supercomputer è formata da un elevato numero di computer dalle prestazioni ridotte collegati tramite rete locale o rete telematica. Questi supercomputer funzionano con prestazioni adeguate solamente con problemi che possono essere scomposti in molti sottoproblemi indipendenti o quasi. Sottoproblemi indipendenti possono essere affrontati da computer collegati da reti lente come le reti telematiche mentre sottoproblemi legati in modo lasco devono essere affrontati da computer collegati in tempo reale per non introdurre un eccessivo collo di bottiglia nei tempi di comunicazione.



*Piccolo cluster di computer basato su software open source*

## **Grid**

Un'evoluzione del calcolo parallelo ha portato allo sviluppo dell'infrastruttura Grid. Il termine Grid computing (letteralmente, "calcolo a griglia") indica un'infrastruttura distribuita per consentire l'utilizzo di risorse di calcolo e di storage provenienti da un numero indistinto di calcolatori (anche e soprattutto di potenza non particolarmente elevata) interconnessi da una rete (solitamente, ma non necessariamente, Internet). Il termine "griglia" deriva dalla similitudine fatta dai primi ideatori del Grid Computing secondo i quali in un prossimo futuro si sarebbe arrivati ad utilizzare le risorse di calcolo alla stessa stregua dell'energia elettrica, ovvero semplicemente attaccando una spina all'infrastruttura energetica, in inglese Power grid. L'idea del Grid computing, di cui recentemente si sente spesso parlare come la prossima rivoluzione dell'informatica (come a suo tempo fu il World Wide Web), risale però a circa metà degli anni Novanta.

Le *griglie di calcolo* vengono prevalentemente utilizzate per risolvere problemi computazionali di larga scala in ambito scientifico e ingegneristico. Sviluppatesi originariamente in seno alla fisica delle alte energie (in inglese HEP), il loro impiego è già da oggi esteso alla biologia, all'astronomia e in maniera minore anche ad altri settori. I maggiori player dell'IT in ambito commerciale hanno già da tempo cominciato ad interessarsi al fenomeno, collaborando ai principali progetti grid world-wide con sponsorizzazioni o sviluppando propri progetti grid in vista di un utilizzo finalizzato al mondo del commercio e dell'impresa.

Una grid è in grado di fornire agli utenti di un gruppo scalabile senza una particolare caratterizzazione geografica (gergalmente detto VO ossia Virtual Organization) la potenzialità di accedere alla capacità di calcolo e di memoria di un sistema distribuito, garantendo un accesso coordinato e controllato alle risorse condivise e offrendo all'utente la visibilità di un unico sistema di calcolo logico cui sottomettere i propri job. L'idea del Grid computing è scaturita dalla constatazione che in media l'utilizzo delle risorse informatiche di una organizzazione è pari al 5% della sua reale potenzialità. Le risorse necessarie sarebbero messe a disposizione da varie entità in modo da creare un'organizzazione virtuale con a disposizione un'infrastruttura migliore di quella che la singola entità potrebbe sostenere.



## Hardware

I supercomputer per ottenere prestazioni sempre più elevate nel corso degli anni hanno esplorato molte strade e sviluppato architetture hardware diverse dalle solite architetture utilizzate nei computer. Queste architetture sono ottimizzate per eseguire alcune tipologie di operazioni mentre sono inefficienti per altre elaborazioni e quindi le architetture sono state sviluppate a seconda dello scopo alla quale era dedicata la macchina.

### Processori

Allo scopo di migliorare le prestazioni dei supercomputer sono stati sviluppati evoluzioni del classico processore ad architettura di von Neumann che migliorano le prestazioni nel caso di applicazioni scientifiche o che coinvolgono un elevato numero di dati. Alcune di queste scoperte tecnologiche sono in seguito state portate nei normali personal computer.

\*Pipeline

\*Processore vettoriale



*Il Cray X-MP il primo supercomputer vettoriale prodotto da Cray*

### Memoria

Con l'aumento delle prestazioni dei processori l'accesso della memoria è diventato un fattore limitante. Per evitare che le memorie diventassero il collo di bottiglia dei sistemi di calcolo vennero sviluppati sistemi di accessi che per incrementare la velocità delle memorie, i dati trasferiti e per ridurre i tempi di accesso.

\*Cache

\*Uniform Memory Access

\*Non-Uniform Memory Access

### Altro

L'evoluzione dei supercomputer costrinse gli ingegneri a risolvere anche problemi non direttamente legati all'elaborazione dei dati. I supercomputer col crescere delle dimensioni e della potenza diventavano sempre più voraci di potenza elettrica e venivano sempre più limitati dal tempo di propagazione degli elettroni nei circuiti. Questo spinse allo sviluppo di metodi di dissipazione innovativi e di nuove topologie che riducessero la lunghezza media dei percorsi tra la memoria e i processori. La mole crescente di dati trattati spinse i produttori hardware a sviluppare sistemi che garantissero flussi elevati e costanti di dati e elevate prestazioni, questi studi portarono alla realizzazione dei sistemi RAID.

•Raffreddamento a liquido

•RAID

## Pipeline

L'elaborazione di un'istruzione da parte di un processore si compone di cinque passaggi fondamentali:

**IF:** Lettura dell'istruzione da memoria

**ID:** Decodifica istruzione e lettura operandi da registri

**EX:** Esecuzione dell'istruzione

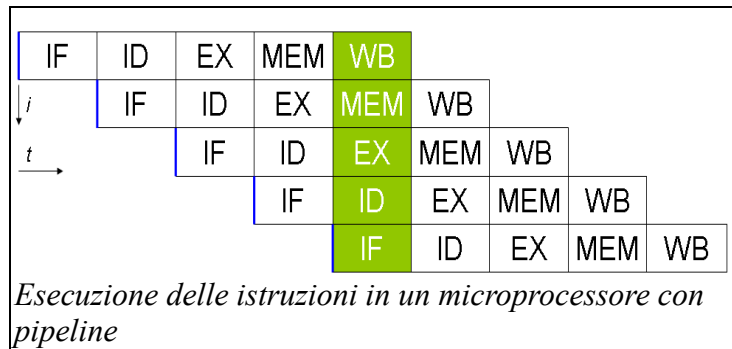
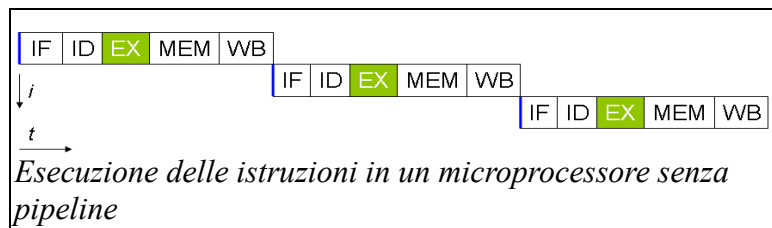
**MEM:** Attivazione della memoria (solo per certe istruzioni)

**WB:** Scrittura del risultato nel registro opportuno

Praticamente ogni CPU in commercio è gestita da un clock centrale e ogni operazione elementare richiede almeno un ciclo di clock per poter essere eseguita. Le prime CPU erano formate da un'unità polifunzionale che svolgeva in rigida sequenza tutti e cinque i passaggi legati all'elaborazione delle istruzioni. Una CPU classica richiede quindi almeno cinque cicli di clock per eseguire una singola istruzione.

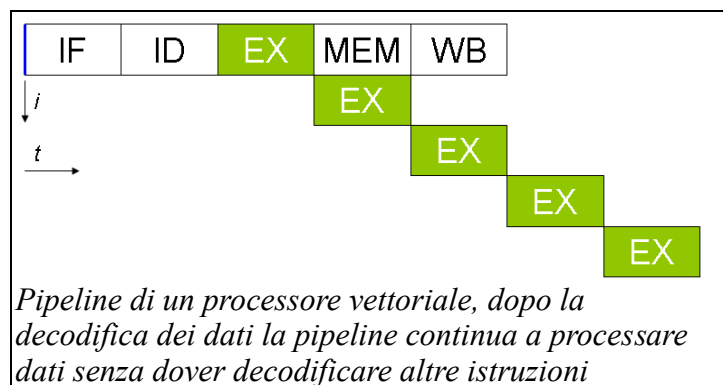
Con il progresso della tecnologia si è potuto integrare un numero maggiore di transistor in un microprocessore e quindi si sono potute parallelizzare alcune operazioni riducendo i tempi di esecuzione. La pipeline dati è la massima parallelizzazione del lavoro di un microprocessore.

Una CPU con pipeline è composta da cinque stadi specializzati, capaci di eseguire ciascuno una operazione elementare di quelle sopra descritte. La CPU lavora come in una catena di montaggio e quindi ogni stadio provvede a svolgere solo un compito specifico. Quando la catena è a regime, ad ogni ciclo di clock esce dall'ultimo stadio un'istruzione completata. Nello stesso istante ogni unità sta elaborando in parallelo i diversi stadi delle successive istruzioni. In sostanza si guadagna una maggior velocità di esecuzione a prezzo di una maggior complessità circuitale del microprocessore, che non deve essere più composto da una sola unità generica ma da cinque unità specializzate che devono collaborare tra loro.



## Processore vettoriale

Un processore vettoriale o array processor è una CPU progettata per svolgere operazioni matematiche su più dati elementari contemporaneamente. Questo in contrasto con l'architettura classica di un processore scalare che prevede l'elaborazione di un singolo dato per volta. La maggior parte dei processori sono scalari (o esternamente lo sembrano). I processori vettoriali sono comuni nelle applicazioni scientifiche e sono spesso alla



base dei supercomputer fin dagli anni 80. Con la fine degli anni 90 i microprocessori sono cresciuti di prestazioni e molti processori per applicazioni generiche si sono dotati di unità vettoriali o sono diventati vettoriali al loro interno. Nel 2000 IBM Toshiba e Sony hanno iniziato lo sviluppo del processore Cell, un microprocessore ad elevate prestazioni dotato di svariate unità vettoriali e rivolto ad applicazioni che spaziano dalle consolle al supercalcolo.

Attualmente praticamente ogni CPU moderna include istruzioni vettoriali tipicamente conosciute come istruzioni SIMD. Le consolle per i videogiochi e le schede grafiche fanno un ampio uso di processori vettoriali dato che l'elaborazione di flussi audio e video in tempo reale è un campo che ben si presta all'elaborazione vettoriale.

## Cache

I processori nel corso degli anni sono diventati sempre più veloci mentre le memorie per computer non hanno avuto un incremento paragonabile quindi mentre negli anni '70 le memorie erano significativamente più veloci dei processori dagli anni '90 in poi i microprocessori sono diventati molto più veloci delle memorie. Per impedire al processore di passare più del 90% del suo tempo ad attendere i dati dalla memoria si è deciso di dotare i processori di una ridotta quantità di memoria molto veloce dove conversare i dati utilizzati di frequente. Questa scelta deriva dalla constatazione che sebbene i programmi utilizzino molta memoria in realtà passano più del 90% del loro tempo ad accedere alle stesse locazioni di memoria.

Essendo le cache una copia della memoria la copia deve essere fedele ed aggiornata, disallineamenti tra ciò che si trova in cache e lo stato della memoria può portare ad errori di elaborazione. Nel caso si voglia realizzare una cache che contenga anche i dati da elaborare bisogna progettare accuratamente il processore in modo da impedire disallineamenti tra lo stato della memoria e lo stato della cache. In un sistema con più processori infatti esistono appositi protocolli che invalidano il contenuto della cache di un microprocessore nel caso la cache non sia aggiornata con lo stato nella memoria. Questi meccanismi aumentano la complessità circuitale e riducono le prestazioni dei processori ma sono indispensabili per ottenere un'elaborazione corretta dei programmi.

Le prestazioni della cache dipendono moltissimo dalla scelta di un'adeguata politica di riempimento della stessa. Nella cache vanno memorizzati i dati utilizzati più di frequente e quindi i microprocessori includono dei circuiti che provvedono a scegliere i dati che nel futuro verranno probabilmente utilizzati. Sebbene vi siano stati molti studi teorici sull'argomento allo stato attuale l'algoritmo più utilizzato provvede a eliminare dalla cache i dati utilizzati meno di recente con i nuovi dati da caricare.

## Uniform Memory Access

L'Uniform Memory Access (UMA) è stata la prima organizzazione di memoria sviluppata per i supercomputer paralleli. Ogni processore aveva un tempo di accesso costante alla memoria indipendentemente dal fatto che la memoria fosse fisicamente vicina al processore oppure no.

Questa architettura veniva ottenuta usualmente tramite un bus e delle unità di arbitraggio che mettevano in comunicazione la memoria con il processore. Questa tipologia di memoria è semplice da implementare e da programmare dato che i tempi di accesso sono prevedibili a priori. Lo svantaggio di questa tipologia di accesso alla memoria è che se la memoria non è molto più veloce dei processori si creano dei colli di bottiglia nell'accesso alla memoria.

Questa tipologia di memoria venne inizialmente utilizzata da sistemi con pochi processori ma venne rapidamente abbandonata dato che con l'aumento del numero di processori e con l'aumento della loro frequenza i tempi di accesso alla memoria penalizzavano eccessivamente i supercomputer. Questo portò allo sviluppo di sistemi come il sistema NUMA.

## Non-Uniform Memory Access

Non-Uniform Memory Access (NUMA) è un'architettura di memoria sviluppata per i sistemi multiprocessore dove i tempi di accesso dipendono dalla posizione della memoria rispetto al processore. Nelle architetture NUMA un processore può accedere rapidamente alla propria memoria locale, più lentamente alle memorie degli altri processori o alla memoria condivisa.

L'architettura NUMA è il logico passo successivo delle architetture SMP. Questa tipologia di architettura fu sviluppata inizialmente da Sequent Computer Systems e Data General durante gli anni 90. Queste tecnologie vennero in seguito adottate da molti sistemi Unix e parzialmente anche nell'architettura di Windows NT e dei successivi.

### **Concetti basilari**

Le moderne CPU sono molto più veloci delle memorie a cui accedono. All'inizio degli anni '70 quando iniziarono a diffondersi i primi supercomputer le memorie erano molto più veloci dei processori e quindi problemi di accesso alla memoria non si ponevano. Con il rapido incremento della frequenza di funzionamento delle CPU i tempi di accesso alla memoria divennero un problema sempre più impellente fino a diventare uno dei principali problemi delle attuali generazioni di processori. Molti sistemi sviluppati alla fine degli anni 80 e 90 si concentravano sull'ottenere sistemi di accesso alla memoria molto veloci piuttosto che unità di calcolo veloci dato che l'obiettivo dei sistemi era quello di poter trattare elevate quantità di dati.

La chiave per ottenere elevate prestazioni con i moderni supercomputer è quello di limitare l'accesso alla memoria a una limitata quantità di memoria molto veloce. Per realizzare questo vengono utilizzate cache di dimensioni sempre crescente all'interno dei processori che mantengono i dati utilizzati di frequente. Queste cache vengono aggiornate con algoritmi sempre più avanzati al fine di mantenere solo le informazioni utilizzate effettivamente di frequente. Questi metodi non sono in grado di compensare completamente il continuo aumento di memoria e di dimensione dei programmi e sistemi operativi che riducono le prestazioni delle cache. Nei sistemi multiprocessore questo diventa ancora più evidente dato che solo un processore può accedere alla memoria mentre gli altri processori se devono accedere alla memoria sono costretti ad attendere il loro turno.

L'architettura NUMA cerca di porre rimedio a questo problema fornendo a ogni processore una piccola zona di memoria ad accesso esclusivo e veloce in modo da evitare la creazione di colli di bottiglia. Nel caso di applicazioni che richiedono la condivisione di dati come nel caso di server e simili l'architettura NUMA migliora le prestazioni se si suddivide la memoria centrale in diversi

banchi e si assegna ad ogni banco un numero ridotto di processori.

Naturalmente i dati non sono realmente separati nelle memorie dei singoli processori e se dei dati devono essere elaborati da più processori questo è possibile. In questo caso l'architettura NUMA prevede che il software o dei dispositivi hardware provvedano a spostare i dati da un banco a un altro. Questa copia dei dati rallenta i processori e quindi l'efficienza delle architettura NUMA dipende molto dai compiti svolti dal sistema.

### **Corenza delle cache e NUMA (ccNUMA)**

Ora mai qualsiasi processore è dotato di una piccola zona di memoria molto veloce ad accesso esclusivo chiamata cache. Queste cache migliorano le prestazioni del sistema perché sfruttano il principio di località dei programmi ma la loro gestione introduce un notevole carico di lavoro aggiuntivo all'architettura NUMA.

Sebbene semplice da progettare un'architettura NUMA senza cache non viene normalmente implementata date le scarse prestazioni del sistema. Un architettura NUMA che non mantenga le cache coerenti sarebbe anch'essa semplice da progettare ma la sua programmazione sarebbe praticamente impossibile. Quindi tutte le architetture NUMA hanno al loro interno dei meccanismi hardware che provvedono a gestire la coerenza delle cache, queste architetture sono chiamate ccNUMA.

Questi sistemi normalmente utilizzano comunicazioni interprocesso tra i gestori delle cache che tengono traccia del fatto che più processori abbiano copie della stessa zona di memoria e questi provengono a invalidare le cache se uno dei processori modifica una di queste zone di memoria controllate. Per questa ragione le architetture NUMA forniscono delle prestazioni scarse se i processori accedono alle stesse zone di memoria in rapida successione. I sistemi operativi che supportano le architetture NUMA cercano di ridurre il problema allocando i processi in modo da sfruttare la divisione dell'architettura e fornendo algoritmi di gestione e di bloccaggio della memoria in modo da ridurre gli accessi contemporanei alla stessa zona di memoria.

### **NUMA e cluster**

L'architettura NUMA può essere vista come una versione ridotta di cluster di computer. L'aggiunta della paginazione della memoria virtuale a un sistema a cluster permette di emulare un'architettura NUMA anche se questa non esiste in hardware. Tuttavia questa soluzione è molto più lenta infatti la comunicazione interprocesso in un'architettura NUMA software è diversi ordini di grandezza più lenta di una comunicazione interprocesso in un'architettura NUMA hardware.

## Raffreddamento a liquido

L'incremento di frequenza dei processori utilizzati nei supercomputer rese le macchine molto voraci dal punto di vista energetico. Questo rese necessario anche sviluppare innovativi metodi di raffreddamento per poter dissipare i MegaWatt consumati dai supercomputer. I Supercomputer inoltre non potevano essere eccessivamente voluminosi sia per permetterne una semplice installazione sia per non penalizzare la prestazioni con collegamenti elettrici troppo lunghi. Diventando i supercomputer *compatti* non si poteva più far affidamento sul raffreddamento a convezione naturale o a convezione forzata (tramite ventole e dissipatori) ma bisognava sviluppare nuove tecnologie in grado di asportare il calore in eccesso. La Cray Inc. per il suo primo computer il Cray-1 realizzò un sistema di raffreddamento a liquidi (vedi in figura) in modo da asportare il calore dai processori (115 KiloWatt di potenza) e convogliarlo verso ampi radiatori che avrebbero provveduto a disperdere il calore. Negli anni successivi molte ditte imitarono la Cray e alla fine degli anni 90 diverse ditte immisero sul mercato soluzioni di raffreddamento a liquidi per i personal computer dato che gli ultimi processori per uso domestico consumavano talmente potenza da richiedere particolari sistemi di raffreddamento per operare correttamente.



## RAID

Un Redundant array of independent disks (Insieme ridondante di dischi indipendenti), conosciuto più comunemente con il nome di RAID, è un sistema che usa un insieme di dischi rigidi per condividere o replicare le informazioni. I benefici del RAID sono di aumentare l'integrità dei dati, la tolleranza ai guasti e/o le prestazioni, rispetto all'uso di un disco singolo. Nella sua implementazione originaria (nella quale l'acronimo era l'abbreviazione di "Redundant Array of Inexpensive Disks", cioè "Insieme ridondante di dischi economici"), il fattore chiave era l'abilità di combinare parecchi dischi a basso costo e obsoleti in modo da rendere il sistema nel suo complesso migliore di un disco di ultima generazione per capacità, affidabilità e/o velocità.

Nel suo livello più semplice, il sistema RAID permette di combinare un insieme di dischi in una sola unità logica. In questo modo il sistema operativo, invece di vedere differenti dischi, ne vede solamente uno. Il RAID è tipicamente usato nei server, e di solito è implementato con dischi di identica capacità. Con il calo del costo dei dischi rigidi e con il diffondersi della tecnologia RAID nei chipset delle schede madri, il RAID è spesso offerto come opzione sia sui computer di fascia alta sia su quelli usati da utenti domestici, specialmente se dedicati a compiti che richiedono un grande immagazzinamento di dati, come il montaggio audio/video.

Le specifiche originali suggerivano un diverso numero di "livelli di RAID", o combinazioni di dischi. Ogni combinazione aveva dei vantaggi e degli svantaggi. Con il passare degli anni, sono nate diverse implementazioni del concetto di RAID. La maggior parte differiscono sostanzialmente nell'implementazione dei livelli RAID ideati inizialmente. Questo può portare spesso a confusione, poiché un'implementazione RAID-5 può essere molto diversa da un'altra. RAID-3 e RAID-4 sono spesso confusi o scambiati tra loro.

La vera definizione di RAID è stata oggetto di dibattito nel corso degli anni. L'uso del termine

ridondante porta a molte discussioni se il RAID-0 sia "vero" RAID. Analogamente, il cambio da economico a indipendente confonde molti dei concetti alla base del RAID. Infatti ci sono alcune implementazioni del RAID che usano un solo disco! Per gli scopi di questo articolo, noi indichiamo come RAID ogni sistema che sviluppa il concetto base di RAID di ricombinare lo spazio fisico di dischi diversi per lo scopo di aumentare l'affidabilità o le prestazioni del sistema nel suo complesso.

## **Software**

Essendo i supercomputer dotati di hardware peculiare ed essendo indirizzati verso lo svolgimento di alcune tipologie specifiche di compiti gli usuali programmi per computer non sono adatti per queste macchine. Le particolari soluzioni architetturelle sviluppate nei supercomputer per essere sfruttate a pieno richiedono l'utilizzo di appositi linguaggi e di opportune primitive software. È da notare che le prestazioni dei programmi eseguiti in parallelo dipendono pesantemente dalla porzione di codice effettivamente parallelizzabile. Una quantità eccessiva di codice non parallelizzabile penalizza pesantemente le prestazioni come dimostra la legge di Amdahl. Il software per il supercalcolo si può dividere in tre macrocategorie.

### ***Linguaggi di programmazione***

Tutti i supercomputer moderni sono basati su un parallelismo massivo e quindi i normali linguaggi di programmazione sviluppati per le macchine sequenziali come i comuni personal computer non sono sufficienti. Questi linguaggi a volte sono classici linguaggi di programmazione con in più alcune librerie e alcuni costruttori appositamente sviluppati per il supercalcolo. Per esempio il linguaggio C è disponibile praticamente per ogni supercomputer. A volte i linguaggi sono sviluppati appositamente per i supercomputer, il più famoso di questi linguaggi è l'Occam. Sebbene esistano diversi linguaggi di programmazione paralleli nessuno di questi permette una programmazione rapida e semplice dei supercomputer. Infatti una delle aree di ricerca più attiva del supercalcolo mira a realizzare linguaggi di programmazione semplici da utilizzare ma che siano di utilizzare in maniera nativa il calcolo parallelo.

### ***Primitive di sistema***

Essendo tutti i supercomputer moderni basati su macchine massivamente parallele si rende necessario implementare metodi efficienti di comunicazione tra i processi paralleli e procedure in grado di bilanciare il carico tra i vari processori evitando di sovraccaricare alcuni processori lasciando scarichi altri processori. Queste primitive possono essere implementate tramite librerie esterne (MPI, PVM), o tramite chiamate innestate nel sistema operativo. Quasi tutti i sistemi Unix commerciali implementano primitive di questo tipo (sebbene alcune implementazioni siano deficitarie). Esistono implementazioni open source come il progetto openMosix che modificano il kernel del sistema operativo Linux per fargli supportare il calcolo parallelo. Anche alcuni sistemi operativi per personal computer come il Mac OS X implementano primitive per il supercalcolo tramite il progetto Xgrid. Il futuro Windows Vista dovrebbe averne un supporto parziale sebbene al momento non è chiaro quanto siano sviluppate e solide le primitive.

### ***Programmi***

I programmi per supercomputer vanno appositamente progettati cercando di rendere gli algoritmi di calcolo il più paralleli possibile. La parallelizzabilità di un algoritmo dipende strettamente dal problema trattato. Un problema facilmente parallelizzabile è un problema che può essere scomposto in tanti sottoproblemi più semplici che non siano collegati o che siano collegati in modo lasco. Difatti in un supercomputer uno dei problemi maggiori è fornire una comunicazione efficiente tra le varie unità difatti il tempo di accesso ad un dato che non si trovi nella memoria locale del processore può essere molto elevato e comunque un numero elevato di comunicazioni tra processori può saturare la banda di comunicazione peggiorando drammaticamente le prestazioni. Quindi è imperativo realizzare programmi che siano il più indipendenti possibile. Un esempio classico di problema facilmente parallelizzabile è la realizzazione di animazioni fotorealistiche con il computer. Una volta impostata l'animazione un computer principale trasmette ai nodi di calcolo i fotogrammi da elaborare. Durante l'elaborazione i nodi non devono comunicare tra loro e la



comunicazione con il nodo principale si ha solamente durante la fase di caricamento dei dati e di salvataggio dei fotogrammi. Nel campo scientifico i calcoli di fluidodinamica o di previsione del tempo sono ben parallelizzabili. Invece programmi che per esempio richiedono frequenti accessi ad ampie matrici sparse sono difficilmente parallelizzabili dato che ogni nodo cercherà di accedere alla memoria comune con elevata frequenza saturando rapidamente i canali di comunicazione.

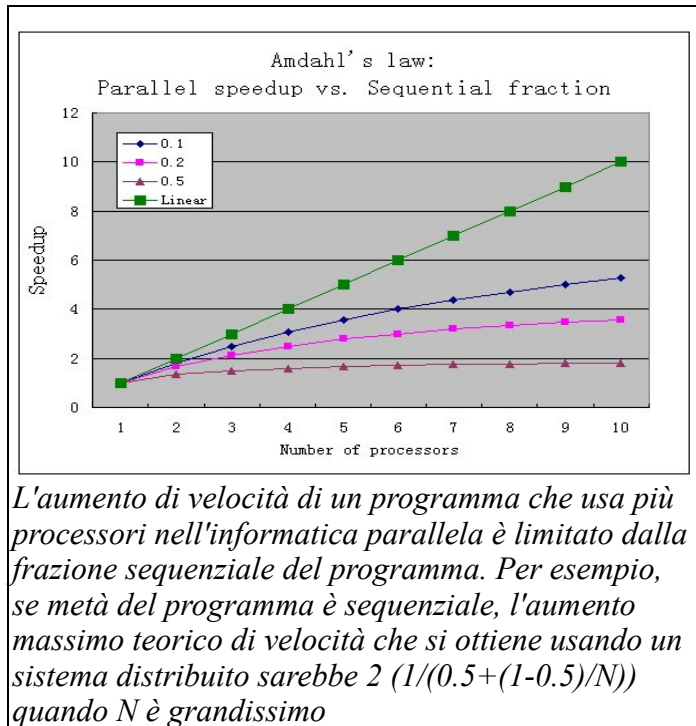
## Legge di Amdahl

La legge di Amdahl, che ha preso il nome del progettista di computer Gene Amdahl, viene usata per trovare il miglioramento atteso massimo in un sistema informatico quando vengono migliorate solo alcune parti del sistema. Viene usata spesso nell'informatica parallela per predire l'aumento massimo teorico di velocità che si ottiene usando più processori.

La legge di Amdahl può essere interpretata in modo più tecnico, ma in parole povere significa che è l'algori

tmo a decidere l'aumento di velocità, non il numero di processori. Prima o poi si raggiunge un punto in cui non si può parallelizzare ulteriormente l'algoritmo.

La legge di Amdahl è una dimostrazione della legge dei rendimenti calanti: anche se si potesse aumentare la velocità di una parte di un computer di cento o più volte, se tale parte influisce solamente sul 12% dell'elaborazione complessiva, al massimo l'accelerazione può essere di un fattore:  $1/(1-0.12)=1.136$



Più tecnicamente, la legge riguarda l'aumento di velocità ottenibile da un miglioramento a un calcolo che influisce per una proporzione P di quel calcolo, dove il miglioramento riduce il tempo di calcolo di un fattore S. (Per esempio, se un miglioramento può velocizzare il 30% del calcolo, P sarà 0.3; se il miglioramento raddoppia la velocità della porzione modificata, S sarà 2.) La legge di Amdahl afferma che l'aumento di velocità complessivo prodotto dal miglioramento sarà:

$$1/((1-P)+P/S)$$

Quindi al crescere del numero di processori che lavorano in parallelo l'incremento di prestazioni diventa sempre minore dato che la componente non parallelizzabile dei programmi diventa sempre più significativa nei tempi di calcolo totali.

## Occam

Occam è un linguaggio di programmazione per architetture parallele sviluppato dalla INMOS per la programmazione di reti di INMOS Transputer e successivamente implementato anche per altre piattaforme. Il nome si riferisce al principio del rasoio di Occam: nella progettazione del linguaggio, la Inmos si è limitata a un insieme minimo di costrutti, evitando di introdurre ciò che non pareva "strettamente necessario" per gli scopi del linguaggio. In effetti, Occam si può anche descrivere come un linguaggio di programmazione relativamente a basso livello, quasi una sorta di assembly. Il modello a cui il linguaggio si ispira sono i Communicating Sequential Processes (CSP) descritti da C.A.R. Hoare nel celebre libro omonimo.

### *Panoramica del linguaggio*

#### Canali

La comunicazione fra processi avviene attraverso canali dotati di nome. Un processo invia dati su un canale attraverso l'operatore ! e legge da un canale con ?. Questi operatori hanno una semantica basata sul rendezvous, ovvero un processo che tenta di inviare o ricevere dati su un canale rimarrà sospeso finché un altro processo non diventa disponibile a eseguire l'operazione complementare (cioè a ricevere o inviare) su quello stesso canale. Sintatticamente, output e input assumono la seguente forma (c è una variabile):

```
canale1 ? c
canale2 ! C
```

Si noti che in tutti gli esempi di codice riportati in questo articolo (a differenza di quanto ci si aspetterebbe da un linguaggio ad alto livello) l'indentazione e l'uso di spazi bianchi è significativo.

#### Strutture di controllo

Le strutture di controllo di Occam sono ispirate a quelle della programmazione strutturata, ma reinterpretate ed estese nel contesto della programmazione parallela. La struttura di controllo **SEQ** introduce una lista di istruzioni che vengono eseguite in sequenza (nella maggior parte dei linguaggi strutturati, questa modalità di esecuzione è implicita):

```
SEQ
  x := x + 1
  y := x * x
```

Come si può facilmente immaginare, := è l'operatore di assegnamento.

La struttura di controllo **PAR** specifica invece che le successive istruzioni devono essere eseguite in parallelo (contemporaneamente)

```
PAR
  x := x + 1
  y := y * 2
```

Il PAR ammette anche una forma più complessa che presenta alcune analogie con un ciclo for, e viene coerentemente indicata con le parole chiave PAR-FOR. Il seguente frammento di codice acquisisce un dato intero da quattro canali in parallelo.

```
PAR i=0 FOR 4
  INT n
  c[i] ? n[i]
```

L'analogia con il "ciclo for" riguarda l'uso del "contatore" i. Come un ciclo for tradizionale, il frammento di codice riportato esegue le operazioni indicate cinque volte, "per i che va da 0 a 4"; tuttavia, le cinque operazioni di input non sono svolte sequenzialmente, bensì in parallelo.

La struttura di controllo **ALT** realizza il concetto di **comando con guardia** dei CSP. Una

guardia è una combinazione di una condizione booleana e una istruzione di input/output (entrambi gli elementi sono opzionali). Una guardia "ha successo" se la sua condizione booleana è vera e la sua istruzione di input è "pronta" (ovvero c'è già un altro processo disponibile al *rendezvous*, cioè a inviare il dato richiesto o ricevere il dato inviato). A ciascuna guardia è associata una istruzione che può essere eseguita se la guardia ha successo:

```
ALT
  count1 < 100 & c1 ? data
  SEQ
    count1 := count1 + 1
    merged ! data
  count2 < 100 & c2 ? data
  SEQ
    count2 := count2 + 1
    merged ! data
  status ? request
  SEQ
    out ! count1
    out ! count2
```

Quando l'istruzione composta **ALT** viene raggiunta dal flusso del programma, le guardie vengono valutate. Se più di una ha successo, una di esse viene scelta (arbitrariamente) e la corrispondente istruzione viene eseguita.

## **Occam 2**

Occam 2 è una estensione di Occam realizzata da INMOS nel 1987. In pratica, Occam 2 si può considerare come la prima versione "utilizzabile" del linguaggio, mentre la prima versione aveva più la connotazione di prototipo. Fra le principali aggiunte di Occam 2 ci sono un sistema di tipi più ricco (con numeri in virgola mobile, caratteri, numeri interi di diverse dimensioni) e le subroutine ("funzioni").

### **Occam 2.1**

Occam 2.1 è la seconda estensione del linguaggio, sviluppata da INMOS nel 1995. A dispetto del nome (che secondo le convenzioni normali indicherebbe modifiche "minori"), Occam 2.1 introduce numerose e rilevanti novità, fra cui:

- \*possibilità di dare nomi definiti dal programmatore ai tipi;
- \*record
- \*Nuovi operatori (p. es. BYTESIN)
- \*Possibilità di definire array di canali (questa caratteristica è stata usata negli esempi riportati sopra)
- \*possibilità di includere gli array come tipo tornato delle funzioni

Per una lista completa delle caratteristiche di Occam 2.1, che a oggi è la versione "definitiva" del linguaggio si rimanda al manuale di riferimento.

## **MPI**

Il Message Passing Interface (MPI) è un protocollo di comunicazione per computer. È de facto lo standard per la comunicazione tra nodi appartenenti a un cluster di computer che eseguono un programma parallelo sviluppato per sistemi a memoria distribuita. Le MPI sono delle librerie che vengono richiamate dai programmi. MPI rispetto alle precedenti librerie utilizzate per il passaggio di parametri tra nodi ha il vantaggio di essere molto portatile (MPI è stata implementata per moltissime architetture parallele) e veloce (MPI viene ottimizzato per ogni architettura).

## **PVM**

La Parallel Virtual Machine (PVM) è un insieme di programmi sviluppati per la gestione parallela di computer in rete. Viene utilizzato per gestire un insieme eterogeneo di computer in rete che vengono utilizzati come fossero un unico processore parallelo.

PVM fu sviluppato dall'Oak Ridge National Laboratory dell'University of Tennessee e dall'Emory University. La prima versione fu scritta dall'ORNL nel 1989 ma in seguito venne totalmente riscritta e la versione 2 venne presentata nel marzo del 1991. La versione 3 venne realizzata nel marzo del 1993 e supportava il fault tolerance e era dotata di una buona portabilità.

Sebbene il PVM non sia più supportato viene considerato una importante evoluzione del calcolo distribuito e del grid computing.

PVM è distribuito per sistemi operativi Windows e per molti sistemi operativi Unix e permette di realizzare con relativa semplicità un singolo computer parallelo. Molti problemi complessi possono essere risolti in modo economico ed efficiente da un aggregato di computer che condividono memoria e potenza di calcolo. Il software è molto portatile e il sorgente è disponibile liberamente infatti è stato compilato per praticamente ogni tipo di computer partendo dai portatili per arrivare ai supercomputer Cray.

PVM consente agli utenti di utilizzare i computer correnti per analizzare e risolvere problemi molto complessi con un aggravio dei costi minimo. Moltissimi laboratori sparsi nel mondo utilizzano PVM per analizzare problemi scientifici, industriali, medici e in aggiunta PVM viene utilizzato per mostrare le basi del calcolo parallelo. Con decine di migliaia di utilizzatori PVM è considerato di fatto lo standard per il calcolo parallelo nel mondo.

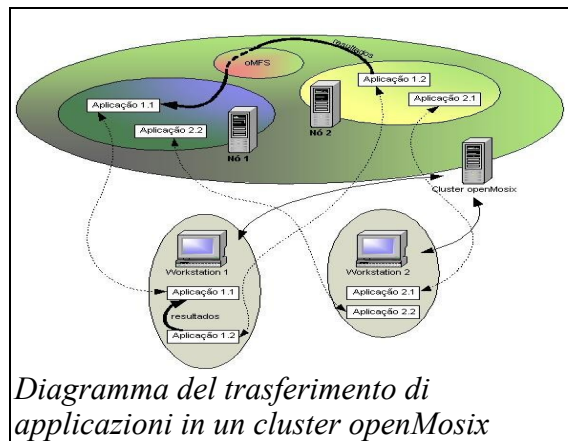
## openMosix

openMosix è un gestore di cluster libero del tipo single-system image che gestisce i singoli nodi e le comunicazioni con il gestore centrale. Il gestore è in grado di gestire la migrazione dei processi (non dei thread) da un nodo ad un altro in modo da bilanciare il carico. openMosix funziona molto bene nel caso di programmi paralleli che effettuano intense operazioni di I/O. Il programma comprende una patch al kernel Linux per abilitare la migrazione dei processi e molte distribuzioni specializzate lo utilizzano.

openMosix attualmente viene considerato stabile per le architetture X86. Il sistema supporta anche i processori con estensioni AMD64.

Attualmente il sistema può utilizzare il kernel Linux versione 2.4 o 2.6 e diversi programmi sono in sviluppo per il kernel 2.6.

Il progetto venne iniziato da Moshe Bar il 10 febbraio 2002 per fornire una alternativa libera al gestore commerciale MOSIX



*Diagramma del trasferimento di applicazioni in un cluster openMosix*

## Misura delle prestazioni

Misurare correttamente le prestazioni nei supercomputer è fin dall'inizio stato fondamentale per permettere di indirizzare gli acquirenti verso la tipologia di supercomputer più adatta a loro e per permettere agli ingegneri di individuare quali architetture sono le migliori da implementare.

La misura delle prestazioni di un sistema di elaborazione è un compito difficile da svolgere in modo affidabile dato che a seconda del problema da risolvere le prestazioni delle varie architettura possono variare in maniera significativa.

Negli anni 50 e 60 per valutare le prestazioni dei calcolatori si misuravano i FLOPS cioè i Floating Point Operations Per Second (operazioni in virgola mobile per secondo). Le operazioni in virgola mobile vengono utilizzate moltissimo nei calcoli scientifici e quindi erano considerati il parametro fondamentale da misurare. Negli anni ci si rese conto che questa misura non era affidabile dato che non teneva conto della tipologia di operazione in virgola mobile né tantomeno del carico complessivo. I FLOPS non sono in grado di individuare se il sistema è dotato di colli di bottiglia come la presenza di un sottosistema di memoria lento o la presenza di altri vincoli come unità di calcolo poco efficienti in caso di salti condizionati.

Per ottenere misure più affidabili si iniziarono ad effettuare test reali cioè a misurare il tempo necessario ai programmi di uso comune per risolvere problemi standardizzati. Tra i vari test che furono proposti emerse il test LINPACK che effettua calcolo numeri di algebra lineare. Questo test viene utilizzato per esempio dal sito TOP500 per stilare la classifica dei 500 più potenti computer del pianeta.

Nel 1998 venne fondata la Standard Performance Evaluation Corporation (SPEC) una organizzazione no-profit il cui scopo è la realizzazione di test che permettano di confrontare in modo oggettivo e indipendente le reali prestazioni dei vari elaboratori. Questa organizzazione sviluppa i test a seconda dell'impiego e quindi esistono test per misurare le prestazioni nel capo della grafica, per misurare le prestazioni nel campo dei siti web e così via. I test SPEC per il supercalcolo vengono considerati molto affidabili e insieme ai test LINPACK permettono di realizzare confronti molto affidabili tra le varie macchine.

## Utilizzi

I supercomputer possono essere utilizzati in tutti i settori che richiedono l'elaborazione di elevate quantità di dati ma il loro costo ne limita l'utilizzo a grandi aziende pubbliche o private o a enti come ministeri o università.

Attualmente il maggior acquirente mondiale di supercomputer è l'National Security Agency. L'agenzia di sicurezza americana utilizza questi computer per compiti di crittografia e di analisi del traffico telematico. Le agenzie governative infatti furono tra le prime ad utilizzare i supercomputer per svolgere compiti balistici di simulazione dei vari scenari strategici e successivamente per la gestione integrata dei sistemi difensivi.

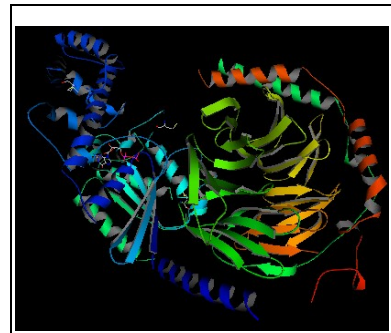
Quando i supercomputer divennero più accessibili vennero utilizzati anche da grandi aziende per la gestione contabile, analisi dei dati e dei magazzini. Le università invece utilizzarono i calcolatori per eseguire pesanti simulazioni numeriche in settori come la fluidodinamica, la meteorologia e nei settori delle alte energie.

Durante gli anni 60 la NASA fu un importante acquirente dei supercomputer dovendo studiare i migliori profili delle navicelle spaziali, ma col trascorrere degli anni l'importanza della NASA in questo settore decrebbe dato che i minicomputer prima e i personal computer dopo poterono assolvere alle richieste della NASA.

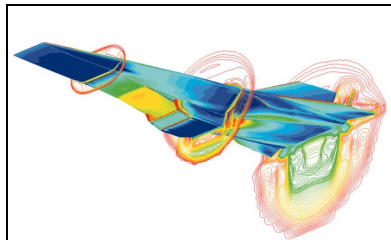
Le aziende private utilizzano i supercomputer per simulazioni numeriche di fluidodinamica (Airbus, Boeing, ecc) analisi delle prospezioni petrolifere (tutte le compagnie petrolifere) simulazioni di biologia molecolare (compagnie farmaceutiche) simulazioni di crash-test (compagnie automobilistiche), ecc.

Negli ultimi anni il settore dei supercomputer ha ricevuto notevoli finanziamenti dal dipartimento americano della difesa e dell'energia per sviluppare simulazioni accurate delle esplosioni nucleari dopo che nel 1993 il presidente statunitense ebbe firmato un trattato di messa al bando delle esplosioni nucleari nel sottosuolo. Le simulazioni servono oltre che a sviluppare armi più efficaci e sicure a sviluppare armi che richiedano poca manutenzione e che possano essere stoccate per molti anni senza perdita apprezzabile delle prestazioni.

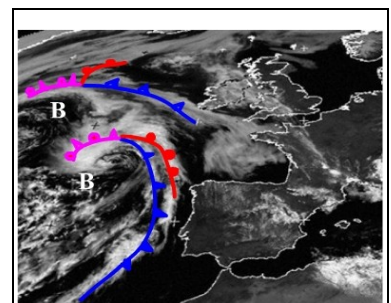
La meteorologia e le simulazioni sull'inquinamento futuro sono stati un altro settore che negli ultimi anni ha assorbito elevata potenza di calcolo dato che solo modelli climatici accurati permettono di far previsioni sensate sull'evoluzione dell'ambiente in funzione degli attuali livelli di inquinamento. Questi modelli dovrebbero permettere l'individuazione anche delle contromisure più appropriate per preservare l'ambiente.



*La bioinformatica è una delle nuove aree di ricerca a più elevato utilizzo di supercomputer*



Dryden Flight Research Center ED97 43968-01  
HYPER-X AT MACH 7: This computational fluid dynamic (CFD) image is of the Hyper-X vehicle at the Mach 7 test condition with the engine operating.  
*Analisi fluidodinamica di un prototipo NASA*



*Previsione meteorologica*



## Bibliografie

Il libro si è basato sulle seguenti voci dell'enciclopedia online [it.wikipedia.org](http://it.wikipedia.org) con varie aggiunte e integrazioni. Anche le schede dei vari supercomputer inseriti del capito Storia derivano dalla suddetta enciclopedia.

- Supercomputer
- Pipeline dati
- Processore vettoriale
- Non-Uniform Memory Access
- Cache
- Calcolo parallelo
- RAID
- Legge di Amdahl
- MPI
- PVM
- openMosix
- Linguaggio di programmazione Occam.

# Licenza

## GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## **2. VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## **3. COPYING IN QUANTITY**

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### **4. MODIFICATIONS**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D.** Preserve all the copyright notices of the Document.
- E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H.** Include an unaltered copy of this License.
- I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your

Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **5. COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts. A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.