

STAGE RESIDENZIALE LNF

2013

Gruppo di
informatica



**STAGES LNF
2013**

**ESTIVI e RESIDENZIALI
XIV Edizione**

Studenti IV anno Scuole Secondarie di secondo grado
Lezioni e Gruppi di Lavoro

Laboratori Nazionali di Frascati
10 - 14 giugno e 17 - 21 giugno 2013



Claudio Federici "Migrazione" 2013

SIDS - Ufficio Comunicazione ed Educazione Scientifica

Contatti: comedu@lnf.infn.it
069403 2423 - 2871 - 2868



INFN

Istituto Nazionale
di Fisica Nucleare

CHE COS'È UN CENTRO DI CALCOLO?



Un centro di calcolo è una struttura di sviluppo, utilizzata per analizzare e memorizzare dati relativi ad attività di ricerca svolte all'interno di un dipartimento matematico-scientifico.



COM'È STRUTTURATO UN CENTRO DI CALCOLO?

- CPU
- Memoria centrale
- Memorie di massa
- Rete



COS'È LA CPU?

La CPU (Central Processing Unit) è una parte fondamentale poiché si occupa dell'elaborazione, gestione ed acquisizione dei dati. Essa deve essere proporzionale alla mole di dati presa in carico dal centro di carico.

Non è da sottovalutare l'aspetto economico.



CHE COS'È LA MEMORIA CENTRALE?

La memoria centrale è una parte essenziale per qualunque tipo di calcolatore e le dimensioni devono essere appropriate al tipo di utilizzo.

In particolare nei centri di calcolo è importante tenere in considerazione che nonostante la maggior disponibilità di memoria aumenta le prestazioni del centro, non è sempre possibile avere a disposizione molta memoria per motivi economici.



CHE COS'È LA MEMORIA DI MASSA?

È la parte di un calcolatore che ci permette di conservare i dati nel tempo.



Dischi Veloci

- Costi elevati
- Elevata velocità

Dischi Medi

- Costi medi
- Velocità media

Nastri

- Costi bassi
- Molto lenti

GESTIONE DEI DATI

Per una corretta memorizzazione è necessario gestire i dati che si desidera conservare, queste tecniche prendono il nome di HSM (High Storage Manager).

Vi sono diversi tipi di File System, ma prenderemo in esame solo quello presente all'interno del sistema operativo UNIX, utilizzato dalle macchine del centro di calcolo.



FILE SYSTEM

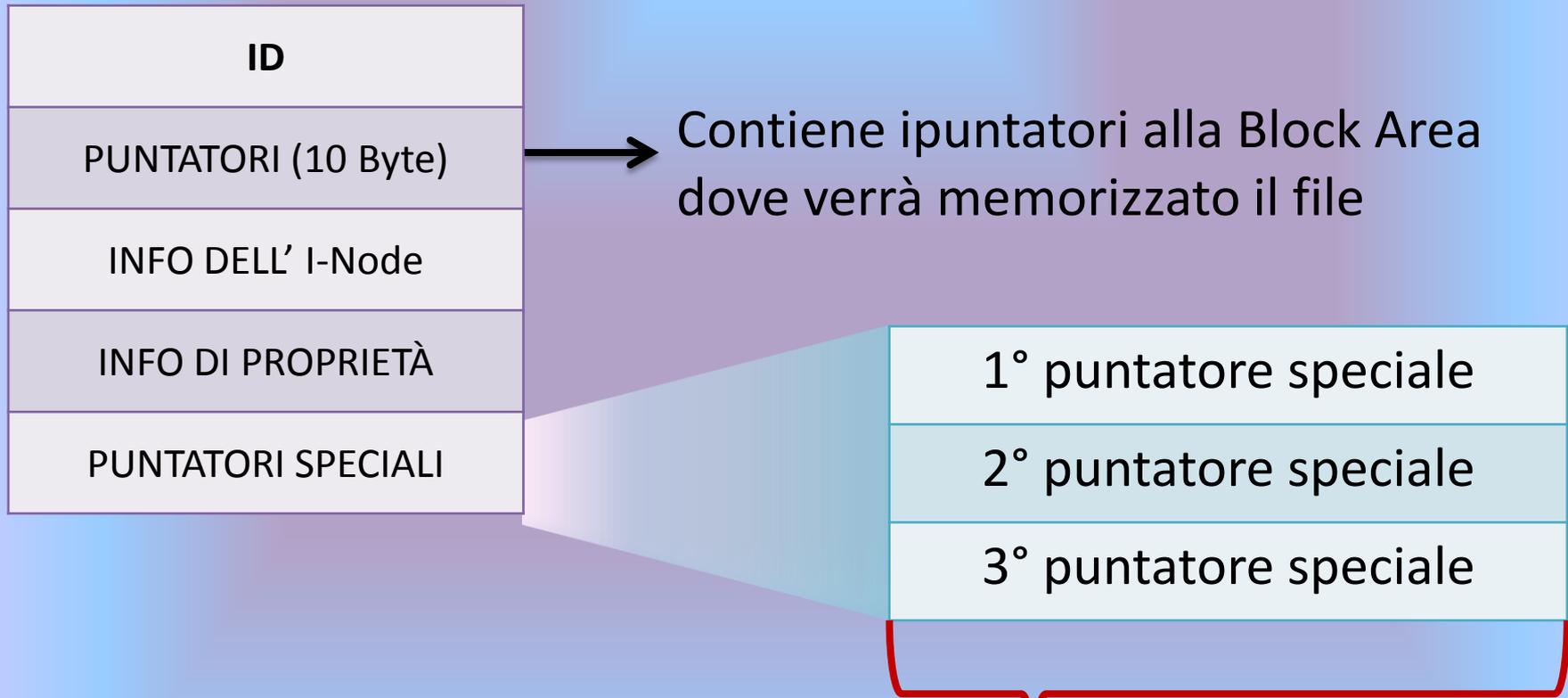
Sotto il sistema operativo UNIX il disco viene diviso in due settori:



- 512 Byte
- Contiene i blocchi necessari alla gestione dei file

- Composto da diversi blocchi (max 10) da 512 Byte ciascuno
- Contiene i veri e propri file

I-Node: la struttura



Questo tipo di puntatori vengono chiamati speciali poiché contengono l'indirizzo di altri blocchi (ognuno da 512 Byte). Essi contengono, a loro volta, 128 puntatori che si riferiscono ai file blocks.

LA RETE

Le macchine hanno la necessità di comunicare tra loro, per farlo si utilizzano Fiber Channel e/o Cavi in Rame.



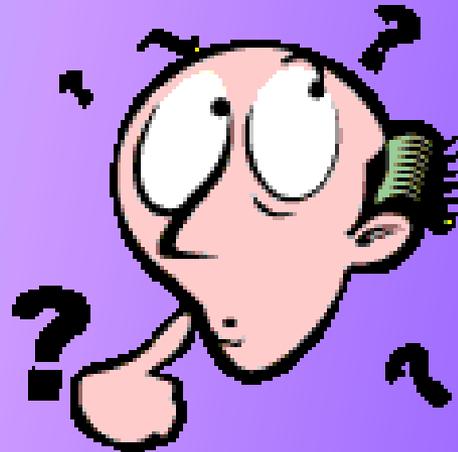
Nell' acquistare un dispositivo per gestire le comunicazioni tra le macchine si deve prestare attenzione ai protocolli che si ha intenzione di utilizzare.

IL NOSTRO PROBLEMA



Per risparmiare memoria, e quindi denaro, è necessario ottimizzare lo spazio necessario alla memorizzazione dei dati, questo è reso possibile mediante algoritmi di compressione.

Importante da notare che necessitiamo di una compressione di tipo non-lossy.



I file RAW

Il file RAW contiene i dati acquisiti dai sensori presenti all'interno dell'acceleratore DAΦNE.

Sono file di circa 2GB costituiti da numeri in virgola mobile di 4 Byte ciascuno (tipo di dato floating point).



```
#include<stdio.h>
main()
{
int i=0;
FILE *fil;
FILE *fil1;
FILE *fil2;
FILE *fil3;
FILE *fil4;

fil1=fopen("out1", "w");
fil2=fopen("out2", "w");
fil3=fopen("out3", "w");
fil4=fopen("out4", "w");
int x=fgetc(stdin);
while(x!=EOF) {
    fputc(x, fil1);
    x=fgetc(stdin);
    fputc(x, fil2);
    x=fgetc(stdin);
    fputc(x, fil3);
    x=fgetc(stdin);
    fputc(x, fil4);
    x=fgetc(stdin);
}
fclose(fil);
fclose(fil1);
fclose(fil2);
fclose(fil3);
fclose(fil4);
}
```

Il Codice

FIRST STEP:

Dividere i dati in quattro file
distinti, ognuno contenente
 $\frac{1}{4}$ dell'informazione.

```
$  
$ time compress out1
```

```
real    0m17.40s  
user    0m12.30s  
sys     0m0.69s
```

Il Codice

SECOND STEP:

Vengono compressi i singoli file mediante "compress".

```
$ ls -l  
total 24720144  
-rwxr-xr-x  1 stud13  kloe           5739 Jun 18 09:55 comp.exe  
-rw-r--r--  1 stud13  kloe           6936 Jun 17 11:13 core  
-rwxr-xr-x  1 stud13  kloe           5437 Jun 17 11:13 leggi.exe  
-rw-r--r--  1 stud13  kloe          29163368 Jun 18 09:58 out1.Z  
-rw-r--r--  1 stud13  kloe          500001280 Jun 18 09:58 out2  
-rw-r--r--  1 stud13  kloe          500001280 Jun 18 09:58 out3  
-rw-r--r--  1 stud13  kloe          500001280 Jun 18 09:58 out4  
-rw-r--r--  1 stud13  kloe          2000005120 Jun 14 15:18 pippo  
-rw-r--r--  1 stud13  kloe          29163368 Jun 18 09:51 pippo_1.Z  
-rw-r--r--  1 stud13  kloe          229464619 Jun 18 09:51 pippo_2.Z  
-rw-r--r--  1 stud13  kloe          368470379 Jun 18 09:51 pippo_3.Z  
-rw-r--r--  1 stud13  kloe          500001280 Jun 18 09:51 pippo_4  
-rw-r--r--  1 stud13  kloe           17170 Jun 14 16:02 prova  
-rw-r--r--  1 stud13  kloe          2000005120 Jun 10 09:35 raw067097M_ALL_f06_1_1  
_1.001  
-rw-r--r--  1 stud13  kloe          2000001024 Jun 10 09:36 raw067097M_ALL_f07_1_1  
_1.000  
-rw-r--r--  1 stud13  kloe          2000005120 Jun 10 09:36 raw067097M_ALL_f07_1_1  
_1.001  
-rw-r--r--  1 stud13  kloe          2000001024 Jun 10 09:36 raw067097M_ALL_f07_1_1  
_1.002  
-rwxr-xr-x  1 stud13  kloe           9475 Jun 18 09:49 split1  
-rwxr-xr-x  1 stud13  kloe           5237 Jun 17 10:45 vet.exe  
-rwxr-xr-x  1 stud13  kloe           5521 Jun 17 11:39 vet2.exe
```

```
$ time compress out2
```

```
real    0m29.30s  
user    0m23.96s  
sys     0m1.36s
```

```
$ compress ou  
ou: No such file or directory  
$ time compress out3
```

Il Codice

```
main()
{
long int vet[3];
int i=0;
int x;
fpos_t dim;

FILE *fil1;
FILE *fil2;
FILE *fil3;
FILE *fil4;
FILE *filout;

fil1=fopen("out1.Z", "r");
fil2=fopen("out2.Z", "r");
fil3=fopen("out3.Z", "r");
fil4=fopen("out4", "r");

filout=fopen("endFile", "w");

fseek(fil1, 0, SEEK_END);
fseek(fil2, 0, SEEK_END);
fseek(fil3, 0, SEEK_END);
vet[0]=ftell(fil1);
fseek(fil1, 0, SEEK_SET);
vet[1]=ftell(fil2);
fseek(fil2, 0, SEEK_SET);
vet[2]=ftell(fil3);
fseek(fil3, 0, SEEK_SET);

printf("\n");

for(i=0; i<3; i++)
{
printf("\nfile[%d]=%ld ", i, vet[i]);
}

printf("\n");

fputc(vet[0], filout);
fputc(vet[1]+vet[0], filout);
fputc(vet[2]+vet[0]+vet[1], filout);

fprintf(filout, "%li\n", vet[0]);
fprintf(filout, "%li\n", vet[0]+vet[1]);
fprintf(filout, "%li\n", vet[0]+vet[1]+vet[2]);

x=fgetc(fil1);
while(x!=EOF){
fputc(x, filout);
x=fgetc(fil1);
}

x=fgetc(fil2);
while(x!=EOF){
fputc(x, filout);
}
```

THIRD STEP:

I diversi file, dopo esser stati compressi, vengono riuniti in un unico file che verrà salvato in memoria di massa.

```
$ bc -l
29163368/500001280
.05832658668393808911
229464619/500001280
.45892806314415835095
368470379/500001280
.73693887143648912258
500001280/500001280
1.0000000000000000000000
(29163368+229464619+368470379+500001280)/2000005120
.56354838031614639066
```

I RISULTATI OTTENUTI:

- 43,65% di compressione
- Più di 400.000 € risparmiati

