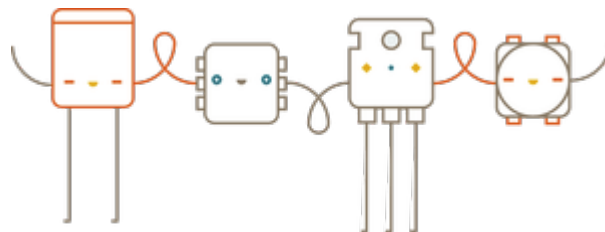


Creazione di un piccolo supercomputer per calcolo parallelo con tecnologia '*cluster*'

Marco Cavallini
KOAN - Bergamo



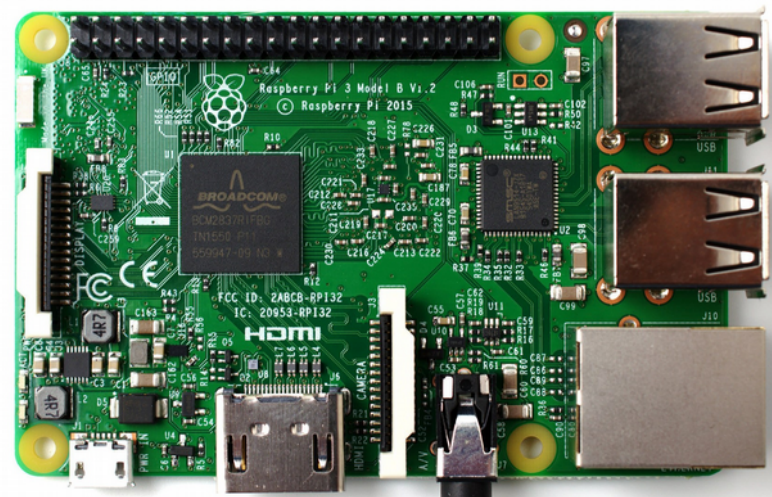
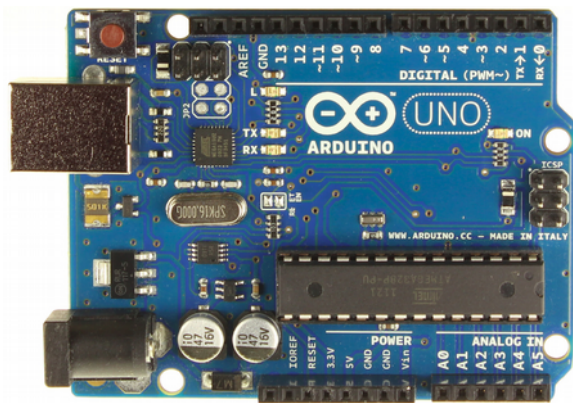
- ▶ Introduzione
 - ▶ FabLab
 - ▶ Architetture
 - ▶ Arduino vs. Raspberry PI
- ▶ Hardware
 - ▶ Raspberry PI
- ▶ Architetture
 - ▶ Multi-core
 - ▶ Cluster
- ▶ Software
 - ▶ Sistema Operativo Linux
- ▶ Calcolo parallelo
 - ▶ MPICH

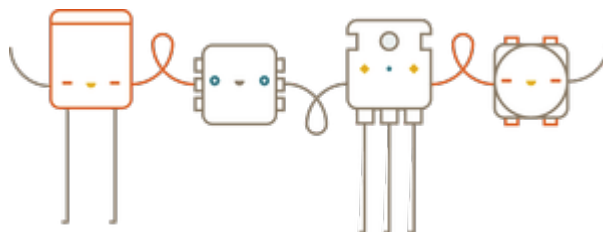


- ▶ FabLab Bergamo : **#ArduinoDay17**
- ▶ FabLab Bergamo : non solo Arduino
- ▶ Ma c'è anche Raspberry Pi (^_^)



- ▶ Una scheda Arduino tipica consiste in un microcontrollore Atmel
 - ▶ 1 core - 8-bit AVR @ 16 Mhz
- ▶ Raspberry PI consiste in un microcontrollore Broadcom
 - ▶ 1 Core - 32-bit ARM @ 700 Mhz
- ▶ Raspberry PI 3
 - ▶ 4 core - 64-bit ARM @ 1.2 GHz (1200 Mhz)



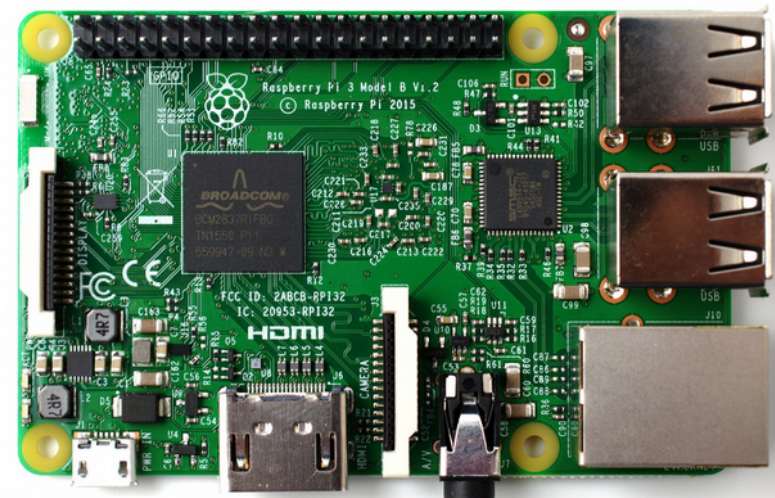


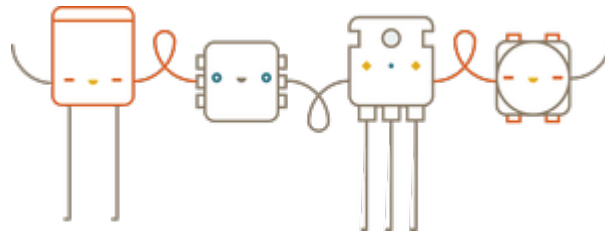
- ▶ Le prime concezioni del Raspberry Pi, nel **2006** si basavano sul microcontrollore Atmel ATmega644
- ▶ La Raspberry Pi Foundation fu fondata nel maggio **2009** nel Cambridgeshire, Regno Unito
- ▶ Il lancio al pubblico è avvenuto il 29 febbraio **2012**
- ▶ Finora, ne sono state prodotte otto versioni
 - ▶ Modelli: A, B, A+, B+, 2, Zero, 3, Zero W

https://it.wikipedia.org/wiki/Raspberry_Pi

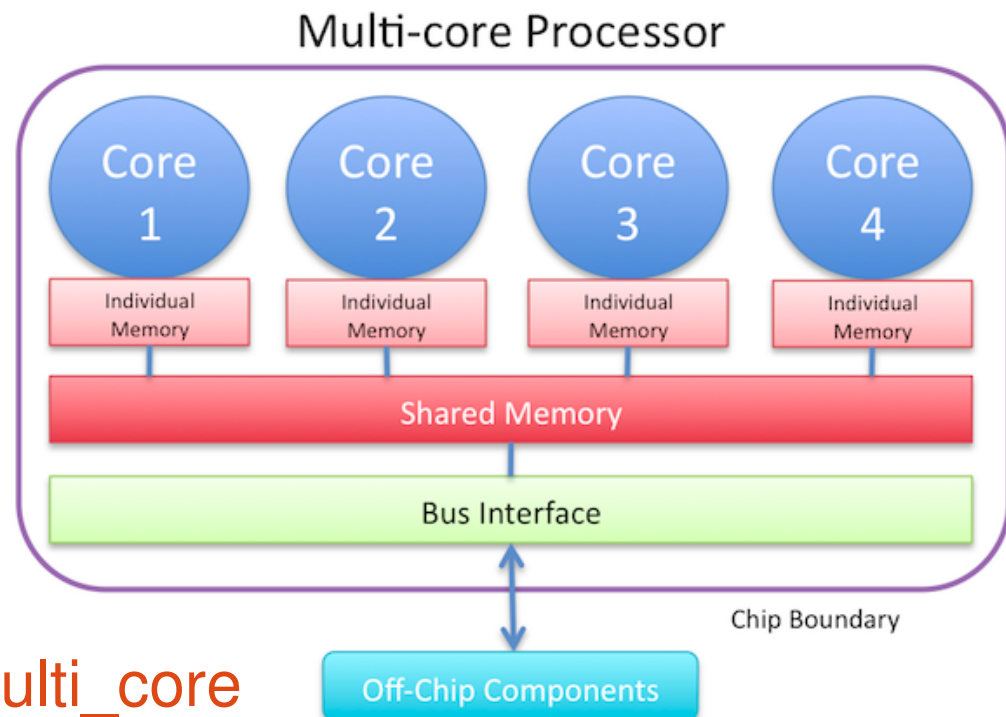


- ▶ CPU Broadcom BCM2837
- ▶ Clock 1.2 GHz 64-bit quad-core ARM Cortex-A53
- ▶ Video Core OpenGL ES 2.0, 400 MHz dual core (1080p60)
- ▶ RAM 1 GB (condivisa con la GPU) LPDDR2 (900 MHz)
- ▶ Ethernet 10/100 Mbit/s, Wireless LAN 802.11n, Bluetooth 4.1
- ▶ Prezzo 35 \$





- ▶ Il termine Multi core viene utilizzato ad indicare una CPU composta da 2 o più core, ovvero da più nuclei di processori "fisici" montati sullo stesso package.
- ▶ Il termine Multi core può essere affiancato anche con altri termini specifici della soluzione adottata per la CPU:
 - ▶ dual core, (2)
 - ▶ quad core, (4)
 - ▶ hexa core, (6)
 - ▶ octa core, (8)



https://it.wikipedia.org/wiki/Multi_core

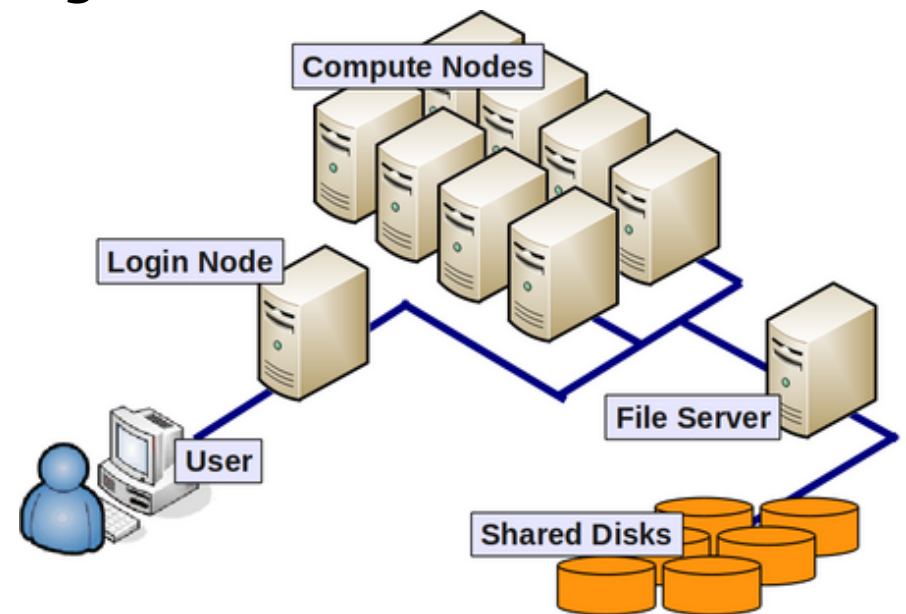


Multi Core



Many Core

- ▶ **Cluster** (dall'inglese grappolo)
- ▶ Un cluster, è un insieme di computer connessi tra loro tramite una rete
- ▶ **Caratteristiche**
 - ▶ I vari computer risultano come una singola risorsa
 - ▶ Viene suddiviso il carico di lavoro



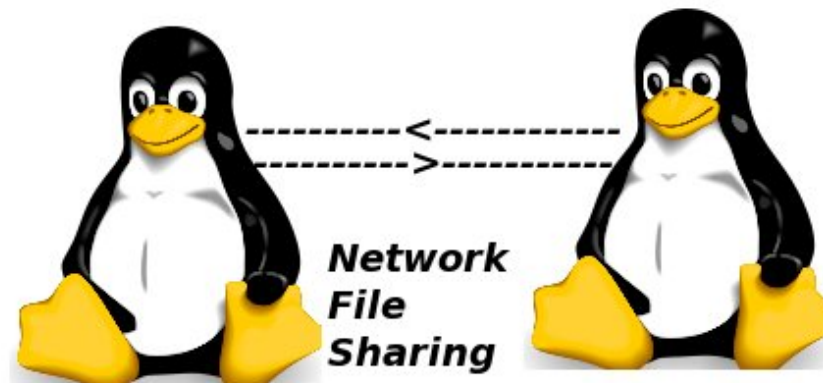
https://it.wikipedia.org/wiki/Computer_cluster

- ▶ Il progetto **SETI@home** sembrerebbe essere il **più grande cluster distribuito esistente**.
- ▶ Utilizza circa **tre milioni** di personal computer sparsi in tutto il mondo per analizzare i dati provenienti dal radiotelescopio di Arecibo, al fine di trovare la prova dell'esistenza di intelligenza extraterrestre.
- ▶ **SETI@home** ("SETI at home") è un progetto di calcolo distribuito volontario che usa computer connessi ad internet ospitato dalla Space Sciences Laboratory all'Università della California, Berkeley, negli Stati Uniti.

<https://it.wikipedia.org/wiki/SETI@home>

► Requisiti per formare un cluster di computer

- Hardware di rete ad elevate prestazioni
- Un sistema operativo distribuito in grado di far funzionare i computer come cluster (per esempio GNU/Linux)
- Linux è il sistema più utilizzato per il clustering
(secondo TOP500 nel 2006 è stato il sistema più usato in assoluto)
- Un algoritmo parallelizzabile



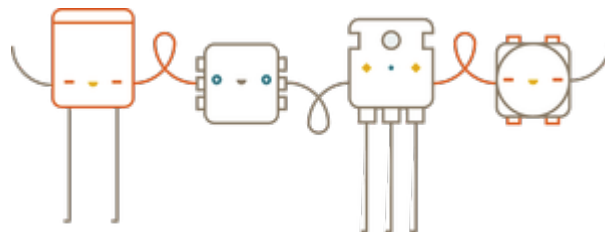
► Vantaggi

- L'economicità, (fino a 15 volte più economici dei supercalcolatori)
- La scalabilità, dal momento che le risorse sono distribuite.
- Facilità di aggiornamento e manutenzione.
- Disponibilità di un gran numero di software Open Source per i cluster
- L'affidabilità, in quanto il sistema continua a funzionare anche in caso di guasti a parti di esso (ovviamente con prestazioni inferiori)

► Svantaggi

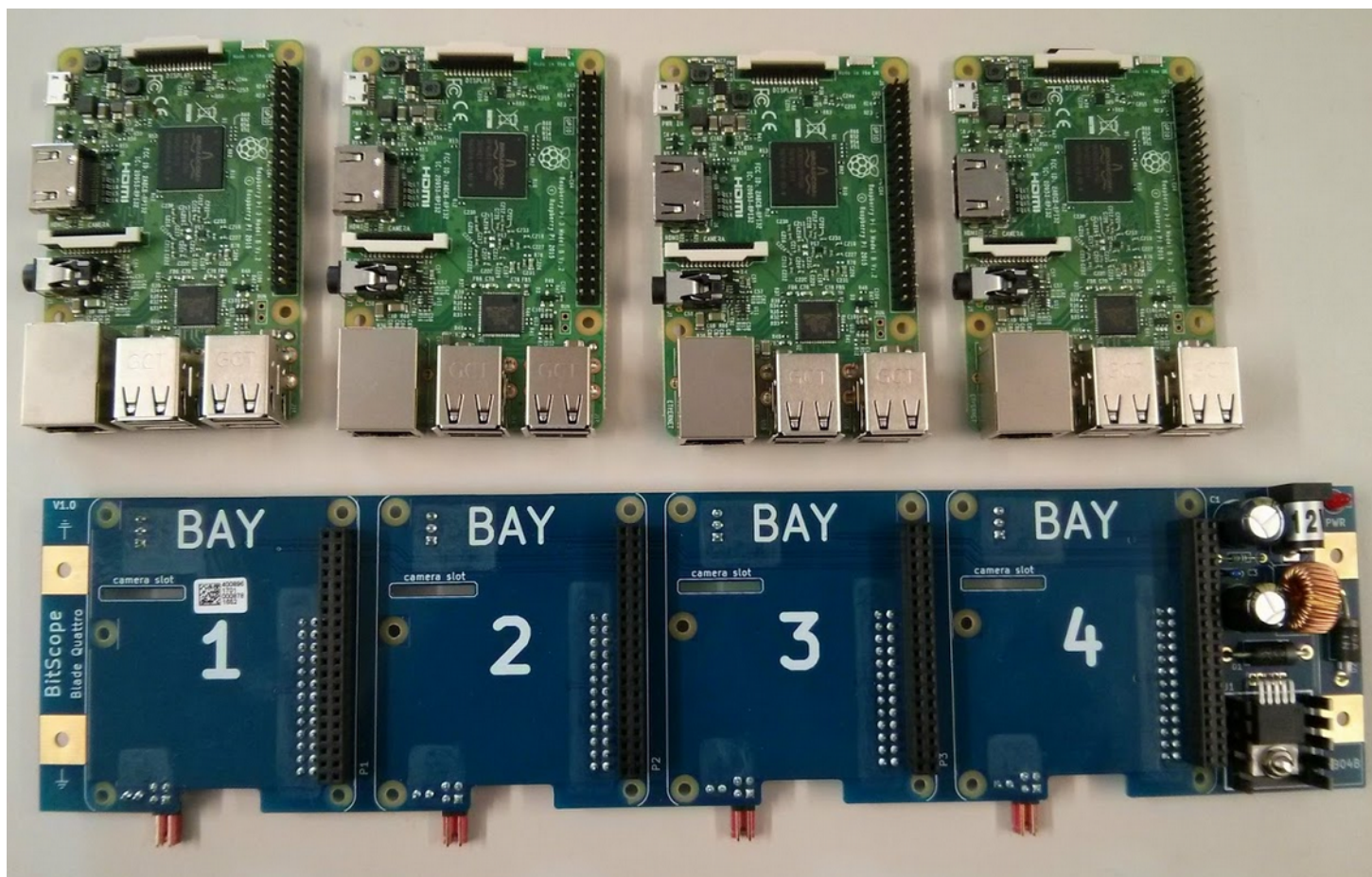
- Difficoltà di gestione di un elevato numero di computer
- Scarse prestazioni nel caso di applicazioni non parallelizzabili
- Occupazione di spazio fisico notevolmente superiore a quella di un singolo server
- Maggiore consumo di energia rispetto a un singolo server (*)

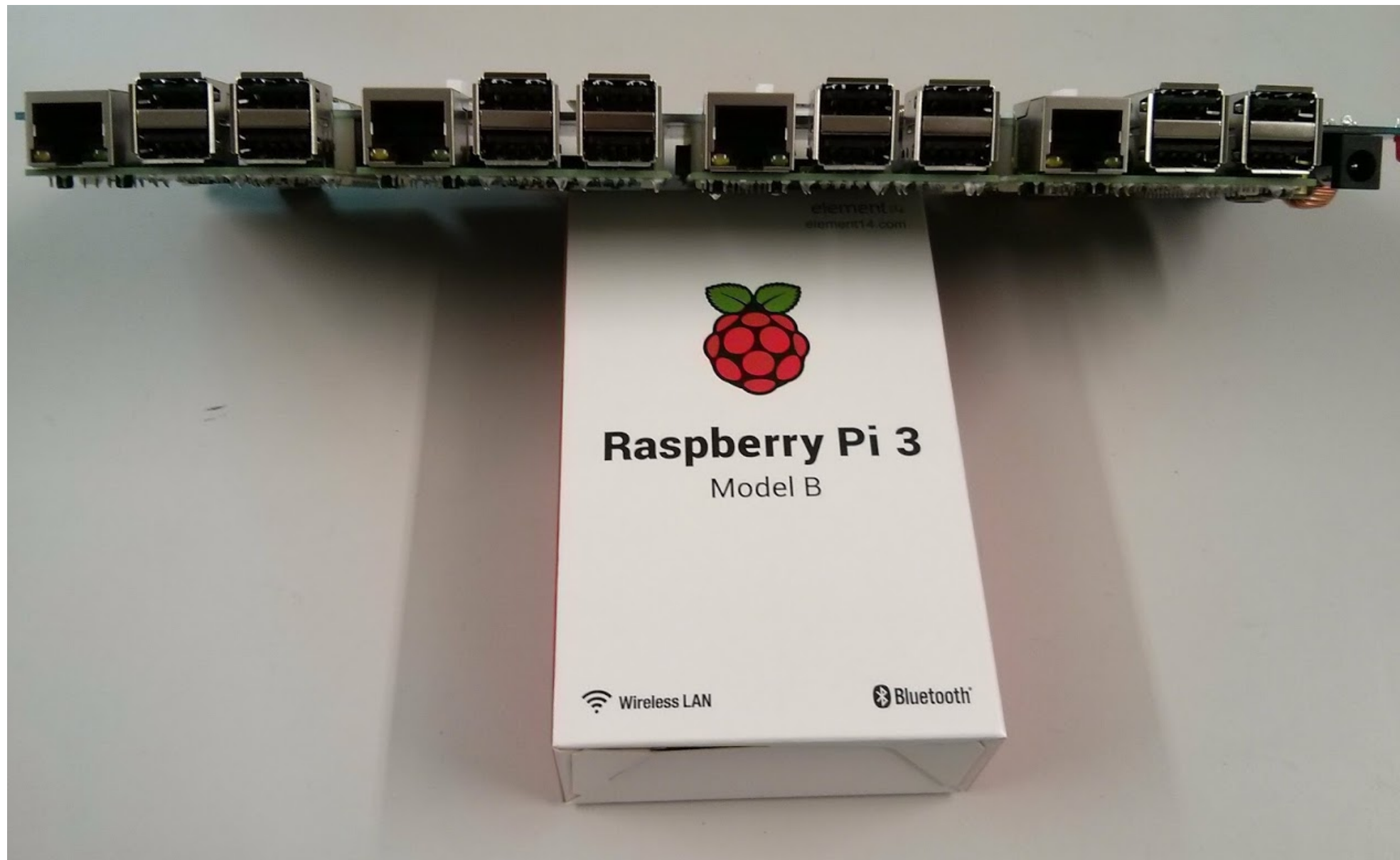
() a parità di architettura*



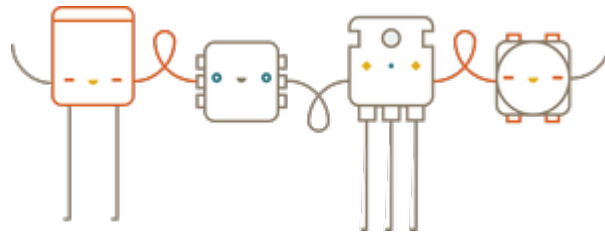


- ▶ Cluster con 4 Raspberry PI 3
- ▶ 4 core * 4 CPU = **16 core**

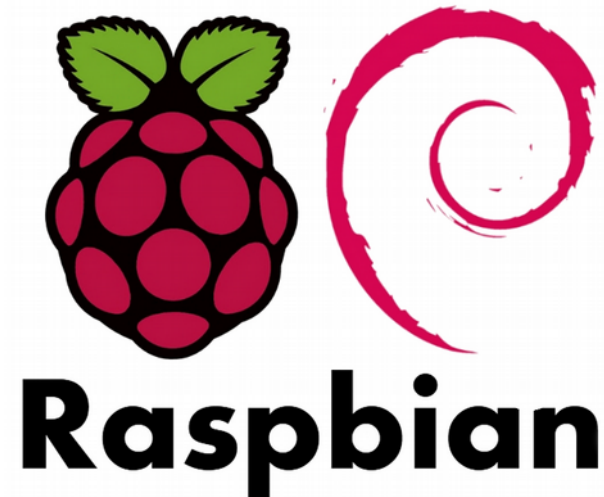
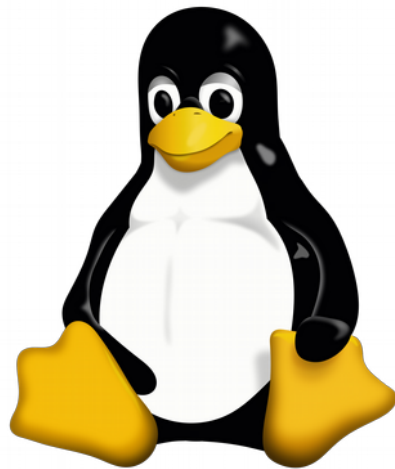






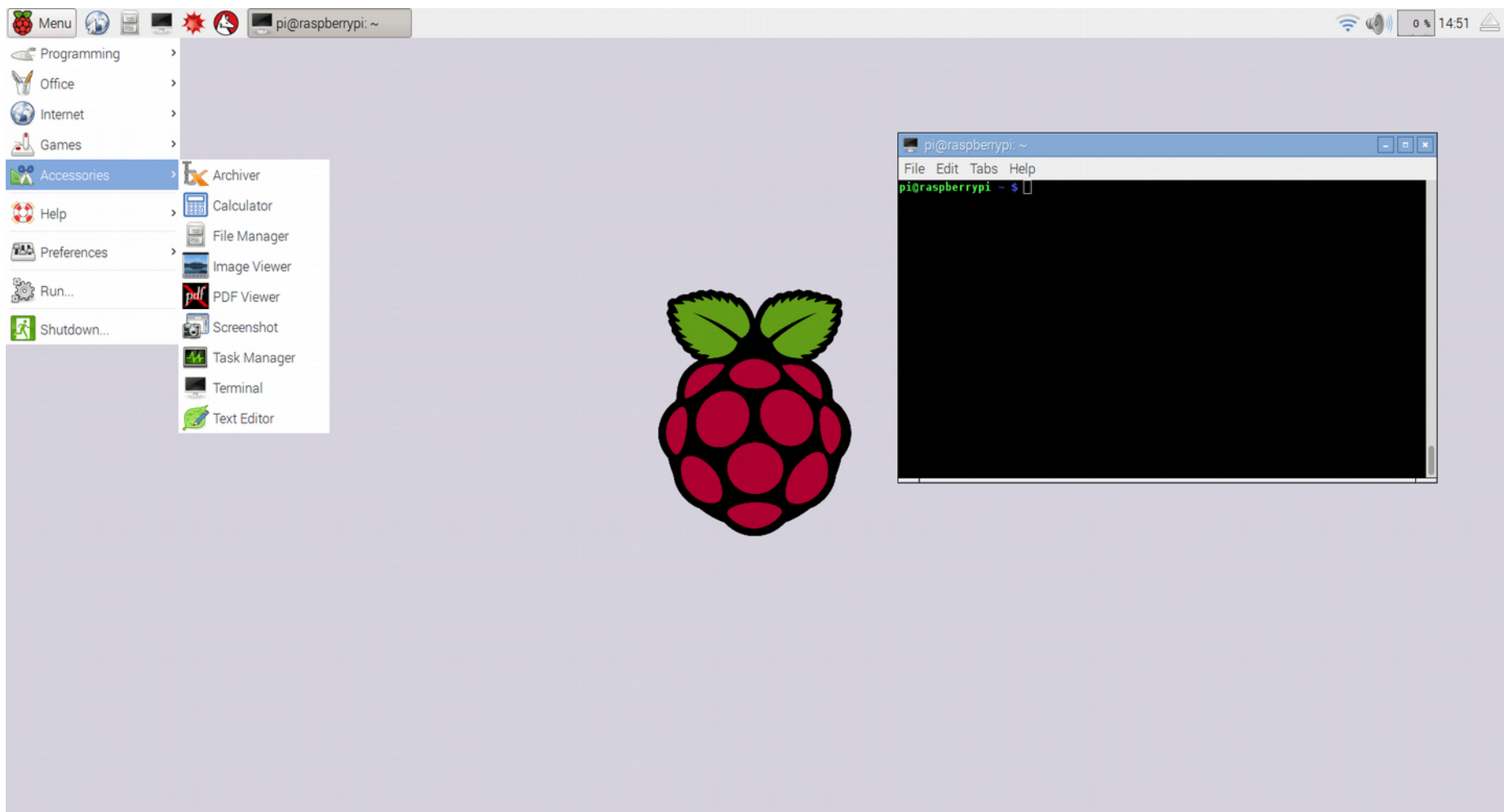


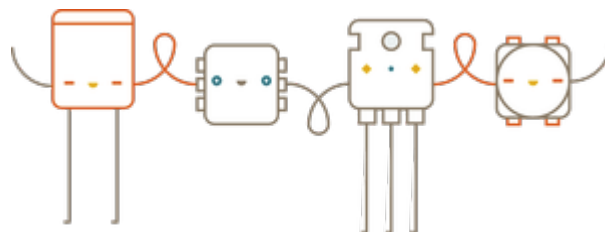
- ▶ **GNU / Linux**
- ▶ Raspberry + Debian = Raspbian



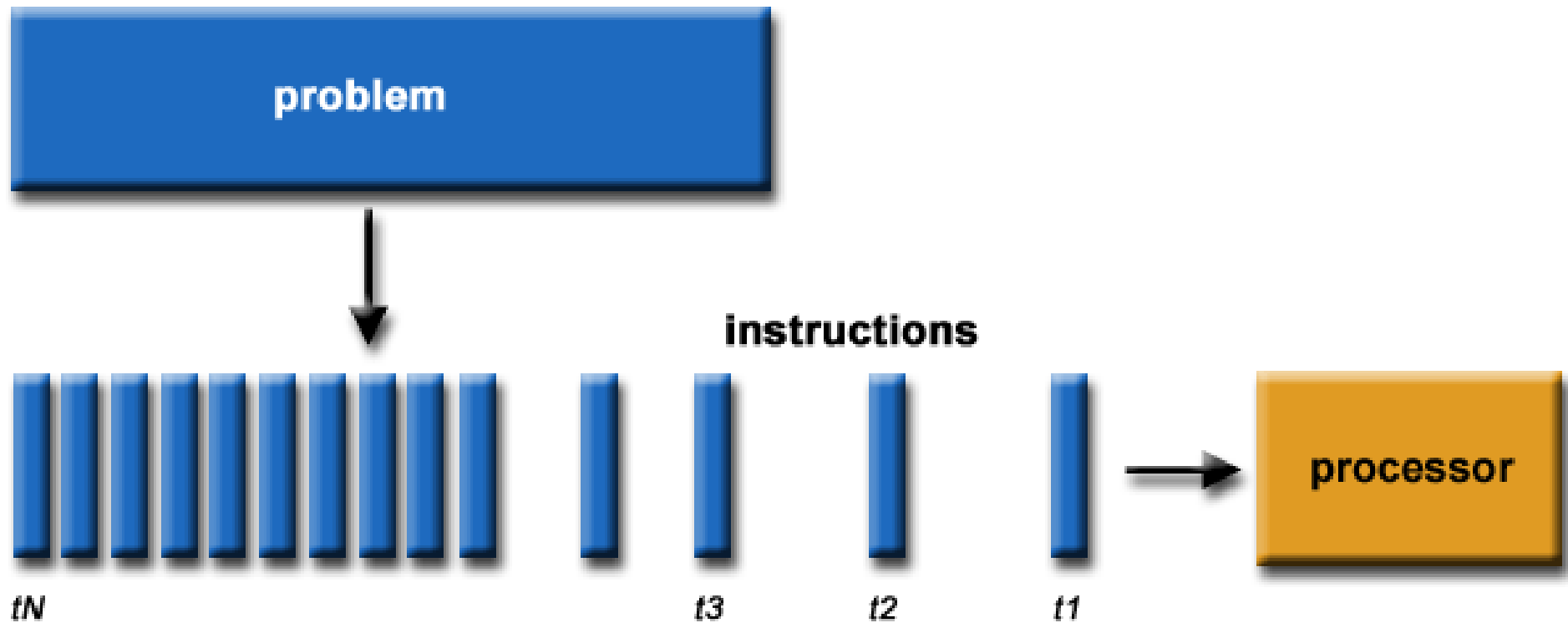
<https://www.raspberrypi.org/downloads/raspbian/>

► Aspetto di una Raspbian (collegata ad un monitor HDMI)





- ▶ **Il calcolo sequenziale**
- ▶ Per essere eseguiti su un computer con una singola CPU
- ▶ Tradizionalmente i programmi sono stati scritti per un modello di computazione sequenziale (Von Neumann).
- ▶ Un problema viene spezzato in sequenze (discrete) di istruzioni che sono eseguite in sequenza (una dopo l'altra).
- ▶ In un dato istante di tempo solo un'istruzione è in esecuzione.

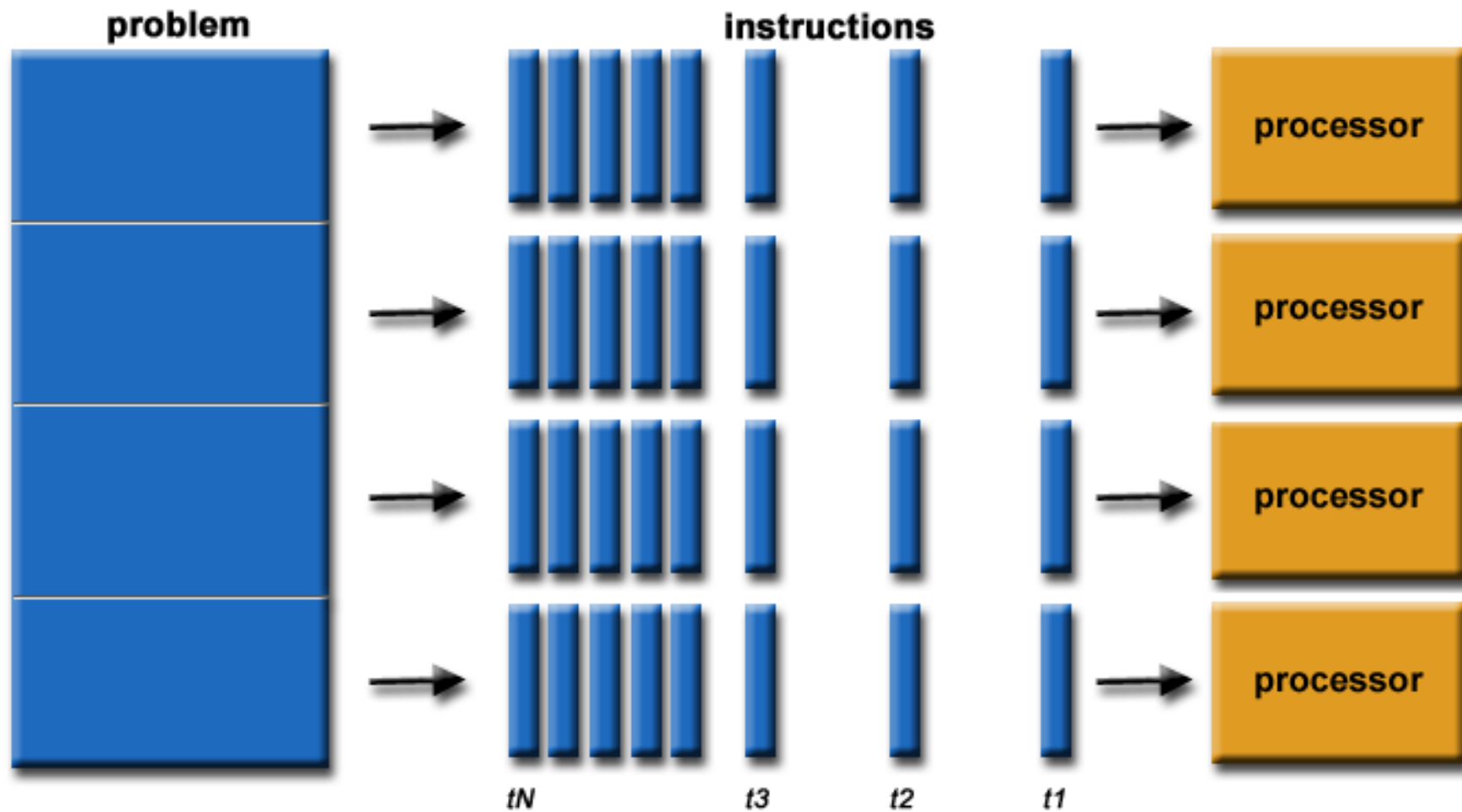


► Il calcolo parallelo

- Esecuzione simultanea del codice sorgente di uno o più programmi su più microprocessori o più core dello stesso processore allo scopo di aumentare le prestazioni di calcolo del sistema di elaborazione.
- Un problema viene decomposto in parti che posso essere risolte in maniera concorrente (in parallelo)
- I processori possono comunicare tra loro per risolvere un problema coordinatamente o funzionare in maniera totalmente indipendente



https://it.wikipedia.org/wiki/Calcolo_parallelo



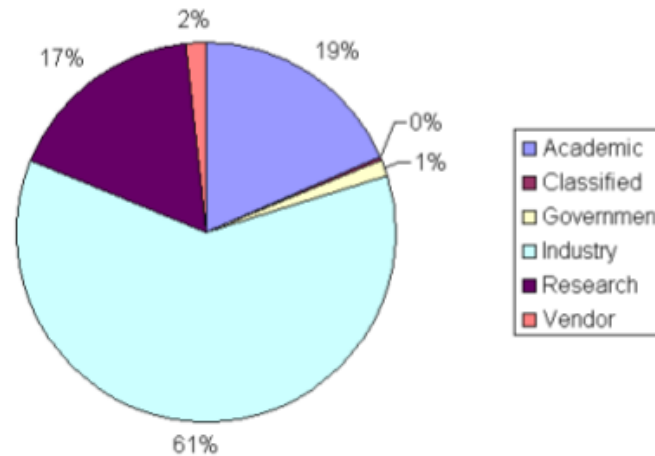
► Quali Applicazioni? (Ieri)

- Storicamente il calcolo parallelo è stato visto come un paradigma costoso e di alto livello.
- È stato quindi utilizzato principalmente per risolvere problemi scientifici e ingegneristici di alto livello:
 - Problemi atmosferici e ambientali
 - Fisica
 - Chimica
 - Geologia, fenomeni sismici
 - Ingegneria meccanica ed elettronica

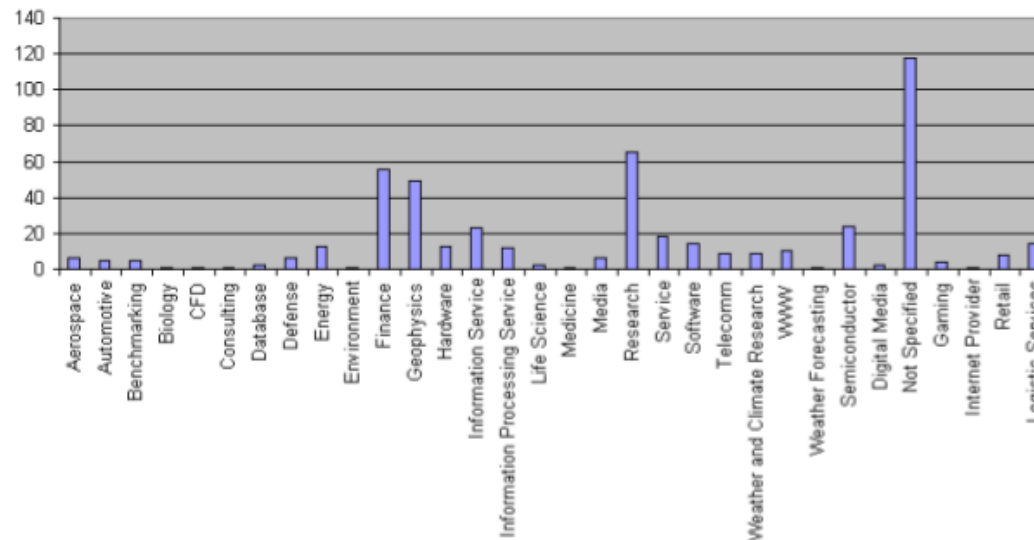
► Quali Applicazioni? (Oggi)

- Le applicazioni commerciali forniscono le motivazioni principali per lo sviluppo di calcolatori sempre più veloci.
- È necessario processare grandi quantità di dati con tecniche sofisticate e velocemente.
- Alcuni esempi
 - Database, data mining
 - Oil exploration
 - Motori di ricerca Web
 - Elaborazione immagini medicali e diagnosi
 - Modelli finanziari ed economici
 - Grafica avanzata e realtà virtuale

Who's Doing Parallel Computing?



What Are They Using It For?



Fonte: Top500.org

- ▶ Per permettere l'utilizzo in modo efficiente delle risorse sono necessari software e librerie dedicate al calcolo parallelo...
- ▶ Il **Message Passing Interface (MPI)** è un protocollo di comunicazione per computer.
- ▶ È soltanto una specifica e non una libreria
- ▶ È de facto lo standard per la comunicazione tra nodi di un cluster.
- ▶ MPI ha il vantaggio di essere molto portatile e veloce.



https://it.wikipedia.org/wiki/Message_Passing_Interface

► **Tipi di librerie**

► **MPICH** è una implementazione libera e portabile di MPI

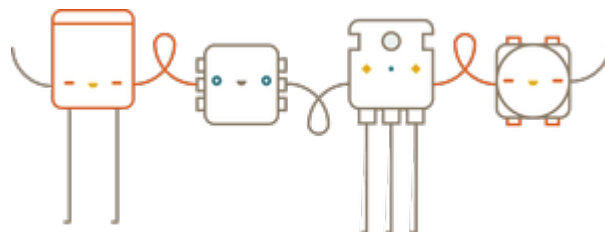
- Licenza BSD-Like
- La prima implementazione di MPICH si chiama MPICH1
- Attualmente l'ultima implementazione si chiama MPICH2 la quale implementa lo standard MPI-2.0

► **Open MPI** è un'altra possibile implementazione di MPI

- Licenza BSD

► MPICH e Open MPI sono entrambi disponibili per varie versioni di Unix (incluso Linux e macOS) e Microsoft Windows.

<https://it.wikipedia.org/wiki/MPICH>



► Installazione

MPI

Ultima versione

mpich-3.2 (stable release)

Versione Debian

Package: mpich

Version: 3.1-5

Architecture: armhf

<http://mpitutorial.com/tutorials/installing-mpich2/>

► Hello World in C (sequenziale)

```
#include <stdio.h>
```

```
int main (int argc, char* argv[])  
{  
    printf( "Hello world\n" );  
    return 0;  
}
```

► Hello World in C con MPICH

```
#include <stdio.h>
#include <mpi.h>

int main (int argc, char* argv[])
{
    int rank, size;

    MPI_Init (&argc, &argv);
    MPI_Comm_rank (MPI_COMM_WORLD, &rank);
    MPI_Comm_size (MPI_COMM_WORLD, &size);

    printf( "Hello world dal processo %d di %d\n", rank, size );

    MPI_Finalize();

    return 0;
}
```

- ▶ **Scoprire informazioni sull'ambiente**
- ▶ Ogni processo può usare run-time due funzioni per sapere
 - ▶ Quanti processi partecipano a questa computazione
 - ▶ Localizzarsi all'interno della computazione (cioè scoprire il suo ID)
 - ▶ **MPI_Comm_size** restituisce il numero di processi della computazione
 - ▶ **MPI_Comm_rank** restituisce il rank (o ID) del processo, che è sempre compreso tra 0 e n-1 (dove n è il numero di processi che partecipano alla computazione)
 - ▶ In MPI i processi possono essere raggruppati attraverso oggetti chiamati comunicatori.
 - ▶ **MPI_COMM_WORLD** è il comunicatore di default che comprende tutti i processi che partecipano alla computazione

- ▶ **Compilare ed eseguire un programma MPI**
- ▶ Lo standard MPI non specifica come eseguire i programmi
- ▶ Ogni implementazione fornisce appositi strumenti (programmi o script) per compilare ed eseguire un programma MPI
- ▶ Ad esempio in OpenMPI e in MPICH

```
mpicc -o emyprog mpi_c.c
```

```
mpiexec -l -n 5 ./myprog
```

Dove `-n <N>` specifica che il programma sarà composto da N processi paralleli

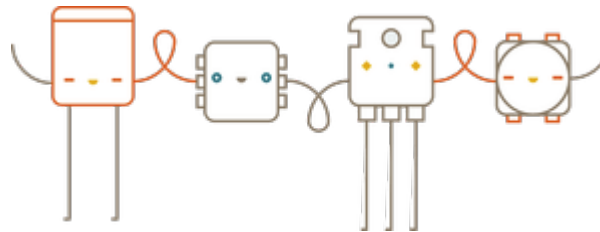
- In output si avrà una cosa del genere:

```
0: Hello world dal processo 0 di 5
1: Hello world dal processo 1 di 5
3: Hello world dal processo 3 di 5
2: Hello world dal processo 2 di 5
4: Hello world dal processo 4 di 5
```

```
pi@rpi0$:
```

```
./mpirun -n 16 -f host_file ./mpi-hello-world/code/mpi_hello_world
```

```
Hello world from processor rpi0, rank 4 out of 16 processors
Hello world from processor rpi0, rank 12 out of 16 processors
Hello world from processor rpi0, rank 8 out of 16 processors
Hello world from processor rpi1, rank 5 out of 16 processors
Hello world from processor rpi1, rank 9 out of 16 processors
Hello world from processor rpi1, rank 1 out of 16 processors
Hello world from processor rpi1, rank 13 out of 16 processors
Hello world from processor rpi0, rank 0 out of 16 processors
Hello world from processor rpi2, rank 2 out of 16 processors
Hello world from processor rpi3, rank 11 out of 16 processors
Hello world from processor rpi2, rank 6 out of 16 processors
Hello world from processor rpi3, rank 15 out of 16 processors
Hello world from processor rpi2, rank 14 out of 16 processors
Hello world from processor rpi3, rank 3 out of 16 processors
Hello world from processor rpi2, rank 10 out of 16 processors
Hello world from processor rpi3, rank 7 out of 16 processors
```





- ▶ **Misurare la temperatura della CPU su raspberry PI**
- ▶ `/opt/vc/bin/vcgencmd measure_temp`

Temperatura normale in idle

temp=42.9°C -> (senza dissipatore)
temp=38.6°C -> (con dissipatore)

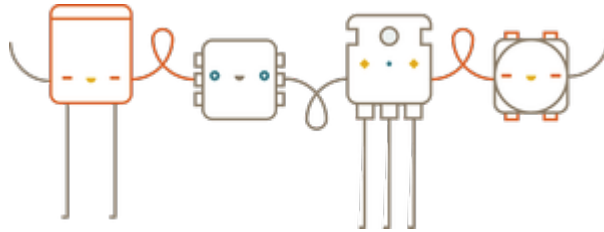
100% carico MPI sul master (dopo 10')

temp=**80.6°C** -> (senza dissipatore)
temp=67.7°C -> (con dissipatore)
temp=72.0°C -> (con dissipatore dopo 30')

100% carico MPI sui client (dopo 10')

temp=60.1°C -> (senza dissipatore)

Domande?



- ▶ Wikipedia
- ▶ **Introduzione a MPI**, Dott. Daniele Loiacono – Politecnico di Milano
- ▶ <http://home.deib.polimi.it/loiacono/>

© Copyright 2017, Marco Cavallini - KOAN sas
[m.cavallini <AT> koansoftware.com](mailto:m.cavallini@koansoftware.com)

Corrections, suggestions,
contributions and translations are welcome!



<ftp://ftp.koansoftware.com/public/talks/ArduinoDay2017>

Attribution – ShareAlike 3.0

You are free

to copy, distribute, display, and perform the work
to make derivative works
to make commercial use of the work

Under the following conditions



Attribution. You must give the original author credit.



Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

For any reuse or distribution, you must make clear to others the license terms of this work.

Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

License text: <http://creativecommons.org/licenses/by-sa/3.0/legalcode>

Embedded Linux Training

- Unix and GNU/Linux basics
- Linux kernel and drivers development
- Real-time Linux
- Yocto Project
- Development and profiling tools
- Lightweight tools for embedded systems
- Root filesystem creation
- Audio and multimedia
- System optimization

KOAN services

Custom Development

- System integration
- BSP creation for new boards
- System optimization
- Linux kernel drivers
- Application and interface development

Consulting

- Help in decision making
- System architecture
- Identification of suitable technologies
- Managing licensing requirements
- System design and performance review

Technical Support

- Development tool and application support
- Issue investigation and solution follow-up with mainstream developers
- Help getting started



<http://koansoftware.com>

