

COMPUTER SCIENCE

SOFTWARE

STATISTICS

MACHINE LEARNING

Prof. Lorian Storch
loriano@storch.org
<https://www.storch.org/>



Software

COMPUTER SCIENCE

- **Software di base** : dedicato alla gestione del computer stesso, ad esempio il sistema operativo.

- **Software applicativo** : dedicato alla creazione di applicazioni specifiche, come la navigazione in internet, l'elaborazione testi o altro.

MACHINE LEARNING



COMPUTER SCIENCE

SISTEMI OPERATIVI

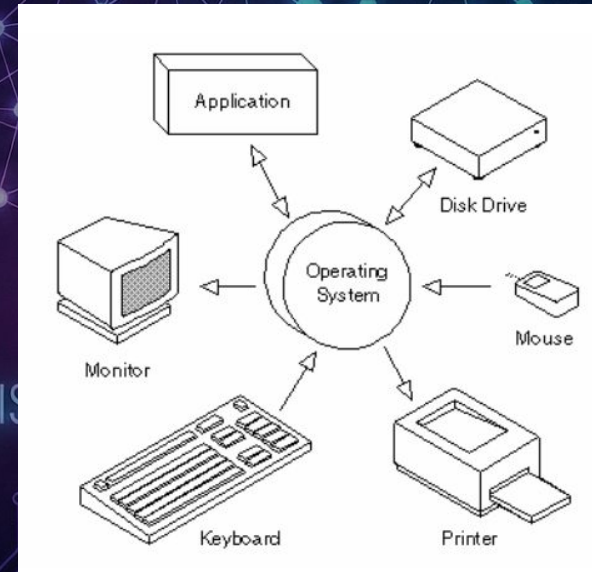
STATISTICS

MACHINE LEARNING



Sistema operativo

- Si tratta di software di basso livello che aiuta l'utente, mentre le applicazioni di alto livello interagiscono con l'hardware e i dati che i programmi memorizzano sul computer.
- Un sistema operativo esegue operazioni di base, come il riconoscimento dell'input dalla tastiera, l'invio dell'output al display, la gestione di directory e file sul disco e il controllo di periferiche come le stampanti.



Sistema operativo

- **Esecuzione dei programmi** : i sistemi operativi forniscono un ambiente in cui l'utente può eseguire programmi applicativi senza doversi preoccupare dell'allocazione della memoria o della pianificazione della CPU.
- **Operazioni di I/O** : ogni applicazione richiede un input e produce un output. Il sistema operativo nasconde all'hardware sottostante i dettagli necessari per gestire questo tipo di operazioni .
- **Comunicazione** : Esistono situazioni in cui due applicazioni devono comunicare tra loro, sia che si trovino sullo stesso computer (processi diversi) sia su computer diversi . Il sistema operativo è responsabile della gestione di questo tipo di comunicazione interprocesso.



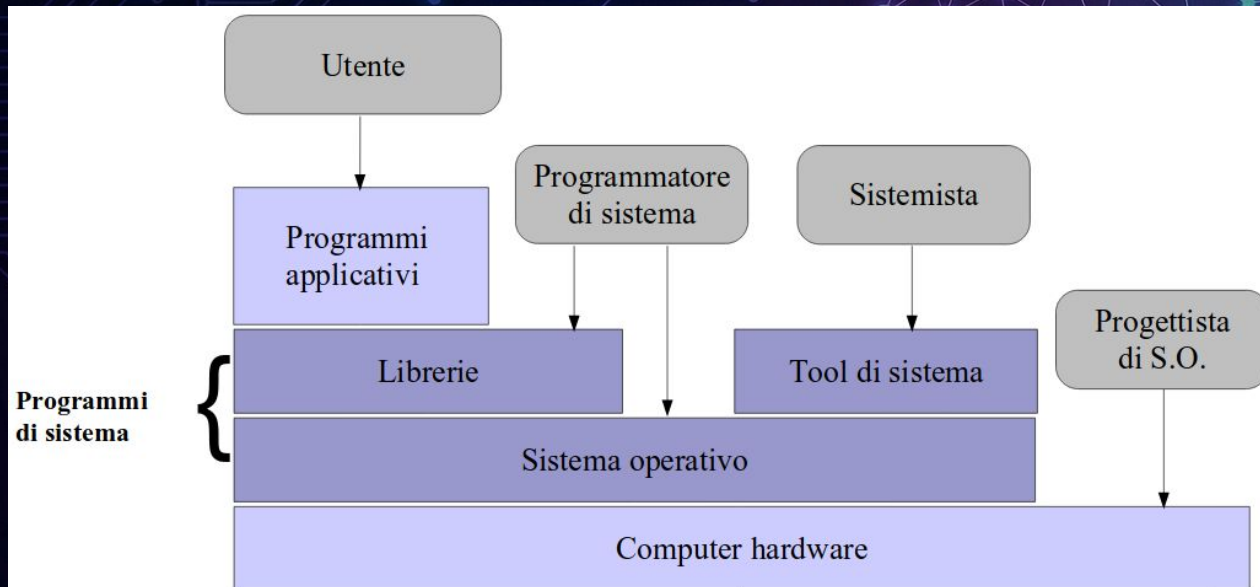
OPERATING SYSTEMS

They make it easier to write application programs that do not have to worry about specific hardware characteristics.



Sistemi operativi

Vista a livelli dei componenti hardware e software, ad esempio fornisce al programmatore **un'API facile da usare** , nasconde i dettagli hardware



MACHINE LEARNING



COMPUTER SCIENCE

STORIA E CARATTERISTICHE PRINCIPALI DEI SISTEMI OPERATIVI

STATISTICS

MACHINE LEARNING



Sistema operativo: storia

- **Babbage (1792-1871) tenta di costruire una macchina meccanica analitica e programmabile senza un sistema operativo.**
 - **La prima programmatrice donna della storia, Lady Ada Lovelace (figlia di Byron)**
- **Le macchine a valvole (1944-1955) sono progettate, costruite e programmate da un unico gruppo di persone.**
 - **Programmato in linguaggio macchina , utilizzato esclusivamente per calcoli numerici.**
 - **Non esiste un sistema operativo , non c'è distinzione tra progettista, programmatore e utente.**
 - **L'utente singolo scrive il programma , lo carica, carica i dati, attende l'output e poi passa all'esecuzione del programma successivo.**
 - **Nessuno immagina che i computer possano mai uscire dai laboratori di ricerca.**



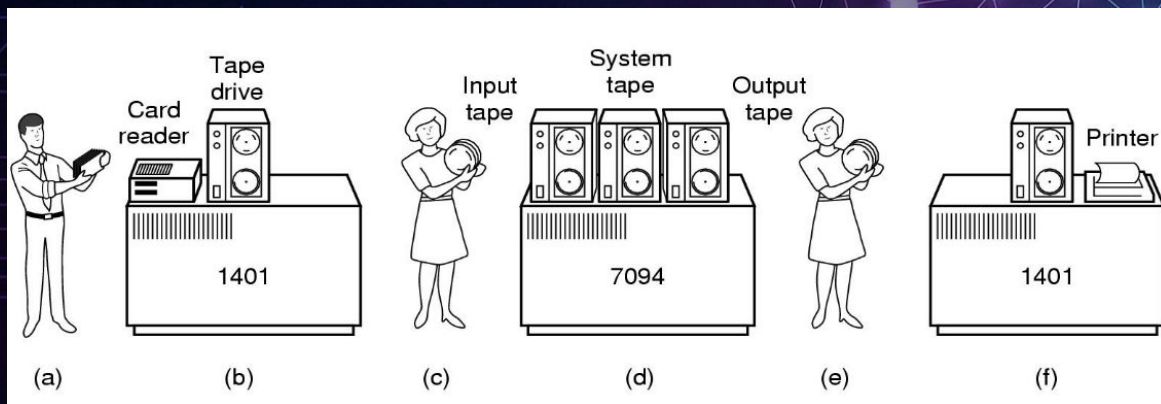
Sistema operativo: storia

- **I transistor (1955-1965) consentono di realizzare macchine più affidabili ed economiche.**
- **Stanno iniziando a essere utilizzati per compiti diversi dal semplice calcolo numerico.**
- **La persona che costruisce la macchina è diversa dalla persona che la programma e quindi la utilizza (utente == programmatore).**
- **vengono introdotti linguaggi di "alto livello" come Assembly e FORTRAN e si utilizzano schede perforate .**
- **Primi esempi di sistemi operativi, chiamati monitor residenti :**
 - **Il controllo della macchina è affidato al monitor**
 - **Una volta terminato il lavoro, il controllo torna al monitor.**



Sistema operativo: storia

- Per evitare tempi di inattività tra l'esecuzione di un lavoro e l'altro, sono stati introdotti i nastri per l'archiviazione dei lavori.



MACHINE LEARNING



Sistema operativo: storia

- **Circuiti integrati (1965-1980): scompare la figura dell'operatore come interfaccia con il computer .**
 - La programmazione viene eseguita **principalmente utilizzando linguaggi di alto livello** come il C
 - Ecco gli **editor di testo** e i terminali che ti permettono di operare
 - **sistemi operativi** con funzionalità moderne:
 - **Interattivo**
 - **Multiprogrammazione**
 - **time-sharing**

STATISTICS

MACHINE LEARNING



Sistema operativo: Multiprogrammazione

- **Utilizza il processore mentre altri processi eseguono operazioni di I/O.**
- **Quindi ci sono più processi in memoria contemporaneamente.**
- **Lo scheduler (componente del sistema operativo) gestisce l'utilizzo della CPU :** mentre un processo esegue operazioni di I/O, l'altro utilizza la CPU, **eliminando così i tempi morti (CPU inattiva).**
- **Quindi le routine di I/O devono essere fornite dal sistema operativo**
- **Il sistema operativo deve occuparsi dell'allocazione della memoria per più processi**
- **Analogamente, il sistema operativo deve essere in grado di allocare le risorse di I/O tra i diversi processi.**



Sistema operativo: time-sharing

- Si tratta essenzialmente di **un'estensione del concetto di multiprogrammazione.**
- Il tempo di esecuzione della CPU è suddiviso in intervalli (**quanti**). Quando un quanto scade, l'esecuzione del processo corrente viene interrotta, anche se non necessita di operazioni di I/O, e viene avviato un altro processo (job). I processi appartengono generalmente a utenti diversi.
- Il **cambio di contesto** è molto rapido e trasparente per l'utente, che ha l'impressione che molti programmi vengano eseguiti in parallelo.
- La presenza di **più utenti** rende necessario includere meccanismi di protezione della memoria e del file system.
- Un lavoro viene caricato dal disco alla memoria e viceversa (**scambio**).



Sistema operativo: storia

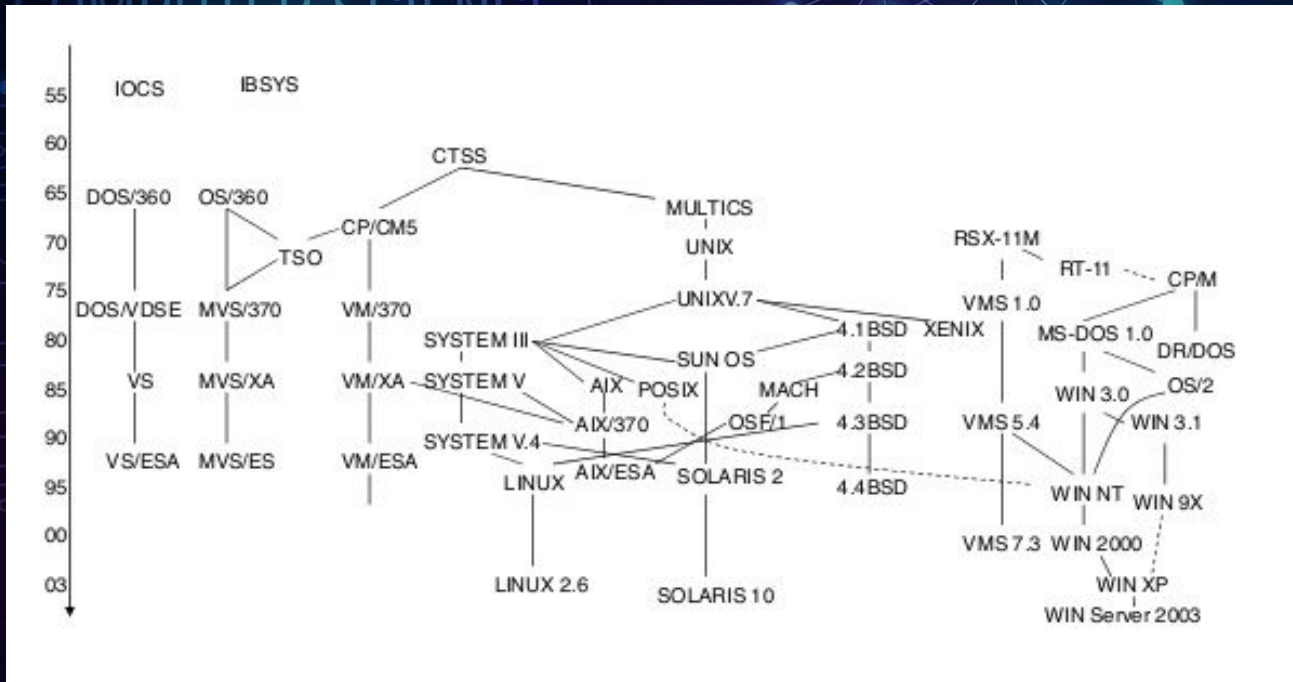
COMPUTER SCIENCE

- **CTSS (Compatible Time-Sharing System) 1965** : Vengono introdotti il concetto di multiprogrammazione e il concetto di time-sharing.
- **Multics 1965** : introduce il concetto di processo
- **Unix 1970** : derivato di Multics e CTSS, inizialmente sviluppato presso i Bell Labs. Inizialmente sviluppato su due architetture specifiche, poi sviluppato in C (inizialmente interamente in linguaggio assembly).

LEARNING



Sistemi operativi: storia



LEARNING



COMPUTER SCIENCE

IN BREVE: I COMPONENTI FONDAMENTALI DI UN SISTEMA OPERATIVO

STATISTICS

MACHINE LEARNING



Sistema operativo: Gestione dei processi

COMPUTER SCIENCE

- Un processo è essenzialmente **un programma in esecuzione che utilizza risorse**
- Il sistema operativo deve essere in grado di:
 - **Creare e terminare i processi stessi**
 - **Gestire la comunicazione tra i processi stessi**
 - **Sospendere e riprendere i processi**

MACHINE LEARNING



Sistema operativo: Gestione della memoria principale

- La memoria principale è essenzialmente un array di byte indirizzabili individualmente, che possono essere condivisi tra la CPU e i dispositivi di I/O.
- Il sistema operativo deve essere in grado di:
 - Ad esempio, tenere traccia di quali aree di memoria vengono utilizzate e da chi (tramite quale processo)
 - Ad esempio, decidere a quali processi allocare una determinata area di memoria quando questa è libera.
 - Infine, assegna spazio per liberare memoria



Sistema operativo: Gestione della memoria secondaria

COMPUTER SCIENCE

- I computer che conosciamo sono dotati di una **memoria secondaria non volatile** di grandi dimensioni (maggiore della memoria principale, che è volatile).
- Il sistema operativo deve essere in grado di gestire le seguenti attività:
 - **Assegna lo spazio necessario e liberalo** quando non viene più utilizzato.
 - **Gestisci la pianificazione dell'accesso ai dispositivi** (ad esempio dischi rigidi, CD/DVD o chiavette USB).



Sistema operativo: Gestione del filesystem

COMPUTER SCIENCE

- Un **file** (archivio) è l'astrazione informatica del concetto di contenitore di informazioni, indipendentemente dal dispositivo su cui è memorizzato.
- Un **filesystem** è composto da molti **file** (una **directory** contiene riferimenti ad altri file). Il sistema operativo deve essere in grado di gestire i seguenti compiti:
 - **Creare ed eliminare file e directory**, manipolarli
 - **Crittografa il filesystem** sulla memoria secondaria
 - Esempi: ext3, ext4, NTFS, FAT32 ...

STATISTICS

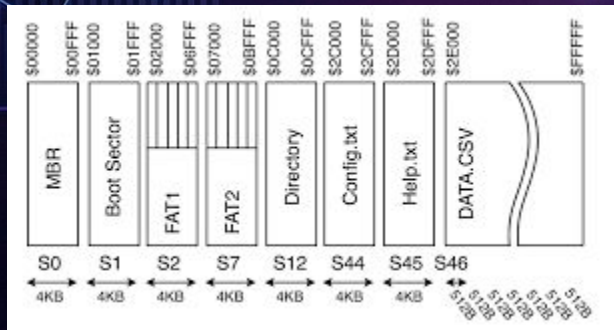
MACHINE LEARNING



Sistema operativo: Gestione del filesystem

COMPUTER SCIENCE

- Il **filesystem** deve gestire l'allocazione dello spazio su disco, ad esempio mantenendo un indice della posizione in cui sono memorizzati i dati relativi a un determinato file. Deve inoltre gestire le caratteristiche di un file, come i dati di accesso, i permessi, i nomi, ecc.



MACHINE LEARNING

Sistema operativo: Gestione I/O

COMPUTER SCIENCE

- Il sistema operativo deve gestire l'input/output e quindi l'interazione con i vari dispositivi hardware:
 - Un'interfaccia comune per la gestione di vari **driver di dispositivo**.
 - Avere diversi **driver (moduli del kernel)** per diversi dispositivi, quindi componenti specifici per i vari componenti hardware del sistema
 - Un sistema per il **buffering e la memorizzazione nella cache** delle informazioni da e verso vari dispositivi

MACHINE LEARNING



Sistema operativo: Protezione

COMPUTER SCIENCE

- Il sistema operativo deve implementare un **meccanismo di protezione del software** . Ciò significa gestire e controllare l'accesso dei vari programmi (processi) alle risorse di sistema condivise (ad esempio, la memoria).
 - Distinguere tra uso autorizzato e non autorizzato
 - Fornire meccanismi di base per attuare la protezione

STATISTICS

MACHINE LEARNING



Sistema operativo: Rete

COMPUTER SCIENCE

- La creazione di reti è ormai una componente essenziale di un sistema operativo, ovvero la capacità di far comunicare due o più computer (processi) tra loro e di far loro condividere le risorse.
- Un sistema operativo fornisce protocolli di comunicazione di base come **TCP/IP, UDP**
- Oltre a servizi di comunicazione di alto livello come i sistemi di file condivisi (SMB, NFS) e molti altri.

STATISTICS

MACHINE LEARNING



Sistema operativo: interprete dei comandi

COMPUTER SCIENCE

- I sistemi operativi forniscono un'interfaccia utente:

- Avviare o terminare un programma
- Interagisci con i componenti di base del sistema operativo, come il filesystem

- Possiamo essenzialmente distinguere due tipi:

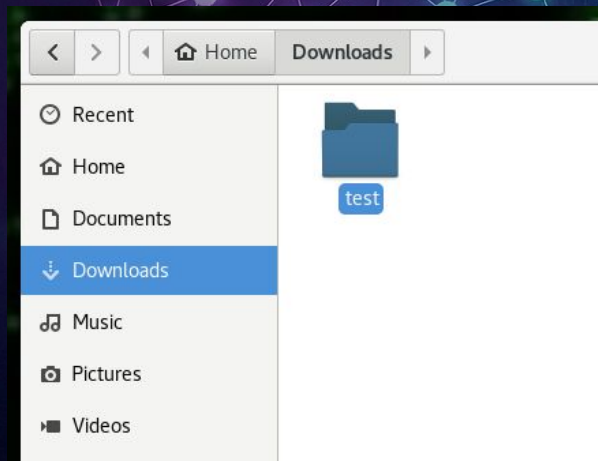
- Grafica, e quindi icone e finestre
- Testo, riga di comando

MACHINE LEARNING



Sistema operativo: interprete dei comandi

```
redo@banquo:~/Downloads  
[redo@banquo ~]$ cd Downloads/  
[redo@banquo Downloads]$ ls  
[redo@banquo Downloads]$ mkdir test  
[redo@banquo Downloads]$ rmdir test/  
[redo@banquo Downloads]$
```



INE LEARNING



COMPUTER SCIENCE

CAMBIAMENTO DI PROSPETTIVA

STATISTICS

MACHINE LEARNING



Sistema operativo: la prospettiva del programmatore

- La chiamata di sistema (o le chiamate di sistema), ad esempio:

UNIX

fork	CreateProcess
waitpid	WaitForSingleObject
execve	-
exit	ExitProcess
open	CreateFile
close	CloseHandle
read	ReadFile
write	WriteFile
lseek	SetFilePointer
stat	GetFileAttributesEx

WIN32

UNIX

mkdir	CreateDirectory
rmdir	RemoveDirectory
link	-
unlink	DeleteFile
mount	-
umount	-
chdir	SetCurrentDirectory
chmod	-
kill	-
time	GetLocalTime

WIN32

MACHINE LEARNING



COMPUTER SCIENCE

Macchine virtuali, Docker e cloud computing

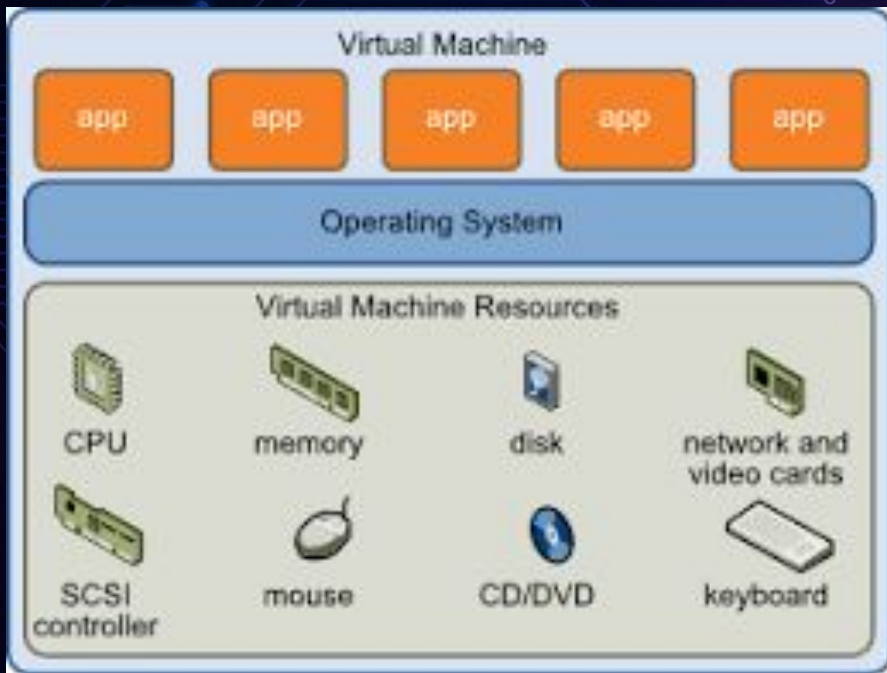
STATISTICS

MACHINE LEARNING



Cloud computing

- **Macchina virtuale: trasformazione di un server fisico in molti server virtuali.**

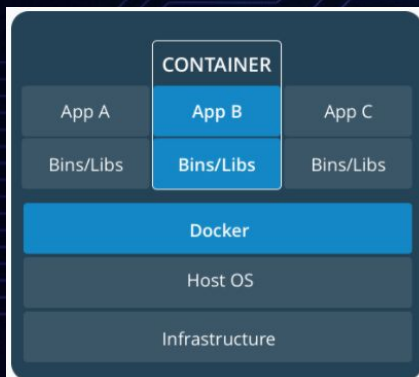


Una macchina virtuale (VM) è un'emulazione software di un computer fisico che esegue un proprio sistema operativo e applicazioni completamente isolati, funzionando esattamente come una macchina fisica separata pur condividendo l'hardware sottostante di un server host.



Cloud computing

- Docker contro macchina virtuale



CONTAINERS

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), and start almost instantly.



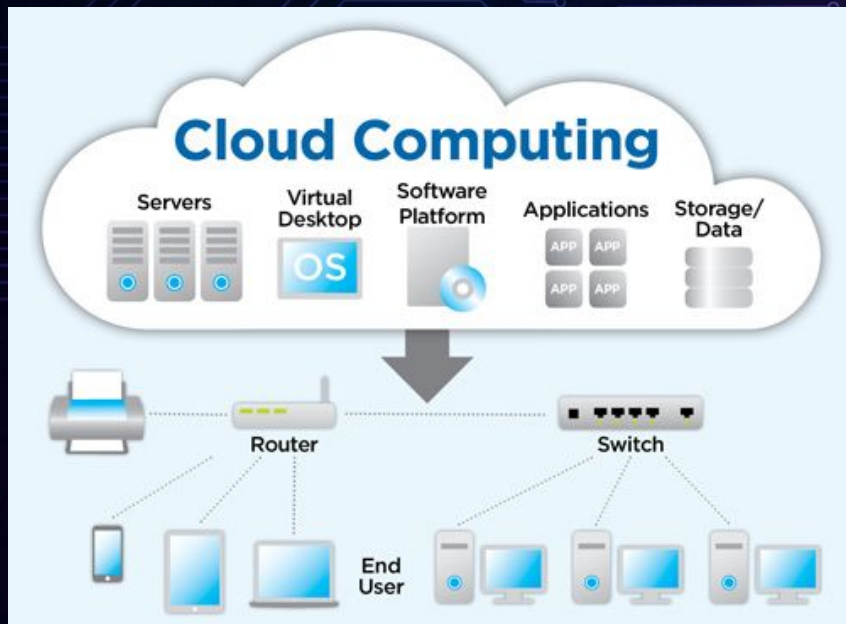
VIRTUAL MACHINES

Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers. The hypervisor allows multiple VMs to run on a single machine. Each VM includes a full copy of an operating system, one or more apps, necessary binaries and libraries - taking up tens of GBs. VMs can also be slow to boot.

Mentre una macchina virtuale (VM) emula un intero sistema informatico con il proprio kernel di sistema operativo complesso, un container Docker racchiude solo l'applicazione e le sue dipendenze, condividendo il kernel del sistema host per un'esecuzione molto

Cloud computing

- Un'infrastruttura virtuale



Il cloud computing è la fornitura su richiesta di servizi informatici, come server, storage, database e software, tramite Internet, consentendo agli utenti di accedere alle risorse tecnologiche come un servizio a consumo, anziché possedere e gestire data center fisici.

