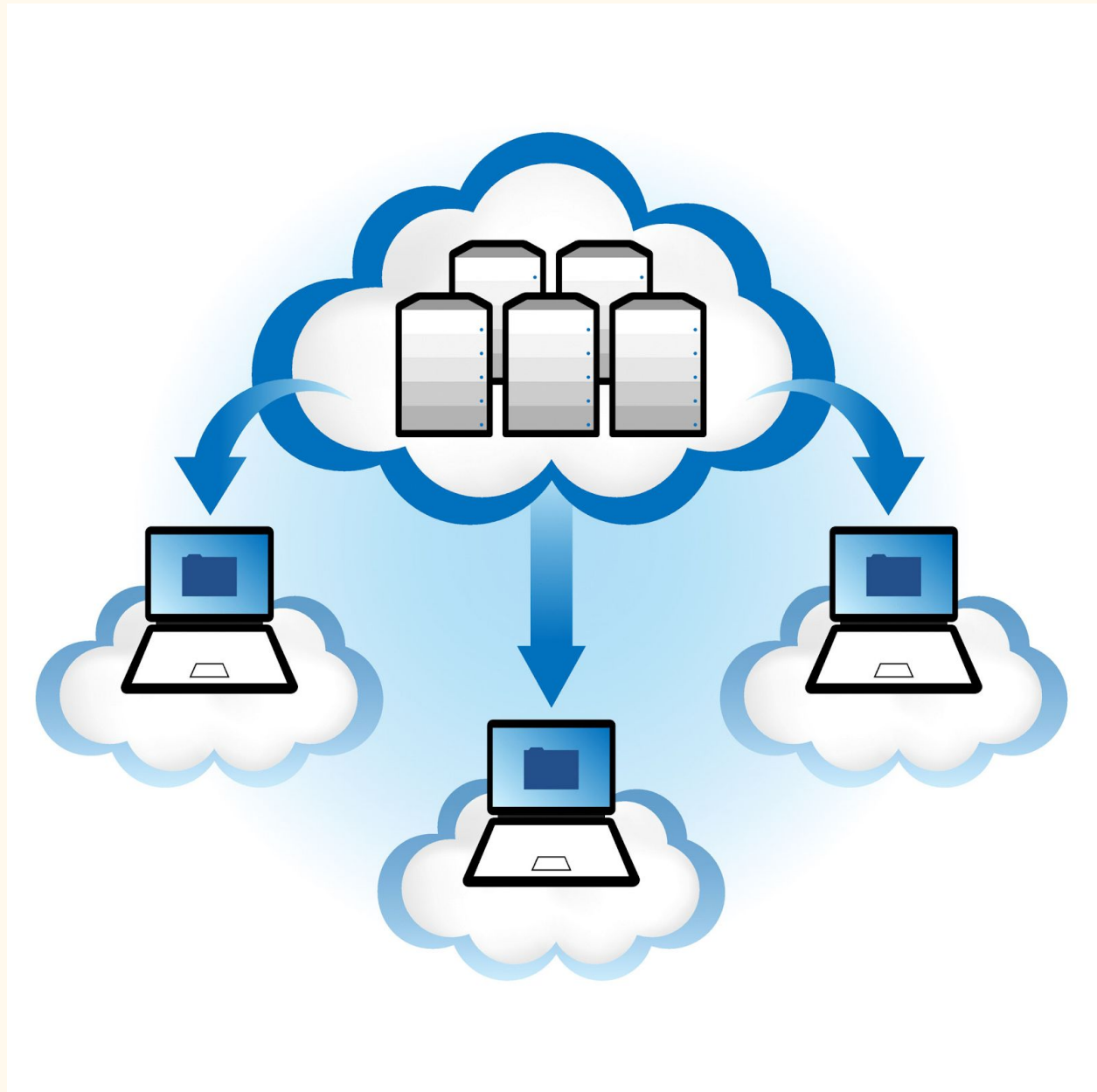


L'architettura Client-Server nell'era degli ZettaByte



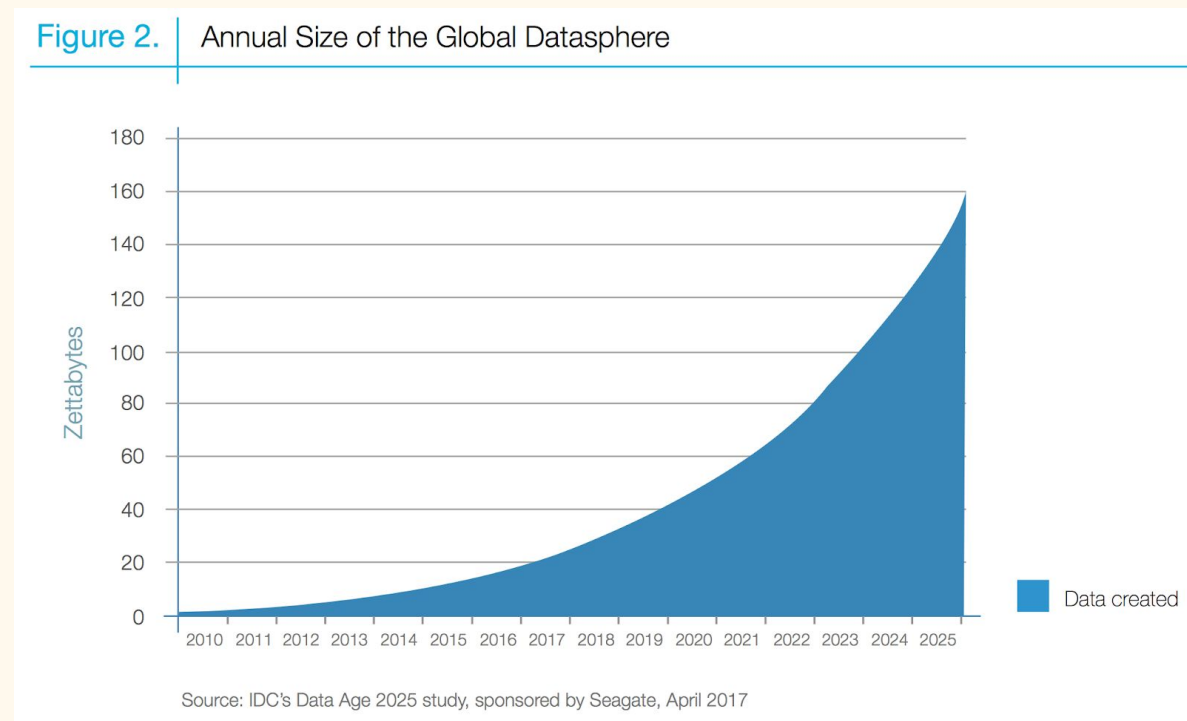
Indice

L'era degli ZettaByte	3
La nascita dell'architettura client-server	4
Significato del termine	4
Esempi di sistemi client/server	5
L'evoluzione dell'architettura a livelli	6
Architettura a un livello - 1 tier	6
Architettura a due livelli - 2 tiers	6
Thin Client	6
Fat Client	7
Architettura a tre livelli - 3 tiers	7
Architettura ad n livelli - n tiers	7
Alternative all'architettura client-server	8
Architettura P2P	8
Architettura Client-Queue-Client	9
Server Crashes	10
Altri esempi	10
Doom	10
Facemash	11
Covid-19 & Click-day	11
Motivi alla base del crash	11
Troubleshooting & Prevenzione	12
Decifrare gli errori	12
Modalità provvisoria	12
Prevenzione	12
Caso di studio: BGA & Lockdown	13
Il cloud e data center	14
Standard per l'infrastruttura dei data center	15
Livelli di certificazione ANSI/TIA-942-ready	15
La virtualizzazione & Docker containers	16
Docker containers	16
Bibliografia e referenze iconografiche	18

L'era degli ZettaByte

5.3 miliardi di utenti connessi a internet, 3.6 dispositivi pro capite e 19.5 **zettabyte (ZB)** di traffico all'anno. [1] Questi erano i dati ipotizzati per il 2020 dallo studio GCI condotto dalla Cisco all'inizio del 2016. Ma l'incremento del mondo IoT, la migrazione dei dati al cloud adottata sempre da più aziende ed infine il lockdown dovuto alla pandemia di Covid-19 hanno cambiato le carte in tavola.

L'IDC (International Data Corporation) ovvero la principale società di informazioni di mercato globale ha rappresentato l'esponentiale aumento del consumo di dati a livello globale.



La stessa società ha poi rivisto le sue previsioni l'8 maggio 2020 a motivo del lockdown e ipotizza che al termine dell'anno saranno stati utilizzati oltre 59 zettabyte (ZB) di dati. [2]

Oggi sarebbe impensabile la vita senza i servizi di streaming, le videochiamate, le applicazioni di messaggistica istantanea, i videogames e molti altri servizi. Qual è la tecnologia alla base della quasi totalità di essi? Come funziona? Che dire dei server crash? Quali sono i progressi degli ultimi anni? E quali sono le tecnologie alla base delle previsioni per il futuro?

ZettaByte (ZB) = 1000^7 Byte = 1000^3 TeraByte

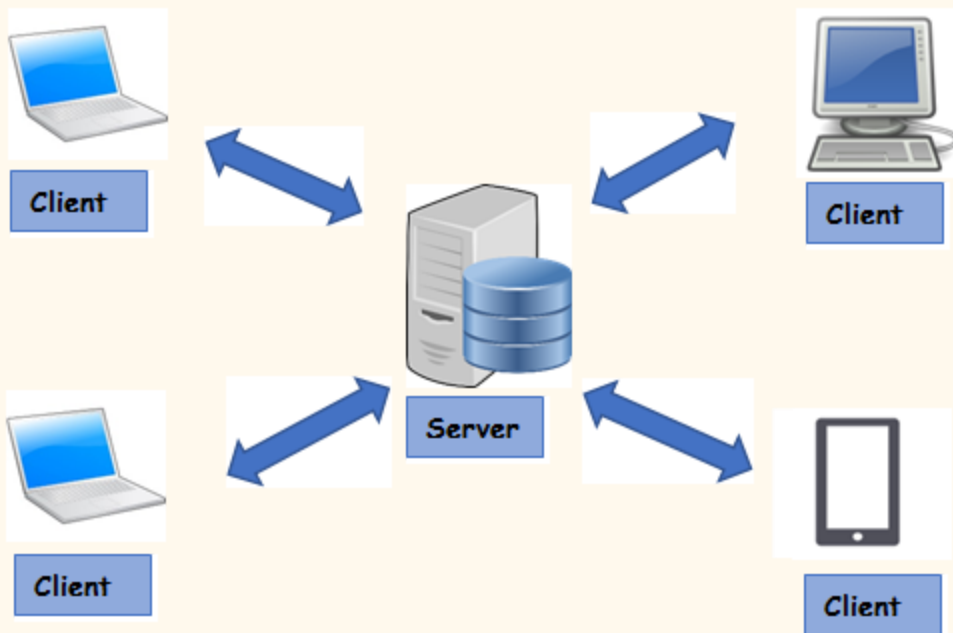
La nascita dell'architettura client-server

La prima forma di architettura client-server venne ideata nel 1964, quando mediante OS/360 di IBM una macchina eseguì la richiesta dell'esecuzione di un lavoro remoto ad un'altra macchina e la risposta fu un output. [3]

In seguito tra gli anni '60 e '70 mentre veniva alla luce **Arpanet**, presso lo SRI (Standard Research Institute International), fu formulato il modello client-server.

Significato del termine

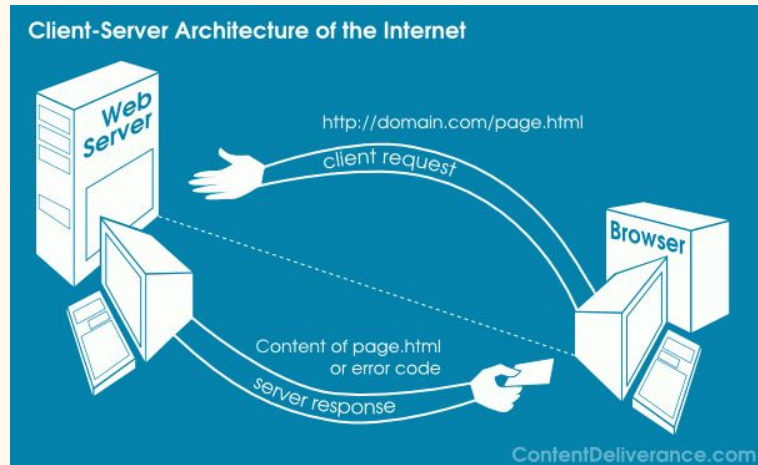
Come indica l'immagine sottostante, l'architettura client-server prevede dei dispositivi “**server**” che mettano a disposizione dei **servizi** all'interno di una rete su determinate porte e dei dispositivi “**client**” che ne usufruiscono.



Esempi di sistemi client/server

- Web server: per la gestione dell'interazione via web tra server e client

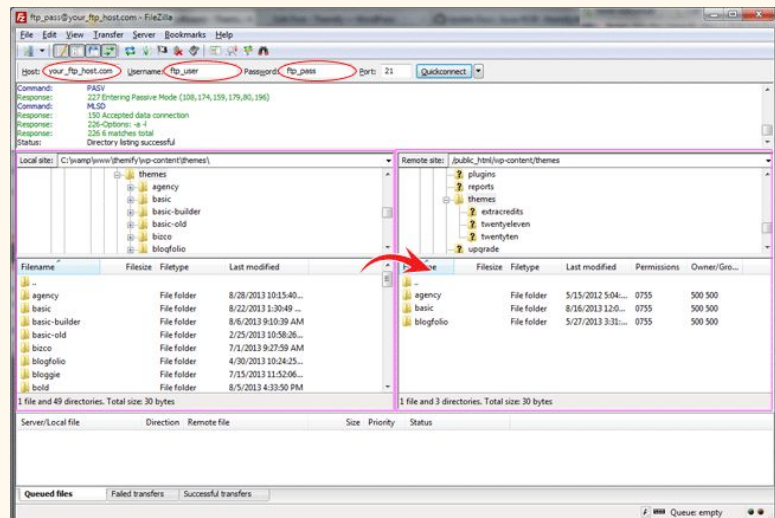
Quando si accede ad una pagina web il browser agisce da client, eseguendo una “**client request**”, mentre il web server risponde mediante una “**server request**”.



Questo è il caso, ad esempio, della richiesta di una pagina “index.html” sulla porta 80 (di default) mediante *Hyper Text Transfer Protocol* e poi della sua ricezione. Le pagine web sono integrate anche da codice Javascript, che permette al client l’esecuzione di azioni indipendenti dal server e codice php, solitamente utilizzato per interfacciarsi ad un Database server

- FTP client: per la gestione dell'upload/download dei file

Un FTP (*File Transfer Protocol*) Server ospita dei file che possono essere caricati o scaricati dall’utente. Basta utilizzare un FTP client come Filezilla indicando l’host, l’username, la password e la porta che di default è la n. 21.



- Database server: per la gestione di grandi moli di dati

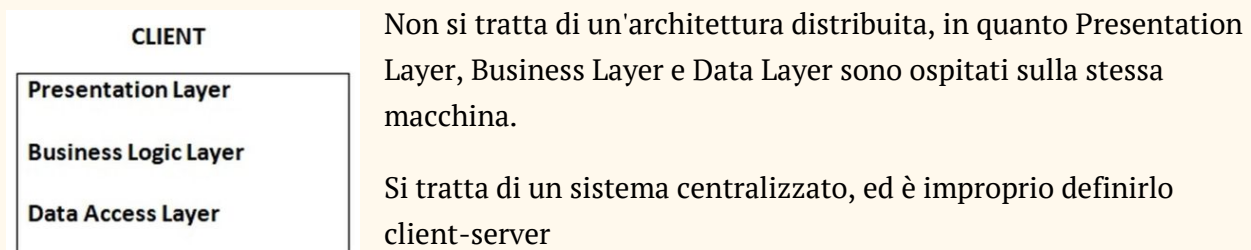
L'evoluzione dell'architettura a livelli

Le architetture client-server sono strutturate su diversi livelli o “tiers” e che le applicazioni distribuite secondo questo paradigma sono caratterizzate da **3 componenti principali** [4 - 5]:

- **Presentation Layer** o **frontend**
- **Business Logic Layer** o **backend**
- **Data Access Layer** per l'accesso ai dati e le risorse

A seconda di come si combinano questi livelli e di quali macchine vengono coinvolte nell'ospitarli, si ottiene un'architettura client-server a 1 o più livelli.

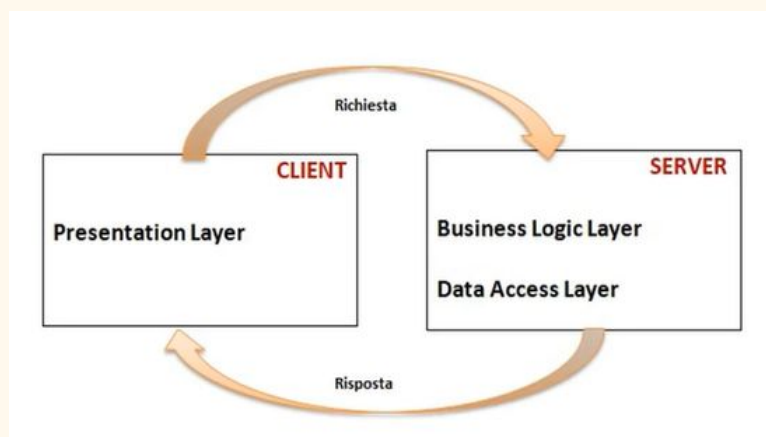
Architettura a un livello - 1 tier



Architettura a due livelli - 2 tiers

Thin Client

Su un client è presente solo il Presentation layer e per questo motivo è detto “Thin Client”.



Vantaggi:

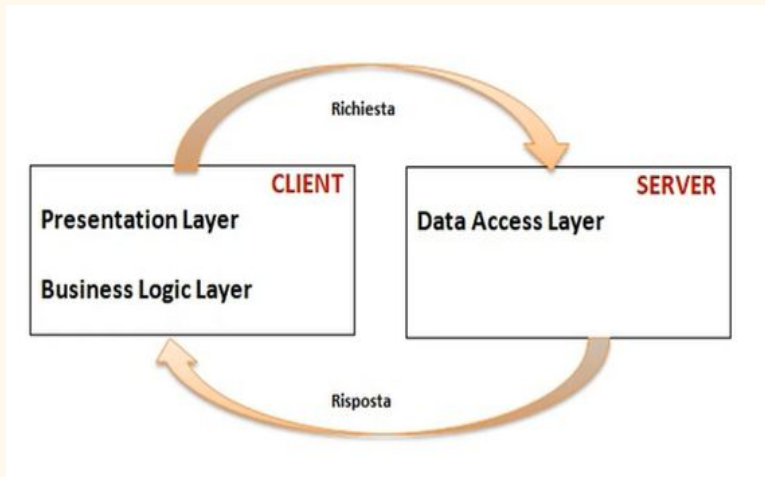
- Adatto ai casi in cui il client ha poca potenza di calcolo

Svantaggi:

- Spreco dell'eventuale potenza di calcolo del client
- Il carico di lavoro è spostato sul server
- Bassa scalabilità

Fat Client

Il client ospita sia il Presentation Layer che il Business Logic Layer, viene dunque definito “Fat Client”.



Vantaggi:

- Il carico di lavoro è distribuito meglio rispetto al Thin Client

Svantaggi:

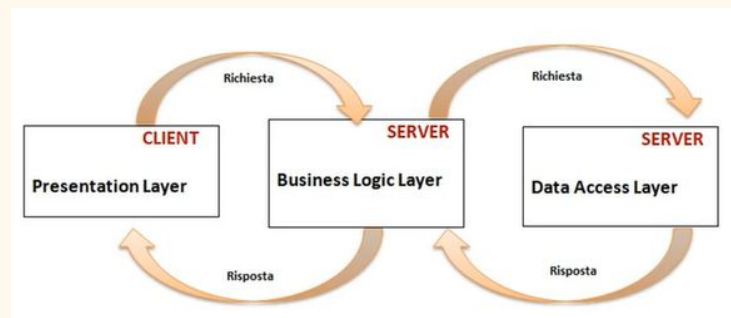
- Nel caso di un update della business logic, si necessita di un aggiornamento su ogni client
- Si genera il “round-trip” per la gestione dei dati ospitati sul server

- Bassa scalabilità

Architettura a tre livelli - 3 tiers

Sul client è presente il Presentation Layer, su un server il Business Logic Layer e su un altro server il Data Access Layer.

Il client sfrutta la potenza del server di backend e questo stesso sfrutta la potenza di elaborazione del server che accede alle risorse.



Vantaggi:

- Buona distribuzione del carico di lavoro
- Buona scalabilità (e possibilità di utilizzo dei cluster)

Svantaggi:

- Maggiore complessità

Architettura ad n livelli - n tiers

Prevede l'introduzione di **server intermedi** che svolgono **compiti specifici**.

Vantaggi:

- Buona distribuzione del carico di lavoro
- Buona scalabilità (e possibilità di utilizzo dei cluster)

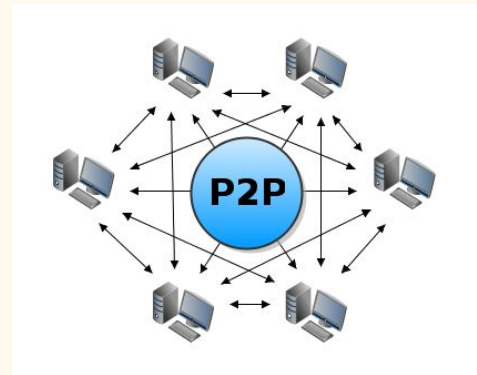
Svantaggi:

- Ulteriore complessità

Alternative all'architettura client-server

Architettura P2P

Tutti i **nodi** di una **rete distribuita** sono **pari** degli altri e svolgono le funzioni sia di client che di server all'evenienza o anche in contemporanea. Reti di questo tipo vengono spesso utilizzate nel contesto del file sharing, come accade per le reti **Torrent**, le blockchain del **Bitcoin** o nel caso di determinati videogiochi, anche se in questo contesto è più difficile ottenere gameplay sincronizzati. [6 - 7 - 8]



Bisogna notare comunque che alcune reti P2P non sono pure e talvolta sfruttano un **Discovery server** per tenere traccia delle macchine attive sulla rete o di un Discovery + **Lookup server** per tenere traccia anche delle risorse messe a disposizione.

Vantaggi:

- Robustezza della rete a livello fisico
- Distribuzione della banda impiegata e maggiore velocità rispetto all'architettura client-server
- Riduzione dei costi di manutenzione
- Abbattimento dei costi di un server

Svantaggi:

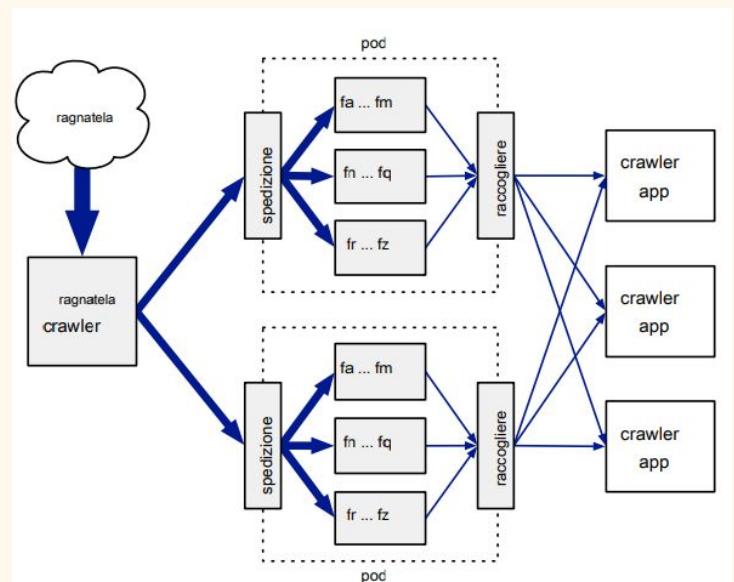
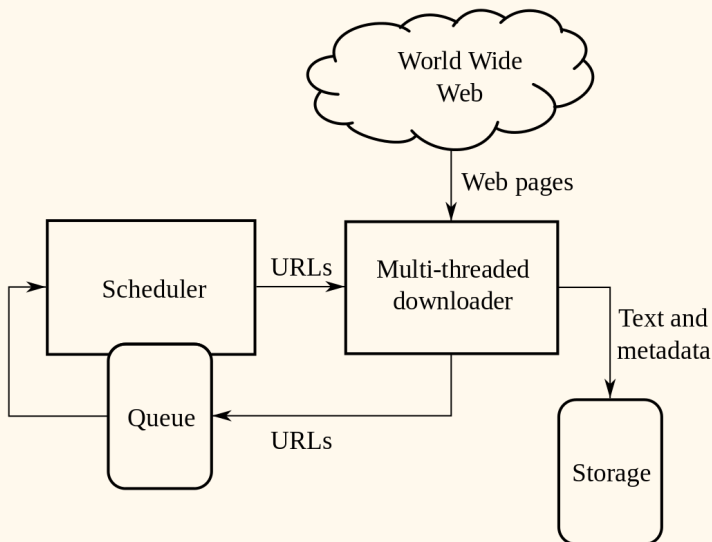
- Complessità nella creazione della rete
- Possibilità di scaricare risorse corrotte (affidabilità)
- Possibilità di riduzione o perdita delle risorse all'interno della rete (distribuzione)

Architettura Client-Queue-Client

Nell'architettura **Client-Queue-Client** tutti gli endpoint si comportano come client. Questa architettura viene definita anche **Passive Queue** perché si tenta di attribuire al **client** la **funzione** di **server** per altri client.

Ad esempio due **Web crawler** [9] operanti su due macchine diverse possono interrogarsi l'un l'altro per comprendere se un determinato URL è indicizzato e noto all'altra macchina.

Web Crawler: a volte detti "spider" o "spiderbot", esplorano il WWW (World Wide Web) per l'indicizzazione dei siti web. Al fine di ottenere risultati ottimali i documenti recuperati vengono analizzati sistematicamente dai **Pod**, ovvero l'insieme di nodi che contiene filtri noti al sistema. (Es.: Il filtraggio per lingua) [10]



Vantaggi:

- Semplifica le implementazioni software ripetute

Svantaggi:

- Utilizzabile soltanto in casi specifici come il Web Crawler
- Obsoleto, sostituito dal P2P

Server Crashes

L'incubo peggiore di ogni amministratore di server è trovarsi di fronte questa **schermata blu** di errore del server:

```
*** STOP: 0x00000019 (0x00000000,0xC00E0FF0,0xFFFFFD4,0xC0000000)
BAD_POOL_HEADER

CPUID: GenuineIntel 5.2.c irq1:1f SYSVER 0xf0000565

Dll Base DateStmp - Name Dll Base DateStmp - Name
80100000 3202c07e - ntoskrnl.exe 80010000 31ee6c52 - hal.dll
80001000 31ed86b4 - atapi.sys 80006000 31ee6c74 - SCSIPORT.SYS
802c6000 31ed06bf - aio78xx.sys 802cd000 31ed237c - Disk.sys
802d1000 31ec6c7a - CLASS2.SYS 8037c000 31eed0a7 - Ntfs.sys
fc698000 31ec6c7d - Floppy.SYS fc6a8000 31ec6ca1 - Cdrom.SYS
fc90a000 31ec6df7 - Fs_Rec.SYS fc909000 31ec6c99 - Null.SYS
fc864000 31ed868b - KSecDD.SYS fc9ca000 31ec6c78 - Beep.SYS
fc6d8000 31ec6c90 - IR942prt.sys fc86c000 31ec6c97 - mouclass.sys
fc874000 31ec6c94 - kbdc1ass.sys fc6f0000 31f50722 - UIDEOPORT.SYS
fffa000 31ec6c62 - mga_mil.sys fc890000 31ec6c6d - vga.sys
fc708000 31ec6ccb - Msfs.SYS fc4b0000 31ec6cc7 - Npfs.SYS
fefbc000 31ecd262 - NDIS.SYS a0000000 31f954f7 - win32k.sys
fefa4000 31f91a51 - mga.dll fec31000 31eedd07 - Fastfat.SYS
feb8c000 31ec6e6c - TDI.SYS feaf0000 31ed0754 - ndf.sys
feacf000 31f130a7 - tcpip.sys feab3000 31f50a65 - netbt.sys
fc550000 31601a30 - el59x.sys fc560000 31f8f864 - afd.sys
fc718000 31ec6e7a - netbios.sys fc858000 31ec6c9b - Parport.sys
fc870000 31ec6c9b - Parallel.SYS fc954000 31ec6c9d - ParUdm.SYS
fc9b0000 31ec6cb1 - Serial.SYS fea4c000 31f5003b - vdr.sys
fea3b000 31f7a1ba - nup.sys fe9da000 32031abe - svx.sys

Address dword dump Build [13811 - Name
fec32484 80143e00 80143e00 80144000 ffdfff00 00070b02 - KSecDD.SYS
801471c8 80144000 80144000 ffdfff00 c0300000 00000001 - ntoskrnl.exe
801471dc 80122000 f0003fa0 f030aee0 e133c4b4 e133cd40 - ntoskrnl.exe
80147304 803023f0 0000023c 00000034 00000000 00000000 - ntoskrnl.exe

Restart and set the recovery options in the system control panel
or the /CRASHDEBUG system start option.
```

A volte i test non bastano, nonostante si siano simulati centinaia o migliaia di connessioni al server, il lancio sul mercato di un nuovo gioco potrebbe intasare i server di gioco, rendendo impossibile l'accesso al videogame come è accaduto nel 2013 al lancio di **SimCity**. [11]

La strategia migliore in questi casi è mantenere i server aggiornati e avere a disposizione 24h al giorno un team tecnicamente preparato, anche se potrebbe essere dispendioso.

Altri esempi

Doom

La prima versione di Doom(famoso gioco multiplayer) fu trasmessa nel server FTP dell'Università di Wisconsin-Madison allo scoccare della mezzanotte del 10 dicembre 1993. In pochi minuti il numero di utenti che aveva eseguito l'accesso al server provocò il crash del sistema. In pratica, le risorse del server e della rete, non erano in grado sostenere un traffico di quella portata.Fu così che i primi ad aver scaricato il gioco, lo copiarono su altri server FTP, condividendolo agli altri utenti. [12]



Facemash



In una notte di ottobre del 2003, Mark Zuckerberg, uno studente di Harvard, creò il predecessore di Facebook, vale a dire Facemash e ne eseguì l'upload sul server dell'università.

In sole 4 ore il sito accolse 450 visitatori e oltre 22.000 click sulle foto, mandando così in sovraccarico il server. [13]

Covid-19 & Click-day

A motivo del lockdown lo Stato italiano decide di rendere disponibile il “Bonus mobilità” sul sito buonomobilita.it e prima di imbattersi nella coda virtuale (per poi essere rigettati da questa) bisogna identificarsi mediante SPID (l'identità digitale) ma il server è in crash a motivo del carico. Come da protocollo, il disservizio che ha scaldato l'animo degli utenti.



Motivi alla base del crash

Sono davvero molti i motivi di un possibile server crash [14], eccone alcuni:

- **Glitch di rete:** Gli utenti non sono in grado di connettersi a motivo di della lenta connettività a Internet
- **Sovraccarico del sistema:** Il server non è più in grado di rispondere a tutte le richieste effettuate a motivo delle risorse limitate per le richieste in corso.
- **Manutenzione del server:** Rende inaccessibili le risorse
- **Errori di configurazione**
- **Malfunzionamento dell'hardware**
- **Backup** che rallentano o bloccano il server
- **Surriscaldamento**
- **Code Breaking:** Aggiornamento del codice
- **Attacchi informatici:** alcuni esempi sono l'attacco **DoS** (Denial-of-Service) in cui una macchina bersagli un determinato server inoltrando un grande quantità di richieste e il **DDoS**, simile al DoS, ma avviene in modo distribuito, eseguito da più macchine malevole [15]

Troubleshooting & Prevenzione

Nel caso del temuto Blue Screen of Death (BSoD) [16 - 17 - 18] è essenziale comprendere esattamente di che errore si tratta e agire di conseguenza. A volte potrebbe trattarsi di un guasto hardware, altre di un driver che ha corrotto il sistema.

Decifrare gli errori

In alcuni casi si trovano informazioni importanti sul “server log”, altre volte bisogna fare riferimento alla BSoD e decifrare gli errori. Segue una lista dei più comuni:

- **KMODE_EXCEPTION_NOT_HANDLED**: Indica un driver è configurato in modo errato.
- **REGISTRY_ERROR**: Problema nel registro.
- **INACCESSIBLE_BOOT_DEVICE**: Il sistema operativo non è in grado di leggere dal disco rigido.
- **UNEXPECTED_KERNEL_MODE_TRAP**: Problemi di memoria.
- **BAD_POOL_HEADER**: Il problema ha qualcosa a che fare con un recente cambiamento nel sistema.
- **KERNEL_DATA_INPAGE_ERROR**: Il sistema operativo non è in grado di leggere una pagina di dati del kernel dal file di paging.
- **NMI_HARDWARE_FAILURE**: Incapacità del livello di astrazione dell'hardware di identificare la causa dell'errore.
- **NTFS_FILE_SYSTEM**: Disco rigido danneggiato.

Modalità provvisoria

Se è possibile avviare la modalità provvisoria probabilmente non si tratta di un problema hardware, ma di un driver che causa il malfunzionamento.

Un modo in cui si può operare è riavviare il server riavviando un driver alla volta in modo da comprendere quale è quello che ha causato l'errore.

A seconda dell'errore rilevato si dovranno eseguire tutte le azioni atte a rendere operativo il sistema al più presto.

Prevenzione

Uno dei metodi migliori per garantire l'affidabilità e il ripristino del sistema in breve tempo è la **schedulazione** di **backup** eseguiti ad **intervalli regolari**, meglio se in remoto.

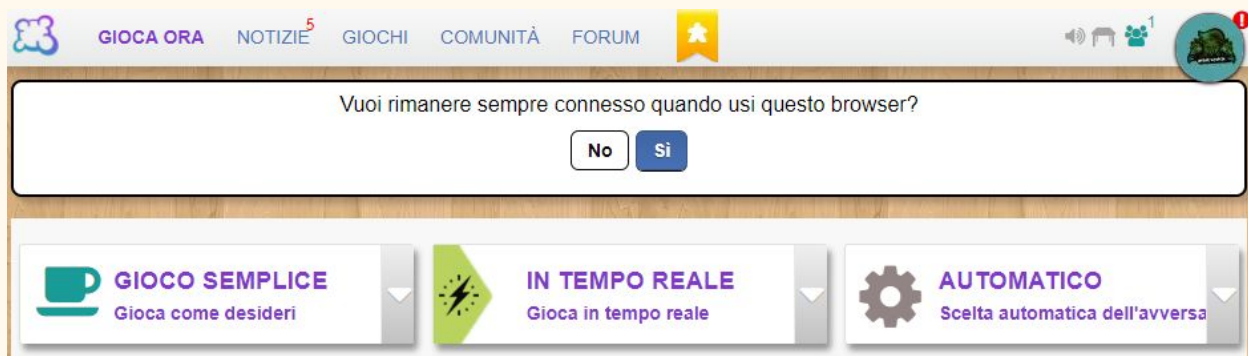
Un altro indicatore importante è la **temperatura** che deve essere compresa tra i **59 e i 20 °C**.

Un'altra buona norma è eseguire degli **stress test**. Ad esempio, nel caso di un server che ospita delle partite in multiplayer si disporrà la creazione di utenti virtuali.

Avere a disposizione un **team di tecnici specializzati** permette il monitoraggio, la rilevazione dell'abbassamento delle prestazioni dei server e garantendo interventi tempestivi in caso di server crash.

Caso di studio: BGA & Lockdown

La piattaforma Board Game Arena, proprio come molte altre, ha subito un grande incremento dei giocatori a motivo del Lockdown dovuto alla pandemia di Covid-19, perciò ha attuato delle strategie atte a mantenere i server attivi e garantendo costantemente il servizio agli iscritti.



Il 26 marzo 2020, gli amministratori hanno reso pubbliche le azioni volte dal team di sviluppo [19]. Inizialmente il limite di giocatori online è stato portato da 5000 a 9000. Tuttavia, notando che non tutte le richieste erano soddisfatte, nel tempo si è arrivati ad ospitare contemporaneamente oltre 30.000 giocatori. Il team spiega che farlo non è stato affatto semplice, in quanto è stato necessario rendere **“distribuita”** la parte del sistema che gestiva la lobby (l'arena che organizza gli incontri tra gli sfidanti), che inizialmente era centralizzata. Certo, era nei programmi, ma le necessità hanno reso necessario anticipare la modifica del software rendendo i DB della lobby distribuiti. Un grande plauso a BGA!

Il cloud e data center

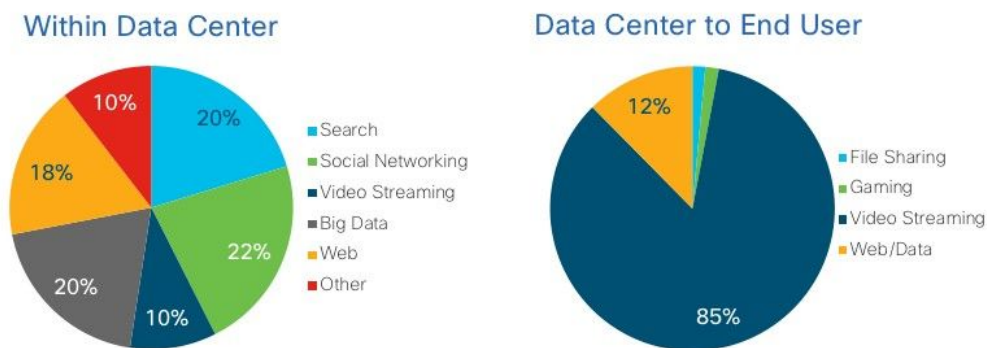
Cloud: Non è un'entità fisica, ma indica la rete globale dei server interconnessi tra loro e che operano come un unico ecosistema. Possono gestire e archiviare dati, eseguire applicazioni o distribuire contenuti e servizi. [20]

Data center: Un edificio attrezzato per ospitare insieme di macchine, di server, di sistemi di storage, di apparati di trasmissione (come router, switch, fibre ottiche), di batterie e sistemi di condizionamento.

Si possono definire anche "Cuore pulsante delle aziende", in cui avvengono le comunicazioni, i processi e i servizi a supporto di qualsiasi impresa che li utilizzi [21 - 22]

Si può affermare che la rete mondiale dei data center costituisce il cloud stesso, e come è stato spiegato inizialmente, l'uomo sta vivendo nell'era degli ZettaByte e si prevede che il flusso in termini di dati trasmessi mediante Internet aumenterà sempre di più.

Data Center Traffic by Application—2021



Big Data is the fastest growing application within the data center, from 12% in 2016 to 20% in 2021
Video is only 10% within the data centers but is 85% of data center to end-user

Source: Cisco Global Cloud Index, 2016-2021

Dal grafico (relativo allo studio GCI condotto dalla Cisco [1]) si comprende che i data center gestiscono la totalità dei servizi in cloud, e benché 85% di dati richiesti "dall'end user" siano Video streaming, essi rappresentano solamente il 10% del traffico nei data center.

Il 18% riguarda il Web e il 20% l'elaborazione e l'archiviazione dei Big Data. Due tipologie di dati alla base di ogni azienda informatizzata.

Standard per l'infrastruttura dei data center

Certamente anche la progettazione dei data center è stata standardizzata. Lo standard più accettato è l'ANSI/TIA-942 che include lo standard per la certificazione **ANSI/TIA-942-ready**.

Esso assicura che un data center sia conforme a uno dei 4 livelli possibili.

Livelli di certificazione ANSI/TIA-942-ready

- **Livello 1:**

Offre una protezione limitata rispetto agli eventi fisici e un percorso di distribuzione unico e **non ridondante**.

- **Livello 2**

Offre una migliore posizione rispetto agli eventi fisici e presenta componenti dalla **capacità ridondante**.

- **Livello 3**

Offre **protezione da quasi tutti gli eventi fisici**, fornisce capacità ridondanti e percorsi di distribuzione indipendenti multipli. I componenti possono essere rimossi e sostituiti **senza** che avvenga l'**interruzione** dei **servizi**.

- **Livello 4**

Offre la tolleranza ai guasti, e i **livelli più elevati di ridondanza**. Essi permettono la manutenzione in qualsiasi punto senza che si causi **mai** l'**interruzione** dei **servizi**.

La virtualizzazione & Docker containers

Ma esattamente che tipo di software viene eseguito nei data center? E quali tecnologie rappresentano il futuro in questo ambito?

In genere si tratta di macchine virtualizzate, ed è sempre più comune per le aziende adottare tecnologie quali i Docker containers.

Virtualizzazione: È l'astrazione di risorse fisiche dedicate, che possono essere assegnate a compiti specifici. Vengono virtualizzati l'archiviazione, l'applicazione, il desktop, il server e la rete [23]

Vantaggi e Svantaggi

La **virtualizzazione** viene eseguita direttamente sul server e questo comporta dei grandi vantaggi alle aziende, quali:

- Abbassamento costi
- Meno manodopera
- Maggiore affidabilità
- Maggiore sicurezza
- Meno server (le risorse vengono ottimizzate rispetto agli utilizzi)

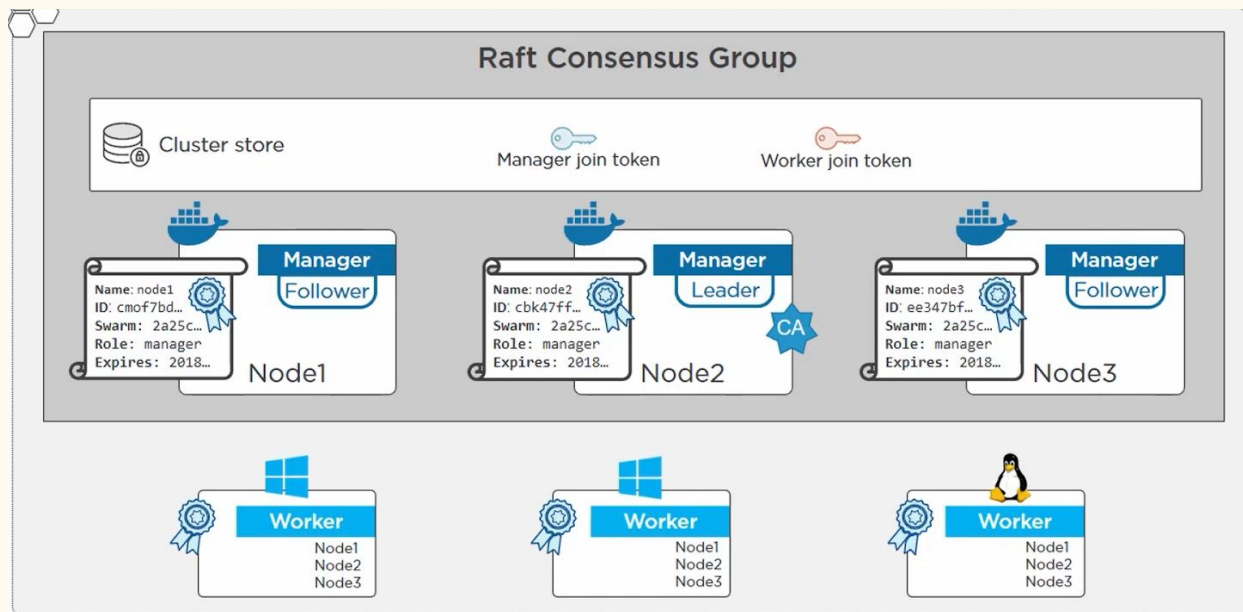
Gli **aspetti negativi** sono:

- la necessità di **tecnici specializzati** per il loro utilizzo
- Ogni VM innestata sul data center **consuma risorse** quali CPU, RAM e memoria (vedere la differenza con i containers)

Docker containers

Partendo dal presupposto che se le applicazioni si interrompono e non garantiscono il servizio agli utenti, l'azienda che gestisce l'applicazione ne risente e potrebbe persino rischiare il fallimento, ogni azienda valuta le migliori strategie per rispondere alla domanda di un determinato mercato.

In passato le aziende investivano nell'acquisto di nuovi server, uno per ogni applicazione. Questo però comportava a volte l'utilizzo effettivo di soltanto il 5-10% delle potenziali capacità. [24] Con l'avvento delle VM e dei Docker containers, le carte in tavola sono cambiate!



I containers permettono di eseguire copie di applicazioni sullo stesso sistema operativo, andando a liberare grandi quantità di risorse come la CPU, la RAM e l'archiviazione. Vengono anche ridotti i costi del sistema e di manutenzione. Questi non sono gli unici vantaggi: Docker può essere orchestrato, ovvero nel caso in cui un container sia malfunzionante, esso viene sostituito o comunque il suo carico di lavoro viene assunto da un altro container in modo autonomo senza l'intervento da parte dei programmatori.

Vantaggi:

- Si risparmiano risorse
- Può essere orchestrato
- I malfunzionamenti sono autogestiti
- Permette il controllo e il monitoraggio delle risorse utilizzate

Svantaggi:

- Necessità di specializzazione

Bibliografia e referenze iconografiche

[Immagine iniziale] <https://forthesakeoftechnology.wordpress.com>

[1] <https://blogs.cisco.com/news/acceleration-of-multicloud-era>

[2] <https://www.idc.com/getdoc.jsp?containerId=prUS46286020>

[3] https://en.wikipedia.org/wiki/Client%E2%80%93server_model

[immagine client-server] <https://www.omnisci.com/technical-glossary/client-server>

[immagine web server-browser] <https://contentdeliverance.com/client-server-architecture/>

[immagine ftp client] <https://themify.me/docs/ftp>

[4] shorturl.at/eoxP1

[5] <https://www.youtube.com/watch?v=V3MnaSf7w2k>

[6] <https://it.wikipedia.org/wiki/Peer-to-peer>

[immagine P2P] shorturl.at/jEHMU

[7] shorturl.at/gilXY

[8] https://en.wikipedia.org/wiki/Game_server

[9] https://en.wikipedia.org/wiki/Web_crawler

[10] https://www.usenix.org/legacy/events/nsdi10/tech/full_papers/hsieh.pdf

[immagine schermata errore server]

<https://www.codementor.io/learn-programming/server-down-what-to-do-when-my-server-crashes>

[11]

<https://www.theguardian.com/technology/2017/jul/04/server-crashes-patches-dlc-video-game-irritations-explained>

[12] <https://it.wikipedia.org/wiki/Doom#Storia>

- [13] <https://www.fastweb.it/social/la-storia-di-facebook/>
[immagine facemash] <https://thesocialnetwork.fandom.com/wiki/Facemash>
- [14] <https://www.cioreview.com/news/most-common-causes-behind-server-crashes-nid-24580-cid-130.html>
- [15] <https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos#:~:text=A%20Denial%2Dof%2DService%20>
- [16] <https://www.insightsforprofessionals.com/it/data-center/how-diagnose-unexpected-server-crash>
- [17] <https://www.codementor.io/learn-programming/server-down-what-to-do-when-my-server-crashes>
- [18] https://en.wikipedia.org/wiki/Server_log
- [19] <https://forum.boardgamearena.com/viewtopic.php?f=5&t=14646>
[immagine BGA] <https://boardgamearena.com/lobby>
- [20] <https://azure.microsoft.com/it-it/overview/what-is-the-cloud/>
- [21] https://www.youtube.com/watch?v=55_yRjSFxPk
- [22] <https://www.iotedge.it/data-center/data-center-cosa-sono-e-come-funzionano/>
- [23] <https://www.cisco.com/c/en/us/solutions/enterprise-networks/what-is-virtualization.html>
- [24] <https://nigelpoulton.com/books>