POLITECNICO DI TORINO

Corso di Laurea:

Ingegneria della Produzione Industriale e dell'Innovazione Tecnologica

Tesi di Laurea Magistrale A.A. 2019-2020

"Il cloud computing: analisi del panorama competitivo odierno, architetture e servizi cloud, strategie di vendita"

Relatori:

Prof. Guido Perboli

Prof.ssa Mariangela Rosano

Candidato:

Lorenzo Rapetti

Luglio 2020

Sommario

L'industria del *cloud computing* sta portando sconvolgenti trasformazioni alle aziende di qualsiasi dimensione e di ogni settore. Lo scopo di questa trattazione è illustrare questo paradigma informatico, spiegando le ragioni del suo successo in termini tecnici e strategici.

L'analisi inizierà descrivendo a grandi linee il settore del cloud computing, la sua storia ed evoluzione.

Successivamente si andranno a dettagliare i lati più tecnici dei moderni servizi ed architetture di cloud computing, in modo da giungere ad una comprensione esaustiva delle motivazioni alla base della notevole popolarità, tra aziende di ogni dimensione, della vasta gamma di soluzioni cloud.

Si passerà poi ad un approfondimento delle principali strategie odierne di vendita e di marketing di servizi cloud a clienti di livello *enterprise*.

Si procederà, infine, ad una descrizione del tirocinio effettuato presso la sede di Google Cloud (uno dei più importanti fornitori di servizi di cloud computing a livello mondiale) a Dublino, spiegando le principali attività svolte, i progetti di stage, i risultati ottenuti ed i concetti appresi.

Introduzione

Rivoluzioni industriali

Secondo Klaus Schwab, fondatore e presidente del World Economic Forum, l'umanità sta entrando in una fase di rivoluzione industriale e tecnologica senza precedenti.

Ripercorrendo la storia, la prima rivoluzione industriale ha trasformato la società con l'introduzione della produzione meccanizzata; la seconda rivoluzione industriale è stata portata dall'avvento dell'elettricità, che ha permesso la produzione di massa; la terza rivoluzione industriale è stata caratterizzata dall'informatica, che ha permesso l'automazione della produzione.

La rivoluzione odierna, classificabile quindi come quarta rivoluzione industriale, pone le proprie basi sulla terza rivoluzione, rappresentandone però non una mera propaggine ma un completo superamento: si passa infatti dalla semplice digitalizzazione all'innovazione basata sulla fusione di più tecnologie, che raggiunge velocità ed impatti sulla popolazione mai visti prima e forza aziende di ogni tipo a rivalutare le proprie strategie; basti pensare all'evoluzione di tecnologie come il *machine learning*, una branca dell'intelligenza artificiale (spesso abbreviata in *IA*) che solo di recente viene utilizzata commercialmente ma che è già riuscita, in un lasso di tempo pressoché insignificante considerando la storia dell'umanità, a portare a veri e propri sconvolgimenti fantascientifici come i veicoli a guida autonoma.

La cosiddetta "legge di Moore" postula che la complessità dei microprocessori usati nei computer (e quindi la potenza di calcolo) raddoppi ogni 18 mesi (Dally, 2010). Sebbene sia una semplificazione della realtà, l'enunciato empirico di Gordon Moore, cofondatore di Intel, è stato per oltre 40 anni considerabile come veritiero. Al giorno d'oggi, nuovi design di microprocessori stanno portando a progressi ancora più marcati nella potenza di calcolo dei computer: si può pensare, ad esempio, a processori progettati per la massima efficacia nei programmi ed applicazioni di machine learning, come le *TPU*, o *Tensor Processing Units*, chip sviluppati da Google ed ottimizzati per calcoli matriciali, più veloci ed efficienti nel loro ambito rispetto a processori tradizionalmente utilizzati come le *GPU* (MLPerf, 2019).

Continuano inoltre senza sosta notevoli progressi nel campo del cosiddetto “quantum computing”, una branca di calcolatori che rappresentano e processano informazioni utilizzando bit quantistici piuttosto che i tradizionali bit. Google ha raggiunto nel 2019 la cosiddetta “supremazia quantistica”, ovvero la risoluzione, tramite un computer quantistico, di un calcolo talmente complesso che sarebbe totalmente impossibile da realizzare tramite un *supercomputer* tradizionale in un lasso di tempo ragionevole. Sebbene non ci sia ragione di pensare che i computer quantistici, a causa delle loro limitazioni intrinseche, potranno completamente soppiantare i computer tradizionali, applicazioni specifiche del quantum computing potrebbero portare a benefici e prestazioni considerabili come fantascienza fino a pochi anni fa.

Un altro tassello rilevante risiede nella sempre crescente importanza dei dati nelle strategie aziendali, nello sviluppo dei prodotti ed in generale nel *decision-making* all’interno di realtà operanti nei settori più disparati. Si pensi al cosiddetto *Internet-Of-Things* (*IoT)*, paradigma informatico che prevede l’integrazione di svariate tipologie di dispositivi ed oggetti con i calcolatori ed i sistemi informativi. Nell’ambito industriale, l’IoT rientra nelle caratteristiche della cosiddetta “industria 4.0” e permette il costante monitoraggio ed il controllo capillare di singoli dispositivi ed apparecchiature nelle fabbriche aziendali.

Queste nuove tecnologie portano con sé, tuttavia, dei costi proibitivi per la grande maggioranza delle imprese: dal dispendio energetico necessario per il raffreddamento dei processori per il machine learning all’infrastruttura richiesta per la gestione e l’immagazzinamento costante di petabyte di dati provenienti da milioni di apparecchiature IoT. Come accade sempre di più nell’evoluzione tecnologica, la specializzazione e l’economia di scala diventano concetti quasi imprescindibili per garantire un servizio efficiente ed affidabile.

Al riguardo è emblematica l’elettricità: durante i primi anni successivi a tale scoperta scientifica, molti utenti facevano uso di generatori situati in prossimità delle proprie abitazioni. Con il passare del tempo, e con la conseguente maturazione dell’industria elettrica ed energetica, si è arrivati alla costruzione di centrali e reti elettriche capaci di

servire milioni di persone. L'elettricità è diventata così un servizio, una spesa *operativa* piuttosto che un investimento.

Lo stesso principio dell'elettricità, lo stesso cambiamento di prospettiva, sta avvenendo oggi per quanto riguarda i calcolatori e le tecnologie protagoniste della quarta rivoluzione industriale: sempre più aziende decidono di rivolgersi a gestori specializzati nell'offrire servizi su larga scala, ricavando benefici sia economici che strategici. In generale, l'outsourcing di tali servizi informatici permette alle aziende di concentrarsi esclusivamente sul proprio *core business*, riducendo spese e complessità operative derivanti dall'acquisizione e manutenzione di *data center* e *server farm* proprie.

Cenni sul *cloud computing*

Il cloud computing può essere definito come un paradigma di servizi informatici avente le seguenti caratteristiche:

- ➔ offerta di accesso on-demand e self-service a una serie di risorse informatiche. E' sufficiente usare una semplice interfaccia Internet per ottenere la potenza di elaborazione, lo spazio di archiviazione o le reti di cui si necessita, senza bisogno di intervento umano da parte del fornitore;
- ➔ accesso a tali risorse informatiche attraverso Internet, da qualsiasi luogo dotato di connessione;
- ➔ possesso di quantità considerevoli di risorse informatiche da parte del fornitore, che le assegna dinamicamente ai propri clienti. Ciò consente al fornitore di ottenere economie di scala acquistando queste risorse all'ingrosso, trasferendo poi i derivanti risparmi ai clienti;
- ➔ elasticità delle risorse informatiche. Se sorge la necessità di avere accesso a maggiori risorse, il cliente ne può rapidamente ottenere di più e viceversa, in caso una parte di queste non siano più ritenute necessarie;
- ➔ pagamento in base all'uso delle risorse. I clienti pagano (salvo particolari eccezioni) solo per ciò che effettivamente usano o prenotano. Se smettono di usare le risorse, smettono di pagare.

(Rice, 2020)

Prima dell'avvento del cloud computing, le aziende di qualsiasi dimensione erano costrette a percorrere principalmente due strade: acquistare intere infrastrutture dedicate ai servizi informatici nonché organizzare un luogo fisico dove conservarle in sicurezza (luogo che era spesso una sala server presso la sede dell'azienda stessa, da cui deriva il termine *on-premises*, abbreviato spesso in *on-prem*, che tradotto dall'inglese significa "presso la sede"), oppure affittare gli spazi fisici offerti dai proprietari di grandi data center per stoccare e mantenere attive le proprie infrastrutture, un concetto noto come *co-location* o *housing*.

In entrambi i casi sorgevano grandi inefficienze in termini di utilizzo delle risorse, poiché le aziende dovevano preventivare di quante risorse informatiche avrebbero avuto bisogno ben prima dell'effettiva implementazione di qualunque sistema software.

Esistono varie categorie di servizi offerti dai fornitori cloud, che saranno approfondite nel corso della trattazione. Tra le principali troviamo:

- ➔ *IaaS* o "Infrastructure as a Service": come si evince dal nome, la categoria IaaS offre all'utente la possibilità di affittare infrastrutture informatiche (tra le quali le macchine virtuali, il cui scopo e funzionamento saranno trattati in seguito) in modo semplice e veloce; starà poi all'utente configurarle ed utilizzarle a proprio vantaggio, senza però la necessità di preoccuparsi di effettuare manutenzione sull'hardware e di acquisire ed allocare spazi fisici atti ad immagazzinare e mettere in sicurezza tale hardware;
- ➔ *PaaS* o "Platform as a Service": i servizi di questa categoria "astraggono" tutte le componenti relative all'infrastruttura e alla gestione dei server, dei sistemi operativi, delle librerie software...In questo modo, il cliente ha l'opportunità di concentrarsi solamente sul codice e sulla logica della propria applicazione, semplificando notevolmente lo sviluppo ad essa relativo;
- ➔ *SaaS* o "Software as a Service": questi servizi sono offerti come software, ovvero applicazioni web pronte per essere utilizzate dall'utente finale senza alcun tipo di configurazione. A questa categoria appartengono, ad esempio, il servizio di videoconferenze Zoom, i pacchetti di produttività (editor di testo, fogli di calcolo, designer di presentazioni) inclusi in *G Suite* di Google e in *Office 365* di Microsoft e la piattaforma Shopify per la creazione e la gestione di siti di ecommerce.

Alphabet, Google e Google Cloud

Cenni sull'azienda

L'attuale logo della società

Google nacque come motore di ricerca per siti Internet nel 1996 dalle menti di Larry Page e Sergey Brin, due dottorandi in informatica all'università di Stanford; all'inizio, il motore di ricerca era chiamato BackRub, e venne rinominato nel 1997 in "Google" da una storpiatura del termine "Googol", che rappresenta il numero 10^{100}, ovvero il numero 1 seguito da 100 numeri zero, ad indicazione della volontà dei fondatori di organizzare l'incredibile vastità di informazioni presenti sul web (l'attuale *mission* aziendale di Google è, infatti, "organizzare le informazioni a livello mondiale e renderle universalmente accessibili e utili").

I cofondatori di Google nel loro primo ufficio, un garage nella periferia di Menlo Park (Google, 2020)

L'azienda venne ufficialmente fondata in California nel settembre del 1998, e diventò una società quotata in borsa nell'agosto del 2004. Nel 2015, in seguito ad una riorganizzazione

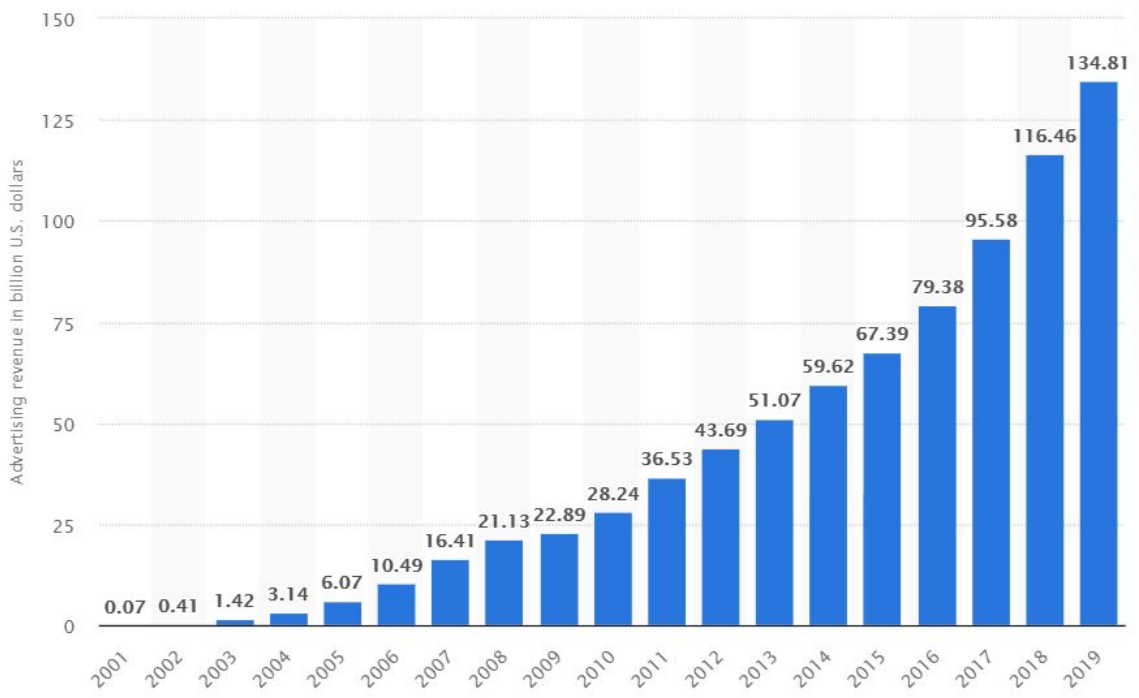
strategica, Google è diventata la principale sussidiaria di Alphabet, una holding company creata appositamente per separare e disaccoppiare le iniziative imprenditoriali più stabili e redditizie della creatura di Page e Brin dalle "scommesse" più rischiose ed innovative come Wing, un servizio di consegna tramite droni, o Calico, un laboratorio di ricerca e sviluppo concentrato sulla scoperta di innovazioni nel campo della longevità umana.

L'attuale logo della holding company Alphabet

Al giorno d'oggi Google offre, oltre ovviamente al motore di ricerca *Google Search* (il sito web più visitato al mondo secondo la piattaforma online di analisi Alexa, sussidiaria di Amazon), una grande varietà di prodotti e servizi, rivolti sia ad aziende sia a consumatori, tra cui:

- ➔ Google Chrome, il browser web più usato al mondo (Liu, 2020);
- ➔ Android, il sistema operativo per dispositivi mobili più usato al mondo (O'Dea, 2020);
- ➔ Google Translate, un servizio di traduzione automatica di contenuti testuali e visivi;
- ➔ YouTube, una piattaforma web di condivisione di video;
- ➔ Gmail, un popolare servizio di posta elettronica;
- ➔ Google Assistant, un assistente virtuale sviluppato tramite intelligenza artificiale;
- ➔ Dispositivi hardware come lo smartphone Pixel;
- ➔ Applicazioni web di produttività come l'editor di testo Google Docs e i fogli di calcolo Google Sheets;
- ➔ Vari servizi di cloud computing per le aziende, tra cui IaaS e PaaS, offerti tramite la suite chiamata Google Cloud Platform.

Il modello di business di Google è principalmente basato sulla pubblicità online: molti dei servizi offerti ai consumatori sono infatti gratuiti e supportati dalla pubblicità. Nel 2019, i ricavi provenienti dagli annunci ammontavano a circa 135 miliardi di dollari, ovvero quasi l'84% dei ricavi totali di Google.



Crescita dei ricavi pubblicitari annuali di Google (Clement, 2020)

La società si serve di una piattaforma software di annunci pubblicitari online denominata Google Ads, tramite cui i clienti pagano per poter mostrare i propri messaggi promozionali entro una o più proprietà di Google, come gli spazi pubblicitari in cima alle pagine dei risultati di ricerca di Google Search. L'intero processo è pressoché automatizzato e sfrutta il modello di prezzo *pay-per-click* o *PPC*, secondo cui i clienti pagano per le proprie inserzioni solamente nel momento in cui qualcuno fa clic su di esse, manifestando quindi interesse.

E' importante notare come, anche considerando l'apporto della holding Alphabet, i ricavi provenienti dagli annunci pubblicitari rappresentino ancora la netta maggioranza degli introiti totali, surclassando le entrate dei servizi di cloud computing e le cosiddette *Other Bets*, ovvero le "altre scommesse", le idee audaci ed innovative su cui sta puntando Alphabet.

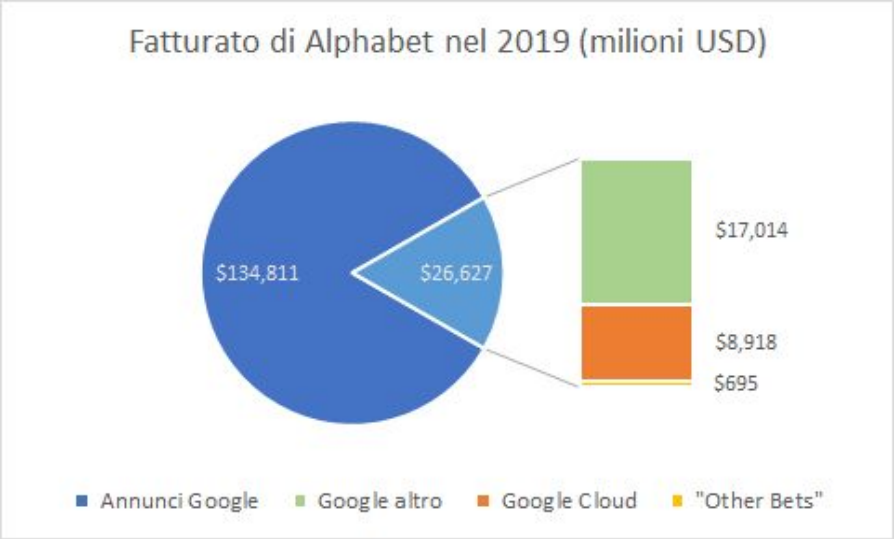


Grafico della suddivisione delle fonti di ricavo di Alphabet (dati Statista, 2020)

La diversificazione messa in atto dalla società rappresenta, tuttavia, una strategia fondamentale per continuare a perseguire l'innovazione tecnologica e rimanere in un approccio "startup" di sfida costante allo status quo.

Google Cloud

Il primo servizio cloud offerto da Google è stato App Engine (Meier, 2017), un servizio PaaS lanciato in anteprima per alcuni sviluppatori nell'Aprile del 2008 e rilasciato ufficialmente al pubblico nel 2011. In contrasto AWS, la piattaforma di servizi cloud creata da Amazon e considerata pioniera per quanto riguarda il cloud computing moderno, fu lanciata ufficialmente cinque anni prima, nel 2006.

Attualmente tutti i servizi cloud offerti da Google per le aziende sono raccolti sotto l'egida di un unico brand: Google Cloud.

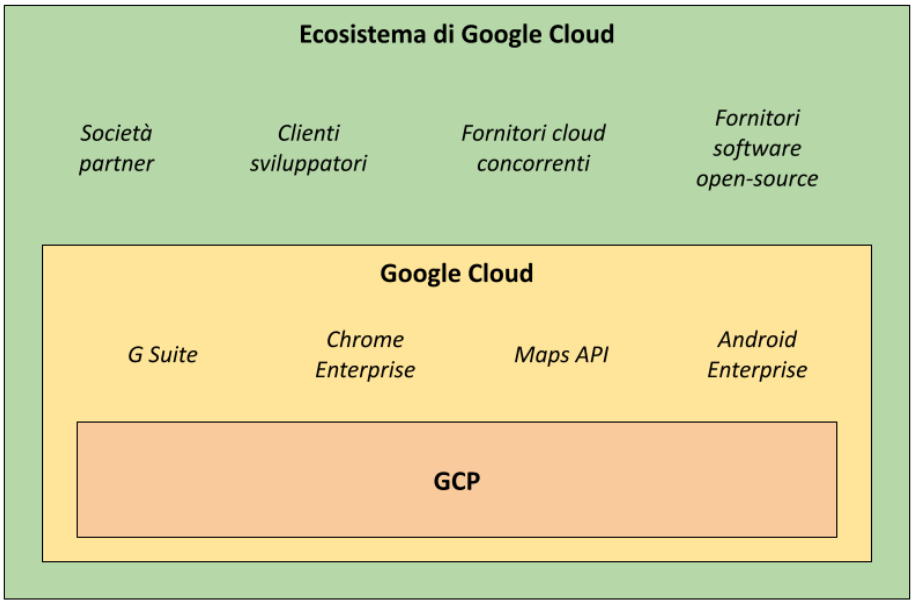
L'attuale logo di Google Cloud

Google Cloud racchiude in sé il già menzionato pacchetto di produttività aziendale Software-as-a-service chiamato G Suite, nonché versioni dei sistemi operativi Android e Chrome OS pensati per un uso in ambito di lavoro ed interfacce di programmazione per servizi come Google Maps.

Come già accennato, la piattaforma di Google Cloud per soluzioni di tipo IaaS e PaaS, così come servizi di networking, gestione dati e machine learning è chiamata *Google Cloud Platform*, spesso abbreviata in *GCP*.

Google Cloud è a sua volta considerabile parte di un intero ecosistema di aziende che offrono servizi e prodotti complementari o simili:

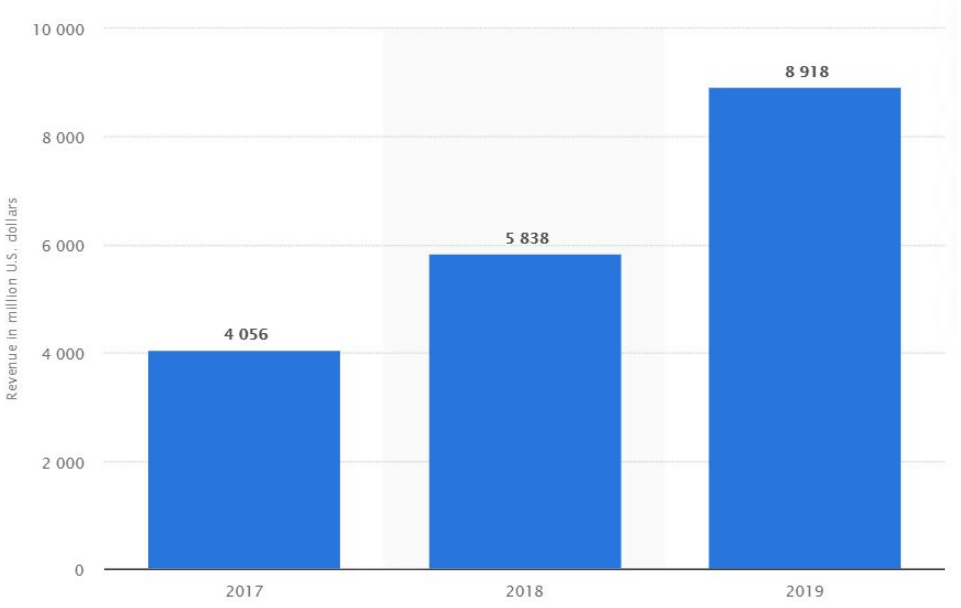
- ➔ società partner di consulenza manageriale, che propongono soluzioni cloud ai propri clienti come mezzo per raggiungere obiettivi strategici;
- ➔ organizzazioni che sviluppano software open source (come la *Apache Foundation*), software che è spesso alla base di molti servizi cloud commerciali;
- ➔ sviluppatori software ed ingegneri informatici, il cui feedback è sempre prezioso al fine di creare servizi cloud che possano effettivamente rispondere a bisogni concreti;
- ➔ fornitori concorrenti di cloud computing, con cui il rapporto è chiaramente di competizione sebbene rappresentino un tassello importante per la creazione di soluzioni ibride, come vedremo in seguito.



Schema dell'ecosistema di Google Cloud

Molte delle più grandi società al mondo, operanti nei settori più disparati, utilizzano al giorno d'oggi i servizi ed i prodotti offerti da Google Cloud, tra cui si possono citare FCA, Toyota, FedEx, Spotify, Bloomberg, Vodafone, HSBC, oltre a vari enti governativi tra cui anche, in Italia, la regione Veneto ed il comune di Bologna.

Nel 2019, le entrate di Google Cloud hanno raggiunto quota 8,9 miliardi di dollari. La sussidiaria di Google è in forte espansione, così come il resto del mercato dei servizi cloud, che verrà approfondito alla sezione seguente.



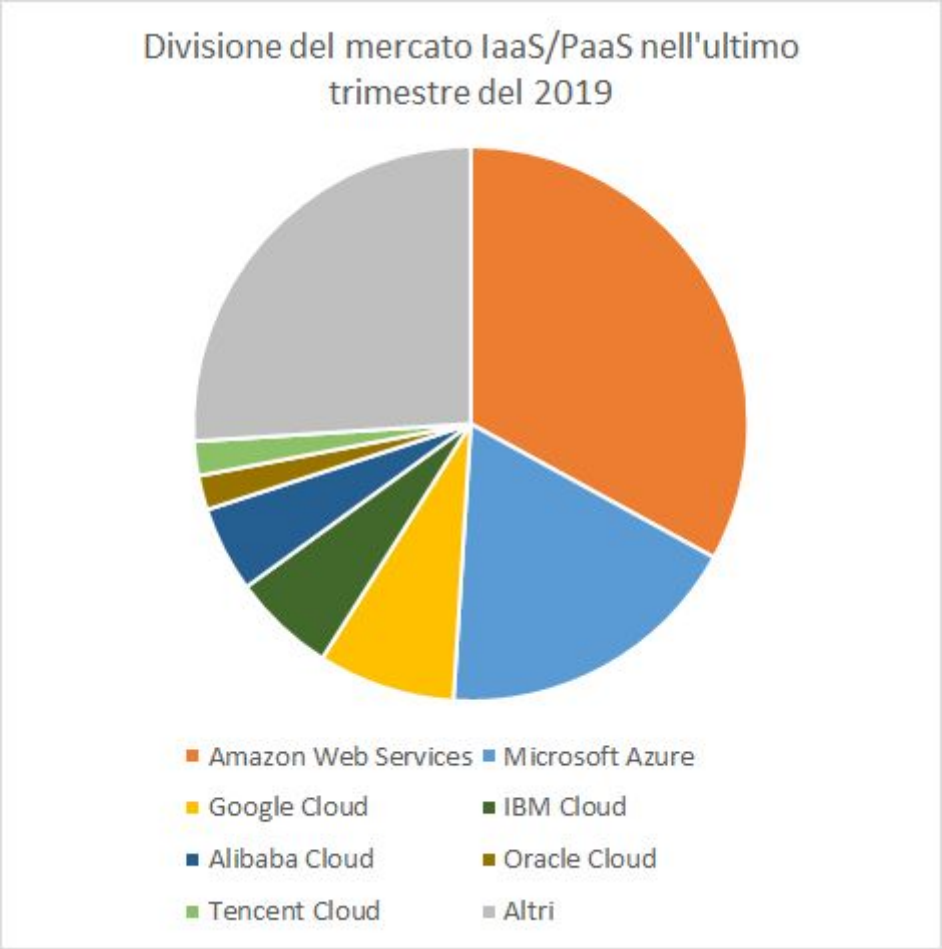
Crescita dei ricavi annuali di Google Cloud (Clement, 2020)

Il mercato del cloud computing

Il cloud computing è un'industria in continua espansione. La società di consulenza strategica Gartner prevede che si verificherà una crescita del mercato globale dei servizi cloud del 17% nel 2020, per un valore totale di 266 miliardi di dollari, e che entro il 2022 il 60% delle società farà uso di servizi cloud esterni, offerti su Internet da fornitori come Google Cloud.

Il mercato è attualmente dominato da tre colossi, che potremmo definire, per tale ragione, i "Big 3": Amazon Web Services, Microsoft Azure e Google Cloud. Si possono inoltre menzionare altri giganti di Internet e dell'informatica operanti nel settore, sebbene con

quote di mercato inferiori ed infrastrutture più contenute, quali IBM Cloud, Alibaba Cloud, Oracle Cloud e Tencent Cloud.



Quote dei leader del settore del mercato globale dei servizi IaaS/PaaS (dati Statista, 2020)

Almeno in termini di fatturato, il netto vincitore a livello globale, al momento, è sicuramente la sussidiaria del gruppo Amazon, che ha raggiunto a fine 2019 il 33% dei ricavi mondiali nel campo Infrastructure-as-a-Service e Platform-as-a-Service. Ciononostante, è interessante notare come negli ultimi anni Google Cloud abbia superato sia AWS sia Azure in termini di crescita percentuale e stia tuttora investendo capitali maggiori rispetto ad entrambi i rivali nel tentativo tutt'altro che banale di accorciare le distanze con i due colossi americani nella classifica del market share (Stevens, 2020).

E' possibile valutare il predominio sul mercato delle tre società leader ricorrendo ad uno strumento di ricerca creato da Gartner, noto come *Magic Quadrant*. Quest'ultimo consiste

in una rappresentazione grafica che mette in relazione l'effettiva capacità di esecuzione di varie società in un certo settore industriale con la completezza delle loro visioni strategiche, distinguendo quattro categorie di imprese:

- ➔ i *leader*, che come suggerisce il termine sono in possesso di strategie all'avanguardia e dei mezzi per rendere concrete tali strategie;
- ➔ i *visionaries*, ovvero i "visionari" che hanno idee avveniristiche per il settore ma mancano ancora della capacità di esecuzione necessaria;
- ➔ i *niche players*, i "giocatori di nicchia", che sono, a seconda dei casi, concentrati su un segmento limitato del mercato oppure sono privi della visione e delle risorse necessarie per superare i concorrenti;
- ➔ infine, i *challengers*, gli "sfidanti" che possiedono notevoli mezzi di produzione o mezzi tecnici ma non dimostrano attualmente una robusta comprensione della direzione del mercato.

Fatte le dovute premesse, si possono ora esplorare alcuni Magic Quadrant rilevanti.

Si può notare nel diagramma seguente come, per i servizi IaaS a livello globale, i "Big 3" siano in netto vantaggio rispetto ai concorrenti, posizionandosi nel quadrante dei leader. Le soluzioni IaaS richiedono infatti enormi investimenti di capitale e molto tempo per supportare la costruzione di server farm in tutto il mondo. Colmare il gap è un compito assai difficile per i competitor, dato che il mercato è in costante crescita e gli attuali leader continuano una sfrenata corsa all'espansione infrastrutturale.

Magic Quadrant per i servizi cloud IaaS (Wright, et al., 2019)

Si può anche osservare come, nel segmento dei servizi cloud di intelligenza artificiale, Amazon, Google e Microsoft siano in netto vantaggio rispetto ai concorrenti. Queste tre società sono infatti da tempo pioniere di un approccio completamente *data-driven*, cioè guidato dai dati, per quanto riguarda quasi ogni aspetto decisionale strategico in azienda. Basti pensare ai complessi programmi di machine learning usati da Amazon per predire con altissima precisione il numero di ordini da soddisfare in un dato lasso di tempo per ogni singolo prodotto offerto.

Magic Quadrant per i servizi cloud agli sviluppatori di IA (Baker, et al., 2019)

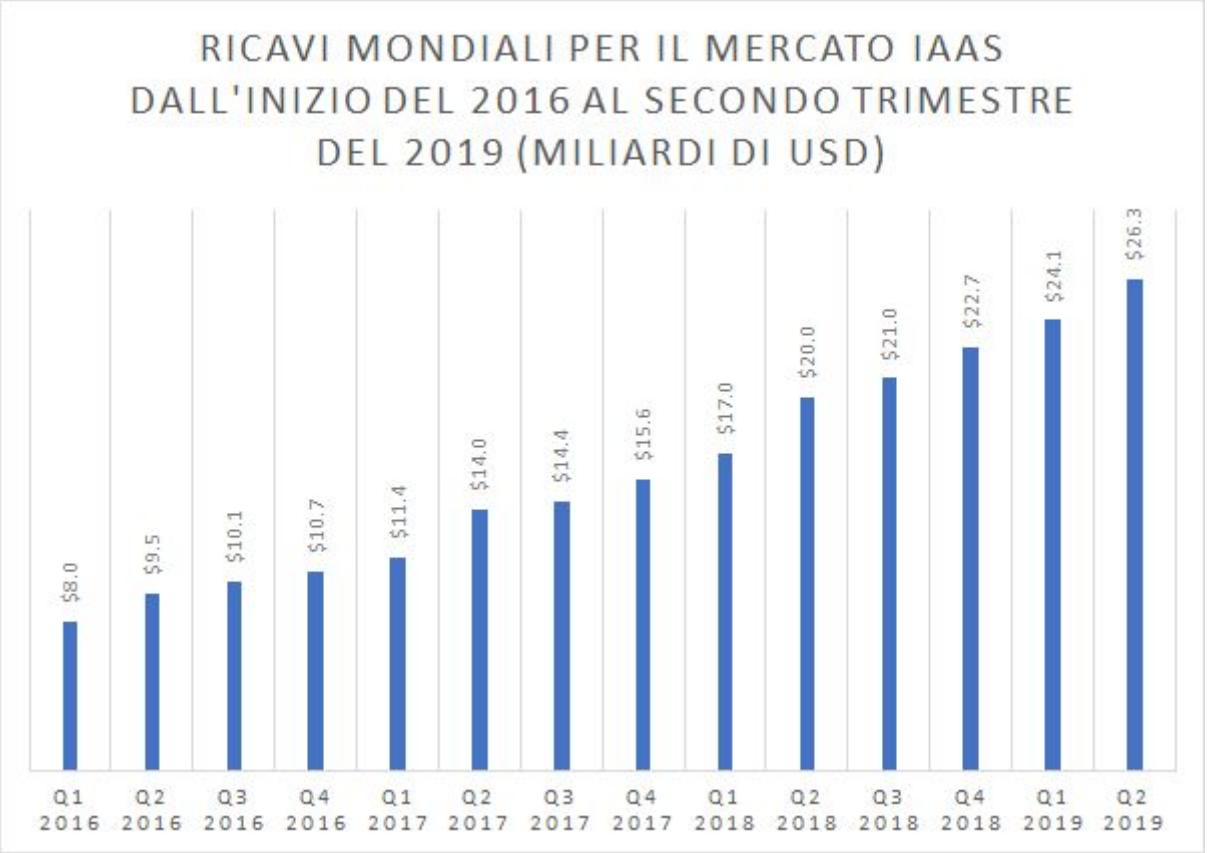
Si ponga inoltre attenzione all'enorme quantità di dati dei clienti accumulati nel tempo dai tre colossi, come i dati di ricerca per quanto riguarda Google, o i dati di acquisti e vendite per Amazon. YouTube, il noto servizio di streaming video offerto da Google, sfrutta ad esempio machine learning estremamente sofisticato per proporre contenuti interessanti ai propri utenti in base alla loro cronologia di visione.

Tali società mettono quindi a disposizione delle proprie aziende clienti dei servizi di IA considerabili *best-in-class*. Il vantaggio competitivo derivante è notevole poiché asset come dati ed algoritmi sono estremamente difficili da replicare.

Cosa si può dire del resto dei player che operano nel settore del cloud computing? Esistono sicuramente aziende considerabili "pesci più piccoli" rispetto ai colossi già menzionati, che vanno però distinti in base alla vastità dei servizi offerti. Ritornando al concetto di "giocatori di nicchia", possiamo considerare tra di essi fornitori cloud "generalisti" (ovvero fornitori di servizi che rispondono ad una grande quantità di bisogni differenti) come Google Cloud e AWS, ma su scala ridotta, oppure fornitori cloud specializzati, concentrati sul risolvere al meglio un sottoinsieme limitato di problemi.

I fornitori "generalisti" sono comunque classificabili come grandi aziende in termini di fatturato o patrimonio, semplicemente per il livello di investimento necessario per costruire e mantenere attiva ed aggiornata un'infrastruttura informatica grande abbastanza da poter ospitare i carichi di lavoro di svariati clienti. I fornitori specializzati, invece, possono effettivamente essere aziende di piccole dimensioni, specialmente quando i servizi erogati non necessitano massicciamente di infrastrutture dedicate.

Passando ad analizzare la crescita ed il futuro del mercato del cloud, si può notare dal grafico seguente come l'espansione non accenni a diminuire, facendo lievitare i ricavi globali, in appena quattro anni, da 8 a 26.3 miliardi di dollari.



Crescita dei ricavi globali del mercato dei servizi di infrastrutture cloud (dati Statista, 2019)

Sebbene queste cifre appaiano elevate, se si contestualizzano mettendole a confronto con la spesa globale per l'informatica in generale, stimata tra le 3 e le 4 *migliaia di miliardi* di dollari (Gartner, 2020), si realizza che il mercato è in realtà ancora molto giovane, ed i margini di crescita restano decisamente ampi.

Si consideri anche come i servizi cloud odierni operino una sorta di "democratizzazione" delle infrastrutture informatiche su larga scala, prima quasi appannaggio esclusivo di medie e grandi aziende ed ora facilmente accessibili on-demand a prezzi contenuti anche per realtà molto piccole, come startup o liberi professionisti, che prima non avrebbero avuto le risorse necessarie per pagare i servizi di colocation di una grande società come IBM, ad esempio. Questo nuovo segmento di piccole imprese e startup si aggiunge ai fattori che possono contribuire alla crescita del mercato cloud.

Impatto di COVID-19

E' interessante, infine, notare come l'attuale crisi mondiale dovuta al diffondersi del nuovo coronavirus non sembri aver impattato negativamente sul mercato del cloud computing in maniera tanto grave quanto sugli altri settori dell'informatica.

In una ricerca del 2020, Gartner evidenzia infatti come la spesa aziendale per l'informatica in tutte le sue sfaccettature calerà in seguito alla pandemia, fatta eccezione per i servizi offerti dai fornitori cloud, in crescita del 19%, specialmente per quanto riguarda la telefonia basata sul cloud (crescita della spesa di 8.9 punti percentuali) e le soluzioni per videoconferenze e meeting virtuali (crescita del 24.3%); l'ascesa di questi servizi potrebbe essere una diretta conseguenza della spinta verso lo *smart working*, il lavoro in remoto, dovuta al distanziamento sociale necessario per fronteggiare il virus.

Architetture di sistemi nel cloud

Prima di affrontare in maniera più approfondita il tema dei vari servizi offerti dai fornitori cloud, è necessario innanzitutto partire da una trattazione della cosiddetta architettura dei sistemi informatici implementabili nel cloud, ovvero l'organizzazione dei componenti fondamentali che costituiscono tali sistemi, nonché le loro interrelazioni.

Un esempio di sistema informatico realizzabile nel cloud è la nota *applicazione web* o *web app*. Quest'ultima è costituita da un qualsiasi programma distribuito ed accessibile tramite una rete informatica, come una intranet aziendale (una rete privata) o Internet (la pubblica "rete delle reti" informatiche).

Verranno spiegati di seguito alcuni concetti architetturali fondamentali per capire le ragioni che sono alla base dell'organizzazione e dell'offerta dei servizi cloud odierni.

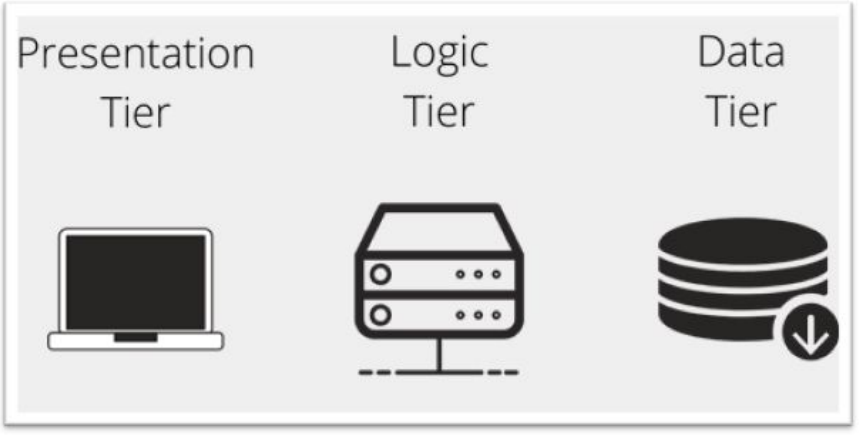
Strati

Un *tier*, o *strato*, rappresenta una separazione fisica tra i componenti di un'applicazione o di un servizio (Microsoft, 2014). Componenti di un'applicazione possono essere, ad esempio, le basi di dati e le interfacce utente. Tali componenti, di cui si discuterà più avanti in dettaglio, asservono a specifici scopi per garantire il funzionamento dell'applicazione. A grandi linee, le architetture per applicazioni possono essere classificate in base a quanti strati vengono utilizzati dalle stesse.

Un'applicazione *single-tier* è un'applicazione in cui lo *strato di presentazione* o interfaccia utente, la *business logic* o logica di business (ovvero i processi e le azioni che deve compiere l'applicazione per soddisfare gli obiettivi per cui è stata creata) e la base di dati risiedono nella stessa macchina. Un esempio di applicazione single-tier è l'editor di testo Microsoft Word.

Un'applicazione a due strati o *two-tier* si compone di un *client* e di un *server*, quindi di una parte che funge da cliente ed una che si comporta da inserviente. Il client contiene l'interfaccia utente e la business logic, mentre il database è affidato al server.

In un'architettura di tipo *three-tier*, infine tutti i componenti descritti prima, ovvero business logic, interfaccia utente e database sono fisicamente separati e residenti in macchine diverse.



Schema dell'architettura three-tier

La suddivisione di un'applicazione in strati viene fatta allo scopo di aderire al principio della cosiddetta *separation of concerns*, cioè la "separazione delle preoccupazioni" o anche la "separazione dei compiti". Questo principio stabilisce che ciascuno strato di un programma si debba occupare di un solo obiettivo, che sia esso la gestione dei dati o la presentazione dell'interfaccia all'utente finale.

Separando le preoccupazioni, diventa più facile mantenere ordine tra i vari strati e permettere a personale specializzato di occuparsi di ciascuno di essi. La potenziale superficie di attacco esposta a malintenzionati, inoltre, si riduce: un attacco di successo verso un computer dove risiede un determinato tier, ad esempio l'interfaccia utente, non garantirebbe automaticamente ad un hacker l'accesso al database dell'applicazione.

Utilizzare più strati, infine, garantisce un certo grado di riusabilità degli stessi in differenti applicazioni (a patto che tali strati non siano eccessivamente interdipendenti), riducendo la necessità di dover "ripartire da zero" nel dover ingegnerizzare un nuovo servizio di rete.

Scalabilità, alta disponibilità, affidabilità

La *scalabilità* è la capacità di un servizio di adattare la propria infrastruttura in base al traffico, e quindi al carico, che grava su tale servizio (Sullivan, 2019). Di solito si distingue la scalabilità in verticale e orizzontale.

La *scalabilità verticale* è l'approccio classico alla gestione di un incremento di domanda di risorse offerte da un servizio web, come può essere ad esempio un blog; scalare verticalmente significa in pratica aumentare il potere di processamento o di memorizzazione del server che si ha a disposizione: si può, in sostanza, aumentare la RAM, passare ad una CPU più performante o incrementare la capacità del disco fisso. La scalabilità verticale è la più facile da attuare e gestire e dovrebbe essere la prima soluzione ad essere presa in considerazione in caso di un aumento di traffico verso il servizio web.

La *scalabilità orizzontale*, d'altro canto, è un approccio che prevede l'aggiunta di più server che possano gestire il traffico di rete a supporto del server iniziale; piuttosto che incrementare la potenza di una singola macchina, si va a costituire un sistema composto da più macchine, ciascuna contenente una replica del servizio che si vuole offrire.

La scalabilità verticale, prima o poi, a causa di limiti tecnici dei singoli componenti di un computer, si arresta; la scalabilità orizzontale, al contrario, è virtualmente senza limiti: basta aggiungere costantemente nuovi server (che, presi singolarmente, non necessitano di hardware eccezionale) al *pool* di risorse disponibili. Il problema principale di questo approccio, in passato, era insito alla complessità di installare e mantenere un numero sempre crescente di server: questo problema è superato dagli odierni provider cloud, che tramite economie di scala, permettono ai propri clienti l'accesso semi-istantaneo ad un'enorme quantità di server, liberandoli degli oneri di gestione fisica delle macchine e dello spazio riservato ad esse. In virtù di ciò, è lecito riferirsi a tali fornitori come *hyperscalers*, data la loro capacità di scalare orizzontalmente, aggiungendo e rimuovendo risorse informatiche dinamicamente ed automaticamente a seconda delle necessità dei singoli clienti (qualità dei sistemi cloud definita come *cloud elasticity*, ovvero *elasticità del cloud*).

Esiste, tuttavia, un'importante condizione da evidenziare, necessaria per l'approccio di *horizontal scaling*: i server dell'applicazione devono necessariamente essere *stateless*. Con questo termine, a cui si affianca il concetto contrario di *stateful*, si intende che i singoli server non possono gestire e far permanere in memoria informazioni relative allo stato particolare dell'applicazione per un singolo utente; ad esempio, se un utente decide di modificare la lingua dell'interfaccia dell'applicazione da inglese ad italiano, l'informazione relativa alla preferenza espressa non potrà essere salvata nella memoria del computer che sta, in quel momento, servendo l'utente.

Il motivo di questa limitazione è semplice: il *load balancer* (bilanciatore di carico) del sistema, ovvero il software (o hardware) che si occupa di direzionare il traffico degli utenti ai singoli server che ospitano l'applicazione, opera in base a criteri come l'utilizzo di CPU o la percentuale di traffico per ogni computer. Nell'esempio precedente, in caso di disconnessione dell'utente, non è affatto detto che, ad una successiva connessione dello stesso utente, il load balancer dirigerà l'utente allo stesso server visitato precedentemente. Ciò significa che l'informazione relativa alla lingua preferita dall'utente potrebbe non essere più disponibile per quell'utente, e anzi potrebbe essere applicata erroneamente ad altri utenti diretti dal load balancer verso il server dove era stata memorizzata quell'informazione.

Date queste ragioni, per ovviare al problema del salvataggio dello stato dell'applicazione per i singoli utenti si ricorre ad altre macchine adibite esclusivamente alla funzione di basi di dati, come si vedrà in seguito, oppure, per informazioni non critiche o sensibili (come la preferenza di un tema colori scuro piuttosto che un tema colori chiaro), al salvataggio dello stato nel dispositivo dell'utente, sotto forma di tecnologie web come i *cookies* o il *session/local storage*.

Utilizzare più server piuttosto che uno solo, come accade scalando orizzontalmente un servizio, aiuta anche in un altro aspetto molto importante: l'*alta disponibilità*. Quest'ultima identifica il funzionamento continuo di un sistema informatico con una capacità tale da soddisfare sufficientemente la domanda di risorse che scaturisce dalle operazioni richieste per tale sistema (Sullivan, 2019). La disponibilità è di solito misurata come la percentuale di

tempo in cui un sistema è in grado di rispondere alle richieste degli utenti con una latenza uguale o inferiore ad un certo valore soglia.

E' evidente che la disponibilità sia importante per qualsiasi applicazione: ad esempio, se un sito di e-commerce diventa irraggiungibile per un certo periodo di tempo si avrà un diretto impatto negativo sulle vendite, che non potranno essere processate. A volte, la non disponibilità di un sistema può essere catastrofica, come per esempio per le applicazioni usate in contesti sanitari o aerospaziali.

L'alta disponibilità nel cloud è raggiunta, come accennato, tramite la *ridondanza*, ovvero incrementando il numero di risorse essenziali per il funzionamento dell'applicazione, ad esempio utilizzando un gruppo di server asserviti allo stesso scopo piuttosto che un solo server. In questo modo, anche in caso di mancanza di disponibilità di una macchina, le richieste degli utenti potranno essere gestite dalle altre macchine ancora online.

Un'ultima caratteristica da considerare nella progettazione di un'applicazione web è l'*affidabilità*. Questo aspetto è molto legato al concetto di disponibilità, poiché misura la probabilità che un'applicazione sia disponibile, appunto, e capace di soddisfare correttamente le dovute richieste del sistema (Sullivan, 2019). Piuttosto che misurare solamente una percentuale di tempo in cui il sistema è disponibile, però, l'affidabilità misura ad esempio la percentuale di richieste a cui l'applicazione risponde *con successo*. Un'applicazione potrebbe funzionare infatti in maniera non corretta, pur essendo disponibile agli utenti.

L'affidabilità richiede più enfasi, rispetto all'alta disponibilità, su temi gestionali ed organizzativi come identificare sistemi di monitoraggio, notifica e risposta rapida agli episodi di fallimento del sistema.

Architetture monolitiche e a microservizi

Un'applicazione è strutturata secondo un'architettura cosiddetta monolitica se il suo intero codice è contenuto in un'unica base di codice (*codebase*). In altre parole, in un'architettura monolitica tutti i componenti di un'applicazione sono fortemente integrati tra di loro e scritti nello stesso linguaggio di programmazione. L'architettura monolitica rappresenta l'approccio standard alla realizzazione di applicazioni web, data la sua immediatezza ed iniziale semplicità.

Le limitazioni dell'architettura monolitica, tuttavia, sono molteplici: innanzitutto troviamo il problema della scalabilità, poiché parti diverse dell'applicazione potrebbero risentire in modo diverso del carico applicato dal traffico Internet ma, essendo tutti i componenti di tale applicazione strettamente interconnessi, non risulta possibile applicare ad ognuno di essi politiche di *scaling* diverse. Un altro fattore che limita la scalabilità è la conservazione dello stato nell'applicazione monolitica: come spiegato in precedenza, la scalabilità orizzontale è attuabile solo in caso di componenti stateless, limitando spesso le applicazioni monolitiche alla scalabilità verticale.

Questi limiti vengono superati dall'architettura a microservizi (*microservices*), paradigma che impone la scomposizione delle diverse funzionalità di un'applicazione in moduli di codice ben distinti che interagiscono tra di loro. Questi moduli devono essere, in gergo, a "basso accoppiamento" (*loosely coupled*), ovvero devono dipendere il meno possibile dagli altri moduli con cui interagiscono: garantendo un grado di accoppiamento ragionevolmente basso, i microservizi possono essere mantenuti e sviluppati da team diversi di programmatori, aumentando la produttività e l'agilità data dal poter implementare rapidamente nuove funzionalità. E' più facile, inoltre, tracciare ed isolare *bug* nel codice, sapendo che un errore riscontrato in una parte dell'applicazione è ascrivibile singolarmente a uno specifico microservizio. Il basso accoppiamento favorisce anche la riusabilità dei microservizi, che possono essere facilmente riadattati per poter funzionare nel contesto di altre, differenti, applicazioni.

L’architettura a microservizi permette la scalabilità orizzontale, dato che le variabili di stato possono essere isolate e confinate in database indipendenti dagli altri elementi che compongono un’applicazione, come ad esempio l’interfaccia utente. E’ più facile individuare *bottlenecks* nella performance e diventa possibile garantire l’adeguata (e dinamica) assegnazione di risorse ai singoli microservizi.

Nonostante questi vantaggi, l’architettura a microservizi non è esente da punti deboli, il principale dei quali è la gestione di una complessità architetturale maggiore: in generale, se si ha la certezza che un’applicazione non si troverà mai ad affrontare significativi problemi di traffico di rete, come nel caso di un servizio web aziendale destinato esclusivamente ad un uso interno, è meglio usare un approccio monolitico, in virtù della sua semplicità ed immediatezza maggiore.

Il cambiamento da un’architettura monolitica ad un’architettura a microservizi è, in ogni caso, sicuramente possibile, sebbene dispendioso di risorse e di certo non immediato. E’ sempre consigliabile quindi valutare con molta attenzione tutti i requisiti di business a cui l’applicazione da sviluppare deve rispondere, in modo da non incorrere in futuro in complicazioni e problemi tecnici.

I microservizi sono oggi l’approccio preferito per grandi e complesse applicazioni di livello enterprise, pensate per sfruttare al meglio le possibilità del cloud computing. Vedremo nella prossima sezione come i microservizi vengono effettivamente implementati a livello infrastrutturale nel cloud, esplorando il concetto di virtualizzazione.

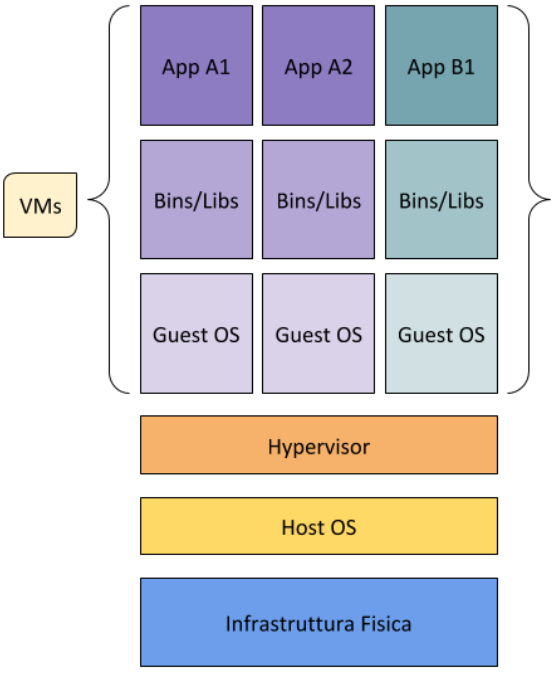
Virtualizzazione

La virtualizzazione è un concetto fondamentale nell'informatica odierna: il primo calcolatore con capacità di virtualizzazione fu infatti realizzato da IBM già nel 1968 (Fedoseenko, 2019); questo paradigma è inoltre alla base del funzionamento del cloud computing.

Si discuterà qui di seguito della virtualizzazione a livello di hardware, ovvero di macchine virtuali, e della virtualizzazione a livello di sistema operativo, tramite il paradigma dei *container*.

Macchine virtuali

Si parla di macchina virtuale, o *virtual machine* (abbreviata sovente in *VM*), per indicare un tipo di software il cui scopo è simulare un calcolatore fisico: in una macchina virtuale, tutti i componenti che costituiscono un computer, come dischi, memoria e scheda di rete diventano astratti e dinamici. Lo schema seguente illustra questo concetto.



Schema logico di funzionamento delle macchine virtuali

Secondo questo modello di software, le risorse dell'infrastruttura fisica vengono frazionate: ogni macchina virtuale riceve una determinata allocazione, ad esempio, della memoria RAM

concreta sottostante. Il grande vantaggio che ne deriva è quello di poter scomporre un singolo calcolatore in un numero a piacimento (limitato teoricamente dalle risorse fisiche disponibili) di calcolatori virtuali senza alcuna necessità di intervenire fisicamente. Ogni macchina virtuale è inoltre ridimensionabile o eliminabile a piacimento e può ospitare un sistema operativo qualsiasi, persino se quest'ultimo è diverso dal sistema operativo su cui è in esecuzione la macchina virtuale.

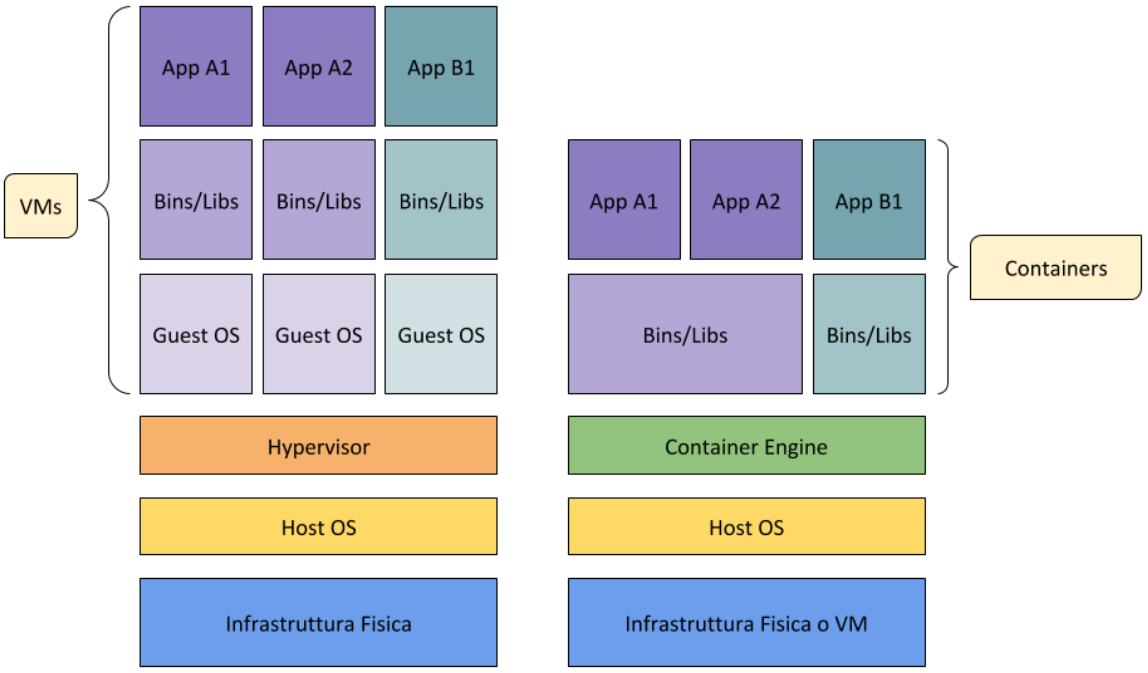
La creazione e la gestione delle macchine virtuali sono compito del cosiddetto *hypervisor*, un "ipervisore", così chiamato perché rappresenta un "supervisore dei supervisori", questi ultimi intesi come i *kernel*, ovvero i programmi di controllo, dei sistemi operativi delle macchine virtuali.

Un altro aspetto fondamentale che deriva dalla virtualizzazione è che diverse VM, installate su uno stesso calcolatore, possono essere completamente isolate tra di loro; questo fa sì che macchine virtuali appartenenti a clienti diversi possano coesistere in uno stesso ambiente, condizione fondamentale per i fornitori cloud, che possono così allocare in maniera estremamente dinamica, efficiente e sicura tutte le risorse computazionali fisiche a propria disposizione. Si vedrà meglio questo concetto alla sezione "Servizi di infrastruttura" più avanti.

Container

Concettualmente, mentre una macchina virtuale astrae l'hardware di un calcolatore, un container opera ad un livello superiore e permette di astrarre il sistema operativo stesso. Lo standard de-facto che si è imposto a livello internazionale per le soluzioni di containerizzazione è fornito dalla società di servizi informatici *Docker*.

Tutti i container operanti su uno stesso computer (sia fisico sia sotto forma di macchina virtuale) vengono eseguiti da un singolo kernel del sistema operativo e riescono quindi a poter usare molte meno risorse rispetto alle VM. Come per le macchine virtuali, una rappresentazione grafica aiuta a comprendere questi concetti:



Container e macchine virtuali a confronto

Tramite i container, si riescono ad evitare i conflitti causati da dipendenze software o librerie mancanti, poiché il codice dell'applicazione viene "impacchettato" insieme alle necessarie librerie e file binari; la distribuzione e replica può così poi avvenire su qualsiasi macchina avente il container engine appropriato. Non solo: data l'astrazione del sistema operativo, i container risultano molto più leggeri in termini di dimensioni e veloci in termini di tempi di creazione, avvio ed eliminazione.

Questo comporta enormi vantaggi di performance e flessibilità nell'utilizzo dei container come mezzo per ottenere una scalabilità orizzontale rapida e reattiva, ragione per cui anche i container rappresentano una tecnologia fondamentale per il cloud. A riprova di tutto ciò, si consideri che tutti i prodotti e servizi Google vengono attualmente eseguiti in container (Google Cloud, 2020).

I container rappresentano il paradigma ideale per i microservizi data la loro maggiore agilità e leggerezza: infatti, avendo a che fare con una grande quantità di servizi operanti entro sistemi separati, risulterebbe troppo inefficiente dover usare macchine virtuali per ognuno

di essi, poiché a ciascun microservizio verrebbe associato un *overhead* inutile dato dai sistemi operativi incapsulati entro ogni VM; questo problema è tuttavia molto meno sentito per quanto riguarda le applicazioni monolitiche, poiché il codice può risiedere interamente in un solo computer e non si riscontrano gli stessi problemi di scalabilità nel dover replicare una singola VM.

Avendo a che fare con una moltitudine di repliche di container diversi, sorge spontaneo il problema di come amministrare al meglio tali container e fare sì che certe azioni di manutenzione e controllo vengano automatizzate. Questo concetto prende il nome di *orchestrazione* dei container, e verrà approfondito in seguito, nella sezione relativa ai servizi cloud per i container.

Basi di dati

Si è accennato in precedenza alle basi di dati, comunemente chiamate con il termine inglese *database*. Un database non è altro che un software informatico creato ed ottimizzato per gestire efficientemente grandi quantità di dati, organizzati secondo strutture più o meno rigide. Dati che si prestano meglio ad essere rigidamente strutturati sono, ad esempio, informazioni relative agli ordini dei clienti in una applicazione ecommerce. Al contrario, i dati non strutturati sono invece, ad esempio, contenuti di tipo multimediale e documenti di grandi dimensioni.

Esistono varie tipologie di database, a seconda sia del tipo di dati su cui si deve lavorare, sia dello scopo ultimo della base di dati. Il database storicamente più utilizzato è quello di tipo *relazionale*, così chiamato perché ottimizzato e pensato per salvare efficientemente entità tra di loro relazionate; le relazioni possono essere del tipo uno-a-uno, uno-a-molti, molti-a-molti, molti-a-uno. I database relazionali sono rigidi, ovvero assicurano il salvataggio dei dati in maniera *normalizzata*: con ciò si intende che le informazioni contenute in una riga di una certa tabella nel database devono rispettare un preciso formato (chiamato *schema* del database), oltre al fatto che tali informazioni non sono innecessariamente duplicate in altre tabelle del database; ad esempio, avendo una tabella relativa agli ordini e una tabella relativa ai clienti, la tabella degli ordini conterrà dati relativi ad ogni singolo ordine, incluso il codice univoco che identifica l'ordine ed il codice univoco che identifica il cliente che ha effettuato l'ordine. Nella tabella degli ordini, però, non saranno presenti altre informazioni rispetto ai clienti, poiché tali informazioni saranno già salvate nella relativa tabella dedicata a tutti i clienti. Evitare la duplicazione di informazioni fa sì che, in caso sia necessario un aggiornamento relativo a queste ultime, esso potrà agire in una sola tabella, assicurando la correttezza globale delle informazioni contenute nel database.

Proseguendo l'esempio, consideriamo di voler mettere in relazione le informazioni di utenti ed ordini in modo da ottenere una lista dei dieci clienti che hanno speso di più, in totale, nel mese di Marzo: questo è reso possibile tramite *SQL*, acronimo di "Structured Query Language", ovvero "linguaggio strutturato di interrogazione", che consiste in una serie di

istruzioni testuali che vengono interpretate dal sistema di gestione del database in modo da ottenere le informazioni desiderate dall'utente.

I database relazionali assicurano inoltre la possibilità di transazioni di tipo *ACID* (IBM, 2020), acronimo che sta per "Atomicity, Consistency, Isolation, Durability", ovvero "Atomicità, Coerenza, Isolamento, Durabilità". In breve, ciò significa che le transazioni, cioè le serie di operazioni che portano ad un aggiornamento di una parte dei dati contenuti nel database, non possono essere effettuate parzialmente: se hanno successo, devono portare ad un cambiamento permanente e comprensivo di tutte le operazioni prefissate; se non hanno successo, invece, il sistema dovrà ripristinare lo stato del database risalente all'inizio della serie di operazioni.

Un esempio classico di applicazione pratica di transazioni ACID è rappresentato dal database di una banca: quando un utente ordina un bonifico bancario, il sistema effettua una transazione, avendo la certezza che le derivanti operazioni di aggiornamento dati o avranno successo e muteranno permanentemente lo stato del database, o falliranno e riporteranno il database allo stato iniziale. Non si verificherà mai un caso in cui le informazioni relative ai bonifici porteranno ad uno stato di incoerenza del database: in parole povere, i conti torneranno sempre.

La limitazione principale dei database relazionali è data dalla loro scarsa scalabilità orizzontale: aumentare le prestazioni di queste basi di dati è, infatti, piuttosto laborioso e complesso, e richiede attenta pianificazione ed intervento umano. Ad esempio, è diffusa la pratica conosciuta come *sharding* (Drake, 2019), secondo cui le tabelle in una base di dati vengono partizionate orizzontalmente e successivamente queste partizioni vengono distribuite in più server; si pensi, come esempio, alla scomposizione di un puzzle: tutti i frammenti contengono informazioni uniche rispetto alla "immagine" totale.

Se la scalabilità è un limite per i database relazionali, per i database *NoSQL* è invece un vantaggio. Come si può evincere dal nome, queste basi di dati prescindono sia dal linguaggio SQL sia, di conseguenza, dalla struttura relazionale rigida, ovvero lo schema, dei sistemi precedentemente descritti. I database NoSQL sacrificano parte dei capisaldi ACID in favore

di una maggiore semplicità nella struttura dei dati e di capacità di scalare orizzontalmente. In aggiunta, a seconda del tipo di applicazione richiesta, la performance in lettura e scrittura potrà essere, in alcuni casi, superiore ad una controparte relazionale.

Uno dei modelli NoSQL più utilizzati è il *database di documenti*, che organizza le informazioni secondo, come suggerisce il nome, dei documenti testuali semi-strutturati, di solito in un formato di tipo *JSON*. JSON, derivato dal linguaggio di programmazione *JavaScript*, è uno standard aperto che usa un formato di testo facilmente interpretabile dagli umani, ed è usato per archiviare e trasmettere informazioni sotto forma di coppie che associano attributi ai valori (anche vettoriali) di tali attributi.

Esempio di documento in formato JSON

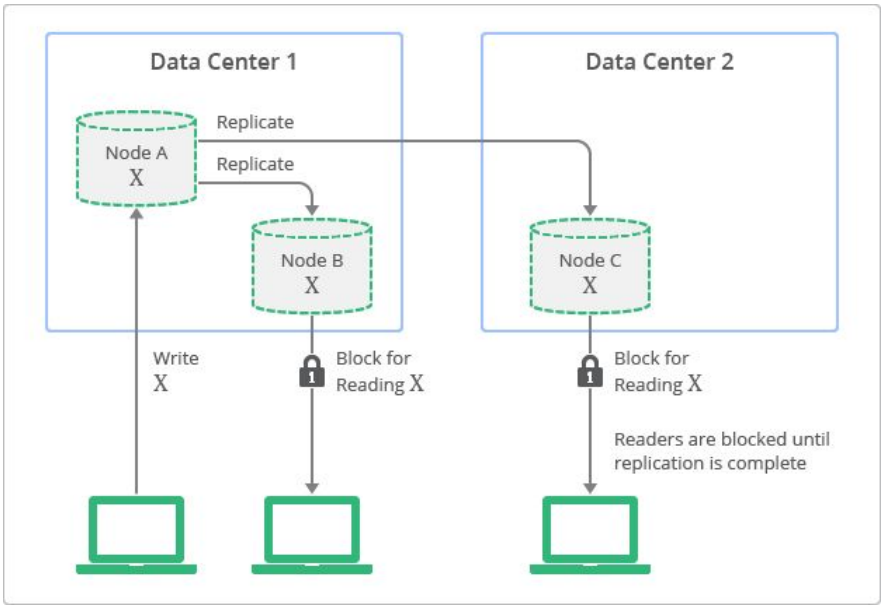
Un altro database NoSQL popolare è il cosiddetto database a grafo, che organizza invece le informazioni sotto forma di *nodi*, che rappresentano delle entità, ed *archi*, che rappresentano le relazioni tra le entità. Tali database sono molto veloci nella lettura di associazioni di dati poiché, al contrario delle basi di dati SQL tradizionali, le relazioni non sono calcolate al momento dell'interrogazione del database ma sono già conservate all'interno del database stesso tramite gli archi appena descritti.

Data la loro natura priva di schema (*schemaless*), i database NoSQL permettono inoltre grande flessibilità in fase di sviluppo e prototipazione: consentono, infatti, di poter aggiungere e togliere informazioni a piacimento da singole entità presenti al loro interno, senza timore di provocare errori a causa della rottura di qualche tipo di relazione tra i dati. Il

rovescio della medaglia è che questa flessibilità e, in un certo senso, mancanza di regole ferree può dare luogo ad incongruenze tra dati appartenenti alle stesse entità ma ubicati in più parti del database.

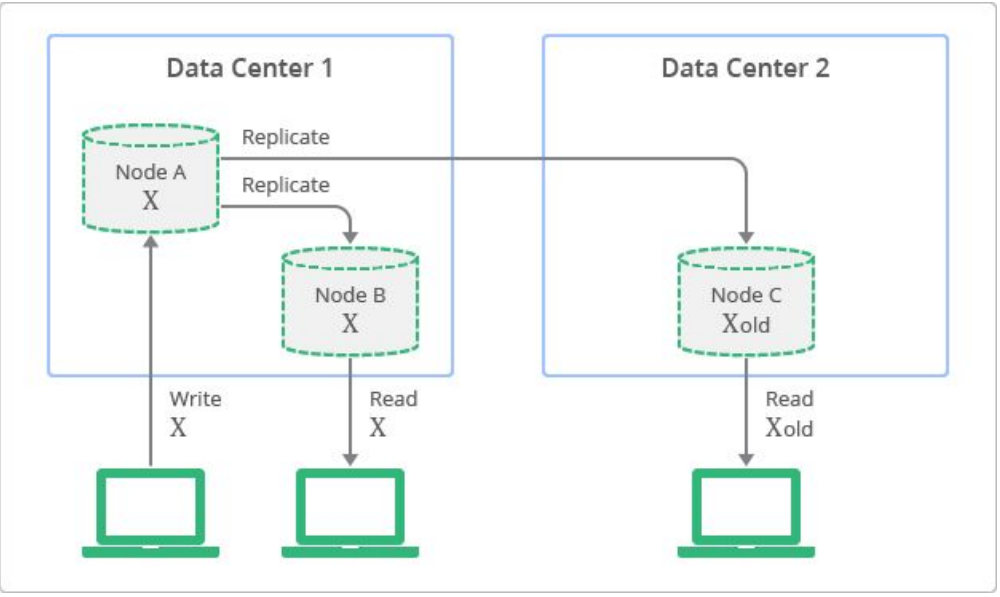
Un'altra importante limitazione dei database non relazionali è data dalla loro tendenza ad essere *eventually consistent*, ovvero *infine coerenti*, piuttosto che *strongly consistent*, o *fortemente coerenti*, al contrario dei database relazionali: questi ultimi vengono anche definiti *immediatamente* coerenti, poiché i dati nel sistema, dopo una qualsiasi operazione che comporti un aggiornamento, saranno globalmente ed immediatamente visti nel loro ultimo stato aggiornato da tutti gli utenti; in altre parole, due utenti diversi vedranno, in qualsiasi momento, un medesimo valore per una particolare entità. I dati saranno quindi coerenti per ogni osservatore.

Il problema di questo approccio è che compromette la scalabilità e la performance dell'applicazione, perché i dati implicati in un'operazione di aggiornamento devono essere bloccati globalmente fino alla fine di tale operazione in modo da garantire che nessun altro utente possa, nel frattempo, generare conflitti nel database tramite operazioni contemporanee di aggiornamento. La figura in seguito illustra questo meccanismo.



Rappresentazione della strong consistency o coerenza immediata (Google Cloud, 2020)

Al contrario, nei database non relazionali il concetto di coerenza dei dati è, generalmente, dilazionato nel tempo: per garantire scalabilità e rapidità, effettuare un'operazione di aggiornamento non comporta un blocco globale dei dati interessati per tutti gli utenti: si potranno quindi verificare situazioni in cui la modifica contemporanea di un dato da parte di più utenti risulterà in uno stato di temporanea incoerenza, in cui due utenti diversi potrebbero riscontrare discrepanze nel valore di una particolare entità.



Rappresentazione della eventual consistency o coerenza finale (Google Cloud, 2020)

La coerenza finale, in molti casi, è sufficiente; si pensi ad esempio al numero di "mi piace" o di votazioni positive e negative per un certo post pubblicato in un social network: il fatto che due utenti possano riscontrare, ad uno stesso momento, 812 voti uno e 850 volti l'altro non compromette l'usabilità dell'applicazione. Al contrario, informazioni più delicate, come l'esito di una transazione finanziaria, devono risultare immediatamente coerenti.

Operazioni sincrone ed asincrone

Nello sviluppo di un'architettura per una web app, i vari processi logici ed i flussi di dati che passano da un componente ad un altro, innescati ad esempio in seguito ad una certa azione dell'utente, possono essere distinti in base alle loro caratteristiche di *sincronia* od *asincronia*.

Un'operazione *sincrona* è un tipo di chiamata o richiesta ad un certo servizio o componente tale per cui l'esecuzione del codice di questa operazione si arresterà e rimarrà in attesa fino a quando il servizio interrogato non avrà completato a sua volta le operazioni necessarie per elaborare un certo tipo di risposta da trasmettere finalmente alla prima operazione.

Questo tipo di interazioni di richiesta e risposta tra servizi può anche concatenarsi ed arrivare ad impegnare più componenti allo stesso tempo: è evidente quindi che, specialmente se tali componenti non risiedono sulla stessa macchina ma sono distribuiti tra più macchine distinte, come accade nel caso di un'architettura a microservizi, la latenza di rete esistente tra gli strati dell'applicazione si somma e può dare luogo a rallentamenti per l'intero sistema.

In alcuni casi, la sincronia nelle operazioni è accettabile, se ad esempio il flusso di operazioni da compiere è molto semplice oppure se può essere garantita una certa velocità di esecuzione; in altri casi, la sincronia è necessaria: si pensi all'erogazione di un servizio a pagamento, che può procedere solamente in caso l'operazione di acquisto sia andata a buon fine.

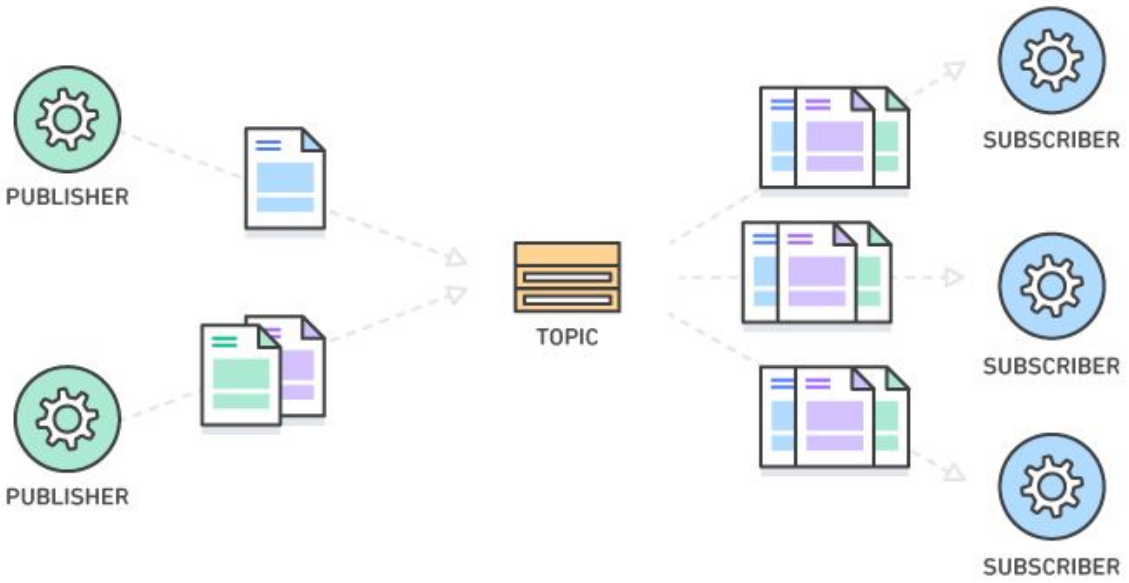
Nei casi in cui la sincronia non sia necessaria e possa dare luogo a ritardi non indifferenti, si può ricorrere ad un sistema di operazioni asincrone. Per implementare operazioni di questo genere si possono utilizzare due modelli differenti: le *code di messaggi* o la messaggistica *Pub/Sub*.

Le code di messaggi rappresentano un paradigma che consente di disaccoppiare il tasso di richieste tra due servizi. In una coda di messaggi, un servizio che agisce da mittente, detto

producer, inoltra la propria richiesta verso un certo altro servizio destinatario, detto *consumer*; la richiesta non è però inviata direttamente al destinatario: essa infatti viene prima inserita in una coda di dati amministrati da un servizio terzo, che agisce come gestore di tutte le richieste asincrone del sistema. In questo modo, il servizio emittente potrà generare una serie di richieste ad un certo tasso, mentre il servizio destinatario potrà rispondere ad un un altro tasso, anche più lento del primo, poiché le informazioni riguardanti tutte le richieste da soddisfare saranno conservate in sicurezza nella coda di messaggi.

Rappresentazione grafica di una coda di messaggi (AWS, 2020)

Nel modello della coda di messaggi, ogni messaggio parte da un singolo producer ed arriva ad un certo singolo consumer: per questa ragione, il modello viene spesso definito *one-to-one* (uno-a-uno) o *point-to-point* (punto-a-punto). Per superare questa limitazione, si può ricorrere ad un sistema di messaggistica Pub/Sub, abbreviazione di *Publisher/Subscriber*, traducibile come "Pubblicante-Iscritto": in questo modello, i servizi a monte, ovvero i publisher, inoltrano informazioni ad un certo *topic* ("argomento"), ovvero una raccolta di messaggi; a questo punto, i messaggi contenuti in questa raccolta verranno schedulati per l'invio a tutti i servizi a valle etichettati come "iscritti" a tale topic.



Rappresentazione grafica di un sistema Pub/Sub (AWS, 2020)

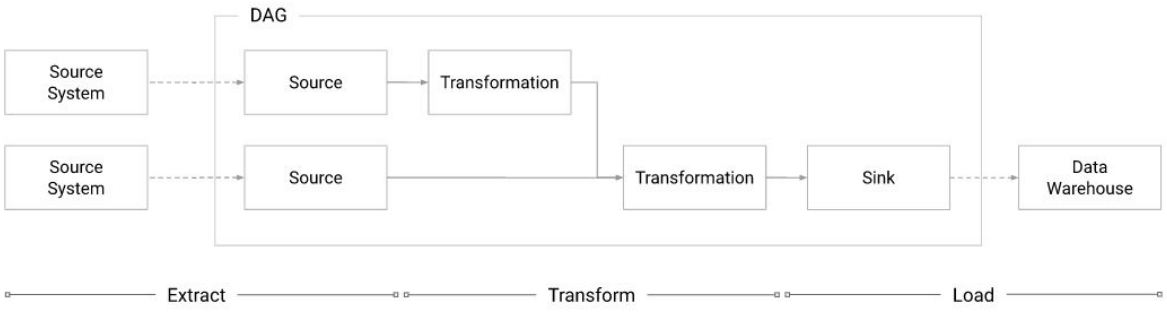
I messaggi saranno conservati nel topic fino a quando gli iscritti non saranno in grado di riceverli ma, al contrario della coda di messaggi, che in genere può garantire un comportamento *FIFO* (*First-In-First-Out*), l'ordine di ricezione delle informazioni potrebbe non rispettare l'ordine cronologico di emissione.

Flussi di dati

E’ spesso necessario, in determinati contesti di lavoro, architettare sistemi per trasferire quantità di dati da una certa fonte, spesso trasformare questi dati in vari modi e poi farli confluire in una determinata destinazione nel sistema informatico, spesso un database, per lo stoccaggio. I percorsi dove “scorrono” virtualmente i dati sono chiamati *data pipelines*, ovvero “tubature” di dati.

Una data pipeline è matematicamente un modello di *grafo aciclico orientato*, in inglese *directed acyclic graph* o *DAG*, ovvero una struttura ordinata di nodi collegati da archi tale per cui percorrendo gli archi non è possibile arrivare ad un certo nodo più di una volta (quindi non esistono loop). Le pipelines favoriscono il flusso efficiente dei dati da un punto A ad un punto B e consentono agli sviluppatori di applicare vari filtri e trasformazioni ai dati che passano al proprio interno.

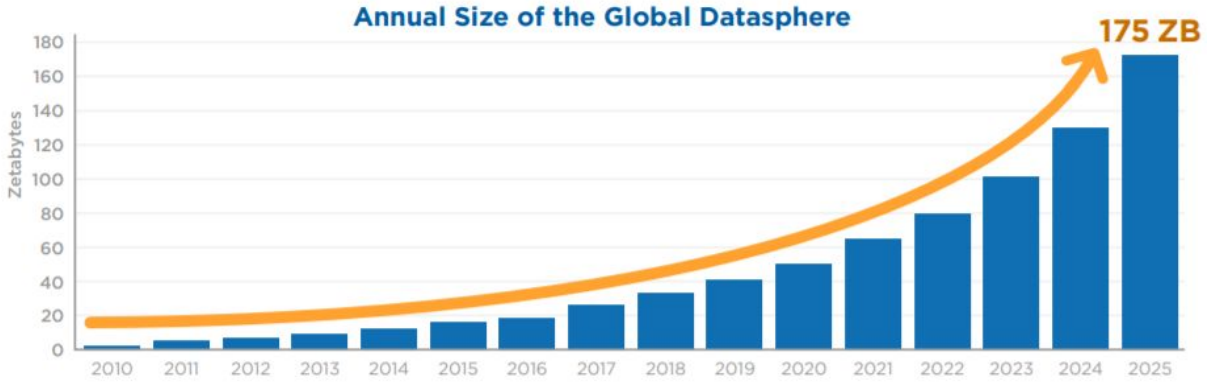
Una delle applicazioni tradizionali delle data pipelines è l’attività nota come *Extract-Transform-Load*, abbreviata in *ETL*, che consiste nell’estrazione di determinate informazioni da una o più fonti e nelle seguenti operazioni di trasformazione (come la conversione in un formato standard, la verifica dell’integrità, aggregazioni e separazioni ed anche complesse funzioni matematiche) al fine di adattarle per il finale inserimento in un qualche tipo di base di dati preposta all’analisi, di solito un *data warehouse* (il cui scopo sarà chiarito più avanti nella trattazione).



Rappresentazione schematica di una data pipeline (Google Cloud, 2020)

Tutto ciò non costituisce in genere un problema nel caso si parli di una quantità di dati regolare, ma è un processo tutt'altro che banale se si considerano i *big data*, termine di cui si sente molto parlare, a volte in maniera non appropriata. I big data, più raramente indicati anche in italiano come *megadati*, sono insiemi di dati la cui dimensione è tale da andare oltre all'abilità di ingestione e manipolazione dei database tradizionali (relazionali e/o monolitici). Attualmente si può indicare una soglia minima dell'ordine dei petabyte, sebbene tale soglia sia irrimediabilmente destinata ad essere superata con il passare del tempo.

IDC, una società multinazionale di ricerca ed analisi del mercato IT, evidenzia infatti in un report del 2018 come il numero di dati prodotti in tutto il mondo (la cosiddetta *datasphere*) sia cresciuto dal 2010 ad oggi in maniera esponenziale, e stima il raggiungimento nel 2025 di un valore di 175 zettabyte a livello mondiale (uno zettabyte è equivalente ad un trilione di gigabyte).



Crescita dei dati prodotti globalmente (IDC, 2018)

La grandezza in termini di bit e byte, inoltre, non è l'unica cosa che conta; si può infatti parlare di quattro aspetti fondamentali che definiscono i big data, chiamati "le quattro V" (IBM, 2018):

- ➔ *Volume*, ovvero la già discussa dimensione dei dati in termini di volume di archiviazione;
- ➔ *Varietà*, cioè il numero di formati differenti assunti dai dati. Considerando ad esempio i social network o l'IoT, i dati accumulati e processati sono delle forme più disparate: video, immagini, audio, testo;

- ➔ *Velocità*, quindi il numero di dati prodotti in un certo lasso di tempo. La velocità dei dati è importante tanto quanto la loro dimensione, e molti sistemi odierni riescono a generare moli di dati in tempi estremamente ridotti, supportati anche da innovazioni della connettività come la fibra ottica e il 5G;
- ➔ *Veracità*, ovvero il grado di accuratezza dei dati raccolti. Con varietà, volumi e velocità elevate, verificare che i dati non siano corrotti, mancanti o impropriamente manipolati può diventare una vera e propria sfida.

I flussi di dati odierni, soprattutto nelle grandi organizzazioni, sono quindi spesso veloci e di grandi dimensioni: è essenziale quindi poter disporre di strumenti affidabili, scalabili e sufficientemente performanti. La soluzione tecnica a questo problema è rappresentata dal *distributed processing*, ovvero la lavorazione dei dati distribuita tra più macchine.

Uno dei framework più popolari per il distributed processing è la piattaforma *Apache Hadoop*, che si serve di uno o più *cluster* (ovvero un gruppo) di vari computer, detti *nodi* in questo contesto. Hadoop suddivide i dati da processare in blocchi e li distribuisce tra i vari nodi di un cluster in modo che essi possano eseguire in parallelo le operazioni richieste dalla data pipeline, aumentando quindi drasticamente il throughput di quest'ultima.

Hadoop permette elevata scalabilità sia verticale sia orizzontale, dato che è possibile aggiungere a piacimento più calcolatori ad un certo cluster così come componenti più performanti (ad esempio CPU a maggiore frequenza di clock), ai nodi esistenti, così come alta disponibilità ed affidabilità, dato che vengono generate e distribuite repliche dei blocchi di dati nel sistema tali che, se anche uno dei nodi nel cluster dovesse improvvisamente andare in crash, i blocchi di dati contenuti in quel nodo sarebbero comunque reperibili dagli altri nodi ancora online, risultando quindi semplicemente in un rallentamento della pipeline e non in un arresto o in una perdita di dati.

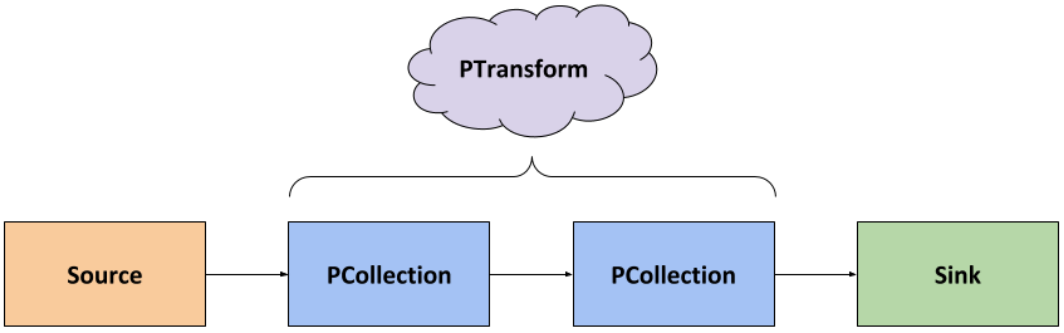
Si deve inoltre specificare che il trattamento dei dati in una pipeline può avvenire secondo due modalità principali: in *lotti* (*batch*) e in *tempo reale*, anche detto in *flusso* (*stream*). Il trattamento in lotti è tipico dei processi ETL e prevede, per ragioni di efficienza, l'esecuzione delle operazioni previste nella pipeline solo in determinati momenti, creando quindi una

lottizzazione dei dati. Al contrario, il trattamento in flusso, tipico ad esempio nei sistemi IoT e per l'analisi dei dati finanziari in tempo reale, comporta una continua esecuzione delle operazioni di trasformazione, che iniziano non appena nuovi dati confluiscono alla fonte della pipeline.

Hadoop è stato pensato principalmente per gestire processi di tipo batch. In contesti di gestione di flussi dati in tempo reale, si ricorre ad altre piattaforme software: tra le più popolari si possono citare *Apache Storm*, *Apache Spark* ed *Apache Beam*.

Apache Beam è un recente modello di programmazione *open-source* (ovvero il cui codice sorgente è disponibile al pubblico ed a cui chiunque può liberamente contribuire) che permette di implementare con relativa facilità pipeline di dati sia in lotti sia in flusso utilizzando una varietà di linguaggi di programmazione differenti.

In Apache Beam i dati sono organizzati in gruppi detti *PCollection*, insiemi che possono avere sia dei limiti di dimensione ben definiti, se si sta creando una pipeline batch, sia essere illimitati ed in continuo aggiornamento in caso di pipeline stream. Alle PCollection vengono applicate operazioni, chiamate *PTransform*, di varia natura: una qualsiasi PTransform modifica una PCollection in entrata e restituisce una nuova PCollection rielaborata.



Schematizzazione di una pipeline in Apache Beam

Servizi cloud

I servizi cloud offerti dagli hyperscalers sono quasi onnicomprensivi, e nuovi servizi vengono aggiunti costantemente, più volte all'anno. Pertanto, una descrizione completa di ognuno di essi esula dai fini di questo documento. In ogni caso, una spiegazione dei servizi principali offerti è di fondamentale importanza per comprendere appieno come il cloud computing possa effettivamente rispondere alle necessità informatiche delle aziende odierne.

Le descrizioni di seguito riguarderanno prevalentemente prodotti Google Cloud in considerazione della maggiore familiarità derivante dall'esperienza di tirocinio compiuta, tuttavia le caratteristiche e le spiegazioni che verranno fornite potranno in larga parte essere facilmente applicate a contesti di altri fornitori cloud, come Amazon Web Services e Microsoft Azure.

Tipi di cloud

Occorre innanzitutto specificare che esistono diversi tipi di cloud, ognuno con caratteristiche ed implicazioni ben diverse.

La dicitura *public cloud*, di cui più spesso si sente parlare, specifica che l'infrastruttura informatica è commercialmente disponibile al pubblico, offerta da un qualche tipo di fornitore cloud. L'aggettivo "public", "pubblico", al contrario di ciò che si possa quindi pensare ad un primo sguardo, non significa dunque che le informazioni salvate in questo tipo di cloud sono disponibili a chiunque: al contrario, le risorse messe a disposizione sono strettamente segregate da cliente a cliente, ed ogni cliente può specificare precise regole di accesso granulari ad ogni genere di risorsa offerta.

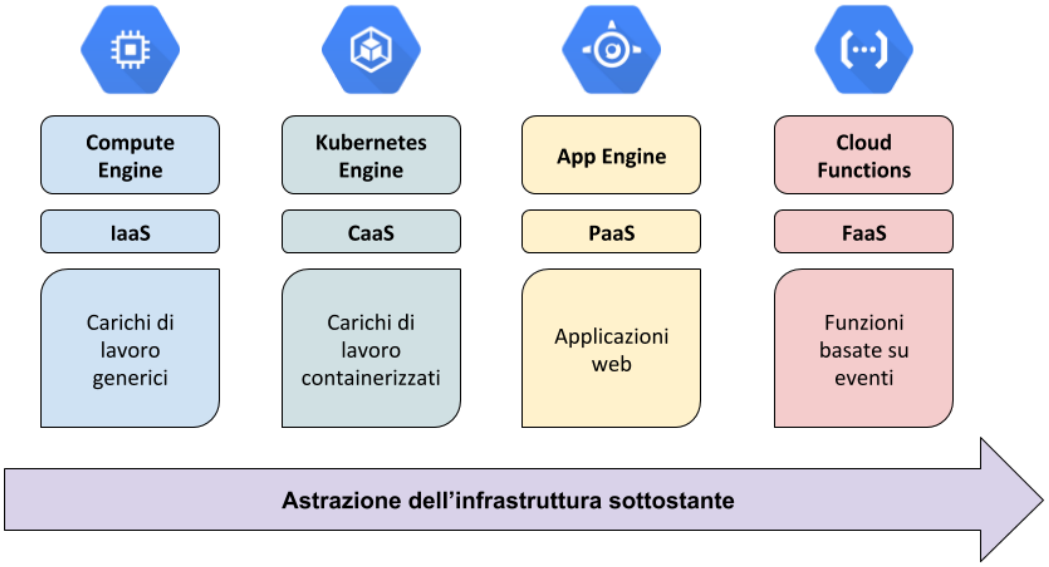
Concettualmente opposta è la *private cloud*, che identifica, come si può intuire, un'architettura cloud ad uso esclusivo e di proprietà esclusiva di una singola organizzazione. Questo tipo di cloud è in genere adottato internamente da grandi società che possono permettersi un importante investimento iniziale ed il successivo reclutamento ed impiego di personale tecnico qualificato per le attività di manutenzione ed aggiornamento del sistema cloud.

Tra i due paradigmi abbiamo la *hybrid cloud*, combinazione ibrida, per l'appunto, di sistemi informatici appartenenti sia a cloud privati, sia a cloud pubblici: sono necessari, in tal caso, interfacce di raccordo che permettano lo scambio di informazioni ed applicazioni tra le diverse "nuvole". L'ambito hybrid cloud è in notevole sviluppo e sembra rappresentare un ottimo compromesso per aziende, in genere, di grandi dimensioni.

In ultimo, si può anche sentir parlare di *community clouds*, ovvero infrastrutture cloud operate, condivise e controllate da un gruppo di aziende aventi interessi comuni; una community cloud permette la condivisione e quindi riduzione dei costi di investimento infrastrutturale rispetto ad una private cloud, sebbene questi costi risultino in genere maggiori rispetto a quelli che si avrebbero diventando clienti di un fornitore di una public cloud.

Servizi di elaborazione

I servizi di elaborazione (o *computing*) rappresentano il nucleo, il cuore pulsante di ogni fornitore cloud classificabile come hyperscaler. Lo scopo di questi servizi è, sostanzialmente, offrire potenza di calcolo e risorse dedicate all'elaborazione elettronica. Quattro sono i principali servizi di computing offerti da Google Cloud: Compute Engine, Kubernetes Engine, App Engine e Cloud Functions.



Schema delle categorie dei servizi di elaborazione

Tali servizi sono sommariamente catalogabili secondo uno spettro che va dalla minima alla massima astrazione delle operazioni di gestione e manutenzione delle risorse e delle infrastrutture informatiche. Si procederà ora ad una spiegazione dettagliata delle loro funzionalità.

Servizi di infrastruttura: IaaS

Sebbene, come già accennato, innovazioni nel campo del *deployment* (il "dispiegamento in produzione" delle applicazioni) e della virtualizzazione abbiano portato a soluzioni come i container o come le piattaforme *serverless*, molto più comode e flessibili, in molti casi, rispetto alle tradizionali macchine virtuali, queste ultime rappresentano ancora un tassello fondamentale per molte aziende, specialmente per le attività di migrazione senza modifiche (o quasi) al codice sorgente di workload da on-prem al cloud, attività dette di *lift-and-shift*,

ovvero “alza-e-sposta”. Ad esempio vecchi programmi informatici, critici per molte aziende ed operanti interamente in server farm private, si prestano bene ad essere “alzati e spostati” nel cloud, liberando le aziende da sforzi di manutenzione sempre più onerosi. Gli stessi programmi potrebbero non essere trasferibili tramite altri paradigmi, come i servizi di piattaforma descritti più avanti. In generale, quando è richiesta la massima libertà possibile in termini di configurazione infrastrutturale e si vuole avere controllo nei minimi dettagli sulle macchine da dispiegare, un servizio di tipo IaaS rappresenta la soluzione più efficace.

Un grande vantaggio delle offerte IaaS è dato dalla trasformazione nel modo in cui viene vista e trattata l’infrastruttura all’interno di un’azienda. Tradizionalmente, l’infrastruttura informatica è un qualcosa di estremamente concreto: enormi complessi industriali, rastrelliere di dischi e processori, cablaggi, interfacce di rete, costante installazione e aggiornamento di software, circuiti di videosorveglianza, impianti di raffreddamento. Modificare caratteristiche infrastrutturali richiede un tempo spesso non indifferente nonché interventi fisici *in loco*.

Tramite il cloud computing, ciò che è stato appena descritto viene completamente *astratto*: dato che la gestione tecnica diretta dei server e dei data center non è più di competenza dell’azienda cliente, bensì viene delegata al fornitore cloud, modifiche infrastrutturali più o meno complesse possono essere effettuate tramite documenti scritti in codice markup, che vengono interpretati ed eseguiti automaticamente dalle piattaforme dei fornitori cloud. Il codice di questi documenti può essere sia proprietario, e quindi funzionare esclusivamente per un particolare fornitore cloud, sia *cross-platform*, ovvero utilizzabile per più provider cloud, come ad esempio il codice della piattaforma **“**Terraform”.

```
variable "base_network_cidr" {
  default = "10.0.0.0/8"
}

resource "google_compute_network" "example" {
  name                    = "test-network"
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "example" {
  count = 4

  name          = "test-subnetwork"
  ip_cidr_range = cidrsubnet(var.base_network_cidr, 4, count.index)
  region        = "us-central1"
  network       = google_compute_network.custom-test.self_link
}
```

Esempio di codice di configurazione infrastrutturale con Terraform

Le modifiche avvengono in un lasso di tempo dell'ordine di pochi minuti e non c'è alcuna necessità di intervenire fisicamente. Forse ancor più importante è il fatto che le risorse dispiegate in questo modo siano altrettanto facilmente eliminabili; la creazione e l'eliminazione di risorse, inoltre, non comporta spese: i costi sono calcolati esclusivamente in base al tempo di utilizzo.

In base a queste caratteristiche, è facile notare come l'infrastruttura, astratta ad un tale livello, assuma caratteristiche ascrivibili a puro codice informatico: per tale ragione si parla di *Infrastructure-as-Code*, o *IaC*. Tramite l'IaC, operazioni come la creazione e lo smantellamento di intere infrastrutture informatiche possono essere facilmente automatizzati e replicabili in pochi click.

Il servizio IaaS offerto da Google cloud è chiamato *Compute Engine*; uno dei punti di forza di tale servizio (così come offerte simili dei competitors, ad esempio *EC2* di Amazon Web Services) è dato dalla possibilità per i clienti di scegliere tra una grande varietà di opzioni relative alle prestazioni dei componenti come CPU e GPU (processore grafico) ed alla capienza di memoria e dischi fissi. Questo fa sì che i clienti possano adattare l'infrastruttura noleggiata nel cloud in base all'effettivo carico di lavoro richiesto. Per esempio, configurazioni che prediligono la memoria RAM sono ideali per operazioni di *data*

warehousing, mentre complesse simulazioni matematiche richiedono CPU adeguatamente performanti.

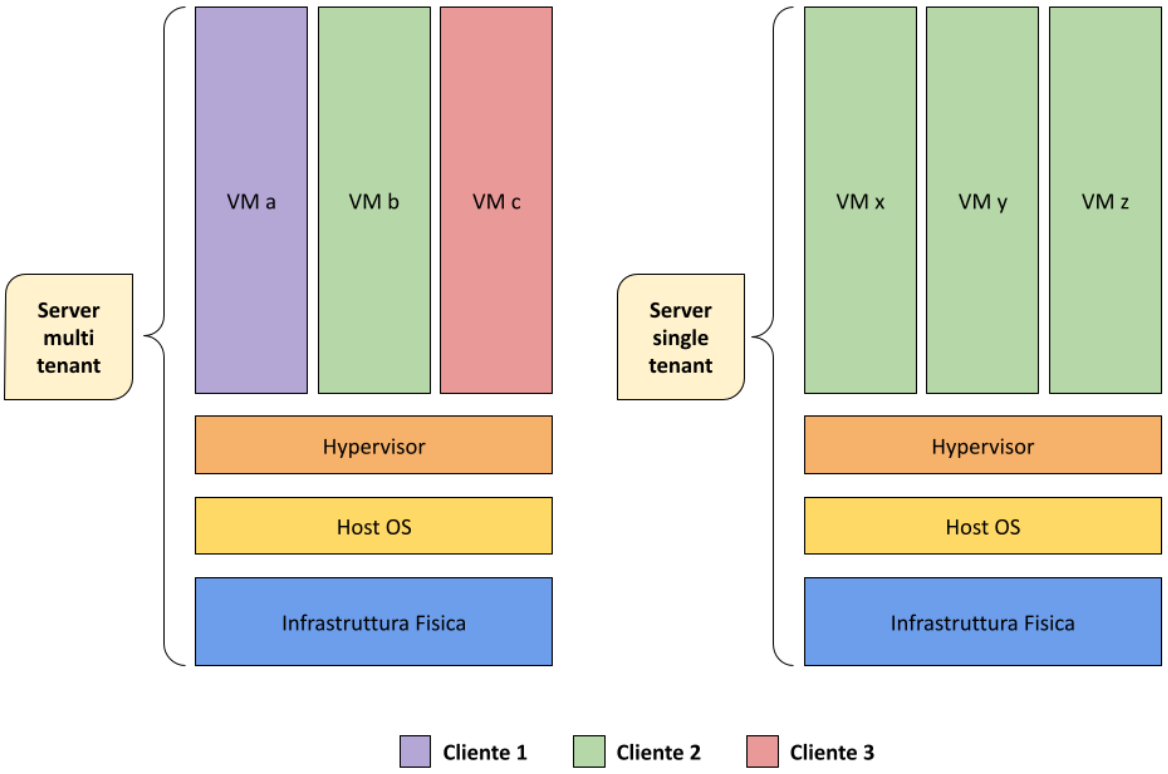
Qualunque sia la necessità, le risorse affittate, una volta esaurito il loro scopo, possono essere in qualunque momento rilasciate senza alcun costo per il cliente: le macchine virtuali nel cloud comportano infatti costi solo mentre sono attive.

Come spiegato all'interno del capitolo sulla virtualizzazione, le macchine virtuali sono alla base del modello di business dei provider di servizi cloud, poiché permettono di scomporre dinamicamente un unico computer fisico in molteplici computer virtuali, per cui macchine virtuali appartenenti a clienti diversi, pur se isolate tra di loro a livello software, potrebbero trovarsi allocate all'interno di un unico server fisico in un data center.

In alcuni casi, sebbene i dati siano sempre crittografati da qualsiasi provider cloud, questa "convivenza" tra clienti potrebbe risultare problematica:

- ➔ Regolamenti e leggi industriali o nazionali, ad esempio leggi sulla privacy dei dati sanitari, potrebbero richiedere un isolamento fisico oltre che virtuale.
- ➔ Carichi di lavoro intensi, propri ad esempio dei videogiochi, potrebbero beneficiare in termini di performance dall'avere accesso esclusivo all'hardware alla base delle macchine virtuali.

Per tali ragioni, vari fornitori di servizi IaaS, tra cui Google Cloud e AWS, offrono la possibilità di noleggiare un accesso esclusivo ad uno o più server, detti *single tenant*, ovvero ad *affittuario singolo*, in modo tale da assicurare un totale isolamento fisico dei dati del cliente.



Confronto tra server standard e server single tenant

Qualità come la scalabilità e l'alta disponibilità in Compute Engine (e nelle offerte parallele dei concorrenti) sono garantite tramite la creazione di cluster di macchine virtuali identiche (dette *istanze*), a cui Google Cloud si occupa di fornire servizi come scalabilità automatica, load balancing, update automatici, monitoraggio continuo e creazione automatica di nuove VM in caso di crash o arresto improvviso delle macchine presenti nel cluster.

Si può infine parlare di una categoria particolare di macchine virtuali, dette *preemptible VMs* in Google Cloud (chiamate anche *low priority VMs* in Microsoft Azure e *EC2 spot instances* in AWS), che permette ai fornitori di sfruttare dinamicamente la capacità in eccesso di risorse informatiche nei propri data center, che andrebbero altrimenti sprecate.

Una preemptible VM è infatti un tipo di macchina virtuale il cui costo per il cliente è circa dell'80% inferiore rispetto ad una regolare macchina virtuale; il rovescio della medaglia è che, essendo tale VM basata su capacità temporaneamente in eccesso nei data center, questa può essere in qualunque momento, e con pochi secondi di preavviso, terminata dal

fornitore cloud nel momento in cui tale capacità sia necessaria per gestire carichi di lavoro di priorità maggiore, come ad esempio le VM regolari.

Un caso d'uso interessante per queste particolari macchine virtuali è la creazione e gestione a bassissimo costo di un cluster di distributed processing in lotti per i dati in una pipeline. Infatti, anche se una delle VM usate per questo scopo dovesse venire terminata dal fornitore cloud, il trattamento dei dati verrebbe semplicemente rallentato ma non si arresterebbe (come si è avuto modo di osservare per il framework Apache Hadoop). Chiaramente, è richiesta esperienza da parte del cliente nell'implementazione di una simile architettura.

Servizi di piattaforma: PaaS

Come già accennato, i servizi di tipo PaaS (*Platform-as-a-Service*) permettono di astrarre ciò che riguarda la gestione dei dettagli tecnici dell'implementazione e della manutenzione di un'applicazione web, ad esempio la dimensione della RAM e la potenza della CPU usata dal calcolatore, il runtime, il sistema operativo, i middleware, i servizi di rete e così via. E' facile osservare come tali semplificazioni riducano drasticamente il tempo necessario al passaggio dalla scrittura del codice dell'applicazione alla distribuzione di tale applicazione su Internet. I servizi di piattaforma forniscono inoltre scalabilità automatica e pressoché illimitata nonché soluzioni di monitoraggio delle applicazioni e logging degli eventi.

Si tende a descrivere i PaaS come offerte *serverless*, ovvero "senza server": infatti, sebbene dei server vengano ovviamente utilizzati dai provider cloud per mantenere e distribuire online il codice applicazione, il cliente non deve preoccuparsi di gestirli direttamente. Al contrario, il cliente semplicemente scrive le necessarie configurazioni in un file di markup, effettua l'upload di tale file congiuntamente al codice della sua applicazione, ed il sistema PaaS provvede a tutti i dettagli infrastrutturali che permetteranno di distribuire l'applicazione online secondo le specifiche progettuali richieste.

Il servizio PaaS offerto da Google Cloud è chiamato *App Engine*; la configurazione degli aspetti essenziali del servizio avviene tramite un file chiamato app.yaml (yaml è un popolare formato di markup). Nel suddetto file, i clienti possono specificare il runtime relativo al

linguaggio di programmazione scelto (Python, Node.js, PHP...), percorsi relativi a cartelle e file, parametri di scaling, impostazioni di caching, variabili di ambiente ed altro ancora. Altri dettagli, come il controllo di versione e le impostazioni del firewall, possono essere gestiti direttamente dal cruscotto web di controllo di Google Cloud.

```yaml
runtime: nodejs12

manual_scaling:
  instances: 1

resources:
  cpu: .5
  memory_gb: 0.5
  disk_size_gb: 10

handlers:
  - url: /
    static_files: build/index.html
    upload: build/index.html
  - url: /
    static_dir: build/
```

Esempio di file di configurazione app.yaml

In generale, le applicazioni distribuite tramite App Engine dovrebbero essere stateless. Le informazioni che devono necessariamente permanere possono essere conservate in istanze di cache o di database al di fuori di App Engine. Inoltre, nel caso siano richieste configurazioni specifiche dell'infrastruttura sottostante all'applicazione, come utilizzare un determinato sistema operativo, è più indicato usare un servizio in cui tali opzioni sono disponibili all'utente, come il precedentemente descritto Compute Engine.

Servizi per container: CaaS

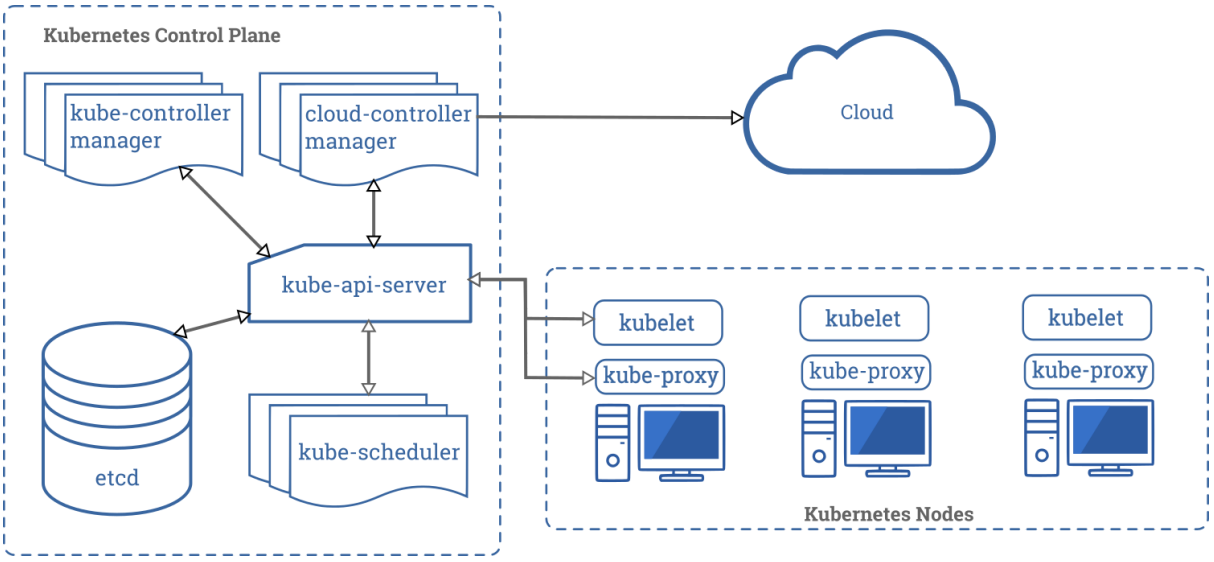
La branca di offerte cloud denominata *Container-as-a-Service* si occupa, come si può intuire, di offrire al cliente la possibilità di creare e gestire un'infrastruttura informatica basata sui container. Le offerte CaaS sono inquadrabili come una "via di mezzo" tra la flessibilità propria delle soluzioni IaaS e la propensione alla scalabilità che si riscontra nel paradigma PaaS.

Lo standard industriale per la gestione, o più propriamente *orchestrazione*, dei container è *Kubernetes* (tra le altre soluzioni presenti sul mercato, si possono menzionare *OpenShift* e *Docker Swarm*). Kubernetes, il cui nome significa timoniere o pilota in greco, è una piattaforma sviluppata internamente da Google e rilasciata come software open-source nel 2014 (The Kubernetes Authors, 2020).

L'orchestrazione dei container permette di automatizzare operazioni fondamentali per una qualsiasi applicazione web, in particolare:

- ➔ scalabilità orizzontale automatica: Kubernetes può essere impostato in modo da aumentare e ridurre elasticamente il numero dei *pod* (la più piccola e semplice unità di elaborazione gestita da Kubernetes, che può contenere uno o più container) dell'applicazione in base a misurazioni come, ad esempio, l'utilizzo di CPU;
- ➔ ottimizzazione e bilanciamento del carico: il traffico web, ad esempio, viene distribuito tra i vari pod in modo da garantire la stabilità complessiva del sistema;
- ➔ gestione dichiarativa dei pod: partendo da un certo pod, si può specificare in maniera dichiarativa (seguendo la filosofia Infrastructure-as-Code) quante repliche sono desiderate per tale pod. Un sistema come Kubernetes si occuperà quindi automaticamente di far arrivare il sistema a regime occupandosi della creazione del numero desiderato di repliche;
- ➔ gestione della salute dei pod: la piattaforma di orchestrazione monitora costantemente lo "stato di salute" dei pod, ovvero verifica se rispondano correttamente ai comandi; in caso di crash o blocco di un determinato pod, la piattaforma automaticamente termina quest'ultimo e ne crea uno nuovo, assicurando che il sistema rimanga a regime.

Kubernetes organizza le risorse necessarie per operare applicazioni containerizzate in cluster di calcolatori (di solito macchine virtuali).



Schema dell'architettura di un cluster Kubernetes (The Kubernetes Authors, 2020)

Un cluster Kubernetes è costituito da due tipi diversi di istanze di macchine: i *nodes* ("nodi") e il *control plane* ("piano di controllo"). I nodes sono calcolatori che contengono container ed eseguono le direttive del control plane. Quest'ultimo è responsabile del funzionamento di quattro servizi essenziali per il controllo dei nodes:

- ➔ Il *controller manager*, che si occupa di eseguire vari servizi di gestione dei componenti astratti di Kubernetes.
- ➔ Un *API server*, che gestisce le interazioni tra i nodi e il control plane nonché le interazioni tra cluster diversi.
- ➔ Lo *scheduler*, ovvero il "pianificatore", che si occupa di determinare in quali nodi vadano dispiegati i vari pod (solitamente sotto forma di gruppi di cosiddette *repliche* identiche) nel cluster.
- ➔ *etcd*, un database NoSQL di tipo chiave-valore usato per conservare informazioni riguardo allo stato dell'intero cluster.

Normalmente, il setup e la gestione nel tempo di un cluster è interamente a carico dell'utente. Kubernetes infatti è una piattaforma software gratuita ed open-source, non un servizio. I fornitori cloud offrono per questo motivo delle soluzioni che utilizzano Kubernetes come base, ma aggiungono vari benefici, uno su tutti il fatto di liberare il cliente da molti degli oneri della gestione del software e dell'infrastruttura sottostante.

Google Kubernetes Engine, abbreviato in *GKE*, è la soluzione completamente gestita offerta da Google Cloud per l'amministrazione di cluster Kubernetes e l'orchestrazione di container. Questo servizio offre vari benefici, tra cui si possono citare:

- ➔ Possibilità di creazione di interi cluster Kubernetes con un solo clic.
- ➔ Possibilità di specificare politiche sia di scalabilità orizzontale, sia di scalabilità verticale automatica dei pod in base a metriche customizzate.
- ➔ Monitoraggio integrato automatico.
- ➔ Interfaccia grafica di controllo.

Servizi per funzioni: FaaS

Il modello *FaaS* è simile, apparentemente, al modello PaaS, sebbene debbano essere considerate alcune rilevanti differenze.

FaaS, o *Functions-as-a-Service*, rappresenta un tipo di servizio di elaborazione leggero, asincrono e basato sugli eventi: esso, in pratica, permette di creare funzioni a scopo singolo che rispondono a precisi eventi *trigger* senza la necessità di dispiegare e gestire direttamente server o sistemi di runtime.

Anche il modello FaaS, dunque, si basa sul paradigma serverless. La libertà di scelta dell'utente è però, rispetto al modello PaaS, ancora più limitata: difatti, l'utente dovrà solo scegliere il runtime relativo al linguaggio di programmazione desiderato, la memoria e l'evento trigger che farà partire l'esecuzione della funzione (potrebbe essere necessario, inoltre, specificare altri dettagli minori in base al cloud provider scelto); fatto ciò, l'utente caricherà il codice vero e proprio della funzione ed il servizio verrà configurato e sarà pronto in pochi secondi. In altre parole, le offerte FaaS "nascondono" all'utente criteri relativi alla scalabilità che possono invece essere impostati in maniera dichiarativa, come si è visto in precedenza, utilizzando un modello PaaS. Le funzioni serverless FaaS, dunque, sono più semplici da impostare e gestire rispetto alle controparti PaaS, a discapito della flessibilità di applicazione.

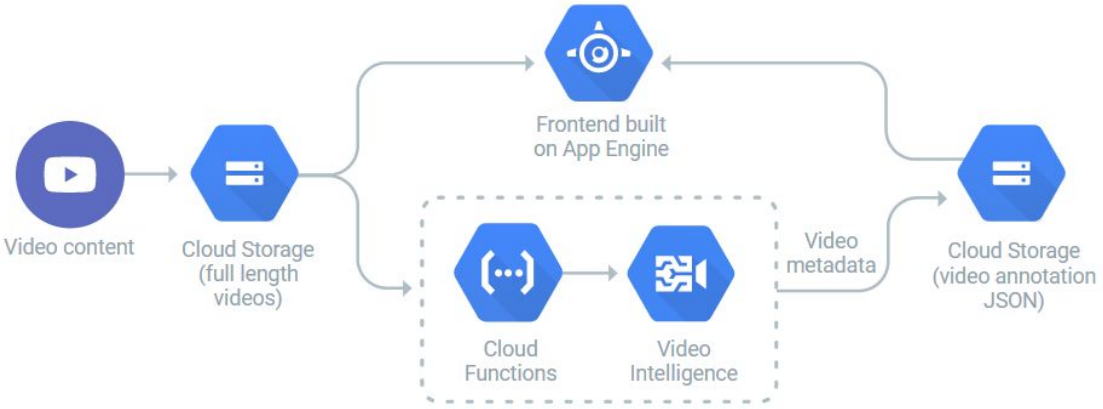
Un esempio di configurazione di una funzione serverless in Google Cloud, che offre FaaS con il nome di *Cloud Functions* (l'attualmente più famoso equivalente offerto da AWS è chiamato invece Lambda), può essere esaminato osservando la figura seguente.

Esempio di configurazione per una Cloud Function (Google Cloud, 2020)

Il modello FaaS è ideale per i casi in cui sia necessario aggiungere particolari funzionalità ad un'applicazione che possano essere attivate in risposta ad un preciso evento e per cui non sia richiesta una disponibilità di calcolo costantemente online ogni giorno, dato che il servizio offerto comporta costi solo quando la funzione sta effettivamente processando il

codice fornito; al contrario, un tipico ambiente PaaS, progettato per gestire maggiore complessità di codice, prevede almeno un'istanza server, o container, che rimane costantemente online, ed a seconda dei criteri di scalabilità nuove istanze vengono automaticamente aggiunte in base al traffico ricevuto.

Vediamo un esempio di architettura che sfrutta una funzione serverless:



Esempio di architettura che sfrutta FaaS (Google Cloud, 2020)

In questa applicazione web, dei video vengono caricati online in Cloud Storage, un servizio di archiviazione di dati eterogenei di cui si discuterà a breve. I video così caricati sono resi fruibili dall'utente finale tramite App Engine, il PaaS che gestisce l'interfaccia utente dell'applicazione. Allo stesso tempo, l'evento "video caricato in Cloud Storage" fa scattare una Cloud Function, il cui codice fa partire un'analisi di intelligenza artificiale (il servizio *Video Intelligence*) sul video stesso: questa analisi restituisce un file di metadati, contenente ad esempio informazioni sulla categoria di contenuti (sport, notizie, cinema...) presenti nel video. A questo punto, la Cloud Function trasferisce il file di metadati in Cloud Storage, e termina la propria esecuzione. I metadati sono prelevati da App Engine, che può così fornire all'utente dell'applicazione una collezione di video completi di informazioni utili per la loro categorizzazione.

Si può notare come, in questa semplice architettura, la soluzione FaaS sia in grado di agire da "colla" tra differenti componenti di un'applicazione, e permetta l'agile ed economica implementazione di logiche in risposta a determinati eventi.

Servizi di gestione dati

I servizi di archiviazione e conservazione di dati offerti dai moderni fornitori cloud si dividono generalmente in due macro-categorie: i servizi per i dati strutturati ed i servizi per i dati non strutturati.

Storage-as-a-Service: STaaS

I dati cosiddetti "non strutturati" sono privi di caratteristiche uniformi e sono, in generale, non catalogabili, etichettabili o interpretabili facilmente dai database tradizionali; esempi di questo genere di dati sono file di immagini o file audio (in generale, file multimediali non testuali). Si è soliti riferirsi a questi dati con il nome di *BLOB*, *Binary Large OBject*, ovvero "oggetti" binari di grandi dimensioni. I BLOB non possono essere inseriti in un database (relazionale, ad esempio) in maniera logica ed appropriata senza l'uso di *metadati*, ovvero dati che descrivono, in maniera coerente, caratteristiche essenziali degli oggetti binari.

E' più efficiente ed efficace, di solito, conservare dati non strutturati tramite servizi di *object storage*: questi ultimi permettono all'utente di caricare BLOB all'interno di cosiddetti *bucket*, ovvero secchi; questo termine descrive efficacemente la mancanza di rigidità strutturale tra i dati caricati online. La struttura a bucket può ricordare un sistema di file e cartelle, come ad esempio il *file system* di Windows, anche se questo paragone è perlopiù erroneo poiché i file system permettono facilmente l'indicizzazione dei vari file e delle varie cartelle; al contrario, i bucket offrono URL specifici ed univoci per ogni BLOB che si decide di caricare, senza capacità di indicizzazione.

Il servizio di BLOB storage offerto da Google Cloud è denominato *Cloud Storage*. E' un servizio completamente gestito: la manutenzione, l'aggiornamento, le misure di ridondanza, la scalabilità e tutte le attività simili sono a carico del fornitore. Oltre ad essere altamente disponibile, un'altra misura importante di un servizio di storage simile è la *durabilità* (o *durability* in inglese) del sistema, che specifica il complementare della probabilità di perdere dei dati conservati all'interno di un qualsiasi bucket in un certo lasso di tempo; per ovvie ragioni, questa probabilità deve essere mantenuta tanto alta quanto possibile: tramite politiche di ridondanza, Cloud Storage (così come servizi simili, ad esempio S3 di AWS) riesce

a raggiungere una durabilità annuale del 99.999999999%, che implica una infinitesima probabilità stimata di perdita di un dato in un anno dello 0,000000001%.

Cloud Storage offre differenti *classi* di archiviazione di dati:

- ➔ *Multi-regional*, progettato per dati per cui è necessario garantire un accesso frequente e rapido in tutto il mondo, ad esempio contenuti multimediali di applicazioni web.
- ➔ *Regional*, per dati ad accesso frequente usati principalmente in una specifica regione geografica servita dai data center di Google Cloud, oppure per dati che, per ragioni legali, devono rimanere confinati entro determinati confini geografici.
- ➔ *Nearline*, per dati ad accesso infrequente (circa una volta al mese), ad esempio dati di backup periodici.
- ➔ *Coldline*, per dati ad accesso estremamente infrequente (circa una volta all'anno), come dati pensati per essere recuperati in caso di disastri o archivi di dati che non sono più utilizzati ma che devono essere conservati per ragioni legali.

In generale, quindi, possiamo distinguere configurazioni che hanno a che vedere con l'ubicazione fisica dei dati e configurazioni basate sulla differente frequenza di accesso richiesta per i dati. Il prezzo richiesto per rendere i dati disponibili globalmente sarà più alto, naturalmente, rispetto al prezzo richiesto per mantenere i dati in un'unica regione; a seconda della frequenza di accesso, invece, il prezzo di mantenimento dei dati online aumenterà ed il prezzo del download dei dati diminuirà o si azzererà al crescere del tasso di scaricamento dei dati richiesto in un certo lasso di tempo. Ad esempio, in Cloud Storage il prezzo di mantenimento di un BLOB nella classe Coldline è relativamente molto basso, mentre il prezzo di download di tale BLOB è relativamente alto, se si comparano i prezzi con l'equivalente offerta Multi-regional. Questo è vantaggioso per il cliente perché il tasso di scaricamento dei dati richiesto per BLOB archiviati in classe Coldline sarà sicuramente basso, mentre la durata di mantenimento online richiesta sarà considerevole.

Un'altra funzionalità da evidenziare è la gestione automatica nel tempo dei file conservati tramite STaaS, denominata *object lifecycle management*: sarebbe impossibile per grandi

quantità di BLOB, infatti, effettuare manualmente operazioni di rimozione od aggiornamento di dati obsoleti.

Una regola di object lifecycle management

Tramite object lifecycle management, l'utente specifica, ad esempio in un file JSON, il tipo di azione da applicare una volta soddisfatta una certa condizione; l'esempio seguente illustra una *policy* secondo cui tutti i file caricati in un certo bucket di Cloud Storage da un mese vengono automaticamente eliminati.

Database-as-a-Service: DBaaS

Il paradigma *DBaaS* prevede offerte di archiviazione ottimizzate per dati strutturati, adatti quindi ad essere inseriti in una base di dati. Le offerte DBaaS sono completamente gestite, ovvero operazioni tecniche come il setup e la configurazione della connettività, l'installazione di aggiornamenti di sicurezza informatica, l'esecuzione periodica di backup, la creazione di repliche, l'incremento di capacità di memorizzazione ed altre operazioni tecniche di questo genere vengono astratte ed automatizzate o sono configurabili dal cliente tramite pochi click (si può parlare, anche qui, di ambiente pressoché serverless).

Un distinguo ulteriore va effettuato a seconda della natura del database stesso: come spiegato alla sezione delle architetture web, infatti, esistono vari tipi di database, ognuno con differenti punti di forza e debolezza a seconda dell'uso richiesto. La linea di

demarcazione principale è quella che separa i database SQL da quelli NoSQL e le offerte di database nel cloud tendono a rispettare questa divisione.

Per quanto riguarda i database SQL, Google Cloud propone due soluzioni: *Cloud SQL* e *Cloud Spanner*.

Cloud SQL (simile ad Aurora per AWS o ad Azure SQL per Microsoft Azure) rappresenta l'offerta più classica di database relazionale. Esso permette agli utenti di utilizzare vari tipi di *relational database management systems* (abbreviato in *RDBMS*); con questo termine si indica il sistema software che permette di creare e gestire basi di dati relazionali: tra i più popolari si possono annoverare MySQL e SQL Server.

La caratteristica distintiva di Cloud SQL (e dei servizi simili offerti dai concorrenti) è la sua natura completamente gestita: ciò assicura qualità importanti come l'alta disponibilità, la scalabilità e l'affidabilità. E' sicuramente possibile adottare un approccio "fai-da-te" e, invece di usare un servizio di database gestito come Cloud SQL, creare istanze di un database relazionale (ad esempio MySQL) su macchine virtuali, che offrono costi di gestione minori; in tal caso, però, l'implementazione tecnica di qualsiasi operazione di espansione o manutenzione viene lasciata interamente al cliente.

Per quanto concerne Cloud Spanner, esso si può definire come una soluzione di database di tipo SQL completamente gestita che permette alta disponibilità, scalabilità orizzontale e coerenza immediata a livello globale; tale soluzione riesce a conciliare queste caratteristiche di solito mutualmente esclusive tramite algoritmi di consenso che sfruttano orologi atomici in ogni data center, per assicurare coerenza immediata dei dati tra i vari nodi del database, senza però necessariamente bloccare (come si è potuto osservare alla sezione "Basi di dati") i server su cui sta avvenendo una scrittura; l'utilizzo degli orologi atomici consente al sistema di gestione del database di determinare in tempo reale l'ordine cronologico corretto di esecuzione delle scritture.

Nel campo delle basi di dati NoSQL, Google Cloud offre *Cloud Firestore*, un'offerta serverless di database orientato ai documenti, capace di scalare orizzontalmente in maniera

automatica. Pur essendo una soluzione di tipo NoSQL, Cloud Firestore offre la possibilità di eseguire query sofisticate, simili a quelle che si potrebbero implementare utilizzando un sistema SQL.

Il servizio supporta inoltre, a discrezione dell'utente, la possibilità di eseguire transazioni ACID, che garantiscono coerenza immediata. Naturalmente, tale possibilità non deve essere abusata, poiché l'utilizzo di transazioni di questo genere per ogni operazione di scrittura e aggiornamento del database annullerebbe la capacità di scalabilità orizzontale e alta disponibilità propria del paradigma NoSQL. Il cliente, nel design della propria architettura cloud, dovrebbe puntare quindi ad un compromesso sensato tra coerenza forte e coerenza finale dei dati.

Esistono altresì database progettati per poter gestire molteplici modelli di stoccaggio dei dati, offrendo supporto per paradigmi NoSQL come il modello a grafo e il modello orientato ai documenti così come il modello SQL relazionale. Un esempio di database del genere è offerto sotto il nome di *Cosmos DB* da Microsoft Azure. Il vantaggio immediato di questa soluzione è quello di riuscire ad evitare la separata amministrazione e l'eventuale interconnessione tra database completamente distinti, incrementando la produttività e riducendo la complessità architetturale del sistema cloud.

Servizi di gestione di big data

Come si è già avuto modo di spiegare nel corso della trattazione, i big data non sono adatti ai database più tradizionali, ragion per cui i fornitori cloud offrono servizi specializzati per ricavare informazioni coerenti dal caos di questi megadati. I dati sono il petrolio del ventunesimo secolo, ed è fondamentale, per le aziende di qualsiasi tipologia, essere equipaggiate con gli strumenti adatti per riuscire a trarne valore.

Il cloud permette l'accesso elastico ed on-demand ad una grande quantità di calcolatori, ed è quindi perfetto in contesti di distributed processing propri delle pipeline impiegate nella movimentazione e trasformazione di big data. Google Cloud offre due servizi principali di questo genere: *Cloud Dataproc* e *Cloud Dataflow*.

Cloud Dataproc è un'offerta per la creazione di cluster di popolari piattaforme software di distributed processing, tra cui Apache Hadoop. Il servizio è completamente gestito, e si ha quindi il grande vantaggio di poter semplicemente specificare i parametri desiderati in un'interfaccia web per poter creare in breve tempo un intero cluster, completo di crittografia per garantire la sicurezza informatica; inoltre, come per la grande maggioranza dei servizi cloud, anche in questo caso il cliente paga solamente in base a quanto usa: una volta esaurito il suo scopo, qualsiasi cluster può essere facilmente eliminato.

Cloud Dataflow è d'altro canto un'offerta focalizzata sulla creazione ed il deployment di pipeline di dati batch e stream secondo il modello di programmazione di Apache Beam. Anche questo servizio è completamente gestito ed astrae inoltre completamente il cliente dall'infrastruttura sottostante (è quindi serverless), permettendo la scalabilità orizzontale automatica a seconda dell'intensità dei carichi di lavoro. Il servizio offre inoltre dei *template* preimpostati per le più comuni operazioni di trasformazione dei dati, permettendo agli utenti di costruire pipeline anche senza una conoscenza tecnica approfondita della programmazione di Apache Beam.

La raccolta massiccia di dati e la loro manipolazione e trasformazione, spesso in tempo reale, è però inutile se non combinata a dei sistemi specializzati di archiviazione ed analisi degli stessi dati. I database di cui si è discusso finora sono in genere inadatti a questi scopi poiché essi sono orientati al cosiddetto *OLTP* o *online transaction processing*, vale a dire che sono ottimizzati per registrare e quindi scrivere molto rapidamente informazioni relative a determinati avvenimenti o ad azioni degli utenti di cui si vuole tenere traccia, come ad esempio gli ordini dei clienti per un sito di ecommerce.

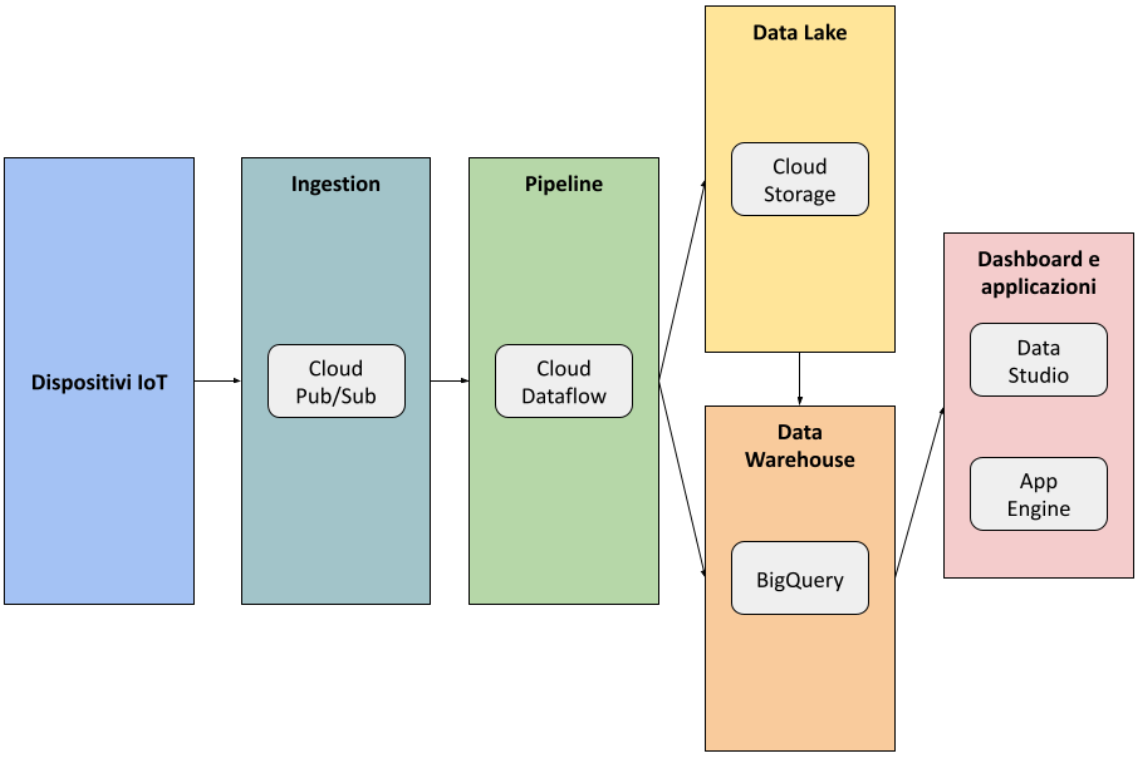
Per scopi di analisi di dati servono altri tipi di basi di dati, orientati alla *OLAP* o *online analytical processing*, il cui punto di forza è la rapida e costante lettura massiccia dei dati in essi memorizzati. Il compito delle già citate pipeline di ETL è infatti trasferire dati da sistemi di tipo OLTP a sistemi di tipo OLAP.

I database OLAP si suddividono in genere in *data warehouse*, termine traducibile come "magazzino di dati", e *data lake* o "lago di dati" (Red Hat, 2020). Un data warehouse riceve

dati accuratamente “puliti” e trasformati in modo da aderire strettamente a precise regole di schema. Al contrario, un data lake accetta dati grezzi e non strutturati, che possono quindi confluire direttamente (e di conseguenza molto velocemente) dalle fonti stesse che hanno generato i dati, senza passare attraverso fasi di trasformazione, che avviene in un secondo momento.

I servizi di gestione di sistemi OLAP nel cloud sono solitamente serverless, permettendo ai clienti di concentrarsi esclusivamente sull’aspetto di analisi ed esplorazione dei dati al fine di raggiungere decisioni di business più informate ed intelligenti. BigQuery è il servizio di data warehousing offerto da Google Cloud, totalmente serverless, altamente disponibile e capace di scalare senza problemi da terabyte ad exabyte di dati. Le informazioni conservate in BigQuery possono essere direttamente analizzate (anche mentre confluiscono attivamente, tramite pipeline, da altre fonti) in un’interfaccia web tramite l’utilizzo del linguaggio SQL.

La trattazione svolta fino a questo punto permette ora di affrontare una completa implementazione architetturale volta all’analisi dei big data, in particolare in un contesto industriale dove si ha la necessità di ricavare dashboard di controllo a partire da sensori IoT.



Schema di un'architettura per analisi di big data

Il flusso di informazioni parte dai dispositivi IoT in un impianto industriale che, tramite connettività wireless, trasmettono costantemente dati di telemetria, come ad esempio misurazioni di temperature o distanze.

Questi dati passano alla fase di *ingestion*, ovvero di *ingestione*, termine con cui si identifica il primo step dell'estrazione di informazioni da un sistema. Tale fase viene gestita da *Cloud Pub/Sub*, soluzione completamente gestita di Google Cloud che offre, come si può intuire dal nome, un servizio di messaggistica Pub/Sub (paradigma affrontato nella sezione precedente del documento) su larga scala. Tale servizio permette di disaccoppiare il tasso di trasmissione delle informazioni (effettuato dai dispositivi IoT) dal tasso di elaborazione di queste ultime, incrementando la resilienza del sistema in caso di picchi di dati in entrata o imprevisti errori hardware/software a valle del flusso di informazioni.

Cloud Pub/Sub, una volta ricevute le informazioni di telemetria, le trasmette a Cloud Dataflow, che a seconda delle necessità effettua una serie di trasformazioni sulle informazioni ricevute sia in tempo reale, sia in lotti.

Successivamente, i dati possono seguire due strade: l'archiviazione in formati eterogenei e poco rielaborati in Cloud Storage, che funge da data lake, oppure il caricamento in BigQuery che, essendo un data warehouse, richiede dei dati rigorosamente strutturati.

A questo punto, i dati conservati nel data lake potranno essere in un secondo momento rielaborati, tramite una pipeline di Dataflow o direttamente sfruttando una delle funzionalità di caricamento di BigQuery, e trasmessi finalmente al data warehouse, mentre i dati ivi già presenti possono essere analizzati tramite varie interrogazioni SQL.

I risultati delle analisi sono finalmente pronti per essere sintetizzati e condivisi tramite apposite soluzioni SaaS di business intelligence, reporting e visualizzazione, che consentono la creazione di dashboard dinamiche ed interattive nel web. Una soluzione di questo genere offerta da Google Cloud è *Data Studio*.

Per maggiore flessibilità nell'utilizzo dei risultati delle analisi è altresì possibile indirizzare questi risultati verso servizi di piattaforma come App Engine, dove del codice personalizzato può sfruttarli come base per effettuare interventi automatizzati nell'impianto in esame, ad esempio inviando un segnale di arresto ad un certo tipo di processo di fabbricazione qualora la temperatura media in un certo lasso di tempo, e per un certo numero di sensori, superi un valore soglia di emergenza.

Strategie di vendita e marketing

Progettare e creare prodotti e servizi di valore, che rispondano alle necessità dei clienti e risolvano i loro problemi, è solo una parte del lavoro necessario per riuscire, come azienda, ad emergere in un qualsiasi mercato, ed il mercato del cloud computing non fa eccezione. Una volta creati, infatti, i prodotti ed i servizi in questione devono essere opportunamente promossi e commercializzati affinché i potenziali clienti sappiano della loro esistenza e, possibilmente, li scelgano perché percepiti come in grado di apportare più valore rispetto ad altre opzioni concorrenti. E' a questo punto che il "testimone" passa dalle funzioni di ingegneria a quelle di marketing e vendita.

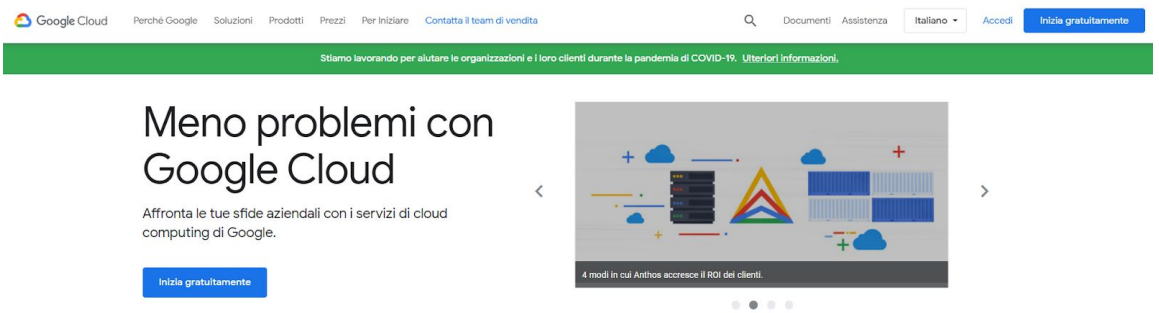
In questa sezione della trattazione verranno spiegate le principali strategie di comunicazione del valore usate nell'ambito del cloud computing, con particolare riguardo alle operazioni di vendita verso grandi aziende.

Lead Generation

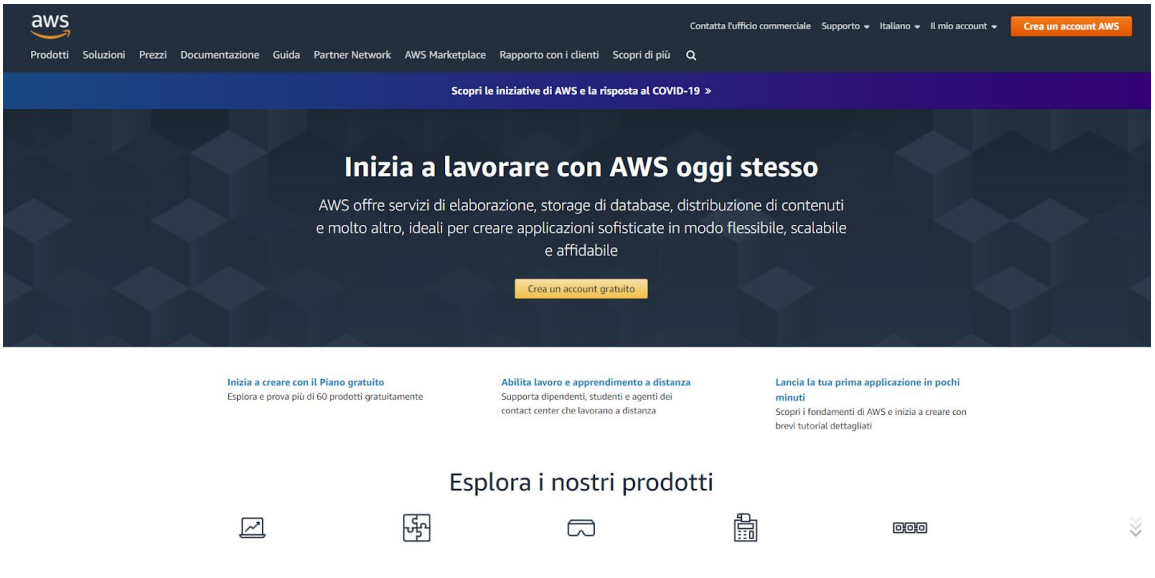
Il percorso che porta le aziende da essere completamente ignare o indifferenti a determinati servizi o prodotti fino a diventare clienti (idealmente soddisfatti) inizia con la *lead generation*. Questo termine indica azioni di marketing volte ad individuare una serie di *leads*, ovvero di potenziali clienti; essa si suddivide in vari aspetti, di seguito elencati e spiegati.

Inbound marketing

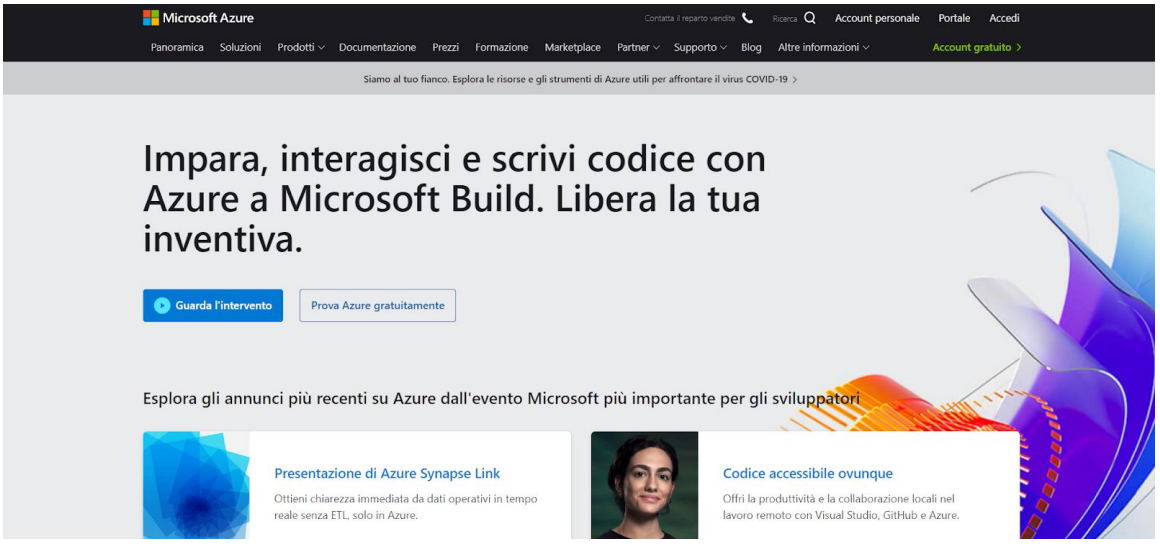
Con *inbound marketing* (marketing "in entrata") si intendono gli sforzi di marketing volti a spingere il cliente verso l'azienda, tramite contenuti e risorse che potrebbero essere ritenuti di interesse dai principali decisori delle aziende con cui si cerca di instaurare un rapporto di clientela. L'inbound marketing è in contrasto con l'*outbound marketing* (marketing "in uscita"), che rappresenta i più tradizionali metodi di "intromissione" nella vita dei clienti, come annunci su Internet, al fine di ottenere la loro attenzione e potenzialmente il loro interesse. L'inbound marketing rappresenta una strategia fondamentale per i principali fornitori cloud, come si può osservare esaminando le loro pagine web dedicate alle offerte cloud, tra di loro molto simili:



cloud.google.com



aws.amazon.com



azure.microsoft.com

Tutti i provider elencati, così come molti altri in questa sede non considerati, mostrano elementi come l'invito a contattare direttamente il reparto vendite, l'offerta di prove gratuite dei propri servizi e la presentazione di soluzioni strategiche, tra cui spiccano in prima posizione quelle per la pandemia di COVID-19.

Altri esempi di inbound marketing sono:

- ➔ I *blog* delle aziende, dove vengono rilasciate le principali novità relative alle piattaforme cloud.
- ➔ I *forum*, che consentono un'interazione diretta utente-impiegato nonché utente-utente, rendendo più facile la risoluzione di problemi e curiosità relative all'ambito cloud.
- ➔ Le *newsletter*, ovvero novità e promozioni recapitate regolarmente in forma di posta elettronica agli iscritti. Le iscrizioni vengono raccolte (previo consenso degli interessati) in vari modi, sia tramite form nei portali online dei fornitori cloud, sia attraverso i recapiti forniti dai partecipanti a conferenze tecniche cloud organizzate dalle aziende stesse.
- ➔ I *podcast*, che offrono contenuti simili ai blog ma in formato audio e con un'atmosfera più rilassata e colloquiale, assumendo sfumature di *infotainment*.
- ➔ I cosiddetti *white paper*, in italiano *libri bianchi*, trattazioni dettagliate delle tecnologie utilizzate per sviluppare ed alimentare prodotti e servizi cloud. I libri bianchi possono essere scritti dai dipendenti stessi dell'azienda o da analisti esterni e partner commerciali. Essi sono, per loro natura, destinati ad un pubblico ristretto di personale tecnico del settore.
- ➔ Le *ristampe*, intese come report offerti da aziende di ricerca specializzate come Forrester e Gartner, le cui licenze di riproduzione e distribuzione vengono acquisite dai provider cloud in modo da offrire tali report gratuitamente agli avventori dei propri siti web. A differenza degli whitepaper, le ristampe si concentrano di solito su analisi di mercato e benefici economici delle soluzioni cloud.
- ➔ I *case studies*, cioè relazioni e report che dettagliano l'impatto positivo riscontrato dai clienti in seguito all'adozione di determinati prodotti o servizi; è fondamentale raccogliere e sfruttare le testimonianze dei clienti soddisfatti in modo da sfruttare la cosiddetta *social proof*, un tipo di fenomeno sociale e psicologico per cui le persone

tendono a copiare le azioni degli altri, specialmente in contesti di incertezza decisionale e se gli "altri" vengono riconosciuti come esperti. La testimonianza favorevole di un cliente di notevoli dimensioni influenza quindi positivamente verso l'acquisto altre aziende di dimensioni simili o inferiori.

cloud.google.com/solutions

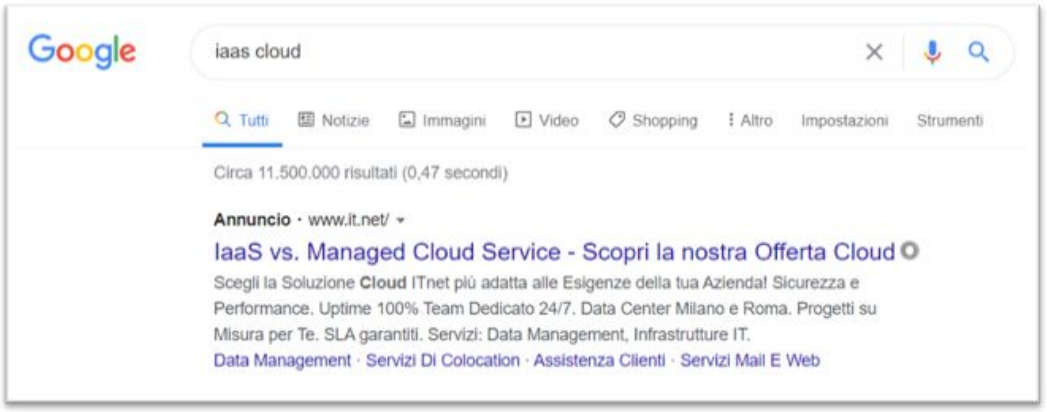
Outbound marketing

L'outbound marketing o marketing in uscita, come già accennato, rappresenta l'approccio più tradizionale e diretto di promozione di prodotti e servizi, in cui è l'azienda che sta cercando di vendere qualcosa a spingere informazioni e messaggi verso i potenziali clienti. Questo tipo di marketing è anche chiamato marketing "di interruzione", poiché richiede l'attenzione dei destinatari interrompendoli, appunto, mentre stanno portando avanti altre attività.

La forma più classica di outbound marketing è l'annuncio pubblicitario, che può essere trasmesso attraverso vari media: si va dai più classici, come radio e televisione, in grado generalmente di raggiungere un pubblico molto ampio ma anche molto eterogeneo, limitando quindi le possibilità che l'audience sia effettivamente ricettiva ed interessata al prodotto o servizio proposto.

Per raggiungere segmenti di pubblico più specifici, si tendono a preferire le pubblicità erogate via Internet, tramite social media come ad esempio Facebook e Twitter o motori di ricerca come Google, che permettono di creare campagne pubblicitarie tali da veicolare gli

annunci solo a persone i cui interessi (desunti dall'attività di navigazione o di ricerca) siano allineati con ciò che l'azienda promotrice sta cercando di vendere.

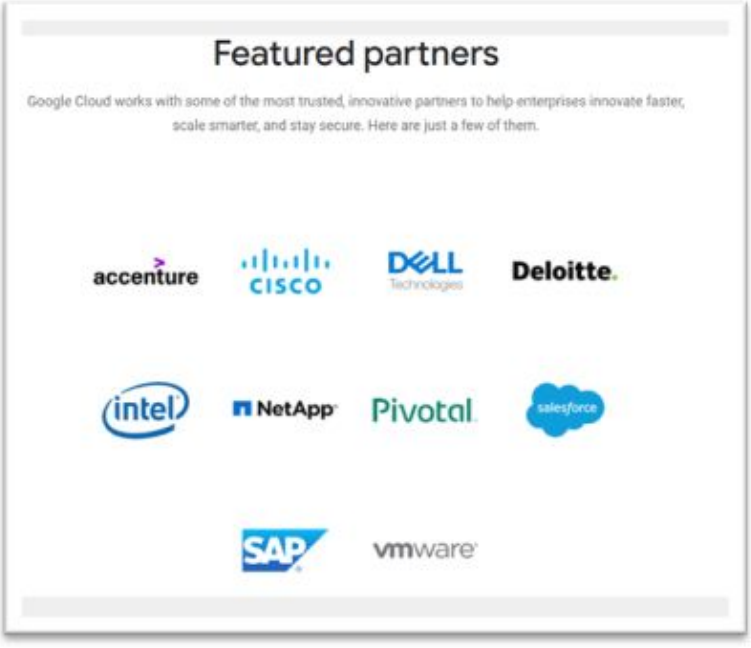


Un esempio di annuncio digitale per la promozione di un servizio cloud

Partnerships

Avere partner prestigiosi e riconosciuti internazionalmente aiuta l'immagine del provider cloud e assicura, allo stesso tempo, un supporto di prima classe nel guidare le aziende lungo il loro percorso di modernizzazione delle infrastrutture informatiche.

Google Cloud, ad esempio, collabora assieme a svariati partner che coadiuvano i clienti sia dal lato dell'implementazione tecnica, sia dal lato della consulenza manageriale.



cloud.google.com/consulting

Referrals

E' famoso l'adagio secondo cui i clienti dovrebbero, in un'azienda sana, contribuire a metà dello sforzo di vendita. E' fondamentale infatti favorire il passaparola tra attuali clienti e potenziali clienti e fare in modo che i propri prodotti vengano "evangelizzati" dai clienti più entusiasti, che possono e dovrebbero diventare importanti sostenitori dei servizi offerti. I potenziali clienti che vengono spinti a provare le offerte di un'azienda dai clienti attuali di tale azienda sono chiamati *referral*, ovvero i "riferiti" da altri.

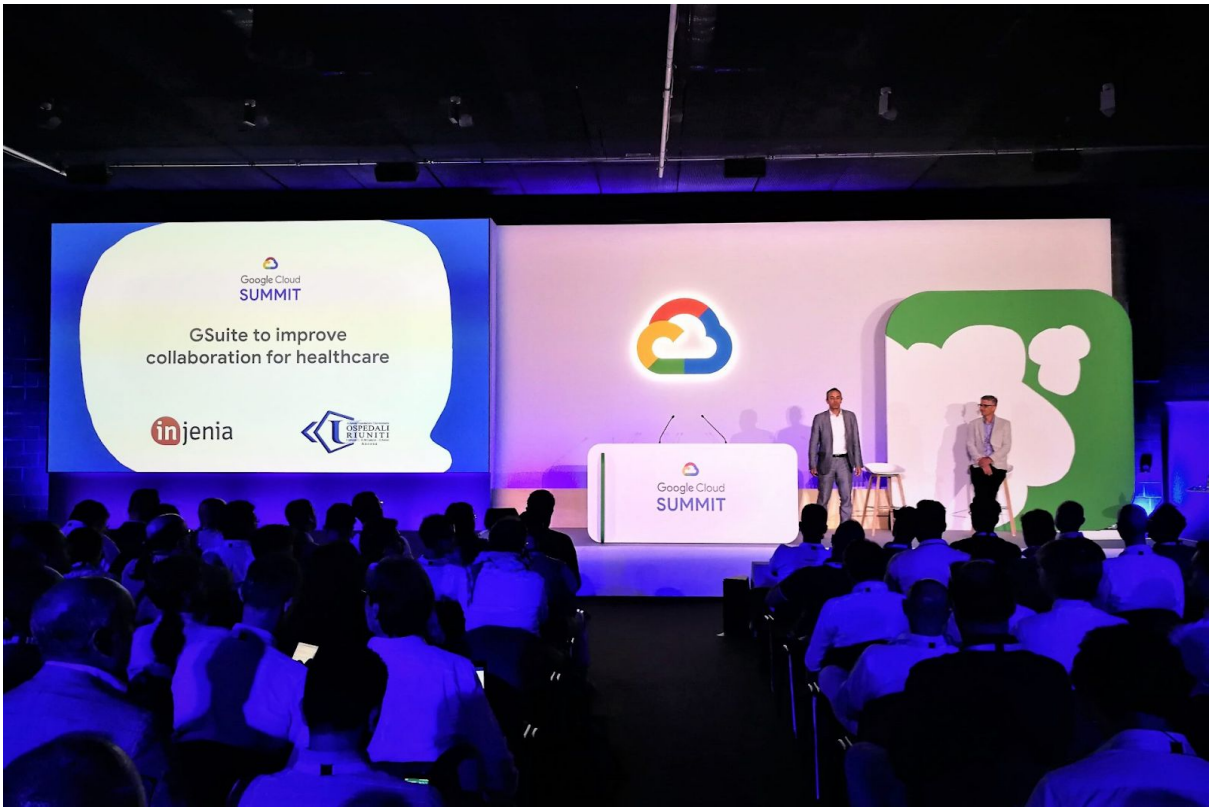
Uno studio del 2014 riportato dall'azienda di ricerca di mercato eMarketer evidenzia come nel mercato B2B i referral siano statisticamente molto più propensi alla conversione in clienti a tutti gli effetti rispetto alle leads ottenute da canali come i siti web aziendali, le fiere commerciali o la pubblicità.

Il meccanismo alla base della grande efficacia del sistema dei referral è la fiducia: l'invito a provare nuovi servizi non proviene infatti da sconosciuti, ma da persone che hanno già avuto modo di conoscersi e lavorare insieme, creando rapporti di mutuo rispetto. Inoltre, è più probabile che i referral abbiano realmente bisogno dei prodotti o servizi offerti dall'azienda venditrice, poiché chi li ha riferiti è in genere informato riguardo ai loro potenziali problemi ed alle loro necessità.

Eventi

Nell'industria del cloud, gli eventi si possono considerare un'attività di marketing fondamentale, poiché rappresentano un'occasione preziosissima di networking e di ingaggio diretto con decisori e dirigenti d'azienda di clienti di livello enterprise; tutti i grandi fornitori cloud organizzano eventi e conferenze con cadenza almeno annuale per varie macroregioni o nazioni in tutto il mondo. Amazon organizza ad esempio gli *AWS Summit*, come Google organizza i *Google Cloud Summit*.

Gli eventi sono di solito suddivisi in vari padiglioni e sale conferenze dove vengono tenute le cosiddette *breakout sessions*, ovvero sessioni tematiche in cui vengono presentati casi di studio e implementazioni di architetture cloud.



Una breakout session al Google Cloud Summit di Milano (Injenia, 2019)

Gli speaker provengono dal pool dei tecnici del fornitore cloud, a cui si affiancano tecnici di clienti enterprise che offrono le proprie dirette testimonianze di come il cloud abbia contribuito al raggiungimento dei propri obiettivi strategici.

Attualmente, la pandemia di COVID-19 ha portato ad un trasferimento interamente in remoto di questi eventi.

Il sales funnel

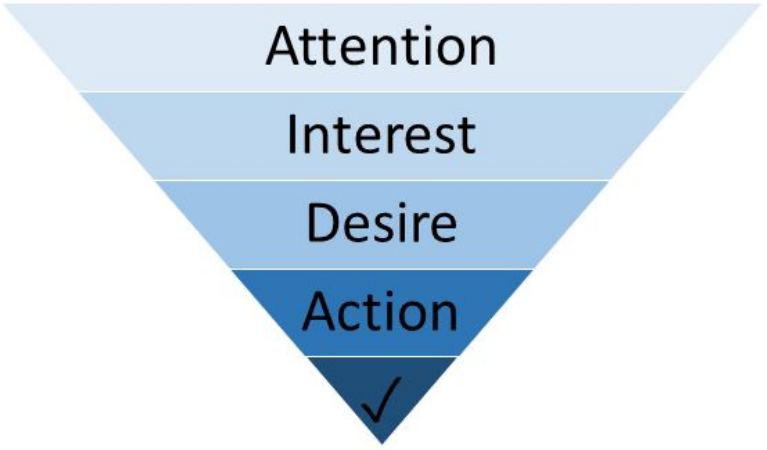
Un concetto basilare nell'ambito delle vendite e del marketing è il cosiddetto *sales funnel*, in italiano traducibile come "imbuto delle vendite". Esso rappresenta il percorso ideale delle aziende che si trovano in un certo mercato, che seguono un cammino di transizione da non clienti a clienti, dall'ignoranza dell'esistenza di un determinato servizio o prodotto fino all'acquisto dello stesso. La forma ad imbuto è usata per specificare che da un certo mercato, rappresentante la totalità dei potenziali clienti raggiungibili (e quindi la parte larga dell'imbuto), non si potrà ricavare che una frazione di effettivi clienti.

Il sales funnel è utile principalmente per due ragioni:

- ➔ Innanzitutto, aiuta i team di vendite a prioritizzare ed a concentrarsi sulle leads più promettenti ed a non sprecare il loro tempo nel perseguire potenziali clienti che, per un motivo piuttosto che un altro, non costituiscono un buon *fit* per l'azienda.
- ➔ In secondo luogo, aiuta a proporre contenuti e materiali sempre rilevanti alle leads, mano a mano che la loro conoscenza dei prodotti e servizi offerti dall'azienda venditrice aumenta, così come aumenta la loro propensione all'acquisto. Questo tempismo è fondamentale per evitare, ad esempio, un approccio di vendita troppo diretto a leads che ancora conoscono solo approssimativamente i prodotti e servizi offerti.

Il sales funnel è diviso in varie fasi, utili per demarcare degli step intermedi di transizione dal primo contatto con una lead fino alla transizione finale della stessa a cliente vero e proprio. Le fasi specifiche del funnel variano da azienda ad azienda, sebbene si possa identificare un modello tradizionale, noto con l'acronimo di *AIDA* (Doyle, 2016), avente queste quattro fasi:

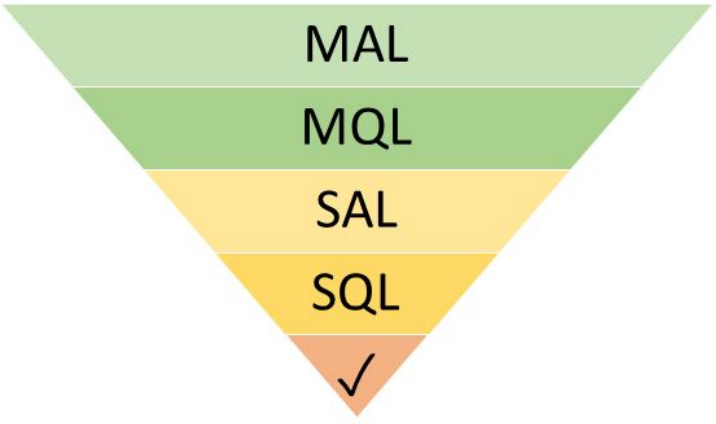
- ➔ *Attention* o *attenzione* del cliente, che è fondamentale catturare per arrivare ad una consapevolezza nei confronti dell'esistenza di un certo prodotto o servizio.
- ➔ *Interest* o *interesse* che si inizia a creare nel cliente riguardo ai benefici di tale prodotto o servizio.
- ➔ *Desire* o *desiderio* di beneficiare dell'impatto positivo creato dal prodotto o servizio.
- ➔ *Action* ovvero la spinta verso l'*azione* concreta, che sia di acquisto oppure di attivazione di un qualche tipo di prova gratuita del prodotto o servizio.



Rappresentazione grafica del modello AIDA

Chiaramente, tale modello va inteso come una guida approssimativa, di carattere generale, su cui basare le proprie scelte strategiche. Al giorno d'oggi si pone invero molta attenzione anche al comportamento dei clienti una volta che l'acquisto è stato completato. Questo ragionamento vale ancor più considerando l'industria del cloud computing, basata in larga parte su servizi in abbonamento. E' evidente come, per riuscire a mantenere gli abbonamenti attivi il più a lungo possibile, sia di primario interesse per i fornitori di servizi cloud fare in modo che i clienti vengano supportati accuratamente anche dopo l'agognata "firma del contratto" e che riescano a raggiungere i propri obiettivi tecnici e strategici con successo.

Nell'ambito delle vendite B2B nell'industria del cloud computing, il sales funnel viene di solito suddiviso in quattro fasi principali, di seguito schematizzate:



Esempio schematico di un sales funnel per il settore dei servizi di cloud computing

Nella prima fase si trovano le *Marketing Accepted Leads* o *MAL*. Per entrare in questa fase, i potenziali clienti vengono sottoposti a una prima valutazione grossolana, in genere dall'esito binario (ovvero la valutazione è superata o non superata) e sulla base di criteri di valutazione espliciti, ovvero dati relativamente stabili ed oggettivi quali l'industria di appartenenza del potenziale cliente, la dimensione dell'azienda, l'ubicazione dell'azienda, che possono aiutare, in una prima scrematura, ad eliminare leads che potrebbero essere fuori dal target di cliente ideale cercato.

Superata questa fase, le MAL vengono sottoposte ad ulteriori valutazioni, questa volta tramite criteri considerabili impliciti e comportamentali: grazie a software appositi, si possono infatti registrare e classificare azioni come, ad esempio, le visite a certe pagine web di marketing dell'azienda venditrice, il download di certi whitepaper o casi studio, la registrazione ad una newsletter. Si può così assegnare un punteggio ad ogni lead, ed una soglia limite: tutte le leads che raggiungono o superano questa soglia diventano quindi *Marketing Qualified Leads*, o *MQL*.

Chi non raggiunge il punteggio necessario per diventare MQL viene spesso inserito in un programma detto di *marketing automation*, in cui vengono inviate varie email personalizzate ai nuovi iscritti ad una newsletter e viene misurato il tasso di apertura e di interazione per queste email, continuando a classificare la lead come MAL fino a quando il suo punteggio (dato dalle interazioni) diventa abbastanza alto da passare alla fase successiva.

A questo punto del funnel si ha un momento considerabile di spartiacque tra la funzione marketing e la funzione vendite, una sorta di delicato "passaggio di testimone" in cui le leads valutate fino a questo momento positivamente sono trasferite ai rappresentanti di vendita per controllare e valutare se valga effettivamente la pena investire del tempo per cercare di convertirle in veri e propri clienti.

Se prima, nel funnel, le interazioni tra lead ed azienda erano impersonali ed automatizzate, in questa fase si richiede quasi sempre una conversazione diretta tra il contatto-lead e un

membro del team di vendite. Quest'ultimo proverà ad ingaggiare il contatto, di solito per via telefonica, cercando di capire nella maniera più oggettiva possibile le sue effettive intenzioni e necessità. Spesso, per questi fini, vengono utilizzati appositi quadri teorici di valutazione. Uno dei più usati è il *BANT*, acronimo che sta per *Budget-Authority-Need-Timing*, che si prefigge di rispondere a queste domande:

- ➔ Il contatto è disposto ad allocare il budget necessario per poter usufruire dei prodotti o servizi proposti?
- ➔ Il contatto ha l'autorità necessaria per influenzare le decisioni di vendita nel suo settore aziendale?
- ➔ Il contatto ha veramente un problema di business abbastanza importante e risolvibile dai servizi o prodotti dell'azienda venditrice?
- ➔ E' possibile riuscire a risolvere il problema di business in un tempo considerato come accettabile dal contatto?

In caso di esito positivo della prima revisione del team di vendite, il potenziale cliente passa quindi al livello di *Sales Qualified Lead*, o *SQL*.

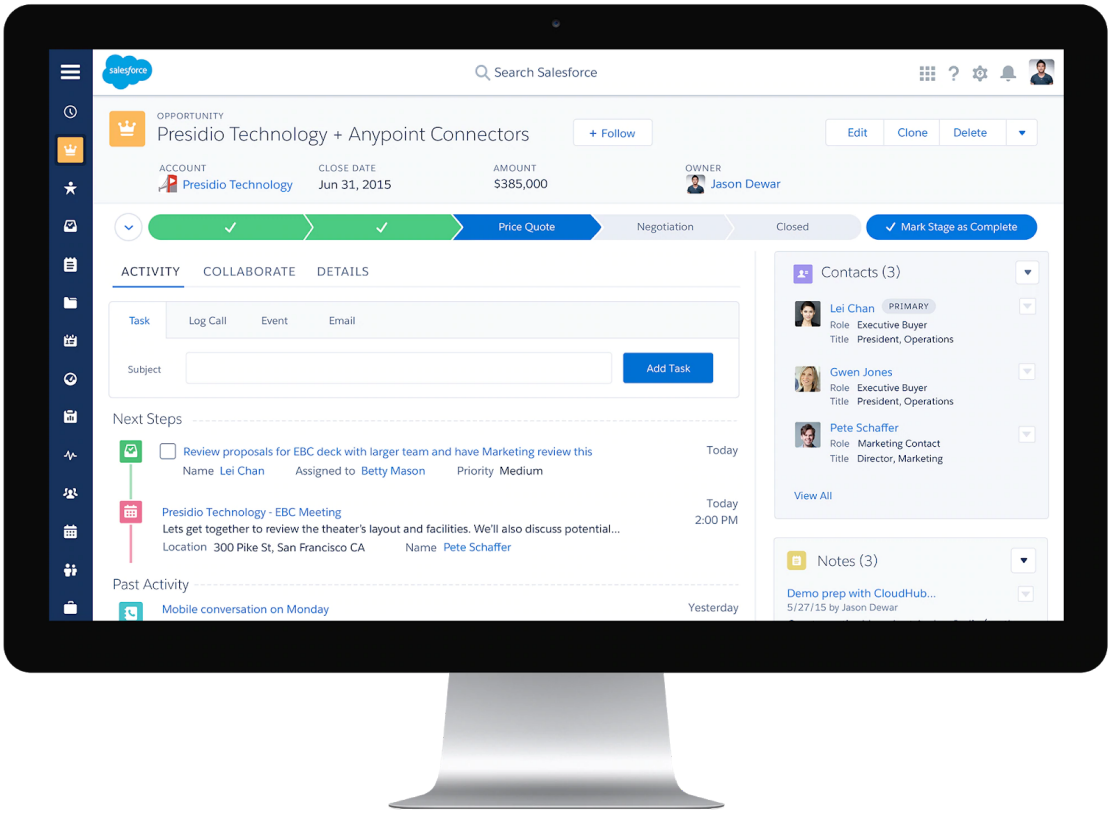
Nell'ultima fase di valutazione, le SQL più promettenti vengono infine considerate come *Sales Accepted Leads* o *SAL* se i contatti in questione, durante successivi incontri *de visu* o per via telematica, esprimono un interesse serio e soprattutto quantificabile (tramite una stima monetaria ed una finestra temporale) per i servizi o prodotti dell'azienda. In genere si inizia a parlare a questo punto di *opportunities*, ovvero di *opportunità* concrete da prioritizzare e perseguire.

Superate le quattro fasi principali, i contatti rimasti dovrebbero essere, a questo punto, regolarmente ingaggiati ed avviati alla negoziazione per l'avvio di un *Proof-of-Concept* o *POC*, ovvero un prototipo in scala ridotta dell'implementazione cloud vera e propria. La progettazione di un POC è necessaria in un contesto enterprise per consentire al cliente di poter testare un'approssimazione concreta della soluzione finale, prima di approvare definitivamente accordi e contratti con i fornitori che comportano, molto spesso, esborsi che vanno dalle centinaia di migliaia ai milioni di dollari.

CRM

Con la sigla *CRM*, che sta per *Customer Relationship Management*, ovvero “gestione delle relazioni con i clienti”, si indica un insieme di pratiche che hanno come scopo la sistematica e rigorosa amministrazione di tutte le interazioni (pianificate ed effettivamente eseguite) tra i responsabili aziendali delle vendite ed i clienti, attuali e potenziali. Comunemente, il termine CRM viene anche utilizzato per descrivere il software aziendale utilizzato per conseguire le finalità di gestione delle relazioni appena descritte.

Una suite di software CRM diventa fondamentale per ogni azienda che supera la soglia di “piccola impresa” e si ritrova a dover gestire efficientemente una quantità sempre maggiore di clienti che attraversano, in momenti diversi ed a velocità diverse, le varie tappe del sales funnel. L’azienda leader mondiale nel campo del CRM è Salesforce.



Una schermata dell’interfaccia utente del software CRM di Salesforce (Salesforce, 2015)

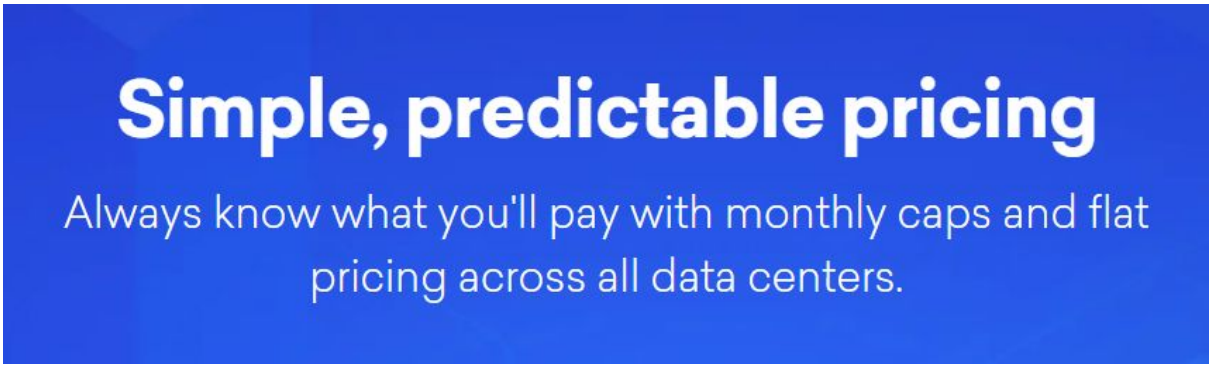
Oltre a permettere ai rappresentanti di vendita di organizzare efficacemente le proprie riunioni, i propri contatti ed i propri appunti, commenti e considerazioni sulle opportunità di vendita, il software CRM è anche utile per il controllo di performance relative ad obiettivi di vendite: ad esempio, si può monitorare il valore totale dei contratti stipulati con i clienti per trimestre, ed alla fine del trimestre verificare se è stato raggiunto o disatteso dal dipartimento un certo obiettivo in termini monetari. Gli obiettivi possono anche essere individuali: il numero di nuovi clienti che sono stati portati all'azienda dai singoli venditori può tradursi (come quasi sempre accade) in bonus assegnati ai dipendenti più performanti.

In ultimo, la suite CRM permette anche di analizzare tutti i dati registrati al proprio interno (direttamente utilizzando l'interfaccia software o, più comunemente, esportando i dati in appositi tool di analisi numerica e rappresentazione grafica come Excel, Tableau ecc.): dati che permettono di rispondere a domande del tipo: quale fase del funnel ha maggior tempo di permanenza, qual è la durata media del ciclo di vendita, in quale fase si verificano la maggior parte dei "fallimenti" e delle perdite di potenziali clienti, ed altre simili considerazioni.

Pricing

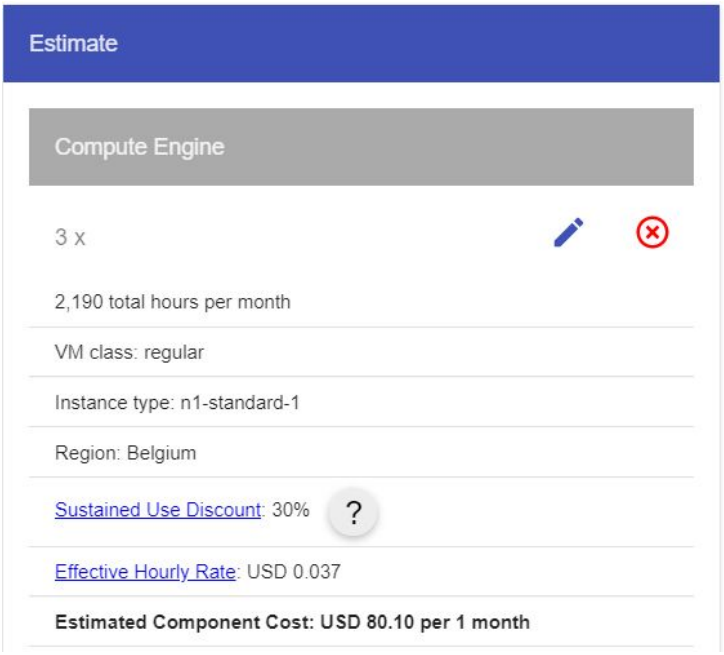
Uno degli aspetti più sensibili, e spesso fonte di confusione, nella scelta di un'azienda di adottare o meno una soluzione cloud, oppure di adottare un provider rispetto ad un altro, riguarda il costo totale sostenuto dal cliente. La moltitudine di servizi offerti da ciascun provider, infatti, e le configurazioni (di prestazioni, geografiche...) di ogni singolo servizio, determinano una struttura di prezzi estremamente complessa e spesso soggetta a cambiamenti.

La semplificazione, in questa realtà di prezzi complicati e volatili, può diventare una caratteristica di positiva distinzione: DigitalOcean, un provider importante a livello internazionale, sebbene meno blasonato rispetto ai "Big 3", evidenzia infatti come vero e vantaggio competitivo la propria struttura di prezzi semplice e prevedibile.



digitalocean.com/pricing

Per queste ragioni, è importante per i fornitori cloud riuscire a comunicare e preventivare il più chiaramente possibile quali costi potrebbero derivare da una determinata implementazione, pensata per soddisfare le singolari necessità di un cliente. Provider come AWS, Azure e Google Cloud offrono a tale scopo dei calcolatori di prezzo, pagine web interattive dove si possono selezionare singoli servizi cloud e specificarne le caratteristiche d'uso (ad esempio, il numero di gigabyte di dati da conservare per un BLOB storage, oppure quante macchine virtuali si pensa di dover utilizzare per un certo lasso di tempo), ricevendo istantaneamente in risposta una stima mensile del costo da sostenere.



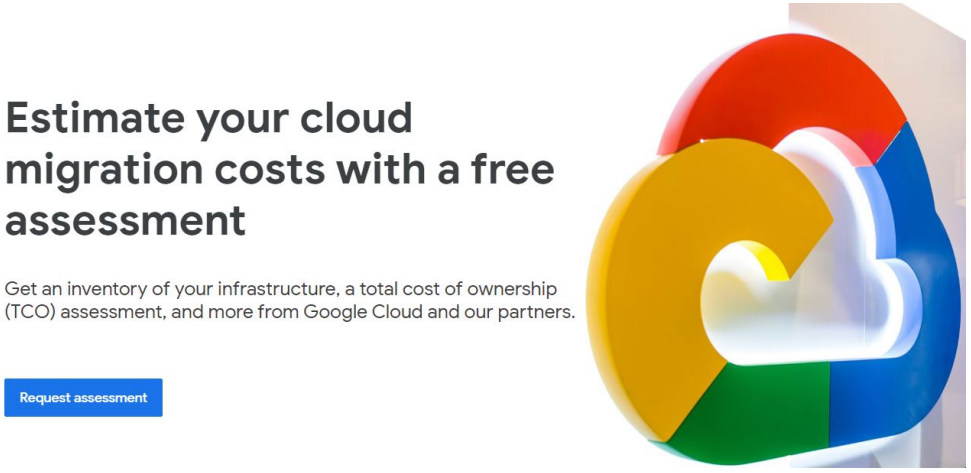
Esempio di stima del calcolatore di prezzi di Google Cloud

In generale, esistono due modelli principali di prezzatura dei servizi nel cloud: il modello cosiddetto *pay-per-use*, tale per cui il cliente paga un certo quantitativo per ogni secondo o minuto in cui i vari servizi cloud sono in funzione (di solito senza costi di attivazione o disattivazione dei servizi) e il modello ad abbonamento, più tipico delle soluzioni SaaS, in cui è richiesto preventivamente un certo ammontare di denaro per ogni mese od anno di utilizzo e per ogni utente che andrà ad usufruire del servizio, indipendentemente dall'uso effettivo delle risorse SaaS, chiaramente entro certi limiti superiori a seconda del livello di abbonamento scelto (ad esempio, "base" o "premium" ed altre diciture simili). Può esistere inoltre, più raramente, una combinazione ibrida dei modelli appena spiegati, tale per cui il cliente paga una base minima in aggiunta al costo pay-per-use.

Si può aggiungere a questi modelli il cosiddetto *free tier*, ovvero un livello gratuito di servizio per molte offerte cloud: entro un determinato tetto di utilizzo delle risorse, il cliente non dovrà pagare nulla, invogliando quindi i curiosi a provare un nuovo servizio senza spendere denaro.

Per comunicare con più efficacia ed immediatezza i costi associati al cloud, nell'industria si è soliti inoltre spostare l'attenzione dal mero prezzo unitario dell'implementazione delle

singole risorse cloud combinate al *total cost of ownership* o *TCO*, ovvero una misura del costo totale di proprietà sviluppato, per il settore informatico, dall'azienda di ricerca e consulenza Gartner; il TCO rappresenta il costo dell'intero ciclo di vita di determinate risorse informatiche impiegate da un'azienda, e permette di creare scenari di previsione *what-if* per evidenziare e quantificare i benefici in termini di risparmio nel passaggio da un data center on-prem al cloud, soprattutto per quanto riguarda le operazioni di installazione, manutenzione e salvaguardia delle infrastrutture IT.

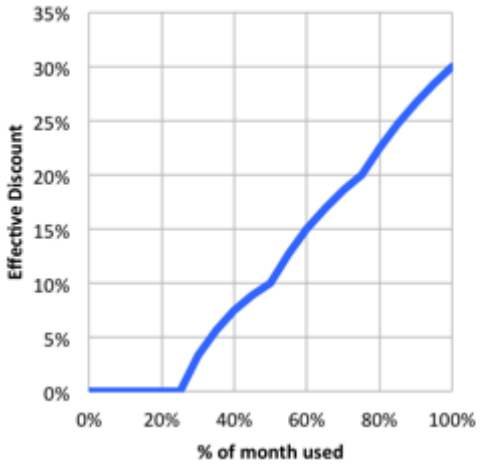


Esempio di promozione per una valutazione di *TCO offerta da Google Cloud nel 2019*

Tramite il TCO possono essere considerati anche fattori come i servizi, offerti dai principali fornitori cloud, progettati per ridurre lo spreco di risorse ed ottimizzare la spesa dei clienti; tali servizi sono noti come *sizing* o *rightsizing recommendations*: essi fanno uso di algoritmi che analizzano automaticamente dati storici relativi ai consumi nel tempo di risorse associate alle macchine virtuali nel cloud, ed offrono ai clienti dei report in cui viene dettagliato se, ad esempio, le CPU o le RAM attualmente in uso sono troppo potenti rispetto ai carichi di lavoro che devono supportare, specificando altresì a quale tipo di VM dalle prestazioni inferiori si potrebbe migrare, consentendo in tal modo di ottimizzare le spese.

In ultimo, si devono considerare anche le principali tipologie di sconti offerti dai fornitori cloud, che ricadono usualmente nelle categorie di *sustained use discounts* e di *committed use discounts*.

I primi sono sconti che si applicano automaticamente a macchine virtuali che vengono usate costantemente per una certa porzione minima del mese, incrementando la quantità dello sconto, fino ad un certo limite, all'aumentare della frazione di tempo impiegata. Il grafico seguente mostra ad esempio il tasso di sconto offerto da Google Cloud in rapporto alla percentuale di tempo in un mese in cui rimangono attive le VM.



Entità dei sustained use discounts offerti da Google Cloud (Google Cloud, 2020)

I committed use discounts sono invece sconti applicati previa sottoscrizione di particolari contratti in base ai quali il cliente si impegna a noleggiare una certa quantità di risorse di elaborazione nel cloud per un periodo di tempo relativamente lungo, di solito qualche anno, in cambio di un prezzo minore per l'utilizzo di tali risorse rispetto alla classica formula cloud del pay-per-use. In pratica, questo tipo di contratto trasforma la fatturazione da pay-per-use ad una forma di abbonamento poiché, per la durata di tempo prestabilita, il cliente dovrà pagare determinate quantità di denaro ogni mese indipendentemente dall'uso effettivo delle risorse.

Team di vendite

Nei contesti di grandi organizzazioni, come quelle dei principali fornitori cloud, esiste una grande varietà di ruoli all'interno della funzione di vendita della società: ogni ruolo ha compiti ben definiti e specializzati in modo da abbracciare tutta la vasta gamma degli aspetti relativi alle attività di vendita.

Quasi sempre si può trovare il ruolo di *account* o *business development representative* (*ADR*/*BDR*). Tale figura è incaricata di valutare e qualificare le leads provenienti dal marketing, usando il metodo BANT o altri modelli affini. La sua attività principale è quindi quella di condurre call quotidiane con più clienti possibili e inoltrare i contatti qualificati ai membri del team di vendita descritti di seguito.

Un'altra figura rilevante è quella del cosiddetto *inside sales representative*, o *ISR*. La dicitura "inside" specifica che il lavoro del rappresentante si svolge perlopiù all'*interno* dell'azienda che lo impiega, dato che gli incontri con i potenziali clienti si svolgono principalmente in remoto. Mentre di solito gli ADR si occupano di qualificare nuovi potenziali clienti, gli ISR sono incaricati di sviluppare le relazioni prioritizzate dagli ADR e portarle a compimento, cercando per quanto possibile di arrivare ad accordi di vendita, oltre che seguire gli attuali clienti nel proprio portfolio e cercare di sfruttare le opportunità di *cross-selling* (vendita di prodotti o servizi correlati) e *up-selling* (vendita di prodotti o servizi dal costo superiore).

Contrapposto al concetto di inside sales si trova il concetto di *outside* o *field sales*, ovvero una forma di vendita sul campo, direttamente presso le sedi dei potenziali clienti, necessaria quando le dimensioni, in termini finanziari, degli accordi di vendita sono tali da richiedere lunghe trattative ed incontri con una moltitudine di decisori. Il ruolo associato a questa pratica è quello del *field sales representative* o *FSR*. Gli FSR agiscono con il supporto degli ISR e rappresentano il punto di contatto principale dell'azienda con i direttori e gli executive dei clienti enterprise.

Un figura di supporto tecnico, pur rimanendo nell'ambito delle vendite, per ISR ed FSR è quella del *customer engineer* o *CE*. Questo ruolo funge da collegamento tra i team interni di

ingegneria del prodotto ed i team di vendita che interagiscono con i clienti. Dove i rappresentanti di vendite contribuiscono alla comunicazione del valore con carisma e capacità di persuasione, i CE provvedono all'elaborazione tecnica delle soluzioni software e di architettura informatica ricercate dai clienti. Il loro ruolo è fondamentale per la realizzazione dei Proof-of-Concept e per il supporto dei team tecnici dei clienti coinvolti nelle trattative.

In Google Cloud è presente inoltre, parallelamente alla figura di inside sales representative, il ruolo di *enterprise customer developer*, o *ECD* in breve. Il team ECD si occupa di fornire analisi strategiche a supporto dell'intero ciclo di vendita. La dicitura enterprise sta ad indicare il fatto che, mentre gli ISR si occupano prevalentemente di aziende medio-grandi, gli ECD sono coinvolti esclusivamente in clienti e potenziali clienti di livello enterprise. Il ruolo del team ECD sarà discusso nel dettaglio nella sezione seguente della trattazione.

Esperienza di tirocinio

Introduzione

Sono stato inserito nel team di *enterprise customer development* (*ECD*) di Google Cloud, il cui focus è elaborare analisi e strategie di vendita volte all'acquisizione ed al rafforzamento delle relazioni con clienti di livello enterprise, vale a dire aziende leader nei rispettivi settori, spesso multinazionali, aventi dimensioni considerevoli in termini di asset e di fatturato.

Il team è diviso per nazioni europee: al suo interno si trovano, ad esempio, dipendenti incaricati di gestire clienti francesi piuttosto che spagnoli o tedeschi. All'inizio del mio percorso sono stato assegnato al territorio italiano, dovendo gestire nello specifico strategie rivolte a clienti nel settore delle telecomunicazioni e della grande distribuzione organizzata.

Il mio tempo in Google Cloud è stato strutturato come segue: circa metà del mio carico di lavoro è stato allocato per portare avanti delle attività cosiddette *core*, ovvero le attività fondamentali svolte quotidianamente o settimanalmente dai membri del team. Il tempo rimanente è invece servito per sviluppare un progetto di tirocinio riguardante l'organizzazione di una serie di eventi a Dublino dedicati ad incontri e workshop con decisori e manager di svariati clienti Google Cloud.

Appena arrivato, dopo aver conosciuto il mio team e molti stagisti provenienti da altri team, ho subito cominciato una serie di corsi volti al rapido apprendimento degli elementi fondamentali di Google Cloud e delle principali strategie di vendita. Una volta affrontati questi corsi sono stato immediatamente "gettato nella mischia", ricevendo grandi responsabilità e ritrovandomi a gestire gli stessi impegni degli altri membri del team, potendo naturalmente fruire di preziosi consigli e supporto in qualunque momento.

Attività core

Le attività del team ECD possono essere definite come consulenza strategica interna di supporto alle vendite. In pratica, ogni membro del team gestisce un portafoglio di clienti enterprise operanti in una certa nazione europea. Ognuno lavora inoltre a stretto contatto con una serie di direttori delle vendite (che hanno il ruolo di field sales representative, o FSR, di cui si è discusso in precedenza) di Google Cloud, i quali si occupano di gestire le relazioni dirette con le aziende e partecipare ad incontri con executive, direttori e responsabili IT dei clienti di loro competenza.

I dipendenti ECD hanno quindi come obiettivo principale il fornire supporto strategico ai direttori delle vendite di Google Cloud tramite una serie di attività. Queste ultime sono molto varie: spaziano infatti dal mappare gli stakeholder di società con cui non esistono ancora rapporti all'organizzare eventi per determinati clienti, dall'elaborare piani strategici di vendita a lungo termine al creare contenuti di supporto (ad esempio presentazioni) che puntino a specifiche esigenze o problemi che il cliente sta cercando di risolvere sfruttando tecnologie di cloud computing.

I clienti enterprise hanno bisogno di grandi attenzioni da parte dei fornitori cloud, poiché i contratti stipulati con essi comportano ingenti investimenti monetari e richiedono, sovente, molto tempo: l'acquisizione di un cliente di questo livello, infatti, raramente impiega meno di un anno.

Riconoscere le necessità di business di questi clienti (tramite analisi strategiche e colloqui esplorativi), capire come comunicare al meglio i benefici di una determinata soluzione per un determinato cliente, identificare i principali decisori che valga la pena contattare, individuare il tipo di tecnologie utilizzate e comprendere quale budget possa essere messo a disposizione: tutti questi compiti, che ricadono nella disciplina della *sales intelligence*, sono essenziali per riuscire a trasformare con successo una grande azienda in un nuovo cliente o per supportare un attuale cliente verso l'implementazione duratura e di successo di una soluzione cloud.

Sebbene molti aspetti della sales intelligence siano sostanzialmente assimilabili alle attività proprie della più generica business intelligence, essa si contraddistingue per il suo specifico scopo di fornire a manager di vendita e rappresentanti sul campo informazioni di livello strategico che possano essere utili, fruibili e rilevanti.

Verranno di seguito approfonditi i temi e le principali attività legate alla sales intelligence.

Analisi del settore industriale

La sales intelligence aggiunge valore alle operazioni di vendita tramite lo sviluppo di ricerche che possano evidenziare come il cloud sia d'aiuto per il raggiungimento degli obiettivi strategici di una determinata azienda.

I vari benefici riscontrabili adottando soluzioni cloud sono stati ampiamente sviscerati nel corso di questa trattazione. Ricapitolando, i principali vantaggi, di cui la maggior parte delle aziende in qualsiasi industria potrebbe beneficiare, sono:

- ➔ trasformare immobilizzazioni di capitale in spese operative stabili e prevedibili;
- ➔ concentrarsi sulle proprie competenze chiave e sul proprio core business facendo outsourcing di tutto ciò che riguarda le risorse computazionali dell'azienda (o parte di esse);
- ➔ garantire scalabilità, elevata disponibilità ed affidabilità di sistemi informatici;
- ➔ dislocare globalmente le proprie risorse informatiche;
- ➔ risparmiare in termini di personale e manutenzione IT (sebbene siano richieste persone con competenze più specifiche, che abbiano familiarità con il cloud);
- ➔ consentire lo sviluppo di algoritmi di machine learning tramite apposito hardware messo a disposizione dai fornitori cloud;
- ➔ facilitare la gestione dei big data;
- ➔ abilitare, tramite SaaS, la collaborazione in tempo reale da remoto;
- ➔ ridurre nel lungo termine il costo totale di proprietà legato alle infrastrutture informatiche;
- ➔ ridurre il rischio relativo alla sperimentazione di nuove tecnologie e soluzioni informatiche, poiché è molto più veloce ed economico attivare (e disattivare)

un'istanza di una macchina virtuale nel cloud tramite qualche clic piuttosto che dover gestire ed espandere un'infrastruttura di proprietà.

Sebbene innegabili, i benefici descritti fino ad ora sono generici: si può comunicare valore molto più efficacemente sapendo cos'è che sta più a cuore al cliente in considerazione della sua specifica situazione. Ad esempio, un messaggio come "il cloud permette lo sviluppo e l'uso su larga scala di potenti algoritmi di machine learning" è meno potente rispetto al messaggio "il cloud permetterà di ridurre del 30% i tempi di risoluzione dei problemi dei vostri clienti tramite la classificazione semantica e la prioritizzazione automatica dei messaggi all'assistenza".

Una solida base di partenza per questi fini è data dall'analisi delle necessità e degli obiettivi strategici propri del settore di appartenenza delle aziende bersaglio.

Per l'industria manifatturiera sono fondamentali soluzioni mirate alla gestione dei big data e dei dispositivi IoT; organizzazioni nell'ambito della finanza hanno particolare interesse a soluzioni che possano incrementare la sicurezza delle informazioni da loro gestite (come dati relativi alle transazioni bancarie), tramite server fisicamente protetti, isolati e crittografati, oltre a servizi antifrode automatici basati sul machine learning.

Ancora, aziende nel ramo GDO (grande distribuzione organizzata) e vendita al dettaglio saranno interessate a servizi di data warehouse veloci ed affidabili in modo da poter pianificare accuratamente i livelli di inventario richiesti nei diversi negozi e centri commerciali; società che producono videogiochi potranno trarre beneficio da offerte di database globalmente distribuiti ed orizzontalmente scalabili per consentire un'espansione massiva senza limiti dei propri prodotti digitali.

Tali analisi devono poi essere integrate con i case studies di clienti attuali del fornitore cloud, in modo tale da mostrare risultati concreti, e fatte confluire in presentazioni mirate, utilizzabili dai direttori di vendite durante gli incontri con i decision maker dei potenziali clienti.

Analisi dei decisori

L'analisi della struttura organizzativa di un'azienda è fondamentale nel caso in cui non esistano (o siano estremamente limitati) precedenti rapporti o contatti con essa. I responsabili della sales intelligence devono spesso occuparsi, per questa ragione, di sviluppare strategie che permettano di individuare e prioritizzare una serie di contatti chiave dei decisori principali delle aziende che devono essere convertite in clienti.

Vi sono molte figure con cui è probabile l'interazione durante il processo di vendita, come i responsabili degli acquisti, i project manager, gli ingegneri, i direttori di dipartimento e i membri del consiglio di amministrazione. Tra questi ruoli va deciso un preciso ordine di approccio, tenendo conto dei loro rapporti di potere e di influenza; si deve inoltre considerare la tipologia di organizzazione societaria del potenziale cliente (società di capitali, società di persone, società cooperative etc.) per poter stabilire l'approccio più performante.

Il messaggio da veicolare dipende in larga parte dal tipo di figura con cui si vuole instaurare un rapporto; va infatti tenuta a mente la possibile dualità di figure professionali presenti nelle aziende: decisori lato business e decisori dei reparti IT. Mentre ai primi vanno evidenziati i contributi del cloud in termini di risparmi o di redditività del capitale investito, gli ultimi saranno più interessati a discutere dettagli tecnici riguardo all'implementazione e di come il cloud possa essere d'aiuto per i loro obiettivi di reparto (alta disponibilità, sicurezza informatica...).

I target ideali, verso cui dovrebbero essere rivolte maggiormente le attenzioni, sono le figure dotate di potere esecutivo che si occupano di IT e tecnologia all'interno dell'azienda perché saranno loro, molto probabilmente, a prendere la decisione finale riguardo all'adozione o meno delle offerte cloud proposte. I ruoli principali di questo genere sono:

- ➔ *CIO*, acronimo di *chief information officer*, che dirige la manutenzione ed il controllo di tutti i sistemi informatici della società;
- ➔ *CTO*, *chief technology officer*, che si occupa di selezionare le tecnologie applicabili ai prodotti o servizi offerti;

- ➔ *CISO* ovvero *chief information security officer*, figura di riferimento per la sicurezza informatica.

Per agevolare la ricerca di informazioni organizzative, si ricorre spesso a programmi appositi dotati di enormi database di contatti. Uno degli strumenti più diffusi è il *Sales Navigator* offerto da LinkedIn, il famoso social network orientato al mondo professionale. Tramite questo software è possibile ricevere liste di tutti gli individui iscritti al social network che hanno specificato nel proprio profilo personale di lavorare presso una certa azienda, e filtrarli poi a seconda del ruolo desiderato.

Analisi dei prodotti e servizi IT

Capire la natura dei prodotti e servizi informatici di un'azienda è di grande aiuto nella formulazione di strategie di vendita mirate.

Riuscire a scoprire che una certa impresa abbia da poco firmato un contratto di fornitura con un altro provider cloud eviterà di proporre servizi inclusi in quel contratto, e consentirà conseguentemente di concentrarsi su offerte complementari, canalizzando la discussione sui benefici per l'azienda di un approccio multi-cloud.

Rilevare invece che una determinata azienda sia in affari con fornitori di servizi legati ai server fisici e di software per la gestione di data center permetterà di direzionare la strategia di vendita sulla modernizzazione di sistemi *legacy* (ovvero sistemi obsoleti) dell'azienda, che nella maggior parte dei casi limitano l'innovazione e sono difficili da mantenere.

E' tuttavia importante specificare che esistono altresì dei sistemi on-premises moderni su cui alcune aziende potrebbero aver pianificato ed investito massicciamente: è determinante, come analista o sales executive, non giungere subito alla conclusione che tutto ciò che è al di fuori del cloud è obsoleto ed inadeguato; al contrario, serve grande capacità di calarsi nei panni del cliente e cercare il più possibile di risolvere problemi concreti. Ad esempio, nel caso in cui un'azienda non si senta limitata dalla propria attuale

architettura informatica, la strategia di vendita corretta potrebbe essere quella di spostare l'attenzione del cliente verso offerte SaaS oppure verso un'architettura cloud ibrida.

Alcune aziende potrebbero aver adottato servizi offerti da provider cloud di dimensione limitata ma altamente specializzati. I provider cloud di notevoli dimensioni possono apparire infatti come troppo generici ed incapaci di fornire adeguate attenzioni ad un singolo cliente; al contrario, aziende più piccole ed agili possono adattarsi meglio a specifiche esigenze pur avendo meno esperienza o risorse.

Per questa ragione, non è raro per i fornitori cloud più grandi acquisire direttamente tali società minori tramite ingenti offerte. Ad esempio, player più piccoli nel mercato come Looker e Tableau, fornitori di soluzioni mirate per la business intelligence, sono stati comprati da giganti come Google e Salesforce.

La strategia di vendita in questi casi può essere quella di evidenziare i vantaggi forniti dalla stretta integrazione dei servizi offerti dai leader del mercato, che permette sinergie non indifferenti: fatturazione unificata, sistemi di *reporting* e *logging* degli eventi più efficaci e precisi, interconnessione semplificata di infrastrutture ed interfacce di programmazione. Il costo certamente può fare la differenza, senza però dimenticare che alcuni diversificatori, come la possibilità di fornire hyperscaling ed il possesso di avanzati modelli di machine learning basati su dati proprietari, costituiscono vantaggi competitivi enormi.

Riuscire a scoprire i prodotti ed i servizi IT usati da un'azienda non avendo mai avuto contatti precedenti con essa costituisce un'attività complessa, poiché non è spesso interesse delle aziende rivelare pubblicamente le tecnologie adottate. Si possono carpire informazioni da interviste rilasciate dai direttori della tecnologia, oppure analizzando le attività principali e le descrizioni dei ruoli del personale tecnico. Per semplificare queste attività si può ricorrere a software specializzati detti *web scraper*, che sfruttano il machine learning per aggregare automaticamente queste informazioni in dashboard facilmente consultabili.

Analisi del budget

Essere a conoscenza del budget per i prodotti e servizi informatici consente conversazioni più proficue con i potenziali clienti: si è equipaggiati meglio per quantificare l'entità del risparmio possibile con il cloud, oltre ad avere un quadro di riferimento per comprendere quanto il potenziale cliente potrebbe allocare per provare le soluzioni cloud proposte; a tal proposito, si può discutere di come un proof-of-concept potrebbe rappresentare una frazione minima della spesa attuale dell'azienda per certe infrastrutture IT ma potrebbe, allo stesso tempo, portare a significativi risparmi.

Una prima stima grossolana del budget IT di un'azienda si può basare su analisi effettuate da società di ricerca e consulenza: Deloitte proponeva nel 2017 una media pesata della frazione della spesa per l'informatica rispetto ai ricavi del 3.28%, con un massimo intorno al 7% per il settore bancario e circa l'1.5% nell'industria edilizia.

Budget IT come percentuale dei ricavi delle aziende in varie industrie (Deloitte, 2017)

E' ragionevole pensare che i budget per l'informatica stiano crescendo a causa di fattori come la sempre più presente digitalizzazione nella vita quotidiana dei consumatori; come già discusso, la pandemia di COVID-19 ha portato ad un enorme incremento di adozione di soluzioni per supportare il lavoro da remoto.

Partendo da tali dati, si possono poi affinare le stime andando ad esplorare i documenti finanziari aziendali, se disponibili pubblicamente, nonché basarsi sui costi dei prodotti e dei servizi informatici adottati dalle aziende bersaglio.

Progetti di tirocinio

Google Cloud Experience

Il mio progetto di tirocinio principale richiedeva la gestione di una serie di incontri e *workshop* tenuti nei centri conferenze di Google a Dublino. Al mio arrivo, questo progetto si trovava ancora in fase embrionale: era stato infatti creato pochi mesi prima da un membro del mio team e da un membro del team di inside sales, ed erano stati organizzati con successo quattro eventi pilota. Ora il progetto andava portato al livello successivo, normalizzandolo e facendolo crescere.

Per fare ciò era necessario un approccio su più fronti: innanzitutto l'intero processo di creazione di un evento doveva essere standardizzato e reso il più semplice possibile e, allo stesso tempo, flessibile abbastanza da poter venire incontro alle diverse richieste dei clienti. In secondo luogo, il *format* doveva essere pubblicizzato internamente affinché i membri dei team di vendite di Google Cloud potessero venirne a conoscenza e cominciare ad utilizzarlo come strumento strategico.

Al fine di promuovere internamente il format in maniera efficace, mi sono in primo luogo occupato di un'analisi dei dati relativi alle opportunità aperte nel 2019, ovvero delle relazioni con clienti promettenti ma non ancora culminate in una vendita o upsell, estraendo tali dati dal software CRM e rielaborandoli in dashboard interattive create utilizzando fogli di calcolo. Ciò mi ha permesso di individuare gli impiegati dei team di vendite che potessero essere potenzialmente più interessati a questi eventi, per concludere efficacemente le proprie opportunità aperte.

Una volta individuati i contatti prioritari ho effettuato incontri e meeting con ciascuno di loro al fine di "evangelizzare" il format di eventi e misurare l'effettivo interesse per gli stessi. Nel contempo ho creato una campagna di marketing interno allo scopo di suscitare attenzione e curiosità tra i *Googlers*: tra le varie attività, ho creato un sito web interno per il progetto, disegnato ed affisso poster negli uffici e presentato i benefici dell'iniziativa a vari membri dei team di marketing Google in Italia, Francia, Paesi Bassi e Regno Unito.

Contemporaneamente ho agito da vero e proprio program manager per tutto ciò che riguardava l'aspetto operativo dell'iniziativa: ho gestito gli incontri e le attività per gli impiegati che si erano convinti ad organizzare un evento con i propri clienti e monitorato il progresso e lo status delle varie componenti del programma tramite dashboard e tracker, creati appositamente da me per consentire una visuale immediata e ad ampio raggio di ogni aspetto chiave del "Google Cloud Experience".

Impatto dei piani di apprendimento

Parallelamente al progetto principale di tirocinio ed alle attività core del ruolo di ECD, ho inoltre intrapreso autonomamente un progetto aggiuntivo, con il benestare dei miei superiori, assieme ad un altro mio collega stagista.

L'obiettivo di questo progetto era di quantificare l'impatto positivo apportato dal mio team nella creazione e gestione di piani di apprendimento volti ad educare il personale dei clienti all'uso delle soluzioni cloud, misurato tramite la crescita di iscritti a tali piani.

La realizzazione è stata molto complessa, richiedendo l'estrazione di dati da più sistemi separati tramite diverse interrogazioni SQL, l'analisi di migliaia di righe in fogli di calcolo ed il design di grafici che sintetizzassero efficacemente i risultati ottenuti.

Fatto ciò, abbiamo creato una presentazione che evidenziasse le informazioni reperite e l'abbiamo esposta al team. Le metriche individuate sono state apprezzate ed utilizzate dai nostri manager, ed abbiamo inoltre avuto l'occasione di presentare i risultati del progetto al team europeo che si occupava della creazione dei piani di apprendimento al fine di condividere best-practice e discutere di possibili miglioramenti procedurali.

Risultati ottenuti

La valutazione complessiva dell'esperienza in Google Cloud si basava sul feedback del manager del progetto di tirocinio nonché sul completamento di un certo numero di attività core del team.

Per quanto riguarda il primo punto, il feedback è stato molto positivo: sono riuscito ad orchestrare con successo l'espansione della serie di eventi interni organizzando rigorosamente le attività del progetto, creando interesse tra i colleghi e coordinando la programmazione dei nuovi incontri.

Rispetto alle attività core, sono riuscito a raggiungere e superare gli obiettivi previsti, tra cui si possono citare:

- ➔ analisi approfondite delle strutture organizzative, del budget e dei principali decisori IT di aziende nel mio portfolio con cui non esistevano ancora relazioni di vendita, riuscendo nell'identificazione di decine di nuovi contatti;
- ➔ creazione e lancio di una campagna di *email marketing* per i contatti individuati nella fase della suddetta analisi, portando all'organizzazione di quattro meeting tra i decisori raggiunti ed il mio direttore di vendite;
- ➔ gestione in maniera autonoma di una *call* con un responsabile IT di una delle società da me analizzate al fine di discutere potenziali implementazioni di servizi cloud;
- ➔ elaborazione di una strategia di vendita basata su un servizio relativamente nuovo tra le offerte di Google Cloud chiamato *Apigee*, orientato alla gestione su larga scala di interfacce di programmazione;
- ➔ ricerca e gestione di contenuti e speaker interni per un importante workshop al quale avrebbero partecipato direttori ed executive di una delle società presenti nel mio portfolio.

Apprendimenti

L'esperienza come membro del team ECD in Google Cloud mi ha permesso di crescere molto dal punto di vista personale e professionale, sotto vari aspetti.

Innanzitutto, ho imparato a gestire le interazioni con una grande quantità di stakeholder: tramite il progetto di tirocinio ho avuto l'occasione di confrontarmi con più di 40 dipendenti di Google Cloud (tra cui direttori del reparto vendite e del reparto marketing), illustrando loro i benefici dell'iniziativa e guidando e supportando gli interessati nella creazione e nella gestione degli eventi.

Avendo a che fare con molte attività parallele da gestire ogni giorno, ho acquisito la capacità di prioritizzare efficacemente il mio tempo, pianificando nel dettaglio gli obiettivi quotidiani e settimanali.

Ho appreso come essere molto proattivo, caratteristica fondamentale nel settore del cloud, un contesto dinamico ed in espansione in cui le opportunità vanno colte immediatamente; non limitandomi a seguire pedissequamente le istruzioni dei miei superiori, ho spesso proposto nuove azioni da intraprendere e prospettive differenti.

Ho migliorato le mie abilità nel dare e ricevere feedback onesto e costruttivo, pratica incoraggiata attraverso dei meeting periodici con i miei superiori dedicati alla valutazione del supporto da loro ricevuto ed alla revisione dei miei risultati fino a quel momento.

Ritengo inoltre di aver internalizzato molti dei cardini della cultura aziendale di Google, tra cui quello che considero più importante è sintetizzato dal motto "aim for 10x, not 10%", ovvero "punta a dieci volte tanto, non al 10% in più", frase che sottolinea l'importanza di sforzarsi di pensare ad innovazioni che possano portare ad un miglioramento decisamente superiore rispetto alla situazione attuale, piuttosto che limitarsi mentalmente a ricercare cambiamenti marginali sicuri ma di basso impatto.

Fonti

Alexa Internet, 2020. “The top 500 sites on the web”. [Online]
Disponibile su: https://www.alexa.com/topsites
[Visitato il 10 05 2020].

Amazon, 2006. “Press Release: Amazon Web Services Launches”. [Online]
Disponibile su:
https://press.aboutamazon.com/news-releases/news-release-details/amazon-web-services-launches-amazon-s3-simple-storage-service
[Visitato il 11 04 2020].

Amazon Web Services, 2020. “Pub/Sub Messaging”. [Online]
Disponibile su: https://aws.amazon.com/pub-sub-messaging/
[Visitato il 03 06 2020].

Amazon Web Services, 2020. “Message Queues”. [Online]
Disponibile su: https://aws.amazon.com/message-queue/
[Visitato il 03 06 2020].

Baker, V., Elliot, B., Sicular, S., Mullen, A. & Brethenoux, E., 2020. “Magic Quadrant for Cloud AI Developer Services”. [Online]
Disponibile su: https://www.gartner.com/en/documents/3981253
[Visitato il 21 06 2020].

Clement, J., 2020. “Annual revenue of Google from 2002 to 2019”. [Online]
Disponibile su: https://www.statista.com/statistics/266206/googles-annual-global-revenue/
[Visitato il 11 04 2020].

Clement, J., 2020. “Annual revenue of Alphabet from 2017 to 2019, by segment”. [Online]
Disponibile su:
https://www.statista.com/statistics/633651/alphabet-annual-global-revenue-by-segment/
[Visitato il 10 04 2020].

Clement, J., 2020. “Global Google Cloud revenues from 2017 to 2019”. [Online]
Disponibile su: https://www.statista.com/statistics/478176/google-public-cloud-revenue/
[Visitato il 11 04 2020].

Dally, B., 2010. “Life After Moore's Law”. [Online]
Disponibile su:
https://www.forbes.com/2010/04/29/moores-law-computing-processing-opinions-contributors-bill-dally.html
[Visitato il 02 04 2020].

Deloitte, 2017. “Technology budgets: From value preservation to value creation”. [Online]
Disponibile su:
https://www2.deloitte.com/us/en/insights/focus/cio-insider-business-insights/technology-investments-value-creation.html
[Visitato il 18 05 2020].

Dougherty, C., 2015. “Google to Reorganize as Alphabet to Keep Its Lead as an Innovator”.
[Online]
Disponibile su:
https://www.nytimes.com/2015/08/11/technology/google-alphabet-restructuring.html
[Visitato il 01 04 2020].

Doyle, C., 2016. A Dictionary of Marketing. 4a ed. Oxford: Oxford University Press.

Drake, M., 2019. “Understanding Database Sharding”. [Online]
Disponibile su:
https://www.digitalocean.com/community/tutorials/understanding-database-sharding
[Visitato il 04 05 2020].

Fedoseenko, V., 2019. “A brief history of virtualization, or why do we divide something at all”. [Online]
Disponibile su: https://www.ispsystem.com/news/brief-history-of-virtualization
[Visitato il 25 05 2020].

Feldman, S., 2019. “The Cloud Market Keeps Moving Upwards”. [Online]
Disponibile su: https://www.statista.com/chart/19039/cloud-infrastructure-revenue/
[Visitato il 11 04 2020].

Gartner, 2019. “Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17% in 2020”. [Online]
Disponibile su:
https://www.gartner.com/en/newsroom/press-releases/2019-11-13-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2020
[Visitato il 03 05 2020].

Gartner, 2020. “Gartner Says Global IT Spending to Decline 8% in 2020 Due to Impact of COVID-19”. [Online]
Disponibile su:
https://www.gartner.com/en/newsroom/press-releases/2020-05-13-gartner-says-global-it-spending-to-decline-8-percent-in-2020-due-to-impact-of-covid19
[Visitato il 12 05 2020].

Google, 2020. “Da un garage al Googleplex”. [Online]
Disponibile su: https://about.google/our-story/
[Visitato il 06 04 2020].

Google Cloud, 2019. “TCO Assessment”. [Online]
Disponibile su: https://inthecloud.withgoogle.com/tco-assessment-19/form.html
[Visitato il 21 06 2020].

Google Cloud, 2020. “Life of Cloud Spanner Reads & Writes”. [Online]
Disponibile su:
https://cloud.google.com/spanner/docs/whitepapers/life-of-reads-and-writes
[Visitato il 16 06 2020].

Google Cloud, 2020. “Sole-tenant nodes”. [Online]
Disponibile su: https://cloud.google.com/compute/docs/nodes/sole-tenant-nodes
[Visitato il 06 05 2020].

Google Cloud, 2020. “Sustained use discounts”. [Online]
Disponibile su: https://cloud.google.com/compute/docs/sustained-use-discounts
[Visitato il 18 05 2020].

Google Cloud, 2020. “Google Cloud customers”. [Online]
Disponibile su: https://cloud.google.com/customers
[Visitato il 22 04 2020].

Google Cloud, 2020. “Balancing Strong and Eventual Consistency with Datastore”. [Online]
Disponibile su:
https://cloud.google.com/datastore/docs/articles/balancing-strong-and-eventual-consistency-with-google-cloud-datastore/
[Visitato il 26 04 2020].

Google Cloud, 2020. “Cloud Firestore”. [Online]
Disponibile su: https://cloud.google.com/firestore
[Visitato il 20 04 2020].

Google Cloud, 2020. “Migrating data warehouses to BigQuery: Data pipelines”. [Online]
Disponibile su: https://cloud.google.com/solutions/migration/dw2bq/dw-bq-data-pipelines
[Visitato il 16 05 2020].

Google Cloud, 2020. “CONTAINER DI GOOGLE: Un modo migliore di sviluppare ed eseguire il deployment delle applicazioni”. [Online]
Disponibile su: https://cloud.google.com/containers?hl=it
[Visitato il 29 04 2020].

Google Cloud, 2020. “Google Cloud Platform Pricing Calculator”. [Online]
Disponibile su: https://cloud.google.com/products/calculator
[Visitato il 10 04 2020].

Google Cloud, 2020. “Storage Classes | Cloud Storage”. [Online]
Disponibile su: https://cloud.google.com/storage/docs/storage-classes
[Visitato il 12 06 2020].

HashiCorp, 2020. “Terraform: use Infrastructure as Code to provision and manage any cloud, infrastructure, or service”. [Online]
Disponibile su: https://www.terraform.io/
[Visitato il 12 05 2020].

IATE, 2013. “megadati”. [Online]
Disponibile su: https://iate.europa.eu/entry/result/3551299/en-es-fr-it-la-mul
[Visitato il 14 06 2020].

IBM, 2018. “The Four V's of Big Data”. [Online]
Disponibile su: https://www.ibmbigdatahub.com/infographic/four-vs-big-data
[Visitato il 14 06 2020].

IBM, 2018. “ACID properties of transactions”. [Online]
Disponibile su:
https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.4.0/product-overview/acid.html
[Visitato il 15 04 2020].

IDC, 2018. “The Digitization of the World From Edge to Core”. [Online]
Disponibile su:
https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf
[Visitato il 12 06 2020].

Injenia, 2019. “5 storie di successo di Injenia al Google Cloud Summit 2019”. [Online]
Disponibile su: https://www.injenia.it/injenia-al-google-cloud-summit-2019/
[Visitato il 14 04 2020].

Kirwin, B. & Mieritz, L., 2005. “Defining Gartner Total Cost of Ownership”. [Online]
Disponibile su:
https://www.gartner.com/en/documents/487157/defining-gartner-total-cost-of-ownership
[Visitato il 05 06 2020].

Licata, P., 2019. “Google ha raggiunto la “quantum supremacy”. Sfida con Ibm”. [Online]
Disponibile su:
https://www.corrierecomunicazioni.it/digital-economy/google-dice-di-aver-raggiunto-la-supremazia-quantistica-ibm-fattibilita-ancora-lontana/
[Visitato il 02 04 2020].

Liu, S., 2020. “Global market share held by leading desktop internet browsers from January 2015 to March 2020”. [Online]
Disponibile su:
https://www.statista.com/statistics/544400/market-share-of-internet-browsers-desktop/
[Visitato il 11 05 2020].

Meier, R., 2017. “An Annotated History of Google’s Cloud Platform”. [Online]
Disponibile su:
https://medium.com/@retomeier/an-annotated-history-of-googles-cloud-platform-90b90f948920
[Visitato il 15 03 2020].

Melendez, S., 2018. “Amid the cloud giants, small providers find their niche”. [Online]
Disponibile su:
https://www.fastcompany.com/40561868/amid-the-cloud-giants-small-providers-find-their-niche
[Visitato il 15 04 2020].

Microsoft, 2014. “Tiered Distribution”. [Online]
Disponibile su:
https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff647195(v=pandp.10)
[Visitato il 16 04 2020].

MLPerf, 2019. “MLPerf Training v0.6 Results”. [Online]
Disponibile su: https://mlperf.org/training-results-0-6/
[Visitato il 19 06 2020].

O’Dea, S., 2020. “Mobile operating systems' market share worldwide from January 2012 to December 2019”. [Online]
Disponibile su:
https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/
[Visitato il 19 04 2020].

Red Hat, 2020. “What is a data lake?”. [Online]
Disponibile su: https://www.redhat.com/en/topics/data-storage/what-is-a-data-lake
[Visitato il 14 05 2020].

Rice, B., 2020. “Google Cloud Platform Fundamentals: Core Infrastructure”. [Online]
Disponibile su: https://www.coursera.org/learn/gcp-fundamentals/
[Visitato il 07 03 2020].

Richter, F., 2020. “Amazon Leads $100 Billion Cloud Market”. [Online]
Disponibile su:
https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/
[Visitato il 11 03 2020].

Rosenbaum, M., 2015. “Welcome to the Future of CRM. Welcome to Salesforce Lightning.”. [Online]
Disponibile su:
https://www.salesforce.com/blog/2015/08/future-of-crm-salesforce-lightning.html
[Visitato il 18 06 2020].

Schwab, K., 2016. “The Fourth Industrial Revolution: what it means, how to respond”. [Online]
Disponibile su:
https://professionallearning.education.gov.scot/media/1352/the-fourth-industrial-revolution-what-it-means-and-how-to-respond-world-economic-forum.pdf
[Visitato il 02 04 2020].

Stevens, L., 2020. “Uncovering Alphabet's Next Big Winner”. [Online]
Disponibile su:
https://seekingalpha-com.cdn.ampproject.org/c/s/seekingalpha.com/amp/article/4353816-uncovering-alphabets-next-big-winner
[Visitato il 17 06 2020].

Sullivan, D., 2019. “Official Google Cloud Certified Professional Cloud Architect Study Guide”. Indianapolis: John Wiley & Sons.

The Kubernetes Authors, 2020. “What is Kubernetes?”. [Online]
Disponibile su: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/
[Visitato il 05 05 2020].

The Kubernetes Authors, 2020. “Kubernetes Components”. [Online]
Disponibile su: https://kubernetes.io/docs/concepts/overview/components/
[Visitato il 05 05 2020].

Vanian, J., 2016. “Google Doubles Down on Enterprise by Re-Branding Its Cloud”. [Online]
Disponibile su: https://fortune.com/2016/09/29/ggoogle-cloud-branding-g-suite/
[Visitato il 04 04 2020].

Wright, D., Smith, D., Bala, R. & Gill, B., 2019. “Magic Quadrant for Cloud Infrastructure as a Service, Worldwide”. [Online]
Disponibile su: https://www.gartner.com/en/documents/3947472
[Visitato il 21 06 2020].