ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Ingegneria e Architettura
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

Social Network Analysis: Architettura Streaming Big Data di Raccolta e Analisi Dati da Twitter

Tesi di laurea in
Big Data

Relatore
Prof. Enrico Gallinucci

Correlatore
Dott. Matteo Casadei

Candidato
Andrea Giannini

Quarta Sessione di Laurea
Anno Accademico 2020-2021

Sommario

Negli ultimi anni i social media, come ad esempio Facebook, Twitter, WhatsApp, YouTube, si sono diffusi a macchia d'olio. Ormai quasi tutti accedono giornalmente su almeno uno di questi per informarsi, esprimere opinioni e interagire con altri utenti. Per questa ragione sono diventati fondamentali per i reparti marketing delle aziende essendo non solo un ottimo canale di comunicazione, ma anche una fonte di informazioni sui clienti e potenziali tali. La tesi si focalizza proprio su quest'ultimo aspetto. Il progetto Social Network Analysis (SNA) vuole essere infatti uno strumento attraverso il quale è possibile visionare e analizzare per intero le reti di interazione tra utenti. Ci si è posti l'obiettivo di realizzare SNA in modo che raccogliesse e si aggiornasse in tempo reale, così da essere sempre al passo con le ultime novità, data la dinamicità delle informazioni all'interno dei social media. Un progetto come SNA comporta dover affrontare diversi ostacoli. Oltre a quello di riuscire a realizzare un'architettura che accolga un flusso continuo di informazioni, uno degli ostacoli più importanti è quello di gestire la grande mole di dati. Per farlo ci si è affidati ad un'architettura distribuita e facilmente scalabile che comprende l'uso di elaborazioni in cluster, di funzioni serverless e di database NoSQL approvvigionati attraverso il servizio cloud di Microsoft, Azure. In questa tesi SNA è stato progettato e implementato basandosi su Twitter, ma è possibile sfruttare la stessa idea su tanti altri social media.

Indice

Sommario **iii**

1 Introduzione **1**

2 Definizione del problema **3**
2.1 Marketing . . . 3
2.2 Social Media . . . 5
2.3 Business e Social Media . . . 13
2.4 Social network Analysis . . . 19

3 Stato dell'arte **23**
3.1 Big Data . . . 24
3.2 Architettura . . . 28
3.2.1 Hardware . . . 28
3.2.2 On-premise o cloud . . . 29
3.2.3 Software . . . 32
3.3 Streaming . . . 35
3.4 Database NoSQL . . . 38

4 Architettura scelta **43**
4.1 Architettura funzionale . . . 43
4.1.1 Loader . . . 43
4.1.2 Dashboard . . . 46
4.2 Architettura tecnologica . . . 48
4.2.1 Loader . . . 49
4.2.2 Dashboard . . . 54
4.3 Implementazione . . . 55
4.3.1 Loader . . . 55
4.3.2 Dashboard . . . 59

5 Test **61**

6 Conclusioni **67**

Elenco delle figure

2.1 Il modello 7P di Booms, in maiuscolo le 4P di McCarthy [2] 5
2.2 Classifica dei social media in base alla quantità di utenti attivi [3] . 6
2.3 Classifica dei social media in base alla loro popolarità [3] 7
2.4 Nido d'ape dei sette blocchi funzionali dei social media [16] 11
2.5 Esempi di equilibrio tra i sette blocchi funzionali [16] 14
2.6 I canali che legano i social media alla performance aziendale 16
2.7 Importanza dei social media nelle varie funzioni aziendali 18

3.1 Infografica del 2021 sulla quantità di dati generati in un minuto [1] 23
3.2 Infografica delle 8V dei Big Data [18] 25
3.3 Classificazione delle architetture 31
3.4 Architettura di riferimento per i big data del NIST [9] 32
3.5 Architettura di riferimento per i big data di Microsoft [11] 33
3.6 Pipeline di un'architettura streaming di big data 37
3.7 Esempio di column family 40

4.1 Architettura funzionale del Loader 44
4.2 Grafo d'esempio con tutti i tipi di nodi e archi 46
4.3 Architettura funzionale della Dashboard 46
4.4 Tipologie di grafici visualizzabili 48
4.5 Architettura tecnologica del Loader 49
4.6 Rappresentazione grafica di Azure Event Hubs 51
4.7 Diagramma per la scelta dell'API su CosmosDB 54
4.8 Architettura tecnologica della Dashboard 55
4.9 Diagramma rappresentante il flusso di dati su Spark 59

5.1 Grafico raffigurante i tempi del Loader con tweet in tempo reale, 1 partizione di EventHubs e 2 worker nel cluster 61
5.2 Grafico raffigurante i tempi del Loader con tweet da blob, 1 partizione di EventHubs e 1 solo worker nel cluster 63

5.3 Grafico raffigurante i tempi del Loader con tweet da blob, 1 partizione di EventHubs e 2 worker nel cluster 64
5.4 Grafico raffigurante i tempi del Loader con tweet da blob, 2 partizioni di EventHubs e 1 worker nel cluster 64
5.5 Grafico raffigurante i tempi del Loader con tweet da blob, 2 partizioni di EventHubs e 2 worker nel cluster 65

Elenco dei listati

4.1 Pseudocodice della funzione TwitterStreaming 56
4.2 Pseudocodice della funzione StreamingController 57

Capitolo 1

Introduzione

Il mondo del marketing ha visto negli ultimi anni l'emergere di un nuovo canale attraverso il quale comunicare con il pubblico e ricevere direttamente feedback: i social media. Come per gli altri canali di comunicazione i social media hanno una grande importanza per le aziende perché sono un ottimo mezzo sia per pubblicizzare i propri prodotti che per raccogliere informazioni grazie alle quali la dirigenza può ottimizzare le proprie decisioni aziendali. Anzi, i contenuti sui social media a volte possono diventare virali e raggiungere una diffusione molto più ampia rispetto a quella che i mezzi di comunicazione tradizionali riuscirebbero ad ottenere. Proprio per questa importanza strategica si sono sviluppati nel tempo sempre nuove metodologie per raggiungere sempre più potenziali clienti.

Spesso però i social media risultano essere molto dispersivi in quanto al loro interno sono presenti una grandissima quantità di utenti che interagiscono fra loro ininterrottamente, andando così a creare una quantità esorbitante di dati. Per una singola persona, ma anche per un team, è impossibile analizzarli tutti e di conseguenza diventa difficile avere un quadro completo delle innumerevoli informazioni ricavabili dal social media.

Il progetto esposto in questa tesi nasce proprio con l'intento di fornire al reparto marketing di un'azienda uno strumento che riesca a visionare in modo completo, preciso e in tempo reale ciò che accade su Twitter, uno dei social media più frequentati. Nello specifico, in questa tesi si è posti l'obiettivo di, una volta scelto un topic, individuare gli argomenti maggiormente discussi e gli influencer che riescono a raggiungere più persone con i propri contenuti, e quindi ad avere un maggiore impatto sull'opinione pubblica.

Data la grande quantità di dati da prendere in considerazione, per svolgere il progetto è stato necessario affrontare le varie problematiche dei big data. Una delle principali difficoltà è sicuramente la scelta dell'architettura sia hardware che software, in quanto dev'essere adatta a sopportare un'elaborazione su larga scala. Inoltre, sempre per quanto riguarda l'architettura, è necessario scegliere se gestire

l'infrastruttura personalmente oppure delegarla a terzi andando su un servizio cloud. Un'altra sfida affrontata è quella di riuscire a raccogliere i dati ed ottenere i risultati quasi in tempo reale.

Infine, una questione non da sottovalutare è la modellazione dei dati. Infatti questa può condizionare in modo sostanziale l'analisi dati e l'architettura stessa del progetto. Per questo motivo di solito viene scelta in base alle esigenze progettuali.

La tesi è composta dai seguenti capitoli:

- **Capitolo 2**: inizialmente si affronta una breve introduzione sul marketing per capire le necessità di tale reparto, ma il capitolo si focalizza soprattutto sulla spiegazione di cosa sono i social media e di come questi possono essere sfruttati per gli interessi aziendali. Viene inoltre illustrata l'importanza in questo ambito degli studi strutturali sulle reti sociali.

- **Capitolo 3**: viene esposto lo stato dell'arte per quanto riguarda i progetti di big data, passando per le diverse tematiche che toccano da vicino il progetto di questa tesi. Quindi si parla delle architetture più adatte per questo tipo di progetti, sia dal punto di vista dell'hardware che del software, delle differenze tra avere un'infrastruttura on-premise o su cloud, delle pipeline per l'elaborazione in streaming e dei vantaggi di archiviare i dati su database NoSQL.

- **Capitolo 4**: qui viene mostrata l'architettura, o meglio, la pipeline scelta che il flusso di dati da Twitter seguirà. Siccome l'architettura è possibile dividerla in due parti, Loader e Dashboard, ogni sezione di questo capitolo ha due sottosezioni, ognuna dedicata ad una delle due parti. In un primo momento viene illustrata l'architettura funzionale, cioè vengono spiegate, componente per componente, le varie funzioni che si devono svolgere nei vari passaggi. Successivamente viene spiegata l'architettura tecnologica, cioè si vanno ad esplorare i vari servizi sfruttati per adempiere le funzioni dell'architettura funzionale. Infine viene spiegato in linea di massima come sono stati implementati i vari componenti dell'architettura andando più nel dettaglio.

- **Capitolo 5**: sono riportati i risultati dei test svolti sull'architettura con diverse configurazioni e in diverse condizioni.

- **Capitolo 6**: inserito a completamento della tesi, è dedicato alle conclusioni e ai possibili sviluppi futuri.

Capitolo 2

Definizione del problema

Il progetto è nato dalla necessità di fornire degli strumenti in grado di aiutare il reparto marketing di un'azienda nella loro funzione di creare una strategia di marketing e gestire i social media. Nel farlo risulta fondamentale essere consapevoli di ciò che li circonda.

2.1 Marketing

Le aziende sono da sempre alla ricerca di nuovi metodi per raggiungere, acquisire e fidelizzare sempre più clienti. Per rispondere a tale esigenza è nato il marketing, cioè quell'insieme di metodi per la gestione sistematica e programmata dei rapporti di interazione con il mercato. Storicamente parlando, questo evento risale al momento in cui si è verificato il passaggio al mercato di massa, dove i prodotti hanno subito una forte standardizzazione. Questo ha spinto le aziende a venire incontro ai bisogni di sempre più persone e ad adottare di conseguenza un orientamento alla vendita attraverso un potenziamento dell'organizzazione, dei metodi e delle strutture commerciali. In questo contesto il marketing si pone l'obiettivo di mantenere l'equilibrio tra le esigenze del consumatore e quelle del produttore, andando a sopperire i bisogni del mercato senza compromettere la produttività aziendale.

All'interno del marketing vengono coinvolti diversi processi, ma tutti girano intorno a due aspetti essenziali [30]:

- La definizione di un target, inteso come l'insieme di potenziali clienti con caratteristiche omogenee verso cui si intende indirizzare l'offerta di prodotti o servizi;
- La scelta degli strumenti, attraverso cui raggiungere la fetta di mercato target.

La combinazione dei mezzi con cui il reparto di marketing può incidere sulle strategie aziendali viene chiamata *marketing mix*. Esso è tradizionalmente costituito dalle cosiddette 4P di E. Jerome McCarthy [21]:

- **Product**: si riferisce a ciò che l'azienda vende. Ciò può essere qualcosa di tangibile, cioè un prodotto, o meno, un servizio. Intorno ad esso si articolano diverse attività che vanno dal design del prodotto all'imballaggio.

- **Price**: si riferisce alla somma di denaro che il cliente paga per un prodotto e comprende tutte quelle attività che vanno ad incidere su di esso. Ad esempio la definizione della strategia di prezzo e gli sconti da applicare.

- **Place**: si riferisce al luogo in cui il cliente può acquistare il prodotto. Le attività che incidono su di esso sono molteplici, ad esempio la copertura territoriale, il collocamento dei negozi, la logistica e così via.

- **Promotion**: si riferisce all'insieme di attività che mirano a promuovere e pubblicizzare il prodotto o l'azienda stessa. Comprende attività come la scelta dei canali di comunicazione, la creazione del messaggio promozionale e la frequenza con cui viene comunicato.

Con il tempo sono stati aggiunti degli altri elementi, arrivando poi al modello 7P proposto da Booms e Bitner in cui oltre alle 4P appaiono Process, People, Physical evidence [7].

Alla base delle decisioni che vengono prese nelle varie attività di marketing c'è la necessità di un'interazione con il pubblico a doppio senso. Questo perché da un lato il marketing ha il compito di promuovere i prodotti della propria azienda inviando messaggi al cliente, chiunque esso sia, un commerciante il cliente finale, ecc. Dall'altro lato il marketing necessita di raccogliere informazioni e feedback dai clienti per comprendere al meglio i loro bisogni, i loro interessi e le variabili che influenzano il loro comportamento. Con la consapevolezza di queste informazioni il reparto di marketing può direzionare le attività sui punti sopracitati verso ad un maggior profitto. In entrambi i casi esistono diverse modalità e canali di comunicazione. Nella direzione business-cliente le modalità usate sono: la pubblicità, la vendita diretta, la promozione delle vendite, la propaganda e le pubbliche relazioni. Nella maggior parte dei casi avvengono attraverso i mass media come televisione, social media, cinema, radio, stampa e affissione. Nella direzione opposta invece si usano generalmente le ricerche di mercato. Vengono svolte attraverso la raccolta, l'analisi e l'elaborazione di dati di cui l'azienda è già in possesso, grazie ad esempio ad associazioni di categoria ed enti pubblici, oppure svolgendo delle indagini ad hoc. Solitamente queste ultime vengono svolte attraverso interviste dirette e sondaggi d'opinione e permettono di delineare con precisione il target di mercato.

Figura 2.1: Il modello 7P di Booms, in maiuscolo le 4P di McCarthy [2]

Per quanto riguarda il target di mercato, le imprese possono decidere di operare in due modi:

- Marketing di massa o indifferenziato: si punta su tutto il pubblico senza fare distinzioni;
- Marketing differenziato: si individuano tra i clienti dei gruppi omogenei e ci si concentra su uno o più di essi, andando a creare delle campagne pubblicitarie differenziate.

Negli ultimi anni tra i canali di comunicazione a disposizione del marketing sono diventati sempre più importanti i social media.

2.2 Social Media

Da quando è stato introdotto Internet, sin dai primi anni, sono nate piattaforme per aiutare lo scambio di informazioni e cultura, come ad esempio ARPANET. Probabilmente il primo sistema informatico che ha introdotto delle forme primitive di funzionalità dei social media è stato PLATO, nato nel 1960. Già nel 1973 offriva servizi come: un forum di messaggi (Notes), una funzione di messaggistica istantanea (TERM-talk), una chat room online (Talkomatic), un quotidiano e

blog in crowdsourcing (News Report) e delle liste d'accesso, che permettevano di condividere file di testo ad un gruppo ristretto di utenti.

Dopo l'arrivo nel 1978 del primo Bulletin Board System, un sistema telematico che permette di creare bacheche elettroniche, nascono tanti altri forum; questi con l'avvento del World Wide Web negli anni '90 si trasferirono sul web.

Con il Web iniziarono a nascere anche le prime piattaforme definibili social media, anche se ancora rudimentali. I primi sono stati GeoCities nel 1994 e Classmates.com nel 1995, ma quello che comunemente viene considerato il primo social media è SixDegrees.com, nato nel maggio 1997 [22]. Questo perché è stato il primo sito web creato per "persone reali", utilizzando i loro veri nomi, includendo profili, elenchi di amici e affiliazioni scolastiche che potevano essere utilizzati dagli utenti registrati.

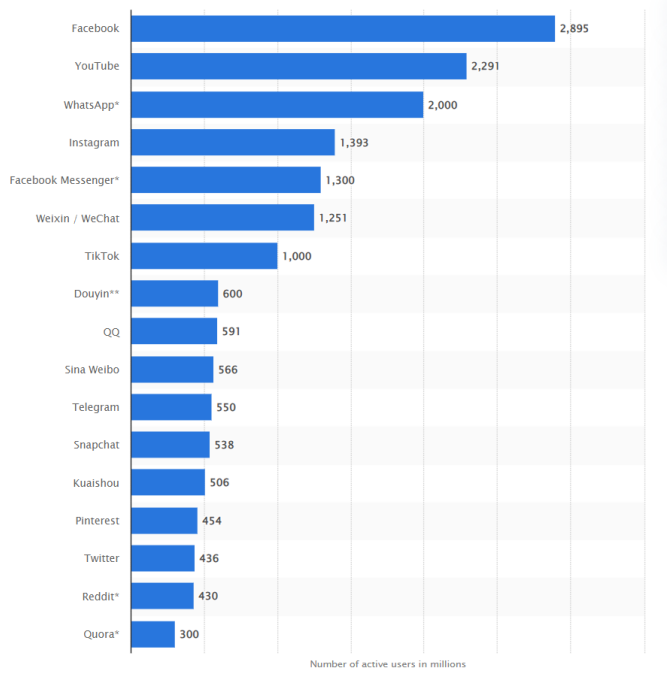

Figura 2.2: Classifica dei social media in base alla quantità di utenti attivi [3]

In realtà è durante il primo decennio del 2000, con l'emergere delle funzionalità del Web 2.0, che i social media hanno iniziato a diffondersi esponenzialmente e a diventare come oggi li conosciamo. Con Web 2.0 ci si riferisce a quei siti web che enfatizzano i contenuti generati dagli utenti, la facilità d'uso, la cultura collaborativa e l'interoperabilità. Questo, insieme al calo dei costi per l'archiviazione dei dati online, ha reso possibile per la prima volta offrire alle grandi masse l'accesso a una serie di spazi incentrati sull'utente. Infatti in questo caso è l'utente stesso a popolare questi spazi con contenuti da lui generati oppure condividendo quelli

di altri. Inoltre esso ha la possibilità di creare in svariati modi collegamenti con altri spazi andando così a formare delle vere e proprie reti sociali virtuali. Non a caso i più grandi social media sono nati proprio in questo periodo. Facendo una classifica in base alla quantità di utenti, come mostrato nella fig. 2.2, troviamo al primo posto Facebook con 2.895 milioni di utenti, seguito da YouTube (2.291 milioni) e WhatsApp (2 miliardi). Facebook è nato nel 2004, YouTube nel 2005 e WhatsApp nel 2010.

Per far capire la popolarità di queste piattaforme, una ricerca del 2015 mostra che il mondo ha trascorso il 22% del proprio tempo online sui social media, senza considerare che in questi ultimi sette anni probabilmente è ulteriormente aumentato. A ottobre 2020, fino a 4,08 miliardi di utenti di social media in tutto il mondo sono stati trovati attivi su smartphone. Questo suggerisce che la popolarità dei social media sia dovuto anche all'elevato uso quotidiano degli smartphone.

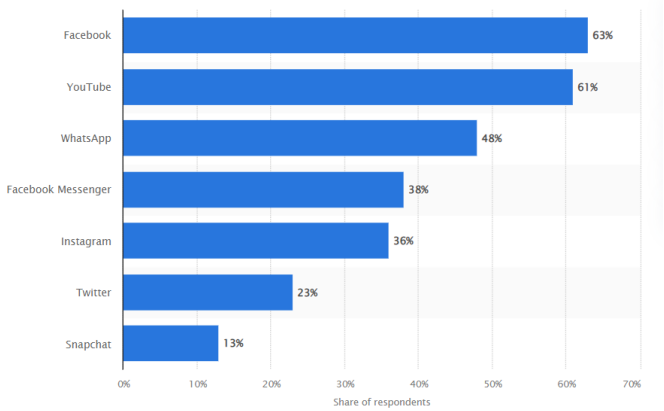

Figura 2.3: Classifica dei social media in base alla loro popolarità [3]

Nel febbraio 2020 è stato fatto un sondaggio per misurare la popolarità dei social media in cui è stato chiesto agli utenti Internet di indicare quelli visitati nell'ultima settimana. I risultati, come si vede nella fig. 2.3, mostrano come i più popolari siano sempre i tre indicati in precedenza.

La moltitudine dei social media, che siano autonomi o integrati, rende la definizione della tecnologia una sfida. Alla domanda "cosa sono i social media?" si potrebbe rispondere pensando alle piattaforme più famose come Facebook e Twitter. Ma pur essendo tra le principali piattaforme, non danno un quadro completo dei social media presenti online. Ad esempio, tra gli adolescenti si stanno diffondendo sempre più altre tipologie di social media leggermente diverse, come Snapchat e Tik Tok. Ma, mentre questi fanno parte della stessa macro categoria, ci sono anche altri social media totalmente differenti come YouTube, Wikipedia e TripAdvisor.

Ci sono due problematiche distinte associate alla concettualizzazione dei social media. In primo luogo, la velocità con cui la tecnologia si sta espandendo ed evolvendo sfida la nostra capacità di definire confini netti attorno al concetto. Le tecnologie dei social media includono un'ampia gamma di piattaforme basate su PC e dispositivi mobili che continuano a essere sviluppate, lanciate, rilanciate, abbandonate e ignorate ogni giorno nei paesi di tutto il mondo e a vari livelli di consapevolezza pubblica. In secondo luogo, i social media possono essere definiti come strumenti che facilitano varie forme di comunicazione e collaborazione. Però, se li definissimo in questo modo dovrebbero farne parte anche il telefono e l'email.

Dunque, sintetizzando le definizioni presenti in letteratura, emergono alcuni punti in comune tra gli attuali social media [23]:

1. **I social media sono (per ora) applicazioni Web 2.0.** Come detto in precedenza, prima che le applicazioni Web 2.0 diventassero popolari, il World Wide Web era principalmente un mezzo di consumo. Fino alla fine degli anni '90 l'utente poteva solo leggere ciò che altre persone avevano scritto, ascoltare e guardare clip audio e video presenti sui siti commerciali. Le applicazioni Web 2.0 hanno cambiato il nostro modo di interagire con il mondo online grazie alle nuove funzionalità che hanno reso Internet più interattivo. Tale passaggio però non deriva da un sostanziale cambiamento nella tecnologia, ma piuttosto nell'ideologia. Infatti l'utente nelle nuove applicazioni cambia ruolo e passa dall'essere un semplice consumatore del servizio a diventarne partecipante [6]. Alcuni ricercatori hanno soprannominato il nuovo ruolo dell'utente "prosumer", una combinazione di consumatore e produttore [28]. Le applicazioni sono progettate per consentire agli utenti di creare, interagire, collaborare e condividere il processo di creazione e consumo di contenuto.

2. **I contenuti generati dagli utenti sono la linfa vitale dei social media.** Mentre il Web 2.0 è l'ideologia su cui si basano social media, il contenuto generato dagli utenti è il loro carburante. Ci sono un'infinità di esempi che si potrebbero fare sui contenuti che alimentano queste piattaforme: dall'informazione più insignificante, come il "Mi Piace" ad un post su Facebook, a quella più significativa, ad esempio la pubblicazione di un video su YouTube. I social media sono popolati da un numero infinito di dati generati dagli utenti i quali collegandosi gli uni con gli altri creano una vastissima rete. Senza questi contenuti i social media sarebbero delle città fantasma, completamente vuote, proprio ciò in cui si sta trasformando MySpace.

3. **Individui e gruppi creano profili utente specifici per un social media.** Lo scheletro dei social media è il profilo utente. Il tipo di informazioni identificative richieste variano considerevolmente da servizio a servizio, ma

spesso includono la possibilità di creare un nome utente, fornire informazioni di contatto e caricare un'immagine. Il motivo per cui il profilo è così importante è dovuto al fatto che permette di creare connessioni tra gli utenti. Infatti senza informazioni identificative, trovare e connettersi con altri sarebbe complesso. Inoltre, molte funzioni dei social media non sarebbero applicabili, come ad esempio la condivisione, il "Mi Piace", il confronto tra punteggi e così via. Anche nei social media in cui gli utenti sembrano anonimi e senza profilo come Yik Yak, in realtà viene creato un profilo univoco per ogni individuo che tiene traccia della loro attività sulla piattaforma. Una pratica comune tra i social media è quella di non consentire a chi non è registrato di accedere al contenuto degli utenti registrati. Una delle poche anomalie è Wikipedia, che consente agli utenti di navigare nell'enciclopedia online gratuita e di apportare modifiche di base senza disporre di un account.

4. **I social media facilitano lo sviluppo di reti sociali online collegando un profilo con quelli di altri individui e/o gruppi**. "La natura e la nomenclatura di queste connessioni possono variare da sito a sito"[8]. Ad esempio, un metodo per creare una rete sociale online consiste nel creare un elenco di persone con cui si desidera connettersi e interagire. Su Facebook e Snapchat consiste nella lista "amici", su Twitter e Instagram sono i "follow" e su LinkedIn le "connessioni". Una volta creato un elenco di connessioni, gli utenti possono rivedere, accedere e modificare la propria rete sociale interagendo con esso. Spesso la visualizzazione dei contenuti e l'interazione sono associati alla creazione di questi elenchi, in quanto generalmente viene fornita agli utenti una forma di *homepage* che aggrega i contenuti di chi è presente nell'elenco. Tuttavia esistono anche altri social media che consentono lo sviluppo di reti sociali senza la creazione di uno di questi elenchi tradizionali. Tinder e Yik Yak, ad esempio, consentono agli utenti di interagire fra loro in base alla propria posizione.

Nonostante l'individuazione di questi punti comuni, rimane l'ambiguità della definizione. Questo soprattutto perché, oltre alla continua evoluzione delle tecnologie, le funzionalità dei social media vengono integrate in prodotti non progettati con lo specifico scopo di creare reti sociali, come ad esempio i videogiochi. Proprio per questo in alcuni articoli si cercano di suddividere in categorie, ma, sempre per lo stesso problema, anche esse risultano temporanee, ambigue e con confini "sfumati". Tredici possibili categorie potrebbero essere [31]:

- **Blog**: il termine deriva dall'unione di "web" e "log". Consiste in un elenco cronologico di post, che possono essere letti e commentati dai visitatori. Possono essere gestiti da individui o aziende. Esempi: The Huffinghton Post, The Blonde Salad, Aranzulla.

- **Reti professionali**: piattaforme nate con lo scopo di permettere agli individui di stabilire e mantenere contatti professionali. Gli utenti creano un profilo personale in cui condividono informazioni come la loro formazione, l'esperienza professionale e le conoscenze specialistiche. Mentre le aziende le utilizzano per cercare nuovi dipendenti o esperti. Esempi: LinkedIn, XING.

- **Progetti di collaborazione**: riunisce utenti al fine di pianificare, sviluppare, migliorare, analizzare e testare progetti. Esempi: Wikipedia, Mozilla.

- **Social network aziendali**: sono social network aperti alla registrazione solo per i dipendenti di un'azienda o di un gruppo specifico. Esempi: Yammer, SocialCast.

- **Forum**: sono piattaforme di discussione in cui gli utenti possono porre e/o rispondere alle domande di altri utenti e scambiare pensieri, opinioni o esperienze. La comunicazione non avviene in tempo reale, come in una chat, ma è ritardata e solitamente visibile al pubblico. Esempi: IGN Boards, XDA-Developers, Stack Overflow.

- **Microblog**: sono simili ai blog, ma limitano la lunghezza dei post. Gli utenti possono iscriversi alle notizie di altri utenti, aziende, marchi o celebrità. Esempi: Twitter, Tumblr.

- **Siti di condivisione foto**: sono piattaforme che permettono il caricamento, la gestione e la condivisione di foto. Spesso le foto possono essere modificate online, organizzate in album e commentate da altri utenti. Esempi: Instagram, Flikr, Photobucket.

- **Siti per recensioni di prodotti e servizi**: sono piattaforme che forniscono informazioni sui prodotti. Gli utenti possono scrivere e leggere le recensioni dei prodotti o servizi. Esempi: TripAdvisor, Amazon.

- **Social bookmarking**: sono siti che permettono di salvare e organizzare dei segnalibri Internet su una piattaforma centralizzata per condividerli con amici e altri utenti. Esempi: Delicious, Pinterest, Reddit.

- **Social gaming**: sono giochi online che consentono o richiedono l'interazione sociale tra i giocatori. Esempi: World of Warcraft, League of Legends.

- **Social network**: sono piattaforme nate con l'obiettivo di connettere persone che si conoscono, condividono interessi comuni o vorrebbero impegnarsi in attività simili. Le aziende utilizzano i social network creando un profilo aziendale per posizionare determinati marchi, per informare e supportare i clienti e per acquisirne di nuovi. Esempi: Facebook, Google+.

- **Siti di condivisione video**: sono piattaforme di condivisione video che consentono agli utenti di caricare e condividere video personali o aziendali. La maggior parte di queste offre l'opportunità di commentare video. Esempi: YouTube, Vimeo, Twitch.

- **Mondi virtuali**: sono piattaforme che ospitano mondi virtuali popolati da molti utenti che possono creare un avatar personale, esplorare in contemporanea il mondo virtuale, partecipare alle sue attività e comunicare con gli altri utenti. A differenza dei videogiochi classici, il tempo continua anche quando l'utente non ha effettuato l'accesso. Esempi: VR Chat, Second Life.

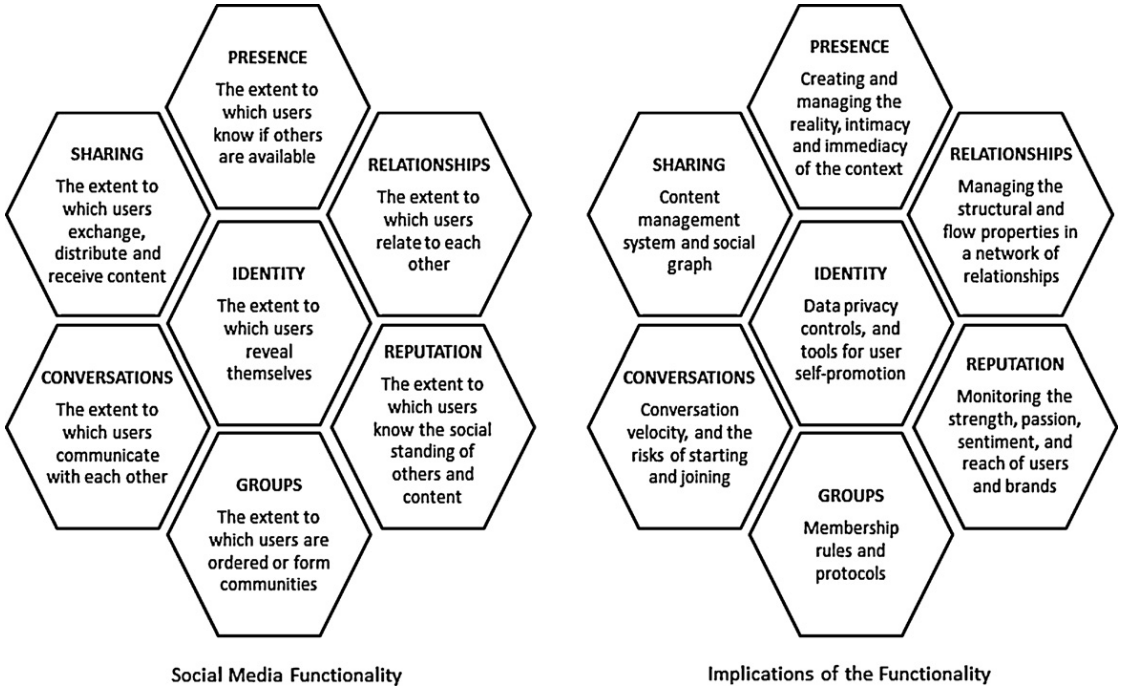

Figura 2.4: Nido d'ape dei sette blocchi funzionali dei social media [16]

Esaminando l'ecosistema dei social media si possono riconoscere sette blocchi funzionali su cui sono costruiti [16]:

- **Identità**: rappresenta la misura in cui gli utenti rivelano la propria identità in un social media. Ciò può includere la divulgazione di informazioni come nome, età, sesso, professione, posizione, ma anche informazioni soggettive come pensieri, sentimenti, simpatie e antipatie. Per molte piattaforme è il pilastro principale e tutto il resto viene costruito intorno ad esso.

- **Conversazioni**: rappresenta la misura in cui gli utenti comunicano con altri utenti in un social media. Molti siti sono progettati principalmente per facilitare le conversazioni tra individui e gruppi. Queste conversazioni possono verificarsi per qualsiasi motivo, dall'incontrare nuove persone al diffondere il proprio pensiero.

- **Condivisione**: rappresenta la misura in cui gli utenti scambiano, distribuiscono e ricevono contenuto. I social media sono costituiti da persone collegate da qualcosa che condividono; può trattarsi di un video, un pensiero, la propria posizione geografica, oppure uno sconto, come nel caso di Groupon. Di per sé la condivisione è un modo di interagire nei social media, ma a seconda dell'obiettivo funzionale della piattaforma, può anche portare gli utenti a voler conversare o addirittura a costruire relazioni tra loro. Ad esempio, gli oggetti della socialità sono le immagini per Flickr e le carriere per LinkedIn.

- **Presenza**: rappresenta la misura in cui gli utenti possono sapere se altri utenti sono "accessibili". Può includere la localizzazione nel mondo virtuale e/o reale degli utenti e se sono disponibili o meno. Ultimamente, data la crescente connettività delle persone in movimento, questa presenza collega il virtuale con la realtà. Ad esempio ci sono applicazioni che permettono la condivisione della propria posizione fisica in tempo reale. Un altro esempio è Trapster, un sistema di condivisione delle posizioni degli autovelox che si basa sulle segnalazioni degli utenti.

- **Relazioni**: rappresenta la misura in cui gli utenti possono essere connessi ad altri utenti. Per "relazione", si intende che due o più utenti hanno una qualche forma di associazione che li porta a conversare, condividere oggetti di socialità, incontrarsi o semplicemente elencarsi come amici o fan. Di conseguenza, il modo in cui gli utenti di una piattaforma di social media sono collegati spesso determina il cosa e il come dello scambio di informazioni. In alcuni casi, queste relazioni sono abbastanza formali, regolamentate e strutturate. Ad esempio LinkedIn consente agli utenti di vedere come sono collegati agli altri e quanti gradi di separazione li divide da un altro membro, magari un datore di lavoro che vorrebbero incontrare. Naturalmente, la crescita di una rete il più ampia possibile riduce probabilmente i gradi di separazione di questi individui. In altri casi, le piattaforme di social media sono incentrate sul mantenimento delle relazioni esistenti, non sull'espansione, ad esempio le piattaforme dedicate alla messaggistica come Skype. In genere i social media che non apprezzano l'identità, non apprezzano neppure le relazioni.

- **Reputazione**: rappresenta la misura in cui gli utenti possono identificare la posizione ricoperta dagli altri, inclusi se stessi, all'interno di un social media. La reputazione può avere significati diversi sulle piattaforme dei social media. Nella maggior parte dei casi è una questione di fiducia. Per determinare l'affidabilità i social media si affidano a strumenti non infallibili che aggregano automaticamente le informazioni generate dagli utenti. Ad esempio, in molti casi può essere determinata l'affidabilità in base alla quantità di amici o follower. Tuttavia, nei social media, la reputazione non si riferisce solo alle persone, ma anche ai loro contenuti, spesso valutati utilizzando sistemi di votazione. Ad esempio, su YouTube, la reputazione dei video potrebbe essere basata sulla quantità di visualizzazioni o di valutazioni positive, mentre su Facebook potrebbe essere basata sui "Mi piace".

- **Gruppi**: rappresenta la misura in cui gli utenti possono formare comunità e sotto-comunità. Esistono due tipi principali di gruppi. In primo luogo, le persone possono ordinare i loro contatti e inserire i loro amici, follower o fan in diversi gruppi auto-creati. In secondo luogo, i gruppi online possono essere analoghi ai club del mondo reale: aperti a chiunque, chiusi (approvazione richiesta) o segreti (solo su invito). Ad esempio, Facebook e Flickr hanno gruppi con amministratori che gestiscono il gruppo, approvano i candidati e invitano altri a partecipare.

Ciò non significa che in un'attività sui social media debbano essere coinvolti tutti i blocchi, ma nemmeno che questi si escludano a vicenda.

Inoltre, esaminando diversi social media, è possibile notare che ognuno di essi ha il proprio equilibrio tra i diversi blocchi funzionali. Alcuni si concentrano maggiormente sull'identità, altri sulla condivisione e così via. Nessuno dei principali social media si concentra esclusivamente su un solo blocco. Nella fig. 2.5 ci sono quattro esempi: LinkedIn, Foursquare, YouTube e Facebook. Più scuro è il colore di un blocco, maggiore è la sua importanza all'interno del social media.

2.3 Business e Social Media

Negli ultimi anni le aziende hanno visto nei social media una grande opportunità di marketing e di business grazie alla loro grande diffusione e alla possibilità di avere una comunicazione a due vie. Sono sempre di più le attività della catena del valore aziendale che necessitano tale mezzo. Nel marketing, i social media non sono solo un canale opzionale del marketing mix, bensì un elemento fondamentale all'interno della strategia commerciale.

Per fare un esempio di successo, nel 2010, American Express ha ideato e promosso il primo Small Business Saturday, una festa americana dedicata allo shop-

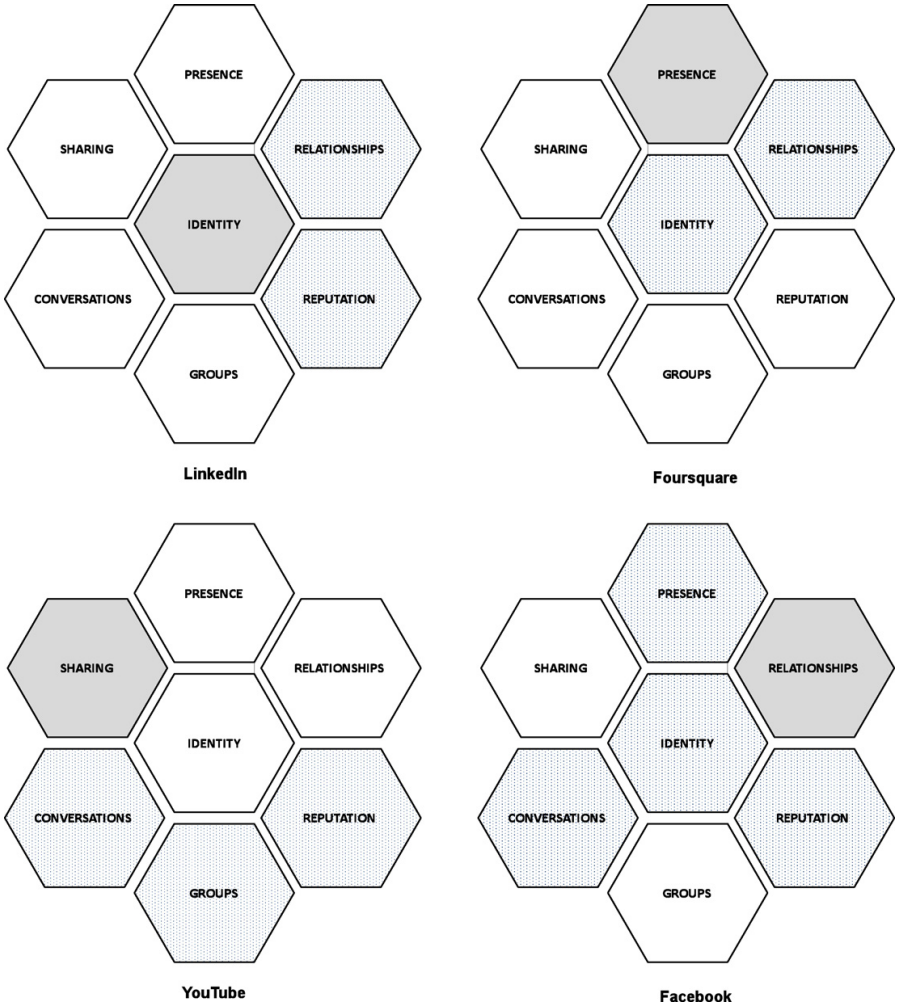

Figura 2.5: Esempi di equilibrio tra i sette blocchi funzionali [16]

ping che si tiene il sabato dopo il Giorno del Ringraziamento. La sua campagna pubblicitaria ha coinvolto principalmente i social media, generando oltre un milione di "Mi piace" su Facebook e quasi 30 mila tweet. Dopo la campagna, il 40% del pubblico era a conoscenza di Small Business Saturday e i ricavi per le piccole imprese sono aumentati del 28%. Tre anni dopo, Small Business Saturday ha generato vendite per 5,5 miliardi di dollari ed è diventata un'iniziativa globale.

Tuttavia, addentrandosi nel mondo dei social le aziende possono incontrare una serie di difficoltà e affrontare rischi sia reputazionali che economici. Una delle difficoltà è proprio la rapidità di questo mondo: gli utenti si aspettano che le loro domande ricevano risposta entro poche ore. Se le aziende ignorano i loro utenti o reagiscono troppo tardi alle critiche, ciò può evolvere in discussioni globali sui

punti deboli dell'azienda stessa o dei suoi prodotti, che alla fine si traducono in un danno economico per l'azienda. Quindi non è sufficiente semplicemente "essere lì" e avere un account sui social media. I profili e i siti Web devono essere aggiornati regolarmente ed essere altamente reattivi alle richieste dei clienti affinché la loro presenza sia veramente efficace. Inoltre, quando le aziende muovono i loro primi passi nel mondo digitale, tendono a fare affidamento su schemi tradizionali di marketing e comunicazione, che spesso non hanno effetti positivi, ma addirittura negativi in termini di reputazione. Le aziende dovrebbero quindi adottare modalità adeguate al mezzo e innovative per poter raggiungere nuovi potenziali clienti.

Ci sono numerosi esempi sugli effetti negativi dei social sulle aziende. Nel luglio 2012, un utente che impersonava un ministro russo attraverso numerosi retweet ha fatto rimbalzare in borsa i *futures* sul petrolio greggio di oltre 1$. Nell'ottobre 2012, Google ha dovuto interrompere la negoziazione delle sue azioni dopo che una fuga di notizie sul rapporto degli utili della società è diventata virale.

Proprio per tutte queste sfide le aziende si trovano spesso nelle sabbie mobili dei social media, costretti a prendere decisioni senza una chiara comprensione degli effetti. Per concettualizzare i meccanismi che influiscono sulle prestazioni aziendali bisogna vedere i social media come risorse da trasformare in capacità, come ad esempio capacità di vendita e reputazione. Più sarà l'efficienza di questo processo, maggiore sarà la prestazione aziendale. Come visto nel capitolo precedente, i social media possono essere scomposti in sette moduli funzionali: identità, conversazioni, condivisione, presenza, relazioni, reputazione e gruppi. Ognuno di essi è una risorsa offerta dai social media. Per quanto riguarda invece le capacità, quelle che subiscono un impatto maggiore sono:

- La performance finanziaria: include il livello e la crescita delle vendite, la redditività e il prezzo delle azioni.

- La performance operativa: si concentra sulla posizione delle azioni, sull'introduzione di nuovi prodotti, sulla loro qualità, sull'efficienza operativa e sulla soddisfazione del cliente.

- La performance sociale d'impresa: dipende in gran parte dalla capacità dell'azienda di stabilire relazioni oneste con la società, ponendo particolare attenzione alla reputazione e al marchio.

I social media influenzano le prestazioni aziendali attraverso quattro canali [26]: capitale sociale, preferenze pubbliche, social marketing e social networking aziendale. Ogni canale convoglia una serie di risorse di social media in un dominio di prestazioni aziendali. Questi canali di performance sui social media non si escludono a vicenda né sono tutti simultaneamente presenti.

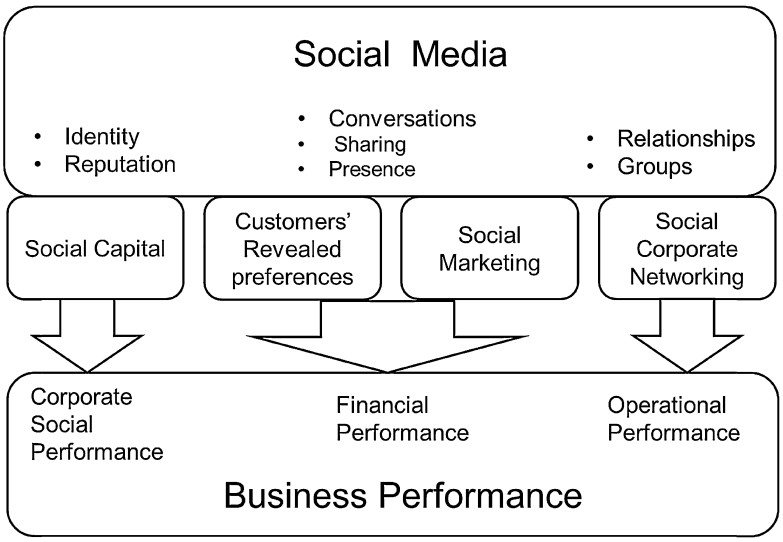

Figura 2.6: I canali che legano i social media alla performance aziendale

Capitale sociale Il canale del capitale sociale rappresenta la misura in cui i social media influenzano le relazioni delle imprese con la società. Attraverso questo canale le risorse di identità e reputazione vengono trasformate nella capacità di performance sociale d'impresa. Il capitale sociale dell'azienda, cioè le relazioni affidabili guadagnate attraverso l'identità e la reputazione, è modellato attraverso l'attività su piattaforme come Wikipedia, blog e motori di ricerca. Nell'ambiente odierno, le aziende non devono solo avere trasparenza, ma anche impegnarsi socialmente per costruire relazioni affidabili che abbiano un impatto sulla performance sociale d'impresa. Molti studi hanno dimostrato che i social media hanno un impatto diretto su di essa. Ad esempio, esaminando i contenuti generati dagli utenti su TripAdvisor, O'Connor ha dimostrato che l'immagine di un hotel può essere gestita dalle opinioni dei clienti sul Web [24]. Il canale del capitale sociale può avere anche un effetto indiretto sulla performance finanziaria o operativa nel lungo periodo. Questo perché comporta implicazioni sulle aree della gestione del marchio e delle relazioni istituzionali con gli stakeholder. È un canale passivo, il che significa che le aziende, generalmente, non possono allocare budget specifici per controllare questo canale, poiché dipende dalla percezione che i social media costruiscono attorno all'azienda.

Preferenze pubbliche dei clienti Il canale delle preferenze pubbliche dei clienti rappresenta la misura in cui i social media espongono i gusti dei clienti. Attraverso esso, le risorse di conversazione, condivisione e presenza vengono convogliate nelle capacità finanziarie. Attraverso siti come Twitter o Facebook, i potenziali

clienti esprimono i loro gusti razionalizzando le azioni osservate nella realtà. Questo canale influisce principalmente sulla performance finanziaria. È dimostrato che gli indicatori finanziari, come i prezzi delle azioni, dipendono in gran parte dalle informazioni del mercato e dalle aspettative sull'impresa. Infatti la conoscenza collettiva sui social media negli ultimi anni è stata utilizzata per prevedere diversi dati del mondo reale, come ad esempio i tassi di disoccupazione, il monitoraggio di una malattia, gli incassi al botteghino, le vendite di automobili, la pianificazione della destinazione di viaggio e la fiducia dei consumatori. Altre aree finanziarie che hanno implicazioni da questo canale sono quelle di gestione strategica e introduzione di nuovi prodotti. Infatti attraverso "Mi piace" e "follow", le aziende possono stimolare il mercato e anticipare la domanda di prodotti e servizi. Questo canale è particolarmente rilevante per gli azionisti e i gestori di portafogli di investimento, poiché ha un impatto diretto sul valore delle azioni. Il canale delle preferenze pubbliche dei clienti è in gran parte passivo, perché le azioni di gestione e il budget hanno un effetto limitato sui giudizi dei clienti. In compenso le aziende possono sfruttare questo canale raccogliendo informazioni sui clienti o potenziali tali per prendere decisioni aziendali più proficue.

Social Marketing Il canale di social marketing rappresenta la misura in cui le risorse di conversazione, condivisione e presenza vengono trasformate in capacità di performance finanziaria. Attraverso queste risorse infatti su Facebook, YouTube o Twitter, le aziende commercializzano attivamente i loro prodotti e servizi. Come detto in precedenza, le aziende hanno adottato i social media come parte essenziale del loro marketing mix. Tuttavia, le tattiche e gli obiettivi degli strumenti pubblicitari dei media tradizionali (es. televisione, radio, stampa, cartelloni pubblicitari) differiscono sostanzialmente da quelli dei social media. Il marketing tradizionale viene fornito direttamente dal venditore e implica consapevolezza, conoscenza e ricordo. D'altra parte, i social network, i blog, i microblog e le comunità si avvicinano ai clienti con mezzi interattivi come conversazione, condivisione, collaborazione e coinvolgimento. L'uso dei social media nel marketing in questo caso è innovativo solo nei suoi mezzi, ma non nei suoi obiettivi. Questi nuovi strumenti di marketing hanno però anche dei punti negativi. A differenza della pubblicità tradizionale, di cui si conosce bene l'impatto aziendale che può dare, con il social media marketing non è così chiaro. Inoltre il social media marketing ha anche un rischio reputazionale maggiore rispetto al marketing tradizionale, sopratutto a causa della velocità di questo mondo. Nonostante questo canale abbia il maggiore impatto sulla performance finanziaria, varie ricerche mostrano come l'investimento nel marketing online sia correlato anche alle prestazioni operative, come il contatto e l'impegno con i clienti. A differenza dei precedenti canali questo è un canale attivo e pertanto richiede un'attenta allocazione del budget.

Social networking aziendale Il canale di social networking aziendale rappresenta la misura in cui le risorse social dell'azienda, come relazioni e gruppi, vengono trasformate in capacità di prestazioni operative. Il social networking aziendale si riferisce ai legami informali del personale aziendale costruiti sui social network. Questo canale coinvolge un diverso insieme di social network, come LinkedIn o ResearchGate, mirati a creare reti professionali e accademiche. Le piattaforme social online forniscono un modo di comunicare a basso costo e altamente accessibile, che consente relazioni con le persone sia all'interno che all'esterno dell'organizzazione. Ad esempio le piattaforme social possono supportare le attività anche attraverso discussioni online, condivisione di conoscenze e ricerca di clienti. Le reti interaziendali aumentano la mobilità del lavoro tra le aziende, fornendo modi efficienti per rivolgersi ai migliori professionisti per le offerte di lavoro. Invece le reti intra-aziendali aiutano a identificare preziose competenze all'interno dell'azienda. Il social networking aziendale è un canale attivo che influisce sulle relazioni con clienti e fornitori. È rilevante per la creazione di capacità relative ai clienti e alle risorse umane.

	Corporate function					
Type of social media	R&D	Marketing	Customer service	Sales	HR	Organisation
Blogs	◔	◑	◔			
Business networks					●	◔
Collaborative projects	●					
Enterprise social networks	◕				◑	●
Forums	◑	◔	●			
Microblogs		◕	◔		◔	
Photo sharing		◑				
Products/services review	◔	◑		●		
Social bookmarking		◑				
Social gaming		◑				
Social networks	◔	●	◑		◔	◔
Video sharing		●	◔			
Virtual worlds	◔	◕		◔		

Importance: (empty) none or almost none; ◔ low; ◑ medium; ◕ high; ● very high

Figura 2.7: Importanza dei social media nelle varie funzioni aziendali

Ovviamente non tutti i tipi di piattaforme sono rilevanti per il business. Prendendo come riferimento la categorizzazione del capitolo precedente, i social network, le piattaforme di condivisione video e le reti professionali sono di grande interesse. Invece altri tipi, come la condivisione di foto, il social bookmarking o il social gaming, potrebbero essere meno importanti e di minore interesse. Anche i siti web che si occupano di recensioni sono molto importanti per le aziende perché sempre più clienti si informano online sulle caratteristiche e prestazioni dei prodotti. Tuttavia, tali siti web, ma anche i blog, i forum e i social network, sono un veicolo di apprendimento per le aziende, poiché i clienti condividono le loro esperienze individuali positive e negative o forniscono soluzioni ai problemi che hanno riscontrato. Anche se questi social media forniscono informazioni preziose per la ricerca e lo sviluppo, i progetti di collaborazione e i social network aziendali sono la fonte primaria per lo sviluppo di nuovi prodotti. Quando si tratta di reclutamento, le reti professionali sono la fonte più importante. Tuttavia un numero crescente di aziende pubblica determinate offerte di lavoro sul proprio microblog o sui social network, soprattutto se il target è giovane e con un'elevata affinità con Internet, la tecnologia e i social media.

2.4 Social network Analysis

È evidente che i social media formano una grande rete composta dagli utenti che attraverso essi partecipano e interagiscono fra loro. Ma non solo: all'interno della rete possono essere tenuti in considerazione anche i vari dati da essi generati. Per questo motivo una delle scienze più importanti per analizzare i dati raccolti nei social media è la social network analysis.

La social network analysis è un settore della sociologia emerso nell'ultimo secolo che si occupa di analizzare strutturalmente l'insieme delle relazioni sociali. Si differenzia in particolar modo dalla sociologia in quanto quest'ultima si è sempre focalizzata principalmente sul comportamento degli individui trascurandone la parte sociale. Per quest'ultima si intende quella parte che riguarda i modi in cui gli individui interagiscono e l'influenza che hanno l'uno sull'altro. Invece nella social network analysis si ha un approccio strutturale che vede le relazioni come suo principale oggetto di ricerca. Questo è un tipo di approccio che si applica a quasi tutti i campi della scienza, basti pensare che ad esempio gli astrofisici studiano come i corpi celesti si condizionano tra loro attraverso la forza gravitazionale. Inoltre, un'altra particolarità della social network analysis sono gli attori tenuti in considerazione all'interno della rete sociale. Essi infatti potrebbero non essere solo esseri umani, bensì comprendere anche tanti altri attori, come ad esempio istituzioni, gruppi, documenti, comunicazioni, connessioni telefoniche, virus, ecc. L'approccio della rete sociale si basa sulla nozione intuitiva secondo la quale il

pattern dei legami sociali, in cui sono inseriti gli attori, ha conseguenze importanti su di essi. Gli analisti di rete, quindi, cercano di scoprire vari tipi di pattern, di determinare in quali condizioni sorgono e di scoprirne le conseguenze.

Prima che emergesse la moderna social network analysis, i ricercatori hanno utilizzato diversi approcci per condurre ricerche strutturali sui fenomeni sociali. Ma solo di recente i vari approcci sono stati integrati in un paradigma organizzato per la ricerca. Grazie a questo si può dire che la moderna social network analysis è basata su [14]:

1. Un'intuizione strutturale basata su legami tra attori sociali considerati quindi unità interdipendenti, piuttosto che indipendenti o unità autonome;

2. Dati empirici di tipo "da attore ad attore" permettendo così un esame sistematico dei pattern sociali presenti;

3. L'uso di rappresentazioni grafiche dei pattern;

4. L'uso di modelli matematici e/o computazionali.

Per quanto riguarda le origini di questa branca della sociologia i ricercatori sono divisi in due linee di pensiero. Alcuni sostengono che tutto abbia avuto inizio nei primi anni '30 con il lavoro di Jacob Levi Moreno. Precisamente nel 1934, quando viene pubblicata l'introduzione di Moreno alla sociometria, "Who Shall Survive?". Fu un punto di svolta per lo sviluppo del settore. Invece altri ricercatori sostengono che la social network analysis non sia iniziata fino all'inizio degli anni '70, quando Harrison White ha iniziato a formare studenti laureati ad Harvard. In quell'epoca, White, insieme ai suoi studenti, produsse un numero incredibile di importanti contributi alla teoria e alla ricerca sulle reti sociali. L'analisi della rete contemporanea non sarebbe mai potuta emergere senza quei contributi. In ogni caso entrambi gli eventi sono stati fondamentali per lo sviluppo degli studi sulla social network analysis.

La causa principale per cui la ricerca sulle reti sociali fino a pochi anni fa era di nicchia dipendeva dall'enorme dispendio di tempo e denaro che implicavano tali ricerche. Il problema maggiore era la raccolta dei dati perché implicava sottoporre molti sondaggi e questionari a tutti i componenti di una rete sociale, o meglio, una porzione di essa. Questo perché sarebbe impossibile indagare su tutta la rete sociale o isolare una parte di essa in modo da non avere influenze esterne. Oltre a questo aspetto un'altra problematica era l'individuazione di pattern, in quanto andrebbero tenuti in considerazione tutti i dati raccolti e tutti i legami che si possono dedurre. Al giorno d'oggi, invece, grazie all'evoluzione tecnologica, queste ricerche sono molto più semplici e meno dispendiose. Infatti, grazie ai nuovi sistemi di comunicazione, la raccolta dati è molto più semplice, mentre, per merito

della capacità di calcolo, la ricerca di pattern e il calcolo di statistiche sociali è molto più veloce. Inoltre, molti dati sono già disponibili nei vari database e con l'arrivo del web 2.0 e dei social media sono gli utenti stessi a pubblicarli di loro iniziativa. I dati raccolti ormai possono essere visualizzati e analizzati come una rete di relazioni premendo semplicemente un pulsante.

Gli elementi costitutivi delle reti sociali sono:

- I nodi che compongono la rete. Possono essere singoli individui, gruppi, istituzioni, documenti, post, ecc.
- Le relazioni che legano tra loro i nodi della rete sociale. Possono essere monodirezionali, bidirezionali o multidirezionali. Le relazioni vengono solitamente rappresentate da linee e frecce.

Intorno alle reti sociali sono nate svariate metriche:

- Grandezza: numero di soggetti presenti nella rete presa in considerazione.
- Omofilia: la misura in cui gli attori formano legami con altri simili o dissimili. La somiglianza può essere definita da sesso, razza, età, occupazione, rendimento scolastico, status, valori o qualsiasi altra caratteristica saliente.
- Multiplessità: il numero di ruoli o di relazioni che connettono tra loro le persone (esempio padre, fratello, vicino di casa, collega di lavoro, ecc.).
- Reciprocità: la misura in cui due attori si scambiano reciprocamente l'amicizia o un'altra interazione.
- Buchi strutturali: l'assenza di legami tra due parti di una rete.
- Ponte: un individuo i cui legami riempiono un buco strutturale, fornendo l'unico collegamento tra due individui o gruppi.
- Centralità: si riferisce a un gruppo di metriche che mirano a quantificare "l'importanza" o "l'influenza" in vari sensi di un particolare nodo (o gruppo) all'interno di una rete.
- Densità: indica il rapporto tra il numero di legami effettivi ed il numero di legami possibili.
- Distanza: indica il numero minimo di legami necessari per connettere tra loro due nodi della rete.
- Raggruppamenti: possibilità di suddividere la rete sociale in sottounità di legami.

- Coefficiente di raggruppamento: misura che indica la probabilità che due associati di un nodo siano associati.

Capitolo 3

Stato dell'arte

Negli ultimi anni l'espansione dell'utilizzo dei social media, insieme alla forte diffusione di dispositivi appartenenti al mondo dell'*Internet of Things*, hanno portato all'accumulo e al consumo di una quantità esorbitante di dati. Basti pensare che già nel 2012 il volume dei dati all'interno dei server di Facebook aumentava 500 terabyte al giorno. Nella fig. 3.1 ci sono altri esempi che possono dare un'idea di

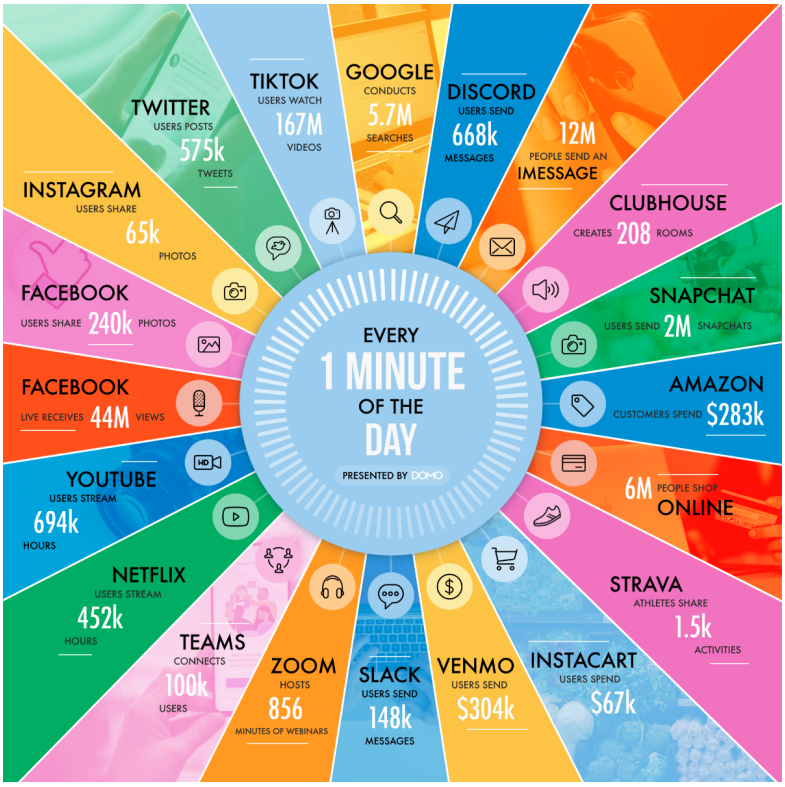

Figura 3.1: Infografica del 2021 sulla quantità di dati generati in un minuto [1]

quanti dati vengono generati in un solo minuto. Quando si parla di raccolte dati di tale entità si può dire di essere davanti ad un big data.

Siccome il progetto Social Network Analysis si basa sui dati derivanti da Twitter, quindi un social media, anche esso può essere considerato un progetto su big data.

3.1 Big Data

Ma qual è il volume di dati oltre al quale una raccolta può essere definita big data? In realtà non c'è un vero e proprio confine e anche per quanto riguarda la definizione non ne esiste una precisa. Secondo un articolo del 2011 su Teradata Magazine, si parla di big data quando "i dati superano la portata degli ambienti hardware e degli strumenti software comunemente utilizzati per raccoglierli, gestirli ed elaborarli in un tempo tollerabile per i suoi utenti". Un'altra definizione è quella data dal The McKinsey Global Institute: "I big data sono set di dati la cui dimensione supera la capacità dei tipici strumenti software di database di acquisire, archiviare, gestire e analizzare" [19]. Nel complesso si è di fronte ad un big data quando si superano le normali capacità di elaborazione, sia per quanto riguarda il volume, sia per le operazioni input/output per secondo da fare.

Nel 2001 Doug Laney ha ideato il "modello delle 3V", cioè un modello di crescita secondo cui i big data con il passare del tempo si ingrandiscono su tre dimensioni [17]:

- **Volume**: si riferisce alla quantità di dati, sia che essi siano stati generati dagli utenti che da macchine in maniera automatica.
- **Velocità**: si riferisce alla velocità con cui i nuovi dati vengono generati. È necessario l'utilizzo di strumenti in grado di garantire una velocità adeguata di immagazzinamento ed elaborazione.
- **Varietà**: si riferisce alla diversità del dato all'interno dei big data, sia per quanto riguarda il formato che la sua struttura. Infatti nei big data non vengono tenuti in considerazione solo i dati strutturati come quelli dei database relazionali, ma anche quelli semi-strutturati e non strutturati.

Poi con il tempo sono state aggiunte delle altre V, diventando prima quattro, con la veridicità, e poi cinque, con il valore [15].

- **Veridicità**: si riferisce alla accuratezza dei dati raccolti. Fa parte delle V dei big data perché in questo caso non va data per scontata, in quanto, vista la varietà dei dati d'origine e la velocità alla quale possono variare, è molto probabile che non si riesca a garantire la stessa qualità di dati dei

sistemi tradizionali. È possibile ad esempio che questi siano inconsistenti, ambigui, obsoleti e approssimativi. Inoltre visto che le decisioni aziendali possono basarsi su i risultati delle analisi eseguite a partire da questi dati è importante che siano il più accurati possibile.

- **Valore**: si riferisce alla capacità di trasformare i dati in valore. Siccome avviare un progetto di big data può comportare un investimento importante, è fondamentale valutare e documentare quale sia il valore effettivo portato al business da esso.

Successivamente sono state aggiunte altre caratteristiche e sono diventate 8V, ma a questo punto è diventata più una sfida commerciale:

- **Visualizzazione**: si riferisce all'aspetto visuale, cioè a come i dati e i risultati delle analisi vengono rappresentati. È importante che siano efficaci perché in base ad essi si prendono decisioni.
- **Viscosità**: si riferisce alla resistenza che si affronta nell'estrarre dai dati d'origine informazioni importanti per il business.
- **Viralità**: si riferisce alla capacità e alla velocità di diffusione dei risultati delle analisi.

Figura 3.2: Infografica delle 8V dei Big Data [18]

Nei progetti di big data il dato ha un proprio ciclo di vita all'interno del quale subisce vari processi, diventando prima informazione e poi conoscenza. Questo ciclo di vita, chiamato anche *pipeline*, può essere considerato un flusso composto dal susseguirsi di diversi step che raffinano sempre di più il dato e che solitamente avvengono nello stesso ambiente. La suddivisione del ciclo di vita ad alto livello è questa:

1. **Acquisizione**: come suggerisce il nome, questo step consiste nell'entrata dei dati all'interno della piattaforma. I dati potrebbero essere generati sia da macchine, come nel caso dei dispositivi dell'Internet of Thing, che da esseri umani, come nel caso dei social media, ma anche da entrambi. Spesso, siccome i dati generati sono tanti, è necessario preoccuparsi sin dal primo step che non siano troppi. Questo perché potrebbero superare le capacità dell'architettura, sia al livello di archiviazione, che al livello di elaborazione. Quindi all'interno di questo passaggio ci sono degli altri sotto-step:

   - Selezione: per ridurre la quantità di dati da elaborare ed immagazzinare può essere eseguita una selezione sui dati in entrata così da tenere solo quelli di valore e ridurre il carico di lavoro sin da subito.
   - Filtraggio e compressione: sempre per lo stesso obiettivo è possibile ridurre la dimensione dei dati d'origine. Con il filtraggio si eliminano i valori non necessari all'interno del dato, mentre con la compressione si cambia il suo formato in uno che occupi meno spazio e che sia più efficiente.
   - Generazione di metadati: in questo step è possibile creare automaticamente i metadati per ogni dato. Questi descrivono che cos'è e come è stato generato il dato. Inoltre vengono trasportati e aggiornati lungo la pipeline per preservare l'interpretazione del dato e tracciare possibili errori. Tale tecnica viene chiamata *Data Lineage*.

2. **Estrazione**: solitamente i dati acquisiti non sono subito pronti per affrontare il processo d'analisi. È dunque fondamentale che da essi vengano estratte le informazioni necessarie per i passaggi successivi. Questo avviene attraverso:

   - Trasformazione e normalizzazione: i dati vengono manipolati in modo da estrarne le informazioni e ricondurli alla struttura adatta per l'analisi.
   - Pulizia e gestione errori: in questo passaggio vengono identificati e gestiti i dati che possiedono dei difetti. Infatti questi potrebbero essere non accurati, inaffidabili e obsoleti. Ciò può succedere ad esempio a causa di sensori fallati e di dati parziali o troppo vecchi per poter essere tenuti in considerazione.

3. **Integrazione**: spesso i dati raccolti sono legati ad altri dati appartenenti a dataset differenti, quindi può essere fondamentale integrarli tra loro. Purtroppo a causa della grande quantità e della forte evoluzione del dato, non è possibile utilizzare l'approccio di integrazione classico dei data warehouse. Tale processo richiederebbe troppo tempo e da lì a poco potrebbe mutare la struttura del dato. Per questo motivo è necessario usare tecniche *pay-as-you-go*, cioè un approccio più agile in cui si definiscono le procedure di integrazione in modo incrementale e non tutte in una volta come nella business intelligence. Quindi in questo step i dati vengono standardizzati e i conflitti nella struttura e nella semantica dei dati vengono risolti. Nel far ciò è necessario trovare il giusto compromesso tra le strategie di modellazione dei dati.

4. **Analisi**: una volta che il dato è stato raffinato dagli step precedenti, può essere applicato un algoritmo di analisi con il fine di estrarne la conoscenza desiderata dal business. Durante lo sviluppo di questa fase possono essere identificati tre passaggi:

    - Esplorazione: il dataset viene esplorato attraverso interazioni real-time per comprenderne la struttura e la semantica. Questo con lo scopo di identificare possibili algoritmi da applicare per estrarre in modo efficiente nuove conoscenze.
    - Analytics: una volta identificate le conoscenze che si vogliono estrarre e l'algoritmo con cui farlo, si procede con lo sviluppo. Al suo interno possono essere usati algoritmi di business intelligence, data mining, machine learning e deep learning. Preferibilmente l'algoritmo dev'essere scalabile e veloce, così da ridurre il più possibile l'attesa dell'utente e, in caso si sia in cloud, i costi.
    - Delivery: un algoritmo di analisi per essere veramente efficace deve essere supportato da una buona rappresentazione delle conoscenze estratte. Senza di esso, anche se venisse sviluppato un ottimo algoritmo, l'utente non coglierebbe il significato di ciò che è stato estratto.

5. **Interpretazione**: dopo aver consegnato all'utente ciò che è stato estratto dai dati, egli dovrà interpretarlo evitando di giungere a conclusioni affrettate. A tal proposito è fondamentale che conosca il dominio, ciò che è stato fatto con il dato e la sua provenienza. Ovviamente è buona norma verificare sempre i risultati facendo accertamenti con piccoli campioni e confermando le ipotesi precedentemente formulate.

6. **Decisione**: grazie a ciò che è stato interpretato dai risultati delle analisi, i manager aziendali possono decidere le strategie aziendali basandosi su di

essi. Ogni volta che si prendono delle nuove decisioni è bene verificarne gli effetti attraverso i feedback ed eventualmente intervenire per un continuo miglioramento.

3.2 Architettura

Data la particolarità del problema e la natura *data-intensive* delle operazioni sorge la necessità di progettare un'architettura appositamente progettata.

3.2.1 Hardware

Uno dei principali problemi dei big data è avere un'architettura che riesca a gestire un tale volume di dati. Il problema non si limita al trovare supporti d'archiviazione che riescano a immagazzinare tutti quei dati, ma è necessario trovare una soluzione anche per la velocità di lettura e scrittura. Infatti, anche se esistesse un hardisk in grado di contenerli tutti, la velocità delle operazioni di input e output non sarebbe abbastanza per svolgere tutte le analisi in modo reattivo.

La soluzione è scalare le risorse dell'architettura. In questo modo è possibile avere maggiore potenza di calcolo e spazio d'archiviazione, ma soprattutto è possibile distribuire i dati su più supporti d'archiviazione e i compiti a più processori. Ci sono due tipologie di scalabilità ognuna con i suoi vantaggi e svantaggi.

- **Scalabilità verticale**: in questo caso si aggiungono risorse all'interno della macchina in questione. Questo lo si può fare aggiungendo nuovi componenti hardware, come memoria RAM o processori, oppure sostituendo quelli vecchi con dei nuovi più performanti e robusti. Ha come vantaggi rispetto alla scalabilità orizzontale di essere meno complesso, di avere un minore consumo energetico, un minor costo per il raffreddamento e per le licenze ed è necessario meno hardware per la comunicazione. Ma in compenso ha gli svantaggi di essere molto più costoso per quanto riguarda l'hardware, si ha difficoltà a reperirlo e spesso si è obbligati a rimanere legati allo stesso fornitore. Per queste ragioni non viene considerata una soluzione a lungo termine.

- **Scalabilità orizzontale**: consiste nel aggiungere nuove macchine che possano lavorare in parallelo con quelle già presenti all'interno dell'architettura. Nonostante questa soluzione porti maggiore complessità di gestione, in questo contesto è considerata la soluzione migliore sia per i costi che per le performance. Ha come vantaggi che l'hardware è meno costoso, è meglio per la *fault tolerance*, è più semplice espansione e, soprattutto, lo si può fare all'infinito. Come svantaggi invece si hanno maggiori costi per le licen-

ze, l'elettricità e il raffreddamento, una maggiore impronta ambientale ed è necessario più hardware di comunicazione.

Tra le diverse architetture quella considerata più adatta per progetti di big data, e quindi per svolgere elaborazioni su grandi quantità di dati, è il cluster. Si usa il modello *shared-nothing*, cioè ogni unità di lavoro ha la propria memoria e il proprio disco, e per farlo si sfrutta la scalabilità orizzontale. Questo permette al cluster di non avere limiti di estendibilità e di non correre il rischio di rimanere legati allo stesso fornitore dell'hardware. È anche possibile partizionare le unità per svolgere compiti differenti. Le macchine di un cluster sono raccolte in *rack*, anche essi connessi tra loro da un ulteriore livello di rete o da uno switch. La comunicazione tra macchine dello stesso rack sarà ovviamente più veloce rispetto a quella tra macchine di rack differenti. Una delle particolarità dei cluster è che sono formati da *commodity hardware*, cioè da hardware standard facilmente reperibile che è possibile scegliere tra tanti fornitori. Ovviamente questo non vuol dire che possa essere di scarsa fattura, in quanto questo porterebbe all'innalzamento del tasso di fallimenti. Per distribuire la computazione in modo efficiente su un'architettura del genere è necessario seguire un approccio *divide et impera*. Quindi servirà suddividere ciò che si vuole fare in task più piccoli che possano essere eseguiti su macchine differenti in parallelo e che poi riuniscano i propri risultati. Questo processo su un cluster presenta diverse insidie: come assegnare i task, come sincronizzare le diverse macchine, come condividere i risultati dei task, cosa fare se una macchina fallisce, come dividere il carico di lavoro, come fare il debug. La soluzione è stata nascondere il problema separando il cosa fare dal come e di quest'ultimo se ne occupa un framework creato appositamente, di solito Apache Hadoop e i vari servizi su di esso.

3.2.2 On-premise o cloud

Tenere un cluster *on-premise*, cioè fisicamente all'interno della propria azienda, non è così semplice. Questo implica che spetta all'azienda stessa preoccuparsi del settaggio e gestione dell'architettura e per far ciò è necessario dedicare tanto tempo esclusivamente a quello. Quando si parla di settaggio e gestione si comprendono tante operazioni diverse: l'installazione e miglioramento dell'architettura, gestione della connessione, dell'hardware, di eventuali rotture di esso, del consumo energetico, del raffreddamento, delle licenze e di eventuali assicurazioni. Quindi vanno fatte diverse considerazioni prima di prendere la decisione di installare un cluster in casa, sia dal lato tecnologico, come la scelta della configurazione o la gestione del flusso di dati, che da quello di business, come costi, rapporto rischi benefici e quantità di tempo e risorse umane da impiegare.

Per sopperire a questi problemi negli ultimi anni è emerso sempre più il *cloud computing*. In base alla definizione data dal NIST, consiste in un modello che consente l'accesso ubiquitario, conveniente e on-demand ad un pool condiviso di risorse configurabili con il minimo sforzo di gestione. Questo permette alle aziende di richiedere risorse in base alle proprie necessità, sia per quanto riguarda la quantità che il quando, di averne l'accesso da ovunque e di pagare solo ciò che viene usato. Ciò garantisce una grande scalabilità delle risorse, sia verticalmente che orizzontalmente, ed elasticità di servizio, in quanto è possibile scalare le risorse in base alle richieste. Oltretutto un servizio del genere permette alle aziende di focalizzarsi sui task strategici, astraendo dall'architettura e con la possibilità di integrare fra loro servizi diversi.

Però, oltre agli aspetti positivi, il cloud ha anche alcuni aspetti critici. I sistemi di cloud computing vengono infatti criticati principalmente per l'esposizione a questi rischi:

- Sicurezza e privacy degli utenti: memorizzare dati personali o sensibili su cloud, espone l'utente a potenziali problemi di violazione della privacy. Ad esempio il cloud provider, in caso di comportamento scorretto o malevolo, potrebbe accedere ai dati personali per eseguire ricerche di mercato e profilazione degli utenti.

- Problemi internazionali di tipo economico e politico: questo genere di problemi potrebbe sorgere quando dati pubblici sono raccolti e conservati in archivi privati, situati in un paese diverso da quelli degli utenti. Di solito le maggiori garanzie vi sono quando il fornitore del servizio appartiene alla stessa nazione del cliente e quindi le normative sulla privacy e sicurezza sono le medesime. Ad esempio la legislazione americana è molto differente da quella italiana ed europea.

- Continuità del servizio offerto: delegando a un servizio esterno la gestione dell'architettura, in caso vada per qualche motivo fuori servizio, l'utente si trova fortemente limitato nelle proprie azioni. Inoltre, essendo dei servizi condivisi, un eventuale malfunzionamento colpirebbe diversi fruitori contemporaneamente. I fornitori di servizi di cloud computing cercano di ridurre la possibilità di guasti visibili dall'utente finale attraverso l'uso di architetture ridondanti e di personale qualificato, ma ciò non elimina il rischio. Un'altra cosa che bisogna considerare è che, per lavorare su cloud, è fondamentale avere una buona connessione perché con una sua interruzione o rallentamento si corre il rischio di paralizzare le attività.

- Difficoltà di migrazione ad un altro fornitore di servizi cloud: non esistendo uno standard, un eventuale cambio di operatore risulta estremamente complesso.

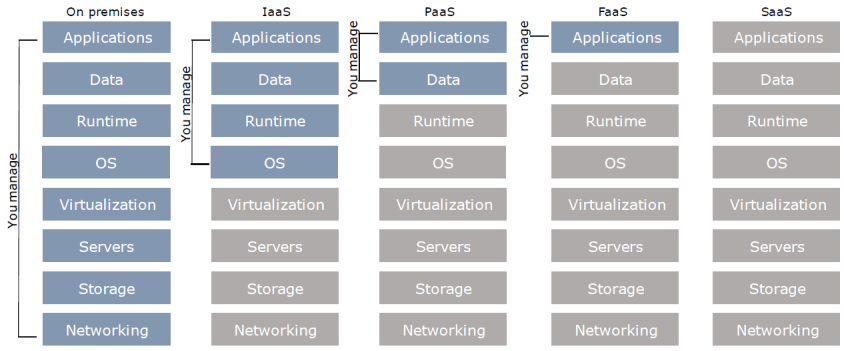

Figura 3.3: Classificazione delle architetture

Per fare in modo che i progetti abbiano successo è importante scegliere l'architettura giusta e i cloud ne offrono varie tipologie suddivisibili in base a ciò che l'azienda vuole gestire personalmente nell'architettura. Come mostrato nella fig. 3.3, le categorie sono queste:

- **Infrastructure as a Service (IaaS)**: dal gestore viene fornita solo l'infrastruttura. L'utente ha la responsabilità di gestire anche il sistema operativo.
- **Platform as a Service (PaaS)**: vengono fornite piattaforme con l'ambiente di sviluppo desiderato, dove l'utente può eseguire il proprio codice evitandosi la gestione delle licenze.
- **Container as a Service (CaaS)**: simile a PaaS, ma si usa un approccio alla virtualizzazione diverso, attraverso l'utilizzo dei container.
- **Function as a Service (FaaS)**: in questo caso l'utente deve solo fornire la funzione, mentre il gestore si occuperà di eseguirla. Grazie a ciò si ha il completo disaccoppiamento della funzione dall'architettura fisica. Questo però comporta una maggiore latenza e difficoltà in caso si decidesse di migrare ad un altro gestore.
- **Software as a Service (SaaS)**: il gestore fornisce direttamente un software fatto e finito eseguito sul proprio cloud. L'utente non ha nessuna responsabilità su di esso, può semplicemente usarlo.

3.2.3 Software

Data la complessità di questi sistemi in letteratura si possono trovare diverse linee guida su come comporre le architetture al livello software. Una di queste è quella proposta dal National Institute of Standards and Technology (NIST) con la collaborazione del mondo accademico e industriale [9].

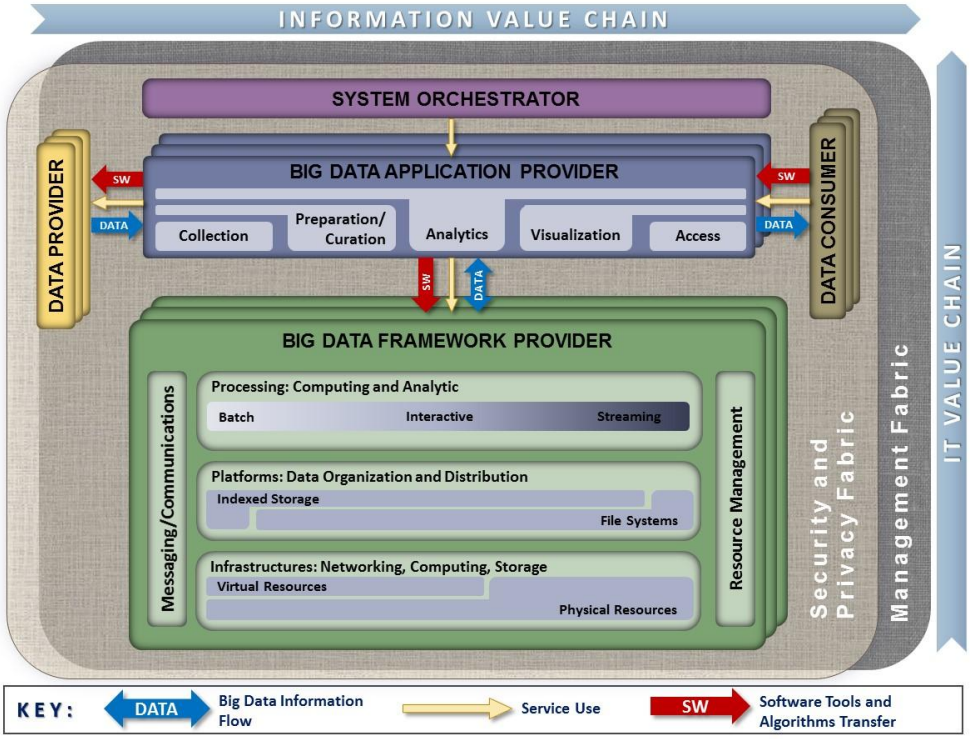

Figura 3.4: Architettura di riferimento per i big data del NIST [9]

Nell'architettura mostrata nella fig. 3.4 si possono individuare, oltre alle problematiche trasversali che interessano tutta l'architettura quali la gestione e la sicurezza e privacy, cinque attori principali:

- **Big Data Framework Provider**: consiste nel livello infrastrutturale di base su cui si appoggiano tutti gli altri livelli. In pratica fornisce risorse o servizi generali che devono essere utilizzati dal Big Data Application Provider nella creazione dell'applicazione specifica. Comprende tre sotto ruoli:
  - Framework infrastrutturale: supporta le funzioni di elaborazione, archiviazione e connessione sottostanti necessarie per implementare l'intero sistema gestendo le risorse.

  - Piattaforma dati: gestisce l'organizzazione e la distribuzione dei dati all'interno del cluster.
  - Framework d'elaborazione: fornisce strumenti per poter eseguire computazioni distribuite.

- **Big Data Application Provider**: il suo ruolo è quello di eseguire una serie specifica di programmi lungo il ciclo di vita dei dati: ingestione, preparazione, analisi, visualizzazione e accesso. Il tutto ovviamente rispettando i requisiti stabiliti dal system orchestrator e i requisiti di sicurezza e privacy. Il Big Data Application Provider è il componente dell'architettura che incapsula la logica di business e la funzionalità che deve essere eseguita dall'architettura.

- **System Orchestrator**: è un componente che si occupa di gestire e integrare lo svolgimento delle varie attività all'interno dell'infrastruttura. Oltre a questo configura e gestisce gli altri componenti dell'architettura per implementare e monitorare uno o più carichi di lavoro. Grazie ad esso l'architettura assegna le risorse fisiche e virtuali in modo elastico.

- **Data Producer**: ha il compito di introdurre all'interno dell'architettura i nuovi dati. Può essere interpretato anche da un essere umano.

- **Data Consumer**: ha il compito di leggere e interpretare i dati accessibili all'interno dell'architettura. Anche questo componente può essere interpretato da un essere umano.

Anche Microsoft ha proposto delle proprie linee guida [11].

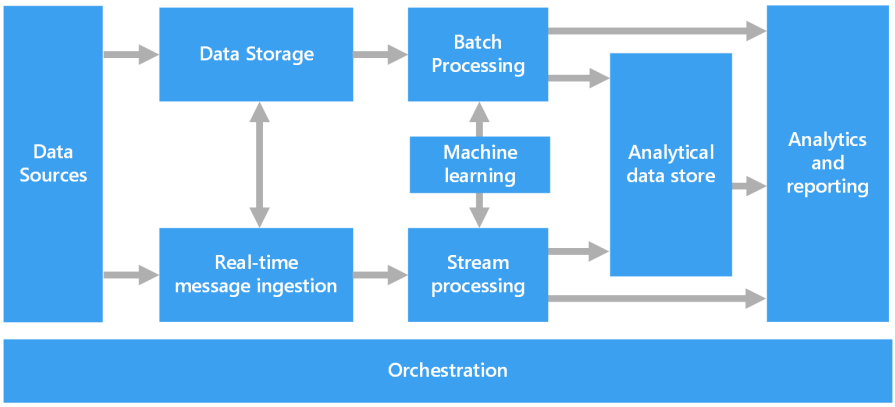

Figura 3.5: Architettura di riferimento per i big data di Microsoft [11]

Come si può vedere nella fig. 3.5, in questo caso ci si focalizza sul ruolo degli strumenti all'interno del flusso di dati e sulla tipologia di componenti coinvolti.

Per Microsoft all'interno di un'architettura big data è possibile riconoscere alcuni o tutti questi componenti:

- **Data Sources**: il punto di partenza di tutte le architetture è costituito da una o più sorgenti dati. Possono essere per esempio:
  - Archivi dati di applicazioni, come i database relazionali.
  - File statici generati dalle applicazioni, come i file di log di server web.
  - Fonti di dati in tempo reale, come i social media e i dispositivi dell'Internet of Thing (IoT).
- **Data Storage**: in questo componente i dati in arrivo dalle sorgenti dati vengono salvati all'interno dell'architettura per poi poter procedere con le operazioni di elaborazione in batch. Il salvataggio dei dati avviene in un archivio di file distribuito che può contenere volumi elevati di file in vari formati e spesso viene chiamato *data lake*.
- **Batch processing**: in questo componente avviene l'elaborazione in batch dei dati archiviati all'interno del data lake. A causa delle dimensioni considerevoli dei dataset, è necessario usare l'elaborazione in batch con il fine di filtrare, aggregare e preparare i dati per l'analisi. In genere questi processi prevedono la lettura dei file di origine, la relativa elaborazione e la scrittura dell'output in nuovi file.
- **Real-time message ingestion**: se il progetto include una fonte dati in tempo reale, l'architettura deve includere un componente che permetta di acquisire e archiviare i messaggi in tempo reale per la successiva elaborazione. Potrebbe trattarsi di un archivio dati semplice in cui i messaggi in ingresso vengono rilasciati in una cartella. Molti progetti richiedono a questo componente che abbia altre particolarità, ad esempio che funga da buffer per i messaggi, che sia scalabile orizzontalmente, che si assicuri del recapito dei messaggi o altri tipi di semantica di consegna dei messaggi (*message delivery semantics*).
- **Stream processing**: questo componente ha il compito di elaborare il flusso di messaggi acquisito in tempo reale, per prepararli all'operazione successiva di analisi. I dati del flusso elaborati vengono dati in output attraverso dei connettori.
- **Analytical data store**: in molti progetti è presente anche questo componente che ha il compito di immagazzinare i dati elaborati dalla piattaforma

in modo da permettere poi di svolgere analisi e di interrogarli. L'analytical data store può essere un data warehouse relazionale come nella maggior parte dei progetti di business intelligence tradizionali. In alternativa, i dati possono essere salvati su una tecnologia NoSQL a bassa latenza.

- **Analytics and reporting**: in fondo alla pipeline c'è il componente che permette il raggiungimento dell'obiettivo del progetto, cioè fornire informazioni dettagliate sui dati sfruttando strumenti di analisi e report. Per consentire agli utenti di analizzare i dati, l'architettura può includere un livello di modellazione dei dati, ad esempio un cubo OLAP multidimensionale o un modello di dati tabulari. Potrebbe inoltre supportare la business intelligence in modalità self-service, usando le tecnologie di modellazione e visualizzazione in Microsoft Power BI o Microsoft Excel. L'analytics and reporting possono anche assumere la forma di esplorazione interattiva dei dati da parte di data scientist o analisti di dati.

- **Orchestration**: questo componente ha il ruolo fondamentale di gestire e automatizzare tutto il flusso di lavoro, dai dati di origine al report o dashboard finale.

3.3 Streaming

Nel progetto Social Network Analysis i dati da Twitter vengono elaborati e caricati su un database. Tale compito è possibile farlo in batch, ma questo causerebbe ritardi nell'aggiornamento dei dati e, di conseguenza, della dashboard. Per questo per risolvere problemi in cui la velocità di aggiornamento è fondamentale sono nate le architetture streaming, cioè che eseguono l'elaborazione su flussi di dati. L'elaborazione in questo caso viene fatta su dati in movimento, un flusso che va gestito in tempo reale o quasi. Questo implica che la computazione deve essere più semplice possibile in modo che sia continua e non rimanga indietro rispetto all'arrivo dei dati. Siccome il flusso di dati viene suddiviso in mini-batch per essere elaborato, la quantità di dati tenuta in considerazione ad ogni computazione è molto piccola. Per questo spesso gli algoritmi usati per le elaborazioni in batch non possono essere usati per quelle in streaming perché per funzionare hanno bisogno di tenere in considerazione tutti i dati.

L'elaborazione in streaming può essere utile quando si vuole:

- Bassa latenza tra l'arrivo dei dati e la visualizzazione dei risultati. Infatti questo tipo di elaborazione permette di avere risultati quasi in tempo reale, man mano che arrivano nuovi dati.

- Bilanciamento del carico di lavoro perché lo si distribuisce nel tempo. In questo modo si ha un minor uso di risorse, ma è necessario che esse siano sempre allocate per l'elaborazione del flusso.

Come si può immaginare, ci sono diversi casi d'uso, come il monitoraggio dei social media, di sensori industriali oppure di oggetti IoT. In base al caso d'uso si ha la necessità di diversi tipi di "tempo reale":

- **Hard Real Time**: serve nei contesti in cui il ritardo non è concesso e si parla di una latenza piccolissima, sull'ordine dei microsecondi arrivando ad un massimo di millisecondi.

- **Firm Real Time**: in questo caso è permessa una latenza un po' più elevata, tra i millisecondi e i secondi. Inoltre l'arrivo in ritardo di dati viene tollerato, ma questi vengono buttati senza essere analizzati.

- **Soft Real Time**: qui il ritardo è totalmente concesso e la latenza sale ancora arrivando anche ai minuti.

In un sistema streaming il motore di elaborazione deve essere progettato per essere in grado di elaborare dataset potenzialmente infiniti e quindi per rimanere in esecuzione all'infinito. È possibile usare anche motori per l'elaborazione in batch, dando in input dei mini-batch, ma anche viceversa.

I dati del flusso possono essere modellati in tre modi diversi:

- **Modello serie temporale**: ogni dato rappresenta il nuovo stato nel tempo.

- **Modello registro di cassa**: in questo caso il singolo dato rappresenta una variazione dallo stato precedente, ma andando solo in positivo.

- **Modello a tornello**: anche in questo modello il dato rappresenta una variazione, ma in questo caso può essere anche negativo.

Se si estraesse la parte streaming dalle pipeline precedentemente esposte si avrebbe la pipeline mostrata nella fig. 3.6. Si possono riconoscere tre livelli principali:

- **Livello di collezione**: è la parte dell'architettura che si occupa dell'ingestione dei dati prodotti dal produttore. È formato da server edge e si possono usare diversi pattern di interazione, ma di solito quello più usato è *publish/subsribe*. Deve essere programmata una strategia per gestire la *fault tolerance*, cioè il caso in cui uno dei nodi fallisca.

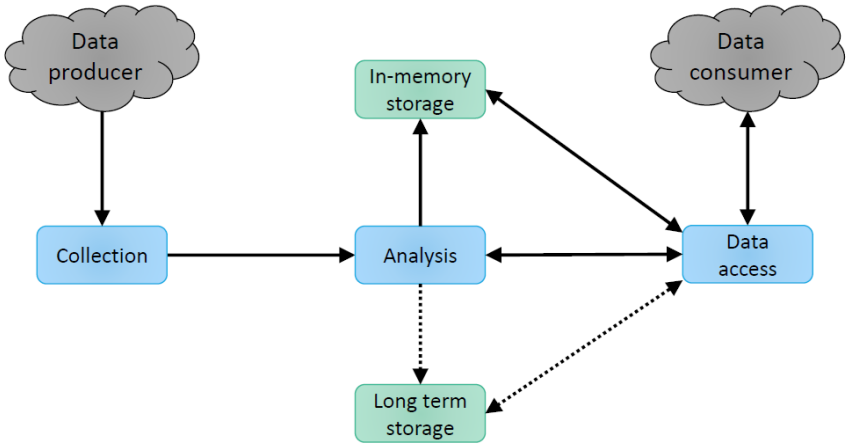

Figura 3.6: Pipeline di un'architettura streaming di big data

- **Livello di analisi**: è il cuore dell'architettura ed è dove si applicano gli algoritmi. In questo livello viene lanciata una query continua sui dati che arrivano con un modello *push*, cioè senza che vengano richiesti. Spesso necessita di uno stato, cioè di un risultato intermedio, che però non deve occupare troppa memoria. Infatti quest'ultima rappresenta uno dei due limiti che hanno questo tipo di query rispetto a quelle tradizionali. L'altro limite è il tempo: considerando che tutti i dati vanno elaborati in tempo reale se l'algoritmo non riesce a starci dietro avviene il *load shedding*, cioè i dati non processati vengono scartati. Inoltre è possibile che avvenga il *concept drift*, cioè in caso venga inserito un qualche tipo di conoscenza che può cambiare nel tempo, c'è il rischio che l'algoritmo perda di efficacia.

- **Livello di accesso**: ha il ruolo di esporre i dati al consumatore che può essere un umano, ma anche un altro strumento o altre pipeline.

I diversi livelli possono essere connessi direttamente tra loro, ma è anche possibile inserire un **livello di comunicazione** svolto da un servizio di accodamento di messaggi con il compito di gestire lo smistamento dei dati tra i diversi livelli. Questo livello può essere utile per disaccoppiare i diversi livelli e per garantire un certo livello di sicurezza della comunicazione, soprattutto nei casi in cui i dati arrivano più rapidamente dell'elaborazione grazie alla gestione delle code. Un altro vantaggio di questo livello è la possibile gestione del *funneling*, cioè di quei casi in cui il numero di flussi è diverso rispetto a quello dei consumatori. Per la consegna di messaggi ci sono tre tipologie diverse di semantica di consegna:

- **Exactly once**: il sistema garantisce che il dato venga inviato una volta e mai perso.

- **At most once**: in questo caso il dato viene inviato una volta sola, ma potrebbe essere perso.
- **At least once**: il sistema garantisce che nessun dato venga perso, ma qualcuno di essi potrebbe essere inviato più volte.

Spesso la semantica di consegna è importante che venga considerata anche nel livello di analisi.

Dopo l'elaborazione da parte del livello di analisi i dati possono essere scartati, inseriti in un'altra pipeline oppure salvati. Ed è in quest'ultimo caso che entra in gioco il **livello di storage**, un altro componente dell'architettura che fa da supporto. I dati possono essere salvati in modo temporaneo su memoria oppure a lungo termine su disco. Il salvataggio in memoria può essere utile per trattenere dei risultati temporanei oppure per tenersi da parte dei dati per altre analisi nel breve periodo. Invece il salvataggio su disco serve a conservare dei dati o risultati per elaborazioni future. Però quest'ultimo è molto dispendioso per cui è consigliabile disaccoppiare dal livello di analisi con un servizio di accodamento di messaggi.

3.4 Database NoSQL

All'interno del progetto Social Network Analysis per raccogliere i dati è stato necessario usare un database. Per questo motivo si è dovuto fare una riflessione sul tipo di database che potesse venire più incontro alle esigenze del progetto.

I database relazionali fino pochi anni fa hanno dominato la scena dei database grazie a diversi punti di forza. La proprietà ACID racchiude in sé la maggior parte di essi: l'atomicità delle transazioni, la consistenza grazie ai vincoli di integrità, l'isolamento delle transazioni e la durabilità. Oltre a questo c'è anche il fatto che il loro modello e il linguaggio di query (SQL) sono standard e molto conosciuti, essendo stati usati per più di quarant'anni. Tutte queste caratteristiche lo rendono molto robusto, ma negli ultimi anni con l'arrivo di nuove esigenze come la variabilità dei dati sono emersi anche alcuni punti deboli:

- **Impedence Mismatch**: conflitto tra la modellazione dei dati su applicazioni rispetto a quella su database.
- **Scalabilità orizzontale**: non è possibile scalare orizzontalmente un database relazionale se non manualmente.
- **Rigidità dello schema**: lo schema definito alla creazione della tabella è difficile e costoso cambiarlo.
- **Consistenza o latenza**: il volere assicurare la consistenza del database va ad incidere negativamente sulla latenza.

Tutti questi punti rendono evidente che nella maggior parte dei casi i database relazionali non sono adatti per i progetti di big data.

Per venire incontro a queste esigenze sono nati i database NoSQL. Inizialmente, nel 1998 il termine era usato per identificare dei *Database Management System* (DBMS) relazionali che usavano un linguaggio differente dal SQL per le query. Adesso invece viene usato per identificare tutti quei DBMS che non sono relazionali. Comunemente questi hanno una modellazione dati differente, non hanno gli schemi rigidi dei database relazionali, evitano le join e soprattutto possono essere distribuiti su un architettura shared-nothing.

Mentre nei database relazionali i dati vengono modellati attraverso righe in tabelle, in quelli NoSQL si parla di collezioni di dati. Queste collezioni possono assumere modelli differenti:

- **Modello chiave-valore**: i dati vengono modellati come coppie chiave-valore ordinate in base alla chiave, come se fosse un dizionario. Quindi la chiave corrisponde all'identificativo, mentre il valore al contenuto del dato. Quest'ultimo può essere qualsiasi cosa, da una semplice stringa di testo a un file vero e proprio. Mentre per i database relazionali è per transazione, il livello di atomicità in questo caso è per coppia chiave-valore, quindi si garantisce che le operazioni eseguite sulla stessa chiave siano atomiche. In questo modello il valore è come se fosse una scatola nera, per cui le interrogazioni assumono una certa rigidità. Infatti sono possibili solo tre tipologie di operazioni:
  - **Put** per inserire una nuova coppia;
  - **Get** per ottenere il valore di una coppia data la chiave;
  - **Delete** per eliminare una coppia con una determinata chiave.
- **Modello documentale**: la collezione in questo caso contiene documenti, cioè un insieme di campi tra cui c'è obbligatoriamente anche la chiave univoca del documento. Di solito sono file JSON, o comunque file con struttura gerarchica. In questo modello il livello di atomicità è per documento. Il contenuto a differenza del modello chiave-valore è visibile per cui è possibile fare interrogazioni più elaborate.
- **Modello wide column**: è simile al modello relazionale, ma al posto delle tabelle ci sono le *column family*, cioè liste di righe con all'interno i dati in formato coppia chiave-valore, come si può vedere nella fig. 3.7. Questo permette al modello di non avere schemi rigidi come i database relazionali e di salvare per ogni riga solo i valori presenti, rendendo così il modello molto efficiente per rappresentare le matrici sparse. Il livello di atomicità in questo modello è per riga. Per quanto riguarda il livello di espressività delle

interrogazioni è una via di mezzo tra quella limitata del modello chiave-valore e quella del modello documentale. Questo perché di solito non sopporta le operazioni di join e, a seconda dell'implementazione, non sempre è presente la possibilità di filtro in base al contenuto.

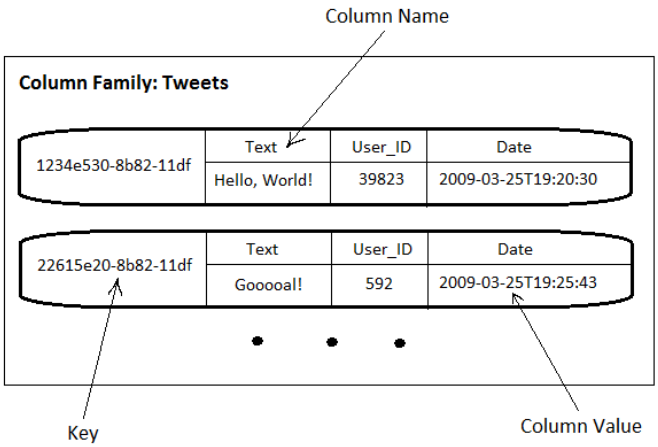

Figura 3.7: Esempio di column family

- **Modello a grafo**: è un modello che dà priorità alla rappresentazione delle relazioni e, come si può intendere dal nome, lo fa inserendo i dati all'interno di un grafo. Quindi i nodi del grafo corrispondono ai record o documenti, che possono essere potenzialmente ognuno differente, mentre gli archi sono le relazioni che li legano fra loro. Gli archi è come se fossero dei puntatori, per cui, grazie ad essi, la navigazione del grafo è molto rapida. In questo caso il livello di atomicità è per transazione, come per i database relazionali. Il linguaggio di interrogazione è completamente diverso: qui si cercano all'interno del database le occorrenze di regole e pattern. La presenza di puntatori rappresenta una limitazione per quanto riguarda la scalabilità. Questo perché le interrogazioni su un grafo distribuito sono complesse a causa dei salti continui tra i documenti che potrebbero essere su macchine diverse. Perciò è necessario limitare la profondità delle interrogazioni. Inoltre, al livello implementativo, prima di saltare ad un'altra macchina, vengono raccolte in batch tutte le query indirizzate ad essa. Questo modello è molto differente dai precedenti e viene usato per contesti particolari, in cui si dà priorità alle relazioni piuttosto che al contenuto dei documenti.

Ad un modello a grafo ci si arriva con una modellazione data-driven, quindi guidata dalla priorità di rappresentare le relazioni. Per quanto riguarda invece gli altri modelli, si segue una modellazione query-driven, cioè guidata dalle modalità

d'uso dei dati da parte delle applicazioni. Infatti questi modelli sono composti da blocchi atomici all'interno dei quali si incapsulano tutte le informazioni necessarie all'applicazione e per questo vengono chiamati anche *aggregate oriented*. Grazie a ciò si evitano il più possibile join tra i dataset, migliorando la disponibilità dei dati e rendendo così più semplice la distribuzione. Però questo comporta anche degli aspetti negativi: è possibile ottimizzare solo determinate interrogazioni e si ha la denormalizzazione dei dati e introduzione di potenziali inconsistenze.

Capitolo 4

Architettura scelta

Come spiegato nel capitolo 2, il reparto marketing di un'azienda, per prendere le giuste decisioni, deve essere consapevole di ciò che lo circonda. A tal proposito, i social media possono offrire una proiezione in tempo reale della realtà. Un occhio umano non è in grado di visionare per intero le reti che si districano nei social media, ne vedrebbe solo un frammento. Ed è qui che entra in gioco il progetto Social Network Analysis (SNA).

SNA vuole essere uno strumento attraverso il quale gli addetti al marketing possono avere una visione completa in tempo reale della rete di interazioni su un certo argomento tra i diversi utenti di Twitter. L'obiettivo del progetto è stato quello di creare una dashboard interattiva che mostrasse i *topic* del momento su cui c'è maggiore discussione e gli influencer di maggiore spicco, cioè quelli che hanno maggiore impatto sull'opinione pubblica.

4.1 Architettura funzionale

Per realizzare SNA prima di tutto è stata progettata, a partire dall'obiettivo sopracitato, l'architettura funzionale. Essa può essere divisa in due componenti:

- **Loader**: architettura streaming che raccoglie i tweet da Twitter in tempo reale, estrae le informazioni di interesse per il progetto e le carica su un database a grafo.
- **Dashboard**: piattaforma che analizza le informazioni raccolte nel database a grafo e mostra i risultati all'utente in una schermata interattiva.

4.1.1 Loader

Quella mostrata nella fig. 4.1 è l'intera pipeline del Loader. Percorrendo la pipeline seguita dal flusso di dati si va in quest'ordine:

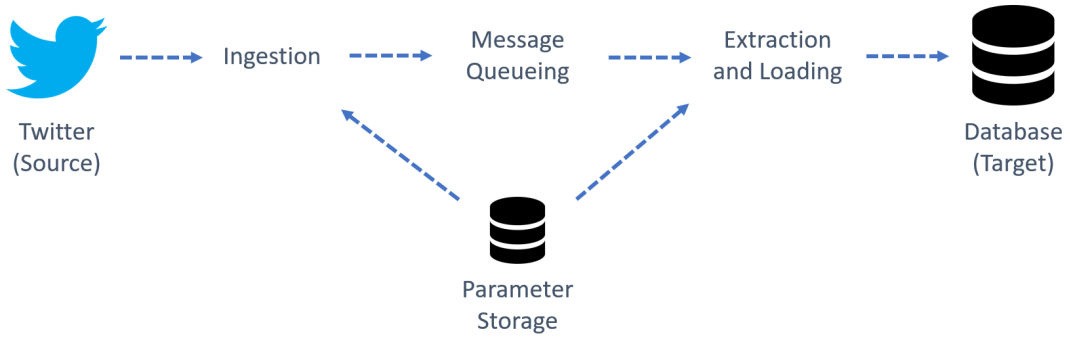

Figura 4.1: Architettura funzionale del Loader

1. **Twitter**: è la sorgente dell'architettura. È uno dei social media più famosi al mondo e fa parte della categoria microblog. Consente di pubblicare brevi post composti da un massimo di 280 caratteri, chiamati *tweet*. Gli utenti possono "seguire" altri utenti, diventando così *follower*, per ricevere aggiornamenti sui loro tweet. I tweet possono essere di tipi diversi:

   - **Tweet comune**: è un messaggio pubblicato su Twitter che contiene una GIF, un video, delle foto e/o del testo;
   - **Menzione**: è un tweet in cui si cita l'username di un altro utente con davanti la chiocciola (@);
   - **Risposta**: è una risposta al tweet di un altro utente;
   - **Retweet**: è la ripubblicazione di un tweet proprio o di un altro utente, così da condividerlo con i propri follower;
   - **Quoted tweet**: è un retweet con l'aggiunta di un proprio commento.

   È pratica comune catalogare i propri post mettendo in evidenza parole chiave e argomenti con l'uso degli *hashtag* per poter raggiungere facilmente le persone interessate a tale argomento. Per fare un hashtag è necessario inserire il simbolo # prima della parola chiave o argomento. Registrandosi come sviluppatore si ha la possibilità di accedere al social in modo programmatico attraverso le API che Twitter stesso rende disponibili. In questo modo si può accedere agli elementi principali di Twitter come: tweet, messaggi diretti, bacheche, elenchi, utenti e altro ancora. È anche possibile creare un canale attraverso il quale passa in tempo reale il flusso di tweet che soddisfano determinati criteri di ricerca impostati in creazione. In questo progetto è stata sfruttata proprio questa funzionalità chiamata *Filtered Stream*.

2. **Ingestione dati**: è il primo passo che fanno i dati di Twitter all'interno dell'architettura proposta. È necessario l'uso di un servizio che rimanga in connessione con il flusso da Twitter ininterrottamente 24 ore su 24. Esso ha il compito in avvio di creare la connessione con Twitter sfruttando la funzionalità delle API Twitter Filtered Stream, dando in input i parametri di ricerca salvati su un altro servizio, *Parameter Storage* nella fig. 4.1. All'interno del Parameter Storage c'è un unico file json in cui sono definiti i diversi topic d'interesse, accompagnati dalle rispettive parole chiave e dalla lingua in cui devono essere scritti i tweet. Come parametri per l'uso della Filtered Stream devono essere usate tutte le parole chiave contemporaneamente senza distinzioni di topic. Una volta avviato il flusso, questo componente si occupa di accogliere ed inviare i tweet al servizio successivo nella pipeline.

3. **Accodamento dei messaggi**: dal servizio che si occupa dell'ingestione, i dati vengono inviati ai servizi che si occuperanno dei passi successivi. Per disaccoppiare i servizi e gestire la coda dei dati ci si affida ad un apposito servizio di accodamento dei messaggi. Grazie ad esso è possibile partizionare in più code i tweet, permettendo in questo modo di parallelizzare il flusso, e trattenerli in memoria, così da gestire eventuali ritardi o fallimenti del servizio successivo nella pipeline.

4. **Estrazione e caricamento dati**: dopo il servizio di accodamento messaggi i dati vengono convogliati a quello che si occupa di estrapolare da essi le informazioni necessarie, di formattarle in modo da poter essere utilizzabili per l'analisi ed infine di caricarle su un database, che nel nostro caso è a grafo.

5. **Database a grafo**: corrisponde all'output della pipeline. In questo punto è necessario un DBMS di database NoSQL a grafo che accolga i dati estratti dai tweet. Le tipologie di nodi e archi all'interno sono quelli mostrati nella fig. 4.2. I nodi possono essere di diversi tipi:

   - **User**: rappresenta gli utenti.
   - **Tweet** rappresenta i vari tweet postati dagli utenti.
   - **Hashtag**: rappresenta gli hashtag contenuti all'interno dei tweet pubblicati dagli utenti.

   Anche gli archi possono appartenere a diverse tipologie:

   - **Post**: mette in relazione gli utenti con i tweet che hanno pubblicato.
   - **Amplifies**: mette in relazione gli utenti che hanno condiviso un tweet con il suo creatore.

- **Talk_about**: mette in relazione gli utenti con gli hashtag che hanno usato nei loro tweet.

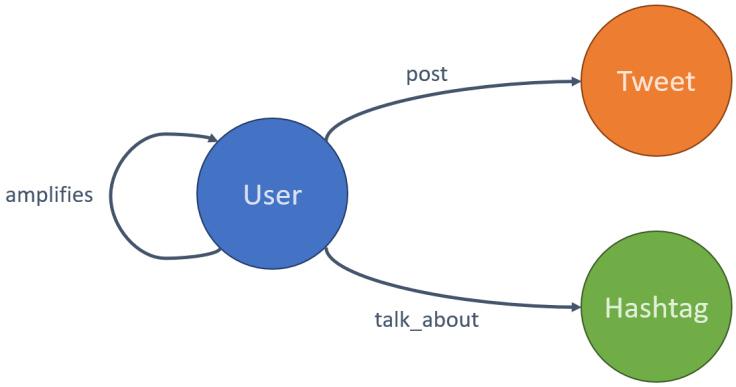

Figura 4.2: Grafo d'esempio con tutti i tipi di nodi e archi

4.1.2 Dashboard

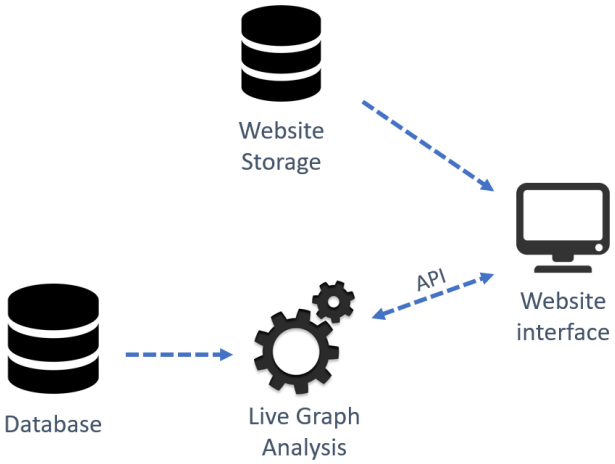

Figura 4.3: Architettura funzionale della Dashboard

La Dashboard è la parte del progetto che si occupa di analizzare i dati raccolti e di gestire l'interfaccia utente. Come si può vedere nella fig. 4.3, all'interno della sua architettura, oltre al database da cui si prendono i dati che il Loader ha caricato, sono riconoscibili altri due componenti principali:

- **Website Storage**: è il componente all'interno del quale è contenuto il sito web. L'utente, accedendo al sito, potrà interfacciarsi con i dati raccolti e i risultati delle analisi sul grafo.

- **Live Graph Analysis**: è il componente che si occupa di interfacciarsi con il database, analizzare i dati estratti da esso e restituire i risultati in output. Siccome il database è in continuo aggiornamento le analisi sul grafo vengono svolte ogni volta che si fa una nuova richiesta al servizio. Per lo svolgimento delle analisi si procede in questo modo:

  1. Si estrae la porzione di grafo che si vuole analizzare. Ad esempio è possibile considerare il grafo derivante da dati di un certo intervallo di tempo, oppure un grafo in cui sono presenti solo gli utenti.
  2. Si applica l'algoritmo PageRank, algoritmo inventato da Google per la indicizzazione dei siti web, ma applicabile a qualsiasi grafo [25]. Consiste in un'analisi che assegna un peso numerico ad ogni elemento di un insieme di nodi connessi tra loro con lo scopo di quantificare l'importanza di ognuno all'interno dell'insieme stesso.
  3. Si applica l'algoritmo *Clauset-Newman-Moore greedy modularity maximization* per ottenere i raggruppamenti maggiormente coesi all'interno del grafo, le comunità [10]. A partire da ogni nodo si creano delle comunità e iterativamente si uniscono quelle che aumentano maggiormente la modularità fino al raggiungimento del numero di comunità definito.
  4. Si calcolano altre statistiche da mostrare all'utente come ad esempio il numero di nodi e di archi.

Per rispondere alle esigenze del business sono state pensate tre tipologie di interfacce da cui poter visualizzare i risultati delle analisi. Per tutte e tre le interfacce vengono usati sempre gli stessi algoritmi di analisi esposti qui sopra, ciò che cambia è semplicemente la tipologia di dati presi in considerazione all'interno del grafo. Le interfacce implementate, visibili nella fig. 4.4, sono:

- **Influencer e comunità**: nel grafo da analizzare vengono considerati solo gli utenti e le relazioni amplifies, così da individuare gli influencer, cioè gli utenti che vengono ripubblicati maggiormente. Inoltre vengono identificate le comunità con l'algoritmo Clauset-Newman-Moore greedy modularity maximization.

- **Topics**: nel grafo, oltre agli utenti, vengono considerati anche gli hashtag così da poter trovare gli argomenti maggiormente discussi e gli hashtag maggiormente usati.

- **EgoNets**: in questo ci si focalizza su un singolo utente e si analizza il grafo formato da lui e gli utenti legati direttamente a lui.

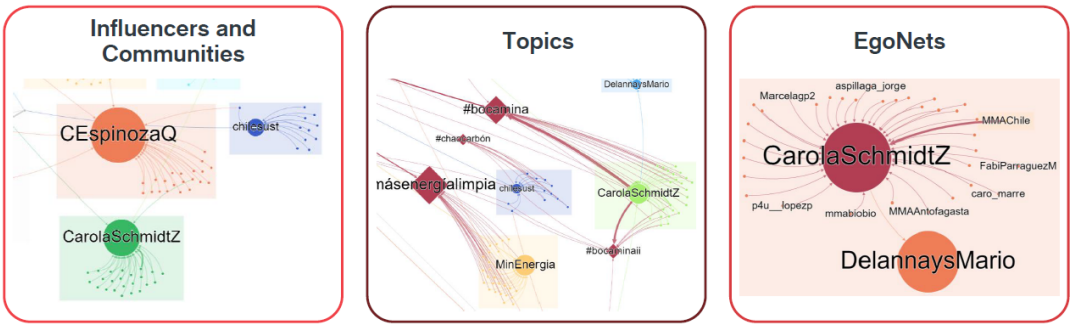

Figura 4.4: Tipologie di grafici visualizzabili

Per rendere più chiara l'architettura della Dashboard analizziamo i passaggi che vengono eseguiti:

1. L'utente accede al sito web e scarica l'interfaccia grafica, ma ancora senza i dati al suo interno.

2. Viene richiesto all'utente di fare il login per accedere. Senza di esso l'utente, anche se ha scaricato l'interfaccia, non ha i permessi per usare il Live Graph Analysis e quindi nemmeno per accedere ai dati del database.

3. Dopo aver fatto l'accesso l'utente può scegliere tra le diverse tipologie di interfacce, selezionare l'intervallo di tempo desiderato e inviare la richiesta al Live Graph Analysis attraverso l'interfaccia del sito web.

4. Il componente di analisi raccoglie dal database i dati necessari per rispondere alla richiesta, li analizza e restituisce i dati e i risultati delle analisi.

5. Il sito web ricevendo la risposta dal Live Graph Analysis aggiunge i dati all'interfaccia attraverso la quale l'utente può esplorare il grafo e i risultati dell'analisi restituiti.

4.2 Architettura tecnologica

Per quanto riguarda il livello tecnologico è stato scelto di usare per tutti i componenti i servizi in cloud di Microsoft, Azure.

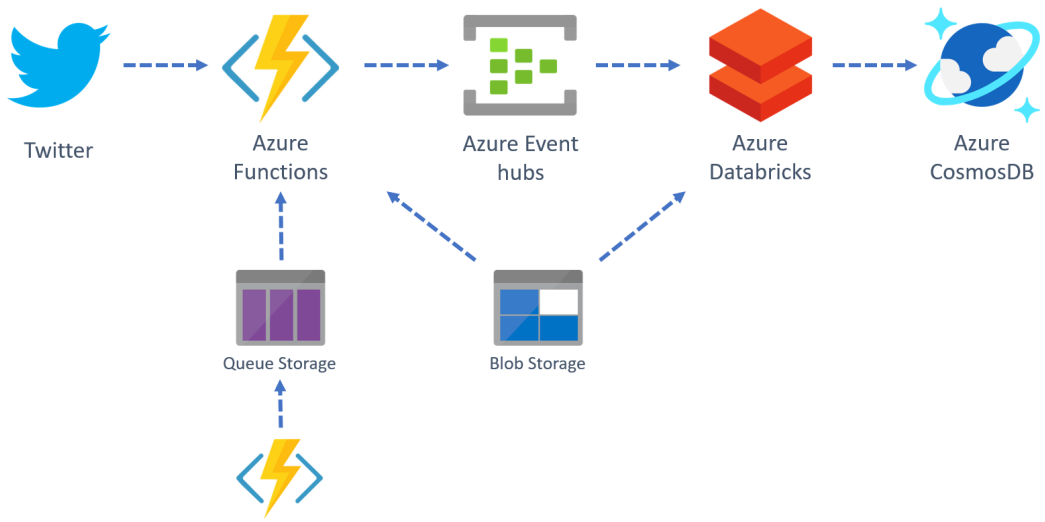

Figura 4.5: Architettura tecnologica del Loader

4.2.1 Loader

Come si può vedere nella fig. 4.5, ad ogni componente dell'architettura funzionale è stato assegnato un servizio di Azure adatto a svolgere la propria funzione.

Azure Functions

Azure Functions è una soluzione di tipo Function as a Service, quindi serverless, che consente di scrivere meno codice, gestire un'infrastruttura meno complessa e risparmiare sui costi. Essendo FaaS, come spiegato nella sezione 3.2.2, non è più necessario preoccuparsi della gestione dell'infrastruttura e delle risorse, in quanto vengono fornite dal cloud. In questo modo ci si può dedicare alla sola definizione delle funzioni che poi il cloud si occuperà di eseguire in risposta a specifici eventi. Questi eventi possono essere ad esempio: richieste tramite apposite API, trigger temporali, modifiche ad un database e caricamenti di nuovi file nello storage.

Un altro vantaggio di questo servizio è che quando le richieste aumentano, queste vengono soddisfatte con il numero necessario di risorse e istanze in esecuzione, ma solo finché servono. Infatti con la diminuzione delle richieste, le eventuali risorse e istanze aggiuntive vengono rimosse automaticamente.

Per l'implementazione le funzioni possono essere scritte in diversi linguaggi: C#, Java, JavaScript, PowerShell o Python. Si può scegliere tra tre piani in base alle esigenze: con il piano a consumo si paga solo per le funzioni in esecuzione, mentre con il piano premium e quello di servizio app si paga giornalmente una

quota in quanto necessitano l'allocazione fissa di una macchina per azzerare i tempi di avvio e garantire altre caratteristiche.

Il servizio Azure Functions è stato usato per gestire l'ingestione dei dati. Siccome è necessario che la funzione rimanga sempre in esecuzione, si è dovuto usare il piano di servizio app in quanto gli altri piani hanno un timeout massimo.

Azure Storage

Azure Storage è il servizio di archiviazione dati in cloud di Azure per gli scenari moderni di archiviazione dei dati. I servizi di base includono al proprio interno diverse tipologie di archiviazione:

- **Blob**: archivio di oggetti scalabile in modo massiccio per testo e dati binari.
- **Files**: servizio di condivisione file per architetture in cloud o on-premise.
- **Queues**: archivio a coda per la messaggistica affidabile tra i componenti dell'applicazione.
- **Tables**: archivio NoSQL per l'archiviazione senza schema dei dati strutturati.
- **Disks**: volumi di archiviazione a livello di blocco per le macchine virtuali di Azure.

Ogni servizio è accessibile tramite un account di archiviazione. Microsoft fornisce librerie client per la gestione di questi servizi in diverse lingue, tra cui .NET, Java, Node.js, Python, PHP, Ruby, Go e altri, oltre alle API REST.

Nel Loader di SNA sono stati usati due dei servizi sopracitati. È stata necessaria l'archiviazione di blob per il salvataggio dei parametri di ricerca su Twitter, usati sia nell'ingestione che nell'elaborazione dei dati. L'altro sevizio usato è l'archiviazione di Queues. È stato necessario per comunicare dei comandi dall'esterno alla funzione che gestisce l'ingestione dei dati, come ad esempio il comando di chiudere la comunicazione con Twitter e terminare. Per aggiungere questi messaggi nella coda di Queues è stata creata un'altra funzione.

Azure Event Hubs

Event Hubs di Azure è una piattaforma che offre un servizio di accodamento di messaggi per architetture streaming di Big Data. Come spiegato nella sezione 3.3, essendo un servizio di accodamento messaggi, si posiziona tra i diversi livelli di un'architettura streaming per smistare i dati. In questo modo si ha il vantaggio di disaccoppiare i produttori di messaggi dai consumatori e di avere un certo livello di

sicurezza nella comunicazione. Può essere utile in svariati scenari: nel rilevamento di anomalie, nelle pipeline di analisi, nell'aggiornamento di dashboard in tempo reale, nell'elaborazione delle transazioni e così via.

Event Hubs è un sistema Platform-as-a-Service (PaaS) completamente gestito da Azure e configurabile con poco sforzo permettendo così all'azienda di concentrarsi sulla soluzione. Usa un modello partizionato, che consente a più applicazioni di elaborare il flusso contemporaneamente e di controllare la velocità di elaborazione in modo indipendente. Un'altra importante caratteristica è la scalabilità: Event Hubs può aumentare la velocità effettiva o scalare il numero di unità di elaborazione in base alle esigenze di utilizzo.

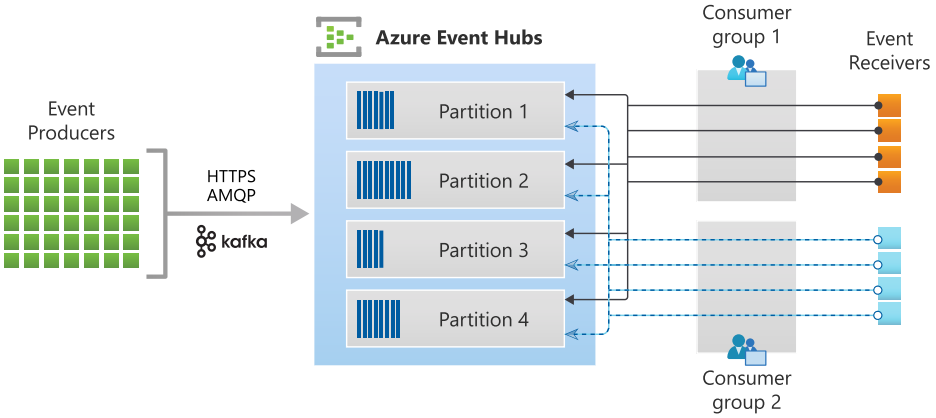

Figura 4.6: Rappresentazione grafica di Azure Event Hubs

Come si può vedere nella fig. 4.6, nella piattaforma di Event Hubs si possono individuare diversi componenti:

- Produttori di eventi: qualsiasi entità che invia dati ad un Event Hub. I produttori possono pubblicare eventi usando HTTPS, AMQP 1.0 o Apache Kafka.

- Partizioni: ogni consumatore legge solo un sottoinsieme specifico del flusso di messaggi e questo viene chiamato partizione.

- Gruppi di consumatori: consentono di utilizzare le applicazioni in modo che ognuna abbia una visualizzazione separata del flusso attraverso la memorizzazione di stato, posizione o offset. Ognuno legge il flusso in modo indipendente, con il proprio ritmo e i propri offset.

- Unità di elaborazione: unità di capacità pre-acquistate che controllano la capacità di velocità effettiva degli Event Hubs.

- Consumatori di eventi: qualsiasi entità che legge i dati da un Event Hubs.

Ovviamente nella nostra architettura Event Hubs ricopre il ruolo di gestione dell'accodamento di messaggi.

Azure Databricks

Azure Databricks è una piattaforma di analisi dei dati ottimizzata per il servizio cloud Azure. É un servizio PaaS e al suo interno ci sono tre ambienti per lo sviluppo di applicazioni a elevato utilizzo di dati con la possibilità di eseguirle su un cluster instanziato on-demand direttamente dalla piattaforma:

- **Databricks SQL**: offre una piattaforma di facile utilizzo per gli analisti che vogliono eseguire query SQL sul proprio data lake, creare più tipi di visualizzazione per esplorare i risultati delle query da prospettive diverse e creare e condividere dashboard.

- **Databricks Data Science Engineering**: offre un'area di lavoro interattiva che consente la collaborazione tra data engineer, data scientist e tecnici di machine learning. Nel caso di una pipeline di Big Data, i dati (non elaborati o strutturati) vengono inseriti in Azure tramite il servizio Azure Data Factory in batch o trasmessi quasi in tempo reale con Apache Kafka, Event Hubs o IoT Hub. In questo caso lo si usa per svolgere la parte di analisi della pipeline, andando a leggere dati da più origini e trasformandoli in informazioni tramite Spark.

- **Databricks Machine Learning**: è un ambiente destinato esclusivamente per il machine learning che incorpora servizi per il rilevamento degli esperimenti, il training dei modelli, lo sviluppo e la gestione delle feature e modelli.

All'interno del progetto SNA è stato usato per realizzare il componente che svolge l'elaborazione dei dati in arrivo da Event Hubs e il caricamento dei risultati sul database a grafo.

Azure Cosmos DB

Azure Cosmos DB è un servizio di database NoSQL sul cloud di Microsoft. É completamente gestito da Azure quindi permette agli utenti di non preoccuparsi dell'amministrazione del database, in quanto la gestione, gli aggiornamenti e l'applicazione di patch vengono eseguiti in modo automatico. Inoltre, il servizio si occupa automaticamente anche della gestione delle capacità, con la possibilità di ridimensionamento in base alle esigenze dell'applicazione.

Cosmos DB è multi-modello, cioè permette di usare diversi modelli NoSQL per rappresentare i dati. In base al modello scelto vengono usate API differenti:

- **API Core(SQL)**: è nativa di Cosmos DB e archivia i dati in formato documento. Essendo nativa offre la migliore esperienza utente e qualsiasi nuova funzionalità verrà pubblicata per la prima volta in questa API. Come suggerisce il nome, supporta l'esecuzione di query con la sintassi SQL.

- **API MongoDB**: archivia i dati, anche in questo caso, in formato documento. È compatibile con il protocollo di MongoDB, ma non usa il suo codice nativo. Questa API è un'ottima scelta se si vuole usare l'ecosistema e le competenze di MongoDB, senza compromettere le caratteristiche di Cosmos DB.

- **API Cassandra**: archivia i dati usando il modello wide column. Apache Cassandra offre un approccio in scala orizzontale altamente distribuito per l'archiviazione di grandi volumi di dati, offrendo allo stesso tempo un approccio flessibile. L'API Cassandra consente di interagire con i dati usando il linguaggio CQL (Cassandra Query Language) e gli strumenti come la shell CQL e i client di Cassandra già noti.

- **API Gremlin**: consente agli utenti di archiviare i dati come archi e nodi, andando a comporre dei grafi. È utile per scenari che coinvolgono dati dinamici con relazioni complesse oppure dati troppo complessi per essere modellati con database relazionali. L'API Gremlin di CosmosDB si basa sul framework di elaborazione a grafo Apache TinkerPop. Come linguaggio usa GQL (Graph Query Language), sia per inserire che per eseguire query sui dati.

- **API Table**: archivia i dati in formato chiave-valore. L'API Table supera i limiti dell'archiviazione di tabelle del servizio Azure Storage in termini di latenza, scalabilità e velocità.

Per scegliere quale API sia più appropriata per il proprio progetto Microsoft ha reso disponibile nella propria documentazione il diagramma nella fig. 4.7.

Ovviamente in SNA Cosmos DB è stato usato come database dove inserire l'output del Loader. Data la necessità di analizzare la rete di interazioni su Twitter, si è scelto di usare l'API Gremlin.

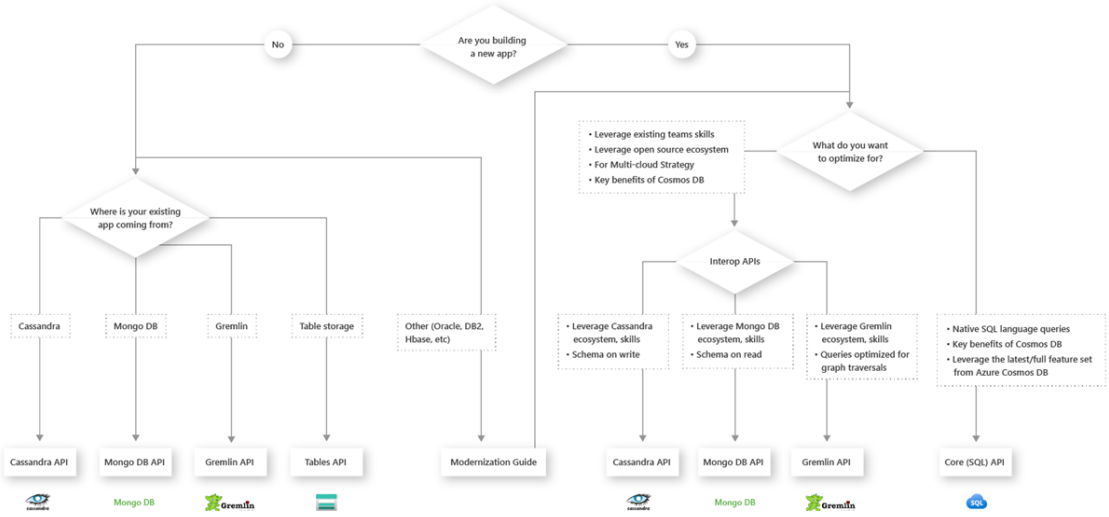

Figura 4.7: Diagramma per la scelta dell'API su CosmosDB

4.2.2 Dashboard

Come mostrato nella fig. 4.8, per la Dashboard è stata scelta un'architettura serverless. Per la parte statica della pagina web si è usato il servizio di archiviazione blob di Azure Storage, già introdotto nella sezione 4.2.1. Questo è possibile grazie alla funzionalità del servizio di fare da hosting per siti web statici. Ovviamente, non essendo lo scopo principale del servizio, presenta alcune limitazioni, ma in questo caso non risulta essere un problema.

Per rendere dinamico il sito la pagina richiama una funzione su Azure Functions attraverso l'interfaccia API esposta. Anche questo servizio è stato già introdotto nella sezione 4.2.1. Tale funzione ha il compito di svolgere diverse operazioni con il seguente ordine:

1. Interpretare la richiesta inviata dalla pagina web;

2. Richiedere al database su Cosmos DB, dove è archiviato l'output del Loader, i dati necessari a rispondere alla richiesta;

3. Analizzare il grafo ricevuto in risposta alla richiesta al database con l'uso di diversi algoritmi illustrati nel sezione 4.1.2;

4. Inviare il grafo e i risultati dell'analisi alla pagina popolandola di contenuto.

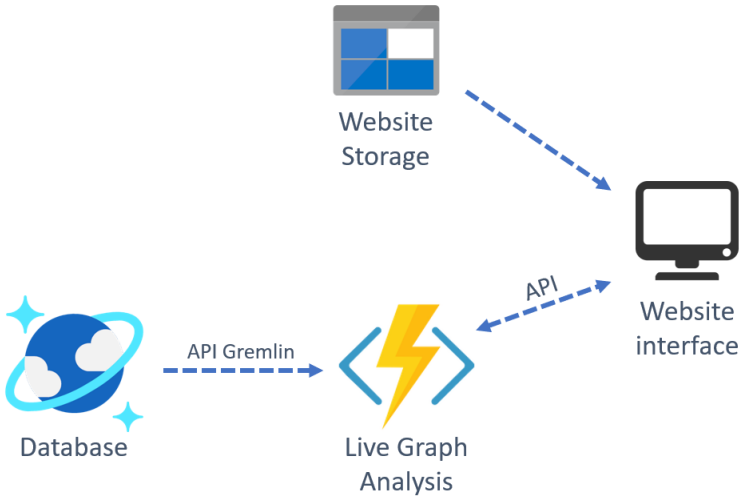

Figura 4.8: Architettura tecnologica della Dashboard

4.3 Implementazione

Quasi tutti i componenti che svolgono operazioni computazionali sono stati implementati in Python. Per quanto riguarda le librerie python necessarie, su Azure Functions è possibile importarle cambiando il file di configurazione, mentre su Azure Databricks è possibile caricarle sul cluster attraverso un'apposita interfaccia grafica.

4.3.1 Loader

TwitterStreaming

TwitterStreaming è la funzione su Azure Functions che si occupa dell'ingestione dei tweet all'interno dell'architettura. Nello pseudocodice nel listing 4.1, che mostra il *main* della funzione, è possibile riconoscere le seguenti operazioni:

1. Instanziamento dei client per comunicare con gli altri servizi:
    - *producer*: è il client per comunicare con Event Hub necessario per accodare i tweet in arrivo;
    - *queue_client*: è il client per la comunicazione con il servizio di Queue di Azure Storage necessario per ricevere comunicazioni dall'esterno;
    - *blob_client*: è il client per comunicare con il servizio di Blob Storage necessario in questo caso per ottenere i parametri di ricerca per Twitter;

Listato 4.1: Pseudocodice della funzione TwitterStreaming

```
producer = EventHubProducerClient.from_connection_string
    ()
queue_client = QueueClient.from_connection_string()
blob_client = BlobClient.from_connection_string()
api = Api(keys)
tags, languages = getParameters()
stream = api.GetStreamFilter(tags, languages)
for tweet in stream:
    sendEvent(tweet)
    if checkAction() == Action.STOP :
        break
stream.close()
return func.HttpResponse("HTTP triggered function
    executed successfully.")
```

- *api*: è il client che si occupa della comunicazione con le API di Twitter ed è importato da python-twitter, libreria in python che mette a disposizione un'interfaccia con i servizi di Twitter.

2. Attraverso la funzione *getParameters* ottenimento dall'apposito blob dei parametri di ricerca da passare successivamente all'API di Twitter;

3. Creazione del flusso di dati da Twitter attraverso l'uso della funzione *GetStreamFilter* del client *api*.

4. Avvio del ciclo sul flusso in entrata da Twitter all'interno del quale si fanno due operazioni:

   - Chiamata alla funzione *sendEvent* che si occuperà di inviare il nuovo tweet alla coda su Event Hub attraverso l'uso del client *producer*;
   - Attraverso la funzione *checkAction* controllo della coda in cui vengono aggiunte le *Action* per comunicare messaggi alla funzione dall'esterno. *Action* è un enumeratore con tutte le comunicazioni possibili e *Action.STOP* è quella che comunica alla funzione di uscire dal ciclo e di terminare.

5. Chiusura del flusso da Twitter attraverso la funzione *close* del client *api*;

6. Terminazione della funzione con l'invio della risposta al chiamante per avvertirlo del completamento della funzione.

Listato 4.2: Pseudocodice della funzione StreamingController

```
queue_client = QueueClient.from_connection_string()
msg = req.route_params.get("msg")
if msg in [action.name for action in Action] :
    queue_client.send_message(msg)
    return func.HttpResponse(f"Add {msg} action in queue
       .")
else :
    return func.HttpResponse(f"Action {msg} isn't
       allowed.")
```

StreamingController

La funzione StreamingController si occupa invece di inserire nuove *Action* all'interno della coda su Azure Storage per comunicare dall'esterno dei comandi alla funzione TwitterStreaming.

Nello pseudocodice mostrato nel listing 4.2 si possono individuare le seguenti operazioni:

1. Instanziamento del client per la comunicazione con la coda.

2. Recupero dalla chiamata alla funzione del contenuto del parametro *msg* e controllo che sia una delle *Action* definite.

3. Se il controllo va a buon fine, la *Action* viene inserita nella coda grazie al client e viene inviata all'utente la risposta con l'esito positivo.

4. Se invece il controllo non va a buon fine, viene inviata all'utente la risposta con esito negativo.

Notebook Databricks

Su Databricks per implementare la parte di estrazione dati e caricamento su database si è usato come ambiente di sviluppo Databricks Data Science Engineering. Questo ha permesso di scrivere il codice in modo interattivo su un notebook.

All'interno di esso è stato possibile usare la libreria Pyspark, l'API python per Apache Spark, in quanto è già installata di base all'interno del cluster instanziato. Apache Spark è un framework su cluster general-purpouse, adatto sia per elaborazioni in batch che in streaming. In questo caso è stato usato in un contesto streaming, più nello specifico è stato usato lo Structured Streaming di Spark.

Lo Structured Streaming è un motore d'esecuzione di flusso scalabile e fault tolerant basato sul motore Spark SQL. Grazie ad esso è possibile esprimere un'elaborazione in streaming nello stesso modo in cui si esprimerebbe un'elaborazione in batch su dati statici. Questo è possibile perché, man mano che i dati in streaming continuano ad arrivare, il motore Spark SQL si occupa di svolgere l'elaborazione in modo incrementale e continuo e di aggiornare il risultato finale. Internamente, per svolgere l'elaborazione, viene utilizzato un motore d'esecuzione a micro-batch, che elabora i flussi di dati come una serie di piccole elaborazioni in batch. Per svolgere le diverse operazioni come se fossero dati statici si usano le astrazioni dataset e dataframe.

All'interno del notebook le operazioni principali che sono state fatte sul dataframe sono:

1. Creazione del dataframe indicando la fonte dei dati, cioè la coda di Event Hubs. Per collegare Event Hubs è stato usato un connettore appositamente creato da Microsoft.

2. Conversione dello schema JSON del tweet nella forma tabellare del dataframe.

3. Individuazione della tipologia del tweet in base ai dati che contiene.

4. Selezione dei tweet appartenenti solo alle tipologie ritenute utili per la rete che si vuole creare, cioè tweet e retweet.

5. Individuazione dei topic di appartenenza dei vari tweet. È stato fatto con un'espressione regolare usando i parametri di ricerca scaricati dal blob. Un tweet può appartenere a più topic.

6. Estrazione delle altre informazioni necessarie come ad esempio gli hashtag usati, gli url e il timestamp di creazione del tweet.

7. Esplosione delle righe in base ai topic individuati. Ciò vuol dire che se una riga ha due maintopic, questa viene sdoppiata in modo che ogni riga abbia un solo maintopic.

8. Creazione di sei flussi diversi, uno per ogni tipologia di nodo e arco.

9. In ogni flusso eliminazione dei dati non necessari e sistemazione del formato di quelli rimasti.

10. Creazione di due dataframe nati dall'unione dei flussi: in uno ci sono tutti i dati relativi ai nodi mentre nell'altro tutti quelli degli archi. Non è possibile riunirli tutti in un unica tabella per problemi di formattazione.

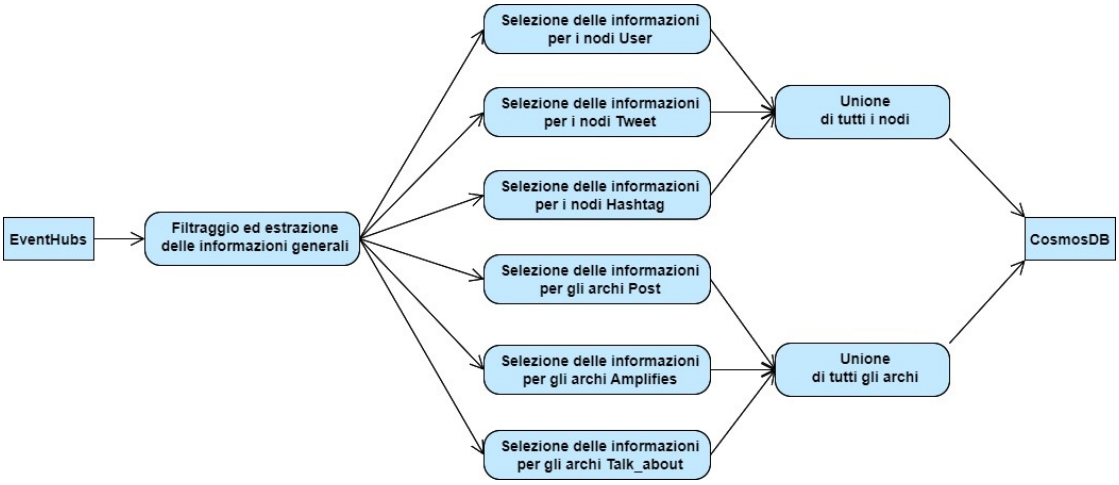

Figura 4.9: Diagramma rappresentante il flusso di dati su Spark

11. Caricamento e aggiornamento del grafo sul database attraverso l'uso di un connettore appositamente creato da Microsoft per collegare Spark a Cosmos DB.

4.3.2 Dashboard

Per quanto riguarda l'implementazione della dashboard è stato possibile sfruttare il codice di un altro progetto simile, ma che usava al posto dei servizi Azure quelli di Amazon Web Services(AWS). Per cui il lavoro svolto in questo progetto per quanto riguarda la dashboard è stato quello di adattare il codice ai servizi Azure. Essendo questi molto simili, l'adattamento non ha comportato molti cambiamenti. Le pagine web non sono state modificate, sono state semplicemente caricate su blob storage attivando il servizio di hosting in modo tale che fossero accessibili come siti web. Nella funzione che gestisce la parte dinamica del sito invece sono stati cambiati i client per la comunicazione con gli altri servizi e il modo in cui la funzione della dashboard richiedeva dati al database.

La funzione che la pagina statica deve richiamare per ricevere i dati è una sola perché si segue il Command Pattern: in base ai parametri della richiesta la funzione ne richiamerà altre che andranno a svolgere ciò che è stato richiesto. Per le interrogazioni al database viene usato gremlin, un linguaggio per query su grafo. A differenza di Neptune, servizio di AWS per la gestione di database a grafo, Cosmos DB non supporta Gremlin Bytecode, ciò non permette di costruire le query gremlin direttamente nel linguaggio di programmazione. Per cui è stato necessario utilizzare la funzione del client *SubmitAsync* mettendo come argomento la query in formato stringa.

Altro elemento fondamentale di questa funzione è l'analisi del grafo. Questa avviene grazie a NetworkX, una libreria python che permette di creare, manipolare e analizzare i grafi. Una volta estratti gli archi del grafo che si vuole analizzare dal database, questi vengono usati per creare il grafo su NetworkX. Dopo di ciò sono state sfruttate le varie funzioni offerte da tale libreria per ottenere le diverse informazioni riguardanti il grafo e per applicare alcuni algoritmi di analisi, come il PageRank di Google e il Clauset-Newman-Moore greedy modularity maximization.

Quando l'utente richiede la visualizzazione di un nuovo grafo, dopo aver definito il tipo di interfaccia, il topic e l'intervallo di tempo che vuole consultare, all'interno della funzione d'analisi avvengono le seguenti operazioni:

1. Interpretazione della richiesta dell'utente;
2. Richiesta al database del grafo che si vuole analizzare attraverso l'uso di interrogazioni in gremlin sfruttando l'apposito client;
3. Importazione del grafo ottenuto sulle strutture dati della libreria NetworkX;
4. Calcolo delle statistiche sul grafo, come ad esempio il numero di nodi e archi;
5. Applicazione dell'algoritmo PageRank per l'individuazione dei nodi più importanti;
6. Applicazione del Clauset-Newman-Moore greedy modularity maximization per l'individuazione delle comunità;
7. Trasformazione dei dati in modo che possano essere utilizzabili dal sito;
8. Invio del grafo e dei risultati delle analisi all'utente.

Capitolo 5

Test

Per misurare le performance dell'implementazione di SNA sono stati svolti diversi test con differenti configurazioni. Ci si è focalizzati principalmente nello svolgimento di test di carico sul Loader, dato che è la parte più critica all'interno dell'architettura. Per tutti i test sono stati usati gli stessi parametri di ricerca, cioè alcune parole chiave di tre topic: Covid 19, Italia e Formula 1.

Per misurare le performance del Loader, ogni volta che i dati facevano un nuovo step nell'architettura, veniva salvato al loro interno il *timestamp* di quando avveniva. In questo modo è stato possibile misurare gli intervalli di tempo impiegati per i vari passaggi.

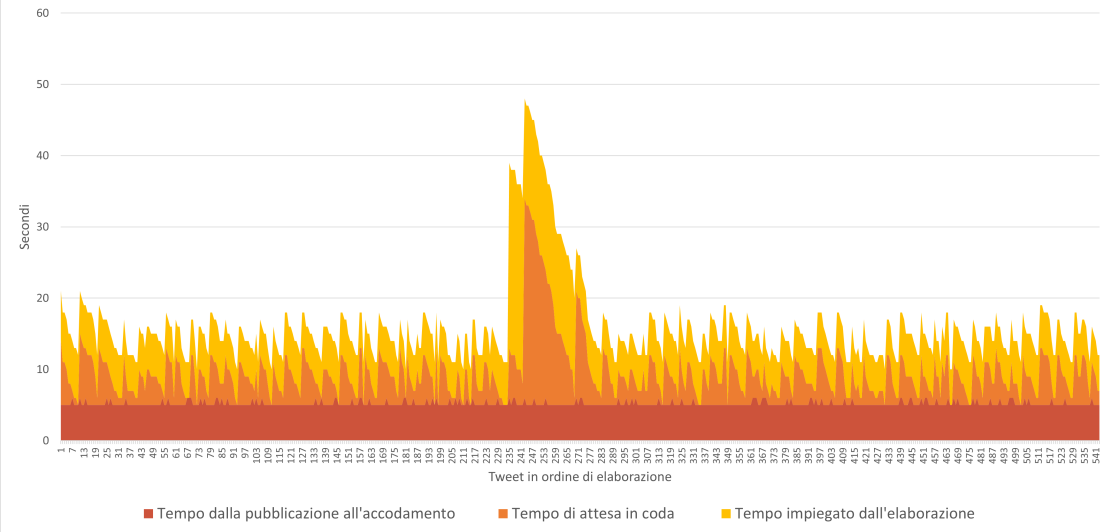

Figura 5.1: Grafico raffigurante i tempi del Loader con tweet in tempo reale, 1 partizione di EventHubs e 2 worker nel cluster

Un primo test è stato svolto con i dati reali, connettendo l'architettura direttamente a Twitter, con questa configurazione: una sola partizione su Event Hubs e un cluster composto da un driver e due worker DS3v2, cioè con istanze *general-purpose*. Nel grafico della fig. 5.1 è possibile vedere gli intervalli di tempo dei passaggi di ogni tweet pubblicati su Twitter in 10 minuti. In questo caso gli intervalli sono tre:

- Intervallo di tempo tra la pubblicazione del tweet e inserimento nella coda su Event Hubs. Comprende quindi il passaggio attraverso la funzione di ingestione. Nel grafico è in rosso e si può notare che è quasi costante sui 5 secondi.

- Tempo di attesa nella coda di Event Hubs: viene fatta la differenza tra il timestamp di entrata nel cluster e quello di accodamento. Nei diversi grafici è in arancione e, anche se nei prossimi è più evidente, già da qui si può notare come varia in base al tempo di elaborazione del batch precedente, facendo assumere al grafico una forma seghettata. In questo caso la media del tempo di attesa in coda è di circa 4 secondi e mezzo.

- Tempo impiegato dall'elaborazione sul cluster: viene fatta la differenza tra il timestamp in entrata e quello in uscita dal cluster. Nei grafici è in giallo ed è possibile notare in ogni grafico che i tweet elaborati nello stesso batch hanno giustamente lo stesso tempo di elaborazione. In questo caso ha una media di circa 6 secondi. È abbastanza variabile perché dipende dal contenuto dei tweet.

Nel complesso in questo test il tempo che passa dalla pubblicazione del tweet al caricamento dei dati estratti sul database a grafo è in media di circa 16 secondi. Nel grafico è evidente la presenza di un picco: è possibile individuare la causa nel prolungamento dei tempi di elaborazione di un batch, probabilmente per la presenza di qualche tweet un po' più ostico da analizzare.

Essendo Twitter la fonte di dati, il flusso normalmente risulta molto variabile. Esso infatti può cambiare ad esempio in base ai momenti della giornata o alle notizie e post del momento, o come si dice su Twitter, in tendenza. Per questa ragione, se i test venissero svolti tutti come quello appena mostrato, i dati ricavati da essi non sarebbero confrontabili e di conseguenza non sarebbe possibile capire quale sia la configurazione migliore. Per aggirare questo problema sono stati salvati 5 mila tweet sul servizio di archiviazione blob di Azure Storage ed è stata aggiunta la possibilità di prendere i tweet da esso alla funzione di ingestione.

Un'altra soluzione a questo problema consiste nello sfruttamento della *retention* di Event Hubs per trattenere i tweet nella coda. In questo modo è possibile eseguire la parte di estrazione sui dati trattenuti inserendo una data di inizio

flusso antecedente a quella del loro arrivo. In questa soluzione però non è possibile misurare il tempo di attesa in coda, per cui nei prossimi test è stata usata la prima soluzione. Ovviamente in nessuno dei due casi è possibile testare la comunicazione con Twitter.

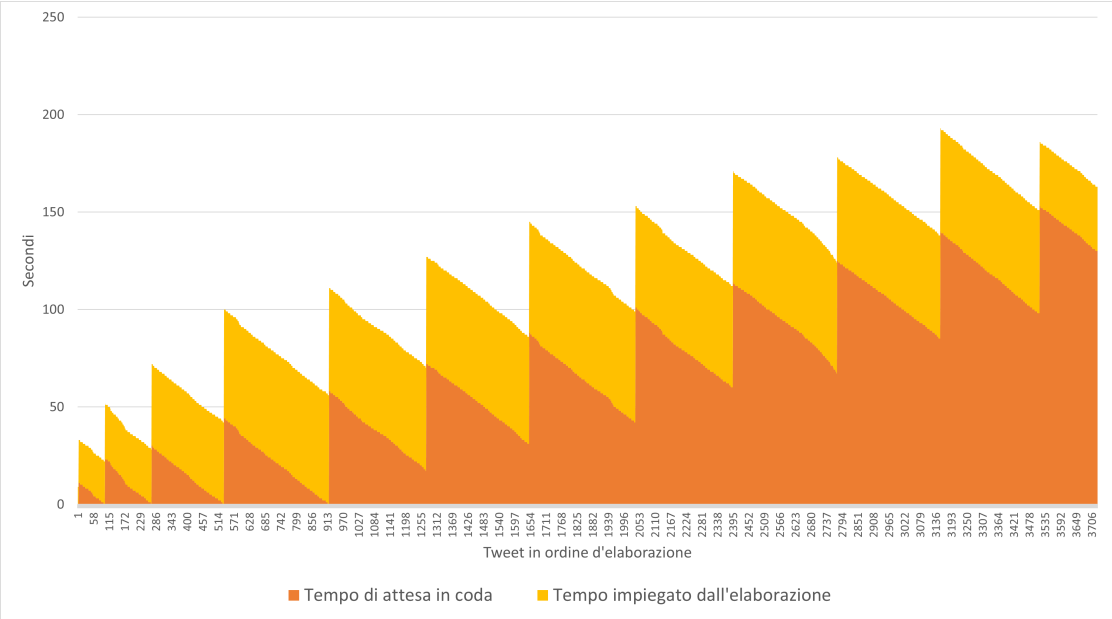

Figura 5.2: Grafico raffigurante i tempi del Loader con tweet da blob, 1 partizione di EventHubs e 1 solo worker nel cluster

Nel test del grafico di fig. 5.2 è stata usata una configurazione con una partizione su Event Hubs e un solo worker nel cluster oltre al driver. Come si può notare nel grafico, il loader rimane gradualmente sempre più indietro rispetto al flusso, non riuscendo nemmeno a recuperare. In media il tempo d'attesa in coda nell'EventHubs è di circa 67 secondi, mentre il tempo di elaborazione nel cluster è di 50 secondi. In totale per elaborare tutti i 5 mila tweet ci ha impiegato 594 secondi.

Nel test del grafico di fig. 5.3 è stata usata una configurazione con una partizione su EventHubs e due worker nel cluster oltre al driver. In media il tempo d'attesa in coda nell'EventHubs è di circa 18 secondi, mentre il tempo di elaborazione nel cluster è di 35 secondi. In totale per elaborare tutti i 5 mila tweet ci ha impiegato 579 secondi.

Nel test del grafico di fig. 5.4 è stata usata una configurazione con due partizioni su EventHubs e un worker nel cluster oltre al driver. In media il tempo d'attesa in coda nell'EventHubs è di circa 9 secondi, mentre il tempo di elaborazione nel

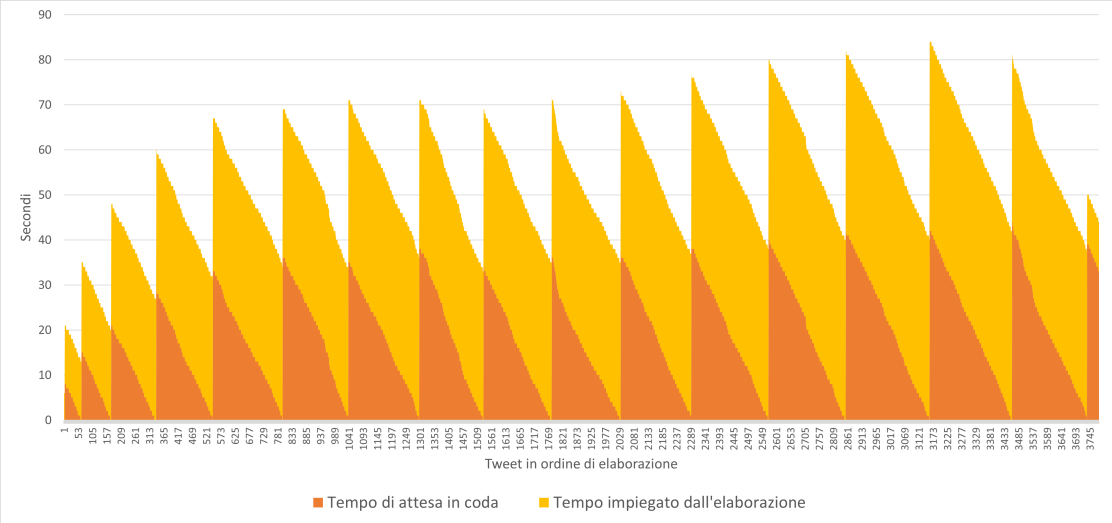

Figura 5.3: Grafico raffigurante i tempi del Loader con tweet da blob, 1 partizione di EventHubs e 2 worker nel cluster

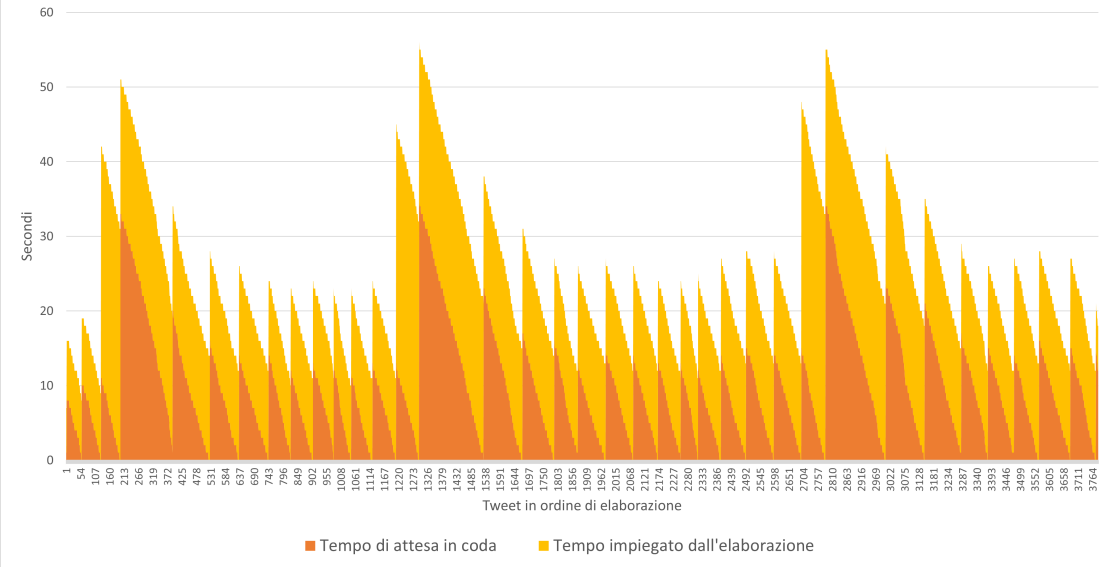

Figura 5.4: Grafico raffigurante i tempi del Loader con tweet da blob, 2 partizioni di EventHubs e 1 worker nel cluster

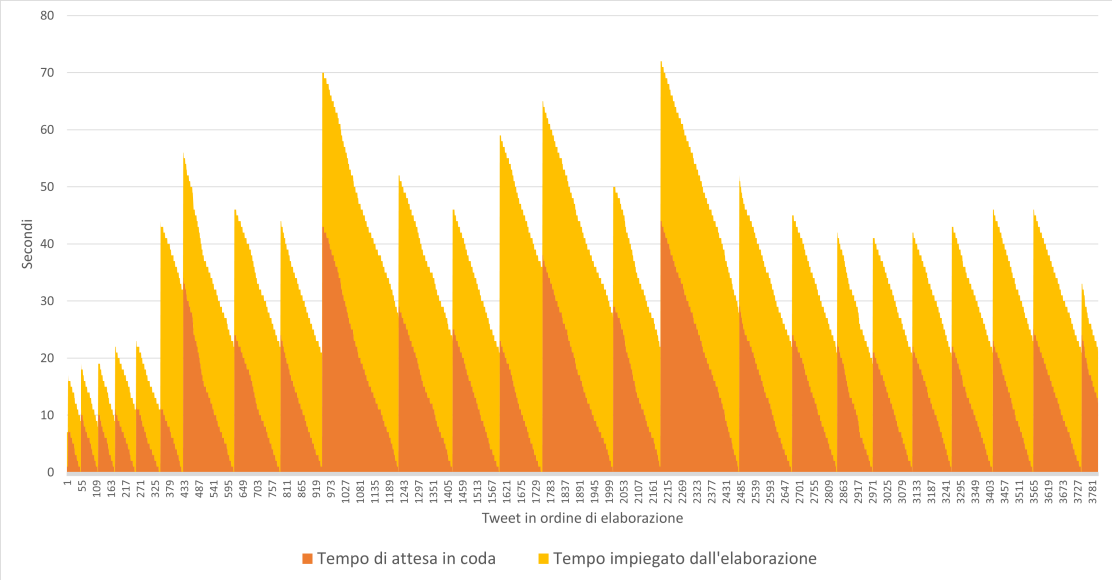

Figura 5.5: Grafico raffigurante i tempi del Loader con tweet da blob, 2 partizioni di EventHubs e 2 worker nel cluster

cluster è di 15 secondi. In totale per elaborare tutti i 5 mila tweet ci ha impiegato 612 secondi.

Nel test del grafico di fig. 5.5 è stata usata una configurazione con due partizioni su EventHubs e due worker nel cluster oltre al driver. In media il tempo d'attesa in coda nell'EventHubs è di circa 14 secondi, mentre il tempo di elaborazione nel cluster è di 22 secondi. In totale per elaborare tutti i 5 mila tweet ci ha impiegato 599 secondi.

N. partizioni	N. worker	Attesa in coda	Elaborazione	Totale
1	1	67	50	594
1	2	18	35	579
2	1	9	15	612
2	2	14	22	599

Tabella 5.1: Media dei tempi di attesa in coda ed elaborazione di ogni tweet in secondi per ogni configurazione di test

Le ultime tre configurazioni sono riuscite tutte e tre a rimanere dietro al flusso anche se in quasi tutti i grafici sono visibili dei picchi momentanei. Quando l'area gialla è la prima che si alza più del normale, quindi nella maggior parte dei casi, è possibile identificare la causa nel rallentamento momentaneo della fa-

se di elaborazione dei tweet, probabilmente per la presenza di tweet un po' più complessi.

Se si dovesse giudicare in base alla durata dei singoli intervalli la migliore configurazione sarebbe quella con due partizioni di Event Hubs e un worker solo nel cluster(fig. 5.4). Nonostante ciò, i tempi di esecuzione totali dicono quasi l'opposto, infatti con i suoi 612 secondi risulta essere il test peggiore. Probabilmente avendo batch più piccoli ne deve fare molti di più e ciò lo rallenta. Se volessimo valutare le configurazioni in base ai tempi d'esecuzione totali, il migliore sarebbe quello con una partizione di Event Hubs e due worker nel cluster, ma, riuscendo tutti a stare dietro al flusso, tutti i tempi sono più o meno allineati. Probabilmente il vero collo di bottiglia in questo caso risulta essere l'invio dei dati alla coda di Event Hubs da parte della funzione di ingestione.

Capitolo 6

Conclusioni

L'obiettivo della tesi era quello di fornire uno strumento che aiutasse il reparto marketing di un'azienda nella scelta delle proprie strategie commerciali. Nello sviluppo della tesi è stato prima di tutto necessario approfondire l'argomento marketing per comprendere meglio le esigenze di un'azienda e in seguito trovare in che modo i social media potevano essere sfruttati dal punto di vista aziendale. Da ciò è stato possibile porre degli obiettivi più specifici: lo strumento doveva essere in grado, una volta scelto il topic, di identificare gli argomenti più discussi e gli influencer capaci di raggiungere più persone. Queste informazioni potrebbero essere sfruttate dall'azienda in vari modi, ad esempio gli argomenti permetterebbero di focalizzarsi sui temi che il pubblico ritiene più importanti, mentre gli influencer potrebbero essere usati come promotori dei propri prodotti.

Per raggiungere tale obiettivo si è progettata e implementata un'architettura streaming basata sui servizi offerti da Azure in grado di raccogliere, estrarre e salvare i dati in una database a grafo, chiamato Loader. Poi in un secondo momento è stata realizzata, sempre con i servizi offerti da Azure, la Dashboard, cioè la parte che si occupa di interrogare il database, analizzare i grafi estratti e rappresentarli in un'interfaccia grafica.

Dalla Dashboard è possibile visualizzare in modo intuitivo tutta la rete di interazioni con le relative statistiche e i risultati delle analisi, raggiungendo così l'obiettivo prestabilito. Anche se non c'è stata la possibilità di utilizzare una configurazione con un cluster composto da più di tre macchine, i test sul Loader hanno mostrato la sua efficacia sia con un carico normale che in sovraccarico riuscendo a rimanere dietro ad un flusso di 5 mila tweet in circa 10 minuti. In situazioni di carico normale l'architettura è stata in grado di aggiornare lo stato del proprio database in media dopo soli 16 secondi dalla pubblicazione del contenuto.

Gli obiettivi che ci si è posti rappresentano solo una parte di ciò che è possibile realizzare con i dati ricavabili dai social media. Durante lo sviluppo sono emersi tanti altri aspetti da approfondire:

- Implementazione di un algoritmo di Natural Language Processing (NLP) che riesca a cogliere il topic d'appartenenza del tweet. Questo eviterebbe l'utilizzo delle espressioni regolari che essendo più "rigide" spesso non assegnano efficacemente il tweet al suo rispettivo topic.

- Estrazione di altre informazioni sfruttabili dal tweet: molte informazioni all'interno del tweet non vengono sfruttate, ad esempio gli url al loro interno. È possibile estrarre anche informazioni più nascoste usando degli algoritmi, ad esempio applicando quelli di NLP sul contenuto del tweet si potrebbe capire se un tweet espone un'opinione positiva, negativa o neutra.

- Implementazione di altre analisi sul grafo. Il grafo attualmente viene sottoposto solo a due algoritmi di analisi, cioè PageRank e Clauset-Newman-Moore greedy modularity maximization. In realtà sulla libreria NetworkX, cioè quella che è stata usata per le analisi sul grafo, sono presenti tanti altri algoritmi che potrebbero essere utili ad individuare altre caratteristiche.

- Implementazione di un sistema d'accesso: per rendere la pagina accessibile solo a chi di dovere sarebbe bene aggiungere una procedura d'accesso. Dato che si hanno tutti i servizi su Azure è possibile sfruttare il servizio d'identità Azure Active Directory combinato con la possibilità di rendere le funzioni di Azure Functions accessibili solo dagli utenti autenticati.

- SNA è stato progettato e implementato su Twitter, ma è possibile portare l'idea su tutti quei social media che permettono di ricevere aggiornamenti in tempo reale sul loro contenuto attraverso l'esposizione di API.

Bibliografia

[1] Data never sleeps 9.0. https://www.domo.com/learn/infographic/data-never-sleeps-9. Accessed: 2021-02-19.

[2] The marketing mix 7p's. https://marketingmix.co.uk/marketing-mix-7ps/. Accessed: 2021-02-10.

[3] Social media - statistics & facts. https://www.statista.com/topics/1164/social-networks/. Accessed: 2021-02-11.

[4] Social networksblogs now account for one in every four and a half minutes online. https://www.nielsen.com/us/en/insights/article/2010/social-media-accounts-for-22-percent-of-time-online/. Accessed: 2021-02-10.

[5] Divyakant Agrawal, Philip Bernstein, Elisa Bertino, Susan Davidson, Umeshwar Dayal, Michael Franklin, Johannes Gehrke, Laura Haas, Alon Halevy, Jiawei Han, et al. Challenges and opportunities with big data. a community white paper developed by leading researchers across the united states. *Computing Research Association, Washington*, 2012.

[6] Kaplan Andreas and Haenlein Michael. Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53:59–68, 02 2010.

[7] Bernard Booms. Marketing strategies and organizational structures for service firms. *Marketing of services*, 1981.

[8] danah m. boyd and Nicole B. Ellison. Social network sites: Definition, history, and scholarship. *J. Comp.-Med. Commun.*, 13(1):210–230, oct 2007.

[9] Wo Chang, David Boyd, and Orit Levin. Nist big data interoperability framework: Volume 6, reference architecture, 2019-10-21 2019.

[10] Aaron Clauset, M Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 70:066111, 01 2005.

[11] Mick Albert Ed Price et al. Big data architectures. `https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/`. Accessed: 2021-02-20.

[12] Byron Ellis. *Real-Time Analytics: Techniques to Analyze and Visualize Streaming Data*. Wiley, 2014.

[13] Bill Franks. *Taming the big data tidal wave: Finding opportunities in huge data streams with advanced analytics*. John Wiley & Sons, 2012.

[14] L.C. Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press, 2004.

[15] Anil Jain. The 5 v's of big data. `https://www.ibm.com/blogs/watson-health/the-5-vs-of-big-data/`. Accessed: 2021-02-19.

[16] Jan H. Kietzmann, Kristopher Hermkens, Ian P. McCarthy, and Bruno S. Silvestre. Social media? get serious! understanding the functional building blocks of social media. *Business Horizons*, 54(3):241–251, 2011. SPECIAL ISSUE: SOCIAL MEDIA.

[17] Doug Laney et al. 3d data management: Controlling data volume, velocity and variety. *META group research note*, 6(70):1, 2001.

[18] M-Brain. M-brain technology. `https://www.m-brain.com/technology/`. Accessed: 2021-02-20.

[19] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers, et al. *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute, 2011.

[20] Katja Mayer. *On the sociometry of search engines. a historical review of methods*, pages 54–72. 01 2009.

[21] Edmund Jerome McCarthy, Stanley J Shapiro, and William D Perreault. *Basic marketing*. Irwin-Dorsey Ontario, CA, USA, 1979.

[22] Chenda Ngak. Then and now: a history of social networking sites. `https://www.cbsnews.com/pictures/then-and-now-a-history-of-social-networking-sites/2/`. Accessed: 2021-02-10.

[23] Jonathan Obar and Steven Wildman. Social media definition and the governance challenge: An introduction to the special issue. *SSRN Electronic Journal*, 01 2015.

[24] Peter O'Connor. Managing a hotel's image on tripadvisor. *Journal of Hospitality Marketing & Management*, 19(7):754–772, 2010.

[25] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[26] Jordi Paniagua and Juan Sapena Bolufer. Business performance and social media: Love or hate? *Business Horizons*, 57:719–728, 07 2014.

[27] Andrew G. Psaltis. *Streaming Data*. Manning, 2017.

[28] George Ritzer and Nathan Jurgenson. Production, consumption, prosumption: The nature of capitalism in the age of the digital 'prosumer'. *Journal of Consumer Culture*, 10(1):13–36, 2010.

[29] Pramod J. Sadalage and Martin Fowler. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Pearson Education, 2013.

[30] W.J. Stanton and R. Varaldo. *Marketing*. Strumenti Il mulino: Economia. Il Mulino, 1986.

[31] Aichner Thomas and Jacob Frank. Measuring the degree of corporate social media use. *International Journal of Market Research*, 57:257–275, 03 2015.