Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження методів динамічного аналізу даних для визначення експертної думки на основі стрімінгової обробки даних з Twitter

Виконав:

студент 2 курсу групи ІПЗм-20-2

Єрусалімцев Д. А.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення

Тип програми Освітньо-наукова

Керівник доц. Каук В. І.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри З.В. Дудар

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерні науки

Кафедра Програмна Інженерія

Рівень вищої освіти перший (бакалаврський)

Спеціальність 121 — Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«___» _____ 202_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Єрусалімцеву Денису Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів динамічного аналізу даних для визначення експертної думки на основі стрімінгової обробки даних з Twitter»

затверджена наказом по університету від «__» _____ 202_ р. №____

2. Термін подання студентом роботи до екзаменаційної комісії «__» ____ 202_ р.

3. Вихідні дані до роботи: середовище проектування Intellij Idea 2021, мова розробки Scala, хмарні сервіси Microsoft Azure, база даних Cassandra, інструмент обробки даних Flink інструмент обробки даних Spark.

4. Перелік питань, що потрібно опрацювати в роботі реферат, вступ, аналіз проблемної галузі, огляд патентної літератури, формування вимог до програмної системи, архітектура та проектування програмної системи, опис прийнятих програмних рішень, тестування розробленого ПЗ, впровадження програмного забезпечення.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	26.02.2022	Виконано
2	Формування вимог до системи	14.03.2022	Виконано
3	Проектування системи	21.03.2022	Виконано
4	Розробка програмного забезпечення	24.03.2022	Виконано
5	Тестування системи	15.04.2022	Виконано
6	Впровадження системи	16.04.2022	Виконано
7	Підготовка пояснювальної записки	20.04.2022	Виконано
8	Підготовка презентації та доповіді	01.05.2022	Виконано
9	Перевірка на плагіат		Виконано
10	Нормоконтроль		Виконано
11	Рецензування		Виконано
12	Занесення диплома в електронний архів		Виконано
13	Попередній захист		Виконано
14	Допуск до захисту у зав. Кафедри		Виконано

Дата видачі завдання _____202_ р.

Студент _____

(підпис)

Керівник роботи _____ доц. Каук В.І

(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи магістра, 74 стор., 20 рис., 18 джерел.

ПРОЕКТ, УПРАВЛІННЯ, AZURE, BIG DATA, CASSANDRA, FLINK, MATERIAL, SPARK, PIPELINES TEXTANALYTICS, TWITTER.

Об’єктом дослідження є методи динамічного аналізу даних для визначення експертної думки на основі стрімінгової обробки даних з Twitter.

Метою роботи є розробка платформи даних, що дозволяє аналізувати повідомлення користувачів щодо вказаних тем та шляхом обробки повідомлень та поділення їх на групи отримувати ті, які мають експертну думку.

Методи рішення базуються на технології та Apache Flink з використанням бази даних Cassandra та Cloud-серверу Azure.

У результаті роботи були проаналізовані методи дослідження потокових даних та в якості експерименту було реалізовано систему для аналізу даних з Twitter.

Explanatory note consists of: 74 p., 20 pictures, 18 sources.

MANAGEMENT, AZURE, BIG DATA, CASSANDRA FLINK, SPARK, PROJECT, PIPELINES, TEXTANALYTICS, TWITTER.

The object of research is the methods of dynamic data analysis to determine expert opinion based on streaming data processing from Twitter.

The aim of the work is to develop a data platform that allows you to analyze user messages on these topics and by processing messages and dividing them into groups to get those who have expert opinion.

The solution methods are based on technology and Apache Flink using the Cassandra database and the Azure Cloud Server.

As a result, the methods of researching streaming data were analyzed and a system for analyzing data from Twitter was implemented as an experiment.

Умови публікації пояснювальної записки

Я,

_____

(прізвище, ім’я, по батькові)

студент(ка) групи _____здобувач вищої освіти на другому (магістерському) рівні

кафедра _____програмної інженерії_____,

(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему

_____,

(назва роботи)

що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

Щодня кількість створюємих даних збільшується. Технологічний процес привів за собою великі зміни у світі зберігання, обробки та сприйняття інформації. Як результат, з'явився новий термін, який описує масштаби нового світу – великі дані.

Майбутнє вже наближається, і великі дані невпинними темпами відкривають нові досягнення у світі технологій. За останні два роки великі дані змінили сам погляд на компанії та спосіб зберігання даних, тепер дозволяє кожному доступні точні маніпуляції з великими обсягами даних, і було виявлено, що щодня виробляється 2,5 квінтильйона байт даних, в майбутньому ця кількість буде тільки збільшуватися.

Кожна компанія, незалежно від її розміру, генерує дані. Це може бути інформація про клієнтів, дані про співробітників, дані про продажі, трафік користувачів їх інтереси, вподобання. Також всім відомо, що підприємства, заводи та виробництва будь-якого розміру генерують велику кількість інформації: стан станків, етапи виробництва, тощо. Не менш важливим та помітним є розвиток так званих розумних міст, де кожний елемент на вулиці може надавати інформацію, яку потім можна обробити.

Незалежно від типу даних, вони відіграють важливу роль, коли мова йде про покращення якості послуг. Існує велика кількість способів, за допомогою яких великі дані вже сьогодні змінюють бізнес.

Великі дані – це не тільки бізнес. Багато лікарів і медичних працівників бачать, як один і той же тип інформації, що використовується для даних про продажі, також може використовуватися для даних про здоров'я. Кінцевою метою було б надання персоналізованої медичної допомоги приблизно так само, як підприємства звертаються до окремих клієнтів за допомогою цільового маркетингу. Цей персоналізований підхід до охорони здоров'я сприяє кращому лікуванню для кожного пацієнта, а також детальні дані, які можна передавати від

лікаря до лікаря, щоб переконатися, що вони мають всю необхідну інформацію, щоб допомогти кожній людині.

Зараз ми живемо в епоху великих даних, коли прориви та революційні зміни відбуваються через регулярні проміжки часу. Одним з таких джерел інформацій стають соціальні мережі. Кожен день люди проводять години за промотуванням стрічки соціальних мереж. Саме через ці стрічки користувачі отримують базове сприйняття подій у світі. У той час існує потреба коректної та швидкої обробки великих обсягів даних. Особливо гостро стоїть питання обробки даних у режимі реального часу, коли рішення необхідно прийняти за секунди. А як результат, попередня проблема веде за собою іншу – з'являється проблема класифікації якості даних, які споживаються.

Для вивчення деяких з таких проблем було прийнято рішення провести дослідження з розробки методів динамічного аналізу даних для визначення експертної думки на основі стрімінгової обробки даних з соціальної мережі Twitter. Це дослідження планується для максимального спрощення аналізу та класифікації вхідної інформації, яка отримана із соціальної мережі Twitter. Аналіз потоку вхідної інформації, а саме твітів, інформації про автора, коментарі інших користувачів, та спеціальні публічні метрики дозволять виділити достовірні джерела та розподілити авторів на різні рівні довіри користувачів.

Планується, що обробка вхідних даних буде виконуватися на кластері, що складається з безлічі серверів. Кількість серверів буде масштабуватися в залежності від навантаження. Для задоволення цієї вимоги в якості платформи для розгортання була обрана Microsoft Azure з сервісом Azure Kubernetes Services, який дозволяє розгортати різні типи віртуальних ресурсів. В якості інструменту розподілених систем був обраний Flink. В якості моделі для розподілених обчислень був обраний MapReduce. Суть цього алгоритму в поділі входять даних на рівні за розміром складові, і обробку кожної складової на окремому сервері за заданим алгоритмом.

Актуальність роботи зумовлена активним зростання попиту на кваліфіковану та достовірну інформацію. Щодня кількість користувач соціальних мереж зростає,

разом із тим зростає кількість фейкової інформації. Така інформація є шкідливою, тому що вона може нести за собою як майнові так і немайнові втрати. Крім того, через розповсюдження пандемії COVID-19 гостро стало питання якості дистанційної освіти [1]. Через це з’явилося багато неякісних джерел інформації, які публікують освітні матеріали низького гатунку. Ці матеріали можуть дестабілізувати навчальний процес у школах та вишах, та нанести велику втрату основному ресурсу – часу [2]. Поточне дослідження допоможе виявляти та відсортовувати подібну інформацію.

Критерії успіху даної системи базуються на тому, що система буде надавати можливість уникнути додаткових витрат на аналіз різних типів інформації, а також заощаджувати час на пошук достовірної інформації, яка буде розподілена за рейтингом довіри користувачів. Із допомогою технології Big Data та Recognition Services інформація буде впорядкована та розподілена заздалегідь, що дозволить кінцевому користувачу отримувати найбільш якісну інформацію [3].

Основною метою цього проекту є реалізація платформи даних, яка буде аналізувати потоки даних з соціальної мережі Twitter та надавати інформацію, яка допоможе отримувати найбільш потрібні знання у заданих темах, що дозволить оптимізувати пошук кваліфікованих авторів. Вся інформація буде зберігатися у хмарі, сама платформа даних буде крос-системної.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Загальний аналіз галузі

У наш час, коли величезна кількість людей проводить багато часу в інтернеті, користуються ним для отримання необхідної інформації та комунікації з іншими людьми, необхідним стає створення сервісів, що спрощують ці дії. Пошук новин, статей та цікавого контенту – все це можливо завдяки сучасному інтернету. У той час, для економії часу для пошуку достовірних джерел, стає актуальним питання щодо можливості фільтрації контенту шляхом його аналізу.

Отримання внутрішньої аналітики є дуже корисною для розуміння, наскільки джерело інформації є надійним. Такий швидкий аналіз контенту, що публікується дозволяє приймати правильні бізнес-рішення. Крім того, скоротити велику кількість часу дозволить інформація про те, чи були джерела інформації помічені у замовних статтях, або мають негативні відгуки.

Така аналітична інформація є особливо корисною, але щоб отримати її необхідно користуватися послугами кваліфікованих аналітиків. Також є варіант займатися аналізом самостійно, але слід враховувати, що через це будуть витрачені важливі ресурси – гроші та час.

У наш час, щоб отримати експертну думку на будь-яку тему, необхідно проводити години за відбором інформації, відвідувати конференції, тощо. Як результат, це забирає час та накладає додаткові фінансові витрати. Крім того, не для кожної теми існують спеціальні спільноти, які б могли гарно викласти необхідну нам інформацію та відповісти на конкретні запитання і як наслідок, потенційно дуже цікавий ідеї з великим потенціалом може закінчитися так і не почавшись.

Для виявлення основних проблем та визначеннях напрямків для створення якісного продукту, було проведено наступний аналіз аналогів сервісів, які використовують дані із соціальних мереж для аналізу контенту, існуючих на ринку.

Абсолютних аналогів даної системи немає. Одними з часткових аналогів є «Citibeats», «Social Market Analytics» та «Dataminr», але ці сервіси надають лише частковий функціонал.

Citibeats – це платформа соціальних даних, яка допомагає приватним і державним організаціям зрозуміти, що важливо для людей у масштабі, в режимі реального часу, спрямовуючи осіб, які приймають рішення, вжити найбільш своєчасних і необхідних дій (див. рис. 1.1) [4].

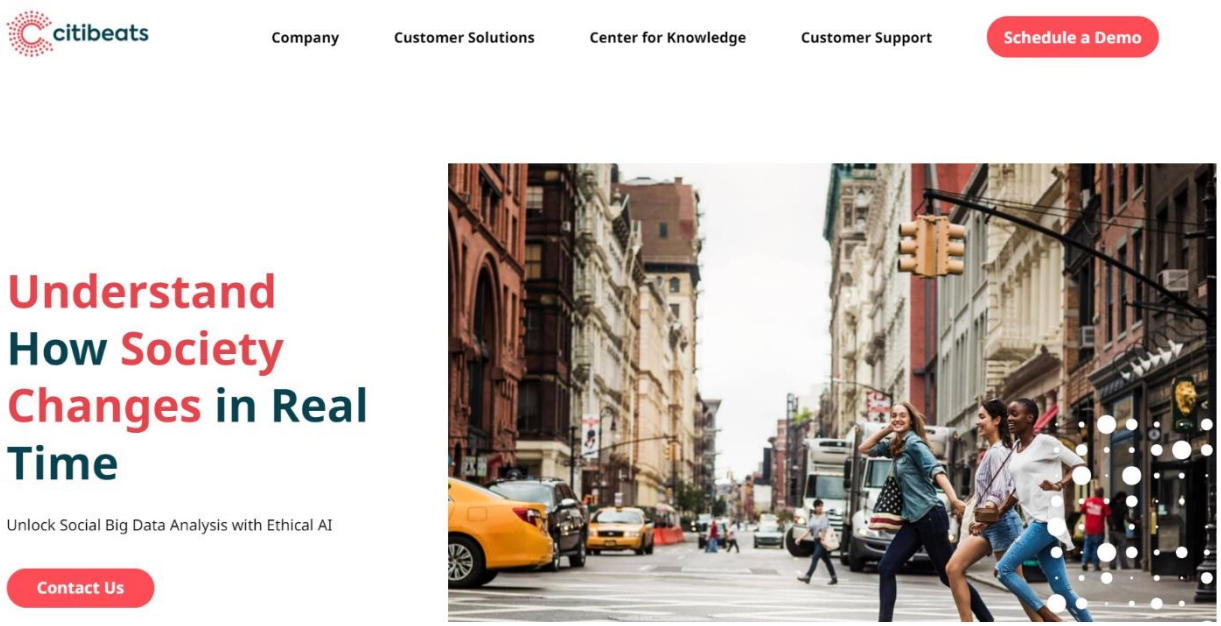


Рисунок 1.1 – Веб-сайт «Citibeats.com»

Алгоритм Citibeats аналізує великі обсяги неструктурованих текстових даних та інших типів файлів з текстом, який можна витягти, щоб визначити соціальні тенденції, думки та проблеми в реальному часі. Ця інформація являє собою цінну корисну інформацію для урядів, багатосторонніх організацій і приватних компаній, на основі яких вони можуть діяти.

У результаті основний функціонал Citibeats складається з:

– сповіщення в режимі реального часу: аномалії даних, відповідні кластери та моніторинг кластерів;

– провідних показників: соціальні сигнали, які повідомляють, що зміниться, перш ніж користувач побачить це в традиційних методах опитування та в ЗМІ.

Основним недоліком Citibeats є функціонал, який орієнтований лише на побудуваннях сповіщень, що можна замінити багатьма відкритими інструментами (наприклад, Grafana та Prometheus). Тобто поточний інструмент надає можливість налаштовувати сповіщення за заданими патернами без глибокого налізу подій, що може негативно сказатись на реагуванні на події.

Social Market Analytics (SMA) об’єднує наміри професійних інвесторів, виражені в неструктурованих даних [5]. Ці дані забезпечують прогнозні показники волатильності ринку та рівнів настроїв, щоб отримати перевагу для прогнозних стратегій спрямованого руху (див. рис. 1.2).

Рисунок 1.2 – Веб-сайт «Socialmarketanalytics.com»

SMA надає послуги готових API, які мають агрегувати дані за запитом:

– SMA S-Factor API Feed об’єднує твіти та порівнює з базовим рівнем, щоб отримати 15 показників щохвилини. Знімки можуть бути отримані з різними частотами на основі 24-годинного огляду назад із 20D базовою лінією із загасанням;

– SMA Activity API Feed ізолює та об’єднує інформацію щохвилини та дозволяє клієнтам створювати власні базові лінії та показники за допомогою необроблених даних;

– Short Squeeze в режимі реального часу повідомляє клієнтів про цінні папери з аномальною популярністю;

– SMA Tweet Level API Feed надає показники SMA на рівні твітів за допомогою оцінювання настроїв обробки природної мови та алгоритму рейтингу облікового запису SMA. Дані доставляються через API через 300 мілісекунд після отримання твіту від Twitter або StockTwits.

Основним недоліком системи SMA – є вузька тематика аналізу вхідних даних, а саме ринок дорогоцінних бумаг та брокерські теми. Отже сервіс не може масштабуватися під запити ринку та отримувати повідомлення про інші події.

Dataminr – це рішення для виявлення інцидентів, яке допомагає підприємствам аналізувати та пом’якшувати глобальні події та ризики з великим впливом [6]. Він використовує технологію штучного інтелекту (AI) для обробки даних у різних форматах, таких як текст, зображення, відео, аудіо та загальнодоступні дані датчиків IoT (див. рис. 1.3).

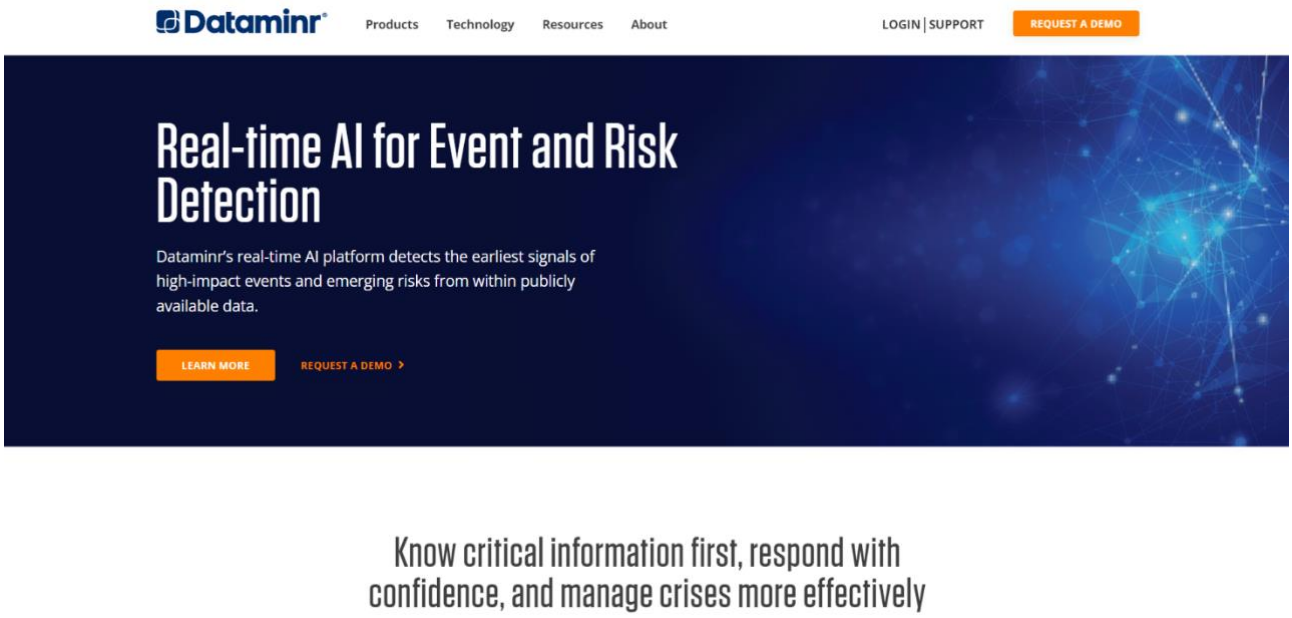


Рисунок 1.3 – Веб-сайт «Dataminr.com»

Dataminr перетворює потік Twitter та інші загальнодоступні набори даних у дійові сигнали, відкриваючи інформацію в режимі реального часу. За допомогою своїх алгоритмів він забезпечує сигнали для останніх новин, подій у реальному світі та нових тенденцій.

Dataminr є найсерйознішим конкурентом, але також має за собою мету лише нотифікувати про нові або аномальні події у світі. Отже, ресурс налаштовується за заданими патернами та реагує на публікації користувачів у Twitter. У той час, сервіс не надає можливість отримати короткий аналіз таких публікацій (чи достовірне джерело, чи ні). Слід зазначити, що у разі використовування Dataminr необхідно самостійно обирати вподобані сторінки. Таким чином велика кількість інформації, яка опублікована на сторінках, які не були задані для створення стрічки повідомлень, буде загублена. Саме втрата авторів-початківців, з малою кількістю підписників може стати критичною при пошуку інформації. Такі автори стають територіально-локальними експертами і лише через місяці і роки можуть попасти до стрічки більшості користувачів.

1.2 Виявлення проблем

При пошуку нових матеріалів для наукових конференцій, публікації новин, оформлення ділових угод або при фінансових інвестиції дуже важливо розуміти, чи є знайдені матеріали достовірними та чи можна на них покластися. Є велика ймовірність, що знайдена інформація може бути фейковою, або містити так званий скам (матеріали розміщені зловмисниками, яка може нашкодити шукачу або іншим особам), що може не дуже добре сказатися на репутації користувачів після публікації. Великі компанії, які мають штат та капітал мають можливість витрачати гроші на послуги фінансових та бізнес аналітиків. Із допомогою таких спеціалістів інформація може бути обміркована, проаналізована та представлена у зручному для ознайомлення вигляді. Крім того, така команда побудує необхідні документи, які будуть мати у собі всі демонстраційні матеріали зі знайденими

першоджерелами. Із допомогою таких дій будь-яка інформація стає зручною для вивчення, а як результат, в таку інформацію можна фінансувати і поширювати. Саме через це, проаналізована та оформлена інформація є дуже дорогим ресурсом, яким торгують у багатьох сферах: фінансові чутки, ринок ставок, аналітика, інформаційні технології, тощо.

Як результат, усі такі інвестиції у фінансових аналітиків та бізнес аналітиків допомагають зберегти значні кошти та час. Завдяки вдалій роботі аналітиків, компанії отримують інформацію про реальні проблеми, потреби та шляхи їх вирішення. Як результат, ринок отримує необхідні інструменти, а компанії – фінансові привілеї.

На жаль, не кожна звичайна людина має можливості для таких дорогих бізнес інструментів. Як результат, звичайній людині, яка, наприклад хоче увійти у світ інформаційних технологій або фінансових інвестицій, необхідно відвідувати спеціальні конференції, на яких можна відслідковувати останні тренди та робити аналіз тих сфер, які цікавлять запитуючого. Цей метод є дуже ефективним для аналізу будь-якої сфери, але через це витрачається найбільш дорогий ресурс – час. Цей ресурс – це одна з основних проблем звичайних людей. Відтак, відсутність часу знищує велику кількість потенційно перспективних угод, як результат, їх місця займають інші. Саме час та швидкість обробки інформації та якість цієї інформації впливають на потенціальний успіх комерційних операцій та розповсюдження фейків. Як відомо, хто володіє інформацією, той володіє світом.

Зі сторони аналізу проектів проблема полягає у великому об'ємі необхідної інформації та великій кількості джерел, які зберігають та публікують цю інформацію. Крім того, є проблема у необхідності швидкого та зручного пошуку та аналізу. Зберігання такої інформації у паперовому виді вимагає великого простору та значних ресурсів і повністю позбавляє можливості швидкого доступу. Іншою проблемою є те, що інформація поділяється на окремі різні формати: текстову, відео, аудіо та фото. За необхідністю отримання необхідної інформації про нові події, у яких зацікавлений користувач, довелося б працювати з декількома масивами різнотипної інформації [1]. Вочевидь, що для аналізу інформації про

вподобані теми необхідна інформаційна платформа, яка буде аналізувати та зберігати велику кількість даних та надавати зручний та швидкий доступ до них. Для спрощення роботи користувача і надання потрібних функції, система повинна буде використовувати базу даних і надавати зручне відображення даних за заданими темами та мати фільтрацію і сортування потрібної інформації.

1.3 Постановка задачі

Після аналізу предметної області та виявлення проблем в ній основною задачею поточного проекту стає вивчення методів обробки потокової інформації на базі соціальної мережі Twitter. Для цього необхідно проаналізувати сучасні існуючі методи обробки потокової інформації, та можливості швидкої обробки текстової інформації. Крім того, необхідно проаналізувати доступні API найвідоміших соціальних мереж, їх можливість до інтеграції до сучасних інструментів обробки інформації. Не менш важливим етапом аналізу є аналіз даних, що надходять із соціальної мережі.

На базі отриманих знань в якості експерименту необхідно створити платформу даних, яка дозволить інтегруватися із соціальними мережами, отримувати дані і паралельно збагачувати інформацію додатковими ознаками (ключовими виразами та настроєм тексту). У інформаційній платформі має бути побудована зручна система фільтрація і пошуку необхідних елементів за такими основними характеристиками, як дата публікації та надійність джерела інформації. Однією з найважливіших частин у системі глибокий аналіз інформації, яку вдалося отримати із соціальних мереж. Ці дані будуть проходити додаткове збагачення, яке дозволить виділити настрій та ключові слова текстових елементів.

У результаті експерименту необхідно перевірити можливість обробки даних, обрати методи обробки потокової інформації, можливість інтегрувати соціальні мережі як джерела інформації. Окрім того, слід перевірити формат даних, які можна отримати із соціальних мереж, їх зміст та можливість використання для

аналізу. Повним результатом експерименту стане відповідь на питання, чи можливо у режимі реального часу виявляти дійсно достовірну інформацію, збагачувати її та прибирати фейкові дані.

2 АНАЛІЗ ІСНУЮЧИХ СОЦІАЛЬНИХ МЕРЕЖ

2.1 Загальний огляд соціальних мереж

Соціальні мережі дуже впливають на життя сучасних людей. Сьогодні з 100 найбільш відвідуваних сайтів у світі 20 – це соціальні мережі. Більше 80% компаній по всьому світу використовують дані з соціальних мереж у своїй повсякденній роботі. Близько 75% людей вважають інформацію з соціальних мереж достовірною. Соціальні мережі стали невід'ємним центром сучасного світу.

На сьогоднішній день соціальні мережі по суті є найбільшою базою даних з найрізноманітнішою та різнотипною інформацією про сотні мільйонів людей по всьому світу. Слід відмітити, що дані у таких мережах є дуже прозорими, гарно організованими та структурованими. Найцікавішим є той факт, що такі мережі містять у собі не тільки персональну інформацію, але й вподобання, відвідані місця, тощо. Саме такі дані дозволяють робити дуже глибокий аналіз людини та створювати її цифровий паспорт.

Для того, щоб зрозуміти важливість соціальних мереж як джерела для аналізу інформації необхідно проаналізувати статистичні дані, які провели видатні європейські та американські видання. Майже 50% населення земної кулі користується соціальними мережами. Це більше 3-х мільярдів користувачів по всьому світу. За інформацією Emarketer у 2019 90,4% мілленіалов, 77,5% покоління X і 48,2% бебі-бумерів були активними користувачами соціальних мереж. GlobalWebIndex пише, що 54% браузерів використовують соціальні медіа для дослідження продуктів. Кожна людина витрачає в середньому 2 години і 22 хвилини на соціальні мережі і обмін повідомленнями. Hootsuite підводить підсумки, що 321 мільйон нових людей приєдналися до соціальних мереж в 2019 році, загальна кількість користувачів соціальних мереж збільшилася з 3,48 мільярда до 3,8 мільярда (зростання на 9%) в 2020 році. Світовий економічний форум зафіксував, що мілленіали заходять в соціальні мережі в середньому на дві години

і 38 хвилин щодня, в той час як покоління Z бувають в соцмережах протягом двох годин і 55 хвилин.

Отже, вибір соцмереж як джерела інформації є абсолютно доцільним. Перш за все, слід зауважити, що вибір соціальної мережі, як джерела інформації, базується на складності аналізу контенту. Такі популярні мережі як Tik-Tok, YouTube, Musically та інші мають у своїй основі відео-контент. Такий вид інформації є дуже затратним для обробки для вилучення інформації необхідні розвинути нейронні мережі, які б чітку розпізнавали голосову інформацію та трансформували її у текстову. Для того, щоб уникнути додаткових проблем та витрат на етапі експериментів було прийнято рішення розглядати лише мережі, у основі яких лежить текстовий контент, або частка такого контенту є більша за 50%. У наш час особливо виділяються три соціальні мережі: Facebook, Instagram та Twitter. Перші дві належать компанії Meta (у минулому Facebook) та представляють собою цілком різнопланові мережі.

На планеті налічується 7,6 мільярда людей, і 2,5 мільярда з них користуються Facebook принаймні раз на місяць. Оскільки майже третина населення світу підключено до однієї соціальної мережі, варто запитати: хто не є цільовим ринком Facebook?

Сказати, що кожен є його цільовим ринком – це не сказати нічого. Станом на кінець 2019 року Facebook налічував 248 мільйонів активних користувачів щомісяця в США та Канаді. Близько 68% усіх дорослих людей США мають обліковий запис. Існують невеликі відмінності, якщо вникати в демографічні показники.

Наприклад, 74% дорослих жінок є користувачами Facebook, у порівнянні з лише 62% дорослих чоловіків. І платформа, як і всі соціальні мережі, більш популярна серед молодих людей. Приблизно 80% людей віком від 18 до 49 років мають обліковий запис. Але літні американці все ще представлені. У віковій групі від 50 до 64 років 65% користуються Facebook, а також 41% людей у віці 65 років і старше.

Facebook може стати ідеальною платформою соціальних медіа для маркетингу малого бізнесу, якщо вміст додає справжню цінність для спільноти. Принаймні 70% ваших дописів мають стосуватися бізнес-порад, останніх подій, місцевих новин або питань опитування.

Близько 24% дорослих американців стверджують, що мають обліковий запис у Twitter. Чоловіки та жінки представляють користувачів Twitter майже порівну. Як і у Facebook, користувачі переважають молодше покоління, особливо серед 18-24-річні. У цій віковій групі 45% повідомляють, що вони використовують Twitter. Це для порівняння: 33% для вікової групи від 25 до 29 років, 27% для вікової групи від 30 до 49 років і 19% для вікової групи від 50 до 64 років. Випускники середньої школи рідше користуються Twitter, ніж випускники коледжу. Його користувачі також, як правило, більш заможні та живуть у місті.

Незважаючи на те, що Twitter має меншу глобальну базу користувачів, ніж Facebook та Instagram, він залишається чудовим інструментом для маркетингу продуктів, бізнес-чуток та новин. Триста тридцять мільйонів користувачів можна націлити за допомогою хештегів для створення цільових публікацій.

Здається, молоді люди не можуть вижити без Instagram. Соціальна мережа для обміну фотографіями та відео особливо популярна серед 18-24-річних. У цій віковій групі 71% кажуть, що мають обліковий запис в Instagram. Використання значно зменшується для літніх американців. У віковій групі від 25 до 29 років лише 54% мають обліковий запис. Лише 40% американців у віковій групі 30-49 років користуються Instagram, а для тих, кому від 50 до 64 років, цей показник становить 21%.

Отже, аналізуючи статистичні дані, слід виділити дві соціальні мережі, які є дуже розповсюдженими та підходять для аналізу контенту – це Facebook та Twitter. Для того, щоб остаточно обрати початкове джерело інформації, необхідно проаналізувати прикладний програмний інтерфейс, який надають соціальні мережі.

2.2 Аналіз API соціальних мереж

Перш за все, було прийнято рішення проаналізувати, які прикладні програмні інтерфейси надають компанії для розробників. Першим набором інструментів, який був досліджено, став Developer kit від компанії Meta (володіє соціальною мережею Facebook та Instagram).

В якості ресурсу для аналізу інформації були обрані текстові пости у мережі Facebook. Вони публікуються найчастіше та сприяють легшому аналізу змісту. Нажаль, фільтрації та пошуку публікацій за заданими темами або позначками хеш-тегів були заборонені компаніє Facebook. На сьогоднішній день існує можливість підключення та моніторинг конкретних сторінок (див. рис. 2.1).

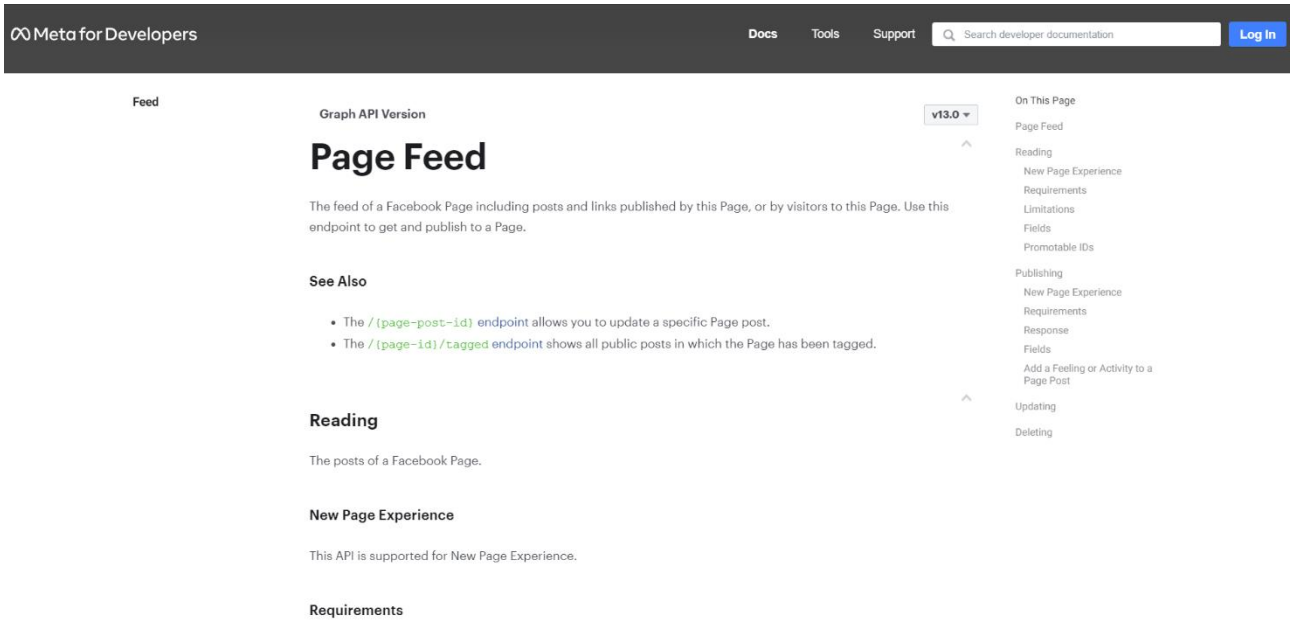


Рисунок 2.1 – Портал розробників Meta.

Доступний функціонал не дозволяє у повній мірі отримувати велику кількість інформації для аналізу. У режимі, наданому Facebook, публікації будуть отримуватись не у потоковому форматі, а у вигляді повідомлень. Це означає, що, наприклад, для отримання та аналізу двадцяти повідомлень з різних джерел, буде необхідно створити двадцять запитів на отримання інформації для конкретних

користувачів Facebook та об'єднати у один потік даних. Це стає дуже незручним та важким для аналізу, крім того, єдиним джерелом інформації стають заздалегідь підготовлені акаунти.

Результатом аналізу API для соціальної мережі Facebook є відсутність необхідних інструментів, які б підходили для комплексного аналізу тематичних подій.

Наступним набором інструментів є Twitter Developer Platform та її новітній інструмент Twitter API v2. Цей набір прикладних програмних інтерфейсів дозволяє отримувати безмежний доступ до публікацій у соціальній мережі – до твітів (див. рис. 2.2).

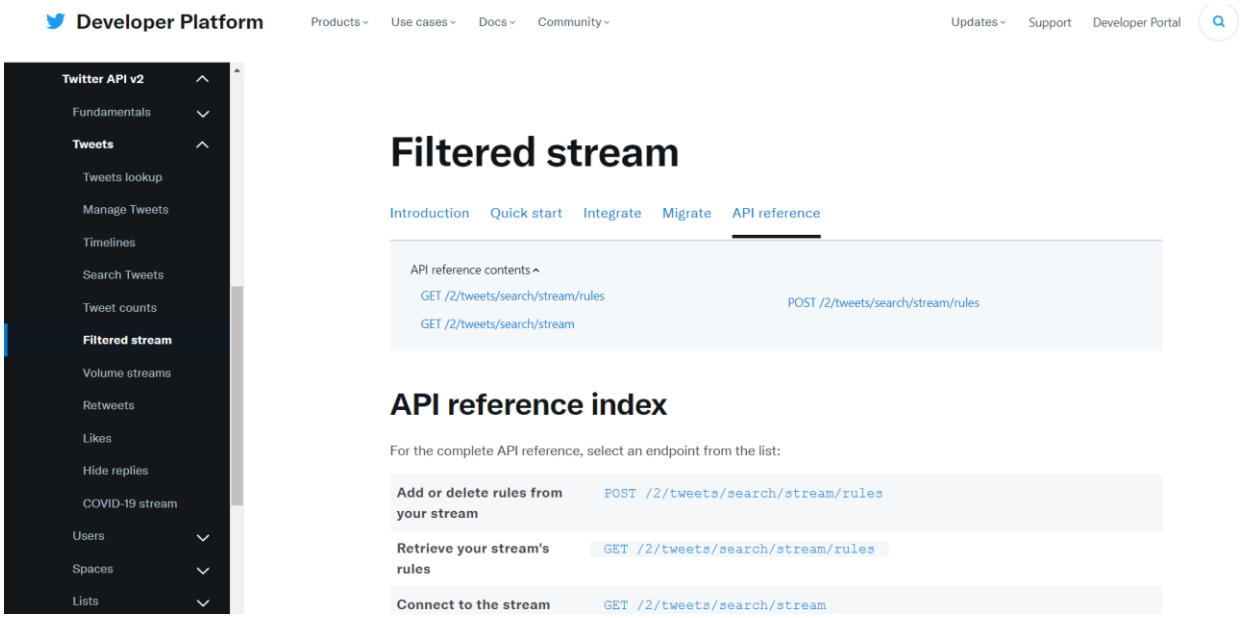


Рисунок 2.2 – Портал розробників Twitter.

Слід виділити два різновиди запитів до Twitter: історичні та потокові. Історичний запит дозволяє за заданими правилами, (правила запиту із конкретною тематикою та хештегами) із заданим періодом часу, отримати опубліковані твіти. Другий вид запитів – стрімінговий, він дозволяє за заданими правилами отримувати твіти у режимі реального часу.

Кінцева точка API відфільтрованого потоку дозволяє фільтрувати потік загальнодоступних твітів у реальному часі. Функціональні можливості включають

кілька кінцевих точок (endpoint), які дозволяють створювати правила та керувати ними, а також застосовувати ці правила для фільтрації потоку твітів у реальному часі, які повертатимуть відповідні загальнодоступні твіти. Ця група кінцевих точок дозволяє користувачам прослуховувати конкретні теми та події в режимі реального часу, стежити за публікаціями, розуміти, як розвиваються тенденції в режимі реального часу, і багато іншого.

У поточному інструменті є можливість використовувати кінцеву точку правил REST для додавання та видалення правил до постійного потокового з'єднання без необхідності від'єднання. Ці правила можна створити за допомогою операторів, які відповідають атрибутам твіту, таким як ключові слова повідомлення, хештеги та URL-адреси. Оператори та умови правил можна комбінувати з булевою логікою та дужками, щоб допомогти уточнити поведінку фільтра.

Після додавання правил можна встановити потокове з'єднання, яке почне доставляти об'єкти Tweet у форматі JSON через постійне потокове з'єднання HTTP. Саме такий функціонал необхідний для експериментів щодо вивчення шляхів динамічного аналізу потокових даних.

Отже, підсумовуючи приведену вище інформацію, можна відмітити, що Twitter API v2 є абсолютно новим релізом інструменту взаємодії із Twitter. Нещодавно компанія вирішила випустити оновлену версію, яка розширює базові можливості за допомогою надання великонавантажених потокових даних із гнучкою системою додавання правил, що стане в нагоді при проведенні експериментів.

2.3 Структура потокових даних з Twitter

Після вибору базового джерела інформації, яке буде використовуватись у проведенні експериментів, необхідно проаналізувати отримані дані, їй зміст та шляхи використання.

Новітній Twitter API v2 являє собою набір кінцевих точок, які повертають відповідь у форматі JSON. Цей документ складається з основних елементів:

- ідентифікаційного коду користувача;
- контекст анотації;
- набір сутностей;
- гео-дані;
- мова твіта;
- публічні метрики;
- джерело;
- вихідний текст.

Такий набір даних дозволяє провести аналіз та отримати необхідну інформацію для розподілу інформації за її валідністю. Для цього будуть підібрані три межі, які розділяють вхідний потік даних на 3 групи: інформація з так званою експертною думкою, інформація валідна для вживання та фейкова інформація

Для оцінювання якості інформації було прийнято використовувати публічні метрики та метрики акаунту-джерела. Twitter відразу надає доступ до метрик, які включають у себе декілька лічильників: лічильник підписників, лічильник сторінок, що відстежуються, кількість опублікованих твітів та перераховані твіти. Саме числа у лічильниках будуть основою для поділення даних.

Таким чином, була проаналізована модель даних, її складова та був обраний шлях аналізу вхідного потоку даних.

3 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Для створення системи, що виконує необхідні, поставлені задачі проводиться моделювання системи та її окремих компонентів, з використанням UML [5, с. 25] діаграм. У початковому аналізі було визначено основну структуру та поведінку системи взаємодію з нею з точки зору користувача.

Метою атестаційної роботи є вивчення сучасних методів обробки потокових даних та проведення практичних експериментів з реалізацією робочої версії платформи даних, яка використовує в якості джерела соціальну мережу Twitter. Платформа даних та сервіси повинні виконувати основні функції, які задані аналізом предметної області. Крім того, платформа даних повинна бути побудована на базі Kappa архітектурі, а сервіси повинні мати мікросервісну архітектуру та мати можливість масштабуватися.

3.1 Основний функціонал системи

Дана інформаційна система буде мати наступний базовий функціонал:

− зберігання та отримання інформації з соціальної мережі Twitter про твіти та акаунти;

− пошук та фільтрація інформації про твіти та акаунти серед загальної кількості;

− аналіз вхідних даних та розбиття їх на групи за рівнем довіри до джерела інформації;

− перевірка вхідних даних за настроєм вмісту;

− архівних аналіз даних, отриманих з мережі Twitter.

3.2 Допущення та залежності

Для коректної роботи усіх частин проекту наведений список допущень:

− користувачі системи мають пристрій з доступом до Інтернету;

− користувачі мають встановлений інтернет браузер.

Також існують наступні залежності:

− додаток буде орієнтуватися на новітні версії браузерів (Internet Explorer 9+ Microsoft Edge, Chrome 75+, Safari 4.0+).

3.3. Середа оточення

Так як предметною областю даного веб-орієнтованого сервісу не передбачено використання певної СУБД для зберігання і роботи з інформацією системи була обрана Cassandra від Apache. Для зберігання секретних даних (строки підключення, паролі, тощо) буде використовуватися сервіс Azure KeyVault. В якості інструменту виявлення сервісів (Service Discovery) буде використовуватися Consul. Для виділення ключових даних буде використовуватися Azure Recognition Service (Azure Text Analytics). Для отримання та аналізу даних буде використана соціальна мережа Twitter та акаунт розробника. Для асинхронної комунікації між платформою даних та сервісами буде використовуватися Apache Kafka та конкретна реалізація під Azure – Azure Event Hub.

Дана платформа та веб-орієнтовані сервіс будуть побудовані із використанням платформи Java, мова програмування Scala версії 2.13. Сервіс буде розроблений у середовищі Intellij Idea 2021.

Користувачі повинні мати комп'ютер під управлінням ОС Microsoft Windows 7 і вище або MacOS OS X 10.9 Mavericks та вище, браузер (Internet Explorer 9+, Microsoft Edge, Chrome 75+, Safari 4.0+).

4 ОГЛЯД НАУКОВОЇ І ПАТЕНТНОЇ ЛІТЕРАТУРИ

4.1 Огляд наукових конференцій

Аналізуючи задану предметну область, її проблеми та методи вирішення, було прийнято рішення звернутися до вітчизняних розробок та вивчити наукові публікації, які були створені українськими колегами.

Перш за все було прийнято рішення проаналізувати найсвіжіші наукові конференції, в яких були представлені розробки українських вишів. На XLVIII науково-технічній конференції факультету комп'ютерних систем і автоматики, яка проходила з 11 по 15 травня 2019 року, Вінницький національний технічний університет представив своє бачення обробки потокової інформації [7]. У науковій публікації яка має назву «Сучасні методи обробки потокових даних» автори В.В. Стецюк та Т.В. Грищук описали основи потокової обробки інформації. Автори зауважили, що роль даних, що генеруються кожну хвилину, зростає з кожним днем. Якщо раніше такі дані генерувалися у сферах інформаційних технологій, авіації, бізнесі, то тепер потокові дані слідують за користувачем всюди: соціальні мережі, ігри, новини, стрімінгові платформи.

Для вирішення проблем обробки потокових даних колеги запропонували використовувати інструменти відкритих джерел (open source) з ліцензіями сімейства Apache: Amazon Kinesis, Apache Spark Streaming, Apache Storm, Apache Samza. Автори описують концепції Apache Samza, яка полягає в обробці повідомлень по мірі їх отримання, та Apache Spark із використанням Structured Streaming, де концепція полягає у розбитті вхідних даних на пакети (micro-batching).

Науковці також виділили, що на момент травня 2019 року Apache Samza та Apache Spark не покривають 100% потреб користувачів у обробці потокових даних. Після аналізу публікації та пропозицій, до яких дійшли автори, були побудовані свої власні висновки щодо цієї теми.

Прикладні варіанти використання Apache Samza (use-cases) обумовлені перевагами та недоліками цієї Big Data системи. Зокрема, Samza зберігає стан додатків (stateful), використовуючи систему відхилення від контрольних точок, реалізовану як локальне сховище значень ключів. Це дозволяє Samza пропонувати гарантовану доставку хоча б один раз (at-least-once), але не забезпечує точного відновлення агрегованого стану, наприклад кількості у разі збою, оскільки дані можуть бути доставлені більше одного разу (див. рис. 4.1).

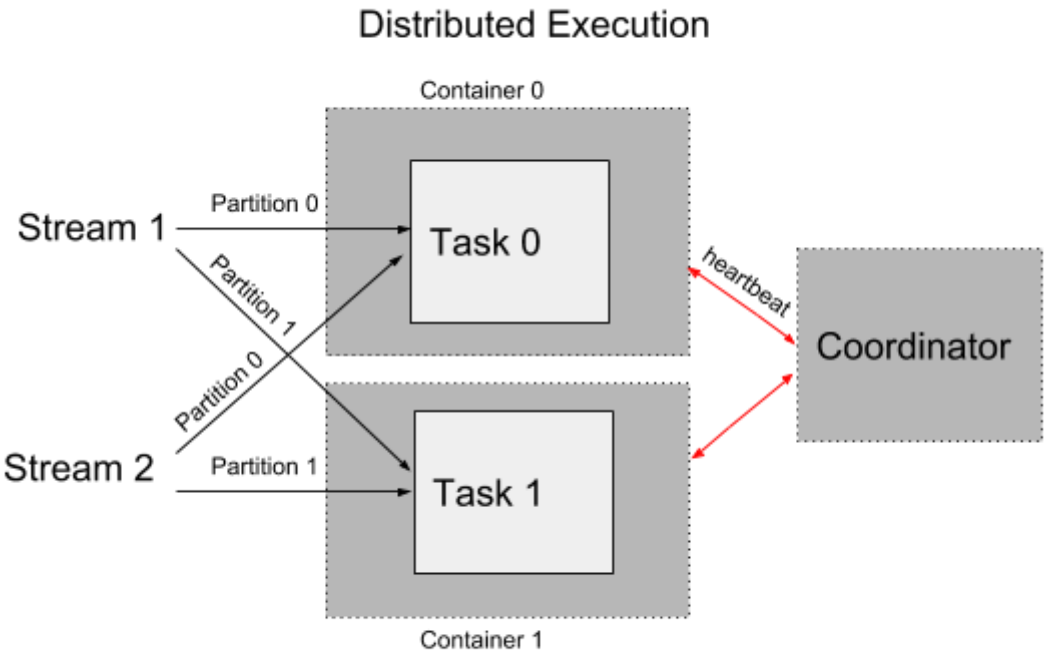


Рисунок 4.1 – Архітектура Apache Samza.

Замість того, щоб обробляти потокові дані по одному запису, Spark Streaming дискретизує потокові дані на крихітні мікропакети, що не перевищує секунди. Іншими словами, приймачі Spark Streaming приймають дані паралельно і буферують їх у пам’ять робочих вузлів Spark. Потім оптимізований за затримками двигун Spark виконує короткі завдання (десятки мілісекунд) для обробки пакетів і виведення результатів в інші системи. На відміну від традиційної моделі безперервного оператора, де обчислення статично розподілені на вузол, завдання

Spark призначаються динамічно на основі місцевості даних і доступних ресурсів. Це дозволяє як краще балансувати gfnнавантаження, так і швидше усунути помилки, як ми проілюструємо далі.

Крім того, кожен пакет даних є стійким розподіленим набором даних Spark Resilient Distributed Database (RDD), який є основною абстракцією відмовостійкого набору даних у Spark (див. рис. 4.2).

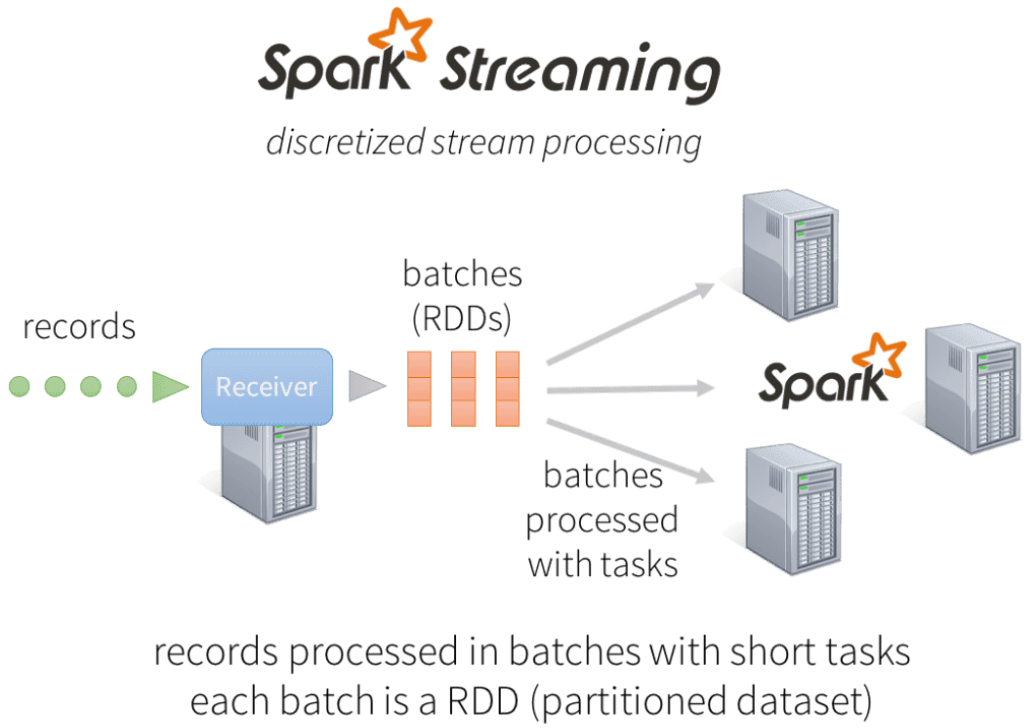


Рисунок 4.2 – Архітектура потокової обробки Apache Spark.

Потім ці RDD обробляються за допомогою таких операцій, як map, reduce, join тощо. Результат цих операцій повертається пакетами (micro-batching). Таким чином, це не обробка в режимі реального часу, а наближених до реального часу.

На момент 2019 року не був широко відомий ще один дуже потужний інструмент обробки потокових даних – Apache Flink.

4.2 Огляд наукових публікацій

Після вивчення вітчизняних наукових публікацій та конференцій було прийнято рішення проаналізувати досвід європейських колег. Для цього був обраний відомий у світі науковий портал ResearchGate. Основними параметру запиту були ключові слова: «distributed systems», «streaming data processing» та «big data» (див. рис. 4.3).

Рисунок 4.3 – Публікація на порталі ResearchGate.

Проаналізувавши декілька публікацій було прийнято рішення детальніше вивчити роботу македонських авторів Ніколети Танталакі, Ставроса Суравласа та Маноса Румеліотіса, яка має назву: «Огляд обробки потоків великих даних у реальному часі та методів планування» («A review on Big Data real-time stream processing and its scheduling techniques») [8]. У своїй роботі автори підіймають важливе питання проблеми обробки великих потокових даних. Вони описують різні семантики доставки, сильні і слабкі сторони кожної та рекомендації щодо їх використання.

Особливістю цієї роботи є детальний опис потрібностей бізнесу у обробці великої кількості різноманітних даних та представлення вже існуючих інструментів, які в тій чи іншій степені покривають ці потреби.

Слід відмітити, що автори розробили дуже детальний аналіз інструментів, які доступні для розробників, серед таких: Spark Structured Streaming, Apache Storm, Apache Samza та Flink.

Основне питання: який фреймворк обрати, щоб він міг покривати максимальну кількість кейсів та мав активну підтримку зі сторони спільноти залишається відкритим. Отже, найкраща відповідь, це те, що вибір залежить від потреб користувачів і бізнесу. Слід також враховувати майбутні міркування. Наприклад, у простих випадках Storm може здатися гарною ідеєю. Тим не менш, якщо розширені вимоги, що включають складну обробку подій, як-от агрегації та об’єднання, виникнуть пізніше, або також слід запровадити пакетно-орієнтовані завдання, перевагу слід віддавати розширеним потоковим структурам, таким як Spark Streaming або Flink. Зміни в існуючій інфраструктурі може призвести до величезних витрат часу та грошей.

Нарешті, необхідно на увазі, що розробка в декларативних системах набагато простіше, ніж у композиційних, оскільки користувачам надаються абстракції вищого рівня та мають на увазі DAG (directed acyclic graph) через своє кодування. Оптимізація може здійснюватися за допомогою системи. Тим не менш, у композиційних системах код повністю контролюється розробником. Якщо є потреба у швидкій і легкій реалізації, перевагу слід віддати таким системам, як Spark або Flink, але якщо потрібен повний контроль над графом додатків, то слід вибрати Storm або Samza. Як можна побачити, вибір інструментів залежить від багатьох факторів. Розуміння механізмів і характеристик доступних архітектур полегшує вибір або принаймні відфільтровує доступні варіанти. Оцінка незавершеної роботи може допомогти користувачам зробити найкращий вибір на основі їхніх потреб. Зміни в конфігурації, налаштування або доступної інфраструктури можуть змінити результати та призвести до серйозних покращень.

Планування завдань сильно впливає на продуктивність і відмовостійкість в системі потокової обробки.

Огляд наукових конференцій та публікацій дозволив достатньо близько познайомитися з методологією потокової обробки великих даних. Були проаналізовані роботи вітчизняних та європейських авторів, які містили у собі експериментальні блоки із конкретними метриками. Дуже корисними виявилися саме наукові конференції, де серед учасників були саме молоді та перспективні студенти. Саме такі молоді науковці запроваджували тренди на вивчення нових технологій та сприянню впровадження open source технологій у маси.

Накопичений досвід дозволяє перейти до наступного кроку – моделювання майбутніх експериментів. Саме моделювання є одним з найважливіших кроків у планування майбутніх експериментів, на якому необхідно проаналізувати сильні та слабкі сторони описаних вище інструментів та обрати ті, які максимально покриють потреби обраного домену.

5 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1. Вибір базової технології

Визначення Big Data зазвичай розшифровують досить просто – це величезний обсяг інформації, часто безсистемної, яка зберігається на будь-якому цифровому носії. Однак масив даних з приставкою «Великі» настільки великий, що звичними засобами структурування та аналітики обробити його неможливо. Тому під терміном «великі дані» розуміють ще і технології пошуку, обробки та застосування неструктурованою інформації в великих обсягах. У світі технологій «великих даних» існує три визначальні властивості, які можуть допомогти зрозуміти основні характеристики. Для цього необхідно відділити правило трьох «V»: volume, velocity, variety. Це ключове значення для розуміння того, як ми можемо вимірювати великі дані, і наскільки різняться «великі дані» від звичних даних (див. рис. 5.1).

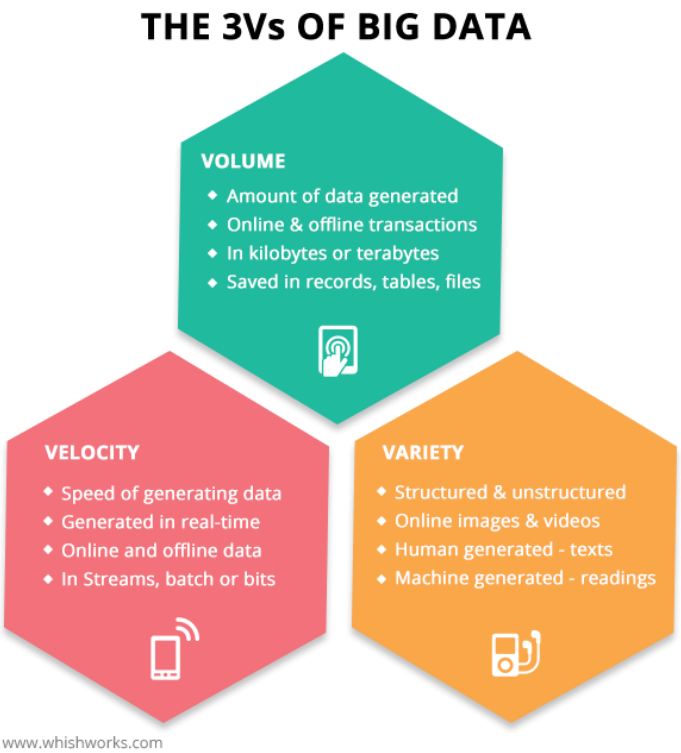


Рисунок 5.1 – Три «V» світу великих даних.

Обсяг (volume) – дані вимірюються по фізичній величині і займаному простору на цифровому носії. До «великих» відносять масиви понад 150 Гб на добу.

Швидкість, оновлення (velocity) – інформація регулярно оновлюється і для обробки в реальному часі необхідні інтелектуальні технології «великих даних».

Різноманітність (variety) – інформація в масивах може мати неоднорідні формати, бути структурованою частково, повністю і накопичуватися безсистемно. Наприклад, соціальні мережі використовують великі дані у вигляді текстів, відео, аудіо, фінансових транзакцій, картинок і іншого.

У сучасних системах розглядаються два додаткових фактори додаються до «правила трьох V»: variability та value.

Мінливість (variability) – потоки даних можуть мати піки і спади, сезонності, періодичність. Сплески неструктурованою інформації складні в управлінні, вимагає потужних технологій обробки.

Значення даних (value) – інформація може мати різну складність для сприйняття і переробки, що ускладнює роботу інтелектуальним системам. Наприклад, масив повідомлень з соцмереж - це один рівень даних, а транзакційні операції - інший. Завдання машин визначити ступінь важливості інформації, що надходить, щоб швидко структурувати.

Принцип роботи технології Big Data заснований на максимальному інформуванні користувача про який-небудь предмет або явище. Завдання такого ознайомлення з даними – допомогти зважити всі «за» і «проти», щоб прийняти вірне рішення. В інтелектуальних машинах на основі масиву інформації будується модель майбутнього, а далі імітуються різні варіанти і відслідковуються результати.

Принципи роботи з масивами даних включають три основні чинники:

− можливість розширення системи. Під нею розуміють зазвичай горизонтальну масштабованість носіїв інформації. Тобто зросли обсяги вхідних даних – збільшилися потужність і кількість серверів для їх зберігання;

− стійкість до відмови. Підвищувати кількість цифрових носіїв, інтелектуальних машин пропорційно обсягам даних можна до нескінченності. Але

це не означає, що частина машин не буде виходити з ладу, застарівати. Тому одним із чинників стабільної роботи з великими даними є відмовостійкість серверів;

− локалізація. Окремі масиви інформації зберігаються і обробляються в межах одного виділеного сервера, щоб економити час, ресурси, витрати на передачу даних.

Чим більше ми знаємо про конкретний предмет або явище, тим точніше осягаємо суть і можемо прогнозувати майбутнє. Знімаючи і обробляючи потоки даних з датчиків, інтернету, транзакційних операцій, компанії можуть досить точно передбачити попит на продукцію, а служби надзвичайних ситуацій запобігти техногенним катастрофам. Наведемо кілька прикладів поза сферою бізнесу і маркетингу, як використовуються технології великих даних:

− охорона здоров'я. Більше знань про хвороби, більше варіантів лікування, більше інформації про лікарські препарати – все це дозволяє боротися з такими хворобами, які 40-50 років тому вважалися невиліковними;

− попередження природних та техногенних катастроф. Максимально точний прогноз в цій сфері рятує тисячі життів людей. Завдання інтелектуальних машин зібрати і обробити безліч показників датчиків і на їх основі допомогти людям визначити дату і місце можливого катаклізму;

− правоохоронні органи. Великі дані використовуються для прогнозування сплеску криміналу в різних країнах і прийняття стримуючих заходів, там, де цього вимагає ситуація;

− стратегії розвитку бізнесу, маркетингові заходи, реклама засновані на аналізі та роботі з наявними даними. Великі масиви дозволяють «перелопатити» гігантські обсяги даних і відповідно максимально точно скорегувати напрямок розвитку бренду, продукту, послуги.

Наприклад, аукціон RTB в контекстній рекламі працюють з Big Data, що дозволяє ефективно рекламувати комерційні пропозиції виділеної цільової аудиторії, а не всім підряд.

Для бізнесу існує велика кількість переваг, серед яких є:

− створення проектів, які з високою ймовірністю стануть затребуваними у користувачів, покупців;

− вивчення і аналіз вимог клієнтів з існуючим сервісом компанії. На основі викладки коригується робота обслуговуючого персоналу;

− виявлення лояльності і незадоволеності клієнтської бази за рахунок аналізу різноманітної інформації з блогів, соцмереж та інших джерел;

− залучення і утримання цільової аудиторії завдяки аналітичній роботі з великими масивами інформації;

− технології використовують в прогнозуванні популярності продуктів, наприклад, за допомогою сервісу Google Trends.

Методики Big Data використовують всі великі компанії: IBM, Google, Facebook. Крім того, багато фінансових корпорацій також використовують ці технології: VISA, Master Card. Все частіше міністерства різних країн світу таке впроваджують технології «великих даних». Наприклад, в Німеччині скоротили видачу допомоги безробітним, вирахувавши, що частина громадян отримують їх без підстав. Так вдалося повернути в бюджет близько 15 млрд. євро.

У 2021 важливість розуміння і головне роботи з масивами інформації зросла в 4-5 разів у порівнянні з початком десятиліття. З масовістю прийшла інтеграція Big Data в сфери малого і середнього бізнесу, стартапи:

− хмарні сховища. Технології зберігання і роботи з даними в онлайн-просторі дозволяє вирішити масу проблем малого і середнього бізнесу: дешевше купити хмара, ніж утримувати дата-центр, персонал може працювати віддалено, не потрібен офіс;

− глибоке навчання, штучний інтелект. Аналітичні машини імітують людський мозок, тобто використовуються штучні нейронні мережі. Навчання відбувається самостійно на основі великих масивів інформації;

− dark data – збір і зберігання не оцифрованих даних про компанії, які не мають суттєвої ролі для розвитку бізнесу, однак вони потрібні в технічному і законодавчому планах;

− блокчейн. Спрощення інтернет-транзакцій, зниження витрат на проведення цих операцій;

− системи самообслуговування – з 2018 року впроваджуються спеціальні платформи для малого і середнього бізнесу, де можна самостійно зберігати і систематизувати дані.

5.2. Архітектура програмного забезпечення

При побудуванні платформи даних були розглянуті новітні архітектури та патерни. Одними з таких є Lambda та Kappa. При всіх перевагах Лямбда-архітектури, головним недоліком цього підходу до проектування Big Data систем вважається його складність через дублювання логіки обробки даних в холодному і гарячому шляхах. Тому в 2014 році була запропонована Каппа - альтернативна модель, яка споживає менше ресурсів, але відмінно підходить для обробки подій в режимі реального часу.

На відміну від лямбда, в Каппа-архітектурі потокові дані проходять по одному шляху. Всі дані приймаються як потік подій в розподіленому і відмовостійкість єдиному журналі - балці подій. Там події упорядковуються, і поточний стан події змінюється тільки при додаванні нового події. Аналогічно рівню прискорення лямбда-архітектури, вся обробка подій виконується у вхідному потоці і зберігається як уявлення в режимі реального часу. Якщо необхідно повторно обчислити весь набір даних, як на пакетному рівні в лямбда-архітектурі, потік відтворюється заново. Для своєчасного завершення обчислень використовується паралелізм.

Це дозволяє швидко і якісно обробляти інформацію. Крім того, слід зазначити, що при Каппа архітектурі уникає можливість створення дублікатів (див. рис 5.2).

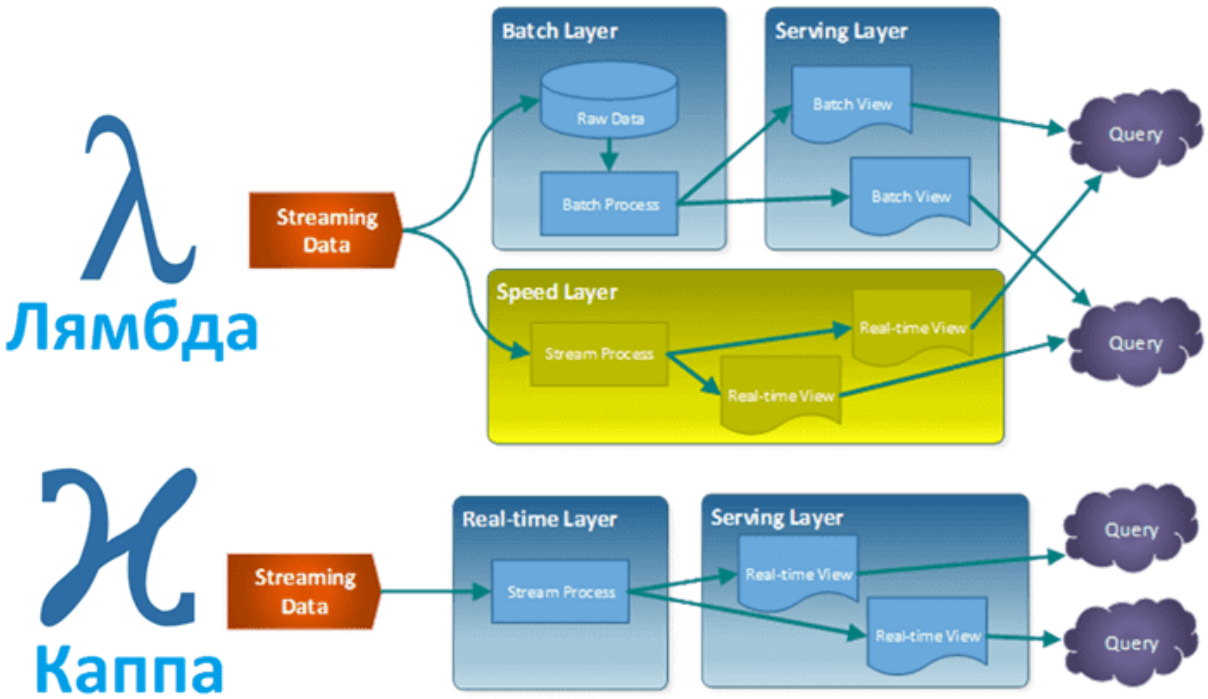


Рисунок 5.2 – Архітектури Lambda та Kappa.

Саме вибір Kappa архітектури надає наступні переваги:

– архітектуру Kappa можна використовувати для розробки систем даних, які навчаються в Інтернеті і тому не потребують пакетного рівня;

– повторна обробка потрібна лише при зміні коду;

– його можна розгорнути з фіксованою пам'яттю;

– його можна використовувати для горизонтально масштабованих систем;

– потрібно менше ресурсів, оскільки машинне навчання здійснюється в режимі реального часу.

В якості платформи обробки потокових даних був обраний Flink. Мовою розробки платформи даних було обрано Scala. Завдяки даному вибору розробка буде швидкою, а код вийде компактним і відмінно читаним. Крім того, має можливість дуже швидко й гнучко масштабуватись. У якості IDE була обрана

Intellij Idea, оскільки він повністю задовольняє мої потреби як розробника і є найкращим в своїй сфері.

У результаті була спроектована система, яка дозволить обробляти дані у режимі реального часу та зберігати результати у базі даних (див. рис. 5.3).

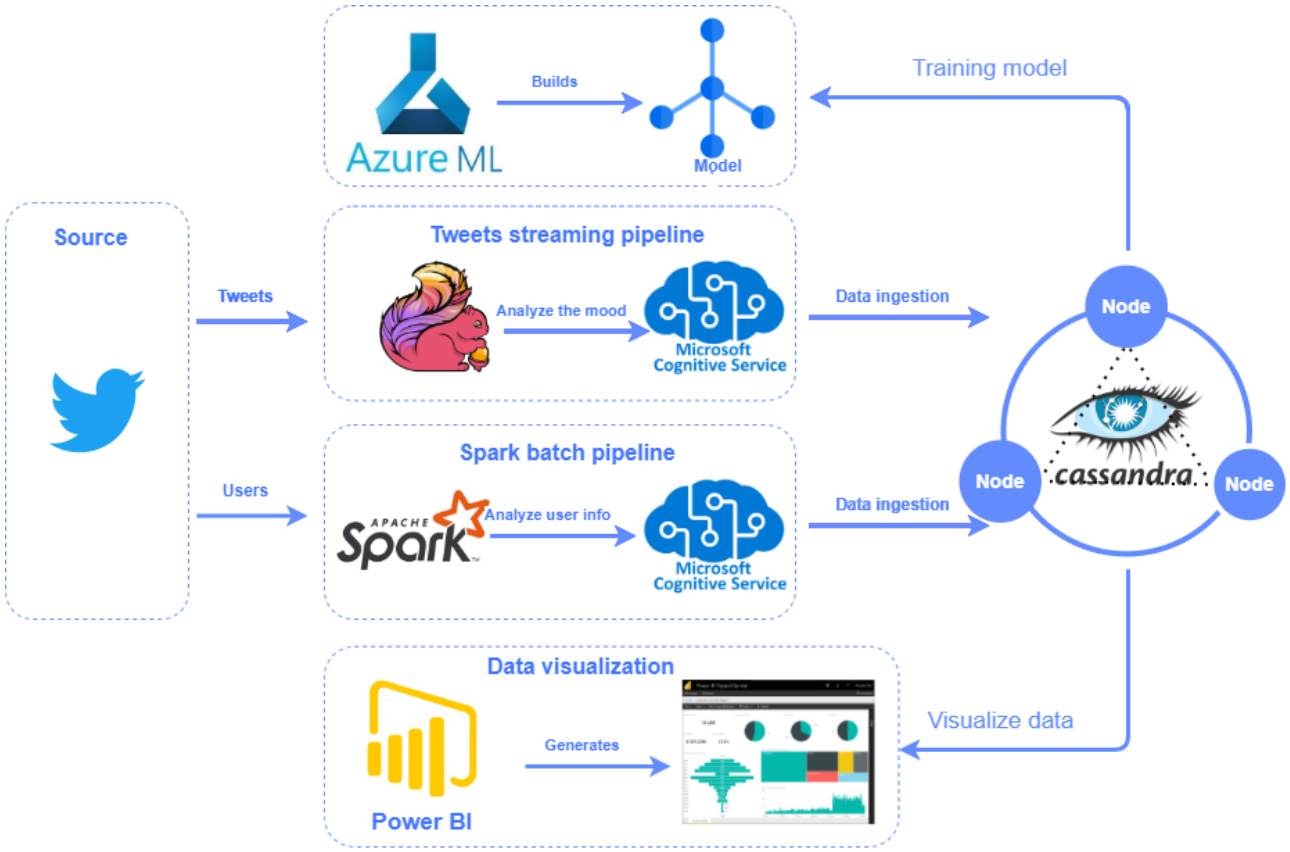


Рисунок 5.3 – Спроектована послідовність обробки даних.

На рисунку 5.3 можна побачити взаємодію між різними компонентами системи. Слід визначити, що джерелом даних усієї системи стала соціальна мережа Twitter [9]. Завдяки новітньому прикладному програмному інтерфейсу платформа даних має можливість як отримувати дані у режимі реального часу (потокові дані) так і будувати історичні запити (дані за запитом). Наступним елементом послідовності є так звані послідовності обробки даних. Для обробки основних повідомлень із мережі Твіттер (твітів) був обраний інструмент Flink. Для обробки інформації, пов’язаної із користув’ачами був обраний інструмент Spark.

У системі існує спеціальна послідовність дій, за якою виконується функція виявлення настрою тексту. У ній використовується провідні сервісі розпізнавання тексту Microsoft Cognitive Services [10, с. 212]. Сама послідовність складається з наступних пунктів:

− надсилання параметрів пошуку з сервіса у Microsoft Bing Spell Check для виправлення граматичних помилок;

− надсилання перевірених параметрів пошуку у Microsoft Text Analytics для виділення найбільш важливих слів;

− використання Microsoft Text Analytics для виявлення настрою тексту.

Після обробки інформації необхідно зберегти її у базу даних. В якості бази даних була обрана open-source версія бази Cassandra. З'єднання та запис у базу відбувається з допомогою протоколу TCP (Transmission Control Protocol, протокол керування передаванням). Детальні переваги Cassandra будуть наведені нижче.

Оброблені та підготовлені для використання дані також використовуються інструментом Azure Machine Learning. Цей інструмент дозволяє тренувати модель, яка дозволяє більш детально і чітко класифікувати вхідні повідомлення та сортувати рівень довіри до користувачів мережі Twitter.

В якості інструменту відображення даних був обраний Azure Power BI, який дозволяє будувати звіти. Ці звіти будуються із допомогою штучного інтелекту та із заданими фільтраціями, агрегаціями та функціями.

Для більш детального аналізу архітектури, побудуємо діаграму розгортання.

Діаграма розгортання – це діаграма, яка показує конфігурацію вузлів обробки часу виконання та компонентів, які на них працюють. Діаграми розгортання — це різновид структурної діаграми, яка використовується при моделюванні фізичних аспектів об'єктно-орієнтованої системи. Вони часто використовуються для моделювання статичного вигляду розгортання системи (топології апаратного забезпечення).

Така діаграма дозволяє побачити та проаналізувати, яку структуру має система, яким чином та якими протоколами користуються елементи всередині, та як елементи захищені. Існує багато видів нотацій для побудування діаграм

розгортання. У поточному дослідженні було прийнято рішення використовувати нотацію від Microsoft Azure (див. рис. 5.4).

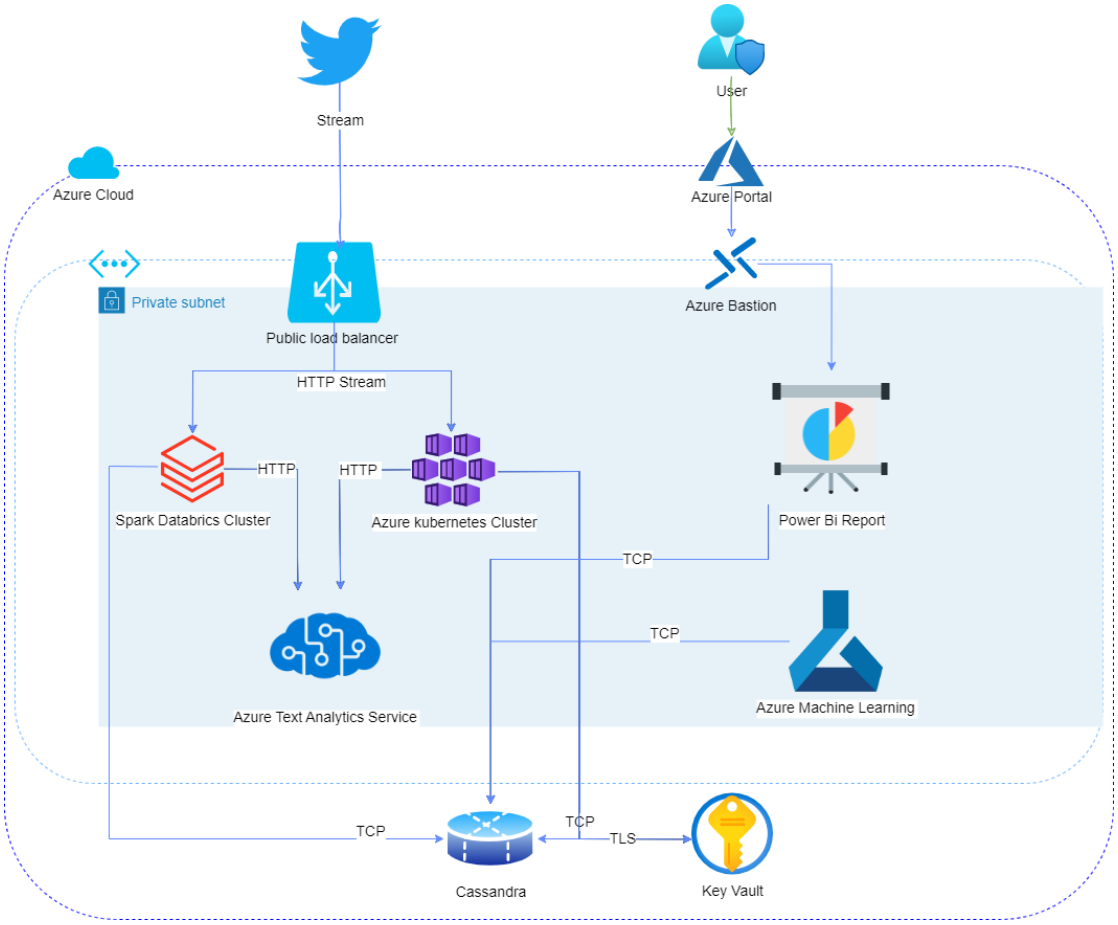


Рисунок 5.4 – Діаграма розгортання

Ця діаграма явно демонструє, з яких інструментів та сервісів складається поточна система, а також, які протоколи та види комунікацій використовуються.

5.3 Будування UML діаграми

Для того, щоб правильно спроектувати архітектуру для майбутньої системи необхідно побудувати діаграму послідовностей [11]. Діаграми послідовності UML – це діаграми взаємодії, які детально описують, як виконуються операції. Вони фіксують взаємодію між об’єктами в контексті співпраці. Діаграми послідовності – це час, і вони візуально показують порядок взаємодії, використовуючи вертикальну

вісь діаграми, щоб відобразити час, які повідомлення надсилаються і коли. (див. рис. 5.5).

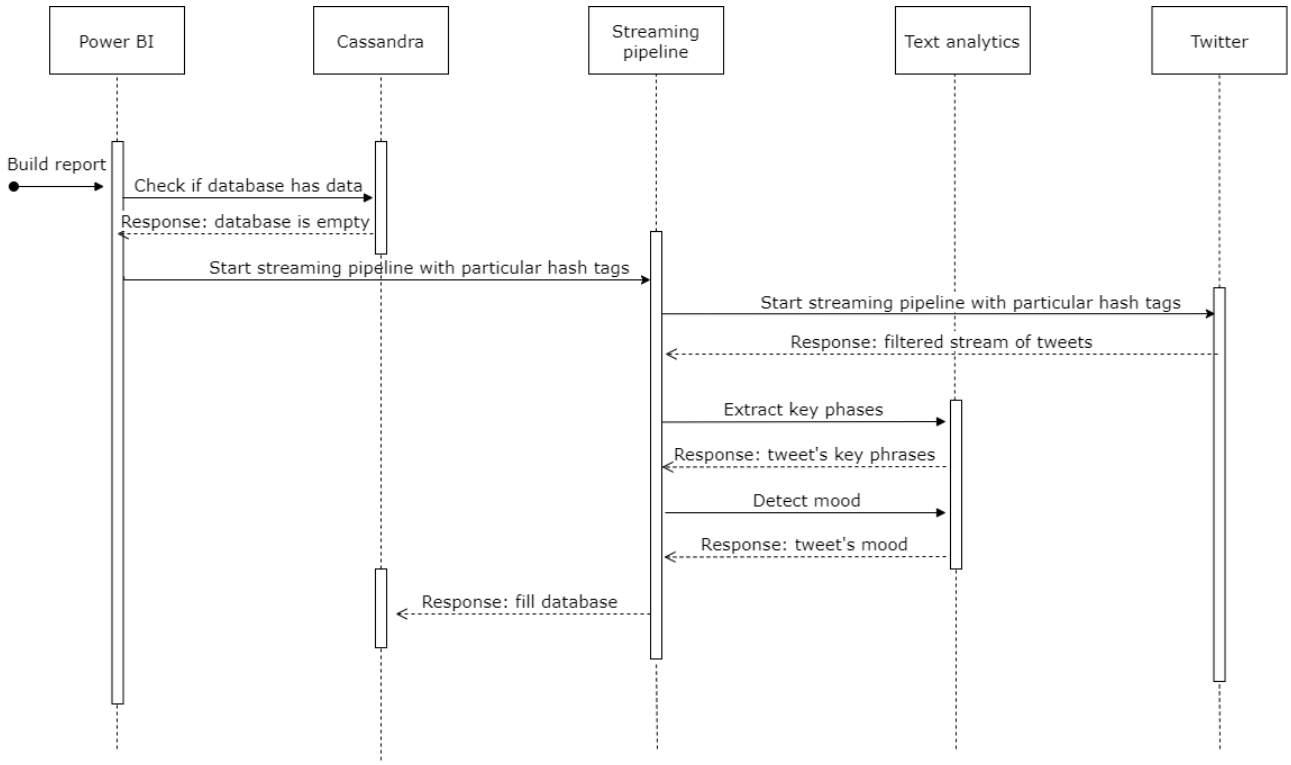


Рисунок 5.5 – Діаграма послідовностей.

Основними елементами у діаграмі є інструмент для отримання звіту, база даних, потоковий конвеєр, інструмент аналізу тексті та джерело – соціальна мережа Twitter.

5.4 Проектування структури зберігання даних

Apache Cassandra – це розподілена база даних з відкритим вихідним кодом або NoSQL, яка забезпечує безперервну доступність, величезний масштаб і розподіл даних у кількох центрах обробки даних і хмарних зонах доступності. Простіше кажучи, він забезпечує високонадійний механізм зберігання даних для додатків, які потребують величезного масштабу.

Моделювання даних – це процес, який використовується для аналізу, організації та розуміння вимог до даних для продукту чи послуги. Моделювання даних створює структуру, в якій будуть жити ваші дані. Воно визначає, як речі позначаються та організуються, а також визначає, як ваші дані можуть і будуть використовуватися. Процес моделювання даних схожий на проектування будинку. Ви починаєте з концептуальної моделі і додаєте деталі, щоб створити остаточний план.

Кінцевою метою моделювання та аналізу даних Cassandra є розробка повного, добре організованого та високопродуктивного кластера Cassandra. Сподіваємося, що дотримання п’яти наведених найкращих методів моделювання даних Cassandra допоможе досягти цієї мети:

П'ять найкращих методів використання Apache Cassandra;

- не потрібно використовувати Cassandra як реляційну базу даних;
- необхідно створювати модель на основі трьох цілей розподілу даних;
- правильно побудувати первинний ключ;
- будувати модель навколо запитів;
- провести тестування для забезпечення продуктивності.

Розподілені системи даних, такі як Cassandra, розподіляють вхідні дані на блоки, які називаються розділами. Cassandra групує дані в окремі розділи, хешуючи атрибут даних, який називається ключ розділу, і розподіляє ці розділи між вузлами в кластері.

Хорошою моделлю даних Cassandra є та, яка:

- рівномірно розподіляє дані між вузлами в кластері;
- установлює обмеження на розмір розділу;
- мінімізує кількість розділів, які повертає запит.

Для того, щоб до кінця розібратися, чим Cassandra відрізняється від будь якої реляційної бази даних необхідно виділити основні особливості при побудуванні таблиць.

Дизайн, побудований на запитах. Необхідно визначити, як планується отримання доступу до таблиць даних на початку процесу моделювання даних, а не в кінці.

Немає об’єднань або похідних таблиць. Таблиці не можуть бути об’єднані, тому, якщо є необхідність у даних з кількох таблиць, таблиці необхідно об’єднати в денормалізовану таблицю.

Денормалізація. Cassandra не підтримує об'єднання або похідні таблиці, тому денормалізація є ключовою практикою в дизайні таблиць Cassandra.

Проектування для оптимального зберігання. Для реляційних баз даних це зазвичай прозоро для дизайнера. Для Cassandra важливою метою дизайну є оптимізація того, як дані розподіляються по кластеру.

Сортування є проектним рішенням. у Cassandra сортування можна виконувати лише за стовпцями кластеризації, зазначеними в первинному ключі.

Щоб створити повну та високопродуктивну модель даних, слід дотримуватися методології моделювання великих даних для Apache Cassandra, яку можна підсумувати як:

− виявлення даних (DD). Це представлення високого рівня даних, які потрібні вашій програмі, і ідентифікує сутності (речі), атрибути сутностей та атрибути, які є ідентифікаторами. Це може бути ітеративним процесом як розробка;

− визначте шаблони доступу (AP). Визначте та перерахуйте запити, які ваша програма захоче виконати. Вам потрібно відповісти: які дані потрібно отримати разом, які критерії пошуку та які шаблони оновлення? Це також може бути ітеративним процесом;

− картографічні дані та запити (MDQ). Зображає запити з даними, визначеними на кроках 1 і 2, щоб створити логічні таблиці, які є представленнями таблиць Cassandra високого рівня;

− створіть фізичні таблиці (PT). Перетворіть логічну модель даних у фізичну модель даних (PDM) за допомогою операторів CQL CREATE TABLE;

− перегляд та уточнення фізичної моделі даних. Підтвердьте, що фізичні таблиці відповідатимуть 3 основним цілям для моделі даних Cassandra (див. рис. 5.6).

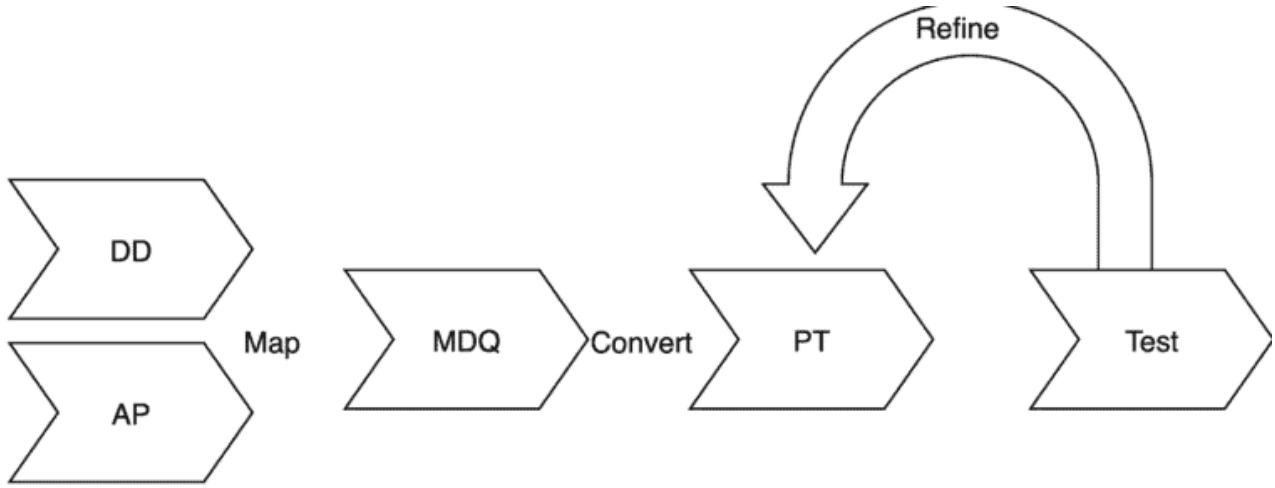


Рисунок 5.6 – Етапи проектування структури зберігання даних Cassandra

Модель даних допомагає визначити проблему, дозволяючи розглянути різні підходи та вибрати найкращий. Це гарантує, що всі необхідні дані збираються та зберігаються ефективно. Моделі документують важливі концепції та жаргон, що є основою для довгострокового обслуговування.

Цілі успішної моделі даних Cassandra полягають у виборі ключа розділу, який розподіляє дані рівномірно між вузлами в кластері, мінімізує кількість розділів, прочитаних одним запитом, і обмежує розмір розділу.

6 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

6.1 Big Data послідовності

Оскільки цифрові дані генеруються надзвичайною швидкістю, розробники та аналітики мають широкий спектр можливостей, коли мова йде про операціоналізацію даних та підготовку їх до аналітики та машинного навчання.

Одне з основних питань, яке потрібно поставити під час планування архітектури даних, – питання пакетної чи потокової обробки: чи оброблюються дані по мірі їх надходження, в режимі реального часу чи майже в реальному часі, чи очікується накопичення даних перед виконувати свою роботу ETL?

При обробці потокових дані оброблюються, щойно вони надходять, що часто також буде дуже близьким до часу їх створення (хоча це не завжди так). Зазвичай це відбувається протягом секунди, тому для кінцевого користувача обробка відбувається в режимі реального часу. Ці операції зазвичай не мають статусу або можуть зберігати стан, тому зазвичай передбачають відносно просту трансформацію або обчислення (див. рис. 6.1).

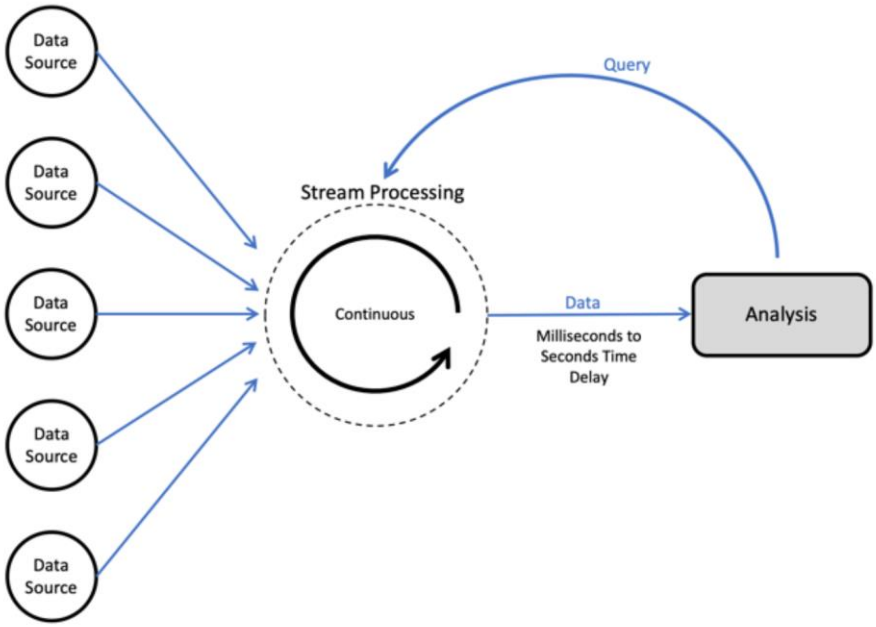


Рисунок 6.1 – Обробка потокових даних.

При такій архітектурі, кожен елемент потоку розглядається як атомарний. Тобто кожен елемент потоку буде проходити послідовність із бізнес-логікою окремо.

Потокова обробка використовується для виявлення складних проблем і надання розумної реакції для кращого результату. Методи машинного навчання для прогнозного обслуговування покладаються на обробку потоку. Прогнозне обслуговування в режимі реального часу забезпечує швидке виявлення та класифікацію несправностей, що розвиваються, а також їх розташування.

Використання складної технології обробки подій дає змогу стежити за станом системи в режимі реального часу. Таким чином, виявлення причин несправностей при оцінці стану технічних компонентів. Ця інформація є основою для початку різних заходів технічного обслуговування в рамках планування технічного обслуговування.

Такий тип архітектури надає наступні переваги:

− при використанні потокової обробки дані завжди оновлюються. Таким чином, можна реагувати на проблеми та події в найкоротші терміни;

− дані оновлюються в режимі реального часу, щоб допомогти виявити закономірності та отримати уявлення про можливі загрози чи можливості;

− випадки затримки під час виконання потокової обробки мінімальні.

6.2 Power BI

Power BI – це хмарний пакет послуг бізнес-аналітики від Microsoft. Він використовується для перетворення вихідних даних у значущу інформацію за допомогою інтуїтивно зрозумілих візуалізацій та таблиць [12]. Можна легко аналізувати дані і на їх основі приймати важливі бізнес-рішення. Power BI – це набір інструментів бізнес-аналітики та візуалізації даних, таких як програмні послуги, програми та конектори даних, які разом утворюють Power BI.

Сервіс дозволяє використовувати набори даних, імпортовані в Power BI, для візуалізації та аналізу даних, створюючи спільні звіти, інформаційні панелі та програми. Power BI – це зручний інструмент, який пропонує вражаючі функції перетягування та можливості самообслуговування. Користувач може розгорнути Power BI як на локальних, так і на хмарних платформах.

Однією з найбільших переваг Power BI є те, що платформа дає можливість організаціям (будь-яких розмірів) краще створювати культури, керовані даними. Культура, яка керується даними, — це культура, в якій рішення приймаються на основі даних, а не почуттів чи інтуїції.

Power BI допомагає компаніям виконати це складне завдання, безперешкодно передаючи активи бізнес-аналітики (наприклад, звіти в режимі реального часу та інформаційні панелі) в руки всіх (за бажанням) в організації. Це створює ефект, коли вся компанія може приймати рішення на основі надійних даних у реальному часі.

Дані стають дійсно цінними, коли вони можуть надати корисну бізнес-ідею та розповідати історії, які допомагають особам, які приймають рішення, приймати кращі рішення.

Power BI дозволяє легко оживити дані. Користувачі можуть підключати свої дані до Power BI і вибирати з різноманітних візуалізацій (стовпчикові діаграми, кругові діаграми, бульбашкові та теплові карти, діаграми розкиду тощо), щоб розповідати історії про свої дані, щоб отримати та поділитися інформацією.

Реалізація цих візуалізацій проста, а функціональність Power BI спрощує перетягування та зміну розташування візуальних елементів для створення глибоких, чистих і добре організованих звітів і інформаційних панелей.

Крім того, ці візуалізації є інтерактивними, що є величезною перевагою Power BI. Споживачі можуть фільтрувати звіти та інформаційні панелі та спостерігати, як їхні візуалізації коригуються та оновлюються за секунди, щоб відповідати їхнім нещодавно застосованим вимогам до фільтрів – і все, щоб отримати більше уявлень про свої дані.

Далі Power BI підключається до сотень джерел даних. Power BI може читати дані з Microsoft Excel і текстові файли, такі як XML і JSON. Power BI підключається до сервера SQL та інших баз даних. Він може зчитувати дані, що зберігаються в хмарі, з джерел Azure та з онлайн-сервісів, таких як Google Analytics і Facebook [13]. Дані можна отримувати з одного або кількох джерел і зберігати в наборах даних для автономного аналізу. Крім того, прямі запити можуть отримувати дані в режимі реального часу, щоб надати до другого перегляду. Microsoft додає більше джерел. Швидше за все, якщо Power BI бачить джерело даних, він зможе його прочитати.

Power BI володіє можливостями штучного інтелекту, які дозволяють користувачам отримувати більшу цінність від своїх даних і звітів. Він пропонує три потужні візуалізації штучного інтелекту, які розробники можуть використовувати, щоб глибше зануритися в свої дані та отримати уявлення та визначити тенденції, які, можливо, важко знайти.

6.3 База даних

Керуєте даними відіграє вирішальну роль у забезпеченні позитивного досвіду користувачів. Зрештою, не має значення, наскільки добре розроблений інтерфейс програми та наскільки чистий код, якщо програма не здатна швидко отримувати, обробляти та передавати інформацію. Більше того, всі ці дані повинні бути захищені, щоб зловмисники не змогли отримати їх. На щастя, цього можна досягти за допомогою грамотно обраної системи керування базами даних.

Існує багато доступних баз даних, і вибір однієї бази даних над іншою є складним рішенням.

Для аналізу вибору бази даних використовувалася теорема CAP. Ця теорема дозволяє швидко та якісно обрати базу даних, яка буде покривати усі необхідні ситуації, в залежності від домену проекту.

Правильний вибір бази даних складає значний відсоток успішності всього проекту (див. рис. 6.2).

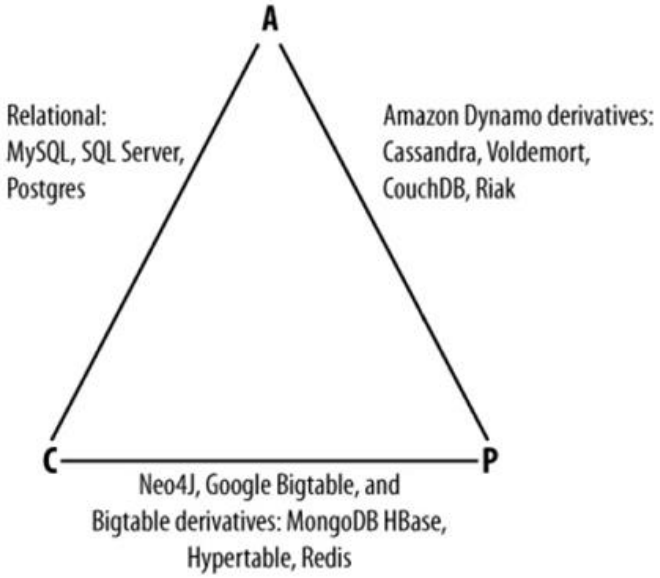


Рисунок 6.2 – Теорема CAP у графічному вигляді.

Ця теорема дозволяє розглянути бази даних, використовуючи знання про домен:

− C (consistency) – узгодженість. Кожне читання дає самий останній запис;

− A (availability) – доступність. Кожен вузол (який зараз існує і функціонує) завжди успішно виконує запити (на читання і запис);

− P (partition tolerance) – стійкість до розподілу. Навіть якщо між вузлами немає зв'язку, вони продовжують працювати незалежно один від одного.

Підсумувавши все, можна побудувати наступний трикутник із прикладами баз даних

На вибір залишилися декілька баз, Cassandra, DynamoDB, Riak, CouchDB. Після розгляду економічної частини та швидкого аналізу конкурентів, була обрана база даних Cassandra. Крім того, бібліотеки для Cassandra існують для різних платформ і мов програмування, що підходить для подальшого розвитку системи із використанням мікросервісної архітектури.

Cassandra заснована на архітектурі розподіленої системи [14]. У найпростішій формі Cassandra може бути встановлена на одній машині або в докер контейнері, і вона добре працює для базового тестування. Один екземпляр Cassandra називається вузлом. Cassandra підтримує горизонтальну масштабованість, яка досягається шляхом додавання більш ніж одного вузла як частини кластера Cassandra. Масштабування працює з лінійним підвищенням продуктивності, якщо ресурси налаштовані оптимально (див. рис. 6.3).

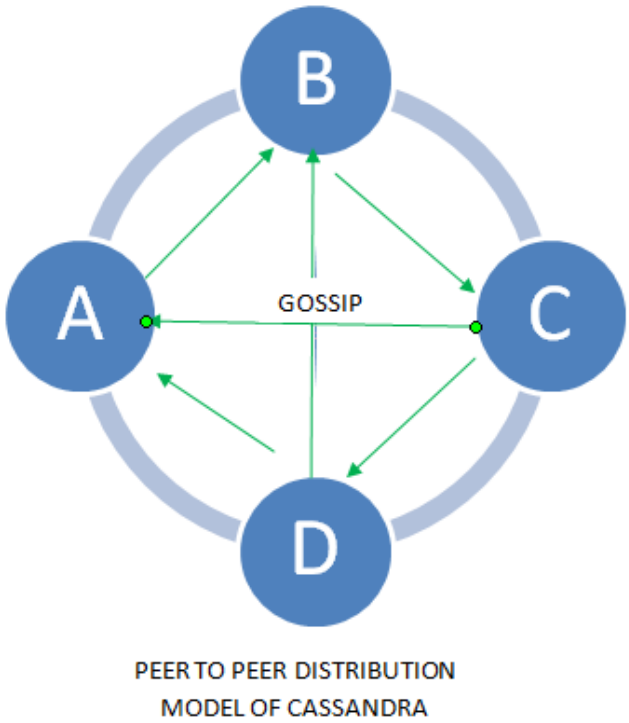


Рисунок 6.3 – Кільцева архітектура Cassandra.

Cassandra працює з одноранговою архітектурою, при цьому кожен вузол підключений до всіх інших вузлів. Кожен вузол Cassandra виконує всі операції з базою даних і може обслуговувати клієнтські запити без потреби в головному вузлі. Кластер Cassandra не має жодної точки збою в результаті одноранговой розподіленої архітектури.

Для доступу до даних використовувався коннектор Cassandra для платформи Flink. Обрана база має велику кількість переваг перед іншими NoSQL базами даних, які аналізувалися для вибору.

Відкрите джерело. Cassandra – проект з відкритим кодом Apache, це означає, що він доступний безкоштовно. Фактично, його природа з відкритим кодом породила величезну спільноту Кассандри, де однодумці діляться своїми поглядами, запитами, пропозиціями, пов’язаними з великими даними. Крім того, Кассандру можна інтегрувати з іншими проектами з відкритим кодом Apache, такими як Hadoop (за допомогою MapReduce), Apache Pig та Apache Hive.

Рівна архітектура. Cassandra дотримується однорангової архітектури, замість архітектури лідера-послідовника.

Отже, в Кассандрі немає жодної точки невдачі. Більше того, будь-яку кількість серверів / вузлів можна додати до будь-якого кластера Cassandra будь-якому з центрів обробки даних. Оскільки всі машини знаходяться на однаковому рівні, будь-який сервер може задовольнити запит будь-якого клієнта. Безперечно, завдяки своїй надійній архітектурі та винятковим характеристикам, Cassandra підняла планку набагато вище, ніж інші бази даних.

Еластична масштабованість. Однією з найбільших переваг використання Cassandra є її еластична масштабованість. Скупчення Cassandra можна легко збільшити або зменшити. Цікаво, що будь-яку кількість вузлів можна додавати або видаляти в кластері Cassandra без особливих завад. Вам не потрібно перезапускати кластер або змінювати запити, пов’язані з програмою Cassandra, під час масштабування вгору чи вниз. Ось чому Cassandra популярна, маючи дуже високу пропускну здатність для найбільшої кількості вузлів. У міру масштабування пропускна здатність читання та запису збільшується одночасно з нульовим простоєм або будь-якою паузою для програм.

Висока доступність та стійкість до несправностей. Ще однією вражаючою особливістю Cassandra є реплікація даних, яка робить Кассандру високодоступною та відмовостійкою. Реплікація означає, що кожні дані зберігаються більш ніж в одному місці. Це пов’язано з тим, що навіть якщо один вузол виходить з ладу,

користувач повинен мати можливість легко отримувати дані з іншого місця. У кластері Cassandra кожен рядок реплікується на основі ключа рядка. Ви можете встановити кількість копій, які потрібно створити. Подібно масштабуванню, реплікація даних може також відбуватися в декількох центрах обробки даних. Це додатково призводить до високого рівня резервного копіювання та відновлення компетенції в Кассандрі.

Висока продуктивність. Основною ідеєю розробки Cassandraбуло використання прихованих можливостей декількох багатоядерних машин. Cassandra здійснила цю мрію. База продемонструвала блискучі результати завдяки великим наборам даних. Таким чином, її люблять ті організації, які щодня мають справу з величезною кількістю даних і в той же час не можуть дозволити собі втратити такі дані.

Орієнтований на стовпці. Cassandra має дуже високу модель даних – вона орієнтована на стовпці. Це означає, що Cassandra зберігає стовпці на основі назв стовпців, що призводить до дуже швидкого нарізання. На відміну від традиційних баз даних, де імена стовпців складаються лише з метаданих, у іменах стовпців Cassandra також можуть бути фактичні дані. Таким чином, рядки Cassandra можуть складатися з маси стовпців, на відміну від реляційної бази даних, яка складається з декількох рядків стовпців. Cassandra наділена багатою моделлю даних.

6.4 Зберігання чутливих даних

У даному проекті існує велика кількість даних, якими не можна ділитися з іншими сервісами. Це паролі до баз даних, спеціальні секрети для хмарних сервісів, ключі до застосунків, тощо. Якщо зберігати їх у локальних файлах налаштувань, то це може поставити під удар безпеку персональних даних та всієї системи в цілому. Саме для таких випадків існують спеціальні сервіси, які дозволяють поділяти цю інформацію лише серед заданої групи додатків. Крім того, для цього використовують спеціальні зашифровані протоколи комунікації.

В якості сервісу для зберігання чутливих даних було обрано хмарний сервіс від Microsoft – Azure KeyVault. Завдяки реєстрації додатків у Azure Active Directory, уся інформація може бути розповсюджена лише серед зареєстрованих учасників. Крім того, налаштування Azure KeyVault дозволяють динамічно додавати чи прибирати доступ до даних. Це дуже корисно у ситуаціях перерозгортання системи на інших ресурсах. У такій ситуації старі ресурси позбавляться доступу, а нові, при умові, що у скриптах розгортання буде вказана адреса та ролі, які слід надати, отримають доступ.

6.5 Визначення експертної думки

Для виявлення експертної думки у отриманих твітах будуть використані декілька основних елементів [15]. По-перше, для первинної фільтрації будуть використані публічні метрики твітів. Такі метрики як кількість вподобань, кількість пересилок, кількість коментарів та кількість підписників у акаунті дозволяють отримати базове розуміння про джерело твіту. Відомо, що акаунти із великої кількістю вподобань, пересилок та коментарів є найбільш улюбленими та перевіреними часом. Первинна фільтрація дозволить поділити джерела на 3 групи. Для цього можна встановити порогові величини метрик (наприклад, достовірне джерело – це те, яке має не менш ніж 1000 підписників та 500 вподобань на публікації). Після встановлення порогових величин, необхідно задати коефіцієнти для кожного елемента публічних метрик та обчислити так званий первинний коефіцієнт для акаунту [16].

Наступним етапом для виявлення експертності є аналіз коментарів під публікацією. На цьому етапі необхідно швидко проаналізувати настрій тексту коментарів. Це дозволить виявити базові емоції (позитивні, негативні, нейтральні) та з’ясувати, чи містить у собі публікація заздалегідь відому фейкову інформації, чи користувачі позитивно ставляться до неї. Після цього, можна задати другий коефіцієнт для акаунту – вторинний коефіцієнт експертності.

Така послідовність дозволяє скласти два коефіцієнти та розділити по пороговим значенням акаунти на 3 групи: достовірні, середньо достовірні та фейкові.

7 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

7.1 Модульне тестування

Одним із різновидів тестування програмного забезпечення є модульне тестування (unit testing). Модульне тестування, або юніт-тестування – процес в програмуванні, що дозволяє перевірити на коректність окремі модулі вихідного коду програми.

Ідея полягає в тому, щоб писати тести для кожної нетривіальною функції або методу. Це дозволяє досить швидко перевірити, чи не призвела чергова зміна коду до регресії, тобто до появи помилок в уже протестованих місцях програми, а також полегшує виявлення і усунення таких помилок [17].

Мета модульного тестування - ізолювати окремі частини програми і показати, що окремо ці частини працездатні.

Модульне тестування має багато переваг:

− заохочення змін. Модульне тестування пізніше дозволяє програмістам проводити рефакторинг, будучи впевненими, що модуль як і раніше працює коректно (регресійні тестування). Це заохочує програмістів до змін коду, оскільки досить легко перевірити, що код працює і після змін;

− спрощення інтеграції. Модульне тестування допомагає усунути сумніви з приводу окремих модулів і може бути використано для підходу до тестування «знизу вгору»: спочатку тестуючи окремі частини програми, а потім програму в цілому;

− документування коду. Модульні тести можна розглядати як «живий документ» для протестованого класу. Клієнти, які не знають, як використовувати даний клас, можуть використовувати юніт-тест в якості прикладу;

− відділення інтерфейсу від реалізації. Оскільки деякі класи можуть використовувати інші класи, тестування окремого класу часто поширюється на пов'язані з ним. Наприклад, клас користується базою даних; в ході написання тесту програміст виявляє, що тесту доводиться взаємодіяти з базою. Це помилка, оскільки тест не повинен виходити за кордон класу. В результаті розробник

абстрагується від з'єднання з базою даних і реалізує цей інтерфейс, використовуючи свій власний mock-об'єкт. Це призводить до менш пов'язаному коду, мінімізуючи залежності в системі.

В якості фреймворку у проекті використовується ScalaTest. Побудова тестових рішень разом із цим фреймворком дозволяє зробити код компактним та зрозумілим для читання (див. лістинг 7.1).

```
it should "successfully search tweets with selected topic" in {

  val testCase = for {
    token <- twitterClient.authenticate
    tweets <- twitterClient.searchTweets("spacex",
token.access_token)
  } yield tweets

  testCase.map(resp => {
    assert(resp.data.nonEmpty)
    assert(resp.meta.nonEmpty)
    assert(resp.errors.isEmpty)
  }).unsafeRunSync()
}
```

Лістинг коду 7.1 – Приклад тест-кейсу із застосуванням ScalaTest.

Як можна побачити, для тестів також використовується особливість модульних тестів: задання поведінки для сервісів (mocking). Для цього використовується бібліотека ScalaMock. Вона дозволяє швидко та зручно підготувати об’єкт для подальшого тестування.

7.2 API тестування

Інший вид тестування, який використовується у даному проекті – є API тестування. На відміну від модульного тестування, API тестує не конкретні складові коду, а повний функціонал. У нашому випадку буде тестуватися вже розгорнута версія продукту.

В якості інструменту для створення API тестів використовувався Postman (див. рис. 7.1).

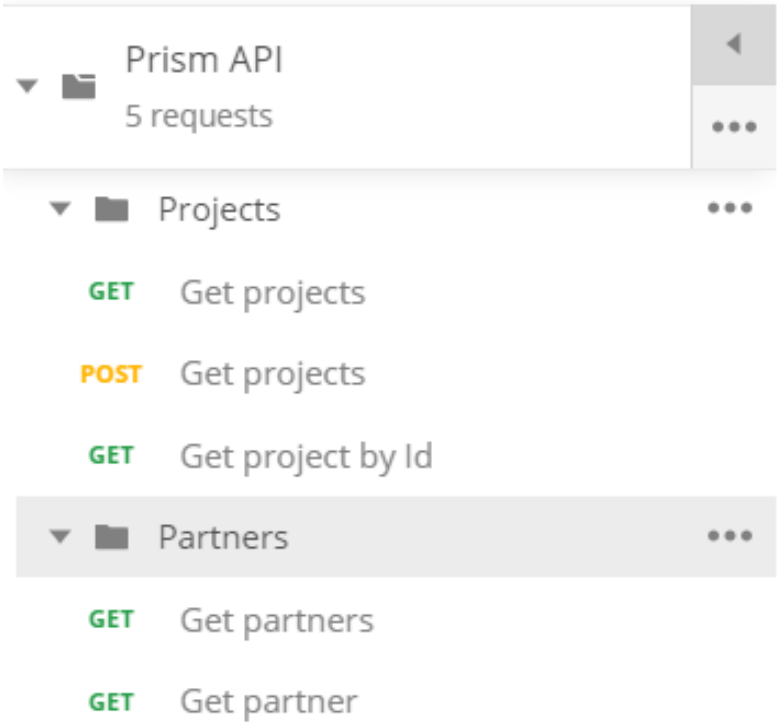


Рисунок 7.1 – Приклад організації Postman колекції

Цей інструмент дозволяє створити колекцію запитів, в яких можна вказати вид запиту (GET, POST, PUT, тощо), корисне навантаження, заголовки, тощо. Для створення послідовності запитів використовується Postman колекція.

На малюнку можна побачити, що колекція у Postman організується із застосуванням звичайних папок. Методи, які тестуються є звичайними точками входу на контролерах. Такий вид тестування допомагає у короткий час протестувати повний функціонал та виявити слабкі та сильні сторони. Крім того, завдяки Postman колекції можна організувати будь-яку послідовність дій (наприклад створення об’єкту у базі даних, його зміну та видалення) так, щоб це не впливало на середовище.

За результатами тестів можна скласти звіти щодо кількості позитивних та негативних результатів. Таку статистику будує сам Postman, та її можна експортувати для використання у різноманітних аналітиках та журналах логів.

8 ОПИС ПРОВЕДЕНИХ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

Для проведення експериментального дослідження було прийнято рішення використовувати реалізовану послідовність на базі інструменту Flink.

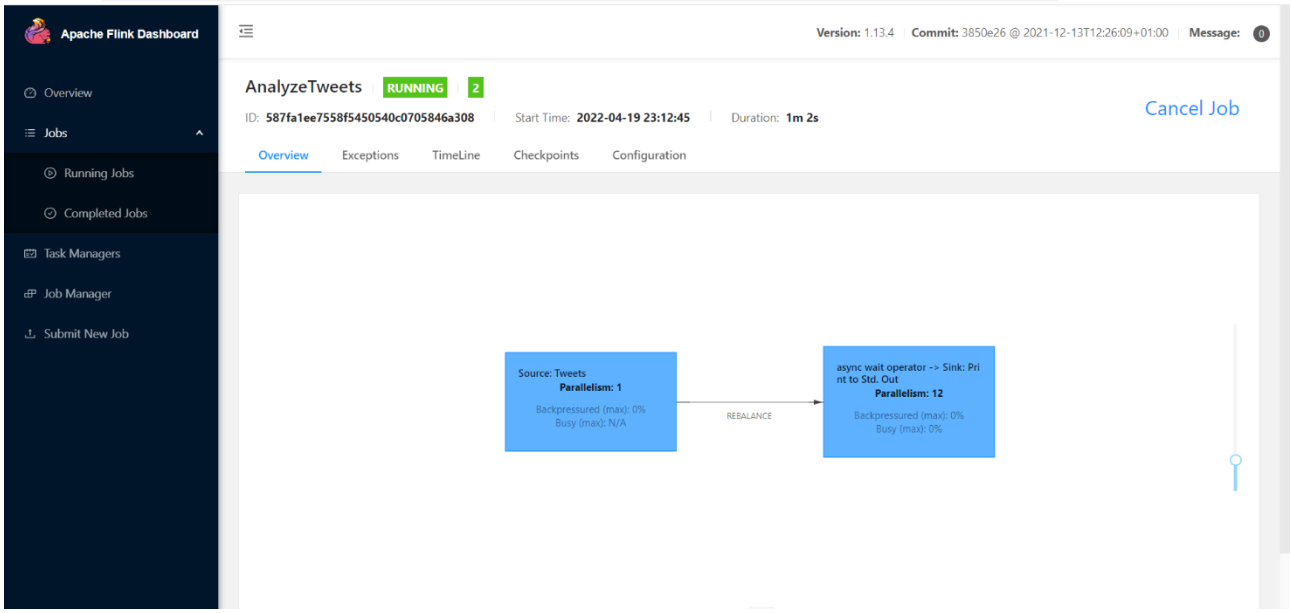


Рисунок 8.1 – Flink UI послідовності Twitter.

Основною метою експерименту є перевірка методу обробки потокової інформації у режимі реального часу. Для цього було необхідно реалізувати поєднувач між соціальною мережею Twitter та Flink.

Для реалізації цієї задачі був використаний Twitter API v2 Streaming, та http4s клієнт. Для десереалізації JSON формату був використаний json4s. У рамках експерименту був використаний рівень parallelism 1, це означає, що кількість робочих вузлів, які паралельно працюють дорівнює одиниці.

Для задання достатнього навантаження для послідовності були обрані найбільш трендові хеш-теги: #elonmusk, #dogecoin, #spacex, #metaverse #ukraine.

Слід відзначити, що такий досить широкий набір різноманітних тем дозволяє отримати навантаження, приблизно від 3 до 10 записів на секунду, в залежності від часу доби. Оскільки Flink використовує концепцію true streaming (кожний новий запис розглядається як окрема одиниця, яку необхідно провести крізь всю бізнес-

логіку), навантаження у 10 записів на секунду буде гарним індикатором для перевірки.

За результатами експерименту можна визначити, що кожен запис був проаналізований менш ніж за 0, 275 секунди, що є дуже чудовим результатом. Це означає, що час між отриманням інформації, її обробкою та доставкою для генерації аналітичного звіту дозволяє отримувати дані у режимі реального часу. Кінцевий користувач не зможе помітити затримки при оновлені інформації.

Також слід зазначити, що даний експеримент проводився зі стабільний зв’язком із мережею Інтернет, яка досягається шляхом використання технології 4G. Для більш навантажених експериментів слід розгортати послідовність у хмарі та налаштовувати віртуальні мережі та пропускні можливості приватних мереж.

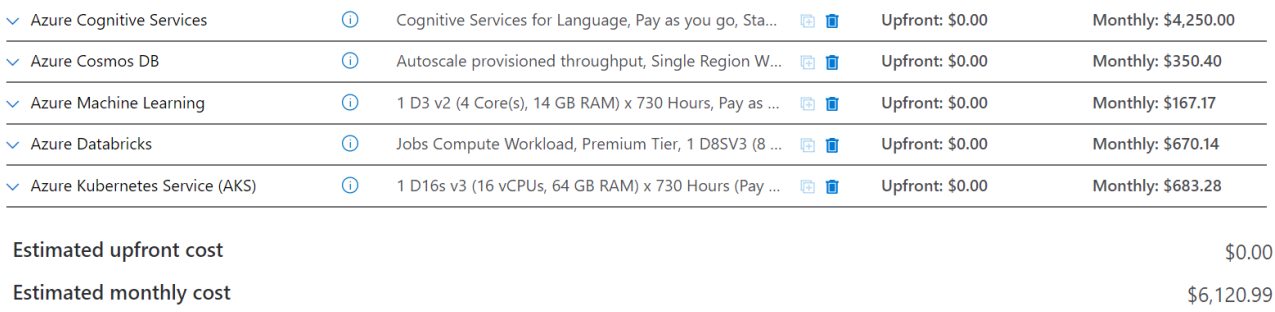
9 ЕКОНОМІЧНА ОЦІНКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для приблизної оцінки вартості підтримки проекту був використаний інструмент від компанії Microsoft Pricing Calculator. Він дозволяє спрогнозувати, скільки необхідно платити щомісяця, для того, щоб підтримувати інфраструктуру.

Основними елементами інфраструктури, згідно з архітектурною діаграмою будуть наступні:

- Kubernetes Services;
- Azure Databricks;
- Azure Cosmos DB;
- Azure Machine Learning;
- Azure Cognitive Services.

Завдяки Pricing Calculator були підібрані оптимальні конфігурації, які дозволять оцінити початку вартість підтримки готової платформи даних (див. рис. 9.1).



Azure Cognitive Services	Cognitive Services for Language, Pay as you go, Sta...	Upfront: $0.00	Monthly: $4,250.00
Azure Cosmos DB	Autoscale provisioned throughput, Single Region W...	Upfront: $0.00	Monthly: $350.40
Azure Machine Learning	1 D3 v2 (4 Core(s), 14 GB RAM) x 730 Hours, Pay as ...	Upfront: $0.00	Monthly: $167.17
Azure Databricks	Jobs Compute Workload, Premium Tier, 1 D8SV3 (8 ...	Upfront: $0.00	Monthly: $670.14
Azure Kubernetes Service (AKS)	1 D16s v3 (16 vCPUs, 64 GB RAM) x 730 Hours (Pay ...	Upfront: $0.00	Monthly: $683.28

Рисунок 9.1 – Оцінка вартості бази Azure Kubernetes Service

Отримана сума у 6,120.99 USD є початковою та у майбутньому може збільшитись. Основними майбутніми тратами можна визначити масштабування кластеру Azure Kubernetes та Azure Databricks, тобто основної обчислюваної сили. Також, при великому масштабуванні можуть зрости витрати на Cosmos DB через додатковий об’єм даних та крос-регіональну реплікацію.

Слід зазначити, що Microsoft надає великі знижки за Enterprise варіанти їх інструментів, тому майбутнє масштабування може вийти плавним та зручним. Крім того, вартість на багатьох сервісах складається тільки з часу використання (pay-as-you-go модель). Така зручна модель оплати ресурсів дозволяє значно заощаджувати та надає можливість розгортати велику кількість сервісів одночасно.

10 ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

У результаті вивчення методів дослідження аналізу потокових даних у режимі реального часу для визначення експертної думки був проведений експеримент із розробкою прототипу платформи даних, яка дозволяє використовувати дані із соціальної мережі Twitter.

Поточна дослідження призвело до створення системи, яка є цілком готовим до використанням продуктом. Крім того, система має великий потенціал та чіткий план розвитку продукту із декількома варіантами розвитку.

Першим і основним варіантом для продовження дослідження є підключення системи до декількох нових джерел, які дозволяють використовувати дані, які генеруються для визначення так званої експертної думки. При цьому варіанту розвитку необхідно проаналізувати дані із нових джерел (дані можуть бути різних типів, як текстові так і відео-матеріали та графіка). Для цього необхідно розробити власну нейрону мережу, яка буде тренуватися на вже існуючих даних із додаванням нового контенту. Таким чином, нейронна мережа буде натренована початковими текстовими даними, та готова для розпізнавання інформації у відео-контенту, що буде розібраний у текст. Різноманітність джерел інформації та власна нейронна мережа дозволить більш детально розпізнавати ключові фактори, які впливають на побудування експертної думки, та більш детально розподіляти акаунти на вихідні категорії.

Другий варіант полягає у більш глибокому зануренні та інтеграції із Twitter. Для цього необхідно зосередитися на аналізі виключно текстового контенту та використовувати велику історію бази знань Twitter. Для цього підійдуть Enterprise Twitter API, які дозволять проаналізувати історичні дані акаунтів на натренувати власну модель.

Отже, обидва варіанти є дуже перспективними для подальшого розвитку та поглибленого вивчення теми динамічного аналізу даних.

ВИСНОВКИ

У результаті роботи були проаналізовані методи аналізу динамічних даних та можливості визначення експертної думки. У роботі було виконано дослідження особливостей використання технології BigData для обробки потокової інформації, методи, які доступні зараз, їх позитивні та негативні сторони. У процесі аналізу методів обробки потокової інформації був обраний метод data streaming, який дозволяє розглядати кожен новий елемент, що надходить для обробки, як окрему одиницю обробки даних.

Під час аналізу предметної сфери було висвітлено основні проблеми використання технології BigData для аналізу потокових даних. Також детально був проаналізований домен соціальних мереж та функції виявлення експертної думки, виконано широкий огляд програмних аналогів та їх порівняння

Для аналізу отриманих теоретичних навичок був проведений експеримент, який детально розкриває і описує правильність вибору методу аналізу потокових даних.

Для набуття практичних навичок та в якості експерименту було спроектовано та реалізовано платформу даних для визначення експертної думки на основі стрімінгової обробки даних з Twitter.

Розробка велась на базі платформи Flink 1.13 з використанням мови програмування Scala. Система була побудована за допомогою архітектурного підходу Distributed Data Mesh. В якості бази даних була обрана Cassandra, а в якості технології доступу до даних – Flink Sink для мови Scala. В якості сервісі для зберігання секретів був використаний Azure KeyVault. Для текстового аналізу вхідних була побудована послідовність дій, яка включає в себе перевірку вхідних даних на орфографічні помилки за допомогою Bing Spellcheck, виявлення ключових фраз за допомогою Azure Text Recognition (Text Analytics API). Серверна частина даного проекту була протестована двома видами тестів: модульними із

використанням фреймворку ScalaTest та API із використанням інструменту Postman.

Розроблена система є активним прототипом та може в поточному стані використовуватися в реальних умовах. Для корекції роботи інформаційної системи необхідно встановити налаштування узлів обробки на рівень, який може утримувати навантаження реальних продуктів. Це можливо зробити завдяки системі оркестрації Azure Kubernetes Services. У AKS оркестратор може динамічно налаштовувати кількість активних вузлів у кластері та заощаджувати ресурси при низькому навантаженню. Саме за таких мінімальних змін система буде готова масштабуватися та бути відмовостійкою.

У результаті експериментів були перевірені можливість обробки потокових даних із соціальних мереж, були обрані методи обробки потокової інформації. Окрім того, були перевірені формати даних, отримані із соціальних мереж. За результатами перевірки була обрана соціальна мережа Twitter, яка має оптимальний зміст даних, які придатні для подальшої роботи. Головним результатом дослідження та експериментальних перевірок стала позитивна відповідь на питання, чи можливо у режимі реального часу виявляти дійсно достовірну інформацію, збагачувати її та прибирати фейкові дані.

Також за результатами роботи були написані тези доповіді для участі у науковій-технічній конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління» [18].

При подальшому вивчені цієї теми можливо додавання нових джерел інформації та розширення існуючого функціоналу шляхом додавання нових варіантів візуалізації. Додатково гілкою. Крім того, можливий перехід на enterprise версію API, що відкриє отримання додаткової інформації про кожну публікацію

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Каук В.І., Гребенюк В.О., Пуголовок К.М., Водяницький Д.В.Виклики, які надають нові можливості Екстренне дистанційне навчання в Україні: Монографія/ За ред. В.М.Кухаренка, В.В. Бондаренка. - Харків:. Вид-во КП "Міська друкарня", 2020.- 409 с.

2. Viktor Kauk, Vyacheslav Grebenyuk, K. Pugolovok, D. Vodyanytskyi. Distance learning techniques and technologies for effective and successful online learning. – Available at: https://www.researchgate.net/publication/351794521_Distance_learning_techniques_and_technologies_for_effective_and_successful_online_learning (дата звернення: 08.05.2022).

3. Tatyana Strelkova, Yuliya Soroka, Olesia Tieliezhkina, Viktor Kauk. Online Learning Methods for Effective Communication Between Teachers and Students. Available at: https://www.researchgate.net/publication/357500923_Online_Learning_Methods_for_Effective_Communication_Between_Teachers_and_Students (дата звернення: 08.05.2022).

4. Understand How Society Changes in Real Time. Available at: https://www.citibeats.com/ (дата звернення: 08.05.2022).

5. Social Market Analytics The Leader in Unstructured Financial Data. Available at: https://www.socialmarketanalytics.com/ (дата звернення: 08.05.2022).

6. Real-time AI for Event and Risk Detection. Available at: https://www.dataminr.com/ (дата звернення: 08.05.2022).

7. В.В. Стецюк та Т.В. Грищук Сучасні методи обробки потокових даних. XLVIII Науково-технічна конференція факультету комп'ютерних систем і автоматики 2019 №5.

8. Nicoleta Tantalaki, Stavros Souravlas, Manos Roumeliotis. A review on Big Data real-time stream processing and its scheduling techniques. Available at: https://www.researchgate.net/publication/331447616_A_review_on_Big_

Data_real-time_stream_processing_and_its_scheduling_techniques (дата звернення: 08.05.2022).

9. Рассел Метью, Классен Михайло. Data mining. Извлечение информации из Facebook, Twitter, LinkedIn, Instagram, GitHub. – Питер, 2020. – 464 с

10. Building Microservices Applications on Microsoft Azure / Harsh Chawla, Hemant Katuria – Apress, 2019, 271 с.

11. Use Case – Wikipedia [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Use_case (дата звернення: 15.04.2022)

12. Mastering Microsoft Azure Infrastructure Services / John Savill – Sybex, 2015, 341 с.

13. Web Applications on Azure: Developing for Global Scale / Rob Reigan – Apress, 2017, 269 с.

14. Deep Learning with Azure: Building and Deploying Artificial Intelligence / Daniel Dean, Matthew Salvaris – Apress, 2018, 324 с.

15. Ondenyi E. Extractive Text Summarization Techniques With sumy / E. Ondenyi // Medium. – 2017. – Available at: https://link.medium.com/0SKJXLxGT1

16. Daniel G. Waddington, Nilabja Roy, Douglas C. Schmidt. Dynamic Analysis and Profiling of Multi-threaded Systems

17. Zaharia, Matei. Discretized Streams: A Fault-Tolerant Model for Scalable Stream Processing,2012 ACM SIGMOD/PODS. New.

18. Каук В. І., Єрусалімцев Д. А. Дослідження методів динамічного аналізу даних для визначення експертної думки на основі стрімінгової обробки даних з Twitter. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління Том 2, Секція 5. С 162.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

1. Каук В.І., Гребенюк В.О., Пуголовок К.М., Водяницький Д.В.Виклики, які надають нові можливості Екстренне дистанційне навчання в Україні: Монографія/ За ред. В.М.Кухаренка, В.В. Бондаренка. - Харків:. Вид-во КП "Міська друкарня", 2020.- 409 с.

2. Viktor Kauk, Vyacheslav Grebenyuk, K. Pugolovok, D. Vodyanytskyi. Distance learning techniques and technologies for effective and successful online learning. – Available at: https://www.researchgate.net/publication/351794521_Distance_learning_techniques_and_technologies_for_effective_and_successful_online_learning (дата звернення: 08.05.2022).

3. Tatyana Strelkova, Yuliya Soroka, Olesia Tieliezhkina, Viktor Kauk. Online Learning Methods for Effective Communication Between Teachers and Students. Available at: https://www.researchgate.net/publication/357500923_Online_Learning_Methods_for_Effective_Communication_Between_Teachers_and_Students (дата звернення: 08.05.2022).