ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ

Пояснювальна записка

до магістерської роботи

на тему: “ Дослідження безсерверних обчислень та їх використання в мережах мобільного зв'язку”

Виконав: студент 7 курсу, групи АРЗМ-71
спеціальності
172 Телекомунікації і радіотехніка
(шифр і назва спеціальності)
Завалій М.С.
(прізвище та ініціали)
Керівник
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)
Нормоконтроль_____

Київ – 2021

Вступ

У наш час неможливо уявити сучасний світ без обчислень, вони оточують нас у всіх сферах життя починаючи від звичайної оплати карткою у супермаркеті до наукових розрахунків. Основна проблема обчислень – зростаюча складність та кількість запитів за секунду. Компаніям, що пов'язані з обчисленнями необхідно весь час збільшувати серверні потужності, працювати над інфраструктурою мережі обробки даних та наймати спеціалістів для підтримки таких систем. Уся складність та витрати на розробку серверних систем припадає на компанію розробника, що є затратним для починаючих компаній (стартапів) та ентузіастів, діяльність яких пов'язана з обчисленнями. Для вирішення таких проблем розробляється технологія “безсерверних обчислень”, технологія являє собою модель хмарних обчислень для яких платформа динамічно керує виділенням машинних ресурсів. Дана технологія надає змогу користувачам використовувати ресурси, що їм фізично не належать, а усі проблеми з налаштування та підтримки обладнання припадають на постачальників послуг. Також за рахунок інтелектуального відстеження навантаження від користувача, система автоматично виділяє необхідну потужність та ресурси для проведення обчислень в даний момент, тобто користувач оплачує лише ті ресурси, що він використовує. Основна проблема технології “безсерверних обчислень” не пристосованість до потреб багатьох сфер та проблеми з сертифікацією та стандартизацією але вони поступово вирішуються, тобто дослідження використання такої технології в телекомунікаційних системах є актуальним для розвитку науки та компаній у країні.

1. Структура оператора стільниковового зв'язку, LTE та аналіз вхідних даних від абонентів.

Телекомунікаційна мережа, як комплекс технічних засобів телекомунікацій та споруд призначених для маршрутизації, комутації, передавання та приймання знаків, сигналів, письмового тексту, зображень і звуків або повідомлень будь-якого роду по радіо, дротових, оптичних чи інших електромагнітних системах між кінцевим обладнанням, зазнала значних змін, а саме у розрізі мобільного зв’язку. Перехід від перших поколінь до LTE відбувся із змінами в інфраструктурі мережі, структурі самих мобільних операторів, різноманітних підходів та принципів побудови та опрацювання і обробки даних. Технологія LTE та її впровадження з сучасними можливостями надає змогу у повному обсязі використовувати безсерверні технології , що дозволяє впливати на значне зменшення витрат для утримання та дозволяє розробляти нові блоки інфраструктури мережі в цілому.

1.1. Структура оператора стільниковового зв'язку та мережі LTE

Мобільний протокол LTE передавання даних проект 3GPP, стандарт з вдосконалення UMTS для задоволення майбутніх потреб у швидкості. Long Term Evolution (LTE) розглядається як еволюція технології UMTS, є стандартом високошвидкісного бездротового зв'язку передачі даних розроблений групою 3GPP визначеного спеціальними специфікаціями. Формально LTE відноситься операторами мобільного зв'язку та виробниками до стандарту 4G, хоча не належить до нього фактично, оскільки не відповідає технічним критеріям мобільного зв'язку 4-го покоління визначеного вимогами «IMT-Advanced» Сектором радіозв'язку Міжнародного телекомунікаційного союзу. Наразі, технологія LTE, є актуальною в Україні, оскільки планується подальше впровадження технології (5G). Постійним розвитком телекомунікаційних технологій, є розробка та впровадження нових стандартів зв’язку, що в свою чергу забезпечує прискорення передачі даних, підвищує якість наданих даних, знижує значні витрати на зв’язок та має значний вплив на зниження витрат на обладнання.

Мережа LTE по архітектурі має в собі наступні складові та принципи:

- Розподілені транспортні підмережі, для споживача та службової інформації.
- Деякі функції мережі пакетної передачі даних та радіо доступу, можуть бути фізично реалізовані на тому ж обладнанні. Транспортні функції мережі не є обов’язковими. Мережа радіо доступу та пакетна, може бути повністю звільнена від функцій транспортних пакетів передачі даних. Адресація та її схема, при реалізації транспортних функцій не повинні бути пов’язані та можуть застосовуватися в однакових структурах мережі.
- Мережа радіо доступу виконує функції управління мобільністю користувача та може бути призначена для власника терміналів зв’язку.
- Інтерфейс повинен мати функціональний розподіл та мати декілька важливих опцій для виконання радіо доступу:
  - Інтерфейс керування даними повинен базуватися на стандартних логічних моделях та блоками;
  - Один фізичний елемент містить у собі декілька логічних блоків.

На рис. 1.1 представлена модель еволюції мереж стільникового зв'язку, на якій видно, що всі технології надходять в одну точку, якою є переважлива технологія мобільних протоколів LTE.

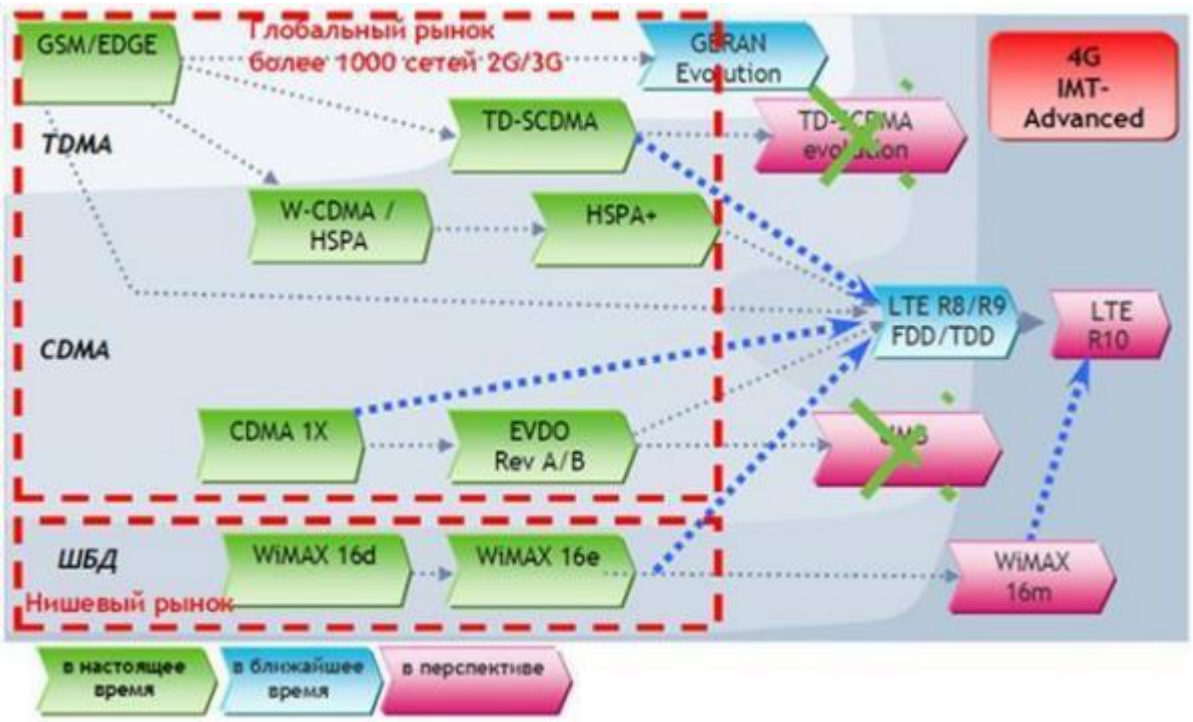


Рис. 1.1. Еволюція мереж мобільного зв'язку

Мережа LTE забезпечує пакетному трафіку доставку покетів з мінімальними затримками, або без втрат даних пакетів передачі (по принципу seamless або mesh) під час мобільності зв'язку та підтримкою технології збереження якості обслуговування (QoS).

Існує два основні типи мобільності: Роумінг, або дискретна мобільність та хендовер, або безперервна мобільність. Хендовер, або безперервна мобільність - процедура передачі активного з'єднання між стільниками. Хендовер дозволяє абонентам не бути прив'язаними до будь-якої географічної точки та дає можливість пересуватися в межах мережі оператора без розриву з'єднання. Причиною хендовера може бути не тільки переміщення абонента в просторі, але і погіршення якості сигналу від поточної базової станції з якими-небудь іншими ознаками. Зокрема між абонентом та БС може виникнути перешкода, погіршення метеоумови, яку обслуговує базова станція, або її частина може вийти з ладу і т.п.

Мережі LTE повинні підтримувати роумінг та хендовер з існуючими мережами, для користувачі (терміналів) має забезпечуватися повсюдне покриття послуг широкосмугового доступу до мережі. Передачу мовного трафіку та

забезпечити надання всіх послуг, дозволяє пакетна передача. Архітектуру LTE можна назвати "плоскою", яка зображена на рис 1.2., оскільки раніше спостерігалась висока ієрархічність, різноманітність та різнотипність мережевих вузлів. Розподілена мережева відповідальність відбувається між двома вузлами, такими як Базова Станція (БС) та Блоком Управління Мобільністю (БУМ). БС - в технічних специфікаціях називається B-вузлом (Node-B, eNB), а БУМ ((MME, або Mobility Management Entity), як реляційна БД включає в себе мережевий шлюз (Gateway або GW). Тобто, мають місце комбіновані блоки MME/GW.

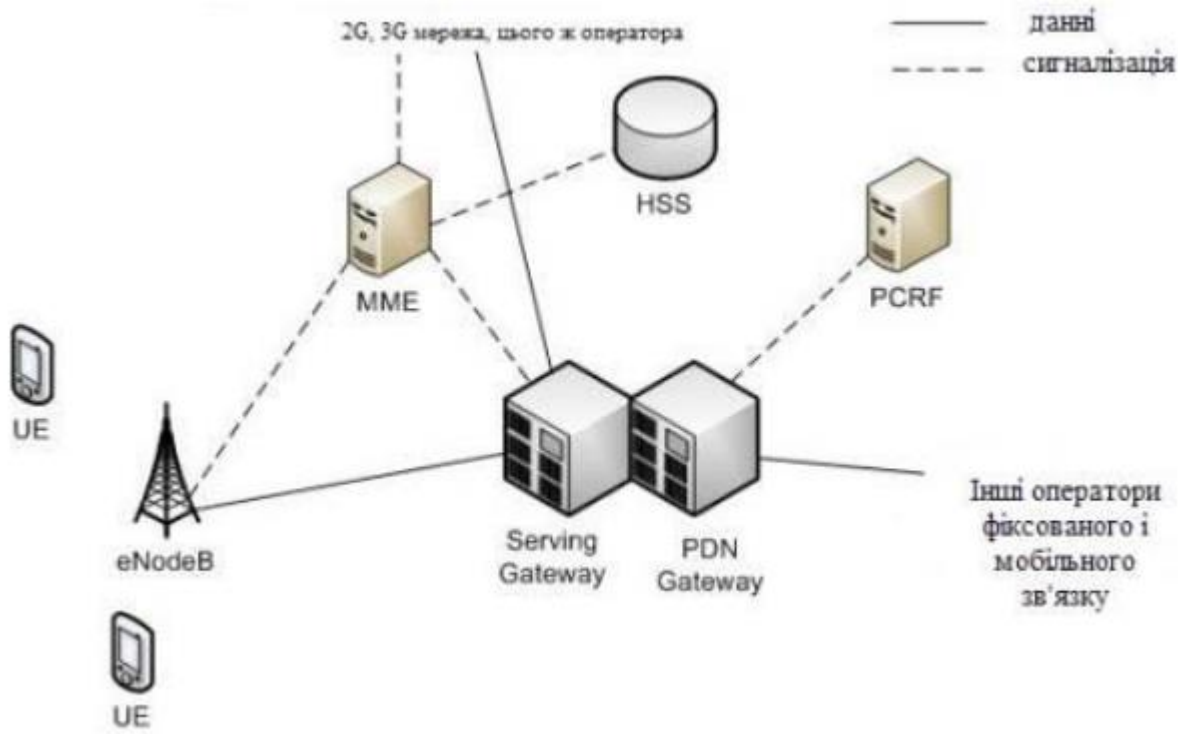


Рис 1.2. Архітектура мережі LTE

Контролер сукупності радіоелектронних засобів (радіомережі), раніше був присутній в структурних схемах та керував потоком даних, відігрував роль в різних мережах більш застарілих поколіннях, але наразі усунутий та не приймає участь у керуванні потоками даних. Традиційні функції контролера радіомережі під час керування радіоресурсами які були передані безпосередньо до базової станції: шифрування даних, забезпечення надійної доставки пакетів данихЮ

стиснення разних заголовків та ін. IP – пакети, що містять інформацію про споживача та через нього не проходять, обробляються мережевою сигналізацією, яка підпорядкована управлінню Блоком Управління Мобільністю та відповідає за службову інформацію. Для передачі службової інформації споживача перевагою блоку сигналізації, є здітність зрощування трафіку, як нежалежна функція. Функціями Блоку Управління Мобільністю (БУМ) є управління терміналами споживача в режимі очікування, забезпечення перенаправлення і здійснення викликів, роумінг, надання службових та призначених для споживача каналів, хендовер, авторизація та аутентифікація.

Еволюція системної архітектури, це архітектура ядра мережі, розроблена консорціумом 3GPP для стандарту бездротового зв'язку LTE. Архітектура SAE знижує експлуатаційні та капітальні витрати. Нова, плоска модель, означає, що потрібно підвищити пропускну здатність вузлів тільки двох типів (базових станцій і шлюзів), щоб вони впоралися з трафіком в разі його значного зростання. Перші концепції 3G були розроблені, з тим, щоб голос як і раніше передавався по системі з комутацією каналів. З тих пір спостерігався перехід до IP-мереж. Відповідно архітектура SAE побудована на базі IP-мережі. Спадний канал (Down Link) буде зі швидкістю понад 100 Мбіт/с, і основна увага системи буде зосереджено мобільності смуги пропускання, від мережі потрібно підтримувати набагато більше рівнів даних. Зі збільшенням необхідних рівнів взаємодії і більш швидких відповідей, концепція SAE забезпечить рівень затримки в районі 10 мс. Підтримує мобільність між декількома гетерогенними RAN, що включає підтримку, як систем типу GPRS, так і не-3GPP систем (наприклад WiMAX). Основним компонентом архітектури SAE є Evolved Packet Core (EPC). EPC служить еквівалентом мережі GPRS. Існує певний розподіл мережевих шлюзів та компонентів EPC.

Вузол Управління Мобільністю - (MME) - це ключовий контролюючий модуль для мережі доступу LTE. Він відповідає за процедури забезпечення мобільності, хендовера, стеження і пейджінга пристрою споживача. MME являє

собою заключну точку мережі для захисту цілісності сигналізації NAS і відповідає за управління безпекою.

Обслуговуючий Шлюз (SGW) - призначений для обробки і маршрутизації пакетних даних, що надходять в підсистему базових станцій. Він керує і зберігає стан користувацького пристрою (КП), вимоги щодо пропускної спроможності для IP-сервісів та внутрішню інформацію з мережевої маршрутизації. Він також надає копію призначених для користувача даних при правомірному перехопленні інформації.

Пакетний шлюз (PGW) - забезпечує з'єднання від КП до зовнішніх пакетних мереж даних, будучи точкою входу і виходу трафіку для КП. КП може мати одночасно з'єднання з більш ніж одним PGW для підключення до декількох мереж. PGW виконує функції захисту, фільтрації пакетів для кожного користувача, підтримку білінгу, правомірного перехоплення і сортування пакетів. Інша важлива роль PGW - бути вузлом управління мобільністю між 3GPP і не-3GPP технологіями.

Вузол виставлення рахунків абонентам (PCRF) - відстежує потік послуг, що надаються, і забезпечує тарифну політику. Для додатків, що вимагають контроль або нарахування плати в режимі реального часу, може використовуватися додатковий мережевий елемент під назвою функції додатків (AF).

В основі LTE як і вмережах попередніх поколінь, особливо третього покоління, при формуванні побудови мережі закладено два основні принципи: фізична реалізація окремих мережевих блоків та формування багатофункціональних зв'язків між ними. Завдання фізичнох реалізації вирішується шляхом використання концепції області (domain), а функціональні зв'язки розглядаються в рамках шару (stratum). Архітектурний поділ мережі на область обладнання користувача та область інфраструктури мережі, є прямим представлення фізичного рівня. Область мережевої інфраструктури поділяється на певні підмережі, такі як мережа радіодоступу E-UTRAN і базова EPC.

Устаткування користувача – це сукупність терміналів з різними рівнями функціональних можливостей, які використовуються абонентами мережі для доступу до LTE-послуг. При цьому в якості призначеного для користувача терміналу може фігурувати: споживач, який користується, послугами голосового трафіку, так і пристроєм призначеним для передачі та прийому певних мережевих додатків, або для додатків споживача.

На рис. 1.3. зображена загальна структура мережі LTE, на якій видно наявність двох шарів функціональних зв'язків: шару радіодоступу (AS, Access Stratum), як функціональний рівень в стеках протоколів бездротового зв'язку UMTS і LTE між радіомережею і призначеним для користувача обладнанням та зовнішнього шару радіодоступу (NAS, NonAccess Stratum), як функціональний рівень в стеках протоколів бездротового зв'язку UMTS і LTE між базовою мережею і призначеним для користувача обладнаннямі де цей рівень використовується для управління у встановленому сеансів зв'язку, для підтримки безперервного зв'язку з призначеним для користувача обладнанням під час його руху. Зображенні на рис. 1.3. овали зі стрілками позначають точки доступу до послуг.

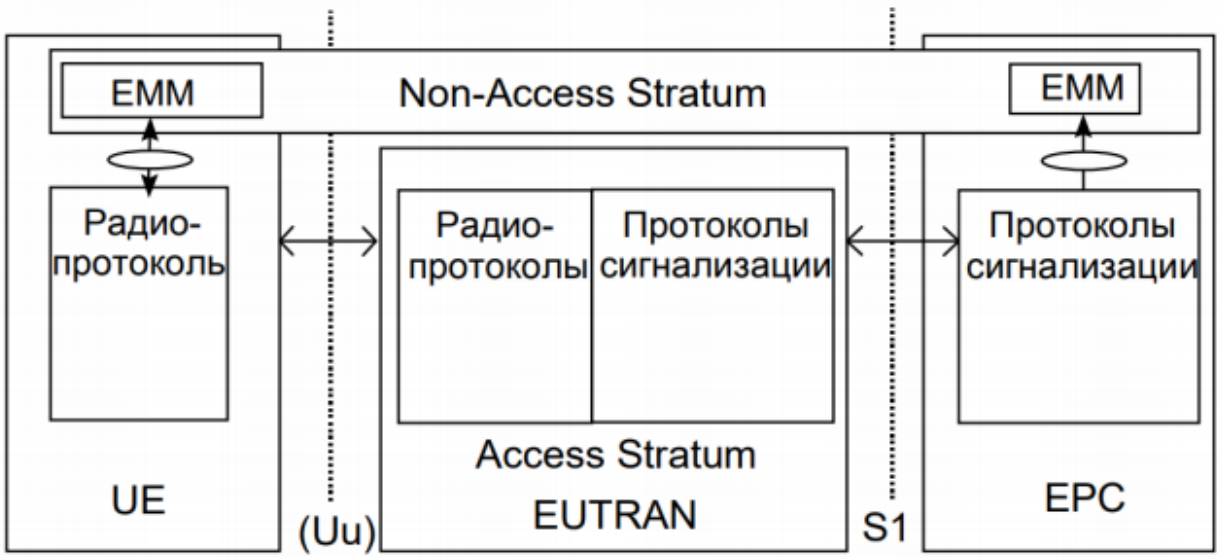


Рис. 1.3. Узагальнена структура мережі LTE

Границя між областями користувацького обладнання та мережі радіодоступу (UTRAN) називається Uu-інтерфейсом; границя між областями мережі радіодоступу та базової мережі EPC - S1-інтерфейсом. Протоколи, що відносяться до інтерфейсів Uu і S1, розділені на дві площині: призначену для користувача площину UP (User Plane) і площину управління CP (Control Plane). Поза шару доступу діють механізми управління мобільністю в базовій мережі (EMM).

У зоні обслуговування споживачів, реалізовані протоколи які забезпечують передачу даних користувача по радіоканалу мережі. До засобів і методологій елементів управління відносяться ті протоколи, які в різних площинах забезпечують з'єднання між терміналом та користувацькою мережею. Також до засобів і методологій відносяться протоколи, призначені для прозорої передачі різних типів повідомлень, що відносяться до надання різних видів послуг. Область мережі радіодоступу логічно розділена на два рівня: рівень радіомережі (RNL) і рівень транспортної мережі (TNL). Взаємодія яка входить до області мережі радіодоступу БС здійснюється на основі X2-інтерфейсу (рис. 1.4.). Крім того, має місце транзитне сполучення між базовими станціями і базової мережею через блок управління мобільністю, або обслуговуючим вузлом – яке показано на рис. 1.4. Таким чином, можно стверджувати, що S1-інтерфейс підтримує множинні зв’язки між набором БС і блоками БУМ та ОВ.

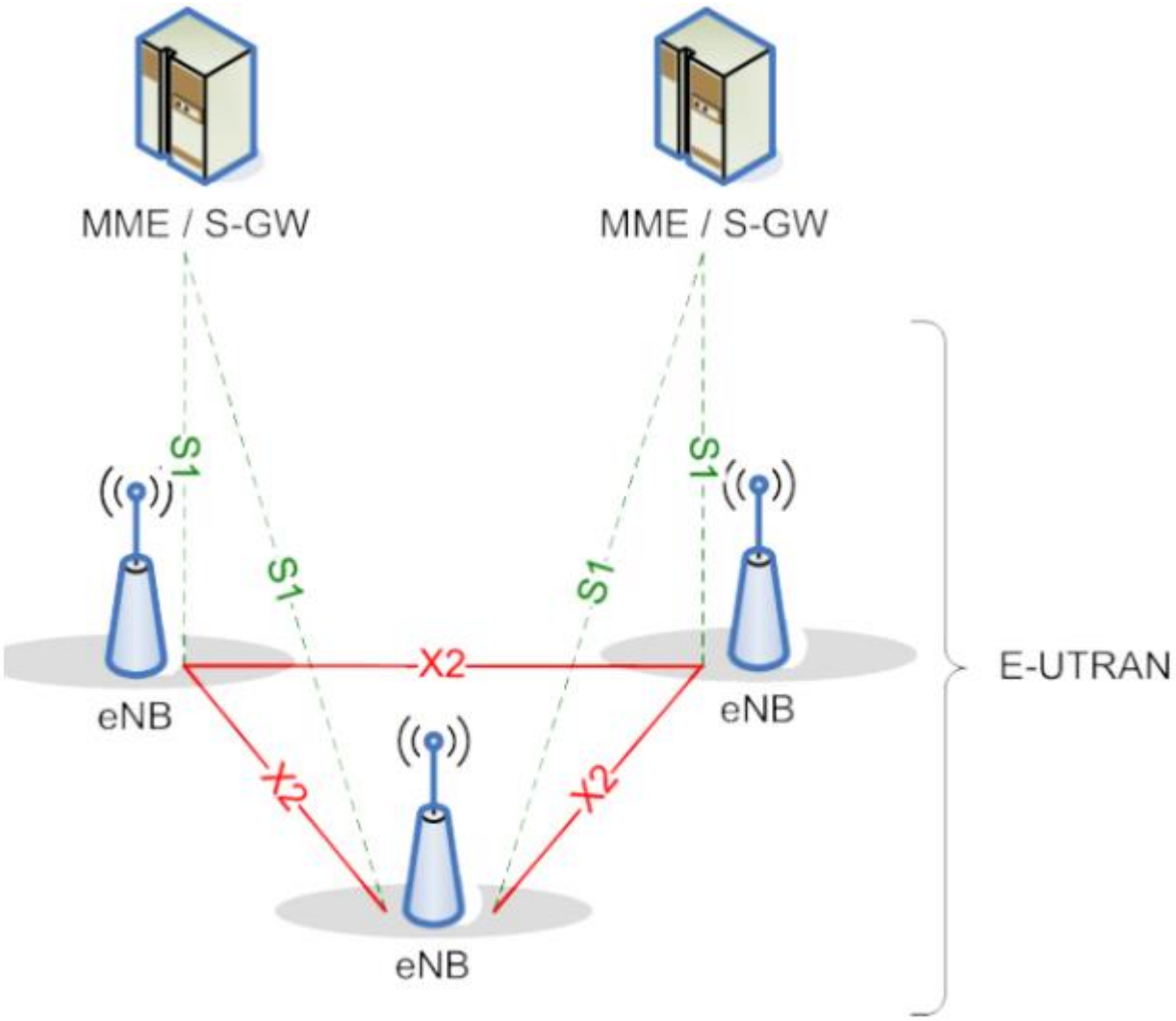


Рис. 1.4. З'єднання функціональних вузлів мережі радіодоступу

1.2. Призначення до мережі радіодоступу функціональних блоків

На БС в мережах LTE покладено виконання таких функцій:

- Управління радіоресурсами дозволяє, розподіл радіоканалів, динамічний розподіл ресурсів в висхідних і низхідних напрямках - так звана диспетчеризація (планування) ресурсів (scheduling) та ін. RRC включають в себе функції встановлення і роз'єднання та з'єднання, трансляцію та передачу системної інформації, встановлення радіоканалу, реконфігурацію і звільнення, процедури мобільності RRC - з'єднання, повідомлення про пошуковий виклик і звільнення, та управління потужністю зовнішнього контуру. За допомогою функцій

сигналізації, RRC конфігурує площину користувача і управління відповідно до стану мережі та дозволяє реалізувати стратегії управління радиоресурсами.

- Метод стиснення заголовків IP-пакетів та шифрування потоку даних користувача, спеціально був створений для поліпшення продуктивності TCP/IP при повільних послідовних з'єднаннях. Метод стиснення зменшує звичайні 40-байт заголовків TCP/IP-пакетів приблизно 10 разів. Це досягається за рахунок запам'ятовування стану TCP-з'єднань на обох кінцях з'єднання, і відсилаються тільки зміни в полях заголовка. Це призводить до досить великої різниці з точки зору інтерактивної продуктивності на низькошвидкісних з'єднаннях, але при цьому цей метод ніяк не бореться із затримками при обробці, яеі властиві більшості модемів для комутованого зв'язку.
- Вибір блоку управління мобільністю при включенні в мережу призначеного для користувача терміналу при відсутності у нього інформації про минуле підключення.
- Маршрутизація в призначеній для споживача зоні пакетів даних у напрямку до обслуговуючого шлюзу.
- Передача викличної та мовної інформації, і диспетчеризація яка отримана від БУМ.
- Передача повідомлень і диспетчеризація PWS (Public Warning System, система тривожного сповіщення), отриманих від БУМ. Система оповіщення населення буде використовуватися для попередження про атаки з повітря, суші або моря, а також про стихійні лиха та інших техногенних катастрофах.
- Обробка, створення та складання відповідних звітів, для керування диспетчеризацією та мобільністю абонентів.

Блок управління мобільністю (БУМ) забезпечує виконання таких функцій:

- Аутентификация.

- Вибір обслуговуючого шлюзу і шлюзу пакетної мережі для мереж радіодоступу різних стандартів.
- Вибір нового блоку управління мобільністю при виконанні хендовера.
- Передача захищеної інформації від точки доступу, до послуг та захищенна лінія управління точками доступу.
- Передача інформації в базову мережу для управління мобільністю між різними мережами радіодоступу.
- Підтримка передачі повідомлень PWS.
- Роумінг.
- Управління БС, що знаходяться в стані очікування, включаючи перенаправлення викликів.
- Управління переліком зон відстеження споживачів.
- Управління радіоканалом, включаючи установку виділеного каналу.

Обслуговуючий вузол відповідає за виконання таких функцій:

- Буферизація пакетів даних в низхідному напрямку, призначених для терміналів, що знаходяться в режимі очікування, і ініціалізація процедури запиту послуги.
- Вибір точки прив'язки локального місця розташування (Local Mobility Anchor) при хендовера.
- Маркування пакетів транспортного рівня.
- Маршрутизація і перенаправлення пакетів даних.
- Санкціонований перехоплення інформації користувачів.
- Створення облікових записів користувачів та ідентифікатора, що відповідає за клас якості обслуговування для відповідної тарифікації.
- Тарифікація абонентів.

Шлюз пакетної мережі забезпечує виконання таких функцій:

- Маркування пакетів транспортного рівня в низхідному напрямку.

- Розподіл IP-адрес для терміналів користувачів.
- Санкціонований перехоплення інформації користувачів.
- Тарифікація послуг, їх селекція.
- Фільтрація користувальницьких пакетів.

1.3. Контроль радіоресурсів

У мережах LTE для терміналу визначені два стани на підрівнях до RRC: з'єднання (RRC CONNECTED) та очікування (RRC IDLE). Протокол управління радіоджерелами (RRC) використовується в UMTS, LTE та 5G. Основні функції протоколу RRC: встановлення і роз'єднання з'єднання, трансляція і передача системної інформації, управління потужністю зовнішнього контуру, процедура мобільності, реконфігурація і звільнення, виклик і звільнення, встановлення радіоканалу, повідомлення про пошук. Завдяки функції сигналізація RRC конфігурує стратегію управління радіоджерелами, реалізує її, дозволяє управління в залежності від стану мережі та конфігурує площини користувача. Під час функціонування, термінал користувача переходить за стану очікування в стан з’єднання (рис. 1.5.), але це можливо на момент успішно встановленого з’єднання. Термінал може повертатися в стан очікування навіть розірвавши з’єднання на підрівні.

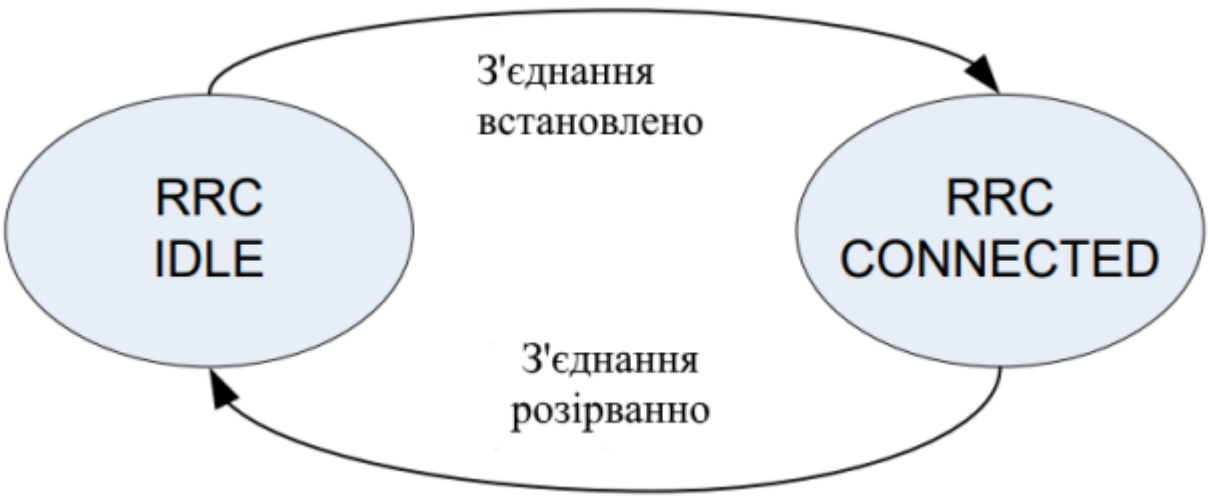


Рис. 1.5. Зміна станів терміналу

В стані очікування (RRC), термінал може: до обслуговуючої соти проводити вимірювання, здійснювати заміну соти, здійснювати моніторинг каналів виклику та робити вимірювання обслуговуючої соти. В стані очікування та за для економії витрат на випромінювання та контролю потужності, здійснюється переривчаста передача та встановлюється на певний цикл. В стані очікування управління мобільністю здійснюється самостійно.

В режимі очікування (RRC), певний термінал під час з'єднання може опрацьовувати передачу індивідуальної інформації в двох напрямках, встановлювати переривчасті цикли передачі даних, а також передавати групову мовну інформацію в низхідному напрямку. Для встановлення зазначених цілей виконується забезпечення зворотного зв'язку, який надає змогу отримувати інформацію про поточний рівень якості наданого сигналу, також робляться необхідні виміри стану, які мають пряме відношення до обслуговуючої соти базової стнації та йде надсилання звітів про вимірювання, які надсилаються в мережу. В стані з'єднання (RRC) управління мобільністю проводить мережа, на відміну від стану очікування.

Самим важливим аспектом мереж всіх мобільних операторів та для мереж LTE, це повна підтримка технології безшовної мобільності кожного абонента в залежності від змін до різних базових станцій, які обслуговують мережеві шлюзи, вузли та мережі в цілому. При використанні пакетів програм, таких як VoIP, які дуже чутливі до затримок певних пакетів, підвищуються вимоги до ефективності мобільності та використання. В основі кожної технології мережі безшовної мобільності використовують та полягають різні процедури хендовера, які при перетині абонентом різних географічних зон використовують естафетну передачу активного абонента, що знаходиться в режимі зв'язку, або з'єднання. Для підготовки до виконання хендовера використовується сигналізація по типу інтерфейсу X2 та між різними базовими станціями. Пов'язано це з тим, що для типових користувачів, здійснення зміни сот базових станцій відбувається значно часто, ніж зміна базової станції, чи обслуговуючого шлюзу.

Найважливішим показником якості роботи мереж є ефективність виконання роботи хендовера, такі як регулювання параметрів на рівні прийняття рішення, магнітний гістерезис та ін., які можуть призвести до зайвого навантаження та завантаження службових каналів і навіть призвести до повної втрати зв’язку. В специфікації TS 36.413 досить детально прописаний алгоритм хендовера, який має виконання в різноманітних мережах. Насправді опису алгоритмів хендовера, присвячена дость велика кількість специфікацій.

Приклад LTE-хендовера при зміні стільника в межах одних і тих же БУМ і ОУ:

В мережах GSM, аналіз оточення для вибору стільника-кандидата на хендовер, здійснюється контролером БС. На відміну від мереж GSM в мережах LTE аналіз навколишнього оточення для вибору стільника-кандидата, виконується самим терміналом, але остаточне рішення про хендовер приймається мережею. На рис. 1.6. приклад хендовера активного КП, який переміщується з соти БС-1 в соту БС-2.

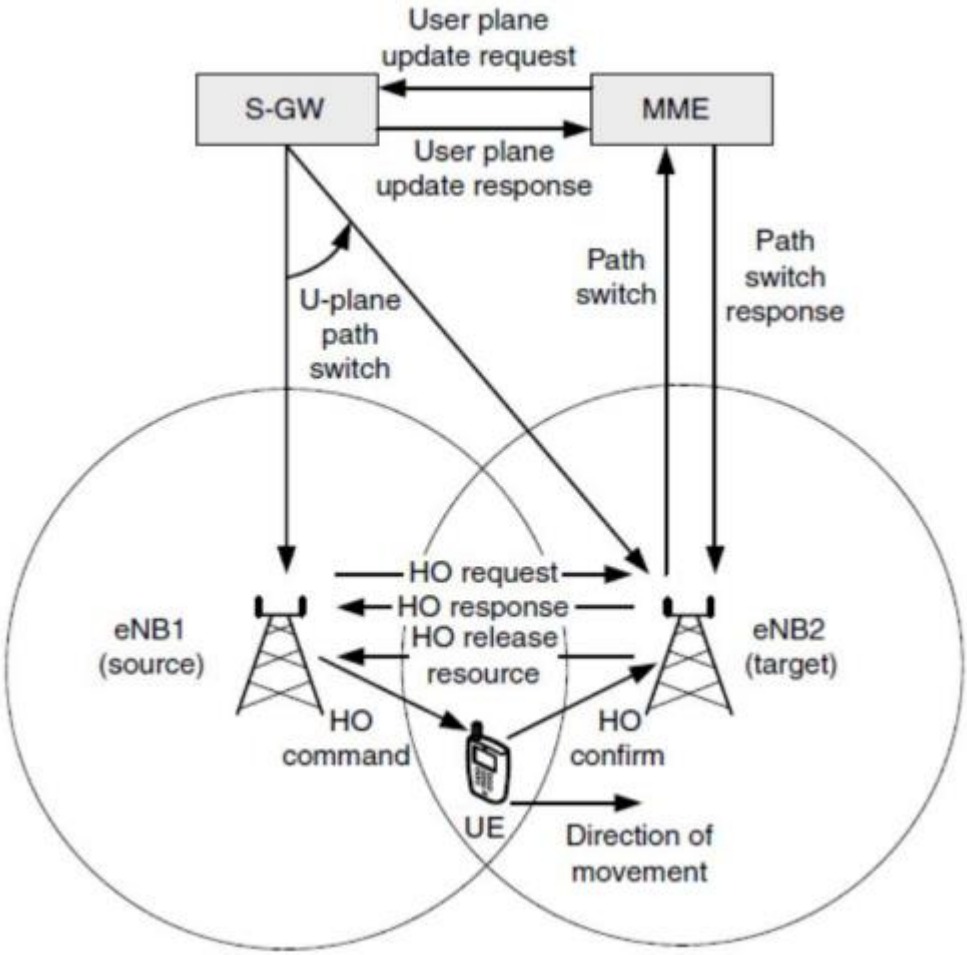


Рис. 1.6. Активний хендовер

Під час руху в певному напрямку КП надсилає звіт з вимірювань базової станції яка є обслуговуючою (БС1), з якого видно, що якість сигналу отриманого від іншої по сусідству базової станції (БС2) краще, ніж на основній БС1. Після підготовки цільового запиту на хендовер, йде підготовка БС1для відправки по інтерфейсу X2в БС2 даних цього запиту:

HO REQUEST

При цьому, на кінцеве та остаточне рішення про хендовер може вплинути поточний користувацький пріоритет, інформацію про який передається EPS-каналом.

БС2-Цільова яка робить налаштування необхідних ресурсів відповідно до прийнятої по EPS-каналу інформації QoS (quality of service) і резервує для

користувача стільник-ідентифікатор C-RNTI, а також, преамбулу каналу доступу. Ідентифікатор C-RNTI є унікальним для кожного терміналу в межах стільника.

Після того, як цільова БС2 відповідає (все ще) обслуговуючої БС1 про готовність виконати хендовер:

HO RESPONSE,

БС1 відправляє команду до терміналу (абонента):

HO COMMAND

про зміну радіоканалу, в якій містяться необхідні параметри: новий тимчасовий ідентифікатор C-RNTI (той, що був зарезервований), преамбула каналу випадкового доступу і час її існування та ін.

Після отримання команди на хендовер КП синхронізується в новій соті, і здійснює доступ в мережу через канал RACH; мережі, в свою чергу, термінал виділяє необхідні ресурси і встановлює необхідний час попередження. Здійснивши доступ до мережі, термінал відправляє своїй новій БС2 повідомлення, яке підтверджує виконання хендовера:

HO CONFIRM,

БС2 сповіщає про це БУМ:

PATH SWITCH,

а на обслуговуючий вузол повідомлення:

USER PLANE UPDATE REQUEST,

в якому термінал визначається і тепер обслуговується БС2.

Далі обслуговуючий вузол виконує перемикання керування терміналом на БС2, відсилаючи туди через БУМ відповідні команди

USER PLANE UPDATE RESPONSE

Та

PATH SWITCH RESPONSE.

Після отримання команди

PATH SWITCH RESPONSE

БС2 інформує БС1 про успішне завершення процедури хендовера, посилаючи їй повідомлення про вивільнення ресурсів:

HO RELEASE RESOURSE,

які до цього були зарезервовані для терміналу. На цьому процедура хендовера вважається виконаною.

Під час підготовки і виконання хендовера між базовими станціями БС1 і БС2 встановлюються (можуть бути встановлені) тунелі: один – для передачі даних висхідного напрямку, інший - для передачі даних спадного напрямку. Це робиться для того, щоб забезпечити передачу довгих пакетів в разі переповнення буферів.

Як раніше вже було зазначено, під час виконання процедури хендовера користувачу присвоюється тимчасовий ідентифікатор C-RNTI. Під час проведення різних типів процедур, які пов’язані з мережею радіо-доступу, базовою мережею, тимчасові аналогічні ідентифікатори присвоюються користувачу, призначених для абонента та мережевого обладнання також.

Так, в елементах управління потужністю TPC по фізичним висхідним каналах PUSCH і PUCH використовуються відповідні ідентифікатори TPC-PUSCH і TPC-PUCCH.

Для глобальної ідентифікації БУМ використовується ідентифікатор GUMMEI, який зконструйований з ідентифікатора мережі мобільного зв'язку, шляхом додавання кодів групи блоку БУМ і кодів самого БУМ.

Ідентифікація на глобальному рівні сот мереж LTE виконується завдяки стільниковому глобальному ідентифікатору ECGI, формування якого залежить від додавання до локального мережевого ідентифікатора з ідентифікацією стільникової мережі CI і використовується в межах однієї мережі. Схожим чином відтворюється ідентифікація БС глобального типу.

Для виконання управління мобільністю КП, яка перебуває в стані очікування, додається ознака зони для відстеження ТА я площини, яка покривається в зоні обслуговування багатьох БС (рис. 1.7). Певний відповідний ідентифікатор, пов’язаний з кожною зоною відслідковування зон TAI. КП може надавати реєстри в електронному вигляді та по кількох зонах відстеження, що в умовах значної мобільності дозволяє економити електроенергію, у зв’язку з відсутністю у необхідності постійного оновлення свого розташування.

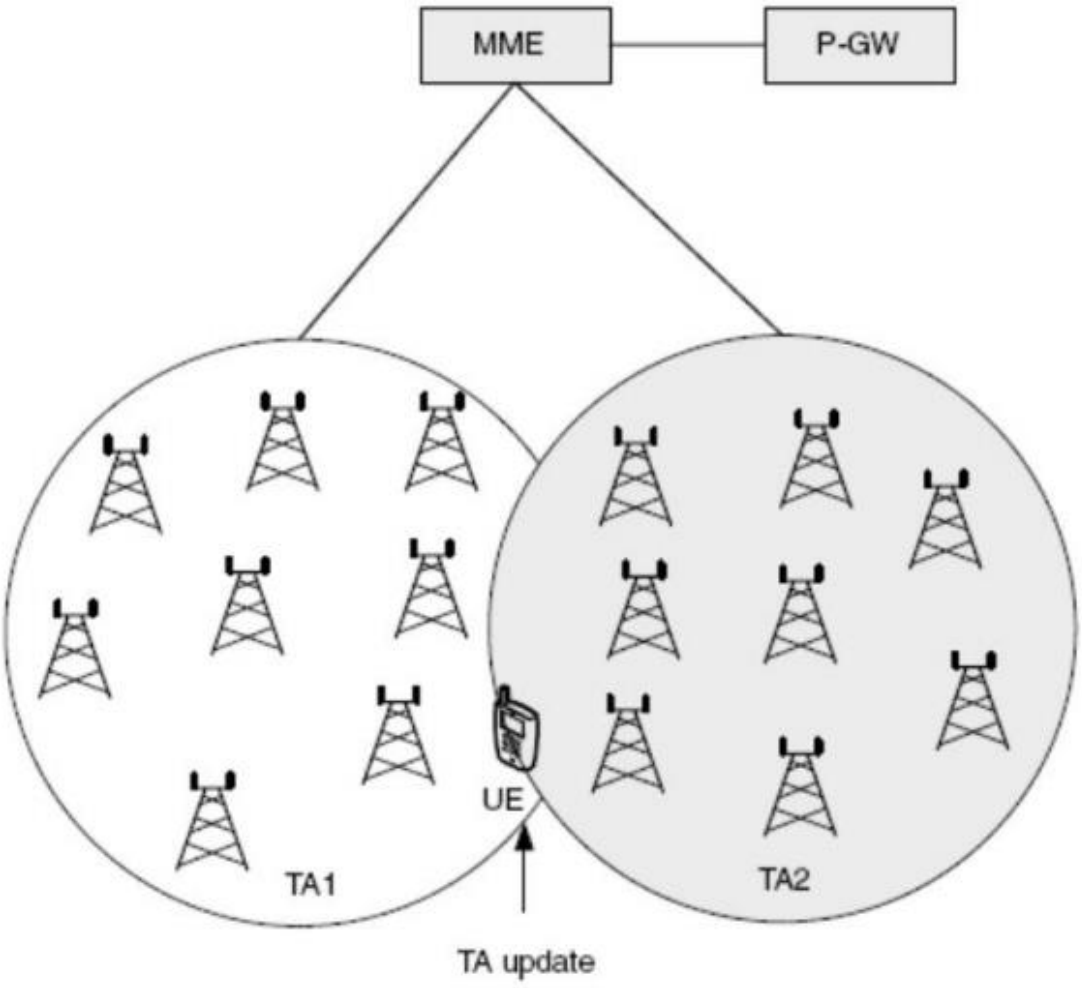


Рис. 1.7. зона відстеження

1.4. Збір та аналіз даних від абонентів

Кожен оператор стільникового зв’язку має змогу відстеження свої абонента (терміналів) та отримувати від них дані, як наприклад під час підключення до мережі, чи під час зміни БС, при проходженні реєстрації в мережі та інших випадках. Оператор стільникового зв’язку має можливість робити збір інформації та даних від абонента, завдяки його підключенню до інтернету. Як приклад додаткових джерел, це WI-FI для отримання координат і запитів, або за допомогою GPS чи інших типів сервісів. Існують як легальні та і не легальні способи отримання інформації від абонентів, тобто ті, які були чи не були здійснені з дозволу абонента. Прикладом легальних способів, є опитування клієнтів через додатки на пристроях в яких дані надходять за допомогою GPS або за допомогою WI-FI. Прикладом не легальних способів, є приклад з технічного боку обладнання, де під час реєстрації, абоненту надається ідентифікатор мережі IMSI в межах GSM, або набір регістрів VLR, HLR, AUC, EIR в межах LTE, де ідентифікація абонента (UE) проходить через MME та HSS.

MME (MultiMedia Extensions) — сімейство програмних інтерфейсів (API) для роботи з мультимедійними пристроями (переважно звуковими платами) і файлами.

HSS (Home Subscriber Server) – сервер абонентських даних мережі стільникового зв'язку стандарту LTE. Являє собою велику базу даних і призначений для збереження даних про абонентів (терміналів). HSS фактично замінює набір регістрів (VLR, HLR, AUC, EIR), які використовувалися в мережах мереж 2G і 3G.

HSS використовується для збереження певних типів інформації:

- призначення для споживачів-ідентифікаторів, даних безпеки абонентів, безпеки номерів і адресної інформації

- інформація контролю доступу в мережу, аутентифікації та авторизації

- інформація місцезнаходження користувача на міжмережевом рівні, особливо коли абонент покине поточну мережу у LTE оператора, то в HSS збережеться інформація про те в яку мережу він перейшов, яка необхідна для його пошуку в разі вхідного дзвінка

- інформація про загальний профіль абонента

Також HSS генерує дані, які необхідні для здійснення шифрування, аутентифікації та інших типів маніпуляцій з ним. Мережа LTE може включати один, або декілька HSS. Кількість HSS залежить від географічної структури мережі та загального числа абонентів.

Присутня можливість пошуку абонентів за допомогою IMEI та так званого вимушеного пошуку через запит від поліції, суду та інших типів офіціального розшуку, але модулі реєстрації IMEI не встановлюються операторами, а ті які встановленні, можуть реєструвати декілька однакових IMEI завдяки тому, що відсутній контроль за так званими «сірими» IMEI.

Отже, основні методи нагляду за переміщенням абонентів в межах мережі оператора:

- за допомогою унікальних ідентифікаторів в мережі (такі як номер телефона), де приблизна точність такого методу приблизно 150 м.

- за допомогою IMEI ідея пошуку в якому полягає, в визначенні поточного номера абонента та потім визначення його місця знаходження тобто як у методі з номером телефона.

- через GPS кординати, так як цей метод надає більш точну інформацію, але потребує підключення абонента до мережі, бажано з доступом до інтернет мобільного оператора (але зазвичай інформація про кординати зашифрована та оператор не отримує до неї доступу), при умові, якщо не встановлено додаток від мобільного оператора з дозволом використання GPS локації і ідентифікації місця знаходження

• поточна дислокація по кординатам Wi-Fi в залежності від розташування на місцевості, де присутній той самий принцип отримання даних с боку мобільного оператора.

В усіх випадках за посну обробку інформації напряму відповідає оператор мобільного (стільникового) зв'язку, це змушує його не тільки зберігати але й оброблювати великі об'єми даних, що потребує значних потреб в якісному обладнані та кваліфікованому персоналі відповідного профілю. Так наприклад для потреб безпеки та відстеження абонентів за запитом поліції дії запиту, відслідковування та надання інформації обґрунтовані, при яких для аналізу анонімних даних дозволено використовувати хмарні технології. Для аналізу навантаження на окремі БС, можна застосовувати обробку хмарних технологій на основі принципу Big-data. Можливість реалізації технології Big-data зображена на рис 1.8.

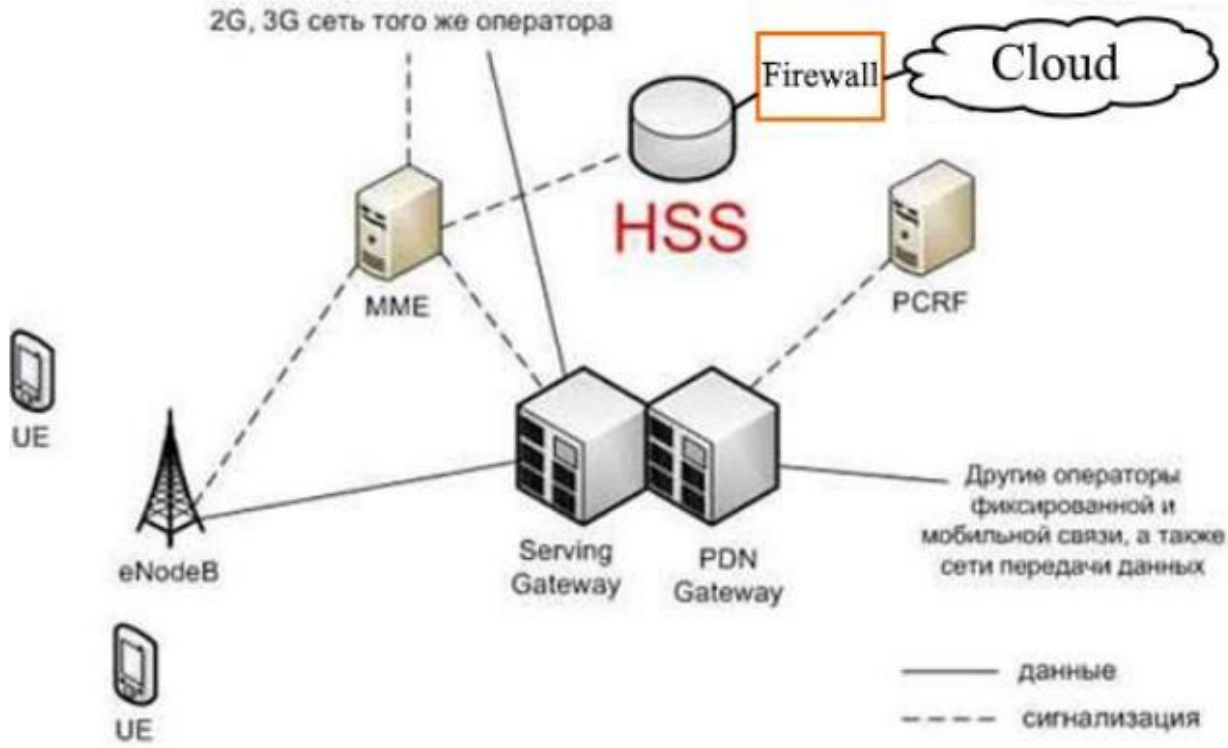


Рис 1.8. Реалізація хмарної технології Big-data в мережі оператора.

Під час розробки технології LTE розглядалась можливість повного застосування хмарних технологій для зменшення участі в усіх процесах

мобільного оператора зв’язку під час обробки даних та надання послуг, та здійснювати лише контроль за роботою Білінгу і підключенням нових абонентів. В такому випадку, оператор лишає позиції обов‘язкового власника технологічної інфраструктури і поступового контролю за абонентами, та центру обробки даних (ЦОД). Оскільки, зазначені можливості змін потребують повного та певного перегляду процесів регулювання в поточних моделях роботи операторів мобільного зв’язку, то це питання залишилось на розгляді у зв’язку з багатьма факторами. Приклад реалізації іншого типу наведено на рис 1.9.

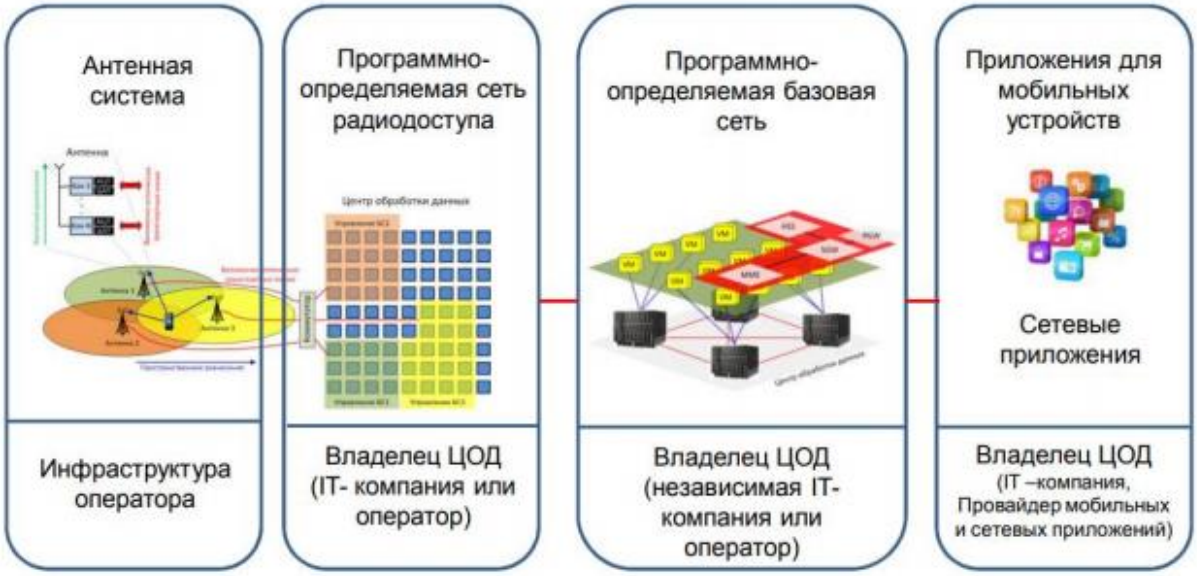


Рис 1.9. Реалізація хмарної інфраструктури оператора

Під час розвитку технологій мобільного зв’язку, спостерігається частковий перехід інфраструктури на хмарні технологій та обчислення, так звана поступова децентралізація.

Висновок: У даному розділі розглянуто та досліджено структуру мобільного оператора зв’язку, технологію LTE (структуру хендовера, підключення абонента та інше), можливість перенесення інфраструктури оператора на безсерверні технології, припустимі втрати позиції оператора як обов’язкового власника усієї технологічної інфраструктури, методи

відслідковування абонентів в мережі за унікальними ідентифікаторами та можливість отримання і аналізу даних від абонентів.

2. Технології обробки Big Data та способи обробки даних

Присутня можливість отримання даних від споживачів (абонентів) мобільних операторів або мережі, це відбувається під час обміну даних між пристроєм і БС та під час переключення абонента між різними базовими станціями. Відбувається це в середньому для одного абонента біля 10 разів на день. Якщо уявити, що в мережі приблизно 2 мільйонів абонентів, то і операцій відбувається приблизно 20 мільйонів записів на день, але це тільки записи переключення між БС і не включає в себе записи у білінгові системи та інші типи даних. Беручи з огляду кількості записів на день, то це Гігабайти даних і для обробки таких об’ємів використовують методи, алгоритми та підходи (Big Data), оскільки стандартним програмно-апаратним комплексом вирішити це буде вкрай складно.

Великі данні (англ. Big Data) – прийшли на заміну традиційних систем управління БД і рішень Business Intelligence, як серія методів і інструментів для обробки різних по структурі (структурованих та не структурованих) даних занадто великих об’ємів, та для отримання значного різноманіття результатів і які можуть сприйматися людиною, які сформовані в умовах безперервного приросту та розподілу по обчисленню вузлів мережі прорахунків і фіксацій різноманітної кількості даних.

Прямим зразком використання Big Data де відбувається аналіз даних з постійними щоденними змінами від користувачів, що вносяться на сайті «Wikipedia.org» рис. 2.1.

Рис. 2.1. Зразок Big data — дані про щоденні зміни, що користувачі вносять у статті на вікіпедії

Як правило, якщо говорити про термін Big Data, використовують найбільш поширене визначення семи «V»:

- Volume – об’єм даних;
- Velocity – обробка з великою швидкістю;
- Variety – різноманіття і структурованість даних;
- Veracity – достовірніість даних;
- Variability – мінливість та точність сприйняття контексту;
- Visualization – візуалізація;
- Value – цінність, витяг користі для збільшення результату.

Наразі, присутнє підтвердження ствердженням експертів, про постійно прискорене зростання обсягів даних, які є об’єктивною реальністю. Джерелами, які здатні постійно генерувати величезні обсяги даних, це соціальні мережі

(Facebook, Twitter та інші), відео-ресурси (Tik tok, Youtube, Instagram, Twitch та інші), мобільні пристрої, вимірювальні пристрої, бізнес-інформація та інші. Значна частина інформації з'являється завдяки різноманітним цифровим пристроям, які взаємодіють один з одним, взаємодіють з різноманітними мережами даних та не створюються користувачами. При поточному зростанні об'ємів інформації, кількість інформації щорічно мінімум подвоюється та матиме постійний коефіцієнт зростання. Дослідження International Data Corporation (IDC) Digital Universe, яке було опубліковане ще в 2012 році, передбачало на найближчі 8 років зріст інформації в світі, який повинен був досягти приблизно 40 Зеттабайт (zettabytes) даних, що в еквіваленті виглядало як 5,2 ТБ (терабайт) даних на кожну людину світу (рис. 1.1). Станом на 2020 рік згідно даних IDC, кількість даних в світі досягло більше 59 Зб, що значно перевищило прогнозований об'єм.

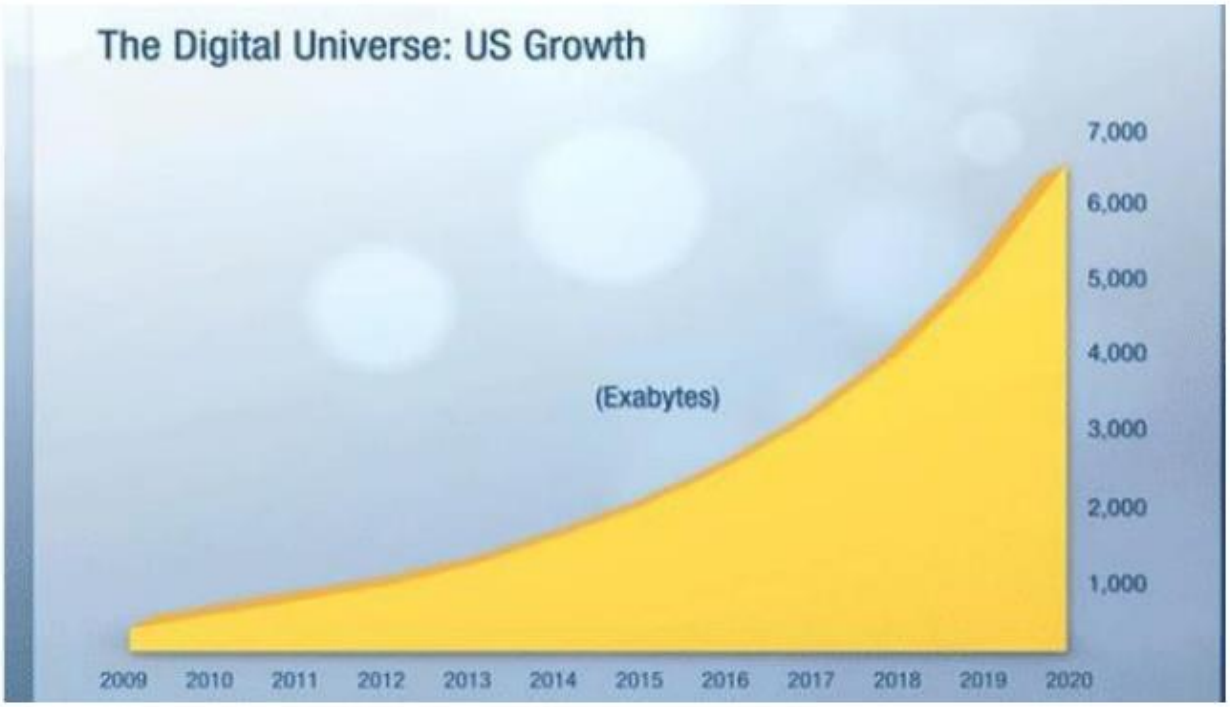


Рис. 2.1 Зростання надходження цифрової інформації в США

Парадигма Big Data визначає наступні типи завдань з даними:

- Дані, з якими звичайні реляційні бази не дозволяють ефективно працювати, повинні бути під управлінням обсягів та зберігатися;

• Неструктурована інформація яка складається з відео, зображень, текстів та інших типів файлів, підлягає обробці та організації;

• Проведення аналітики великих даних, яка включає в себе як методи класу Data Mining так і розпізнання образів, інтелектуальна аналітика, машинне навчання та штучні нейронні мережі;

• Віртуалізація даних, як процес подання інформації у вигляді різноманітних малюнків, діаграм, з використання інтерактивних можливостей та анімації як для отримання результатів, так і для використання в якості вихідних даних для подальшого аналізу.

Робота з Big Data підлягає та підкоряється наступним принципам:

• Горизонтальна масштабованість. Система, яка розроблена для обробки великих даних, повинна легко розширятися та модифікуватися. При швидкому зростанні обсягу даних, де необхідна швидкодія та збереження заданої продуктивності, кількість витрат має бути прямо пропорційна до відповідного зростання.

• Відмовостійкість. Горизонтальна масштабованість передбачає збільшення кількості машин в сервері, де частина з цих машин буде виходити з ладу і потрібна буде постійна підтримка стану налагодження роботи. Метод роботи передбачає врахування збоїв при роботі з великими даними, можливість їх виникнення та можливість пережити їх без будь яких Наслідків.

• Локальність даних. Принцип передбачає так звану децентралізацію даних, але з локальним збереження для зменшення параметрів транспортування та підвищення якості обробки. Як приклад, дані фізично знаходяться на одному сервері, а обробляються на іншому. При цьому витрати на передачу даних можуть перевищувати суму витрат на саму обробку. Тому одним з найважливіших принципів під час проектування рішень Big Data, є застосування локальності

даних, при якому обробка даних проводиться на тій машині, де ці дані зберігаються, а не там де присутній найближчий сервер для обробки даних.

2.1. Технології використання обробки Big Data

На початку розвитку різних методів та різноманітних технологій для побудови Big Data, оскільки однією з проблем, є проблема телекомунікацій у вигляді обчислення занадто великих об'ємів даних, виникло та було запропоновано багато різних платформ, методів та рішень, які в наш час є основою і стандартом в конкретній сфері використання.

Apache Hadoop

Однією з ключових технологій реалізації концепцій Big Data є платформа Hadoop.

В рамках Apache Hadoop project розробляється відкрите програмне забезпечення для надійних, масштабованих розподілених обчислень.

Apache Hadoop - вільно поширюваний набір утиліт, бібліотек і фреймворк для розробки і виконання розподілених програм, що працюють на кластерах з сотень і тисяч вузлів. Використовується для реалізації пошукових і тематичних механізмів багатьох високонавантажених веб-сайтів. Розроблено на Java в рамках обчислювальної парадигми MapReduce, згідно з якою додаток поділяється на велику кількість однакових елементарних завдань, здійсненних на вузлах кластера та природним чином приводяться в кінцевий результат.

Проект складається з чотирьох модулів - Hadoop Common (сполучна програмне забезпечення - набір інфраструктурних програмних бібліотек і утиліт, використовуваних для інших модулів і споріднених проектів), HDFS (розподілена файлова система), YARN (система для планування завдань і управління кластером) і Hadoop MapReduce (платформа програмування і виконання розподілених MapReduce-обчислень), раніше в Hadoop входив цілий ряд інших

проектів, які стали самостійними в рамках системи проектів Apache Software Foundation.

Вважається однією з основоположних технологій «великих даних». Навколо Hadoop утворилася ціла екосистема з пов'язаних проектів і технологій, багато з яких розвивалися спочатку в рамках проекту, а згодом стали самостійними. З другої половини 2000-х років йде процес активної комерціалізації технології, кілька компаній будують бізнес цілком на створенні комерційних дистрибутивів Hadoop і послуг з технічної підтримки екосистеми, а практично всі великі постачальники інформаційних технологій для організацій в тому чи іншому вигляді включають Hadoop в продуктові стратегії і лінійки рішень.

HDFS

Розподілена Файлова Система Hadoop (HDFS) – це файлова система, призначена для зберігання файлів великих розмірів, по блоках розподілених між вузлами обчислювального кластера. Всі блоки в HDFS (крім останнього блоку файлу) мають однаковий розмір, і кожен блок може бути розміщений на декількох вузлах, розмір блоку і коефіцієнт реплікації (кількість вузлів, на яких повинен бути розміщений кожен блок) визначаються в налаштуваннях на рівні файлу. Завдяки реплікації забезпечується стійкість розподіленої системи до відмов окремих вузлів. Файли в HDFS можуть бути записані лише одного разу (модифікація не підтримується), а запис в файл в один час може вести тільки один процес. Організація файлів в просторі імен - традиційна ієрархічна: є кореневий каталог, підтримується вкладення каталогів, в одному каталозі можуть розташовуватися і файли, і інші каталоги.

HDFS стійка до багатьох проблем, які підпадають всі інші розподілені файлові системи:

- HDFS створена для зберігання величезного обсягу даних, тому дані розподілені між великим числом машин. Також HDFS підтримує куди більші розміри файлів, ніж інші файлові системи, наприклад NFS.

- HDFS забезпечує надійність зберігання даних: при несправності окремої машини в кластері, дані будуть доступні.

- HDFS забезпечує швидкий та масштабований доступ до інформації. HDFS обслуговує велику кількість споживачів, шляхом додавання додаткових машин в кластер.

- HDFS добре взаємодіє з Hadoop MapReduce, дозволяючи зчитувати і обробляти дані локально тоді, коли, це доступно та можливо.

HDFS є блочно-структурованої файлової системою: окремі файли розбиваються на блоки фіксованого розміру. Ці блоки зберігаються в кластері або в одній з машин з великим об'ємом для зберігання даних. Окремі машини в кластері називаються вузлами даних (DataNodes). Файл може складатися з декількох блоків, і вони необов'язково будуть зберігатися на одній і тій же машині; цільові машини, кожна з яких зберігає по блоку, вибираються випадково, за принципом «block-by-block».

У HDFS файли зберігаються як набір великих блоків, розподілених між декількома машинами, тому ці файли не є частиною звичайної файлової системи. HDFS використовує окремий простір імен, ізольоване від вмісту локальних файлів. Файли всередині HDFS зберігаються в окремому каталозі, керованому сервісом DataNode (рис. 1.2). Взаємодіяти з файлами HDFS за допомогою звичайних інструментів файлу в Linux (наприклад, ls, cp, mv і інші) не можна. Однак, в HDFS входить її власна утиліта управління, яка працює приблизно так само, як ці інструменти.

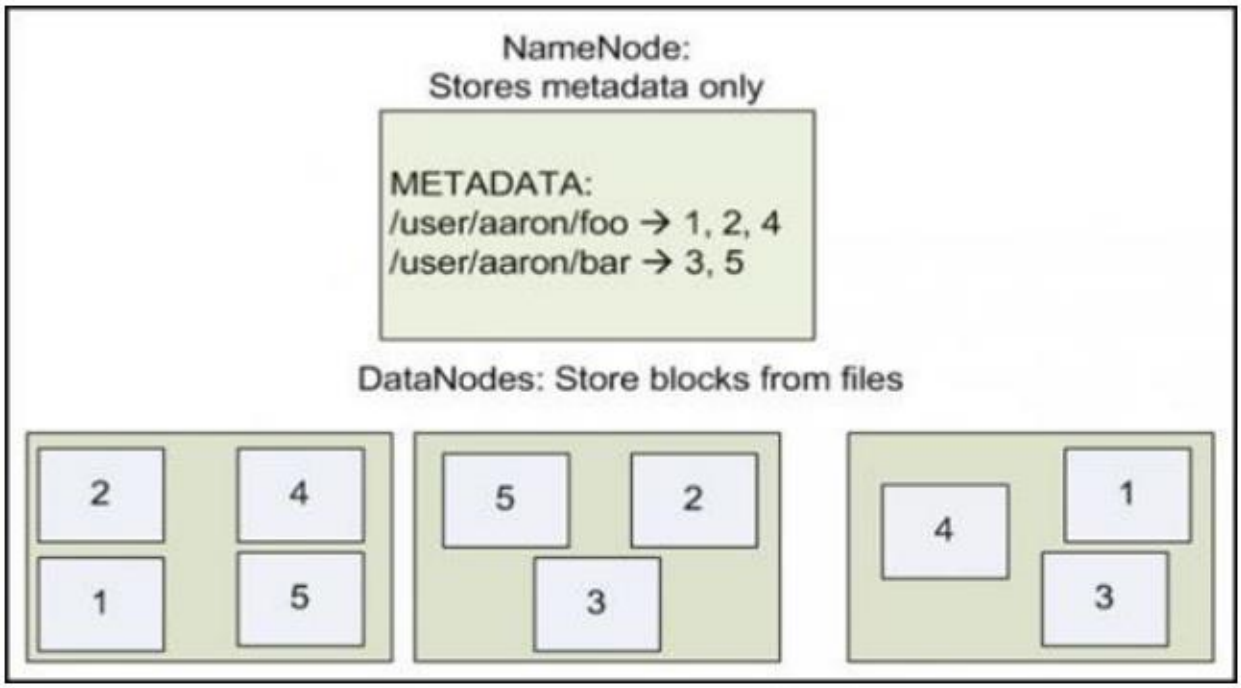


Рис. 1.2 Пример структуры HDFS

Структура метаданих в HDFS змінюється великою кількістю одночасних клієнтів. Для збереження її цілісності всі звернення обробляється однією машиною, званої NameNode. NameNode зберігає всі метадані файлової системи.

Щоб відкрити файл, клієнт звертається до NameNode і отримує список розташування блоків, складових файл. Ці адреси ідентифікують DataNode, який зберігає кожен блок. Потім клієнти зчитують інформацію безпосередньо з серверів DataNode, можливо, паралельно.

На відміну від більшості блочно-структурованих систем, де використовується розмір блоку близько 4 або 8 Кб, в HDFS розмір блоку за замовчуванням – 64 Мб. За рахунок цього в HDFS знижений обсяг сховища метаданих, необхідних для кожного файлу. Також це дозволяє прискорити потокове читання даних, оскільки великі обсяги інформації послідовно викладені на диску.

Обробка даних за допомогою Hadoop

Для обробки даних за допомогою Hadoop можна написати MR- програму, скориставшись streaming-інтерфейсом Hadoop або java MapReduce API. Також можна скористатися різними додатковими фреймворками, що встановлюються поверх Hadoop. Найбільш популярними з них є Apache Hive, Apache Pig, Cascading, Apache Spark. Фреймворки Apache Storm і Apache Spark також надають можливості по розподіленої обробки великих потоків даних в реальному часі. Більш докладний опис всіх підходів наведено нижче. Робота з даними в Hadoop може проводиться з використанням трьох двигунів: стандартного MapReduce, Apache Spark і Apache Tez.

YARN - модуль, що з'явився в 2013 р. і відповідає за управління ресурсами кластерів і планування завдань. Якщо в попередніх випусках ця функція була інтегрована в модуль MapReduce, де була реалізована єдиним компонентом (JobTracker), то в YARN функціонує логічно самостійний модуль - планувальник ресурсів, абстрагуючий усі обчислювальні ресурси кластера і керуючий їх наданням додатків розподіленої обробки. Працювати під керуванням YARN можуть як MapReduce-програми, так і будь-які інші розподілені додатки, що підтримують відповідні програмні інтерфейси.

YARN забезпечує можливість паралельного виконання декількох різних завдань в рамках кластера і їх ізоляцію (за принципами мультіарендності). Розробнику розподіленого додатка необхідно реалізувати спеціальний клас управління додатком (ApplicationMaster), який відповідає за координацію завдань в рамках тих ресурсів, які надасть планувальник ресурсів. Планувальник ресурсів же відповідає за створення екземплярів класу управління додатком і взаємодії з ним через відповідний мережевий протокол.

YARN може бути розглянутий як кластерна операційна система в тому сенсі, що виступає інтерфейсом між апаратними ресурсами кластера і широким класом додатків, що використовують його потужності для виконання обчислювальної обробки.

MapReduce

MR – це модель розподіленої обробки даних, запропонована компанією Google для обробки великих обсягів даних на комп'ютерних кластерах..

Робота MapReduce передбачає, що вихідні дані представлені в вигляді деяких записів.

Обробка даних відбувається в 3 стадії (рис. 2.3):

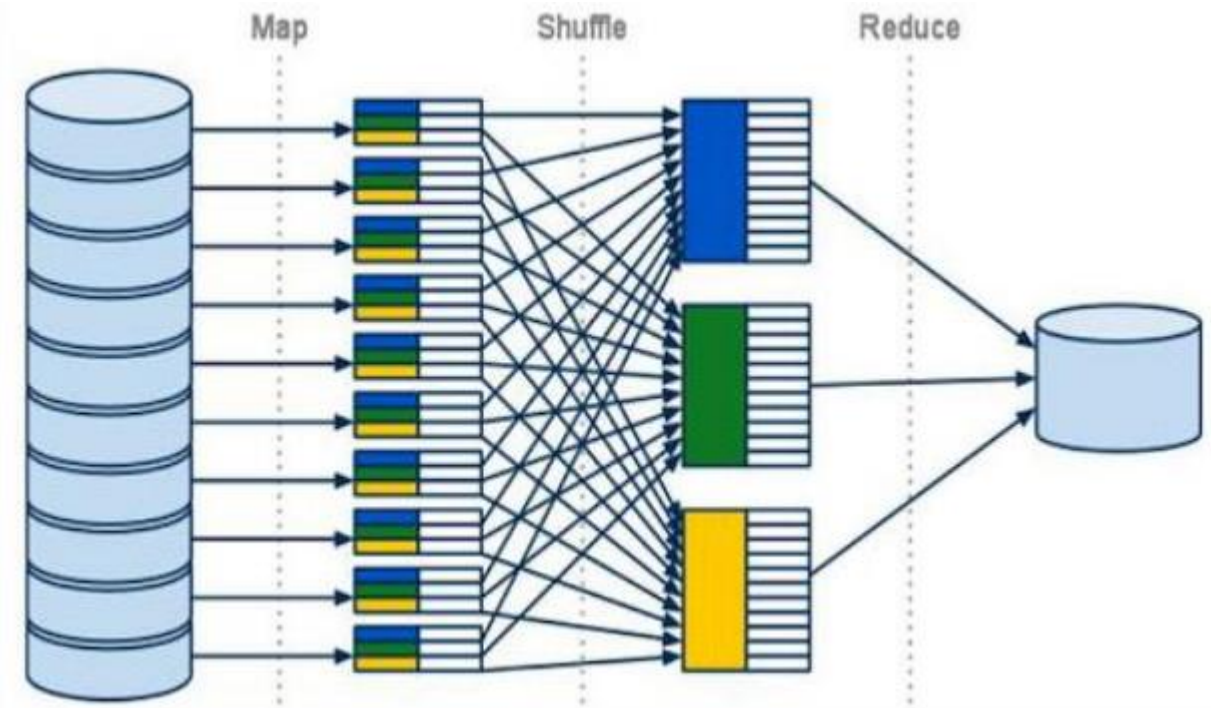


Рис. 2.3 Схема MR завдання

- Map. На стадії map відбувається попередня обробка вихідних даних за допомогою функції map, яка визначається користувачем. Робота цієї стадії полягає в попередній обробці і фільтрації даних.

- Функція map застосовується до однієї вхідний записи і видає безліч пар ключ-значення. Зміст ключа і значення визначається розробником, при цьому дані з однаковим ключем надалі потрапляють в один екземпляр функції reduce.

- Shuffle. Ця стадія проходить непомітно для розробника. В її процесі відбувається розбиття виведення функції map «по кошиках», де кожна корзина

відповідає одному ключу виведення стадії map. Ці кошики служать входом для стадії reduce.

• Reduce. На цій стадії кожна «кошик» із значеннями, сформована раніше на стадії shuffle, потрапляє на вхід функції reduce. Функція reduce визначається розробником і обчислює фінальний результат для кожної окремої «кошика». Отримане в результаті роботи функції reduce безліч значень є фінальним результатом MapReduce завдання.

Hadoop MapReduce дозволяє створювати завдання як з базовими обработчиками, так і зі згортками, написаними без використання Java.

Утиліти Hadoop streaming дозволяють використовувати в якості базових оброблювачів і згорток будь-який виконуваний файл, який працює зі стандартним вводом-висновком операційної системи (наприклад, утиліти командної оболонки UNIX), є також SWIG-сумісний прикладний інтерфейс програмування Hadoop pipes на C ++. Також, до складу дистрибутивів Hadoop входять реалізації різних конкретних базових оброблювачів і згорток, найбільш типово використовуваних в розподіленій обробці.

У перших версіях Hadoop MapReduce включав планувальник завдань (JobTracker), починаючи з версії 2.0 ця функція перенесена в YARN, і починаючи з цієї версії модуль Hadoop MapReduce реалізований поверх YARN. Програмні інтерфейси здебільшого збережені, однак повної сумісності немає (тобто для запуску програм, написаних для попередніх версій API, для роботи в YARN в загальному випадку потрібно їх модифікація або рефакторинг, і лише при деяких обмеженнях можливі варіанти зворотної двійкової сумісності.

Деякі властивості MapReduce:

• Всі запуски функцій map і reduce можуть працювати незалежно один від одного, отже, можливо їх паралельне виконання, в тому числі і на різних машинах кластера.

• Стадія Shuffle практично є паралельною сортуванням, а отже, також може працювати паралельно на різних машинах.

Два аспекти які забезпечують принцип горизонтальної масштабованості.

• Зазвичай функція map виконується на тій же машині, де зберігаються дані, що відповідає принципу локальності даних.

MR – роботу зручно проілюструвати на класичній Word Count задачі: для кожного слова, що зустрічається в великому корпусі документів, необхідно порахувати сумарну кількість таких слів в корпусі.

В даному випадку, один документ буде одними вхідними записом для MR – завдання, функція map перетворюватиме вхідний документ в набір пар, shuffle – перетворювати цей набір в пари, reduce - підсумовувати одиниці і повертати фінальний відповідь для кожного слова – кількість його входжень в корпус.

Взагалі кажучи, для деяких типів завдань (фільтрація даних, робота зі сховищами даних) можна обійтися тільки стадією map.

Крім того, при реалізації MR-завдання в hadoop, для попередньої агрегації результатів декількох map функцій можна визначити комбінує функцію (рис. 2.4). За своєю структурою комбінує функція схожа на reduce функцію - вона приймає на вхід вихід частини map функцій і видає для них агрегований результат. Найчастіше можна використовувати reduce функцію як комбінує. Відмінністю комбінує функції від reduce є те, що на неї потрапляють в повному обсязі значення, відповідні одному ключу. Використання combine функції може істотно збільшити швидкість виконання MR-завдання. Однак комбінує функція не може бути застосовна для всіх типів завдань (наприклад, для пошуку за медіанне значення по ключу, де необхідна попередня сортування всіх значень).

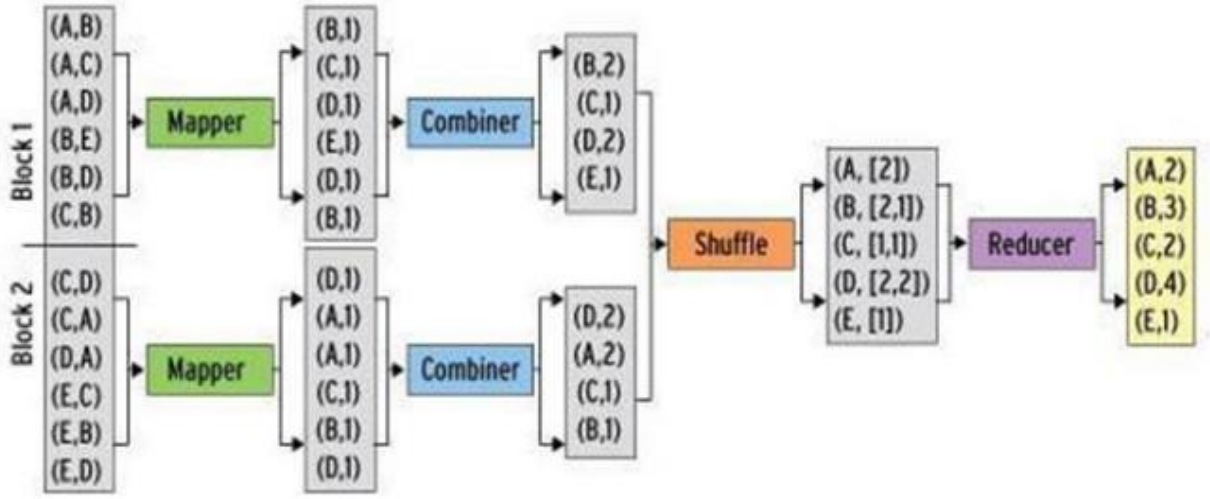


Рис. 2.4 Використання комбінує функції

Також варто відзначити, що до складу дистрибутивів Hadoop входять реалізації різних конкретних базових map і reduce функцій, найбільш часто використовуваних в розподіленої обробки.

Spark та TEZ движки.

Основним недоліком реалізації MapReduce в Hadoop є те, що в більшості реальних завдань не обійтися єдиною MR-роботою. Зазвичай при обробці даних виробляються кілька послідовних і паралельних операцій. Реалізація MR в Hadoop передбачає, що всі результати – як кінцеві, так і проміжні – записуються на диск. В результаті час роботи з диском може в кілька разів перевищувати час самих обчислень.

Для боротьби з цією проблемою розроблені два інших движка – Spark та Tez. Spark використовує спеціалізовані примітиви для рекуррентной обробки в оперативній пам'яті, завдяки чому дозволяє отримувати значний виграш в швидкості роботи для деяких класів задач. Tez представляє задачу у вигляді спрямованого ациклического графа (DAG) компонентів-обробників (рис. 2.5). Планувальник запускає обчислення графа і, при необхідності, має можливість динамічно переконфігурувати його, оптимізуючи під дані.

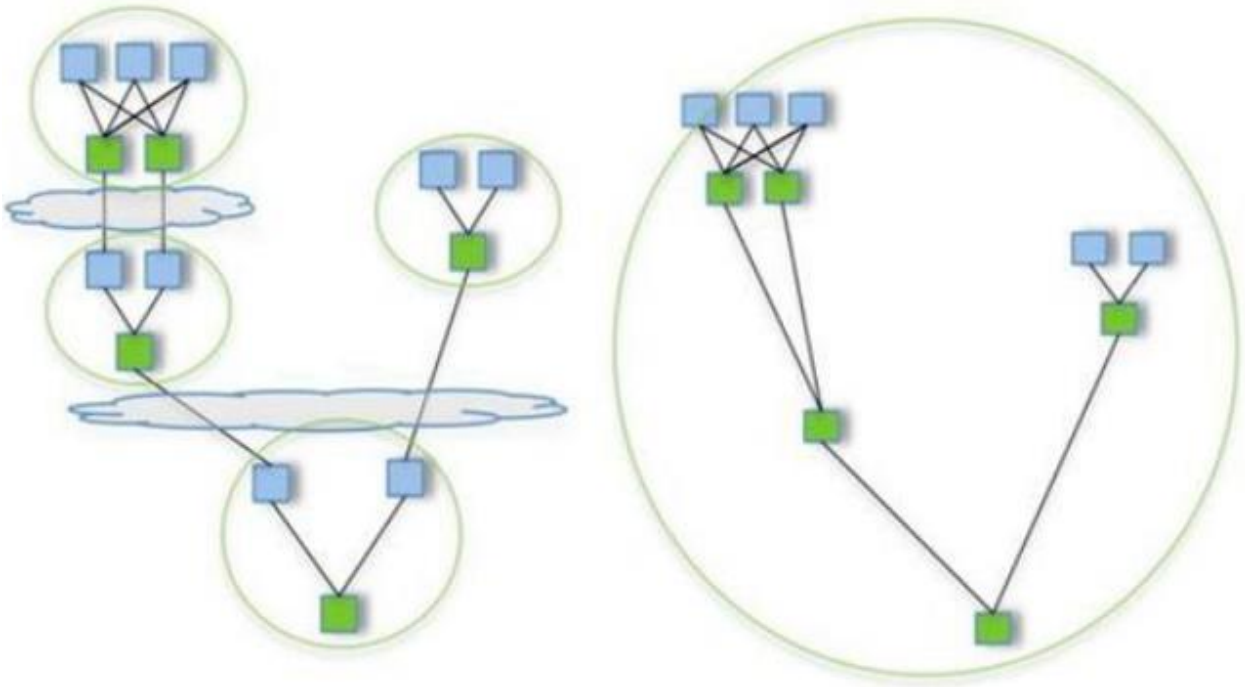


Рис. 2.5 Виконання завдання в режимах MR і Tez

Apache Hive

Apache Hive – це фреймворк, який дозволяє полегшити читання, запис і управління великими наборами даних в розподілених файлових системах, використовуючи SQL. Hive входить в екосистему Hadoop і дозволяє розробнику описувати процес обробки даних на більш високому абстрактному рівні, беручи на себе всю технічну реалізацію MR або Tez.

Hive надає стандартний SQL-функціонал, включаючи більшість нововведень 2003 і 2011 років. Функціонал Hive SQL може бути розширений за допомогою певних користувачем функцій (UDFs), агрегаторів (UDAFs) і табличних функцій (UDTFs).

Також Hive має стандартний набір конекторів для різних форматів файлів, таких як: текстові CSV файли, Apache Parquet, Apache ORC і інші. Крім того, розробник має можливість написання власних конекторів для будь-яких інших форматів файлів.

Apache Pig

Фреймворк Apache Pig створений для тих же цілей, що і Hive - підвищити рівень абстракції при обробці даних. Однак, основна його відмінність полягає в тому, що він заснований не на SQL, а на власному процедурному мовою Pig Latin. До складу Pig входить виконавче середовище для запуску сценаріїв Pig Latin. Сценарії можна запускати на локальній JVM або на кластері Hadoop в режимах MR і Tez.

Як і Hive, Apache Pig вміє працювати з різними типами файлів. При необхідності є можливість написання власних конекторів до будь-яким іншим форматам.

Робота в Pig проводиться зі спеціальними структурами даних: tuple і bag.

- Tuple - це упорядкований набір полів. До полям у tuple можна звертатися за індексом або імені.

- Bag - колекція Tuple.

До базових функцій Pig Latin відносяться: LOAD, STORE, GENERATE, JOIN, GROUP, FILTER, UNION, DISTINCT і ORDER. Також в Pig є набір обчислювальних (COUNT, MAX, MIN) та математичних (ABS, SIN, COS) функцій. Для розширення можливостей розробник може визначати власні UDF на шести мовах: Java, Jython, Python, JavaScript, Ruby і Groovy.

Приклад використання Pig (витяг з файлу, що містить записи виду id: password поля id, і запис результату в файл):

A = load 'passwd' using PigStorage (':'); - завантаження файлу 'passwd'

B = foreach A generate $ 0 as id; - виділення поля id store B into 'id.out'; - запис результату в файл 'id.out'

Основними перевагами Pig над Hive є простота написання більш складних вибірок, а також процедурний підхід, що веде до більшої впорядкованості - логіку

обробки даних можна розбивати на блоки і контролювати правильність роботи кожного блоку.

Cascading

Фреймворк, що застосовується для обробки даних доступним для Hadoop, є Cascading. За принципом роботи Cascading сильно нагадує Apache Pig – він також дозволяє управляти даними в Hadoop, абстрагуючись від написання низькорівневих MR-завдань. Однак, на відміну від Pig, обробка даних проводиться не за допомогою власної мови (як Pig Latin), а на java. У Cascading існують бібліотеки, що дозволяють виробляти стандартні операції над даними (угруповання, фільтрацію і інші). Розробник може визначати свої функції. Cascading надає більше можливостей для контролю за потоками даних, але при цьому вимагає більш низький рівень програмування, ніж Pig.

Apache Storm

Фреймворк Apache Storm є аналогом Apache Hadoop, тільки для обробки великих потоків даних в реальному часі.

Основними елементами Storm є:

- Tuple: Елемент представлення даних. За замовчуванням може містити поля: Long, Integer, Short, Byte, String, Double, Float, Boolean і byte. Також Tuple може містити Серіалізуемое призначені для користувача типи.

- Stream: Послідовність з tuple. Повинен містити схему іменування полів в tuple.

- Spout: Джерело даних для Stream. Spout отримує дані із зовнішніх джерел, формує з них tuple і відправляє в stream. Storm має готові spouts для більшості популярних систем обміну повідомленнями: RabbitMQ, Kestrel, JMS, Kafka.

- Bolt: Оброблювач даних. На вхід в bolt надходять tuple. На вихід кілька tuple.

• Sink: Фінальний bolt. Може записувати tuple в різні сховища даних (hdfs, mongodb, hbase і інші).

• Topology (топологія): Сукупність елементів (spout, bolt, sink) з описом їх взаємозв'язку. Здійснює передачу tuple між цими елементами.

Написання storm обробника даних полягає в конфігурації топології і, при необхідності, написанні власних обробників або джерел для даних. Топології виконуються за рахунок подання завдання у вигляді спрямованого антициклічного графа (DAG) компонентів-обробників.

Фреймворк storm надає масштабованість, гарантований захист від втрати даних і відновлення після збоїв.

Apache Spark

Apache Spark є фреймворком як для обробки потоків даних в реальному часі (аналог Storm), так і для обробки великих обсягів даних (аналог MapReduce в Hadoop). Останнім часом Spark набирає все більшої популярності.

Spark використовує спеціалізовані примітиви для рекурентної обробки даних в оперативній пам'яті і за рахунок цього на деяких завданнях показує значний виграш в швидкості роботи в порівнянні з використанням MapReduce. Spark може працювати як в середовищі кластера Hadoop під керуванням YARN, так і без компонентів ядра Hadoop. При цьому підтримується кілька розподілених систем зберігання, в тому числі і HDFS.

2.2. Візуалізація даних

Крім завдання обробки даних важливим є візуальне відображення результатів для їх аналізу.

Elasticsearch Kibana

Для цих візуалізації створені інструменти за типом Kibana. Kibana – це фреймворк з відкритим вихідним кодом для візуалізації великих обсягів даних, розташованих в Elasticsearch кластері. Elasticsearch – це пошуковий движок, який працює з json документами. Elasticsearch відповідає принципам горизонтальної масштабованості, тому добре підходить для роботи з великими обсягами даних.

Kibana володіє web-інтерфейсом, що дозволяє будувати різні типи діаграм і графіків, а також відображати зміни в реальному часі.

Mapbox

Для відображення даних на географічній карті може використовуватися платформа Mapbox. Mapbox дозволяє створювати кастомізовані карти і виводити на них необхідну інформацію за допомогою web-інтерфейсу або використовуючи API. Інформація для карти Mapbox береться з відкритих джерел, таких як OpenStreetMap і Nasa. На відміну від сервісів Google Maps і Yandex Maps, у Mapbox немає обмежень на кількість звернень до геокодеру протягом доби з одного ip-адреси.

2.3. Кластеризації даних

Кластеризація (кластерний аналіз) – задача розбиття заданої вибірки об'єктів на підмножини, які називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися (рис. 2.6). Головною відмінністю кластеризації від класифікації є те, що перелік груп (кластерів) спочатку не заданий і визначається безпосередньо в процесі роботи алгоритму. Синонімами терміна кластеризація є автоматична класифікація, навчання без учителя і таксономія.

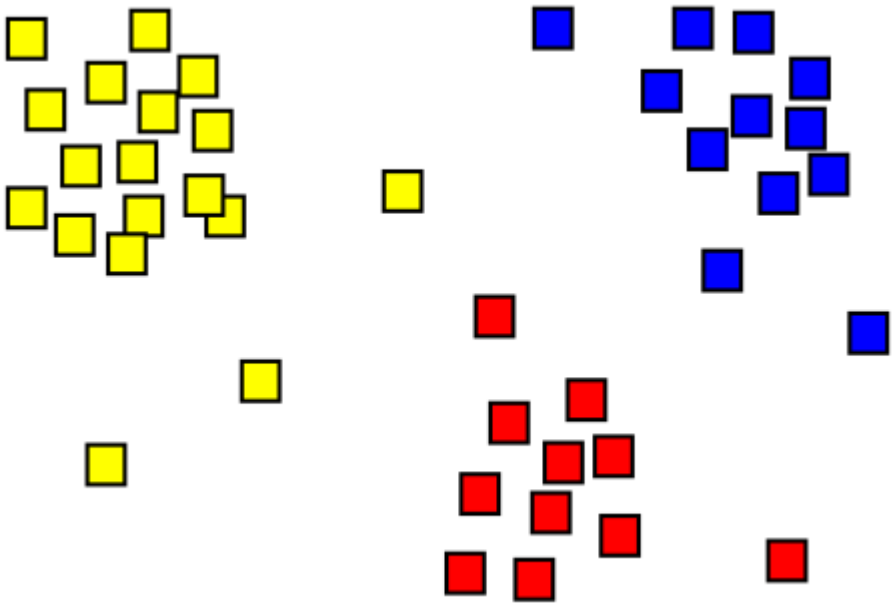


Рис. 2.6 Результат кластерного аналізу позначений розфарбуванням точок відповідно до приналежністю до одного з трьох кластерів.

Кластеризація застосовують для вирішення наступних завдань:

- Розробка типології або класифікації.
- Дослідження корисних концептуальних схем групування об'єктів.
- Породження гіпотез на основі дослідження даних.
- Перевірка гіпотез або визначення, чи дійсно деякі типи (групи) присутні в наявних даних.

У загальному вигляді кластерний аналіз складається з наступних етапів:

- Відбір вибірки об'єктів для кластеризації.
- Визначення безлічі змінних, за якими будуть оцінюватися об'єкти у вибірці. При необхідності застосовується нормалізація значень змінних.
- Обчислення значень міри схожості між об'єктами.

- Застосування методу кластерного аналізу для створення груп схожих об'єктів (кластерів).

- Представлення результатів кластерного аналізу.

Найчастіше, після отримання та аналізу результатів проводиться коригування обраної метрики і методу кластеризації до отримання оптимального результату.

У розробленій системі кластерний аналіз проводився за допомогою мови програмування R. R - це об'єктно-орієнтована мова програмування, створена для статистичної обробки даних і графічного представлення результатів. Мова R підтримує широкий спектр статистичних та чисельних методів, володіючи при цьому гарною розширюваністю за допомогою пакетів. В тому числі, R включає в себе набір різних методів кластеризації і можливість відображення даних.

2.4. Системи обробки інформації, яка надходить від абонентів.

Телекомунікаційні компанії в Україні поступово впроваджують у себе рішення з обчислення великих об’ємів даних. Наприклад, мобільні оператори надають абоненту можливість відслідковування даних, такі системи як «ЗнайДе» від Vodafone (МТС) або «Маячок» від Київстар, наразі оператор Lifecell не надає такої послуги. Ці послуги доступні для усіх абонентів за окрему плату. Back end таких послуг представляє собою спеціальний софт для аналізу та обчислення даних, що надходять від абонентів та відповідно потужності (сервери), що базуються на території оператора.

Нещодавно компанія Київстар придбала програму Deep Packet Inspection (DPI) за $20 млн, для аналізу інтернет трафіку, також цей софт використовують для боротьби з забороненим трафіком та кіберзагрозами. Крім того цей пакет програм буде застосовуватися для аналізу трафіку, розподілення навантаження. Як заявляє компанія Київстар система DPI допоможе рівномірно розподіляти

навантаження в мережі Київстар для абонентів але так само – це не завадить відслідковувати абонентів в мережі.

2.5. Технологія накопичення статистичних даних (DPI)

Система Deep Packet Inspection DPI – технологія накопичення статистичних даних, перевірки і фільтрації мережевих пакетів по їх вмісту. На відміну від брандмауерів, Deep Packet Inspection аналізує не тільки заголовки пакетів, а й повне вміст трафіку на рівнях моделі OSI з другого і вище (Рис. 2.7.). Deep Packet Inspection здатне виявляти і блокувати віруси, фільтрувати інформацію, що не задовольняє заданим критеріям.

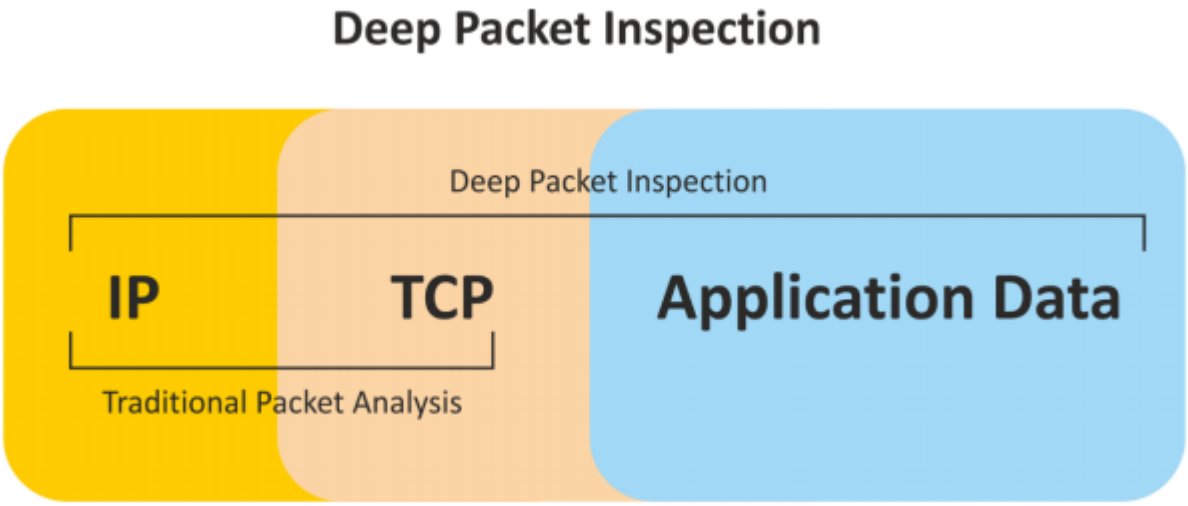


Рис. 2.7. Трафік, що аналізується DPI

Deep Packet Inspection може приймати рішення не тільки по вмісту пакетів, але і за непрямими ознаками, властивим якимось певним мережевим програмами і протоколам. Для цього може використовуватися статистичний аналіз (наприклад статистичний аналіз частоти певних символів, довжини пакета).

Deep Packet Inspection часто використовуються провайдерами для контролю трафіку, а іноді і для блокування деяких протоколів, таких як BitTorrent. За допомогою Deep Packet Inspection можна визначити, яка програма згенерувало або отримує дані, і на підставі цього зробити будь-яку дію. Крім блокування, Deep Packet Inspection може збирати детальну статистику з'єднання кожного

користувача окремо. Також, за допомогою quality of service Deep Packet Inspection може управляти швидкістю передачі окремих пакетів, піднявши її або, навпаки, зменшивши. На думку деяких Інтернет-провайдерів, Deep Packet Inspection дозволяє стримувати додатки, що забивають Інтернет-канал, змінювати пріоритети передачі різних типів даних, наприклад, прискорюючи відкриття Інтернет сторінок за рахунок зменшення швидкості завантаження великих файлів. Крім того, Deep Packet Inspection, згідно бездоказовим заявами виробників, нібито здатне виявляти серед загального потоку трафіку шматочки, відповідні комп'ютерним вірусам і нібито блокувати їх, підвищуючи, таким чином, безпеку мережі. Іноді Deep Packet Inspection використовується в великих корпораціях для запобігання випадкових витоків даних, а також для захисту від відправки по e-mail внутрішніх захищених файлів.

Приклад роботи Deep Packet Inspection

Ідентифікація протоколу транспортного рівня мережевої моделі OSI.

У структурі пакета протоколу IPv4 виділений спеціальний байт для вказівки номера протоколу транспортного рівня. Їм є десятий байт від початку заголовка IPv4 пакета. Наприклад: номер дорівнює шести – для TCP протоколу, номер дорівнює сімнадцяти – для UDP протоколу.

У структурі пакета IPv6 також існує спеціальна область, в якій знаходиться аналогічний ідентифікатор протоколу транспортного рівня. Ця область називається Next Header.

Ідентифікація HTTP

Для ідентифікації HTTP протоколу досить перевірити, що пакет є TCP, і вміст цього TCP пакету починається з однієї з наступних команд: «GET», «POST», «HEAD». Крім того, після команди повинен стояти пробіл, а також через деякий проміжок повинен зустрітися текст «HTTP /». Якщо все це виконується, то цей пакет несе в собі HTTP запит.

Ідентифікація RTSP

Для того, щоб виявити серед всього трафіку пакети RTSP, досить переконатися, що пакет є TCP і вміст цього TCP пакету починається з однієї з наступних команд: «OPTIONS», «DESCRIBE», «ANNOUNCE», «PLAY», «SETUP», «GET_PARAMETER», «SET_PARAMETER», «TEARDOWN». Після команди повинен стояти пробіл. Також, через деякий проміжок повинен зустрітися текст «RTSP /».

Реалізація QoS

З точки зору експлуатації, оператор може контролювати утилізацію підключених через DPI каналів на рівні додатків. Раніше він вирішував завдання реалізації QoS (Quality of Service) виключно засобами побудови черг на підставі маркування трафіку службовими бітами в заголовках IP, 802.1q і MPLS, виділяючи найбільш пріоритетний трафік (різного роду VPN'и, IPTV, SIP та т. Д.) і гарантуючи йому певну пропускну здатність в будь-який момент часу. Трафік типу Best Effort, до якого належить весь інтернет трафік домашніх абонентів (HSI – High Speed Internet), залишався фактично без контролю, що давало можливість того ж Bittorrent забрати собі всю вільну смугу, що, в свою чергу, вело до деградації будь-яких інших веб-додатків. З використанням DPI у оператора з'являється можливість розподілити канал між різними додатками. Наприклад, в нічні години дозволити трафіку Bittorrent забирати собі більше смуги, ніж вдень, в години-пік, коли в мережі ходить велика кількість іншого веб-трафіку. Інша популярна міра у багатьох мобільних операторів – блокування Skype-трафика, а також будь-яких видів SIP-телефонії. Замість повного блокування оператор може вирішувати роботи даних протоколів, але на дуже низькій швидкості з відповідною деградацією якості надання сервісу у конкретного додатка, щоб змусити користувача платити за послуги традиційної телефонії, або за спеціальний пакет послуг, що дозволяє доступ до VoIP-сервісів.

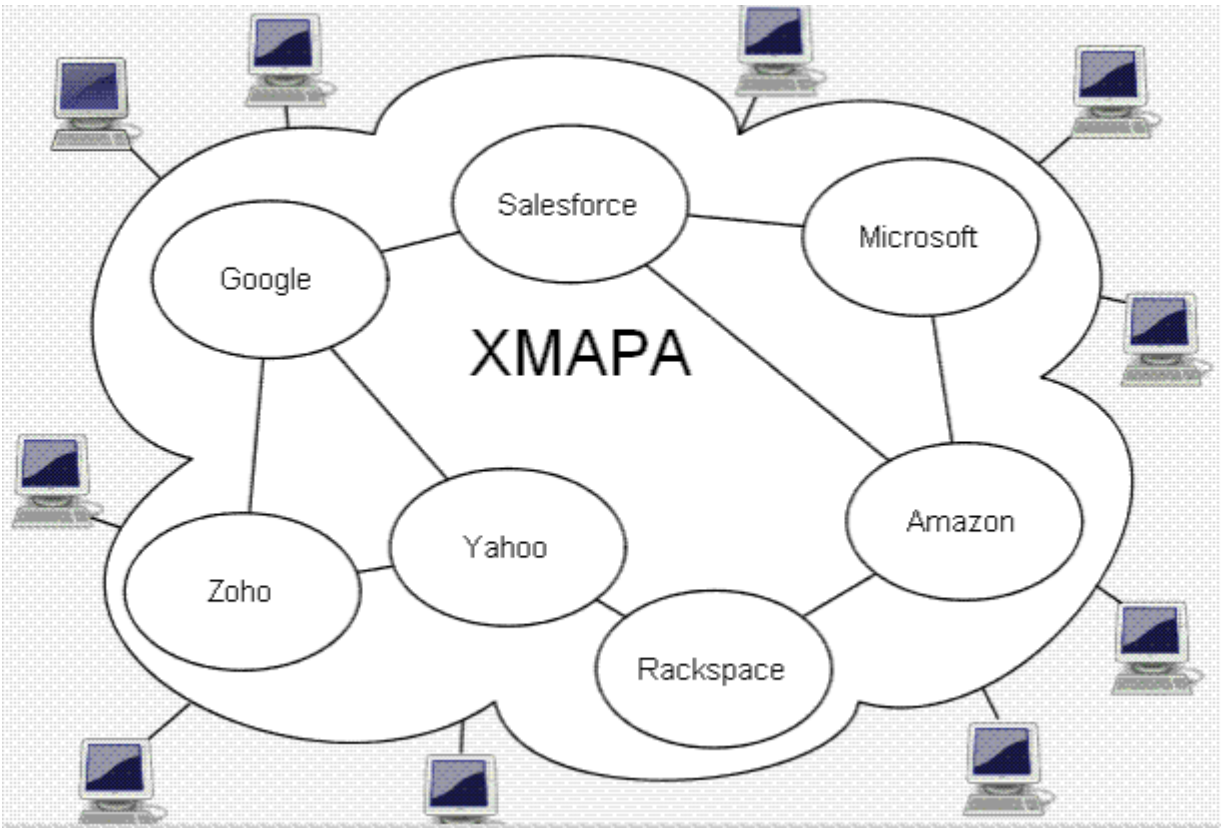
Використання в Україні, відомо, що на даний момент, цю програмну реалізацію використовує оператор мобільного зв’язку Київстар та деякі інтернет провайдери але використання цієї технології не вирішує проблему зберігання даних та подальших маніпуляцій з ними, а навпаки збільшує їх надходження, а подальше зберігання даних випадає на мобільного оператора, якщо дані конфіденціальні, то їх обробкою та зберіганням варто займатися мобільному оператору але для неконфіденціальні даних є можливість їх перенесення та зберігання на стороні хмарних сервісів (безсерверних технологій).

Висновок: У даному розділі розглянуто питання обробки даних існуючі методи та технології обробки даних Big Data, системи обробки/аналізу великих даних операторами в Україні та системи відслідковування та аналізу трафіку абонентів.

3. Технології хмарних обчислень

3.1. Сучасні хмарні обчислення.

Вікіпедія дає таке визначення хмарних обчислень. Хмарні обчислення (англ. Cloud computing) – технологія розподіленої обробки даних, в якій комп'ютерні ресурси і потужності надаються користувачеві як Інтернет-сервіс. Надання користувачеві послуг як Інтернет-сервіс є ключовим. Однак під Інтернет-сервісом не варто розуміти доступ до сервісу тільки через Інтернет, він може здійснюватися також і через звичайну локальну мережу з використанням веб-технологій.



Діаграма, що представляє «хмару»

Я видно з визначення та історії розвитку хмарних технологій, основне, що вплинуло на розвиток технологій стало створення таких гігантів як Google, Amazon, Microsoft, а також технічний прогрес, що говорить про те, що поява та подальший розвиток технологій є лише питанням часу. Основні чинники які вплинули на доступність технології:

1) Розвиток процесорів та технології літографії:

- збільшення продуктивності на ядро, при збереженні розмірів обладнання;
- зниження енергоспоживання хмарної системи;
- зниження тепловиділення;
- зниження вартості обладнання, а як наслідок експлуатаційних витрат.

2) Збільшення розмірів дисків для зберігання інформації та відповідно зниження вартості зберігання 1 Мб інформації дозволило:

- «безмежно» збільшити обсяги інформації, що доступні користувачам;
- знизити вартість обслуговування сховищ інформації, відповідно значно збільшивши обсяги даних;
- збільшити відмовостійкість (збільшення надійності носіїв).

3) Розвиток технології програмування з розподіленням потоків призвело до:

- більш ефективного використанню багатопроцесорних систем та застосуванню усіх можливостей операційних систем;
- розподіл навантаження на різні обчислювальні потужності хмар, наприклад за допомогою алгоритму «балансеру».

4) Розвиток технологій віртуалізації призвів до:

- програмне забезпечення дозволяє створювати віртуальну інфраструктуру не залежно від кількості наданих апаратних ресурсів;
- легкість клонування, масштабування та нарощування нових систем;
- зменшення витрат на адміністрування хмарних систем;
- доступність віртуальної інфраструктури через мережу Інтернет (підключення за допомогою засобів SSH, HTTPS та інше).

5) Збільшення пропускної здатності призвело до:

- покращення роботи з віртуальним інтерфейсом;
- збільшення швидкості роботи та відклику систем та віртуальних носіїв;
- зниження ціна на Інтернет, що дозволяє відгружати великі обсяги інформації;
- доступність для населення, багато компаній пропонують пробний період користування послугами хмарних обчислень для населення.

Перераховані вище фактори дозволили створити конкуренцію звичайним серверним обчисленням та поширити хмарні обчислення на маси.

Переваги хмарних обчислень:

a) доступність – за допомогою Інтернета користувач з будь-якої точки світу та будь-якого комп’ютера може отримати доступ до своєї хмарної інфраструктури, не має необхідності виїжджати в офіс або в серверну для проведення налаштувань, оновлення, ви просто заходите на сервіс і користуєтеся послугами за які заплатили тобто за фактичне використання;

b) низька вартість - основні чинники знизили вартість використання хмар наступні:

- зниження витрат на обслуговування віртуальної інфраструктури, викликане розвитком технологій віртуалізації, за рахунок чого потрібен менший штат для обслуговування всієї IT інфраструктури підприємства;
- оплата фактичного використання ресурсів, користувач хмари платить за фактичне використання обчислювальних потужностей хмари, що дозволяє йому ефективно розподіляти свої грошові кошти. Це дозволяє користувачам (підприємствам) економити на покупці ліцензій до ПЗ;
- використання хмари на правах оренди дозволяє користувачам знизити витрати на закупівлю дорогого устаткування, і зробити акцент на вкладення грошових коштів на наладку бізнес процесів підприємства, що в свою чергу дозволяє легко почати бізнес;
- розвиток апаратної частини обчислювальних систем, в зв'язку з чим зниження вартості обладнання.
- гнучкість - необмеженість обчислювальних ресурсів (пам'ять, процесор, диски), за рахунок використання систем віртуалізації, процес масштабування і адміністрування "хмар" ставати досить легким завданням, так як «хмара» самостійно може надати вам ресурси, які вам необхідні, а ви платите тільки за фактичне їх використання.

- надійність - надійність «хмар», особливо що знаходяться в спеціально обладнаних ЦОД, дуже висока так, як такі ЦОД мають резервні джерела живлення, охорону, професійних працівників, регулярне резервування даних, високу пропускну здатність Інтернет каналу, висока стійкість до DDOS атак.
- безпека - «хмарні» сервіси мають досить високу безпеку при належному її забезпеченні, однак при недбалому ставленні ефект може бути повністю протилежним.
- великі обчислювальні потужності - ви як пользовательє «хмарної» системи можете використовувати всі її обчислювальні можливості, заплативши тільки за фактичний час використання. Підприємства можуть використовувати цю можливість для аналізу великих обсягів даних.

c) постійне з'єднання з мережею - для отримання доступу до послуг «хмари» необхідно постійне з'єднання з мережею Інтернет. Однак у наш час це не такий і великий недолік особливо з приходом технологій стільникового зв'язку 3G і 4G.

d) програмне забезпечення та його кастомізація - є обмеження ПО яке можна розгортати на «хмарах» і надавати його користувачеві. Користувач ПО має обмеження в використовуваному ПО і іноді не має можливості налаштувати його під свої власні цілі.

e) конфіденційність - конфіденційність даних, що зберігаються на публічних «хмарах» в даний момент викликає багато суперечок, але в більшості випадків експерти сходяться в тому, що не рекомендується зберігати найбільш цінні для компанії документи на публічній "хмарі", так як в даний час немає технології яка б гарантувала 100% конфіденційність даних, що зберігаються.

f) надійність – що стосується надійності інформації, що зберігається, то з упевненістю можна сказати що якщо ви втратили інформацію збережену в "хмарі", то ви її втратили назавжди.

g) безпека - "хмара" саме по собі є досить надійною системою, однак при проникненні на нього зловмисник отримує доступ до величезного сховища даних. Ще один мінус це використання систем віртуалізації, в яких в якості гипервизора використовуються ядра стандартні ОС такі, як Linux, Windows і ін., Що дозволяє використовувати віруси.

h) вартість обладнання - для побудови власного хмари компанії необхідно виділити значні матеріальні ресурси, що не вигідно щойно створеним і малим компаніям.

Види послуг що надаються хмарними системами:

Що стосується послуг, що надаються, то в даний час концепція хмарних обчислень передбачає надання наступних типів послуг своїм користувачам:

- все як послуга (Everything as a Service); При такому вигляді сервісу користувачеві буде надано все від програмно апаратної частини і до управлінням бізнес процесами, включаючи взаємодію між користувачами, від користувача вимагається тільки наявність доступу в мережу Інтернет. На мій погляд, даний вид сервісу це більш загальне поняття по відношенню до нижченаведеними послуг, які є більш приватними випадками.

- інфраструктура як послуга (Infrastructure as a service); Користувачеві надається комп'ютерна інфраструктура, зазвичай віртуальні платформи (комп'ютери) пов'язані в мережу. Які він самостійно налаштовує під власні цілі.

- платформа як послуга (Platform as a service); Користувачеві надається комп'ютерна платформа, зі встановленою операційною системою можливо і з програмним.

- програмне забезпечення як послуга (Software as a service); Даний вид послуги зазвичай позиціонується як «програмне забезпечення на вимогу», це програмне забезпечення розгорнуте на віддалених серверах і користувач може отримувати до нього доступ за допомогою Інтернету, причому всі питання

оновлення та ліцензій на дане програмне забезпечення регулюється постачальником даної послуги. Оплата в даному випадку проводитися за фактичне використання програмного забезпечення.

- апаратне забезпечення як послуга (Hardware as a Service); В даному випадку користувачеві послуги надається обладнання, на правах оренди яке він може використовувати для власних цілей. Даний варіант дозволяє економити на обслуговуванні даного обладнання, хоча за своєю суттю мало чим відрізняється від виду послуги «Інфраструктура як сервіс» за винятком того що ви маєте голе обладнання на основі якого розвертаєте свою власну інфраструктуру з використанням найбільш підходящого програмного забезпечення.

- робоче місце як послуга (Workplace as a Service); В даному випадку компанія використовує хмарні обчислення для організації робочих місць своїх співробітників, налаштувавши і встановивши все необхідне програмне забезпечення, необхідне для роботи персоналу.

- дані як послуга (Data as a Service); Основна ідея даного виду послуги полягає в тому, що користувачеві надається дисковий простір, яке він може використовувати для зберігання великих обсягів інформації.

- безпеку як сервіс (Security as a Service). Даний вид послуги надає можливість користувачам швидко розгортати, продукти дозволяють забезпечити безпечне використання веб-технологій, безпеку електронного листування, а також безпеку локальної системи, що дозволяє користувачам даного сервісу економити на розгортанні та підтримці своєї власної системи безпеки.

3.2. Класифікація хмарних сервісів:

В даний час виділяють три категорії «хмар»:

1) Публічні;

2) Приватні;

3) Гібридні

Публічне хмара – це IT-інфраструктура використовується одночасно безліччю компаній і сервісів. Користувачі даних хмар не мають можливості управлять і обслуговувати дане хмара, вся відповідальність з цих питань покладено на власника даного хмари. Абонентом пропонованих сервісів може стати будь-яка компанія і індивідуальний користувач. Вони пропонують легкий і доступний за ціною спосіб розгортання веб-сайтів або бізнес-систем, з великими можливостями масштабування, які в інших рішеннях були б недоступні. Приклади: онлайн сервіси Amazon EC2 і Simple Storage Service (S3), Google Apps/Docs, Salesforce.com, Microsoft Office Web.

Приватне хмара – це безпечна IT-інфраструктура, контрольована і експлуатована в інтересах однієї-єдиної організації. Організація може керувати приватним хмарою самостійно або доручити це завдання зовнішньому підряднику. Інфраструктура може розміщуватися або в приміщеннях замовника, або у зовнішнього оператора, або частково у замовника і частково у оператора. Ідеальний варіант приватного хмари це хмара розгорнуте на території організації, що обслуговує і контрольоване її співробітниками.

Гібридна хмара – це IT інфраструктура використовує кращі якості публічного і приватного хмари, при вирішенні поставленого завдання. Часто такий тип хмар використовується, коли організація має сезонні періоди активності, іншими словами, як тільки внутрішня IT-інфраструктура не справляється з поточними завданнями, частина потужностей перекидається на публічне хмара (наприклад великі обсяги статистичної інформації, які в необробленому вигляді не уявляють цінності для підприємства), а також для надання доступу користувачам до ресурсів підприємства (до приватного хмари) через публічне хмара.

3.3. Огляд сучасних технологій хмарних сервісів

Google Cloud Platform – це набір різних хмарних технологій, що базується на платформі Google, які працюють на тій самі інфраструктурі, що й сервіси Google такі як Google Search та YouTube. Крім інструментів керування також представлені модульні (готові рішення) хмарних служб: хмарні обчислення, зберігання даних, аналіз даних та машинне навчання. Компанія предаставляє можливості навчання для нових клієнтів на платформі Coursera та можливість використовувати пільговий (тестовий) період користування.

Основні сервіси, що входять до цієї платформи:

- App Engine – платформа як послуга для хостингу додатків.
- BigQuery – інфраструктура як послуга, що масштабується аналітика для баз даних.
- BigTable - інфраструктура к послуга, що масштабується NoSQL база даних.
- Cloud AutoML – набір продуктів для машинного навчання, які дозволяє розробникам з обмеженим досвідом роботи в області машинного навчання використовувати технології навчання і створення нейронних мереж.
- Cloud Datastore – документоорієнтована хмарна база даних.
- Cloud Pub/Sub – послуга для публікації і підписки на потоки даних і повідомлення. Додатки можуть обмінюватися даними через публікацію/підписку, без прямого обміну повідомленнями .
- Compute Engine – інфраструктура як послуга, надає віртуальні машини.
- Kubernetes Engine – система автоматичного розгортання, масштабування і управління додатків в контейнерах для Kubernetes.
- Google Genomics – аналіз геномів в хмарі.
- Google Video Intelligence – аналіз відеоконтенту.

- Cloud Vision – аналог пакету від MS Visio.

- Storage – інфраструктура як послуга, надає онлайн REST – доступ до файлів і змістом сховищ даних.

Amazon Web Services або AWS – продукт компанії Amazon, що надає платформу хмарних обчислень в оренду приватним особам та компанія. Існує безкоштона підписка на перші 12 місяців але з значно обмеженими функціональними можливостями. Технологія базується на серверних кластерах (фермах), що розташовані по всьому світу. На відміну від Google Cloud Platform, платформа AWS більш орієнтується на продаж послуг великим компаніям та корпоративним клієнтам.

Список продуктів поділяється на такі сервіси:

- Обчислення
- Мережа
- До правлення вмісту
- Контакт центр
- Зберігання даних та до правлення вмісту
- База даних
- Розгортання
- Менеджмент
- Служби адміністрування
- Аналітика
- Різне

Загалом кількість служб, що надає компанія значно більша в порівнянні з Google Cloud Platform але кожна зі служб є спрямованою на вузьке коло задач та має менш масштабовані можливості.

Microsoft Azure – хмарна платформа від компанії Microsoft. Аналогічний проект який створювався на основі розроблених принципів та технологій, наймолодший з представлених проектів. Має велику підтримку різних мов програмування та гнучке налаштування сервісів також має сертифікати інформаційної безпеки європейського стандарту. Платформа більш орієнтується на корпоративних клієнтів. Платформа більш орієнтована на повну інтеграцію з різними сервісами компанії.

Основні сервіси:

- Інфраструктурні сервіси:
  - Обчислення
  - Зберігання
  - Робота в мережі
- Платформені сервіси
  - Мобільні сервіси
  - Керування даними
  - Обмін повідомленнями
  - Медіа-сервіси
  - Машинне навчання
  - IoT
  - Azure Blockchain Workbench
  - Сервіси для розробників

3.4. Вибір хмарних технологій порівнюючи основні сервіси

Основні критерії для вибору сервісу хмарних технологій: доступність різних технологій, легкість його використання та вартість. Аналізувати будемо відповідно системи Google Cloud Platform, Amazon Web Services, Microsoft Azure.

Порівнюючи основні сервіси представлені цими компаніями та ті, що можуть використовуватися у розробках в мережі мобільного оператора табл 3.1., можна зробити висновок, що вони майже повністю повторюють один одного.

Таблиця 3.1.

Google Cloud Platform	**Amazon Web Services**	**Microsoft Azure**	**Примітка**
Google Compute Engine	Amazon EC2	Azure Virtual Machines	обчислювальні потужності
Google App Engine	AWS Elastic Beanstalk	Azure Cloud Services	хостинг інтернет-додатків
Google Kubernetes Engine	Amazon EC2 Container Service	Azure Container Service	хостинг контейнерних додатків
Google Cloud Functions	AWS Lambda	Azure Functions	бессерверні обчислення
Google BigQuery	Amazon Redshift	Microsoft Azure SQL Database	SQL бази даних
Google Cloud Bigtable, Google Cloud Datastore	Amazon DynamoDB	Azure Cosmos DB	NoSQL бази даних
Google Cloud Storage	Amazon S3	Azure Blob Storage	Сховища слабоструктурованих даних

Отже, порівняння за цим критерієм не може дати точну відповідь яку технологію використовувати.

Скористаємося порівнянням актуальних цін на технології табл. 3.2. та відповідно курсів для отримання базових навичок (їх також можна вважати вартістю послуги, оскільки навчанням співробітників має займатися компаній замовник послуг):

Таблиця 3.2.

Machine Type	AWS	Azure	GCP
Пробний період	AWS Пропонує базовий набір сервісів на 12 місяців	Azure пропонує 200$ кредиту на будь-який сервіс на 1 місяць	GCP пропонує кредит 300$ на 12 місяців
Малі віртуальні машини	Що стосується AWS, то базовий екземпляр, який включає 2 віртуальних процесора та 8 ГБ оперативної пам'яті та обійдеться приблизно в 69 доларів США на місяць відповідно до офіційного сайту.	Подібний екземпляр з 2 віртуальними процесорами і 8 ГБ оперативної пам'яті в Azure обійдеться вам приблизно в 70 доларів США на місяць.	Порівняно з AWS, GCP надає екземпляр, що містить 2 віртуальні процесори та 8 ГБ оперативної це обійдеться приблизно в 52 долари США / місяць.
Найбільший об'єм віртуальних машин	Найбільший екземпляр, який пропонує AWS, який включає 3,84 ТБ оперативної пам'яті та 128 віртуальних процесорів, коштуватиме	Найбільший екземпляр, який пропонує Azure, включає 3,89 ТБ оперативної пам'яті та 128 віртуальних процесорів.	GCP займає лідируючі позиції з найбільшим екземпляром, який включає 3,75 ТБ оперативної пам'яті та 160 вКПУ. Це обійдеться
	приблизно 3,97 дол. США / годину.	Коштує близько 6,79 дол. / Годину.	приблизно в 5,32 дол. США / годину.

Як видно з таблиці, найбільш цікавим з параметрів вартості є GCP та AWS але GCP пропонує дешевший мінімальний комплект, та найбільші з можливих

обчислювальні потужності, а також найбільшу кількість документації по своїм сервісам. Звичайно для більш глибшого порівняння систем необхідно мати доступ до усіх її параметрів та провести більш детальний аналіз вартості усіх служб. Для проекту демонстрації та для компанії які ніколи не використовували хмарні технології GCP пропонує найлегший старт у цій сфері (найбільша спільнота та майже всі курси є безкоштовними та у відкритому доступі), отже оптимальним вибором для початкових проектів є Google Cloud Platform.

Висновок: Розглянуто комплексні рішення у вигляді безсерверних (хмарних) технологій, класифікація таких рішень, огляд існуючих сервісів. Також обрано найоптимальнішу систему хмарних обчислень.

4. Створення та застосування системи обчислень на практиці

4.1. Створення системи обчислення за допомогою хмарних технологій

На основі розглянутих технологій безсерверних обчислень та методів Big Data, можна зробити висновок, що мобільні оператори будуть поступово переходити, використовувати такі технології, мережі LTE та наступних поколінь розробляються з упором на хмарні технології та їх можливості, отже маючи певні дані від абонентів, що отримує оператор, можна розробити систему на основі хмарних технологій та впровадити її в інфраструктуру оператора, дана розробка покликана в першу чергу показати перспективи та зручність використання таких технологій телекомунікаційними компаніями.

Предметом розробки є система яка дозволяє, передавати та зберігати отриманні дані від користувачів з можливістю одночасної фільтрації за певними параметрами, оператору мобільного зв’язку без використання власних серверів. Також «опціонально» оператор може надавати доступ до цих даних або до статистики іншим компаніям.

Вхідні дані представляють собою набір записів у форматі CSV, що містить згенерований дані мобільного оператора, (оскільки такі дані є конфіденційними та оператор їх не передає третім особам тому використовуємо згенеровані дані) наприклад для визначення навантаження (використання даних) на трасі між Києвом та Чопом.

Результат роботи має представляти собою автоматизовану систему, що виконує з’єднання з хмарними сервісами, створення потоку даних та передачу їх безпосередньо на хмарний сервер зберігання з можливістю виділення за основними критеріями та перегляду потоку даних в реальному часі, а також можливістю створити доступне посилання для інших компаній.

Система повинна відповідати наступним вимогам:

1) Система повинна відповідати принципу горизонтальної масштабованості, оскільки передбачається обробка великих обсягів даних, система повинна бути легко розширювана;

2) Система має обробляти дані з CSV файлу та передавати їх на хмарний сервер;

3) Система має мати змогу вибору записів відповідно до певних характеристик;

4) Передача основних даних за допомогою посилання, без можливості їх зміни.

4.2. Архітектура системи

До системи входить рис. 4.1.:

1) Генератор даних представляє собою мобільного оператора як користувача, тобто дані від мобільного оператора передаються на систему хмарних обчислень. В реально житті – це можуть бути дані від абонентів, навантаження трафіку на певному шляху тощо.

2) Процес відправки процес відправки можна об’єднати разом із Dataflow, являє собою процес передачі даних на систему хмарних обчислень, з використанням спеціально розроблених та створених служб.

3) Data storage – персональне сховище даних клієнта на сервері хмарних служб.

4) Other API – інші прикладні сервіси, які може використовувати клієнт для подальшої обробки та аналізу даних.

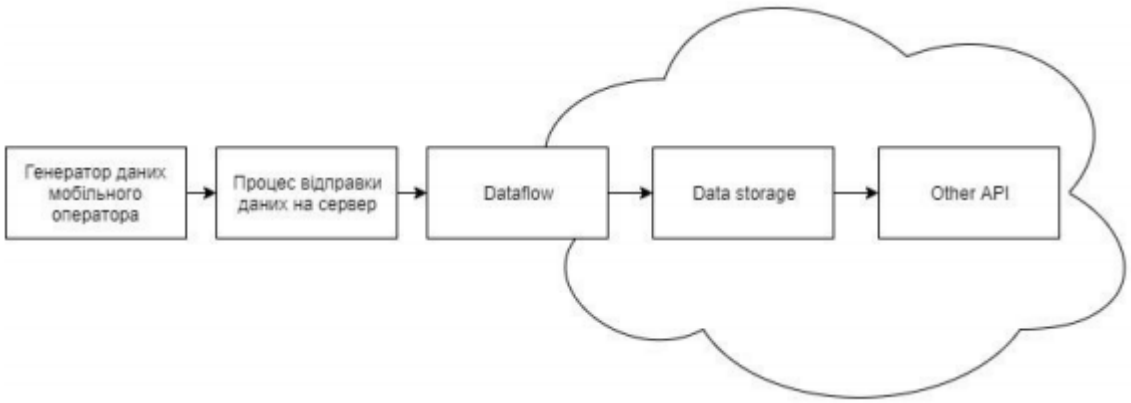


Рис 4.1. Приклад використання системи хмарних обчислень мобільним оператором

4.3. Реалізація системи

У минулому розділі обрано найоптимальніший варіант хмарних систем, яким є Google Cloud Platform. Для вирішеня поставленої задачі, необхідно створити дві віртуальні машини, одна з яких виступає у ролі генератора трафіку, а на іншу іде запис цього трафіку, звичайно, у реальній ситуації роль генератора трафіка може бути передана базовим станціям або отримання певного трафіку відслідковування абонентів на трасі під час руху.

Отже, для реалізації системи потрібно:

1) Створити віртуальну машину для генерації трафіку

2) Налаштувати та запустити потік даних та запустити завдання потоку даних

3) Отримати протікання даних типу Dataflow

4) Підключити потік даних до Pub/Sub та BigQuery

5) Перевірити, як автоматичне масштабування Dataflow налаштовує обчислювальні ресурси для оптимальної обробки вхідних даних

6) Дослідити навантаження на систему в реальному часі

Перш за все необхідно зареєструватись та створити віртуальну машину, для цього переходи на сайт та створюємо машину з мінімальними параметрами рис 4.2.

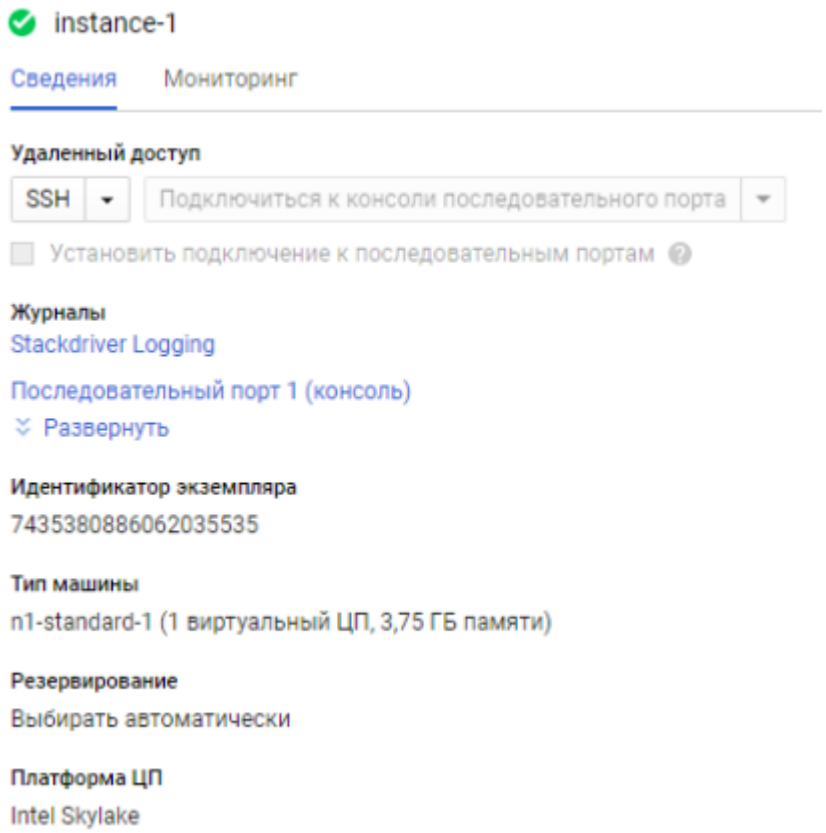


Рис 4.2. Приклад переходу на сайт та створення машини з мінімальними параметрами

Після успішного створення віртуальної машини переходимо до запуску генератора даних, для цього спочатку потрібно створити «набір даних» та «склад даних» (Bucket).

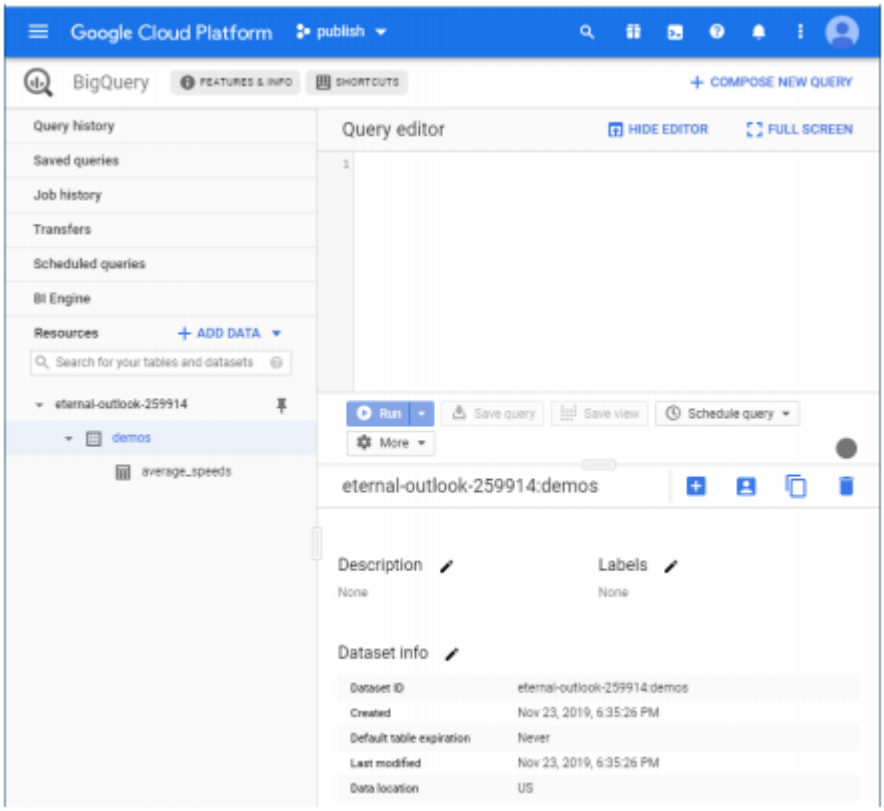


Рис 4.3. Створений проект та набір даних

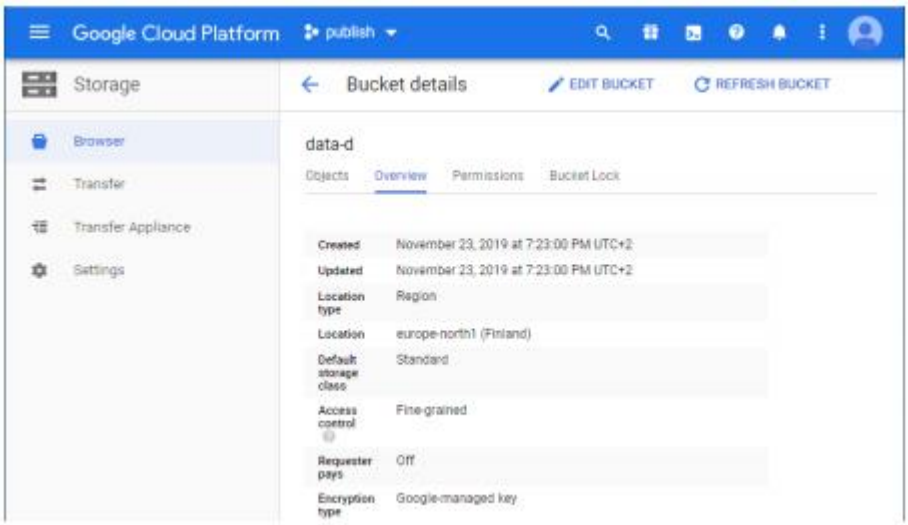


Рис 4.4. Створений проект та набір даних

Наступний крок за нашим планом це запуск генератора даних. Короткий опис генератора даних, це програма, що отримує дані з файлу формату CSV та передає їх на сервер хмарних технологій через спеціальний поток типу Dataflow.

Отже, необхідно зайти на створений сервер підключити його да нашого проекту,

для цього використовують команды:

$ gcloud auth application-default login

$ sudo pip install google-cloud-pubsub

$ virtualenv cpb104

$ source cpb104/bin/activate

$ pip install google-cloud-pubsub==0.38.0

$ gcloud auth application-default login

Після виконання цих команд, запускаємо генератор даних:

$ ./send_sensor_data.py --speedFactor=60 --project=YOUR-PROJECT-ID

Необхідно відкрити ще одну консоль та запустит :

$ training-data-analyst/courses/streaming/process/run_oncloud.sh yourproject

yourbucket AverageSpeeds

На рис. 4.5. зображено активний поток даних з генератора та на рис. 4.6. більш детальні характеристики даного процесу

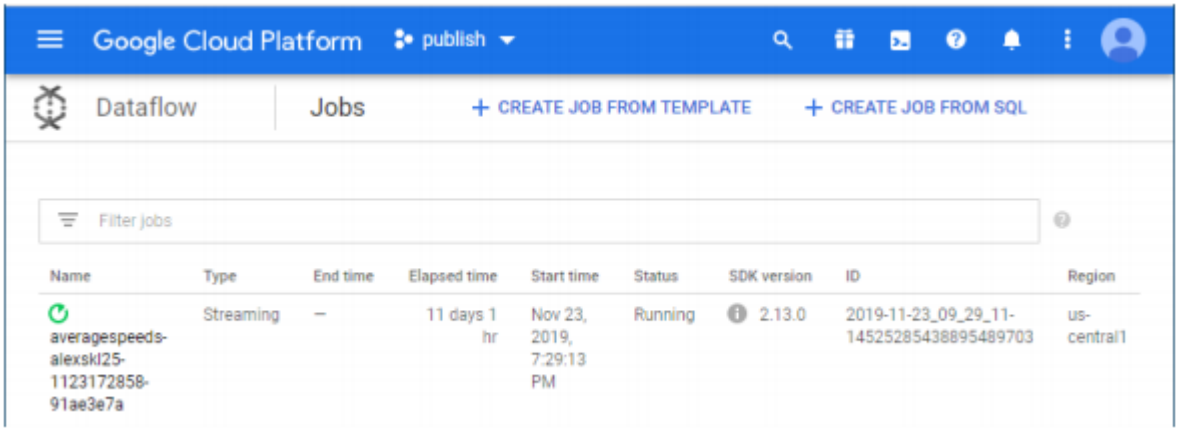


Рис. 4.5. Активний поток даних

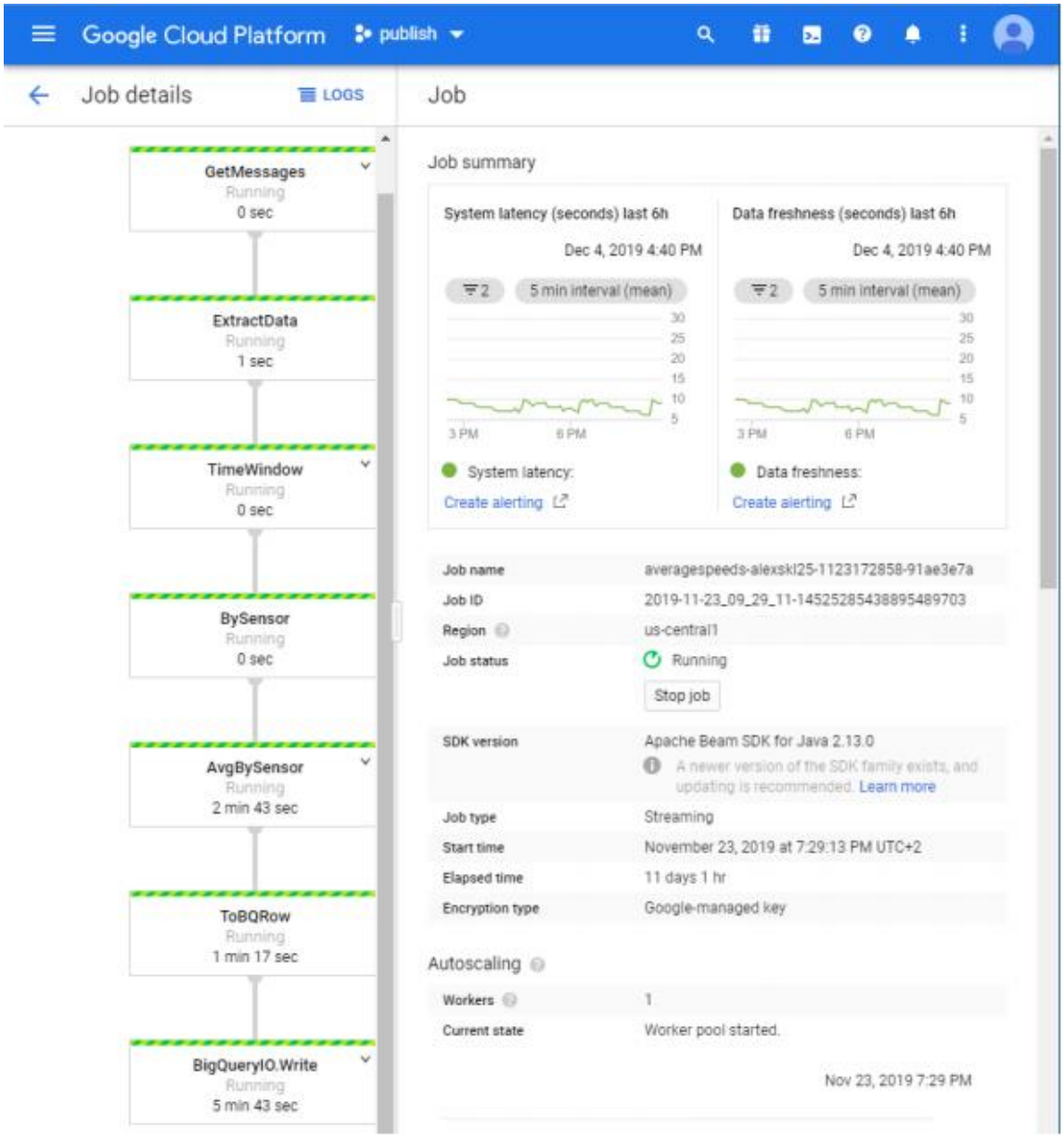


Рис. 4.6. Активне завдання потоку даних

Також можна побачити навантаження на віртуальну машину в реальному часі на панелі та аналітику потоку даних та використання каналу даних рис 4.7.

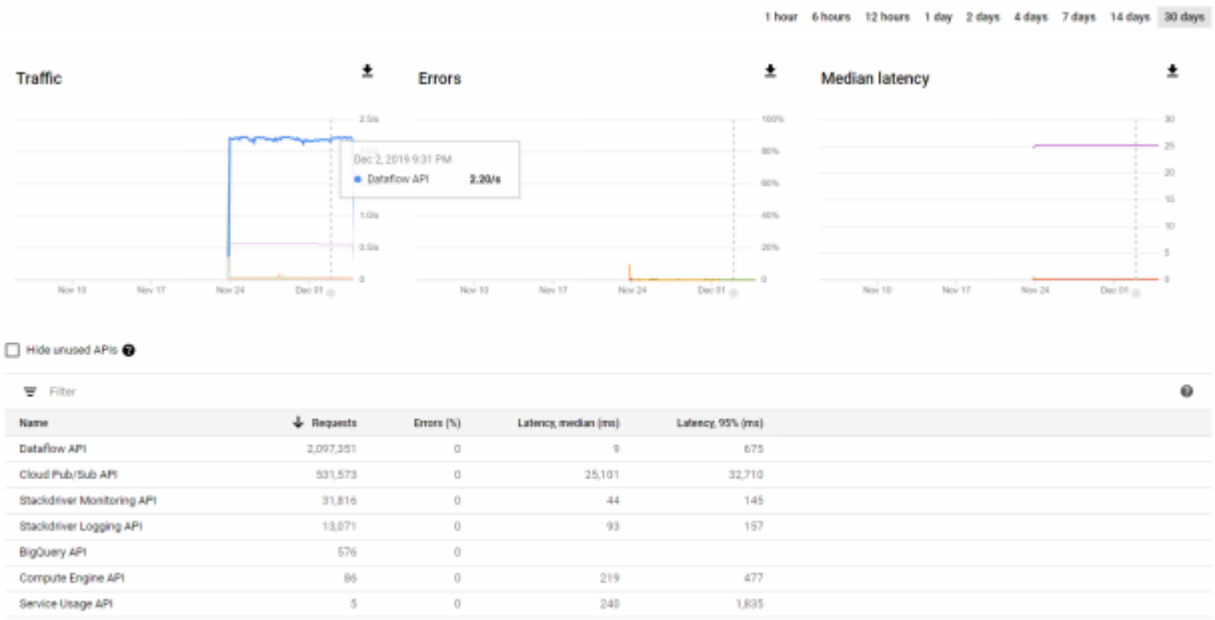


Рис. 4.7. Аналітика використання API та трафіку даних

Висновок: Створено та протестовано систему обчислень великого об’єму інформації від оператора на основі використання сервісу хмарних обчислень та інструментів для роботи з Big Data.

Висновок

У ході роботи над магістерською дисертацією досліджено структуру мобільного оператора, технологію LTE (структуру підключення абонентів, хендовера та інше.), можливість отримання та аналізу даних від абонентів, методи відслідковування абонентів в мережі за унікальними ідентифікаторами, а також можливість перенесення інфраструктури оператора на безсерверні технології, тобто втрати позиції оператора як обов’язкового власника усієї технологічної інфраструктури.

Розглянуто питання обробки даних існуючі методи та технології обробки даних Big Data, системи обробки/аналізу великих даних операторами в Україні та системи відслідковування та аналізу трафіку абонентів.

Розглянуто комплексні рішення у вигляді безсерверних (хмарних) технологій, класифікація таких рішень, огляд існуючих сервісів. Також обрано найоптимальнішу систему хмарних обчислень.

На основі розглянутих рішень та сучасних технологій створено та протестовано систему обчислень великого об’єму інформації від оператора на основі використання сервісу хмарних обчислень та інструментів для роботи з Big Data.

На підставі проведеної роботи з дослідження використання технології безсервеної обробки даних в мережі мобільного оператора, можна зробити висновок, що впровадження хмарних технологій в мобільних мережах значно зменшить витрати оператора на обслуговування інфраструктури та допоможе більш зорієнтуватися на надані якісних послуг.