Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем
(повна назва інституту/факультету)

Кафедра телекомунікацій
(повна назва кафедри)

«На правах рукопису»
УДК _____

До захисту допущено
В.о. завідувача кафедри

_____ Явіся В.С.___
(підпис) (ініціали, прізвище)

“___”_____2018_р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 172 Телекомунікації та радіотехніка,
(код і назва)

спеціалізація Апаратно-програмні засоби електронних комунікацій

на тему: Розробка алгоритмів розподілу навантаження в перспективних мобільних мережах на базі хмарної інфраструктури.

Виконав: студент 2 курсу, групи ТЗ-71мп
(шифр групи)

Полонський Андрій Михайлович _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник Міночкин Дмитро Анатолійович _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант_____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018_ рік

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

AMPS	Advanced Mobile Phone Service – аналоговий стандарт мобільного зв’язку
GSM	Global System for Mobile Communications – глобальна система 8 мобільного зв'язку
GPRS	General Packet Radio Service – загальний сервіс пакетної радіопередачі
LTE	Long Term Evolution – мобільного протоколу передачі даних
RAN	Radio access network - мережа радіодоступу
WiFi	Технологія бездротової локальної мережі
MIMO	Multiple Input Multiple Output – метод просторового кодування сигналу
QoE	Quality of Experience – якість використання
HSDPA	High-Speed Downlink Packet Access – доступ високошвидкісного приймання пакетних даних
HSPA	High Speed Packet Access – протокол високошвидкісного пакетного доступу
IAAS	Інфраструктура як сервіс
PAAS	Платформа як сервіс
SAAS	Програмне забезпечення як сервіс
QA	Quality Assurance – забезпечення якості
OPEX	Зменшення операційних витрат
CAPEX	Майбутні капітальні видатки
MNOs	Оператори мобільної мережі
MVNOs	Оператори мобільної віртуальної мережі
EPC	Evolved packet core – розвинуте пакетне ядро
SDN	Форма віртуалізації мережі
RTT	Round-trip delay – час затримки повернення
UE	User element – термінал користувача
VM	Vitrual machine – віртулальна машина

RR	Round robin – алгоритм розподілу навантаження
WRR	Weighted round robin - алгоритм розподілу навантаження
IWRR	Improved weighted round robin - алгоритм розподілу навантаження

ВСТУП

Мобільні мережі стали невід’ємною частиною сучасного життя. Неможливо представити людину у цивілізованому світі, яка хоча б якимось чином не зіштовхувалася з мобільними телефонами та не користувалася б послугами мобільних операторів.

Але досвід використання мобільних мереж може бути для кожного різним, адже якщо навантаження на мобільну мережу занадто велике, то користувач змушений буде мати справу з непомірними затримками у відповіді з серверу.

Мобільні мережі на базі хмарних інфраструктур мають ідентичну проблему, адже якщо сервер перенавантажити, то він не зможе відповідати на усі запити вчасно.

Це приводить до пошуку оптимізації та пришвидшення роботи мобільних мереж. Одним зі способів втілити це у життя є правильний розподіл навантаження на мережу.

Розподіл навантаження на мобільні мережі є одним з найважливіших процесів під час проектування та підтримки мобільної мережі.

Взагалі, розподіл навантаження – це процес покращення розподілення навантаження поміж багаточисленними сервісами, до яких звертаються термінали.

Розподіл навантаження націлений на оптимізацію використання ресурсу, отримання максимальної пропускної здатності та мінімального часу відповіді, також він використовується для уникнення перенавантаження будь-яких одиничних ресурсів.

Використання багаточисленних компонентів з розподілом навантаження замість єдиного компоненту може збільшити надійність та доступність через надлишок ресурсу.

Виходячи з вищесказаного стає очевидним факт того, що правильний розподіл навантаження на мобільних мережі є, мабуть, ключовим фактором надійної та задовільної з точки зору швидкості мобільної мережі.

Це приводить до необхідності дослідження та використання різних способів розподілу навантаження задля отримання найближчого до оптимального результату оптимізації мережі, що і буде виконано в даній роботі.

1 МОБІЛЬНІ МЕРЕЖІ

1.1 Історія мобільних мереж

Покоління стільникового зв'язку - це набір функціональних можливостей роботи мережі, а саме: реєстрація користувача, встановлення виклику, передача інформації між мобільним телефоном і базовою станцією по радіоканалу, процедура встановлення виклику між абонентами, шифрування, роумінг в інших мережах, а також набір послуг, що надаються абоненту.

Еволюція систем мобыльного зв'язку включає в себе кілька генерацій 1G, 2G, 3G і 4G. Ведуться роботи в області створення мереж мобільного зв'язку нового п'ятого покоління (5G). Стандарти різних поколінь, в свою чергу, поділяються на аналогові (1G) і цифрові системи зв'язку (інші).

Розглянемо їх докладніше.

Зв'язок завжди мала велике значення для людства. Коли зустрічаються дві людини, для спілкування їм досить голосу, але при збільшенні відстані між ними виникає потреба в спеціальних інструментах. Коли в 1876 році Олександр Грехем Белл винайшов телефон, був зроблений значний крок, який дозволив спілкуватися двом людям, однак для цього їм необхідно було перебувати поруч зі стаціонарно встановленим телефонним апаратом! Понад сто років провідні лінії були єдиною можливістю встановлення телефонного зв'язку для більшості людей. Системи радіозв'язку, які не залежать від проводів для організації доступу до мережі, були розроблені для спеціальних цілей (наприклад, армія, поліція, морський флот і замкнуті мережі автомобільної радіозв'язку), і, врешті-решт, з'явилися системи, що дозволили людям спілкуватися по телефону, використовуючи радіозв'язок. Ці системи призначалися головним чином для людей, що пересувалися на машинах, і стали відомі як телефонні системи рухомого зв'язку.

Перше покоління мобільного зв'язку (1G)

Офіційним днем народження стільникового зв'язку вважається 3 квітня 1973 року [1], коли глава підрозділу мобільного зв'язку компанії Motorola Мартін Купер подзвонив начальнику дослідного відділу AT & T Bell Labs Джоелю Енгелю,

перебуваючи на заповненій Нью-йоркської вулиці. Саме ці дві компанії стояли біля витоків мобільної телефонії. Комерційну реалізацію дана технологія отримала 11 років по тому, в 1984 році, у вигляді мобільних мереж першого покоління (1G), які були засновані на аналоговому способі передачі інформації.

Основними стандартами аналогового стільникового зв'язку стали AMPS (Advanced Mobile Phone Service - вдосконалена рухома телефонна служба) (США, Канада, Центральна і Південна Америка, Австралія), TACS (Total Access Communications System - тотальна система доступу до зв'язку) (Англія, Італія, Іспанія , Австрія, Ірландія, Японія) і NMT (Nordic Mobile Telephone - північний мобільний телефон) (країни Скандинавії і ряд інших країн). Були й інші стандарти аналогового мобільного зв'язку - З-450 в Німеччині і Португалії, RTMS (Radio Telephone Mobile System - радіотелефонна мобільна система) в Італії, Radiocom 2000 у Франції. В цілому мобільний зв'язок першого покоління представляла собою клаптикову ковдру несумісних між собою стандартів.

Таблиця 1.1 Характеристики аналогових стандартів мобільного зв'язку

Характеристика	AMPS	TACS	NMT-450	NMT-900	Radiocom 2000	NTT
Діапазон частот, МГц	825-845 870-890	935-950 (917-933) 890-905 (872-888)	453-457,5 463-467,5	935-960 890-915	424.8-427.9 418.8-421.9	925-940 870-885
Радіус соти, км	2-20	2-20	2-45	0,5-20	5-20	5-10
Потужність передавача БМ, Вт	45	50	-	-	-	25
Ширина полос частот каналу, кГц	30 (12,5)	25	25	25/12,5	12,5	25

Час переключення на межі соти, мс	250	290	1250	270	-	800
Мінімальне віношення сигнал\шум, дБ	10 (6,5)	10	15	15	-	15

За часів 1G ніхто не думав про послуги передачі даних - це були аналогові системи, задумані і розроблені виключно для здійснення голосових викликів і деяких інших скромних можливостей. Модеми існували, однак через те, що бездротовий зв'язок більш схильна до шумів і спотворень, ніж звичайна дротова, швидкість передачі даних була неймовірно низькою. До того ж, вартість хвилини розмови в 80-х була такою високою, що мобільний телефон міг вважатися розкішшю.

У всіх аналогових стандартах застосовується частотна (ЧМ) або фазова (ФМ) модуляція для передачі мови і частотна маніпуляція для передачі інформації управління. Цей спосіб має ряд істотних недоліків: можливість прослуховування розмов іншими абонентами, відсутність ефективних методів боротьби з завмираннями сигналів під впливом навколишнього ландшафту і будівель або внаслідок пересування абонентів. Для передачі інформації різних каналів використовуються різні ділянки спектру частот - застосовується метод множинного доступу з частотним поділом каналів (Frequency Division Multiple Access - FDMA). З цим безпосередньо пов'язаний основний недолік аналогових систем - відносно низька ємність, що є наслідком недостатньо раціонального використання виділеної смуги частот при частотному поділі каналів.

У кожній країні була розроблена власна система, несумісна з іншими з точки зору обладнання та функціонування. Це спричинило виникнення необхідності у створенні спільної європейської системи рухомого зв'язку з високою пропускною спроможністю і зоною покриття всієї європейської території. Останнє означало, що одні й ті ж мобільні телефони могли використовуватися у всіх Європейських країнах, і що вхідні дзвінки повинні були автоматично направлятися в мобільний

телефон незалежно від місцезнаходження користувача (автоматичний роумінг). Крім того, очікувалося, що єдиний Європейський ринок із загальними стандартами призведе до здешевлення призначеного для користувача устаткування і мережевих елементів незалежно від виробника.

Друге покоління мобільного зв'язку (2G)

У 1982 році CEPT (франц. Conférence européenne des administrations des postes et télécommunications - Європейська конференція поштових і телекомунікаційних відомств) сформувала робочу групу, названу спеціальною групою по рухомого зв'язку GSM (франц. Groupe Spécial Mobile) для вивчення і розробки пан-Європейської наземної системи рухомого зв'язку загального користування - друге покоління систем стільникового телефонії (2G). Назва робочої групи GSM також стало використовуватися в якості назви системи рухомого зв'язку. У 1989 році обов'язки CEPT були передані в Європейський інститут стандартів в телекомунікації ETSI (англ. European Telecommunications Standards Institute). Спочатку GSM призначалася тільки для країн-членів ETSI. Однак багато інших країн також мають впроваджену систему GSM, наприклад, Східна Європа, Середній Схід, Азія, Африка, Тихоокеанський регіон і Північна Америка (з похідною від GSM, названої PCS1900). Назва GSM стало означати "глобальна система для рухомого зв'язку", що відповідає її сутності.

Перші мобільні мережі другого покоління (2G) з'явилися в 1991 році. Їх основною відмінністю від мереж першого покоління став цифровий спосіб передачі інформації, завдяки чому з'явилася, улюблена багатьма, послуга обміну короткими текстовими повідомленнями SMS (англ. Short Messaging Service). При будівництві мереж другого покоління Європа пішла шляхом створення єдиного стандарту - GSM, в США більшість 2G-мереж було побудована на базі стандарту D-AMPS (Digital AMPS - цифровий AMPS), що є модифікацією аналогового AMPS. До речі, саме ця обставина стала причиною появи американської версії стандарту GSM - GSM1900. З розвитком і поширенням Інтернет, для мобільних пристроїв мереж 2G, був розроблений WAP (англ. Wireless Application Protocol - бездротовий

протокол передачі даних) - протокол бездротового доступу до ресурсів глобальної мережі Інтернет безпосередньо з мобільних телефонів.

Основними перевагами мереж 2G в порівнянні з попередниками було те, що телефонні розмови були зашифровані за допомогою цифрового шифрування; система 2G представила послуги передачі даних, починаючи з текстових повідомлень СМС.

Зростаюча потреба користувачів мобільного зв'язку в використанні Інтернет з мобільних пристроїв стала основним поштовхом для появи мереж покоління 2,5G, які стали проміжними між 2G і 3G. Мережі 2,5G використовують ті ж стандарти мобільного зв'язку, що і мережі 2G, але до наявних можливостей додалася підтримка технологій пакетної передачі даних - GPRS (англ. General Packet Radio Service - пакетна радіозв'язок загального користування), EDGE (англ. Enhanced Data rates for GSM Evolution - підвищена швидкість передачі даних для розвитку GSM) в мережах GSM. Використання пакетної передачі даних дозволило збільшити швидкість обміну інформацією при роботі з мережею Інтернет з мобільного пристроїв до 384 кбіт / с, замість 9,6 кбіт / с у 2G-мереж.

Система HSCSD (англ. High Speed Circuit Switched Data - високошвидкісна передача даних) є найпростішим поліпшенням системи GSM, призначеної для передачі даних. Суть цієї технології полягала у виділенні одному абоненту не одного, а декількох (теоретично до восьми) тимчасових інтервалів. Таким чином, максимальна швидкість збільшувалася до 115,2 кбіт / с. HSCSD забезпечувала швидкість, достатню для виходу в Інтернет, однак, при передачі даних інформаційні пакети розділені невизначеними за часом проміжками, таким чином, використання цієї технології вкрай марнотратно. Справа в тому, що мережі HSCSD, як і класичні мережі GSM, засновані на технології комутації каналів, в яких за абонентом закріплюють двобічний канал на весь час сеансу зв'язку. Через пауз у передачі канальний ресурс витрачався нераціонально.

Подальшою еволюцією системи GSM стала технологія GPRS. Її впровадження сприяло більш ефективному використанню канального ресурсу і створення комфортного середовища при роботі з мережею Інтернет. Система GPRS

розроблена як система пакетної передачі даних з теоретичної максимальною швидкістю передачі близько 170 кбіт / с. GPRS співіснує з мережею GSM, повторно використовуючи базову структуру мережі доступу. Система GPRS є розширенням мереж GSM з наданням послуг передачі даних на існуючій інфраструктурі, в той час як базова мережа розширюється за рахунок накладення нових компонентів і інтерфейсів, призначених для пакетної передачі.

Прогрес не стояв на місці і, для збільшення швидкості передачі даних, була винайдена нова система - EDGE. Вона передбачала введення нової схеми модуляції. В результаті стала досяжна швидкість в 384 кбіт / с. EDGE була введена в мережах GSM з 2003 фірмою Cingular (нині AT & T) в США.

Технології GPRS і EDGE в різних джерелах називали по-різному. Вони вже переросли в другому поколінні, але ще не дотягували до третього. Найчастіше GPRS називали 2,5G, EDGE - 2,75G.

Третє покоління мобільного зв'язку (3G)

Подальшим розвитком мереж стільникового зв'язку став перехід до третього покоління (3G). 3G - це стандарт мобільного цифрового зв'язку, який під абревіатурою IMT-2000 (англ. International Mobile Telecommunications - міжнародна мобільний зв'язок 2000) об'єднує п'ять стандартів - W-CDMA, CDMA2000, TD-CDMA / TD-SCDMA, DECT (англ. Digital Enhanced Cordless Telecommunication - технологія поліпшеного цифровий бездротового зв'язку). З перерахованих складових частин 3G тільки перші три є повноцінними стандартами стільникового зв'язку третього покоління. DECT - це стандарт бездротової телефонії домашнього або офісного призначення, який в рамках мобільних технологій третього покоління, може використовуватися тільки для організації точок гарячого підключення (хот-спотів) до даних мереж.

Стандарт IMT-2000 дає чітке визначення мереж 3G - під мобільною мережею третього покоління розуміється інтегрована мобільна мережа, яка забезпечує: для нерухомих абонентів швидкість обміну інформацією не менш 2048 кбіт / с, для абонентів, що рухаються зі швидкістю не більше 3 км / год - 384 кбіт / с, для абонентів, які прямують зі швидкістю не більше 120 км / ч - 144 кбіт / с. При

глобальному супутниковому покритті мережі 3G повинні забезпечувати швидкість обміну не менше 64 кбіт / с. Основою всіх стандартів третього покоління є протоколи множинного доступу з кодовим розділенням каналів. Подібна технологія мережевого доступу не є чимось принципово новим. Перша робота, присвячена цій темі, була опублікована в СРСР ще в 1935 році Д.В. Агеєвим.

Технічно мережі з кодовим поділом каналів працюють таким чином - кожному користувачеві привласнюється певний числовий код, який поширюється по всій смузі частот, виділених для роботи мережі. При цьому будь-яке тимчасове поділ сигналів відсутня, і абоненти використовують всю ширину каналу. При цьому, природно, сигнали абонентів накладаються один на одного, але завдяки числовому коду можуть бути легко диференційовані. Як було згадано вище, дана технологія відома досить давно, проте до середини 80-х років минулого століття вона була засекреченою і використовувалася виключно військовими і спецслужбами. Після зняття грифів секретності почалося її активне використання і в цивільних системах зв'язку.

Покоління 3,5G

Подальшим розвитком мереж стала технологія HSPA (англ. High Speed Packet Access - високошвидкісний пакетний доступ), яку стали називати 3,5G. Спочатку вона дозволяла досягти швидкості в 14,4 Мбіт / с, проте зараз теоретично досяжна швидкість 84 Мбіт / с і більше. Вперше HSPA була описана в п'ятій версії стандартів 3GPP. В її основі лежить теорія, згідно з якою при порівнянних розмірах сот застосування многокодовой передачі дозволяє досягати пікових швидкостей.

Четверте покоління мобільного зв'язку (4G)

У березні 2008 року відділ радіозв'язку Міжнародного союзу електрозв'язку (МСЕ-Р) визначив ряд вимог для стандарту міжнародної рухомий бездротового широкосмугового зв'язку 4G, який отримав назву специфікацій International Mobile Telecommunications Advanced (IMT-Advanced) [2], зокрема встановивши вимоги до швидкості передачі даних для обслуговування користувачів : швидкість 100 Мбіт / с повинна надаватися високоподвіжной абонентам (наприклад, поїздам і

автомобілів), а абонентам з невеликою рухливістю (наприклад пішоходам і фіксованим абонентам) повинна надано ться швидкість 1 Гбіт / с.

Так як перші версії мобільного WiMAX (англ. Worldwide Interoperability for Microwave Access - всесвітня сумісність для мікрохвильового доступу) і LTE (англ. Long Term Evolution - довгостроковий розвиток) підтримують швидкості значно менше 1 Гбіт / с, їх не можна назвати технологіями, відповідними IMT- Advanced, хоча вони часто згадуються постачальниками послуг, як технології 4G. 6 грудня 2010 року MCE-P визнав, що найбільш просунуті технології розглядають як 4G.

Основний, базової, технологією четвертого покоління є технологія ортогонального частотного ущільнення OFDM (англ. Orthogonal Frequency-Division Multiplexing - мультиплексування з ортогональним частотним поділом каналів). Крім того, для максимальної швидкості передачі використовується технологія передачі даних за допомогою N антен і їх прийому M антенами - MIMO (англ. Multiple Input / Multiple Output - безліч входів / безліч виходів). При даній технології передають і прийомні антени рознесені так, щоб досягти слабкою кореляції між сусідніми антенами.

Таким чином, еволюцію стандартів мобільного зв'язку можна представити в наступному вигляді:

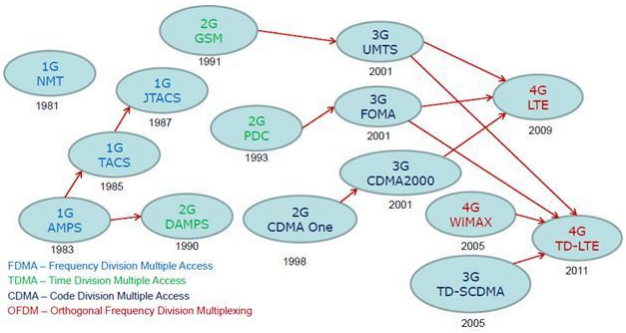

Рисунок 1.1 Еволюція стандартів мобільного зв'язку

Порівняльні характеристики стандартів різних поколінь мобільного зв'язку можна звести в наступну таблицю:

Таблиця 1.2 Еволюція мобільної телефонії

1G	2G	3G
▪ Аналогова телефонія ▪ Мобільність ▪ Базові послуги ▪ Несумісність стандартів	▪ Цифрова телефонія і передача повідомлень ▪ Мобільність і роумінг ▪ Підтримка передачі даних ▪ Додаткові послуги ▪ Полуглобальное рішення	▪ Широкосмугова передача даних і передача мови по протоколу IP (VoIP) ▪ Мобільність і роумінг ▪ Сервісна концепція і моделі ▪ Глобальне рішення
с 1980-х	с 1990-х	с 2000-х

П'яте покоління мобільного зв'язку (5G)

В даний час ведуться науково-дослідні роботи в напрямку розробки та створення мереж 5G. До мереж п'ятого покоління висунуті такі вимоги (в порівнянні з LTE):

- Зростання в 10-100 разів швидкості передачі даних в розрахунку на абонента;
- Зростання в 1000 разів середнього споживаного трафіку абонентом на місяць;
- Можливість обслуговування більшого (в 100 разів) числа відключених до мережі пристроїв;
- Багаторазове зменшення споживання енергії абонентських пристроїв;
- Скорочення в 5 і більше разів затримок в мережі;
- Зниження загальної вартості експлуатації мереж п'ятого покоління.

Розробкою мереж 5G займаються кілька країн по всьому світу. В даний час завдання - визначитися, на базі яких технологій будуть розгортатися нові мережі. Оптимізація і стандартизація обладнання, а також перші дослідні запуски заплановані на 2015-2018 роки, а в 2018-2020 очікується розгортання перших некомерційних мереж 5G для дослідної експлуатації. Комерційний запуск мереж п'ятого покоління очікується не раніше 2020 року.

1.2 Сучасні мобільні мережі.

4G мережі.

4G є четвертим поколінням технологій мобільних телефонів. Це випливає з існуючих мобільних технологій 3G (третього покоління) та 2G (другого покоління).

2G-технологія, запущена в 1990-х і здатна робити цифрові телефонні дзвінки та надсилання текстів. Тоді 3G з'явився в 2003 році і дозволив переглядати веб-сторінки, робити відеодзвінки та завантажувати музику та відео на ходу.

Технологія 4G спирається на те, що пропонує 3G, але робить все на набагато швидшій швидкості.

4G LTE

Стандарт LTE розроблявся в рамках 3GPP (The 3rd Generation Partnership Project) як продовження CDMA і UMTS і спочатку не відносився до четвертого покоління мобільного зв'язку. Міжнародним союзом електрозв'язку як стандарт зв'язку, що відповідає всім вимогам бездротового зв'язку четвертого покоління, був обраний десятий реліз LTE - LTE Advanced, який вперше був представлений японською компанією NTT DoCoMo. Так як даний стандарт можна реалізувати на існуючих стільникових мережах, то він став більш популярний у операторів стільникового зв'язку. У квітні 2008 року компанія Nokia заручилася підтримкою ряду компаній (Sony Ericsson, NEC) для розвитку стандарту LTE і надання цьому стандарту конкурентоспроможності проти WiMAX . У тому ж році аналітична компанія Analysys Mason спрогнозувала збільшення зростання потреби стільникових технологій, таких як LTE, ніж WiMAX .

Перша комерційна мережа LTE була запущена 14 грудня 2009 шведської телекомунікаційної компанією TeliaSonera спільно з Ericsson, в Стокгольмі і Осло [3].

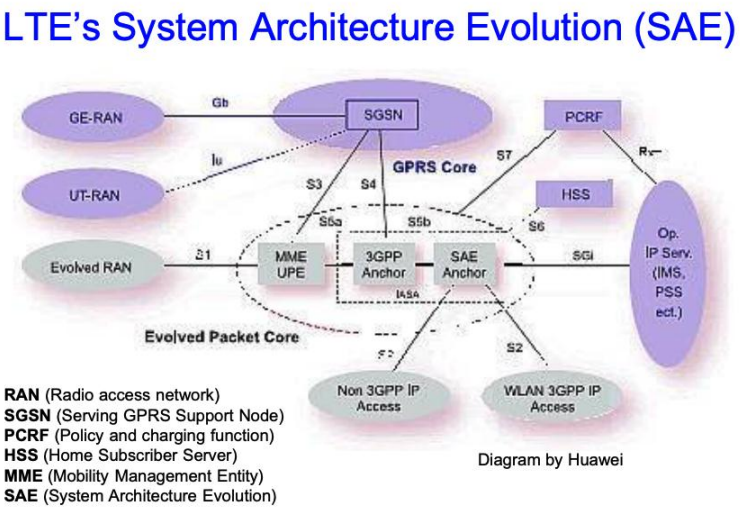

Рисунок 1.2 Еволюція системної архітектури LTE

Переваги 4G LTE:

- покращена швидкість завантаження / виванртаження
- зменшена затримка
- кристально чисті голосові дзвінки

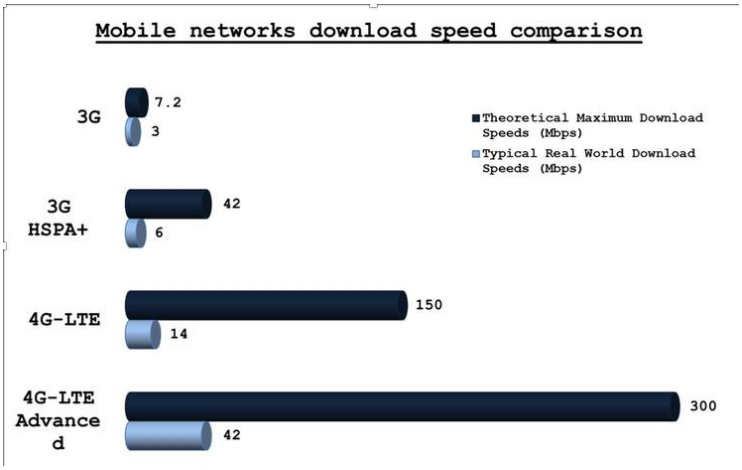

Рисунок 1.3 Порівняння мобільних мереж

Стандартний 4G (або 4G LTE) приблизно в п'ять-сім разів швидше, ніж 3G, пропонуючи теоретичні швидкості до близько 150 Мбіт / с. Це дорівнює максимальній потенційній швидкості приблизно 80 Мбіт / с у реальному світі (хоча навіть швидкість, яка настільки висока, рідкісна, і, як правило, вимагає щось на кшталт "Double Speed" EE або сервісу "Extra Speed" BT Mobile).

З стандартним 4G ви можете завантажити 2 Гб відео HD протягом 3 хвилин 20 секунд на стандартній мобільній мережі 4G, тоді як стандартна мережа 3G буде виконувати цю ж дію більше 25 хвилин.

Однак нова ще більш швидка версія 4G називається 4G LTE-Advanced (також відома як LTE-A, 4.5G або 4G +).

Це забезпечує теоретичну швидкість до 1,5 Гбіт / с, однак поточна мережа LTE-A має максимальну потенційну швидкість 300 Мбіт / с, а реальна швидкість падає набагато нижче.

Ви можете очікувати, що доступність LTE-A збільшиться в найближчі роки, і швидкість може ще швидше зростати. Наприклад, компанія Ericsson розробила технологію, яка могла б дозволити досягти максимальної швидкості 4 ГГц на швидкості 1 Гбіт / с, хоча це, швидше за все, не буде доступним, якщо взагалі.

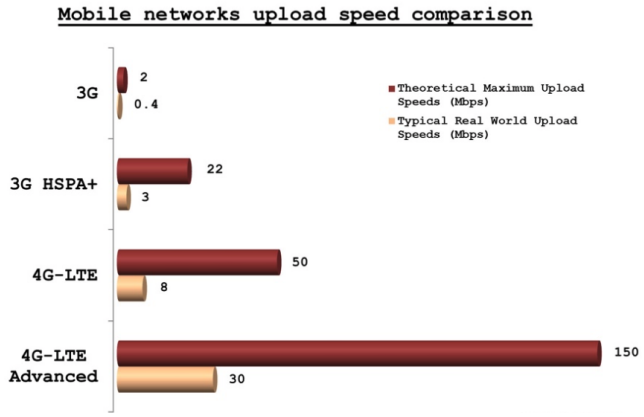

Рисунок 1.4 Порівняння мобільних мереж

Навіть більше - 5G, який вже розробляється, і є наміри почати використовувати цю технологію до 2020 року і забезпечити суттєво швидші темпи, ніж ви могли б сподіватися отримати з 4G. Поточні очікування це швидкість завантаження понад 5 Гб, щось блізько 1-10 Гбіт / с.

Швидкість завантаження - це не єдине, що було вдосконалено, оскільки 4G також має кращий час відгуку, ніж 3G - через зниження "затримки". Це означає, що пристрій, підключений до мобільної мережі 4G, отримає швидше відповідь на запит, ніж той самий пристрій, підключений до мобільної мережі 3G.

Зменшення затримки

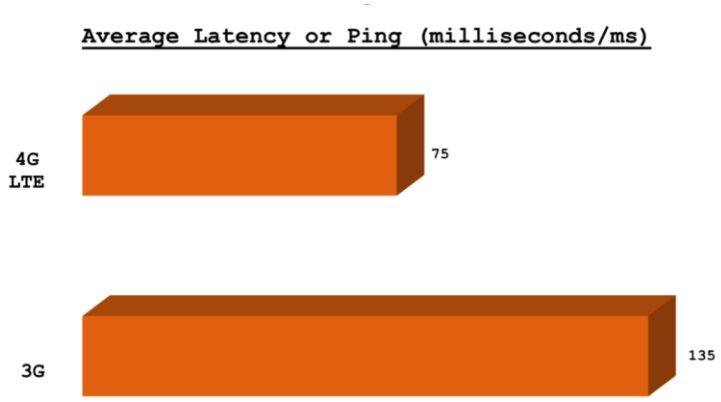

Рисунок 1.5 Порівняння затримки 3g і 4g

Покращене час затримки, зменшений з 120 мілісекунд (3G) до приблизно 75 мілісекунд (4G), може здатися не таким значним на папері. Однак, вони можуть істотно відрізнятись, граючи в онлайн-іграх та транслюючи пряме відео. Дізнайтеся більше про переваги 4G для ігор.

Покращення дзвінків в 4G-LTE

Voice over LTE (VoLTE) схожий на Voice over Internet Protocol (VoIP), який використовує голосові додатки, такі як Skype, для підтримки голосових дзвінків через Інтернет. Ефективно, VoLTE їде на задній панелі мережі 4G і виводить кристально чисті голосові дзвінки та відеочат на свій 4G мобільний телефон.

Інша перевага полягає в тому, що вона дозволяє вам дзвонити та наносити текст, коли у вас є тільки 4G-з'єднання, де раніше ви не змогли б це зробити.

VoLTE тепер доступний від Three, EE, O2, Vodafone, iD Mobile та Virgin Mobile, але тільки в певних місцях і на деяких телефонах. Ознайомтеся з нашим повним керівництвом VoLTE для детальної інформації про доступність.

4G LTE Advanced

Основні особливості LTE Advanced

З роботою, починаючи з LTE Advanced, з'являється ряд ключових вимог та ключових функцій. Незважаючи на те, що в специфікаціях ще не зафіксовано, існує багато цілей високого рівня для нової специфікації LTE Advanced. Потрібно буде перевірити їх, і в специфікаціях все ще потрібно виконати багато роботи, перш ніж

вони будуть зафіксовані. В даний час деякі основні заголовки цілей для LTE Advanced можна побачити нижче [4]:

1. Максимальна швидкість передачі даних: низхідна лінія зв'язку - 1 Гбіт / с; uplink - 500 Мбіт / с.
2. Ефективність спектру: у 3 рази більше, ніж LTE.
3. ККД пікового спектра: низхідна лінія - 30 к / с; висхідна лінія зв'язку - 15 б.п. / Гц.
4. Використання спектру: здатність підтримувати масштабоване використання смуги пропускання та агрегацію спектру
5. де необхідно використовувати несуміжний спектр.
6. Затримка: від Idle до Connected менш ніж за 50 мс, а потім менше, ніж 5 мс в одному напрямку
7. для індивідуальної передачі пакета.
8. Пропускна спроможність клієнтської сторони дорівнює вдвічі більшою, ніж LTE.
9. Середня пропускна здатність користувача в 3 рази більше, ніж LTE.
10. Мобільність: так само, як у LTE
11. Сумісність: LTE Advanced може працювати з застарілими системами LTE та 3GPP.

Спектр частот LTE-Advanced

Однією з ключових проблем для LTE-Advanced є наявність радіочастотного спектра. Збільшення пропускної здатності даних за рахунок коштів, що вимагають ширшої пропускної здатності. Незважаючи на те, що спектр може використовуватися більш ефективно за допомогою таких методів, як MIMO та формування променів, все ще необхідно мати більший рівень доступного спектру. У результаті 2007 року на Всесвітній радіо конференції були визначені нові групи для використання IMT / IMT Advanced технологій. Можливі групи включали:

- 450-470 МГц
- 698-862 МГц

- 790-862 МГц
- 2,3-2,4 ГГц
- 3.4-3.6 ГГц

Зверніть увагу, що діапазони частот вважаються незалежними від властивостей фунцій, а це означає, що допустиме розгортання більш раннього випуску продукту в діапазоні, не визначеному до пізнішого випуску.

Підтримка більш широко-смугових агрегаторів-передавачів

Основною функцією LTE-Advanced буде гнучке використання спектра. Основу для технології повітряного інтерфейсу LTE-Advanced в основному визначають використання ширших смуг пропускання, потенційно навіть до 100 МГц, безперервного розгортання спектра, також називається агрегацією спектру або носія, а також необхідністю використання гнучкого спектра. Концепція складається з двох чи кількох компонентів кар'єри, щоб підтримувати ширші смуги пропускання до 100 МГц. Слід встановити можливість налаштування всіх компонентних носіїв, які є сумісними з LTE Rel-8, принаймні, коли агреговані числа компонентних носіїв у UL та DL є однаковими. Не всі компоненти носіїв можуть обов'язково бути сумісними з LTE Rel-8. Термінал LTE-Advanced з можливостями прийому та / або передачі для агрегації носіїв може одночасно отримувати та / або передавати на декількох носіях компонентів.

Скоординовані багато-точкові передача і прийняття (COMP)

Ще одна методика є координованою передачею та прийомом багатопозиційних точок (CoMP). Широко обговорюється в контексті LTE-Advanced. Основна ідея CoMP – це застосовувати тісну координацію між передачами на різних сайтах камер, тим самим досягти більш висока ємність системи і, особливо важливо, поліпшення швидкості передачі даних у стільникових мережах.

Переваги CoMP

- Хоча LTE Advanced CoMP, Coordinated Multipoint є складним набором методів, це дає багато переваг як користувачеві, так і оператору мережі.
- Краще використовувати мережу: забезпечуючи підключення до декількох базових станцій відразу, використовуючи CoMP, дані можна передавати через найменш завантажені базові станції для кращого використання ресурсів.
- Забезпечує покращену продуктивність прийому: Використовуючи декілька сайтів для кожного з'єднання означає, що загальний прийом буде покращено, а кількість випавших дзвінків повинна бутизменшено
- Кілька прийомів сайту збільшує отриману потужність: спільний прийом з декількох
- Базові станції або сайти, що використовують технологію LTE Coordinated Multipoint, дають загальний доступ отримана потужність на телефоні повинна бути збільшена.
- Зменшення перешкод: за допомогою спеціалізованих методів комбінування можна
- використовувати конструктивні, а не руйнівні перешкоди, тим самим зменшуючи
- рівні перешкод.
- Координаційні схеми можна розділити на дві категорії, які використовуються окремо або разом:
- Динамічне планування координації між кількома клітинами.
- Спільна передача / прийом з декількох осередків.

У першому випадку, CoMP може певною мірою розглядатися як продовження координації взаємодії між клітин, яка вже існує в LTE Rel-8. У LTE-Advanced координування може бути з точки зору планування на різних сайтах клітин, тим самим досягти ще більш динамічної та адаптивної координації міжклітинних перешкод. Альтернативно або як доповнення, передачі можуть

здійснюватися спільно з мобільними терміналами з кількох ділянок мобільних пристроїв, тим самим не тільки зменшуючи перешкоди, але і збільшуючи отриману потужність. Передача з клітинних сайтів також може враховувати умови миттєвого каналу, тим самим забезпечуючи досягнення багатокористувацьких променів або попереднього кодування. Оцінка каналу, необхідна для демодуляції передачі низхідної лінії зв'язку на терміналі, може, в основному, бути отримана, використовуючи або специфічні для клітинок, або специфічні для користувача опорні сигнали. Щоб визначити обробку CoMP для застосування на стороні передавача, знання каналу потрібне мережею. Цього можна досягти шляхом узагальнення звітів про статус каналу в одному камері в LTE до звітів про багатоканальні процеси, тобто термінал повинен повідомляти про якість сприйманого каналу з кількох комірок. Оцінювання якості каналу може бути здійснено шляхом експлуатації вже переданих клітинних опорних сигналів. У висхідній лінії обробка приймача є конкретною для реалізації, і не передбачається вплив основних специфікацій. Отже, це може бути застосоване і до терміналів Rel-8. В принципі, Uplink CoMP схожий на більш м'який перенос, який вже застосовується в мережах WCDMA. Комбінування максимального співвідношення та сполучення відхилень - це приклади схем, які можуть бути використані для поєднання передачі висхідної лінії, отриманої в декількох точках.

Ретрансляція

Дуже високі темпи передачі даних, націлені на LTE-Advanced, вимагають, як вже було сказано, більш жорсткої інфраструктури. Координаційна багатоточкова передача, описана вище, є однією можливістю для розгортання більш щільної інфраструктури. Ще однією можливістю забезпечення більш щільної інфраструктури з точки зору бюджету зв'язку є розгортання різних типів ретрансляційних рішень. По суті, наміром є зменшення відстані між передавачем і приймачем, тим самим забезпечуючи більш високі швидкості передачі даних. Залежно від застосованої схеми можна передбачити різні типи ретрансляційних рішень, хоча всі вони поділяють основне властивість ретрансляції зв'язку між клітиною донора та терміналом. Донорська клітина, крім обслуговування одного

або декількох реле, також може безпосередньо зв'язуватися з іншими терміналами. Релейний вузол (RN) бездротовим чином підключається до донорної комірки донорського eNB через інтерфейс Un, а UE підключаються до RN через інтерфейс Uu, як показано на малюнку нижче

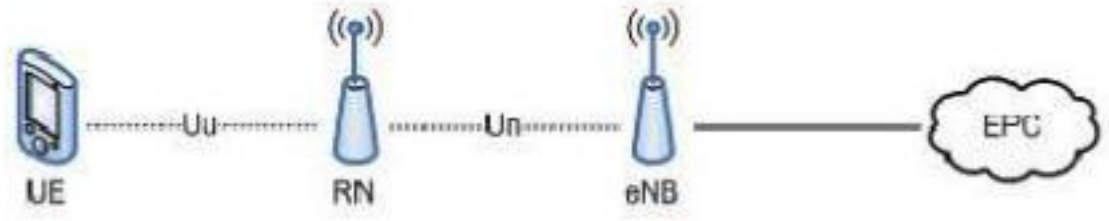

Рисунок 1.6 Схематичне зображення ретрансляції

Що стосується використання ретрансляційного вузла спектру, його операцію можна розділити на:

- Inband, в цьому випадку посилання eNB-RN поширюється на ту ж саму несучу частоту з посиланнями RN-UE. У цьому випадку UE-пристрої Rel-8 повинні мати можливість підключення до донорської клітини.
- Outband, в цьому випадку посилання eNB-RN не працюють на тій самій несучої частоті, як RN-UE. У цьому випадку UE-пристрої Rel-8 повинні мати можливість підключення до донорської клітини.

Покращене MIMO

Окрім ширшої пропускної спроможності, також очікується, що LTE-Advanced забезпечить більш високі швидкості передачі даних та покращену продуктивність системи завдяки подальшому розширенню підтримки для багатоантенntної передачі порівняно з першим випуском LTE. Для нинішньої лінії зв'язку можна передавати до восьми шарів, використовуючи конфігурацію антени 8x8, що дозволяє отримати максимальну спектральну ефективність, що перевищує вимогу 30 біт / с / Гц, і передбачає можливість передачі даних понад 1 Гбіт / с в смузі пропускання 40 МГц і навіть більш високі швидкості передачі даних з ширшою пропускною здатністю. Це вимагає введення додаткових еталонних сигналів не тільки для оцінки каналів, але також для вимірювань, таких як якість

каналу, для забезпечення адаптивної багатоантенної передачі. Необхідно враховувати зворотну сумісність, і можливі кандидати можуть бути як додатковими, так і додатковими опорними сигналами UE.

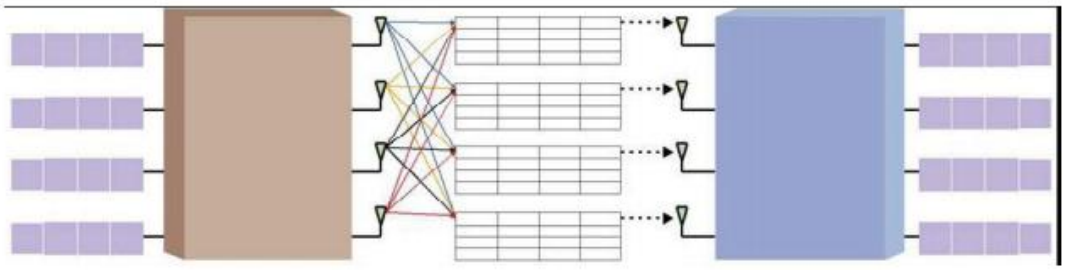

Рисунок 1.7 Схематичне зображення покращеного MIMO

Крім того, LTE-Advanced включає в себе просторову мультиплексування до чотирьох рівнів для висхідної лінії зв'язку. З чотиришаровою передачею у висхідній лінії зв'язку може бути досягнута максимальна спектральна ефективність вертикальної лінії зв'язку, яка перевищує 15 біт / с / Гц. Багато методів, які використовуються для просторового мультиплексування низхідної лінії зв'язку вже в LTE Rel-8, такі як кодовий та некодексний, на основі каналу, залежать від попереднього кодування, можуть розглядатися для того, щоб не тільки підвищити пікові швидкості, а й швидкості передачі даних на стороні клітини.

Підвищена підтримка гетерогенних розгортань мереж

Пом'якшення мереж радіодоступу може допомогти досягти майбутнього попиту на трафік та швидкість передачі даних. Це включає в себе доповнення макро-коміркового шару з додатковими піко-елементами малої потужності, які можуть розширювати можливості трафіку та швидкості передачі даних відповідно до вимог. Таке гетерогенне розгортання мережі (HetNet) можливе у поточному мобільному телефоні

мереж зв'язку, включаючи перший випуск LTE. Тим не менш, LTE Release 10 має функції, які можуть додатково пом'якшити перешкоди між клітинками, тим самим збільшуючи обсяги розгортання HetNet.

Спектральний збіг

Ключовим достоїнством усіх випусків LTE є те, що вони забезпечують конвергенцію в термінах доступу до радіо для парного та непарного спектра, що дозволяє більш ефективно використовувати спектр. Є два дуплексні альтернативи для мобільного зв'язку - FDD для спареного спектра та TDD для непарного спектра. Ці дуплексні схеми підтримуються різними технологіями 3GPP радіодоступу - GSM і WCDMA / HSPA для FDD і TD-SCDMA для TDD. Хоча FDD історично була домінуючою дуплексною схемою для мобільного зв'язку, інтерес до TDD зростає. Однією з причин цього є наявність непарного спектра, для ефективного використання якого необхідна високоефективна та глобально прийнята технологія TDD. Підтримуючи як FDD, так і TDD через ту ж технологію радіодоступу, LTE забезпечує конвергенцію радіодоступу для парного та непарного спектра в єдину загальноприйняту технологію. Це особливо корисно для TDD та використання непарного спектра, який до цих пір страждав від обмеженої доступності терміналу та ринкового імпульсу.

5G Мережі

Дорожня карта 5G мереж

Малюнок нижче ілюструє дорожню карту для 5G. Ми перебуваємо в ранньому етапі досліджень для створення прототипів. Очікується, що новий спектр буде погоджено в WRC 2015, що дозволить IMT визначити вимоги. Далі слідує діяльність з стандартизації та етап розвитку продукту до 2020 року. [5] Очікується, що перша хвиля мереж 5G буде оперативно перебігти 2021 року.

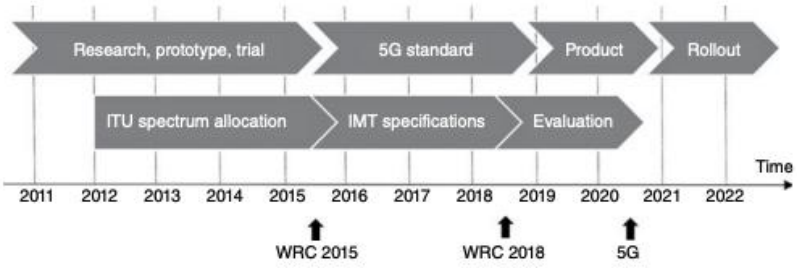

Рисунок 1.8 Дорожня карта 5G мереж

10 основних особливостей 5G

1. Еволюція існуючих технологій радіодоступу

Як згадувалося раніше, 5G навряд чи буде специфічним RAT [5], швидше, це, швидше за все, буде збіркою RAT, включаючи еволюцію існуючих, доповнену новими революційними конструкціями. Таким чином, перше і найбільш економічне рішення для подолання кризи потужності 1000x - це покращення існуючих рівнів ризику в термінах SE, EE та latency, а також підтримка гнучкого розподілу RAN серед кількох постачальників. Зокрема, LTE необхідно розвинути для підтримки масивного / 3D MIMO для подальшого використання просторового ступеня

Свободи (DOF) за допомогою вдосконаленого багатокористувацького формування променя, для подальшого вдосконалення можливостей для скасування інтерференції та координації можливостей перешкод при надмірному сценарії розгортання дрібних комірок. WiFi також має розвиватися, щоб краще використовувати наявний неліцензійний спектр. IEEE 802.11ac, останній розвиток технології WiFi, може забезпечити широкосмугові безпроводові канали з швидкістю передачі даних на декількох Гбіт. Він використовує ширшу смугу пропускання до 160 МГц на менш забрудненій смузі частот 5 ГГц, використовуючи до 256 квадратурної амплітудної модуляції (QAM). Він також може підтримувати одночасне передавання до чотирьох потоків за допомогою багатокористувацької технології MIMO. Технологія об'єднаного променя збільшила покриття на кілька порядків порівняно з попередником (IEEE 802.11n). Нарешті, великі телекомунікаційні компанії, такі як Qualcomm, нещодавно працювали над розробкою LTE в неліцензійному спектрі, а також інтеграцію трансиверів 3G / 4G / WiFi у єдину багатофункціональну базову станцію (БС). У зв'язку з цим передбачається, що майбутній UE буде достатньо розумним, щоб вибрати найкращий інтерфейс для підключення до RAN на основі вимог QoS виконуваної програми.

2. Надщільне розгортання малих комірок

Рощгортання малих комірок є ще одним перспективним рішенням для задоволення 1000ої кризи потужності, а також додаткові ЕЕ для системи. Це інноваційне рішення, також називається HetNet, може допомогти значно підвищити спектральну ефективність (б / с / Гц / м2). Взагалі, існує два способи реалізації HetNet: (i) накладання стільникової системи з невеликими клітинами однієї і тієї ж технології, тобто з мікро-, піко- або фемточеллами; (ii) накладання невеликими клітинами різних технологій на відміну від лише стільникового (наприклад, високошвидкісного пакетного доступу (HSPA), LTE, Wi-Fi тощо). Перший називається багаторівневим HetNet, тоді як останній називається мульти-RAT HetNet. Qualcomm, провідна компанія у вирішенні проблеми 1000х потужності через надмірну розгортання дрібних клітин, показала, що додавання невеликих клітин може майже лінійно збільшити пропускну спроможність мережі. Тобто ємність подвоюється кожен раз, коли ми подвоюємо кількість маленьких клітин.

3. Самоорганізована мережа

Інший ключовий компонент 5G - функція самоорганізації мережі (SON). Оскільки населення маленьких клітин збільшується, SON отримує більший імпульс. Майже 80% бездротового трафіку створюється в приміщенні. Щоб нести цей величезний трафік, ми потребуємо надмірного розміщення дрібних клітин у будинках - встановлених та обслуговуваних в основному користувачами - поза контролем операторів. Ці внутрішні маленькі клітини повинні бути самоконфігурованими і встановлюватися у вигляді підключення та відтворення. Крім того, вони повинні мати здатність SON усвідомлено адаптуватися до сусідніх невеликих клітин, щоб звести до мінімуму перешкоди між клітинами. Наприклад, маленька клітина може це зробити, автономно синхронізувавшись з мережею та вміло налаштувавши своє радіопокриття.

4. Машинний тип зв'язку

Окрім людей, сполучення мобільних машин є ще одним важливим аспектом 5G. Комунікація типу машини (МТС) - це нова програма, в якій один або обидва кінцевих користувачів сеансу зв'язку включають в себе машини. МТС нав'язує дві

основні проблеми в мережі. По-перше, кількість пристроїв, які потрібно підключити, є надзвичайно великим. Ericsson (одна з провідних компаній у вивченні 5G) передбачає, що у майбутньому мережевому суспільстві потрібно підключити 50 мільярдів пристроїв; компанія передбачає, що "все, що може бути корисним для підключення, буде з'єднане". Інший виклик, який нав'язує MTC, полягає у пришвидшенні попиту на мережу в режимі реального часу та дистанційного керування мобільними пристроями (такими як транспортні засоби). Це вимагає надзвичайно низької затримки менш ніж за мілісекунду, так званого "тактильного Інтернету", що передбачає 20x латентний покращення від 4G до 5G.

5. Розвиток радіодоступу міліметрових хвиль

Традиційний спектр під-3 ГГц [5] стає все більш перевантаженим, і теперішні РАТ наближаються до межі потужності Шеннона. Таким чином, дослідження щодо вивчення смуг cm- і mmWave для мобільних комунікацій вже розпочато. Незважаючи на те, що дослідження в цій галузі все ще перебувають у стані дитинства, результати виглядають багатообіцяючими. Існує три основні перешкоди для мобільного зв'язку mmWave. По-перше, втрати контуру відносно вище в цих смугах, порівняно зі звичайними смугами, що підходять для 3GHz. По-друге, електромагнітні хвилі, як правило, поширюються в напрямку лінії зору (LOS), що робить радіозв'язок уразливими для блокування рухомими об'єктами або людьми. Нарешті, але не менш важливо, втрати проникнення через будинки значно вищі у цих смугах, блокуючи зовнішні RAT для внутрішніх користувачів.

6. Редизайн зворотних зв'язків

Редизайн зворотних ліній зв'язку - це наступна критична проблема 5G. Паралельно з вдосконаленням RAN, ретранслюючі зв'язки також потребують реінжинірингу, щоб нести величезний обсяг користувацького трафіку, який генерується в камерах. В іншому випадку, зворотні зв'язки скоро стануть вузькими місцями, загрожуючи належній роботі всієї системи. Проблема посилюється, коли населення маленьких клітин збільшується. Можна розглянути різні засоби зв'язку, включаючи оптичні волокна, мікрохвильові печі та mmWave. Зокрема, для забезпечення надійного автоподачі без перешкод з використанням інших осередків

або з каналами доступу можна враховувати посилання mmWave між точками, що експлуатують масивні антени з дуже різкими променями.

7. Енергоефективність

ЕЕ залишатиметься важливим дизайнерським питанням при розробці 5G. Сьогодні інформаційно-комунікаційна технологія (ІКТ) споживає не менше 5% електроенергії, виробленої по всьому світу, і відповідає приблизно на 2% світових викидів парникових газів, приблизно еквівалентних викидам, створеним авіаційною промисловістю. Більше того, це те, що якщо ми не вживемо жодних заходів для скорочення викидів вуглецю, передбачається, що до 2020 року цей внесок буде подвоїтись. Отже, необхідно вживати енергоефективні підходи до проектування з боку RAN та зворотних трактів до UE.

8. Розподіл нового спектру на 5G

Ще однією критичною проблемою 5G є виділення нового спектру для забезпечення бездротового зв'язку в наступному десятилітті. 1000х трафік напруги важко управляти, тільки поліпшення спектральної ефективності або гіпер-ущільнення. Фактично, провідні телекомунікаційні компанії, такі як Qualcomm і NSN, вважають, що, крім технологічних інновацій, для задоволення попиту потрібен в 10 разів більше спектру. Розподіл смуги пропускання близько 100 МГц в смузі 700 МГц та ще одну смугу пропускання 400 Мгц на частоті близько 3,6 ГГц, а також потенційний розподіл смуги пропускання в декілька ГГц в діапазонах cm- або mmWave до 5G стануть центром наступної конференції WRC , організований MCE-R у 2015 році.

9. Спектральний обмін

Регуляторний процес для розподілу нових спектрів часто дуже трудомісткий, тому ефективне використання доступного спектру завжди має вирішальне значення. Інноваційні моделі розподілу спектру (відмінні від традиційного ліцензованого чи неліцензованого розподілу) можуть бути прийняті для подолання існуючих регуляторних обмежень. Традиційно було виділено багато радіочастотного спектра для військових радіолокаторів, де спектр не використовується повною мірою (24/7) або у всьому географічному регіоні. З

іншого боку, очищення спектру є дуже важким, оскільки якийсь спектр ніколи не може бути очищений чи може бути очищений лише дуже довго; Крім того, спектр може бути очищений в деяких місцях, але не у всій країні. Таким чином Qualcomm запропонував модель авторизованої / ліцензованої спільного доступу (ASA / LSA) для експлуатації спектру в малих клітинках (з обмеженим охопленням), не заважаючи діючим користувачам (наприклад, військовим радіолокаторам). Така модель розподілу спектра може компенсувати дуже повільний процес очищення спектру. Варто також відзначити, що, як прискорюється зростання мобільного трафіку, рефармінство спектру стає важливим, щоб очистити раніше виділений спектр і зробити його доступним для 5G. Когнітивні концепції Радіо також можуть бути переглянуті для спільного використання ліцензованих та неліцензованих спектрів. І, нарешті, може знадобитися нова модель спільного використання спектра, оскільки мережа операцій з багатьма орендарями стає широко поширеною.

10. Віртуалізація радіодоступу

Останнє, але не менш важливе значення для 5G, - це віртуалізація RAN, що дозволяє обмінюватися бездротовою інфраструктурою між кількома операторами. Мережеву віртуалізацію потрібно відштовхнути від провідної базової мережі (наприклад, комутаторів і маршрутизаторів) до RAN. Для мережевої віртуалізація, інтелект потрібно вилучати з апаратного забезпечення RAN і контролювати централізовано за допомогою програмного мозку, який можна виконувати в різних мережевих шарах. Мережева віртуалізація може принести безліч переваг бездротовому домену, включаючи як Capex (Capital Expenditure), так і Opex заощадження через спільне використання мереж та обладнання, вдосконалену ЕЕ, збільшення або зменшення необхідних ресурсів за замовчуванням та збільшення мережі спритність за рахунок скорочення часу до ринку для інноваційних послуг (від 90 годин до 90 хвилин), а також легке технічне обслуговування та швидке усунення несправностей завдяки підвищенню прозорості мережі. Віртуалізація також може служити для зближення провідних та бездротових мереж, спільно керуючи цілою мережею з центральної одиниці оркестровки, і надалі підвищуючи

ефективність мережі. І, нарешті, мультирежимні RAN, що підтримують 3G, 4G або WiFi, можуть бути прийняті там, де різні радіосигнали можуть бути включені або вимкнені через центральний блок керування програмним забезпеченням, щоб покращити ЕЕ або якість досвіду (QoE) для кінцевих користувачів.

Висновки

Підсумовуючи перший розділ, можна зробити висновок що мобільні мережі рухаються досить швидкими кроками, і для того, щоб зробити досвід використання мобільними мережами ще кращим, а доступ до них ще легшим, треба імплементувати їх взаємодію з хмарними інфраструктурами. Цю взаємодію ми і розглянем у наступних розділах.

2 ХМАРНІ ІНФРАСТРУКТУРИ

2.1 Що таке «хмара»?

Термін "хмара", схоже, походить із діаграм мережі, які представляють Інтернет чи різні його частини, як схематичні хмари. "Хмарне обчислення" було створене за те, що відбувається, коли додатки та послуги переносяться в "хмару" в Інтернеті. Хмарне обчислення - це не те, що раптово з'явилося в один момент; в якійсь формі він може відстежуватись до того часу, коли комп'ютерні системи віддалено використовували часові обчислювальні ресурси та програми. Більше того, хоча в даний час, хмарне обчислення відноситься до багатьох різних видів послуг і додатків, що постачаються в інтернет-хмарі, а також про те, що в багатьох випадках пристрої, що використовуються для доступу до цих служб і додатків, не вимагають спеціальних програмних додатків.

Хмарні обчислення - це комп'ютерна парадигма, в якій великий пул систем з'єднаний у приватних або загальнодоступних мережах, для забезпечення динамічно масштабованої інфраструктури для застосування, зберігання даних та зберігання файлів. З появою цієї технології значно скоротилася вартість обчислень, хостинг додатків, зберігання вмісту та доставки.

Хмарне обчислення - це практичний підхід до досвіду отримання прямих витрат, і він має потенціал для перетворення центру обробки даних з капіталомісткого налаштування до змінної цінової середовища. Ідея хмарних обчислень базується на дуже фундаментальному принципі "багаторазового використання IT-можливостей".

Різниця в тому, що хмарне обчислення в порівнянні з традиційними поняттями "графічних обчислень", "розподілені обчислення", "комунальні обчислення" або "автономні обчислення" полягає в розширенні горизонту через організаційні межі.

Forrester визначає хмарні обчислення як:

"Набір абстрактної, високомасштабної та добре-керованої обчислювальної інфраструктури, здатної розміщувати додатки кінцевого споживача та брати оплату за використання".

Багато компаній надають послуги з хмари. Деякі помітні приклади включають в себе наступне:

Google - має приватну хмару, яку він використовує для доставки Документів Google та багатьох інших послуг своїм користувачам, включаючи доступ до електронної пошти, документи, текстові переклади, карти, веб-аналітики та багато іншого.

Microsoft - надає онлайн-службу Microsoft® Office 365, яка дозволяє переносити інструменти вмісту та бізнес-аналітики в хмару, а корпорація Майкрософт в даний час надає доступ до офісних програм у хмарі.

Salesforce.com - запускає додаток, встановлений для своїх клієнтів у хмарі, а продукти Force.com та Vmforce.com надають розробникам платформи для створення налаштовуваних хмарних сервісів.

Але, що таке хмарне обчислення? У наступних розділах зазначаються характеристики хмарних обчислень [6], моделі послуг, моделі розгортання, переваги та проблеми.

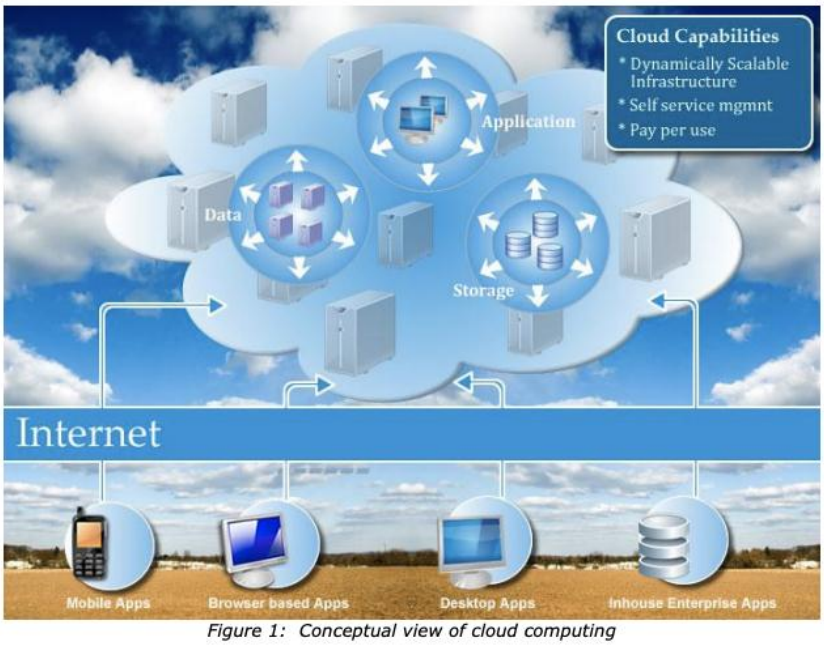

Figure 1: Conceptual view of cloud computing

Рисунок 2.1 Концептуальний вигляд хмарних обчислень

Характеристика

Хмарне обчислення має ряд характеристик, головними з яких є:

- Спільна інфраструктура - Використовує віртуалізовану модель програмного забезпечення, що дозволяє обмінюватися фізичними службами, зберіганням та можливостями мережі. Хмарна інфраструктура, незалежно від моделі розгортання, прагне максимально використовувати наявну інфраструктуру для багатьох користувачів.
- Динамічне забезпечення - Дозволяє надавати послуги на основі поточних вимог до попиту. Це робиться автоматично за допомогою автоматизації програмного забезпечення, що дозволяє розширювати та зменшувати можливості обслуговування, за потреби. Це динамічне масштабування потрібно зробити, зберігаючи високий рівень надійності та надійності.
- Доступ до мережі. Потрібно мати доступ до Інтернету з широкого кола пристроїв, таких як ПК, ноутбуків та мобільних пристроїв, використовуючи стандартні API (наприклад, на основі HTTP). Розгортання служб у хмарі включає все, від використання бізнес-додатків до найновішої програми на найновіших смартфонах.
- Кероване вимірювання - Використовує вимірювання для управління та оптимізації послуги та надання звітів та платіжної інформації. Таким чином, споживачі платять за послуги за даними, наскільки вони фактично використовуються протягом розрахункового періоду. Коротше кажучи, хмарне обчислення дає змогу обмінюватися та масштабувати розгортання послуг, коли це необхідно, з практично будь-якого місця, і для якого клієнт може платити рахунок на основі фактичного використання.

Моделі хмарних обчислень

Хмарні провайдери пропонують послуги, які можна розподілити на три категорії, представлені на рисунку 2.2:

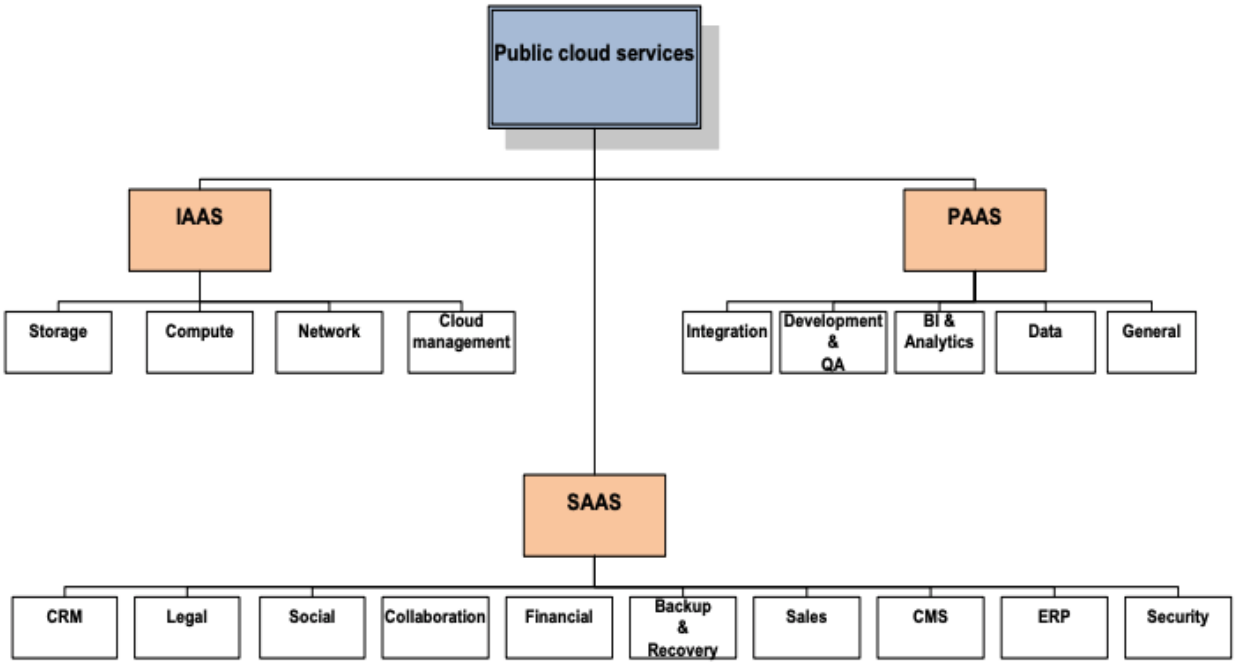

Рисунок 2.2 Сервіси публічних хмар

2.2 Інфраструктура як сервіс (IaaS)

У цій моделі повна програма пропонується клієнту, як послуга на вимогу. Один екземпляр служби працює в хмарі [7] та обслуговується кількома кінцевими користувачами. На стороні клієнтів немає потреби в авансовому інвестуванні в сервери або ліцензії на програмне забезпечення, тоді як для постачальника витрати знижуються, оскільки лише одна програма повинна бути розміщена та підтримується. Сьогодні SaaS пропонують такі компанії, як Google, Salesforce, Microsoft, Zoho та ін. У цій моделі споживачі купують можливість доступу та використання програми або служби, розміщеної в хмарі. Показовим прикладом цього є Salesforce.com, як обговорювалося раніше, де необхідна інформація для взаємодії споживача та сервісу розміщується як частина сервісу в хмарі.

На наступному малюнку перераховані найважливіші учасники ринку в кожній з областей IAAS.

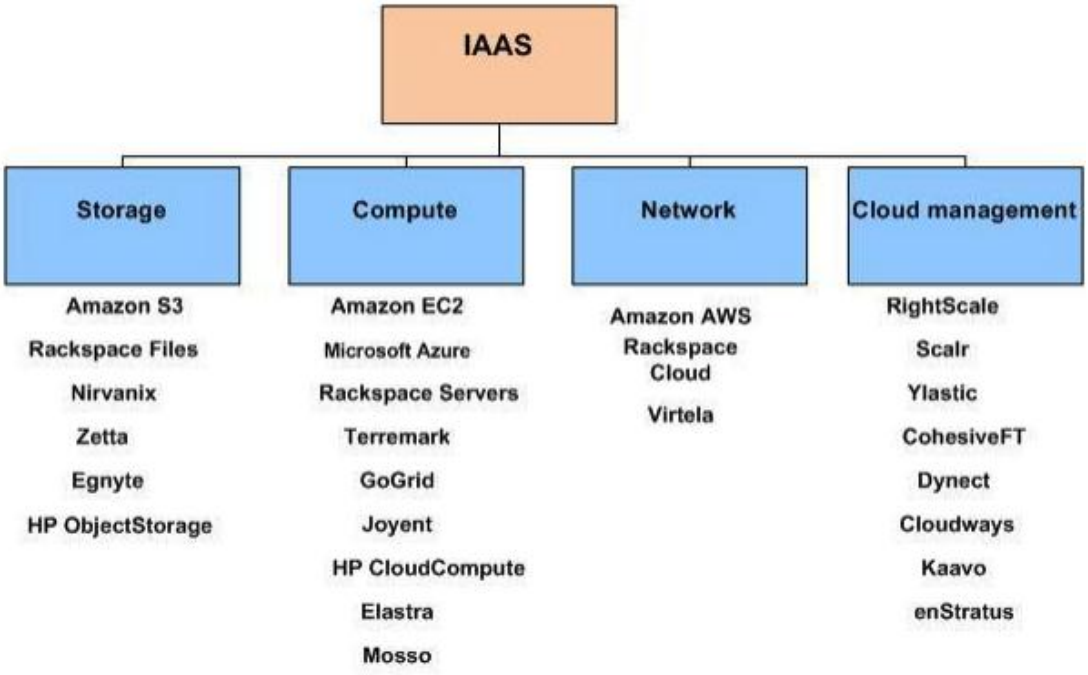

Рисунок 2.3 IaaS публічні сервіси

IaaS: Сховище даних

Служби зберігання дозволяють [7] компаніям зберігати дані на пристроях зберігання сторонніх провайдерів. Cloud Storage доступний в Інтернеті та представлений клієнтам як набір басейнів для зберігання даних або ковшів, доступних за допомогою багатих інтерфейсів, таких як API-програми, веб-інтерфейси або інструменти командного рядка. Складність хмарної архітектури зберігання приховується від клієнтів, але на задній панелі вона досить складна - зазвичай складається з розподілених пристроїв зберігання даних, якими керує централізоване програмне забезпечення. У складному програмному забезпеченні для зберігання даних використовуються алгоритми, що управляють даними, розподіленими на декількох пристроях зберігання даних.

Проте хмарне сховище може не відповідати потребам кожної організації. Потенційні недоліки включають латентність мережі, залежність від доступності Інтернету, міркування щодо безпеки та обмежений контроль. Затримка мереж вище, ніж у домашньому сховищі, оскільки дата-центр постачальника хмар розташовується в іншому географічному розташуванні, принаймні, у кількох мережевих хапах із місцезнаходження клієнта. Клієнт, який зберігає всі дані в

загальнодоступній хмарі та не має локальної копії, повністю залежить від підключення до Інтернету. Постачальник хмар повинен запропонувати високий рівень безпеки, щоб уникнути втрати інформації або компромісу, а передача даних повинна бути зашифрована.

Характеристики типового обласного зберігання:

- Високо надійний і зайвий
- Автоматично масштабований
- Служба самообслуговування доступна для клієнтів
- Доступні через багаті інтерфейси (веб-консоль, API, CLI)
- Модель оплати за вашим рахунком

IaaS: Обчислення

Обчислювальні служби забезпечують обчислювальні ресурси для клієнтів. Ці сервіси включають процесор, оперативну [7] пам'ять (RAM) та ресурси вводу / виводу. Обчислювальні параметри ціноутворення ресурсів можуть відрізнятися між різними постачальниками, проте, як правило, варіанти ціноутворення визначаються сумою обчислювальних ресурсів та загальними моделями оплати. Обчислювальні ресурси пропонуються як приклади віртуальних машин, типи екземплярів і призначені ціни залежать від комбінації процесора, оперативної пам'яті та місткості вводу / виводу. Провайдери пропонують декілька типів екземплярів, які охоплюють більшість користувацьких облікових записів та полегшують вибір клієнта (наприклад, малі, середні, великі тощо).

IaaS: Мережа

Існує два основні мережеві сервіси, що пропонуються державними [7] постачальниками хмар: балансування навантаження та DNS (системи доменних імен). Детальні технічні характеристики цих послуг виходять за рамки цього документа, але ми коротко вказуємо на ці технології, щоб визначити контекст для подальшого аналізу.

Розподіл навантаження

Балансування навантаження забезпечує єдину точку доступу до декількох серверів [7], які працюють за ним. Балансування навантаження - це мережеве пристрій, що розподіляє мережевий трафік між серверами, використовуючи спеціальні алгоритми розподілу навантаження. Існують різні алгоритми балансування навантаження, хоча найбільш популярними є наступні:

- Round-robin: навіть розподіл зв'язку на всіх серверах
- Weighted round-robin: розподіл з'єднання пропорційно вазі, призначеному для кожного сервера
- Dynamic round-robin: подібно до зваженого круглого столу, але вага сервера динамічно визначається на основі безперервного моніторингу сервера
- Least connections: з'єднання надсилається на сервер з найменшою кількістю поточних з'єднань
- Fastest: поширює нові з'єднання на сервер на основі швидкого відповіді часу сервера

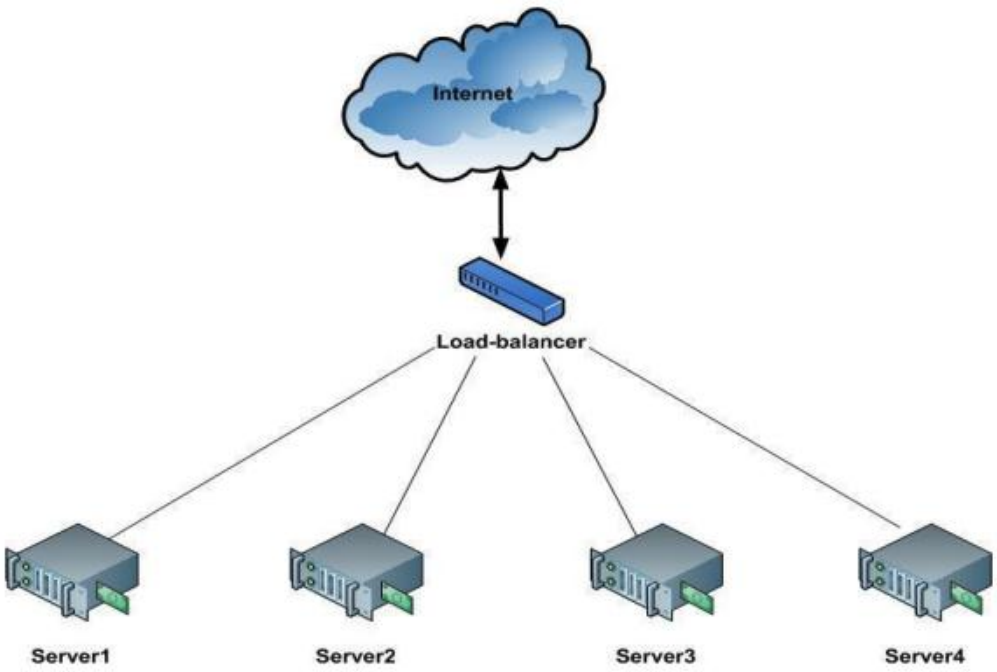

Рисунок 2.4 Схематичне зображення розподілу навантаження.

Є кілька переваг від балансу навантаження: відмова від передачі даних - у разі виникнення певної помилки сервера, балансувальник навантаження

автоматично переведе мережевий трафік на інші сервери; продуктивність - оскільки завантаження трафіку розподіляється між декількома серверами, час відгуку на мережу, як правило, є швидшим; масштабованість - клієнти можуть швидко додавати сервери під балансувальником навантаження, щоб збільшити обчислювальні потужності, не впливаючи на інші компоненти мережі / системи.

IaaS: Управління хмарою

Постачальники послуг Cloud Management - це гравці з нішевими ринками, які допомагають клієнтам спростити управління хмарами. Служби керування хмарою (CMS), як правило, заповнюють деякі розриви інструментів керування хмарою великих постачальників хмар. Найважливіші постачальники хмарності зосереджуються на основному бізнесі, надаючи клієнтам масштабовану та надійну хмарну інфраструктуру, але в деяких випадках постачальники можуть бракувати ресурсів (або нахилів) для розробки додаткових послуг керування хмарою. Великі постачальники хмарних мереж зазвичай використовують підхід "досить добре" для додаткових інструментів керування хмарою, оскільки ці інструменти безпосередньо не сприяють надходженню. Провайдери CMS працюють безпосередньо з певними постачальниками хмар для інтеграції послуг постачальника з існуючим хмарою клієнта або для створення додаткових послуг на додаток до існуючої інфраструктури постачальника. Клієнти отримують вигоду від додаткових можливостей, наданих службами керування хмарою, які можуть включати послуги інтеграції, служби безпеки та контролю доступу, функції високої доступності та послуги реплікації баз даних. CMS можна розділити на три основні гілки: служби керування додатковими ресурсами хмар, служби інтеграції з хмарами та брокерські програми для хмарних сервісів.

2.3 Платформа як сервіс (PaaS)

Тут шар програмного забезпечення або середовища розробки інкапсулюється та пропонуються як служба, на якій можуть бути побудовані інші вищі рівні обслуговування. Клієнт має право створювати власні програми, які працюють на інфраструктуру постачальника. Для задоволення вимог керованості та

масштабованості додатків, постачальники PaaS пропонують попередньо встановлену комбінацію серверів ОС та додатків, таких як платформа LAMP (Linux, Apache, MySql і PHP), обмежені J2EE, Ruby та ін. Google App Engine, Force.com тощо - це деякі з популярних прикладів PaaS. Тут споживачі купують доступ до платформ, дозволяючи їм розгорнути власне програмне забезпечення та програми в хмарі. Операційні системи та доступ до мережі не керуються споживачем, і можуть існувати обмеження щодо того, які програми можуть бути розгорнуті. Приклади включають Amazon Web Services (AWS), Rackspace та Microsoft Azure.

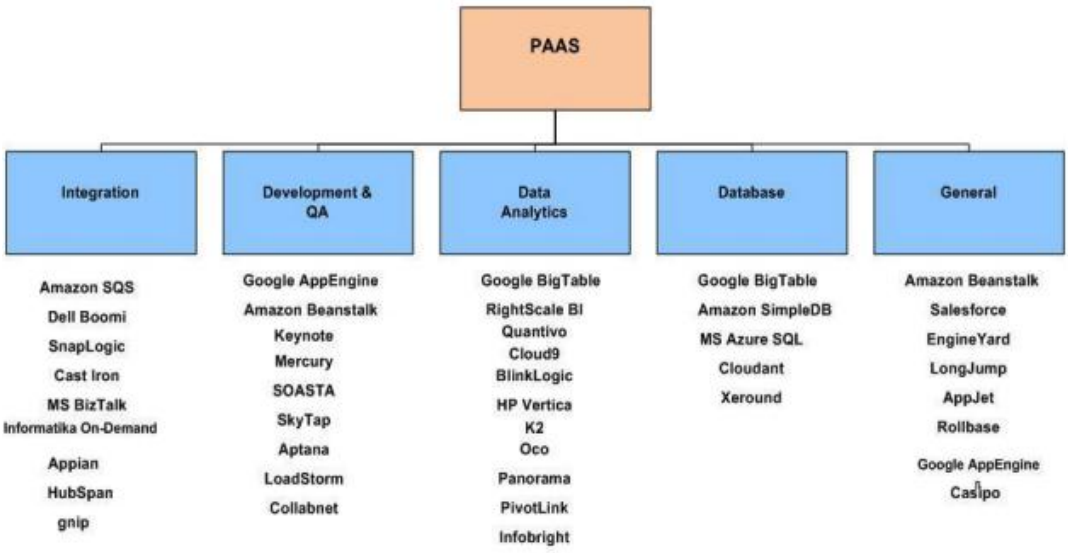

Рисунок 2.5 PaaS публічні сервіси

Характеристики сервісу PaaS

- Масштабованість та автоматичне надання базової інфраструктури
- Безпека та надмінність
- Інструменти для розробки та розгортання для швидкого управління та розгортання додатків
- Інтеграція з іншими компонентами інфраструктури, такими як веб-сервіси, бази даних та LDAP
- Багаторазова оренда, платформа, яка може використовуватися багатьма одночасними користувачами
- Журнальний журнал, звітність та інструменти для коду

- Інтерфейси керування та / або API

Сегмент ринку PaaS менш зрілий, ніж IaaS - існує лише декілька великих провайдерів PaaS, а також декілька гравців із менш ринковою нішею. Хоча PaaS та SaaS можуть вимагати глибокого технічного опрацювання порівняно з IaaS, вони також не вимагають стільки початкових інвестицій в інфраструктуру. Досить легко і доступно побудувати інфраструктуру PaaS або SaaS в межах існуючих хмар IaaS великих постачальників, таких як Amazon або Rackspace. Це маршрут, за яким зазвичай слідують менші компанії, що входять на ринок PaaS і SaaS. Сегмент PaaS ще дуже розвинений, і надає значних потенційних можливостей новаторам. Величезна кількість програм програмного забезпечення та програмного забезпечення проміжного програмного забезпечення, які в даний час запускаються всередині країни, швидше за все, будуть виведені в загальну хмару в найближчі кілька років.

PaaS: Інтеграція

Платформи інтеграції Cloud допомагають компаніям інтегрувати дані з різних джерел. Ці джерела можуть включати в себе приватні приватні базі даних Cloud, публічні обласні бази даних, сторонні бази даних постачальників, системи CRM та системи обміну повідомленнями. Інтеграція даних - це дуже складне завдання, а проблеми інтеграції даних включають підтримку якості даних, сумісності та стандартів.

PaaS: Розробка та тестування

Послуги PaaS Development та QA допомагають компаніям покращити взаємодію команд якості та співпраці з командою розробників, а також прискорити розробку програмного забезпечення та безперервний цикл інтеграції. Багато розроблювальних колективів вважають за краще використовувати маневровий метод розробки програмного забезпечення над традиційною моделлю водоспадів. Метод спритної розробки передбачає високий рівень співпраці та інтенсивний цикл безперервної доставки з частими випуском кодів, тестами та розгортаннями. Сучасні команди розробки часто є віртуальними (або географічно розподіленими)

і, отже, можуть скористатися інструментами PaaS, які спрощують та оптимізують взаємодію з командою та створюють більш швидкий цикл обігу на ринку.

Як зазначалося раніше, процес розробки програмного забезпечення зазвичай займає менше часу, ніж наступне тестування, технічне обслуговування та підтримка програмного забезпечення. Це відома проблема, і розрив, який багато служб розробки PaaS працюють для вирішення. Більшість послуг розвитку PaaS орієнтовані на процес гнучкого розвитку; це не означає, що ці послуги PaaS не застосовуються до методу водоспаду, але компанії, ймовірно, не будуть реалізовані ті ж переваги, що й у гнучкої методології.

Розробка та експлуатація програмного забезпечення включає в себе сім фундаментальних етапів (див. Малюнок 21): план, код, побудова, тестування, випуск, розгортання та експлуатація. У гнучкому процесі розробки, фази від коду до випуску є дуже ітеративними, оскільки кодове видання є дуже частими і тестування є безперервним. Багато компаній використовують гнучку методологію та впроваджують нові випуски коду кілька разів на тиждень або навіть за день: "SlideShare розгортається кілька разів на день (від 2 до 20 разів ... Це відчувається ризикованим спочатку, але це насправді набагато менш ризиковано, ніж великий Якщо виникне щось не так, просто можна трикутнути причину, і якщо щось не збігається, то легко довіряти цьому коду "- Director, LinkedIn (SlideShare).

Багато частих невеликих випусків може бути більш ефективним, ніж один великий сукупний випуск:

- Швидша реакція на відгуки клієнтів про продукт
- Невеликі випуски простіше усунення неполадок і відкат
- Цикл сталого розвитку
- Постійна мотивація та увага з боку команди
- Швидка адаптація до нових вимог та тенденцій

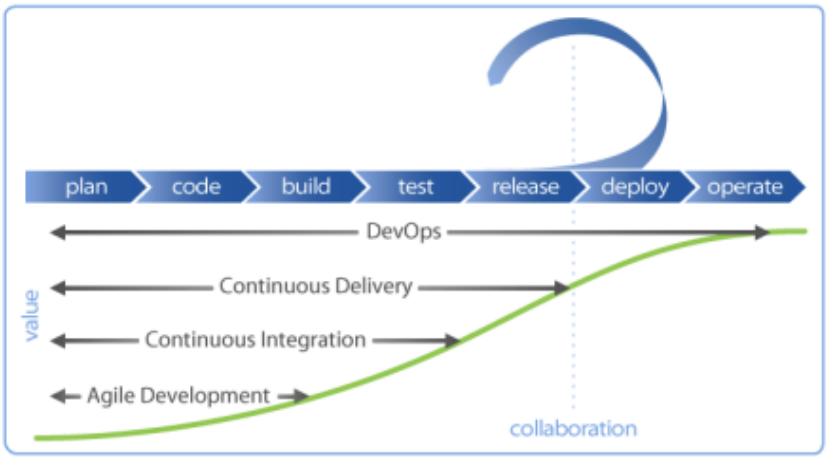

Рисунок 2.6 Процес розробки та оперування програмного забезпечення

У корпоративній організації кожний з процесів розробки та управління має окреме завдання працівника - це кілька працівників з різними функціональними ролями, які зазвичай беруть участь у процесі розробки та постачання програмного забезпечення. Однак така функціональна сегрегація обов'язків часто затримує процес доставки програмного забезпечення та підвищує шанси на випадкові помилки та помилки. У високо-сегментованих групах працівники з розробки не завжди відчувають повну відповідальність за свої дії і, як правило, залишають окремі функції іншим. Наприклад, розробники можуть відчувати, що вони не повністю відповідають за якість коду, і зосереджуються на тому, щоб їх код було передано інженерам з питань якості, з надією на те, що всі помилки потраплять до наступної команди. Зайве говорити, що така стратегія погано працює для кінцевого продукту.

2.3 Програмне забезпечення як сервіс (SaaS)

SaaS надає базові можливості для зберігання та обчислення як стандартизовані послуги в мережі. Сервери, системи зберігання даних, мережеве обладнання, простір інформаційного центру та ін. об'єднуються та доступні для обробки робочих навантажень. Замовник, як правило, розгорне власне програмне забезпечення в інфраструктурі. Деякими загальними прикладами є Amazon, GoGrid, 3 Tera і т. Д. Споживачі контролюють та управляють системами з точки

зору операційних систем, програм, зберігання та мережного підключення, але не контролюють хмарну інфраструктуру.

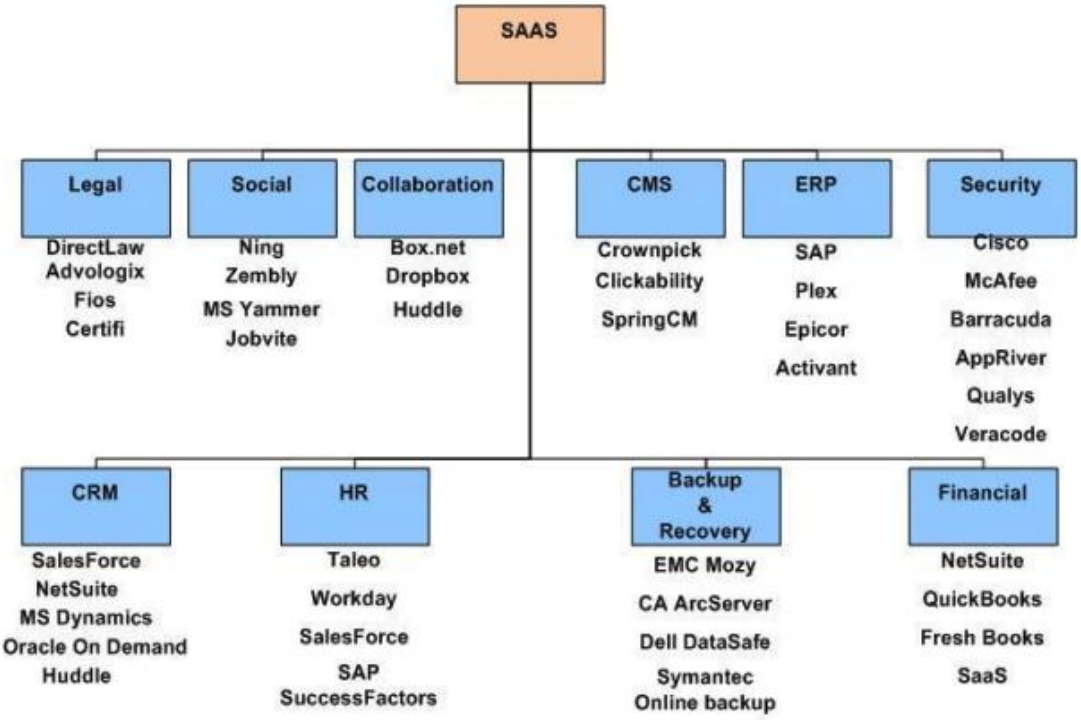

Рисунок 2.7 SaaS публічні сервіси

SaaS - це великий ринок із сильним потенціалом зростання. Група Gartner оцінює, що до 2015 року дохід від SaaS досягне 22,5 млрд доларів. Продовженням моделі ASP (постачальника додатків), основні послуги SaaS з'явилися наприкінці 90-х років, коли компанії, такі як Salesforce, запропонували клієнтам аутсорсинговий хостинг та управління своїм програмним забезпеченням. Модель ASP також пропонувала централізоване хостинг сторонніх додатків, але відрізнялося від SaaS, оскільки хостинг та операції все ще вимагали втручання вручну. Крім того, ASP іноді вимагав встановлення товстого настільного клієнта (локально запущеного програмного забезпечення, яке виконує більшість обчислювальних завдань на комп'ютері користувача, а не на стороні сервера), тоді як програми SaaS, як правило, доступні через веб-переглядачі або мобільні додатки.

Платформи SaaS використовують архітектури, що складаються з декількох орендарів, де таке ж платформне обладнання та програмне забезпечення можна

розподілити між декількома клієнтами. Хостинг та сервісна автоматизація на декількох ділянках дозволяє постачальникам послуг SaaS зберегти низькі витрати на обслуговування. Залежно від постачальника SaaS та типу програми деякі налаштування програми часто доступні, але це, як правило, є більш обмеженим, ніж за допомогою хмарних рішень PaaS та IaaS. Початкові ідеї SaaS викликали серйозне занепокоєння щодо безпеки, продуктивності та доступності сервісів. Проте протягом багатьох років SaaS технології та бізнес-моделі суттєво дозріли, і подолали багато з цих початкових турбот. Компанії зараз активно використовують SaaS, і це навряд чи зміниться.

SaaS: Виклик в бізнесі

Основною конкурентною перевагою, що забезпечує зростання SaaS, є здатність постачальників SaaS пропонувати якісні послуги за більш низькими цінами, ніж постачальники програмного забезпечення. Крім того, підприємства залучаються до послуг SaaS, оскільки такі витрати трактуються як операційні (OpEx) замість капіталу (CapEx). Як обговорювалися в попередніх розділах, компанії віддають перевагу OpEx за CapEx за податкову перевагу. Бізнес-модель SaaS може здатися ідеальною - вона залишає повний контроль над керуванням додатками в руках постачальника SaaS. Таким чином, постачальник вирішує, коли оновлювати програмне забезпечення, коли додавати нові функції, а коли піднімати ціни.

Проте, немає безкоштовного обіду, а вартість переваг бізнес-моделей SaaS. Можливості замикання можуть бути високими за допомогою деяких популярних сервісів SaaS, таких як CRM або ERP - ті, хто знайомий із системами CRM або ERP, знають, наскільки важко перейти до постачальників. Крім того, багато компаній SaaS, навіть ті, у кого є тисячі клієнтів, не можуть досягти прибутковості протягом багатьох років після початку служби. Найбільша компанія SaaS, Salesforce, мала негативну маржу чистого прибутку в останньому кварталі (2 квартал 2012 року). По суті, з 2006 року Salesforce не має квартальної норми прибутку вище, ніж 10% (див. Графік 27), хоча доходи компаній постійно зростають. Для порівняння, маржа прибутку SAP у другому кварталі 2012 року склала близько 17%.

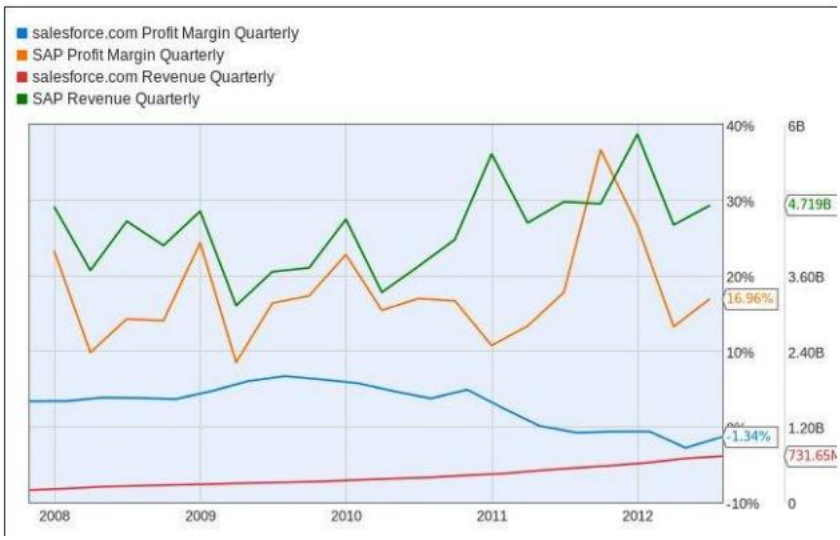

Рисунок 2.8 SAP проти Salesforce, дохід та маржа чистого прибутку

Отже, чому гігант SaaS, як Salesforce, з доходами в 2012 році в розмірі 2,3 млрд. Доларів за квартал матиме негативну маржу прибутку? Причиною є високі витрати на отримання клієнтів клієнтів (наприклад, витрати на продаж та маркетинг). Бізнес-модель SaaS відповідає принципу економ-масштабу - це низько ціна, великий обсяг бізнесу. У Salesforce в даний час близько 90 000 бізнес-клієнтів, величезна кількість. Щоб підтримувати стійке зростання свого бізнесу, Salesforce повинна підтримувати низький рівень рухливості клієнтів та постійно прискорювати темпи споживання покупців. Вони також повинні зберігати свої ціни на низькому рівні, щоб конкурувати з оригінальними постачальниками CRM, іншими гравцями SaaS та новими учасниками ринку.

	SAP (SAP)	**Salesforce (CRM)**
Revenues	14.3 billion (EUR)	2.26 billion (USD)
Cost of Goods (% of revenue)	28%	21%
Sales, marketing, and administration	**26%**	**66%**
R&D expenses	13%	13%
EBT(earnings before tax)	31%	- 1.5%
NET income	**24%**	**-0.5%**

Рисунок 2.9 Інформація про доходи

На рисунку вище ми можемо чітко бачити, що "продажі та маркетинг" є більшістю витрат Salesforce. Витрати на науково-дослідні роботи приблизно рівні

(у% доходу від доходу) між двома компаніями. Для Salesforce вартість товарів нижча, що має сенс, оскільки їх веб-рішення дешевше розробляти та підтримувати в порівнянні з встановленим програмним забезпеченням на замовлення. Salesforce також працює в централізованій, веб-інфраструктурі з кількома орендарями, яка не несе певних витрат, характерних для програмного забезпечення на основі таких програм, як підтримка багатоплатформ або інтеграція з різними внутрішніми системами інфраструктури. Чи може Salesforce вижили і стати такими успішними без таких великих інвестицій у продажі та маркетинг? SaaS і хмара взагалі є відносно новим ринком - потрібен час, щоб навчати клієнтів та встановити рівень довіри. Конкуренція є жорстокою, і хоча бар'єри для входу на ринок SaaS вище, ніж у IaaS, вони все ще недостатньо високі, щоб витримати конкурентів; Провайдери SaaS повинні зберігати низькі ціни, щоб залишатися конкурентоспроможними. Вони можуть мінімізувати витрати на розробку та доставку продукції, але в той же час вони не мають однакових можливостей отримати прибуток у консалтингових та професійних сервісах, як це робить власники програмного забезпечення. Великі постачальники програмного забезпечення, як правило, генерують 30-40% своїх доходів від послуг, тоді як Salesforce доходи від послуг складають лише 7-8%.

Salesforce, безумовно, повинен скоротити свої інвестиції в продажу та маркетинг в певний момент, коли ринок SaaS дозріє і консолідується.

2.4 Публічні і приватні хмари

Підприємства можуть вибирати для розгортання додатків на публічних, приватних або гібридних хмар [8]. Хмарні інтегратори можуть відігравати важливу роль у визначенні правильного хмарного шляху для кожної організації. Розгортання хмарних обчислень може відрізнятись залежно від вимог, а також визначено наступні чотири моделі розгортання, кожна з яких має специфічні характеристики, що підтримують потреби служб і користувачів хмар в певних напрямках

Публічна хмара - хмарна інфраструктура доступна для громадськості на комерційній основі хмарним постачальником послуг. Це дозволяє споживачеві розробляти та розгортати сервіс в хмарі з дуже невеликими фінансовими витратами

в порівнянні з вимогами щодо капітальних витрат, які зазвичай пов'язані з іншими варіантами розгортання. Публічні хмари належать і експлуатуються третіми сторонами; вони забезпечують споживачам чудову економію масштабу, оскільки витрати на інфраструктуру поширюються серед користувачів, надаючи кожному окремому клієнту привабливу, недорогу модель "Pay-as-you-go". Всі клієнти мають такий самий пул інфраструктури з обмеженою конфігурацією, захистом безпеки та відхиленнями доступності. Вони управляються та підтримуються хмарним постачальником. Однією з переваг "публічної хмари" є те, що вони можуть бути більшими, ніж об'єми хмар, таким чином забезпечуючи можливість безперебійного масштабування на вимогу.

Приватна хмара - інфраструктура хмар була розгорнута, вона підтримується та експлуатується для певної організації. Оперування може проводитись внутрішніми ресурсами компанії або третьою стороною. Приватні хмари будуються виключно для одного підприємства. Вони спрямовані на вирішення проблем із захистом даних та забезпечують більший контроль, який, як правило, відсутній у загальнодоступній хмарі. Існує два варіанти приватної хмари:

Приватна хмара як така: приватні хмари, які також називають внутрішніми хмарами, знаходяться в одному власному центрі даних. Ця модель забезпечує більш стандартизований процес та захист, але обмежується аспектами розміру та масштабованості. IT-відділи також повинні понести капітальні та операційні витрати на фізичні ресурси. Це найкраще підходить для додатків, які потребують повного контролю та налаштовування інфраструктури та безпеки.

Розміщена зовнішньо приватна хмара: цей тип приватного хмара розміщений на вулиці з постачальника хмарних мереж, де провайдер сприяє ексклюзивному хмарному середовищу з повною гарантією конфіденційності. Це найкраще підходить для підприємств, які не віддають перевагу загальнодоступному хмару за рахунок спільного використання фізичних ресурсів.

Хмара спільноти - хмарна інфраструктура поділяється між низкою організацій з подібними інтересами та вимогами. Це може допомогти обмежити капітальні витрати на його створення, оскільки витрати поділяються між

організаціями. Оперування може проводитись внутрішніми ресурсами компанії або третьою стороною.

Гібридна хмара - хмарна інфраструктура складається з безлічі хмар будь-якого типу, але хмари мають можливість через свої інтерфейси дозволяти переносити дані та / або програми з однієї хмари в іншу. Це може бути поєднання приватних та загальнодоступних хмар, які підтримують вимогу зберегти деякі дані в організації, а також необхідність пропонувати послуги в хмарі. За допомогою Hybrid Cloud постачальники послуг можуть повністю або частково використовувати сторонніх постачальників хмар, таким чином підвищуючи гнучкість обчислень. Гібридне хмарне середовище здатне забезпечити за вимогою зовнішню шкалу. Можливість збільшити приватну хмару з ресурсами загальнодоступної хмари може використовуватися для управління будь-якими несподіваними навантаженнями на робочу навантаження.

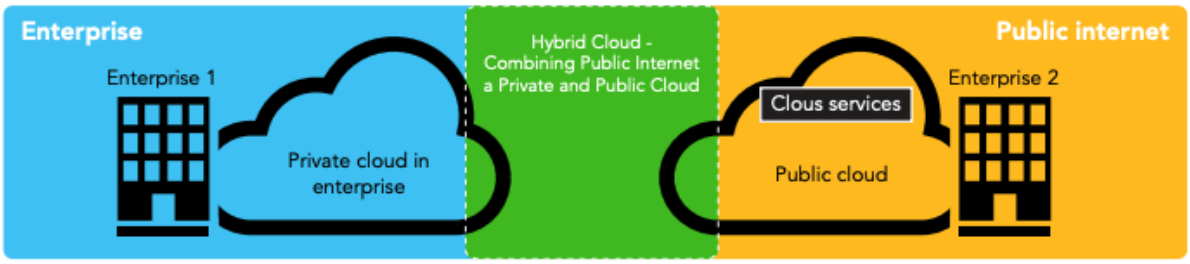

Рисунок 2.10 Приклад розгортання публічної, приватної та гібридної.

2.5 Переваги використання хмарних інфраструктур

Нижче наведено деякі з можливих переваг для тих, хто надає послуги та програми на основі хмарних обчислень.

Економія вартості - компанії можуть скоротити свої капітальні витрати та використовувати операційні витрати для збільшення їх обчислювальних можливостей [8]. Це нижчий бар'єр для входу, а також потребує меншого обсягу внутрішніх інформаційних ресурсів, щоб забезпечити підтримку системи. Платіжна модель - плата за користування; інфраструктура не купується, знижуючи технічне обслуговування. Початкові витрати та періодичні витрати значно нижчі, ніж традиційні обчислення.

Масштабованість / гнучкість - компанії можуть розпочати з невеликого розміщення та досить швидко розростати до великого розміру, а потім, якщо необхідно, зменшити масштаб. Крім того, гнучкість хмарних обчислень дозволяє компаніям використовувати додаткові ресурси в пікові години, що дозволяє їм задовольнити потреби споживачів.

Надійність - послуги, що використовують кілька резервних сайтів, можуть підтримувати безперервність бізнесу та аварійне відновлення.

Технічне обслуговування - постачальники хмарних сервісів проводять технічне обслуговування, а доступ до них здійснюється за допомогою інтерфейсу API, що не вимагає установки додатків на комп'ютерах, що значно знижує вимоги до технічного обслуговування.

Мобільний доступ - мобільні працівники збільшили продуктивність через системи, доступні в будь-якій інфраструктурі.

Збільшення обсягу зберігання - з масовою інфраструктурою, яку сьогодні пропонують хмарні постачальники, зберігання та обслуговування великих обсягів даних є реальністю. Раптові наслідки робочого навантаження також ефективно і ефективно управляються, оскільки хмара може масштабувати динамічно.

2.6 Виклики

Нижче наведені деякі важливі проблеми, пов'язані з хмарними обчисленнями, і хоча деякі з них можуть призвести до сповільнення при наданні більшої кількості послуг у хмарі, більшість з них також можуть забезпечити можливості, якщо вони будуть вирішені з належною обережністю та увагою на етапах планування. Незважаючи на зростаючий вплив [8], проблеми щодо хмарних обчислень залишаються. На нашу думку, переваги перевершують недоліки, і модель варто дослідити. Деякі загальні проблеми:

Безпека та конфіденційність - можливо, два з найбільш "гарячих" кнопок, що стосуються хмарних обчислень, стосуються зберігання та захисту даних та моніторингу використання хмарами постачальниками послуг. Ці проблеми, як правило, пов'язані із сповільненням розгортання хмарних сервісів. Ці виклики можуть бути вирішені, наприклад, зберігаючи внутрішню інформацію організації,

але дозволяючи її використовувати в хмарі. Однак, щоб це відбулося, механізми безпеки між організацією та хмарою повинні бути надійними, а гібридне хмарне середовище може підтримувати таке розгортання. Підприємства не хочуть купувати гарантію захищеності бізнес-даних від постачальників. Вони побоюються втрати даних для конкуренції та конфіденційності даних споживачів. У багатьох випадках фактичне місце зберігання не розкривається, що додають до проблем безпеки підприємств. У існуючих моделях брандмауери між даними центрами (належать підприємствам) захищають цю конфіденційну інформацію. У моделі хмар постачальники послуг несуть відповідальність за збереження безпеки даних, і підприємства повинні покладатися на них.

Відсутність стандартів - хмари мають документовані інтерфейси; однак, з цими стандартами не пов'язані, і тому малоймовірно, щоб більшість хмар були сумісними. Відкритий Grid Forum розробляє інтерфейс обчислювального інтерфейсу Open Cloud для вирішення цієї проблеми, а Open Cloud консорціум працює над стандартами та практикою обчислень у хмарі. Висновки цих груп потребують дозрівання, але невідомо, чи будуть вони задовольняти потреби людей, які розгортають служби, та конкретні інтерфейси, які потребують ці служби. Проте, якщо вживати відповідних заходів, вони зможуть використовувати їх за останні роки.

Постійно розвиваються - вимоги користувача постійно розвиваються, як і вимоги до інтерфейсів, мереж та зберігання. Це означає, що "хмара", особливо публічна, не залишається статичною і постійно розвивається.

Занепокоєння щодо дотримання вимог - закон Sarbanes-Oxley (SOX) в США та директиви щодо захисту даних в ЄС є лише двома серед багатьох проблем, пов'язаних із відповідністю хмарних обчислень, залежно від типу даних та програм, для яких використовується хмара. ЄС має законодавчу підтримку захисту даних у всіх державах-членах, проте в США захист даних різний і може відрізнятися залежно від штату. Як і раніше, це стосується безпеки та конфіденційності, що, як правило, призводить до розгортання гібридного хмарності, при цьому одна хмара зберігає дані, що входять до організації.

Відновлення даних і доступність - у всіх бізнес-додатках існують угоди про рівень обслуговування, які суворо дотримуються. Оперативні команди відіграють ключову роль у управлінні угодами про рівень сервісу та управління додатками. У виробничих середовищах підтримуються оперативні команди:

- Відповідна кластеризація та збій
- Реплікація даних
- Системний моніторинг (моніторинг транзакцій, моніторинг журналів тощо)
- Обслуговування (Runtime Management)
- Аварійного відновлення
- Управління ємністю та продуктивністю
- Якщо будь-яка з перелічених вище служб не обслуговується постачальниками хмар, збиток та вплив можуть бути серйозними.

Можливості управління - незважаючи на наявність декількох хмарних провайдерів, управління платформою та інфраструктурою все ще перебуває у зародковому стані. Наприклад, такі функції, як «автоматичне масштабування», є ключовою вимогою для багатьох підприємств. Існує величезний потенціал для покращення масштабованості та функцій балансування навантаження, наданих сьогодні.

За допомогою хмарних обчислень діяння переміщується до інтерфейсу, тобто до інтерфейсу між постачальниками послуг та кількома групами споживачів послуг. Обласні сервіси вимагатимуть експертизи у розподілених службах, закупівлях, оцінці ризиків та ведення переговорів - сфери, на яких багато підприємств користуються лише невеликими можливостями.

2.7 Зв'язок у Cloud

Для розробників послуг, надання послуг у хмарі залежить від типу послуги та пристроїв, які використовуються для доступу до неї. Процес може бути настільки ж простий, як користувач, який натискає потрібну веб-сторінку, або може включати додаток, що використовує API, що має доступ до служб у хмарі. Telcos починають використовувати хмари для випуску власних послуг і тих, що

розробляються іншими, але використовуючи інфраструктуру Telco та дані. Очікується, що інфраструктура зв'язку Telco забезпечує можливість отримання доходів.

Використання служб зв'язку

Коли у хмарі комунікаційні служби можуть розширювати свої можливості або самостійно надавати послуги, або надавати нові можливості інтерактивності для поточних послуг.

Служби зв'язку на базі Cloud дозволяють підприємствам вставляти комунікаційні можливості в бізнес-програми, такі як Enterprise

Системи керування ресурсами (ERP) та управління взаємовідносинами з клієнтами (CRM). Для людей, що перебувають у дорозі, діти можуть отримувати доступ до них за допомогою смартфону, що підтримує підвищену продуктивність, коли вони знаходяться на відстані від офісу.

Ці сервіси перевершують підтримку розгортання сервісів VoIP-систем, систем співпраці та систем конференц-зв'язку для голосового та відеозв'язку. Вони можуть бути доступними з будь-якого місця та пов'язані з поточними службами, щоб розширити свої можливості, а також самостійно надавати послуги.

З точки зору соціальних мереж, використання хмарних комунікацій забезпечує можливості натискання викликів із сайтів соціальних мереж, доступ до систем миттєвого обміну повідомленнями та відеозв'язку, розширення взаємозв'язку людей у соціальному колі.

2.8 Доступ до веб-інтерфейсів

Доступ до можливостей зв'язку в обласній середовищі досягається за допомогою API-інтерфейсів [8], в першу чергу, інтерфейсів API RESTful Web 2.0, що дозволяє розробляти додатки поза межами хмари, щоб скористатися перевагами інфраструктури зв'язку в ній.

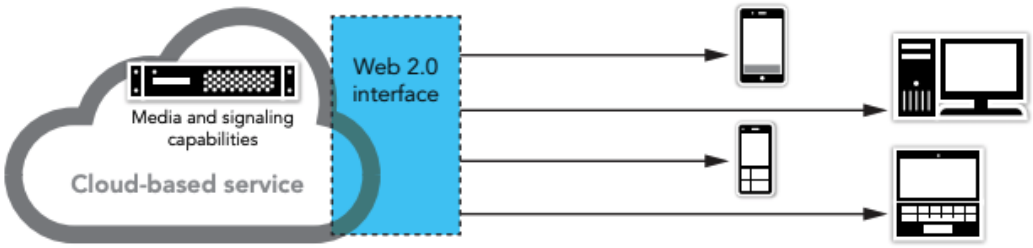

Рисунок 2.11 Веб-інтерфейси в хмарі.

Ці API відкривають широкий спектр можливостей зв'язку для хмарних служб, обмежуючись обмеженими можливостями медіа та сигналізації в межах хмари. Сьогоднішні медіа сервіси дозволяють спілкуватися та керувати голосом та відео через складний асортимент кодеків та типів транспорту. Використовуючи веб-API, ці складнощі можна спростити, а засоби масової інформації можуть бути легко доставлені на віддалене пристрій. API також дозволяють спілкуватися з іншими службами, надаючи нові можливості та допомагаючи керувати середніми доходами на одного користувача (ARPU) та швидкістю вкладень, особливо для Telcos.

2.9 Інтерфейси керування медіа-сервером

Під час створення можливостей зв'язку в «ядро хмари», де їм буде доступ іншої служби, можна використовувати API Web 2.0, а також комбінацію SIP або VoiceXML та стандартних інтерфейсів керування мультимедіа, таких як MSML, MSCML , і JSR309. Комбінації забезпечують різні набори можливостей, але з MediaCTRL розробляється в Робочій групі Інтернету (IETF), очікується, що MediaCTRL замінить MSML і MSCML і збільшить кількість наявних даних та інших подій після його ратифікації. JSR309 є відомим вибором для тих, хто шукає розробку Java, оскільки забезпечує інтерфейс Java для керування медіа.

На наступному рисунку наведено приклад доступу до служб у хмарі через інтерфейси інтерфейсу Web 2.0 та інтерфейсу керування мультимедіа.

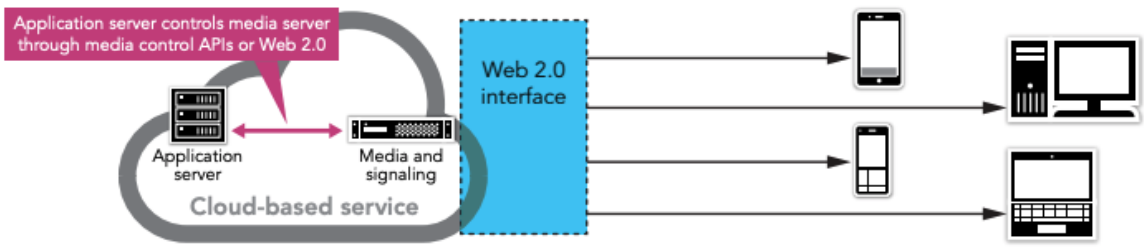

Рисунок 2.12 Доступ до засобів зв'язку зсередини Cloud

Незалежно від того, чи підприємства розгортають комунікаційні служби для доступу з-за меж хмарної зони або в межах неї, середовище є тим, яке підтримує швидкий розвиток та впровадження цих можливостей.

Масштабованість зв'язку

Щоб забезпечити вимоги щодо масштабованості для розгортання в обласному середовищі, програмне забезпечення для зв'язку має бути здатним працювати у віртуальних середовищах. Це дозволяє легко збільшувати та зменшувати щільність сеансів, виходячи з потреб того часу, одночасно зберігаючи вимоги до фізичних ресурсів на серверах до мінімуму.

Висновки

З другого розділу видно, що хмарні технології дуже широко використовуються в наші дня для надання будь-яких послуг, роблять можливими ряд опцій для імплементування сервісів, що означає велику кількість варіантів взаємозв’язку між хмарними технологіями і мобільними інфраструктурами. Саме види взаємозвязку між мобільною мережею та хмарною інфраструктурою, їх взаємовикористання для покращення роботи і швидкості взаємодії з мережею.

3 ІСНУЮЧІ МОДЕЛІ ОБМІНУ ДАНИМИ ТА РІШЕННЯ НА ОСНОВІ ХМАРНИХ ТЕХНОЛОГІЙ

Зростання доходів у зрілих мобільних мережах є обмеженим, або загальний прибуток галузі може навіть зменшитися. У цій ситуації єдиним способом збільшення грошових потоків є зменшення операційних витрат (OPEX), а також майбутніх капітальних видатків (CAPEX). Розподіл інфраструктури між операторами мобільної мережі (MNOs) та операторами мобільних віртуальних мереж (MVNOs) є основною і добре знайомою моделлю, яка зменшує обидва OPAEX та CAPEX. Наприклад, обмін мережами між AT & T та T-Mobile в США [9], 3G RAN, який використовується між T-Mobile & 3 у Великобританії, Vodafone & 3 у Швеції та Orange & Vodafone в Іспанії. Це значне спільне використання призводить до 43% економії в CAPEX та 49% у OPEX. Окрім того, запланована економія CAPEX на розподілі інфраструктури на Близькому Сході та в регіоні Африки становить 8 мільярдів доларів. MVNOs є важливим гравцем для спільного використання інфраструктури, а також приносять бізнес для MNOs. В додаток, ринки MVNOs ростуть швидше, і за оцінками CAGR абонента MVNO на 10% протягом наступних п'яти років. Фактично, MVNOs вимагають найнижчих інвестицій (спільна інфраструктура з MNOs (тобто мережа доступу)), а також вимагає дуже короткого часу для вступу на ринок. З іншого боку, нова тенденція почала знижувати OPEX, а також майбутні CAPEX, такі як віртуалізація та мережа хмарних мереж (наприклад, C-RAN та vEPC). Передбачається, що віртуалізація та мобільна хмара становлять ринок до 400 мільйонів доларів до 2018 року [10]. Основна ідея віртуалізації та технології хмарних технологій полягає в тому, щоб запускати пристрої на серверах великого обсягу, а не працювати на спеціальному фірмовому апаратному забезпеченні (тобто на фізичних монолітних пристроях, залежних від постачальників). Наприклад, в рамках проекту C-RAN, проведеного компанією China Mobile, є централізація всіх функцій BBU, 2-го і 3-го рівня, а також оцінка того, що скорочення OPEX та CAPEX складе 53% та 30%, включаючи енергію та підтримку мережі та об'єднання бази групові одиниці.

3.1 Хмарна мобільна мережа

Переміщення мобільної мережі на хмарних платформах дає змогу мобільним операторам перейти до зовсім іншої парадигми мережі, де такі структурні підрозділи, як BBU, MME, SGW та PGW, реалізуються програмами, що працюють на IT-апаратурі, що є частиною хмарна інфраструктура. Всі ці функції пов'язані з конкретним місцем розташування та динамічно збільшуються / зменшуються на основі вимог часу. Наприклад, деякі функції можуть зменшуватись, переміщаючи активне навантаження на іншу таку ж функцію в тому самому місці або в іншому хмарному центрі, не викликаючи жодних перешкод для користувачів. Крім того, наявність статичної функціональності суб'єктів може бути більш динамічною реалізацією, заснованою на час доби.

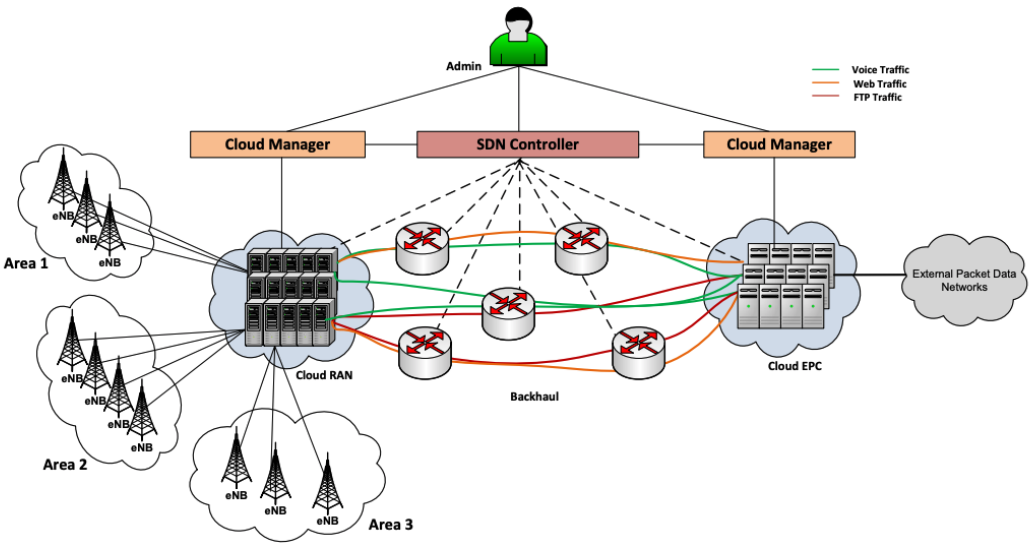

Рисунок 3.1 Мобільна мережа на базі хмарної інфраструктури

На малюнку вище показано запропоновану на хмарній основі мобільну мережу. Хмара RAN підключається до різних eNode Bs в басейнах з одного боку, а з іншого боку, вона з'єднується з Cloud EPC. Дані та керування трафіком з UE переходить до відповідної областності EPC через зворотний тракт. Ця хмарна інфраструктура, керована хмарним менеджером і зворотним трактом, яка включає перемикачі та маршрути, управляється контролером програмного забезпечення (SDN). Адміністратор переглядає та запрограмує мережу через контрольну платформу, таку як обласний менеджер та контролер SDN, використовуючи

інтерфейс графічного інтерфейсу користувача (GUI) / інтерфейс прикладного програмування (API), а адміністратор також може наказати мережі змінити поведінку мережі або змінити розташування або виділити більше ресурсів на конкретний сайт. Суб'єкти контрольної платформи, такі як обласний менеджер та SDN-контролер, мають інтерфейс між собою. Ці інтерфейси обмінюються повідомленнями, якщо будь-які зміни у поведінці мережі або будь-якій мережі об'єкта збільшуються / зменшуються в хмарі, особливо в обласній EPC. Наприклад, якщо будь-яка особа в обласній EPC переміщується в інше місце в межах одного хмара чи іншої хмари поблизу місцеположення для кращого QoE. У цьому випадку контролери будуть динамічно керувати активними сесіями до нового веб-сайту хмари без будь-яких переривань.

3.2 Хмарна RAN, мережа радіодоступу

Cloud RAN (C-RAN) складається з централізованих програмних базових частотних блоків (BBUs) та розподілених недорогих радіочастот (RRH), а також антен, розташованих на віддаленому сайті, наприклад eNB. RRH перетворює сигнали цифрового базового сигналу з BBU і складається з пристроїв РЧ (AMP) та блоків обробки сигналів, включаючи цифровий аналоговий перетворювач. BBU відповідає за обробку цифрових базових сигналів, і це є точкою завершення для IP-пакетів і сигналів основної смуги, як показано на малюнку нижче. Наприклад, сигнали основної смуги, отримані з віддалених комірських сайтів, демодульовані, а IP-пакети передаються в EPC. ББК та блок віддаленого мобільного пристрою пов'язані один з одним за допомогою оптичного інтерфейсу стандартного інтерфейсу CPRI (Common Public Radio Interface) [11].

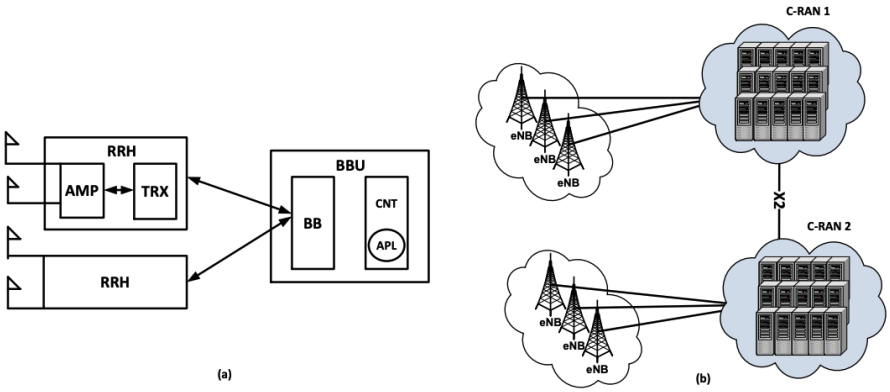

Рисунок 3.2 (а) eNB архітектура, (б) Кожна платформа з’єднана з іншою за допомогою інтерфейсів X2

На платформі C-RAN, заснований на обсязі трафіку, який може обробляти BBU, він може контролювати один або декілька одиниць RRH. На додаток до цього, у хмарній платформі потужність, споживана кондиціонером та кількістю приміщень обладнання на об'єктах, може суттєво зменшитися. З іншого боку, ці централізовані BBU можуть збільшуватися / зменшуватись, щоб максимізувати ресурси. Наприклад, якщо оператор хоче збільшити ємність мережі, традиційним способом оператор збільшить продуктивність, встановивши новий сайт клітинки, включаючи всі обладнання RAN, такі як BBUs, RRU та ін. У C-RAN оператору потрібно лише для розгортання антени та RRU і підключення до басейну BBU на платформі для хмар. Оператору потрібно лише оновити обладнання басейнів BBU, коли збільшується потужність обробки мережі. Крім того, будь-які оновлення в радіоінтерфейсі можуть бути легко виконані за допомогою програмного забезпечення, яке встановлено радіо. Отже, оператори можуть отримати вигоду з точки зору OPEX та CAPEX, а також зменшити емісію вуглецю шляхом зменшення енергії (шляхом відключення вибраної базової станції протягом ночі) та кількості базових станцій. Крім того, централізований підхід покращує ефективність роботи базових станцій при динамічному керуванні навантаженням, а також забезпечує нові можливості для бізнесу з точки зору спільного використання з багатьма операторами.

3.3 Хмарне EPC, розвинуте пакетне ядро

Хмарне EPC складається з усіх компонентів EPC, таких як MME, SGW, PGW та HSS [12]. Ці компоненти все ще підтримують однакові стандартні інтерфейси між ними щоб відповідати стандарту 3GPP. Ці компоненти можуть бути складені в єдину службову функцію (SF), включаючи MME, SGW та PGW, або вони можуть бути незалежними функціями. У нашому випадку ми припустили, що кожен компонент складається з однієї функції (віртуальна машина), і ця функція працює на вершині виділеного обладнання в хмарній платформі, як показано на рисунку 3.3(a). Менеджер VM знаходиться між віртуальними машинами та апаратним забезпеченням, і діє як гіпервізор між програмним та апаратним забезпеченням.

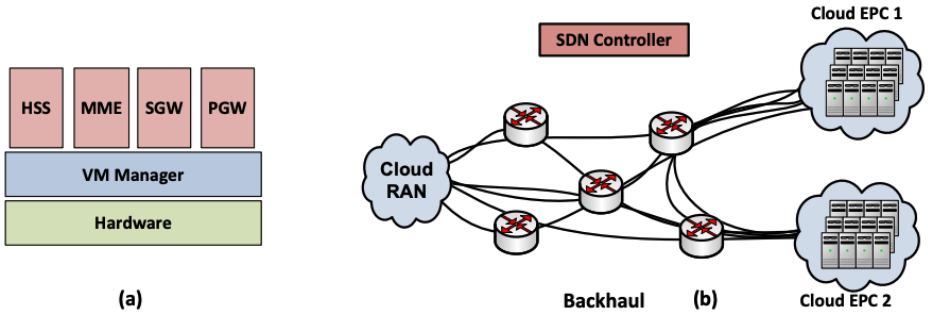

Рисунок 3.3 (а) Приєднані апаратні EPC компоненти в віртуальних машинах, (б) оборот трафіку між хмарними EPC з використанням SDN контроллеру впродовж того, як елементи віртуальних машин рухаються між хмарами.

У хмарній платформі важливо перенести контекст користувачів або файл віртуального зображення з одного місця хмарності в інше місце, щоб оптимізувати ефективність мережі та зменшити навантаження при перевантаженні. У випадку EPC існує потреба в передачі інформації про активний або незайнятий профіль користувачів від місця розташування в хмарі до іншого кращого місця розташування, яке розташоване поблизу місця розташування користувача (див. малюнок вище). Наприклад, в існуючих компонентах EPC, в умовах перевантаження дуже важко скоротити завантаження через статичну поведінку кожного компонента. Дійсно, 3GPP також запропонував неприскорене рішення щодо збалансування навантаження між компонентами. У хмарній платформі

хмарний менеджер завжди відстежує завантаження кожної віртуальної машини. Якщо будь-яка функція перевантажена, диспетчер облаків негайно розгортає нову віртуальну машину та переміщує активних користувачів до нової розгорнутої функції. У цьому випадку контекст активних користувачів буде переносити з старої віртуальної машини на нову розгорнуту віртуальну машину без переривання активних сеансів. Розгортання VM може бути в одному місці або в іншому місці. Однак це залежить від місцезнаходження UE та наявних ресурсів у хмарі. Насправді, розгортання буде близьким до UE, щоб зменшити затримку. Контролер SDN керує мережею зворотної мережі на основі інформації, отриманої від хмарних менеджерів. Наприклад, якщо зображення компонент EPC переміщено з одного хмарного центру в інший хмарний центр, контролер SDN динамічно керує активним трафіком до нового місцезнаходження за допомогою протоколу OpenFlow. Протокол OpenFlow є початковим протоколом, який застосовує концепцію SDN. Це дає змогу віддаленому контролеру на основі програмного забезпечення керувати підключеними комутаторами OpenFlow через чітко визначений набір інструкцій для переадресації. Істотним є те, що EPC хмар переходить від сьогоднішніх статичних розгортків до високодинамічних мережевих реалізацій, таких як динамічне забезпечення та конфігурація.

3.4 Cloud Manager, хмарний менеджер

Хмарний менеджер може приймати рішення щодо використання віртуальних ресурсів. Це рішення ґрунтується на всіх параметрах (наприклад, політиці орендарів, розташуванні користувачів / центрі обробки даних) та діє в напрямі копії (cc) для їх виконання. Хмарний менеджер налаштує SF під час масштабування вхід / вихід. Наприклад, виконуючи SF в мережі, менеджер запускає існування цього SF з іншими SF, щоб маршрут трафіку. Функція моніторингу в обласному диспетчері буде відслідковувати про пропускну спроможність, що запитується в певній географічній зоні в певний час, агрегованою групою користувачів. Дійсно, менеджер буде відповідальним суб'єктом для початкової конфігурації кожного SF, включаючи базову мережу та конфігурування параметрів сервісу, таких як

параметри 3GPP LTE / EPC (наприклад, ідентифікатори, політика QoS, параметр PCRF тощо) [12].

3.5 Моделі спільного використання мобільних мереж на основі Cloud

Обмін мережною інфраструктурою є альтернативним рішенням для операторів мобільного зв'язку для зменшення CAPEX та OPEX Спільна інфраструктура в хмарних мережах перетворює телекомунікацію на зовсім іншу мережеву парадигму, де оператори можуть розгортати мережеві компоненти на підставі запитів від орендаря, це може бути короткотерміновим або довготривалим. Основними гравцями для обміну мобільними мережами є між операторами або MVNOs.

3.6 Pay as you go, плати за використання

Розподіл інфраструктури між операторами, безсумнівно, призводить до зменшення інвестицій кожного оператора, що бере участь у процесі спільного використання мережі. Дійсно, існуючі моделі обміну мережею є чисто статичними та стаціонарними ресурсами. Наприклад, оператор мобільного зв'язку (орендатор) створює угоду про рівень обслуговування (SLA) з власником інфраструктури (тобто операторам мобільних телефонів є інфраструктура) для фіксованої кількості використання, наприклад, пропускної здатності та фіксованого періоду часу, наприклад протягом одного року. На основі SLA власник розробляє свою мережу на основі навантаження, передбаченого в години пік. Дизайн мережі грунтується на спеціальному апаратному забезпеченні, і його потрібно статично забезпечити та налаштувати. Отже, мережева архітектура не набуває еластичності та функцій на вимогу. Наприклад, BBUs розгортаються в клітинному сайті на основі пікової навантаження та SLA. Це розгортання - це статичне та нестандартне устаткування, і дуже важко збільшити продуктивність на основі попиту на трафік. Крім того, важко оптимізувати ресурс під час не пікових годин, це призводить до втрати ресурсів. Ці витрати настільки дорогі, і це безпосередньо впливає на OPEX. З іншого боку, в існуючих моделях обміну, орендар не може придбати додатковий ресурс за певний період часу від власника через стабільну мережеву

функціональність та обмежені ресурси. Отже, кінцевий користувач стикається з поганим QoE протягом пікових годин. Хмарна мобільна мережа, оператор може розділити частину мережі на певні періоди часу. Наприклад, у мережі може виникнути будь-яке пошкодження мережі, скажімо, SGW (обслуговуючий шлюз) в мережі LTE. У цьому випадку оператор може зажадати нову SGW за лічені хвилини від оператора хмарності та налаштувати за допомогою своєї мережі, замість того, щоб зберігати мережу впродовж певного періоду часу. Після відновлення відновлення, виділені шлюзові випуски та відповідний ресурс, такі як процесор та пам'ять, можуть бути використані для інших служб. Власник сплачуватиме орендар на основі використаних сервісних ресурсів порівняно з усією інфраструктурою (тобто, орендарем буде той спосіб, що він використовував такий ресурс, як «Оплата за вами»). Ця модель впровадить нові бізнес-стратегії, такі як короткостроковий обмін (тобто годину на основі попиту).

3.7 RAN-as-a-Service (RANaaS)

Моделі RANaaS відбивають нову бізнес-парадигму для операторів мобільного зв'язку. Справді, протягом десятиріччя загальний доступ до мережі при збереженні логіки управління всередині контролера RAN кожного оператора незалежно один від одного. Цей тип рішення може бути складним для оптимізації ресурсу під час не пікових годин, що може призвести до втрати ресурсів. RANaaS дозволить ефективно використовувати ресурси центрів обробки даних за допомогою динамічного розгортання необхідних ресурсів. Отже, MNOs (постачальники хмарних сервісів) можуть надавати RAN за вимогою та пружним способом [12]. У моделі RANaaS орендар може бути ініційований для виконання вузлів RAN в конкретному стільниковому сайті протягом певного періоду часу. Наприклад, через несподівані події (наприклад, футбольні ігри, публічні страйки тощо) у конкретному місці розташування, арендатору потрібні додаткові ресурси в цих конкретних місцях, щоб уникнути перевантажень або збоїв у мережі. Орендатор може вимагати додатковий ресурс (наприклад, BBUs та RRU) для певних сайтів стільникового зв'язку за певний період часу. Власник ініціює вимогу орендаря на вказаному сайті клітинки, хмарний менеджер налаштовує ці вимоги та

виконує їх у мережі. Наприклад, хмарний менеджер буде виконувати нові BBU та RRU для певного веб-сайту клітинки в хмарі та налаштувати ці пристрої в мережі, включаючи мережу з основною мережею орендарів. Після завершення вимог надаються ресурси для орендаря, будуть звільнені та стягуються за користування ресурсами, і ці ресурси можуть бути використані для інших послуг.

3.8 EPC-as-a-Service

Помутніння EPC створює можливість для MNOs та MVNOs перейти до зовсім іншої мережевої парадигми, де мережеві функції, які раніше використовувалися для реалізації на фізичних коробках, розгорнуті в певних точках присутності (PoPs), стають робочими навантаженнями, що працюють на вершині хмарна інфраструктура. Постачальники хмарних сервісів (CSP) надаватимуть об'єкти EPC як службу (тобто HSS, MME, SGW та PGW як службу), запускаються у віртуалізованому середовищі, який реалізується у вигляді робочих навантажень на платформі хмар (подібно до хмарної мережі) . Наприклад, CSP може інтегрувати функції, поєднуючи послуги, що надаються MVNOs, і створити конкретну хмарну мережу MVNO у хмарній платформі. Таким чином, MVNO може отримати вигоду на низькому рівні CAPEX і OPEX у своїй діяльності замість створення власної фізичної мережі (наприклад, низькі CAPEX та OPEX для побудови та підтримки сервера додатків, білінгових одиниць та основного обладнання тощо).

3.9 Аналіз інвестицій в мобільну мережу та переваги моделей обміну

Мобільні оператори, агресивні переслідування слабких ділових моделей, призвели до еволюції, перетворившись на обмін інфраструктурою як життєздатний варіант. Інвестиції на постійно мінливі технології та посилення конкуренції між глобальними та віртуальними операторами примушували операторів мобільного зв'язку до нових шляхів розробки бізнес-стратегій та підтримання низької вартості мережі. Важливо, що збереження може бути передачею для оновлення своїх мереж та забезпечення кращого розгортання та покриття для кінцевих користувачів.

Інвестиції на пасивні елементи, такі як фізичні об'єкти, будівлі, притулки, башти, джерело живлення та запасне резервне копіювання, складатимуть 20% і залишиться на 20% у базовій мережі. Таблиця 1 показує CAPEX для окремих елементів. Таблиця. 2 показає типовий індивідуальний OPEX у мобільній мережі [13]. У країнах з перехідною економікою та країнами, що розвиваються ці значення можуть відрізнятися залежно від недостатньої інфраструктури.

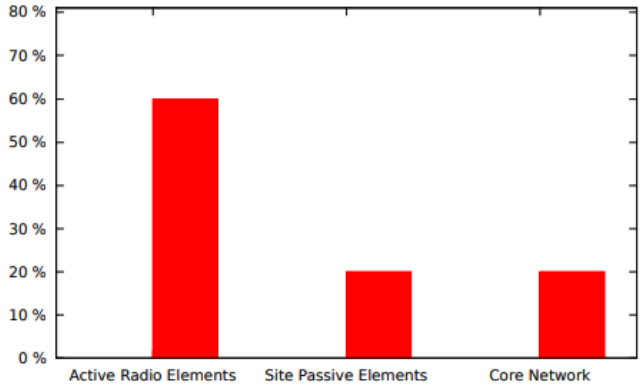

Рисунок 3.4 Інвестиції в розгортання мобільних мереж

Фактично, країни, що розвиваються, мають історію ненадійного постачання електроенергії та часті вимикання. Африканські країни, менш охоплені державною мережею, більшість телекомунікаційних операторів залежать від дизельних генераторів та сонячної енергії. Дизель-генератори викликають головні недоліки, такі як збільшення технічного обслуговування та транспортування до станції, особливо в сільській місцевості. Істотно, це безпосередньо впливає на операційні витрати. Спільне використання Cloud на мобільній мережі між операторами або віртуальними операторами, безсумнівно, призводить до зменшення інвестиційних та операційних витрат. На рисунку 3.5 показано економію інвестицій у мережі, створеній у хмарі. Під час обміну мережею на основі хмар зменшується приблизно 50% загальних інвестицій. Фактично, оператори або MVNOs будуть орендувати ресурс на основі вимог і платити за використання ресурсів. Це правда, що зниження інвестицій постачальника хмарних послуг може змінюватись залежно від побудови інфраструктури. Проте, постачальник послуг отримає вигоду від інших операторів, продаючи послуги, ці переваги можуть подолати всі інвестиції в майбутньому. Для клієнтів (інших операторів / MVNOs) 50% загальних інвестицій на активні

елементи (сукупні інвестиції, показані на рис. 5), очевидно, зменшиться. Наприклад, клієнтові не потрібно інвестувати на вузлові перемикачі доступу та деякі елементи RAN, такі як BBUs та RRU і т. Д. Клієнт може обмінюватися цими елементами з обласним постачальником на основі вимог динамічно. Аналогічно, з елементами пасивної та базової мережі, в пасивному елементі ми можемо отримати більше користі (тобто 60%) і вимагати менше інвестицій. Наприклад, обмін у хмарі не потребує інвестування на будівництво фізичного майданчика, орендної плати та притулків тощо.

3.10 Впровадження та обговорення

На рисунку 3.5 показано схему тестування на основі хмарної мобільної мережі. Для того, щоб забезпечити реалістичну тест-версію, ми використовували програмні елементи мобільної мережі, що надаються Fraunhofer FOKUS OpenEPC. OpenEPC - еталонна реалізація Evolved Packet Core (EPC) 3GPP (релізи 11 і 12), розроблена Центром компетентності Fraunhofer FOKUS для мереж інфраструктури наступного покоління (NGNI).

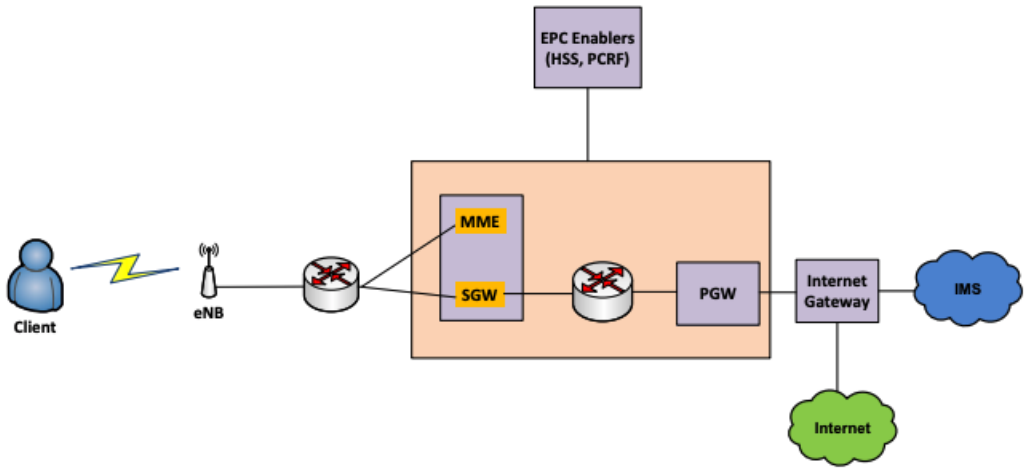

Рисунок 3.5 Реалізована LTE/EPC архітектура на базі openEPC

У цій емуляції кожен модуль LTE / EPC, такий як MME, SGW, PGW, eNB, HSS та UE, працює на кожній віртуальній машині. Всі ці віртуальні машини працюють на одній машині на базі Linux. KVM (віртуальна машина на базі ядра) використовується як гіпервізор між віртуальними машинами в ядрі. Всі віртуальні машини мають таку ж конфігурацію, як оперативна пам'ять 500 Мб і жорсткий диск

на 20 Гб, кожен з яких працює з операційною системою Ubuntu 12.04. Щоб перевірити хмарну мережу, було проведено тест, щоб з'ясувати затримку мережі OpenEPC порівняно з прямим мережею Інтернету. Було створено звичайний IP-трафік, такий як пінг-трафік між мережею OpenEPC та сервером Google DNS (8.8.8.8), а також HTTP-трафіком до Інтернету. Точно так само було створено такий самий трафік із прямим доступом до мережі Інтернет (тобто без OpenEPC).

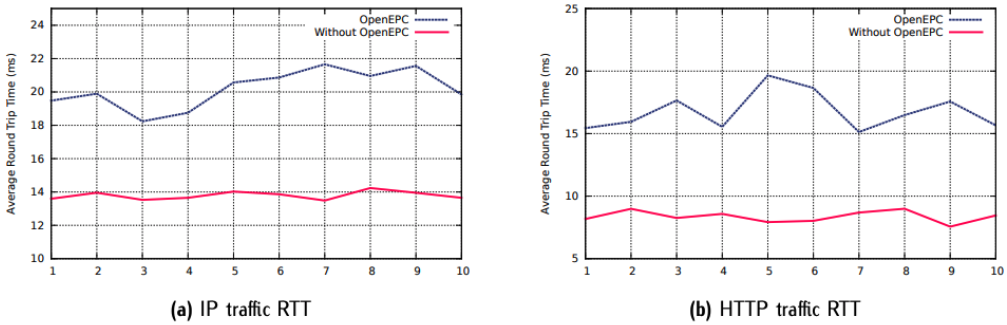

Рисунок 3.6 Середній RTT (час повернення) між LTE/EPC мережею використовуючою OpenEPC і іґмережею з прямим доступом в інтернет.

Результати випробувань наведено на рисунку 3.6. Це чітко показує, що мережа OpenEPC матиме більше часу, ніж пряма мережа доступу до мережі Інтернет. Це пов'язано з тим, що мережа LTE / EPC матиме додаткові накладні витрати на IP з тунелюванням GTP у вузлах користувальницької площини (тобто тунелювання між eNB і SGW, а також між SGW і PGW). Коли UE генерує трафік, трафік тунелюється в eNB і передається на SGW. Аналогічним чином, SGW буде тунель такого ж трафіку і пересланий PGW. Через це інкапсуляція та декапсуляція тунелів GTP на площині користувача, час круглої поїздки в мережі LTE / EPC збільшується в порівнянні з прямим доступом до мережі Інтернет. Цікаво, що латентність трафіку IP (див. Рис. 8a) є більшою, ніж трафік HTTP (див. Рисунок 3.6). Проте ця латентність повністю залежить від вибору шляхів і розташування серверів. Насправді, обидва сервери розташовані в одному місці (наприклад, США), і вони вибирають різні шляхи. Проте затримка мережі LTE / EPC становить близько 22мс для трафіку IP та 20мс для HTTP-трафіку, і ці затримки є прийнятними для аудіо та відеодзвінків. Заснована на вимогах IMT-2000 та якості

обслуговування, така латентність є прийнятною (максимальна затримка у зворотному напрямку, яку може допустити сприйняття людини, становить 400 мс).

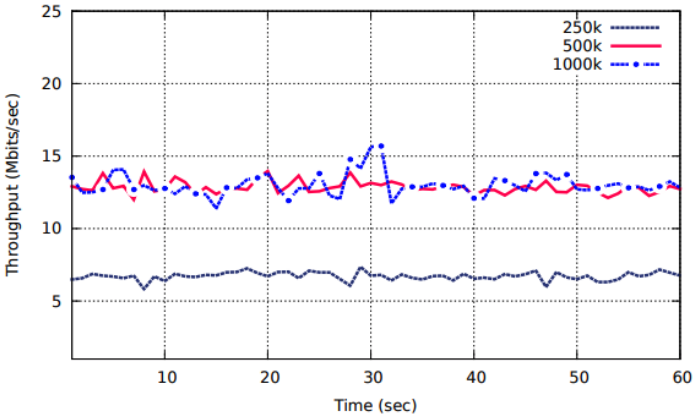

Рисунок 3.7 OpenEPC UE пропускна здатність в вікнах TPC різного розміру

На рисунках 3.7 та 3.8 показано оцінку пропускної здатності в мережі OpenEPC. На рисунку 3.8 показана пропускна спроможність в трафіку UDP (тобто вона може розглядатися як трафік VOIP [14], який також є UDP), що стосується нормальної мережі доступу до мережі Інтернет. З іншого боку, на рисунку 3.7 показана пропускна здатність у різних розмірах вікна TCP, таких як 250k, 500k та 100k. Для всієї цієї оцінки було використано інструмент генерації трафіку, такий як Iperf, який здатний генерувати трафіки TCP і UDP з декількома паралельними з'єднаннями. Клієнт Iperf, що працює у OpenEPC UE, генерує трафік TCP / UDP на сервер, який підключений до PGW. Під час збільшення розміру вікна TCP продуктивність також збільшується. Однак, після насичення держави, навіть якщо інкрементальний в розмір вікна не збільшиться. Аналогічним чином, у мережі OpenEPC розмір вікна вище 1000к, пропускна спроможність приблизно однакова. Однак ця пропускна здатність є прийнятною для звичайного підключення до Інтернету, що зазвичай вимагає мінімум 1 Мбіт / сек. Фактично, ми вважаємо, той самий OpenEPC працює у великих обсягах серверів у хмарі, показники набагато кращі, а затримка зменшиться до 10 мс.

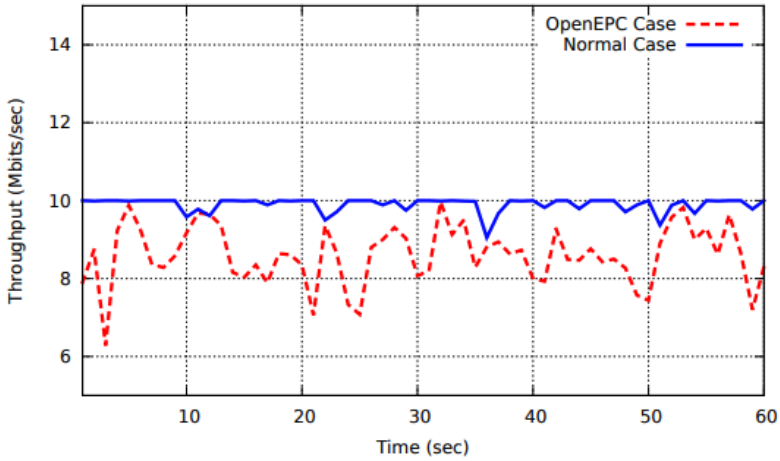

Рисунок 3.8 Пропускна здатність UE для UDP трафіку, за умови що ширина полоси пропускання обмежена 10МБіт/с

Аналогічним чином, трафік UDP генерується клієнтом Iperf у OpenEPC, де пропускна здатність обмежена до 10 Мбіт / с. Пропускна здатність становить близько 8,5 середнього, що менше, ніж обмежена смуга пропускання (на рис 10). Для кращого порівняння ми оцінили той же тест в звичайній мережі Інтернет, де пропускна здатність становить близько 9,5 Мбіт / с, що також менше обмеженої пропускної здатності. Однак ця пропускна здатність є прийнятною для високоякісних відео та VOIP-послуг. Ця реалізація може бути дзеркалом хмарної мобільної мережі.

Основні переваги цієї хмарної мобільної мережі це те, що3 мережа може бути легко масштабним вводом / виводом на основі попиту та вводить гнучкість і еластичність мережі. Наприклад, зображення VEM OpenEPC кожного вузла LTE / EPC можна легко перенести з одного центру обробки даних в інший центр обробки даних та масштабувати під час запиту. Процес міграції віртуальних машин не входить в рамки цього документа. Результати тестів показують, затримка в 30мс є прийнятною в IP і випадку HTTP-трафіку, а також голосового і відео трафіку (який є UDP-трафік), яка також є більш прийнятною, ніж час затрички (тобто з кінця в кінець 150мс, в тому числі мережі доступу) при використанні цієї хмарної мережі.

3.11 Виклики

Спільне використання мобільних мереж у хмарі створює певні технічні проблеми для операторів.

- Головними проблемами в обласній мережі RAN є зменшення використання пропускної здатності між антеною та BBU для того, щоб виконувати сигналізацію базової смуги. Наприклад, для швидкості передавання даних потрібно 10 Гбіт / с, щоб запустити вісім антенних LTE з частотою 20 МГц. З іншого боку, централізація BBUs та розподіл радіоприймачів потребує повноцінної волоконної мережі, яка є економічно ефективним рішенням. Це може бути вирішено за допомогою мікрохвильових ліній для малих конфігурацій (низька кількість антен, 5 або 10 МГц спектру та невеликі відстані) завдяки пропускній здатності міліметрової хвилі
- Коли компонент змінює місцеположення з одного хмара на іншу хмару, передача активного вмісту на нове місце стане проблемою для постачальника хмарних мереж. Крім того, збільшення масштабу вниз / вниз віртуальних машин та їх налаштування з іншими VM може спричинити такі проблеми, як затримка. Наприклад, коли MME масштабування, він повинен чекати, поки всі інтерфейси будуть встановлені з іншими VM, таких як S11, S1- MME, S6a. З іншого боку, балансування навантаження між віртуальними машинами є ще однією технічною проблемою. Наприклад, коли віртуальні машини масштабують вгору / вниз, активне балансування трафіку з іншим VM та тунелями, інкапсуляцією / декапсуляцією тощо. Однак, можливо, можна буде повторно використовувати існуючий балансувальник навантаження в центрах обробки даних з розширенням у мобільну мережу.
- У мережі обміну, білінгові моделі та реалізація цих моделей є складними проблемами. Наприклад, кожен орендатор (MNOs та MVNOs) матимуть свої власні стандарти соціального обслуговування з їхніми користувачами, а також з обласним провайдером. У цьому випадку всі ці SLA, які повинні бути впроваджені в мережу та моніторинг у реальному часі, потрібні. З іншого боку, кожен орендатор вимагає інтерфейсів для моніторингу своїх ресурсів та впровадження будь-якої нової служби в мережу. Однак це можна вирішити, надаючи інтерфейси з платформою керування, але постачальник хмар повинен дбати про перегляд та керування лише користувачем орендаря.

Висновки

Як видно з інформації, наведеної вище, мобільна мережа розгорнута на базі хмарної інфраструктури з використанням OpenEPC (відкритих розвинутих пакетних ядер) є абсолютно виправданою і показники такої мережі, а саме час затримки та швидкість обробки інформації за допомогою ресурсів віртуальних машин, є вищими для пристроїв зв’язку котистувачів. Але є ще проблема розподілу навантаження на такі мережі. Саме алгоритми розподілу навантаження будуть розглянуті у наступному розділі.

4 АЛГОРИТМИ РОЗПОДІЛУ НАВАНТАЖЕННЯ

Існує 4 найбільш часто використовуваних алгоритмів розподілу навантаження, особливості яких будуть наведені нижче.

4.1 Алгоритм Round Robin

Round Robin, без сумніву, є найпоширенішим алгоритмом. Його легко реалізувати та легко зрозуміти. Ось як це працює [15]. Скажімо, у вас є 2 сервери, які чекають запитів за вашим навантажувачем. Після того, як перший запит надійде, балансувальник завантажуватиме цей запит на 1-й сервер. Коли надходить другий запит (імовірно, від іншого клієнта), цей запит буде переадресовано на другий сервер.

Оскільки другий сервер є останнім у цьому кластері, наступний запит (наприклад, 3-й) буде перенесено на перший сервер, четвертий запит на другий сервер і так далі, циклічно.

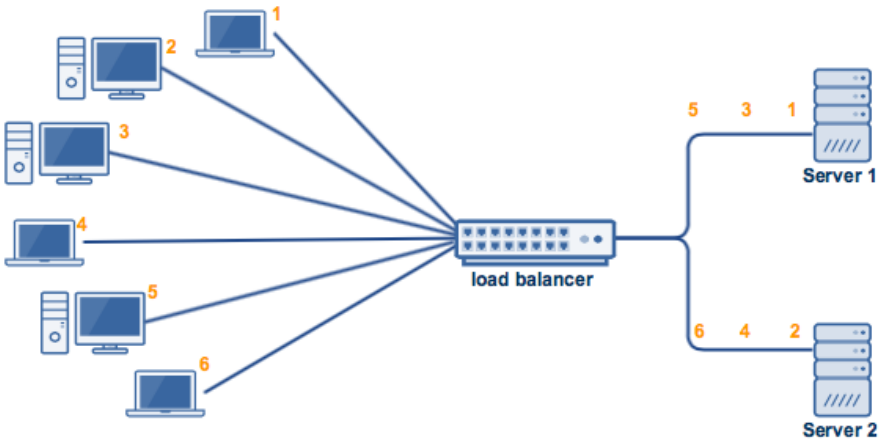

Рисунок 4.1 схематичне зображення алгоритму Round Robin

Як видно, метод дуже простий. Однак, це не буде добре в певних сценаріях.

Наприклад, що якби сервер 1 мав більше процесорів, оперативної пам'яті та інших специфікацій порівняно з сервером 2? Сервер 1 повинен мати можливість обробляти більше навантаження, ніж сервер 2, чи не так?

На жаль, балансирувальник навантаження, який працює за алгоритмом Round Robin, не зможе відповідно обробляти два сервери. Незважаючи на непропорційну потужність двох серверів, балансувальник навантаження все одно буде розподіляти

запити однаково. В результаті сервер 2 може перевантажуватись швидше і, можливо, навіть знизиться. Ви б не хотіли, щоб це сталося.

Алгоритм Round Robin найкраще підходить для кластерів, що складаються з серверів з ідентичними специфікаціями. Для інших ситуацій, можливо, вам доведеться подивитися на інші алгоритми, як наведені нижче.

4.2 Алгоритм Weighted Round Robin

Для другого сценарію, згаданого вище, наприклад, сервер 1, що має вищі характеристики, ніж сервер 2, ви можете віддати перевагу алгоритм, який призначає додаткові запити серверу, з більшою спроможністю обробляти більшу навантаження. Одним з таких алгоритмів є ваговий круглий Робін.

Weighted Round Robin аналогічно Round Robin в тому сенсі, що спосіб, яким запити присвоюються вузлам, як і раніше, є циклічними, хоча і з твіст. У вузлі з вищими специфікаціями буде розподілено більше число запитів.

Але як балансувальник навантаження знає, який вузол має більшу продуктивність? Простий. Ви говорите це заздалегідь. В принципі, коли ви встановлюєте балансування навантаження, ви призначаєте "ваги" для кожного вузла. Звичайно, вузол з вищими характеристиками повинен мати більшу вагу.

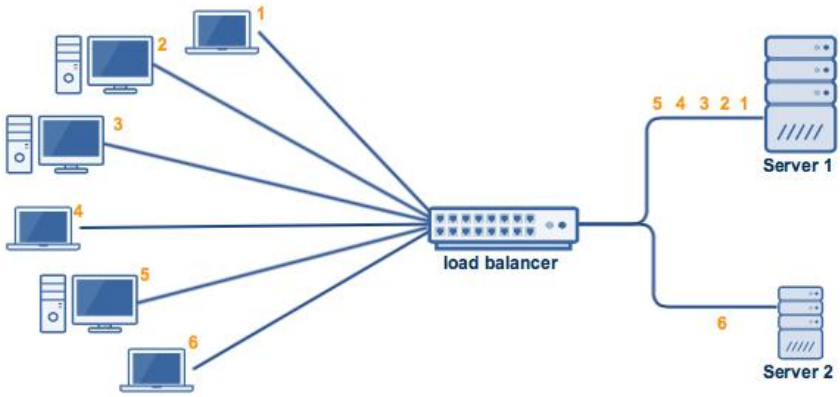

Рисунок 4.2 схематичне зображення алгоритму Weighted Round Robin

Ви зазвичай вказуєте ваги пропорційно фактичним потужностям. Так, наприклад, якщо пропускна здатність сервера 1 становить 5x більше, ніж сервер 2, то ви можете призначити йому вагу 5, а сервер 2 - вагою 1.

Тому, коли клієнти починають входити, перші 5 будуть призначатися для вузла 1 та 6-го для вузла 2. Якщо з'явиться більше клієнтів, буде дотримуватися така ж послідовність. Тобто, 7, 8, 9, 10 і 11 всі йдуть на Server1, а 12-й на сервер 2 і так далі.

Потужність не є єдиною причиною для вибору алгоритму Weighted Round Robin (WRR). Іноді ви хочете використовувати його, якщо скажете, що ви хочете, щоб один сервер отримував значно меншу кількість з'єднань, ніж однаково здатний сервер, оскільки на першому сервері працюють критично важливі програми, і ви не хочете, щоб це було легко перевантажений.

4.3 Алгоритм Least Connections

Можуть бути випадки, коли навіть якщо два сервери в кластері мають точно такі ж характеристики (див. Перший приклад / малюнок), один сервер все ще може перевантажуватись значно швидше, ніж інший. Однією з можливих причин буде те, що клієнти, що підключаються до сервера 2, залишаються на зв'язку набагато довше, ніж ті, що підключаються до сервера 1.

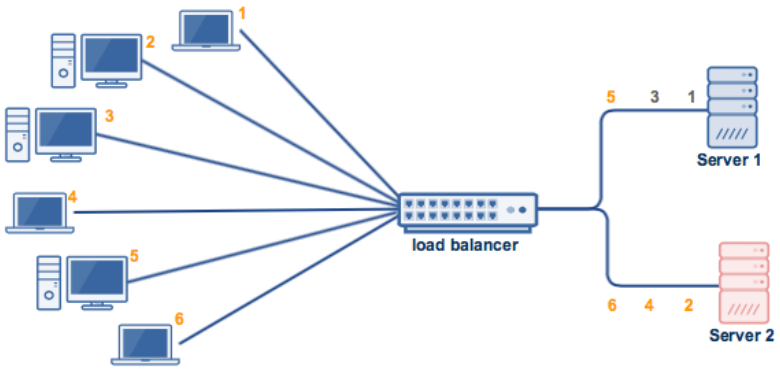

Рисунок 4.3 схематичне зображення алгоритму Least Connections

Це може спричинити накопичення сукупних поточних з'єднань на сервері 2, а сервери 1 (за умови, що клієнти з'єднуються та від'єднуються за коротший час) практично залишаться незмінними. В результаті ресурси сервера 2 можуть закінчитися швидше. Це зображено нижче, де клієнти 1 і 3 вже відключені, а 2, 4, 5 та 6 все ще підключені.

У подібних ситуаціях алгоритм найменших зв'язків буде кращим. Цей алгоритм враховує кількість поточних з'єднань кожного сервера. Коли клієнт

намагається з'єднатись, балансування навантаження намагатиметься визначити, який сервер має найменшу кількість з'єднань, а потім призначити нове підключення до цього сервера.

Отже, якщо сказати (продовжуючи наш останній приклад), клієнт 6 намагається з'єднатись після того, як 1 і 3 вже відключені, але 2 і 4 все ще підключені, балансувальник навантаження призначить клієнтові 6 сервер 1, а не сервер 2.

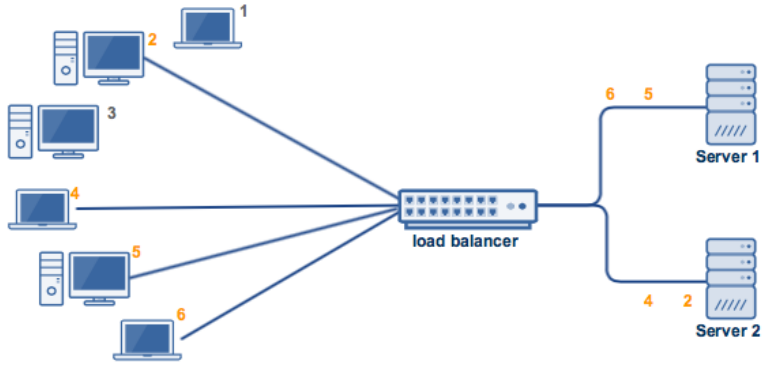

Рисунок 4.4 схематичне зображення алгоритму Least Connections

4.4 Алгоритм Weighted Least Connections

Алгоритм зважених найменших зв'язків робить для найменших зв'язків, який ваговий круглий Робін робить до Round Robin. Тобто він вводить "ваговий" компонент на основі відповідних можливостей кожного сервера. Як і в Weighted Round Robin, вам доведеться заздалегідь вказати "вагу" кожного сервера.

Балансувальник навантаження, який реалізує алгоритм зважених найменших зв'язків, зараз враховує дві речі: вага / ємність кожного сервера і поточна кількість клієнтів, які в даний час підключені до кожного сервера.

5 АЛГОРИТМ РОЗПОДІЛУ НАВАНТАЖЕННЯ ДЛЯ МЕРЕЖІ РОЗГОРНУТОЇ НА БАЗІ ХМАРНОЇ ІНФРАСТРУКТУРИ.

5.1 Обґрунтування вимог до алгоритму розподілу навантаження для мереж з використанням хмарних технологій.

Баланс навантаження неприпустимих залежних завдань на віртуальних машинах (VM) є важливою характеристикою планування завдання в хмарах. Кожного разу, коли певні віртуальні машини перевантажені, навантаження повинно бути спільним з підвантаженими віртуальними машинами для досягнення оптимального використання ресурсів для найменшого терміну виконання завдань. Більш того, величензнеий внесок ідентифікації ресурсів та міграції завдань мають бути розглянуті в алгоритмах рохподілу навантаження. Загалом, віртуальна машина використовує два різні механізми виконання завдання, такі як простір і час спільного використання. У механізмі розділення простору завдання виконуватимуться один за іншим. Звідси випливає, що в його процесорі виконується лише одне завдання на процесор / ядро. Інші завдання, призначені для цієї віртуальної машини, повинні бути в черзі очікування. В результаті міграція задачі в балансуванні навантаження буде простішою в цьому механізмі спільного простору шляхом визначення завдання в черзі очікування перевантаженої віртуальної машини та присвоєння його недогруженій віртуальній машині. Тим не менше, у механізмі спільного використання часу, завдання виконуються одночасно в стиснутому вигляді, що нагадує виконання завдань у паралельному режимі. Тут міграція задач у балансуванні навантаження буде дуже складною через час виконання всіх завдань. Таким чином, майже в 90% випадків певний відсоток інструкцій завдань буде в завершеному стані на механізм, який виконується за часом. Рішення про ідентифікацію завдання, яке потрібно перенести з вищої завантаженої віртуальної машини до нижчої завантаженої віртуальної машини, є дуже дорогим через втрату раніше виконаної частини у вищій завантаженій віртуальній машині та через вплив виконання завдання на час виконання інших завдань у попередньо-завантаженій віртуальній машинині. Таким чином, алгоритм планування та завантаження балансу повинен досягти оптимальної / мінімальної

міграції завдань з однаковим розподілом навантаження між ресурсами на основі ресурсної можливості без будь-якого простою будь-якого ресурсу в будь-який момент часу. Цей метод / процес допомагає досягти оптимального / мінімального часу виконання в обласному середовищі. Алгоритм повинен також враховувати непередбачуваний характер прибуття запитів на роботу та їх розподіл на відповідні віртуальні машини, розглядаючи роботу з багатозадачними завданнями та взаємозалежність. Алгоритм повинен бути придатним як для однорідних, так і для різнорідних середовищ з різними робочими характеристиками. З огляду на цю мету та на основі літературних напрацювань можна запропонувати наступний алгоритм.

Алгоритм балансування навантаження має на меті досягти збалансованого навантаження на віртуальних машинах, щоб максимізувати пропускну спроможність та збалансувати пріоритети завдань на віртуальних машинах. Отже, кількість часу очікування завдань у черзі має бути мінімальною. Використовуючи цей алгоритм, можна удосконалити середнє виконання часу та зменшення часу очікування завдань у черзі.

Головною метою є оптимізація продуктивності віртуальних машин за допомогою комбінації статичного та динамічного розподілу навантаження шляхом визначення тривалості робочих завдань, ресурсних можливостей, взаємозалежності декількох завдань, ефективного прогнозування недодержаних віртуальних машин та уникнення перевантаження на будь-якому з віртуальних машин . Цей додатковий параметр розгляду "тривалості роботи" може допомогти розподілити роботу в потрібних віртуальних машинах у будь-який момент і зможе доставити відповідь у мінімальний час виконання. Ефективне планування за цим алгоритмом також мінімізує перевантаження на віртуальній машині, а згодом також мінімізує міграцію завдань.

Проаналізовано ефективність вдосконаленого алгоритму WRR (Improved Weighted Round Robin, далі IWRR) та проведено оцінку алгоритму щодо існуючого Round Robin алгоритму та Weighted Round Robin. Далі вважається, що завдання містить кілька підзавдань, і підзавдання мають взаємозалежність між ними. Завдання може використовувати кілька віртуальних машин для виконання різних

завдань для завершення всієї інструкції з обробки. Крім того, завдання може використовувати декілька елементів обробки одної VM на основі конфігурації та доступності.

5.2 Розподіл навантаження та планування.

На рисунку 5.1 показано схему балансу навантаження та планування, в якій планувальник має таку логіку, щоб знайти найбільш підходящу віртуальну машину та призначити завдання для віртуальних машин на основі запропонованого алгоритму. Планувальник розміщує запити на роботу у найбільш підходящих віртуальних машинах на основі найменш завантаженої віртуальної машини на на момент приходу запиту на роботу.

Load Balancer вирішує міграцію завдання від сильно завантаженої віртуальної машини до незайманої віртуальної машини або найменш завантаженої віртуальної машини під час виконання, коли вона знаходить неактивну віртуальну машину або найменш завантажену віртуальну машину, використовуючи інформацію про поточний стан ресурсів. Ресурсний монітор зв'язується з усіма ресурсами віртуальних машин і збирає можливості VM, поточну навантаження на кожну віртуальну машину та кількість завдань у черзі виконання / очікування в кожній віртуальній машині. Вимога до завдання забезпечується користувачем, який включає довжину виконуваних завдань та передає вимоги до планувальника для його оперативних рішень.

Планування та налагодження балансу навантаження

Запит на роботу дає користувач через інтерфейс і передається менеджеру завдань для аналізу залежних та незалежних завдань. Цей модуль отримує запит та перевіряє, чи завдання є повним незалежним завданням або містить декілька завдань. Якщо запит містить декілька завдань, то він перевіряє незалежну чергу завдання. Залежні завдання будуть сповіщені планувальнику, так що батьківські завдання заплановані після виконання дочірніх завдань. Черга завдань залежності буде містити завдання, яке залежить від інших завдань, наявних у віртуальних машинах.

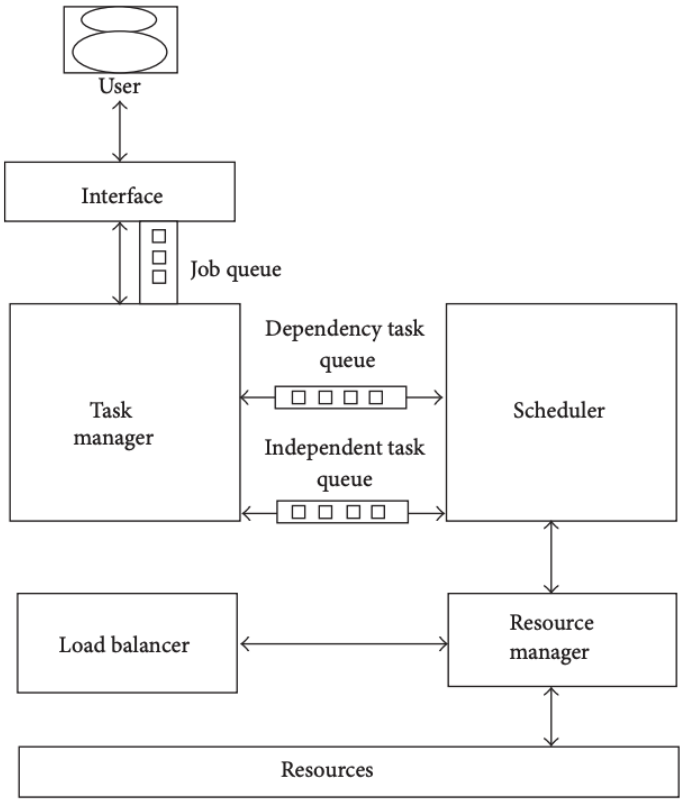

Рисунок 5.1 Розподіл навантаження тапланування.

Коли всі дочірні завдання, що знаходяться в цій черзі, завершують виконання, батьківське завдання буде прийнято для виконання шляхом присвоєння його VM, тоді як незалежна черга завдання містить незалежні завдання. Незалежна задача черги та залежні завдання вводяться в планувальник. Планувальник вибирає відповідну віртуальну машину на основі алгоритму IWRR. Цей планувальник збирає інформацію про ресурси з менеджера ресурсів. Він вираховує обчислювальні потужності кожної з віртуальних машин, а потім застосовує пропонований алгоритм для пошуку відповідної віртуальної машини для даного завдання. Крім того, кожна віртуальна машина підтримує інформацію про JobExecutionList, JobPauseList та JobWaitingList. JobExecutionList містить поточний список виконуваних завдань, а JobPausedList містить тимчасово призупинені завдання в VM. Аналогічно, черга JobWaitingList містить завдання, що очікують на конкретній віртуальній машині, але це буде виконано після отримання JobExecutionList, JobPauseList та JobWaitingList з кожного з віртуальних машин; розрахунок найменш використаної віртуальної машини виконується для кожного запиту прибуття. Потім ця найменш використана інформація про віртуальну

машину буде повернута до планувальника. Менеджер ресурсів зв'язується з усіма віртуальними машинами, щоб збирати інформацію про можливості кожної з них, отримуючи кількість оброблюваних елементів та обсяги обробки для кожного з елементівна віртуальних машинах. Цей менеджер ресурсів додатково обчислює ваговий коефіцієнт для кожної віртуальної машини на основі розподіленої їй технологічної потужності. Це також визначає пам'ять, налаштовану у кожномій з віртуальних машин. Балансувач навантаження розраховує співвідношення між кількістю завдань та кількістю віртуальних машин. Якщо співвідношення менше 1, він повідомляє планувальник для ідентифікації VM для роботи; інакше він обчислить навантаження на кожен з віртуальних машин, використовуючи список виконаних робіт віртуальних машин. Якщо використання менше 20%, то найменш використана віртуальна машина буде виділена; також планувальник буде повідомлений для ідентифікації найбільш підходящої віртуальної машини для роботи. Після того як відповідна VM буде ідентифікована, завдання буде призначено для цієї віртуальної машини. Налаштовані центри обробки даних містять хости, а їх віртуальна машина з відповідними елементами обробки створює набір ресурсів, доступних для обчислень. Ресурси досліджуються на холостиу роботу та на важке навантаження, так що запити на завдання ефективно розподілялися на відповідний ресурс.

Розрахунок фактору розбалансування навантаження

Сума навантажень всіх віртуальних машин визначається як (1)

$$L = \sum_{i=1}^{k} l_i, \qquad (1)$$

де *i* показує кількість віртуальних машин в дата центрі.

Навантаження на одну машину візначається як (2)

$$\mathrm{LPC} = \frac{L}{\sum_{i=1}^{m} c_i} \qquad (2)$$

Поріг $T_i = \mathrm{LPC} * c_i,$

де c_i представляє ємність машини.

Фактор розбалансування однієї віртуальної машини:

$$\text{Якщ} \quad \text{VM} \begin{cases} < \left| T_i - \sum_{v=1}^{k} L_v \right|, & \text{Недовантажен} \\ > \left| T_i - \sum_{v=1}^{k} L_v \right|, & \text{Перевантаже} \\ = \left| T_i - \sum_{i=1}^{k} l_i \right|, & \text{Збалансова} \end{cases} \qquad (3)$$

Переміщення завдання з перевантаженої віртуальної машин и в недонавантажену віртуальну машину може бути допустимою доти, поки навантаження на перевантажену віртуальну машину не опуститься нижче порогу, а різниця становить μ_i.

Віртуальна машина ідентифікується як недогружена, де сума навантажень всіх віртуальних машин знаходиться нижче порогового значення цієї віртуальної машини. Недонавантажена VM приймає навантаження з перевантаженої віртуальної машини, доки навантаження на VM не перевищить порогове значення, а різниця λ_j, як показано нижче.

Передача навантаження з перевантаженої VM здійснюється до тих пір, поки його навантаження не стане нижчим порогового значення. Недонавантажена віртуальна машина може приймати навантаження лише до свого порогу, щоб уникнути перевантаження. Це означає, що об'єм навантаження, який може бути перенесений з недогруженої VM, має бути в діапазоні μ і λ.

5.3 Алгоритм IWRR

Запропонований алгоритм IWRR є найбільш оптимальним, і він розподіляє роботу на найбільш підходящі віртуальні машини на основі інформації з віртуальної машини, такої як його пропускна спроможність, навантаження на віртуальні машини та довжина поставлених завдань з її пріоритетом. Статичне планування цього алгоритму використовує обчислювальні потужності віртуальних

машин, кількість вхідних завдань та довжину кожного завдання, щоб вирішити розподіл на відповідній віртуальній машині.

Динамічне планування (під час виконання) даного алгоритму додатково використовує навантаження на кожну віртуальну машину поряд з вказаною вище інформацією, щоб вирішити розподіл завдання на відповідну віртуальну машину. Існує ймовірність під час виконання, що в деяких випадках завдання може зайняти більше часу виконання, ніж початковий розрахунок, завдяки виконанню більшої кількості ітерацій (наприклад, циклу) за тими самими вказівками, що базуються на складному режимі роботи з даними. У таких ситуаціях балансування навантаження рятує контролер планування та переставляє завдання відповідно до вільного місця, доступного в інших невикористаних / недонавантажених VM, перемістивши завдання з черги перевантаженої машини на вільну VM. Балансувальник навантаження визначає невикористані / недостатньо використані віртуальні машини за допомогою перевірки ресурсу, коли завдання завершується на будь-якій з віртуальних машин. Якщо всі віртуальні машини зайняті, то балансувальник навантаження не виконуватиме міграцію завдань серед віртуальних машин. Якщо він знаходить будь-яку невикористану / недостатньо використану віртуальну машину, то перенесе завдання з перевантаженої віртуальної машини на невикористану / недостатньо навантажену віртуальну машину. Балансувальник навантаження аналізує навантаження ресурсу (VM) лише після завершення будь-якого завдання на будь-якому з віртуальних машин. Він ніколи не перевіряє завантаження ресурсу (VM) самостійно в будь-який час, щоб обійти накладні витрати на віртуальних машинах. Це допоможе зменшити кількість міграцій завдань між віртуальними машинами та кількість запитів на стан ресурсів в віртуалдьних машинах.

Реалізація алгоритму

Реалізація системи складається з п'яти основних модулів:

- Static scheduler, Статичний планувальник (початкові місця розміщення).

- Dynamic scheduler, Динамічний планувальник (розміщення часу запуску).
- Load balancer, Балансувальник навантаження (рішення про міграцію завдання під час виконання)
- Task interdependent scheduler, Завдання взаємозалежного планувальника.
- Resource monitor, Монітора ресурсів.

Статичний планувальник має функцію знаходження найбільш підходящої віртуальний машини та призначення цих завдань віртуальним машинам на основі алгоритмів (RR, WRR, IWRR), що застосовуються в планувальнику. Динамічний планувальник має функцію розміщення завдань в реальном учасі до найбільш підходящих віртуальних машин на основі даних про найменш використану віртуальну машину в даний момент часу. Балансувальник навантаження / планувальника контролюють міграцію завдання із сильно завантаженої віртуальної машини на незайману віртуальну машину або найменш завантажену віртуальну машину кожного разу, коли він знаходить відповидну віртуальну машину, використовуючи інформацію від монітора ресурсів. Монітор ресурсів зв'язується з усіма засобами пошуку ресурсів віртуальних машин і збирає дані про можливості віртуальної машини, поточну завантаженість кожної віртуальної машини та кількість завдань у черзі виконання / очікування в кожній віртуальній машині, щоб обрати відповідні віртуальні машини для завдань. Оцінювач вимог до завдання визначає довжину завдань, що підлягають виконанню, і передає очікувані результати балансувальнику навантаження для своїх оперативних рішень.

На малюнку 5.2 ілюструється архітектура системи, в якій система складається з брокера центрів обробки даних та декількох центрів обробки даних.

Центр обробки даних може розміщувати будь-яку кількість хостів, і цей хост може розміщувати будь-яку кількість віртуальних машин, вираховану на основі його потужності. Брокер центрів обробки даних має важливі компоненти, щоб ефективно планувати та розподіляти нвантаження, тоді як алгоритми цих компонентів варіюються відповідно до плгоритмів RR, WRR, IWRR.

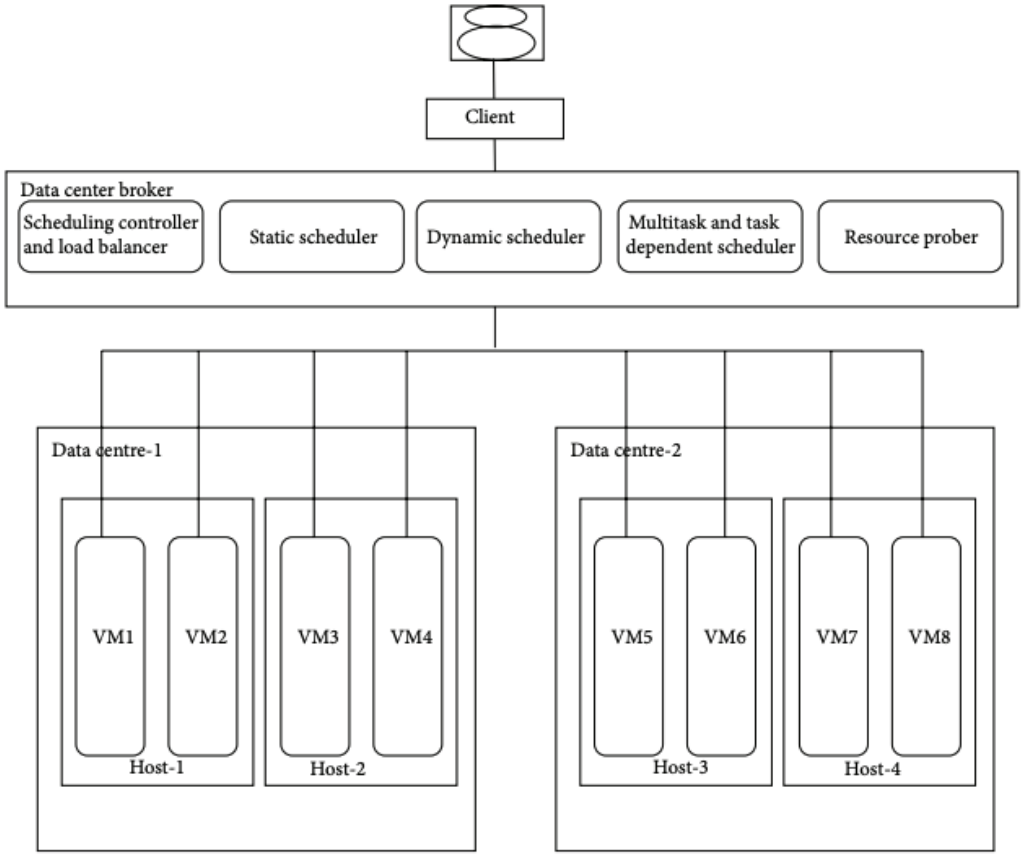

Рисунок 5.2 Архітектура системи

Динамічний планувальник в балансувальнику завантаження IWRR

Динамічне планування в балансувальнику завантаження IWRR досягається за допомогою ініціалізації, відображення (планування), балансу навантаження та виконання, як це пояснюється в алгоритмі планувальника нижче. Ініціалізація здійснюється шляхом збору очікуваного часу виконання завдання з кожної створеної віртуальної машини і розташування його у порядку зростання часу очікування, а потім організація часу виконання поставлених завдань у черзі на основі пріоритету. Планування передбачає вибір завдання, що знаходиться у верхній частині черги, та розрахунку його часу завершення в кожній віртуальній машині. Тоді задача присвоюється найбільш відповідній віртуальній машині на основі часу очікування та часу завершення виконання. Балансування навантаження здійснюється шляхом додавання відповідних завдань та часу виконання до час очікування віртуальних машин. Потім відбувається реорганізація списку задіяних віртуальних машин на основі даних завантаження, за яким слід відправити

завдання до черги обробки віртуальних машин і додати завдання до призначеного списку. Коли всі завдання присвоєні віртуальним машинам, завдання починають виконуватися.

Опишемо прототип алгоритму роботи планувальника для даної системи. Цифрами позначено коментарі до дій алгоритму, літерами визначені кроки логіки самого алгоритму.

(1) Визначте очікуваний час виконання кожного з віртуальних машин, зібравши тривалість очікування виконання від виконання, очікування та призупинення списку.

(a) Set pendingJobsTotLength = JobsRemainingLengthInExecList + JobsRemainingLengthInWaitList + JobsRemainingLengthInPauseList

(b) Cvm - це можлівості обробки VM.

(c) Set pendingETime = pendingJobsTotLength/CVm

(2) Розташуйте віртуальні машини в список від найменшого часу очікування до найбільшого часу очікування і згрупуйте їх у випадку однакових параметрів у декількох машин. Такий елемент Map (це список обьєктів "ключ-значення") має містити час очікування в якості ключа та відповідну віртуальну машину в якості значення.

(a) Відсортуйте VMMap за часом очікування кожної VM

(3) Реорганізуйте завдання зо надходять за довжиною та пріоритетом.

(a) Відсортуйте JobSubmittedList за довжиною і пріоритетом.

(4) Ініціалізуйте змінні vmIndex, jobIndex variable & totalJobs

(a) Set vmIndex = 0

(b) Set totalJobs = length of JobSubmittedList

(c) Set totalVMsCount = size of VMMap

(d) Set jobIndex = 0

(e) Set jobToVMratio = totalJobs/totalVMsCount

(5) Назначте завдання що надходять вірутальним машинам на основі найменшого часу очікування у віртуальній машині та її можливостей обротки.

(a) While (true)

Set job = JobSubmittedList.get(jobIndex)

Set jobLength = lengthOf(job)

Set newCompletiontimeMap = EmptyMap

For startNumber from 0 by 1 to totalVMsCount do {

Set vm = VMMap.getValue(startNumber)

Set probableNewCompTime = jobLength/CVm + VMMap.getKey(startNumber)

```
                newCompletiontimeMap.add(probableNewCompTime, vm)
        }
        SortByCompletionTime(newCompletiontimeMap)
        Set selectedVM = newCompletiontimeMap.getValue(0)
        selectedVM.assign(job)
        For startNumber from 0 by 1 to totalVMsCount do {
                Set vm = VMMap.getValue(startNumber)
                If (vm equals selectedVM)
                Set currentLength = VMMap.getKey(startNumber)
                Set newCurrentLength = currentLength + newCompletiontimeMap.getKey(0)
                VMMap.removeItem(startNumber)
                VMMap.add(newCurrentLength, vm)
                EndIf
        }
        sortByCompletionTime(VMMap)
        Increase the jobIndex by 1
        If (jobIndex equals totalJobs)
        Break
    (b) End While
(6) Видаліть усі призначені завдання з JobSubmittedList
```

Балансувальник навантаження в IWRR.

Балансувальник навантаження ініціалізується, збираючи очікуваний час виконання з кожної створеної віртуальної машини, потім організовуючи його в порядку зростання, щоб визначити кількість завдань у кожній віртуальній машині та організувати їх за пріорітетом у порядку зростання. Планування здійснюється шляхом отримання завдання від VM із вищим часом очікування, потім відбувається ідентифікування найбільш підходящої VM для виконання цього завдання, шляхом обчислення часу завершення цього завдання у всіх віртуальних машинах та присвоєння вибраної задачі ідентифікованій віртуальній машині. Нарешті, балансування навантаження здійснюється шляхом перестановки замовлення на основі нового часу виконання в кожному VM (див. Алгоритм балансування навантаження).

Опишемо прототип алгоритму розподілу наванатження IWRR. Цифрами позначено коментарі до дій алгоритму, літерами визначені кроки логіки самого алгоритму.

(1) Визначте число задач що виконуються/овікуються в кожній віртуальній машині і організуйте його в порядку зростання в черзі.

(a) Set numTaskInQueue = Number of Executing/Waiting Tasks в VM і організовані в порядку -зростання.

(2) Якщо число задач в першому елементі черги більше або рівне 1, то відмініть логіну балансування навантаження, в іншому разі перейдіть до кроку 3.

(a) If (numTaskInQueue.first() ≥1) then

Return;

(3) Якщо число задач в першому елементі черги менше або рівне 1, то відмініть логіну балансування навантаження, в іншому випадку перейдіть до кроку 4.

(a) If (numTaskInQueue.last()≤1) then

Return;

(4) Визначте час очікування в кожеій VM шляхом додавання довжичини очіквання виконання зі списків виконання, очікування і паузи, і потів розділіть отримкне значення на можлівості опрацювання віртуальної машини.

(a) Set pendingJobsTotLength = JobsRemainingLengthInExecList + JobsRemainingLengthInWaitList + JobsRemainingLengthInPauseList

(b) Set pendingExecutionTime = pendingJobsTotLength/𝒞Vm

(5) Розставте віртуальні машини від найменшого до найбільшого часу очікування і згрупуйте хз у разі однакових показників часу очікування.

(a) Відсортуйте VMMap за часом очікування кожної VM

(6) Видаліть завдання з віртуальнох машини з більшим часом очікування яка має більше ніж одну задачу та запишіть цю задачу віртуальній машині з меншим часом очіквання.

```
(a) While (true)
        Set OverLoadedVM = VMMap.get(VMMap.size())
        Set LowLoadedVM = VMMap.get(0)
        Varlowerposition = 1;
        Varupperposition = 1;
        (b) While(true)
            If (OverLoadedVM.taskSize() > 1 &&LowLoadedVM.taskSize() < 1)
            Break;
```

Else if (OverLoadedVM.taskSize() > 1)

LowLoadedVM = VMMap.get(lowerposition)

Lowerposition++

Else if (LowLoadedVM.taskSize() < 1)

OverLoadedVM = VMMap.get(VMMap.size() - upperposition)

Upperposition++

Else

Break The Outer While Loop

(c) End While

Set migratableTask = OverLoadedVM.getMigratableTask()

LowLoadedVM.assign(migratableTask)

Break

(d) End While

Таке розподілення навантаження буде викликане кожним завершенням завдання незалежно від віртуальної машини.

Чергування в багаторівневих взаємозалежних завданнях.

Багаторівневі взаємозалежні завдання пояснені на малюнку 4.3.

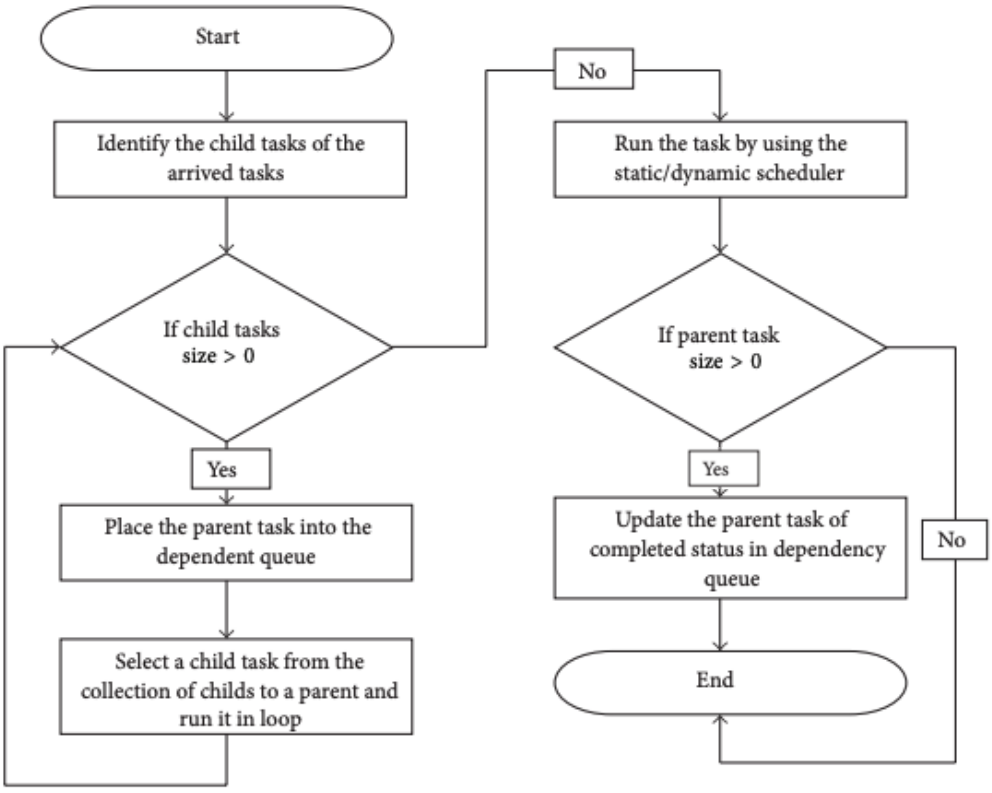

Рисунок 5.3 Блок-схема багаторівневих взаємозалежних завдань

5.4 Модель системи розподілу навантаження та планування.

Нехай VM = (VM1, VM2, ..., VMm) - це набір "m" віртуальних машин, який повинен обробляти "n" завдань, представлені набором T = ($T1$, $T2$, ..., Tn). Всі віртуальні машини працюють паралельно і не пов'язані між собою, і кожна віртуальна машина працює з власних ресурсів. Ресурси не діляться між віртуальними машинами. Ми плануємо залежні завдання для цих віртуальних машин, в яких "n" завдань, призначені для "m" VM, представлені у вигляді моделі з (4) - (5).

Час обробки. Нехай PT_{ij} - час обробки призначення завдання "i" для "j" VM і визначається як (4)

$$x_{ij} = \begin{cases} 1, & \text{if task “}i\text{” is assigned,} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Тоді модель лінійного програмування задана як (4)

$$\begin{aligned} \text{Minimize} \quad & Z = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathrm{PT}_{ij} x_{ij} \\ \text{Subject to:} \quad & \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, 2, \ldots, m \\ & x_{ij} = 0 \text{ or } 1. \end{aligned} \quad (5)$$

Використання ресурсів. Максимізація використання ресурсів є ще однією важливою метою, яка випливає з (6) та (7). Досягнення високого використання ресурсів стає викликом.

$$\text{Average utilization} = \frac{\sum_{j \in \text{VMs}} CT_j}{\text{Makespan} * \text{Number of VMs}}, \quad (6)$$

де makepan можна виразити як (7)

$$\text{Makespan} = \max\left\{\frac{CT_j}{j \in \text{VMs}}\right\}. \quad (7)$$

Потужність віртуальної машини. Будемо вважати що

$$C_{\text{VM}} = \mathrm{pe}_{\text{num}} * \mathrm{pe}_{\text{mips}}, \quad (8)$$

де C_{VM} - це ємність VM (див. (8)), pe_{num} - це кількість оброблюваних елементів у VM, а pe_{mips} - це мільйон інструкцій у секунду PE.

Потужність всіх віртуальних машин. Будемо вважати що

$$C = \sum_{j=1}^{m} C_{\text{VM}_j}, \tag{9}$$

де C є підсумком потужностей всіх віртуальних машин, потужність, що розподіляється на додаток / середовище (див. (9)).

Довжина підзавдання. Будемо вважати що

$$\text{TL} = T_{\text{mips}} * T_{\text{pe}}. \tag{10}$$

Довжина завдання. Будемо вважати що

$$\text{JL} = \sum_{k=1}^{p} \text{TL}_i, \tag{11}$$

де "p" - це кількість взаємозалежних завдань для роботи. Коефіцієнт задачі завантаження. Коефіцієнт навантаження задачі розраховується в (12) та (13) для ідентифікації та розподілу завдань на віртуальні машини. Він визначається як

$$\text{TLR}_{ij} = \frac{\text{TL}_i}{C_{\text{VM}_j}}, \quad i = 1, 2, \ldots, n \text{ tasks}, \ j = 1, 2, \ldots, m \text{ Virtual machines}, \tag{12}$$

де TLi - це тривалість завдання, яка оцінюється на початку виконання, і C_{VM} є потенціалом VM. Розглянемо

$$\text{If } \text{TLR}_{ij} = \begin{cases} 0, & \text{assign the task to VM}, \\ \text{Otherwise}, & \text{Do not assign}. \end{cases} \tag{13}$$

4.5 Експериментальна перевірка продуктивності запропонованого алгоритму IWRR.

Продуктивність алгоритму IWRR була проаналізована завдяки результатам симуляції цього алгоритму зробленої в симуляторі CloudSim. Далі в алгоритмах RR, WRR та IWRR аналізується час реакції, кількість міграцій завдань, сукупний

час простою всіх завдань та кількість затримних завдань при комбінації різнорідних та однорідних завдань з неоднорідними ресурсами. Деталі конфігурації наведено на рисунку 5.4, який є принтскріном з таблиці конфігурації CloudSim.

Sl. number	Entity	Quantity	Purpose
1	Data center	1	Data center having the physical hosts in the test environment
2	Number of hosts in DC	500 hosts (200 Nos-4-CoreHyperThreadedOpteron270, 200 Nos-4-Core HyperThreadedOpteron2218 and 100 Nos 8-Core HyperThreaded XeonE5430)	Number of physical hosts used in the experiment
3	Number of process elements	8/8/16	Number of executing elements in each of the hosts. The host has 8 or 16 processing elements
4	PE processing capacity	174/247/355 MIPS	Each host has any one of the processing capacities
5	Host ram capacity	16/32 GB	Each host has any of these RAM memories
6	Number of VM	10 to 100 with an increment of 10	Number of virtual machines used in the experiment
7	Number of PE to VM	1	Processing element allotted in each VM
8	VM's PE processing capacity	150/300/90/120/93/112/105/225	Virtual machine's processing capacity
9	VM RAM capacity	1920 MB	The RAM's memory capacity of the VM
10	VM manager	Xen	The operating system runs on the physical machine to manage the VMs. It provides the virtualization
11	Number of PE in Tasks	1	The job/task's maximum usable processing element
12	Task length/instructions	500000 to 200000000	Tasks length in million instructions. The heterogeneous job length test having the variations from the mentioned minimum to maximum

Рисунок 5.4 Деталі конфігурації "Хмари"

Порівняння повного часу виконання.

На рисунках 5.5 та 5.6 доведено, що IWRR забезпечує швидший час завершення, ніж інші 2 алгоритми балансування навантаження (RR і WRR) за умови неоднорідних ресурсів (VM) та однорідних завдань. Алгоритм статичного планувальника IWRR розглядає задану тривалість завдання разом із обсягом обробки гетерогенних віртуальних машин для призначення завдання. Таким чином, більша кількість робочих місць потрапляє на VM з вищою продуктивністю. Це допомагає завершити роботу за короткий час.

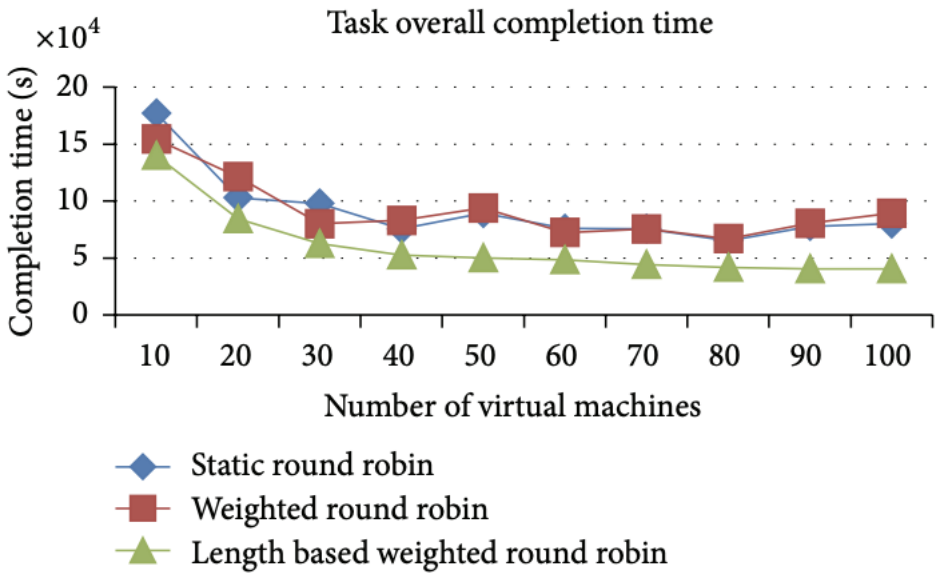

Рисунок 5.5 Час завершення виконання (розділення простору)

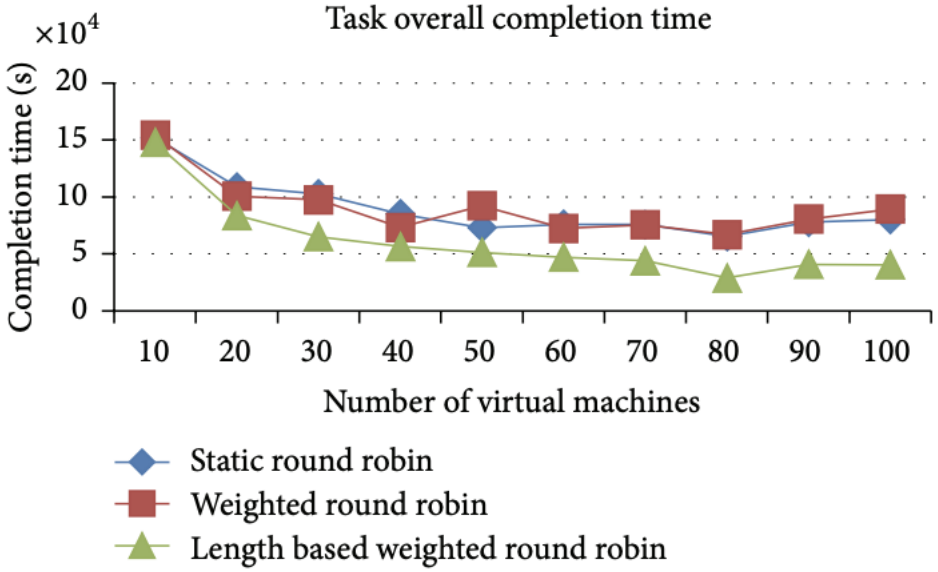

Рисунок 5.6 Час завершення виконання (розділення часу)

Балансувальник навантаження в IWRR починає працювати в кінці кожного завдання завершення. Якщо балансувальник завантаження знаходить, що будь-який VM завершує всі призначені ним завдання, то він буде ідентифікувати високогружену віртуальну машину з групи та обчислювати можливий час завершення цих завдань, присутніх у високо завантаженій VM і найменш завантаженій VM. Якщо найменш завантажена віртуальна машина може завершити будь-яке завдання, що присутнє у високо завантаженій віртуальній машині в

коротший термін, тоді ця робота буде переміщена до менш завантаженої віртуальної машини.

WRR розглядає співвідношення пропускної здатності віртуальної машини до загальної ємності віртуальної машини та призначає пропорційну кількість прибулих завдань у віртуальну машину. Але якщо на основі вищезгаданого розрахунку призначено будь-які довгі завдання для віртуальних машин з низькою продуктивністю, це призведе до затримки часу завершення виконання.

Простий RR не розглянув жодних данних про середовище, можливості віртуальної машини та тривалість роботи. Він просто призначає завдання віртуальним машинам одне за одним в упорядкованому стані. Таким чином, час завершення завдань вище, ніж 2 інших алгоритмів.

Порівняння міграції завдань.

Міграції завдань є мінімальними в алгоритмі IWRR завдяки надлишковому алгоритму статичного та динамічного планувальника для виявлення найбільш підходящої віртуальної машини для кожної з задач, що показано на рисунках 5.7 та 5.8. Таким чином, балансувальник навантаження не зміг знайти подальшої оптимізація для виконання завдання в найкоротший термін. Але у випадку алгоритмів WRR та RR, статичний та динамічний планувальник не врахував тривалість роботи. Натомість він розглядає лише ресурсні можливості та список отриманих завдань. Таким чином, балансувальник навантаження зумів знайти подальшу оптимізацію під час виконання, і він зміщує роботу з завантаженої VM на невикористані VM. Цей процес змушує більшу кількість міграцій завдань у WRR і RR алгоритмах. Кількість міграці в меншій кількості ресурсів висока в алгоритмах WRR та RR.

Порівняння затримних завдань та комбінованого часу очікування всіх завдань (розділення простору)

Кількість затримених завдань та комбінований час простою всіх завдань вищі в IWRR, ніж в інших двох алгоритмах, що показано на рис. 5.9 і 5.10. Це збільшення відбулося через виділення більшої кількості задач віртуальним

машинам з більш високими можливостями. У будь-який момент часу в системі з розділенням простору може опрацьовуватися лише одне завдання, навіть якщо VM має можливості для обробки більшої кількості. Отже, якщо ще одне завдання булу розподілено на ту саму VM завдяки її вищої потужності, то це завдання повинно бути в стані очікування, доки не завершитсья обробка попереднього завдання. Це збільшує кількість затриманих завдань та час простою в алгоритмі IWRR.

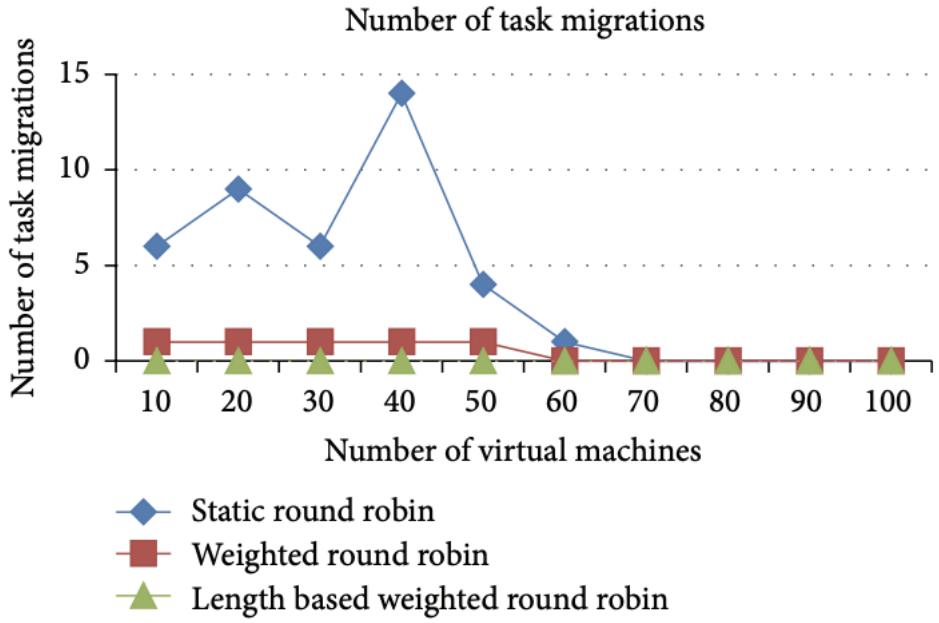

Рисунок 5.7 Число міграцій завдань (розділення простору)

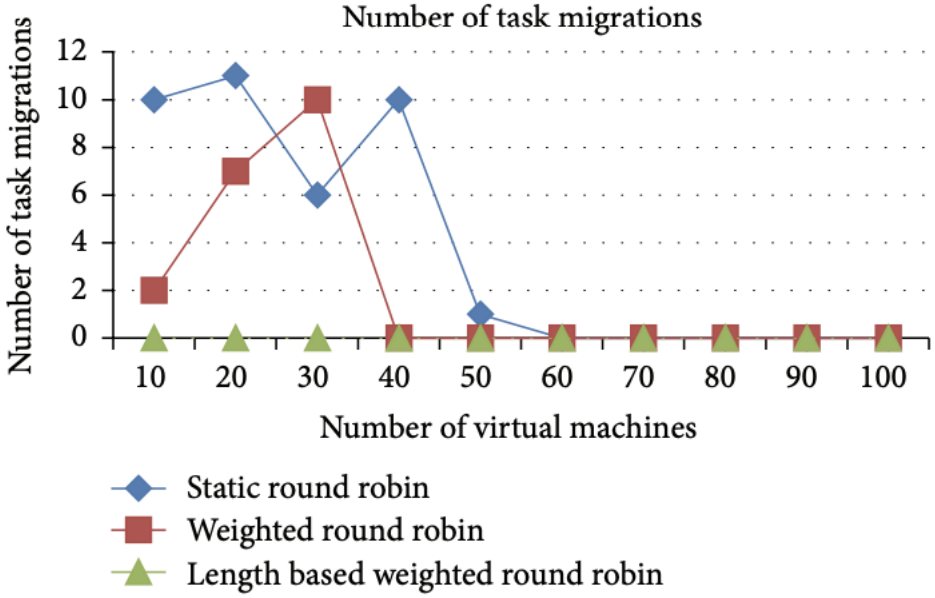

Рисунок 5.8 Число міграцій завдань (розділення часу)

Але в двох інших алгоритмах завдання потрапляють навіть на віртуальні машини нижчої потужності без точного прогнозування можливого часу завершення роботи в різних віртуальних машинах. Таким чином, кількість затриманих завдань та комбінований час простою всіх завдань нижче в RR і WRR.

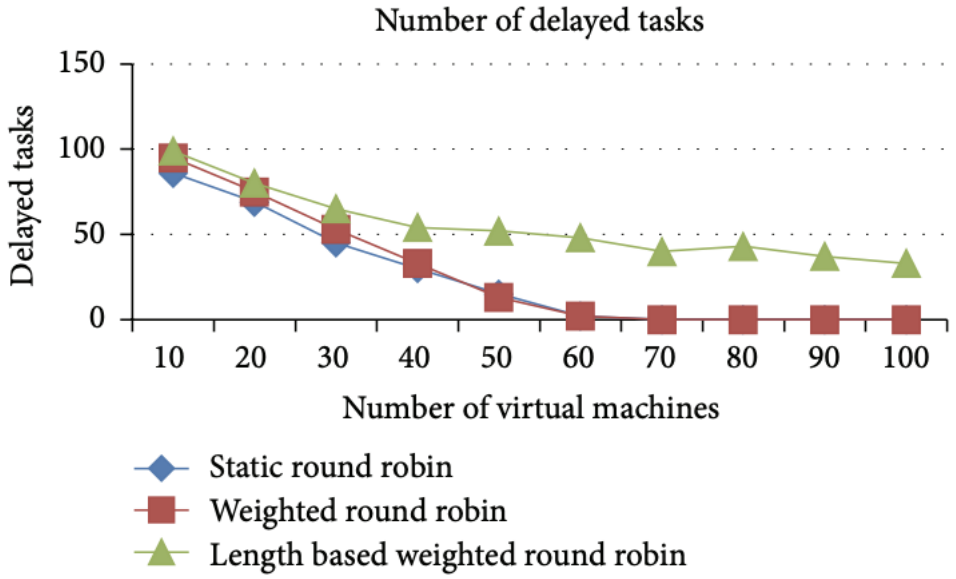

Рисунок 5.9 Число затриманих завдань (розділення простору)

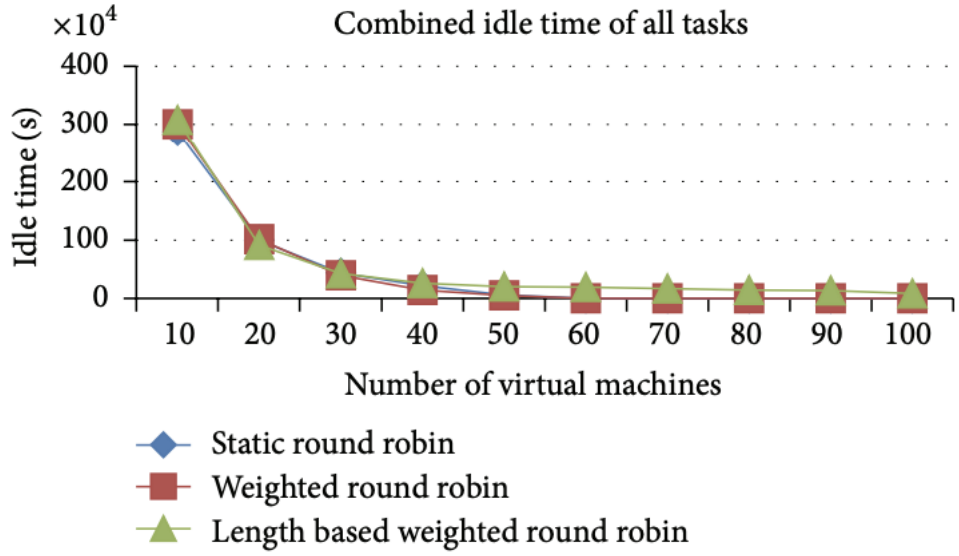

Рисунок 5.10 Час простою всіх задач (розділення простору)

ВИСНОВКИ

З результатів експериментального порівняння широко відомих та використовуваних алгоритмів RR, WRR і нового запропонованого IWRR алгоритму розподілу навантаження, основний принцип якого базується на вираховуванні можливостей кожнох віртуальної машини до того як призначити на неї задачу, видно що останній за рахунок набагато більш продумного способу розподілу задач є безумовно кращим у питаннях швидкості вирішення задач та кількості міграцій між віртуальними машинами.

Єдиним фактором, який виявився гіршим у розробленому алгоритмі є число затриманих задач та час простою всіх задач, які є взаємозалежними мірами. Це викликано саме тим фактом, що запропонованй алгоритм вираховує потужності машин і обирає машину виключно за умови, що завдання на ній виконається найшвидше. Це означає, що коли одна віртуальна машина має набагато більші можливості ніж інша, планувальник розподілу задач призначить декількі задач на першу віртулальну машину, і за рахунок цього задачі будуть затримані на деякий час. Але при цьому загальний час виконання всіх задач є швидшим за рахунок прорахованого розподіленнязавдань.

ПЕРЕЛІК ПОСИЛАНЬ

1. Generations of mobile networks [Електронний ресурс] // Режим доступу: https://www.slideshare.net/imrajdss/generation-of-mobile-networks
2. 4G mobile networks [Електнґронний ресурс] // Режим доступу: https://www.4g.co.uk/what-is-4g/
3. Introduction to wireless LTE 4G architecture [Електронний ресурс] // Режим доступу: http://www.cost605.org/cost605school2011/1-LF-Pau-LTE.pdf
4. 4G LTE Advanced [Електронний ресурс] // Режим досутпу: https://www.radio-electronics.com/info/cellulartelecomms/lte-long-term-evolution/3gpp-4g-imt-lte-advanced-tutorial.php
5. Fundamentals of 5g Mobile Networks [Електронний ресурс] // Режим доступу: http://5g.itrc.ac.ir/sites/default/files/Fundamentals%20of%205G%20Mobile%20Networks-Wiley%20%282015%29.pdf
6. Cloud computing overview [Електронний ресурс] // Режим доступу: https://www.thbs.com/downloads/Cloud-Computing-Overview.pdf
7. Cloud computing models [Електронний ресурс] // Режим доступу: http://web.mit.edu/smadnick/www/wp/2013-01.pdf
8. Introduction to Cloud computing [Електронний ресурс] // Режим доступу: https://www.dialogic.com/~/media/products/docs/whitepapers/12023-cloud-computing-wp.pdf
9. Suzanne, C., “AT&T and T-Mobile network sharing for those in Sandy’s path", NBC News, Oct. 31, 2012.
10. China Mobile,“C-RAN The Road Towards Green RAN", White Paper, China Mobile Research Institute, Oct 2011.
11. Common Public Radio Interface (CPRI); Interface Specification, V4.2, Sept 2010.
12. Cloud based mobile network sharing [Електронний ресурс] // Режиим доступу: https://www.researchgate.net/publication/276463953
13. Market Research Report: Emerging Market Opportunities, Analysis Mason, May 2010.

14. Market Research Report: Emerging Market OpportunitiesaBarb Gonzalez, “All About Internet Speed Requirements for Hulu, Netflix, and Vudu Movie Viewing"., Analysis Mason, May 2010.
15. Comparing Load Balancing Algorythms [Електронний ресурс] // Режим доступу: https://www.jscape.com/blog/load-balancing-algorithms