

## ЛЕКЦІЯ 10

### ТУМАННІ ОБЧИСЛЕННЯ ТА ЇХ ТОПОЛОГІЯ

Туманні обчислення – це модель, яка забезпечує обчислення та зберігання даних між кінцевими пристроями та традиційними центрами хмарних обчислень. Туманні обчислення – концепція за якою частина даних обробляються в локальних мережах, а не виключно в дата-центрі. Будь-який пристрій, що має обчислювальні здібності, сховище та мережу підключення, може бути вузлом туману. Приклади пристроїв включають промислові контролери, комутатори, маршрутизатори, вбудовані сервери та камери відеоспостереження.

У 2011 році виникла потреба у розширенні хмарних обчислень за допомогою туманних обчислень, щоб впоратися з величезною кількістю пристроїв IoT та великими обсягами даних для реальних програм із низькою затримкою.

В 2015 року Cisco Systems, ARM Holdings, Dell, Intel, Microsoft та Princeton University створили консорціум OpenFog для просування інтересів та розвитку в туманних обчислень. Керуючий директор Cisco-старший Гельдер Ентюнес став першим головою консорціуму, а головний голова IoT-стратег Джефф Феддерс став його першим головою.

#### **Основні характеристики туманних обчислень:**

- ☐ Проінформованість про місцезнаходження пристрою.
- ☐ Низька затримка. Оскільки туманні обчислення знаходяться ближче до кінцевих пристроїв, вони забезпечують меншу затримку при обробці даних кінцевих пристроїв.
- ☐ Мобільність. Для багатьох програм Fog важливо спілкуватися безпосередньо з мобільними пристроями, і підтримка відповідних протоколів (наприклад, LISP).

- Взаємодія в режимі реального часу. Програми для туманних обчислень забезпечують взаємодію між вузлами у режимі реального часу, а не пакетної обробки, що використовується в хмарних обчисленнях.
- Сумісність. Вузли туманних обчислень можуть взаємодіяти та працювати з різними доменами та постачальниками послуг.

### **Переваги туманних обчислень**

Контроль конфіденційності. Завдяки туманним обчисленням ви можете краще контролювати рівень конфіденційності. Ви можете обробляти та аналізувати чутливі дані локально, а не надсилати їх до централізованої хмари для аналізу.

Підвищення продуктивності. Туманні обчислення допомагають підвищити продуктивність і збільшити швидкість бізнес-процесів. Вони можуть дозволити пошук лише тих даних, які потребують негайної взаємодії людини, а не всіх даних, скорочуючи час і зусилля, необхідні для пошуку потенційних проблем.

Безпека даних. Туманні обчислення дозволяють під'єднати до мережі кілька пристроїв. Замість одного централізованого місця, яке може стати вразливим, діяльність відбувається між різними локальними кінцевими точками, що полегшує ідентифікацію загроз, таких як заражені файли, потенційні хаки або зловмисне програмне забезпечення.

### **Недоліки туманних обчислень**

Складність. Система, яка включає в себе безліч пристроїв, які зберігають і аналізують власні дані, які можуть бути розташовані в будь-якому місці, в будь-який час, додає складності мережі, яка колись надсилала всі свої дані до централізованого місця.

Ризики. Збільшення кількості з'єднань означає збільшення ризиків. Завдяки численним пристроям і численним користувачам, ризик виходу

зіпсованих або заражених файлів, додатків або інформації в основний потік даних компанії різко зростає.

Локальні дані. Використовуючи туманні обчислення, великий об'єм даних зберігається на самих пристроях. Ці пристрої часто знаходяться за межами фізичного місцезнаходження офісу, і багато компаній або менеджерів підприємств вважають, що ця конфігурація збільшує ризик порушення даних

### **Порівняння туманних, граничних і хмарних обчислень**

Граничні обчислення як обробку, що переміщається близько до того місця, де генеруються дані. У разі IoT граничним пристроєм може бути сам датчик з невеликим мікроконтролером або вбудованою системою, здатною до WAN-зв'язку. В інших випадках межа буде шлюзом в архітектурі з особливо обмеженими кінцевими точками, що змушують шлюз зависати. Гранична обробка також зазвичай згадується в контексті «машина-машина», де існує щільна кореляція між краєм (клієнтом) і сервером, розташованим в іншому місці.

Граничні обчислення існують, як зазначено, для усунення проблем із затримкою і непотрібним завантаженням смуги пропускання, а також для додавання таких сервісів, як денатурація і безпека, близьким до джерела даних. У граничного пристрою може бути зв'язок з хмарним сервісом ціною затримки і несучої; він не бере активної участі в хмарній інфраструктурі. Хмарні обчислення трохи відрізняються від парадигми граничних обчислень. У хмарних обчисленнях в першу чергу розділяється API інфраструктури та стандарти зв'язку з іншими туманними вузлами і/або хмарною службою. Туманні вузли є розширеннями хмари, тоді як граничні пристрої можуть використовувати або не використовувати хмару.

Іншим ключовим принципом туманних обчислень є те, що туман може існувати в ієрархічних шарах. Туманні обчислення також можуть балансувати навантаження і керувати даними зі сходу на захід і з півночі на

південь, щоб допомогти в балансуванні ресурсів. З визначення хмари і його сервісів, які були описані в попередньому розділі, можна подумати про ці туманних вузлах як просто про ще одні (хоча і менш потужні) інфраструктури в гібридній хмарі.

### **Туманні топології**

Туманні топології можуть існувати в багатьох формах, і архітектору необхідно враховувати кілька аспектів при розробці туманної системи «від краю до краю». Зокрема, при розробці топології вступають в силу обмеження, такі як вартість, навантаження на процесор, інтерфейс виробника і передача між сходом-заходом. Туманна мережа може бути простою, як граничний маршрутизатор з підтримкою туману, що з'єднує датчики з хмарним сервісом. Вона також може ускладнюватися в багаторівневу туманну ієрархію туману з різним ступенем здатності обробки в кожному шарі і ролями на кожному рівні, одночасно перерозподіляючи навантаження обробки, коли і де це необхідно (зі сходу на захід і з півночі на південь).

Визначальні фактори моделей засновані на наступному:

- зменшення обсягу даних - наприклад, система збирає неструктуровані відеодані з тисяч датчиків або камер, агрегує дані і шукає конкретні події в режимі реального часу. Якщо це так, то скорочення набору даних буде значним, так як тисячі камер будуть щодня виробляти сотні ГБ даних, а туманним вузлам потрібно буде перевести великі обсяги даних в прості форми: «так», «ні», «небезпека», «токени безпеки події»;
- кількість граничних пристроїв - якщо система IoT являє із себе всього лише один датчик, то набір даних малий і може не виправдовувати використання граничного туманного вузла. Однак, якщо число датчиків зростає або, в гіршому випадку, зростання кількості датчиків

непередбачуване і динамічне, то туманна топологія може зажадати масштабування вгору або вниз динамічно. Ми розглядаємо як приклад стадіон з використанням Bluetooth-маяка. Оскільки аудиторія зростає на певних майданчиках, система повинна мати можливість масштабуватися нелінійно. В інших випадках стадіон може займати лише невелику площу, і йому потрібні тільки обмежені ресурси обробки та підключення;

- можливості туманного вузла - в залежності від топології і вартості деякі вузли можуть краще підходити для підключення до систем WPAN, тоді як інші вузли в ієрархії можуть мати додаткові можливості по обробці для машинного навчання, розпізнавання образів або роботи із зображеннями. Прикладом можуть бути граничні туманні вузли, які управляють безпечною сіткою Zigbee і мають спеціальне обладнання для аварійних ситуацій або безпеки WPAN. Вище цього рівня буде існувати вузол обробки туману, який буде мати додаткову оперативну пам'ять і GPGPU для підтримки потоків необроблених даних з шлюзів WPAN;
- надійність системи - архітекторів може знадобитися розгляд форм відмови в моделі IoT. Якщо один крайній туманний вузол дав збій, інший міг би зайняти його місце для виконання якої-небудь дії або сервісу. Цей випадок важливий в життєво-критичних випадках або при роботі в реальному часі. Таким же чином додаткові туманні вузли можуть підключатися на вимогу; надлишкові вузли можуть знадобитися в ситуаціях забезпечення відмовостійкості. У разі відсутності додаткових надлишкових вузлів деяка обробка може спільно здійснюватися сусідніми вузлами за рахунок використання системних ресурсів і затримки, але система буде продовжувати функціонувати. Крайовий варіант використання - це те, де сусідні вузли діють як сторожові пси один для одного. В разі збою туманного

вузла або збою зв'язку з вузлом сторожовий таймер сигналізує про подію збою для хмари і може виконувати локальні критичні дії. Хорошим прикладом є випадок, коли туманний вузол дає збої при контролі трафіку на шосе; сусідній вузол може побачити відмову точки, попередити хмару про подію і подати сигнал на щит на шосе, щоб зменшити швидкість.

Найпростішим рішенням для туману є блок обробки на краю (шлюз, тонкі клієнти, маршрутизатор), розташовані поруч з матрицею датчиків. Тут туманний вузол може використовуватися в якості шлюзу для мережі або mesh-мережі WPAN і обмінюватися даними з хостом (рис. 10.1).

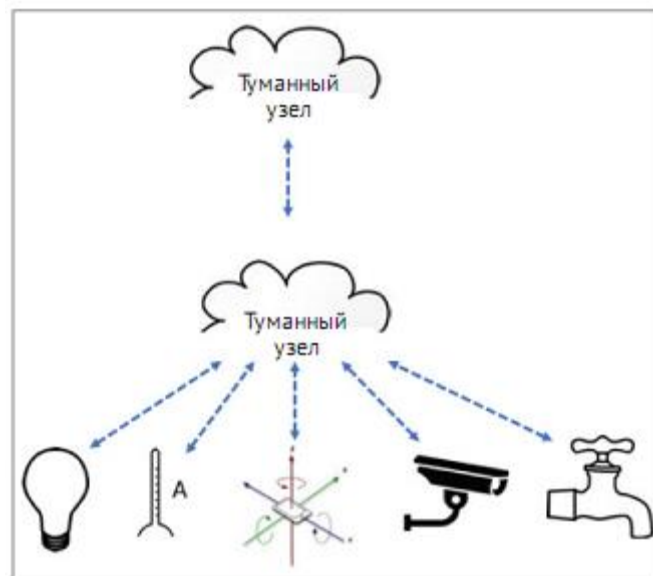


Рис. 10.1 Проста туманна топологія. Гранично-туманний пристрій управляє масивом датчиків і може взаємодіяти з іншим туманним вузлом на основі M2M

Наступна базова туманна топологія включає хмару як батька по туманній мережі. Туманний вузол в цьому випадку буде збирати дані, захищати край і виконувати обробку, необхідну для зв'язку з хмарию. Цю модель відокремлює від граничних обчислень те, що сервісні і програмні

рівні туманного вузла поділяють відносини з хмарним фреймворком (рис. 10.2).

Наступна модель використовує кілька туманних вузлів, відповідальних за послуги і обробку на краю, і кожен з них підключається до набору датчиків.

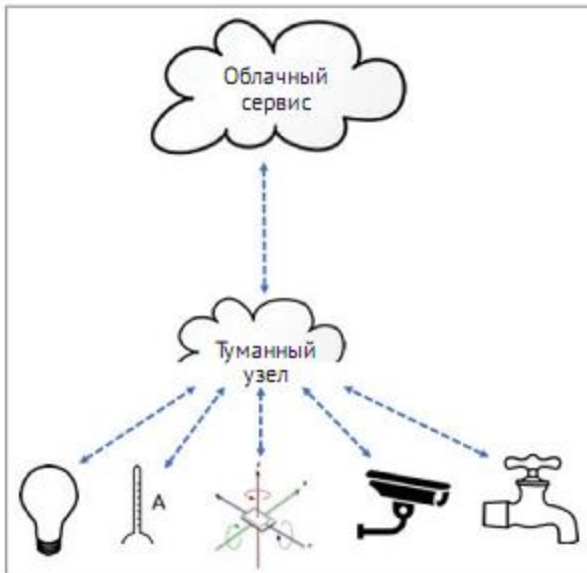


Рис. 10.2 Туман до хмарної топології. Тут туманний вузол встановлює відносини з постачальником хмари

Батьківська хмара забезпечує кожен туманний вузол, так як якщо б він був єдиним вузлом. Кожен вузол має унікальний ідентифікатор, щоб забезпечити унікальний набір сервісів на основі розташування. Наприклад, кожен туманний вузол може перебувати в іншому місці для роздрібної мережі. Туманні вузли також можуть пов'язувати і передавати дані зі сходу на захід між граничними вузлами. Прикладом можуть служити умови холодного зберігання, коли необхідно підтримувати і управляти вентиляторами і морозильниками, щоб запобігти псуванню продуктів харчування. У роздрібного продавця може бути кілька вентиляторів в різних місцях, все управляється єдиною хмарною службою, але працюють з туманними вузлами на краю (рис. 10.3).

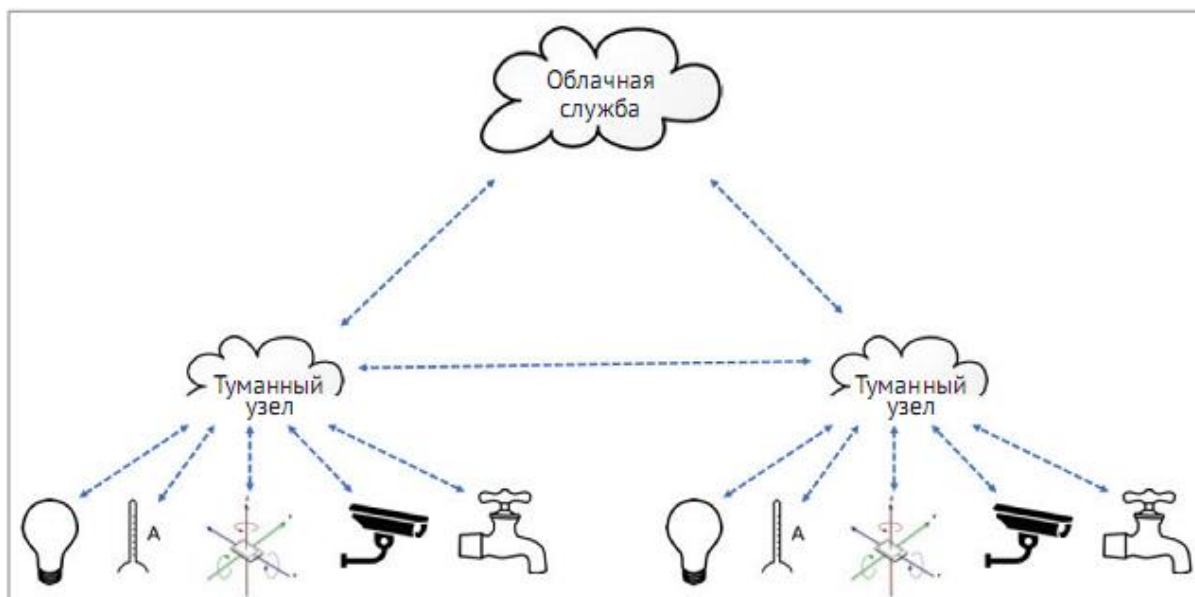


Рис. 10.3 Кілька туманних вузлів з однією основним хмарою

Наступна модель розширює другу топологію, додаючи можливість надійно і конфіденційно спілкуватися з декількома постачальниками хмар від декількох туманних вузлів. У цій моделі можуть бути розгорнуті кілька батьківських хмар. Наприклад, в розумних містах можуть існувати численні географічні райони, і вони можуть охоплюватися різними муніципалітетами. Кожен муніципалітет може вибрати одного хмарного провайдера, порівнявши його з іншими, але всі муніципалітети управляються з використанням одного затвердженого та бюджетного виробника камер і датчиків. У цьому випадку виробник камер і датчиків матиме один екземпляр хмари, що співіснує з декількома муніципалітетами. Туманні вузли повинні мати можливість доставляти дані декільком постачальникам хмарних обчислень (рис. 10.4).



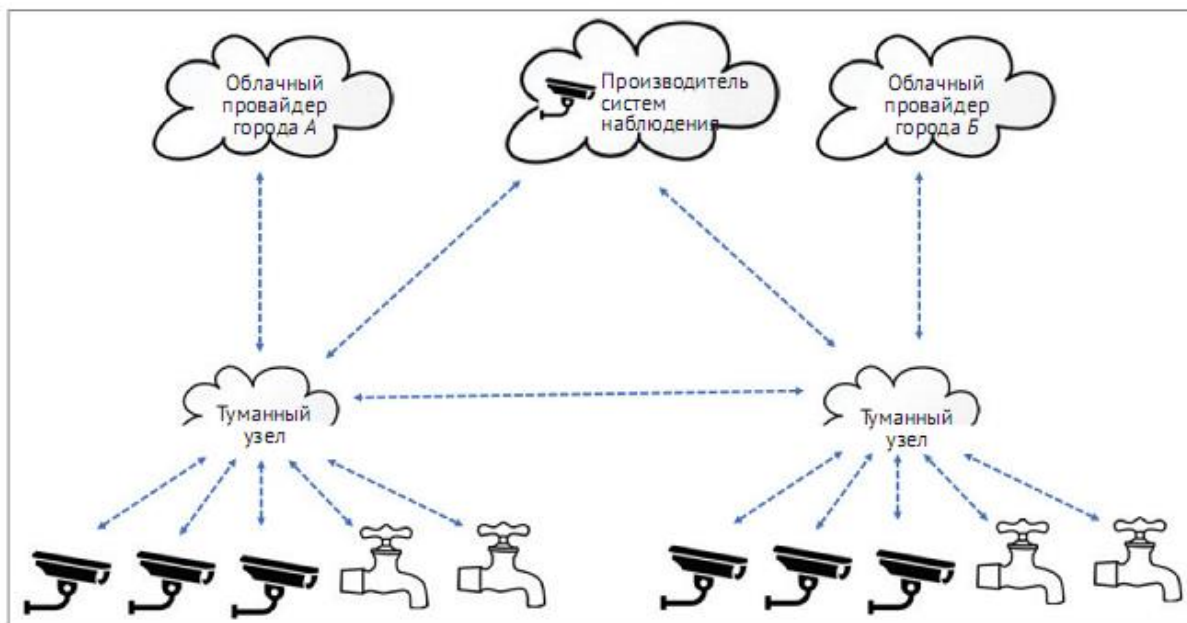


Рис. 10.4 Кілька туманних вузлів з кількома хмарними провайдерами. Хмари можуть являти собою суміш громадських і приватних хмар

Туманні вузли також не потребують індивідуального з'єднання, що зв'язує датчики з хмарами «один до одного». Туманні вузли можуть бути поміщені в стек, багаторівневими або навіть законсервованими до тих пір, поки не знадобляться. Ієрархія рівнів туманних вузлів один над одним може здаватися суперечливою, якщо ми намагаємося зменшити затримки, але, як згадувалося раніше, вузли можуть бути спеціалізованими. Наприклад, вузли, розташовані ближче до датчиків, можуть надавати послуги в режимі реального часу або мати обмеження за вартістю, що вимагає від них мінімального обсягу даних для зберігання і обчислення. Рівні над ними можуть надавати обчислювальні ресурси, необхідні для агрегованого зберігання, машинного навчання або розпізнавання зображень з використанням додаткових пристроїв, що запам'ятовують або GPGPU-процесорів. Наступний приклад ілюструє використання на прикладі освітлення міста.

Тут кілька камер відчують рух пішоходів і трафік; туманні вузли, які мають найтісніший контакт з камерами, виконують агрегацію і витяг

ознак і передають ці дані вгору до наступного рівня. Батьківський туманний вузол отримує дані і виконує необхідне розпізнавання зображень за допомогою алгоритму глибокого навчання. Якщо буде спостерігатися цікава подія (наприклад, пішохідна прогулянка вночі вздовж шляху), подію буде відправлено в хмару. Компонент хмари реєструє подію і сигналізує множині вуличних ліхтарів в околиці пішохода, щоб вони збільшили освітленість. Ця картина буде тривати до тих пір, поки туманні вузли бачать рух пішохода. Кінцевою метою є загальна економія енергії, щоб кожна вулична лампочка не працювала на повну інтенсивність весь час (рис. 10.5)

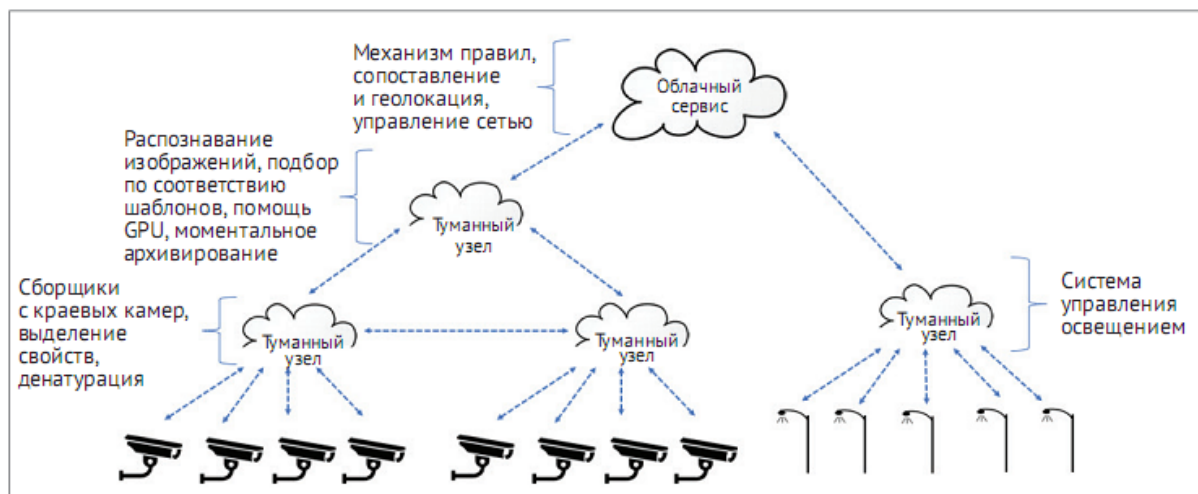


Рис. 10.5 Багаторівнева туманна топологія. Туманні вузли складаються в стек в ієрархії рівнів, щоб забезпечити додаткові послуги або абстракції

Туманні обчислення роблять хмарну обробку ближче до краю для вирішення проблем із затримкою, безпекою та витратами на зв'язок. Хмарні та туманні технології працюють разом, щоб забезпечити роботу аналітичних пакетів у вигляді движків правил для складних агентів обробки подій. Вибір моделі хмарних провайдерів, фреймворків, туманних вузлів і модулів аналітики є важливим завданням, і безліч матеріалів присвячено глибокому розгляду семантики програмування і створення цих сервісів. Архітектор повинен зрозуміти топологію і кінцеву мету системи, щоб

побудувати структуру, яка задовольняє сьогоднішні потреби і масштабується в майбутньому.