Франків О. В., Муляревич О. В.

Національний університет "Львівська політехніка",

кафедра електронних обчислювальних машин

**Опрацювання даних за допомогою хмарних технологій безсерверних обчислень**

*© Франків О. В., Муляревич О. В. 2019*

**Автором розглянуто проблему вибору методів обробки даних за допомогою технологій безсерверних обчислень, проаналізовано типові методи обробки даних з інтернету речей, розглянуто структурну схему життєвого циклу обробки даних, побудованого для прийому і обробки даних з допомогою технологій безсерверних обчислень.**

**Ключові слова: безсерверні обчислення, методи обробки, інтернет речей, AWS Lambda.**

**The author considers the problem of choosing data processing methods using technologies without server computing, analyzes the typical methods of data processing from the Internet of Things, considers the structural diagram of the data processing lifecycle built for receiving and processing data using technologies without server computing.**

**Keywords: serverless computing, processing methods, Internet of Things, AWS Lambda.**

**Вступ.** Останні декілька років технології хмарних обчислень набувають все більшої популярності. З кожним днем сфера їх використання росте – сервіси потокової передачі музики та відео, опрацювання великих пакетів даних, машинне навчання тощо. Один з основних факторів чому хмарні обчислення набувають такої популярності - це концепція «інтернет-сервіс», завдяки якій [користувач](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%87) має доступ до власних даних, але не може керувати і не повинен піклуватися про інфраструктуру, [операційну систему](https://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) і [програмне забезпечення](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F), з яким він працює. «Хмарою» метафорично називають [інтернет](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82), який приховує всі технічні деталі.

**Аналіз досліджень та публікацій.** На даний момент існуюча модель «безсерверних» хмарних обчислень інтерпретується двома способами. Одна з інтерпретацій це представлення цієї технології як технології моделі «Бекенд як служба» або BaaS, мета цього напрямку надати користувачу перелік API сервісів, які легко вбудувати в свої аплікації, що дозволить вирішити декілька важливих задач. Другий напрям – це концепція «функція як сервіс» або FaaS, цей архітектурний патерн використовується для керованих подіями систем. Їх часто використовують як зв’язуючу ланку між сервісами або керованими подіями системами.

**Постановка задачі**. Дослідження методів опрацювання данних за допомогою моделі «безсерверних» обчислень хмарних технологій.

**Основний матеріал. «**Безсерверні» технології - це своєрідн**а** модель хмарних обчислень, де платформа автоматично керує виділенням машинних ресурсів. Самі ж хмарні обчислення представляють собою розподілену систему обробки даних, де ресурси необхідні для виконання опрацювання даних, відносяться до моделі «інтернет-сервіс». Під час роботи з хмарними технологіями, зберігання даних, опрацювання інформації і проведення обчислень та інші операції проходять на серверах Всесвітньої Мережі Інтернет.

Обчислення безсервера - це метод надання сервісів, що використовуються як альтернатива методу «серверного» розташування ресурсів. Провайдер «безсерверних» ресурсів дозволяє користувачам писати та розгортати код, не турбуючись про базову інфраструктуру. Плата яку компанія, яка отримує сервісні послуги від постачальника «безсерверних» ресурсів, стягується з урахуванням їх обчислень, і не потрібно резервувати та оплачувати фіксовану кількість пропускної здатності або кількість серверів, оскільки послуга включає в себе автоматичне масштабування. Хоча модель і називається «безсерверною», фізичні сервери, все ще використовуються, але розробникам не потрібно турбуватись про них. У перші дні Інтернету кожен, хто хотів створити веб-додаток, повинен був володіти фізичним обладнанням, необхідним для роботи сервера, що є громіздким і дорогим завданням.

Потім з’явилася хмара, де фіксовану кількість серверів або кількість серверного простору можна було взяти в оренду віддалено. Розробники та компанії, які орендують ці фіксовані одиниці серверного простору, зазвичай перекупляють, щоб переконатись, що сплеск трафіку чи активності не перевищив би їх щомісячні ліміти та порушив їхні програми. Це означало, що велика частина серверного простору, за який платили, зазвичай витрачається на простій. Хмарні постачальники запровадили моделі автоматичного масштабування для вирішення проблеми, але навіть при автоматичному масштабуванні небажаного спалаху активності, наприклад DDoS Attack, це може виявитися дуже дорогим.

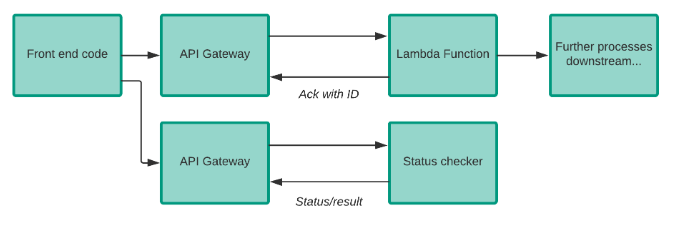
Переваги безсерверної моделі

Обслуговування безсерверів дозволяє розробникам купувати сервісні ресурси на гнучких принципах "оплата за час", тобто розробники повинні платити лише за послуги, які вони використовують. Це як перехід від плану даних мобільного телефону із встановленим місячним обмеженням, до такого, який стягує плату лише за кожен байт даних, який фактично використовується.

Термін "без сервера" дещо вводить в оману, оскільки досі існують сервери, що надають ці сервіси, але всі проблеми серверного простору та інфраструктури займаються постачальником. Без сервера означає, що користувачам такої платформи не треба мати справу з налаштуванням серверів для запуску коду: усі серверні налаштування, планування обчислювальних ресурсів цілком приховані від користувачів і керуються платформою. Розробка додатків, як правило, розділена на дві сфери: фронтенд і бекенд. Фронтенд - це частина програми, яку користувачі бачать та взаємодіють, наприклад, візуальний макет. Бекенд - це частина, яку користувач не бачить; сюди входить сервер, де живуть файли програми, та база даних, де зберігаються дані користувачів та логіка бізнесу.

Frontend vs Backend програми

Наприклад, уявимо веб-сайт, який продає квитки на концерти. Коли користувач вводить запит у вікно браузера, він пересилає запит на сервер, який відповідає на дані веб-сайту. Потім користувач побачить веб-сайт, який буде містити текст, зображення та поля форми, які користувач повинен заповнити. Потім користувач може взаємодіяти з одним із полів форми на передній панелі для пошуку свого улюбленого музичного акту. Коли користувач натисне "Надіслати", це викликає ще один запит на бекенд.



Бекенд код перевіряє свою базу даних, щоб побачити, чи існує виконавець з цим ім'ям, і якщо так, то коли він буде виступати наступного разу, і скільки квитків доступно. Потім опрацьовані дані будуть передані назад на фронтенд, а фронтенд відображатиме результати таким чином, щоб це мало сенс для користувача. Аналогічно, коли користувач створює обліковий запис і вводить фінансову інформацію для придбання квитків, відбудеться інше зворотне спілкування між фронтендом і бекендом. Більшість постачальників безсерверених послуг пропонують своїм клієнтам послуги баз даних та зберігання, а багато хто також мають платформи функціональних послуг (FaaS), як Cloudflare Workers. Ці платформи можуть виконувати фрагменти коду на краю, не зберігаючи жодних даних.

Переваги обчислень без сервера:

Нижчі витрати. Обчислення без сервера, як правило, дуже економічно вигідне, оскільки традиційні хмарні постачальники сервісних сервісів (розподіл сервера) часто призводять до того, що користувач платить за невикористаний простір або простою час процесора.

Спрощена масштабованість. Розробникам, які використовують архітектуру без сервера, не потрібно турбуватися про політику збільшення масштабу коду. Постачальник без сервера обробляє все масштабування на вимогу.

Спрощений код доповнення. За допомогою програми FaaS розробники можуть створювати прості функції, які незалежно виконують єдину мету, наприклад здійснення дзвінка API.

Швидший поворот. Архітектура без сервера може значно скоротити час на ринок. Замість необхідності складного процесу розгортання для виправлення помилок та нових функцій, розробники можуть додавати та змінювати код на основі деталей.

**Висновки**. В роботі розглянуто структуру та переваги безсерверної моделі технологій хмарних обчислень, а також наведено базовий приклад веб сайту з використанням безсерверної технології.

**Література**

1. [Andrea Passwater When (and why) not to go serverless](https://serverless.com/author/andreapasswater/) **Mode of access: World Wide Web:**  [<https://serverless.com/blog/when-why-not-use-serverless/>](http://www.gartner.com/id=2057415/) **(viewed on September 11, 2019). — Title from the screen.**

2. Науменко Т.О. безсерверна технологія (Function as service) для створення хмарних мікросервісних додатків [Электронний ресурс] / Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» [Київ: «комп’ютерно-інтегровані технології освіта, наука, виробництво», 2018].

3. Serverless [Электронний ресурс] / Хабрахабр. — Электронные данные. — [Москва: TechMedia, 2015]. — Режим доступа: <https://habr.com/ru/company/selectel/blog/452266/> (дата обращения 15.9.2019). — Название с экрана.

4 Bai, Haishi. (2019). Serverless. 10.1201/9780429278594-7. [Електронний ресурс]

// Інтернет-видання ResearchGate. — Електронні дані.

Режим доступу <https://www.researchgate.net/publication/333212660_Serverless> (дата звернення 17.9.2019 р.). Назва з екрану.

5. Alteen, Nick & Fisher, Jennifer & Gerena, Casey & Gruver, Wes & Jalis, Asim & Osman, Heiwad & Pagan, Marife & Patlolla, Santosh & Roth, Michael. (2019). Serverless Compute. 10.1002/9781119549451.ch12. . [Електронний ресурс]

// Інтернет-видання ResearchGate. — Електронні дані.

Режим доступу https://www.researchgate.net/publication/335515135\_Serverless\_Compute (дата звернення 17.9.2019 р.). Назва з екрану.