

П. Е. СИТНИКОВА, О. П. ГОВДЕРЧАК

AWS STEP FUNCTIONS ЯК НЕВІД'ЄМНА СКЛАДОВА АРХІТЕКТУРИ СИСТЕМИ З ПОСЛІДОВНИМ ВИКОНАННЯМ БЕЗСЕРВЕРНИХ ФУНКЦІЙ

У статті досліджено використання сервісу Step Functions хмарного провайдера Amazon Web Services (AWS) у системах, що мають послідовне виконання функцій/дій в алгоритмі свого виконання. Досліджено також сервіс AWS, що може бути використаний в будь-якій організації для будь-якої системи задля отримання стану виконання усього ланцюга функцій.

1. Вступ

Створюючи безсерверні додатки, розробники, як правило, хочуть масштабувати програму відповідно до зростаючого навантаження, зберігаючи при цьому низькі витрати і дозволяючи декільком командам одночасно працювати над різними частинами програми. У безсерверній моделі однією з найкращих практик для досягнення цих цілей є розділення бізнес-логіки програми на набір нерозв'язаних служб. Велика безсерверна програма може складатися з десятків або сотень безсерверних служб [1].

Проблема виникає, коли така велика кількість служб потребує доступу до різних частин спільного стану. Для ефективної роботи цих служб команди також повинні мати можливість організувати потік даних через усі служби додатків в одному місці, і це саме те, що обробляє крокові функції AWS. Step Functions стали найважливішим елементом безсерверної екосистеми завдяки всьому стану та керуванню даними, необхідними для ефективної роботи безсерверних систем у масштабі.

Більше того, AWS Step Functions дозволяє організувати ваші завдання AWS за допомогою візуального редактора робочих процесів. Ви використовуєте цей редактор для створення діаграм стану машини або покрокових функцій, які потім стають основою побудови, спільного використання та модифікації поведінки додатків [2].

2. Постановка задачі

На сьогоднішній день дуже багато додатків використовують хмарні обчислення для спрощення або покращення їхньої роботи, тому питання, чи використовувати безсерверні обчислення у даній роботі не розглядається, це взято за основу системи. Як провайдер сервісів хмарних обчислень розглядається Amazon Web Services (AWS).

У даній статті розглядатиметься проблема послідовного виклику безсерверних функцій (Serverless, FaaS - Function as a Service), які у даного представника хмарних послуг називаються AWS Lambda, та можливості вирішення цих проблем з допомогою сервісу AWS Step Functions.

Досліджуючи необхідність мікросервісних додатків запускати деякі частини логіки системи послідовно, розбиваючи їх на окремі сервіси, можна дійти до висновку, що найпоширенішим рішенням є використання синхронного виклику хмарних функцій як сервісів, у випадку AWS - Lambdas. Накопичений досвід експлуатації веб-служб Amazon (AWS) дозволяє зробити висновок про те, що багато організацій-користувачів AWS спочатку використовували AWS Lambda (продукт Amazon's Function-as-a-Service) для автоматизації робочих процесів навколо запущеного та масштабованого коду. Хоча таке рішення дозволило розробникам простих програм заощаджувати час, з підвищенням складності програм у компаній, які використовували AWS Lambda, виникали проблеми [3].

Ці проблеми об'єднувалися у дві основні групи: збільшення тривалості роботи та управління станом. Зокрема, під час виконання front-end сервісів AWS Lambda дозволяє виконувати функції не довше 15 хвилин. Після цього сервіс закінчується, залишаючи завдання незавершеними [4].

Очевидно, цей недолік є критичним для організацій, які використовують складні програми, що вимагають тривалих обчислень. Крім того, наслідком є неможливість повторного

виконання незавершених задач, оскільки функція, що виконувалася, заново викликатися не може. Кожен новий запуск цієї функції сприймається як окремий новий запуск із своїм особистим станом.

Для виходу з подібної ситуації можливо скористатися контейнерами за допомогою Amazon Elastic Container Service (ECS). Але у випадку, коли контейнери зайняті іншими завданнями, треба додати стан.

Ще один недолік полягає в тому, що AWS Lambda не створена для ефективного управління станами. Розробники повинні писати код у своїх програмах для зберігання цього самого стану, що ускладнює управління та збільшує час обробки. Отже, користувачі змушені вибирати між запуском вимогливих програм та забезпеченням достатнього стану для максимального використання [5].

З огляду на описану вище проблему, метою даної статті є дослідження, чи є AWS Step Functions самостійним вирішенням питання контексту та довготривалості виконання та чи варто використовувати цей підхід з точки зору ціноутворення.

3. Використання Step Functions архітектури системи з послідовним виконанням функцій

Розглянемо, що є AWS Step Functions. AWS Step Functions - це послуга, що надається Amazon Web Services, яка спрощує організацію декількох служб AWS для виконання завдань. Step Functions дозволяють створювати кроки в процесі, де результат одного кроку стає входом для іншого кроку, і це все робиться за допомогою візуального редактора робочого процесу.

Step Functions забезпечують досить зручну функціональність: автоматичну обробку повторних спроб, активацію та відстеження для кожного кроку робочого процесу та забезпечення виконання кроків у правильному порядку. Хоча спочатку цей список може здатися не вражаючим, виявляється, зовсім не тривіально забезпечувати, щоб усі ці речі відбувалися правильно в робочих процесах, що містять десятки кроків і сотні паралельних виконань. Step Functions виконує багато функцій, які потребує робити програма. Іншими словами Step Functions є автоматом стану, його основні абстракції називаються станами. Конфігурація функцій кроку є картою всіх можливих кроків та переходів між ними. Стани та їх переходи визначаються з використанням мови штатів Амазонки. Ця мова базується на JSON і є власністю Amazon [6].

Крокові функції AWS є невід'ємною частиною безсерверної екосистеми завдяки наступному. Створюючи безсерверні додатки, розробники, як правило, хочуть масштабувати програму відповідно до зростаючого навантаження, зберігаючи при цьому низькі витрати і дозволяючи декільком командам одночасно працювати над різними частинами програми. У безсерверній моделі однією з найкращих практик для досягнення цих цілей є розділення бізнес-логіки програми на набір нерозв'язаних служб. Велика безсерверна програма може складатися з десятків або сотень безсерверних сервісів [7]. Проблема виникає, коли така велика кількість служб потребує доступу до різних частин спільного стану. Для ефективної роботи цих служб команди також повинні мати можливість організувати потік даних через усі сервіси додатків в одному місці, і саме цим займаються AWS Step Functions. Вони стали найважливішим елементом безсерверної екосистеми впродовж усього стану системи і управління даними, необхідними для ефективної роботи безсерверних систем у масштабі.

4. Переваги AWS Step Functions

По-перше, Step Functions «полегшують життя» безсерверного розробника, дозволяючи швидко створювати складні послідовності задач в AWS, одночасно беручи на себе функцію обробки помилок і повторної спроби виконання логіки та дозволяючи відокремити бізнес-логіку додатка від логіки оркестрації. Нижче буде розглянуто особливості використання цієї технології, що може допомогти під час дизайну та розробки системи.

По-друге, завдяки цьому сервісу надається можливість швидкого створення складних послідовностей задач. Організація послідовності з десяти окремих безсерверних додатків, управління їх повторними спробами та усунення будь-яких помилок може бути надзвичайно складним завданням. Коли ви додаєте ще більше функцій, складність управління ними зростає в геометричній прогресії.

По-третє, завдяки графічному інтерфейсу та вбудованому механізму оперативного контролю Step Functions управляє послідовністю задач і знімає великий оперативний тягар з команди розробників.

Крім того є можливість керування станом між виконанням різних безсерверних та безстанових функцій. Для багатьох безсерверних робочих процесів налаштування черг та баз даних для зв'язку між усіма безсерверними службами може зайняти багато часу та спричинити помилки, не кажучи вже про те, що не всі випадки використання вимагають реальної бази даних або реальної системи черг.

Step Functions дозволяють також легко налаштувати управління станом на ранніх етапах, і це продовжує працювати добре з приростом масштабу завдань, коли використовуються багато сервісів разом.

Із використанням цієї технології логіка робочого процесу додатків стає від'ємною від бізнес-логіки. Тут є ще одна гарна практика безсерверної розробки. Додавання логіки робочого процесу та складних серверних конфігурацій до додатків, які повинні обробляти лише бізнес-логіку, збільшує складність програм і може легко створювати проблеми. Крім того, управління станом окремо від бізнес-логіки дозволяє розробникам зберігати ясність при роботі в безсерверній системі.

До плюсів можна також віднести ефективніші робочі процеси з паралельним виконанням. Коли є необхідність самостійно керувати станом різних функцій, отримати високу продуктивність від системи може бути складно. Деякі оркестровані фрагменти можуть обробляти лише одну задачу за раз, що може сповільнити роботу всієї програми. Step Functions дозволяють одночасно виконувати багато паралельних робочих процесів, переконавшись, що продуктивність масштабується залежно від завантаження програми [8].

Пропонується розглянути дві однакові системи, де у базі даних будуть 100 000 записів, запускатимуться декілька послідовних функцій, кожна з яких виконуватиме операцію фільтрації за своїм параметром. Різниця у тому, що в першій системі кожна функція викликати наступну за протоколами Amazon, без використання Step Functions (щільно зв'язані функції), а друга система буде запускати Step Functions, де функції будуть незалежними. Виконаємо 10 запитів та порівняємо результати (рис. 1).

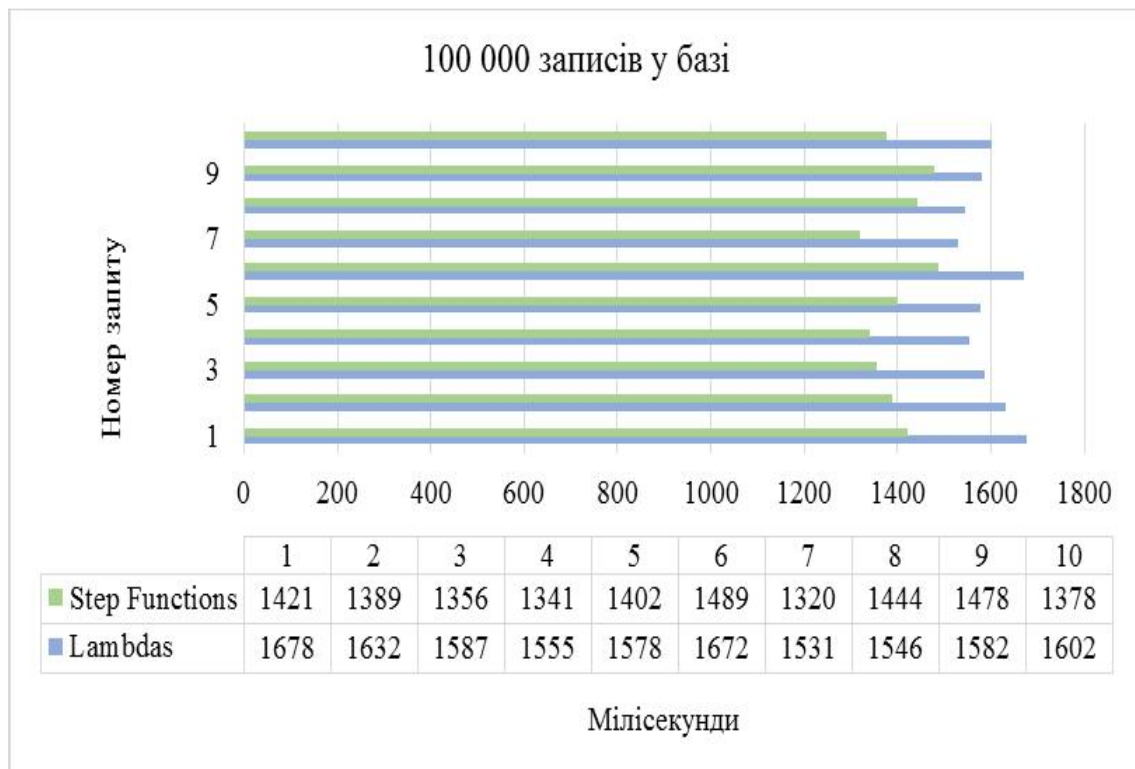


Рис 1. Порівняння підходів з/без використання Step Functions

З рисунку можна зробити висновок, що продуктивність системи з використанням Step Functions виявилась кращою за продуктивність системи без використання цього сервісу.

Функції кроку AWS можуть підвищити продуктивність роботи в кількох напрямках.

Перший напрям - обробка даних. Інші види обробки даних також добре підходять для функцій кроку AWS, особливо якщо джерело та адреса знаходяться всередині AWS. Функції кроку спрощують взаємозв'язок декількох кроків обробки даних, включаючи ті, що зроблені за допомогою функцій AWS Lambda, простої служби черги та простої служби сповіщень.

Другий напрям - організація робочого процесу без серверів. Основна цінність крокових функцій AWS полягає в простій організації програм, які вимагають взаємозв'язку множини безсерверних функцій. Для сукупності бізнес-процесів, для отримання кінцевого результату яких потрібна комбінація декількох роз'єднаних безсерверних програм, функція Step може бути правильним вибором для зручності організації.

5. Недоліки AWS Step Functions

Усі зазначені вище переваги можуть забезпечити високу цінність для багатьох команд та випадків використання, але перед тим, як взяти участь у запуску функцій Step у виробництві, слід також врахувати потенційні недоліки використання цієї програми у додатку. Хоча AWS Step Functions вирішують багато проблем для безсерверної спільноти, перехід усіх шарів оркестрації на Step Functions не обов'язково є правильним рішенням. Наступне зауваження треба враховувати при використанні покрокових функцій у виробництві.

Конфігурація з мовою станів Amazon є досить складною; його синтаксис заснований на JSON і тому оптимізований для машинної читабельності замість читабельності людьми. Вивчення мови може бути складним завданням, і, оскільки воно є власністю Amazon, набуті навички не застосовуватимуться ніде, крім контексту AWS Step Functions [9]. Ця мова є власністю Amazon і може використовуватися лише на Amazon Web Services. Тому при зміні постачальника хмарних послуг потрібно повторно впроваджувати рівень оркестровки або замінити його альтернативою іншого постачальника.

6. Механізм ціноутворення AWS Step Functions

Механізм ціноутворення є додатковою причиною переконатися у необхідності використання Step Functions в архітектурі системи, що має ланцюг послідовно виконуваних функцій.

Розглянемо основні характеристики цього механізму. Безкоштовний рівень AWS обмежений 4000 переходів стану функцій кроку AWS на місяць. Цей показник щомісячно поновлюється, тому можливо скористатися безкоштовним рівнем, навіть якщо обліковий запис AWS не є абсолютно новим.

За межею безкоштовного рівня використання Step Functions коштує 0,025 доларів США за 1000 переходів стану. Розглянемо роботу механізму ціноутворення в цьому випадку на такому прикладі. Нехай є робочий процес, який перетворює зображення для веб-сайту у чотири різних розміри, причому робочий процес складається із десяти переходів стану в середньому для кожного зображення і близько 10% виконання робочого циклу передбачає одну повторну спробу. Припустимо, що завантажуються 100 000 зображень на місяць. Кількість переходів робочого процесу буде складати 1 000 000 (10 переходів x 100 000 оброблюваних зображень). Додатково повторні спроби обробки дадуть 100 000 переходів (((100 000 оброблюваних зображень/100) x 10) x 10) переходів). Усього за місяць робочий процес потребує 1 100 000 переходів, 4 000 з яких будуть безкоштовними. У цьому випадку вартість використання функцій кроку AWS буде дорівнювати 27,4 \$ ((1 100 000 - 4 000) x 0,025 / 1 000).

Плата за AWS Step Functions доповнює будь-які збори за передачу даних та вартість будь-яких послуг AWS, якими ви користуєтесь у робочих процесах. Наприклад, якщо ці 100 000 виконання робочого процесу призводять до плати за AWS Lambda 600 \$ на місяць та до плати за передачу даних 100 \$ на місяць, загальна вартість за місяць складе 600 \$ + 100 \$ + 27,4 \$ (плата за використання крокових функцій) = 727,4 \$ на місяць для всієї системи.

7. Аналіз конкурентоспроможності AWS Step Functions

Розглянемо конкурентоспроможність AWS Step Functions відносно альтернатив від інших хмарних постачальників. Насамперед, слід зазначити, що Step Functions не мають прямого конкурента серед лідерів хмарних провайдерів таких, як Google (Google Cloud Platform, GCP) або Microsoft (Azure). Найближчим за походженням можна взяти Google Cloud Dataflows (GCP - Google Cloud Platform)

Розглянемо основні відмінності AWS Step Functions та Google Cloud Dataflow.

AWS Step Functions - це програмне забезпечення для створення розподілених додатків за допомогою візуальних робочих процесів. Ця технологія спрощує координацію компонентів розподілених програм та мікропослуг за допомогою візуальних робочих процесів. Створення програм з окремих компонентів, кожна з яких виконує дискретні функції, дозволяє швидко масштабувати та змінювати програми.

Google Cloud Dataflow є повністю керованою хмарною послугою та моделлю програмування для пакетної та потокової обробки великих даних. Google Cloud Dataflow також може використовуватися як керований сервіс для розробки та виконання широкого спектру шаблонів обробки даних, включаючи пакетні обчислення та безперервні обчислення. Cloud Dataflow забезпечує автоматичне управління ресурсами та оптимізацію продуктивності.

AWS Step Functions належить до категорії «Хмарне управління задачами» у технологічному стеці, тоді як Google Cloud Data Flow можна класифікувати як продукт, що належить до категорії «Обробка даних у реальному часі».

SWOT-таблицю аналізу цих продуктів наведено у табл. 1.

Таблиця 1 -
SWOT AWS Step Functions порівняно з конкурентами (GCP Dataflows)

Ключові фактори	Краще за конкурентів	Гірше за конкурентів
Графічне відображення виконання алгоритму програми	✓	
Конфігурування послідовних або паралельних кроків алгоритму програми	✓	
Контроль переходу стану між різними сервісами	✓	
Легкість конфігурування		✓
Швидка обробка великих об'ємів даних		✓
Рівень популярності серед розробників	✓	
Рівень ціноутворення	✓	
Рівень популярності бренду		✓
Підтримка інтеграції з різними сервісами того ж самого хмарного провайдера	✓	

Переваги використання саме AWS Step Functions стають очевидними, тому що наразі немає конкурентного сервісу від інших постачальників, що може графічно відображувати інтеграції з різними сервісами, контролювати стан переходу між ними та їх вхідні параметри.

Стосовно найближчого конкурента, GCP Dataflows, слід зазначити: це сервіс, створений більше для моніторингу обробки великого набору даних, автоматичного збільшення пам'яті для їх обробки, але не для конфігурації послідовного чи паралельного виконання різних сервісів Google.

За іншими критеріями - популярність у всьому світу та легкість конфігурування - AWS Step Functions виявилася гіршою, ніж система Google. Щодо популярності, звісно, Амазон не може сперечатися з Google, але не у світі розробників, де AWS існує вже давно та досі є безперечним лідером з постачання сервісів для хмарних обчислень. Щодо легкості конфігурування, AWS Step Functions конфігурується спеціальною мовою, заснованою на JSON шаблоні, тому розробникам потрібно буде спочатку навчитися її використовувати.

Слід також пам'ятати про ціноутворення, а саме, про те, що AWS надає 4000 безкоштовних виконань цього сервісу на місяць. Це означає, що якщо система буде використовуватися нечасто, то кошти не потрібно буде сплачувати взагалі. Також слід зазначити, що неважливо, скільки сервісів було викликано під час одного виконання автомату стану Амазону, це все буде вважатися одним виконанням Step Functions, і кошти підуть окремо за сервіси та окремо за одне виконання Step Functions.

8. Висновки

У даній роботі було розглянуто проблему використання безсерверних функцій, а саме, відсутність стану використання системи, відсутність механізму повторних запитів до ланцюга виконуваних послідовно функцій у випадку некоректної праці будь-якого елемента, та сильну зв'язаність між усіма послідовними функціями у вигляді прямого запиту кожної з них до наступної.

Також було досліджено особливості AWS Step Functions як окремого сервісу та шляхи вирішення усіх проблем архітектури системи з послідовним виконанням безсерверних функцій. Для цього пропонується знову звернути увагу на визначення Step Function та на проблему, що розглядалася.

Step Functions слід належним чином описувати як «стан як послуга». Без нього неможливо було б підтримувати стан кожного виконання, що має множину лямбда-функцій/дій.

Технологія мікросервісів передбачає, що кожен сервіс - це незалежна частина системи, яка може бути навіть написана іншою мовою програмування. Тому коли потрібне послідовне виконання декількох функцій як сервісів, у випадку AWS - Lambdas, то впровадження Step Functions - необхідний крок для завдання порядку виконання, який залишає сервіси незалежними. Без цієї технології доведеться на кожному кроку, у кожній функції задавати ім'я наступної, тобто сервіси будуть знати кожен про кожного, що означає - тісне зв'язування і порушення концепту мікросервісної архітектури.

Як один з недоліків Step Functions було визначено, що їх доволі складно конфігурувати, тому що потрібно знати нову мову AWS - Amazon State Language. Складно також уявити конфігурацію функцій, що послідовно викликаються та знають одна про одну, як вже було сказано, для того, щоб виключити або просто змінити один з кроків, щоб після 3-го кроку, наприклад, викликався 6-й, а потім 4-й. Функція, викликаючи наступну у своєму ж коді, тісно зв'язана із цією наступною функцією. Тому зміна порядку викликів або виключення одного з етапів виконання призведе до зміни коду майже кожної функції окремо у той час, коли, опанувавши ASL та сконфігурувавши Step Functions, можна з легкістю маніпулювати порядком або робити щось інше, не торкаючись окремих функцій.

AWS Step Functions вирішують питання контексту між усіма послідовними функціями, що, безперечно, є їхнім плюсом, адже концепт Lambda як раз говорить, що це функція безконтекстна та швидка. Тому вигадування логіки передачі його без Step Functions суперечить самому принципу безсерверних функцій та робить логіку виконання та підтримки складнішою та повільнішою. Як говорить один з принципів розробки SOLID, літера S - Single Responsibility, що означає «єдиний обов'язок». Тобто, якщо лямбди не вміють працювати з контекстом і за своїм принципом не повинні, - не треба їх змушувати, нехай про це подбає той сервіс, що був створений для цього, - AWS Step Functions.

І нарешті ціноутворення. Як вже було сказано вище, Step Functions - не дуже дорогий сервіс. Якщо компанія дозволяє собі систему, яка побудована на мікросервісній архітектурі і використовує AWS, то у випадку послідовного виконання функцій необхідно замислитись над використанням технології автомату стану Амазона. Адже, навіть якщо сто або більше функцій виконується послідовно, коли вони сконфігуровані у Step Functions - це все одно лише одне виконання автомату, а тисяча таких виконань коштує лише 0,025 \$.

Тому можна дійти до висновку, що саме цей сервіс є невід'ємною складовою мікросервісної архітектури системи із послідовним виконанням функцій. Step Functions - це відносно новий продукт AWS, який, безсумнівно, змінить показники продуктивності, дозволивши розбити програми на основні компоненти служби з можливістю маніпулювати кожним із цих компонентів окремо.

Список літератури: 1. *Ньюмен С.* Создание микросервисов. СПб.: Пітер, 2018. 304 с. 2. *AWS Step Functions: Developer Guide.* Режим доступу: <https://www.amazon.com/AWS-Step-Functions-Developer-Guide-ebook/dp/B078XBSLY53> 3. *Fowler M.* Microservices. Режим доступу: <https://martinfowler.com/articles/microservices.html> 4. *Witting A., Witting M.* Amazon Web Services in Action. Shelter Island, 2015. 200p. 5. *Офіційний сайт C++ Micro Services.* - Режим доступу: <http://cppmicroservices.org/> 6. *Boricha V, Rajani M., Atamna A.* Learn AWS Serverless Computing. Packt Publishing, 2019. 174 p. 7. *Richardson, C.* Pattern: Microservice Architecture // Kong. - Access mode: <http://microservices.io/patterns/microservices.html> 8. *AWS Step Functions.* Режим доступу: <https://aws.amazon.com/ru/step-functions/?step-functions.sort-by=item.additionalFields.postDateTime&step-functions.sort-order=desc> 9. *Офіційний сайт Google:* Compare AWS and Azure services to Google Cloud. Режим доступу: <https://cloud.google.com/free/docs/aws-azure-gcp-service-comparison>

Надійшла до редколегії 06.05.2021

Ситнікова Поліна Едуардівна, кандидат технічних наук, доцент, доцент кафедри системотехніки ХНУРЕ. Наукові інтереси: Data Mining and Knowledge Discovery, логічне моделювання даних, прийняття рішень в умовах невизначеності. Адреса: Україна, 61166, м. Харків, пр. Науки, 14, тел. (057) 702 10 06; e-mail: polina.sytnikova@nure.ua.

Говдерчак Олексій Павлович, студент групи СПРМ-19-2 ХНУРЕ, магістрант. Наукові інтереси: розробка програмного забезпечення, розробка архітектури систем. Адреса: Україна, м. Харків, 61166, пр. Науки, 14, тел. (057) 702 14 46; e-mail: oleksii.hovderchak@nure.ua.

УДК 004.048:004.89

DOI: 10.30837/0135-1710.2021.177.035

А.П. САФОНИК, І.М. ГРИЦЮК, М.М. МІЩАНЧУК, І.В. ІЛЬКІВ

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВИЗНАЧЕННЯ ЗАЛІЗА В КОАГУЛЯНТІ НА ОСНОВІ НЕЙРОННОЇ МЕРЕЖІ

Розглядаються питання побудови інтелектуальної системи для визначення концентрації заліза в коагулянті за його кольором на базі нейронної мережі. На основі проведеного аналізу різних типів нейронних мереж обрано найбільш відповідну архітектуру нейронної мережі для вирішення задачі визначення концентрації заліза в коагулянті. Описано процес проектування архітектури, дослідження методів навчання, підготовки даних для проведення навчання нейронної мережі для визначення концентрації заліза в коагулянті за його кольором. Розроблено структурно-функціональну схему нейронної мережі, яка складається із вхідного, прихованих та вихідного шарів, описано активаційні функції. Проаналізовано точність навчання нейронної мережі шляхом порівняння даних, отриманих з використанням різних оптимізаторів, із використанням бібліотеки TensorFlow. Розроблений веб-додаток може бути використаний в якості складової інформаційно-аналітичної системи автоматизованого керування технологічним процесом електрокоагуляційної очистки.

1. Вступ

Очищення стічних вод - це одна з проблем сучасного людства. Останнім часом для очищення води набувають популярності невеликі очисні споруди. Оснащення водоочисних споруд такими системами особливо важливе для сільського господарства, легкої промисловості тощо. Але використання реагентів зумовлює створення служб забезпечення їх доставки та зберігання, що є накладним в порівнянні з електрохімічними методами, які, в свою чергу, дають змогу добувати реагенти на місці з наявної сировини. Одним із найбільш перспективних методів, які надають таку можливість, є метод електрокоагуляції. На даний час все більше уваги приділяється дослідженням цього процесу за допомогою математичних моделей, що допомагає поліпшити конструктивні особливості пристрою та зменшити витрати в експлуатації, а також прогнозувати ефективність процесу в широкому діапазоні операцій з