УДК 004.021

А. М. Сало, В. В. Загорняк
Національний університет «Львівська політехніка»,
кафедра електронних обчислювальних машин

ЗАСОБИ ОПТИМІЗАЦІЇ МОДУЛЯ АДМІНІСТРУВАННЯ ВЕНДИНГОВИХ КІБЕРФІЗИЧНИХ СИСТЕМ

Розглянуто особливості роботи вендингових кіберфізичних систем після тривалої експлуатації. Проаналізовано вже готові способи масштабування кіберфізичних систем. Визначено основні недоліки та переваги переходу вже готової системи на платформу стороннього розробника. Запропоновано способи оптимізації баз даних із великою кількістю інформації. Наведено аналітичні дані, які привели до рішення оптимізувати наявну базу даних. Показано елементи вендингової кіберфізичної системи та можливі зміни в її структурі при переході на інші платформи. Описано функції модулів процесінгу та аналітичної системи. Запропоновано два способи для оптимізації роботи модуля адміністрування. Доведено ефективність запропонованих рішень за допомогою аналітичних даних.

Ключові слова: вендинг, кіберфізична система, вендингові кіберфізичні системи, бази даних, оптимізація, реплікація, процесінг.

A. Salo, V. Zahorniak
Lviv Polytechnic National University,
Computer Engineering Department

MEANS OF VENDING CYBER PHYSICAL SYSTEM ADMINISTRATION MODULE OPTIMIZATION

Features of vending cyber physical systems performance after durable exploitation have been examined. Ready-made means of cyber physical systems scaling have been analyzed. Main advantages and disadvantages of a ready system switchover to a third-party developer's platform have been defined. Means of large databases optimization have been offered. Analytical data that led to the decision to optimize the existing database have been produced. The elements of vending cyber physical system and possible changes in its structure due to the switchover to different platforms have been covered. Processing modules and analytical system functions have been described. Two ways of administration module optimization have been offered. The efficiency of the proposed solutions has been proved with analytical data.

Key words: vending, cyber physical system, vending cyber physical systems, database, optimization, replication, processing.

Вступ

Вендинг – це продаж товарів та послуг за допомогою автоматизованих систем. Автоматизованими системами в цьому випадку є торгові автомати. Вендингові автомати можуть виконувати безліч функцій, зокрема продавати напої (охолоджені напої, гарячі напої, молоко, пиво, тощо), їжу (снеки, цукерки, заморожені продукти), квитки, оплачувати парковку, працювати із готівкою (термінали, банкомати). Приклад реалізації вендингового автомату для продажу очищеної питної води наведено в [6].

Кіберфізичні системи – це інтеграція обчислень, комунікаційних мереж та фізичних процесів [5]. Важливо розуміти, що мережу вендингових автоматів не можна вважати кіберфізичною системою, якщо в ній відсутні моніторинг, керування та інші функції, які властиві таким системам [2].

Вендингова кіберфізична система (ВКФС) забезпечує постійний моніторинг та керування системою адміністратором. Взаємодія користувача із ВКФС може відбуватися безпосередньо з автоматом (екран, кнопки), мобільними додатками (сканування QR-кодів з автоматів та оплата через телефон), веб-інтерфейсом. Вендинговий автомат має бути з'єднаний з інтернетом для передавання статистичних даних, даних про транзакції (оплата послуг користувачем). Залежно від автомату, з'єднання з інтернетом може бути постійним або періодичним (передавання даних заданим інтервалом). Постійне збирання статистичних даних помагає своєчасно обслуговувати автомати та визначати економічну вигоду від конкретного автомата. На відміну від ВКФС, автономні вендингові автомати не мають таких переваг [1].

1. Об'єкт дослідження

Об'єктом дослідження є вендингова кіберфізична система. Вона складається з плати керування вединговим автоматом, модема та програмного модуля, який містить процесінг, аналітичну систему та засоби персонального сервісу, генератора конфігурацій для плати (див. рис. 1).

До функцій аналітичної системи входять:

– контроль за своєчасним обслуговуванням автомата (заміна деталей, інкасація, поповнення запасів товару тощо)

– екстрені ситуації, які потребують негайного втручання;

– контроль за економічною вигодою від автомата.

Загалом функції моніторингу дають змогу запобігати критичним ситуаціям, визначати оптимальну схему роботи автомата.

До функцій керування входять:

– оновлення ПЗ автомата;

– увімкнення/вимкнення окремих функцій автомата;

– аудит роботи автомата.

Вищезгадані функції допомагають підтримувати нормальну роботу ВКФС. Але з часом мережа вендингових автоматів може зростати, разом із кількістю даних від автоматів. Якщо журнал роботи автомата з часом можна видаляти, то журнал транзакцій платежів має зберігатися постійно. Це спричиняє поступове сповільнення роботи всієї системи, а в гіршому випадку – вихід системи з ладу.

Сповільнення роботи спричинене блокуванням БД під час читання даних. Це відома проблема багатьох реляційних баз даних, які мають великі розміри.

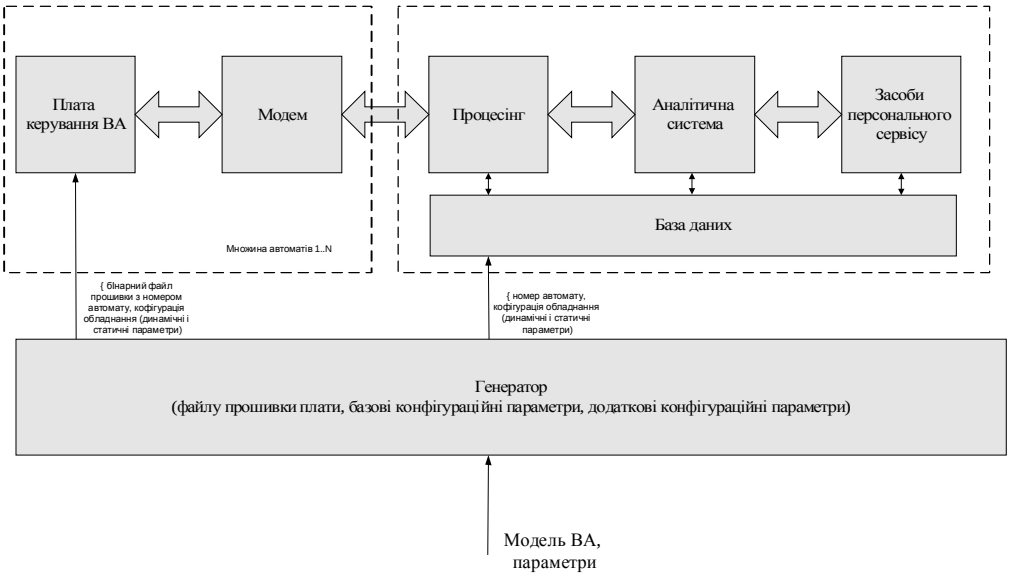

Рис. 1. Структура вендингової кіберфізичної системи

Основною причиною оптимізації стали блокування БД під час читання великих обсягів даних. Тести, наведені в табл. 1, проводили з інтервалом 5 хвилин.

Таблиця 1

Статистика обробки запитів до БД

SQL запит, с	Обробка PHP, с	Запит HTTP, с	Пам'ять, МБ	Кількість рядків із БД
41,2	8,65	49,85	212,95	35610
1,29	2,86	4,15	212,95	35610
1,29	2,89	4,18	212,95	35610
42,79	4,11	46,9	195,2	31843
1,05	2,59	3,64	195,2	31843
1,05	2,75	3,8	195,2	31843

Із наведеної вище таблиці можна побачити затримки в 41,2 та 42,79 секунди, які виникають час від часу незалежно від кількості даних, які отримують з БД. Ці аналітичні дані стали поштовхом для оптимізації бази даних у ВКФС, оскільки затримок в інших складових системи немає.

2. Мета та задачі досліджень

Метою роботи є визначення способу оптимізації бази даних (БД) ВКФС для підтримання стабільної роботи.

Для виконання поставленої мети потрібно виконати такі завдання:

– визначити якнайкращий спосіб оптимізації БД;
– мінімізувати витрати на оптимізацію;
– мінімізувати час простою ВКФС під час оптимізації БД.

3. Дослідження існуючих рішень

Розглянемо такі можливі варіанти оптимізації системи керування базами даних: перехід на інший тип баз даних, реорганізація існуючої бази даних та програмного забезпечення модуля адміністрування ВКФС, масштабування існуючої архітектури БД, використання готових рішень для IoT.

3.1. Використання нереляційних баз даних

NoSQL – база даних, яка забезпечує механізм зберігання та видобування даних відмінний від підходу реляційних баз даних. Перевагою таких баз даних є значно більша швидкість отримання даних, значно простіше горизонтальне масштабування на кластери машин, контроль доступності. Мінусами такого рішення є те, що після змін у базі оновлені дані не завжди можна отримати відразу або ж вони будуть неточними. Не у всіх реалізаціях NoSQL можна використовувати транзакції та використовувати join. Для використання аналогу SQL join у NoSQL існує декілька підходів:

– множинні запити: розбиття складних запитів на підмножину простіших запитів. Таке рішення у NoSQL базах даних є оптимальним, тому що сумарний час виконання цих кількох запитів все одно буде швидшим, ніж один складний запит у SQL;

– кешування, реплікація, ненормалізовані дані – зберігання зовнішніх значень разом з основними даними. Такий підхід є вдалим у разі частого читання даних, але надзвичайно ускладнює зміну даних у БД;

– вкладені дані: у документних БД можна зберігати велику кількість даних у меншій кількості колекцій. Такий підхід використовують у MongoDB.

MongoDB – документо-орієнтована СКБД з відкритим вихідним кодом. Вона не потребує опису схеми таблиць. Основні можливості:

– документо-орієнтоване сховище (JSON-подібна схема даних);
– динамічні запити;
– журналювання всіх операцій;
– ефективне зберігання великих бінарних файлів;
– асинхронна реплікація.

За можливості БД можна розгорнути на двох серверах, використовуючи реплікацію master-slave. При виході з ладу однієї БД інша продовжує працювати автономно.

3.2. Оптимізація існуючої БД

Оптимізація існуючої БД може значно покращити роботу системи. Такий підхід є доволі затратним, оскільки для оптимізації вже існуючої архітектури знадобляться послуги адміністратора баз даних та розробника. Така оптимізація передбачає:

– оптимізацію та зміну SQL-запитів із подальшим тестуванням;
– розроблення та зміну існуючої архітектури БД;
– перенесення вже існуючого масиву даних на нову архітектуру.

Найбільш затратним за такого підходу буде міграція на нову архітектуру із збереженням усіх даних. Для цього розробникам потрібно розробити міграції, які автоматично перенесуть всі дані із існуючої БД до нової.

Такий метод обмежений ефективністю апаратних засобів на яких працює БД, оскільки навантаження на кіберфізичну платформу постійно зростає; такий підхід тільки тимчасово дасть змогу зменшити навантаження на БД та пришвидшити роботу кіберфізичної системи.

Також можна використовувати постійний кеш для пришвидшення завантаження даних. Постійний кеш генерується тільки один раз при першому запиті до даних. Після вибірки та обчислень дані зберігаються до збірної таблиці в БД. Такий підхід дає змогу значно зменшити час доступу до даних, але потребує більшого дискового простору.

3.3. Використання іншої системи керування базами даних

Якщо в проекті використано SQL базу даних, то можна спробувати використати іншу систему керування базами даних (СКДБ). Із використанням MySQL є можливість повністю перейти на рушій MariaDB, який має відкритий вихідний код та розповсюджується під ліцензією GNU GPL. MariaDB практично повністю підтримує сумісність із MySQL, оскільки початково він розроблявся як відгалуження від MySQL.

Такий підхід допоможе оцінити зміну продуктивності бази даних без великих економічних та часових витрат. Враховуючи те, що MariaDB використовує нове сховище даних Aria, яке замінило MyISAM, але має повну зворотну сумісність з ним, можна точно сказати, що запис даних у БД відбуватиметься швидше. Нове сховище PBXT підтримує мультиверсійний метод організації даних, цим самим дозволяючи позбавитися від блокування під час читання даних.

MariaDB Galera – кластерна СКБД, яка підтримує синхронну реплікацію multi-master. Під час синхронної реплікації всі вузли завжди містять актуальні дані. Транзакції фіксуються тільки після поширення даних всіма вузлами. Збійні вузли системи вилучаються із системи та можуть бути знову запущені вручну. Особливістю цієї СКБД є можливість масштабування і читання, тоді як решта рішень пропонують тільки масштабування читання (рис. 2).

PostgreSQL – об'єктно-реляційна система керування базами даних. На відміну від інших СКБД не контролюється якоюсь однією компанією. Особливістю є можливість одночасно модифікувати дані декільком користувачам. Завдяки цьому відпадає потреба в блокуванні під час зчитування даних.

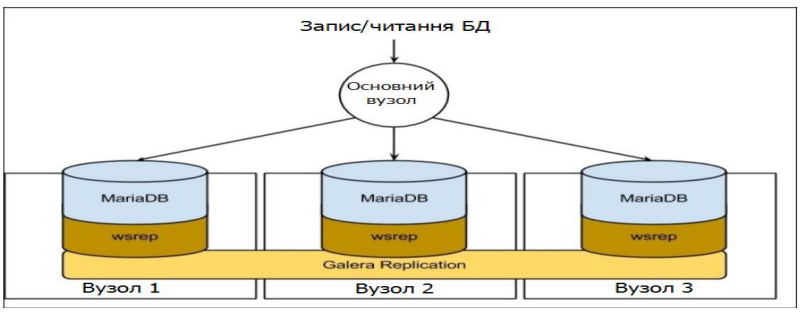

Рис. 2. Приклад конфігурації MariaDB Galera кластера

При переході на рушій БД без зворотної сумісності виникає багато проблем. Для коректної роботи БД на новому рушії потрібно буде витратити багато часу. Якщо перенос даних, можливо, не викликатиме багато проблем, то налаштування всіх таблиць, тригерів, зв'язків і т. д. може значно сповільнити перехід. Також потрібно буде провести тестування нової БД, щоб визначати, чи коректно все працює.

3.4. Масштабування існуючої архітектури БД

Масштабування архітектури можна поділити на два типи: вертикальне і горизонтальне. Вертикальне масштабування передбачає покращення обчислювальних характеристик та модернізацію наявного обладнання. Таке масштабування простіше виконати, але з часом такий підхід втрачатиме ефективність, оскільки кількість даних у БД зростатиме, і знову виникатимуть проблеми, як раніше.

Горизонтальне масштабування реалізувати складніше: воно відбувається шляхом реплікації наявної БД на інші машини (рис. 3). Сучасні засоби контролю хмарних застосунків (Google, Amazon) забезпечують реплікацію БД на велику кількість машин. У такому випадку підключення відбувається до головного сервера, який розподіляє навантаження. Наприклад, операції вибірки з БД виконуються на одній машині, а операції запису даних – на іншій. Такий підхід є дуже ефективним, оскільки він оптимально розподіляє навантаження на БД та гарантує цілісність даних та їх однорідність на всіх машинах.

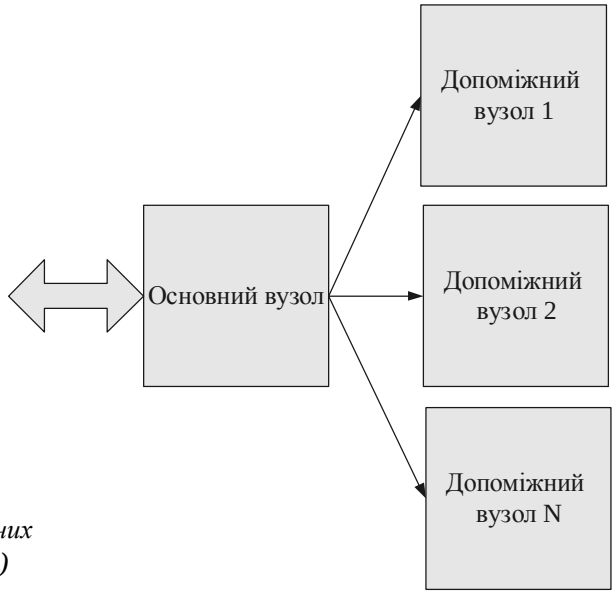

Рис. 3. Приклад реплікації бази даних (горизонтальне масштабування)

За такого підходу до оптимізації можна не користуватися послугами розробників та адміністраторів баз даних, оскільки потрібно буде змінити тільки кінцеву адресу БД для підключення. Знадобиться тільки системний адміністратор, якому потрібно налаштувати нову платформу для розгортання існуючої бази даних.

Горизонтальне масштабування також має свої недоліки, які переважно виражаються вартістю реплікації БД.

3.5. Використання готових рішень для IoT

Найефективнішим підходом можна вважати використання вже готових рішень, розроблених провідними компаніями. Одним з таких сервісів є Amazon Web Services для IoT. Ця платформа є однією з найпопулярніших у світі та може забезпечити безперебійну роботу декількох мільйонів пристроїв одночасно. Завдяки своїм обчислювальним можливостям та можливості створювати фактично безмежну кількість реплікацій БД ця система є найстабільнішою та найефективнішою для використання. Єдиним недоліком цієї системи є її вартість. Тому невеликим компаніям та розробникам не вигідно використовувати вже готове рішення [4].

Стабільність роботи системи забезпечується виділеними серверами, на яких знаходиться копія БД. Завдяки рушію, який було розроблено в компанії, гарантія однорідності даних на всіх екземплярах БД забезпечується майже із 100% точністю. У разі виникнення конфліктів беруться дані, які збігаються у більшості БД. Отже, щодо цілісності даних можна не перейматися.

Основним вузлом системи є AWS IoT Core, який займається балансуванням навантаження на систему. Він підтримує протоколи HTTP, WebSockets і спрощений протокол зв'язку MQTT. Цей вузол спроектовано так, щоб працювати в умовах нестабільного з'єднання та максимально зменшувати навантаження на мережу стисканням даних.

Підключають пристрої за допомогою платформи AWS IoT Device Management. Ця платформа дає можливість вносити до свого реєстра атрибути пристроїв (назва, тип, серія, рік виробництва), що спрощує групування та пошук пристрою. Ця платформа також слідкує за станом пристроїв та повідомляє про вихід компонентів системи із ладу.

AWS IoT DM дає можливість віддалено оновлювати ПЗ пристроїв, встановлювати виправлення безпеки тощо вже після встановлення їх до системи. Така автоматизація значно полегшує роботу з IoT системою, якщо вона складається з кількох сотень або тисяч пристроїв.

Існують обмеження для віддаленого оновлення ПЗ пристроїв. Основне обмеження – це вимога використовувати спеціальну операційну систему на мікроконтролері Amazon FreeRTOS, яка є розробкою Amazon і підтримує повну інтеграцію з сервісами компанії. Сервіс Amazon IoT Greengrass Core дозволяє створити завдання для оновлення ПЗ пристроїв по «повітрю». Під час наступного підключення пристрою до платформи на нього завантажиться оновлена прошивка, і він оновиться.

Загалом платформи для робот з IoT пристроями дуже сильно відрізняються. Але якщо брати тільки основний функціонал (ядро) кожної платформи, то різниці практично немє. Всі платформи дозволяють оновлювати ПЗ пристроїв, всі підтримують основні протоколи передавання даних: HTTP для періодичного передавання інформації, WebSockets для постійного з'єднання. Також кожна платформа має свої протоколи передавання даних власного розроблення. Перехід на нову платформу змушує переробити весь програмний модуль платформи. Процесінг та аналітику виконуватимуть засобами Amazon. Для цього потрібно повністю переписати ці модулі задіючи розробників. Процесінг повністю перейде під контроль Amazon IoT Core, а аналітичну частину виконуватиме платформа Amazon Analytics. Такі зміни в архітектурі платформи займуть багато часу та ресурсів (рис. 4).

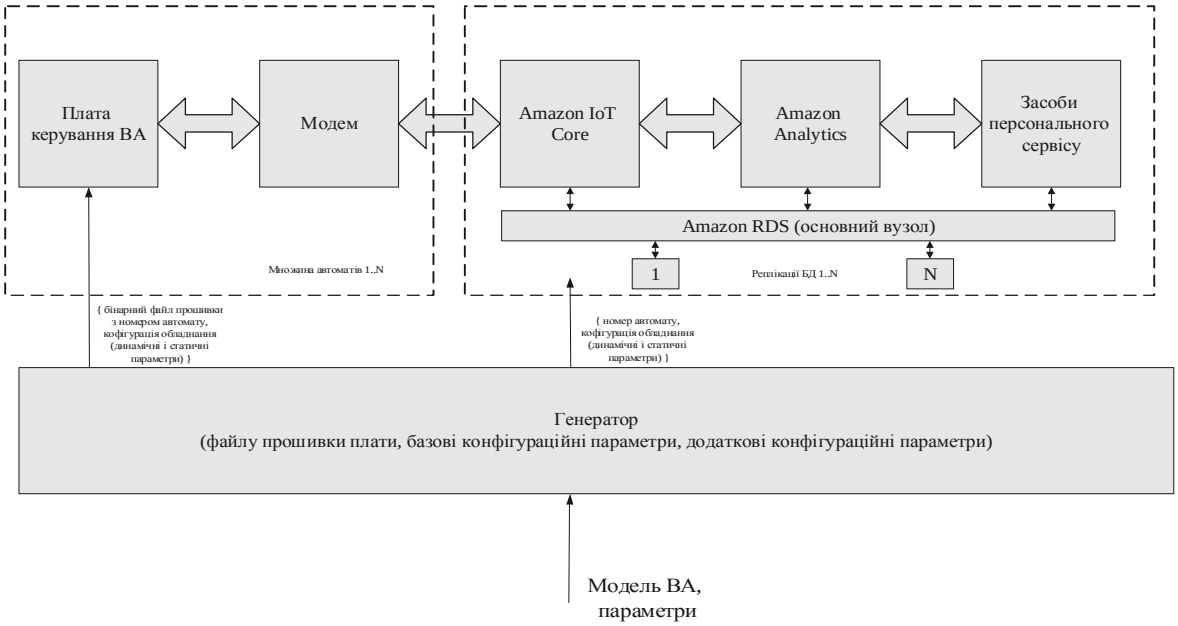

Рис. 4. Структура вендингової кіберфізичної системи після переходу на платформу Amazon IoT

База даних мігрує на сервіс Amazon RDS, а засоби персонального сервісу залишаться без змін.

3.6. Міграція існуючої архітектури БД на готове рішення

Основні вимоги до готового рішення полягають у постійній доступності БД та зменшенні часу обробки запитів вибірки. Також потрібно запобігти блокуванню БД під час читання.

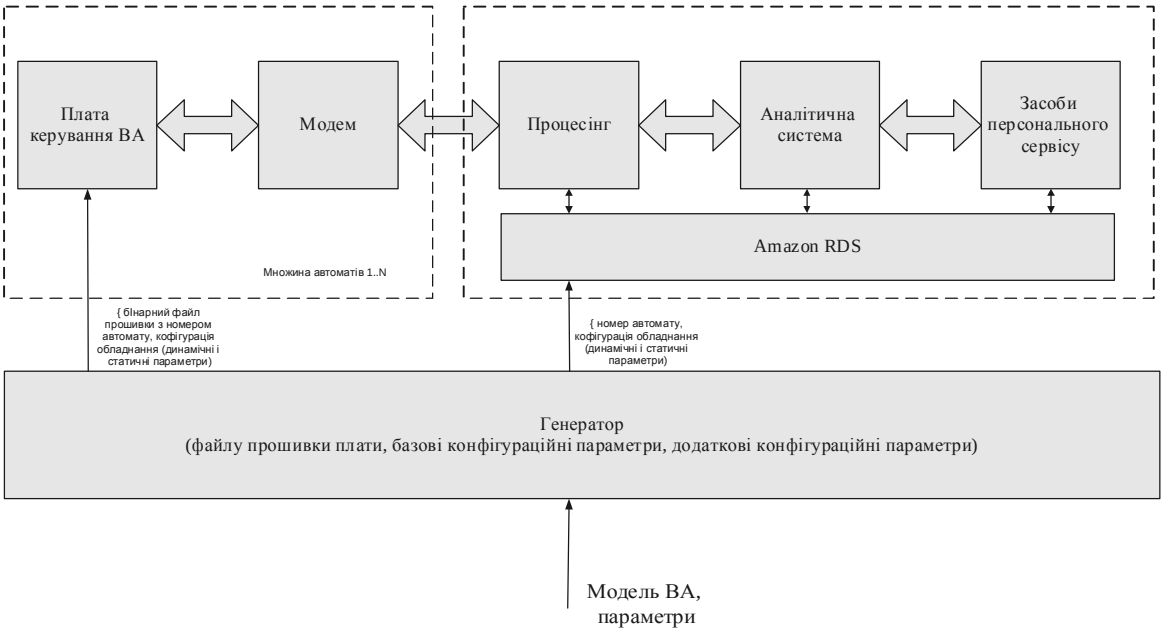

Рис. 5. Структура вендингової кіберфізичної системи після міграції на Amazon RDS

Можна використати готове рішення Amazon RDS. Для налаштування постійної доступності в Amazon RDS існує опція Multi A-Z. Вона дозволяє розмістити копії БД у різних регіонах. Це запобігатиме блокуванню під час читання та за допомогою горизонтального масштабування паралельно виконувати читання з БД без видимих затримок.

Для переходу на Amazon RDS потрібно тільки перенести існуючу БД на нову платформу (рис. 5). Для цього використовується сервіс AWS Database Migration Service. Його можна використовувати для однорідних міграцій типу MySQL до MySQL або неоднорідних типу Oracle в Amazon Aurora. Також цей сервіс можна використовувати для безперервної реплікацій даних з основної БД. Під час реплікації та міграції час простою основної БД зведена до мінімуму. Для початку роботи з новою платформою потрібно тільки змінити кінцевий шлях підключення до БД та в налаштуваннях безпеки відкрити доступ до програмної частини ВКФС.

4. Вирішення проблеми

Виходячи із поставленої задачі та аналізу вже існуючих рішень, можна виділити два основні способи вирішення проблеми, які можна комбінувати: горизонтальне масштабування, використання іншого рушія БД разом із горизонтальним масштабуванням або використання готової платформи для розгортання БД, яка вже підтримує горизонтальне та вертикальне масштабування.

Інші способи нас не влаштовують через такі причини:

– перехід на нереляційну БД потребує великих часових та фінансових затрат; важко передбачити продуктивність системи, поки не буде повністю розгорнуто БД; вихідний код всього проекту необхідно переписати для використання нових запитів до БД (складні SQL запити потрібно переписувати в масив простіших NoSQL запитів);

– використання іншого рушія не дає гарантій значного прискорення роботи БД; є висока імовірність, що перехід на новий рушій може тільки сповільнити роботу;

– вертикальне масштабування може бути ефективним тільки певний час, поки БД знову не збільшиться до критичного розміру; також такий підхід не вирішує блокування БД під час читання або запису даних;

– використання готового рішення для масштабування спонукає встановлювати стороннє ПЗ від розробника, реєструвати всі пристрої в системі, налаштовувати оновлення. Такий підхід рівносильний побудові системи з нуля, з тією різницею, що пристрої вже фізично встановлено в систему. Також під час переходу на повністю готове рішення може виявитися, що наше апаратне забезпечення (контролери, давачі) не підтримується цією системою. Також є обмеження в протоколах передавання даних.

В сукупності ці фактори свідчать про те, що готова платформа є якнайкращим рішенням для розроблення системи з нуля, яка обслуговуватиме величезну кількість пристроїв, але ніяк не про те, що перехід готової системи на зовсім іншу платформу виявиться нам вигідним.

Найкращим рішенням є перехід на готову платформу для розгортання БД, яка використовуватиме ту саму СКБД, що й наявна база даних. Готова платформа дозволяє без встановлення стороннього ПЗ використовувати горизонтальне масштабування. Також для забезпечення ще швидшого доступу до використовуваної інформації можна задіяти власний кеш: дані, які хоча б раз запитувалися і є незмінними, в майбутньому (сума транзакції, журнал подій тощо) записуються в постійний кеш БД, який знаходиться в проміжній таблиці. Час життя такого кешу необмежений та гарантує швидку доступність до вже раніше запитуваної інформації.

Міграція на нову платформу дозволила зменшити час виконання SQL-запитів та запобігти блокуванню БД під час читання (див. табл. 2). Це означає, що блокування зникли через існуючі копії БД, які використовують для читання даних. Основний вузол розподіляє навантаження між вузлами таким чином, що запити йдуть до менш навантажених реплікацій для швидшої обробки запитів. З табл. 2 видно, що після першого запиту даних всі наступні запити тих самих даних виконуються за однаковий час, оскільки дані знаходять у кеші. Також зникли блокування БД під час читання даних, що спричинено наявністю реплік та постійного кешу.

Таблиця 2

Статистика обробки запитів до БД після міграції на Amazon RDS

SQL запит, с	Обробка PHP, с	Запит HTTP, с	Пам’ять, МБ	Кількість рядків із БД
1,7	7,56	9,26	212.95MB	35610
1,29	2,85	4,14	212.95MB	35610
1,29	2,89	4,18	212.95MB	35610
2,9	4,61	7,51	195.2MB	31843
1,05	2,59	3,64	195.2MB	31843
1,05	2,75	3,8	195.2MB	31843
1,31	0,71	2,02	56.51MB	7375
0,9	0,77	1,67	56.51MB	7375
1,83	0,665	2,50	56.52MB	7384
1,98	0,65	2,63	57.01MB	7695

Висновки

Наведено структуру модуля адміністрування вендингових кіберфізичних систем. Розглянуто проблематику швидкого росту кількості даних БД у ВКФС, яке значно сповільнює роботу системи або призводить до її некоректної роботи та відмов.

Вибрано оптимальне рішення, яке дозволить масштабувати БД горизонтально без великих часових та фінансових витрат. Наведено аналітичні дані, які підтверджують ефективність обраного способу.

Таке рішення можна вважати довговічним вирішенням проблеми, оскільки за потреби в майбутньому БД ВКФС можна продовжувати масштабувати в тому самому напрямі без простою системи. Відсутність простоїв у роботі забезпечує наявність реплікації, яка гарантує цілісність даних навіть у разі виходу з ладу одного або декількох вузлів БД.

1. Сало А. М. Принцип побудови вендінгової мережі з моніторингом // Вісник Нац. ун-ту “Львівська політехніка” “Комп'ютерні системи та мережі”. - 2013. - No 773, С. 112-118. 2. Lee E. A. and Seshia S. A.. Introduction to Embedded Systems – A Cyber-Physical Systems Approach, Second Edition, MIT Press, – 2017. – p. 9–16. 3. Главные тренды мирового рынка торговых автоматов [Електронний ресурс]: / Режим доступу: http://www.vendoved.ru/glavnye-trendy-mirovogo-rynka-torgovyh-avtomatov/. – Назва з екрана. 4. Приложения и решения для IoT [Електронний ресурс]: / Режим доступу: https://aws.amazon.com/ru/iot/. – Назва з екрана. 5. E. A. Lee and S. A. Seshia, Introduction to Embedded Systems: A Cyber-Physical Systems Approach, Second Edition. http://leeseshia.org, 2015. 6. A. Salo. Simulation of water purification machine for vending cyber physical systems. Technology audit and production reserves – No 2/2(40), 2018. – p. 16 – 21.