

ВИЗНАЧЕННЯ МЕТРИК ЯКОСТІ СЕРВІСУ У ГЕТЕРОГЕННІЙ РОЗПОДІЛЕНІЙ CLOUD-СИСТЕМІ

© Лаврів О. А., Бак Р. І., Пашкевич В. З., Климаш М. М., 2019

Досліджено процес навантажувального тестування розподіленої гетерогенної інформаційно-телекомунікаційної системи, яка базується на cloud інфраструктурі. За результатами навантажувального тестування виявлено вузьке місце у досліджуваній системі, а також отримано оцінки використання інфраструктурних компонентів системи і значень метрик якості обслуговування та якості сприйняття послуг користувачами досліджуваної інформаційно-телекомунікаційної системи. Показано, що оновлення програмного коду системи робить неможливим розроблення та використання універсальних моделей, оскільки навіть незначні зміни програмного коду призводять до зміни поведінки системи, що не може бути враховано як у аналітичній, так і у імітаційній моделях.

Ключові слова: розподілена гетерогенна інформаційно-телекомунікаційна система, якість надання послуг, навантажувальне тестування, cloud інфраструктура.

O. Lavriv, R. Bak, V. Pashkevych, M. Klymash
Lviv Polytechnic National University

SERVICE QUALITY METRICS CALCULATION FOR HETEROGENEOUS DISTRIBUTED CLOUD SYSTEM

© Lavriv O., Bak R., Pashkevych V., Klymash M., 2019

The rapid development of information technology today is inextricably connected with telecommunication systems that provide the transmission of information between components of distributed information services. Since the partition of services, which are not limited to the only computer system physically prevailing the partition of centralized services, the role of telecommunications is very important. Along with the client-server and peer-to-peer architecture of distributed information and telecommunication systems, Cloud-systems, Fog-systems appear, and the widespread of Internet technology leads to the fact that telecommunication systems have the function of supporting the universal information space. We performed the load testing of the distributed heterogeneous information and telecommunication system based on the cloud infrastructure. The results of load testing revealed a bottleneck in the system. We also simulated utilization of infrastructure components and metrics of service quality and quality of experience. It has been shown that updating of the system code makes it impossible to design and use universal models of the system, since even minor changes to the code which logically combine the components of the system leads to a change in the behavior of the system, which cannot be considered in both analytical and simulation models. In this regard, the proposed methodology for the investigation of distributed heterogeneous information and telecommunication systems based on cloud infrastructure is an important tool in their design and operation and allows solving the problem of harmonizing the levels of the proposed conceptual structure of information and telecommunication systems with improvement of service quality metrics and optimal distribution of resources of such systems.

Key words: distributed heterogeneous information and telecommunication system, load balancing, quality of service provision, load testing, cloud infrastructure, inter-level interaction.

Вступ

Сучасні системи надання інформаційно-телекомунікаційних послуг характеризуються суттєвим зміщенням своїх функцій у бік програмної реалізації, що зумовлено потребою клієнтів отримати адаптивну систему. Ця система має забезпечувати можливість оновлення як функціонального, так і презентаційного. Це означає, що програмна реалізація дає змогу досягати розширення функціоналу системи зміною програмного коду. Такі зміни можуть стосуватися як аспектів логіки, яка закладена в системі, взаємодії зі сховищами даних, базами даних, так і зміни тих аспектів, які стосуються способу подання системи кінцевому користувачеві.

Враховуючи вищевикладене, легко дійти висновку, що завдання моделювання поведінки такої системи вкрай складне, а подекуди розв'язку цього завдання не існує взагалі. Це пов'язано із тим, що тривалість розроблення адекватної моделі зазвичай перевищує тривалість існування конкретної версії системи, а будь-яка зміна у програмному кодї може призвести до змін у поведінці системи, які розроблювана модель просто не здатна врахувати (Жураковський, 2014).

Розуміння поведінки гетерогенної розподіленої інформаційно-телекомунікаційної системи на прикладному рівні є надзвичайно важливим, оскільки цей рівень забезпечує безпосередній контакт користувача із системою. Якщо розробити модель системи вкрай складно або й узагалі неможливо, то єдиним способом оцінювання її поведінки залишається експериментальне дослідження. Планування експерименту для дослідження поведінки складної інформаційно-телекомунікаційної системи є складним та нетривіальним завданням. Попри сам метод дослідження поведінки, окремої уваги потребують також компоненти системи, які підлягають дослідженню, із урахуванням того, що сама система повинна бути неперервно доступною для клієнтів та кінцевих користувачів (Жураковський, 2014; Лаврів, 2018).

Модель корпоративного клієнта системи надання інформаційно-комунікаційних послуг на базі cloud послуги “програмне забезпечення як сервіс”

Для проведення експериментального дослідження поведінки таких систем, здійснено розроблення моделі корпоративного клієнта, структурну схему якої наведено на рис. 1.

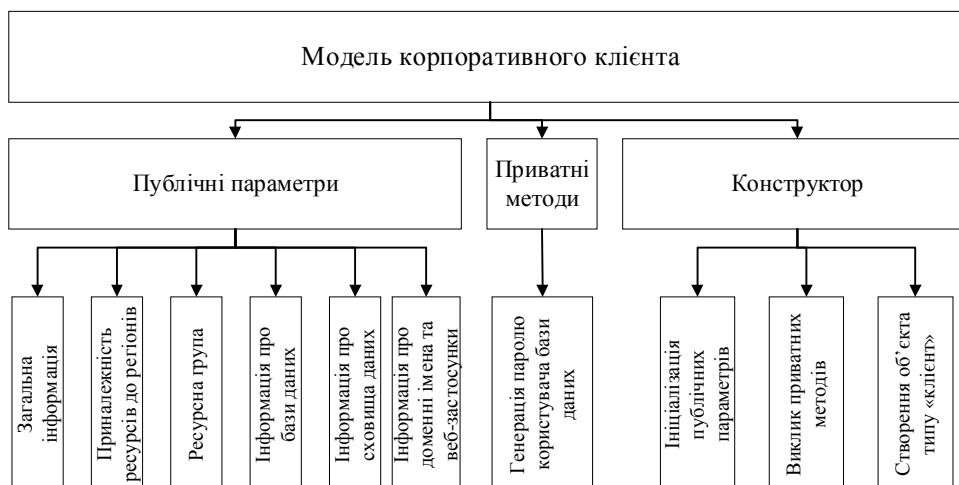


Рис. 1. Структурна схема моделі корпоративного клієнта розподіленої гетерогенної інформаційно-комунікаційної системи надання послуг

Модель являє собою набір властивостей та методів, які ініціалізуються викликом конструктора. Саму модель реалізовано у вигляді програмного коду із використанням мови програмування С#. Вона дає змогу відтворити структурно та функціонально поведінку реального корпоративного клієнта розподіленої інформаційно-комунікаційної системи. Оскільки модель є адаптивною, то вона забезпечує можливість змінної конфігурації ресурсів клієнтів, що, своєю чергою, дає можливість моделювати різні характери впливів різного роду користувачів на поведінку досліджуваної системи.

Назвемо цю модель програмною, оскільки вона дає змогу із використанням засобів мови програмування створювати віртуальні структурні образи реальних клієнтів інформаційно-комунікаційної системи. Модель базується на принципі агрегування функцій корисності (Толюпа та ін., 2014). Цю модель покладено в основу експериментального процесу навантажувального тестування. Модель використовує технологію імітаційного моделювання систем.

У результаті ініціалізації моделі набором вхідних параметрів формується програмний об'єкт, який володіє всіма властивостями програмної моделі із конкретною конфігурацією, яка відповідає типу корпоративного клієнта системи. Зокрема, йдеться про розміщення cloud ресурсів клієнта, наявність чи відсутність геореплікації даних клієнта, параметри бази даних та сховища даних клієнта, розміщення приватної інформації клієнта, яка забезпечує можливість решті компонентів розподіленої інформаційно-телекомунікаційної системи зі спільною інфраструктурою взаємодіяти із ресурсами клієнта. Важливою особливістю цієї моделі є те, що її використання забезпечує не лише ініціалізацію об'єкта, який характеризує клієнта, але й розгортання cloud ресурсів клієнта на базі cloud провайдера.

Так досягають можливості досліджувати поведінку складної розподіленої гетерогенної інформаційно-телекомунікаційної інфраструктури без використання ресурсів реальних клієнтів системи для створення навантаження на її компоненти. Це дає змогу уникнути небажаного завантаження ресурсів клієнта, а також убезпечує систему від загроз безпеці даних кінцевих користувачів.

Тестові клієнти, створені на підставі запропонованої моделі, використані у процесі навантажувального тестування з метою дослідження поведінки інформаційно-телекомунікаційної системи.

У результаті застосування об'єкта, що представляє клієнтські ресурси, до cloud інфраструктури, утворюється набір таких ресурсів: сервери баз даних відповідно до кількості регіонів, у яких розміщено дані користувача, відповідні репліки баз даних на цих серверах, відмовостійка група серверів баз даних, що забезпечує безперебійну роботу системи у разі недоступності ресурсів клієнта у первинному регіоні, сховище даних користувача, де зберігаються пов'язані із записами у базі даних артефакти (здебільшого це медіаконтент або ж аналітичні дані, які збирає система для подальшої оптимізації), набір пов'язаних із корпоративним клієнтом доменних імен спільних cloud ресурсів, здійснюється конфігурування приватних даних, які використовуються іншими спільних ресурсами для комунікації із призначеними ресурсами клієнта. Варто зазначити, що у випадку розміщення ресурсів клієнта тільки в одному регіоні група відмовостійкості не ініціалізується, що означає відсутність реплік бази даних та, відповідно, зменшення навантаження від такого клієнта на інформаційну та телекомунікаційну складові загальної системи.

Модель розроблена вперше, а її застосування дає змогу підвищити ефективність методу експериментального дослідження поведінки складної інформаційно-телекомунікаційної інфраструктури шляхом навантажувального тестування внаслідок врахування впливу на систему користувачів із різною конфігурацією ресурсів.

Модель спільної Cloud інфраструктури системи надання інформаційно-комунікаційних послуг

Для того, щоб врахувати у моделі корпоративного клієнта структурно-функціональні параметри спільної cloud інфраструктури, розроблено її програмну модель. Під програмною моделлю спільної cloud інфраструктури будемо розуміти програмний код, написаний мовою програмування C#, який дає змогу генерувати екземпляри спільних cloud ресурсів, інформація про які необхідна у процесі розгортання клієнтських ресурсів у розподілених гетерогенних інформаційно-телекомунікаційних системах. Оскільки інфраструктура базується на принципі мультиклієнтського використання (орендування), то деякі її елементи підлягають спільному використанню множиною корпоративних клієнтів. Кожен корпоративний клієнт, своєю чергою, містить набір кінцевих користувачів, тим самим реалізуючи сформульований раніше принцип агрегації функцій корисності. Тобто сумарна корисність послуги для всіх користувачів є корисністю послуги для клієнта.

Спільна cloud інфраструктура (рис. 2) являє собою набір таких компонентів: веб-застосунки, спільні системні бази даних, сховища приватних даних клієнтів, спільна зона сервісу доменних імен. Оскільки спільна інфраструктура реалізує принцип географічної надлишковості, то в моделі реалізовано також балансувальники трафіку, які відповідають за спрямування запитів користувачів до застосунків у відповідному регіоні за критерієм максимальної продуктивності (мінімального часу очікування відповіді від сервера застосунку).

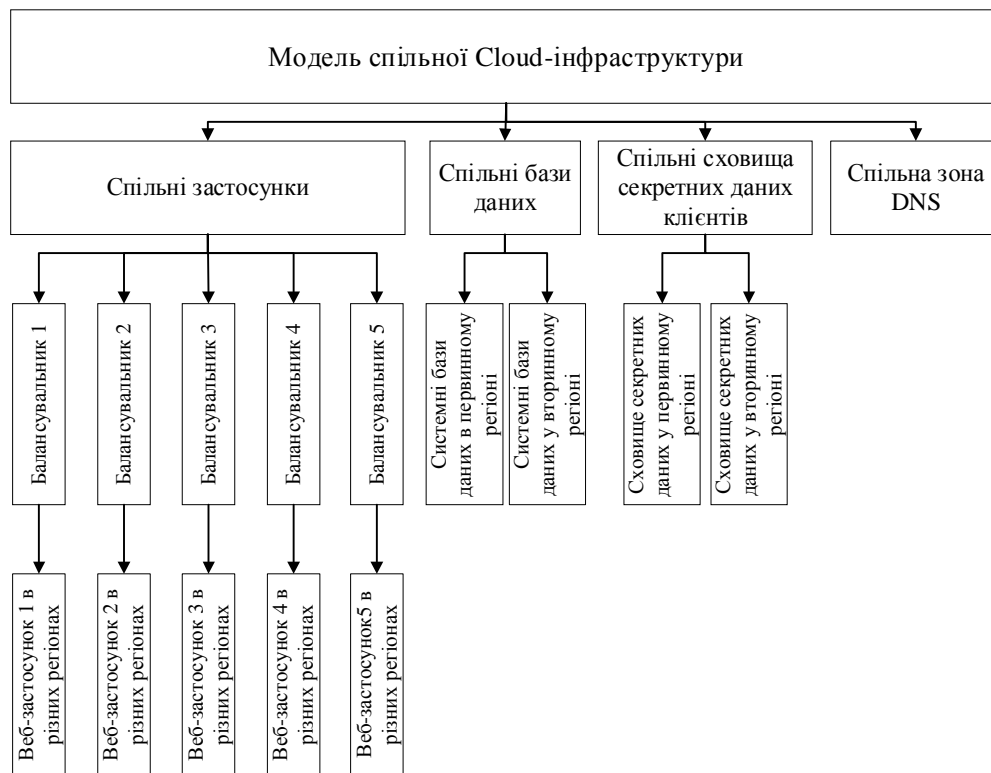


Рис. 2. Структурна схема моделі спільної інфраструктури розподіленої гетерогенної інформаційно-комунікаційної системи надання послуг

Подібно до програмної моделі корпоративного клієнта, програмна модель спільної інфраструктури ініціалізується набором вхідних параметрів. Результатом її ініціалізації є створення програмного об'єкта спільної інфраструктури, властивості якого використовують у процесі розгортання інфраструктури корпоративного клієнта у розподіленій гетерогенній інформаційно-телекомунікаційній системі. Запропонована програмна модель базується на технології імітаційного моделювання. Вона відрізняється від відомих здатністю адаптуватися під інфраструктурні зміни шляхом інтелектуального формування властивостей об'єкта, що представляє ресурси спільної cloud інфраструктури. Ця здатність дає змогу враховувати інфраструктурні зміни спільних ресурсів, як, наприклад, зміну кількості спільних веб-застосунків або перехід на іншу зону сервісу доменних імен у процесі створення ресурсів корпоративного клієнта, що дає змогу урізноманітнити та додатково деталізувати метод дослідження поведінки таких складних систем на основі навантажувального тестування (Жураковський, 2014).

Представлена модель спільної cloud інфраструктури базується на принципі функцій вартості (Лаврів, 2018). Йдеться про те, що спільна інфраструктура зазнає впливу множини клієнтів у процесі користування їхніми сервісами, що надає досліджувану розподілену інформаційно-комунікаційну систему. Функція вартості дає змогу оцінити ступінь впливу кожного окремого клієнта на зниження продуктивності спільної інфраструктури для нових клієнтів.

У сукупності описані вище дві моделі дають змогу узагальнити та описати процес комунікації між множиною користувачів та cloud ресурсів на рівні надання послуг. Не слід забувати, що при цьому особливої уваги заслуговують питання безпеки даних користувачів та катастрофостійкості розподілених інформаційно-телекомунікаційних систем надання послуг.

Навантажувальне тестування інформаційно-комунікаційної системи, побудованої на базі cloud сервісу «платформа як сервіс», та визначення її метрик якості сервісу

Завдання та цілі навантажувального тестування інформаційно-комунікаційних систем

У процесі побудови та надання хмарних застосунків клієнтам постачальнику дуже важливо розуміти, яку кількість одночасних користувачів може підтримувати система за умови її використання з урахуванням різних сценаріїв (Толюпа та ін., 2014). Враховуючи, що будь-яка зміна програмного коду або ж інфраструктури впливає на цю кількість, подекуди дуже важко або й неможливо коректно змоделювати поведінку такої системи. Тому отримання оцінки максимальної кількості користувачів проводять експериментальним шляхом на основі навантажувального тестування.

Серед основних завдань навантажувального тестування інформаційно-комунікаційних систем слід виділити такі:

- Встановлення максимальної кількості одночасних користувачів у системі за різних сценаріїв використання cloud застосунку.
- Виявлення вузьких місць у системі.
- Встановлення компонентів системи, які потребують вдосконалення або заміни з метою збільшення ємності системи.
- Виявлення нерегулярних відмов та помилок, які їх спричиняють.
- Визначення прийняттого рівня помилок, за якого систему можна передати у використання клієнтам.

Оскільки навантажувальне тестування за своєю природою є експериментальним процесом, то перед його початком слід спланувати цей експеримент, визначити ключові точки навантажувального тестування, тобто елементи системи, які підлягають перевірці, визначити тривалість, хід, параметри ітерацій та інші характеристики експерименту.

Окремо слід реалізувати програмно розроблений алгоритм експерименту, оскільки йдеться про навантажувальне тестування розподілених програмних систем.

Конфігурування параметрів системи у процесі навантажувального тестування

Реалізовано систему навантажувального тестування, яка орієнтована на роботу з інформаційно-комунікаційною системою корпоративного рівня, розрахованою на обслуговування не менше ніж 5 тисяч користувачів по всьому світу. Слід зазначити, що комплекс засобів навантажувального тестування є окремим програмним продуктом, який розробляють конкретно для тестування певної інформаційно-комунікаційної системи, оскільки сам набір тестів та умови їх використання безпосередньо залежать від кінцевої системи. Система навантажувального тестування лише дає змогу зімітувати одночасну роботу кінцевих користувачів за певними сценаріями.

Навантажувальне тестування завжди організовують перед передаванням інформаційно-комунікаційної системи у користування клієнтам. Цей процес гарантує, що у реальних умовах експлуатації системи її поведінка відповідатиме очікуваній, а якість сприйняття буде знаходитися у заданих межах незалежно від кількості одночасних користувачів до досягнення певного порогу, який і є ємністю системи. Варто зазначити, що для різних сценаріїв поведінки користувачів ємність системи може різнитися. Тому за ємність системи приймають найменше із отриманих значень.

Навантажувальне тестування дає змогу визначити показники, які можуть бути закладені в угоди про рівень сервісу (SLA), які укладаються між клієнтом та постачальником cloud послуги.

Конфігурувати систему в процесі навантажувального тестування слід з метою калібрування параметрів системи. Дуже часто таке калібрування дає змогу досягти планованої ємності системи без необхідності вносити зміни на рівні програмного коду. Проте, це можливо лише за умови незначних відхилень реальної поведінки системи від очікуваної.

Якщо ж ємність системи, отримана в результаті навантажувального тестування, суттєво відрізняється від планованої, то без змін програмного коду неможливо досягти необхідної ємності системи.

Метрики навантажувального тестування

У процесі навантажувального тестування одним із етапів є збирання метрик, які характеризують інфраструктуру cloud системи, а також метрик, які характеризують якість сприйняття користувача. До першої групи належать: завантаження апаратних ресурсів віртуальних машин, які забезпечують хостинг застосунків (процесора, оперативної пам'яті тощо), завантаження системних та користувацьких баз даних (за обсягом переданих на вхід/вихід даних, завантаженням процесора, оперативної пам'яті тощо), завантаження інших системних компонентів, наприклад, редіс кешу за кількістю активних з'єднань. До другої групи метрик належать тривалість відгуку сервера, тривалість завантаження веб-сторінки, кількість запитів, на які сервер відповів з помилкою, що спричинила відмову в обслуговуванні запиту.

Моніторинг метрик навантажувального тестування, які належать до першої групи, може здійснюватися через інтерфейс користувача, який надається постачальником cloud сервісу.

Моніторинг метрик навантажувального тестування, які належать до другої групи, здійснюється в середовищі запуску програмного продукту, який реалізує функціонал навантажувального тестування.

Ітеративний процес навантажувального тестування

Оскільки зміни у програмному коді впливають на результати навантажувального тестування, то цей процес має ітеративний характер. Ітерацією навантажувального тестування вважаємо кожен цикл виконання цього процесу після необхідного запланованого обсягу змін у програмному коді. На підставі двох послідовних ітерацій процесу навантажувального тестування формується порівняльний звіт, який в відсотковому співвідношенні дає змогу оцінити покращення / погіршення метрик другої групи.

Паралельно після збирання метрик першої групи слід здійснити порівняльний аналіз для отримання повної картини щодо зміни поведінки інформаційно-комунікаційної програмної cloud системи під впливом коливань користувацького навантаження.

На підставі отриманих порівняльних оцінок є можливість прийняти рішення про готовність продукту до передавання у користування реальними клієнтами, чи, навпаки, є потреба змінити програмний код з метою забезпечення відповідності планованій ємності системи із заданим рівнем помилок, що спричиняють відмову у процесі обробки запиту.

Слід зазначити, що програмний продукт, який реалізує сценарії навантажувального тестування, є лише інструментальним засобом, який сам по собі не дає повної об'єктивної оцінки поведінки системи за умови, що він не охоплює всіх можливих сценаріїв поведінки системи.

Результати навантажувального тестування інформаційно-комунікаційних систем

Параметри та характеристики навантажувального тестування можна подати як таблицю для кращого представлення.

Спільна cloud інфраструктура на платформі Microsoft Azure (в назвах сервіс планів "Standard" сервісний рівень, який надається на сервери, цифра після нього означає кількість серверів, а слова "Medium" або "Large" визначають рівень конфігурації):

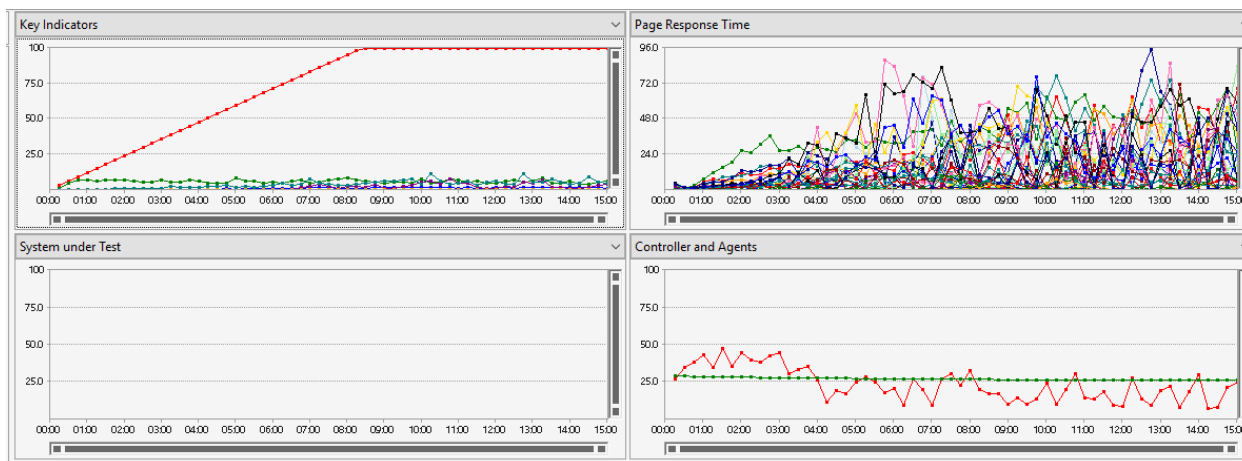
- Сервіс-план 1 Standard 1 Medium (сервер, на якому розміщено частину системних застосунків);
- Сервіс-план 2 Standard 4 Medium (сервер, на якому розміщено клієнтські веб-застосунки);
- Сервіс-план 3 Standard 8 Large (сервер, на якому розміщено веб-застосунок безпеки)
- системна база S0;
- база безпеки S2;
- база клієнта S3 (MaxPoolSize=1000);
- Редіс кеш C1 Standard.

Параметри процесу навантажувального тестування досліджуваної розподіленої інформаційно-телекомунікаційної системи

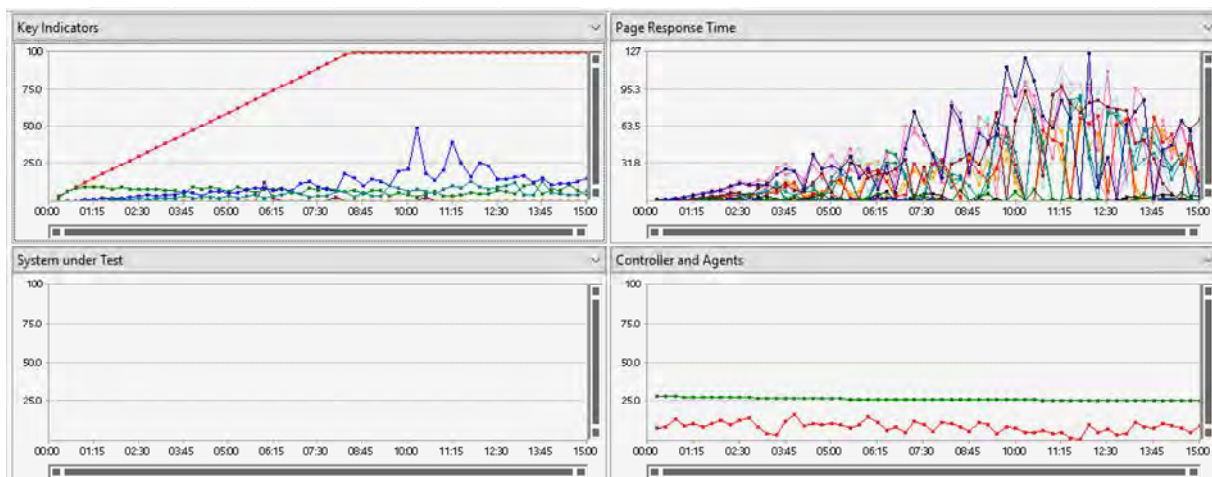
№ з/п	Максимальна кількість користувачів у системі	Тривалість процесу, хв	Початкова кількість віртуальних користувачів	Крок збільшення, с	Обсяг збільшення користувачів	Холодний старт	Середній час відгуку сервера, с	Максимальний час відгуку сервера, с	Масштабування системи	Збої зафіксовано на кількості користувачів	Статус експерименту
1	5000	15	10	5	10	Ні	3,66	11,7	вручну	5000	Невдача
2	5000	15	10	5	10	Ні	3,71	12,8	вручну	5000	Невдача

Проведено два експерименти, у яких змінювався лише програмний код самої досліджуваної системи. Конфігурація спільної інфраструктури, інфраструктури клієнта та програмної системи навантажувального тестування залишалася незмінною. Отримані результати демонструють, що дослідження поведінки такого класу систем неможливе із використанням аналітичних, чи навіть імітаційних моделей, оскільки немає змоги врахувати природу, складність та зміни програмного коду на всіх рівнях тривірневої архітектури cloud застосунку із резервуванням.

Нижче наведено порівняння результатів двох експериментів.



експеримент 1

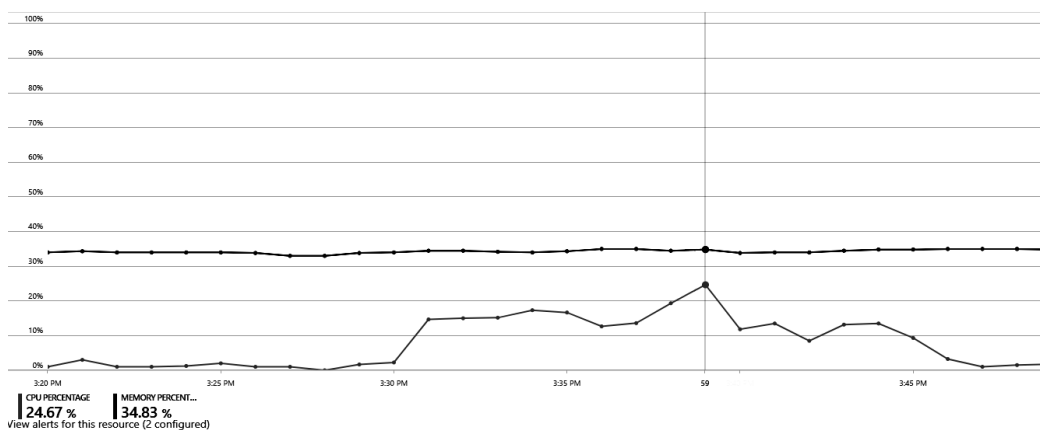


експеримент 2

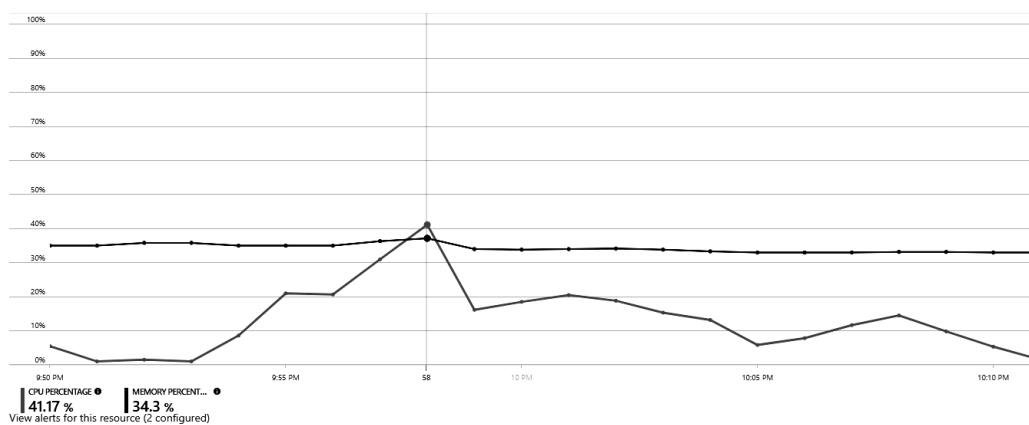
Рис. 3. Результати навантажувального тестування системи, яку реалізовано на основі моделей спільної cloud інфраструктури та корпоративного клієнта

Перша чверть рис. 3 ілюструє динаміку зростання кількості користувачів у системі (у відсотках від максимального значення), друга чверть ілюструє тривалість завантаження сторінок у веб-браузері клієнта при надсиланні запитів на різні компоненти інформаційно-комунікаційної системи, третя чверть відображає, що процес тестування завершено, а остання чверть ілюструє завантаження машини, з якої розпочинали процес навантажувального тестування. Варто зазначити, що загальне задеклароване навантаження створювалося з 5 незалежних машин, оскільки процес навантажувального тестування активно використовує обчислювальні ресурси у процесі формування значної кількості запитів на різні компоненти інформаційно-телекомунікаційної системи.

Нижче наведено рисунки, які демонструють реакцію інфраструктури досліджуваної інформаційно-телекомунікаційної системи на процес навантажувального тестування. З представлених графіків легко бачити періоди часу, в які система зазнавала зростання навантаження. На рис. 4 наведено ступінь використання ресурсів процесора та оперативної пам'яті Сервіс-плану 1. Бачимо, що завантаження знаходиться в допустимих межах (менших за 70 %) навіть за максимальної кількості користувачів, які одночасно використовують систему.



експеримент 1

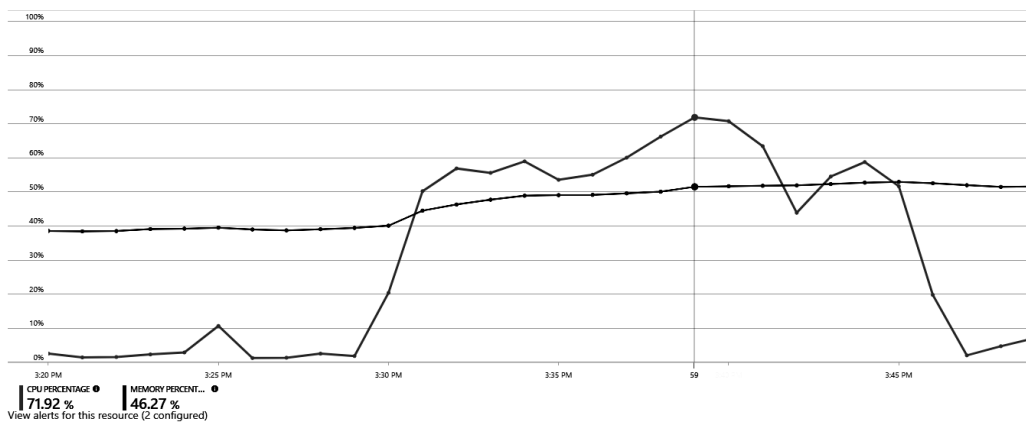


експеримент 2

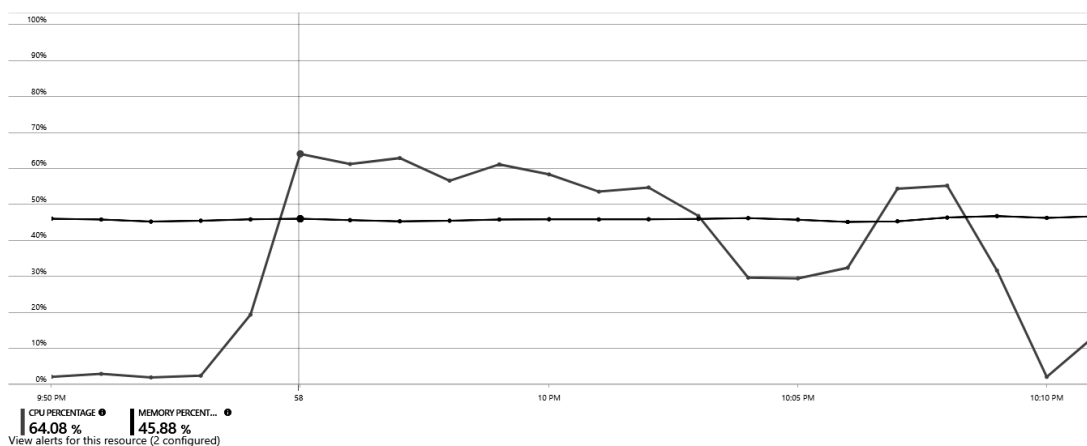
Рис. 4. Завантаження ресурсів Сервіс плану 1

Наступні графіки відображають поведінку Сервіс-плану 2 (рис. 5), який відповідає за автентифікацію користувачів в системі. Бачимо, що попри значний обсяг ресурсів, виділених для цього сервісу, їх завантаження перевищує 70 %, що має розглядатися як необхідність вдосконалення системи.

Наступні графіки відображають реакцію Сервіс-плану 3 на процес навантажувального тестування. Варто зазначити, що для цього компоненту попередньо підібрано так, щоб дотриматися допустимих показників завантаження ресурсів системи. На рис. 6 добре видно, що, хоча максимальне значення завантаження відрізняється і не суттєво, значно змінюється поведінка системи під навантаженням, що зумовлено змінами у програмному коді.



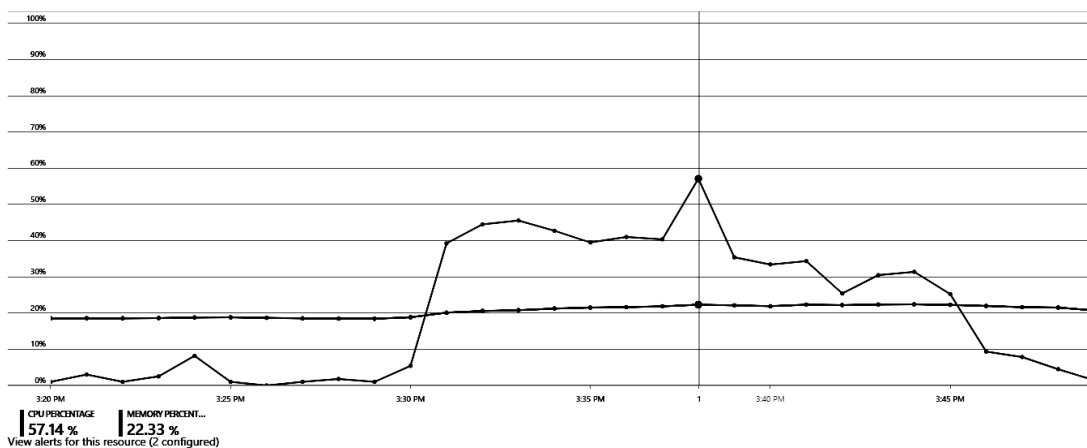
експеримент 1



експеримент 2

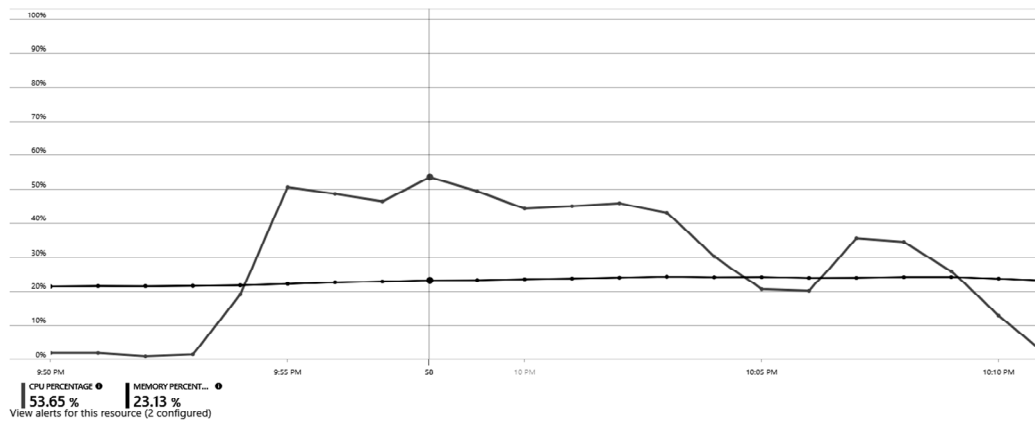
Рис. 5. Завантаження ресурсів Сервіс-плану 2

Реакцію системної бази не представлено, оскільки реалізовані сценарії навантажувального тестування не чинили фактично жодного навантаження на цю базу у зв'язку з тим, що при дослідженні прийнято, що всі користувачі належать до одного корпоративного клієнта, а звертання до цієї бази відбувається тільки у випадку часткої зміни клієнтів, користувачі яких створюють навантаження. Відповідно до моделі спільної інфраструктури ця база теж не є її компонентом.



експеримент 1

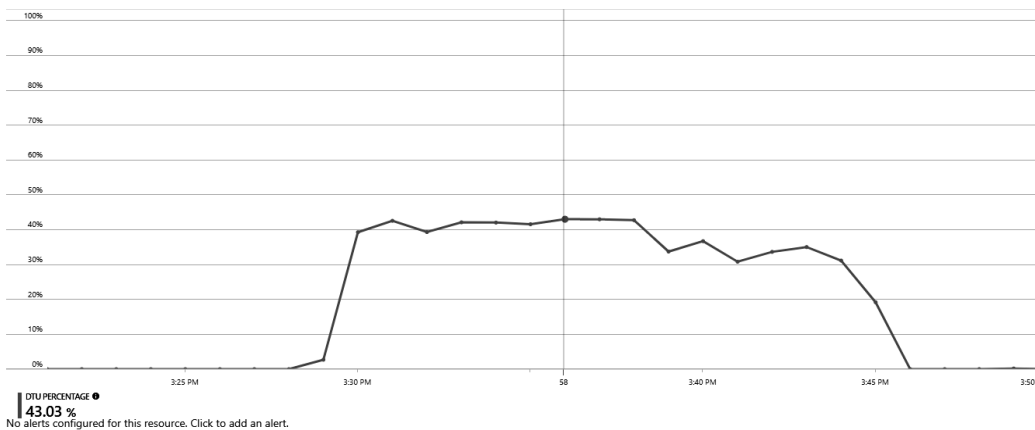
Рис. 6. Завантаження ресурсів Сервіс-плану 2



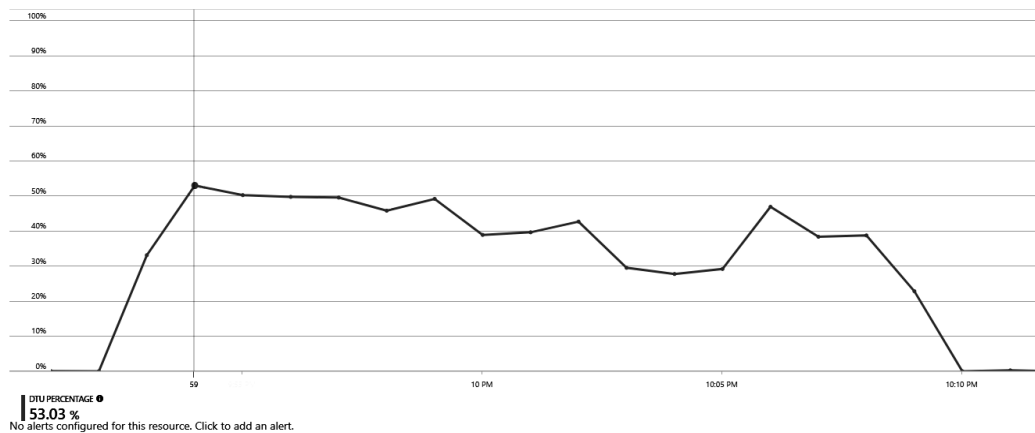
експеримент 2

Рис. 6 (продовження). Завантаження ресурсів Сервіс-плану 2

Реакцію бази даних безпеки представлено на рис. 7. Варто зазначити, що величина, яку відображено на графіку, є агрегованим показником, який називається переданими одиницями даних (Data Transfer Unit, DTU) і враховує використання процесора, оперативної пам'яті та інтерфейсів входу/виходу сервера бази даних. Ресурси баз даних не підлягають адаптивному масштабуванню, тому запас по DTU повинен бути не меншим 50%.



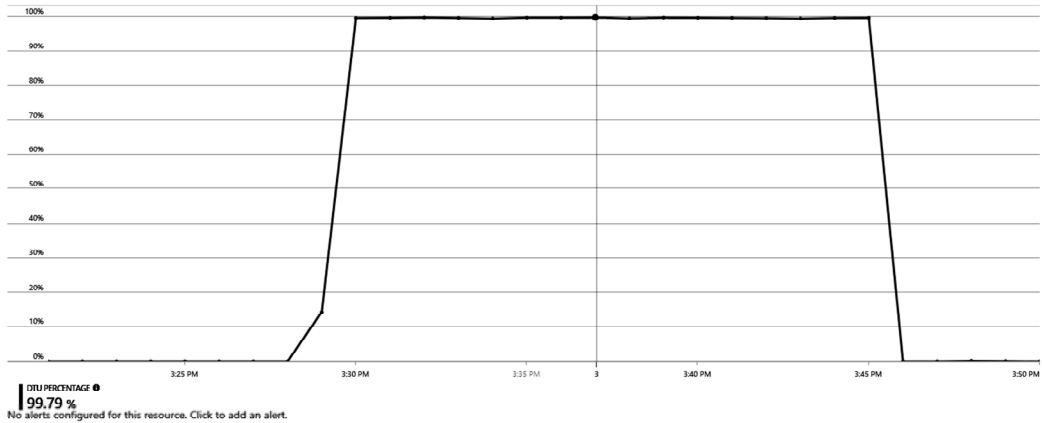
експеримент 1



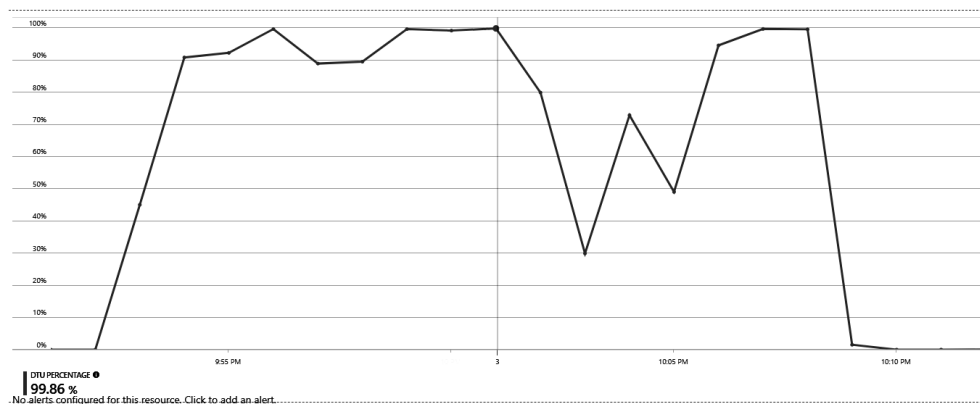
експеримент 2

Рис. 7. Завантаження бази даних безпеки у процесі навантажувального тестування

На рис. 8 представлено використання клієнтської бази. Результати свідчать, що у програмному кодї, який відповідає за взаємодію із цією базою даних, наявні суттєві дефекти, оскільки завантаження цієї бази даних протягом майже всього періоду проведення експерименту було близьким до 100%. Це означає, що вузьке місце системи, яке обмежує її подальше масштабування, виявлено, що є однією із цілей процесу навантажувального тестування. Можна також спостерігати зміну поведінки інфраструктури зі зміною програмного коду досліджуваної системи.

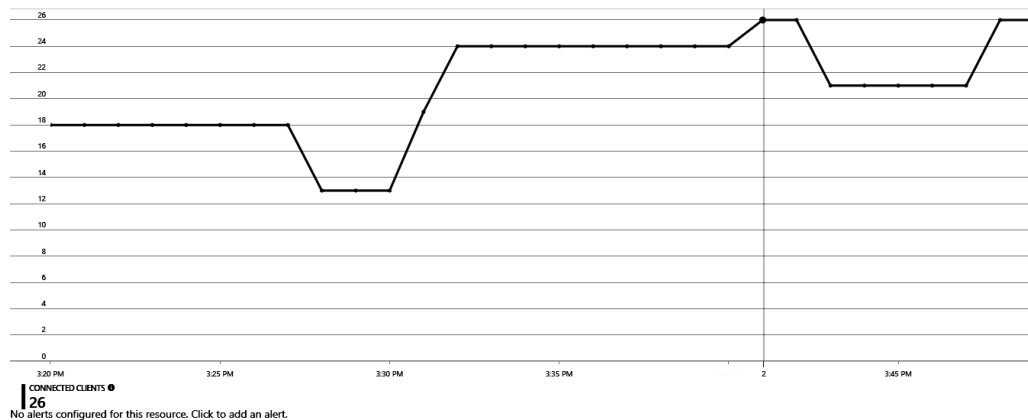


експеримент 1



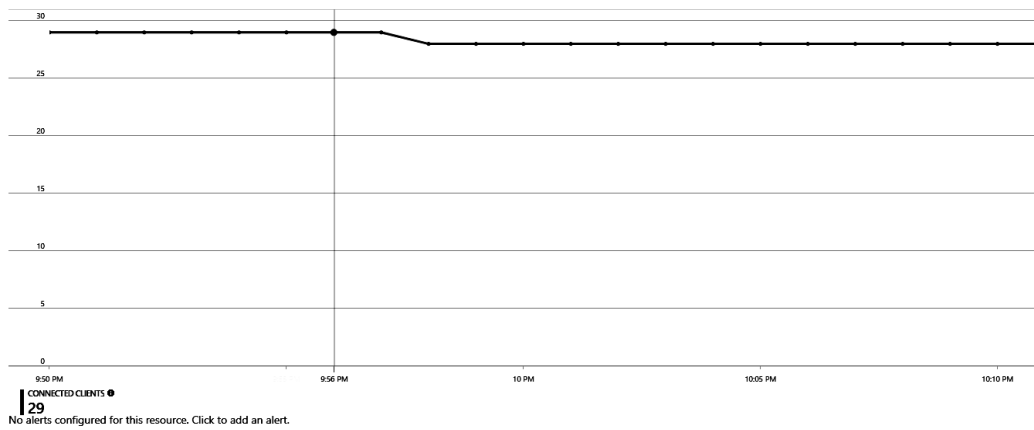
експеримент 2

Рис. 8. Завантаження бази даних корпоративного клієнта у процесі навантажувального тестування



експеримент 1

Рис. 9. Завантаження редіс кешу у процесі навантажувального тестування



експеримент 2

Рис. 9 (продовження). Завантаження редіс кешу у процесі навантажувального тестування

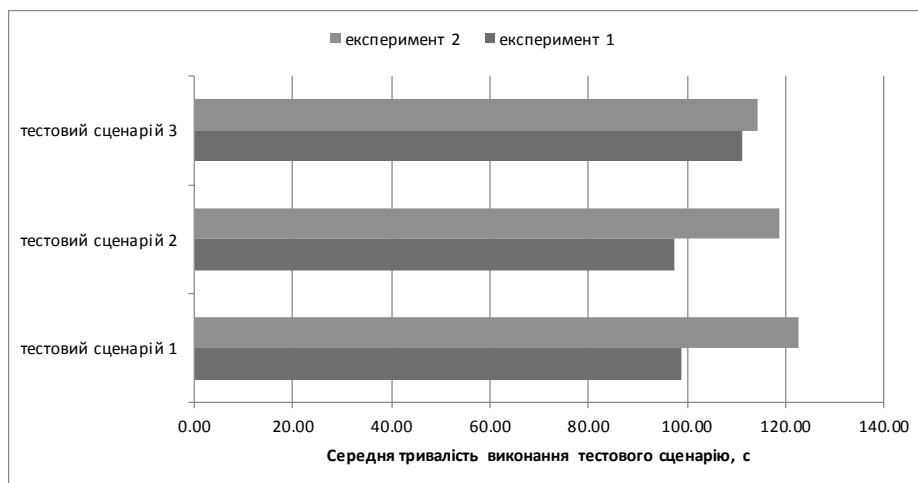


Рис. 10. Порівняння тривалості виконання тестових сценаріїв у двох експериментах

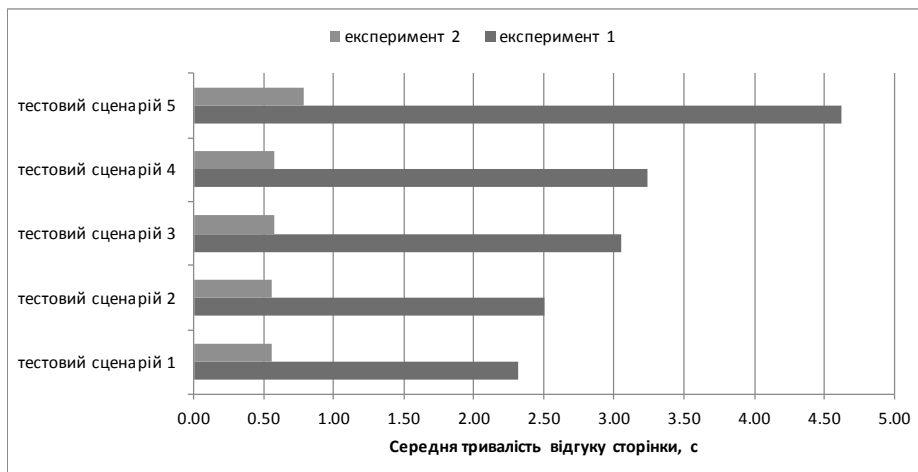


Рис. 11. Порівняння тривалості компонентів досліджуваної системи у двох експериментах

На рис. 9 представлено реакцію редіс кешу на процес навантажувального тестування. У зв'язку з особливістю поведінки програмно-апаратної системи кількість одночасних з'єднань до цього компонента не перевищує 30, однак обсяг трафіку, який проходить через цей компонент, значний, оскільки він призначений для кешування токенів доступу до системи після того, як користувач пройшов процес автентифікації.

Підставою для оцінювання покращення/погіршення процесу навантажувального тестування є наведені нижче рисунки, які дають змогу оцінити середню тривалість виконання тестових сценаріїв та кількість охоплених тестових сценаріїв (рис. 10), а також середню тривалість завантаження сторінки (рис. 11) (Shehada et al., 2013; Vergados et al., 2013). Варто зазначити, що на обох рисунках наведено найбільш відмінні результати, хоча фактична їх кількість є на декілька порядків більшою.

Висновки

Здійснено навантажувальне тестування досліджуваної розподіленої гетерогенної інформаційно-телекомунікаційної системи, яка базується на cloud інфраструктурі. У процесі навантажувального тестування використано розроблений програмний сервіс системи навантажувального тестування, запропоновано моделі корпоративного клієнта та спільної cloud інфраструктури. За результатами навантажувального тестування виявлено вузьке місце у досліджуваній системі, а також отримано оцінки використання інфраструктурних компонентів системи і метрики якості обслуговування та якості сприйняття послуг користувачами досліджуваної інформаційно-телекомунікаційної системи. Показано, що оновлення програмного коду системи робить неможливим розроблення та використання навіть імітаційних універсальних моделей системи загалом, оскільки навіть незначні зміни програмного коду, який логічно поєднує компоненти системи, призводить до зміни поведінки всієї системи, що не можна врахувати в аналітичній та у імітаційній моделях (Катеринчук і Періг, 2014). У зв'язку з цим запропонована методика дослідження розподілених гетерогенних інформаційно-телекомунікаційних систем на базі cloud інфраструктури є важливим інструментом у процесі їх проектування та експлуатації, яка дає змогу вирішити проблему гармонізації рівнів запропонованої концептуальної структури інформаційно-телекомунікаційних систем із покращенням значень метрик якості обслуговування та оптимальним розподілом ресурсів таких систем.

Вдосконалення процесу навантажувального тестування пов'язане із розширенням набору охоплених сценаріїв використання програмної інформаційно-комунікаційної системи.

Також процес навантажувального тестування може бути покращений ретельним плануванням ітеративного процесу, параметрів користувацького навантаження тощо.

1. Shehada, M., Fu, B., Thakolsri, S., Kellerer, W., 2013. *QoE-based resource reservation for unperceivable video quality fluctuation during Handover in LTE*. 2013 IEEE 10th Consum. Commun. Netw. Conf. CCNC 2013 171–177. <https://doi.org/10.1109/CCNC.2013.6488442>. 2. Vergados, D. J., Sgora, A., Michalas, A., Vergados, D. D., Laulajainen, J.P., Jiang, Y., 2013. *A QoE-driven adaptation scheme for video content delivery in LTE networks*. Proc. Int. Symp. Wirel. Commun. Syst. 9, 773–777. 3. Жураковський, Б. Ю., 2014. *Розробка структури програмного забезпечення при об'єктному підході. Системи управління, навігації та зв'язку* 165–170. 4. Катеринчук, І. С., Періг, В. М., 2014. *Удосконалення аналітичних залежностей показників якості функціонування телекомунікаційних систем з урахуванням умов невизначеності. Збірник наукових праць НАДПСУ* 75–79. 5. Лаврів, О. А., 2018. *Методи та моделі надання послуг в гетерогенних розподілених інформаційно-телекомунікаційних системах. Національний університет “Львівська політехніка.”* 6. Толіопа, С. В., Недайбіда, Ю. П., Котова, Ю. В., 2014. *Современные проблемы создания сложных информационно-управляющих систем реального времени в условиях конфликта. Сучасні інформаційні технології у сфері безпеки та оборони* 100–106.