

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Підвищення ефективності тестування десктопних застосунків шляхом використання хмарних технологій»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
(код, найменування спеціальності)
освітньо-професійної програми «Інженерія програмного забезпечення»
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Ілля ЗАЇЧКО
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-61
Ілля ЗАЇЧКО

Керівник: _____ Олег ІЛЬІН
д.т.н., професор

Рецензент: _____

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« _____ » _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Заїчку Іллі Олеговичу

1. Тема кваліфікаційної роботи: «Підвищення ефективності тестування десктопних застосунків шляхом використання хмарних технологій»

керівник кваліфікаційної роботи Олег ІЛЬІН д.т.н., професор,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» жовтня 2023 р. №145.

2. Строк подання кваліфікаційної роботи «29» грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, основи тестування програмного забезпечення вимоги до хмарних сервісів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1.Дослідження методів тестування десктопних застосунків.

2.Аналіз рівнів та видів тестування програмного забезпечення.

3.Розробка методів використання віртуального обладнання під час тестування.

5. Перелік графічного матеріалу: *презентація*

1. Задачі створення схеми тестування десктопного застосунку.
2. Традиційна схема тестування.
3. Критерії оцінювання процесу тестування.
4. Схема методу тестування десктопних застосунків.
5. Порівняльний аналіз тестування.

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз проблем тестування десктопних застосунків, підбір та дослідження визначеної літератури	19.10-12.11.23	
2	Аналіз існуючих підходів до тестування та вибір методів для тестування десктопних застосунків	13.11-19.11.23	
3	Дослідження особливостей використання хмарних технологій для тестування десктопних застосунків	20.11-26.11.23	
4	Розробка методу та засобів тестування десктопних застосунків	27.11-03.12.23	
5	Апробація системи та аналіз отриманих результатів	04.12-10.12.23	
6	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
7	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти

(підпис)

Ілля ЗАЇЧКО

Керівник
кваліфікаційної роботи

(підпис)

Олег ІЛЬІН

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 55 стор., 6 табл., 10 рис., 29 джерел.

Мета роботи – підвищення ефективності тестування десктопних застосунків за рахунок використання віртуалізації, яка базується на хмарних технологіях.

Об'єкт дослідження – процес тестування десктопного застосунку.

Предмет дослідження – методи та технології тестування десктопних застосунків.

Короткий зміст роботи: У роботі проведено дослідження методів тестування десктопного програмного забезпечення. Проаналізовано основні методи використання віртуалізації. Проаналізовано принципи роботи хмарних гіпервізорів націлених на тестування десктопних програмних продуктів. Проаналізовано традиційну схему тестування програмного застосунку. Створено власну схему тестування десктопного програмного забезпечення.

Визначено особливості методу тестування десктопних застосунків з використанням хмарного гіпервізору.

Досліджено позитивні та негативні характеристики тестування окремо на локальному та тестування окремо на віртуальному тестовому обладнанні

Проведено порівняльний аналіз тестування на локальному та віртуальному обладнанні за розробленою схемою тестування десктопного програмного забезпечення. Підведено підсумки результатів тестування.

КЛЮЧОВІ СЛОВА: ДЕСКТОПНИЙ ЗАСТОСУНОК, ВІРТУАЛІЗАЦІЯ, КЛАСТЕР, ВІДМОВОСТІЙКІСТЬ, МАНУАЛЬНЕ ТЕСТУВАННЯ, АВТОМАТИЗОВАНЕ ТЕСТУВАННЯ, ЛОКАЛЬНЕ ОБЛАДНАННЯ, ВІРТУАЛЬНЕ ОБЛАДНАННЯ.

ABSTRACT

Text part of the master's qualification work:

55 pages, 10 pictures, 6 table, 29 sources.

The purpose of the work increase the efficiency of testing desktop applications by using cloud-based virtualization.

Object of research – the process of testing a desktop application.

Subject of research – methods and technologies for testing desktop applications.

Summary of the work: The paper studies the methods of testing desktop software. The main methods of using virtualization are analyzed. The principles of operation of cloud hypervisors aimed at testing desktop software products are analyzed. The traditional scheme of testing a software application is analyzed. An own scheme for testing desktop software is created.

The features of the method of testing desktop applications using a cloud hypervisor are determined.

The positive and negative characteristics of testing separately on local and testing separately on virtual test equipment are investigated

A comparative analysis of testing on local and virtual hardware according to the developed scheme for testing desktop software is carried out. The test results are summarized.

KEYWORDS: SOFTWARE, DESKTOP APPLICATION, VIRTUALIZATION, CLUSTER, FAULT TOLERANCE, MANUAL TESTING, AUTOMATED TESTING, LOCAL HARDWARE, VIRTUAL HARDWARE.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	10
ВСТУП.....	11
РОЗДІЛ 1 Вивчення методів тестування десктопного програмного забезпечення.....	13
1.1 Визначення ролі тестування при розробці десктопного програмного продукту.....	13
1.2 Аналіз критеріїв якості десктопного застосунку.....	15
1.3 Визначення традиційних методів тестування десктопного програмного застосунку.....	22
1.4 Дослідження автоматизації при тестуванні десктопного програмного забезпечення.....	26
РОЗДІЛ 2 Дослідження ролі віртуалізації в тестуванні десктопних застосунків.....	29
2.1 Існуючі типи організації віртуальних середовищ.....	29
2.2 Визначення переваг та недоліків тестування на локальному обладнанні..	31
2.3 Визначення існуючих методів віртуалізації та їх характеристик.....	33
2.4 Дослідження переваг тестування використовуючи хмарні технології.....	37
РОЗДІЛ 3 Аналіз методу тестування десктопних застосунків з використанням хмарного гіпервізору.....	39
3.1 Визначення особливостей методу тестування десктопних застосунків з використанням хмарного гіпервізору.....	39
3.2 Дослідження показників продуктивності на локальному обладнанні.....	42
3.3 Дослідження показників продуктивності на віртуальному обладнанні....	44
3.4 Аналіз результатів тестування на локальному та віртуальному обладнанні.....	46
ВИСНОВКИ.....	48
ПЕРЕЛІК ПОСИЛАНЬ.....	49
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....	50

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення

DB – DataBase

SQL – Structured Query Language

URL – Uniform Resource Locator

API – Application Program Interface (інтерфейс прикладної програми)

MS – Microsoft

QA – Quality Assurance (гарантія якості)

TMS – Testing Management System (система тест-менеджменту)

ВСТУП

Підтримка та розробка програмного забезпечення завжди потребувала рішень для підвищення ефективності процесу тестування розроблених застосунків. В наш час, більшість популярних застосунків – це якісне програмне забезпечення. Кожна компанія використовує десятки десктопних застосунків різного виду та призначення для виконання поставлених задач, тому щодня розробляється та покращується програмне забезпечення. Завжди десктопні застосунки використовуються на різних підприємствах та організаціях, у військовій та медичній сферах, у цих сферах застосунок може напряду впливати на людину. Тестування десктопних програмних застосунків відбувається значно повільніше ніж цього очікує замовник та інші зацікавлені особи. Тому крім рішення автоматизації тестування, розробляються методи підняття ефективності тестування десктопного програмного забезпечення. Ефективне тестування допомагає покращити продуктивність програмного забезпечення, зменшуючи кількість помилок та забезпечуючи високу стабільність системи. Тестування гарантує, що програмний продукт відповідає вимогам замовника і виконує функції, які йому призначені. Якщо програмне забезпечення добре протестоване, його розгортання та підтримка стає більш простою і менш часом та ресурсозатратною задачею. Отже, тестування десктопних застосунків є важливою частиною розробки програм. Людські помилки можуть призвести до дефекту або збою на будь-якому етапі життєвого циклу розробки програмного забезпечення. Наслідки помилки мажуть бути непомітними для користувача або мажуть стати катастрофічними для компанії.

Існує багато різних видів тестування, які використовуються у десктопному, мобільному і вебтестуванні. Вони поділяються за рівнями, типами і видами, а також за часом, коли вони використовуються. Багато часу витрачається на проведення різних видів тестів, на різних проєктах – різні підходи, але дуже часто використовуючи різні види тестів навіть тоді, коли їх використання не принесе

бажаного ефекту, тестувальники витрачають час, який можна було би використати більш раціонально. Ця проблема існує на великих проєктах і на малих, у різних компаніях і на проєктах з різними методологіями.

Мета роботи – підвищення ефективності тестування десктопних застосунків за рахунок використання віртуалізації, яка базується на хмарних технологіях.

Для досягнення мети в роботі було вирішено наступні завдання:

1. Провести аналіз проблем тестування десктопних застосунків.
2. Провести аналіз існуючих підходів до тестування десктопних застосунків.
3. Дослідити особливості використання хмарних технологій для тестування десктопних застосунків.
4. Розробити метод та засоби тестування десктопних застосунків з використанням хмарних технологій.
5. Провести апробацію системи та виконати аналіз отриманих результатів.

Об'єкт дослідження – процес тестування десктопного застосунку.

Предмет дослідження – методи та технології тестування десктопних застосунків.

Методи дослідження: вимірювання та порівняння показників ефективності тестування десктопного застосунку.

Наукова новизна та практична значущість отриманих результатів: розроблено метод та систему для тестування десктопних застосунків, які на відміну від існуючих, ґрунтуються на використанні хмарних технологій.

Апробація результатів та публікації. Робота пройшла апробацію на науково-практичних конференціях: Всеукраїнський науково-практичний форум «Так» – Луцьк: ДВНЗ «ДонНТУ», 2023. За результатами апробації опубліковано тези доповідей [15] [19].

Структура роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1. АНАЛІЗ ОСНОВНИХ МЕТОДІВ ТЕСТУВАННЯ ДЕСКТОПНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Визначення ролі тестування при розробці десктопного програмного продукту

В наш час процес розробки будь-якого виду програмного забезпечення неможливо уявити без тестування. Кожний розроблений застосунок потребує перевірки та визначення відповідності до вимог заявлених замовником. Сам процес тестування вже давно визначений одним із основних етапів під час розробки програмного застосунку. Кожен користувач хоче використовувати якісне програмне забезпечення, саме тому для залучення великої кількості користувачів нам необхідно випускати якісний та корисний продукт. За перевірку цих основних показників відповідає тестування.

Тестування програмного забезпечення – визначається як процес порівняння заявлених вимог замовника з тим, що по цим вимогам було створено, перевірка відбувається при штучно створених ситуаціях, які ще називають тестові випадки, на обмеженому наборі тестів, які виконуються згідно визначеної стратегії тестування[10].

Тестування також відповідає за контроль якості програмного продукту – це сукупність дій, які відбуваються над продуктом в процесі розробки для отримання інформації про актуальний стан продукту, відповідність заявленому рівню продукту, відповідність до вимог та готовність до релізу. [7]

Через різноманітність операційних систем та архітектури програм, нам необхідно виконувати різні види тестування.

Саме через це десктопне тестування є однією з найважливіших етапів розробки програмного забезпечення. Важко уявити підприємство та організацію, яка не використовує десктопне програмне забезпечення.

Вони широко застосовуються майже у всіх сферах, в наш час неможливо уявити навіть військову сферу без десктопних програм.

Тестування з'явилося за рахунок людських помилок які призводять до багів у програмному застосунку. Вартість деяких помилок в програмному забезпеченні куштує мільйони доларів.

Саме тому, для покриття програмного продукту тестуванням проходить через 3 основні етапи:

- Виявлення дефектів в програмному забезпеченні
- Покращення недоліків в системі або компоненті
- Підвищення якості програмного забезпечення

Важливість виявлення критичних проблем в програмах на ранніх етапах розробки дозволяє зменшити вартість виправлення. (рис 1.1)

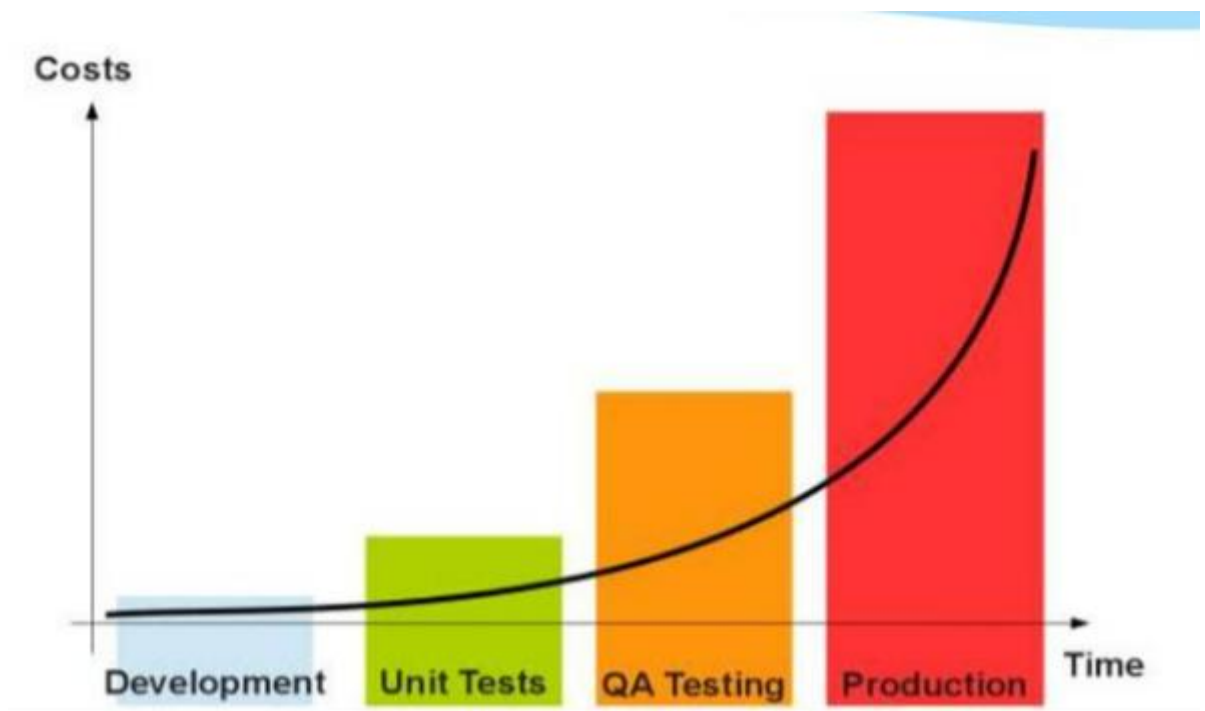


Рис 1.1 – Схема розрахунку вартості знайденого дефекту на різних етапах розробки.[19]

Нажаль деякі помилки в програмному забезпеченні можуть вартувати людського життя, це доводить що проведення тестування та раннє знаходження

помилки в програмному продукті, застосовуючи необхідні види та методи тестування, забезпечує якість та надійність застосунку.

Для значного зменшення виявлення дефектів на кінцевих стадіях розробки, було прийнято запровадити раннє тестування. Тестування починається вже на етапі збору та формування вимог замовника для виявлення дефектів на перших етапах розробки програмного забезпечення. [8]

Роль тестування для розробки якісного програмного застосунку формується за рахунок таких критеріїв:

- без тестування неможливе існування якісного застосунку, так як воно знаходить дефекти або помилки перед релізом продукту;
- тестування забезпечує програмам надійність та зручність у використанні;
- якісно проведене тестування програмного забезпечення забезпечує зручну та продуктивну роботу програмного продукту.

Певні методи тестування, використовуються на кожному рівні тестування, у поєднанні з рівнем досвіду тестера гарантує надійне програмного забезпечення. В кінцевому результаті клієнт отримує програмне забезпечення яке відповідає усім визначеними критеріям якості.

1.2 Визначення критеріїв якості десктопного застосунку

Забезпечення якості програмного забезпечення це завжди актуальна тема у процесі розробки програмного забезпечення. Чим вища якість програмного продукту тим більша кількість користувачів та потенційних партнерів та клієнтів.

На протязі багатьох років, різні вчені та цілі організацій визначали значення “якість” по-різному.

Уолтер Демінг вважав, що проблеми у визначенні якості є переклад майбутніх вимог користувачів у вимірні характеристики, таким чином програмний продукт має бути розроблений та розповсюджений, для забезпечення задоволення з приводу ціни, яку клієнт буде платити[16]. Це не є легкою справою, і як тільки

один споживач задоволений продуктом, згодом потреби користувачів змінюються та з'являються конкуренти.

Основна думка Демінга з приводу терміну “якість” в тому, що якість програмного продукту визначається з точки зору задоволеності споживачів, це дає кращу характеристику, ніж “відповідність до вимог”

Корпорація IBM запровадила свою теорію, “якість, керується ринковими потребами”.

Більш частіше, термін якості використовується відповідно до визначених стандартів якості ISO 9000, це стандарт визначається як ступінь відповідності характеристик до вимог[3],[7]. Якість програмного забезпечення визначається існуючими та потенційними клієнтами, так як ніхто краще клієнта не може перевірити рівень користі певного продукту для потреб які виконує компанія.

Вчений Джозеф М. Джуран визначив два терміни якості:

- Якість складається з тих характеристик продукту, які відносяться до заявлених клієнтом потреб, цим самим забезпечуючи успіх продукту.
- Якість включає відсутність недоліків.

Якість програмного забезпечення є відносним поняттям, так як визначення якості залежить від умов та області застосування певного продукту.

Якість ПЗ, визначається декількома основними аспектами:

- Якісно побудовані процеси життєвого циклу ПЗ
- Якість програмного продукту
- Якість впровадження та підтримки.

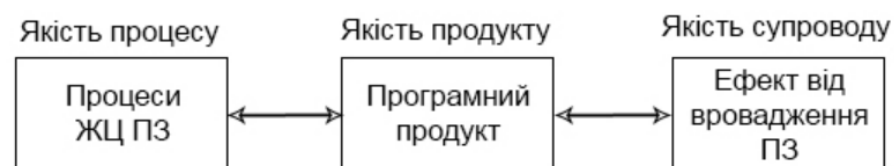


Рис1.2 – Основні аспекти якості ПЗ [17]

Проаналізувавши основні аспекти якості ПЗ, важливу роль у забезпеченні якості програмного забезпечення відіграє якість супроводу вже випещеного програмного продукту.

Визначення якості десктопного програмного застосунку буде залежати від обладнання та області застосування продукту. Якість програмного забезпечення визначається, як вся сукупність його характеристик, які відносяться до можливості задовільнити висловлених та передбачуваних вимог усіх зацікавлених осіб. Поверхневе визначення якості дає нам чітку відповідь на питання “Які задачі вирішує система?” [15].

Згідно з моделлю якості програмного забезпечення ISO 9126-1 визначено 6 основних характеристик, до якої віднесли функціональність, надійність, зручність використання, ефективність, зручність супроводу, портативність.



Рис 1.2 – Модель якості програмного забезпечення (ISO 9126-1)[20]

Всього задіяно в тестуванні якості вісім основних характеристик:

- Функціональність

Основною задачею є перевірка усіх наявних функцій в системі на відповідність до вимог та наскільки функції системи вирішують потреби замовника.

- Перевірка на функціональну повноту забезпечує впевненість що всі необхідні функції існують

- Функціональна коректність перевіряє чи всі функції працюють так, як це описано у вимогах.

- Головним аспектом в тестуванні функціональності є важливість перевірити функціональну доцільність, тобто провести тестування чи система не містить зайвих та непотрібних замовнику функцій.

- Надійність.

Показник надійності програмного застосунку, дає замовнику та іншим зацікавленим особам розуміння наскільки розроблений продукт є надійним. Необхідним є визначити скільки система протягом визначеного часу буде працювати стабільно та без збоїв. Тестування надійності відбувається за наступними параметрами:

- Завершеність – визначає скільки система може працювати без помилок.

- Готовність – визначення потенційних навантажень та наскільки система готова обробляти великі потоки даних.

- Відмовостійкість – визначення поведінки програми при відключеннях від мережі або інших непередбачуваних ситуаціях.

- Відновлюваність – як система зберігає данні під час збоїв програми.

- Зручність використання.

Даний показник перевіряє наскільки дане ПЗ зручне при вивченні та роботі в ньому, а також має бути привабливим для користувачів. Тестування зручності використання відбувається за наступними параметрами:

- Навченість – як швидко новий користувач зможе використовувати систему

- Керованість – програма повинна містити усі необхідні кнопки та функції для зручного використання
- Захищеність від помилки користувача – користувач своїми діями не повинен зламати програму
- Естетика – користувацький інтерфейс повинен бути гарним та приємним
- Доступність – перевірка, як система оптимізована для використання людей з обмеженими можливостями
- Продуктивність\Ефективність.

Показник продуктивності програми залежить від кількості задіяної пам'яті при виконанні певного набору функцій або розрахунків. Ефективність будь-якого десктопного програмного продукту залежить від процесора, пам'яті та накопичувача. Тестування продуктивності дозволяє знаходити проблеми в системі з метою запобігання їх негативному впливу на процеси програми:

- Визначення основної кількості користувачів продукту.
- Заміри часу виконання функцій або обчислень програмним продуктом.
- Визначення ефективності системи з різними рівнями навантаження.
- Аналіз допустимих меж продуктивності при різних навантаженнях.

Визначення потенційних можливостей.

- Перевірка скільки ресурсів використовує програма під час навантаження.
- Зручність супроводу.

На даному етапі відбувається перевірка архітектури програмного продукту та програмних рішень, для розуміння наскільки зручно впроваджувати вже в існуючий програмний код нові зміни. Зміни та виправлення не повинні руйнувати та впливати на функції зміни в яких не відбувались.

Критерії по яким відбувається тестування зручності супроводу:

- Модульність – Для зручності система повинна бути розділена на модулі.

- Можливість багаторазового використання – використання раніше створених функцій та рішень в інших програмах.

- Аналізуємість – проста оцінка впливу різних модулів на систему. Проста архітектура забезпечить зручну діагностику та пошук недоліків в майбутньому.

- Модифікованість – оцінка зручності внесення змін в систему. При внесенні виправлень не з’являються нові проблеми в програмі

- Можливість протестувати – кожен функцію яка існує в програмі ми можемо перевірити.

- Переносимість\Кросплатформність.

Даний етап відповідає за перевірку поведінки програмного продукту під час переносу в іншу систему. Основна задача перевірити чи система правильно працює на операційних системах які зазначив у вимогах замовник або інші зацікавлені особи.

Основними показниками переносимості є:

- Адаптованість – можливість використовувати ПЗ на різних платформах та операційних системах

- Встановлюваність – програма без проблем встановлюється на платформи та операційні системи які зазначені у вимогах до продукту.

- Взаємозамінність – програма оновлюється на нові версії без втрати даних та помилок.

- Безпека.

Будь-яка програма повинна гарантувати безпеку даних та відслідковувати дії користувачів в програмі. Дане тестування спрямоване на пошук і виявлення прогалин в захищеності даних користувачів системи. Якість безпеки перевіряється за наступними показниками:

- Неподільність – обмежений доступ до даних, визначення потенційної шкоди при втраті тих чи інших даних користувачів.

- Конфіденційність - під цим критерієм перевіряється обмеженість певним користувачам системи доступу до даних.

- Цілісність - цілісною системою вважається та програма в яку не може проникнути зловмисник.
- Доступність – функції та дані програми повинні бути доступні тільки авторизованим користувачам.
- Відслідковуваність – програма має відслідковувати усі дії користувачів і зберігати цю інформацію в окремому модулі або файлі.
- Справжність – програма має бути ліцензованою, зловмисники не повинні мати змогу підробити ваш продукт.
- Співіснування.

Перевірка поведінки програмного забезпечення з іншими запущеними програмами на певній операційній системі. Основною задачею є переконатись що система не заважає роботі інших систем. Система також має правильно розподіляти ресурси з іншими програмами

Після проведення тестування по описаним характеристикам якості ми визначаємо придатність до використання програмного забезпечення. Також при тестуванні якості потрібно розуміти що усі користувачі різні, відповідно програму кожен використовує по різному. У програми може бути безліч клієнтів клієнтів з різних сфер і кожен клієнт може покривати різні потреби нашим застосунком. Тому продукт має володіти різними варіантами використання. Як зазначав Juran, кожен варіант використання – це показника якісного програмного забезпечення, і тому продукт також може кваліфікуватись за категоріями.

Для того щоб вирішити задачі кожного клієнта, необхідно розуміти потреби кожного і чітко визначити чи зможе наш продукт задовільнити ці потенційні потреби.

Тому якісне програмне забезпечення для замовника, в першу чергу це той продукт який повністю задовольняє його реальні та життєві потреби, як специфіковані та і інші.

1.3 Визначення традиційних підходів до тестування десктопного програмного застосунку

Усі види та типи тестування можна розділити на функціональне та нефункціональне тестування.

Основною метою тестування функціональності десктопного застосунку є перевірка усіх системних функцій на відповідність функціональним вимогам які висловлені замовником або іншими зацікавленими особами.

При тестуванні десктопного застосунку функціональна перевірка виконується найчастіше. Відбувається це за рахунок створення штучних ситуацій або при наданні системі тестових даних для обробки, по результатам роботи функції яка перевіряється, тестер порівнює очікуваний результат роботи з тим результатом який отримали.

Більшість звикла говорити що функціональне тестування – це тестування чорної скриньки, так як тестування відбувається за допомогою графічного інтерфейсу користувача, без доступу до вихідного коду програми.

Усі наступні види тестування які використовуються в тестуванні десктопних застосунків відносяться до нефункціонального виду тестування:

- Тестування локалізації – даний вид тестування є одним із основних так як під час проведення тестування локалізації потрібно використовувати гнучкі підходи для швидкого та якісного виконання тестів на різних операційних системах та специфічних налаштуваннях які можуть бути визначені у вимогах до продукту.

- Тестування конфігурації продукту – цей вид тестування не менш важливіший у проведенні тестування десктопного застосунку. Так як кожен десктопний застосунок повинен мати хоча б одну власну БД. Задачею тестування є визначити проблеми інтеграції з БД, АРІ та іншими застосунками або сервісами на різних операційних системах.

- Тестування інсталяції десктопного застосунку, перевіряє процес встановлення, оновлення та видалення програми. Тестується також які файли та

додаткові служби встановлюються разом з програмним забезпеченням, які існуючі служби використовуються для роботи програми.

- Тестування продуктивності – визначивши які служби та інші процеси використовує встановлена програма, необхідно визначити сумарну кількість використання ресурсів комп'ютера безпосередньо нашим програмним застосунком. Також основною перевіркою під час тестування продуктивності є визначення потенційних можливостей програмного продукту виходячи із кількості користувачів які визначив замовник або інші зацікавлені особи.

- Тестування сумісності - даний вид тестування перевіряє як програма поводить себе разом з іншими встановленими програмами. Чи не викликають виникають проблеми в інших програмах через роботу нашого десктопного застосунку.

- Тестування безпеки – саме в даному виді тестування перевіряються усі вразливості системи до проникнення та викрадення даних користувачів.

Для якісного тестування безпеки тестерам необхідно мати додаткові знання в даній області.

Виявлення потенційних загроз, запровадження дій у разі проникнення сторонніх в систему, проведення аудиту на наявність вразливих місць в системі – усі ці перевірки відносяться до тестування безпеки. Також важливим показником безпеки є відслідковуваність, тому необхідно перевіряти як система відслідковує дії користувачів і куди зберігає ці дані.

- Тестування на відновлення та відмову.

Даний вид тестування перевіряє як система оброблює та зберігає дані під час нестандартних ситуацій, таких як втрата мережі, відключення світла та інші ситуації. Надійна система повинна бути готовою до будь-якої ситуації та коректно зберігати данні при незапланованих збоях системи.

- Тестування зручності використання (UI)

Головна мета даного виду тестування перевірити, чи всі поля та інші елементи графічного інтерфейсу користувача розміщені згідно макетів та вимог які

визначив замовник. Даний вид тестування передбачає також перевірку на зручність роботи з продуктом нових користувачів.

Залежно від вимог в даний вид тестування можуть входити і інші перевірки які стосуються зручності використання додатку, наприклад перевірка інтернаціоналізації або перевірка функціоналу для користувачів з обмеженими можливостями.

Визначивши основні види тестування десктопного програмного забезпечення створено традиційну схему проведення тестування десктопних застосунків.

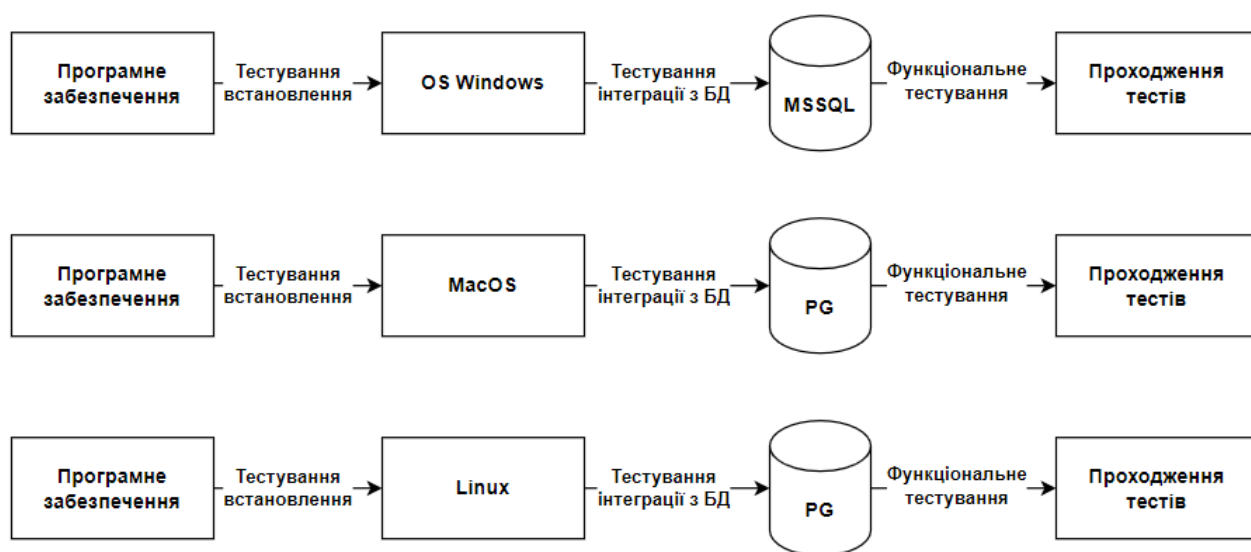


Рис 1.3 – Схема традиційного тестування десктопних застосунків [6]

На традиційній схемі тестування показано:

- Тестування встановлення. Процес інсталяції застосунку на три різні операційні системи.
- Тестування локалізації.

Цей етап тестування ми спостерігаємо по трьом незалежним один від одного процесам встановлення десктопного програмного забезпечення на різні операційні системи (Windows, MacOS, Linux).

- Тестування інтеграції.

На схемі представлено три незалежні БД які окремо інтегруються з встановленим програмним забезпеченням.

- Функціональне тестування. Тестування функцій десктопного застосунку має початись після успішного завершення усіх попередніх етапів тестування.

Недоліком даного методу є те, що відбувається 3 незалежних процеси тестування десктопного застосунку на різних операційних система та з використанням окремих БД. Такий підхід є дорогим у використанні тестового обладнання та повільним порівняно з іншими підходами до тестування десктопних застосунків.

1.4 Дослідження автоматизації при тестуванні десктопного програмного забезпечення.

Автоматизоване тестування грає важливу роль в процесі розробки програмного забезпечення і може призвести до численних переваг[9]. Ось деякі з ключових ролей автоматизованого тестування в розробці:

- Швидкість виконання тестів: Автоматизовані тести можуть бути виконані набагато швидше, ніж ручні, дозволяючи швидше виявляти помилки і проблеми в коді. Це особливо корисно в агільних та ітераційних методологіях розробки.

- Повторюваність: Автоматизовані тести можуть бути запущені в будь-який момент ілюбові, що забезпечує повторюваність результатів і допомагає виявляти проблеми, які можуть виникати лише під певними умовами.

- Збереження часу і ресурсів: Зменшення необхідності в ручному тестуванні дозволяє командам розробників і QA фокусуватися на інших аспектах розробки, зберігаючи час і ресурси.

- Виявлення помилок на ранніх етапах розробки: Автоматизовані тести можуть бути включені у неперервну інтеграцію (CI) і неперервну поставку (CD), допомагаючи виявляти помилки на ранніх етапах розробки, коли вони ще не стали серйозними проблемами.

- Збереження історії тестування: Автоматизоване тестування зазвичай включає в себе створення тестових скриптів і скриптів для перевірки результатів. Це створює документацію тестів і дозволяє вести історію змін, що може бути корисно для виправлення помилок і аналізу результатів.

- Розширення тестового покриття: Автоматизовані тести можуть легко охоплювати велику кількість тестових сценаріїв та конфігурацій, забезпечуючи велике тестове покриття.

- Сприяння реалізації DevOps-процесів: Автоматизоване тестування є ключовою складовою DevOps-процесів, забезпечуючи швидке та автоматизоване внесення змін у програмний код та його тестування.

- Хоча автоматизоване тестування має численні переваги, важливо пам'ятати, що воно не може повністю замінити ручне тестування. Комбінація обох методів може забезпечити найкращий результат у забезпеченні якості продукту.

Проаналізуємо особливості методу автоматизованого тестування. Автоматизоване тестування має свої особливості, які роблять його важливим та ефективним інструментом у розробці програмного забезпечення[6]. Ось деякі основні особливості автоматизованого тестування:

- Спрощення повторюваних тестових сценаріїв: Автоматизація найбільше ефективна для тестів, які можуть бути виконані багато разів з однаковими чи подібними вхідними даними. Це дозволяє виявляти помилки та проблеми в різних версіях програми.

- Неперервна інтеграція (CI) і неперервна поставка (CD): Автоматизоване тестування легко інтегрується в CI/CD-процеси, дозволяючи виконувати тести при кожній зміні коду та автоматично впроваджувати зміни в продакшн.

- Підтримка різних платформ і конфігурацій: Автоматизовані тести можуть бути виконані на різних операційних системах, браузерях, пристроях та інших конфігураціях, що робить їх універсальними для тестування різноманітності вимог.

- Спеціалізовані інструменти: Існує багато інструментів для автоматизації тестування, спеціалізованих для різних видів тестів (наприклад, функціональне, навантажувальне, інтерфейсне тестування). Ці інструменти надають розширені можливості для створення, виконання та аналізу автоматизованих тестів.

- Можливість паралельного виконання тестів: Автоматизовані тести можуть бути виконані паралельно, що дозволяє скоротити час виконання тестів, особливо в умовах неперервної інтеграції та неперервної поставки.

- Генерація звітів і логів: Більшість інструментів автоматизованого тестування надають можливість генерувати детальні звіти та логи після виконання тестів. Це полегшує виявлення і виправлення помилок.

- Масштабованість тестових наборів: Автоматизоване тестування може легко масштабуватися для включення нових тестових сценаріїв або розширення покриття тестування з ростом проекту.

- Вартість та час: Хоча створення автоматизованих тестів вимагає початкових витрат на розробку, у довгостроковій перспективі вони можуть значно зекономити час та кошти, особливо при регулярному виконанні тестів на тривалий період часу.

- Загальною метою автоматизованого тестування є поліпшення ефективності, надійності та якості програмного забезпечення.

Проаналізувавши особливості автоматизованого тестування можна зробити висновок, що даний метод один із традиційних методів який гарантує швидке тестування. Але не завжди даний метод можна застосовувати на різних операційних системах як правило автоматизація тестування широко використовується у веб-розробці.

2 ДОСЛІДЖЕННЯ РОЛІ ВІРТУАЛІЗАЦІЇ В ТЕСТУВАННІ ДЕСКТОПНИХ ЗАСТОСУНКІВ

2.1 Існуючі типи організації віртуальних середовищ

Статистичні дані показують, що середнє завантаження обчислювальних потужностей серверного обладнання з програмним забезпеченням Windows не перевищує 10%. Для систем, що використовують Unix як операційну систему, цей показник зростає, але ніколи не перевищує 20%.

Низький рівень використання серверного обладнання пояснюється підходом "один додаток-один сервер", який переважав на початку 1990-х років. Згідно з цим підходом, кожного разу, коли компанія розгортає новий додаток, потрібно купувати новий сервер. Це стрімко збільшує фінансові витрати на придбання, управління та обслуговування серверного обладнання, споживання електроенергії та утримання великих площ для встановлення обладнання.

Емуляція – це комплекс програмних, апаратних засобів або їх поєднання, що призначене для копіювання (або емуляції) функцій однієї обчислювальної системи (гостя) на іншу, відміну від першої, обчислювальної системи (хосту). Таким чином, емулювання полягає у копіюванні функції якомога ближче до функції оригінальної системи (гостя)[6]. Метою є максимально точне відтворення поведінки на відміну від різних форм комп'ютерного моделювання, в яких імітується поведінка деякої абстрактної моделі. Наприклад, моделювання урагану або хімічної реакції не є емуляцією.

Програмна віртуалізація – це віртуалізація, яка відбувається на рівні ядра операційної системи[6]. Всі сучасні віртуальні машини застосовують загальне модифіковане серверне ядро. Для користувачів кожна віртуальна машина виглядає як повноцінний окремий сервер.

Апаратна віртуалізація – це віртуалізація на апаратному рівні, тобто чесний «розпил» одного сервера на ряд більш слабких машин. Так, кожен сервер повністю ізольований від сусідніх, при цьому лімітуються майже всі ресурси.

Динамічна трансляція – при динамічній (бінарної) трансляції проблемні команди гостьової операційної системи перехоплюються гіпервізором.

Паравіртуалізація – спеціальна операційна система, що взаємодіє з програмою гіпервізору, який надає їй гостьовий прикладний програмний інтерфейс (API). Використовується на заміну використання таблиці сторінок пам'яті.

Контейнеризація – це метод віртуалізації, при застосуванні якого операційна система підтримує декілька ізольованих один від одного ділянок простору для користувачів замість одного. Ці ділянки (контейнери, зони) за поглядами користувачів системи повністю ідентичні за своїми характеристиками окремій операційній системі. Для систем на основі операційної системи Unix ця технологія схожа на поліпшену реалізацію механізму chroot. Ядро операційної системи забезпечує повну ізоляцію контейнерів між собою. Тому програми, що відносяться до різних контейнерів, не можуть мати впливу між собою. [6]

Для програмного забезпечення віртуалізація додатків – це процес використання додатка, перетвореного з такого, що вимагає установки в операційну систему, у такий, що не потребує (потрібно тільки запустити). Для віртуалізації додатків програмне забезпечення віртуалізації визначає при установці віртуалізує додатки, які потрібні компоненти ОС, і емулює їх [11]. Таким чином, створюється необхідне спеціалізоване середовище для конкретного додатка, що віртуалізується. Тим самим забезпечується ізолюваність роботи цього додатка. Для створення віртуального додатку він поміщається у контейнер, що оформлений, як правило, у вигляді папки. При запуску віртуального додатку запускається і контейнер, який є для нього робочим середовищем. Робоче середовище запускається і надає локальні раніше створені ресурси. Вони включають у себе ключі реєстру, файли та інші компоненти, що необхідні для запуску і роботи програми [11]. Таке віртуальне середовище працює як прошарок між додатком і операційною системою, що дозволяє уникнути конфліктів між додатками.

Віртуалізація серверів – це емуляція поведінки системних компонентів, необхідних для запуску програми з метою налагодження і тестування (англ. Application Under Test). Замість віртуалізації компонентів цілком, ця технологія віртуалізує тільки необхідні частини. [11]

2.2 Визначення переваг та недоліків тестування на локальному обладнанні

В сучасному вимоги до десктопних застосунків визначають кросплатформність застосунку, тобто можливість використання на різних операційних системах.

Виходячи з вимог замовника процес тестування десктопного застосунку підлаштовується під кожну операційну систему яка визначена у вимогах до програми.

Для проведення тестування використовуючи локальне оточення, нам необхідно розгорнути окремий комп'ютер під кожну операційну систему з різними конфігураціями і виконати визначений набір тестів.

У вимогах може бути визначено декілька операційних систем та безліч версій цих ОС, які повинен підтримувати десктопний застосунок. І для того щоб переконатись що програма коректно працює на усіх визначених у вимогах ОС та їх версіях нам необхідно провести певний набір тестів на кожній ОС окремо. Для цього необхідно розгортати стільки локальних комп'ютерів, скільки визначено ОС та версій у специфікації. Даний підхід не є гарним рішенням з точки зору бюджету який закладений на проведення тестування.

Отже, основними недоліками в тестуванні використовуючи локальне обладнання можна визначити:

- Бюджет.

Необхідно постійно оновлювати тестове обладнання.

- Прозорість усіх процесів тестування.

Не усі тести розпочинаються одночасно, тому важко визначити загальний статус тестування та на яких моментах ми заблоковані в подальшому тестуванні.

- Відсутність гнучкого тестування.

Тестер не має можливості підключитися на перевірку іншої операційної системи одночасно з іншим інженером.

- Низький рівень тестування продуктивності.

Для тестування ефективності програмного застосунку необхідно мати комп'ютери з високими характеристиками пам'яті та процесору.







 Memory	4.00 GiB [balloon=0]
 Processors	4 (1 sockets, 4 cores)
 BIOS	Default (SeaBIOS)
 Display	Default
 Machine	Default (i440fx)
 SCSI Controller	VirtIO SCSI
 CD/DVD Drive (ide2)	none,media=cdrom

Рис 2.1 Характеристики тестового обладнання для проведення тестування продуктивності.

Визначивши недоліки тестування десктопного застосунку на локальному оточенні, проаналізуємо переваги. Тестування на локальному оточенні корисно для тих програмних продуктів як створені під певну операційну систему.

Також корисно для програмних продуктів з мінімальною кількістю даних, які зберігаються на тому ж оточенні, де і встановлений застосунок або якщо збереження даних відбувається в хмарі, наприклад AWS S3 storage.

Дослідивши переваги та недоліки тестування з використанням локального оточення, можна зазначити що даний підхід корисний для тих проектів, які використовують тільки операційну систему одного виду. Сучасна розробка програмного забезпечення визначає малу кількість десктопних застосунків, які не містять кросплатформність, зазвичай це військові та медичні десктопні застосунки.

2.3 Визначення існуючих методів віртуалізації та їх характеристик

Вперше віртуалізацію використала компанія IBM, інженери компанії намагались запровадити спільне використання робочого часу для власних потреб.

Використання гіпервізорів являють собою технологію, яка розділяє операційну систему та обладнання комп'ютера.

Існує два основних типи гіпервізорів:

- Гіпервізори металів – таку назву вони отримали із-за специфіки роботи, даний тип гіпервізорів розгортається безпосередньо на апаратному забезпеченні[27]. Прикладом може бути ПЗ компанії Microsoft, гіпервізор HyperV.

- Гіпервізори операційної системи – цей тип гіпервізорів розгортається на ОС комп'ютера. Для прикладу визначимо продукт компанії Oracle, гіпервізор Oracle VM Virtualbox.

Основна ціль віртуалізації полягає в тому щоб розгорнути декілька операційних систем, які будуть виконуватись паралельно на одному і тому ж апаратному забезпеченні, за рахунок створення екземплярів обладнання окремо для кожної віртуальної машини.

Розглянемо детальніше гіпервізор HyperV який створює віртуальні машини безпосередньо на апаратному забезпеченні. Кожна віртуальна машина розгорнута за допомогою гіпервізора HyperV це окремий комп'ютер який майже не відрізняється від локального обладнання, та дозволяє керувати операційною системою та програмами. Надає гнучкість у визначенні обчислювальних ресурсів які необхідні віртуальній машині для виконання поставлених завдань. [27]

Hyper-V розгортає кожну віртуальну машину в окремому ізольованому просторі, для можливості одночасно використовувати інше віртуальне обладнання, яке створено на тому ж самому обладнанні.

Це допомагає не залежати віртуальним машинам від помилок та збоїв, які виникли а іншому віртуальному обладнанні. В тестуванні цей підхід часто використовується для проведення тестування продуктивності десктопного програмного застосунку.

Основні переваги Hyper-V полягають в:

- Зручність використовувати розгорнуте віртуальне обладнання.

Розділення робочого навантаження одного комп'ютера між серверами та іншими комп'ютерами, для меншого навантаження на цільове обладнання.

- Безперервність виконання операцій.

Мінімізація впливу незапланованих простоїв робочих навантажень.

- Гнучкість приватного хмарного середовища.

Можливість переносити та розширяти використання ресурсів, в залежності від необхідності та попиту.

- Ефективність розробки та тестування програмного забезпечення.

Можливість створювати різноманітні обчислювальні середовища та використання необхідних програм без ліцензій які необхідно купувати використовуючи ті ж самі програми на локальному обладнанні.

Гіпервізор VMware Workstation дозволяє створювати ізольовані один від одного віртуальні машини які безпосередньо використовують ресурси фізичного обладнання.

За рахунок цього, кожна віртуальна машина має свою власну ОС та унікальний набір програм а також визначений ресурс центрального процесору та оперативної пам'яті.

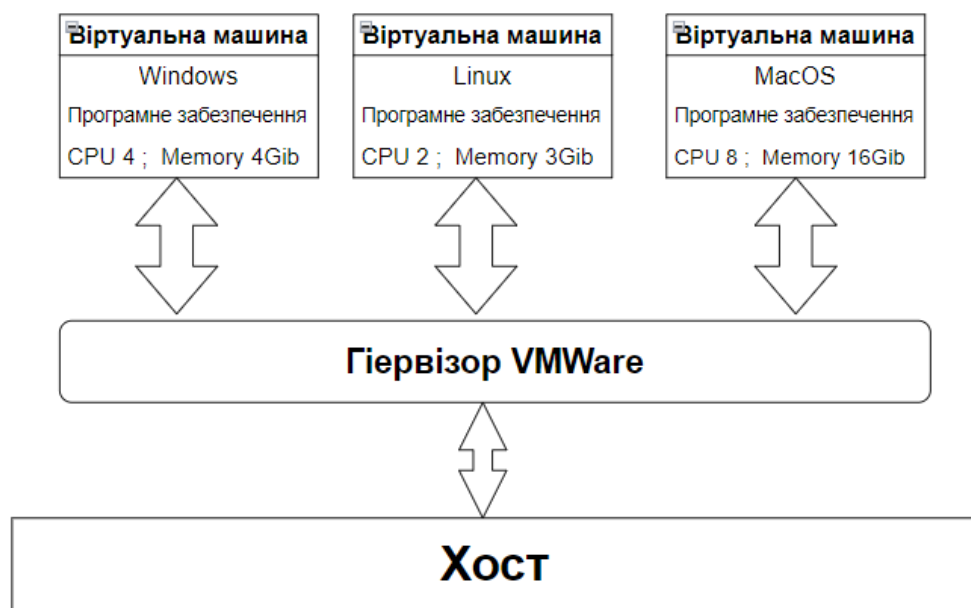


Рис 2.2 Схема роботи гіпервізору VMware Workstation

До переваг можна віднести простоту встановлення та налаштування самого гіпервізору VMWare Workstation. Встановлення гіпервізору нічим не відрізняється від встановлення будь-якої іншої програми на ОС Windows, Linux або MacOS, також не потрібно виконувати маніпуляції по розділенню дискового простору цільового комп'ютеру на якому буде розміщений гіпервізор. Завдяки імітації реального робочого комп'ютеру, на віртуальній машині відсутні проблеми з встановленням та налаштуванням програмного забезпечення. Розгортання операційної системи на віртуальній машині також нічим не відрізняється від встановлення ОС на локальному обладнанні. Особливою перевагою можна зазначити зрозумілий інтерфейс, з яким легко розібратися новому користувачу, який вперше працює з подібним ПЗ. Широкий набір інструментів дає змогу реалізовувати складні функції, які існують в інших гіпервізорах. Основним недоліком гіпервізора VMware Workstation є його велика вартість ліцензії, вона необхідна для використання додаткових інструментів. Є і безкоштовна версія, але більшість функціоналу в безкоштовній версії недоступно. До недоліків варто віднести малу кількість гостьових операційних систем та низький різновид дистрибутивів Linux.

Компанії Oracle належить гіпервізор VirtualBox – це програмний продукт з відкритим вихідним кодом, та має особливі функції для резервного копіювання даних. VirtualBox призначений в основному для віртуалізації комп'ютерів з ОС Windows, але можливість розгортання обладнання з ОС Linux та MacOS також є. Програмне забезпечення відноситься до 2 типу гіпервізорів, та працює на вже розгорнутій операційній системі, як звичайний додаток. Перевагою VirtualBox є можливість безкоштовного завантаження програмного продукту з офіційних джерел. Плюсами у використанні є можливість створювати швидкі збереження системи за рахунок інтеграції програми з покажчиком миші. Створені збереження є можливість використовувати з інших ПК. Особливістю гіпервізору прийнято вважати можливість розділення буферу обміну між віртуальною та локальною операційною системою. Тобто ми маємо можливість напряду копіювати текст та

файли з локального ПК та переносити ці дані на віртуальну машину, що є дуже зручно та швидко.

На основі проаналізованих характеристик, переваг та недоліків гіпервізорів, які використовуються для розгортання віртуального обладнання створено порівняльну таблицю в якій визначено характеристики та недоліки гіпервізорів Hyper-V, VMware Workstation та VirtualBox.

В таблиці 2.1 визначено характеристики та основні недоліки трьох вище описаних гіпервізорів.

Таблиця 2.1

Порівняльна характеристика гіпервізорів

Назва інструменту віртуалізації	Характеристика гіпервізору	Недоліки
Hyper-V	Дозволяє запустити на комп'ютері декілька віртуальних операційних систем одночасно. Кожна віртуальна машина може виконувати свою власну операційну систему.	При використанні з ОС Windows необхідна серверна ліцензія Microsoft.
VMware Workstation	Дозволяє створення ізольованих віртуальних машин з високою продуктивністю	Основні інструменти віртуалізації надаються тільки для платної версії.
VirtualBox	Дає змогу створювати і запускати версію програмного забезпечення комп'ютера, яка називається віртуальною машиною.	Не має можливості збереження через певні проміжки часу, це зменшує надійність роботи операційної системи, яка розміщена на віртуальному обладнанні.

2.4 Дослідження переваг тестування використовуючи хмарні технології

В тестуванні хмарні технології надають різноманітні способи обчислення використовуючи мережу. Можливість використовувати хмарні технології в тестуванні десктопних застосунків з'явилося завдяки технології віртуалізації. За допомогою цієї технології ми маємо можливість тестувати застосунок на різних операційних системах одночасно.

Основою для усіх створених віртуальних машин є сервер, на якому і розміщуються усі віртуальні машини які ми створили та використовуємо.

З досліджень попереднього підрозділу ми визначили що основною перевагою віртуалізації є те що усі створені віртуальні машини, хоч і знаходять на одному сервері який розміщений в хмарі але вони повністю ізольовані одна від одної.

Додатковими перевагами використання хмарних технологій у тестуванні десктопних програмних застосунків можна визначити:

- Можливість дистанційної роботи.

В наш час дана перевага є однією із основних, так як більшість інженерів працює віддалено. Обладнання в хмарі доступне з будь-якої точки світу де є інтернет мережа.

- Економія

Відсутня необхідність купувати локальне обладнання, що значно зменшує витрати та дозволяє спрямувати і кошти на більш необхідні речі.

- Аварійне відновлення хмарного обладнання та резервування.

Використовуючи хмарні технології у нас є можливість виконувати резервне копіювання даних на випадок технічного збою системи.

- Необмежена масштабність

Існують замовники зі специфічними вимогами ресурсів до обладнання. Хмарні рішення забезпечують гнучку зміну ресурсів пам'яті на віртуальній машині

- Кількість місця для зберігання даних програми

Сховище, яке розміщене в хмарі має майже необмежену кількість ресурсів, які не залежать від фізичного пристрою на якому запущене віртуальне тестове обладнання.

Визначивши основні переваги тестування програмного забезпечення а допомогою віртуалізації, можна зазначити що майже жоден десктопний застосунок не тестується без використання віртуалізації.

3 АНАЛІЗ МЕТОДУ ТЕСТУВАННЯ ДЕСКТОПНИХ ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ ХМАРНОГО ГІПЕРВІЗОРУ

3.1 Визначення особливостей методу тестування десктопних застосунків з використанням хмарного гіпервізора

Дослідивши переваги та недоліки ролі хмарних технологій під час тестування десктопних застосунків, стало зрозуміло що успіх тестування десктопного застосунку важко уявити без використання віртуалізації. Саме тому під час використання віртуального тестового обладнання застосовуються різні методики та підходи для ефективного тестування.

Проаналізувавши сучасні методики тестування, було розроблено власну схему тестування десктопних застосунків з використанням хмарного гіпервізора.

Для створенні схеми тестування з хмарним гіпервізором, було визначено три операційні системи (Windows, Linux, MacOS) та дві бази даних різних типів (MSSQL, Postgres).

Згідно визначених операційних систем та БД, було розроблено унікальну схему тестування десктопного програмного забезпечення на усіх ОС одночасно, з використанням двох баз даних.

На рис. 3.1 спостерігаємо три одночасні процеси тестування на різних операційних системах які не залежать одна від одної. Все тестове обладнання розгорнуте на одному хмарному гіпервізорі.

В цьому випадку, гіпервізор може бути піправлінням та підтримуваним постачальником хмарних послуг. Інженери мають можливість використовувати цей гіпервізор для створення та управління своїми віртуальними машинами через інтерфейс хмари.

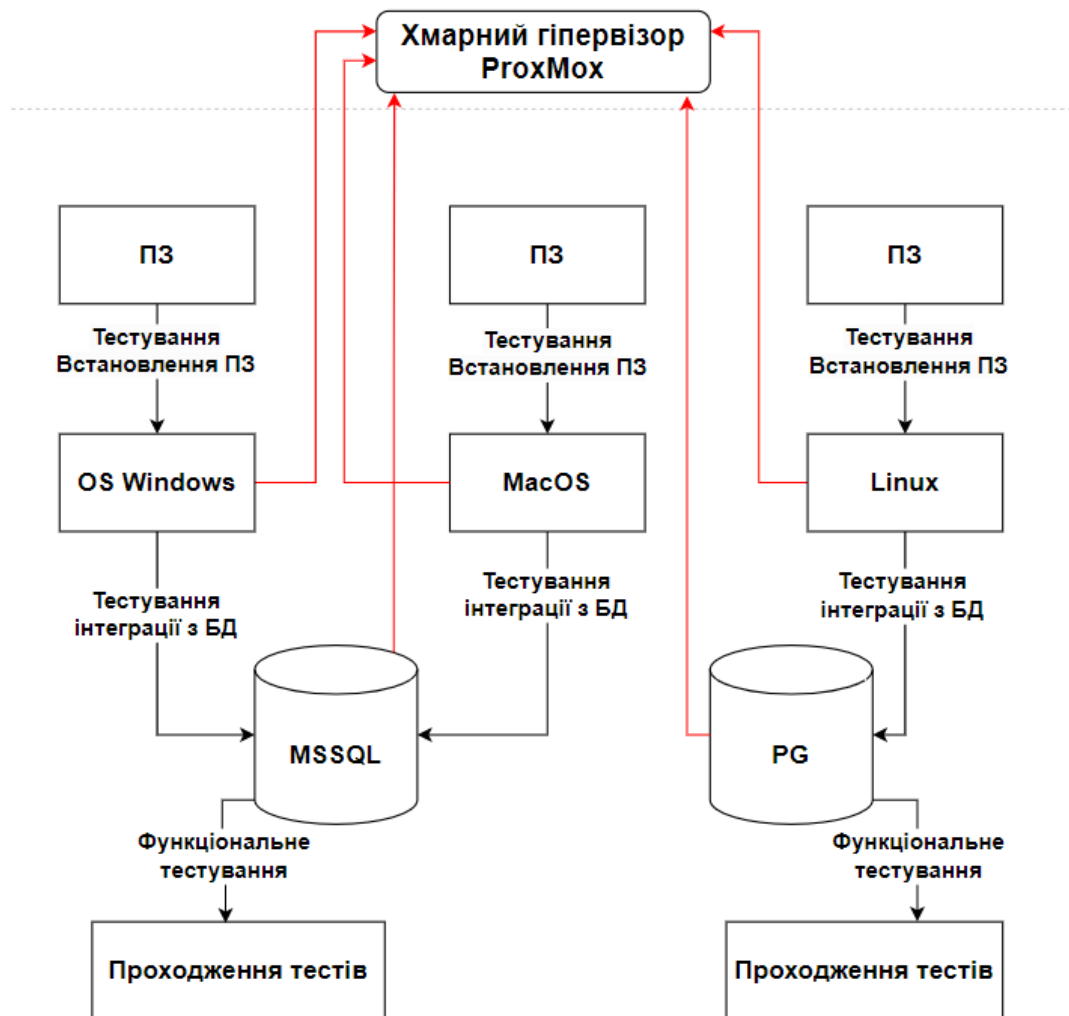


Рис 3.1 Схема методу тестування десктопних застосунків з використанням хмарного гіпервізору

Хмарні гіпервізори можуть включати в себе різні функції, такі як автомасштабування, моніторинг ресурсів, мережеву ізоляцію та інші характеристики, спрямовані на оптимізацію роботи в хмарному середовищі.

В схемі методу тестування з використанням хмарного гіпервізору, широко застосовується функція мережевої ізоляції, яку надає хмарний гіпервізор.

Під час створення схеми було визначено усі основні види тестування десктопного програмного застосунку, а саме:

- Тестування встановлення ПЗ.
- Інтеграційне тестування.
- Функціональне тестування

Результатом тестування за представленою схемою на Малюнку 3.1, є десктопний програмний застосунок, який:

- не містить критичних помилок під час встановлення на будь-яку операційну систему яка визначена у вимогах
- відсутність помилок, під час інтеграції з БД з різних операційних систем
- можливість швидкого проведення повторного тестування на одній або декількох операційних системах, для перевірки виправлення раніше знайденої помилки.

3.2 Дослідження показників продуктивності на локальному обладнанні

Проведено аналіз показників продуктивності тестування на локальному обладнанні, використовуючи локальне обладнання з характеристиками, наведеними в таблиці 3.1.

Таблиця 3.1

Характеристики локального обладнання

Назва обладнання	Acer Swift 3 SF314-511
Процесор	Intel Core i3-1115G4 (3.0 - 4.1 ГГц)
Оперативна пам'ять	4 ГБ
Пам'ять	SSD 256 ГБ
Графічний процесор	Intel UHD Graphics

Для проведення тестування було встановлено тестове десктопне програмне забезпечення svchost.exe на локальне обладнання.

Svchost.exe – це тестовий десктопний застосунок, який був взятий із власного архіву програм, для проведення тестування наступних видів тестування на локальному обладнанні:

- Тестування встановлення.

- Тестування сумісності.
- Тестування функціональності.

Під час проведення тестування було замірено використання ресурсів на локальному обладнанні використовуючи програму Process Hacker. Дана утиліта необхідна для відстежування системних служб та процесів, які використовує та створює програма Svchost.exe, які запущені на різних операційних системах, в нашому випадку 64-розрядна ОС Windows.

Під час проведення тестування ми звертаємо увагу на наступні показники утиліти Process Hacker:

- статистика процесів
- графік продуктивності
- виділена процесом пам'ять
- повна історія виконання процесу
- можливість завершити процес

Використовуючи утиліту Process Hacker показники відображаються в зручному графічному інтерфейсі (рис.3.1).

Провівши тестування на локальному обладнанні за допомогою утиліти Process Hacker, визначено наступні показники ефективності тестування на локальному обладнанні (табл. 3.2)

Таблиця 3.2

Показники ефективності тестування на локальному обладнанні

Критерії продуктивності тестування	Показник використаних ресурсів
CPU usage	3.96 з 4
Memory usage	73.95% (2.95 GiB з 4 GiB)

Визначивши показники ефективності тестування десктопного застосунку, ми спостерігаємо що під час проведення тестування використовується більше ніж половина ресурсів на локальному обладнанні.

Name	PID	CPU	I/O total ...	Private b...	User name	Description
RuntimeBroker.exe	6176	0.02		5.02 MB	EKRAN-3\illia.zaichko	Runtime Broker
RuntimeBroker.exe	6332			6.74 MB	EKRAN-3\illia.zaichko	Runtime Broker
RuntimeBroker.exe	6772			5.54 MB	EKRAN-3\illia.zaichko	Runtime Broker
smartscreen.exe	6956			8.05 MB	EKRAN-3\illia.zaichko	Windows Defender SmartScre...
ApplicationFrame...	5952			5.45 MB	EKRAN-3\illia.zaichko	Application Frame Host
dllhost.exe	7020			1.78 MB	EKRAN-3\illia.zaichko	COM Surrogate
dllhost.exe	8600			3.53 MB	EKRAN-3\illia.zaichko	COM Surrogate
svchost.exe	932	0.03		5.97 MB		Host Process for Windows Ser...
svchost.exe	976			2.88 MB		Host Process for Windows Ser...
svchost.exe	760	2.50	1.39 kB/s	58.55 MB		Host Process for Windows Ser...
rdpclip.exe	5004	0.10		9.41 MB	EKRAN-3\illia.zaichko	RDP Clipboard Monitor
svchost.exe	828			2.15 MB		Host Process for Windows Ser...
svchost.exe	1152			1.74 MB		Host Process for Windows Ser...
svchost.exe	1160			1.75 MB		Host Process for Windows Ser...
svchost.exe	1168			5 MB		Host Process for Windows Ser...
svchost.exe	1260			2.12 MB		Host Process for Windows Ser...
svchost.exe	1316	0.03		3.46 MB		Host Process for Windows Ser...
svchost.exe	1348			12.73 MB		Host Process for Windows Ser...
svchost.exe	1452			12.22 MB		Host Process for Windows Ser...
svchost.exe	1492			5.06 MB		Host Process for Windows Ser...
svchost.exe	1528	0.02		1.38 MB		Host Process for Windows Ser...
svchost.exe	1544			6.25 MB		Host Process for Windows Ser...
svchost.exe	1608			2.81 MB		Host Process for Windows Ser...
svchost.exe	1616			1.27 MB		Host Process for Windows Ser...
svchost.exe	1624			1.95 MB		Host Process for Windows Ser...
svchost.exe	1700			2.05 MB		Host Process for Windows Ser...
svchost.exe	1772			2.86 MB		Host Process for Windows Ser...
svchost.exe	1848			5.06 MB		Host Process for Windows Ser...
taskhostw.exe	2408			4.45 MB	EKRAN-3\illia.zaichko	Host Process for Windows Tas...
svchost.exe	1936			2.13 MB		Host Process for Windows Ser...
svchost.exe	1960			1.85 MB		Host Process for Windows Ser...
svchost.exe	2044			2.03 MB		Host Process for Windows Ser...
svchost.exe	1820			1.71 MB		Host Process for Windows Ser...
svchost.exe	2112			1.82 MB		Host Process for Windows Ser...
svchost.exe	2192			2.54 MB		Host Process for Windows Ser...
svchost.exe	1860			6.41 MB	EKRAN-3\illia.zaichko	Host Process for Windows Ser...

CPU Usage: 20.15% | Physical memory: 2.95 GB (73.74%) | Processes: 123

Рис 3.1 Показники продуктивності тестування застосунку на локальному обладнанні.

3.3 Дослідження показників продуктивності на віртуальному обладнанні

Для проведення тестування додатку Svchost.exe та його процесів, було взято віртуальне обладнання, яке розгорнуто за допомогою гіпервізора Virtual Environment.

Proxmox має дуже гнучкий інтерфейс, який надає зручність у використанні віртуальних машин, гіпервізор – надає змогу розгортати віртуальне обладнання без втрати продуктивності обладнання.

Для тестування десктопного застосунку Svchost.exe, було створено віртуальну машину з характеристиками, наведеними в таблиці 3.3.

Характеристики віртуальної машини

CPU	4
Memory	4 GiB
Hard Disk	32 GiB
OS	Windows 11

На створене віртуальне обладнання встановлено десктопний застосунок svchost.exe та запущено усі процеси.

На віртуальному обладнанні виконано тестування:

- Тестування встановлення.
- Тестування сумісності.
- Тестування функціональності.

Під час проведення кожного виду тестування було виконано заміри використання ресурсів віртуального обладнання.

На відмінну від локального обладнання, нам не потрібно використовувати додаткові утиліти для визначення показників використання ресурсів. Гіпервізор Proxmox надає повну інформацію про використання ресурсів певного віртуального комп'ютера.

На рис 3.2 наведено використання ресурсів у вигляді графіків, і можна робити висновки, під час якого виду тестування система використовувала найбільше ресурсів.

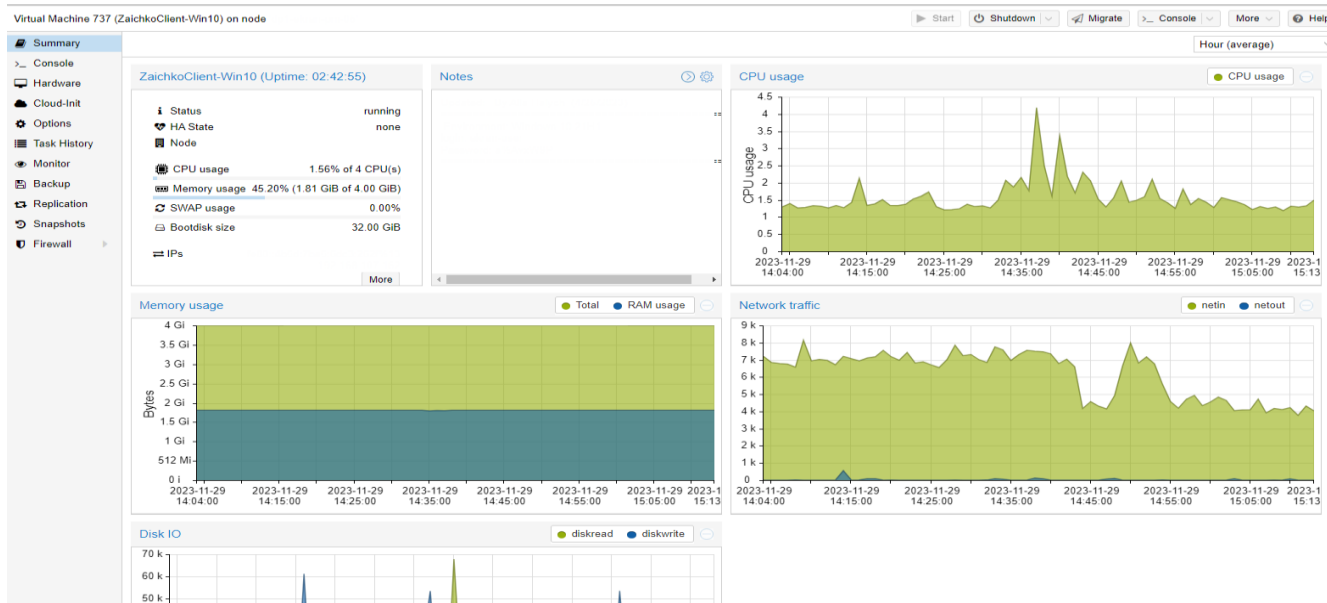


Рис 3.2 - Показники продуктивності тестування застосунку на віртуальному обладнанні.

Також, завдяки зручному інтерфейсу Proxmox, можна спостерігати за використанням ресурсів не тільки на графіках, а і в окремому вікні (рис.3.3).

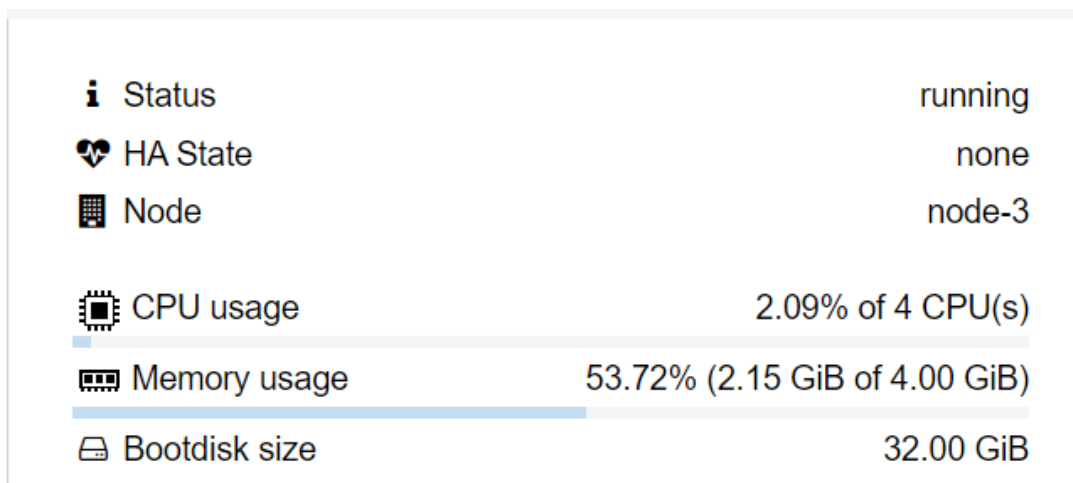


Рис 3.3 - Показник використання ресурсів віртуального комп'ютера

Після виконання тестування десктопного застосунку svchost.exe було визначено наступні показники використання ресурсів на віртуальному обладнанні (табл. 3.4).

Таблиця 3.4

Показники використання ресурсів на віртуальному обладнанні

Критерії продуктивності тестування	Показник використаних ресурсів
CPU usage	1.56 з 4
Memory usage	45.25% (1.81 GiB з 4 GiB)

Дослідивши показники використання ресурсів під час тестування десктопного застосунку на віртуальному обладнанні, можна зробити висновок що ресурсів використовується менше половини.

3.4 Аналіз результатів тестування на локальному та віртуальному обладнанні.

Дослідивши окремо тестування на локальному та окремо на віртуальному обладнанні, було створено порівняльну таблицю 3.5, де визначено скільки ресурсів використано під час проведення тестування десктопного застосунку на локальному на віртуальному обладнанні.

Таблиця 3.5

Оцінка продуктивності тестування

Критерії продуктивності тестування	Локальне обладнання	Віртуальне обладнання
CPU usage	3,96 з 4	1,56 з 4
Memory usage	73.95% (2.95 GiB з 4 GiB)	45.25% (1.81 GiB з 4.00 GiB)

В таблиці 3.5 бачимо велику різницю у використанні ресурсів виконуючи тестування одного і того ж самого десктопного застосунку на різних типах обладнання.

Використання CPU на локальному обладнанні майже досягло свого піку (3,96 з 4), в той час коли на віртуальному обладнанні не використано навіть половини (1,56 з 4). З використанням пам'яті комп'ютера аналогічна ситуація на локальному обладнанні пам'ять використана на 73.95 відсотків, в той час коли на віртуальному обладнанні використано тільки 45.25 відсотка.

Проаналізувавши показники ефективності тестування десктопного застосунку на локальному та віртуальному обладнанні можна зробити висновок, що тестування на віртуальному обладнанні займає менше ресурсів, тим самим є більш ефективнішим ніж проведення тестування десктопного програмного забезпечення на локальному обладнанні.

ВИСНОВКИ

1. Проведено аналіз проблем тестування десктопних застосунків та визначено критерії оцінки процесу тестування десктопного програмного забезпечення.
2. Проведено аналіз існуючих підходів та існуючих традиційних методик тестування десктопного програмного забезпечення.
3. Досліджено особливості використання хмарних технологій процес автоматизації під час тестування десктопного застосунку та визначено ефективність тестування під час використання віртуального обладнання. завдяки цьому дослідженню визначено роль віртуалізації при тестуванні програмного забезпечення.
4. Розроблено метод та засоби тестування десктопного застосунку з використанням хмарних технологій
5. Розроблено схему методу тестування десктопного програмного забезпечення з використанням хмарного кластеру. Показано особливості та основні переваги в порівнянні з традиційною моделлю тестування.
6. Проведено апробацію системи та виконано порівняльний аналіз отриманих результатів

ПЕРЕЛІК ПОСИЛАНЬ

1. Ляшко, С.Ю. (2019). Автоматизоване тестування веб-додатків.
2. Удовенко С.Г. (2019) Використання шаблонів автоматичного тестування в проектах з розробки веб-додатків.
3. Заїчко І.О. Аналіз методів та технологій тестування десктопних застосунків. // Всеукраїнський науково-практичний форум «Так» – Луцьк: ДВНЗ «ДонНТУ», 2023. Подано до публікації
4. Заїчко І.О. Використання хмарного гіпервізору для для тестування десктопного додатку // Всеукраїнський науково-практичний форум «Так» – Луцьк: ДВНЗ «ДонНТУ», 2023. Подано до публікації
5. Кріспін Л. (2011). Гнучке тестування: практичне керівництво для тестувальників ПО та гнучких команд.
6. Virtualization of computer network for modernization and optimization the work of the LAN within IT department// Матеріали та програма МІЖНАРОДНОЇ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ студентів та молодих учених / – Суми: МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ МІЖНАРОДНИЙ УНІВЕРСИТЕТ «АСТАНА», 2022. – С. 84
7. Порошин С.М. (2016) Методологія проведення реп-тестування вебдодатків.
8. Бондаренко О.В. (2020) Дослідження інструментів тестування вебдодатків.
9. Слободян Р. (2020) Автоматизоване тестування веб-додатків шляхомвзаємодії з користувачьким інтерфейсом.
10. Твердохліб І. (2019) Методики тестування безпеки веб-додатку на основі інформації з відкритих джерел
11. Максим Л. Технологія віртуалізації. 2013 - 403 с.
12. Регресійне тестування. URL: <https://qlearning.com.ua/theory/lectures/material/regression-testing-crossbrowser/> (дата звернення 20.10.2023)

13. Модульне тестування. URL:https://msn.khnu.km.ua/pluginfile.php/208290/mod_resource/content/2/%D0%9B%D0%A0%20%E2%84%964.pdf. (дата звернення 22.10.2023).
14. Джек Фолк, Сем Канер, Енг Кек Нгуєн (2001). Тестування програмного забезпечення
15. Заїчко І.О. Аналіз методів та технологій тестування десктопних застосунків. // Всеукраїнський науково-практичний форум «Так» – Луцьк: ДВНЗ «ДонНТУ», 2023, 213-214.
16. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, International Journal of Engineering and Information Systems, 6(2), pp. 59-66
17. . Tvoroshenko, I., & Tkachenko, D. (2020). Mechanisms of image classification based on descriptors of local features.
18. Asaad Ma. Babker, Abd Elgadir A. Altoum, Irina Tvoroshenko, and Vyacheslav Lyashenko (2019) Information Technologies of the Processing of the Spaces of the States of a Complex Biophysical Object in the Intellectual Medical System HEALTH, International Journal of Advanced Trends in Computer Science and Engineering, 8(6), pp. 3221-3227.
19. Заїчко І.О. Використання хмарного гіпервізору для для тестування десктопного додатку // Всеукраїнський науково-практичний форум «Так» – Луцьк: ДВНЗ «ДонНТУ», 2023, 348.
20. Кучеренко, Є. І., Творошенко, І. С., Анопрієнко, Т. В. (2016) Моделювання та оцінювання станів складних об'єктів із застосуванням формальної логіки. Системи обробки інформації, (2), 72-82.
21. . Творошенко, І. С. (2018). Особливості застосування сучасних принципів штучного інтелекту до розробки ефективних механізмів моделювання складних систем. Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland, 110-121.
22. Полубехін А. (2022) Вивчення методів тестування десктопних застосунків, Abstracts of XXXVI International Scientific and Practical Conference «The

main prospects for the development of science in modern life» (September 13 – 16, 2022).
Warsaw, Poland, pp. 333-366

23. Регресійне тестування. URL: <https://qlearning.com.ua/theory/lectures/material/regression-testing-crossbrowser/> (дата звернення 02.10.2022).

24. Дубовік Я.С. (2019) Необхідність використання сучасних технологій моделювання бізнес-процесів.

25. Бродський Ю.Б. (2018) Методологічні аспекти структурнофункціонального програмування в економіці.

26. Слободян Р. (2020) Автоматизоване тестування веб-додатків шляхомвзаємодії з користувацьким інтерфейсом

27. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень. Сучасні інформаційні системи, 6 (3), С. 5-12.

28. Matarneh Rami, Tvoroshenko Irina, and Lyashenko Vyacheslav (2019) Improving Fuzzy Network Models For the Analysis of Dynamic Interacting Processes in the State Space, International Journal of Recent Technology and Engineering, 8(4), pp. 1687-1693.

29. Harley Stagner Pro Hyper-V. (Apress 2009), 2009, - 401 с.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО -
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Магістерська робота

«Підвищення ефективності тестування десктопних застосунків шляхом використання хмарних технологій»

Виконав: студент групи ПДМ-61 Заїчко Ілля Олегович

Керівник: д.т.н., професор, професор кафедри ПЗ Ільїн Олег Юрійович

Київ - 2024

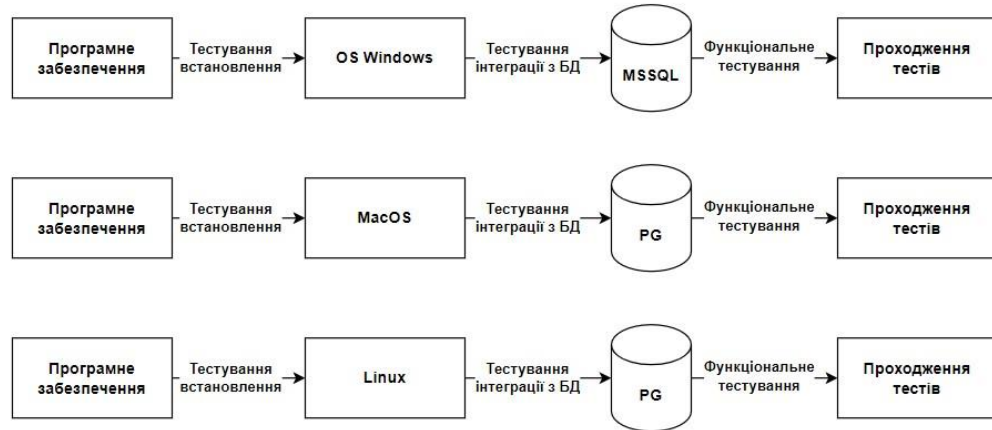
МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: підвищення ефективності тестування десктопних застосунків за рахунок використання віртуалізації, яка базується на хмарних технологіях .

Об'єкт дослідження: процес тестування десктопного застосунку .

Предмет дослідження: методи та технології тестування десктопних застосунків .

ТРАДИЦІЙНА СХЕМА ТЕСТУВАННЯ ДЕСКТОПНИХ ЗАСТОСУНКІВ



3

КРИТЕРІЇ ОЦІНЮВАННЯ ПРОЦЕСУ ТЕСТУВАННЯ ДЕСКТОПНИХ ЗАСТОСУНКІВ

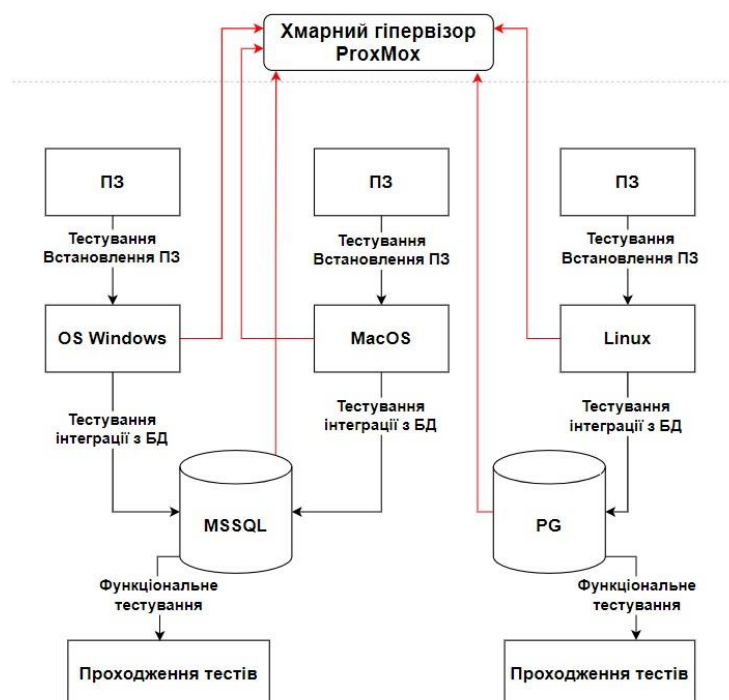
1. Функціональність.
2. Надійність.
3. Кросплатформність.
4. Продуктивність.
5. Переносимість.
6. Співіснування.
7. Безпека.

ХАРАКТЕРИСТИКИ ІСНУЮЧИХ ІНСТРУМЕНТІВ ВІРТУАЛІЗАЦІЇ

Назва інструменту віртуалізації	Характеристики гіпервізору	Недоліки
VMware Workstation	Дозволяє запустити на комп'ютері декілька віртуальних операційних систем одночасно. Кожна віртуальна машина може виконувати свою власну операційну систему	Можливість використання тільки на платній основі
VirtualBox	Дозволяє створення ізольованих віртуальних машин з високою продуктивністю	Не має можливості збереження через певні проміжки часу, це зменшує надійність роботи з ВМ
Hyper-V	Дає змогу створювати і запускати версію програмного забезпечення комп'ютера, яка називається віртуальною машиною.	При використанні з ОС Windows необхідна серверна ліцензія Microsoft

5

СХЕМА МЕТОДУ ТЕСТУВАННЯ ДЕСКТОПНИХ ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ ХМАРНОГО ГІПЕРВІЗОРУ



6

ПОРІВНЯННЯ ПРОДУКТИВНОСТІ ТЕСТУВАННЯ НА ЛОКАЛЬНОМУ ТА ВІРТУАЛЬНОМУ ОБЛАДНАННІ

Name	PID	CPU	I/O total	Private b...	User name	Description
RuntimeBroker.exe	6176	0.02		5.02 MB	EKRAN-3\ilia.zaichko	Runtime Broker
RuntimeBroker.exe	6332			6.74 MB	EKRAN-3\ilia.zaichko	Runtime Broker
RuntimeBroker.exe	6772			5.54 MB	EKRAN-3\ilia.zaichko	Runtime Broker
smartscreen.exe	6956			8.05 MB	EKRAN-3\ilia.zaichko	Windows Defender SmartScree...
ApplicationFrame...	5952			5.45 MB	EKRAN-3\ilia.zaichko	Application Frame Host
dllhost.exe	7020			1.78 MB	EKRAN-3\ilia.zaichko	COM Surrogate
dllhost.exe	8600			3.53 MB	EKRAN-3\ilia.zaichko	COM Surrogate
svchost.exe	932	0.03		5.97 MB		Host Process for Windows Ser...
svchost.exe	976			2.88 MB		Host Process for Windows Ser...
svchost.exe	760	2.50	1.39 kB/s	58.55 MB		Host Process for Windows Ser...
rdpclip.exe	5004	0.10		9.41 MB	EKRAN-3\ilia.zaichko	RDP Clipboard Monitor
svchost.exe	828			2.15 MB		Host Process for Windows Ser...
svchost.exe	1152			1.74 MB		Host Process for Windows Ser...
svchost.exe	1160			1.75 MB		Host Process for Windows Ser...
svchost.exe	1168			3 MB		Host Process for Windows Ser...
svchost.exe	1260			2.12 MB		Host Process for Windows Ser...
svchost.exe	1316	0.03		3.46 MB		Host Process for Windows Ser...
svchost.exe	1348			12.73 MB		Host Process for Windows Ser...
svchost.exe	1452			12.22 MB		Host Process for Windows Ser...
svchost.exe	1492			5.06 MB		Host Process for Windows Ser...
svchost.exe	1528	0.02		1.38 MB		Host Process for Windows Ser...
svchost.exe	1544			6.25 MB		Host Process for Windows Ser...
svchost.exe	1608			2.81 MB		Host Process for Windows Ser...
svchost.exe	1616			1.27 MB		Host Process for Windows Ser...
svchost.exe	1624			1.95 MB		Host Process for Windows Ser...
svchost.exe	1700			2.05 MB		Host Process for Windows Ser...
svchost.exe	1772			2.86 MB		Host Process for Windows Ser...
svchost.exe	1848			5.06 MB		Host Process for Windows Ser...
taskhost.exe	2408			4.45 MB	EKRAN-3\ilia.zaichko	Host Process for Windows Tas...
svchost.exe	1936			2.13 MB		Host Process for Windows Ser...
svchost.exe	1960			1.85 MB		Host Process for Windows Ser...
svchost.exe	2044			2.03 MB		Host Process for Windows Ser...
svchost.exe	1820			1.71 MB		Host Process for Windows Ser...
svchost.exe	2112			1.82 MB		Host Process for Windows Ser...
svchost.exe	2192			2.54 MB		Host Process for Windows Ser...

7

ПОРІВНЯННЯ ПРОДУКТИВНОСТІ ТЕСТУВАННЯ НА ЛОКАЛЬНОМУ ТА ВІРТУАЛЬНОМУ ОБЛАДНАННІ



8

ПОРІВНЯЛЬНИЙ АНАЛІЗ ХАРАКТЕРИСТИК ВІРТУАЛЬНОГО ТА ЛОКАЛЬНОГО ТЕСТУВАННЯ

Критерії продуктивності тестування	Локальне обладнання	Віртуальне обладнання
CPU usage	3,96 з 4	1,56 з 4
Memory usage	73.95% (2.95 GiB з 4 GiB)	45.25% (1.81 GiB з 4.00 GiB)
Hard Disk size	32 GiB	32 GiB
OS	Windows 11	Windows 11, MacOS, Linux

9

ВИСНОВКИ

1. Проведено аналіз проблем тестування десктопних застосунків та визначено критерії оцінки процесу тестування десктопного програмного забезпечення.
2. Проведено аналіз існуючих підходів та існуючих традиційних методик тестування десктопного програмного забезпечення.
3. Досліджено особливості використання хмарних технологій процес автоматизації під час тестування десктопного застосунку та визначено ефективність тестування під час використання віртуального обладнання. завдяки цьому дослідженню визначено роль віртуалізації при тестуванні програмного забезпечення.
4. Розроблено метод та засоби тестування десктопного застосунку з використанням хмарних технологій
5. Розроблено схему методу тестування десктопного програмного забезпечення з використанням хмарного кластеру. Показано особливості та основні переваги в порівнянні з традиційною моделлю тестування.
6. Проведено апробацію системи та виконано порівняльний аналіз отриманих результатів

10

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Заїчко І.О. Аналіз методів та технологій тестування десктопних застосунків. // Всеукраїнський науково-практичний форум «Так» – Луцьк: ДВНЗ «ДонНТУ», 2023. 213-214
2. Заїчко І.О. Використання хмарного гіпервізору для тестування десктопного додатку // Всеукраїнський науково-практичний форум «Так» – Луцьк: ДВНЗ «ДонНТУ», 2023. 348

ДЯКУЮ ЗА УВАГУ!