ДОДАТОК А

Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії

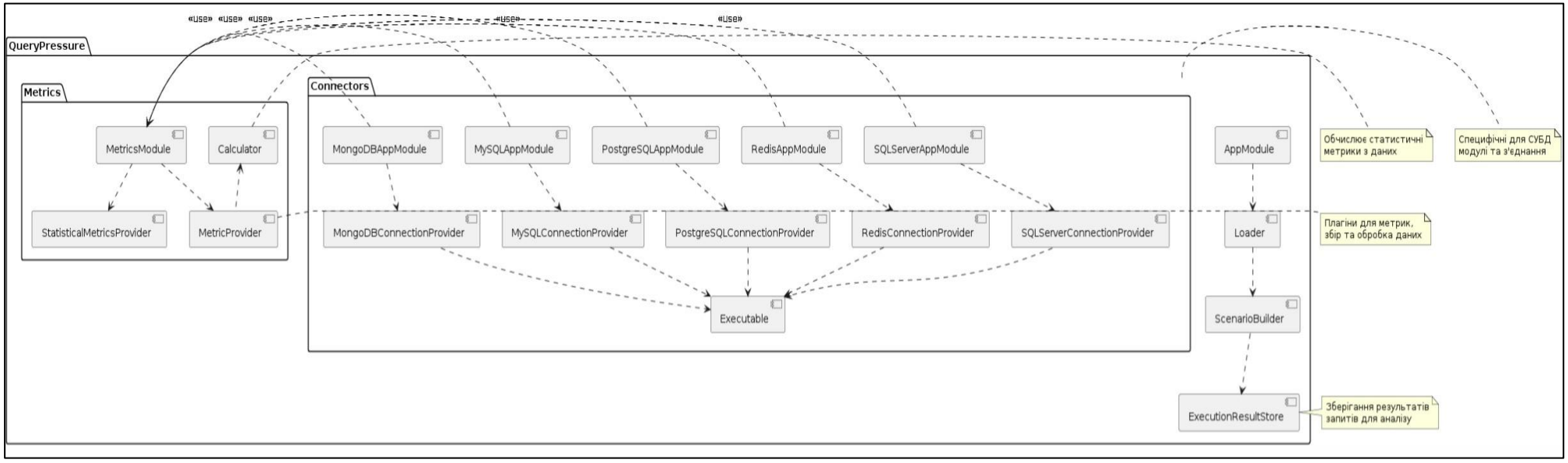
10. Optimization Factors in Modeling and Testing Hardware and Semiconductor Defects by Dynamic Discrete Event Simulation \ Arabian, J. H., Видавництво: ХНУРЕ, 2009р., URL: https://openarchive.nure.ua/entities/publication/be918fba-8755- 4d34-be6d-fc1464089d77 (дата звернення: 20.04.2024).

11. Falatiuk H., Shirokopetleva M., Dudar Z. Investigation of Architecture and Technology Stack for e-Archive System. 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8-11October 2019, URL: https://doi.org/10.1109/picst47496.2019.9061407 (дата звернення: 05.05.2024).

12. DATA EXCHANGE MODEL IN THE INTERNET OF THINGS CONCEPT / I. Afanasieva et al. Telecommunications and Radio Engineering. 2019. Vol. 78,no. 10.P. 869–878. URL: https://doi.org/10.1615/telecomradeng.v78.i10.30 (date of access: 11.05.2024).

ДОДАТОК Б

Зображення структури класів системи



Б.1 – Структура класів системи

ДОДАТОК В

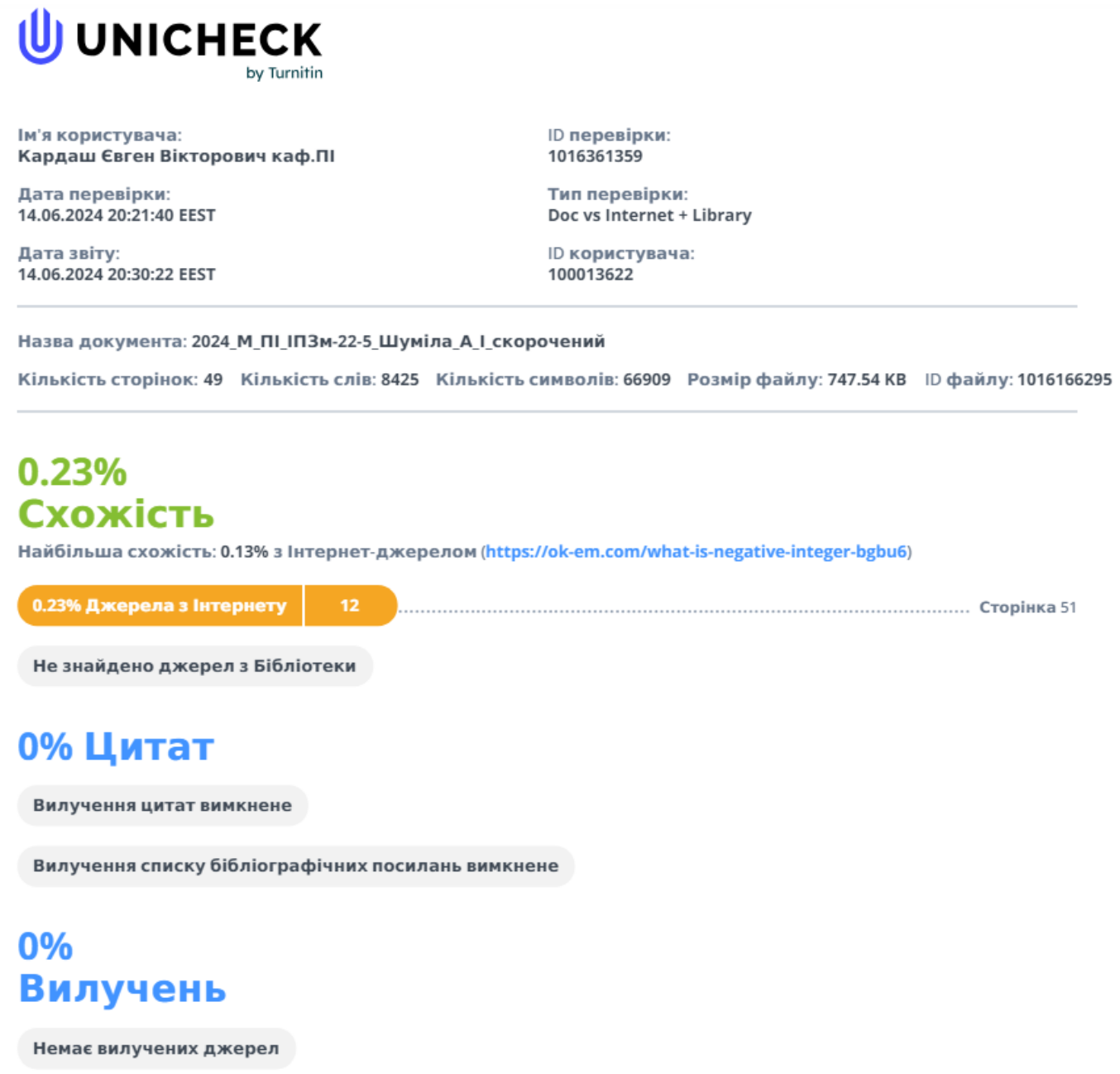
Код методу збору статистики для відхилення

```
      using System.Text;
      using Perfolizer.Horology;
      using
Perfolizer.Mathematics.Histograms;
using QPressure.App.Console;
      using QPressure.App.Interfaces;

      namespace QPressure.Metrics.App.Formatters;
      public class HistogramConsoleMetricFormatter :
IConsoleMetricFormatter
      {
        private readonly ConsoleOptions
  _consoleOptions; private readonly IDictionary<string,
  string> _locale;
      public HistogramConsoleMetricFormatter(ConsoleOptions
consoleOptions,IResourceManager resourceManager)
        {
          _consoleOptions = consoleOptions;
       _locale =
resourceManager.GetResources(consoleOptions.CultureInfo.Name,
ResourceFormat.Plain);
        }
        public uint Priority => 1;
        public bool CanFormat(string metricName, object metricValue)
        {
          return metricValue is Histogram;
        }
        public string Format(string metricName, object metricValue)
        {
          var value = (Histogram)metricValue;
      if (!_locale.TryGetValue($"metrics.{metricName}.title", out
varmetricDisplayName))
          {
            metricDisplayName = metricName;
          }
          var sb = new StringBuilder();
          var header = $"-------------------- {metricDisplayName} ---------
    ------
      ____"
          .Replace('-',
    _consoleOptions.RowSeparatorChar);
    sb.AppendLine(header);
    sb.AppendLine(value.ToString(x =>
          {
            var timeInterval = TimeInterval.FromNanoseconds(x);
            return timeInterval.ToString(_consoleOptions.CultureInfo);
          }));
          sb.Append(string.Empty.PadRight(header.Length,
      _consoleOptions.RowSeparatorChar));
          return sb.ToString();
        }}
```

ДОДАТОК Г

Звіт результатів перевірки кваліфікаційної роботи на унікальність тексту



UNICHECK
by Turnitin

Ім'я користувача:
Кардаш Євген Вікторович каф.ПІ

ID перевірки:
1016361359

Дата перевірки:
14.06.2024 20:21:40 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
14.06.2024 20:30:22 EEST

ID користувача:
100013622

Назва документа: 2024_М_ПІ_ІПЗм-22-5_Шуміла_А_І_скорочений

Кількість сторінок: 49 Кількість слів: 8425 Кількість символів: 66909 Розмір файлу: 747.54 KB ID файлу: 1016166295

0.23%
Схожість

Найбільша схожість: 0.13% з Інтернет-джерелом (https://ok-em.com/what-is-negative-integer-bgbu6)

0.23% Джерела з Інтернету 12 ..... Сторінка 51

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

ДОДАТОК Д

Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015

Експертний висновок результатів перевірки кваліфікаційної роботи

студент (посада) | програмної інженерії (кафедра) | ІПЗм-22-5 (група)

Шуміла Андрій Іванович

(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні познаки	
	7.15 Додатки	

зауважень немає

Експерт _____ (підпис) Олена ОЛІЙНИК (прізвище, ініціали)

17.06.2024

ДОДАТОК Е

Апробація результатів роботи

УДК 004.415.53

ДОСЛІДЖЕННЯ МЕТОДІВ НАВАНТАЖУВАЛЬНОГО ТЕСТУВАННЯ ДЛЯ ПЕРЕВІРКИ ШВИДКОСТІ ЗАПИТІВ ДО БД

Шуміла А.І.
Науковий керівник – к.т.н., доцент Бабій А.С.
Харківський національний університет радіоелектроніки, каф. ПІ,
м. Харків, Україна
e-mail: andrii.shumila@nure.ua

This work studies load testing methods to assess database query speed, addressing the need for effective testing in modern information systems. It analyzes existing techniques, evaluates their pros and cons, and proposes optimal methods for measuring query speed. Experiments validate these methods and provide practical recommendations. The findings help improve the efficiency and reliability of information systems by identifying performance bottlenecks and optimizing database performance, offering valuable insights for IT professionals.

Мета дослідження полягає в аналізі існуючих підходів та методів навантажувального тестування для виявлення найоптимальнішого рішення оцінки швидкодії запитів до баз даних.

Основна проблема, яку має вирішити дослідження, полягає у визначенні ефективності обробки запитів в реальних умовах експлуатації систем. Важливість цієї проблеми виходить із зростання обсягів даних і збільшення вимог до швидкості їх обробки в сучасних інформаційних системах. Вирішення цієї проблеми включає в себе розробку і аплікацію методів тестування, які дозволяють ефективно симулювати реальні навантаження, визначати потенційні "вузькі місця" і оптимізувати роботу системи для забезпечення високої продуктивності.

Також важливою є задача оптимізації запитів до баз даних, яка передбачає розробку методів для покращення часу реакції на запити, що включає аналіз існуючих запитів, оптимізацію SQL-команд і структур даних, а також вдосконалення архітектури баз даних.

Вибір метода оцінки продуктивності баз даних у нашому дослідженні був обґрунтований необхідністю забезпечення всебічного аналізу швидкодії запитів, що включає виявлення не тільки загальних трендів, але й відхилень та потенційних проблем у відповідях системи. Далі розгляньмо підходи.

Процентилі (90-й, 95-й, та 99-й): процентилі використовуються для визначення часів відповіді, які перевищують звичайні, але не є екстремальними, а також для аналізу "хвостів" розподілу, які містять потенційні проблемні зони. Це дозволяє оцінити реальну продуктивність під час пікових навантажень, а не тільки при середніх умовах.

Середній час відгуку: цей показник використовується для оцінки загальної продуктивності системи. Він дає змогу визначити базовий рівень ефективності обробки запитів, що є корисним для порівняльного аналізу та планування масштабування.

Медіана: ммедіанний час відгуку вказує на те, який час обробки є типовим для половини запитів, ігноруючи аномальні випадки. Це корисно для виявлення стандартної продуктивності системи без упереджень через виключні значення.

Аналіз нормального розподілу: цей аналіз дозволяє зрозуміти, як розподіляються часи відгуку, виявляючи відхилення від норми, що можуть вказувати на наявність технічних чи програмних проблем.

Застосування методу нормального розподілу також дає змогу виконати прогнозування та моніторинг ефективності. Враховуючи динаміку розподілу в часі, можна спрогнозувати, як система поведе себе під впливом різних умов навантаження та оптимізації.

Також є можливість виявлення аномалій чи викидів в розподілі часів відповіді. Поєднуючи цю інформацію зі статистикою викидів, можна легко визначити та вивчати незвичайні патерни, які можуть вказувати на проблеми в системі або неочікувані зміни.

Застосування методу нормального розподілу також дає змогу виконати прогнозування та моніторинг ефективності. Враховуючи динаміку розподілу в часі, можна спрогнозувати, як система поведе себе під впливом різних умов навантаження та оптимізації.

Виходячи із зазначеного – до нашого методу нормального розподілу варто додати метод міжкварального розмаху. Міжквартальний розмах (IQR) – це міра розмаху, яка визначається як різниця між верхнім (Q3) та нижнім (Q1) квартилями набору даних.

Цей показник дозволяє виявити різницю між центральними 50% значень в наборі даних, і він широко використовується для виявлення викидів та визначення ступеня варіабельності в даних[1].

Застосування IQR в нашій виборці, може бути вкрай корисним. IQR дозволяє визначити "типовий" діапазон значень, які лежать в межах між Q1 та Q3. Це дозволяє оцінити ступінь стабільності та неперервності швидкісного режиму системи.

IQR може бути використаний для визначення асиметрії розподілу швидкості, але важливо також розглядати відхилення від середнього за допомогою методу нормального розподілу.

Загальна ідея полягає в тому, що IQR дозволяє визначити та відокремити центральні 50% значень у наборі даних, що може бути корисним при виявленні аномалій та змін в швидкодії системи. Поєднання цього методу з методом нормального розподілу надає більш комплексний підхід для розуміння та оптимізації швидкодії запитів.

Хоча метод наведені вище є доволі самостійними, але нам необхідно буде бачити усю картину, усі параметри часу нашої вибірки, тому варто застосувати мультимодальний підхід який буде реалізовано програмно, а візуальна складова буде відображена у консольному режимі.

Список використаних джерел:

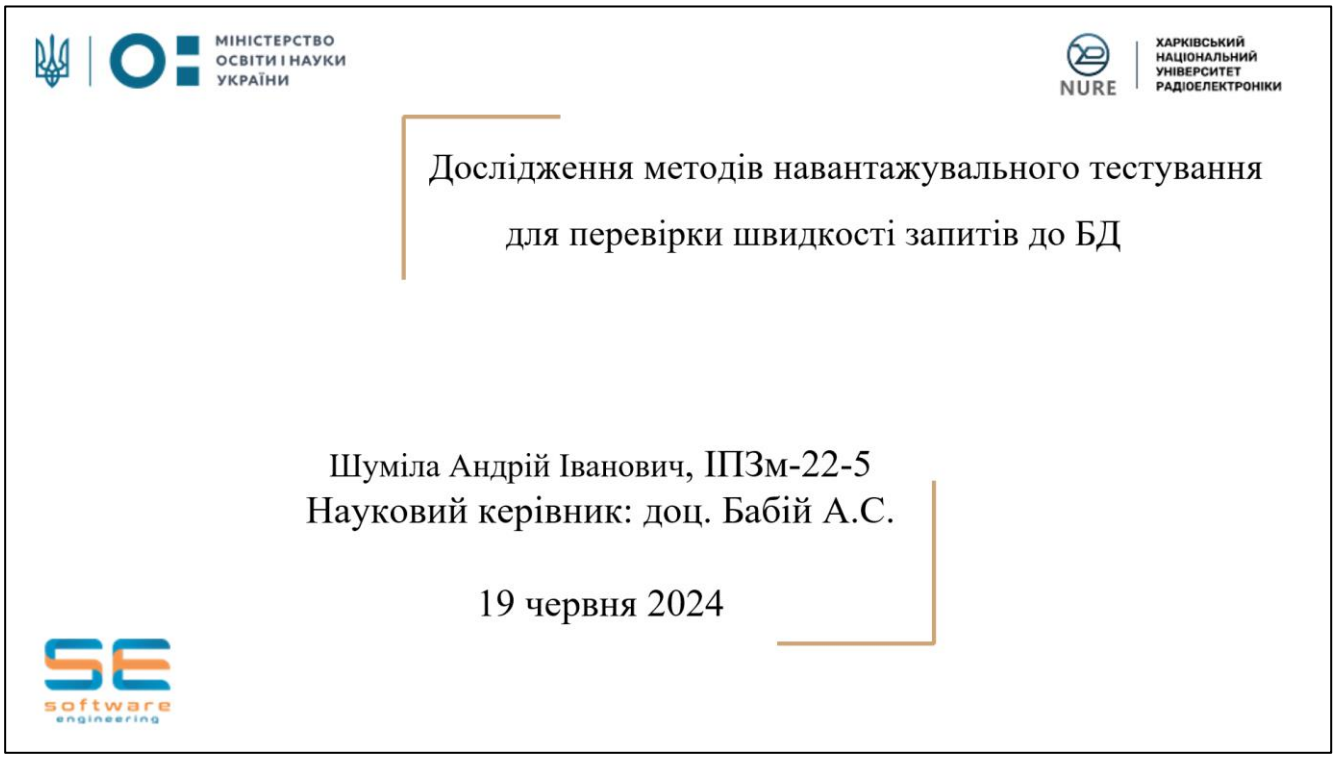
1. Normal Distribution in Statistics (Statistics By Jim) 2018. statisticsbyjim. URL: https://statisticsbyjim.com/basics/normal-distribution/ (дата звернення: 16.02.2024).

2. Woodcock M., Li L. Database Performance Tuning and Optimization. 2nd edition. United States: O'Reilly Media, 2022. 352 p.

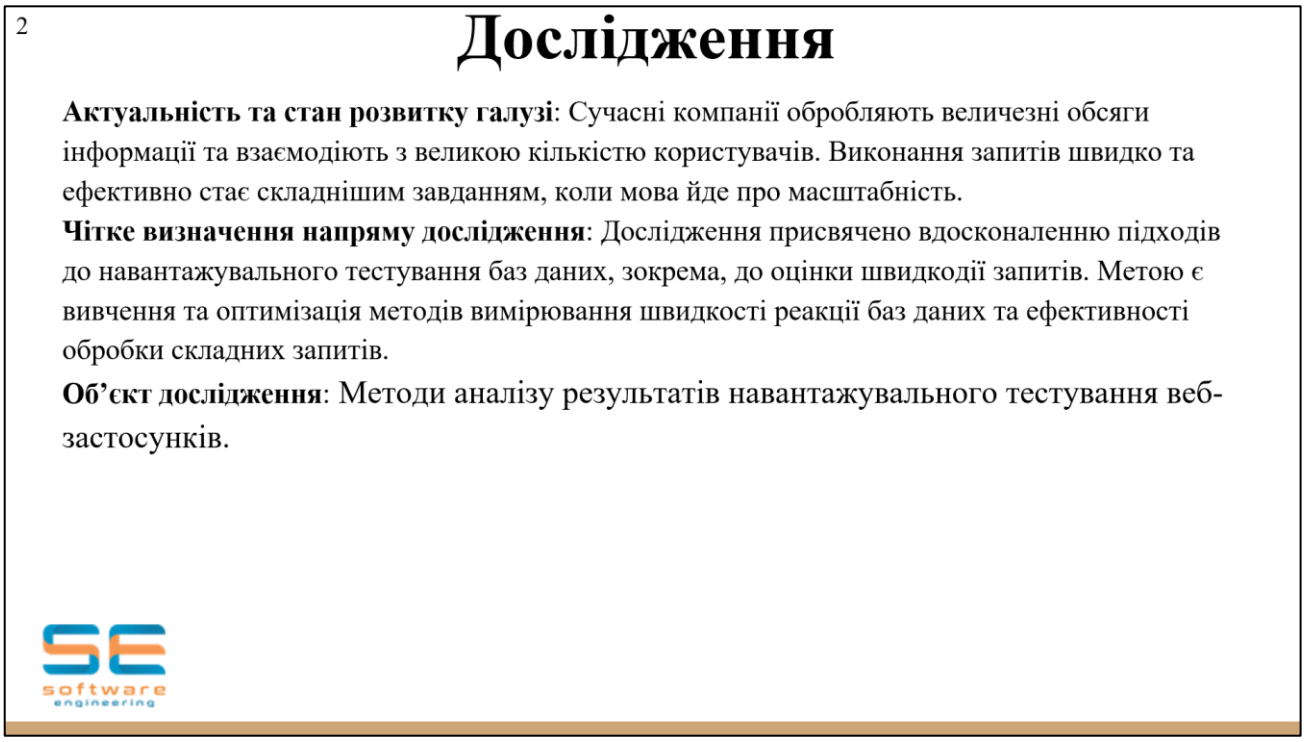
3. Greenwald J. SQL Performance Explained. 3rd edition. Germany: Redgate Publishing, 2021. 298 p.

ДОДАТОК Ж

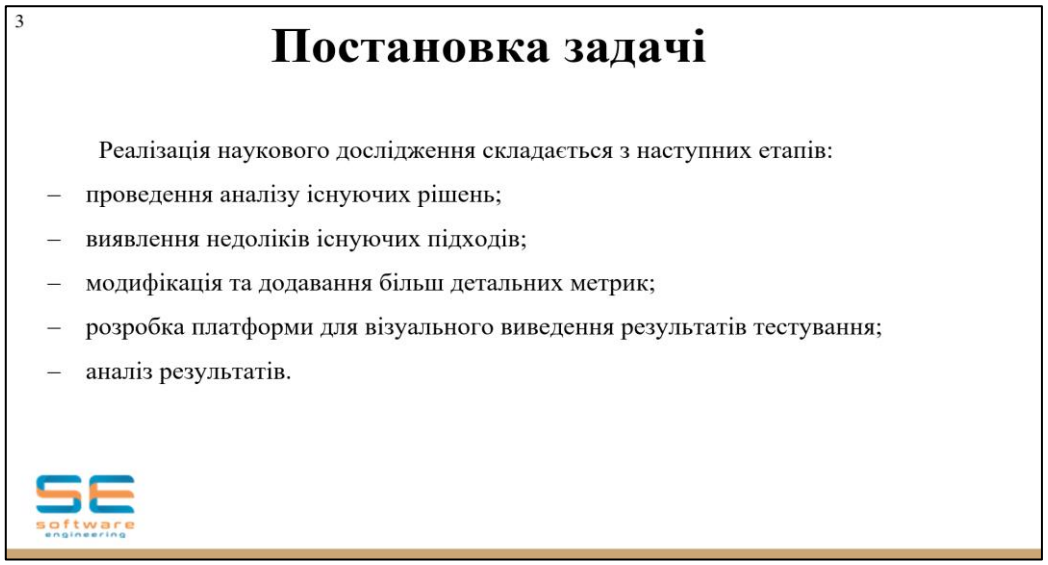
Слайди презентації



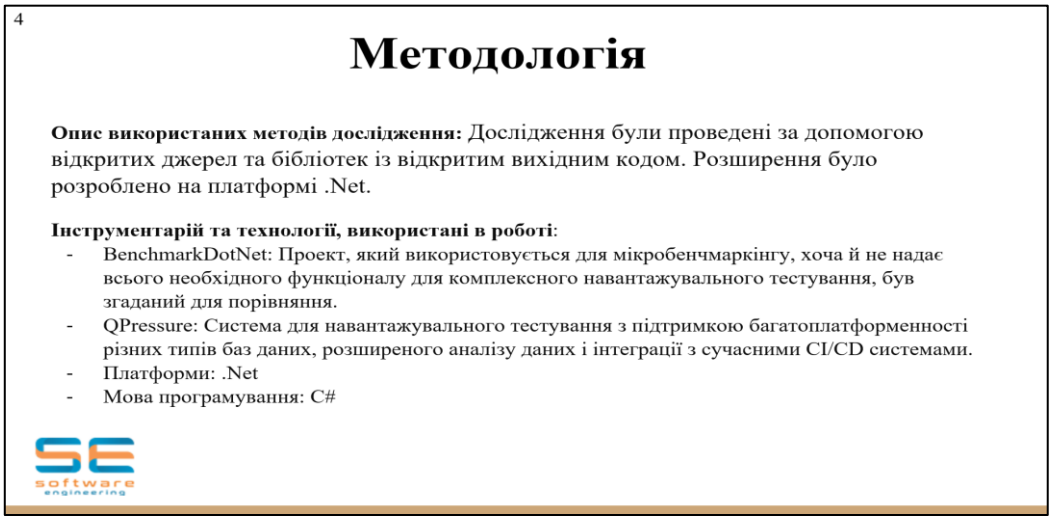
Слайд Ж.1 – Слайд 1



Слайд Ж.2 – Слайд 2

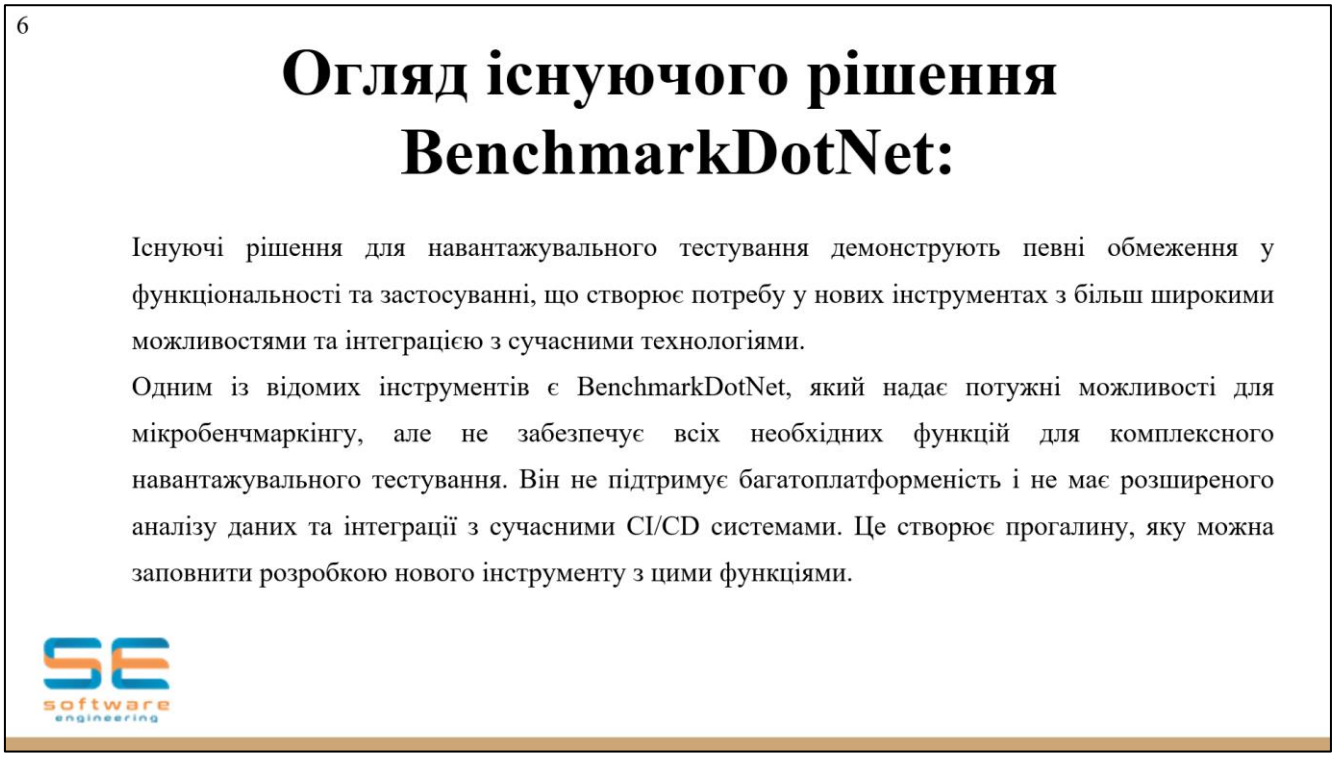


Слайд Ж.3 – Слайд 3

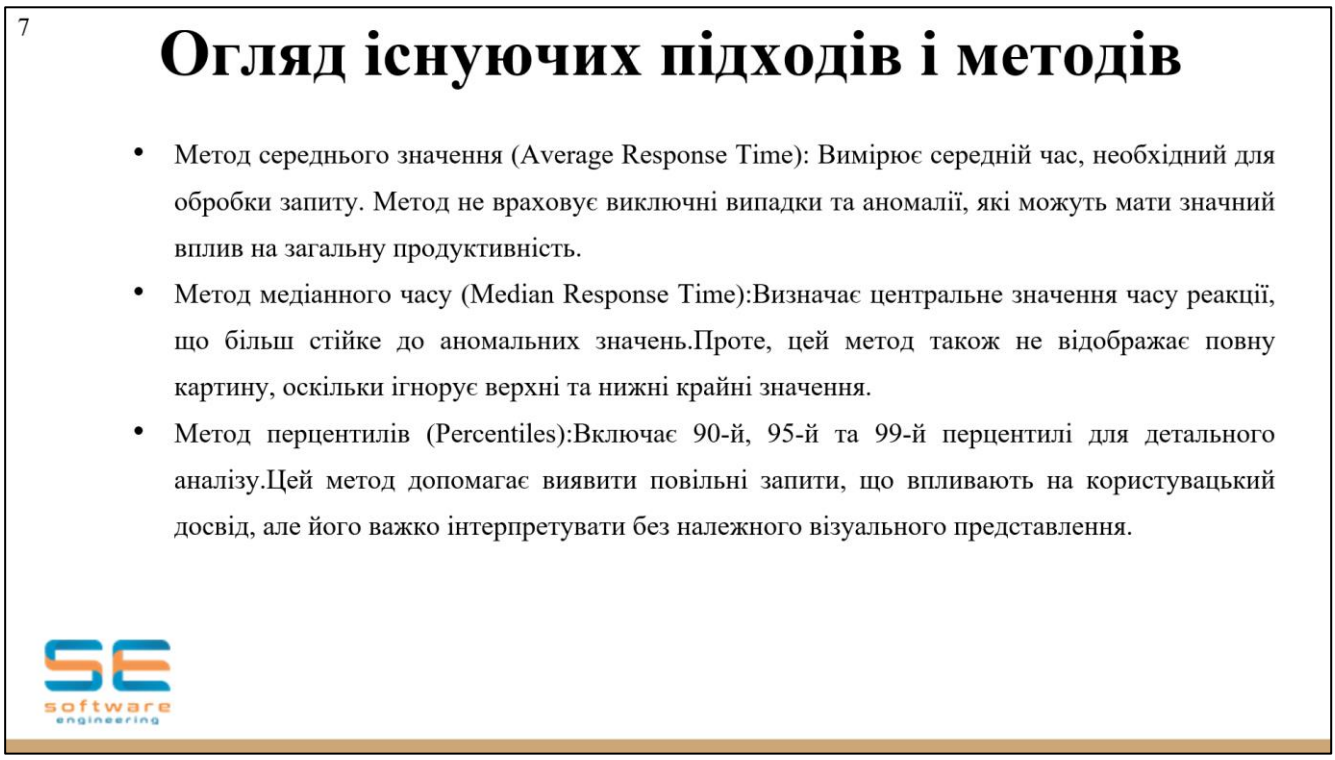


Слайд Ж.4 – Слайд 4

Слайд Ж.5 – Слайд 5



Слайд Ж.6 – Слайд 6

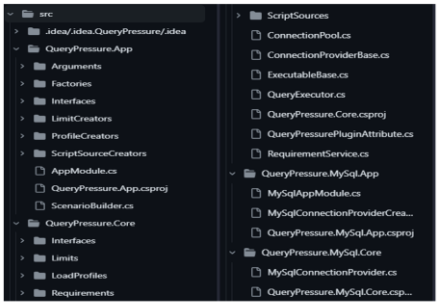


Слайд Ж.7 – Слайд 7

8

Файлова структура розширення

Наша система "QPressure" розроблена як інструмент для проведення навантажувального тестування та аналізу продуктивності баз даних і веб API. Цей інструмент дозволяє інженерам та розробникам здійснювати детальний моніторинг і оптимізацію відповідей систем на різні типи запитів під час різних умов використання.
На малюнку можна побачити основні класи архітектури :



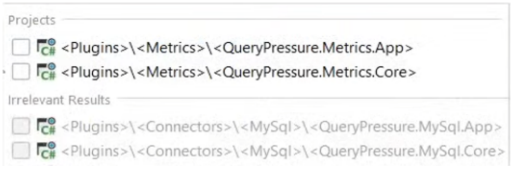
Слайд Ж.8 – Слайд 8

9

Доступ до СУБД

У нашій системі взаємодія з базами даних та використання плагінів для різних СУБД відіграють ключову роль у функціонуванні та гнучкості. Система налаштована таким чином, щоб легко інтегруватися з різними типами баз даних, використовуючи спеціалізовані плагіни.

Також нами було прийнято рішення використовувати плагіни до різних СУБД. Архітектура побудована таким чином, що вона підтримує плагіни для різних СУБД. Це означає, що для кожної СУБД може бути розроблений спеціалізований плагін, який імплементує інтерфейси та класи, необхідні для взаємодії з цією конкретною системою. Плагіни та вибір конфігурації із ними зображено на рисунку:



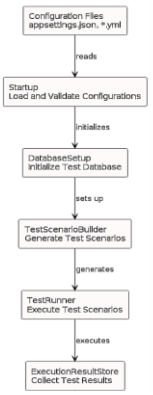
Слайд Ж.9 – Слайд 9

10

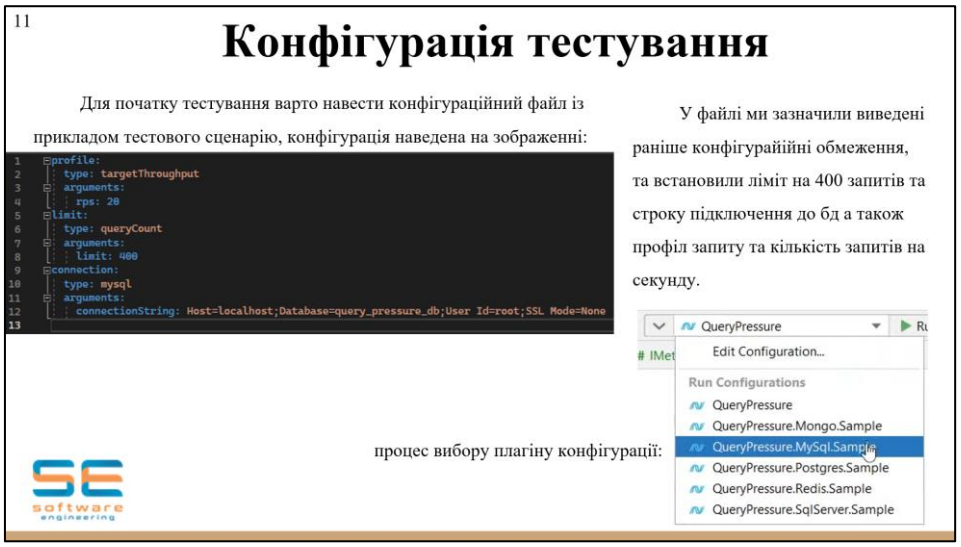
Приклади архітектурних рішень

На зображенні структура роботи тесту, від вводу конфігураційного файлу – до отримання результатів метрик. Далі розглянемо складові та їх функціонал:

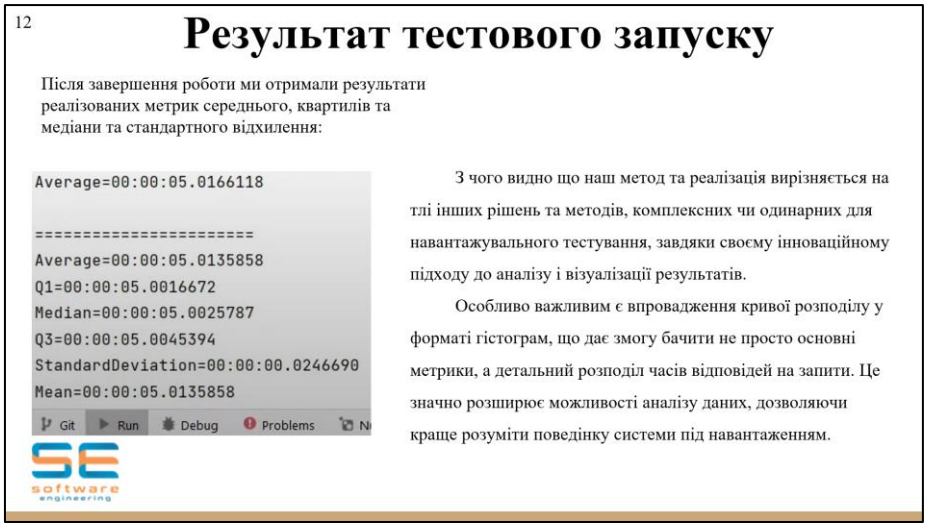
- введення даних та налаштувань тесту: файли конфігурації: appsettings.json, sample.mysql.yml.
- завантаження конфігурації: клас: loader або startup для ініціалізації та встановлення параметрів тестування;
- ініціалізація тестового середовища: файли: docker-compose.yml для налаштування середовищ.
- створення тестових сценаріїв:файли: querytemplates.sql (гіпотетична назва) для sql скриптів, запуск тестів.
- збір результатів:клас: executionresultstore для зберігання результатів виконання;



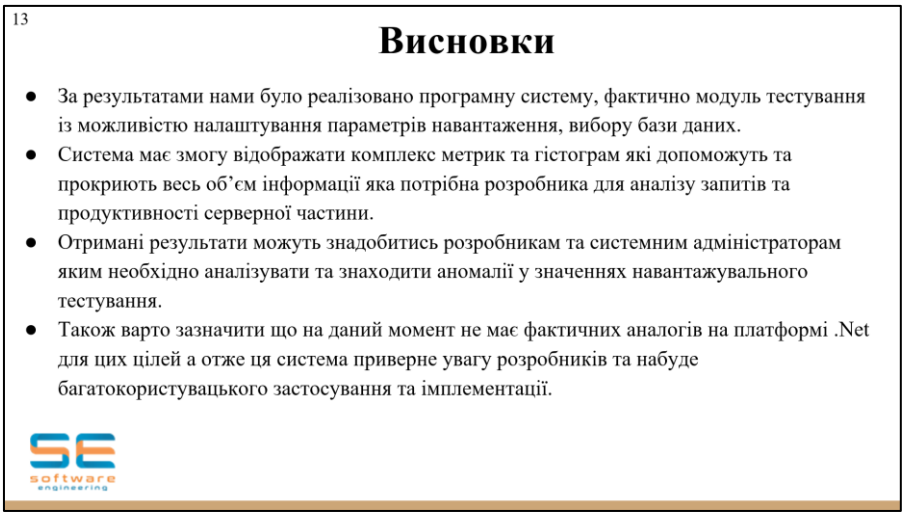
Слайд Ж.10 – Слайд 10



Слайд Ж.11 – Слайд 11



Слайд Ж.12 – Слайд 12



Слайд Ж.13 – Слайд 13