Міністерство освіти і науки України
Національний технічний університет України „КПІ”
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки
інформації та управління

ЗВІТ

з лабораторної роботи № 1
дисципліни
“ТЕХНОЛОГІЇ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ В УМОВАХ ВЕЛИКИХ ДАНИХ”
на тему:

„Паралельні обчислення в моделі зі спільною пам’яттю”

Виконали студенти

- *ІП-01мн Семченко Андрій*
- *ІП-01мн Кошовець Євген*
- *ІТ-01мн Васюк Владислав*
- *ІТ-01мн Минзар Богдан*

(№ групи, прізвище, ім’я, по батькові)

Прийняв *доц. Жереб К. А.*
(прізвище, ім’я, по батькові)

Київ 2021

ЗМІСТ

1 ПОСТАНОВКА ЗАДАЧІ

Для обраної задачі необхідно реалізувати послідовну (однопоточну) реалізацію, а також мультипоточну реалізацію зі спільною пам’яттю.

В якості задачі було обрано проблему пошуку вразливостей, що є у публічних проектах. В якості сховища публічних проектів програмного забезпечення було обрано платформу Github.

Ідея полягає у тому, щоб шляхом аналізу вмісту публічного репозиторію знайти потенційні вразливості, що дозволяють втрутитись у роботу програмного забезпечення. Причому вразливості не тільки у самій реалізації програмного забезпечення, але і вразливості, спричинені недбалим обігом sensitive data, наприклад:

- Зберігання ключів доступу у файлах, що відстежуються VCS
- Зберігання ключів доступу прямо у тексті програмного забезпечення
- Зберігання бекапів у файлах, що відстежуються VCS

Зберігання sensitive data у файлах, що відстежуються системою контролю версій призводить до того, що будь хто може завантажити ці дані і використати для втручання у роботу програмного забезпечення, викрадення даних користувачів, тощо.

2 ВИКОРИСТАНІ БІБЛІОТЕКИ, ФРЕЙМВОРКИ

Додаток було розроблено мовою програмування Java. Для роботи з Github було використано бібліотеку JGit [1].

Для роботи з файловою системою було використано бібліотеку Apache Commons IO [2].

Робота з потоками була організована стандартними засобами мови програмування Java:

- Callable interface
- Thread Executor Services

Для логування роботи програми була використана бібліотека Log4j2. Робота з JSON організована засобами бібліотеки Jackson Json.

Для спрощення читабельності коду застосовано бібліотеку Lombok, що автоматично генерує boilerplate код.

3 ОПИС РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Загальна концепція

Програмне забезпечення завантажує вміст публічного Github-репозиторію у тимчасову директорію. Далі, кожен файл даної директорії аналізується на предмет наявності в ньому певних паттернів, що можуть свідчити про вразливість. Сам паттерн описується за допомогою регулярного виразу.

Паралельна обробка організована наступним чином. Кожен потік завантажує вміст призначеного йому репозиторію та аналізує його. У випадку, коли потік завершив обробку одного репозиторію, він переходить до обробки наступного.

Можна було також організувати роботу таким чином:

1) Спочатку виконати паралельне завантаження вмісту всіх репозиторіїв, що необхідно проаналізувати
2) Потім паралельно обробляти всі отримані файли

Але така реалізація здалася нам недоцільною, оскільки час обробки кожного файлу досить невеликий, а накладні витрати на перемикання контексту між потоками, запуск/зупинення потоку здалися нам досить значними, в порівнянні с витратами на файловий ввід-вивід.

3.2 Деталі роботи програмного забезпечення

Робота програмного забезпечення організована наступним чином. При запуску програми на основі command line аргументів створюється екземпляр класу AppConfiguration, що зберігає інформацію про параметри роботи програми (такі як кількість потоків, посилання на Github-репозиторії, timeout, тощо).

Далі створюється ExecutorService, що керує створенням потоків та виконанням на них завдань.

Завдання представлені у вигляді класу RepoAnalysisCallable, що повністю описує роботу з аналізу вмісту репозиторія (включаючи також його завантаження локально на диск).

Завантаження вмісту репозиторію описано в класі RepoDownloader.

Код, що описує аналіз вмісту репозиторію розбито на 2 основні класи:

- ContentAnalyzer – аналіз вмісту файлу
- PathAnalyzer – аналіз шляху до файлу у репозоторії (назва файлу, розширення, тощо)

Для того, щоб забезпечити можливість подальшого розширення функціональності програмного забезпечення було створено інтерфейс LineChecker, який описує перевірку на наявність вразливості у загальному вигляді.

Інформація про деякі види вразливостей описана у вигляді JSON-файлу, вміст якого зчитується при запуску програми.

4 РОЗГОРТАННЯ ТА ЗАПУСК ЗАСТОСУНКУ

4.1 Розгортання застосунку

Для зручності, застосунок було запаковано у docker-образ. Докеризація за допомогою Dockerfile.

Завантажити готовий до використання Docker-образ можна за посиланням.

При запуску необхідно передати наступні cli-аргументи:

1. Кількість потоків
2. Timeout у секундах
3. Шлях, куди потрібно записати файл з результатом аналізу

Приклад команди для запуску контейнера:

```
docker run --rm --mount type="bind,source=C:\Users\asem\Desktop\KPI_SEMESTER_3\parallel\studying-parallel-programming\lab1\target,target=/result" asemchenko/parallel-lab-1 7 8000 /result
```

Для того, щоб зберегти файл з результатом аналізу у host os, а не тільки у container-os, був використаний ключ –mount.

4.2 Вимірювання часу роботи в залежності від кількості потоків

Для вимірювання часу було використано утиліту GNU time, що дозволяє вимірювати час виконання програми використовуючи статистику, отриману з ядра операційної системи.

Дана утиліта дає можливість виміряти:

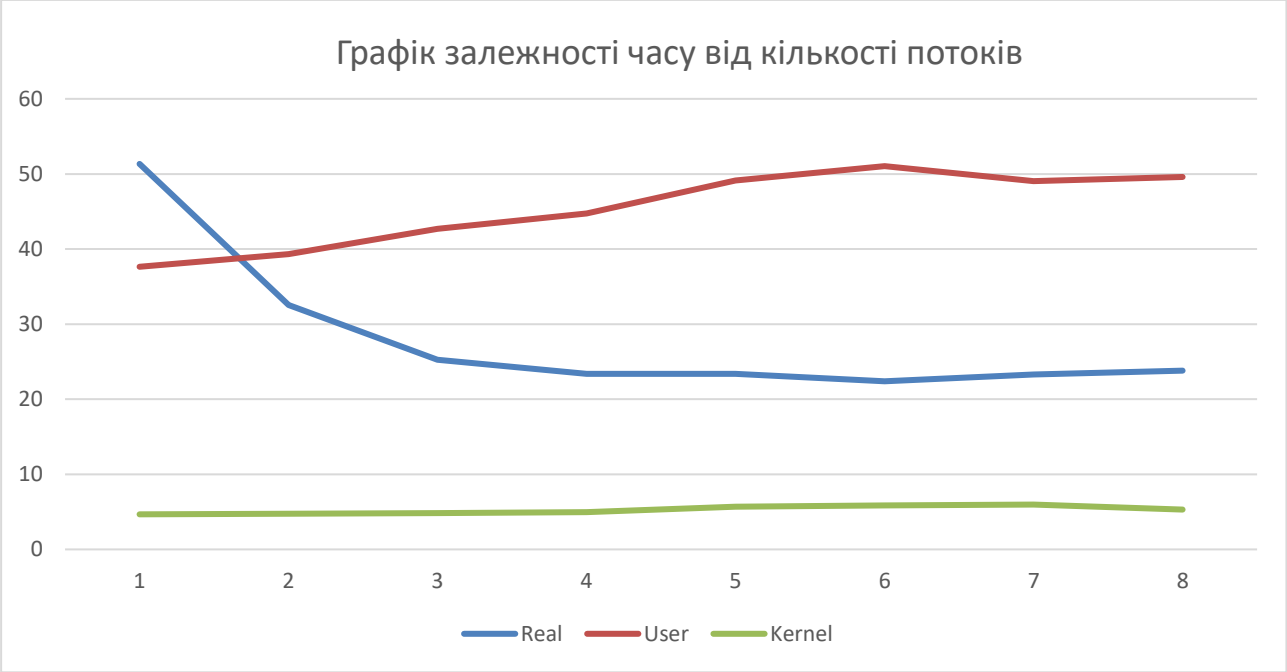
- Реальний час виконання (real)
- Час, проведений у user-space (user)
- Час, проведений у kernel-space (kernel)

Результати проведених вимірів наведено у таблиці нижче.

Кількість потоків	Real	User	Kernel
1	51.34	37.64	4.67
2	32.54	39.34	4.74
3	25.24	42.71	4.85

4	23.36	44.73	4.98
5	23.37	49.15	5.68
6	22.39	51.04	5.84
7	23.31	49.03	5.97
8	23.82	49.59	5.29

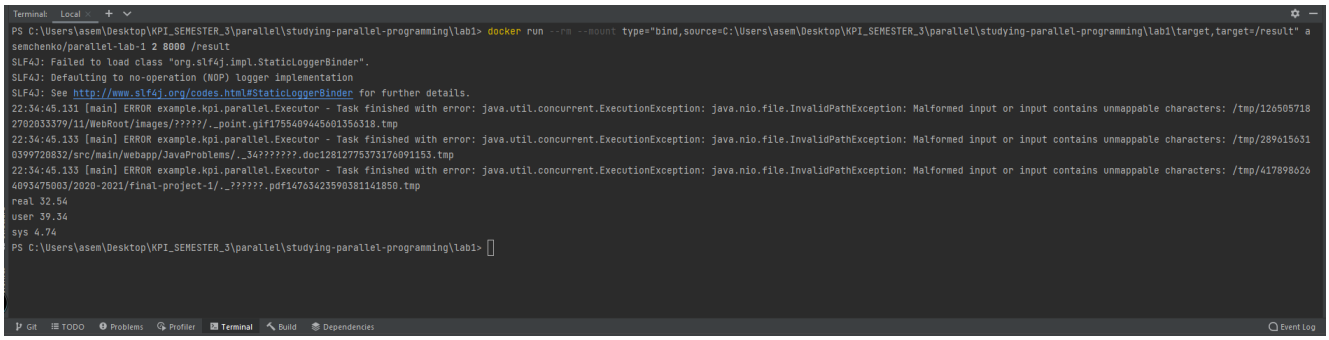
Запуск з більшою кількості потоків вважаємо недоцільним, оскільки видно, що при кількості потоків 4+ значного пришвидшення роботи не спостерігається.



Ймовірно, після деякої кількості потоків програма перестає працювати швидше через принципові обмеження застосованого алгоритму. Очевидно, що пришвидшення, що може бути досягнуте, обмежено розміром послідовної частини алгоритму (закон Амдала це формально показує).

В даному випадку послідовною частною алгоритму є обробка файлів самого репозиторія.

Нижче, для наочності, наведено результат вимірювання часу виконання при 2-ох потоках.



4.3 Оцінка послідовної частини програми

Використаємо закон Амдала, щоб оцінити яку частину склала послідовна частина програми.

$$S_p = \frac{1}{d + \frac{1-d}{p}}$$

Дано:
$p = 4$
$S_p = 2.2$

$$2.2 = \frac{1}{d + \frac{1-d}{4}};$$

$$d + \frac{1-d}{4} = \frac{1}{2.2};$$

$$4d + (1-d) = \frac{4}{2.2};$$

$$3d = \frac{4}{2.2} - 1;$$

$$d = \left(\frac{4}{2.2} - 1\right) / 3;$$

Маємо, що $d = 27\%$

Тобто, послідовна частина використаного алгоритма складає приблизно 27%.

4.4 Знайдені вразливості

Знайдені в результаті роботи вразливості в основному стосувалися настуних причин:

- Зберігання SQL-дампів “under version control”
- Зберігання конфігураційних файлів оболонки командного рядка “under version control”

Повний перелік знайдених вразливостей наведено у додатку 1.

5 ВИСНОВОК

В рамках даної лабораторної роботи було розроблено програмне забезпечення, що проводить аналіз Github-репозиторіїв на наявність в них типових вразливостей.

В рамках реалізованого алгоритму вдалося отримати пришвидшення в порівнянні з однопоточною реалізацією у 2.2 рази.

За приблизними оцінками, послідовна частина використаного алгоритму склала 27%.

Найпопулярнішою, зі знайдених вразливостей є зберігання SQL дампів у системі контролю версій.

6 ПОСИЛАННЯ

[1] https://www.eclipse.org/jgit/
[2] https://commons.apache.org/proper/commons-io/

7 ДОДАТОК 1 - ЗНАЙДЕНІ ВРАЗЛИВОСТІ


```
{
  "repoResults" : [ {
    "repositoryName" : "https://github.com/asemchenko/Hotello-Spring.git",
    "issues" : [ ]
  }, null, {
    "repositoryName" : "https://github.com/eomjinyoung/JavaWebProgramming.git",
    "issues" : [ {
      "issue" : {
        "issueType" : "SQL dump file",
        "issueDescription" : null
      },
      "fileName" : "SPMS.sql",
      "filePath" : "/tmp/2806842220360010586/Lesson04/docs/SPMS.sql",
      "lineNumber" : null
    }, {
      "issue" : {
        "issueType" : "SQL dump file",
        "issueDescription" : null
      },
      "fileName" : "SPMS.sql",
      "filePath" : "/tmp/2806842220360010586/Lesson05/docs/SPMS.sql",
      "lineNumber" : null
    }, {
      "issue" : {
        "issueType" : "SQL dump file",
        "issueDescription" : null
      },
      "fileName" : "SPMS.sql",
      "filePath" : "/tmp/2806842220360010586/Lesson06/docs/SPMS.sql",
      "lineNumber" : null
    }, {
      "issue" : {
        "issueType" : "SQL dump file",
        "issueDescription" : null
      },
      "fileName" : "SPMS.sql",
      "filePath" : "/tmp/2806842220360010586/Lesson07/docs/SPMS.sql",
      "lineNumber" : null
    } ]
  }, {
    "repositoryName" :
"https://github.com/Tastenkunst/brfv4_javascript_examples.git",
    "issues" : [ ]
  }, {
    "repositoryName" : "https://github.com/cschneider4711/Marathon.git",
    "issues" : [ {
      "issue" : {
        "issueType" : "Shell profile configuration file",
        "issueDescription" : "Shell configuration files might contain
information such as server hostnames, passwords and API keys."
      },
      "fileName" : "profileDetails.js",
      "filePath" :
"/tmp/7195263732987381593/src/main/resources/js/profileDetails.js",
      "lineNumber" : null
    } ]
  }, null, {
    "repositoryName" : "https://github.com/sonngotung/JWebMVC.git",
```

```
    "issues" : [ {
      "issue" : {
        "issueType" : "SQL dump file",
        "issueDescription" : null
      },
      "fileName" : "script.sql",
      "filePath" : "/tmp/6986736060817116137/JWebMVCDemo/script.sql",
      "lineNumber" : null
    } ]
  }, {
    "repositoryName" : "https://github.com/tsultana2/EducationalWebSite.git",
    "issues" : [ ]
  }, {
    "repositoryName" : "https://github.com/mikemelon/JavaWebEducation.git",
    "issues" : [ ]
  }, {
    "repositoryName" : "https://github.com/Ocryst/Web3JavascriptEducation.git",
    "issues" : [ ]
  }, null, {
    "repositoryName" : "https://github.com/infinity23/family-education-
platform.git",
    "issues" : [ ]
  } ]
}
```