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 № 2

дисципліни

“ТЕХНОЛОГІЇ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ В УМОВАХ ВЕЛИКИХ ДАНИХ”

на тему:

„Паралельні обчислення в моделі з розподіленою 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 3**](#_heading=h.tyjcwt)

[**2 ВИКОРИСТАНІ БІБЛІОТЕКИ, ФРЕЙМВОРКИ 4**](#_heading=h.3dy6vkm)

[**3 ОПИС РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 5**](#_heading=h.1t3h5sf)

[3.1 Загальна концепція 5](#_heading=h.4d34og8)

[3.2 Деталі роботи програмного забезпечення 6](#_heading=h.2s8eyo1)

[**4 ОТРИМАНІ РЕЗУЛЬАТИ 7**](#_heading=h.17dp8vu)

[**5 ВИСНОВОК 10**](#_heading=h.1ksv4uv)

[**6 ПОСИЛАННЯ 11**](#_heading=h.44sinio)

[**7 ДОДАТОК 1 - ЛОГ РОБОТИ ПРОГРАМИ 12**](#_heading=h.2jxsxqh)

# Постановка задачі

Необхідно реалізувати вирішення обраної задачі у вигляді послідовної програми та паралельної програми з розподіленою пам’яттю. Необхідно порівняти швидкодію послідовної та паралельної реалізації, а також за можливості продемонструвати залежність часу роботи від розміру задачі, кількості процесів та інших параметрів (залежно від задачі). Можна використовувати різні засоби паралелізму – зокрема, використати кілька процесів, або використати модель акторів в рамках одного процесу. Можна також реалізувати одразу декілька версій (як мультипроцесну, так і з моделлю акторів) та порівняти їх, за це можна отримати додаткові бали. Результатом виконання даної лабораторної роботи є працююча програма, а також звіт про використані технології та можливості, з результатами вимірів.

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.

# ВИКОРИСТАНІ БІБЛІОТЕКИ, ФРЕЙМВОРКИ

Розроблена програма використовує модель акторів, а саме фреймворк [Akka](https://github.com/akka/akka) з використанням мови програмування Java. Мотивація викоритовувати Java, а не Scala – перевикористання коду; частина програмного забезпечення вже була розроблена мовою Java в минулій роботі. Для роботи з Github було використано бібліотеку JGit [1].

# ОПИС РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Загальна концепція

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

В системі є 2 класи акторів:

* [**RepoAnalyzerMain**](https://github.com/asemchenko/studying-parallel-programming/blob/master/lab2/src/main/java/example/kpi/parallel/actors/RepoAnalyzerMain.java) **-** головний актор, який оркеструє екземпляри акторів RepoAnalyzerWorker. Існує тільки в одному екземплярі
* [**RepoAnalyzerWorker**](https://github.com/asemchenko/studying-parallel-programming/blob/master/lab2/src/main/java/example/kpi/parallel/actors/RepoAnalyzerWorker.java) **-** клас акторів, що займається обробкою окремо взятого репозиторію, а саме завантажує файли, аналізує їх вміст, генерує звіт по знайденим вразливостям тощо

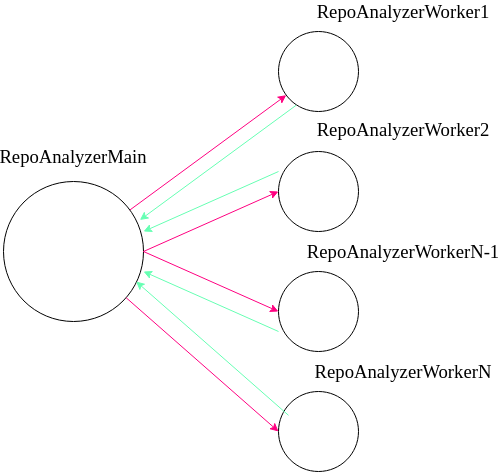


Рис. 1. Загальна схема взаємодії акторів

## Деталі роботи програмного забезпечення

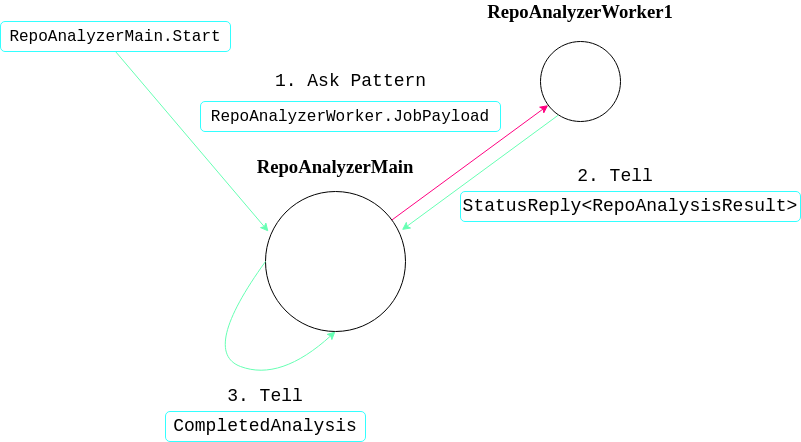


Рис. 2. Взаємодія акторів RepoAnalyzerMain та RepoAnalyzerWorker в деталях

Загальний принцип роботи:

1. Запускається програма (процес), генерується конфігурація на основі аргументів, з якими була запущена програма.
2. Створюється ActorSystem<RepoAnalyzerMain.Command> з єдиним актором – RepoAnalyzerMain.
3. До ActorSystem надсилається повідомлення RepoAnalyzerMain.Start з конфігурацією та списком репозиторіїв, що потрібно проаналізувати.
4. Актор RepoAnalyzerMain при отриманні повідомлення початку роботи читає конфігурацію, та створює N акторів-нащадків RepoAnalyzerWorker. Кожен з Worker’ів отримує повідомлення з частиною роботи – а саме URL репозиторію що потрібно проаналізувати. Взаємодія акторів RepoAnalyzerMain та RepoAnalyzerWorker організована за допомогою [паттерну взаємодії Ask](https://doc.akka.io/docs/akka/current/typed/interaction-patterns.html#request-response-with-ask-between-two-actors). Кожен з акторів RepoAnalyzerWorker при отриманні повsзавантажує вміст репозиторію, зберігає його до тимчасового сховища та аналізує на файли на вразливості, після чого генерує звіт по окремо взятому репозиторію -- RepoAnalysisResult. Повідомлення передається назад до RepoAnalyzerWorker з використанням паттерну akka [StatusReply](https://doc.akka.io/japi/akka/current/akka/pattern/StatusReply.html)
5. RepoAnalyzerMain отримує повідомлення від Worker’ів та зберігає по мірі надходження до списку. Коли отримано відповідь (або timeout чи помилку) від кожного з Worker’ів, головний актор надсилає повідомлення CompletedAnalysis самому собі. Далі це повідомлення обробляється, генерується зведений звіт та зберігається на диск.

Робота Worker’ів організована в окремому Dispatcher'і, який використовує власний Executor (в Akka є вибір між thread-pool-executor та fork-join-executor). Зроблено так, оскільки обробка репозиторію - це блокуюча операція (Більшою мірою CPU-bound). Щоб не блокувати обробку повідомлень в основному Dispatcher’і, коли один з Worker акторів блокує потік виконання, використовується окремий – my-blocking-dispatcher, згідно рекомендації документації[2].

# ОТРИМАНІ РЕЗУЛЬТАТИ

## Вимірювання часу роботи в залежності від параметрів

Всі тести проводились на CPU Intel(R) Core(TM) i7-10510U (4 фізичних ядра, 8 потоків, hyperhreading).

Параметри dispatcher’а для Worker’ів:

my-blocking-dispatcher {

type = Dispatcher

executor = "thread-pool-executor"

thread-pool-executor {

fixed-pool-size = 8

}

throughput = 1

}

При **fixed-pool-size** одночасно працює лише 1 Worker, тому *можна вважати, що в цьому разі обробка виконується послідовно.*

**Початкові результати тесту**

**thread-pool-executor**

| **fixed-pool-size** | **time (seconds)** |
| --- | --- |
| **1** | **115.7** |
| **2** | **70.2** |
| **4** | **69.5** |
| **8** | **78.0** |

Було помічено, що обробка репозиторію №1 (починаючи з 0) займає занадто багато часу (на порядок більше) порівняно з іншими.

Тому зроблено також тест з іншим навантаженням *(без репозиторію, на порядок більшого за інші)*

| **fixed-pool-size** | **time (seconds)** |
| --- | --- |
| **1** | **77.0** |
| **2** | **45.4** |
| **4** | **30.1** |
| **8** | **22.8** |

## 

**Тест з використанням Fork-join-Executor**

my-blocking-dispatcher {

*# Dispatcher is the name of the event-based dispatcher*

type = Dispatcher

*# What kind of ExecutionService to use*

executor = "fork-join-executor"

*# Configuration for the fork join pool*

fork-join-executor {

*# Min number of threads to cap factor-based parallelism number to*

parallelism-min = 2

*# Parallelism (threads) ... ceil(available processors \* factor)*

parallelism-factor = 2.0

*# Max number of threads to cap factor-based parallelism number to*

parallelism-max = 8

}

*# Throughput defines the maximum number of messages to be*

*# processed per actor before the thread jumps to the next actor.*

*# Set to 1 for as fair as possible.*

throughput = 100

}

| **parallelism-max** | **time (seconds)** |
| --- | --- |
| **2** | **29.9** |
| **4** | **23.0** |
| **8** | **27.3** |

# 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 3.4 рази на конкретно взятому прикладі навантаження.

## Отже, найкращий результат отримано на thread-pool-executor з параметром fixed-pool-size=8 (22.8 c). З fork-join executor було отримано співставний показник швидкості роботи, але при параметрі parallelism-max=4 (23.0 c) (що означає максимум 4 одночасно працюючих паралельних задач)

# ПОСИЛАННЯ

1. <https://www.eclipse.org/jgit/>

[2] <https://doc.akka.io/docs/akka-http/current/handling-blocking-operations-in-akka-http-routes.html>

# ДОДАТОК 1 - ЛОГ РОБОТИ ПРОГРАМИ

**[2021-12-15 11:12:57,433]** [INFO] [akka.event.slf4j.Slf4jLogger] [actorModelExecutor-akka.actor.default-dispatcher-3] [] - Slf4jLogger started

**[2021-12-15 11:12:57,525]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-5] [akka://actorModelExecutor/user/Worker0] - Worker received job payload to analyze repo: https://github.com/asemchenko/Hotello-Spring.git

**[2021-12-15 11:12:57,526]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user/Worker1] - Worker received job payload to analyze repo: https://github.com/eomjinyoung/JavaWebProgramming.git

**[2021-12-15 11:12:57,527]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-9] [akka://actorModelExecutor/user/Worker2] - Worker received job payload to analyze repo: https://github.com/Tastenkunst/brfv4\_javascript\_examples.git

**[2021-12-15 11:12:57,527]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-5] [akka://actorModelExecutor/user/Worker3] - Worker received job payload to analyze repo: https://github.com/cschneider4711/Marathon.git

**[2021-12-15 11:12:57,528]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user/Worker4] - Worker received job payload to analyze repo: https://github.com/mikemelon/java-signin.git

**[2021-12-15 11:12:57,528]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-5] [akka://actorModelExecutor/user/Worker5] - Worker received job payload to analyze repo: https://github.com/sonngotung/JWebMVC.git

**[2021-12-15 11:12:57,529]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-9] [akka://actorModelExecutor/user/Worker6] - Worker received job payload to analyze repo: https://github.com/tsultana2/EducationalWebSite.git

**[2021-12-15 11:12:57,529]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-5] [akka://actorModelExecutor/user/Worker7] - Worker received job payload to analyze repo: https://github.com/mikemelon/JavaWebEducation.git

**[2021-12-15 11:12:57,530]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user/Worker8] - Worker received job payload to analyze repo: https://github.com/Ocryst/Web3JavascriptEducation.git

**[2021-12-15 11:12:57,533]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-5] [akka://actorModelExecutor/user/Worker9] - Worker received job payload to analyze repo: https://github.com/mihail-petrov/netit-webdev-java.git

**[2021-12-15 11:12:57,533]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-akka.actor.default-dispatcher-5] [akka://actorModelExecutor/user/Worker10] - Worker received job payload to analyze repo: https://github.com/infinity23/family-education-platform.git

**[2021-12-15 11:12:59,076]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-16] [] - done doing job for repo: https://github.com/mikemelon/JavaWebEducation.git

**[2021-12-15 11:12:59,081]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 7

**[2021-12-15 11:12:59,081]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 1 results so far

**[2021-12-15 11:12:59,084]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-15] [] - done doing job for repo: https://github.com/Ocryst/Web3JavascriptEducation.git

**[2021-12-15 11:12:59,085]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 8

**[2021-12-15 11:12:59,085]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 2 results so far

**[2021-12-15 11:12:59,195]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-8] [] - done doing job for repo: https://github.com/asemchenko/Hotello-Spring.git

**[2021-12-15 11:12:59,195]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 0

**[2021-12-15 11:12:59,195]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 3 results so far

**[2021-12-15 11:12:59,255]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-13] [] - done doing job for repo: https://github.com/sonngotung/JWebMVC.git

**[2021-12-15 11:12:59,256]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 5

**[2021-12-15 11:12:59,256]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 4 results so far

**[2021-12-15 11:13:01,412]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-12] [] - done doing job for repo: https://github.com/mikemelon/java-signin.git

**[2021-12-15 11:13:01,412]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 4

**[2021-12-15 11:13:01,412]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 5 results so far

**[2021-12-15 11:13:05,457]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-11] [] - done doing job for repo: https://github.com/cschneider4711/Marathon.git

**[2021-12-15 11:13:05,457]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 3

**[2021-12-15 11:13:05,457]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 6 results so far

**[2021-12-15 11:13:05,901]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-18] [] - done doing job for repo: https://github.com/infinity23/family-education-platform.git

**[2021-12-15 11:13:05,901]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 10

**[2021-12-15 11:13:05,901]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 7 results so far

**[2021-12-15 11:13:06,682]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-14] [] - done doing job for repo: https://github.com/tsultana2/EducationalWebSite.git

**[2021-12-15 11:13:06,682]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 6

**[2021-12-15 11:13:06,682]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 8 results so far

**[2021-12-15 11:13:14,392]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-10] [] - done doing job for repo: https://github.com/Tastenkunst/brfv4\_javascript\_examples.git

**[2021-12-15 11:13:14,392]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 2

**[2021-12-15 11:13:14,392]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 9 results so far

**[2021-12-15 11:13:15,915]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-17] [] - done doing job for repo: https://github.com/mihail-petrov/netit-webdev-java.git

**[2021-12-15 11:13:15,915]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 9

**[2021-12-15 11:13:15,915]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 10 results so far

**[2021-12-15 11:13:21,430]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerWorker] [actorModelExecutor-my-blocking-dispatcher-7] [] - done doing job for repo: https://github.com/eomjinyoung/JavaWebProgramming.git

**[2021-12-15 11:13:21,430]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got response from worker: 1

**[2021-12-15 11:13:21,430]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got 11 results so far

**[2021-12-15 11:13:21,430]** [INFO] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got the last one!

**[2021-12-15 11:13:21,456]** [ERROR] [example.kpi.parallel.actors.RepoAnalyzerMain] [actorModelExecutor-akka.actor.default-dispatcher-6] [akka://actorModelExecutor/user] - Got error while saving file