

## РОЗДІЛ 3

### ПРОГРАМНІ ЗАСОБИ УПРАВЛІННЯ, ЗБОРУ, ОБРОБКИ І ПЕРЕДАЧІ ІНФОРМАЦІЇ

УДК 004.652

В.Ю. Каштан<sup>1</sup>, Г.Л. Хара<sup>1</sup>, О.Д. Денисенко<sup>1</sup>

<sup>1</sup>Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

#### ПРОГРАМНИЙ ІНСТРУМЕНТ МОНІТОРИНГУ ЗАВАНТАЖЕНОСТІ ВЕБ-СЕРВЕРІВ

**Анотація.** У роботі запропоновано новий програмний інструмент моніторингу завантаженості веб-серверів на основі docker-контейнерів, який легко інтегрується у комп'ютерну мережу з розширеними інструментальними можливостями.

**Ключові слова:** *технології моніторингу, Docker, Zabbix, Nagios, Munin, Docker-Compos, Ланцюг Маркова, YAML, JSON.*

**Вступ.** На сьогоднішній день, кожна компанія, яка веде бізнес в Інтернеті має власний веб-сервер на якому розміщуються один веб-сайт чи декілька. Тому, чим більше даних зберігається та оброблюється на сервері, тим більшу цінність несе безперервна працездатність. Так як кожна секунда непрацездатності системи може привести до збитків або зменшення прибутків, що впливає на ведення бізнесу. До того ж, величина збитків буде залежати від того, як швидко зможуть знайти проблему та виправити її завдяки черговим системним адміністраторам, які відповідальні за працездатність серверів.

Для контролю стану веб-сервера використовують інструменти моніторингу [1]. Це дозволяє інформувати системних адміністраторів про наявність проблем у системі за допомогою листа по електронній пошті чи за допомогою SMS-повідомлення. Тому, розуміючи як правильно налаштовувати та оптимально використовувати функціонал системи моніторингу, можливо реалізувати процес попереджування проблем, без проблем у працездатності.

У глобальній мережі є в наявності декілька десятків різних систем моніторингу [2]. Вони можуть бути безкоштовними чи з пропріетарним кодом. Існуючі системи моніторингу мають один суттєвий недолік це обмеженість функціоналу, що не дозволяє використовувати увесь функціонал програмного забезпечення та отримувати від цього усю користь. Це проявляється у тих випадках, коли конкретна система моніторингу не може збирати метрики про стан досліджуваного серверу для подальшого аналізу, які необхідні черговим системним адміністраторам. Ці обмеження можуть бути різними, наприклад інструмент моніторингу не може збирати дані про мережевий трафік або дане програмне забезпечення може працювати на пристроях тільки під керуванням операційної системи Windows.

Тому, метою даної роботи є розробка та дослідження власної структури програмно-технічних засобів комп'ютерної системи контролю завантаженості веб-серверу.

**Постановка задачі.** Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- дослідити існуючі системи моніторингу;
- розробити модель системи контролю завантаженості веб-серверу;
- розробити структурну схему системи контролю завантаженості веб-серверу;
- розробити програмний інструмент власної системи контролю завантаженості веб-серверу;
- провести тестування розробленого програмного забезпечення.

**Основний зміст роботи.** Програмне забезпечення (ПЗ) системи контролю завантаженості веб-серверу було вирішено розділити на три різні програми, через те, що кожна з них виконує різні завдання.

Перше ПЗ має бути використаною для налаштування нового серверу для зменшення витраченого часу за рахунок зменшення рутинної праці. Вона повинна встановлювати, видаляти та оновлювати пакети. До того ж, програма має проводити налаштування безпеки та керування запущеними процесами. Окрім цього, до її функцій має ще відноситися керування користувачами. Програма повинна підготувати оточення для розгортання інструментів та файлової структури, які будуть використанні для створення системи контролю завантаженості веб-серверу.

Друге ПЗ буде використано для встановлення на веб-сервер для збору та надсилання даних до серверу моніторингу. Програма повинна видаляти використані Docker-контейнери, які містять налаштування для збору даних з веб-серверу та будувати нові. Агент, який працює усередині контейнера, повинен збирати метрики про завантаженість процесору, оперативної пам'яті та накопичувача досліджуваного серверу. При цьому, забезпечуючи зв'язок між веб-сервером та Docker-контейнером з активним агентом. Отримані дані повинні бути відправлені до серверу моніторингу.

Третє ПЗ буде створено для отримання, обробки, зберігання та візуалізації отриманих метрик з веб-серверу. Програма повинна мати функціонал для отримання, обробки, зберігання та візуалізації отриманих даних з веб-серверу. Для вирішення цієї задачі мають бути застосовані Docker-контейнери [3] для оптимізації використаних ресурсів сервером моніторингу. Тому, програма повинна мати можливості для організації зв'язку контейнерів між собою та фізичним вузлом використовуючи інструмент Docker-Compose. До цього ж, забезпечувати постійну роботу Docker-контейнерів.

Тому, у якості технічного засобу для розробки програми є комп'ютер під керуванням операційної системи CentOS 8, який має встановлену оболонку BASH (Born Again Shell). У якості програмного засобу використовується мова програмування bash та YAML-синтаксис.

Після цього, створюється файлова структура, де будуть розміщені необхідні файли для функціонування серверу моніторингу та встановлюється

Docker-Compose. Наступною дією є створення контейнеру, який би забезпечив гарантійну доставку метрик з веб-серверу. Це є корисним, коли з'являються мережеві проблеми і отримані дані з веб-серверу будуть втрачені. До цього контейнеру імпортуються наступні директорії: /home/admin/monitoring/rabbitmq/etc. У ній будуть зберігатися файли для налаштувань rabbitmq, до них відносяться такі файли: rabbitmq.conf, enabled\_plugins, definitions.json. Файл rabbitmq.conf має опції для налаштування клієнту. Файл definitions.json має налаштування для автоматичного створення черг для надсилання метрик при завантаженні контейнеру. Це економить купу часу, так як до цього, при перезапуску контейнеру для створення черг необхідно було підключатися до графічного клієнту у браузері та створювати усе спочатку. Файл enabled\_plugins має список підключених плагінів, які необхідні для функціонування контейнеру з rabbitmq.

На рисунку 1 представлено головну сторінку інтерфейсу програми.

Використання програмного забезпечення системи контролю завантаженості веб-серверу дозволить відмовитися від витрат на придбання ліцензій для моніторингу серверів. Так як, усі компоненти які входять до створеної системи є безкоштовними. В той же час, один із основних конкурентів ПЗ для моніторингу Nagios, коштує від 1995 доларів в залежності від обраного тарифу.

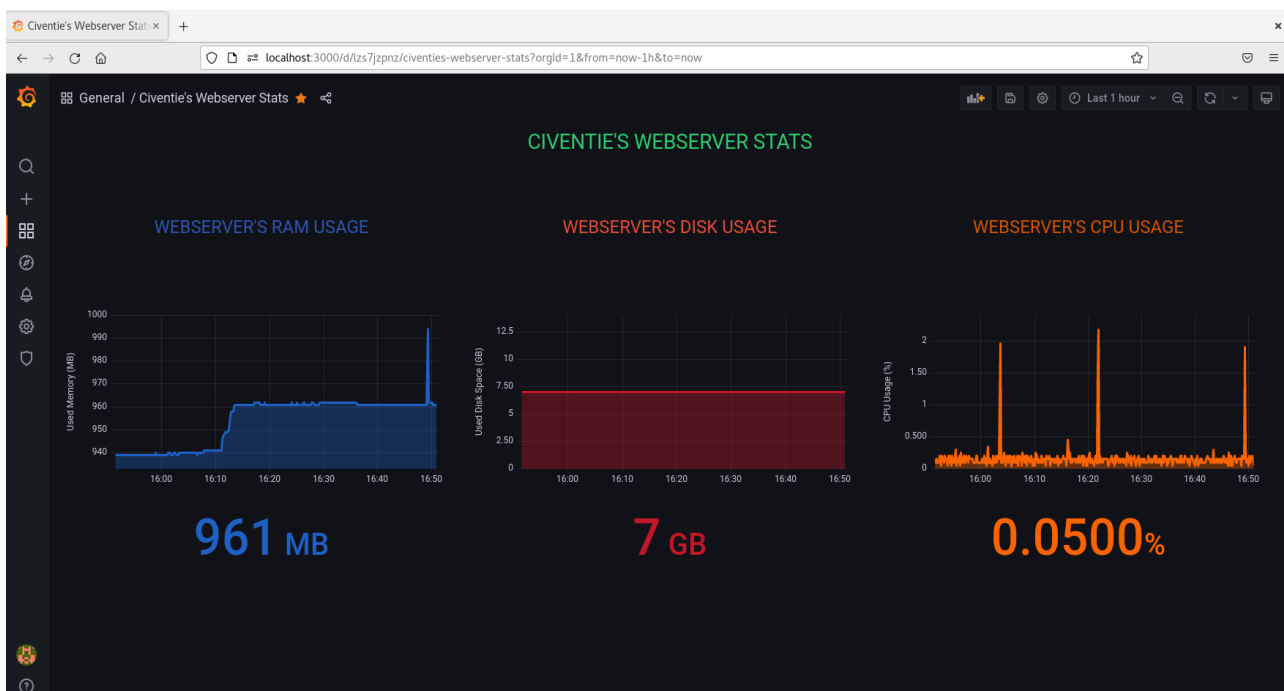


Рис. 1. Головна сторінка інтерфейсу програми

Тестування розробленого програмного забезпечення було проведено на прикладі веб-серверу ІТ-компанії Civenty на основі методу безпосередньої оцінки. Це дозволило виявити значення використаних ресурсів системою контролю завантаженості веб-серверу та при рівних заданих умовах, порівняти з конкуруючою системою. Експеримент пройшов вдало, так як, усі задані умови були виконані.

**Наукова новизна** полягає у створенні нового програмного інструменту завантаженості веб-сервера на основі комплексного використання Docker-контейнерів та YAML-синтаксису. Це дозволило забезпечити безперервну та надійну доставку даних з досліджуваного серверу на сервер моніторингу.

**Висновки.** В даній роботі вирішено науково-практичну задачу створення програмно-технічного інструменту контролю завантаженості веб-серверу. Проведений аналіз існуючих систем моніторингу дозволив правильно вибрати мережеве обладнання для системи контролю завантаженості веб-серверу; створити програмне забезпечення з урахуванням недоліків існуючих програм моніторингу. Розроблене програмне забезпечення містить три ключових аспекти: налаштування серверу моніторингу на основі YAML-синтаксис; збір даних з досліджуваного веб-серверу на основі мови програмування bash та опрацювання отриманих даних з веб-серверу на основі YAML-синтаксису. Для автоматизації певних завдань були створені шаблони за допомогою текстового формату JSON.

### ПЕРЕЛІК ПОСИЛАНЬ

1. 15 Best IT Infrastructure Monitoring Tools & Software. [Електронний ресурс] – Режим доступу: <https://sematext.com/blog/infrastructure-monitoring-tools/#toc-10-nagios-9>.
2. Nagios XI Enterprise Server and Network Monitoring Software. [Електронний ресурс]: Режим доступу: <https://www.nagios.com/products/nagios-xi/>
3. Docker Installation and Configuration Requirements. [Електронний ресурс]: Режим доступу: <https://docs.cambridgesemantics.com/anzograph/v2.2/userdoc/install-docker.htm>

УДК 004.4'275

Н.О. Соколова<sup>1</sup>, Є.В. Головка<sup>1</sup>

<sup>1</sup>Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

### РОЗРОБКА ЛОГІКИ ПОВЕДІНКИ РУХОМИХ ПЕРСОНАЖІВ НА ІГРОВОМУ ДВИГУНІ UNITY 3D

**Анотація.** Описано процес створення логіки поведінки персонажів гри. Виконано 3D моделювання нерухомих та рухомих ігрових об'єктів з використанням сучасного інструментарію 3D-моделювання на візуалізованих ігрових сценах.

**Ключові слова:** ігровий двигун Unity 3D, 3D-моделі, логіка поведінки ігрових персонажів.

**Вступ.** Ринок комп'ютерних ігор пережив нечуваний сплеск (майже на 10%) у 2020 році на фоні пандемії, навіть при уповільненні зросту у 2021 році,