Микола Стиранець, Володимир Яркун*
[1] НЛТУ України, Львів, Україна, 23styranets.m@nltu.lviv.ua
[2] НЛТУ України, Львів, Україна. ORCID: 0000-0002-4876-1111,
yarkun@nltu.edu.ua

Архітектура системи збору та аналізу логів у базах даних

Анотація. Запропоновано архітектуру, адаптовану для зниження навантаження на базу даних шляхом введення кількох компонентів на різних етапах життєвого циклу даних з акумуляційними властивостями та асинхронною передачею даних в БД. Виконано аналіз логів на предмет виявлення частоти та відсоткового складу типів помилок і ранжування їх за під'єднаними застосунками та токенами.

Ключові слова – логи, бази даних, асинхронність, токени доступу, аналіз.

Зростання обсягів даних і складності програмних систем робить системи збору логів незамінними для виявлення, аналізу та вирішення проблем у великих розподілених додатках. Вони дають змогу виявляти аномальність та помилки, що відбуваються в реальному часі, та оперативно реагувати на них. Поєднуючи великі обсяги даних з потужними алгоритмами аналізу, такі системи стають ефективним інструментом для покращення якості програмного забезпечення та забезпечення безперебійної роботи додатків.

Збір та аналіз логів у базах даних є критично важливим процесом для забезпечення безпеки вебресурсів і самих баз даних, оскільки дає змогу виявляти аномалії, моніторити підозрілу активність та своєчасно реагувати на потенційні загрози. Використання методів інтелектуального аналізу даних у процесі оброблення логів сприяє виявленню шаблонів поведінки та ідентифікації вразливостей, що знижує ризик несанкціонованого доступу та забезпечує цілісність інформаційних систем.

У статті [1] розглядається процес використання методів машинного навчання для аналізу журналів вебсервера NGINX, розглянуто підходи до збору, оброблення та аналізу даних журналів, а також використання цих даних для підвищення безпеки та ефективності вебсерверів. В роботі [2] проаналізовано статистику та атаки, класифіковані загрози безпеки інформації та розглянуто

* науковий керівник

практичні приклади використання аналізу лог-файлів для виявлення загроз. Окрім цього – впроваджено систему для аналізу лог-файлів з використанням ELK, враховуючи встановлення Elasticsearch, Logstash і Kibana, а також розробку шаблонів та фільтрів для аналізу лог-файлів. Зокрема у [3], здійснено аналіз логів для отримання даних про необхідність використання фаєрволу. В якості інформаційної бази дослідження були використані публікації, наукові видання, навчальні посібники. У [4] розглянуто актуальні завдання моніторингу та синтаксичного аналізу даних при розробленні автоматизованих систем, що забезпечують ефективний моніторинг вступної кампанії до закладу вищої освіти (ЗВО).

Архітектура системи. Для ефективної роботи зі значним об'ємом даних таким як логи, є необхідність використовувати логіку орієнтовану на оптимізацію цих завдань та програмних компонентів.

Джерелом даних є будь-який застосунок в який підключена бібліотека для запису логів в дану систему. З деякою періодичністю, визначеною, в тому числі, рівнем логів – вони надсилаються на сервіс (рис. 1). Це компенсує кількість запитів, аніж при миттєвому їх надсиланні щоразу як з'являється повідомлення. На сервері що обробляє запити, знову ж таки, перевіряється рівень логів. При отриманні таких, що перевищують вказаний поріг – надсилається повідомлення визначеним шляхом для термінового реагування спеціаліста. Далі вони передаються в чергу брокера повідомлень, який працює асинхронно і гарантує їх доставку.

Програмна реалізація. Chron сервіс з певною періодичністю запускає обробник, який переписує набори логів з черги в базу даних, систематизуючи випадковий порядок викликів у чіткі інтервали. Таким чином посередництво брокера повідомлень знижує навантаження на базу даних.

Переглянути усі логи клієнт може в застосунку, який, в свою чергу, запитує відформатовані дані з бази даних.

* науковий керівник

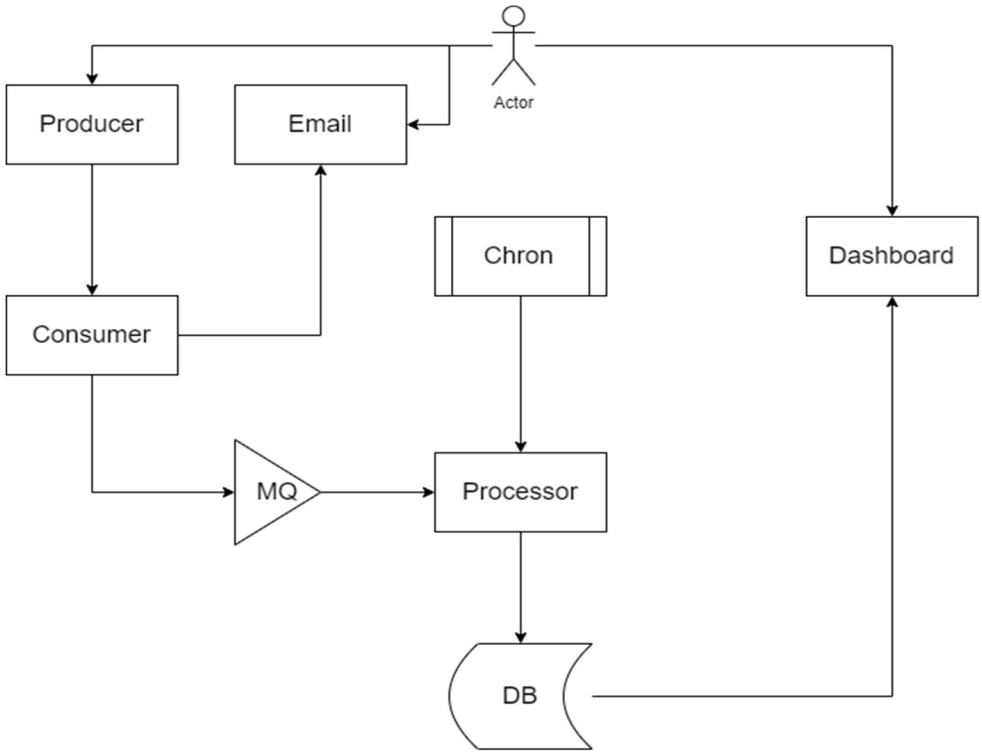

Рисунок 1. Архітектура програмного рішення

Інформаційна панель системи збору на аналізу догів відображає список логів в хронологічному порядку з усіма деталями (рис. 2).

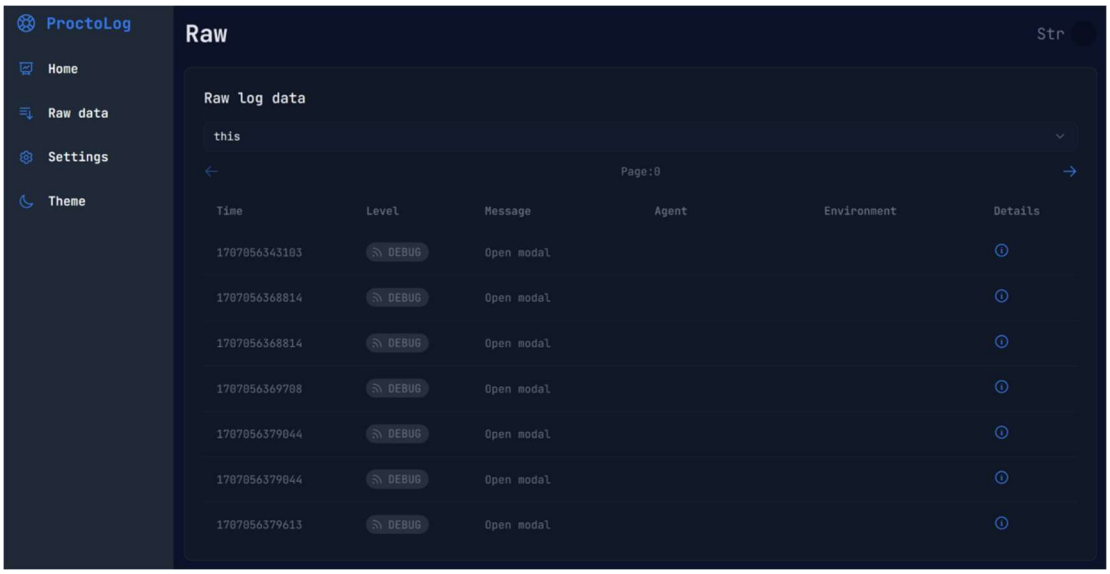

Рисунок 2. Список логів

Висновок. Реалізація система з використанням запропонованої архітектури зменшує використання часу на аналіз логів. Система дає змогу автоматично обробляти та оптимізувати логи, що сприяє значному скороченню часу на їх аналіз. Автоматизований процес збору та оброблення логів призводить до зменшення витрат на технічне обслуговування та аналіз даних. Забезпечене надій-

* науковий керівник

не логування, що сприяє підвищенню достовірності інформації та швидкості виявлення проблем. Надається зручний інтерфейс для детального аналізу логів, що дає змогу користувачам отримувати кількісну оцінку ефективності програмного забезпечення.

Список використаних літературних джерел

[1] Багнюк Н., Лінчук О., & Шипулін О. (2023). Аналіз журналів трафіку NGINX за допомогою машинного навчання. Комп'ютерно-інтегровані технології: освіта, наука, виробництво, (53), 86-91. https://doi.org/10.36910/6775-2524-0560-2023-53-13.

[2] Чурбаков К. О. Аналіз лог-файлів з використанням ELK для виявлення потенційних загроз безпеці: кваліфікаційна робота бакалавра за спеціальністю 125 - Кібербезпека / К. О. Чурбаков. – Тернопіль: ТНТУ, 2023. – 84 с.

[3] Маланчук М. І. Аналіз логів з використанням фаєрволу FortiGate: кваліфікаційна робота бакалавра за спеціальністю 125 - Кібербезпека / М. І. Маланчук – Тернопіль: ТНТУ, 2023. – 65 с.

[4] В. Карашецький, В. Яркун, «Автоматизована система моніторингу вступної кампанії закладу вищої освіти» Інформаційні технології та комп'ютерна інженерія, № 51 (2), 2021. doi: 10.31649/1999-9941-2021-51-2-12-16.

Architecture of the system of collection and analysis of logs in databases

Mykola Styranets, Volodymyr Yarkun

An architecture adapted to reduce the load on the database by introducing several components at different stages of the data life cycle with accumulative properties and asynchronous data transfer to the database is proposed. Log analysis was performed to identify the frequency and percentage composition of error types and rank them by connected applications and tokens.

* науковий керівник