УДК 004.77 (045)

Т. І. Трояновська, Л. А. Савицька, В. Л. Комаров

ЗАСОБИ ТА МОДЕЛЬ МОНІТОРИНГУ ДАНИХ МІКРОСЕРВІСНОЇ СИСТЕМИ

Вінницький національний технічний університет, м. Вінниця

Анотація. Дана робота присвячена дослідженню методів та засобів для комплексного моніторингу додатків, що побудовані за принципами мікросервірсої архітектури з метою швидкої локалізації проблем у роботі таких систем. Зокрема, виконано аналіз основних засобів збирання інформації про роботу мікросервісних додатків; аналізу цієї інформації, її систематизації та класифікації; виконано аналіз технологічного стеку для графічного відображення зібраної інформації з метою подальшого моніторингу та вмконання аналітики; розглянуто найбільш критичні задачі у впровадженні моніторингу мікросервісної системи; запропоновано модель роботи частини мікросервісної системи, що призначена для аналізу логів, системних звітів.

Ключові слова: методи та засоби моніторингу мікросервісних додатків, моніторинг, мікросервісна архітектура, MSA (Micro Service Architecture), задачі моніторингу, модель моніторингу мікросервісної системи, алгоритм обробки логів, ELK (Elasticsearch Logstash Kibana), Elasticsearch, Logstash, Kibana, Filebeat, Analitycs.

Аннотация. Данная работа посвящена исследованию методов и средств для комплексного мониторинга приложений, построенных на принципах микросервисной архитектуры с целью быстрой локализации проблем в работе системы. В частности, выполнен анализ основных средств сбора информации о работе микросервисных приложений; анализа такой информации, систематизации и классификации; выполнен анализ технологического стека для графического представления собранной информации для дальнейшего мониторинга и аналитики; рассмотрено наиболее критические задачи при ведении мониторинга микросервисной системы; предложено модель работы части микросервисной системы предназначеной для анализа логов, системных отчетов.

Ключевые слова: методы и средства мониторинга микросервисных приложений, мониторинг, микросервисная архитектура, MSA (Micro Service Architecture), задачи мониторинга, модель мониторинга микросервисной системы, алгоритм обработки логов, ELK (Elasticsearch Logstash Kibana), Elasticsearch, Logstash, Kibana, Filebeat, Analitycs.

Abstract. This work is devoted to methods and tools for complex monitoring of applications, built on the principles of microservice architecture in order to quickly locate problems in the system. In particular, the analysis of the main technologies for collecting information about the operation of microservice applications; its analysis, systematization and classification as well as the technological stack for graphically displaying the collected information for further monitoring and analytics; the most critical tasks in the implementation of monitoring of the micro-service system; the model of the operation of the part of the microservice system, which is intended for analysis of logs, system reports.

Keywords:methods and tools for monitoring of microservice applications, monitoring, microservice architecture, MSA (Micro Service Architecture), monitoring tasks, microservice system monitoring model, log processing algorithm, ELK (Elasticsearch Logstash Kibana), Elasticsearch, Logstash, Kibana, Filebeat, Analitycs.

Вступ

З розвитком технологій росте й обсяг даних, які необхідно обробляти щодня. Відповідо росту даних зростає й складність програмних та архітектурних рішень для підтримки швидкодії та змоги обробляти все більші обсяги даних. Все більше відомих інтернет ресурсів віддають перевагу мікросервісній архітектурі. Рішення про перехід на цілковито нову архітектуру зумовлене все більшою складністю горизонтального масштабування обчислювальних здатностей будь-яких сервісів. Проте, дане рішення тягне за собою ряд задач. Важливим компонентом будь-якої системи є її моніторинг та аналітика. У мікросервісних системах моніторинг є складнішим ніж у звичайних монолітних додатках.

Актуальність

Головною умовою успішної та безперебійної роботи будь-якого онлайн-сервісу є його стабільність. Для цього важливо мати механізм моніторингу, виявлення та усунення помилок. Аналітика роботи таких сервісів сприятиме оптимізації процесу локалізації помилок [1-3]. Водночас, із зростанням обчислювальних потужностей, зростає і складність програмних та апаратних засобів. Багато компаній застосовують мікросервісну архітектуру, оскільки вона дає можливості зручного масштабування будь-якого проекту, який заздалегідь налаштований та спроектований за принципами мікросервісних додатків. Проте, моніторинг таких сервісів є цілком інакшим у порівнянні з моніторингом монолітних додатків, де сам процес моніторингу даних реалізувати простіше. Тому виникає потреба знайти нові підходи та моделі впровадження ефективного моніторингу мікросервісної системи, що пришвидшать виявлення та локалізацію помилок і, таким чином, дозволять зменшити витрати ресурсів.

Мета

Метою статті є розробка моделі моніторингу даних мікросервісних дадатків з метою оптимізації процесу локалізації помилок у роботі мікросервісної системи за рахунок впровадження ефективного програмного стеку логування та моніторингу та реалізації аналітики роботи та життєвого стану мікросервісних додатків в режимі реального часу.

Для досягнення поставленої у статті мети необхідно вирішити задачі, що наведені нижче:

1. Виконати аналіз критичних задач у процесах моніторингу мікросервісних додатків.

2. Виконати огляд сучасних засобів моніторингу мікросервісних додатків та визначити найбільш дієві з них.

3. Розробити модель моніторингу сервісів мікросервісної системи.

Аналіз критичних задач у процесах моніторингу мікросервісних додатків

Процеси моніторингу мікросервісних систем є набагато складнішими, ніж моніторинг систем, де функціонал не розподілено на незалежні частини. Це зумовлено наявністю великої кількості відокремлених сервісів, які не залежать один від одного і можуть кардинально відрізнятись архітектурою, типом мови програмування або навіть ядром операційної системи, тому важливо узагальнити критичні моменти ведення загального моніторингу між окремими сервісами (табл. 1).

Визначимо деякі ключові задачі, що можуть виникати під час процесу моніторингу мікросервісної системи [2]:

1) Збір даних для моніторингу. У роподіленій системі є окремі частини і іноді їх кількість може сягати сотень або навіть тисяч. При цьому усі вони можуть бути незалежними за архітектурою, мовами програмування, операційними системами. Збирати дані з усіх систем є непростою задачею.
2) Попередня обробка даних. Відповідно до різноманітності застосованих програмних засобів, моніторингові дані також є різноманітними. Тому перед пошуком будь-якої інформації по цим даним, їх необхідно привести до єдиного формату, виділивши ключові дані, необхідні для пошуку та індексації.
3) Пошук у великих обсягах даних. Сотні або навіть тисячі окремих сервісів можуть генерувати терабайти моніторингової інформації щодня. Відповідно, пошук у таких обсягах інформації є складною задачею, якщо не застосовано оптимальну індексацію.
4) Візуалізація оброблених даних. Проіндексовані та оброблені дані ще не готові до моніторингу. Вони доступні для пошуку, аналізу, але відсутність дружнього графічного призводить до труднощів для нетехнічного персоналу.

Таблиця 1 – Критичні задачі інтеграції моніторингу мікросервісних додатків та методи їх усунення

№ п/п	Задача	Існуючі методи та засоби вирішення	Методи та засоби, що пропонуються
1	Збір моніторингових даних (логів)	Обробка даних в кожному сервісі	Збір даних за допомогою Filebeat
2	Попередня обробка даних	Ручне перетворення даних	Автоматизована обробка даних за допомогою Logstash
3	Пошук у великих обсягах даних	Індексовані поля MySQL/NoSql, grep	Індексація даних за допомогою Elasticsearch
4	Візуалізація оброблених даних	Власні адміністративні панелі	Використання панелі моніторингу Kibana в режимі реального часу

У результаті аналізу критичних задач, що виникають під час процесів моніторигу мікросервісних систем, можна визначити, що головним правилом моніторингу є зручність та швидкість виявлення проблем. Щошвидше помилка буде знайдена та локалізована, то менше збитків та витрат буде витрачено на їх локалізацію [2-3]. Одночасно з'являється можливість слідкувати за перебігом бізнес-процесів у мікросервісній системі в режимі реального часу.

Огляд сучасних засобів моніторингу мікросервісних додатків

Одним з найпопулярніших засобів моніторингу мікросервісної системи є впровадження процесів логування за допомогою програмного стеку ELK (Easticsearch Logstash Kibana). Цей програмний стек складається з трьох програмних продуктів:

1. Elasticsearch – індексована база даних та рушій пошуку;
2. Logstash – препроцессор вхідної інформації;
3. Kibana – графічний інтерфейс відображення моніторингових даних [4-5].

Розробником цих продуктів є одна і та ж компанія і, відповідно, ці продукти комплексно працюють один з одним.

Засобом збору моніторингових даних є собою утиліта Logstash. Вона дозволяє збирати дані будь-яких форматів та протоколів, фільтрувати ці дані та перенаправляти їх далі на зберігання, обробку або індексацію (рис 1).

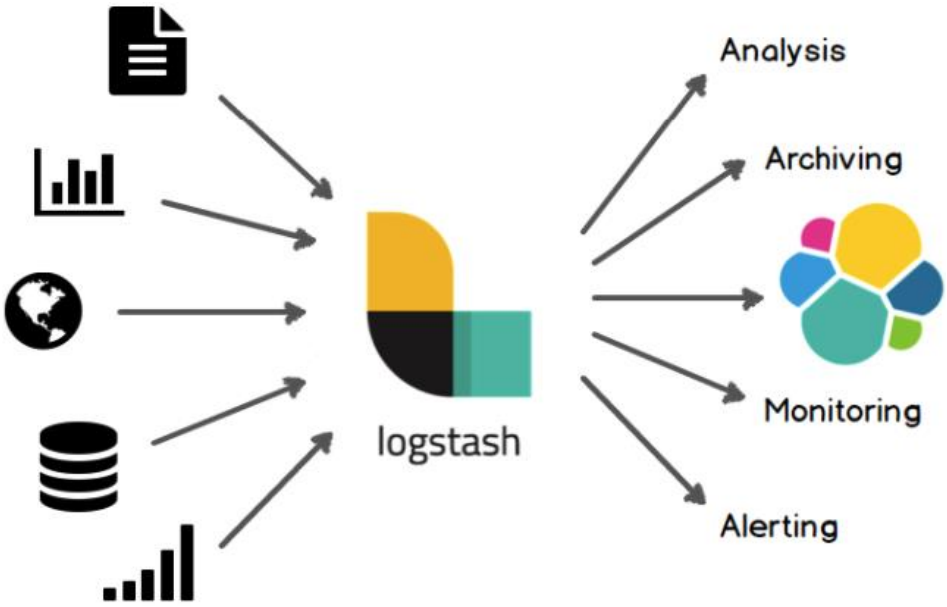

Рисунок 1 – Схема обробки даних засобами Logstash

Схожим рішенням є утиліта Fluentd, але вона поступається універсальностю Logstash та підтримкою значної частини наперед підготованих шаблонних типів вхідних даних. Logstash підтримує більш ніж 200 наперед підготованих шаблонів вхідних даних [5]. Основні дані, які здатний приймати Logstash є такими:

– логи веб-серверів Apache, Nginx та додатків на прикладі log4j для Java;
– системні логи фаєрволів, мережевих пристроїв, Windows event logs, syslog;
– метрики таких додатків як Ganglia, Collectd, NetFlow, JMX;
– обробка будь-яких пристроїв та додатків інфраструктури через TCP та UDP.

Logstash оптимізований для передачі даних у Elasticsearch для подальшої індексації та архівації. Зокрема, Elasticsearch є повноцінним механізмом пошуку та зберігання даних. Він легко масштабується та налаштовується. З самого початку розвитку даного продукту був реалізований повнотектсовий пошук на основі пошукового механізму Apache Lucene, влдночас із додатковими можливостями легкого масштабування, реплікації та ін. Таким чином, даний продукт став засобом для highload-проектів із великими обсягами даними [6].

Особливістю Elasticsearch є його проста конфігурація. Даний програмний засіб працює одразу після інсталяції та нескладних налаштувань. Окрім того, навколо цього продукту успішно сформована спільнота розробників, які можуть вирішити задачі налаштування кластеру у кожному конкретному випадку.

Із зростанням популярності та універсальності даного програмного засобу, інженери стали використовувати його не лише для повнотекстового пошуку, а й для централізованого зберігання моніторингових даних та різноманітної аналітики.

З метою відображення підготованихданних застосовується панель Kibana. Це один з елементів програмного стеку ELK, що відповідає за графічне представлення будь-якої інформації. Kibana створена виключно для роботи з Elasticsearch [6] і за її допомогою можна шукати, переглядати, взаємодіяти з даними, що зберігаються в індексах Elasticsearch. Даний програмний засіб дозволяє обробляти величезні обсяги даних за допомогою зручного інтерфейсу та у режимі реального часу. Може бути розширений для інтеграції нового нестандартного функціоналу.

Модель моніторингу даних мікросервісної системи

Запропонуємо графічне представлення та опис роботи моделі обробки моніторингових даних мікросервісної системи (рис. 2).

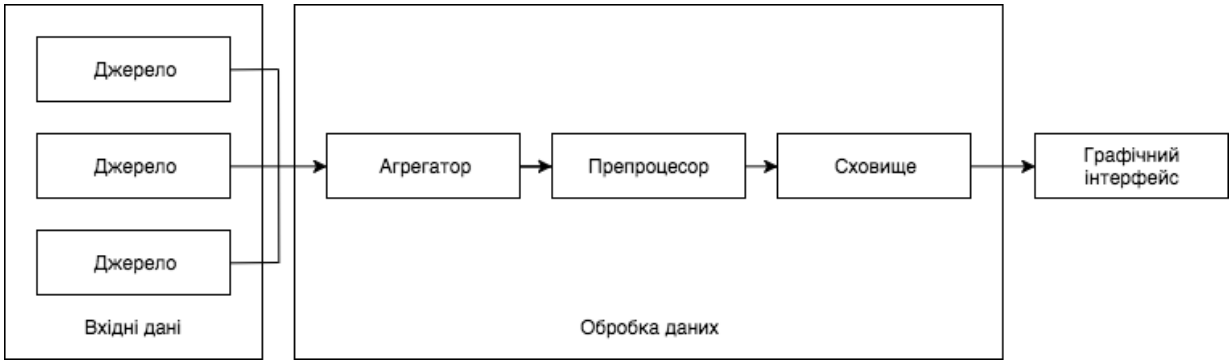

Рисунок 2 – Схема процесу обробки моніторингових даних мікросервісної системи

Схема процесу обробки моніторингових даних мікросервісної системи містить такі компоненти:

1) Джерело – сервіси, що генерують необроблені дані (Syslog). Тут генеруються логи (logs). Багато програмних засобів, додатків, середовищ генерують логи різних форматів та розширень. Ці дані передаються на агрегатор, який встановлений у кожному сервісі, і за цим агрегатором є необхідність спостерігати.
2) Агрегатор необроблених даних. Даний компонент є вхідною точкою для збору (агрегування) моніторингових даних. Він відповідає за транспортування моніторингових даних до препроцесора Logstash.
3) Препроцесор агрегованих даних (Logstash). Дані, потрапивши на вхід цього сервісу, розбиваються на структурні частини, далі виділяються необхідні дані, які передаються на вихід у єдиному форматі для подальшого їх збереження, індексування та аналізу в Elasticsearch.
4) Сховище оброблених даних (Elasticsearch). Централізоване сховище відповідає за збереження моніторингових даних. Тут надається можливість повнотекстового пошуку у великих обсягах даних з використанням пошукового механізму Apache Lucene.
5) Графічний інтерфейс взаємодії з даними (Kibana). Кінцевою точкою роботи моделі є графічне відображення проаналізованої інформації у вигляді різноманітних графіків, схем, діаграм.

Останні три компоненти, власне стек ELK має бути встановлений окремо від інших сервісів системи та повинен бути доступним для всіх сервісів [7].

Після встановлення програмного засобу ELK наступним кроком є конфігурація даних продуктів. Спочатку необхідно визначити вхід та вихід Logstash. Типова конфігурація складається з трох секцій – "input", "filter", "output". В секції "input" необхідно визначитись тим, звідки будуть надходити дані.

```
input {
  stdin {
    type => "rails"
  }
  syslog {}
}
```

Секція "filter" відповідає за розбиття моніторингових даних на структурні складові, які потім будуть передані на вихід.

```
 filter {
  grok {
    match => ["message", "%{RAILS}"]
  }
  date {
    match => [ "timestamp", "YYYY-MM-dd HH:mm:ss Z"]
    target => "@timestamp"
    remove_field => "timestamp"
  }
```

Згодом після обробки дані потрапляють на вихід у централізоване сховище.

```
output {
   elasticsearch {
      host => "localhost:9200"
```

```
        }
    }
```

Далі необхідно скористатися можливостями Elasticsearch. Коли система буде готова до імпорту логів, сервіси відправлятимуть дані на опрацювання запропонованою моделлю, необхідно налаштувати панель Kibana. На даному кроці весь процес налаштування зводиться до налаштування графіків та агрегацій даних за допомогою інтерфейсу самої панелі (рис. 3).

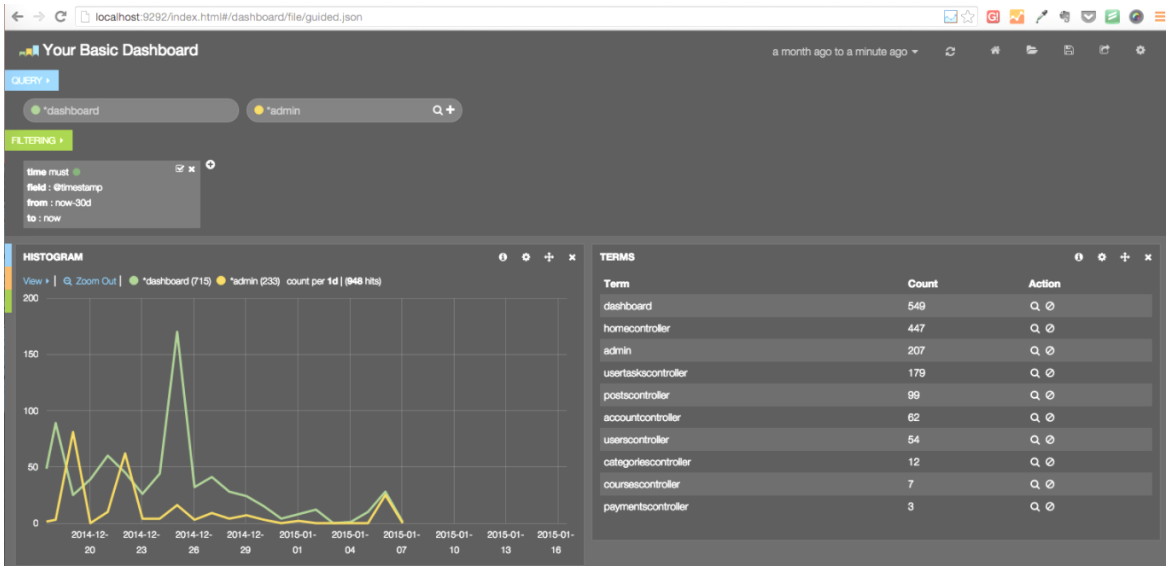

Рисунок 3 – Графічний інтерфейс Kibana

Наведена модель обробки моніторингових даних є найпростішим прикладом використання програмних засобів ELK. Звісно, ці можливості значно ширші, ніж описані в даній статті. Дана модель здатна обробляти дані сотень або тисяч сервісів, агрегувати дані різних програмних стеків, технологій та форматів. Проте, варто зазначити, що представлена модель моніторингу є ефективною лише у тих проектах, де доцільно її використовувати – у проектах з великою кількістю моніторингових даних.

Висновки

У даній статті був проведений аналіз методів та засобів моніторину мікросервісних додатків. Зокрема, проведено огляд сучасних засобів моніторингу мікросервісних додатків, обробки та збирання інформації з окремих сервісів системи для подальшої її аналітики. Також, проаналізовано критичні задачі, що виникають під час процесів впровадження моніторингу, що дало можливість запропонувати шляхи їх уникнення.

Запропонована модель моніторингу даних мікросервісної системи дозволяє оперативно обробляти статистичну інформацію, реагувати на критичні події в роботі окремих сервісів а також аналізувати отриману інформацію з метою передбачення можливих проблем. Наведені в статті засоби на достатньому рівні задовольняють потреби моніторингу великих мікросервісних систем.

Список літератури

1. Трояновська Т. І. Інформаційна технологія доставки контенту у системах комп’ютеризованої підготовки спеціалістів. // Гороховський О. І., Трояновська Т. І., Азаров О. Д. Монографія. Вінниця : ВНТУ, 2016.–160 с.

2. Григорьев А.Н. InfoStream. Мониторинг новостей из Интернет: технология, система, сервис: научно-методическое пособие. / А.Н. Григорьев, Д.В. Ландэ, С.А. Бороденков, Р.В. Мазуркевич, В.Н. Пацьора // Киев, ООО "Старт-98", 2007., с. 40-41.

3. Ньюмен С. Создание микросервисов. / Ньюмен С. // Питер, O’Reilly, 2016., с. 32-46.

4. Таллоч М. Знакомство с Windows Azure. Для ИТ-специалистов / М. Таллоч // М.:ЭКОМ Паблишерз, 2014. — 154 с.

5. Подробное описание возможностей разработки с Microsoft Azure Cloud Services. [Електронний ресурс]. Режим доступу: URL: http://habrahabr.ru/company/microsoft/blog/242543/

6. Тімінський О.Г. Виникнення, розвиток і проблеми інформаційних технологій управління [Текст] / О.Г. Тімінський // Управління розвитком складних систем. – 2016. – No 25. – С. 86 –90.

7. Трояновська, Т. Алгоритм структурованої візуалізації XML-файлів [Текст] / Трояновська Т. І., Бойко О. В. // „Intrenet-Education-Science" : Міжнародна науково-технічна конференція, 11–14 жовтня 2016 р. – Вінниця : КІВЦ ВНТУ, 2016. – С. 142–144. ISBN 966–641–102–4.

Стаття надійшла: 28.12.2017.

Відомості про авторів

Трояновська Тетяна Іванівна, к.т.н., доцент кафедри обчислювальної техніки, ВНТУ, кафедра обчислювальної техніки, tetianatroianovska@gmail.com, Вінниця, Хмельницьке шосе, 95

Савицька Людмила Анатоліївна, к. т. н., доцент кафедри обчислювальної техніки, ВНТУ, кафедра обчислювальної техніки, ludik0304@gmail.com, Вінниця, Хмельницьке шосе, 95

Комаров Володимир Леонідович, студент кафедри обчислювальної техніки, ВНТУ, кафедра обчислювальної техніки, vovikems@gmail.com, Вінниця, Хмельницьке шосе, 95

T. I. Troianovska, L. A. Savytska, V. L. Komarov

MEANS AND MODEL OF MICROSERVICE SYSTEM DATA MONITORING

Vinnitsa national technical university. Vinnitsa