

WYŻSZA SZKOŁA INFORMATYKI I UMIEJĘTNOŚCI  
Wydział Informatyki i Zarządzania  
Kierunek: Informatyka

Sebastian Florek  
nr albumu: 29571

Praca Magisterska  
Monitorowanie otoczenia za pomocą mikrokontrolerów Raspberry Pi  
w oparciu o system zarządzania klastrem Kubernetes

Praca napisana pod kierunkiem  
dr inż. Grzegorz Zwoliński

Rok akademicki 2016/2017

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>3</b>
1.1	Problematyka i zakres pracy . . . . .	3
1.2	Metoda badawcza . . . . .	4
1.3	Przegląd literatury w dziedzinie . . . . .	4
1.4	Układ pracy . . . . .	4
<b>2</b>	<b>Podstawy teoretyczne</b>	<b>6</b>
2.1	Podstawowe definicje . . . . .	6
2.1.1	Wirtualizacja oparta na Linuksie . . . . .	6
2.1.2	Wirtualizacja oparta o nadzorcę . . . . .	6
2.1.3	Wirtualizacja oparta na kontenerach . . . . .	6
2.1.4	Mikrokontroler RaspberryPi . . . . .	6
2.1.5	Kompilacja skrośna . . . . .	6
2.2	Systemy zarządzania kontenerami . . . . .	6
2.2.1	Kubernetes . . . . .	6
2.2.2	Docker Swarm . . . . .	6
2.2.3	Resin.io . . . . .	6
<b>3</b>	<b>Projekt KubePi</b>	<b>7</b>
3.1	Analiza wymagań . . . . .	7
3.2	Użyte technologie . . . . .	7
3.3	Projekt . . . . .	7
3.4	Opis użytkowania . . . . .	7
3.5	Przykład użycia . . . . .	7
3.6	Możliwości rozszerzania aplikacji . . . . .	7
<b>4</b>	<b>Podsumowanie</b>	<b>8</b>
	<b>Bibliografia</b>	<b>8</b>

<i>SPIS TREŚCI</i>	3
<b>Spis rysunków</b>	<b>9</b>
<b>Spis tabel</b>	<b>10</b>

# Rozdział 1

## Wstęp

### 1.1 Problematyka i zakres pracy

Wraz z rozwojem Internetu Rzeczy na świecie powstają coraz to nowe urządzenia mające na celu automatyzację działań i ułatwienie życia człowieka. Dzięki ich zdolności do wzajemnej komunikacji, wymiany informacji oraz zdalnego zarządzania zasobami stają się one coraz bardziej popularne, a wręcz wymagane w życiu codziennym coraz większej grupy osób. Sterowanie oświetleniem, radiem czy innymi sprzętami elektronicznymi za pomocą naszego smartfona już nikogo nie dziwi. W wielu przypadkach urządzenia te muszą zbierać bardzo duże ilości danych oraz przysyłać je do centralnego punktu. Powoduje to ogromny wzrost ilości danych, które nie są w stanie zostać obsłużone przez jeden serwer. Powstaje więc potrzeba stworzenia niezawodnych, wydajnych i bezpiecznych systemów o wysokiej dostępności.

Technologie wirtualizacji <sup>1</sup> powstały w celu realizacji tych wymagań. Wirtualizacja serwerów dawno już wyparła tradycyjne serwery, które zostały zastąpione przez rozwiązania chmurowe. Niezależność sprzętowa, lepsza utylizacja zasobów, większe bezpieczeństwo, łatwa migracja danych i redukcja kosztów to tylko niektóre z wielu zalet wirtualizacji. Właśnie ta niezależność sprzętowa pozwala na coraz lepsze wykorzystanie urządzeń opartych na architekturze ARM, których głównymi zaletami są mały koszt i niewielki pobór mocy, a dzięki coraz lepszej optymalizacji systemów i postępującej miniaturyzacji, również rosnąca wydajność.

Proponowanym rozwiązaniem powyższych problemów będzie projekt o nazwie KubePi. Projekt ten skupia się na wirtualizacji opartej o kontenery Dockera <sup>2</sup> zarządzane przez system zarządzania klastrem Kubernetes i ma na celu stworzenie

---

<sup>1</sup>[?]

<sup>2</sup>[?]

rozproszonego systemu służącego do monitorowania otoczenia. Przykładowy klastr opierać się będzie na dwóch mikrokontrolerach RaspberryPi. Do pierwszego urządzenia będącego zarazem głównym węzłem klastra podłączone zostaną trzy czujniki: temperatury, wilgotności oraz alkoholu. Jego zadaniem będzie udostępnianie zbieranych informacji. Drugie urządzenie zostanie natomiast wyposażone w wyświetlacz LED <sup>3</sup>, co pozwoli na odczyt i wyświetlanie temperatury raportowanej przez pierwsze urządzenie. Dodatkowo w klastrze zostanie zainstalowana aplikacja webowa <sup>4</sup> pozwalająca na zdalny monitoring. W celu lepszego zobrazowania komunikacji między urządzeniami zostanie ona uruchomiona na drugim urządzeniu.

## 1.2 Metoda badawcza

## 1.3 Przegląd literatury w dziedzinie

## 1.4 Układ pracy

Celem pracy jest zaproponowanie architektury i sprawdzenie w działaniu rozproszonego systemu wysokiej dostępności służącego do monitorowania otoczenia.

Rozdział pierwszy zawiera szczegółowy opis problemu. Zostają w nim przedstawione różne problemy związane z wydajnością oraz bezpieczeństwem tradycyjnych rozwiązań, wraz z opisem metod badawczych użytych do analizy tematu. Podsumowane zostają również główne założenia i cele pracy. Na koniec przeprowadzony zostaje przegląd literatury związanej z tematem, z naciskiem na kluczowe zagadnienia dotyczące wirtualizacji, rozwiązań chmurowych oraz mikrokontrolerów opartych na architekturze ARM, wraz z krótkim opisem użytych źródeł.

W rozdziale drugim przybliżona zostaje tematyka systemów zarządzania klastrami pod kątem ich wymagań, bezpieczeństwa oraz komunikacji sieciowej. Kolejnym krokiem jest dokładniejsze zapoznanie się z wirtualizacją, a konkretniej wirtualizacją opartą o kontenery Dockera, co pozwoli lepiej zrozumieć ideę pracy. Następnie po krótkce przedstawione zostają tematyki związane z mikrokontrolerami oraz Internetem Rzeczy.

Rozdział 3 skupia się na analizie istniejących systemów zarządzania klastrami oraz ich pochodnych. Dodatkowo przedstawiona zostaje analiza kilku systemów typu Smart Home <sup>5</sup>.

---

<sup>3</sup>[?]

<sup>4</sup>[?]

<sup>5</sup>[?]

Kolejny rozdział opisuje fazę projektowania i implementacji projektu KubePi. Spisane zostają wymagania funkcjonalne aplikacji, a także ograniczenia projektowe. Wymienione i opisane zostają użyte technologie. Opisany zostaje proces konfiguracji urządzeń, sieci oraz systemu. Następnie wskazane zostają kluczowe punkty aplikacji wraz z kodem źródłowym i opisem. W kolejny kroku przechodzimy do fazy testów stworzonych aplikacji jak i całego systemu.

W podsumowaniu pracy opisane zostają słabe i mocne strony przedstawionego rozwiązania. Na podstawie uzyskanych wyników następuje ocena możliwości i przydatności zaproponowanego rozwiązania. Na końcu omówione zostają możliwe perspektywy rozwoju projektu.

## **Rozdział 2**

### **Podstawy teoretyczne**

#### **2.1 Podstawowe definicje**

##### **2.1.1 Wirtualizacja oparta na Linuksie**

##### **2.1.2 Wirtualizacja oparta o nadzorcę**

##### **2.1.3 Wirtualizacja oparta na kontenerach**

Docker

##### **2.1.4 Mikrokontroler RaspberryPi**

Protokoły komunikacji

##### **2.1.5 Kompilacja skrośna**

#### **2.2 Systemy zarządzania kontenerami**

##### **2.2.1 Kubernetes**

Architektura

##### **2.2.2 Docker Swarm**

##### **2.2.3 Resin.io**

## **Rozdział 3**

### **Projekt KubePi**

**3.1 Analiza wymagań**

**3.2 Użyte technologie**

**3.3 Projekt**

**3.4 Opis użytkowania**

**3.5 Przykład użycia**

**3.6 Możliwości rozszerzania aplikacji**



## **Rozdział 4**

### **Podsumowanie**

# Bibliografia

[1] Some books

## **Spis rysunków**

## **Spis tablic**