COMPUTER SCIENCES AND INFORMATION TECHNOLOGIES

UDC 621.382.2 (045)
DOI:10.18372/1990-5548.76.17661

I. V. Zakutynskyi

FINDING THE OPTIMAL NUMBER OF COMPUTING CONTAINERS IN IOT SYSTEMS: APPLICATION OF MATHEMATICAL MODELING METHODS

Faculty of Air Navigation, Electronics and Telecommunications,
National Aviation University, Kyiv, Ukraine
E-mail: ihor.zakutynskyi@nau.edu.ua ORCID 0000-0003-2905-3205

***Abstract*—*The integration of computing containers into Internet of Things (IoT) systems created a lot of challenges and opportunities in the connected devices and cloud computing industries. In this paper, the author proposed a mathematical modeling method to analyze and optimize the deployment of computing containers into an IoT-based ecosystem. By implementing mathematical modeling techniques, such as queuing theory, optimization algorithms, and statistical analysis, we aim to address key concerns related to resource allocation, workload distribution, and performance optimization. Proposed models take the dynamic nature of an IoT system, considering factors such as real-time data streams and varying workloads for the satisfaction of scalability requirements. The author aids in identifying the optimal placement strategies for computing containers, ensuring efficient resource utilization and workload balancing across the IoT network.*

Index Terms—Internet of Things; cloud computing; computing containers; mathematical modeling; performance optimization; resource allocation.

I. INTRODUCTION

In recent years, the integration Internet of Things (IoT) solutions has revolutionized various domains, ranging from smart homes and healthcare to industrial automation and transportation. These systems consist of interconnected devices, sensors, and actuators that generate massive amounts of data. IoT systems require efficient and scalable architectures to handle the increasing complexity and heterogeneity of devices. IoT devices generate an enormous amount of data, requiring efficient and scalable computing solutions. Computing containers, such as Docker and Kubernetes, have emerged as promising technologies for managing and deploying applications in IoT systems. These containers encapsulate software components along with their dependencies, providing isolation, portability, and scalability.

To effectively design and optimize IoT systems that employ computing containers, it is crucial to develop mathematical models that capture their behavior and performance characteristics. Mathematical modeling enables us to analyze and understand the intricate interactions between IoT devices, the underlying network infrastructure, and the computing containers that orchestrate the system's computational tasks.

The objective of this paper is to present some mathematical modeling methods for computing containers in the context of IoT systems. By leveraging mathematical techniques, we aim to address critical challenges associated with resource allocation, task scheduling, energy consumption, and performance optimization.

In this study we will focucs on the key factors influencing the performance of computing containers, including resource utilization, task latency, and scalability.

Proposed models will consider parameters such as the number of containers, available resources, network bandwidth, and the characteristics of the underlying IoT devices. Also, we will analyze the advantages and limitations of the mathematical modeling approach for improving the efficiency and scalability of computing containers in IoT-based systems.

II. PROBLEM STATEMENT

The deployment and management of computing containers in IoT systems give rise to several critical challenges that need to be addressed. These challenges include.

A. Resource Utilization and Scalability

IoT systems typically consist of a large number of resource-constrained devices with varying computational capabilities. Optimally allocating computing resources to containerized applications in such systems is crucial to ensure efficient utilization and scalability. However, determining the optimal

resource allocation strategy requires a comprehensive understanding of the system characteristics and the behavior of containerized applications. Mathematical modeling techniques can provide insights into resource requirements, workload patterns, and scalability considerations, thereby enabling efficient resource utilization.

B. Performance Analysis and Optimization

Containerized IoT applications often need to meet stringent performance requirements, such as low latency and high throughput, to ensure timely data processing and decision-making. However, predicting and optimizing the performance of containerized applications in complex IoT environments is a challenging task. Mathematical models can capture the dynamics of application performance, allowing to analyze and optimize key performance metrics, such as response time, throughput, and resource utilization.

C. Workload Characterization

Understanding the workload characteristics of IoT applications is crucial for effective resource management and performance optimization. IoT workloads are typically heterogeneous, dynamic, and unpredictable, making it challenging to capture their behavior accurately. Mathematical modeling techniques, such as stochastic modeling and queuing theory, can aid in characterizing IoT workloads, enabling the estimation of resource requirements and the identification of potential performance bottlenecks.

D. Container Placement and Scheduling

Determining the optimal placement and scheduling of containerized applications in IoT systems is critical for efficient resource utilization and load balancing. The placement and scheduling decisions must consider factors such as network connectivity, computational capabilities of devices, and application dependencies. Mathematical optimization algorithms can assist in finding optimal or near-optimal solutions to the container placement and scheduling problem, taking into account various constraints and objectives.

III. THE LITERATURE REVIEW

In recent years, researchers have proposed many mathematical models to study computing containers behavior. These models consider parameters such as computing resource allocation, workload distribution, and deployment strategies. Some studies [1] – [4] focus on optimizing resource allocation strategies based on workload characteristics and client's (IoT devices in our case) capabilities.

Others investigate the impact of container migration and orchestration policies on system performance overall. For example, in this paper "Mathematical model for searching the optimal resources size for the virtual service node" [5] authors provide a mathematical model for calculating the optimal size of computing resources, such as CPU, memory, etc, based on the architecture of cloud computing.

Also, in cloud computing environments, database systems hold a significant position as a crucial class of services. The performance optimization for database resources is proposed in this paper [2] "Mathematical model for higher utilization of database resources in cloud computing". The author provides an approach based on mathematical formulation and linear programming methodology to optimize database performance.

Another important research topic is mathematical models for evaluating and ensuring security for cloud computing systems. In this study "A Mathematical Model for Securing Cloud Computing" [3] the authors provide solutions based on mathematical modeling methods for cloud security framework.

This research employed simulation-based approaches, mathematical analysis, and empirical evaluations to validate the proposed models.

IV. COMPUTING CONTAINERS SCALING

Computing containers operate as a higher-level abstraction within the application layer, facilitating the bundling of code and its associated dependencies. With the ability to execute numerous containers on a single machine while sharing the operating system (OS) kernel, each container functions autonomously as an isolated process within the user space. In contrast to virtual machines (VM), containers exhibit reduced space requirements, with container images typically spanning tens of megabytes (MB) in size. This compactness enables containers to accommodate a larger number of applications while necessitating fewer VM and operating systems, thereby optimizing resource utilization. This is particularly useful in IoT systems where multiple applications or services need to run on the same device without conflicts.

In Figure 1 are listed typical schemes for containerized applications.

For simplicity, this scheme is presented for a one-server system, but also containers can be distributed between several physical machines for horizontal scaling. Figure 2 gives a general scheme for an IoT system with distributed and containerized applications.

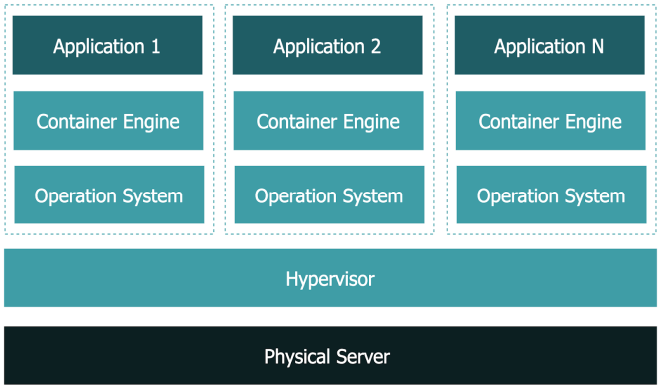

Fig. 1. Containerized Application Scheme

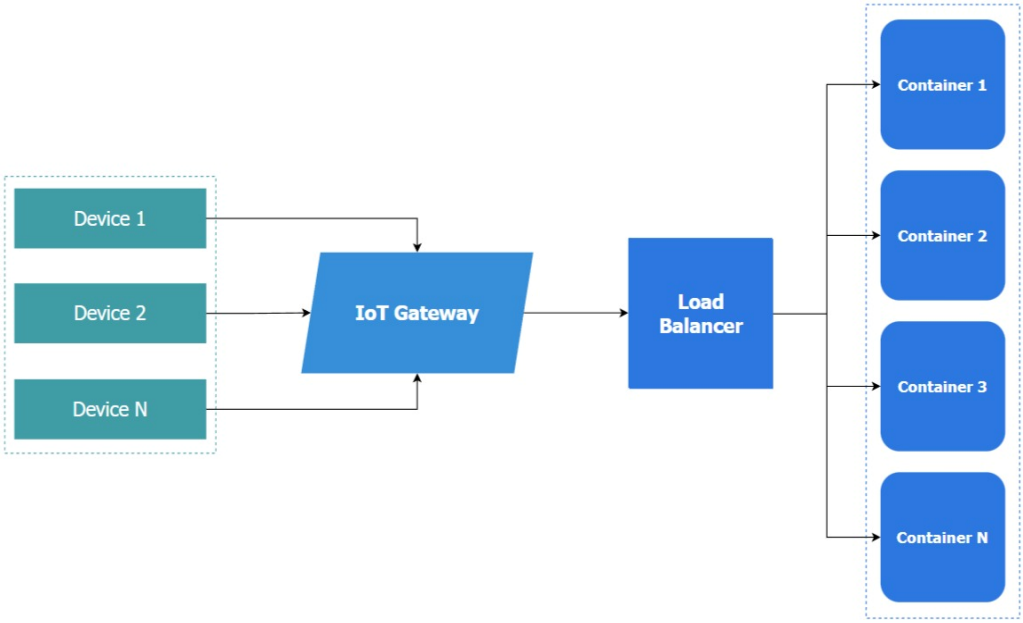

Fig. 2. IoT system with distributed computing containers

Finding the optimal number of computing containers for an IoT system involves determining the right balance between resource utilization and system performance. Below are listed methods for optimal container number finding based on Queuing theory and the Stochastic model. Also, proposed a new method based on the Mixed Integer Linear Programming (MILP) model.

A. Queuing theory-based method

In this section, presented a mathematical model based on queuing theory for analyzing the scaling of computing containers in IoT systems. This model aims to evaluate the impact of scaling the number of containers on system performance, including response time and resource utilization.

$$P_i = \left(1+\frac{\lambda}{\mu}\right)^{-N} \times \frac{\sum_{i=0}^{N-i}\left(\frac{\lambda}{\mu}\right)^i}{i!} + \left(\frac{\lambda}{\mu}\right)^i \times \frac{N!}{N\left(\mu-\lambda\right)^N}. \quad (1)$$

Based on (1), the probability of no requests in the system (P_0) can be calculated by substituting $i = 0$. This formula provides the probability of having zero requests in the system. It serves as a key parameter for evaluating the performance metrics of the queuing system, such as the average number of requests in the system and in the queue, as well as the average waiting time in the system and in the queue.

In the algorithm, the arrival rate is provided as an input to the scaling analysis. It is used in conjunction with the service rate (μ) and the number of computing containers (N) to calculate performance metrics such as the average number of requests in the system (L), the average number of requests in the queue (L_q), the average waiting time in the system (W), and the average waiting time in the queue (W_q).

By analyzing the impact of different arrival rates on the system performance, we can determine the

optimal number of computing containers required to handle incoming requests efficiently, ensuring that desired performance targets are met.

Input:

λ – arrival rate.

μ – service rate of a single container.

N – number of computing containers.

Output:

L – average number of requests in the system.

L_q – average number of requests in the queue.

W – average waiting time in the system.

W_q – average waiting time in the queue.

1) Calculate the service rate of a single container:

$$\mu \leftarrow \frac{1}{S}, \tag{2}$$

where S is the average service time.

2) Calculate arrival rate:

$$\lambda \leftarrow \frac{L}{W}. \tag{3}$$

3) Calculate probability of no requests in the system:

$$P_0, \text{ based on (1).} \tag{4}$$

4) Calculate average number of requests in the system:

$$L \leftarrow \frac{\lambda}{\mu - \lambda} \times P_0. \tag{5}$$

5) Calculate average number of requests in the queue:

$$L_q \leftarrow \frac{\lambda^2}{\mu(\mu - \lambda)} \times P_0. \tag{6}$$

6) Calculate average waiting time in the system:

$$W \leftarrow \frac{L}{\lambda}. \tag{7}$$

7) Calculate average waiting time in the queue:

$$W_q \leftarrow \frac{L_q}{\lambda}. \tag{8}$$

In this algorithm, μ refers to the rate at which a single computing container can process requests in the current IoT system state. It represents the average number of requests per unit of time that a single container can handle. This is an essential parameter in the queuing theory. It helps determine the capacity and efficiency of the computing containers in processing incoming requests.

By varying the number of containers (N) and observing the corresponding changes in performance metrics, we can analyze the impact of scaling on the system's ability to handle incoming application requests. This analysis can provide insights into the optimal number of containers required to achieve desired performance targets, balancing resource utilization and response time.

B. Stochastic model-based method

Let $X(t)$ denote the state of the system at time t, representing the number of active computing containers. The system can have a finite number of states, ranging from 0 to N, where N is the maximum number of containers. The state transitions occur based on arrival and departure rates of requests, and can be modeled using Markov chains.

The transition rates, denoted as $a_{i,j}$, represent the probability of transitioning from state i to state j. These rates can be modeled using suitable stochastic models such as Poisson processes. By constructing the transition rate matrix $\mathbf{A}$, with elements $a_{i,j}$ the system behavior can be analyzed.

The steady-state probabilities of the Markov chain, denoted as $\pi = [\pi_0, \pi_1, \ldots, \pi_N]$, represent the probabilities of being in each state. These probabilities satisfy the equation $\pi \cdot \mathbf{A} = 0$, subject to the normalization condition $\sum_{i=0}^{N} \pi_i = 1$.

Solving this system of equations provides insights into the behavior of the system, such as container utilization, system throughput, and resource allocation.

Using the steady-state probabilities, performance metrics can be derived to evaluate the system's behavior. Metrics such as the average number of containers, average waiting time, and system throughput can be calculated based on steady-state probabilities and transition rates (Fig. 3).

C. MILP model-based method

Below we provide formulation of the problem as a mathematical optimization model, specifically a mixed-integer linear programming (MILP) problem. The objective function and constraints are defined to express the optimization problem, where the objective is to minimize the number of computing containers while satisfying the processing requirements and the communication rate constraints (Fig. 4).

Algorithm 1 Containers Scaling Model

Input: N, λ , μ
Output: Steady-state probabilities π_i for $0 \leq i \leq N$
Construct the transition rate matrix $\mathbf{A}$ with size $(N+1) \times (N+1)$
for $i = 0$ to N **do**
  for $j = 0$ to N **do**
    if $i = j$ **then**
      $a_{i,j} \leftarrow -(\lambda + i\mu)$
    else if $j = i + 1$ **then**
      $a_{i,j} \leftarrow \lambda$
    end if
  end for
end for
Solve $\pi \cdot \mathbf{A} = \mathbf{0}$ subject to $\sum_{i=0}^{N} \pi_i = 1$
Return Steady-state probabilities π_i for $0 \leq i \leq N$

Fig. 3. Containers Scaling model based on stochastic model

Algorithm 2 Optimal Number of Computing Containers

Input: N, P_i, T, C, R, CPU, M
Output: S
Set $S \leftarrow 1$
Calculate total workload demand: $D \leftarrow \frac{1}{C \cdot S} \sum_{i=1}^{N} P_i$
Assign IoT devices to containers
Calculate maximum workload demand: $M \leftarrow \max\left\{\frac{1}{S} \sum_{i=1}^{N} P_i\right\}$
while $M > D$ **do**
  Increase number of containers: $S \leftarrow S + 1$
  Update workload demand: $D \leftarrow \frac{1}{C \cdot S} \sum_{i=1}^{N} P_i$
  Calculate maximum workload demand: $M \leftarrow \max\left\{\frac{1}{S} \sum_{i=1}^{N} P_i\right\}$
end while
return S

Fig. 4. Optimal Number of Computing Containers Based on Connected IoT Devices

Input:

N – total number of IoT devices.
P_i – processing requirement of device i,
$\forall_i \in [1, N]$.
T – communication rate.
C – capacity of a single computing container.
CPU – Number of CPUs in computing container.
M – available memory in each container.

Output:

S – total number of computing containers.

V. CONCLUSIONS

Proposed mathematical model provides a straightforward and intuitive approach to determine the optimal number of computing containers for an IoT system.

Model can handle varying numbers of devices and their processing requirements, making it adaptable to different system configurations.

A. Advantages of the proposed method

1) Resource-aware: The model takes into account the specific parameters of the computing containers, such as RAM, number of CPUs, and memory, ensuring that the resource constraints of the containers are respected.

2) Quick estimation: The iterative improvement approach allows for a relatively quick estimation of the optimal number of containers, which can serve as a starting point for resource allocation decisions.

B. Disadvantages of the proposed method

1) Heuristic nature: The model uses an iterative improvement approach, which may not guarantee finding the globally optimal solution. The solution obtained may be suboptimal or near-optimal depending on the workload distribution and resource constraints.

2) Sensitivity to initial conditions: The performance of the algorithm can be sensitive to the

initial number of containers chosen. Starting with an inappropriate initial number of containers may lead to a longer convergence time or suboptimal solutions.

Overall, while this method provides a practical and resource-aware approach for determining the optimal number of computing containers, it is important to consider its heuristic nature and potential limitations when applying it to real-world IoT systems with varying requirements and constraints. Further analysis and refinement may be necessary to obtain more precise solutions or to incorporate additional factors specific to the system's requirements.

REFERENCES

[1] X. Zou, "Research on cloud computing task scheduling based on calculus mathematical equation," *In Highlights in Science, Engineering and Technology*, vol. 9, 2022, pp. 218–226. Darcy & Roy Press Co. Ltd. https://doi.org/10.54097/hset.v9i.1779

[2] P. R. Kaveri, & P. Lahande, "Reinforcement Learning to Improve Resource Scheduling and Load Balancing in Cloud Computing," *In SN Computer Science*, vol. 4, Issue 2, 2023. Springer Science and Business Media LLC. https://doi.org/10.54097/hset.v9i.1779

[3] R. Tasneem, & M. A. Jabbar, "An Insight into Load Balancing in Cloud Computing," *In Proceeding of 2021 International Conference on Wireless Communications, Networking and Applications*, 2022, pp. 1125–1140. Springer Nature Singapore. https://doi.org/10.1007/978-981-19-2456-9_113

[4] R. Alakbarov, "An Optimization Model for Task Scheduling in Mobile Cloud Computing," *In International Journal of Cloud Applications and Computing*, vol. 12, Issue 1, pp. 1–17, 2022. IGI Global. https://doi.org/10.4018/IJCAC.297102

[5] M. Skulysh, "Mathematical model for searching the optimal resources size for the virtual service node," *Advanced Information Systems*, 2(2), 2018, pp. 30–34. https://doi.org/10.20998/2522-9052.2018.2.05

[6] P. R. Kaveri, & V. Chavan, "Mathematical model for higher utilization of database resources in cloud computing," *In 2013 Nirma University International Conference on Engineering (NUiCONE). 2013 Nirma University International Conference on Engineering (NUiCONE), IEEE*, 2013. https://doi.org/10.1109/NUiCONE.2013.6780095

[7] Zico Mutum, *A Mathematical Model for Securing Cloud Computing*, 2015.

Received March 09, 2023

Zakutynskyi Ihor. ORCID 0000-0003-2905-3205. PhD student.
Radio Electronic Devices and Systems Department, Faculty of Air-navigation, Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine.
Education: National Aviation University, Kyiv, Ukraine, (current time).
Research area: neural networks, software architecture, automation systems, cloud computing, IoT systems.
Publications: 7.
E-mail: ihor.zakutynskyi@nau.edu.ua

I. B. Закутинський. Визначення оптимальної кількості обчислювальних контейнерів в системах Інтернету речей: застосування методів математичного моделювання.

Інтеграція обчислювальних контейнерів у системи Інтернету речей (IoT) створює багато можливостей, водночас багато викликів для індустрії розумних пристроїв, а також для галузі серверних та хмарних технологій. Важливим завданням є вибір оптимальної кількості обчислювальних ресурсів, а також можливість їх адаптації до робочого навантаження. У цій статті пропонується підхід на основі математичного моделювання для аналізу та оптимізації ресурсів обчислювальних контейнерів у системах IoT. Використовуючи математичні методи, такі як теорія масового обслуговування, алгоритми оптимізації та статистичний аналіз, запропонована моделі для розв'язання проблем пов'язаних із розподілом ресурсів, а також визначення оптимальної кількості активних обчислювальних контейнерів. Запропоновані моделі враховують динамічну природу систем IoT, а отже враховуючи такі фактори, як потоки даних у реальному часі, зміну робочого навантаження, а також враховують вимоги до масштабованості. Впровадження запропонованих моделей дозволить забезпечити ефективне використання обчислювальних ресурсів, а також забезпечити балансування робочого навантаження в системах Інтернету речей.

Ключові слова: інтернет речей; хмарні обчислення; обчислювальні контейнери; математичне моделювання; оптимізація продуктивності; розподіл ресурсів.

Закутинський Ігор Володимирович. ORCID 0000-0003-2905-3205. Аспірант.
Кафедра електроніки, робототехніки і технологій моніторингу та Інтернету речей, Факультет аеронавігації, електроніки та телекомунікацій, Національний авіаційний університет, Київ, Україна.
Освіта: Національний авіаційний університет, Київ, Україна, (2019).
Напрям наукової діяльності: нейронні мережі, архітектура програмного забезпечення, системи автоматизації, хмарні обчислення, системи інтернету речей.
Кількість публікацій: 7.
E-mail: ihor.zakutynskyi@nau.edu.ua