**Кирил Мугуєв, Володимир Русінов**

РОЗРОБКА МАСШТАБОВАНОЇ AI-ПЛАТФОРМІ НА ОСНОВІ ІНТЕГРАЦІЇ EDGE ОБЧИСЛЕНЬ ТА CLOUD ТЕХНОЛОГІЙ

**Kyryl Muhuiev, Volodymyr Rusinov**

DEVELOPMENT OF A SCALABLE AI PLATFORM BASED ON INTEGRATION OF EDGE COMPUTING WITH CLOUD TECHNOLOGIES

У статті розглядається питання використання гетерогенних систем CPU-GPU для прискорення вирішення задач, пов’язаних з навчанням нейронних мереж.

**Ключові слова:** ШІ, хмари, масштабування.

Рис.: 3. Табл.: 1. Бібл.: 9.

The article describes and analyzes the process of development of an AI-platform using contemporary cloud technologies.

**Keywords:** AI, cloud, scalability.

Fig.: 3. Tab.: 1. Bibl.: 9.

**Relevance of the research topic.** Latest trends show that AI is becoming more permeant throughout all spheres of life. One of such wide spheres is Edge computing and how to accurately leverage its pros and cons to deliver quality services to the end user. Contemporary Cloud systems have numerous datacenters throughout the globe to increase productivity and improve latency, however, usually, they do not offer the same flexibility, in terms of general user experience as well, for end users and companies that operate them.

To measure the overall usability of the system with a certain degree of accuracy we will introduce several metrics that will help us establish the parameters of the system and help us compare it with the alternative solutions.

**Target setting.** Deployment of AI software must be accompanied by a thorough analysis of the target system. This article will analyze different approaches to AI deployment on Edge systems. One will involve a smaller model, fit to run on low-powered devices and the other one will be a full-scale model. Several tests will be conducted to measure the time and accuracy metrics of the proposed system.

**Actual scientific researches and issues analysis.** Scientific research in areas related to improving the speed and accuracy of delivery of AI-supported systems is not widely researched. Edge and fog computing technologies offer a broad range of performance optimization options; however, little research has been published regarding running models on Edge/Fog platforms.

**Uninvestigated parts of general matters defining.** Contemporary solutions, usually, involve using standalone IoT devices [1] or Cloud solutions [2]. Edge computing is a relatively novel model of creating a device network, thus creating demand for solutions capable of fast deployment and scaling capacity. Most research effort goes into creating a more capable AI model or slimming down the model to attempt to run it onboard the IoT device. Little ongoing research involves using Cloud to facilitate AI permeation and exploring ways to improve the quality of service.

**The research objective.** The goal of the article is to determine the feasibility of Edge computing integration with Cloud computing and to see how well it performs using AI tasks.

**The statement of basic materials.** The basic idea of the proposed system is creating a pipeline that will allow it to effectively bypass any bottlenecks posed by the IoT design. The current state of IoT, although it allows for relatively capable GPUs onboard ASICs, it does not have nearly enough penetration into the sphere, thereby leaving behind IoT devices without onboard GPUs. To address this issue, we propose taking advantage of Cloud technologies as well as ever-increasing connectivity, through LTE, 5G, or simply Wi-Fi.

Raspberry Pi 3B+ will be taking the role of an IoT device. It is important to note that any IoT device may be used in its stead, that is capable to reproduce, at least to some degree, its capabilities. The device is connected to the Internet via Wi-Fi, the most likely case for a device with such a role. It has an ARM Cortex-A53 SoC with 1 GB of LPDDR2 RAM. SD card takes the role of persistent storage. [3]

What kind of bottlenecks does this device have? First, its SoC, which acts as both CPU and GPU for the device, is by design low power and its capabilities are limited. This allows only for simpler tasks to be run on the system, and for our purposes, simpler ML models. Secondly, IoT devices are limited by power, therefore their capacity for additional hardware is limited. To minimize the effect of this issue, usually, onboard network devices are low-powered, significantly decreasing their throughput.

To alleviate some of the aforementioned issues, a Cloud system will be introduced. Modern Cloud systems and providers allow to create and deploy different systems with different configurations with relative ease, when compared to onsite GRID systems, and also allow for high fault-tolerance, with uptime of the most of their services being above 99.99%. We will be using Amazon Web Services (AWS) as the main cloud provider. [4] It is one of the three biggest providers available, with GCP and Azure being the main competitors that also provide the services displayed in this article.



Figure 1. Proposed system architecture

As shown in the diagram above, multiple different cloud services will be used to create an AI pipeline. IoT Core is responsible for controlling MQTT traffic using rules. Once appropriate data reaches IoT Core from the IoT device, it will automatically trigger Lambda which will send it to EC2 Instance. Depending on the contents of the data it will either be stored onto DynamoDB or be processed by the more sophisticated AI model on the instance. Another Lambda function is triggered every time new data arrives from the corporate network and reaches one of the endpoints defined in the API Gateway. [4]

Brewer’s (CAP) theorem states that any scalable distributed data system can only guarantee two of three properties: Consistency, Availability, and Partition tolerance. Traditional RDBMS sacrifice availability and NoSQL solutions sacrifice consistency. The proposed system will utilize BASE, therefore sacrificing consistency. However, in our system, we will guarantee eventual consistency, at least in terms of general awareness of new data. [6]

As a base communication protocol between IoT device and Cloud MQTT will be used. It is lightweight, however, allows for quality of service features, like integrity checking. Research shows that it performs better in IoT tasks, like sending sensor data from a device to the server. It follows the pub-sub model, which is great for the purposes of this article as we do not wish to waste resources on maintaining the connection. In the table below, there is a comparison between HTTP and MQTT message throughput. [7] This test emulates the worst-case scenario when the sender has to send the same packet, consisting of an arbitrary 2-byte JSON message.

Table 1

MQTT/HTTP comparison

|  |  |  |
| --- | --- | --- |
| Raspberry PI 3B+ | HTTP | MQTT |
| Messages | 9243 | 11728 |
| Messages/second | 0.97 | 1.15 |
| CPU usage | 65% | 60% |
| Power consumption | 0.22 MAh | 0.17MAh |

The dataset contains 2,029 companies with 31 indicators (columns) each. 25 of the indicators are financial data used for further forecasting of the credit rating. [5] 80 percent of the dataset was used for training, and the other 20 percent was used for validation and testing. For the purposes of modeling, data was cleaned from outliers, and data was normalized. To further streamline the data, MinMaxScaler is used to balance out the data points to be in-between values of ‘0’ and ‘1’. Columns with nonessential information like company name were subsequently removed.

As a base preemptive model that will be used on the IoT device, we will use Logistical Regression. Sophisticated models like Neural Networks can be trained and deployed on Edge systems; however, they do not possess enough resources to do so efficiently, and therefore their usage becomes less reliable at a higher cost.

 (1)

Neural network acts as a more sophisticated model, deployed on the Cloud, with it being much more resource-consuming. Table 1 describes the difference in resource consumption. To create an effective neural network architecture we need to consider several key issues: how many layers and neurons should be in the network, hyperparameter tuning, regularization, activation function, and optimization. [8] There are even more techniques that can be used to improve the performance of the proposed neural network, however, it is out of the scope of this text.



Figure 2. Proposed ANN architecture

The final neural network is composed of 1,888 neurons split into 6 layers as shown in figure 2. Between them, there are also Dropout layers that "turn off" random neurons as well as L2 regularization to prevent overtraining. The choice of hyperparameters was determined empirically. The main hyperparameters that were adjusted during neural network training are subsample size, learning rate, and optimizer. Adam was used as an optimizer. Adam (Adaptive Moment Estimation) is an adaptive optimizer that uses first- and second-order gradients to update neural network weights. For every except the last layer, swish function, which is derived from ReLU, is used, with slightly better results compared to ReLU for classification tasks. [9] The early stopping technique was implemented to save time.



Fig. 3. Processing time for AI task on Raspberry PI and PC

 As shown in Figure 3, PC, simulating the VM environment of the Cloud PaaS, is much faster, however, it still requires significantly more time to process the data using a more sophisticated model. Therefore, the logistical regression model can be used to make a quick assessment of the data, losing in accuracy however getting results significantly faster. The ANN model can be tuned further to allow for real-time results, however, this was not the task of this article. The resulting accuracy of logistical regression is 40.43% and ANN scores 65.19%.

The resulting system may be applied in a variety of existing industries in order to obtain tangible economical and social benefits at a relatively low cost (especially for the large-scale deployment enviroinment).

 The most prominent industries for the usage of the resulting system include, but are not limited to:

1. Manufacturing industry – use-cases include monitoring production lines, detecting potential defects, optimizing and streamlining quality control procedures and enabling predictive maintenance of machinery and equipment by analyzing real-time sensor data at the edge, the platform can predict equipment failures before they occur, minimizing unplanned downtime.;
2. Healthcare industry – use-cases include the implementation of interconnectivity of medical devices, wearable devices and remote monitoring systems in order to perform real-time analysis of vital signs, identify anomalies, assist in early diagnosis, and enable personalized medicine without patients being forced into on-site monitoring scenarios. By analyzing patient data at the edge, the platform can support personalized treatment plans, medication recommendations, and disease management, leading to more effective and efficient healthcare delivery;
3. Retail industry – use-cases include enabling personalized recommendations based on the data collected from smart shelves, cashier-less checkout systems in the remote locations and inventory management with real-time tracking and demand prediction functionality. Additionally, the sysetem can analyze customer demographics, preferences, and purchasing patterns to deliver targeted marketing campaigns and promotions, leading to improved marketing ROI and increased sales both in long-term and short-term period;
4. Transportation and logistics industry – use-cases include the traffic management optimization models, smart transportation systems, truck fleet performance monitoring and, ultimately, the creation of autonomous vehicles with real-time perception and decision-making capabilities;
5. Energy and utilities industry – use-cases include optimizing energy consumption by using pattern prediction algoritms, enabling predictive maintenance of infrastructure, improving preventive fault detection and supporting smart grid management initiatives.

**Conclusion.** This study shows a way to improve the efficiency of onboard AI on IoT devices using Edge architecture. To illustrate this, a system was developed, and it has shown to be capable of fast delivery of ML results over the network. By utilizing Cloud computing resources, we were able to deploy such a system much faster, than by using a conventional on-site cluster, however, such approach is not limited to Cloud.

There is room for improvement. The accuracy of the DNN can be increased through different means, for example, further hyperparameter tuning or making changes to the architecture might bring benefits. Using RNNs might be useful for time-sensitive data, or other datasets can be used to improve accuracy by augmenting the initial dataset. All of this can be easily done on the cloud. This research proves that we can bolster edge computing capabilities with cloud services.

**Uninvestigated parts.** Reliability of this solution may be coupled with its design architecture. It is worthwhile using the proposed method using different setups, for example: scaling up the amount of EC2 instances, to allow for a bigger data bandwidth, creating a separate service to manage data. The on-premises solution presented in this publication might be improved by using tuning the settings and topological organization of the network. Perhaps, more of the Cloud services could be client-side to cut costs. Lastly, more research should conducted to find out the financial gains of a system derived from the one presented.

**References**

1. Xiao, L., Wan, X., Lu, X., Zhang, Y., & Wu, D. (2018). IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?. IEEE Signal Processing Magazine, 35(5), 41-49.
2. Surya, L. (2018). Streamlining cloud application with AI technology. International Journal of Innovations in Engineering Research and Technology [IJIERT] ISSN, 2394-3696.
3. Raspberry PI Documentation [Електронний ресурс]. – Режим доступу: https://www.raspberrypi.com/documentation/
4. AWS Documentation [Електронний ресурс]. – Режим доступу: https://docs.aws.amazon.com/
5. Corporate Credit Rating [Електронний ресурс]. – Режим доступу: https://www.kaggle.com/datasets/agewerc/corporate-credit-rating
6. De Angelis, S., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., & Sassone, V. (2018). PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain. In CEUR workshop proceedings (Vol. 2058). CEUR-WS.
7. Yokotani, T., & Sasaki, Y. (2016, September). Comparison with HTTP and MQTT on required network resources for IoT. In 2016 international conference on control, electronics, renewable energy and communications (ICCEREC) (pp. 1-6). IEEE.
8. Allen-Zhu, Z., Li, Y., & Liang, Y. (2019). Learning and generalization in overparameterized neural networks, going beyond two layers. Advances in neural information processing systems, 32.
9. Dasgupta, R., Chowdhury, Y. S., & Nanda, S. (2021). Performance comparison of benchmark activation function relu, swish and mish for facial mask detection using convolutional neural network. In Intelligent Systems: Proceedings of SCIS 2021 (pp. 355-367). Springer Singapore.

**РОЗШИРЕНА АНОТАЦІЯ**

**Кирило Мугуєв, Володимир Русінов**

РОЗРОБКА МАСШТАБОВАНОЇ AI-ПЛАТФОРМІ НА ОСНОВІ ІНТЕГРАЦІЇ EDGE ОБЧИСЛЕНЬ ТА CLOUD ТЕХНОЛОГІЙ

**Актуальність теми дослідження.** Останні тренди показують, що ШІ стає більш розповсюдженим у всіх сферах життя. Одна така широка сфера це Edge обчислення, та як найбільш точно використовувати їх переваги та недоліки для того, щоби доставити якісні сервіси до кінцевого користувача. Сучасні хмарні системи використовують значну кількість датацентрів, які знаходяться по всьому земному шару, для того щоби підвищити продуктивність та покращити затримку, проте, часто, вони не пропонують гнучкості, з точки зору користувацького досвіду, як для користувачів так і для компаній, які оперують ними.

Щоби максимально точно отримати розуміння на скільки може система бути використана, ми вводимо декілька метрик, які допоможуть отримати параметри системи, за якими можна порівняти запропоновану систему з альтернативними рішеннями.

**Постановка проблеми.** Розгортання програмного забезпечення ШІ має супроводжуватися ретельним аналізом цільової системи. У цій статті буде проаналізовано різні підходи до розгортання ШІ в системах Edge. Один буде включати меншу модель, придатну для роботи на малопотужних пристроях, а інший буде повномасштабною моделлю. Буде проведено декілька тестів для вимірювання показників часу та точності запропонованої системи.

**Аналіз останніх досліджень і публікацій.** Наукові дослідження в областях, пов’язаних із підвищенням швидкості та точності доставки систем, що підтримують ШІ, не досліджуються широко. Технології обчислень Edge і Fog пропонують широкий спектр оптимізації продуктивності, однак було опубліковано мало досліджень щодо запуску моделей ШІ на платформах Edge/Fog.

**Постановка завдання.** Мета статті — визначити доцільність інтеграції Edge-обчислень із хмарними обчисленнями та побачити, наскільки добре вони виконуються за допомогою завдань ШІ.

**Викладення основного матеріалу.** Сучасні рішення, як правило, передбачають використання автономних пристроїв IoT [1] або хмарних рішень [2]. Граничні обчислення є відносно новою моделлю створення мережі пристроїв, що створює попит на рішення, здатні до швидкого розгортання та масштабування потужності. Більшість дослідницьких зусиль спрямовується на створення більш потужної моделі штучного інтелекту або зменшення моделі, щоб спробувати запустити її на пристрої IoT. Невеликі поточні дослідження включають використання хмари для сприяння проникненню штучного інтелекту та вивчення шляхів покращення якості обслуговування. Отриману систему можна застосувати в різноманітних існуючих галузях промисловості, для того щоб отримувати значні економічні та соціальні переваги при відносно низькій вартості впровадження.

**Висновки.** Це дослідження показує спосіб підвищення ефективності вбудованого штучного інтелекту на пристроях IoT за допомогою архітектури Edge. Щоб проілюструвати це, було розроблено систему, яка продемонструвала здатність швидко доставляти результати ML через мережу. Використовуючи ресурси хмарних обчислень, ми змогли розгорнути таку систему набагато швидше, ніж використовуючи звичайний локальний кластер, однак такий підхід не обмежується хмарою.

Є місце для вдосконалення. Точність DNN можна підвищити за допомогою різних засобів, наприклад, подальше налаштування гіперпараметрів або внесення змін до архітектури може принести переваги. Використання RNN може бути корисним для чутливих до часу даних, або інші набори даних можна використовувати для підвищення точності шляхом розширення початкового набору даних. Все це можна легко зробити в хмарі. Це дослідження доводить, що ми можемо розширити передові обчислювальні можливості за допомогою хмарних сервісів.

**Невивчені аспекти.** Надійність цього рішення може бути пов'язана з його архітектурою. Цілком варто дослідити запропонований метод з різними налаштуваннями, наприклад: збільшення кількості інстансів EC2 для забезпечення більшого пропускного потоку даних, створення окремого сервісу для управління даними. Рішення в обласній інфраструктурі, представлене у цій публікації, може бути покращене за допомогою налаштування параметрів та топологічної організації мережі. Можливо, більше хмарних сервісів можуть бути розміщені на клієнтському боці задля зменшення витрат. Нарешті, слід провести додаткові дослідження для визначення фінансових вигід системи, отриманої за даною методикою.

**Ключові слова:** ШІ, хмари, масштабування.