

**МИКРОСЕРВИСНО РЕШЕЊЕ ЗА ПРЕДВИЂАЊЕ ПОТРОШЊЕ ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ У CLOUD ОКРУЖЕЊУ****A MICROSERVICE SOLUTION FOR PREDICTING ELECTRICITY POWER CONSUMPTION IN A CLOUD ENVIRONMENT**

Душан Носовић, Факултет техничких наука, Нови Сад

**Oblast – ПРИМЕЊЕНО СОФТВЕРСКО ИНЖЕЊЕРСТВО**

**Kratak sadržaj** – Опис, анализа и приказ резултата времена извршавања тренинга модела машинског учења и предвиђања потрошње електричне енергије. Приказани резултати времена извршавања су тестирани на различитим Cloud computing платформама које се заснивају на PaaS моделу Cloud computing-a.

**Кључне речи:** Cloud computing, Микросервисна архитектура, Platform as a Service, Машинско учење.

**Abstract** – Description, analysis and presentation of machine learning model training execution time results and prediction of electricity power consumption. Presentation of execution time results tested on various Cloud computing platforms based on the PaaS model of Cloud computing.

**Keywords:** Cloud computing, Microservice architecture, Platform as a service, Machine learning.

**1. УВОД**

У претходних неколико година долази до убрзаног развоја технике и технологије, чија је последица настанак нових технологија, што доводи до промена у свим сферама софтверске индустрије. Једну од најважнијих и најпримењенијих нових технологија представља Cloud computing, који доводи до есенцијалних промена у рачунарству. Cloud computing омогућава стварање нових архитектура и апликација, и утиче на начин на који размишљамо о развоју и примени софтвера. Битност Cloud computing технологије потврђују водеће светске компаније које су се усредредиле на његов развој и примену.

Поред бенефита у софтверској индустрији, Cloud computing увео је многе промене које су утицале и на бизнис. Капиталне инвестиције у локални хардвер сведене су на минимум и замењене су оперативним улагањима у Cloud computing. Такође, он је увео и нов начин плаћања по принципу: “плати колико потрошиш”, што је резултирало драстичним смањењем трошкова.

**НАПОМЕНА:**

Овај рад проистекао је из мастер рада чији ментор је био др Себастијан Стоја.

Један од могућих начина имплементације Cloud computing окружења представља микросервисни архитектурални приступ. Филозофија микросервисног архитектуралног стила суштински се изједначава са Unix филозофијом: “Ради једну ствар и ради је добро”. Историја и порекло микросервиса подразумева сталне напоре да се обезбеди боља комуникација између различитих платформи, једноставнијих система и система једноставнији за корисника.

Главни задатак овог рада јесу детаљан опис, анализа и приказ резултата времена извршавања тренинга модела машинског учења и предвиђања потрошње електричне енергије. Приказани резултати времена извршавања тестирани су на различитим Cloud computing платформама које се заснивају на PaaS моделу Cloud computing-a. У раду се директно пореде AWS Lambda и Azure Functions. Поређење укључује резултате извршавања, анализу резултата, приказ могућности, као и предности и мане оба решења.

**2. ТЕОРИЈСКЕ ОСНОВЕ****2.1 Cloud Computing**

Cloud computing [2] је све заступљенији приступ пружања рачунарских услуга на флексибилан, ефикасан и лако доступан начин. Представља суштинску промену у начину на који компаније и појединци предају одговорност компанијама које пружају cloud услуге за обезбеђивање IT инфраструктуре и услуга. Традиционални IT outsourcing аранжмани обухватају уговоре са јасно дефинисаним објектима за складиштење и обраду података. За разлику од класичне инфраструктуре, количина IT ресурса може да се мења током времена, често брзо и динамично као одговор на тренутну потражњу за IT ресурсима. Купац ресурса углавном није свестан где се налази услужна инфраструктура.

Cloud-base пословни модел обухвата следеће:

- Смањење трошкова: дељењем ресурса између групе купаца и куповином инфраструктуре на велико, гигантске мултинационалне компаније које пружају услуге Cloud computing-a могу постићи уштеде које ће се пренети на купце ресурса,
- Трансформисање CAPEX у OPEX: премештање пословних операција у Cloud омогућава смањење трошкова за интерну IT инфраструктуру,
- Краће време до пословне примене: могућност брзог добављања нових капацитета скраћује време од развоја до пословне примене,

- Заштита животне средине: иако велики центри података изгледају као интезивни потрошачи енергије, у пракси се показало да је управљање једним таквим центром података енергетски ефикасније од еквивалентног капацитета састављеног од појединачних рачунара и сервера.

Амерички аналитичар Гартнер дефинише *Cloud computing* као: “*Стил рачунарства где су скалабилне и еластичне ИТ могућности обезбеђене као услуга за више корисника који користе интернет технологије.*”

Иако ова дефиниција није савршена, она обухвата кључне атрибуте које чине *Cloud computing* привлачним купцима. Под скалабилним подразумева се количина коришћених рачунарских ресурса, било да се ради о обради или складиштењу података. Под појмом еластична, Гартнер подразумева да се скалирање може обавити брзо као одговор на промене у потражњи за ресурсима.

Једну од главних предности *Cloud computing* представља дељење скупа рачунарских ресурса између клијената. Клијенти обично имају променљиве захтеве за ресурсима, који ретко достижу максимум у исто време, тако да провајдери *Cloud* услуга обезбеђују базно оптерећење свих клијената које је мање од максималног оптерећења свих клијената.

*Cloud computing* је веома широк појам који се користи за описивање различитих аспеката у рачунарству. Модели *cloud computing*-а се деле на различите нивое услуга према степену виртуелизације [1], а то су *Platform as a service* (PaaS), *Infrastructure as a Service* (IaaS) и *Software as a Service* (SaaS).

## 2.2 Infrastructure as a Service

*IaaS* [3] је облик *hosting*-а, који укључује приступ мрежи, услуге рутирања, и складиштење. *IaaS* провајдер ће генерално обезбедити *hardware*, административне услуге потребне за чување апликација и платформу за покретање апликација. Добављач услуга поседује опрему и задужен је за њено складиштење, вођење и одржавање. *Infrastructure as a service* је еволуција традиционалног *hosting*-а који не захтева никакву дугорочну посвећеност и омогућава корисницима да обезбеде ресурсе на захтев. За разлику од *PaaS*, *IaaS* провајдери су задужени за управљање само у случају одржавања оперативног *Data* центра. Корисници примењују софтверске услуге и управљају њима баш онако како би то радили у својим *Data* центрима.

## 2.3 Platform as a Service

*PaaS* [4] је платформа која се гради на врху *IaaS* слоја, и обезбеђује комплетну инфраструктуру потребну за рад апликације преко интернета. Корисници *PaaS* не размишљају о позадинским процесима и сложености позадинског система. Она пружа платформу за креирање софтвера и омогућава програмерима да праве прилагођене апликације на мрежи без потребе да се баве сервирањем, складиштењем и управљањем података. У суштини, *PaaS* модел услуга омогућава предузећима да се концентришу на сам софтвер без промене постојеће инфраструктуре. Као и услужни програми, *PaaS* модел и даље пружа клијентима сервере и центре података у којима могу да похрањују

своје информације, али у случају *PaaS* развијају једну апликацију која ће потом бити испоручена преко интернета. *PaaS* омогућава лак прелазак на хибридни модел *Cloud*-а, који представља комбинацију различитих окружења, а такође обезбеђује и приступ различитим ресурсима у групи апликација, укључујући програмске језике, оперативне системе и базе података. *PaaS* је заснован на моделу мерења или претплате, тако да корисници плаћају оно што потроше (енг. *Pay as you go model*).

## 2.4 Software as a Service

*SaaS* [1] је најчешће коришћен од свих слојева на *Cloud*-у гради се на врху *IaaS* и *PaaS* слојева. *SaaS* представља алтернативу локално покренутим апликацијама и стога је најпривлачнији крајњим корисницима. Потпуно је скалабилан и не мора да се преузима или инсталира на појединачним уређајима да би се применио на читав тим или компанију. Ова функција је посебно корисна за дистрибуиране глобалне тимове и хибридна радна окружења.

## 2.5 Микросервисна архитектура

Микросервисна архитектура [5] је архитектура сервисно-оријентисаног архитектуалног стила који организује апликацију у колекцију слабо повезаних услуга. Састоји се од колекције малих, аутономних услуга. Свака услуга је самостална и треба да имплементира једну пословну способност у оквиру система. Свака услуга је засебна, којом може да управља мали развојни тим. Сервиси се развијају независно, омогућавајући на тај начин унапређење одеђеног сервиса без утицаја на друге делове система. Услуге комуницирају преко јасно дефинисаних *API*-ја скривајући на тај начин детаље интерну имплементацију других услуга.

## 3. ARHITEKTURA

Уводни део рада је послужио за представљање *Cloud computing*-а, микросервисне архитектуре, као и коришћених технологија. У овом поглављу биће анализирана архитектура имплементираних решења. Посебно ће бити објашњена свака компонента *Azure functions* и *AWS Lambda* система како би се што боље стекао увид у улогу сваке компоненте.

Апликација је намењена за тренирање модела машинског учења употребом *ML.NET*-а, предвиђање потрошње електричне енергије, као и приказ резултата предвиђања. Улазни подаци тренинга представљају временске сатне податке и сатне податке потрошње електричне енергије у периоду од 2016. до 2019. године. У оквиру једног микросервиса имплементиран је алгоритам машинског учења врши корелацију временских услова и потрошње електричне енергије за наведени период. Такође, имплементиран је алгоритам за филтрирање и парсирање података, као последњи корак подаци се деле на независне (временски) и зависне (потрошња електричне енергије) прослеђују алгоритму машинског учења. Резултат тренинга је модел машинског учења, пошто се ради о неструктурираним подацима, неопходно је њихово складиштење на *S3 Bucket* или *Azure Blob*. Предвиђање резултата потрошње уз коришћење сачуваних и истренираних

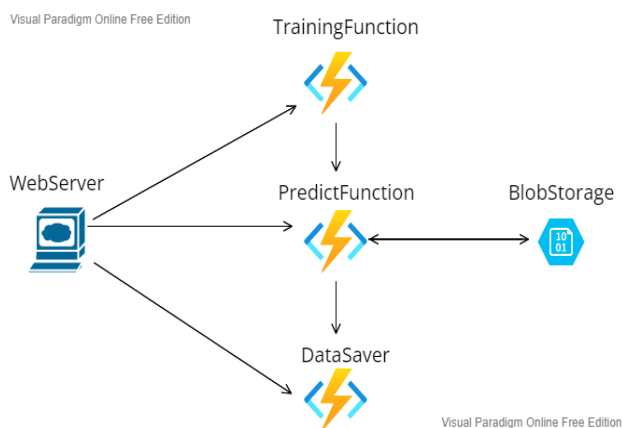
модела представља посебан микросервис. Приликом предвиђања у оквиру прослеђеног документа садржани су само независни подаци, а зависни подаци представљају резултат предвиђања на основу независних података.

Чување резултата предвиђања имплементирано је као посебан микросервис.

У оквиру микросервиса локално се чувају резултати последњег предвиђања. Клијент приликом тренинга и предвиђања податке шаље путем *HTTP* захтева у (*.xlsx*) документу.

### 3.1. Архитектура решења Azure functions

Имплементирано решење заснива се на микросервисној архитектури коришћењем *Azure functions* сервиса и приказана је на слици 1.



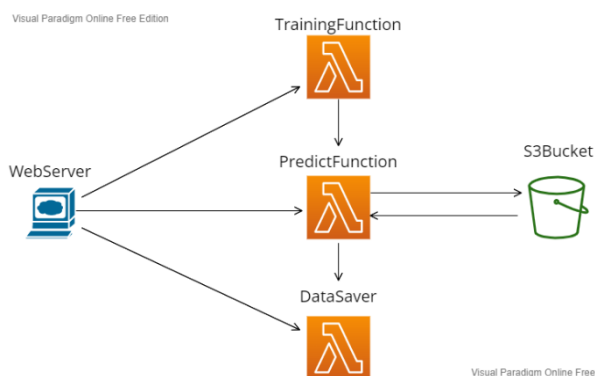
Слика 1. Архитектура Azure решења.

Компоненте система су следеће:

- *Web Server* – *On-premise* компонента која представља везу између *Azure functions* и корисника. Омогућава графички интерфејс за крајњег корисника као и комуникацију између корисника и одговарајућих функција система,
- *Training Function* – Прослеђени документ од стране корисника парсира и филтрира, затим са тим подацима врши тренинг података. Добијени модел машинског учења прослеђује функцији *Predict Function*,
- *Prediction Function* - Добијен модел чува локално и на *Blob Storage*-у. Приликом предвиђања парсира и филтрира улазни документ затим уз помоћ сачуваног модела машинског учења врши предикцију података. Добијене податке прослеђује функцији *Data Saver*,
- *Data Saver* – Добијене податке чува локално. На захтев клијента прослеђује податке на *Web Server*, и
- *Blob Storage* – Чува последње истренирани модел у *.Zip* формату.

### 3.2 Архитектура решења AWS Lambda

Имплементирано решење заснива се на микросервисној архитектури коришћењем *AWS Lambda* сервиса и приказана је на слици 2.



Слика 2. Архитектура AWS решења.

Компоненте система су следеће:

- *Web Server* – *On-premise* компонента која представља везу између *AWS Lambda* и корисника. Омогућава графички интерфејс за крајњег корисника као и комуникацију између корисника и одговарајућих функција система,
- *Training Function* – Прослеђени документ од стране корисника парсира и филтрира, затим са тим подацима врши тренинг података. Добијени модел машинског учења прослеђује функцији *Predict Function*,
- *Prediction Function* - Добијен модел чува локално и на *S3*. Приликом предвиђања парсира и филтрира улазни документ затим уз помоћ сачуваног модела машинског учења врши предикцију података. Добијене податке прослеђује функцији *Data Saver*,
- *Data Saver* – Добијене податке чува локално. На захтев клијента прослеђује податке на *Web Server*, и
- *S3 Bucket* - Чува последње истренирани модел у *.Zip* формату.

## 4. РЕЗУЛТАТИ ТЕСТИРАЊА

У овом поглављу биће анализирани резултати времена извршавања *Azure functions* и *AWS Lambda*. Да би добијени резултати били што релевантнији апликације су биле подигнуте у оквиру својих *Cloud* окружења, улазни подаци приликом тестирања увек су били исти, као и број итерација приликом тренинга.

**Потребан услов сваког теста:** Софтверско решење за тренирање и превиђање и тренутно чување резултата је подигнуто на *Cloud*, покренуто и подешено за рад. У овом раду су извршена три тест сценарија:

#### Сценарио тест (1):

- Покретање одговарајућег веб сервера,
- Одабир тренинга на графичком интерфејсу,
- Одабир одговарајућег документа за тренинг, и
- Иницирање функције *Training*.

#### Сценарио тест (2):

- Покретање одговарајућег веб сервера,

- Одабир предвиђања на графичком интерфејсу,
- Одабир одговарајућег документа за предвиђање података, и
- Иницирање функције *Predict*.

### Сценарио тест (3):

- Покретање одговарајућег веб сервера,
- Одабир функције *Data Saver* и иницирање исте, и
- Приказ података предвиђања.

Табела 1. Поређење времена извршавања функција у зависности од Cloud платформе.

Тестирана функција	Просечно време извршавања Azure	Просечно Време извршавања AWS Lambda
Сценарио тест (1), време извршавања функције[s]	38.4999	75.2485
Сценарио тест (1), HTTP Захтев и одговор [s]	4.3318	3.4144
Сценарио тест (2), време извршавања функције[s]	2.0031	2.5968
Сценарио тест (2), HTTP Захтев и одговор[s]	0.6934	0.617
Сценарио тест(3), време извршавања функције[s]	0.0000633	0.0002166
Сценарио тест(3), HTTP Захтев и одговор[s]	0.66176	0.1959

Мерењима из табеле 1 показана је разлика у брзини извршавања функција у зависности од платформе. Треба приметити да приликом извршавања процесорски захтевних функција као што је тренирање модела машинског учења *Azure functions* остварују 48.99% боље време извршавања функције. Приликом преноса веће количине података, као што су подаци за тренинг *AWS Lambda* је у просеку за 21.2% остварио бољи резултат. Мање процесорски захтевно предвиђање резултата доводи до смањења разлике у времену извршавања, па приликом предвиђања *Azure functions* остварује 22.86% бољи резултат. Слабе мање количине података у захтеву приликом предвиђања резултата одржава се на мању разлику између *AWS Lambda* и *Azure Functions* у реализацији захтева и одговара, у случају предвиђања *AWS Lambda* је просечно 0.0765 секунди брже реализовала захтев и одговор у односу на *Azure Functions* што представља просечно 11% мање времена за реализацију захтева и одговара. Додављање података представља процесорски

најмање захтевну операцију, иако се ради о делићима секунде *Azure Functions* је приликом извршавања функције остварио 70% боље просечно време извршавања. Треба приметити да *AWS Lambda* треба просечно 0.4648 секунди мање за реализацију захтева, извршавање функције и одговор што представља 70.35 % мање времена потребног за извршавање захтева у односу на *Azure Functions*. Иако је време укупно време извршавања краће треба узети у обзир више времена потребног за извршење функције што може довести до већих трошкова у односу на *Azure Functions*.

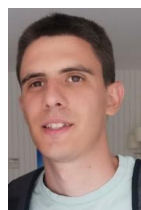
## 5. ЗАКЉУЧАК

Модерна софтверска решења све чешће користе микросервисни архитектурални стил у оквиру *Cloud computing* платформе. За разлику од монолитне архитектуре која се састоји од само једног сервиса, микросервисна архитектура представља колекцију малих независних сервиса. Рад детаљно описује и анализира резултате два микросервисна софтверска решења у оквиру две различите платформе: *PaaS* модела и *Cloud computing*-а. Анализиране су технологије и стандарди коришћени приликом израде одговарајућих софтверских решења и предочене су предности коришћења *Cloud computing* микросервисне архитектуре.

## 6. ЛИТЕРАТУРА

- [1] S. Nagaprasad, A. VinayaBabu, K. Madhukar, D. Marlene G Verghese, V. Mallaiah and A. Sreelatha, "Reviewing some platforms in cloud computing", *International Journal of Engineering and Technology*, vol. 2, no.5, pp. 348-353, 2010.
- [2] S. Bradshaw, C. Millard, and I. Walden, "Contracts for clouds: Comparison and analysis of the terms and conditions of cloud computing services", *International Journal of Law and Information Technology*, vol. 19, no. 3, pp. 187-223, 2011.
- [3] S. Bhardwaj, L. Jain, and S. Jain, "Cloud computing: A study of infrastructure as a service (IAAS)" *International Journal of engineering and information Technology*, vol. 2, no. 1, pp. 60-63, 2010.
- [4] K. Gurudatt, P. Khatawkar, and J. Gambhir. "Cloud computing-platform as service", *International Journal of Engineering*, vol. 1, 2011.
- [5] EdPrice-MSFT, "Microservice architecture style - azure architecture center," Azure Architecture Center | Microsoft Learn. [Online]. Available: <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>. [Accessed: 24-Oct-2022].

## Kratka biografija:



Душан Носовић рођен је 1996. године у Шапцу. Завршио је Шабачку гимназију 2015. године. Факултет техничких наука у Новом Саду уписао је 2015. године, смер Примењено софтверско инжењерство. Завршио је основне студије 2021. године и након уписао мастер студије на Факултету техничких наука у Новом Саду, смер Примењено софтверско инжењерство.