

СОФТВЕРСКО РЕШЕЊЕ ЗА ОБРАДУ ПОДАТАКА О БЕРЗАМА ЗАСНОВАНО НА АРХИТЕКТУРИ SERVERLESS**A SOFTWARE SOLUTION FOR STOCK MARKET DATA PROCESSING BASED ON THE SERVERLESS ARCHITECTURE**

Вук Милановић, Факултет техничких наука, Нови Сад

Област – Електротехничко и рачунарско инжењерство

Кратак садржај – Данас, све више људи жели да улаже на берзи, али се често сусрећу са изазовима у избору и праћењу инвестиција. Постоји потреба за алатом који ће бити прилагодљив, бесплатан и који ће обухватити све основне функционалности за улагање и праћење инвестиција. Циљ овог мастер рада је да се корисницима омогући правовремено праћење промена цена акција, као и да им се пружи увид у пословање компанија од интереса. Стога, имплементирано је софтверско решење засновано на архитектури *serverless*, које омогућава преузимање, складиштење и визуелни приказ берзанских података. Основне функционалности система су приказ кретања цена акција компанија у реалном времену, комплетан преглед више врста финансијских извештаја, као и системско обавештавање крајњих корисника, услед наглих промена цена акција на тржишту.

Кључне речи: Рачунарство у облаку, архитектура *serverless*, обрада података у реалном времену, пакетна обрада података

Abstract – Today, an increasing number of people wish to invest in the stock market, but often encounter challenges in selecting and monitoring their investments. There is a need for a tool that is adaptable, free, and covers all the essential functionalities for investing and tracking investments. The goal of this master thesis is to enable users to timely monitor stock price changes and gain insights into the operations of companies of interest. Therefore, a software solution based on the *serverless* architecture has been implemented, allowing for the retrieval, storage, and visual display of the stock market data. The core functionalities of the system include real-time tracking of stock price movements and a comprehensive overview of various types of financial reports. Additionally, to enhance the user experience, a notification system has been implemented to alert end users of sudden stock price changes.

Keywords: Cloud computing, *serverless* architecture, real time processing, batch processing

НАПОМЕНА:

Овај рад је проистекао из мастер рада чији ментор је био др Владимир Димитриески, ванредни професор.

1. УВОД

Историја берзе [1] датира још од почетка деветнаестог века, када је по лондонским кафетеријама почела да кружи трговина акција Британске источноиндијске компаније (енгл. *British East India Company*). Управо овај период обележио је настанак прве светске берзе у Лондону 1801. године. Крајем деветнаестог века, појавом телеграфа и телефона, комуникација се побољшала, омогућавајући трговину на већој удаљености и повезујући различита тржишта. Другу половину двадесетог века обележила је појава прве електронске берзе *NASDAQ*, што је довело до убрзања и поједностављења трговине акцијама уз помоћ рачунара. Електронске платформе су омогућиле шире учешће инвеститора и бољу анализу тржишних података, што је утицало на инвестиционе стратегије. У савременом периоду, све више људи жели да инвестира на берзи, али се често суочавају са потешкоћама у избору и праћењу сопствених инвестиција. Постоји потреба за алатом који ће бити прилагодљив, бесплатан и који ће обухвата све основне функционалности за управљање и праћење инвестиција. Циљ овог рада је да се крајњи корисници на правовремен начин информишу о променама цена акција компанија од интереса и да им се омогући адекватан увид у њихово пословање. Како би се то постигло имплементирано је савремено софтверско решење засновано на архитектури *serverless* [2], коришћењем технологија рачунарства у облаку, које омогућава преузимање, складиштење и визуелни приказ берзанских података.

2. АНАЛИЗА ПОСТОЈЕЋИХ СИСТЕМА

Кроз историју развијан је велики број система за обраду података, с различитим архитектурама и технологијама, у циљу добијања правовремених и тачних података. У наставку поглавља биће наведени описи неколико система сличних софтверском решењу које прати овај рад, као и њихови недостаци.

2.1. Блумберг терминал

Блумберг терминал је сложен систем из области финансија који нуди приступ подацима у реалном времену, као и анализу истих. Укључује историјске податке о акцијама, валутама, дериватима, а прва верзија система објављена је осамдесетих година прошлог века. Пружа напредне аналитичке алате и могућност комуникације између клијената. Иако се доказао као поуздан и ефикасан алат, Блумберг

терминал представља једно релативно старије (енгл. *legacy*) решење. С тим у вези, до пре неколико година, систем се ослањао искључиво на програмске језике Ц и Фортран, док није отпочела транзиција ка новијим језицима, као што су Јаваскрипт и Ц++. Такође, одржавање овог комплексног система било је тешко због одсуства технологија рачунарства у облаку, што је отежавало скалирање и управљање хардверским ресурсима и захтевало велике инвестиције у инфраструктуру и људске ресурсе.

2.2. Google finance

Компанија Google [3] је 2006. године представила бесплатну платформу која корисницима пружа разноврсне финансијске информације, укључујући вести и податке о тржишним трендовима. Корисници могу у реалном времену пратити цене акција светских компанија и анализирати различите временске периоде, уз приступ историјским, годишњим и кварталним извештајима. Ова платформа је послужила као мотивација за развој софтверског решења разматраног у овом раду, с намером да се постигне одређена предност у односу на постојеће решење. С тим у вези, Google користи свој апликативни програмски интерфејс (енгл. *API*), код ког се могу испољити одређена ограничења при изненадном повећању броја корисничких захтева, што може довести до кашњења и проблема са ажурирањем података. Поред тога, визуелизација података на Google-овој платформи је мање флексибилна, јер сви корисници добијају идентичну командну таблу. Насупрот томе, софтвер описан у овом раду омогућава корисницима да сами креирају или добију командну таблу по жељи, прилагођену њиховим потребама.

2.3. Yahoo finance

Yahoo finance [4] је платформа коју је компанија Yahoo лансирала 1997. године, као алат намењен за праћење и анализу финансијских података. Истиче се по својој интеграцији са широким спектром алата из области анализе финансијских тржишта (*TradingView*, *MetaTrader* и сл.). Исто тако, платформа интерно нуди вести, анализе и напредне алате, укључујући графичке приказе техничких индикатора и историјских података. У поређењу са Google finance-ом, Yahoo има више интерактивних графикана и опција за прилагођавање приказа. Такође, корисницима је омогућена пријава за различите врсте обавештења, али овај опција није бесплатна, као и већина претходно споменутих. Међутим, слична функционалност је предвиђена да буде део апликације коју прати овај рад, где би корисници били у стању да специфицирају обавештења за њима релевантне компаније, без било какве претплате.

3. АРХИТЕКТУРА СИСТЕМА

Систем је заснован на архитектури *serverless*, коришћењем Амазонових веб сервиса [5]. Одабиром оваквог приступа омогућено је динамичко скалирање, оптимизација ресурса и аутоматизација задатака система. Поред тога, систем има подразумеване карактеристике високе доступности и отпорности, које гарантује одабрани пружалац услуга у облаку.

Архитектура система је подељена у три целине тј. модула, што се може видети на слици 3.1, и ти модули ће у наставку поглавља бити описани у појединачним одељцима. Пре него што се приступи детаљном опису сваког модула појединачно, важно је споменути да сва озбиљнија софтверска решења, посебно она која складиште и анализирају велике количине података, садрже критичне компоненте које захтевају појачану безбедност и контролисану заштиту. Због тога је овај систем постављен у виртуелну приватну мрежу (енгл. *Virtual Private Cloud - VPC*), са конфигурисаним јавним и приватним подмрежама (енгл. *subnets*). Ресурси осетљиве природе, попут објектних складишта и база података, смештени су у приватну мрежу и доступни су само ресурсима који су део овог система. С друге стране, ресурси из јавне мреже, попут функција *Lambda* које преузимају податке са изворног *API*-ја, могу комуницирати са спољним светом.

3.1. Модул увлачења података у систем

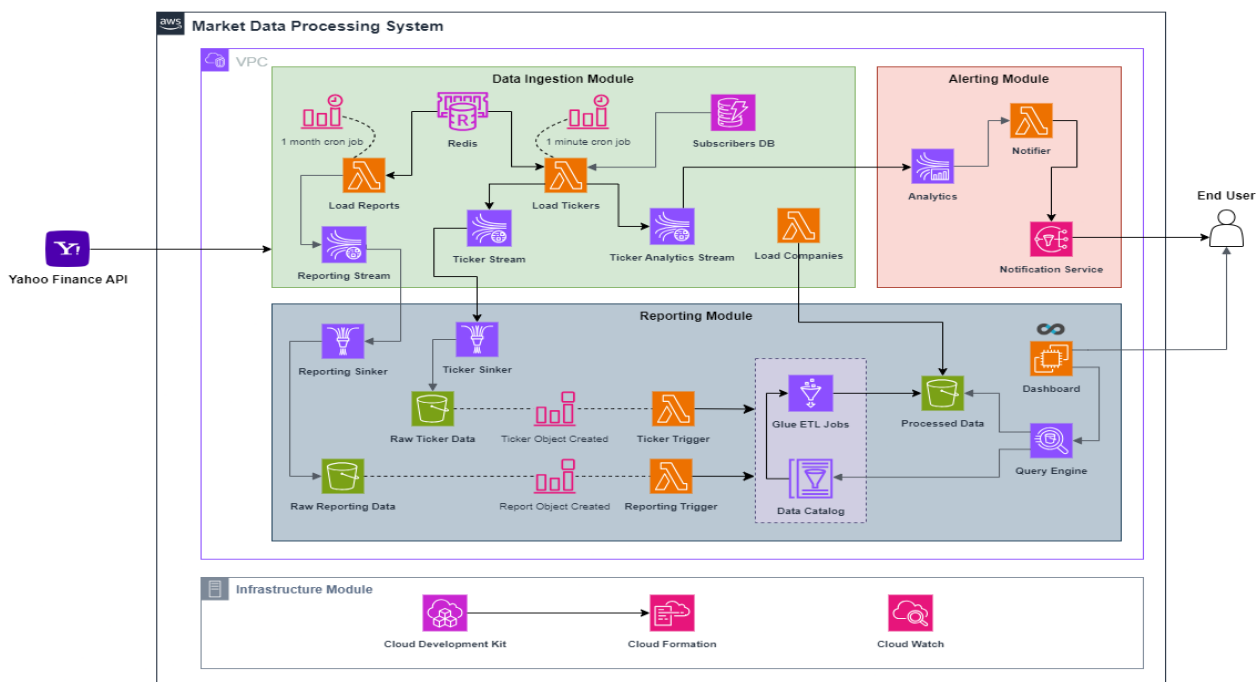
Модул увлачења, односно *Data Ingestion Module* на слици 3.1, представља срж овог софтверског решења, без којег систем не би могао да функционише. У систем се увлаче различите врсте података уз помоћ функција *Lambda*, где свака укључује скрипту написану у Пајтон програмском језику, која реализује имплементирану логику. У наставку ове секције биће описана намена скрипти за увлачење података:

1. *Load Reports* - скрипта се покреће једном месечно од стране сервиса *AWS Event Bridge* и проверава нове податке о годишњим финансијским извештајима. Разлог одабраног периода покретања лежи у томе што се финансијска година разликује од компаније до компаније.
2. *Load Tickers* - идемпотентна функција *Lambda* која се покреће сваки минут и инкрементално добавља податке о ценама акција свих праћених компанија.
3. *Load Companies* - скрипта која се користи за иницијално попуњавање или ажурирање табеле са информацијама о компанијама које систем прати.

Архитектура решења укључује две базе података с начином чувања података кључ-вредност: *Amazon ElastiCache* и *Amazon DynamoDB*. *Amazon ElastiCache* чува време последњег захтева ка *API*-ју како би систем избегао поновну обраду истих података, док *Amazon DynamoDB* чува трајне податке о корисницима. Ипак, разлог коришћења две сличне базе података лежи у пар кључних аргумената, а то су:

1. Трајност података - *Amazon DynamoDB* гарантује трајно чување података о корисницима, док *Amazon ElastiCache* не нуди трајну перзистенцију.
2. Безбедност података - податке о корисницима треба шифровати, док временски записи о увлачењу података, који су јавни, не захтевају шифровање.

Иако се на први поглед чини да је *DynamoDB* адекватно решење, због ниске латенције и напредних могућности кеширања, *ElastiCache* је боља опција за чување временских записа. Након добављања, подаци се прослеђују на даљу обраду и анализу помоћу токова



Слика 3.1 – Архитектура система

података. За потребе нашег система изабран је сервис *Amazon Kinesis* на три различита места у овом модулу:

1. *Reporting Stream* - прослеђује податке о финансијским извештајима у део система за обраду података у циљу извештавања.
2. *Ticker Stream* - прослеђује податке о тикерима у део система за обраду и приказ података крајњим корисницима.
3. *Ticker Analytics Stream* - прослеђује податке у нешто измењеном облику, у део система за потенцијално обавештавање корисника о наглим променама цена акција.

Услед могућег повећања оптерећења и времена извршавања функција *Lambda*, посебно мислећи на ону са дневним увлачењем цена акција компанија, очекује се да ће број корисника и обим података временом расти. Стога, постоје пар опција у виду будуће замене претходно споменутог сервиса, а то су *Amazon Elastic Container Service (ECS)* и *Amazon Elastic Cloud Compute (EC2)*. Генерално гледано, функције *Lambda* нису намењене за већа оптерећења, што оправдава потенцијалну надоградњу. Исто тако, када је реч о чувању података о корисницима, алтернатива за *DynamoDB* може бити било која релациона база података сервиса *Amazon RDS*.

3.2. Модул извештавања

Модул извештавања, или *Reporting Module* на слици 3.1, је најсложенији део система. Сирови подаци из токова података уливају се у сервисе *Amazon Data Firehose* који их прикупљају, трансформишу и прослеђују ка објектним складиштима. У овом модулу постоје два сервиса *Data Firehose*:

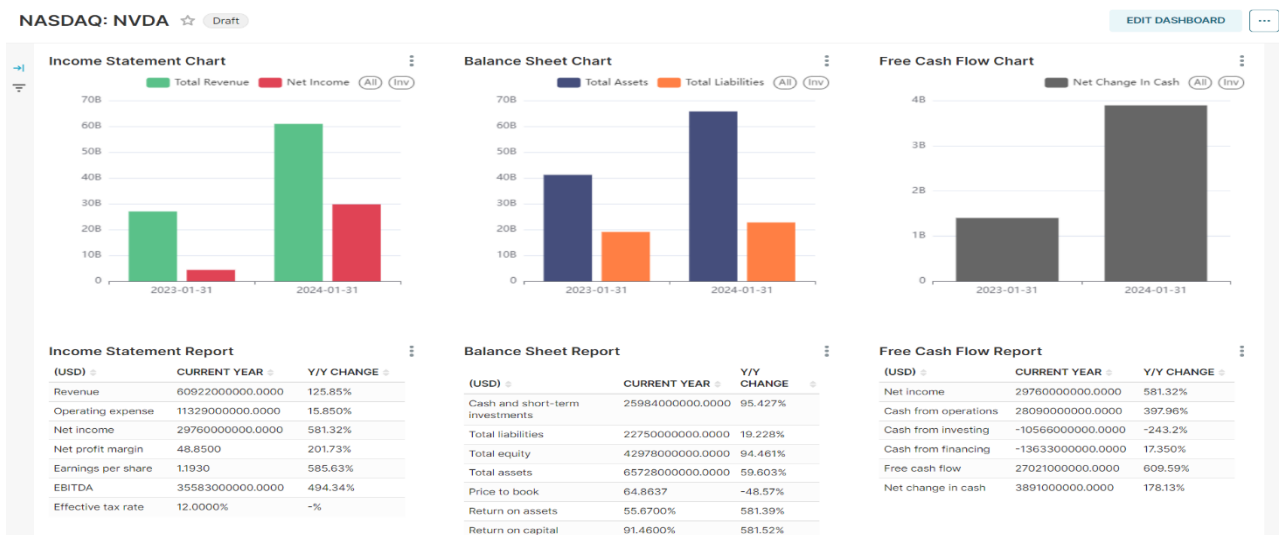
1. *Reporting Sink* - за податке о финансијским извештајима.
2. *Ticker Sink* - за податке о тикерима.

Крајње дестинације података из сервиса *Data Firehose* су два објектна складишта: *Amazon Simple Storage Service (S3)*. Објектна складишта су

популаран начин за чување великих количина података разних формата због своје скалабилности и ниских трошкова. У нашем решењу, сирови подаци типа *JSON* се континуално складиште у језеру података (енгл. *data lake*), користећи *Yahoo Finance API* као једини извор. Сервиси који покрећу обраду сирових података су функције *Lambda*, које реагују при сваком доласку новог објекта у објектно складиште. Обработом података управља сервис *AWS Glue* у комбинацији са *Data Catalog*-ом. Процес обраде је имплементиран помоћу Пайтон скрипти и радног оквира *Apache Spark*, који је познат по брзој и паралелној обради података. Након обраде, подаци се смештају у *S3* сервис *Processed data*, који представља складиште података (енгл. *data warehouse*) система. *AWS Athena* је изабрана као *query engine* због своје ефикасности и подршке за разне формате података. У будућности, за складиштење већих количина података, могло би се размислити о интеграцији *Amazon Redshift*-а, али за тренутне потребе би био превелик трошак. Након обраде, потребно је испроцесирани податке графички приказати. Сервис *Amazon QuickSight* је квалитетан, али и скуп као *Redshift*. Стога, алтернативу представља *Apache Superset*, који захтева посебан начин интегрисања у систем, с обзиром да није експлицитан Амазонов сервис. Између *Elastic Container Service* и *Elastic Compute Cloud*, изабрана је друга опција због једноставнијег постављања и подешавања *Superset*-а.

3.3. Модул упозоравања крајњих корисника

Модул упозоравања, односно *Alerting Module* на слици 3.1, део је нашег система који омогућава брзо обавештавање крајњих корисника о изненадним променама цена акција компанија које их интересују. Први корак је преузимање података из модула за



Слика 4.1 – Део контролне табле с годишњим финансијским извештајима

увлачење, који се затим анализирају у сервису *Amazon Kinesis Analytic*. Овај сервис, користећи *SQL* скрипту, пропушта само податке који испуњавају задати критеријум. Уколико дође до изненадног пораста или пада неке од цена акција компанија, која излази из кориснички дефинисаног опсега, подаци се прослеђују *Lambda* функцији *Notifier*. Уз посредство сервиса за слање нотификација (енгл. *Amazon Simple Notification Service*), циљном кориснику се аутоматски шаље обавештење у виду *SMS* поруке.

3.4. Инфраструктурни модул

Инфраструктурни модул на дну слике 3.1, састоји се из неколико компоненти. Системска инфраструктура је дефинисана у Пајтон програмском језику уз помоћ *AWS Cloud Development Kit*. Ова библиотека омогућава корисницима да одаберу и конфигуришу већину сервиса, након чега се инфраструктура испоручује у окружење у облаку путем *AWS Cloud Formation*, припремајући систем за рад. Осим тога, важно је нагласити улогу *Amazon Cloud Watch*-а, који омогућава надгледање (енгл. *monitoring*) кључних метрика сервиса и записивање активности (енгл. *logging*) функција *Lambda*. Ово омогућава правовремену реакцију на уска грла и унапређује процес дијагностике, доприносећи бржем решавању проблема у продукцији.

4. ЗАКЉУЧАК

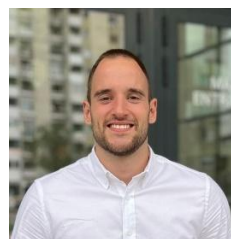
Примарни циљ овог рада био је да се корисници система, на правовремен начини, информишу о ценама акција компанија, као и стању пословања истих. Развијен је софтверски пакет за инвеститоре и аналитичаре који омогућава праћење цена акција у реалном времену, комплетан преглед више врста финансијских извештаја, као и системско слање обавештења о наглим променама на тржишту. Систем се истиче као потпуно бесплатна услуга за праћење инвестиција, ослањајући се на Амазонове сервисе и интеграцију са *Apache Superset*-ом. Корисници могу да прилагоде командну таблу са слике 4.1, према својим потребама, што представља додатну предност у односу на друге системе. Инфраструктура је постављена коришћењем *AWS Cloud Development Kit*-а у Пајтону,

уз разматрање алтернатива за будуће скалирање. Иако је ово функционално софтверско решење, постоје аспекти система који се могу додатно побољшати. Пре свега, кориснички интерфејс се може проширити додавањем функционалности за регистрацију корисника и измену личних података у складу са законом о заштити података. Такође, формати складиштених података могу бити унапређени коришћењем *Parquet* формата уместо *JSON*-а, што би побољшало перформансе при аналитичким упитима. Поред тога, било би корисно омогућити слање различитих врста обавештења, укључујући електронску пошту, као и унапређење увлачења података у различитим форматима ради флексибилнијег рада са различитим изворима података.

5. ЛИТЕРАТУРА

- [1] StreetStocks. Stock Market Technology: – StockMarketLore ---streetstocks.medium.com.
<https://streetstocks.medium.com/stock-market-technology-tracing-its-historical-evolution> stockmarketlore-a2d7bf60708e.
- [2] Dr Rajan. Serverless architecture - a revolution in cloud computing. pages 88--93, 12 2018.
- [3] Google Finance - Wikipedia --- en.wikipedia.org.
https://en.wikipedia.org/wiki/Google_Finance.
- [4] Yahoo Finance - Wikipedia --- en.wikipedia.org.
https://en.wikipedia.org/wiki/Yahoo_Finance.
- [5] Frederic P. Miller, Agnes F. Vandome, and John McBrewster. Amazon Web Services. Alpha Press, 2010.

Кратка биографија:



Вук Милановић рођен је у 15. септембра 2000. године у Новом Саду. Завршио је гимназију „Исидора Секулић“ 2019. године након чега уписује Факултет техничких наука, смер Рачунарство и аутоматика. Дипломски рад је одбранио 2023. године када уписује мастер студије на студијском програму Рачунарство и аутоматика.