Cloud Based Architecture of the Core Banking System

Andrii Roskladka[1[0000-0002-1297-377X]], Nataliia Roskladka[1[0000-0001-7333-4050]], Ganna Kharlamova[2[0000-0003-3614-712X]] and Roman Baglai[1[0000-0002-4067-4929]]

[1] Kyiv National University of Trade and Economics,
Kyoto st. 19, 02156, Kyiv, Ukraine
a.roskladka@knute.edu.ua, n.roskladka@knute.edu.ua,
romanbaglai@gmail.com
[2] Taras Shevchenko National University of Kyiv,
Vasylkivska st. 90, 03022, Kyiv, Ukraine
akharlamova@ukr.net

Abstract. The article contains a study of cloud technology and standards applicable to Core Banking System (CBS). National regulators often require storage of the data on physical servers of the country where the bank is registered. This is probably due to the lack of awareness of cloud technology data protection capabilities on the regulator side. Although main cloud service providers comply with international security standards, such as Payment Card Industry Data Security Standard (PCI DSS), International Organization for Standardization (ISO 9001:2015, ISO/IEC 27001:2013, ISO/IEC 27017:2015) and many other national security standards [3]. This means they offer much higher degree of information security that the bank can afford within own infrastructure. Modern mathematical systems and methods have been used to define an optimal configuration of cloud based platform for CBS. An analytical model was built based on the EC2 memory optimized class instances configuration.

Keywords: core banking system, OLTP, OLAP, Postgresql, Amazon elastic cloud compute

1 Introduction

Cloud technology today is enabling innovation and digital transformation of banking industry. Lagging behind FinTech startups large system banks start to apply cloud computing in various spheres of their activity.

Digital public channels for customers like Android and IOs applications, corporate web site etc. is usually the first step in this direction. Second step in migrating to the cloud is high performance computing for public data analytics as for instance risk scoring models. Third step is typically IaaS solutions to ensure business continuity and moving the secondary data center to the cloud.

CBS is usually one of the last steps in cloud migration as it is heavier than two steps mentioned earlier and involves risk related to customer data confidentiality, integrity and availability. Main risks attached are related to moving customer data to the cloud.

European General Data Protection Regulation (GDPR) requirements specify that customer data should not be stored permanently without legitimate reason of it processing, storing customer data in the cloud require either customer consent or unonimization. Banks are sitting on Big Data scale infrastructure which has a tendency to grow exponentially. Those volumes demand ever increasing computing capacity for processing.

Migration CBS to the cloud is the subject of this research. The aim of the article is to research the architecture and configuration of services to enable the cloud deployment of CBS. The research is based on the data protection requirements mentioned above including architecture schemes, data flow UML diagrams and mathematical modeling.

2 Research methodology

The theoretical and methodological background is the fundamental principles of the systematic approach, the methods of scientific abstraction, analysis and synthesis, induction and deduction, the dialectical method of knowledge of information technology.

In particular the following scientific methods are used in research:

- graphical method – Unified Modeling Language (UML) and the Open Group Architecture Framework (TOGAF) were used by the authors to visualize the design of the system;
- classification method – Common Attack Pattern Enumeration and Classification (CAPEC) , Open Web Application Security Project (OWASP) when determining the priority directions of application of different IT security mechanisms for specific models of cloud technologies deployment;
- quantitative analysis method – multicriteria optimization in Mathcad to minimize the time and cost of using the cloud service.

The information basis for the study is data of Amazon web services, Microsoft, Temenos and other technical documentation available publically.

3 Literature and hypothesis development

The problems of using cloud technologies in various socio-economic spheres are studied in [12]. Among the main directions of solving the problem of application of the Public Cloud and Hybrid Cloud deployment models, identified in the resources of the world scientific periodical, publications [15, 16, 17] can be singled out. These publications take into account the specifics of securing information that constitutes banking secrecy, but there are no proposals for data depersonalization. In particular, work [20] is devoted to solving the problem of confidentiality of data by encryption, but this imposes significant restrictions on data processing.

The work of other scientists [12] does not fully take into account the specifics of IT security provision for banking institutions. Works [17, 12] contain proposals on the use of cloud technologies, but do not contain a comprehensive analysis of the architecture of banking information systems based on cloud technologies, taking into account regulatory constraints.

As mentioned earlier Core Banking System is usually not the first choice for cloud migration. Even if migration offers great cost saving potential banks don't have enough risk appetite to go for it. Maintaining such a status quo requires keeping a lot of hardware capacity On-Premise to deal with CBS workloads.

Most of this capacity is utilized during end of day procedures when the system is not available for users as resources are completely engaged for on-line analytical processing (hereinafter OLAP) tasks (see Fig.1.). The rest of the day the load on the system is rather low, which means that resources are underutilized and used completely inefficient. In such situation huge capital investments should be made for procurement and operative expenditures for further maintenance of such hardware.

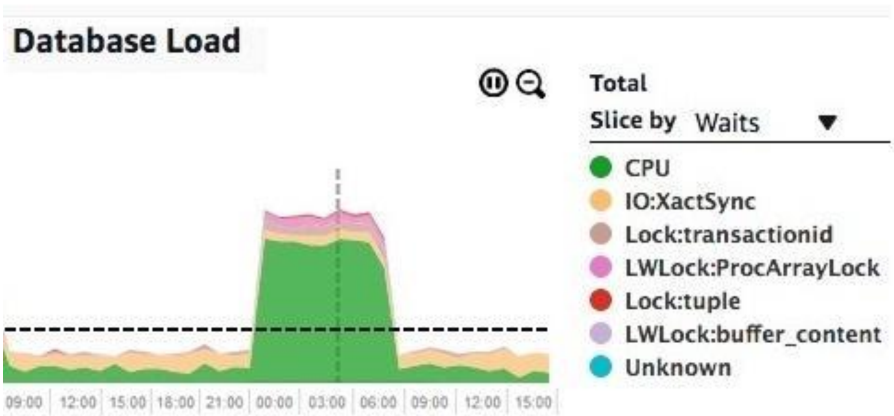

Fig.1. CBS Data Base server resources work load report (Source: [10]).

Moving the OLAP processing work load to the cloud provides possibility to achieve major cost saving and potentially make CBS system available 24/7. As OLAP tasks will be carried out in the cloud on-premise replica will be available for user transactions workload.

Cloud technology offers great efficiency of hardware utilization due to scalability and flexibility, which leads to major cost saving and improved time to market, as underling infrastructure administration tasks are handled by cloud service provider. To uncover this potential it is necessary to manage the risks attached.

Major risks are GDPR requirements (Banks must act as GDPR agents and to be in control of customer data all the time) and local regulators requirements to store the data within boarders of the home country. Unanimousation of the data allows to avoid both risks.

4 Objective and Context of Research

The objective of the research is to:

- to develop the concept on how to migrate huge computing workloads to the cloud, still being compliant with GDPR and national regulator requirements. Unanimousation of the customer data is described as a solution;
- to choose the cloud service provider based on TCO;
- to prepare the IT solution architecture which combines both real time and batch data processing. Unlike traditional use case the data should not only be migrated to the cloud database but also replicated back on premise. Security requirements for data confidentiality integrity and availability must be met;
- to find mathematical solution for the problem of selection of optimal configuration for the cloud computing instance.

The Bank enterprise high-level architecture scheme (see Fig. 2) can be logically separated into 5 blocks.

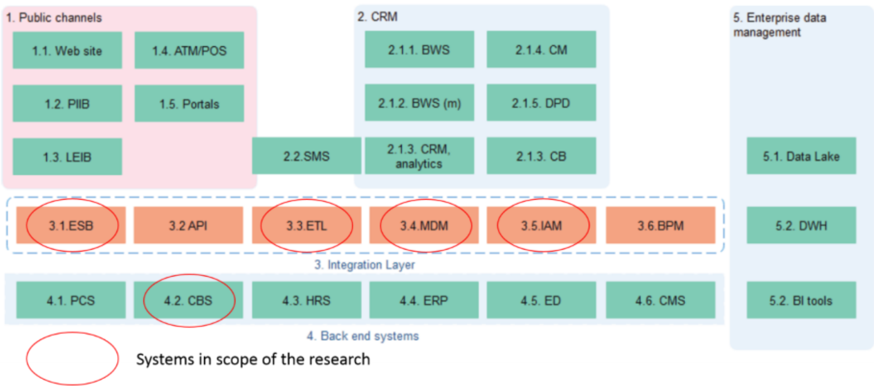

Fig.2. High level architecture scheme of bank`s IT landscape (Source: [10])

Those include Public Channels, Customer Relationship Management, Integration Layer, Back-end Systems and Enterprise Data Management. In scope of this research following systems of IT landscape will be considered:

Integration layer - Enterprise Service Bus, Extract Transform Load jobs, Master Data Management, Identity Access Management

Back end systems - Core Banking System.

Unanimousation of customer data is supplementary measure to ensure confidentiality pursuant to requirements of GDPR, PCI DSS and other similar standards. To depersonalize the customer data replacing it with secret ID in all CBS Data Base (hereiafter DB) schema tables where it resides.

Secret ID formed from unique customer ID applying specified algorithm in Table 1. The data access to be restricted to DB administrators only.

Table 1. Simplified structure of Master Data Management system Dictionary

Unique customer ID	Secret customer ID	Customer ID in system X,Y,Z

MDM system should act as a source and universal point of truth about customer data for all other systems of Bank`s IT landscape. Pre-requisite for MDM is a single front end to enter the customer data and real time distribution to consuming systems including CBS. MDM is a source of Unique and secret customer IDs for CBS and handles customer deduplication process. As this is fully on-premise procedures integration will not be further subject in the research.

5 Results

5.1 Choosing the Cloud Service Provider Based on TCO

Relational DB platform as a service (PaaS) providers which were named leaders by Gartner considered for Total cost of ownership comparison. TCO (Table 2) was calculated with the help of TCO calculators available on official company web sites.

Table 2: Amazon vs Microsoft TCO comparison

Parameters	Amazon RDS	Azure sql
Region	EU Frankfurt	West Europe
Environment	On premise	On premise
Servers type	Virtual Machines	Virtual Machines
Number of virtual machines	2	4
Number of cores	64	64
RAM GB	256	256
DB type	My sql	My sql
Storage type	NAS	NAS
Storage capacity GB	3000	3000
Max concurrent users	NA	1000
Bandwidth GB	3752	2000
Cost per Kw USD	0,36	0,36
on-premise cost USD	370,401	327,228
AWS USD	120,676	208,201
Saving absolute amount USD	249,725	119,027
Saving relative amount	67%	36%

Source: [4, 7, 22, 23]

TCO calculator comparing on-premise and cloud costs with parameterization functionality was not found on Google web site. Amazon and Microsoft TCO were compared.

My SQL Relational database service (hereinafter RDS) was chosen as a base on-premise vs Cloud for TCO comparison. All the parameters were made as similar as possible to get the relevant result. Based on the results of TCO comparison Amazon web services (hereinafter AWS) solution was selected (almost twice more efficient).

5.2 AWS Solution Architecture Overview

The following diagram shows the solution architecture (Fig. 3).

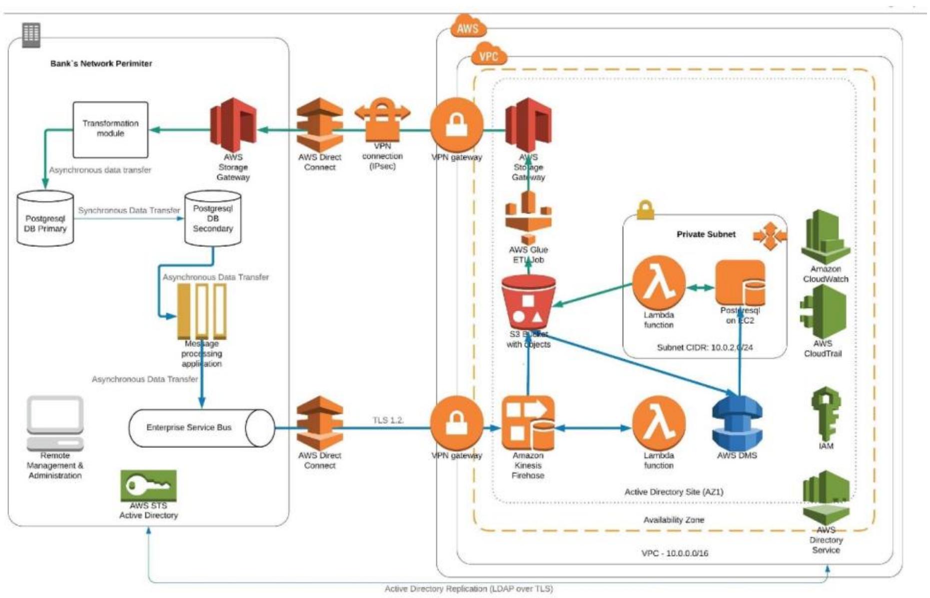

Fig.3. IT solution architecture based on AWS

The principles behind this architecture are the following:

1. Compliance with Service oriented architecture principles.
2. Server less hybrid cloud – heavy workloads uploaded to the cloud to optimize cost, on-premise infrastructure is used for customer data processing.
3. Scalability and maximum efficiency of computing utilization with AWS "pay as you go" auto scaling model.
4. Enhanced security. Integrity, confidentiality and availability is ensured by the following stack of technology - TLS 1.2, Virtual Private Cloud, Encryption, IPSec VPN connection, LDAP over TLS, SAML and IAM roles for each and every Application Programming Interface (hereinafter API) function, audit of every API call.
5. Separation of on-line transaction, analytical processing and batch loads using AWS Lambda on each step of data replication process.
6. Two dataflows:

- outgoing (blue line Fig. 3) from on-premise to the cloud.
- incoming (green line Fig. 3) from Cloud to on-premise.

7. Enhanced reconciliation – replication of all changes in data base real time and validation in Transformation Module after bulk ETL load on-premise.
8. Efficiency – two similar schemas on-premise (customer data included) and Virtual Private Cloud (Unonimoused data) allow to replace INSERT and UPDATE SQL statements with INNER JOIN which is much more efficient in terms of DB server resources consumption.
9. Optimization of concurrent database connections – By buffering records to Simple Storage Service (hereinafter S3) and using Lamda functions as elastic scheduler it is possible to limit the Postgres DB uptime only to actual end of day procedures duration after logical midnight to 3 am, thus optimizing the cost.

The on-premise infrastructure consists of Postgresql DB, Message processing application, ESB, Trasformation module, AWS Storage Gateway, Remote management and administration.

The Postgersql DB – Open source database management systems which offers enterprise scale reliable solution at no licensing cost. Primary and Stand-by databases with synchronous data transfer to ensure failover disaster recovery with minimum downtime.

The Message procession application – Java based application connected with Postgresql DB Stand-by replica with Java DataBase Connectivity (hereinafter JDBC) read only interface. Such approach ensures that there is no negative impact on Primary DB performance due to additional read requests on big number of tables.

The Enterprise service bus – middle ware which uses Message Queue (hereinafter MQ) protocol to route message processing application JavaScript Object Notation (hereinafter JSON) messages to AWS Kinesis Firehose.

The Transformation module–a staging area, which acts as intermediate storage area used for data processing ETL process to replicate data from the AWS cloud to on-premise.

In relational DB with Atomicity, Consistency, Isolation, Durability (ACID) requirements the SQL statements take less server computing resources when the data is stored closer i.e. ideally in one table. The greater the number of table and relations the bigger amount of computing resources is consumed.

Under such circumstances INSERT and UPDATE statements on every row of each table consume a lot of DB server resources. Therefore cost saving effect of migrating workload to the cloud could be totally diminished.

To avoid such effect:

a) Schemas of on-premise and Cloud Postgresql DB replica contain the same columns except those containing customer data. Those columns are cropped and replaced with Secret customer ID during migration to the cloud using AWS Database Migration Service (initial load).

b) INSERT and UPDATE SQL statements are not used by ETL process in Transformation module for daily replication of the calculation results from the Cloud Postgresql DB to on-premise replica.

c) For daily replication of calculation results from the Cloud Postgresql DB to on-premise and adding columns containing customer data the INNER JOIN SQL expression is used. INNER JOIN is much more efficient and does not require such computing power.

The AWS offers enhanced IT Security capabilities for all critical attributes of data security (Integrity, Confidentiality, Availability).

The Virtual Private Cloud is a Virtual perimeter securing systems data, interfaces and endpoints from unauthorized access and modification.

Security events Monitoring is provided by Amazon CloudWatch (CloudWatch) with triggers for suspicious activity and security alerts configurable policies.

Using AWS CloudTrail (CloudTrail) it is possible to audit every API call and store this data historically with visualization in the form of reports and dashboards.

Integrity of inter system data transfer is ensured by Transport layer security protocol version 1.2 (TLS 1.2.)

The AWS Kinesis Firehose is a streaming data processor which transmits the event data from on-premise infrastructure for further processing in the cloud in the form of JSON files [1].

The AWS Lamda is automated scheduler which runs AWS services code via API according to the per-defined schedule or when specific event triggers the Labda function. In proposed IT solution Lambda function also transforms JSON files into comma-separated value (hereinafter CSV) for further processing by Database migration service (hereinafter DMS).

The AWS S3 buckets are used to store the data on all sequence of ETL processes.

The AWS DMS in most frequent use cases is used to perform bulk data upload (Initial load) from on-premise DB replica to cloud DB replica to perform further switch off of on-premise replica. DMS service also supports continuous replication to achieve the same data state/synchronize both on-premise and cloud DB replicas. In proposed solution the on-premise Replica is not switched off therefore after initial load synchronization capabilities of DMS is used perpetually. To optimize costs DMS and Postgresql DB instances uptime is regulated by scheduled lambda function.

The AWS Aurora Postgresql compatible DB – AWS Database system management service for deployment of relational open source DB Postgresql in the AWS cloud. According to Amazon Aurora DB also offers up to three times better performance in AWS cloud than original Postgresql.

The AWS Glue ETL is Extract Transform Load engine which can use triggers to initiate jobs either on a schedule or as a result of a specified event and determines where target data resides and which source data populates the target. In proposed IT Solution AWS Glue is used in conjunction with AWS Lambda functions. The AWS Glue Data Catalog contains references to data that is used as sources and targets of ETL jobs in AWS Glue. The AWS Glue Data Catalog is an index to the location, schema, and runtime metrics of the data. It is possible to run a crawler to take inventory of the data in the data stores and add metadata tables into the Data Catalog. In proposed IT solution AWS glue write a Python ETL script that uses the metadata in the Data Catalog to do the following:

• Join the data in the different source files together into a single data table (that is, denormalize the data).
• Filter the joined table into separate tables by type of legislator.
• Write out the resulting data to separate schema in the Transformation module for further matching with personalized customer data on-premise using INNER JOIN SQL expression.

RDS currently does not offer such a scalability options and memory optimized instances classes as EC2. As specified by CBS performance testing results [11] for relational database input/output operations per second for storage and RAM plays even bigger role than vCPU computing power and number. Therefore memory optimized instance classes were selected for the proposed IT solution. RDS currently does not provide such instance class options [5].

5.3 AWS Auto Scaling Capabilities

The following diagram shows the solution scaling capabilities (Fig. 4).

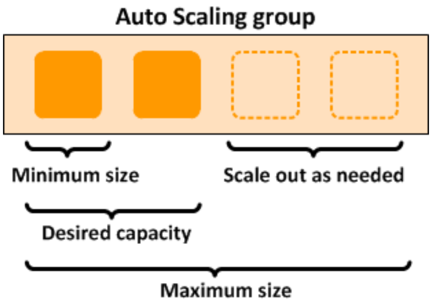

Fig. 4. EC2 horizontal auto scaling (Source: [5])

Horizontal scaling means that EC2 instances can be added automatically when demand for cloud computing resources is increasing.

Auto Scaling creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value.

A target tracking scaling policy assumes that it should perform scale out when the specified metric is above the target value for instance the target value could be 80% of CPU utilization. The AWS EC2 can either launch instances (scale out) or terminate instances (scale in), within the range that the user choose, in response to one or more scaling policies.

Vertical scaling. AWS EC2 allows to customize the number of CPU per instance which is configured manually before instance is launched, changing the configuration requires restart. CPU number settings persist when instances are added automatically (horizontal scaling). Decreasing number of CPUs allow to optimize the licensing costs of software with an instance that has sufficient amounts of RAM for memory-intensive workloads but fewer CPU cores.

5.4 The Mathematical Model for Choosing the Optimal Configuration for a Cloud Server

Consider setting the task of choosing the optimal IT PaaS solution (Postgresql on the EC2 module) for CBS, including type and configuration.

It is necessary to evaluate the workload of the cloud replica of the Postgresql database when executing end of day procedures for closing the Bank's operational day in order to minimize the time and cost of using the cloud service. Empirical observations made it possible to determine the basic requirements for scalability and configuration:

- the size of the database is about three terabytes;
- the client base has up to 15 million clients;
- the number of transactions is 50 million per day;
- the number of customer accounts is 25 million.

Given these parameters and the specifics of ACID requirements for a relational database, OLAP workloads require high computing power that can meet the following minimum server configuration requirements:

- the size of RAM - not less than 256 GB;
- the size of permanent memory is not less than 3 TB.

A typical operating day closing rule usually includes 4 OLAP tasks (indicative list, depending on system architecture, supplier, and implementation).

Task 1. Aggregation of billing and payments according to client's agreements.

Task 2. Commissions for cash services, accrual of interest on current accounts, overdraft, loans and deposits.

Task 3. Aggregation of data on overdue loans, and the beginning of the reference days of overdue debt. Calculation of effective interest rate on loans and deposits. Revaluation of currency position. Calculation of provisions for impairment losses.

Task 4. Formation of the daily balance file (aggregate assets and liabilities). Formation of statistical reporting files.

To solve each task, the minimum required number of processor cores in the server configuration is determined.

Let us introduce the symbols.

k – quantity of tasks of end of day procedure for closing the operational day of the bank;

n – quantity of the module's types (instance);

CPU_j – maximum quantity of cores (CPU) in module of type j;

$f_i\left(CPU_j\right)$ – function of dependency of processing time for the task number i from the maximum quantity of CPU in module of type j;

p_i – minimum required quantity CPU to solve the task number i;

RAM_j – size of RAM in module of type j;

$Storage_j$ – size of storage in module of type j;

R_i – minimum required size of RAM to solve the task number i;

S_i – minimum required size of storage to solve the task number i;

c_j – the cost of using a module of type *j* per hour;

t_{ij} – time of using a module of type *j* to solve the task number *i*.

x_{ij} – quantity of modules of type *j*, which used to solve the task number *i*.

The mathematical model of the problem has the form:

$$F_1 = \sum_{i=1}^{k} \sum_{j=1}^{n} t_{ij} = \sum_{i=1}^{k} \sum_{j=1}^{n} x_{ij} \cdot f_i \left(CPU_j \right) \to \min \tag{1}$$

$$F_2 = \sum_{i=1}^{k} \sum_{j=1}^{n} c_j \cdot t_{ij} = \sum_{i=1}^{k} \sum_{j=1}^{n} c_j \cdot x_{ij} \cdot f_i \left(CPU_j \right) \to \min \tag{2}$$

$$\begin{cases} \sum_{j=1}^{n} CPU_j \cdot x_{ij} \ge p_i, \\ \sum_{j=1}^{n} x_{ij} \cdot RAM_j \ge R_i, \\ \sum_{j=1}^{n} x_{ij} \cdot Storage_j \ge S_i, \\ i = 1,...,k \end{cases} \tag{3}$$

$$x_{ij} \ge 0, \; x_{ij} \text{ – whole.}$$

The problem (1) - (3) relates to the class of tasks of multicriteria optimization. The target function (1) is the condition of minimizing the time of using the cloud service and the target function (2) minimize the cost of its use.

Solving problems with several functions is a rather difficult problem, even for two functions [19]. With the practical use of the model, we can apply a method of priorities, that consists in the fact that at the first stage we will solve a one-problem problem with a smaller target priority (for example, the problem (1), (3)), and in the second stage, the problem (2), (3) with the include to the system of restrictions (3) an inequality, which does not allow to deteriorate the optimal value of the objective function (1) obtained in solving the optimization problem (1), (3) in the first stage.

Consider a numerical example of using the model. In [14] shows the options for possible configurations of cloud servers. Consider setting the task of choosing the optimal IT PaaS solution (Postgresql on the EC2 module) for CBS, including type and configuration.

5.5 Program realization of the mathematical model

To solve the problem (1) - (3) the system MathCAD 15.0 was used. In Fig. 6 shows the initialization process of the model's initial data.

When constructing a matrix M, the dependence of the time of the task on the quantity of server cores was used to empirical investigations of the authors of the article, which was conducted using a 32-core processor. The time for each of the four closure tasks is presented in Table 3.

Table 3: Time to solve the task by 32-core processor

Task number	1	2	3	4
Time to solve the task	5	30	40	45

Source: Creation of authors

The values of the elements of the matrix M is offset by extrapolating empirical data to another number of processor cores.

In Fig. 5-6 depicts a phased process of forming a mathematical model for optimizing the time of use of cloud service.

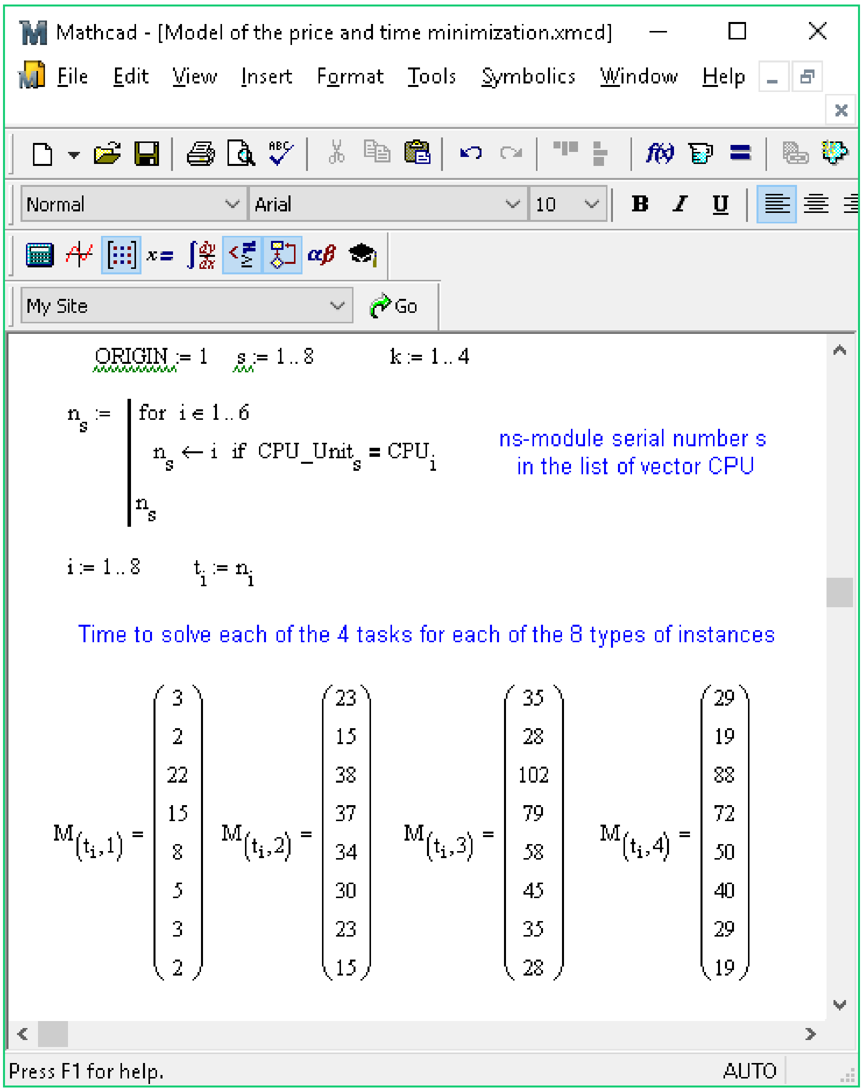

Fig. 5. Find the time to solve the task depending on the type of instance (Source: Creation of authors

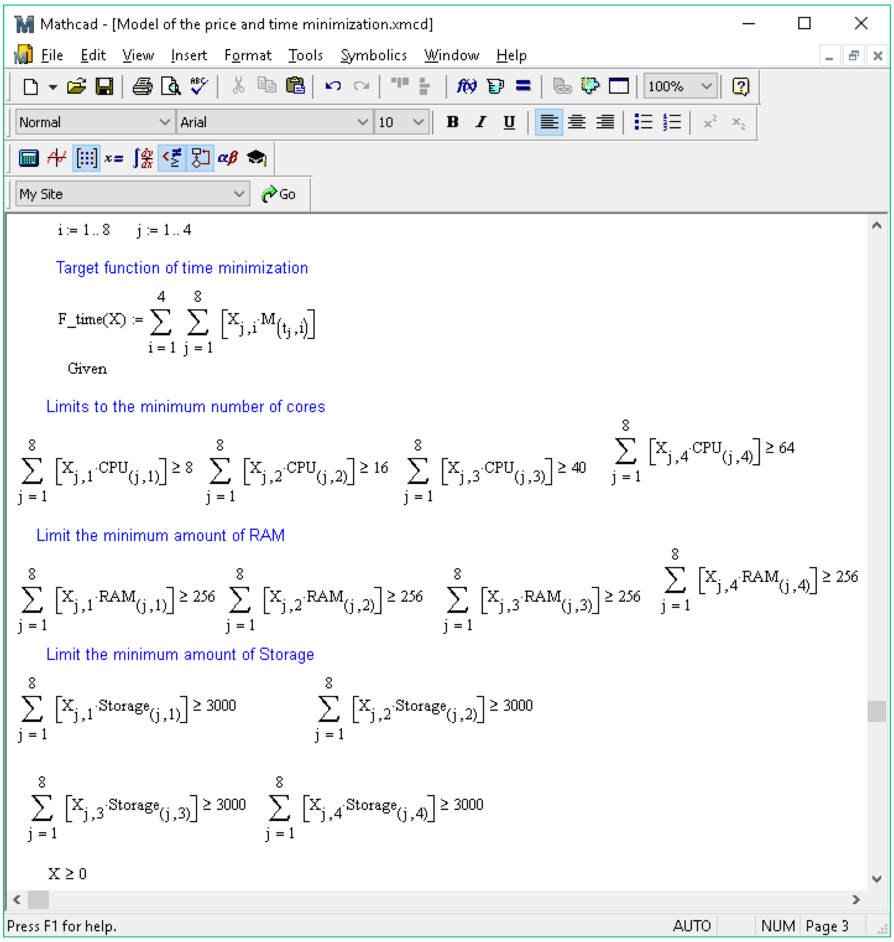

Fig. 6. Formation of a mathematical model for minimizing the use of cloud service time (Source: Creation of authors)

When constructing a matrix M, the dependence of the time of the task on the quantity of server cores was used to empirical investigations of the authors of the article, which was conducted using a 32-core processor. The time for each of the four closure tasks is presented in Table 3.

When constructing a matrix M, the dependence of the time of the task on the quantity of server cores was used to empirical investigations of the authors of the article, which was conducted using a 32-core processor. The time for each of the four closure tasks is presented in Table 3.

In fig. 6 shows the result of time optimization of the cloud service. To get the result, use the Minimize function of the MathCAD system. According to the obtained solution, the configuration is optimal, shown in the table 4. The minimum server time for such a hardware configuration is 64 minutes.

Table 4: The optimal configuration of a set of servers

Task number	1	2	3	4
Name of server type	X1 Extra High-Memory 32xlarge	X1 Extra High-Memory 32xlarge	X1 Extra High-Memory 32xlarge	X1 Extra High-Memory 32xlarge
Task time, min	2	15	28	19

Source: Creation of authors

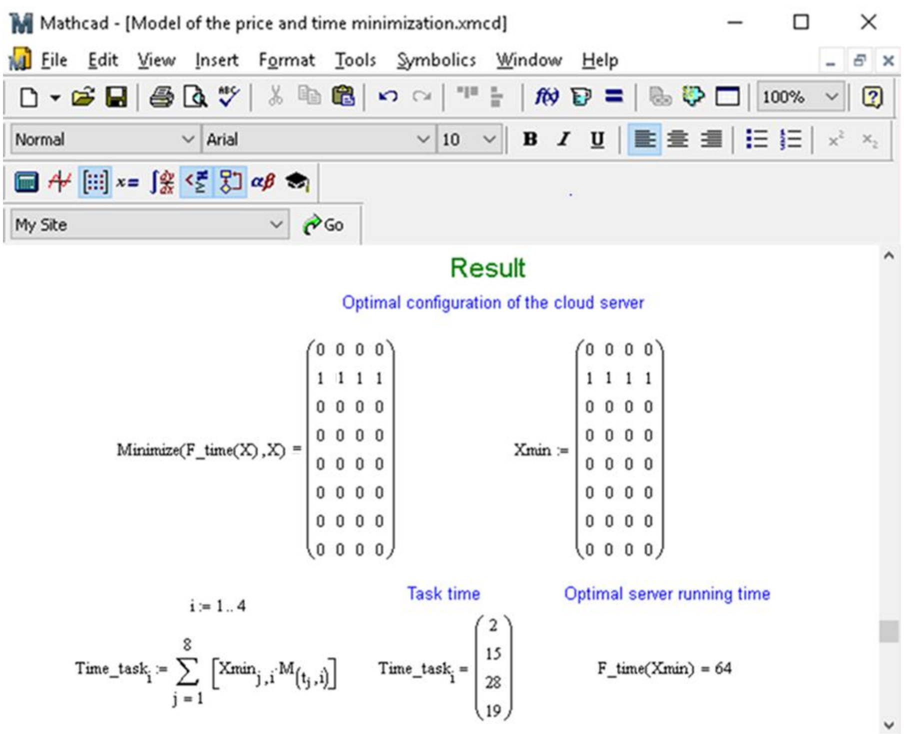

Fig. 7. The optimal configuration of a set of servers (Source: Creation of authors)

Solving the multicriteria problem (1) - (3) with the main target function of minimizing the cost of using cloud service is by connecting to the system (3) the restriction

$$\sum_{i=1}^{k} x_{ij} \cdot CPU_j \cdot f_i\left(CPU_j\right) \leq 64$$

In this case, the minimum value of the cost of using the cloud service is

$$F_{2\min} = 15.15145\$$$

6 Conclusion

IT solution of cloud based banking system designed in scope of the research allows to migrate huge computing workloads to the cloud, still being compliant with GDPR and national regulator requirements. Unanimousation of the customer data is described as a solution for mitigation of risks related to customer data confidentiality and necessity for customer consent to place the data to the cloud.

Based on the results of TCO comparison AWS solution was selected. AWS solutions are also beneficial in terms of Vendor dependency risks minimization. As AWS offers platforms compatible with most of the commercial and open source software and it is possible to migrate the data from cloud back to on-premise if necessary. In the solution described by the author this data migration functionality is used for daily replication of the data from the cloud to on-premise.

IT solution architecture designed by the Author combines both real time and batch data processing. Unlike traditional use case the data is not only be migrated to the cloud database but also replicated back on-premise. Security requirements regulated by the standards for data confidentiality integrity and availability are fully met with respective cloud based technology.

Mathematical solution for the problem of selection of optimal configuration for the the cloud computing EC2 instance was found. Following the model objectives of target function of minimum cost and maximization of computing capacity EC2 memory optimized instance class type X1 Extra High-Memory 32xlarge was found as optimal. In the multi criteria options for decision and constrains defined the calculation result showed that this is the most cost efficient instance which is also the most powerful instance offered by AWS. Single instance capacity is enough horizontal scaling is not necessary. The CBS end of day procedures workload is defined to large extent by the number of customers and customer transactions per day. As customer base and average number of transactions are relatively stable numbers fluctuation does not show huge peaks and falls.

Vertical scalability (decreasing number of CPUs) was not considered as Postgersql DB is open-source SW and does not require license procurement. According to Amazon pricing model decreasing number of EC2 instance CPUs does not effect cost. In case of commercial DB management system it might make sense to decrease the number of CPUs as license pricing is usually linked to the number of CPUs.

References

1. Amazon Kinesis cloud service. Cloud guru web site 2019, https://read.acloud.guru/deep-dive-into-aws-kinesis-at-scale-2e131ffcfa08, last accessed 2019/05/11
2. Amazon S3 web service SLA. Amazon web services official web site 2019, https://aws.amazon.com/ru/s3/sla/, last accessed 2019/05/11
3. Amazon web services standards compliance. Amazon web services official web site 2019, https://aws.amazon.com/ru/compliance/programs/, last accessed 2019/05/11
4. Amazon web services TCO calculator. Amazon web services official web site 2019, https://aws.amazon.com/ru/tco-calculator/, last accessed 2019/05/11
5. Amazon web services technical documentation. Amazon web services official web site 2019, https://docs.aws.amazon.com/index.html, last accessed 2019/05/11
6. Antonopoulos, N., & Gillam, L. (2010). Cloud computing. London: Springer, 382.
7. AWS vs Azure vs Google cloud comparison. Red pixie web site 2019, https://www.redpixie.com/blog/aws-vs-azure-vs-google-cloud, last accessed 2019/05/11
8. Baglai, R.: Cloud identity access management for banks. Scientific Letters of Academic Society of Michal Baludansky, no. 2A, 16-25 (2018).
9. Baglai, R.: Cloud technology security threats for banks. Information processing systems, 1 (152), 127-135 (2018).
10. Baglai, R.: Research of deployment models of cloud technologies for banking information systems. Technology audit and production reserves, 3/4(41), 47-52 (2018).
11. Benchmark Results for Temenos T24 (2010). Intel official web site, https://www.intel.com.br/content/dam/doc/report/performance-xeon-7500-temenos-t24-with-sql-server-2008-r2-on-intel-based-nec-servers-report.pdf, last accessed 2019/05/11
12. Carlin, S., Curran, K.: Cloud computing technologies. International Journal of Cloud Computing and Services Science, 1(2), 59 (2012).
13. de Meijer, C., Brown, A.: Transaction banking in the cloud: Towards a new business model. Journal of Payments Strategy & Systems, 8(2), 206-223 (2014).
14. EC2 instances configuration database. Ec2instances info website 2019, https://www.ec2instances.info/
15. Ekanayake, J., Fox, G.: High performance parallel computing with clouds and cloud technologies. In International Conference on Cloud Computing, Springer, Berlin, Heidelberg, 20-38 (2009).
16. Fenu, G., Surcis, S.: A cloud computing based real time financial system. In 2009 Eighth International Conference on Networks, 374-379 (2009).
17. Ganon, Z., Zilbershtein, I. E.: Cloud-based performance testing of network management systems. In Computer Aided Modeling and Design of Communication Links and Networks, 2009. CAMAD'09. IEEE 14th International Workshop on, 1-6 (2009).
18. Microsoft Azure TCO calculator. Microsoft official web site 2019, https://azure.microsoft.com/en-gb/pricing/, last accessed 2019/05/11
19. Yemets, O. A., Roskladka, A. A.: Algorithmic solution of two parametric optimization problems on a set of complete combinations. Cybernetics and Systems Analysis, Volume 35, Issue 6, 981–986 (1999).
20. Sharma, R., Trivedi, R. K.: Literature review: cloud computing–security issues, solution and technologies. International Journal of Engineering Research, 3(4), 221-225 (2014).
21. Google TCO calculator. Google official web site 2019, https://cloud.google.com/pricing/
22. Microsoft Azure TCO calculator. Microsoft official web site 2019, https://azure.microsoft.com/en-gb/pricing/, last accessed 2019/05/11