Characteristics of Ecosystems of Quantum Computing and Prospects for Their Use in Transport

Petr Polikarpov[a], Nikolay *Uvarov*[b] and Anatoly Khomonenko[c]

[a] *LLC "Abakus, Bolshaya Pushkarskaya street, 47 lit a, St. Petersburg, 197101, Russian Federation*
[b] *St. Petersburg information and computing center - structural unit of the main computer center - branch of the open joint stock company "Russian Railways", Borovaya street, 57 St. Petersburg, 192007, Russian Federation*
[c] *Emperor Alexander I St. Petersburg State Transport University, 9 Moskovsky pr., Saint Petersburg, 190031, Russia*

Abstract
An overview of modern software ecosystems enabling quantum computing is presented. The differences between ecosystems of quantum computing (EQC) and ecosystems of classical computing are highlighted. Provides a brief description of the most advanced quantum computing ecosystems (Google, IBM, Rigetti, Azure, Amazon, Intel, Strangeworks, IonQ). The current state of EQC, the possibilities of their use for various applications and development prospects are analyzed. Perspective areas of application of quantum computing in transport are considered.

Keywords
Ecosystems of quantum computing, quantum computing in transport, big data, logistic optimization problem, Google, IBM, Rigetti, Azure, Amazon, Intel, Strangeworks, IonQ, quantum communication, machine learning.

1. Introduction

[1] Advanced countries, including Russia, have adopted programs for the development of technologies united by the general term "Quantum computing" [1]. The progress made in this area in recent years allows us to talk about the "Quantum Revolution 2.0". Russia is one of the seventeen countries that have adopted and implemented a state strategy in the field of quantum technologies. The main directions of development here are: actually, quantum computing, quantum communication and quantum machine learning.

As noted in [4], at present, quantum computers already exist, but they cannot yet be widely used due to the shortcomings of the equipment. The problems and main directions of the development of quantum computers are increasing the number of qubits, reducing the noise level and increasing the lifetime of quantum states (the so-called "depth of calculations"). The increase in the number of kills, all other things being equal, allows us to use algorithms that correct, due to redundancy, errors caused by noise. The development of error-resistant algorithms is one of the priority tasks in the development of quantum computing technologies. Now there is a time of noisy quantum computers of medium scale, the so-called NISQ (Noisy Quantum of Intermediate scale).

As shown in [5], quantum computing cannot be used independently at the present time. The infrastructure of data preparation, calculation management, interpretation of output data and presentation of results is necessary.

2. About the term "Ecosystem" and its varieties

The term "Ecosystem" was introduced by the English geobotanist Arthur Tensley in [6] to denote the integration of the biotic

[1] Intelligent Transport Systems. Transport Security - 2021, May 14, St. Petersburg, Russia
EMAIL: p.polikarpov@abaqs.ru (P.V. Polikarpov); nick553@mail.ru (N.K. Uvarov); khomon@mail.ru (A.D. Khomonenko); ORCID: 0000-0003-4084-9634 (P.V. Polikarpov); ORCID: 0000-0002-4046-2619 (N.K. Uvarov); ORCID: 0000-0003-0411-258X (A.D. Khomonenko)

community and its physical environment as a fundamental unit of ecology in the hierarchy of physical systems that span the range from the atom to the universe. Ecosystems are the main structural units that make up the biosphere.

Over time, analogs of ecosystems have appeared, denoting systems to ensure functioning in modern society, such as "Business Ecosystems". In particular, in [7], groups of enterprises and the relationships between them, interacting with each other within the same niche in the software and services market, are defined as software ecosystems. These relationships between parts of an ecosystem often rely on a common technology platform. Examples of software ecosystems today are Apple, Google, Microsoft, and the open source ecosystem.

Among software ecosystems, software development ecosystems stand out. In particular, [8] provides a brief topical overview of the current state and prevalence of software development tools among developers.

The software development ecosystem should include the following main components:

- programming languages;
- tools for creating program code;
- tools for building diagrams that display the structure of the program code (class diagrams);
- software debugging tools;
- sets of libraries and components for the implementation of algorithms, exchange and presentation of data.

For example, the Microsoft software development ecosystem [9] includes:

- programming languages C ++, C # and others;
- a set of development tools for MS Visual Studio, including, in particular, debugging tools, a class designer, a data structure designer;
- numerous SDKs and libraries;
- cloud environment for software development and deployment;
- Windows operating systems of various versions.

With the emergence and development of quantum computing, one can distinguish quantum computing ecosystems (QCE), a distinctive feature of which is the use of a quantum computing module and / or its emulator. These ecosystems provide access to quantum computing and serve to:

- manage existing physical quantum computing devices,
- evaluate the efficiency of the implementation of quantum algorithms on future devices,
- study the concepts of quantum computing,
- check quantum algorithms and their implementations,
- teach quantum computing.

3. Quantum computing ecosystems

General remarks

Quantum computing consists of the following main steps:

1) data preparation,
2) preparing a quantum computing scheme,
3) loading a quantum computing scheme and raw data into a quantum processor,
4) performing computation in a quantum processor,
5) receiving data from a quantum processor,
6) interpretation of data.

On a conventional (classical) processor, steps 1), 2) and 6) are performed, steps 3) and 5) are associated with the transfer of data between the quantum and classical processors, and step 4) is performed on the quantum processor. When debugging software at stage 4), a quantum processor simulator is used.

Let's consider the most developed, at the present time, QCE.

Google ecosystem

Google is developing various tools built into the company's extensive hardware and software infrastructure [10]. The main components are: the Cirq framework and the TensorFlow Quantum and Fermion application libraries.

Cirq works with quantum computing schemes. Quantum programs in Cirq are represented by "Scheme" and "Schedule", where "Scheme" represents the quantum circuit, and "Schedule" represents the quantum circuit with information about the sequence of actions. Programs can run on local simulators.

TensorFlow Quantum is a special library that allows Cirq circuits to be used as TensorFlow tensors, in addition, it contains specialized layers (tf.keras.layers) for recurrent neural networks. This library is an example of a mixed quantum-classical approach.

Open Fermion is a specialized library of algorithms for modeling chemical processes.

QCE Google provides access to its quantum ***50-qubit computer*** and related ***simulators***.

IBM ecosystem

QCE developed by IBM is presented in [11]. The software shell that provides access to the IBM quantum computer is called ***Qiskit***. The basic programming language for using Qiskit is Python. The ecosystem includes a quantum assembler that provides access to operations with the lowest-level qubits – ***OpenQASM***.

The QCE includes the IBM Quantum Composer tools for constructing quantum computing circuits and the IBM Quantum Lab development environment. The user is provided with the following main modules:

- module for composing quantum programs at the level of circuits and pulses with optimization and taking into account the physical characteristics of a particular physical quantum computer (***Terra***);
- simulator of quantum computing and simulator of noise errors (***Aer***);
- subsystem of noise reduction in quantum circuits (***Ignis***);
- library of quantum algorithms (***Aqua***).

QCE provides access to a line of proprietary quantum computers. At the time of this writing, the most powerful were a 27-qubit processor with a long characteristic lifetime of quantum states and a 67-qubit computer with a shorter characteristic lifetime of quantum states.

For debugging programs and researching algorithms in this QCE, access to various simulators is provided, with a capacity of up to 1000 qubits.

Rigetti ecosystem

The Rigetti QCE is described in [12]. A feature of this QCE is the desire of developers to provide a minimum delay for transferring data between quantum and classical processors at the stages of loading data into a quantum processor, transferring a computation circuit to a quantum processor, and receiving computation results from a quantum processor.

QCE Rigetti contains all the necessary modules to perform quantum computing.

Quantum operating system. Access to the quantum operating system is provided through network APIs. At this level, basic services are implemented, such as:

- user authentication, service authorization,
- control of the computation scheme and its transfer to the quantum processor,
- memory management,
-control of simultaneously-running processes.

This API is accessed using the Rigetti SDK software.

Quil instruction language [13] for programming quantum computing. This language describes quantum circuits at the lowest level, interaction with a classical processor and memory management. The Quil-T extension provides access to the lowest level of qubit control.

ForestSDK programming tools include the ***pyQuil Python*** library and the ***Quilc*** optimizing compiler, which can be configured to create programs on non-Rigetti quantum processors.

A ***31-qubit quantum computer*** is used to execute programs. Various noise simulation simulators are used for debugging.

Azure Quantum ecosystem

Microsoft is developing a set of tools and technologies [14, 15] integrated with the ecosystem of classic software. QCE uses the ***Q#*** programming language and the ***Quantum SDK*** library of tools. User access is provided through the Azure Quantum cloud platform.

This QCE does not have its own quantum computer. Access to Honewell and Quantum Circuits quantum computers is provided.

Amazon ecosystem

QCE, developed by Amazon, bears the commercial name Amazon Braket [16]. EKV uses widely used open source development tools for classic software: Jupiter notebooks with libraries installed in them. The Amazon Braket SDK is a development platform that you can use to create quantum algorithms and run them on any compatible hardware accessed through Amazon Braket. This platform contains popular quantum algorithms and components for working with neural network training. For debugging, several simulators with added noise and a tensor neural network simulator are used. The programs are launched on quantum computers Rigetti, Wave-D, IonQ.

Intel

Intel reports that work is underway to create an QCE [17], but so far nothing has been presented.

Strangeworks ecosystem

This ecosystem collects various quantum services for access through a single entry point [18]. Provides its own development environment for Python programs and access to quantum computers Honeywell, Rigetti and others.

IonQ

QCE IonQ provides access to its own quantum computer [19] through its own software based on the Python language, as well as through the services Amazon Braket and Azure Quantum.

4. Prospects for the development and application of QCE in transport

The material [20], presented by specialists from one of the industry leaders (IBM), describes the development plan for QCE. This plan reflects the general prospects for the industry: by the end of 2022, the achievement of 400 qubits and the real use of QCE in scientific research, machine learning, solving optimization problems and in the financial industry.

It is assumed that with the development of QCE, software shells will develop that hide from the user the details of the implementation of quantum algorithms. The development of QCE will thus make it possible to solve applied problems for the solution of which the computational power of classical supercomputers is not enough.

With the development of quantum computers and QCE, quantum computing algorithms will also develop in applications for optimization problems and for analyzing big data. The importance of big data analysis in transport is shown in [21].

5. Logistic optimization problem

Let us consider a possible approach to solving the problem of optimizing transport logistics using a quantum computer. The problem of optimization of Mozhe-Kantorovich logistics in the trivial formulation of a linear programming problem can be represented as follows [21]:

$$\sum_{j}^{n} x_{i,j} = a_i ,$$

where i=1,…, m.

$$\sum_{i}^{m} x_{i,j} = a_j ,$$

where j=1,…, n.

$$\sum_{i}\sum_{j} C_{i,j}\, x_{i,j} \rightarrow \min,$$

where: m, n – the number of points of consumption and production of a homogeneous product;

a_i – production volume at the i-th production point;

b_j – the volume of production at the j-th consumption point;

C_{ij} – costs of transporting a unit of goods between nodes i and j.

This problem, as has been repeatedly shown, is NP-hard, which means that there is no time-polynomial algorithm for solving this problem.

As shown in [22], the combinatorial optimization problem, to which the transport logistics optimization problem is reduced, can be reduced to the problem of fulfilling the boundary conditions defined using n bits and m constraints:

$$C(z) = \sum_{a=1} C_a(z) ,$$

where z – an n-bit string and C_a (z) = 1 if z satisfies constraint a and 0 otherwise.

Considering the representation of n bits by n qubits, in [23] is presented the Quantum Approximate Optimization Algorithm (QAOA). It is shown in [24] that the QAOA allows solving the problem in polynomial time. The QAOA functionality are implemented in the main QCE, such as Azure Quantum, Cirq, Qiskit.

6. Big data on transport

First things first, we need to confirm that big data methods are really that important in modern transport area. Initially the sign of big data was a set of 3 Vs: volume, velocity and variety [25].

Even the basic analysis of today's state of transport is enough to say that it works with big data:

Volume. How it was stated before, transport now is under a huge digitalization, despite the fact that there is a lot to achieve yet. Thanks to that, lots of new sources of data were discovered. More sources mean more data. Today every transportation machine is equipped with tons of sensors, which data needs analysis. And this is only a few examples of data sources: data can be taken from logistical systems, road sensors, CCTV etc.

Variety. Sources themselves became more heterogeneous. Some stream bit data, some – text data. There are sources with video and audio data. It all depends on porpoise of source and possibilities of application its data.

Velocity. Modern data transmitted through streams. To provide relevant processing and analysis of data, systems must have enough power. This concerns transport too, where computation done in time can prevent delays or even accidents.

Some classifications add different signs of big data, like veracity viability. One worth mentioning is value [26]. It denotes the expediency of processing data for economical purposes. In modern transport area applicability of this sign is obvious, the correct analysis of data can optimize technical and logistical side of business, that can lead to decrease in expenses.

Big data processing opens a door to many new services and upgradeability of old ones. These are only few of them [27]:

- Monitoring of infrastructure. With bigger data amount, infrastructural analysis becomes more accurate, resulting in more effective management of infrastructure itself.

- Increasing in mobility of services. Today users need to have everything in their hand – on their smartphone. Transport services are no exception. Processing of big data grant a possibility of providing services in a user-oriented way, also giving company additional data, which can be useful in future.

- Understanding client's needs. One of the most popular sources of big data – social networks. Analyzing information from there can give a lot of useful business-details. First and foremost, this concerns client's demands, which is important on transport as on any other area.

- Human flow visualization. Visualization is sometimes added to big data signs, which stresses its importance. Indeed, this is very powerful tool, that can solve a lot of tasks of transport management, also being simple and understandable to average person and not only IT-experts.

- Traffic control. This item speaks for itself. More sources and data results in more accurate prediction and more correct traffic control.

- Machine state diagnostics. This item is especially relevant on the railroad, because technical checkup and repairs are one of the most important parts of transportation on train, and they are not achievable without big data.

It is worth to mention Smart Transport initiative in a Smart City idea. The are many definitions of Smart City, but for the sake of simplicity let's focus on that is a city, where all of main components of infrastructure are most intellectual, connected and efficient because of IT usage [28].

The initiative itself implies intellectualization of transport in the city, including community transport and roads. These are main aspects of Smart Transport [29]:

- Smart Roads (changing road markings, accident detection, smart lights, sensors equipment).

- Smart traffic lights (data analysis to regulate traffic relatively to traffic density).

- Smart cars (processing data from various sources to provide autopilot and driver safety).

- Smart road signs (analysis of data about road state, sign changing).

In the introduction to this article, the problem of power of modern IT-systems is concerned. One of the solutions to this is quantum computers. Next, we give a short specification of quantum computing and its capabilities of processing big data.

7. Quantum computing and Big Data

The main difference between quantum and classical computing is a presence of two physical phenomena, observed only in microworld: quantum superposition and entanglement.

Quantum computers operate not in bits, like classical PCs, but in qubits. The difference between them is in the amount of their possible states. Classical bit interprets the presence or absence of electrical current and can be either 0 or 1. Qubit interprets one of the properties of a particle (for example electron's spin or photon's polarization) and can not only be $|0\rangle$ or $|1\rangle$, but in a state of superposition of these values ($d_0|0\rangle + d_1|1\rangle$), where d_0^2 and d_1^2 – possibilities of qubit transition to that state.

Quantum superposition phenomena is hard to describe, because in classical physics there are no suitable experimental examples. To put it simply, superposition is a state, where qubit is 0 and 1 at the same time. Unfortunately, this phenomenon cannot be seen or felt by a human. Every attempt to measure it results in qubit collapsing in one of the states with certain possibility. However, qubit in that state still can be worked with.

Its important to know, that qubit states are written in brackets because they are actually vectors. For example, vector $|0\rangle$ can be written like $\begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle$ can be viewed as $\begin{bmatrix}0\\1\end{bmatrix}$.

Second distinctive feature of quantum world – entanglement phenomena. Two or more qubits can transit into one superposition for all of them. Measurement of one of these qubits will instantly collapse superposition not only for it, but for every other entangled qubit. All of them will transit into one prescribed state. For example, if one will measure first qubit in state ($d_0|00\rangle + d_1|11\rangle$) and get 0, then it certain, that second qubit is 0 too.

Entanglement phenomena is used in computations, but do not give any speedups. It mostly usable for the quantum security. From the other side, superposition can give that necessary push in computer power for big data processing.

For example, we can consider two bits and two qubits. Two bits can potentially turn into one of four states (00, 01, 10, 11). But in this moment in time two bits can represent only one of these states. Two qubits can also transit into one of these states plus their superposition. This means two qubits can represent in this moment in time all of these sates at once [30].

This brings us to an ideal model of quantum computing's superiority over classical computing, represented on Figure 1.

We need to imagine some function and sets of income data for it. The task is to get every outcome value for every income. In classical computing we need to run function for every income value. In quantum computing, we can set all income values into superposition and run function only for this state. By this method we can compute function only once, instead of n.

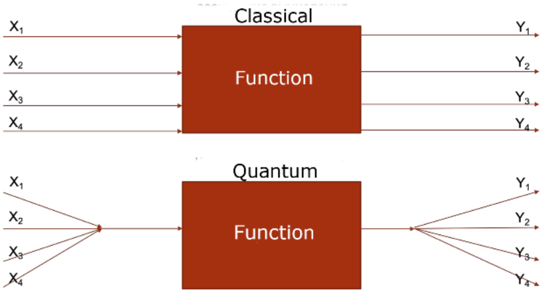

Figure 1: Idealistic explanation of quantum computing's superiority

Of course, this model is very abstract and everything is not so simple on practice. There are many pitfalls, preventing this ideal course of things. To show this, we can look at already implemented on a real quantum computers model – Grover's algorithm of quantum search [31, 32].

Grover's algorithm

Detailed algorithm description is a separate topic. This is a glimpse on principal of its work. There is scheme of algorithm on Figure 2.

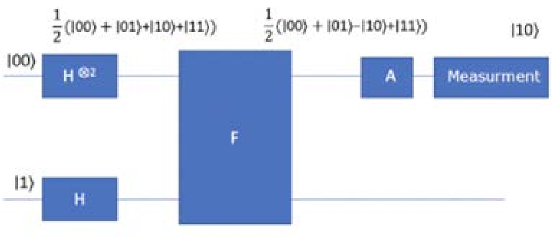

Figure 2: Grover's algorithm scheme

The circuit consists of quantum gates – orthogonal matrices, which apply to qubit states with tensor product. Also, circuit has oracle F – function, representing black box. It is also a orthogonal matrix.

This circuit is designed for two qubits. There is a function $f(x_0,x_1)$ and 4 possible qubits states ($|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$). Function returns 0 or 1, while returning 1 only for one of possible states. We need to find this state.

On circuit entrance two qubits being put in state |00⟩ and one additional qubit in |1⟩ state. Let's observe two upper qubits. This pair goes through Hadamard gate and transits into superposition with equal possibilities for each state. After oracle, one of the states changes its operator to '-'. This is right state. To get the answer, we need to increase its possibility and decrease possibility of other states. For this, superposition goes through A gate – a gate that flips possibilities over average value. At the end, needed state has possibility of 1, and any other has 0.

Quantum speedups

Unfortunately, Grover's algorithm won't be able to work so perfectly with 3 or more qubits. In this situation, after possibilities flip, the possibilities of wrong answers won't go into 0. This problem resolves by repeating the circuit for out outcome data. Every time the wrong possibilities will decrease.

Grover himself computed a value of needed repeats of algorithm for the most believable answer. It is $O(\sqrt{n})$ times [32]. In the other hand, classical algorithm needs $O(n)$ runs. Quantum algorithm gives a quadratic speedup over the classical one. In theory, classical search, running for 100 hours can be run for 10 hours on quantum computer.

Despite the dramatic decrease in time of computing, for the average person tis speedup may seem not so impressive. However, it shows, that quantum technologies already can be used in real tasks as more efficient way.

It needs to be remembered, that quantum computing is in its dawn and many things are simply not yet discovered. Quantum computing's potential is well shown by Shor's algorithm [33]. This operation will take minutes on quantum computer, while it takes decades on classical.

It is worth to mention a quantum machine learning area. ML is vastly used in big data analysis and its speedup would dramatically affect this domain. On picture 3 there are few ML methods, which potentially can be accelerated by quantum computing [34]. On this scheme the theoretical speedups over classical methods are given.

Machine Learning	
Method	**Quantum speedup**
Bayesian inference	$O(\sqrt{N})$
Least squares	$O(\log N)$
Principal component analysis	$O(\log N)$
Support-vector machine	$O(\log N)$
Reinforcement learning	$O(\sqrt{N})$

Figure 3: Theoretical quantum speedups over classical methods

As you can see some methods receive exponential speedup. However, there is still a lot work to do to get an actual value.

8. Conclusions

An important difference between QCE and ecosystems of classical computing is the large role played by the scheme of quantum elements in quantum computing.

Various emulators of quantum computers with support for emulation of different types of noise are used for debugging.

QCE contains the same basic modules as classical computing ecosystems.

Specific features of ECU:

- Quantum computing scheme development environment.
- A set of components for accessing a quantum computer.
- Emulator of quantum computer.
- Quantum computer.

The QCE types by mode of development:

1. On advanced classical computing ecosystems basing such as Microsoft and Google.
2. Completely newly developed systems such as IBM and Rigetti.
3. Some QCE collect a variety of services: software libraries, emulators and quantum computers, for example, Strangeworks.

In terms of application, Google QCE is distinguished, containing a developed toolkit of the machine learning system TensorFlow Quantum.

Google, Azure Quantum, IBM and Rigetti QCE can be used for training purposes, providing a convenient user interface and advanced documentation with examples.

It is proposed to consider as promising tasks for further research on the possibilities of using quantum computing ecosystems:

- applied use of QAOA [24] for solving problems of transport logistics optimization;

- machine learning tasks for monitoring the characteristics of cyber-physical systems and assessing the quality of IT services [35-37];

- problems of integrating tools for processing big data and neural networks for the classification of images and complex objects [38].

References

[1] J. Kung и M. Fancy, «A Quantum Revolution: Report on global policies for quantum technology». Available: https://cifar.ca/wp-content/uploads/2021/05/QuantumReport-EN-May2021.pdf.

[2] F. Arute, K. Arya, R. Babbush и e. al, «Quantum supremacy using a programmable superconducting processor» Nature, pp. 504, 505-510, 24 октябрь 2019.

[3] E. Pednault, J. Gunnels, D. Maslov и J.Gambetta, «On "Quantum Supremacy"» 2019.https://www.ibm.com/blogs/research /2019/10/on-quantum-supremacy/.

[4] J. Preskill, «Quantum computing in the NISQ era and beyound» Quantum, т. 2:79, 2018.

[5] S. S. Andreev, [etc], «Quantum model of computing from a classical programmer point of view» Preprints. Keldysh Institute of Applied Mathematics., v. 178, p. 30, 2018.

[6] Tansley, A.G. The use and abusec of vegetational concepts and terms / A.G. Tansley. Ecology.1935. Vol. 16. No 3. P. 284-307.

[7] D.G. Messerschmitt и C. Szyperski, Software ecosystem: understanding an indispensable technology and industry, Cambridge, Mass.: MIT Press, 2003. 424 p.

[8] JetBrains, «Software development ecosystems» 2020. Available: https://www.jetbrains.com/lp/devecosystem-2020/.

[9] «Overview of the state of Microsoft Software development ecosystem 2020» 2021. Available: https://dev.to/endbird/overview-of-the-state-of-microsoft-software-development-ecosystem-2020-6j6.

[10] «Google quantum» Google, 2021. Available: https://quantumai.google/.

[11] «IBM Quantum» IBM, 2021. Available: https://quantum-computing.ibm.com/.

[12] «Rigetti»: https://www.rigetti.com/what.

[13] «Quil» https://github.com/quil-lang/quil.

[14] «Azure Quantum» Microsoft, 2021. Available: https://azure.microsoft.com/ru-ru/solutions/quantum-computing/.

[15] «Overview azure quantum» https://github.com/MicrosoftDocs/quantum-docs/blob/main/articles/overview-azure-quantum.md.

[16] «Amazon Braket» Amazon, Available: https://aws.amazon.com/ru/braket/.

[17] «Reinventing Data Processing with Quantum Computing» Intel, Available: https://www.intel.com/content/www/us/en/research/quantum-computing.html.

[18] «Strangeworks» https://strangeworks.com/

[19] «IonQ,» Available: https://ionq.com/.

[20] K. Wehden, I. Faro и J. Gambetta, «IBM's roadmap for building an open quantum software ecosystem» 04 February 2021. https://www.ibm.com/blogs/research/2021/02/quantum-development-roadmap/. https://www.ibm.com/blogs/research/2021/02/quantum-development-roadmap/.

[21] A. I. Torre-Bastida, J. D. Ser, I. Laña, M. Ilardia, M. N. Bilbao и S. Campos-Cordobés, «Big Data for transportation and mobility: recent advances, trends and challenges,» IET, т. 12, pp. 742-755, 2018.

[22] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford. «Introduction to Algorithms». 3rd. MIT Press, 2009.

[23] E. Farhi и J. Goldstone, «A Quantum Approximate Optimization Algorithm» 2014. https://arxiv.org/abs/1411.4028.

[24] B. Mansouri, «An Analysis of the Quantum Approximation Optimization Algorithm» 23 March 2021. Available: https://arxiv.org/pdf/2103.12791.pdf.

[25] Laney D. et al. 3D data management: Controlling data volume, velocity and variety. META group research note. 2001. V. 6. No. 70. P. 1.

[26] Chen M. et al. Big data: related technologies, challenges and future prospects. Heidelberg : Springer, 2014. V. 96.

[27] Torre-Bastida A. I. et al. Big Data for transportation and mobility: recent advances, trends and challenges. IET

Intelligent Transport Systems. 2018. V. 12. No. 8. Pp. 742-755.
[28] Nam T., Pardo T. A. Conceptualizing smart city with dimensions of technology, people, and institutions //Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times. 2011. Pp. 282-291.
[29] Chris Bernhardt. (2019), Quantum Computing for Everyone. The MIT Press. Cambridge, Massachusetts, London, England. 2019. 214 p.
[30] Pandey A., Ramesh V. Quantum computing for big data analysis. Indian Journal of Science. 2015. T. 14. No. 43. Pp. 98-104.
[31] Figgatt C. et al. Complete 3-qubit Grover search on a programmable quantum computer. Nature communications. 2017. V. 8. No 1. Pp. 1-9.
[32] Grover L. K. A fast quantum mechanical algorithm for database search. Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. 1996. Pp. 212-219.
[33] Shor P. W. Algorithms for quantum computation: discrete logarithms and factoring. Proceedings 35th annual symposium on foundations of computer science. IEEE, 1994. Pp. 124-134.
[34] Biamonte J. et al. Quantum machine learning. Nature. 2017. V. 549. No 7671. Pp. 195-202.
[35] Maksim A. Bolshakov, Igor A. Molodkin, Sergei V. Pugachev. Comparative analysis of machine learning methods to assess the quality of IT services. CEUR Workshop Proceedings, 2020. Vol. 2803, pp.142-149. http://ceur-ws.org/Vol-2803/paper20.pdf.
[36] Lokhvitskii, V.A., Khomonenko, A.D., Bol’shakov, M.A. On the Construction of a Cybervisor for the Intelligent Monitoring and Control of Data Centers. Automatic Control and Computer Sciences, 2019, 53(8), pp. 870–873.
[37] Shardakov, K., Bubnov, V., Kornienko, S. Modeling the operation of a distributed high-load monitoring system for a data transmission network in a non-stationary mode. CEUR Workshop Proceedings, 2020, v. 2803, pp. 107–116.
[38] Kosykh, N.E., Khomonenko, A.D., Bochkov, A.P., Kikot, A.V. (2020), Integration of big data processing tools and neural networks for image classification. CEUR Workshop Proceedings, 2020, Vol. 2556, pp. 52–58. http://ceur-ws.org/Vol-2556/paper9.pdf.