УДК 517.9:621.325.5:621.382.049.77 http://doi.org/10.15407/pp2023.02.049

M. Kosovets, L. Tovstenko

ARTIFICIAL INTELLIGENCE IN CLOUD-BASED MOBILE RADAR COMPUTING

The introduction of Artificial Intelligence into mobile radar computing based on cloud resources has made it possible to combine radar resources at the stage of receiving, processing and presenting information. The radar system has become integral and flexible. The convergence of mobile applications in portable devices with cloud computing is a revolution in the organization of distributed computing. Attention is paid to the architecture of the client part as a neurocomputer distributed in space with deep learning capabilities. Providing analysis of radar data in Real Time for a mobile platform is very important and its implementation with packet data transmission at different stages accelerate the analysis process.
Keywords: Service Oriented Architecture (SOA), Sense-Compute-Actuate (SCA), High Performance Computing (HPC), Message Passing Interface (MPI), virtual machine (VM), Machine Learning and Data Analysis (MLDM), Distributed File System (HDFS), Java Database Connectivity (JDBC).

Introduction

There has been an intensive development of radar technology over the past fifty years in the military, medicine, biology, security, agriculture and others. This trend continues today at an ever faster pace. The development of scientific approaches to the acquisition and processing of radar information is always determined by the achievements of manufacturing technologies for more sensitive radar sensors, which are small in size, weight and low radiation power, as well as the development of new radio wave bands. Computing technologies have received significant development: data parallelization, streaming and neural network calculations, and mathematical modeling. The mechanism of access and data exchange has improved.

Cloud computing makes data mobile by giving users access to the cloud of their choice from any Internet-enabled device without requiring a detailed understanding of the underlying computing technology. The increasing ability of radars to sense the world around them is leading to a growing need for electronic database applications driven by data under the control of workflows. The main focus is on the collection, origin of the data of these workflows, necessary to validate the workflow and determine the quality of the generated data. The challenge is to capture and use a single origin metadata that matches the needs of the domain, minimizing the modification burden on the services and the overhead of the workflow engine and services. The framework is based on the generation of discrete data related to the origin, during the work cycle of the life cycle execution can be aggregated to form complex data and process origin graphs, which can cover different workflows. The implementation uses a loosely coupled architecture where the capabilities of the system meet the needs of detailed information gathering.

This idea raised the level of abstraction for the programmer. Klient needs additional capacity based on demand, drives hybrid cloud concept. That is, the use of a public cloud as a continuation of the internal infrastructure – a platform for automating machine learning procedures, analyzing streaming data; building your own cloud with open source software and cloud security.

1. Client cloud environment for radar information processing

The topic of data analytics in the cloud is huge and booming in the new data-driven open data and design paradigm [1]. Consider data from radar sensors on board an autonomous vehicle. If the value of radar data rapidly diminishes with time coming in every second, and the volume is so large that it cannot be stored, real-time processing or data reduction is the only way to analyze data coming from unlimited data streams, which has created part of the intellectual foundation of modern systems [2].

Machine learning has become central to cloud computing programs. Although machine learning is considered as a part of the field of artificial intelligence, it has roots in statistics and the theory and practice of mathematical optimization. It has gained in importance in recent years as a number of critical applications have taken place. This includes real-time radar image recognition. Algorithmic progress and more powerful computers make it possible to train deep neural networks. It introduces some of the core machine learning tools available in public clouds, as well as toolkits that can be installed in a private cloud.

Summarizing the above, it can be noted that in recent years a significant contribution has been made to the organization of the client part – the operator's workplace, improving access to information databases [3]. They began to introduce elements of artificial intelligence, teaching search engines to find information on the distinguished features of the problem being solved. Also, attention was paid to the architecture of the client part, which can be explained by a small selection of standard computing tools. Today, the client part is a neurocomputer distributed in space with deep learning capabilities. While batch analysis of big data is important, real-time data analysis is becoming increasingly critical. For example, data from radars that control complex systems for targeting weapons to objects on board a vehicle [4].

To improve performance, we introduce the idea of container programs, which allow us to create an application container that will run on any machine, and we do not need to make changes to the machine itself. This is a way to share ready-to-run applications such as a mobile radar server or a simulation application. There are various ways to scale applications in the cloud for greater parallelism. You can create an HPC (High Performance Computing) style cluster in the cloud to run MPI (Message Passing Interface) applications, or based on Docker on a distributed cluster. Docker is a software platform for rapid development, testing and deployment of applications. Docker packages software into standardized building blocks called containers. Each container includes everything the application needs to run: libraries, system tools, code, and runtime.

However, it was clear that the need for reproducible container-based environments was especially important for radar computing, and would really help both users and cluster administrators. As a rule, the centers provide standard libraries that meet the needs of most users. Containers allow you to install software [5].

Large data sets from radar sensors, represented by the Internet of Things (IoT), can perceive information, exchange data, calculate and receive data streams from radar sensors and contribute to the emergence of the big data paradigm. We will discuss the new architecture of the Internet of Things (IoT) radar system, large-scale additional sensor networks, the integration of sensor networks, radar sensor data and related methods for collecting context, problems of cloud management, storage, archiving and processing of radar sensor data.

The world is filled with devices including sensors and data processors. This concentration of computing resources makes it possible to detect, capture, collect and process real-time data from connected radar devices serving many different applications, including environmental monitoring [6]. These developments have led us to the era of the Internet of Things (IoT). However, a sense of the environment and the objects that inhabit this environment has become synonymous with the introduction of pervasive or ubiquitous computing. Sensor networks are the main enabler of IoT. The IoT has three unique features: Periodic Probing, Regular Data Collection, and Sense-Compute-Actuate (SCA) cycles. But this will only be useful if the "terabytes" of data it generates can be collected, analyzed and interpreted [7].

There are a number of problems associated with the processing and analysis of data. Therefore, extensive data have become popular [8] and are analyzed based on some of its characteristics. There are three characteristics that can be used to define big data, also known as 3Vs: volume, variety, and speed.

• Volume: data size (terabytes), etc.;

• Diversity: data types in the form of big data received from radar sensors;

• Rate: The rate at which data is generated.

The frequency of processing depends on the user's requirements for obtaining the result. Generally, we can distinguish three main categories: random, frequent and real-time. There are a number of problems associated with big data – collection, storage, search, analysis and virtualization. Additional technologies are identified, such as database parallel processing (MPP), distributed file systems, cloud computing technologies to complement radar data management.

Methods for solving problems have been developed: extraction, transformation, integration, sorting and manipulation of data. The main methods of obtaining data consist of five stages: definition, search, transformation, entity resolution, response to a request. These conceptual steps are more related to the traditional field of data science. However, the technology behind these conceptual steps will differ significantly due to the unique characteristics of the radar data. Radar systems provide data using very complex and advanced sensors. They need high computing power for processing and storage.

Provision of a resource as an Internet technology for external clients. Cloud computing plays a significant role in the IoT paradigm. Cloud storage and processing capabilities are critical to delivering the service vision; a model that has emerged from cloud computing as a style of computing where large-scale IT-related capabilities are delivered «as a service» using IoT.

Cloud computing includes three main layers or models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). In addition to the above core layers, some other layers are also introduced and discussed in literature such as Database as a Service (DBaaS), Data as a Service (DaaS), Ethernet as a Service (EaaS), Network as a Service (NaaS).

In the IoT domain, sensor data is not the only information that will be stored. More importantly, relevant contextual information will always be added to the original sensor data for further retrieval. For example, Sensor-Cloud is a framework that aims to manage physical sensors by connecting them to the cloud. The Sensor-Cloud system uses sensors to describe sensor metadata.

The problems associated with rich data can be divided into two categories: engineering and semantic. The challenge in engineering is to efficiently perform activities such as query and storage. The semantic challenge is to extract information from large amounts of unstructured raw data.

The main problems of big data management are:

• High volume processing using low power digital processing architecture.

• Use of adaptive machine learning methods for real-time data analysis.

• Developping scalable data warehouses that enable efficient data analysis.

Big data consists of useful information mixed with «dirty» (noise, false and raw) data. Advanced systems and innovative technologies will effectively process huge amounts of «raw» data and highlight useful information. This will require a concentrated effort of IoT researchers and practitioners. The achievement is an overview of the current and future uses of rich data in the IoT environment, but there is no claimed disclosure of the architecture of the IoT environment using rich data by abstraction layers [9].

Using the services of Hadoop, YARN, HDFS and others simplifies the work of information processing. So, Hadoop is a freely distributing set of utilities, libraries and a framework for developing and executing distributed programs running on clusters of hundreds and thousands of nodes – this is the world of big data. Highload starts writing logs, then big data is already required for their processing. In general, big data is a very broad field with machine learning and semantic analysis. Actually, there exist four tasks: the first is reading data and three data processing tasks. These tasks are proposed to be solved using Hadoop.

YARN is Yet Another Resource

Negotiator, which is responsible for running tasks on multiple machines, manages the computing resources of the cluster, and simply submits the task for execution. It doesn't know what tasks it passes off, and MapReduce directly executes the tasks that it starts.

When designing HDFS, we simultaneously took into account the ideas that the system should be fault-tolerant, that is, data should be stored on different servers. The system should be distributed so that it can be easily added, removed new servers. A file represented by a name and their content, the content is broken into a set of blocks. When developers and admins work with the command line and data, the work goes through an API written in Java

The second part related to data processing is the second, third and fourth tasks. This is Yet Another Resource Negotiator/MapReduce. The principle of construction is the same as HDFS – there is a master node that controls the entire process and there are nodes – managers – these are daemons running on each of the servers on which data is processed, and with the known availability of resources on this server, you can run any task.

You can try to break them down into data management categories, that is, into files, resources, and allocating computing resources. A framework is the organization of a calculation, the launch of processing tables or other programs – the code of which the user does not write.

Hadoop usually runs on a powerful computing platform using a lot of memory at startup, so if memory is low on the server, Hadoop can be run but complex calculations can't be done. But if there are many small projects, all the data of these projects can be combined within one Hadoop cluster. Hadoop works well with low power servers and can extend their lifespan.

Adding deep learning to the system with the construction of clusters with the ideology of neural networks removes all difficulties and takes storage and processing of huge amounts of information to a new level, otherwise a deadlock is inevitable.

The persistent need to consider large-scale graph-structured data in machine learning and data mining (MLDM) is a significant challenge. Since the sizes of datasets decay, statistical theory suggests that we should use more complex models to eliminate the influence of simpler models and determine more complex signals from the data. At the same time, the computational and storage complexity of large models, combined with rapidly growing datasets, has exhausted the limit of single-step calculations.

2. Limitations of service-oriented architecture and its combination with cloud computing

Cloud computing is dynamically scalable resources and SOA (Service Oriented Architecture) is the concept of loosely coupled services. Each service is independent of the other. Together they can form a complete system. We will give an overview of cloud computing and SOA, and also propose a decision that SOA should be combined with the cloud in order to eliminate the limitations of SOA. Combining the cloud with SOA will increase the availability and reliability of SOA and reduce messaging costs.

Different documents have different definitions of Service Oriented Architecture (SOA). From an architectural point of view, a service is defined as a way to access the functionality of a system or any individual function. Some standard interfaces are used to access these features, and there are also some predefined rules for accessing these services, which are set out in the service description. The service is defined in terms of its elements as an organization for the promotion of structured information standards. SOA is based on loosely connected software parts; each software component provides an individual service. The fundamental idea behind SOA is open access and encapsulation. Although SOA does not include a cloud architectural style, in order to get the maximum benefit from SOA, we must combine service-oriented architecture with cloud computing. This will help us overcome the limitations of SOA.

The main concept of the cloud is

fast delivery and resource scalability. The resources offered by cloud computing are dynamically scalable. Another advantage of cloud computing over traditional computing is its low cost and location independence. Through SOA, different components of application development can be integrated. New features can be added to existing applications or can be retained, but very little research has been done on SOA maintenance tools. Data can be collected with the system by deploying it in the environment in which it is intended to operate. The collected data is analyzed [10].

From an engineering point of view, a service-oriented architecture has the following advantages:

• Language-independent integration: Services in a service-oriented architecture use the XML standard. It focuses on converting data generated in XML format and passing it to another component.

• Multiple components: SOA is based on the concept of loosely coupled components; once these components are developed, they can be used separately. These components can be used with proper reliability and safety assurance. In addition, these services can be combined together to create a new system with higher capabilities.

• Rapid Application Development: Offered by components that are developed using a Service Oriented Architecture; meet some of the organization's business requirements. These blocks can be used separately and later they can be quickly integrated.

• Definition of existing system service and setting standards. With Service Oriented Architecture, we can integrate new systems with old legacy systems without rewriting the new system. This will save capital costs as well as time. Thus, organizations without the overhead of developing new systems from the core.

There are many reasons for the adoption of service-oriented computing in enterprise environments. It's just that the services provided by service-oriented computing are reusable and flexibly integrated into new services.

The cloud computing model deals with dynamically scalable resources. These resources are provided over the network. There are many other benefits that can be taken from cloud computing. This reduces hardware costs, maintenance costs, and the cost of installing access to hardware and software in the cloud. In addition, it promotes resource agility, scalability, and reliability. Locations and devices are independent, adaptive and resilient. There are three unique cloud layers depending on the resource type.

Software as a Service (SaaS) is the most popular layer that provides users with a ready-made program. It ensures that the user will use the Internet host software without using client resources such as installing and running applications on the client's local computer. Each data item has a read lock or a write lock, and there is a distributed cloud consistency and convergence mechanism.

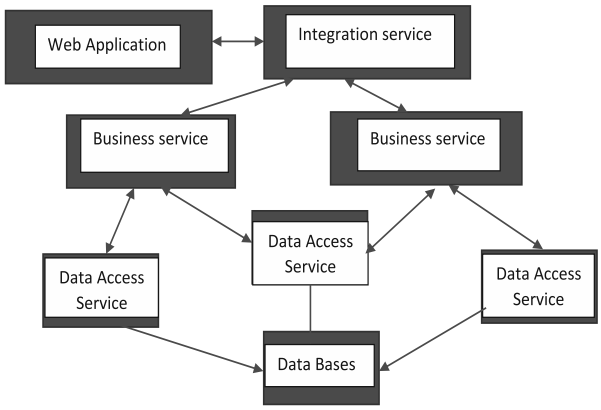

Fig. 1. Service-oriented architecture.

Platform as a Service (PaaS) is the second layer that contains the environment for running the software. An application server can be one that allows developers to deploy a web application without purchasing or configuring their actual servers. The purpose of this model is to ensure data protection, which is very important in an environment where we consider storage as a service. To provide a load balancing service, it is important to ensure safety against shutdowns.

Infrastructure as a Service (IaaS) – This layer shows the exchange of hardware resources to perform services, classically using virtualization technology. With this approach, many users can use the available resources and the resources can be increased on demand. Resources in IaaS are virtual machines that must be managed.

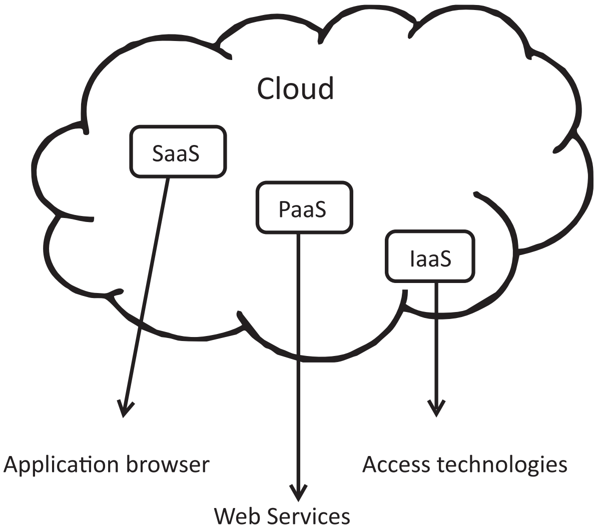

Fig. 2. Three basic cloud services

Radar Data cloud computing is implemented using a hybrid or private cloud. Hybrid clouds: types of clouds that have some of the characteristics of public clouds and some of private ones; so we can say that it is a cloud that has a mixture of properties and belongs to the public or a private cloud. Organizational responsibilities are often shared between cloud providers and developers. This cloud infrastructure is more agile in handling critical processes, as the user can store their important sensitive data in the private cloud and use the public cloud for normal routine services.

Overlapping of CLOUD and SOA functions. SOA has many benefits, but there are also some limitations. The very first problem that SOA faces is the use of an inefficient XML messaging format. To a certain extent, service-oriented architectures and cloud computing are related. A service-oriented architecture provides an architectural template. Whereas cloud computing offers highly scalable, dynamic resources and a flexible platform for a service-oriented architecture. SOA and cloud computing can exist simultaneously, supporting and balancing each other. One of the main advantages of cloud computing is the execution of the same request on multiple servers, resulting in low communication costs. Cloud computing and SOA are mutual. So, everything that happens in a service-oriented architectural environment will be sent as an event to the cloud. The event can be a data transaction or a user service request. The request can be used for any hardware resource or data. Therefore, it will be easy to add a new service to the program, and resources will become more scalable.

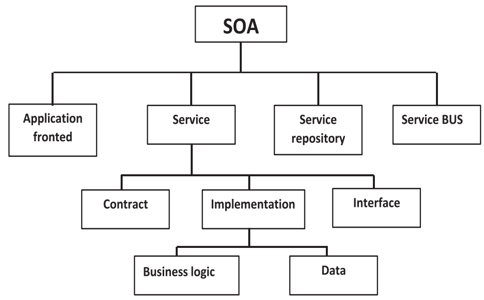

Fig. 3. SOA infrastructure

A solution to one of the limitations of SOA is proposed. It uses the XML format for messaging, which consumes a lot of network bandwidth. Combining Cloud with SOA, given the common points, will increase the reliability, availability of SOA and reduce the cost of its messaging. The use of Artificial Intelligence will lead to the development of a collaborative service and the development of new computing architectures [11].

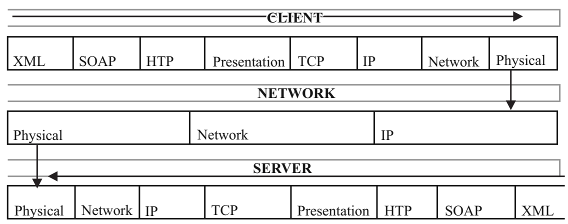

Fig. 4. SOA message transfer

3. Virtual machines for mobile radar computing in the cloud environment

Virtual machines have become an attractive approach for the radar computing platform, as applications running on virtual machines are isolated from each other to counteract the propagation of failures and security, but can simultaneously use physical machines. An important property of a virtual computing platform is how quickly it can respond when a request changes. That is, we can talk about the reaction of the virtual computing system to reassign resources as the flexibility of the platform to rebuild when the task changes. We are talking about a hosting service provider environment where the entire platform is under the control of a single administrative domain and applications often create clusters at the application level. In this work, resource reassignment mechanisms in these applications are explored from the point of view of flexibility and a new mechanism that uses the properties of the virtual utility of the computing platform.

This new mechanism uses haze virtual machines (VMs) that participate in application clusters but do not process client requests until they are enabled by system resource management. We're evaluating this, along with other mechanisms in the service computing testbed. The results show that this VM approach outperformed other approaches in terms of flexibility and allowed a new virtual machine to be added to an existing application. This approach, increasingly used in hosting platforms, has been confirmed by the observation that a virtual machine can quickly recover from a suspended state, provided that the suspended VM remains in memory. Unfortunately, servers typically implement a particular application in a software cluster; on the application servers on client machines, the cluster will be considered unhealthy and, as shown later, it will take a long time to re-enter the cluster when reactivated.

So, we propose to have a small margin of virtual machine resources on each physical machine that remains active but disconnected from the Internet. Since they are not available to clients, these virtual machines will not serve requests. Therefore, they are sometimes called ghosts because their existence is invisible to receiving and redirecting the content of client requests. Ghost VMs consume a central cluster management resource. But CPU time is negligible because only a small fraction of the ghost machines are kept on each physical machine. Remapping the source only constitutes a reconfiguration of the content selector.

By migrating a virtual machine from an overloaded host to a relatively free host, we can achieve better load balancing and use our resources more efficiently. There are several ways that a migration can be done.

1. Migration of a virtual machine can be achieved by stopping the virtual machine on one host and booting a previously saved mirrored virtual machine on a different host machine. In this case, it is the destination virtual machine that does not display the current state of the source VM.

2. The virtual machine of one host can be cloned to another. This method requires first suspending the VM on the output host, then copying it between hosts, and finally restoring it on the target host. The source VM state will be preserved, but this method requires moving the VM on machines in the critical path.

3. VM migration methods have been developed, where the source virtual machine is not stopped and the source of information is updated. The state of the virtual machine is transmitted to the target host when running. This method drastically reduces downtime during migration, but by itself does not prove the flexibility of reallocating resources. First, the target host will only be able to start doing its job after a delay similar to cloning technology. Secondly, the egress machine will not receive overload relief due to the operation of the outgoing VM until the migration is complete.

Promotion of the VM from ghost to active state can be combined. In most cases, most of the delay occurs just before the network switch reconfigures. This VM forms the basis for a flexible reallocation of resources. At the initial stage, an active list of virtual machines is formed. This is enough to serve the client's requests. If we migrate a virtual machine by stopping it, it is in the ghost list.

We have considered an important issue of virtualization of utility computing platforms – the need to quickly respond to changing requirements. We mean the response time of a platform as its flexibility. We are targeting a provider environment hosting utility where the entire management platform for a single administrative domain and application resides, often clustered at the program level. Ghost virtual machines are used that contain running application servers and are part of a cluster with active virtual machines. Considered a more advanced test client – tools for this environment, in addition to load generators, in more representative test programs to understand the time of rapid redeployment of resources in the application computing platform and a better understanding of the underlying resources.

The concept of ghost VMs introduces a hierarchy for a virtual machine on a radar computing platform: active ghost, paused, and not booted. This hierarchy provides more opportunities for global and local resources, control algorithms.

The new vision of mobile computing relieves mobile devices of serious resource constraints, allowing resource-intensive programs to use cloud computing without delays, jitter, congestion and failures in the global network. Through the efforts of many researchers, basic concepts, methods and mechanisms have been developed to provide the basis for this area of rapidly developing informatics [1]. Mobile computing enhances cognition through computationally intensive capabilities such as computer vision and graphics, machine learning, planning and decision making.

A cloud-based mobile computing strategy using a temporarily configured infrastructure as the mobile device moves with its user in the physical world. The crisp interactive response needed to seamlessly expand human cognition is easily achieved in the architecture due to the physical proximity of the cloud and network latency per clock. The use of a cloud package also makes it easier to meet the peak throughput of multiple users who interactively generate and receive radar imagery, high-definition video, and high-resolution images. Rapid infrastructure setup for various applications is becoming an important requirement.

Lack of resources is a major hurdle for many applications to seamlessly expand human cognition, as such programs typically require processing and energy far beyond what mobile hardware can provide.

The obvious solution to the problem of resource scarcity of mobile devices is the use of cloud computing. The mobile device can execute a resource-intensive program on a remote high-performance computer server or computing cluster and support user interaction with the thin client program over the Internet.

In wireless networks, a common power-saving technique is to turn on the mobile device's receiver only for short periods of time to receive and acknowledge packets buffered at the base station, which increases the average end-to-end packet delay as well as jitter. On the other hand, throughput is unlikely to be affected by these methods, since it is an aggregate, not an instantaneous figure. While throughput will continue to improve over time, latency is unlikely to decrease rapidly.

Wireless LAN throughput is typically two orders of magnitude greater than the wireless Internet bandwidth available to a mobile device. For instance the nominal throughput of the fastest currently available wireless LAN (802.11n) and HSPDA 400 Mbps wireless Internet and 2 respectively. From a user experience standpoint, the difference in transmission latency across these bandwidths can be very significant: 80 milliseconds instead of 16 seconds for a 4MB JPEG image, a huge difference for deep immersive programs.

Instead of relying on a remote «cloud», we could bridge the resource gap of a mobile device with the help of a nearby resource-rich cloud. Thus, we could meet the need for real-time interactive response through low latency, one-click, high bandwidth wireless access to the cloud packet.

Conclusion

The mobile device works as a thin client, and all significant computing takes place in the neighboring cloud. The physical proximity of this cloud is important: the end-to-end response time of applications running on it must be fast (a few milliseconds) and predictable. If there is no cloud nearby, the mobile device can flexibly switch to a fallback mode that includes a remote cloud or, in the worst case, only its own resources. Full functionality and performance may return later when the device detects a nearby cloud [12,13].

The virtual machine approach is more reliable than process migration or software virtualization. It is also less restrictive and more general than language-based virtualization that requires programs to be written in Java or C#. Another approach is called dynamic VM synthesis. The mobile device provides a small VM add-on to the cloud software infrastructure that already has the underlying VM from which the add-on was derived. Summing up, we can say that the topic of convergence of mobile applications in portable devices with cloud computing has become a revolution in the organization of distributed computing in the direction of closer interaction between hardware and software, the use of elements of artificial intelligence, and deep learning of a neural network.

References

1. Y. Simmhan, B. Plale, D. Gannon. A survey of data provenance techniques 2005. Computer Science Department, Indiana University, Bloomington IN 47405, 69
2. D. Gannon. Theoretical Problems in the Design of Tools to Aid in the Construction of Parallel Programs. 1989. Opportunities and Constraints of Parallel Computing, 39-48
3. A framework for collecting provenance in data-centric scientific workflows 2006. IEEE International Conference on Web Services (ICWS'06), 427-436
4. F. Bodin, P Beckman, D Gannon, S Narayana, SX Yang. Distributed pc++ basic ideas for an object parallel language, 1993. Scientific Programming 2 (3), 7-22.
5. C. Catlett, W.E. Allcock, P. Andrews, R. Aydt, R. Bair, N. Balac, B. Banister, ... Teragrid: Analysis of organization, system architecture, and middleware enabling new types of applications. 2008. High performance computing and grids in action, 225-249
6. Kevin Echton. (2009, June). Things about the Internet of Things In the real world, things matter more than ideas. http://www.rfidjournal.com/article/print/4986
7. Pankech Patel, Animech Pyatak, Tiago Teichery « On the way to the development of supplements for the Internet of speeches», New York, USA. 2011, p.5:1-5: 6. http://doi.acm.org/10.1145/209.190.2093195
8. Pol Zikopulos. Big Data IBM: What is Big Data., pat 1 and 2. 2012. http://www.youtube.com/watch?v=B27SpLOOh
9. E. Akerman and E. Huizu, «5 technologies, what formed net», Spectrum, IEEE, vol. 48, p. 40-45, June 2011 p.
10. K. H. Bennett, V. T. Rajlich and N. Wilde, "Software Evolution and the Staged Model of the Software Lifecycle", Advances in Computers, Volume 56, Academic Press, pp. 1 – 54, 2002.
11. Gaoyun Chen, Jun Lu and Jian Huang, Zexu Wu, "SaaAS – The Mobile Agent based Service for Cloud Computing in Internet Environment. Sixth International Conference on Natural Computation, ICNC 2010, pp. 2935-2939, IEEE, Yantai, Shandong, China, ISBN: 978-1-4244-5958-2, 2010.
12. Nancy Wilkins-Diehr, Chaitan Baru, Dennis Gannon, Kate Keahey, John McGee, Marlon Pierce, Rich Wolski, Wenjun Wu: "Harnessing clouds and software services for science". Journal "Cloud Computing and Software Services", 2010, p.17.
13. M. P. Papazoglou, P. Traverso, S. Dustdar and F. Leymann, "Service-Oriented Computing state of the art and research.

Received: 14.04.2023

About the authors:

Mykola Kosovets
Leading Constructor,
Number of scientific publications in Ukrainian publications -56,
Number of scientific publications in foreign publications -17,
Index Girsh – 5
https://orcid.org/0000-0001- 8443-7805
Scopus Author ID: 5644007500

Lilia Tovstenko
Leading Software Engineer,
Number of scientific publications in Ukrainian publications -24,
Number of scientific publications abroad -8.
Index Girsh -7.

Place of work:

Mykola Kosovets
SPE "Quantor", Chief
03057, c. Kyiv-57, str. E. Potye, 8-A

Ph.: (380)66-2554143
E-mail: quantor.nik@gmail.com

Lilia Tovstenko
Glouchkov Institute of Cybernetics of
of the National Academy of Science of
Ukraine, 03187, Kyiv-187, Academician
Glouchkov Avenue, 40.
Ph.: (380)67-7774010
E-mail: 115lili@incyb.kiev.ua