Virtualization technologies.

Virtualization technology is one of the most significant technological innovations underlying cloud computing.

Virtualization is the provision of a set of computing resources (their logical combination), which is abstracted from the hardware implementation and at the same time provides logical isolation of computing processes that are performed on one physical resource.

Basic concepts of virtualization technology

A virtual machine (VM) is an isolated software container that works with its own OS and applications like a computer. A VM acts like a regular physical computer and includes virtual RAM, a hard disk, and a network adapter. Vms are independent of the physical equipment on which they work.

The main features of VM are:

- compatibility (vms are compatible with all standard computers, vms run their own OS and run their own applications);

- isolation (vms are isolated from each other);

- encapsulation (vms completely encapsulate the computing environment).

A host OS is an OS installed on real hardware, on which virtualization software is installed as a normal application.

A virtual machine emulator is software that is installed on a host OS and consists of a VM monitor and a graphical shell.

The VM monitor is a program that ensures all interactions between virtual and real equipment, supporting the operation of one or more created virtual machines and installed guest operating systems. The graphical shell provides user interaction with the virtual machine application, allowing you to customize the created virtual machines according to your needs and manage its operation.

A guest operating system is an operating system installed on a created virtual machine. Windows, Linux, etc. Can be used as guest operating systems.

When using virtualization technology, a hierarchical structure of interaction between virtual computers and real hardware is obtained. At the bottom of this hierarchy is the real hardware, which is shared between the host operating system and the virtual machine emulator.

The host operating system and the emulator share the resources of a real computer and constitute the second level of the hierarchy. The host OS manages the applications running on it and distributes real computer resources between them.

A virtual machine emulator manages virtual machines with guest operating systems installed on them, distributing the resources of a real computer between them in such a way that users get the impression of working on real equipment.

All existing VM monitors can be divided into four types that use:
- hardware,
- hardware and software,
- software,
- domain virtualization.

This is a conditional division, because most VM monitors use both software and hardware virtualization.

Domain virtualization is based on the logical distribution of resources into separate parts (domains). It is mainly used in mainframes. This type of virtualization appeared first and was used to distribute the resources of large computers between individual users.

VM monitors that use hardware and software virtualization technology. Some instructions are executed on the processor itself, and some are emulated.

There are three types of software instruction emulation:
- full emulation of instructions;
- selective emulation of instructions;
- API emulation.

When using full emulation, the instructions will be interpreted and converted into instructions. Perceived by the real processor. In this case, it becomes possible to create virtual machines that simulate the operation of equipment that is not compatible in terms of architecture with a real computer. For example, you can run a virtual machine simulating the operation of a RISC-architecture processor on a real computer with a CISC-architecture processor. This is possible due to the fact that the emulation is carried out at the level of basic arithmetic and logical instructions, which are present in almost any processor.

The interpretation of each instruction leads to a significant consumption of real computer resources and reduces the speed of applications running in the guest operating system. Modern servers and personal computers have more and more performance. Therefore, virtualization, using the interpretation of instructions, is gaining popularity. Representatives of this class of virtual machines are: Microsoft Virtual PC, Bochs, Simics, etc.

All running programs interact with the equipment using the API interface. Therefore, it is possible to intercept the flow of programs running under the control of the guest operating system to the API, convert it to the form accepted by the host operating system, and relay the received result to the API of the host operating system.

The result of the execution of the request by the host OS will be converted to a form perceived by the guest operating system and transmitted to the program that issued the request.

If the guest and host operating systems are compatible in terms of their API, then it is not necessary to convert traffic, it is enough to redirect them.

However, such a virtualization system has disadvantages:

Not all software can be run in a virtual machine with this virtualization principle, as undocumented apis and direct-to-hardware routing are often used.

Operating systems are actively developing, adjustments are made to API and new features are added. Therefore, API emulators quickly become obsolete and higher costs are required for their modernization.

API emulators are tied to specific operating systems. This narrows the circle of their use and consumer properties.
However, using ARI emulation avoids significant performance losses.
As an example of virtual machines using AR emulation. You can bring such products as:
Wine (Wine Is Not an Emulator), used to run Dos and Windows - applications under the control of the Linux operating system.
UML (User Mode Linux), which is built into the Linux kernel and which allows you to run several copies of the operating system on one computer.

Security in virtual clouds.

Virtualization is the transition from the technology behind server and data center consolidation to the core components to create a flexible, on-demand infrastructure. When implementing virtualization in any environment, there are many tasks that need to be solved.

A typical hypervisor is more specialized than a general-purpose operating system and less open to attack. The hypervisor changes infrequently and does not run applications from third-party developers. A vulnerable guest OS does not have direct access to the hypervisor. The hypervisor is transparent to the network graph. Unless you count the inbound and outbound traffic of the dedicated hypervisor management interface.

Another security risk in virtualization is that how resources such as local storage relative to virtual machines are allocated and freed. During the deployment and operation of the virtual machine, data is written to physical memory. If the memory is not freed before passing it on to the next virtual machine, there is a possibility of data compromise.

Another risk associated with virtualization is the possibility of undetected network attacks between virtual machines located on the same physical server.

One practical way to manage traffic between virtual machines is to use virtual LAN to isolate virtual machine traffic. Belonging to different customers. But for this approach to be effective, support for virtual LAN needs to be extended beyond the core switching infrastructure to the physical servers that host the guest systems. Such support is used almost everywhere when using virtualization.

The next challenge is to scale the functionality of virtual loms beyond existing boundaries to support large-scale clouds.

Types of virtualization:

- server virtualization:

- virtualization at the level of operating systems:

- network virtualization;

- application virtualization;

- virtualization of CAD workplaces:

- storage virtualization.

Server virtualization.

The architecture of modern x86 servers provides for the execution of only one OS on the server. Such structural limitations can be overcome with the help of virtualization of x86 servers. This technology abstracts the operating system and applications from the physical hardware layer, making the server environment less complex and more adaptable and cost-effective. Thanks to virtualization, several operating systems can be run on one physical server as virtual machines, each of which has access to the server's computing resources.

Server virtualization technology allows you to run several logical units on one server - virtual machines that fully reproduce the work of independent physical servers. This makes it possible to place dozens of independent operating systems and corporate applications on one piece of equipment, using the IT infrastructure more efficiently.

Virtualization can be described as the abstraction of physical system resources, which allows you to create many logical partitions in which different operating systems will work (simultaneously on one physical server). Each partition (also called a virtual machine) is a software environment that provides resources (using hardware or device emulation). You can install an operating system on top of it, as well as one or more applications.

The beginning of commercial virtualization technology was laid by IBM in the mid-60s of the XX century, when the System / 360 Model 67 computer allowed to support several simultaneously working guest virtual machines (in each of which the operating system could work for one user). IBM achieved this by developing two separate operating systems - the Virtual Machine (VM) and the Conversational Monitor System (CMS), referred to as VM / CMS. A VM created and managed virtual machines, and a CMS (single-user operating system) ran inside a virtual machine, providing access to system resources for each user. Currently, IBM continues to develop and promote VM (after rebranding it is called z / VM), which can now even execute itself inside a virtual machine.

In recent decades, research on virtualization technology (and the development of related products) has taken off with an emphasis on the use of x86 platforms (32 and 64-bit). AMD and Intel released versions of the x86 family of processors with new instructions and extensions that were designed specifically for hardware support for virtualization. Though. Although the details of these implementations differ, AMD Virtualization Technology (AMD-V) and Intel Virtualization Technology (VT) have provided such hardware virtualization capabilities that can be used by software vendors to simplify the code of their virtualization software products and to evolve the virtualization solution architecture itself.

When implementing VMM, three possible methods are used to create an interface between virtual machines and virtualized system resources: full virtualization. Own virtualization and paravirtualization.

Full virtualization.

When using this method, the VMM monitor (to abstract the virtual machine from the real hardware) creates and maintains a complete virtual system. This approach allows you to run the operating system in a virtual machine without any modifications to it.

The advantage of full virtualization and the full physical hardware and virtual machine solution approach is the ability to easily migrate virtual machines between servers with different physical configurations.

This flexibility is achieved at the cost of losing performance due to the overhead of maintaining virtual machine states and binary broadcast delays.

Own virtualization.

Such virtualization depends on the architecture of the intended processor (for example, such as in the AMD-V and Intel VT series of processors). These processors have new execution modes, instructions and data structures in their hardware, which are designed to reduce the complexity of the VMM.

With your own virtualization of the VMM monitor, it is not necessary to maintain the characteristics of the resources of the virtual machine and its state in the software. Just like with full virtualization, operating systems can run inside virtual machines without modifying them. Hyper-V uses this method to run legacy operating systems.

This type of implementation has many advantages - from simplifying the VMM architecture to significantly increasing productivity (as a result of reducing software overhead).

Paravirtualization.

Paravirtualization was developed as an alternative to using binary translation when processing non-virtualizable instructions of the x8b processor. This approach requires modification of guest operating systems.

A strict implementation of paravirtualization gives increased performance on standard x8b hardware (due to the elimination of expensive operations in full virtualization and binary translation}. However, this is done at the expense of the lack of support for unmodified guest operating systems and the migration of virtual machines back to the physical server. Given these limitations, software products based on paravirtual implementations also use hardware virtualization (to run unmodified operating systems). This approach allows for broader support, including legacy operating systems, and allows for newer operating systems to be upgraded (taking advantage of improvements and performance improvements that can be obtained from paravirtualization).

Server virtualization solutions:

- **vmware vsphere - a platform for server virtualization that provides maximum levels of availability and responsiveness for all applications and services.**
- **vmware vcenter - a single management console for the internal virtual infrastructure of data centers**
- **vmware vcloud Director - a tool for managing internal and external "clouds" of the organization.**
- **Citrix xenserver - Enterprise-level solution for server virtualization.**
- **Microsoft Server Hyper-V - virtualization system in the Microsoft server environment. Which provides organizations with wide opportunities for managing and scaling IT resources.**
- **Microsoft System Center Virtual Machine Manager - a solution designed to manage physical and virtual machines.**

Virtualization at the level of operating systems.

Virtualization at the operating system level is based on the idea of the host operating system supporting several isolated partitions (or virtual environments, virtual environment (VE). Virtualization is achieved by multiplexing access to the kernel (with system security against virtual environments).

Virtualization at the level of the OS kernel has. Refers to the use of one core of the host OS to create independent parallel operating environments. Only its own network and hardware environment is created for the guest software. This option is used in Virtuozzo (for Linux and Windows). Openvz (a free version of Virtuozzo) and Solaris Containers.

Network virtualization.

Network virtualization is a complete reproduction of a physical network using a software method. Virtual networks are similar to physical networks in terms of reliability and capabilities. However, they have many additional operational benefits, such as hardware independence, rapid initialization, non-disruptive deployment, automated maintenance, and support for both modern and legacy applications.

Virtualized networks connect workloads to logical network devices and services, such as logical ports, switches, routers, firewalls, load balancers, vpns, and more. Applications in virtual networks work exactly the same as in physical ones.

For example, vmware network virtualization software supports the creation of adaptive, scalable network environments with high operational efficiency. Vmware solutions not only implement continuous monitoring and control of quality of service and security, but also help to perform initialization, troubleshooting and shutdown much faster than using physical infrastructures.

With the ability to deploy full-fledged, software and virtual mobile networks for virtual machines on any network equipment with IP protocol support, vmware NSX takes the place of the best platform for virtualization of networks and security systems.

Application virtualization is a technology that aims to separate and isolate applications on the client side (running under the local operating system). Applications are isolated in a virtual environment between the operating system and the application stack. The virtual environment is loaded into the application, isolated from other applications and the operating system, and also prevents the application from modifying local resources (such as files and registry settings). Applications can read information from the local system registry and files, but writable versions of these resources are maintained inside the virtual environment. This technology allows you to use several incompatible applications simultaneously on one computer, or rather in the same operating system. Application virtualization allows users to run the same preconfigured applications or a group of applications from the server. At the same time, applications will work independently of each other, without making any changes to the operating system. This option of virtualization is used in Sim Java Virtual Machine, Microsoft Application Virtualization (formerly Softgiid), Thiiistall (in early 2008 p became a part of vmware), Symantec / Altiris.

Product-based application virtualization and terminal access solutions:

• Citrix heparr - the actual standard for delivering any Windows applications with minimal costs to any part of the world. Citrix heparr provides secure terminal access to corporate resources and their centralized management.

• Citrix netscaler - a solution for optimization of the web application delivery process.

• Citrix Branch Repeater - a solution for accelerating and optimizing the delivery of applications and desktops.

• vmware Zimbra - a system for automating the joint activities of work groups (mail, shared documents).

• Microsoft App-V - solution for virtualization and delivery of applications, quick update, launch of incompatible applications, rights management.

Virtualization of workplaces.

RM virtualization implies emulation of the user interface, that is, the user sees the application and works with it on his terminal.

Application of the technology of virtualization of employees' workplaces based on the infrastructure of virtual pcs - Virtual Desktop Infrastructure (VDI). VDI allows you to separate the software intended for the user from the hardware part, as well as access to client applications through terminal devices.

Virtual desktop

This virtualization is intended to change the management of computer resources of end users (by virtualizing desktop computers and consolidating them on centralized servers).

Desktop virtualization is implemented using the Virtual Desktop Infrastructure. This term refers to a combination of hardware, virtualization software, and management tools.

Various solutions for organizing a remote desktop.

• VDI from Amazon (AWS)

Amazon Workspace is remote desktops in the Amazon AWS cloud with pre-installed office application packages with the ability to integrate the service with a local directory service.

• daas service (Remote Desktop)

The daas service is fast access to the desktop of users from any point, from any device. Users get the opportunity to work more efficiently, quickly accessing their data at any time.

Virtualization of CAD workplaces.

CAD workstation virtualization technology, built on the basis of Citrix solutions and NVIDIA GRID technology.

NVIDIA GRID technology is a solution for GPU virtualization, remote access and session management that allows multiple users to simultaneously work on graphically intensive applications using shared GPU resources. The technology is designed to solve the problem of virtualization and remote access to workplaces in such areas as design automation (CAD), construction information management (BIM), product life cycle management (PLM), automation of activities of credit and financial institutions, work with backup systems and transfer of images in the field of health care, photo and video editing.

Thanks to this, as well as the use of Citrix xenserver GPU pass-through and Citrix xendesktop HDX 3D technologies, it became possible to run 3D applications on the server side and provide access to them in terminal mode with full support for professional NVIDIA GRID graphics.

Advantages:

• Minimizing the risk of losing corporate data due to their absence at the local workplace and the possibility of direct access to them.

• Possibility of remote work of employees.

Storage virtualization.

The goal of storage virtualization as one of the software storage components is to increase productivity without purchasing additional data storage equipment.

Virtualized storage must support fast initialization to ensure that high-performance, efficient storage can be deployed at the same speed as a VM. The storage management model should focus on the needs of vms and ensure the convenience of work for administrators of virtual environments whose task is to manage storage resources. For convenient work with virtualized storage, a high degree of integration with the hypervisor platform is required from them.

Vmware storage virtualization is not a combination of capabilities that forms an abstraction layer for physical storage resources and supports their addressing, optimization, and administration in a virtual environment.

Storage virtualization technologies provide an efficient way to manage storage resources in a virtual infrastructure and realize the following advantages:

- **Significant increase in the efficiency and flexibility of storage use:**
- **Simplifying the installation of OS patches and reducing driver requirements, eliminating dependence on the storage topology;**
- **Increasing the uptime of applications and simplifying standard processes:**
- **Use and addition of the existing storage infrastructure.**

Analysis of application of fpga (field-programmable gate array) technology included in the cloud infrastructure.

Cloud computing technologies are actively developing and allow an ordinary user to gain access to various computing resources, including distributed database storage, in a short period of time. Thus, they are the main and indispensable technology for providing on-demand access over the global network to computing software and hardware resources and data stores presented in the form of services.

FPGA-class programmable logic is attractive in many computing areas due to the ability to provide performance close to ASI (Application Specific Integrated Circuit) technology, achieve high throughput, predictable latency with greater design flexibility, and support low power consumption. From the point of view of cloud computing, FPGA technology is interesting in several directions.

First, many resource-intensive tasks related to digital data processing (audio data, video images, photographs, etc.) Require high-quality performance and can be effectively solved using FPGA technology.
Secondly, the FPGA element base can be used directly as a platform on which some part (most often hardware) of the cloud infrastructure (virtual machines, network switches, etc.) Is deployed. This will be especially effective where the speed of reaction to changes (reconfiguration) is required. In connection with the growing interest and high demand for the solution of this kind of tasks, it is necessary to consider the issues related to the integration of FPGA technology into the Cloud infrastructure.

Analysis of options for interaction of FPGA technology as part of cloud services.

An analysis of options for the joint use of FPGA and Cloud technologies allows us to say that the following forms of interaction between FPGA and cloud computing are possible:

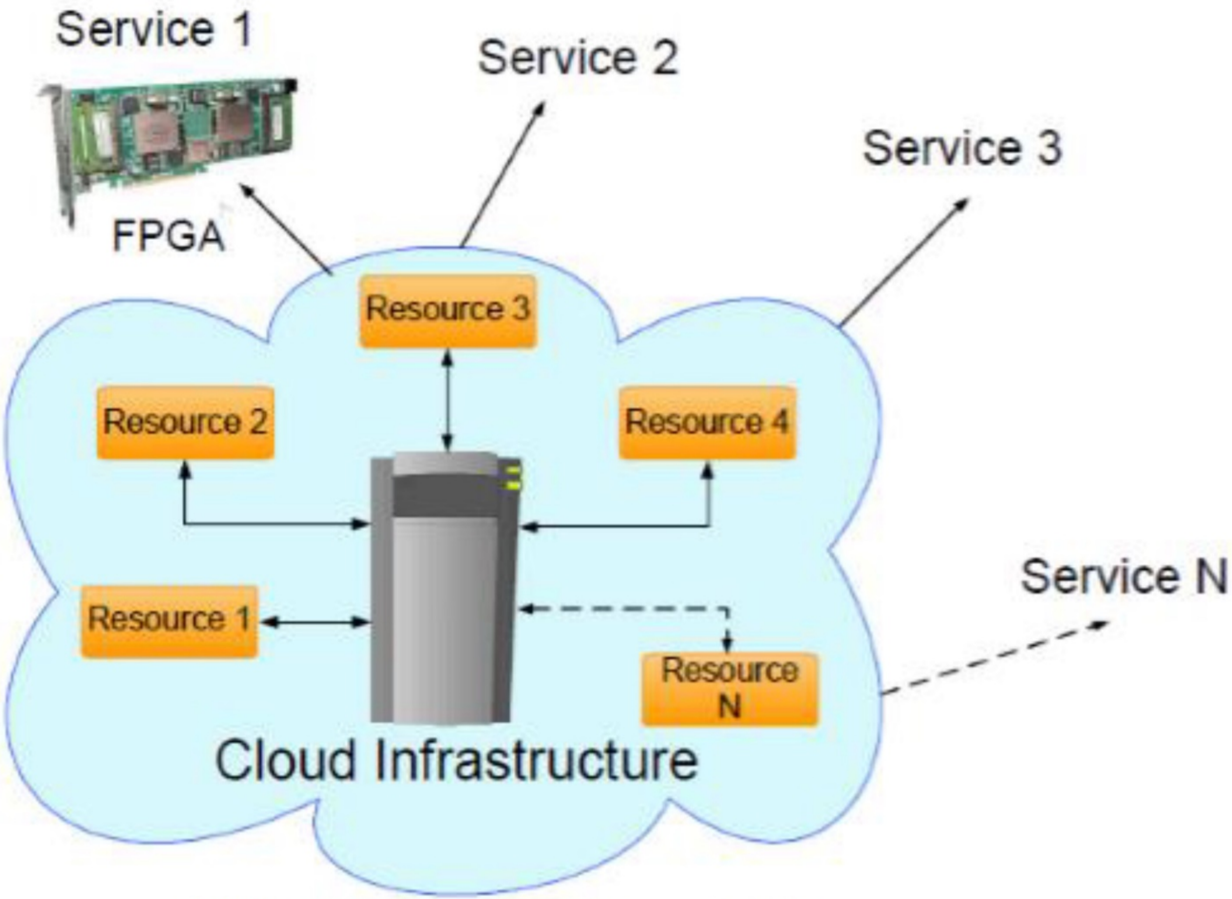
- **Cloud computing for FPGA;**
- **FPGA for cloud computing;**
- **FPGA as a "flexible" service;**
- **FPGA and cloud computing competition.**

In the first case, cloud computing is used to support the development, verification and certification processes of FPGA projects and systems on FPGA - Cloud Computing for FPGA. For example, an FPGA project can enter the cloud infrastructure as a test sample, and all processes related to the formation of test data and the subsequent fixation of simulation results are performed using cloud services. The second option, when the FPGA platform is used for cloud computing (FPGA for Cloud Computing), involves the development and implementation of additional services implementing the FPGA as a Service (faas) approach. In this case, the principle of hardware acceleration of various types of calculations is used, from tasks in the field of cryptography to digital data processing - streaming video, audio, image processing (cartography, tomography, etc.). Thus, similar hardware accelerators can be used in data centers to provide high-performance computing services.

The approach of FPGA as a "flexible" service (FPGA for flexible Cloud Computing) assumes that the FPGA platform is used as a reconfigurable service. That is, the user is given the opportunity to implement his own design solutions, and to change the "flashing" (project architecture) of FPGA accelerators in accordance with the requirements of the solved task to achieve greater productivity, compared to solutions based on unchanged architectures.

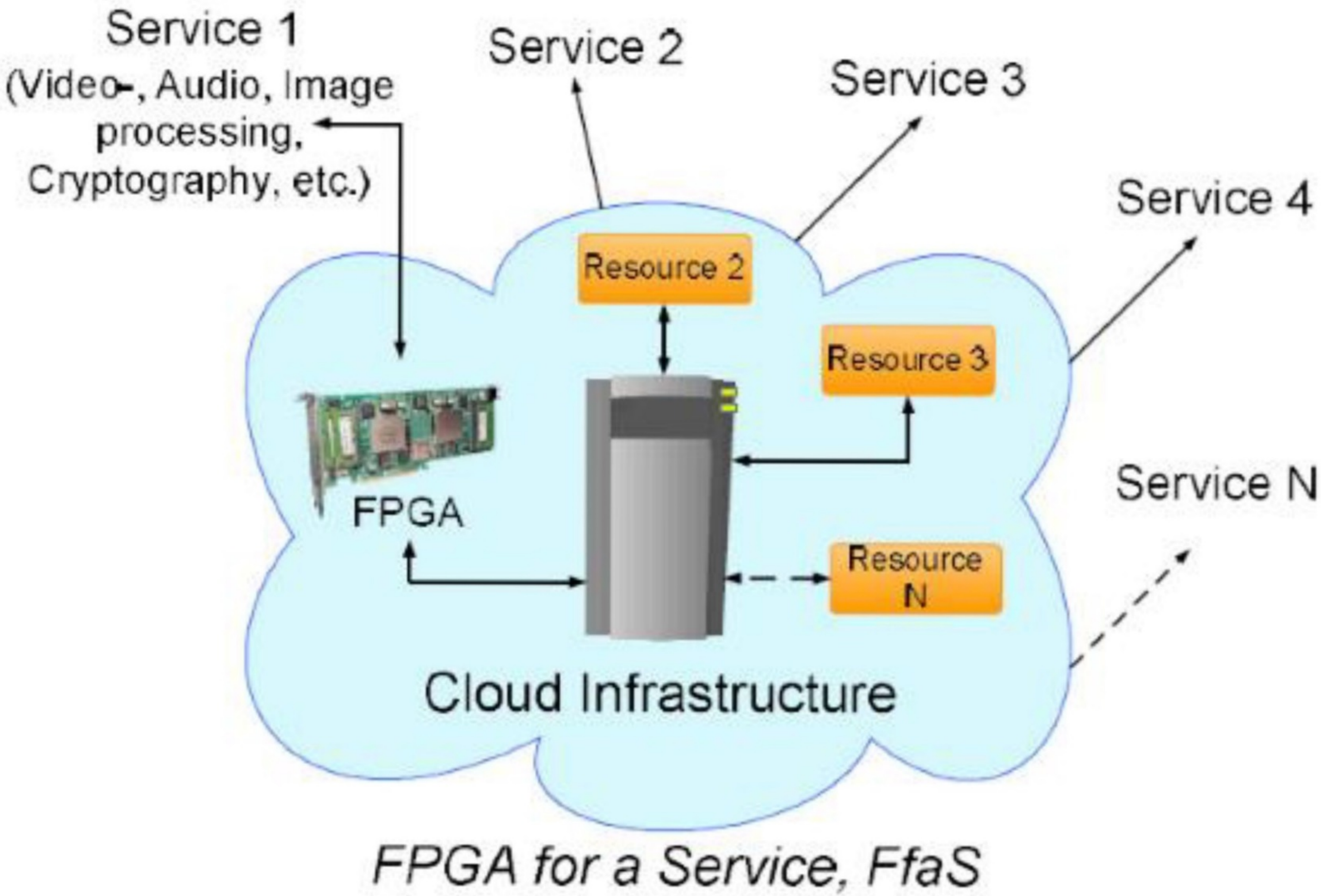
Application of FPGA in basic models of cloud services

Options for using FPGA technology in the basic models of cloud services and the effect of their implementation are presented in the table. They differ in accessibility for the end user. So, for example, the use of FPGA technology as an accelerator in Sa type services is completely "transparent" for the user. In a Raa-type service, the user can upload his project to the FPGA, but the process of resource allocation and reconfiguration is managed directly by the server. Finally, in services of the Iaa type, the user has the opportunity, in addition to downloading the project, to manage the allocation and reconfiguration of the FPGA in the same way as other, "traditional" cloud resources. Because the basic models of cloud services are implemented according to the principle of nesting, that is, in addition to the listed individual options for sharing FPGA and cloud technologies, it is possible to combine them within one service. In other words, all cloud models on which a specific service is built can use FPGA capabilities independently of each other.



FPGA as a Service, FaaS

Variant of interaction of FPGA technology and Cloud infrastructure "FPGA as a Service".



Variant of interaction of FPGA technology and Cloud infrastructure "FPGA for Service".

The need to provide FPGA as a resource.

Given the growing popularity of cloud technologies and fpgas, it is safe to assume that the need to host FPGA projects in a cloud environment will grow. Previously, it was possible to observe a similar increase in the popularity of the use of graphics processors in the Cloud environment, the resources of which are widely used even today. With the help of such integration, it was possible to increase the productivity of cloud data centers (data centers). However, the use of graphics processors is not an energy-efficient solution. One more disadvantage of graphics processors is their incompatibility with multi-user access modes and, therefore, difficulty in task management.

Unlike gpus, fpgas consume less power while providing significant performance gains. Today, leading IT manufacturers (IBM, Intel, Microsoft) have offered technologies for integrating fpgas into cloud data centers. The conducted studies show a significant increase in productivity with a slight increase in energy consumption. However, all proposed solutions do not assume direct use of FPGA resources by users of cloud services.

Introduction to distributed computing systems.

There is currently no formal definition of a distributed system (PC).

One of the definitions:

"A distributed system is a set of independent computers connected by communication channels, which from the point of view of the user of some software look like a single entity."

This definition captures two essential points specific to PC:

- **Node autonomy:**
- **Presentation of the system by the user as a single structure.**

At the same time, the main connecting link of distributed systems is software.

Classification of distributed systems:

1) **By sizes and methods of administration:**
   a) **Clusters, tens of pcs, manual administration;**
   b) **PC of corporate level, hundreds of pcs united by certain rules, direct administrative measures;**
   c) **Global pcs, millions of pcs in the world are connected in a global network, administrative software is built-in.**
2) **By functionality:**
   a) **Computer systems - the main computer resource is processor power;**
   b) **Information systems - the main computer resource is the volume of data memory; such systems are considered as data stores.**

A distributed computing system (DSC) is a hardware and software complex focused on solving certain tasks. On the one hand, each computing node is an autonomous element. On the other hand, the software component of DSC should provide users with the visibility of working with a single computing system.

In this regard, the following important characteristics of DSC are distinguished:

1. The possibility of working with different types of devices:

- with different device providers:

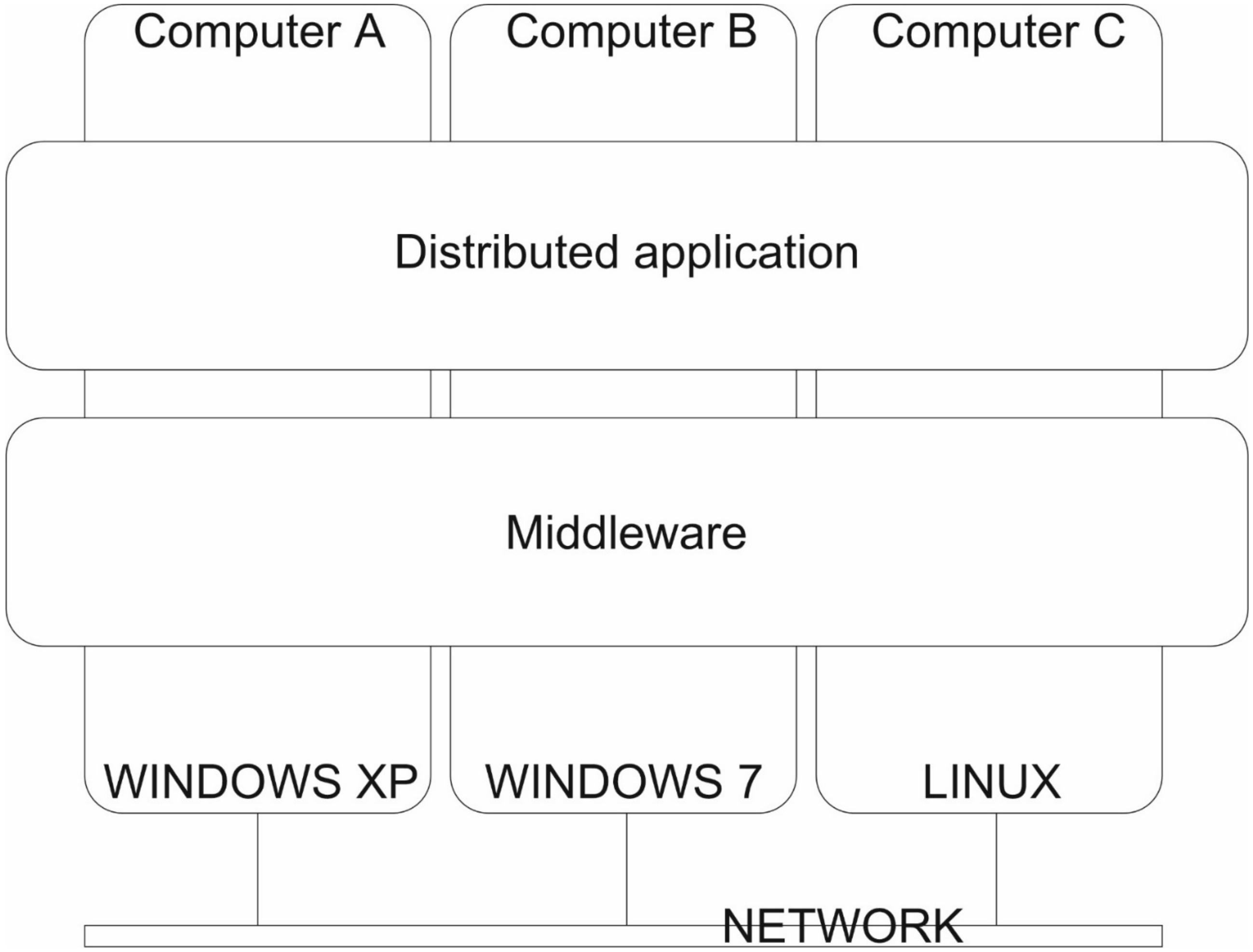
- with different operating systems,

- with different hardware platforms. Computing environments consisting of many computing systems based on various hardware and software platforms are called heterogeneous.

2. The possibility of easy expansion and scaling;

3. Permanent (constant) availability of resources (even if some elements of DSC may be unavailable for some time);

4. Hiding communication features from users.


Computer A
Computer B
Computer C
Distributed application
Middleware
WINDOWS XP
WINDOWS 7
LINUX
NETWORK

Their functionality is based on the lower layer - middleware that interacts with the system software and the network layer to ensure the transparency of applications in DSC.

In order for DSC to be presented to the user as a single system, the following types of transparency are used in DSC:

- transparent access to resources - the difference in data presentation and methods of access to ROS resources should be hidden from users;

- transparent location of resources - the physical location of the required resource should not be essential for the user;

- replication - hiding from the user that in reality there is more than one copy of the used resources;

- parallel access - the possibility of joint (simultaneous) use of the same resource by different users independently of each other. At the same time, the fact of sharing the resource must remain hidden from the user;

- transparency of failures - failure (disconnection) of any DSC resources should not affect the work of the user and his application.

DSC terminology.

Resource - any software or hardware entity presented or used in a distributed network (for example, computer, storage device, file, communication channel, service, etc.)

A node is any hardware device in a distributed computing system.

A server is a provider of information in DSC (for example, a web server).

The client is a consumer of information in DSC (for example, a web browser).

Banket is a node that combines both the client and server parts (that is, both the supplier and the consumer of information at the same time).

A service is a network entity that provides certain functionality (for example, a web server can provide a file transfer service using the HTTP protocol). Within the framework of one node, slightly different services may be provided.

Classification of dsc		
By the method of identifying resources	**On the availability of resources**	**By the method of interaction of resources**
Centralized		
A request for search is sent to some center, a server (for example, a DNS service), which returns its IP address based on the name of the site	**There is one server that provides a resource or service**	**Interaction between nodes is performed through the center. Server, one node cannot address another directly**
Decentralized		
A search request is sent to all nodes	**Each node plays the role of a client and a server that can provide resources and services similar to other devices in this network**	**Direct interaction between DSC nodes, because each node plays the role of a client and a server**

The history of the development of distributed computing (DC).

The first generation of DC. The beginning of the 1990s

Union of capacities of supercomputers in order to provide public resources for a certain number of high-performance applications

(FAFNER, I-WAY)

Second generation DC, late 1990s

Allocation of resources among dynamic choices of users and provision of homogeneous interaction of high-performance heterogeneous DSC

(Grid, object-oriented systems, distributed object systems)

Third generation DC, early 2000s

Development of virtual cooperation and virtual organizations.

Use of service-oriented methods of providing access to VR.

Automation of management, error processing, recovery of public process

(P2P, SOA, agent networks, cloud computing).

The FAFNER project (Factoring via Network-Enabled Recursion) was created in 1995 to solve the problem of factoring large numbers based on the capabilities of geographically distributed computing systems. Finding prime factors of large numbers allows you to decipher data encrypted on the basis of the RSA algorithm (abbreviation from the first letters of the surnames of the developers of this method: Rivest, Shamir and Adleman).

Specifics:

- **implementation of NFS - Network File System, allowed even small workstations (with 4 MB of RAM) to perform useful work, calculating their small fragment of the task:**
- **anonymous registration of participants;**
- **hierarchical structure of web servers**

I-WAY - Information Wide Area Year - an experimental high-performance network, designed in early 1995, with the aim of unifying high-speed networks that existed at that time. As part of the project, a hardware infrastructure was built, which was used to access I-WAY network resources. It consisted of basic workstations running the UNIX operating system on which special software was installed (I-POP). The system I-POP assumed the functions of a gateway to the resources of the I-WAY network. Each such server supported standard procedures for authentication, resource reservation, process creation, and communication. The I-WAY project was used to solve the following tasks: super computer calculations; access to extracted resources; virtual reality tasks.

Peer-to-peer networks (P2P) began to appear in 1999-2000. In such networks, computers exchange resources directly with each other, without using a central server. The use of such technology simplifies the scalability of the network and increases its resilience, since the failure of any computing node does not lead to a complete shutdown of the network. Disadvantages of P2P include:

1. Increasing performance requirements for each computer included in such a network:

2. A low degree of security of machines participating in the P2P network due to the provision of open access to their resources;

3. The need to overcome the possible heterogeneity of hardware and software of its potential participants.

4. Labor-intensive and resource-intensive search for available resources without the use of a centralized control point.

Examples of P2P include such projects as Gnutella, bittorrent, Napster, and Skype.

Agent systems began to appear in the mid-1990s in order to build large-scale computer networks adapted to function in a dynamically changing environment. A software agent is an autonomous process capable of responding to the runtime environment and causing changes to the runtime environment, possibly in cooperation with users or other agents. The main operating principles of agent networks are: - autonomy - agents function autonomously, without the possibility of external intervention in their internal state; - social behavior - agents interact with each other using a certain language (Agent Communication Languages, acls); - activity - agents interact with the environment, receiving certain signals and responding to them;

- pro-activity - agents act purposefully.

One of the best-known examples is the interaction architecture of FIPA (Foundation for Intelligent Physical Agents), which standardizes methods of interaction between agents and agent systems.