4. Kolpakov V.M. Theory and practice of managerial decision-making: textbook. allowance. K.: IAPM, 2004.504 s.
5. Kostin V.N.: Optimization tasks of the electric power industry: Textbook. allowance. - SPb.: SZTU, 2003 - 120 s.
6. Leblanc R. Board of Directors - an inside look. The principles of formation, management, efficiency analysis / trans. from English M.: Alpina Business Books, 2006.267 s.
7. Simonova N.F. Strategic decision-making models: textbook. allowance / Perm. state un-t Perm, 2003.
8. Siddikov I.Kh., Porubay O.V., Lazareva M.V., Abdulhamidov A.A. "Trends in the development of intelligent systems in making managerial decisions in Uzbekistan" Resource: http://7universum.com/ru/tech/archive/item/8827
9. Siddikov I.Kh. The construction of models and algorithms for the intellectualization of control processes of dynamic objects in the face of uncertainty. “MERIYUS”, Toshkent, 2016. - 88 p.
10. Teacher Yu.G. Development of management decisions: a textbook. M.: Unity-Dana, 2008.
11. Filinov N.B. Development and adoption of management decisions: textbook. allowance. M.: INFRA-M, 2009.308 p.
12. Chernorutsky I.G. Methods of decision making: textbook. allowance. SPb.: BHV-Petersburg, 2005.
13. Shemetov P.V. Management decisions: technology, methods and tools: textbook. allowance. M.: Omega-L, 2013.398 s.
14. Yarez O.B. Methods of making management decisions: textbook. allowance / Vladimir. state un-t Vladimir, 2011.
15. Kahn H. Thinking about unthinkable. N. Y., 1984.

ORGANIZATION OF RESOURCES OF DISTRIBUTED COMPUTING SYSTEMS FOR SIMULATION AND RESEARCH OF COMPLEX OBJECTS

T.F.Bekmuratov[1], R.K.Bazarov[2]

[1] Scientific and innovative center of information and communication technologies at the Tashkent University of Information Technologies named after Muhammad al-Khworazmi
Address: Boz-2 district,17a, 100025, Tashkent city, Republic of Uzbekistan
academician, doctor of science, E-mail:[1]bek.tulkun@yandex.ru
[2] Tashkent University of Information Technologies, Uzbekistan
Address: Amir Temur street, 108, 100200, Tashkent city, Republic of Uzbekistan
doctoral candidate, e-mail: [2]rustam.bazarov@gmail.com

Abstract: *Computing systems designed to study complex objects have evolved from single-processor computers to fog infrastructures. The article describes the composition, hardware and software features of modern distributed grid-, cloud- and fog computing systems. It is shown how these features affect the study time of complex objects. This time is a measure of computing system effectiveness. Unfortunately, there is still no assessment of modern computing system architectures in terms of that measure. The article fills this deficiency. Computing systems should have the properties of complex objects under study - the main conclusion of the publication.*

Keywords: *time to study of complex object, distributed computing system, grid-, cloud-, fog-computing systems.*

Introduction

Any object under research have a certain level of organizing and complexity. The more the properties of an object as a whole differ from the sum of its properties, the higher it’s organizing level. There are simple and complex objects. Complex objects are distinguished by their emergence, i.e. the irreducibility of the system's properties to the sum of its constituent elements. In addition to the fundamental property of emergence, a complex object has other properties: multi-dimensionality, multi-criteria, multiplicity in modeling and studying its properties, the variety of elements and forms of connections between them, the dynamic nature of changes in composition [1].

The behavior of any CO can be described by a stochastic process $\left(\| Z \|_t, t \geq 0\right)$, where $\| Z \|$ is the norm of Z, a vector of states of a countable set of CO subsystems. Changing the positive feedback between CO elements to negative and vice versa, destabilizes and, accordingly, stabilizes CO behavior [2].

During periods of instability, when the CO shows its unpredictability and complexity, it is important to collect as much data about the object as possible and process it as quickly as possible.

Since complex unpredictably changes the structure and composition of its elements, its behavior during observation time T is described in general by a sequence of functional operators:

$$Y(t) = F < R(t) > (t, X_0, ..., X_{n-1}), t \in [0,T]$$

Where F – structure of the model; $Y(t)$ – intermediate result of a computing experiment at the time-stamp t. $X_0,\ldots,X_{n-1}$ – input sets at the previous time-stamps. $R(t)$ - requirements set for the computing system (CS), where experiment is running on: RAM, CPU, GPU etc.)

In other words, CS for a CO study should have a changeable architecture and can reply: a) to a multi-dimension of CO by gathering and then processing the largest possible volume of monitoring data about the object; b) to a variety of elements and forms of connections by a variety of architectures of CS' components; c) to a diversity by a variety of data representation forms about CO with details; d) to a multi-criteria by a variety of models and applications; e) to repeated structural changes of the object by the organizing of observation and the appropriate response – for example, rapidly changing CS architecture during the simulation.

Only modern distributed computing systems (DCS) fully meet these requirements. The purpose of this study is finding ways to organize DCS resources, intended for simulation and monitoring of complex objects.

In an ideal system with an unlimited computing resources, a complete search of the structure F and factors of X of a model is possible. When computing resources are limited, a plan of computational experiment is required. This plan has to be reconstructed based on analysis of the computing experiment intermediate results and observational data Z .

A plan is a "tree" or workflow of simple jobs. Each of them $i = 1,|W|$ converts the input data vector x_i into the result one $y_i : y_i = f_i < r_i > (x_i)$. Here r_i - are system requirements for a computing resource.

There are two extreme ways to organize resources into DSC (see Figure 1):

a) computing system for incoming data real-time processing, and the CO behavior forecasting and managing;

b) information system for storing raw data with the possibility of their subsequent processing with limited computing resources.

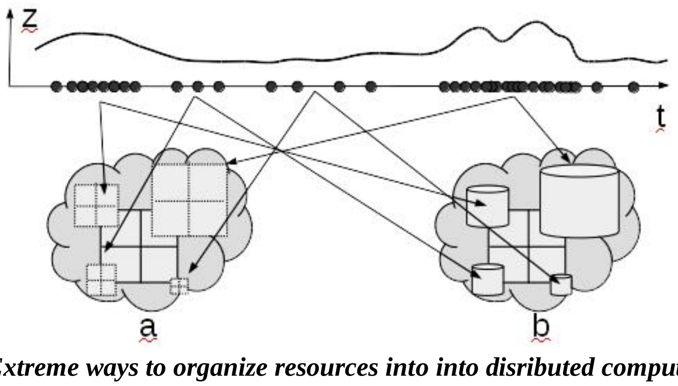

Figure 1. ***Extreme ways to organize resources into into disributed computing system.***

An example of first method is the fog architecture. An example of the second one is the grid architecture has been created to extract the data from the Large Hadron Collider [3].

With the advent of new technology, scientists are trying to use this technology to solve previously unsolved problems. These books [4-6] are devoted to the features of CO simulation in supercomputers, grids and clouds, respectively.

The qualitative characteristic of the assessment is the time of the CO study. In [7], the time of CO research is equated with the time of simulation modeling and does not take into account the time for interpretation of results. However, the computing power of DCS is at best an algebraic sum of its elements power. The idea of DCS as a simple system that capable of simulating CO is incorrect. The diversity of opinions of interpreters takes into account the "Multi-Model object-event approach" [8-9], but the author did not reflect the ways of organizing resources, and did not provide a qualitative assessment of the time of study of CO. This report provides ways to organize the resources of computing, physical resources and storage of DCS and how they affect the time of CO research.

2. Distributed computing system organizing

DCS is an interconnected set of hardware, software, and information resources. Hardware consists of physical, computing elements and storages - the disk arrays connected to working nodes - computers. Physical elements are represented by sensors - devices for collecting data about some object and actuators - devices for external influences on the object. Computing elements are represented by multiprocessor or multi-machine complexes. Software resources - a set of software tools for object modeling, searching for, and providing access to simulation tools, that are adequate for the problem. Information resources — a set of databases for storing simulation models, job description files, source and observational data, results of computational experiments, and assessments are based on observational data and on model simulation results.

Complex object study time consists of the time T_R spent on organizing resources(see Figure 2) calculations T_W , and publishing T_P the results of simulation:

$$T_{co} = T_R + T_W + T_P \ .$$

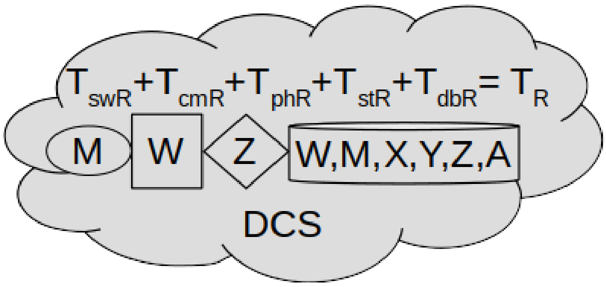

Figure 2. ***Resource types and resource organizing time into disributed computing system.***

The time $T_R = T^{cmR} + T^{phR} + T^{stR} + T^{dbR}$ includes the time to organize computing (cmR), physical (phR) resources, storage systems (stR), software (swR) and information (dbR) resources.

The time for organizing computational experiments (workflow) consists of the time for preparing, sending, performing calculations, extracting, and analyzing intermediate results to assess the feasibility of continuing the computing experiment:

$$T_w = T_w^{pr} + T_w^{sb} + T_w^{rn} + T_w^{et} + T_w^{an}$$

Each of these times is the sum of the times for the corresponding elementary jobs.

3. Computing resources organizing

Worker node (wn), a structural element of multi-machine systems, in turn consist of the storage (memory, M), general and special purpose processors (P). The data for processing are received from peripheral devises (edge, E). There are two fundamental principles for organizing data processing elements: with shared and distributed memory. All the variety of ways to design computing resources is due to the combination of these two principles.

Figure 3 shows a WN structure including graphics processing unit (GPU), which is the most often used for intensive calculations. GPU is combining graphic clusters (GC) with their shared global memory (GM). Each such cluster consists of multiprocessors (MP).

A multiprocessor includes cuda-cores, connected to the shared memory. Each core contains modules for the special (fP) and integer arithmetic operations (iP). These modules are shared operand (oM) and results queue memories (rqM). Cuda-cores are allocated individual register memory. Modern central processing units (CPUs) are multi-core. The cores share a level 3 cache and contain a data processor with individual level 2 and level 1 caches. The level 1 cache consists of blocks for storing instructions and data. In multiprocessors CPUs interacts via system memory.

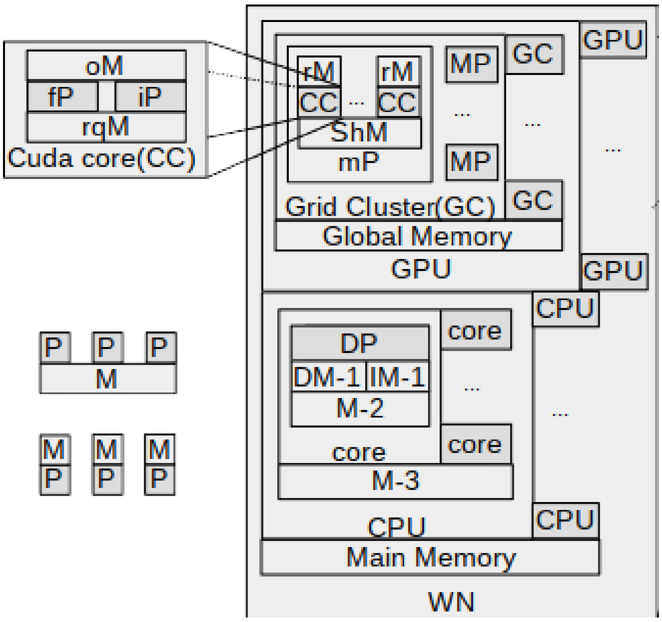

Figure 3. ***A worker node of a computing resource.***

In order to reduce the time of calculation, the same type of worker nodes are combined into multimachine computing systems, clusters. Clusters with shared disk memory of one of the nodes, called the managing node, are used for batch processing of jobs. Batch processing is a method for organizing the sharing of a high-performance computing resource (cluster) by a group of users from the same resource center.

Combining geographically distributed computing resources into grid-system, metacomputer, has become possible by the internet (Figure 4). Grid was designed for computing load balancing and provide access to resources with preinstalled applications. It was necessary due to a lot of time spent on compiling and installing applications for CO simulation. Users from different resource centers who shared a subset computing resources of the metacomputer are grouped into virtual organizations (VOs).

The members of the same VO issue their workflows from the user interface (UI). Local resource management system (LRMS) installed on a control node, selects a subset of available worker nodes, while workload management node (WMS), the most appropriate computing resource for each job of given workflow.

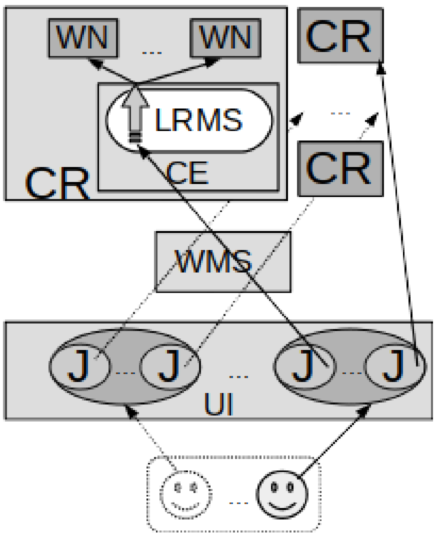

Figure 4. ***The model of providing resources in grid systems.***

Long-job queues, in which jobs are idle most of the time: $T_w^{sb} \gg T_w^{rn}$, have prevented the widespread formation of virtual organizations. In addition, graphics and phi- processing units canceled, for a while, the lack of computing power for grid users. The surplus of high-performance computing resources gives us to change the resource allocation model from grid to cloud. And now resources are allocated not to user workflow, but to a user himself in form of virtual infrastructure, with ability of extraction and transfer it`s components to some other hardware.

In cloud (see Figure 5) computing resources are selected from a pool of computing nodes (CN), and information resources are from a pool of storage nodes (SN). A network node (NN) is responsible for providing network ones.

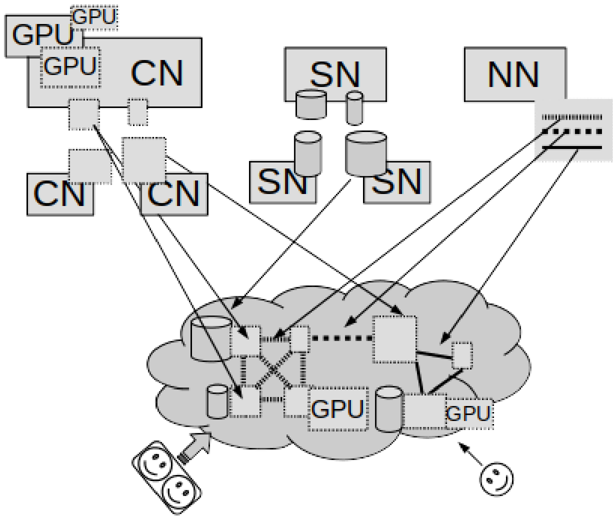

Figure 6. ***The model of providing resources in cloud systems.***

A tenant, accessing a cloud via a management node (not shown in Figure 5), orders the required configuration from virtual worker nodes equipped with the same virtual graphics coprocessors and connected by the same virtual network. A tenant promptly manages the number of resources allocated for the virtual CS in such a way as to minimize the downtime of tasks in queues. The software tools at its disposal give you to consolidate computing resources in the cloud by creating virtual analogues of multi-machine systems, and disperse them by performing the reverse transformation. Cloud give you to accumulate your own or use someone else's experience in creating and problem-orienting of the DCS, thereby minimizing the time required to prepare tools for a researcher of COs.

Thus, the transition from virtual organizations to virtual CS was made. The latter can dynamically change it`s architecture at the request of the researcher or based on the volume of data received about the CO.

4. Fog computing systems.

If in grids jobs are migrate in search of computing power, and in clouds computing power itself migrates from one cloud to another, then in fogs computing power is dispersed close to the sources of processed data. A widespread of low-cost sensors has caused an increase in the volume of transmitted data for monitoring complex objects, and a large flow — a significant load on the network and a long time to extract the results of computing experiments. Fog computing systems constituting a hierarchy of computing resources can reduce this time. Fog computing resource is a layer of so called fog nodes, which combine data transfer and processing (Figure 7).

In fact, fog is an intermediate layer between a distributed array of data sources and cloud storage. The upper layer's fog nodes are connected to clouds, and the lower layer is connected to edge nodes (EN), which receive raw data directly from the sensors. Modern sensors are equipped with wireless data transfer modules. Thus, the distribution of metacomputer peripherals takes place in the fog computing system.

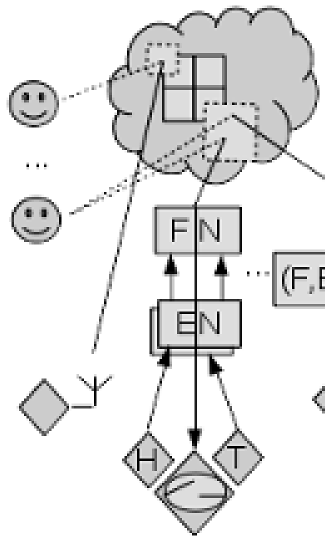

Figure 7. ***An example of fog organizing.***

Data from sensors, for example, temperature (T) and humidity (H) of the environment are collected by edge nodes - microcontrollers and transmitted to the nearest fog node (FN) — a single-board computer. The latter sends data to a virtual node in the cloud with a time series database. The user sees real-time graphs of changes in environmental parameters, and if necessary, changes the indicators of the external environment remotely including humidification and ventilation systems. It can also remotely download the management program for these systems to the corresponding fog or edge node.

Conclusion

A distributed computing system with a fog infrastructure is the most effective for observation, modeling, and management of complex objects. Such system as well as the complex objects under simulation, are a set of computing subsystems with time-varying topology. Fog system is the result of the evolution of computing system from multiprocessor and multi-machine systems to grid and cloud infrastructures. In the report we have showed the ways of resources organizing in DCS, which are ensured the minimum of complex object research time. In the future, we are plan to explore the problem-oriented fog infrastructure for biomedical research.

References

1. A.M. Malyshenko, The mathematical foundations of systems theory [Matematicheskie osnovy teorii system] - Tomsk: Polytechnic University Publ., 364 p. (in Russian)
2. H. Vangheluwe. Foundations of modelling and simulation of complex systems // Proceedings of the 7-th International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2008), Vol.10. pp 59
3. V.V. Korenkov: Distributed system for processing, storing and analyzing experimental data of the Large Hadron Collider [Raspredelennaya systema dlya obrabotki, hraneniya i analiza experimentalnyh dannyh Bolshogo adronnogo kollaydera // Proceedings of XII International scietific and practial conference "Modern information technologies and IT education", Moscow, Moscow State University, 2012, - pp. 8-20. (in Russian)
4. M.K. Chimeh, P. Richmond, Simulating heterogeneous behaviours in complex systems on GPUs, Simulation Modelling Practice and Theory, Volume 83, 2018, pp 3-17
5. K. Kurowski, W. Back, W. Dubitzky, etc. Complex System Simulations with QosCosGrid. Computional Sciences - ICCS 2009, vol. 5544, 2009, pp. 387–396.
6. D. Marinescu. Complex systems and clouds: a self-organization and self-management perspective. - Elsevier, 2017. - 238 p.
7. V.D. Boev. Simulation of systems [Imitatcionnoe modelirovanie system]: textbook for universities.- M. Yurait publ., 2020, 253 p. (in Russian)
8. Brodsky Y., Tokarev V. Fundamentals of simulation for complex systems. // System Analysis and Modeling of Integrated World Systems. V. 1, Oxford: EOLSS Publishers Co. Ltd., 2009. P. 235–250.
9. Yusupbekov, N., Igamberdiev, H., Mamirov, U. Algorithms of sustainable estimation of unknown input signals in control systems. in Journal of Multiple-Valued Logic and Soft ComputingVolume 33, Issue 1-2, 2019, Pages 1-10.