Virtual Testbed: Concept and Applications

Alexander Bogdanov[1], Alexander Degtyarev[1,2(✉)], Ivan Gankevich[1], Vasily Khramushin[1], and Vladimir Korkhov[1,2]

[1] Saint Petersburg State University, Universitetskaya nab., 7-9, 199034 Saint Petersburg, Russia
{a.v.bogdanov, a.degtyarev, i.gankevich, v.korkhov}@spbu.ru, khram@mail.ru

[2] Plekhanov Russian University of Economics, Stremyanny Lane, 36, 117997 Moscow, Russia

Abstract. In this paper the virtual testbed as a problem-solving environment is considered in different aspects. Fundamental questions for virtual testbed development are (1) characteristics of mathematical models and their interaction; (2) computational aspects and mapping of algorithms onto hardware; (3) information streams and data smanagement. The authors propose the concept of a virtual private supercomputer as a tool for virtual testbed computer environment. Examples of the implementation of the virtual testbed in different areas are given. The article summarizes achievements of the authors in the field of virtual testbed during last years.

Keywords: Virtual testbed · Virtual private supercomputer · Direct simulation · Big data

1 Introduction

The development of computer technology and information technology has stimulated the emergence of new mathematical models and numerical methods for solving complex problems over the past 20-30 years. The ability to solve complex problems in an increasingly complete and close to natural conditions formulation has led to the fact that the mathematical models have reached such a high degree of adequacy that they can often be used as a substitute for a physical experiment if the physics of the investigated phenomenon is fully understood. In the vast majority of cases of modeling and designing complex technical objects (ship, underwater vehicle, aircraft, automobile, turbine, etc.) this situation can be considered fair. Carrying out such an experiment using exclusively computer technology based on mathematical modeling can be considered as a virtual experiment, the cost of which is much lower than a physical experiment, and the range of possible conditions for its implementation is incomparably wider, since it allows the study of extreme and potentially dangerous situations in natural conditions. Thus, the concept of a "virtual testbed" emerged as a problem-oriented environment [1].

The emergence of this concept, in the first place, was associated with the need to consider increasingly complex models to study the behavior of dynamic objects that

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12254, pp. 3–17, 2020.
https://doi.org/10.1007/978-3-030-58817-5_1

require the use of high-performance computing. Currently, the use of such types of computing tools requires the researcher to have more knowledge of the features of modern computing technologies. In many ways, this fact becomes an obstacle to their implementation, and as a result, a decrease in the effectiveness of research in various subject areas. The strong gap between the high level of "hardware" and the low level of its application has led to the emergence of a new concept for the use of information technology [2, 3].

When considering the elements of a virtual testbed, the study of complex technical objects requires the use of many models that describe various phenomena. Some of them are independent, and some of them depend on each other. Often the very nature of these applications precludes the possibility of their joint launch in a homogeneous computing environment. In real time, the simulation of all processes of complex objects that affect the final result of complex objects behavior cannot be organized on a single computing node just because it requires the adequate use of various computer resources, for example, high-performance computing, data processing, visualization, etc. Thus, in certain situations, it is necessary to interact parallel computing applications of different complexity and nature, databases, information assimilation, visualization of results, etc. So, the testbed is a complex of multi-level applications, which requires a distributed computing environment [4, 5]. Therefore, in the general case, the use of a heterogeneous computing environment is not the result of scientific experiments, but the only possible tool for implementing of a problem-oriented environment.

This article summarizes the research conducted by the authors over the past 15 years towards the development of the concept of virtual testbed. The references contain the main key works published in this direction.

2 Computational Aspects of Virtual Testbed

Under the accurate mathematical modeling (when nature of the phenomenon is well-known) of the functioning of an object and the possibility of its detailed description, modern computational tools make it possible to almost completely reproduce its behavior. Such an approach allows, in the overwhelming number of situations, to replace an expensive model experiment with a computational simulation. At the same time, a computational experiment is free from such disadvantages as scale effects, limited possibilities to reproduce external excitations, significant difficulties or the impossibility to study complex critical situations in a model experiment, etc. However, in order to computational experiment fully reproduces the real behavior of an object, it is necessary to have good models of the object dynamics and the external environment. These models include the fundamental laws of physics, such as the laws of conservation and the closing relations. For example, for technical objects like ships, aircrafts, cars, etc. we can consider just only three laws: conservation of mass, momentum and energy [6].

$$\begin{gathered} \frac{D\rho}{Dt} + \rho\left(\vec{\nabla}\cdot\vec{V}\right) = 0 \\ \rho\frac{D\vec{V}}{Dt} = \rho\vec{f} + \vec{\nabla}\cdot\Pi_{ij} \\ \frac{\partial E_t}{\partial t} + \vec{\nabla}\cdot E_t\vec{V} = \frac{\partial Q}{\partial t} - \vec{\nabla}\cdot\vec{q} + \rho\vec{f}\cdot\vec{V} + \vec{\nabla}\cdot\left(\Pi_{ij}\cdot\vec{V}\right) \end{gathered} \tag{1}$$

where ρ is density, $\vec{V}$ is velocity, $\vec{f}$ is force of potential nature per unit volume, Π_{ij} are components of stress tensor, E_t is total energy of unit volume, Q is heat dissipation of external sources, $\vec{\nabla}\cdot\vec{q}$ is heat losses due to thermal conductivity through the control surface per unit time. The third and fourth terms in the last equation correspond to the work of mass and surface forces.

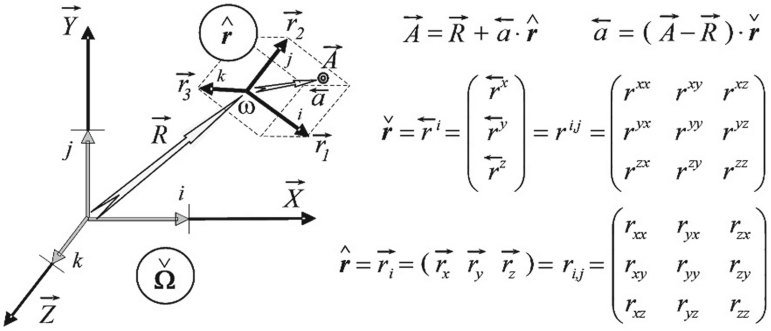
In fact, there is no need to implement these laws in a computational experiment in the form of differential equations or other usual approaches. It is possible to use direct computer simulation with the corresponding fundamental laws applied to all the elementary objects in the space under consideration. This approach requires high-performance computer resources and mapping of the computational algorithms onto the hardware architecture.

Such attempts have been made for a long time, since it became clear that computer technology can really ensure the application of the Lagrange approach in describing the behavior of a continuous medium [7]. The development of this idea over the past 50 years has formed a fairly wide area of CFD, both based on various approaches to solving the Navier-Stokes equation (based on the second equation in (1)), and various particle methods that reproduce the Lagrange approach [7–10]. Currently, one of the most popular methods in this direction is SPH (smooth particle hydrodynamics) [9, 10]. However, all these approaches, ultimately, are based on the finite-difference representation of conservation laws (1), which include derivatives of at least second order. Due to the peculiarities of the conservation equations, their finite-difference analogues are solved using implicit numerical schemes. Such an approach reduces the solution of the problem to systems of linear equations of high dimension. It makes it possible to ensure the stability of the numerical implementation, but it is difficult to parallelize and does not provide step-by-step monitoring of each calculation cell.

At the same time, the development of continuum-particle methods based on the "large particle method" [8] leads to computational models of tensor mathematics with independent control of the state of each calculated cell-fluid particle, computational algorithms and functional logic of the synthesis of physical phenomena and processes for which is provided by parallel-running arithmetic-logic cores, which exactly corresponds to the development trends of computer technology at the request of graphic visualization of three-dimensional space phenomena and dynamic processes with them. It is the use of tensor algebra for direct modeling of physical phenomena and processes as part of generalized tensor mathematics that allows us to efficiently synthesize the hydrodynamic and geometric aspects of the computing process as a whole. This has long been understood in field theory. Such a program was outlined and brilliantly implemented for quantum gravity.

This approach makes it possible to implement direct numerical modeling of unsteady processes using explicit schemes [11–13]. It has an excellent historical analogue in the form of the calculus of fluxes of Isaac Newton.

Geometrical construction of spatial problem includes scalar, vector and tensor numerical objects. For description of large mobile elementary particles in a three-dimensional space we introduce two coordinate systems: an absolute one and a mobile local (associated with the particle) one (Fig. 1).



$$\vec{A} = \vec{R} + \vec{a} \cdot \hat{\vec{r}} \qquad \vec{a} = (\vec{A} - \vec{R}) \cdot \check{\vec{r}}$$

$$\check{\vec{r}} = \vec{r}^{\,i} = \begin{pmatrix} \vec{r}^{\,x} \\ \vec{r}^{\,y} \\ \vec{r}^{\,z} \end{pmatrix} = r^{i,j} = \begin{pmatrix} r^{xx} & r^{xy} & r^{xz} \\ r^{yx} & r^{yy} & r^{yz} \\ r^{zx} & r^{zy} & r^{zz} \end{pmatrix}$$

$$\hat{\vec{r}} = \vec{r}_i = (\vec{r}_x \ \vec{r}_y \ \vec{r}_z) = r_{i,j} = \begin{pmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{pmatrix}$$

Fig. 1. Geometry of global space (Ω (with check)) and local basis (***r*** (with hat)); *i, j, k* are unitary vectors in connected reference systems [12]

In this case, the continuum-corpuscular approach is constructed according to numerical first-order schemes with sequential difference integration of the laws of motion at conjugate stages with respect to the scalar argument - time. Separation of the stages of computations according to the total physical processes allows for end-to-end control and hybrid rearrangement of mathematical dependencies according to current estimates of the state of the simulated continuous medium, taking into account the intensity of the physical interaction of adjacent shell-cells as virtual numerical objects.

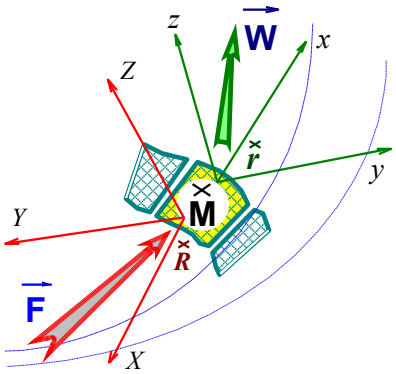


Fig. 2. Separation of calculation stages by physical processes.

Such a separation of calculations by physical stages allows us to divide a computational experiment into three successive stages [11–13]:

Stage 1 – Kinematic parameters are calculated for the centers of large fluid particles. For this purpose, the current source data at fixed nodes of Eulerian coordinates are used.

Stage 2 – Lagrange or large deformable fluid particles are involved in free motion. They redistribute the internal properties of the original Euler cells to adjacent space.

Stage 3 – Laws of conservation of mass and energy are consistent. This is achieved by deformation of shifted fluid particles. The next step is re-interpolation of characteristics of current in initial nodes of the fixed Euler computational mesh.

They are based on computational schemes of mixed Lagrange and Eulerian approaches. Idea of mixing of these two approaches is not new [14]. Approach proposed for the virtual testbed realization has advantages in combination with tensor mathematics and effective realization in code.

The formal construction of physical objects and operations in tensor mathematics leads to rigorous definitions for a kind of "model of the world" of computational hydromechanics illustrated on Fig. 2:

1) the continuum-corpuscular computational model of the "large particles" method in tensor notation reduces to a double linear difference interpolation of physical fields (instead of integrating second-order equations of motion).
2) the movement and interaction of large fluid particles are constructed in the operations of the multiplication, that is more consistent with the physics of spatial processes (there are no restrictions on the smallness of differential approximations).
3) the use of explicit numerical schemes and discrete numerical fields increase the efficiency of direct computational experiments, and do not exclude the possibility of checking the correctness, and, if necessary, using hybrid meshes to achieve adequate engineering results of direct numerical simulation.

3 Virtual Testbed as a Case for Big Data

As we considered in [15] a virtual testbed solves complex problems of modeling and working with big amount of data. The main aspects of a virtual testbed are the following:

- Computing machinery – hardware;
- Uniform information environment – GRID, middleware;
- Program repository – libraries;
- System integration – principles of testbed operation;
- Concept of real time systems.

Combining of all these components in one place makes it possible to organize such system in which all the accumulated data are linked to each other and used when necessary. For example, drawings of technical object created during design process are used in the future to implement the simulators. Training results of operators/navigators/pilots in simulator are used for modification of the object. The data obtained as a result

of direct simulation are used to form knowledge base of on-board intelligent system, etc. We see that this is a problem of Big Data.

However, the problem is the following. As we published earlier [16, 17], Big Data has different nature. In accordance with CAP theorem we divided it in 6 different types [16] (one of them is not implemented). In this classification [16, 17], our system belongs to the type characterized not so much by volume, as by heterogeneity and complex hierarchy. Here we are dealing with both data processing, and complex computations. In the first case, we should consider the data as a whole, in the second case, data are exchanged between different branches. At the same time, in the process of modeling, for example, it must be done at least twice – in preprocessing and postprocessing. As a result, we get a huge amount of heterogeneous data that change the original state of knowledge. With this data and knowledge it is necessary to work differently under different conditions.

The problem is that no software stack describes all classes. Analysis showed that existing Hadoop systems do not cover even 40% of all cases. Our class belongs to this exclusion and selecting tools for working with large amounts of data in this case is a separate task for the development team. Not infrequently, the architecture had to be drastically changed because of increased data loads and control of stored data was lost and the collection of statistics became more and more difficult. There is a need for a solution that allows not only to store all sorts of information with the ability to download from different sources but also has a set of tools to analyze the collected information. A data lake is a concept, an architectural approach to centralized storage that allows you to store all structured and unstructured data with the possibility of unlimited scaling. A data lake can store structured data from relational databases (rows and columns), semi-structured data (CSV, journals, XML, JSON), unstructured data (emails, documents, PDF files) and binary data (images, audio, video). Quite popular is the approach in which incoming data is converted into metadata. This allows you to store data in its original state, without special architecture or the need to know which questions you may need to answer in the future, without the need to structure the data and have various types of analytics - from dashboards and visualizations to big data processing, real-time analytics and machine learning to make the right decisions.

As a result of the analysis of existing solutions, the following functional modules were identified that are the most necessary and need to be developed in the universal solution:

- Storage for all data with the ability to create separate storage for hot/cold data, for ever-changing data or to handle fast streaming
- Security module
- Databases for structured data
- The module of tools for working with data (analysis, data engines, dashboards, etc.)
- Machine learning module
- Services for the development of add-ons, modifications and deployment of storage.

4 Virtual Private Supercomputer as a Tool for Virtual Testbed Organization

Handling massive amounts of data is commonplace for most modern scientific, engineering, and business applications. The Virtual testbed is a good example of a complex system encompassing a number of applications that represent its functional modules listed in the previous section. As these applications need to target a number of big data related challenges, while delivering expected results in a timely manner, they frequently pose large computing power requirements. In this context, High Performance Computing (HPC) becomes a key factor for speeding up data processing, while also enabling faster time to market, lower capital expenditures, and higher valued innovation. To this end, HPC solutions have traditionally taken advantage of cluster and datacenter infrastructures for running applications having those computing power requirements. In addition, practitioners have also been leveraging cloud computing resources for meeting HPC demands when available resources do not suffice. In fact, the pay-per-use cost model and resource elasticity makes cloud computing an interesting environment for HPC, which can be provided with instant availability and flexible scaling of resources, among other benefits.

In spite of the benefits of using cloud computing for HPC, a current approach has been the allocation of physical infrastructures in dedicated mode for fast HPC provisioning. Although convenient, it frequently leads to underutilized resources, e.g., an application may not fully utilize provided CPU and/or network resources. It also prevents dealing adequately with those applications whose resource demands grow beyond available capacity. Traditional virtualization technologies can help solving the problem but the overhead of both a) bootstrapping a virtual infrastructure for each application and b) sharing physical resources among several virtual instances might be significant. Boosting available physical resources by using cloud computing, in turn, has been hampered because of limited support for shifting HPC applications to the cloud. These issues hinder the wide adoption of cloud computing by the HPC community, thus becoming paramount to understand how one can perform smooth and effective migration of HPC applications to the cloud.

Traditional cloud-based solutions are oriented towards long running stateful services that are flexible in respect to consumed CPU and network load at a single timeframe. In turn, HPC applications are in essence batch processes that have clear input and output data requirements. Also, in most cases they are fine-tuned for stable computational environment settings and size. Thus, they significantly differ from cloud-based services in terms of possible management, i.e., their requirements can be quantified. The solution that brings together the flexibility of virtualized cloud-based computing environments and performance of traditional computing clusters is needed to create an application-centric distributed system that provides each application with a customized virtual environment with as much resources as the application needs or is allowed to use.

The Virtual Private Supercomputer [4, 18] is a universal environment for monitoring and managing a high-performance computing cluster, in the structure of which virtual elements are included. Virtual containers are application containers with various

file systems, configuration of modules and software libraries. The tools for combining these elements located on different nodes of the cluster, as well as non-virtualized network equipment and other auxiliary devices and systems into a single system for general-purpose calculations, are the core part of the virtual supercomputer. The virtual supercomputer isolates the user from a number of technical limitations of computing devices using virtualization technologies, allowing users to vary the characteristics of computing elements and balance the workload. To create a personal computing environment with the specified characteristics, lightweight virtualization technologies are used primarily, which leads to minimal overhead, but allows users to create isolated virtual computing systems. The developed tools for managing a virtual supercomputer automate many processes, provide monitoring of node load and task execution, optimally select the right amount of resources based on application requirements and change it if necessary. The virtual supercomputer is designed to solve problems within the time period established by an individual user agreement, which is achieved by selecting and setting up an optimal set of middleware components for the task being solved. In general, the concept of a virtual supercomputer allows users to consolidate heterogeneous resources into a single complex that adapts to the solution of the problem from a computational point of view, taking into account individual requirements determined by the user agreement.

One of the main parts of the virtual supercomputer is a task scheduler for batch processing of data and a software interface to it, designed to develop distributed applications and, in particular, programs for parallel computing and parallel data processing. The scheduler ensures the smooth operation of such programs in the conditions of unpredictable technical (hardware) failures of the cluster computing nodes, by automatically re-executing parts of the task that were executed on failed nodes, on the remaining nodes. The performance of programs written using the provided software interface is not inferior to traditional technologies of parallel computing and running tasks on a cluster (MPI + PBS/SLURM), provided that the cluster nodes do not fail during calculations. Otherwise, the performance significantly exceeds analogues, since not the whole task is restarted, but only a small part of it.

To ensure fault tolerance, each task is divided into control objects - entities that describe the program execution logic. These objects are combined in a hierarchy, which is used to uniquely determine the restart point of the program without interrogating the cluster nodes involved in the calculations. All nodes of the cluster are also combined in a hierarchy used to evenly distribute the load. The calculations are carried out by creating a large number of control objects (one for each logical part of the task) and mapping their hierarchy to the hierarchy of cluster nodes. This mapping has an arbitrary form with the only condition that control objects directly connected to each other are either on the same cluster node or directly connected to each other.

Another major component of the virtual supercomputer is the system for automatic configuration of a distributed computing cluster based on virtualized resources in accordance with application requirements and implementation of a user and software interface to it [19]. One of the key parameters for launching tasks, for determining the value of which the software launch system is responsible, is the amount of required computing resources or cluster configuration. It includes the number of nodes, threads in each node, memory and communication speed between nodes. The user who starts

the task and even the programmer who developed it cannot always determine the accuracy, which cluster configuration or the task will run faster or which configuration uses unreasonably many resources. Within the framework of the virtual supercomputer, a method for automated determination of the optimal cluster configuration, which is used in the developed of a system to launch computing tasks on virtualized resources, was designed and implemented [20, 21]. For users, a web interface is provided in which they can select an application, select computing resources and send a task to be performed. At the same time, all the details of creating and configuring computing resources and performing tasks can be hidden from the user. After completing the task, the software system will upload the output of the task to the user's cloud storage. Depending on the parameters of the task, the software system offers the user the optimal configuration of computing resources (the number of nodes, threads and memory) at which the task execution time will be minimal, or offers to explicitly enter the desired configuration. An API has also been developed for launching tasks, managing and monitoring cluster status based on open standard data exchange protocols.

5 Applications of Virtual Testbed Concept

Let us briefly review applications of developed concept in different directions.

5.1 Marine Virtual Testbed

This application served as a starting point for the development of a general concept for a virtual testbed. An attempt to create a modeling environment for marine objects in the mid-90s based on mathematical models of the behavior of marine objects, the external environment and their interaction, which were widespread at that time, was unsuccessful. The reason for the failure was both the features of the models and the inconsistency of approaches of specialists from various scientific fields: physics, hydrometeorology, and engineering research. The mathematical models for describing the behavior of marine objects were initially of a qualitative nature. This determined the approach in describing external perturbations as a harmonic wave of a given amplitude and frequency. Any complication of mathematical models, consisting in a more accurate description of the disturbing forces, was based on these assumptions [22]. As a result, the complexity of the mathematical description grew, it was no longer possible to obtain solutions of new mathematical models in any form, and it was impossible to use these complex models for direct simulation [23]. The latter is due, first of all, to the fact that the real windwave surface, which is the source of the ship's motion at sea, could not be inserted into any of the serious models developed at that time. They were all focused on the sine wave. On the other hand, there were also no models that could adequately reproduce the spatio-temporal fields of wind waves, due to their irregularity.

To create an integrated modeling environment for marine objects, it was necessary:

1. To develop models of the objects themselves and the external environment that allows direct simulation;

2. To develop approaches to link these heterogeneous models with each other so that the results of one model can serve as the initial data of another model;
3. To provide high performance collaboration of heterogeneous models in a distributed computing environment.

As these tasks were solved, the general concept of the virtual testbed was formed.

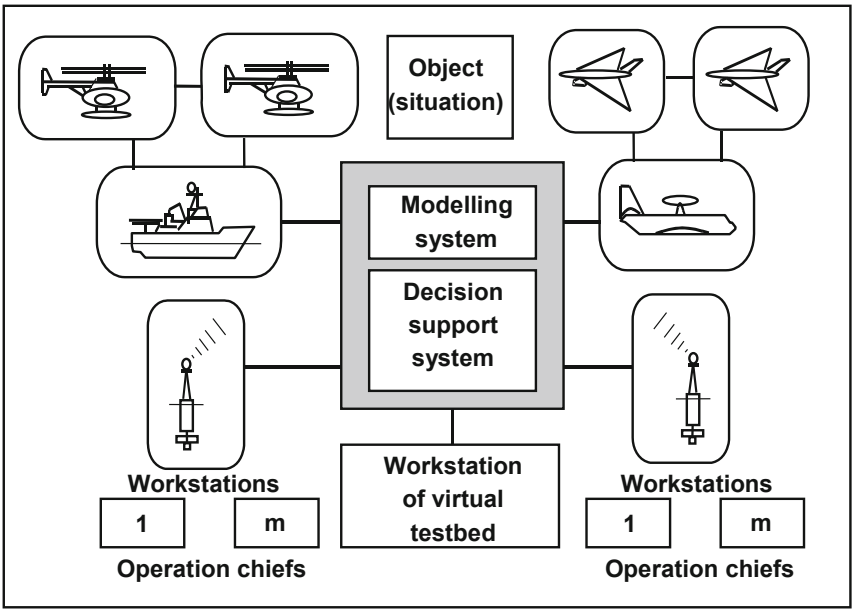


Fig. 3. Functional diagram of complex environment organization

Initially, a model of wind-wave perturbations was developed [24, 25, etc.]. In its final form, from the point of view of application to a virtual testbed, this model is given in [26, 27]. A number of works were carried out to increase the computational efficiency of this model [5, 28, 29, etc.]. This made it possible to develop models of direct simulation of the behavior and interaction of marine objects in the sea. In its final form, the marine virtual testbed is a highly efficient environment for modeling, studying critical situations, planning operational scenarios [30], data assimilation and storage [15] onto a database and a visualization system for results [31]. This virtual testbed became the prototype of a fully functional simulator with elements of a decision support system that provides the following functions:

- Collection and analysis of information about the current state of the facility and the environment, remote monitoring.
- Evaluation and coordination of joint actions of the facility, based on current conditions, with the goal of optimal solution of the general problem.

- Centralized decision support system (DSS) of operators of control systems of dynamic objects in non-standard situations, organization of information support for the interaction of decision makers in the management of ongoing operations.
- Computer modeling of possible scenarios for the development of the situation in order to choose the optimal strategy for managing the facilities of the complex.
- Centralized management of fully automated technical means.
- Document management and information accumulation in dynamic databases.

Schematically, these functions are shown in Fig. 3.

5.2 Knowledge Base Intelligence System Testing

The creation of a fully-functional marine virtual testbed makes it possible to conduct experiments in conditions that are impossible or extremely dangerous in full-scale or model experiments: extreme operating conditions and external influences, catastrophic development of emergency situations (capsizing, avalanche flooding of compartments with loss of buoyancy, etc.), tracking scenarios of joint actions of various technical objects (landing of an airplane/helicopter on the flight deck, mooring and transfer of cargo to the sea, etc.). The on-board decision support systems are called upon to make safe the operation of technical facilities in various conditions and the skill level of the crew. Any of these real-time systems are based on a complex of measuring equipment, processing of dynamic processes, a subsystem of mathematical modeling and a system of rules (knowledge base of an onboard intelligent system).

The formation of a hypothesis testing plan and the implementation of decisions in DSS is carried out on the basis of mathematical modeling data for the vessel dynamics scenarios in the considered extreme situation. The decision-making procedure includes assessing the minimum time for the implementation of the decision with an acceptable level of risk. Decision making is carried out in conditions of uncertainty and lack of time. In this case, there is a risk of incorrect operator actions. For intelligent operator support, fuzzy models are used in this situation. The specificity of such models consists in the use of fuzzy estimates and graph interpretation, which makes it possible to attribute the formation of operator actions to the number of combinatorial problems on graphs [1].

Modeling situations in the face of uncertainties, testing the knowledge base, and ensuring the fastest search for the optimal solution require a detailed recreation of the picture of the interaction between the dynamics of the object and the environment. In this sense, a virtual testbed can be considered as a tool for filling and testing the knowledge base [32].

5.3 Virtual Particle Accelerator

A virtual accelerator reflects the other side of the concept of a virtual testbed. Elementary particle physics is still not fully known. It is for this purpose that various accelerators are built. Each such installation costs a lot of money, requires a lot of effort, but most importantly, after its creation it cannot be used outside the range of parameters narrow enough for which it was designed. A striking example is the LHC.

Therefore, it is fundamentally important in the initial stages to conduct a comprehensive simulation of various components of the accelerator in order to create the optimal design. In particle accelerator physics the problem is that we cannot see what is going on inside the working machine. It is important to represent the space charge forces of beam a software based on analytical models for space charge distributions. For these purposes we need special algorithms for predictor-corrector method for beam map evaluation scheme including the space charge forces. It allows us to evaluate the map along the reference trajectory and to analyze beam envelope dynamics. Such Virtual Accelerator provides a set of services and tools of modeling beam dynamics in accelerators on distributed computing resources [33, 34].

5.4 Virtual Testbed for Financial Modeling

One of the typical spheres of "complex applications", i.e. computational tasks that deal with large amounts of significantly irregular information when the rate of input data subject to processing in a reasonably limited time varies by several orders, is financial mathematics. Finally, the end user to make a decision should have information in a comprehensible form. Such applications are characterized by the following factors.

- Tremendous number of end users (brokers)
- Large variety of heterogeneous sources of information
- Unpredictable moments of sudden data volume "explosion"
- Necessity to keep in mind as long prehistory as possible to make the prognoses more precise
- Limited time to make decisions, in practice in a real time manner

The current state and dynamics of changes in global financial system based on the financial markets play nowadays a significant role in the life of the world economic community. The financial markets are as active as never before. In modern electronic markets, stock prices may change several times within a few milliseconds. Participating traders (that can also be computers) have to evaluate the prices and react very quickly in order to get the highest profit, which requires a lot of computational effort. Information of huge volume received from a large variety of heterogeneous sources is then to be processed using properly adequate mathematical tools.

Over the years, increasingly sophisticated mathematical models and derivative pricing strategies have been developed, but their credibility was violated by the financial crisis of 2007–2010. Contemporary practice of mathematical finance has been subjected to criticism from such a notable figures within the field, as Nassim Nicholas Taleb in his book "The Black Swan" [35]. The book focuses on the extreme impact of certain kinds of rare and unpredictable events and humans' tendency to find simplistic explanations for these events retrospectively. Taleb claims that the prices of financial assets cannot be characterized by the simple models currently in use, rendering much of current practice at best irrelevant, and, at worst, dangerously misleading. Many mathematicians and applied fields scientists are now attempting to establish more effective theories and methods.

Generally speaking, the fundamental computational problem for adequate providing activity of the army of brokers consists in huge amount and large variety of input

heterogeneous sources of information still drastically enlarged by archives' stored data, limited time to make decisions, in practice in a real time manner, and in unpredictable moments of sudden data volume "explosion". Just the case for a virtual testbed [36, 37].

6 Conclusion

Thus, the approach proposed for modeling wind and wave impacts on marine objects turned out to be amazingly effective for a wide range of problems of modeling complex technological and natural systems. It turns out that it is well suited for working with modern hybrid GPU-based calculators, as well as multi-threaded processors. New opportunities for it are opened by the Virtual Private Supercomputer paradigm described above and the approach to classification of Big Data. The combination of these tools allows you to create a flexible toolkit for creating virtual repositories oriented to work with a wide range of applied tasks.

Acknowledgments. The paper has been prepared within the scope of the project of St.Petersburg State University (id 51129503, 51129820) and partly supported by Russian Fund for Basic Research (grant N 17-29-04288).

Competing Interests. The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Bogdanov, A., Degtyarev, A., Nechaev, Y.: Problems of virtual testbed development for complex dynamic processes modelling. In: Proceedings of International conference "Supercomputer systems and its applications" (SSA'2004), Minsk, Belorussia, in Russian. pp. 31–37 (2004)
2. Bogdanov, A., Degtyarev, A., Korkhov, V.: Desktop supercomputer: what can it do? Phys. Part. Nucleic. Lett. **14**(7), 985–992 (2017). https://doi.org/10.1134/S1547477117070032
3. Bogdanov, A., Degtyarev, A., Korkhov, V.: New approach to the simulation of complex systems. EPJ Web Conf. **108**, 01002 (2016). https://doi.org/10.1051/epjconf/201610801002
4. Bogdanov, A., Degtyarev, A., Korkhov, V., Gaiduchok, V., Gankevich, I.: Virtual supercomputer as basis of scientific computing. In: Horizons in Computer Science Research — New York: Nova Science Publishers. vol. 11, pp. 159–198 (2015)
5. Gankevich, I., Degtyarev A.: Model of distributed computations in virtual testbed. In: 9th International Conference on Computer Science and Information Technologies (CSIT 2013), Revised Selected Papers, 6710358. pp. 240–244 (2013). https://doi.org/10.1109/csitechnol.2013.6710358
6. Pletcher, R.H., Tannehill, J.C., Anderson, D.A.: Computational Fluid Mechanics and Heat Transfer, 3rd edn. CRC Press, Taylor&Francis Group (2013)
7. Alder, B., Fernbach, S., Rotenberg, M.: Methods in Computational Physics: Advances in Research and Applications. Fundamental Methods in Hydrodynamics, vol. 3. Academic Press, Cambridge (1964)
8. Belocerkovski, O.M., Davidov, Y.M.: Method of large particles in gas dynamics. Computational Experiment (1982) [in Russian]

9. Monaghan, J.J.: Smoothed particle hydrodynamics. Rep. Prog. Phys. **68**, 1703–1759 (2005)
10. Violeau, D.: Fluid Mechanics and the SPH Method: Theory and Applications. Oxford University Press, UK (2012)
11. Degtyarev, A., Khramushin, V.: Design and construction of computer experiment in hydrodynamics using explicit numerical schemes and tensor mathematics algorithms. Math. Mod. **26**(11), 4–17 (2014)
12. Degtyarev, A., Khramushin, V., Shichkina, Y.: Tensor methodology and computation geometry in direct computational experiments in fluid mechanics. AIP Conf. Proc. **1863**, 110006 (2017). https://doi.org/10.1063/1.4992291
13. Bogdanov, A., Khramushin, V.: Tensor arithmetic, geometric and mathematic principles of fluid mechanics in implementation of direct computational experiments. In: EPJ Web Conf. **108**, 02013 (2016). https://doi.org/10.1051/epjconf/201610802013
14. Hirt, C.W., Amsden, A.A., Cook, J.L.: An arbitrary Lagrangian-Eulerian computing method for all flow speeds. J. Comput. Phys. **135**, 203–216 (1997). https://doi.org/10.1016/0021-9991(74)90051-5
15. Degtyarev, A., et al.: Virtual testbed as a case for big data. CEUR Workshop Proc. **2267**, 58–64 (2018). eid = 2-s2.0-85060084592
16. Bogdanov, A., et al.: Big Data as the future of information technology. CEUR Workshop Proc. **2267**, 26–31 (2018). eid=2-s2.0-85060087168
17. Bogdanov, A., et al.: Is the Big Data the future of information technologies? In: The 20th Small Triangle Meeting on theoretical physics, pp. 15–28 (2019)
18. Gankevich, I., Korkhov, V., Balyan, S., Gaiduchok, V., Gushchanskiy, D., Tipikin, Y., Degtyarev, A., Bogdanov, A.: Constructing virtual private supercomputer using virtualization and cloud technologies. In: Murgante, B. (ed.) ICCSA 2014. LNCS, vol. 8584, pp. 341–354. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09153-2_26
19. Korkhov, V., Kobyshev, S., Krosheninnikov, A., Degtyarev, A., Bogdanov, A.: Distributed computing infrastructure based on dynamic container clusters. In: Gervasi, O. (ed.) ICCSA 2016. LNCS, vol. 9787, pp. 263–275. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42108-7_20
20. Kuchumov, R., Korkhov, V.: Fair resource allocation for running HPC workloads simultaneously. In: Misra, S. (ed.) ICCSA 2019. LNCS, vol. 11622, pp. 740–751. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24305-0_55
21. Kuchumov, R., Petrunin, V., Korkhov, V., Balashov, N., Kutovskiy, N., Sokolov, I.: Design and implementation of a service for cloud HPC computations. In: Gervasi, O. (ed.) ICCSA 2018. LNCS, vol. 10963, pp. 103–112. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95171-3_9
22. Lugovsky, V.V.: Nonlinear problems of seaworthiness. L.: Sudostroenie, pp. 236 (1966). (in Russian)
23. Belenky, V., Sevastianov, N.: Stability and Safety of Ships - Risk of Capsizing, 2nd edn. SNAME, 444 pp. (2007)
24. Bukhanovskii, A.V., Degtyarev, A.B., Lopatukhin, L.I., Rozhkov, V.A.: Probabilistic modeling of sea wave climate. Izv. Atmos. Ocean Phys. **34**(2), 235–239 (1998)
25. Boukhanovsky, A., Rozhkov, V., Degtyarev, A.: Peculiarities of computer simulation and statistical representation of time—spatial metocean fields. In: Alexandrov, V.N., Dongarra, J.J., Juliano, B.A., Renner, R.S., Tan, C.J.K. (eds.) ICCS 2001. LNCS, vol. 2073, pp. 463–472. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45545-0_55
26. Degtyarev, A., Reed, A.: Synoptic and short-term modeling of ocean waves. Int. Shipbuild. Prog. **60**, 523–553 (2013). https://doi.org/10.3233/isp-130091

27. Degtyarev A.B.: New approach to wave weather scenarios modeling. In: Almeida Santos Neves M., Belenky V., de Kat J., Spyrou K., Umeda N. (eds) Contemporary Ideas on Ship Stability and Capsizing in Waves. Fluid Mechanics and Its Applications, vol. 97, pp. 599–617. Springer, Dordrecht (2011). https://doi.org/10.1007/978-94-007-1482-3_34
28. Degtyarev, A., Gankevich, I.: Hydrodynamic pressure computation under real sea surface on basis of autoregressive model of irregular waves. Phys. Part. Nuclei. Lett. **12**(3), 389–391 (2015). https://doi.org/10.1134/S1547477115030073
29. Bogdanov, A., Degtyarev, A., Nechaev, Y.: Parallel algorithms for virtual testbed. In: Proceedings of 5th International Conference Computer Science & Information Technologies (CSIT 2005), Erevan, pp. 393–398 (2005)
30. Degtyarev, A., Khramushin, V., Gankevich, I., Petriakov, I., Gavrikov, A., Grigorev, A.: Virtual testbed: ship motion simulation for personal workstations. In: Misra, S. (ed.) ICCSA 2019. LNCS, vol. 11622, pp. 717–728. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24305-0_53
31. Gavrikov, A., et al.: Real-time visualization of ship and wavy surface motions based on GPGPU computations. CEUR Workshop Proc. **2267**, 415–418 (2018)
32. Nechaev, Yu., Degtyarev, A., Boukhanovsky, A.: Complex situations simultation when testing intelligence system knowledge base. In: Alexandrov, V.N., Dongarra, J.J., Juliano, B. A., Renner, R.S., Tan, C.J.K. (eds.) ICCS 2001. LNCS, vol. 2073, pp. 453–462. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45545-0_54
33. Kulabukhova, N., Bogdanov, A., Degtyarev, A.: Problem-solving environment for beam dynamics analysis in particle accelerators. In: Gervasi, O. (ed.) ICCSA 2017. LNCS, vol. 10408, pp. 473–482. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62404-4_35
34. Kulabukhova, N., et al.: Simulation of space charge dynamics on HPC. In: Proceedings of the 5th International Particle Accelerator Conference IPAC14, June 15–20, Dresden, Germany, pp. 1609-1611. JACoW (2014). https://doi.org/10.18429/jacow-ipac2014-tupri024
35. Taleb, N.N.: The black swan: the impact of the highly improbable. Random House Trade Paperback Edition, 2nd edn. pp. 446 (2010)
36. Bogdanov, A.V., Stepanov, E.A., Khmel, D.S.: Assessment of the dynamics of Asian and European option on the hybrid system. J. Phys. Conf. Ser. **681**(1), 012007 (2016). https://doi.org/10.1088/1742-6596/681/1/012007
37. Bogdanov, A.V., Bogdanov, S.A., Rukovchuk, V.P., Khmel, D.S.: Deep learning approach for prognoses of long-term options behavior. In: Misra, S. (ed.) ICCSA 2019. LNCS, vol. 11620, pp. 631–640. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24296-1_51