

PAPER • OPEN ACCESS

High performance computers: from parallel computing to quantum computers and biocomputers

To cite this article: G Yerlanova *et al* 2021 *J. Phys.: Conf. Ser.* **1889** 032032

View the [article online](#) for updates and enhancements.

You may also like

- [pH tunability and influence of alkali metal basicity on the plasmonic resonance of silver nanoparticles](#)

Vijay D. Yadav, R. Akhil Krishnan, Lalit Borade *et al.*

- [Multiplicity of 're-entrant' cholesteric structures in DNA liquid-crystalline dispersions](#)

Yu M Yevdokimov, S G Skuridin, V I Salyanov *et al.*

- [Improvement of the fidelity of molecular DNA computations: control of DNA duplex melting using Raman spectroscopy](#)

T A Dolenko, S A Burikov, K A Laptinskiy *et al.*



The Electrochemical Society
Advancing solid state & electrochemical science & technology

242nd ECS Meeting

Oct 9 – 13, 2022 • Atlanta, GA, US

Early hotel & registration pricing
ends September 12

Presenting more than 2,400
technical abstracts in 50 symposia

The meeting for industry & researchers in

BATTERIES

ENERGY TECHNOLOGY

SENSORS AND MORE!



Register now!



ECS Plenary Lecture featuring
M. Stanley Whittingham,
Binghamton University
Nobel Laureate –
2019 Nobel Prize in Chemistry



High performance computers: from parallel computing to quantum computers and biocomputers

G Yerlanova¹, M Serik¹ and A Kopyltsov²

¹ Department of Computer Science, Eurasian National University, 2 Satpayeva Street, Nur-Sultan, Republic of Kazakhstan

² Physics Department, Saint Petersburg State University of Aerospace Instrumentation, 67 BolshayaMorskaya Street, St. Petersburg, Russia

E-mail: kopyl2001@mail.ru

Abstract. Various programming methods are considered. Particular attention is paid to parallel programming, quantum computers and biocomputers. This attention is due to the fact that in recent years, high-performance computing has been intensively developing. One of the main ideas for increasing the speed of information processing is to carry out calculations in parallel. For classical programming methods this is achieved thanks to the advent of multiprocessor computers. Such computers allow computational tasks to be parallelized by introducing parallelization elements into classical programming languages. Another approach to speed up computation is based on the idea of a quantum computer. The use of qubits in quantum computers leads to the fact that all possible states of the system are simultaneously processed. Another approach leading to increased computing performance is based on the development of biocomputers. This approach is based on the idea of using DNA chains consisting of a sequence of four nitrogenous bases (adenine, guanine, thymine, and cytosine). The information is stored and processed as a sequence of these nitrogenous bases. An increase in the speed of calculations is carried out due to the fact that biochemical reactions can take place simultaneously on different parts of the DNA - chains.

1. Introduction

In recent years, in connection with the development of the economy, industry, science and other fields of activity, the amount of various processed information is increasing dramatically. Traditional computers are not up to the task. Therefore, various methods are proposed for performing high-performance computing. In particular, by including in classical programming methods (imperative, logical, functional, etc.) constructions that allow for a parallel computational process [1, 2]. Another approach that allows high-performance computing is based on what is proposed new in the field of programming, in particular, a quantum computer based on the concepts of quantum physics [3, 4].

A paradigm is understood as scientific attitudes, ideas and terms on which modern scientific research is based, in particular, in the field of programming. We know from history that paradigms can be both right and wrong. Practice will show which of the paradigms will develop in the future. Historically, programming has developed several paradigms, each of which is designed to solve a certain class of problems on certain computing systems [5, 6].



2. Imperative programming

Initially, the idea of imperative programming arose, which is reduced to the assignment operator and the jump operator "goto", i.e. the calculation process is carried out step by step and is fully controlled at each step. This idea was implemented on the first computers in 1940-1970 and in such well-known programming languages as Fortran (1954), Algol (1960), Pascal (1970), C (1972), ICON (1974), etc. The main area of use is calculations according to given formulas or algorithms, as for the military and civilian purposes [7, 8].

3. Functional programming

Functional programming is based on the concept of a function, i.e. the idea of representing a program as a complex function. In 1960-1980. great hopes were pinned on functional programming in the development of artificial intelligence systems, which were based on studies of the functioning of the human brain. Functional programming languages include such languages as Lisp (1958), REFAL (1968), Scheme (1975), FP (1977), ML (1978), Miranda (1985), Haskell (1990), etc. The main area of use is a class of tasks that are difficult to formulate in the terminology of sequential operations, in particular, tasks of artificial intelligence, visual perception, speech recognition, decision-making, translation between languages, etc.

4. Logical programming

Logic programming is based on the idea that a problem is described in the form of facts (axioms) and logical formulas, and its solution can be obtained by logical inference, i.e. the logical proof mechanism is applicable to the whole program, based on axioms and inference rules (clauses). The first logical programming language was Prolog (1971). 1970-1990 logical programming has developed in three directions. The first direction is associated with the modification of the Prolog language, by introducing modularity, using more powerful logical tools, etc. The languages of this direction include Prolog II (1980), IC-Prolog (1982), MProlog (1983), T-Prolog (1983), LQF (1984), Goedel (1992) et al. The second direction can be characterized as a combination of logical and functional programming. Among the languages of this direction are LOGLISP (1982), LCA (1982), LEAF (1985), Mercury (1993) and others. The third direction is characterized by the use of parallelization constructs. The languages related to this area are Concurrent Prolog (1983), PARLOG (1983), GHC (1986) and others. Logic programming is used to solve problems of artificial intelligence, natural language processing problems, vision, etc.

5. Object-oriented programming

Object-oriented programming is designed to solve problems in which you can select objects and establish connections between them (information exchange). Object-oriented programming languages usually include languages that allow you to define objects that belong to classes and have the properties of encapsulation, polymorphism, inheritance. Three groups of languages can be distinguished: pure, hybrid and stripped-down object-oriented programming languages. The first group includes Simula (1962), Smalltalk (1972), Beta (1975), Self (1986), Cecil (1992), etc. The second includes C++ (1983), Object Pascal (1984), etc. The third group includes Java (1995), C# (2000), etc. The difference between the three groups of languages is that the first group of languages most clearly reflects the methodology of object-oriented programming. The second group of languages appeared through the introduction of object-oriented programming constructs into widely known imperative programming languages. The third group evolved from the second by removing constructs that were unnecessary or dangerous from the point of view of object-oriented programming. The main area of use is modeling relationships between objects of different subject areas.

6. Parallel programming

In connection with the development of computer technology and, in particular, with the advent of multi-core processors, parallel programming began to develop very quickly. Imperative parallel programming,

logical parallel programming, and others appeared. A feature of imperative parallel programming is that in a program you can explicitly select fragments (modules) that can be executed in parallel. If such modules can be selected in a program, then on a multiprocessor computer system the time spent on running the program can be significantly reduced. When performing parallel computations, the problem of synchronizing the interaction between simultaneously executed program fragments arises, which is successfully solved in modern computing systems. The parallelism is usually distinguished at the following levels:

- microinstructions;
- operators;
- cycles (iterations);
- subroutines (functions, procedures);
- control flows;
- processes;
- applications.

To increase computational performance, it is need to use parallelism at all levels.

When writing programs intended for parallel execution, explicit constructions are used. The main approaches to parallel programming can be distinguished, such as programming in a universal language (for example, in the Ada programming language), programming in a widely used programming language (C, C++, Pascal, etc.), to which constructions have been added that allow parallel computations, automatic parallelization of sequential programs by compilers. Programming languages that contain explicit means of parallelism include Algol-68 (1968), Concurrent Pascal (1972), Modula-2 (1978), CSP (1978), Edison (1980), Occam (1982), Ada (1983), Linda (1985), Oblig (1993), etc. The area of application of parallel programming is modeling, real-time systems, processing large data arrays, etc. Logical parallel programming allows natural parallelization, since proof processes can be implemented simultaneously. The languages of logical parallel programming include, first of all, Concurrent Prolog (1983).

7. Quantum computer

A quantum computer is usually understood as a computing device that, in its calculations, uses such concepts of quantum mechanics as quantum superposition, quantum entanglement, quantum parallelism, etc. If an ordinary computer has bits (two stable states denoted by either 0 or 1), then in a quantum computer has qubits (quantum bits capable of accepting both 0 and 1 with some probability). This theoretically makes it possible to simultaneously process all possible states, which for some algorithms the process of obtaining results is accelerated several times.

The idea of a quantum computer was first proposed about 60 years ago [3]. The development of a quantum computer as a physical device is one of the most important tasks of modern physics. There are various approaches to the development of quantum computers. The logical qubits are as follows:

- the direction of the spin (electronic or nuclear) in a given quantum point on a semiconductor;
- the presence or absence of a charge (electron) at a certain point on the semiconductor;
- superconducting elements (presence or absence of a Cooper pair at a certain point in space);
- the ground or excited state of an external electron in an ion;
- quantum states of light, etc. [4, 9]

The problems that arise in the construction of such computers are the accuracy of measurement, minimization of external influences that can destroy a quantum system, minimization of the number of errors, etc.

For quantum computers, programming languages are being developed, for example, Open Quantum Assembly Language (OpenQASM), Q#, LIQUi (Language-Integrated Quantum Operations), Quantum Computation Language (QCL), Quipper, etc.

It is assumed that the main tasks of using quantum computers will be the following:

- problems of cryptography and encryption, for example, Shor's factorization algorithm, etc. ;
- optimization problems, for example, a labyrinth, Grover's algorithm, finding solutions to equations, etc.;
- quantum machine learning;
- simulation of a quantum system, for example, the problem of creating new materials, new drugs, high-temperature superconductors, etc.

The field of application of quantum computers is the study of multiparticle systems in the fields of physics, chemistry, biology, economics, as well as weather forecasting, the creation of new drugs, financial modeling, etc.

At present, quantum computers have been developed, consisting of several dozen qubits. In the near future, quantum computers will be built from several hundred qubits. These computers will allow you to solve problems that a classical computer cannot solve.

8. Biocomputers

One of the important problems is the construction of a computer similar to the human brain. From the point of view of anatomy, the structure of the human brain has been studied in sufficient detail. The structure and location of neurons are known, various areas of the human cerebral cortex are identified, connections between them are established, etc. The studies were carried out on various living objects (earthworms, laboratory mice, rabbits, cats, dogs, primates, etc.). However, it remains a mystery how the brain works? Various approaches are used to solve this problem. In particular, artificial neural networks have been developed. The idea was put forward that the manifestation of intelligence requires a large number of elements, comparable to the number of neurons in the human brain (about 14 billion). Experiments were carried out to create a computer in a test tube (solution) based on nuclear magnetic resonance. Experiments carried out in this area give results that do not yet exceed the results obtained on classical computers.

One of the first publications was article [10], in which it was shown that using deoxyribonucleic acid (DNA) in a test tube, it is possible to solve combinatorial problems, for example, determining the shortest path to traverse the vertices of a graph (the traveling salesman problem). This approach allows using known biochemical reactions to generate all possible solutions, since biochemical reactions can be carried out on different DNA regions simultaneously and independently of each other, i.e. in parallel. However, with this approach, significant difficulties arise, such as the need to control a large number of biochemical reactions. With an increase in the number of vertices of the graph, it is required to significantly increase the volume of the DNA solution.

The idea of such a biocomputer is based on the fact that DNA is a double helix, which includes four nitrogenous bases (adenine, guanine, thymine, and cytosine). The information is encoded by the sequence of these four nitrogenous bases. This information can be changed using enzymes, in particular, to shorten and complete DNA strands. Thus, such biocomputers can store information for a long time and process information quickly enough, since biochemical reactions take place independently on different parts of the DNA chain, i.e. in parallel, which leads to an increase in the speed of calculations.

It is estimated that 1 cm³ contains more than 10 trillion DNA molecules. Such a number of molecules will allow storing more than 10 Terabytes of information and they can perform more than 10 trillion operations per second.

9. Conclusion

Various programming methods are considered: imperative, functional, logical, object-oriented, parallel, as well as quantum computers and biocomputers. It is shown that each of the programming methods is designed to solve a certain range of problems arising at different stages of the development of computer technology. In recent years, high performance computing is required in various fields of activity, such as industry, science, economics and others. One of the main trends in the development of computing technology is that multiprocessor computing systems have appeared that allow parallel computations. This is achieved by including elements in classical programming languages that allow parallelizing computational processes.

In recent decades, in connection with the development of physics and information technology, the fields associated with quantum computing and biocomputers have developed. Quantum computing is based on the use of ideas from quantum mechanics. Quantum computers built on this idea allow solving some problems tens of times faster than on classical computers. However, the implementation of these computers encounters significant technical difficulties, such as maintaining low temperatures, computation time, calculation accuracy, etc. In recent years, the idea of building biocomputers has been intensively developed, which is based on the use of DNA chains composed of sequences of nitrogenous bases (adenine, guanine, thymine, cytosine). The information is stored as sequences of these nitrogenous bases. Information processing is carried out by changing the sequences of these nitrogenous bases using enzymes, i.e. shortening, building up chains, replacing elements in chains, etc. Acceleration of computations is carried out due to the fact that biochemical reactions that provide the computational process on different parts of the DNA - chains can take place simultaneously, ie parallel.

Thus, in classical, quantum and biocomputers, the increase in computational performance is carried out due to the parallelization of computational processes. Therefore, when training future specialists in the field of information technology in high-performance computing, it is necessary to take into account the trends in the development of computer technology and programming, which are associated with the development of mathematics, physics, chemistry, biology, and computer science.

References

- [1] Ceruzzi P E 1998 *History of Computing* (Cambridge, Massachusetts: MIT Press)
- [2] Hartmann A K 2009 *Practical Guide to Computer Simulations* (Singapore:World Scientific)
- [3] Feynman R 1982 *Int. J. Theor. Phys.* **21** 467-88
- [4] Knill G J, Laflamme R and Milburn G J 2001 *Nature* **409** 46-52
- [5] Gries D 1981 *The Science of Programming* (New York: Springer-Verlag)
- [6] Kopyltsov A V 2019 *J. Phys.: Conf. Ser.* **1399** 033016
- [7] Kopyltsov A A and Kopyltsov A V 2020 *J. Phys.: Conf. Ser.* **1679** 032049
- [8] Kopyltsov A, Serik M and Bakiev M 2020 *J. Phys.: Conf. Ser.* **1515** 032018
- [9] Nilsen M and Chang I 2010 *Quantum Computation and Quantum Information* (New York: Cambridge University Press)
- [10] Adleman L M 1994 *Science* **266** 1021-4