11th International Conference
"Distributed Computing and Grid Technologies in Science and Education"

Supercomputing Co-Design: To Know, To Be Able, To Master
(Суперкомпьютерный кодизайн: знать, уметь, владеть)

Prof. Vladimir Voevodin
Research Computing Center, MSU, Director
MSU Branch in Sarov, Director
CMC Faculty, MSU, Head of Department

voevodin@parallel.ru

July, 7th, 2025, JINR, Dubna

Models of polymer-based molecular machines and self-organization processes in polymer systems: theory and computer simulation

Ivanov V.A.[1], I.I. Potemkin[1], Buglakov A.I.[2], Markina A.A.[3], Astakhov A.M.[3]
1) Lomonosov Moscow State University 2) INEOS RAS 3) FRCCP RAS

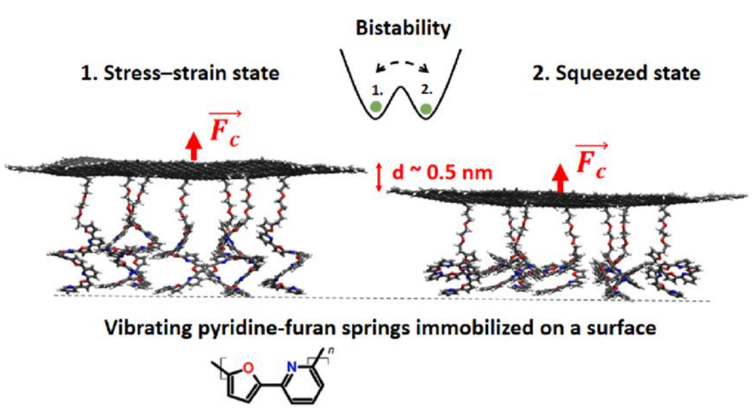
DRIVER. *Study of phase behavior and dynamic properties of different (macro)molecular systems.*

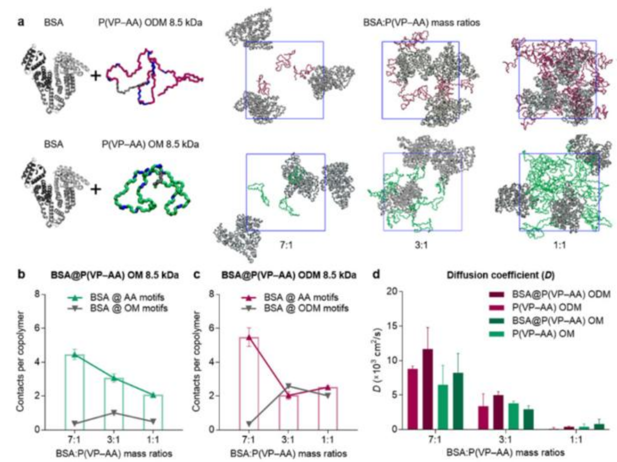
STRATEGY. *Multiscale computer simulations: atomistic and coarse-grained molecular dynamics, dissipative particle dynamics.*

OBJECTIVE. *Development of effective methods of multiscale computer simulation for prediction of macroscopic properties of (macro)molecular systems from the chemical structure of molecules.*

IMPACT. *Efficient, quick and cheap rational development of new functional and constructive polymer materials.*

USAGE. *Physical chemistry of polymers.*





Numerical modeling of geodynamic processes

Zakharov V.S., Perchuk A.L.
Lomonosov Moscow State University

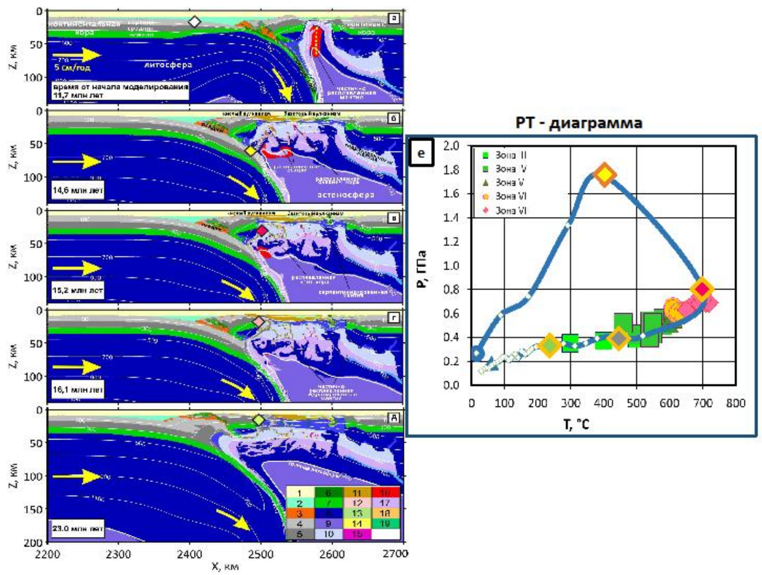
DRIVER. *The study of character and features of main geodynamic processes (subduction, continental collision, rifting, spreading, thermal history of geological bodies, etc.).*

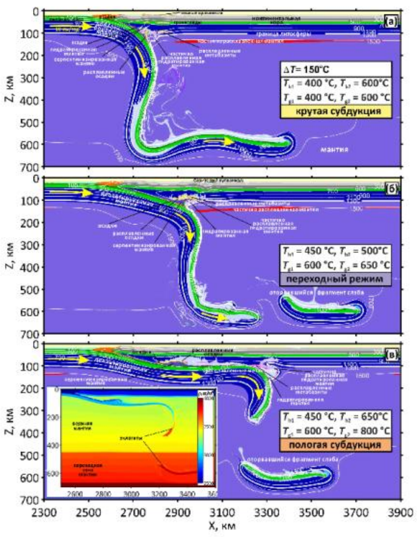
STRATEGY. *Complex thermomechanical and petrological models based on the finite difference method and marker-in-cell techniques.*

OBJECTIVE. *Creation of consistent models of basic geodynamic processes (subduction, continental collision, rifting, spreading, thermal history of geological bodies, etc.).*

IMPACT. *Progress in understanding the evolution of plate tectonics during Earth's history.*

USAGE. *Geology, geotectonics, geodynamics.*





Gravitational field and rotation of the Earth

Zharov V. E., Ayukov S. V., Loginov A. V., Sementsov V. N.
Lomonosov Moscow State University

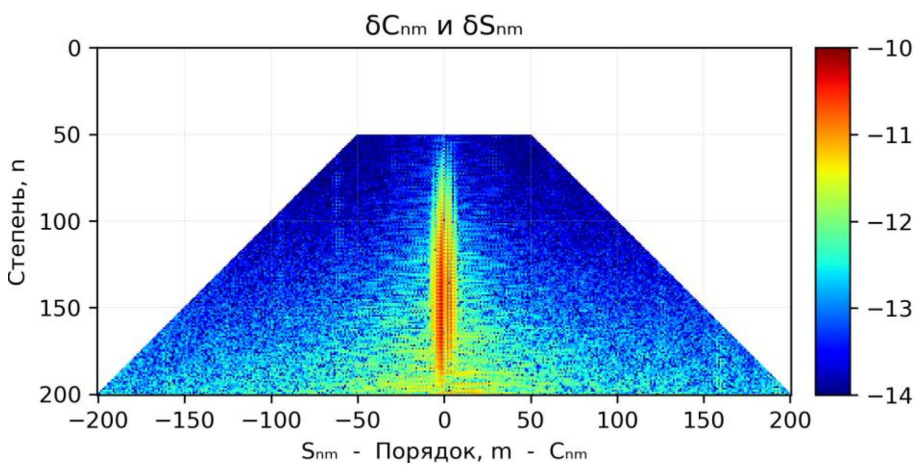
DRIVER. *Reconstruction of the Earth's gravity field from simulated and real measurements of the gravity gradiometer.*

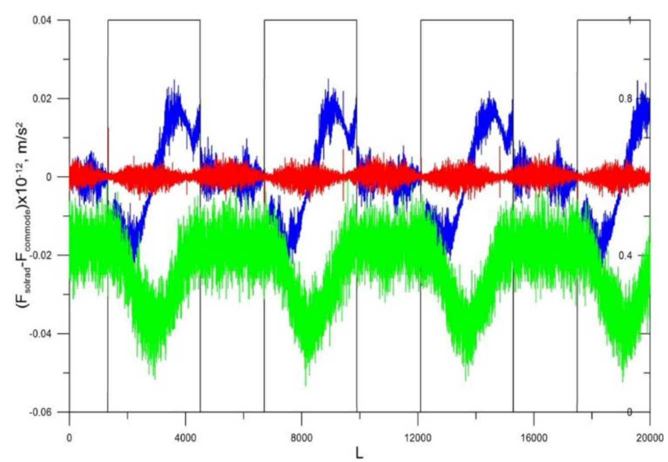
STRATEGY. *Processing large daily matrices for geofield modeling requires high computing power, parallelization and innovative methods.*

OBJECTIVE. *Software development for modeling and processing of measurement data of the gravity gradiometer installed onboard the spacecraft, as well as other onboard scientific and telemetry equipment.*

IMPACT. *Increase of calculation speed, refinement and sophistication of gravity field models due to parallel processing of daily observation intervals and use of large memory capacity on supercomputer nodes.*

USAGE*. Space technologies relevant to telecommunications.*





Eddy-resolving modeling of the general circulation of the ocean and the study of the physical aspects of its eddy variability

Diansky N.A., Bagatinskaya V.V., Budnikov A.A., Kolesov S.V., Sementsov K.A.
Lomonosov Moscow State University

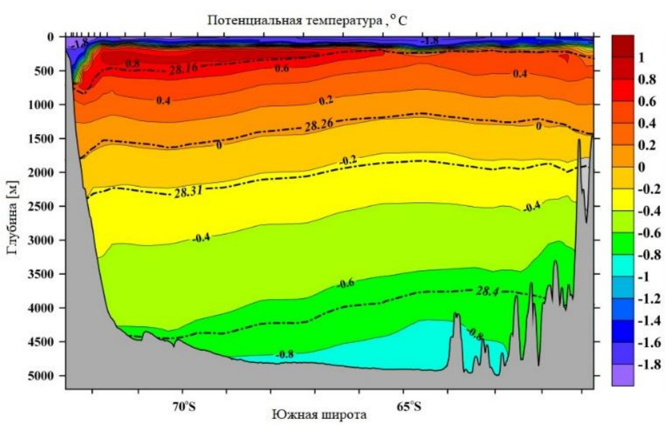
DRIVER. *Conducting laboratory and field studies and mathematical modeling of the influence of physical and chemical inhomogeneities on energy and mass transfer in the ocean atmosphere system.*

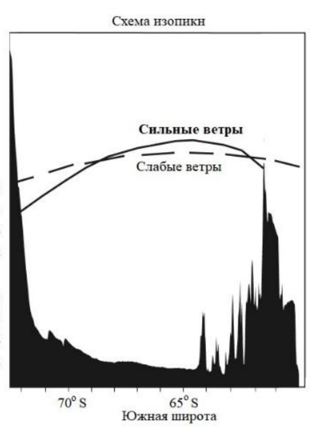
STRATEGY. *Creation of experimental setups and research methods.*

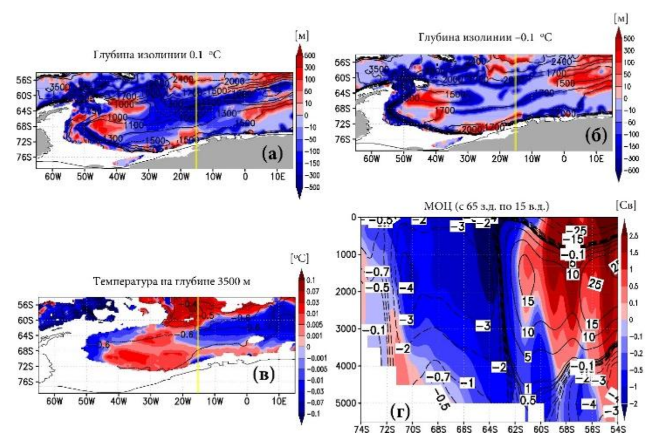
OBJECTIVE. *Development of systems for forecasting the main parameters for the atmosphere and ocean, in the boundary layers of this system. Evaluation of the operation of tsunami warning systems.*

IMPACT. *NMOM model using CUDA technology.*

USAGE. *Computer science, geophysics, oceanology, atmospheric physics, land geography and hydrology, algorithmic and software.*







Simulation of complex turbulent flows using unstructured meshes

A.P. Duben, I.V. Abalakin, P.A. Bakhlovalov, V.G. Bobkov, A.V. Gorobets, N.S. Zhdanova, T.K. Kozubskaya, P.V. Rodionov Keldysh Institute of Applied Mathematics of RAS

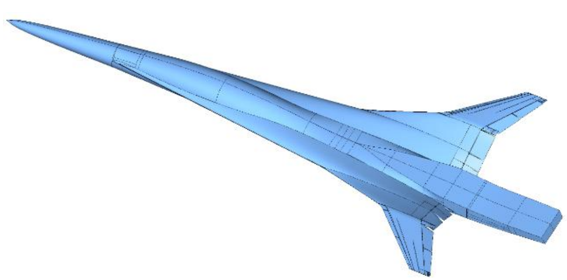
DRIVER. *Numerical investigation of aerodynamic and aeroacoustic charateristics of bluff aircraft elements and complex turbulent flows*.

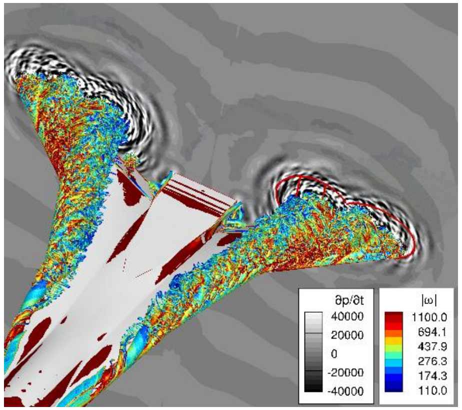
STRATEGY. *Providing computational experiments on different configurations, comparison experimental and numerical data, identification of optimal configurations.*

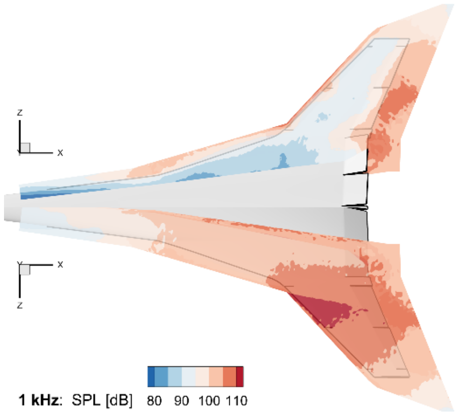
OBJECTIVE. *Investigation and improvement of aerodynamic and aeroacoustic characteristics of aircraft elements and complex turbulent flows.*

IMPACT. *Improvement of aircraft aerodynamic characteristics; reduce of airframe noise in near and far field.*

USAGE. *Aircraft industry; computational mathematics; computer science.*







Simulation of branched molecules, microgels and multi-component polymer systems in solvents and melts

Gumerov R.A.
Lomonosov Moscow State University

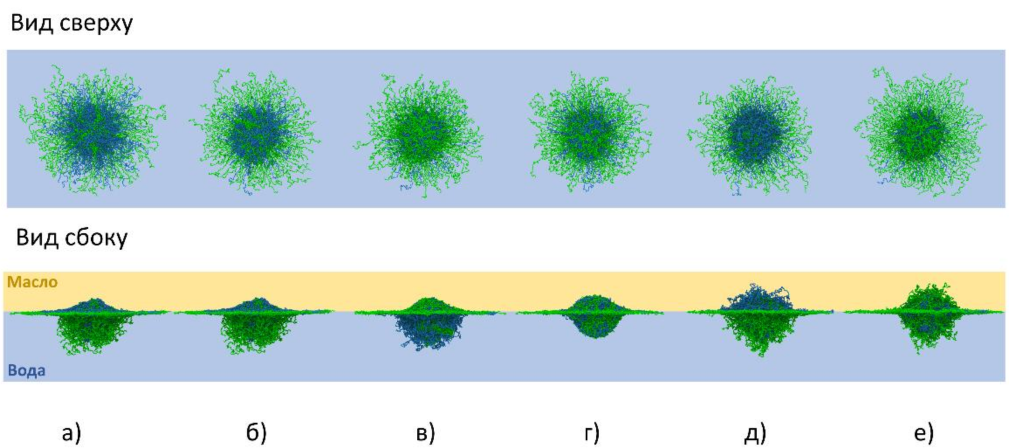
DRIVER. *Study the adsobtion of arborescent molecules, microgels on the interfaces and changes of structural characteristics in block copolymer films and nanocomposites during swelling.*

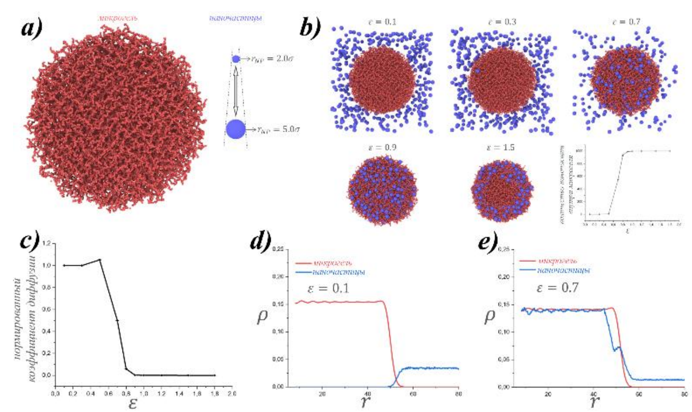
STRATEGY. *Placing a few molecules in a solvent, observe its behavior. The behavior if simulated by the dissipative particle dynamics method.*

OBJECTIVE. *To obtain the stable emultions and long-range ordered films. In addition, the morphology of molecules and films must correspond to the particular expectances.*

IMPACT. *The microgels and arborescent molecules can cover the bigger area of liquid's interfaces in comparison with linear polymers and surfactants. Swollen thin block copolymer films have inhomogeneous distribution of solvent along the domains.*

USAGE. *High adsobtional properties are used for the obtainment of stable micro- and nanoemultions. Thin films can be used as nanopumps.*





Multi-scale modeling of cytoskeletal microtubules and associated proteins

Aleksandrova V.V.[1], Vinogradov D.S.[1,2], Lopanskaya Yu.N.[2], Fedorov V.A.[2], Kovalenko I.B.[2], Gudimchuk N.B.[1,2,3]

1) Center for Theoretical Problems of Physico-Chemical Pharmacology, RAS 2) Lomonosov Moscow State University 3) Federal Research Clinical Center for Pediatric Hematology, Oncology and Immunology

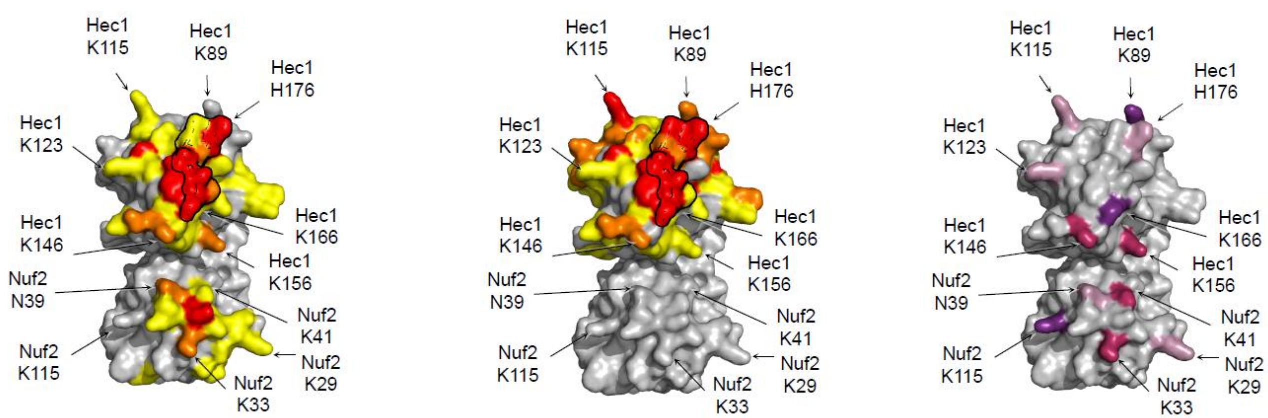
DRIVER. *Microtubules are key intracellular structures that play a major role in such processes as cell division, intracellular transport, directional cell migration, neuron germination, etc.*

STRATEGY. *In order to achieve the goals of the project, molecular models of microtubules with different levels of detail will be created.*

OBJECTIVE. *Create a multiscale model of the tubulin microtubule and its complexes with associated proteins in order to establish the relationship between the behavior of the cell skeleton on scales from atoms to the whole cell, from femtoseconds to seconds.*

IMPACT. *A detailed understanding of the factors and mechanisms that determine the behavior of microtubules will bring valuable fundamental information about a large number of cellular processes and will shed light on their mechanisms.*

USAGE. *Pharmacology, medicine, molecular and cell biology.*



The study of polylactic acid byatomistic molecular dynamics

Guseva D.V., Lazutin A.A., Vasilevskaya V.V.
A.N.Nesmeyanov Institute of Organoelement Compounds of RAS

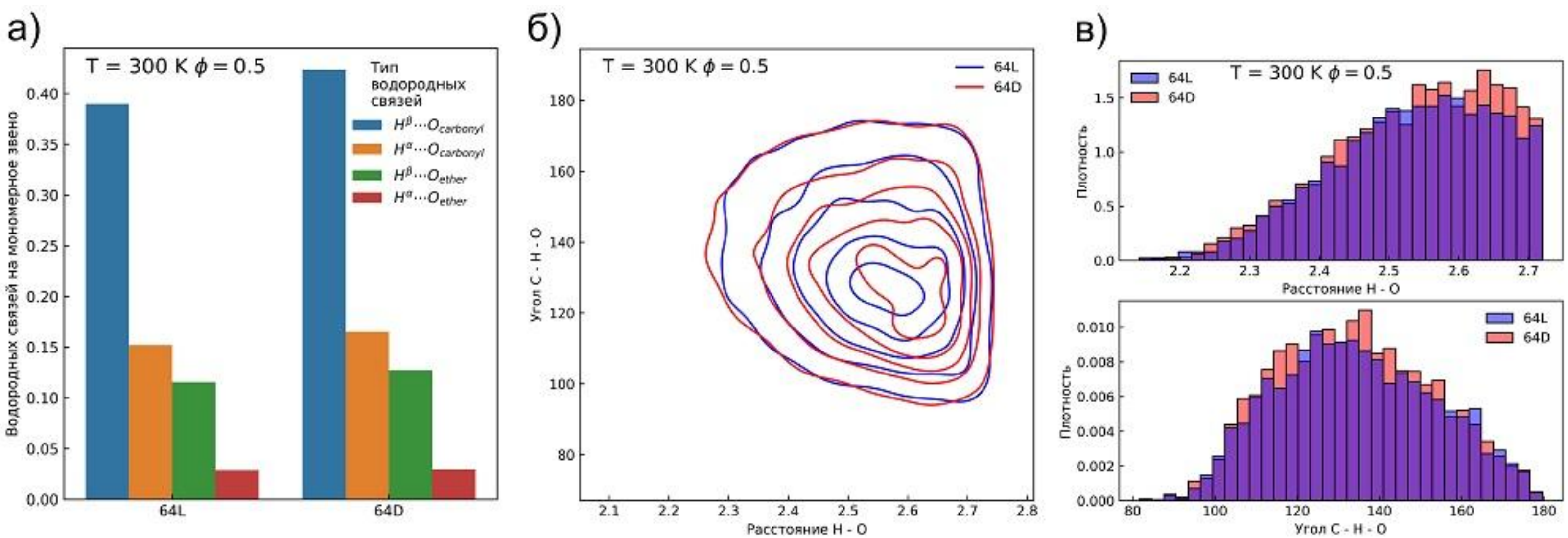
DRIVER. *Design of polymeric materials based on polylactic acid with improved structural properties.*

STRATEGY. *Investigation of the effect of chain length and the relative content of L- and D- monomer units in polylactic acid on the final properties of the system.*

OBJECTIVE. *Develop an atomistic model of polylactic acid and computer simulation technique.*

IMPACT. *Improved methods for efficient and predictable preparation of polymeric materials based on polylactic acid.*

USAGE. *Computer-aided design of biodegradable polymeric materials with improved structural properties.*



Development of an automated method for optimizing the SPHERE telescope model

Galkin V.I. Lomonosov Moscow State University

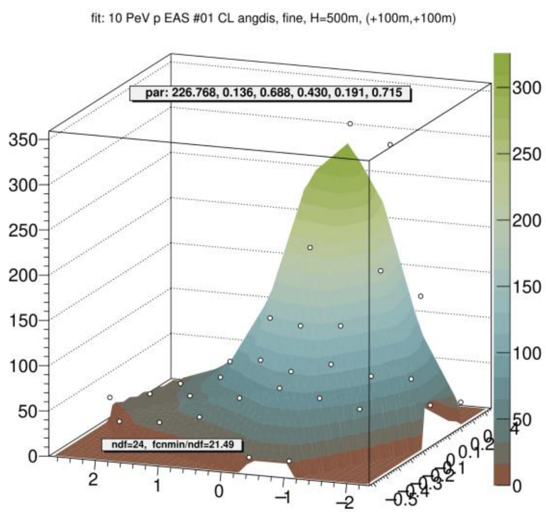
DRIVER. *Development and implementation of the primary cosmic rays.*

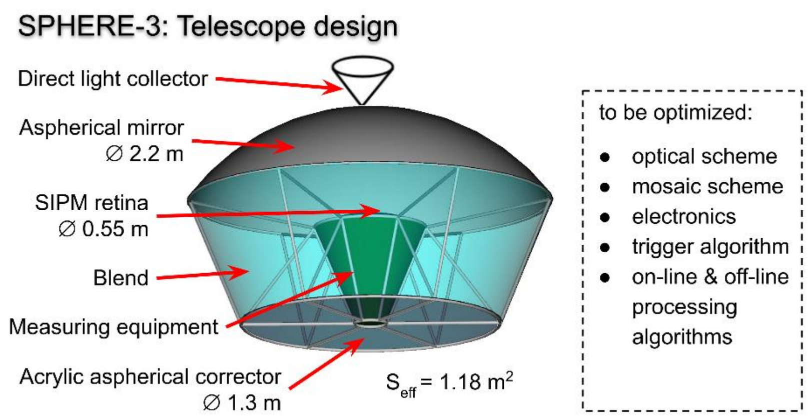
STRATEGY. *Detailed statistical modeling of the development of extensive air showers (EAS) .*

OBJECTIVE. *Construct an algorithm for choosing the best telescope design from a certain series; show the efficiency of this optimization technology.*

IMPACT. *An automated method for optimal design of a SPHERE-type telescope will be developed.*

USAGE. *The implementation of the project will create new knowledge in the neighboring fields of physics and experimental engineering, atmospheric physics and environmental monitoring, mathematical methods of experimental processing and big data systems.*





Theoretical modelling of electronic and magnetic properties of endohedral fullerenes and their derivatives

Ioffe I.N., Lukonina N.S., Mazaleva O.N., Sudarkova S.M., Pykhova A.D., Khinevich V.E.
Lomonosov Moscow State University

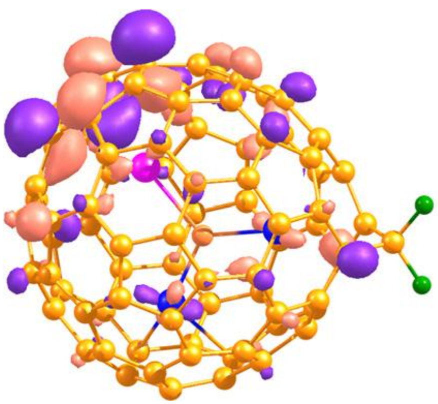
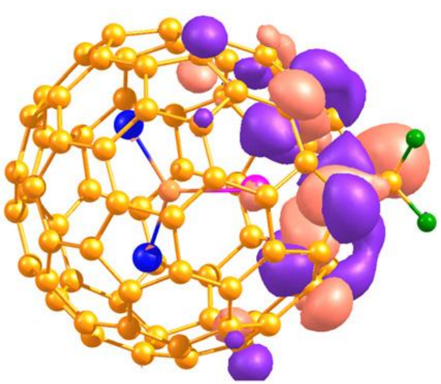
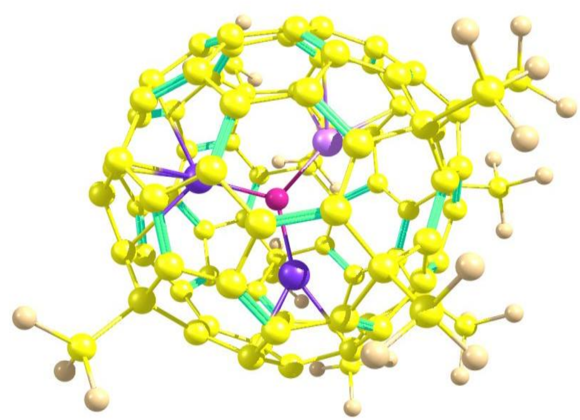
DRIVER. *The present project is aimed at identification of the promising single-molecule magnets based on the endohedral metallofullerenes.*

STRATEGY. *All possible orientations of the endohedral atoms within the carbon cage will be considered in order to elucidate the effect of the exohedral addends on their dynamics.*

OBJECTIVE. *Elucidation of the dynamics of the endohedral clusters inside various carbon cages and of its interplay with the exohedral addition patterns.*

IMPACT. *The project will provide better understanding of the fundamental aspects of the electronic and magnetic properties of endohedral fullerenes. It will further help to identify those endohedral fullerenes that may be applicable as novel materials for spintronics, information storage, etc.*

USAGE. *Fundamental and applied material science.*

The fundamental research effort on post-genom investigations and technologies

Domnin M. V., Aslanly A. G., Lyagin I. V.
Lomonosov Moscow State University

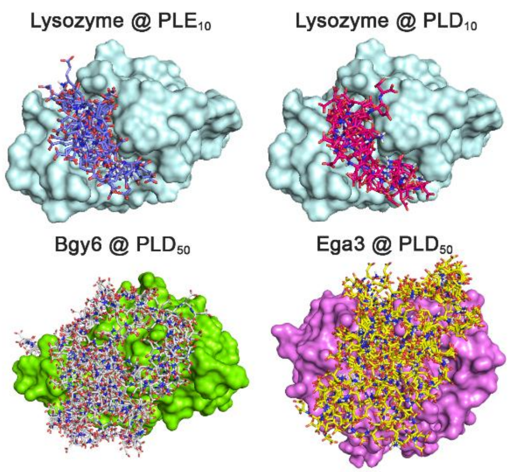
DRIVER. *The project aims to model nanocomplexes of quorum-quenching enzymes, enzymes that degrade fungal components and/or mycotoxins, with polymers and various subgroups of antifungal agents.*

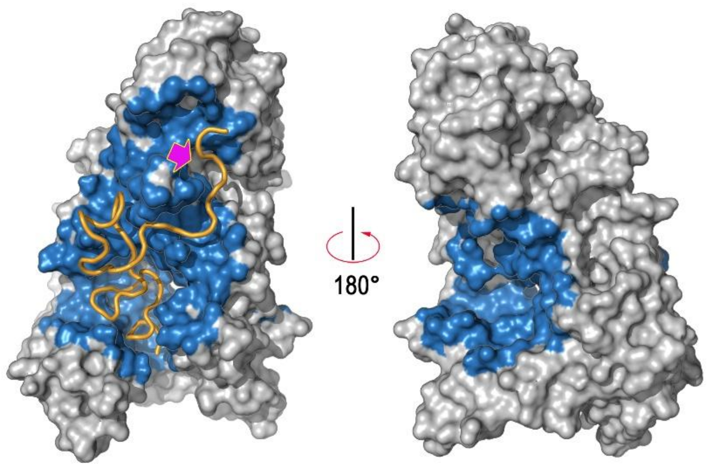
STRATEGY. *Computer calculations of formation energy of complexes between enzymes and various ligands.*

OBJECTIVE. *To create new approaches for the rational design and subsequent design of nanobiocatalysts based on complexes of quorum-quenching enzymes.*

IMPACT. *An effective and available approach to the development of complex antifungal preparations.*

USAGE. *Used to solve a wide range of problems associated with contamination by microscopic fungi in agriculture, in the food industry, as part of tissue materials, in the processes of cleaning industrial wastewater and waste, etc.*





Computer modeling of multicomponent polymeric systems

Lyulin S.V., Gurtovenko A.A., Larin S.V., Nazarichev V.M., Volgin I.V.
Institute Of Macromolecular Compounds RAS

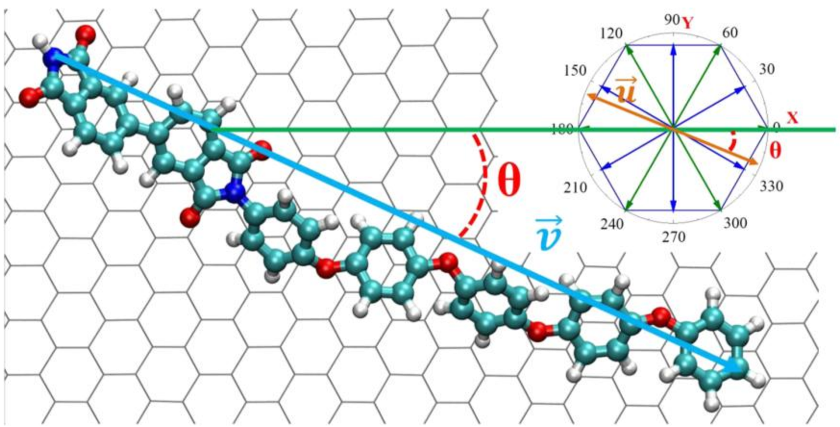
DRIVER. *Study and prediction of structure and dynamic properties of polymeric multicomponent materials.*

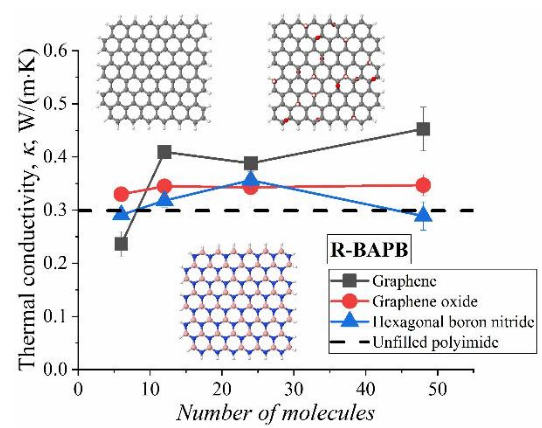
STRATEGY. *Development of models of polymeric multicomponent materials, further computer simulation of their structure and dynamic properties on the base of quantum mechanics and full-atomic molecular dynamics.*

OBJECTIVE. *Virtual design and testing of advanced polymer nanocomposites, including development of nanocomposite models, further computer simulation of their structure and dynamic properties, study of reinforcement mechanisms.*

IMPACT. *Methodology of calculation and prediction of properties of polymeric multicomponent materials.*

USAGE. *Development of new polymeric multicomponent materials.*





Using multiprocessor systems to solve ill-posed inverse problems of science and technology

Lukyanenko D.V.
Lomonosov Moscow State University

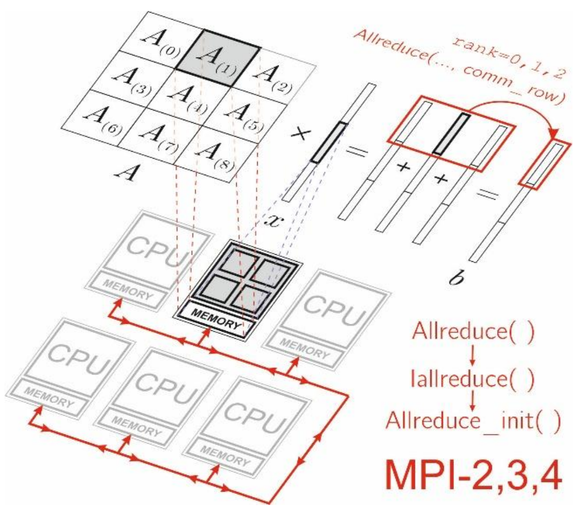
DRIVER. *Development of efficient parallel software implementations of modern numerical methods for solving ill-posed inverse problems.*

STRATEGY. *Develop software implementations of advanced regularizing algorithms that allow solving such problems in a general form without using simplifications.*

OBJECTIVE. *Creation of efficient parallel software implementations of modern regularizing algorithms for solving ill-posed inverse problems.*

IMPACT. *General methods for solving multidimensional ill-posed problems.*

USAGE. *Various fields of science and technology in which it is necessary to solve ill-posed inverse problems.*



Computer simulation of polyelectrolyte complexes, which are mixtures of polypeptides and polyanions

Kravchenko V.S.
Lomonosov Moscow State University

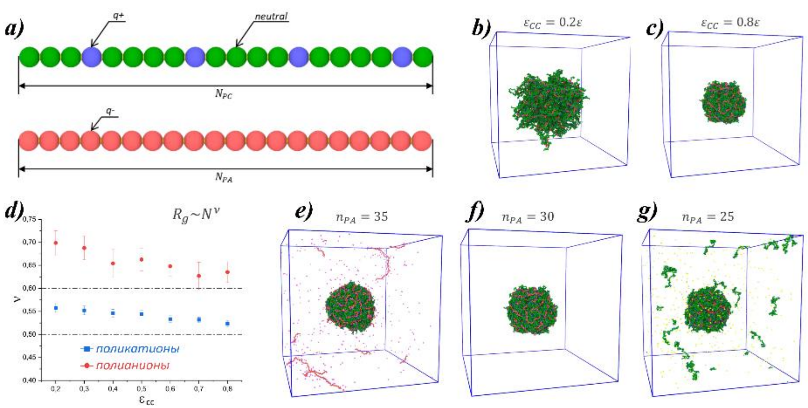
DRIVER. *Investigation of polyelectrolyte complexes containing polycations and polyanions.*

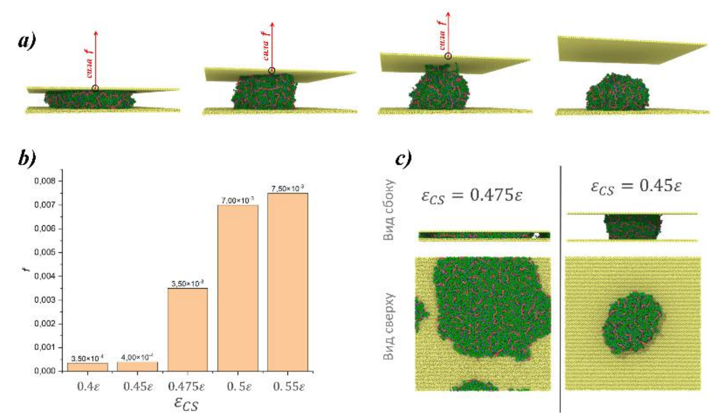
STRATEGY. *Simulation of systems containing charged polymer chains.*

OBJECTIVE. *The purpose of this project is to study cohesion and adhesion in systems containing mixtures of polycations and polyanions.*

IMPACT. *Computer simulation of such mixtures will help to answer the question of how the parameters of the polymers included in the mixture affect the properties of the mixture itself.*

USAGE*. The potential use of these mixtures as biocompatible adhesives in medicine.*





Modeling of jet flows in multicomponent gas mixtures in containment building

Glotov V.Yu., Kanaev A.A., Danilin A.V.
Lomonosov Moscow State University

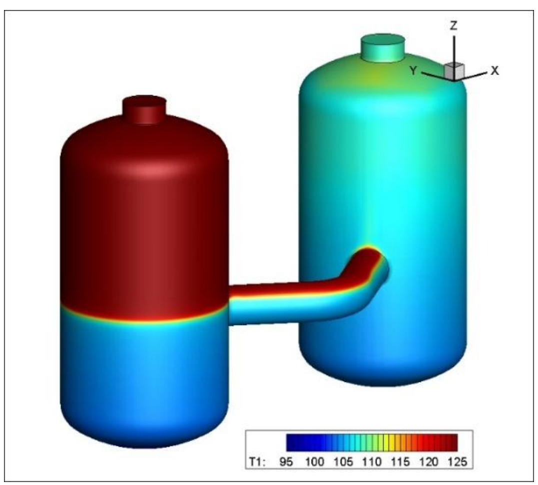
DRIVER. *Investigation of large-scale gas mixing and stratification phenomena.*

STRATEGY. *Conducting a series of calculations , results analysis, comparison with experimental data.*

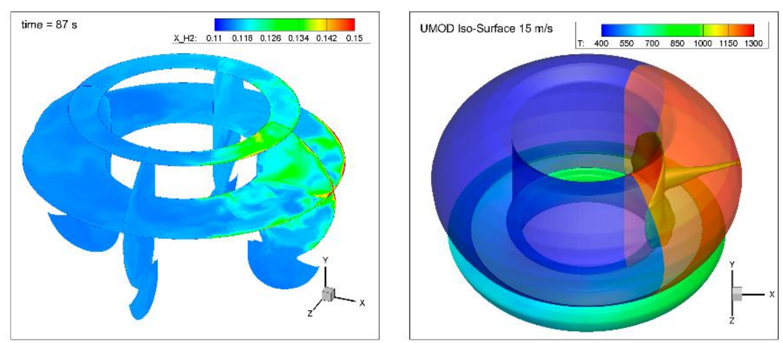
OBJECTIVE. *Development and verification of a modern CFD engineering software for modelling of large-scale gas mixing and stratification phenomena.*

IMPACT. *Nuclear safety.*

USAGE. *Scientific.*



CFD-моделирование упрощенных сценариев аварий ex-vessel LOCA на ТЯУ типа ИТЭР с генерацией или утечкой водорода



Распределение объемной доли водорода

Изоповерхности скорости

Ultrasound tomography in medicine

Goncharsky A.V., Romanov S.Yu., Seryozhnikov S.Yu.
Lomonosov Moscow State University

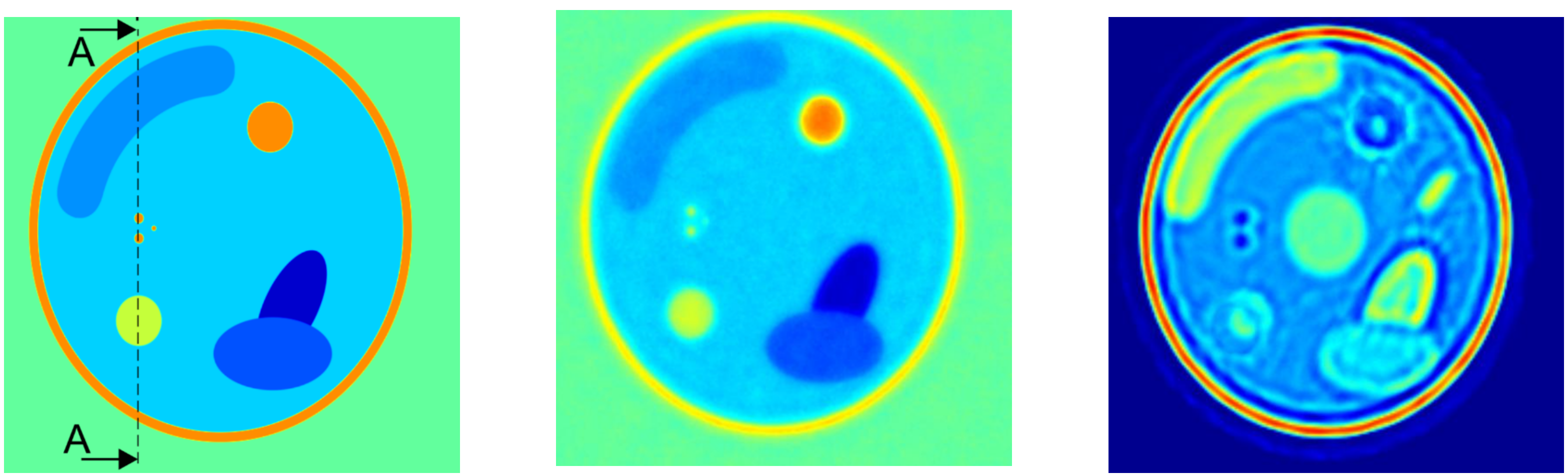
DRIVER. *Non-destructive investigation of the internal structure of the object.*

STRATEGY. *Solution of three-dimensional nonlinear coefficient inverse problems for wave equations on supercomputers.*

OBJECTIVE. *Study of the internal structure of the object by wave sounding with super resolution.*

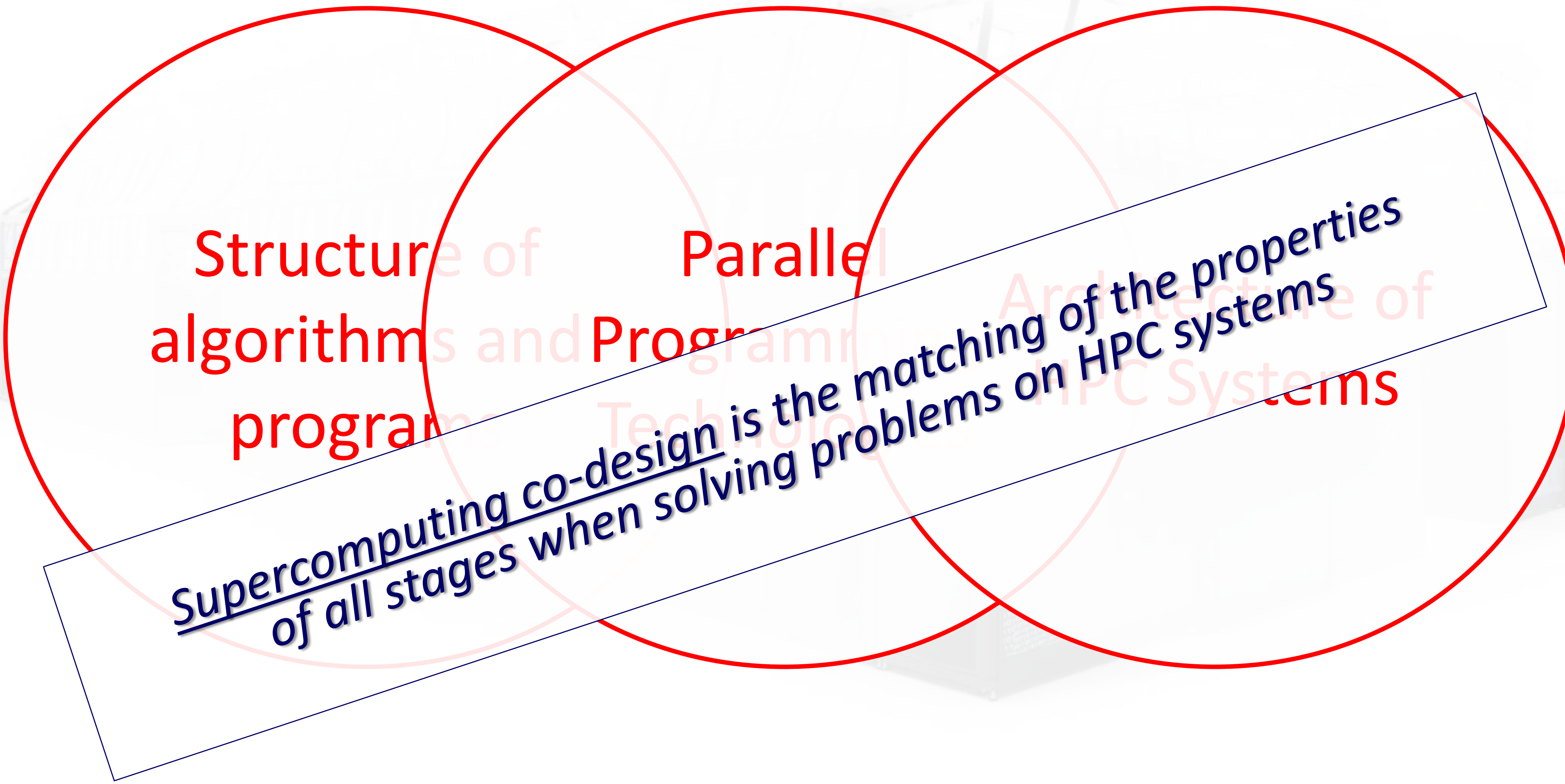
IMPACT. *Break-throw technology of computer aided non-destructive diagnostics with wave soundin.*

USAGE. *Ultrasound tomography in medicine, wave sounding of the surface layers of the Earth, industrial tomography.*

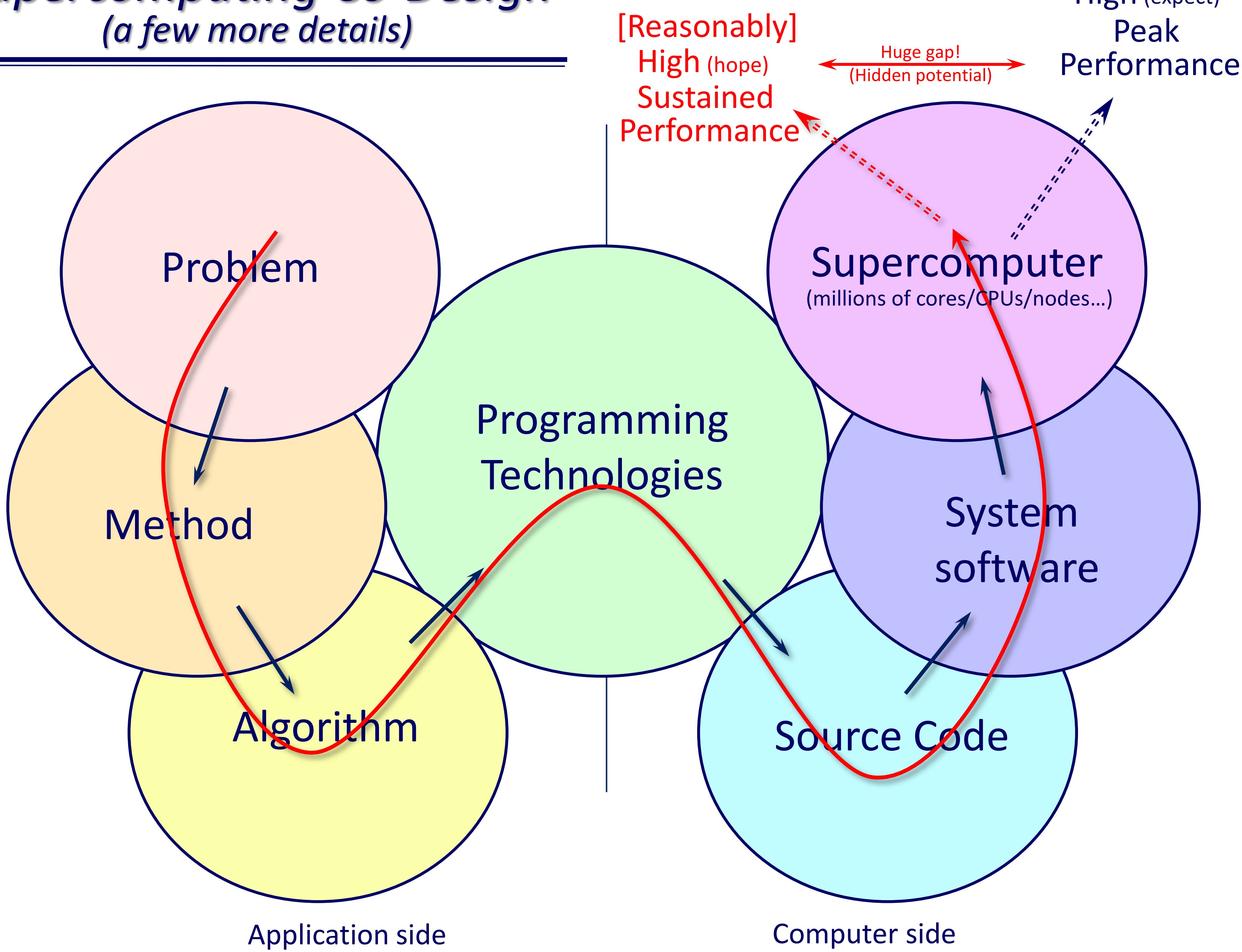


Variety of Algorithms, Applications, Architectures

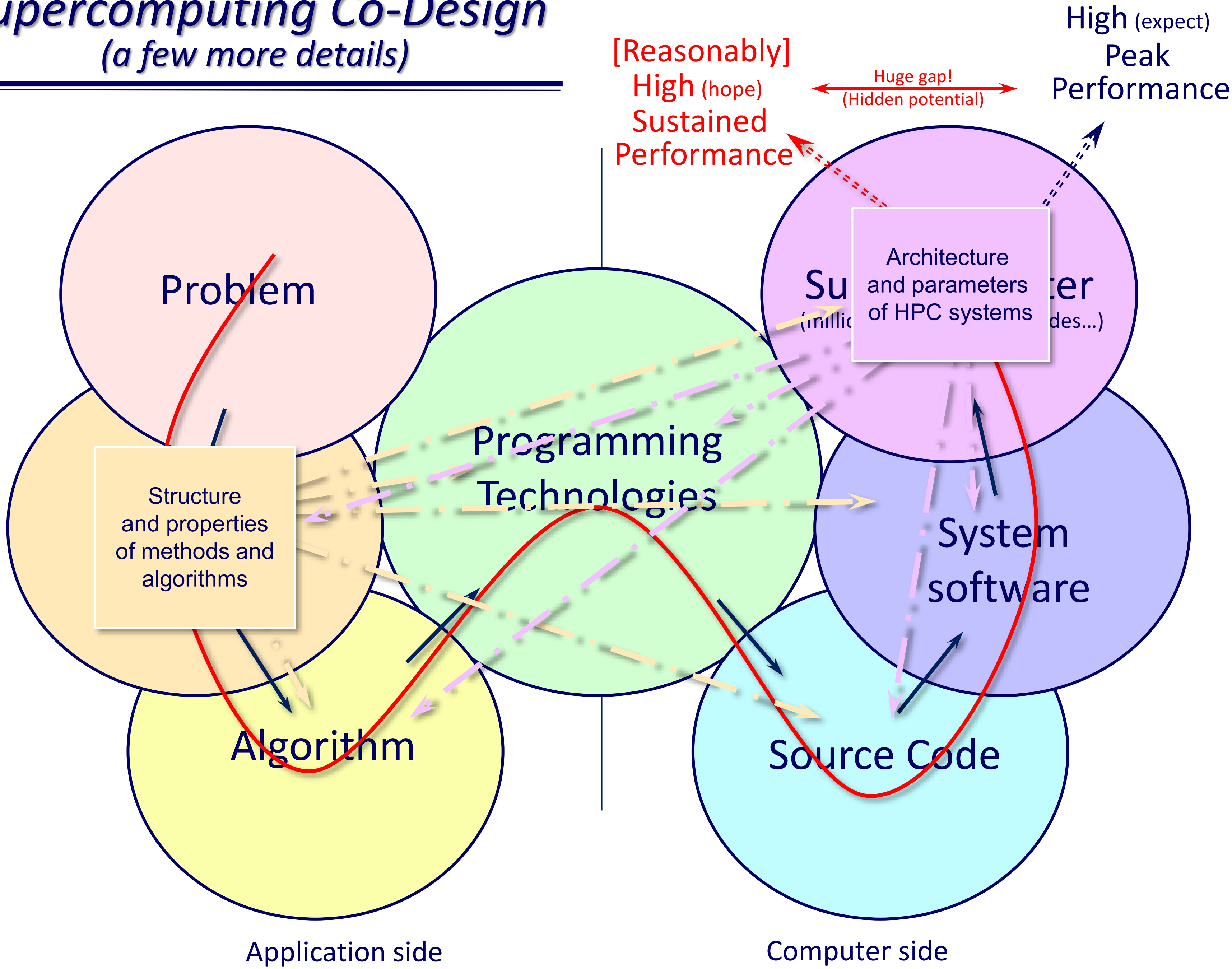
(MSU Supercomputing Center)



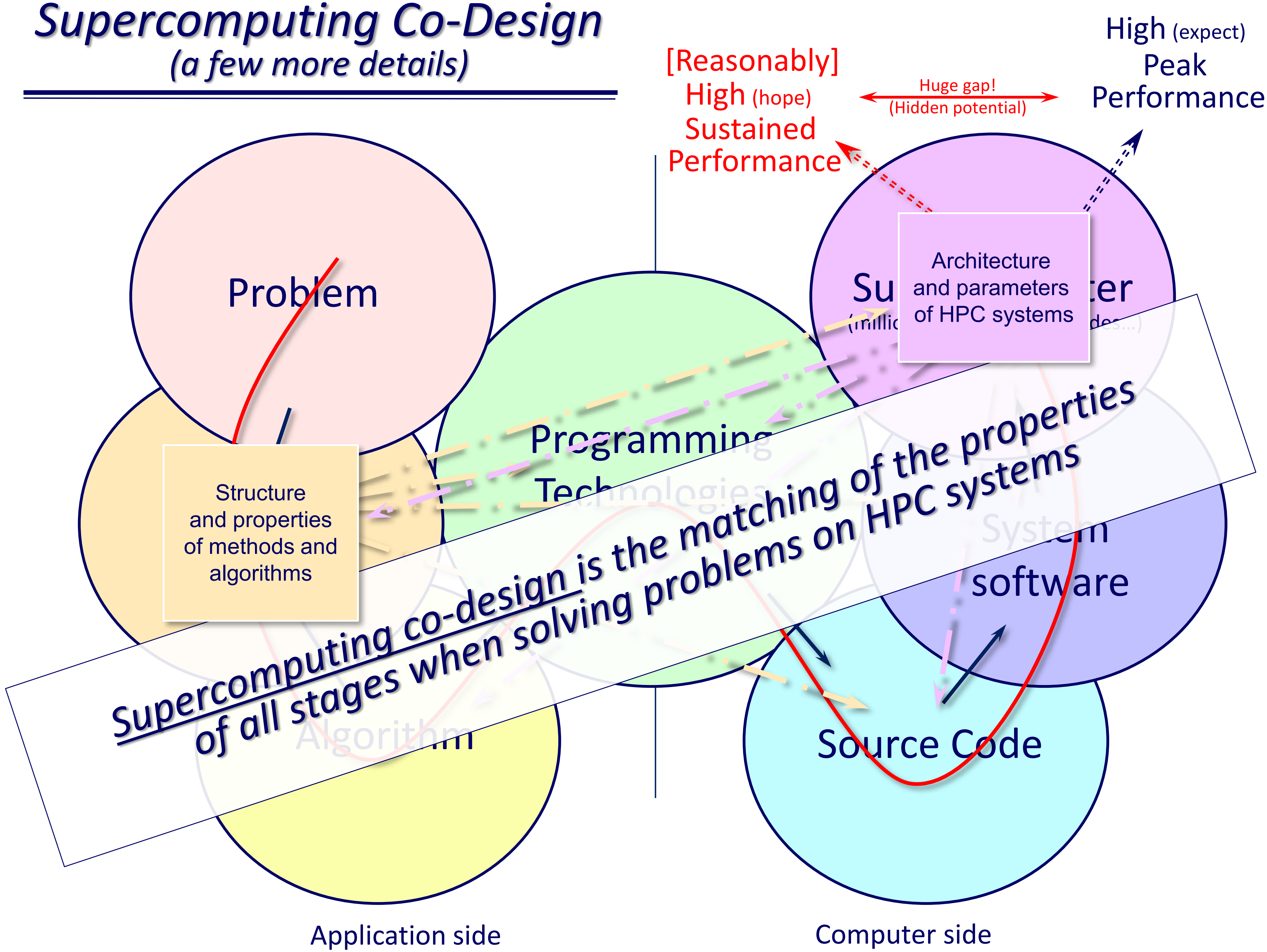
Supercomputing Co-Design
(a few more details)



Supercomputing Co-Design
(a few more details)

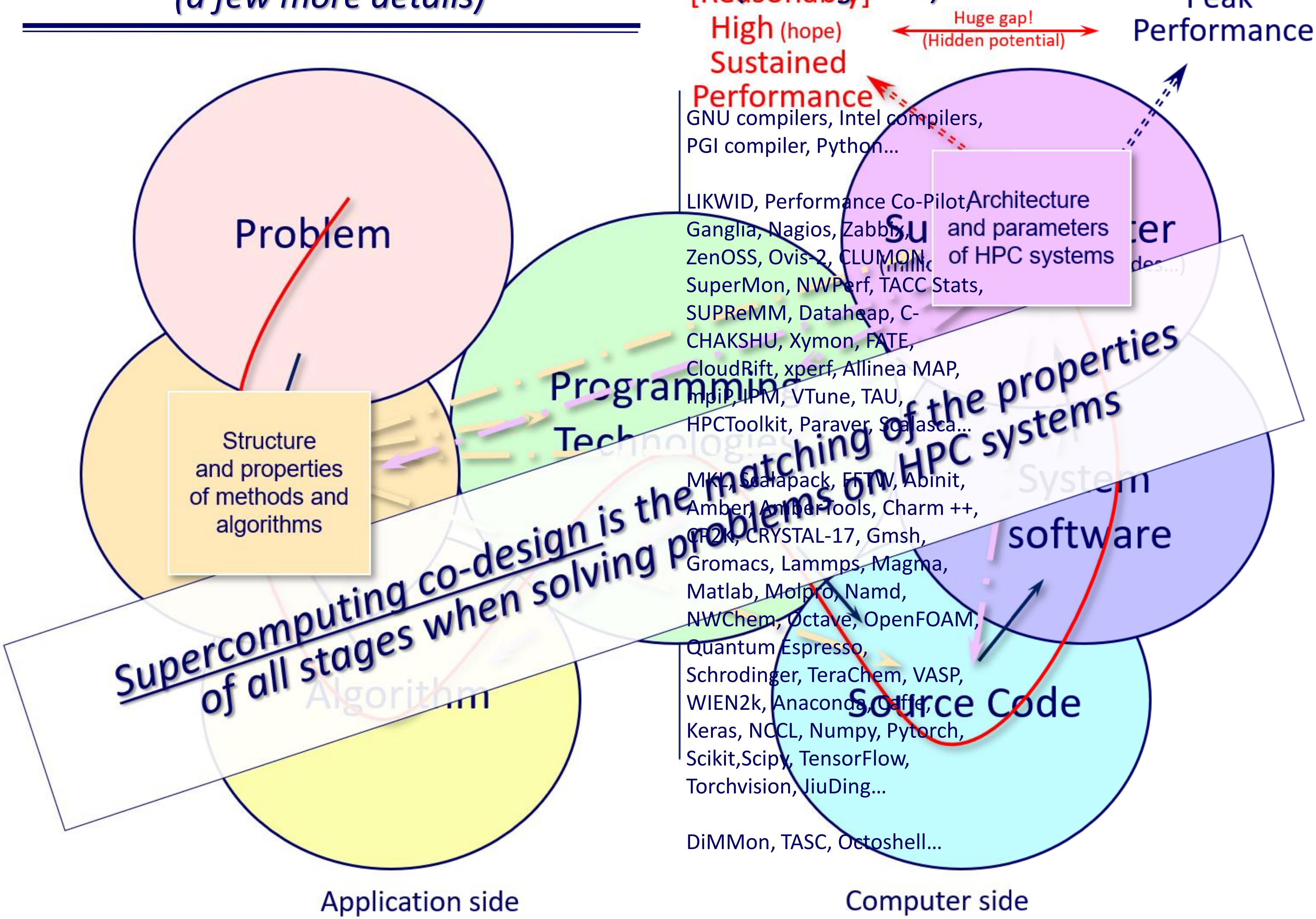


Supercomputing Co-Design
(a few more details)



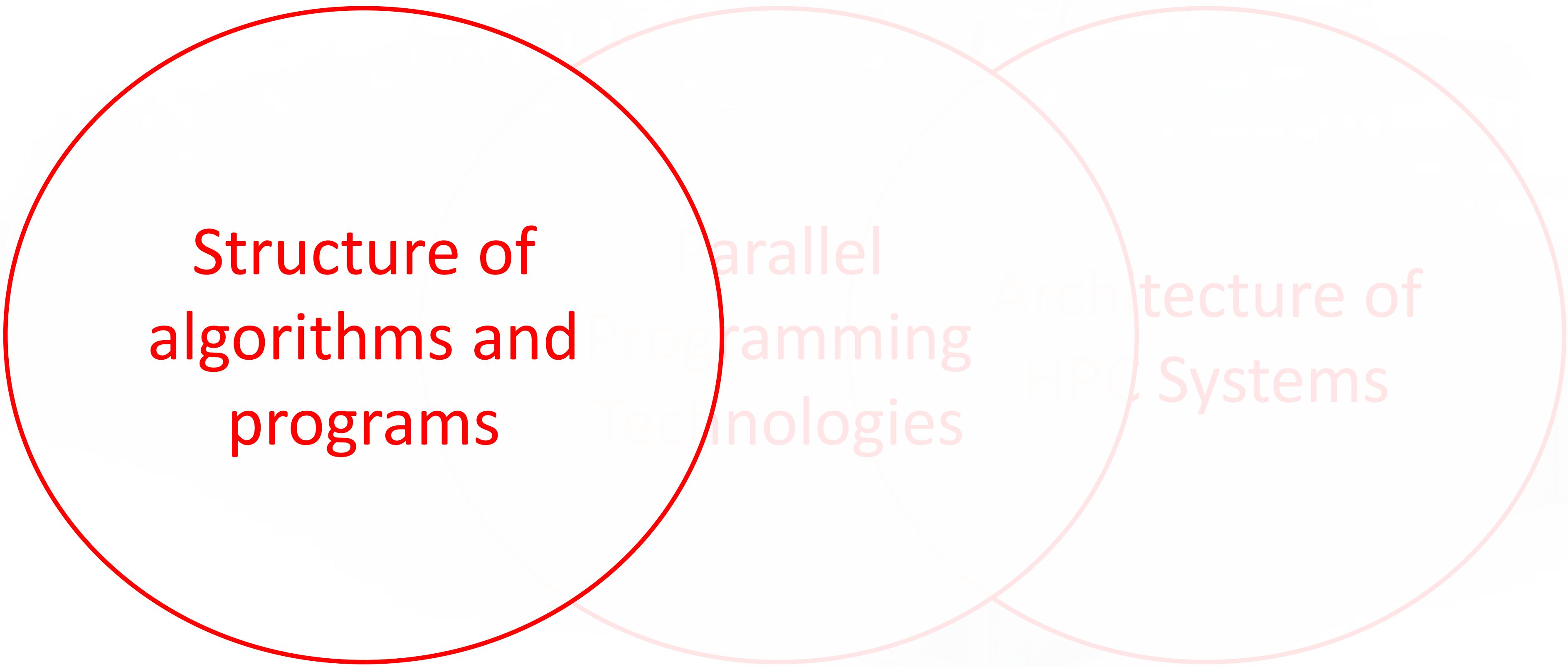
Supercomputing Co-Design
(a few more details)

Supercomputing Co-Design
(existing tools)



Supercomputing Co-Design

(detailed description of the three stages: structure of algorithms)



The AlgoWiki Project:
description of parallel structure and properties of algorithms

Open Encyclopedia of Parallel A

algowiki-project.org/en/Open_Encyclopedia_of_Parallel_Algorithmic_Features

AlgoWiki

Page | Discussion | Read | View source | View history | Search Algowiki

Main page
Forum
Recent changes

File storage

New files
Upload file

Tools

What links here
Related changes
Special pages
Printable version
Permanent link
Page information

In other languages

Русский

Open Encyclopedia of Parallel Algorithmic Features

Welcome! Join us!

AlgoWiki is an open encyclopedia of **algorithms' properties** and **features of their implementations** on different hardware and software platforms from mobile to extreme scale, which allows for collaboration with the worldwide computing community on algorithm descriptions.

AlgoWiki provides an exhaustive description of an algorithm. In addition to classical algorithm properties such as serial complexity, AlgoWiki also presents additional information, which together provides a complete description of the algorithm: its parallel complexity, parallel structure, determinacy, data locality, performance and scalability estimates, communication profiles for specific implementations, and many others.

Read more: About project.

Project structure

Algorithm classification — the main section of AlgoWiki which contains descriptions of all algorithms. Algorithms are added to the appropriate category of the classification, and classification is expanded with new sections if necessary.

Featured article

Cholesky decomposition

1 Properties and structure of the algorithm

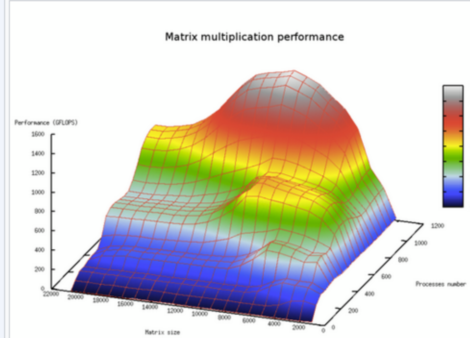
1.1 General description

The Cholesky decomposition algorithm was first proposed by Andre-Louis Cholesky (October 15, 1875 - August 31, 1918) at the end of the First World War shortly before he was killed in battle. He was a French military officer and mathematician. The idea of this algorithm was published in 1924 by his fellow officer and, later, was used by Banachiewicz in 1938 [7]. In the Russian mathematical literature, the Cholesky decomposition is also known as the square-root method [1-3] due to the square root operations used in this

Properties of the algorithm:

- Sequential complexity: $O(n^3)$
- Height of the parallel form: $O(n)$
- Width of the parallel form: $O(n^2)$
- Amount of input data: $\frac{n(n+1)}{2}$
- Amount of output data: $\frac{n(n+1)}{2}$

Today's featured picture



Performance for dense matrix multiplication

Work organization

Description of algorithm properties and structure

Guides to writing sections of the algorithm's description

Glossary

Help with editing

Readiness of articles

Finished articles:

- Gaussian elimination, compact scheme for tridiagonal matrices, serial version
- Householder (reflections) method for reducing of a matrix to

http://AlgoWiki-Project.org/en

Description of Algorithms

(What properties of algorithms should be included in the description?)

AlgoWiki:

I. Algorithms: Theoretical Part

(machine-independent properties and theoretical potential of algorithms ***"Once and for all" **** *)*

II. Algorithms: Implementation Issues

Information Graph
Determinacy
Computational kernel
Macrostructure
Locality of computations
Performance
Scalability
Data locality
Mathematical description
Communication profile
Serial Complexity
Efficiency
Resource of Parallelism
Computational intensity
Input / Output data

** Changes in architectures require changes in implementations, not in algorithms!*

http://AlgoWiki-Project.org/en

Description of Algorithms
(What properties of algorithms should be included in the description?)

For positive definite Hermitian matrices (*symmetric matrices in the real case*), we use the decomposition $A = LL^*$, where L is the lower triangular mat

General Description

ion $A = U^*U$, where U is the upper triangular matrix. These forms of the Cholesky decomposition are equivalent in the sense of the amount of arithmetic operations and are different in the sense of data represntation.

ed uniquely for the elements of the matrix L from

features.

Input data: a symmetric positive definite matrix A whose elements are denoted by a_{ij}).

Output data: the lower triangular matrix L whose elements are denoted by l_{ij}).

The Cholesky algorithm can be represented in the for

Mathematical Description

$$l_{11} = \sqrt{a_{11}},$$

$$l_{j1} = \frac{a_{j1}}{l_{11}}, \quad j \in [2, n],$$

$$l_{ii} = \sqrt{a_{ii} - \sum_{p=1}^{i-1} l_{ip}^2}, \quad i \in [2, n],$$

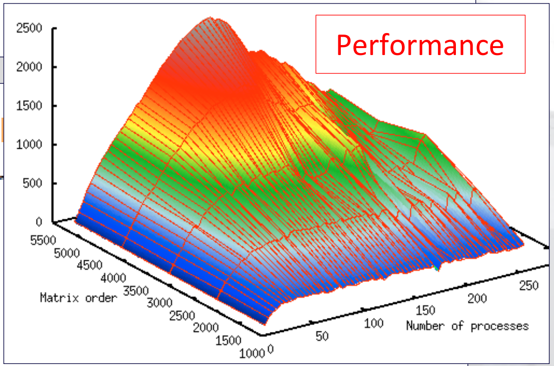
$$l_{ji} = \left(a_{ji} - \sum_{p=1}^{i-1} l_{ip} l_{jp}\right) / l_{ii}, \quad i \in [2, n-1], j \in [i+1, n].$$

The following number of operations should

matrix of order n using a serial version of th

- n square roots,
- $\frac{n(n-1)}{2}$ divisions,
- $\frac{n^3-n}{6}$ multiplications and $\frac{n^3-n}{6}$ additions (subtractions): the main amount of computational wor

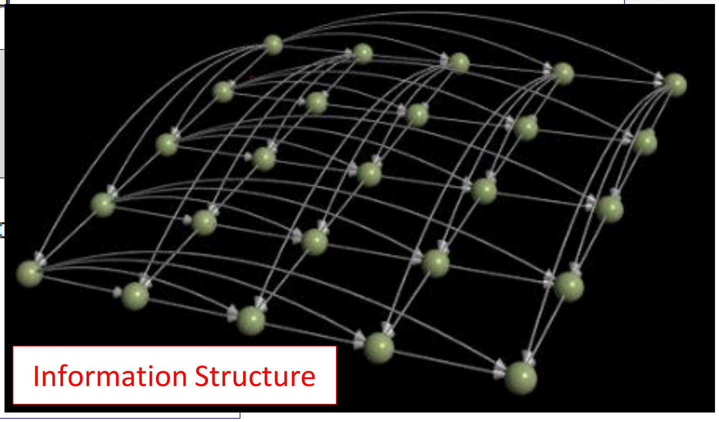
Serial Complexity



A computational kernel of its serial version can be composed of $\frac{n(n-1)}{2}$ dot products o

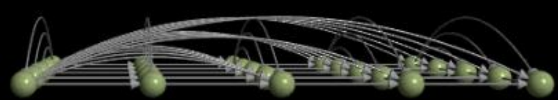
$$\sum_{p=1}^{i-1} l_{ip} l_{jp}.$$

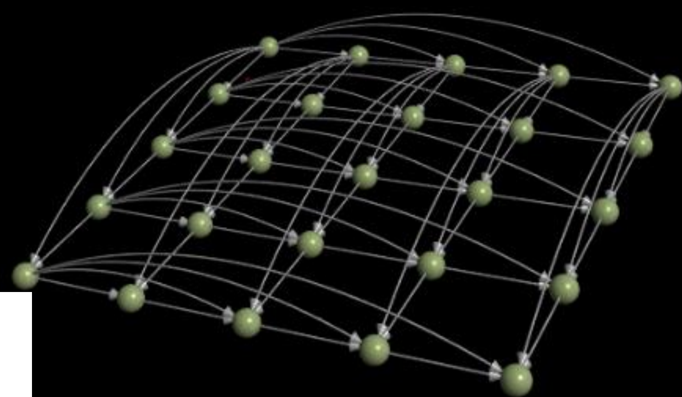
Computational Kernel

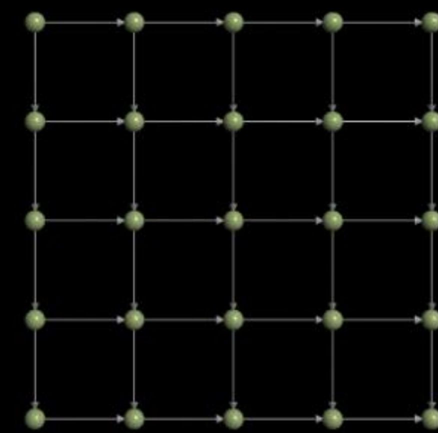


Information Structure of Algorithms

(one of the key elements for supercomputing co-design)





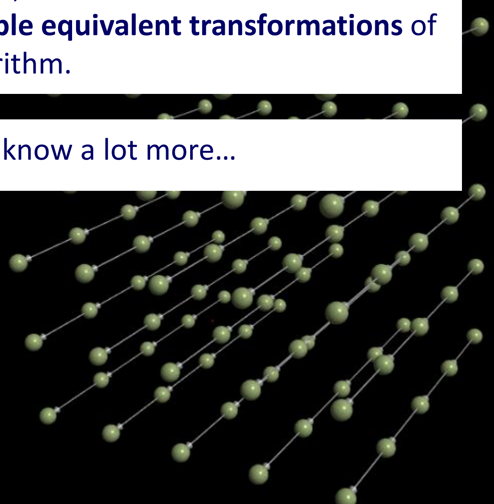


Typical algorithmic structures

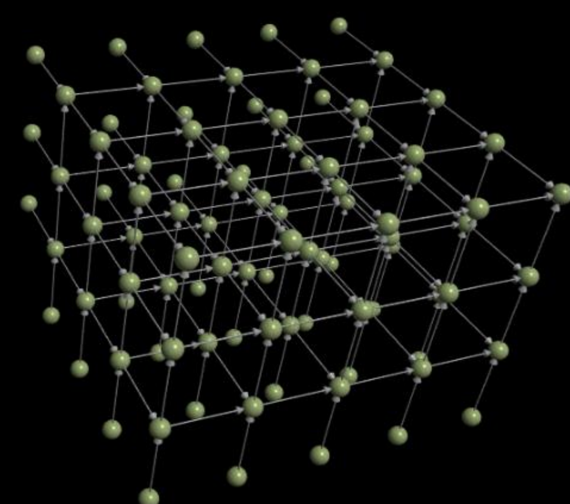
If you know information structure of an algorithm then you know its two the most important properties:

- the **total resource of parallelism** of the algorithm,
- all **possible equivalent transformations** of the algorithm.

In fact, you know a lot more…



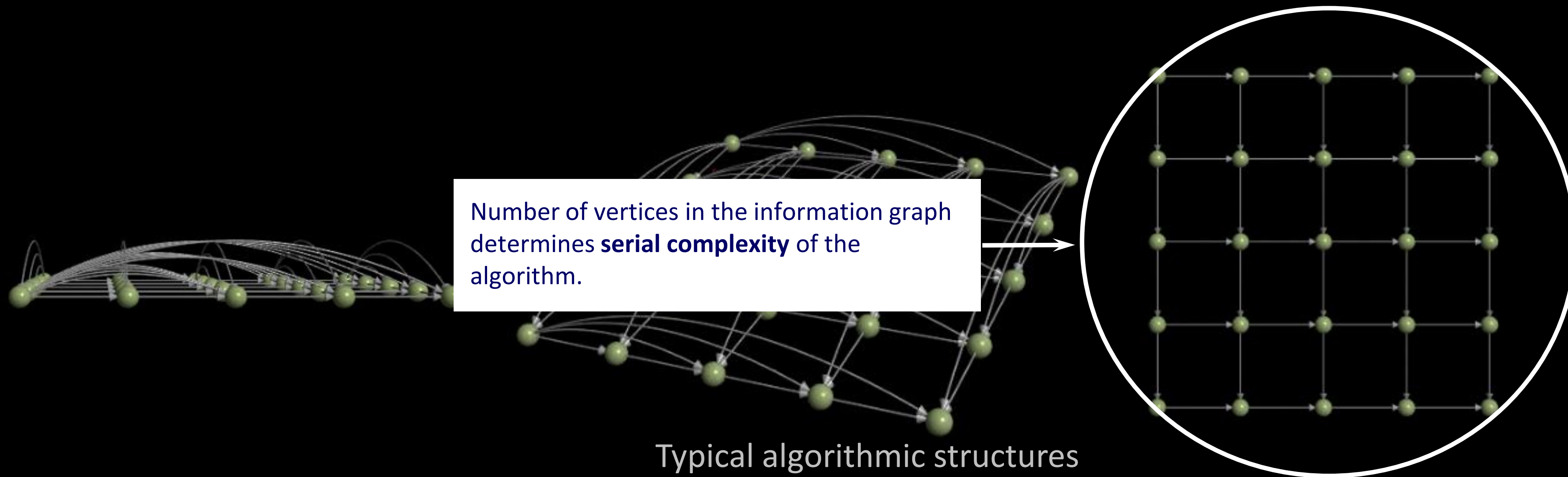
Huge resource of parallelism in the algorithms



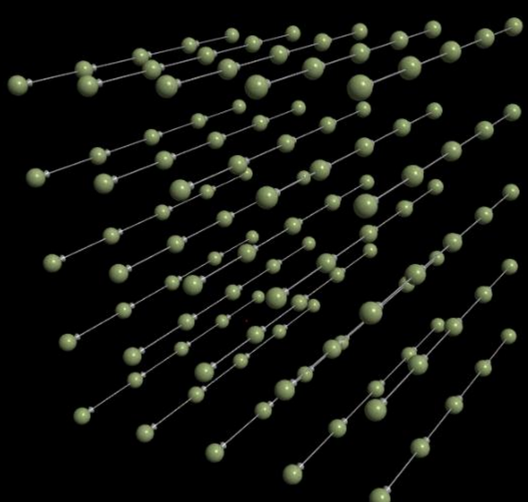
Skewed parallelism

Information Structure of Algorithms

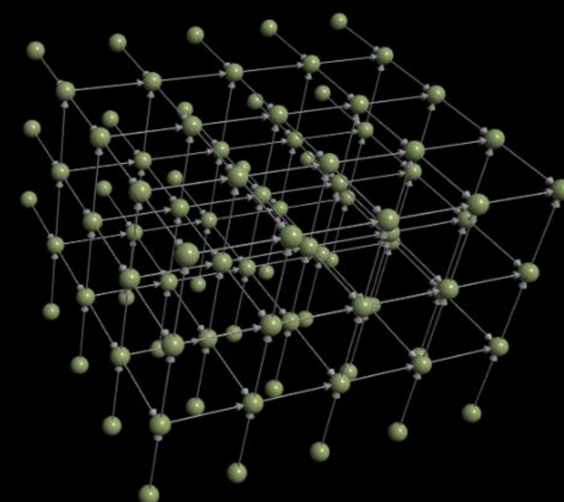
(one of the key elements for supercomputing co-design)



Typical algorithmic structures



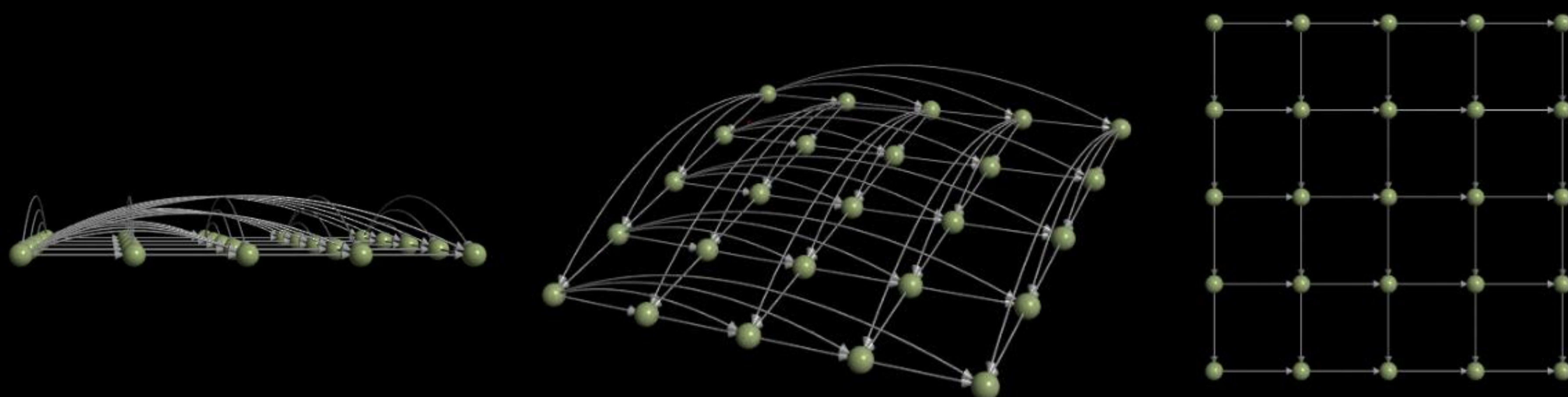
Huge resource of parallelism in the algorithms



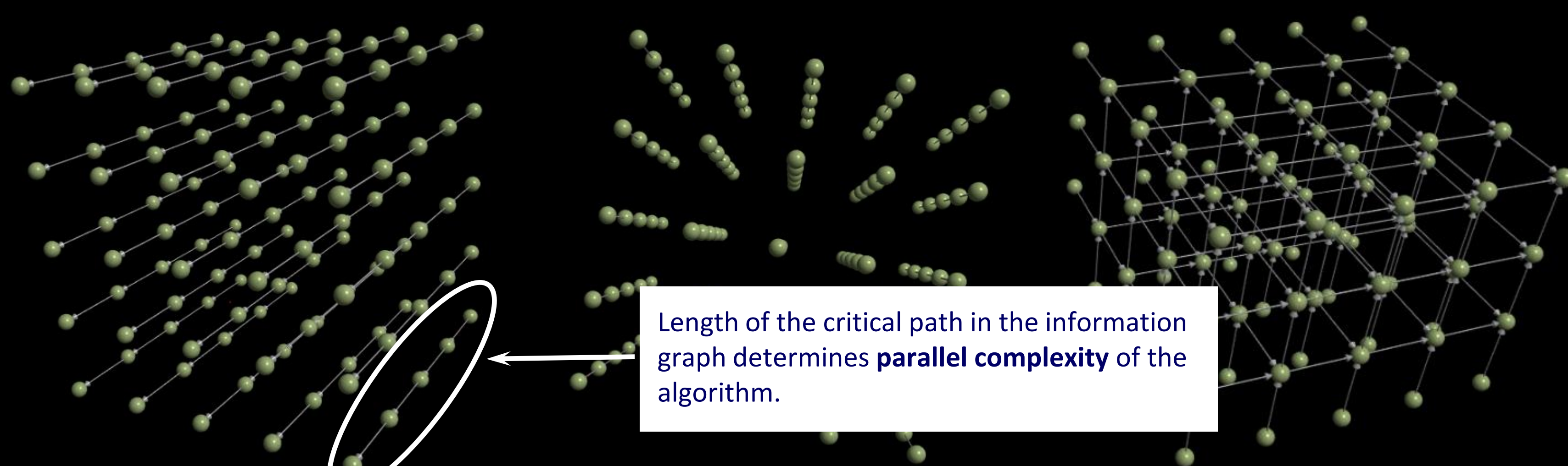
Skewed parallelism

Information Structure of Algorithms

(one of the key elements for supercomputing co-design)



Typical algorithmic structures

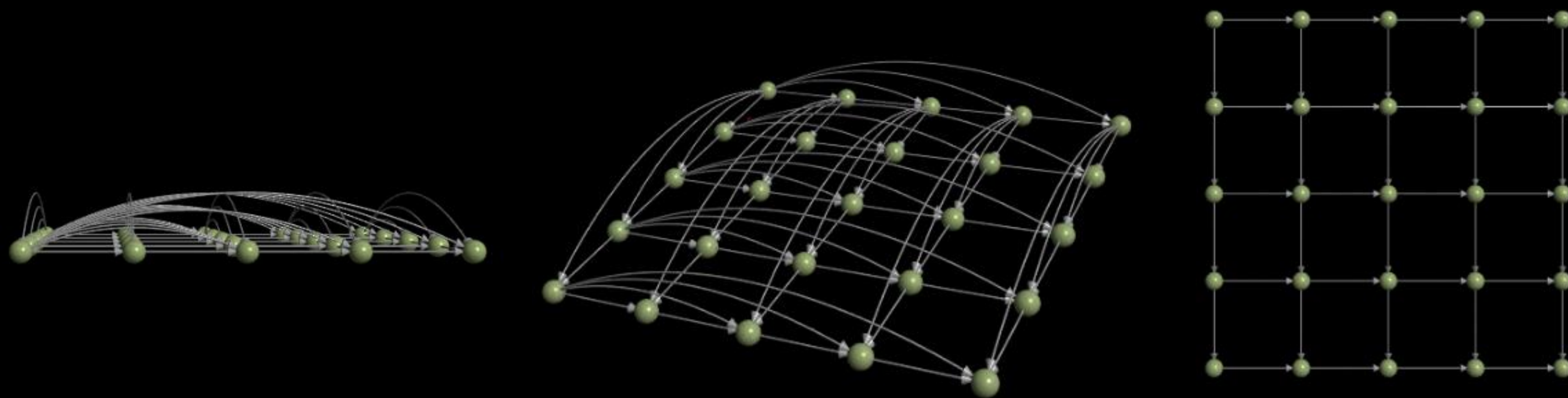


Huge resource of parallelism in the algorithms

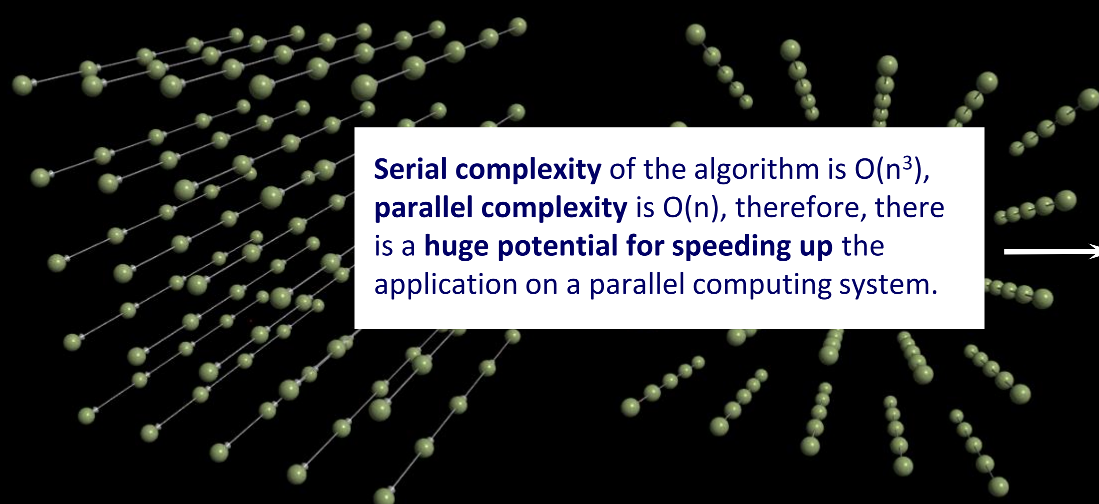
Skewed parallelism

Information Structure of Algorithms

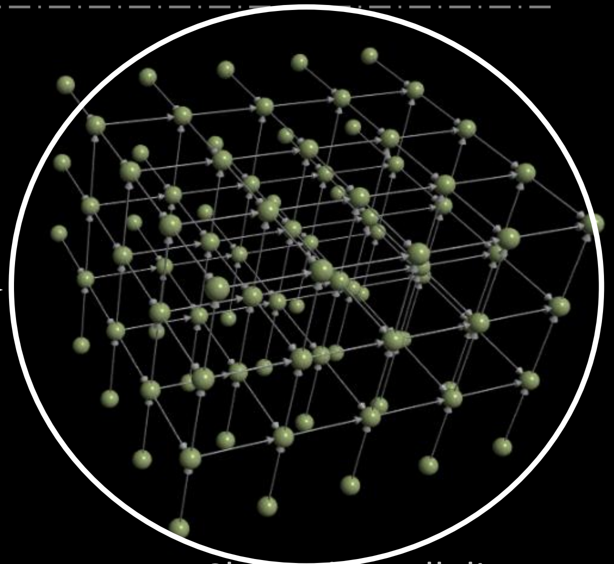
(one of the key elements for supercomputing co-design)



Typical algorithmic structures



Serial complexity of the algorithm is O(n³), **parallel complexity** is O(n), therefore, there is a **huge potential for speeding up** the application on a parallel computing system.

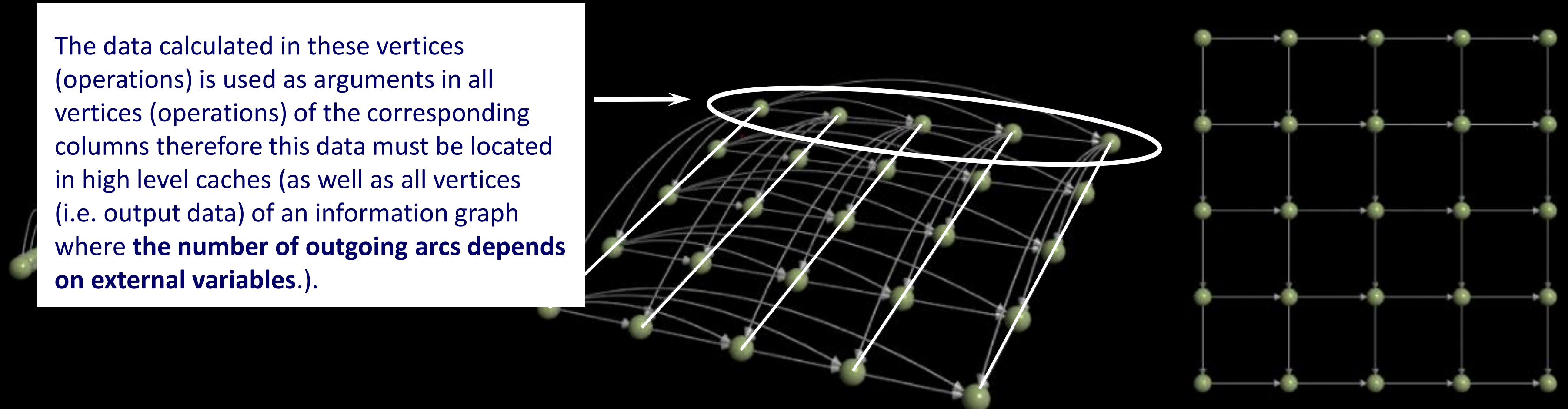


Huge resource of parallelism in the algorithms

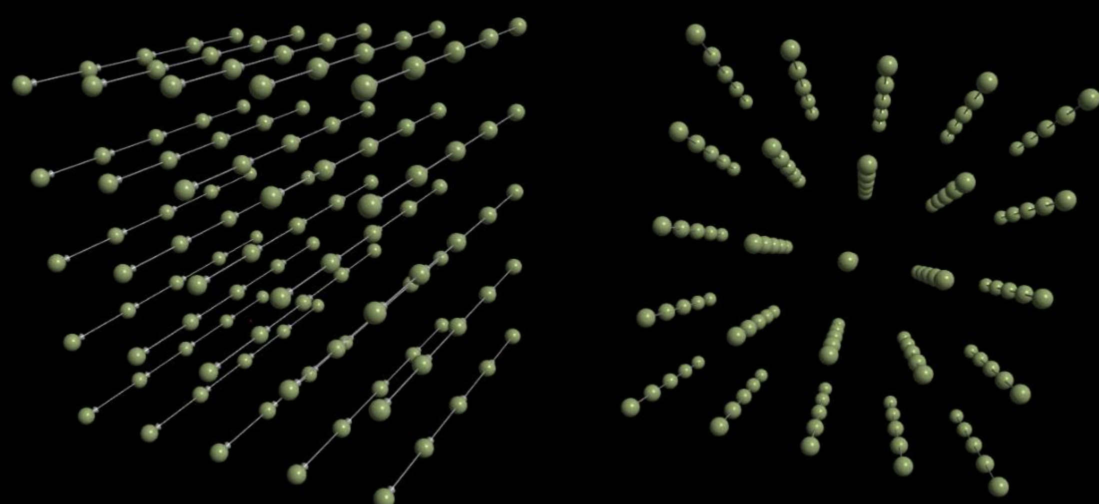
Skewed parallelism

Information Structure of Algorithms

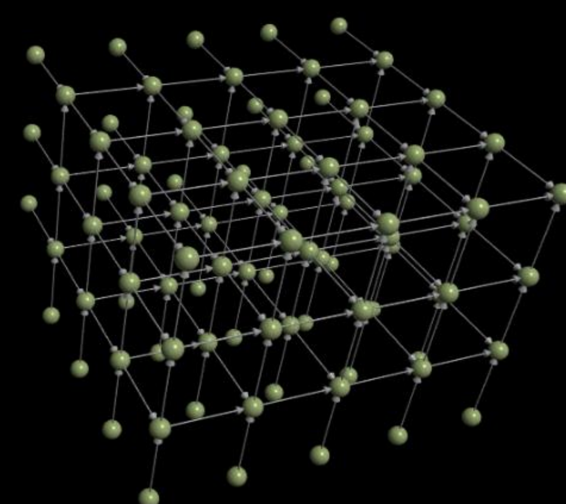
(one of the key elements for supercomputing co-design)



Typical algorithmic structures



Huge resource of parallelism in the algorithms



Skewed parallelism

Description of Algorithms

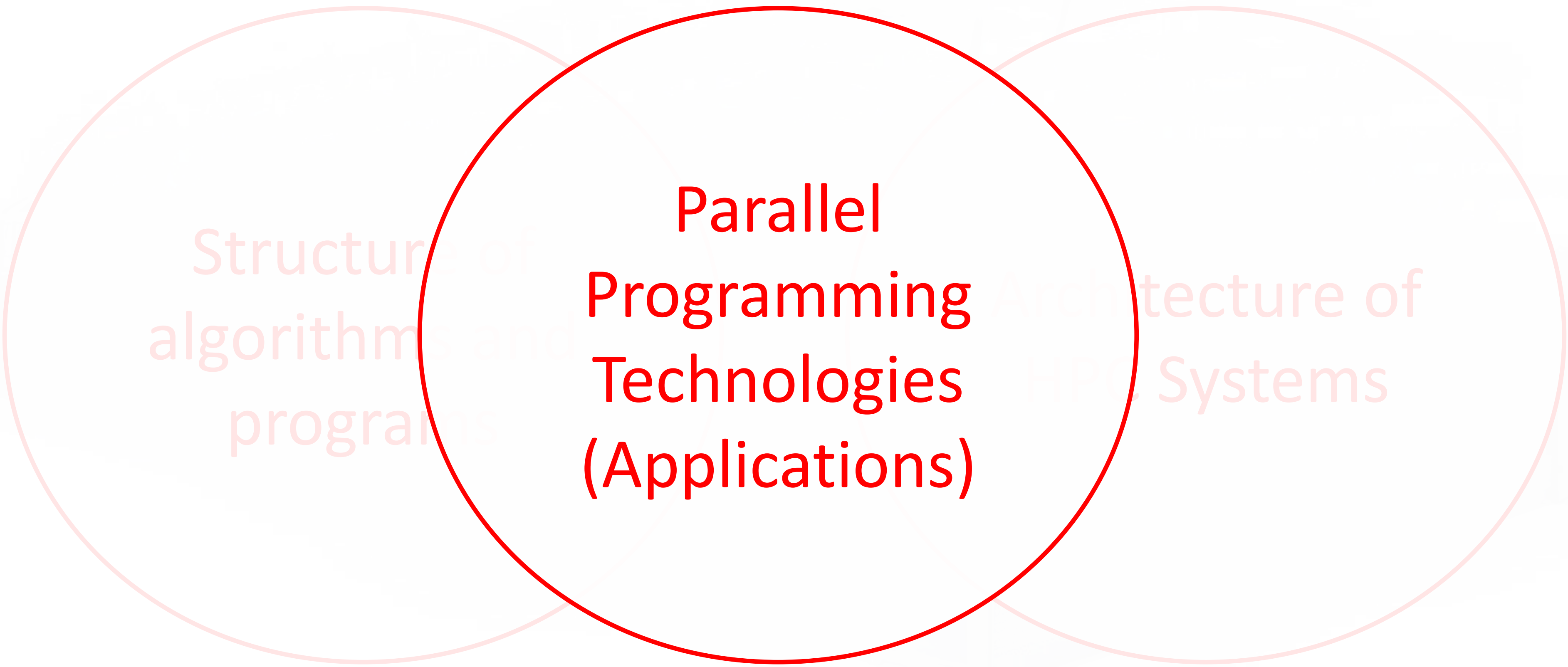
(practice says: this is a complete description and it is suitable for any algorithm)

1. General description
2. Mathematical description
3. Computational kernel
   (operations, data format, data structures)
4. Macrostructure of algorithm
   (macrooperations, typical algorithmic structures)
5. General scheme of serial algorithm
6. Complexity of serial algorithm
   (arithmetic operations, read/write operations)
7. Information graph
8. Resource of parallelism
   (complexity of parallel algorithm, available parallelism)
9. Input and output data
10. Properties of algorithm
    (computational intensity, stability, determinism, imbalance…)
11. Data locality, locality of computations
12. Possible approaches to parallel implementations
    (features, properties, programming technologies…)
13. Possible obstacles for scalability
    (limited resource of parallelism, load imbalance, synchronizations…)
14. Dynamic properties and parameters: features of parallel implementations
15. Existing implementations

Practice has shown that
the proposed structure is complete,
and all the properties of any algorithm
necessary for its effective parallel
implementation can be
fully described using this structure.

Supercomputing Co-Design

(detailed description of the three stages: structure of applications)



Structure and Properties of Applications
(partially combined with the proposed description of algorithms)

- Partially combined with the proposed description of the algorithms:

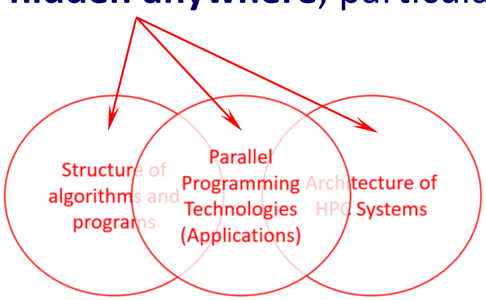
  …

  11. Data locality, locality of computations
  12. Possible approaches to parallel implementations
      (features, properties, programming technologies…)
  13. Possible obstacles for scalability
      (limited resource of parallelism, load imbalance, synchronizations…)
  14. Dynamic properties and parameters: features of parallel implementations
  15. Existing implementations

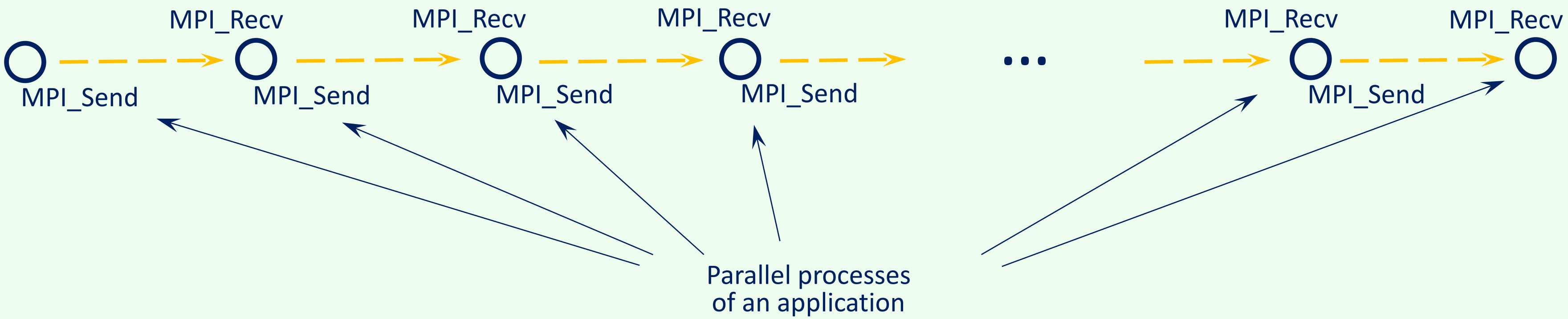
- There are many concepts associated with parallel applications that are important for supercomputing co-design, e.g. “loop profile”, “data locality”, “communication profile”, “scalability”, etc…

- **Concept of scalability** is particularly important since it is tightly related to potential speedup. At the same time there are dozens of **reasons that limit scalability of applications** or even make their scaling impossible. These reasons **may be hidden anywhere**, particularly at the level of algorithms, applications, or architectures, e.g.:

algorithms – limited resource of parallelism,
applications – unnecessary synchronization,
architecture – using high latency interconnect.



Scalability of Applications
(implicit or unnecessary synchronization)

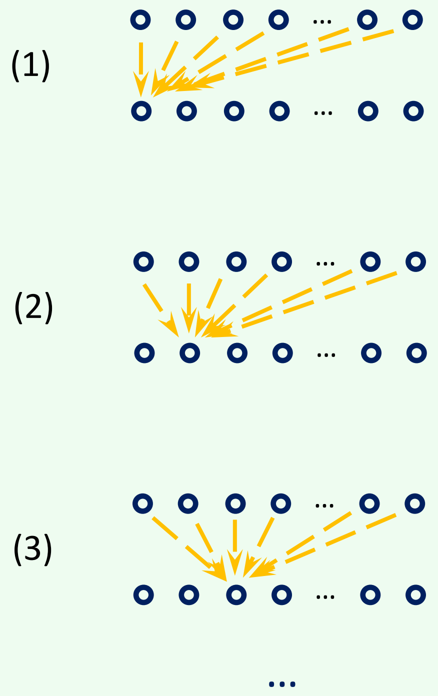


Possible solution:

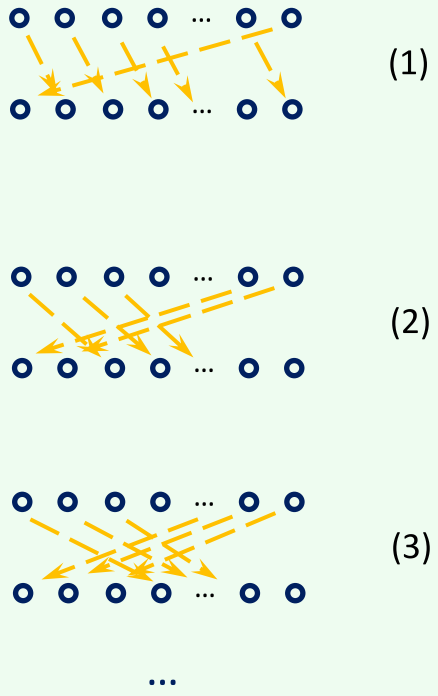
MPI_Recv → MPI_Irecv

MPI_Send → MPI_Isend

Scalability of Applications
(bad implementation for a scalable operation)



The original non-scalable version

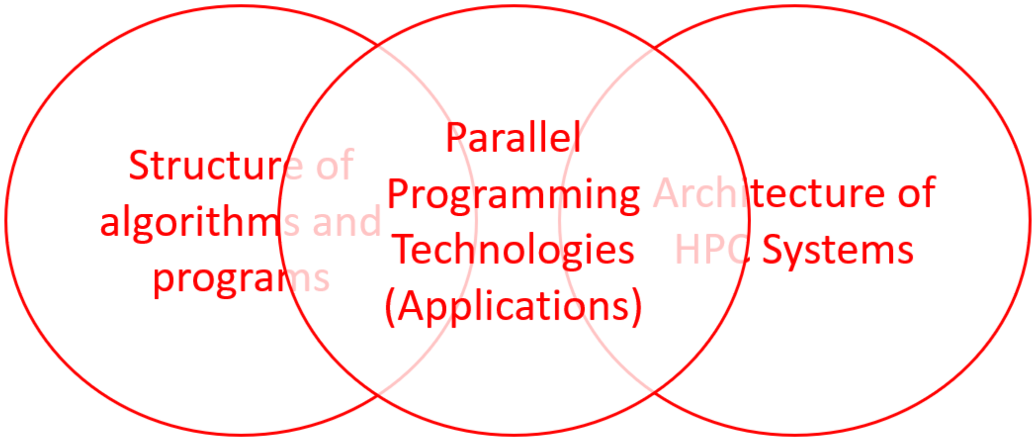


Possible corrected version

Scalability and Supercomputing Co-Design
(scalability through the three stages)

We actively use the concept of scalability, but we don't know everything about it...

- What are the similarities and fundamental differences between the concepts of "algorithm scalability", "application scalability" and "architecture scalability"?



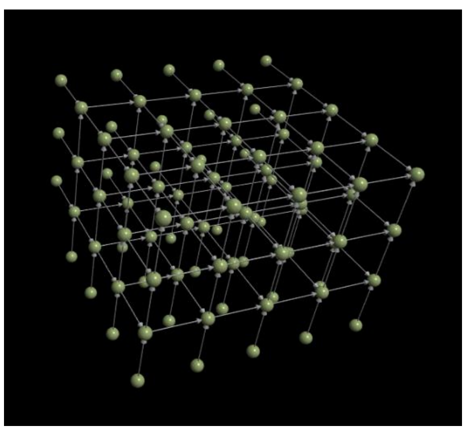
- How can we use a **common element for all co-design stages, i.e. scalability**, as a part of supercomputing co-design technologies?
- Is it possible to make the concepts of "algorithm scalability", "application scalability" and "architecture scalability" comparable to each other?

Common Elements for All Co-Design Stages
(scalability is not the only one)

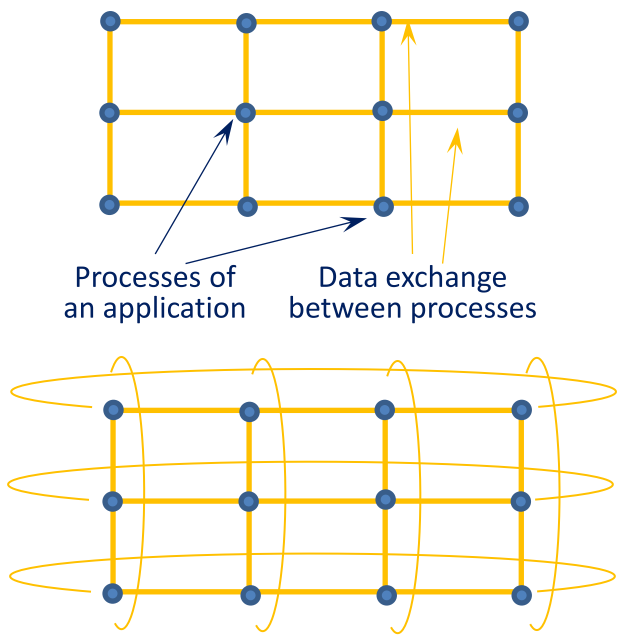
Common elements for all three stages of co-design are needed to be deeply investigated since they lead to development of effective supercomputing co-design technologies.

Algorithms ⟷ **Applications** ⟷ **Architectures**

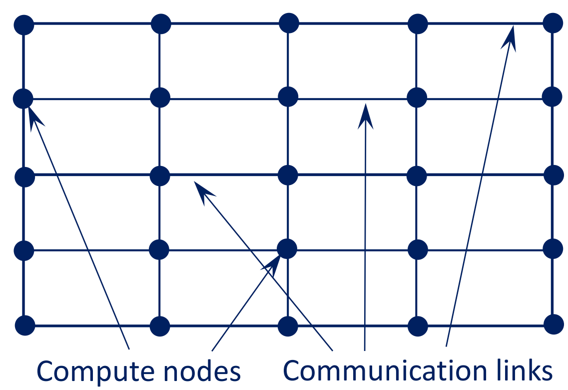
"Information structure of algorithms"



"Communication profile of parallel applications "



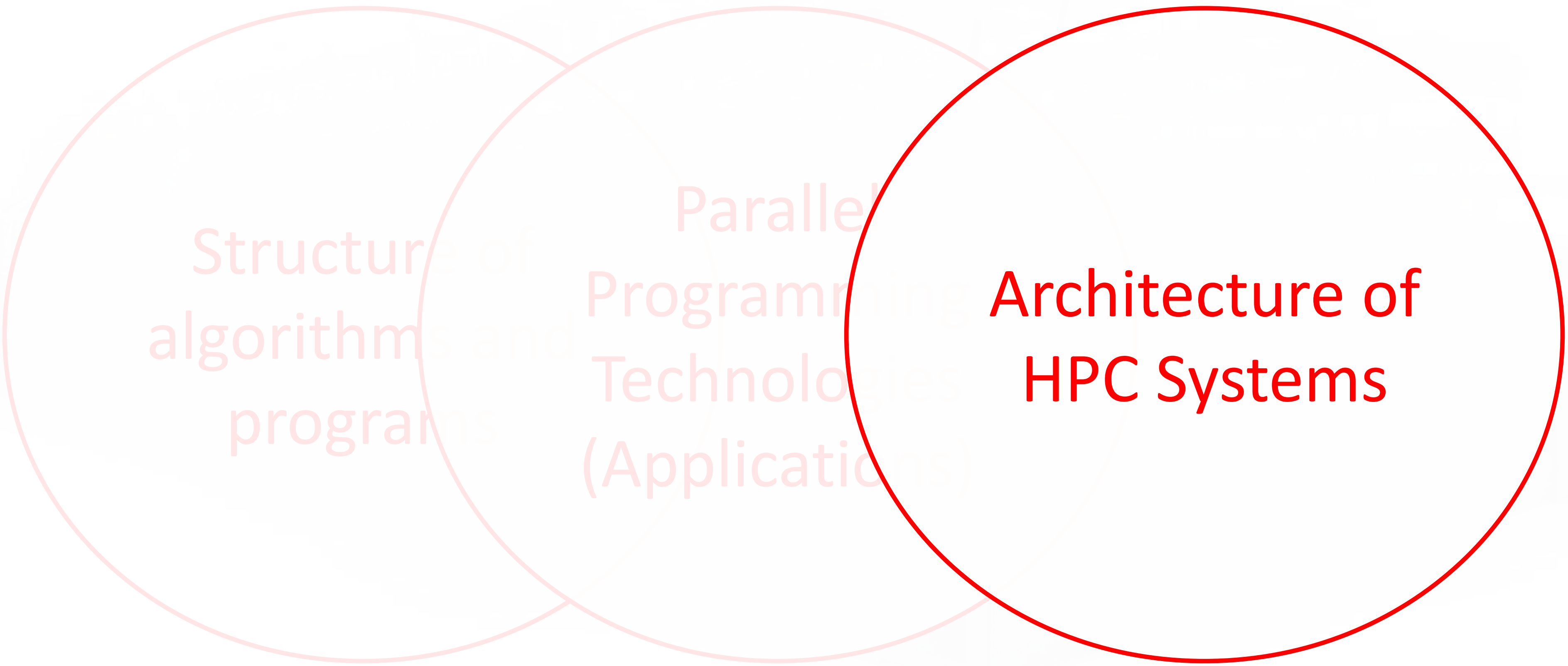
"Topology of communication network "



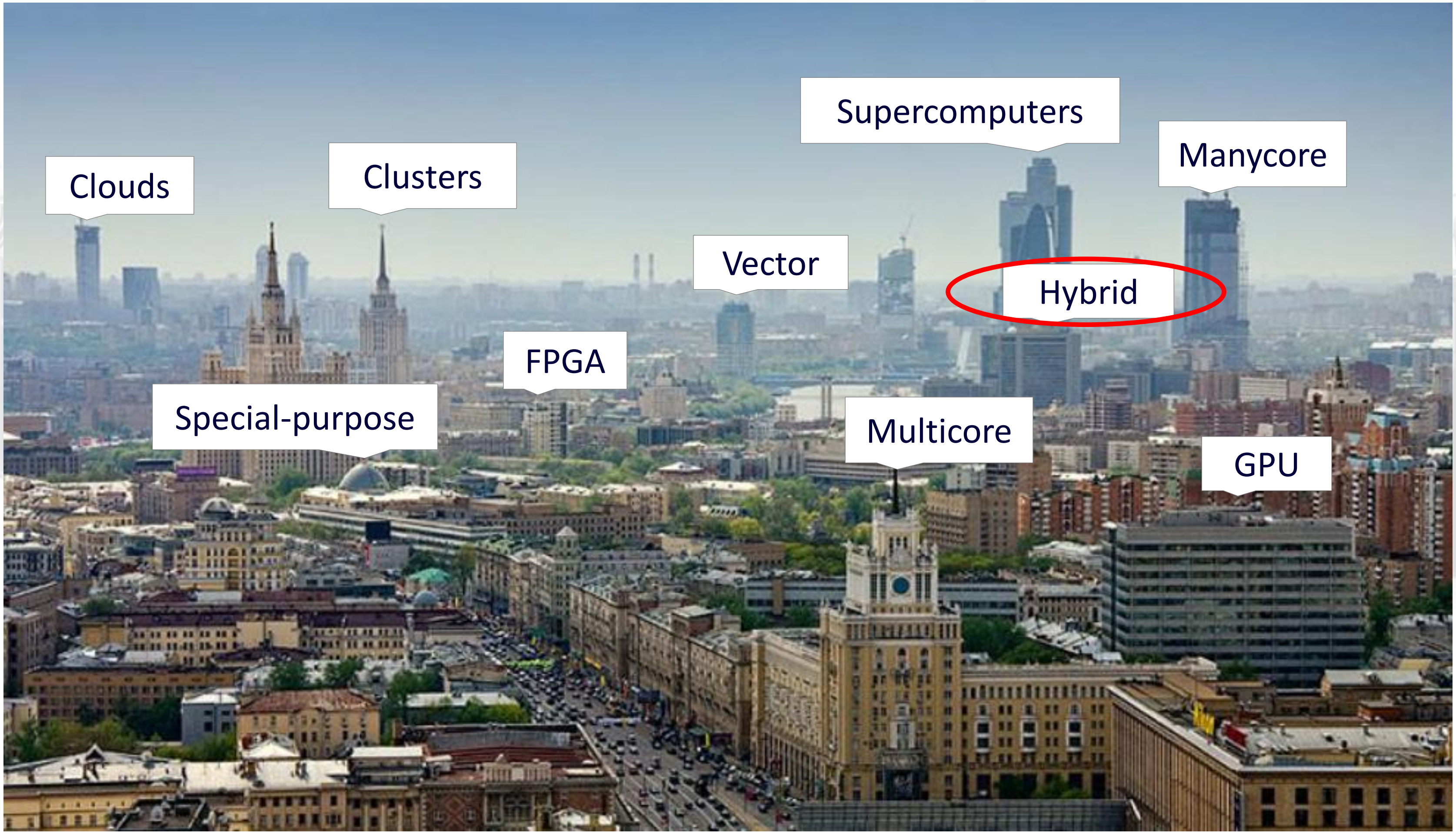
What methods can be used to analyze these concepts together?

Supercomputing Co-Design

(detailed description of the three stages: computer architectures)



Computer architecture landscape today
(How diverse is a computer world today?)



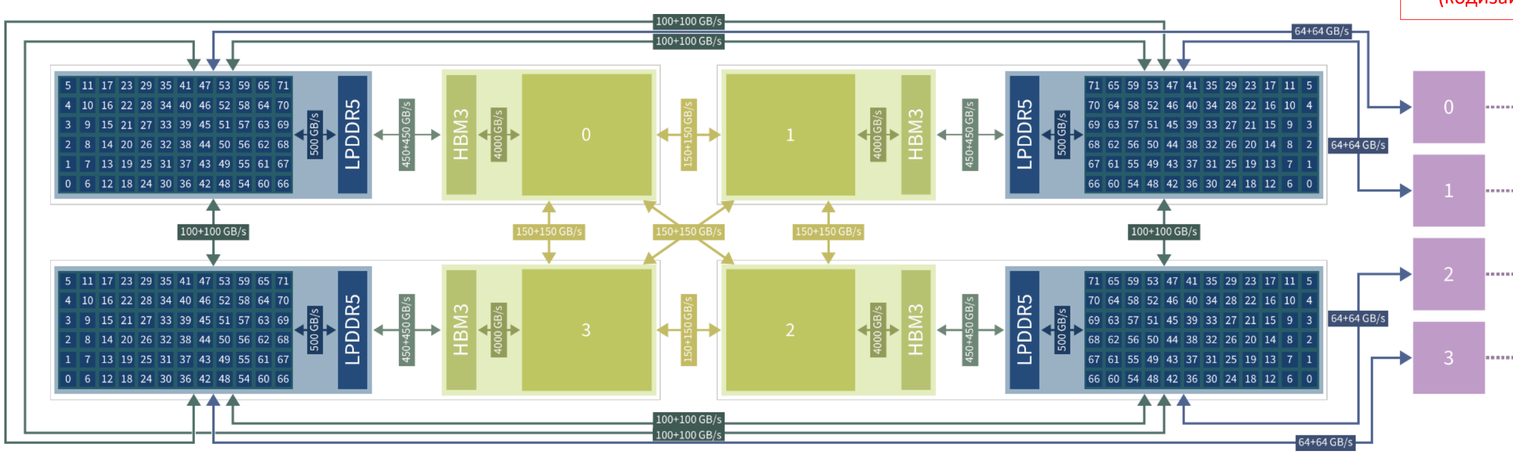
We should think about the choice of computer platforms that match properties of algorithms…

Суперкомпьютер Jupiter, Germany, Julich
(кодизайн – это непросто)

Jupiter – первый суперкомпьютер класса Exascale в Европе. Программа EuroHPC JU.
Стоимость – 500 млн.евро, включая расходы на приобретение и эксплуатацию.

Jupiter, две основные части: **Booster** и **Cluster**.
Booster: 6000 выч.узлов, 1 Eflops (fp64, HPL), 70+ Eflops (8-bits),
  Выч.узел – 4 * NVIDIA Grace-Hopper; 4 * InfiniBand NDR (200 Gbit/s),
  (4 * CPU Grace (ARM, 72 cores, LPDDR5X, 500GB/s)) * CPU NVLink (100GB/s bi-directional bw)
  (4 * GPU Hopper H100, HBM3, 4TB/s) * NVLink 4 (150GB/s per direction)
Cluster: 1300 выч.узлов, 5 Pflops (fp64, HPL),
  Выч.узел – 2 * (CPU Rheal, 80 ARM cores), 2*64GB HBM+512GB DDR5; InfiniBand NDR (200 Gbit/s)
Interconnect: InfiniBand NDR (200 Gbit/s), DragonFly+ groups (FatTree inside each group) for Booster, Cluster, storage, administrative infrastructure.

Linpack, КПД = 85% !
(кодизайн – это непросто, но решаемо)



Booster: структура вычислительного узла

The properties of algorithms determine computer architecture

1. General description
2. Mathematical description
3. Computational kernel
   (operations, data format, data structures)
4. Macrostructure of algorithm
   (macrooperations, typical algorithmic structures)
5. General scheme of serial algorithm
6. Complexity of serial algorithm
   (arithmetic operations, read/write operations)
7. Information graph
8. Resource of parallelism
   (complexity of parallel algorithm, available parallelism)
9. Input and output data
10. Properties of algorithm
    (computational intensity, stability, determinism, imbalance…)
11. Data locality, locality of computations
12. Possible approaches to parallel implementations
    (features, properties, programming technologies…)
13. Possible obstacles for scalability
    (limited resource of parallelism, load imbalance, synchronizations…)
14. Dynamic properties and parameters: features of parallel implementations
15. Existing implementations

If computational kernel is data intensive then a large number of memory channels could be more important than number of computing cores.

The properties of algorithms determine computer architecture

1. General description
2. Mathematical description
3. Computational kernel
   (operations, data format, data structures)
4. Macrostructure of algorithm
   (macrooperations, typical algorithmic structures)
5. General scheme of serial algorithm
6. Complexity of serial algorithm
   (arithmetic operations, read/write operations)
7. Information graph → Information structure leads to a certain communication profile of an application. The profile should match a topology of communication network.
8. Resource of parallelism
   (complexity of parallel algorithm, available parallelism)
9. Input and output data
10. Properties of algorithm
    (computational intensity, stability, determinism, imbalance…)
11. Data locality, locality of computations
12. Possible approaches to parallel implementations
    (features, properties, programming technologies…)
13. Possible obstacles for scalability
    (limited resource of parallelism, load imbalance, synchronizations…)
14. Dynamic properties and parameters: features of parallel implementations
15. Existing implementations

The properties of algorithms determine computer architecture

1. General description
2. Mathematical description
3. Computational kernel
   (operations, data format, data structures)
4. Macrostructure of algorithm
   (macrooperations, typical algorithmic structures)
5. General scheme of serial algorithm
6. Complexity of serial algorithm
   (arithmetic operations, read/write operations)
7. Information graph
8. Resource of parallelism
   (complexity of parallel algorithm, available parallelism)
9. Input and output data
10. Properties of algorithm
   (computational intensity, stability, determinism, imbalance…)
11. Data locality, locality of computations
12. Possible approaches to parallel implementations
   (features, properties, programming technologies…)
13. Possible obstacles for scalability
   (limited resource of parallelism, load imbalance, synchronizations…)
14. Dynamic properties and parameters: features of parallel implementations
15. Existing implementations

Large resource of SIMD operations is the necessary but not sufficient condition for GPU using.

The properties of algorithms determine computer architecture

1. General description
2. Mathematical description
3. Computational kernel
   (operations, data format, data structures)
4. Macrostructure of algorithm
   (macrooperations, typical algorithmic structures)
5. General scheme of serial algorithm
6. Complexity of serial algorithm
   (arithmetic operations, read/write operations)
7. Information graph
8. Resource of parallelism
   (complexity of parallel algorithm, available parallelism)
9. Input and output data
10. Properties of algorithm
    (computational intensity, stability, determinism, imbalance…)
11. Data locality, locality of computations
12. Possible approaches to parallel implementations
    (features, properties, programming technologies…)
13. Possible obstacles for scalability
    (limited resource of parallelism, load imbalance, synchronizations…)
14. Dynamic properties and parameters: features of parallel implementations
15. Existing implementations

High computational intensity of the kernel is the necessary but not sufficient condition for GPU using.

The properties of algorithms determine computer architecture

1. General description
2. Mathematical description
3. Computational kernel
   (operations, data format, data structures)
4. Macrostructure of algorithm
   (macrooperations, typical algorithmic structures)
5. General scheme of serial algorithm
6. Complexity of serial algorithm
   (arithmetic operations, read/write operations)
7. Information graph
8. Resource of parallelism
   (complexity of parallel algorithm, available parallelism)
9. Input and output data
10. Properties of algorithm
    (computational intensity, stability, determinism, imbalance…)
11. Data locality, locality of computations →
12. Possible approaches to parallel implementations
    (features, properties, programming technologies…)
13. Possible obstacles for scalability
    (limited resource of parallelism, load imbalance, synchronizations…)
14. Dynamic properties and parameters: features of parallel implementations
15. Existing implementations

If data locality is low then structure of cache is not so important as low latency of the main memory.

The properties of algorithms determine computer architecture

1. General description
2. Mathematical description
3. Computational kernel
   (operations, data format, data structures)
4. Macrostructure of algorithm
   (macrooperations, typical algorithmic structures)
5. General scheme of serial algorithm
6. Complexity of serial algorithm
   (arithmetic operations, read/write operations)
7. Information graph
8. Resource of parallelism
   (complexity of parallel algorithm, available parallelism)
9. Input and output data
10. Properties of algorithm
    (computational intensity, stability, determinism, imbalance…)
11. Data locality, locality of computations
12. Possible approaches to parallel implementations
    (features, properties, programming technologies…)
13. Possible obstacles for scalability
    (limited resource of parallelism, load imbalance, synchronizations…)
14. Dynamic properties and parameters: features of parallel implementations
15. Existing implementations

Large amount and high intensity of small messages require a low latency interconnect.

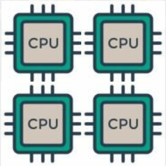
Supercomputing Co-Design of Target HPC Systems
(Use-case: 10 problems, 20 typical algorithms of the specific subject area)

Three-partition HPC system was designed after careful analysis of structure and properties of 20 target algorithms should be implemented and used on this system:

GPU partition

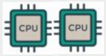
- HGX-H100-based platform;
- 8 * GPU H100 80GB per node;
- High performance on a particular class of problems.

SMP partition



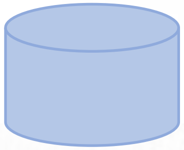
- Multi-socket (4+) multi-core CPU (64+) platform with large and fast shared memory; no GPUs;
- Target highly parallel problems required shared memory.

General purpose cluster partition



- General purpose multi-core CPU (16+) servers with generally available parameters;
- Multi-node general purpose cluster with GE interconnect and 48-port switch.

Data Storage



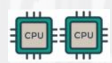
- Parallel file system with generally available parameters.

Supercomputing Co-Design of Target HPC Systems

(Use-case: an HPC system for simultaneous use by two different subject areas – specific problems of personalized medicine and climate modeling)

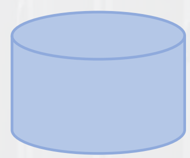
One-partition HPC cluster system with high-speed interconnect and
high-speed parallel data storage and file system was designed for these two subject areas:

General Purpose Cluster



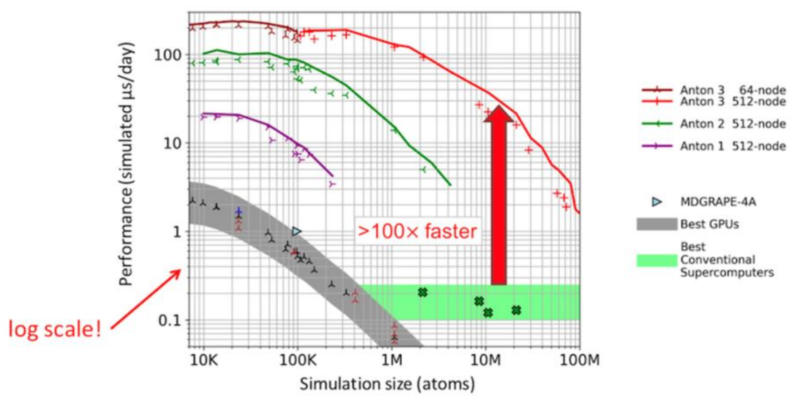
- Multi-core CPU (32+) servers with moderate but fast memory; maximum number of memory ports; no GPUs;
- Multi-node cluster (16-64) with high bandwidth and low latency interconnect.

Data Storage



- Parallel data storage and file system with highspeed I/O.

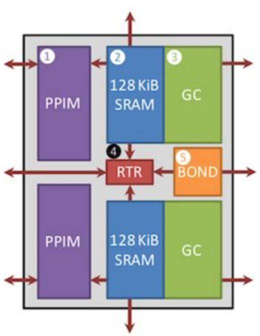
Anton Supercomputer – Apotheosis of Co-Design for Molecular Dynamics Simulations
(D.E. Shaw Research company)



Molecular dynamics on various computer platforms

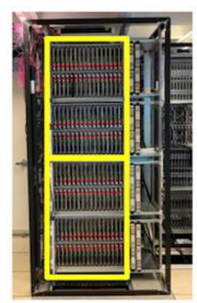
	ANTON	ANTON 2	ANTON 3
Tape-out	2007	2012	2020
CPU cores	8+4+1	66	528*
PPIMs	32	76	528*
Flex SRAM	0.125 MiB	4 MiB	66 MiB*
Atoms / node	460	8,000	110,000*
Clock frequency	0.485/0.970 GHz	1.65 GHz	2.8+ GHz
Channel bandwidth	0.607 Tbps	2.7 Tbps	5.6+ Tbps
Process node	90 nm	40 nm	7 nm
Transistors	0.2 G	2.0 G	31.8 G
Die size	299 mm²	410 mm²	451 mm²
Power	30 W	190 W	360 W

Generations of the Anton family processors

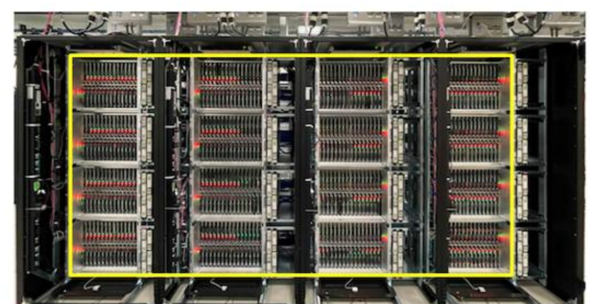


- Evolutionary changes
  - ① Support additional functional forms
  - ② Increase memory capacity
  - ③ Tune instruction set for MD application
  - ③ Increase code density
- **Revolutionary changes**
  - ④ Co-locate compute resources
  - ⑤ Specialize bonded force computation
  - ① Double effective density of pairwise interaction calculation
  - ②④ Implement fine-grained synchronization within memory and network

8×8 nodes



2×64 nodes



512 nodes

D E Shaw Research

Paradigms and problem-solving technologies
(current trends)

- **Mathematical modeling**
  - Solving systems of equations with initial and boundary conditions,
  - The traditional method of numerical analysis and predictive modeling,
  - Double precision calculations are required.
- **Big Data technologies**
  - They have received significant development due to the availability of large datasets,
  - They allow you to find hidden trends, facts and patterns using data analysis methods,
  - The data representation format varies from single to double precision.
- **Artificial Intelligence Technologies**
  - What contributed to the emergence of technology?
    - available powerful supercomputer resources,
    - available large datasets,
    - deep learning technologies.
  - They use large datasets and powerful computer resources to train neural networks,
  - Technologies for fast convolution calculations and tensor operations are needed,
  - High precision is not required for operations with real and integer numbers.

ANN models	Application	Computation for training
AlexNet	Image classification	4.7×10^{17} (470PFLOPs)
VGG16	Image classification	8.5×10^{18} (8.5EFLOPs)
YOLOv3	Image target detection	5.1×10^{19} (51EFLOPs)
Transformer	Natural language processing	7.4×10^{18} (7.4EFLOPs)
GPT-3	Natural language processing	3.1×10^{23} (310ZFLOPs)

https://docs.google.com/spreadsheets/d/1AAIebjNsnJj_uKALHbXNfn3_YsT6sHXtCU0q7OIPuc4

It is important:

-- All three approaches today coexist together, not replacing, but complementing each other.

-- All three approaches are based on supercomputing technologies and high-performance computing infrastructure.

-- **Parallel computing as a core of HPC is one of the pillars in supercomputing education**.

Intersect360 RESEARCH

Addison Snell,
Market share:
HPC only, no AI – 10%
AI only, no HPC – 10%
Both, HPC and AI – 80%

Supercomputing Education
(basic principles)

What is important for us in supercomputing education:
- -- completeness in parallel computing,
- -- fundamentality (not just skills, but understanding of fundamentals, not just training, but education),
- -- focusing on knowledge and competencies; complementing knowledge-based learning with competence-based learning.

Completeness is ensured by using **Body of Knowledge in Parallel Computing** - a set of basic concepts, terms, models, and technologies that make up the field of parallel computing.

Body of Knowledge in Parallel Computing **consists of five parts**:
1. Mathematical foundations of parallel computing.
2. Parallel computing systems (computer fundamentals).
3. Parallel programming technologies (fundamentals of software engineering).
4. Parallel algorithms for solving problems.
5. Parallel computing, grand challenges and specific subject areas.

Each part in the Body of Knowledge consists of sections, subsections and entries (250+ entries).

Each entry in the Body of Knowledge has a certain **level of difficulty** to be able to compose materials for various education programs.

Structure of Two Master's degree programs
(professional disciplines only)

(1) Master's Degree program
"Computational methods and modeling techniques"

Program supervisors :
E.E.Tyrtyshnikov, academician, RAS,
B.N.Chetverushkin, academician, RAS.


(2) Master's Degree program
"Supercomputer technologies of mathematical modeling and data processing"

Program supervisor :
V.V.Voevodin, corr.member, RAS.

The basic (common) part of both programs
- Introduction to Applied Mathematics
- Modern programming methods and technologies
- Probabilistic models
- The finite element method
- Supercomputing co-design technologies
- Hyperbolic type equations: methods of numerical solution

The variable part of the program
- Algebraic calculations, tensors, optimization
- Numerical methods in the solid mechanics
- Methods for constructing computational grids
- Parallel methods of problem solving
- Numerical methods of continuum mechanics
- Numerical methods in integral equations and their applications

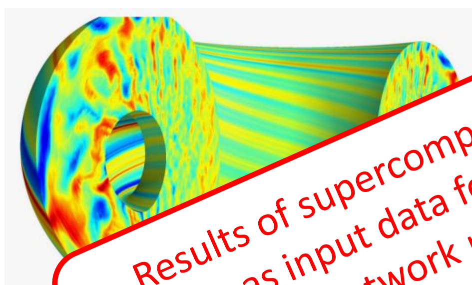
The variable part of the program
- Parallel algorithms
- Optimization methods
- High-performance computing on clusters using graphics accelerators
- Parallel programming
- Numerical methods
- Data mining
- Introduction to information security

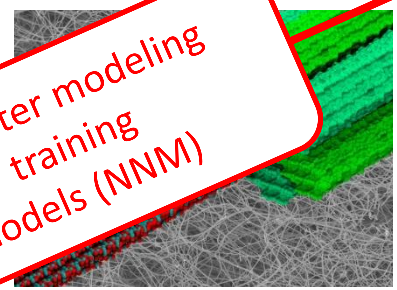
+ a set of special courses for each program …

A Paradigm Shift in Computing Sciences

Traditional use of high-performance computing (HPC)

Input parameters of an HPC-model → Calculation (hours/days/weeks) → Getting results





Results of supercomputer modeling as input data for training neural network models (NNM)

A cycle of scientific experiment

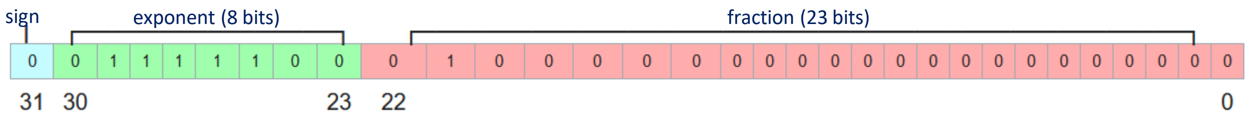
High-p… ce computing augmented by AI technologies

Collecting/generating data for NNM training → NNM training (weeks/months) → Working with NNM, inference (seconds/minutes) → Getting results

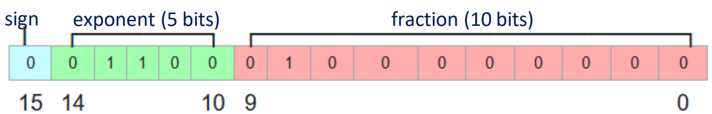
SC / AI : floating-point formats

(co-design: performance, compactness)

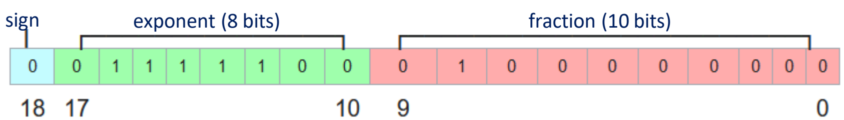
IEEE 754 single-precision 32-bit float



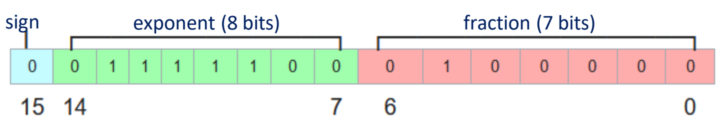
IEEE half-precision 16-bit float



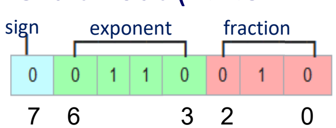
NVIDIA's Tensor float (19 bits)



bfloat16



8-bit float (1.4.3 minifloat)



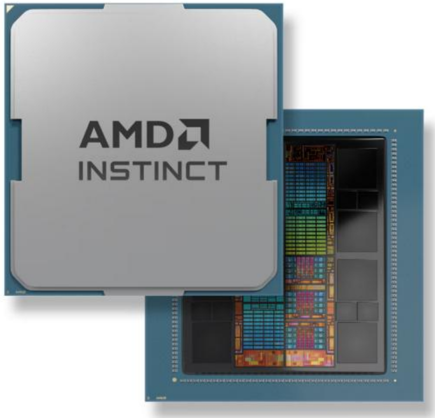
NVIDIA GPUs:

	Hopper	Ampere	Turing	Volta
Tensor cores operations	FP64, TF32, bfloat16, FP16, FP8, INT8	FP64, TF32, bfloat16, FP16, INT8, INT4, INT1	FP16, INT8, INT4, INT1	FP16
CUDA cores operations	FP64, FP32, FP16, bfloat16, INT8	FP64, FP32, FP16, bfloat16, INT8	FP64, FP32, FP16, INT8	FP64, FP32, FP16, INT8

SC / AI : floating-point formats
(co-design: performance, compactness)

AMD Instinct MI300A: peak performance of the graphics accelerator integrated in this processor (depending on operations and data formats) :

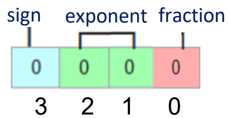
Eight-bit Precision	(FP8, E5M2, E4M3)	1960.0	TFLOPs (1.96 PFLOPs)
Eight-bit Precision	(FP8, with Structured Sparsity, E5M2, E4M3)	3920.0	TFLOPs (3.92 PFLOPs)
Half Precision	(FP16)	980.6	TFLOPs
Half Precision	(FP16) with Structured Sparsity	1960.0	TFLOPs (1.96 PFLOPs)
Single Precision	(TF32 Matrix)	490.3	TFLOPs
Single Precision	(TF32) with Structured Sparsity	980.6	TFLOPs
Single Precision Matrix	(FP32)	122.6	TFLOPs
Double Precision Matrix	(FP64)	122.6	TFLOPs
Single Precision	(FP32)	122.6	TFLOPs
Double Precision	(FP64)	61.3	TFLOPs



The less precision algorithms require, the higher performance of computing systems.

SC / AI : floating-point formats – FP4
(co-design: performance, compactness)

4-bit float



The floating-point format of a 4-bit real number contains all the necessary fields!

A table with **all possible values** in FP4 format !

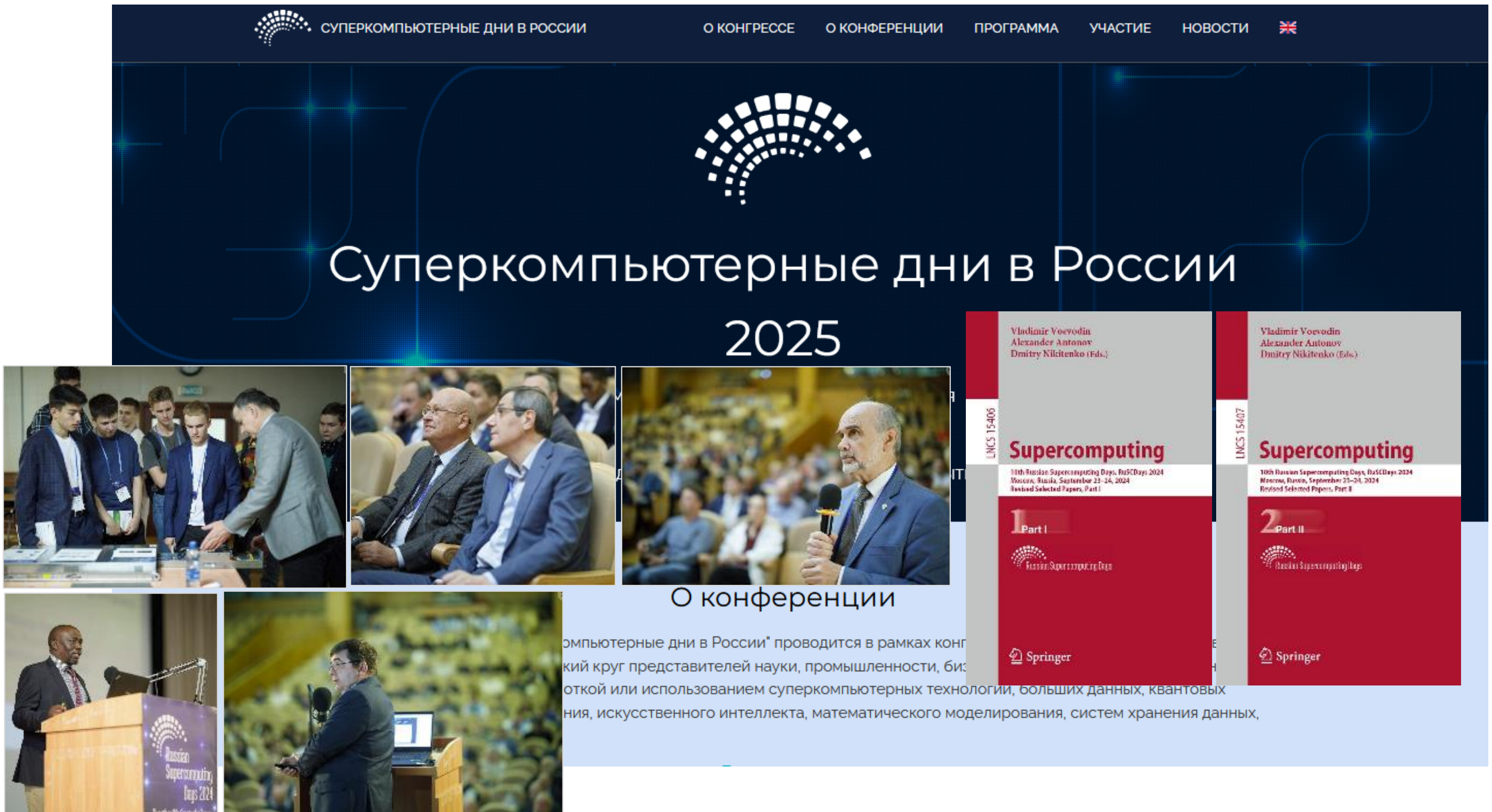
Exponent \ Sign and fraction	0 × × 0	0 × × 1	1 × × 0	1 × × 1
× 0 0 ×	0	0.5	-0	-0.5
× 0 1 ×	1	1.5	-1	-1.5
× 1 0 ×	2	3	-2	-3
× 1 1 ×	Inf	NaN	-Inf	NaN

Why does using such a small and fixed set of these numbers in LLM give meaningful results of the model?

Again, co-design + fundamentality of education are required…

Международная научная конференция «Суперкомпьютерные дни в России» – 29-30 сентября 2025

- *Смешанный формат (Шуваловский корпус)*
- *364 участника (более 330 очное участие)*
- *50 городов (более 130 организаций)*
- *Участники из Китая, ЮАР, Бразилии, Индии*
- *Вручение премии имени В.П. Гергеля за лучший доклад конференции молодых ученых*
- *Публикация трудов в серии LNCS Springer*



https://RussianSCdays.org/

Thank you for your attention!

- *Supercomputing co-design is a key point for High Performance in Computing, but fundamental research is required to unlock its potential.*
- *Completeness, Fundamentality and Competences should be integral parts of Supercomputing Education.*

voevodin@parallel.ru