Model of problem-oriented cloud computing environment

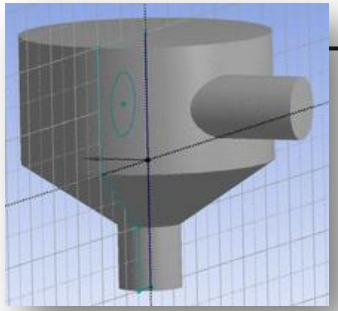
G. RADCHENKO

SOUTH URAL STATE UNIVERSITY

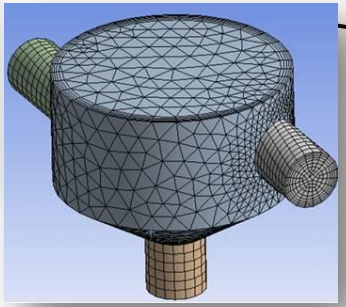
Problem definition

- Modern problems of computational science are characterised by high demands on provided computing resources and complex computing tasks structure, which can be defined as a workflow.
- Also, problems of this type are characterised by usage of multivariant calculations computing task is run hundreds or thousands of times with different variations of the input parameters.

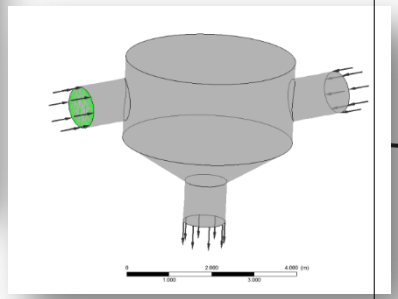
Example workflow for mixer simulation



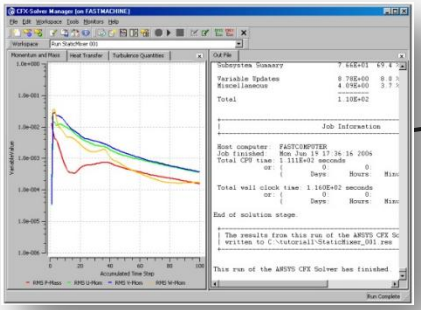
Design Modeler



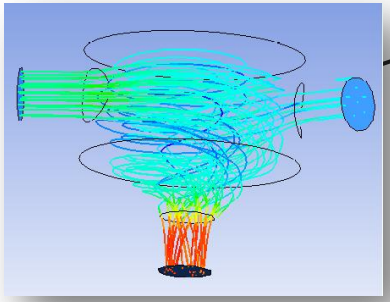
CFX-Mesh



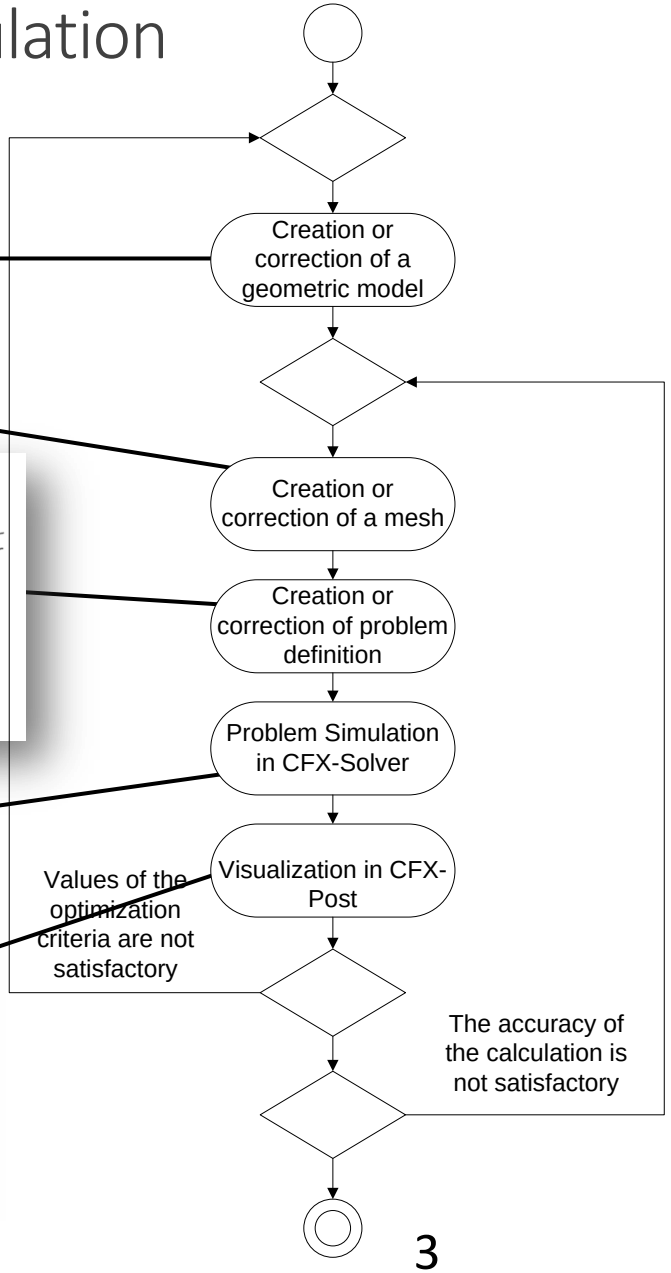
CFX-Pre



CFX-Solver



CFX-Post

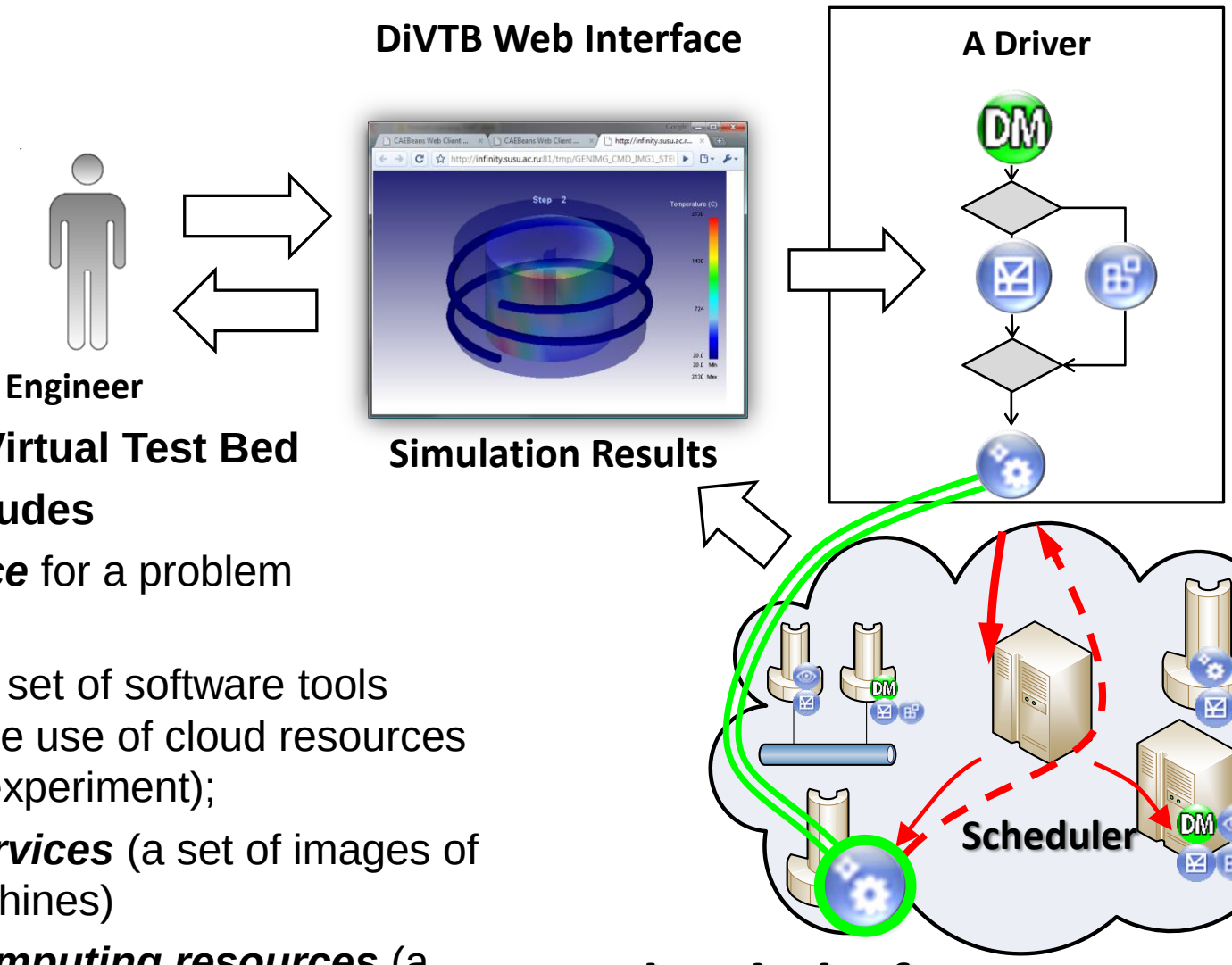


Distributed Problem-Oriented Environments

Goal:

- To develop methods and algorithms of workflow applications scheduling to meet the challenges of computational science in cloud computing
- The principal objective is to develop a technology allowing to take into account the specifics of the problem-oriented subject areas while providing the recourses of cloud computing environments.
- This technology aimed to create “intelligent” middleware providing users with easy, transparent and secure access to distributed computing resources and allowing them to solve specific classes of applied problems.

Distributed Virtual Test Bed



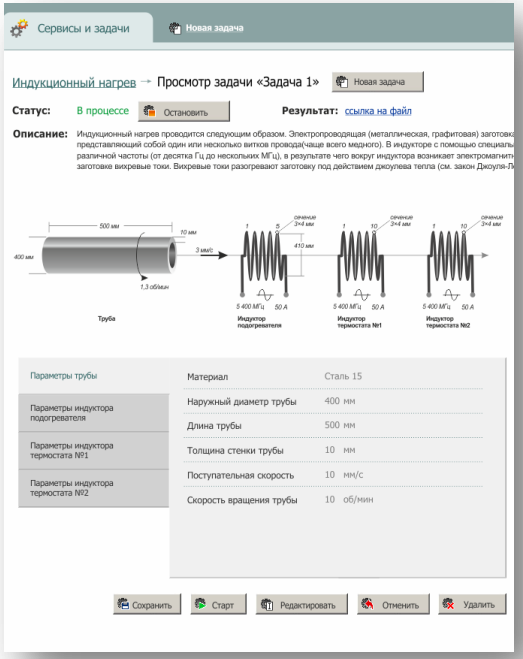
Distributed Virtual Test Bed (DiVTB) includes

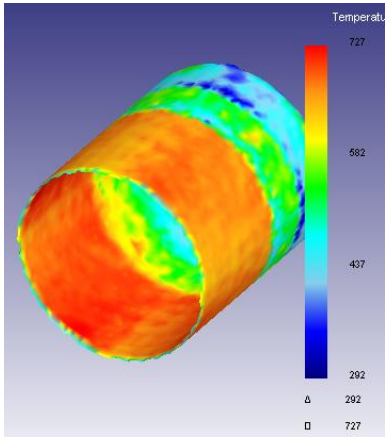
- an ***interface*** for a problem statement;
- a ***driver*** (a set of software tools enabling the use of cloud resources for virtual experiment);
- a set of ***services*** (a set of images of virtual machines)
- a set of ***computing resources*** (a cloud computing environment)

DiVTB «Simulation of Tube Ovalization during Hardening»

Customer:
Chelyabinsk Tube-Rolling Plant

Purpose of study:
simulation of tube deformation during hardening during production on tube rolling plants





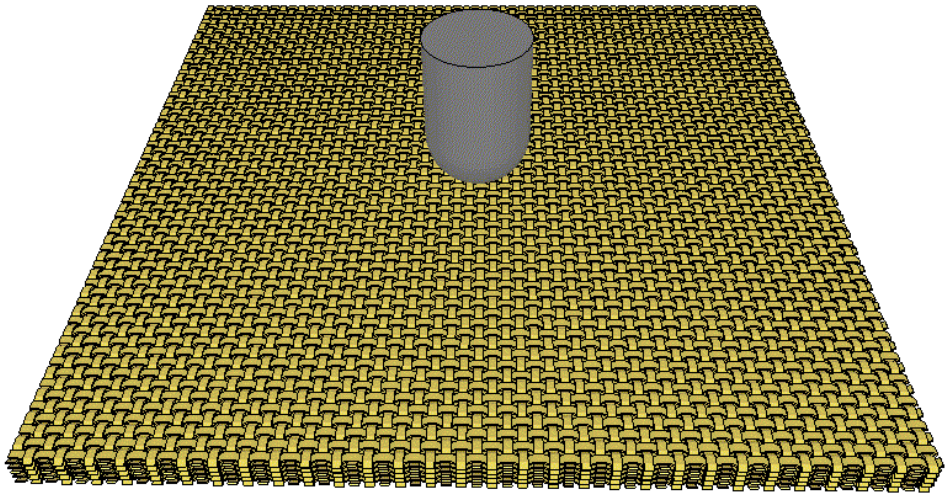
Temperature distribution during water sprayer pipe cooling

Deformation and Destruction of Fabric Bulletproof Vest by a Bullet Strike

Customer:
"FORT Technology"

Purpose of study:
development of new body armor design

Result:
bulletproof vests are on arms of special units of the Russian Federation and foreign countries (Germany, France, Israel, Poland, Czech Republic, etc.)



Tissue concealable bulletproof vest

Subject area in problem-oriented computing environments

- Feature of distributed problem-oriented computing environments (POE) is that they enable problem simulation within **a specific subject area**.
- On the one hand, it imposes restrictions on the classes of problems that can be resolved through POE.
- On the other hand, this restriction allows you to use **information about the subject area to predict computational characteristics of tasks** during the tasks scheduling and distribution, increasing efficiency in the use of available computing resources.

Subject area definition

Information object

A set of basic information objects $\mathcal{B}$ is counting set of objects that are treated as atomic within the subject area (i.e. do not have any parts), and divided into k disjoint sets *of classes of basic information objects* B_i:

$$\mathcal{B} = \bigcup_{i=1}^{k} B_i \,, (\forall i \neq j) B_i \cap B_j = \emptyset.$$

Classes of basic information objects examples:

- strings;
- integers;
- Boolean variables.

Class of Information objects

We define *the class of Information objects* C as follows:

- a class of basic information objects defines the class of information objects:

$$C = (B), \forall B \in \mathcal{B};$$

- an ordered set of classes of basic information objects defines a class of information objects:

$$C = (B_1, \dots, B_m), \left(\forall l \in \{1, \dots, m\}\right) B_j \in \mathcal{B};$$

- an ordered set of previously defined classes of information objects will define a class of information objects:

$$C = (C_1, \dots, C_u).$$

Information object

Thus, the information object I of the class $C = (C_1, \dots, C_u)$ is an ordered set of finite number of information objects $I(I_1, \dots, I_u)$.

Examples of information objects:

- 3-d model of the physical process;
- Computational mesh;
- Simulation results.

Set of classes of information objects

We define *a set of classes of information objects* $\mathcal{C}$ as a finite set of all classes of information objects defined within the subject area:

$$\mathcal{C} = \bigcup_{i=1}^{|\mathcal{C}|} C_i \,.$$

Набор функций над $\mathcal{C}$

We define *a set of functions* $\mathcal{F}$ over the $\mathcal{C}$:

$$\mathcal{F} = \bigcup_{i=1}^{|\mathcal{F}|} f_i \,, where\ (\forall i \in \{1, \dots, |\mathcal{F}|\})\ f_i : \mathcal{C}_i^{in} \rightarrow \mathcal{C}_i^{out}$$

Each function $f \in \mathcal{F}$ receives n information objects $\mathcal{I}^{in} = \left(I_1^{in}, \dots, I_n^{in}\right)$. The result of the function are m new information objects $\mathcal{I}^{out} = (I_1^{out}, \dots, I_m^{out})$ of the relevant classes $\mathcal{C}^{out} = (C_1', \dots, C_m')$.

Examples of such functions:

- function that converts computational mesh from one format to another;
- function that generate a mesh from a defined geometry;
- function that provides visualisation of simulation results.

Subject area definition

We define *a subject area* $\mathfrak{P}$ as ordered triad:

$$\mathfrak{P} = \langle \mathcal{B}, \mathcal{C}, \mathcal{F} \rangle$$

where

- $\mathcal{B}$ - a set of basic information objects
- $\mathcal{C}$ - a set of classes of information objects
- $\mathcal{F}$ - a set of functions defined over the $\mathcal{C}$

Problem-oriented cloud computing environment

Problem-oriented cloud computing environment

We define *a problem-oriented* cloud computing environment as a quadruple:

$$\mathfrak{C}(\mathfrak{N}, \mathfrak{E}, \mathfrak{M}, \mathfrak{P}),$$

where

- $\mathfrak{N}$ - a set of nodes of a computer system;
- $\mathfrak{E}$ - a set of network links connecting the nodes;
- $\mathfrak{M}$ - a set of images of virtual computing machines available to deploy to nodes from a $\mathfrak{N}$ set,
- $\mathfrak{P}$ - basic subject area

Computing node

We define a computing node $\mathfrak{n} \in \mathfrak{N}$ as a shared memory computing system, defined by a triad:

$$(P_{\mathfrak{n}}, m_{\mathfrak{n}}, \Pi_{\mathfrak{n}}),$$

where

- $P_{\mathfrak{n}}$ - is an ordered set of computing cores;
- $m_{\mathfrak{n}}$ - is the amount of RAM available on the computing node;
- $\Pi_{\mathfrak{n}}$ - is a vector of performance characteristics of computing node.

Virtual Machine Image

We define a virtual machine image $\mathfrak{m} \in \mathfrak{M}$ as a triad:

$$(P_{\mathfrak{m}}, m_{\mathfrak{m}}, \Pi_{\mathfrak{m}}),$$

where

- $P_{\mathfrak{n}}$ - is an ordered set of computing cores, dedicated to a virtual machine;
- $m_{\mathfrak{m}}$ - is the amount of memory allocated to the virtual machine;
- $\Pi_{\mathfrak{m}}$ is a vector of performance characteristics of a virtual machine.

Performance characteristic

We define a performance characteristic of a computing machine as a mapping operator:

$$\pi: \mathfrak{m} \to \mathbb{Z}_{>0},$$

where $\mathfrak{m}$ is the image of a virtual machine (or computing node) in a comping system $\mathfrak{C}$.

Examples of performance characteristics: numerical characteristics of the machine, the results of synthetic tests (Dhrystone, SuperPI, LAPACK, LINPACK, etc.) or the results of the test execution of specific classes of functions with pre-defined sets of input data.

We assign an r-vector of performance characteristics to all machines in $\mathfrak{C}$:

$$\Pi: \mathfrak{M} \to \mathbb{Z}_{>0}^{r}$$

Operators of the problem-oriented environment

We will require that within a problem-oriented computing environment for each class of problems were identified the following operators for the prediction of the function execution characteristics depending on the values of the input parameters:

- an operator of evaluation of the expected output (when input parameters are defined);
- an operator of evaluation of the expected execution time (when input parameters and machine performance characteristics vector defined).

Operator of expected output

For each function, $f \in F$ of the subject area $\mathfrak{P}$, running in problem-oriented environment $\mathfrak{C}$, we define *expected output operator* $\nu(f, \mathcal{J}^{in})$ - an operator that returns the expected total byte size of all output information objects $\mathcal{J}^{out}$:

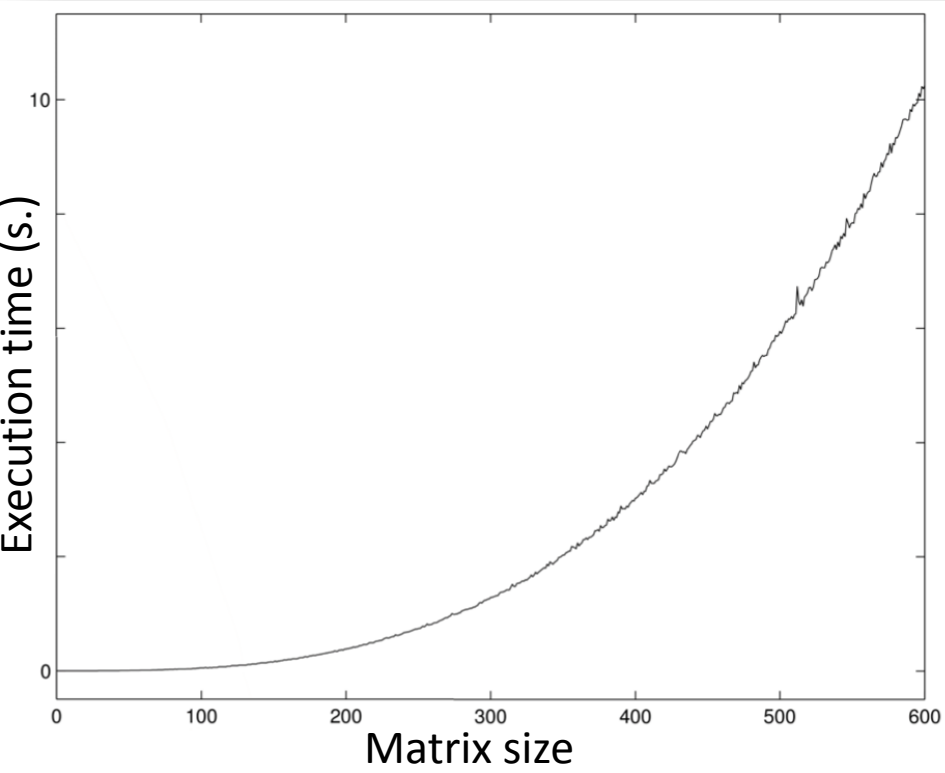
$$\nu\left(f, \mathcal{J}^{in}\right) = |\mathcal{J}^{out}| = \sum_{\forall I \in \mathcal{J}^{out}} \sum_{\forall b \in I} |b| .$$

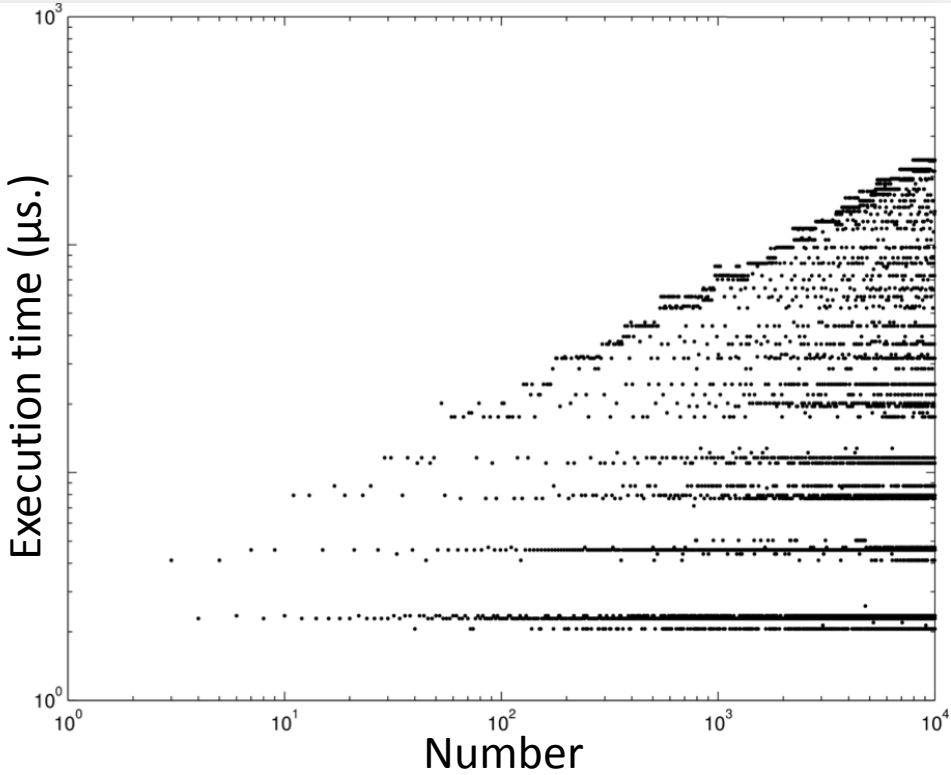
Operator of the expected execution time

Also, for each function $f \in F$ we define an *operator of the expected execution time* $\tau\left(f, \mathcal{I}^{in}, \Pi\right)$, which returns the estimated execution time (in seconds) of the function f given a set of input information objects $\mathcal{I}^{in}$ on a machine with an given vector of performance characteristics Π:

$$\tau: \left(f, \Pi, \mathcal{I}^{in}\right) \rightarrow \mathbb{N}.$$

Expected execution time as a random variable





The execution time of Cholesky algorithm is strongly depends on the size of the input matrix

The execution time of the trial division algorithm is weakly correlated with the input value

Expected execution time as a random variable

Execution time evaluation function can be simulated as a random variable:

$$\chi(f, \Pi, \mathcal{I}^{in}) = \tau\left(f, \Pi, \mathcal{I}^{in}\right) + \alpha,$$

where

$\tau\left(f, \Pi, \mathcal{I}^{in}\right)$ - a deterministic function that represents the correlation between execution time of function f on a machine with performance characteristics vector Π and information objects input vector $\mathcal{I}^{in}$,

α – is a stochastic variable with zero expectation function ($M[\alpha] = 0$), that represents a set of factors that are not included in the model.

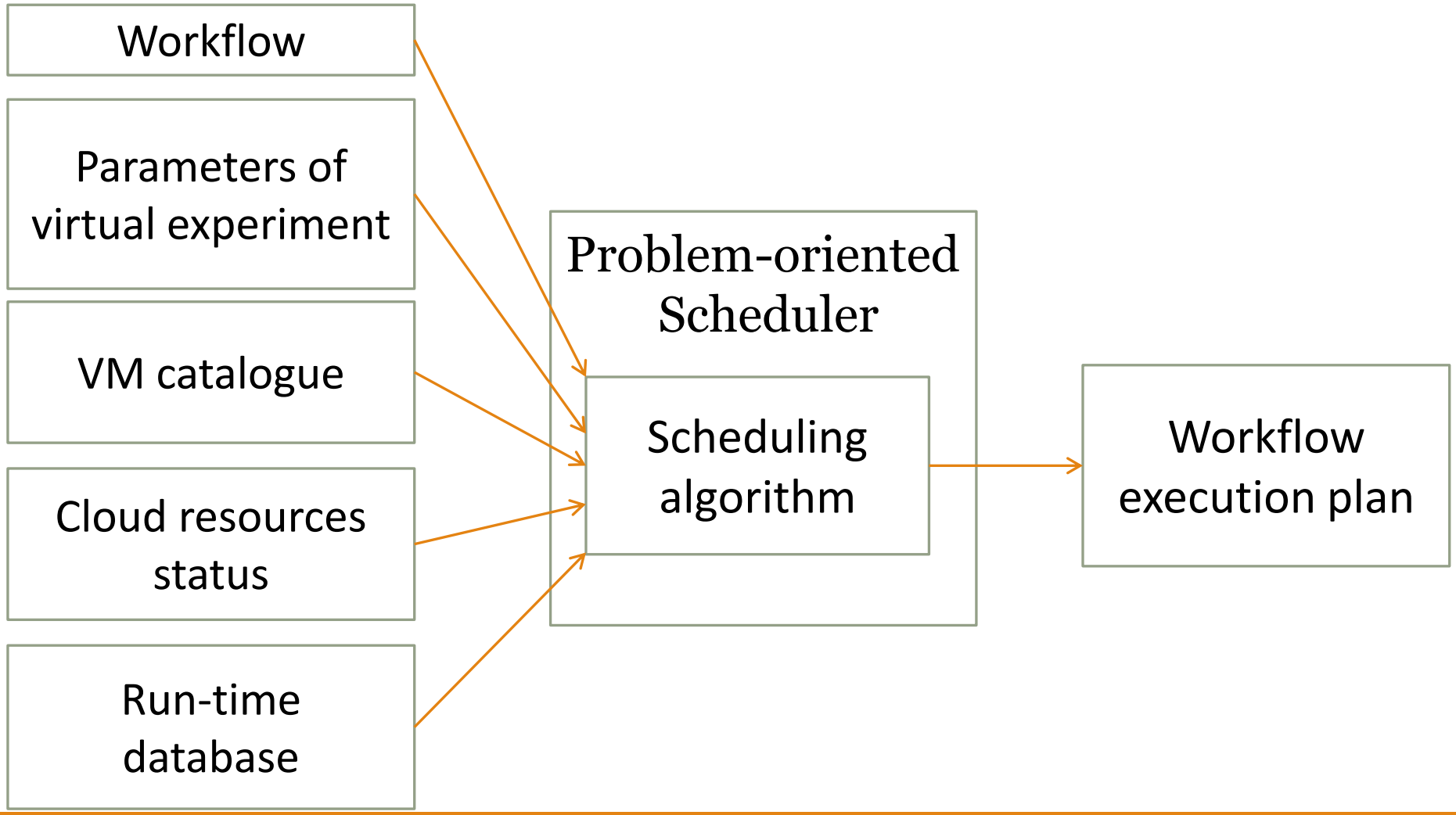
Model application and future work

Problem-oriented scheduler

To implement a problem-oriented cloud computing environment a Problem-oriented scheduler component is being developed:

- database of runtime data from previous runs used for statistical prediction algorithm;
- correction of workflow prediction time during the execution of the workflow through the collection and continuous analysis of in-process information on the execution of the current workflow.

Problem-oriented Scheduler



Tasks that need to be addressed

Future work:

- Development of a database to support an estimation of execution characteristics of calculation tasks.
- Development of a model of workflow execution in cloud computing environment.
- Development of workflow applications scheduling algorithm with effective virtual machines allocation and possibility of dynamically adjustment of the execution plan of the application.
- Development of an experimental “Problem-oriented Scheduler” system.