ITMO UNIVERSITY

Saint Petersburg, Russia

The multi-level adaptive approach for efficient execution of multi-scale distributed applications with dynamic workload

Denis Nasonov , Nikolay Butakov , Michael Melnik , Alexandr Visheratin,
Alexey Linev, Pavel Shvets , Sergey Sobolev , and Ksenia Mukhina

ITMO University

2018

Introduction

ITMO UNIVERSITY

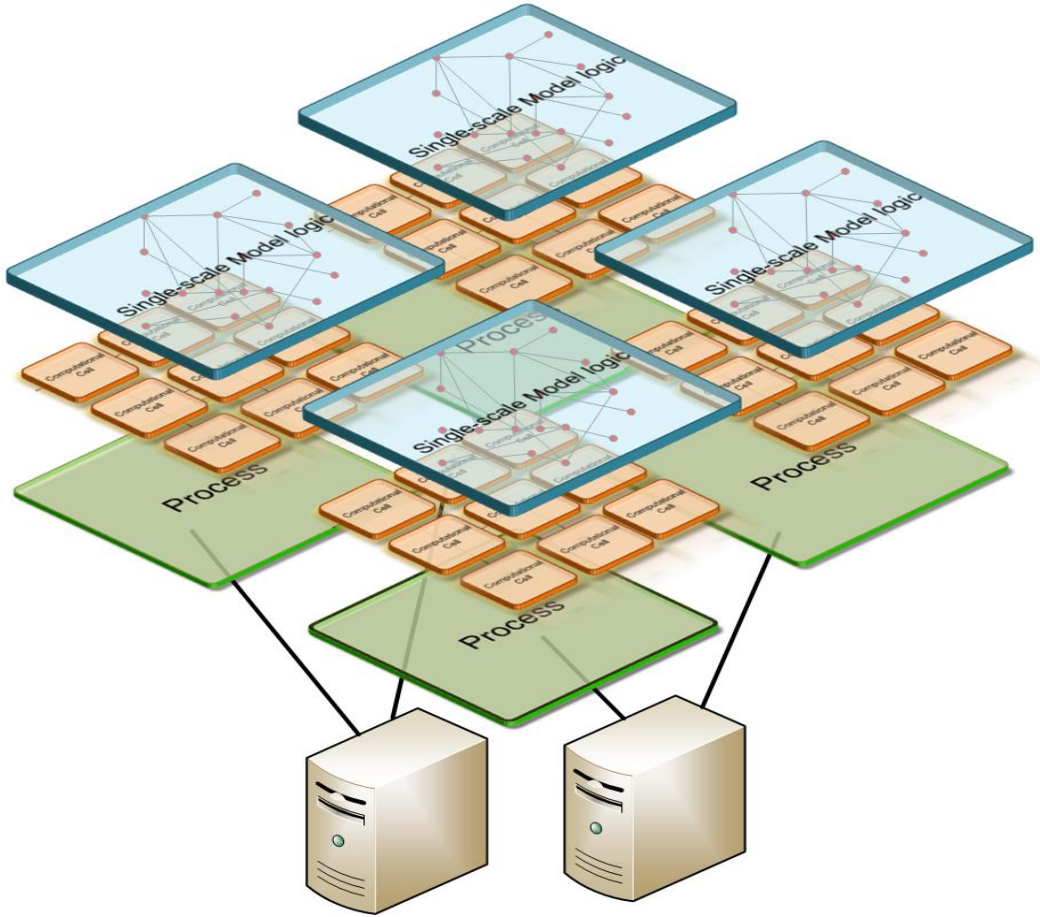
The performance increase of SC is also associated with the growing complexity of the CS architecture. It leads to the following challenges:

1. the need to use multi-scale and multi-physical models, various modelling methods (grid and drains) in the solution of one applied problem;
2. the use of specialized computation resources (for example, graphics processing units);
2. the problem of balanced spatial decomposition due to the complexity of the geometry of the domain of definition;
3. dynamic change in the complexity of different parts of the problem: with spatial decomposition due to the change in the geometry of the system or due to the emergence of areas of high computational complexity (for example, clustering of agents in multi-agent systems, slow convergence regions for grid methods;
4. the development of cloud computing technologies, in which the SC architecture is hidden from the user, and the need to meet their requirements.
5. etc.

ITMO more than a UNIVERSITY

Codesign-oriented execution approach

ITMO UNIVERSITY



3 main levels are supposed**:**

- **Model logic execution layer** – implements the internal behavior of the model and has no connection to infrastructure part
- **Execution environment** – is a middle layer that allows model and infrastructure to communicate with each other through space model and computation distributed agents
- **Infrastructure container** – contains scheduling algorithm that manage application node workload changing simulation space
-

Problem statement

ITMO UNIVERSITY

Computational graph G <V, E>. $V = \{v_j\}$ – model blocks, $E = \{e_{j1,j2}\}$- edges, $W = \{w_j\}$ – current workload, R={r_m} – computational resources. If $S = \{w_j\}$

$$f(S_1, S_2) = \sum \frac{c_{w_{j`}^{j}}}{e_{j,j`}} \delta_{j`}^{j},$$

$$\delta_{j`}^{j} = \begin{cases} 1, & if\ j\ and\ j`\ on\ the\ same\ resource \\ 0, & otherwise \end{cases} \qquad \forall\ j, j` = 1, \dots, J_m,$$

where $c_{w_{j`}^{j}}$ - is amount of metadata needed to transfer load from $v_j$ to $v_{j`}$, $e_{j,j`}$ - corresponds to network channel throughput.

$$T(S) = \max_m (\sum_j^{J_m} \frac{w_j}{r_m} + \sum_j \sum_{j`} \frac{w_{j`}^{j}}{e_{j,j`}} \cdot \tau_j^i,),$$

where

$$\tau_j^i = \begin{cases} 1, & if\ j\ and\ j`\ on\ the\ same\ resource \\ 0, & otherwise \end{cases}$$

Then condition to change the environment configuration is:

$$T(S_{prev}) \cdot \theta > f(S_{prev}, S_{new}) + T(S_{new}) \cdot \theta$$

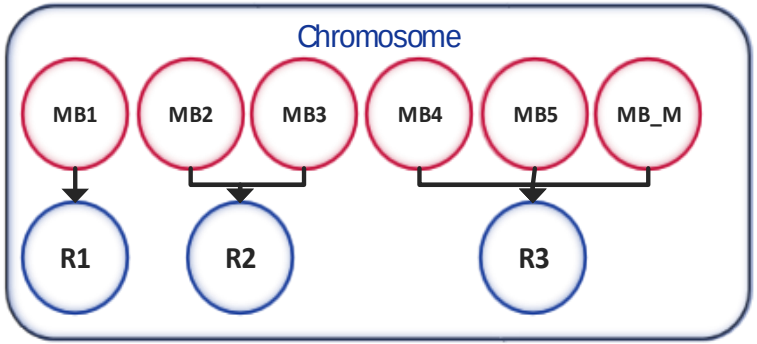
Scheduling algorithm implementation

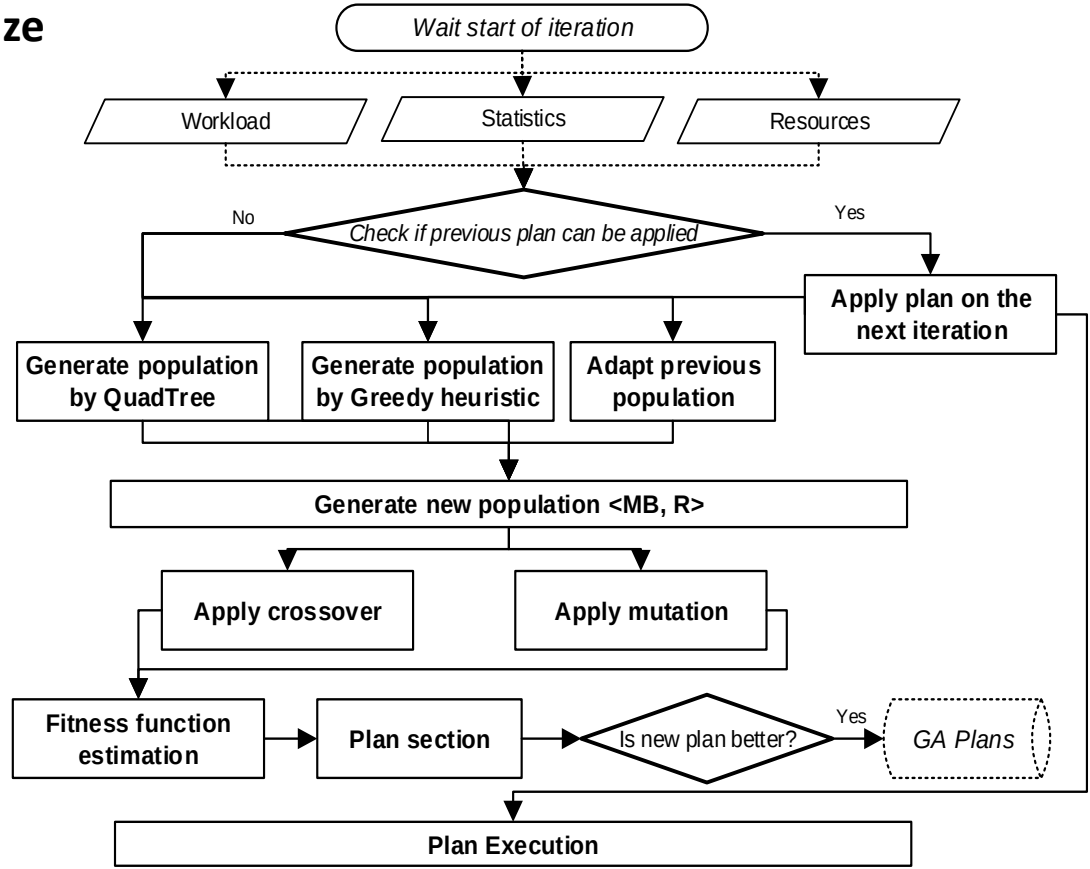
ITMO UNIVERSITY

We use genetic algorithm (GA) to optimize distribution of model workload:

The parameters of the algorithm:

- size of population – 100;
- count of generations – 300;
- mutation probability - 0.7;
- crossover probability - 0.3;
- selection operator - roulette wheel.

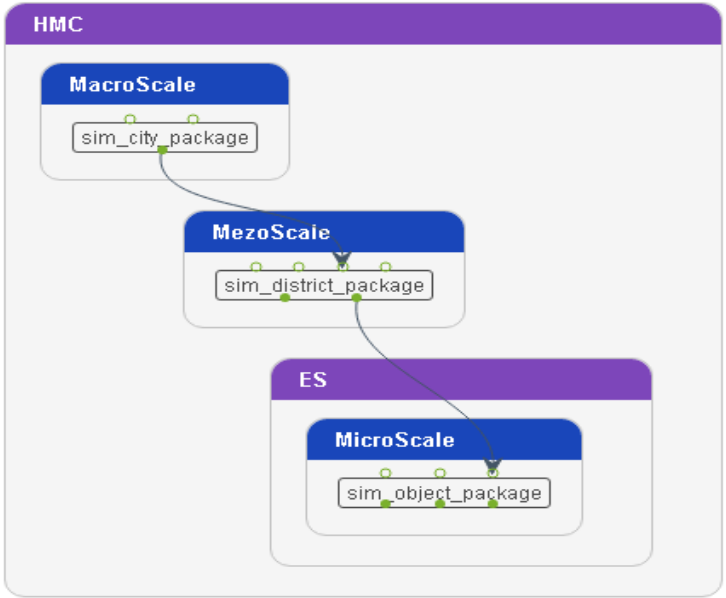


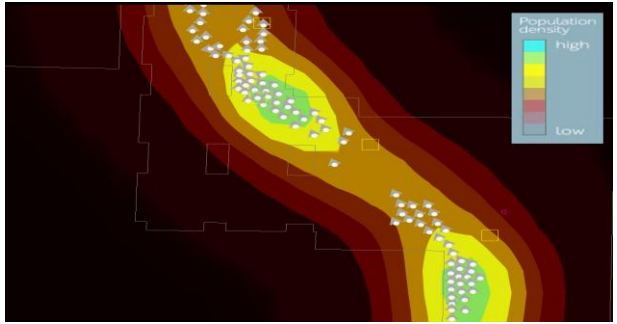


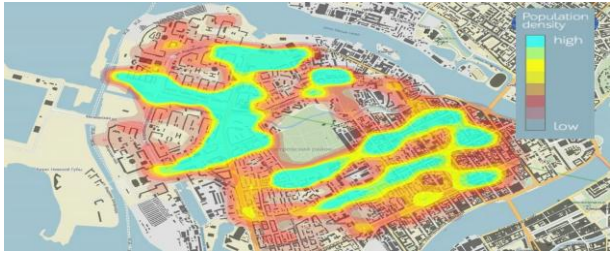
CitySimulator multiscale application

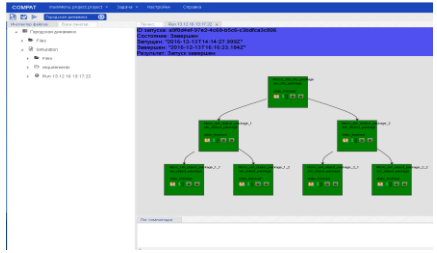
ITMO UNIVERSITY

The goal of City-Simulator is to simulate urban mobility of the population with in multiscale city model. Here we have 3 scales: city, district (urban infrastructure) and agents, that can interact with each other and allow different urban scenarios. Because of high CPU consumption this application should be executed on supercomputer.



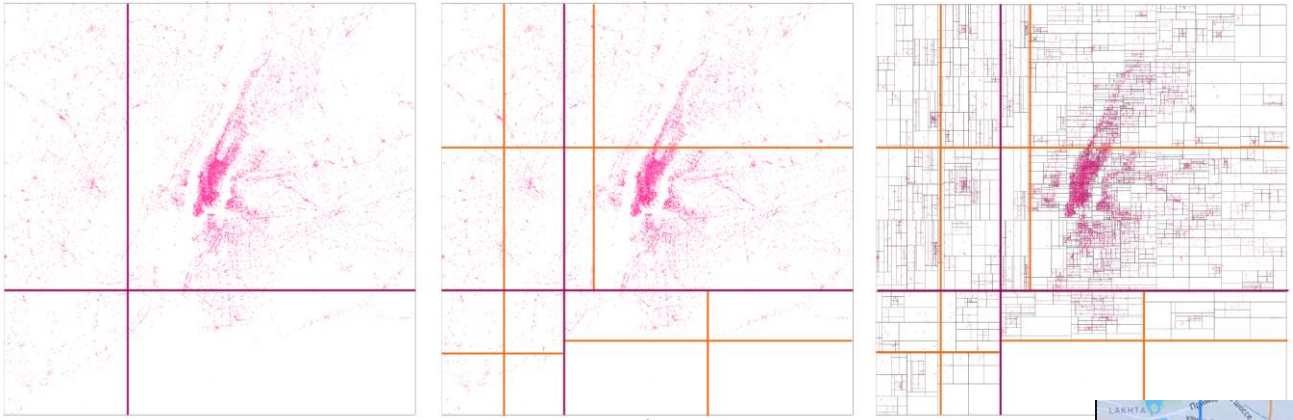






CitySimulator event detection

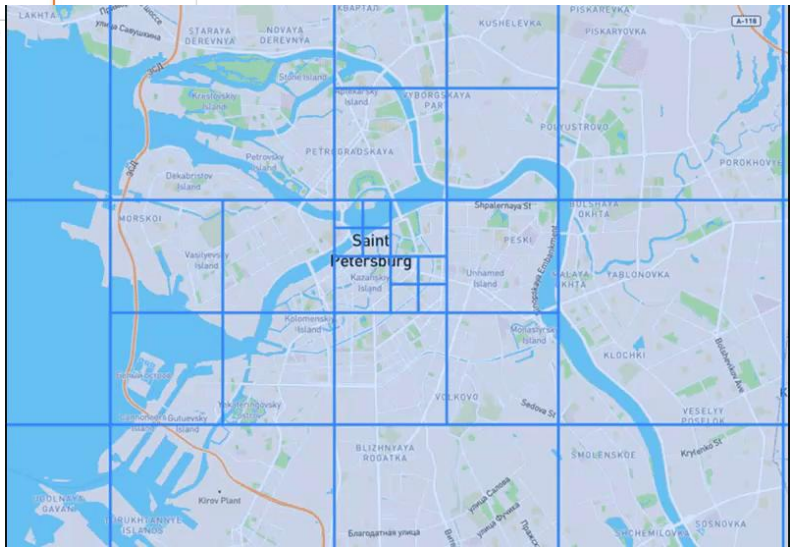
ITMO UNIVERSITY



Fast and efficient detection of events of various scales – from very local (indie band concert) to city (holiday march) and even country scale (political protest)

Convolutional quadtree – enhanced quadtree that considers spatial data distribution

Adaptive geogrid – representation of the urban area state by workload



ITMO more than a UNIVERSITY

Codesign-oriented execution approach

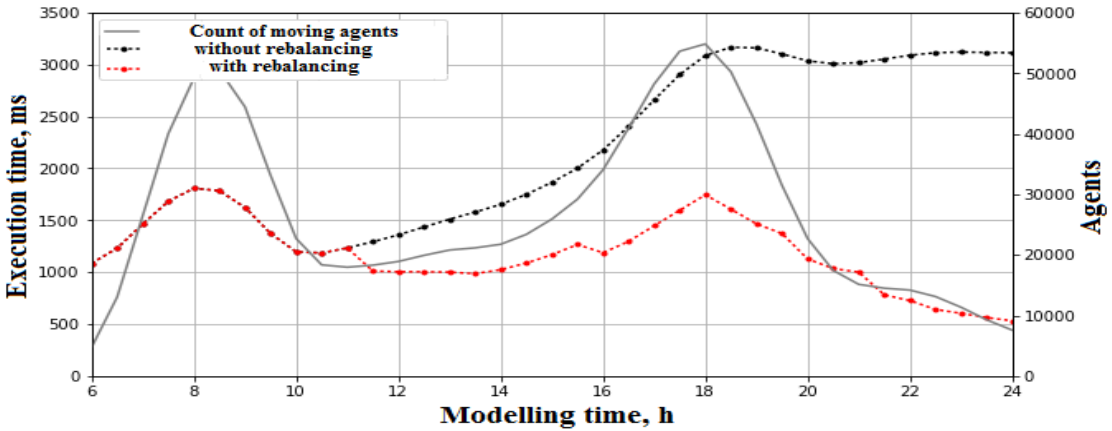
ITMO UNIVERSITY

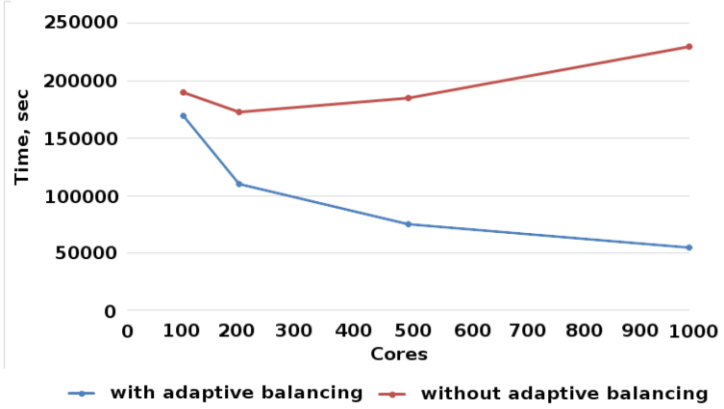
- Agents move in the space forming different density in the space splitted between resources
- Workload are non-uniform: low on some nodes, and high on others
- Rebalancing by provided schedulers allows to reduce the execution time

- 6 million agents to produce simulations on a city scale both regarding the size of the calculation area and regarding the size of the population of agents.
- The application was planned for 100, 200, 500 and 1000 cores.

Increased performance results:

- up to 32% in case of 200 cores,
- up to 18% in case of 500 cores

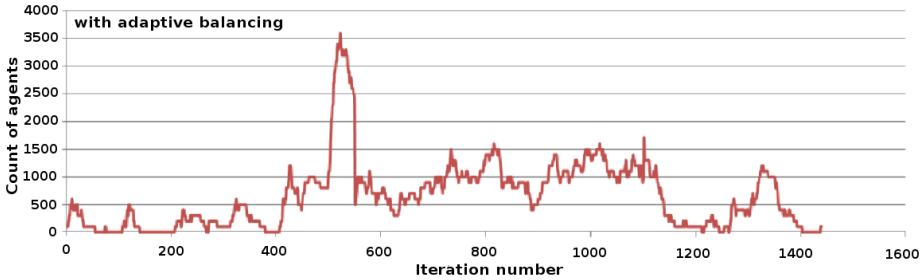


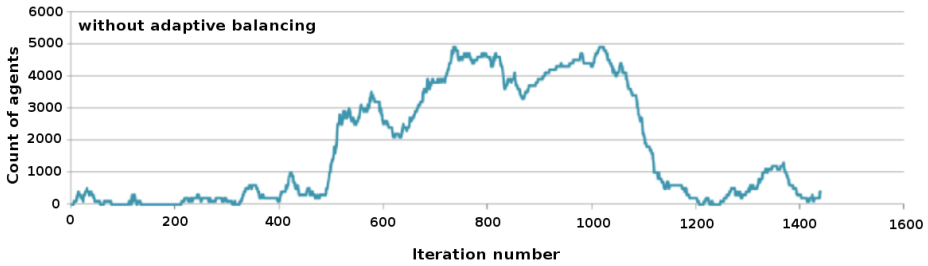


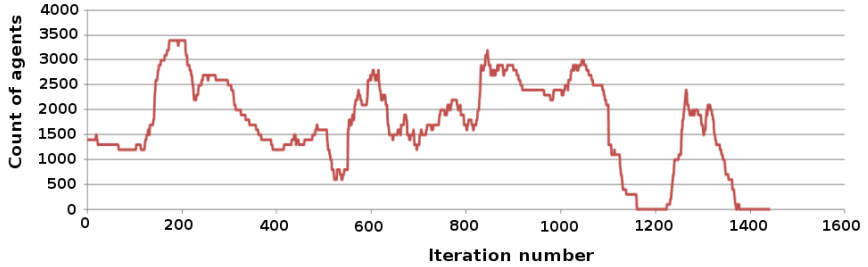
Adaptation to the infrastructure

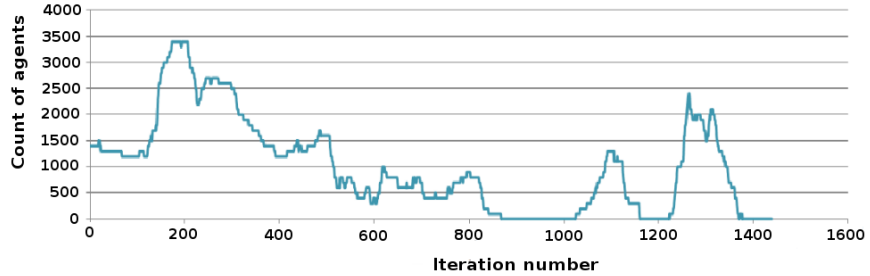
ITMO UNIVERSITY

- Example of workload on the most “busy” nodes and on the most “empty” nodes
- For adaptive schema, the algorithm tries to decrease dispersion among all execution nodes and through the whole execution time (for busy case we have 3500 and 5000 max values). It means that high-loaded nodes share their workload with low-loaded nodes









Conclusion and future plans

ITMO UNIVERSITY

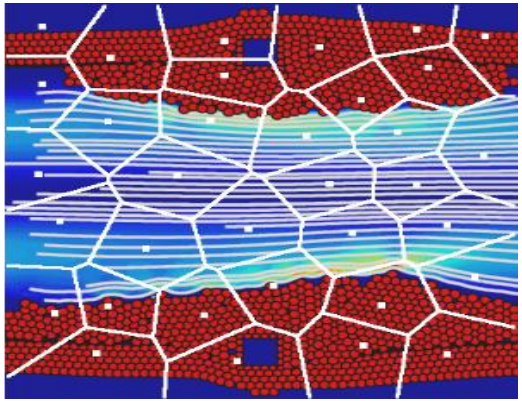
- The experimental case study demonstrated an improvement in execution time with the growth of exploited cores for:
  - up to 32% in case of 200 cores,
  - up to 18% in case of 500 cores thus showing that the proposed approach is able to deliver.
- Significant improvement in efficiency of scaling for distributed applications.

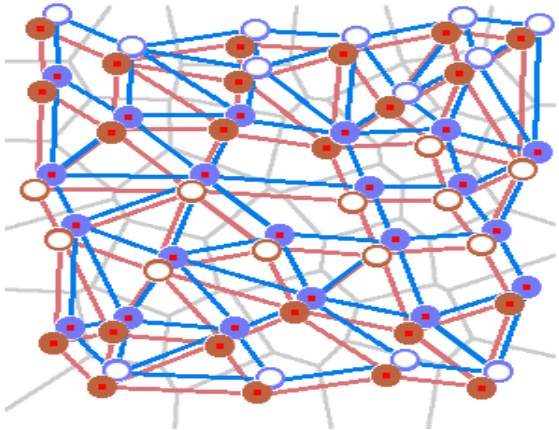
Plans:

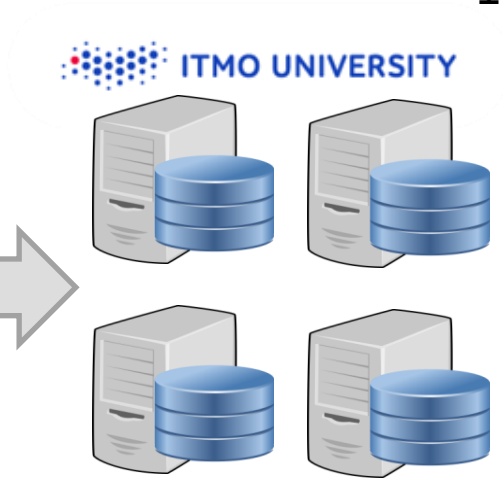
- Adaptation for more complex one-model scenario
- Extension for cooperation of several models together

ITMO more than a UNIVERSITY

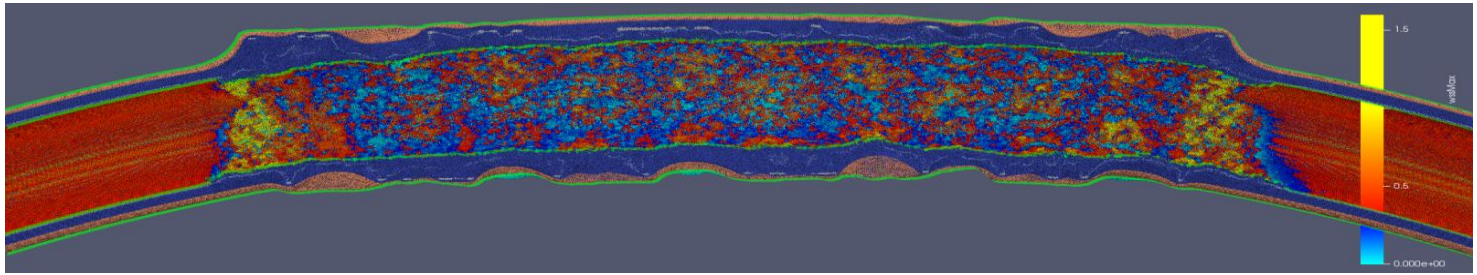
Restenosis application







- This multiscale model includes two single-scale models: blood flow dynamics and vascular wall growth, acting in the same domain
- As the growth progresses, the load of these two models changes: the amount of wall elements increases, and the flow area shrinks



ITMO More than a UNIVERSITY

Thank you for your attention!
Q&A?

www.ifmo.ru