Jupyter extension for creating CAD designs and their subsequent analysis by the finite element method

O. Iakushkin, A. Kondratiuk, O. Sedova, V. Grishkin

Saint Petersburg State University, 7/9, Universitetskaya emb., Saint Petersburg, 199034, Russia

E-mail: o.yakushkin@spbu.ru

Creating designs in CAD and performing their stress-strain analysis are complex computational tasks. Their successful solution depends on a number of prerequisites: availability of large computational power; comprehensive knowledge of physical and mathematical computing; and solid skills of programming and working in a variety of separate software products that are not integrated to each other directly.

The paper presents a system aimed at CAD models development and verification from the ground up. The system integrates geometry construction, mesh model creation and deformation analysis into a uniform computing environment operating as a SaaS solution. It is based exclusively on open source software and allows to use the Python programming language and SALOME, GMSH, FEniCS and ParaView libraries. The system's architecture and certain issues of working with libraries are discussed. The paper also presents a browser-based tool for CAD design creation and analysis, which tool is the front end of the software product we created.

Keywords: CAD, Cloud, Services, IPython

The work was supported by RFBR research project № 16-07-011113 and SPBU projects 0.37.155.2014 and 9.37.157.2014

Introduction

There are three major reasons prompting a researcher to develop his own FEM solver to analyse the state of solids. Firstly, an attempt to build a product analogous to SolidWorks or ANSYS. We will not dwell upon this option, since it is very unlikely to become a success story. Secondly, the software mentioned might not meet the specific features of a particular problem, impose demanding hardware requirements, or exceed the budget due to high licensing fees. Thirdly, the choice might be driven by discrepancy between solutions obtained analytically and through automation [Bogdanov 2015, Тазиева 2016]. The third case results in software development which can verify a solution and thereby enhance the quality of research. Therefore, this case appears to be the most well-grounded, both theoretically and practically.

New applications analysing the state of solids will specialise in a particular sphere, e.g. in stress and strain analysis of pipelines under season-specific geological conditions. They will also be distinguished by the use of innovative computing technology: cloud resources and coprocessors. These will allow to optimise the running costs of the software developed. The use of available open libraries on computational mechanics of a deformed body is also important for such software development. This paper is focused on the use of libraries providing solution algorithms for solid-state analysis problems in the form of a cloud service.

The software solutions of both research and commercial nature become increasingly complex and resource-intensive. The growing number of such solutions adopt the cloud service model of distribution. This trend is combined by the hardware specialisation: there are systems with GPGPU; with coprocessors, e.g. Xeon Phi and FPGA with OpenCL support; or with specific networking functionality. The system components often need to be developed using programming languages: C++, C#, .Net, Python, etc. Furthermore, such components might impose additional system requirements. Cloud platforms allow to create computing networks employing virtual heterogeneous compute nodes. These nodes run various operation systems and applications.

Today there is no shortage in FEM solvers based on open-source libraries. We have studied some of FEM solvers most cited in academic literature. They are used both in open-source and commercial applications: PETSc, FENiCS, and Trilinos. These widely used libraries enable work with mesh objects, triangulation, and solution of non-linear variational problems. They are also thoroughly described in numerous academic works and technical manuals. The libraries are available under free LGPL licences, which allows their application in both open- and closed-source software. The support is exercised by such reputable organisations as Argonne National Laboratory, Sandia National Laboratories, Simula Research Laboratory, and University of Cambridge.

The libraries allow to develop software using GPU and FPGA coprocessors functionality and to apply MPI in order to distribute workload between compute nodes. The libraries have a remarkable feature: the parallel work of CPUs is accomplished through MPI, not streams. This reveals storing adherence of the developers to high-performance computing. The coprocessors can be managed through a number of interfaces, i.e. OpenCL and CUDA. Coprocessors are mostly employed in solvers dealing with systems of equations, not with geometrical problems. They are capable of enhancing the computational speed up to one order of magnitude [Iakushkin 2015, Iakushkin 2014]. The libraries were tested on hyper-elasticity problems. The experiment indicated they can load all cores of CPUs within an MPI cluster to the full capacity.

The cloud computing is primarily distinguished by its flexibility, which means a large number of compute nodes can be activated on demand. After the task is completed, the nodes can be deactivated. Cloud services providers, such as Microsoft Azure and Amazon EC2, do not allow to activate an MPI cluster in one click: each allocated node must be configured manually. However, the libraries under discussion require high-performance cluster systems, which challenges their use within a cloud platform. The question arises: how can we activate a cluster of the needed capacity and to launch a solution of a specific problem? The development of a service enabling a number of concurrent queries is even more problematic. The difficulty pertains to the monitoring of clusters activation and deactivation; controlling their settings; and storing the results yielded. In other words, any service which employs one of the libraries mentioned should be mainly focused on solving problems through FEM, while every single problem might be time-consuming and require the entire capacity of the system. This is a radical departure from the philosophy of cloud computing, i.e. the on-demand increase of computational capacity.

The goal of the study was as follows: to automatically activate and set up new nodes; and to unite them into an MPI cluster. We have built a number of control programmes for Microsoft Azure cloud application platform which are capable of attaining this goal. In particular, they allow to execute a programme, which is provided by a user, on the computing resource. After the programme is executed, the programme complex returns the computation result and deallocates the cloud resources provisioned for the task. The programme complex was tested on Microsoft Azure platform with the finite element libraries mentioned.

The complex allows to activate the required number of concurrently running MPI clusters in order to solve a particular problem: e.g. problem-solving by multiple solvers or performing computation with different parameters simultaneously. The integrated solution presented is distinguished by the possibility of parallel high-performance computing and flexible use of computing resources mobilised only when needed.

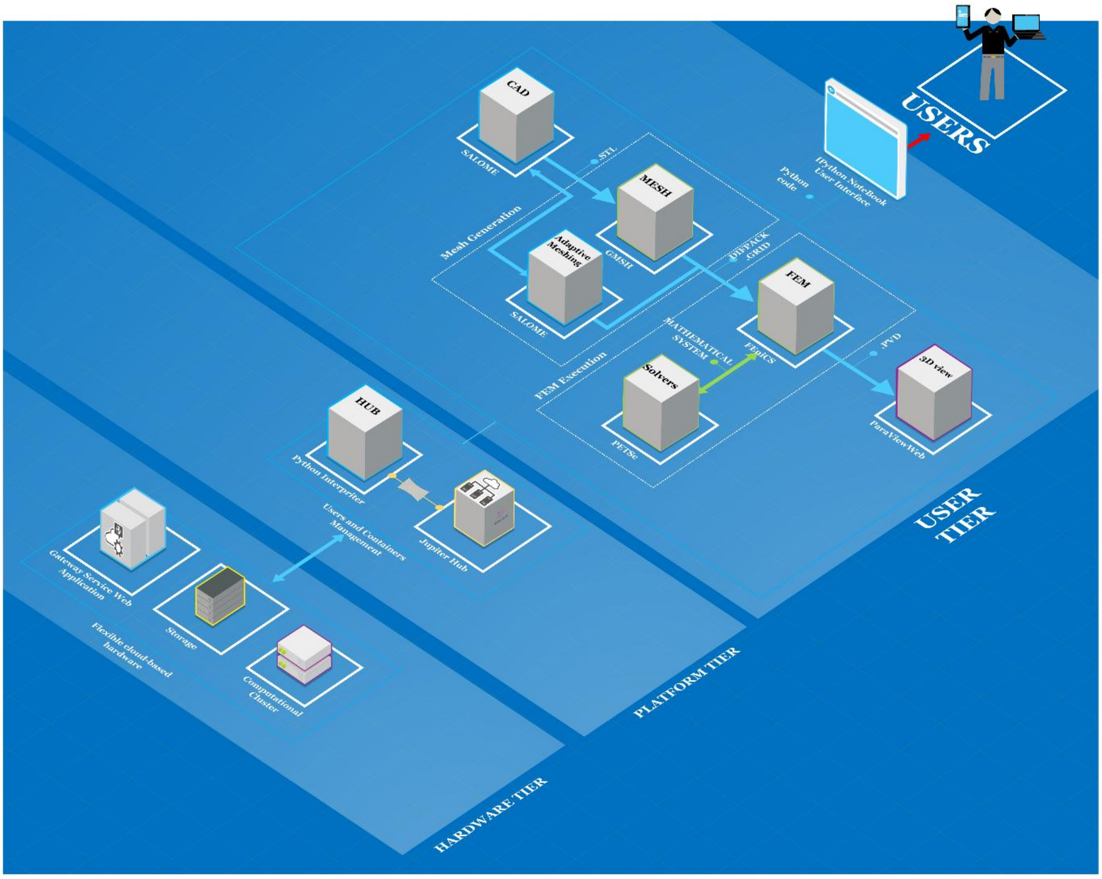
Problem statement

The goal of the study was to automatically deploy and set up new computing nodes and to connect them into a single MPI cluster. We developed a set of control programs to solve these tasks on Microsoft Azure cloud platform. Specifically, the programs allow to run a user-provided computing application on the allocated resources. After the application is completed, the software returns the result to the user and releases the allocated cloud resources. The software complex was tested on Microsoft Azure using the FEM libraries described in this paper.

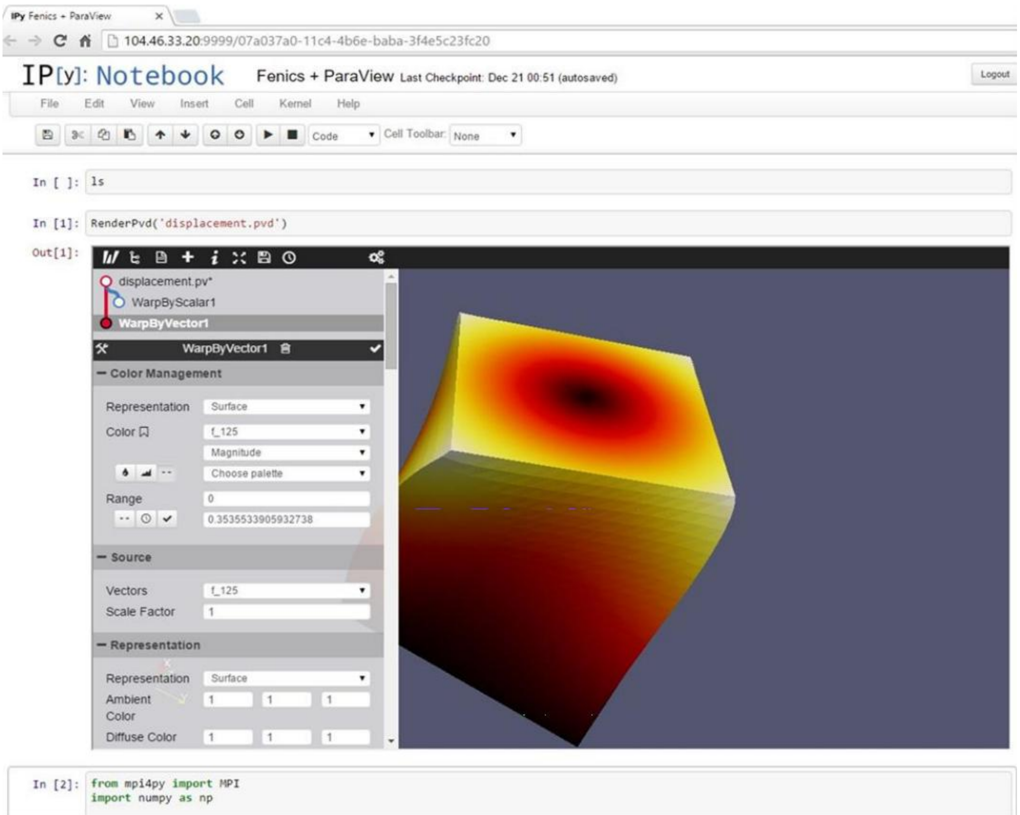
Our Implementation

The architecture of software complex is shown in Picture 1. It allows to simultaneously deploy the required number of MPI clusters to solve tasks. For instance, one task can be solved by different solvers or the calculation can be performed in parallel using different parameters. The solution described here has an important distinguishing feature: it can run high-performance computing tasks in parallel by using the flexibility of cloud systems and allocating computing resources only when needed.

The described system is aimed at development and verification of CAD models from scratch. It integrates geometry creation, mesh modelling and deformation analysis into a single computing system of “system as a service” type. The system is based solely on open source software and allows to use the Python language as well as SALOME, GMSH, FEniCS and ParaView libraries to solve the assigned tasks. The paper describes the system’s architecture and provides an analysis of selected issues related to libraries. It also illustrates the results produced by the developed software—the tool for creating and analyzing CAD models in a web browser (Picture 2).



Picture 1: screenshot of a visualization system demonstrating an object after force was applied to it



Picture 2: Screenshot of a visualization system demonstrating an object after force was applied to it.

References

Bogdanov A.V., Degtyarev A.B., Khramushin V.N. High performance computations on hybrid systems: will "grand challenges" be solved? // Computer Research and Modeling. — 2015. — Vol. 7, No. 3. — P. 429–438 (in Russian).

Iakushkin, Oleg, and Valery Grishkin. "Unification of control in P2P communication middleware: Towards complex messaging patterns." PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON NUMERICAL ANALYSIS AND APPLIED MATHEMATICS 2014 (ICNAAM-2014). Vol. 1648. No. 1. AIP Publishing, 2015.

Iakushkin O.O., Degtyarev A.B., Shvemberger S.V. Decomposition of the modeling task of some objects of archeological research for processing in a distributed computer system // Computer Research and Modeling. — 2014. Vol 7, No. 3. – P. 533-537 (in Russian).

Tazieva L.M., Iakushkin O.O. Creation problems of three-dimensional models and textures from photographs of the object / Control Processes and Stability 2016, Vol. 3(19), № 1, PP. 527-533 (in Russian).