Redesign of PAK's interfaces to fit OSICE and CloudiFacturing requirements

Snezana Vulović*, Vladimir Dunić**, Miroslav Živković**, Vladimir Milovanović **, Jelena Živković**

* Institute of Information Technology, University of Kragujevac, Kragujevac, Serbia

** Faculty of Engineering, University of Kragujevac, Kragujevac, Serbia

vsneza@kg.ac.rs, dunic@kg.ac.rs, zile@kg.ac.rs, vladicka@kg.ac.rs, jelenazivkovic@kg.ac.rs

***Abstract* - Simulation-based optimization integrates optimization techniques into a simulation analysis. Because of the complexity of the simulation, the objective function becomes difficult and expensive for an evaluation. In our example, the simulation considers structural analysis using the finite element method (FEM), the estimation of the welds number, the costs associated with the welding process, and the simplicity of the assembly process. To overcome the computing complexity, we are using OSICE, a comprehensive, cost-effective, and easy-to-use HPC/Grid and Cloud-based optimization service for solving large-scale optimization problems using parallel evolutionary algorithms. For this particular Application Experiment, after the initial definition of technical requirements, PAK input/output interfaces were redesigned to fit OSICE and CloudiFacturing requirements.**

I. Introduction

For a product design process, companies use commercial CAD software with the structural optimization possibility included. It is not enough to design the product to satisfy only optimal technical and material requirements, but also the technological process should be optimal: the number of welds, expenses of welding process, assembly process, and transportation. The optimization problems require large scale computing resources, which are above the personal computer level. The problem's scale is determined by the complexity and type of finite element method (FEM) simulation and the type of optimization. The cloud service is one of the possible solutions.

II. Technical Approach

In the framework of the H2020 CloudiFacturing project - Experiment 13, Milanovic Engineering (MVE) company wants to use the OSICE cloud service for the large-scale optimization of the production process of metal structures. It is an innovative ICT concept which offers the possibility to use extensive computer resource of cloud computing for the computational needs of small and medium enterprises.

OSICE is developed at the Faculty of Science University of Kragujevac (FSUKG). OSICE is a comprehensive, cost-effective, and easy-to-use HPC/Grid and Cloud-based optimization service for solving large-scale optimization problems using parallel evolutionary algorithms. It utilizes an optimization framework WoBinGO [1]. Modular system OSICE consists of WoBinGO, JARE Service, JARE Manager, Binder Manager, monitoring, and management of Work Binder service intended for HPC/IaaS admins for monitoring IaaS instances, cleanup, and specifying users' Quality of Service (QoS) requests. WoBinGO is the framework for parallel execution of the evolutionary algorithms, which consists of JARE - evolutionary algorithms library, and Work Binder - the component in charge of the elastic allocation of the distributed HPC/Cloud resources. JARE Service is the optimization service that exposes functionalities of the JARE library. JARE Manager is user exposed web application for optimization management and monitoring, while the Binder Manager is the web application for the administration.

To simulate the mechanical behavior of metal structures, the FEM based software PAK [2, 3], developed at the Faculty of Engineering, University of Kragujevac (FINK), will be used integrated into the OSICE environment. The architecture of the optimization solution is given in Fig. 1.

OSICE utilizes a parallel genetic algorithm in master-slave mode, executing expensive objective function evaluations on the computing instances launched automatically by the Work Binder component. Work Binder maintains the pool of IaaS instances ready to execute given evaluations. To achieve the platform independence, FSUKG partner deployed the objective function evaluator (including PAK solver and other cost evaluators) in the form of the Docker/Singularity container(s). They also considered CloudiFacturing workflows as a potentially useful deployment form of the evaluator components. Building the evaluator as a CloudiFacturing compatible workflow, a user can customize the evaluation process, using components specific to a particular problem. Establishing a standardized modular approach to the objective function evaluator development has a broader potential to bring more industrial partners into CloudiFacturing.

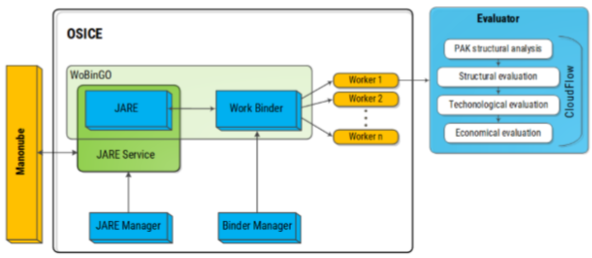

Figure 1. The logical architecture of manufacturing optimization solution

An example of a possible application is demonstrated for the optimization of the transportation pallets for the train parts. The optimization is proposed for the product and the production process. The optimized palette's key factors are materials cost, the time of production process (number of welds, welding cost, mounting process), personal cost, the weight of the structure.

MVE provided the necessary data of the technological process of structure production. They also defined the available resources, the optimization parameters, and the criteria for analyzing the results. The PAK input and output command programs are adopted and redesigned to fit the CloudiFacturing interface and OSICE.

III. INPUT/OUTPUT MODULES OF PAK

MVE engineers developed the 3D model of a structure which needs to be optimized. From the engineering point of view, taking into account possible model variations, several models have to be offered for evaluation, and the best solution should be selected. Based on CAD models, the researchers at the Faculty of Engineering University of Kragujevac developed FEM models using shell FEM elements (Fig. 2). Each part of the CAD model is defined by material and property, which defines the thickness of the element. Colors in Figure 2 (right) match the various thicknesses of the shell elements.

For each CAD model, MVE engineers have defined welds positions. Some of the models allow the optimization of welds (Fig. 3), while the other models need continual weld zones (Fig. 4). Minimal length of the weld is prescribed. The average length of the element is less or equal to the minimal length of the welding segment in case of structures with the welding process optimization.

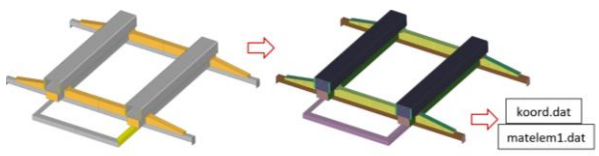

Figure 2. Preparation of finite element mesh by the pre-processing tool

The FEM model with loading and boundary conditions is written in ASCII file with the DAT extension. The input file (DAT) is separated into two files: *koord.dat* and *matelem1.dat* (Fig. 2). The *koord.dat* file is unchanged during the optimization process. For the FEM models with the possibility of welds optimization, the number of nodes is different, so each model has *koord.dat* file. The *matelem1.dat* file is the same for all weld lengths for the specific configuration. The *matelem1.dat* file is changed during the optimization process.

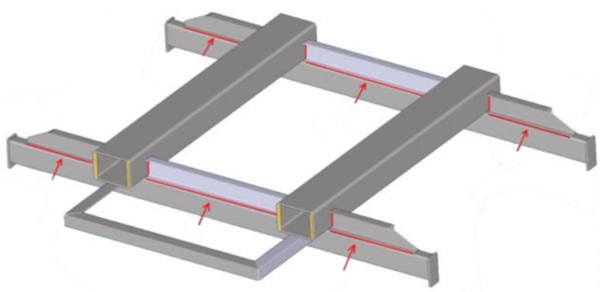

Figure 3. Welds for optimization

For this project experiment, it was necessary to create input/output for PAK software to connect to the evaluator. For that purpose, the files INPUT.CSV, MASS.CSV, MAX_STRESS.CSV, and WELD_MAX_STRESS.CSV are created. For the models with the welding optimization,

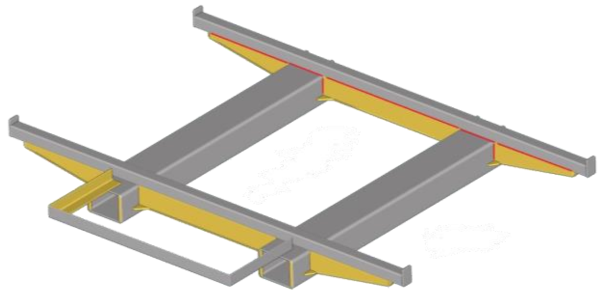

Figure 4. The model with continual weld zones

one additional file is created WELDS.DEF. The file WELDS.DEF contains the data about the elements in welds.

The numerical simulation is processed at the Linux platform. For the numerical analysis, the FEM solver *pakopt* and script shell (wrapper [4, 5, 6]) *ChangeTicknes.sh* are used (Fig. 5). The files with the extension CSV and *ChangeTicknes.sh*, in Figure 5, are input/output modules of PAK (*pakopt*) adopted to fit the OSICE interface.

The shell script is executed by command:

./ChangeTicknes.sh

from the command line. The script reads the data from the input INPUT.CSV file and writes them into the material data of input file *matelem1.dat*. The thickness data from the INPUT.CSV file overwrite the thickness of the element defined in the file with the element data. The file *elemTest.dat* is created. Combining the files *koord.dat* and *elemTest.dat*, the input file *pakOpt.dat* is ready for the simulation. The next step is the execution of the *pakopt*, which reads the names of input and output files from the *inp.txt* file. During the execution of the *pakopt*, for the models with the welding optimization, the file WELDS.DEF is read.

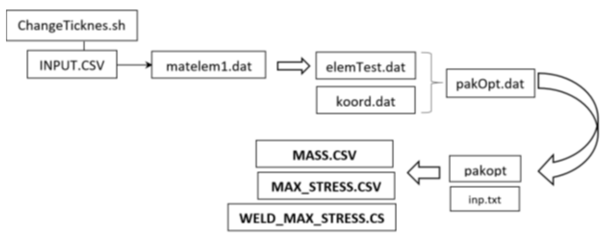

Figure 5. Flow chart of the wrapper

At the end of the computation, *pakopt* prints structure parts masses into the MASS.CSV file, maximal effective stress of structure parts into the MAX_STRESS.CSV file, and in combination with the WELDS.DEF file, the maximal effective stress for each weld, is printed into the WELD_MAX_STRESS.CSV file. The MASS.CSV file's value is used for the economic analysis, while the values from MAX_STRESS.CSV and WELD_MAX_STRESS.CSV files are necessary for structural evaluation. Stress fields of structures with continual weld zones are shown in Figure 6 and Figure 7.

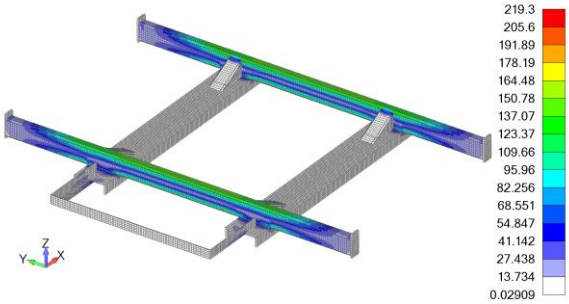

Figure 6. Stress fields of structure 2 with continual weld zones

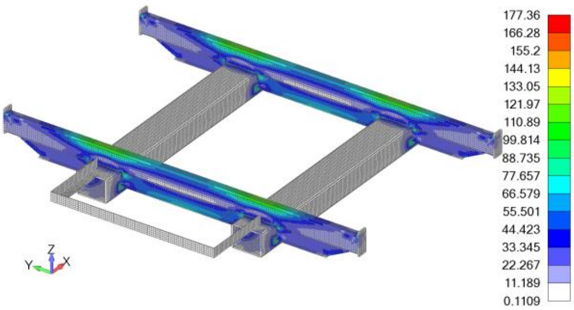

Figure 7. Stress fields of structure 4 with continual weld zones

IV. THE STRUCTURE OF CHANGETICKNES.SH FILE

The INPUT.CSV file contains data about the materials: Young modulus, Poisson's ratio, and the thickness of each property. Commas (,) separate the inserted values. By variation of parameters i.e., Young modulus and Poisson's ratio, it is possible to change the type of the material (steel, aluminum). For the models with the weld optimization, the last value in the file is the configuration number, which defines the FEM model used for the analysis.

The bash script *ChangeTicknes.sh* is used for the data sort and solver call. The bash script is executed in terminal windows. The communication with servers is usually performed by OpenSSH (free network protocol Secure Shell). GNU/Linux systems do not use the extension to define the executable scripts, so the file can be executed if it has permissions defined by:

$ chmod u+k name_of_the_script.

The execution of the script *ChangeTicknes.sh*, reads the data from INPUT.CSV. In the script *ChangeTicknes.sh* in the beginning, the file *templateDat=matelem1.dat* is opened by command cat:

cat $templateDat

Furthermore, we use command *sed* for the parts of the text change [7].

```
sed -r 's/^([ ]*[0-9]*[ ]{4}[1][ ]{4}[0][ ]{4}[0][ ]{4}[0])[ ]{2}8\.[0]{6}(.*)/\1'"$Thickness_1"'\2/g'
```

In the above line, '*g*' option is used in the `*sed*` command to replace all occurrences of matching patterns. The previous code changes the thickness of the element for the material ID 1, and 8.000000 into the value *"$Thickness_1"* from INPUT.CSV file. For each material/property, there is *sed* command in the script. The change of the *matelem1.dat* file is saved as *elemTest.dat* (Fig.5).

For the models with the weld optimization, the *pakOPT.dat* is created based on *koord.dat* and *elemTest.dat* files, what is visible in code:

```
if [ $var = "1" ]
 then
    cat koord1.dat elemTest.dat > pakOpt.dat
          echo $var
 fi
```

The value of variable *$var* of ID configuration is read from the file INPUT.CSV.

V. THE STRUCTURE OF OUTPUT FILES

The MASS.CSV file contains data about ID materials (parts) and mass of material in structures. Commas (,) separate the inserted values.

The MAX_STRESS.CSV file contains data about ID materials and maximum stress in material of structures in MPa. Commas (,) separate the inserted values.

The WELD_MAX_STRESS.CSV file contains data about the ID of welds and maximum stress in welds in MPa.

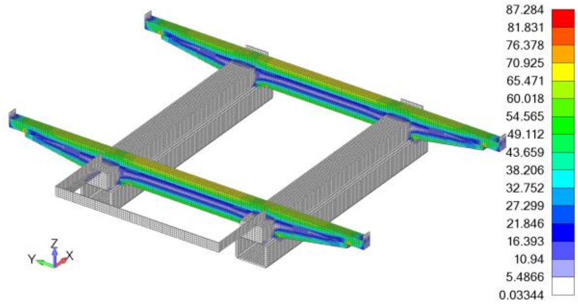

Figure 9. Stress fields of structure 1 with continual weld zones, initial configuration

The file MASS.CSV is the input file of an economic evaluator (Fig. 1). The files MAX_STRESS.CSV and WELD_MAX_STRESS.CSV are input files for structural evaluation. Besides the data obtained by the FEM analysis, for the maximal permissible stresses in the base material, maximal permissible stress in welds, welding process cost related to the length of welds are also necessary for the evaluation. The maximal permissible stress in the weld is related to the type of weld. Based on all these data, the thickness of parts and length of welds is estimated in structures with the welding process

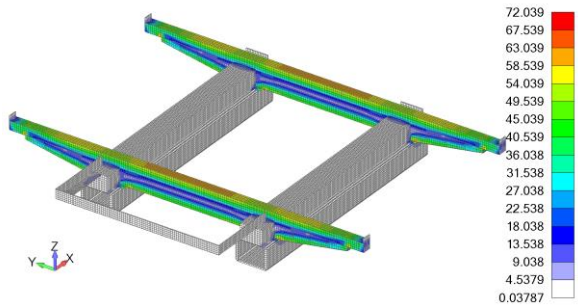

Figure 8. Stress fields of structure 1 with continual weld zones, after optimization

optimization possibility. The evaluated thicknesses are written in the file INPUT.CSV and the script *ChangeTicknes.sh* is rerun. The analysis is repeated until the criteria in the evaluator are satisfied.

The final user collects information at the end of the evaluation process: the thickness of parts, the mass of the optimized structure, and the maximal stresses in each part of the structure. For the models with the welding optimization, the length of the necessary welds and maximal stresses in welds are also available. In Figs 8 and 9, the stress field is presented for one of the models before

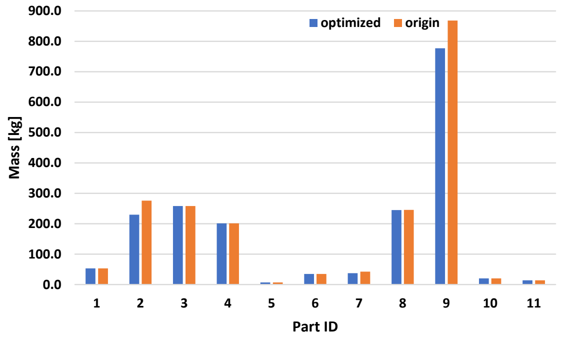

Figure 10. The parts mass before and after the optimization

and after the optimization procedure. For the same model, in Fig 10, the mass change before and after the optimization is presented. The maximal value of equivalent von Mises stresses in the model parts of model 1 is given in Fig 11.

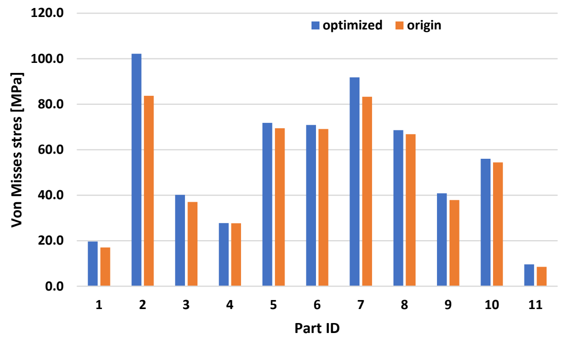

Figure 11. The maximal von Mises stress of the specific parts before and after the optimization

VI. Conclusion

The main task of the script is data preparation in ASCII files as input files for FEM analysis. The script also executes the FEM solver. The input file consists of data about nodes and coordinates (geometry of the model), material parameters, loading, and boundary conditions, and other data necessary for the analysis. Based on the computed results and values of stress in primary material and welds, the material parameters are updated in input files, and analysis is executed again. These scripts are also used for the combination of the text files. As the final result, the provided solution can be used to develop the optimization process further.

Acknowledgment

CloudiFacturing: Cloudification of Production Engineering for Predictive Digital Manufacturing. Horizon2020. Experiment 13: “Optimization of the Production Process of Metal Structures Using OSICE." Grant agreement by the EC no. 768892. Subcontractor of Fraunhofer, Germany, 2019.

The authors gratefully acknowledge partial support by the Ministry of Science and Technology of the Republic of Serbia, grant TR32036.

References

[1] M. Ivanovic, V. Simic, B. Stojanovic, A. Kaplarevic-Malisic, and B. Marovic, “Elastic grid resource provisioning with WoBinGO: A parallel framework for genetic algorithm-based optimization," Future Generation Computer Systems 42, pp. 44-54, 2015.

[2] M. Kojić, R. Slavković, M. Živković and N. Grujović, PAK - Program for structural analysis, Faculty of Engineering, University of Kragujevac.

[3] M. Kojić, R. Slavković, M. Živković and N. Grujović, *Metod konačnih elemenata I, Linearna analiza*, Kragujevac: Mašinski fakultet, Univerzitet u Kragujevcu, 1998.

[4] Y. Liguo, “An empirical study of software market share: Diversity and symbiotic relations,” *First Monday*, vol. 17, no. 8, 2012.

[5] S. H. Hong and L. Rezende, “Lock-in and unobserved preferences in server operating systems: A case of Linux vs. Windows,” Journal of Econometrics, vol. 167, no. 2. pp. 494-503, 2012.

[6] G. Speake, *Eleventh Hour Linux+, Chapter 5 - Using Bash*, Syngress, 2010.

[7] https://linuxhint.com/50_sed_command_examples