Processing Complex Medical Workflows in the EurValve Environment

Bubak, M.[1,2], Gubała, T.[2], Hose D.R.[3], Kasztelnik, M.[2], Malawski, M.[1], Meizner, J.[2], Nowakowski, P.[2], Wood, S.[4]

[1]Department of Computer Science, AGH University of Science and Technology, Kraków, Poland

[2]ACCentre Cyfronet AGH University of Science and Technology, Kraków, Poland

[3]Department of Infection, Immunity and Cardiovascular Disease, University of Sheffield, UK

[4]Sheffield Teaching Hospitals, NHS Foundation Trust, Sheffield, UK

1. Introduction

In this paper we present the outcome of three years of development work in the EurValve project [1] which resulted in the creation of an integrated solution for medical simulations referred to as the Model Execution Environment (MEE). Starting with a definition of the problem (which involves simulating valvular heart conditions and the outcomes of treatment procedures) we provide a description of the high-performance computational environment utilized to process retrospective use cases in order to create a knowledge base which underpins the EurValve Decision Support System (DSS). We also provide specific examples of MEE usage and the corresponding statistics.

2. Definition of the problem

One of the goals of the EurValve project was to elaborate and operate a flexible, easy to use environment for the development, deployment and execution of large-scale simulations, required for learning process development and for sensitivity analyses (a Model Execution Environment; MEE) and for the associated data storage. The simulations themselves require high-performance computing resources and scientific toolkits such as Matlab and ANSYS [2]; however, their outcome, in the form of decision rulesets, is transferred from the research infrastructure to the Decision Support System (DSS), enabling access to EurValve in a clinical setting. The challenge, then, was to create an environment which enables clinical researchers (also referred to as domain scientists) to carry out complex simulations with the use of HPC resources and manage their results in a secure manner, while not requiring in-depth knowledge of interacting with high performance computing infrastructures and distributed storage resources.

At the core of the problem which EurValve addresses is the concept of deriving a heart model (described by a set of parameters) which, when coupled with patient-specific data, such as medical scans, can be used to compute the likely outcome of various treatment regimens. In a clinical setting, however, this computation cannot rely on HPC resources – this is due to the fact that most hospitals do not operate their own HPC infrastructures, and legal restriction prevent patient data from being moved out of the clinical settings for external processing. Consequently, a heart model must be precomputed instead. The initial step is the creation of a Reduced

Order Model – ROM, which involves multiple runs of a full Computational Fluid Dynamics (CFD) simulation. The ROM is then used in the Heart Model, which combines it with the corresponding 0D model. The Heart Model is subsequently used in the full sensitivity analysis pipeline (cf. Fig. 1).

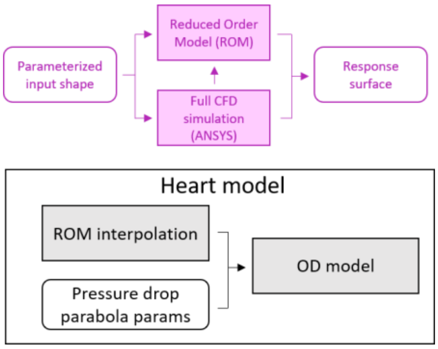
Parameterized input shape
Reduced Order Model (ROM)
Full CFD simulation (ANSYS)
Response surface
Heart model
ROM interpolation
Pressure drop parabola params
OD model

Figure 1 *The EurValve data processing pipeline – from patient data images to DSS-ready heart models.*

Interactions within the Heart Model, which includes the 0D model and ROM are as follows:

- The 0D model includes a ROM representation of the mitral or aortic valve, or both, which relates the flow through the valve to the pressure gradient across it. The purpose of the ROM, which requires significant 3D/4D computations in its development and validation, is to support rapid computation of the cardiac and systemic circulation physiology for a patient-specific valve anatomy when there is no time to produce full 3D/4D analyses.
- For each of the aortic and mitral valves, a ROM describes the relationship between transvalvular flow and pressure gradient as a function of anatomical/geometric parameters describing the valve shape. The systems model uses these relationships, together with cardiac and circulation parameters, to compute the flows and pressures throughout the system for the individual patient in one or more physiological states.
- To calculate the pressure drop, the ROM builder needs to create a response surface which establishes the relationship between the geometrical parameters and flow, and the pressure drop in a form that is understandable by the ROM interpolation tool. To create the response surface, the ROM builder has to run full 3D simulations multiple times with different input parameters. The result of each simulation is a single pressure drop value. The number of runs needed to characterize the response surface depends (non-linearly) on the number of input parameters. This process needs to be run on HPC for the final generation of the model.

3. The Model Execution Environment – architecture and usage

Fig. 2 presents the architecture of the Model Execution Environment as implemented during the EurValve project to facilitate execution of the above mentioned HPC computations. The environment itself consists of a user interface enabling execution of computational pipelines and retrieval of results, along with a middleware layer which takes care of submitting computations to computational resources, monitoring their status and orchestrating execution pipelines.

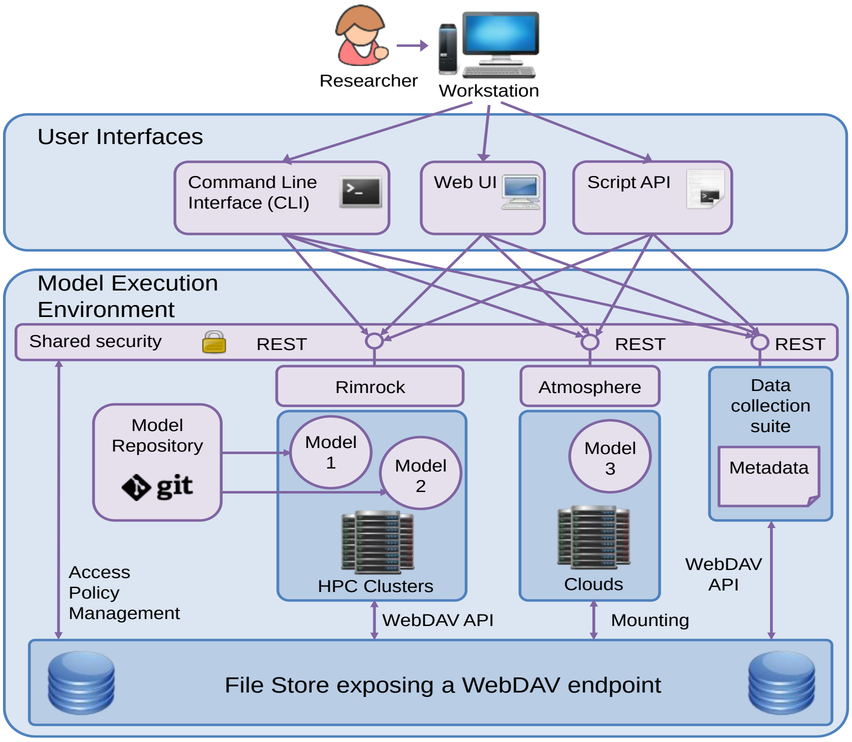

Figure 2 *Architecture of the Model Execution Environment*

The environment provides added value to users of the underlying HPC resources, specifically by:

- Integrating secure data storage repositories (including anonymized data describing patient cases which is derived from medical databases, such as the Trial Connect system deployed at Sheffield Teaching Hospitals);
- Allowing computational pipelines which process this data to be run on HPC resources using a simple user-friendly GUI without requiring domain scientists to become familiar with the specifics of HPC operation – this is facilitated by the Rimrock subcomponent, as seen in Fig. 2;
- Enabling individual steps of computational pipelines to be scheduled for execution on either HPC or cloud-based infrastructures, with the latter facilitated by the Atmosphere subcomponent, as seen in Fig. 2;
- Implementing a shared security layer on top of all resources, with single sign-on access to all parts of the infrastructure (including computations on HPC clusters);
- Providing a range of visualization and data comparison interfaces to enable users to monitor the progress of their computations, visualize intermediate and final results, and compare the outputs of multiple simulation runs.

The Model Execution Environment organizes data and computations into context domains (referred to as *Patients*), while for each Patient an arbitrary number of simulations (referred to as *Models* in Fig. 2) can be performed to generate the ROM described in Section 2. This workflow-like approach lends itself well to a variety of use cases where complex data has to be processed using HPC in the context of a specific entity (patient, project, observation, event etc.)

4. Results

Preparation of heart model data for the EurValve DSS called for processing of a cohort of patients referred to as "retrospective" (i.e. patients who had already undergone treatment for valvular heart conditions, enabling researchers to compare the outcome predicted by DSS under a variety of conditions with the actual outcome as observed for each patient). The analysis of each case was divided in to two computationally intensive parts. The first one, characterization of the valve which extracted the valve coefficients, required running a set of five computational fluid dynamics simulations for each case. The simulation sets took up to 6 hours to complete using 12 cores. The second part, personalization of the valve, was run using 24 cores for the GA analysis. Altogether, the system was used to process approximately 60 distinct patient cases, which involved consumption of over 250 thousand CPU-hours on an HPC cluster – specifically, the Prometheus cluster at ACC Cyfronet AGH [3].

5. Conclusions and future work

As the Model Execution Environment is a generic architecture, capable of supporting a wide range of simulations and HPC studies, not necessarily limited to medical sciences, work is underway to extend its usage to applications which call for simultaneous processing of data across a range of HPC centres. This is being done in the context of the PROCESS project [4] and involves developing support for simulation steps encapsulated in application containers (e.g. Docker and Singularity) which can be freely distributed to multiple computing infrastructures for better parallelization of computational tasks.

Acknowledgments

This work was supported by the EU project EurValve Personalised Decision Support for Heart Valve Disease, H2020 PHC-30-2015 689617 and by the PL-Grid Infrastructure (www.plgrid.pl)

References

1. The EurValve Project website, http://www.eurvalve.eu/ (accessed 28 May 2019)
2. Hoeijmakers M.J.M.M., Wächter-Stehlec, I., Weesec, J., Silva Sotod, D.A., Kasztelnik, M., Hose, D.R., van de Vosse, F.N., Estimation of Valvular Resistance of Segmented Aortic Valves Using Computational Fluid Dynamics, Journal of Biomechanics (submission pending)
3. ACC Cyfronet AGH – HPC infrastructure website, http://www.cyfronet.krakow.pl/13243,artykul,computers.html (accessed 28 May 2019)
4. The PROCESS project website, https://www.process-project.eu/ (accessed 28 May 2019)