Cloud and automated computations in modern personalized medicine - AirPROM project perspective

Michal Kierzynka, Marcin Adamski, Andreas Fritz, Dmitriy Galka, Ian Jones, Dieter Maier, Oleksii Shtankevych, Andrew Wells

***Abstract*—Personalized medicine may be defined in general as a customization of medical approach to an individual patient, involving diagnosis, treatment and other medical procedures. The EU-founded AirPROM project is a prime example of joint cooperation that aims to develop a more personalized treatment in the area of respiratory medicine. In particular, project partners develop models and software tools that help to predict the progression of asthma and COPD (chronic obstructive pulmonary disease) as well as response to treatment for individual patients. However, such development would not be possible without computer science, its methods and technologies. The large amount of data produced for each patient, e.g. lung and airway models, together with complex simulations and effective data/results sharing are just selected challenges that need to be addressed. Therefore, a lot of effort has been spent to integrate the specific software tools used in the project with cloud-based infrastructure that allows scalable storage and computing. The computational automation achieved in the project translates into more time that may be spent on direct patient care. The story presented in this paper proves that personalized medicine is not only a matter for the future but a reality that is already happening.**

***Keywords*—**personalized medicine, asthma, COPD, automation, high performance computing, cloud systems, international projects

I. Introduction

IN the recent years modern ICT technologies have tremendously improved many areas of life, including medicine-related sectors. In particular, methods based on advanced medical imaging, simulation and statistical software tools analyzing large DICOM and genomic data sets have benefited most from this development. As a result, personalized medicine has evolved from being only a future dream to reality, though it is still far from being ubiquitous due to its complexity [1] and costs involved. Nevertheless, the advances in this area would not be possible without a wide support from the ICT sector. Sometimes it takes a supercomputing center to analyze all the data coming from a hospital [2]. Moreover, dedicated computational workflows are needed in order to save time by making the computations as automatic as possible [3], remembering that someone's health depends on them.

One important application area for these techniques is respiratory medicine. Lung diseases such as asthma and chronic obstructive pulmonary disease (COPD) affect the lives of over 500 million people worldwide [4] and cost the European Union alone more than 56 billion euros per year. Even though doctors have access to an immense amount of information and data regarding these diseases, few new therapies have been developed [5], [6]. The AirPROM (Airway Disease Predicting Outcomes through Patient Specific Computational Modeling) project aims to develop tools to predict the progression of disease and response to treatment for individual subjects. The project aims also at building multi-scale simulation models of the whole airway system, as a new way of characterizing asthma and COPD. Ultimately, this leads to a personalized treatment, i.e. the ability to find the best possible treatment for each patient.

In order to make things happen for a large number of patients and still keep the whole process transparent and understandable, the project is assisted by a Knowledge Management (KM) platform connected to a storage system, where the actual data are stored. Using the web portal doctors may browse individual patient's data and start simulations with desired parameters, e.g. ANSYS LungModeller, on a high performance remote computing system using the QCG infrastructure. Moreover, some computational workflows are triggered automatically as soon as the data are available. As a result, the medical staff may concentrate more on medical aspects of the disease and spend more time on patient care. The AirPROM project also focuses on several other ICT-related aspects, e.g. multi-scale computational models of the airways [7], but this paper focuses mainly on the integration of selected software tools and their automation.

The notion of the *cloud* in this paper refers to various aspects. From the user's perspective all the data and computations are located *in the cloud* and may be accessed from a regular Internet browser. Speaking more technically, the patient-specific large data are stored in a typical cloud-based storage system, i.e. OpenStack Swift. The computations, although executed in the grid environment (PL-Grid), may also be seen as cloud computations. After all, cloud computing is a type of grid computing that evolved by addressing the quality

M. Kierzynka is with Poznań Supercomputing and Networking Center and with Poznań University of Technology, Institute of Computing Science, Poznań, Poland, e-mail: michal.kierzynka@man.poznan.pl

M. Adamski is with Poznań Supercomputing and Networking Center, Poznań, Poland

A. Fritz is with Biomax Informatics AG, Munich, Germany

D. Galka is with Materialise NV, Kiev, Ukraine

I. Jones is with ANSYS, Inc., Oxford, UK

D. Maier is with Biomax Informatics AG, Munich, Germany

O. Shtankevych is with Materialise NV, Kiev, Ukraine

A. Wells is with ANSYS, Inc., Oxford, UK

of service and reliability problems, which to some extent are managed by the QCG middleware. As a result, the doctors may perceive the applications used in the project as services. This in turn follows the software as a service (SaaS) cloud model. However, the ultimate goal of this research project is to use these technical advances to bring the idea of personalized medicine to everyday use.

The rest of the paper is organized in the following way. Section II describes the architecture of QCG which is used as middleware for remote high performance computations as well as the two storage systems: one of them was used at the beginning of the project and the other that is currently in use. Section III explains how these facilities are utilized. In particular, Section III-A outlines the integration between individual components of the whole system and describes the KM portal in more detail. Section III-B, in turn, presents the two selected domain-specific software tools developed within the project. Finally, conclusions are drawn in Section IV.

II. Materials and Methods

A. Computational middleware

In order to facilitate the use of advanced high performance computing (HPC) infrastructure for the non-ICT-experts, Poznan Supercomputing and Networking Center (PSNC) has designed and developed special middleware software called QCG (QosCosGrid) [8], [9], [10]. Its main role is to create an unified access interface to different computing resources that are usually managed by various queuing systems, like TORQUE [11], Slurm [12], etc. QCG offers highly efficient mapping, execution and monitoring capabilities for different types of application scenarios, such as parameter sweep, workflows, MPI or hybrid MPI-OpenMP. The QCG middleware allows also the large-scale applications, multi-scale or complex computing models to be automatically distributed over a network of computing resources with guaranteed quality of service (QoS). Finally, the middleware also provides a number of unique features, e.g. co-allocation of distributed resources or advance reservation.

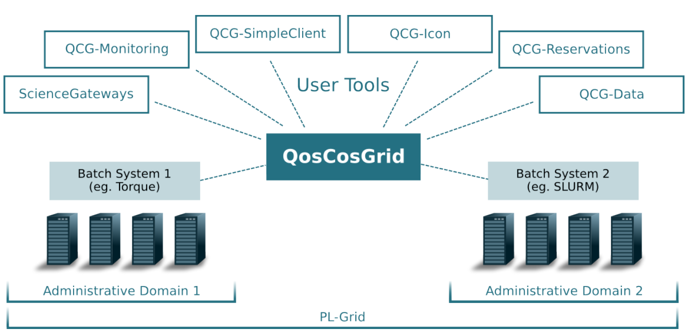


Fig. 1: A simplified architecture of the QCG middleware. Different user tools may use the same unified interface to access various computing resources that are possibly managed by different resource managers.

Figure 1 presents a simplified architecture of the QCG middleware. In general, the middleware is divided into two logical levels: grid domain and administrative domain. The former, i.e. grid-level services, control, schedule and generally supervise the execution of end-users applications, which may be spread across independent administrative domains. The administrative domain represents a single HPC cluster or data center participating in a certain grid or cloud environment by sharing its computational resources. The logical separation of the administrative domains results from the fact that they are owned by separate institutions which may still control its own resource allocation and sharing policies.

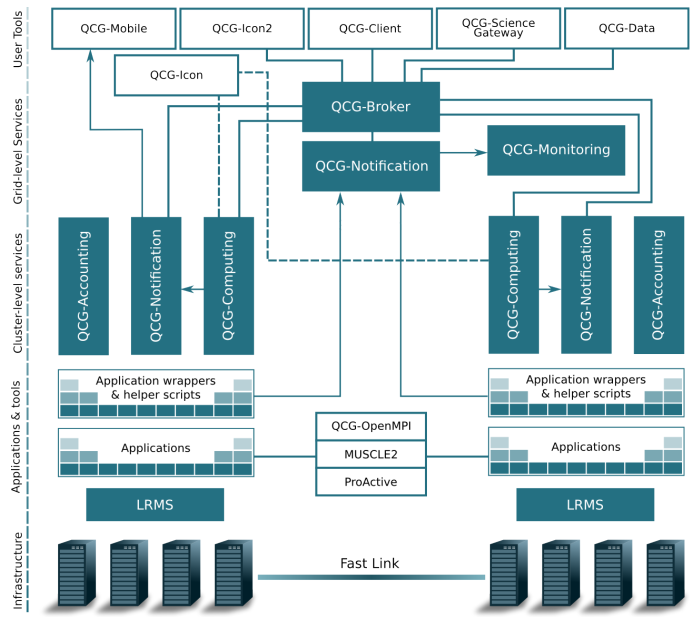


Fig. 2: A more detailed architecture of the QCG middleware illustrating the location of individual services like QCG-Computing, QCG-Notification or QCG-Broker.

Figure 2, in turn, presents a bit more detailed architecture of the QCG middleware. A key component of every administrative domain in QCG is the QCG-Computing which provides the remote access to queuing systems resources. It supports advance reservations and parallel execution environments, e.g. OpenMPI, ProActive and MUSCLE [13]. Moreover, it supports QCG Data Movement services for automatic management of input and output data transfers. QCG-Notification is another service at the administrative domain and as the name suggests is responsible for notifications, e.g. once a job has finished or failed for some reason. It is worth noting that administrative domain services are tightly coupled with the grid-level services. One of the key grid service is QCG-Broker which is a meta-scheduling framework controlling executions of applications on the top of queuing systems using QCG-Computing services. Based on dynamic resource selection, advance reservation and various scheduling methodologies, QCG-Broker efficiently deals with various meta-scheduling challenges, e.g. co-allocation, remote job control, job migration, load balancing etc. The latter was identified as especially important in modern cloud environments [14].

The QCG middleware is used in the Polish grid infrastructure, called PL-Grid, which consists of five large supercomputing centers, namely: PSNC, Cyfronet, ICM, TASK and WCSS.

Moreover, it is also used in the European Grid Infrastructure (EGI) platform and the number of its users is still growing.

The ease of use for the end-user is realized through several client interfaces, cf. Figure 2. One of them, used in the AirPROM project, is called QCG-Icon. It is a lightweight application for Windows, MAC OSX and Linux platforms, aiming to provide transparent access to applications installed on PL-Grid resources and made available via QCG services. Apart from a long list of pre-installed applications, like MATLAB, NAMD, ANSYS, Gaussian, etc., the user may easily submit, monitor and control a job defined in a form of a bash script, i.e. perform any kind of HPC computations remotely without explicit logging into the actual head node of the computational cluster. Since the QCG-Icon client may be parametrized, it is used in a semi-automatic way in the project. The user is presented with a choice of the computational resources, while all the patient-specific input/output data is defined automatically. This greatly reduces the burden associated with processing multiple subjects.

B. Modern storage for medical data

1) Platon U4 system: The AirPROM partners tend to generate a lot of patient-specific data, e.g. DICOM images of the lung lobes and airways, or the genetic data. Therefore, there is a need for a large, secure, reliable and fast storage system available to all the partners in the project. At the beginning of the project (early 2011) cloud-based storage systems, like OpenStack Swift, were still it their infancy, and for this reason PSNC proposed to use an archiving service called Platon U4, which was developed as a service within PL-Grid infrastructure and was already stable at that time. The main assumption was to use this reliable storage until the cloud-based systems become more mature.

The Platon U4 system [15] is a several PB (petabyte) storage system based on tapes and traditional hard disk drives which act as a cache system. The data are replicated geographically and therefore are safe with respect to physical damage of parts of the system. It is also very secure due to fastidious security policy. The security-related issues are of utmost importance when it comes to medical data storage [16]. For example, in Platon U4 every user must be authenticated with a personal X.509 certificate and may access data only from a given location. Furthermore, all the transfers are encrypted. The data is usually accessed with SFTP protocol, using WinSCP, FileZilla or SSHFS client tools. However, the storage is also accessed by batch computations that are run on HPC machines. In this case the authentication is done via personal X.509 proxy certificate, which is generated automatically by QCG-Icon upon job submission. Summing up, some of the main properties of the Platon U4 system include:

- high reliability and availability: automatic and transparent replication of data and metadata,
- high storage safety and security:
  - end-to-end data encryption,
  - data integrity control,
  - SSL-protected data transmission,
  - security procedures employed at storage sites,
- fast access through 10Gb/s optical fiber,
- disk and tape nodes: 2,5+PB of tapes + 0,2PB of disks,
- easy access via: SFTP, WebDAV, GridFTP, sshfs,
- X.509 certificates support.

Moreover, the data on the storage system have a specific structure, hierarchy and naming convention. For example, every subject has a separate directory in which every partner has its own subdirectory. Therefore, it is easy not only to browse the data manually, but also it is a convenient data structure for the software to read/write the data from/to a specific location. This is crucial from the automation perspective.

2) OpenStack Swift: The notion of cloud computing has been one of the key aspects of the AirPROM project from its very beginning. An integral part of any cloud is the storage system. It plays an important role also in the AirPROM project. PSNC has been working on a cloud-based storage system for several years, and by the end of 2013 also started to introduce it to the AirPROM project. The main goal of this transition was to deliver a storage system that could fit the project requirements even better than the abovementioned Platon U4 system.

Some of the requirements for the target storage system have been defined as follows: data integrity, redundancy and high availability, role based access control and ease of integration with the applications and the Knowledge Management (KM) platform. The Platon U4 system, despite its obvious advantages, lacks some of the desired properties, e.g. the integration with the KM portal is somewhat difficult, as the user needs to have two separate accounts: one to access the portal and the other for the storage. Also different software tools need to be used when the user wants to browse information in the KM and at the same time interact with the data, e.g. download or upload DICOM images. Programmable integration of software tools with the storage was a bit complex too. For these reasons, PSNC proposed to seamlessly switch to cloud-based OpenStack Object Storage Swift software for the storage purposes. The new storage system was welcomed as it offers more possibilities for the AirPROM partners.

When it comes to the level of security in the OpenStack Swift storage system, it is very similar to what is offered by Platon U4. On the other hand, the OpenStack Swift is very different from the Platon U4. First of all, it features a flat directory structure. Fortunately, subdirectories may be easily emulated by the "/" sign in the name of a file (here known as object). When it comes to authentication, only user name and password combination are accepted, and there is no support for X.509 certificates. Therefore PSNC has also developed additional custom authentication mechanism based on certificates. This was needed as the results of automatic computations/workflows are uploaded using proxy certificates and not with user and password credentials. Interestingly, once the user is authenticated, they receive a token which may be used to perform any storage operations without additional authentication for a given period of time, usually 24 hours. However, most importantly the OpenStack Swift may be accessed using REST API over HTTPS. This allows for relatively easy integration of the storage system with other software tools that are used in the project, possibly using one of the already existing libraries for Java, Python, Ruby or other programming languages. This means that applications may read the input data and store the results directly to the remote storage system as if they were using a local hard drive (an example is presented in Section III-B). Likewise in the case of the Platon U4 system, 10GB/s connection guarantees a high throughput data access. Furthermore, the KM portal is now enabled to read the real content of the storage system much more easily than it was before.

Additionally, the redundant and highly scalable architecture of the OpenStack Swift storage system facilitates also automatic data and metadata replication as well as maintaining high availability of the system. Apart from the security issues, these properties are crucial for medical data too. Moreover, the architectural design allows multiple automatically synchronized instances of the storage system to coexist, possibly in different geographical locations. This feature may be especially desired in worldwide projects, when latency related issues pose a real challenge. However, this particular feature is not used in the AirPROM project, mainly because all the partners are located in Europe.

III. Results and Discussion

A. *Whole system integration*

1) High level overview: Both the high performance storage system and the computational resources are located at PSNC. One of the advantages of their close proximity is the low latency associated with data access by software tools running on clusters. As already described, the computational resources are accessed by the QCG middleware, which means that the software could be, if needed, seamlessly scaled out by deploying on other HPC facilities in the PL-Grid. Currently however, no such need was reported. Nevertheless, it should be stressed that the possibility to scale out the system without changing its architecture is a vital property allowing the project to grow, and ultimately serve even multiple hospitals.

Figure 3 presents a high level scheme of the integration between systems. The dark blue clouds represent the cloud-based storage and the KM portal. The latter has full access to the storage system, including patient-specific data listing, reading and writing. Thanks to the knowledge regarding the availability of data, the KM portal is able to trigger some of the computations automatically, e.g. the Mimics Robot. The computational applications, presented on the light blue clouds, are installed at PSNC and have a high performance network connection to the storage system. They may safely access the storage thanks to the certificate and token authentication mechanisms. Consequently, the results of simulations are uploaded automatically to the storage system and become visible to the user in the KM portal. Moreover, further application triggering is possible as well. The KM portal is an important software part of this workflow and therefore is described in more detail in the following subsection.

2) Knowledge Management Portal: The Knowledge Management portal is a web-based central access and data source in the AirPROM project. It integrates clinical and experimental data (GWAS, MRI, gene expression experiments) with information from public resources (see Table I). The clinical data

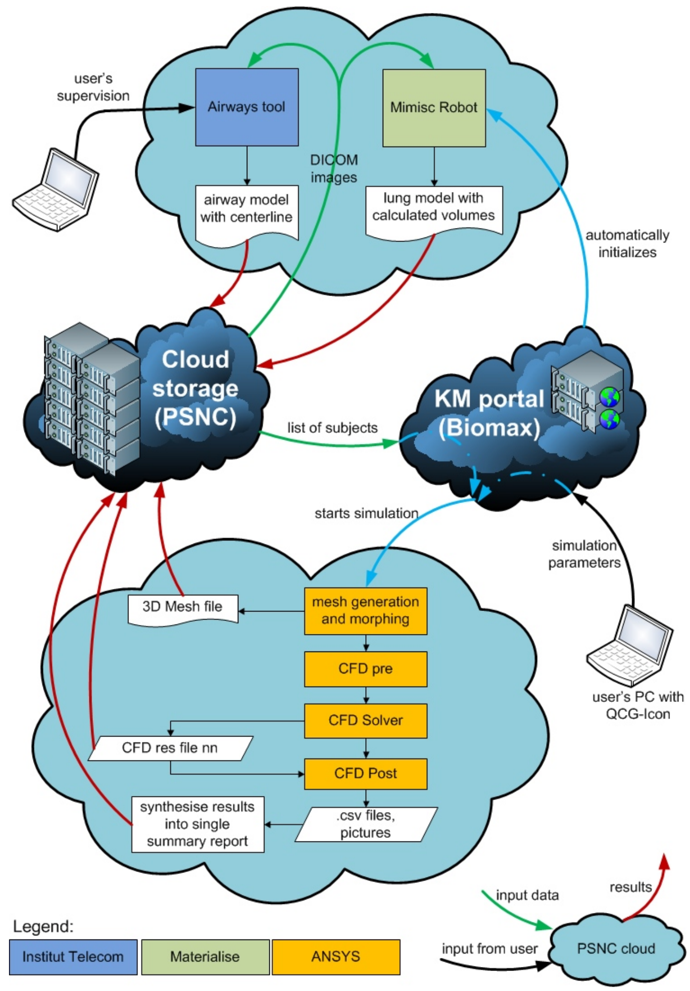


Fig. 3: A high level overview of the integrated system. Individual clouds represent: the storage system, the computational tools deployed in the HPC environment (PSNC cloud) and the Knowledge Management platform.

comes from different clinics and studies. Integration of other data sources can be easily implemented in the KM portal as well.

TABLE I: Publicly available databases and ontologies which are integrated into the AirPROM KM portal installation.

Database	Description
GenBank	nucleotide data bank
RefSeq	nucleotide, protein database
UniProt	protein database
PubMed	literature abstracts
Entrez Gene	gene dictionaries
LocusLink	nucleotide data bank
PDB	3D structure data bank
GO	ontology for gene products
NCBI taxonomy	species taxonomy

Based on semantic mappings, the clinical world of diagnostics as well as physiological and laboratory attributes are connected with molecular experimental measurements and computational models which describe biological processes. Large scale data such as images are stored in the OpenStack

Swift environment while the corresponding metainformation i.e. the documentation of the data as well as access rights (ACL) are managed within the KM portal. The ACLs can be adjusted very precisely based on projects, groups and personal access rights. The metadata are retrieved during data upload procedure as well as during the simulation process. In addition, HPC resources connected directly to the OpenStack storage provide the horsepower for expensive computational simulations. Based on web services the OpenStack, KM portal and different simulation programs integrate and provide a unified experience to the user. The graphical user interface of the KM portal allows users to search for specific patients by their associated clinical as well as experimental information. The user can subsequently start imaging based simulations from web-start clients that, if needed, receive their authorization from the KM portal. The required data is automatically transferred from the storage system to the HPC machines for simulation. In addition, the KM portal controls availability of new image data on the OpenStack and automatically starts image segmentation processes on the HPC resources which results in additional information that in turn is searchable in the KM portal. The graphical overview of the system is presented in the Figure 4.

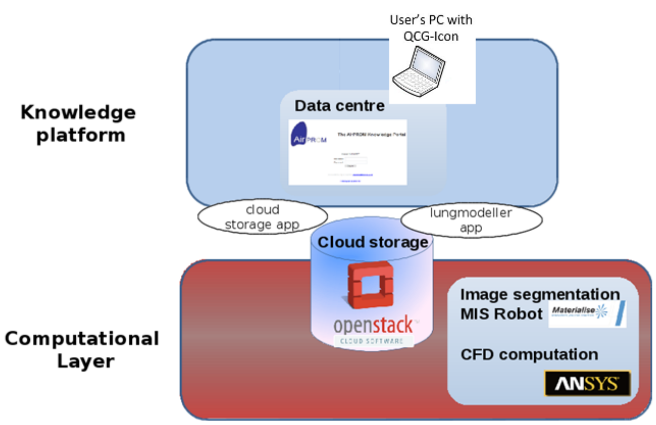


Fig. 4: A diagram presenting lung simulation environment which consists of multiple connected components: KM portal, OpenStack Swift storage, QCG managed computational resources, web-start applications and the domain specific applications.

Therefore, the KM portal acts as a central point of information about data available in the project and their semantics. With this information, the KM portal may automatically trigger computations that do not require human supervision, and the other simulations may be launched manually. Moreover, in the course of the project the KM portal has also become a central data access point. Thanks to the REST API of the OpenStack Swift, the user may browse the content of the storage using the KM portal. They may also upload or download any object using a web-start application that was developed especially for this purpose. Importantly, the application was designed to be able to resume broken uploads and was optimized for efficient retrieval of objects metadata. As a result the user does not have to use any third-party tools for data transfer, unlike at the beginning of the AirPROM project.

B. Example applications

This section presents two example software tools that are used in the project and have been integrated with both the storage system and the KM portal for either automatic or semi-automatic runs in a HPC environment at PSNC.

1) Lung lobe volume computation: The segmentation of lungs is an important and fundamental step for many clinical applications. Analysis of volume and geometry of lungs tissue plays significant role in diagnosis and treatment of patients suffering from pulmonary diseases including cancer, asthma and chronic obstructive pulmonary diseases. Fully automatic lungs segmentation algorithm was developed in Materialise to process thoracic CT scans. The segmentation is done in several steps. In the first step parenchyma and airways are detected as a single mask by separating them from the outer air which could be present in the image. In the second step we apply trachea and lower airways detection algorithm which results in separation between lungs and airways on the B-branches level (which have "B" labels according to IKEDA nomenclature [17]). The airways segmentation algorithm, which is a part of the automatic lung segmentation workflow, does not aim for a deep segmentation of airways but is rather targeted to reach the diameter of lower airways where the separation between airways and lungs is optimal for further algorithmic steps. Even after separating airways from the lungs it is still possible that left and right lungs are connected either in anterior or posterior area. It may happen in the case of diseased patients due to over inflation or because of partial volume effect during the scanning procedure. Therefore, as the third step in the case of connected lungs (after airways subtraction), we apply watershed segmentation algorithm to accurately separate the two lungs. Depending on the need for the watersheds step, processing time may vary significantly on different cases. Detected watershed voxels are finally used to separate the lungs. The overall result of the automatic lungs segmentation is the mask of the right lung, the mask of the left lung and the airways mask, which are further used to generate 3D models.

Materialise Mimics Robot is a server-side solution developed for unattended processing of large medical data sets. In particular, Mimics Robot is used for preforming the above-described automatic segmentation of medical images coming from e.g. CT, MRI, micro-CT, CBCT, 3D Ultrasound and Confocal Microscopy. The use of such imaging methods was demonstrated to particularly improve diagnosis in personalized medicine [18]. The resulting 3D models may be saved in different formats for a variety of medical and research applications. Moreover, the software is capable of measuring certain parameters of 3D models such as volume, dimensions etc. These results can be then inserted in external databases or saved locally for a later analysis. Mimics Robot is capable of supporting virtually any type of input and output storages with miscellaneous authentication methods. Storage types include but are not limited to NAS, FTP, database, OpenStack etc.

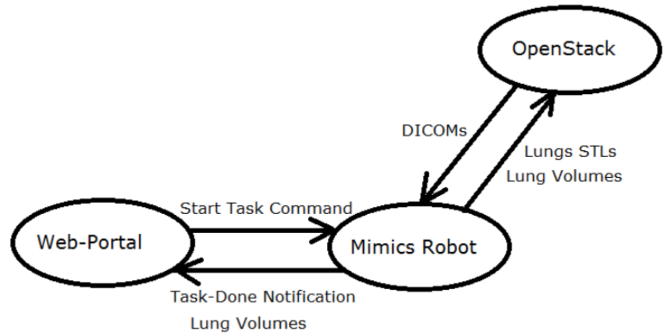


Fig. 5: An overview of the Mimics Robot workflow in the AirPROM project. The computations are started by the KM portal and the communication with the remote OpenStack storage system is done automatically. KM portal is informed once the results are uploaded to the storage and ready to use.

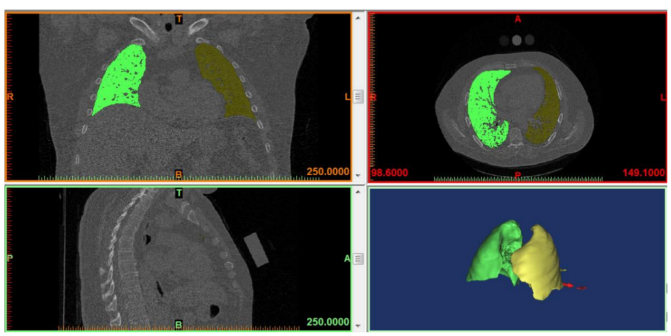


Fig. 6: The visualization of the lung lobe geometry segmented by the Mimics Robot. The left and right lung volumes are calculated correspondingly.

The solution was designed to be scalable as it includes a built-in load balancer which takes full advantage of multi-core hardware and even multi-server network environments. The start of data processing, usually referred to as a Mimics Robot Task, can be initiated by end users via built-in Graphical User Interface or over a number of listening interfaces which can be easily integrated with external web portals. When Mimics Robot Task completes, the external web portal can be notified via web-service call mechanism or, alternatively, user can be notified with an email message.

The Mimics Robot deployment in the AirPROM project is shown in the Figure 5. In this setup the KM portal (Web-Portal), which is geographically located in Germany at Biomax, starts the Mimics Robot Tasks over the Internet via a web-service call. The Start Task Command identifies a study in the OpenStack storage which needs to be processed by Mimics Robot. Mimics Robot and the OpenStack storage are located in Poland at PSNC which enables relatively fast data transfer. Mimics Robot Task performs token-based OpenStack authentication, downloads relevant DICOM images, generates a Mimics project file, performs automatic lungs segmentation, measures its volumes and finally uploads resulting files to the OpenStack storage. The resulting models of lungs (in STL format) that were segmented fully automatically are shown in the Figure 6. Additionally, Mimics Robot sends a ”Task-Done” notifications and lung volumes data to the KM portal via a web-service call. This way the KM portal may make these data available to interested users.

2) Airways resistance simulation with CFD: Within the AirPROM project, the LungModeller application has been developed as an ”Extension” to the ANSYS Workbench and its CFD software, to automate the computation of the air flow in lungs, on a patient-specific basis. It has been designed from the outset to exploit cloud computing, so that the computationally expensive parts can be run using high performance computing servers, using lung geometry obtained from segmented CT scans which are stored on a centralized repository.

The user interface of the application is a web-start program written in Java. The application outputs a text file describing the inputs to the process, and calls a batch program to start the workflow engine. The workflow is based on the ANSYS Workbench platform [19], and is implemented using Iron Python scripts, complemented by application-specific scripts. The workflow can be run locally, or use the QCG middleware system to submit the process to the HPC cloud, and upload the relevant data files to the storage system. Alternatively, it can be initiated from the Knowledge Management system which points to specific lung geometries located in the storage, in which case the QCG is also used to start and monitor the computations. Importantly, the communication with the storage system is performed in an automatic way and its complexity is hidden to the end user. The complete workflow of the integrated LungModeller application is presented in Figure 7.

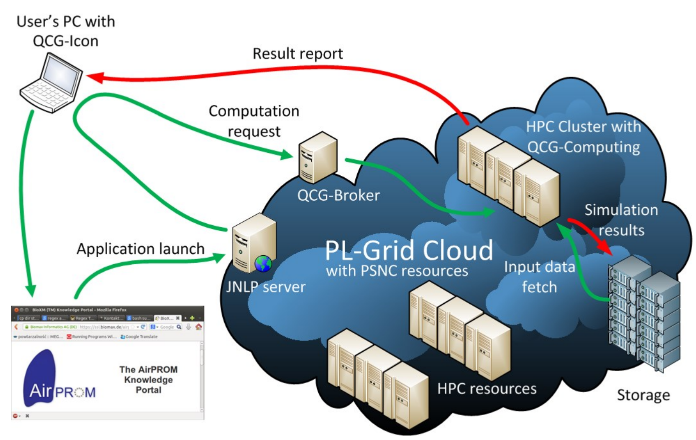
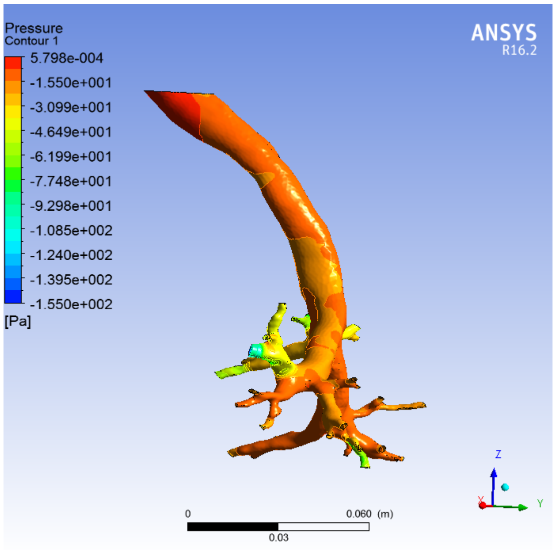


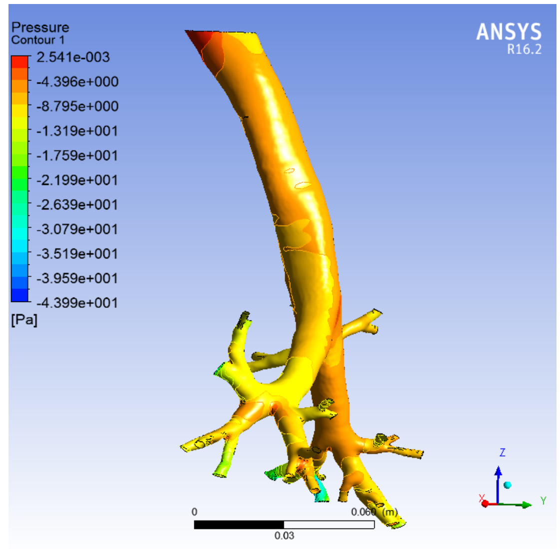
Fig. 7: A simplified diagram presenting the data flow of the LungModeller application integrated with the KM system, QCG middleware, PL-Grid infrastructure and the storage system.

The features of the CFD patient-specific workflow are:

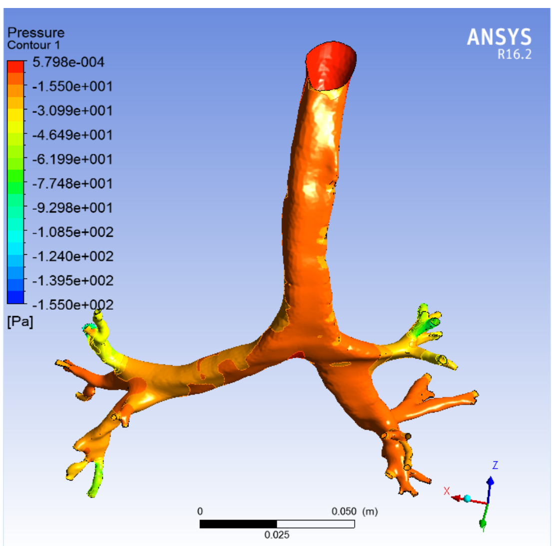
- Import of surface description meshes created by the segmentation software.
- The core of the Framework is based around the ANSYS WorkBench II workflow engine, using Iron Python as the scripting language. Future developments of ANSYS software are based around this workflow engine, and so it



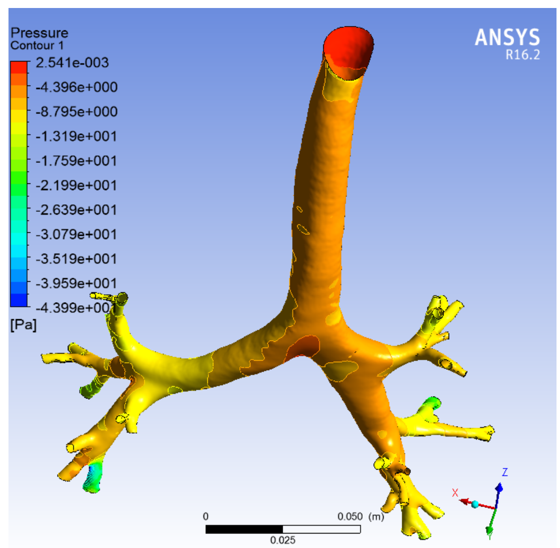
(a) Side view, FRC



(b) Side view, TLC



(c) Rear view, FRC



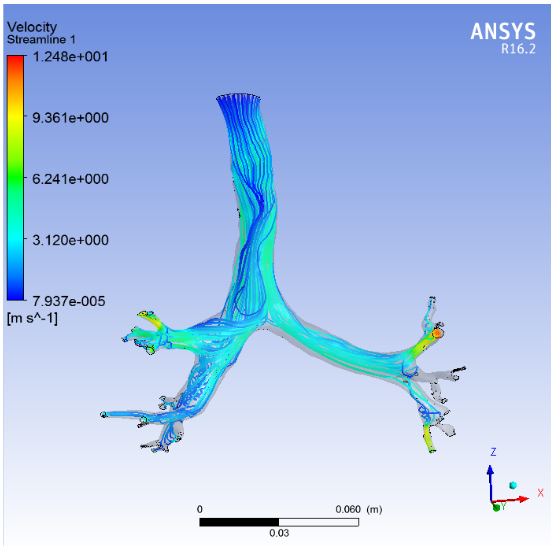
(d) Rear view, TLC

Fig. 8: Snapshots of the airways, color shaded by pressure differences, taken from the 3D viewer.

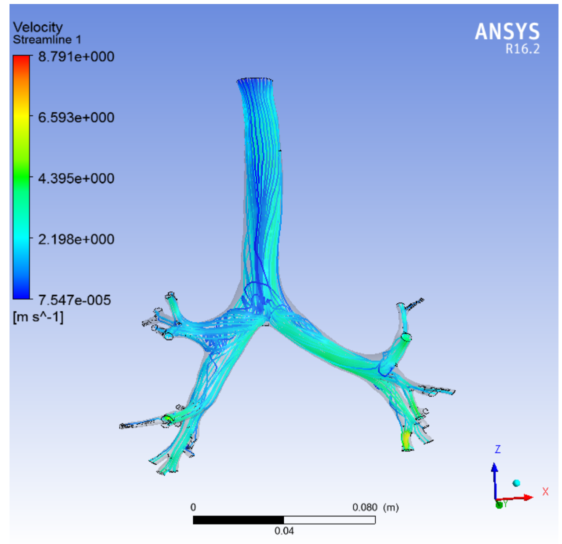
will be easy to update it to include new ANSYS software developments.

- Automated workflow, from GUI to reporting.
- Addition of boundary labels and boundary conditions, according to the project requirements.
- Optional creation of a skeleton, a network description of the branching structure of the airways. This is used to identify different parts of the lungs for post-processing, in particular to allow the aggregation of important clinical indicators such as flow resistance, on a lobe-by-lobe-basis. These can also be tracked in time for cases where the full transient cycle is included in the simulations.
- Volume meshing, with different options for accuracy and robustness.
- Options for using several ANSYS CFD packages, initially Fluent and CFX, to enable cross-validation of results and models.
- Automated reporting, to produce an html document with the outputs needed by clinicians and for accuracy assessment and validation. These can be viewed directly by the end user, without needing to download large results files. They include a 3D representation of the results, which can be viewed in a free 3D viewer, as well as tabulated values of the flow resistance, an important clinical indicator of the degree of severity of asthma.

Figure 8 shows snapshots from the 3D representation of the surfaces of the airways, color coded by the pressure difference from the average value across the trachea. These are from the same patient, an asthmatic, with steady state simulations of inspiration carried out at the two ends of the breathing cycle, when the lungs are inflated, at Total Lung Capacity (TLC), and deflated, at Function Residual Capacity (FRC).



(a) FRC



(b) TLC

Fig. 9: Instantaneous streamlines showing the presence of swirling secondary flows.

These show quite large differences in the shape of the airways, with a pronounced curvature along the trachea. The shape of the cross-section of the trachea also changes remarkably, from circular at TLC, and with a pronounced indentation at the back of the trachea at FRC, due to the horseshoe shaped cartilage bands in the trachea. The results at FRC also indicate larger changes in the pressure differences due to the larger resistance of the smaller airways at FRC.

The increased curvature of the airways also has a significant effect on the detailed flow patterns at FRC. The main flow direction is along the airway passages. However, the curvature of the airways give rise to secondary flows causing the flows to spiral. Figure 9 shows instantaneous streamlines for the flow. At TLC, the main secondary flows are created at the junctions of the airway branches. However, at FRC, the curvature of the trachea causes a much more pronounced secondary flow in the trachea itself. This could have a significant effect in the effectiveness of a dry particle inhaler, for example, as the drugs could deposit higher up the airway tree.

As well as a graphical representation of the flow patterns, the automated reports contain tabulated values of the flow resistance: overall (global), the individual lobes, the trachea (central) and the lobes combined (all peripheral). Figure 10 shows the resistance values at TLC and FRC for the above case, quantifying the increase in the resistance due to the smaller airways at FRC. Most of the studies of flow airways have been carried out assuming steady breathing using the geometry at TLC. The workflow described here permits both steady state and transient flows in static geometries in an automated fashion, enabling a large number of studies to be carried out on the data sets from the AirPROM project. In reality, of course, the airway geometry changes between TLC and FRC during the breathing cycle. An extended workflow has been prototyped to include the motion of the lungs during the breathing cycle. As a result, the computed resistances can be used to predict the results that arise taking the motion of the lungs into account.

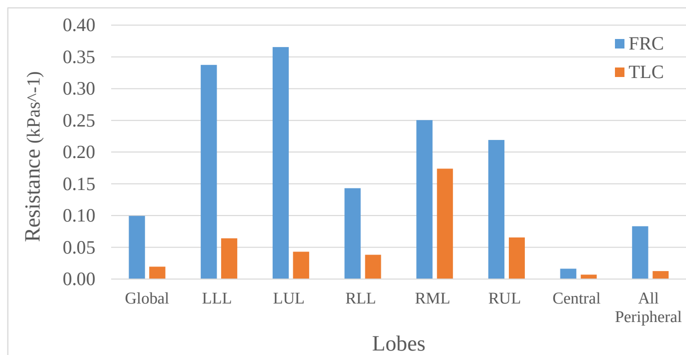


Fig. 10: Bar Chart with the resistances of the airways at TLC and FRC: total resistance of the simulated lungs (Global), Left Lower Lobe (LLL), Left Upper Lobe (LUL), Right Lower Lobe (RLL), Right Middle Lobe (RML), Right Upper Lobe (RUL), Central trachea and All Peripheral total resistance excluding the trachea.

IV. Conclusions

The joint work of several partners within the EU-founded AirPROM project has proved that the automation in modern personalized medicine is not only possible but can also greatly facilitate everyday work of medical specialists. The doctor may browse well-structured data of an individual in the KM portal to better understand cause and effect sequence for each patient. Additionally, they may analyze the patient-specific source data using the direct connection of the KM portal with high performance OpenStack Swift storage. Moreover, the possibility of running a simulation of air flow in lungs and other software tools in a reasonably short time using HPC environment gives additional value to the whole system. The AirPROM project may be therefore perceived as a prime example of how the computer scientists may help to develop a means to improve the overall experience of modern personalized medicine.

V. Conflict of Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgements

This work was funded by the EU Seventh Framework Programme FP7/2007–2013 under grant agreement no. 270194 and is presented on behalf of the AirPROM consortium (www.airprom.eu). This work was also supported by the PL-Grid Infrastructure in Poland.

References

[1] S. Boccia, P. Boffetta, P. Villari, *Screening for Complex Diseases and Personalized Health Care*, BioMed Research International, 2015

[2] Y. Kadooka, T. Iwamura, M. Nakagawa, M. Watanabe, *Developing Biomedical Simulations for Next-generation Healthcare*, FUJITSU Sci. Tech. J., Vol. 51, No. 3, pp. 69-76, 2015

[3] N.S. Labeeb, A. Hamdy, Iman A. Badr, Z. El Sanabary, A.M. Mossa, R. Khattab, *Automatic Classification of the Severity Level of the Retinal Ischemia*, WSEAS Transactions on Biology and Biomedicine (12), 44-50, 2015.

[4] World Health Organization, *2008 World health statistics*, Geneva, Switzerland: WHO.

[5] Hogg JC., State of the art. *Bronchiolitis in chronic obstructive pulmonary disease*, Proc. Am. Thorac. Soc. 3, 489-493, 2006

[6] Pavord ID, Korn S, Howarth P, Bleecker E, Buhl R, Keene O. *Mepolizumab for severe eosinophilic asthma (DREAM): a multicentre, double-blind, placebo-controlled trial*, Lancet 380, 651-659, 2012

[7] Burrowes KS, De Backer J, Smallwood R, Sterk PJ, Gut I, Wirix-Speetjens R, Siddiqui S, Owers-Bradley J, Wild J, Maier D, Brightling C, the AirPROM Consortium, *Multi-scale computational models of the airways to unravel the pathophysiological mechanisms in asthma and chronic obstructive pulmonary disease (AirPROM)* Interface Focus 3: 20120057, 2013

[8] B. Bosak, P. Kopta, K. Kurowski, T. Piontek, M. Mamoski, *New QosCosGrid Middleware Capabilities and Its Integration with European e-Infrastructure*, In eScience on Distributed Computing Infrastructure, Springer International Publishing Switzerland, 2014, 34-53.

[9] M. Radecki, T. Szymocha, T. Piontek, B. Bosak, M. Mamoski, P. Wolniewicz, K. Benedyczak, R. Kluszczyski, *Reservations for Compute Resources in Federated e-Infrastructure*, In eScience on Distributed Computing Infrastructure, Springer International Publishing Switzerland, 2014, 80-93.

[10] K. Kurowski, T. Piontek, P. Kopta, M. Mamoski, B. Bosak, *Parallel Large Scale Simulations in the PL-Grid Environment*, Computational Methods in Science and Technology, Special Issue 2010, 47-56.

[11] TORQUE Resource Manager, http://www.adaptivecomputing.com/products/open-source/torque/

[12] J.B. Layton, *Caos NSA and Perceus: All-in-one Cluster Software Stack*, Linux Magazine, 5 February 2009.

[13] J. Borgdorff, M. Mamonski, B. Bosak, K. Kurowski, M. Ben Belgacem, B. Chopard, D. Groen, P. V. Coveney, and A. G. Hoekstra, *Distributed Multiscale Computing with MUSCLE 2, the Multiscale Coupling Library and Environment*, Journal of Computational Science. 5 (2014) 719731

[14] S.C. Wang, S.C. Tseng, S.S. Wang, K.Q. Yan, *A Hybrid Load Balancing Policy underlying Cloud Computing Environment*, WSEAS Transactions on Computers (14), 580-587, 2015.

[15] M. Brzezniak, N. Meyer, R. Mikoajczak, G. Jankowski, M.Jankowski, *Popular Backup/Archival Service and its Application for the Archival of the Network Traffic in the PIONIER Academic Network*, CMST Special Issue (1), 109-118, 2010.

[16] S. Jaganathan, D. Veerappan, *CIADS: A Framework for Secured Storage of Patients Medical Data in Cloud*, WSEAS Transactions on Information Science and Applications (12), 22-35, 2015.

[17] S. Ikeda, *Atlas of flexible bronchofiberoscopy*, Tokyo: Igaku Shoin Ltd., p. 58-80, 1974

[18] H. Zhang, M. Tian, I. Carrio, A.C. Civelek, Y. Fujibayashi, *Molecular Image-Guided Theranostic and Personalized Medicine 2014*, BioMed Research International, 2015

[19] ANYS Workbench, http://www.ansys.com/Products/Workflow+Technology/ANSYS+Workbench+Platform