Grant agreement no. 675451

CompBioMed

Research and Innovation Action

H2020-EINFRA-2015-1

Topic: Centres of Excellence for Computing Applications

D6.3 Report on Workflow system provision

Work Package:	6	
Due date of deliverable:	Month 18	
Actual submission date:	23 March 2018	
Start date of project:	October 01, 2016	Duration: 36 months

Lead beneficiary for this deliverable: *USFD*

Contributors: *UCL, UvA, UNIGE, SARA*

Project co-funded by the European Commission within the H2020 Programme (2014-2020)		
Dissemination Level		
PU	Public	YES
CO	Confidential, only for members of the consortium (including the Commission Services)	
CI	Classified, as referred to in Commission Decision 2001/844/EC	

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

Disclaimer

The content of this deliverable does not reflect the official opinion of the European Union. Responsibility for the information and views expressed herein lies entirely with the author(s).

Table of Contents

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

1 Version Log

Version	Date	Released by	Nature of Change
V1.0	14/02/2018	Marco Viceconti	First Draft
V1.1	15/02/2018	Marco Viceconti	Additions form various co-authors
V1.2	16/02/2018	Marco Viceconti	Completed Taverna section
V1.3	24/02/2018	Marco Viceconti	Revision to address the comments of internal reviewers
V1.5	8/3/2018	Marco Viceconti	Final revision to include comments of coordinator and addition on BAC

2 Contributors

Name	Institution	Role
Alberto Marzo	USFD	Author
Marco Viceconti	USFD	Author
Shannon Li	USFD	Author
Andrew Narracott	USFD	Author
Alessandro Melis	USFD	Author
Alfons Hoekstra	UVA	Author
Bastien Chopard	UNIGE	Author
Marco Verdicchio	SARA	Author
Gianni de Fabritiis	UPF	Reviewer
Franck Chevalier	ACE	Reviewer
Emily Lumley	UCL	Reviewer
Peter Coveney	UCL	Reviewer

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

3 Definition and Acronyms

Acronyms	Definitions
μCT	Micro Computed Tomography
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BAC	Binding Affinity Calculator
BoneDVC	Bone Digital Volume Correlation
CBM	CompBioMed
CLT	Command Line Tool
CoE	Centre of Excellence
EnMD-TK	Ensemble Molecular Dynamics – Toolkit
ESRF	European Synchrotron Radiation Facility
EU	European Union
FTP	File Transfer Protocol
GUI	Graphical User Interface
HPC	High Performance Computing
HPDA	High Performance Data Analysis
HTC	High Throughput Computing
I/O	Input/Output
ISR	In stent restenosis
MMSF	Mulit-Scale Modelling & Simulation Framework
MPI	Message Passing Interface
MSMF	Multi-Scale Modelling Framework
OGF	Open Grid Forum
PBS	Portable Batch System
PM	Project Month
QCG	QosCosGrid Middleware
RDF	Research Data Facility
RP	RADICAL-Pilot
SAGA	Simple API for Grid Applications
ShIRT	Sheffield Image Registration Toolkit
SME	Small and Medium Enterprises
TIFF	Tagged Image File Format
VPH-HF	Virtual Physiological Human – Hypermodelling Framework
WMS	Workflow Management System
WP	Work Package

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

4 Executive Summary

The purpose of this deliverable is to describe the initial workflow system deployment conducted in T6.5 (Workflow Infrastructure Deployment) in the first 18 months of the project. The aim was to review the available technologies designed to facilitate execution and management of the workflows, with particular reference to the specific problem of orchestrating multiple models developed with different simulation packages and select a deployment strategy driven by the specific needs of *in silico* medicine, their potential end-users, and the way they may benefit from such solutions. At the end of this process we focus on five technologies: Taverna, which is found ideal to address the most basic workflow orchestration needs; MMSF, which on the contrary enable extremely complex orchestrations and support strongly coupled executions, but requires specialist support to be used; two custom solutions to specific problems (RADICAL to handle the tight scheduling problem in BAC, the large scale Monte Carlo cloud execution of OpenBF); and an implementation to include in the workflow the geographic transfer of large datasets from research facilities such as the Diamond Light Source. The portfolio that emerges is composite, as it is the field of *in silico* medicine. It also reflects the trend toward hybrid computing environments, where High Performance Computing (HPC) and High Performance Data Analytics (HPDA) architectures coexist.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

5 Introduction

Many computational medicine codes are monolithic in structure. But in many other cases, the required solution can only be provided as an orchestration of two or more distinct codes. There are a number of reasons for this, but the three most common are:

a) Orchestration where each code performs a specialised function (e.g. image segmentation, meshing, material mapping, model solution);
b) Orchestration where each code solves a separate model that describes the problem at a different space-time scale;
c) Orchestration where each code computes one part of a tightly coupled process.

A number of tools are available to orchestrate heterogeneous codes, many available in Open Source; for a review see [1]. However, most of these tools are being used in high-throughput computing (HTC), rather than high-performance computing (HPC) applications. Also, at the time of the proposal writing, some of the consortium partners had under development specific tools (VPH-HF, MSMF), which were seen as solutions potentially more suitable for HPC applications. Thus, task 6.5 Workflow Infrastructure Deployment was introduced in the work plan; by PM18 it was expected we would had adopted a clear strategy that would have been summarised in D6.3, the present document.

This document reports the results of activities conducted during months M13-M18 in WP6 task T6.5:

Task 6.5: Workflow Infrastructure Deployment (M1-M36) [Fast Track]

Leader: USFD (8 PM), Partners: UvA (5 PM), UNIGE (5), UPF (5), SARA (2)

This activity will focus on improving existing facilities for the execution and management of the workflows, with particular reference to the specific problem of orchestrating multiple models developed with different simulation packages. We will develop the software infrastructure needed to execute such workflows on HPC, data service and cloud infrastructures. The VPH Hypermodelling Framework (VPH-HF) and the Multi-Scale Modelling Framework (MSMF) will be the core software stacks, alone or in combination, to expose and execute workflows for end-users. We will use an agile software engineering model, where for each feature requested, an expert is identified, to provide the detailed specifications and check that the software developed fulfils them. As the selected workflow environments sit between the rest of the CoE infrastructure and the end users workflows, we will have need of two types of expertise, one for features related to their integration with the rest of the e-infrastructure, the other drawn from the workflow developers and their end users. The main activities will take place in Year 1 and Year 2; activities in Year 3 will be geared towards tool maintenance and sustainability.

6 End-users need analysis and adopted strategy

Once CompBioMed started, we systematically reviewed with all core and associate partners the actual needs for workflow execution management software in an HPC environment. This review showed a much less complex scenario than we expected at the time of the proposal writing. Most solutions under development in CompBioMed were comprised of monolithic code, and where two or more codes were involved the composition and orchestration were handled by ad hoc scripts; in many cases, the developers felt that the need to have complete control of the code and its execution was essential to achieve the desired scalability.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

There were three exceptions:

I) Some solutions had been developed as orchestrations of a large number of separate codes, of which only one or two had serious requirements for HPC. This was seen mostly as a deployment problem. These workflows were very simple topologically and most could be represented with Direct Acyclic Graphs (serial execution). In these cases, the users wanted the workflow management system (WMS) and all the non-HPC codes to run on one HPC node, and to orchestrate the execution of the HPC codes through the available queuing system;

II) In a few cases, it was suggested that complex strongly coupled executions had to be handled efficiently. This was seen mostly as a research problem, and no clear specifications for a single solution emerged.

III) On the other hand, for some solutions we were faced with the issue of handling the data flow across disparate geographical locations. In various computational medicine applications, the data required to initialise the model computation are generated in a dedicated facility geographically separated from the HPC centre. When the input datasets are very large, moving these data from the source facility to the HPC system becomes a relevant logistic issue; thus, we decided to extend the scope of this deliverable to explore this scenario.

7 HPC execution of Direct Acyclic Graph workflows with Taverna

With respect to the requirement I) in section 6, after a systematic review of the end-users needs, we decided to adopt a WMS that was simple, robust, EU-developed, and available in Open Source. Amongst those, Taverna[1] emerged as the most popular among our consortium members.

In Taverna, workflows are represented in terms of direct acyclic graphs. Each node contains a part of the workflow in terms of calls to external software. These building blocks can be run independently provided that all the required inputs are available at runtime. The graph edges represent message passing operations between the workflow building blocks.

Taverna is provided in two versions aimed at prototyping and production phases, respectively. Taverna Workbench is a desktop client including a GUI to ease the creation and editing of workflows through a drag and drop interface. The Taverna Command Line Tool (CLT) is a stand-alone Java application that allows Taverna workflows to be run without loading the GUI. The CLT requires only the Java Runtime Environment (v7+) to be installed in the computing system. Therefore, the CLT is the version of choice for production in HPC systems.

The Taverna CLT v2.5 was successfully deployed in ARCHER, the EPCC tier-1 HPC system in Edinburgh, and in ShARC, the USFD tier-3 HPC system in Sheffield. The BoneDVC workflow, developed in Taverna Workbench, was tested and benchmarked, both on ARCHER and ShARC; the workflow description and the unpublished results from this study are reported in Section 4.

While this exploratory work showed that it is possible to deploy a Taverna workflow in an HPC environment without any particular difficulty, and without any severe performance penalty, it also highlighted that the opportunity to use this approach is limited. If the orchestration is one-

[1] https://taverna.incubator.apache.org

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

off, with no expectation of reuse, ad hoc scripts are a more popular solution. Also, Taverna design choices make it quite difficult to handle strongly coupled models or heavily multiscale models, which is why we supported also the alternative pathway described in section 8 below. Still, if a group is developing multiple workflows and there is a high rate of reuse of different sub-modules, and/or there is a lot of redesign and testing of different orchestrations of the same sub-models, Taverna is an effective tool, and in those cases, we recommend it for all CompBioMed users.

While this has not been tested experimentally in this project, the extension to use Taverna to orchestrate a workflow executing in the Cloud is not particularly challenging. The Taverna project web site[2] lists seven large research projects where this was addressed successfully. The Giza team that developed the Galaxy-Taverna integration (called Tavaxy) also offer tools for cloud execution [2]. Robert Haines published a slide set[3] where he explains how to optimise the execution of complex Taverna workflows in the cloud.

8 HPC Execution of strongly-coupled workflows: MMSF

With respect to the requirement II) in section 6, it was decided that the best approach was to drop the VPH-HF framework, which was found to be unnecessarily complex for the use cases of this project, and to focus instead on the Multiscale Modelling and Simulation Framework (MMSF), [3], which in the proposal was referred to as Multi-Scale Modelling Framework (MSMF). While Taverna is the default offer to WMS problems, groups that have problems that could not be effectively solved with Taverna are referred to partners UvA and UNIGE where, case by case, specific solutions based on MMSF can be devised.

Partners UvA, UCL, and UNIGE continue to investigate ways to improve the MMSF by designing new software functionalities and merging them in a single middleware instance. This is performed in collaboration with a number of European and national projects (e.g. VECMA, ComPat, and eMusc). Two recent papers provide an overview of their main ideas about HPC multiscale computing and describe a tool to help users predict the performance of distributed multiscale applications [4, 5]. Moreover, we have proposed a family of algorithms for Uncertainty Quantification of multiscale models [6] and are currently testing them on the in-stent restenosis (ISR) application from partner UvA and associated partner ITMO.

The MMSF and associated MUSCLE software have been successfully ported to associated partner ITMO, who installed it at their local computational infrastructure in Saint Petersburg and on the Lomonosov 2 supercomputer in Moscow. This allowed them, in collaboration with partner UvA, to perform a substantial number of simulations of a full blown three-dimensional version of the ISR model, demonstrating its validity against porcine data [7]. We are currently running more ISR simulations to compare the results against extremely detailed histological data that we obtained from partner USFD (see fig. 1 for partial -unpublished- results).

[2] https://taverna.incubator.apache.org/introduction/taverna-in-use/#cloud
[3] https://www.slideshare.net/mygrid/taverna-workflows-in-the-cloud

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

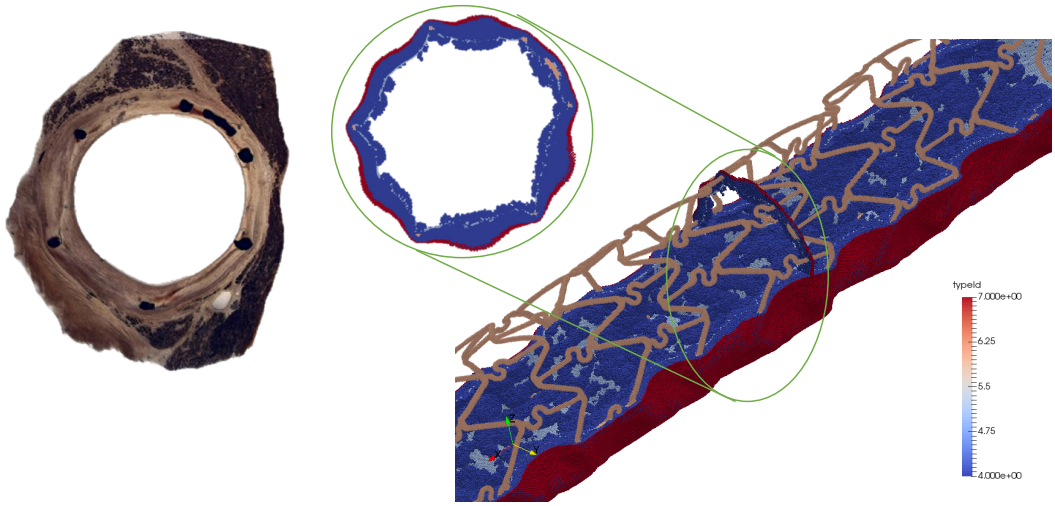

Figure 1. Results of an ISR run using MMSF/MUSCLE, showing histology of neointima 14 days post stenting (right) compared to in-vivo histology (left). Note the much smoother *in vivo* histology as compared to the simulated histology. We attribute this to the fact that the ISR model does not yet contain extracellular matrix, which is currently being added to the model.

In collaboration with the ComPat project we have ported the QCG middleware to partner SURFsara. The MMSF can interface with QCG, allowing execution of multiscale models developed in MUSCLE in High Performance - or Distributed Multiscale Computing mode. We are currently working with ComPat to explore options to use their Multiscale Computing Patterns for a few selected CompBioMed applications.

9 HPC execution of drug binding affinity calculations

A third class of workflow seeks to address the limitations imposed by HPC queuing systems in order to create efficient large-scale hybrid applications. An example of such an application is the Binding Affinity Calculator (BAC), a decision support tool which uses molecular level computer simulation to reliably predict the binding affinities (free energies) of molecules with target proteins, and therefore to identify those most likely to bind to the protein. BAC has been built to integrate and automate the multi-step process of model building, simulation, and data analysis for molecular level drug-receptor interactions. It constitutes a sophisticated computational pipeline built from selected software tools and services, and which relies on access to a range of computational resources.

BAC depends on the ability to perform hundreds of separate parallel simulations on a high-performance computing platform, each of which can require 50-200 cores depending on the system. The BAC workflow automates much of the complexity of running and marshalling these simulations, and collecting and analysing data. This requires a workflow management tool that integrates closely with the queuing system on a HPC resource, in order to efficiently allocate replicas between available compute nodes, and also manage the execution and data staging between different steps of the simulation protocol. For this purpose, BAC uses RADICAL-Cybertools, a suite of abstractions-based and standards-driven tools that provide a common, consistent, and scalable approach to high-performance and distributed computing. RADICAL-

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

Cybertools consists of three fundamental components: (i) SAGA – an API that is an OGF community standard for application-level jobs and data movement. (ii) RADICAL-Pilot (aka BigJob) which is a tool that provides the ability to aggregate large number of tasks into a single-container job, enables the flexible assignment of tasks to different resources (via late-binding) and supports application-level scheduling and adaptive execution strategies to improve the through-put and volume of tasks. (iii) Ensemble MD Toolkit (EnMD-TK) which builds upon RADICAL-Pilot (RP) as the execution layer to support different patterns of ensemble-based computing, such as replica-exchange, pipelines, and simulation-analysis loops. Both EnMD-TK and RP use SAGA as the underlying access layer to heterogeneous infrastructure, thus providing uniformity and consistent access to the infrastructure layer.

The protocol employed by BAC takes a generic molecular model and uses patient specific genomic data to personalise the model. This model is then used to run multiple replicas of the simulation from which to generate statistically significant results. The protocol is automated through a software system called the Binding Affinity Calculator (BAC), which automates the creation of models, management of data and execution of simulation. Typically, these simulations require 1000s of cores on a high-performance computing machine in order to be calculated in a clinically relevant timeframe. The main applications are twofold: a) as a tool for the pharmaceutical industry to assist with drug design and experimentation with novel molecules which could become potential medicines, and b) in the treatment of patients with specific conditions, as a treatment planning aid for clinicians. BAC will make currently available treatments more effectively targeted by allowing clinicians to only select those treatments which will have an effect on the individual, thus eliminating potential side effects from ineffective treatments. BAC also provides a means to analyse genomic data and include it in the treatment of an individual, which will help eliminate misdiagnosis.

10 Cloud execution of embarrassingly parallel parameter sweep studies

In the case of large scale parameter sweep studies, several thousands of medium to small size serial simulations are run with different input parameters. There is no communication occurring between the single simulations and each job is independent from the others, therefore they can be run concurrently using all the cores available on the machine where they are executed. The Cloud HPC architecture is particularly well suited for this task, as the size and computing power of the system can be adapted to scale with the increase of the parameters (input) space.

Partners USFD and SURFsara investigated the use of Cloud computing infrastructures for a cardiovascular related study. In particular, a patient-generic 1D model of the cardiovascular system (openBF, USFD) was used to generate several models representative of a range of individuals. This virtual population was built through an embarrassingly parallel parameter sweep approach. The deployment of the openBF code, has been done using Singularity containers [8]. This technology allows users to create and run reproducible environments (software containers) that can easily be copied and executed on other platforms, providing a secure and reproducible way to distribute software and computing environments. For a detail description of Linux Containers (such as Singularity and Dockers) we refer the reader to Section 2 of Priedhorsky et. al. [8]. In this work, we have used Singularity not only to have a self-containing instance of openBF which was easily ported and run on different systems, but also to orchestrate the execution of the several thousand of simulations required to build the above mentioned virtual population.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

Physiological ranges of the model parameters were identified from the literature including length, lumen radius, wall Young's modulus, and the Windkessel model parameters used at outlets. A Latin Hypercube algorithm was used to ensure a homogeneous coverage of the input space that led to 5000 simulations with different combinations of the input parameters. Each simulation was run at three mesh density values to ensure mesh independence, leading to a total of 15,000 simulations. The single simulation took between 20 and 30 minutes of execution time for each set of input parameters. The single-core simulations, which require more than 200 days if executed sequentially, were run over a multi-core Virtual Machine (HPC cloud, SURFsara, NL), a tier-1 (Cartesius, SURFsara, NL) and a tier-3 HPC system (ShARC, Sheffield, UK), allowing us to reduce the total compute time linearly with the number of cores available.

The convergence of mean and standard deviation across the population for aortic and brachial artery systolic and diastolic pressures was monitored to check Monte Carlo analysis convergence (Figure 2).

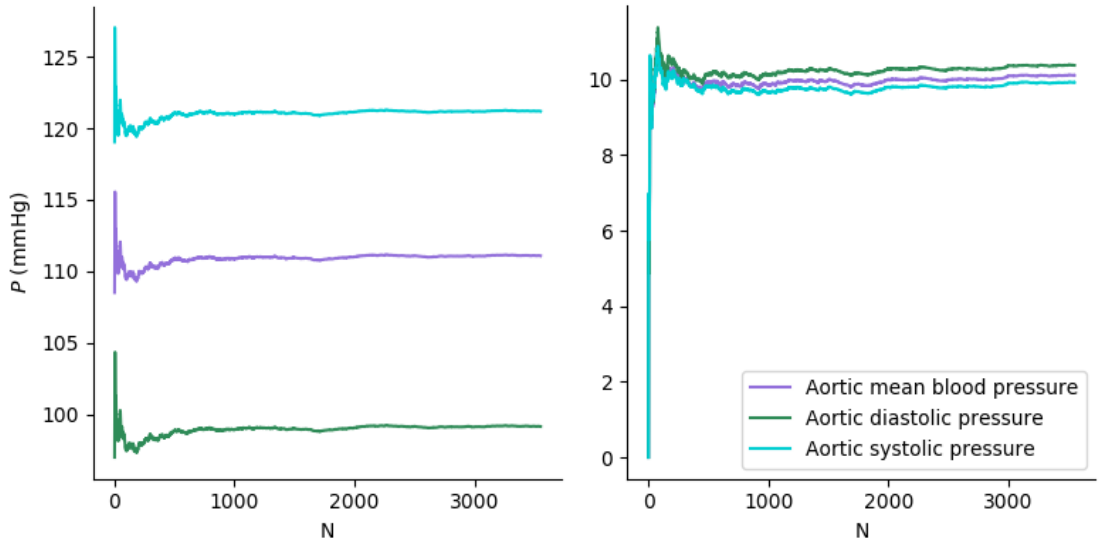

Figure 2: Monitor of the aortic pressure values convergence across the virtual population. The mean, systolic, and diastolic pressure are reported in terms of population wise mean and standard deviation values.

An analysis of the resulting pulse pressures was performed to exclude non-physiological waveforms and the respective models. Final results (Figure 3) were validated through a comparison of diastolic and systolic pressure distributions at specific locations with data published in the literature for similar population studies.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

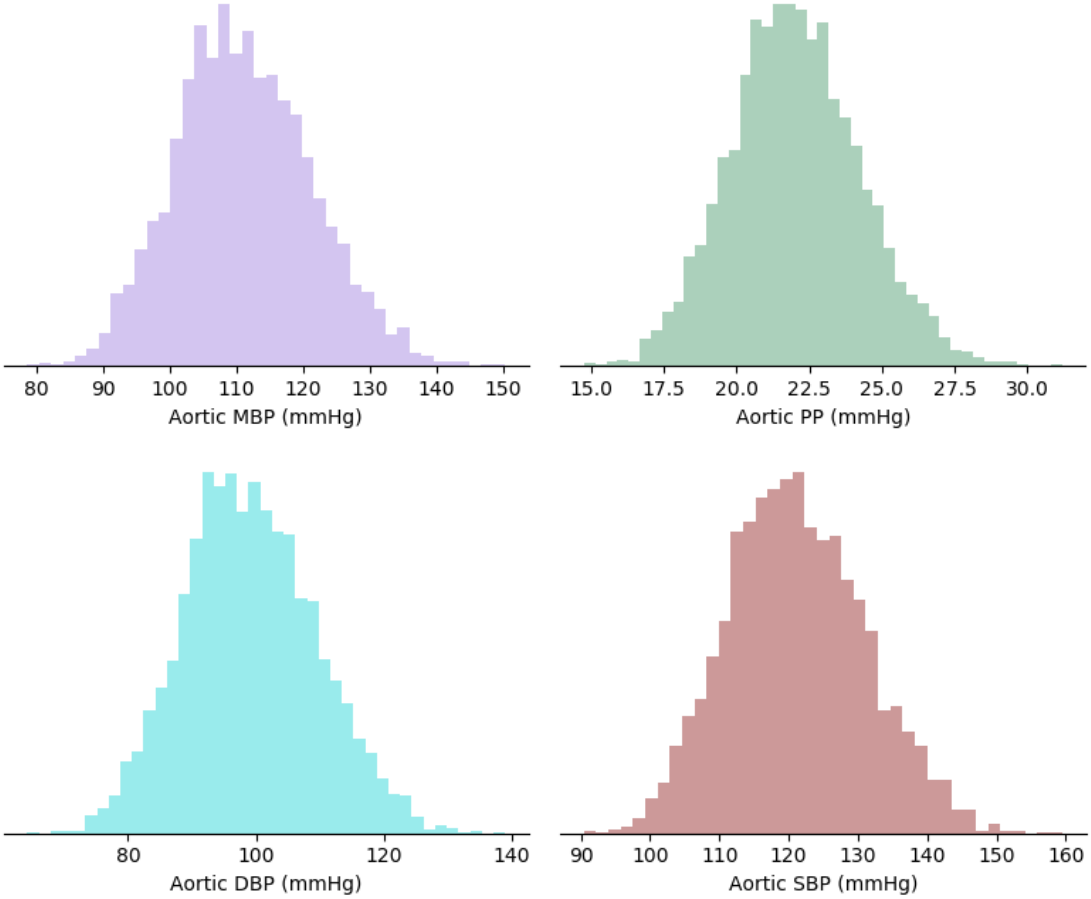

Figure 3: Resulting distributions of aortic pressure values within the virtual population (clockwise from top left: mean blood pressure, pulse pressure, systolic blood pressure, and diastolic blood pressure).

11 Geographical data flow integration: The Diamond Light Source pilot

11.1 Overview

With respect to the requirement III) in section 6, it was decided to conduct a pilot study to assess the complexity of including geographic data transfer in the workflow. One of the solutions developed by CompBioMed includes a number of tools (BoneDVC, MicroFE, etc.) for biomechanics and mechanobiology research on bone tissue. These models are initialised by very large X-ray micro- and nano-tomography datasets collected at synchrotron light source facilities such as the European Synchrotron Radiation Facility (ESRF), or the Diamond Light Source (UK's national synchrotron science facility). Each of these datasets is typically some hundred Gigabytes, so the transfer of these data to the HPC facility where they are processed should ideally be automated and embedded in the workflow.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

The Data Transfer and BoneDVC workflow is an extension of the existing boneDVC analysis workflow developed at the University of Sheffield, which demonstrates the ability to perform analysis of very large (of the order of hundreds of gigabytes) 3D image data transferred on-demand from an external experimental facility to a Tier-1 HPC facility. The workflow must address three primary tasks which are:

1. Efficient transfer of large datasets from the experimental facility (in this case the Diamond Light Source), to the HPC Facility (in this case Archer, using the co-located Research Data Facility (RDF) as the primary file storage system).
2. Conversion and pre-processing of the data to enable use with the existing workflow.
3. Deployment of the existing workflow to the new HPC environment.

A schematic overview of the workflow is given in Figure 4.

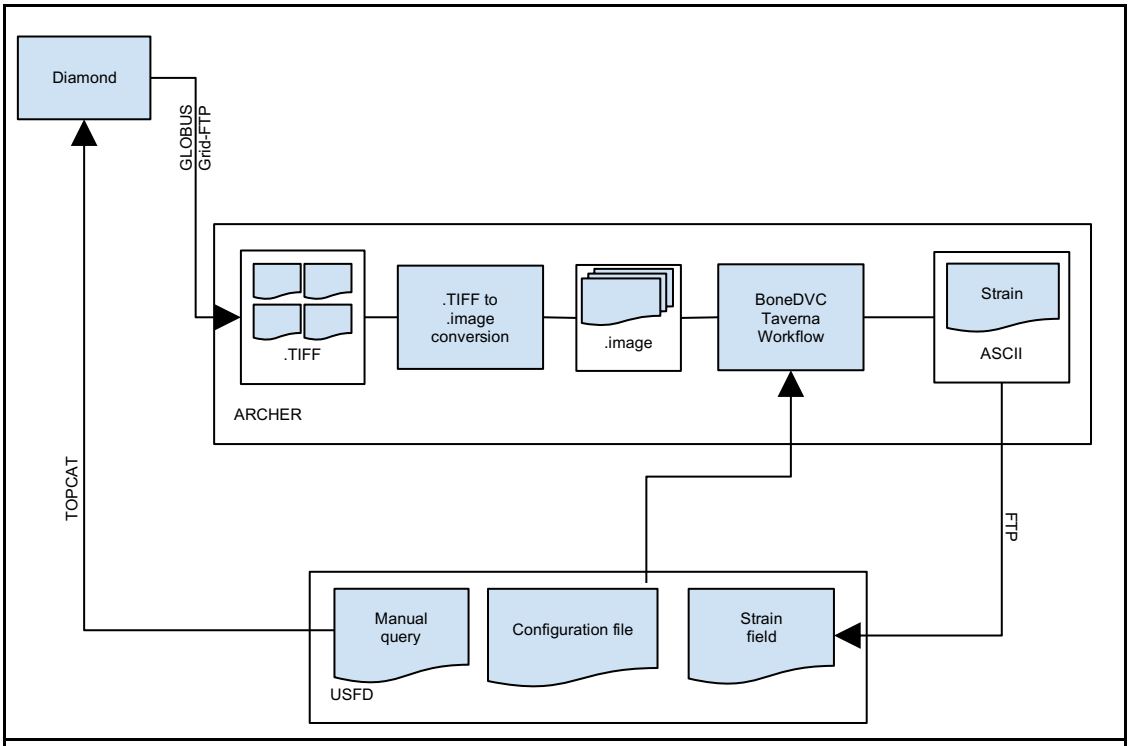

Figure 4. Workflow scheme depicting data transfers and data handling main steps within ARCHER.

11.2 The Workflow stages

This section provides a detailed description of the workflow, divided into its constituent functional blocks.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

11.2.1 Data Staging from Diamond

Workflow data stored at the Diamond facility may be located either on short term online storage or in long term tape archive storage. In the case of data in archival storage it must be first 'staged' to online storage. 'Staging' refers to the process of reading the data from tape and recovering it to an internet connected server such that it may be transferred elsewhere.

For archived Diamond data this is achieved using the 'TopCat'[4] web interface. In order to stage data the user must first log in, and will be presented with a list of their available data. Figure 5 shows a screenshot of this interface. Individual files as well as directories of files may be selected and added to the 'download cart'. Once all desired files have been added to the download cart, staging of the data can be requested by opening the cart and selecting download of data.

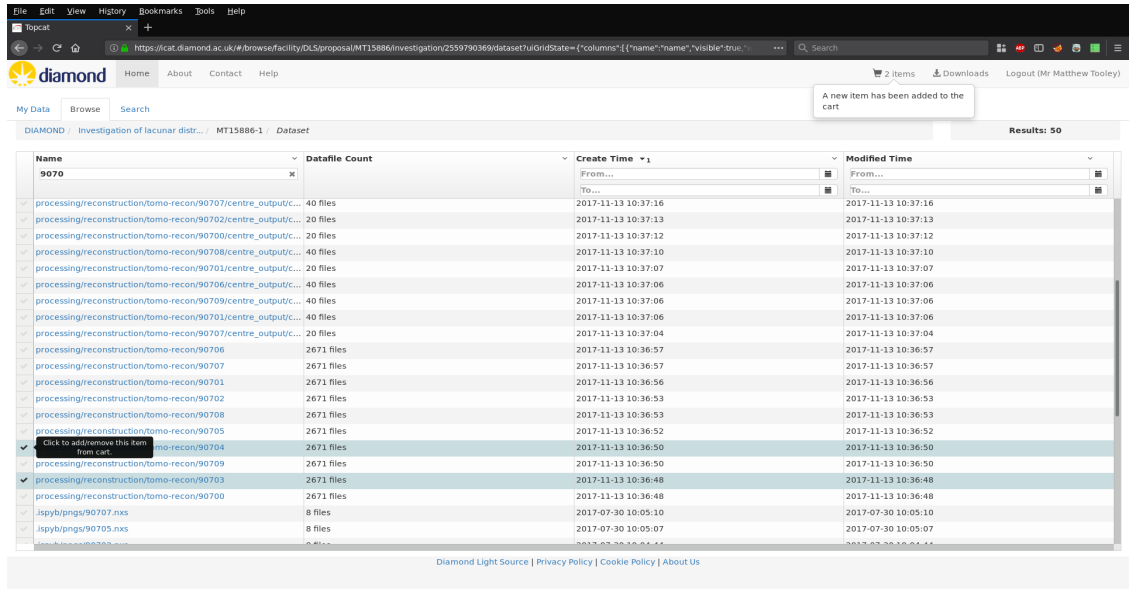

Figure 5. The TopCat web interface, showing the addition of desired directories to the download cart.

Two options are available for downloading the data as shown in the screenshot in Figure 6. It can be made available for download by the http protocol using a standard web browser, or it can be made available for parallel ftp transfer using the Globus[5] system. For transfer to the Research Data Facility[6] attached to the Archer[7] HPC, only the parallel ftp method is viable as the http download method is both slower and requires the use of a web browser on HPC, which is an impractical use of Archer resources.

[4] "Topcat - Diamond Light Source." https://topcat.diamond.ac.uk/. Accessed 1 Feb. 2018.

[5] "Research data management simplified. | globus." 25 Apr. 2017, https://www.globus.org/. Accessed 1 Feb. 2018.

[6] "ARCHER » UK Research Data Facility (UK-RDF) Guide." http://www.archer.ac.uk/documentation/rdf-guide/. Accessed 1 Feb. 2018.

[7] "archer.ac.uk." http://www.archer.ac.uk/. Accessed 1 Feb. 2018.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

Once staging is requested the data is copied from tape to the data transfer node at the Diamond Facility. This process can take several hours to complete, and the user can opt to receive an email when the data is ready for transfer.

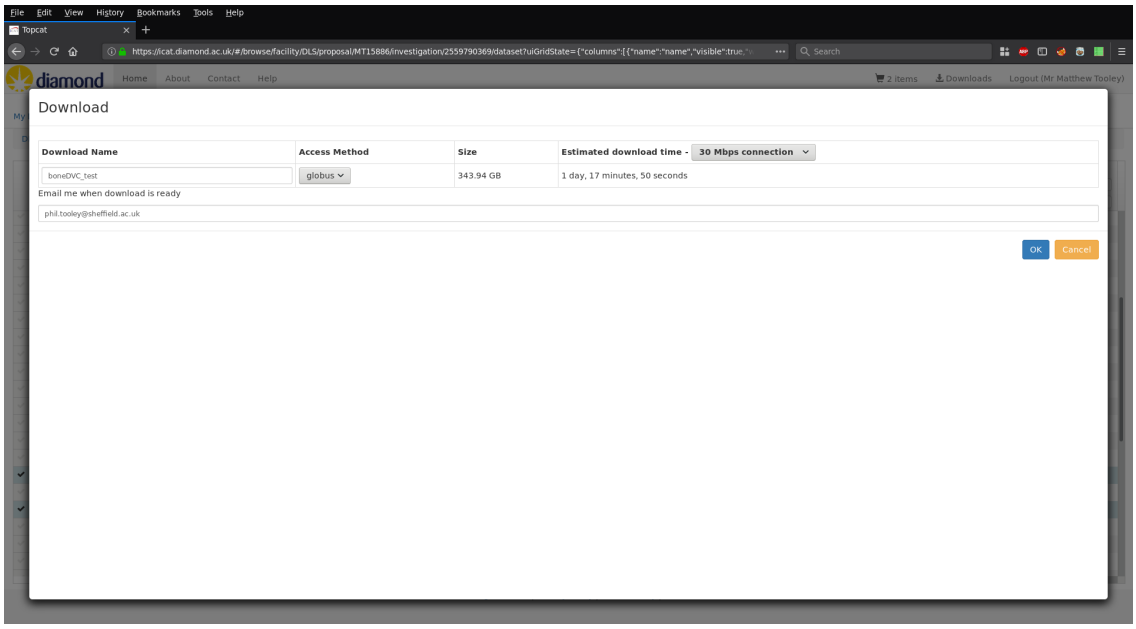

Figure 6. The TopCat download options interface.

11.2.2 Data transfer with Globus

Once the user has received notification that the staging process is complete, data transfer can be performed using the Globus service. This transfer can be achieved in one of two ways: either manually using the Globus web interface, or in an automated fashion using the API interface to Globus.

The manual approach using the web interface for transfer is well documented in the Globus Manual[8]. After logging into the Globus online system, the user may authenticate with the individual endpoints at the Diamond and RDF using the login credentials provided by these facilities. The filesystems may be explored and transfer of files between the two endpoints requested.

[8] "How To Log In and Transfer Files with Globus - Globus Docs." https://docs.globus.org/how-to/get-started. Accessed 1 Feb. 2018.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

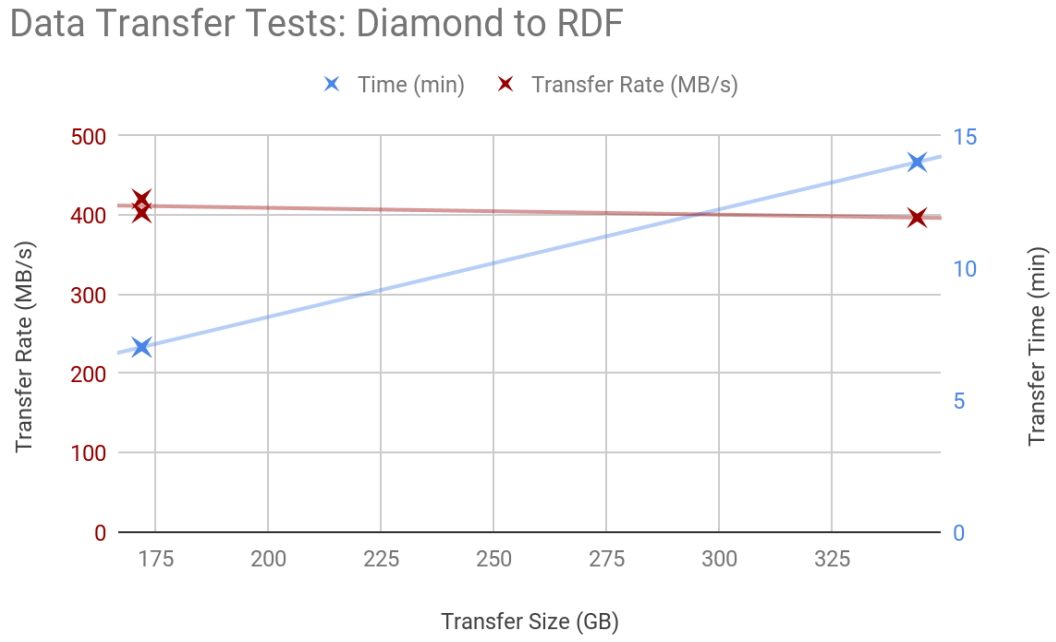

Figure 7. Data transfer rates for multiple attempted transfers from Diamond to the RDF using Globus.

Alternatively, the transfer may be automated using a python scripted interface to the Globus System. This makes use of the Globus-SDK[9], and additional code has been written to provide the end user with an easily configurable method to perform automated transfers. This scripted approach allows Globus file transfers to be easily included in any automated workflow, as well as allowing users to queue multiple transfers more rapidly than can be achieved using the graphical web interface approach. The use of such an interface is an important building block in the production of a completely automated workflow solution.

Multiple transfers have been performed between the Diamond and RDF endpoints to determine the scalability of this transfer method to large datasets. Figure 7 shows the results of this for transfers of datasets of hundreds of gigabytes. Data transfer rates were found to reliably be in the region of 400MB/s for transfers with parallelism=8, meaning that each parallel thread is capable of approximately 50MB/s transfer. This means that it takes approximately 40 minutes to transfer 1 Terabyte of data.

11.3 Image Pre-processing

Tomograms from the Diamond facility are initially presented in the form of a large number of TIFF images, each representing a single slice through the object. The further processing of the data requires that the images be represented as a single 3D image object, therefore pre-

[9] "Globus SDK for Python." http://globus-sdk-python.readthedocs.io/en/stable/. Accessed 1 Feb. 2018.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

processing of the images into this form is required. In order to achieve this, a custom image-pre-processor has been written in the Python[10] language with the Numpy[11] and Pillow[12] libraries to allow efficient processing of these into the 3D binary data format required by the ShIRT program in the next stage of the workflow.

Individual images from Diamond are approximately 4000 x 4000 x 2000 voxels in size, with double precision greyscale voxel format. This translates to a RAM usage of approximately 130GB for the image array which is too large to be processed on a single node of the Archer HPC. Whilst a future deep-track activity is to perform MPI-enabled parallel processing of such images, the current approach to allow the images to be analysed is to perform subsampling of the image, creating a lower resolution image that can be processed with existing software. In order to perform this sub-sampling, the entire image must still be loaded and analysed, therefore a memory-efficient block-processing algorithm was implemented to allow this.

The block processing algorithm works by considering small subsets or "blocks" of the output image, which correspond to blocks in the original image which are small enough to be loaded into memory. This smaller block of the original image is then rescaled to a smaller size and stored, before the memory is freed to allow another block to be loaded. Considering all blocks in turn in this fashion, only ever loading subsets of the complete image into memory, the algorithm is able to process the image without running out of memory.

This pre-processing tool is currently single-threaded and has not been aggressively optimised. The execution time with respect to the desired output image size is shown in Figure 8. Execution time scales linearly with the size of the output image. This is an expected result since the image pre-processing is an IO-bound operation, and so the runtime scales with the amount of data that must be loaded, which itself scales linearly with the output image size.

[10] "Python.org." https://www.python.org/. Accessed 1 Feb. 2018.

[11] "NumPy — NumPy." http://www.numpy.org/. Accessed 1 Feb. 2018.

[12] "Pillow — Pillow (PIL Fork) 5.1.0.dev0 documentation." https://pillow.readthedocs.io/. Accessed 1 Feb. 2018.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

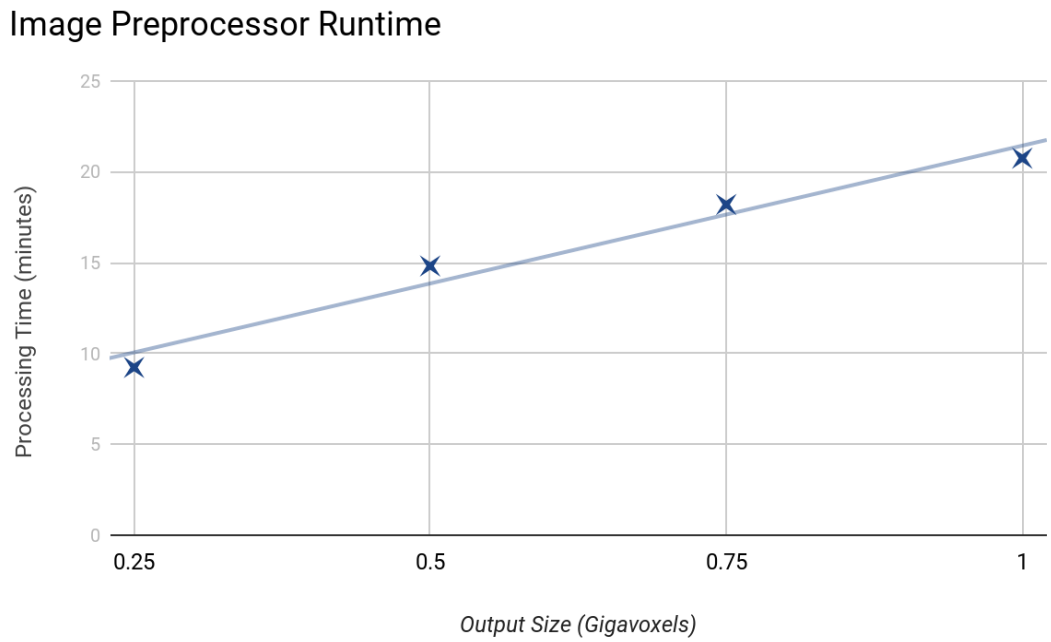

Figure 8. Runtime scaling for the image resizing pre-processor step. The output image retains the same aspect ratio as the input image. A ~1 gigavoxel image has dimensions 1260 x 1260 x 630 voxels, with the scaling factor of the side length being the cube root of the scaling factor of the total voxel number.

11.4 The BoneDVC Taverna workflow

The BoneDVC workflow analyses two stacks of µCT images of the same bone tissue specimen. The former is fixed (reference configuration) and the latter is imaged under compression (displaced configuration). The linear displacement field mapping the second volume to the first is calculated by means of the digital volume correlation (DVC) method. The strain field in the compressed specimen is obtained by differentiation of the computed displacement field.

The BoneDVC was developed as a composition of commercial, licensed, and open-source software. The workflow input consists in the pair of µCT images in the custom binary ".image" format required by ShIRT and a configuration file containing the numerical parameters needed to run the image registration code. The BoneDVC workflow, illustrated in figure 9. was staged in a sequence of steps:

1. The image registration software, ShIRT[13], is executed within a Taverna[14] workflow inside a Python wrapper. The Python wrapper collects the pair of images and runs the PBS (Portable Batch System) scheduling script.

[13] "Automatic segmentation of medical images using image registration ...." https://www.ncbi.nlm.nih.gov/pubmed/15804853. Accessed 1 Feb. 2018.

[14] "Apache Taverna - Apache Taverna (incubating)." https://taverna.incubator.apache.org/. Accessed 1 Feb. 2018.z

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

2. The displacement field from the registration step is used as input for the calculation of the strain field. This step is done by using a MATLAB (MATLAB R2016a, TheMathWorks, Inc.) script previously compiled and ran via the Matlab runtime environment.
3. The workflow output is the computed strain field in ASCII format.

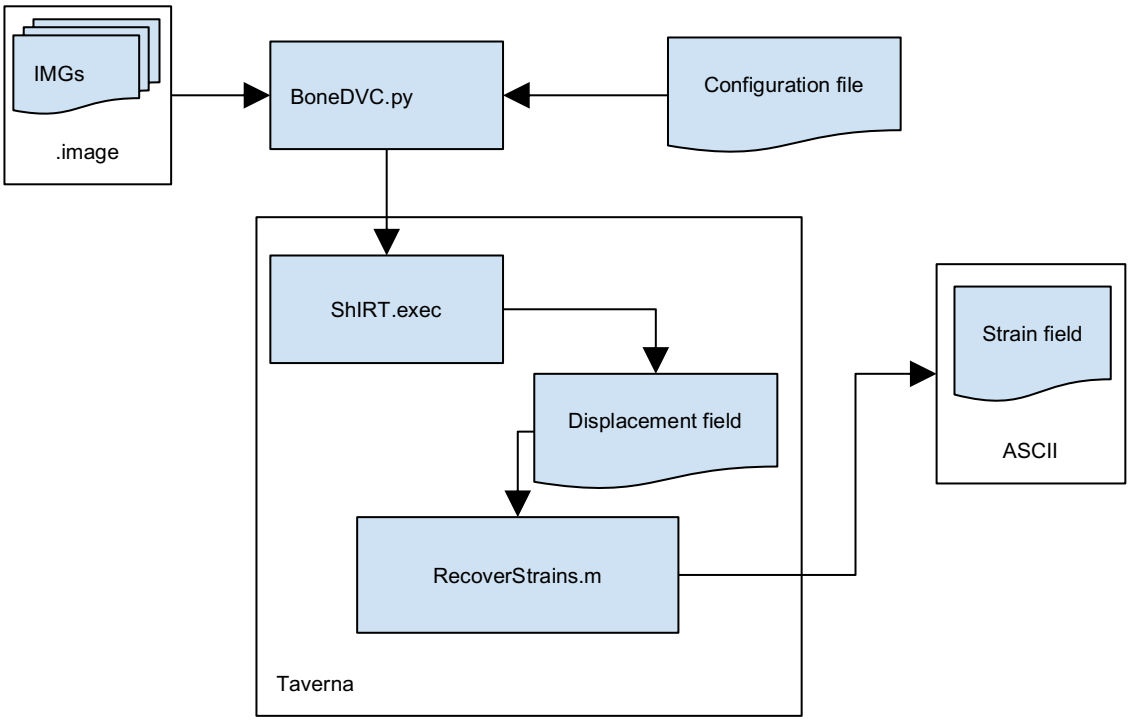

Figure 9. BoneDVC workflow. A Python script collects the pipeline input (configuration file and image data) and initialises the Taverna workflow which orchestrates the registration and differentiation steps. The final output comprises the computed strain field.

11.4.1 Tier-1 vs Tier-3 performance comparison

The BoneDVC workflow was tested on both tier-1 (ARCHER, EPCC) and tier-3 (ShARC, USFD) infrastructures in serial mode, i.e., only one node was employed for the workflow execution. This is because ShIRT is currently single-threaded, and its execution would not benefit from the allocation of multiple nodes. The tier-1 serial node consisted in a 2.0GHz 10-core Intel Xeon E7-4850 (Westmere) processor with 1TB physical memory available; the tier-3 node had a 2.4GHz 8-core Intel Xeon E5-2630 (Haswell) series processor and 64GB RAM. The performance was assessed in terms of wall-clock time and maximum virtual memory allocation (Fig. 9) on four datasets of increasing size (10, 100, 1000MB, and 4000MB in size for each 3D image). The workflow was run three times for each dataset and the results were reported in terms of mean and standard deviation values. Note that although ShIRT is not parallelised, the strain field calculation is performed in parallel across the cores of a single node. Therefore, there is some benefit to running on an entire node.

The computational time (figure 10a) increases with the input image size for both machines, and the workflow execution is in all the cases faster on the tier-3. The tier-1 execution time is 3.2, 2.3, and 2.2 times the tier-3 one for the three datasets, respectively. The memory consumption

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

is reported with respect to the single image size for each dataset (figure 10b). In the case of the smallest dataset, the tier-1 allocates three times the maximum virtual memory used by the tier-3. The performance is similar for the 1GB image dataset as both machines allocate about 80GB of virtual memory. In the case of the largest image dataset, the tier-3 peak requirement is of 90GB and it is almost twice the tier-1 memory allocation. Due to the memory requirement, the 4GB image could not be tested on the tier-3 system.

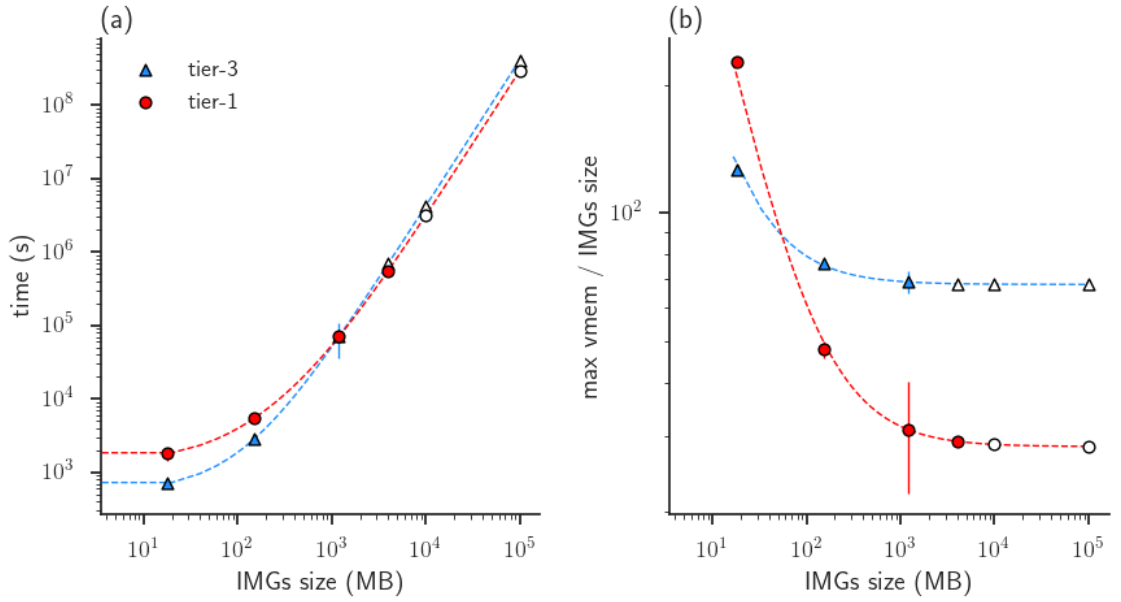

Figure 10. (a) Wall-clock time required for the execution of BoneDVC workflow for the three datasets of increasing size. (b) Normalised virtual memory requirement for each registration task. The memory requirement is scaled and reported with respect to the dataset size. Execution time and memory consumption for untested large images are linearly extrapolated from benchmark results (blank markers).

11.4.2 Profiling

A small pair of µCT images (30MB each) was used as a test case and profiling was performed by collecting during runtime the timestamp at the beginning and ending of each main block in the BoneDVC workflow (figure 11). As expected, the majority of the computational time was employed by the image registration process (ShIRT). The time required to compute the strain field (ANSYS-post) was less than 1% of the total running time. Interestingly, the compilation of the MATLAB scripts calling ShIRT improved the system I/O overhead and the total running time decreased by 9.3%. In both versions, the Taverna overhead was negligible.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

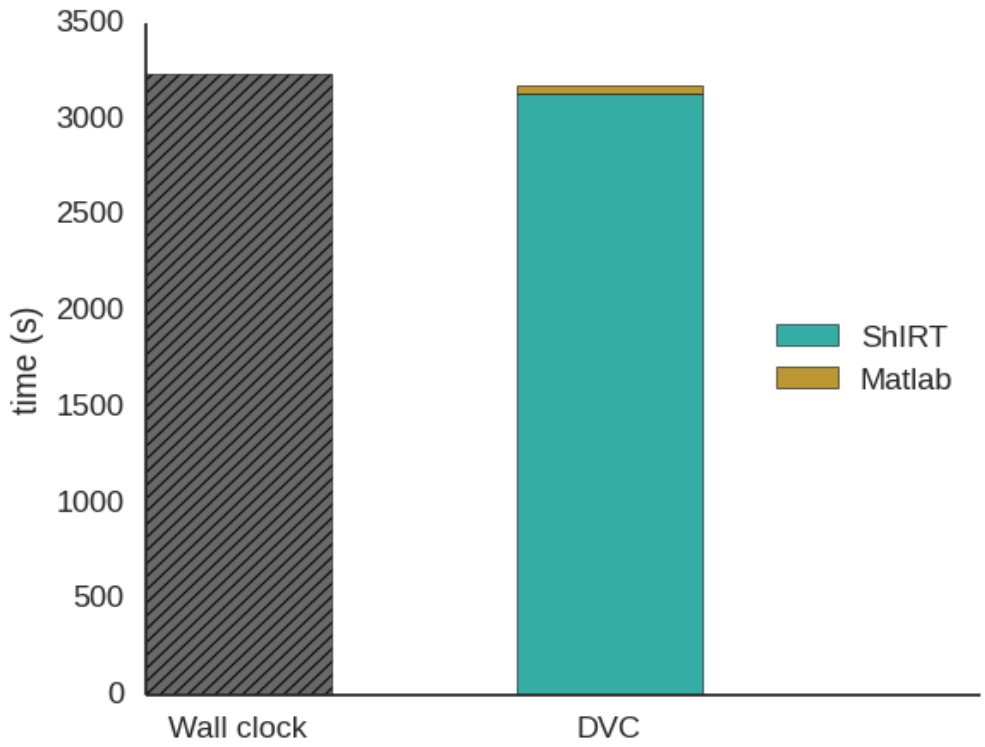

Figure 11. Time profiling for the BoneDVC workflow execution. The first bar shows the wall clock time including system I/O. The second bar shows separately the execution time of ShIRT and Matlab scripts.

11.5 Further development potential/Outlook

The profiling of BoneDVC shows a clear bottleneck with the current ShIRT library. This is a custom-code written in C in early 2000 which was never intended to scale to problems the size of the current Diamond Tomograms.

The original source code was analysed with a view to potentially updating and parallelising the code, however several issues came to light during this process:

1. The current code is subject to licensing restrictions due to previous commercialisation. This makes reuse of the existing codebase incompatible with the aims to make the code open-source and accessible.
2. The current implementation of the core solver is such that parallelisation would require a complete rewrite.

Following this analysis, it was concluded that the only viable option to improve BoneDVC on the Tier-1 system is to rewrite the ShIRT library into a modern, well-documented and parallel code.

A further potential development opportunity, should there be larger scale appetite to make use of this workflow is to work closely with the facilities which host the experiments and data archives to improve the current abilities to access the data. For example, by using available APIs

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

to create a single interface which allows users to select data from different facilities and orchestrate the transfer and analysis of the data from a single interface.

11.6 Extension to other Synchrotron facilities

The experience we have had with the transfer of large data objects from the Diamond facility to the Archer HPC system will now be shared with all the other partners in the CompBioMed consortium. During 2018 the team of Jazmin Aguado-Sierra at partner BSC, in collaboration with partner UPF, will deploy a similar data transfer service to transfer high-resolution imaging data on cardiac tissue from the European Synchrotron Radiation Facility based in Grenoble (FR) to the BSC Mare Nostrum HPC system in Barcelona (ES). We expect similar experiences to follow in other centres.

12 Conclusions

As originally expected the work done that this deliverable summarises gave us the opportunity to engage with the end-users within and beyond the project network, gather their requirements and use these to adapt existing solutions in the ways that the Computational Medicine community needs. The portfolio that emerges is composite, as it is the field of *in silico* medicine. It also reflects the trend toward hybrid computing environments, where High Performance Computing (HPC) and High Performance Data Analytics (HPDA) architectures coexist.

Some complex technologies that on paper were very promising, such as VPH-HF, were dropped because - from an end users' point of view - the need for such complication is rarely required. Judging the right level of complexity that is truly necessary for a specific problem is essential to ensure not only efficiency of workflow operation, but also uptake by the broader community of end-users. The CompBioMed Centre of Excellence emerges from this first block of activities with a robust portfolio of workflow execution technologies, all deployed on the appropriate architectures, which we believe can collectively address the vast majority of needs that might emerge from the Computational Medicine community.

13 Bibliography / References

[1] Viceconti M, Bnà S, Tartarini D, Sfakianakis S, Grogan J, Walkeer D, Gamble S, Testi D. VPH-HF: a software framework for the execution of complex subject-specific physiology modelling workflows. Journal of Computational Science, (2018). *In press*.

[2] Abouelhoda M, Issa SA, Ghanem M. Tavaxy: integrating Taverna and Galaxy workflows with cloud computing support. BMC Bioinformatics. 2012 May 4;13:77. doi: 10.1186/1471-2105-13-77.

[3] B. Chopard, J. Borgdorff, A.G. Hoekstra, A framework for multi-scale modelling, Philos. Trans. R. Soc. A. 372 (2014) 20130378. doi:10.1098/rsta.2013.0378.

[4] S. Alowayyed, D. Groen, P. V. Coveney, A.G. Hoekstra, Multiscale Computing in the Exascale Era, J. Comput. Sci. 22 (2017) 15–25. doi:10.1016/j.jocs.2017.07.004.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

[5] Bastien Chopard, Jean-Luc Falcone, Pierre Kunzli, Lourens Veen, and Alfons Hoekstra. Multiscale Modeling: recent progress and open questions. Multiscale and Multidisciplinary Modeling, Experiments and Design (MMED). Vol 1, pp.1-12, 2018. doi:10.1007/s41939-017-0006-4.

[6] Anna Nikishova, Alfons Hoekstra, Semi-intrusive Uncertainty Quantification for Multiscale models, Journal of Uncertainty Quantification, submitted, 2018

[7] P.S. Zun, T. Anikina, A. Svitenkov, A.G. Hoekstra, A Comparison of Fully-Coupled 3D In-Stent Restenosis Simulations to In-vivo Data, Front. Physiol. 8 (2017) 284. doi:10.3389/fphys.2017.00284.

[8] Kurtzer GM, Sochat V, Bauer MW (2017) Singularity: Scientific containers for mobility of compute. PLOS ONE 12(5): e0177459.https://doi.org/10.1371/journal.pone.0177459

[9] Priedhorsky R, Randles TC. Charliecloud: Unprivileged containers for user-defined software stacks in HPC. Los Alamos National Lab. (LANL), Los Alamos, NM (United States); 2016. LA-UR-16-22370.

“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“