10.47952/gro-publ-204

Deployment of an HPC-Accelerated Research Data Management System: Exemplary Workflow in HeartAndBrain Study

V. Telezki[1,2,3] H. tom Wörden[4] F. Spreckelsen[4] H. Nolte[5] J. Kunkel[5] U. Parlitz[6,7,8] S. Luther[6,7,8,9] M. Uecker[1,2,10] M. Bähr[3]

***Abstract*—We present our workflow and research data management (RDM) within the HeartAndBrain research project of the Department of Neurology at the University Medical Center Göttingen. Here, we aim to investigate waste clearance mechanisms in the human brain [1], [2]. Therefore, we collect (longitudinal) data from multiple sources, in particular from Magnetic Resonance Imaging (MRI), ECG, SpO2, breathing belt, laboratory analysis of blood and urine. Our RDM System (RDMS) allows us to integrate these inhomogeneous data sources in one data base [3] where it is accessible via structured queries either via API or GUI. Furthermore, we developed (semi-) automatic post-processing pipelines that take care of routinely used post-processing steps. Computationally demanding tasks were set up to utilize high-performance computing (HPC) infrastructure, with automatic job submission and re-integration into the data base. Job submission can also be triggered via a GUI, which allows access to advanced, computationally demanding post-processing tools for non-expert users.**

***Index Terms*—workflow, big data, research data management, high-performance computing**

I. Introduction

We aim to characterize waste-clearance mechanisms in the human brain both in a healthy state and under selected pathological conditions. The goal is to establish surrogate markers for early detection of deficiencies in the waste-clearance system. To this end, we collected (longitudinal) data from healthy volunteers after approval from the local ethical committee. The collected data consist of multiple inhomogeneous sources, in particular an exhaustive magnetic resonance (MR) imaging protocol, both under native conditions as well as after contrast agent administration, physiological data from ECG, SpO2 and breathing belt, and laboratory analysis of blood and urine samples. The challenge here is to store a large amount of inhomogeneous data and make it findable and accessible for analysis in a highly cross-disciplinary research group. In particular, being able to access and select by meta-information of MR images in a quick and adaptable manner was a crucial aspect that needed to be realized within a Research Data Management System (RDMS). RDM is a developing field with various, different solutions. The core tool of our RDMS in this project is realized by the open-source toolkit LinkAhead[1]. An in depth analysis on how requirements of RDM are fulfilled by this tool and how it compares to other solutions can be found here [4]. Key aspects, why LinkAhead was chosen here, are its adaptability and open source access which allows integration of automated analysis pipelines, its flexibility of the underlying data model which enables management of (meta) data relevant to this project and its integrated semantic search lets us access data conveniently and precisely.

In the following, we present detailed aspects of our RDMS and workflow as realized within this research project.

II. Workflow

A. Data Integration

All acquired research data is first being anonymized and then stored according to the *Guidelines for a Standardized Filesystem* [5]. Data is then integrated automatically into the LinkAhead data base [3], [4], based on a custom data model, sorting MR imaging data and corresponding physiological and laboratory data. Key aspects of this custom data model were adapted from the Brain Imaging Data Structure (BIDS) [6]. For easier integration of meta-information from MR image data, we automatically convert images from the DICOM file format into NIfTI file format with meta-information residing in a sidecar file (JSON) [7]. Another advantage of the NIfTI file format is the dramatically reduced number of files that need to be handled in potential post-processing tasks compared to DICOM file format. This aspect is particularly important when analysing large data sets based on real-time MR imaging or diffusion tensor imaging were a single acquisition protocol typically results in thousands of DICOM files.

Integrated data can then be accessed and retrieved via structured queries (e.g. ”FIND T1WeightedImage”) within the

1. Cluster of Excellence Multiscale Bioimaging: from Molecular Machines to Networks of Excitable Cells (MBExC2067), University of Göttingen, Göttingen, Germany
2. Department of Interventional and Diagnostic Radiology, University Medical Center Göttingen, Göttingen, Germany
3. Department of Neurology, University Medical Center Göttingen, Göttingen, Germany
4. IndiScale GmbH, Göttingen, Germany
5. Gesellschaft für wissenschaftliche Datenverarbeitung mbH, Göttingen, Germany
6. Max Planck Institute for Dynamics and Self-Organization, Göttingen, Germany
7. Institute for the Dynamics of Complex Systems, University of Göttingen, Göttingen, Germany
8. German Center for Cardiovascular Research (DZHK e.V., partner site Niedersachsen), Göttingen, Germany
9. Institute of Pharmacology and Toxicology, University Medical Center Göttingen, Göttingen, Germany
10. Institute of Biomedical Imaging, Graz University of Technology, Graz, Austria

[1]https://gitlab.com/linkahead

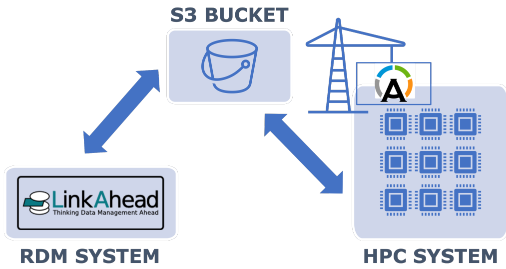

Fig. 1: Schematic illustrating communication between the RDM system and HPC system via an S3 bucket.

LinkAhead Graphical User Interface[2] (GUI) or its Application Programming Interfaces (APIs), e.g., via the LinkAhead Python Client[3], either individually or in custom defined groups for batch processing.

B. Post-Processing

In the process of trying to answer research questions, the collected data needs to be processed by typically computer-intensive algorithms and tools. Two particular examples in our study are volumetric segmentation and analysis of brain images and intravoxel-incoherent motion (IVIM) analysis [8], [9]. To use these computationally demanding tools efficiently, we utilize resources of high-performance computing (HPC) infrastructure. We realized the connection to the HPC system in the following way (see Fig. 1).

Pre-defined job scripts, including all necessary software (e.g. volumetric segmentation with FreeSurfer [10], or in house-implemented IVIM analysis) reside containerized on HPC-side. Job submission is triggered from LinkAhead GUI, while communication between HPC system and RDMS is realized via an S3 bucket and hence secured with a HTTPS protocol. Job status is monitored and can be tracked within the LinkAhead GUI. After the submitted job is finished, data is automatically transferred back and integrated into the RDMS, where it is stored including LOG files, ensuring a reliable and reproducible workflow. Therefore, once a post-processing task is set-up in such a manner, computationally demanding analysis steps can now be executed by non-expert users with a simple click of a button. In the future, we plan to transfer this workflow to a dedicated HPC API [11], [12], [13].

C. Details of the Communication Layer

When outsourcing compute-intensive tasks to an HPC system we need to establish communication between our RDMS and the HPC system within the user space of the individual user who triggered the computation. Requiring users to provide private SSH keys and therefore grant full terminal access to the RDMS is undesirable since it leaves both the individual user account and the HPC system in total at risk. In addition, the provider of such a RDMS is exposed to potential warranty claims. Furthermore, using ssh-based force commands is also discouraged, since it would require HPC administrators to add a `Match` clause to the `sshd_config` for each SSH key. Considering all theses issues, we chose to switch the used protocol altogether and rely on a REST interface where only pre-configured jobs can be executed. This solution offers the same security as force commands do while being completely configurable in user space.

To implement such a mechanism, we used a Simple Storage Service (S3) as a middle layer between the RDMS and the HPC system. Each job that is executed on the HPC system gets its own prefix assigned, comparable to a job id. Within this prefix, all input data for a job and a job specification file are provided. The job specification file, which is in JSON format, defines all required variables. These variables can include for instance the function which should be executed, arguments passed to this specific function invocation, path to input data, output data path, arbitrary environment variables used in the context of the HPC job, or a callback URL to the RDMS which is called after the function is finished.

In order to actually trigger execution of an HPC job, the RDMS creates a unique prefix, uploads all data and the job specification file in it, and creates a status file, where the status is set on `new`. On the HPC system, a pre-configured agent (cron job) is running which is monitoring the S3 bucket for new functions. Once this agent detects a new function prefix with a state `new` it will read the job specification file, stage the input data accordingly, and submit the job with all optional arguments and environment variables to the batch system. Then, the state of the function is updated to `running` and stores the corresponding Slurm job-id. The job status of all `running` jobs is tracked by the agent. After a job is finished, the output data is uploaded back into the S3 bucket into the same prefix, the state is updated to `finished` and an optionally provided callback URL is triggered to alarm the RDMS that the job has been finished and that the output data can be fetched from the S3 bucket.

This mechanism assumes that the user has access to the S3 bucket and that all functions, that shall be executed, are already available in the user's `HOME` directory, ideally as a containerized image.

This workflow has the advantage, that it can be implemented completely in user space and no additional firewall settings are required. Therefore, it is very portable. It also drastically reduces potential damage when access tokens are leaked, since an attacker can only execute pre-configured jobs and can not gain full shell access which is required to exploit a local vulnerability. However, the here illustrated workflow does not scale arbitrarily, since traversing through all prefixes by the agent is (unnecessarily) time-consuming. In addition, one has to implement any synchronization carefully since the eventual consistency of S3 does not allow syncing processes via file locks.

Another way is to use a dedicated service, which provides a REST interface for an HPC system, e.g., [11], [12], [13]. In this case, it can be used as a drop-in replacement for steering the control flow, i.e., submitting commands from the RDMS to the HPC system, because it also uses similar JSON files to communicate job specifications. In addition, it is transaction-

[2]https://gitlab.com/linkahead/linkahead-webui
[3]https://gitlab.com/linkahead/linkahead-pylib

safe, and it scales almost without any overhead. However, these systems are still experimental, thus requiring the above-described communication layer for production systems.

The different advantages and disadvantages of the discussed methods are summarized in Table I. Therefore, considering the easy transition of our workflow to a dedicated HPC REST API, our current implementation can be considered superior to SSH-based access due to its security and user space compatibility.

III. Summary

We presented parts of our RDM within the HeartAndBrain project at the University Medical Center Göttingen. At the center of this workflow is a RDMS realized with the LinkAhead toolkit and a data model adapted from BIDS. Particular computationally demanding post-process tasks were carried out utilizing HPC resources. Communication between RDMS and HPC system was established via an S3 bucket. Once all requirements within this workflow for a particular post-processing task are met, this task can then be executed by non-expert users with a simple click of a button, ensuring a reliable and reproducible workflow.

Acknowledgment

We thank for the computing time at NHR@Göttingen as part of the NHR infrastructure during testing period. We acknowledge funding by Else Kröner-Fresenius Foundation and the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy—EXC 2067.

Conflict of interests

H. tom Wörden is co-founder of IndiScale GmbH, a company that provides commercial services for the RDM toolkit LinkAhead. Furthermore, H. tom Wörden and F. Spreckelsen are currently employed at IndiScale GmbH.

References

[1] H. Mestre, Y. Mori, and M. Nedergaard, "The Brain's Glymphatic System: Current Controversies," *Trends in Neurosciences*, vol. 43, no. 7, pp. 458–466, Jul. 2020.

[2] M. Nedergaard, "Neuroscience. Garbage truck of the brain," *Science (New York, N.Y.)*, vol. 340, no. 6140, pp. 1529–1530, Jun. 2013.

[3] T. Fitschen, A. Schlemmer, D. Hornung, H. tom Wörden, U. Parlitz, and S. Luther, "CaosDB—Research Data Management for Complex, Changing, and Automated Research Workflows," *Data*, vol. 4, no. 2, p. 83, Jun. 2019.

[4] D. Hornung, F. Spreckelsen, and T. Weiß, "Agile Research Data Management with Open Source: LinkAhead," *ing.grid*, vol. 1, no. 2, Jan. 2024.

[5] F. Spreckelsen, B. Rüchardt, J. Lebert, S. Luther, U. Parlitz, and A. Schlemmer, "Guidelines for a Standardized Filesystem Layout for Scientific Data," *Data*, vol. 5, no. 2, p. 43, Jun. 2020.

[6] K. J. Gorgolewski, T. Auer, V. D. Calhoun, R. C. Craddock, S. Das, E. P. Duff, G. Flandin, S. S. Ghosh, T. Glatard, Y. O. Halchenko, D. A. Handwerker, M. Hanke, D. Keator, X. Li, Z. Michael, C. Maumet, B. N. Nichols, T. E. Nichols, J. Pellman, J.-B. Poline, A. Rokem, G. Schaefer, V. Sochat, W. Triplett, J. A. Turner, G. Varoquaux, and R. A. Poldrack, "The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments," *Scientific Data*, vol. 3, no. 1, p. 160044, Jun. 2016.

[7] X. Li, P. S. Morgan, J. Ashburner, J. Smith, and C. Rorden, "The first step for neuroimaging data analysis: DICOM to NIfTI conversion," *Journal of Neuroscience Methods*, vol. 264, pp. 47–56, May 2016.

[8] D. Le Bihan, E. Breton, D. Lallemand, M. L. Aubin, J. Vignaud, and M. Laval-Jeantet, "Separation of diffusion and perfusion in intravoxel incoherent motion MR imaging." *Radiology*, vol. 168, no. 2, pp. 497–505, Aug. 1988.

[9] D. Le Bihan, "What can we see with IVIM MRI?" *NeuroImage*, vol. 187, no. December 2017, pp. 56–67, 2019.

[10] B. Fischl, "FreeSurfer," *Neuroimage*, vol. 62, no. 2, pp. 774–781, 2012.

[11] S. Bingert, C. Kohler, W. Alamgir, and H. Nolte, "An API to Include HPC Resources in Workflow Systems," in *INFOCOMP 2021: The Eleventh International Conference on Advanced Communications and Computation*, 2021, pp. 15–20.

[12] M. H. Biniaz, S. Bingert, C. Köhler, H. Nolte, and J. Kunkel, "Secure Authorization for RESTful HPC Access," in *INFOCOMP 2022, The Twelfth International Conference on Advanced Communications and Computation*, 2022, pp. 12–17.

[13] C. Köhler, M. H. Biniaz, S. Bingert, H. Nolte, and J. Kunkel, "Secure Authorization for RESTful HPC Access with FaaS Support," *International Journal on Advances in Security*, vol. 15, no. 3 and 4, pp. 119 –131, 2022.

Method	Security	User Space	Transaction Safe	Scalability	Portability
SSH Keys	-	+	+	+	+
SSH Force Commands	+	-	+	+	-
our workflow	+	+	0	-	+
HPC API	+	+	+	+	0

TABLE I: Qualitative comparison of the different evaluated communication layers. A + indicates that the requirement is always fulfilled, a 0 indicates that it is not always fulfilled but might be, and a - states that this requirement is never fulfilled.