Real-time Web-based Visualisation:

The ViaLactea Service in NEANIAS

Evgeniya Malikova and Mel Krokos
School of Creative Technologies, University of Portsmouth, UK

Introduction

ViaLactea is a toolkit of visual analytics interfaces for the Astrophysics and Planetary sectors facilitating exploitation of multi-wavelength (from near infrared to the radio spectrum) observations of the Milky Way [VSB+18]. The underlying ecosystem is strongly driven by FAIR principles[1] and Open Science practices[2] and is integrated within the European Open Science Cloud (EOSC[3]). ViaLacteaWeb (VLW[4]) is a recently developed toolkit component for an integrated web solution for cloud-based visualisation functionality. The aim is to provide a testbed for realising remote and interactive correlation analysis workflows to study star formation processes in the Milky Way. VLW sports a rich web UI supporting multiple user scenarios exploiting a remote knowledge database that provides object catalogues and spectral energy distribution model outputs (VLKB[5]). The overall performance is limited only by the remote visualisation scalability capacity of the underlying infrastructure and it can be CPU and/or GPU based. VLW is accessed via a dedicated client running in common web browsers and is based on specific VLKB queries to perform sophisticated interactive visualisation operations remotely.

Real-time web-based visualisation

Cloud infrastructures provided as a service (IaaS) is one of the most convenient solutions adopted for high-performance data processing and visualization [OCJ+15]. Modern web-based visualisation services commonly rely on client-server architectures. A server is deployed to deal with scalability of data processing and rendering while a web client displays the obtained images, pulls any data upon user requests as necessary and handles user interaction together with other not so computationally expensive tasks. Such services also provide a great degree of accessibility for end users as they can be deployed from within common web-browsers either on desktop or mobile platforms requiring only standard internet connections. Typically client-server communication protocols involve WebSockets[6] and REST[7] interfaces as supported by modern web browsers as well as handling of binary streams and JSON and XML formats for asynchronous data pulling. Then Node.js is often

---

[1] https://www.openaire.eu/how-to-make-your-data-fair
[2] https://ec.europa.eu/info/research-and-innovation/strategy/strategy-2020-2024/our-digital-future/open-science_en
[3] https://digital-strategy.ec.europa.eu/en/policies/open-science-cloud
[4] https://vlw.readthedocs.io/en/latest/
[5] https://vlkb.readthedocs.io/en/latest/
[6] https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
[7] https://developer.mozilla.org/en-US/docs/Glossary/REST

used as a base to implement a client as it provides all core library packages required. On the server side the upper coding layer for CPU or GPU deployment is provided through appropriate scripting bindings, e.g. by using Python [SKR+15], [FH20], [MK15] or Go [PR17]. The server side remote visualisation performance depends on a variety of aspects such as input data types, specific graphics pipeline mechanisms deployed, e.g. rasterization or ray-tracing, and these need to be adapted within the context of the available GPU/CPU environments [WJA+16], [PBD+10]. Often there is no ready off-the-shelf visualisation solution to deploy directly and some adapting/optimising according to the underlying cloud environment capabilities is required. The architecture of VLW (see next section) has been realised within the space services of NEANIAS[8] and is based on Kitware's approach [SML06]. We support multi-user operational scenarios for remote visualisation by integrating into our system proxy configuration [MFGM04] (as conventionally these are forwarded to the client) and user authentication and authorisation modules [JATG19].

ViaLactea Web

The VLW functionality and operational scenarios are inspired by VLVA[9] and are realised as a service fully integrated in NEANIAS which is establishing cutting edge space services for the EOSC ecosystem. The VLW focus is to provide a dedicated service for scalable web-based visualisation [OCJ+15], [PR17], [JATG19] exploiting existing knowledge extraction solutions [MBB+16]. We deploy VLKB to access astrophysics data catalogues and integrating within a modular architecture thus allowing design of customised astrophysics visualisation pipelines within Python for WebSockets streaming. Computationally expensive visualisation is performed on the server, while our client deploys a lightweight UI designed with Node.js that can be accessed through modern web browsers. VLW has so far been deployed on a dedicated server (GPU based rendering) and the GARR cloud[10] infrastructure (CPU based rendering). To enable multi-user support, VLW adapts a conventional configuration architecture based on Apache proxy server and WSLink launcher as used in the ParaViewWeb[11] solution. The overall schema of VLW is shown in Fig. 1; the core architectural components are:

[8] https://www.neanias.eu/
[9] https://vlva.readthedocs.io/en/latest/
[10] https://cloud.garr.it
[11] https://kitware.github.io/paraviewweb/docs/ubuntu_14_04.html

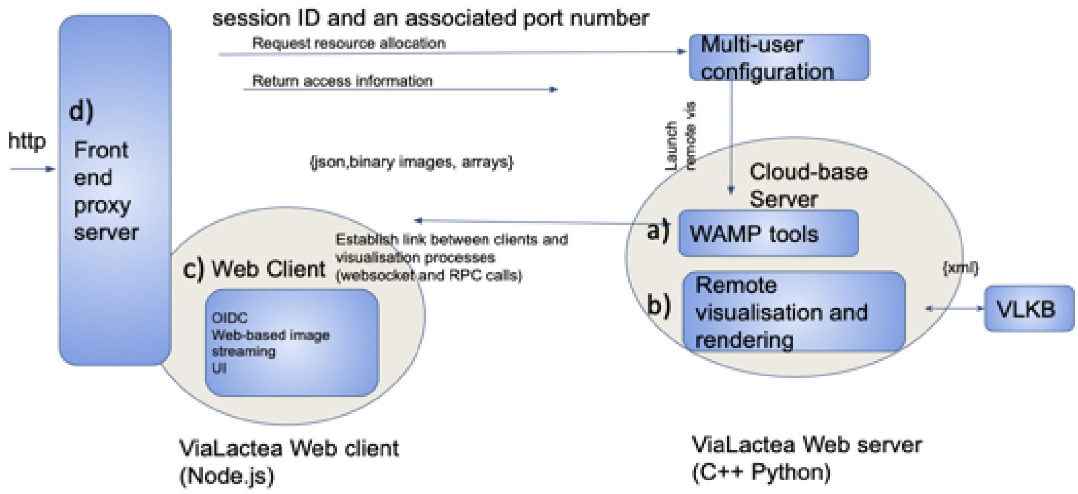

Figure 1: VLW concepts and technologies involved.

- A mechanism of management of WebSocket procedures for visualisation, e.g. handling binary image streaming and UI execution of Remote Procedure Calls (RPC). The Web Application Messaging Protocol can simplify the design of complex WebSocket solutions [WAM] (see Fig. 1 (a));
- A remote and interactive cloud-based visualisation mechanism to execute VLKB queries and generate off-screen visualisations of the results (see Fig. 1 (b));
- A web client that runs in common modern browsers and provides web rendering and UIs for user operations (see Fig. 1 (c));
- A multi-user configuration manager with a front-end proxy server and additional tools to provide resource allocation (e.g. port available), configuration (e.g. session ID) and starting a fresh visualisation process for each user that requests this (see Fig. 1 (d)).

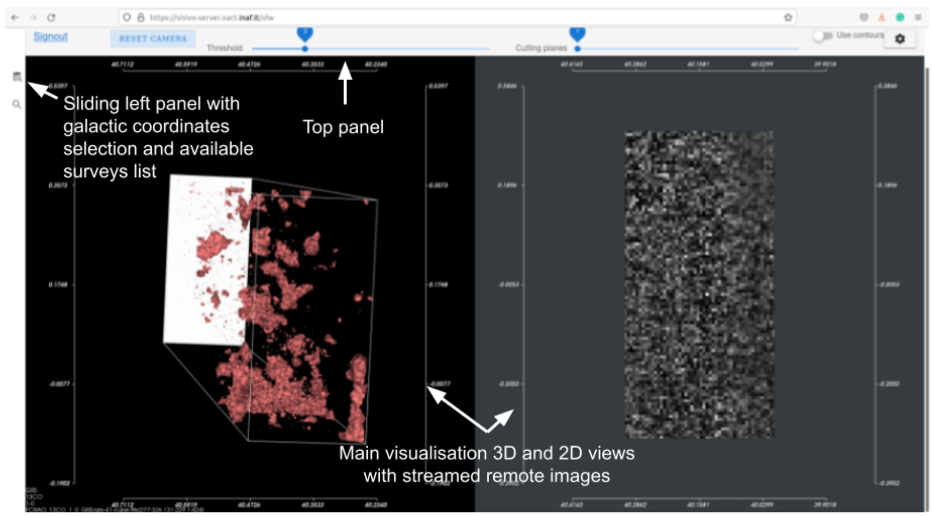

Figure 2: The main UI of VLW showing a survey dataset loaded

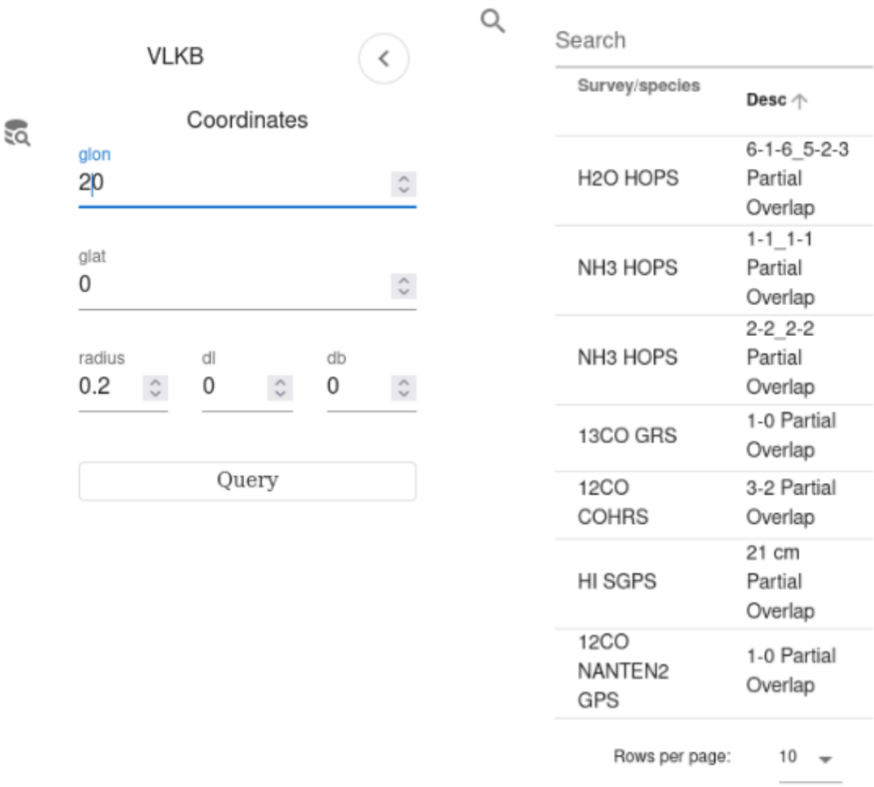

Figure 3: The VLW left sliding panel showing the galactic coordinates selection menu together with the identified list of available visualisation survey datacubes.

The main UI of VLW is shown in Fig. 2. To access a selected survey represented as a datacube cut [MHKJH16] the user has executed VLKB queries through the sliding left panel functionality. Firstly, the parameters of the selection region in galactic coordinates need to be defined. Secondly, the available VLKB surveys are loaded and can be subsequently selected by the user specifically for subsequent datacube visualisation (see Fig. 3).

Conclusion

We have presented VLW which is an integrated solution realised as a NEANIAS service to target high performance remote and interactive visualisation for data analysis in astrophysics. The VLW architecture is designed with accessibility in mind - the adopted solution can be accessed from any modern browser on any platform. Furthermore, its remote visualisation capability can be adapted for different cloud infrastructures both CPU and GPU based. VLW has recently undergone through formal validation and has been subsequently released within the NEANIAS space services. VLW is envisaged to be developed further with additional interactive procedures for astrophysics data analysis and further integration with other services within the NEANIAS ecosystem.

[FH20] Loraine Franke and Daniel Haehn. Modern scientific visualizations on the web. Informatics, 7(4), 2020.

[JATG19] Vahid Jalili, Enis Afgan, James Taylor, and Jeremy Goecks. Cloud bursting galaxy: federated identity and access management. Bioinformatics, 36(1):1–9, 06 2019.

[MBB+16] Marco Molinaro, Robert Butora, Marilena Bandieramonte, Ugo Becciani, Massimo Brescia, Stefano Cavuoti, Alessandro Costa, Anna M. Di Giorgio, Davide Elia, Akos Hajnal, and et al. Vialactea knowledge base homogenizing access to milky way data. Software and Cyberinfrastructure for Astronomy IV, Aug 2016.

[MFGM04] Jens Muller, Stefan Fischer, Sergei Gorlatch, and Martin Mauve. A proxy server-network for real-time computer games. In Marco Danelutto, Marco Vanneschi, and Domenico Laforenza, editors, Euro-Par 2004 Parallel Processing, pages 606–613, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[MHKJH16] P. Mohan, C. Hawkins, R. Klapaukh, and M. Johnston-Hollitt. Three tools to aid visualisation of fits files for astronomy, 2016.

[MK15] David Masad and Jacqueline Kazil. Mesa: An agent-based modeling framework. pages 51–58, 01 2015.

[OCJ+15] Patrick O'Leary, Mark Christon, Sébastien Jourdain, Chris Harris, Markus Berndt, and Andrew C. Bauer. Hpccloud: A cloud/web-based simulation environment. In 7th IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2015, Vancouver, BC, Canada, November 30 - December 3, 2015, pages 25–33. IEEE Computer Society, 2015.

[PBD+10] Steven G. Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, and Martin Stich. Optix: A general purpose ray tracing engine. ACM Trans. Graph., 29(4), jul 2010.

[PR17] Daniel Probst and Jean-Louis Reymond. FUn: a framework for interactive visualizations of large, high-dimensional datasets on the web. Bioinformatics, 34(8):1433–1435, 11 2017.

[SKR+15] Tarek Sherif, Nicolas Kassis, Marc-Étienne Rousseau, Reza Adalat, and Alan C. Evans. Brainbrowser: distributed, webbased neurological data visualization. Frontiers in Neuroinformatics, 8:89, 2015.

[SML06] Will Schroeder, Ken Martin, and Bill Lorensen. The Visualization Toolkit–An Object-Oriented Approach To 3D Graphics. Kitware, Inc., fourth edition, 2006.

[VSB+18] F. Vitello, E. Sciacca, U. Becciani, A. Costa, M. Bandieramonte, M. Benedettini, M. Brescia, R. Butora, S. Cavuoti, A. M. Di Giorgio, and et al. Vialactea visual analytics tool for star formation studies of the galactic plane. Publications of the Astronomical Society of the Pacific, 130(990):084503, Jun 2018.

[WAM] The web application messaging protocol. https://wamp-proto.org.

[WJA+16] Ingo Wald, G.P. Johnson, J. Amstutz, C. Brownlee, Aaron Knoll, Jim Jeffers, J. Gunther, and Paul Navratil. Ospray – a cpu ray tracing framework for scientific visualization. IEEE Transactions on Visualization and Computer Graphics, 23:1–1, 01 2016.