The 3rd Italian Conference on Big Data and Data Science

Workshop Scientific HPC in the pre-Exascale era

FROM LOCAL TO REMOTE: VISIVO VISUAL ANALYTICS IN THE ERA OF THE SQUARE KILOMETRE ARRAY

18/09/2024 – CNR Pisa

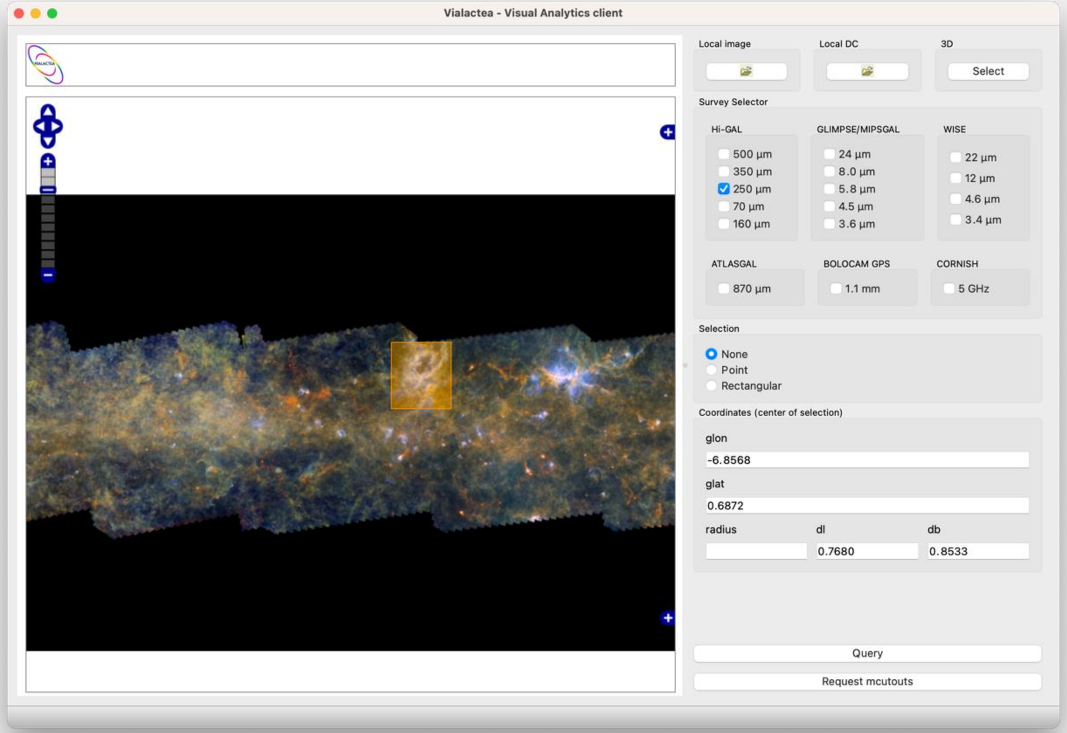
Giuseppe Tudisco

VISIVO SUITE HISTORY

2005	VisIVO Desktop
2009	VisIVO Server
2010	VisIVO Web
2011	VisIVO Library
2013	VisIVO Science Gateway
2014	VisIVO Mobile
2015	Vialactea Visual Analytic (VLVA)
2017	VLVA - EOSC
2019	VLVA - NEANIAS
2021	Cirasa
2022	

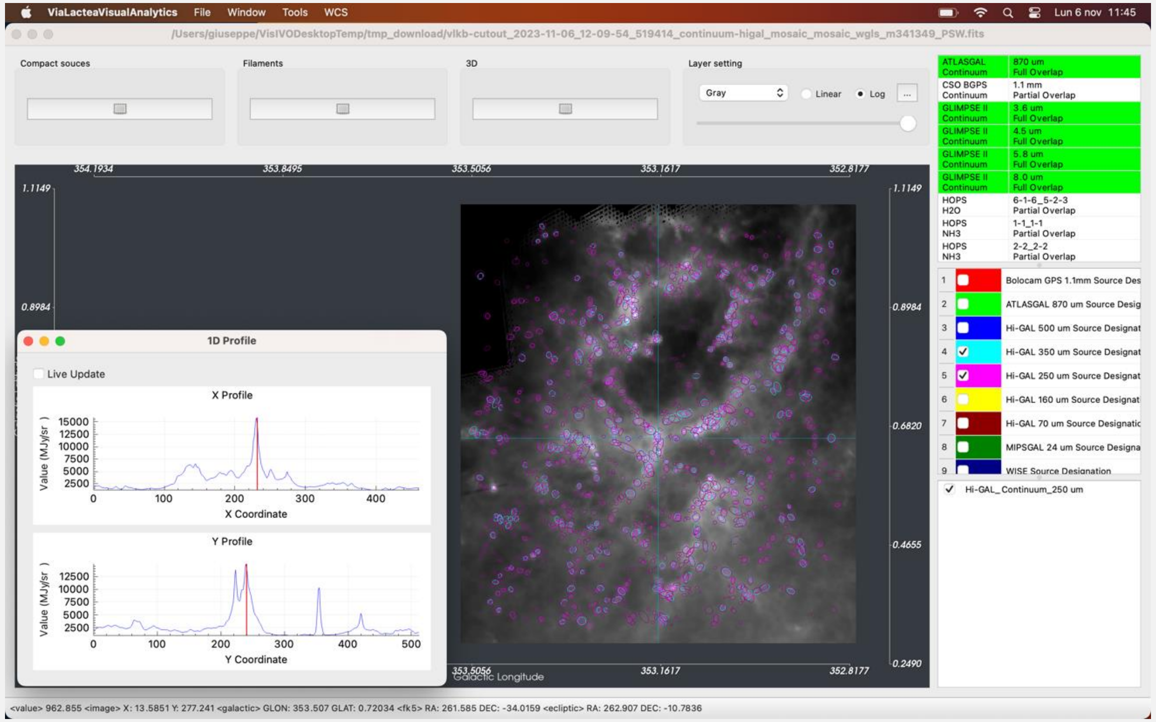
VISIVO VISUAL ANALYTICS

- Joined the VisIVO Framework in 2015.
- Open-source desktop application that provides a visual analytic environment to analyze the correlation between different kinds of data, such as 2-D intensity images with 3-D molecular spectral cubes.
- Development supported by many projects during past years (H2020 NEANIAS, INAF PRIN CIRASA, ERC ECOGAL).



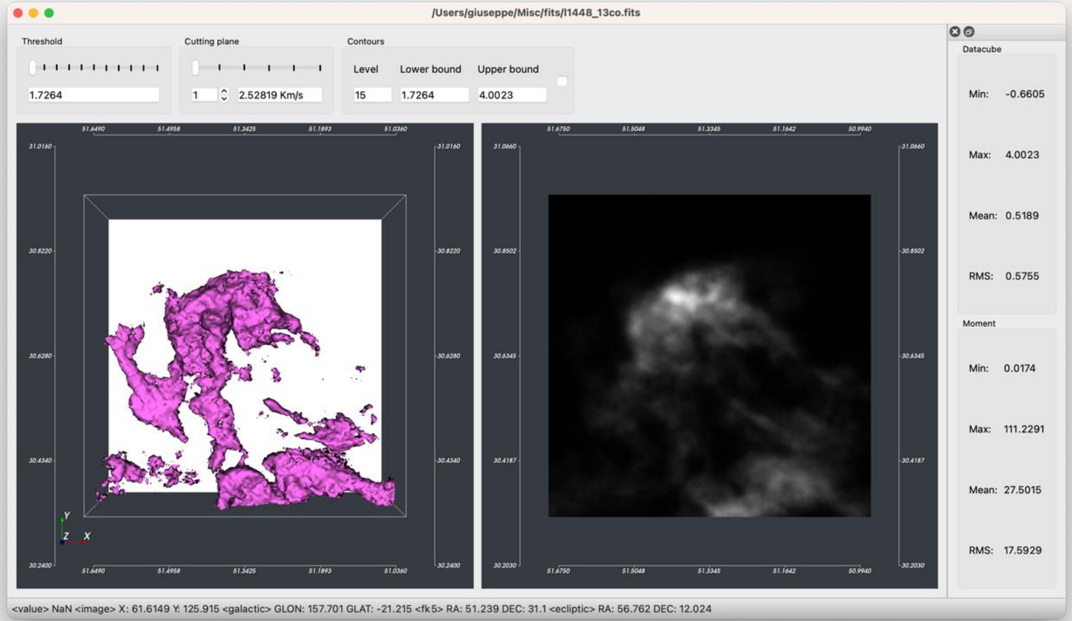
FEATURE OVERVIEW

- 2D Image visualization.
- Image layers stack.
- Visualization of compact and extended sources.
- Spectral Energy Distribution analysis.



FEATURE OVERVIEW

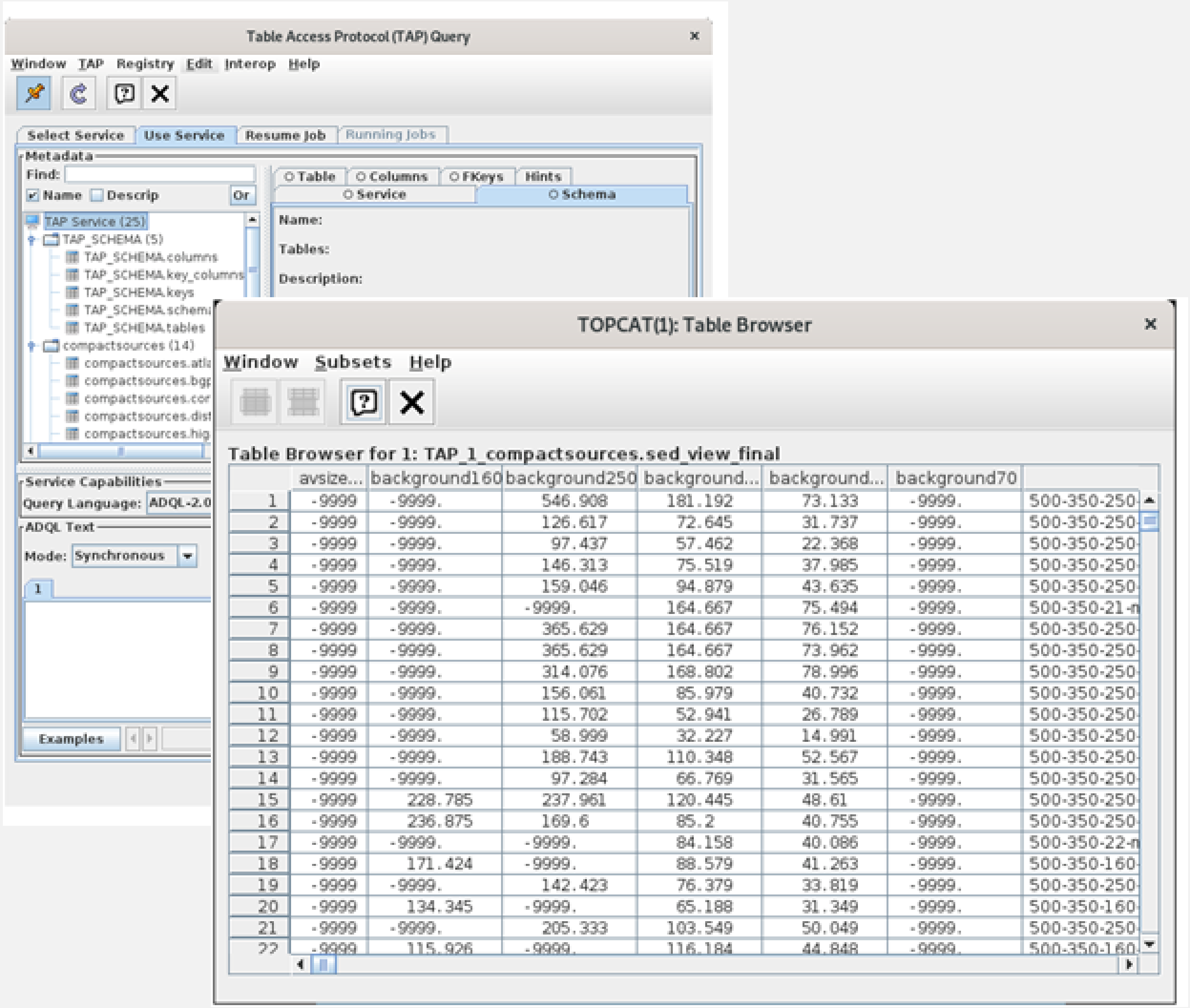
- Volume rendering.
- Slice visualization.
- Moment maps.
- Position-Velocity diagrams.
- Spectra plot.



VIALACTEA KNOWLEDGE BASE (VLKB)

The ViaLactea Knowledge Base (VLKB) identifies a set of data collections, catalogues and services enabling discovery and access on them. It consists of:

- a TAP service to deploy the catalogue data contents, and
- a dataset search, cutout and merge combined service API for accessing 3D radial velocity cubes and 2D images.



SKA REGIONAL CENTRES (SRC)

Each year the SKAO will generate around 700 PB of data.

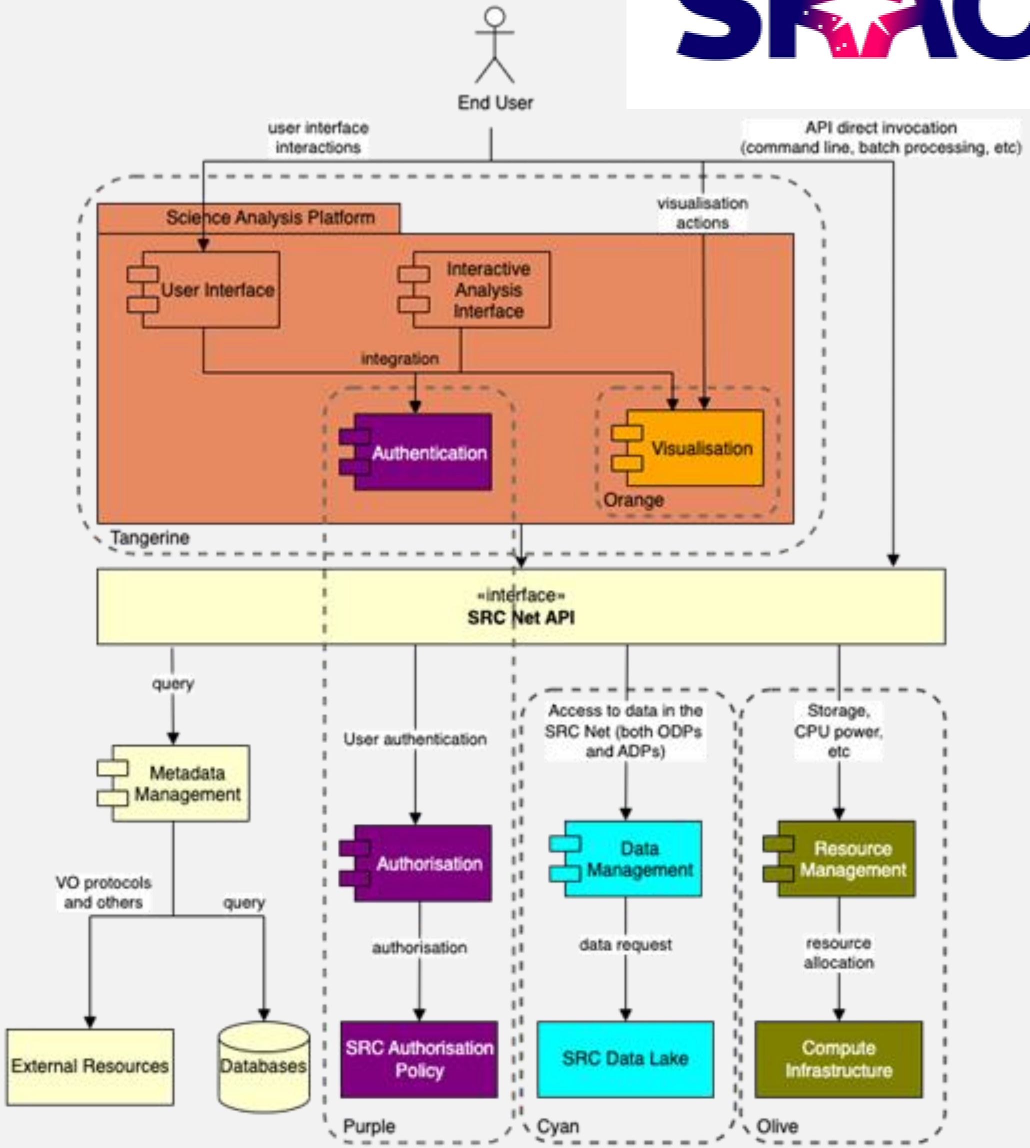
The data volumes are so large that direct delivery to end users is not practical.

Functionalities to find, access, manipulate and visualize SKA data products needs to be made available on shared computational resources.

An SKA Regional Centres (SRC) is a virtual entity that provides:

- access to data products
- platforms for advanced scientific research
- a place for development of software tools (analysis, modelling, visualization)

A global network of SRCs distributed around the world is being developed by SRCNet Agile Teams.



REMOTE VISUALIZATION

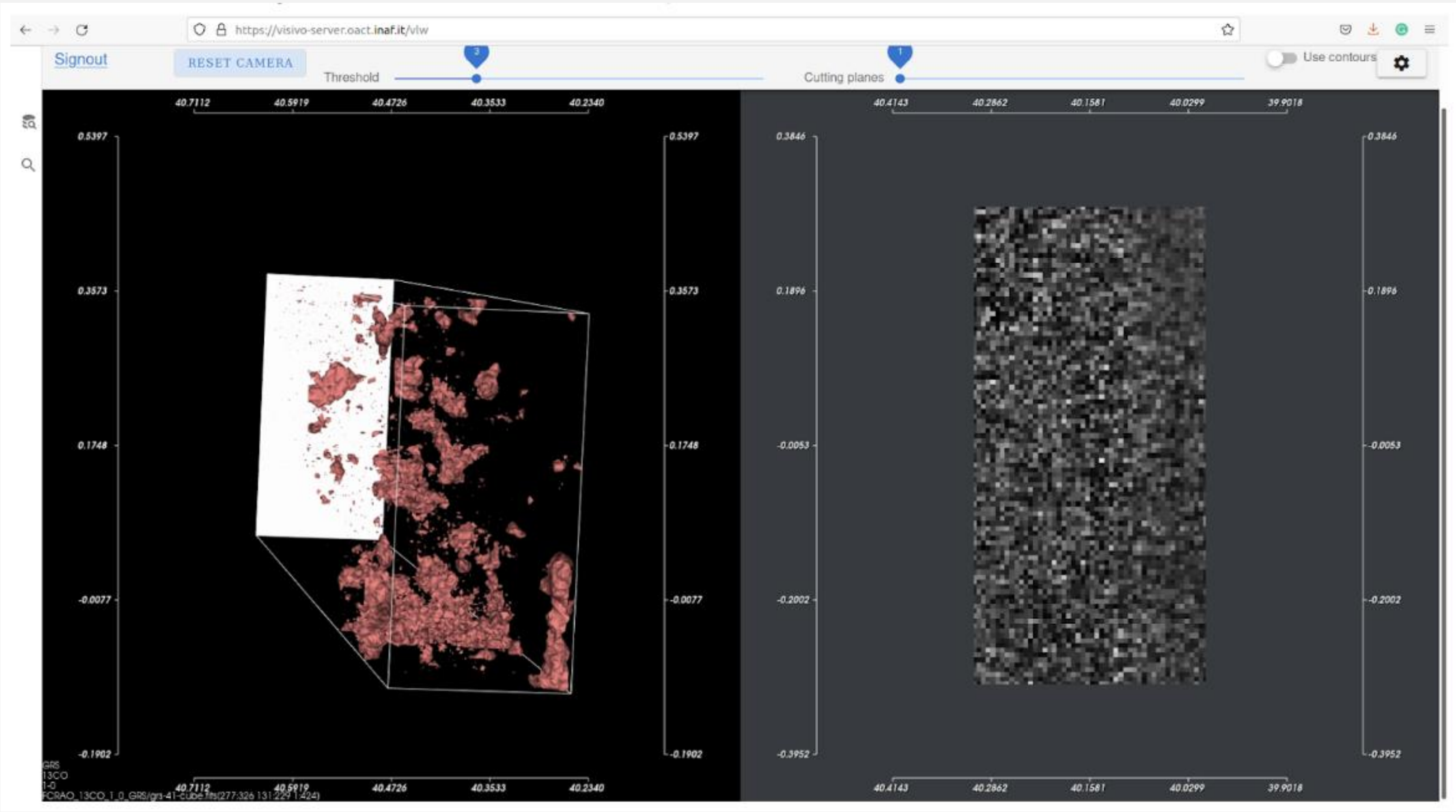
- Remote Visualization refers to any visualization where some or all of the data is on a different machine from the one used to look at the images.
- Benefits
  - Storage: the data to visualize may exceed by several orders of magnitude the storage and memory available in user's machine.
  - Computing: such large amount of data cannot be handled by personal computers in an interactive way.
  - Bandwidth: no need to move the data.

We have explored many approaches to achieve Remote Visualization in VisIVO:

- Web application
- Remote container
- Science Platform
- Client-server architecture

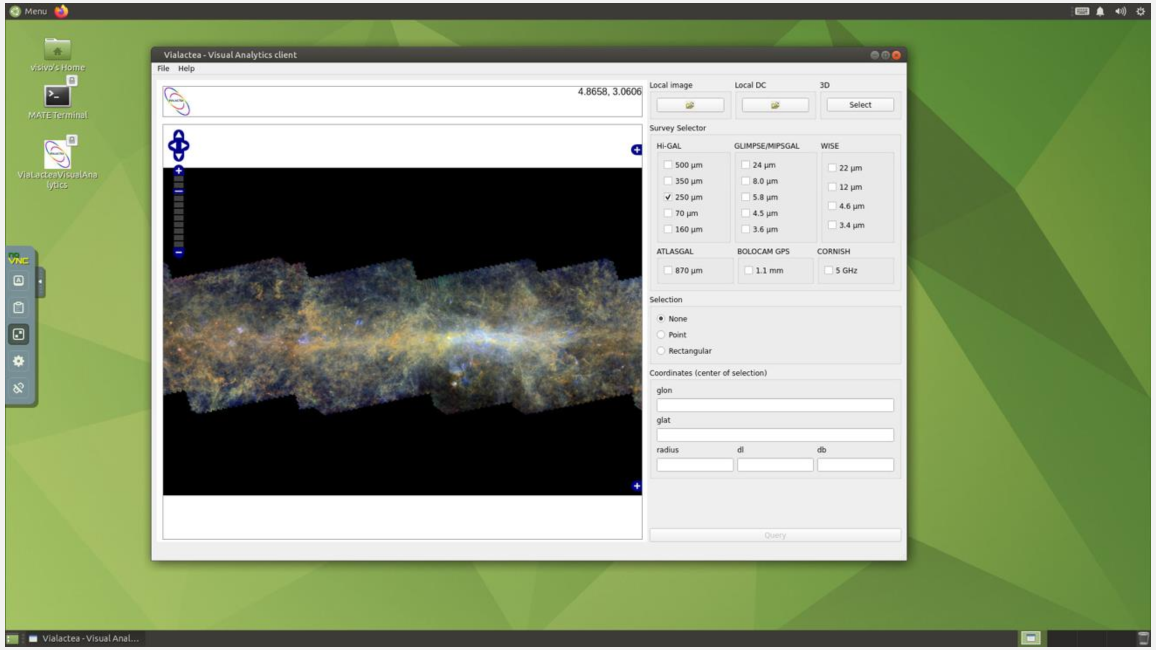
VIALACTEA WEB

- Simplified web version developed in collaboration with University of Portsmouth (UK)
- Frontend Node.JS
- Backend Python, C++



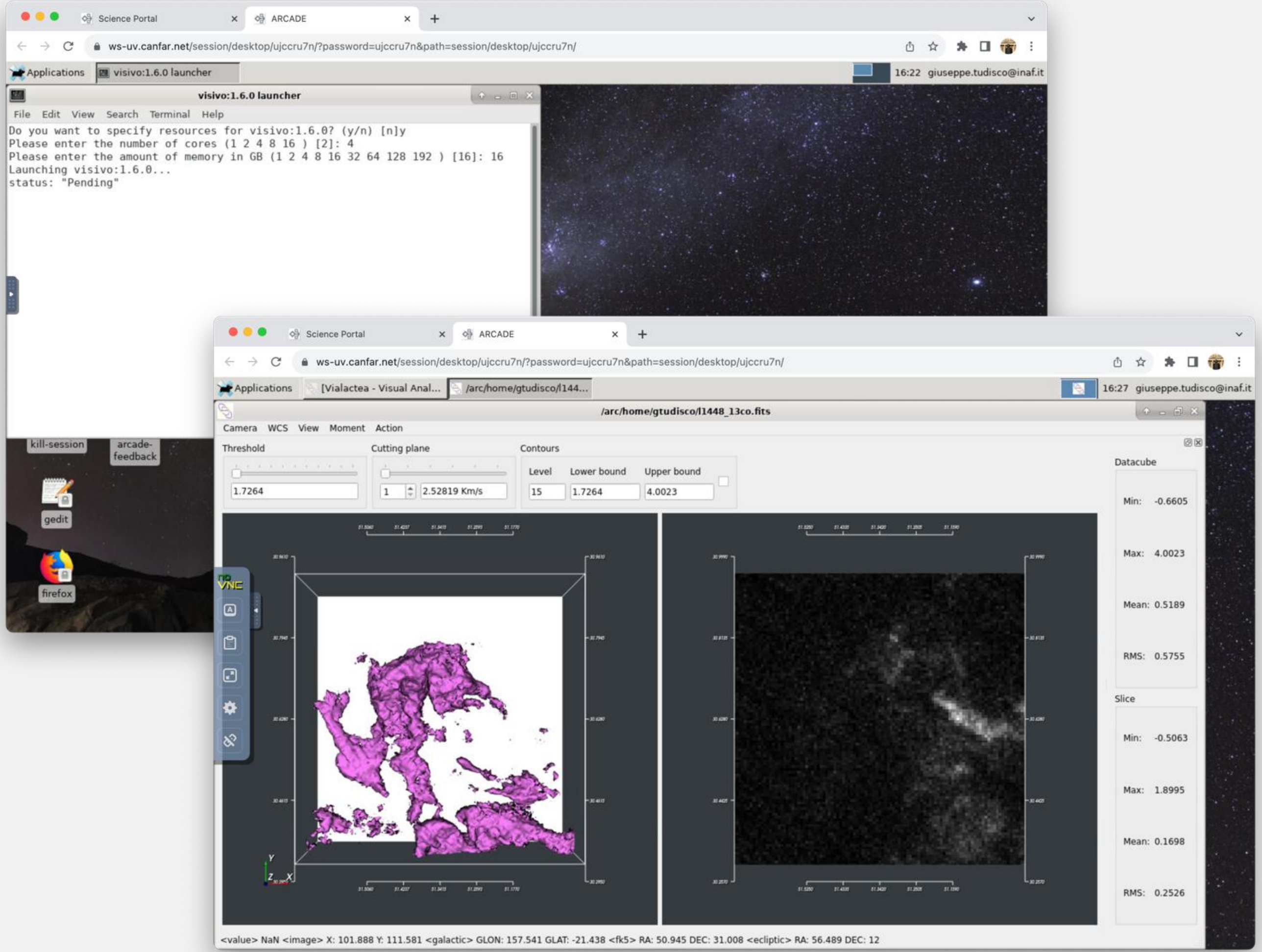
REMOTE CONTAINER

- Containerized as it is.
- No need to change code.
- Can be easily deployed on any server running Docker.
- Accessible via noVNC in a browser.
- Limited scalability.



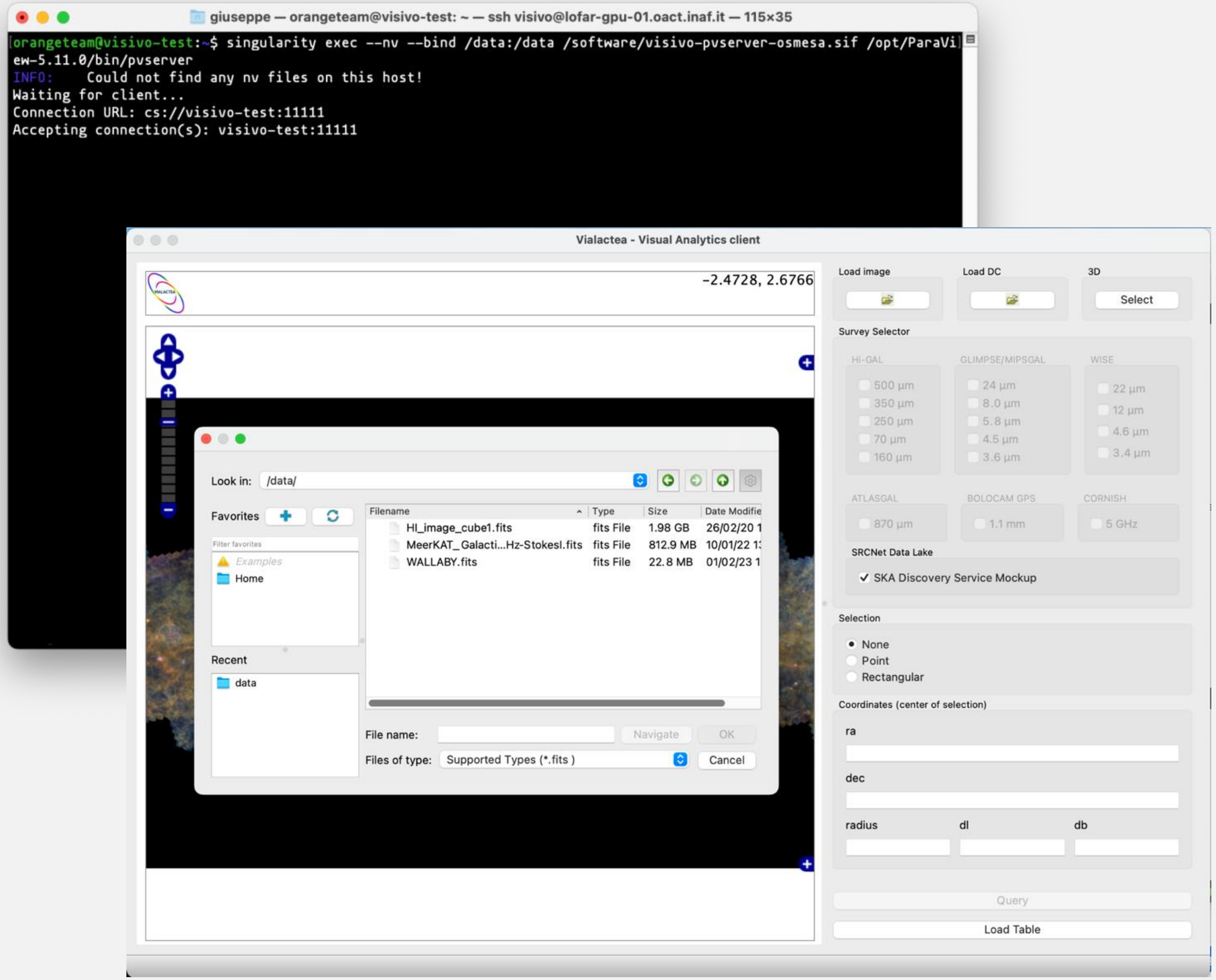
SCIENCE PLATFORM

- We have recently made available VisIVO on CANFAR Science Platform.
- Custom Dockerfile (different from our own version) due to skaha requirements.
- Can be assigned resources at startup in a desktop session.



CLIENT-SERVER ARCHITECTURE

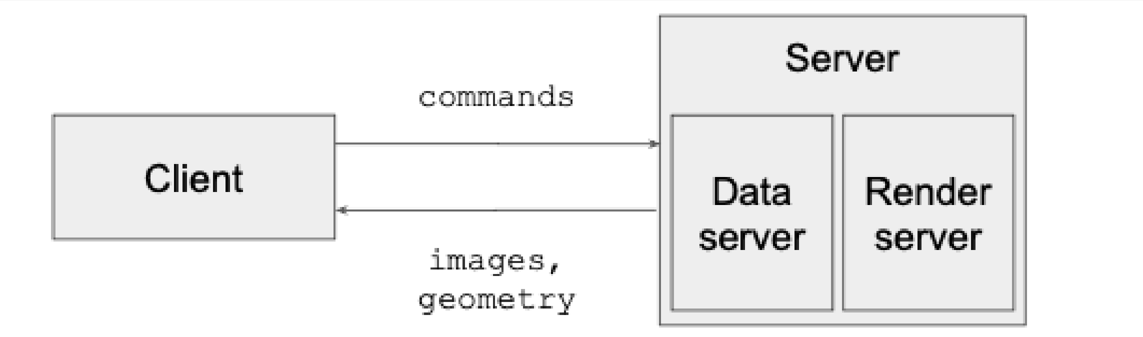
- Rewriting parts of the main application to adopt a client-server architecture.
- Only the server has to be containerized and deployed.
- Fully scalable (HPC oriented).
- Can achieve also Parallel Visualization.



VISIVO BASED ON PARAVIEW

Three main logical components:

- a client, responsible for the user interface,
- a data server, to read and process data sets to create final geometric models,
- a render server, which renders that final geometry.



REMOTE +PARALLEL VISUALIZATION

Processing the data in parallel, simultaneously using multiple workers. Workers can be different processes running on a multicore machine or on several nodes of a cluster.

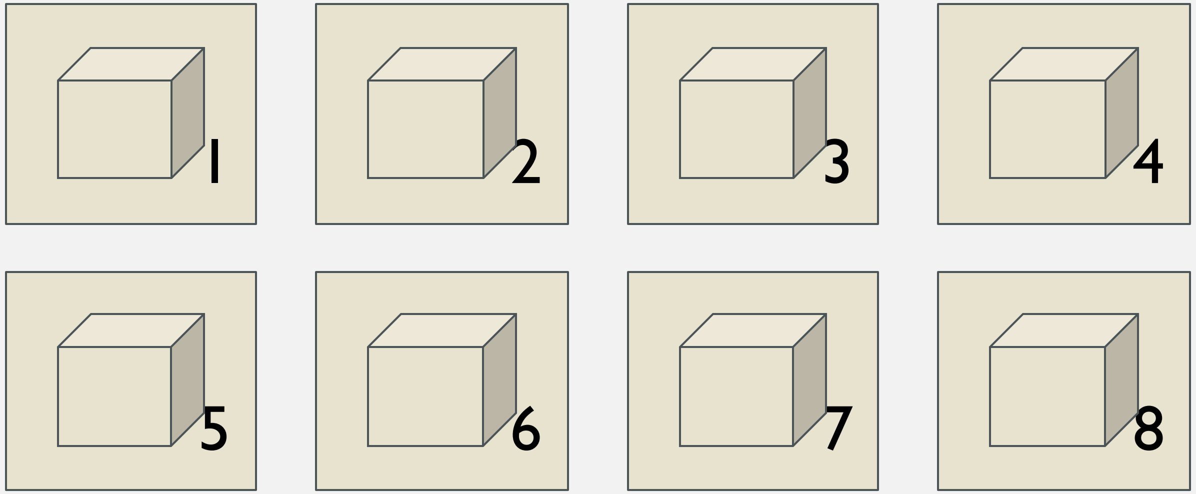
The data server and the render server in this case are a set of processes that communicate with MPI.

Each render server receives geometry from data servers in order to render a portion of the screen.

PARALLEL VISUALIZATION

Steps:

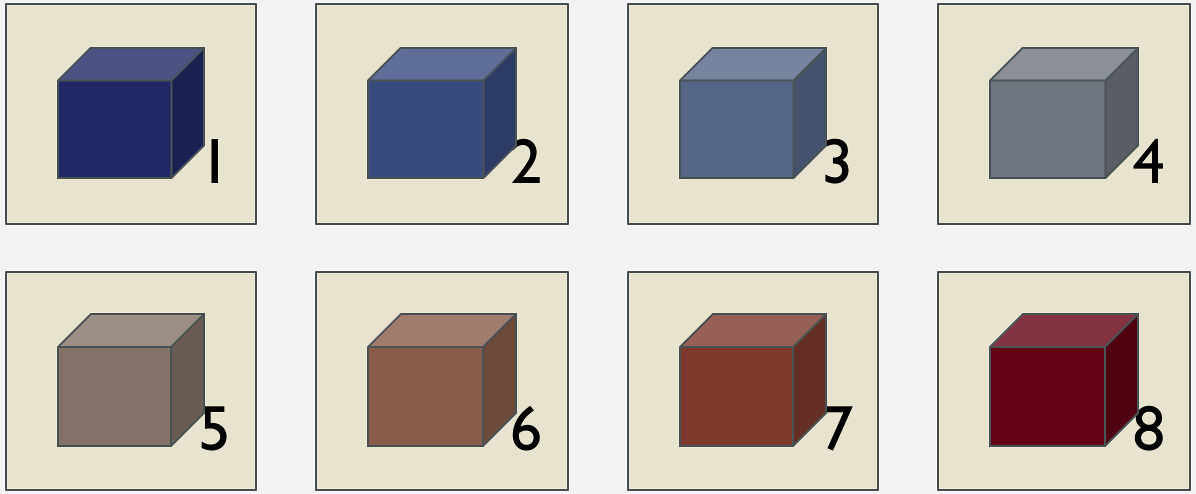
1. Partitioning input



PARALLEL VISUALIZATION

Steps:

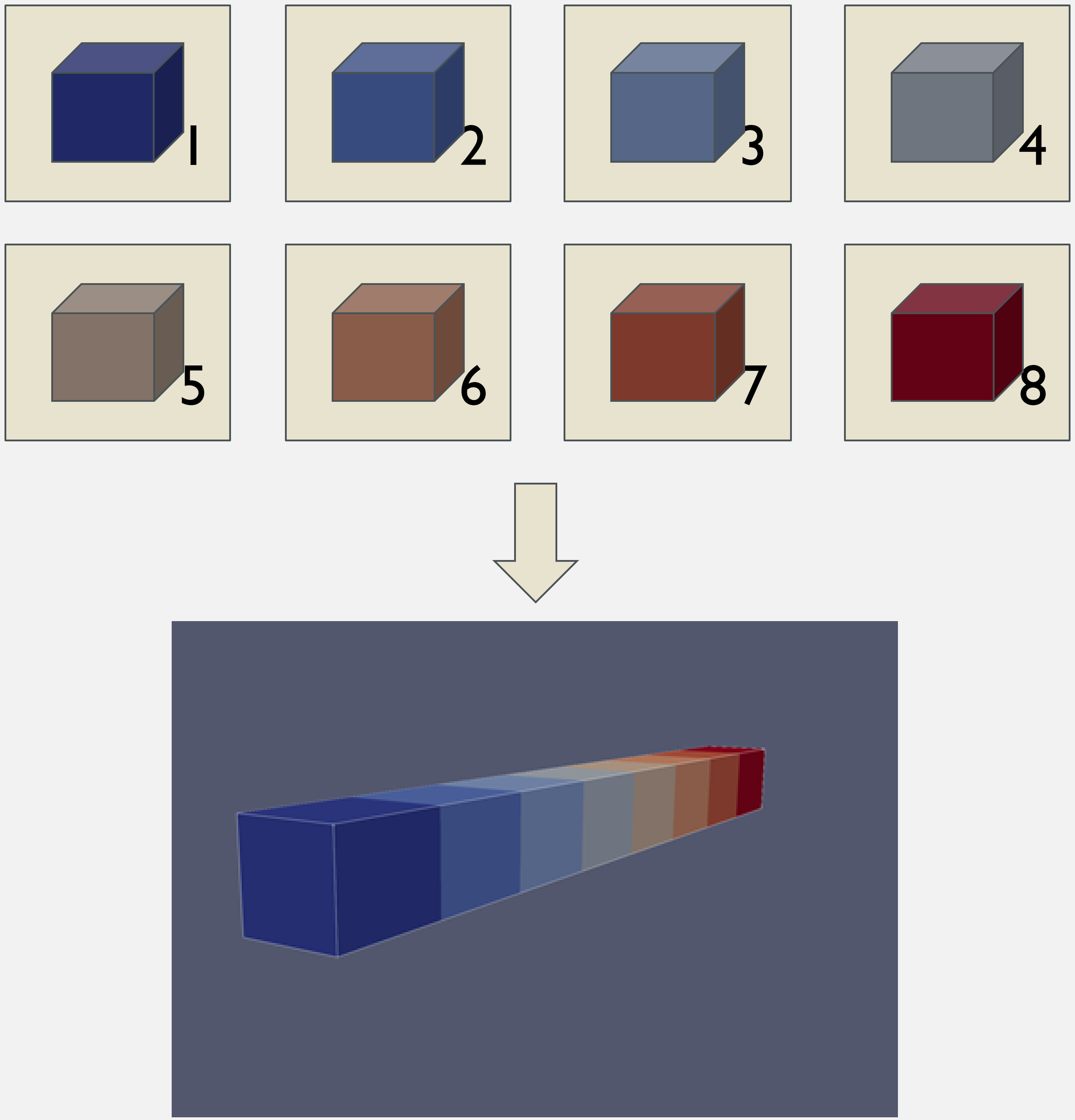
1. Partitioning input
2. Each worker works on its own chunk of data and produces partial results



PARALLEL VISUALIZATION

Steps:

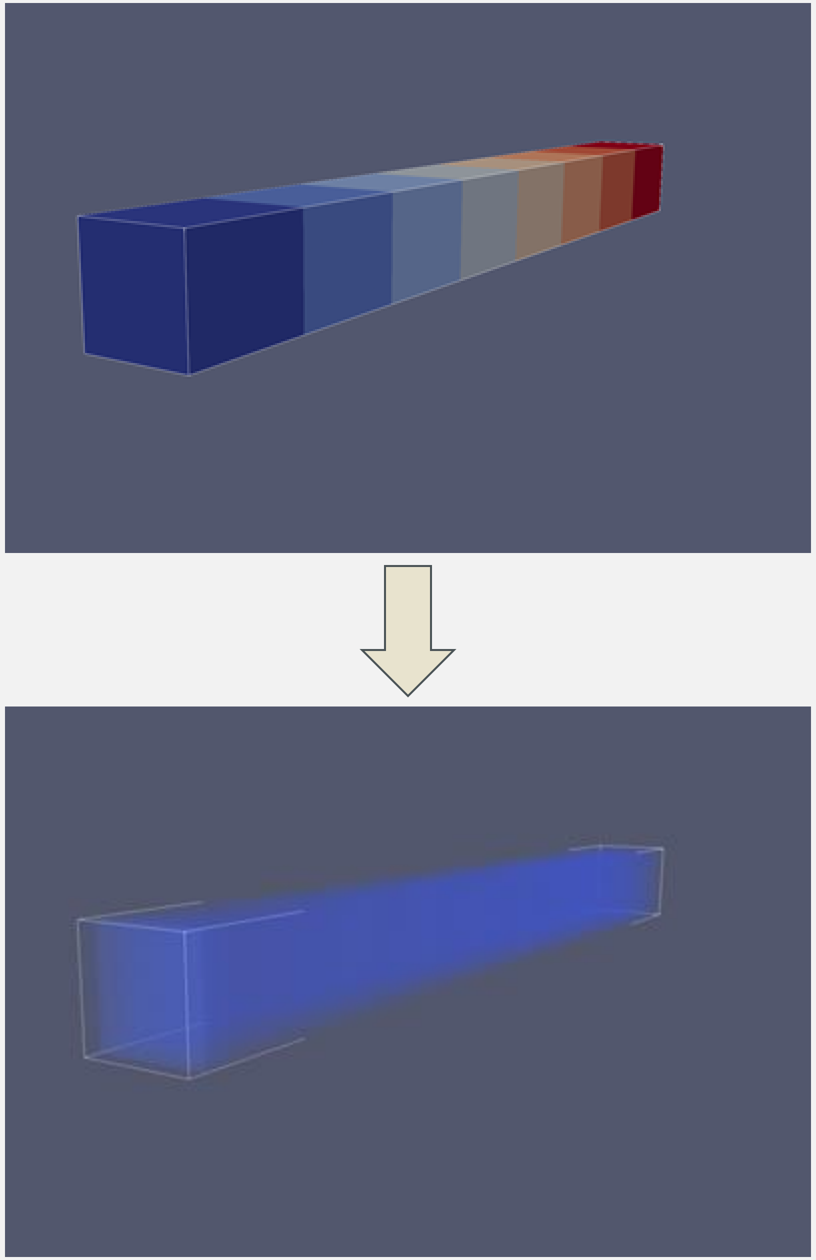
1. Partitioning input
2. Each worker works on its own chunk of data and produces partial results
3. Construct the final output



PARALLEL VISUALIZATION

Steps:

1. Partitioning input
2. Each worker works on its own chunk of data and produces partial results
3. Construct the final output
4. Send the result to the client



MANAGER MIDDLEWARE

- To facilitate interaction between client and server, we are also developing a service (VisIVO-pvManager).
- Clients will be able to launch new server instances, retrieve logs and monitoring process status.
- Manager will also support the porting of features from desktop-only to client-server architecture (es VLKB interaction).

HTTP Method	**Route**	**Description**
GET	/info	Retrieves server information and tells clients if an MPI runtime has been detected.
POST	/server	Starts a new server instance. The client specifies the port and, if allowed, the number of MPI processors for parallel execution.
GET	/server	Get info on a running server instance such as start time, process status, and working directory.
GET	/logs	Retrieves the logs of the currently running server instance, providing insights into execution details and errors (if any).

CONCLUSIONS

- We have explored different approaches to move from a local only visualizer to a remote-capable visualizer.
- The client-server based architecture, despite requiring more rewriting work, is the most efficient and suitable solution for achieving both remote and parallel visualization.

Thanks!