

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/319867871>

TurBase: A Software Platform for Research in Experimental and Numerical Fluid Dynamics

Conference Paper · July 2017

DOI: 10.1109/HPCS.2017.18

CITATIONS

0

READS

117

10 authors, including:



Luca Biferale

University of Rome Tor Vergata

267 PUBLICATIONS 5,562 CITATIONS

SEE PROFILE



Herman J.H. Clercx

Technische Universiteit Eindhoven

254 PUBLICATIONS 3,076 CITATIONS

SEE PROFILE



Federico Toschi

Technische Universiteit Eindhoven

271 PUBLICATIONS 5,400 CITATIONS

SEE PROFILE



Giovanni Erbacci

Cineca

40 PUBLICATIONS 324 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



European joint doctorate (HPC-LEAP) in computational fluid dynamics [View project](#)



HPC-LQCD [View project](#)

All content following this page was uploaded by [Herman J.H. Clercx](#) on 19 November 2017.

The user has requested enhancement of the downloaded file.

TurBase: a software platform for research in experimental and numerical fluid dynamics

R.Benzi, L.Biferale and
F.Bonaccorso (corresponding author)
Department of Physics and INFN
University of Tor Vergata
Via Della Ricerca Scientifica 1, 00133
Rome, Italy
Email: fabio.bonaccorso@roma2.infn.it
Phone: +39 06 7259 4504

H.J.H. Clercx, A.Corbetta,
W.Möbius and F.Toschi
Department of Applied Physics
University of Eindhoven
Eindhoven, Netherlands

F.Salvadore, C.Cacciari and G.Erbacci
Department of SCAI, CINECA
Bologna, Italy

Abstract—We present a software infrastructure for the research community working on turbulence and complex flows (TurBase), an easily accessible web platform for high quality data. Its main goal is to host, standardize and manage a large collections of heterogeneous experimental and numerical data sets from high-end European fluid dynamics experimental facilities and from High Performance Computational centres. TurBase offers scalable performances when accessing/uploading/searching data, providing at the same time maximum flexibility and power (through Jupyter notebooks) when doing online computation directly on big datasets

Index Terms—Big-data, repository, web platform, online analysis.

I. INTRODUCTION

In the turbulence research community there is often a need for long term availability of high quality experimental and numerical data: advances in the analysis techniques can also lead to new results. One of the keys to address this issue is the creation of a large data repository which is both user-friendly when posting results while maintaining top speed when searching for particular experiments and when trying to use data produced by other research groups (different file formats, different rules for organizing data, etc).

The John Hopkins turbulence repository (JHTDB [2]) and the 4TU.ResearchData portal (previously known as 3TU.Datacentrum [3]) are two examples offering interesting insights into the features to be implemented not least from a technological point of view. The JHTDB offers data from 4 datasets with a simple and well known data structure. Obviously the service offers no generic search capability but an inner dataset search is implemented relying on the specific structure of the datasets. Querying the values of custom points or performing simple post-processing activities (such as differentiation or interpolation) is possible via a simple web interface; full access is possible via common programming languages. Such an interface, tightly coupled with the underlying data, requires ad-hoc implementations of post-processing tools which, however, can be guaranteed through several restrictions on the file format and the data structures. The 4TU.ResearchData repository has many desirable features, such as a complex metadata definition, a search engine,

the navigation of data files in a directory-like fashion, and an advanced support for field specific file formats (HDF5 or NetCDF).

In this paper we present the experience developed to create the online web repository of the EuHIT project (European High performance Infrastructures in Turbulence)[1]. EuHIT is a consortium meant to integrate cutting-edge European facilities for turbulence research across national boundaries, in order to significantly advance the competitive edge of European turbulence research with special focus on providing the knowledge for technological innovation and for addressing grand societal challenges. Current members of EuHIT include 25 research institutes and 2 industrial partners from 10 European countries. Total 14 cutting-edge turbulence research infrastructures, most of which are developed on national funds and are run by EuHIT member institutes, consist the material basis of EuHIT. These infrastructures, together with the knowledge developed upon them, are interconnected by a Networking program and Joint Research Activities within EuHIT.

Due to the heterogeneity of the research laboratories involved and of the applications studied, the data repository required: (i) an open-access approach, (ii) handling complex metadata structure, (iii) special care in exploiting high network bandwidth between research centres, (iv) the ability of working with big data and with large amount of small files, (v) the need of a good search engine, and (vi) to enable online work on the data beside the download. All the previous issues are not fully addressed in the already developed examples.

We considered some well established solutions to create a scientific data repository with open access such as:

- CKAN [4]
- Nesstar [5]
- Dataverse [6]
- Midas Platform [7]
- Invenio [8]

Notably each of these focuses in one aspect relaxing the other requirements, with no complete solution for this big data integrated platform.

In the following, we present our software platform, named **TurBase**, which addresses these problems in a integrated fashion, based on the four years interaction with the turbulence community from both experimental and numerical fields.

II. ARCHITECTURE

Our architecture, as shown in fig. 1 is based on several open source software and is structured in the following components:

- Our primary **storage** facility, the Digital Library of Turbulence Data (DLTD), is the data-storage which hosts most of the repository, with essential features including accessibility, data safety, etc. The DLTD may offer additionally services like federation of distributed storages. It leverages the state-of-the-art iRODS[9] technology in managing big data at HPC centers.
- The **database** and the **metadata**: Scientific data are grouped into datasets, which refer to single experiments or sub-cases. They are annotated with data descriptors, using a versioned syntax (XML based), to enable search using keywords and/or categories. The database itself contains only the tables for mapping keywords to datasets and is currently implemented using SQLite (this is not mandatory and can be easily replaced with higher performance distributed options as the database communication occurs through an abstraction layer).
- The **search engine**: it offers the usual “online” interface for finding datasets in the database, with real-time hints for word completion.
- The **web portal** and **APIs** of TurBase: This is the software stack responsible for dataset access and interfacing with the end-user. It comes either as a user-friendly web-portal or as callable web APIs.
- The **interactive portal**: it enables the end-users to develop and execute scripts for data analysis. The whole process, inclusive of e.g. plots of the data, occurs while connected to the TurBase web page. Notably (i) several programming languages are allowed, (ii) no client software is requested user-side, (iii) the execution occurs in a read-only way for data safety.

III. HOSTING SITES

Data from experimental measurements and numerical simulations may require huge amount of storage. Most of the datasets of interest for EuHIT will derive from gathering the time repetitions of the same base data (e.g. a camera snapshot or spatial configuration of numerical outputs). Moreover, since the range of active relevant scales in a turbulent flow might well exceed over a few orders of magnitude, the amount of information to be stored is large for most experiments as well as for almost all numerical computations. A common experimental dataset in TurBase needs on the order of 1TB of storage; our installation currently hosts more than 150TB. Some results are obtained using hardware that needs a long calibration procedure, so it is crucial to preserve the data. The TurBase architecture allows one to host the data in a

distributed set of storage servers (**hosting sites**) to fulfill the needed High Availability and for storage bandwidth/space.

In our installation at CINECA we rely on a consolidated iRODS technology for its inherent strengths:

- Designed to manage large sets of files, i.e. large amount of files and/or files with large sizes.
- High-performance network data transfer. iRODS transfers data across the network in an integrated manner (get/put, read/write; parallel threads for large files), efficiently using up to 70% of available bandwidth. Its GridFTP plugins provides same performance via simple web interface (GlobusOnline [10])
- Controlled access. iRODS provides fine-grained controlled access, by user or group.

Besides iRODS servers, each hosting site can also use its native file system, accessed (by the back-end Web portal) using SSH.

IV. ACCESS RULES

The default access to TurBase is read-only: users can view the datasets but no upload of further data is possible. TurBase users can create datasets once they are granted a special permission by an administrator: at that moment they become part of a group of users – TurBase **organization** – with permissions to store data on a set of hosting sites, as **editors**. The organization is the owner of the dataset. This feature allows one to exploit the same search engine capabilities of TurBase without requiring the data to be moved. This enables data from different groups to be reachable within the same portal.

The TurBase infrastructure is designed with open access in mind, but also taking into account the confidentiality need of fresh experimental data produced by research groups: datasets can be created directly to be of public domain, requiring only accepting a disclaimer for scientific work based on them, or can be restricted to the users of the owner organization. The duration of the period in which the data cannot be accessed by other users is at most 2 years, though this is an implementation detail. After this **embargo** period, they become publicly accessible. In any case, the metadata are always public, i.e. the datasets can be immediately found through the search engine. Each organization can decide to use one of the available hosting sites for optimized access time to the data. In fact, choosing an hosting site with good bandwidth to one of the Web servers speeds up page serving and online computations. This can lead to complex scenarios as in fig. 2.

V. METADATA

An exhaustive classification of data is the first step to ensure their efficient recovery and usage of the data. In TurBase we defined three sets of metadata for each dataset:

- **Basic**: this is the minimal set of metadata which defines the name of the dataset, the owner organization, the hosting site, and the embargo information. At the moment of the dataset creation, the storage folders at the hosting

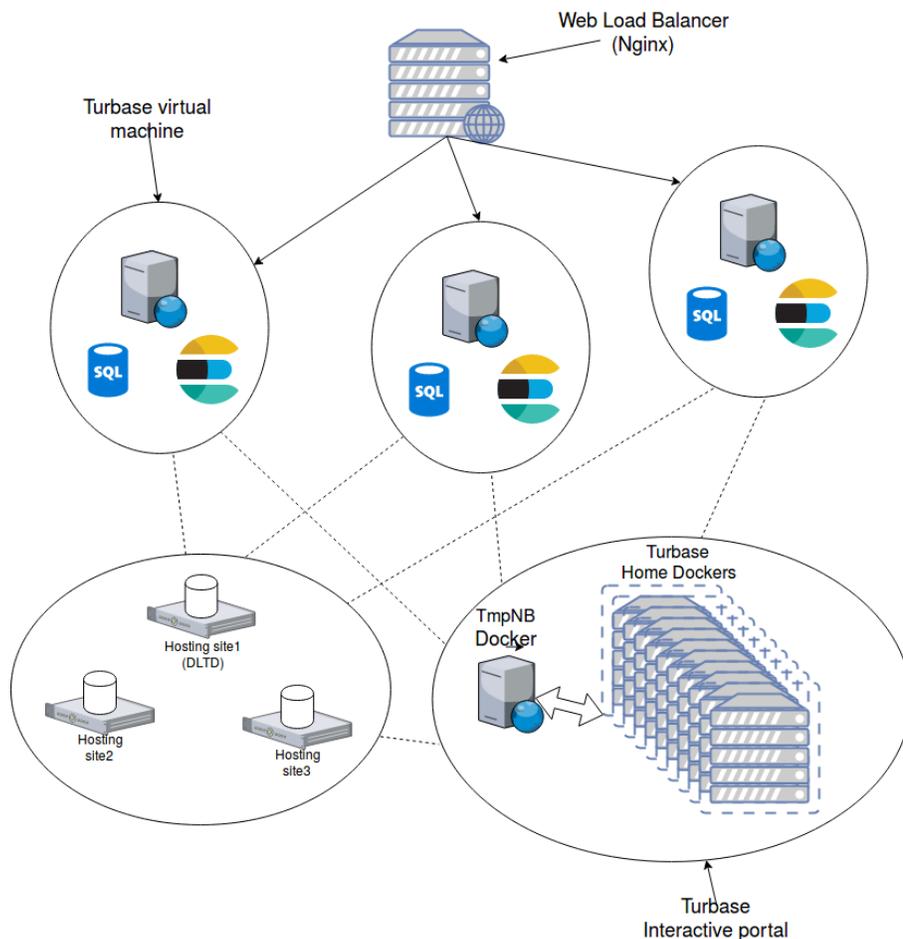


Fig. 1. TurBase software architecture: the web load balancer directs to the TurBase web servers, which access data from different hosting sites and collaborate with the Service Portal

site are defined. Afterwards, that these metadata cannot be changed without an administrator intervention.

- **Presentation:** this set contains some additional information which can help quickly present the dataset, i.e. a title, a short description and tags. A predefined set of tags – category tags – are proposed when creating the dataset but free tags can be added by the user, also. Word completion is active to suggest already used words to prevent the proliferation of different synonyms for the same concept. Periodically, the TurBase administrator can promote the most used free tags to category tags thus allowing the website to adapt to the needs of users.
- **Detailed:** this set of metadata contains the detailed information. The detailed metadata *schema* is configurable by the administrator via the web portal, i.e. many *schema* can be active in TurBase and the dataset author is able to select the *schema* which best fits his/her needs. In TurBase we propose a simple *schema* based on key-values

plus two more complex options suitable optimized for scientific data. The most advanced *schema* contains five groups of metadata:

- *Contacts* includes details on the authors of the dataset, with their role in producing the data;
- *Science* includes information on the scientific description with links to the publications using the data;
- *Tools* specifies details on the tools used to produce the data;
- *Support* contains description and links on codes capable of reading/writing the data. Having codes capable of reading the data is crucial to make it really usable by other scientists. When creating a dataset, a *support* folder is automatically created by TurBase server so that the user can upload support data – e.g. source codes – useful to understand the data of the dataset;

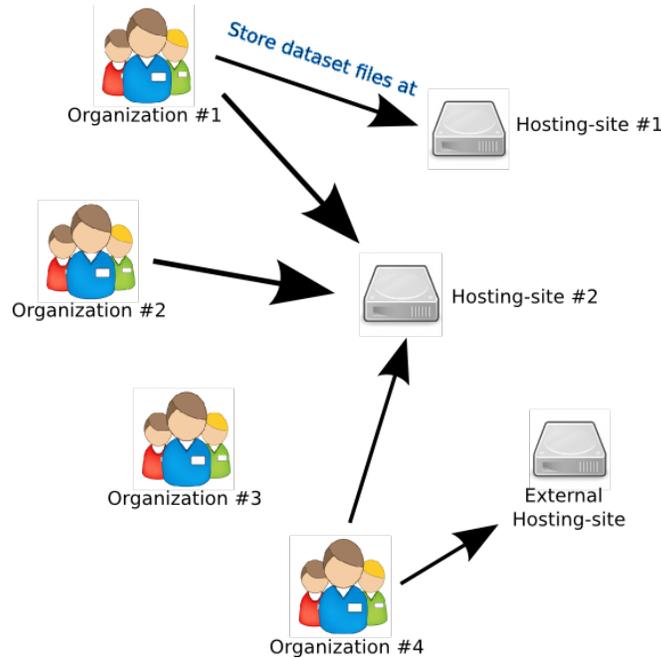


Fig. 2. Hosting sites and organizations

- *Quality* is the only section which specifically refers to turbulence information, e.g. allowing to define how the flow has been generated, commenting on the statistical accuracy, on the post-treatment applied to the data, etc...

VI. FILE TRANSFER: NETWORK BANDWIDTH LIMIT AND SOLUTIONS

Since TurBase is a web platform, the speed of data transfer is the aspect most impacting on the user experience. The software permits a simple web browser based download and upload of data, although this does not provide peak performances. To achieve almost 70% efficiency in using the available network bandwidth, TurBase provides a way to use GridFTP clients (`globus-url-copy`) or web based interfaces (GlobusOnline). In this case only the GSI protocol is supported. The performance can be optimized through fine-tuning options and the usability is good. Each user can register to TurBase and in his/her profile there are the parameters needed for the **High Speed** transfer mode, which can be used even for TBs of data with only a few clicks on GlobusOnline. The infrastructure will take care of the process and will send an email when the transfer is complete. This transmission length is currently limited to a week, but this will go without any user intervention.

VII. WEB USER INTERFACE

The web portal, hosted at <https://turbase.cineca.it>, is implemented in Python, using: (i) **Web2py** [11] (a robust framework for creating web servers), which handles natively authentication and offers an easy way to provide APIs to users for

the underlying operations of the web-server; (ii) **AngularJS**, [12] used as front-end framework, with intuitive interaction for the users; (iii) **ElasticSearch** [13], a search-engine library, which is powerful and highly customizable and (iv) **Nginx** web server for load balancing and proxying. The portal is hosted in a virtual machine that can be replicated for scaling with increasing number of concurrent users, to offer a comfortable user experience.

The main workflows in TurBase are:

- **Dataset creation and data uploading**
TurBase is modeled after the concept of datasets. Each editor of TurBase can create a dataset containing a related group of experimental or numerical data. Several editors can be defined for a single dataset, allowing concurrent modifications and access to the data (even during the embargo period). The editor is in charge of defining an initial set of metadata, guided by the web interface: this metadata can be refined at a later point, adding new keywords to better describe the dataset.
TurBase also offers a native way to upload data to a dataset (called direct upload) that can be used without any further configuration step by the editor, while remaining on the TurBase Portal website. Under the dataset section there is a tab named “files”. A simple interface will pop-up that permits to upload files, to create subdirectories, and to delete files. The interface translates these intuitive actions into functions operating on the different hosting sites, transparently to the end-user. To check the correctness of the transfers the checksums can be downloaded. In a first step the editor selects files from his/her computer

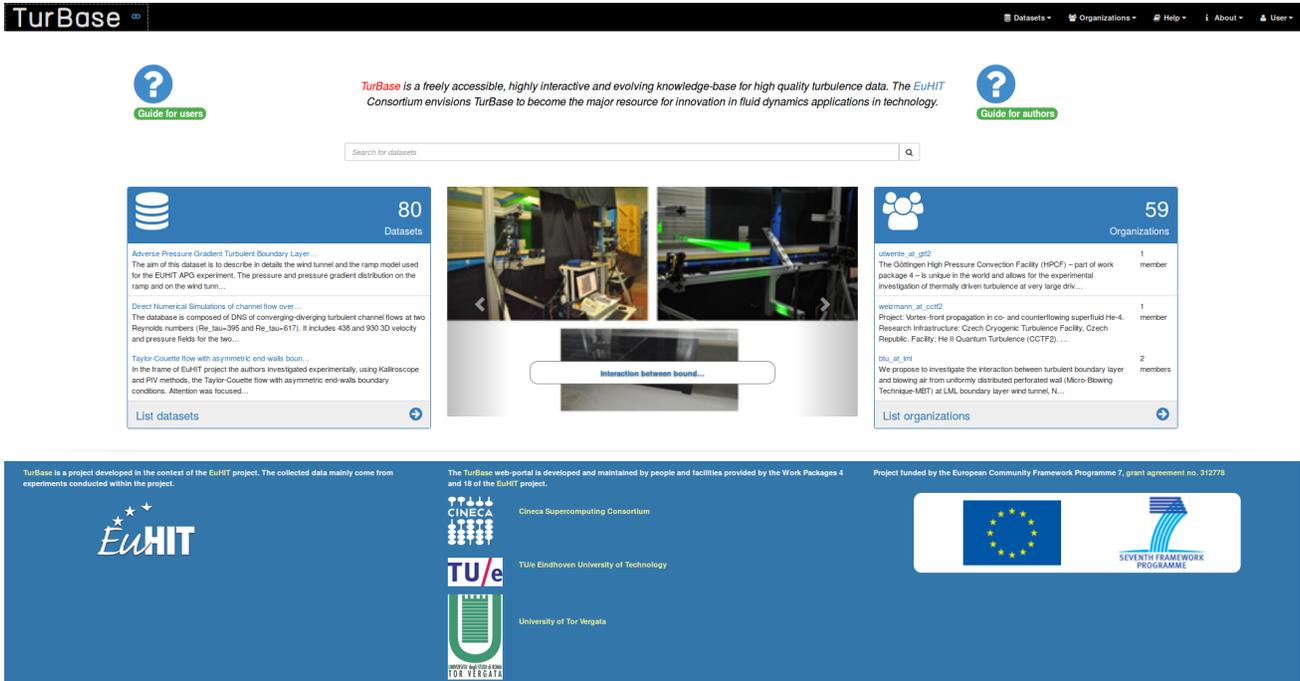


Fig. 3. TurBase Main Page

and add them to the queue, then the upload can be started via a button. The progress of the upload is monitored, but the web session needs to be alive in order to complete the operation.

While this can be very useful to see the file tree, create sub-folders and upload smaller files like supporting materials (pictures, scripts, ...), the direct upload is not suggested for bigger data files, because its performances are not optimal compared to the other option which is using the GlobusOnline service.

- Searching datasets and downloading files dataset search can be accomplished without registration. The interface allows one to select a free search with autocomplete activated and a structured search where the category tags are mainly used. In addition public/embargo selection can be specified as well as with/without online preview via notebook availability. The search is powered by the ElasticSearch engine since all the dataset metadata are always automatically replicated by the TurBase server. Concerning file exchange, normal users of TurBase Portal can use a version of the same interface for the upload but limited to the download of the the dataset they are interested in. The same considerations with respect to the network bandwidth utilization for the upload are still valid so to download huge amount of data, the usage of GlobusOnline service described before should be considered.
- API
TurBase provides its functionalities via a wide set of

RESTful APIs. The web application employs the popular JavaScript AngularJS framework to build the interface retrieving and posting data through the APIs. All the functionalities are also available directly via the APIs using the most convenient client, e.g. command line tools can be used to automatize tasks in scripts. The APIs are grouped in 8 categories according to their tasks:

- 1) Users and organizations
- 2) Datasets
- 3) Tags and categories
- 4) Hosting Sites
- 5) Metadata *schema*
- 6) Files
- 7) Reviews
- 8) Comments

and a proper documentation is available.

VIII. ONLINE INTERACTIVE ANALYSIS

For analyzing datasets and understanding how to use their data, we implemented the TurBase Service Portal, which allows users to operate interactively. It appears as an interactive web computer application, which creates a completely dedicated session for each user, based on Jupyter [14] notebooks and the tmpnb software [15]. It is capable of executing Python and Octave programs that can read datasets not under embargo: its peculiar feature is that it is a Matlab-like environment in which several tools of data analysis and visualization are already present. The underlying tmpnb software takes care of providing temporary Jupyter notebooks, with a private scratch area in which the user can write partial computations or files

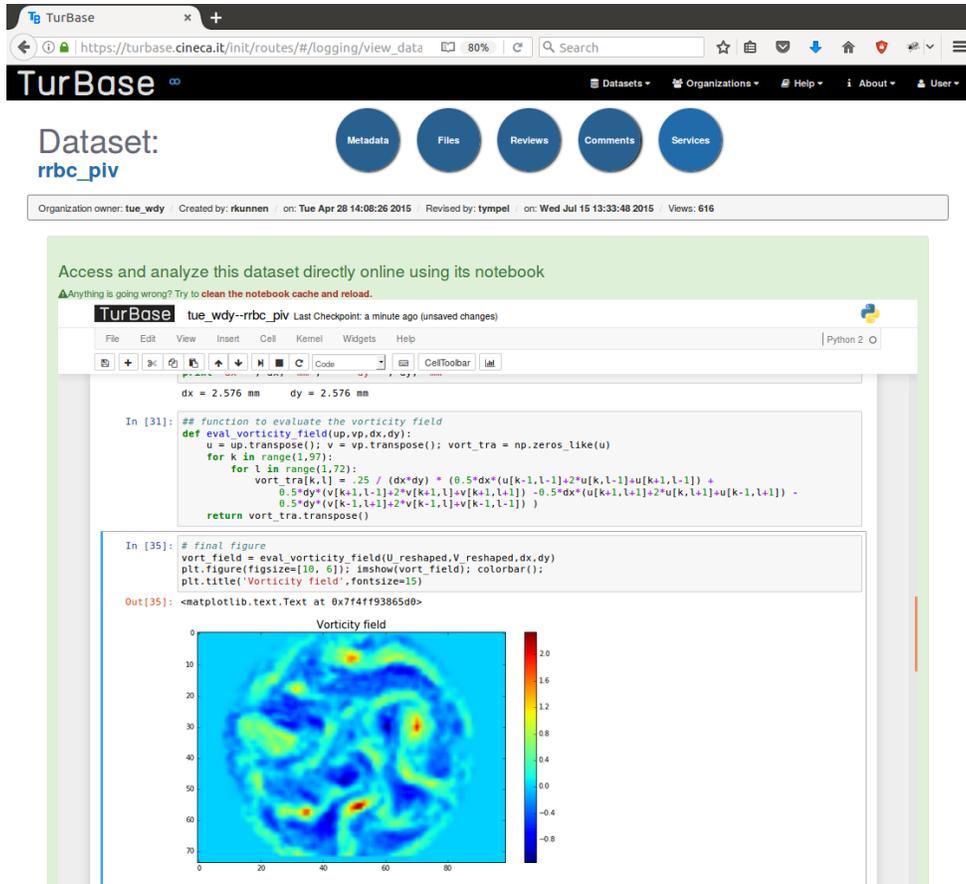


Fig. 4. Online computation on a dataset

that will be discarded when the web session expires. The user only needs a web browser and **no software** needs to be installed.

The datasets are transparently readable in the notebooks using their hosting site name, organization name, and dataset name, in a safe read-only way. To achieve better network performance while accessing the DLTD, the current implementation runs in a virtual machine near the DLTD.

Ideally, each dataset is accompanied by a short Jupyter iPython notebook that immediately explains how to derive relevant quantities from the experimental data. This tool allows one to have a quick understanding of the dataset.

The TurBase Service Portal is based on well known technologies in the Python world which are worldwide used by the scientific numerical community, with very detailed documentation: the users can do complex computations using the data in TurBase with the only limiting factor given by the hardware resources (currently 12 CPUs and 12 GB of ram) using the examples as starting point. With very little effort, it is possible to change the analysis with respect to the dataset its acting on or the mathematical algorithm.

In the iPython environment several tools are readily available:

- NumPy, Fast vector and matrix operations for linear algebra operations, sorting algorithm and selection criteria of data
- SciPy, FFT, ode solvers, signal processing primitives
- MatplotLib, the main plotting library for 2D and static 3D visualization
- Sympy, Symbolic calculus like derivation, integration, limits
- Pandas, to work with relational data, correlating quantities from different datasets/files
- HDF5 and netCDF4 packages, for interfacing with binary formats used in this scientific field
- OpenCV package, for advanced image processing capabilities

The online visualization capabilities are limited to 2D plots and static 3D perspective plots. Further work is ongoing on the Service Portal to achieve real-time 3D rendering.

IX. COMMUNITY FEEDBACK

A common feature nowadays is the possibility to receive feedback by the users on the quality and on the convenience of using datasets. TurBase implements a comment section which

collects the opinion of the users, which can assign a vote on a dataset contributing to its final ranking.xs

X. CONCLUSIONS AND FUTURE WORK

A software infrastructure for the research community working on turbulence and complex flows (**TurBase**) has been presented. TurBase is an easily accessible web platform for high quality data, offering scalable performances when accessing/uploading/searching data and providing at the same time maximum flexibility/maximum power when doing online computation directly on big datasets. The current implementation of TurBase was realized under the EuHIT project using only open source software and technologies. Some work to publish the source code and a guide for its installation and administration on GitHub.com is under consideration and will be reported elsewhere. Future work will include fine-tuning of the runtime parameters that affects the overall speed of TurBase and the addition of more packages in the Service Portal. Moreover, TurBase is customized for the EuHIT research infrastructure but the software product is flexible enough to be easily adopted by other research communities.

XI. ACKNOWLEDGMENTS

The authors of the present paper would like to acknowledge the financial support of the EUHIT project in the frame of its Transnational Access activity. EuHIT: "European High performance Infrastructures in Turbulence" is a Project coordinated by Max Plank Institute Gttingen and funded by the European Commission under Grant agreement Grant agreement no: 312778, in the frame of the Research Infrastructures Integrating Activity framework of FP7. We acknowledge the useful collaboration during the development of TurBase with S. Tympel, A. Holten and F. Tesser.

REFERENCES

- [1] EUHIT *European High performance Infrastructures in Turbulence*, <http://euhit.org>
- [2] JHTDB *The Johns Hopkins Turbulence Database*, <http://turbulence.pha.jhu.edu>
- [3] 4TU *4TU.Centre for Research Data*, <http://researchdata.4tu.nl>
- [4] CKAN *The open source data portal software*, <https://ckan.org>, The Open Knowledge Foundation
- [5] *Nesstar*, <http://www.nesstar.com>, NSD - Norwegian Centre for Research Data
- [6] The Dataverse Project, *Open source research data repository software*, <http://dataverse.org>
- [7] The Midas Platform, <http://www.midasplatform.org/> Kitware
- [8] Invenio *Invenio Digital Library Framework*, <http://invenio-software.org>
- [9] iRODS, *The Integrated Rule-Oriented Data System*, <https://irods.org>
- [10] GlobusOnline, <https://www.globus.org>
- [11] Web2py, *Free open source full-stack framework for web-based applications*, <http://www.web2py.com>
- [12] AngularJS, *Superheroic JavaScript MVW Framework*, <https://angularjs.org>
- [13] ElasticSearch, *A real-time distributed search and analytics engine*, <https://www.elastic.co/products/elasticsearch>
- [14] Project Jupyter, <http://jupyter.org>
- [15] tmpnb, *The temporary notebook service*, <https://github.com/jupyter/tmpnb>