COSMO: a Research Data Service Platform and Experiences from the BlueTides Project

Julian A. Uran
Pittsburgh Supercomputing Center, Carnegie Mellon University
Pittsburgh, Pennsylvania, United States

Wanting Huang*
Pittsburgh Supercomputing Center, Carnegie Mellon University
Pittsburgh, Pennsylvania, United States

Juan Puerto*
Pittsburgh Supercomputing Center, Carnegie Mellon University
Pittsburgh, Pennsylvania, United States

Nianyi Chen*
McWilliams Center for Cosmology, Physics Department, Carnegie Mellon University
Pittsburgh, Pennsylvania, United States

Mei-Yu Wang*
Pittsburgh Supercomputing Center, Carnegie Mellon University
Pittsburgh, Pennsylvania, United States

Yueying Ni
McWilliams Center for Cosmology, Physics Department, Carnegie Mellon University
Pittsburgh, Pennsylvania, United States

Tiziana Di Matteo
McWilliams Center for Cosmology, Physics Department, Carnegie Mellon University
Pittsburgh, Pennsylvania, United States

Paola A. Buitrago
Pittsburgh Supercomputing Center, Carnegie Mellon University
Pittsburgh, Pennsylvania, United States

ABSTRACT

We present details and experiences related to the COSMO project advanced by the Pittsburgh Supercomputing Center (PSC) and the McWilliams Center for Cosmology at Carnegie Mellon University for the BlueTides Simulation project. The design of COSMO focuses on expediting access to key information, minimizing data transfer, and offering an intuitive user interface and easy-to-use data-sharing tools. COSMO consists of a data-sharing web portal, API tools that enable quick data access and analysis for scientists, and a set of recommendations for scientific data sharing. The BlueTides simulation project, one of the most extensive cosmological hydrodynamic simulations ever performed, provides voluminous scientific data ideal for testing and validating COSMO. Successful experiences include COSMO enabling intuitive and efficient remote data access which resulted in a successful James Webb Telescope proposal to observe the first quasars in the first observing cycle.

CCS CONCEPTS

• **Applied computing** → **Astronomy**; • **Information systems** → **Wrappers (data mining)**; *Mediators and data integration*; *Enterprise search*; **Search interfaces**; **Query representation**.

*Authors contributed equally to this research.

PEARC '22, July 10–14, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9161-0/22/07.
https://doi.org/10.1145/3491418.3535166

KEYWORDS

Cosmology, Data sharing portal, Web portal, API, Cosmological hydrodynamic simulations, Open source, Data management, Data analysis, REST API, FastAPI

ACM Reference Format:
Julian A. Uran, Wanting Huang, Juan Puerto, Nianyi Chen, Mei-Yu Wang, Yueying Ni, Tiziana Di Matteo, and Paola A. Buitrago. 2022. COSMO: a Research Data Service Platform and Experiences from the BlueTides Project. In *Practice and Experience in Advanced Research Computing (PEARC '22), July 10–14, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3491418.3535166

1 INTRODUCTION

Data-intensive science requires scalable data management tools to distribute content from a curated data server, remotely share scientific simulation or experiment data with collaborators, enable analyses of hosted data by researchers (e.g., [6]), among others.

In response to the data management requirements of data-intensive research collaborations, the Pittsburgh Supercomputing Center (PSC) proposes the COSMO project. COSMO is a platform that allows researchers to easily integrate data access and management processes into their scientific workflows.

COSMO targets the sharing and accessibility challenges of large scale data management in order to enable scientific collaboration at scale. COSMO includes data sharing, data management, and data analysis tools that can be accessed through simple and flexible web portals and APIs. The current COSMO version includes file sharing and transferring tools such as Globus endpoints, a data-sharing portal, and a REST API. COSMO also includes a set of domain-agnostic recommendations for scientific data sharing. More details are given in Section 3.

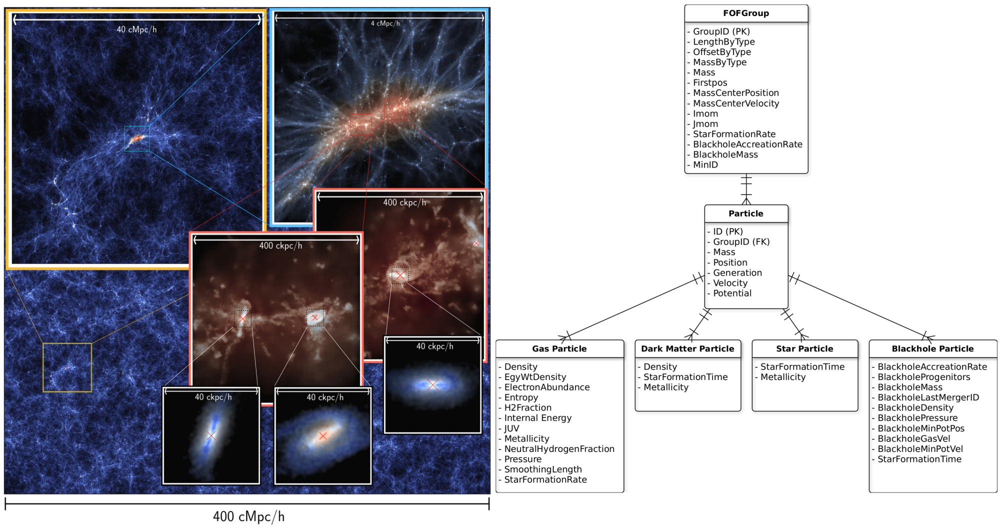


Figure 1: Left Panel: A zoomed-in visualization of the most massive halo ($10^{13} M_{\odot}$) at z = 6.5 in the BlueTides simulation. The image represents the wide dynamic range covered by BlueTides, showing the view of the whole 400 cMpc/h down to galaxy hosts of the most massive black holes on the scale of 40 ckpc/h. Right Panel: the UML diagram for the simulation FOF group catalogs.

Here, we present results from the collaboration with the BlueTides simulation project (PI: Tiziana Di Matteo) [3], which provides an excellent COSMO proof-of-concept. The BlueTides team has led the development of cosmological codes optimized for petascale NSF leadership high-performance facilities such as Blue Waters[1]. BlueTides aims to understand how supermassive black holes and galaxies formed in the first billion years of the universe's history (see left panel of Fig. 1). With close to one trillion particles simulated, BlueTides is the only simulation in the whole field of cosmology that can make direct predictions for the current and next generation of astronomical telescopes. The simulations dataset is of fundamental importance to the scientific community and the public access to it is crucial. The volume and data access requirements of BlueTides call for enhanced data management approaches providing an ideal test case for COSMO.

2 THE BLUETIDES SIMULATIONS

The BlueTides simulation is based on the massively-parallel cosmology simulator, MP-Gadget[2]. It implements a variety of sub-grid models for the physics governing galaxy and black hole formation and their feedback processes. Those include star formation with the effects of molecular hydrogen formation, energetic feedback from supernovae, super-massive black holes [2], and patchy reionization, that varies the optical depth based on local density [1] . The BlueTides simulation is one of the largest cosmological hydrodynamic simulations yet performed, enclosing a box withg the side of $400h^{-1}Mpc$ and a total of 0.7 trillion simulated particles.

The dataset files comprise the following three types of data:

- **Simulation snapshots:** The structure follows a modified version of the Gadget-2 format[3]. The file format is "BigFile"[4], that supports transparent file-level striping and substantially eases post-production data analysis for simulations at the BlueTides scale. The snapshot data is organized in blocks

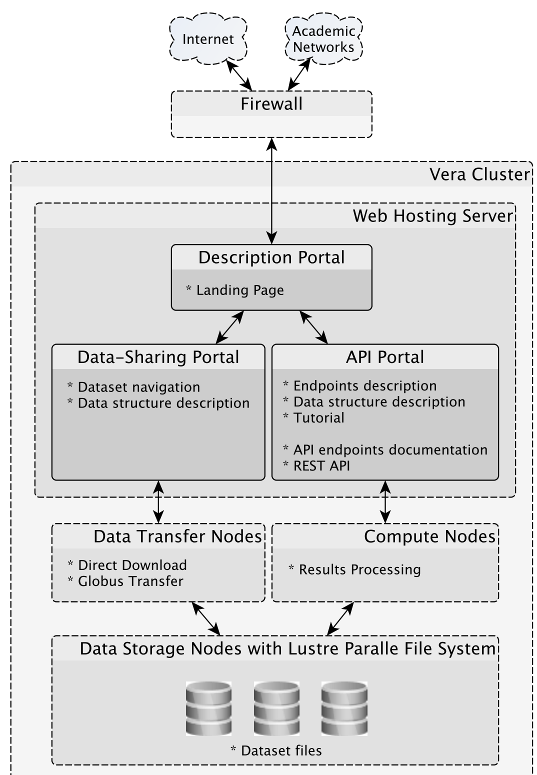


Figure 2: System Architecture of COSMO hosted by the Vera cluster.

[1]Blue Waters Supercomputer. https://bluewaters.ncsa.illinois.edu/
[2]Massively Parallel Cosmological SPH Simulation Software - MP-Gadget. https://github.com/bluetides-project/MP-Gadget
[3]GADGET-2: A code for cosmological simulations of structure formation. https://wwwmpa.mpa-garching.mpg.de/gadget
[4]BigFile: A reproducible massively parallel IO library for large, hierarchical datasets. http://github.com/rainwoodman/bigfile/

containing information about each type of particle – dark matter, gas, star, and black hole.

- **Friends-of-Friends (FOF) groups catalogs:** These catalogs contain properties about the gravitationally-bound particle groups (halos) identified by the FOF group finder algorithm, such as their total mass in terms of dark matter or gas component and the center-of-potential coordinates.
- **Particles in group (PIG) catalogs:** These catalogs consist of individual particle information enclosed in the halos, which includes the particle IDs, their properties such as mass and velocity, and the their associated Halo IDs. In the right panel of Fig. 1, we show a UML diagram of the catalog.

3 COSMO SYSTEM DESIGN AND IMPLEMENTATION

3.1 System Architecture

The COSMO deployment for serving the BlueTides dataset is hosted on the Vera cluster, operated by PSC on behalf of the McWilliams Center for Cosmology. The service is persistent through a non-ephemeral virtual machine.

In Fig. 2, we show a representation of the COSMO system architecture. The dotted and solid line boxes represent hardware and software elements, respectively. Arrows indicate data transfer channels. For illustration purposes, let's consider the following case. A researcher can first connect to the BlueTides description portal for general information about the dataset. They then browse through the *Data-Sharing Portal* and closely inspect the available files. Once a data download/query request is received, either the data-transfer optimized data transfer nodes (DTN), or the compute nodes are leveraged to get the data from the Lustre parallel file system on the Vera cluster.

The physical system hosts are dedicated and are designated for different tasks. Regarding the software stack, the system back-ends are written in Python, using the Django and FastAPI frameworks. COSMO also uses relational databases to maintain a record of the data skeleton. The portal utilizes Nginx as a reverse proxy and Bootstrap and JQuery are leveraged for the front-end development. All these components run on GNU/Linux with open source technologies. The code base is open source and is publicly available via GitHub[5]. More information including computational requirements, scope, and constraints, are available in the project website[6].

3.2 Web Portal

The BlueTides Web Portal (see Fig. 2) is designed to provide a high level view of key information prior to direct interaction with the data. It includes the *Description Web Portal*[7] (hosts landing page), the *Data-Sharing Portal* (includes *Data Structure* and the actual *Data-Sharing Portal* which provides access to key data meta-information and to individual data files via Globus endpoints), and the *API Portal*. The *API Portal* contains the *API Reference* (API endpoint descriptions) and *API Tutorial* (a tutorial for utilizing the API tools via Python scripts).

[5]COSMO GitHub Repository. https://github.com/pscedu/cosmo

[6]COSMO Project Website. https://www.cmu.edu/psc/aibd/cosmo.html

[7]Main/Description Web Portal: https://bluetides.psc.edu/

3.3 REST API

The goal of the REST API is to provide users with a convenient way to access key subsets of simulation data using the Vera cluster. Here, we adopt the developer-friendly FastAPI[8] framework for our implementation. The REST API was customized to navigate and expose the BlueTides data in a functional manner. Best practices for software development are followed with SOLID code and Test-Driven Development (TDD). All the API endpoints were tested and validated with unit tests.

```
GET /pig/
  {
    "LIST": [
    {
      "id": "230",
      "name": "PIG_230",
      "num_halos": 266771522,
      "time": 6.8500000034945225
    }, {...}]
  }
```

Figure 3: An example of "Read PIG" endpoint: `GET /pig/`

The API endpoint design follows the data structure of the BlueTides simulation data. For example, for particles-in-group catalogs, the canonical endpoints have a top-down structure of `/pig/{id}/{ptype}/{feature}/{group_id}`, where the query returns the data with selected feature `{feature}` from particle type `{ptype}` in group `{group_id}` from snapshot with the `{id}` identifier. The API endpoints are grouped based on functionality (see Fig 5) as follows:

- Particles-in-Group (`pig`): allows information queries of the Particles-in-Groups (PIG) catalogs.
- Length By Type (`lengthbytype`): retrieves the number of entities of each data category.
- Particles (`particle`): retrieves data for a specific particle type in PIG group catalogs.
- Advanced Queries (`advanced`): allows advanced search by specifying criteria for bulk queries.

The API first provides a list of valid identifiers to be queried, which return JSON structures and numpy arrays for more processing or for initiating downloads of specific files after switching to the data-sharing portal. For example, to retrieve information from a PIG catalog, we first make a query to read the `pig` endpoint (an example is shown in Fig. 3). This returns a response with a list containing information including the PIG ID. Next, those values are used as inputs for the "Read Snapshot FoF Info" endpoint, as shown in Fig. 4, which generates responses with the specified data.

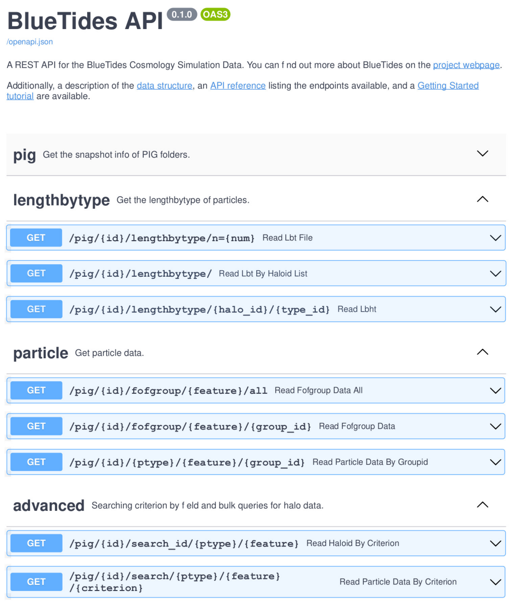
```
GET /pig/230/fofgroup
  {
    "id": 230,
    "fof_subdirs": [
      "Imom",
      "LengthByType",
      ...
      "MassByType",
      "MassCenterPosition" ]
  }
```

Figure 4: An example of "Read FoF Info" endpoint: `GET /pig/230/fofgroup/`

Advanced queries, such as the endpoint

`GET /pig/{id}/search/{ptype}/{feature}/{criterion}/?{range_query}`

[8]FastAPI framework. https://fastapi.tiangolo.com/




```python
import matplotlib.pyplot as plt
import requests
import json
import numpy as np

base = "https://bluetides-api.psc.edu/"
response = requests.get(base, verify=False)


def get_group_data(group_url):
    response = requests.get(group_url, verify=False)
    key = list(response.json().keys())[0]
    response_arr = np.array(json.loads(response.json()[key]))
    return response_arr


pig_url = base + "pig/"
ptype = ['dm', 'gas', 'star', 'bh']
ctype = ['slategray', 'royalblue', 'khaki', 'firebrick']
fig, ax = plt.subplots(figsize=(8, 6))
for c, p in zip(ctype, ptype):
    pos = get_group_data(pig_url + '251/' + p + '/Position/1')
    if p == 'bh':
        ax.scatter(pos[:, 0], pos[:, 1], s=80, marker='.', label=p, color=c)
    else:
        ax.scatter(pos[:, 0], pos[:, 1], s=0.01, marker='.', label=p, color=c)
plt.title('Particles in Snapshot 251 Halo 1')
leg = ax.legend(fontsize=12)
for ll in leg.legendHandles:
    ll.sizes = [50]
plt.show()
```

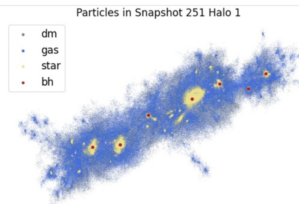


Figure 5: Left Panel: The API web interface for the PIG group catalog. The Upper-Right Panel: A Python code snippet utilizing the API to download the particle and plot the particle spatial distribution in one simulated halo. The Lower-Right Panel: the plot generated using the code snippet above.

enables the user to search for groups meeting a specific criterion (e.g., within a range of total dark matter mass). We note that regular queries work out of the box from the COSMO virtual machine, while more complex queries may launch Slurm jobs with parallel workers to ensure a fast response.

We provide both an interactive web interface[9] (see left panel in Fig. 5), and a programmatic language-agnostic API interface that can be used with any modern preferred language. A tutorial for Python can be found on the website[10]. Fig. 5 illustrates a query for the spatial coordinates of various simulated particles (dark matter, gas, star, and black hole) using the Python code snippet (in the upper-right panel), and a generated plot of their spatial distribution in one halo.

4 COSMO DATA SHARING RECOMMENDATIONS

From the experiences with the BlueTides project, we collected a set of recommendations for scientific data sharing. For additional information, discussion, and recommendations, please refer to the project website[11].

- **Enabling a browsable dataset:** In this work, this function is manifested by the *Data-Sharing Portal*, which contains data structure information with different levels of granularity.
- **Allowing the dataset to be downloadable by key segments:** The data index metadata (MBs) can be downloaded first for identifying sections of interest before downloading the main content (GBs).
- **Sharing the data files using different data transfer protocols:** Anonymous Globus endpoints support direct downloads and regular Globus collection allow automatic transfers for anyone with a Globus Connect Personal account [12].
- **Providing detailed dataset descriptors:** By sharing description of the data structures as well as the relationship between data fields, the users are equipped to access the data efficiently.
- **Supporting data querying via a REST API:** This approach allows researchers to browse and interact with the data via a commonly-used programmatic protocol. COSMO includes data endpoints, instructions, tutorials, and a website that allows directly querying the API from the browser.

5 DISCUSSION AND CONCLUSION

The BlueTides team was able to share their data with collaborators at the University of Melbourne via COSMO. The simulations were utilized to make predictions for what the James Webb Telescope [4] will see as the host galaxies of the first quasars. The utilization of the COSMO platform for remote data access resulted in a successful James Webb Telescope proposal to observe the first quasars in the first observing cycle and several related peer-review journal publications (e.g., [5]). These works are also featured in press-releases[13].

This paper presents the COSMO project, which delivers a proof-of-concept for broadly sharing specialized datasets across the scientific community. We demonstrate that leveraging COSMO and

[9] BlueTides API Access. https://bluetides-api.psc.edu/
[10] BlueTides API Tutorial. https://bluetides.psc.edu/tutorial/
[11] COSMO Project Website. https://www.cmu.edu/psc/aibd/cosmo.html
[12] Globus Connect Personal https://www.globus.org/globus-connect-personal
[13] NASA (2020 October 14) Simulations Show Webb Telescope Can Reveal Distant Galaxies Hidden in Quasars' Glare [Press release]. https://www.nasa.gov/feature/goddard/2020/simulations-show-webb-telescope-can-reveal-distant-galaxies-hidden-in-quasars-glare

allowing researchers to access information using different protocols significantly improves the scientific collaboration at scale.

COSMO tackles the issues of analyzing voluminous data with a hybrid approach of combining API tool development with direct data access. It has successfully provided easy-to-use tools for non-experts to explore the data in a language-agnostic way while allowing users to explore the full dataset when ready. Future plans include extending the API endpoints to support more data features and integrating with other frameworks such as CKAN[14], Globus Action Providers, or Globus Flows. Applying the successful experiences with the BlueTides simulations dataset to applications in other scientific domains is also of interest. We open-source our implementation and welcome contributions.

ACKNOWLEDGMENTS

We thank Derek Simmel for providing Vera cluster support and Globus Endpoints configuration, Sergiu Sanielevici for valuable comments, Erica Anderson and David Kapcin for supporting grant budgeting and management, and the PSC systems team for overall technical support. This project was sponsored by the McWilliams Center for Cosmology Seed Grant Program. It leveraged Bridges (NSF-1445606), Bridges-2 (NSF-1928147), and the Vera cluster.

REFERENCES

[1] N. Battaglia, H. Trac, R. Cen, and A. Loeb. 2013. Reionization on Large Scales. I. A Parametric Model Constructed from Radiation-hydrodynamic Simulations. *Astrophysical Journal* 776, 2, Article 81 (Oct. 2013), 81 pages. https://doi.org/10.1088/0004-637X/776/2/81

[2] Tiziana Di Matteo, Volker Springel, and Lars Hernquist. 2005. Energy input from quasars regulates the growth and activity of black holes and their host galaxies. *Nature* 433, 7026 (Feb. 2005), 604–607. https://doi.org/10.1038/nature03335

[3] Yu Feng, Tiziana Di-Matteo, Rupert A. Croft, Simeon Bird, Nicholas Battaglia, and Stephen Wilkins. 2016. The BlueTides simulation: first galaxies and reionization. *Monthly Notices of the Royal Astronomical Society* 455, 3 (Jan. 2016), 2778–2791.

[4] Jonathan P. Gardner and et al. 2006. The James Webb Space Telescope. *Space Science Reviews* 123, 4 (April 2006), 485–606.

[5] Madeline A Marshall, Stephen Wilkins, Tiziana Di Matteo, William J Roper, Aswin P Vijayan, Yueying Ni, Yu Feng, and Rupert A C Croft. 2022. The impact of dust on the sizes of galaxies in the Epoch of Reionization. *Monthly Notices of the Royal Astronomical Society* 511, 4 (02 2022), 5475–5491.

[6] Evgeny Toropov, Paola A. Buitrago, and José M. F. Moura. 2019. Shuffler: A Large Scale Data Management Tool for Machine Learning in Computer Vision. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning)* (Chicago, IL, USA) *(PEARC '19)*. Association for Computing Machinery, New York, NY, USA, Article 23, 8 pages. https://doi.org/10.1145/3332186.3333046

[14]CKAN: The Open Source Data Portal Software. https://ckan.org/