The Research Repository for Data and Diagnostics (R2D2): An Online Database Software System for High Performance Computing and Cloud-based Satellite Data Assimilation Workflows

Eric J. Lingerfelt[1], Tariq J. Hamzey[2], Maryam Abdi-Oskouei, Jérôme Barré, Fábio Diniz, Clémentine Gas, Ashley Griffin, Dominikus Heinzeller, Stephen Herbener, Evan Parker, Benjamin Ruston, Christian Sampson, Travis Sluka, Kristin Smith, Yannick Trémolet[1]

[1] University Corporation for Atmospheric Research, Boulder CO 80307, USA
[2] Science Systems and Applications, Inc., Lanham MD 20706, USA
eric2@ucar.edu
tariq.hamzey@nasa.gov

Abstract. The Joint Center for Satellite Data Assimilation (JCSDA) is a multi-agency research center established to improve the quantitative use of satellite data in atmosphere, ocean, climate and environmental analysis and prediction systems. At the JCSDA, scientists and software engineers within the Joint Effort for Data Assimilation Integration (JEDI) are developing a unified data assimilation framework for research and operational use. To harness the full potential of ever-increasing volumes of data from new and evolving Earth observation systems, a new online database software system has been developed and deployed by the JCSDA. As a core component of JEDI, the Research Repository for Data and Diagnostics (R2D2) performs data registration, management, and configuration services for data assimilation computational workflows. We present an overview of R2D2's distributed system of data stores, SQL data model, intuitive python API, and user support efforts. In addition, we will detail R2D2's utilization by environmental prediction applications developed by the JCSDA and its partners.

Keywords: earth system data assimilation, fair principles, earth observations, sql data model, skylab, joint effort for data assimilation integration, jedi, high performance computing, data store, r2d2, research repository for data and diagnostics, ioda, interface for observation data access, saas.

1 Introduction

Computational data assimilation (DA) in numerical weather prediction is the process of combining observational data with model simulations to estimate the current state of the atmosphere, ocean, and climate. In pursuit of higher accuracy and finer resolutions, DA models rely on ever-increasing volumes of observational data, particularly from new satellites [1]. A robust database system is thus crucial for efficient storage and

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_34

retrieval of observations, both historical and in real-time. Managing model input and output is equally crucial for accessing past experiments and reanalysis of earlier time periods. Environmental modeling centers spend decades developing in-house database solutions for operational use and remote data retrieval. The European Centre for Medium-Range Weather Forecasts (ECMWF) [2] conceived the Meteorological Archival Retrieval System (MARS) in 1985, which currently manages petabytes of data [3]. In 2001, the National Centers for Environmental Prediction (NCEP) [4] developed the National Operational Model Archive and Distribution System (NOMADS) [5] based on the Open-Source Project for a Network Data Access Protocol (OPeNDAP) [6]. Together with its partner organizations, the Joint Center for Satellite Data Assimilation (JCSDA) [7] has developed a completely original software system called the Research Repository for Data and Diagnostics (R2D2) that performs data management, registration, and configuration services for DA workflows. As a central component of the Joint Effort for Data assimilation Integration (JEDI) project and its flagship SkyLab application [8, 9], R2D2 satisfies these requirements by offering a centralized, model agnostic platform for handling diverse datasets. Moving toward a unified DA framework, R2D2 provides a generic data solution for DA scientists and is not restricted to any one organization or data storage platform. In the past two years, R2D2 has been developed into a highly sophisticated data management system with internal knowledge of numerous high performance computing (HPC) systems throughout the United States, all of which are interconnected by a centralized cloud database. This paper details R2D2's architecture and data holdings, its utilization across partner organizations, and its near-term development objectives that are bringing JEDI's premise of a unified DA framework to fruition.

2 Distributed Data Access

Accessible from any place at any time, R2D2 relieves data assimilation scientists from the technical hurdles of data ingestion and restoration by linking the physical locations of data via a network of interconnected, mirrored data hubs established at several high performance computing centers, including the NASA Center for Climate Simulation [10], NCAR-Wyoming Supercomputing Center [11], Space Science and Engineering Center [12], and Mississippi State University's High Performance Computing Collaboratory [13]. Cloud-based data storage services provided by Amazon Web Services' Simple Storage Service (AWS S3) [14], Microsoft Azure's Blob Storage [15], and the Google Cloud [16] are also utilized as R2D2 data hubs. Each data hub is a collection of R2D2 data stores where each is physically instantiated as a posix directory or a cloud object storage container such as an AWS bucket (see Fig. 1). The data stores are populated by a subdirectory tree partitioned by the type of data file stored and the valid date for the file. In effect, these data type and date subdirectories are used as hash buckets for consumed data files indexed by the R2D2 service. Permissions for read and write operations to these data stores are managed in three ways: (1) Data stores are identified as requiring administrative write permission by R2D2, (2) R2D2 users are identified as

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_34

either regular or administrative users where administrative users are granted modification access to all data stores, (3) Administrative permissions are set via role accounts at high performance computing centers and authorization policies executed on cloud platforms. R2D2 data stores are characterized by the level of data protection and the total size of the data. “Experiments” data stores provide data file storage for all R2D2 users and are made available on all data hubs. “Archive” data stores require administrative write permission and provide all the input files needed for a variety of JSCDA workflows. Mirrored across all data hubs, “archives” supply a variety of input data types including observation and bias correction files as well as deterministic (i.e., non-ensemble) forecast and analysis files. “Ensemble” data stores are like “archives”, but they are not mirrored due to their large disk footprint which can range from 10 to 100TB. These data stores contain model-specific, ensemble forecast input files and currently make up over 90 percent of R2D2’s total data holdings. Computing resources must be registered in R2D2 to gain access to the collection of R2D2's data hubs. All compute resources are assigned one local data hub and one or more cloud-based data hubs.

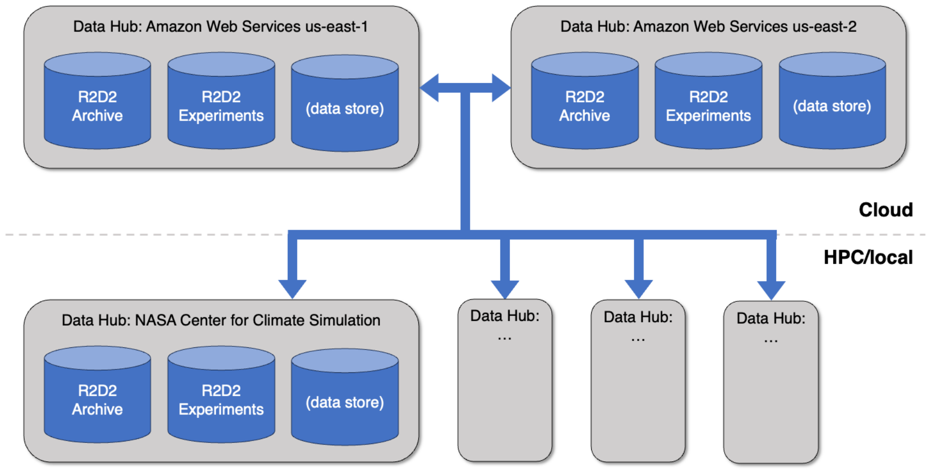

Fig. 1. A schematic representation of R2D2's data hubs and data stores.

3 SQL Design and Implementation

R2D2 persists knowledge through an SQL relational database designed to serve as an information indexer that stores data file attributes along with metadata concerning experiments, data vocabularies, data hubs, and supported compute host configurations. R2D2’s SQL schema represents related but unique and non-overlapping categorizations of metadata and data files. To optimize internal efficiency and logical consistency, R2D2’s MySQL database utilizes three design components: normalization, orthogonality, and extensibility. R2D2 tables are normalized to the third normal form (3NF) [17]. Orthogonality is ensured with the distinction of R2D2 *data*, *index*, and *register* tables,

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_34

where *data* and *index* types are called *items*. A *data item* encapsulates all unique identifiers for the physical data files stored in the system. An *index item* is not associated with a data file; it stores metadata records only. *Register* tables link *data* and *index items* to physical infrastructure by logging one-to-many relationships between data stores and data files and between compute hosts and data hubs. All tables are granted a unique index by the MySQL server, and all *index* items have human-readable, unique *name* values. Lastly, R2D2 promotes extensibility by dynamically extracting the SQL schema at runtime, enabling the Python-based abstraction layer to be completely *item*-agnostic. Hence, extensions to the database design are automatically reflected at runtime and do not require source code updates.

4 Software Architecture

Secure access to this centralized database server and R2D2's end user functionality is currently enabled through an intuitive, cross-platform Python API (see Fig. 2). At the top of R2D2's object-oriented class hierarchy is R2D2Object whose children include R2D2Item and R2D2Register. R2D2Items are R2D2Objects that can be stored in the database, and R2D2Registers are R2D2Objects that establish one-to-many relationships between R2D2Items. As reflected in R2D2's database schema, R2D2Item subclasses come in two flavors: *data* and *index*. In addition, the R2D2Register class is currently instantiated as two subclasses, R2D2DataRegister and R2D2ComputeHostRegister, which log relationships between data stores to data files and compute hosts to data hubs, respectively. R2D2's software layer also includes a set of utility packages. The *mysql* package contains classes for securely connecting to R2D2's cloud-based MySQL server, executing queries against the database, and parsing the database schema for instance population. The *error* package provides a suite of custom exception handling classes specific to R2D2's functionality. The *util* package furnishes administrative functions such as data store synchronization, date formatting methods, logging, and local and cloud-based file manipulation and transfer. Finally, the *test* package utilizes the cmake utility [18] to robustly stress the system.

5 Adherence to FAIR

State-of-the-art numerical weather prediction models are impacted by Big Data issues brought on by an increasing amount of incoming observational data, an explosion of the number of points as the model grid shrinks, and an increase in the frequency of forecast cycling. FAIR principles [19] "emphasize machine-actionability because humans increasingly rely on computational support to deal with data as a result of the increase in volume, complexity, and creation speed of data" [20]. R2D2 follows these guidelines in several ways. The system enhances "findability" of data by assigning metadata (and data) a globally unique and persistent identifier from a searchable resource. Identifiers utilized by R2D2 are minted by the MySQL server. It enables "accessibility" by making metadata (and data) retrievable by their identifier using a stand-

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_34

ardized, authenticated communications protocol that is open, free, and universally implementable. Another JCSDA-supported software product, the Interface for Observation Data Access (IODA) [21], enables "interoperability" and "reusability". By implementing multiple data storage backends such as Network Common Data Form (netCDF) [22], Hierarchical Data Format (HDF5) [23], and Observation DataBase (ODB) [24], IODA allows observational data to seamlessly integrate with other data, applications, and workflows while being well-described using standard vocabularies for data replication and combination.

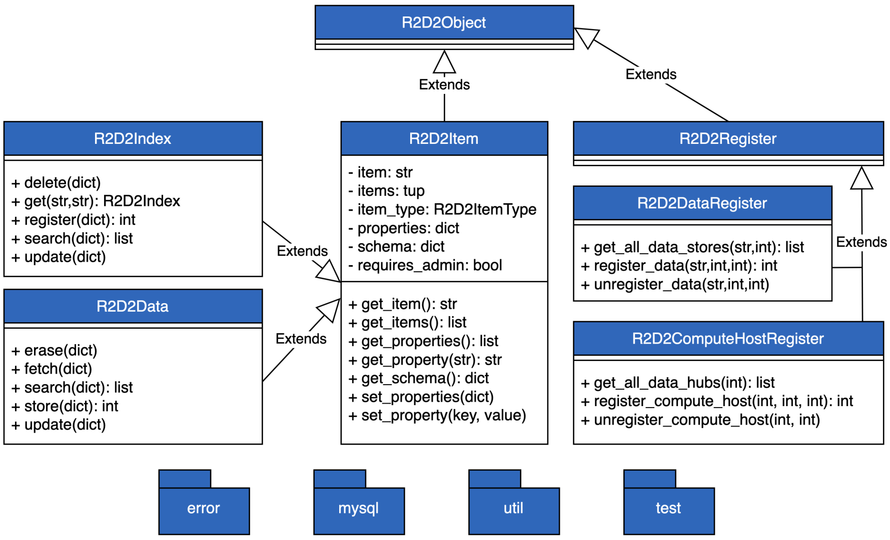

Fig. 2. A UML diagram of R2D2's core python API.

6 End User Support

Documentation provides a key interface between users and developers of a system [25]. Good documentation and user support encourages system adoption, increases end-user satisfaction with the product, and avoids time-intensive troubleshooting that can snowball into project delays and emergency support to partner agencies or other users [26]. For R2D2, three documents have proven key for ease of use (and lack of troubleshooting requests to the R2D2 team): tutorial, key reference tables, and the FAQ. The tutorial documentation covers system prerequisites and configurations, installation instructions, and step-by-step examples of how to use R2D2. Code snippets at each step ensure that the instructions are clear and easy to follow. Key lookup was formerly one of the main causes of confusion and support requests in R2D2. This problem was resolved by a new document containing lookup tables of required keys, key types, possible values for keys with an enumerated set of possible values, and which tables can only be changed by an R2D2 administrator. Use-instructions accompanying this document include code snippets and step-by-step examples to ensure clear communication. The FAQ document

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_34

addresses the most frequent questions from end users, including inquiries targeted at data search and ingestion tasks. Collating assorted requests into one publicly accessible document has cut down on the time spent troubleshooting by the R2D2 team and made it easy for users to find answers to most of their questions, increasing ease of use.

7 Utilization

The JCSDA has developed a comprehensive workflow called SkyLab to facilitate the execution of diverse data assimilation scenarios. These scenarios can involve different instruments, algorithm choices such as background error covariance or cost function, and different models. The SkyLab workflow serves as the orchestrator, driving the underlying JEDI code to run these different experiments. Models currently supported in SkyLab are the Unified Forecast System (UFS) [27], the Model for Prediction Across Scales (MPAS) [28], and the Goddard Earth Observing System - Composition Forecasting (GEOS-CF) system [29], and work is in progress to integrate the Modular Ocean Model (MOM6) [30] and the GEOS and UFS Marine and Aerosols components. Existing workflows, called *suites*, range from simple HofX [31] calculations to more complex, fully cycling ensemble and deterministic data assimilation and forecasts. The individual tasks composing these *suites* involve data file handling using R2D2, execution of JEDI binaries, or other miscellaneous tasks such as extracting performance information from logs or creating visualization artifacts. R2D2 supports SkyLab by providing storage and access to data, and through experiment registration, tracking, and configuration. R2D2 supplies JCSDA workflows with the default runtime arguments for the scalable execution of JEDI binaries and other processing tasks using common job schedulers such as SLURM [32] or PBS [33]. All experiments registered with R2D2 are granted a globally unique identifier consisting of four alphanumeric characters and the name of the supervising research or operational center. This identification scheme allows centers to track, categorize, and interrelate results associated with each experiment. In addition to the JCSDA, NASA's Global Modeling and Assimilation Office (GMAO) [34], a JEDI partner, is also leveraging R2D2 in new data assimilation toolkits to handle experimental data storage. Its current data holdings include approximately 12K files comprising 250 GB of bias correction, forecast, and observational data. As new toolkits are developed and tested using R2D2's database solution, the GMAO is continually evaluating how best to expand R2D2's scope for both experimental and production workflows. Since its initial release in early 2023, R2D2 has registered over 9K experiments and stored 17M data files distributed across 9 data hubs and 40 data stores for 50 end user accounts. The results of a single computational cycle of a production JEDI-UFS experiment managed in R2D2 requires storage and tracking of approximately 50K files and 30 TB of data.

8 Conclusion

The Research Repository for Data and Diagnostics database software system attempts to address the requirements of Big Data management, access, and curation for high

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_34

performance computational data assimilation experiments. Its multi-tier system comprises a series of distributed data hubs interconnected and indexed via an extendable SQL database schema and a dynamic, intuitive python API. R2D2 adheres to FAIR principles by deploying a system managing "findable" and "accessible" data files and artifacts. Multiple user support products, such as tutorials, reference tables, API examples, and a FAQ, have been created to provide in-depth instruction and easy adoption by new users. Organizations such as the JCSDA and NASA's GMAO are currently applying the R2D2 system for world-class modeling and data assimilation. Current development paths include a redesign of R2D2 into a scalable, flexible client/server system by utilizing the OpenAPI 3.0 Specification [35] for RESTful web services [36], Swagger tooling [37], and the Flask micro web framework [38].

Acknowledgments. The Joint Center for Satellite Data Assimilation is a partnership between NOAA, NASA, the US Navy, and the US Air Force, committed to improving and accelerating the quantitative use of research and operational satellite data in weather, ocean, climate and environmental analysis and prediction systems. University Corporation for Atmospheric Research (UCAR) is hosting the core team of our multi-agency research center.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. McCarty, W., Coy, L., Gelaro, R., Huang, A., Merkova, D., Smith., E., Sienkiewicz, M., and Wargan, K: MERRA-2 Input Observations: Summary and Assessment. In: Koster, R. (ed.) NASA Technical Report Series on Global Modeling and Data Assimilation, vol. 46 (2016)
2. ECMWF Homepage, https://www.ecmwf.int/
3. Raoult, B.: Architecture of the new MARS server. In: Sixth Workshop on Meteorological Operational Systems, Reading, United Kingdom (1997)
4. NCEP Homepage, https://www.weather.gov/ncep/
5. Rutledge G.K., Alpert J., Ebisuzaki W.: NOMADS: a climate and weather model archive at the National Oceanic and Atmospheric Administration. In: Bulletin of the American Meteorological Society **87**(3) 327-341 (2006)
6. OPeNDAP Homepage, https://www.opendap.org/
7. JCSDA Homepage, https://www.jcsda.org, last accessed 2024/02/16
8. Heinzeller, D., Abdi-Oskouei, M., Herbener, S., Lingerfelt, E., Trémolet, Y., and Aligné, T.: The Joint Effort for Data assimilation Integration (JEDI): A unified data assimilation framework for Earth system prediction supported by NOAA, NASA, U.S. Navy, U.S. Air Force, and UK Met Office. In: EGU General Assembly 2023, EGU23-10697, Vienna, Austria (2023)
9. Herbener, S., Heinzeller, D., Griffin, A., Lingerfelt, E., Parker, E., Trémolet, Y., Auligne, T.: Accelerating Research to Operations with the Joint Effort for Data Assimilation Integration (JEDI) and Skylab Systems. In: 104th AMS Annual Meeting, Baltimore, MD (2024)
10. NCCS Homepage, https://www.nccs.nasa.gov, last accessed 2024/02/16
11. NWSC Homepage, https://www2.cisl.ucar.edu/ncar-wyoming-supercomputing-center, last accessed 2024/02/16
12. SSEC Homepage, https://www.ssec.wisc.edu, last accessed 2024/02/16

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_34

13. MSU-HPC2 Homepage, https://www.hpc.msstate.edu, last accessed 2024/02/16
14. AWS S3 Homepage, https://aws.amazon.com/s3, last accessed 2024/02/16
15. Microsoft Azure Blob Homepage, https://azure.microsoft.com/en-us/products/storage/blobs, last accessed 2024/02/16
16. Google Cloud Storage Homepage, https://cloud.google.com/storage, last accessed 2024/02/16
17. Codd, E.F.: Further Normalization of the Data Base Relational Model. In: Research Report / RJ / IBM, **RJ909** (1971)
18. Wilke, J., Arndt, D., Lebrun-Grandie, D., Madsen, J., Ellingwood, N., Trott, C.: Modern problems require modern solutions: How modern CMake supports modern C++ in Kokkos. In: International Workshop on Performance, Portability & Productivity in HPC (2020)
19. Stall, S., Yarmey, L., Cutcher-Gershenfeld, J., Hanson, B., Lehnert, K., Nosek, B., Parsons, M., Robinson, E., and Wyborn, L.: Make scientific data FAIR. Nature **570**, 27-29 (2019)
20. FAIR Principles, https://www.go-fair.org/fair-principles, last accessed 2024/02/16
21. Kim, J., Teng, Y-C., Vernieres, G., Sluka, T., Paturi, S., Hao, Y., Worthen, D., Li, B., Wang, J., Zhu, J., Kim, H-C., Kleist, D.: NOAA-NCEP Next Generation Global Ocean Data Assimilation System (NG-GODAS). Research activities in Earth system modelling. Working Group on Numerical Experimentation. Report 51: 8-05 (2021)
22. Rew, R., and Davis, G.: NetCDF: an interface for scientific data access. IEEE Computer Graphics and Applications **10**(4) 76-82 (1990)
23. Folk, M., Heber, G., Koziol, Q., Pourmal, E., Robinson, D.: An overview of the HDF5 technology suite and its applications. In: AD 2011, Proceedings of the EDBT/ICDT 2011 workshop on array databases, pp. 36-47 (2011)
24. Fouilloux, A.: ODB (Observational DataBase) and its usage at ECMWF. In: Twelfth Workshop on Meteorological Operational Systems (2009)
25. Silva, L., Unterkalmsteiner, M., and Wnuk, K.: Towards identifying and minimizing customer-facing documentation debt. In: 2023 ACM/IEEE International Conference on Technical Debt (TechDebt), pp. 72-81 Melbourne, Australia (2023)
26. Burnett, J., Dale, R., Hou, C-Y., Palomo-Munoz, G., Whitney, K., Aulenbach, S., Bristol, R., Valle, D., Wellman, T.: Ten simple rules for creating a scientific web application. PLoS Comput Biol **17**(12): e1009574. (2021)
27. UFS Homepage, https://epic.noaa.gov/get-code/ufs-weather-model, last accessed 2024/02/16
28. MPAS Homepage, https://www.mmm.ucar.edu/models/mpas, last accessed 2024/02/16
29. GMAO Weather Analysis & Prediction, https://gmao.gsfc.nasa.gov/weather_prediction, last accessed 2024/02/16
30. MOM6 Homepage, https://www.gfdl.noaa.gov/mom-ocean-model, last accessed 2024/02/16
31. HofX applications in OOPS, https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/inside/jedi-components/oops/applications/hofx.html, last accessed 2024/02/16
32. Jette, M. and Wickberg, T.: Architecture of the Slurm Workload Manager. In: Job Scheduling Strategies for Parallel Processing. Springer Nature Switzerland, (2023)
33. Jones, J.: PBS: Portable Batch System. In: Sterling, T. (ed) Beowulf Cluster Computing with Linux, MIT Press (2001)
34. GMAO Homepage, https://gmao.gsfc.nasa.gov, last accessed 2024/02/16
35. OpenAPI Specification Homepage, https://swagger.io/resources/open-api/
36. What is a RESTful API? Homepage, https://aws.amazon.com/what-is/restful-api/
37. Swagger Tools Homepage, https://swagger.io/tools/
38. Flask Homepage, https://flask.palletsprojects.com/en/3.0.x/

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_34