The Data Analytics Framework for XDMoD

Aaron Weeden[1], Joseph P. White[1], Robert L. DeLeon[1], Ryan Rathsam[1], Nikolay A. Simakov[1], Conner Saeli[1], Thomas R. Furlani[2]

[1] Center for Computational Research, University at Buffalo, Buffalo, NY

[2] Roswell Park Comprehensive Cancer Center, Buffalo, NY

Metrics2023

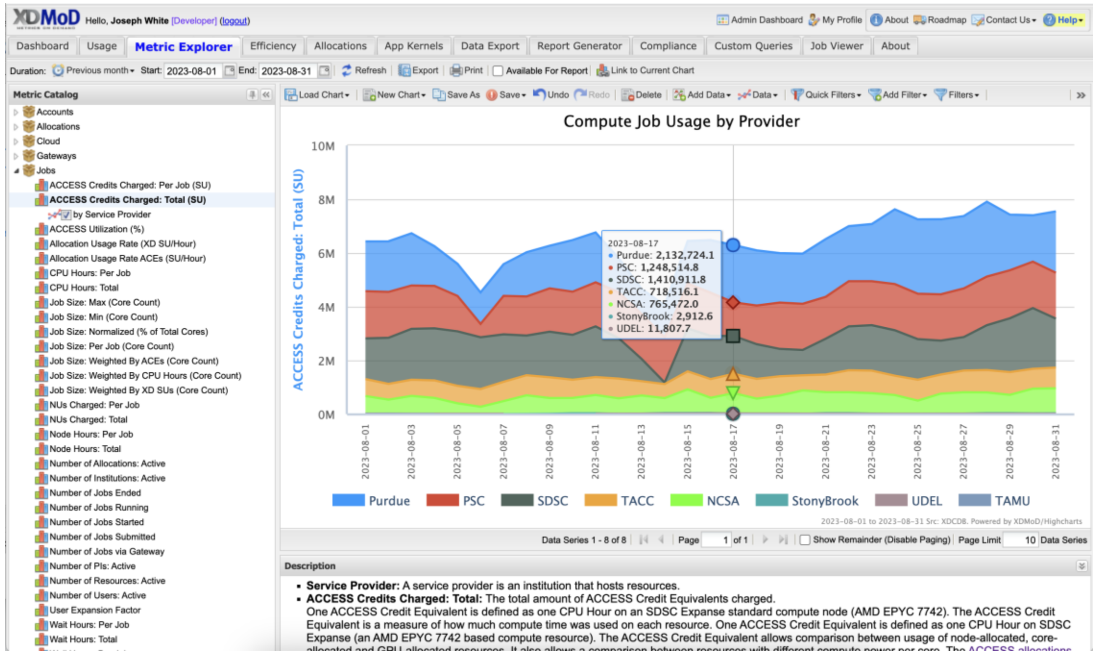
NSF Award: OCI 2137603

Introduction

- The US National Science Foundation (NSF) invests in a cyberinfrastructure (CI) ecosystem
- A major component of this is the **A**dvanced **C**yberinfrastructure **C**oordination **E**cosystem: **S**ervices & **S**upport (ACCESS) program
- Comprehensive instrumentation, monitoring, measurement, and reporting of ACCESS are essential
- The ACCESS Metrics team provides this service for ACCESS and other NSF programs, e.g., Campus Cyberinfrastructure (CC*) and Cyberinfrastructure for Sustained Scientific Innovation (CSSI)
- Extension of successful Technology Audit Service (TAS) and XD Metrics Service (XMS) programs that monitored NSF E**x**treme **S**cience and **E**ngineering **D**iscovery **E**nvironment (XSEDE)

Open XDMoD

- Software developed and used by ACCESS Metrics team
- Web-based portal for data exploration, visualization, and export
- Role-based views for various CI stakeholders
- ACCESS XDMoD has historical usage data from NSF TeraGrid, XSEDE, and ACCESS programs
- Open XDMoD also has 400+ known installations at CI centers



Duration: Previous month Start: 2023-08-01 End: 2023-08-31 □ Available For Report

Page 1 of 1 □ Show Remainder (Disable Paging) Page Limit 10 Data Series

- **Service Provider:** A service provider is an institution that hosts resources.
- **ACCESS Credits Charged: Total:** The total amount of ACCESS Credit Equivalents charged. One ACCESS Credit Equivalent is defined as one CPU Hour on an SDSC Expanse standard compute node (AMD EPYC 7742). The ACCESS Credit Equivalent is a measure of how much compute time was used on each resource. One ACCESS Credit Equivalent is defined as one CPU Hour on SDSC Expanse (an AMD EPYC 7742 based compute resource). The ACCESS Credit Equivalent allows comparison between usage of node-allocated, core-allocated and GPU allocated resources. It also allows a comparison between resources with different compute power per core. The ACCESS allocations

ACCESS Metrics

XDMoD METRICS ON DEMAND

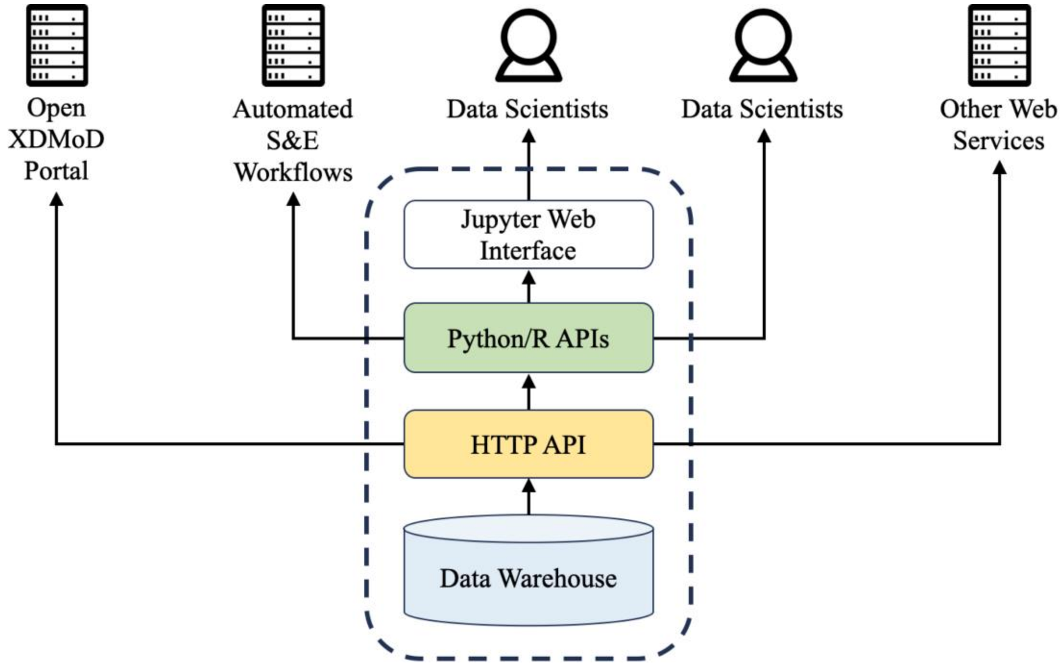
NSF

Motivation

- Open XDMoD is a well-established tool, but limited
- Charting is limited (e.g., no histograms, scatter plots, maps)
- Existing *Report Generator* has limited customization options
- Workload analyses benefit from data, but must circumvent portal and use external analysis tools
- Customization of portal requires extensive knowledge of software and time to run data pipelines
- Existing export capability is slow (daily batch job)

Design goals

- Data Analytics Framework for programmatic access to data in Open XDMoD
- Simple, documented, stable, versioned Application Program Interfaces (APIs) in HTTP, Python, and R
- Jupyter notebooks for documentation, training materials, and templates for analysis and reporting



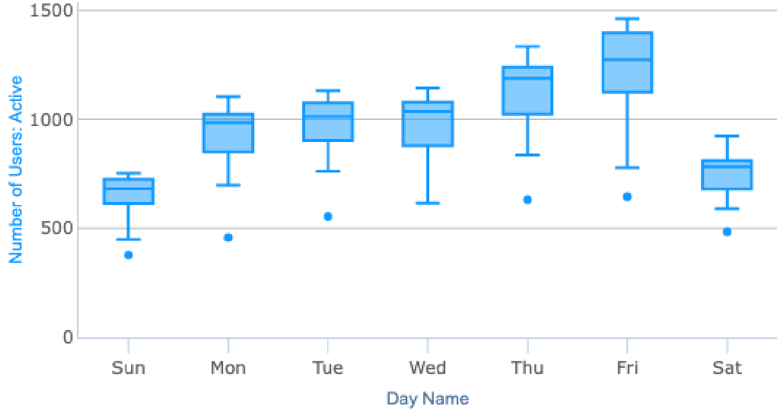
Python API

- Request data, load into Pandas data frames
- `pip install xdmod-data`
- Consistent functionality and terminology with Open XDMoD portal
- Version 1.0.0 released July 2023
- Compatible with Open XDMoD ≥10.5.0

```
data = data_warehouse.get_data(
    duration=('2023-01-01', '2023-04-30'),
    realm='Jobs',
    metric='CPU Hours: Total',
    dimension='Field of Science',
    filters={'Resource': 'Expanse GPU'},
    dataset_type='timeseries',
    aggregation_unit='Day'
)
```

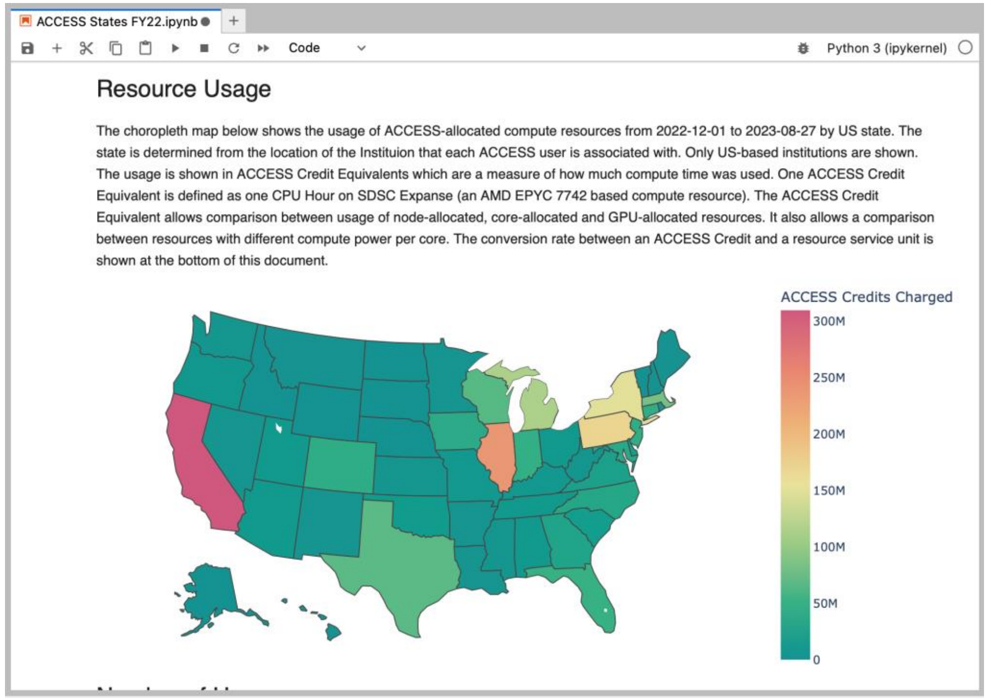
Example Jupyter notebooks

- GitHub repository: https://github.com/ubccr/xdmod-notebooks
- Instructions for running via Anaconda or Docker
- Used in PEARC23 tutorial
- **XDMoD-Data-First-Example.ipynb:**
  - Get data, make plots similar to what you can make in portal, make plots you cannot make in portal
- **XDMoD-Data-Raw-Data-Example.ipynb:**
  - Get raw data, group, filter, and plot
- **XDMoD-Data-Machine-Learning-Example.ipynb:**
  - Get raw data, run random forest regression model



Case study #1

- Usage of ACCESS-allocated resources by US state over nine-month period
- Use Python API to fetch data from *Jobs* realm of ACCESS XDMoD
  - Metrics:
    - Number of ACCESS credits charged
    - Number of active users
    - Number of active institutions
  - Group by *User State*
  - Filter by *User Country: United States*
- Join data from other sources
  - EPSCoR jurisdictions
  - State populations
- Create Markdown tables and choropleth plots

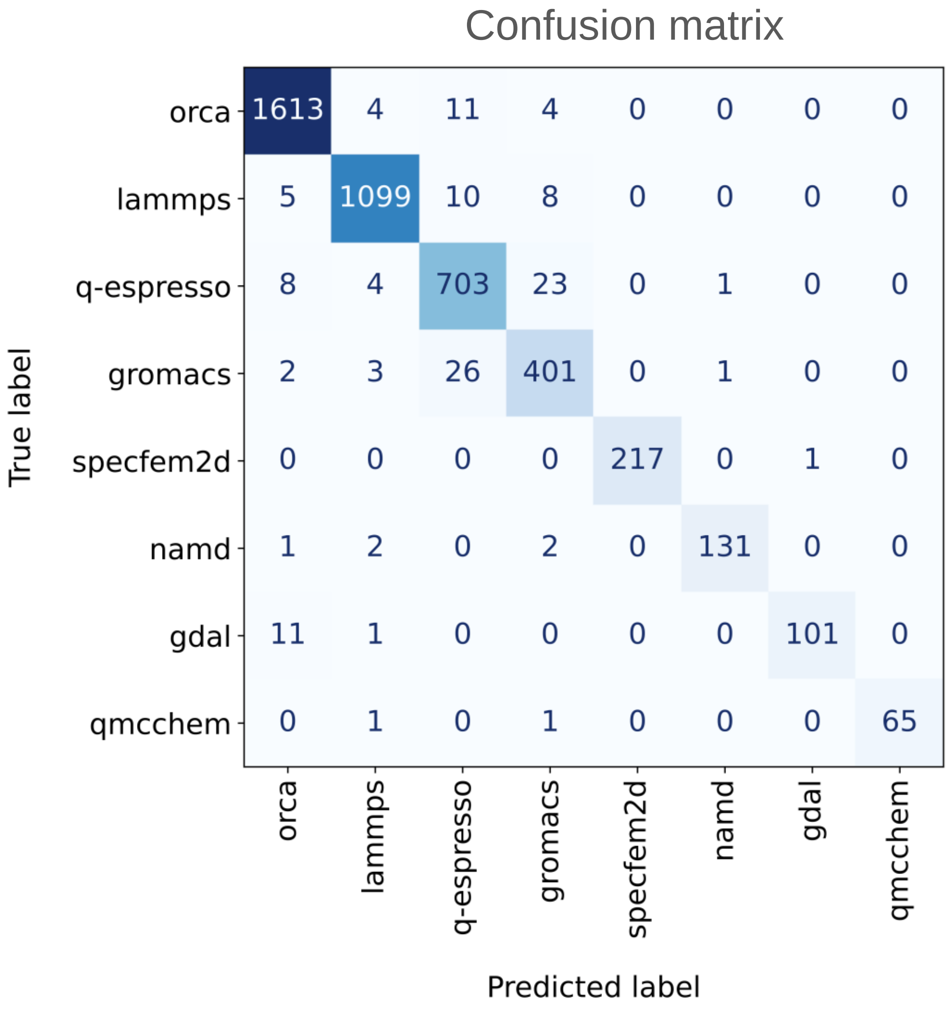


Resource Usage

The choropleth map below shows the usage of ACCESS-allocated compute resources from 2022-12-01 to 2023-08-27 by US state. The state is determined from the location of the Instituion that each ACCESS user is associated with. Only US-based institutions are shown. The usage is shown in ACCESS Credit Equivalents which are a measure of how much compute time was used. One ACCESS Credit Equivalent is defined as one CPU Hour on SDSC Expanse (an AMD EPYC 7742 based compute resource). The ACCESS Credit Equivalent allows comparison between usage of node-allocated, core-allocated and GPU-allocated resources. It also allows a comparison between resources with different compute power per core. The conversion rate between an ACCESS Credit and a resource service unit is shown at the bottom of this document.

Case study #2

- Machine learning random forest classification
- Predict software application given characteristics of compute job
- Use Python API to fetch raw data from Job Performance (*SUPReMM*) realm of ACCESS XDMoD
- Predictors: *CPU User, Wall Time, Total memory used, Net Ib0 Rx, Net Ib0 Tx, CPU User cov, Memory Used Cov, Net Ib0 Rx Cov, Net Ib0 Tx Cov*
- Filter top 8 applications over 2-month period
- Use scikit-learn
- 40,134 training rows, 4,460 test rows
- Out-of-bag accuracy: 97%



Confusion matrix

True label \ Predicted label	orca	lammps	q-espresso	gromacs	specfem2d	namd	gdal	qmcchem
orca	1613	4	11	4	0	0	0	0
lammps	5	1099	10	8	0	0	0	0
q-espresso	8	4	703	23	0	1	0	0
gromacs	2	3	26	401	0	1	0	0
specfem2d	0	0	0	0	217	0	1	0
namd	1	2	0	2	0	131	0	0
gdal	11	1	0	0	0	0	101	0
qmcchem	0	1	0	1	0	0	0	65

Future plans

- Workload and utilization analyses
- Regular software updates in tandem with Open XDMoD
- Improved performance of retrieving raw data (esp. in the *Jobs* realm)
- Improved consistency and simplicity of API
- Improved options for filtering data
- Improved R support
- Hosted Jupyter notebooks (no need to install software)
- Additional outreach

Contributions welcome

- GitHub Pull Requests
  - Example Jupyter Notebooks: https://github.com/ubccr/xdmod-notebooks
  - `xdmod-data` Python API: https://github.com/ubccr/xdmod-data
  - Open XDMoD: https://github.com/ubccr/xdmod

Acknowledgements

- This material is based upon work supported by the National Science Foundation under Grant No. OAC 2137603
- Thanks to the rest of the ACCESS MMS team (Matthew D. Jones, Abani K. Patra, Gregary Dean, Andrew Stoltman, Joshua Furlani, Jennifer Schopf, Shava Smallen, Vipin Chaudhary, Zahra Rahmani, Stephen Harrell, and Matt Cawood) and the other ACCESS teams
- Thanks to Nick Cruz for assistance with initial prototyping of the Python and Jupyter implementations