

The ATLAS data management system Rucio: Web-based monitoring and controlling interface

Mario Lassnig and Thomas Beermann, CERN PH-ADP, on behalf of the ATLAS Collaboration

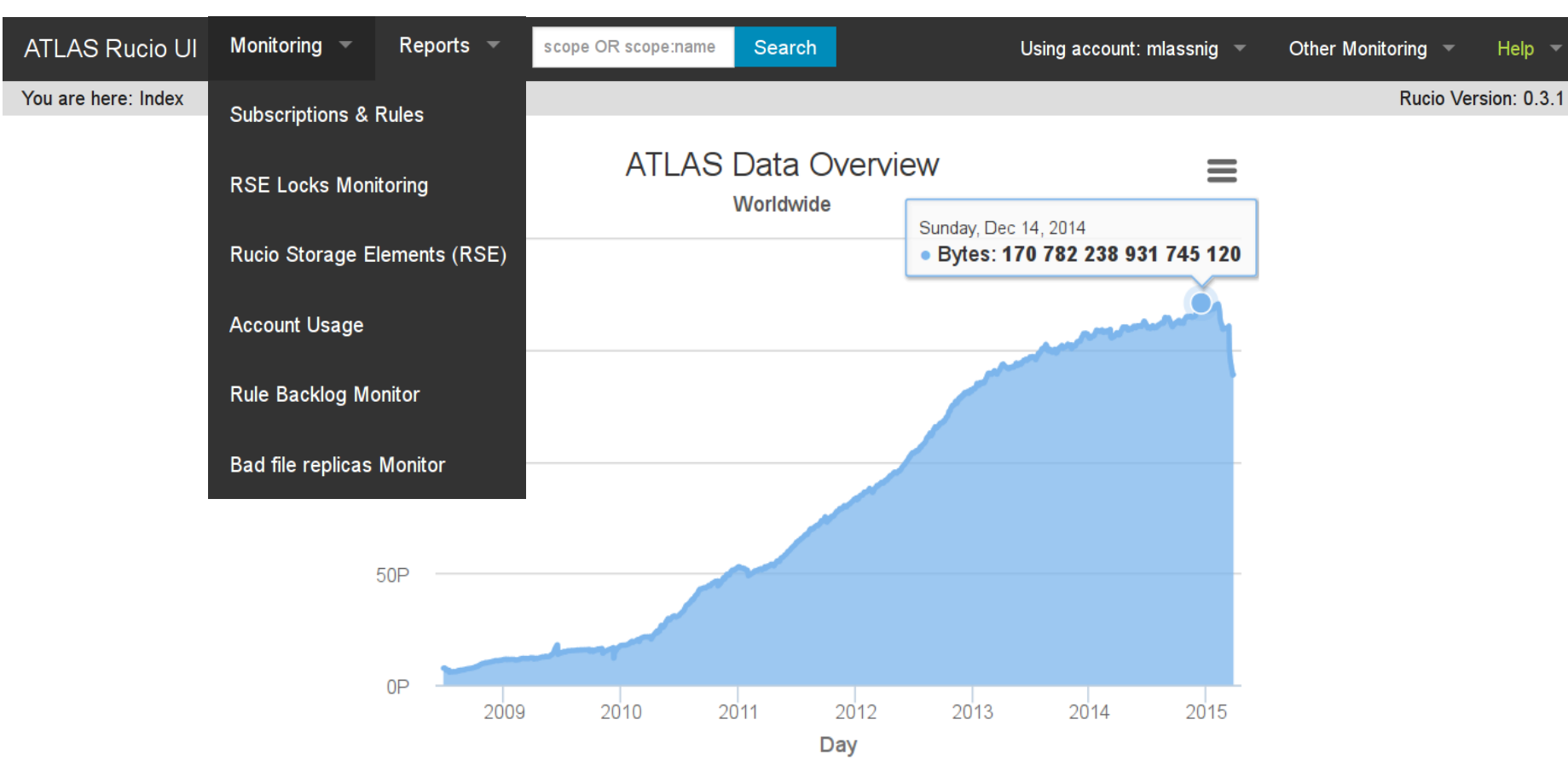
Rucio

Rucio manages all analysis, production, and user data for the ATLAS experiment. It is one of the technical underpinnings of the ATLAS Distributed Computing project, and provides a unified interface to globally distributed data. The Rucio components are in charge of indexing, querying, transferring, deleting, and monitoring. Reliability and performance are the priorities.

Web-based user interface

The users of Rucio require a comfortable, cross-platform interface to manage their data. The web-based interface provides a way to view the current state of the system, the contents and locations of data, occupancy of storage systems, the progress of operations, as well as other features like quotas. Additionally, users can insert new data placement requests through a guided interface. The Rucio web-interface is complementary to the CERN IT Dashboard, which provides real-time information on the underlying infrastructure.

Frontpage



Technology

web.py mod_wsgi Apache

- Scalable Python web stack
- Lightweight
- Maps URLs from Apache to Python
- Public domain
- Originally built for Reddit, serving millions of daily requests
- No external dependencies
- Works with event_mpm!

Foundation jQuery

- Foundation
- Responsive layout
- Templating for jQuery
- Choice of the HEP Software Foundation
- jQuery
- Async JavaScript library
- Implements Rucio API with REST
- Authenticate with Rucio via CORS

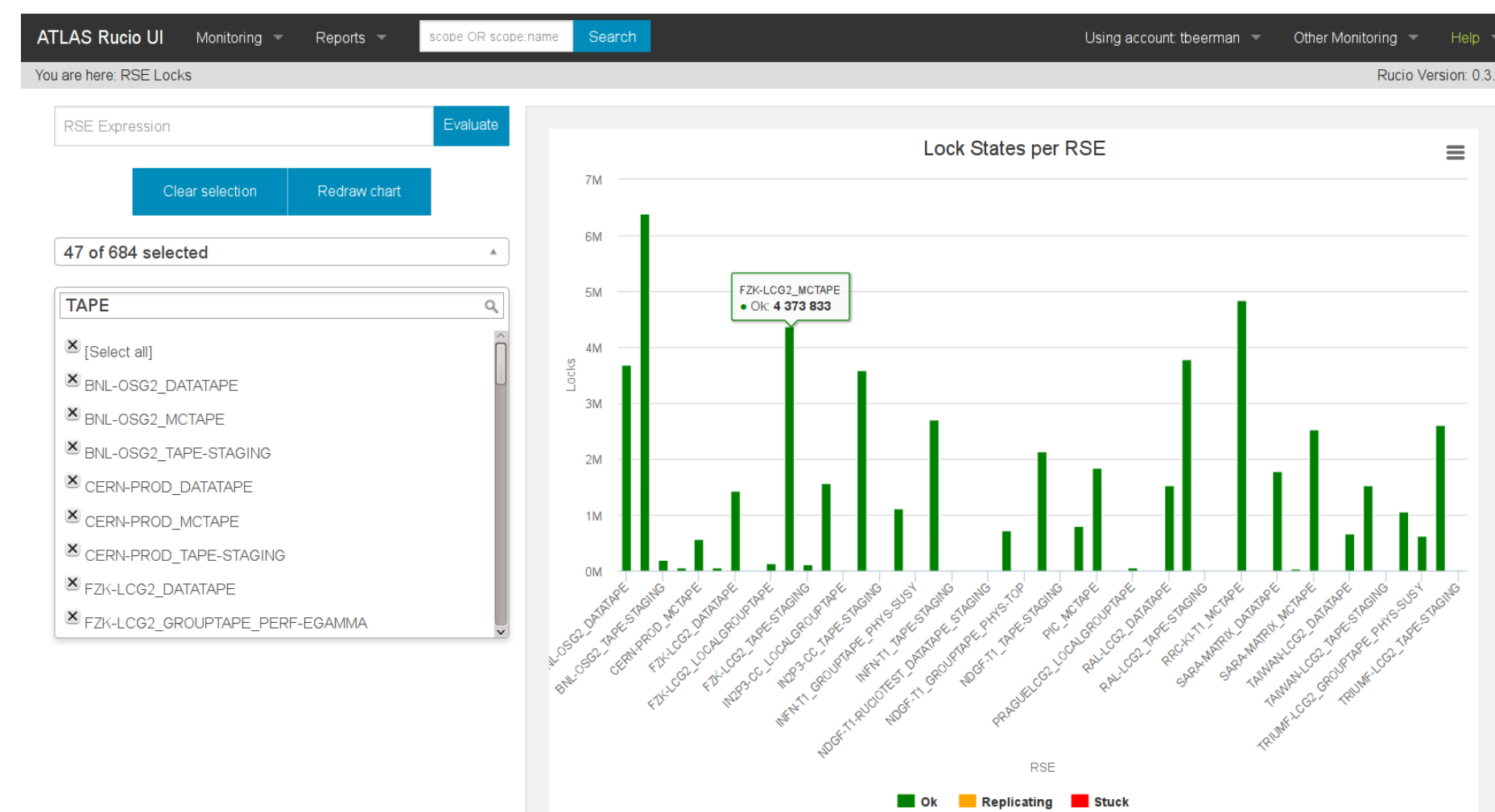
DataTables Highcharts

- DataTables
- jQuery plugin
- Pagination, instant search and multi-column ordering
- Highcharts
- Interactive plotting
- Highly Customised
- Very fast even on large graphs
- Filled via jQuery

Links

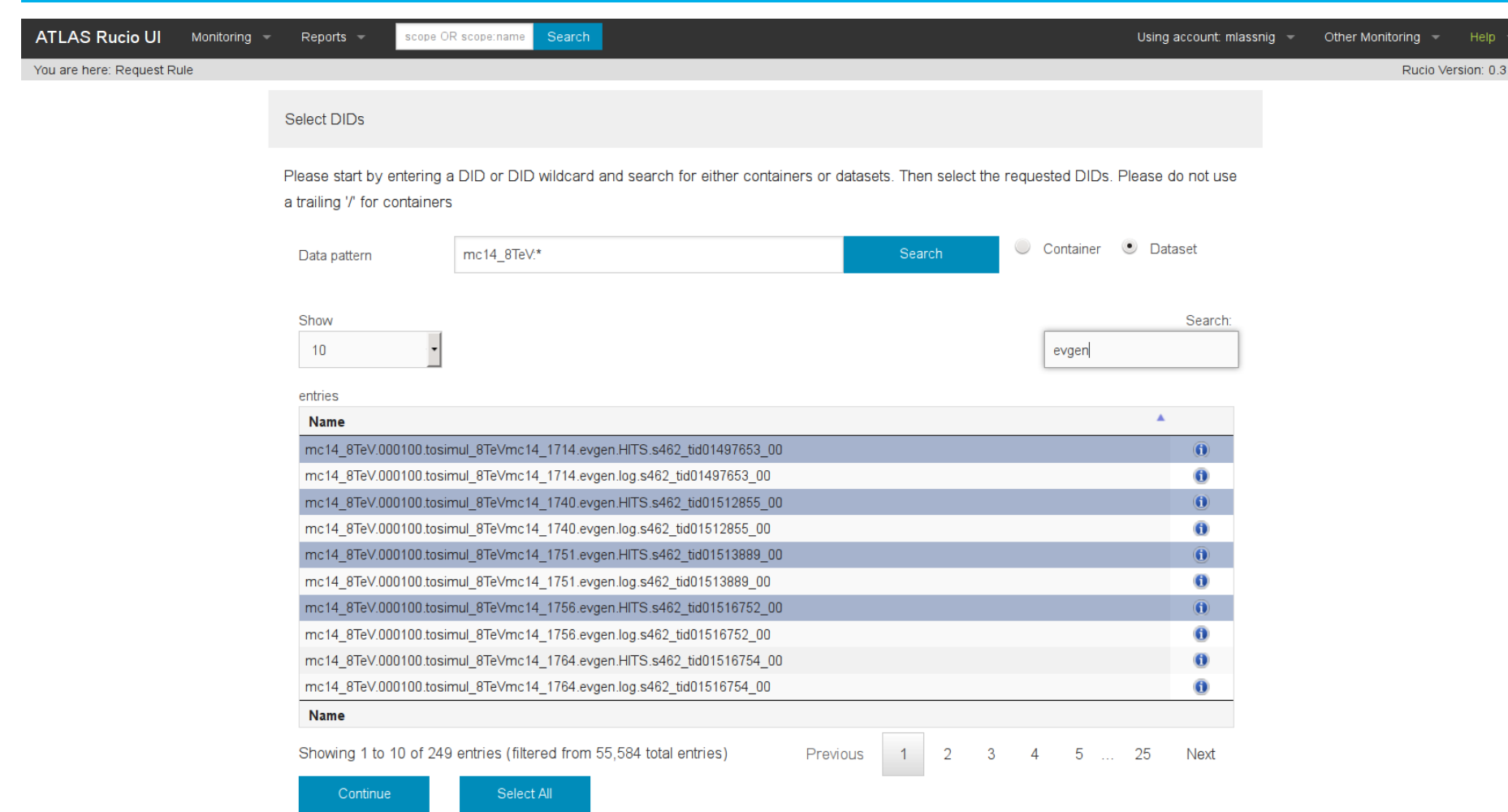
- <https://webpy.org/>
- <http://foundation.zurb.com/>
- <https://www.datatables.net/>
- <https://highcharts.com/>

Replication status of datasets



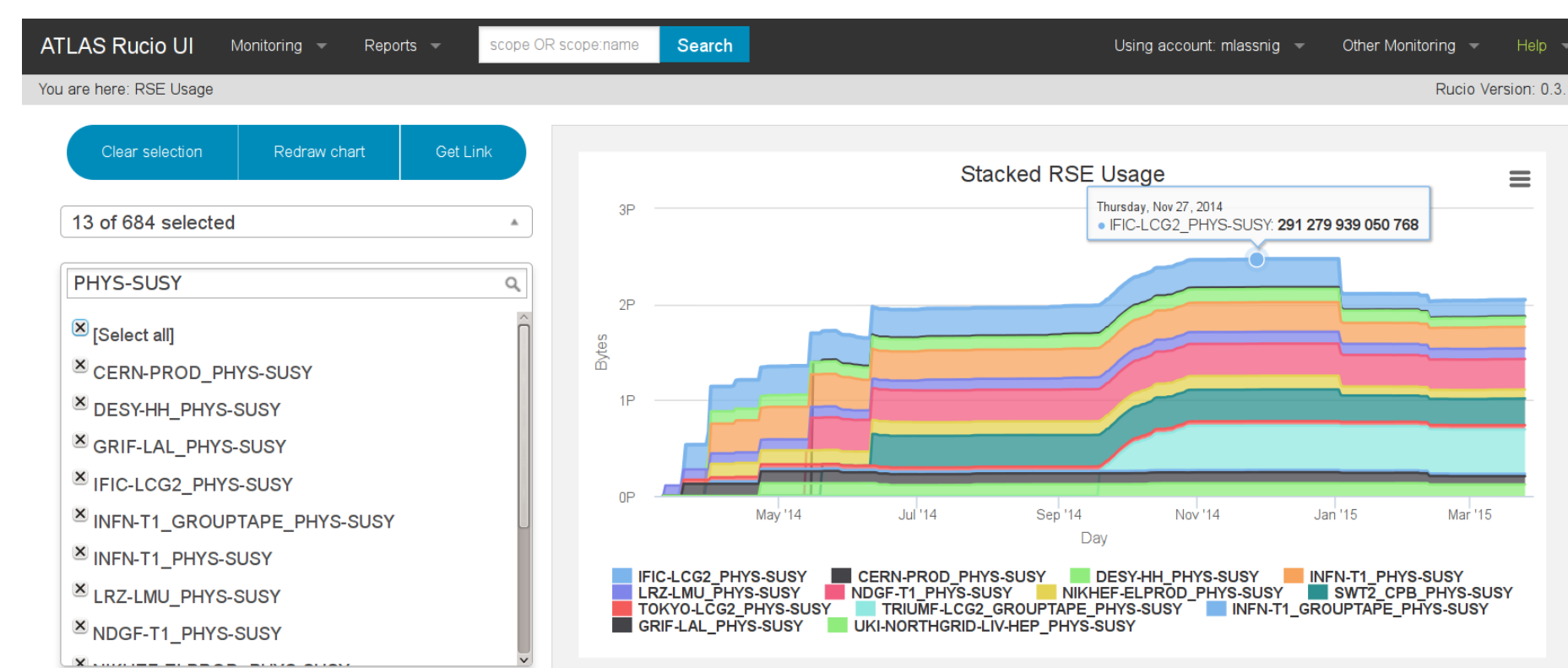
Locks on datasets are the principal data management unit. Users implicitly request locks on data to prevent their deletion, using replication rules. With this interface, it is possible to view the number of locks on a given Rucio Storage Element (RSE) with their according state. Dataset locks in state OK mean that all files which are attached to the dataset are physically available on the given RSE.

Requesting replication rules



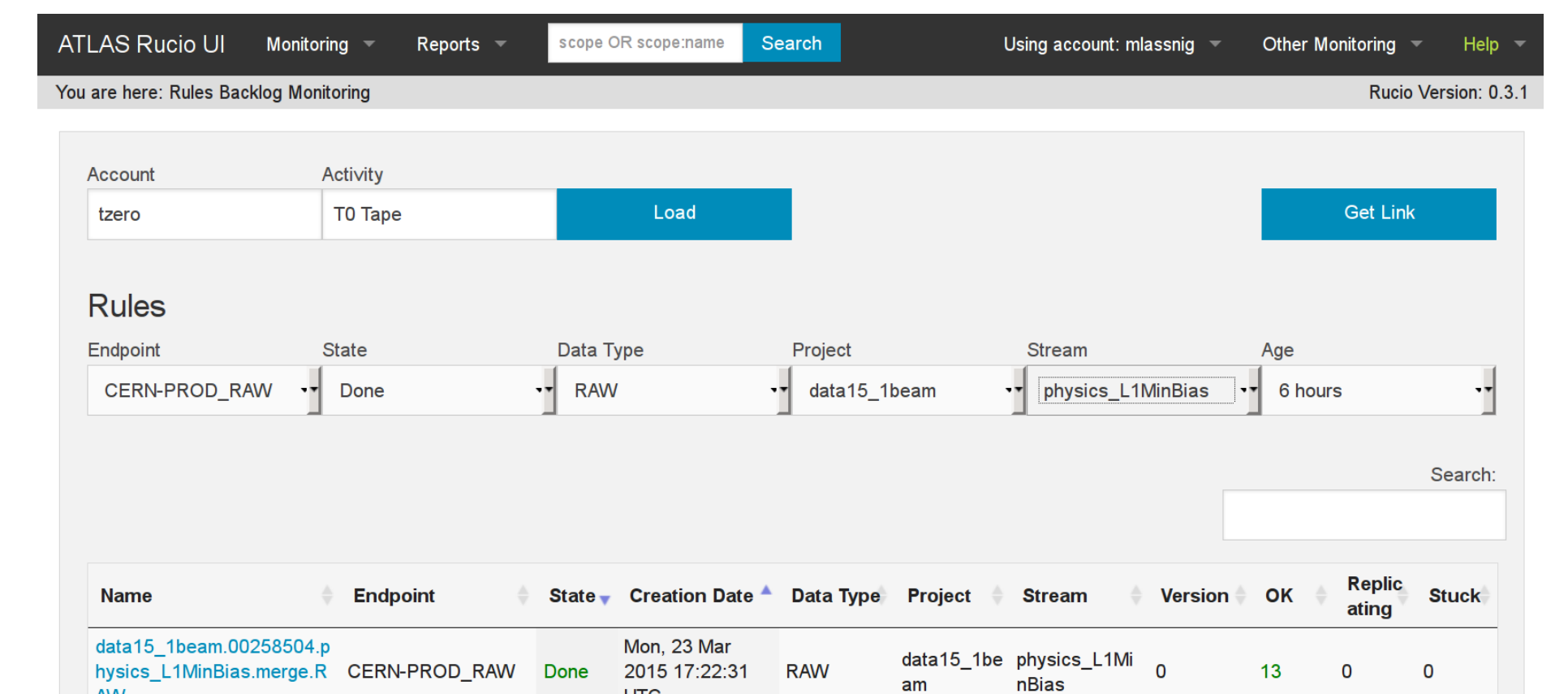
The second principal data management concept in Rucio is the replication rule. Requesting a rule creates the necessary dataset locks, which in turn potentially triggers data transfers. The rules also prevent data deletion, and support lifetimes and quotas. The interface provides a streamlined way to request rules, with a wizard that will guide the user through the process. Requested rules that would go over quota are automatically rejected.

Site occupancy



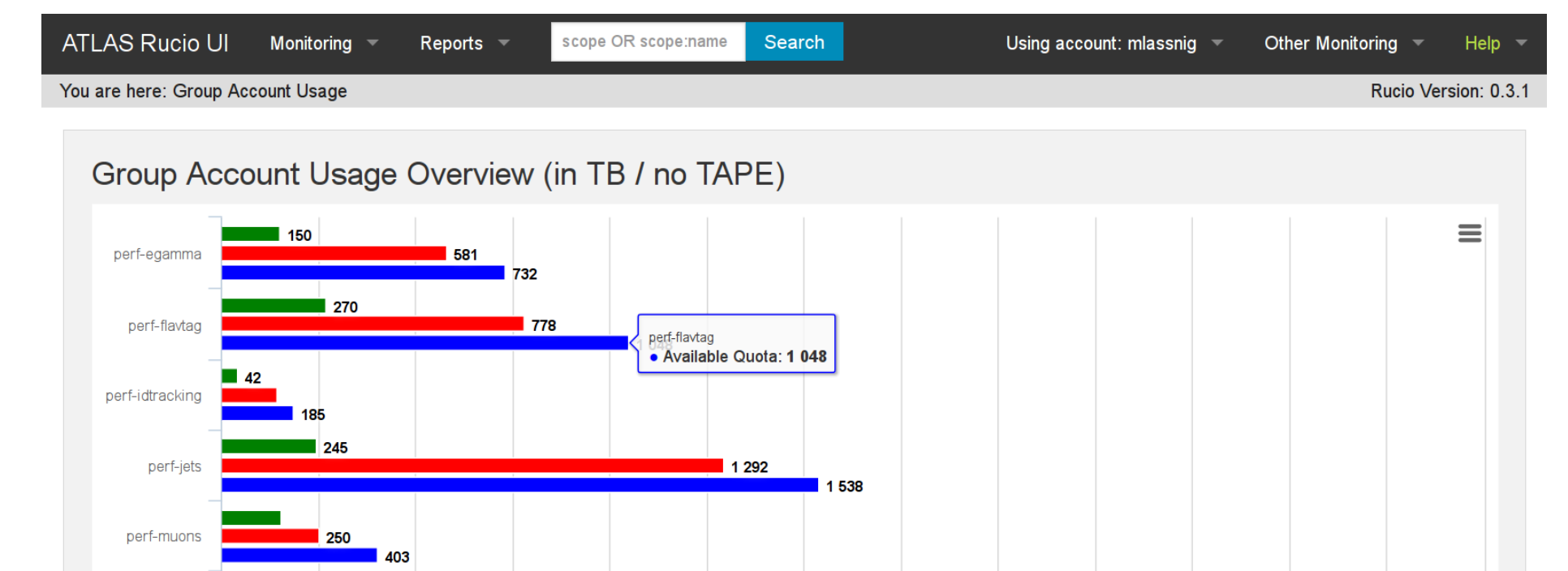
The selection of different RSEs can be done via set-theoretic expressions, or via a configurable dropdown form. The data is loaded asynchronously via jQuery into the Highcharts plot, and updates in real-time when a new selection is made. Grouped selections are automatically stacked, while retaining the single RSE data for comparison. This customizable approach superseded a multitude of previously static and pre-rendered monitoring webpages.

Backlog monitoring



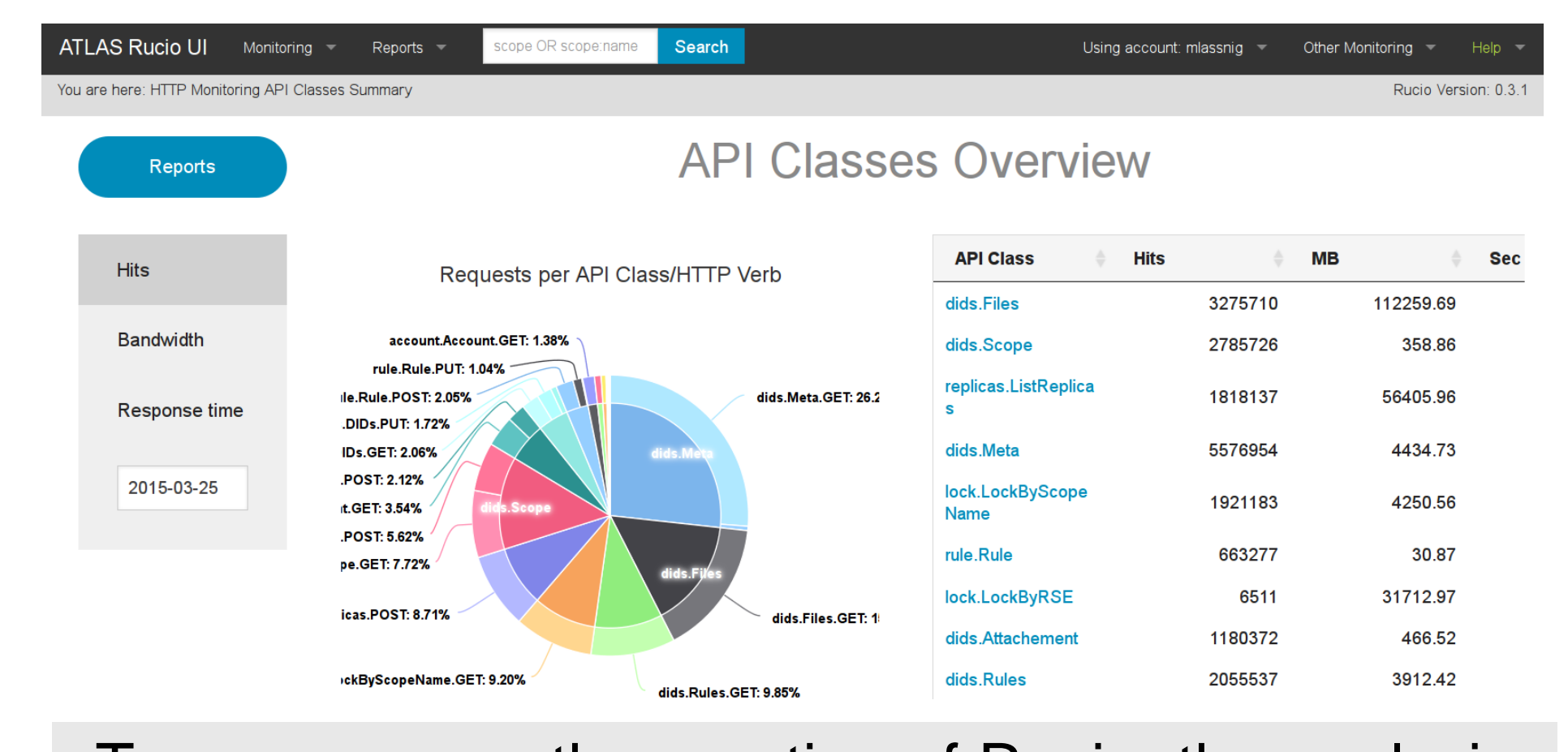
One of the most important views for daily operation is the backlog monitoring. Here, a user can view the distribution state of their account data, in a metadata breakdown. Sometimes, data is in a transitional state, for example, when it needs to be transferred to another data centre. The backlog monitoring makes it easy to follow the progress of these operations. This is especially critical for accounts like ATLAS Tier-0, which handle the export of data from CERN to the Tier-1s, and the archival of RAW data to tape. Additionally, users can oversee the progress of their replication rules.

Hadoop and Dumps



All Rucio servers and services stream their logfiles to Hadoop. Additionally, the majority of all Rucio catalogues is dumped to Hadoop periodically. These dumps are used for offline calculation of accounting metrics, site consistency checks, group space management, and more. These snapshots are streamed via an Apache Tomcat proxy, using a Java servlet, straight off HDFS, as bzip2 compressed, tab-separated files. Internally, they are stored in the Apache Avro column-based format. Internally, these dumps are used for views in the Rucio-UI that don't update often, for example, the group account monitoring.

Access monitoring



To ensure smooth operation of Rucio, the analysis of access patterns is required. This makes it easy to understand bottlenecks and hotspots. Breakdowns are provided by hits, bandwidth, and response time, grouped by accounts, API, resources, and scripts.

References

- Garonne et al., Rucio – The next generation of large scale distributed system for ATLAS Data Management, 2014 J. Phys.: Conf. Ser. 513 042021
- Barisits et al., ATLAS Replica Management in Rucio: Replication Rules and Subscriptions, 2014 J. Phys.: Conf. Ser. 513 042003