iRODS Logical Quotas
Policy Plugin

Be Boulder.
University of Colorado Boulder
Research Computing

- Compute clusters
- Data storage
- Networking
[root@irods1 ~]# ilsresc
plArchiveDouble:passthru
  pbox1Double:random
    strongbox2_05:unixfilesystem
    strongbox2_06:unixfilesystem
    strongbox2_07:unixfilesystem
    strongbox2_08:unixfilesystem
plArchiveSingle:passthru
  pbox1Single_slk:random
    strongbox1_01:unixfilesystem
    strongbox1_02:unixfilesystem
    strongbox1_03:unixfilesystem
    strongbox1_04:unixfilesystem

Globus
Login nodes
NFSRODS
iRODS catalog
logical quotas
StrongBox LTFS
iRODS resources
StrongBox shares
tape
tape
Logical Quotas

- Available as an RPM and DEB
- Operates as a discrete rule engine

```
"rule_engines": [{
  "instance_name": "irods_rule_engine_plugin-logical_quotas-instance",
  "plugin_name": "irods_rule_engine_plugin-logical_quotas",
  "plugin_specific_configuration": {
    "namespace": "irods::logical_quotas",
    "metadata_attribute_names": {
      "maximum_number_of_data_objects": "maximum_number_of_data_objects",
      "maximum_size_in_bytes": "maximum_size_in_bytes",
      "total_number_of_data_objects": "total_number_of_data_objects",
      "total_size_in_bytes": "total_size_in_bytes"
    }
  }
}, ...]
```
mkdir file-limited
ilogq start-monitoring /tempZone/home/janderson/file-limited
ilogq touch empty-file
for $f$ in file-limited/empty-file.{1..5}; do input empty-file $f$; done
ilogq ls /tempZone/home/janderson/file-limited

attribute: irods::logical_quotas::total_size_in_bytes
value: 0
units:
----

attribute: irods::logical_quotas::total_number_of_data_objects
value: 5
units:

ilogq set-max-objects /tempZone/home/janderson/file-limited 9
for $f$ in file-limited/empty-file.{6..10}; do input empty-file $f$; done
Level 0: Logical Quotas Policy Violation: Adding object exceeds maximum number of objects limit

ilogq ls /tempZone/home/janderson/file-limited

attribute: irods::logical_quotas::total_number_of_data_objects
value: 9
units:
----

attribute: irods::logical_quotas::maximum_number_of_data_objects
value: 9
units:
----

attribute: irods::logical_quotas::total_size_in_bytes
value: 0
units:

https://github.com/anderbubble/irods_rule_engine_plugin_logical_quotas/tree/ilogiq
[janderson@irods1 ~]$ ils file-limited
/tempZone/home/janderson/file-limited:
  empty-file.1
  empty-file.2
  empty-file.3
  empty-file.4
  empty-file.5
  empty-file.6
  empty-file.7
  empty-file.8
  empty-file.9
[janderson@irods1 ~]$
[janderson@irods1 ~]$ mkdir size-limited
[janderson@irods1 ~]$ iilogq start-monitoring /tempZone/home/janderson/size-limited
[janderson@irods1 ~]$ iilogq set-max-size /tempZone/home/janderson/size-limited 35MB
[janderson@irods1 ~]$ dd if=/dev/urandom of=random-file bs=1M
10+0 records in
10+0 records out
10485760 bytes (10 MB) copied, 0.0652263 s, 161 MB/s
[janderson@irods1 ~]$ for f in size-limited/random-file.{1..4} file $f; do
Level 0: Logical Quotas Policy Violation: Adding object exce e in bytes limit
[janderson@irods1 ~]$ ls size-limited
/tempZone/home/janderson/size-limited:
    random-file.1
    random-file.2
    random-file.3
[janderson@irods1 ~]$ iilogq ls /tempZone/home/janderson/size-limited
AVUs defined for collection /tempZone/home/janderson/size-limited:
attribute: irods::logical_quotas::maximum_size_in_bytes
type: value: 35000000 units: ----
attribute: irods::logical_quotas::total_number_of_data_objects
type: value: 3 units: ----
attribute: irods::logical_quotas::total_size_in_bytes
type: value: 31457280 units: ----
Logical Quotas Experience

• Real-world vs synthetic
• Experienced some drift in synthetic workloads
• Expecting to be able to clean it up in batch
• Maybe locking later?

https://gist.github.com/anderbubble/027de8b80366cd63df34b3838a7bab1d
Design and Implementation

- Initially implemented using the Native Rule Language
  - Pre-PEPs: Compute deltas and detect violations
  - Post-PEPs: Update totals using previously computed deltas
  - Successful as a proof of concept
  - Limited
- Reimplemented as a Rule Engine Plugin using C++
  - Made it easier to implement
  - Enabled better encapsulation and design
  - Performance

https://github.com/irods/irods_rule_engine_plugin_logical_quotas
C++ Rule Engine Plugin Design

```cpp
class pep_api_data_obj_copy final
{

public:
  pep_api_data_obj_copy() = delete;

static auto reset() noexcept -> void;
static auto pre(...) -> irods::error;
static auto post(...) -> irods::error;

private:
  // Shared data.
};

map<string_view, handler_type> handlers{
  {"pep_api_data_obj_copy_pre", pep_api_data_obj_copy::pre},
  {"pep_api_data_obj_copy_post", pep_api_data_obj_copy::post}
};
```
Performance over Correctness

Correctness Pros / Cons
- Pro: Reported numbers perfectly reflect the zone
- Con: Requires zone-wide synchronization
- Con: Decreased responsiveness on deeply nested collections

Performance Pros / Cons
- Pro: Does not require synchronization at all
- Pro: Interaction stays responsive
- Pro: Can always synchronize the logical totals with the real data
- Pro: The size of the data set does not affect performance
- Con: Reported numbers are relative due to drift
Interesting Findings

- Opening a data object with O_TRUNC was not reflected in the catalog immediately
  - Some operations require the data object’s size for proper operation
  - Use dstream’s seek operation
- Helped lead to the development of istream
  - Required a tool to test stream handlers
- The iRODS Filesystem library drastically simplified development
- Lead to the development of the [metadata_guard](https://github.com/irods/irods_rule_engine_plugin_logical_quotas) rule engine plugin

[GitHub repository](https://github.com/irods/irods_rule_engine_plugin_logical_quotas)
Questions and Contact Info

- github.com/irods/irods_rule_engine_plugin_logical_quotas
- irods.org
- colorado.edu/rc
- jonathon.anderson@colorado.edu
- irod-chat@googlegroups.com