



Parallel Transfer Between Python Client and S3 Storage

Justin James / Daniel Moore
(Applications Engineering)
iRODS Consortium

June 8-11, 2021
iRODS User Group Meeting 2021
Virtual Event

Server:

- iRODS Storage plugin abstracting an S3 "bucket"
 - https://github.com/irods/irods_resource_plugin_s3

Client:

- Python iRODS Client (PRC)
 - <https://github.com/irods/python-irodsclient>

New "Multi-1247" Parallel Transfer

- Multithreaded / Multiprocess
- For N threads ($1 \leq N \leq 4$ usually) there are N client-initiated connections instead of server-maintained high ports
- Client can re-use e.g. login credentials on all connects.
- Multiple processes on the iRODS server and S3 plugin end must match the client threads in "intent" (offset, length).

Challenges

- S3 should work with old and new styles of Parallel Transfer
 - For present, PRC must agree with iput/iget conventions
- S3 is non-POSIX - not as simple as `open()`, `read/write()`, `close()`
 - Imposed restrictions include minimum "multipart" size.
 - Multiprocess transfers requires shared memory for coordination between processes.
 - Failure recovery requires a shared memory timeout mechanism



Thank You!

Questions?