GoCommands: A cross-platform Command-line Client for iRODS

Illyoung Choi
CyVerse /
University of Arizona
iychoi@arizona.edu

Edwin Skidmore
CyVerse /
University of Arizona
edwin@cyverse.org

Nirav Merchant
CyVerse /
University of Arizona
nirav@arizona.edu

June 14, 2023
iRODS User Group Meeting

iCommands: An official command-line client

- Binary distributions for well known Linux distros
  - CentOS 7, Ubuntu 20, Debian 11 …
- Data access functions for users
  - Create, Delete, Rename, List, Read, Write collections and data objects
  - high performance data access using parallel data transfer (iput, iget)
- System control for admins
  - Control users, groups, resources (iadmin, igroupadmin)

Downside of iCommands

- Binary distributions are not available
  - For new Linux distros: Ubuntu 22.04
  - For other Linux distros: Arch Linux, OpenSUSE …
  - For other OSes: Windows, MacOS
  - For other CPU Architectures: Raspberry Pi (ARM 64) …
  - Not easy to build binaries from code by yourself
- Requires escalated privileges for installation
  - Not easy to obtain escalated privileges on institutional systems

    i.e., HPC, telescope control system

GoCommands: A cross-platform command-line client

- A re-implementation of iCommands in Go language
  - Cross-platform support by Go
  - More functions for user-friendliness (e.g., bput)
- Binary distributions available for
  - Almost any systems (Linux, Windows, MacOS)
- No installation
  - Download a single binary file then run (no dependencies)
  - Perfect for running it as a guest user on institutional systems

Supported environments

Binary Distribution	OS	CPU Architecture	Examples
Linux AMD64	Any Linux	Intel/AMD 64bit CPU	• Ubuntu 22.04 laptop
Linux ARM64	Any Linux	ARM 64bit CPU	• AlmaLinux with Amazon EC2 ARM64 instance • Raspberry Pi • Nvidia Jetson
Windows i386	Windows	Intel/AMD 32bit CPU	• Windows 7 old laptop (before 2010)
Windows AMD64	Windows	Intel/AMD 64bit CPU	• Windows 10 laptop
Darwin AMD64	MacOS	Intel/AMD 64bit CPU	• MacBook with Intel CPU
Darwin ARM64	MacOS	ARM 64bit CPU	• MacBook with M1/M2 CPU

Available commands

- Configuration
  - ★ “**init**”: Initialize iCommands configuration
  - “**env**”: Display current environment
- User
  - “**passwd**”: Change password
  - ★ “**copy-sftp-id**”: Copy SFTP public key
- System
  - “**ps**”: Display iRODS processes
  - “**svrinfo**”: Display iRODS server status
- GoCommands
  - ★ “**upgrade**”: Self upgrade to latest release
- Data Management
  - “**cd**”, “**pwd**”, “**ls**”, “**mv**”, “**rmdir**”, “**rm**”, “**cp**”, “**cat**”
  - “**bclean**”: Clear “bput” temporary files
  - “**bun**”: Extract bundle files in iRODS
  - ★ “**get**”, “**put**”: Download/upload data in parallel
  - ★ “**bput**”: Upload data using bundling + parallel (for many small files)
  - ★ “**sync**”: Differential data transfer (using “get”, “put”, “bput”, “copy”)
- Ticket Management
  - “**lsticket**”, “**mkticket**”, “**rmticket**”, “**modticket**”

Configurations

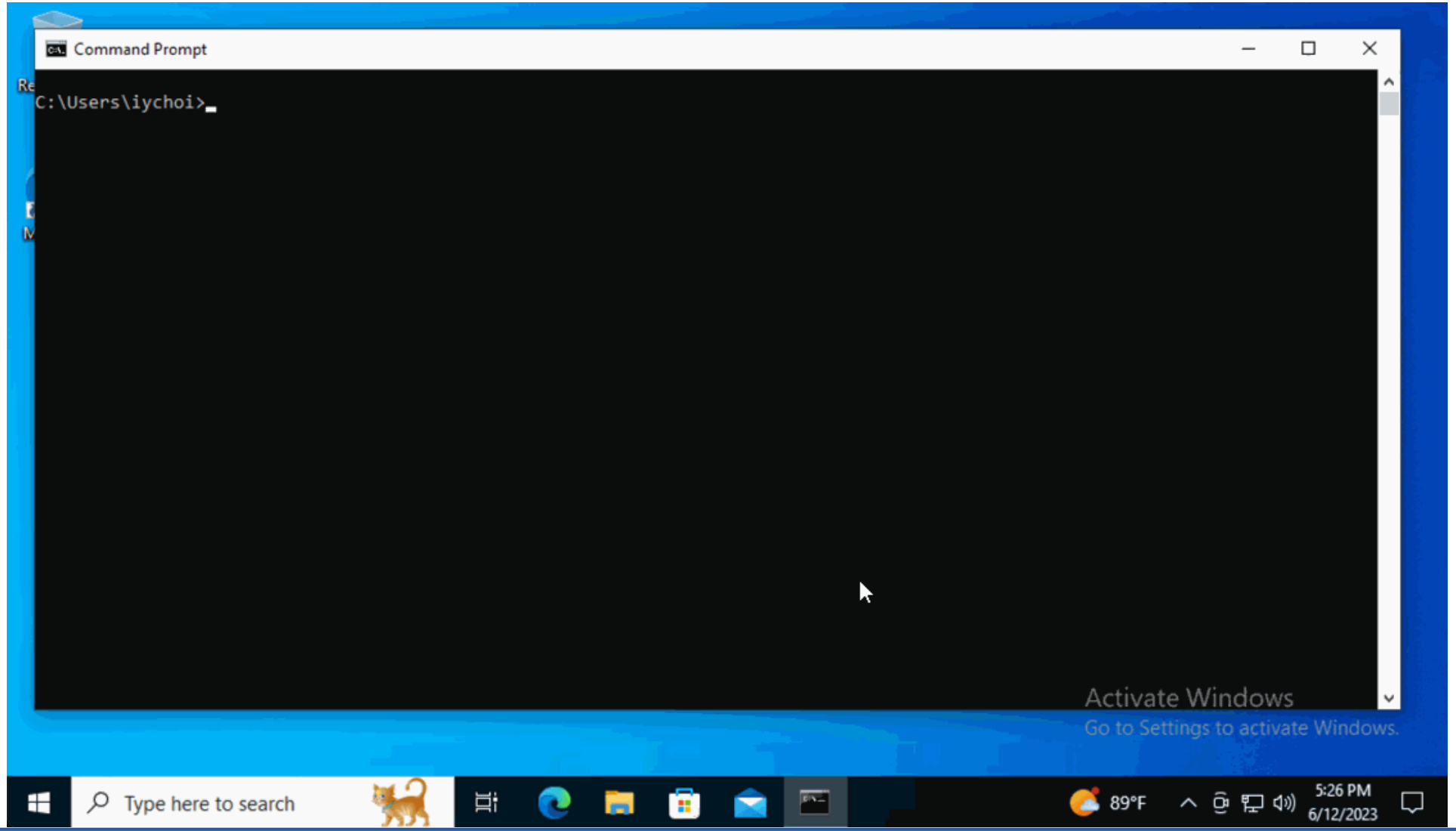
- GoCommands configuration
  - JSON or YAML file
  - Environmental variables

```
$ cat ./gocmd_config.yaml
irods_host: "data.cyverse.org"
irods_port: 1247
irods_user_name: "iychoi"
irods_zone_name: "iplant"
```

- iCommands configuration File
  - ~/.irods/irods_environment.json: configuration file
  - ~/.irods/.irodsA: encrypted password file

```
$ cat .irods/irods_environment.json
{
    "irods_host": "data.cyverse.org",
    "irods_port": 1247,
    "irods_user_name": "iychoi",
    "irods_zone_name": "iplant"
}
```

Quick demo on Windows 10

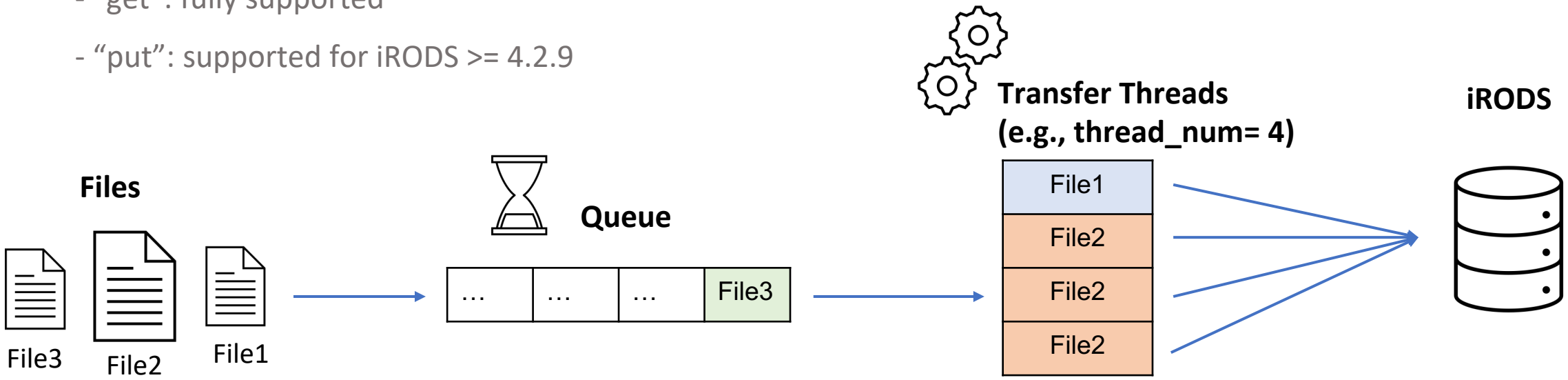


Transfer optimizations

- Multi-threaded data transfer for “put” and “get”
  - Over port 1247
  - Parallel transfer of files
  - Parallel transfer of data blocks in a file

    - “get”: fully supported

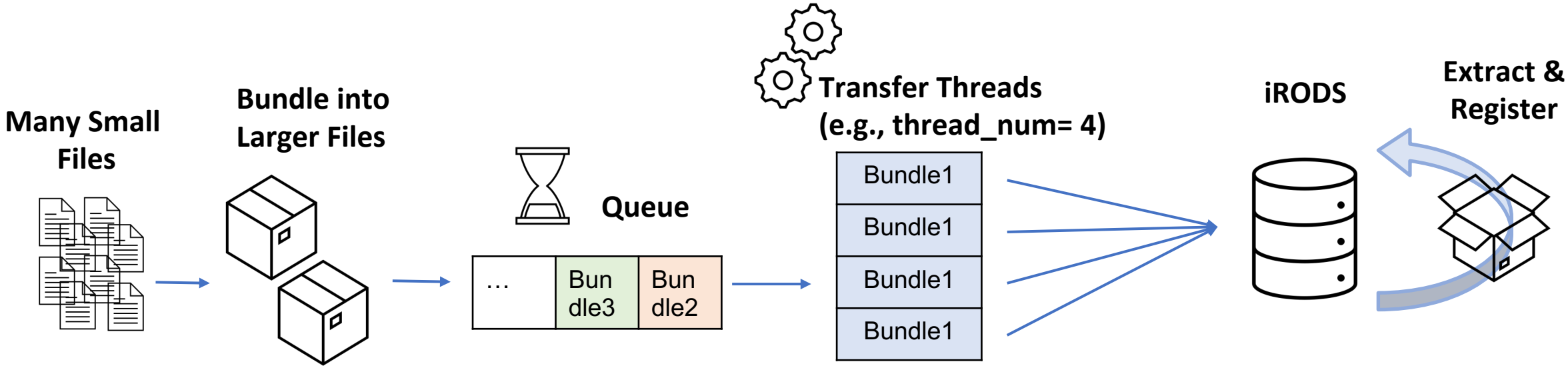
    - “put”: supported for iRODS >= 4.2.9



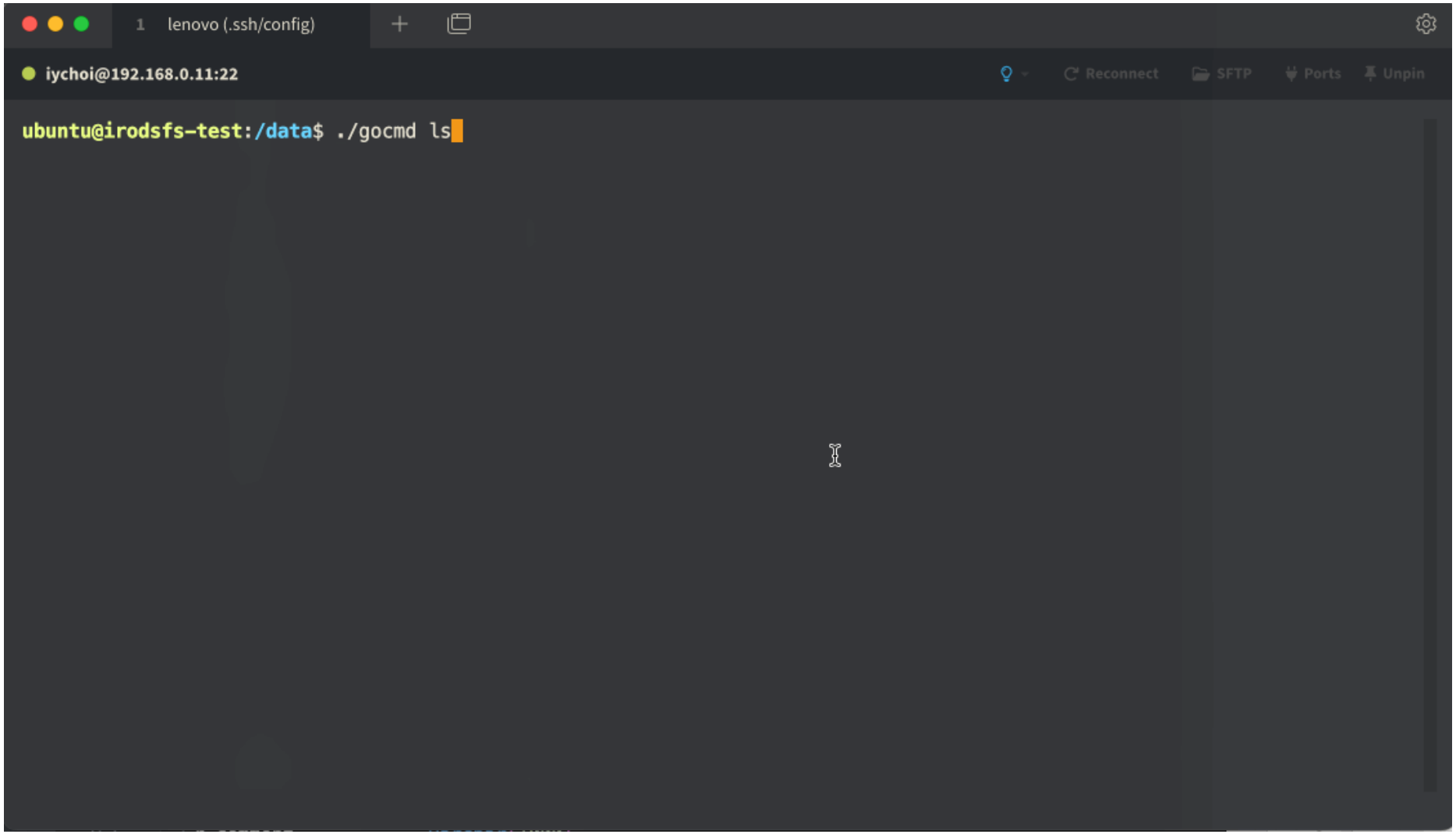
Transfer optimizations

- Bundled parallel data transfer
  - Bundling many small files into GB-size tarballs and transfer in parallel for efficient bandwidth use
  - Extract bundle files in the iRODS server
  - Best for many small files (50+, <= 1GB files)

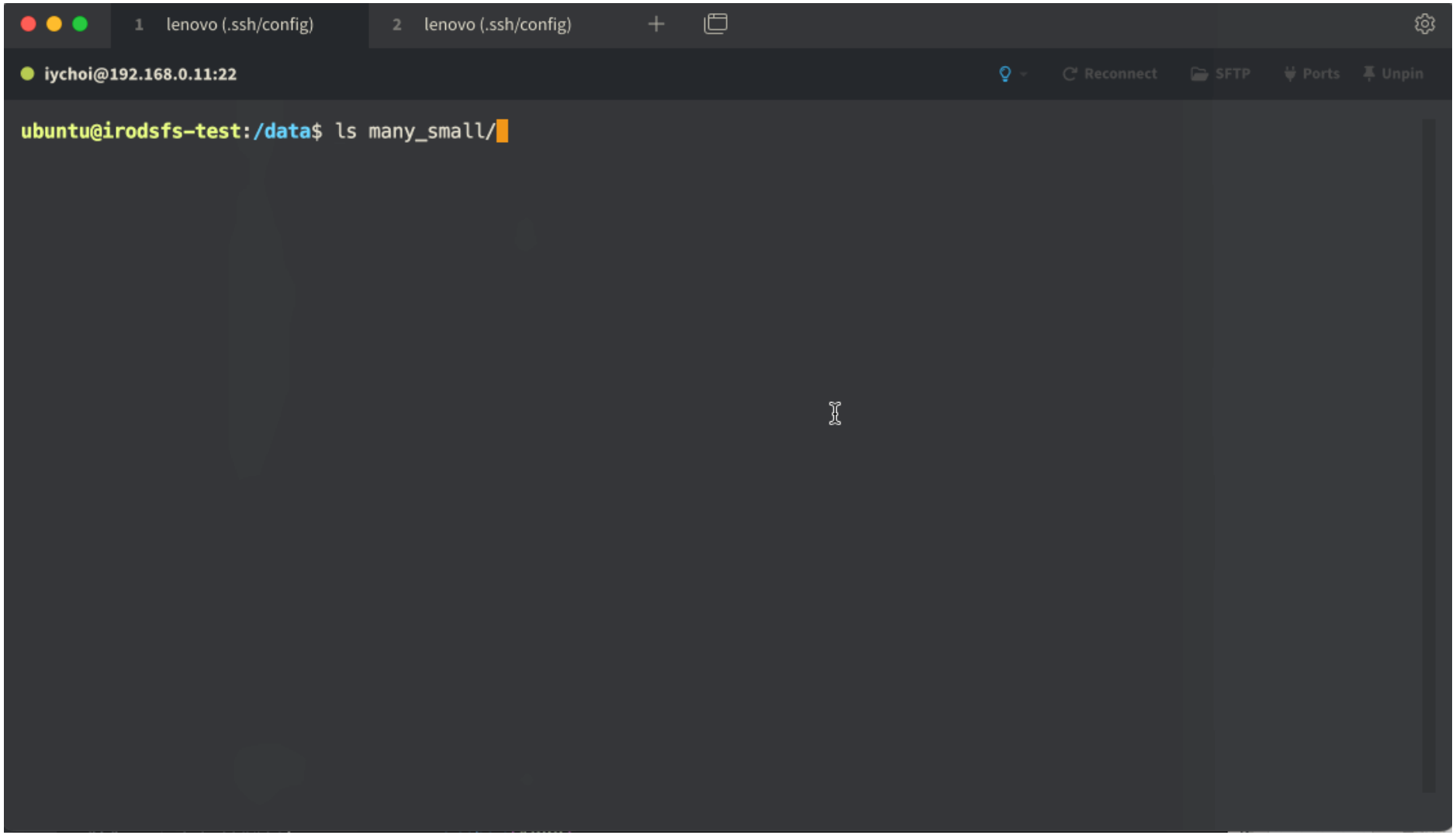
    - Used to transfer large telescope image data from Chile to CyVerse (US, Arizona)



Quick Demo - get



Quick Demo - bput



Failover

- Retry at failure
  - Data transfer commands, “put”, “get”, “bput”, and ”sync”, provide “--retry” flag

    - “--retry”: retry if fails with any error

    - “--retry_interval”: retry after certain interval (e.g., 10m)
  - Differential data transfer: “--diff” and “--no_hash”

    - “--diff”: compare the source and target files and transfer if only different

    - “--no_hash”: compare files by file sizes, not by using MD5 hash

Use-cases: Astronomic Data Transfer

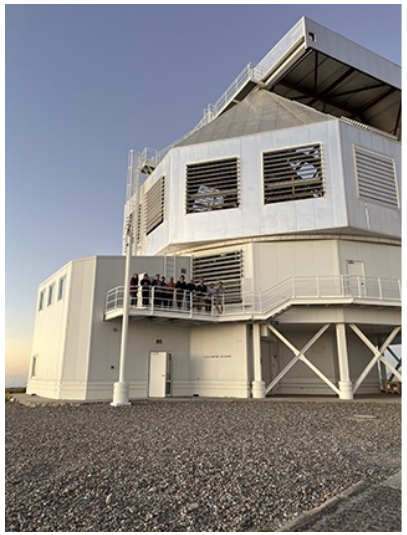
- MagAO-X (Magellan Adaptive Optics Extreme)
  - Previously, had to travel with a lot of hard disks
  - Transfer image data between Las Campanas Observatory in Chile and CyVerse in AZ, US

    - High-latency network connection

    ➔ larger TCP socket buffer & jumbo frames
  - A lot of short exposure images

    - many small files ➔ ”bput”
  - Roughly 20-25 MB/s






Magellan Clay telescopes at Las Campanas Observatory in Chile
By Joseph Long, University of Arizona

Use-cases: Aerial Image Transfer

- Open Forest Observatory
  - Transfer forest image data captured by drones

    - Many small files ➔ “bput”
  - From the field laptop via cellular network

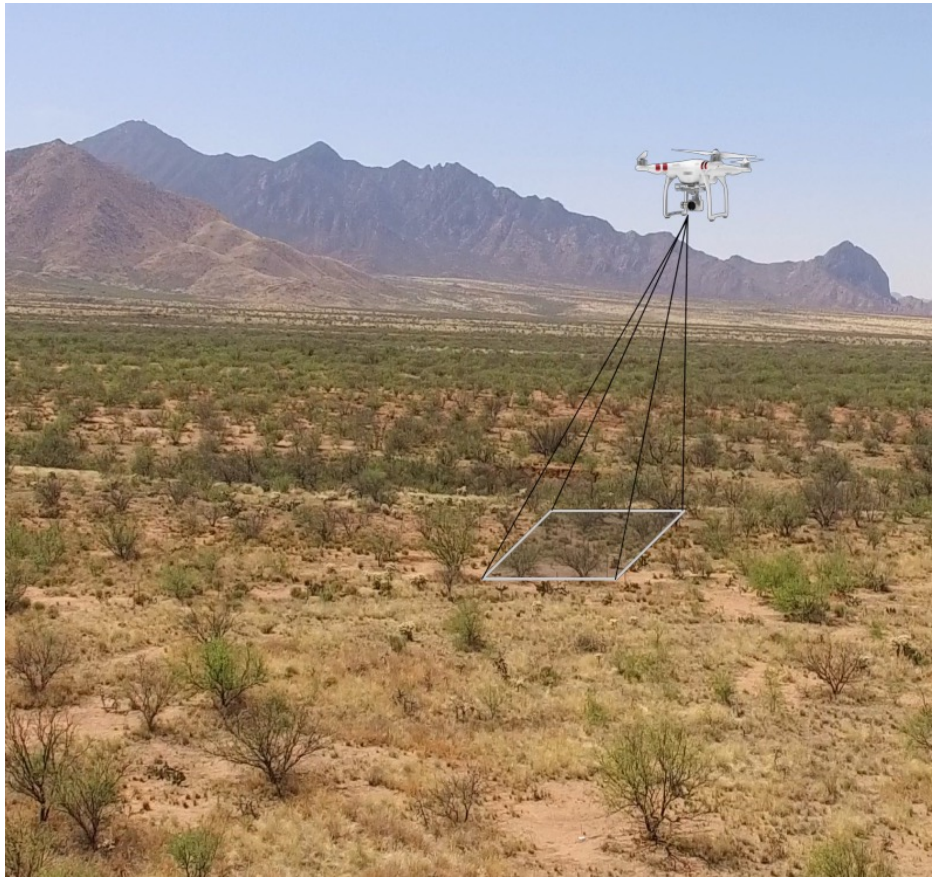
    - Low bandwidth, high latency network

      (Plan to switch to StarLink in the future)

    - Macbook laptop
  - Between CyVerse Data Store and compute clusters

    - CyVerse Discovery Environment and Jetstream2

    - No more `libssl` version issues on Ubuntu 22.04



Resource Mapping with Drones

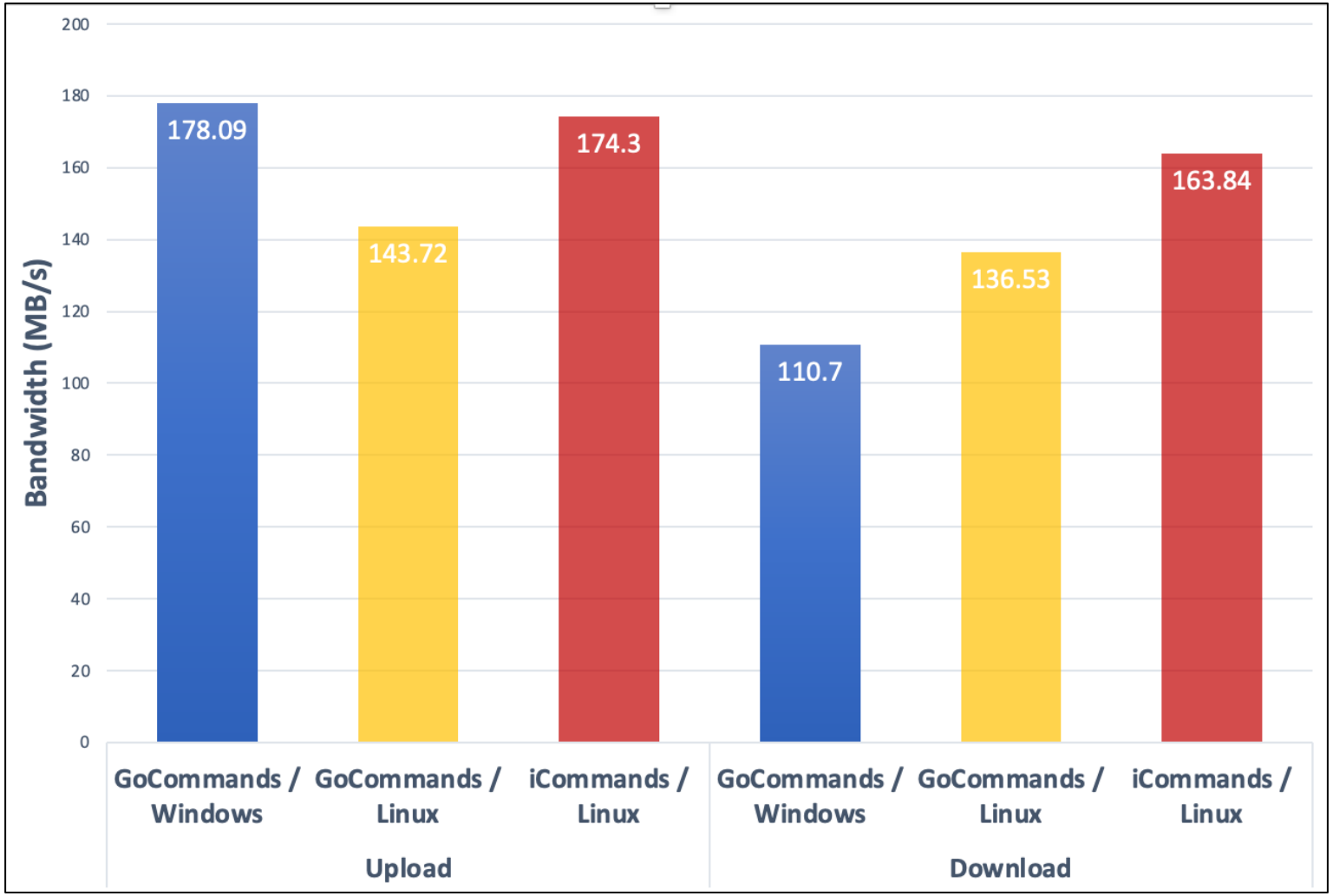
Use-cases: Ad-hoc Data Transfer

- CyVerse Discovery Environment
  - Replaced iCommands with GoCommands

    - No admin commands needed for general users

    - No dependency issues (some apps run on recent Linux distros)

Data Transfer Performance of GoCommands



GoCommands showed reduced performance than iCommands.

- No direct access to resource servers
- Yet unknown bottlenecks

Future work

Bugfixes

- Known issues
  (i.e., replication fail, metadata caching error, iCommands password decryption fail)
- New bugs reported by users

New features

- Access control commands (i.e., change ACL, list groups)
- AVU control commands (i.e., list/add/remove AVU)

Performance optimizations

- Implement direct data transfer from/to resource servers (like iget and iput)
- Investigate the current bandwidth bottleneck in Data Store

Open-source

https://github.com/cyverse/gocommands

Use github issues to ask questions or report bugs

Pull-Request is always welcomed!

Conclusion

A cross-platform command-line client

- Works on almost any systems
- No installation required
- Provides user commands sufficient for data access

Features

- Parallel file transfer & bundled parallel file transfer
- Retry at failure
- Differential data transfer

Deployed in production

- Included in Featured Apps in CyVerse Discovery Environment
- Used by our collaborators

Questions?