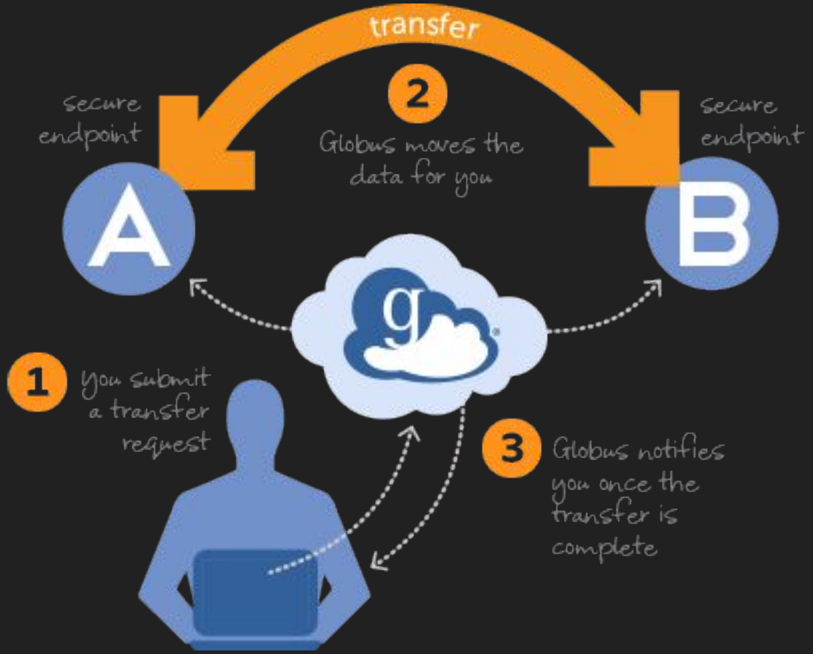
Automating globus®

Globus

Service that provides direct transfer of data between two endpoints.

Users submit requests via the web interface or the command line interface.


transfer
secure endpoint
2
Globus moves the data for you
secure endpoint
A
B
1
you submit a transfer request
3
Globus notifies you once the transfer is complete

Globus SDK: Automation

Why automate?

- Manual transfers are time consuming.

Globus Software Development Kit (SDK)

- A Pythonic interface to Globus’ REST API.

```
get_endpoint()
create_endpoint()
operation_mkdir()
submit_transfer()
submit_delete()
get_task()
cancel_task()
```

How It Works

Submits a transfer, comprised of the files in a specified directory that were added or modified since the last transfer.

```
globus_auto/
├── __init__.py
├── config.py
├── datastore.db
├── log
├── main.py
├── refresh_token
├── set_time.py
├── test_config.py
└── utils.py
```

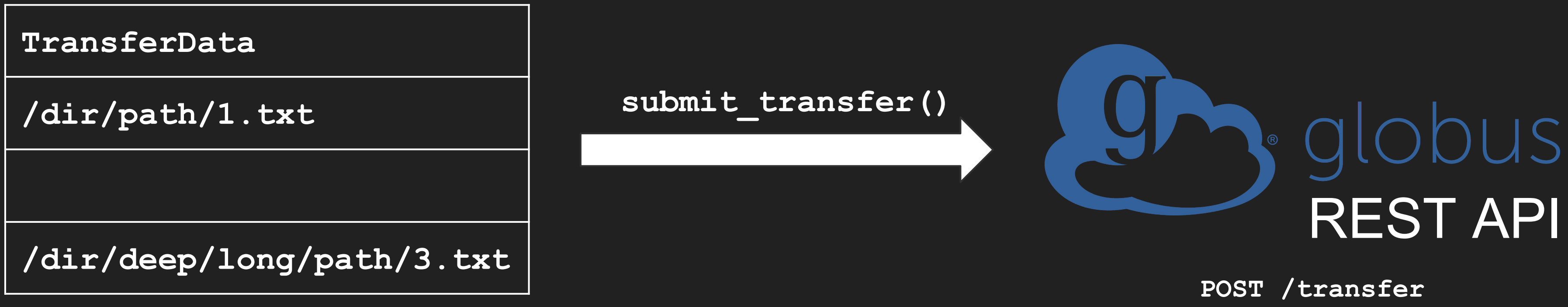
How It Works: Checking for Changes

Path	Last Transferred	Last Modified
`/dir/path/1.txt`	`2018-10-25 11:00:00`	`2018-10-25 11:13:42`
`/dir/long/path/2.txt`	`2018-10-25 11:00:00`	`2018-10-25 10:57:37`
`/dir/deep/long/path/3.txt`	`2018-10-25 11:00:00`	`2018-10-25 11:12:12`

How It Works: Checking for Changes

Path	Last Transferred	Last Modified
`/dir/path/1.txt`	`2018-10-25 11:15:00`	`2018-10-25 11:13:42`
`/dir/long/path/2.txt`	`2018-10-25 11:00:00`	`2018-10-25 10:57:37`
`/dir/deep/long/path/3.txt`	`2018-10-25 11:15:00`	`2018-10-25 11:12:12`

How It Works: Submitting a Transfer



TransferData
/dir/path/1.txt
/dir/deep/long/path/3.txt

`submit_transfer()`

globus REST API

`POST /transfer`

How It Works: Globus SDK Function Signatures

```
AuthClient.oauth2_get_authorize_url(additional_params=None)
AuthClient.oauth2_exchange_code_for_tokens(auth_code)

NativeAppAuthClient.oauth2_start_flow(requested_scopes=None, redirect_uri=None,
state='_default', verifier=None, refresh_tokens=False, prefill_named_grant=None)

RefreshTokenAuthorizer(refresh_token, auth_client)

TransferClient(authorizer=None, **kwargs)
     operation_mkdir(endpoint_id, path, **params)
     get_endpoint(endpoint_id, **params)
     endpoint_get_activation_requirements(endpoint_id, **params)
     submit_transfer(data)

TransferData(transfer_client, source_endpoint, destination_endpoint, label=None,
submission_id=None, sync_level=None, verify_checksum=False, preserve_timestamp=False,
encrypt_data=False, deadline=None, recursive_symlinks=u'ignore', **kwargs)
```

How It Works: Automation

Automation comes from Linux cron.

Schedules a job to run periodically at fixed times, dates, or intervals.

`main.py` every five minutes:

```
*/5 * * * * python3 main.py
```

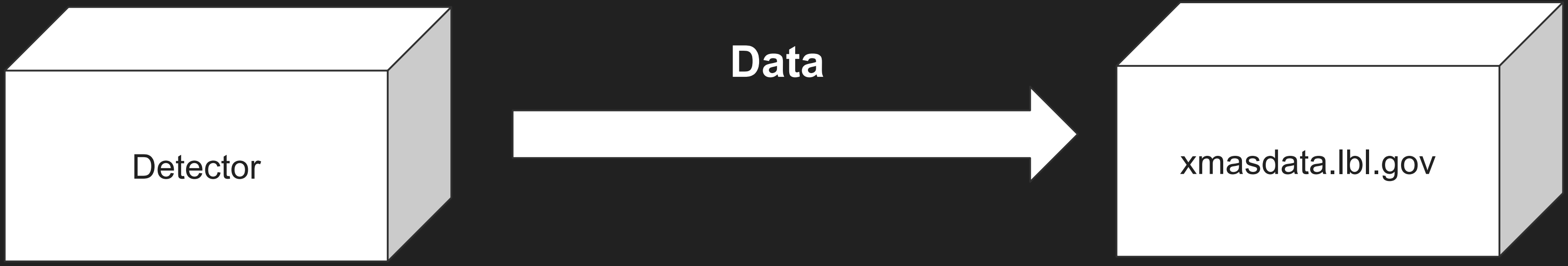
`main.py` at midnight every Sunday:

```
0 0 * * 0 python3 main.py
```

Fun(?) Fact: “cron” is named after the Greek God Chronos

Chronos and his Child, by Giovanni Francesco Romanelli

ALS Microdiffraction: Incoming



"/xmasdata/2018"

Format: .tif (Tagged Image File Format)

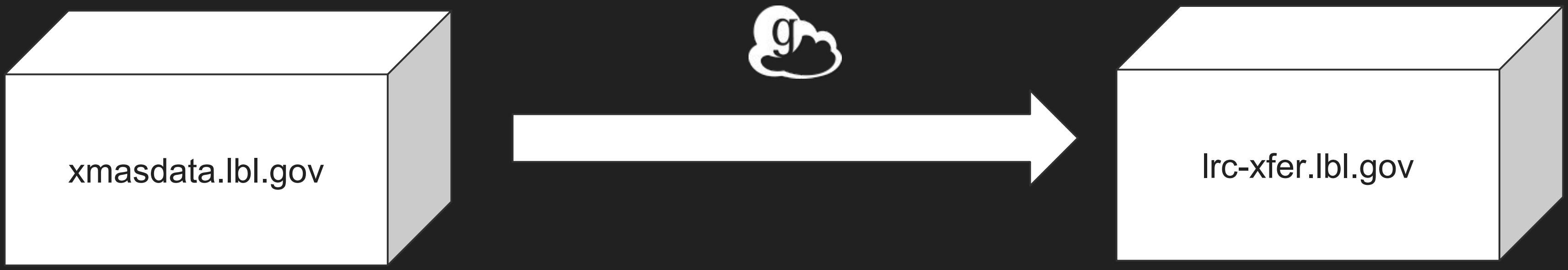
Size: 4.0 MB / file

Rate (5 minute window):

Average: < 600 files (2.4 GB)

Maximum: 3,000 files (12.0 GB)

ALS Microdiffraction: Outgoing



xmasdata.lbl.gov

lrc-xfer.lbl.gov

`"/xmasdata/2018"`

detdata
Globus Connect Personal (GCP)

`"/clusterfs/xmas/
xmas-storage/2018"`

lbnl#lrc
Globus Connect Server (GCS)

ALS Microdiffraction: Configuration

```
# The absolute path to the source directory in the source endpoint.
SRC_DIR = "/xmasdata/2018"
# The absolute path to the destination directory in the destination endpoint.
DST_DIR = "/clusterfs/xmas/xmas-storage/2018"
# The ID of the source endpoint.
SRC_ID = "01ifstxr-6a8m-vy9p-y7rc-d7tm5oavkawc"
# The ID of the destination endpoint.
DST_ID = "x2201bp1-1h0g-yq0w-g6gt-mozyu2lxjqf0"
# The ID of the client application performing Globus transfers.
CLIENT_ID = "c0qeiflo-3rju-yhjr-s4xj-kqcilwz0111h"
# The absolute path to the directory in which main.py resides.
CODE_PATH = "/home/det/software/globus_auto"

# IDs were randomly generated and are (probably) invalid.
```

ALS Microdiffraction: cron

```
[det@xmasdata ~]$ crontab -l
*/15 * * * * /usr/local/bin/python3
/home/det/software/globus_auto/main.py
```

Storage: Whole

Absolute paths are stored directly as keys, each pointing to its timestamp.

Highly redundant storage of prefixes.

Optimization: Lazily prepend the longest common prefix only when needed.

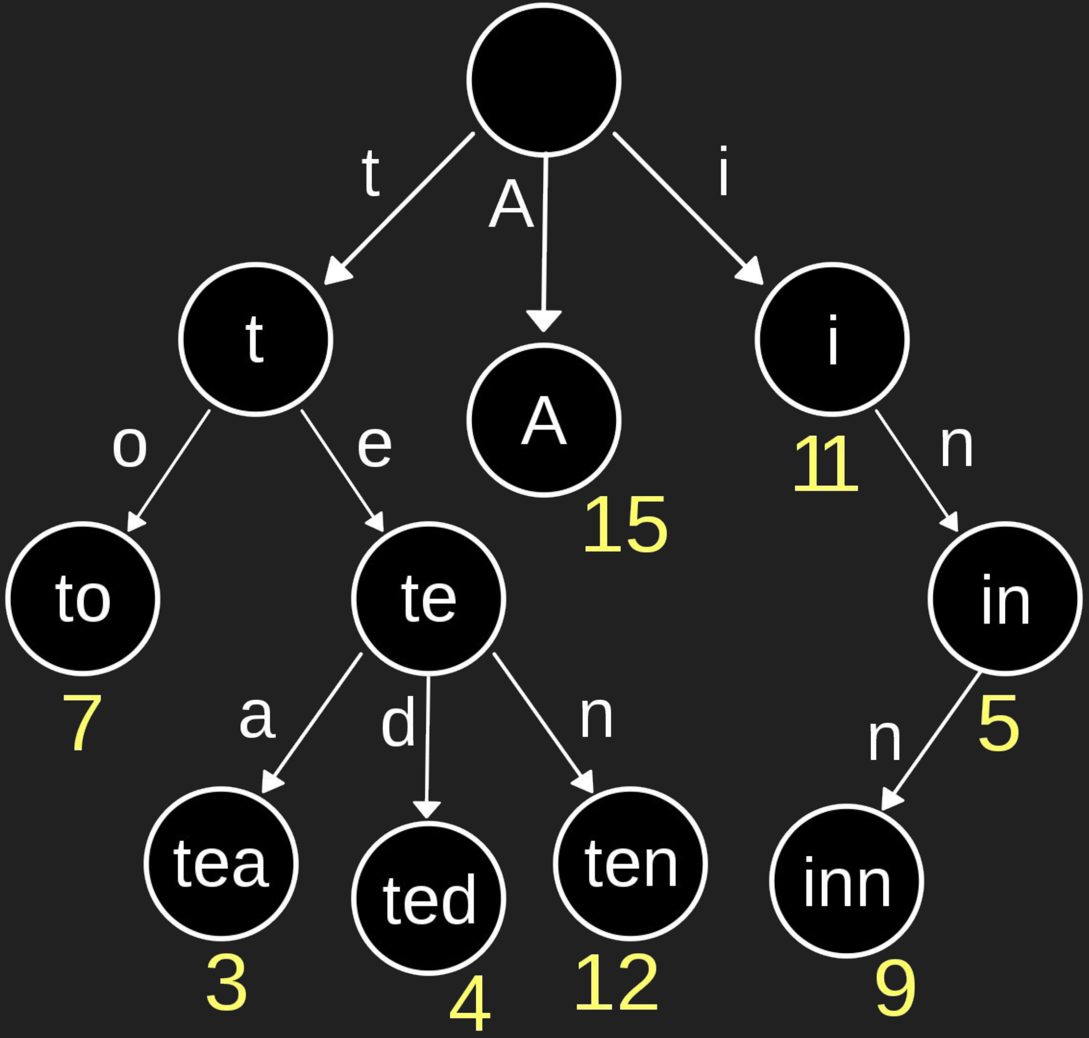
```
/Users/matthewli/Desktop/fg_dir/0
/Users/matthewli/Desktop/fg_dir/0/vSUzPkz3g3UGFHoo
/Users/matthewli/Desktop/fg_dir/0/0
/Users/matthewli/Desktop/fg_dir/0/0/0
/Users/matthewli/Desktop/fg_dir/0/0/atelR45jHFJqhF3f
/Users/matthewli/Desktop/fg_dir/0/0/1ERWnEQx9pe6wPTG
/Users/matthewli/Desktop/fg_dir/0/1
/Users/matthewli/Desktop/fg_dir/0/1/0
/Users/matthewli/Desktop/fg_dir/0/1/1
/Users/matthewli/Desktop/fg_dir/0/1/1/SbNUe6kWEnLJhC6A
/Users/matthewli/Desktop/fg_dir/0/1/1/9dpOz3t5fpgC0eOq
/Users/matthewli/Desktop/fg_dir/0/1/tBoSPujmCpAzmQmw
/Users/matthewli/Desktop/fg_dir/0/2vrDrsGuFTVf84hP
/Users/matthewli/Desktop/fg_dir/1
/Users/matthewli/Desktop/fg_dir/1/0
/Users/matthewli/Desktop/fg_dir/1/0/7mSOJltZ5ohr3aFG
/Users/matthewli/Desktop/fg_dir/1/0/0
/Users/matthewli/Desktop/fg_dir/1/0/1
/Users/matthewli/Desktop/fg_dir/1/Vn3Lsnj15RKKplKY
/Users/matthewli/Desktop/fg_dir/yaMVWuX3nxQn4HRd
```

Trie

Structure for efficient storage of data with shared prefixes.

A node's position in the tree determines the key to the data.

Applications: Autocomplete, Spell Check



Antidisestablishmentarianism

A

An

Ant

Anti

Antidisestablish

Antidisestablishment

Antidisestablishmentarian

Antidisestablishmentarianism

99 characters

A→n→t→i→d→i→s→e→s→t→a→b→l
→i→s→h→m→e→n→t→a→r→i→a→n
→i→s→m

28 characters (+ 27 pointers)

Storage: Directory Trie

Each node in the trie represents a file or a directory and points to its timestamp.

Benefits increase with path lengths and the number of files.

```
/Users/matthewli/Desktop/file_generator/fg_dir
├── 0
│   ├── 0
│   │   ├── 0
│   │   ├── 1ERWnEQx9pe6wPTG
│   │   └── atelR45jHFJqhF3f
│   ├── 1
│   │   ├── 0
│   │   ├── 1
│   │   │   ├── 9dpOz3t5fpgC0eOq
│   │   │   └── SbNUe6kWEnLJhC6A
│   │   └── tBoSPujmCpAzmQmw
│   ├── 2vrDrsGuFTVf84hP
│   └── vSUzPkz3g3UGFHoo
├── 1
│   ├── 0
│   │   ├── 0
│   │   ├── 1
│   │   └── 7mSOJltZ5ohr3aFG
│   └── Vn3Lsnj15RKKplKY
└── yaMVWuX3nxQn4HRd
```

Path Storage: The Difference

Long Paths

```
/xmasdata/2018/redacted_Sep2018/YAGline9test/YAGline9test_00001.tif
/xmasdata/2018/redacted_Sep2018/YAGline9test/YAGline9test_00002.tif
...
/xmasdata/2018/redacted_Sep2018/YAGline9test/YAGline9test_01791.tif
/xmasdata/2018/redacted_Sep2018/YAGline9test/YAGline9test_01792.tif
```

Many Files

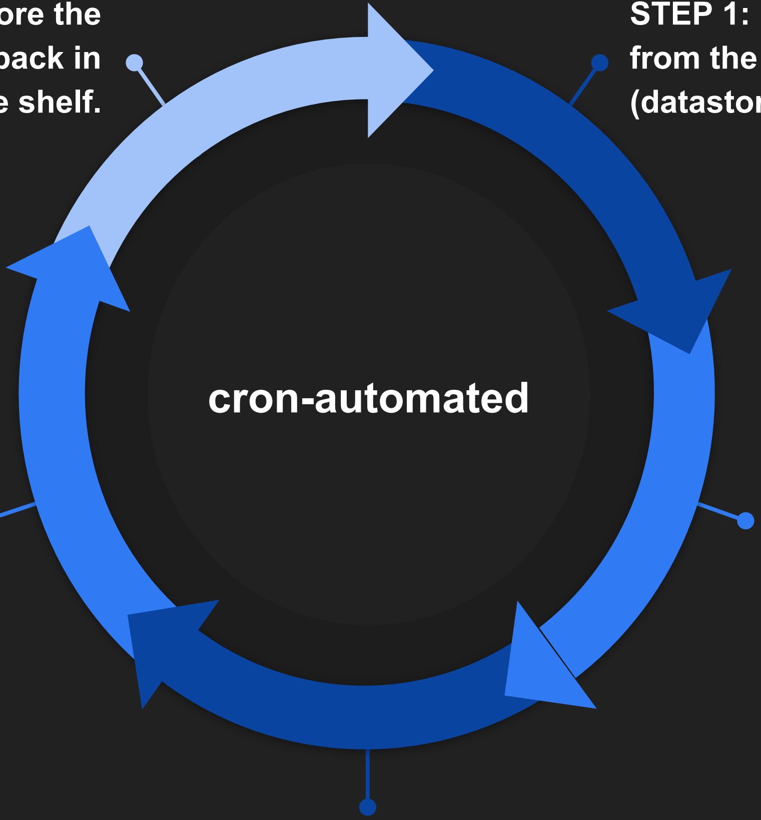
```
[det@xmasdata /]$ find /xmasdata/2018 -type f | wc -l
1289344
```

Whole

```
[det@xmasdata ~]$ du -h datastore
6.3G    datastore
```

Directory Trie

```
[det@xmasdata ~]$ du -h datastore
78M datastore
```



cron-automated

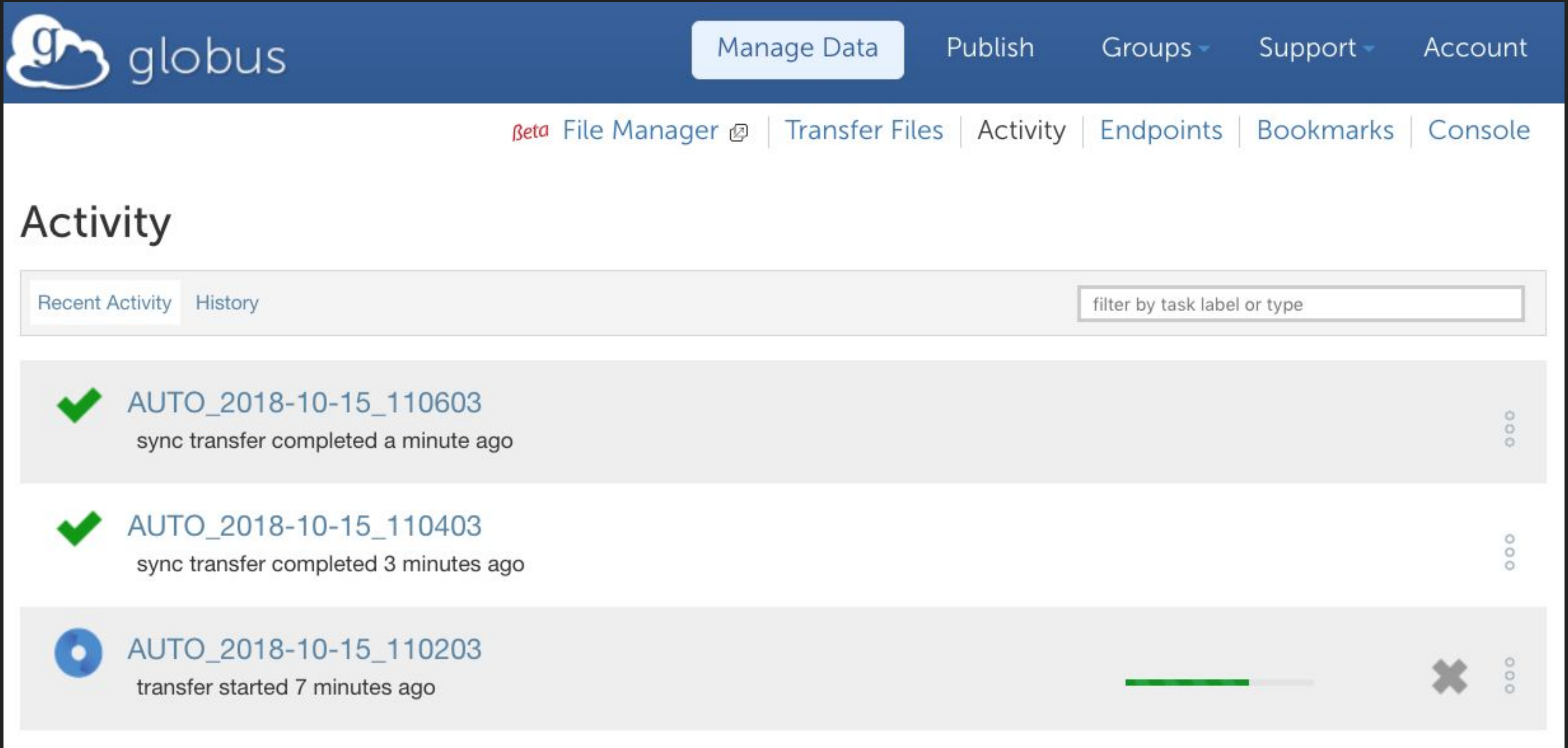
STEP 1: Load the trie from the shelf (datastore).

STEP 2: Scan the directory and add new files to the trie.

STEP 3: Iterate over the trie, generating transfer paths.

STEP 4: Transfer necessary files.

STEP 5: Store the updated trie back in the shelf.

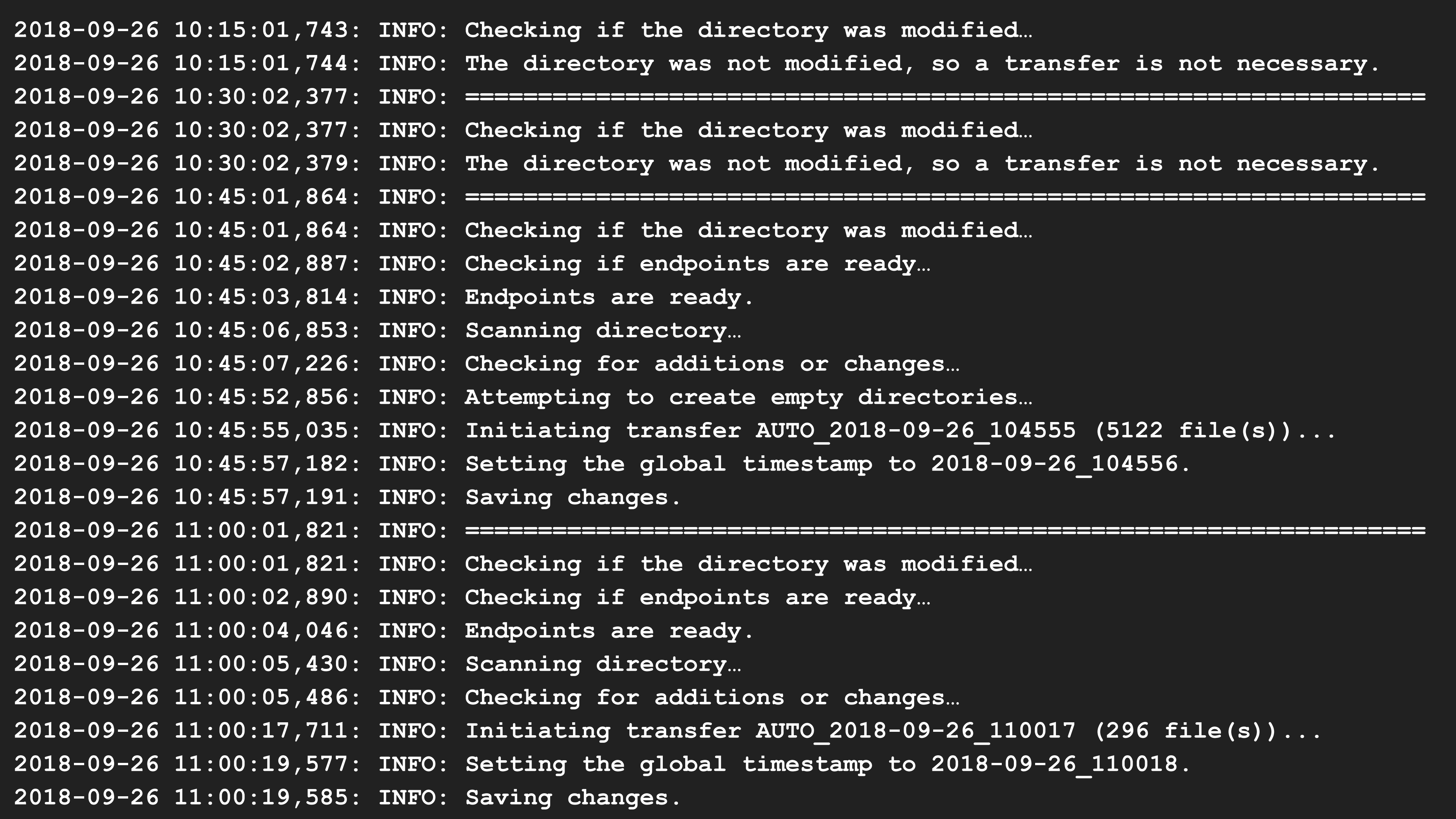


Activity

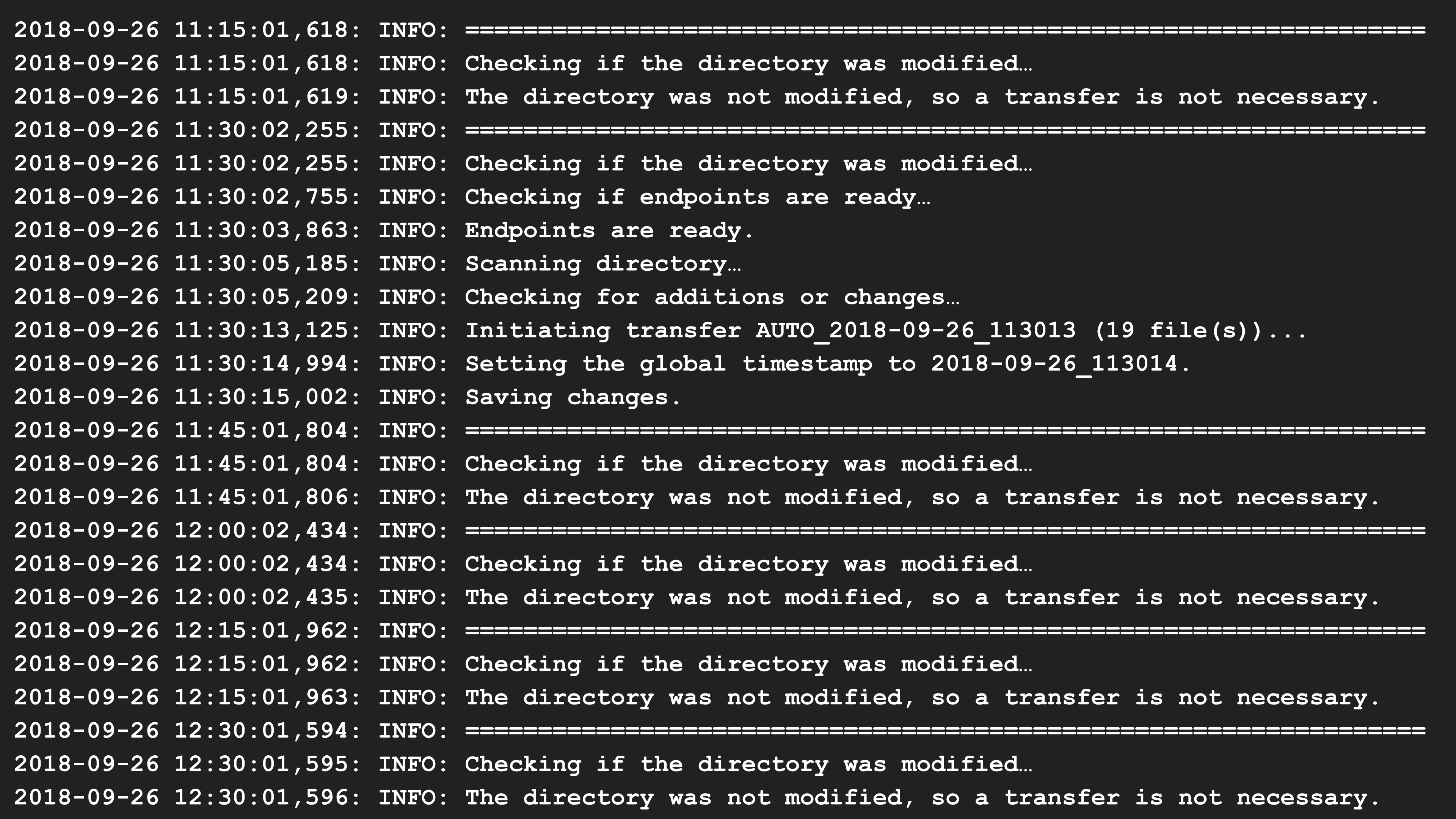
✓ AUTO_2018-10-15_110603
sync transfer completed a minute ago

✓ AUTO_2018-10-15_110403
sync transfer completed 3 minutes ago

AUTO_2018-10-15_110203
transfer started 7 minutes ago



```
2018-09-26 10:15:01,743: INFO: Checking if the directory was modified…
2018-09-26 10:15:01,744: INFO: The directory was not modified, so a transfer is not necessary.
2018-09-26 10:30:02,377: INFO: ==================================================================
2018-09-26 10:30:02,377: INFO: Checking if the directory was modified…
2018-09-26 10:30:02,379: INFO: The directory was not modified, so a transfer is not necessary.
2018-09-26 10:45:01,864: INFO: ==================================================================
2018-09-26 10:45:01,864: INFO: Checking if the directory was modified…
2018-09-26 10:45:02,887: INFO: Checking if endpoints are ready…
2018-09-26 10:45:03,814: INFO: Endpoints are ready.
2018-09-26 10:45:06,853: INFO: Scanning directory…
2018-09-26 10:45:07,226: INFO: Checking for additions or changes…
2018-09-26 10:45:52,856: INFO: Attempting to create empty directories…
2018-09-26 10:45:55,035: INFO: Initiating transfer AUTO_2018-09-26_104555 (5122 file(s))...
2018-09-26 10:45:57,182: INFO: Setting the global timestamp to 2018-09-26_104556.
2018-09-26 10:45:57,191: INFO: Saving changes.
2018-09-26 11:00:01,821: INFO: ==================================================================
2018-09-26 11:00:01,821: INFO: Checking if the directory was modified…
2018-09-26 11:00:02,890: INFO: Checking if endpoints are ready…
2018-09-26 11:00:04,046: INFO: Endpoints are ready.
2018-09-26 11:00:05,430: INFO: Scanning directory…
2018-09-26 11:00:05,486: INFO: Checking for additions or changes…
2018-09-26 11:00:17,711: INFO: Initiating transfer AUTO_2018-09-26_110017 (296 file(s))...
2018-09-26 11:00:19,577: INFO: Setting the global timestamp to 2018-09-26_110018.
2018-09-26 11:00:19,585: INFO: Saving changes.
```



```
2018-09-26 11:15:01,618: INFO: ==================================================================
2018-09-26 11:15:01,618: INFO: Checking if the directory was modified…
2018-09-26 11:15:01,619: INFO: The directory was not modified, so a transfer is not necessary.
2018-09-26 11:30:02,255: INFO: ==================================================================
2018-09-26 11:30:02,255: INFO: Checking if the directory was modified…
2018-09-26 11:30:02,755: INFO: Checking if endpoints are ready…
2018-09-26 11:30:03,863: INFO: Endpoints are ready.
2018-09-26 11:30:05,185: INFO: Scanning directory…
2018-09-26 11:30:05,209: INFO: Checking for additions or changes…
2018-09-26 11:30:13,125: INFO: Initiating transfer AUTO_2018-09-26_113013 (19 file(s))...
2018-09-26 11:30:14,994: INFO: Setting the global timestamp to 2018-09-26_113014.
2018-09-26 11:30:15,002: INFO: Saving changes.
2018-09-26 11:45:01,804: INFO: ==================================================================
2018-09-26 11:45:01,804: INFO: Checking if the directory was modified…
2018-09-26 11:45:01,806: INFO: The directory was not modified, so a transfer is not necessary.
2018-09-26 12:00:02,434: INFO: ==================================================================
2018-09-26 12:00:02,434: INFO: Checking if the directory was modified…
2018-09-26 12:00:02,435: INFO: The directory was not modified, so a transfer is not necessary.
2018-09-26 12:15:01,962: INFO: ==================================================================
2018-09-26 12:15:01,962: INFO: Checking if the directory was modified…
2018-09-26 12:15:01,963: INFO: The directory was not modified, so a transfer is not necessary.
2018-09-26 12:30:01,594: INFO: ==================================================================
2018-09-26 12:30:01,595: INFO: Checking if the directory was modified…
2018-09-26 12:30:01,596: INFO: The directory was not modified, so a transfer is not necessary.
```

Issues

Time Delay

Users sometimes experienced a delay between the transfer time in the log and the appearance of the transfer on the web application.

Symlinks

Globus has some restrictions on symlinks, but the code outright rejects all symlinks.

Memory

The entire trie is loaded into memory when retrieved from the shelf.

Concurrency

`n` pairs of (config.py, datastore) are needed to handle `n` source directories.

Try It Yourself

1. Download the code: https://bit.ly/2OyAmoi.
2. Install the Globus SDK to Python 3.3+: **`pip install globus-sdk`**.
3. Get a Globus Client App and **`CLIENT_ID`**.
4. Fill in **`config.py`**. Test it using **`test_config.py`**.
5. Set the initial time using **`set_time.py`**.
6. Install a cron job: **`crontab -e`**.

Globus Google Drive Connector

- Unlimited storage for all Berkeley Lab users.
- Up to 750 GB of transfer / day.
- Details: https://bit.ly/2CP8J3P
- Insert UUID into `config.py` to automate transfers to/from Google Drive.

Questions?

Matthew Li | meli@lbl.gov