Big Data Analytics

Report on Globus File Transfer for Common Crawl

1. Introduction

The Common Crawl is an **open repository of web crawl data** collected over last 7 years which contains several petabytes of data. It contains web page data, meta data extracts and text extracts.

Academia and research constantly requires annotated or classified data which helps researchers carry out their experiments using the data. The data is in **WET**, **WARC** and **WAT** file formats stored as chunks of data on **Amazon Web Services - Public Data Sets** across multiple academic cloud platforms. We need this data for 2 Big Data Processing Projects:

- **LORELEI language classification**
- **Biological Data classification**

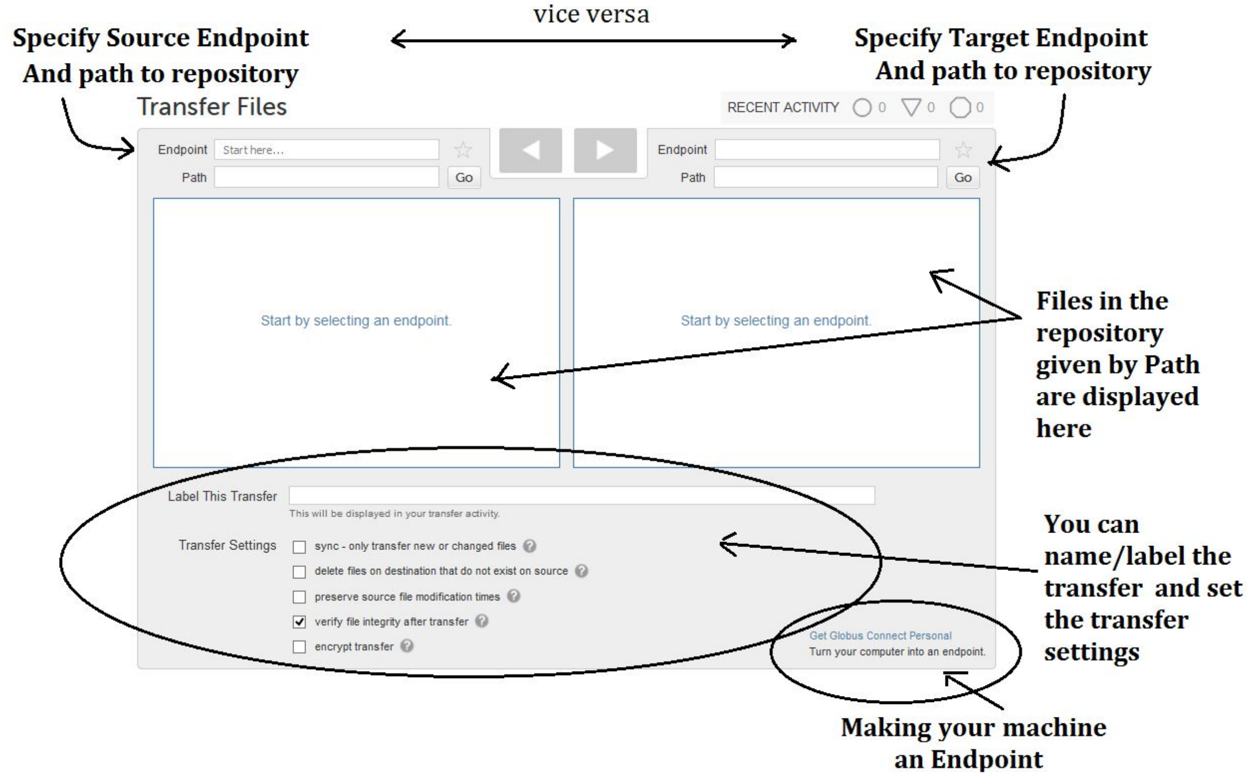
Globus File Transfer is needed to transfer the data to the **Pittsburgh Supercomputing Center's (PSC)** storage for carrying out experiments in the above mentioned projects.

2. Globus File Transfer

2.1. About Globus

The **Globus transfer** service provides high-performance, secure, file transfer and synchronization between endpoints. [Globus] Globus can address challenges that are encountered in file transfer such as: recover from transfer failure due to network unreliability, ensure integrity of transferred files, maintain required levels of security across different resources with different security configurations. Globus can be used with the web browser interface or the command line interface which contributes to its ease of use. It also enable in-place sharing of data, which means, users don't need to upload files to a separate cloud platform or create a different repository. They can directly share data files from their existing repositories. Endpoints need to be created on the machines on which the data resides and also on the destination machines of the file transfer. Globus also maximizes the bandwidth for the transfer and notifies when the transfer has completed or failed. Globus is Software as a Service (SaaS).

2.2. Initial Experiments with Globus Web Interface



vice versa
Transfer Files
RECENT ACTIVITY ○ 0 ▽ 0 ○ 0
Endpoint
Start here...
Path
Go
Endpoint
Path
Go
Start by selecting an endpoint.
Start by selecting an endpoint.
Get Globus Connect Personal
Turn your computer into an endpoint.

Label This Transfer

This will be displayed in your transfer activity.

Transfer Settings

- □ sync - only transfer new or changed files
- □ delete files on destination that do not exist on source
- □ preserve source file modification times
- ☑ verify file integrity after transfer
- □ encrypt transfer

Specify Source Endpoint And path to repository

Specify Target Endpoint And path to repository

Files in the repository given by Path are displayed here

You can name/label the transfer and set the transfer settings

Making your machine an Endpoint

Initially we transferred data between PSC Bridges endpoint and our own PC using the web browser. For that we had to create a personal endpoint as shown in the above picture bottom right corner. Once we get the Globus Connect personal and logging into the PSC bridges with XSEDE Authentication, we can transfer files from these endpoints.

We felt that we needed an API to transfer larger chunks of data at a time, especially because the transfer would not be between PCs and PSC end points, but between PSC and Common Crawl AWS s3 endpoints.

2.3 Experiments with Globus SDK for Python

The Globus SDK is a Pythonic Interface. It provides **Transfer API** and **Globus Auth API** which can be used to transfer files between two resources with different security configurations.

- Firstly, install the Globus SDK package using pip package manager using the command below.
  - pip install globus-sdk

This will install all the dependencies as well.

- Secondly, to use the API, we require an authentication using an access token. Hence, we need to create Authentication and Transfer tokens to accomplish file transfer. We can obtain it from here (Globus web Tokens). These tokens expire within 48 hours and are rendered unuseful.
- After creating the tokens, place them in the Config file named 'globus.cfg' located in the globus_sdk package on the computer in the following format
    - [general]
    - auth_token = <auth_token> here
    - transfer_token = <transfer_token> here
- The Transfer client needs the Endpoint IDs of the source and destination endpoints.
- Since we have very large data, in the order of petabytes, 48hours are not sufficient to transfer all chunks of data.
- We therefore tried refreshing the tokens after certain intervals of time and assigning a portion of the dataset - few chunks from the entire dataset to a transfer.
- We were able to achieve this goal but the 2 endpoints that participated in the transfer were PSC Bridges endpoint and our PCs. Transferring such large chunks to the Pittsburgh Supercomputing Centre required some configurations that the Globus SDK was not able to provide conveniently.
- We therefore dropped the idea of using the Globus SDK and thus any API provided by Globus until we were able to figure out the transfer protocols of PSC.

2.4 Common Crawl S3 endpoint creation

Though the common crawl data could be downloaded directly using http links (with help of tools like wget), the magnitude of the corpus makes it really hard to setup a reliable data transfer. We started using the globus service for that reason.

As discussed earlier, globus can transfer data from one globus endpoint to another endpoint. So in our case, the goal was to transfer data from the *common crawl s3 endpoint* to the *PSC Bridges endpoint* where we plan to store the whole data and process it.

Common Crawl corpus is hosted on *Amazon S3* (more information available at this link: https://aws.amazon.com/public-datasets/common-crawl/), organized by months. Each month has around 40-80 TB of data (August 2016 -> 83 TB, July 2016 -> 45 TB) and there are roughly 3 - 4 years of data which we'll be downloading, which requires approximately 2 Petabytes of storage space in the PSC Bridges Supercomputer in order to work on the current projects. Since we have already tested the project code on smaller data on our local clusters or on Amazon AWS and the next step would be to the run the code on the common crawl corpus.

To achieve the transfer the first step is to setup the globus S3 endpoint for the common crawl data. We tried following the steps at this link :

https://docs.globus.org/how-to/amazon-aws-s3-endpoints/ , but later realized that the creation of endpoints **requires a globus subscription**. After a few days of work, we figured that we can get the **globus subscription** from being members of the **XSEDE Plus Sponsor** group on globus. This gave us enough permissions to create a generic globus endpoint, **but not S3 endpoint.**

We then resorted to the common crawl community for creating a read only s3 endpoint which we can use to get the data transfer done, but they were not into globus endpoints. They did offer to a create a read-only access policy but that was already existing for the common crawl s3 bucket.
(https://groups.google.com/forum/#!topic/common-crawl/UAz_bPzeOHo)

Derek Simmel from PSC who was really active in helping us from the beginning managed to get enough permission to create a **managed endpoint.** Managed Endpoint is a type of endpoint which can only be created by the subscription managers for a subscribers group which in our case is the XSEDE Plus Sponsers group. S3 endpoint falls under the managed endpoints category. Given that we do not own the common crawl s3 data, we were not sure if the S3 endpoint we create would work. We realized that if we create a S3 policy with read only access to an AWS bucket then the endpoint we create need not be for a bucket which we own. So in this way, Derek was able to create an S3 endpoint and give us the permission to access the common crawl data(currently the permission on the endpoint is given to nbadam@andrew.cmu.edu).

Few steps to create the S3 endpoint (from an account which is a subscription manager):

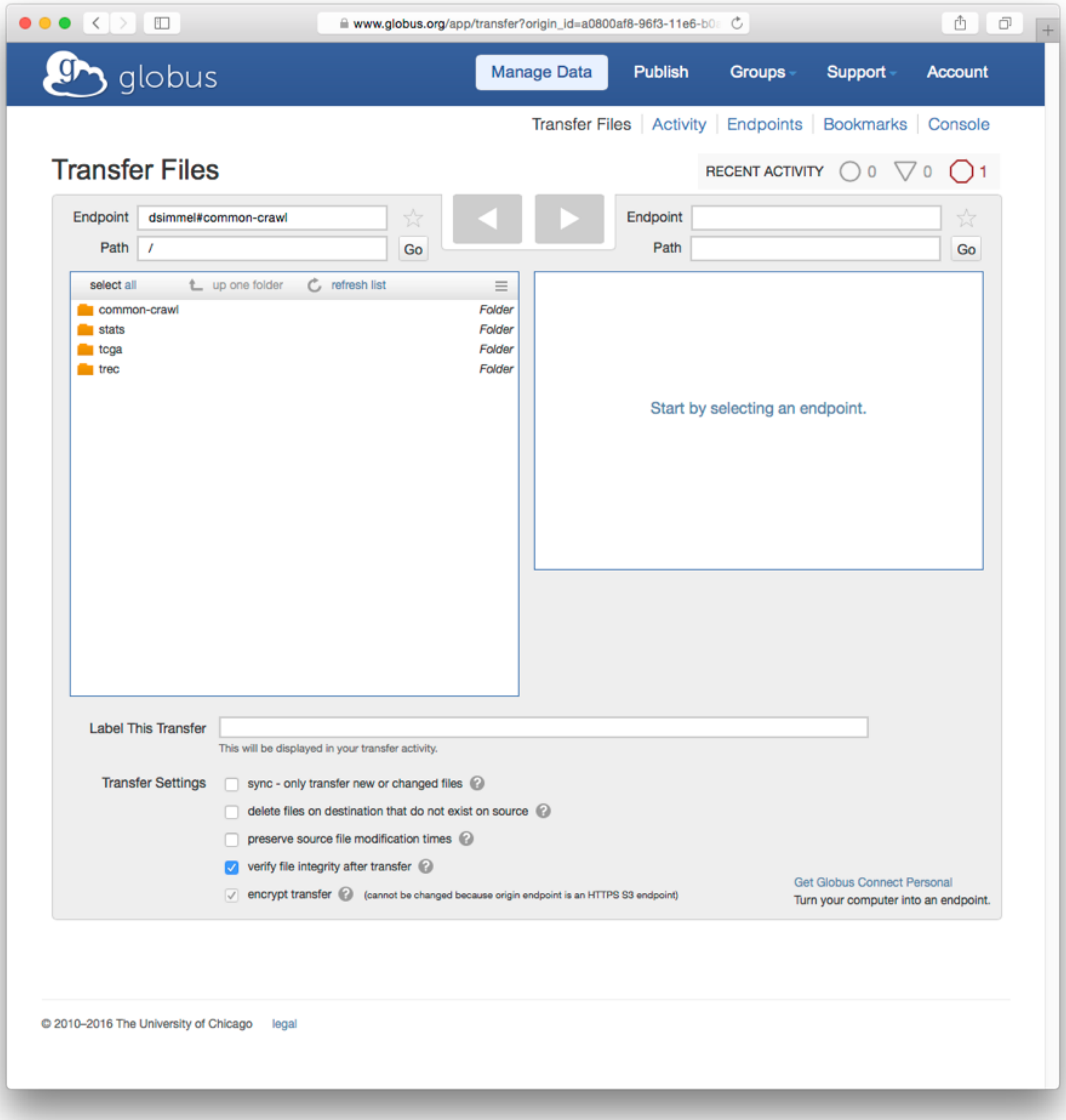
https://docs.globus.org/how-to/amazon-aws-s3-endpoints/

In Part A, skip steps 2-6 (Creating a policy): standard policy available works..

In Part A, steps 7-13 (creating a role): step 12 (Attach Policy Page): add AmazonS3ReadOnlyAccess - the ARN for this is arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess.

In Part B, step 2 (specifying the region specific URL for the S3 bucket): Use the following link https://s3.amazonaws.com/aws-publicdatasets

Then, activate the endpoint and test it out from web interface or the CLI.



Transfer Files

Endpoint: dsimmel#common-crawl

Path: /

common-crawl	*Folder*
stats	*Folder*
tcga	*Folder*
trec	*Folder*

Endpoint:

Path:

Start by selecting an endpoint.

Label This Transfer:

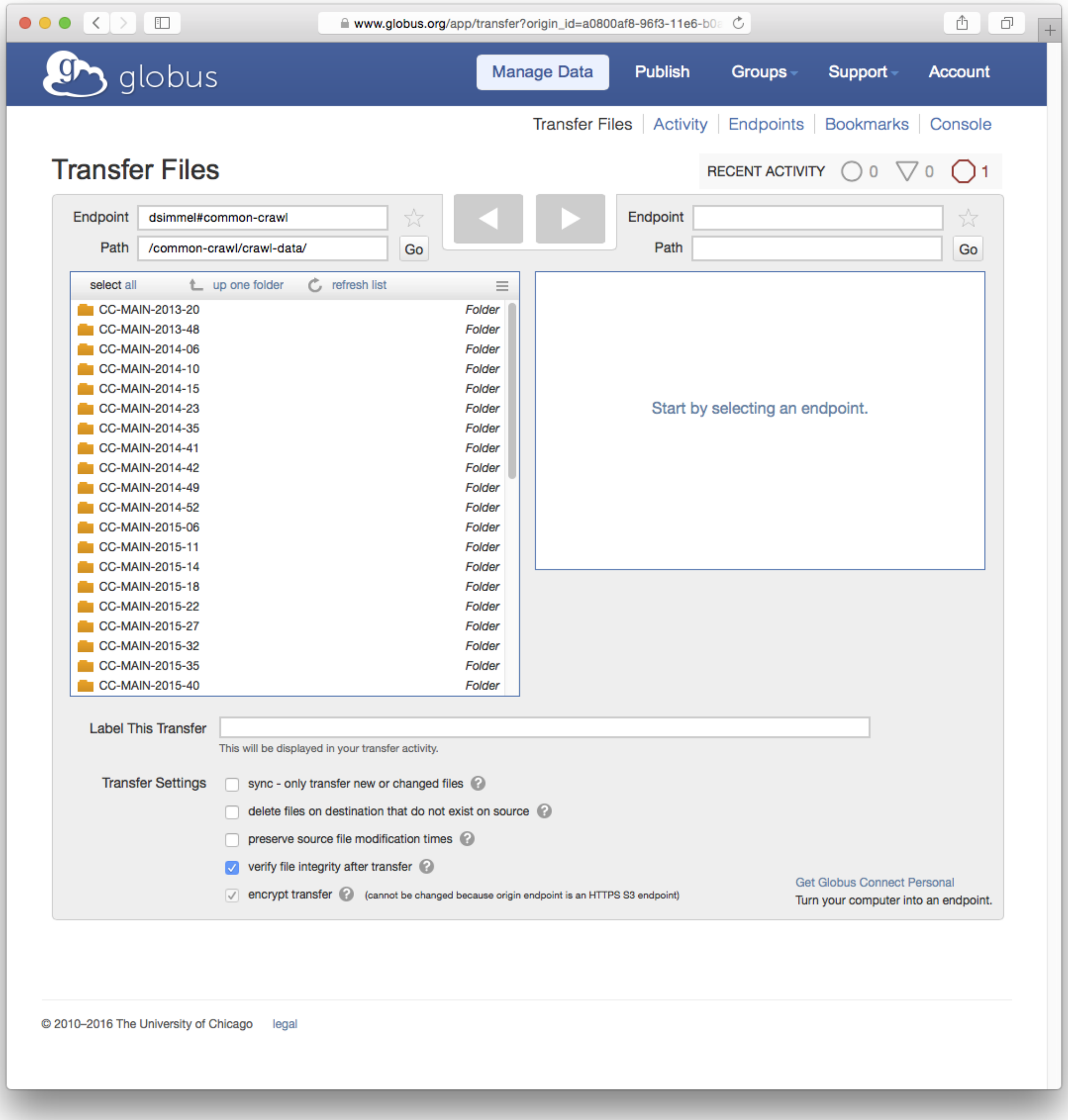
This will be displayed in your transfer activity.

Transfer Settings

- □ sync - only transfer new or changed files
- □ delete files on destination that do not exist on source
- □ preserve source file modification times
- ☑ verify file integrity after transfer
- ☑ encrypt transfer (cannot be changed because origin endpoint is an HTTPS S3 endpoint)

Get Globus Connect Personal
Turn your computer into an endpoint.

© 2010–2016 The University of Chicago legal



Transfer Files

Endpoint: dsimmel#common-crawl
Path: /common-crawl/crawl-data/

CC-MAIN-2013-20	Folder
CC-MAIN-2013-48	Folder
CC-MAIN-2014-06	Folder
CC-MAIN-2014-10	Folder
CC-MAIN-2014-15	Folder
CC-MAIN-2014-23	Folder
CC-MAIN-2014-35	Folder
CC-MAIN-2014-41	Folder
CC-MAIN-2014-42	Folder
CC-MAIN-2014-49	Folder
CC-MAIN-2014-52	Folder
CC-MAIN-2015-06	Folder
CC-MAIN-2015-11	Folder
CC-MAIN-2015-14	Folder
CC-MAIN-2015-18	Folder
CC-MAIN-2015-22	Folder
CC-MAIN-2015-27	Folder
CC-MAIN-2015-32	Folder
CC-MAIN-2015-35	Folder
CC-MAIN-2015-40	Folder

Endpoint:
Path:

Start by selecting an endpoint.

Label This Transfer:
This will be displayed in your transfer activity.

Transfer Settings
- □ sync - only transfer new or changed files
- □ delete files on destination that do not exist on source
- □ preserve source file modification times
- ☑ verify file integrity after transfer
- ☑ encrypt transfer (cannot be changed because origin endpoint is an HTTPS S3 endpoint)

© 2010–2016 The University of Chicago

2.5 Data Movement

As discussed earlier the data that need to be transferred is atleast 2 TB and can be much more higher as new data keeps coming in monthly. With the help from Derek, we spotted a few issues with the transfer speed from Amazon S3 endpoints to PSC Bridges endpoint. Currently we were only able to get around 25-50 MB/s, which means the transfer can end up running for days or months, even though we setup parallel transfers to improve the data download speeds. Derek has downloaded August 2016 and July 2016 data to his directory in the pylon

We have been using the globus web interface to move the data which is a simple drag and drop.