Large File Ingest and Retrieval

- Ingesting Large Files via the REST API
- Serving Large Files via Filesystem Federation
- Direct Comparison of Different Transfer Methods
    - Comparison of Upload and Download Times for Different Transfer Methods
    - Copying Files Between Federated Filesystem and Repository Storage
- Range Retrieval

Large files can be uploaded via the REST API, or projected into the repository using filesystem federation. Transfer times for uploading to the repository via the REST API are about the same as copying using NFS, and moderately faster than using SCP. Uploading via the REST API to a federated filesystem is significantly slower and requires a large temp directory capacity.

Java command-line options and System properties can be used

- `-Xmx2048m` – maximum memory Java can use
- `-Dfcrepo.home=/path/to/data` – set the directory for permanent data
- `-Djava.io.tmpdir=/path/to/tmpdir` – set the directory for temp files. Data uploaded to a federated filesystem via the REST API is written to a temp file in this directory before being moved to the federated filesystem, so this directory should have enough free space for the largest files you will upload.

Ingesting Large Files via the REST API

Based on the tests below, we believe arbitrarily-large files can be ingested and downloaded via the REST API (tested up to 1TB). The only apparent limitations are disk space available to store the files, and a sufficiently large Java heap size.

To enable fast access to large files, it is necessary to set **"contentBasedSha1" : "false".** Otherwise the repository will run a SHA1 on the content for identification that could take hours when reaching into the range of > 50Gb. For more on this benchmarking see: Design - LargeFiles.

- Platform: lib-devsandbox1.ucsd.edu (all data on NAS to handle large files)
- Repository Profile: Minimal
- Workflow Profile: Upload/Download Roundtrip

File Size	Upload	Download
256GB	3h51m34s (18.87MB/sec)	43m09s (101.25MB/sec)
512GB	7h49m43s (18.60MB/sec)	1h29m15s (97.90MB/sec)
1TB	15h41m21s (18.57MB/sec)	3h21m44s (86.63MB/sec)

Serving Large Files via Filesystem Federation

Based on the tests below, we believe arbitrarily-large files can be projected into the repository via filesystem federation and downloaded via the REST API (tested up to 1TB). The only apparent limitations are disk space available to store the files, and a sufficiently large Java heap size.

- Platform: lib-devsandbox1.ucsd.edu (all data on NAS to handle large files)
- Repository Profile: Minimal, with filesystem federation:

```
"externalSources" : {
    "filesystem" : {
        "classname" : "org.fcrepo.connector.file.FedoraFileSystemConnector",
        "directoryPath" : "/mnt/isilon/fedora-dev/federated",
        "projections" : [ "default:/projection => /" ],
        "readonly" : true,
        "addMimeTypeMixin" : true,
        "contentBasedSha1" : "false"
    }
}
```

File Size	Download
256 GB	1h09m26s (62.92MB/sec)
512 GB	2h00m15s (72.67MB/sec)
1 TB	3h57m25s (73.61MB/sec)

Direct Comparison of Different Transfer Methods

Based on the tests below, we believe arbitrarily-large files can be uploaded and downloaded via the REST API, using either repository storage or a federated filesystem (tested up to 1TB). The only apparent limitations are disk space available to store the files, temp directory capacity, and a sufficiently large Java heap size.

- Platform: lib-devsandbox1.ucsd.edu (all data on NAS to handle large files)
- Repository Profile: Federation
- Workflow Profile: Repository/Federation/NFS/SCP Comparison

Comparison of Upload and Download Times for Different Transfer Methods

Transfer Method	File Size	Upload	Download
REST API (Federated)	1TB	732 min (84 GB/sec)	246 min (250 GB/sec)
REST API (Repository)	1TB	339 min (181 GB/sec)	250 min (246 GB/sec)
SCP	1TB	383 min (160 GB/sec)	
NFS	1TB	336 min (183 GB/sec)	

Copying Files Between Federated Filesystem and Repository Storage

Source	Destination	File Size	Copy Time
Repository storage	Federated filesystem	1TB	402 min (153 GB/sec)
Federated filesystem	Repository storage	1TB	345 min (178 GB/sec)

Range Retrieval

Retrieving a byte range is supported and has been tested with 1TB files for both repository storage and federated filesystem. There is an integration test in the standard test suite for verifying that range retrieval works. By default, this test uses a small binary size to avoid slowing down the test suite, but the size is configurable so it is easy for a developer to test files as large as local disk space allows.