Fedora 6.0 Performance and Scale Criteria

This page is being used to capture performance and scale related expectations for Fedora: 6.0. The following table is a summary; more detail is captured below.

Institution	Repository Size	Number of Objects	Ingest Rate	Access Response
Columbia University	10 PB (external)	25 million	No worse than Fedora 3	No requirements
National Library of Medicine	70 TB	20 million	10K objects / hour	No requirements
UNC Chapel Hill	200 TB	10 million	< 50ms / RDF Container	Similar to Fedora 4/5
Berlin State Library	2 PB	100 million (multiple Fedoras)	10K objects / hour	No requirements
Zuse Institute Berlin	100 TB	20 million	6K objects / hour	20ms / object
Saxon State Library Dresden	1 PB	30 million	~1 object / s	sub-second latency

If you have performance and scale expectations please provide details in whatever granularity makes sense regarding the composition and needs of your current or expected Fedora repository. You can edit this page and add your institution below.

Relevant points of detail may include such areas as:

- Repository size in number of objects and/or number of bytes
- Expected ingest rates
- Expected access response times
- Migration scenarios: size and time expectations
- etc.

Columbia University

1. Number of Metadata Items/Objects - up to 25 million items over the next 5 years
2. Storage (external) - up to 10 petabytes, we typically don't pull from Fedora
3. Access Response - Discovery & Item Level View - end-user facing - performance mitigated via local SOLR indices, limited use cases for item level view
4. Write Performance - Real-time CRUD for single object - sub-second response (staff-facing)
5. Ingest Rates - Batch processing - faster is better, but less stringent requirements - no worse than Fedora 3
6. Migration Scenario - Less about technical requirements for speed than staff time to prepare, migrate and validate, and how much time system is unavailable for CRUD by staff members. Some validation/reassurance that migration can scale horizontally using multiple threads/processors /memory. Some metrics to understand time to migrate for 1 million objects, 2 million objects, 5 million objects, etc.

National Library of Medicine

1. Currently 9M objects, 90M datastreams, 70 TB. Up to 20M objects over the next 5 years. Currently these datastreams are generally loosely coupled (by reference, type E/R), so that most of the 70 TB is not directly managed by Fedora 3.
2. Access requirements
3. Expected ingest rates: Approx. 10K objects per hour would be nice. This would allow us to perform routine batch ingests of 20K-100K objects within a day.
4. Side-loading will likely be an important use case, as this is essentially our current approach with Fedora 3. We prepare and locate all of the binaries in advance, compute a FOXML file in advance, and then notify Fedora to ingest the FOXML file.
5. Migration scenarios: It would be nice to be able to accomplish the Fedora portion of the migration (not including staff validation) in perhaps two weeks. Automated validation tools, and reporting tools, are important in giving confidence that the migration was successful. Parallelization would be helpful but is not critical; we explored parallel ingest to Fedora 3 in the past with limited success. Complete and successful migration, with validation, is paramount, and is more important than the migration time.

University of North Carolina at Chapel Hill Libraries

1. Number of objects - Currently around 800k repository objects (roughly 4 million fedora container resources), which will grow by about 2 million repository objects in the next few years (~10 million fedora resources). There are around 4 million datastreams, including original files and metadata files. I would estimate the number of datastreams would grow by around 9-10 million.
2. Storage - Currently around 40tb, stored externally. Expected to grow by 130-150tb.
   a. Ideally, OCFL overhead would not be massively larger than the overhead of FOXML documents in earlier versions, but it is difficult to give exact metrics.
   b. We do not currently use S3 storage for files stored by Fedora, but this is a likely future use case.
3. Access Requirements - No slower than Fedora 3, preferably similar to Fedora 4/5. HEAD requests should be very efficient as we use them extensively for caching and verification purposes.
4. Write Performance - Similar to Fedora 4/5. Our model involves multiple RDF resources to represent a single repository object, so maintaining < 50ms times to create small RDF resources would be important. Our writing of binary resources currently happens outside of Fedora since we use external binaries. It's not clear if we would switch to using internal binaries in the future to take advantage of OCFL.
5. Sideloading - we do not currently have plans to use this feature actively.

6. Migration scenarios - We will be migrating a Fedora 5 instance within the next few years, with some portion of the projected growth listed above. There would be some adjustments to our model to account for ArchivalGroups, and consideration of whether to continue using external binaries. Otherwise, the modeling would likely be the same. We will also be migrating a Fedora 4 Hyrax instance with 100k objects, but that will likely be with the Hyrax tooling when it exists.

Berlin State Library

1. Number of objects; up to 100 million, split into various Fedora instances
2. Storage: around 2 P at the moment, slowly growing
3. Access response time: not that important. Most access to (meta)data is provided via Solr
4. Write performance, not worse than Fedora 4, 10 k objects per hour would be nice
5. Migration: should be faster than reingest, good and clear documentation needed, mentioning pitfalls
6. Sideloading: quite interesting, should be faster than ingest data, also good documentation essential

Zuse Institute Berlin

- Repository size in number of objects and/or number of bytes:

10 to 20 million objects, up to 100 TB (estimated) in the next couple of years

- Expected ingest rates:

Archivematica output is being batch-ingested with plastron. We would estimate that about 6000 objects/hour would be sufficient.

- Expected access response times:

Fast for front-end access, so around 20ms for both containers and binaries would be good. This is also needed for a couple of hundred consecutive GET requests on multiple resources (for grouped display of multiple child resources).

- Migration scenarios: size and time expectations:

migrating with probably around 100k objects (from Fedora 5.1.1), expected at 100 objects/minute.

Saxon State Library Dresden

1. Repository size in number of objects and/or number of bytes
    a. ~500k objects
    b. ~30 million individual resources
    c. ~11 TB online
    d. ~1 PB off-site tapes
2. Expected ingest rates
    a. metadata objects: ~1/s
    b. binaries: latency: <1s, speed: close to network bandwidth (non-blocking I/O)
3. Expected access response times
    a. sub-second latency
4. Migration scenarios: size and time expectations
    a. custom migration, ingesting new resources on masse; ~50k/day
    b. referencing a lot of externally stored content
    c. possibly parallel ingests
5. Side loading
    a. I'd rather not use this and use the API at all times. Only if performance degrades to much.