Real World Performance

Below are the results of performance testing comparing performance of Fedora-based applications with real-world data.

Plum Ingest

Ingesting a large book with 1000 100MB TIFF images, repeated with Fedora 4.5.1 release (based on Modeshape 4), and the experimental Modeshape 5 branch (in both cases, Fedora was configured to use the PostgreSQL database object store). Durations are reported as HH:MM:SS, for batches of 100 images loaded using Princeton's Hydra head, Plum.

Batch	Duration (Modeshape4)	Duration (Modeshape5)	Improvement
1	0:19:19	0:13:52	28.2%
2	0:27:03	0:23:19	13.8%
3	0:39:16	0:33:41	14.2%
4	0:52:13	0:43:43	16.3%
5	1:06:22	0:56:36	14.7%
6	1:23:29	1:10:46	15.2%
7	1:41:26	1:26:30	14.7%
8	2:02:22	1:43:08	15.7%
9	3:17:40	2:37:31	20.3%
10	3:47:48	3:10:14	16.5%

Retrieving Objects With Many Links to Repository Objects

Compared to objects with a large number of literal properties or URI properties, objects with a large number of links to repository objects are much slower. E.g., an object with 10,000 properties where the objects are literals or non-repository URIs can be retrieved in 200 milliseconds, but an object with 10,000 properties where the objects are repository objects takes 7-36 seconds, depending on the settings, storage backend, etc.

There are also significant differences between LevelDB and PostgreSQL/MySQL backends, with LevelDB being much faster: 7-10 seconds as opposed to 30+ seconds for the object with 10,000 links to repository objects.

Version/Branch	LevelDB	MySQL	PostgreSQL
4.5.0	8	n/a	n/a
4.5.1	10	43	36
master (a58f5a05)	7	32	29
modeshape5 (c177adc8)	n/a	89	30

See test scripts.

Testing initially focused on:

- using properties explicitly set on the object, as compared to IndirectContainers
- debugging the RDF-generation code that produces the IndirectContainer triples
- running under Tomcat instead of Jetty

However, those do not appear to significantly impact performance. So the process of looking up which node a proxy points to and converting the node reference to a URI seem to be the problem. The process is:

- List the children of a direct container and load each node.
- Load the node the proxyFor property points to.
- Convert the member node to a URI.

Each of these steps is reasonably fast (~1msec). But as the number of members grows, even 3 msec per member eventually adds up. For example, a collection with 10,000 members would take 30 seconds.

Some possible options for improving performance include:

- Caching nodes: this can improve the time to look up the member node and convert it to a URI.
- Using properties explicitly set on the collection object instead of proxies: this can eliminate the extra node lookup for loading the proxy node.
- Using Modeshape's internal query functionality: in theory this could be more efficient than iterating over the proxies. However, it appears that Modeshape uses the database as a document store, and so winds up loading all of the members anyway, with performance very similar to just iterating over all the children.