Fedora 4 Federation - File System Connector

Fedora Federation/Projection

Configuration Properties

Properties supported by Fedora 4

- `directoryPath` - base directory for all files shared with the repository
- propertiesDirectoryPath - (optional) a path, that causes computed properties to be stored in an external file structure
- `projections` - lists one or more mappings from the repository to the filesystem. The format is "{workspace}:{repository path} => {path relative to `directoryPath`}". See the section "Multiple Directories" below for how to handle multiple mappings.
- `contentBasedSha1` - controls how internal identifiers are computed for files. By default (`contentBaseSha1` = true), Modeshape computes the SHA-1 checksum of a file's content every time the file is accessed. For small files this creates a modest overhead. For large files, however, this dramatically reduces performance, since generating the checksum can take several seconds per gigabyte of data. For this reason, we recommend setting `contentBasedSha1` to false when serving files larger than 100MB
- `readonly` - controls whether the contents of the filesbase directory for all files shared with the repository are read-only (⚠ currently read-only is the only supported mode)
- `extraPropertiesStorage` - sets the format for storing "extra" properties (properties that can't be set using filesystem attributes). Recommended values are "json" for the current JSON properties format, or "none" for disabling extra properties. This property is ignored if propertiesDirectoryPath is set, since an external properties store will be used. (A warning or notice will appear in the logs indicating that the asserted preference here is overridden.)
- `cacheTtlSeconds` - the maximum time that cached entries are held before being refreshed. Setting to a low value will make changes to the filesystem (like adding new files) show up more quickly in the REST API. Setting to a higher value will improve performance for files that don't change often.

Properties supported by Modeshape, but not in Fedora 4

- inclusionPattern:Optional property that specifies a regular expression that is used to help determine which files and folders in the underlying file system are exposed through this connector. The connector will expose only those files and folders with a name that matches the provided regular expression (as long as they also are not excluded by the exclusionPattern). If no inclusion pattern is specified, then the connector will include all files and folders that are not excluded via the exclusionPattern.
- exclusionPattern:Optional property that specifies a regular expression that is used to help determine which files and folders in the underlying file system are not exposed through this connector. Files and folders with a name that matches the provided regular expression will *not*be exposed by this source.
- addMimeTypeMixin:A boolean flag that specifies whether this connector should add the 'mix:mimeType' mixin to the 'nt:resource' nodes to include the 'jcr:mimeType' property. If set to true, the MIME type is computed immediately when the 'nt:resource' node is accessed, which might be expensive for larger files. This is false by default.
- isQueryable:Optional property that specifies whether or not the content exposed by this connector should be indexed by the repository. This acts as a global flag, allowing a specific connector to mark it's entire content as non-queryable. By default, all content exposed by a connector is queryable.
- pageSize:***(Added in ModeShape 3.4.0.Final)*** Optional *advanced* property that controls the number of children that the connector should include in a single page; the default is 20. For example, if a folder contains 200 items (e.g., files or folders) and the page size is 20, then the connector will include in the document representing this folder only the properties of the folder and the first 20 items (that are readable, that satisfy the inclusion pattern, and that does not match the exclusion pattern). As additional children are needed (e.g., as the ModeShape client navigates or accesses the folder's child nodes), ModeShape will request additional pages, each with up to 20 items.

Fedora Federation/Projection Practice

Setup Fedora File System Connector

1. pull down the latest version of the source code (https://github.com/fcrepo4/fcrepo4)
2. update fcrepo-configs/src/main/resources/config/minimal-default/repository.json to look like the following:

```
"externalSources" : {
    "filesystem" : {
        "classname" : "org.modeshape.connector.filesystem.FileSystemConnect
        "directoryPath" : "/mnt/working_data",
        "projections" : [ "default:/federated => /" ],
               "contentBasedSha1" : "flase"
        "readOnly" : "true",
        "extraPropertiesStorage" : "json",
        "cacheTtlSeconds" : 5
    }
}
```

3. build everything (mvn clean install -DskipTests -Dcheckstyle.skip=true)
4. in fcrepo-webapp, run "mvn clean jetty:run" or move your war file to tomcat container

Experiment Server configuration

- 4 core cpu
- 16G RAM
- 32G System disk
- 2.5T data disk

Federated file system structure

There are 23 top-level directories in Fedora 4 repository. Among them, there is one federated node which has 24 top children as following:

1. big_files: has 42 big files (.mov or .mp4) and 3 sub-directories, each directory has 10 to 20 image files (total size of the big_files directory is about 900+G)
2. groups_of_1000: has 1000 sub-directories, each directory has 1000 small files (total size of the groups_of_1000 directory is 198G)
3. million_files: has 1 million files , no sub-directory, size is 198G
4. temp: 2 big files and 1 sub-directory
5. Smallfile-1 to smallfile_20: each has 10 to 100 sub-directories, each directory has 100 small files.

Experiments on federated file system

1. Before I ingested anything into fedora reository, I tested 'GET' http://birch.dlib.indiana.edu:8080/fedora4f/rest/federated/groups_of_1000, it is around 11s to 16s. But it is the baseline.
2. Then I ingested some xml files and image files into the repository. I also ingested the whole directory of groups_of_1000 into the repository, named binary1, so in http://birch.dlib.indiana.edu:8080/fedora4/rest/binary1, it has 1000 sub-folder, each folder has 1000 small files, just like groups_of_1000. Now the top level of the repository has 23 nodes (1 is the federated node, other 22 are internal nodes), and total has over 1 million files ingested into the repository.
3. Testing the 'GET' time for groups_of_1000 after the ingest done, the 'GET' time is from 15s to 23s. 'GET' the same structure internal node 'binary1' only take 0.12s to 3s.

Results:

	time to 'GET' federated directory	time to 'GET' same structure internal directory
when fedora repository is empty	11s to16s	--
when fedora repository has ingested 1 million files	15s to 23s	0.12s-3s
when fedora repository has ingested 1.5 million files	15s to 31s	0.5s to 15s

Another Option – HPSS connector

GPFS

We can use GPFS to map HPSS to mountable file system. But GPFS needs kernel buildup, that means if system upgrade, we have to rebuild it. Also it is uncertain for GPFS performance.

HPSS connector

if write our own HPSS connector, we need more investigations on the following:

- decide what kinds of the properties of the federated directory (at least top level) need to store/cache. This can help improving the performance.
- what is the best way can we retrieve federated documents/directories by ID, name, or path. Since HPSS has its own structure, but we need simulate it as file system structure. Or maybe we define our own structure here based on Brian's existing HPSS database?
- checksum (compute/get sha1?)
- read document from HPSS
- write document to HPSS?
- update document to HPSS
- remove document from HPSS
- touch parent (get parent)
- versioning/last modified date...

Those are open questions need more time to investigate.

Some useful links