Replication and Mirroring

Introduction

Using the multicast journal transport, it is possible to achieve replication or read-only mirroring of Fedora servers via journaling. Journal entries are created on a read/write Fedora server (the leader), and sent to other Fedora servers (followers) operating in journal recover mode.

Components

To implement replication between Fedora servers, at present three components are necessary:

1. A leader configured to log journals to all follower repositories
2. Any number of following Fedora servers to read and execute the journal entries
3. An RMI journal receiver deployed for each follower that receives journal messages, and writes journal files for the followers to play back.

Leader (Master) Fedora Repository

The leader operates in full read/write mode, and creates journals for any write operations requested of it. This leader must be configured to use the multicast journal writer, and all follower RMI receivers must be listed in the journal configuration in `fedora.fcfg`. All follower receivers must be running when the leader is started. Adding new receivers requires adding them to the server `fedora.fcfg`, and re-starting the leader Fedora instance. The leader configuration determines the size and age of all journal files. All transports, remote or local, produce identical journal files in response to journal events.

The multicast journal writer is able to designate any transport as 'crucial' (i.e. Fedora will no longer service write requests if it fails), and at least one transport must be designated as crucial. Common practice is to have a local file transport writing to the leader's file system that is designated as crucial, with all remote RMI transports as not crucial.

RMI Journal Receivers

RMI receivers listen for journal events from the leader, and write journal files into a directory containing journal entries. The size, duration, and name of each file is determine entirely by the leader server, since the server is responsible for sending 'open file' and 'close file' events. In the end, the journal files produced by a receiver are identical to that those created locally by the leader server or any other receiver. The receiver application is completely separate from Fedora and merely writes files to a directory that follower Fedora instances read at their convenience. Just as is the case for a Fedora instance logging journals to the local filesystem, journals that are open and still being written to will be prefixed with an _underscore.

The RMI receiver application is distributed as a separate download in the Fedora distribution, or may be built from source by the `rmi-journal-receiver` Maven build. The receiver itself is a single executable jar file `RmiJournalReceiver.jar`. When invoked, it will expect a command-line argument which names the directory in which it deposits files, as in

```
java -jar RmiJournalReceiver.jar /path/to/journals
```

The RMI receiver will run continuously, and will print any logging error messages to the standard output console. In production, it is recommended to run this as a daemon or service, and direct the output to a log file.

Journal events are pushed from the server to the follower. As such, the machines on which the RMI receiver is deployed must have two ports available: 1099 and 1100. Firewalls on follower machines must allow connections to these ports.

Follower (Slave) Fedora Repositories

Follower repositories are simply Fedora instances that are configured to run in recover mode, and use a following journal reader. In their `fedora.fcfg`, the journal reader must be specified as either `fedora.server.journal.readerwriter.multifile.MultiFileJournalReader` or `fedora.server.journal.readerwriter.multifile.LockingFollowingJournalReader`. These readers continuously poll a directory for the presence of journal files, and execute journal playback whenever a non-closed journal file becomes available. The journal directory should be the same directory that the RMI journal receiver is configured to write to.

While the RMI receiver must run continuously, the follower Fedora server can be stopped and started at will. The consequence of shutting down a follower Fedora is merely that journal files will pile up in the specified directory. When started, the follower Fedora will quickly play back journals and empty the directory until caught up.

The follower Fedora server is always in read-only mode. Query and read-only API operations are available at all times. Because journal files are read only when they are closed, any data written to the journal file will not appear in the follower until its file is closed, which depends on the leader configuration. Small journal file size limits and shorter lifetimes will result in the followers being closer to the leader state at any point in time.

Also note that the follower Fedora, when playing back journals, behaves exactly as the leader server when it initially received each request. This has implications for JMS. For example, if a follower has messaging enabled, each journal operation _will_ result in a JMS message. If the follower has a Gsearch index, it will update from the follower exactly as one for the leader instance would. Thus, it is possible to replicate entire stacks of services around each follower.

Scenarios

The replication scheme described here is useful in a number of situations:

Backups

Because data in Fedora is distributed between the filesystem and multiple databases (object registry database, resource index, etc), it is nearly impossible to obtain a completely consistent snapshot/dump of the all Fedora content from a live system. Thus, it is helpful to deploy a follower that can be shut down as necessary to allow for database and filesystem dumps to be assembled and archived - allowing the leader to run uninterrupted.

Upgrades and Maintenance

Followers may be manually turned into leaders by swapping `fedora.fcfg` configuration and re-starting. In this fashion, a leader may be taken down and replaced by one of its followers with minimal downtime (about as long as it takes for Fedora to start), freeing up the machine for upgrades or other scheduled maintenance. Once finished and caught up on journals as a follower, it may be switched back to a leader. Three servers allow for maintenance without loss of redundancy.

Load Balancing

For high-volume mostly read-only sites, followers may be used to provide data as part of a load-balancing scheme. The caveat here is that the followers will be slightly behind the leader as the journal files are written, closed, and processed. Small journal sizes and short lifetimes will mitigate this issue slightly.