Arvados - Feature #12314

[FUSE] Incremental collection subdirectory load

09/25/2017 01:37 PM - Peter Amstutz

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assigned To:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Collections are currently managed by the Collection API in the Python SDK and FUSE as a single unit. This means to access a single file in a Collection, the entire manifest is loaded, and associated ArvadosFile (and FuseArvadosFile) objects are created.

The manifest format is an unindexed flat file without normalization guarantees, so to do any operation requires at least one full scan of the manifest text. However, a full scan doesn't have to result in immediately creating a Python object for every file.

- Scan the manifest and determine if is one stream per directory (requires that no stream name is duplicated, and no file name contains '/'). If this isn't true, apply manifest normalization.
- Initialize the directory structure only and assign the raw manifest text of the stream associated with each directory (if it exists). Record the offset where the file list starts.
- On lookup, first check if a file with that name has already been loaded. If not, scan the file list from the stream and create a file object if found.
- On readdir, load the whole stream.
- Manage metadata cache for each directory individually. On cache eviction, discard file objects only, keep directory structure.

Questions:

- When can we forget the manifest text? Could remember only the offsets into the stream, allow discarding manifest text and refetching it.
- Works for collections by PDH which are strongly immutable. Collections referenced by uuid or are writable may still be loaded up front in order to incorporate updates (can't reference segments of manifest text by offset, will change).

History

#1 - 09/25/2017 02:08 PM - Peter Amstutz

- Description updated