

Arvados - Feature #2620

Keep supports I/O locking

04/16/2014 03:21 PM - Tim Pierce

Status:	Resolved	Start date:	04/01/2014
Priority:	Normal	Due date:	
Assigned To:	Tim Pierce	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	2014-05-07 Storing and Organizing Data		
Description			
Notes:			
<ul style="list-style-type: none">• Lock is per-volume. It's the sysadmin's problem to ensure volumes don't share a spindle.• Lock is in-process (not flock). It's the sysadmin's problem to ensure keep servers don't share a volume.• Enable per-volume lock with -serialize flag.• Exempt from lock behavior: HEAD, index, and the process of discovering whether a block is present during a GET or PUT. In other words, we aren't strictly limiting IO to one client per spindle: we're assuming the OS has good enough cache behavior to support hordes of readdir operations alongside a big read/write without (statistically) causing thrash.			
Subtasks:			
Task # 2695: Review 2620-keep-serialize-io			Resolved
Task # 2629: I/O locking unit tests			Resolved
Task # 2560: Serialize Keep I/O			Resolved

Associated revisions

Revision afe657ca - 04/18/2014 04:00 PM - Tim Pierce

Reorganize Keep code in preparation for dividing into modules.

Refs #2620

Revision 8202cb5f - 04/18/2014 04:54 PM - Tim Pierce

Adding a Volume interface and UnixVolume implementation.

refs #2620

Revision b66ad0b2 - 04/22/2014 11:20 AM - Tim Pierce

Bug fix: GetBlock must report a CorruptError immediately.

Refs #2620.

Revision da3e84a0 - 04/23/2014 06:09 PM - Tim Pierce

Added UnixVolume unit tests. (refs #2620)

Revision 93db9aec - 04/23/2014 10:32 PM - Tim Pierce

Moved the MD5 verification check back to GetBlock.

Policy tests such as checksumming belong in the Keep front end. The back end should be limited to moving data in and out of a volume. (refs #2620)

Revision 78b5bfcb - 04/23/2014 10:35 PM - Tim Pierce

Updated UnixVolume unit tests (refs #2620).

Revision dbce5332 - 04/24/2014 06:02 PM - Tim Pierce

Added -serialize flag.

Added IORequest and IOResponse types for communicating I/O requests over channels with a goroutine.

New IOHandler method on UnixVolume. IOHandler receives requests on a

command channel, handles them, and delivers responses. Whenever a UnixVolume is created with a non-nil queue, an IOHandler must be started to handle its requests.

UnixVolume methods Get and Put now handle external I/O requests. These methods serialize I/O requests if serialization is enabled for that volume; otherwise they call Read and Write directly.

New unit tests: TestGetSerialized and TestPutSerialized.

Refs #2620.

Revision b860994f - 04/25/2014 04:58 PM - Tim Pierce

Cleaned up unit tests. (refs #2620)

Added a MockVolume implementation to use in unit tests for the Keep front-end handlers.

Simplified IsValidLocator and keep_test.go:setup code.

Revision e05f165a - 04/25/2014 05:06 PM - Tim Pierce

Cleaned up unit tests. (refs #2620)

Added a MockVolume implementation to use in unit tests for the Keep front-end handlers.

Simplified IsValidLocator and keep_test.go:setup code.

Revision f37f7c06 - 04/25/2014 05:23 PM - Tim Pierce

Make sure an IOHandler is running when -serialize is on.

Use MakeUnixVolume to create new UnixVolumes, to ensure that an IOHandler is started whenever necessary. (refs #2620)

Revision f74cbd15 - 04/25/2014 05:47 PM - Tim Pierce

Reorganized Volume code for easier comprehension.

Refs #2620

Revision 30f834ee - 04/29/2014 03:24 PM - Tim Pierce

Incorporating code review comments (refs #2620)

- use filepath.Join to build file paths instead of fmt.Sprintf
- use ioutil.ReadAll to ensure that we read the whole file

Revision 57b46f9b - 04/30/2014 01:49 PM - Tim Pierce

Replaced KeepVolumes with a VolumeManager interface.

A VolumeManager interface may be used to set policy for writing to a set of Keep volumes. The sole implementation at present is RRVolumeManager, which implements simple round-robin scheduling; other implementations could include a manager which selects the least-loaded disk first, the fastest one, the one with the fewest pending writes, etc. etc.

Refs #2620

Revision a6439c7d - 04/30/2014 04:22 PM - Tim Pierce

Merge branch '2620-keep-serialize-io' (closes #2620)

History

#1 - 04/16/2014 03:35 PM - Ward Vandewege

- Assigned To set to Tim Pierce

#2 - 04/16/2014 03:43 PM - Tim Pierce

- Tracker changed from Bug to Feature

- Subject changed from *Keep I/O locking* to *Keep supports I/O locking*

An admin should be able to enable I/O locking: a per-spindle lock that prevents Keep from reading or writing to more than one file at a time. I/O operations on different spindles may still be executed concurrently.

#3 - 04/16/2014 03:45 PM - Tim Pierce

- Story points set to 3.0

#4 - 04/16/2014 03:45 PM - Tim Pierce

- Story points changed from 3.0 to 2.0

#5 - 04/16/2014 04:50 PM - Tim Pierce

- Target version deleted (*2014-05-07 Storing and Organizing Data*)

#6 - 04/16/2014 04:57 PM - Tim Pierce

- Target version set to *2014-05-07 Storing and Organizing Data*

#7 - 04/17/2014 06:03 PM - Tom Clegg

- Description updated

#8 - 04/30/2014 04:20 PM - Tim Pierce

- Status changed from *New* to *Resolved*

- % Done changed from 92 to 100

Applied in changeset `arvados|commit:a6439c7ddc3407193889386051f06df2a1e74ea9`.