Arvados - Feature #7748

[Data Manager] -dry-run command line flag: log how many blocks would be deleted/moved, but do not issue any changes to keepstore

11/10/2015 07:42 PM - Tom Clegg

Status:	Resolved	**Start date:**	11/26/2015
Priority:	Normal	**Due date:**	
Assigned To:	Radhika Chippada	**% Done:**	100%
Category:	Keep	**Estimated time:**	0.00 hour
Target version:	2015-12-02 sprint		

Description

This means literally do everything Data Manager currently does **except** writing to Keepstores (e.g., sending pull or trash lists).

Logs sent to the API server should indicate whether Data Manager was running in dry run mode.

Subtasks:

Task # 7871: Review branch: 7748-datamanager-dry-run	**Resolved**

Associated revisions

Revision a6650f13 - 12/02/2015 05:37 PM - Radhika Chippada

closes #7748
Merge branch '7748-datamanager-dry-run'

History

#1 - 11/16/2015 07:55 PM - Brett Smith

- Target version set to Arvados Future Sprints

#2 - 11/17/2015 07:50 PM - Brett Smith

- Description updated

- Category set to Keep

#3 - 11/26/2015 10:37 PM - Radhika Chippada

- Status changed from New to In Progress

- Assigned To set to Radhika Chippada

- Target version changed from Arvados Future Sprints to 2015-12-02 sprint

#4 - 11/26/2015 10:41 PM - Radhika Chippada

commit 0ec4051e60f34786bf7cf78f5b07f50796c68235

- Added dry-run command line argument and added test
- "Logs sent to the API server should indicate whether Data Manager was running in dry run mode" : Updated all references to arvLogger.Update to log os.Args in run_info
- While at it, also converted the last set of Fatalf's from datamanager/summary/file.go -> ReadData func into error notifications.

#5 - 11/30/2015 07:39 PM - Tom Clegg

At 0ec4051 on 7748-datamanager-dry-run

I don't think this quite accomplishes "log how many blocks would be deleted/moved" yet. Until BuildTrashLists we don't check which blocks are too new to delete; and until ComputePullServers we don't know how many blocks need to be pulled by each server.

How about, rather than interrupting singlerun(), putting the "if dryRun { log #blocks/bytes; return }" bit in SendTrashLists before preparing the PUT request?

Even though WritePullLists already just writes to a local file, we should do the same thing there: if dryRun, log #blocks/bytes that each server should pull, instead of writing the local file.

Is it really necessary to keep updating runInfo["args"] every time we log something else? os.Args doesn't change, so it seems we should be able to

just set it once in LogRunInfo and then leave it alone, like we do with runInfo["started_at"].

#6 - 11/30/2015 09:49 PM - Radhika Chippada

> How about, rather than interrupting singlerun(), putting the "if dryRun { log #blocks/bytes; return }" bit in SendTrashLists before preparing the PUT request?

Updated accordingly

> Even though WritePullLists already just writes to a local file, we should do the same thing there: if dryRun, log #blocks/bytes that each server should pull, instead of writing the local file.

Updated to log and not write file

> Is it really necessary to keep updating runInfo["args"] every time we log something else? os.Args doesn't change, so it seems we should be able to just set it once in LogRunInfo and then leave it alone, like we do with runInfo["started_at"].

Removed the redundant runInfo["started_at"] statements

#7 - 12/01/2015 03:11 PM - Tom Clegg

When "dryRun", like an error, has the effect of skipping the remainder of a block/goroutine, instead of indenting the entire "normal" part of the loop

- [redacted]

```
        // do stuff
        if dryRun {
                // ...
        } else {
                // rest of loop
        }
}
```

...we should just continue

- [redacted]

```
        // do stuff
        if dryRun {
                // ...
                continue
        }
        // rest of loop
}
```

Also, I'm not sure sending the entire pull/trash list in a log entry is a good idea. From the issue title I was expecting something to appear in the terminal. How about we *always* mention sizes in our server logs, like this:

- [redacted]

```
                // We need a local variable because Update doesn't call our mutator func until later,
                // when our list variable might have been reused by the next loop iteration.
                listLen := len(list)
        arvLogger.Update(func(p map[string]interface{}, e map[string]interface{}) {
            pullListInfo := logger.GetOrCreateMap(p, "pull_list_len")
            pullListInfo[host] = listLen
        })
    }
        if dryRun {
                log.Print("dry run, not sending pull list to service %s with %d blocks", host, len(list))
                continue
        }
        // ...regular code...
```

Logging "started_at" seems weird in these updates; it seems to mean "when last pull list would have been sent". (Is it just a copy/paste remnant?)

#8 - 12/01/2015 03:57 PM - Radhika Chippada

> When "dryRun", like an error, has the effect of skipping the remainder of a block/goroutine, instead of indenting the entire "normal" part of the loop ... continue

Updated accordingly. I went back and forth on this initially, but this is much better with much smaller diff

> Also, I'm not sure sending the entire pull/trash list in a log entry is a good idea. From the issue title I was expecting something to appear in the terminal. How about we always mention sizes in our server logs ...

Updated as suggested. My concern with just logging the size was, if an admin wants to know which blocks will be trashed / pulled, this information won't be available with logging just the size. If that is not going to be the case, then this is good enough.

> Logging "started_at" seems weird in these updates; it seems to mean "when last pull list would have been sent". (Is it just a copy/paste remnant?)

Looking at the logger implementation, it does not seem like a timestamp at which point the event occurred is not included unless it is included in the info being logged and hence I added it. However, an attribute such as "started_at" does not seem useful anyways because only the last time would be printed. If I really wanted it I should do something like "timestamp_for_<url>". Just removed it instead.

#9 - 12/02/2015 04:58 PM - Brett Smith

- Target version changed from 2015-12-02 sprint to 2015-12-16 sprint

#10 - 12/02/2015 05:29 PM - Tom Clegg

These (in WritePullLists) need to come before the call to arvLogger.Update. The comment is in the right place but these are one line too late:

```
+                                host := host
+                                listLen := len(list)
```

SendTrashLists is functionally correct but the comment should move down one line so it is adjacent to the shadowed variables it's explaining.

The rest LGTM.

#11 - 12/02/2015 05:38 PM - Radhika Chippada

- Target version changed from 2015-12-16 sprint to 2015-12-02 sprint

#12 - 12/02/2015 05:40 PM - Radhika Chippada

- Status changed from In Progress to Resolved

Applied in changeset arvados|commit:a6650f13fe461641defa4f281972df0ce1567594.