Arvados - Feature #16513

Get reference Keep performance numbers for Keep-on-S3

06/10/2020 02:15 PM - Ward Vandewege

Status:	Resolved	**Start date:**	06/15/2020
Priority:	Normal	**Due date:**	
Assigned To:	Ward Vandewege	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2020-07-01 Sprint		

Description

Subtasks:

Task # 16528: review 16513-keep-exercise-improvements	**Resolved**

Related issues:

Related to Arvados - Story #10477: [keepstore] switch s3 driver from goamz to...	**Resolved**	**11/08/2016**	
Related to Arvados - Feature #16518: [keep] Allow clients to set a header to ...	**New**		
Related to Arvados - Feature #16519: [keepstore] optimize md5sum calculations	**New**		
Blocks Arvados Epics - Story #16516: Run Keepstore on local compute nodes	**In Progress**	**10/01/2021**	**11/30/2021**

Associated revisions

Revision 9706aef4 - 06/19/2020 03:37 PM - Ward Vandewege

Merge branch '16513-keep-exercise-improvements'

refs #16513

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

History

#1 - 06/10/2020 03:50 PM - Ward Vandewege

- Related to Story #16514: Actionable insight into keep usage added

#2 - 06/10/2020 04:09 PM - Ward Vandewege

- Related to deleted (Story #16514: Actionable insight into keep usage)

#3 - 06/10/2020 04:09 PM - Ward Vandewege

- Blocks Story #16516: Run Keepstore on local compute nodes added

#4 - 06/12/2020 02:52 PM - Ward Vandewege

e710f1b2da3095d6152ac7f6ed1ffab8bfc2c0c7 on branch 16513-keep-exercise-improvements is ready for review.

#5 - 06/12/2020 03:46 PM - Ward Vandewege

- Target version set to 2020-06-17 Sprint

- Status changed from New to In Progress

#6 - 06/15/2020 09:16 PM - Tom Clegg

I have a few nits / suggested improvements but you could ignore them and/or merge e710f1b in the meantime.

Repeating the expression float64(bytesOut) / elapsed.Seconds() / 1048576 is a bit crufty. Should probably compute that once as rateOut and then use it 3 times.

We probably don't need 2 different stats reporting formats. We could print the header line at start, then print a CSV row once every stats-interval plus one at the end.

Printing the final summary on SIGINT/SIGALRM would be a nice touch. (then "alarm 60 keep-exercise ..." would work well, fwiw)

endChan could be a Timer rather than a Ticker. context.WithDeadline() and <-ctx.Done() would be another way to do it.

If we send the CSV data to stdout and logs to stderr, we'll be more ... | tee stats.csv -friendly.

#7 - 06/17/2020 03:38 PM - Ward Vandewege

- Target version changed from 2020-06-17 Sprint to 2020-07-01 Sprint

#8 - 06/17/2020 04:38 PM - Ward Vandewege

Tom Clegg wrote:

> I have a few nits / suggested improvements but you could ignore them and/or merge e710f1b in the meantime.
>
> Repeating the expression float64(bytesOut) / elapsed.Seconds() / 1048576 is a bit crufty. Should probably compute that once as rateOut and then use it 3 times.
>
> We probably don't need 2 different stats reporting formats. We could print the header line at start, then print a CSV row once every stats-interval plus one at the end.
>
> Printing the final summary on SIGINT/SIGALRM would be a nice touch. (then "alarm 60 keep-exercise ..." would work well, fwiw)
>
> endChan could be a Timer rather than a Ticker. context.WithDeadline() and <-ctx.Done() would be another way to do it.
>
> If we send the CSV data to stdout and logs to stderr, we'll be more ... | tee stats.csv -friendly.

I've implemented everything in cba1b4145e8fcc57a851839f77fd020e5aaff722, ready for another look.

#9 - 06/18/2020 05:43 PM - Tom Clegg

LGTM @ a5a6111e3, thanks!

#10 - 06/30/2020 08:43 PM - Ward Vandewege

Arvados version: 2.0.2; AWS VPC with S3 endpoint

Single-threaded write to Keep backed by S3: ~42 MiB/sec
Single-threaded read from Keep backed by S3: ~62 MiB/sec

Single-threaded write to S3 with a 3rd party client (s3-cli): ~46 MiB/sec
Single-threaded read from S3 with a 3rd party client (s3-cli): ~106 MiB/sec

It's worth noting that S3 and Keep are optimized for aggregate throughput. With X reader/writer processes, you would expect to see roughly X times the single thread performance, up to the capacity (CPU/bandwidth/memory) of the keepstores (and the clients, but these tend to be spread out over many machines).

That said, we have identified a few areas for future improvement:

a) Keep write to S3 does not currently use multipart writes, because the S3 library we use does not support it. Using multipart writes is recommended to increase write throughput. We are looking into adopting the official AWS S3 go library (#10477). Our Keep S3 backend predates the official AWS S3 go library.

b) Keep's single-threaded read performance: some of the slowdown is caused by the md5sum that Keepstore does on reading every block. We are considering adding an option to disable the md5sum on read in Keepstore (#16518). We are investigating additional performance improvements as well (e.g. #16519).

#11 - 06/30/2020 08:43 PM - Ward Vandewege

- Related to Story #10477: [keepstore] switch s3 driver from goamz to a more actively maintained client library added

#12 - 06/30/2020 08:43 PM - Ward Vandewege

- Related to Feature #16518: [keep] Allow clients to set a header to disable md5sum calculations in keepstore added

#13 - 06/30/2020 08:43 PM - Ward Vandewege

- Related to Feature #16519: [keepstore] optimize md5sum calculations added

#14 - 06/30/2020 08:43 PM - Ward Vandewege

- Status changed from In Progress to Resolved

#16 - 10/07/2020 02:11 AM - Peter Amstutz

- Release set to 25